SONiX Technology Co., Ltd.

SN8F5702A Series Datasheet

8051-based Microcontroller

SN8F5702AP/S/TG
SN8F5702AJG
SN8F570210ASG
SN8F570211ASG
SN8F570213ASG
SN8F570212AS/TG
SN8F570200AAG
SN8F570202ASG

1 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 灵活的 CPU 频率
  - 内部 32MHz 时钟发生器（IHRC）
- 4 KB Flash 存储器（IROM），支持 In-System Program 功能
- 256 字节内部 RAM（IRAM）
- 13 个中断源：可控制中断的优先等级以及独立的中断向量
  - 12 个内部中断
  - 1 个外部中断：INT0
- 1 组 DPTR
- 2 组 8/16 位定时器，有四种工作模式。
- 1 组 16 位定时器，带有 4 路比较输出（PWM）和捕获通道。
- 1 组 16 位 PWM 发生器
  - 每组 PWM 都有 4 个输出通道，带有反相器和死区控制功能
- 10+1 通道 12 位 SAR ADC
  - 10 个外部 ADC 输入
  - 1 个内部电池检测
  - 4 个内部参考电压（VDD/4V/3V/2V）
- SPI/UART/I2C 接口，支持 SMBus
- 工作模式
  - NORMAL 模式
  - IDLE 模式
  - STOP 模式
- 片上调试
  - 单线调试接口
  - 2 个硬件断点
  - 无限制软件断点
  - ROM 数据安全保护
- 看门狗和可编程的外部复位
- 1.8V 低电压检测
- 工作电压范围大（1.8 V – 5.5 V），温度范围为-40 °C 到 85 °C

1.2 应用领域

- 无刷直流电机
- 家用自动化产品
- 家电
- 其它

1.3 产品性能表

	I/O	PWM Channels	I2C	SPI	UART	ADC ext. Channels	OPA	CMP	Ext. INT	Package Types
SN8F5702A	18	8	V	V	V	10	-	-	1	DIP20，SOP20 TSSOP20，QFN20
SN8F570212A	14	6	V	V	TX*(1)	8	-	-	1	SOP16，TSSOP16
SN8F570210A	12	5	-	-	V	6	-	-	1	SOP14
SN8F570211A	12	5	V	-	TX*(1)	6	-	-	1	SOP14
SN8F570213A	12	5	-	-	V	6	-	-	1	SOP14
SN8F570200A	8	3	-	-	TX*(1)	5	-	-	1	MSOP10
SN8F570202A	6	3	-	-	TX*(1)	4	-	-	1	SOP8

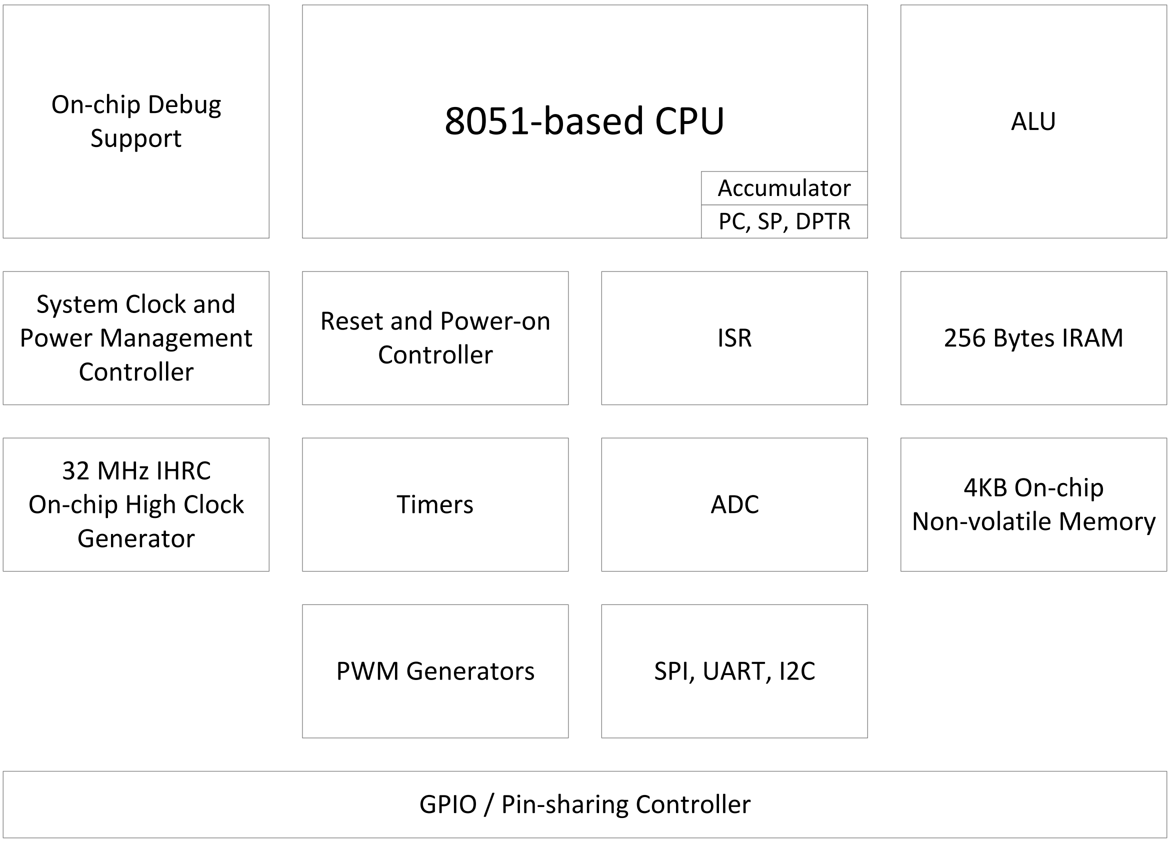
*(1)只支持 UART TX 模式。

1.4 SN8F5702 移植到 SN8F5702A

Item	SN8F5702	SN8F5702A
电气特性	1. STOP 模式功耗：4uA-9uA 2. IO 灌电流：20mA@5V 3. ADC 内部参考精度：±2.0%	1. STOP 模式功耗：1.6uA-4uA 2. IO 灌电流：35mA@5V 3. ADC 内部参考精度：±0.5%~±1.0%
T0	1. T0 时钟源 fosc 2. T0 不能唤醒 STOP 模式	1. T0 时钟源 fosc 或 flosc 2. T0 能唤醒 STOP 模式 (T0 时钟源 flosc 和 STWK=1)
系统寄存器	1. Original	1. 增加 SYSMOD 寄存器

- **SN8F5702A 兼容 SN8F5702。**
- **SN8F5702 的代码可以直接移植到 SN8F5702A 。 SN8F5702 的十六进制源码可直接修改为 SN8F5702A，无需再 IDE 中重新编译。**

1.5 结构框图


On-chip Debug Support
8051-based CPU
Accumulator
PC, SP, DPTR
ALU
System Clock and Power Management Controller
Reset and Power-on Controller
ISR
256 Bytes IRAM
32 MHz IHRC On-chip High Clock Generator
Timers
ADC
4KB On-chip Non-volatile Memory
PWM Generators
SPI, UART, I2C
GPIO / Pin-sharing Controller

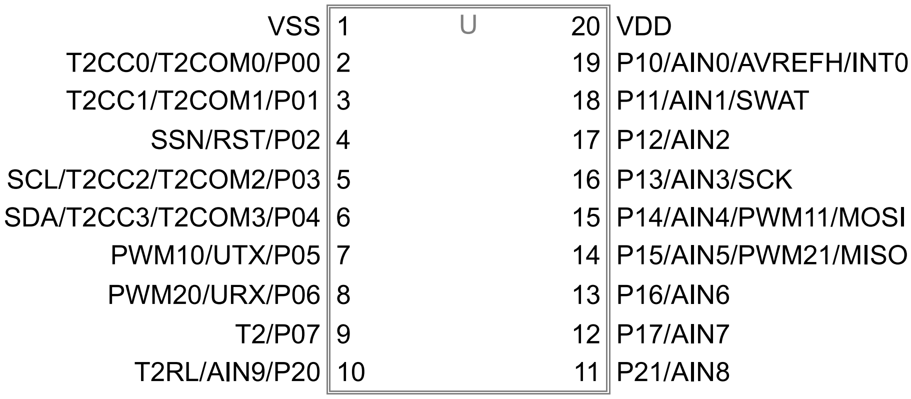
2 目录

3 修订记录

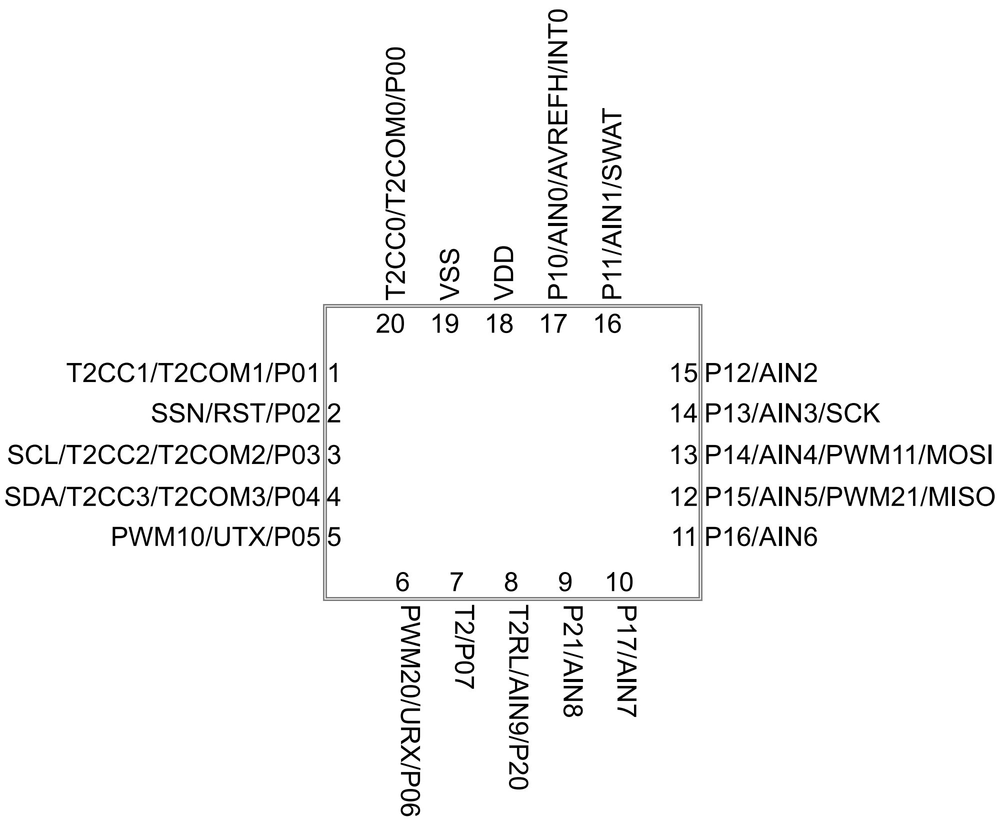
版本	时间	修订说明
1.0	Nov. 2020	初版。
1.1	Sep. 2022	修改唤醒时间和 ADC VREFH 寄存器描述。
1.2	Sep. 2023	增加 SN8F5702A 系列引脚描述。
1.3	Jun.2024	1. 增加引脚配置和 GPIO 部分未封出引脚的注释； 2. 增加 ISP 操作的注释； 3. 修改电气特性章节； 4. 修改特性表中内存描述； 5. 修改订购信息； 6. 修改书写错误； 7. 移除示例代码。
1.4	Oct.2024	1. 修改 GPIO 寄存器描述； 2. 修改 PWM 占空比描述。
1.5	Dec.2024	1. 修改封装信息。

4 引脚配置

4.1 SN8F5702AP/S/TG (DIP20/SOP20/TSSOP20)



4.2 SN8F5702AJG (QFN20)



✸ *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。*

4.3 SN8F570210ASG (SOP14)

VSS	1	14	VDD
T2CC0/T2COM0/P00	2	13	P10/AIN0/INT0/AVREFH
T2CC1/T2COM1/P01	3	12	P11/AIN1/SWAT
SSN/RST/P02	4	11	P12/AIN2
PWM10/UTX/P05	5	10	P13/AIN3/SCK
PWM20/URX/P06	6	9	P14/AIN4/PWM11/MOSI
T2/P07	7	8	P20/AIN9/T2RL

✸ *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。*

4.4 SN8F570211ASG (SOP14)

VSS	1	14	VDD
T2CC0/T2COM0/P00	2	13	P10/AIN0/INT0/AVREFH
SSN/RST/P02	3	12	P11/AIN1/SWAT
SCL/T2CC2/T2COM2/P03	4	11	P12/AIN2
SDA/T2CC3/T2COM3/P04	5	10	P13/AIN3/SCK
PWM10/UTX/P05	6	9	P14/AIN4/PWM11/MOSI
T2/P07	7	8	P20/AIN9/T2RL

✸ *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。*

4.5 SN8F570213ASG (SOP14)

VDD	1	14	VSS
T2CC0/T2COM0/P00	2	13	P10/AIN0/INT0/AVREFH
T2CC1/T2COM1/P01	3	12	P11/AIN1/SWAT
SSN/RST/P02	4	11	P12/AIN2
PWM10/UTX/P05	5	10	P13/AIN3/SCK
PWM20/URX/P06	6	9	P14/AIN4/PWM11/MOSI
T2/P07	7	8	P20/AIN9/T2RL

> ✸ *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。*

4.6 SN8F570212AS/TG (SOP16/TSSOP16)

VSS	1	16	VDD
T2CC0/T2COM0/P00	2	15	P10/AIN0/AVREFH/INT0
SSN/RST/P02	3	14	P11/AIN1/SWAT
SCL/T2CC2/T2COM2/P03	4	13	P12/AIN2
SDA/T2CC3/T2COM3/P04	5	12	P13/AIN3/SCK
PWM10/UTX/P05	6	11	P14/AIN4/PWM11/MOSI
T2/P07	7	10	P15/AIN5/PWM21/MISO
T2RL/AIN9/P20	8	9	P16/AIN6

> ✸ *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。*

4.7 SN8F570200AAG (MSOP10)

VDD	1	U	14	P10/AIN0/AVREFH/INT0
VSS	2		13	P11/AIN1/SWAT
T2CC0/T2COM0/P00	3		12	P12/AIN2
SSN/RST/P02	4		11	P13/AIN3/SCK
PWM10/UTX/P05	7		8	P14/AIN4/PWM11/MOSI

✸ 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。

4.8 SN8F570202ASG (SOP8)

VDD	1	U	14	VSS
T2CC0/T2COM0/P00	2		13	P12/AIN2
PWM10/UTX/P05	3		12	P14/AIN4/PWM11/MOSI
AIN0/AVREFH/INT0/P10	7		8	P11/AIN1/SWAT

✸ 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。

4.9 引脚说明

电源引脚

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

P0

引脚名称	类型	功能说明
P0.0 T2COM0 T2CC0	Digital I/O Digital Output Digital Input	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 Timer2：比较器 0 输出。 Timer2：捕获 0 输入。
P0.1 T2COM1 T2CC1	Digital I/O Digital Output Digital Input	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 Timer2：比较器 1 输出。 Timer2：捕获 1 输入。
P0.2 Reset SSN	Digital I/O Digital Input Digital Input	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 系统复位（低电平有效）。 SPI：从动选择引脚（从动模式）。
P0.3 T2COM2 T2CC2 SCL	Digital I/O Digital Output Digital Input Digital I/O	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 Timer2：比较器 2 输出。 Timer2：捕获 2 输入。 I2C：时钟输出（主机模式）时钟输入（从机模式）。
P0.4 T2COM3 T2CC3 SDA	Digital I/O Digital Output Digital Input Digital I/O	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 Timer2：比较器 3 输出。 Timer2：捕获 3 输入。 I2C：数据引脚。
P0.5 UTX PWM10	Digital I/O Digital Output Digital Output	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 UART：数据输出引脚。 PWM：可编程的 PWM 输出。
P0.6 URX PWM20	Digital I/O Digital Input Digital Output	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 UART：数据输入引脚。 PWM：可编程的 PWM 输出。
P0.7 T2	Digital I/O Digital Input	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 Timer2：事件计数器输入引脚。

P1

引脚名称	类型	功能说明
P1.0 AIN0 INT0 AVREFH	Digital I/O Analog Input Digital Input Analog Input	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 ADC：输入通道 0。 INT0：外部中断 0。 ADC：外部高参考电压。
P1.1 AIN1 SWAT	Digital I/O Analog Input Digital I/O	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 ADC：输入通道 1。 单线仿真接口。
P1.2 AIN2	Digital I/O Analog Input	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 ADC：输入通道 2。
P1.3 AIN3 SCK	Digital I/O Analog Input Digital Output	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 ADC：输入通道 3。 SPI：时钟输出（主机模式）时钟输入（从机模式）。
P1.4 AIN4 MOSI PWM11	Digital I/O Analog Input Digital I/O Digital Output	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 ADC：输入通道 4。 SPI：发送引脚（主机模式）接收引脚（从机模式）。 PWM：可编程的 PWM 输出。
P1.5 AIN5 MISO PWM21	Digital I/O Analog Input Digital I/O Digital Output	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 ADC：输入通道 5。 SPI：接收引脚（主机模式）发送引脚（从机模式）。 PWM：可编程的 PWM 输出。
P1.6 AIN6	Digital I/O Analog Input	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 ADC：输入通道 6。
P1.7 AIN7	Digital I/O Analog Input	GPIO：双向引脚，施密特触发，内置上拉电阻，具有唤醒功能。 ADC：输入通道 7。

P2

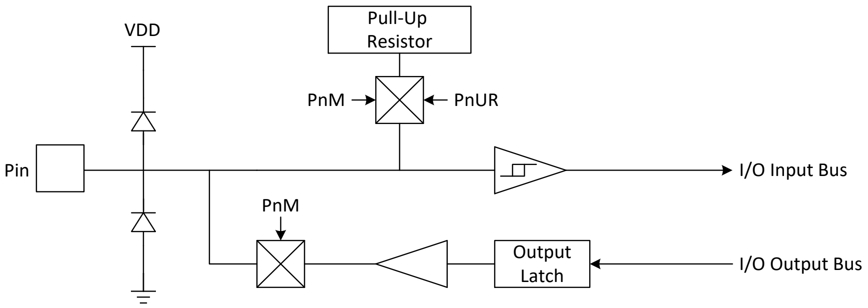
引脚名称	类型	功能说明
P2.0 AIN9 T2RL	Digital I/O Analog Input Digital Input	GPIO：双向引脚，施密特触发，内置上拉电阻。 ADC：输入通道 9。 Timer2：重装触发输入引脚。
P2.1 AIN8	Digital I/O Analog Input	GPIO：双向引脚，施密特触发，内置上拉电阻。 ADC：输入通道 8。

4.10 引脚特性

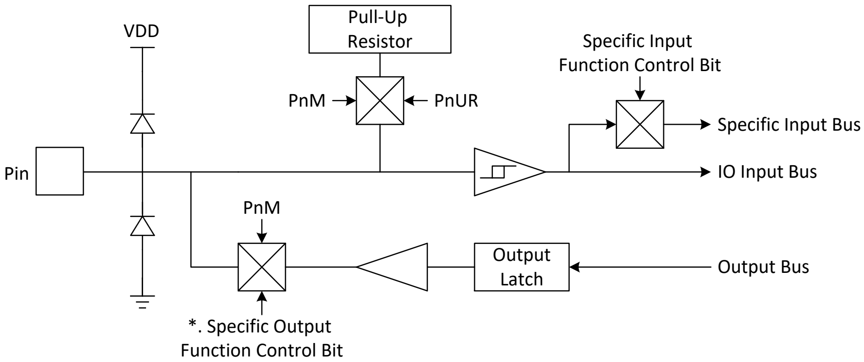
Port	Open-Drain	Sink Current 100mA VSS+1.5V	Sink Current 20mA VSS+0.5V	External Interrupt	Wakeup (Level change)	Shared Pin
P0.0	-	V	-	-	V	T2CC0/T2COM0
P0.1	-	V	-	-	V	T2CC1/T2COM1
P0.2	-	V	-	-	V	SSN/RST
P0.3	-	V	-	-	V	SCL/T2CC2/T2COM2
P0.4	-	V	-	-	V	SDA/T2CC3/T2COM3
P0.5	V	V	-	-	V	PWM10/UTX
P0.6	V	V	-	-	V	PWM20/URX
P0.7	-	-	V	-	V	T2
P1.0	-	-	V	V	V	AIN0/AVREFH/INT0
P1.1	-	-	V	-	V	AIN1/SWAT
P1.2	-	-	V	-	V	AIN2
P1.3	V	-	V	-	V	AIN3/SCK
P1.4	V	-	V	-	V	AIN4/PWM11/MOSI
P1.5	V	-	V	-	V	AIN5/PWM21/MISO
P1.6	-	-	V	-	V	AIN6
P1.7	-	-	V	-	V	AIN7
P2.0	-	-	V	-	-	T2RL/AIN9
P2.1	-	-	V	-	-	AIN8

4.11 引脚电路图

普通双向 I/O 引脚。

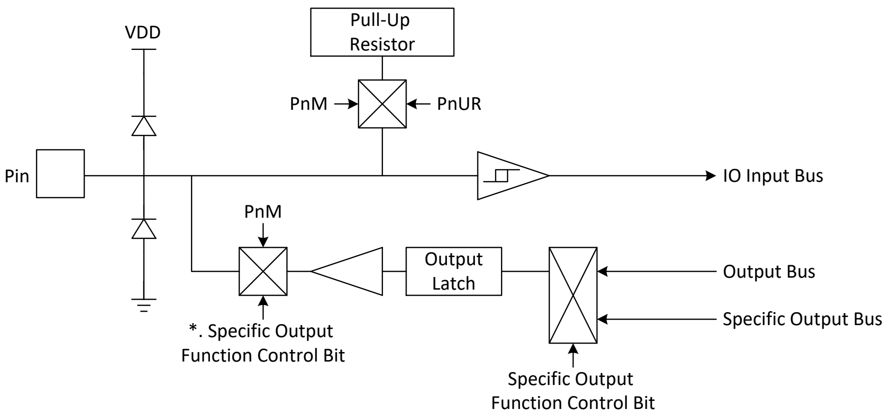


双向 I/O 引脚共用特殊数字输入功能，例如 INT0、事件计数器、SIO、UART。



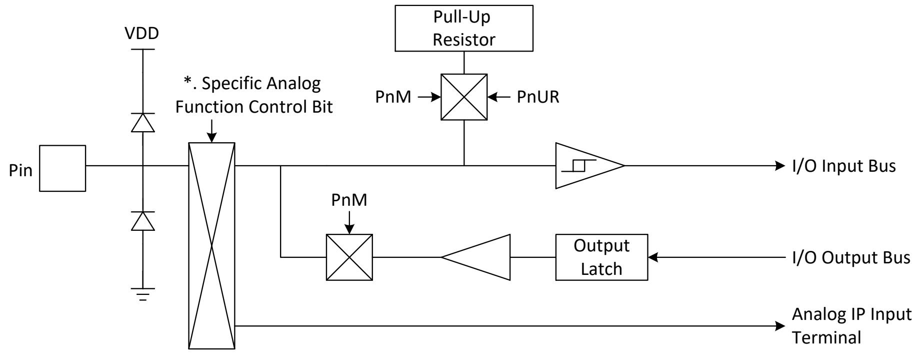
***. Some specific functions switch I/O direction directly, not through PnM register.**

双向 I/O 引脚共用特殊数字输出功能，例如 PWM、SIO、UART。



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

双向 I/O 引脚共用特殊模拟输入功能，例如 ADC。



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

5 CPU

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

5.1 存储器结构

SN8F5702A 内建了两个存储器：内部 RAM（IRAM）和程序存储器（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）。程序存储器是一个 4KB 的 FLASH 存储器，最快的存取速度可达 8MHz。

5.2 内部 RAM (IRAM)

256 x 8bit RAM(内部数据存储器)

地址	RAM 位置		
000h 01Fh	工作寄存器区		00h-7Fh 的 RAM 支持直接寻址和间接寻址
020h 02Fh	位可寻址区		
030h … … 07Fh	通用区		
080h … … … 0FFh	通用区 （直接寻址）	特殊功能寄存器 （直接寻址）	080h-0FFh 存储特殊功能寄存器 Bank 0 结束

内部数据存储器的 256 字节 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：

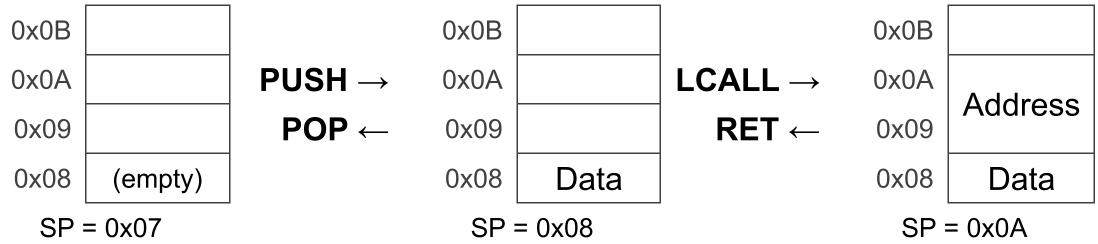
- 0x00-0x1F： 工作寄存器区包含 4 个 bank，每个 bank 有 8 个工作寄存器（R0-R7），由寄存器 PSW 的 RS0/RS1 位选择；
- 0x20-0x2F：位可寻址区域(地址)；

在位可寻址区域, 用户可以使用唯一的地址来读或写任何一个位。支持 16 字节的位可寻址存储区域，提供 128 个可寻址位。在 0x00 到 0x7F 区域内每个位都有单独的地址，因此都可直接位寻址。字节 0x20 内存的 BIT0 位地址为 0x00，BIT7 位地址为 0x07。字节 0x2F 内存的 BIT0 位地址为 0x78，BIT7 位地址为 0x7F。当设置“SETB 42H”，意思是设置字节 0x28 的 BIT2 位地址。

	Byte Address	Bite 0	Bite 1	Bite 2	Bite 3	Bite 4	Bite 5	Bite 6	Bite 7
Bit Addressable Area	**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 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 RAM 中的其他变量重叠。堆栈溢出或者下溢也可能会导致误写 RAM 中的其他变量，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 0x07，就是指堆栈的区域从 IRAM 地址中的 0x08 开始。换句话说，如果计划想将堆栈区域设置为从 IRAM 的 0xC0 开始，就要在系统复位后将 SP 寄存器设置为 0x0BF。

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。

✸ 注：堆栈与内部RAM(IRAM)共用相同的内存空间，当程序占用太多的IRAM空间而造成堆栈不够时 Keil C51编译器不会提示"错误"或"警告"，因此用户必须注意这种情况。

5.4 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（参考 In-System Program 章节）。

地址	ROM	注释
0000H	复位向量	复位向量
0001H 0002H	通用区域	用户程序
0003H	**INT0**	**中断向量**
000BH	**TIMER0**	
001BH	**TIMER1**	
0023H	**UART**	
002BH	**TIMER2**	
0043H	**I2C**	
004BH	**SPI**	
0053H	**T2COM0**	
005BH	**T2COM1**	
0063H	**T2COM2**	
006BH	**T2COM3**	
0083H	**PWM1**	
008BH	**ADC**	
008CH	通用区域	用户程序 用户程序结束
0FF6H 0FF7H . 0FFEH 0FFFH	保留	

ROM 包含复位向量，中断向量，通用区域和保留区域。复位向量是程序的起始地址，中断向量是当任何中断发生时的中断服务程序索引，通用区域是主程序区域包含主循环，子程序，数据表。

- 0x0000 复位向量：程序指针指向 0x0000 当发生复位事件后（上电复位，外部复位，看门狗复位，LVD 复位…）；
- 0x0001-0x0002：通用区域处理系统复位操作；
- 0x0003-0x008B：多中断向量区域，每个中断事件都有一个唯一的中断向量；
- 0x008C-0x0FDF：用户程序和 ISP 通用区域；
- **0x0FE0-0x0FF5：用户通用区域程序，禁止执行 ISP；**
- **0x0FF6-0x0FFF：保留区域，禁止执行 ISP；**

5.5 程序存储器安全

SN8F5702A 提供加密机制，防止 Flash 存储器资料被破解. 当使能加密功能时，ROM 代码受到保护，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。

5.6 数据指针

在执行 MOVC 指令时，数据指针可帮助指定 IROM 地址。该单片机有 1 组数据指针（DPH/DPL）。

5.7 堆栈和数据指针寄存器

Register	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

SP Register (0x81)

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

DPL Register (0x82)

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

DPH Register (0x83)

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

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	-	-	-	PFLAG
F0	B	P0UR	P1UR	P2UR	-	-	-	SRST
E8	-	-	-	-	-	-	-	-
E0	ACC	SPSTA	SPCON	SPDAT	P0OC	CLKSEL	CLKCMD	TCON0
D8	S0CON2	-	I2CDAT	I2CADR	I2CCON	I2CSTA	SMBSEL	SMBDST
D0	PSW	IEN4	ADM	ADB	ADR	VREFH	P1CON	-
C8	T2CON	-	CRCL	CRCH	TL2	TH2	-	SYSMOD
C0	IRCON	CCEN	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3
B8	IEN1	IP1	S0RELH	PW1DL	PW1DH	PW1A	PW1CH	IRCON2
B0	-	-	-	-	-	-	-	-
A8	IEN0	IP0	S0RELL	PW1M	PW1YL	PW1YH	PW1BL	PW1BH
A0	P2	-	-	-	-	-	-	-
98	S0CON	S0BUF	IEN2	-	-	P0CON	P2CON	-
90	P1	P1W	-	-	PECMD	PEROML	PEROMH	PERAM
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	PEDGE
80	P0	SP	DPL	DPH	-	-	WDTR	PCON

✸ *注：最左边列所有的特殊功能寄存器都是位可寻址（每个**0x0/0x8**结尾的**SFR**地址都是位可寻址）。*

6.2 特殊功能寄存器说明

0x80 - 0x9F 寄存器说明

Register	Address	Description
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	-
-	0x93	-
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	-
P0CON	0x9D	P0 配置控制寄存器。
P2CON	0x9E	P2 配置控制寄存器。
-	0x9F	-

0xA0 - 0xBF 寄存器说明

Register	Address	Description
P2	0xA0	P2 数据缓存器。
-	0xA1	-
-	0xA2	-
-	0xA3	-
-	0xA4	-
-	0xA5	-
-	0xA6	-
-	0xA7	-
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。
PW1M	0xAB	PW1 控制寄存器。
PW1YL	0xAC	PW1 周期控制缓存器低字节。
PW1YH	0xAD	PW1 周期控制缓存器高字节。
PW1BL	0xAE	PW1 B point 死区控制缓存器低字节。
PW1BH	0xAF	PW1 B point 死区控制缓存器高字节。
-	0xB0	-
-	0xB1	-
-	0xB2	-
-	0xB3	-
-	0xB4	-
-	0xB5	-
-	0xB6	-
-	0xB7	-
IEN1	0xB8	中断使能寄存器。
IP1	0xB9	中断优先权寄存器。
S0RELH	0xBA	UART 重装高字节寄存器。
PW1DL	0xBB	PW1 占空比控制寄存器低字节。
PW1DH	0xBC	PW1 占空比控制寄存器高字节。
PW1A	0xBD	PW1 A point 死区控制缓存器。
PW1CH	0xBE	PW1 通道控制缓存器。
IRCON2	0xBF	中断请求寄存器。

0xC0 - 0xDF 寄存器说明

Register	Address	Description
IRCON	0xC0	中断请求寄存器。
CCEN	0xC1	T2 比较器/捕捉功能使能寄存器。
CCL1	0xC2	T2 比较器/捕捉功能模块 1 低字节寄存器。
CCH1	0xC3	T2 比较器/捕捉功能模块 1 高字节寄存器。
CCL2	0xC4	T2 比较器/捕捉功能模块 2 低字节寄存器。
CCH2	0xC5	T2 比较器/捕捉功能模块 2 高字节寄存器。
CCL3	0xC6	T2 比较器/捕捉功能模块 3 低字节寄存器。
CCH3	0xC7	T2 比较器/捕捉功能模块 3 高字节寄存器。
T2CON	0xC8	T2 控制寄存器。
-	0xC9	-
CRCL	0xCA	T2 比较器/捕捉功能模块 0＆重装功能低字节寄存器。
CRCH	0xCB	T2 比较器/捕捉功能模块 0＆重装功能高字节寄存器。
TL2	0xCC	T2 计数低字节寄存器。
TH2	0xCD	T2 计数高字节寄存器。
-	0xCE	-
SYSMOD	0xCF	系统控制寄存器。
PSW	0xD0	系统标志寄存器。
IEN4	0xD1	中断数据寄存器。
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。
P1CON	0xD6	P1 配置控制寄存器。
-	0xD7	-
S0CON2	0xD8	UART 波特率控制寄存器。
-	0xD9	-
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

Register	Address	Description
ACC	0xE0	ACC 寄存器。
SPSTA	0xE1	SPI 状态寄存器。
SPCON	0xE2	SPI 控制寄存器。
SPDAT	0xE3	SPI 数据缓存器。
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	-
-	0xE9	-
-	0xEA	-
-	0xEB	-
-	0xEC	-
-	0xED	-
-	0xEE	-
-	0xEF	-
B	0xF0	乘法/除法指令数据缓存器。
P0UR	0xF1	P0 上拉电阻控制寄存器。
P1UR	0xF2	P1 上拉电阻控制寄存器。
P2UR	0xF3	P2 上拉电阻控制寄存器。
-	0xF4	-
-	0xF5	-
-	0xF6	-
SRST	0xF7	软件复位控制寄存器。
-	0xF8	-
P0M	0xF9	P0 输入/输出模式寄存器。
P1M	0xFA	P1 输入/输出模式寄存器。
P2M	0xFB	P2 输入/输出模式寄存器。
-	0xFC	-
-	0xFD	-
-	0xFE	-
PFLAG	0xFF	复位标志寄存器。

6.3 系统寄存器

Register	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	Description
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）或者由借位（C 或 AC），以及相等情况（P）发生时，该标志位会在 PSW 寄存器中进行设置。

B Register (0xF0)

Bit	Field	Type	Initial	Description
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 scratch-pad 寄存器来保留临时数据。

PSW Register (0xD0)

Bit	Field	Type	Initial	Description
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 操作时没有从 ACC 的第三位开始执行； 1：BCD 操作时从 ACC 的第三位开始执行。
5	F0	R/W	0	通用标志位 0，用户可用。
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（Bank0）； 01：08H-0FH（Bank1）； 10：10H-17H（Bank2）； 11：18H-1FH（Bank3）。
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，用户可用。
0	P	R	0	奇偶标志位。反应 ACC 中 1 的个数。 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

SN8F5702A 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 recongnize the 8051-specific predifined special registers.
#include <SN8F5702A.H>
```

使用 C 语言进行编程是，增加下面的句子：

```
#include <SN8F5702A.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
SN8F5702A	SN8F5702A.h	OPTIONS_SN8F5702A.A51
SN8F570210A	SN8F570210A.h	
SN8F570211A	SN8F570211A.h	
SN8F570212A	SN8F570212A.h	
SN8F570213A	SN8F570213A.h	
SN8F570200A	SN8F570200A.h	
SN8F570202A	SN8F570202A.h	

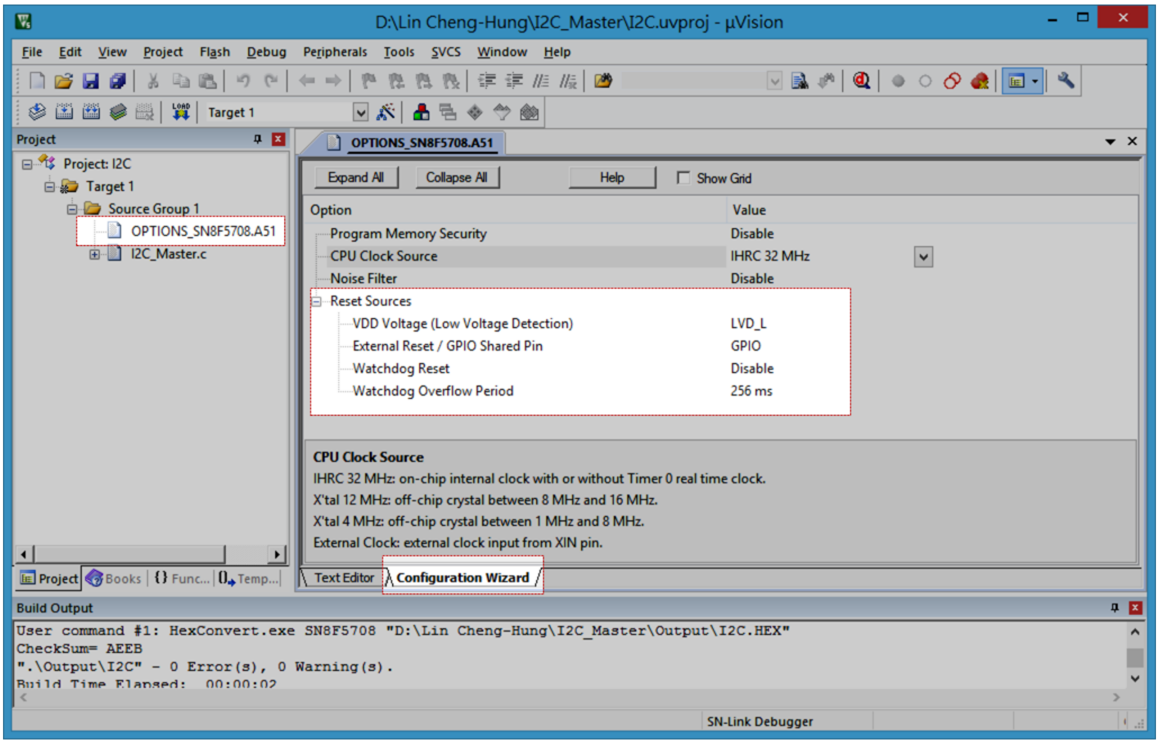
7 复位和上电控制

复位和上电控制有以下 4 种方式：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，并使程序从复位向量（ROM 地址 0x0000）处重新开始执行。

7.1 复位配置和上电控制

SONiX 发布了一个 SN8F5702A_OPTIONS.A51 文件，该文件包含在 SN8F5702A 软件包中（从松翰官网 www.sonix.com.tw 下载）。该文件包含了复位源和 CPU 时钟源选择的合适参数，强烈建议将这个文件加到 Keil 项目中。SN8F5000 的调试工具手册提供了更详细的内容。

- 程序内存加密
- CPU 时钟源
- Noise Filter
- 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

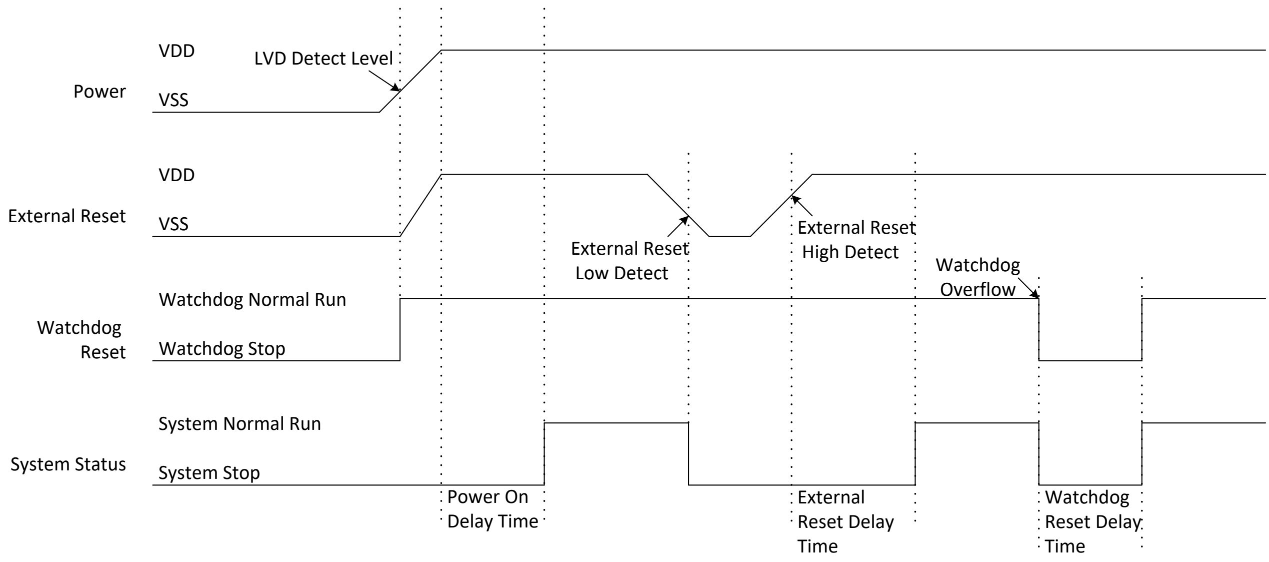


Code option 是系统硬件配置包含：晶振类型，Noise Filter 选项，看门狗时间操作，LVD 选项，复位引脚选项和 Flash ROM 加密控制。

Code option	选项	功能描述
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
CPU Clock Source	IHRC 32MHz	内部高速 32M RC
LVD	LVD_L	如果 VDD 低于 1.8V，LVD 复位
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，在复位信号结束之后开始上电流程，上电完成之后开始运行程序。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。

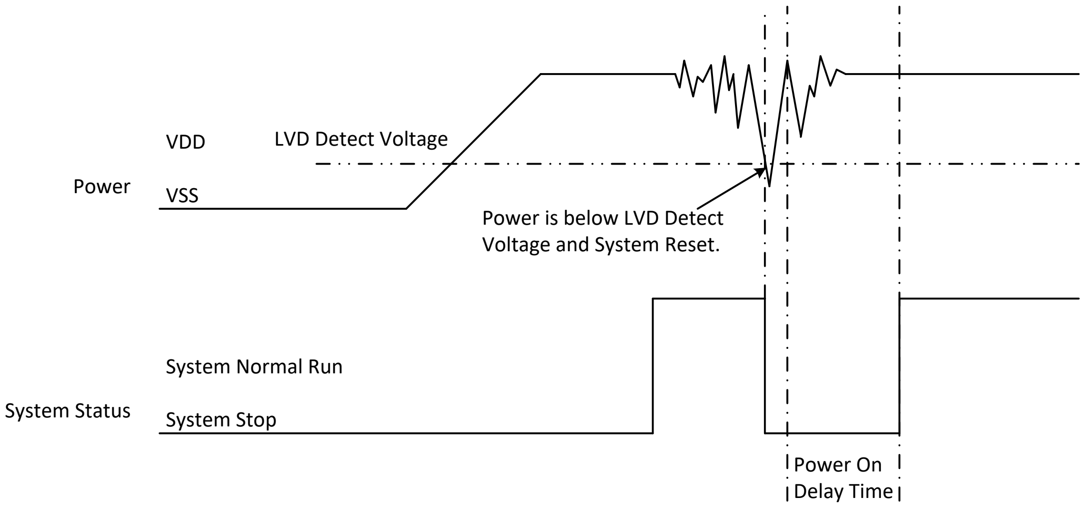


一般情况下，电源稳定期花费 4.6ms，之后单片机会自动获取 CPU 时钟源的配置。所选择的时钟源驱动起来之后，系统会计数 2048 个时钟周期以及 5 个内部低速振荡时钟以确保时钟已稳定。

> ***✱ 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，只有 1 个电压点：1.8V。根据 LVD 的配置，比较结果可作为系统的复位信号。下表显示了 LVD 配置，LVD_L 以及 VDD 引脚状态结果。



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 0x5A 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=256*1/(内部低速振荡器频率/WDT 前置频率)

=256/(FILRC/WDT 前置频率)…SEC

内部低速振荡器	WDT 前置频率	看门狗间隔时间
F_{ILRC}=16 kHz	F_{ILRC} /4	256/(16000/4)=64ms
	F_{ILRC} /8	256/(16000/8)=128ms
	F_{ILRC} /16	256/(16000/16)=256ms
	F_{ILRC} /32	256/(16000/32)=512ms

看门狗的工作模式可在选项文件中配置：

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。

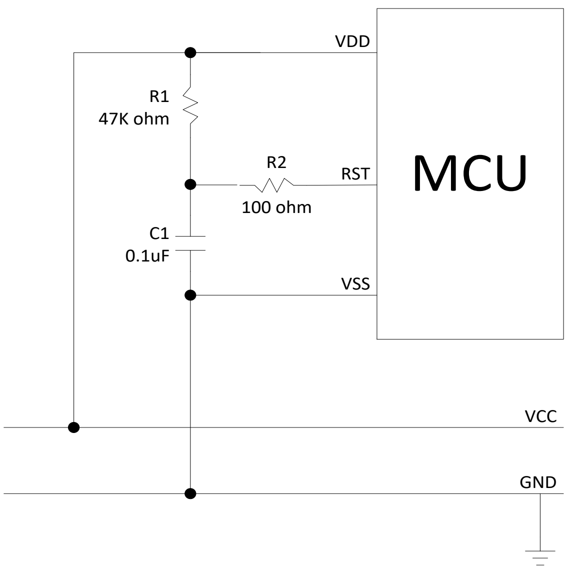
Disable Mode：其内部定时器在 CPU 的所有工作模式下都不计数，在这种模式下不会触发看门狗复位。

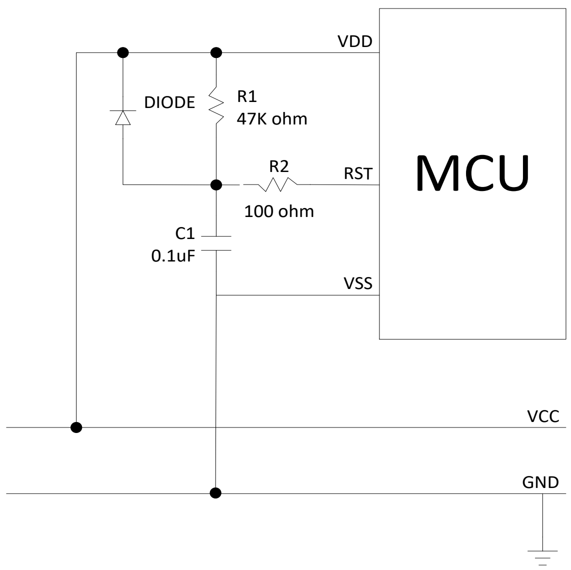
看门狗的工作模式为 Always on 时，系统的功耗较大。

7.5 外部复位引脚

可在选项文件中配置可编程的外部复位引脚，使能外部复位引脚后，系统会监控该复用引脚的逻辑电平，低电平（小于 30％VDD）会立即触发系统复位，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，该过程可使软件有权限去复位单片机（如更新软件之后的复位）。下面这段 C 程序代码通过重复设置 SRST 寄存器的最低位来实现软件复位。

```
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 Register

Bit	Field	Type	Initial	Description
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..3	Reserved	R	0	
0	Reserved	R	0	

SRST Register

Bit	Field	Type	Initial	Description
7..1	Reserved	R	0	
0	SRSTREQ	R/W	0	读：单片机软件复位时该位自动置起 写：连续设置该位 2 次触发软件复位。

WDTR Register (0x86)

Bit	Field	Type	Initial	Description
7..0	WDTR[7:0]	W	-	WDTR 寄存器控制清空看门狗。将 0x5A 数据移入 WDTR，将触发看门狗计时器。

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。

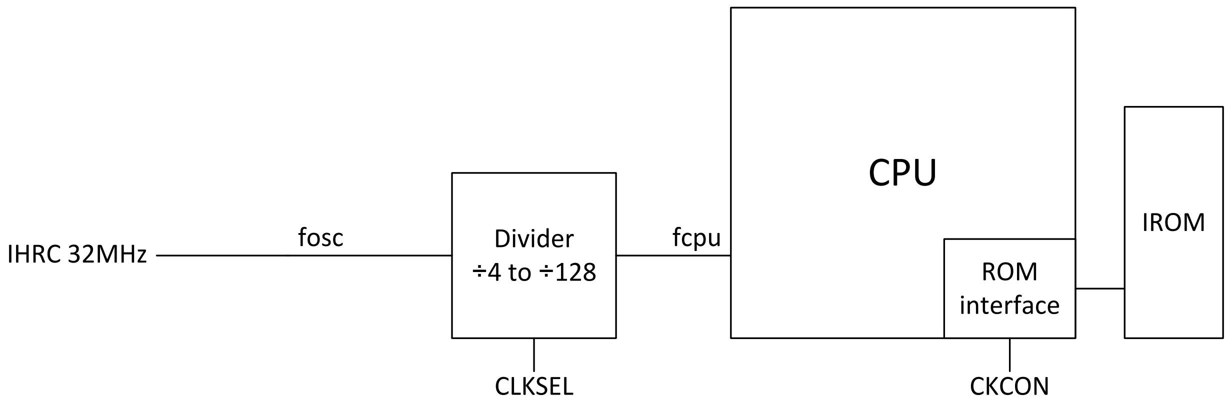
NORMAL 模式是指 CPU 和外设功能都正常工作，系统时钟由所选择的时钟源、时钟分频数以及程序存储器等待状态来共同决定的。IDLE 模式是指 CPU 时钟以及程序运行都暂停了的状态，但保留了外设功能（如定时器，PWM，SPI，UART 和 I2C）。与之相反的是，STOP 模式则禁止所有功能和时钟发生器，直至唤醒信号将系统唤醒进入 NORMAL 模式。

8.1 系统时钟

该单片机包括内置时钟发生器（IHRC 32MHz）。在复位和上电过程中，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 输入分频之后的时钟作为它的时钟频率（称作 Fcpu）。写入 0x69 到 CLKCMD 寄存器时设置 CLKSEL。

```
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。例如：CPU 计划运行在 32MHz，而 IROM 只能运行在 8MHz 以下，则在 CKCON 寄存器中必须设置 ROM 读取周期为增加 3 个以上的 Fcpu。

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。

Code Option CPU 时钟源	Fcpu = CLKSEL[2:0]	IROM Fetch = CKCON[6:4]
IHRC 32M	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

8.2 高速时钟和实时时钟

高速时钟仅内部时钟。内部高速振荡器是 32MHz RC 类型。这些高速振荡器由 SN8F5702_OPTIONS.A51 选择。

- IHRC 32M：系统高速时钟源是内部高速 32MHz RC 型振荡器。

8.3 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。此时，CPU 以及所有的外设都正常工作（称之为 NORMAL 模式）。

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 时钟源仍使能。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）仍然正常工作。P0/P1 输入的任何改变和中断事件都会导致单片机返回到 NORMAL 模式，IDLE 位自动清零。

- IDLE 模式下所有功能可以工作，仅 CPU 暂停。
- IDLE 模式的唤醒源为 P0 / P1 电平改变以及任何中断事件。

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继续执行，STOP 位自动清零。

- CPU、外设功能、时钟发生器均暂停。
- STOP 模式唤醒源为 P0 / P1 电平变化。

对于使用 C 语言编程的用户，强烈建议 IDEL 和 STOP 宏控制单片机的系统模式，而不是直接设置 IDLE 和 STOP 位。

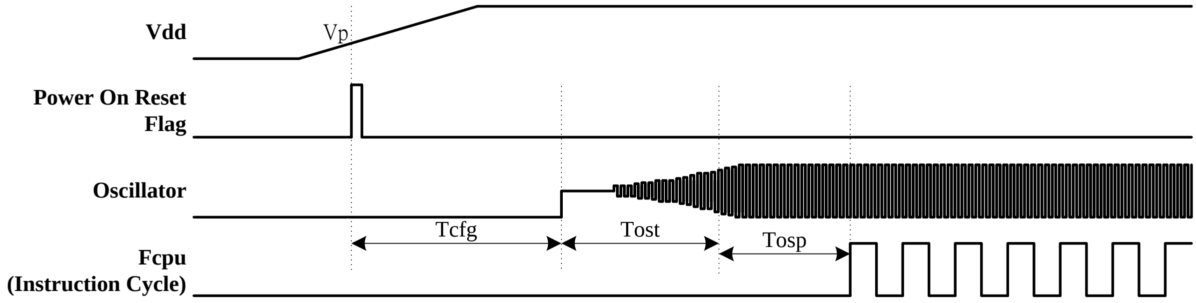
```
IDLE();
STOP();
```

✸	***注：通过汇编语言进入 IDLE 模式或 STOP 模式必须使用 MOV 语句。***

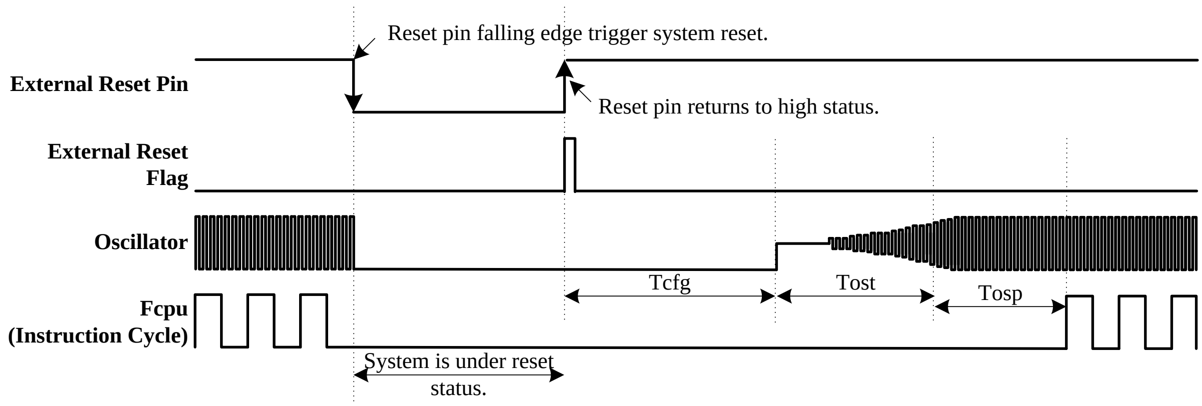
8.4 系统时间计时

参数	符号	说明	典型值
硬件配置时间	Tcfg	8*FILRC + 2^{17}*FIHRC	4.6ms @ F_{ILRC} = 16KHz & F_{IHRC} = 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）	377us @ F_{hosc} = 32MHz
		掉电模式唤醒条件下的振荡器预热时间。 64*Fhosc+5*FILRC……RC 型振荡器，例如内部高速 RC 型振荡器。	RC： 315us @ F_{hosc} = 32MHz

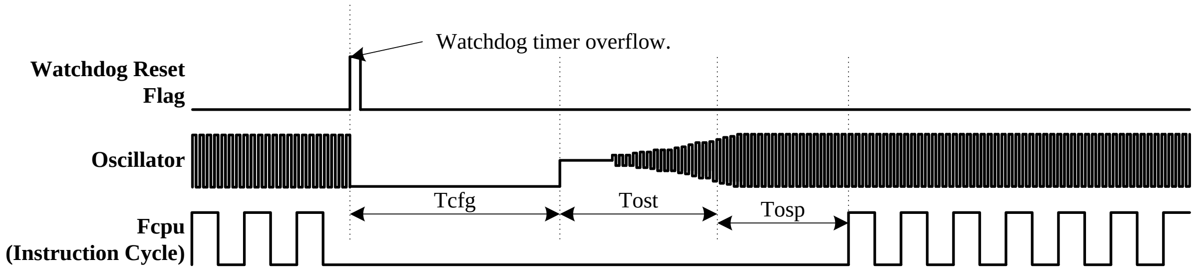
- 上电复位计时



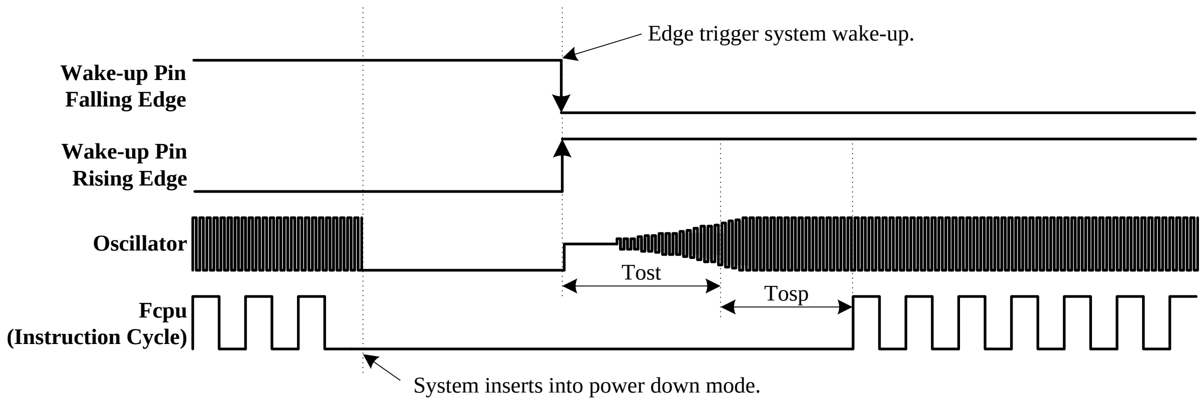
- 外部复位引脚复位计时



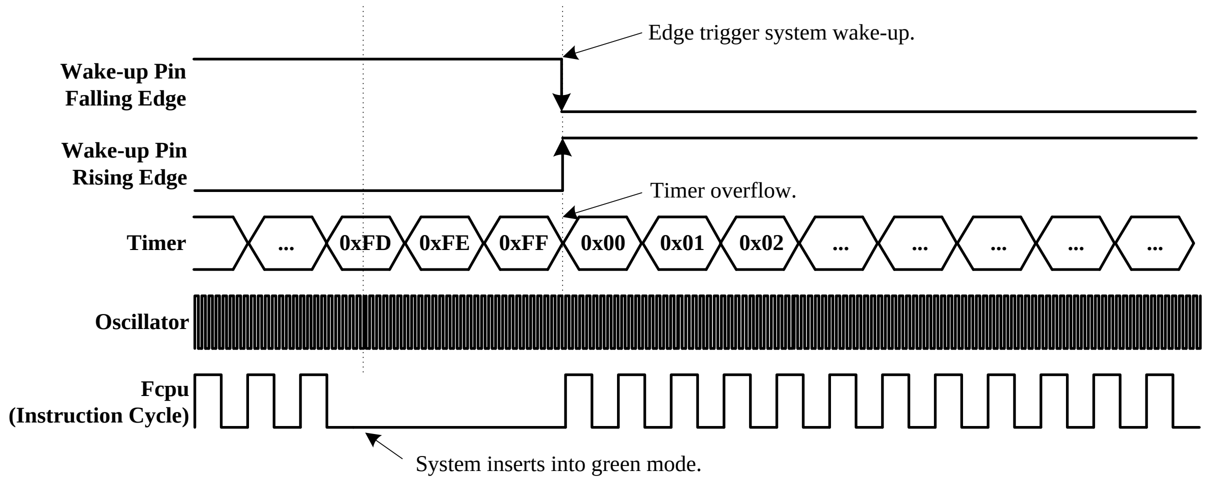
- 看门狗复位计时



- STOP 模式唤醒计时

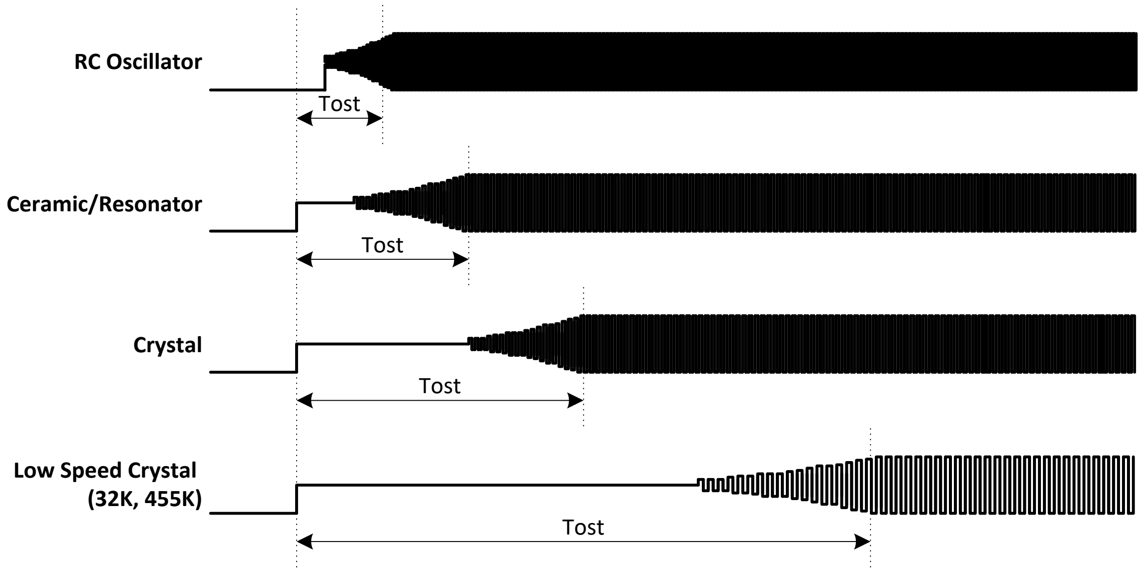


- IDLE 模式唤醒计时



● 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。



8.5 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	ESYN	EWSC2	EWSC1	EWSC0
CLKSEL	-	-	-	-	-	CLKSEL2	CLKSEL1	CLKSEL0
CLKCMD	CMD7	CMD6	CMD5	CMD4	CMD3	CMD2	CMD1	CMD0
PCON	SMOD	-	-	-	-	GF0	STOP	IDLE
P1W	P17W	P16W	P15W	P14W	P13W	P12W	P11W	P10W
SYSMOD	-	-	-	-	-	-	T0LCK	STWK

CKCON Register (0x8E)

Bit	Field	Type	Initial	Description
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。
Else	Reserved	R	0001	

CLKSEL Register (0xE5)

Bit	Field	Type	Initial	Description
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； 其它：保留。

CLKCMD Register (0xE6)

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

PCON Register (0x87)

Bit	Field	Type	Initial	Description
7				参考其它章节。
6..3	Reserved	R	0x00	
2	GF0	R/W	0	通用标志位。
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。

P1W Register (0x91)

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

SYSMOD Register (0xCF)

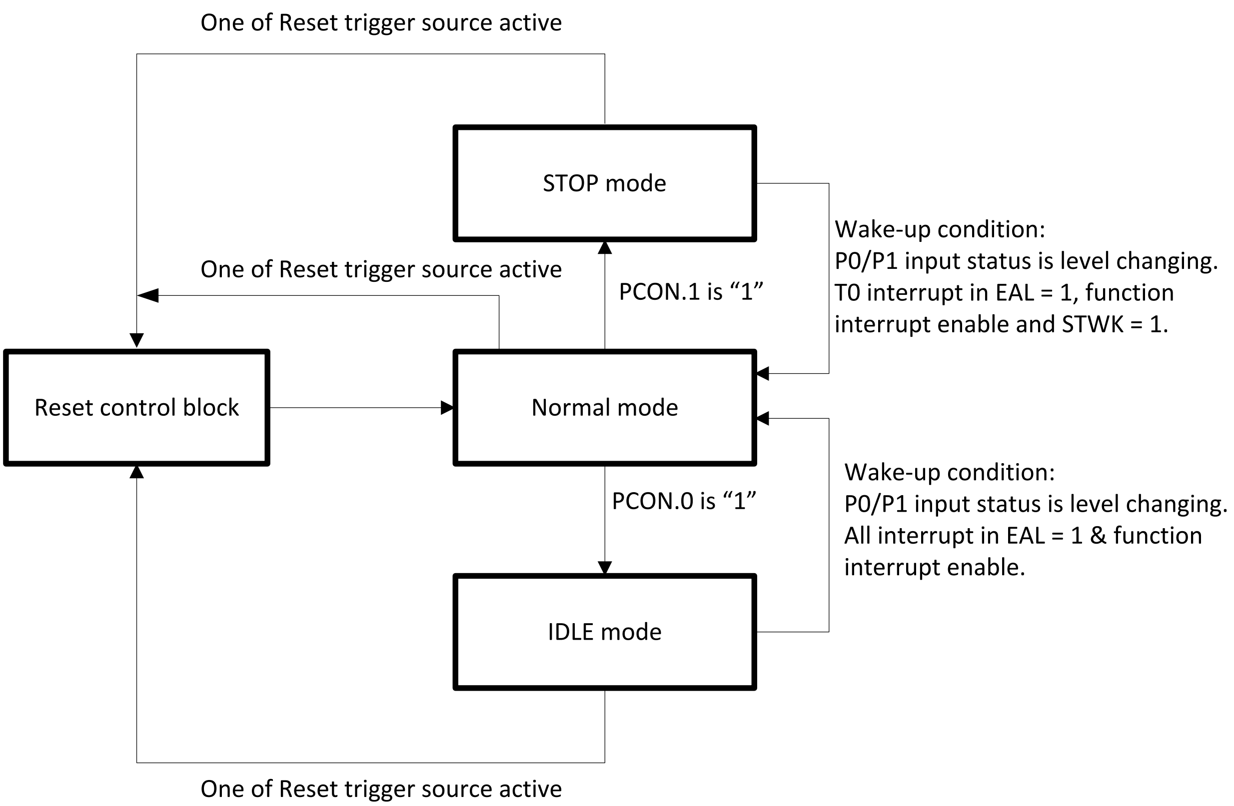
Bit	Field	Type	Initial	Description
7..2	Reserved	R	0x00	
1	T0LCK	R/W	0	T0 时钟源选择 0 = fosc； 1 = flosc。
0	STWK	R/W	0	0：fosc 和 flosc 禁止，在 STOP 模式； 1：flosc 运行，在 STOP 模式*。

* 在进入 STOP 模式前，必须先设置 STWK 位。

9 系统操作模式

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：运行	IHRC：运行	Stop
ILRC	运行	运行	STWK：运行 Watchdog always：运行 其它：禁止
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	T0 时钟源为 Flosc & STWK=1 时激活
定时器 1 （定时器，事件计数器）	由 TR1 激活	由 TR1 激活	未激活
定时器 2 （定时器，事件计数器）	设置为使能时有效	设置为使能时有效	未激活
PWM	设置为使能时有效	设置为使能时有效	未激活
UART	设置为使能时有效	设置为使能时有效	未激活
SPI	设置为使能时有效	设置为使能时有效	未激活
I2C	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	未激活
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	T0 时钟源为 Flosc & STWK=1 时 T0 中断激活 其它：禁止
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 中的总中断 = 1 & 功能中断使能	（1） P0, P1，复位 （2） T0 使能 & 时钟源为 Flosc（ILRC）& STWK = 1 &功能中断使能

- 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 模式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STOP/IDLE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。只有内部稳压器激活，保持所有控制 IO 状态、寄存器状态和 SRAM 内容。STOP 模式由 P0/P1 硬件电平改变触发。P0 唤醒功能始终是使能的。P1 唤醒功能由 P1W 寄存器控制。当 T0 时钟源为 Flosc & STWK = 1 时 T0 中断可唤醒 STOP 模式。STOP 模式唤醒为 NORMAL 模式。进入 STOP 模式由 PCON 寄存器的 PCON.1 位控制。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/低速振荡器，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 硬件电平改变触发，和 T0 中断（T0 时钟源为 Flosc & STWK = 1）。

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 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 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。

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。唤醒触发是在 IO 上升或下降沿的电平改变。P0 和 P1 具有唤醒功能。P0 唤醒功能一直使能，但 P1 由 P1W 寄存器控制。

P1W Register (0x91)

Bit	Field	Type	Initial	Description
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

SN8F5702A 包含 13 个中断源（1 个外部中断和 12 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）。多数中断请求标志位必须由软件清零，而有些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。所有的中断源，中断向量，优先等级以及控制位，如下表所示：

Table 10-1 中断表

中断源	使能中断标志位	请求标志位 (IRQ)	IRQ 清除	优先级 / 向量
系统复位	-	-	-	0 / 0x0000
INT0	EX0	IE0	自动清 0	1 / 0x0003
PWM1	EPWM1	PWM1F	软件清 0	2 / 0x0083
I2C	EI2C	SI	软件清 0	3 / 0x0043
Timer 0	ET0	TF0	自动清 0	4 / 0x000B
ADC	EADC	ADCF	软件清 0	5 / 0x008B
SPI	ESPI	SPIF / WCOL SSERR / MODF	软件清 0	6 / 0x004B
T2COM0	ET2C0	TF2C0	自动清 0	7 / 0x0053
Timer 1	ET1	TF1	自动清 0	8 / 0x001B
T2COM1	ET2C1	TF2C1	自动清 0	9 / 0x005B
UART	ES0	TI0 / RI0	软件清 0	10 / 0x0023
T2COM2	ET2C1	TF2C2	自动清 0	11 / 0x0063
Timer 2	ET2 / ET2RL	TF2 / TF2RL	软件清 0	12 / 0x002B
T2COM3	ET2C3	TF2C3	自动清 0	13 / 0x006B

✸ *注：当中断请求标志可以通过硬件自动清除时，不要在通过软件清除。*

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（0x03-0x8B），系统执行相对应的中断服务程序 ISR。

10.2 中断优先级

每个中断源都有其默认的优先级。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）。所有的中断源分为 6 个等级组（Group0-Group5），可分别为每组设置一个优先级别，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，按照默认的优先级来排序。

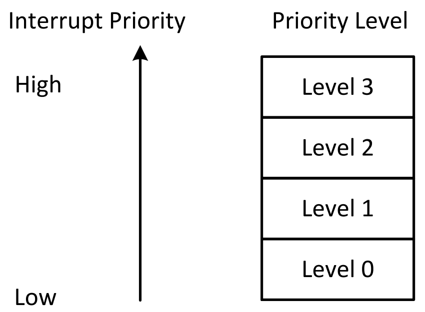
优先级别	IP1.x	IP0.x
Level 0	0	0
Level 1	0	1
Level 2	1	0
Level 3	1	1

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。

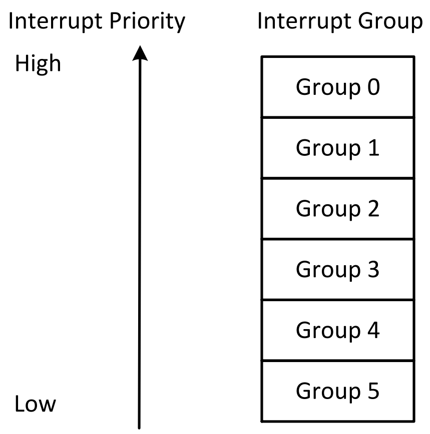
Group	中断源			
Group 0	INT0	PWM1	I2C	-
Group 1	T0	ADC	SPI	-
Group 2	-	-	T2 COM0	-
Group 3	T1	-	T2 COM1	-
Group 4	UART	-	T2 COM2	-
Group 5	T2	-	T2 COM3	-

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。



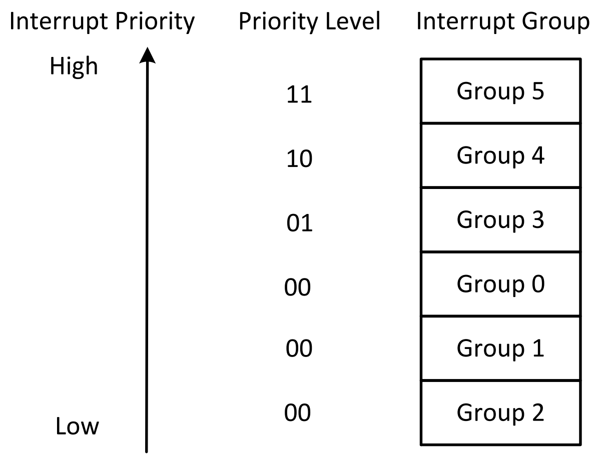
Higher priority level has higher priority



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。

```
MOV    IP0, #00101000B        ; 设置 group0 - group5 的不同优先级。
MOV    IP1, #00110000B
```



Priority Level: GP5>GP4>GP3>GP2=GP1=GP0

Interrupt Priority: GP5>GP4>GP3>GP0>GP1>GP2

IP0, IP1 Registers

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 Register (0XA9)

Bit	Field	Type	Initial	Description
5..0	IP0[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相对应的中断的优先级别。
Else	Reserved	R	0	

IP1 Register (0XB9)

Bit	Field	Type	Initial	Description
5..0	IP1[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	

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	-	ET2	ES0	ET1	-	ET0	EX0
IEN1	ET2RL	-	ET2C3	ET2C2	ET2C1	ET2C0	ESPI	EI2C
IEN2	-	-	-	-	-	-	EADC	-
IEN4	EPWM1	-	-	-	PWM1F	-	-	-
IRCON	TF2RL	TF2	TF2C3	TF2C2	TF2C1	TF2C0	-	-
IRCON2	-	-	-		-	-	-	ADCF
TCON	TF1	TR1	TF0	TR0	-	-	IE0	-
S0CON	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0
SPSTA	SPIF	WCOL	SSERR	MODF	-	-	-	-
I2CCON	CR2	ENS1	STA	STO	SI	AA	CR1	CR0

IEN0 Register (0XA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	所有中断使能控制位。 0：禁止； 1：使能。
5	ET2	R/W	0	T2 定时器中断控制位。 0：禁止； 1：使能。
4	ES0	R/W	0	UART 中断控制位。 0：禁止； 1：使能。
3	ET1	R/W	0	T1 定时器中断控制位。 0：禁止； 1：使能。
1	ET0	R/W	0	T0 定时器中断控制位。 0：禁止； 1：使能。
0	EX0	R/W	0	外部中断 P1.0（INT0）中断控制位。 0：禁止； 1：使能。
Else	Reserved	R	0	

IEN1 Register (0XB8)

Bit	Field	Type	Initial	Description
7	ET2RL	R/W	0	T2 定时器外部重装中断控制位。 0：禁止； 1：使能。
5	ET2C3	R/W	0	T2 定时器 COM3 中断控制位。 0：禁止； 1：使能。
4	ET2C2	R/W	0	T2 定时器 COM2 中断控制位。 0：禁止； 1：使能。
3	ET2C1	R/W	0	T2 定时器 COM1 中断控制位。 0：禁止； 1：使能。
2	ET2C0	R/W	0	T2 定时器 COM0 中断控制位。 0：禁止； 1：使能。
1	ESPI	R/W	0	SPI 中断控制位。 0：禁止； 1：使能。
0	EI2C	R/W	0	I2C 中断控制位。 0：禁止； 1：使能。
Else	Reserved	R	0	

IEN2 Register (0X9A)

Bit	Field	Type	Initial	Description
1	EADC	R/W	0	ADC 中断控制位。 0：禁止； 1：使能。
Else	Reserved	R	0	

IEN4 Register (0XD1)

Bit	Field	Type	Initial	Description
7	EPWM1	R/W	0	PWM1 中断控制位。 0：禁止； 1：使能。
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。
Else	Reserved	R	0	

IRCON Register (0xC0)

Bit	Field	Type	Initial	Description
7	TF2RL	R/W	0	T2 定时器外部重装中断请求标志位。 0：无 TF2RL 中断请求； 1：TF2RL 请求中断。
6	TF2	R/W	0	T2 定时器中断请求标志位。 0：无 T2 中断请求； 1：T2 请求中断。
5	TF2C3	R/W	0	T2 定时器 COM3 中断请求标志位。 0：无 T2COM3 中断请求； 1：T2COM3 请求中断。
4	TF2C2	R/W	0	T2 定时器 COM2 中断请求标志位。 0：无 T2COM2 中断请求； 1：T2COM2 请求中断。
3	TF2C1	R/W	0	T2 定时器 COM1 中断请求标志位。 0：无 T2COM1 中断请求； 1：T2COM1 请求中断。
2	TF2C0	R/W	0	T2 定时器 COM0 中断请求标志位。 0：无 T2COM0 中断请求； 1：T2COM0 请求中断。
Else	Reserved	R	0	

IRCON2 Register (0XBF)

Bit	Field	Type	Initial	Description
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	

TCON Register (0X88)

Bit	Field	Type	Initial	Description
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。
1	IE0	R	0	外部中断 P1.0（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。

S0CON Register (0X98)

Bit	Field	Type	Initial	Description
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。在在模式 0 的第 8 位结束时，或者其他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，当 UART 串行接收完成时，由硬件置 1。在模式 0 的第 8 位结束时，或者其他模式的停止位的中间时，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。

SPSTA Register (0XE1)

Bit	Field	Type	Initial	Description
7	SPIF	R	0	SPI 完成通讯标志位。 通讯结束时自动设置为 1； 读取 SPSTA、SPDAT 寄存器时自动清零。
4	MODF	R	0	模式错误标志位。
Else				参考其它章节。

I2CCON Register (0XDC)

Bit	Field	Type	Initial	Description
7	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

例：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     0X000B          ; Jump to interrupt service routine address.
            JMP     ISR_T0
            ORG     0X001B
            JMP     ISR_T1
            …
            ORG     0X008B
            JMP     ISR_ADC
            …
            ORG     0X00EC
START:                              ; 00ECH, The head of user program.
            …                       ; User program.
            …
            JMP     START           ; End of user program.
            …
ISR_T0:                             ; 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_ADC:                            ;
            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_T1                              ;
            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 GPIO

SN8F5702A 共有 18 个 GPIO 引脚，不同于 8051 只有开漏输出，SN8F5702A 还内置推挽式输出结构，以增强其驱动能力。

11.1 输入输出控制

由 P0M-P2M 寄存器控制输入输出模式。这些位指定每个引脚的输出或输入模式。

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	-	-	-	-	-	-	P21M	P20M
P0OC	-	-	-	P15OC	P14OC	P13OC	P06OC	P05OC

P0M Register (0xF9)

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

1	P01M	R/W	0	P0.1 的模式选择控制位。 0：输入模式； 1：输出模式。
0	P00M	R/W	0	P0.0 的模式选择控制位。 0：输入模式； 1：输出模式。

P1M Register (0xFA)

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

P2M Register (0xFB)

Bit	Field	Type	Initial	Description
7..2				其它
1	P21M	R/W	0	P2.1 的模式选择控制位。 0：输入模式； 1：输出模式。
0	P20M	R/W	0	P2.0 的模式选择控制位。 0：输入模式； 1：输出模式。

P0OC Register (0xE4)

Bit	Field	Type	Initial	Description
7..5		R/W	000	参考 PWM 章节。
4	P15OC	R/W	0	P1.5 开漏输出模式控制位。 0：禁止； 1：使能，输出高电平时变为输入模式。
3	P14OC	R/W	0	P1.4 开漏输出模式控制位。 0：禁止； 1：使能，输出高电平时变为输入模式。
2	P13OC	R/W	0	P1.3 开漏输出模式控制位。 0：禁止； 1：使能，输出高电平时变为输入模式。
1	P06OC	R/W	0	P0.6 开漏输出模式控制位。 0：禁止； 1：使能，输出高电平时变为输入模式。
0	P05OC	R/W	0	P0.5 开漏输出模式控制位。 0：禁止； 1：使能，输出高电平时变为输入模式。

11.2 输入和输出数据

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	-	-	-	-	-	-	P21	P20

P0 Register (0x80)

Bit	Field	Type	Initial	Description
7	P07	R/W	1	读：P0.7 为逻辑电平； 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	Description
7	P17	R/W	1	读：P1.7 为逻辑电平； 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	Description
7..2				其它。
1	P21	R/W	1	读：P2.1 为逻辑电平； 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 为逻辑电平； 写入 1/0：输出高电平/低电平（P20M=1 时使能）。

11.3 内置上拉寄存器

P0UR-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	-	-	-	-	-	-	P21UR	P20UR

P0UR Register (0xF1)

Bit	Field	Type	Initial	Description
7	P07UR	R/W	0	P0.7 的内置上拉电阻控制位。 0：禁止*； 1：使能。
6	P06UR	R/W	0	P0.6 的内置上拉电阻控制位。 0：禁止*； 1：使能。
5	P05UR	R/W	0	P0.5 的内置上拉电阻控制位。 0：禁止*； 1：使能。
4	P04UR	R/W	0	P0.4 的内置上拉电阻控制位。 0：禁止*； 1：使能。
3	P03UR	R/W	0	P0.3 的内置上拉电阻控制位。 0：禁止*； 1：使能。
2	P02UR	R/W	0	P0.2 的内置上拉电阻控制位。 0：禁止*； 1：使能。
1	P01UR	R/W	0	P0.1 的内置上拉电阻控制位。 0：禁止*； 1：使能。
0	P00UR	R/W	0	P0.0 的内置上拉电阻控制位。 0：禁止*； 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	Description
7	P17UR	R/W	0	P1.7 的内置上拉电阻控制位。 0：禁止*； 1：使能。
6	P16UR	R/W	0	P1.6 的内置上拉电阻控制位。 0：禁止*； 1：使能。
5	P15UR	R/W	0	P1.5 的内置上拉电阻控制位。 0：禁止*； 1：使能。
4	P14UR	R/W	0	P1.4 的内置上拉电阻控制位。 0：禁止*； 1：使能。
3	P13UR	R/W	0	P1.3 的内置上拉电阻控制位。 0：禁止*； 1：使能。
2	P12UR	R/W	0	P1.2 的内置上拉电阻控制位。 0：禁止*； 1：使能。
1	P11UR	R/W	0	P1.1 的内置上拉电阻控制位。 0：禁止*； 1：使能。
0	P10UR	R/W	0	P1.0 的内置上拉电阻控制位。 0：禁止*； 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	Description
7..2				其它
1	P21UR	R/W	0	P2.1 的内置上拉电阻控制位。 0：禁止*； 1：使能。
0	P20UR	R/W	0	P2.0 的内置上拉电阻控制位。 0：禁止*； 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。

11.4 与模拟功能共用的引脚

SN8F5702A 内置模拟功能，如 ADC。若使能引脚的模拟功能，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	-	-	-	-	-	-	-
P1CON	P1CON7	P1CON6	P1CON5	P1CON4	P1CON3	P1CON2	P1CON1	P1CON0
P2CON	-	-	-	-	-	-	P2CON1	P2CON0

P0CON Register (0x9D)

Bit	Field	Type	Initial	Description
7	P0CON7	R/W	0	P0.7 施密特触发控制位。 0：使能； 1：禁止。
6..0				其它

P1CON Register (0xD6)

Bit	Field	Type	Initial	Description
7	P1CON7	R/W	0	P1.7 施密特触发控制位。 0：使能； 1：禁止。
6	P1CON6	R/W	0	P1.6 施密特触发控制位。 0：使能； 1：禁止。
5	P1CON5	R/W	0	P1.5 施密特触发控制位。 0：使能； 1：禁止。
4	P1CON4	R/W	0	P1.4 施密特触发控制位。 0：使能； 1：禁止。
3	P1CON3	R/W	0	P1.3 施密特触发控制位。 0：使能； 1：禁止。
2	P1CON2	R/W	0	P1.2 施密特触发控制位。 0：使能； 1：禁止。
1	P1CON1	R/W	0	P1.1 施密特触发控制位。 0：使能； 1：禁止。
0	P1CON0	R/W	0	P1.0 施密特触发控制位。 0：使能； 1：禁止。

P2CON Register (0x9E)

Bit	Field	Type	Initial	Description
7..2				其它
1	P2CON6	R/W	0	P2.1 施密特触发控制位。 0：使能； 1：禁止。
0	P2CON5	R/W	0	P2.0 施密特触发控制位。 0：使能； 1：禁止。

12 外部中断

外部中断源 INT0，内置边沿触发功能，由 PEDGE 寄存器控制。使能外部中断（EX0）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）置 1，程序计数器跳转到中断向量（ORG 0x0003）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。

12.1 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	-	-	-	-	-	-	EX0G1	EX0G0
IEN0	EAL	-	ET2	ES0	ET1	-	ET0	EX0
TCON	TF1	TR1	TF0	TR0	-	-	IE0	-

PEDGE Register (0X8F)

Bit	Field	Type	Initial	Description
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 Register (0XA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	总中断使能控制位。 0：禁止； 1：使能。
0	EX0	R/W	0	外部中断 INT0 控制位。 0：禁止； 1：使能。
Else				参考其它章节。

TCON Register (0X88)

Bit	Field	Type	Initial	Description
1	IE0	R/W	0	外部中断 INT0 请求标志位。 0：无 INT0 中断； 1：INT0 中断请求。
Else				参考其它章节。

✸ ***注：在通过软件手动清除 TF0，TF1，IE0 或 IE1 中任意一个时，用户必须确认其它的中断请求标志不为 1。***

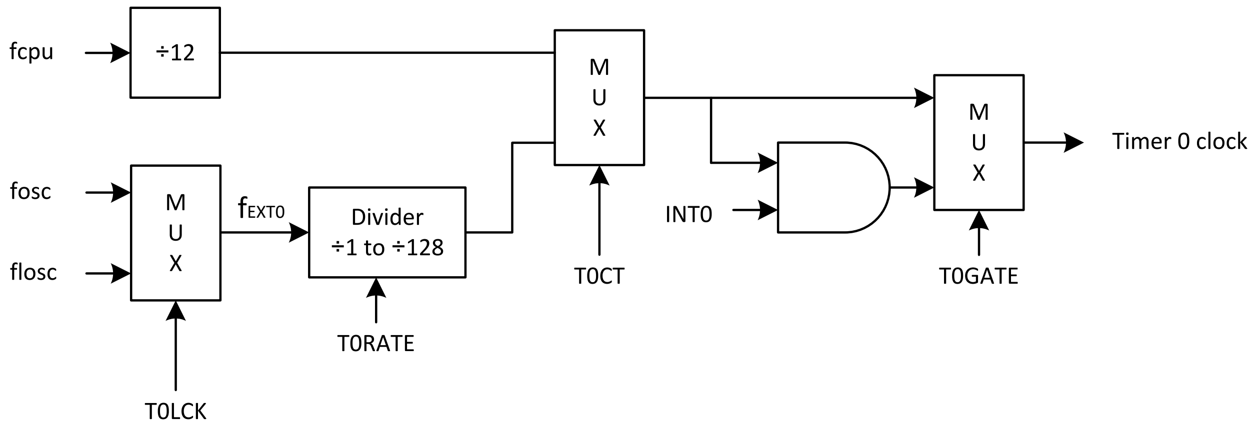
13 定时器 T0 和定时器 T1

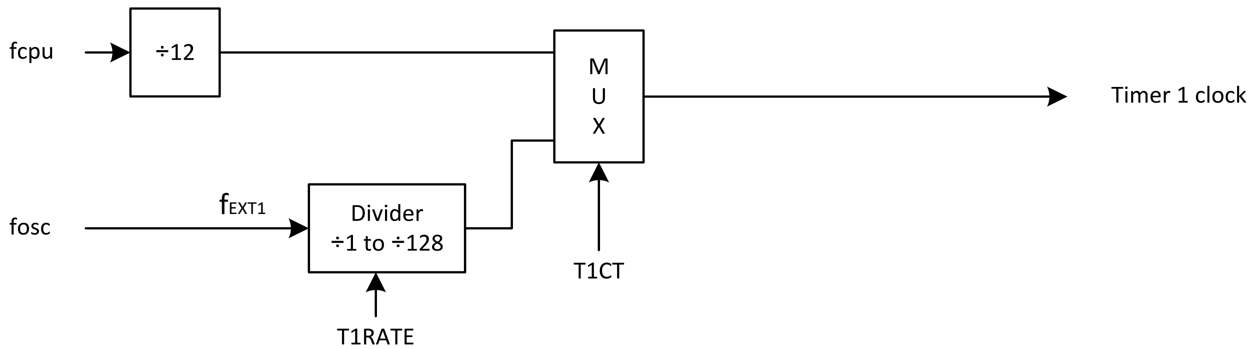
T0 和 T1 是 2 个独立的二进制定时器。T0 共有 4 种不同的的操作模式：模式 1：13 位向上计数定时器；模式 2：16 位向上计数定时器；模式 3：8 位向上计数寄存器，支持指定的重装值；模式 4：独立的 2 个 8 位向上计数定时器。而 T1 只有与 T0 相同的模式 0 到模式 2 三种操作模式。T0 和 T1 分别支持 ET0 和 ET1 中断。

当 T0 时钟源为 Flosc 且 STWK=1 时，T0 能在 STOP 模式运行并且 T0 中断可唤醒 STOP 模式。

13.1 T0 和 T1 时钟选择

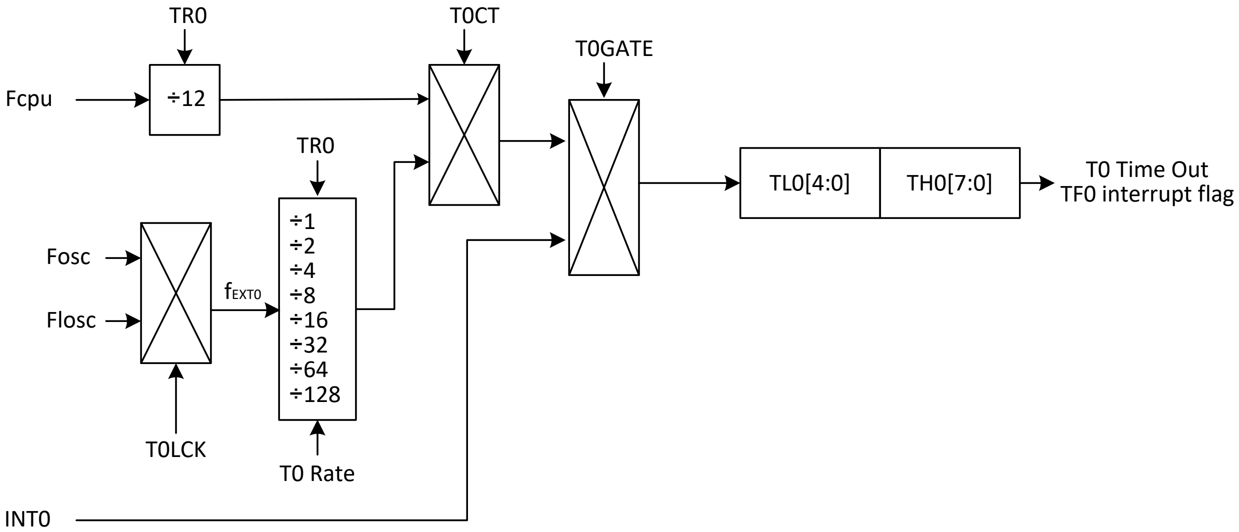
下图阐明了 T0 和 T1 的时钟选择电路。T0 有三个时钟源选项：Fcpu，Fosc 和 Flosc。如果应用了 T0GATE，则所有时钟源都可以由 INT0 引脚控制（暂停）。T1 时钟源选项：Fcpu 和 Fosc。总的来说，这两个定时器之间的主要区别是 T0 额外支持 flosc 时钟源(低速时钟)。

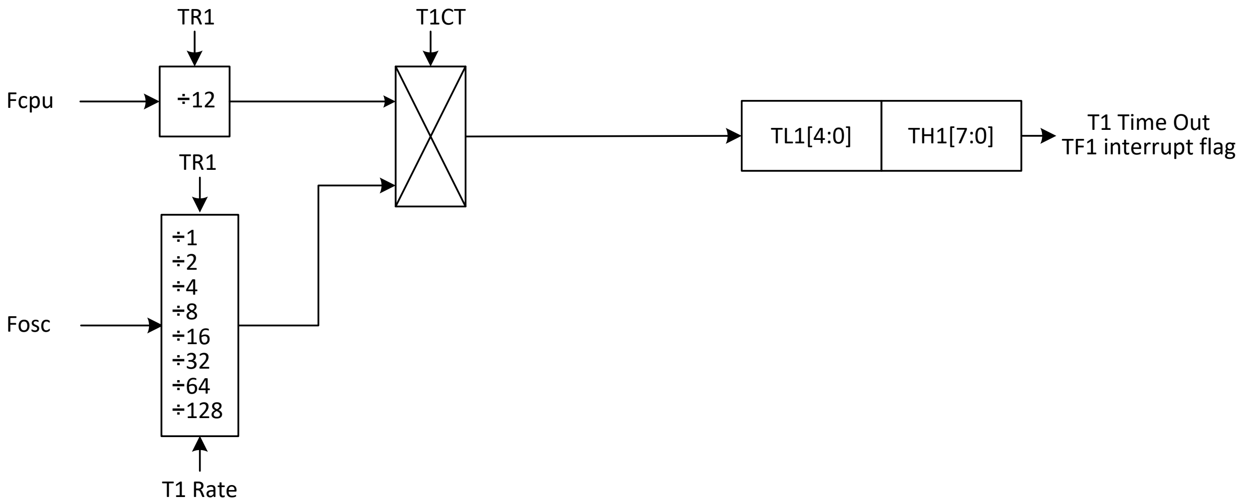




13.2 模式 0：13 位向上计数定时器

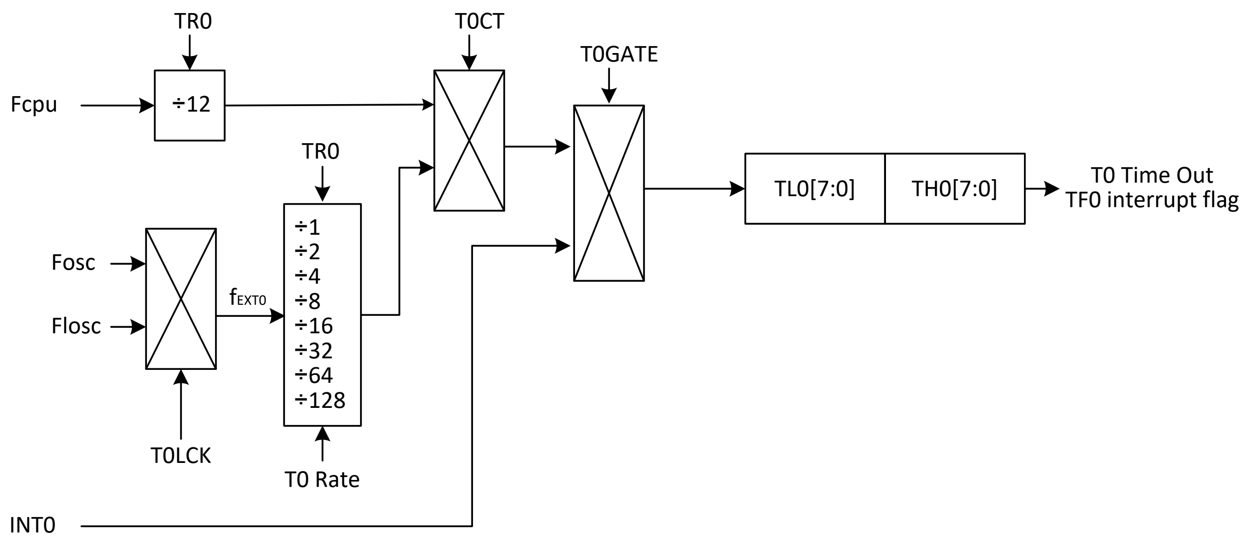
模式 0 是 13 位向上计数定时器(TL0 的高 3 位无效)。一旦定时器的计数器溢出(从 0xFF1F 到 0x0000)，会立即发出 TF0/TF1 标志。没有使能 ET0/ET1 时，由软件读写该标志；使能 ET0/ET1 时，则由中断控制器进行控制。

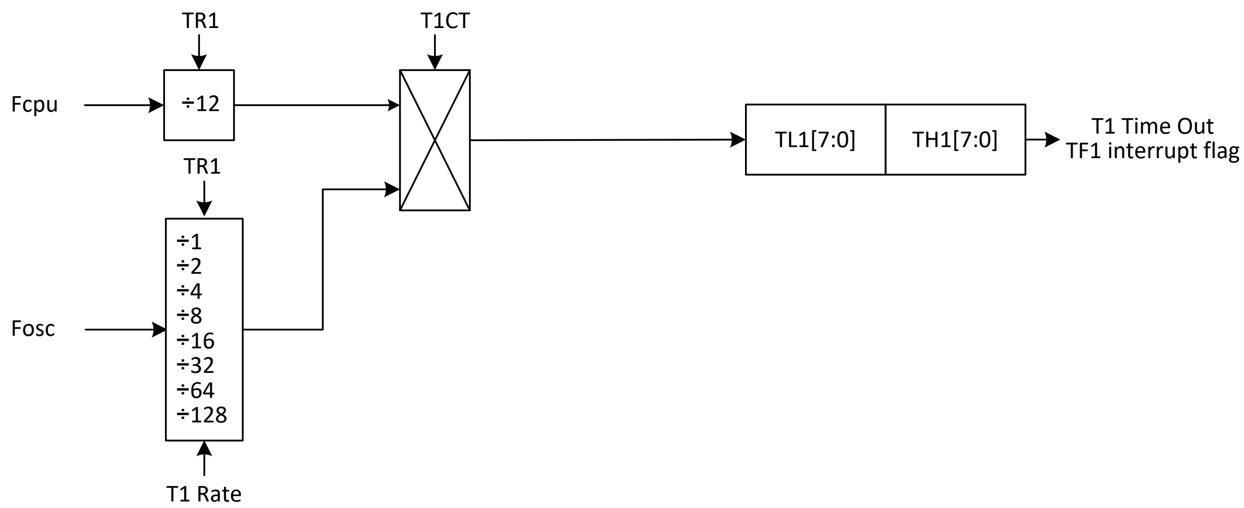




13.3 模式 1：16 位向上计数定时器

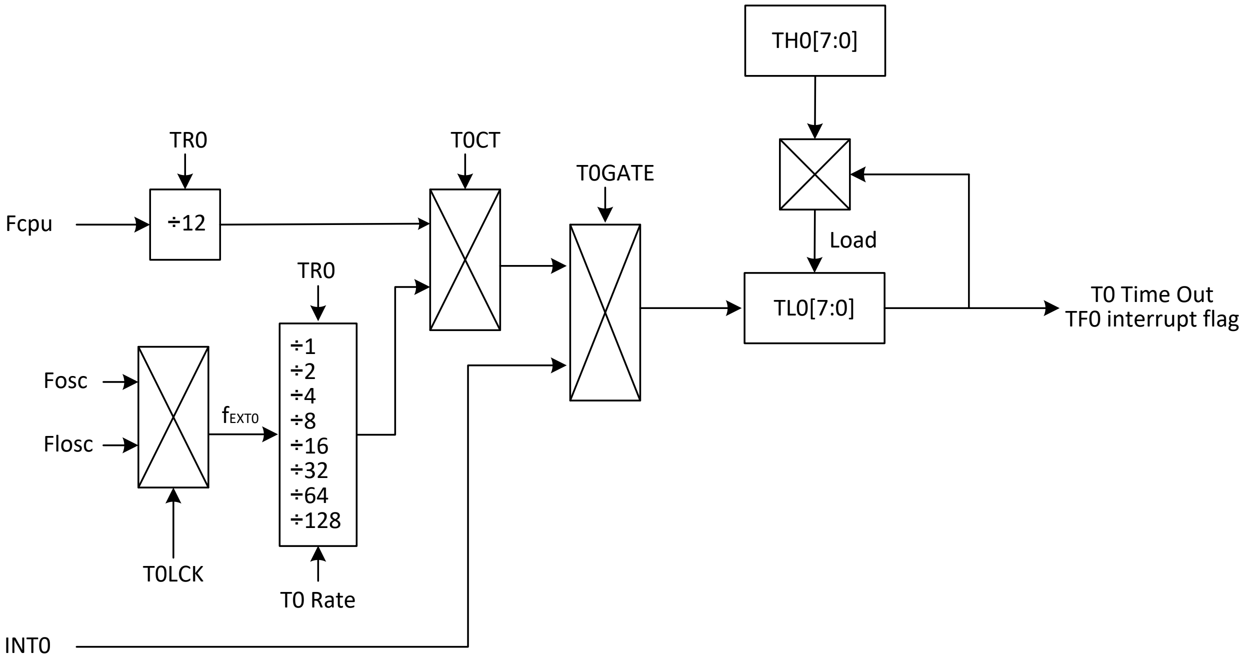
模式 1 是 16 位向上计数定时器。一旦定时器的计数器溢出（从 0xFFFF 到 0x0000），会立即发出 TF0/TF1。没有使能 ET0/ET1 时，由软件读写该标志；使能 ET0/ET1 时，则由中断控制器进行控制。

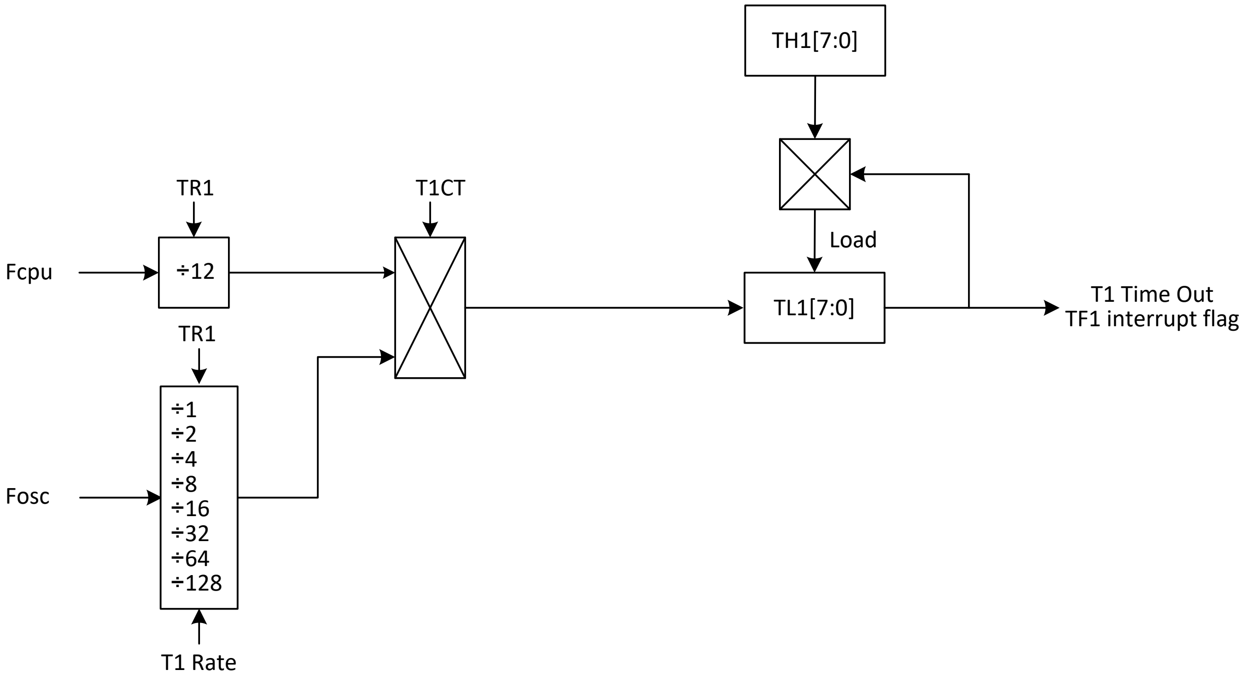




13.4 模式 2：8 位向上计数定时器（指定重装值）

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（TL0/TL1 计数从 0xFF 到 0x00）释放 TF0/TF1 标志给 FW 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 0xFF 计数到 TH0/TH1 的值。



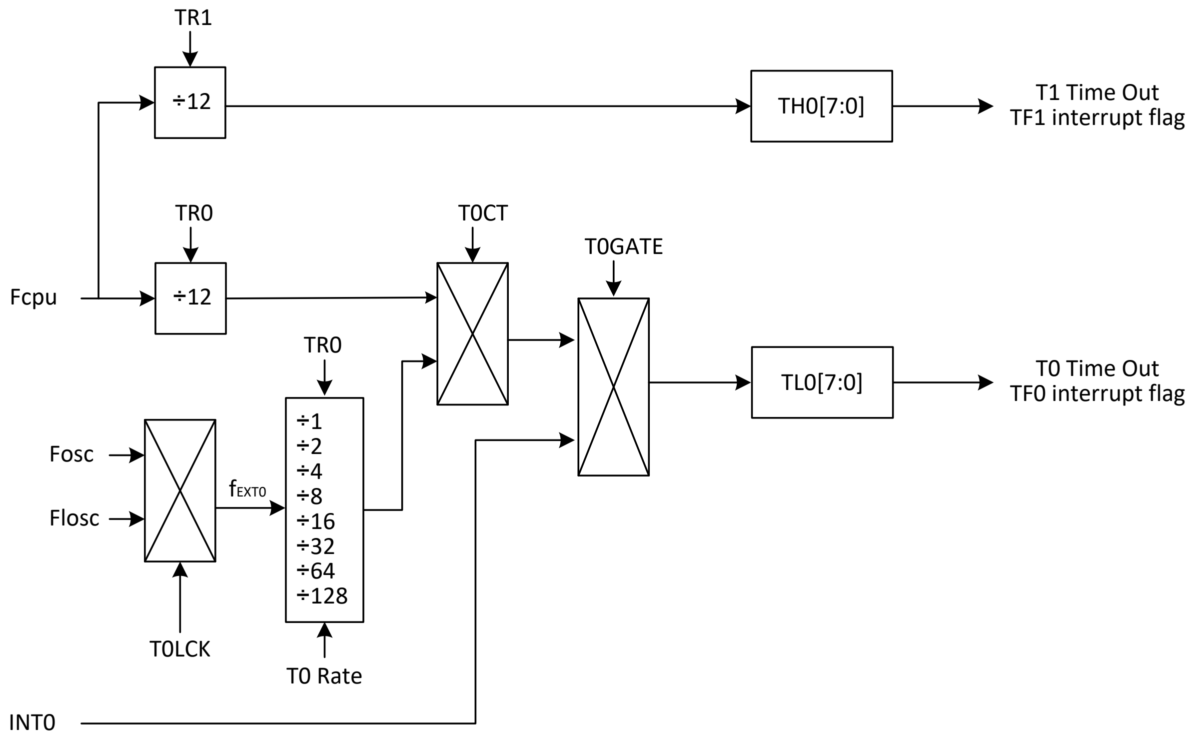


13.5 模式 3 (只有 T0)：独立的 2 个 8 位向上计数定时器

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 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，TLO 时钟源的使能和停止可以由 INT0 引脚控制。

在模式 3 下，由 TR0 使能 TL0 计数器，TL0 溢出后 TF0 标志位置 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的自计数器。



13.6 T0 和 T1 寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCON	TF1	TR1	TF0	TR0	-	-	IE0	-
TCON0	-	T0RATE2	T0RATE1	T0RATE0	-	T1RATE2	T1RATE1	T1RATE0
TMOD	-	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	EAL	-	ET2	ES0	ET1	-	ET0	EX0
SYSMOD	-	-	-	-	-	-	T0LCK	STWK

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	中断使能位，参考中断章节。
3	ET1	R/W	0	T1 中断控制位。 0：禁止； 1：使能。
1	ET0	R/W	0	T0 中断控制位。 0：禁止； 1：使能。

TH0 / TH1 Registers (TH0: 0x8C, TH1: 0x8D)

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

TL0 / TL1 Register (TL0: 0x8A, TL1: 0x8B)

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

SYSMOD Register (0xCF)

Bit	Field	Type	Initial	Description
7..2	Reserved	R	0x00	
1	T0LCK	R/W	0	T0 f_{EXT0} 时钟源选项。 0：fosc； 1：flosc。
0	STWK	R/W	0	0：STOP 模式下停止 fosc 和 flosc ； 1：STOP 模式下 flosc 保持运行*。

* 进入 STOP 模式前，设置 STOP 位之前必须先设置 STWK 位。

TCON Register (0x88)

Bit	Field	Type	Initial	Description
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..2	Reserved	R	0	
1	IE0	R/W	0	参考 INT0。
0	Reserved	R	0	

> ✱ ***注：在通过软件手动清除 TF0，TF1 或 IE0 中任意一个时，用户必须确认 TCON 寄存器中其它的中断请求标志不为 1。***

TCON0 Register (0xE7)

Bit	Field	Type	Initial	Description
7	Reserved	R	0	
6..4	T0RATE[2:0]	R/W	000	T0 外部时钟源的时钟分频控制位。 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控制位。 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。

*(1) f_{EXT1} = fosc.

*(2) f_{EXT0} = fosc or flosc

TMOD Register (0x89)

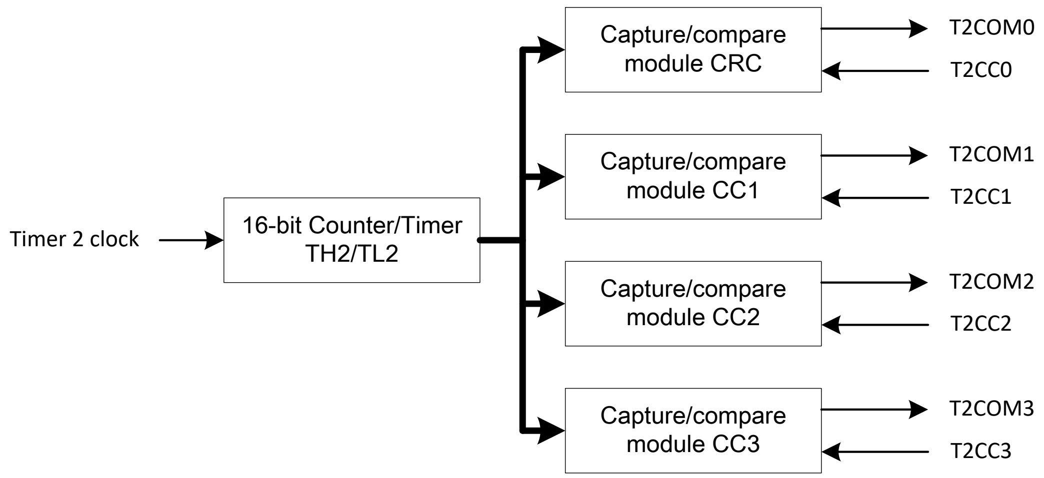
Bit	Field	Type	Initial	Description
7	Reserved	R	0	
6	T1CT	R/W	0	T1 时钟源选择。 0：F Timer1 = fcpu / 12； 1：F Timer 1 = f_{EXT1} / T1RATE (参考 T1RATE) *(1)。
5..4	T1M[1:0]	R/W	00	T1 操作模式控制位。 00：13 位向上计数定时器； 01：16 位向上计数定时器； 10：8 位向上计数定时器，支持重装； 11：保留。
3	T0GATE	R/W	0	T0 gate 模式控制位。 0：禁止； 1：使能，T0 时钟源由 INT0 gated。
2	T0CT	R/W	0	T0 时钟源选择。 0：F_{Timer0} = fcpu / 12； 1：F_{Timer0} = f_{EXT0} / T0RATE (参考 T0RATE) *(2)。
1..0	T0M[1:0]	R/W	00	T0 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：独立的 2 个 8 位向上计数定时器。

*(1) f_{EXT1} = fosc.

*(2) f_{EXT0} = fosc or flosc.

14 定时器 T2

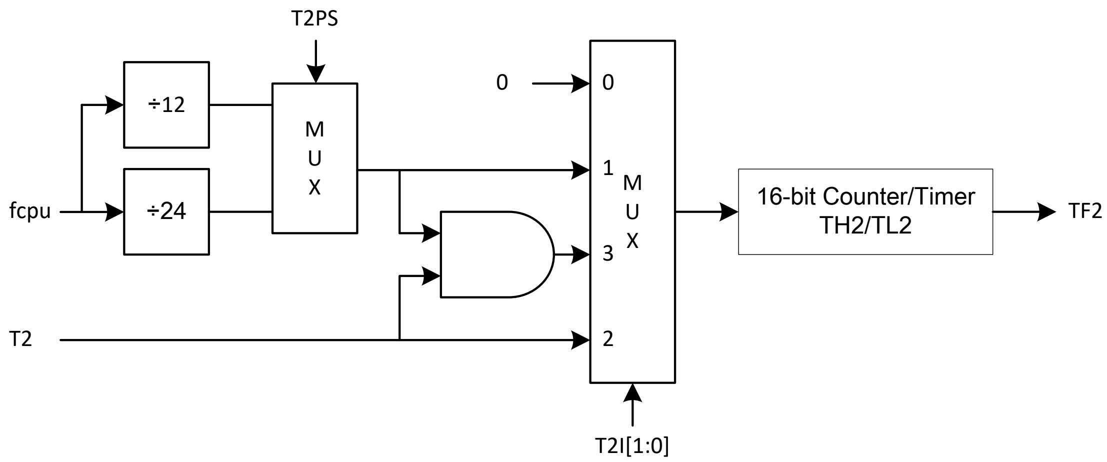
T2 是 16 位向上计数定时器，具有好几个可选的扩展功能：指定重装值，比较器输出（PWM）和捕捉功能。T2 由 1 个 16 位计数/定时器和 4 个 16 位捕捉/比较器模块组成：每个捕捉/比较器模块在使能状态时都有自己相关的 IO 引脚；且每个捕捉/比较器模块都可以作为下列模式独立工作：比较器，上升沿时捕捉，以及写入寄存器时捕捉。



14.1 T2 向上计数控制

T2 共有 3 种操作模式及相应的时钟源：特定的 Fcpu 时钟（Fcpu/12 和 Fcpu/24）、特定的 Fcpu 时钟带停止控制、以及外部时钟输入。下表对 3 种操作模式和相关的寄存器（T2I1，T2I0 和 T2PS）进行了分类，定时器的计数器一旦溢出（从 0xFFFF 到 0x0000），立即置位 TF2，并由软件进行读/写。T2 的中断功能由 ET2 控制。

T2I1	T2I0	T2PS	T2 时钟源
0	0	X	禁止 T2
0	1	0	fcpu/12
0	1	1	fcpu/24
1	1	0	fcpu/12（T2 引脚为低电平时停止计数，变为高电平后再开始计数）
1	1	1	fcpu/24（T2 引脚为低电平时停止计数，变为高电平后再开始计数）
1	0	X	T2 引脚上升沿（T2 引脚与 P0.7 共有，时钟率≤ 0.5 * fcpu）

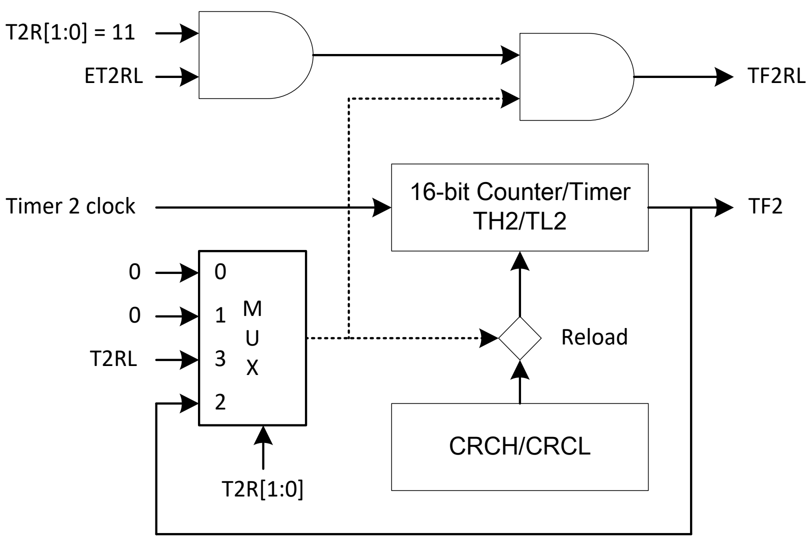


14.2 指定 T2 重装值

指定重装值是一项可选择的功能，通过溢出或者外部控制引脚对 T2 计数器进行重装。

若选择溢出重装初值功能后，T2 溢出后，通过溢出信号自动复制 CRCH/CRCL 的值到计数器 TH2/TL2。因此，T2 就从 CRCH/CRCL 值重复地计数到 0xFFFF。

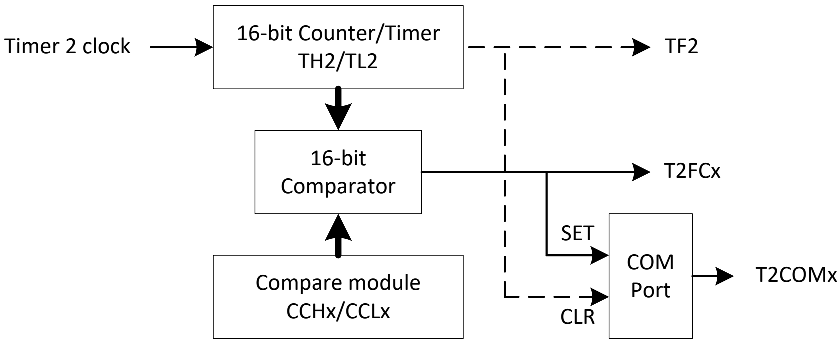
从另一方面来说，外部引脚 T2RL（与 P2.0 共用）的下降沿可以选择为重装信号。在这种情况下，若 T2RL 引脚保持稳定，T2 从 0x0000 开始计数直到 0xFFFF，但只要 T2RL 引脚有下降信号，CRCH/CRCL 的值随时可能替换计数器的值。然后 T2 继续从 CRCH/CRCL 的值开始计数，若使能中断功能（ET2RL 和 ET2 都置 1），则置位外部重装中断标志（TF2RL）。外部重装中断向量与 T2 中断向量共用，由软件去判断。

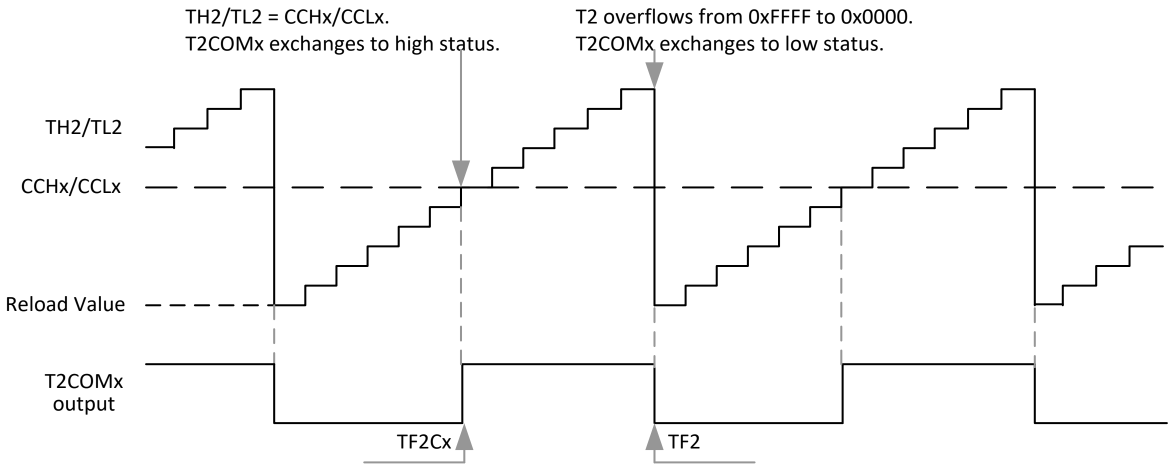


14.3 比较器输出 (PWM)

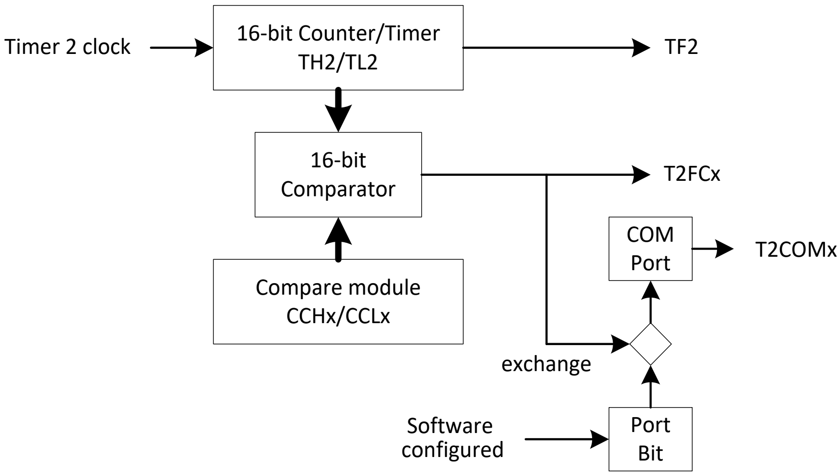
T2 共有 4 组比较输出，每组（CRC/CC1/CC2/CC3）分别与 T2 计数器（TH2/TL2）进行比较，并通过 T2COM0-T2COM3 引脚（与 P00、P01、P03 和 P04 共用）输出比较结果，比较结果有 2 种输出方式：直接输出和间接输出。

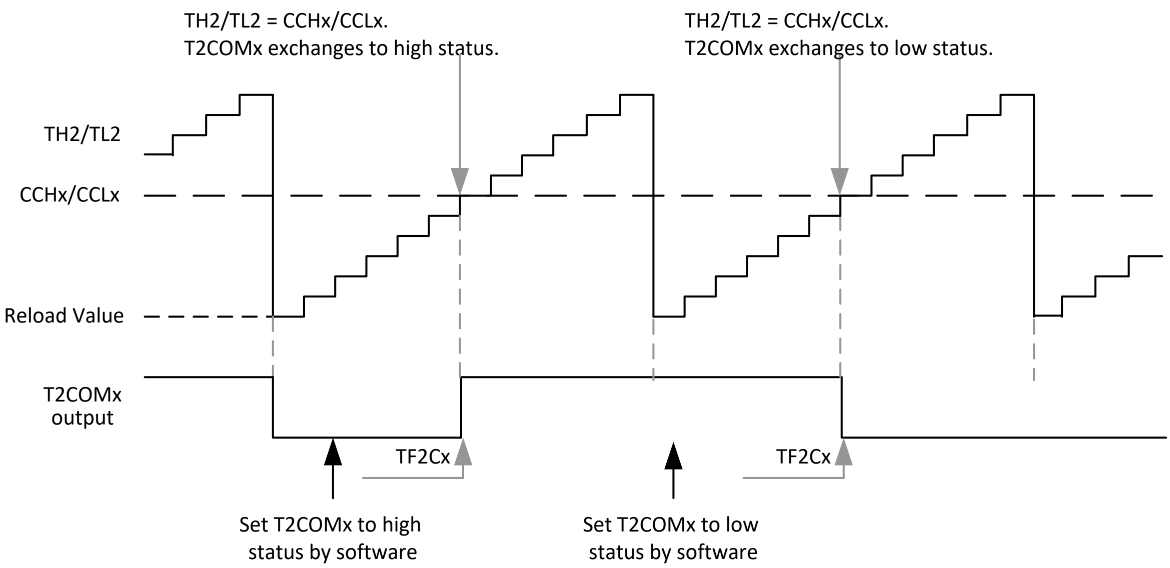
直接输出是指若 CRC/CC1/CC2/CC3 寄存器小于 T2 计数器，其对应的引脚输出低电平；反之若 CRC/CC1/CC2/CC3 大于或等于 T2 计数器则输出高电平。因此其输出状态可在交叉点更改 2 次。CRC/CC1/CC2/CC3 等于 T2 计数器时，给出一个 TF2C0/FT2C1/TF2C2/TF2C3 标志并通过软件进行读/写。比较中断功能由 ET2C0/ET2C1/ET2C2/ET2C3 控制。





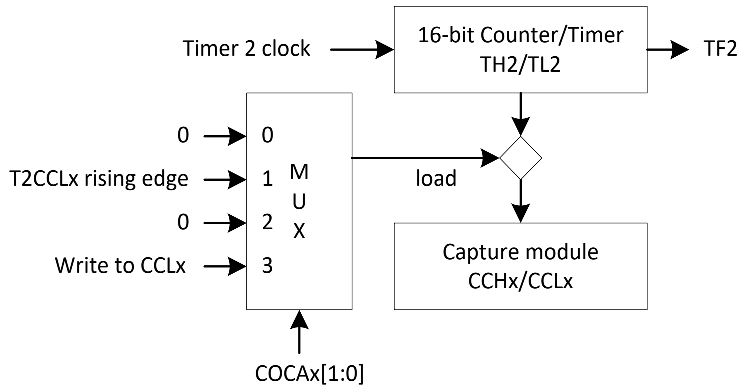
相比之下，间接输出是指保持相对应引脚先前的输出设置状态直到 T2 计数器的值超过 CRC/CC1/CC2/CC3 寄存器的值。在该模式下，可以由软件来控制输出信号的转换，换句话说，就是 TH2/TL2 和 CRC 寄存器的值相等时，P0.0 寄存器位会影响 T2COM0/P0.0 引脚。而 T2 的溢出则不会导致输出变化。



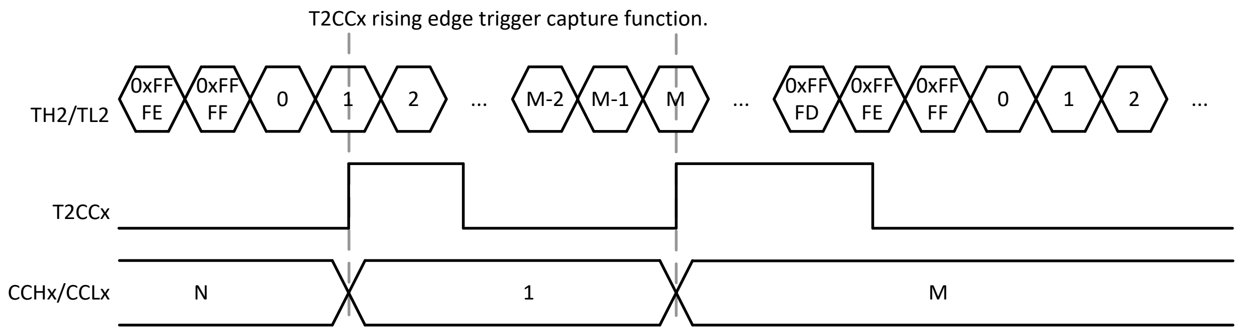


14.4 捕获功能

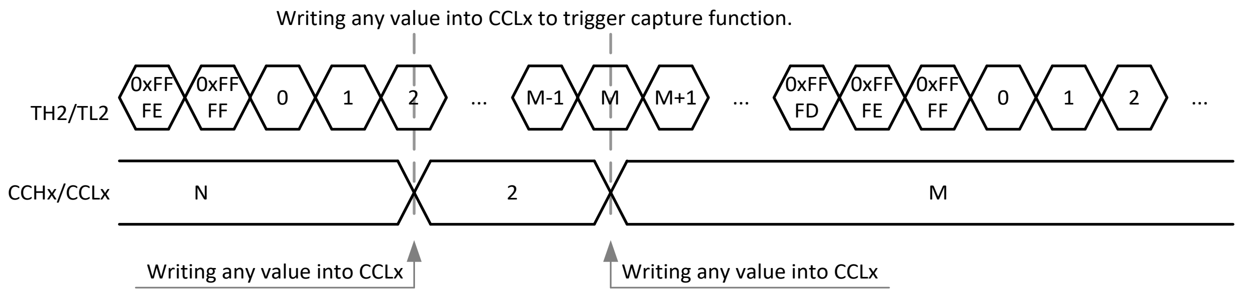
捕捉功能类似于秒表的 split/lap 按键，当 T2 计数器（TH2/TL2）开始向上计数时，一个 split 事件在 CRC/CC1/CC2/CC3 寄存器中记录计数器的值。



split 事件可由硬件或软件提供，T2CC0 引脚可触发硬件 split 事件，复制 TH2/TL2 的值到 CRCH/CRCL 寄存器，而 T2CC1、T2CC2 和 T2CC3 分别控制 CC1-CC3 寄存器。



软件 split 事件由写入 CRCL/CCL1/CCL2/CCL3 寄存器的任何值触发。执行一条写指令到这些寄存器时， TH2/TL2 的当前值会记录在对应的寄存器中。



14.5 T2 寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CON	T2PS	I3FR	-	T2R1	T2R0	T2CM	T2I1	T2I0
CCEN	COCA31	COCA30	COCA21	COCA20	COCA11	COCA10	COCA01	COCA00
TH2	TH27	TH26	TH25	TH24	TH23	TH22	TH21	TH20
TL2	TL27	TL26	TL25	TL24	TL23	TL22	TL21	TL20
CRCH	CRCH7	CRCH6	CRCH5	CRCH4	CRCH3	CRCH2	CRCH1	CRCH0
CRCL	CRCL7	CRCL6	CRCL5	CRCL4	CRCL3	CRCL2	CRCL1	CRCL0
CCH3	CCH37	CCH36	CCH35	CCH34	CCH33	CCH32	CCH31	CCH30
CCL3	CCL37	CCL36	CCL35	CCL34	CCL33	CCL32	CCL31	CCL30
CCH2	CCH27	CCH26	CCH25	CCH24	CCH23	CCH22	CCH21	CCH20
CCL2	CCL27	CCL26	CCL25	CCL24	CCL23	CCL22	CCL21	CCL20
CCH1	CCH17	CCH16	CCH15	CCH14	CCH13	CCH12	CCH11	CCH10
CCL1	CCL17	CCL16	CCL15	CCL14	CCL13	CCL12	CCL11	CCL10
IEN0	EAL	-	ET2	ES0	ET1	-	ET0	EX0
IEN1	ET2RL	-	ET2C3	ET2C2	ET2C1	ET2C0	ESPI	EI2C
IRCON	TF2RL	TF2	TF2C3	TF2C2	TF2C1	TF2C0	-	-

T2CON Register (0xC8)

Bit	Field	Type	Initial	Description
7	T2PS	R/W	0	T2 前置分频器。 0：Fcpu/12； 1：Fcpu/24。
6	I3FR	R/W	0	比较模式下： 0：T2 的内容与 CRC 寄存器的不相等时发生 COM0 中断； 1：T2 的内容与 CRC 寄存器的相等时发生 COM0 中断。 捕捉模式 0 下： 0：T2CC0 下降沿时，T2 的内容锁存在 CRC 寄存器中； 1：T2CC0 上升沿时，T2 的内容锁存在 CRC 寄存器中。
5	Reserved	R/W	0	
4..3	T2R[1:0]	R/W	00	指定 T2 的重装值。 00：禁止； 01：禁止； 10：通过计数器溢出加载 CRCH/CRCL 的值到 TH2/TL2； 11：通过 T2RL 引脚加载 CRCH/CRCL 的值到 TH2/TL2。

2	T2CM	R/W	0	T2 比较输出： 0：直接输出； 1：间接输出，指定下一个输出状态。
1..0	T2I[1:0]	R/W	00	T2 向上计数控制。 00：禁止； 01：由 T2PS 定义时钟频率； 10：时钟源来自 T2 引脚； 11：由 T2PS 定义时钟频率，带 T2 引脚控制定时停止。

CCEN Register (0xC1)

Bit	Field	Type	Initial	Description
7..6	COCA3[1:0]	R/W	00	CC3 的比较和捕捉功能。 00：禁止； 01：T2CC3 引脚的上升沿捕捉； 10：比较功能； 11：写入 CCL3 寄存器捕捉。
5..4	COCA2[1:0]	R/W	00	CC2 的比较和捕捉功能。 00：禁止； 01：T2CC2 引脚的上升沿捕捉； 10：比较功能； 11：写入 CCL2 寄存器捕捉。
3..2	COCA1[1:0]	R/W	00	CC1 的比较和捕捉功能。 00：禁止； 01：T2CC1 引脚的上升沿捕捉； 10：比较功能； 11：写入 CCL1 寄存器捕捉。
1..0	COCA0[1:0]	R/W	00	CRC 的比较和捕捉功能。 00：禁止； 01：T2CC0 引脚的上升沿捕捉； 10：比较功能； 11：写入 RCCL 寄存器捕捉。

TH2/TL2 Registers (TH2: 0xCD, TL2: 0xCC)

Bit	Field	Type	Initial	Description
7..0	TH2/TL2	R/W	0x00	T2 16 位计数寄存器。

CRC Registers (CRCH: 0xCB, CRCL: 0xCA)

Bit	Field	Type	Initial	Description
7..6	CRCH[15:0]	R/W	0x00	16 位比较/捕捉寄存器。

CCH3/CCL3 Registers (CCH3: 0xC7, CCL3: 0xC6)

Bit	Field	Type	Initial	Description
7..6	CCH3/CCL3	R/W	0x00	16 位比较/捕捉寄存器。

CCH2/CCL2 Registers (CCH2: 0xC5, CCL2: 0xC4)

Bit	Field	Type	Initial	Description
7..6	CCH2 /CCL2	R/W	0x00	16 位比较/捕捉寄存器。

CCH1/CCL1 Registers (CCH1: 0xC3, CCL1: 0xC2)

Bit	Field	Type	Initial	Description
7..6	CCH1/CCL1	R/W	0x00	16 位比较/捕捉寄存器。

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	使能中断，请参考中断章节。
5	ET2	R/W	0	使能 T2 中断。
Else				请参考其它章节。

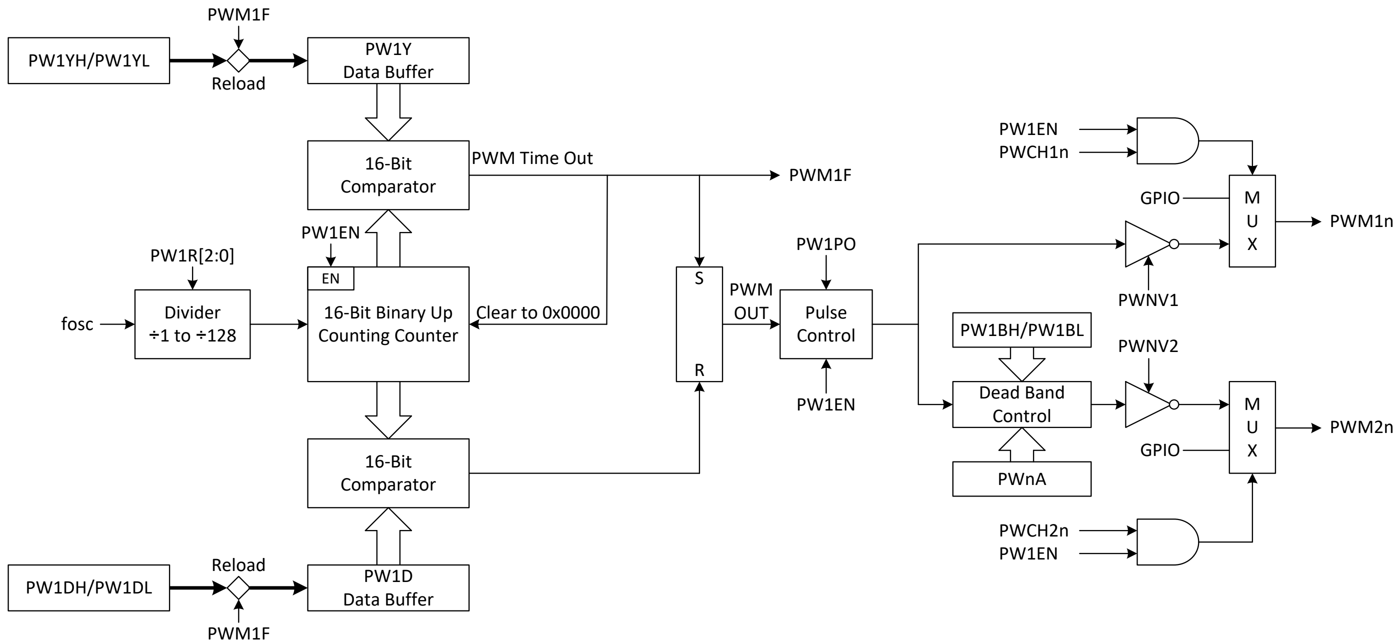
IEN1 Register (0xB8)

Bit	Field	Type	Initial	Description
7	ET2RL	R/W	0	T2 定时器外部重装中断控制位。 0：禁止； 1：使能。
6	Reserved	R	0	
5	ET2C3	R/W	0	T2 定时器 COM3 中断控制位。 0：禁止； 1：使能。
4	ET2C2	R/W	0	T2 定时器 COM2 中断控制位。 0：禁止； 1：使能。
3	ET2C1	R/W	0	T2 定时器 COM1 中断控制位。 0：禁止； 1：使能。
2	ET2C0	R/W	0	T2 定时器 COM0 中断控制位。 0：禁止； 1：使能。
Else				请参考其它章节。

15 PWM

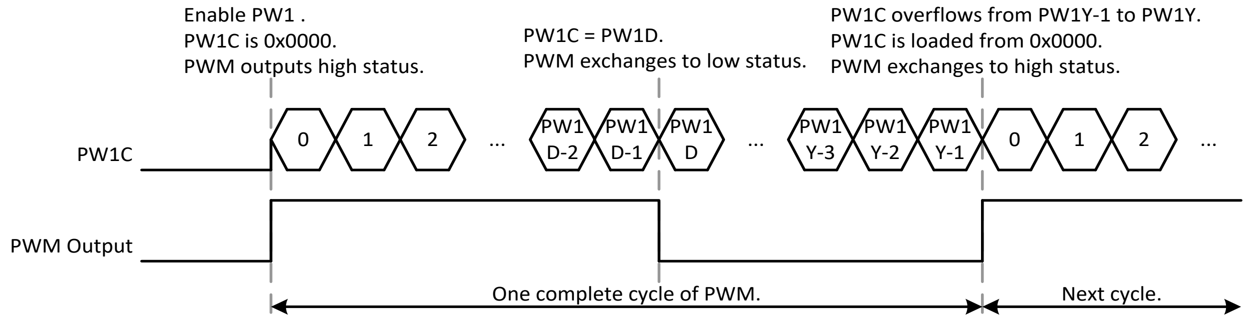
PW1 定时器包含一个 16 位二进制 4 通道 PWM，一个单脉冲 PWM。计数器计数到上限值（PW1Y）后，计数器清零并触发一个中断信号。PWM 的占空比周期由 PW1D 控制。

PWM 还支持单脉冲输出信号，在首个 PWM 周期结束时自行禁止，因此，在这种情况下只产生单个脉冲。PWM 共有 4 个可编程控制的 PWM 通道，与 GPIO 引脚共用，由 PW1CH[5:4, 1:0]位控制。通过使能 PW1CH[5:4, 1:0]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 PW1CH[5:4, 1:0]位时，PWM 通道返回到上一个 GPIO 模式。若使能中断，PW1 内置 IDEL 模式唤醒功能。PWM 定时器溢出时（PW1Y-1 到 PW1Y），立即释放 PWM1F，提供软件进行读/写。由 EPWM1 控制 PW1 的中断功能。



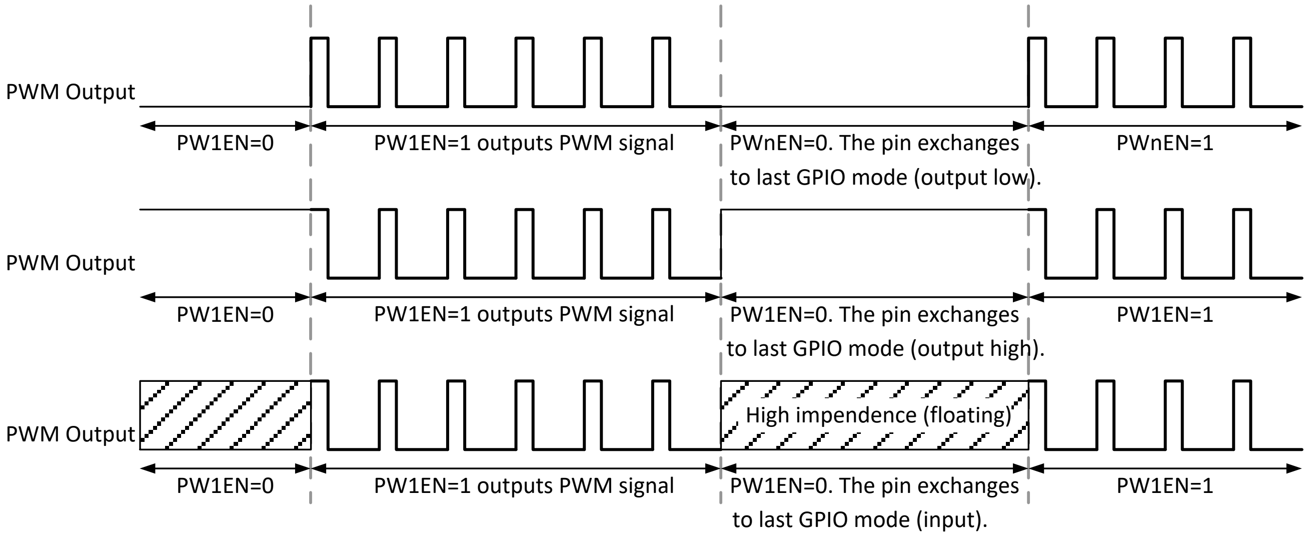
15.1 普通 PWM

PW1 定时器内置 PWM 功能，由 PW1EN 和 PW1CH 寄存器控制，PWM10，PWM11，PWM20，PWM21 为输出引脚，并与 GPIO 引脚共用(P0.5，P0.6，P1.4 和 P1.5)，由 PW1CH 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 通道返回上一个 GPIO 模式。PW1Y 和 PW1D 进行比较，其比较结果产生 PWM 信号。PW1C 从 0x0000 开始计数时，PWM 输出高电平，即 PWM 的初始状态。PW1C 加载 PW1Y 寄存器的值决定 PWM 的周期和分辨率。PW1C 继续计数，系统比较 PW1C 和 PW1D 的值，PW1C=PW1D 时，PWM 输出低电平，PW1C 继续计数。PW1 定时器溢出（PWnY-1 到 0x0000）后，PWM 完成一个信号周期。PW1C 自动重新加载 0x0000，并输出高电平以等待下一个周期。PW1D 决定高电平占空比时间，PW1Y 决定 PWM 的分辨率和周期。PW1D 的值不能大于 PW1Y 的值，否则 PWM 信号会出错。



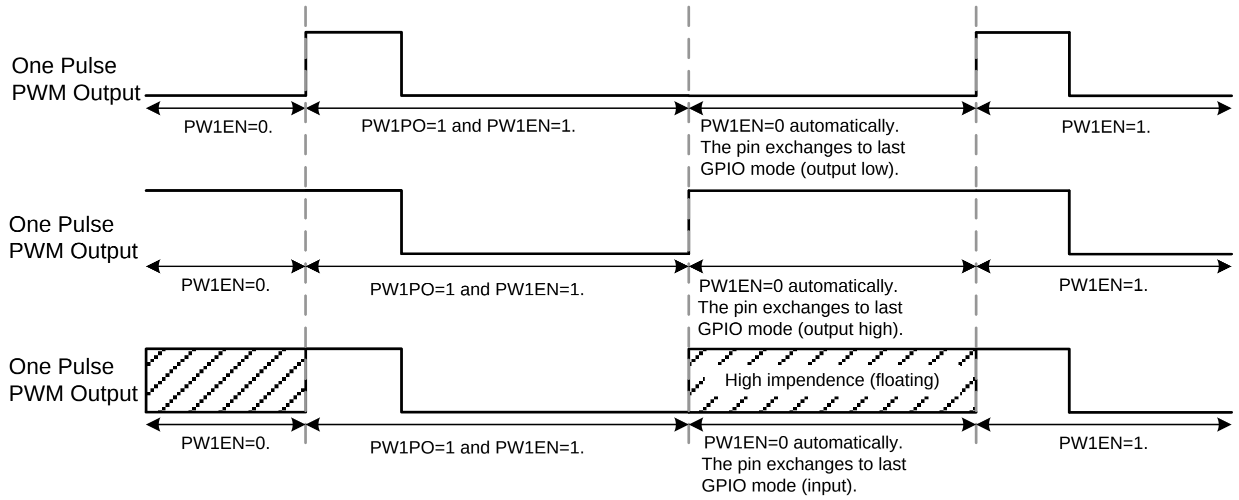
PWM 周期 = PW1Y

PWM 占空比 = PW1D：PW1Y



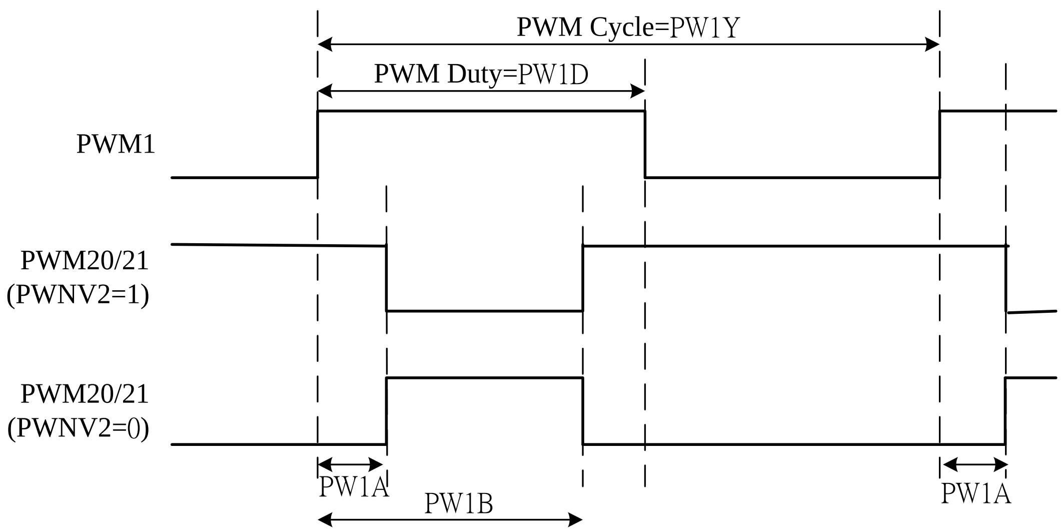
15.2 单脉冲 PWM

PW1PO = 0 时，PW1 为 PWM 功能模式。PW1PO = 1 且 PW1EN=1 时，PW1 将输出单脉冲 PWM，同时随着 PW1 计数器溢出而置位 PWM1F。PW1EN 自动清零，脉冲输出引脚返回到上一个 IDLE 状态。若要输出下一个脉冲，则要通过程序重新设置 PW1EN 位为 1。单脉冲 PWM 通道由 PW1CH 寄存器进行选择。输出单脉冲 PWM 时，必须设置 PW1PO=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。单脉冲 PWM 输出完成后，PW1EN=0 时，PWM 通道返回上一个 GPIO 模式。



15.3 反向和死区

PWM 内置反向输出功能，PWNVn=1 时，PWM 有一个反向 PWM 信号。PWNVn=1 时，PW1 输出 PWM1 的反向 PWM 信号；PWNV=0 时，PW1 输出 PWM1 的正常 PWM 信号。反向 PWM 输出波形如下图所示：



PWM 的死区是在 PWM 高脉冲宽度区设置的，其死区周期由 PW1A 和 PW1D-PW1B 寄存器编程控制。PWM 脉冲宽度两边的死区时间可以设置为对称或不对称。如死区周期的长度大于 PWM 的占空比，就不输出 PWM。

15.4 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	PWNV2	PWNV1	-	PW1PO
PW1CH	-	-	PWCH21	PWCH20	-	-	PWCH11	PWCH10
PW1YH	PW1Y15	PW1Y14	PW1Y13	PW1Y12	PW1Y11	PW1Y10	PW1Y9	PW1Y8
PW1YL	PW1Y7	PW1Y6	PW1Y5	PW1Y4	PW1Y3	PW1Y2	PW1Y1	PW10
PW1BH	PW1B15	PW1B14	PW1B13	PW1B12	PW1B11	PW1B10	PW1B9	PW1B8
PW1BL	PW1B7	PW1B6	PW1B5	PW1B4	PW1B3	PW1B2	PW1B1	PW1B0
PW1DH	PW1D15	PW1D14	PW1D13	PW1D12	PW1D11	PW1D10	PW1D9	PW1D8
PW1DL	PW1D7	PW1D6	PW1D5	PW1D4	PW1D3	PW1D2	PW1D1	PW1D0
PW1A	PW1A7	PW1A6	PW1A5	PW1A4	PW1A3	PW1A2	PW1A1	PW1A0

PW1M Registers (PW1M: 0xAB)

Bit	Field	Type	Initial	Description
7	PW1EN	R/W	0	PW1 功能控制位。 0：禁止； 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	PWNV2	R/W	0	PWM20/21 引脚输出控制位。 0：正常输出； 1：反向输出。
2	PWNV1	R/W	0	PWM10/11 引脚输出控制位。 0：正常输出； 1：反向输出。

1	Reserved	R/W	0	
0	PW1PO	R/W	0	单脉冲功能。 0：禁止； 1：使能。

*周期设置 0x0000 时，PWM 设置使能位后，PWM 将停止，周期不再更新。

PW1CH Register (0xBE)

Bit	Field	Type	Initial	Description
7..6	Reserved	R/W	0	
5 4	PWCH21 PWCH20	R/W	0	PWM1 共用引脚控制位。 0：GPIO； 1：PWM 输出（与 P0.6/ P1.5 共用）。
3..2	Reserved	R/W	0	
1 0	PWCH11 PWCH10	R/W	0	PWM1 共用引脚控制位。 0：GPIO； 1：PWM 输出（与 P0.5/ P1.4 共用）。

PW1YH/PW1YL Registers (PW1YH: 0xAD, PW1YL: 0xAC)

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

*在开始 PWM 功能之前，周期配置必须设置完成。

PW1DH/PW1DL Registers (PW1DH: 0xBC, PW1DL: 0xBB)

Bit	Field	Type	Initial	Description
7..0	PW1DH/L	R/W	0x00	16 位 PWM1 占空比控制位。

PW1BH/PW1BL Registers (PW1BH: 0xAF, PW1BL: 0xAE)

Bit	Field	Type	Initial	Description
7..0	PW1BH/L	R/W	0x00	16 位 PWM1 死区控制位。

PW1A Registers (PW1A: 0xBD)

Bit	Field	Type	Initial	Description
7..0	PW1A	R/W	0x00	8 位 PWM1 死区控制位。

IEN0 Register (0xA8)

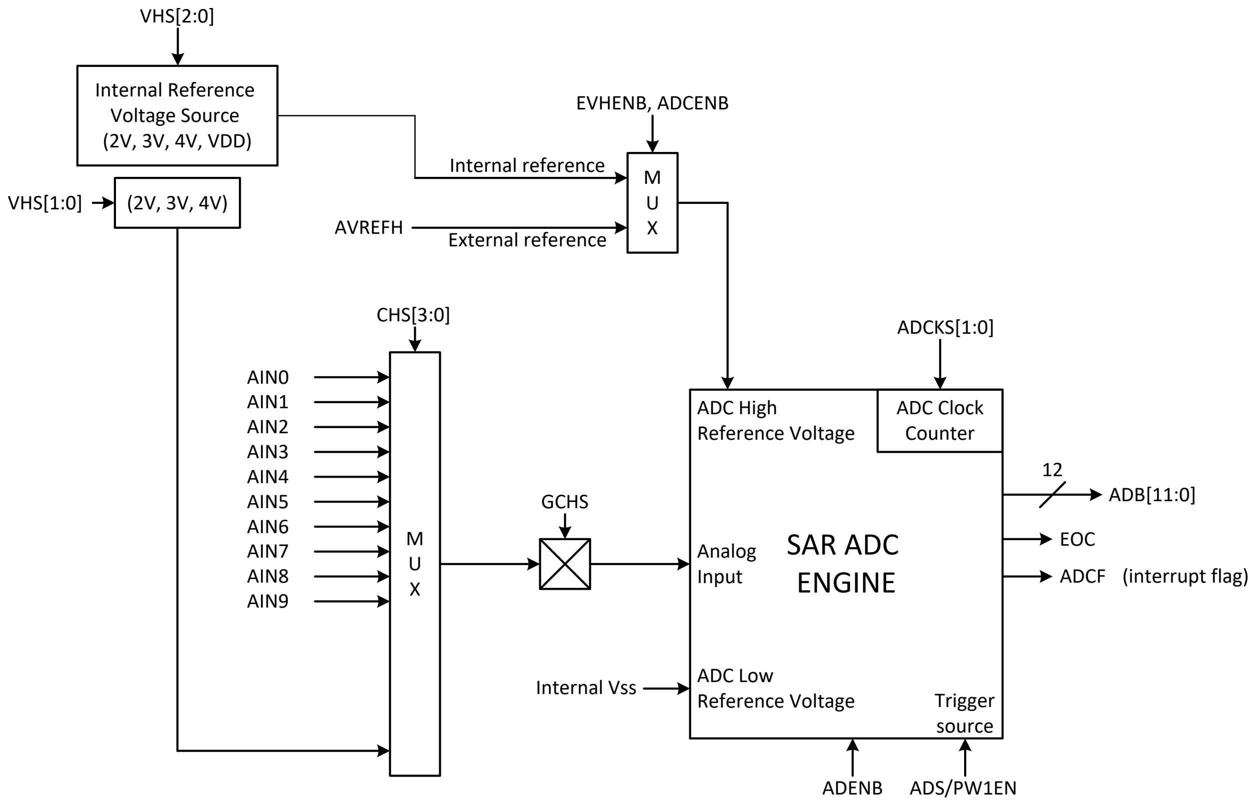
Bit	Field	Type	Initial	Description
7	EAL	R/W	0	所有中断使能控制位。参考其它章节。
Else				参考其它章节。

IEN4 Register (0XD1)

Bit	Field	Type	Initial	Description
7	EPWM1	R/W	0	PWM1 中断控制位。 0：禁止； 1：使能。
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。
Else	Reserved	R	0	

16 ADC

模拟数字转换（ADC）是一个 SAR 结构，内置 10 个输入通道（AIN0~AIN9），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 10 个通道输入源，可以测量 10 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 种内部参考源（Vdd、4V、3V 和 2V）和外部参考源（由 AVREFH 引脚提供）。ADC 内置 P2CON/P1CON 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。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。



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[3:0]位和 GCHS 位设置）；
2. ADC 输入通道的 GPIO 模式应设置为输入模式（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（由 VREFH 寄存器设置）；
6. 选择 ADC 时钟频率（由 ADCKS[1:0]位设置）；
7. 设置 ADENB 位后，ADC 准备开始转换模拟信号为数字数据。

由 ADENB 位使能 ADC 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 开始将模拟信号转换为数字信号。转换结束后，ADS 复位为逻辑 0，ADC 电路将 EOC 和 ADCF 位置 1，并将转换结果存入 ADB 和 ADR 寄存器中。若使能 ADC 中断（EADC=1），ADC 请求中断，ADC 完成后，ADCF=1 时执行中断服务程序。中断时必须由程序将 ADCF 清零。ADPWS=1 时，若使能 PWM 作为转换源，则 ADC 继续转换直至禁止 PWM。

16.2 ADC 输入通道

SN8F5702A 的 ADC 内置 10 个输入通道（AIN0~AIN9），用于测量 10 种不同的模拟信号源，由 CHS[3:0] 和 GCHS 位控制。AIN10 则是 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，AIN10 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。

ADC 输入通道：

CHS[3:0]	Channel	Pin name	备注
0000	AIN0	P1.0	-
0001	AIN1	P1.1	-
0010	AIN2	P1.2	-
0011	AIN3	P1.3	-
0100	AIN4	P1.4	-
0101	AIN5	P1.5	-
0110	AIN6	P1.6	-
0111	AIN7	P1.7	-
1000	AIN8	P2.1	-
1001	AIN9	P2.0	-
1010	AIN10	Internal 2V or 3V or 4V	电池检测器通道
1011 – 1111	-	-	保留

16.2.1 引脚配置

ADC 输入通道引脚与 P1 和 P2 引脚共用，由 CHS[3:0]选择控制。在同一时间，只能设置 P1 和 P2 其中的一个引脚为 ADC 输入通道引脚，P1 和 P2 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，并先要由程序使能 P1CON/P2CON。通过 CHS[3:0]选择 ADC 输入通道后，GCHS 位设置为 1，使能 ADC 通道功能。

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 结构端口时，尤其当模拟信号为 1/2 VDD 时，可能产生额外的漏电流。STOP 模式下，上述漏电流会严重影响到系统的整体功耗。当 P1 和 P2 输入多个模拟信号时，也会产生额外的漏电流。将 PnCON 置 1，其对应的引脚将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 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 参考高电压由内部电压源提供，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。外部高参考电压源的限制条件为，最高为 VDD，最低为内部最低电平，否则默认为 VDD。若选择 AIN10 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（不可更改）；高电压包括 VDD/4V/3V/2V 和外部参考电压（由 P1.0/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 参考低电压范围为 VSS=0V，故 ADC 参考高电压范围为 2V~VDD，外部参考高电压也要在此范围内。

- 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 的时钟源为 Fosc，包括 Fosc/1，Fosc/2，Fosc8，Fosc/16，由 ADCKS[1:0]位控制。

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。

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

ADC 转换时间：

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

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，ADB 寄存器存放结果的高字节（bit4~bit11），ADR（ADR[3:0]）存放低字节（bit0~bit3）。ADC 寄存器是 8 位寄存器，包括位 4~位 11 的 ADC 数据。将 ADB 寄存器和 ADR 的低字节相组合，会得到完整的 12 位 ADC 数据缓存器。ADC 数据缓存器是只读寄存器，且初始状态系统复位后是未知的。

表 16-1 AIN 输入电压 vs. ADB 输出数据

AIN n	ADB 11	ADB 10	ADB 9	ADB 8	ADB 7	ADB 6	ADB 5	ADB 4	ADB 3	ADB 2	ADB 1	ADB 0
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	-	CHS3	CHS2	CHS1	CHS0
ADB	ADB11	ADB10	ADB9	ADB8	ADB7	ADB6	ADB5	ADB4
ADR	-	GCHS	ADCKS1	ADCKS0	ADB3	ADB2	ADB1	ADB0
VREFH	EVHENB	-	-	ADPWS	-	VHS2	VHS1	VHS0
P1CON	P1CON7	P1CON6	P1CON5	P1CON4	P1CON3	P1CON2	P1CON1	P1CON0
P2CON	-	-	-	-	-	-	P2CON1	P2CON0
IEN2	-	-	-	-	-	-	EADC	-
IRCON2	-	-	-		-	-	-	ADCF

ADM Register (0xD2)

Bit	Field	Type	Initial	Description
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（硬件自动设置为 1，软件手动清零）。
4	Reserved	R	0	
3..0	CHS[3:0]	R/W	0x00	ADC 输入通道选择位。 0000：AIN0；0001：AIN1； 0010：AIN2；0011：AIN3； 0100：AIN4；0101：AIN5； 0110：AIN6；0111：AIN7； 1000：AIN8；1001：AIN9； 1010：AIN10*(1)；其它：保留；

*(1) AIN10 为内部 2V/3V/4V 输入通道，没有外部信号输入，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 Register (0xD3)

Bit	Field	Type	Initial	Description
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 ADC 转换结果，其高字节存入 ADB 寄存器，低字节存入 ADR[3:0]位。

ADR Register (0xD4)

Bit	Field	Type	Initial	Description
6	GCHS	R/W	0	ADC 全局通道控制位。 0：禁止 AIN 通道； 1：使能 AIN 通道。
5..4	ADCKS[1:0]	R/W	00	ADC 时钟源选择位。 00 = Fosc/16；01 = Fosc/8；10 = Fosc/1；11 = Fosc/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 ADC 转换结果，其高字节存入 ADB 寄存器，低字节存入 ADR[3:0]位。

VREFH Register (0xD5)

Bit	Field	Type	Initial	Description
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。
4	ADPWS	R/W	0	PWM 触发 ADC 开始控制位。 0：禁止 PWM 触发 ADC 开始； 1：使能 PWM 触发 ADC 开始。
2	VHS[2]	R/W	0	ADC 内部参考高压选择位（AIN10）。 0：由 VHS[1:0]*(2)选择 ADC 内部 VREFH 功能； 1：ADC 内部 VREFH 功能为内部 VDD。
1..0	VHS[1:0]	R/W	00	ADC 内部参考高压选择位。 000：2.0V； 001：3.0V； 010：4.0V； 011：VDD。 100：VDD ，AIN10=2.0V； 101：VDD ，AIN10=3.0V； 110：VDD ，AIN10=4.0V； 其它：保留。

*(1) AVREFH 的电压值必须在 VDD 到 2.0V 范围内。

*(2) AIN10 为内部 2V/3V/4V 输入通道，没有外部信号输入，ADC 参考高电压必须为内部 VDD 或外部电压，不能是内部 2V/3V/4V。

P1CON Register (0xD6)

Bit	Field	Type	Initial	Description
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。

* 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 (0x9E)

Bit	Field	Type	Initial	Description
1..0	P2CON[1:0]	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。

* 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。

IEN2 Register (0x9A)

Bit	Field	Type	Initial	Description
1	EADC	R/W	0	ADC 中断控制位。 0：禁止； 1：使能。
Else				请参考其它章节。

IRCON2 Register (0xBF)

Bit	Field	Type	Initial	Description
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				请参考其它章节。

17 UART

UART（通用同步异步收发器）提供 0.25MHz 灵活的全双工传输模式。UART 共有 4 种操作模式：一种同步，3 种异步的。同步模式发送 8 位数据，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。ES0 是 UART0 传输中断功能控制位。ES0 控制 UART 的发送和接收功能。ES0 = 0，禁用收发器的中断功能。ES0 = 1，启用 UART 收发器的中断功能。UART 的发送和接收中断功能共用中断向量 0x0023。UART 中断功能使能时，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 UART0 中断请求标志，也是 UART 操作状态指示符。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 Rate</th><th>起始位</th><th>数据位</th><th>停止位</th><th>UART 引脚模式和数据</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：
P05M=1 和 P05=1
URX 引脚：
发送器：
P06M=1 和 P06=1
接收器：
P06M=0 和 P06=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：
P05M=1 和 P05=1
URX 引脚：
P06M=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

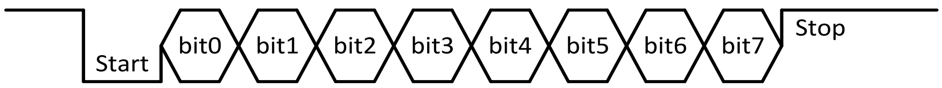
模式 0 是移位寄存器模式，作为同步收发器。UTX 引脚都输出移位时钟用于传输和接收状态。URX 引脚用于传输和接收数据。首先，发送/接收带 LSB 的 8 位数据。波特率为 Fcpu/12。通过写数据到 S0BUF 寄存器开始传输。第 8 位传输结束后，设置 TI0 标志位。通过设置 REN0 位来控制接收，清 RI0 位。REN0=1 和 RI0 由 1 到 0 时，开始传输数据，且在接收完第 8 位时，设置 RI0 标志位。

17.3 模式 1：异步 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。

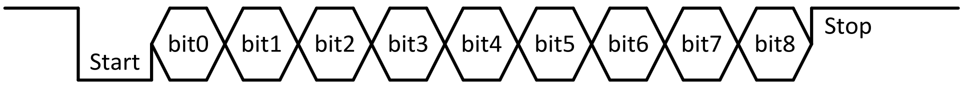


17.4 模式 2：固定波特率异步 9 位收发器

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。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。

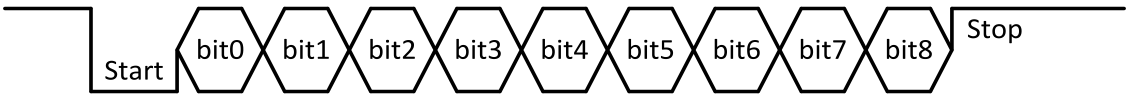


17.5 模式 3：可变波特率 9 位收发器

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，URX0 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，低 8 位输入数据存储在 S0BUF 寄存器中，第 9 位存储在 RB80 中。



17.6 多处理器通讯

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 SMOD 寄存器选择控制，其选项分别为 FCPU/64SMOD=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：

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

- 表 17-1 UART 波特率的建议设置如下表（Fcpu=8MHz）：

波特率	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。

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

● 表 17-2 建议设置 T1 溢出间隔时间（T1 时钟=32M），用于 UART 波特率（Fcpu=8MHz）

波特率	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 模块带有时钟 gating 功能以省电。REN0=0 时，停止 UART 模块内部时钟以减少功耗，此时不能访问 UART 相关寄存器（S0CON、S0CON2、S0BUF、S0RELL、S0RELH 和 SNOD）。

反之，REN0=1 时，UART 模块内部时钟正常运行，也能正常访问相关寄存器。初始化 UART 之前，REN0 位必须置为 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	-	-	-	-	-	-	-
S0BUF	S0BUF7	S0BUF6	S0BUF5	S0BUF4	S0BUF3	S0BUF2	S0BUF1	S0BUF0
PCON	SMOD	-	-	-	P2SEL	GF0	STOP	IDLE
S0RELH	-	-	-	-	-	-	S0REL9	S0REL8
S0RELL	S0REL7	S0REL6	S0REL5	S0REL4	S0REL3	S0REL2	S0REL1	R0REL0
IEN0	EAL	-	ET2	ES0	ET1	EX1	ET0	EX0
P0OC	-	-	-	P15OC	P14OC	P13OC	P06OC	P05OC
P0M	P07M	P06M	P05M	P04M	P03M	P02M	P01M	P00M
P0	P07	P06	P05	P04	P03	P02	P01	P00

S0CON Register (0x98)

Bit	Field	Type	Initial	Description
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（以及接收功能） 0：禁止，以省电*； 1：使能，以进行 URAT 操作。
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，模块内部时钟停止。

✸ *注：中断使能，通过软件清空* ***TI0*** *和* ***RI0****。*

S0CON2 Register (0xD8)

Bit	Field	Type	Initial	Description
7	BD	R/W	0	波特率产生器选择位（模式 1、3）。 0：T1 溢出周期； 1：由寄存器 S0RELH/S0RELL 控制。
6..0	Reserved	R	0x00	

S0BUF Register (0x99)

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

PCON Register (0x87)

Bit	Field	Type	Initial	Description
7	SMOD	R/W	0	UART 波特率控制位。 模式 0：无效； 模式 1，3：波特率计算公式如 17.7 章(波特率控制)所示； 模式 2： 0：Fcpu/64； 1：Fcpu/32。
6..0				请参考其它章节。

IEN0 Register (0xA8)

Bit	Field	Type	Initial	Description
7	EAL	R/W	0	使能中断，请参考中断章节。
4	ES0	R/W	0	使能 UART 中断。
Else				请参考其它章节。

P0OC Register (0xE4)

Bit	Field	Type	Initial	Description
1	P06OC	R/W	0	0：切换 P0.6（URX）为输入模式（要求）； 1：切换 P0.6（URX）为开漏模式*。
0	P05OC	R/W	0	0：切换 P0.5（UTX）为推拉模式； 1：切换 P0.5（UTX）为开漏模式。
Else				请参考其它章节。

*设置 P06OC 为高电平会导致 URX 不能接收数据。

P0M Register (0xF9)

Bit	Field	Type	Initial	Description
6	P06M	R/W	0	0：设置 P0.6（URX）为输入模式（要求）； 1：设置 P0.6（URX）为输出模式*。
5	P05M	R/W	0	0：设置 P0.5（UTX）为输入模式*； 1：设置 P0.5（UTX）为输出模式（要求）。
Else				请参考其它章节。

*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 Register (0x80)

Bit	Field	Type	Initial	Description
6	P06	R/W	0	随时读取该位，以监控总线状态。
5	P05	R/W	0	0：设置 P0.5（UTX）始终为低电平*； 1：P0.5（UTX）输出 UART 数据（要求）。
Else				请参考其它章节。

* 初始化 P05 时一定需要设置为高电平，因为 P05 输出低电平会认为 UART 通信开始，造成会发出一包错误的数据。

18 SPI

SPI 是一个串行通信接口，用于单片机之间或其他硬件外围设备的数据交换。它是一个简单的 8 位接口，没有保护定义协议、包或控制位。SPI 收发机包括三个引脚、时钟(SCK)、数据输入和数据输出(MISO/MOSI)，用于在主终端和从终端之间发送数据。从机模式下可通过寄存器使能使能片选引脚（SSN）。SPI 接口构建 4 种模式，即时钟空闲状态和时钟阶段。

- 全双工，3 线同步数据传输。
- 主机 (SCK 是 clock 输出) or 从机 (SCK 是 clock 输入) 操作。
- 7 个 SPI 主波特率。
- 从机时钟高达 fcpu/8.
- 8-bit 数据传输 MSB first, LSB last.
- 具有可编程极性和相位的串行时钟。
- 主机模式在 MCU 中断产生错误标志。
- 写冲突标志保护。

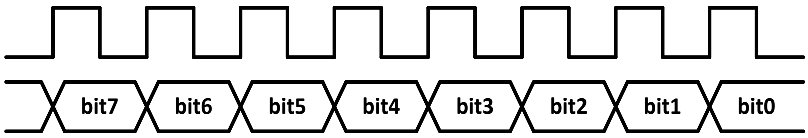
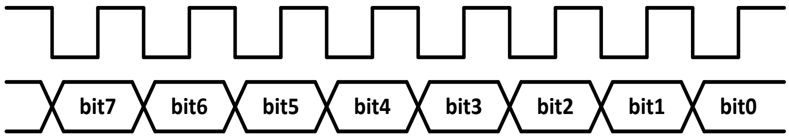
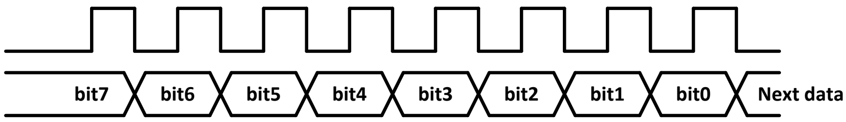
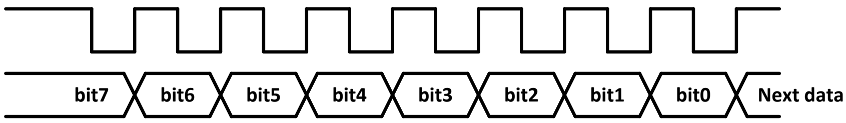
18.1 SPI 操作

SPCON 寄存器能控制 SPI 操作： 例如:发送/接收，时钟速率，数据传输方向, SPI 时钟空闲状态和时钟控制相位等，并室内此电路。SPI 电路通过在 SPCON 寄存器中设置 SPEN 和读写 SPDAT 寄存器来自动传输或接收 8 位数据。

CPOL 位用于控制 SPI 时钟空闲状态。CPHA 位用于控制时钟接收数据的边缘方向。CPOL 和 CPHA 位决定 SPI 格式。SPI 数据传输方向为 MSB 位到 LSB 位。

SPI 支持 CPOL 和 CPHA 位控制的 4 模式格式。边沿方向是“数据传输边沿”。当设置上升边时，意味着在 SCK 上升沿接收和传输 1bit 数据，在 SCK 下降沿转换数据。设置下降沿时，即在 SCK 下降沿接收和传输 1bit 数据，在 SCK 上升沿转换数据。

“CPHA”是时钟相位位，控制采样数据所在时钟的相位。当 CPHA=1 时，SCK 的第一个边沿用于转换数据，接收和传输数据位于 SCK 的第二个边。当 CPHA=0 时，第 1bit 已经固定，SCK 的第一个边沿是接收和传输数据。SPI 数据传输时序如下图所示：

CPOL	CPHA	Diagrams	Description
0	1		SCK 低电平空闲 传输起始位 MSB 下降沿锁存数据
1	1		SCK 高电平空闲 传输起始位 MSB 上升沿锁存数据
0	0		SCK 低电平空闲 传输起始位 MSB 上升沿锁存数据
1	0		SCK 高电平空闲 传输起始位 MSB 下降沿锁存数据

SPI 支持中断功能。ESPI 是 SPI 中断功能控制位。ESPI=0，禁用 SPI 中断。ESPI=1，使能 SPI 中断。当 SPI 中断使能时，程序计数器指向中断向量，在 SPI 运行后执行 SPI 中断服务程序。SPIF 是 SPI 中断请求标志，当 ESPI= 0 时也是 SPI 状态指示，但通过读取 SPSTA、SPDAT 寄存器清除。

SPI 内置芯片选择功能，实现 SPI 多设备模式。在 SPI 总线中，一个主设备与多个从设备通信，芯片的选择决定了所选设备的指向。芯片选择引脚为 SSN 引脚。

SPI 引脚也支持开漏结构。此选项由 PnOC 位控制。当 PnOC=0 时，禁用 SPI 开漏结构。当 PnOC=1 时，使能 SPI 开漏结构。如果使能开漏结构，SPI 引脚必须设置输入模式，并需要外部上拉电阻。

18.2 SPI 主控模式

SPI 主控制模式共有 7 种类型的时钟发生器，Fcpu/2~Fcpu/128，产生的时钟由 SCK 引脚（与 P1.3 共用）输出，其 IDLE 状态由 CPOL 控制。

输入输出数据的相位由 CPHA 寄存器自动指定。主控模式下，MOSI 引脚（与 P1.4 共用）负责输出数据，MISO 引脚（与 P1.5 共用）负责获取来自从机设备的数据。通过写入数据到 SPDAT 寄存器开始 SPI 通讯；数据发送完成后，从 MISO 引脚读取接收到的数据。

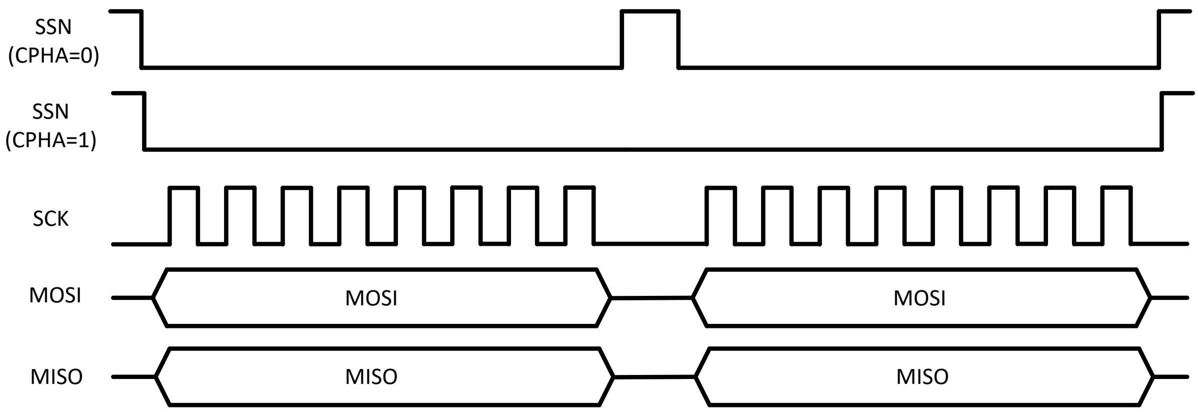
主控模式下，有 2 种包含中断功能的状态标志：

- SPIF 寄存器显示一个字节数据通讯的结束，若此时使能 ESPI 位则释放中断。
- 发送时由 SSN（与 P0.2 共用）低电平状态释放 MODF，通过设置 SSDIS 位屏蔽中断源。

18.3 SPI 从动模式

SPI 从动模式监控 SCK 引脚控制 MISO 和 MOSI 通讯，但其最大时钟率限制在 Fcpu/8。在连接到 SPI 总线的配置相同时，从动设备可用于指定 CPOL 和 CPHA 设置。

输入数据时可将从动模式看作是 MOSI 引脚，发送数据时可看作是 MISO 引脚。默认情况下，SSDIS 寄存器为低电平状态，即从动选择引脚（SSN）是有功能的。若 SSN 引脚为低电平状态，则处理 SPI 通讯；而 SSN 为高电平状态时，从动设备挂起。但是在 CPHA= 0 中，SSN 必须严格遵循每一个 8 位数据都需要包含下降边和上升边，而 CPHA=1 不受限制。



从动模式下，有 2 种包含中断功能的状态标志：

- SPIF 寄存器显示一字节数据通讯的结束，发送 SPDAT 的初始值后，再通过 SPDAT 读取从 MOSI 接收的到的数据。
- MODF 显示的是：在完成一字节数据通讯之前改变从动选择引脚 SSN 为高电平，换句话说就是打断最后一次 SPI 通讯。

18.4 省电

SPI 模块带有时钟 gating 功能以省电。SPEN=0 时，停止 SPI 模块内部时钟以减少功耗，此时不能访问 SPI 相关寄存器（SPCON、SPSTA 和 SPDAT）；反之，SPEN=1 时，SPI 模块内部时钟正常运行，也能正常访问相关寄存器。初始化 SPI 之前，SPEN 位必须置为 1。

18.5 SPI 寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPCON	SPR2	SPEN	SSDIS	MATR	CPOL	CPHA	SPR1	SPR0
SPSTA	SPIF	WCOL	SSERR	MODF	-	-	-	-
SPDAT	SPDAT7	SPDAT6	SPDAT5	SPDAT4	SPDAT3	SPDAT2	SPDAT1	SPDAT0
IEN0	EAL	-	ET2	ES0	ET1	-	ET0	EX0
IEN1	ET2RL	-	ET2C3	ET2C2	ET2C1	ET2C0	ESPI	EI2C
P0OC	-	-	-	P15OC	P14OC	P13OC	P06OC	P05OC
P1M	P17M	P16M	P15M	P14M	P13M	P12M	P11M	P10M

SPCON Register (0xE2)

Bit	Field	Type	Initial	Description
7,1,0	SM[2:0]	R/W	000	SPI 波特率发生器（仅在主控模式下有效）。 000：Fcpu/2； 001：Fcpu/4； 010：Fcpu/8； 011：Fcpu/16； 100：Fcpu/32； 101：Fcpu/64； 110：Fcpu/128； 111：保留。
6	SPEN	R/W	0	SPI 通讯功能。 0：禁止，以省电*； 1：使能，以进行 SPI 操作。
5	SSDIS	R/W	0	从动选择引脚功能（仅在 MSTR = 0，CPHA = 0 时有效）。 0：使能从动选择引脚 SSN 功能； 1：禁止从动选择引脚 SSN 功能。
4	MSTR	R/W	1	SPI 模式选择位。 0：从动模式； 1：主控模式。

3	CPOL	R/W	0	SCK 引脚空闲状态位。 0：SCK 低电平空闲； 1：SCK 高电平空闲
2	CPHA	R/W	1	数据的时钟相位锁存控制位。 0：由第一个时钟沿锁存数据； 1：由第二个时钟沿锁存数据。

*SPEN=0 时，不能访问 SPI 相关寄存器，且模块内部时钟停止。

SPSTA Register (0xE1)

Bit	Field	Type	Initial	Description
7	SPIF	R	0	SPI 通讯完成标志位。 在通讯结束时自动设置为 1； 通过读取 SPSTA，SPDAT 寄存器自动清零。
6	WCOL	R	0	写操作冲突标志位。 在通讯时对 SPDAT 执行写操作自动设置为 1； 通过读取 SPSTA，SPDAT 寄存器自动清零。
5	SSERR	R	0	同步从动选择引脚错误位。 若 SSN 错误控制时自动设置为 1； 通过清 SPEN 自动清零。
4	MODF	R	0	模式错误标志位。
3..0	Reserved	R	0x00	

SPDAT Register (0xE3)

Bit	Field	Type	Initial	Description
7..0	SPDAT	R/W	0x00	主控模式：写操作触发 SPI 通讯；接收的数据可在一字节通讯结束时读取（SPIF 自动设置为 1）。 从动模式：通过 SCK 输入发送写入的数据；接收的数据可在一字节通讯结束时读取（SPIF 自动设置为 1）。

IEN0 Register (0xA8)

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

IEN1 Register (0xB8)

Bit	Field	Type	Initial	Description
1	ESPI	R/W	0	使能 SPI 中断。
Else				请参考其它章节。

P0OC Register (0xE4)

Bit	Field	Type	Initial	Description
4	P15OC	R/W	0	0：切换 P1.5（MISO）为输入或输出模式； 1：切换 P1.5（MISO）为开漏模式。
3	P14OC	R/W	0	0：切换 P1.4（MOSI）为输入或输出模式； 1：切换 P1.4（MOSI）为开漏模式。
2	P13OC	R/W	0	0：切换 P1.3（SCK）为输入或输出模式； 1：切换 P1.3（SCK）为开漏模式。
Else				请参考其它章节。

P1M Register (0xFA)

Bit	Field	Type	Initial	Description
5	P15M	R/W	0	0：设置 P1.5（MISO）为输入模式[从控模式]； 1：设置 P1.5（MISO）为输出模式[主动模式]。
4	P14M	R/W	0	0：设置 P1.4（MOSI）为输入模式[从动模式]； 1：设置 P1.4（MOSI）为输出模式[主控模式]。
3	P13M	R/W	0	0：设置 P1.3（SCK）为输入模式[从动模式]； 1：设置 P1.3（SCK）为输出模式[主控模式]。
Else				请参考其它章节。

1 在从动模式下本就要设置 SCK 为输入模式；在主控模式下建议设置为输出模式。

2 在主控模式下本就要设置 MISO 为输入模式；在从动模式下建议设置为输出模式。

3 在从动模式下本就要设置 MOSI 为输入模式；在主控模式下建议设置为输出模式。

P0M Register (0xF9)

Bit	Field	Type	Initial	Description
2	P02M	R/W	0	0：设置 P0.2（SSN）为输入模式*；。 1：设置 P0.2（SSN）为输出模式*。

*若从动模式下使能 SSN 功能：本就是要设置 SSN 为输入模式。

19 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。

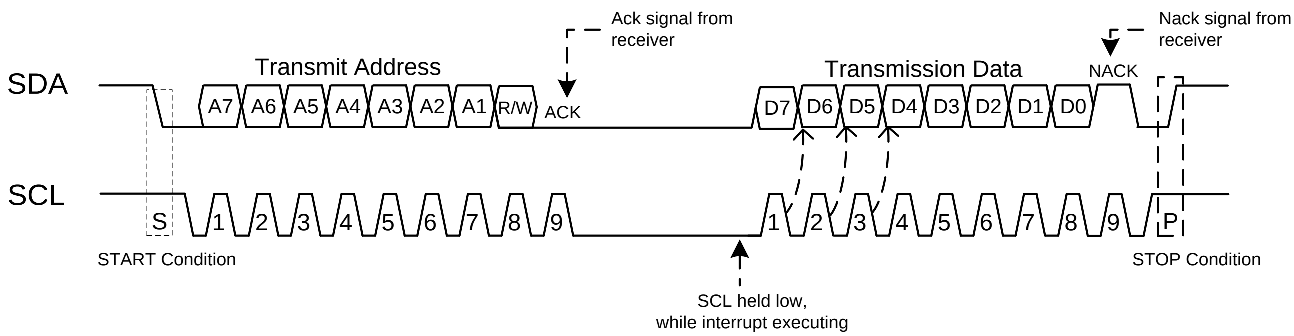
- 主 Tx, Rx 模式
- 从 Tx, Rx 模式（通用地址调用），以在单主机情况下进行 multiplex slave
- 双线同步数据收发器
- 支持 100K/400K 时钟率

19.1 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。

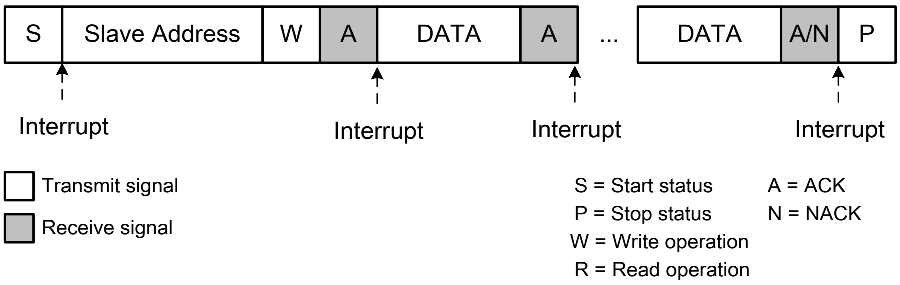


19.2 I2C 传输模式

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。

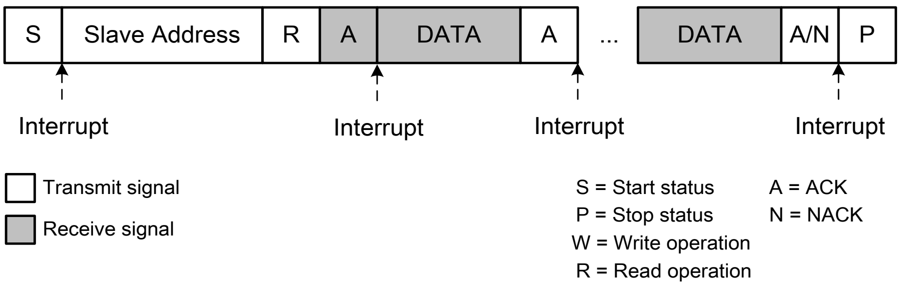
19.2.1 主机发送模式

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。



19.2.2 主机接收模式

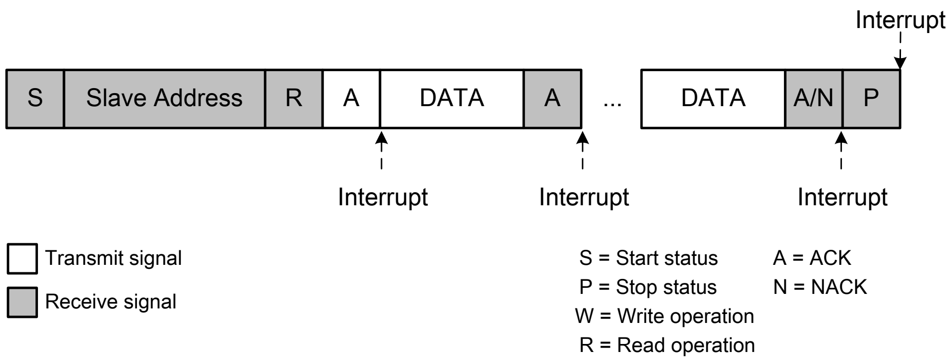
主机接收模式为主机接收来自从机的数据，串行时钟由 SCL 输出时，串行数据由 SDA 输入。主机通过发送 START 信号来开始数据接收。发送 START 信号后，开始发送从机设备指定的地址字节。地址字节包含 7 位地址位，第 8 位为数据方向位（R/W），该位设为 1 时使能主机接收。接下来，主机接收来自从机的一个或多个数据字节，每接收一个字节之后，通过 AA 标志位来将 ACK 或 NACK 信号发送给从机。最后，主机产生 STOP 信号来结束此次数据传输。



19.2.3 从机发送模式

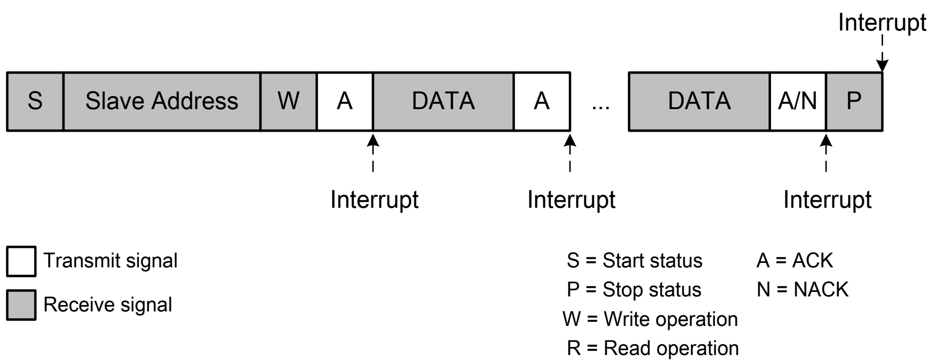
从机发送模式为从机发送数据给主机，串行时钟由 SCL 输入时，串行数据由 SDA 输出。接收到来自主机的 START 信号后开始数据发送。接收到 START 信号后，开始接收从机设备指定的地址字节。地址字

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。



19.2.4 从机接收模式

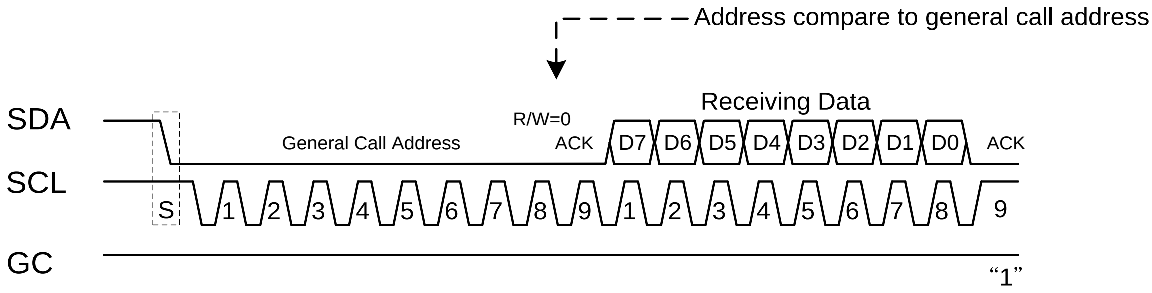
从机接收模式为从机接收来自主机的数据，串行时钟和串行数据都由 SCL 和 SDA 输入。接收到来自主机的 START 信号后开始数据接收。接收到 START 信号后，开始接收从机设备指定的地址字节。地址字节包含 7 位地址位，第 8 位为数据方向位（R/W），该位设为 0 时使能从机接收。若接收到的地址字节与 I2CADR 寄存器中地址相匹配，从机将产生 ACK 信号；另外，若广播呼叫地址标志位设为 1（GC=1），接收到广播呼叫地址（0x00）后，从机设备也会产生一个 ACK 信号。接下来，从机接收一个或多个来自主机的数据字节，每接收一个字节之后，从机产生 ACK 或 NACK 信号，通过 AA 标志位来将 ACK 或 NACK 信号发送给主机。最后，从机接收从主机产生 STOP 信号来结束此次数据传输。



19.3 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。



19.4 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。

$$\text{SCL Clock Rate} = \frac{\text{Fcpu}}{\text{Prescaler}} (\text{Prescaler} = 256\sim 60)$$

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 Prescaler	Bit 频率 (kHz) 6MHz	8MHz
0	0	0	256	23	31
0	0	1	224	27	36
0	1	0	192	31	42
0	1	1	160	37	50
1	0	0	960	6.25	8
1	0	1	120	50	67
1	1	0	60	100	133
1	1	1	(Timer 1 overflow rate)/8		

✸ *注:*

1．IIC 操作第一步即将 SDA/SCL 引脚设置为输入模式 。

2．当 SCL 时钟源是 T1 定时器的溢出频率时，T1 定时器的最大计时值为 0XFB（只支持 0x00~0xFB），且当 T1 定时器的时钟率是 IHRC_32MHz 时，最大的 SCL 时钟速率是 800KHz。

3．如果用户想要生成的 SCL 时钟频率是 100 kHz/400kHz，可以设置 T1 计数器值为 0XD8/0XF6。

19.5 同步与仲裁

在多主机条件下，同一时间可以有多个主机在总线上传输数据。需要决定由哪个主机可以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。

19.6 系统管理总线（SMBus）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：

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

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。

类型	超时周期	Fcpu=8MHz	
		十进制	十六进制
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	Description
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	Description
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

19.7 省电

I2C 模块带有时钟 gating 功能以省电。ENS1=0 时，停止 I2C 模块内部时钟以减少功耗，此时不能访问 I2C 相关寄存器（I2CDAT、I2CADR、I2CCON、I2CSTA、SMBSEL 和 SMBDST）；反之，ENS1=1 时，I2C 内部时钟正常运行，也能正常访问相关寄存器。初始化 I2C 之前，ENS1 位必须置为 1。

19.8 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	-	-	-	-	SMBSTP2	SMBSTP1	SMBSTP0
SMBDST	SMBD7	SMBD6	SMBD5	SMBD4	SMBD3	SMBD2	SMBD1	SMBD0
IEN0	EAL	-	ET2	ES0	ET1	-	ET0	EX0
IEN1	ET2RL	-	ET2C3	ET2C2	ET2C1	ET2C0	ESPI	EI2C
P0M	P07M	P16M	P05M	P04M	P03M	P02M	P01M	P00M
P0	P07	P16	P05	P04	P03	P02	P01	P00

I2CDAT Register (0xDA)

Bit	Field	Type	Initial	Description
7:0	I2CDAT[7:0]	R/W	0x00	I2CDAT 寄存器存储的是要通过 I2C 总线发送出去的一字节数据，或者是刚从 I2C 总线上接收到的一字节数据。当它不在字节移位的过程中时，CPU 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 Register (0xDB)

Bit	Field	Type	Initial	Description
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 Register (0xDC)

Bit	Field	Type	Initial	Description
7,1,0	CR[2:0]	R/W	0	I2C 时钟 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 使能位。 0：禁止，以省电*； 1：使能，以进行 I2C 操作。
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，不能访问 I2C 相关寄存器，且模块内部时钟停止。

I2CSTA Register (0xDD)

Bit	Field	Type	Initial	Description
7:3	I2CSTA[7:3]	R	11111	I2C 状态代码。
2..0	I2CSTA[2:0]	R	000	SMBus 状态代码。

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；	无动作	0	0	0	0	接收数据字节；返回 NACK。

		并已接收 ACK	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 时仲裁丢失；并接收到了对应的 SLA+W，已返回 ACK	无动作	X	0	0	0/1	接收数据字节；返回 NACK/ACK
	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	X	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 Register (0xDE)

Bit	Field	Type	Initial	Description
7	SMBEXE	R/W	0	SMBus 扩展功能。 0：禁止； 1：使能。
2..0	SMBSTP[2:0]	R/W	000	SMBus 超时寄存器。

SMBDST Register (0xDF)

Bit	Field	Type	Initial	Description
7..0	SMBD[7:0]	R/W	0x00	SMBDST 寄存器是用来提供一个读写 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 Register (0xA8)

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

IEN1 Register (0xB8)

Bit	Field	Type	Initial	Description
0	EI2C	R/W	0	中断使能位。请参考中断章节。
Else				请参考其它章节。

P0M Register (0Xf9)

Bit	Field	Type	Initial	Description
4	P04M	R/W	0	0：时钟 P0.4（SDA）为输入模式（要求）。 ~~1：时钟 P0.4（SDA）为输出模式。*~~
3	P03M	R/W	0	0：时钟 P0.3（SCL）为输入模式（要求）。 ~~1：时钟 P0.3（SCL）为输出入模式。*~~
Else				请参考其它章节。

* 要求 P03M 和 P04M 分别设为输入模式。

20 In-System Program

SN8F5702A 内置 4KB 程序存储（IROM），均分为 128 页（每页 32 字节）。In-System Program 就是使能 FW 能随意更改每页的数据，换句话说，就是存储数据到 Flash 存储器或者现场升级 FW 的一个通道。

0x0FFF 0x0FE0	Page 127
0x0FDF 0x0FC0	Page 126
	…
0x003F 0x0020	Page 1
0x001F 0x0000	Page 0

程序存储器 (IROM)

20.1 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。

<table>
<tr><th colspan="2">ISP ROM MAP</th><th>ROM 地址 bit0~bit4 (hex) =0</th></tr>
<tr><td rowspan="16">ROM address bit5~bit15 (hex)</td><td>0000</td><td rowspan="6">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。</td></tr>
<tr><td>0020</td></tr>
<tr><td>0040</td></tr>
<tr><td>…</td></tr>
<tr><td>00C0</td></tr>
<tr><td>00E0</td></tr>
<tr><td>0100</td><td>一个 ISP 程序页</td></tr>
<tr><td>0120</td><td>一个 ISP 程序页</td></tr>
<tr><td>…</td><td>一个 ISP 程序页</td></tr>
<tr><td>0400</td><td>一个 ISP 程序页</td></tr>
<tr><td>0420</td><td>一个 ISP 程序页</td></tr>
<tr><td>…</td><td>一个 ISP 程序页</td></tr>
<tr><td>0700</td><td>一个 ISP 程序页</td></tr>
<tr><td>0720</td><td>一个 ISP 程序页</td></tr>
<tr><td>…</td><td>一个 ISP 程序页</td></tr>
<tr><td>0FE0</td><td>该页包括 ROM 保留区。强烈建议保留该区，不做 ISP 擦除动作。</td></tr>
</table>

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 0xA5A 到 PECMD[11:0]开始 ISP 功能。在写入 0x5A 到 PECMD[7:0]之前，PECMD[11:8]必须写入 0xA。

5、写 2 个 NOP 指令。

举例来说，假设程序存储器的第 126 页（IROM，0x0FC0~0x0FDF）为计划升级区域，已经存入 IRAM 地址 0x60~0x7F 的内容，执行 In-System Program，只要写入开始 IROM 地址 0x0FC0 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 0xA5A 到 PECMD[11:0]寄存器，复制缓存器的数据到 IROM 的第 126 页。

通常而言，每页的内容都可通过 In-System Program 进行修改，但是第一页和最后一页（page0 和 page127）分别的存储复位向量和上电控制信息，都不可以编程，否则会导致上电错误或复位错误。

> ✸ *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。*

20.2 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	-	PECMD11	PECMD10	PECMD9	PECMD8
PECMD	PECMD7	PECMD6	PECMD5	PECMD4	PECMD3	PECMD2	PECMD1	PECMD0

PERAM Register (0x97)

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

PEROMH Register (0x96)

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

PEROML Register (0x95)

Bit	Field	Type	Initial	Description
7..5	PEROM[7:5]	R/W	000	编程页（IROM）的第一个地址（7th– 5th）。
4	Reserved	R	0	
3..0	PECMD[11:8]	R/W	0x0	0xA：使能 In-System Program； 其它值：禁止 In-System Program*。

*禁止 In-System Program 可避免误触发 ISP 功能。

PECMD Register (0x94)

Bit	Field	Type	Initial	Description
7..0	PECMD[7:0]	W	0x0	0x5A：开始整页编程*(1)。 其它值：保留*(2)。

* 必须先写入 0xA 到 PECMD[11:8]，在写入 0x5A 到 PECMD[7:0]。

*（2）不允许写入其它值到 PECMD 寄存器。

21 电气特性

21.1 极限参数

Voltage applied at VDD to VSS .... - 0.3V to 6.0V
Voltage applied at any pin to VSS .... - 0.3V to VDD+0.3V
Operating ambient temperature .... -40°C to 85°C
Storage ambient temperature .... -40°C to 125°C

21.2 系统操作特性

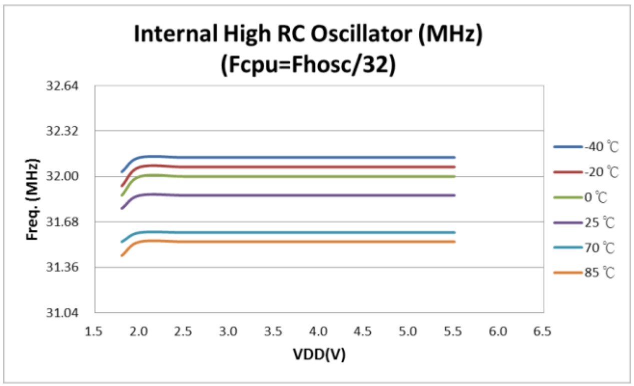
Parameter		Test Condition	Min	TYP	MAX	UNIT
VDD	Operating voltage	fcpu = 1MHz	1.8		5.5	V
V_{DR}	RAM data retention Voltage		0.55			V
V_{POR}	VDD rising rate*		0.05			V/ms
I_{DD1}	Normal mode supply current	VDD = 3V, fcpu = 0.25MHz		2.20		mA
		VDD = 5V, fcpu = 0.25MHz		2.25		mA
		VDD = 3V, fcpu = 1MHz		2.25		mA
		VDD = 5V, fcpu = 1MHz		2.30		mA
		VDD = 3V, fcpu = 4MHz		2.70		mA
		VDD = 5V, fcpu = 4MHz		2.75		mA
		VDD = 3V, fcpu = 8MHz		3.20		mA
		VDD = 5V, fcpu = 8MHz		3.25		mA
I_{DD2}	STOP mode supply current	VDD = 3V		1.4	3.8	μA
		VDD = 5V		1.6	4.0	μA
I_{DD3}	STOP mode supply current (ILRC enable STWK=1) *	VDD = 5V		2.2	4.6	μA
I_{DD4}	IDLE mode supply current (fcpu = 1MHz)	VDD = 3V, 32MHz IHRC		0.63		mA
		VDD = 5V, 32MHz IHRC		0.65		mA
F_{IHRC}	Internal high clock generator	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}	Internal low clock generator	VDD = 5V, 25°C	12	16	24	kHz

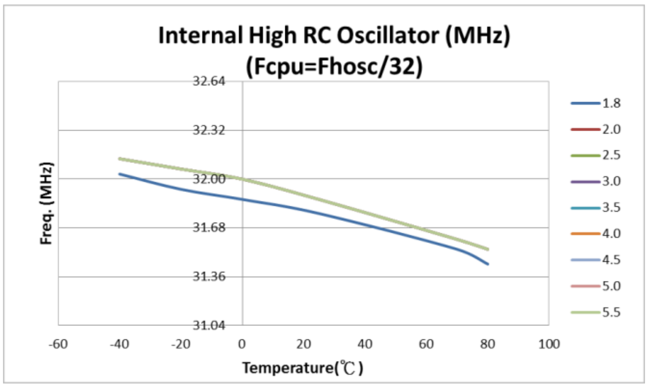
V_{LVD18}	LVD18 detect voltage	25°C	1.6	1.7	1.8	V
		-40°C to 85°C	1.5	1.7	1.9	V
V_{ESD_HBM}	ESD human body mode		4000			V
V_{ESD_MM}	ESD machine mode		400			V

* 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 – 温度图

IHRC 图用于设计指导，无测试与保证。在一些图表中，所提供的数据是在指定的操作范围之外的。它只是信息，而且设备只能在指定范围内正常运行。

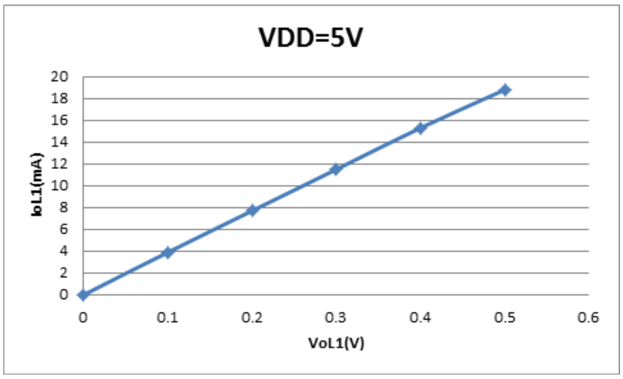


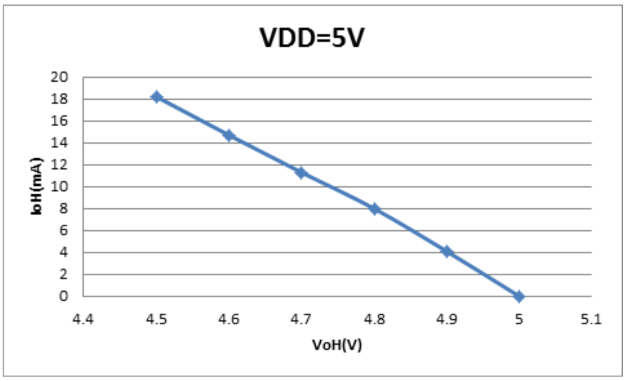


21.3 GPIO 特性

Parameter		Test Condition	Min	TYP	MAX	UNIT
V_{IL}	Low-level input voltage		VSS		0.3VDD	V
V_{IH}	High-level input voltage		0.7VDD		VDD	V
I_{LEKG}	I/O port input leakage current	V_{IN} = VDD			2	μA
R_{UP}	Pull-up resister	VDD = 3V	100	200	300	kΩ
		VDD = 5V	50	100	150	kΩ
I_{OH}	I/O output source current	VDD = 5V, V_O = VDD-0.5V	12	16		mA
I_{OL1}	I/O sink current (P07, P1, P2)	VDD = 5V, V_O = VSS+0.5V	15	35		mA
I_{OL2}	I/O sink current (P00-P06)	VDD = 5V, V_O = VSS+1.5V	80	100		mA

* 环境温度为 25°C。





21.4 ADC 特性

Parameter		Test Condition	Min	TYP	MAX	UNIT
V_{ADC}	Operating voltage		2.0		5.5	V
V_{AIN}	AIN channels input voltage	VDD = 5V	0		V_{REFH}	V
V_{REFH}	AVREFH pin input voltage	VDD = 5V	2		VDD	V
V_{IREF}	Internal VDD reference voltage	VDD = 5V		VDD		V
	Internal 4V reference voltage	VDD = 5V	-1%	4	+1%	V
	Internal 3V reference voltage	VDD = 5V	-1%	3	+1%	V
	Internal 2V reference voltage	VDD = 5V	-0.5%	2	+0.5%	V
I_{AD}	ADC current consumption	VDD = 3V		0.65		mA
		VDD = 5V		0.90		mA
f_{ADCLK}	ADC clock	VDD = 5V			32	MHz
f_{ADSMP}	ADC sampling rate	VDD = 5V			500	kHz
t_{ADEN}	ADC function enable period	VDD = 5V	100			μs
DNL	Differential nonlinearity*	f_{ADSMP} = 62.5kHz		±2		LSB
		f_{ADSMP} = 250kHz		±2		LSB
		f_{ADSMP} = 500kHz		±5		LSB
INL	Integral Nonlinearity*	f_{ADSMP} = 62.5kHz		±2		LSB
		f_{ADSMP} = 250kHz		±2		LSB
		f_{ADSMP} = 500kHz		±5		LSB
NMC	No missing code*	f_{ADSMP} = 62.5kHz	10	11	12	Bit
		f_{ADSMP} = 250kHz		11		Bit
		f_{ADSMP} = 500kHz		9		Bit
V_{OFFSET}	Input offset voltage	Non-trimmed	-10	0	10	mV

* 带*号的参数：VDD = 5V, VREFH = 2.4V, 25°C.

21.5 Flash 存储器特性

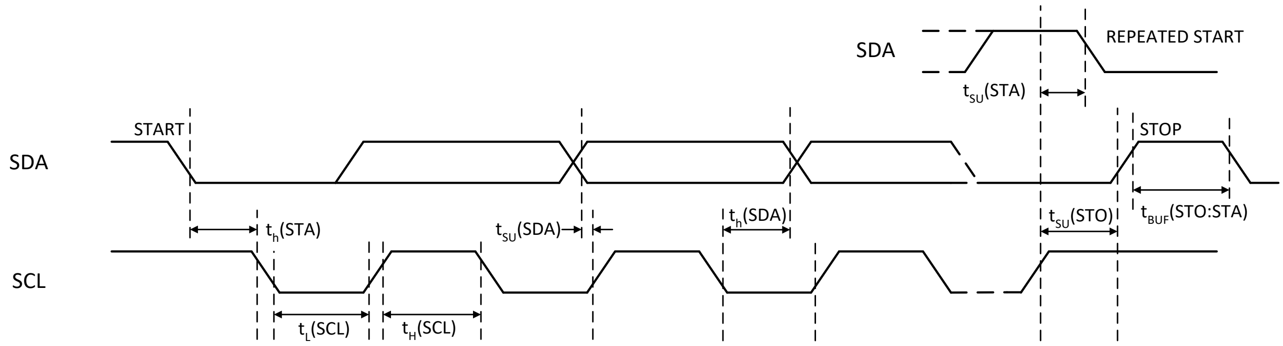
Parameter		Test Condition	Min	TYP	MAX	UNIT
V_{dd}	Supply voltage		1.8		5.5	V
T_{en}	Endurance time	25°C		*100K		cycle
I_{wrt}	Write current	25°C		3	4	mA
T_{wrt}	Write time	Write 1 page=32 bytes, 25°C		6	8	ms

* 带*号的参数还未经过验证的设计参考值。

21.6 I2C 接口特性

Parameter		Typical Formula	UNIT
t_H(SCL)	SCL clock high time	2* fsck	μs
t_L(SCL)	SCL clock low time	2* fsck	μs
t_{SU}(SDA)	SDA setup time	2* fsck -7*fcpu	μs
t_h(SDA)	SDA data hold time	7*fcpu	μs
t_h(STA)	START condition hold time	4* fsck	μs
t_{SU}(STA)	Repeated START condition setup time	2* fsck	μs
t_{SU}(STO)	STOP condition setup time	4* fsck +8*fcpu	μs
T_{BUF}(STO:STA)	STOP to START condition time	4/3* fsck +10*fcpu	μs

*标准模式 I2C Fsck = 100KHz，快速模式 I2C Fsck = 400KHz 。



22 指令集

本章对 SN8F5702A 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、数据传送运算、布尔操作和程序分支指令，这些指令全部都与标准 8051 兼容。

符号说明

Symbol	Description
Rn	工作寄存器 R0~R7
direct	128 个内部 RAM 地址之一或其它特殊功能寄存器
@Ri	寄存器 R0 或 R1 寻址的间接内部或外部 RAM 地址
#data	8 位常量（立即操作数）
#data16	16 位常量（立即操作数）
bit	内部 RAM 中的 128 个软件标志之一，或其它位-可寻址特殊功能寄存器的标志
addr16	LCALL 或 LJMP 的目标地址，可以是程序存储器地址空间的 64K 字节页面内的任一地址
addr11	ACALL 或 AJMP 的目标地址，程序存储器地址的同一个 2K 字节页面内，作为下一指令的第一个字节
rel	SJMP 和所有条件跳移，包括一个 8 位偏移字节。其范围是+127/-128 字节，相对于下一指令的第一个字节
A	累加器

运算指令

Mnemonic	Description
ADD A, Rn	加寄存器到累加器
ADD A, direct	加直接寻址数据到累加器
ADD A, @Ri	加间接寻址数据到累加器
ADD A, #data	加立即数据到累加器
ADDC A, Rn	加寄存器到累加器，带进位
ADDC A, direct	加直接寻址数据到累加器，带进位
ADDC A, @Ri	加间接寻址数据到累加器，带进位
ADDC A, #data	加立即数据到累加器，带进位
SUBB A, Rn	从累加器减去寄存器，带借位
SUBB A, direct	从累加器减去直接寻址数据，带借位
SUBB A, @Ri	从累加器减去间接寻址数据，带借位
SUBB A, #data	从累加器减去立即数据，带借位
INC A	累加器递增
INC Rn	寄存器递增
INC direct	直接寻址地址递增
INC @Ri	间接寻址地址递增
INC DPTR	数据指针递增
DEC A	累加器递减
DEC Rn	寄存器递减
DEC direct	直接寻址地址递减
DEC @Ri	间接寻址地址递减
MUL AB	A 乘以 B
DIV	A 除以 B
DA A	十进制调整累加器

逻辑运算指令

Mnemonic	Description
ANL A, Rn	寄存器与累加器
ANL A, direct	直接寻址数据与累加器
ANL A, @Ri	间接寻址数据与累加器
ANL A, #data	立即数据与累加器
ANL direct, A	累加器与直接寻址地址
ANL direct, #data	累加器与间接寻址地址
ORL A, Rn	寄存器或累加器
ORL A, direct	直接寻址数据或累加器
ORL A, @Ri	间接寻址数据或累加器
ORL A, #data	立即数据或累加器
ORL direct, A	累加器或直接寻址地址
ORL direct, #data	立即数据或直接寻址地址
XRL A, Rn	寄存器异或累加器
XRL A, direct	直接寻址数据异或累加器
XRL A, @Ri	间接寻址数据异或累加器
XRL A, #data	立即数据异或累加器
XRL direct, A	累加器异或间接寻址地址
XRL direct, #data	累加器异或直接寻址地址
CLR A	清零累加器
CPL A	累加器取反
RL A	循环左移累加器
RLC A	带进位符循环左移累加器
RR A	循环右移累加器
RRC A	带进位符循环右移累加器
SWAP A	累加器高低四位互换

数据传输指令

Mnemonic	Description
MOV A, Rn	传送寄存器至累加器
MOV A, direct	传送直接寻址数据至累加器
MOV A, @Ri	传送间接寻址数据至累加器
MOV A, #data	传送立即数据至累加器
MOV Rn, A	传送累加器至寄存器
MOV Rn, direct	传送直接寻址数据至寄存器
MOV Rn, #data	传送立即数据至寄存器
MOV direct, A	传送累加器至直接寻址数据
MOV direct, Rn	传送寄存器至直接寻址数据
MOV direct1, direct2	传送直接寻址数据至直接寻址地址
MOV direct, @Ri	传送间接寻址数据至直接寻址地址
MOV direct, #data	传送立即数据至直接寻址地址
MOV @Ri, A	传送累加器至间接寻址地址
MOV @Ri, direct	传送直接寻址数据至间接寻址地址
MOV @Ri, #data	传送立即数据至间接寻址地址
MOV DPTR, #data16	加载带 16 位立即数据的数据指针
MOVC A, @A+DPTR	加载带相对于 DPTR 的代码字节的累加器
MOVC A, @A+PC	加载带相对于 PC 的代码字节的累加器
MOVX A, @Ri	传送外部 RAM（8 位地址）至累加器
MOVX A, @DPTR	传送外部 RAM（16 位地址）至累加器
MOVX @Ri, A	传送累加器至外部 RAM（8 位地址）
MOVX @DPTR, A	传送累加器至外部 RAM（16 位地址）
PUSH direct	直接寻址数据入栈
POP direct	直接寻址地址出栈
XCH A, Rn	寄存器与累加器交换
XCH A, direct	直接寻址地址与累加器交换
XCH A, @Ri	间接 RAM 与累加器交换
XCHD A, @Ri	间接地址操作数低半字节与累加器低半字节内容互换

布尔运算指令

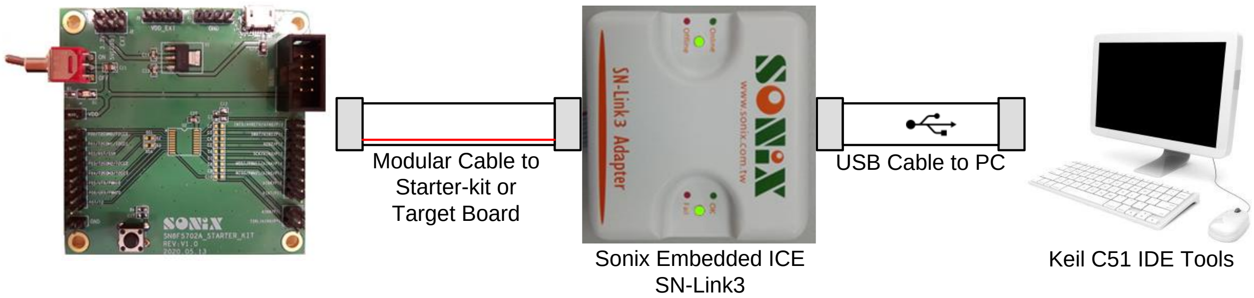
Mnemonic	Description
CLR C	清除进位标志
CLR bit	清除直接寻址位
SETB C	设置进位标志
SETB bit	设置直接寻址位
CPL C	标志位取反
CPL bit	直接寻址位取反
ANL C, bit	直接寻址位与进位标志
ANL C, /bit	直接寻址位取反与进位
ORL C, bit	直接寻址位或进位标志
ORL C, /bit	直接寻址位取反或进位
MOV C, bit	传送直接寻址位至进位标志
MOV bit, C	传送进位标志至直接寻址位

程序跳转指令

Mnemonic	Description
ACALL addr11	绝对调用 addr11
LCALL addr16	长调用 addr16
RET	返回下一条指令处
RETI	返回自中断至下一条指令处
AJMP addr11	短跳转 addr11
LJMP addr16	长跳转 addr16
SJMP rel	相对跳转（相对地址）
JMP @A+DPTR	散转移
JZ rel	判断 0 转移
JNZ rel	判断非 0 转移
JC rel	判断进位标志为 1 时，跳转
JNC rel	判断进位标志为 0 时，跳转
JB bit, rel	判断直接寻址位为 1 时，跳转
JNB bit, rel	判断直接寻址位为 0 时，跳转
JBC bit, rel	判断直接寻址位为 1 时，跳转并定位清零
CJNE A, direct, rel	比较直接寻址数据和累加器，如果不相等，则跳转
CJNE A, #data, rel	比较立即数据和累加器，如果不相等，则跳转
CJNE Rn, #data, rel	比较直接寻址数据和寄存器，如果不相等，则跳转
CJNE @Ri, #data, rel	比较立即数据和间接寻址数据，如果不相等，则跳转
DJNZ Rn, rel	寄存器减一，不为 0 跳转
DJNZ direct, rel	直接寻址地址减一，不为 0 跳转
NOP	一个周期的空操作

23 开发环境

SONIX 提供嵌入式 ICE 仿真器系统，用于 SN8F5702A 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、SN8F5702A 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 SN8F5702A 中，可提供真正的开发环境。



23.1 最低要求

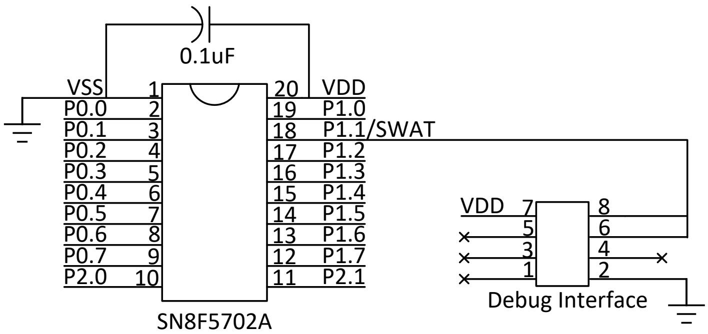
下面几项内容是建立合适的开发环境的基本要求，其兼容性已经经过验证，在后面的版本中可以很好的执行。SN-Link 的相关信息可以 SONiX 网址 www.sonix.com.tw 下载，Keil C51 可从 www.keil.com/c51 下载。

- **SN-Link3 Adapter** ：更新到 V1.02。
- **SN-Link Driver for Keil C51**：V1.00.317。
- **Keil C51**：V 9.50a 和 9.54a 或更高。

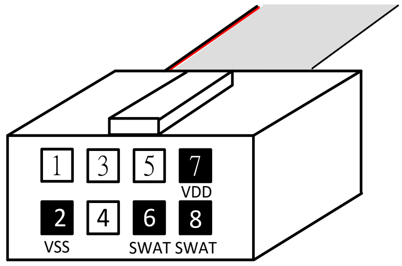
23.2 硬件调试界面

下面的电路图显示了如何正确地连接单片机到 SWAT 引脚和 SN-Link Adapter3。

开始调试之前，必须切断单片机的电源（VDD）。把 SWAT 引脚连接到 SN-Link 的第 6 脚和第 7 脚，SN-Link 的第 7 脚和第 2 脚分别连接 VDD 和 VSS。打开单片机，就会自动开始握手操作，SN-Link 的红色指示灯（Run）显示连接成功。（详细内容请参考 SN8F5000 调试工具使用手册。）



示例电路

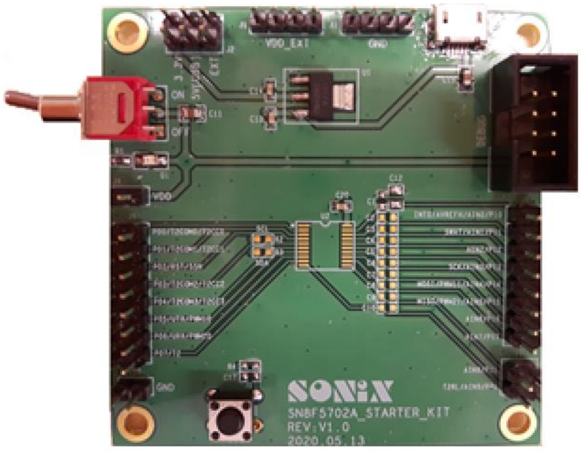


SN-Link 接口

23.3 开发工具

SN-Link3 适配器

Starter-Kit 支持 SN8F5702A, 570210A, 570211A, 570212A, 570213A, 570200A, 570202A



MP5 烧录器

24 SN8F5702A Starter-Kit

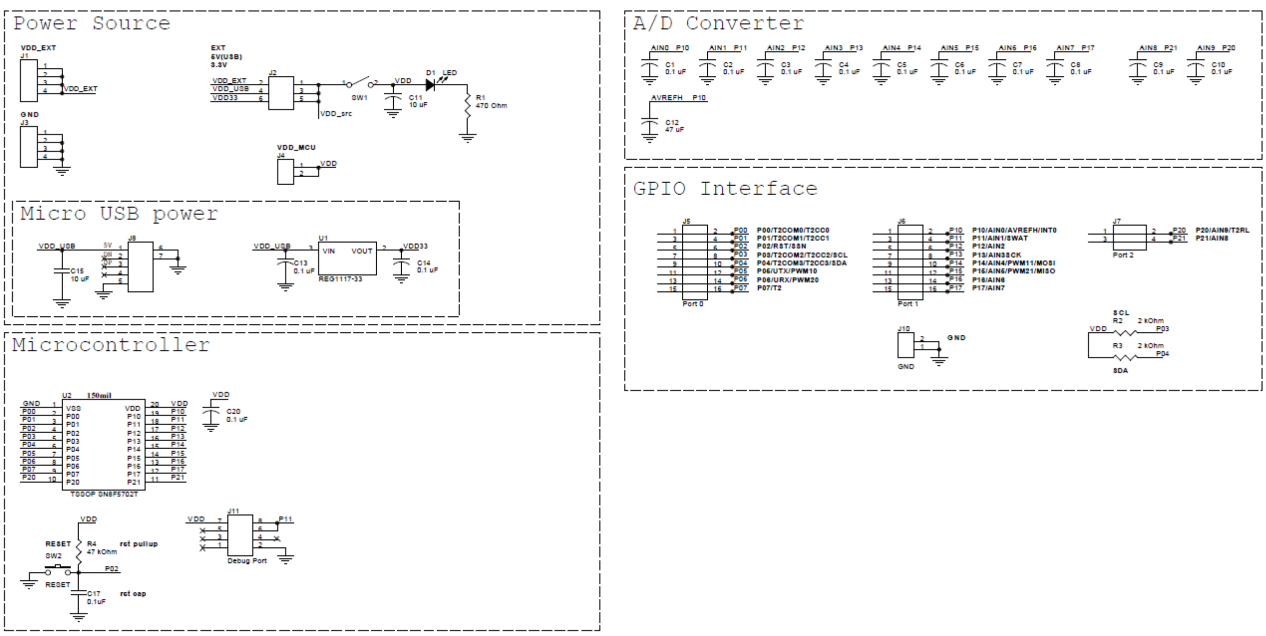
SN8F5000 Starter-Kit 提供一个简易的开发平台。它包含 SN8F5000 系列 IC 和输入信号或用户应用程序驱动设备的 IO 接口。当目标板没有准备好时，它是一个简单开发平台用于应用程序开发。Starer-Kit 可以被目标板所取代，因为 SN8F5000 系列集成了嵌入式的电路调试器电路。

24.1 配置电路

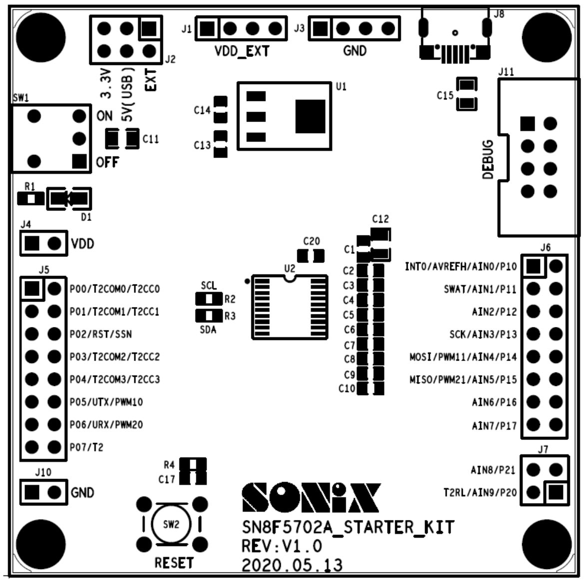
在开始利用 Start-Kit 开发前必须完成以下配置：

1. 确认电路板元件是否完整；
2. starkit 电路的电源是从外部电源或 Micro 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 Micro USB；
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. 调试端口可以连接 SN-LINK 适配器进行仿真或下载代码；
7. 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
C1,C2,C3,C4,C5,C6,C7,C8, C9,C10	10 通道 ADC 电容
C12	AVREFH 电容
D1	MCU LED
J8	Micro USB 供电
J1, J3	外部电源
SW2	外部复位触发源
J2	VDD 电源选择为外部电源， USB 或 3.3V
J11	调试接口
J5, J6, J7	I/O 口
J4	MCU VDD
J10	MCU GND
R4, C17	外部复位上拉电阻和电容
R2, R3	I2C 上拉电阻
SW1	目标 (VDD) 开关
U2	SN8F5702AT 实际芯片
U1	REG1117-33
C20	SN8F5702AT 电容

25 ROM 烧录引脚

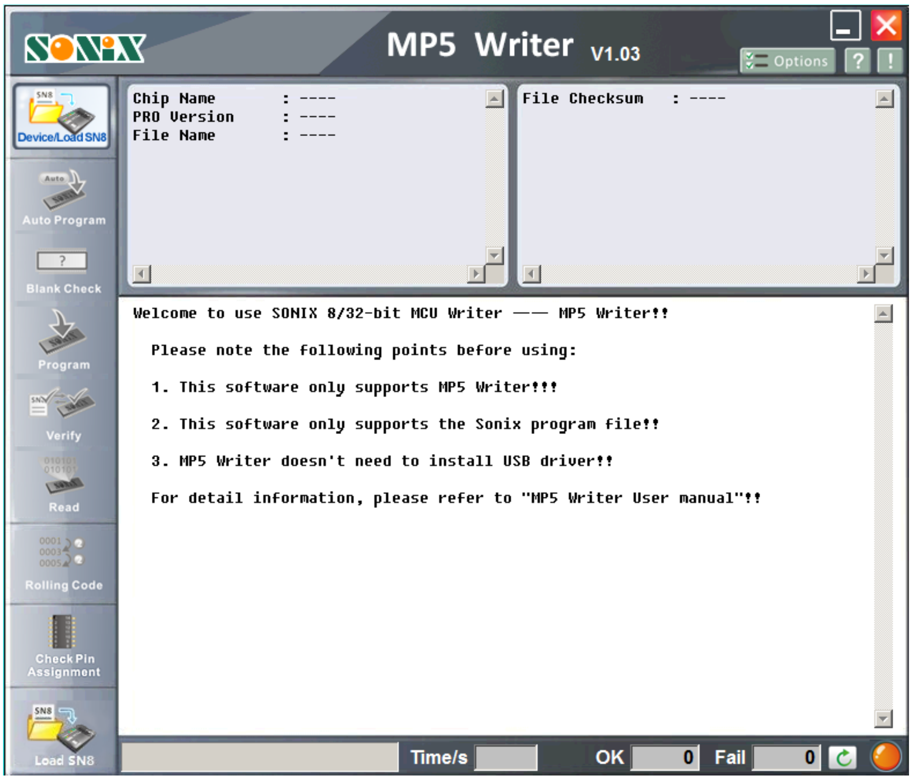
SN-LINK 和 MP5 Writer 支持 SN8F5702A 系列 Flash ROM 的擦除/烧录/校正。

- **SN-LINK：调试界面。**
- **MP5 Writer：SN8F5702A 系列。**

25.1 MP5 硬件连接

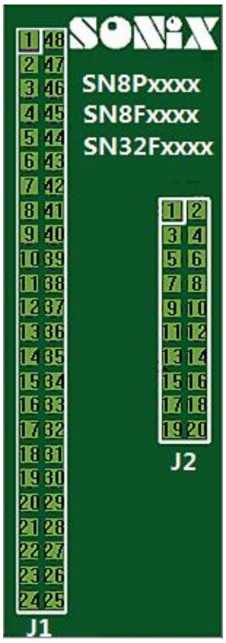
不同封装 MCU 烧录时的硬件接法如下，SOP，TSSOP 和 QFN 说明。

MP5 软件操作界面如下：



25.2 MP5 烧录转接板引脚配置

MP5 烧录转接板：



25.3 MP5 Writer 烧录引脚配置

烧录器连接器		MCU 引脚编号	SN8F5702AS/T		SN8F5702AJ		SN8F570200AA		SN8F570202AS	
J2 引脚编号	J2 引脚名称		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	20	34	18	32	1	20	1	21
2	GND	VSS	1	15	19	33	2	21	8	28
7, 9	SWAT	P1.1	18	32	16	30	9	28	5	25
20	PDB	P0.5	7	21	5	19	5	24	3	23

烧录器连接器		MCU 引脚编号	SN8F570210AS		SN8F570211AS		SN8F570212AS/T		SN8F570213AS	
J2 引脚编号	J2 引脚名称		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	14	31	14	31	16	32	1	18
2	GND	VSS	1	18	1	18	1	17	14	31
7, 9	SWAT	P1.1	12	29	12	29	14	30	12	29
20	PDB	P0.5	5	22	6	23	6	22	5	22

25.4 SN-Link ISP 烧录

SN-Link ISP 烧录硬件和软件如下：

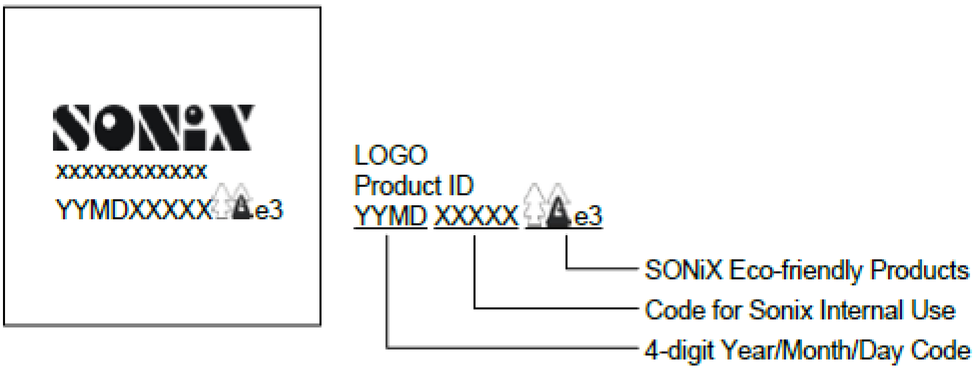
25.5 SN-Link ISP 烧录引脚图

SN-Link 连接器 引脚编号	SN-Link 连接器 引脚名称	MCU 引脚编号	SN8F5702AP/S/T 引脚编号	SN8F5702AJ 引脚编号	SN8F570200AA 引脚编号	SN8F570202AS 引脚编号
7	VDD	VDD	20	18	16	1
2	GND	VSS	1	19	1	14
6, 8	SWAT	P1.1	18	16	14	12

SN-Link 连接器 引脚编号	SN-Link 连接器 引脚名称	MCU 引脚编号	SN8F570210AS 引脚编号	SN8F570211AS 引脚编号	SN8F570212AS/T 引脚编号	SN8F570213AS 引脚编号
7	VDD	VDD	20	18	16	1
2	GND	VSS	1	19	1	14
6, 8	SWAT	P1.1	18	16	14	12

26 订购信息

SONiX 的单片机的表面印有三栏信息：商标，单片机全称和日期码。



26.1 命名规则

单片机全称	封装类型
SN8F5702AW	Wafer
SN8F5702AH	Dice
SN8F5702APG	PDIP，20 pins，绿色封装
SN8F5702ASG	SOP，20 pins，绿色封装
SN8F5702ATG	TSSOP，20 pins，绿色封装
SN8F5702AJG	QFN，20 pins，绿色封装
SN8F570200AAG	MSOP，10 pins，绿色封装
SN8F570202ASG	SOP，8 pins，绿色封装
SN8F570210ASGR	SOP，14 pins，TR 绿色封装
SN8F570211ASG	SOP，14 pins，绿色封装
SN8F570212ASG	SOP，16 pins，绿色封装
SN8F570212ATG	TSSOP，16 pins，绿色封装
SN8F570213ASG	SOP，14 pins，绿色封装

* *注：* ***TR*** *包装（卷带&编带包装）在封装名称后加“**R**”字符。*

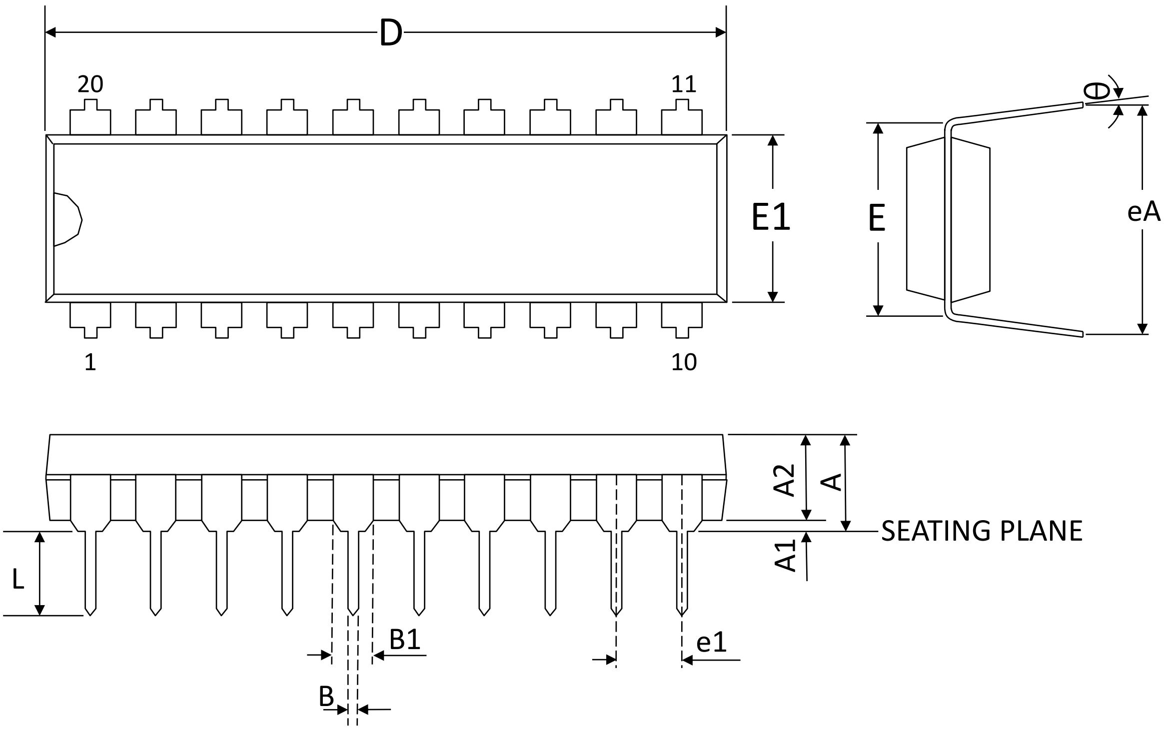
26.2 日期码

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

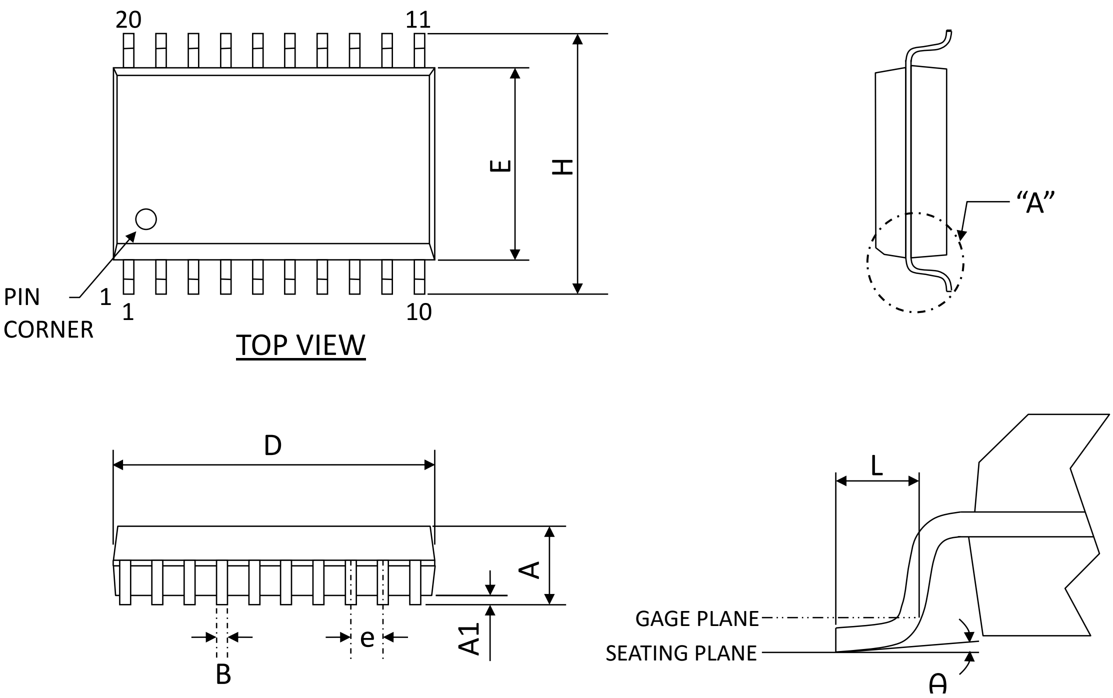
27.1 DIP20



SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.		MIN.	NOM.	
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

27.2 SOP20



TOP VIEW

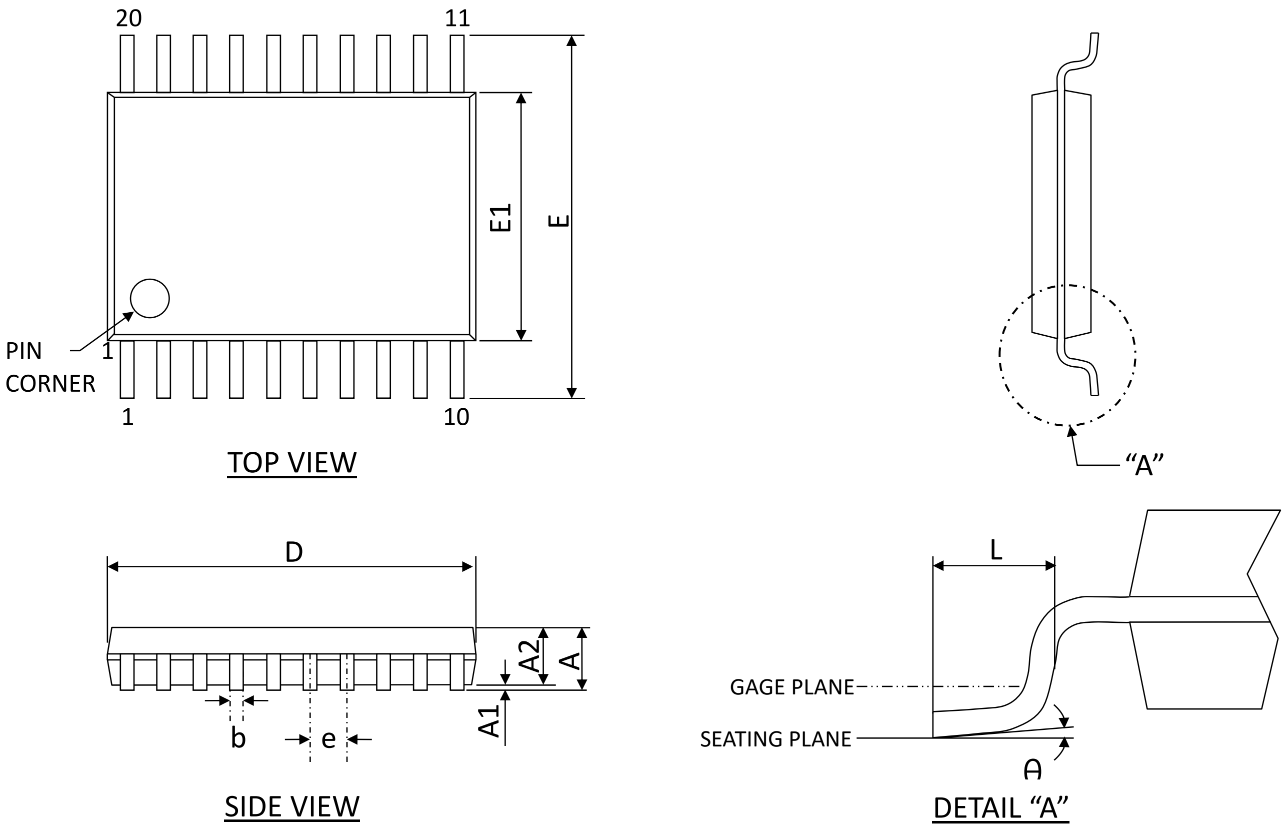
SIDE VIEW

DETAIL "A"

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

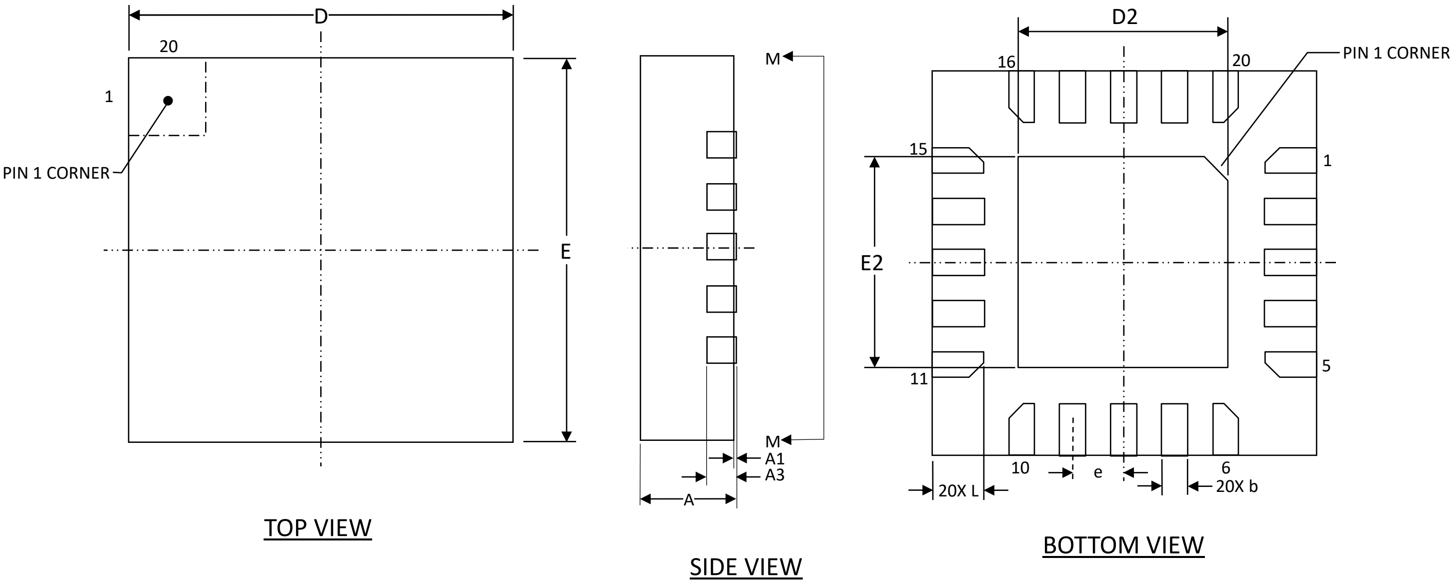
27.3 TSSOP20



SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.20	--	--	0.047
A1	0.05	--	0.15	0.002	--	0.006
A2	0.80	--	1.05	0.031	--	0.041
b	0.19	--	0.30	0.007	--	0.012
D	6.40	6.50	6.60	0.252	0.256	0.260
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.

27.4 QFN20 3X3

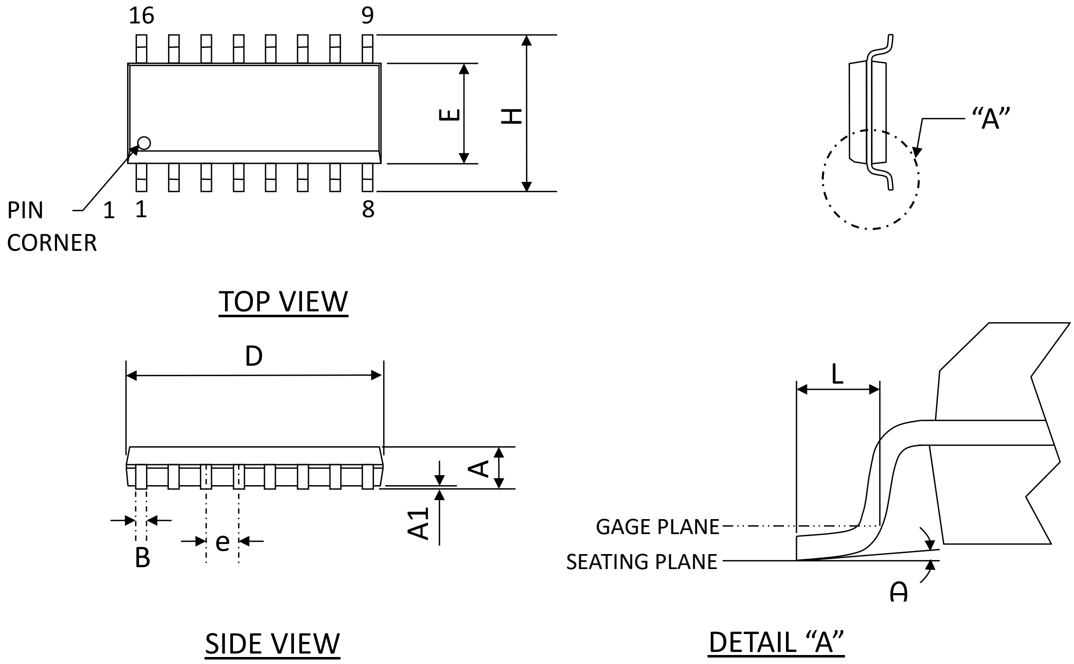


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.80	0.90	0.028	0.031	0.035
A1	0.00	0.02	0.05	0.000	0.001	0.002
A3	0.203 REF			0.008 REF		
b	0.15	0.20	0.25	0.006	0.008	0.010
D	3.00 BSC			0.118 BSC		
E	3.00 BSC			0.118 BSC		
e	0.40 BSC			0.016 BSC		
D2	1.55	1.65	1.75	0.61	0.65	0.69
E2	1.55	1.65	1.75	0.61	0.65	0.69
L	0.30	0.40	0.50	0.012	0.016	0.020

Notes：

1. CONTROLLING DIMENSION：MILLIMETER (mm)

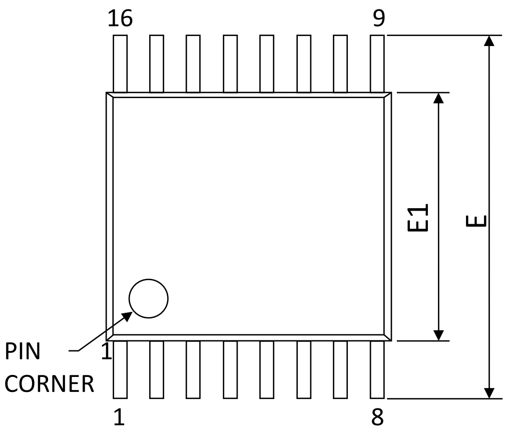
27.5 SOP16

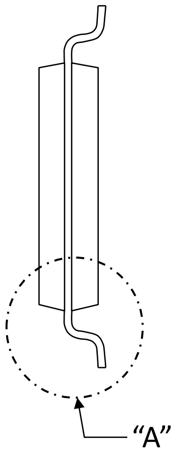


SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.75	--	--	0.069
A1	0.10	--	0.25	0.004	--	0.010
B	0.31	0.41	0.51	0.012	0.016	0.020
D	9.90 BSC			0.389 BSC		
E	3.90 BSC			0.153 BSC		
e	1.27 BSC			0.050 BSC		
H	6.00 BSC			0.236 BSC		
L	0.40	--	1.27	0.016	--	0.050
θ	0°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

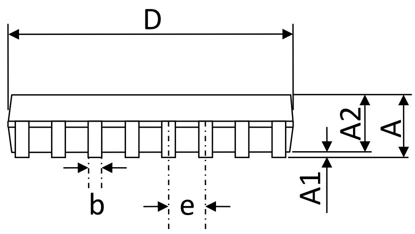
Notes：
1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MS-012 AC

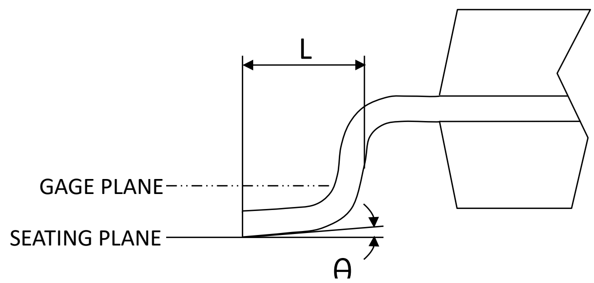
27.6 TSSOP16





TOP VIEW





SIDE VIEW

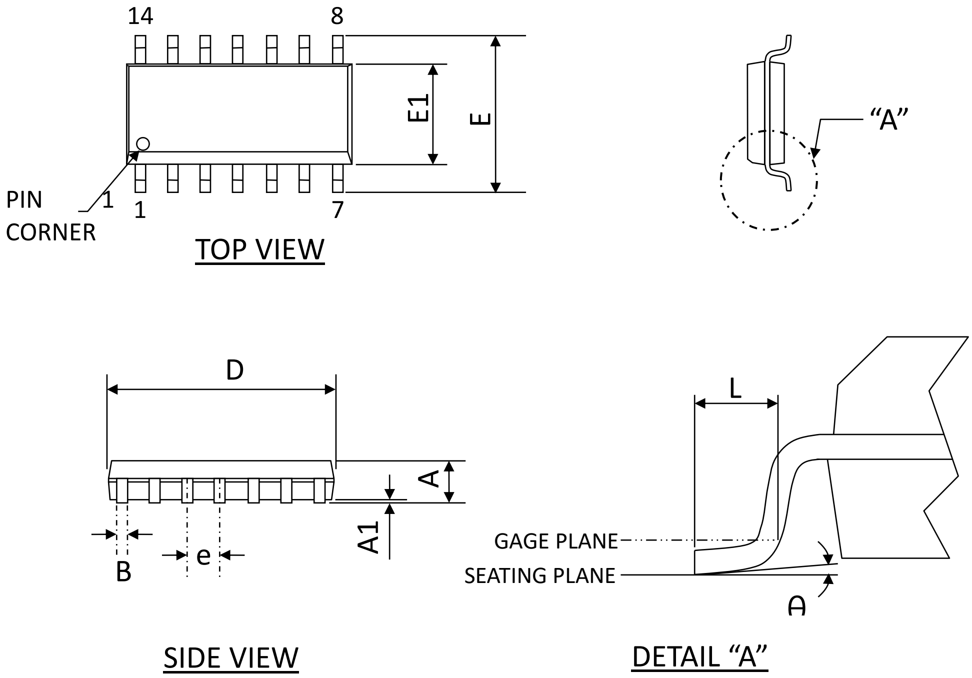
DETAIL "A"

SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.20	--	--	0.047
A1	0.05	--	0.15	0.002	--	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	4.80	5.00	5.20	0.189	0.196	0.205
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.

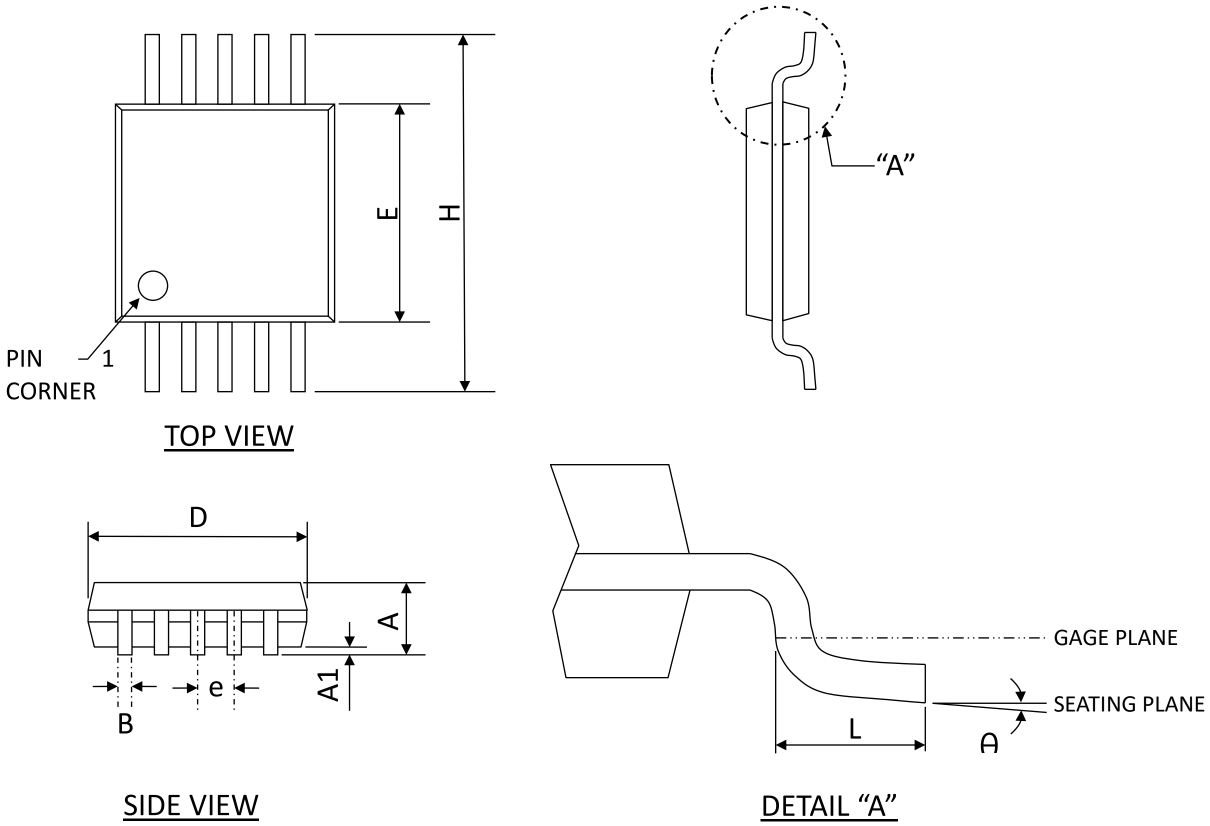
27.7 SOP14



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

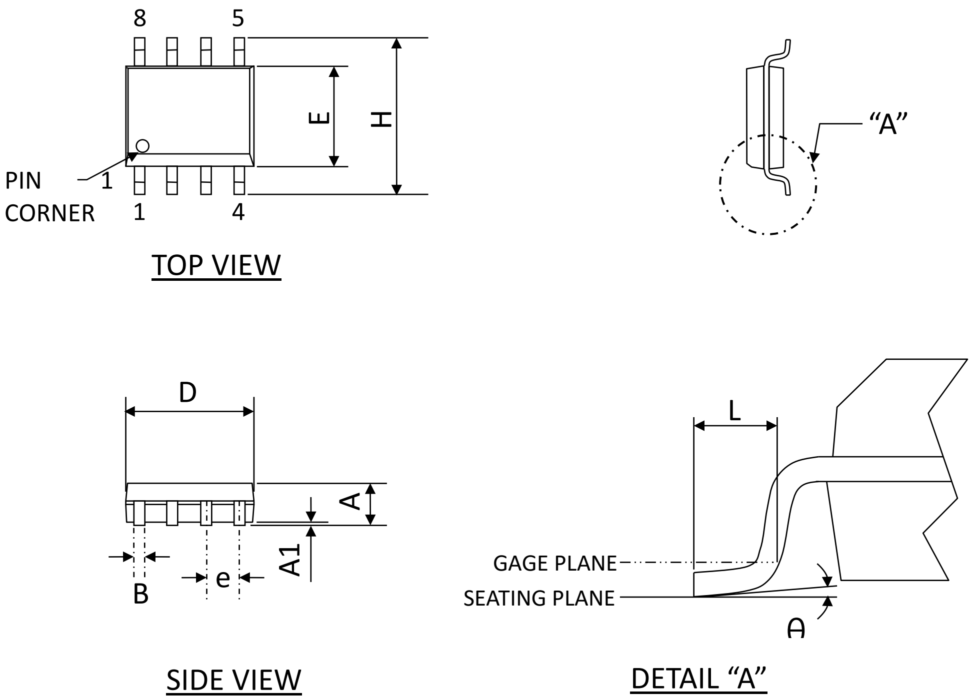
27.8 MSOP10



SYMBOLS	Dimension in mm			Dimension in inch		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	--	--	1.10	--	--	0.043
A1	0.00	--	0.15	0.000	--	0.006
B	0.17	--	0.27	0.007	--	0.011
D	3.00 BSC			0.118 BSC		
E	3.00 BSC			0.118 BSC		
e	0.50 BSC			0.020 BSC		
H	4.9 BSC			0.193 BSC		
L	0.40	0.60	0.80	0.016	0.024	0.031
θ	0°	4°	8°	0°	4°	8°

Notes：
1. CONTROLLING DIMENSION：mm
2. JEDEC OUTLINE : MO-187 BA

27.9 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 Packaging 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。

总公司：
地址：台湾新竹县竹北市台元街 36 号 10 楼之一
电话：886-3-5600-888
传真：886-3-5600-889
台北办事处：
地址：台北市松德路 171 号 15 楼之 2
电话：886-2-2759 1980
传真：886-2-2759 8180
香港办事处：
地址：香港新界沙田炭禾盛街 11 号，中建电讯大厦 26 楼 03 室
电话：852-2723 8086
传真：852-2723 9179
松翰科技（深圳）有限公司
地址：深圳市宝安区新安二路中粮紫云大厦 26F
电话：86-755-2671 9666
传真：86-755-2671 9786
技术支持：
Sn8fae@SONiX.com.tw