盛群知识产权政策

专利权

盛群半导体公司在全球各地区已核准和申请中之专利权至少有 160 件以上，享有绝对之合法权益。与盛群公司 MCU 或其它产品有关的专利权并未被同意授权使用，任何经由不当手段侵害盛群公司专利权之公司、组织或个人，盛群将采取一切可能的法律行动，遏止侵权者不当的侵权行为，并追讨盛群公司因侵权行为所受之损失、或侵权者所得之不法利益。

商标权

盛群之名称和标识、Holtek 标识、HT-IDE、HT-ICE、Marvel Speech、 Music Micro、 Adlib Micro、 Magic Voice、 Green Dialer、 PagerPro、 Q-Voice、 Turbo Voice、 EasyVoice 和 HandyWriter 都是盛群半导体公司在台湾地区和其它国家的注册商标。

著作权

规格书中所出现的信息在出版当时相信是正确的，然而盛群对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明，盛群不保证或不表示这些应用没有更深入的修改就能适用，也不推荐它的产品使用在会由于故障或其它原因可能会对人身造成危害的地方。盛群产品不授权使用于救生、维生器件或系统中做为关键器件。盛群拥有不事先通知而修改产品的权利，对于最新的信息，请参考我们的网址 http://www.holtek.com.tw; http://www.holtek.com.cn

技术相关信息

技术相关信息

- 工具信息
- FAQs
- 应用范例
- 一HA0075S MCU 重置电路及振荡电路应用

特性

- 工作电压：
  f_{SYS}=4MHz：2.2V~5.5V
  f_{SYS}=8MHz：3.3V~5.5V
  f_{SYS}=16MHz：4.5V~5.5V
- OTP 程序空间 2K×15
- 数据空间 128×8
- 16 个双向输入/输出口
- 1 个与输入/输出口共用引脚的外部中断输入
- 8 位可编程定时/计数器，具有溢出中断和 7 级预分频器
- 16 位可编程定时/计数器，具有溢出中断
- 外部晶体和 RC 振荡电路
- 看门狗定时器
- 具有 PFD 功能，可用于发声
- HALT 和唤醒功能可降低功耗
- 在 V_{DD}=5V，系统频率为 16MHz 时，指令周期为 0.25μs
- 4 层硬件堆栈
- 位操作指令
- 查表指令
- 63 条指令
- 指令执行时间为 1 或 2 个指令周期
- 低电压复位
- 3 个集成运算放大器，其中一个带增益控制功能
- 一个比较器，相应中断功能开放。
- HT46RS03P 中内置 5V LDO
- HT46RS03E 及 HT46RS03P 带 128×8 EEPROM
- 多种封装

概述

此系列 MCU 是 8 位高性能精简指令集单片机，专门为需要运算放大器应用的产品而设计，比较常见的是应用在超声波感测器方案中。

低功耗、I/O 使用灵活、可编程分频器、计数器、振荡类型选择、内置比较器及运算放大器、暂停和唤醒功能，使此系列单片机可以广泛地应用在遥控控制系统、汽车倒车系统、各类位计量器等等。

HT46RS03E 及 HT46RS03PE 两款产品中内带 EEPROM 存储器，主要是为了满足一些数据需要保存不被改写的应用，诸如：校正参数值，序列号等。不同的封装类型产品中除了 LDO 及 EPPROM 功能外，其他特性都是一样的。

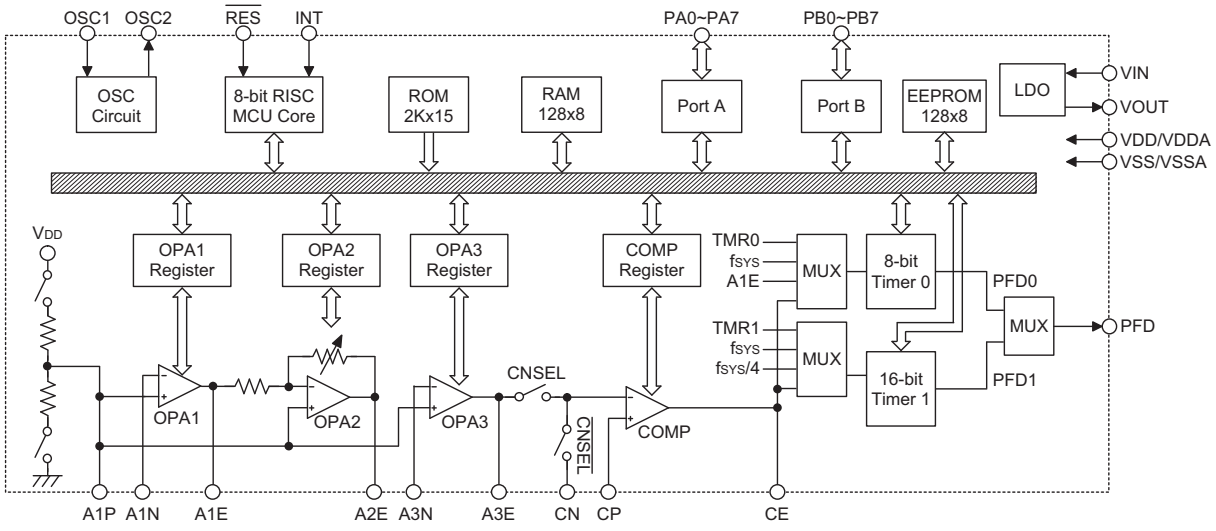
选型表

此系列MCU中大部分的特性都是相同的，各封装一样它们主要的区别在于是LDO及EPPROM功能。下表描述了每款产品的主要特性。

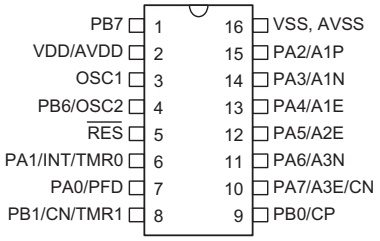
Part No.	VDD	VIN	Program Memory	Data Memory		I/O	Timer		interrupt		LDO	OPA	CMP	PFD	Stack	Package Types
				SRAM	EEPROM		8-bit	16-bit	外部	内部						
HT46RS03	2.2V~5.5V	—	2K×15	128×8	—	16	1	1	1	3	—	3	1	√	4	16NSOP 16/20DIP 16/20SSOP
HT46RS03E					128×8											20SOP
HT46RS03P	5.0V	5.5V~24V	2K×15	128×8	—	15	1	1	1	3	√	3	1	√	4	16NSOP 16/20DIP 16/20SSOP
HT46RS03PE					128×8	16										20/24SOP

方框图

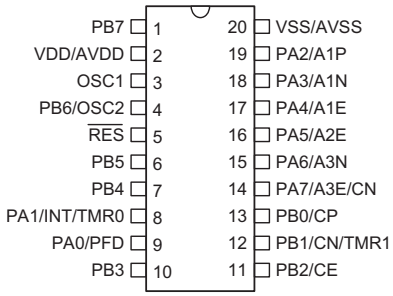
下面这张图主要列出了一些主要功能模块。



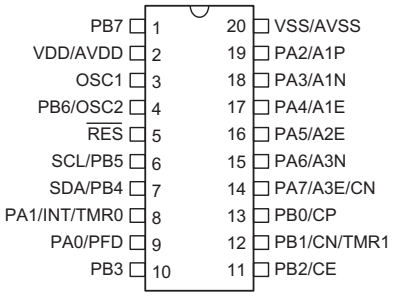
引脚图



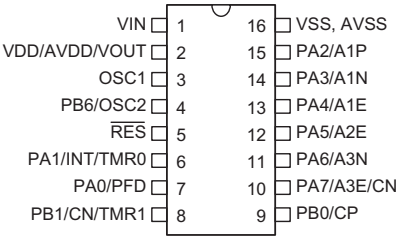
HT46RS03
16 DIP-A/NSOP-A/SSOP-A



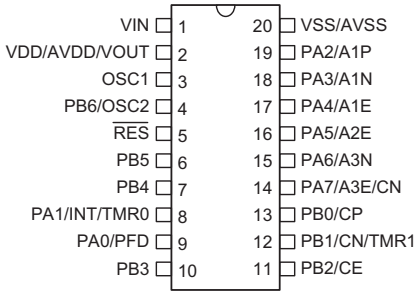
HT46RS03
20 DIP-A/SSOP-A



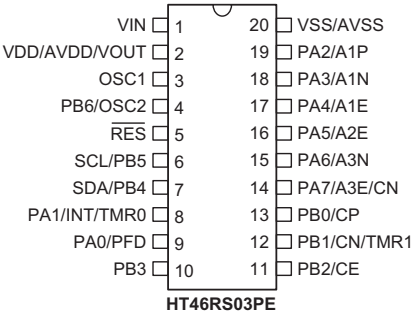
HT46RS03E
20 SOP-A



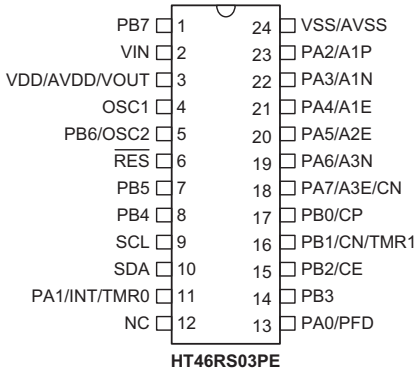
HT46RS03P
16 DIP-B/NSOP-A/SSOP-B



HT46RS03P
20 DIP-B/SSOP-B



HT46RS03PE
20 SOP-A



HT46RS03PE
24 SOP-A

引脚说明

引脚名称	输入/输出	掩膜选项	功能说明
PA0/PFD PA1/INT/TMR0 PA2/A1P PA3/A1N PA4/A1E PA5/A2E PA6/A3N PA7/A3E/CN	输入/输出	上拉电阻 唤醒功能 PA0 或 PFD	8 位双向输入/输出口。每一位可由掩膜选项设置为唤醒输入。可由软件设置为 CMOS 输出、带或不带上拉电阻（由上拉电阻选项决定：位选择）的施密特触发输入。 PA0 与 PFD 共用引脚。 PA1 与 INT/TMR0 共用引脚。 PA2 ，PA3，PA4，PA5，PA6 及 PA7 与 A1P，A1N，A1E，A2E，A3N，A3E 及 CN 输入脚共用引脚。可由软件设定这些管脚作为输入/输出或模拟信号 OPA/比较器功能。选择模拟信号功能后，输入/输出功能和上拉电阻自动失效。
PB0/CP PB1/CN/TMR1 PB2/CE PB3~PB5 PB7	输入/输出	上拉电阻	7 位双向输入/输出口。可由软件设置为 CMOS 输出、带或不带上拉电阻（由上拉电阻选项决定：位选择）的施密特触发输入。 PB0，PB1，PB2 分别与比较器的 CP，CN 及 CE 共用引脚。PB1 同时还与 TMR1 共用引脚。 可以软件设置这些管脚为输入/输出或模拟信号比较器功能。一旦选择模拟信号功能，输入/输出功能和上拉电阻自动失效。
OSC1 PB6/OSC2	输入 输入/输出	RC 或晶振	OSC1 和 OSC2 被连接到外部 RC 振荡或者外部晶振(由掩膜选项选择）提供系统时钟。如果选择了 RC 系统时钟选项，通过掩膜选项可以将 PB6/OSC2 脚设置为带/不带上拉电阻的输入/输出口或用于输出 1/4 系统时钟。在外部晶振模式，上拉电阻功能自动失效。
$\overline{RES}$	输入	—	斯密特触发复位输入，低电平有效。
VIN	—	—	5V LDO 电压输入口（最高到 24V）
VOUT	—	—	3.3V 或 5V 输出。此引脚与 VDD 及 AVDD 连在一起
VSS	—	—	负电源，接地。
AVSS	—	—	模拟信号负电源（为 OPAS 及比较器）。AVSS 与 VSS 内部连在一起，为同一根 PIN 脚。
VDD	—	—	正电源。
AVDD	—	—	模拟信号正电源（为 OPAS 及比较器）。AVDD 与 VDD 内部连在一起，为同一根 PIN 脚。

注：20 脚封装的 HT46RS03PE 的 EEPROM 的 SDA 和 SCL 脚在内部已经与 PB4 和 PB5 分别相连。

极限参数

电源供应电压……V_{SS}-0.3V~V_{SS}+6.0V
端口输入电压……V_{SS}-0.3V~V_{DD}+0.3V
端口总灌电流…….. 150mA
总功耗…….. 500mW
储存温度……-50℃~125℃
工作温度……-40℃~85℃
端口总源电流……. -100mA

注：这里只强调额定功率，超过极限参数所规定的范围将对芯片造成损害，无法预期芯片在上述标示范围外的工作状态，而且若长期在标示范围外的条件下工作，可能影响芯片的可靠性。

直流电气特性

Ta=25℃

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
V_{DD}	工作电压	—	f_{SYS}=4MHz	2.2	—	5.5	V
		—	f_{SYS}=8MHz	3.3	—	5.5	V
		—	f_{SYS}=16MHz	4.5	—	5.5	V
I_{DD1}	工作电流（晶体振荡、RC 振荡）	3V	无负载，f_{SYS}=4MHz	—	1	2	mA
		5V		—	2.5	5	mA
I_{DD2}	工作电流（晶体振荡、RC 振荡）	3V	无负载，f_{SYS}=8MHz	—	2	4	mA
		5V		—	4	8	mA
I_{DD3}	工作电流（晶体振荡，RC 振荡）	5V	无负载，f_{SYS}=16MHz	—	6	12	mA
I_{STB1}	静态电流（看门狗打开）	3V	无负载，系统 HALT	—	—	5	μA
		5V		—	—	10	μA
I_{STB2}	静态电流（看门狗关闭）	3V	无负载，系统 HALT	—	—	1	μA
		5V		—	—	2	μA
V_{IL1}	PA、TMR0、TMR1、INT 的低电平输入电压	—	—	0	—	0.3 V_{DD}	V
V_{IH1}	PA、TMR0、TMR1、INT 的高电平输入电压	—	—	0.7V_{DD}	—	V_{DD}	V
V_{IL2}	低电平输入电压（$\overline{RES}$）	—	—	0	—	0.4 V_{DD}	V
V_{IH2}	高电平输入电压（$\overline{RES}$）	—	—	0.9 V_{DD}	—	V_{DD}	V
V_{LVR1}	低电压复位 1	—	掩模选项：2.1v	1.98	2.1	2.22	V
V_{LVR2}	低电压复位 2	—	掩模选项：3.15v	2.98	3.15	3.32	V
V_{LVR3}	低电压复位 3	—	掩模选项：4.2v	3.98	4.2	4.42	V
I_{OL}	输入/输出口灌电流	3V	V_{OL}=0.1V_{DD}	4	8	—	mA
		5V		10	20	—	mA
I_{OH}	输入/输出口源电流	3V	V_{OH}=0.9V_{DD}	-2	-4	—	mA
		5V		-5	-10	—	mA
R_{PH}	上拉电阻	3V	—	20	60	100	kΩ
		5V		10	30	50	kΩ
V_{OPA+}	0.5V_{DD} OPAS 的偏置电压	3V ~ 5.5V	—	0.475 V_{DD}	0.5 V_{DD}	0.525V_{DD}	V
R_{OPA}	A1P 与 VDD 之间的电阻值。A1P 与 GND 之间的电阻值。	5V	—	40	100	160	kΩ

LDO 直流电气特性

Ta=25℃

符号	参数	测试条件		最小	典型	最大	单位
		V_{IN}	条件				
V_{OUT}	输出电压	7V	I_{OUT}=10 mA	4.85	5	5.15	V
I_{OUT}	输出电流	7V	—	20	30	—	mA
$\triangle V_{OUT}$	负载校准电压	7V	1 mA≤I_{OUT}≤30 mA	—	60	100	mV
V_{DIF}	电压损耗	—	I_{OUT}=1 mA	—	100	—	mV
I_{SS}	电流消耗	7V	无负载	—	2.5	5	μA
$\frac{\Delta V_{OUT}}{\Delta V_{IN} \times V_{OUT}}$	线性校准	—	6V≤V_{IN}≤24V I_{OUT}=1 mA	—	0.2	—	%/V
V_{IN}	输入电压	—	—	—	—	24	V

EEPROM 直流电气特性

Ta=25℃

符号	参数	测试条件		最小	典型	最大	单位
		V_{IN}	条件				
V_{DD}	工作电压	—	—	2.2	—	5.5	V
I_{CC1}	工作电流	5V	读，在 100KH_Z	—	—	2	mA
I_{CC2}	工作电流	5V	写，在 100KH_Z	—	—	5	mA
V_{IL}	输入低电平	—	—	-1	—	0.3V_{DD}	V
V_{IH}	输入高电平	—	—	0.7V_{DD}	—	V_{DD}+0.5	V
V_{OL}	输出低电平	2.4V	I_{OL}=2.1 mA	—	—	0.4	V
I_{LI}	灌电流	5V	V_{IN}=0 或 V_{DD}	—	—	1	uA
I_{LO}	源电流	5V	V_{OUT}=0 或 V_{DD}	—	—	1	uA
I_{STB1}	静态电流	5V	V_{IN}=0 或 V_{DD}	—	—	4	uA
I_{STB2}	静态电流	2.4V	V_{IN}=0 或 V_{DD}	—	—	3	uA
C_{IN}	输入电容（注）	—	f=1MHZ 25℃	—	—	6	pF
C_{OUT}	输出电容（注）	—	f=1MHZ 25℃	—	—	8	pF

注：这些电容只是经过抽样测试并非 100%经过测试。

交流电气特性

Ta=25℃

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
f_{SYS1}	系统时钟—晶振	—	2.2V~5.5V	400	—	4000	kHz
		—	3.3V~5.5V	400	—	8000	kHz
		—	4.5V~5.5V	400	—	16000	kHz
f_{SYS2}	系统时钟—RC 振荡	—	4.5V~5.5V	—	8000	—	kHz
		—	4.5V~5.5V	—	16000	—	kHz
f_{16MRC}	16MHz 外部 RC 振荡	5V	R=130 kΩ, 25℃	15.2	16.0	16.8	MHz
		4.5V~5.5V	R=130 kΩ, -40℃~85℃（注）	14.4	16	17.2	MHz
f_{TIMER}	定时器 I/P 频率- TMR	—	2.2V~5.5V	0	—	4000	kHz
		—	3.3V~5.5V	0	—	8000	kHz
		—	4.5V~5.5V	0	—	24000	kHz
t_{WDTOSC}	看门狗振荡周期	3V	—	45	90	180	μs
		5V	—	32	65	130	μs
t_{WDT1}	看门狗溢出周期（WDT 内部时钟源）	—	预分频器 WS2,WS1,WS0=111	—	2^{15}	—	t_{WDTOSC}
t_{WDT2}	看门狗溢出周期（系统时钟）	—	预分频器 WS2,WS1,WS0=111	—	2^{17}	—	$*t_{SYS}$
t_{RES}	外部复位低电平脉宽	—	—	1	—	—	μs
t_{SST}	系统启动延迟时间	—	从 HALT 状态唤醒	—	1024	—	t_{SYS}
t_{INT}	中断脉冲宽度	—	—	1	—	—	μs
t_{LVR}	复位低电压宽度	—	—	0.25	1	2	ms

注：$*t_{SYS}=1/f_{SYS1}$ 或 $1/f_{SYS2}$

16MHz RC 振荡，需要接 130 kΩ到 VDD（工厂校准）。

运算放大器电气特性

Ta=25℃

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
直流电气特性							
	工作电压	—	—	3.0	—	5.5	V
	静态电流	5V	—	—	—	0.1	μA
V_{OPOS1}	输入偏置电压	5V	A_XOF3~0=1000	-15	—	15	mV
V_{OPOS2}	输入偏置电压	5V	通过校准	-4	—	4	mV
V_{CM}	共模电压范围	—	—	VSS	—	V_{DD}-1.4V	V
PSRR	电源抑制比	—	—	59	80	—	dB
CMRR	共模抑制比	5V	V_{CM}=0~ V_{DD}-1.4V	59	80	—	dB
交流电气特性							
A_{OL}	开环增益	—	—	80	100	—	dB
SR	正向压摆率，负向压摆率	—	空载	1.8	2.5	—	V/μs
GBW	增益带宽	—	R_L=1MΩ，C_L=100PF	—	500	—	kHz

比较器电气特性

Ta=25℃

符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
	比较器工作电压	—	—	3.0	—	5.5	V
	比较器工作电流	5V	—	—	—	200	μA
	比较器静态电流	5V	比较器除能	—	—	0.1	μA
V_{CMPOS1}	输入偏置电压	5V	COF3~0=1000	-15	—	15	mV
V_{CMPOS2}	输入偏置电压	5V	通过校准	-4	—	4	mV
V_{CM}	共模电压范围	—	—	V_{SS}	—	V_{DD}-1.4V	V
t_{PD}	比较器响应时间	—	有 2mV 的过载能力	—	—	2	μs

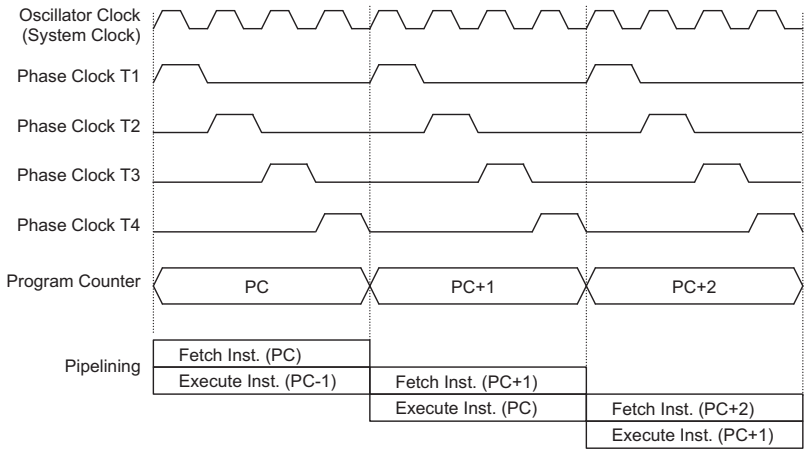
系统功能说明

内部系统结构是 HOLTEK 单片机具有良好运行性能的主要因素。由于采用 RISC 结构，此系列单片机具有高运算速度和高性能的特性。通过流水线的方式，指令的取得和执行同时进行，此举使得除了分支和调用指令外，其它指令都能在一个指令周期内完成。8 位的 ALU 参与指令集中所有的运算，它可完成算术运算、逻辑运算、移位、加、减和分支等功能，而内部的数据存取则以通过累加器或 ALU 的方式加以简化。有些寄存器在数据存储器中被实现，且可以直接或间接寻址。简单的寄存器寻址方式和结构特性，确保了在提供最大可靠度和灵活性的 I/O 控制系统时，仅需要少数的外部器件。这使得这些单片机适合用在低成本高产量的控制应用上。

指令执行时序

系统时钟由晶体/陶瓷振荡器，或是由 RC 振荡器提供，由四个内部生成的非重叠时序 T1~T4 组成。程序计数器在 T1 时自动加一并抓取一条新的指令。剩下的 T2~T4 时钟完成解码和执行功能，因此一个 T1~T4 时钟形成一个指令周期。虽然指令的取得和执行发生在连续的指令周期，但单片机流水线的结构会保证指令在一个指令周期内被有效的执行，特殊的情况发生在程序计数器的内容被改变的时候，如子程序的调用或跳转，在这情况下指令将需要多一个指令周期的时间去执行。

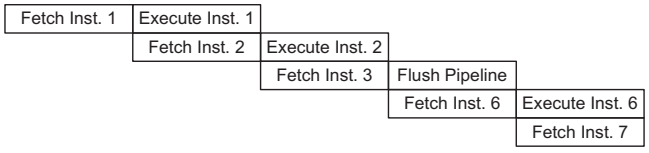
如果指令牵涉到分支，例如跳转或调用等指令，则需要两个机器周期才能完成指令执行。需要一个额外周期的原因是程序先用一个周期取出实际要跳转或调用的地址，再用另一个周期去实际执行分支动作，因此用户必须特别考虑额外周期的问题，尤其是在执行时间要求较严格的时候。



指令执行时序

```
1       MOV A,[12H]
2       CALL DELAY
3       CPL [12H]
4       :
5       :
6       NOP

    DELAY:
```



指令获取

程序计数器 — PC

程序执行期间，程序计数器用来指向下一条要执行的指令地址。除了 JMP 或 CALL 这些要求跳转到一个非连续的程序存储器地址之外，它会在每条指令执行完后自动增加一。程序计数器宽度会随着所选择单片机的程序空间容量而变化。必须要注意只有低 8 位，即程序计数器低字节寄存器 PCL，是可以让使用者直接读写。

当执行的指令要求跳转到非连续的地址时，如跳转指令、子程序调用、中断或复位等，单片机通过载入所需的地址到程序计数器来控制程序。对于条件跳转指令，一旦条件符合，下一条在现在指令执行时所取得的指令即会被摒弃，而由一个空指令周期来加以取代。

程序计数器低字节，即程序计数器低字节寄存器 PCL，可以通过程序控制取得，且它是可以读取和写入的寄存器。通过直接传送数据到这寄存器，一个程序短跳转可以直接被执行，然而因为只有低字节的运用是有效的，因此跳转被限制在同页存储器，即 256 个存储器地址的范围内，当这样一个程序跳转要执行时，需注意会插入一个空指令周期。

程序计数器低字节在程序控制下是完全可用的。PCL 的使用可能导致程序分支，所以额外的周期需要预先取得。有关 PCL 寄存器更多的信息可在特殊功能寄存器部份中找到。

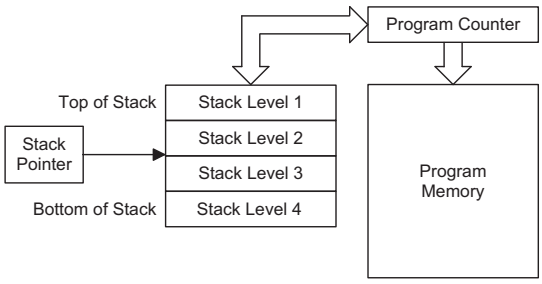
模式	程序计数器										
	b10	**b9**	**b8**	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
初始化复位	0	0	0	0	0	0	0	0	0	0	0
外部中断	0	0	0	0	0	0	0	0	1	0	0
定时/计数器 0 溢出	0	0	0	0	0	0	0	1	0	0	0
定时/计数器 1 溢出	0	0	0	0	0	0	0	1	1	0	0
比较器转换中断	0	0	0	0	0	0	1	0	0	0	0
条件跳跃	PC+2										
装载 PCL	**PC10**	**PC9**	**PC8**	@7	@6	@5	@4	@3	@2	@1	@0
跳转、子程序调用	#10	#9	#8	#7	#6	#5	#4	#3	#2	#1	#0
从子程序返回	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

程序计数器

注：PC10~PC8 ：目前程序计数器位　　S10 ~ S0 ：堆栈寄存器位
#10~ #0 ：指令代码位　　@7 ~ @0 ：PCL 位

堆栈寄存器 — STACK

堆栈寄存器是特殊的存储器空间，用来保存 PC 的值。堆栈寄存器有 4 层，它既不是数据存储器的一部分，也不是程序存储器的一部分，而且它不能读写。堆栈的使用是通过堆栈指针（SP）来实现的，堆栈指针也不能读出或写入。当发生子程序调用或中断响应时，程序计数器（PC）的值会被压入堆栈；在子程序调用结束或中断响应结束时（执行指令 RET 或 RETI），堆栈将原先压入堆栈的内容弹出，重新装入程序计数器中。在系统复位后，堆栈指针会指向堆栈顶部。



如果堆栈已满，并且发生了不可屏蔽的中断，那么只有中断请求标志会被记录下来，而中断响应会被抑制，直到堆栈指针（执行 RET 或 RETI 指令）发生递减，中断才会被响应。这个功能可以防止堆栈溢出，使得程序员易于使用这种结构。同样，如果堆栈已满，并且发生了子程序调用，那么堆栈会发生溢出，首先进入堆栈的内容将会丢失，要注意此类操作的结果会导致程序可能会执行到不可预期的段落分支。

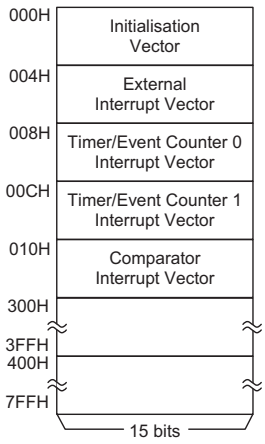
算术逻辑单元 —ALU

算术逻辑单元 ALU 是单片机中很重要的部份，执行指令集中的算术和逻辑运算。ALU 连接到单片机的数据总线，在接收相关的指令码后执行需要的算术与逻辑运算，并将结果储存在指定的寄存器，当 ALU 计算或操作时，可能导致进位、借位或其它状态的改变，而相关的状态寄存器会因此更新内容以显示这些改变，ALU 所提供的功能如下：

- 算术运算：ADD、ADDM、ADC、ADCM、SUB、SUBM、SBC、SBCM、DAA
- 逻辑运算：AND、OR、XOR、ANDM、ORM、XORM、CPL、CPLA
- 移位运算：RRA、RR、RRCA、RRC、RLA、RL、RLCA、RLC
- 递增和递减：INCA、INC、DECA、DEC
- 分支判断：JMP、SZ、SZA、SNZ、SIZ、SDZ、SIZA、SDZA、CALL、RET、RETI

程序存储器

程序存储器用来存放用户代码即储存程序。此系列单片机提供一次可编程存储器（OTP），使用者可编写他们的应用码到单片机中。使用适当的编程工具，OTP 单片机可以提供使用者灵活的方式来自由开发他们的应用，这对于除错或需要经常升级与改变程序的产品是很有帮助的。对于中小型量产，OTP 亦为极佳的选择。



程序存储器

结构

此系列单片机的程序存储器的容量是 2K×15 位。程序存储器用程序计数器来寻址，其中也包含数据、表格和中断入口，数据表格可以设定在程序存储器的任何地址，由表格指针来寻址。

特殊向量

在程序存储器中，某些位置为一些特殊的应用而预留的，如复位和中断。

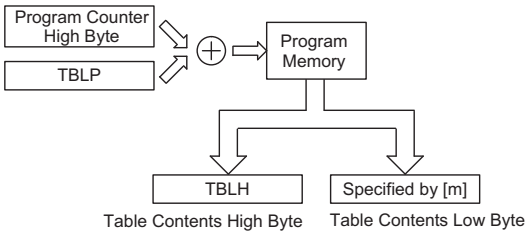
- 地址 000H
  该地址为程序初始化保留。系统复位后，程序总是从 000H 开始执行。
- 地址 004H
  该地址为外部中断中断服务程序保留。一旦 INT 引脚有触发信号输入，如果中断允许且堆栈未满，则程序会跳转到 004H 地址开始执行。外部中断的触发沿可以是上升沿,下降沿或者两种触发沿，该功能选择在 MISC 中。
- 地址 008H
  该地址为定时/计数器 0 中断服务程序保留。当定时/计数器 0 计数溢出，如果中断允许且堆栈未满，则程序会跳转到 08H 地址开始执行。
- 地址 00CH
  该地址为定时/计数器 1 中断服务程序保留。当定时/计数器 1 计数溢出，如果中断允许且堆栈未满，则程序会跳转到 0CH 地址开始执行。
- 地址 010H
  该地址为比较器中断服务程序保留。当比较器输出引脚有触发沿，如果中断允许且堆栈未满，则程序会跳转到 010H 地址开始执行。比较器中断的触发沿可以是下降沿也可以是两种触发沿，该功能选择由 MISC 中的 CMPES 位来决定。

查表

程序存储器中的任何地址都可以定义成一个表格，以便储存固定的数据。使用表格时，表格指针必须先行设定，其方式是将表格的低字节地址放在表格指针寄存器 TBLP 中。这个寄存器定义表格较低的 8 位地址。

在设定完表格指针后，表格数据可以使用“TABRDC [m]”或“TABRDL [m]”指令从当前的程序所在的存储器页或存储器最后一页中来查表读取。当这些指令执行时，程序存储器中表格数据低字节，将被传送到使用者所指定的数据存储器[m]，程序存储器中表格数据的高字节，则被传送到 TBLH 特殊寄存器，而高字节中未使用的位将被读取为“0”。

下图是查表中寻址/数据流程：



查表范例

以下范例说明在此系列单片机中，表格指针和表格数据如何被定义和执行。这个例子使用的表格数据用 ORG 伪指令储存在存储器的最后一页，在此 ORG 伪指令中的值为 700H，即 2K 程序存储器此系列单片机单片机中最后一页存储器的起始地址，而表格指针的初始值则为 06H，这可保证从数据表格读取的第一笔数据位于程序存储器地址 706H，即最后一页起始地址后的第六个地址。值得注意的是，假如“TABRDC [m]”指令被使用，则表格指针指向当前页。在这个例子中，表格数据的高字节等于零，而当“TABRDL [m]”指令被执行时，此值将会自动的被传送到 TBLH 寄存器。

```
tempreg1    db ?    ; temporary register #1
tempreg2    db ?    ; temporary register #2
      :
      :

mov     a,06h       ; initialize table pointer – note that this address
                    ; is referenced

mov     tblp,a      ; to the last page or present page
      :
      :

tabrdl tempreg1     ; transfers value in table referenced by table pointer
                    ; to tempreg1
                    ; data at prog. memory address 706H transferred to
                    ; tempreg1 and TBLH

dec     tblp        ; reduce value of table pointer by one

tabrdl tempreg2     ; transfers value in table referenced by table pointer
                    ; to tempreg2
                    ; data at prog. memory address 705H transferred to
                    ; tempreg2 and TBLH
                    ; in this example the data ″1AH″ is transferred to
                    ; tempreg1 and data  ″0FH″ to register tempreg2
                    ; the value ″00H″ will be transferred to the high byte
                    ; register TBLH
      :
      :

org   700h          ; sets initial address of last page
dc      00Ah, 00Bh, 00Ch, 00Dh, 00Eh, 00Fh, 01Ah, 01Bh
      :
      :
```

指令	表格区										
	b10	**b9**	**b8**	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
TABRDC[m]	PC10	PC9	PC8	@7	@6	@5	@4	@3	@2	@1	@0
TABRDL[m]	1	1	1	@7	@6	@5	@4	@3	@2	@1	@0

表格区

注： PC10 ~ PC8 ： 当前程序指针位　　@7 ~ @0 ： 表格指针 TBLP 位

因为 TBLH 寄存器是只读寄存器，不能重新储存，若主程序和中断服务程序都使用表格读取指令，应该注意它的保护。使用表格读取指令，中断服务程序可能会改变 TBLH 的值，若随后在主程序中再次使用这个值，则会发生错误。因此建议避免同时使用表格读取指令。然而在某些情况下，如果同时使用表格读取指令是不可避免的，则在执行任何主程序的表格读取指令前，中断应该先除能，另外要注意的是所有与表格相关的指令，都需要两个指令周期去完成操作。

数据存储器

数据存储器是内容可更改的 8 位 RAM 内部存储器，用来储存临时数据，且分为两部份。第一部份是特殊功能寄存器，这些寄存器有固定的地址且与单片机的正确操作密切相关。大多数特殊功能寄存器都可在程序控制下直接读取和写入，但有些被加以保护而不对用户开放。第二部份数据存储器是做一般用途使用，都可在程序控制下进行读取和写入。

结构

数据存储器的两个部份，即专用和通用数据存储器，位于连续的地址。全部 RAM 为 8 位宽度，数据存储器的起始地址都是 00H。常见的寄存器，如 ACC 和 PCL 等，全都具有相同的数据存储器地址。

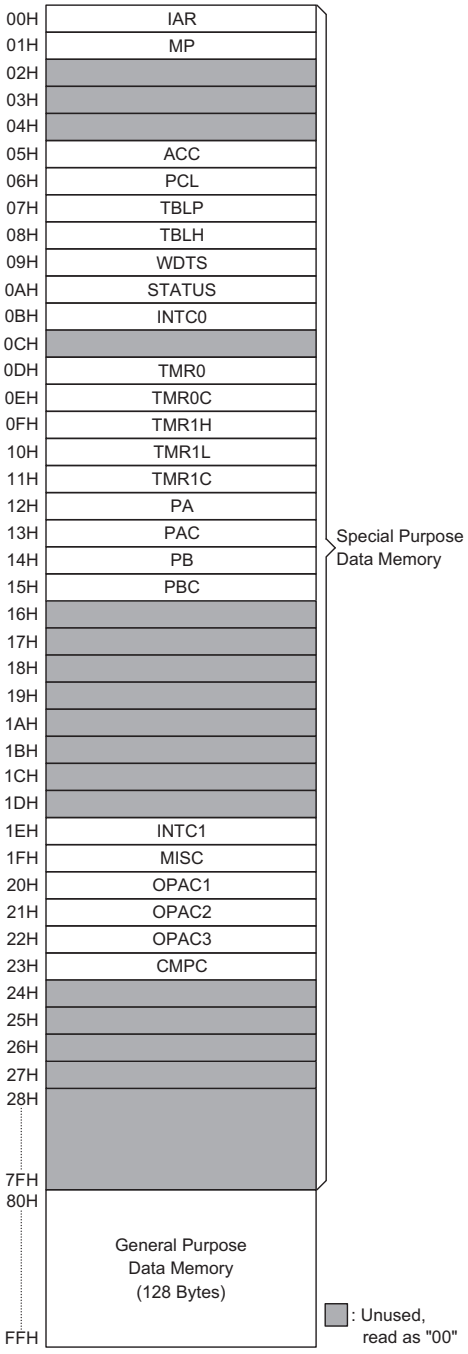
通用数据存储器

所有的单片机程序需要一个读/写的存储区，让临时数据可以被储存和再使用。该 RAM 区域就是通用数据存储器。这个数据存储区可让使用者进行读取和写入的操作。使用“SET [m].i”和“CLR [m].i”指令可对个别的位做置位或复位的操作，方便用户在数据存储器内进行位操作。

特殊数据存储器

这个区域的数据存储器是存放特殊寄存器的，这些寄存器与单片机的正确操作密切相关，大多数的寄存器可进行读取和写入，只有一些是被保护而只能读取的，相关细节的介绍请参看有关特殊功能寄存器的部份。要注意的是，任何读取指令对存储器中未定义的地址进行读取将得到“00H”的值。

注：除部分特殊位，大部分数据存储区可以通过“SET [m].i”和“CLR [m].i”直接寻址。数据存储器也可以通过指针 MP 间接寻址。



数据存储器

特殊功能寄存器

为了确保单片机能成功的操作，数据存储器中设置了一些内部寄存器。这些寄存器确保内部功能（如定时器和中断等）和外部功能（如 I/O 数据控制和比较器操作）的正确操作。在数据存储器中，这些寄存器以 00H 作为起始地址。在特殊功能寄存器和通用数据存储器的起始地址之间，有一些未定义的数据存储器，被保留用来做未来扩充，若从这些地址读取数据将返回 00H 值。

间接寻址寄存器 — IAR

间接寻址 IAR 寄存器位于数据存储区，并没有实际的物理地址。间接寻址的方法准许使用间接寻址指针做数据操作，以取代定义实际存储器地址的直接存储器寻址方法。在间接寻址寄存器 IAR 上的任何动作，将对间接寻址指针 MP 所指定的存储器地址产生对应的读/写操作。因为间接寻址寄存器并没有物理地址，所以直接读取 IAR 寄存器将返回 00H 的结果，而直接写入此寄存器则不做任何操作。

间接寻址指针 — MP

对于单片机，系统提供一个间接寻址指针 MP。由于这个指针在数据存储器中能象普通的寄存器一般被写入和操作，因此提供了一个寻址和数据追踪的有效方法。当对间接寻址寄存器进行任何操作时，单片机指向的实际地址是由间接寻址指针所指定的地址。全程寻址过程中 MP 的第 7 位没有作用。然而必须注意当读取间接寻址指针时，第 7 位将返回“1”。

以下例子说明如何清除一个具有 4 RAM 地址的区块，它们已事先定义成地址 adres1 到 adres4。

```
data .section ´data´
adres1  db ?
adres2  db ?
adres3  db ?
adres4  db ?
block           db ?
code .section at 0 ´code´
org     00h
start:
        mov a,04h                   ;setup size of block
        mov block,a
        mov a,offset adres1   ;Accumulator loaded with first RAM address
        mov mp,a                    ;setup memory pointer with first RAM address

loop:
        clr IAR                 ;clear the data at address defined by mp
        inc mp                  ;increment memory pointer
        sdz block                   ;check if last memory location has been cleared
        jmp loop

continue:
```

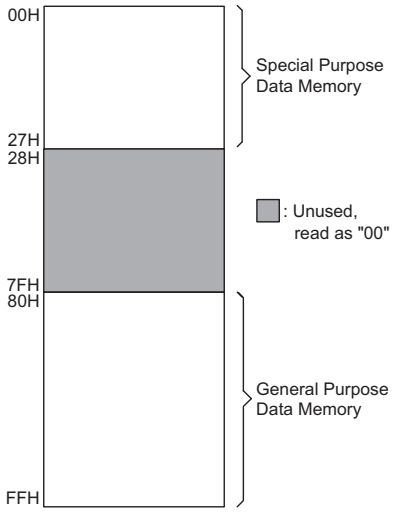
值得注意的是在上面的例子中并没有确定 RAM 地址。

累加器— ACC

对任何单片机来说，累加器是相当重要的，它与 ALU 所完成的运算有密切关系，所有 ALU 得到的运算结果都会暂时储存在 ACC 累加器。若没有累加器，ALU 必须在每次进行如加法、减法和移位的运算时，将结果写入到数据存储器，这样会造成程序编写和时间的负担。另外数据传送也常常牵涉到累加器的临时储存功能，例如在使用者定义的寄存器和另一个寄存器之间传送数据时，由于两寄存器之间不能直接传送数据，因此必须通过累加器来传送数据。

程序计数器低字节寄存器 — PCL

为了提供额外的程序控制功能，程序计数器较低字节设置在数据存储器的特殊功能区域内，程序员可对此寄存器进行操作，很容易的直接跳转到其它程序地址。直接给 PCL 寄存器赋值将导致程序直接跳转到程序存储器的某一地址，然而由于寄存器只有 8 位的长度，因此只允许在本页的程序存储器范围内进行跳转，而当使用这种运算时，要注意会插入一个空指令周期。



数据存储器结构

表格寄存器 — TBLP，TBLH

这两个特殊功能寄存器对储存在程序存储器中的表格进行操作。TBLP 为表格指针，指向表格数据的地址。它的值必须在任何表格读取指令执行前加以设定，由于它的值可以被如“INC”或“DEC”的指令所改变，这就提供了一种简单的方法对表格数据进行读取。表格读取数据指令执行之后，表格数据高字节存储在 TBLH 中。注意的是，表格数据低字节会被传送到使用者指定的地址。

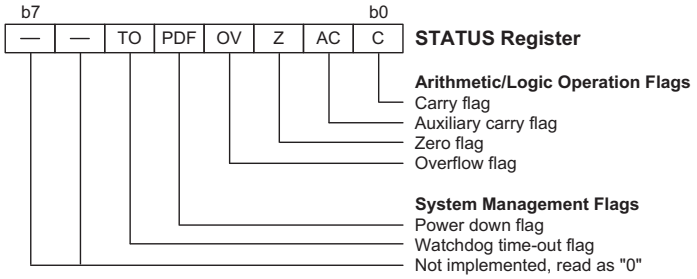
状态寄存器 — STATUS

8 位的寄存器（0AH）包含零标志位（Z）、进位标志位（C）、辅助进位标志位（AC）、溢出标志位（OV）、暂停标志位（PDF）和看门狗溢出标志位（TO），用来记录状态数据和控制运算顺序。

除了 TO 和 PDF 标志位外，状态寄存器中的位像其它大部份寄存器一样可以被改变，但任何数据写入到状态寄存器将不会改变 TO 或 PDF 标志位。另外，执行不同的指令后，与状态寄存器有关的运算可能会得到不同的结果。TO 标志位只会受系统上电、看门狗溢出、或执行“CLR WDT”或“HALT”指令影响。PDF 标志位只会受执行“HALT”或“CLR WDT”指令或系统上电影响。

Z、OV、AC 和 C 标志位通常反映最近运算的状态：

- 当加法运算的结果产生进位，或减法运算的结果没有产生借位时，则 C 被置位，否则 C 被清零，同时 C 也会被带进位/借位的移位指令所影响。
- 当低半字节加法运算的结果产生进位，或高半字节减法运算的结果没有产生借位时，AC 被置位，否则 AC 被清零。
- 当算术或逻辑运算结果是零时，Z 被置位，否则 Z 被清零。
- 当运算结果高两位的进位状态异或结果为 1 时，OV 被置位，否则 OV 被清零。
- 系统上电或执行“CLR WDT”指令会清零 PDF，而执行“HALT”指令则会置位 PDF。
- 系统上电或执行“CLR WDT”或“HALT”指令会清零 TO，而当 WDT 溢出则会置位 TO。



状态寄存器

另外当进入一个中断程序或执行子程序调用时，状态寄存器不会自动压入到堆栈保存。假如状态寄存器的内容是重要的且子程序可能改变状态寄存器的话，则需谨慎的去做正确的储存。

中断控制寄存器 — INTC0，INTC1

8 位的 INTC0，INTC1 寄存器用来控制外部和内部中断。通过标准的位操作指令来设定这些寄存器的位，每个中断的使能/除能功能可分别被控制。寄存器内的总中断位 EMI 控制所有中断的使能/除能，用来设定所有中断使能位的开或关。当一个中断程序被响应时，EMI 位将被清除，自动屏蔽其它中断，而执行“RETI”指令则会置位 EMI 位。

定时/计数寄存器

此系列单片机提供 1 个 8 位和 1 个 16 位定时/计数器。对于 8 位定时/计数器，TMR0 用于保存计数值，TMR0C 是控制寄存器。对于 16 位定时/计数器，TMR1L 和 TMR1H 用于保存计数值，TMR1C 是控制寄存器。

输入/输出端口和控制寄存器

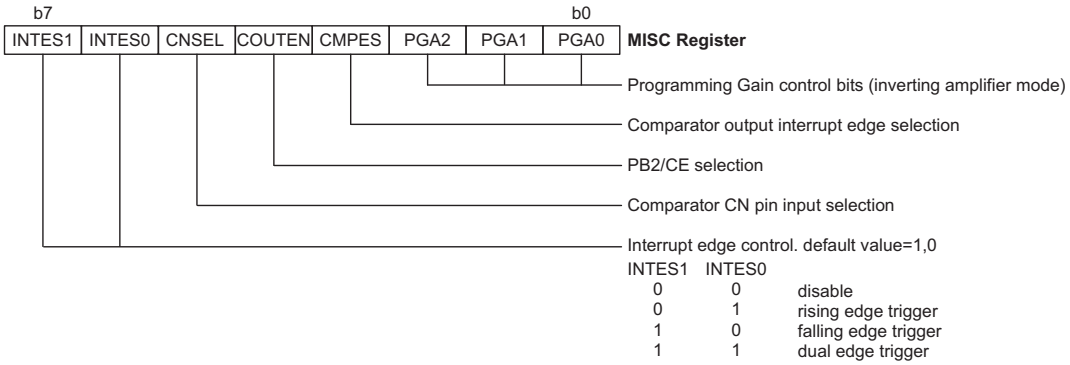
在特殊功能寄存器中，PA 口及其相应的控制寄存器 PAC 扮演了相当重要的角色。这些输入/输出端口在数据存储器的特殊存储器部分都有相对应的寄存器，PA 数据寄存器的内容表示的是输入或输出的值、而控制寄存器 PAC 里的内容是用来指定端口中哪一位为输出模式或哪一位为输入模式。要设定一个引脚为输入，控制寄存器对应的位必须设定成逻辑高，若引脚设定为输出，则控制寄存器对应的位必须设为逻辑低。程序初始化期间，在从输入/输出端口中读取或写入数据之前，必须先设定控制寄存器的位以确定引脚为输入或输出。使用“SET [m].i”和“CLR [m].i”指令可以直接设定这些寄存器的某一位。这种在程序中可以通过改变输入/输出端口控制寄存器中某一位而直接改变该端口输入/输出状态的能力是此系列单片机非常有用的特性。

比较器输入/输出

该系统中有两个比较器输入口一个比较器输出口，三个端口分别与 PB0、PB1 及 PB2 共用引脚。该比较器功能带有相应的中断功能，中断触发类型可由 MISC 寄存器中的位来决定。如果做为 I/O 口使用，那么全部的上拉电阻功能都可以保留，然而一旦做为比较器输入脚，则其相应的上拉电阻将会自动断开失效。

多功能控制寄存器 — MISC

该寄存器用于控制某些内部功能包括运算放大器的增益倍数，比较器中断触发沿选择，比较器输出标志，比较器输出口或 I/O 选择和外部中断触发沿选择。



多功能控制寄存器-MISC

运算放大器控制寄存器 —OPAC1，OPAC2，OPAC3

有三个寄存器用于控制三组运算放大器功能。三个寄存器中都有相应控制位来使能或除能运算放大器，有镜像输出及偏置电压校准功能。

EEPROM 数据存储器

HT46RS03E/HT46RS03PE 内部包含一个 128×8 位结构的 EEPROM 存储器。EEPROM 是一个可编程电可擦除只读存储器，是非失忆的存储器，当系统断电后数据依然被保存。各种数据被整合，使应用者可以适合多种应用。EEPROM 存储器允许直接存入像产品识别码、校准值、特殊用户数据、系统设定数据或产品信息等。

EEPROM 数据存储器的存取

内部 EEPROM 数据存储器有一个 I^2C 结构经由 2 线串行接口完成数据传送。这两根线是穿行数据线 SDA 和串行时钟线 SCL。双向 SDA 线用于从 EEPROM 写入和读取数据。SCL 是一个输入线，是用来提供读取和写入 EEPROM 数据的时钟信号。EEPROM 的两个脚与 I/O 口共用，如图表所示。掩膜选项的任何上拉电阻对 EEPROM 的共用脚仍然有效。需要注意的是如果这些管脚被用于普通 I/O 口，在这些管脚上

的任何信号都可能被认为是对 EEPROM 的读取或写入操作命令。如果这些信号不经意的在 EEPROM 的 SDA 线上产生，将造成意想不到的错误。内部 EEPROM 能够被共用的 I/O 口直接控制或被直接连接到一个外部 I^2C 总线并被一些其他的外部主控器件控制。对于后者的情形需要注意确认这些公用的 I/O 口作为 SDA 和 SCL 线都必须被设置为输入。

功能	HT46RS03PE			
	20-Pin		24-Pin	
EEPROM Pin	SDA	SCL	SDA	SCL
I/O Pin	PB4	PB5	—	—
Capacity	128×8 bits			

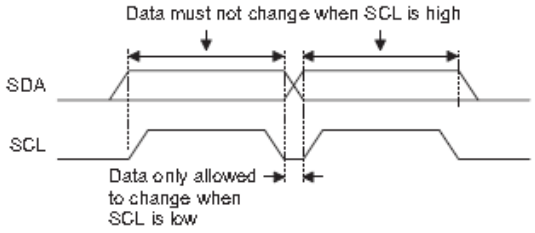
EPROM I/O 共用脚

- SCL 脚

这是 EEPROM 的时钟输入脚。当管脚的状态由低电平到高电平时写入数据到 EEPROM，由高电平到低电平时从 EEPROM 读出数据。SDA 线上的任何数据在下一个时钟上升沿时都将被发送到 EEPROM，因此只允许在 SCL 线为低电平时改变数据状态。如果 SCL 线为高电平，并且在 SDA 上的数据改变，这时将被认为是启动或停止状态。

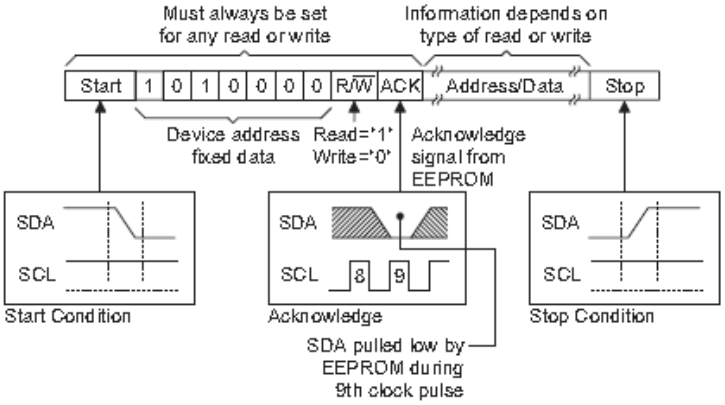
- SDA 脚

这是 EEPROM 的数据信号脚。这个管脚被用于读取和写入数据的双向脚。同样这个管脚有一个开漏输出，可以被写或者连接到其他外部开漏或者开集电极输出。否则它必须被连接到一个上拉电阻才能正确操作，利用掩膜选项可为相应的共用 I/O 口提供上拉电阻。



时钟和数据关系

读取或写入数据到 EEPROM 的流程如下图所示。所有操作必须以 START 条件开始和 STOP 条件结束。嵌入 START 和 STOP 条件之间的是设备地址、读/写位地址和数据信息。所有的控制信息、地址和数据以 8 位格式发送到 EEPROM，如果 EPROM 成功接收将回复一个应答信号已允许下一个 8 位被接收或传送。



数据传送协议

- START 状态

起始状态必须在器件地址和任何其它地址和数据信息传送之前被传送。起始状态是在 SCL 线为高电平时 SDA 线由高到低跳变。

- 器件地址

器件地址总是紧跟在起始状态后传送，即在 START 位后会输入 EEPROM 一个“1010000”的 7 位器件地址，器件地址在 SCL 线的上升沿时被送入 EEPROM。SDA 线上的数据必须在 SCL 线处于低状态时确定。当 SCL 线为高时，在 SDA 线上的任何改变都可能被认为是起始或停止状态。

- 读/写位

读/写操作位紧跟在器件地址之后，用于通知 EEPROM 是执行读或写操作。读操作时该位需要设置为高电平，写操作时该位需要设置为低电平。

- 应答

EEPROM 成功接收任何 8 位信息之后将拉低 SDA 线来传送一个应答信号。在 SCL 线提供第九个时钟脉冲时 EEPROM 产生应答信号。因此在 7 位器件地址和一个读/写位，共 8 位被传送到 EEPROM 后，在下一个时钟周期时 EEPROM 将回复一个应答信号。然后 8 位数据地址信息能够被传送到 EEPROM，然后 EEPROM 将在第九个时钟脉冲时再次回复一个应答信号。数据信息将按照相似的方法发送或接收。

- 数据地址

尽管 EEPROM 内部数据结构是 128×8 位，只需要 7 位地址存取数据，但一个 8 位地址必须被传送到 EEPROM。地址传送按照 MSB 位在前的格式，和 8 位传送一样，MSB 位（最高位）的是多余的，可以是任意 0 或 1。需要注意的是 SCL 时钟的上升沿时地址数据被送入 EEPROM。

- 停止状态

在任何读或写操作结束的最后停止信号必须被传送到 EEPROM。EEPROM 成功的接收到停止信号之后将进入低功耗模式并且等待下一个启动位。当 SCL 线为高电平 SDA 线由低电平到高电平的跳变时将执行停止命令。

读操作

有三种读取操作。有当前地址读取，随机读取和连续读取。

- 当前地址读取

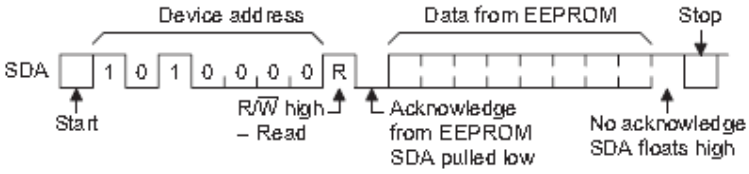
在 EEPROM 里面是一个内部计数器指向现在的 EEPROM 地址。每次一个读或写操作被执行后，内部计数器将自动加一。只要 EEPROM 通电，计数器的值将被存储，直到掉电这个计数器的值将丢失。如果需要读取数据，计数器将指向这个地址，不需要发送任何地址到 EEPROM。因此器件地址和读/写位信息被发送之后当前地址读操作将被执行，如果读/写位被设置为高，那么 EEPROM 将在第九个时钟脉冲时发送一个使 SDA 线拉低的应答信号，紧接着之后的 8 个时钟脉冲，它将传送这个内部地址处的 8 位数据。这 8 为数据被传送完成后，没有应答信号被发送，但是停止信号可能被接收器件传送，完成读操作。

- 随机读取

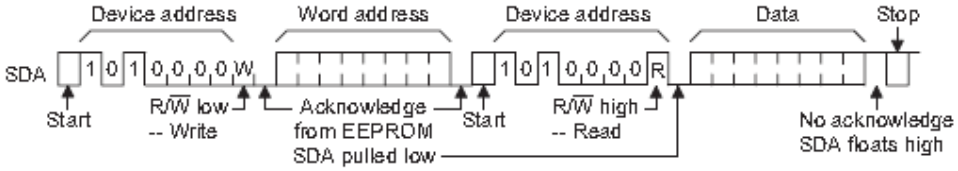
随机读取操作允许 EEPROM 内的任何地址处的数据被读取。实现这种情况需要首先发送起始信号、器件地址和读/写位。然后 EEPROM 发送应答信号之后，必须传送 EEPROM 内部地址信息。在这种情况下，当一个 EEPROM 地址被写入数据时，读/写位应该被设置为低电平。按照 MSB 在前的格式，8 位地址被传送，然后 EEPROM 将回复一个应答信号。这时内部计数器正指向被请求的地址，现在这个数据可以被发送来的另一个起始信号，器件地址和读/写位信息读出，但是这次读/写位设置为高表示读取操作。然后 EEPROM 通常会应答，相应请求地址的 8 位数据被读出。随机读取操作和当前地址读取操作相同，没有应答信号产生，但是停止信号必须在读取完成后被发送到 EEPROM。

- 连续读取

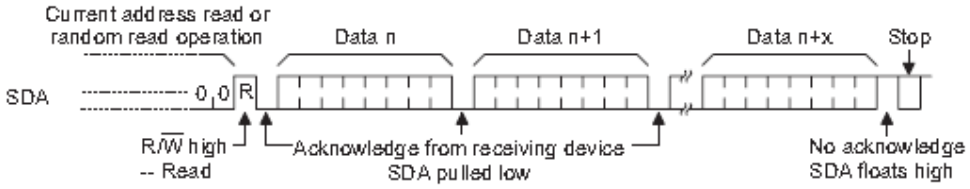
连续读取允许从 EEPROM 中连续读出超过一个字节以上的数据。它利用了指向 EEPROM 的内部地址的内部计数器每次读出数据后自动加一。连续读取是以随机读取或当前地址读取器件的第一个地址数据开始，然后利用当前地址读取或随机读出第一个字节数据后并不是发送停止信号，而是发送一个应答信号。应答信号是 8 个数据位发送完成后由接收设备拉低 SDA 线的时钟脉冲。当 EEPROM 接收到这个应答信号后，内部计数器将自动加一允许下一个字节数据发送。需要注意的是当地址计数器达到它的最大值时，它自动加一的下一个值将转为 0。



当前地址读取



随机读取



连续读取

写操作

这里仅仅只有一个写字节操作。

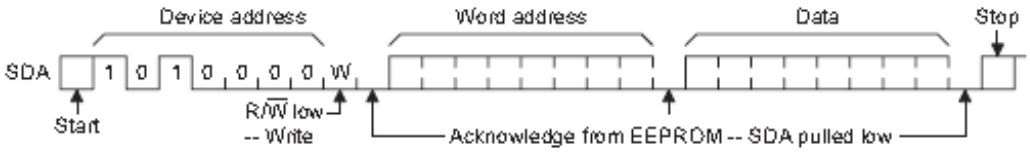
- 字节写

字节写操作允许一次单独写一个字节数据到 EEPROM。写字节操作需要起始信号，器件地址和读/写位首先被传送，然后收到由 EEPROM 发出的应答信号。EEPROM 的写数据内部地址信息必须被传送，假如要写一个地址到 EEPROM，读/写位必须设置为低电平，然后 8 位地址以 MSB 方式传送，EEPROM 将返回又一个应答信号，这时内部计数器正指向请求的地址。现在 8 位的数据开始以 MSB 方式写入到 EEPROM。当最后一位被发送到 EEPROM 之后，将返回一个应答信号。然后停止信号将被发送到 EEPROM。EEPROM 收到停止信号之后将进入内部写周期。当 EEPROM 运行时内部写周期被完全同步控制，其他的 EEPROM 操作不能执行。

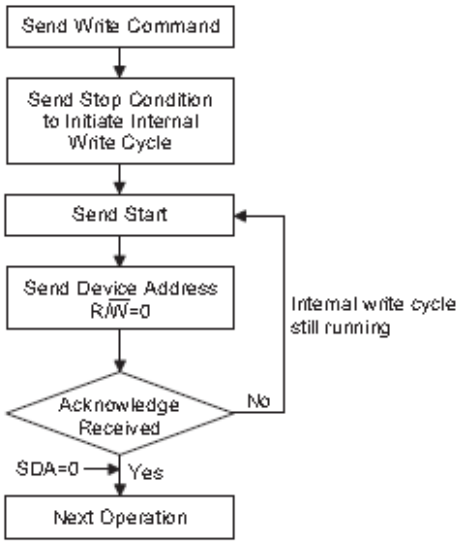
- 内部写周期

写操作完成并且当 EEPROM 接收到停止信号之后，将受到内部写周期的控制。当内部写周期执行时，EEPROM 将不能执行其他操作。因此在 EEPROM 写周期之前，延时时间必须与最大的写周期时间相等，t_{WS} 的值在 EEPROM 的交流电气特性表格有规定。然而最大时间只是其中的一种方法，有一个更好的方法是，可以用循环查询方式判断写周期是否完成。内部写周期需要起始信号，然后是从器件地址和读/写位为

低电平都必须被传送到 EEPROM。如果内部写周期完成，EEPROM 将返回一个应答信号，即将 SDA 线的电平拉低。如果内部写周期没有完成，则 EEPROM 将不会返回应答信号，即 SDA 线仍然为高电平。



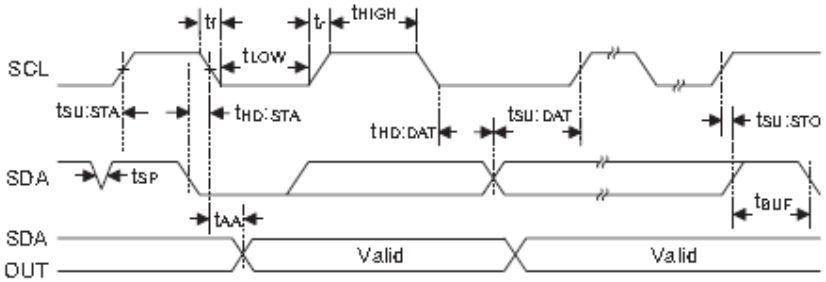
字节写



内部写周期循环查询

EEPROM 时序图

下面的时序图更详细的展示了 SCL 和 SDA 之间的关系。更详细的时序的时间值在 EEPROM 交流电气特性表里有介绍。这些时序在编程应用中必须小心处理，尤其是在高时钟速度的情况下。



时序图

输入/输出端口

HOLTEK 单片机的输入/输出端口控制具有很大的灵活性。此系列单片机的 16 个输入/输出引脚在程序设计时可设置为输入或输出功能，而且每一个引脚都有上拉电阻选项和某些指定引脚具有唤醒功能，提供给用户的是同一种输入/输出结构，以满足广泛应用的可能性。

每种单片机都有 PA 口，其相应的数据寄存器标示为 PA。这个数据存储器的在特殊寄存器表中都有对应指定地址。其中有 7 个端口可以实现输入/输出功能，另外一个仅能做为输入口使用。作为输入口操作时，

输入/输出引脚无锁存功能，输入数据必须在执行 “MOV A,[m]” 指令的 T2 上升沿准备好，m 表示端口地址。对于输出操作，所有数据是锁存的，而且持续到输出锁存被重写。

上拉电阻

在许多产品应用中 IO 端口处于输入状态时，需要外加一个上拉电阻来实现上拉的功能。为了免去这个外加的电阻，所有的输入/输出引脚（PA0~PA7；PB0~PB7）设置为输入时，可由内部连接上拉电阻（当然在系统频率选择外部晶振时，PB6 的上拉电阻是一直失效的）。这些上拉电阻可通过掩膜选项来选定，由一个 weak PMOS 晶体管来实现。

PA 口唤醒

HALT 指令可以使单片机会进入暂停功能以降低功耗，此特性对于电池供电系统或低功耗应用很重要。唤醒单片机有很多种方法，其中之一就是改变 PA0~PA7 引脚的逻辑电平，从高电平转为低电平。当使用暂停指令“HALT”迫使单片机进入暂停状态以后，单片机将保持闲置或低功耗状态，直到 PA 口上被选为唤醒输入的引脚电平发生下降沿跳变。这个功能特别适合于通过外部开关来唤醒的应用。值得注意的是 PA 端口的每一个引脚可以单独选为唤醒功能。

I/O 口控制寄存器

PA、PB 口分别有相应的控制寄存器 PAC、PBC，用来控制输入/输出配置。通过操作该控制寄存器，可以通过软件动态地控制当前 I/O 是做为 CMOS 输出还是带或不带上拉电阻的输入。每根 I/O 引脚都直接对应相应控制寄存器中的某一位。当要将一根 I/O 引脚置为输入状态时，仅需要将控制寄存器中相应的位置“1”，就可以保证该引脚外部的逻辑状态能够通过指令读到。当控制寄存器中的某一位被写“0”时，那么相应的 I/O 口即被当做 CMOS 输出口使用。另外当当前 I/O 口被做为输出口使用时，指令仍能读取寄存器的值，但要注意的是此时读到的值并非外部端口的逻辑状态，而是输出锁存器中的数据。

引脚共用功能

引脚的共用功能可以增加单片机的灵活程度。有限的引脚个数会严重的限制设计者，但是引脚的复用功能特性，可以很好的解决此类问题。复用功能输入/输出引脚的功能选择，有些是由掩膜选项进行设定，有些则是在应用程序中进行控制。

- 外部中断输入

  外部中断引脚 INT 与输入/输出引脚 PA1 共用。对于不需要外部中断输入的应用，这个共用引脚可做为普通的输入/输出引脚使用。

- 外部定时/计数器时钟输入

  外部定时器时钟输入引脚 TMR0，TMR1 分别与 PA3 和 PA4 共用。如果设定为定时/计数器为外部事件计数模式或脉宽模式，则该引脚即被选中做为定时/计数器输入，此时要保证定时/计数器相应的控制寄存器设置正确，同时该引脚必须通过端口寄存器设置为输入。注意的是即使该引脚设置为外部定时器时钟输入，输入/输出功能仍然有效。如果共用脚是做为普通 I/O 口使用，则外部计数输入功能必须除能，即一定要确定定时/计数器是工作在内部计数或没有工作的模式下。

- PFD 输出

  此系列单片机均提供 1 个 PFD 信号输出，引脚与 PA0 共用。PFD 输出功能可以通过掩膜选项选定。注意的是，必须将 PAC.0 设为输出以使能 PFD 输出。若 PAC 设置为输入，即使选择了 PFD 输出功能，此引脚仍将作为带上拉电阻的一般输入引脚使用。

- 比较器输入/输出

  芯片有两个比较器输入和一个比较器输出脚，分别与 PB0，PB1 和 PB2 共用。比较器功能和比较器中断转换类型由 MISC 寄存器中的位来选择。如果用于普通输入/输出口，则选择的上拉电阻仍然有效，如果用于比较器输入功能，则被选择的上拉电阻自动失效。

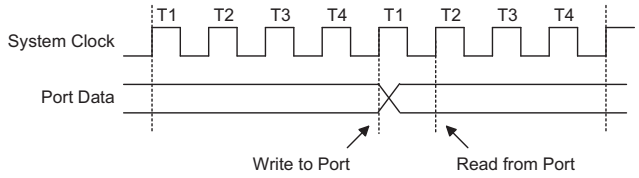
输入/输出端口结构

下列为输入/输出端口的内部结构图，可能与芯片内部实际的结构并不完全相同，只是用于帮助用户理解输入/输出端口。

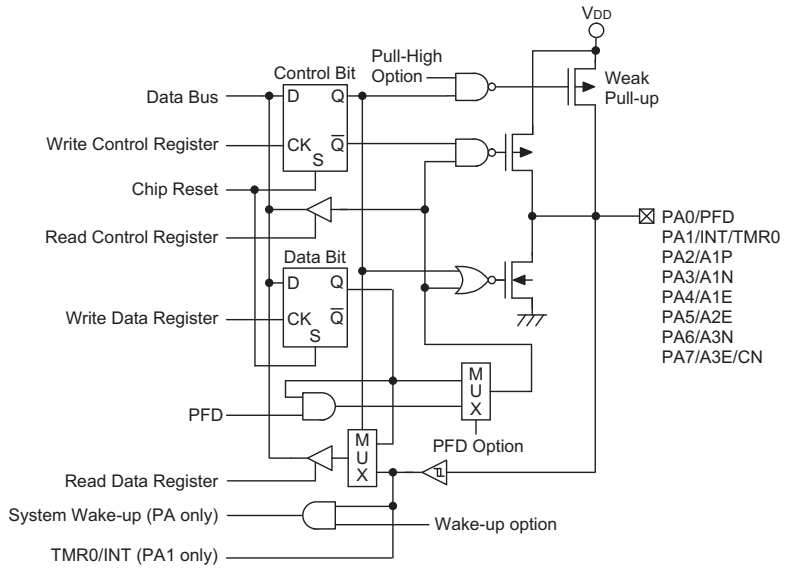
编程注意事项

在使用者的程序中，最先要考虑的事情之一是端口的初始化。复位之后，输入/输出口的数据寄存器及端口控制寄存器会被全部置高。这即意味着所有这些口会默认为输入状态，而其电平则取决于其它相连接电路以及是否选择了上拉电阻。如果程序有对控制寄存器操作，将某些引脚设定为输出状态，这些输出引脚最初都会有一个高电平输出，除非数据寄存器端口在程序中被预先设定。设置哪些引脚是输入及哪些引脚是输出，可通过设置正确的值到适当的端口控制寄存器，或者使用指令“SET [m].i”及“CLR [m].i”来设定端口控制寄存器中个别的位。注意的是当使用这些位控制指令时，一个读-修改-写的操作将会发生。单片机必须先读入整个端口上的数据，修改个别的位，然后重新把这些数据写入到输出端口。

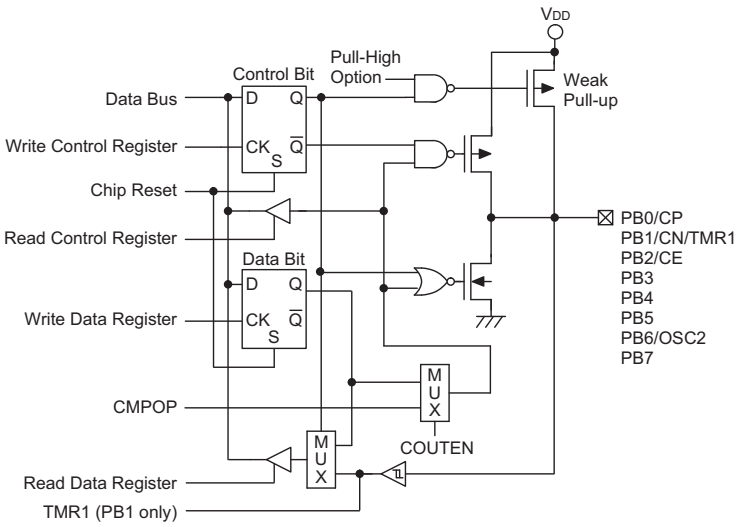
PA 口的每一个引脚都可以通过掩膜选项选择唤醒功能。当单片机在 HALT 状态时有很多方法可以唤醒单片机，其中之一就是 PA 口任一个引脚电平由高到低的转换，可以设定 PA 口的一个或多个引脚有这项功能。



读/写时序



PA0~PA7 输入/输出口



PB0~PB7 输入/输出口

定时/计数器

定时/计数器在任何单片机中都是一个很重要的部分，提供程序设计者一种实现和时间有关功能的方法。此系列单片机提供 1 个 8 位和 1 个 16 位的向上定时/计数器。每个定时/计数器有三种或四种不同的工作模式，可以被当作普通定时器、外部事件计数器、比较器事件计数器或脉冲宽度测量使用。定时器里的预分频器扩大了定时的范围。

有两种类型的寄存器和定时/计数器相关，第一种用来存储实际的计数值，赋值给此寄存器可以设定初始计数值，读取此寄存器可获得定时/计数器的内容。第二种是定时/计数器的控制寄存器，用来设置定时/计数器的选项，控制定时/计数器的工作模式。定时/计数器的时钟源可选项设定来自内部系统时钟或外部引脚输入。

在计数器工作在外部事件模式，外部时钟源就被使用到，外部时钟源会被提供到相应计数器（TMR0 或 TMR1 由控制器来选择）的引脚上。这些外部计数引脚是与其他输入/输出口复用。计数输入脚有下降沿或上升沿，会使内部计数将会开始向上加 1，这由定时/计数器的控制寄存器中 T0E 和 T1E 来决定。

配置定时/计数器输入时钟源

定时/计数器的时钟由其内部计数控制器设定可以来自不同的时钟源，当定时/计数器工作在定时器模式或脉冲宽度测量模式时，使用系统时钟作为计时来源。定时/计数器 0 的时钟还可以通过预分频器进行分频，分频系数由 TMR0C 寄存器中的 T0PSC0~T0PSC2 三位决定。

当定时/计数器工作在外部事件计数模式下，时钟源来自外部引脚 TMR0/ TMR1 或比较器输出。计数控制寄存器中 T0E 或 T1E 位用来决定有效计数触发沿是上升沿或下降沿，当外部引脚 TMR0/ TMR1 接收到有效的触发信号，比较器或运放放大器 1 输出口状态发生变化满足触发信号要求时，计数器值将会加一，比较器输出做为时钟源是 T0M0/T0M1，COUTSEL，T1M0/T1M1 位来选择。

定时/计数寄存器 — TMR0，TMR1L，TMR1H

定时/计数寄存器是特殊功能寄存器，用于储存实际定时器值。对于 8 位定时/计数器，该寄存器为 TMR0，而 16 位定时/计数器是用一对 8 位寄存器来保存数据，即 TMR1L 和 TRM1H。当用作内部定时且收到一个内部计时脉冲时，或用作外部计数且外部定时/计数器引脚状态发生跳变，比较器或运放放大器 1 输出口状态发生变化时，此寄存器的值将会加一。定时器将从预置寄存器所载入的值开始计数，直到计满 FFH（8 位定时/计数器）或 FFFFH（16 位定时/计数器），此时定时器溢出且会产生一个内部中断信号。定时器自动重新加载计数器初值并继续计数。

为了得到定时/计数器的最大计数范围 FFH（8 位定时/计数器）或 FFFFH（16 位定时/计数器），预置

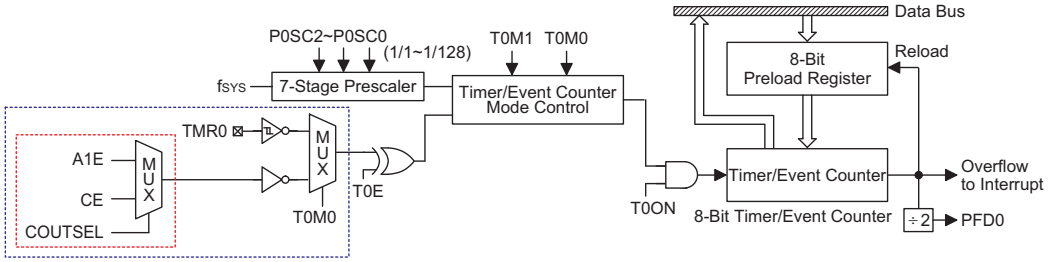
寄存器必须先清除为零。注意的是，上电后预置寄存器处于未知状态。定时/计数器在 OFF 条件下，如果把数据写入预置寄存器，这数据将被立即写入实际的定时器。而如果定时/计数器已经被打开且正在计数，在这个周期内写入到预置寄存器的任何新数据将保留在预置寄存器，只有在下一个溢出发生时才被写入实际定时器。

16 位的定时/计数器有低字节和高字节两个寄存器，即 TMR1L 和 TRM1H，对它们进行读写的时候需要注意。当对 TMR1L 寄存器写值时，只会将值装入到内部低字节 8 位缓冲器中，而不是直接写入 TMR1L 低字节寄存器。只有对高字节寄存器 TMR1H 写值时，才会把低字节缓冲器的数据写到低字节寄存器中。而对 TMR1H 写值，将直接写入 TMR1H，与此同时，将低字节缓冲器的数据写入低字节寄存器。因此，对 16 位寄存器赋值时，要先写低字节寄存器。若想读取低字节寄存器的内容，则先读取 TMR1H，把低字节缓冲器数据锁存到低字节寄存器中，之后，才能读取低字节寄存器。而读 TMR1L 只能读取低字节缓冲器的数据而不是低字节寄存器的内容。

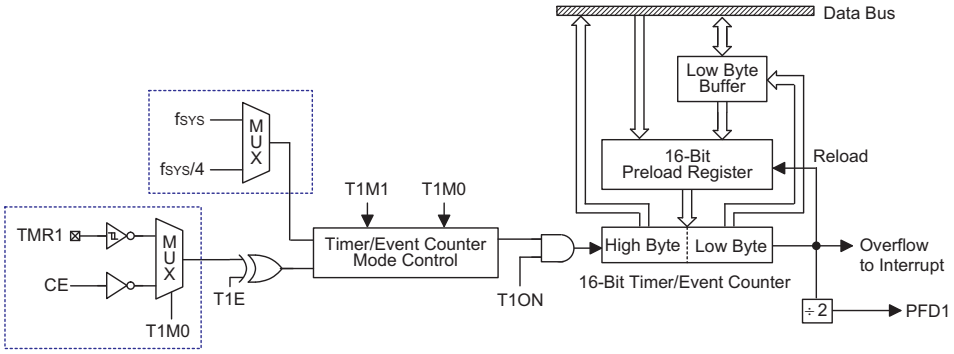
定时/计数控制寄存器 — TMR0C，TMR1C

HOLTEK 单片机的定时/计数器能灵活地应用于四种不同的工作模式，至于选择工作在哪一种模式则是由控制寄存器的内容决定。定时/计数器有 2 个控制寄存器 TMR0C 和 TMR1C。定时/计数控制寄存器和定时/计数寄存器一起控制定时/计数器的全部操作。在使用定时/计数器之前，必须正确地设定定时/计数控制寄存器，以便保证定时器能正确操作，而这个过程通常在程序初始化期间完成。

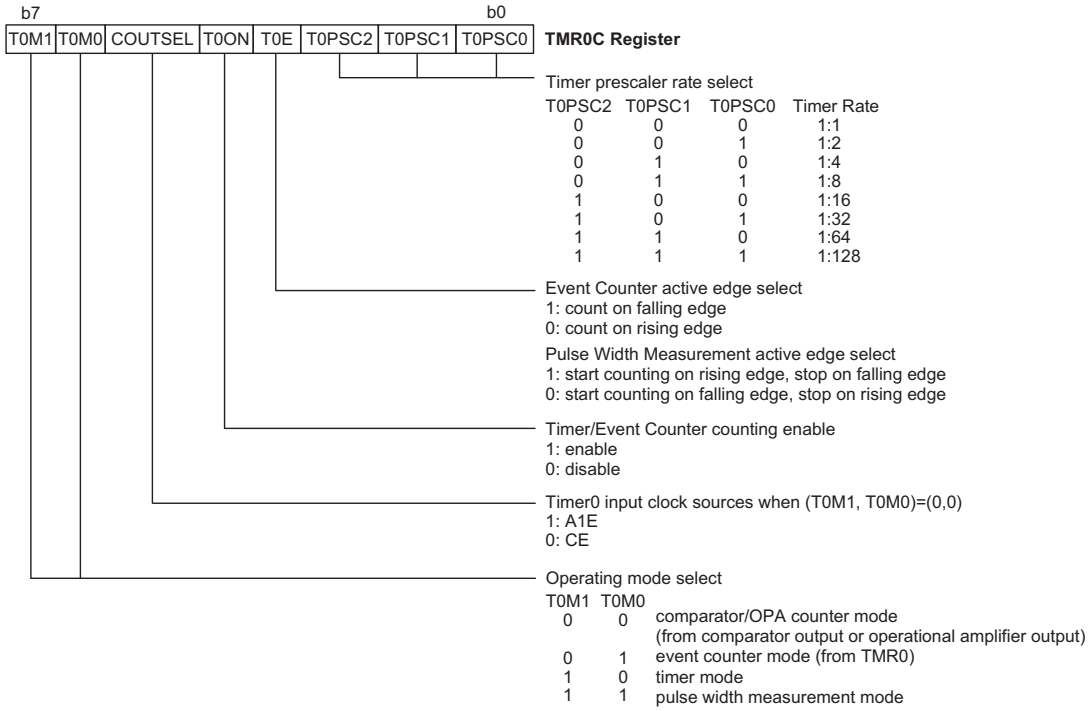
为了确定定时/计数器工作在哪一种模式，定时器模式，外部事件计数模式、比较器/运算放大器计数模式或脉冲宽度测量模式，必须正确设置定时/计数控制寄存器第 7 位和第 6 位的逻辑电平，即 T0M1/T0M0 或 T1M1/T1M0 位（取决于所选定时/计数器）。定时/计数器打开位 T0ON 或 T1ON（取决于所选定时/计数器），即定时/计数控制寄存器的第 4 位，是定时器控制的开关，设定为逻辑高时，计数器开始计数，而清零时则停止计数。对于定时/计数器 0，相应的控制寄存器第 0 位~第 2 位决定输入定时预分频器中的分频系数。如果使用外部计时源，预分频器将不作用。如果定时器工作在事件计数或脉冲宽度测量模式，T0E 或 T1E（取决于所选定时/计数器）的逻辑电平，即定时/计数控制寄存器的第 3 位将可用来选择上升沿或下降沿触发。



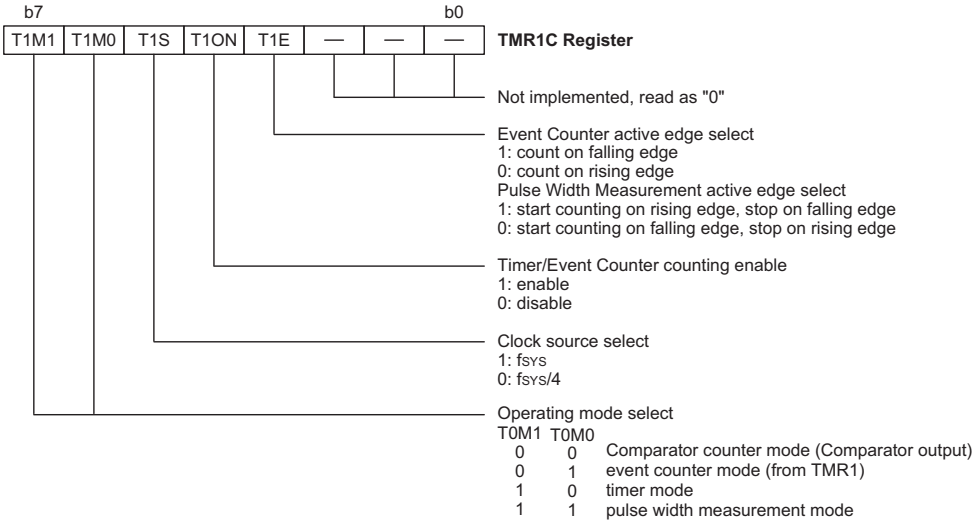
8 位定时/计数器 0 的结构



16 位定时/计数器 1 的结构



定时/计数器 0 控制寄存器



定时/计数器 1 控制寄存器

配置定时器模式

在这个模式，定时/计数器用来测量固定时间间隔，当定时器发生溢出时，就会产生一个内部中断信号。要工作在这个模式，位对 T0M1/T0M0 或 T1M1/T1M0（由所选的定时/计数器决定）必须分别设为 1 和 0。

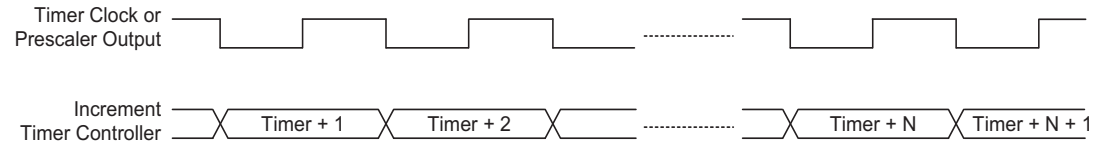
这个模式下，内部时钟作为定时器的时钟源。对 TMR0 而言，这个内部时钟就是 f_{SYS}，可以通过控制寄存器的 T0PSC2~T0PSC0 预分频选择位对时钟源进行分频。对于 TMR1 而言，内部时钟就是 $f_{SYS}/4$，无预分频可选。控制寄存器的第 4 位 T0ON/T1ON，为使能位，将其设置为高电平，即可使定时器开始工作。每次内部时钟有下降沿发生时定时器会加一。当定时器已满溢出时，会产生中断信号且定时器会重新载入预置寄存器内的值，然后续向上计数。计时中断所产生的中断请求信号也是一个唤醒睡眠状态的信号源。若对寄存器 INTC0 中的定时/计数器中断使能位（ET0I 或 ET1I）清零，则定时器中断禁止。

配置事件计数模式

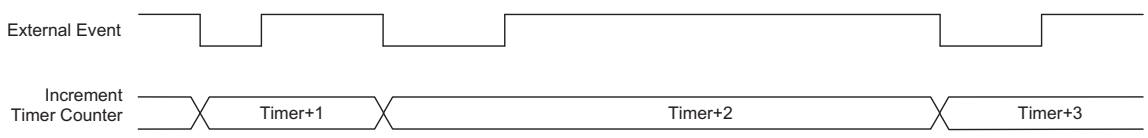
在这个模式下，发生在外部事件计数引脚上逻辑事件改变的次数，可以通过内部计数器来记录。为了使定时/计数器工作在外部事件计数模式，要工作在这个模式，位对 T0M1/T0M0 或 T1M1/T1M0（由所选的定时/计数器决定）必须分别设为 0 和 1。这个模式下，外部时钟作为定时器的时钟源，不能进行预分频。控制寄存器的第 4 位 T0ON/T1ON，为使能位，将其设置为高电平，即可使定时器开始工作。当触发沿选择位，即控制寄存器的第 3 位 T0E/T1E，设置为低电平时，外部时钟引脚每次接收到由低到高的电平跳变将使计数器值加一。如果触发沿选择位设置为高电平，外部时钟引脚每次接收到由高到低的电平跳变将使计数器值加一。与其他模式一样，当计数器计满时，计数器将溢出且自动从预置寄存器中加载初值。若定时中断允许，将会产生一个中断信号。因外部计数引脚是与输入/输出口复用引脚，为了确保定时/计数器工作在事件计数器模式，必须注意两点，首先是要将 T0M1/T0M0 或 T1M1/T1M0 位设定在事件计数器模式，其次是确定端口控制寄存器将这个引脚设定为输入状态。值得注意的是计数器的溢出也是唤醒暂停模式的一种方法，不管中断是否允许。若对寄存器 INTC0 中的定时/计数器中断使能位（ET0I 或 ET1I）清零，则定时器中断禁止。

配置比较器/运算放大器计数模式

当计数器工作在比较器/运算放大器计数模式时，T0M1/T0M0 两位必须清 0。计数使能位 T0ON 位必需置高以来启动 TMR0 开始工作。如果 T0E 设置为低电平时，当比较器/运算放大器输出一个低到高的电平跳变将使计数器值加一。如果触发沿选择位 T0E 设置为高电平，则比较器/运算放大器输出一个高到低的电平跳变将使计数器值加一。与其他模式一样，当计数器计满时，计数器将溢出产生一个内部中断请求信号，且自动从预置寄存器中加载初值。因外部计数引脚是与输入/输出口复用引脚，为了确保定时/计数器工作在相应模式，必须注意一点，即要将 T0M1/T0M0 位设定在比较器/运算放大器计数器模式。值得注意的是计数器的溢出也是唤醒暂停模式的一种方法，不管中断是否允许。若对寄存器 INTC0 中的定时/计数器中断使能位 ET0I 或 ET1I 清零，则相应定时器中断禁止。



定时器模式时序图

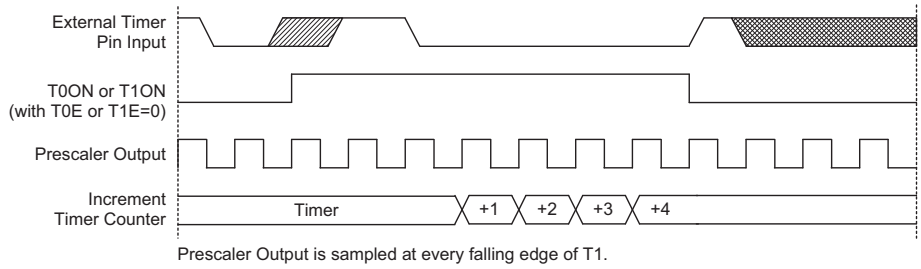


事件计数器模式时序图

配置脉冲宽度测量模式

在这个模式下，可以测量外部定时/计数器引脚上外部脉冲宽度。为了使定时/计数器工作在脉冲宽度测量模式，控制寄存器中的工作模式选择位 T0M1/T0M0 或 T1M1/T1M0（取决于哪一个被使用）全部要置高。控制寄存器的 T0ON/T1ON 为使能位，当其设置为高电平后，使能定时/计数器工作。当外部定时/计数器引脚接收到触发沿时，定时/计数器才开始计数。当触发沿选择位 T0E/T1E，设置为低电平时，当外部定时/计数器引脚接收到一个由高到低的电平跳变时，定时/计数器将开始计数直到外部定时/计数器引脚回到它原来的高电平，此时定时/计数器使能位将自动清除为零，且定时/计数器停止计数。如果触发沿选择位 T0E/T1E 设置为高电平，当外部定时/计数器引脚接收到一个由低到高的电平跳变时，定时/计数器将开始计数直到外部定时/计数器引脚回到它原来的低电平，此时定时/计数器使能位 T0ON/T1ON 将自动清除为零，且定时/计数器停止计数。注意，在脉冲宽度测量模式中，当外部定时/计数器引脚上的外部控制信号回到它原来的电平时，使能位将自动地清除为 0。而在其它两种模式下，使能位只能在程序控制下才会被清除为 0。这时定时/计数器中的值可被程序读取，并由此得知外部定时/计数器引脚接收到的脉冲的长度。因为使能位已被清除，任何在外部定时/计数器引脚的电平跳变将被忽略，直到使能位再次被程序设定为逻辑高，定时/计数器才开始脉冲宽度测量。用这种方法可轻松地完成单个脉冲的测量。

注意的是在这种模式下，定时/计数器是通过外部定时/计数器引脚上的逻辑跳变来控制，而不是通过逻辑电平。与其他的几种模式一样，当计数器计满，计数会溢出并产生相应的中断信号。定时器会重新载入已经载入到预置寄存器的值，然后续向上计数。如果外部信号输入的引脚是与其他输入/输出口复用，为保证其是做为脉宽量测的输入脚，有两点要注意操作：首先是要将 T0M1/T0M0 或 T1M1/T1M0 位设定在外部脉宽计数器模式，其次是确定端口控制寄存器将这个引脚设定为输入状态。值得注意的是计数溢出也是唤醒暂停模式的一种方法，不管中断是否允许。若对寄存器 INTC0 中的定时/计数器中断使能位（ET0I 或 ET1I）清零，则定时器中断禁止。



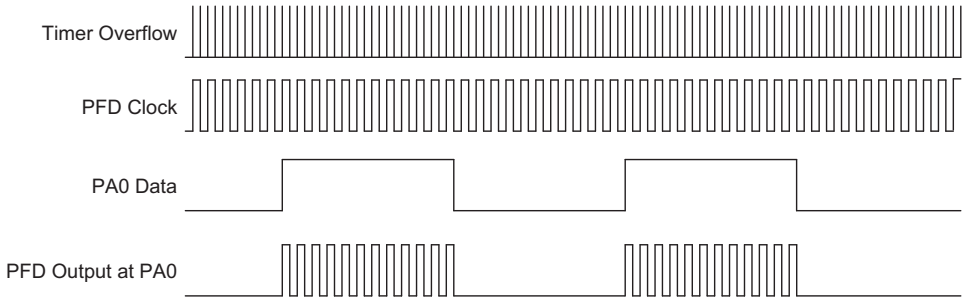
脉冲宽度测量模式时序图

可编程分频器 – PFD

PFD 输出引脚与 PA0 引脚共用。PFD 时钟源和开/关由掩膜选项选定，如果不选择该功能，则这个引脚作为普通的输入/输出引脚使用。PFD 电路使用定时器溢出信号作为它的时钟源。PFD 的输出频率由载入到定时/计数器寄存器的值和定时器预分频器的值控制。系统时钟被预分频器分频后的时钟源，进入定时器计时，定时器从预置寄存器的值开始往上计数，直到计数值满而产生溢出信号，导致 PFD 输出改变状态。定时器将自动地重新载入预置寄存器的值，并继续向上计数。

要使 PFD 正确运作，必须将 PA 控制寄存器 PAC 的第 0 位设置为输出。如果把它设置为输入，则 PFD 输出不工作，该引脚仍是作为普通的输入引脚使用。只有把 PA0 位置“1”，PFD 输出引脚才会有输出。这个输出数据位被用作 PFD 输出的开/关控制。注意，如果 PA0 输出数据位被清为“0”，PFD 输出将为低电平。

假如系统时钟使用晶体振荡器，则使用这种频率产生的方法可以产生非常精确的频率值。



PFD 输出控制

预分频器

TMR0C 控制寄存器的 T0PSC0~T0PSC2 可以用来定义定时/计数器 0 中内部时钟源的预分频系数。

输入/输出接口

当定时/计数器 0 和定时/计数器 1 工作在外部事件计数模式或脉冲宽度测量模式时，定时/计数器需要使用外部计数引脚以确保正确的动作。由于此批引脚为共用，必须保证将其设置为定时/计数器 0 和定时/计数器 1 的外部输入而不是普通的输入/输出，这需要将定时/计数器 0 和定时/计数器 1 控制寄存器内的模式选择位设置为外部事件计数或脉宽测量模式，另外端口控制寄存器 PAC 中的相应位必须设置为高，以保证处于输入状态。即使该引脚作为定时/计数器 0 和定时/计数器 1 输入，上拉电阻仍有效。

编程注意事项

当定时/计数器工作在定时器模式时，定时器的时钟源使用内部系统时钟器，与单片机所有运算都能同步。在这个模式下，当定时器寄存器溢出时，单片机将产生一个内部中断信号，使程序进入相应的内部中断向量。对于脉冲宽度测量模式，定时/计数器的时钟源也是使用内部系统时钟，但定时/计数器只有在正确的逻辑条件出现在外部定时/计数器输入引脚时才执行动作。由于这个外部事件没有和内部定时器时钟同步，只有当下一个定时器时钟到达时，单片机才会看到这个外部事件，因此在测量值上可能有小的差异，需要程序设计者在程序应用时加以注意。同样的情况发生在定时器设置为外部事件计数模式时，它的时钟来源是外部事件，与内部系统时钟或者定时器时钟不同步。

当读取定时/计数器值或写数据到预置寄存器时，计数时钟会被禁止以避免发生错误，但这样做可能会导致计数错误，所以程序设计者应该考虑到这点。在第一次使用定时/计数器之前，要仔细确认有没有正确地设定初始值。中断控制寄存器中的定时器使能位必须正确的设置，否则相应定时/计数器内部中断仍然无效。定时/计数器控制寄存器中的触发边沿选择、定时/计数器工作模式和时钟源控制位也必须正确的设定，以确保定时/计数器按照应用需求而正确的配置。在定时/计数器打开之前，必须确保先载入定时/计数器寄存器的初始值，这是因为在上电后，定时/计数器寄存器中的初始值是未知的。定时/计数器初始化后，可以使用定时/计数器控制寄存器中的使能位来打开或关闭定时器。需要注意的只有当定时/计数器的工作模式设置好以后，才把定时/计数控制寄存器中的使能位置高，打开定时/计数器。若在定时/计数器使能位为高的情况下对工作模式选择位进行修改，则定时/计数器将产生误动作。

当定时/计数器产生溢出，中断控制寄存器中相应的中断请求标志将置位。若中断允许，将会产生一个中断信号。不管中断是否允许，在 HALT 状态下，定时/计数器的溢出也会产生唤醒信号。这种情况可能发生在外部事件计数模式下，当外部信号状态不停地变化的时候，定时/计数器不停地向上计数直至溢出并唤醒系统。若在 HALT 模式下，不需要定时器中断唤醒系统，可以在进入 HALT 前将相应中断请求标志位置位。

定时/计数器应用范例

下面范例程序以一个 8 位定时/计数器，一个 16 位定时/计数器为对象，说明通过操作相应的的寄存器，如何实现中断使能及管理。另外还需注意的是，怎样通过寄存器的第 4 位来使能定时/计数器和通过对该位清零来关闭定时/计数器。此应用范例设置定时/计数为定时模式，时钟来源于内部的系统时钟。

```
org 04h                    ; external interrupt vectors
reti
org 08h                    ;Timer Counter 0 interrupt vector
jmp tmr0int                ;jump here when Timer 0 overflows
org 0Ch                    ;Timer Counter1 interrupt vector
jmp tmr1int                ;jump here when Timer 1 overflows
:
:
org 20h                    ; main program
:
:
; internal timer 0 interrupt routine
tmr0int:
:
; timer0 main program placed here
:
reti
:
; internal timer 1 interrupt routine
tmr1int:
:
; timer1 main program placed here
:
reti
:
begin:
; setup Timer0 registers
mov a, 09bh                ; setup Timer0 preload value
mov tmr0,a
mov a, 081h                ; setup timer0 control register
mov tmr0c, a               ; timer mode and prescaler set to /2
;setup timer1 registers
clr tmr1l
clr tmr1h                  ;clear timer register to give maximum count value
mov a, 080h                ; setup timer1 control register
mov tmr1c, a               ;timer mode - Timer 1 has no prescaler
;setup interrupt register
 mov a, 00dh               ; enable master interrupt and both timer
;interrupts
mov intc0, a
:
:
set tmr0c.4                ; start timer0
set tmr1c.4                ; start timer1
```

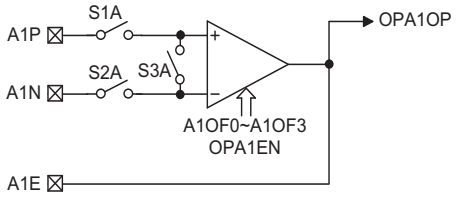
OPA（运算放大器）

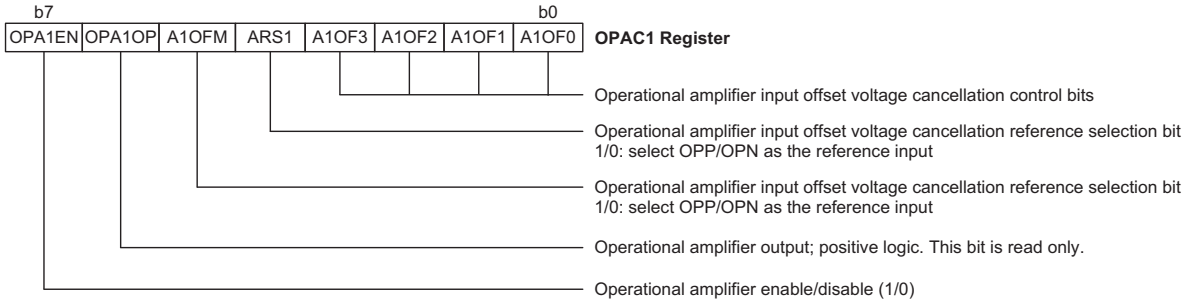
在芯片中提供三组运算放大器，OPA1，OPA2，OPA3。这些 OPAs 能用来根据用户的特定要求将信号放大。它们可能控制软件控制来使能或除能。其中 OPA2 还包含有一个增益控制功能，是通过一个中断寄存器设定的。

运算放大器寄存器

内部运算放大器通过三个寄存器，OPAC1、OPAC2、OPAC3 完全控制每一个放大器。它们控制使能/除能功能，极性和基准功能。

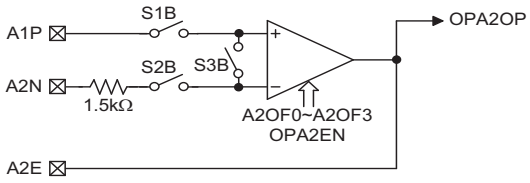
ARS1	A1OFM	S1A	S2A	S3A
0	0	ON	ON	OFF
0	1	OFF	ON	ON
1	0	ON	ON	OFF
1	1	ON	OFF	ON

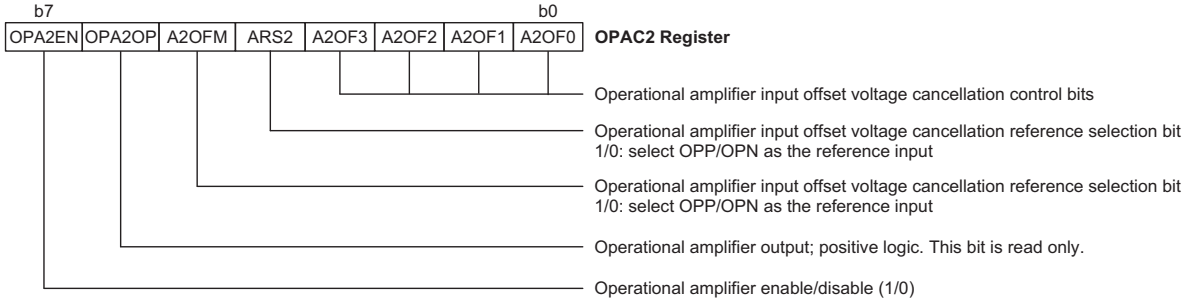




运算放大器控制寄存器 ——OPAC1

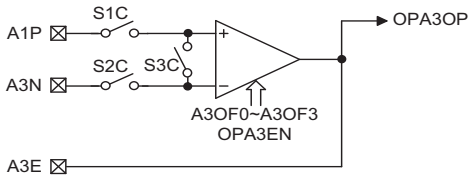
ARS2	A2OFM	S1B	S2B	S3B
0	**0**	**ON**	**ON**	**OFF**
0	**1**	**OFF**	**ON**	**ON**
1	**0**	**ON**	**ON**	**OFF**
1	**1**	**ON**	**OFF**	**ON**

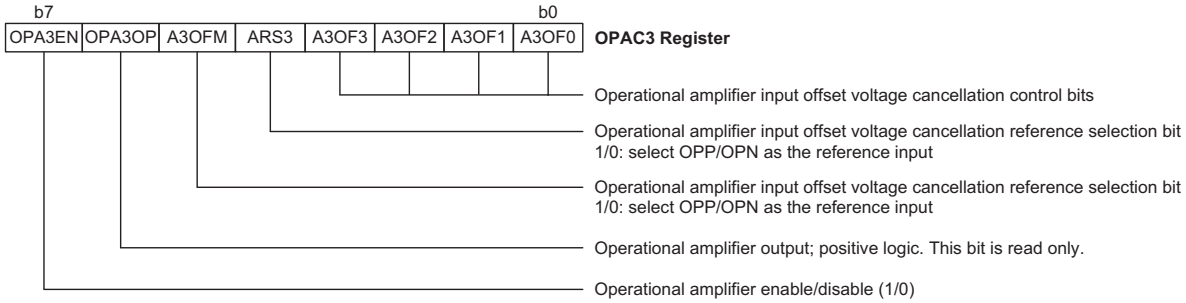




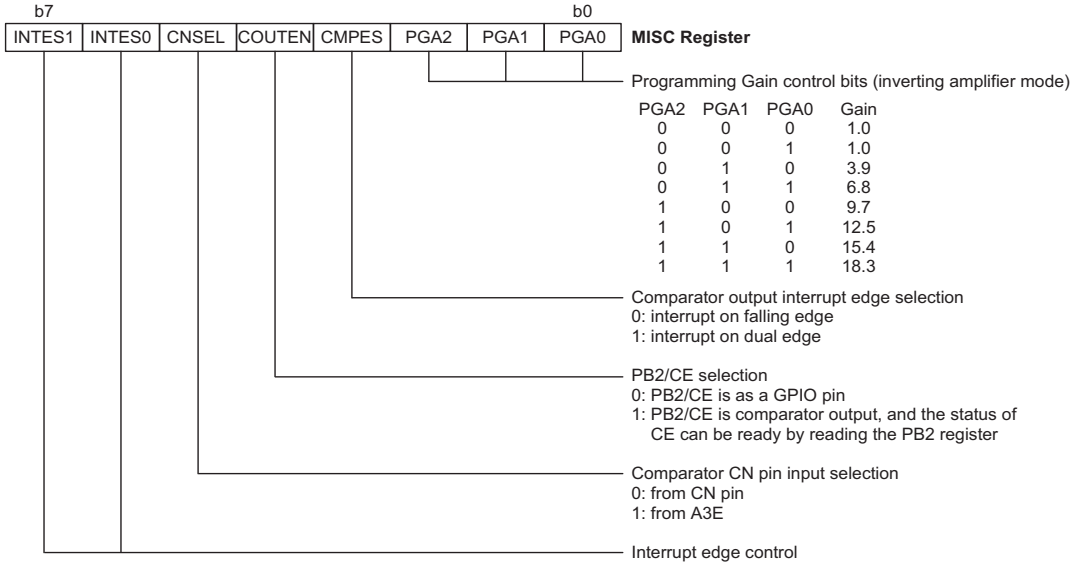
运算放大器控制寄存器——OPAC2

ARS3	A3OFM	S1C	S2C	S3C
0	0	ON	ON	OFF
0	1	OFF	ON	ON
1	0	ON	ON	OFF
1	1	ON	OFF	ON





运算放大器控制寄存器——OPAC3



多功能控制寄存器——MISC

运算放大器操作

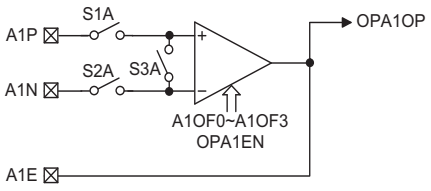
运算放大器与内部的连接如方框图所示。OPA3 的输出与内部比较器的连接方框图所示。每一个内部运算放大器都有自己的控制寄存器，即 OPAC1、OPAC2 和 OPAC3，每一个都能用于控制它们自己的使能/除能、输出极性和校正程序功能。MISC 寄存器用于程序控制 OPA2 的增益功能。

每个内置的 OPAs 的输入偏置电压可在共模输入模式下校准电压值。

校准步骤如下：

1. 置 A1OFM=1，选择偏置电压补偿模式 - 关闭开关 S3A。
2. 置位 ARS1 位，选择参考电压输入引脚 – 关闭开关 S1 或 S2。
3. 调整位 A1OF0~A1OF3 直到输出状态发生变化。
4. 置 A1OFM=0，选择正常操作模式。
5. 对于 OPA2 及 OPA3 的校准也是重复以上 1~4 的动作，操作的寄存器不同。

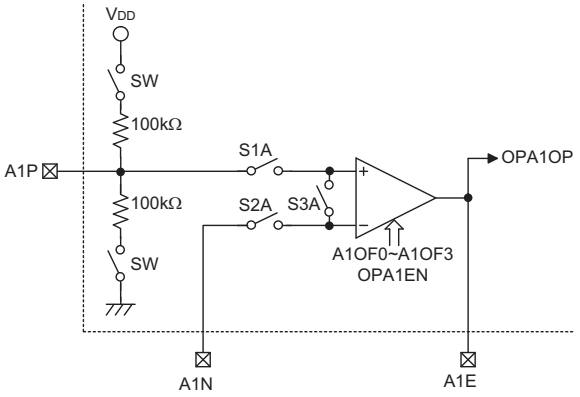
ARS1	A1OFM	S1A	S2A	S3A
0	0	ON	ON	OFF
0	1	OFF	ON	ON
1	0	ON	ON	OFF
1	1	ON	OFF	ON



有一个内置的偏压电阻用来为 OPAs 产生 1/2VDD 电压。当三组 OPAs 中的任何一组相应的 OPAEN=1，则该功能就被激活。如果掩膜选项选择了内部基准，则必须设置使能 OPAxEN=1，使三个 OPAs 被激活。

当配置文件中选定由外部电路来提供偏置电压时，两个内部的开关将被打开以允许 A1P 引脚处于浮空状态。用户可以使用外部电阻来得到想要的各个偏置电压值。

ROPA 是 100 kΩ电阻。VOPA+是 A1P 引脚上的电压。



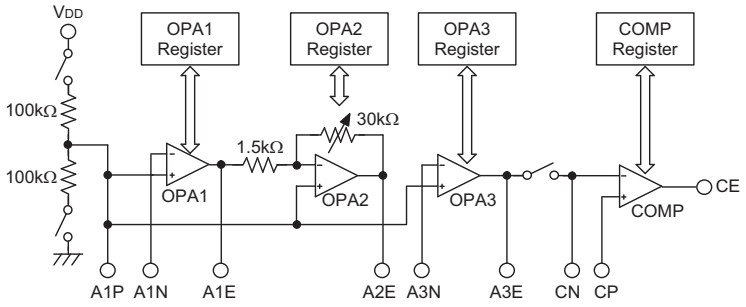
当满足条件（1）和（2）时，开关将关闭。

（1） 配置文件选择内部系统产生偏压。

（2） OPA1EN 或 OPA2EN 或 OPA3EN 被置位为 1。当满足条件（3）或（4）时，开关将打开。

（3） 配置文件选择内部系统产生偏压，但 OPA1EN、OPA2EN、OPA3EN 中任何一位均没被置位为 1。

（4） 配置文件选定偏置电压由外部提供。

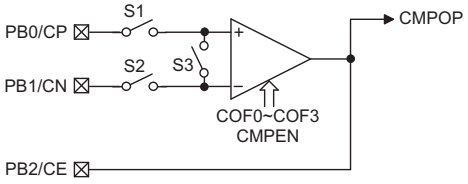


运算放大器方框图

比较器

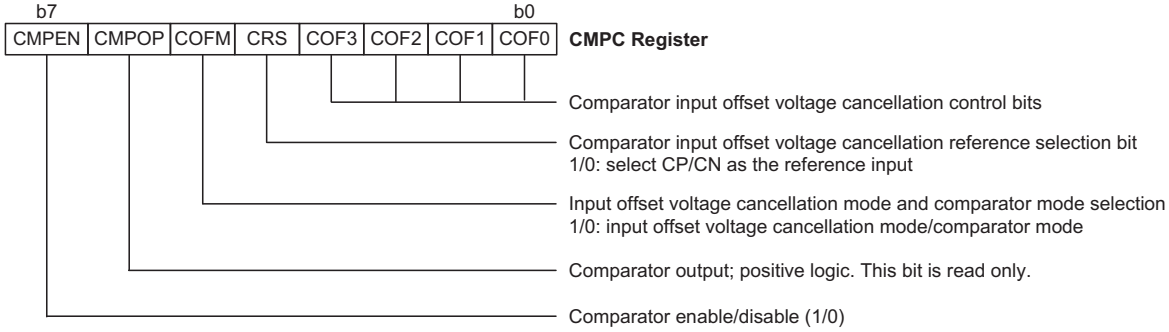
在芯片中有一组集成在 IC 中的比较器。其工作是由比较器的控制寄存器 CMPC 来控制。其中 CMPEN 是使能控制位，如果 CMPEN＝“0”，比较器除能，PB0/CP，PB1/CN，P2/CE 为普通 IO 口；反之 CMPEN ＝“1”那么比较器使能，PB0/CP，PB1/CN 为比较器的输入口，PB2/CE 为比较器的输出口。必须注意的是为了保证比较器能工作，寄存器 MISC 中的 COUTEN 位也必须置位为“1”。

CP，CN 和 CE 分别与 PB0，PB1 和 PB2 共享引脚。一旦比较器功能使能，内部与 PB0 和 PB1 相关的寄存器就不能使用，PB2 也仅能做为输入口使用，能读到 CE 的状态（如果 COUTEN 为“1”），要注意当 COUTEN 为“1”时 PB0 和 PB1 的输入/输出功能及 PB2 的输出功能及它们相应的上拉电阻都会自动失效。当 COUTEN 为“0”时，PB2 能做为普通的输入/输出口使用。比较器功能是否启动由软件指令来决定。



CRS	COFM	S1	S2	S3
0	0	ON	ON	OFF
0	1	OFF	ON	ON
1	0	ON	ON	OFF
1	1	ON	OFF	ON

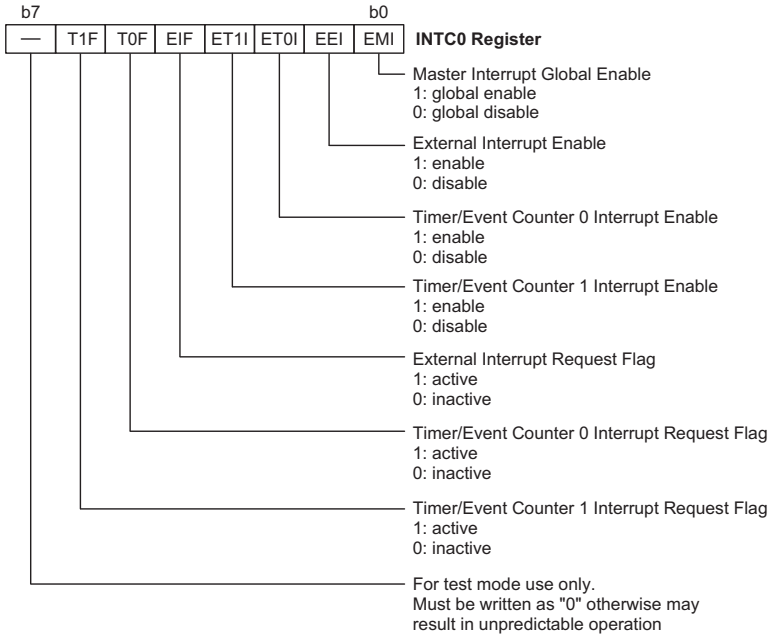
CMPEN	COUTEN	PB0，PB1，PB2
0	X	PB0，PB1，PB2 是普通输入/输出口
1	0	PB0,PB1 是比较器输入口，PB2 普通输入/输出口.
1	1	PB0，PB1 是比较器输入口，PB2 是比较器输出口。 PB2 可以读出当前比较的输出状态。



CMPC 寄存器

中断

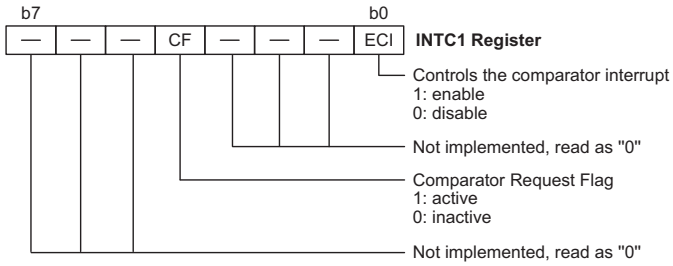
中断是任一系列单片机系统的一个重要部分。当外部事件或内部功能如定时/计数器、比较器向微控制器发出请求时，系统会强制中止当前的主程序，而转去执行相应的中断服务程序。此系列单片机均提供 1 个外部中断和几个内部中断，外部中断由外部引脚 INT 的动作控制，而内部中断由 2 个定时/计数器溢出，比较器中断控制。



INTC0 寄存器

中断寄存器

所有中断允许和请求标志均由数据存储器内 INTC0 和 INTC1 寄存器控制。通过控制相应的中断允许位使能或禁止相关的中断。当发生中断，相应有中断请求标志将被置位。总中断请求标志清零将禁止所有中断允许。



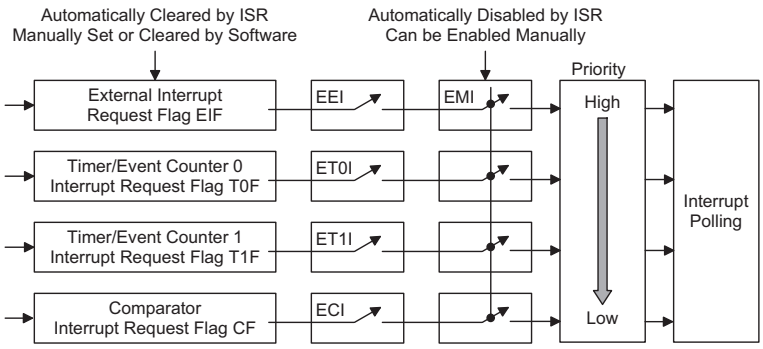
INTC1 寄存器

中断操作

定时/计数器计数溢出、外部中断输出口上有一个效的触发沿或比较器输出状态的变化都会置位相应的中断请求标志，如果相应的中断允许，那么下条原本将被执行指令的 PC 值将被压入堆栈，而相应的中断向量地址会加载至 PC 中，系统将从此向量取下条指令。中断向量处通常为跳转指令，以跳转到相应的中断服务程序。中断服务程序必须以 RETI 指令返回，取回原被压入堆栈中的 PC 值回到主程序，以执行中断发生以前程序执行的部分。

各个中断使能位以及相应的请求标志位，以优先级的顺序请见下文相应图档。

一旦中断子程序被响应，系统将自动清除 EMI 位，所有其它的中断将被屏蔽，这个方式可以防止任何进一步的中断嵌套。如果在此期间，其它中断请求发生，只有中断请求标志位会被记录。如果某个中断服务子程序正在执行，此时有另一个中断要求响应，EMI 位可以被置位，以允许此中断被响应。如果堆栈已满，即使此中断使能，中断请求也不会被响应，直到 SP 减少为止。如果要求立即执行子程序，则堆栈必须避免成为储满状态。



中断结构

中断优先级

当中断发生在两个连续的 T2 脉冲上升沿之间时，如果相应的中断请求被允许，中断将在后一个 T2 脉冲响应。下表指出同时提出请求的情况下所提供的优先级，可以通过重新设定 EMI 位来加以屏蔽。

中断源	优先值
外部中断	1
定时/计数器 0 溢出中断	2
定时/计数器 1 溢出中断	3
比较器中断	4

假使外部和内部定时中断均被使能，若外部和内部定时中断同时发生，则外部中断永远优先处理，首先被响应。使用控制寄存器 INTC0，INTC1 适当地屏蔽个别中断，可以防止同时发生的情况。

外部中断

若要产生一个外部中断发生，则总中断控制位 EMI、外部中断使能位 EEI 必须先被置位，一旦 EIF 请求标志位置 1 时，外部中断即会发生。当 INT 引脚上电平产生一个状态变化时，不管是上升沿还是下降沿（由 MISC 寄存器的第 6 位和第 7 位来决定），EIF 就被置位。同样 MISC 的这两位也能除能外部中断功能。

INTES1	INTES0	触发沿类型
0	0	外部中断除能
0	1	上升沿触发
1	0	下降沿触发
1	1	双沿触发（上升/下降沿都可以）

外部中断输入脚是与 PA2 共享，当外部中断允许且 MISC 寄存器中的有效触发沿已设定好后，该引脚就会被配置为外部中断输入脚，此时请保证一定要通过对 PAC.1 进行操作确保其为输入状态。当中断使能、堆栈未满且 INT 引脚上产生一个有效的触发沿时，将调用位于地址 04H 处的子程序。当响应外部中断服务子程序时，外部中断请求标志位 EIF 会被复位且 EMI 位会被清零以除能其它中断。注意的是即使这些引脚作为外部中断输入时，引脚的上拉电阻仍然有效。

定时/计数器中断

若使定时/计数器中断发生，则总中断控制位 EMI、相应的计数中断使能位 ET0I 或 ET1I 必须先被置位。当定时/计数器发生溢出，相应的中断请求标志位 T0F 或 T1F 将置位并触发定时/计数器中断。若中断使能，堆栈未满，当发生定时/计数器发生溢出时，若是定时/计数器 0，将调用位于地址 08H 处的子程序或若是定时/计数器 1，将调用位于地址 0CH 处的子程序。当响应定时/计数器中断服务子程序时，中断请

求标志位 T0F 或 T1F 会被复位且 EMI 位会被清零以除能其它中断。

比较器中断

若要允许模拟比较器中断发生，则总中断控制位 EMI、及相应的中断使能位 ECI 必须先被置位。当比较器的输出端发生变化时，相应的中断请求标志位 CF 将置位。若中断使能，堆栈未满，CF 一旦被置位（比较器的输出端发生变化），将触发模拟比较器中断，调用位于地址 10H 处的子程序。当响应模拟比较器中断服务子程序时，中断请求标志位 CF 会被复位且 EMI 位会被清零以除能其它中断。

编程注意事项

通过禁止中断使能位，可以屏蔽中断请求，然而，一旦中断请求标志位被置位，它们会被保留在中断控制寄存器 INTC0，INTC1 内，直到相应的中断服务子程序执行或被软件指令清除。

建议在中断服务子程序中不要使用“CALL 子程序”指令。中断通常发生在不可预料的情况或是需要立刻执行的某些应用。假如只剩下一层堆栈且没有控制好中断，当“CALL 子程序”在中断服务子程序中执行时，将破坏原来的控制序列。

所有中断都具有唤醒功能。当进入中断服务程序，系统会将程序计数器的内容压入堆栈。如果有影响主程序流程的寄存器或状态寄存器被中断服务程序改变，应事先将这些数据保存起来。

复位和初始化

复位功能是任何单片机中基本的部分，使得单片机可以设定一些与外部参数无关的先置条件。最重要的复位条件是在初次提供电源给单片机后，经短暂延迟，内部电路将使得单片机被定义在良好的状态且准备执行第一条程序语句。上电复位之后，在程序未开始执行前，部分重要的内部寄存器将会被预先设定状态。程序计数器就是其中之一，它会被清除为零，使得单片机从最低的程序存储器地址开始执行程序。

除了上电复位外，即使单片机正在执行状态，有些情况的发生也迫使单片机必须加以复位。其中一个例子是当提供电源给单片机以执行程序后，$\overline{\text{RES}}$引脚被强制拉至低电平。这个例子为正常操作复位，单片机中只有一些寄存器受影响，而大部分寄存器不受影响，以便复位引脚回复至高电平后，单片机仍可以正常操作。复位的另一种形式是看门狗定时器溢出而复位单片机，所有复位操作类型导致不同的寄存器条件被加以设定。

另外一种复位为低电压复位，即 LVR 复位，在电源提供电压低于某一临界值的情况下，一种和$\overline{\text{RES}}$引脚复位类似的完全复位将会被执行。

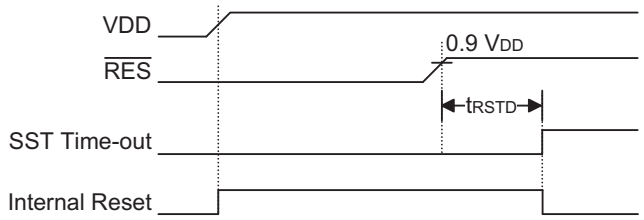
复位功能

通过内部与外部事件触发复位，单片机共有五种复位方式：

- 上电复位

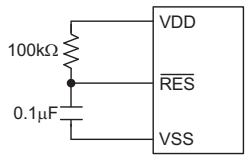
这是最基本且不可避免的复位，发生在单片机上电后。除了保证程序存储器会从起始地址开始执行，上电复位也使得其它寄存器被设定在预设条件，所有的输入/输出端口寄存器和输入/输出端口控制寄存器在上电复位时会保持高电平，以确保上电后所有引脚被设为输入状态。

虽然单片机有一个内部 RC 复位功能，如果 VDD 电源上升缓慢或刚上电时电源不稳定，内部 RC 振荡可能会导致芯片复位不良，所以推荐使用和$\overline{\text{RES}}$引脚连接的外部 RC 电路。由 RC 电路所造成的时间延迟使得$\overline{\text{RES}}$引脚在电源供应稳定前的一段延长周期内保持在低电平。在这段时间内，单片机的正常操作是被禁止的。$\overline{\text{RES}}$引脚达到一定电压值后，再经过延迟时间 tRSTD，单片机可开始进行正常操作。下图中 SST 是系统延迟周期 System Start-up Timer 的缩写。



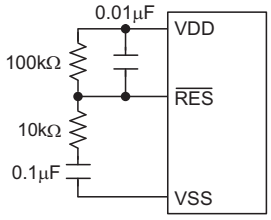
上电复位时序图

在许多应用场合，可以在 VDD 与$\overline{\text{RES}}$之间连接电阻，而在$\overline{\text{RES}}$与 VSS 之间连接电容作为复位电路，为了减少干扰影响，至$\overline{\text{RES}}$引脚的连线应尽量短。



基本复位电路

当系统在较强干扰的场合工作时，强烈建议使用增强型复位电路。

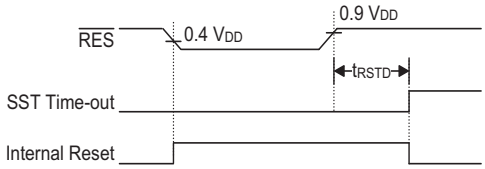


增强型复位电路

欲知更多外部复位电路的相关信息可参考 HOLTEK 网站应用范例 HA0075S。

- **RES**引脚复位

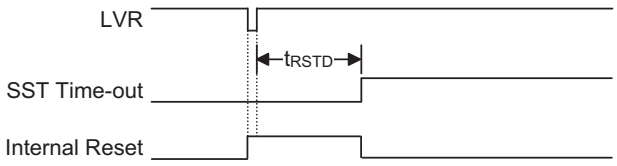
当单片机正常工作，RES引脚通过外部硬件(如外部开关)强迫拉至低电平时，此种复位形式即会发生。这种复位形式与其它复位一样，程序计数器会被清除为零且程序从头开始执行。



RES引脚复位时序图

- 低电压复位

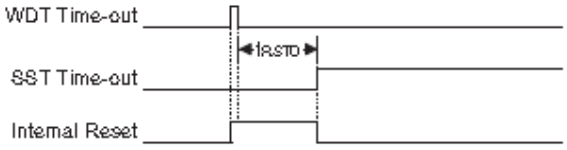
单片机具有低电压复位电路，用来监测它的电源电压，可通过掩膜选项进行选择。例如在更换电池的情况下，单片机供应的电压可能会落在 0.9V~V_{LVR} 的范围内，这时 LVR 将会自动复位单片机。LVR 包含以下的规格：有效的 LVR 信号，一个低电压，即在 0.9V~V_{LVR} 的低电压状态的时间，必须超过交流电气特性中 t_{LVR} 参数的值。如果低电压存在的状态没有超过 t_{LVR} 参数的值，则 LVR 将会忽略它且不会执行复位功能。V_{LVR} 的实际值可以通过掩膜选项选择支持的电压范围。



低电压复位时序图

- 正常操作时看门狗溢出复位

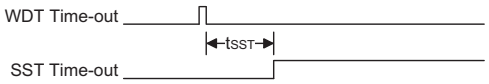
除了看门狗溢出标志位 TO 将被设为 1 之外，正常操作时看门狗溢出复位和 $\overline{RES}$ 复位相同。



正常操作时看门狗溢出复位时序图

- 暂停时看门狗溢出复位

暂停时看门狗溢出复位和其它种类的复位有些不同，除了程序计数器与堆栈指针将被清 0 及 TO 位被设为 1 外，绝大部份的条件保持不变。图中 tSST 的详细说明请参考交流电气特性。



暂停时看门狗溢出复位时序图

复位初始状态

不同的复位形式以不同的途径影响复位标志位。这些标志位，即 PDF 和 TO，存放在状态寄存器中，由暂停功能或看门狗计数器等几种控制器操作控制。复位标志位如下所示：

TO	PDF	复位条件
0	0	上电时的 $\overline{RES}$ 复位
u	u	正常运行时的 $\overline{RES}$ 复位或 LVR 复位
1	u	正常运行时的 WDT 溢出复位
1	1	HALT 暂停时的 WDT 溢出复位

注：“u”代表不改变

在单片机上电复位之后，各功能单元初始化的情形，列于下表。

项 目	复位后情况
程序计数器	清除为零
中断	所有中断被禁止
看门狗定时器	WDT 清除并重新计时
定时/计数器	定时/计数器停止
预分频器	定时/计数器之预分频器内容清除
输入/输出口	所有 I/O 设为输入模式
堆栈指针	堆栈指针指向堆栈顶端

不同的复位形式以不同的途径影响单片机中的内部寄存器。为保证复位发生后程序的正常执行，在特定的复位发生后，了解单片机内的情况是非常重要的。下表描述了不同的复位如何影响单片机的内部寄存器。

寄存器	复位 (上电复位)	WDT 溢出 (正常运行)	$\overline{RES}$复位 (正常运行)	$\overline{RES}$复位 (暂停模式)	WDT 溢出 (暂停模式)*
PC	000H	000H	000H	000H	000H
MP	1xxx xxxx	1uuu uuuu	1uuu uuuu	1uuu uuuu	1uuu uuuu
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	--xx xxxx	--uu uuuu	--uu uuuu	--uu uuuu	--uu uuuu
WDTS	0000 0111	0000 0111	0000 0111	0000 0111	uuuu uuuu
STATUS	--00 xxxx	--1u uuuu	--uu uuuu	--01 uuuu	--11 uuuu
INTC0	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu
INTC1	---0 ---0	---0 ---0	---0 ---0	---0 ---0	---u ---u
TMR0	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR0C	00-0 0100	00-0 0100	00-0 0100	00-0 0100	uu-u uuuu
TMR1H	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR1L	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR1C	00-0 1---	00-0 1---	00-0 1---	00-0 1---	uu-u u---
PA	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PAC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
MISC	1000 0000	1000 0000	1000 0000	1000 0000	uuuu uuuu
OPAC1	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
OPAC2	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
OPAC3	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
CMPC	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu

注： “*”表示“热复位”

“u”表示不变化

“×”表示不确定

“—”表示未定义

振荡器

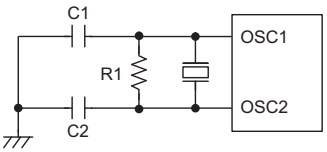
不同的振荡器选择可以让使用者在不同的应用需求中获得更多范围的功能。有 4 种系统时钟可供选择，而看门狗定时器又有多种时钟源选项，提供了使用者最大的灵活性。所有的振荡器选项都通过掩膜选项来完成。

系统时钟配置

有 2 种方法产生系统时钟，使用外部晶体/陶瓷振荡器、内部 RC 电路，可以通过掩膜选项来选择其中一种。如果选择频率在 10MHz 左右的晶体/陶瓷振荡器，那么掩膜选项必须被选择。

系统晶体/陶瓷振荡器

对于晶体振荡器的结构配置，只要简单地将晶体连接至 OSC1 和 OSC2，则会产生所需的相移及反馈，而不需其它外部的器件。对于某些类型的晶体和频率，需要使用两个小电容，即 C1 和 C2 。当使用陶瓷共振器，通常要连接 2 个小阻值的电容，C1 和 C2，来产生振荡。C1 和 C2 的具体数值与客户选择的晶体/陶瓷晶振有关。在许多应用场合，R1 并不一定需要，然而当 LVR 禁止，R1 的作用是在低电压的时候确保关闭振荡，即电源电压低于单片机的工作电压。



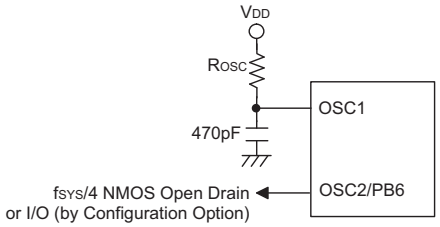
晶体/陶瓷振荡器

欲知更多关于系统振荡的相关信息可参考 HOLTEK 网站上应用范例 HA0075S。

外部系统 RC 振荡器

通过掩膜选项选定外部 RC 电路作为系统振荡器，需要在 OSC1 和 VDD 之间连接一个阻值在 24kΩ 到 1.5MΩ 之间的电阻，OSC1 与 GND 之间连接一个 470pF 的电容。在 OSC2/PB6 脚上生成 1/4 f_{SYS}，可以用于外部同步信号时钟。由于 OSC2 与 PB6 脚共用，因此必须首先在掩膜选项中选择所需要的功能。虽然此振荡器配置成本较低，但振荡频率会因 VDD、温度和芯片本身的制程而改变，因此不适合用来做严格计时或需要精确振荡器频率的场合。对于外部电阻 R_{OSC} 的阻值，请参考附录章节中典型 RC 振荡器对温度和 VDD 特性曲线。

注意的是，这是唯一的内部电路需要结合外部电阻才能决定系统频率的一种方式。右图中所示的外部电容对频率值并无影响，它的作用只是为了保证系统频率的稳定，尤其是在使用开漏结构的 OSC2 输出脚做为系统频率 4 分频输出应用时。



外部 RC 振荡器

看门狗振荡器

WDT 振荡器是单片机内部 RC 型振荡器，不需要任何外部元件。在 5V 工作系统下其振荡周期大约为 65μS。该功能可通过掩膜选项来选择，如果选定了，即使系统进入暂停模式，系统时钟停止，WDT 振荡器仍会运作，保持看门狗的功能。但是，在暂停模式时，若打开看门狗，将会增加一定的功耗。所以，在某些应用中为了节省电源，可在选项设定中关闭 WDT 振荡器。

暂停模式和唤醒

暂停模式

所有 HOLTEK 单片机都具有暂停功能，即 HALT 模式或者 Sleep 模式。当进入此模式，芯片的工作电流会降低到静态电流等级。这主要是因当系统进入此模式后，系统振荡器停止振荡，以降低功耗到一个最低水平。由于单片机保持了内部工作条件，它可以被唤醒并继续运行，而不需要重新进行复位。这个特性在单片机需要持续供电以保证其工作在已知状态，但电源容量有限的场合（如电池供电系统）特别重要。

进入低功耗模式

仅有一种方式能使芯片进入到省电模式，即程序执行“HALT”指令，一旦“HALT”指令被执行以下情况将会发生：

- 系统振荡器将被关闭，应用程序停止在“HALT”指令处
- 芯片内 RAM 和寄存器的内容保持不变
- 如果 WDT 时钟源是来自 WDT 振荡器，则 WDT 将被清除然后再重新计数；若来源于系统时钟，则停止计数
- 所有输入/输出端口状态保持不变
- 状态寄存器中 PDF 标志位被置位而 TO 标志位被清零

静态电流注意事项

将系统的电流消耗尽可能的降到最低，仅几毫安的情况，除了需要单片机进入 HALT 模式，还要考虑到电路的设计。特别注意输入/输出口的状态，所有高阻抗输入引脚必须接高电平或低电平，否则会造成引脚浮空而引起内部振荡，增加系统功耗。另外还需注意单片机输出端口上的负载，原则是尽量减小电路上的牵引电流或与其他不存在牵引电流的电路诸如 CMOS 输入相连。若选项设定使能内部看门狗振荡器，单片机进入暂停模式时 WDT 继续运作，将消耗部分额外的电流。在对功耗要求比较严格的应用中建议使用系统时钟做为 WDT 的时钟源。

唤醒

当系统进入 HALT 模式下，可以通过以下几种方式唤醒：

- 外部复位
- PA 口引脚下降沿
- 系统中断
- WDT 溢出

若由外部 RES 引脚唤醒，系统会经历完全复位的过程。若由 WDT 溢出唤醒，则看门狗计数器将被清除。这两种唤醒方式都会使系统复位，可以通过状态寄存器中 TO 和 PDF 位来判断它的唤醒源。系统上电或执行清除看门狗的指令，PDF 被清除；执行 HALT 指令，PDF 将置位。看门狗计数器溢出将会置位 TO 标志并唤醒系统，同时复位程序计数器和堆栈指针，其它标志保持原有状态。

PA 口上的每一个引脚可以通过选项设定独立的下降沿唤醒功能。当一个 PA 口引脚唤醒后，程序将在“HALT”的下一条指令处继续执行。

如果系统是通过中断唤醒，则有两种可能发生，假如中断禁止或中断使能但堆栈已满，程序将在“HALT”指令下一条处继续执行，此时中断只唤醒单片机，但相应的中断服务程序只有在中断使能或堆栈空闲后被执行；假如中断使能且堆栈未满，则正常的中断响应将会发生。假设在进入暂停模式之前某个中断请求标志位被设为“1”，相关中断的唤醒功能将无效。

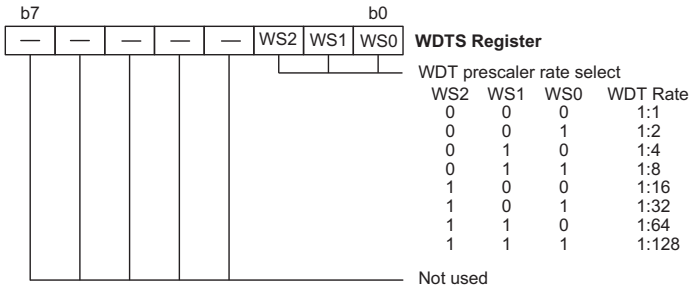
不管唤醒功能由谁触发，一旦唤醒事件发生，回到正常运行将需要 1024 个系统时钟周期。如果唤醒由中断发生，则真实的中断子程序执行将延迟一个或数个周期。如果唤醒后接着去执行“HALT”下一条指令，则它将在 1024 个系统时钟周期结束后立刻执行。

看门狗计数器

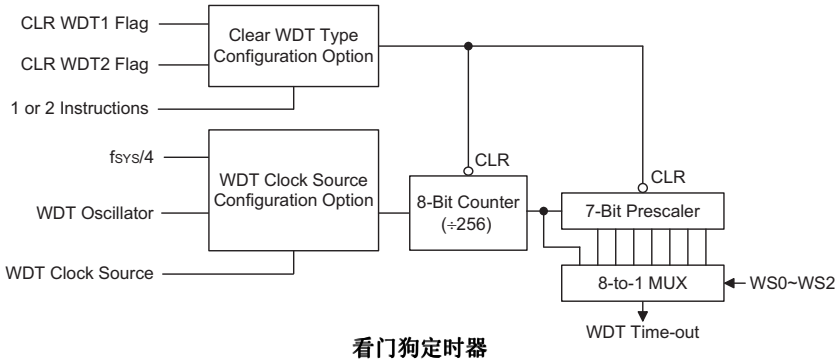
看门狗定时器 WDT 的功能在于防止诸如电燥声等干扰因素造成的程序不正常动作或跳转到未知的地址空间。当 WDT 溢出时，它产生一个芯片复位的动作。注意的是当 WDT 选项设定为禁止时，任何相关的指令将无效。WDT 定时器不同时钟源，可通过选项设定进行选择。

该系统中 WDT 的振荡源有两种，一个完全内置的 RC 内部振荡器或 $f_{SYS}/4$（可通过配置文件选项设定）。WDT 内部振荡器在供应电压为 5V 时周期大约为 65μs, 这个周期可以随着 VDD、温度和制作工艺而改变。如果使用 $f_{SYS}/4$ 作为时钟源，需要注意的是系统进入暂停模式后，指令时钟将会停止，WDT 将失去其保护目的。当系统操作在干扰严重的环境时，强烈建议使用内部 WDT 振荡器。不管哪一种振荡源被选择，都可以通过一个 8 位的计数器来进行除频最大到 256 倍而且仍可以通过使用 7 位预分频器得到较长的溢出周期。写值到 WDTS 寄存器中第 0~2 位，即 WS0~WS2，将会得到更长的溢出周期。如果将 WS0~WS2 都设置为逻辑高，此时分频系数为 1:128，将得到最长溢出时间约为 2.1S。

系统在正常运行状态下，WDT 溢出将导致芯片复位，且置位 TO 状态标志位。然而，如果系统处于暂停模式，WDT 溢出复位将使系统复位，置位 TO 状态标志位并复位程序计数器和堆栈指针。有三种方法可以用来清除 WDT 的内容。第一种是通过外部硬件复位，即外部硬件复位即 $\overline{RES}$ 引脚低电平，第二种是软件清 WDT 指令，第三种是运行 HALT 指令。通过软件指令有两种方法清除看门狗寄存器，由选项设定进行选择。选择使用“CLR WDT”指令或使用“CLR WDT1”和“CLR WDT2”两条指令。对于第一种选择，只要执行“CLR WDT”便清除 WDT。而第二种选择，必须交替执行“CLR WDT1”和“CLR WDT2”才能成功清除 WDT。当选择第二种方法，如果“CLR WDT1”用来清除 WDT，接着再执行这条指令是无效的，只有执行“CLR WDT2”指令才能清除 WDT。同样地执行“CLR WDT2”指令后，只有接着执行“CLR WDT1”指令才可以清除 WDT。



看门狗定时器寄存器



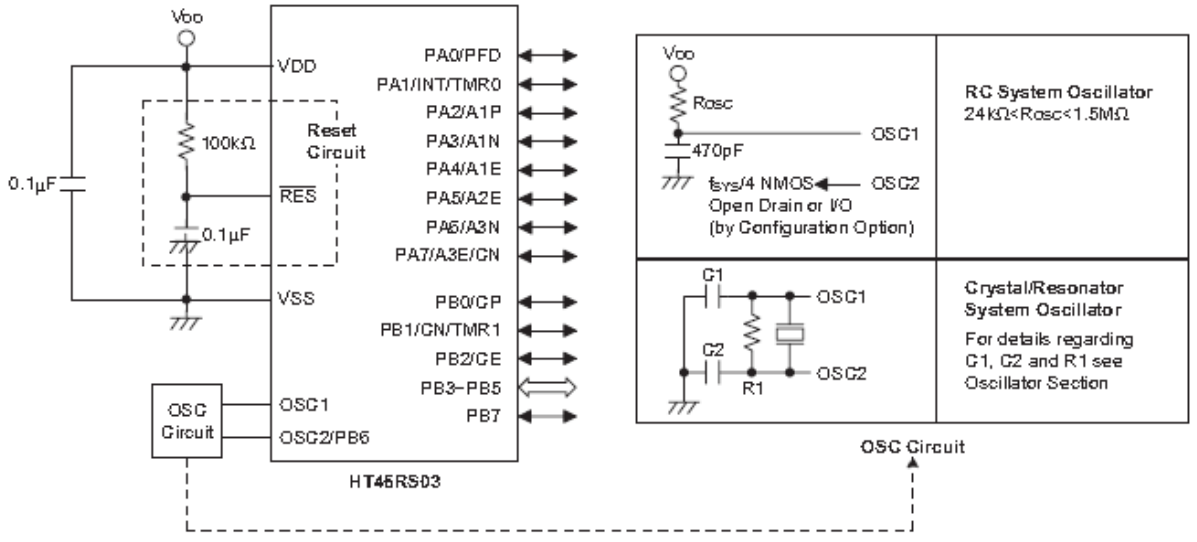
看门狗定时器

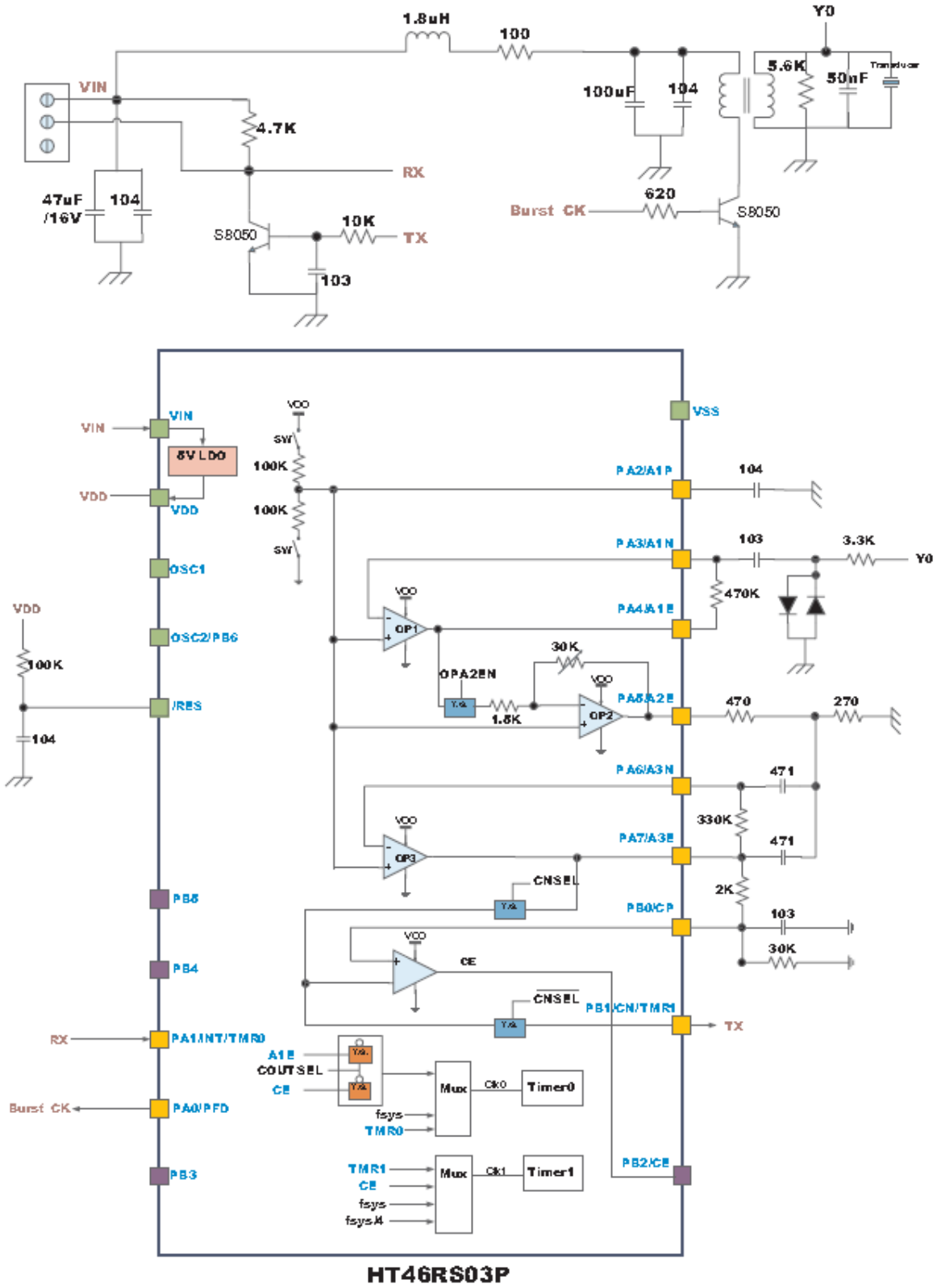
掩膜选项

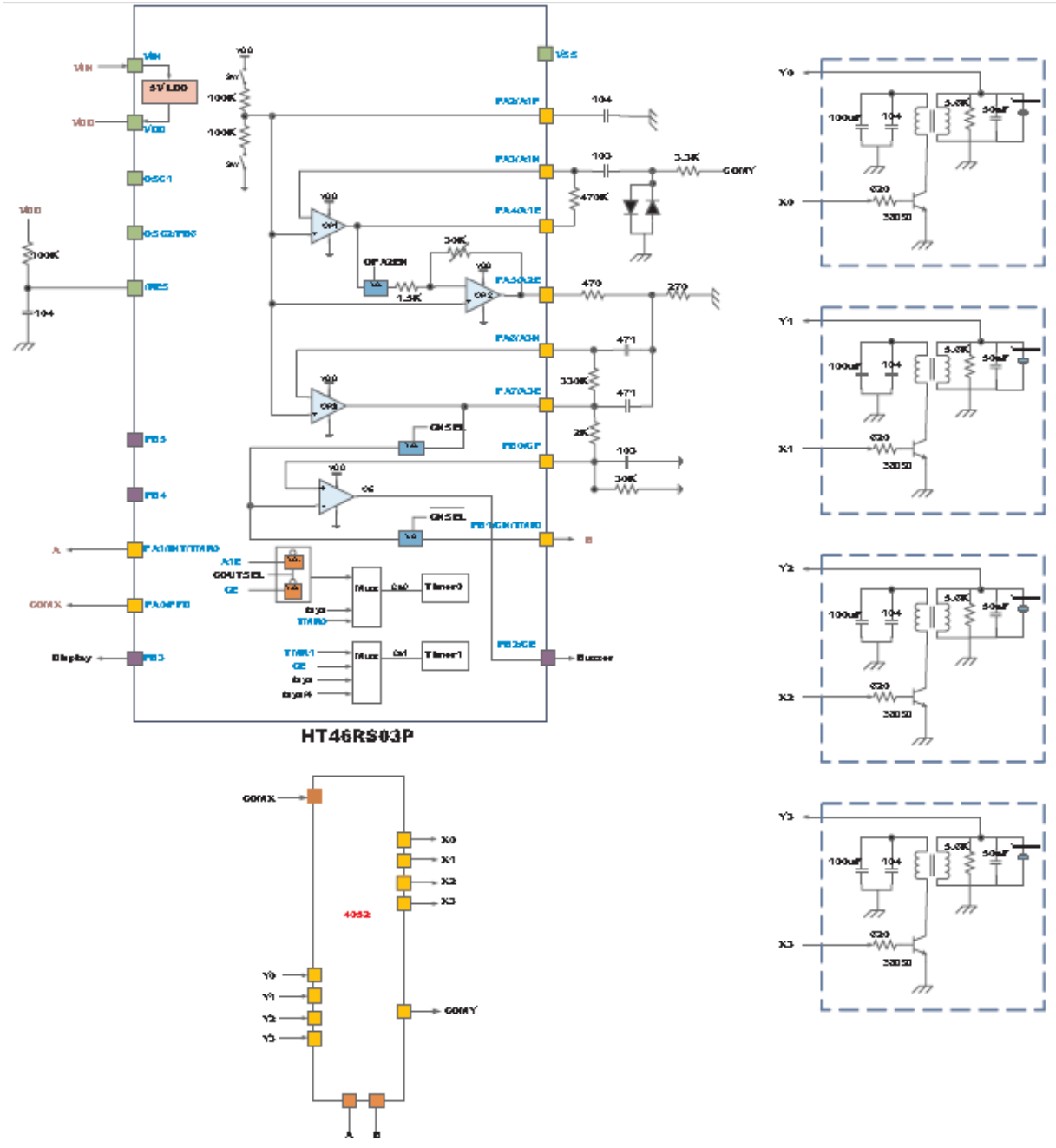
掩膜选项在烧写程序时写入芯片。通过 HT-IDE 的软件开发环境，使用者在开发过程中可以选择掩膜选项。当掩膜选项烧入单片机后，无法再通过应用程序修改。所有位必须按系统的需要定义，具体内容可参考下表：

编号	选项
输入/输出口选项	
1	PA7~PA0 唤醒：使能或除能（位选择）
2	PA 口上拉电阻使能或除能（位选择）
3	PB 口上拉电阻使能或除能（位选择）
4	PA0：普通输入/输出或 PFD 输出
振荡器选项	
5	OSC 类型选择：RC+OSC2（f_{sys}/4）、RC+I/O（PA6）或晶体振荡
6	XTAL 高或低：（>10MHz）或（<10MHz）
PFD 选项	
7	PFD：来自计数器 0 或计数器 1
看门狗选项	
8	WDT 计数：使能或除能
9	WDT 时钟源：WDT 内部振荡器，f_{SYS}/4
10	CLRWDT 指令：1 或 2 条
运算放大器偏置电压选项	
11	偏置电压控制：外部或内部
低电压复位选项	
12	LVR 功能：使能或除能
13	LVR 电压：2.1V、3.15V 或 4.2V
锁定选项	
14	锁住全部
15	局部锁

应用电路




1.8uH
100
Y0
VIN
4.7K
RX
100uF
104
5.6K
50nF
47uF
/16V
104
S8050
10K
TX
103
620
Burst CK
S8050
HT46RS03P
VIN
5V LDO
VDD
OSC1
OSC2/PB6
/RES
PB5
PB4
PA1/INT/TMR0
PA0/PFD
PB3
VSS
PA2/A1P
PA3/A1N
PA4/A1E
PA5/A2E
PA6/A3N
PA7/A3E
PB0/CP
PB1/CN/TMR1
PB2/CE
SW
100K
OP1
OP2
OP3
OPA2EN
1.5K
30K
470K
3.3K
470
270
471
330K
2K
CNSEL
CE
A1E
COUTSEL
Mux
Ck0
Timer0
fsys
TMR0
Ck1
Timer1
fsys/4


HT46RS03P
4052
COMX
COMY
Y0
Y1
Y2
Y3
X0
X1
X2
X3
A
B

指令集介绍

指令集

任何单片机成功运作的核心在于它的指令集，此指令集为一组程序指令码，用来指导单片机如何去执行指定的工作。在盛群单片机中，提供了丰富且易变通的指令，共超过六十条，程序设计师可以事半功倍地实现他们的应用。

为了更加容易了解各式各样的指令码，接下来按功能分组介绍它们。

指令周期

大部分的操作均只需要一个指令周期来执行。分支、调用或查表则需要两个指令周期。一个指令周期相当于四个系统时钟周期，因此如果在 8MHz 的系统时钟振荡器下，大部分的操作将在 0.5μs 中执行完成，而分支或调用操作则将在 1μs 中执行完成。虽然需要两个指令周期的指令通常指的是 JMP、CALL、RET、RETI 和查表指令，但如果牵涉到程序计数器低字节寄存器 PCL 也将多花费一个周期去加以执行。即指令改变 PCL 的内容进而导致直接跳转至新地址时，需要多一个周期去执行。例如“CLR PCL”或“MOV PCL, A”。对于跳转命令必须注意的是，如果比较的结果牵涉到跳转动作将多花费一个周期，如果没有则需一个周期即可。

数据的传送

单片机程序中数据传送是使用最为频繁的操作之一，使用三种 MOV 的指令，数据不但可以从寄存器转移至累加器(反之亦然)，而且能够直接移动立即数到累加器。数据传送最重要的应用之一是从接收端口接收数据或者传送数据到输出端口。

算术运算

算术运算和数据处理是大部分单片机应用所需具备的能力，在盛群单片机内部的指令集中，可直接实现加与减的运算。当加法的结果超出 255 或减法的结果少于 0 时，要注意正确的处理进位和借位的问题。INC、INCA、DEC 和 DECA 指令提供了对一个指定地址的值加一或减一的功能。

逻辑和移位运算

标准逻辑运算例如 AND、OR、XOR 和 CPL 全都包含在盛群单片机内部的指令集中。大多数牵涉到数据运算的指令，数据的传送必须通过累加器。在所有逻辑数据运算中，如果运算结果为零，则零标志位将被置位，另外逻辑数据运用形式还有移位指令，例如 RR、RL、RRC 和 RLC 提供了向左或向右移动一位的方法。移位指令常用于串行端口的程序应用，数据可从内部寄存器转移至进位标志位，而此位则可被检验，移位运算还可应用在乘法与除法的运算组成中。

分支和控制的转换

程序分支是采取使用 JMP 指令跳转至指定地址或使用 CALL 指令调用子程序的形式，两者之不同在于当子程序被执行完毕后，程序必须马上返回原来的地址。这个动作是由放置在子程序里的返回指令 RET 来实现，它可使程序跳回 CALL 指令之后的地址。在 JMP 指令中，程序则只是跳到一个指定的地址而已，并不需如 CALL 指令般跳回。一个非常有用的分支指令是条件跳转，跳转条件是由数据存储器或指定位来加以决定。遵循跳转条件，程序将继续执行下一条指令或略过且跳转至接下来的指令。这些分支指令是程序走向的关键，跳转条件可能是外部开关输入，或者是内部数据位的值。

位运算

供数据存储器中单个位的运算指令是盛群单片机的特性之一。这特性对于输出端口位的规划尤其有用，其中个别的位或端口的引脚可以使用“SET [m].i”或“CLR [m].i”指令来设定其为高位或低位。如果没有这特性，程序设计师必须先读入输出口的 8 位数据，处理这些数据，然后再输出正确的新数据。这种读入-修改-写出的过程现在则被位运算指令所取代。

查表运算

数据的储存通常由寄存器完成，然而当处理大量固定的数据时，它的存储量常常造成对个别存储器的不便。为了改善此问题，盛群单片机允许在程序存储器中设定一块数据可直接存储的区域，只需要一组简易的指令即可对数据进行查表。

其它运算

除了上述功能指令外，其它指令还包括用于省电的“HALT”指令和使程序在极端电压或电磁环境下仍能正常工作的看门狗定时器控制指令。这些指令的使用则请查阅相关的章节。

指令集摘要

下列表格是按照指令功能来分类描述的，可作为基本指令的参考，其使用了如下惯例。

表格惯例：

x：立即数

m：数据存储器地址

A：累加器

I：0~7 号位

Addr：程序存储器地址

指令集摘要

助记符		说明	指令周期	影响标志位
算术运算				
ADD	A,[m]	ACC 与数据存储器相加，结果放入 ACC	1	Z,C,AC,OV
ADDM	A,[m]	ACC 与数据存储器相加，结果放入数据存储器	1(1)	Z,C,AC,OV
ADD	A,x	ACC 与立即数相加，结果放入 ACC	1	Z,C,AC,OV
ADC	A,[m]	ACC 与数据存储器、进位标志相加，结果放入 ACC	1	Z,C,AC,OV
ADCM	A,[m]	ACC 与数据存储器、进位标志相加，结果放入数据存储器	1(1)	Z,C,AC,OV
SUB	A,x	ACC 与立即数相减，结果放入 ACC	1	Z,C,AC,OV
SUB	A,[m]	ACC 与数据存储器相减，结果放入 ACC	1	Z,C,AC,OV
SUBM	A,[m]	ACC 与数据存储器相减，结果放入数据存储器	1(1)	Z,C,AC,OV
SBC	A,[m]	ACC 与数据存储器、进位标志相减，结果放入 ACC	1	Z,C,AC,OV
SBCM	A,[m]	ACC 与数据存储器、进位标志相减，结果放入数据存储器	1(1)	Z,C,AC,OV
DAA	[m]	将加法运算中放入 ACC 的值调整为十进制数，并将结果放入数据存储器	1(1)	C
逻辑运算				
AND	A,[m]	ACC 与数据存储器做“与”运算，结果放入 ACC	1	Z
OR	A,[m]	ACC 与数据存储器做“或”运算，结果放入 ACC	1	Z
XOR	A,[m]	ACC 与数据存储器做“异或”运算，结果放入 ACC	1	Z
ANDM	A,[m]	ACC 与数据存储器做“与”运算，结果放入数据存储器	1(1)	Z
ORM	A,[m]	ACC 与数据存储器做“或”运算，结果放入数据存储器	1(1)	Z
XORM	A,[m]	ACC 与数据存储器做“异或”运算，结果放入数据存储器	1(1)	Z
AND	A,x	ACC 与立即数做“与”运算，结果放入 ACC	1	Z
OR	A,x	ACC 与立即数做“或”运算，结果放入 ACC	1	Z
XOR	A,x	ACC 与立即数做“异或”运算，结果放入 ACC	1	Z
CPL	[m]	对数据存储器取反，结果放入数据存储器	1(1)	Z
CPLA	[m]	对数据存储器取反，结果放入 ACC	1	Z
递增和递减				
INCA	[m]	递增数据存储器，结果放入 ACC	1	Z
INC	[m]	递增数据存储器，结果放入数据存储器	1(1)	Z
DECA	[m]	递减数据存储器，结果放入 ACC	1	Z
DEC	[m]	递减数据存储器，结果放入数据存储器	1(1)	Z
移位				
RRA	[m]	数据存储器右移一位，结果放入 ACC	1	无
RR	[m]	数据存储器右移一位，结果放入数据存储器	1(1)	无
RRCA	[m]	带进位将数据存储器右移一位，结果放入 ACC	1	C
RRC	[m]	带进位将数据存储器右移一位，结果放入数据存储器	1(1)	C
RLA	[m]	数据存储器左移一位，结果放入 ACC	1	无
RL	[m]	数据存储器左移一位，结果放入数据存储器	1(1)	无
RLCA	[m]	带进位将数据存储器左移一位，结果放入 ACC	1	C
RLC	[m]	带进位将数据存储器左移一位，结果放入数据存储器	1(1)	C
数据传送				
MOV	A,[m]	将数据存储器送至 ACC	1	无
MOV	[m],A	将 ACC 送至数据存储器	1(1)	无
MOV	A,x	将立即数送至 ACC	1	无
位运算				
CLR	[m].i	清除数据存储器的位	1(1)	无
SET	[m].i	置位数据存储器的位	1(1)	无

助记符		说明	指令周期	影响标志位
转移				
JMP	addr	无条件跳转	2	无
SZ	[m]	如果数据存储器为零，则跳过下一条指令	1(2)	无
SZA	[m]	数据存储器送至 ACC，如果内容为零，则跳过下一条指令	1(2)	无
SZ	[m].i	如果数据存储器的第 i 位为零，则跳过下一条指令	1(2)	无
SNZ	[m].i	如果数据存储器的第 i 位不为零，则跳过下一条指令	1(2)	无
SIZ	[m]	递增数据存储器，如果结果为零，则跳过下一条指令	1(3)	无
SDZ	[m]	递减数据存储器，如果结果为零，则跳过下一条指令	1(3)	无
SIZA	[m]	递增数据存储器，将结果放入 ACC，如果结果为零，则跳过下一条指令	1(2)	无
SDZA	[m]	递减数据存储器，将结果放入 ACC，如果结果为零，则跳过下一条指令	1(2)	无
CALL	addr	子程序调用	2	无
RET		从子程序返回	2	无
RET	A,x	从子程序返回，并将立即数放入 ACC	2	无
RETI		从中断返回	2	无
查表				
TABRDC	[m]	读取当前页的 ROM 内容，并送至数据存储器和 TBLH	2(1)	无
TABRDL	[m]	读取最后页的 ROM 内容，并送至数据存储器和 TBLH	2(1)	无
其它指令				
NOP		空指令	1	无
CLR	[m]	清除数据存储器	1(1)	无
SET	[m]	置位数据存储器	1(1)	无
CLR	WDT	清除看门狗定时器	1	TO,PDF
CLR	WDT1	预清除看门狗定时器	1	TO(4),PDF(4)
CLR	WDT2	预清除看门狗定时器	1	TO(4),PDF(4)
SWAP	[m]	交换数据存储器的高低字节，结果放入数据存储器	1(1)	无
SWAPA	[m]	交换数据存储器的高低字节，结果放入 ACC	1	无
HALT		进入暂停模式	1	TO,PDF

注： x：立即数

m：数据存储器地址

A：累加器

i：第 0~7 位

addr：程序存储器地址

√：影响标志位

—：不影响标志位

(1)：如果数据是加载到 PCL 寄存器，则指令执行周期会被延长一个指令周期(四个系统时钟)。

(2)：如果满足跳跃条件，则指令执行周期会被延长一个指令周期(四个系统时钟)；否则指令执行周期不会被延长。

(3)：(1)和(2)

(4)：如果执行 CLR WDT1 或 CLR WDT2 指令后，看门狗定时器被清除，则会影响 TO 和 PDF 标志位；否则不会影响 TO 和 PDF 标志位。

指令描述

ADC A, [m] 累加器与数据存储器、进位标志相加，结果放入累加器
说明： 本指令把累加器、数据存储器值以及进位标志相加，结果存放到累加器。
运算过程： ACC←ACC+[m]+C
影响标志位 OV，Z，AC，C

ADCM A, [m] 累加器与数据存储器、进位标志相加，结果放入数据存储器
说明： 本指令把累加器、数据存储器值以及进位标志相加，结果存放到存储器。
运算过程： [m]←ACC+[m]+C
影响标志位 OV，Z，AC，C

ADD A, [m] 累加器与数据存储器相加，结果放入累加器
说明： 本指令把累加器、数据存储器值相加，结果存放到累加器。
运算过程： ACC←ACC+[m]
影响标志位 OV，Z，AC，C

ADD A, x 累加器与立即数相加，结果放入累加器
说明： 本指令把累加器值和立即数相加，结果存放到累加器。
运算过程： ACC←ACC+x
影响标志位 OV，Z，AC，C

ADDM A, [m] 累加器与数据存储器相加，结果放入数据存储器
说明： 本指令把累加器、数据存储器值相加，结果放到数据存储器。
运算过程： [m]←ACC+[m]
影响标志位 OV，Z，AC，C

AND A, [m] 累加器与数据存储器做“与”运算，结果放入累加器
说明： 本指令把累加器值、数据存储器值做逻辑与，结果存放到累加器。
运算过程： ACC←ACC “AND” [m]
影响标志位 Z

AND A, x 累加器与立即数做“与”运算，结果放入累加器
说明： 本指令把累加器值、立即数做逻辑与，结果存放到累加器。
运算过程： ACC←ACC “AND” x
影响标志位 Z

ANDM A, [m] 累加器与数据存储器做“与”运算，结果放入数据存储器
说明： 本指令把累加器值、数据存储器值做逻辑与，结果放到数据存储器。
运算过程： [m]←ACC “AND” [m]
影响标志位 Z

CALL addr 子程序调用
说明： 本指令直接调用地址所在处的子程序，此时程序计数器加一，将此程序计数器值存到堆栈寄存器中，再将子程序所在处的地址存放到程序计数器中。
运算过程： Stack←PC+1
PC ← addr
影响标志位 没有

CLR [m] 清除数据存储器
说明： 本指令将数据存储器内的数值清零。
运算过程： [m] ←00H
影响标志位 没有

CLR [m] . i 将数据存储器的第 i 位清“0”
说明： 本指令将数据存储器内第 i 位值清零。
运算过程： [m].i ←0
影响标志位 没有

CLR WDT 清除看门狗定时器
说明： 本指令清除 WDT 计数器(从 0 开始重新计数)，暂停标志位(PDF)和看门狗溢出标志位(TO)也被清零。
运算过程： WDT ← 00H
PDF&TO ← 0
影响标志位 TO，PDF

CLR WDT1 预清除看门狗定时器
说明： 必须搭配 CLR WDT2 一起使用，才可清除 WDT 计时器(从 0 开始重新计数)。当程序只执行过该指令，没有执行 CLR WDT2 时，系统只会不会将暂停标志位(PDF)和计数溢出位(TO)清零，PDF 与 TO 保留原状态不变。
运算过程： WDT ← 00H
PDF&TO ← 0
影响标志位 TO，PDF

CLR WDT2 预清除看门狗定时器
说明： 必须搭配 CLR WDT1 一起使用，才可清除 WDT 计时器(从 0 开始重新计数)。当程序只执行过该指令，没有执行 CLR WDT1 时，系统只会不会将暂停标志位(PDF)和计数溢出位(TO)清零，PDF 与 TO 保留原状态不变。
运算过程： WDT ← 00H
PDF&TO ← 0
影响标志位 TO，PDF

CPL [m] 对数据存储器取反，结果放入数据存储器
说明： 本指令是将数据存储器内保存的数值取反。
运算过程： [m] ← [$\overline{m}$]
影响标志位 Z

CPLA [m] 对数据存储器取反，结果放入累加器
说明： 本指令是将数据存储器内保存的值取反后，结果存放在累加器中。
运算过程： ACC ← $[\overline{m}]$
影响标志位 Z

DAA [m] 将加法运算后放入累加器的值调整为十进制数，并将结果放入数据存储器
说明 本指令将累加器高低四位分别调整为 BCD 码。如果低四位的值大于"9"或 AC=1，那么 BCD 调整就执行对原值加"6"；否则原值保持不变。如果高四位的值大于"9"或 C=1，那么 BCD 调整就执行对原值加"6"。本质上，就是基于累加器的值和标志位的基础上，通过对 BCD 加 00H，06H，60H 或 66H 进行十进制调整。只有进位标志位(C)受该指令的影响，它标志原 BCD 码之和是否大于 100，它保证了通过十进制数进行乘法计算的精确度。
操作 [m] ←(ACC+00H)或
[m] ←(ACC+06H)或
[m] ←(ACC+60H)或
[m] ←(ACC+66H)或
影响标志位 C

DEC [m] 数据存储器的内容减 1，结果放入数据存储器
说明： 本指令将数据存储器内的数值减一再放回数据存储器。
运算过程： [m] ← [m]-1
影响标志位 Z

DECA [m] 数据存储器的内容减 1，结果放入累加器
说明： 本指令将存储器内的数值减一,再放到累加器。
运算过程： ACC ← [m]-1
影响标志位 Z

HALT 进入暂停模式
说明： 本指令终止程序执行并关掉系统时钟，RAM 和寄存器内的数值保持原状态，WDT 计数器清"0"，暂停标志位(PDF)被设为 1， WDT 计数溢出位(TO)被清为 0。
运算过程： TO ← 0
PDF ← 1
影响标志位 TO，PDF

INC [m] 数据存储器的内容加 1，结果放入数据存储器
说明： 本指令将数据存储器内的数值加一,结果放回数据存储器。
运算过程： [m] ← [m]+1
影响标志位 Z

INCA [m] 数据存储器的内容加 1，结果放入累加器
说明： 本指令是将存储器内的数值加一,结果放到累加器。
运算过程： ACC ← [m]+1
影响标志位 Z

JMP addr 无条件跳转
说明： 本指令是将要跳到的目的地直接放到程序计数器内。
运算过程： PC ← addr
影响标志位 没有

MOV A, [m] 将数据存储器送至累加器
说明： 本指令是将数据存储器内的数值送到累加器内。
运算过程： ACC ← [m]
影响标志位 没有

MOV A, x 将立即数送至累加器
说明： 本指令是将立即数送到累加器内。
运算过程： ACC ← x
影响标志位 没有

MOV [m], A 将累加器送至数据存储器
说明： 本指令是将累加器值送到数据存储器内。
运算过程： [m] ← ACC
影响标志位 没有

NOP 空指令
说明： 本指令不作任何运算，而只将程序计数器加一。
运算过程： PC ← PC+1
影响标志位 没有

OR A, [m] 累加器与数据存储器做“或”运算，结果放入累加器
说明： 本指令是把累加器、数据存储器值做逻辑或，结果放到累加器。
运算过程： ACC←ACC “OR” [m]
影响标志位 Z

OR A, x 累加器与立即数做“或”运算，结果放入累加器
说明： 本指令是把累加器值、立即数做逻辑或，结果放到累加器。
运算过程： ACC←ACC “OR” x
影响标志位 Z

ORM A, [m] 累加器与数据存储器做“或”运算，结果放入数据存储器
说明： 本指令是把累加器值、存储器值做逻辑或，结果放到数据存储器。
运算过程： [m]←ACC “OR” [m]
影响标志位 Z

RET 从子程序返回
说明： 本指令是将堆栈寄存器中的程序计数器值送回程序计数器。
运算过程： PC ← Stack
影响标志位 没有

RET A, x 从子程序返回，并将立即数放入累加器
说明： 本指令是将堆栈寄存器中的程序计数器值送回程序计数器，并将立即数送回累加器。
运算过程： PC ← Stack
ACC ← x
影响标志位 没有

RETI 从中断返回
说明： 本指令是将堆栈寄存器中的程序计数器值送回程序计数器，与 RET 不同的是它使用在中断程序结束返回时，它还会将中断控制寄存器 INTC 的 0 位(EMI)中断允许位置 1，允许中断服务。
运算过程： PC ← Stack
EMI ← 1
影响标志位 没有

RL [m] 数据存储器左移一位，结果放入数据存储器
说明： 本指令是将数据存储器内的数值左移一位，第 7 位移到第 0 位，结果送回数据存储器。
运算过程： [m].(i+1) ←[m].i;(i=0~6)
[m].0 ←[m].7
影响标志位 没有

RLA [m] 数据存储器左移一位，结果放入累加器
说明： 本指令是将存储器内的数值左移一位，第 7 位移到第 0 位，结果送到累加器，而数据存储器内的数值不变。
运算过程： ACC.(i+1) ←[m].i; (i=0~6)
ACC.0 ←[m].7
影响标志位 没有

RLC [m] 带进位将数据存储器左移一位，结果放入数据存储器
说明： 本指令是将存储器内的数值与进位位左移一位，第 7 位取代进位标志，进位标志移到第 0 位，结果送回数据存储器。
运算过程： [m].(i+1) ←[m].i; (i=0~6)
[m].0 ←C
C ← [m].7
影响标志位 C

RLCA [m] 带进位将数据存储器左移一位，结果放入累加器
说明： 本指令是将存储器内的数值与进位位左移一位，第七位取代进位标志，进位标志移到第 0 位，结果送回累加器。
运算过程： ACC.(i+1) ←[m].i; (i=0~6)
ACC.0 ←C
C ← [m].7
影响标志位 C

RR [m] 数据存储器右移一位，结果放入数据存储器
说明： 本指令是将存储器内的数值循环右移，第 0 位移到第 7 位，结果送回数据存储器。
运算过程： [m].i ←[m].(i+1); (i=0~6)
[m].7 ←[m].0
影响标志位 没有

RRA [m] 数据存储器右移一位，结果放入累加器
说明： 本指令是将数据存储器内的数值循环右移，第 0 位移到第 7 位，结果送回累加器，而数据存储器内的数值不变。
运算过程： ACC.i ←[m].(i+1); (i=0~6)
ACC.7 ←[m].0
影响标志位 没有

RRC [m] 带进位将数据存储器右移一位，结果放入数据存储器
说明： 本指令是将存储器内的数值加进位位循环右移，第 0 位取代进位标志，进位标志移到第 7 位，结果送回存储器。
运算过程： [m].i ←[m].(i+1); (i=0~6)
[m]. 7 ←C
C ← [m].0
影响标志位 C

RRCA [m] 带进位将数据存储器右移一位，结果放入累加器
说明： 本指令是将数据存储器内的数值加进位位循环右移，第 0 位取代进位标志，进位标志移到第 7 位，结果送回累加器，数据存储器内的数值不变。
运算过程： ACC.i ←[m].(i+1) ; (i=0~6)
ACC.7 ←C
C ← [m].0
影响标志位 C

SBC A,[m] 累加器与数据存储器、进位标志相减，结果放入累加器
说明： 本指令是把累加器值减去数据存储器值以及进位标志的取反，结果放到累加器。
运算过程： ACC← ACC - [m] - $\overline{C}$
影响标志位 OV，Z，AC，C

SBCM A,[m] 累加器与数据存储器、进位标志相减，结果放入数据存储器
说明： 本指令是把累加器值减去数据存储器值以及进位标志取反，结果放到数据存储器。
运算过程： [m]←ACC - [m] - $\overline{C}$
影响标志位 OV，Z，AC，C

SDZ [m] 数据存储器减 1，如果结果为“0”，则跳过下一条指令

说明： 本指令是把数据存储器内的数值减 1，判断是否为 0，若为 0 则跳过下一条指令，即如果结果为零，放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以取得正确的指令(二个指令周期)。否则执行下一条指令(一个指令周期)。

运算过程： [m]← [m]-1

如果[m]=0，跳过下一条指令执行再下一条。

影响标志位 没有

SDZA [m] 数据存储器减 1，将结果放入累加器，如果结果为“0”，则跳过下一条指令

说明： 本指令是把数据存储器内的数值减 1，判断是否为 0, 为 0 则跳过下一行指令并将减完后数据存储器内的数值送到累加器,而数据存储器内的值不变，即若结果为 0，放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以取得正确的指令(二个指令周期)。否则执行下一条指令(一个指令周期)。

运算过程： ACC ←[m]-1

如果 ACC =0，跳过下一条指令执行再下一条。

影响标志位 没有

SET [m] 置位数据存储器

说明： 本指令是把存储器内的数值每个位置为 1。

运算过程： [m] ← FFH

影响标志位 没有

SET [m]. i 将数据存储器的第 i 位置“1”

说明： 本指令是把存储器内的数值的第 i 位置为 1。

运算过程： [m].i ←1

影响标志位 没有

SIZ [m] 数据存储器加 1，如果结果为“0”，则跳过下一条指令

说明： 本指令是把数据存储器内的数值加 1，判断是否为 0。若为 0，跳过下一条指令，即放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以取得正确的指令(二个指令周期)。否则执行下一条指令(一个指令周期)。

运算过程： [m] ←[m]+1

如果 [m]=0，跳过下一行指令

影响标志位 没有

SIZA 数据存储器加 1，将结果放入累加器，如果结果为“0”，则跳过下一条指令

说明： 本指令是把数据存储器内的数值加 1，判断是否为 0, 若为 0 跳过下一条指令，即放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以取得正确的指令(二个指令周期)，并将加完后存储器内的数值送到累加器，而数据存储器的值保持不变。否则执行下一条指令(一个指令周期)。

运算过程： ACC ←[m]+1

` 如果 ACC =0，跳过下一行指令

影响标志位 没有

SNZ [m]. i 如果数据存储器的第 i 位不为“0”，则跳过下一条指令
说明： 本指令是判断数据存储器内的数值的第 i 位, 若不为 0，则程序计数器再加 1, 跳过下一行指令，放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以取得正确的指令(二个指令周期)。否则执行下一条指令(一个指令周期)。
运算过程： 如果 [m].i≠0，跳过下一行指令。
影响标志位 没有

SUB A, [m] 累加器与数据存储器相减，结果放入累加器
说明： 本指令是把累加器值、数据存储器值相减，结果放到累加器。
运算过程： ACC←ACC - [m]
影响标志位 OV，Z，AC，C

SUB A, x 累加器与立即数相减，结果放入累加器
说明： 本指令是把累加器值、立即数相减，结果放到累加器。
运算过程： ACC←ACC - x
影响标志位 OV，Z，AC，C

SUBM A, [m] 累加器与数据存储器相减，结果放入数据存储器
说明： 本指令是把累加器值、存储器值相减，结果放到存储器。
运算过程： [m]←ACC - [m]
影响标志位 OV，Z，AC，C

SWAP [m] 交换数据存储器的高低字节，结果放入数据存储器
说明： 本指令是将数据存储器的低四位和高四位互换,再将结果送回数据存储器。
运算过程： [m].3~[m].0↔[m].7~[m].4
影响标志位 没有

SWAPA [m] 交换数据存储器的高低字节，结果放入累加器
说明： 本指令是将数据存储器的低四位和高四位互换，再将结果送回累加器。
运算过程： ACC.3~ACC.0← [m].7~[m].4
ACC.7~ACC.4← [m].3~[m].0
影响标志位 没有

SZ [m] 如果数据存储器为“0”，则跳过下一条指令
说明： 本指令是判断数据存储器内的数值是否为 0，为 0 则跳过下一行指令，即放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以得正确的指令(二个指令周期)。否则执行下一条指令(一个指令周期)。
运算过程： 如果 [m] = 0, 跳过下一行指令。
影响标志位 没有

SZA [m] 数据存储器送至累加器，如果内容为"0"，则跳过下一条指令
说明： 本指令是判断存储器内的数值是否为 0，若为 0 则跳过下一行指令，即放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以得正确的指令(二个指令周期)。并把存储器内值送到累加器，而存储器的值保持不变。否则执行下一条指令(一个指令周期)。
运算过程： 如果[m] = 0，跳过下一行指令，并 ACC← [m]。
影响标志位 没有

SZ [m]. i 如果数据存储器的第 i 位为"0"，则跳过下一条指令
说明： 本指令是判断存储器内第 i 位值是否为 0，若为 0 则跳过下一行指令，即放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以得正确的指令(二个指令周期)。否则执行下一条指令(一个指令周期)。
运算过程： 如果 [m].i = 0，跳过下一行指令。
影响标志位 没有

TABRDC [m] 读取 ROM 当前页的内容，并送至数据存储器和 TBLH
说明： 本指令是将表格指针指向程序寄存器当前页，将低位送到存储器，高位直接送到 TBLH 寄存器内。
运算过程： [m] ←程序存储器低字节
TBLH←程序存储器高字节
影响标志位 没有

TABRDL [m] 读取 ROM 最后一页的内容，并送至数据存储器和 TBLH
说明： 本指令是将 TABLE 指针指向程序寄存器最后页，将低位送到存储器，高位直接送到 TBLH 寄存器内。
运算过程： [m] ←程序存储器低字节
TBLH←程序存储器高字节
影响标志位 没有

XOR A, [m] 累加器与立即数做"异或"运算，结果放入累加器
说明： 本指令是把累加器值、数据存储器值做逻辑异或，结果放到累加器。
运算过程： ACC←ACC "XOR" [m]
影响标志位 Z

XORM A, [m] 累加器与数据存储器做"异或"运算，结果放入数据存储器
说明： 本指令是把累加器值、数据存储器值做逻辑异或，结果放到数据存储器。
运算过程： [m]←ACC "XOR" [m]
影响标志位 Z

XOR A, x 累加器与数据存储器做"异或"运算，结果放入累加器
说明： 本指令是把累加器值与立即数做逻辑异或，结果放到累加器。
运算过程： ACC←ACC "XOR" x
影响标志位 Z

封装尺寸

16-pin DIP (300mil)外形尺寸

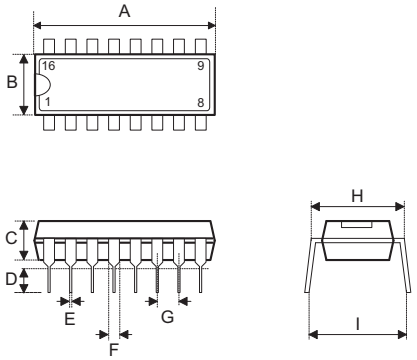


Fig1. Full Lead Packages

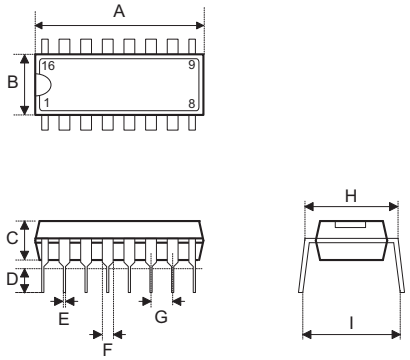


Fig2. 1/2 Lead Packages

MS-001d（见 fig 1）

符号	尺寸 (单位：mil)		
	最小	典型	最大
A	780	—	880
B	240	—	280
C	115	—	195
D	115	—	150
E	14	—	22
F	45	—	70
G	—	100	—
H	300	—	325
I	—	—	430

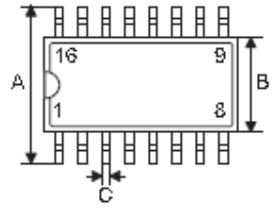
MS-001d（见 fig 2）

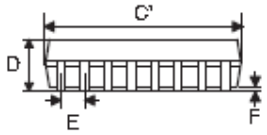
符号	尺寸 (单位：mil)		
	最小	典型	最大
A	735	—	775
B	240	—	280
C	115	—	195
D	115	—	150
E	14	—	22
F	45	—	70
G	—	100	—
H	300	—	325
I	—	—	430

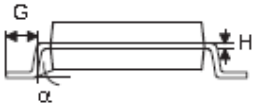
MO-095 a（见 fig 2）

符号	尺寸 (单位：mil)		
	最小	典型	最大
A	745	—	785
B	275	—	295
C	120	—	150
D	110	—	150
E	14	—	22
F	45	—	60
G	—	100	—
H	300	—	325
I	—	—	430

16**-pin NSOP (150mil)外形尺寸**



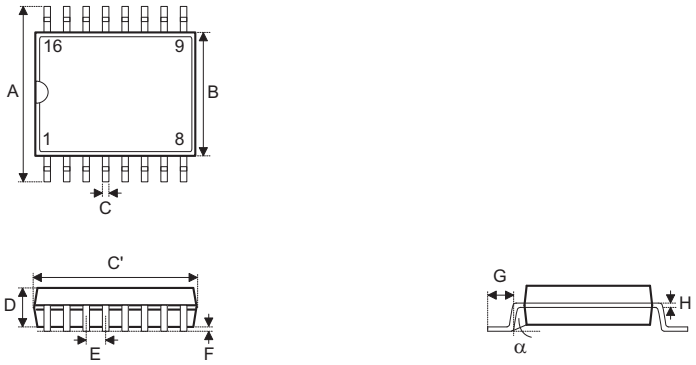




MS-012

符号	尺寸 (单位：mil)		
	最小	典型	最大
A	228	—	244
B	150	—	157
C	12	—	20
C'	386	—	394
D	—	—	69
E	—	50	—
F	4	—	10
G	16	—	50
H	7	—	10
α	0°	—	8°

16-pin SSOP (150mil)外形尺寸



符号	尺寸 (单位：mil)		
	最小	典型	最大
A	228	—	244
B	150	—	157
C	8	—	12
C’	189	—	197
D	54	—	60
E	—	25	—
F	4	—	10
G	22	—	28
H	7	—	10
α	0°	—	8°

20-pin DIP (300mil)外形尺寸

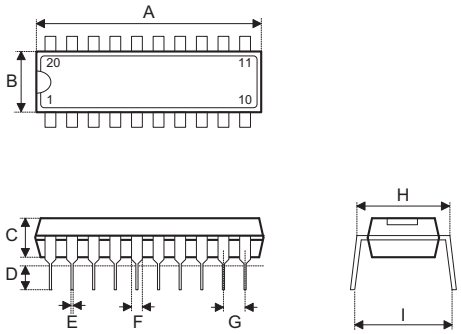


Fig1. Full Lead Packages

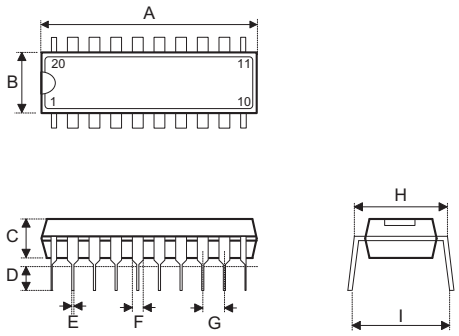


Fig2. 1/2 Lead Packages

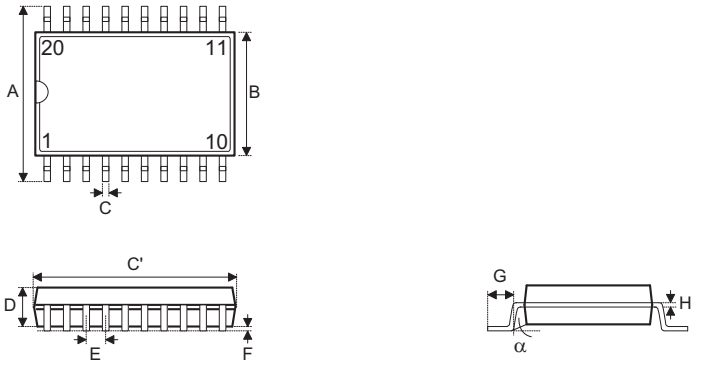
MS-001d（见 fig 1）

符号	尺寸 (单位：mil)		
	最小	典型	最大
A	980	—	1060
B	240	—	280
C	115	—	195
D	115	—	150
E	14	—	22
F	45	—	70
G	—	100	—
H	300	—	325
I	—	—	430

MS-095a（见 fig 2）

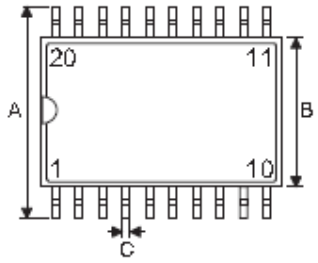
符号	尺寸 (单位：mil)		
	最小	典型	最大
A	945	—	985
B	275	—	295
C	120	—	150
D	110	—	150
E	14	—	22
F	45	—	60
G	—	100	—
H	300	—	325
I	—	—	430

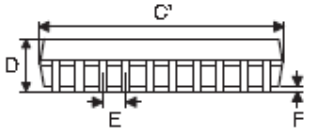
20**-pin SSOP (150mil)外形尺寸**

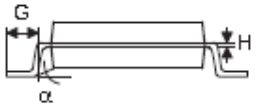


符号	尺寸 (单位：mil)		
	最小	典型	最大
A	228	—	244
B	150	—	158
C	8	—	12
C'	335	—	347
D	49	—	65
E	—	25	—
F	4	—	10
G	15	—	50
H	7	—	10
α	0°	—	8°

20**-pin SOP (300mil)外形尺寸**



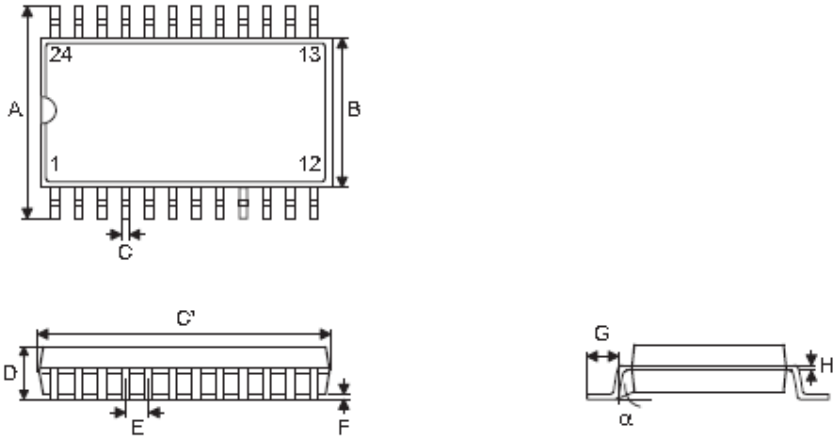




MS-013

符号	尺寸 (单位：mil)		
	最小	典型	最大
A	393	—	419
B	256	—	300
C	12	—	20
C’	496	—	512
D	—	—	104
E	—	50	—
F	4	—	12
G	16	—	50
H	8	—	13
α	0°	—	8°

24-pin SOP (300mil)外形尺寸

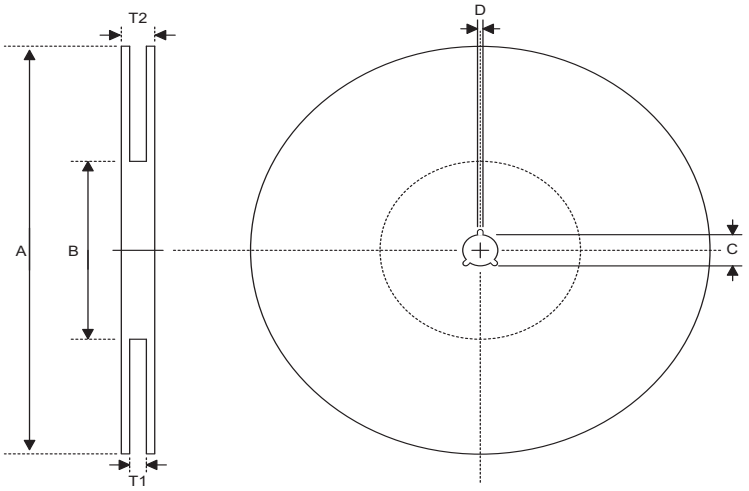


MS-013

符号	尺寸 (单位：mil)		
	最小	典型	最大
A	393	—	419
B	256	—	300
C	12	—	20
C'	598	—	613
D	—	—	104
E	—	50	—
F	4	—	12
G	16	—	50
H	8	—	13
α	0°	—	8°

包装带和卷轴规格：

卷轴尺寸：



SOP 16N（150mil），SSOP 20S（150mil）

标号	描述	尺寸(mm)
A	卷轴外圈直径	330.0±1.0
B	卷轴内圈直径	100.0±1.5
C	轴心直径	13.0+0.5/-0.2
D	缝宽	2.0±0.5
T1	轮缘宽	16.8+0.3 -0.2
T2	卷轴宽	22.2±0.2

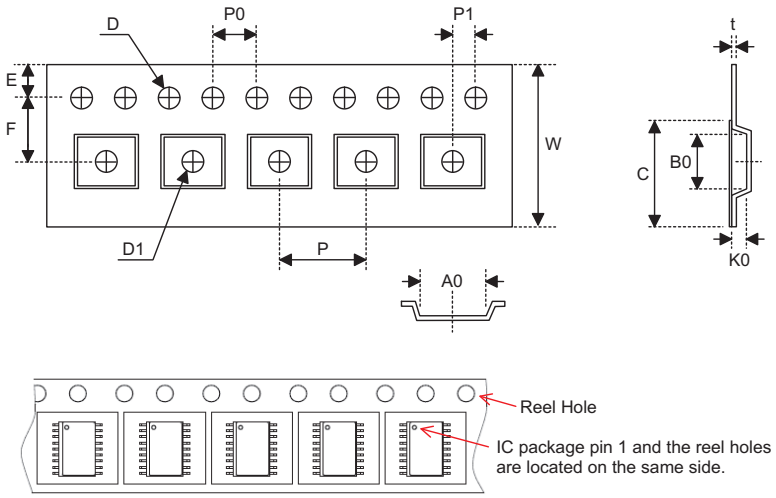
SSOP 16S

标号	描述	尺寸(mm)
A	卷轴外圈直径	330±1.0
B	卷轴内圈直径	100±1.5
C	轴心直径	13.0+0.5 -0.2
D	缝宽	2.0±0.5
T1	轮缘宽	12.8+0.3 -0.2
T2	卷轴宽	18.2±0.2

SOP 20W，SOP 24W

标号	描述	尺寸(mm)
A	卷轴外圈直径	330±1.0
B	卷轴内圈直径	100±1.5
C	轴心直径	13.0+0.5 -0.2
D	缝宽	2.0±0.5
T1	轮缘宽	24.8+0.3 -0.2
T2	卷轴宽	30.2±0.2

运输带尺寸：



SOP 16N（150mil）

标号	描述	尺寸(mm)
W	运输带宽	16.0±0.3
P	空穴间距	8.0±0.1
E	穿孔位置	1.75±0.1
F	空穴至穿孔距离（宽度）	7.5±0.1
D	穿孔直径	1.55+0.1
D1	空穴中之小孔直径	1.5+0.25
P0	穿孔间距	4.0±0.1
P1	空穴至穿孔距离（长度）	2.0±0.1
A0	空穴长	6.5±0.1
B0	空穴宽	10.3±0.1
K0	空穴深	2.1±0.1
t	传输带厚度	0.3±0.05
C	覆盖带宽度	13.3±0.1

SSOP 16S

标号	描述	尺寸(mm)
W	运输带宽	12.0+0.3 -0.1
P	空穴间距	8.0±0.1
E	穿孔位置	1.75±0.1
F	空穴至穿孔距离（宽度）	5.5±0.1
D	穿孔直径	1.55+0.1
D1	空穴中之小孔直径	1.5+0.25
P0	穿孔间距	4.0±0.1
P1	空穴至穿孔距离（长度）	2.0±0.1
A0	空穴长	6.4±0.1
B0	空穴宽	5.2±0.1
K0	空穴深	2.1±0.1
t	传输带厚度	0.3±0.05
C	覆盖带宽度	9.3±0.1

SSOP 20S（150mil）

标号	描述	尺寸(mm)
W	运输带宽	16.0+0.3 -0.1
P	空穴间距	8.0±0.1
E	穿孔位置	1.75±0.1
F	空穴至穿孔距离（宽度）	7.5±0.1
D	穿孔直径	1.5+0.1
D1	空穴中之小孔直径	1.5+0.25
P0	穿孔间距	4±0.1
P1	空穴至穿孔距离（长度）	2±0.1
A0	空穴长	6.5±0.1
B0	空穴宽	9.0±0.1
K0	空穴深	2.3±0.1
t	传输带厚度	0.3±0.05
C	覆盖带宽度	13.3±0.1

SOP 20W

标号	描述	尺寸(mm)
W	运输带宽	24.0+0.3 -0.1
P	空穴间距	12.0±0.1
E	穿孔位置	1.75±0.1
F	空穴至穿孔距离（宽度）	11.5±0.1
D	穿孔直径	1.5+0.1
D1	空穴中之小孔直径	1.5+0.25
P0	穿孔间距	4±0.1
P1	空穴至穿孔距离（长度）	2±0.1
A0	空穴长	10.8±0.1
B0	空穴宽	13.3±0.1
K0	空穴深	3.2±0.1
t	传输带厚度	0.3±0.05
C	覆盖带宽度	21.3±0.1

SOP 24W

标号	描述	尺寸(mm)
W	运输带宽	24.0±0.3
P	空穴间距	12.0±0.1
E	穿孔位置	1.75±0.1
F	空穴至穿孔距离（宽度）	11.5±0.1
D	穿孔直径	1.55+0.1
D1	空穴中之小孔直径	1.5+0.25
P0	穿孔间距	4±0.1
P1	空穴至穿孔距离（长度）	2±0.1
A0	空穴长	10.9±0.1
B0	空穴宽	15.9±0.1
K0	空穴深	3.1±0.1
t	传输带厚度	0.35±0.05
C	覆盖带宽度	21.3±0.1

盛群半导体股份有限公司（总公司）
新竹市科学工业园区研新二路 3 号
电话: 886-3-563-1999
传真: 886-3-563-1189
网站: www.holtek.com.tw

盛群半导体股份有限公司（台北业务处）
台北市南港区园区街 3 之 2 号 4 楼之 2
电话: 886-2-2655-7070
传真: 886-2-2655-7373
传真: 886-2-2655-7383 (International sales hotline)

盛扬半导体有限公司（上海业务处）
上海宜山路 2016 号合川大厦 1 号楼 3 楼 G 室 201103
电话: 021-5422-4590
传真: 021-5422-4596
网站: www.holtek.com.cn

盛扬半导体有限公司（深圳业务处）
深圳市南山区科技园科技中三路与高新中二道交汇处生产力大楼 A 单元五楼 518057
电话: 0755-8616-9908,8616-9308
传真: 0755-8616-9722

盛扬半导体有限公司（北京业务处）
北京市西城区宣武门西大街甲 129 号金隅大厦 1721 室 100031
电话: 010-6641-0030, 6641-7751, 6641-7752
传真: 010-6641-0125

盛扬半导体有限公司（成都业务处）
成都市东大街 97 号香槟广场 C 座 709 室 610016
电话: 028-6653-6590
传真: 028-6653-6591

Holtek Semiconductor(USA), Inc.（北美业务处）
46712 Fremont Blvd., Fremont, CA 94538
电话: 510-252-9880
传真: 510-252-9885
网站: www.holtek.com