SONiX Technology Co., Ltd.

SN8F5713 Series Datasheet

8051-based Microcontroller

SN8F5713SG
SN8F5713TG
SN8F5713JG
SN8F5712PG
SN8F5712SG
SN8F5711PG
SN8F5711SG
SN8F57131PG
SN8F57131SG
SN8F57112SG
SN8F57113SG

1 产品简介

1.1 功能特性

- ◆ 增强型 8051 微控制器：减少指令周期时间（高达 80C51 的 12 倍）
  - 高达 8MHz 的灵活 CPU 频率
  - 内部 32 MHz 时钟发生器(IHRC), 1 MHz 至 16 MHz 的晶振，以及外部同步时钟源可选
  - 1 组外部低速时钟 32.768 kHz 晶振
- ◆ 8 KB flash 存储器 (IROM)，支持 In-System Program 功能
- ◆ 256 字节内部 RAM (IRAM)
- ◆ 512 字节外部 RAM (XRAM)
- ◆ 10 个中断源，可控制中断的优先等级以及独立的中断向量
  - 7 个内部中断
  - 3 个外部中断：INT0, INT1, INT2
- ◆ 1 组 DPTR
- ◆ 2 组 8/16 位定时器，包括 4 种工作模式
- ◆ 1 组 16 位 PWM 发生器：
  - PWM 发生器有 8 个输出通道,带独立占空比控制功能
- ◆ 13+3 通道 12 位 SAR ADC
  - 13 个外部 ADC 输入通道
  - 3 个内部通道
  - 4 个内部参考电压（VDD/4V/3V/2V）
- ◆ 15 个通道的电容式触摸功能
- ◆ UART, I2C 接口，支持 SMBus
- ◆ 工作模式
  - NORMAL 模式
  - IDLE 模式
  - STOP 模式
- ◆ 支持片上调试:
  - 单线调试接口
  - 3 个硬件断点
  - 无限制软件断点
  - ROM 数据安全/保护
- ◆ 看门狗和可编程外部复位
- ◆ 1.8/2.4/3.3-V 低电压检测
- ◆ 工作电压范围大(1.8 V – 5.5 V)，温度范围 (-40 °C to 85 °C)
- ◆ 硬件乘除法单元

1.2 应用领域

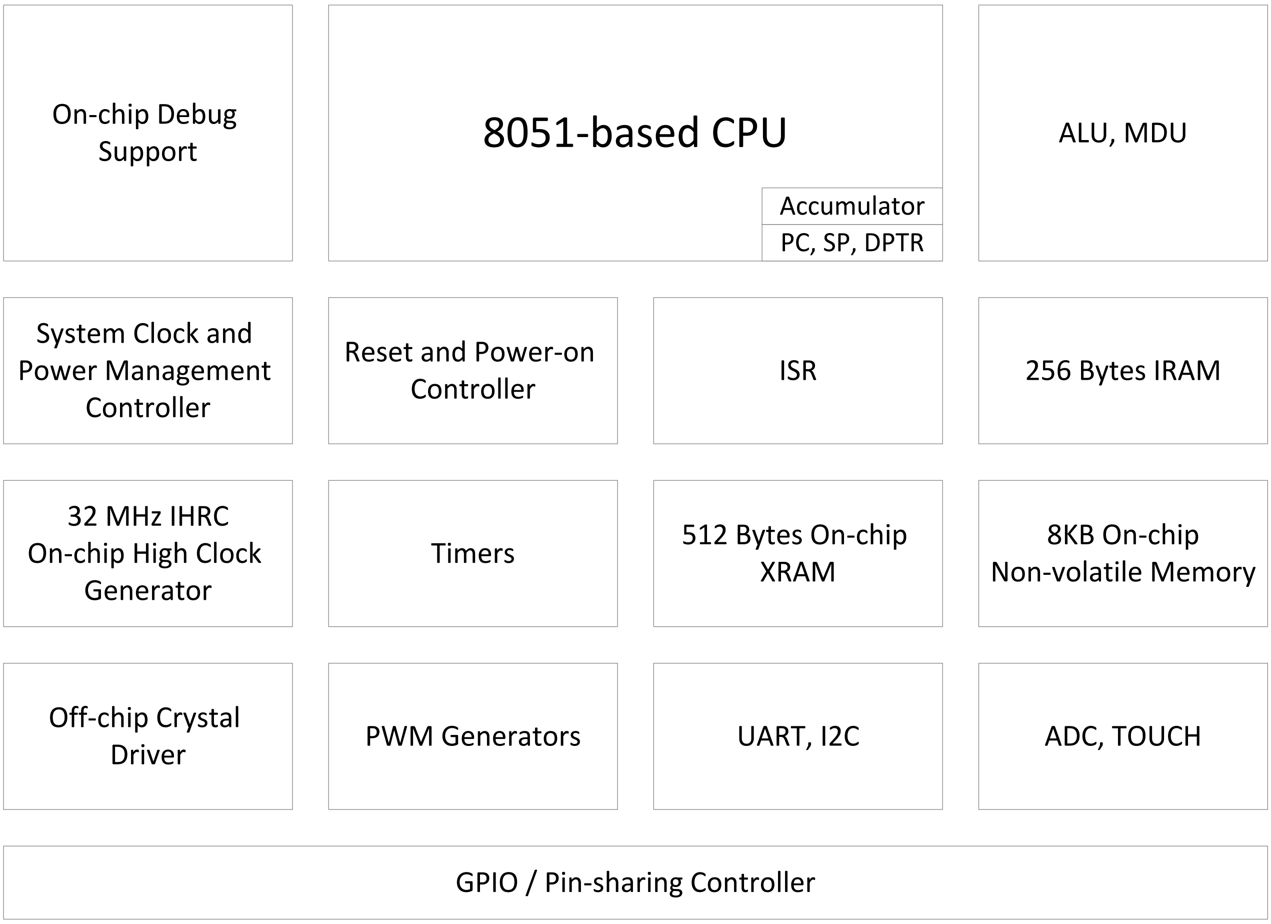
- 家用自动化产品

-触控

1.3 产品性能表

	I/O	PWM Channels	I2C	SPI	UART	ADC ext. Channels	Touch	Ext. INT	Package Types
SN8F5713SG/TG/JG	21	8	V	-	V	13	15	3	SOP24,TSSOP24, QFN24
SN8F5712	17	8	V	-	V	11	12	2	DIP20,SOP20
SN8F5711	13	8	V	-	V	9	10	2	DIP16,SOP16
SN8F57131	11	7	V	-	V	7	8	2	DIP14,SOP14
SN8F57112	5	3	V	-	V	3	3	-	SOP8
SN8F57113	6	4	V	-	V	4	-	-	SOP8

1.4 框图



2 目录

3 修改记录

版本	时间	说明
1.0	Dec. 2017	初版。
1.1	Apr.2018	修改 UART 和 PWM 寄存器。
1.2	May.2018	1. 增加 ILRC 使能时 STOP 模式电流。 2. 将 CPU 频率从 32 MHz 修改为 8 MHz。 3. 修改 UART 波特率表。
1.3	Nov.2018	1. 修改错误，遗漏，等。 2. MP5 烧录器烧录引脚图增加 NORMAL 模式与高速模式。 3. 移除 SKDIP 封装类型。 4. 修改引脚电路图章节。 5. 修改 Start-kit 章节。 6. 增加 P25 引脚信息。
1.4	Apr.2019	1. 修改错误，遗漏，等。 2. 修改 ADC 输入偏移范围。 3. 修改上电顺序和系统时钟。 4. 修改封装信息章节。 5. 修改 T0/T1 章节。 6. 移除 ADT 寄存器和偏移误差校正描述。 7. 增加 SN8F57112SG 和 SN8F57113SG 封装。 8. 移除 Noise Detect 选项。
1.5	Aug.2019	1. 移除 ADC VDD/VSS 通道； 2. 修改产品性能表和设备命名。
1.6	Apr.2020	1. 修改 IO/PSW 寄存器描述； 2. 修改引脚配置说明； 3. 修改看门狗定时器说明； 4. 修改唤醒时间； 5. 移除 I2C 6MHz 频率； 6. 移除 MP5 高速模式； 7. 修改电气特性章节； 8. 修改 ROM 烧录引脚部分。
1.7	Feb.2021	1. 移除 IEN1 寄存器。
1.8	Feb.2023	1. 更新封装信息。
1.9	April.2024	1. 增加未封出引脚设置注意事项； 2. 增加 ISP 注意事项； 3. 修改电气特性章节； 4. 修改内存说明； 5. 修改书写错误。
2.0	Jun.2024	1. 修改订购信息； 2. 移除示例代码。
2.1	Feb.2025	1. 修改 PWM 占空比； 2. 修改 GPIO 说明。

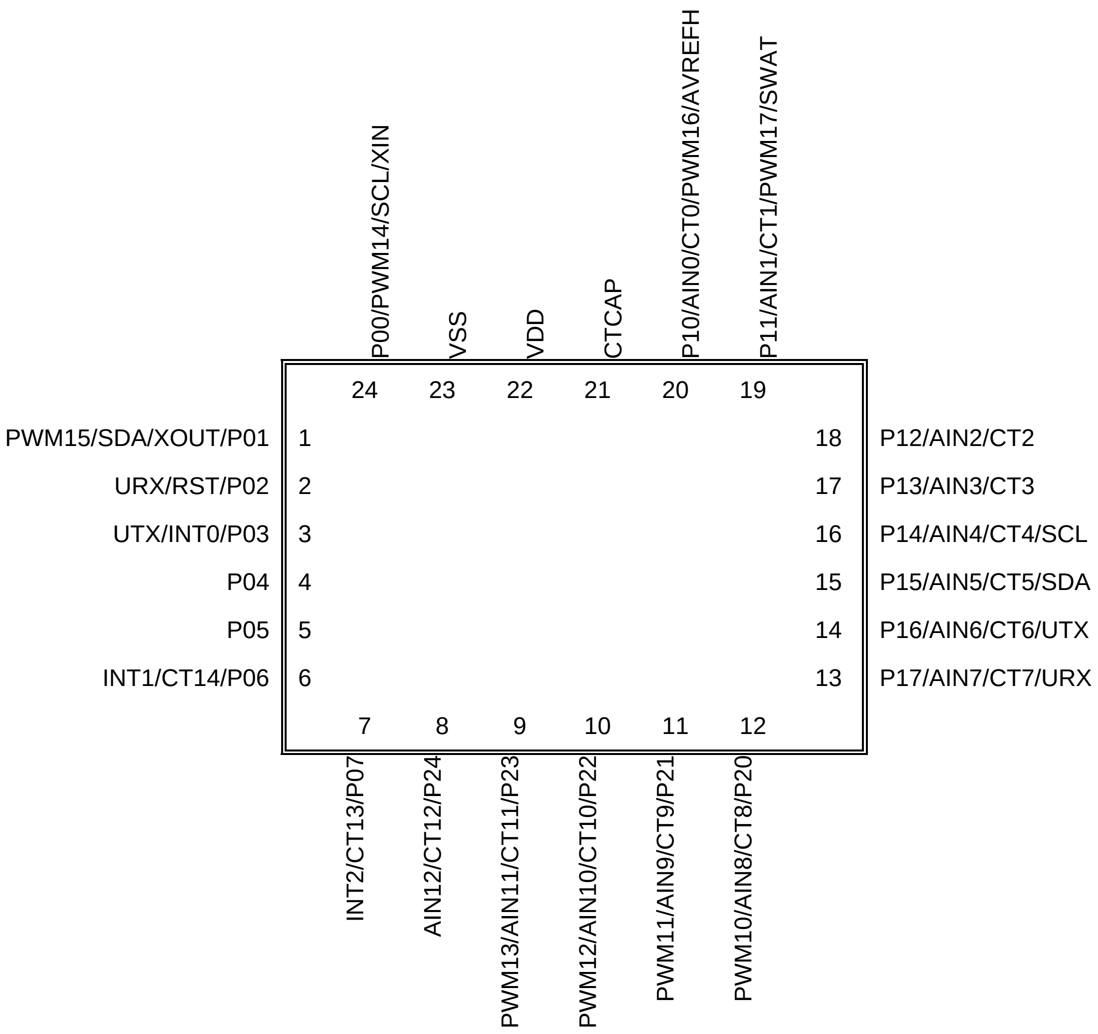
4 引脚配置

4.1 引脚图

- **SN8F5713SG/TG (SOP24/TSSOP24)**

VSS	1	U	24	VDD
PWM14/SCL/XIN/P00	2		23	CTCAP
PWM15/SDA/XOUT/P01	3		22	P10/AIN0/CT0/PWM16/AVREFH
URX/RST/P02	4		21	P11/AIN1/CT1/PWM17/SWAT
UTX/INT0/P03	5		20	P12/AIN2/CT2
P04	6		19	P13/AIN3/CT3
P05	7		18	P14/AIN4/CT4/SCL(1)
INT1/CT14/P06	8		17	P15/AIN5/CT5/SDA(1)
INT2/CT13/P07	9		16	P16/AIN6/CT6/UTX(1)
AIN12/CT12/P24	10		15	P17/AIN7/CT7/URX(1)
PWM13/AIN11/CT11/P23	11		14	P20/AIN8/CT8/PWM10
PWM12/AIN10/CT10/P22	12		13	P21/AIN9/CT9/PWM11

- **SN8F5713JG(QFN24)**



● SN8F5712PG/SG (DIP20/SOP20)

VSS	1	U	20	VDD
PWM14/SCL/XIN/P00	2		19	CTCAP
PWM15/SDA/XOUT/P01	3		18	P10/AIN0/CT0/PWM16/AVREFH
URX/RST/P02	4		17	P11/AIN1/CT1/PWM17/SWAT
UTX/INT0/P03	5		16	P14/AIN4/CT4/SCL(1)
P05	6		15	P15/AIN5/CT5/SDA(1)
INT1/CT14/P06	7		14	P16/AIN6/CT6/UTX(1)
AIN12/CT12/P24	8		13	P17/AIN7/CT7/URX(1)
PWM13/AIN11/CT11/P23	9		12	P20/AIN8/CT8/PWM10
PWM12/AIN10/CT10/P22	10		11	P21/AIN9/CT9/PWM11

*** 注：低功耗模式下，用户需要正确设置MCU中未封出的引脚，可以降低功耗。强烈的建议将这些引脚设置为输入上拉。***

● SN8F5711PG/SG (DIP16/SOP16)

VSS	1	U	16	VDD
PWM14/SCL/XIN/P00	2		15	CTCAP
PWM15/SDA/XOUT/P01	3		14	P10/AIN0/CT0/PWM16/AVREFH
UTX/INT0/P03	4		13	P11/AIN1/CT1/PWM17/SWAT
INT1/CT14/P06	5		12	P16/AIN6/CT6/UTX(1)
AIN12/CT12/P24	6		11	P17/AIN7/CT7/URX(1)
PWM13/AIN11/CT11/P23	7		10	P20/AIN8/CT8/PWM10
PWM12/AIN10/CT10/P22	8		9	P21/AIN9/CT9/PWM11

*** 注：低功耗模式下，用户需要正确设置MCU中未封出的引脚，可以降低功耗。强烈的建议将这些引脚设置为输入上拉。***

● SN8F57131PG/S G(DIP14/SOP14)

VSS	1	U	14	VDD
PWM14/SCL/XIN/P00	2		13	CTCAP
PWM15/SDA/XOUT/P01	3		12	P11/AIN1/CT1/PWM17/SWAT
UTX/INT0/P03	4		11	P16/AIN6/CT6/UTX(1)
INT1/CT14/P06	5		10	P17/AIN7/CT7/URX(1)
PWM13/AIN11/CT11/P23	6		9	P20/AIN8/CT8/PWM10
PWM12/AIN10/CT10/P22	7		8	P21/AIN9/CT9/PWM11

*** 注：低功耗模式下，用户需要正确设置MCU中未封出的引脚，可以降低功耗。强烈的建议将这些引脚设置为输入上拉。***

SN8F57112SG (SOP8)

VSS	1	U	8	VDD
PWM15/SDA/XOUT/P01	2		7	CTCAP
PWM14/SCL/XIN/P00	3		6	P11/AIN1/CT1/PWM17/SWAT
UTX(1) / CT6/AIN6/P16	4		5	P17/AIN7/CT7/URX(1)

**** 注：低功耗模式下，用户需要正确设置MCU中未封出的引脚，可以降低功耗。强烈的建议将这些引脚设置为输入上拉。***

SN8F57113SG (SOP8)

VSS	1	U	8	VDD
PWM15/SDA/XOUT/P01	2		7	P10/AIN0/CT0/PWM16/AVREFH
PWM14/SCL/XIN/P00	3		6	P11/AIN1/CT1/PWM17/SWAT
UTX(1) / CT6/AIN6/P16	4		5	P17/AIN7/CT7/URX(1)

**** 注：低功耗模式下，用户需要正确设置MCU中未封出的引脚，可以降低功耗。强烈的建议将这些引脚设置为输入上拉。***

注 1：I2C 输出可选 SCL/SDA or /SCL(1) /SDA(1)。 UART 输出可选 UTX/URX or /UTX(1) /URX(1)

4.2 引脚特性

SN8F5713S/T/J(SOP24/TSSOP24/QFN24)

Port	开漏功能	1/2*VDD偏压	灌电流100mA VSS+1.5V	外部中断	唤醒功能(电平变换)	ADC	电容触摸	UART	I2C	PWM	外部时钟	外部复位	调试接口
P0.0	-	V	V	-	V				SCL	PWM14	XIN		
P0.1	-	V	V	-	V				SDA	PWM15	XOUT		
P0.2	V	V	V	-	V			URX				RST	
P0.3	V	V	V	INT0	V			UTX					
P0.4	-	-	-	-	V								
P0.5	-	-	-	-	V								
P0.6	-	-	-	INT1	V		CT14						
P0.7	-	-	-	INT2	V		CT13						
P1.0	-	V	V	-	V	AVREFH/AIN0	CT0			PWM16			
P1.1	-	V	V	-	V	AIN1	CT1			PWM17			SWAT
P1.2	-	V	V	-	V	AIN2	CT2						
P1.3	-	V	V	-	V	AIN3	CT3						
P1.4	-	-	-	-	V	AIN4	CT4		SCL[1]				
P1.5	-	-	-	-	V	AIN5	CT5		SDA[1]				
P1.6	V	-	-	-	V	AIN6	CT6	UTX[1]					
P1.7	V	-	-	-	V	AIN7	CT7	URX[1]					
P2.0	-	-	-	-	-	AIN8	CT8			PWM10			
P2.1	-	-	-	-	-	AIN9	CT9			PWM11			
P2.2	-	-	-	-	-	AIN10	CT10			PWM12			
P2.3	-	-	-	-	-	AIN11	CT11			PWM13			
P2.4	-	-	-	-	-	AIN12	CT12						
CTCAP	-	-	-	-	-		CTCAP						

注 1：I2C 输出可选 SCL/SDA or /SCL(1) /SDA(1)。 UART 输出可选 UTX/URX or /UTX(1) /URX(1)

SN8F5712P/S(DIP20 /SOP20)

Port	开漏功能	1/2*VDD偏压	灌电流100mA VSS+1.5V	外部中断	唤醒功能(电平变换)	ADC	电容触摸	UART	I2C	PWM	外部时钟	外部复位	调试接口
P0.0	-	V	V	-	V				SCL	PWM14	XIN		
P0.1	-	V	V	-	V				SDA	PWM15	XOUT		
P0.2	V	V	V	-	V			URX				RST	
P0.3	V	V	V	INT0	V			UTX					
P0.5	-	-	-	-	V								
P0.6	-	-	-	INT1	V		CT14						
P1.0	-	V	V	-	V	AVREFH/AIN0	CT0			PWM16			
P1.1	-	V	V	-	V	AIN1	CT1			PWM17			SWAT
P1.4	-	-	-	-	V	AIN4	CT4		SCL(1)				
P1.5	-	-	-	-	V	AIN5	CT5		SDA(1)				
P1.6	V	-	-	-	V	AIN6	CT6	UTX(1)					
P1.7	V	-	-	-	V	AIN7	CT7	URX(1)					
P2.0	-	-	-	-	-	AIN8	CT8			PWM10			
P2.1	-	-	-	-	-	AIN9	CT9			PWM11			
P2.2	-	-	-	-	-	AIN10	CT10			PWM12			
P2.3	-	-	-	-	-	AIN11	CT11			PWM13			
P2.4	-	-	-	-	-	AIN12	CT12						
CTCAP	-	-	-	-	-		CTCAP						

注 1：I2C 输出可选 SCL/SDA or /SCL(1) /SDA(1)。 UART 输出可选 UTX/URX or /UTX(1) /URX(1)

SN8F5711P/S(DIP16 /SOP16)

Port	开漏功能	1/2*VDD偏压	灌电流100mA VSS+1.5V	外部中断	唤醒功能(电平变换)	ADC	电容触摸	UART	I2C	PWM	外部时钟	外部复位	调试接口
P0.0	-	V	V	-	V				SCL	PWM14	XIN		
P0.1	-	V	V	-	V				SDA	PWM15	XOUT		
P0.3	V	V	V	INT0	V			UTX					
P0.6	-	-	-	INT1	V		CT14						
P1.0	-	V	V	-	V	AVREFH/ AIN0	CT0			PWM16			
P1.1	-	V	V	-	V	AIN1	CT1			PWM17			SWAT
P1.6	V	-	-	-	V	AIN6	CT6	UTX(1)					
P1.7	V	-	-	-	V	AIN7	CT7	URX(1)					
P2.0	-	-	-	-	-	AIN8	CT8			PWM10			
P2.1	-	-	-	-	-	AIN9	CT9			PWM11			
P2.2	-	-	-	-	-	AIN10	CT10			PWM12			
P2.3	-	-	-	-	-	AIN11	CT11			PWM13			
P2.4	-	-	-	-	-	AIN12	CT12						
CTCAP	-	-	-	-	-		CTCAP						

注 1：I2C 输出可选 SCL/SDA or /SCL(1) /SDA(1)。 UART 输出可选 UTX/URX or /UTX(1) /URX(1)

SN8F57131P/S(DIP14 /SOP14)

Port	开漏功能	1/2*VDD偏压	灌电流100mA VSS+1.5V	外部中断	唤醒功能(电平变换)	ADC	电容触摸	UART	I2C	PWM	外部时钟	外部复位	调试接口
P0.0	-	V	V	-	V				SCL	PWM14	XIN		
P0.1	-	V	V	-	V				SDA	PWM15	XOUT		
P0.3	V	V	V	INT0	V			UTX					
P0.6	-	-	-	INT1	V		CT14						
P1.1	-	V	V	-	V	AIN1	CT1			PWM17			SWAT
P1.6	V	-	-	-	V	AIN6	CT6	UTX[1]					
P1.7	V	-	-	-	V	AIN7	CT7	URX[1]					
P2.0	-	-	-	-	-	AIN8	CT8			PWM10			
P2.1	-	-	-	-	-	AIN9	CT9			PWM11			
P2.2	-	-	-	-	-	AIN10	CT10			PWM12			
P2.3	-	-	-	-	-	AIN11	CT11			PWM13			
CTCAP	-	-	-	-	-		CTCAP						

注 1：I2C 输出可选 SCL/SDA or /SCL[1] /SDA[1]。 UART 输出可选 UTX/URX or /UTX[1] /URX[1]

SN8F57112S(SOP8)

Port	开漏功能	1/2*VDD偏压	灌电流100mA VSS+1.5V	外部中断	唤醒功能(电平变换)	ADC	电容触摸	UART	I2C	PWM	外部时钟	外部复位	调试接口
P0.0	-	V	V	-	V				SCL	PWM14	XIN		
P0.1	-	V	V	-	V				SDA	PWM15	XOUT		
P1.1	-	V	V	-	V	AIN1	CT1			PWM17			SWAT
P1.6	V	-	-	-	V	AIN6	CT6	UTX(1)					
P1.7	V	-	-	-	V	AIN7	CT7	URX(1)					
CTCAP	-	-	-	-	-		CTCAP						

注 1：I2C 输出可选 SCL/SDA or /SCL(1) /SDA(1)。 UART 输出可选 UTX/URX or /UTX(1) /URX(1)

SN8F57113P/S(SOP8)

Port	开漏功能	1/2*VDD偏压	灌电流100mA VSS+1.5V	外部中断	唤醒功能(电平变换)	ADC	电容触摸	UART	I2C	PWM	外部时钟	外部复位	调试接口
P0.0	-	V	V	-	V				SCL	PWM14	XIN		
P0.1	-	V	V	-	V				SDA	PWM15	XOUT		
P1.0	-	V	V	-	V	AVREFH/AIN0	CT0			PWM16			
P1.1	-	V	V	-	V	AIN1	CT1			PWM17			SWAT
P1.6	V	-	-	-	V	AIN6	CT6	UTX(1)					
P1.7	V	-	-	-	V	AIN7	CT7	URX(1)					

注 1：I2C 输出可选 SCL/SDA or /SCL(1) /SDA(1)。 UART 输出可选 UTX/URX or /UTX(1) /URX(1)

4.3 引脚说明

电源引脚

引脚名称	类型	功能说明
VDD	电源	电源(1.8~5.5V)
VSS	电源	接地(0V)

P0

引脚名称	类型	功能说明
P0.0 XIN SCL PWM14	Digital I/O Analog Input Digital I/O Digital Output	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 系统时钟：外部时钟输入。 I2C：时钟输出（主控）时钟输入（从动）。 PWM：可编程 PWM 输出。
P0.1 XOUT SDA PWM15	Digital I/O Analog Output Digital I/O Digital Output	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 系统时钟：外部晶振/振荡器输出。 I2C：数据引脚。 PWM：可编程 PWM 输出。
P0.2 RST URX	Digital I/O Digital Input Digital Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 系统复位（低电平有效）。 UART：数据输入引脚。
P0.3 UTX INT0	Digital I/O Digital Output Digital Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 UART：数据输出引脚。 INT0：外部中断 0 。
P0.4	Digital I/O	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。
P0.5	Digital I/O	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。
P0.6 INT1 CT14	Digital I/O Digital Input Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 INT1：外部中断 1。 CT14：电容感应触摸输入通道 14。
P0.7 INT2 CT13	Digital I/O Digital Input Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 INT2：外部中断 2。 CT13：电容感应触摸输入通道 13。

P1

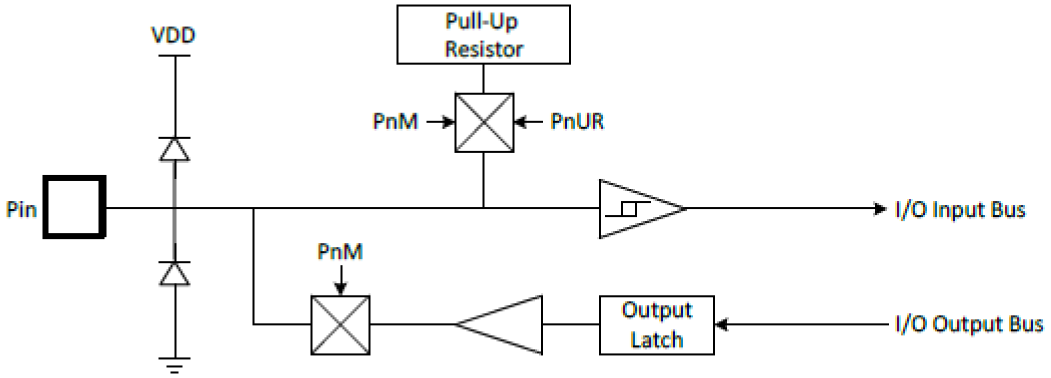
引脚名称	类型	功能说明
P1.0 AIN0 AVREFH PWM16 CT0	Digital I/O Analog Input Analog Input Digital Output Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 ADC：输入通道 0。 ADC：外部参考电压。 PWM：可编程 PWM 输出。 CT0：电容感应触摸输入通道 0。
P1.1 AIN1 SWAT PWM17 CT1	Digital I/O Analog Input Digital I/O Digital Output Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 ADC：输入通道 1。 单线仿真口。 PWM：可编程 PWM 输出。 CT1：电容感应触摸输入通道 1。
P1.2 AIN2 CT2	Digital I/O Analog Input Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 ADC：输入通道 2。 CT2：电容感应触摸输入通道 2。
P1.3 AIN3 CT3	Digital I/O Analog Input Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 ADC：输入通道 3。 CT3：电容感应触摸输入通道 3。
P1.4 AIN4 SCL(1) CT4	Digital I/O Analog Input Digital I/O Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 ADC：输入通道 4。 I2C：时钟输出（主控）时钟输入（从动）。 CT4：电容感应触摸输入通道 4。
P1.5 AIN5 SDA(1) CT5	Digital I/O Analog Input Digital I/O Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 ADC：输入通道 5。 I2C：数据引脚。 CT5：电容感应触摸输入通道 5。
P1.6 AIN6 UTX(1) CT6	Digital I/O Analog Input Digital Output Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 ADC：输入通道 6。 UART：传输引脚。 CT6：电容感应触摸输入通道 6。
P1.7 AIN7 URX(1) CT7	Digital I/O Analog Input Digital Input Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻，电平变换时具有唤醒功能。 ADC：输入通道 7。 UART：接收引脚。 CT7：电容感应触摸输入通道 7。

P2

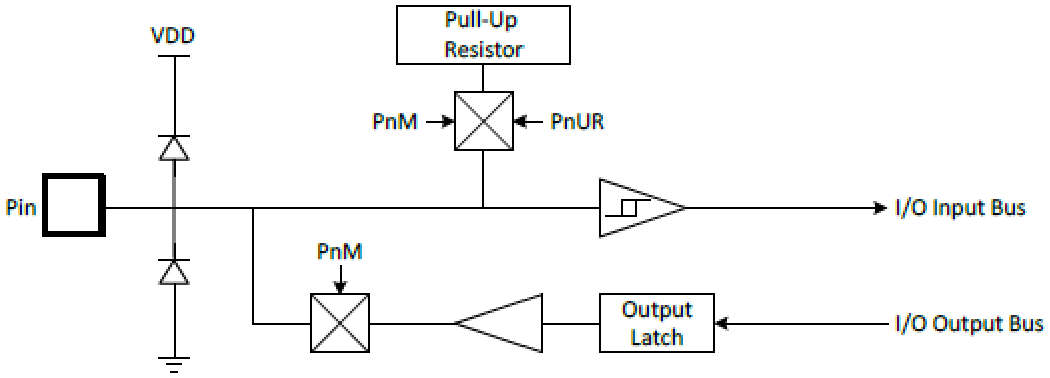
引脚名称	类型	功能说明
P2.0 AIN8 PWM10 CT8	Digital I/O Analog Input Digital Output Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻。 ADC：输入通道 8。 PWM：可编程 PWM 输出。 CT8：电容感应触摸输入通道 8。
P2.1 AIN9 PWM11 CT9	Digital I/O Analog Input Digital Output Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻。 ADC：输入通道 9。 PWM：可编程 PWM 输出。 CT9：电容感应触摸输入通道 9。
P2.2 AIN10 PWM12 CT10	Digital I/O Analog Input Digital Output Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻。 ADC：输入通道 10。 PWM：可编程 PWM 输出。 CT10：电容感应触摸输入通道 10。
P2.3 AIN11 PWM13 CT11	Digital I/O Analog Input Digital Output Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻。 ADC：输入通道 11。 PWM：可编程 PWM 输出。 CT11：电容感应触摸输入通道 11。
P2.4 AIN12 CT12	Digital I/O Analog Input Analog Input	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻。 ADC：输入通道 12。 CT12：电容感应触摸输入通道 12。
P2.5	Digital I/O	GPIO：双向输入输出引脚，输入模式时施密特触发，内置上拉电阻。

4.4 引脚电路图

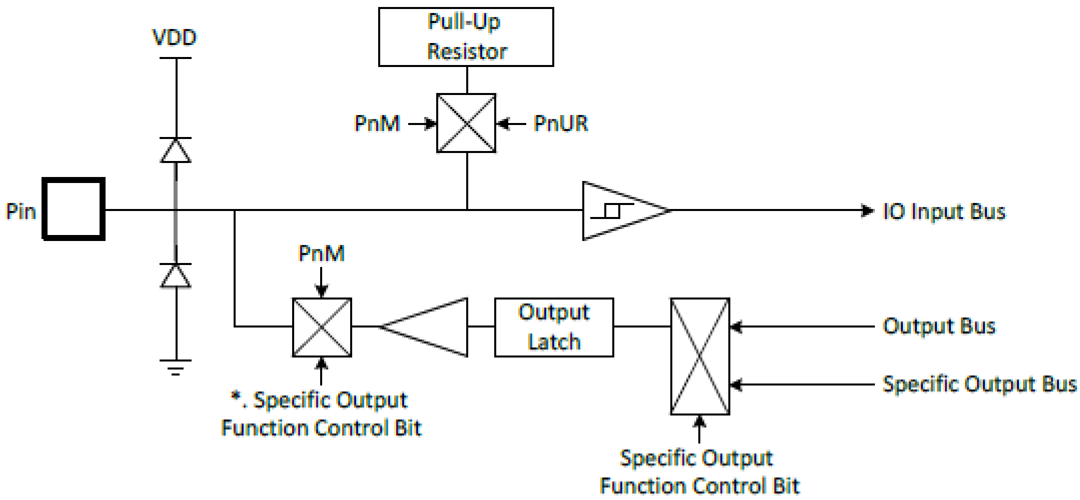
GPIO



双向 I/O，与特殊数字输入功能引脚共用，如 INT2。

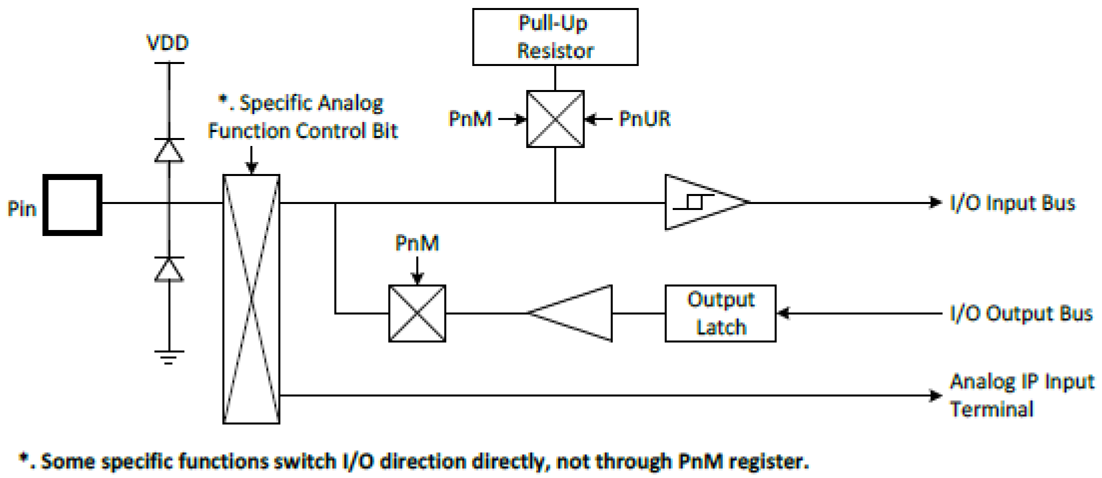


双向 I/O，与特殊数字输出功能引脚共用，如 PWM, UART。

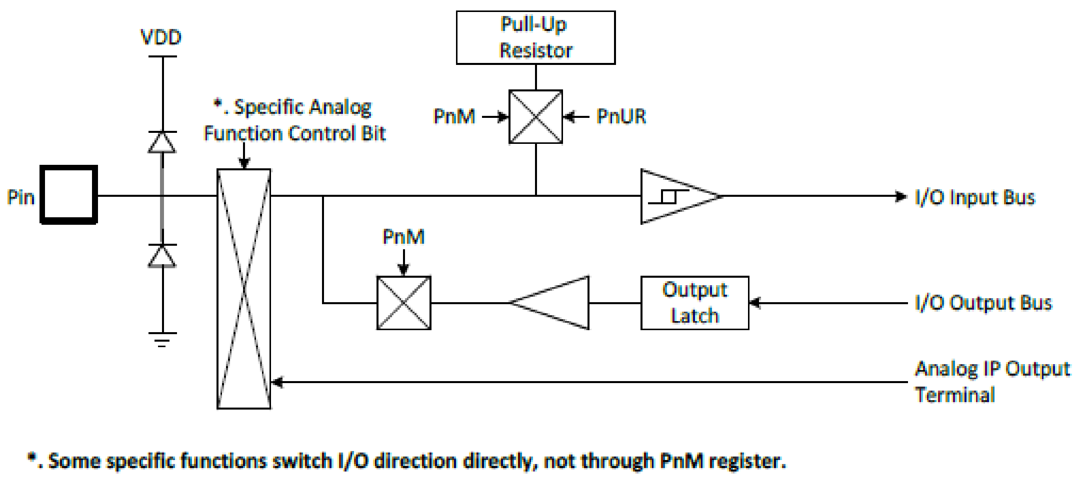


***. Some specific functions switch I/O direction directly, not through PnM register.**

双向 I/O，与特殊模拟输入功能引脚共用，如 XIN, ADC。



双向 I/O，与特殊模拟输出功能引脚共用，如 XOUT…

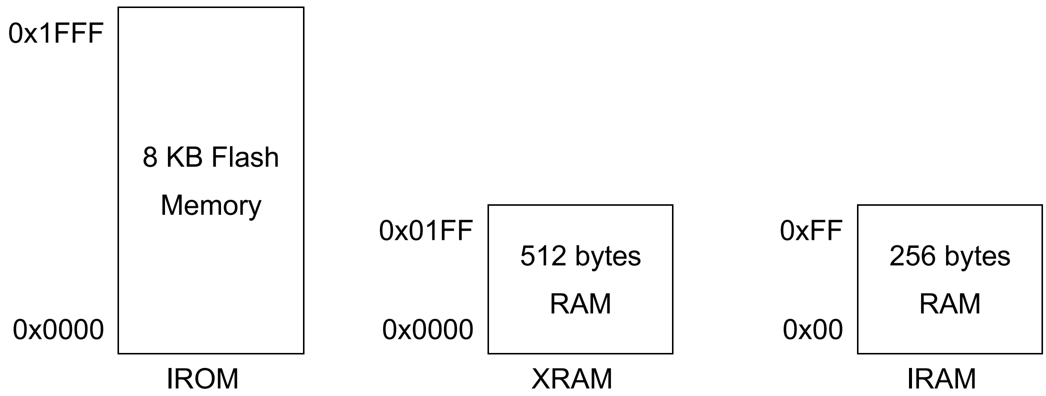


5 CPU

SN8F5000 系列是一颗增强型的 8051 微控制器，且完全兼容 MCS-51 指令集，因此可以使用目前流行的编译仿真环境（例如 Keil C51）。在相同的频率下，SN8F5000 CPU 比原来的 8051 快 9.4 到 12.1 倍。

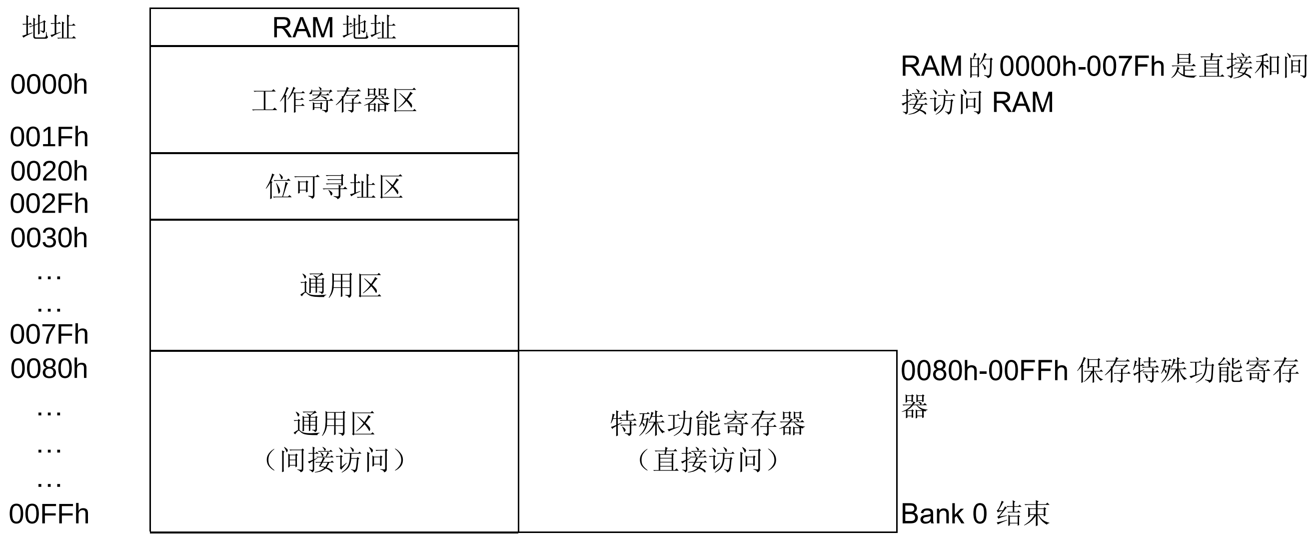
5.1 存储器结构

SN8F5713 内建了三个存储器：内部 RAM(IRAM)，外部 RAM(XRAM)，以及程序存储器(IROM)。内部 RAM 由 256 个字节组成，具有较高的存取性能（支持直接寻址和间接寻址）。相比之下，外部 RAM 有 512 字节大小，但需要更长的存取周期。程序存储器是一个 8KB 的 Flash 存储器，最快的存取速度可达 8MHz。



5.2 内部 RAM (IRAM)

256 X 8 位 RAM （内部数据存储器）



内部数据存储器里的256字节数据RAM是一个标准的8051 RAM访问配置。以下为存储器空间分配：

- 0x00-0x7F：低 128 字节的内部 RAM 可直接寻址或间接寻址；
- 0x80-0xFF：高 128 字节的内部 RAM 只能间接寻址；

高 128 字节内部 RAM 占用的地址空间与特殊功能寄存器(SFR)相同，但在物理上与 SFR 的空间是分离的。当一个指令访问地址高于 0x7F 时，CPU 可以根据指令的寻址方式访问高 128 字节的数据 RAM 还是访问 SFR：

- 0x80-0xFF：特殊功能寄存器(SFR)只能直接寻址；

低 128 字节包含工作寄存器区域与位可寻址区域：

- 0x20-0x2F：位可寻址区。

在位可寻址区，用户可以使用任意位的唯一地址对此范围内的位进行读写。支持 16 字节位可寻址 RAM 区有 128 个可寻址位。每个位都有单独的地址，范围从 00H~7FH。因此，可以对该位进行直接寻址。字节 20H 的 Bit0 为位地址 00H，字节 20H 的 Bit7 为位地址 07H。字节 2FH 的 Bit0 为位地址 78H，字节 2FH 的 Bit7 为位地址 7FH。“SETB 42 h”置为 1 时，即说明字节 28 H 的 Bit2 置为 1。

	字节地址	Bite 0	Bite 1	Bite 2	Bite 3	Bite 4	Bite 5	Bite 6	Bite 7
位可寻址区	0x20	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
	0x21	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
	0x22	0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
	0x23	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
	0x24	0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
	0x25	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
	0x26	0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37
	0x27	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
	0x28	0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47
	0x29	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F
	0x2A	0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57
	0x2B	0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F
	0x2C	0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67
	0x2D	0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F
	0x2E	0x70	0x71	0x72	0x73	0x74	0x75	0x76	0x77
	0x2F	0x78	0x79	0x7A	0x7B	0x7C	0x7D	0x7E	0x7F

5.3 外部 RAM (XRAM)

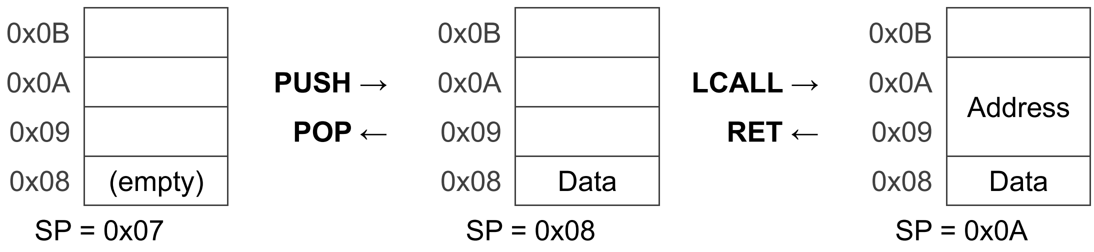
512 X 8 位 XRAM （扩展数据存储器）

外部 RAM 增加了变量的容量，但跟内部 RAM 相比，它的访问能力是最低的，这是因为频繁地使用变量和本地变量是为了存入内部 RAM，而外部 RAM 的绝大多数用法都是特定的。外部 RAM 可以作为优先级别较低的，或者在 ROM 中预装查表以加速访问周期的变量存储区域。

地址	RAM 地址	
0000h		RAM Bank 0 扩展
…	通用区	
01FFh		Bank 0 结束

5.4 堆栈

可从内部 RAM（IRAM）中分出任意一部分作为堆栈使用，但要求手动分配以保证堆栈区域不会与其它 RAM 的变量重叠。堆栈溢出或者下溢会导致其它 RAM 的变量重写错误，故在分配堆栈的区域时必须考虑到这些问题以避免这种情形的发生。



默认情况下，堆栈指针（SP 寄存器）指向 07H，就是指堆栈的区域从 IRAM 地址中的 08H 开始。换句话说，如果计划想将堆栈区域设置为从 IRAM 的 0C0H 开始，就要在系统复位后将 SP 寄存器设置为 0BFH。

一条汇编 PUSH 指令占用堆栈中的一个字节，LCALL，ACALL 指令以及中断分别占用堆栈中的两个字节。POP 指令释放一个字节，RET/RETI 指令释放两个字节。

5.5 程序存储器 (IROM)

程序存储器是 Flash 存储器，可以存储程序代码，ROM 查表数据以及其它的临时修改数据。可通过调试工具如 SN-Link3 进行升级，也可以通过在线程序处理来自行更新。

地址	ROM	注释
0000H	复位向量	复位向量
0001H 0002H	通用区	用户程序
0003H	**INT0 中断向量**	**中断向量**
000BH	**PWM1 中断向量**	
0013H	**UART TX 中断向量**	
001BH	**ADC 中断向量**	
0043H	**INT2 中断向量**	
0053H	**UART RX 中断向量**	
0063H	**TIMER0 中断向量**	
0083H	**INT1 中断向量**	
0093H	**I2C 中断向量**	
00A3H	**TIMER1 中断向量**	
00ECH	通用区	用户程序 用户程序结束
1FF6H 1FF7H . 1FFDH 1FFEH 1FFFH	保留	

ROM 包括复位向量、中断向量、通用区和保留区。复位向量是程序开始地址。中断向量的任何中断发生时的中断服务程序头。通用区是主程序区，包括主循环、子程序和数据表。

- 0x0000 复位向量：发生复位事件后(上电复位、复位引脚复位、看门狗复位、LVD 复位…)，程序计数器指向 0x0000。
- 0x0001 ~ 0 x0002：通用区处理系统复位操作。
- 0x0003 ~ 0 x00EB：多中断向量区。每个中断事件都有一个独特的中断向量。
- 0x00EC ~ 0 x1FE0：用户程序和 ISP(EEPROM 功能)的通用区。
- **0x1FE0 ~ 0 x1FF6：用户程序通用区。不执行 ISP。**
- **0x1FF6 ~ 0 x1FFF：保留区域。不执行 ISP。**

ROM 安全规则保护所有地址 ROM 数据，且输出 0x0000。

5.6 程序存储器安全

SN8F5713 内置 ROM 加密机制，防止 Flash ROM 资料被破解。当使能加密功能时，就无法读出 ROM 里面的内容，所有的 ROM 地址都只能读到 0x00 的数据。

5.7 数据指针

在执行 MOVX 和 MOVC 指令时，数据指针可帮助指定 XRAM 和 IROM 地址。该单片机有 1 组数据指针（DPH/DPL）。DPC 寄存器控制自动加减 DPTR 功能。

自动加减 DPTR 功能是在执行 MOVX@DPTR 指令后使指针指向的地址自动加 1 或减 1。因此，它能够连续的访问外部存储器，而不需要重复的去指定 DPTR 值。

5.8 堆栈和数据指针寄存器

寄存器	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SP	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
DPL	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0
DPH	DPH7	DPH6	DPH5	DPH4	DPH3	DPH2	DPH1	DPH0
DPC	-	-	-	-	-	ATMS	ATMD	ATME

SP 寄存器（0x81）

Bit	Field	Type	Initial	说明
7..0	SP	R/W	0x07	堆栈指针。

DPL 寄存器（0x82）

Bit	Field	Type	Initial	说明
7..0	DPL[7:0]	R/W	0x00	DPTR0 的低字节。

DPH 寄存器（0x83）

Bit	Field	Type	Initial	说明
7..0	DPH[7:0]	R/W	0x00	DPTR0 的高字节。

DPC 寄存器 (0x93)

Bit	Field	Type	Initial	说明
7..3	Reserved	R	0x0	
2..1	ATMS/ATMD	R/W	00	自动加减 DPTR（使能 ATME 位时有效）。 00：执行 MOVX @DPTR 指令后+1； 01：执行 MOVX @DPTR 指令后-1； 10：执行 MOVX @DPTR 指令后+2； 11：执行 MOVX @DPTR 指令后-2。
0	ATME	R/W	0	自动加减 DPTR 功能。 0：禁止； 1：使能。

6 特殊功能寄存器

6.1 特殊功能寄存器存储器

BIN / HEX	000	001	010	011	100	101	110	111
F8	-	P0M	P1M	P2M	FRQL	FRQH	FRQCMD	PFLAG
F0	B	P0UR	P1UR	P2UR	P0BIAS	P1BIAS	-	SRST
E8	-	MD0	MD1	MD2	MD3	MD4	MD5	ARCON
E0	ACC				P0OC	CLKSEL	CLKCMD	TCON0
D8	S0CON2	-	I2CDAT	I2CADR	I2CCON	I2CSTA	SMBSEL	SMBDST
D0	PSW	-	ADM	ADB	ADR	VREFH	P2CON	-
C8	-	PW16DL	PW16DH	PW17DL	PW17DH	-	SYSMOD	-
C0	CSCON	CSCON1	CSCON2	SGMOD1	SGMOD2	-	-	-
B8	-	IP1	S0RELH	PW14DL	PW14DH	PW15DL	PW15DH	IRCON2
B0	-	CSCH	CSCL	CCAL	-	-	PWCH	-
A8	IEN0	IP0	S0RELL	PW12DL	PW12DH	PW13DL	PW13DH	
A0	P2	PW1M	PW1YL	PW1YH	PW10DL	PW10DH	PW11DL	PW11DH
98	S0CON	S0BUF	IEN2	-	-	-	P0CON	P1CON
90	P1	P1W	-	DPC	PECMD	PEROML	PEROMH	PERAM
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PEDGE
80	P0	SP	DPL	DPH	-	-	WDTR	PCON

> ✸ *注：最左栏里的所有* ***SFR*** *都是位可寻址的。（每个以* ***0x0/0x8*** *结束的* ***SFR*** *地址都是位可寻址的）。*

6.2 特殊功能寄存器说明

0x80 - 0x9F 寄存器说明

寄存器	地址	说明
P0	0x80	P0 数据缓存器。
SP	0x81	堆栈指针寄存器。
DPL	0x82	数据指针 0 低字节寄存器。
DPH	0x83	数据指针 0 高字节寄存器。
-	0x84	-
-	0x85	-
WDTR	0x86	看门狗定时器清零寄存器。
PCON	0x87	系统模式寄存器。
TCON	0x88	T0/1 控制寄存器。
TMOD	0x89	T0/1 模式寄存器。
TL0	0x8A	T0 计数低字节寄存器。
TL1	0x8B	T1 计数低字节寄存器。
TH0	0x8C	T0 计数高字节寄存器。
TH1	0x8D	T1 计数高字节寄存器。
CKCON	0x8E	扩展周期寄存器。
PEDGE	0x8F	外部中断边沿控制寄存器。
P1	0x90	P1 数据缓存器。
P1W	0x91	P1 唤醒控制寄存器。
-	0x92	-
DPC	0x93	数据指针控制寄存器。
PECMD	0x94	In-System Program 命令寄存器。
PEROML	0x95	In-System Program ROM 地址低字节。
PEROMH	0x96	In-System Program ROM 地址高字节。
PERAM	0x97	In-System Program RAM 分配地址。
S0CON	0x98	UART 控制寄存器。
S0BUF	0x99	UART 数据缓存器。
IEN2	0x9A	中断使能寄存器。
-	0x9B	-
-	0x9C	-
-	0x9D	-
P0CON	0x9E	P0 配置控制寄存器。
P1CON	0x9F	P1 配置控制寄存器。

0xA0 - 0xBF 寄存器说明

寄存器	地址	说明
P2	0xA0	P2 数据缓存器。
PW1M	0xA1	PW1 控制寄存器。
PW1YL	0xA2	PW1 周期控制缓存器低字节。
PW1YH	0xA3	PW1 周期控制缓存器高字节。
PW10DL	0xA4	PW10 占空比控制缓存器低字节。
PW10DH	0xA5	PW10 占空比控制缓存器高字节。
PW11DL	0xA6	PW11 占空比控制缓存器低字节。
PW11DH	0xA7	PW11 占空比控制缓存器高字节。
IEN0	0xA8	中断使能寄存器。
IP0	0xA9	中断优先权寄存器。
S0RELL	0xAA	UART 重装低字节寄存器。
PW12DL	0xAB	PW12 占空比控制缓存器低字节。
PW12DH	0xAC	PW12 占空比控制缓存器高字节。
PW13DL	0xAD	PW13 占空比控制缓存器低字节。
PW13DH	0xAE	PW13 占空比控制缓存器高字节。
-	0xAF	-
-	0xB0	-
-	0xB1	-
-	0xB2	-
-	0xB3	-
-	0xB4	-
-	0xB5	-
PWCH	0xB6	PWM 通道控制缓存器
-	0xB7	-
-	0xB8	-
IP1	0xB9	中断优先权寄存器。
S0RELH	0xBA	UART 重装高字节寄存器。
PW14DL	0xBB	PW14 占空比控制缓存器低字节。
PW14DH	0xBC	PW14 占空比控制缓存器高字节。
PW15DL	0xBD	PW15 占空比控制缓存器低字节。
PW15DH	0xBE	PW15 占空比控制缓存器高字节。
IRCON2	0xBF	中断请求寄存器。

0xC0 - 0xDF 寄存器说明

寄存器	地址	说明
-	0xC0	-
-	0xC1	-
-	0xC2	-
-	0xC3	-
-	0xC4	-
-	0xC5	-
-	0xC6	-
-	0xC7	-
-	0xC8	-
PW16DL	0xC9	PW16 占空比控制缓存器低字节。
PW16DH	0xCA	PW16 占空比控制缓存器高字节。
PW17DL	0xCB	PW17 占空比控制缓存器低字节。
PW17DH	0xCC	PW17 占空比控制缓存器高字节。
-	0xCD	-
SYSMOD	0xCE	系统模式寄存器
-	0xCF	-
PSW	0xD0	系统标志寄存器。
-	0xD1	-
ADM	0xD2	ADC 控制寄存器。
ADB	0xD3	ADC 数据缓存器。
ADR	0xD4	ADC 分辨率选择寄存器。
VREFH	0xD5	ADC 参考电压控制寄存器。
P2CON	0xD6	Port 2 配置控制寄存器。
-	0xD7	-
S0CON2	0xD8	UART 波特率控制寄存器。
ADT	0xD9	ADC 偏移校准控制寄存器。
I2CDAT	0xDA	I2C 数据缓存器。
I2CADR	0xDB	I2C 从动地址。
I2CCON	0xDC	I2C 接口操作控制寄存器。
I2CSTA	0xDD	I2C 状态代码。
SMBSEL	0xDE	SMBUS 模式控制寄存器。
SMBDST	0xDF	SMBUS 内部超时寄存器。

0xE0 - 0xFF 寄存器说明

寄存器	地址	说明
ACC	0xE0	ACC 寄存器。
-	0xE1	-
-	0xE2	-
-	0xE3	-
P0OC	0xE4	开漏功能控制寄存器。
CLKSEL	0xE5	时钟切换选择寄存器。
CLKCMD	0xE6	时钟切换控制寄存器。
TCON0	0xE7	T0/1 时钟控制寄存器。
-	0xE8	-
MD0	0xE9	MDU 控制寄存器 0。
MD1	0xEA	MDU 控制寄存器 1。
MD2	0xEB	MDU 控制寄存器 2。
MD3	0xEC	MDU 控制寄存器 3。
MD4	0xED	MDU 控制寄存器 4。
MD5	0xEE	MDU 控制寄存器 5。
ARCON	0xEF	MDU 算法控制寄存器。
B	0xF0	乘除法指令数据缓存器。
P0UR	0xF1	Port 0 上拉电阻控制寄存器。
P1UR	0xF2	Port 1 上拉电阻控制寄存器。
P2UR	0xF3	Port 2 上拉电阻控制寄存器。
P0BIAS	0xF4	Port 0 1/2*VDD 偏压控制寄存器。
P1BIAS	0xF5	Port 1 1/2*VDD 偏压控制寄存器。
-	0xF6	-
SRST	0xF7	软件复位控制寄存器。
-	0xF8	-
P0M	0xF9	Port 0 输入/输出模式寄存器。
P1M	0xFA	Port 1 输入/输出模式寄存器。
P2M	0xFB	Port 2 输入/输出模式寄存器。
FRQL	0xFC	时钟微调控制缓存器低字节。
FRQH	0xFD	时钟微调控制缓存器高字节。
FRQCMD	0xFE	时钟微调控制命令寄存器。
PFLAG	0xFF	复位标志寄存器。

6.3 系统寄存器

寄存器	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ACC	ACC7	ACC6	ACC5	ACC4	ACC3	ACC2	ACC1	ACC0
B	B7	B6	B5	B4	B3	B2	B1	B0
PSW	CY	AC	F0	RS1	RS0	OV	F1	P

ACC Register (0xE0)

Bit	Field	Type	Initial	说明
7..0	ACC[7:0]	R/W	0x00	8 位数据寄存器用于转移或操控 ALU 和数据存储器之间的数据，若操作结果溢出（OV）或有进位/借位（CY 或 AC），以及奇校验（P），会在 PSW 寄存器中将对应的标志位置 1。

B Register (0xF0)

Bit	Field	Type	Initial	说明
7..0	B[7:0]	R/W	0x00	B 寄存器在使用乘法和除法指令时使用，而且还能作为高速暂存寄存器来保留临时数据。

PSW Register (0xD0)

Bit	Field	Type	Initial	说明
7	CY	R/W	0	进位标志。 0：加法运算后没有进位、减法运算有借位发生或移位后移出逻辑“0”或比较运算的结果<0； 1：加法运算后有进位、减法运算没有借位发生或移位后移出逻辑“1”或比较运算的结果≥0。
6	AC	R/W	0	辅助进位标志。 0：BCD 码加法运算中 bit3 没有向 bit4 产生进位； 1：BCD 码加法运算中 bit3 有向 bit4 产生进位。
5	F0	R/W	0	通用标志位 0，可任意设定。
4..3	RS[1:0]	R/W	00	寄存器 bank 选择控制位，用于选择工作寄存器 bank。 00：00H – 07H (Bnak0)； 01：08H – 0FH (Bnak1)； 10：10H – 17H (Bnak2)； 11：18H – 1FH (Bnak3)。
2	OV	R/W	0	溢出标志。 0：算术操作时，ACC 没有溢出； 1：算术操作时，ACC 溢出。
1	F1	R/W	0	通用标志位 1，可任意设定。
0	P	R	0	奇偶标志位。 0：ACC 中 1 的个数为偶数； 1：ACC 中 1 的个数为奇数

6.4 寄存器宣告

SN8F5713 通过不同的寄存器控制不同的功能，但在 C51/A51 编译器中没有预先定义 SFR 的名称。为编程更容易，就需要增加一个 header 文件来宣告 SFR 名称。

使用汇编语言进行编程时，增加下面的句子：

```
$NOMOD51 ;Do not recognize the 8051-specific predefined special register.
#include <SN8F5713.H>
```

使用 C 运用进行编程是，增加下面的句子：

```
#include <SN8F5713.H>
```

增加 header 文件后，用户可使用寄存器的名称进行编程。编译过程中，编译器通过 header 文件将寄存器的名称转换出寄存器的位置。

不同的设备需要使用不同的 header 文件进行宣告，但 option 文件是一样的。

Device	Header file	Options file
SN8F5713	SN8F5713.h	OPTIONS_SN8F5713.A51
SN8F5712	SN8F5712.h	
SN8F5711	SN8F5711.h	
SN8F57131	SN8F57131.h	
SN8F57112	SN8F57112.h	
SN8F57113	SN8F57113.h	

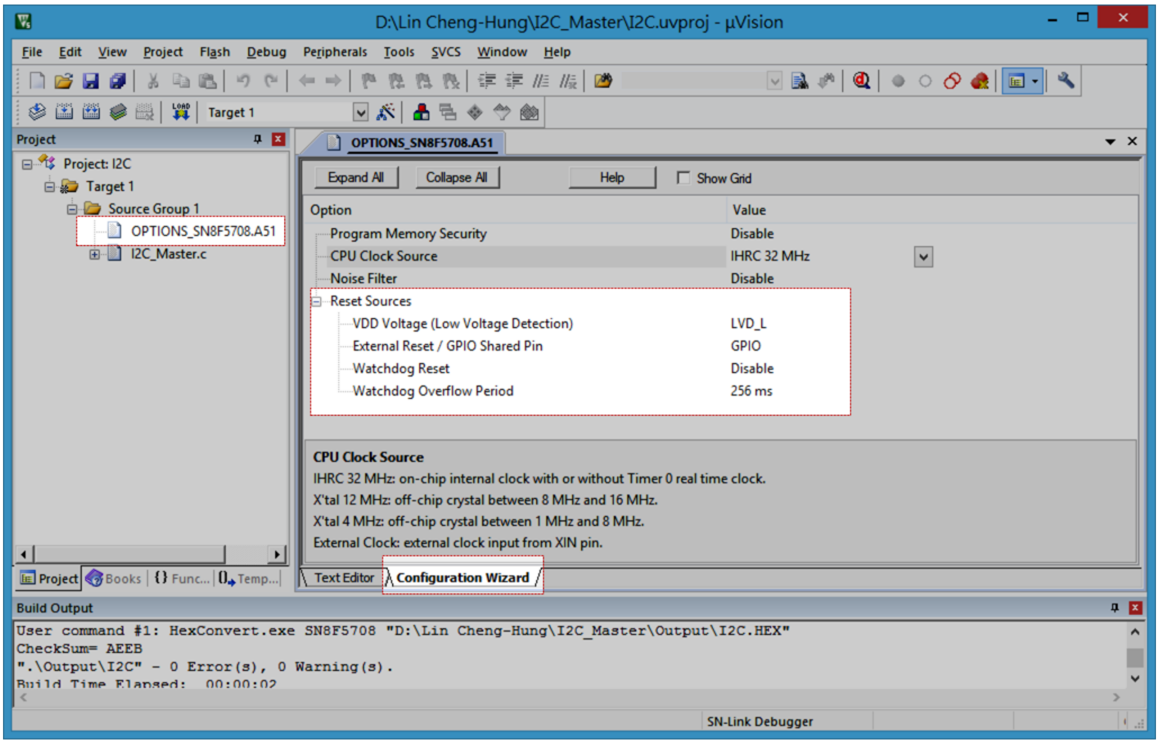
7 复位和上电控制

复位和上电控制有下列方式：低电压检测（LVD），看门狗，可编程的外部复位引脚和软件复位。前面三种方式可触发额外的上电流程，随后单片机初始化所有的寄存器，并是程序从复位向量（ROM 地址 0x0000）处重新开始执行。

7.1 复位和上电控制的配置

SONiX 在 SN-Link Driver for Keil C51.exe 中（从松翰官网 www.sonix.com.tw 下载）有发布 SN8F5713_OPTIONS.A51 文件，该文件包含复位源和 CPU 时钟源选择等合适的参数，强烈建议添加该文件到 Keil 工程中。SN8F5000 的调试工具手册提供了更详细的内容。具体如下所示：

- 程序内存加密
- 时钟源
- Noise Filter
- CK_Fine_Tuning
- ISP 编程区
- 复位源：VDD 电压（低电压检测）
- 复位源：外部复位/ GPIO 共用引脚
- 复位源：看门狗复位&溢出周期

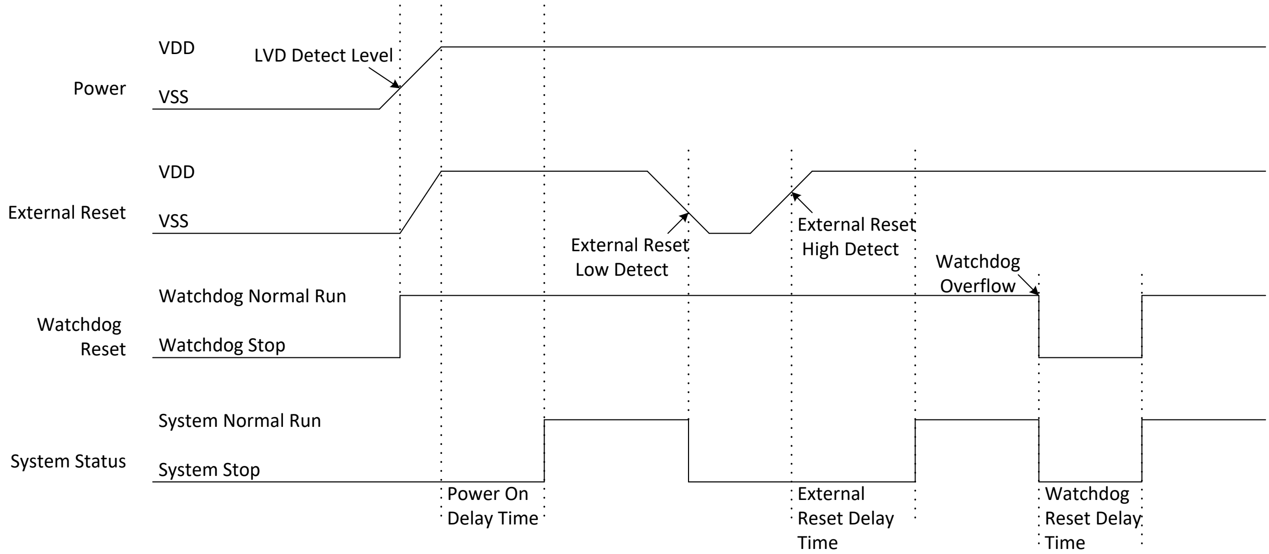


代码选项是系统硬件配置，包括振荡器类型、噪声过滤选项、看门狗定时器操作、LVD 选项、复位引脚选项和 Flash ROM 安全控制。代码选择项如下表:

编译选项	内容	功能说明
Program Memory Security	Security Disable	禁用 ROM 代码加密功能
	Security Enable	使能 ROM 代码加密功能
	Security Configuration	保护所有地址的 ROM 数据，地址 0x1F00 ~ 0x1FDF 除外，只能访问地址 0x1F00 ~ 0x1FDF 的 ROM 数据。
CPU Clock Source	IHRC 32MHz	高速内部 32MHz RC. XIN/XOUT 引脚为双向 GPIO 模式。
	IHRC 32MHz with RTC	高速内部 32MHz RC. XIN/XOUT 引脚由外部 32KHz 晶振驱动。
	X'tal 12MHz	外部高速时钟振荡器的高速晶振/谐振器（例，12MHz）。
	X'tal 4MHz	外部高速时钟振荡器的标准晶振/谐振器（例，4M）。
	External Clock	XIN 脚连接外部时钟（1M ~32M），XOUT 脚为双向 GPIO 模式。
Noise Filter	Disable	禁用 Noise Filter
	Enable	使能 Noise Filter
CK_Fine_Tuning	Disable	禁用 CK_Fine_Tuning
	Enable	使能 CK_Fine_Tuning
ISP Program Area	All Page	所有地址都能执行 ISP 功能。
	Page 248~ Page 254	只有地址 0x1F00 ~ 0x1FDF 能执行 ISP 功能。
LVD	LVD_L	如果 VDD 低于 1.8V，LVD 会复位芯片。
External Reset	Reset with De-bounce	使能外部复位引脚，带 De-bounce
	Reset without De-bounce	使能外部复位引脚，不带 De-bounce
	GPIO with P02	使能内部复位功能，P02 为双向 IO
Watchdog Reset	Always	始终开启看门狗定时器，即使在 STOP 模式和 ILDE 模式下也处于开始状态
	Enable	开启看门狗定时器，但在 STOP 和 ILDE 模式时关闭
	Disable	关闭看门狗定时器
Watchdog Overflow Period	64ms	看门狗定时器时钟源 FILRC /4
	128ms	看门狗定时器时钟源 FILRC /8
	256ms	看门狗定时器时钟源 FILRC /16
	512ms	看门狗定时器时钟源 FILRC /32

7.2 上电流程

LVD，看门狗和外部复位引脚可触发上电流程，在复位信号结束和执行程序之间的时候发生。总的来说，上电流程包括 2 个阶段：电源稳定期和时钟稳定期。

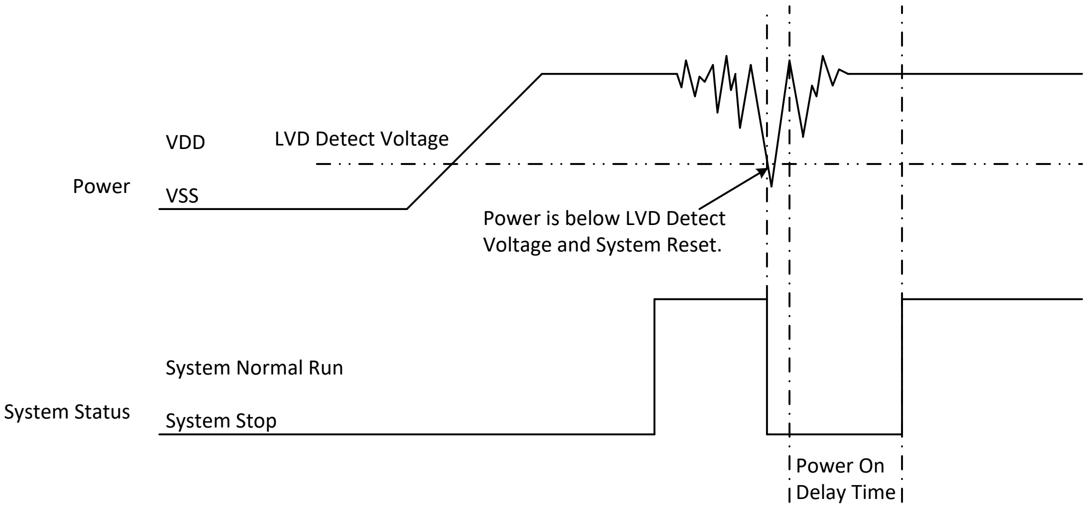


一般情况下，电源稳定期花费 4.6ms，之后单片机会自动获取 CPU 时钟源的配置。所选择的时钟源驱动起来之后，系统会计数 2048 个时钟周期以及 5 个内部低速振荡时钟以确保时钟已稳定。

✸ ***注：在强干扰环境下,用户可以接一个 10 欧姆电阻在 0.1uF 电容和 VDD 引脚之前,这样形成 RC 滤波电路，可以有效的抑制高压脉冲干扰，避免 IC 受损。***

7.3 LVD 复位

低电压检测监控 VDD 引脚的电压，有 1 个电压点：1.8V。根据低电压检测，其比较结果可以作为系统复位信号。下表列出了低电压检测点 LVD_L，以及相对应的 VDD 引脚的条件。



Condition	LVD_L
VDD ≤ 1.8 V	Reset

7.4 看门狗复位

看门狗是周期性的复位信号发生器，用于监控程序的执行流程。其内部定时器可在程序流程的检测点被清零，因此，只有在发生软件问题后才会产生实际的复位信号。通过写入 5AH 到 WDTR 就可以在程序中设置检测点。

```
WDTR = 0x5A;
```

看门狗定时器间隔时间 = 1024 * 1/ (内部低速振荡器频率/WDT 前置分频)

= 1024 / (FILRC /WDT Pre-scaler) …sec

内部低速振荡器	WDT 前置分频	看门狗间隔时间
FILRC =16 kHz	FILRC /4	256/(16000/4)=64ms
	FILRC /8	256/(16000/8)=128ms
	FILRC /16	256/(16000/16)=256ms
	FILRC /32	256/(16000/32)=512ms

看门狗的工作模式如下所示：

Always mode：其内部定时器在 CPU 的所有工作模式（NORMAL，IDLE，SLEEP）下都在计数。

Enable mode：其内部定时器只在 CPU 的 NORMAL 模式下计数，在 IDLE 和 SLEEP 模式下不会触发看门狗复位。

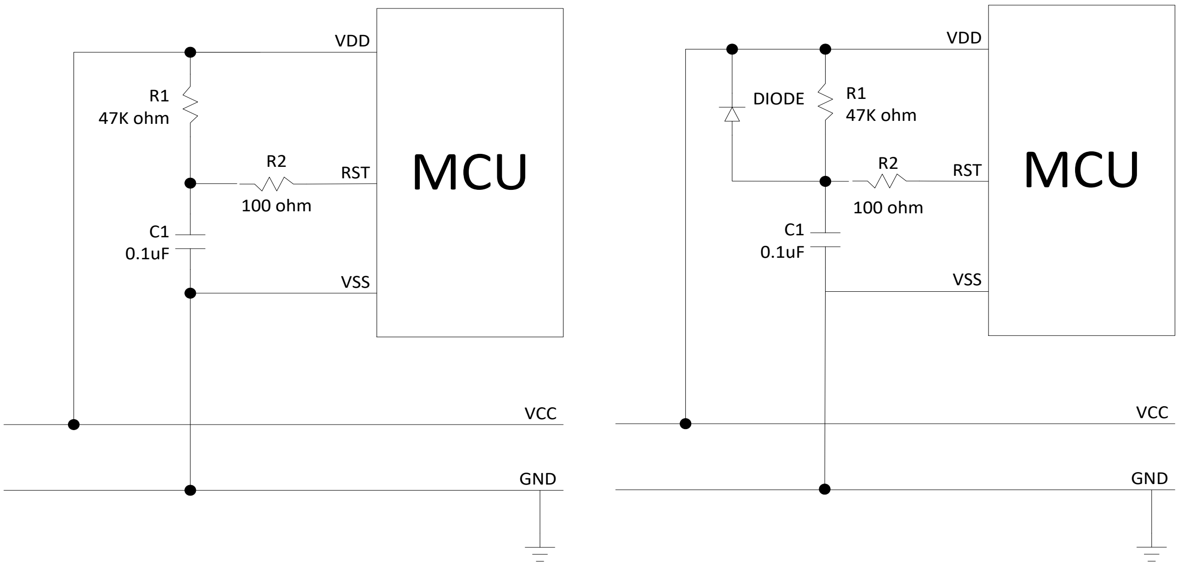
Disable mode：其内部定时器在 CPU 的所有工作模式下都不计数，在这种模式下不会触发看门狗复位。

当看门狗一直处于工作状态时，系统会消耗额外的功耗。

7.5 外部复位引脚

可编程的外部复位引脚的工作状态如下：使能外部复位引脚后，检测其共用引脚的逻辑电平，检测到低电平（小于 30%VDD）后立即触发系统复位直到该引脚恢复到高电平（大于 70%VDD）。

可选择的消抖周期可以增强复位信号的稳定性，区别于立即复位，系统复位需要 8ms 长的逻辑低电平时间来避免跳键的发生。小于这个消抖周期的信号都不会影响 CPU 的执行。



> ✸ *注：*
>
> ***1. 左边的复位电路接法，针对异常电源跌落或掉电复位的情况没有任何保护。***
>
> ***2. 右图 R2 100ohm 电阻的"简单复位电路"和"二极管&RC 复位电路"是必要的,它可限制电流从外部电容 C 灌入复位引脚,导致 ESD 与 EOS 损坏 MCU 引脚。***

7.6 软件复位

连续设置 SRSTREQ 寄存器后会产生软件复位，因此，该程序使能固件的性能以复位单片机（如更新固件后复位）。下面的这段 C 程序代码就因为反复地设置 SRST 寄存器的最小位而导致软件复位。

```
SRST = 0x01;
SRST = 0x01;
```

7.7 复位和上电控制寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PFLAG	POR	WDT	RST	-	-	-	-	-
SRST	-	-	-	-	-	-	-	SRSTREQ
WDTR	WDTR7	WDTR6	WDTR5	WDTR4	WDTR3	WDTR2	WDTR1	WDTR0

PFLAG 寄存器

Bit	Field	Type	Initial	说明
7	POR	R	-	若单片机已经由 LVD 触发复位，则该位自动置 1。
6	WDT	R	-	若单片机已经由看门狗触发复位，则该位自动置 1。
5	RST	R	-	若单片机已经由外部复位引脚触发复位，则该位自动置 1。
4	Reserved	R	0	
3	Reserved	R	0	
0..2	Reserved	R	0	

SRST 寄存器

Bit	Field	Type	Initial	说明
7..1	Reserved	R	0	
0	SRSTREQ	R/W	-	读：若软件已经复位单片机，则该位自动置 1； 写：连续设置该位 2 次触发软件复位。

WDTR 寄存器 (0x86)

Bit	Field	Type	Initial	说明
7..0	WDTR[7:0]	W	-	由 WDTR 寄存器清看门狗定时器，写入 05A 和到 WDTR 寄存器复位看门狗定时器。

8 系统时钟和电源管理

单片机内置 3 个不同的操作模式：NORMAL 模式，IDLE 模式和 STOP 模式以省电。

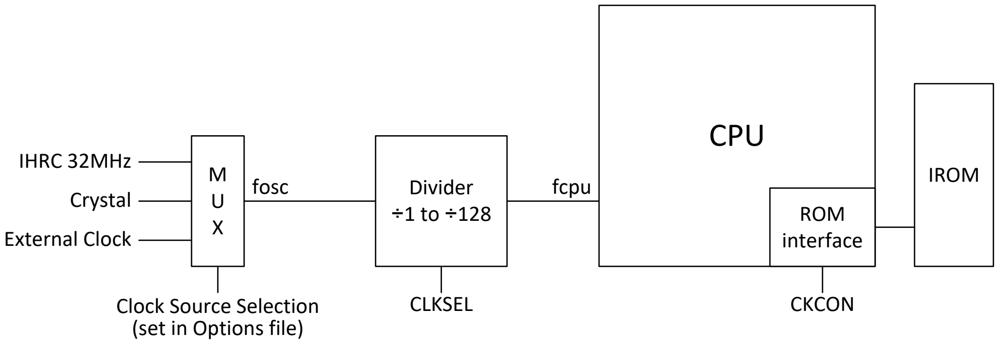
NORMAL 模式是指 CPU 和外设功能都正常工作，系统时钟是通过所选择的时钟源，程序设置的分频数，以及程序设置的 ROM 等待时间所确定的。IDLE 模式是指 CPU 时钟暂停的状态，但保留了外设功能（如定时器，PWM，SPI，UART 和 I2C）。与之相反的是，STOP 模式则禁止所有功能和时钟发生器，直至唤醒信号将系统唤醒进入 NORMAL 模式。

8.1 系统时钟

该单片机包括内置时钟发生器（IHRC 32MHz），晶体/陶瓷驱动器和外部时钟输入。在复位或上电过程中，系统自动加载所选择时钟源。该时钟源可以看作是 Fosc，Fosc 一旦选定就不能再改变。

之后，Fosc 可以分频为 Fosc/4~Fosc/128，由 CLKSEL 寄存器控制。CPU 使用分频之后的时钟作为它的时钟频率（称作 Fcpu）。通过写入 0x69 到 CLKCMD 寄存器使 CLKSEL 的设置生效。

```
CKCON = 0x70;       // For change safely the system clock
CLKSEL = 0x05;      // Set fcpu = fosc / 4
CLKCMD = 0x69;      // Apply CLKSEL's setting
CKCON = 0x00;       // IROM fetch = fcpu / 1
```



ROM 接口位于 CPU 和 IROM（程序存储器）之间，可设置 ROM 读取周期以支持低速程序存储器。

IROM fetching cycle（Instruction cycle） ≤ 8MHz

✷ ***注：对于用户在 C 语言或汇编语言中开发的程序，程序的第一行"必须设置"CLKSEL= 0x05~0x00, CLKMD= 0x69 然后设置 CKCON= 0x00~0x70，这个优先级不能被修改。***

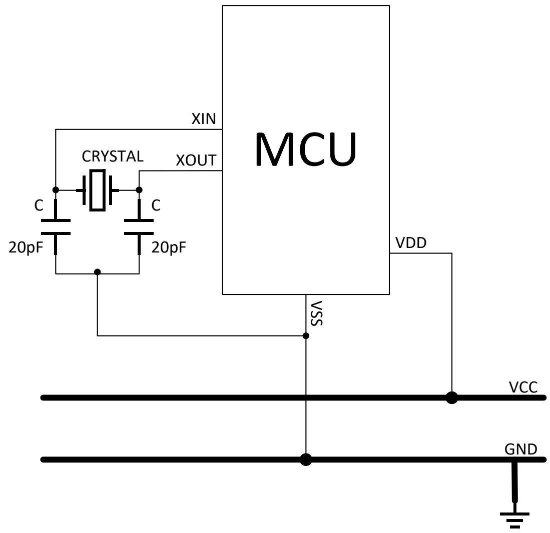
系统时钟速率和程序内存扩展周期限制见下表。

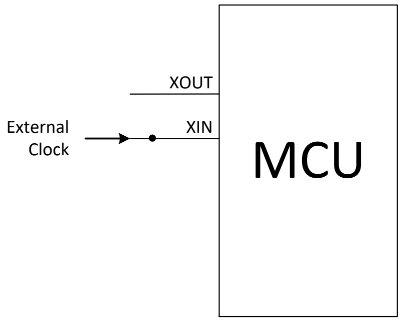
代码选项 CPU 时钟源	Fcpu = CLKSEL[2:0]	IROM Fetch = CKCON[6:4]
IHRC 32M IHRC 32M with RTC External Clock (16-32MHz)	仅支持 000 = fosc / 128 001 = fosc / 64 010 = fosc / 32 011 = fosc / 16 100 = fosc / 8 101 = fosc / 4	仅支持 **000 = fcpu / 1 => 建议!** 001 = fcpu / 2 010 = fcpu / 3 011 = fcpu / 4 100 = fcpu / 5 101 = fcpu / 6 110 = fcpu / 7 111 = fcpu / 8
X’tal 12M (Crystal 8-16MHz) External Clock (8-16MHz)	仅支持 000 = fosc / 128 001 = fosc / 64 010 = fosc / 32 011 = fosc / 16 100 = fosc / 8 101 = fosc / 4 110 = fosc / 2	
X’tal 12M (Crystal 4-8MHz) X’tal 4M (Crystal 1-4MHz) External Clock (1-8MHz)	仅支持 000 = fosc / 128 001 = fosc / 64 010 = fosc / 32 011 = fosc / 16 100 = fosc / 8 101 = fosc / 4 110 = fosc / 2 111 = fosc / 1	

8.2 高速时钟

高速时钟分为内部和外部两种类型。外部高速时钟包括 4MHz、12 MHz crystal/ceramic 和外部时钟输入模式。内部高速振荡器是 32MHz RC 类型。这些高速振荡器由 SN8F5713_OPTIONS.A51 选择。

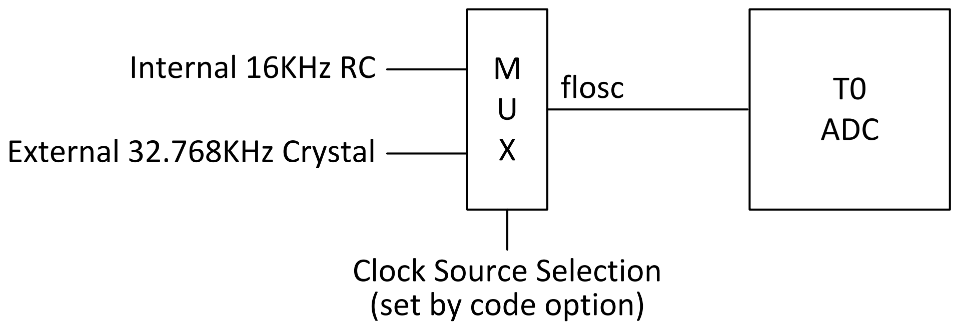
- IHRC 32M：系统高速时钟源是内部高速 32MHz RC 型振荡器。在此模式下，XIN 和 XOUT 引脚是双向 GPIO 模式，且不与任何外部振荡器设备连接。
- IHRC 32M，带 RTC：系统高速时钟源是内部高速 32MHz RC 型振荡器。RTC 时钟源是外部低速 32768Hz 晶振。在此模式下，XIN 和 XOUT 引脚限定为驱动外部 32768Hz 晶振，且禁用 GPIO 功能。
- X'tal 12M：系统高速时钟源是外部高速 crystal/ceramic。振荡器带宽是 10MHz~16 MHz，用 20 pf 电容器连接到 XIN/ XOUT 引脚接地。
- X'tal 4M：系统高速时钟源是外部高速 crystal/resonator。振荡器带宽是 1MHz~10MHz，用 20 pf 电容器连接到 XIN/ XOUT 引脚接地。
- 外部时钟：系统高速时钟源为外部时钟输入模式。输入信号只连接到 XIN 引脚，而 XOUT 引脚是双向 GPIO 模式。





8.3 低速时钟

SN8F5713 支持低速时钟（Flosc）用于特殊功能，如 T0 和 ADC。Flosc 有 2 个时钟源：内部 16KHz RC（ILRC）和外部 32.768KHz 晶振，由代码选项 IHRC 32M RTC 等控制。在 IHRC 32M RTC 模式下，P0.0 和 P0.1 引脚切换为晶振模式，驱动芯片外 32.768KHz 晶振。



8.4 电源管理

复位信号和上电结束后，CPU 以 Fcpu 的速率开始执行程序。在此情况下系统和所有外设以 NORMAL 模式开始工作。

PCON 寄存器的最低 2 位（bit0-IDLE 和 bit1-STOP）控制单片机的电源管理部分。

若 IDLE 位设置为 1，只有 CPU 时钟源被关闭。因此，在这种状态下，外设功能（如定时器，PWM 和 I2C）和时钟发生器（IHRC 32MHz/晶振驱动器）仍然正常工作。P0/P1 输入的任何改变和中断时间都会导致单片机返回到 NORMAL 模式，IDLE 位自动清零。

- IDLE 模式下所有功能正常工作，只有 CPU 时钟源被关闭。
- IDLE 模式下唤醒源是 P0/P1 电平翻转和中断事件。

若 STOP 位设置为 1，CPU，外设和时钟发生器都处于停止状态，在这个模式下，寄存器中存储的数据和 RAM 都保持不变。P0/P1 输入的任何改变都可将单片机唤醒并使系统重新开始执行，STOP 位自动清零。

- STOP 模式下 CPU，外设和时钟发生器都处于停止状态。
- STOP 模式下唤醒源是 P0/P1 电平翻转。

用户用 C 编译器开发时，强烈建议使用 IDLE 和 STOP 宏来控制单片机的系统模式，来代替直接设置 IDLE 和 STOP 位。

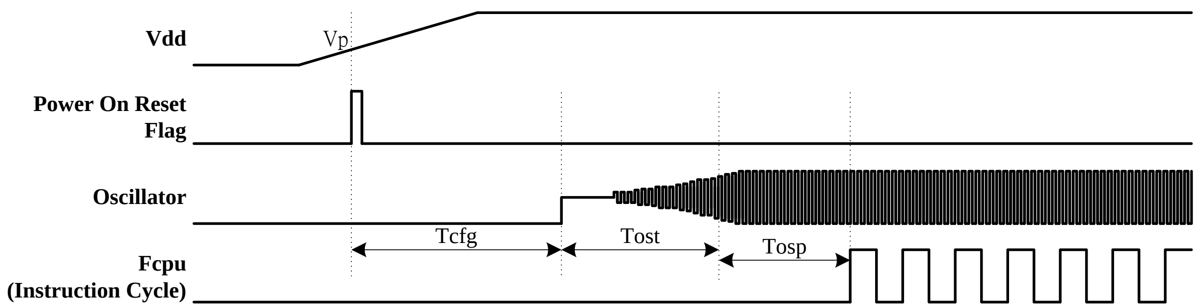
```
IDLE();
STOP();
```

STOP 模式下，SN8F5713 内置 STWK 位（PCON 寄存器的第四位）来使能或禁止 Flosc 时钟。若 STWK=0，Fosc 和 Flosc 都禁止；STWK=1，Flosc 保持运行。

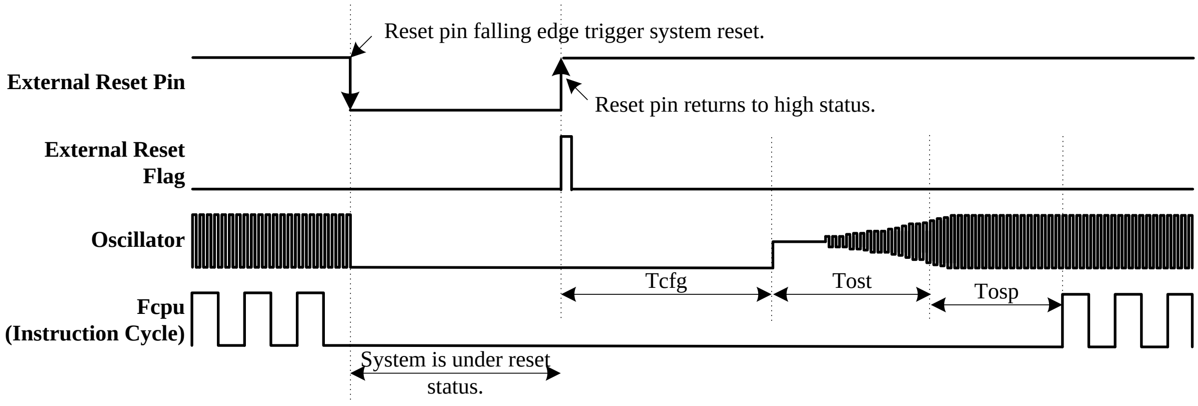
8.5 系统时钟计时

参数	符号	说明	典型值
硬件配置时间	Tcfg	$8^*F_{ILRC} + 2^{17*}F_{IHRC}$	4.6ms @ FILRC = 16KHz FIHRC = 32MHz
振荡器启动时间	Tost	启动时间取决于振荡器的材料、制造厂和架构。通常情况下，低速振荡器的启动时间低于高速振荡器的。RC 型振荡器的启动时间快于晶体型振荡器。	-
振荡器预热时间	Tosp	复位条件下的振荡器预热时间。 2048*Fhosc+ 5*FILRC （上电复位，LVD 复位，看门狗复位，外部复位引脚激活）	64ms @ Fhosc = 32KHz 825us @ Fhosc = 4MHz 441us @ Fhosc = 16MHz
		掉电模式唤醒条件下的振荡器预热时间。 2048*Fhosc+5*FILRC ……晶体/共鸣器型振荡器，例如 32768Hz crystal, 4MHz crystal, 16MHz crystal… 64*Fhosc+5*FILRC……RC 型振荡器，例如内部高速 RC 型振荡器。	X'tal: 825us @ Fhosc = 4MHz 441us @ Fhosc = 16MHz RC: 315us @ Fhosc = 16MHz

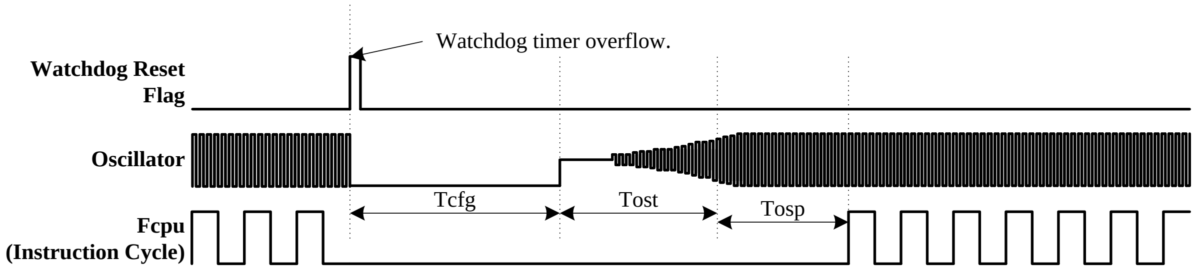
- 上电复位计时



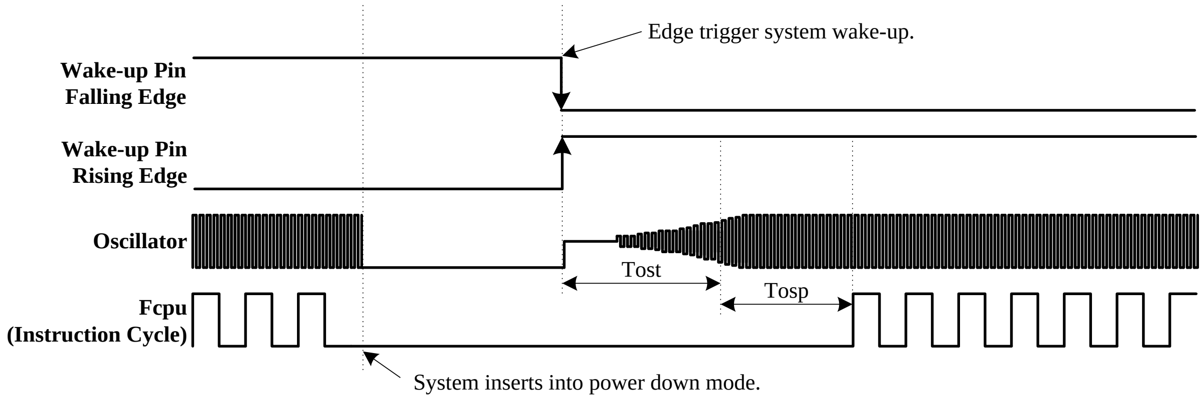
- 外部复位引脚复位计时



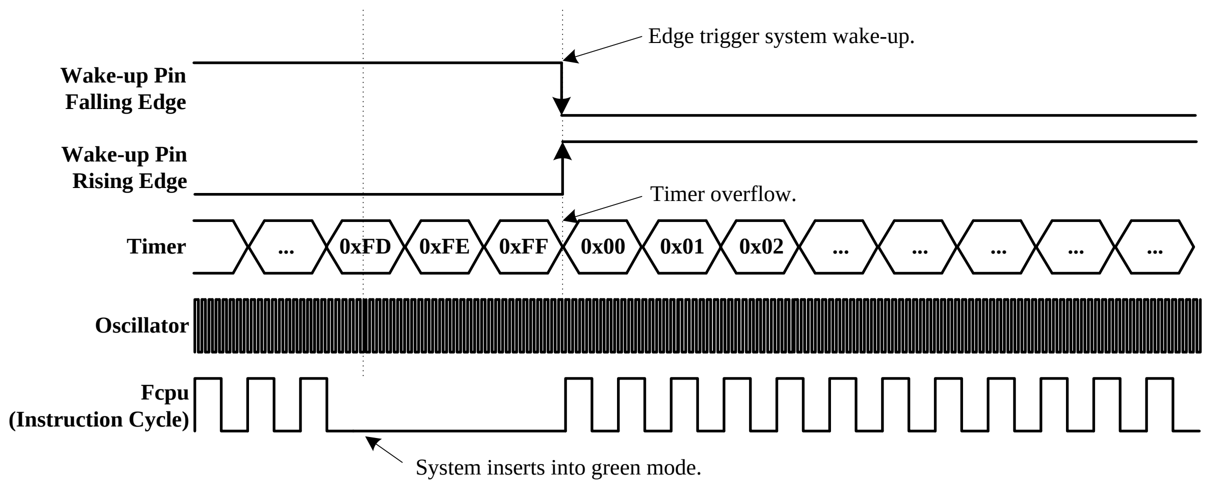
- 看门狗复位计时



- 掉电模式唤醒计时

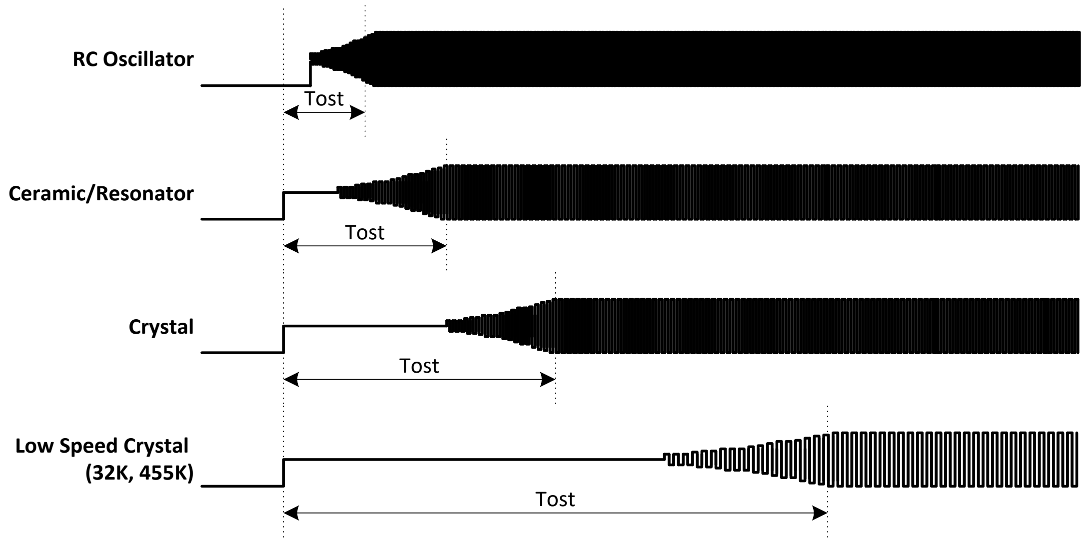


- 空闲模式唤醒计时



- 振荡器启动时间

启动时间取决于振荡器的材料、制造厂和架构。通常情况下，低速振荡器的启动时间低于高速振荡器的。



8.6 系统时钟和电源管理寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CKCON	-	PWSC2	PWSC1	PWSC0	-	-	-	-
CLKSEL	-	-	-	-	-	CLKSEL2	CLKSEL1	CLKSEL0
CLKCMD	CMD7	CMD6	CMD5	CMD4	CMD3	CMD2	CMD1	CMD0
PCON	SMOD	-	-	-	P2SEL	GF0	STOP	IDLE
P1W	P17W	P16W	P15W	P14W	P13W	P12W	P11W	P10W
SYSMOD	-	-	-	-	-	-	BIASEN	STWK

CKCON 寄存器(0x8E)

Bit	Field	Type	Initial	说明
7	Reserved	R	0	
6..4	PWSC[2:0]	R/W	111	增加读取程序存储器的周期。 000：无； 001：1 个周期； 010：2 个周期； 011：3 个周期； 100：4 个周期； 101：5 个周期； 110：6 个周期； 111：7 个周期；
3..0	Reserved	R	0	

CLKCMD 寄存器(0xE6)

Bit	Field	Type	Initial	说明
7..0	CMD[7:0]	W	0x00	写入 0x69 来应用 CLKSEL 的设置。

P1W 寄存器(0x91)

Bit	Field	Type	Initial	说明
7..0	P1nW	R/W	0	0：禁用 P1.n 唤醒功能； 1：使能 P1.n 唤醒功能。

CLKSEL 寄存器(0xE5)

Bit	Field	Type	Initial	说明
7..3	Reserved	R	0x00	
2..0	CLKSEL[2:0]	R/W	111	CLKSEL 中的设置将在写 CLKCMD 之后生效。 000：fcpu = fosc / 128； 001：fcpu = fosc / 64； 010：fcpu = fosc / 32； 011：fcpu = fosc / 16； 100：fcpu = fosc / 8； 101：fcpu = fosc / 4； 110：fcpu = fosc / 2； 111：fcpu = fosc / 1。

PCON 寄存器(0x87)

Bit	Field	Type	Initial	说明
7				参考其他章节
6..4	Reserved	R	0x00	
3	P2SEL	R/W	1	高位地址字节配置位。执行 MOVX @Ri 期间选择地址的高字节（XRAM[15:8]）。 0：“XRAM[15:8]” = “P2REG”，“P2REG”为 P2 输出寄存器的内容； 1：“XRAM[15:8]” = 0x00。
2	GF0	R/W	0	通用标志。
1	STOP	R/W	0	1：单片机切换到 STOP 模式。
0	IDLE	R/W	0	1：单片机切换到 IDLE 模式。

SYSMOD 寄存器 (0xCE)

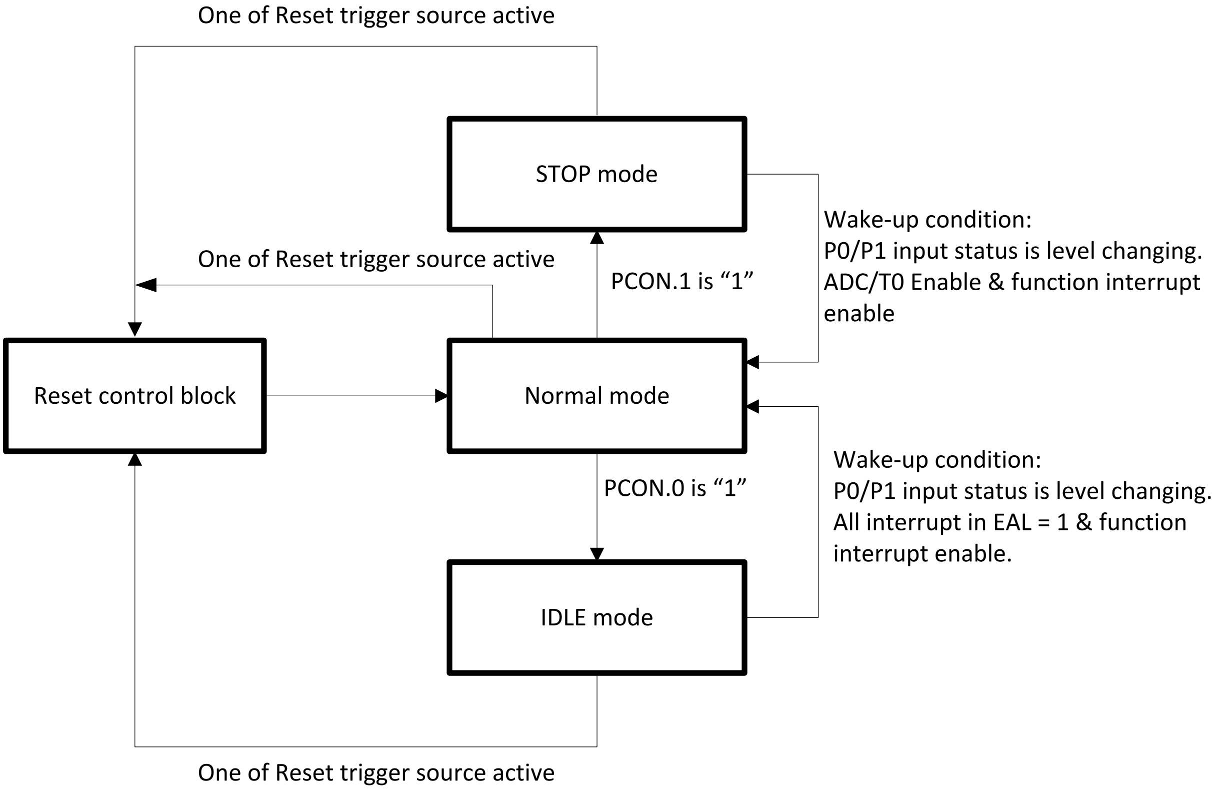
Bit	Field	Type	Initial	说明
7..2	Reserved	R	0x00	
1				参考其他章节。
0	STWK	R/W	0	0：STOP 模式下 fosc 和 flosc 均挂起； 1：STOP 模式下 flosc 继续运行。

* 进入 STOP 模式之前，必须要先于 STOP 位设置 STWK 位。

9 系统工作模式

芯片在不同的时钟频率和不同的省电原因下有三种操作模式。这些模式控制振荡器、op-code 操作和模拟外围设备的操作。

- NORMAL 模式：系统高速工作模式
- IDLE 模式：系统空闲模式
- STOP 模式：系统省电模式



工作模式时钟控制见下表：

工作模式	NORMAL 模式	IDLE 模式	STOP 模式
IHRC	IHRC, IHRC RTC：运行 Ext. OSC：禁止	IHRC, IHRC RTC：运行 Ext. OSC：禁止	Stop
ILRC	运行	运行	STWK=1：运行 Watchdog always：运行
Ext. OSC	IHRC：禁止 IHRC RTC, Ext. OSC：运行	IHRC：禁止 IHRC RTC, Ext. OSC：运行	Stop
CPU 指令	执行	Stop	Stop
定时器 0 （定时器，事件计数器）	由 TR0 激活	由 TR0 激活	未激活
定时器 1 （定时器，事件计数器	由 TR1 激活	由 TR1 激活	未激活
PWM1	设置为使能时有效	设置为使能时有效	未激活
I2C/UART	设置为使能时有效	设置为使能时有效	未激活
ADC	设置为使能时有效	设置为使能时有效	激活为 ADC 时钟源是 Flosc (ILRC or Ext. 32kHz) & STWK=1
看门狗定时器	通过看门狗 代码选项	通过看门狗 代码选项	通过看门狗 代码选项
内部中断	全部激活	全部激活	全部未激活
外部中断	全部激活	全部激活	全部未激活
唤醒源	-	P0, P1，复位, EAL 中的所有中断 = 1 & 功能中断使能	P0, P1，复位, ADC/T0 使能&时钟源是 Flosc (ILRC or Ext. 32kHz) & STWK=1 &功能中断使能

- Ext. OSC：外部高速振荡器 (XIN/XOUT).
- IHRC：内部高速振荡器 RC 型
- ILRC：内部低速振荡器 RC 型

9.1 NORMAL 模式

NORMAL 是系统高速时钟工作模式。系统时钟源来自高速振荡器。执行程序。上电以及复位触发释放后，系统进入到 NORMAL 模式执行程序。当从停止/空闲模式下唤醒系统，系统也将进入 NORMAL 模式。在 NORMAL 模式下，高速振荡器激活，功耗是所有工作模式中最大的。

- 执行程序，且所有功能是可控的。
- 系统高速运行。
- 高速振荡器和内部低速 RC 振荡器激活。
- NORMAL 模式可以通过 PCON 寄存器切换到其他操作模式。
- STOP/IDLE 模式可唤醒为 NORMAL 模式。

9.2 STOP 模式

STOP 模式没有程序执行，也没有振荡器工作。只有内部保持电路激活，保持所有控制 IO 状态、寄存器状态和 SRAM 内容。STOP 模式由 P0/P1 电平改变触发。P0 唤醒功能通常是使能的。STOP 模式唤醒为 NORMAL 模式. 进入 STOP 模式由 PCON 寄存器的 PCON.1 控制. 当 PCON.1=1 时，系统进入 STOP 模式。从 STOP 模式系统唤醒后，PCON.1 自动清零。

- 程序停止执行，且禁用所有功能。
- 振荡器包括外部高速振荡器，内部高速振荡器和内部低速振荡器停止。
- 内部保持电路激活，保持所有控制 IO 状态、寄存器状态和 SRAM 内容
- 系统从 STOP 模式唤醒后进入到 NORMAL 模式。
- STOP 模式唤醒源是 P0/P1 电平改变触发, ADC/T0 使能&时钟源为 Flosc(ILRC 或 Ext.32 khz)& STWK = 1 &中断功能开启。

9.3 IDLE 模式

IDLE 模式与 STOP 模式不同。在 STOP 模式下，所有功能和硬件设备都被禁用。但在 IDLE 模式下，系统时钟源持续运行，所以 IDLE 模式的功耗大于 STOP 模式的功耗。在 IDLE 模式下，程序不执行，但定时器唤醒功能是激活启用的, 定时器时钟源是不间断的系统时钟. IDLE 模式有 2 个唤醒源. 一个是 P0/P1 电平改变触发唤醒。另一个是 EAL=1 中断&功能中断使能。这意味着用户可以用中断使能设置功能，而系统直到中断发出才被唤醒。进入 IDLE 模式由 PCON 寄存器的 PCON.0 位控制。PCON.0=1 时，系统进入 IDLE 模式。系统从 IDLE 模式唤醒后，PCON.0 位自动清零。

- 停止执行程序，且禁用全部功能。
- 定时器溢出中断可以唤醒系统。
- 作为系统时钟源的振荡器保持运行，而其他振荡器是否工作取决于系统操作模式配置。
- 如果从 NORMAL 模式进入 IDLE 模式，系统在唤醒后进入 NORMAL 模式。
- IDLE 模式的唤醒源是 P0/P1 电平改变触发。
- 时钟源来自于系统时钟的硬件模块，如果有被使能，在 ILDE 模式下都处于工作状态，例如：定时器、PWM、事件计数器…。
- 在有打开总中断和使能中断功能时，任何中断事件都会唤醒 IDLE 模式。

9.4 唤醒

STOP 模式或 IDLE 模式下，不执行程序。唤醒触发可以唤醒系统恢复为 NORMAL 模式。唤醒触发来源是外部触发(P0/P1 电平改变)和内部触发（EAL= 1 &功能使能）。唤醒功能内置有中断操作，发出请求标志位，触发系统执行中断服务程序，将系统唤醒。

当系统在 STOP 模式下，高时钟振荡器停止。当从 STOP 模式唤醒时，MCU 等待 2048 个外部高速振荡时钟+ 4 个内部低速振荡时钟，以及 64 个内部高速振荡器时钟+ 4 个内部低速振荡时钟的唤醒时间，以稳定振荡电路。唤醒时间后，系统进入 NORMAL 模式。

外部高速时钟振荡器唤醒时间值如下。

唤醒时间 = 1/Fosc * 2048 (sec) + 1/Flosc * 4+ 高速时钟启动时间

例：在 STOP 模式下，唤醒系统。唤醒时间后，系统进入 NORMAL 模式。唤醒时间如下。

唤醒时间= 1/Fosc * 2048 + 1/Flosc * 4 = 0.762 ms (Fosc = 4MHz，Flosc=16KHz)

总唤醒时间 = 0.762 ms + 振荡器启动时间

RC 型内部高速时钟振荡器唤醒时间值如下。

唤醒时间= 1/Fosc * 64 (sec) + 1/Flosc * 4 +高速时钟启动时间

例：在 STOP 模式下，唤醒系统。唤醒时间后，系统进入 NORMAL 模式。唤醒时间如下。

唤醒时间= 1/Fosc * 64 +1/Flosc * 4= 252 us (Fhosc = 32MHz，Flosc=16KHz)

> ✸ ***注：高速时钟的启动时间与 VDD 和振荡器类型有关。***

在 STOP 模式和 IDLE 模式下，具有唤醒功能的 I/O 口能够唤醒系统进入 NORMAL 模式。唤醒触发边缘是在上升或下降沿边缘的电平改变。P0 和 P1 具有唤醒功能。P0 唤醒功能一直是使能的，但 P1 由 P1W 寄存器控制。

P1W 寄存器(0x91)

Bit	Field	Type	Initial	说明
7..0	P1nW	R/W	0	0：禁用 P1.n 唤醒功能； 1：使能 P1.n 唤醒功能。

10 中断

这颗 MCU 包含 10 个中断源（3 个外部中断和 7 个外部中断），分为 4 个优先级别。每个中断源包含 1 个或多个中断请求标志。有中断发生时，相对应的中断请求位置为逻辑 1。若同时使能中断使能位和全局中断使能位（EAL=1）时，有中断请求时则执行该中断服务程序（ISR）。有些中断请求标志位需要软件清零，但大多数中断请求标志位由硬件自动清零。最后，执行 RETI 指令后，整个 ISR 结束。中断源概述，中断向量，优先级别和控制位如下表所示：

中断	使能中断	请求(IRQ)	IRQ 清除	优先级/向量
系统复位	-	-	-	0 / 0x0000
INT0	EX0	IE0	自动	1 / 0x0003
INT1	EX1	IE1	自动	2 / 0x0083
INT2	EX2	IE2	自动	3 / 0x0043
PWM1	EPWM1	PWM1F	自动	4 / 0x000B
UART TX	EU0TX	TI0	软件清零	5 / 0x0013
I2C	EI2C	SI	软件清零	6 / 0x0093
UART RX	EU0RX	RI0	软件清零	7 / 0x0053
ADC	EADC	ADCF	自动	8 / 0x001B
Timer 1	ET1	TF1	自动	9 / 0x00A3
Timer 0	ET0	TF0	自动	10 / 0x0063

10.1 中断操作

中断操作由中断请求标志位和中断使能位控制。中断请求标志位显示中断源的状态，与中断功能的状态（使能或禁止）无关。同时使能中断使能位和全局中断使能位（EAL=1）且中断请求标志位有效时，程序计数器指向中断向量（0x03-0x93），系统执行相对应的中断服务程序 ISR。

10.2 中断优先级

每个中断源都有特定的优先级别。若同时发生 2 个中断，系统会先执行优先级别高的 ISR，然后再执行优先级别低的 ISR。下一次的 ISR 必须要等待前面的 ISR 执行完成之后才可以执行，不用理会中断的优先级别。

对应特定的优先权需求，需要用到 4 级的优先级别（级别 0-级别 3）。所有的中断源按优先级别分为 6 大类（Group0-Group5），每组设置为同样的特定优先级别，由寄存器 IP0/IP1 设置，级别 3 最高，级别 0 最低。同组的中断源共用同样的优先级别，对于同样的优先级别，优先权的规则按照默认的优先权。

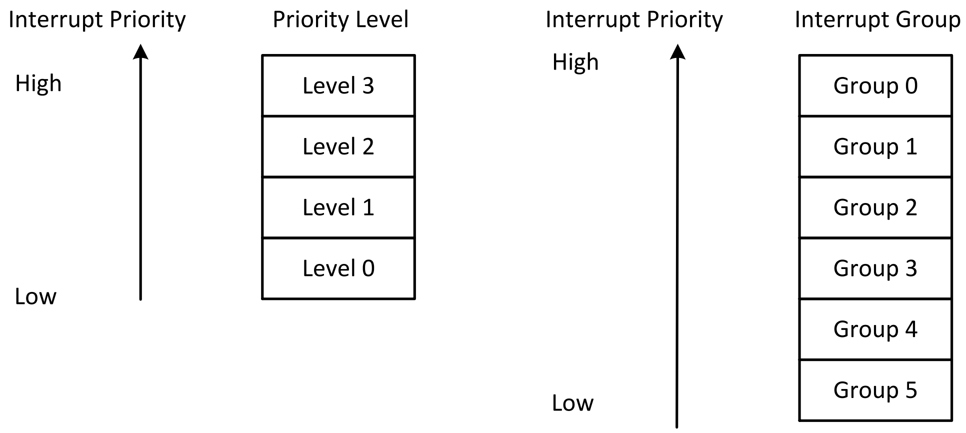
优先级别	IP1.x	IP0.x
Level 0	0	0
Level 1	0	1
Level 2	1	0
Level 3	1	1

首先执行优先级别较高的 ISR，甚至可以打断执行中的优先级别较低的 ISR，直到优先级别较高的 ISR 执行完成之后再执行优先级别较低的 ISR。

Group	中断源			
Group 0	INT0	INT1	INT2	
Group 1	PWM1			
Group 2	UART TX	I2C	UART RX	
Group 3	ADC			
Group 4	T1	T0		
Group 5				

当多个中断请求发生时，必须先执行最高优先级的请求。根据自然优先权和优先级，选择最高优先级的请求。步骤如下：

1. 选择所有组之间优先级最高的组。
2. 在最高优先级组之间选择自然优先权最高的组。
3. 选择最高优先级组内的有最高自然优先权的 ISR。

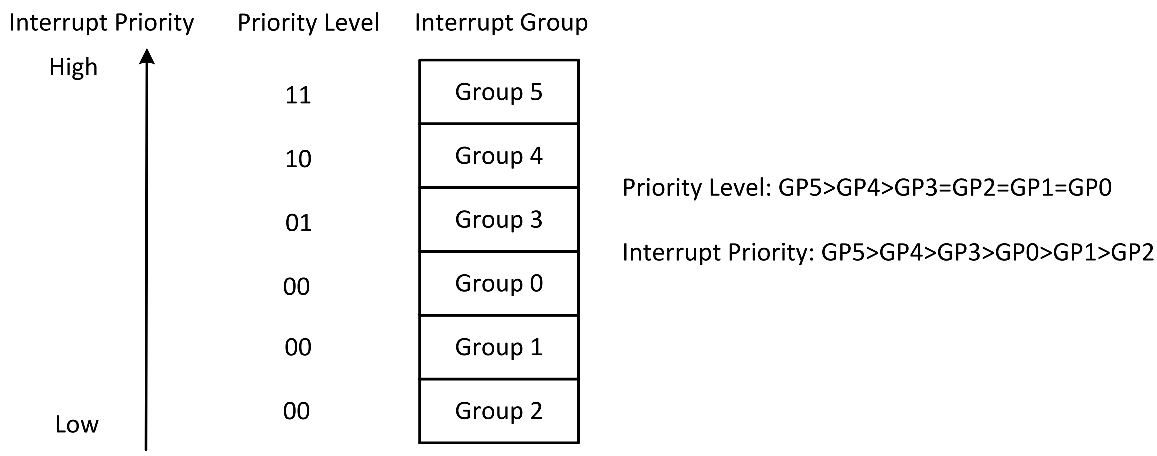


Higher priority level has higher priority

All groups within the same priority level
IP0.0 = IP0.1 = IP0.2 = IP0.3 = IP0.4 = IP0.5
IP1.0 = IP1.1 = IP1.2 = IP1.3 = IP1.4 = IP1.5

例，group5 具有最高优先级，而 group0 ~ group2 具有最低优先级。这意味着 group5 中的中断向量具有最高的中断优先权，group4 具有第二中断优先权，而 group3 具有第三中断优先权。Group0 ~ group2 具有相同的优先级，因此遵循自然优先规则。故而中断优先权：group5 > group4 > group3 > group0 > group1 > group2。

```
MOVIP0, #00101000B        ; 设置 group0 - group5 的不同优先级。
MOVIP1, #00110000B
```



IP0, IP1 寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IP0	-	-	IP05	IP04	IP03	IP02	IP01	IP00
IP1	-	-	IP15	IP14	IP13	IP12	IP11	IP10

IP0 寄存器(0XA9)

Bit	Field	Type	Initial	说明
5..0	IP0[5:0]	R/W	0	中断优先权。和 IP1 寄存器的相对应的位结合在一起可以指定相对应的中断的优先级别。
Else	Reserved	R	0	

IP1 寄存器(0XB9)

Bit	Field	Type	Initial	说明
5..0	IP1[5:0]	R/W	0	中断优先权。和 IP0 寄存器的相对应的位结合在一起可以指定各中断的优先级组。
Else	Reserved	R	0	

10.3 中断寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN0	EAL	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
IEN2	-	-	-	-	EPWM1	EX2	-	EADC
IRCON2	-	-	-	-	PWM1F	IE2	-	ADCF
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
S0CON	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
I2CCON	CR2	ENS1	STA	STO	SI	AA	CR1	CR0

IEN0 寄存器(0XA8)

Bit	Field	Type	Initial	说明
7	EAL	R/W	0	所有中断控制使能位。 0：禁用所有中断功能； 1：使能所有中断功能。
6	EI2C	R/W	0	I2C 中断控制位。 0：禁用 I2C 中断功能； 1：使能 I2C 中断功能。
5	EU0RX	R/W	0	UART RX 中断控制位。 0：禁用 UART RX 中断功能； 1：使能 UART RX 中断功能。
4	EU0TX	R/W	0	UART TX 中断控制位。 0：禁用 UART TX 中断功能； 1：使能 UART TX 中断功能。
3	ET1	R/W	0	T1 定时器中断控制位。 0：禁用 T1 中断功能； 1：使能 T1 中断功能。
2	EX1	R/W	0	外部 P0.6 中断 (INT1)控制位。 0：禁用 INT1 中断功能； 1：使能 INT1 中断功能。
1	ET0	R/W	0	T0 timer 中断控制位。 0：禁用 T0 中断功能； 1：使能 T0 中断功能。
0	EX0	R/W	0	外部 P0.3 中断(INT0) 控制位。 0：禁用 INT0 中断功能； 1：使能 INT0 中断功能。

IEN2 寄存器 (0X9A)

Bit	Field	Type	Initial	说明
3	EPWM1	R/W	0	PWM1 中断控制位。 0 = 禁用 PWM1 中断功能； 1 = 使能 PWM1 中断功能。
2	EX2	R/W	0	外部 P0.7 中断(INT2) 控制位。 0：禁用 INT2 中断功能； 1：使能 INT2 中断功能。
0	EADC	R/W	0	ADC 中断控制位。 0：禁用 ADC 中断功能； 1：使能 ADC 中断功能。
Else	Reserved	R	0	

TCON 寄存器 (0X88)

Bit	Field	Type	Initial	说明
7	TF1	R/W	0	T1 定时器外部重装中断请求标志位。 0：无 T1 中断请求； 1：T1 请求中断。
5	TF0	R/W	0	T0 定时器外部重装中断请求标志位。 0：无 T0 中断请求； 1：T0 请求中断。
3	IE1	R/W	0	外部中断 P0.6（INT1）中断请求标志位。 0：无 INT1 中断请求； 1：INT1 请求中断。
1	IE0	R/W	0	外部中断 P0.3（INT0）中断请求标志位。 0：无 INT0 中断请求； 1：INT0 请求中断。
Else				参考其它章节。

IRCON2 寄存器 (0XBF)

Bit	Field	Type	Initial	说明
3	PWM1F	R/W	0	PWM1 中断请求标志位。 0：无 PWM1 中断请求； 1：PWM1 请求中断。
2	IE2	R/W	0	外部中断 P0.7（INT2）中断请求标志位。 0：无 INT2 中断请求； 1：INT2 请求中断。
0	ADCF	R/W	0	ADC 中断请求标志位。 0：无 ADC 中断请求； 1：ADC 请求中断。
Else	Reserved	R	0	

S0CON Register (0X98)

Bit	Field	Type	Initial	说明
1	TI0	R/W	0	UART 发送中断请求标志位，显示 UART 串行传输的完成状态。在模式 0 位 8 结束时，或者其它模式的停止位开始时由硬件设置为 1；必须由软件清零。 0：无 UART 发送中断请求； 1：UART 发送请求中断。
0	RI0	R/W	0	UART 接收中断请求标志位，显示 UART 串行接收的完成状态。在模式 0 位 8 结束时，或者其它模式的停止位中间时由硬件设置为 1；必须由软件清零。 0：无 UART 接收中断请求； 1：UART 接收请求中断。
Else				参考其它章节。

I2CCON 寄存器(0XDC)

Bit	Field	Type	Initial	说明
3	SI	R/W	0	串行中断标志位。 当进入了 I2C 的 26 个状态当中的 25 个状态时，SI 标志位会被硬件置 1，只有当 I2C 状态寄存器为 F8h 时，SI 标志位才没有被置 1，表示没有可用的相关状态信息。SI 标志位必须由软件清零，必须通过写 0 到 SI 标志才可以清 SI 标志位，写入 1 到该位并不能更改 SI 的值。
Else				参考其它章节。

10.4 示例代码

定义中断向量。中断服务程序在用户程序之后。

```
            ORG     0               ; 0000H
            JMP     START           ; Jump to user program address.
            …
            ORG     0X0003          ; Jump to interrupt service routine address.
            JMP     ISR_INT0
            ORG     0X0043
            JMP     ISR_INT2
            …
            ORG     0X0063
            JMP     ISR_T0
            …
            ORG     0X00EC
START:                              ; 00ECH, The head of user program.
            …                       ; User program.
            …
            JMP     START           ; End of user program.
            …
ISR_INT0:                           ; The head of interrupt service routine.
            PUSH    ACC             ; Save ACC to stack buffer.
            PUSH    PSW             ; Save PSW to stack buffer.
            …
            POP     PSW             ; Load PSW from stack buffer.
            POP     ACC             ; Load ACC from stack buffer.
            RETI                    ; End of interrupt service routine.
ISR_T0:                             ;
            PUSH    ACC             ; Save ACC to stack buffer.
            PUSH    PSW             ; Save PSW to stack buffer.
            …
            POP     PSW             ; Load PSW from stack buffer.
            POP     ACC             ; Load ACC from stack buffer.
            RETI                    ; End of interrupt service routine.
            …
ISR_INT2:                           ;
            PUSH    ACC             ; Save ACC to stack buffer.
            PUSH    PSW             ; Save PSW to stack buffer.
            …
            POP     PSW             ; Load PSW from stack buffer.
            POP     ACC             ; Load ACC from stack buffer.
            RETI                    ; End of interrupt service routine.

            END                     ; End of program.
```

11 MDU

乘除法单元是内置的算术协同处理器，能使单片机执行额外扩展的算术运算。此单元提供 32 位无符号除法、16 位无符号乘法、移位和规范化操作。这些操作由写 MD0 到 MD5 寄存器的不同顺序来定义。

11.1 乘法（16 位 x16 位）

乘法的要素包括三个部分：被乘数、乘数和乘积。开始一个乘法需要按以下的写顺序：MD0(被乘数的低字节)、MD4(乘数的低字节)、MD1(被乘数的高字节)和 MD5(乘数的高字节)。

写 MD5 寄存器结束之后，乘法是自动开始，并用 11 个 CPU 周期完成。这一项操作的乘积通过一个特定的序列：MD0(LSB)、MD1、MD2、MD3(MSB)寄存器便能读取。

11.2 除法（32 位/16 位和 16 位/16 位）

MDU 支持两种除法：32 位/16 位、16 位/16 位。第一种需要 17 个 CPU 周期来计算，而第二种只需要 9 个周期。

32 位的除法由写寄存器的一个特定序列来开始：MD0,MD1、MD2,MD3,MD4 和 MD5。在这种情况下，32 位的被除数存储在 MD3(最高有效位)到 MD0 寄存器中，以及 16 位除数存储在 MD5 和 MD4 寄存器中(MSB 在 MD5 寄存器中)。

16 位的除法只需要四个寄存器合作。16 位被除数存储在 MD1 和 MD0 寄存器中，16 位除数存储在 MD5 和 MD4 寄存器(MD1 和 MD5 的最高有效位)。正确执行顺序是“MD0、MD1、MD4 和 MD5。

MD5 写寄存器后，MDU 开始计算。在生成结果之前，这需要 9 到 17 个 CPU 周期，取决于被除数的长度。32 位除法中，商存储在 MD3 到 MD0 寄存器中，而 16 位除法中则存储在 MD1 到 MD0 寄存器中 (LSB 在 MD0 寄存器中)。余数存储在 MD5(MSB)和 MD4 寄存器中，无论是何种类型的除法。然而，读 MD5 寄存器必须是最后一个操作，以指示已完成了整个除法。

11.3 移位和规范化

移位和规范化操作需要在特定的或者不确定的一段时间将 32 位寄存器(MD3 到 MD0、 MSB 在 MD3 中)旋转。

在移位操作中，32 位无符号整数向左或向右移动一个指定的位数。方向和移动数载 ARCON 寄存器中指定。移位操作需要 3 到 18 个 CPU 周期，这取决于移位时间。

在规范化操作中，32 位无符号整数反复将左移，直到最高有效位(MD3 寄存器的第 7 位)为 1。规范化操作需要 4 至 19 个 CPU 周期，这取决于实际移位时间。

移位和规范化操作都是由写寄存器的一个特定顺序来开始：MD0、MD1、MD2、MD3、最后 ARCON 寄存器。结果会存在 MD0 至 MD3 寄存器中，应按此顺序读取：MD0、MD1、MD2 和 MD3。

11.4 与 Keil C51 配合

因为 Keil C51 同时支持硬件和软件的乘法或除法运算符，C 语言中需要命令行‘#pragma mdu_r515’以使能硬件 MDU 功能，以获得更高的性能。随后，Keil C51 在 MDU 支持下编译数学运算符。

```
#include <SN8F5713.H>
#pragma mdu_r515                  //Keil C51 MDU command line
```

11.5 错误标志（MDEF)

“MDEF”错误标志指执行不当的操作（一个算术运算被一个新的运算重启或者打断)。错误标志机制，第一次写“MD0”操作时自动使能，在第三阶段由“MD3”(乘法或移位/规范化)或“MD5”(除法)的最终读取指令禁用。

以下情况，需设置错误标志：

- MDU 操作的两个阶段(重启或计算中断)期间，写访问‘MDx’寄存器(MD0 至 MD5 和 ARCON 之一）。
- MDU 操作的两个阶段期间，错误标志机制开启时，读访问 MDx 寄存器之一。在这种条件下，设置错误标志，但计算不受打扰。
- 读访问“ARCON”寄存器后，才重置错误标志。错误标志是只读的。

11.6 溢出标志（MDOV)

发生以下情况时，设置 MDOV 溢出标志：

- 除以 0
- 大于 0000 FFFFh 乘法的结果
- 如果设置了 MD3 的最高有效位(MD3.7 = 1)，开始规范化
- MDU 任何不符合上述条件的操作将清零溢出标志。注意溢出标志是完全由硬件控制。不能写。

11.7 MDU 寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
MD0	MD07	MD06	MD05	MD04	MD03	MD02	MD01	MD00
MD1	MD17	MD16	MD15	MD14	MD13	MD12	MD11	MD10
MD2	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20
MD3	MD37	MD36	MD35	MD34	MD33	MD32	MD31	MD30
MD4	MD47	MD46	MD45	MD44	MD43	MD42	MD41	MD40
MD5	MD57	MD56	MD55	MD54	MD53	MD52	MD51	MD50
ARCON	MDEF	MDOV	SLR	SC4	SC3	SC2	SC1	SC0

MD 寄存器(MD0 – MD5: 0xE9 – 0xEE)

Bit	Field	Type	Initial	说明
7..0	MD[7:0]	R/W	0x00	乘除法寄存器。

ARCON 寄存器(0xEF)

Bit	Field	Type	Initial	说明
7	MDEF	R/W	0	MDU 错误标志 MDEF; 指示执行不当的操作（一个算术运算被一个新的运算重启或者打断）。
6	MDOV	R/W	0	MDU 溢出标志; MDU 操作中发生了溢出。
5	SLR	R/W	0	移位方向。 0：左移操作; 1：右移操作。
4..0	SC[4:0]	R/W	0x00	移位计数器。 写 0x00：执行规范化。操作后，可读取实际移位时间; 写其它值：指定移位操作的时间。

12 GPIO

该单片机共有 21 个双向 GPIO 引脚，和 8051 只有开路输出不同，SN8F5713 还内置推挽输出结构，以增强其驱动能力。

12.1 输入输出控制

输入输出方向控制可通过 P0M 到 P2M 寄存器配置。这些位指定各个引脚为输入或输出模式。

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0M	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M
P1M	P17M	P16M	P15M	P14M	P13M	P12M	P11M	P10M
P2M	-	-	P25M	P24M	P23M	P22M	P21M	P20M
P0OC	-	-	-	-	P17OC	P16OC	P03OC	P02OC

P0M Register (0xF9)

Bit	Field	Type	Initial	说明
7	P07M	R/W	0	P0.7 的模式选择位。 0：输入模式； 1：输出模式。
6	P06M	R/W	0	P0.6 的模式选择位。 0：输入模式； 1：输出模式。
5	P05M	R/W	0	P0.5 的模式选择位。 0：输入模式； 1：输出模式。
4	P04M	R/W	0	P0.4 的模式选择位。 0：输入模式； 1：输出模式。
3	P03M	R/W	0	P0.3 的模式选择位。 0：输入模式； 1：输出模式。
2	P02M	R/W	0	P0.2 的模式选择位。 0：输入模式； 1：输出模式。
1	P01M	R/W	0	P0.1 的模式选择位。 0：输入模式； 1：输出模式。
0	P00M	R/W	0	P0.0 的模式选择位。 0：输入模式； 1：输出模式。

P1M Register (0xFA)

Bit	Field	Type	Initial	说明
7	P17M	R/W	0	P1.7 的模式选择位。 0：输入模式； 1：输出模式。
6	P16M	R/W	0	P1.6 的模式选择位。 0：输入模式； 1：输出模式。
5	P15M	R/W	0	P1.5 的模式选择位。 0：输入模式； 1：输出模式。
4	P14M	R/W	0	P1.4 的模式选择位。 0：输入模式； 1：输出模式。
3	P13M	R/W	0	P1.3 的模式选择位。 0：输入模式； 1：输出模式。
2	P12M	R/W	0	P1.2 的模式选择位。 0：输入模式； 1：输出模式。
1	P11M	R/W	0	P1.1 的模式选择位。 0：输入模式； 1：输出模式。
0	P10M	R/W	0	P1.0 的模式选择位。 0：输入模式； 1：输出模式。

P2M Register (0xFB)

Bit	Field	Type	Initial	说明
7..6	保留	R	0	
5	P25M	R/W	0	P2.5 的模式选择位。 0：输入模式； 1：输出模式。
4	P24M	R/W	0	P2.4 的模式选择位。 0：输入模式； 1：输出模式。
3	P23M	R/W	0	P2.3 的模式选择位。 0：输入模式； 1：输出模式。
2	P22M	R/W	0	P2.2 的模式选择位。 0：输入模式； 1：输出模式。
1	P21M	R/W	0	P2.1 的模式选择位。 0：输入模式； 1：输出模式。
0	P20M	R/W	0	P2.0 的模式选择位。 0：输入模式； 1：输出模式。

* *注：低功耗模式下，用户需要正确设置MCU中未封出的引脚，可以降低功耗。强烈的建议将这些引脚设置为输入上拉。*

P0OC Register (0xE4)

Bit	Field	Type	Initial	说明
7..4	保留	R	0	
3	P17OC	R/W	0	P1.7 开漏控制位。 0：禁用； 1：使能。
2	P16OC	R/W	0	P1.6 开漏控制位。 0：禁用； 1：使能。
1	P03OC	R/W	0	P0.3 开漏控制位。 0：禁用； 1：使能。
0	P02OC	R/W	0	P0.2 开漏控制位。 0：禁用； 1：使能。

12.2 输入数据和输出数据

当从 P0~P2 寄存器进行读操作时，当前引脚的逻辑电平将取决于它的外部状态。在某些情况下，当 IO 口在和其他功能复用时，例如 UART 或 I2C，读操作依然可行。

写给 P0~P2 寄存器的值将被马上锁存, 然而, 只有在 P0M ~P2M 被设置为输出模式之后才会被输出。如果这个引脚已经是输出模式了，任何写到 P0~P2 寄存器的值将会马上输出到这个引脚上。

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0	P07	P06	P05	P04	P03	P02	P01	P00
P1	P17	P16	P15	P14	P13	P12	P11	P10
P2	-	-	P25	P24	P23	P22	P21	P20

P0 Register (0x80)

Bit	Field	Type	Initial	说明
7	P07	R/W	1	读：P0.7 为逻辑低电平； 写入 1/0：输出高电平/低电平（P07M=1 时使能）。
6	P06	R/W	1	读：P0.6 为逻辑低电平； 写入 1/0：输出高电平/低电平（P06M=1 时使能）。
5	P05	R/W	1	读：P0.5 为逻辑低电平； 写入 1/0：输出高电平/低电平（P05M=1 时使能）。
4	P04	R/W	1	读：P0.4 为逻辑低电平； 写入 1/0：输出高电平/低电平（P04M=1 时使能）。
3	P03	R/W	1	读：P0.3 为逻辑低电平； 写入 1/0：输出高电平/低电平（P03M=1 时使能）。
2	P02	R/W	1	读：P0.2 为逻辑低电平； 写入 1/0：输出高电平/低电平（P02M=1 时使能）。
1	P01	R/W	1	读：P0.1 为逻辑低电平； 写入 1/0：输出高电平/低电平（P01M=1 时使能）。
0	P00	R/W	1	读：P0.0 为逻辑低电平； 写入 1/0：输出高电平/低电平（P00M=1 时使能）。

P1 Register (0x90)

Bit	Field	Type	Initial	说明
7	P17	R/W	1	读：P1.7 为逻辑低电平； 写入 1/0：输出高电平/低电平（P17M=1 时使能）。
6	P16	R/W	1	读：P1.6 为逻辑低电平； 写入 1/0：输出高电平/低电平（P16M=1 时使能）。
5	P15	R/W	1	读：P1.5 为逻辑低电平； 写入 1/0：输出高电平/低电平（P15M=1 时使能）。
4	P14	R/W	1	读：P1.4 为逻辑低电平； 写入 1/0：输出高电平/低电平（P14M=1 时使能）。
3	P13	R/W	1	读：P1.3 为逻辑低电平； 写入 1/0：输出高电平/低电平（P13M=1 时使能）。
2	P12	R/W	1	读：P1.2 为逻辑低电平； 写入 1/0：输出高电平/低电平（P12M=1 时使能）。
1	P11	R/W	1	读：P1.1 为逻辑低电平； 写入 1/0：输出高电平/低电平（P11M=1 时使能）。
0	P10	R/W	1	读：P1.0 为逻辑低电平； 写入 1/0：输出高电平/低电平（P10M=1 时使能）。

P2 Register (0xA0)

Bit	Field	Type	Initial	说明
7..6	保留	R	0	
5	P25	R/W	1	读：P2.5 为逻辑低电平； 写入 1/0：输出高电平/低电平（P25M=1 时使能）。
4	P24	R/W	1	读：P2.4 为逻辑低电平； 写入 1/0：输出高电平/低电平（P24M=1 时使能）。
3	P23	R/W	1	读：P2.3 为逻辑低电平； 写入 1/0：输出高电平/低电平（P23M=1 时使能）。
2	P22	R/W	1	读：P2.2 为逻辑低电平； 写入 1/0：输出高电平/低电平（P22M=1 时使能）。
1	P21	R/W	1	读：P2.1 为逻辑低电平； 写入 1/0：输出高电平/低电平（P21M=1 时使能）。
0	P20	R/W	1	读：P2.0 为逻辑低电平； 写入 1/0：输出高电平/低电平（P20M=1 时使能）。

12.3 内置上拉寄存器

P0UR 至 P2UR 寄存器管理每个引脚的内部 100KΩ（典型值）的上拉电阻。

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0UR	P07UR	P06UR	P05UR	P04UR	P03UR	P02UR	P01UR	P00UR
P1UR	P17UR	P16UR	P15UR	P14UR	P13UR	P12UR	P11UR	P10UR
P2UR	-	-	P25UR	P24UR	P23UR	P22UR	P21UR	P20UR

P0UR Register (0xF1)

Bit	Field	Type	Initial	说明
7	P07UR	R/W	0	P0.7 的内置上拉电阻控制位。 0：禁用*； 1：使能。
6	P06UR	R/W	0	P0.6 的内置上拉电阻控制位。 0：禁用*； 1：使能。
5	P05UR	R/W	0	P0.5 的内置上拉电阻控制位。 0：禁用*； 1：使能。
4	P04UR	R/W	0	P0.4 的内置上拉电阻控制位。 0：禁用*； 1：使能。
3	P03UR	R/W	0	P0.3 的内置上拉电阻控制位。 0：禁用*； 1：使能。
2	P02UR	R/W	0	P0.2 的内置上拉电阻控制位。 0：禁用*； 1：使能。
1	P01UR	R/W	0	P0.1 的内置上拉电阻控制位。 0：禁用*； 1：使能。
0	P00UR	R/W	0	P0.0 的内置上拉电阻控制位。 0：禁用*； 1：使能。

* 如果引脚为输出模式或者模拟功能，建议禁止上拉电阻。

P1UR Register (0xF2)

Bit	Field	Type	Initial	说明
7	P17UR	R/W	0	P1.7 的内置上拉电阻控制位。 0：禁用*； 1：使能。
6	P16UR	R/W	0	P1.6 的内置上拉电阻控制位。 0：禁用*； 1：使能。
5	P15UR	R/W	0	P1.5 的内置上拉电阻控制位。 0：禁用*； 1：使能。
4	P14UR	R/W	0	P1.4 的内置上拉电阻控制位。 0：禁用*； 1：使能。
3	P13UR	R/W	0	P1.3 的内置上拉电阻控制位。 0：禁用*； 1：使能。
2	P12UR	R/W	0	P1.2 的内置上拉电阻控制位。 0：禁用*； 1：使能。
1	P11UR	R/W	0	P1.1 的内置上拉电阻控制位。 0：禁用*； 1：使能。
0	P10UR	R/W	0	P1.0 的内置上拉电阻控制位。 0：禁用*； 1：使能。

* 如果引脚为输出模式或者模拟功能，建议禁止上拉电阻。

P2UR Register (0xF3)

Bit	Field	Type	Initial	说明
7..6	保留	R	0	
5	P25UR	R/W	0	P2.5 的内置上拉电阻控制位。 0：禁用*； 1：使能。
4	P24UR	R/W	0	P2.4 的内置上拉电阻控制位。 0：禁用*； 1：使能。
3	P23UR	R/W	0	P2.3 的内置上拉电阻控制位。 0：禁用*； 1：使能。
2	P22UR	R/W	0	P2.2 的内置上拉电阻控制位。 0：禁用*； 1：使能。
1	P21UR	R/W	0	P2.1 的内置上拉电阻控制位。 0：禁用*； 1：使能。
0	P20UR	R/W	0	P2.0 的内置上拉电阻控制位。 0：禁用*； 1：使能。

* 如果引脚为输出模式或者模拟功能，建议禁止上拉电阻。

12.4 与模拟功能共有的引脚

该单片机内置模拟功能，如 AD 和 LCD。若使能引脚的共用模拟功能，强烈建议关闭输入通道的施密特触发。

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0CON	P0CON7	P0CON6	-	-	-	-	-	-
P1CON	P1CON7	P1CON6	P1CON5	P1CON4	P1CON3	P1CON2	P1CON1	P1CON0
P2CON	-	-	P2CON5	P2CON4	P2CON3	P2CON2	P2CON1	P2CON0
SYSMOD	-	-	-	-	-	-	BIASEN	STWK

P0CON Register (0x9E)

Bit	Field	Type	Initial	说明
7	P0CON7	R/W	0x00	P0.7 配置控制位*。 0：P0.7 可以是模拟输入或数字 GPIO 引脚； 1：P0.7 是纯模拟输入引脚，不作为数字 GPIO 引脚。
6	P0CON6	R/W	0x00	P0.6 配置控制位*。 0：P0.6 可以是模拟输入或数字 GPIO 引脚； 1：P0.6 是纯模拟输入引脚，不作为数字 GPIO 引脚。
5..0	保留	R	0x00	

* P0CON [7:0]将相关的 Port0 引脚配置为纯模拟输入引脚，以避免电流泄漏。

P1CON 寄存器 (0x9F)

Bit	Field	Type	Initial	说明
7	P1CON7	R/W	0x00	P1.7 配置控制位*。 0：P1.7 可以是模拟输入或数字 GPIO 引脚； 1：P1.7 是纯模拟输入引脚，不作为数字 GPIO 引脚。
6	P1CON6	R/W	0x00	P1.6 配置控制位*。 0：P1.6 可以是模拟输入或数字 GPIO 引脚； 1：P1.6 是纯模拟输入引脚，不作为数字 GPIO 引脚。
5	P1CON5	R/W	0x00	P1.5 配置控制位*。 0：P1.5 可以是模拟输入或数字 GPIO 引脚； 1：P1.5 是纯模拟输入引脚，不作为数字 GPIO 引脚。
4	P1CON4	R/W	0x00	P1.4 配置控制位*。 0：P1.4 可以是模拟输入或数字 GPIO 引脚； 1：P1.4 是纯模拟输入引脚，不作为数字 GPIO 引脚。
3	P1CON3	R/W	0x00	P1.3 配置控制位*。 0：P1.3 可以是模拟输入或数字 GPIO 引脚； 1：P1.3 是纯模拟输入引脚，不作为数字 GPIO 引脚。
2	P1CON2	R/W	0x00	P1.2 配置控制位*。 0：P1.2 可以是模拟输入或数字 GPIO 引脚； 1：P1.2 是纯模拟输入引脚，不作为数字 GPIO 引脚。
1	P1CON1	R/W	0x00	P1.1 配置控制位*。 0：P1.1 可以是模拟输入或数字 GPIO 引脚； 1：P1.1 是纯模拟输入引脚，不作为数字 GPIO 引脚。
0	P1CON0	R/W	0x00	P1.0 配置控制位*。 0：P1.0 可以是模拟输入或数字 GPIO 引脚； 1：P1.0 是纯模拟输入引脚，不作为数字 GPIO 引脚。

* P1CON [7:0] 将相关的 Port1 引脚配置为纯模拟输入引脚，以避免电流泄漏。

P2CON Register (0XD6)

Bit	Field	Type	Initial	说明
7..6	保留	R	0x00	
5	P2CON5	R/W	0x0	P2.5 配置控制位*。 0：P2.5 可以是模拟输入或数字 GPIO 引脚； 1：P2.5 是纯模拟输入引脚，不作为数字 GPIO 引脚。
4	P2CON4	R/W	0x0	P2.4 配置控制位*。 0：P2.4 可以是模拟输入或数字 GPIO 引脚； 1：P2.4 是纯模拟输入引脚，不作为数字 GPIO 引脚。
3	P2CON3	R/W	0x0	P2.3 配置控制位*。 0：P2.3 可以是模拟输入或数字 GPIO 引脚； 1：P2.3 是纯模拟输入引脚，不作为数字 GPIO 引脚。
2	P2CON2	R/W	0x0	P2.2 配置控制位*。 0：P2.2 可以是模拟输入或数字 GPIO 引脚； 1：P2.2 是纯模拟输入引脚，不作为数字 GPIO 引脚。
1	P2CON1	R/W	0x0	P2.1 配置控制位*。 0：P2.1 可以是模拟输入或数字 GPIO 引脚； 1：P2.1 是纯模拟输入引脚，不作为数字 GPIO 引脚。
0	P2CON0	R/W	0x0	P2.0 配置控制位*。 0：P2.0 可以是模拟输入或数字 GPIO 引脚； 1：P2.0 是纯模拟输入引脚，不作为数字 GPIO 引脚。

* P2CON [5:0] 将相关的 Port2 引脚配置为纯模拟输入引脚，以避免电流泄漏。

SYSMOD Register (0xCE)

Bit	Field	Type	Initial	说明
7..2	保留	R	0x00	
1	BIASEN	R/W	0	0：禁用 1/2*VDD 偏压功能； 1：使能 1/2*VDD 偏压功能。
0				参考其它章节。

P0BIAS Register (0XF4)

Bit	Field	Type	Initial	说明
7..4	保留	R	0x00	
3	P0BIAS3	R/W	0x0	P0.3 1/2*VDD 偏压控制位。 0：禁用 1/2*VDD 偏压输出； 1：使能 1/2*VDD 偏压输出。
2	P0BIAS2	R/W	0x0	P0.2 1/2*VDD 偏压控制位。 0：禁用 1/2*VDD 偏压输出； 1：使能 1/2*VDD 偏压输出。
1	P0BIAS1	R/W	0x0	P0.1 1/2*VDD 偏压控制位。 0：禁用 1/2*VDD 偏压输出； 1：使能 1/2*VDD 偏压输出。
0	P0BIAS0	R/W	0x0	P0.0 1/2*VDD 偏压控制位。 0：禁用 1/2*VDD 偏压输出； 1：使能 1/2*VDD 偏压输出。

P1BIAS Register (0XF5)

Bit	Field	Type	Initial	说明
7..4	保留	R	0x00	
3	P1BIAS3	R/W	0x0	P1.3 1/2*VDD 偏压控制位。 0：禁用 1/2*VDD 偏压输出； 1：使能 1/2*VDD 偏压输出。
2	P1BIAS2	R/W	0x0	P1.2 1/2*VDD 偏压控制位。 0：禁用 1/2*VDD 偏压输出； 1：使能 1/2*VDD 偏压输出。
1	P1BIAS1	R/W	0x0	P1.1 1/2*VDD 偏压控制位。 0：禁用 1/2*VDD 偏压输出； 1：使能 1/2*VDD 偏压输出。
0	P1BIAS0	R/W	0x0	P1.0 1/2*VDD 偏压控制位。 0：禁用 1/2*VDD 偏压输出； 1：使能 1/2*VDD 偏压输出。

13外部中断

外部中断源 INT0，INT1 和 INT2 内置边沿触发功能，由 PEDGE 寄存器控制。使能外部中断（EX0/EX1/EX2）和全局中断（EAL）后，发生边沿触发事件时，外部中断请求标志位（IE0/IE1/IE2）置1，程序计数器跳转到中断向量（ORG 0x0003/0x0043/0x0083）并执行中断服务程序。在执行 ISR 之前由硬件清中断请求标志。

13.1外部中断寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEDGE	-	-	EX2G1	EX2G0	EX1G1	EX1G0	EX0G1	EX0G0
IEN0	EAL	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
IEN2	-	-	-	-	EPWM1	EX2	-	EADC
IRCON2	-	-	-	-	PWM1F	IE2	-	ADCF

PEDGE 寄存器(0X8F)

Bit	Field	Type	Initial	说明
5..4	EX2G[1:0]	R/W	10	外部中断 INT2 触发沿控制位。 00：保留； 01：上升沿； 10：下降沿（默认）； 11：上升/下降沿。
3..2	EX1G[1:0]	R/W	10	外部中断 INT1 触发沿控制位。 00：保留； 01：上升沿； 10：下降沿（默认）； 11：上升/下降沿。
1..0	EX0G[1:0]	R/W	10	外部中断 INT0 触发沿控制位。 00：保留； 01：上升沿； 10：下降沿（默认）； 11：上升/下降沿。
Else	Reserved	R	0	

IEN0 寄存器 (0XA8)

Bit	Field	Type	Initial	说明
7	EAL	R/W	0	使能所有中断控制位。 0：禁用所有中断功能； 1：使能所有中断功能。
2	EX1	R/W	0	外部 P0.6 中断(INT1) 控制位。 0：禁用 INT1 中断功能； 1：使能 INT1 中断功能。
0	EX0	R/W	0	外部 P0.3 中断(INT0)控制位。 0：禁用 INT0 中断功能； 1：使能 INT0 中断功能。
Else				参考其它章节。

TCON 寄存器 (0X88)

Bit	Field	Type	Initial	说明
3	IE1	R/W	0	外部 P0.6 中断（INT1）请求标志位。 0：无 INT1 中断请求； 1：INT1 请求中断。
1	IE0	R/W	0	外部 P0.3 中断（INT0）请求标志位。 0：无 INT0 中断请求； 1：INT0 请求中断。
Else				参考其它章节。

IEN2 寄存器 (0X9A)

Bit	Field	Type	Initial	说明
2	EX2	R/W	0	外部 P0.7 中断(INT2)控制位。 0：禁用 INT2 中断功能； 1：使能 INT2 中断功能。
Else				参考其它章节。

IRCON2 Register (0XBF)

Bit	Field	Type	Initial	说明
2	IE2	R/W	0	外部 P0.7 中断(INT2)请求标志位。 0：禁用 INT2 中断请求； 1：使能 INT2 中断请求。
Else				参考其它章节。

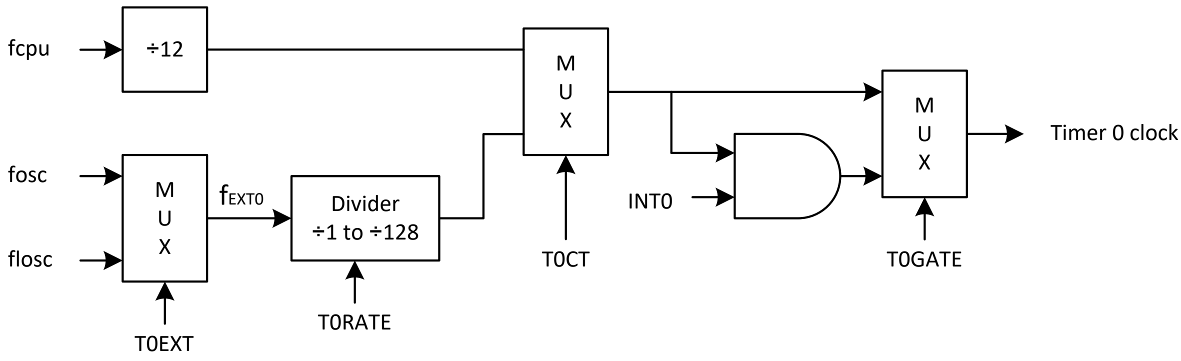
14 定时器 0 和定时器 1

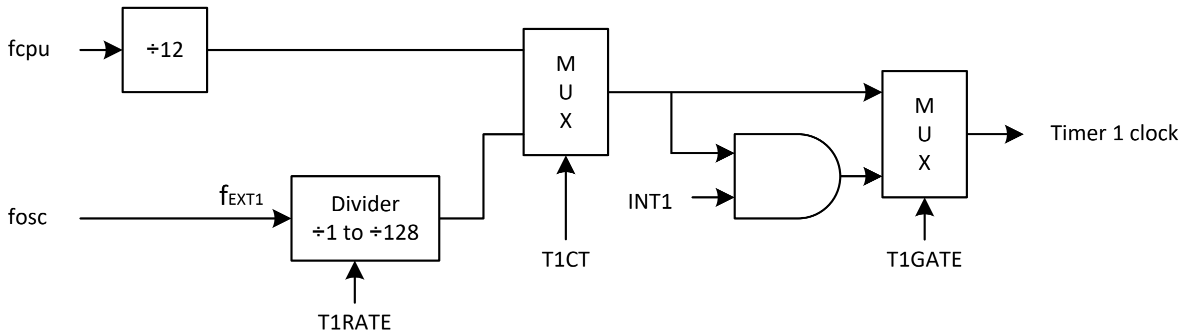
T0 和 T1 是 2 个独立的二进制定时器。T0 共有 4 种不同的的操作模式：模式 0：13 位向上计数定时器；模式 1：16 位向上计数定时器；模式 2：8 位向上计数寄存器，并指定重装值；模式 3：独立的 2 个 8 位向上计数定时器。而 T1 只有 T0 的模式 0 和模式 2 两种操作模式。T0 和 T1 分别支持 ET0 和 ET1 中断。

当 T0 时钟源为 Flosc 且 STWK=1 时，T0 可在 STOP 模式下工作，由 T0 中断将系统从 STOP 模式中唤醒。

14.1 T0 和 T1 时钟选择

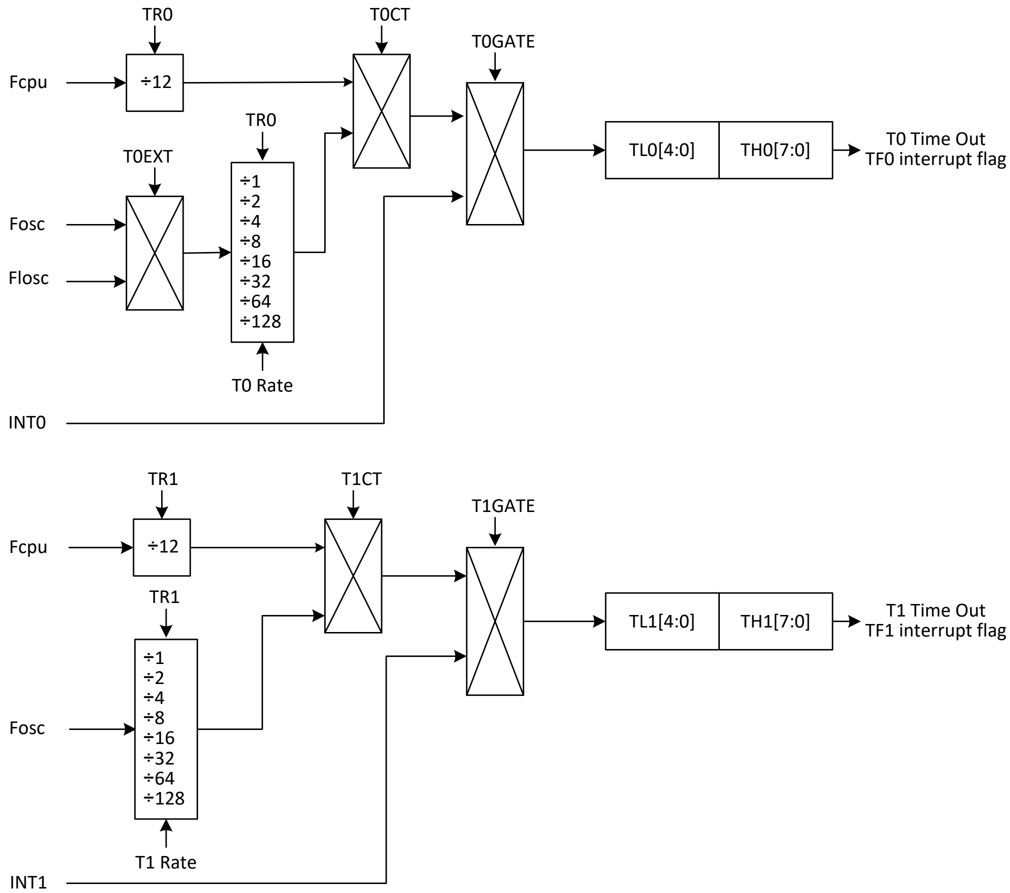
下图阐明了 T0 和 T1 的时钟选择电路。T0 有 3 组时钟源：Fcpu、Fpsc 和 Flosc，若使用 T0GATE，由 INT0 引脚来关闭（暂停）时钟源。而 T1 有 2 组时钟源：Fcpu 和 Fosc，若使用 T1GATE，由 INT1 引脚来关闭（暂停）时钟源。总的来说，T0 和 T1 这两个计数器最主要的区别在于 T0 支持 Flosc 时钟源（低速时钟）。





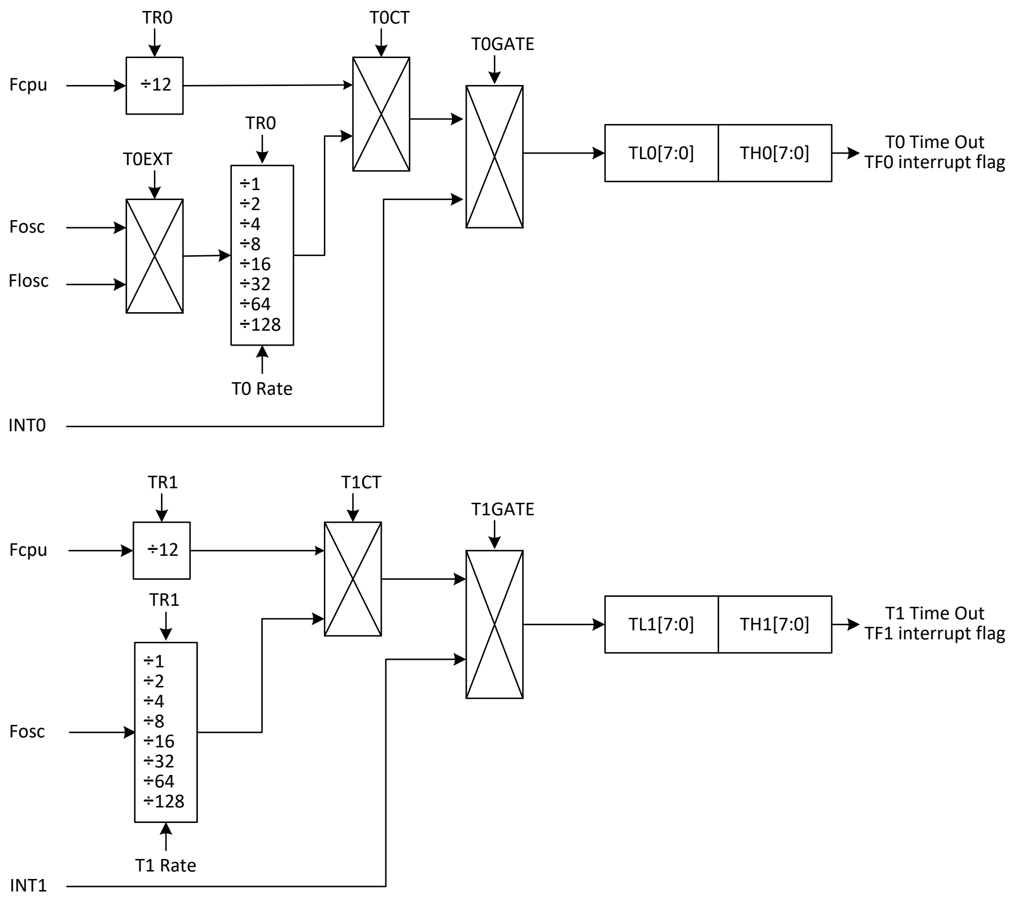
14.2 模式 0：13 位向上定时器

模式 0 是 13 位向上计数定时器（TL0/TL1 的高 3bit 无效）。一旦定时计数器溢出（由 0xFF1F 到 0x0000），会立即置位 TF0/TF1 标志。没有使能 ET0/ET1 时，由软件读取该标志；使能 ET0/ET1 时，则由中断控制器进行控制。



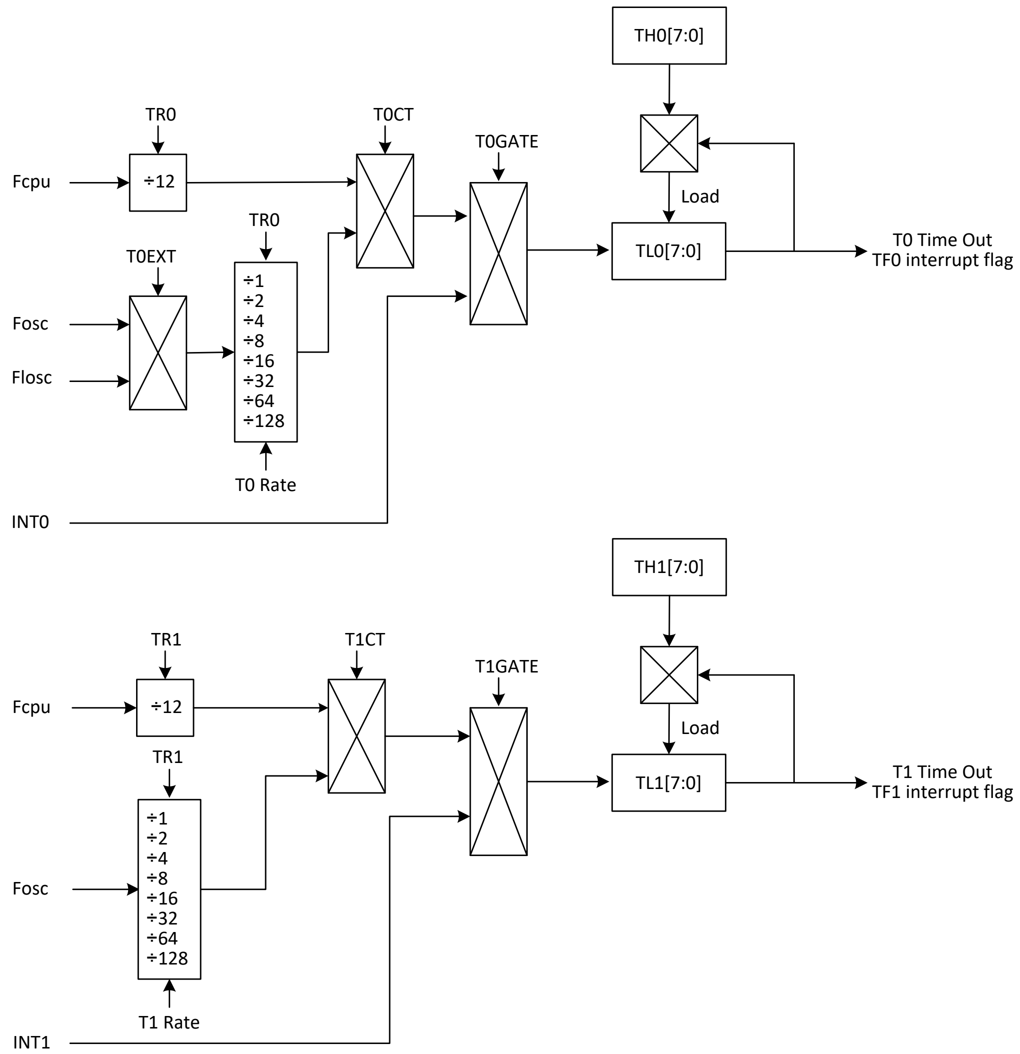
14.3 模式 1：16 位向上计数定时器

模式 1 是 16 位向上计数定时器。一旦定时器的计数器溢出（从 0xFFFF 到 0x0000），会释放 TF0/TF1，由软件读取标志；使能 ET0/ET1 时，则由中断控制器进行控制。



14.4 模式 2：8 位向上计数定时器（指定重装值）

模式 2 是 8 位向上计数定时器（TL0/TL1），指定重装值。计数器溢出时（从 0FFH 到 00H）释放 TF0/TF1 标志给固件或中断控制器；同时定时器复制 TH0/TH1 的值给 TL0/TL1 寄存器。因此，定时器实际上是从 0FFH 计数到 TH0/TH1 的值。

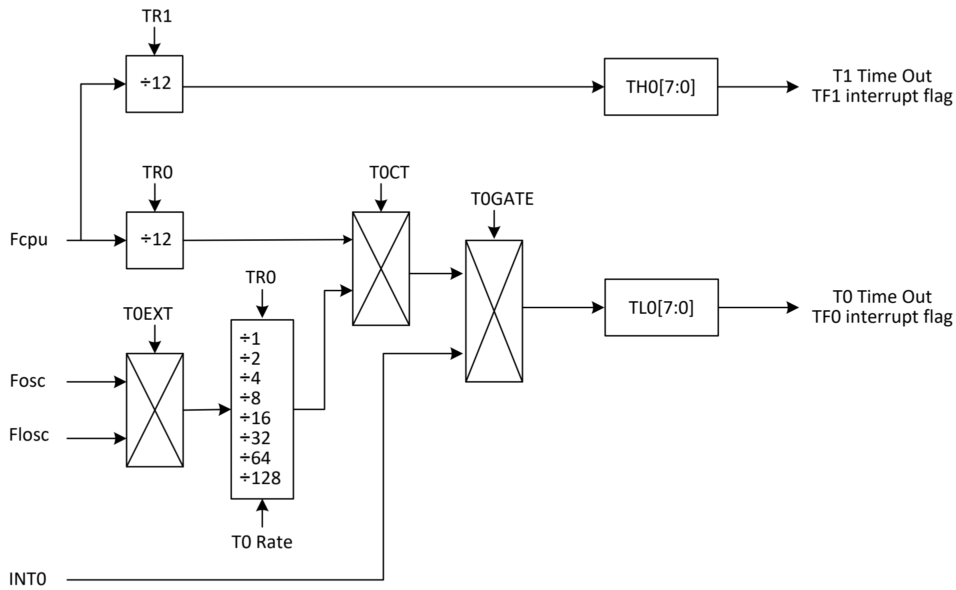


14.5 模式 3（仅 T 0）：独立的 2 个 8 位向上计数定时器

模式 3 是把 TH0 和 TL0 看作独立的 2 个 8 位定时器。8 位向上计数定时器 TL0 支持 RTC，有 2 种时钟源选择（Fcpu 和 Fosc）；而 TH0 的时钟源固定为 Fcpu/12。当 T0GATE 设置为 1 时，定时器的使能和停止可以由 INT0 引脚控制。

在该模式下，由 TR0 使能 TL0 计数器，TL0 溢出后 TF0 置 1。TH0 计数器则由 TR1 控制，TH0 溢出后 TF1 置 1。

在该模式下，T1 不会发生溢出事件，可以看作是没有溢出标志产生计数器。



14.6 T0 和 T1 寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IEN0	EAL	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
TCON	TF1	TR1	TF0	TR0	IE1	-	IE0	-
TCON0	T0EXT	T0RATE2	T0RATE1	T0RATE0	-	T1RATE2	T1RATE1	T1RATE0
TMOD	T1GATE	T1CT	T1M1	T1M0	T0GATE	T0CT	T0M1	T0M0
TH0	TH07	TH06	TH05	TH04	TH03	TH02	TH01	TH00
TL0	TL07	TL06	TL05	TL04	TL03	TL02	TL01	TL00
TH1	TH17	TH16	TH15	TH14	TH13	TH12	TH11	TH10
TL1	TL17	TL16	TL15	TL14	TL13	TL12	TL11	TL10

IEN0 寄存器(0xA8)

Bit	Field	Type	Initial	说明
7	EAL	R/W	0	中断使能位，参考中断章节。
3	ET1	R/W	0	T1 中断。 0：禁用； 1：使能。
1	ET0	R/W	0	T0 中断。 0：禁用； 1：使能。
Else				参考其它章节。

TH0 / TH1 寄存器 (TH0: 0x8C, TH1: 0x8D)

Bit	Field	Type	Initial	说明
7..0	TH0/TH1	R/W	0x00	T0 和 T1 计数器的高字节。

TL0 / TL1 寄存器 (TL0: 0x8A, TL1: 0x8B)

Bit	Field	Type	Initial	说明
7..0	TL0/TL1	R/W	0x00	T0 和 T1 计数器的低字节。

TCON 寄存器 (0x88)

Bit	Field	Type	Initial	说明
7	TF1	R/W	0	T1 溢出事件显示位。 0：T1 没有溢出； 1：T1 溢出。 该位由中断助力器自动清零，或者由固件手动清零。
6	TR1	R/W	0	T1 功能控制位。 0：禁止； 1：使能。
5	TF0	R/W	0	T0 溢出事件显示位。 0：T0 没有溢出； 1：T0 溢出。 该位由中断助力器自动清零，或者由固件手动清零。
4	TR0	R/W	0	T0 功能控制位。 0：禁止； 1：使能。
3	IE1	R/W	0	参考 INT1。
2	Reserved	R	0	
1	IE0	R/W	0	参考 INT0。
0	Reserved	R	0	

TCON0 寄存器(0xE7)

Bit	Field	Type	Initial	说明
7	T0EXT	R/W	0	T0 f EXT0 时钟源选择位。 0：fosc； 1：flosc。
6..4	T0RATE[2:0]	R/W	000	T0 外部时钟源的时钟分频位。 000：F_{EXT0} / 128； 001：F_{EXT0} / 64； 010：F_{EXT0} / 32； 011：F_{EXT0} / 16； 100：F_{EXT0} / 8； 101：F_{EXT0} / 4； 110：F_{EXT0} / 2； 111：F_{EXT0} / 1。
3	Reserved	R	0	
2..0	T1RATE[2:0]	R/W	000	T1 外部时钟源的时钟分频位。 000：F_{EXT1} / 128； 001：F_{EXT1} / 64； 010：F_{EXT1} / 32； 011：F_{EXT1} / 16； 100：F_{EXT1} / 8； 101：F_{EXT1} / 4； 110：F_{EXT1} / 2； 111：F_{EXT1} / 1。

TMOD 寄存器(0x89)

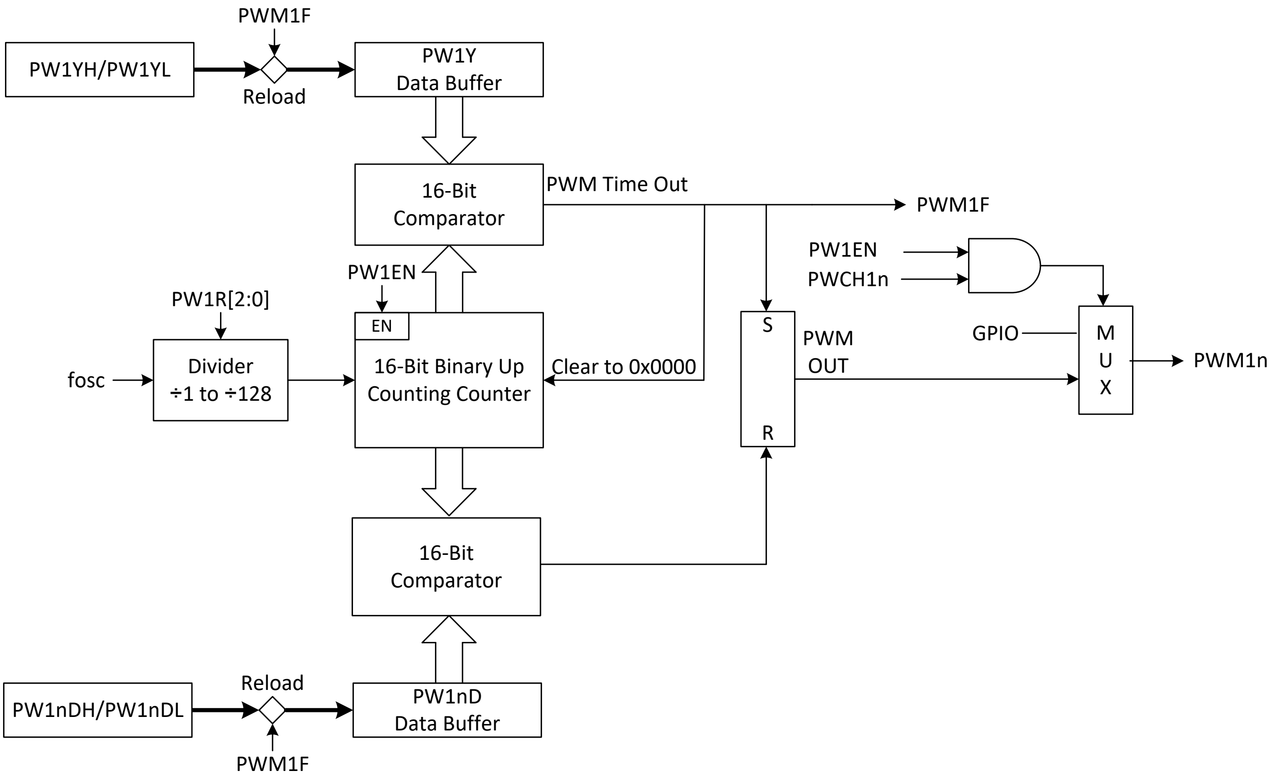
Bit	Field	Type	Initial	说明
7	T1GATE	R/W	0	T1 gate 控制模式。 0：禁用； 1：使能，T1 时钟源的停止由 INT1 控制。
6	T1CT	R/W	0	T1 时钟源选择。 0：FTimer1 = Fcpu / 12； 1：FTimer 1 = F_{EXT1} / T1RATE (参考 T1RATE) *(1)。
5..4	T1M[1:0]	R/W	00	T1 操作模式控制位。 00：13 位向上计数定时器； 01：16 位向上计数定时器； 10：8 位向上计数定时器，支持重装； 11：保留。
3	T0GATE	R/W	0	T0 gate 控制模式。 0：禁用； 1：使能，T0 时钟源的停止由 INT0 控制。
2	T0CT	R/W	0	T0 时钟源选择。 0：FTimer0 = fcpu / 12； 1：FTimer0 = F_{EXT0} / T0RATE (参考 T0RATE) *(2)。
1..0	T0M[1:0]	R/W	00	T0 操作模式。 00：13 位向上计数定时器； 01：16 位向上计数定时器； 10：8 位向上计数定时器，支持重装； 11：独立的 2 个 8 位向上计数定时器。

*(1) fEXT1 = fosc。
*(2) fEXT0 = fosc 或 flosc。

15 PWM

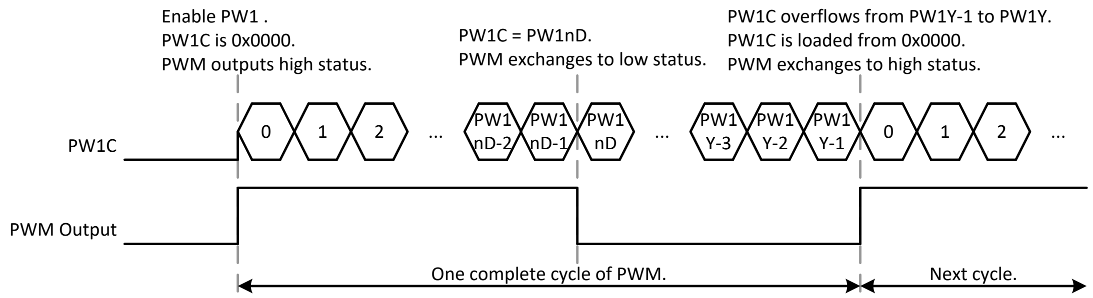
PW1 定时器是 16 位向上计数定时器，支持 8 通道通用 PWM 功能。计数器计数到上限值（PW1Y）后，计数器清零并触发一个中断信号。PWM 的占空比周期由寄存器 PW10D~PW17D 控制。每个 PWM 通道都有各自的占空比控制。

PWM 共有 8 个可编程控制的 PWM 通道，与 GPIO 引脚共用，由 PWCH[7:0]位控制。通过使能 PWCH[7:0]位的对应位/通道执行输出操作。使能的 PWM 通道从 GPIO 切换到 PWM 输出。禁止 PWCH[7:0]位时，PWM 通道返回到 GPIO 模式的上一个状态。若使能中断，PW1 定时器内置 IDEL 模式唤醒功能。定时器溢出时（从 PW1Y-1 到 PW1Y 计数），立即释放 PWM1F，其能通过 FW 进行读/写。由 EPWM1 控制 PW1 中断功能。



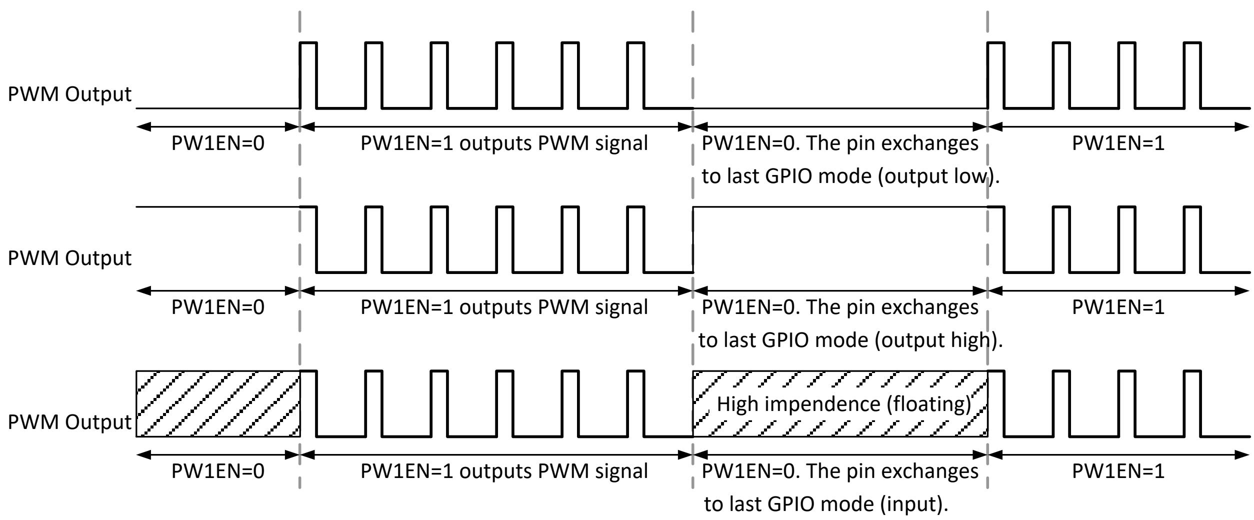
15.1 通用 PWM

PW1 定时器内置 PWM 功能，由 PW1EN 和 PWCH 寄存器控制。PWM10 到 PWM17 为输出引脚，并与 GPIO 引脚共用，由 PWCH[7:0]位控制。输出 PWM 时，必须设置 PW1EN=1。PWM 输出信号同步完成后，PWM 通道从 GPIO 模式切换到 PWM 输出。PW1EN=0 时，PWM 通道返回 GPIO 模式和上一个状态。PW1Y 和 PW1nD 进行比较，其比较结果产生 PWM 信号。当 PW1C 从 0x0000 计数时，PWM 输出高状态，也就是 PWM 初始状态，PW1C 从 PW1Y 寄存器加载新数据，以决定 PWM 的周期和分辨率。PW1C 继续计数，系统比较 PW1C 和 PW1nD 的值，PW1C=PW1nD 时，PWM 输出状态变为低电平，PW1C 继续计数。PW1 定时器溢出（PW1Y-1 到 0x0000）后，PWM 完成一个信号周期。PW1C 自动重新加载 0x0000，并且输出状态变为高电平，以等待下一个周期。PW1nD 决定高电平占空比时间，PW1Y 决定 PWM 的分辨率和周期。PW1nD 的值不能大于 PW1Y 的值，否则 PWM 信号会出错。PWM 时钟源是 fosc，PW1RATE[2:0] bits: 000 = fosc/128，001 = fosc/64，010 = fosc/32，011 = fosc/16，100 = fosc/8，101 = fosc/4，110 = fosc/2，111 = fosc/1。



PWM 周期= PW1Y

PWM 占空比= (PW1nD)： (PW1nD - PW1Y)



15.2 PWM 寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PW1M	PW1EN	PW1R2	PW1R1	PW1R0	-	-	-	-
PWCH	PWCH17	PWCH16	PWCH15	PWCH14	PWCH13	PWCH12	PWCH11	PWCH10
PW1YH	PW1Y15	PW1Y14	PW1Y13	PW1Y12	PW1Y11	PW1Y10	PW1Y9	PW1Y8
PW1YL	PW1Y7	PW1Y6	PW1Y5	PW1Y4	PW1Y3	PW1Y2	PW1Y1	PW1Y0
PW10DH	PW10D15	PW10D14	PW10D13	PW10D12	PW10D11	PW10D10	PW10D9	PW10D8
PW10DL	PW10D7	PW10D6	PW10D5	PW10D4	PW10D3	PW10D2	PW10D1	PW10D0
PW11DH	PW11D15	PW11D14	PW11D13	PW11D12	PW11D11	PW11D10	PW11D9	PW11D8
PW11DL	PW11D7	PW11D6	PW11D5	PW11D4	PW11D3	PW11D2	PW11D1	PW11D0
PW12DH	PW12D15	PW12D14	PW12D13	PW12D12	PW12D11	PW12D10	PW12D9	PW12D8
PW12DL	PW12D7	PW12D6	PW12D5	PW12D4	PW12D3	PW12D2	PW12D1	PW12D0
PW13DH	PW13D15	PW13D14	PW13D13	PW13D12	PW13D11	PW13D10	PW13D9	PW13D8
PW13DL	PW13D7	PW13D6	PW13D5	PW13D4	PW13D3	PW13D2	PW13D1	PW13D0
PW14DH	PW14D15	PW14D14	PW14D13	PW14D12	PW14D11	PW14D10	PW14D9	PW14D8
PW14DL	PW14D7	PW14D6	PW14D5	PW14D4	PW14D3	PW14D2	PW14D1	PW14D0
PW15DH	PW15D15	PW15D14	PW15D13	PW15D12	PW15D11	PW15D10	PW15D9	PW15D8
PW15DL	PW15D7	PW15D6	PW15D5	PW15D4	PW15D3	PW15D2	PW15D1	PW15D0
PW16DH	PW16D15	PW16D14	PW16D13	PW16D12	PW16D11	PW16D10	PW16D9	PW16D8
PW16DL	PW16D7	PW16D6	PW16D5	PW16D4	PW16D3	PW16D2	PW16D1	PW16D0
PW17DH	PW17D15	PW17D14	PW17D13	PW17D12	PW17D11	PW17D10	PW17D9	PW17D8
PW17DL	PW17D7	PW17D6	PW17D5	PW17D4	PW17D3	PW17D2	PW17D1	PW17D0
IEN0	EAL	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
IEN2	-	-	-	-	EPWM1	EX2	-	EADC
IRCON2	-	-	-	-	PWM1F	IE2	-	ADCF

PWCH 寄存器 (0xB6)

Bit	Field	Type	Initial	说明
7	PWCH17	R/W	0	PWM1 共用引脚控制位。 0：GPIO； 1：PWM 输出（与 P1.1 共用）。
6	PWCH16	R/W	0	PWM1 共用引脚控制位。 0：GPIO； 1：PWM 输出（与 P1.0 共用）。
5	PWCH15	R/W	0	PWM1 共用引脚控制位。 0：GPIO； 1：PWM 输出（与 P0.1 共用）。
4	PWCH14	R/W	0	PWM1 共用引脚控制位。 0：GPIO； 1：PWM 输出（与 P0.0 共用）。
3	PWCH13	R/W	0	PWM1 共用引脚控制位。 0：GPIO； 1：PWM 输出（与 P2.3 共用）。
2	PWCH12	R/W	0	PWM1 共用引脚控制位。 0：GPIO； 1：PWM 输出（与 P2.2 共用）。
1	PWCH11	R/W	0	PWM1 共用引脚控制位。 0：GPIO； 1：PWM 输出（与 P2.1 共用）。
0	PWCH10	R/W	0	PWM1 共用引脚控制位。 0：GPIO； 1：PWM 输出（与 P2.0 共用）。

PW1M Registers (PW1M: 0xA1)

Bit	Field	Type	Initial	说明
7	PW1EN	R/W	0	PW1 功能 0：禁用； 1：使能*。
6..4	PW1R[2:0]	R/W	000	PWM 定时器时钟源 000：fosc / 128； 001：fosc / 64； 010：fosc / 32； 011：fosc / 16； 100：fosc / 8； 101：fosc / 4； 110：fosc / 2； 111：fosc / 1。
3..0	Reserved	R	0	

* 当该周期的值设为 0x0000 时，使能 PWM 后，PWM 停止工作，且该周期不能更新。

PW1YH/PW1YL 寄存器(PW1YH: 0xA3, PW1YL: 0xA2)

Bit	Field	Type	Initial	说明
7..0	PW1YH/L	R/W	0x00	16 位 PWM1 周期控制位*。

* 必须在开始 PWM 功能之前完成对该周期配置的设置。

PW10DH/PW10DL 寄存器 (PW10DH: 0xA5, PW10DL: 0xA4)

Bit	Field	Type	Initial	说明
7..0	PW10DH/L	R/W	0x00	16 位 PWM1 占空比控制位。

PW11DH/PW11DL 寄存器(PW11DH: 0xA7, PW11DL: 0xA6)

Bit	Field	Type	Initial	说明
7..0	PW11DH/L	R/W	0x00	16 位 PWM1 占空比控制位。

PW12DH/PW12DL 寄存器 (PW12DH: 0xAC, PW12DL: 0xAB)

Bit	Field	Type	Initial	说明
7..0	PW12DH/L	R/W	0x00	16 位 PWM1 占空比控制位。

PW13DH/PW13DL 寄存器(PW13DH: 0xAE, PW13DL: 0xAD)

Bit	Field	Type	Initial	说明
7..0	PW13DH/L	R/W	0x00	16 位 PWM1 占空比控制位。

PW14DH/PW14DL 寄存器(PW14DH: 0xBC, PW14DL: 0xBB)

Bit	Field	Type	Initial	说明
7..0	PW14DH/L	R/W	0x00	16 位 PWM1 占空比控制位。

PW15DH/PW15DL 寄存器(PW15DH: 0xBE, PW15DL: 0xBD)

Bit	Field	Type	Initial	说明
7..0	PW15DH/L	R/W	0x00	16 位 PWM1 占空比控制位。

PW16DH/PW16DL 寄存器(PW16DH: 0xCA, PW16DL: 0XC9)

Bit	Field	Type	Initial	说明
7..0	PW16DH/L	R/W	0x00	16 位 PWM1 占空比控制位。

PW17DH/PW17DL 寄存器 (PW17DH: 0xCC, PW17DL: 0xCB)

Bit	Field	Type	Initial	说明
7..0	PW17DH/L	R/W	0x00	16 位 PWM1 占空比控制位。

IEN0 寄存器 (0xA8)

Bit	Field	Type	Initial	说明
7	EAL	R/W	0	中断使能，参考中断章节。
Else				参考其它章节。

IEN2 寄存器 (0X9A)

Bit	Field	Type	Initial	说明
3	EPWM1	R/W	0	PWM1 中断控制位。 0 = 禁用 PWM1 中断功能。 1 = 使能 PWM1 中断功能。
Else				参考其它章节。

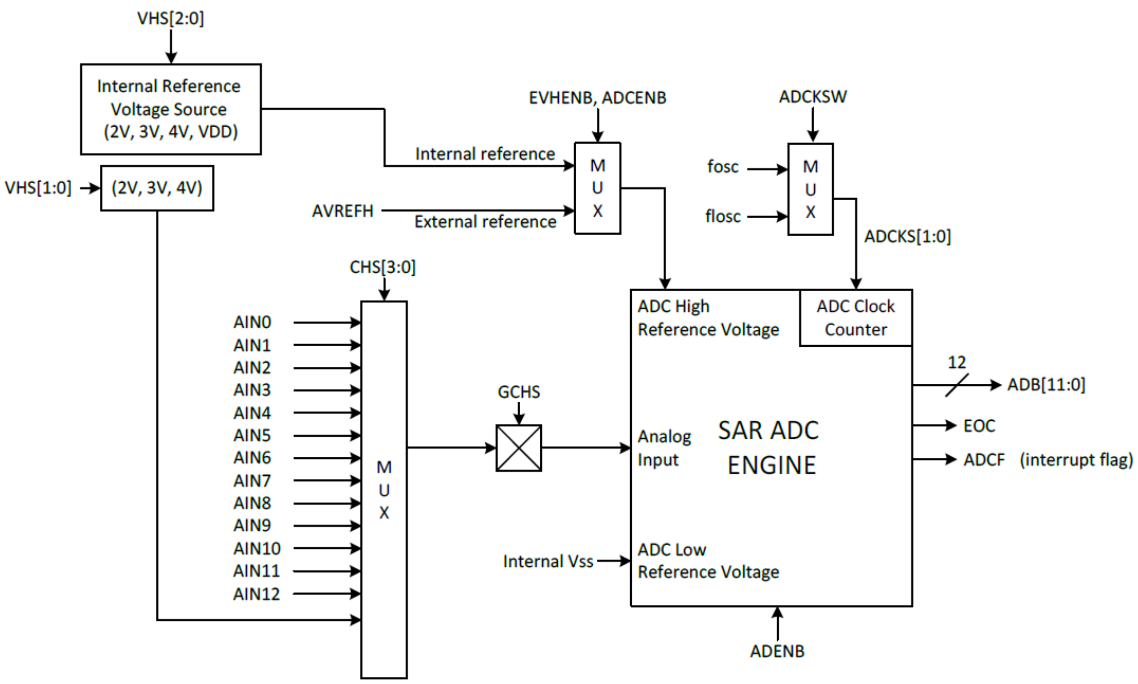
IRCON2 寄存器 (0XBF)

Bit	Field	Type	Initial	说明
3	PWM1F	R/W	0	PWM1 中断请求标志位。 0：无 PMW1 中断请求； 1：PWM1 请求中断。
Else				参考其他章节

16 ADC

模拟数字转换（ADC）是一个 SAR 结构，内置 13 个输入通道（AIN0~AIN12），高达 4096 阶的分辨率，能将一个模拟信号转换成相应的 12 位数字信号。ADC 内置的 13 个模拟通道可以测量 13 种不同的模拟信号源，ADC 的分辨率为 12 位。ADC 共有 4 种时钟 rate 来决定 ADC 的转换速率。ADC 参考高电压包括 5 种。

4 种内部电压源（Vdd、4V、3V 和 2V），另一个是外部参考源输入引脚（由 AVREFH 引脚提供）。ADC 内置 P1CON/P2CON 寄存器来设置模拟输入引脚。设置好 ADENB 和 ADS 位后，ADC 开始转换模拟信号为数字信号。除 ADS 位可以开始转换模拟信号外，PW1EN 也可以转换模拟信号。ADC 可在 IDLE 模式下工作，ADC 结束后，若使能中断，则将系统从 IDLE 模式下唤醒进入 NORMAL 模式。ADC 时钟源为 Flosc 且 STWK=1 时，ADC 可在 STOP 模式工作，由 ADC 中断将系统从 STOP 模式唤醒。



16.1 操作配置

ADC 开始转换前必须先设置好下列配置，具体如下：

1. 选择和使能 ADC 输入通道（由 CHS[4:0]位和 GCHS 位设置）；
2. 将 ADC 输入通道设置为输入模式（由 PnM 寄存器设置）；
3. 必须禁止 ADC 输入通道的内部上拉电阻（由 PnUR 寄存器设置）；
4. 必须将 ADC 模拟输入通道的控制位设置为 1（由 PnCON 寄存器设置）；
5. 选择 ADC 高参考电压和低参考电压（由 VREFH 寄存器设置）；
6. 选择 ADC 时钟源和时钟频率（由 ADCKSW 和 ADCKS[1:0]位设置）；
7. 设置 ADENB 位后，ADC 准备开始转换。

由 ADENB 位使能 ADC IP 功能时，要保证由程序启动 ADC。写入 1 到 ADM 寄存器的 ADS 位。设置 ADENB 和 ADS 位后，ADC 开始转换模拟信号为数字信号。转换结束后，ADS 位复位为逻辑 0。转换结束后，ADC 电路将 EOC 和 ADCF 置 1，并将数字数据输出到 ADB 和 ADR 寄存器中。若使能 ADC 中断（EADC=1），ADC 请求中断，ADC 转换完成后，ADCF=1 时执行中断服务程序。中断时必须由硬件将 ADCF 清零。

16.2 ADC 输入通道

ADC 内置 13 个输入通道（AIN0~AIN12），用于测量 13 种不同的模拟信号源，由 CHS[4:0]和 GCHS 位控制。AIN13 通道保留。AIN14 则是内部 2V/3V/4V 的输入通道。外部没有输入引脚，ADC 参考电压必须为内部 VDD 或外部电压，不能是内部 2V/3V/4V。在电池应用中，AIN14 可作为很好的电池检测器。选择一个合适的 AVREFH 值和比较值，系统可内置一个高性能、廉价的电池检测器。

CHS[4:0]	Channel	Pin name	Remark
00000	AIN0	P1.0	
00001	AIN1	P1.1	
00010	AIN2	P1.2	
00011	AIN3	P1.3	
00100	AIN4	P1.4	
00101	AIN5	P1.5	
00110	AIN6	P1.6	
00111	AIN7	P1.7	
01000	AIN8	P2.0	
01001	AIN9	P2.1	
01010	AIN10	P2.2	
01011	AIN11	P2.3	
01100	AIN12	P2.4	
01101	AIN13	-	保留
01110	AIN14	Internal 2V or 3V or 4V	电池检测通道
01111~11111	-	-	保留

16.2.1 引脚配置

ADC 输入通道引脚与 P1 和 P2 引脚共用。ADC 通道选择由 CHS[4:0]控制。同时，只能设置 P1 和 P2 中的一个引脚为 ADC 输入。P1 和 P2 中配置为 ADC 输入通道的引脚必须设置为输入模式，禁止内部上拉电阻，并先由程序使能 P1CON/P2CON。通过 CHS[4:0]选择 ADC 输入通道后，GCHS 位设置为 1，以使能 ADC。

ADC 输入引脚与数字 I/O 引脚共用，当输入一个模拟信号到 CMOS 数字输入引脚时，尤其当模拟信号电平为 1/2 VDD 时，可能产生额外的漏电流。掉电模式下，上述漏电流是个大问题。不幸的是，如果用户将多个模拟输入信号输入到 P1、P2，会造成上述漏电流的情况。将 1 写入 PnCON 寄存器，使其对应的引脚被设为纯模拟信号输入引脚，从而避免上述漏电流的产生。

注意：ADC 引脚为 GPIO 引脚时，P1CON /P2CON 位必须置 0，否则数字 I/O 引脚的信号会被隔离。

16.3 参考电压

ADC 内置 5 种高参考电压，由 VREFH 寄存器控制：包括 1 个外部参考电压和 4 个内部参考源（VDD、4V、3V、2V）。EVHENB = 1 时，ADC 参考电压由外部参考源提供（AVREFH/P1.0），此时必须设置 P1.0 为输入模式并禁止上拉电阻。

EVHENB = 0 时，ADC 参考电压由内部参考源提供，并由 VHS[1:0]选择控制。VHS[1:0] = 11 时，ADC 参考源选择 VDD；VHS[1:0] = 10 时，ADC 参考源选择 4V； VHS[1:0] = 01 时，ADC 参考源选择 3V；VHS[1:0] = 00 时，ADC 参考源选择 2V。外部参考源的限制条件为，最高为 VDD，最低为内部最低电平，否则默认为 VDD。若选择 AIN14 为 2V/3V/4V 的输入通道，而不从外部输入，这样 ADC 的高参考电压必须是内部 VDD 或者外部参考电源，不是内部 2V/3V/4V。

16.3.1 信号格式

ADC 采样电压范围为参考电压高/低电平之间。ADC 参考低电压为 VSS，高电压包括 VDD/4V/3V/2V 和外部参考电压（由 P3.0/AVREFH 引脚提供），由 EVHENB 控制。ADC 参考电压的范围为：（ADC 参考高电压-ADC 参考低电压）≧ 2V。ADC 参考低压为 VSS=0V，故 ADC 参考高电压范围为 2V~VDD。外部参考电压需在此范围之内。

- ADC 内部参考低电压=0V。
- ADC 内部参考高电压=VDD/4V/3V/2V。（EVHENB = 0）
- ADC 外部参考高电压= 2V~VDD。（EVHENB = 1）

ADC 采样输入信号电压必须在 ADC 参考低电压和 ADC 参考高电压之间，若 ADC 输入信号的电压不在此范围内，则 ADC 的转换结果会出错（满量程或者为 0）。

- ADC 参考低电压≦ ADC 采样输入信号电压≦ ADC 参考高电压

16.4 转换时间

ADC 转换时间是指从 ADS=1（开始 ADC）到 EOC=1（ADC 结束）所用的时间，由 ADC 时钟频率控制，12 位 ADC 的转换时间为 1/(ADC 时钟/4)*16 S。ADC 的时钟源有 2 种：Fosc 和 Flosc，由 ADCKSW 位控制。ADCKS[1:0] 位: 00 = fosc/16 或 flosc/16, 01 = fosc/8 或 flosc/8, 10 = fosc/1 或 flosc/1, 11 = fosc/2 或 flosc/2。

ADC 的转换时间会影响 ADC 的性能，如果输入高频率的模拟信号，必须要选择一个高频率的 ADC 转换时钟。如果 ADC 的转换时间比模拟信号的转换频率慢，则 ADC 的结果出错。故选择合适的 ADC 时钟频率和 ADC 分辨率才能得到合适的 ADC 转换频率。

$$12\text{ 位 ADC 转换时间} = \frac{16}{\text{ADC 时钟率}/4}$$

ADCKSW=0 时：

ADCKS[1:0]	ADC时钟率	fosc = 16MHz		fosc = 32MHz	
		转换时间	转换率	转换时间	转换率
00	fosc/16	1/(16MHz/16/4)*16 = 64us	15.625kHz	1/(32MHz/16/4)*16 = 32us	31.25kHz
01	fosc/8	1/(16MHz/8/4)*16 = 32us	31.25kHz	1/(32MHz/8/4)*16 = 16us	62.5kHz
10	fosc	1/(16MHz/4)*16 = 4us	250kHz	1/(32MHz/4)*16 = 2us	500kHz
11	fosc/2	1/(16MHz/2/4)*16 = 8us	125kHz	1/(32MHz/2/4)*16 = 4us	250kHz

ACKSW=1 时：

ADCKS[1:0]	ADC 时钟率	flosc = 16KHz		flosc = 32.768KHz	
		转换时间	转换率	转换时间	转换率
00	flosc/16	1/(16KHz/16/4)*16 = 64ms	15.625Hz	1/(32.768KHz/16/4)*16 = 31.25ms	32Hz
01	flosc/8	1/(16KHz/8/4)*16 = 32ms	31.25Hz	1/(32.768KHz/8/4)*16 = 15.625ms	64Hz
10	flosc	1/(16KHz/4)*16 = 4ms	250Hz	1/(32.768KHz/4)*16 = 1.953ms	512Hz
11	flosc/2	1/(16KHz/2/4)*16 = 8ms	125Hz	1/(32.768KHz /2/4)*16 = 3.906ms	256Hz

16.5 数据缓存器

ADC 数据缓存器共 12 位，用来存储 AD 转换结果，8 位只读寄存器 ADB 存放结果的高字节（bit4~bit11），ADR（ADR[3:0]）存放低字节（bit0~bit3）。ADC 数据缓存器是只读寄存器，系统复位后处于未知状态。

表 16-1 AIN 输入电压与 ADB 输出数据

AIN n	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4	ADB3	ADB2	ADB1	ADB0
0/4096*VREFH	0	0	0	0	0	0	0	0	0	0	0	0
1/4096*VREFH	0	0	0	0	0	0	0	0	0	0	0	1
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
4094/4096*VREFH	1	1	1	1	1	1	1	1	1	1	1	0
4095/4096*VREFH	1	1	1	1	1	1	1	1	1	1	1	1

16.6 ADC 寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADM	ADENB	ADS	EOC	CHS4	CHS3	CHS2	CHS1	CHS0
ADB	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4
ADR	ADCKSW	GCHS	ADCKS1	ADCKS0	ADB3	ADB2	ADB1	ADB0
VREFH	EVHENB	-	-	-	-	VHS2	VHS1	VHS0
P1CON	P1CON7	P1CON6	P1CON5	P1CON4	P1CON3	P1CON2	P1CON1	P1CON0
P2CON	-	-	P2CON5	P2CON4	P2CON3	P2CON2	P2CON1	P2CON0
IEN0	EAL	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
IEN2	-	-	-	-	EPWM1	EX2	-	EADC
IRCON2	-	-	-	-	PWM1F	IE2	-	ADCF

ADM 寄存器(0xD2)

Bit	Field	Type	Initial	说明
7	ADENB	R/W	0	ADC 控制位，STOP 模式下，禁止 ADC 以省电。 0：禁止； 1：使能。
6	ADS	R/W	0	ADC 转换控制位。 写 1：开始 AD 转换（转换结束后自动清零）。
5	EOC	R/W	0	ADC 状态位。 0：AD 转换过程中； 1：AD 转换结束（硬件自动设置，FW 手动清零）。
4..0	CHS[4:0]	R/W	0x00	ADC 输入通道选择位。 00000：AIN0，00001：AIN1, 00010：AIN2，00011：AIN3, 00100：AIN4，00101：AIN5, 00110：AIN6，00111：AIN7, 01000：AIN8，01001：AIN9, 01010：AIN10，01011：AIN11, 01100：AIN12，01101：保留, 01110：AIN14*(1)，其它：保留。

*(1) AIN14 为内部 2V/3V/4V 输入通道，没有外部输入引脚，ADC 参考电压必须为内部 VDD 或外部电压，不能是内部 2V/3V/4V。

ADB 寄存器(0xD3)

Bit	Field	Type	Initial	说明
7..0	ADB[11:4]	R	-	12 位 ADC 分辨率模式下的 ADC 结果位[11:4]*

* ADC 数据缓存器的长度为 12 位，用于存储 AD 转换结果，其高字节存入 ADB 寄存器，低字节存入 ADR[3:0]位。

ADR 寄存器 (0xD4)

Bit	Field	Type	Initial	说明
7	ADCKSW	R/W	0	ADC 时钟源选择位。 0：fosc； 1：flosc。
6	GCHS	R/W	0	ADC 全局通道控制位。 0：禁用 AIN 通道； 1：使能 AIN 通道。
5..4	ADCKS[1:0]	R/W	00	ADC 的时钟率选择位。 00 = fosc/16，01 = fosc/8，10 = fosc/1，11 = fosc/2 or 00 = flosc/16，01 = flosc/8，10 = flosc/1，11 = flosc/2。
3..0	ADB[3:0]	R	-	12 位 ADC 分辨率模式下的 ADC 结果位[3:0]*。

* ADC 数据缓存器的长度为 12 位，用于存储 AD 转换结果，其高字节存入 ADB 寄存器，低字节存入 ADR[3:0]位。

VREFH 寄存器 (0xD5)

Bit	Field	Type	Initial	说明
7	EVHENB	R/W	0	ADC 内部参考高电压控制位。 0：使能 ADC 内部 VREFH 功能，AVREFH/P1.0 为 GPIO 引脚； 1：禁止 ADC 内部 VREFH 功能，AVREFH/P1.0 为外部 AVREFH*(1)输入引脚。
2..0	VHS[2:0]	R/W	00	ADC 内部参考高压选择位*(2)。 000：VREFH = 2.0V； 001：VREFH = 3.0V； 010：VREFH = 4.0V； 011：VREFH = VDD； 100：VREFH = VDD and AIN14 = 2.0V； 101：VREFH = VDD and AIN14 = 3.0V； 110：VREFH = VDD and AIN14 = 4.0V； 其它：保留。
Else	Reserved	R/W	0	

*(1) AVREFH 的电压值必须在 VDD 到 2.0V 范围内。

*(2) AIN14 为内部 2V/3V/4V 输入通道，没有外部输入引脚，ADC 参考电压必须为内部 VDD 或外部电压，不能是内部 2V/3V/4V。

P1CON 寄存器 (0x9F)

Bit	Field	Type	Initial	说明
7..0	P1CON[7:0]	R/W	0x00	P1 配置控制位*。 0：P1 可作为模拟输入引脚（ADC 输入引脚）或者数字 GPIO 引脚； 1：P1 只能作为模拟输入引脚，不能作为数字 GPIO 引脚。

* P1CON [7:0] 配置相关的 P1 引脚为纯模拟输入引脚以避免漏电流。

P2CON Register (0XD6)

Bit	Field	Type	Initial	说明
7..0	P2CON[7:0]	R/W	0x0	P2 配置控制位*。 0：P2 可作为模拟输入引脚（ADC 输入引脚）或者数字 GPIO 引脚； 1：P2 只能作为模拟输入引脚，不能作为数字 GPIO 引脚。

* P2CON [7:0] 配置相关的 P2 引脚为纯模拟输入引脚以避免漏电流。

IEN0 Register (0xA8)

Bit	Field	Type	Initial	说明
7	EAL	R/W	0	中断使能。参考中断章节。
Else				参考其它章节。

IEN2 Register (0x9A)

Bit	Field	Type	Initial	说明
0	EADC	R/W	0	ADC 中断控制位。 0：禁用 ADC 中断功能； 1：使能 ADC 中断功能。
Else				参考其它章节。

IRCON2 Register (0xBF)

Bit	Field	Type	Initial	说明
0	ADCF	R/W	0	ADC 中断请求标志位。 0：无 ADC 中断请求； 1：ADC 中断请求。
Else				参考其它章节。

17 UART

UART 是一种灵活的全双工同步异步收发器，这一系列接口提供 1MHz 灵活的全双工传输。UART 共有 4 种操作模式：一种同步，三种异步的。模式 0 是移位寄存器模式，作为同步收发器。模式 1-模式 3 下，UART 是 8 位或 9 位数据的异步收发器。传输格式有起始位、8 位/ 9 位数据和停止位。通过写 S0BUF 寄存器开始传输。接收后，S0BUF 寄存器完成接收后，输入数据可用。TB80/RB80 位可用作 9 位 UART 模式下传输和接收的第九位。可编程的波特率支持不同速度的外围设备。

UART 特征如下：

- 全双工，双线同步/异步数据传输。
- 可编程的波特率。
- 8 位移位寄存器：同步收发器。
- 8 位/ 9 位 UART：异步收发器，带 8 或 9 位数据位和可编程的波特率。

17.1 UART 操作

UART UTX 和 URX 引脚与 GPIO 共用。在同步模式下，UTX/URX 共用引脚必须由软件设置为输出高。在异步模式下(8 位/ 9 位 UART)，UTX 共用引脚必须由软件设置为输出高， URX 设置为输入高。因此，URX/UTX 引脚将转为 UART 用途。禁用 UART 时，UART 引脚返回 GPIO 的上一个状态。

UTX/URX 引脚也支持开漏结构。开漏选项由 PnOC 位控制。PnOC = 0 时，禁用 UTX/URX 开漏结构。PnOC = 1 时，启用 UTX/URX 开漏结构。如果启用开漏结构，UTX/URX 引脚必须设定为高电平（IO 模式控制将被忽略），需要外部上拉电阻。

UART 支持中断功能。EU0TX 和 EU0RX 是 UART 传输中断功能控制位。EU0TX 和 EU0RX 分别控制 UART 的发送和接收功能。EU0TX/EU0RX = 0，禁用收发器的中断功能。EU0TX/EU0RX = 1，启用 UART 收发器的中断功能。UART 中断功能启用时，UART 操作之后，程序计数器指向中断向量执行 UART 中断服务程序。当禁用中断时，TI0/RI0 是 UART 中断请求标志，也是 UART 操作状态指示符。由软件 TI0 和 RI0 必须由软件清零。

UART 提供四个操作模式（一个同步和三个异步），均由 S0CON 寄存器控制。这些模式可以支持不同的波特率和通信协议。

<table>
<tr><th>SM0</th><th>SM1</th><th>模式</th><th>同步</th><th>时钟率</th><th>开始位</th><th>数据位</th><th>停止位</th><th>UART 引脚的模式和数据</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>同步</td><td>Fcpu/12</td><td>X</td><td>8</td><td>X</td><td>UTX 引脚：
P03M=1 和 P03=1
P16M=1 和 P16=1
URX 引脚：
发送器：
P02M=1 和 P02=1
P17M=1 和 P17=1
接收器：
P02M=0 和 P02=1
P17M=0 和 P17=1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>异步</td><td>波特率生成器或 T1 溢出率</td><td>1</td><td>8</td><td>1</td><td rowspan="3">UTX 引脚：
P03M=1 和 P03=1
P16M=1 和 P16=1
URX 引脚：
P02M=0
P17M=0</td></tr>
<tr><td>1</td><td>0</td><td>2</td><td>异步</td><td>Fcpu/64 或 Fcpu/32</td><td>1</td><td>9</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>3</td><td>异步</td><td>波特率生成器或 T1 溢出率</td><td>1</td><td>9</td><td>1</td></tr>
</table>

17.2 模式 0：同步 8 位收发器

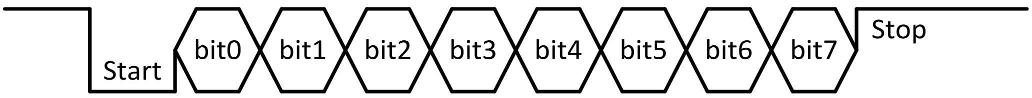
模式 0 是移位寄存器模式，作为同步收发器。发送和接收条件下，UTX 引脚都输出移位时钟用于传输和接收状态。URX 引脚用于传输和接收数据。首先，发送/接收带 LSB 的 8 位数据。波特率为 Fcpu/12。通过写数据到 S0BUF 寄存器开始传输。第 8 位传输结束后，设置 TI0 标志位。通过设置 REN0 位来控制接收，清 RI0 位。REN0=1 和 RI0 由 1 到 0 时，开始传输数据，且在接收完第 8 位时，设置 RI0 标志位。

17.3 模式 1：可变波特率的 8 位收发器

模式 1 支持可变波特率的异步 8 位 UART。传输格式包括 1 位起始位、8 位数据位(LSB 在前)和 1 位停止位。通过 UTX 引脚传输数据，通过 URX 引脚接收数据。波特率时钟源可以是波特率生成器或 BD 位控制的 T1 溢出。BD = 0 时，波特率时钟源来自 T1 溢出。BD = 1 时，波特率时钟源是波特率生成器，由 S0RELH 和 S0RELL 控制。此外，波特率可以通过 SMOD 位翻倍。

由 REN0 位控制数据传输。传输配置之后，加载传输数据到 S0BUF，然后 UART 开始传输数据包。然后，在停止位的开始处设置 TI0 标志位。

由 REN0 位控制数据接收。REN0 = 1 时，数据接收功能启用。对于主终端，通过接收起始位开始数据接收，URX 检测到起始位的下降沿，然后在停止位的中间设置 RI0 标志位。直到接收完成，输入数据存储在 S0BUF 寄存器中，停止位存储在 RB80 中。

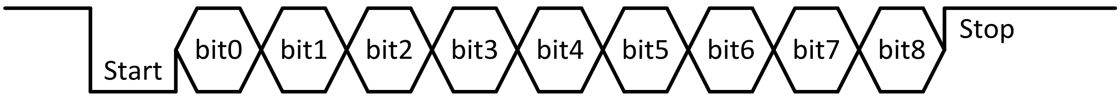


17.4 模式 2：固定波特率的 9 位收发器

模式 2 支持固定波特率的异步 9 位 UART。传输格式包括 1 位起始位、9 位数据位(LSB 在前)和 1 位停止位。通过 UTX 引脚传输数据，通过 URX 引脚接收数据。波特率时钟源可以是波特率生成器或 BD 位控制的 T1 溢出。BD = 0 时，波特率时钟源固定为 fcpu/64 或 fcpu/32，且由 SMOD 位控制。SMOD=0 时，波特率是 fcpu/64。SMOD=1 时，波特率是 fcpu/32。

由 REN0 位控制数据传输。传输配置之后，加载传输数据到 S0BUF，然后 UART 开始传输数据包。从 TB80 获取第 9 位数据位。在停止位的开始处设置 TI0 标志位。

由 REN0 位控制数据接收。REN0 = 1 时，数据接收功能启用。对于主终端，通过接收起始位开始数据接收，URX 检测到起始位的下降沿，然后在停止位的中间设置 RI0 标志位。直到接收完成，输入数据的低 8 位存储在 S0BUF 寄存器中，第 9 位存储在 RB80 中。

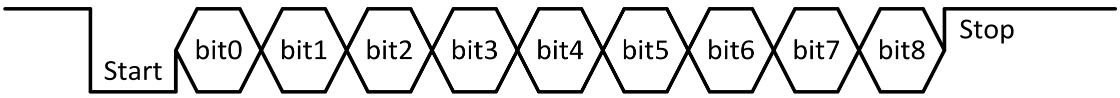


17.5 模式 3：可变波特率的 9 位收发器

模式 3 支持可变波特率的异步 9 位 UART。传输格式包括 1 位起始位、9 位数据位(LSB 在前)和 1 位停止位。通过 UTX 引脚传输数据，通过 URX 引脚接收数据。模式 2 和模式 3 的不同之处在于波特率的选择。在模式 3 条件下，波特率时钟源可以是波特率生成器或 BD 位控制的 T1 溢出。BD = 0 时，波特率时钟源来自 T1 溢出。BD = 1 时，波特率时钟源是波特率生成器，由 S0RELH 和 S0RELL 控制。此外，波特率可以通过 SMOD 位翻倍。

由 REN0 位控制数据传输。传输配置之后，加载传输数据到 S0BUF，然后 UART 开始传输数据包。从 TB80 获取第 9 位数据位。在停止位的开始处设置 TI0 标志位。

由 REN0 位控制数据接收。REN0 = 1 时，数据接收功能启用。对于主终端，通过接收起始位开始数据接收，URX 检测到起始位的下降沿，然后在停止位的中间设置 RI0 标志位。直到接收完成，输入数据存储在 S0BUF 寄存器中，第 9 位存储在 RB80 中。



17.6 多处理器通信

模式 2 和模式 3(9-bit UART)下，UART 支持主设备和一个或多个从设备之间的多处理器通信。主标识符通过使用第 9 位数据位纠正从标志符。通信开始时，主设备发送特定地址字节，且第 9 位设为“1”，以选择的从设备，然后在接下来的传输中发送一个数据字节，且第 9 位设为“0”。

多处理器通信由 SM20 位控制。当 SM20 =0 时，禁用多处理器通信。当 SM20 = 1，启用多处理器通信。如果设置了 SM20，只在第 9 位接收到的位为“1” (RB80) 时生成 UART 接收中断。从设备将通过软件比较接收的数据和已有的地址数据。如果地址字节匹配，在接下来的数据传输中从设备将清零 SM20，以启用中断功能。如果地址不匹配，则 SM20 保持为“1”， 在接下来的数据传输中不会产生中断。

17.7 波特率控制

UART 模式 0 的波特率是固定的，为 Fcpu/12；模式 2 有 2 个波特率选项，由 SMOCD 寄存器选择控制，其选项分别为 FCPU/64（SMOD=0）和 Fcpu/32（SMOD=1）。

UART 模式 1 和模式 3 的波特率由 S0RELH/SORELL 寄存器（BD=1）或者 T1 溢出周期（BD=0）产生。通过设置 SMOD 位可以将波特率增加 1 倍。

在模式 1 和模式 3 中若选择 S0RELH/S0RELL（BD=1），则波特率由下面公式产生：

$$\text{Baud Rate} = 2^{\text{SMOD}} \times \frac{\text{fcpu}}{64 \times (1024 - \text{S0REL})} \text{bps}$$

UART 波特率的设置建议见表 17-1（Fcpu=8MHz）：

Baud Rate	SMOD	S0RELH	S0RELL	Accuracy
4800	0	0x03	0xE6	-0.16 %
9600	0	0x03	0xF3	-0.16 %
19200	1	0x03	0xF3	-0.16 %
38400	1	0x03	0xF9	-6.99 %
56000	1	0x03	0xFB	-10.71 %
57600	1	0x03	0xFC	8.51 %
115200	1	0x03	0xFE	8.51 %
128000	1	0x03	0xFE	-2.34 %
250000	1	0x03	0xFF	0 %

在模式 1 和模式 3 中若选择 T1 溢出周期（BD=0），则波特率由下面公式产生。T1 必须为 8 位自动重装模式，这样才能产生周期性的溢出信号。

$$\text{Baud Rate} = 2^{\text{SMOD}} \times \frac{\text{T1 clock rate}}{32 \times (256 - \text{TH1})} \text{bps}$$

UART 波特率的 T1 溢出周期（T1 时钟=32M）设置建议见表 17-2（Fcpu=8MHz）：

Baud Rate	SMOD	Timer Period	TH1/TL1	Accuracy
4800	0	6.510 us	0x30	0.16 %
9600	1	6.510 us	0x30	0.16 %
19200	1	3.255 us	0x98	0.16 %
38400	1	1.628 us	0xCC	0.16 %
56000	1	1.116 us	0xDC	-0.80 %
57600	1	1.085 us	0xDD	-0.80 %
115200	1	0.543 us	0xEF	2.08 %
128000	1	0.488 us	0xF0	-2.40 %

✸ *注：*

1. 波特率产生器源是 T1 溢出率，最大计数值是 0xFB（只支持 0x00~0xFB）。
2. 波特率产生器源是 T1 溢出率，系统时钟 fcpu 必须大于等于 T1 时钟率。

17.8 省电

UART 模块具有时钟门控功能，用于省电。REN0=0 时，停止 UART 模块内部时钟以省电，UART 相关寄存器（S0CON，S0CON2，S0BUF，SORELL，S0RELH 和 SMOD 位）都停止工作。

反之，REN0=1 时，UART 内部时钟工作，相关寄存器也可以访问。在初始化 UART 之前，REN0 必须设置为 1。

17.9 UART 寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
S0CON	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
S0CON2	BD	-	-	-	-	-	-	URMX
S0BUF	S0BUF7	S0BUF6	S0BUF5	S0BUF4	S0BUF3	S0BUF2	S0BUF1	S0BUF0
PCON	SMOD	-	-	STWK	P2SEL	GF0	STOP	IDLE
S0RELH	-	-	-	-	-	-	S0REL9	S0REL8
S0RELL	S0REL7	S0REL6	S0REL5	S0REL4	S0REL3	S0REL2	S0REL1	R0REL0
IEN0	EAL	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
P0OC	-	-	-	-	P17OC	P16OC	P03OC	P02OC
P0M	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M
P0	P07	P06	P05	P04	P03	P02	P01	P00
P1M	P17M	P16M	P15M	P14M	P13M	P12M	P11M	P10M
P1	P17	P16	P15	P14	P13	P12	P11	P10

S0CON 寄存器(0x98)

Bit	Field	Type	Initial	说明
7..6	SM[0:1]	R/W	00	UART 模式选择位。 00：模式 0； 01：模式 1； 10：模式 2； 11：模式 3。
5	SM20	R/W	0	多重处理器通讯控制位（模式 2、3）。 0：禁止； 1：使能。
4	REN0	R/W	0	UART 模块（接收功能）控制位。 0：禁止以省电*； 1：使能以使 UART 正常工作。
3	TB0	R/W	0	发送数据的第 9 位（模式 2、3）。
2	RB0	R/W	0	接收数据的第 9 位。
1	TI0	R/W	0	发送 UART 的中断标志。
0	RI0	R/W	0	接收 UART 的中断标志。

* REN0=0 时，不能访问 UART 相关寄存器，停止模块内部时钟。

> ✸ *注：中断使能，通过硬件清空 TI0 和 RI0 。*

S0CON2 寄存器 (0xD8)

Bit	Field	Type	Initial	说明
7	BD	R/W	0	波特率发生器选择位（模式 1、3）。 0：T1 溢出周期； 1：由寄存器 S0RELH/S0RELL 控制。
6..1	Reserved	R	0x00	
0	URMX	R/W	0	0 = UART TX 和 RX 引脚是 P03 和 P02； 1 = UART TX 和 RX 引脚是 P16 和 P17。

S0BUF 寄存器(0x99)

Bit	Field	Type	Initial	说明
7..0	S0BUF	R/W	0x00	写入数据的操作触发 UART 通信（首先 LSB），可在数据包结束时读取接收到的数据。

PCON 寄存器(0x87)

Bit	Field	Type	Initial	说明
7	SMOD	R/W	0	UART 波特率控制位（UART 模式 0、2）。 0：Fcpu/64； 1：Fcpu/32。
6..0				请参考其它章节。

S0RELH/S0RELL 寄存器(S0RELH: 0xBA, S0RELL: 0xAA)

Bit	Field	Type	Initial	说明
15..10	Reserved	R	0x00	
9..0	S0REL[9:0]	R/W	0x00	S0RELH[1:0] &S0RELL[7:0]，UART 重装寄存器，用于产生 UART 波特率。

IEN0 寄存器(0xA8)

Bit	Field	Type	Initial	说明
7	EAL	R/W	0	使能中断，请参考中断章节。
5	EU0RX	R/W	0	使能 UART RX 中断。
4	EU0TX	R/W	0	使能 UART TX 中断。
Else				请参考其它章节。

P0OC 寄存器(0xE4)

Bit	Field	Type	Initial	说明
3	P17OC	R/W	0	0：切换 P1.7 (URX)为推拉模式； 1：切换 P1.7 (URX) 为开漏模式。
2	P16OC	R/W	0	0：切换 P1.6 (UTX) 为推拉模式； 1：切换 P1.6 (UTX) 为开漏模式。
1	P03OC	R/W	0	0：切换 P0.3 (UTX) 为推拉模式； 1：切换 P0.3 (UTX) 为开漏模式。
0	P02OC	R/W	0	0：切换 P0.2(UTX) 为推拉模式； 1：切换 P0.2 (UTX) 为开漏模式。

P0M 寄存器(0xF9)

Bit	Field	Type	Initial	说明
3	P03M	R/W	0	~~0：设置 P0.3 (UTX) 为输入模式*；~~ 1：设置 P0.3 (UTX) 为输出模式（要求）。
2	P02M	R/W	0	0：设置 P0.2 (URX) 为输入模式（要求）； ~~1：设置 P0.2 (URX) 为输出模式*。~~
Else				参考其它章节。

* URX 和 UTX 分别要求为输入模式和输出模式，以对应的接收和发送数据。

P0 寄存器(0x80)

Bit	Field	Type	Initial	说明
3	P03	R/W	1	~~0：设置 P0.3 (UTX)一直为低电平*；~~ 1：使 P0.3 (UTX)能输出 UART 数据（要求）。
2	P02	R/W	1	为了监控总线状态，可以读取该位。
Else				参考其它章节。

* 初始化 P03 时一定需要设置为高电平，因为 UART 模块只驱动共用引脚低电平。

P1M 寄存器(0xFA)

Bit	Field	Type	Initial	说明
7	P17M	R/W	0	0：设置 P1.7 (URX) 为输入模式（要求）； ~~1：设置 P1.7 (URX) 为输出模式*。~~
6	P16M	R/W	0	~~0：设置 P1.6 (UTX) 为输入模式*；~~ 1: 设置 P1.6 (UTX) 为输出模式（要求）。
Else				参考其它章节。

* URX 和 UTX 分别要求为输入模式和输出模式，以对应的接收和发送数据。

P1 寄存器(0x90)

Bit	Field	Type	Initial	说明
7	P17	R/W	1	为了监控总线状态，可以读取该位。
6	P16	R/W	1	~~0：设置 P1.6 (UTX) 一直为低电平*；~~ 1：使 P1.6 (UTX) 能输出 UART 数据（要求）。
Else				参考其它章节。

* 初始化 P16 时一定需要设置为高电平，因为 UART 模块只驱动共用引脚低电平。

18 I2C

I2C 是串行通讯接口，在单片机与单片机，或单片机与其它外设之间进行数据传输。I2C 可作为主机或从机进行双向 IO 传输数据，SDA（串行数据输出）和 SCL（串行时钟输出）。

主机发送数据给从机时，叫做“WRITE”操作；从机发送数据给主机时，叫做“READ”操作。I2C 也支持多主机通讯，通过一种仲裁方式来决定哪个主机拥有控制总线以及传输数据的权限，从而保证数据传送的正确性。

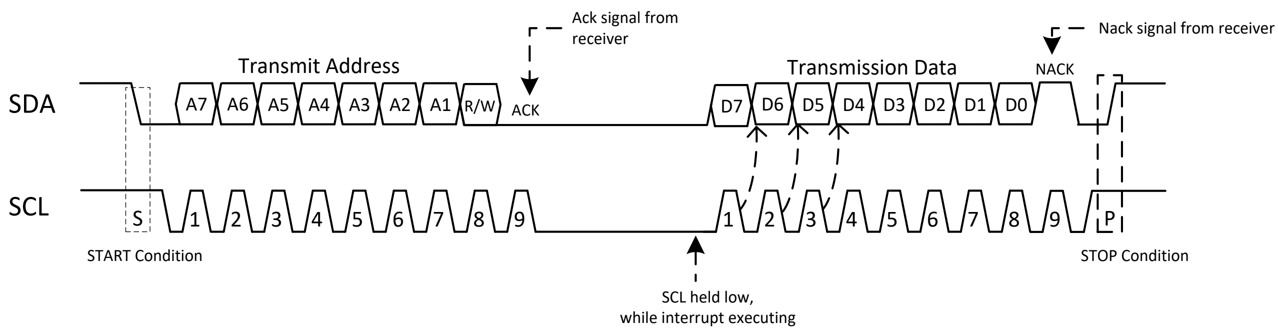
- 主 Tx, Rx 模式
- 从 Tx, Rx 模式（通用地址调用），以在单主机情况下进行 multiplex slave
- 双线同步数据收发器
- 支持 100K/400K 时钟率

18.1 I2C 协议

I2C 发送结构包括 START(S)开始信号，8 位地址字节，一个或多个数据字节，以及 STOP (P)结束信号。开始信号由主机产生，以便对传输进行初始化。

发送的数据为最高有效位（MSB）优先。在地址字节中，高 7 位为地址位，低位为数据方向位（R/W）。R/W=0 时，表示该传输为“WRITE”操作；R/W=1 时，表示该传输为“READ”操作。

接收到每个字节后，接收器（主机或从机）必须发送一个 ACK 信号。若发送器没有接收到 ACK 信号，则会识别为 NACK 信号。在 WRITE 操作中，主机发送数据给从机，然后等待从机返回 ACK 信号。在 READ 操作中，从机发送数据给主机，然后等待主机返回 ACK 信号。最后，主机产生一个 STOP 信号来结束数据传输。

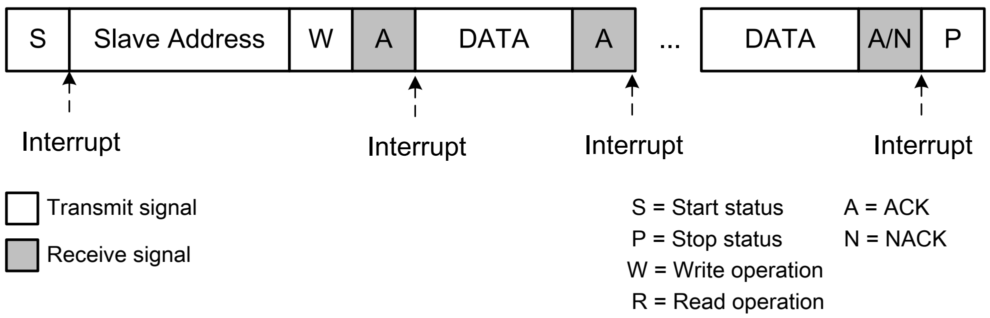


18.2 I2C 传输模式

I2C 可作为主机/从机，执行 8 位串行数据的发送/接收操作，因此，该模块共有 4 种操作模式：主机发送，主机接收，从机发送和从机接收。

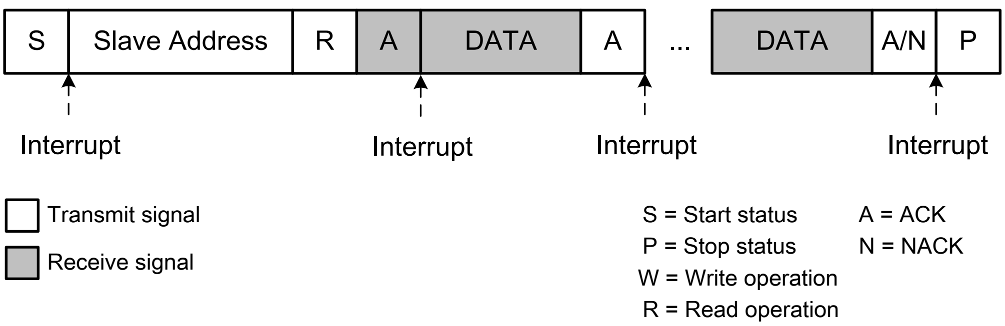
18.2.1 主机发送模式

主机发送模式为主机发送数据给从机，串行时钟由 SCL 输出时，串行数据由 SDA 输出。主机通过发送 START 信号来开始数据发送。发送 START 信号后，开始发送从机设备指定的地址字节。地址字节包含 7 位地址位，第 8 位为数据方向位（R/W），该位设为 0 时使能主机发送。接下来，主机发送一个或多个数据字节给从机，每发送一个字节之后，主机要等待从机返回的 ACK 信号。最后，主机产生 STOP 信号来结束此次数据传输。



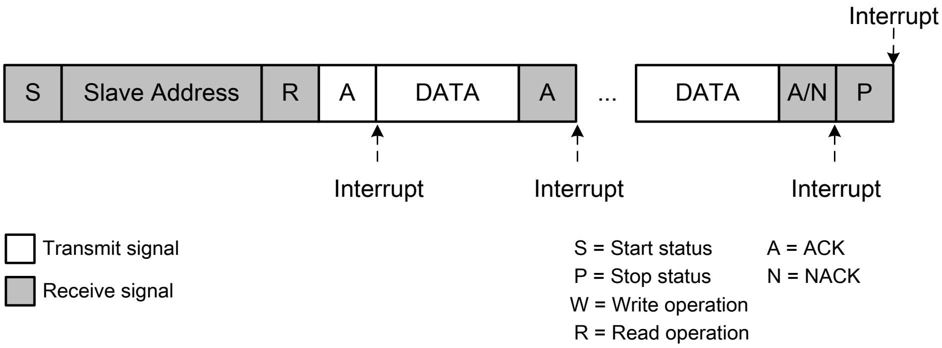
18.2.2 主机接收模式

主机接收模式为主机接收来自从机的数据，串行时钟由 SCL 输出时，串行数据由 SDA 输入。主机通过发送 START 信号来开始数据接收。发送 START 信号后，开始发送从机设备指定的地址字节。地址字节包含 7 位地址位，第 8 位为数据方向位（R/W），该位设为 1 时使能主机接收。接下来，主机接收来自从机的一个或多个数据字节，每接收一个字节之后，通过设置 I2CCON 寄存器的 AA 标志位来将 ACK 或 NACK 信号发送给从机。最后，主机产生 STOP 信号来结束此次数据传输。



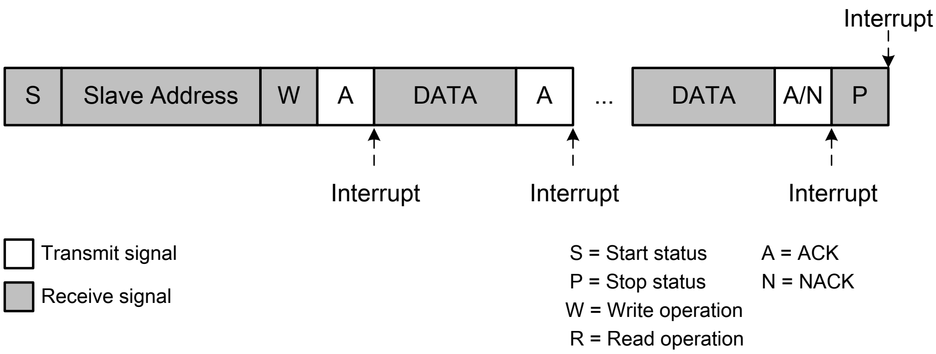
18.2.3 从机发送模式

从机发送模式为从机发送数据给主机，串行时钟由 SCL 输入时，串行数据由 SDA 输出。接收到来自主机的 START 信号后开始数据发送。接收到 START 信号后，开始接收从机设备指定的地址字节。地址字节包含 7 位地址位，第 8 位为数据方向位（R/W），该位设为 1 时使能从机发送。若接收到的地址字节与 I2CADR 寄存器中的地址相匹配，从机将发送 ACK 信号；另外，若广播呼叫地址标志位设为 1（GC=1），接收到广播呼叫地址（0x00））后，从机设备也会发送一个 ACK 信号。接下来，从机发送一个或多个数据字节给主机，每发送一个字节之后，从机要等待主机返回的 ACK 信号。最后，主机产生 STOP 信号来结束此次数据传输。



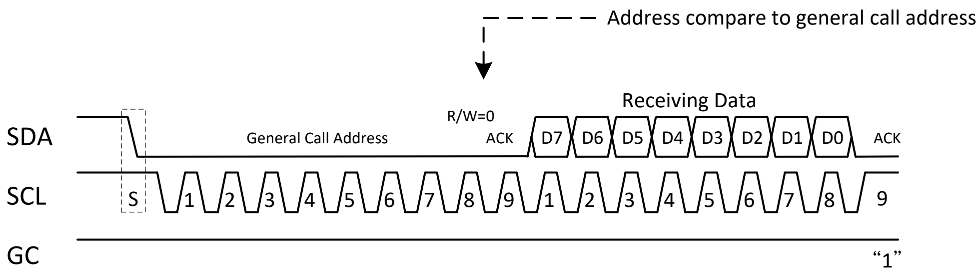
18.2.4 从机接收模式

从机接收模式为从机接收来自主机的数据，串行时钟和串行数据都由 SCL 和 SDA 输入。接收到来自主机的 START 信号后开始数据接收。接收到 START 信号后，开始接收从机设备指定的地址字节。地址字节包含 7 位地址位，第 8 位为数据方向位（R/W），该位设为 0 时使能从机接收。若接收到的地址字节与 I2CADR 寄存器中地址相匹配，从机将产生 ACK 信号；另外，若广播呼叫地址标志位设为 1（GC=1），接收到广播呼叫地址（00H）后，从机设备也会产生一个 ACK 信号。接下来，从机接收一个或多个来自主机的数据字节，每接收一个字节之后，从机产生 ACK 或 NACK 信号，通过设置 I2CCON 寄存器的 AA 标志位来将 ACK 或 NACK 信号发送给主机。最后，主机产生 STOP 信号来结束此次数据传输。



18.3 广播呼叫地址

在 I2C 总线中，开始信号之后的第一个字节中的头 7 位为从机地址，只有该地址与从机地址相匹配时，从机才会响应 ACK 信号。只有广播呼叫地址是个例外，广播呼叫地址可以寻址所有的从机设备。当总线上出现广播呼叫地址时，所有设备应当响应一个 ACK 信号。广播呼叫地址是一个由全 0 组成的 7 位特殊地址。广播呼叫地址功能通过 GC 标志位来控制，设置 GC 标志位为 1 将使能广播呼叫地址，清 0 将关闭广播呼叫地址。当 GC=1 时，将会识别是否是广播呼叫地址，否则当 GC=0 时，将会忽略广播呼叫地址。



18.4 串行时钟发生器

在主机模式下，SCL 时钟速率发生器由 I2CCON 寄存器中的 CR[2:0]标志位来控制。

当 CR[2:0]=000~110 时，SCL 时钟速率来自内部时钟源。

$$\text{SCL Clock Rate} = \frac{\text{Fcpu}}{\text{Prescaler}} (\text{Prescaler} = 256\sim60)$$

当 CR[2:0]=111 时，SCL 时钟速率来自 T1 定时器的溢出频率。

$$\text{SCL Clock Rate} = \frac{\text{Timer 1 Overflow}}{8}$$

下表列出了不同设置下的 SCL 时钟速率。

CR2	CR1	CR0	I2C	Bit Frequency (kHz)
			Prescaler	8MHz
0	0	0	256	31
0	0	1	36	36
0	1	0	192	42
0	1	1	160	50
1	0	0	960	8
1	0	1	120	67
1	1	0	60	133
1	1	1	(Timer 1 overflow rate)/8	

> ✱ *注：*
> ***1. I2C 操作的第一步是设置 I2C 引脚的模式。在 SDA/SCL 引脚必须设置为“输入模式”。***
> ***2. 当时钟源是 T1 溢出率时，最大的计数器值为 0xFB(仅支持 0x00~0xFB)。此时，如果 T1 时钟率是 IHRC_32MHz，SCL 最大时钟率是 800kHz。***
> ***3. 若用户想产生 100KHz/400KHz 的 SCL 时钟速率，需提前设置 T1 计数器的值为 0xD8/0xF6。***

18.5 同步与仲裁

在多主机条件下，同一时间只有一个主机可在总线上传输数据。需要决定由哪个主机可控制总线以及实现传输。时钟同步与仲裁应用于配置多主机传输。时钟同步会在和其他设备同步 SCL 信号时运行。

当同一时间有两个主机要传输时，时钟同步将会在 SCL 由高电平转变成低电平的时候开始。如果主机 1 先将 SCL 切换为低电平，那么主机 1 会将 SCL 锁定在低电平状态直到将其切换成高电平状态。不过，如果有其他主机的 SCL 时序依然保持在低电平状态的话，那么主机 1 将 SCL 时序由低电平切换成高电平状态的过程将不会改变 SCL 的状态，SCL 时序将仍保持在低电平状态。也就是说，SCL 线被有最长低电平周期的主机保持在低电平状态。当所有设备的时钟周期都转换到高电平时，SCL 线将由低转成到高电平状态。在这其间，主机 1 将保持在由低电平到高电平的等待状态，之后再继续它的传输。经过时钟同步之后，所有设备的时钟和 SCL 时钟是一样的了。仲裁是用来决定哪个主机能够通过 SDA 信号来完成它的传输。两个主机可能会在同一时间发出开始信号以及传输数据，导致两者会互相影响。仲裁会强制使得其中一个主机失去总线的控制权。数据传输依然会继续，直到两个主机输出了不同的数据信号。如果其中一个主机传输了高电平状态，而另外一个主机传输了低电平状态，SDA 线会被拉低。输出高电平状态的主机将会检测到 SDA 线上的异常，并失去总线的控制权。输出低电平状态的主机成功获得总线的控制权，并继续它的传输，仲裁的过程中不会丢失数据。

18.6 系统管理总线扩展

可选的系统管理总线(SMBus) 协议的硬件支持三种类型的超时检测： (1) Tmext 超时检测：一个字节的累积持续时钟周期；(2)Tsext 超时检测：开始信号束信号之间的累积持续时钟周期；(3)超时检测：低电平时钟周期的测量。

通过 SMBSEL 和 SMBDST 寄存器来控制超时检测。SMBSEL 寄存器中的 SMBEXE 标志位是 SMBus 拓展功能的使能位。当 SMBEXE=1 时，使能 SMBus 拓展功能。否则，关闭 SMBus 拓展功能。超时类型和周期设置由 SMBTOP[2:0]和 SMBDST 寄存器来控制。SMBus 超时的周期由 Tmex，Tsext 以及 Tout 这个 16 位的暂存器来控制。计算公式如下：

$$\text{Tmext/Tsext/Tout} = \frac{\text{Timeout Period(sec)xFcpu(Hz)}}{1024}$$

Tmext 由 Tmext_L 和 Tmext_H 这两个 8 位寄存器组成，Tmext_L 为低字节，Tmext_H 为高字节。Tsext 由 Tsext_L 和 Tsext_H 这两个 8 位寄存器组成，Tsext_L 为低字节，Tsext_H 为高字节。Tout 由 Tout_L 和 Tout_H 这两个 8 位寄存器组成，Tout_L 为低字节，Tout_H 为高字节。

Type	Time out period	Fcpu=8MHz	
		DEC	HEX
Tmext	5ms	39	27
Tsext	25ms	195	C3
Tout	35ms	273	111

通过设置 SMBTOP[2:0]来选择要设置的寄存器类型（如下表所示），将要配置的数据写到 SMBDST 寄存器中即可。

SMBTOP[2:0]	SMBDST	说明
000	Tmext_L	选择 Tmext 寄存器的低字节
001	Tmext_H	选择 Tmext 寄存器的高字节
010	Tsext_L	选择 Tsext 寄存器的低字节
011	Tsext_H	选择 Tsext 寄存器的高字节
100	Tout_L	选择 Tout 寄存器的低字节
101	Tout_H	选择 Tout 寄存器的高字节

当 SMBus 拓展功能使能之后，I2CSTA 寄存器的低三位指示超时的相关信息，如下表所示：

I2CSTA	说明
XXXX X000	没有超时错误
XXXX XXX1	Tout 超时错误
XXXX XX1X	Tsext 超时错误
XXXX X1XX	Tmext 超时错误

18.7 省电

I2C 模块具有时钟门控功能用于省电功能。ENS1=0 时，停止 I2C 模块内部时钟以省电，I2C 相关寄存器（I2CDAT，I2CADR，I2CCON，I2CSTA，SMBSEL 和 SMBDST）都停止工作。ENS1=1 时，I2C 内部时钟正常工作，相关寄存器也正常工作。在初始化 I2C 之前，ENS1 必须设置为 0。

18.8 I2C 寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I2CDAT	I2CDAT7	I2CDAT6	I2CDAT5	I2CDAT4	I2CDAT3	I2CDAT2	I2CDAT1	I2CDAT0
I2CADR	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0	GC
I2CCON	CR2	ENS1	STA	STO	SI	AA	CR1	CR0
I2CSTA	I2CSTA7	I2CSTA6	I2CSTA5	I2CSTA4	I2CSTA3	I2CSTA2	I2CSTA1	I2CSTA0
SMBSEL	SMBEXE	-	-	-	I2CMX	SMBSTP2	SMBSTP1	SMBSTP0
SMBDST	SMBD7	SMBD6	SMBD5	SMBD4	SMBD3	SMBD2	SMBD1	SMBD0
IEN0	EAL	EI2C	EU0RX	EU0TX	ET1	EX1	ET0	EX0
P0M	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M
P1M	P17M	P16M	P15M	P14M	P13M	P12M	P11M	P10M

I2CDAT 寄存器(0xDA)

Bit	Field	Type	Initial	说明
7:0	I2CDAT[7:0]	R/W	0x00	I2CDAT 寄存器存储的是要通过 I2C 总线发送出去的一个字节数据，或者是刚从 I2C 总线上接收到的一个字节数据。当它不在字节移位的过程中时，控制器可以读写这个 8 位的直接寻址特殊功能寄存器。由于 I2CDAT 寄存器没有缓存和双缓冲，因此当发生 I2C 中断时，用户只能读 I2CDAT 寄存器。

I2CADR 寄存器 (0xDB)

Bit	Field	Type	Initial	说明
7:1	I2CADR[6:0]	R/W	0x00	I2C 从机地址。
0	GC	R/W	0	广播呼叫地址(0X00) 确认。 0：忽略； 1：识别。

I2CCON 寄存器(0xDC)

Bit	Field	Type	Initial	说明
7,1,0	CR[2:0]	R/W	0	I2C clock rate。 000：fcpu/256； 001：fcpu/224； 010：fcpu/192； 011：fcpu/160； 100：fcpu/960； 101：fcpu/120； 110：fcpu/60； 111：T1 溢出周期/8。
6	ENS1	R/W	0	I2C 功能。 0：禁用以省电*； 1：使能以使 I2C 正常工作。
5	STA	R/W	0	START 标志位。 0：没有发送 START 开始信号； 1：若总线空闲则发送 START 开始信号。
4	STO	R/W	0	STOP 标志位。 0：没有发送 STOP 结束信号； 1：若 I2C 总线为主机模式，则发送 STOP 结束信号。
3	SI	R/W	0	串行中断标志位。 当进入了 I2C 的 26 个状态当中的 25 个状态时，SI 标志位会被硬件置 1，只有当 I2C 状态寄存器为 F8h 时，SI 标志位才没有被置 1，表示没有可用的相关状态信息。SI 标志位必须由软件清零，必须通过写 0 到 SI 标志才可以清 SI 标志位，写入 1 到该位并不能更改 SI 的值。
2	AA	R/W	0	有效 ACK 标志位。 0：接收到 1 个字节后返回 NACK； 1：接收到 1 个字节后返回 ACK。

* ENS1=0 时，禁止 I2C 相关寄存器，停止模块内部时钟。

I2CSTA 寄存器 (0xDD)

Bit	Field	Type	Initial	说明
7:3	I2CSTA[7:3]	R	11111	I2C 状态码
2..0	I2CSTA[2:0]	R	000	SMBus 状态码

I2C 状态码和状态

模式	状态代码	I2C 的状态	应用软件响应					I2C 硬件引起的下步操作
			To/from I2CDAT	TO I2CCON				
				STA	STO	SI	AA	
主机发送器/接收器	08H	已发送 START 开始信号	载入 SLA+R	X	0	0	X	发送 SLA+R；接收 ACK。
	10H	已发送重复 START 开始信号	载入 SLA+R	X	0	0	X	发送 SLA+R；接收 ACK。
			载入 SLA+W					发送 SLA+W；I2C 切换到主机发送模式。
主机发送器	18H	已发送 SLA+W；并已接收到 ACK	载入数据字节	0	0	0	X	发送数据字节；接收 ACK。
			无动作	1	0	0	X	发送重复 START 开始信号。
			无动作	0	1	0	X	发送 STOP 结束信号；复位 STO 标志位。
			无动作	1	1	0	X	发送 STOP 结束信号后接着再发送 START 开始信号；复位 STO 标志位。
	20H	已发送 SLA+W；并已接收 NACK	载入数据字节*	0	0	0	X	发送数据字节；接收 ACK。
			无动作	1	0	0	X	发送重复 START 开始信号。
			无动作	0	1	0	X	发送 STOP 结束信号；复位 STO 标志位。
			无动作	1	1	0	X	发送 STOP 结束信号后接着再发送 START 开始信号；复位 STO 标志位。
	28H	已发送 I2CDAT 中的数据字节；并已接收 ACK	加载数据字节	0	0	0	X	发送数据字节；接收 ACK。
			无动作	1	0	0	X	发送重复 START 开始信号。
			无动作	0	1	0	X	发送 STOP 结束信号；复位 STO 标志位。
			无动作	1	1	0	X	发送 STOP 结束信号后接着再发送 START 开始信号；复位 STO 标志位。
	30H	已发送 I2CDAT 中的数据字节；并已接收 NACK	加载数据字节*	0	0	0	X	发送数据字节；接收 ACK。
			无动作	1	0	0	X	发送重复 START 开始信号。
			无动作	0	1	0	X	发送 STOP 结束信号；复位 STO 标志位。
			无动作	1	1	0	X	发送 STOP 结束信号后接着再发送 START 开始信号；复位 STO 标志位。
主机接收器	40H	已发送 SLA+R；并已接收 ACK	无动作	0	0	0	0	接收数据字节；返回 NACK。
			无动作	0	0	0	1	接收数据字节；返回 ACK。
	48H	已发送 SLA+R；并已接收 NACK	无动作	1	0	0	X	发送重复 START 开始信号。
			无动作	0	1	0	X	发送 STOP 结束信号；复位 STO 标志位。
			无动作	1	1	0	X	发送 STOP 结束信号后接着再发送 START 开始信号；复位 STO 标志位。
	50H	已接收数据字节；并已返回 ACK	读数据字节	0	0	0	0	接收数据字节；返回 NACK。
			读数据字节	0	0	0	1	接收数据字节；返回 ACK。
	58H	接收数据字节；并已返回 NACK	读数据字节	1	0	0	X	发送重复 START 开始信号。
			读数据字节	0	1	0	X	发送 STOP 结束信号；复位 STO 标志位。
			读数据字节	1	1	0	X	发送 STOP 结束信号后接着再发送 START 开始信号；复位 STO 标志位。

模式	状态代码	I2C 的状态	应用软件响应					I2C 硬件引起的下步操作
			To/from I2CDAT	TO I2CCON				
				STA	STO	SI	AA	
从机接收器	60H	已接收到对应的 SLA+W ；已返回 ACK	无动作	X	0	0	0/1	接收数据字节；返回 NACK/ACK
	68H	主机在发送 SLA+R/W	无动作	X	0	0	0/1	接收数据字节；返回 NACK/ACK

		时仲裁丢失；并接收到了对应的 SLA+W，已返回 ACK						
	70H	接收到了广播呼叫地址（00H）；已返回 ACK	无动作	X	0	0	0/1	接收数据字节；返回 NACK/ACK
	78H	主机在发送 SLA+R/W 时仲裁丢失；并接收到广播呼叫地址（00H）；已返回 ACK	无动作	X	0	0	0/1	接收数据字节；返回 NACK/ACK
	80H	从机地址已寻址成功；已接收 DATA；并返回 ACK	读数据字节	X	0	0	0/1	接收数据字节；返回 NACK/ACK
	88H	从机地址已寻址成功；已接收 DATA；并返回 NACK	读数据字节	0	0	0	0	切换至未寻址的从机模式；没识别到从机地址或广播呼叫地址。
			读数据字节	0	0	0	1	切换至未寻址的从机模式；从机地址或广播呼叫地址将会被识别到。
			读数据字节	1	0	0	0	切换至未寻址的从机模式；没识别到从机地址或广播呼叫地址；总线空闲之后将发送开始信号。
			读数据字节	1	0	0	1	切换至未寻址的从机模式；从机地址或广播呼叫地址将会被识别到；总线空闲之后将发送开始信号。
	90H	广播呼叫地址已寻址成功；已接收 DATA；并返回 ACK	读数据字节	X	0	0	0/1	接收数据字节；返回 NACK/ACK
	98H	广播呼叫地址已寻址成功；已接收 DATA；并返回 NACK	读数据字节	0	0	0	0	切换至未寻址的从机模式；没识别到从机地址或广播呼叫地址。
			读数据字节	0	0	0	1	切换至未寻址的从机模式；从机地址或广播呼叫地址将会被识别到。
			读数据字节	1	0	0	0	切换至未寻址的从机模式；没识别到从机地址或广播呼叫地址；总线空闲之后将发送开始信号。
			读数据字节	1	0	0	1	切换至未寻址的从机模式；从机地址或广播呼叫地址将会被识别到；总线空闲之后将发送开始信号。
	A0H	当从机接收器或者从机发生器仍然被寻址时，接收到了 STOP 结束信号或者重复 START 开始信号	无动作	0	0	0	0	切换至未寻址的从机模式；没识别到从机地址或广播呼叫地址。
			无动作	0	0	0	1	切换至未寻址的从机模式；从机地址或广播呼叫地址将会被识别到。
			无动作	1	0	0	0	切换至未寻址的从机模式；没识别到从机地址或广播呼叫地址；总线空闲之后将发送开始信号。
			无动作	1	0	0	1	切换至未寻址的从机模式；从机地址或广播呼叫地址将会被识别到；总线空闲之后将发送开始信号。
从机发送器	A8H	已接收对应的 SLA+R；已返回 ACK	载入数据字节	X	0	0	0	发送最后一个字节的数据，并将接收到 ACK 信号
			载入数据字节	X	0	0	1	发送一个字节的数据，并将接收到 ACK 信号。
	B0H	主机在发送 SLA+R/W 时仲裁丢失了；已接收到了对应的 SLA+R，且已返回 ACK 信号。	载入数据字节	X	0	0	0	发送最后一个字节的数据，并将接收到 ACK 信号
			载入数据字节	X	0	0	1	发送一个字节的数据，并将接收到 ACK 信号。
	B8H	数据字节已经发送，将接收到 ACK 信号。	载入数据字节	X	0	0	0	发送最后一个字节的数据，并将接收到 ACK 信号
			载入数据字节	X	0	0	1	发送一个字节的数据，并将接收到 ACK 信号。
	C0H	数据字节已经发送，并已接收到 NACK 信号	无动作	0	0	0	0	切换至未寻址的从机模式；没识别到从机地址或广播呼叫地址。
			无动作	0	0	0	1	切换至未寻址的从机模式；从机地址或广播呼叫地址将会被识别到。

			无动作	1	0	0	0	切换至未寻址的从机模式；没识别到从机地址或广播呼叫地址；总线空闲之后将发送开始信号。
			无动作	1	0	0	1	切换至未寻址的从机模式；从机地址或广播呼叫地址将会被识别到；总线空闲之后将发送开始信号。
	C8H	最后一个字节的数据已经发送，并已接收到 ACK 信号。	无动作	0	0	0	0	切换至未寻址的从机模式；没识别到从机地址或广播呼叫地址。
			无动作	0	0	0	1	切换至未寻址的从机模式；从机地址或广播呼叫地址将会被识别到。
			无动作	1	0	0	0	切换至未寻址的从机模式；没识别到从机地址或广播呼叫地址；总线空闲之后将发送开始信号。
			无动作	1	0	0	1	切换至未寻址的从机模式；从机地址或广播呼叫地址将会被识别到；总线空闲之后将发送开始信号。
其他	F8H	没有可用的相关状态信息；SI=0	无动作	No action				等待或者进行当前的传输
	38H	仲裁丢失	无动作	0	0	0	X	I2C 总线将被释放；将发送开始信号。
			无动作	1	0	0	X	当总线空闲时（进入了主机模式）
	00H	在主机模式下或已选址的从机模式时，总线出错。	无动作	0	1	0	1	在主机模式或已寻址的从机模式时，只有内部硬件才会被影响。在所有情况下，总线已被释放，I2C 接口已切换至未寻址的从机模式，STO 标志位已复位。

"SLA"表示从机地址，"R"表示 R/W=1，"W"表示 R/W=0。

*表示接收到 NACK 之后不意味着通信的结束。

SMBSEL 寄存器(0xDE)

Bit	Field	Type	Initial	说明
7	SMBEXE	R/W	0	SMBus 扩展功能。 0：禁用； 1：使能。
3	I2CMX	R/W	0	0 = I2C SCL 和 SDA 引脚为 P00 和 P01； 1 = I2C SCL 和 SDA 引脚为 P14 和 P15。
2..0	SMBSTP[2:0]	R/W	000	SMBus 超时寄存器

SMBDST 寄存器 (0xDF)

Bit	Field	Type	Initial	说明
7..0	SMBD[7:0]	R/W	0x00	该寄存器用于提供一个读写 SMBus 超时寄存器的端口。从 SMBDST 寄存器读写的数据实际上是对由 SMBSEL 指定的超时寄存器进行读写的数据。

IEN0 寄存器 (0xA8)

Bit	Field	Type	Initial	说明
7	EAL	R/W	0	中断使能位。参考中断章节。
Else				参考其它章节。

P0M 寄存器 (0xF9)

Bit	Field	Type	Initial	说明
1	P01M	R/W	0	0：设置 P0.1 (SDA) 为输入模式（要求）； ~~1：设置 P0.1 (SDA) 为输出入模式。*~~
0	P00M	R/W	0	0：设置 P0.0 (SCL) 为输入模式（要求）； ~~1：设置 P0.0 (SCL) 为输出入模式。*~~
Else				参考其它章节。

* 要求 P00M 和 P01M 设为输入模式。

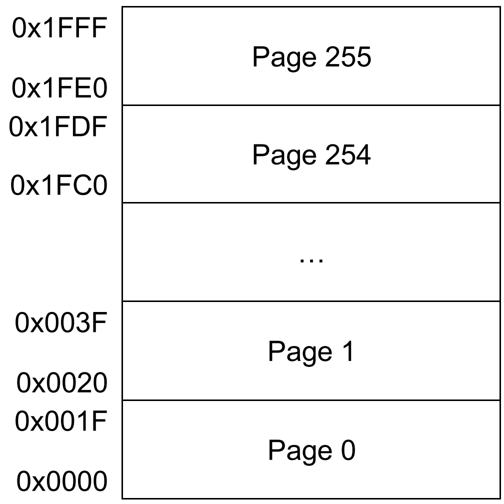
P1M 寄存器 (0xFA)

Bit	Field	Type	Initial	说明
5	P15M	R/W	0	0：设置 P1.5 (SDA) 为输入模式（要求）； ~~1：设置 P1.5 (SDA) 为输出入模式。*~~
4	P14M	R/W	0	0：设置 P1.4 (SCL) 为输入模式（要求）； ~~1：设置 P1.4 (SCL) 为输出入模式。*~~
Else				参考其它章节。

* 要求 P14M 和 P15M 分别设为输入模式。

19 In-System Program

SN8F5713 内置 8KB 程序存储（IROM），均分为 256 页（每页 32 字节）。In-System Program 就是使能软件随意的更改每页的数据，换言之，就是存储数据到 Flash 存储器或者现场升级软件的一个通道。



程序存储器 (IROM)

19.1 页编程

由于程序存储器的每页都是 32 字节的长度，页编程就要求 32 字节 IRAM 作为其数据缓存器。

ISP ROM MAP		ROM 地址 bit0~bit4 (hex) =0
ROM address bit5~bit15 (hex)	0000	这些页包括复位向量和中断向量。强烈建议保留该区，不做 ISP 擦除动作。
	0020	
	0040	
	…	
	00C0	
	00E0	
	0100	一个 ISP 程序页
	0120	一个 ISP 程序页
	…	一个 ISP 程序页
	1000	一个 ISP 程序页
	1020	一个 ISP 程序页
	…	一个 ISP 程序页
	1700	一个 ISP 程序页
	1720	一个 ISP 程序页
	…	一个 ISP 程序页
	1FE0	该页包括 ROM 保留区。强烈建议保留该区，不做 ISP 擦除动作。

在开始页编程之前必须将下列配置设置完成，按以下步骤配置 ISP：

1、保存编程数据到 IRAM，这些数据连续 32 字节。
2、在 PERAM 中设置数据位置的起始地址。
3、在 PEROM[15:5]中设置计划升级区域的起始地址（由 PEROMH/PEROML 寄存器设置）。
4、写入 0x5A 到 PECMD[7:0]，触发 ISP 功能。
5、写 NOP 指令两次。

举例来说，假设程序存储器的第 254 页（IROM，0x1FC0 – 0x1FDF）为计划升级区域，内容已经存入 IRAM 地址 0x60 – 0x7F。为了执行 In-System Program，只要写入开始 IROM 地址 0x1FC0 到 EPROMH/EPROML 寄存器，然后指定缓存器开始地址 0x60 到 EPRAM 寄存器，随后写入 0x5A 到 PECMD[7:0]寄存器，复制缓存器的数据到 IROM 的第 254 页。

通常而言，每页的内容都可通过 In-System Program 进行修改，但前 10 页和最后一页都不可以编程，会导致上电错误或复位错误。

19.2 字节编程

字节编程支持一个字节存储器程序，一个字节程序不止要求一个字节 IRAM 作为其数据缓存器。

在开始字节编程之前必须将下列配置设置完成，按以下步骤配置 ISP：

1、保存编程数据到 IRAM，这些数据只有 1 个字节。
2、在 PERAM 中设置数据位置的起始地址。
3、在 PEROM[15:0]中设置计划升级区域的起始地址（由 PEROMH/PEROML 寄存器设置）。
4、写入 0x1E 到 PECMD[7:0]，触发 ISP 功能。
5、写 NOP 指令两次。

举例来说，假设 IRPM 的地址 0x1FC5 为计划升级区域，内容已经存入 IRAM 地址 0x60。为了执行 In-System Program，只要写入开始 IROM 地址 0x1FC5 到 EPROMH/EPROML 寄存器，然后指定缓存器开始地址 0x60 到 EPRAM 寄存器，随后写入 0x1E 到 PECMD[7:0]寄存器，复制缓存器的数据到 IRPM 的地址 0x1FC5。

✷ ***注：***

1. ***在 Flash 写（编程）操作前，必须清看门狗，否则在 ISP 运行过程中，看门狗定时器会溢出并导致系统复位；***

2. ***第一页和最后一页不要执行 ISP FLASH ROM，否则影响编程操作；***

3. ***ISP 操作(页编程)实际为后台执行 Flash 擦除和编程进程，禁止在低电压(例如 VDD<2.5V）执行 ISP Flash ROM 编程，否则可能导致掉电前 ISP 操作未完成。***

19.3 In-System Program 寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PERAM	PERAM7	PERAM6	PERAM5	PERAM4	PERAM3	PERAM2	PERAM1	PERAM0
PEROMH	PEROM15	PEROM14	PEROM13	PEROM12	PEROM11	PEROM10	PEROM9	PEROM8
PEROML	PEROM7	PEROM6	PEROM5	PEROM4	PEROM3	PEROM2	PEROM1	PEROM0
PECMD	PECMD7	PECMD6	PECMD5	PECMD4	PECMD3	PECMD2	PECMD1	PECMD0

PERAM 寄存器(0x97)

Bit	Field	Type	Initial	说明
7..0	PERAM[7:0]	R/W	0x00	数据缓存器（IRAM）的第一个地址。

PEROMH 寄存器(0x96)

Bit	Field	Type	Initial	说明
7..0	PEROM[15:8]	R/W	0x00	编程页（IROM）的第一个地址（15th– 8th bit）。

PEROML 寄存器(0x95)

Bit	Field	Type	Initial	说明
7..0	PEROM[7:0]	R/W	000	编程页（IROM）的地址（7th–5th bit）。

PECMD 寄存器(0x94)

Bit	Field	Type	Initial	说明
7..0	PECMD[7:0]	W	-	0x5A：开始整页编程； 0x1E：起始字节编程步骤； ~~其它值：保留*(1)。~~

*(1)不允许写其它值到 PECMD 寄存器。

20 时钟微调

SN8F5713 内置时钟微调功能，这个功能是一个通过固件微调系统时钟频率的程序。通过代码选项(CK_Fine_Tuning)使能该功能。CK_Fine_Tuning = 0 时，禁用时钟微调功能。CK_Fine_Tuning = 1 时，启用时钟微调功能。系统上电后，10 位初始时钟调整值将由硬件加载到 FRQ[9:0]缓冲器，该调整值应该是 IHRC 32 MHZ。改变值 FRQ[9:0]的调整值会改变内部时钟频率。

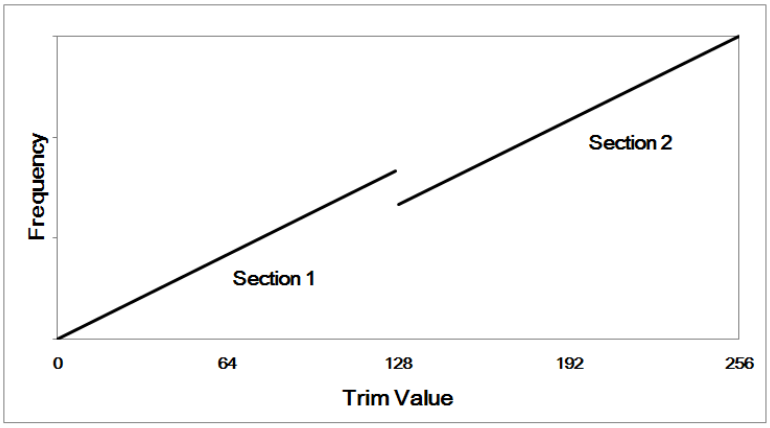
20.1 时钟调整部分

时钟微调有 8 个调整部分，见表 20-1。每个部分包括 128 个调整步骤，在相同的部分，每个步骤应该是(32 MHZ * 0.1%)。调整值越大，频率越快。

表 20-1 时钟调整部分

部分	调整值	频率
1	000H ~ 07FH	低
2	080H ~ 0FFH	
3	100H ~ 17FH	
4	180H ~ 1FFH	
5	200H ~ 27FH	
6	280H ~ 2FFH	
7	300H ~ 37FH	
8	380H ~ 3FFH	高

每个相邻部分有一个频率差，见下图。因此调整值的频率= 127 要快于调整值= 128。



20.2 时钟微调步骤

在开始时钟微调之前必须将这些配置设置完成，步骤如下：

1. 选择代码选项 CK_Fine_Tuning = 1，以使能时钟微调功能。
2. Max. IROM 取指令周期是 8MHz，建议先设置 PWSC[2:0]=7，以避免系统出错。
3. 从 FRQ[9:0]读取 10 位 32MHz 调整值。
4. 按表 17-1，检查微调范围。
5. 新时钟调整值写入 FRQ[9:0]。
6. ‘0x3C’写入 FRQCMD [7:0]，以触发时钟微调功能。

> ✷ ***注：请检查 IROM 取指令周期< 8MHz，以避免系统出错。***

20.3 时钟微调寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FRQH	-	-	-	-	-	-	FRQ9	FRQ8
FRQL	FRQ7	FRQ6	FRQ5	FRQ4	FRQ3	FRQ2	FRQ1	FRQ0
FRQCMD	FRQCMD7	FRQCMD6	FRQCMD5	FRQCMD4	FRQCMD3	FRQCMD2	FRQCMD1	FRQCMD0

FRQ 寄存器(FRQH:0xFD, FRQL:0xFC)

Bit	Field	Type	Initial	说明
9..0	FRQ[9:0]	R/W	0x00	系统时钟校准值。

FRQCMD 寄存器 (0xFE)

Bit	Field	Type	Initial	说明
7..0	FRQCMD[7:0]	W	0x00	0x3C：开始时钟微调程序； 其它值：保留*(1)。

*(1) 不允许写其它值到 FRQCMD 寄存器。

21 电气特性

21.1 极限参数

VDD 至 VSS 的电压……………………………………………………………………………- 0.3V~6.0V
任意引脚至 VSS 的电压…………………………………………………………………… - 0.3V~VDD+0.3V
运行环境温度…………………………………………………………………………………-40°C~85°C
存储环境温度…………………………………………………………………………………-40°C~125°C

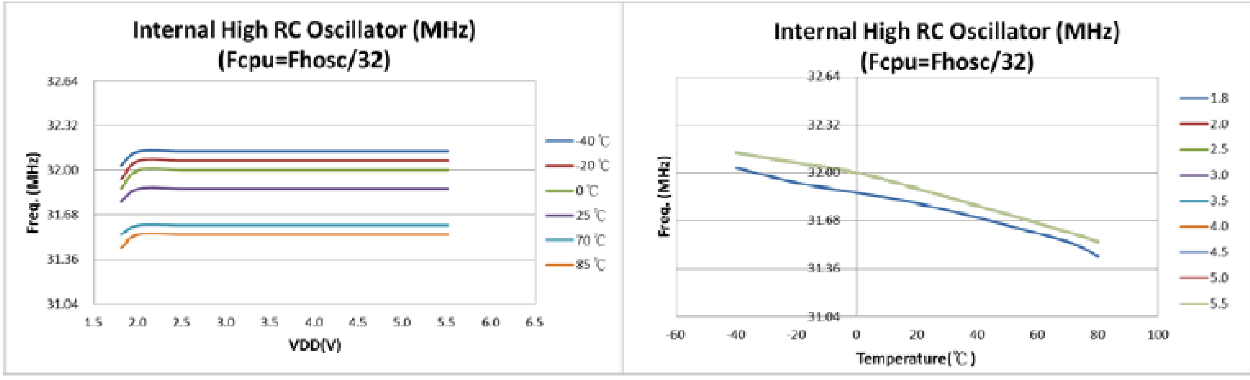
21.2 系统操作特性

参数		测试条件	Min	TYP	MAX	单位
VDD	工作电压	fcpu = 1MHz	1.8		5.5	V
V_{DR}	RAM 数据保持电压		0.55			V
V_{POR}	VDD 上升速率**		0.05			V/ms
I_{DD1}	NORMAL 模式供电电流	VDD = 3V, fcpu = 1MHz		2.75		mA
		VDD = 5V, fcpu = 1MHz		2.80		mA
		VDD = 3V, fcpu = 8MHz		3.65		mA
		VDD = 5V, fcpu = 8MHz		3.70		mA
I_{DD2}	STOP 模式供电电流	VDD = 3V		4.5	8.5	μA
		VDD = 5V		5.0	9.0	μA
I_{DD3}	STOP 模式供电电流 (ILRC enable STWK=1)	VDD = 5V		5.5		μA
I_{DD4}	IDLE 模式供电电流 (fcpu = 1MHz)	VDD = 3V, 32MHz IHRC		0.83		mA
		VDD = 5V, 32MHz IHRC		0.85		mA
		VDD = 3V, 16MHz Crystal		0.65		mA
		VDD = 5V, 16MHz Crystal		1.20		mA
		VDD = 3V, 4MHz Crystal		0.28		mA
		VDD = 5V, 4MHz Crystal		0.47		mA
F_{IHRC}	内部高时钟发生器	VDD = 1.8V to 5.5V, 25°C	-0.5%	32	+0.5%	MHz
		VDD = 1.8V to 5.5V, -40°C to 85°C	-2.0%	32	+2.0%	MHz
F_{ILRC}	内部低时钟发生器	VDD = 5.0V, 25°C	-25%	16	+50%	kHz
V_{LVD18}	LVD18 检测电压	25°C	1.6	1.7	1.8	V
		-40°C to 85°C	1.5	1.7	1.9	V
V_{ESD_HBM}	ESD human body mode		30000			V
V_{ESD_MM}	ESD machine mode		300			V

* 带*号的参数还未经过验证的设计参考值。环境温度为 25°C。

- IHRC 频率 – 温度关系图

IHRC 图表用于设计指导， 不是检验或保证。在一些图表中，所提供的数据是在指定的操作范围之外的。这是仅供参考的，并且保证设备只在指定范围内正常运行。



21.3 GPIO 特性

参数		测试条件	Min	TYP	MAX	单位
V_{IL}	低电平输入电压		VSS		0.3VDD	V
V_{IH}	高电平输入电压		0.7VDD		VDD	V
I_{LEKG}	I/O 口输入漏电流	VIN = VDD			2	μA
R_{UP}	上拉电阻	VDD = 3V	100	200	300	kΩ
		VDD = 5V	50	100	150	kΩ
I_{OH}	I/O 输出 source 电流	VDD = 5V, VO = VDD-0.5V	12	16		mA
I_{OL1}	I/O sink 电流(P04 – P07 , P14 – P17, P20 – P24)	VDD = 5V, VO = VSS+0.5V	15	20		mA
I_{OL2}	I/O sink 电流 (P00 – P03, P10– P13)	VDD = 5V, VO = VSS+1.5V	80	100		mA
V_{bias}	1/2*VDD Bias Voltage (P00 – P03, P10– P13)	VDD = 5V	2.4	2.5	2.6	V

* 环境温度为 25°C.

21.4 ADC 特性

参数		测试条件	Min	TYP	MAX	单位
V_{ADC}	工作电压		2.0		5.5	V
V_{AIN}	AIN 通道输入电压	VDD = 5V	0		VREFH	V
V_{REFH}	AVREFH 引脚输入电压	VDD = 5V	2		VDD	V
V_{IREF}	内部 VDD 参考电压	VDD = 5V		VDD		V
	内部 4V 参考电压	VDD = 5V	-2%	4	+2%	V
	内部 3V 参考电压	VDD = 5V	-2%	3	+2%	V
	内部 2V 参考电压	VDD = 5V	-2%	2	+2%	V
I_{AD}	ADC 功耗	VDD = 3V		0.75		mA
		VDD = 5V		0.80		mA
f_{ADCLK}	ADC 时钟	VDD = 5V			32	MHz
f_{ADSMP}	ADC 时钟率	VDD = 5V			500	kHz
t_{ADEN}	ADC 功能使能周期	VDD = 5V	100			μs
DNL	微分非线性**	f ADSMP = 62.5kHz		±1		LSB
		f ADSMP = 250kHz		±1		
		f ADSMP = 500kHz		±3.5		
INL	积分非线性*	f ADSMP = 62.5kHz		±2		LSB
		f ADSMP = 250kHz		±2		
		f ADSMP = 500kHz		±3		
NMC	无遗漏码*	f ADSMP = 62.5kHz	10	11	12	Bit
		f ADSMP = 250kHz		10		
		f ADSMP = 500kHz		9		
V_{OFFSET}	输入偏移电压	Non-trimmed	-5	0	5	mV

* 带*号的参数：VDD = 5V, V_{REFH} = 2.4V, 25°C.

21.5 Flash 存储器特性

参数		测试条件	Min	TYP	MAX	单位
V_{dd}	供电电压		1.8		5.5	V
T_{pen}	持续时间	25°C		*100K		cycle
T_{ben}	字节持续时间	25°C		*10K		cycle
I_{wrt}	写电流	25°C		3	4	mA
T_{wrt}	写时间	写 1 页= 32 字节，25℃		6	8	ms

*带*号的参数还未经过验证的设计参考值。

22 指令集

本章对 SN8F5713 的综合汇编指令进行分类，总共包括 5 类：算术运算、逻辑运算、数据传输运算、布尔操作和程序跳转指令，这些指令全部都与标准 8051 兼容。

符号说明

符号	功能描述
Rn	工作寄存器 R0 - R7
direct	直接地址：内部 RAM 地址（00H~7FH）或特殊功能寄存器地址（80H~FFH）
@Ri	通过寄存器 R0 或者 R1 来间接寻址内部或外部 RAM 地址
#data	8 位常量（立即操作数）
#data16	16 位常量（立即操作数）
bit	位于内部 RAM（0x20~0x2F）中的 128 个位变量之一，或者是可以位操作的特殊功能寄存器当中的任何标志位
addr16	LCALL 和 LJMP 的目标地址，可以是 64KB 程序存储空间当中的任意地址
addr11	ACALL 和 AJMP 的目标地址，与下一条指令的第一个字节在同一个 2KB 区内（即 16 位地址中的高 5 位地址相同
rel	SJMP 和所有条件跳转指令中的一个 8 位偏移字节，相对于下一条指令的第一个字节，可偏移的范围为+127/-128 字节
A	累加器

算数运算

指令	功能描述
ADD A, Rn	将累加器与寄存器的内容相加，结果存回累加器
ADD A, direct	将累加器与直接地址的内容相加，结果存回累加器
ADD A, @Ri	将累加器与间接地址的内容相加，结果存回累加器
ADD A, #data	将累加器与常量相加，结果存回累加器
ADDC A, Rn	将累加器与寄存器的内容及进位 C 相加，结果存回累加器
ADDC A, direct	将累加器与直接地址的内容及进位 C 相加，结果存回累加器
ADDC A, @Ri	将累加器与间接地址的内容及进位 C 相加，结果存回累加器
ADDC A, #data	将累加器与常量及进位 C 相加，结果存回累加器
SUBB A, Rn	将累加器的值减去寄存器的值减借位 C，结果存回累加器
SUBB A, direct	将累加器的值减去直接地址的值减借位 C，结果存回累加器
SUBB A, @Ri	将累加器的值减去间接地址的值减借位 C，结果存回累加器
SUBB A, #data	将累加器的值减常量减借位 C，结果存回累加器
INC A	将累加器的值加 1
INC Rn	将寄存器的值加 1
INC direct	将直接地址的内容加 1
INC @Ri	将间接地址的内容加 1
INC DPTR	将数据指针寄存器的值加 1
DEC A	将累加器的值减 1
DEC Rn	将寄存器的值减 1
DEC direct	将直接地址的内容减 1
DEC @Ri	将间接地址的内容减 1
MUL AB	将累加器的值与 B 寄存器的值相乘，乘积的低字节存回累加器，高字节存回 B 寄存器
DIV	将累加器的值除以 B 寄存器的值，结果的商存回累加器，余数存回 B 寄存器
DA A	将累加器的值作十进制调整

逻辑运算

指令	功能描述
ANL A, Rn	将累加器的值与寄存器的值作逻辑与运算，结果存回累加器
ANL A, direct	将累加器的值与直接地址的内容作逻辑与运算，结果存回累加器
ANL A, @Ri	将累加器的值与间接地址的内容作逻辑与运算，结果存回累加器
ANL A, #data	将累加器的值与常量作逻辑与运算，结果存回累加器
ANL direct, A	将直接地址的内容与累加器的值作逻辑与运算，结果存回直接地址
ANL direct, #data	将直接地址的内容与常量作逻辑与运算，结果存回直接地址
ORL A, Rn	将累加器的值与寄存器的值作逻辑或运算，结果存回累加器
ORL A, direct	将累加器的值与直接地址的内容作逻辑或运算，结果存回累加器
ORL A, @Ri	将累加器的值与间接地址的内容作逻辑或运算，结果存回累加器
ORL A, #data	将累加器的值与常量作逻辑或运算，结果存回累加器
ORL direct, A	将直接地址的内容与累加器的值作逻辑或运算，结果存回直接地址
ORL direct, #data	将直接地址的内容与常量作逻辑或运算，结果存回直接地址
XRL A, Rn	将累加器的值与寄存器的值作逻辑异或运算，结果存回累加器
XRL A, direct	将累加器的值与直接地址的内容作逻辑异或运算，结果存回累加器
XRL A, @Ri	将累加器的值与间接地址的内容作逻辑异或运算，结果存回累加器
XRL A, #data	将累加器的值与常量作逻辑异或运算，结果存回累加器
XRL direct, A	将直接地址的内容与累加器的值作逻辑异或运算，结果存回直接地址
XRL direct, #data	将直接地址的内容与常量作逻辑异或运算，结果存回直接地址
CLR A	将累加器的值清零
CPL A	将累加器的值取反
RL A	将累加器的值左移一位
RLC A	将累加器的值带进位 C 左移一位
RR A	将累加器的值右移一位
RRC A	将累加器的值带进位 C 右移一位
SWAP A	将累加器的高 4 位与低 4 位的内容交换

数据传输操作

指令	功能描述
MOV A, Rn	将寄存器的内容赋值给累加器
MOV A, direct	将直接地址的内容赋值给累加器
MOV A, @Ri	将间接地址的内容赋值给累加器
MOV A, #data	将常量赋值给累加器
MOV Rn, A	将累加器的内容赋值给寄存器
MOV Rn, direct	将直接地址的内容赋值给寄存器
MOV Rn, #data	将常量赋值给寄存器
MOV direct, A	将累加器的内容赋值给直接地址
MOV direct, Rn	将寄存器的内容赋值给直接地址
MOV direct1, direct2	将直接地址 2 的内容赋值给直接地址 1
MOV direct, @Ri	将间接地址的内容赋值给直接地址
MOV direct, #data	将常量赋值给直接地址
MOV @Ri, A	将累加器的内容赋值给间接地址
MOV @Ri, direct	将直接地址的内容赋值给间接地址
MOV @Ri, #data	将常量赋值给间接地址
MOV DPTR, #data16	16 位常量赋值给数据指针寄存器
MOVC A, @A+DPTR	累加器的值加数据指针寄存器的值作为其所指定地址，将该地址的内容赋值给累加器
MOVC A, @A+PC	累加器的值加程序计数器的值作为其所指定地址，将该地址的内容赋值给累加器
MOVX A, @Ri	将间接地址所指定外部存储器的内容赋值给累加器（8 位地址）
MOVX A, @DPTR	将数据指针所指定外部存储器的内容赋值给累加器（16 位地址）
MOVX @Ri, A	将累加器的内容赋值给间接地址所指定的外部存储器（8 位地址）
MOVX @DPTR, A	将累加器的内容赋值给数据指针所指定的外部存储器（16 位地址）
PUSH direct	将直接地址的内容压入堆栈区
POP direct	将栈顶的内容弹出至直接地址
XCH A, Rn	将累加器与寄存器的内容互换
XCH A, direct	将累加器与直接地址的内容互换
XCH A, @Ri	将累加器与间接地址的内容互换
XCHD A, @Ri	将累加器的低 4 位与间接地址的低 4 位互换

布尔运算

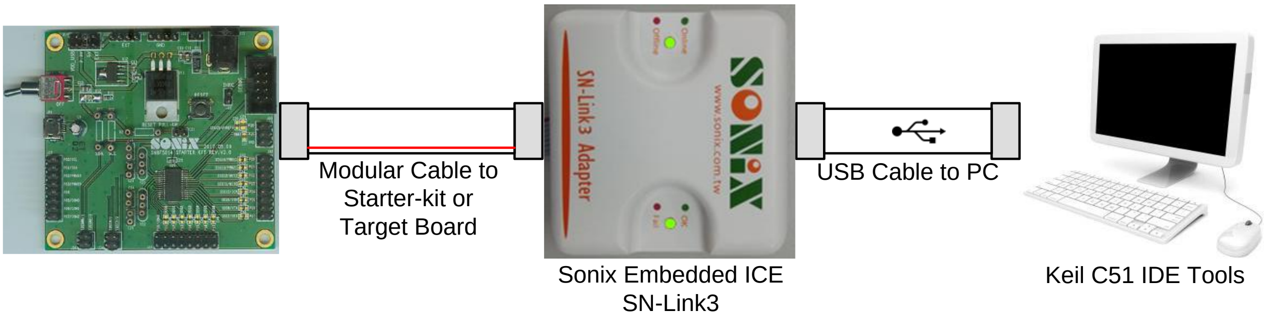
指令	功能描述
CLR C	将进位 C 清零
CLR bit	将直接寻址位清零
SETB C	设置进位 C 为 1
SETB bit	设置直接寻址位为 1
CPL C	将进位 C 的值取反
CPL bit	将直接寻址位的值取反
ANL C, bit	将进位 C 的值与直接寻址位的值作逻辑与运算，结果存回进位 C
ANL C, /bit	将进位 C 的值与直接寻址位的反码作逻辑与运算，结果存回进位 C
ORL C, bit	将进位 C 的值与直接寻址位的值作逻辑或运算，结果存回进位 C
ORL C, /bit	将进位 C 的值与直接寻址位的反码作逻辑或运算，结果存回进位 C
MOV C, bit	将直接寻址位的值赋值给进位 C
MOV bit, C	将进位 C 的值赋值给直接寻址位

程序跳转指令

指令	功能描述
ACALL addr11	绝对调用
LCALL addr16	长调用
RET	从子程序返回
RETI	从中断返回
AJMP addr11	绝对跳转
LJMP addr16	长跳转
SJMP rel	短跳转（相对地址）
JMP @A+DPTR	跳至累加器的值加数据指针寄存器的值所指定的地址
JZ rel	若累加器的值为 0，则跳转至 rel 所指定的地址
JNZ rel	若累加器的值不为 0，则跳转至 rel 所指定的地址
JC rel	若进位 C 的值为 1，则跳转至 rel 所指定的地址
JNC rel	若进位 C 的值为 0，则跳转至 rel 所指定的地址
JB bit, rel	若直接寻址位的值为 1，则跳转至 rel 所指定的地址
JNB bit, rel	若直接寻址位的值为 0，则跳转至 rel 所指定的地址
JBC bit, rel	若直接寻址位的值为 1，则跳转至 rel 所指定的地址，并将该位值清零
CJNE A, direct, rel	将累加器的内容与直接地址的内容比较 若不相等则跳转至 rel 所指定的地址
CJNE A, #data, rel	将累加器的内容与常量比较，若不相等则跳转至 rel 所指定的地址
CJNE Rn, #data, rel	将寄存器的内容与常量比较，若不相等则跳转至 rel 所指定的地址
CJNE @Ri, #data, rel	将间接地址的内容与常量比较，若不相等则跳转至 rel 所指定的地址
DJNZ Rn, rel	将寄存器的内容减 1，若不等于 0 则跳转至 rel 所指定的地址
DJNZ direct, rel	将直接地址的内容减 1，若不等于 0 则跳转至 rel 所指定的地址
NOP	空指令

23 开发环境

SONIX 提供嵌入式 ICE 仿真器系统，用于 SN8F5713 的 FW 开发。该平台是内电路调试器，并由 Microsoft Windows 平台上的 Keil C51 IDE 软件控制。该平台包括 SN-Link3、SN8F5713 starter-kit 和 Keil C51 IDE 软件，是高速、低成本、功能强大和多任务的开发环境，包括仿真器、调试器和程序员。执行仿真如同在真正的芯片上，因为仿真器电路集成在 SN8F5713 中，可提供真正的开发环境。



23.1 最低要求

为了建立一个合适的开发环境，以下事项至关重要的。所列版本已通过兼容性验证，预计下一个版本会更完善。SN-Link 的相关信息可以从 SONiX 网站(www.sonix.com.tw) 下载;

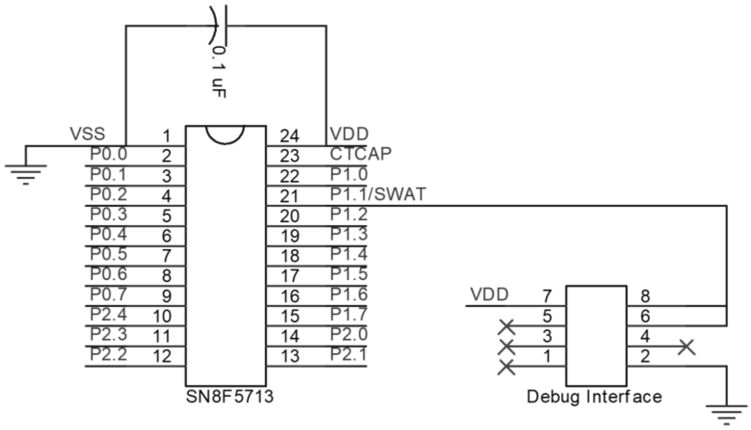
从 www.keil.com/c51 下载 Keil C51。

- **SN-Link3 适配器**，FW 版本升级为 1.02
- **Keil C51 的 SN-Link 驱动器**，版本 1.00.317
- **Keil C51** 版本 9.50a 和 9.54a，或更高

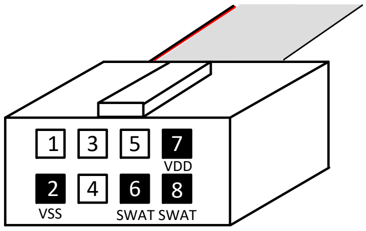
23.2 调试接口硬件

下面的电路图显示了如何正确地连接 MCU SWAT 引脚和 SN-Link3 适配器。

开始调试之前，必须切断 MCU 的电源（VDD）。将 SWAT 引脚连接到 SN-Link 的第 6 脚和第 8 脚；分别将 VDD 引脚和 VSS 引脚连接到 SN-Link 的第 7 脚和第 2 脚；对 MCU 上电，就会自动与 SN-Link 进行握手；当 SN-Link 指示灯亮绿色（运行），则代表连接成功，可以进行仿真调试（更多内容请参考 SN8F5000 调试工具手册）。



示例电路



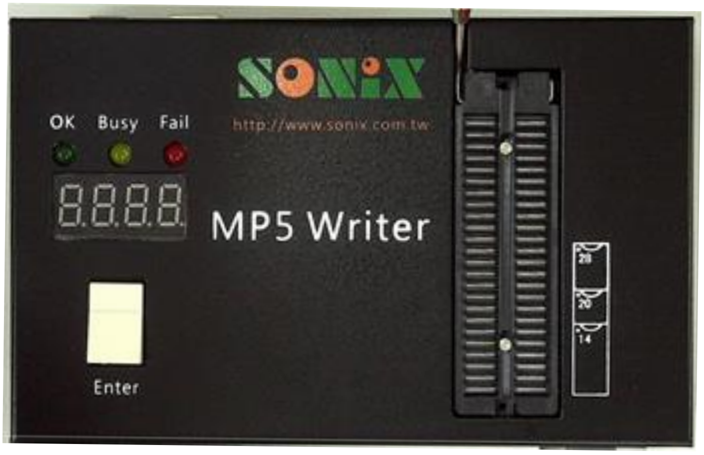
SN-Link header

23.3 开发工具

SN-Link3 适配器

Starter-Kit 支持 SN8F5713, SN8F5712, SN8F5711/131

MP5 烧录器



24 SN8F5713 Start-Kit

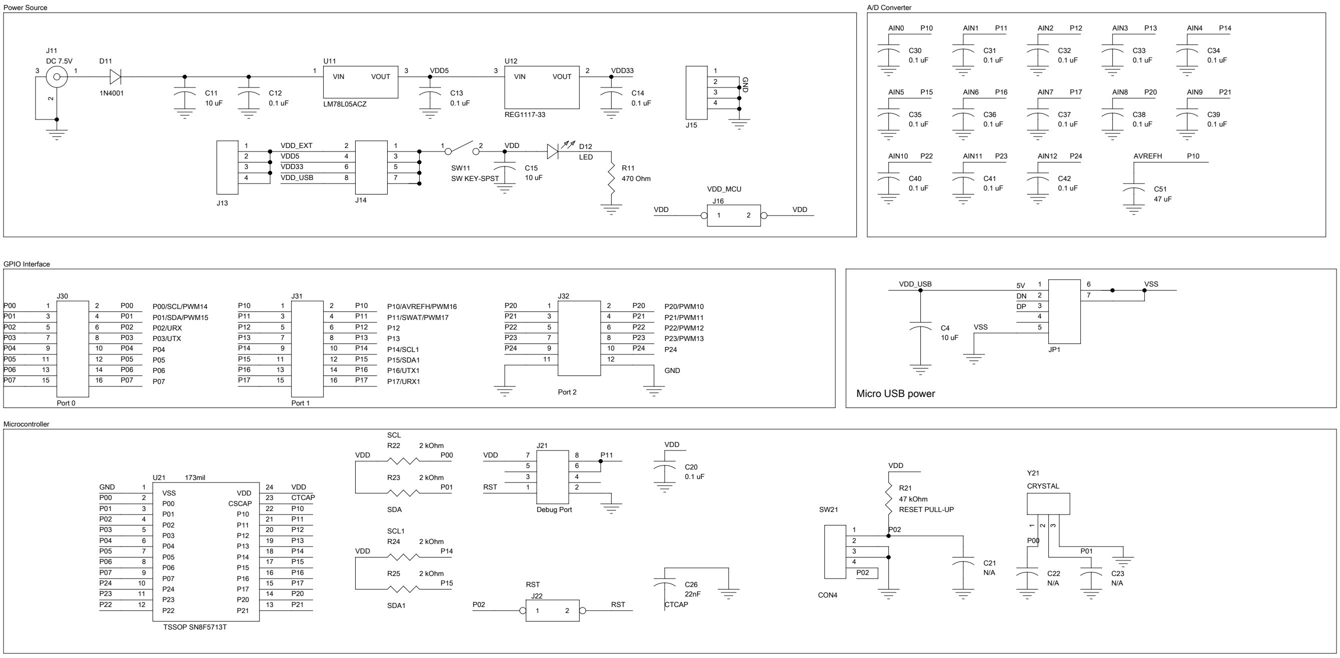
SN8F5000 Starter-Kit 提供一个简易的开发平台。它包含 SN8F5000 系列 IC 和输入信号或用户应用程序驱动设备的 IO 接口。当目标板没有准备好时，它是一个简单开发平台用于应用程序开发。Starer-Kit 可以被目标板所取代，因为 SN8F5000 系列集成了嵌入式的电路调试器电路。

24.1 配置电路

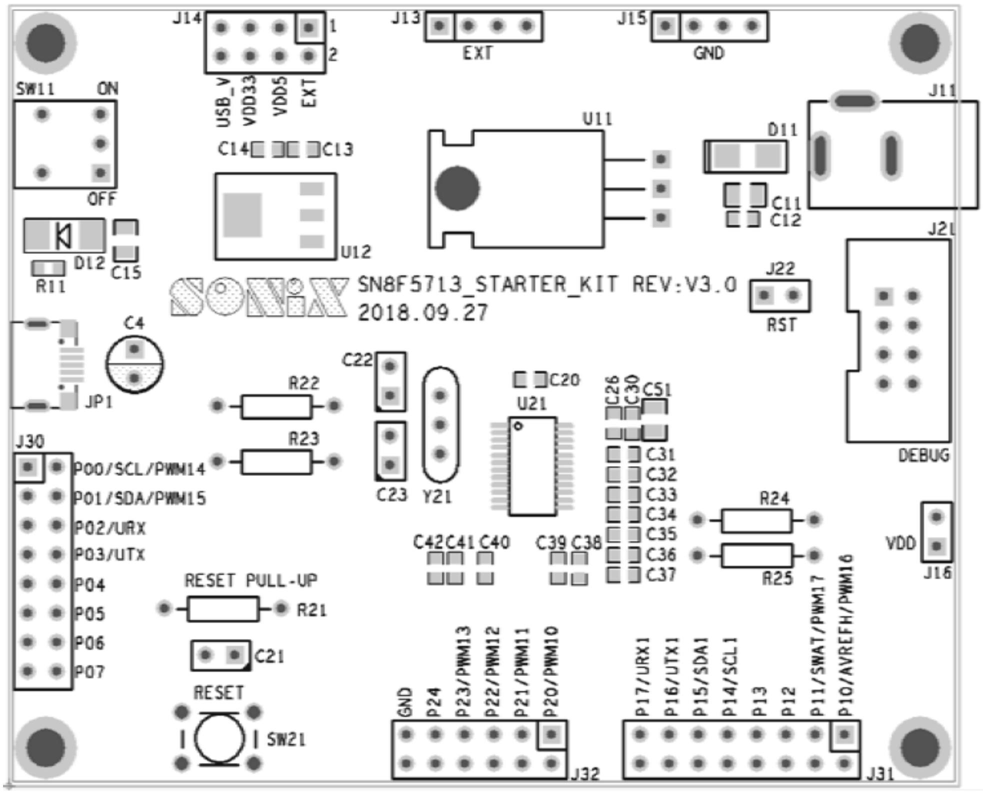
在开始利用 Start-Kit 开发前必须完成以下配置：

1. 确认电路板元件是否完整；
2. starkit 电路的电源是从 5.0 V，3.3 V，外部电源或 MINI USB 接口中选择的；
3. 如果电源来自 5.0 V 或 3.3 V，必须连接 7.5 V 直流电源适配器；
4. 如果电源来自外部电源，则外部电源连接到 EXT；
5. 当选择外部复位时，“RST”引脚需要将上拉电阻与 VDD 连接起来；
6. 当系统时钟设置晶振或 RTC 模式时，“XIN”和“XOUT”引脚需要接晶振；
7. 当系统时钟设置外部时钟输入模式时，“XIN”需要连接外部时钟源；
8. 调试端口可以连接 SN-LINK 适配器进行仿真或下载代码；
9. 当接通电源VDD时，MCU　LED灯会亮起来，SN8F5000系列IC将上电。

24.2 电路



24.3 PCB 布局平面图



24.4 元件描述

符号	功能描述
C30-C42	13 通道 ADC 电容
C51	AVREFH 电容
D12	MCU LED
J11	DC 7.5V 电源适配器
J13/J15	外部电源
SW21	外部复位触发源
J14	VDD 电源选择 5.0V，3.3V 或外部电源
J21	仿真接口
J30-J32	IO 口
R21，C21	外部复位上拉电阻和电容
R22，R23，R24，R25	IIC 上拉电阻
SW11	目标电源开关
U21	SN8F5713 实际芯片
Y21，C22，C23	外部晶振/谐振器部分
JP1	Micro USB 接口

25 ROM 烧录引脚

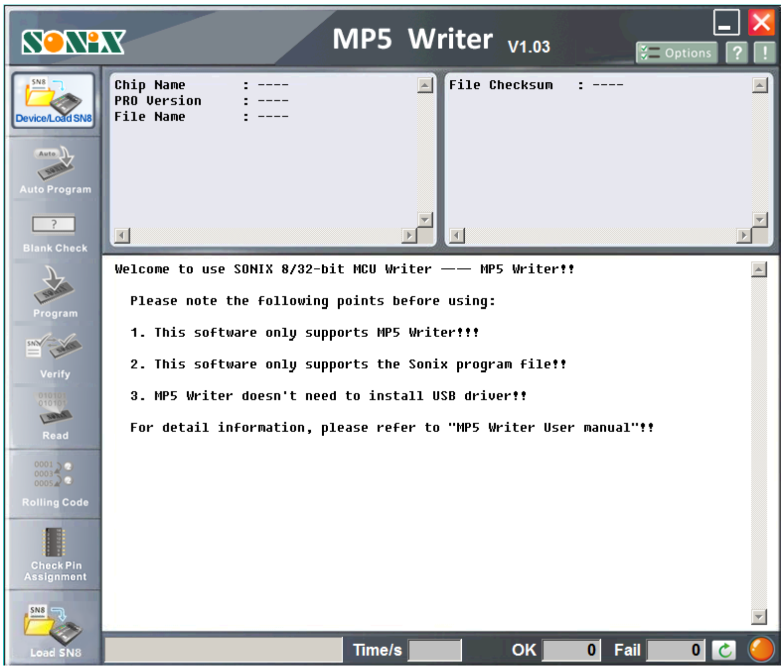
SN8F5713 系列 Flash ROM 的擦除/编程/验证支持 SN-Link 和 MP5 烧录器：

- SN-Link：调试界面和主要用于板上烧录；
- MP5 Writer：主要用于 SN8F5713 系列版本的批量烧录。

25.1 MP5 硬件连接

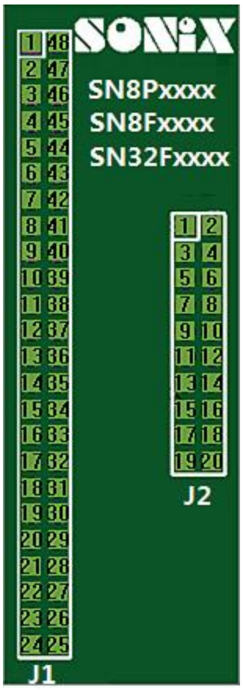
不同封装 MCU 烧录时的硬件接法如下，DIP 和 SOP/TSSOP 说明。

操作界面如下：



25.2 MP5 烧录器转接板插座引脚配置

MP5 烧录器转接板：



25.3 MP5 烧录器烧录引脚图

烧录器连接器		MCU 引脚名称	SN8F5713SG/TG		SN8F5712PG/SG		SN8F5711PG/SG		SN8F57131PG/SG	
J2 引脚编号	J2 引脚编号		MCU 引脚编号	J1 引脚编号	MCU 引脚编号	J1 引脚编号	MCU 引脚编号	J1 引脚编号	MCU 引脚编号	J1 引脚编号
1	VDD	VDD	24	36	20	34	16	32	14	31
2	GND	VSS	1	13	1	15	1	17	1	18
7.9	SWAT	P1.1	21	33	17	31	13	29	12	29
20	PDB	P1.6	16	28	14	28	12	28	11	28

烧录器连接器		MCU 引脚名称	SN8F5713JG		SN8F57112SG		SN8F57113SG	
J2 引脚编号	J2 引脚编号		MCU 引脚编号	J1 引脚编号	MCU 引脚编号	J1 引脚编号	MCU 引脚编号	J1 引脚编号
1	VDD	VDD	22	34	8	28	8	28
2	GND	VSS	23	35	1	21	1	21
7.9	SWAT	P1.1	19	31	6	26	6	26
20	PDB	P1.6	14	26	4	24	4	24

25.4 SN-Link ISP 烧录

SN-Link ISP 烧录硬件和软件如下。

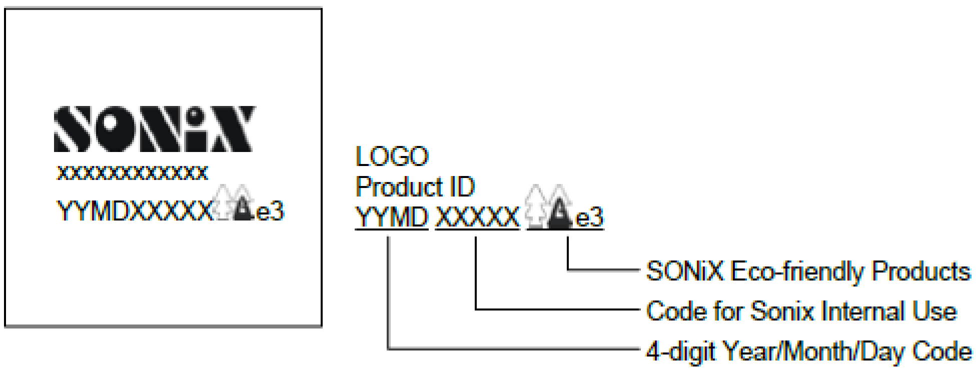
25.5 SN-Link ISP 烧录引脚图

SN-Link 连接器		MCU 引脚编号	SN8F5713S/T	SN8F5712P/S	SN8F5711P/S	SN8F57131P/S
引脚编号	引脚编号		引脚编号	引脚编号	引脚编号	引脚编号
7	VDD	VDD	24	20	16	14
2	GND	VSS	1	1	1	1
6.8	SWAT	P1.1	21	17	13	12

SN-Link 连接器		MCU 引脚编号	SN8F5713J	SN8F57112S	SN8F57113S
引脚编号	引脚编号		引脚编号	引脚编号	引脚编号
7	VDD	VDD	24	20	16
2	GND	VSS	1	1	1
6.8	SWAT	P1.1	21	17	13

26 订购信息

SONiX 的单片机表面印有三栏信息：商标，单片机全称和日期码。



26.1 命名规格

Full Name	Packing Type
SN8F5713W	Wafer
SN8F5713H	Dice
SN8F5713SG	SOP, 24 脚，绿色封装
SN8F5713TG	TSSOP, 24 脚，绿色封装
SN8F5713JG	QFN, 24 脚，绿色封装
SN8F5712PG	DIP, 20 脚，绿色封装
SN8F5712SG	SOP, 20 脚，绿色封装
SN8F5711PG	DIP, 16 脚，绿色封装
SN8F5711SG	SOP, 16 脚，绿色封装
SN8F57131PG	DIP, 14 脚，绿色封装
SN8F57131SG	SOP, 14 脚，绿色封装
SN8F57112SG	SOP, 8 脚，绿色封装
SN8F57113SG	SOP, 8 脚，绿色封装

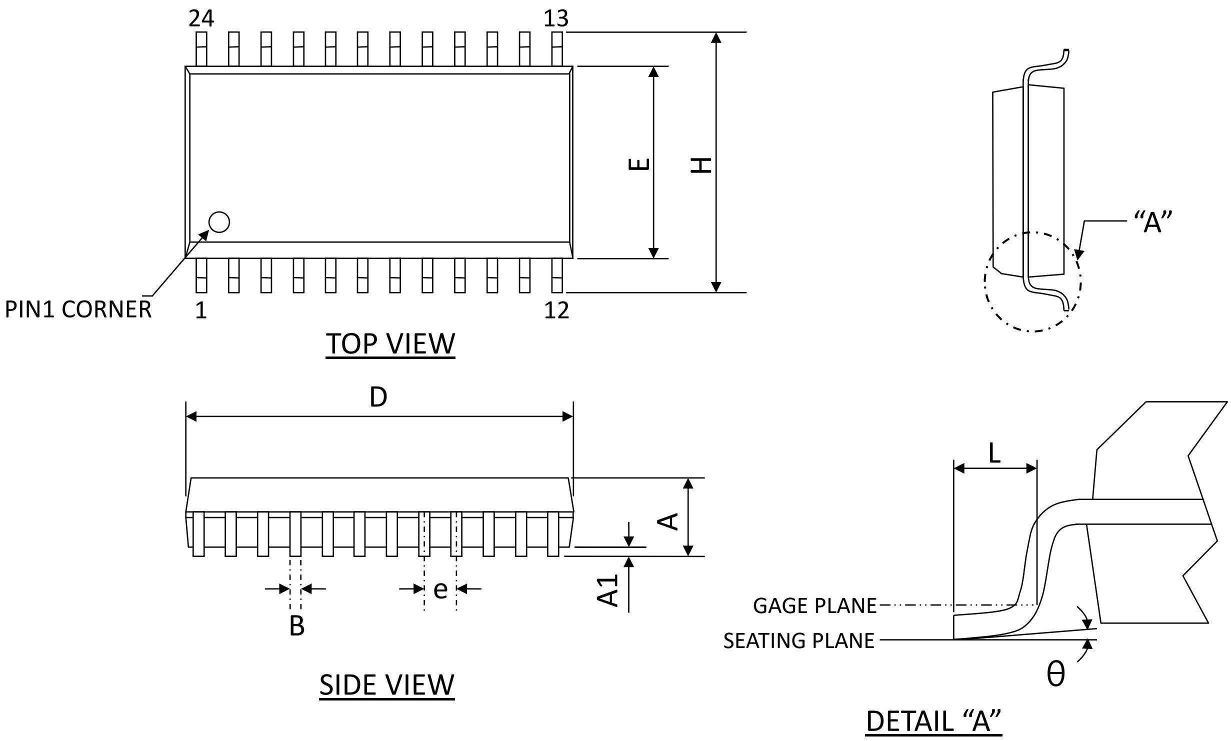
26.2 日期码

日期码包括 2 个信息：生产日期和产品序列号。生产日期是公共信息，详见下表。

Year	18：2018 19：2019 20：2020 21：2021 et cetera
Month	1：January 2：February 3：March 4：April 5：May 6：June 7：July 8：August 9：September A：October C：November D：December
Date	1：01 2：02 3：03 4：04 5：05 6：06 7：07 8：08 9：09 A：10 C：11 D：12 E：13 F：14 G：15 H：16 J：17 K：18 L：19 M：20 N：21 P：22 Q：23 R：24 S：25 T：26 U：27 V：28 W：29 X：30 Y：31

27 封装信息

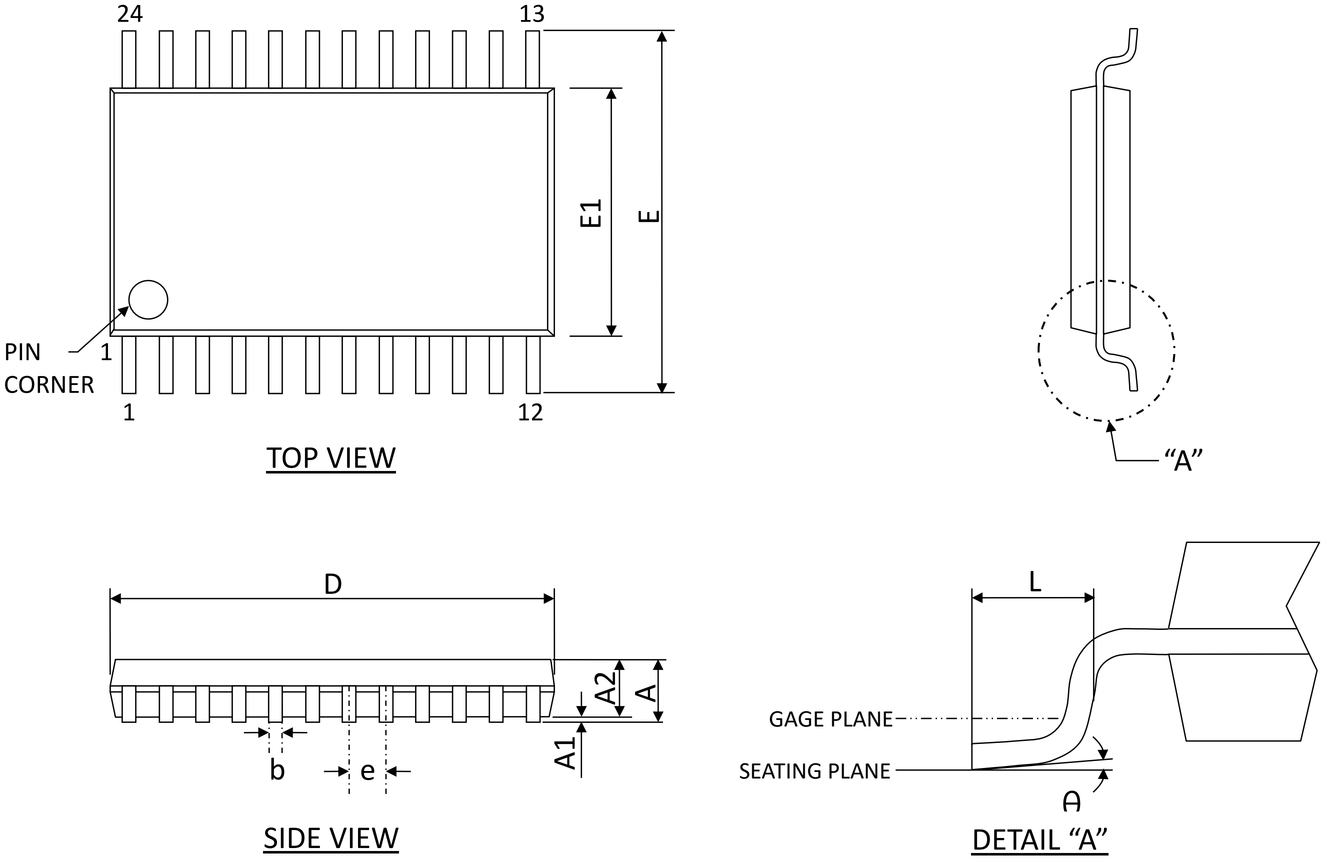
27.1 SOP24



SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	2.65	--	--	0.104
A1	0.10	--	0.30	0.004	--	0.011
B	0.31	0.41	0.51	0.012	0.016	0.020
D	15.30	15.50	15.70	0.602	0.618	0.618
E	7.50 BSC			0.295 BSC		
e	1.27 BSC			0.050 BSC		
H	10.30 BSC			0.405 BSC		
L	0.4	--	1.27	0.015	--	0.05
θ	0°	4。	8°	0°	4。	8°

Notes：
1. CONTROLLING DIMENSION： mm
2. JEDEC OUTLINE : MO-119 AA

27.2 TSSOP24

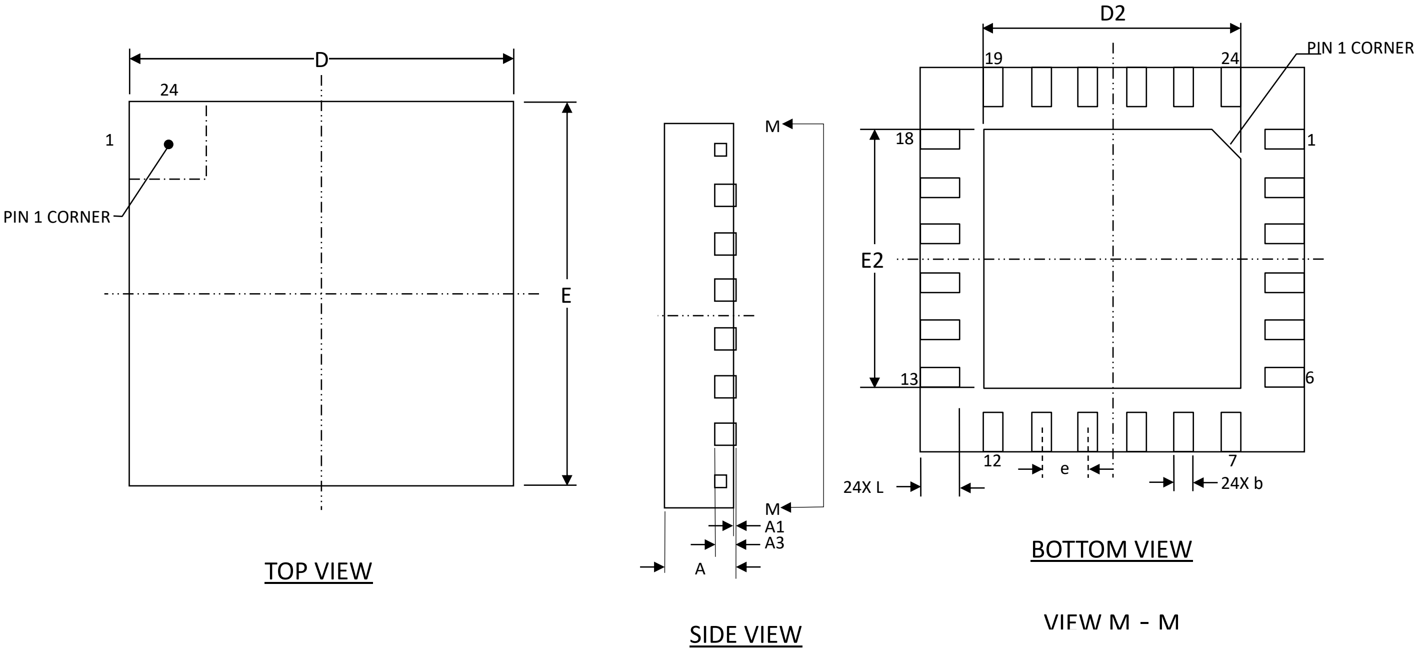


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.20	--	--	0.047
A1	0.00	--	0.15	0.000	--	0.006
A2	0.80	1.00	1.05	0.031	0.039	0.041
b	0.19	--	0.30	0.007	--	0.012
D	7.70	7.80	7.90	0.303	0.307	03.11
E	6.40 BSC.			0.252 BSC.		
E1	4.30	4.40	4.50	0.169	0.173	0.177
e	0.65 BSC.			0.026 BSC.		
L	0.45	0.60	0.75	0.018	0.024	0.030
θ	0°	--	8°	0°	--	8°

Notes：

1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MO-153
3. DIMENSION 'D' DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BERRES.
4. DIMENSION 'E1' DOES NOT INCLUDE INTERLEAD FLASH OR PROTRUSION.
5. DIMENSION 'b' DOES NOT INCLUDE DAMBAR PROTRUSION.

27.3 QFN24 4X4

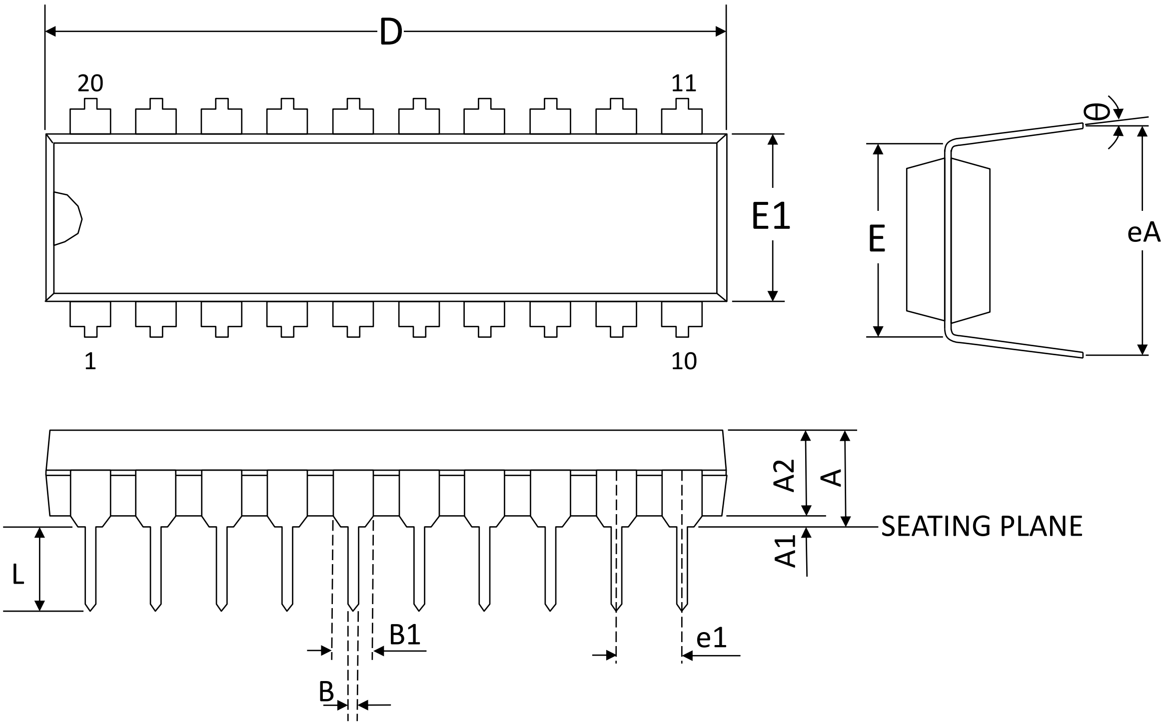


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.80	0.90	0.028	0.031	0.035
A1	0.00	0.02	0.05	0.000	0.001	0.002
A3	0.203 REF			0.008 REF		
b	0.15	0.25	0.30	0.007	0.010	0.012
D	4.00 BSC			0.157 BSC		
E	4.00 BSC			0.157 BSC		
e	0.50 BSC			0.020 BSC		
D2	1.90	2.35	2.80	0.075	0.093	0.110
E2	1.90	2.35	2.80	0.075	0.093	0.110
L	0.30	0.40	0.50	0.012	0.016	0.020

Notes：

1. CONTROLLING DIMENSION：MILLIMETER (mm)

27.4 DIP20

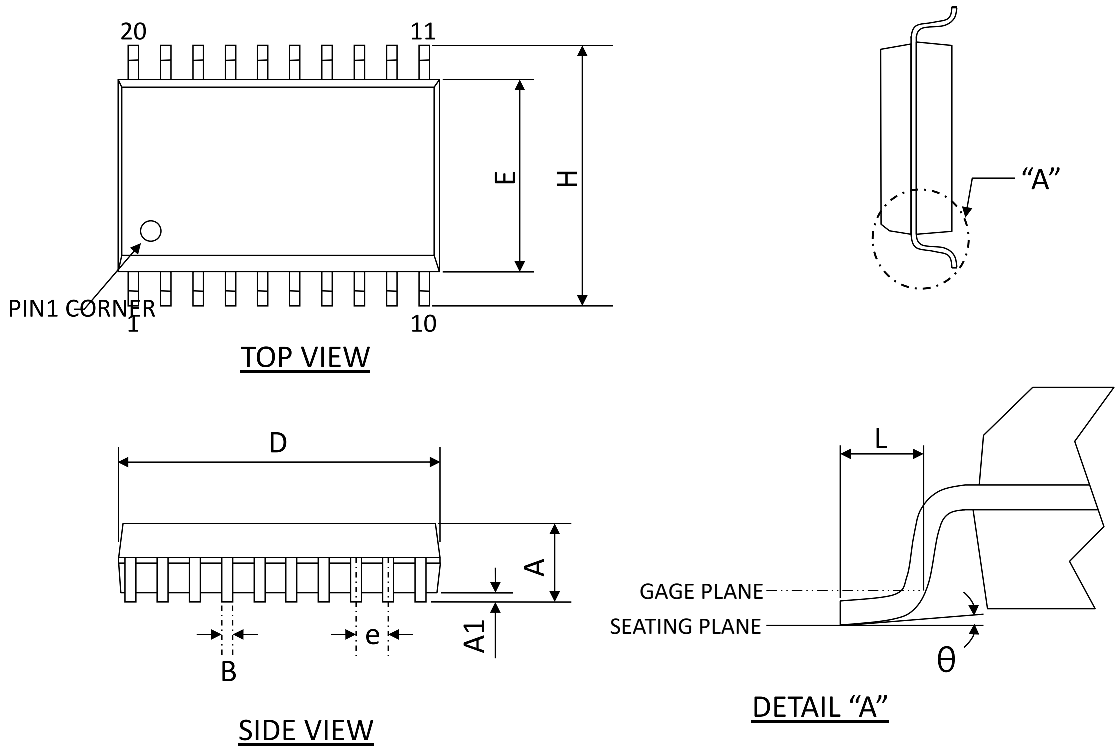


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	4.45	--	--	0.175
A1	0.35	--	--	0.015	--	--
A2	3.18	3.30	3.43	0.125	0.130	0.135
B	0.46 typ.			0.018 typ.		
B1	1.52 typ.			0.060 typ.		
D	25.70	26.06	26.42	1.012	1.026	1.040
E	7.62 BSC.			0.300 BSC.		
E1	6.05	6.35	6.65	0.238	0.250	0.261
e1	2.54 typ.			0.100 typ.		
L	3.05	3.30	3.56	0.120	0.130	0.140
eA	7.62	9.02	9.53	0.300	0.355	0.375
θ	0˚	7˚	15˚	0˚	7˚	15˚

Notes：
1. JEDEC OUTLINE : MS-001 AD
2. CONTROLLING DIMENSION：inch

3.

27.5 SOP20

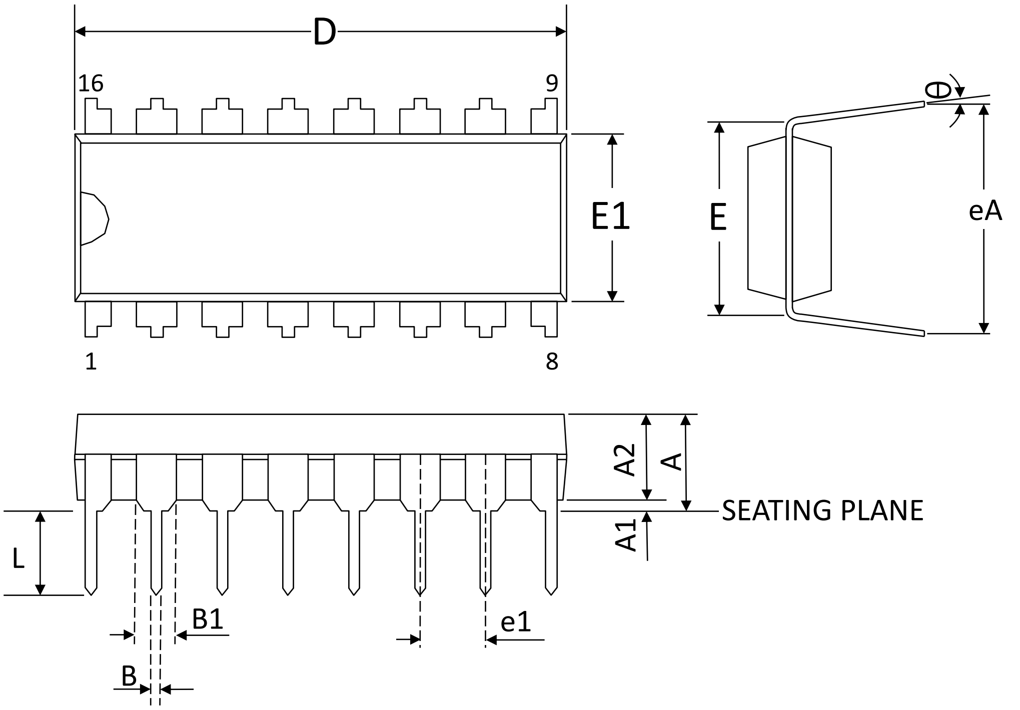


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	2.65	--	--	0.104
A1	0.10	--	0.30	0.004	--	0.012
B	0.31	0.41	0.51	0.012	0.016	0.020
D	12.80 BSC			0.503		
E	7.50 BSC			0.295		
e	1.27 BSC			0.050 BSC		
H	10.30 BSC			0.405		
L	0.40	--	1.27	0.016	--	0.050
θ	0°	4°	8°	0°	4°	8°

Notes：

1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MO-013 AC

27.6 DIP16

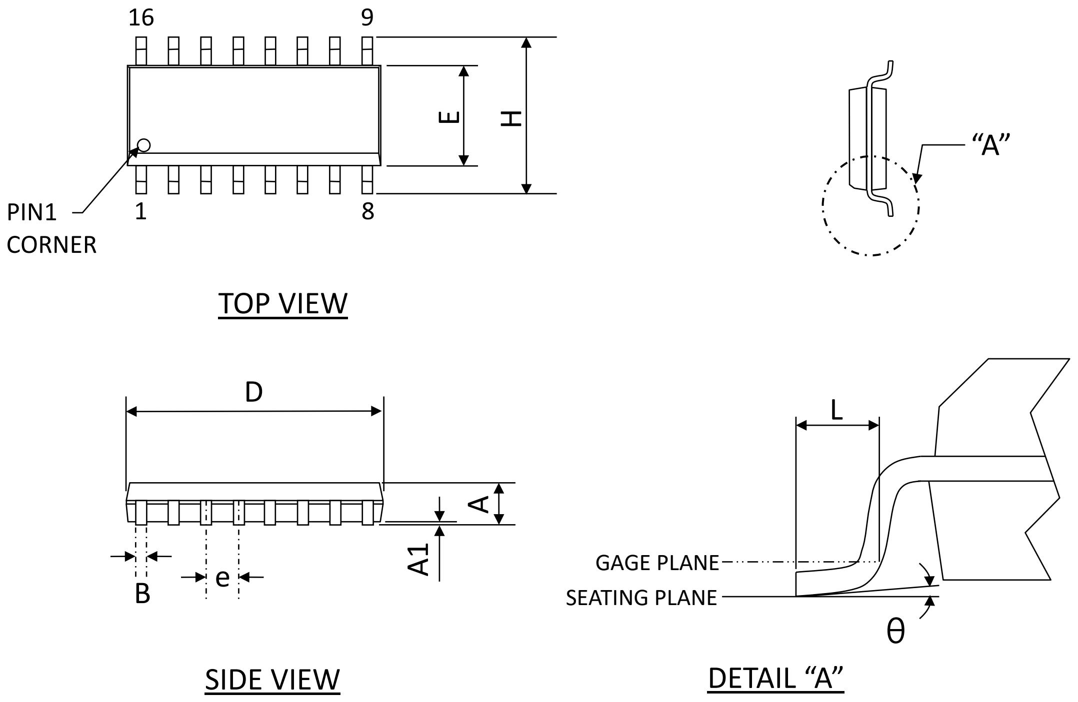


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	5.33	--	--	0.210
A1	0.38	--	--	0.015	--	--
A2	3.18	3.30	3.43	0.125	0.130	0.135
B	0.46 typ.			0.018 typ.		
B1	1.52 typ.			0.060 typ.		
D	18.67	19.18	19.69	0.735	0.755	0.775
E	7.62 BSC.			0.300 BSC		
E1	6.22	6.35	6.48	0.245	0.250	0.255
e1	2.54 typ.			0.100 typ.		
L	2.92	3.30	3.81	0.115	0.130	0.150
eA	7.62	9.02	9.53	0.300	0.355	0.375
θ	0°	7°	15°	0°	7°	15°

Notes：

1. JEDEC OUTLINE : MS-001 BB
2. CONTROLLING DIMENSION：inch

27.7 SOP16

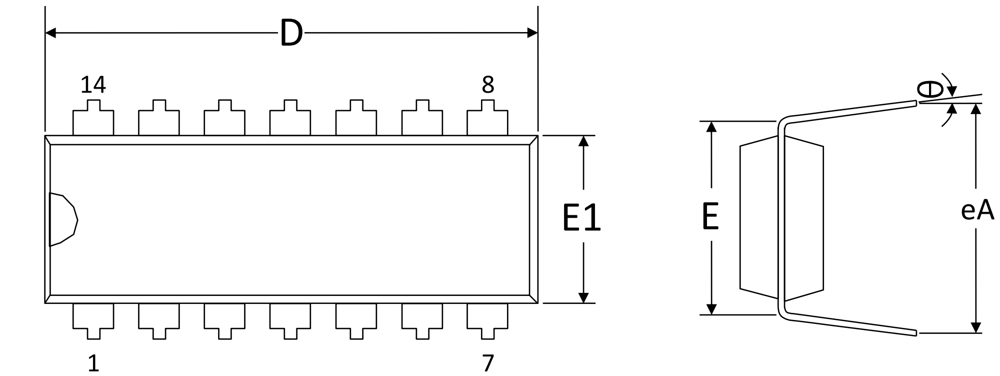


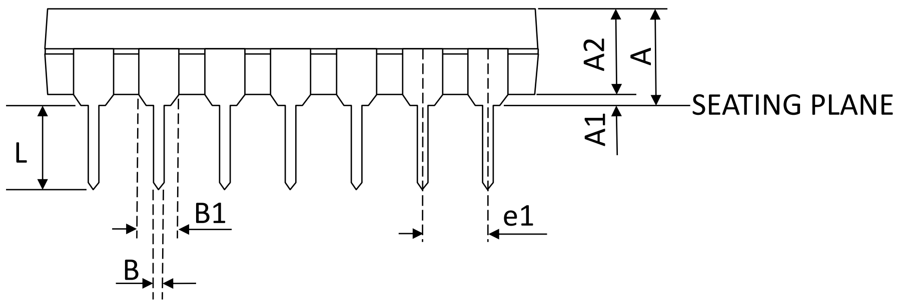
SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.75	--	--	0.069
A1	0.10	--	0.25	0.004	--	0.010
B	0.31	0.41	0.51	0.012	0.016	0.020
D	9.90 BSC			0.389 BSC		
E	3.90 BSC			0.153 BSC		
e	1.27 BSC			0.050 BSC		
H	6.00 BSC			0.236 BSC		
L	0.40	--	1.27	0.016	--	0.050
θ	0°	4°	8°	0°	4°	8°

Notes：

1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MS-012 AC

27.8 DIP14

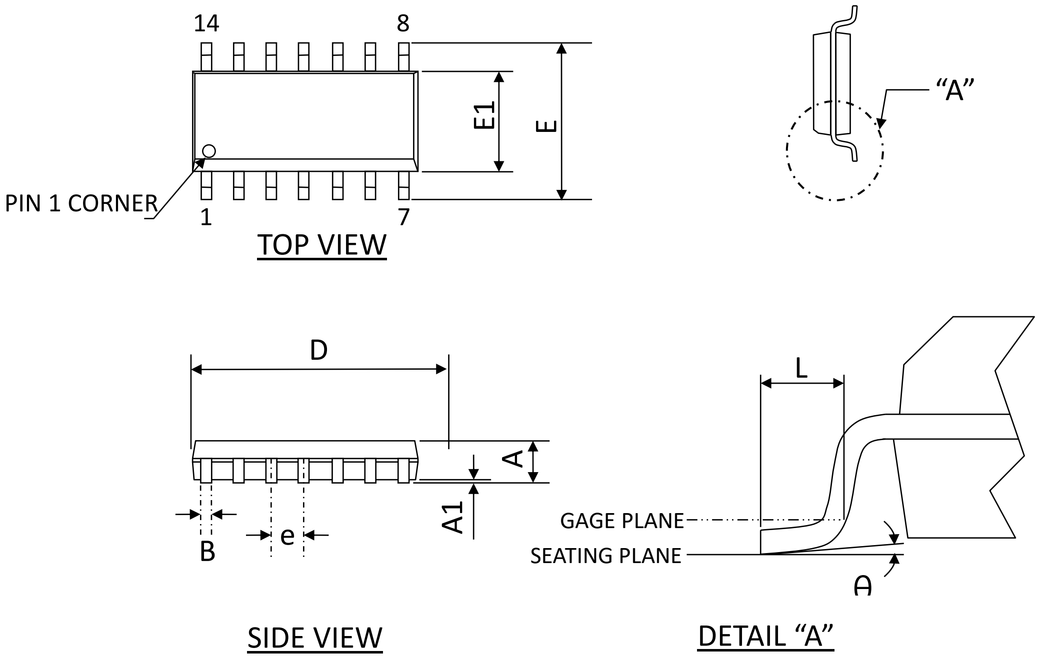




SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	5.33	--	--	0.210
A1	0.38	--	--	0.015	--	--
A2	3.18	3.30	3.43	0.125	0.130	0.135
B	0.46 typ.			0.018 typ.		
B1	1.52 typ.			0.060 typ.		
D	18.67	19.05	19.69	0.735	0.750	0.775
E	7.62 BSC			0.300BSC		
E1	6.22	6.35	6.48	0.245	0.250	0.255
e1	2.54 typ.			0.100 typ.		
L	2.92	3.30	3.81	0.115	0.130	0.150
eA	7.62	9.02	9.53	0.300	0.355	0.375
θ	0°	7°	15°	0°	7°	15°

Notes：
1. JEDEC OUTLINE : MS-001 AA
2. CONTROLLING DIMENSION：inch

27.9 SOP14

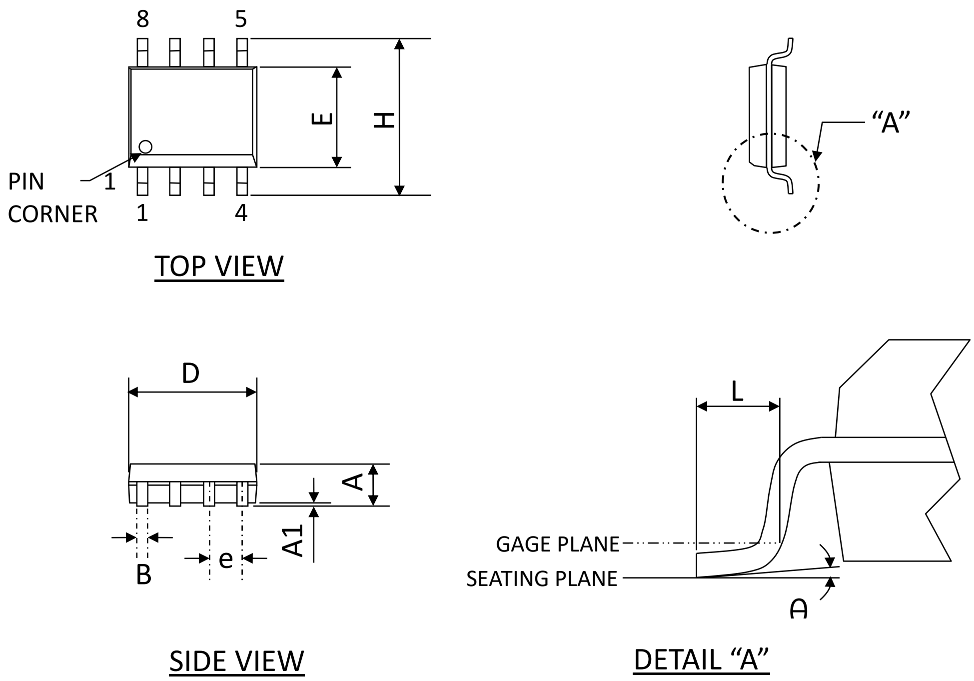


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.75	--	--	0.069
A1	0.05	--	0.25	0.002	--	0.010
B	0.31	--	0.51	0.012	--	0.02
D	8.65 BSC			0.340 BSC		
E1	3.90 BSC			0.154 BSC		
e	1.27 BSC			0.050 BSC		
E	6.00 BSC			0.236 BSC		
L	0.4	--	1.27	0.015	--	0.050
θ	0°	--	8°	0°	--	8°

Notes：

1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MS-012 AB

27.10 SOP8



SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.75	--	--	0.069
A1	0.10	--	0.25	0.004	--	0.010
B	0.31	--	0.51	0.012	--	0.020
D	4.90 BSC			0.193 BSC		
E	3.90 BSC			0.153 BSC		
e	1.27 BSC			0.050 BSC		
H	6.00 BSC			0.236 BSC		
L	0.40	--	1.27	0.016	--	0.050
θ	0°	--	8°	0°	--	8°

Notes：

1. CONTROLLING DIMENSION： mm
2. JEDEC OUTLINE : MS-012 AA

28附录：参考文档

SONiX 为用户提供了参考文档，可以更快地熟悉 SN8F5000 的性能。（从 SONIX 官网：www.sonix.com.tw 下载）。

文档名称	文档说明
SN8F5000 Starter-Kit User Manual	该文档介绍 SN8F5000 系列的 Starter-Kit，帮用户先选择合适的 Starter-Kit 进行产品开发。
SN8F5000 Family Instruction Set	该文档详细介绍 8051 的指令集，并进行简单示例说明。
SN8F5000 Family Instruction Mapping Table	该文档介绍 8-bit Flash/OTP Type 与 8051 Flash Type 相对应的指令。
SN8F5000 Package Information	该文档介绍 SN8F5000 系列单片机的机械数据，如高度、宽度和倾斜度等信息。
SN8F5000 Debug Tool Manual	该文档教用户按照 Keil C51 软件，并生产一个新的工程用于产品开发。

SN8F5713 Series Datasheet

8051-based Microcontroller

总公司
台湾新竹县竹北市台元街 36 号 10 楼之一（台元科技园区）
TEL: +886-3-5600888
FAX: +886-3-5600889

台北销售处
台北市松德路 171 号 15 楼 2
TEL: +886-2-27591980
FAX: +886-2-27598180
mkt@sonix.com.tw
sales@sonix.com.tw

香港办事处
香港新界沙田火炭禾盛街 11 号，中建电讯大厦 26 楼 03 室
TEL: +852-2723-8086
FAX: +852-2723-9179
hk@sonix.com.tw

松翰（深圳）
中国广东省深圳市宝安区
TEL: +86-755-2671-9666
FAX: +86-755-2671-9786
mkt@sonix.com.tw
sales@sonix.com.tw

USA 办事处
TEL: +1-714-3309877
TEL: +1-949-4686539
tlightbody@earthlink.net

日本办事处
日本东京千代田区九段南 4 丁目-8-27 二楼
TEL: +81-3-6272-6070
FAX: +81-3-6272-6165
jpsales@sonix.com.tw

FAE 邮箱地址
8 位单片机产品：
sa1fae@sonix.com.tw
所有产品：fae@sonix.com.tw