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

Copyright © 2007 by HOLTEK SEMICONDUCTOR INC.

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

- 工具信息
- FAQs
- 应用范例

特性

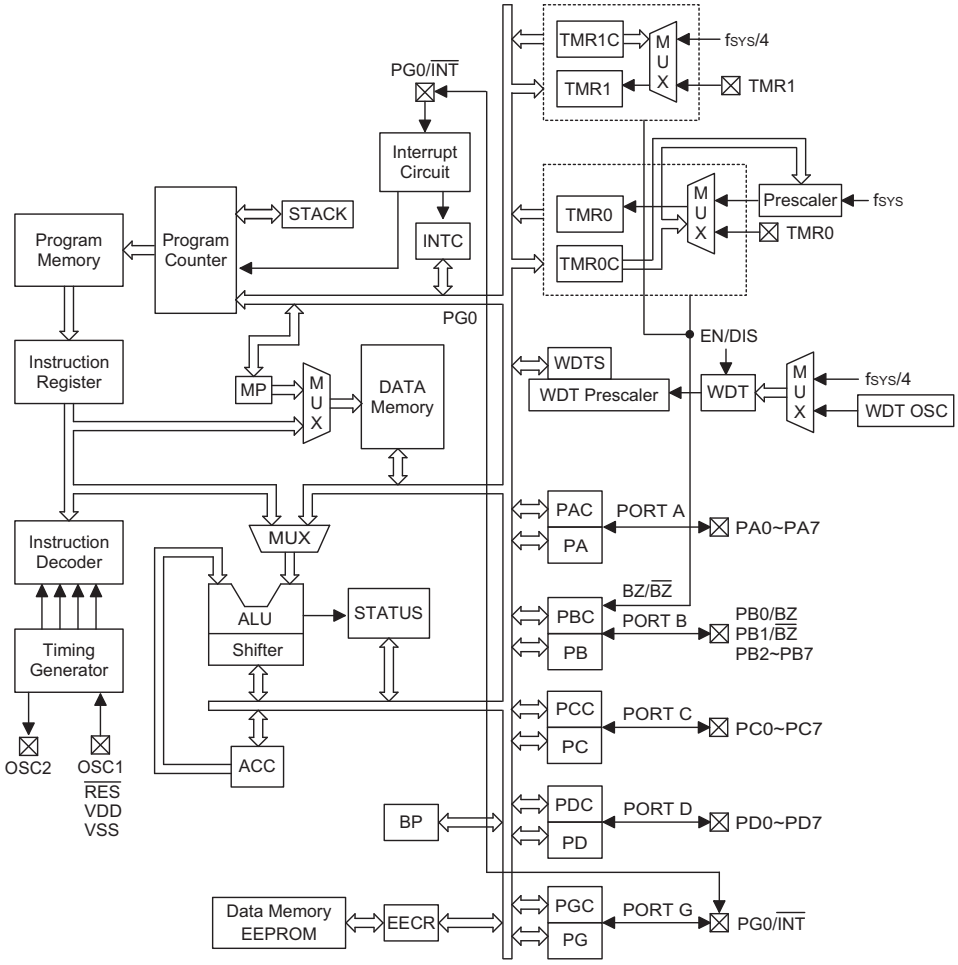
- 工作电压：
  f_{SYS} = 4MHz ： 2.2V ~ 5.5V
  f_{SYS} = 8MHz ： 3.3V ~ 5.5V
- 低电压复位功能
- 最多 33 个双向输入/输出口
- 1 个与输入/输出共用引脚的外部中断输入
- 8 位可编程定时/计数器，具有溢出中断和 8 级预分频器
- 内置晶体振荡和 RC 振荡电路
- 看门狗定时器
- 100,000 次可擦/写闪存程序存储器
- 4096×15 程序存储器（FLASH）
- 256×8 数据存储器 EEPROM
- 160×8 数据存储器 RAM
- HALT 功能和唤醒可降低功耗
- 6 层硬件堆栈
- 在 VDD=5V，系统时钟为 8MHz 时，指令周期为 0.5 μs
- 位操作指令
- 查表指令，表格内容字长 15 位
- 63 条指令
- 10^6 次可擦/写 EEPROM 数据存储器
- EEPROM 数据有效期>10 年
- 所有指令在 1 或 2 个指令周期内完成
- 在线烧写功能（ISP）
- 28-pin SKDIP/SOP，48-pin SSOP 封装

概述

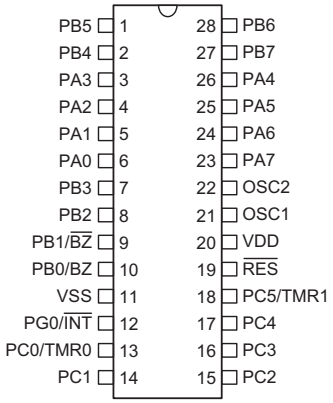
HT48F50E 是一款八位高性能精简指令集单片机，专为经济型多输入/输出控制的产品设计。

拥有低功耗、灵活的 I/O 口功能、定时器功能、振荡选择、省电和唤醒功能、看门狗定时器、蜂鸣器驱动、以及低价位等优势，使此款多功能芯片可以广泛地适用于各种应用，例如工业控制、消费类产品、子系统控制器等。

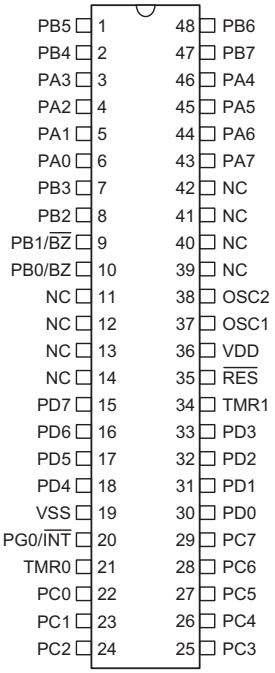
方框图


TMR1C
TMR1
MUX
fSYS/4
TMR1
PG0/INT
Interrupt Circuit
Prescaler
fSYS
TMR0
TMR0C
TMR0
STACK
Program Memory
Program Counter
INTC
PG0
EN/DIS
Instruction Register
MP
DATA Memory
WDTS
WDT Prescaler
WDT
fSYS/4
WDT OSC
PAC
PA
PORT A
PA0~PA7
Instruction Decoder
MUX
ALU
Shifter
STATUS
BZ/BZ
PBC
PB
PORT B
PB0/BZ
PB1/BZ
PB2~PB7
Timing Generator
PCC
PC
PORT C
PC0~PC7
OSC2
OSC1
RES
VDD
VSS
ACC
BP
PDC
PD
PORT D
PD0~PD7
Data Memory EEPROM
EECR
PGC
PG
PORT G
PG0/INT

引脚图



HT48F50E
– 28 SKDIP-A/SOP-A



HT48F50E
– 48 SSOP-A

引脚说明

引脚名称	输入输出	掩膜选项	说明
PA0~PA7	输入/输出	上拉电阻* 唤醒功能 CMOS/斯密特触发输入	双向 8 位输入/输出口。每一位能被掩膜选项设置为唤醒输入。软件指令设置为 CMOS 输出或斯密特触发输入，作为 CMOS 输入时可由掩膜选项设置是否带上拉电阻（由上拉电阻选项决定）。
PB0/BZ PB1/$\overline{BZ}$ PB2~PB7	输入/输出	上拉电阻* I/O 或 BZ/$\overline{BZ}$	双向 8 位输入/输出口。软件指令设定为 CMOS 输出或带上拉电阻的斯密特触发输入（由上拉电阻选项确定）。 PB0 和 PB1 分别与 BZ 和$\overline{BZ}$共用引脚。一旦 PB0 和 PB1 被选为蜂鸣器输出，输出信号来自内部 PFD 发生器（与定时/计数器 0 共用）。
PD0~PD7	输入/输出	上拉电阻*	双向输入/输出口。软件指令设定为 CMOS 输出或带上拉电阻的斯密特触发输入（由上拉电阻选项确定）。
VSS	—	—	负电源，接地。
PG0/$\overline{INT}$	输入/输出	上拉电阻*	双向输入/输出口。软件指令设定为 CMOS 输出或带上拉电阻的斯密特触发输入（由上拉电阻选项确定）。外部中断输入$\overline{INT}$与 PG0 共用引脚。外部中断输入信号下降沿有效。
TMR0	输入	—	定时/计数器 0 斯密特触发输入（无上拉电阻）。
PC0~PC7	输入/输出	上拉电阻*	双向输入/输出口。软件指令设定为 CMOS 输出或带上拉电阻的斯密特触发输入（由上拉电阻选项确定）。
TMR1	输入	—	定时/计数器 1 斯密特触发输入（无上拉电阻）。
$\overline{RES}$	输入	—	斯密特触发复位输入端，低电平有效。
VDD	—	—	正电源。
OSC1 OSC2	输入 输出	晶体振荡 或 RC 振荡	OSC1 和 OSC2 连接 RC 或晶体振荡器（由掩膜选项确定）以产生内部系统时钟。在 RC 方式下，OSC2 为系统时钟四分频输出端口。

注：“*”所有输入/输出口（PA，PB，PC，PD，PG）的上拉电阻由一个选择位控制。

PA 口 CMOS 或斯密特触发由一个选择位控制。

极限参数

电源供应电压 …… V_{SS} - 0.3V 至 Vss + 6.0V
端口输入电压 …… Vss - 0.3V 至 V_{DD} + 0.3V
端口总灌电流 …… 150mA
总功耗 …… 500mW

储存温度 …… -50℃至 125℃
工作温度 …….. -40℃至 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
I_{DD1}	工作电流 (晶体振荡)	3V	无负载 f_{SYS} =4MHZ	—	1	2	mA
		5V		—	3	5	mA
I_{DD2}	工作电流 (RC 振荡)	3V	无负载 f_{SYS} =4MHZ	—	1	2	mA
		5V		—	2.5	4	mA
I_{DD3}	工作电流 (晶体振荡，RC 振荡)	5V	无负载 f_{SYS} =8MHZ	—	4	8	mA
I_{STB1}	静态电流 (看门狗打开)	3V	无负载* 暂停模式	—	—	5	μA
		5V		—	—	10	μA
I_{STB2}	静态电流 (看门狗关闭)	3V	无负载* 暂停模式	—	—	1	μA
		5V		—	—	2	μA
V_{IL1}	输入/输出口的低电平输入电压	—	—	0	—	$0.3V_{DD}$	V
V_{IH1}	输入/输出口的高电平输入电压	—	—	$0.7V_{DD}$	—	V_{DD}	V
V_{IL2}	低电平输入电压 ($\overline{RES}$)	—	—	0	—	$0.4V_{DD}$	V
V_{IH2}	高电平输入电压 ($\overline{RES}$)	—	—	$0.9V_{DD}$	—	V_{DD}	V
V_{LVR}	低电压复位	—	LVR 打开	2.7	3.0	3.3	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

注：“*”所有的测试都参照 I/O 口被设置为输出并输出低电平的情况进行。

交流电气特性

Ta=25℃

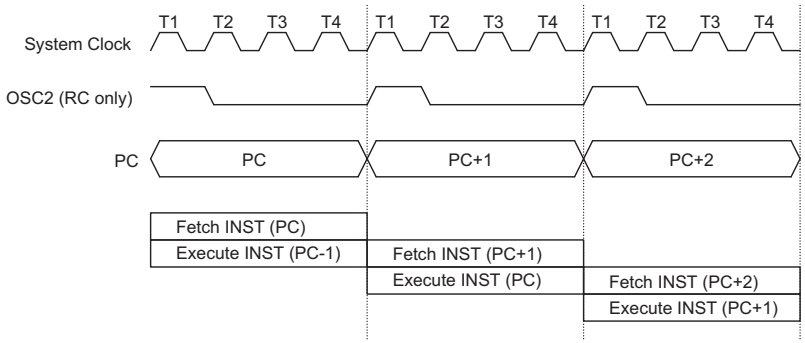
符号	参数	测试条件		最小	典型	最大	单位
		V_{DD}	条件				
f_{SYS1}	系统时钟（晶体振荡）	—	2.2V~5.5V	400	—	4000	kHz
		—	3.3V~5.5V	400	—	8000	kHz
f_{SYS2}	系统时钟（RC 振荡）	—	2.2V~5.5V	400	—	4000	kHz
		—	3.3V~5.5V	400	—	8000	kHz
f_{TIMER}	定时器输入频率(TMR)	—	2.2V~5.5V	0	—	4000	kHz
		—	3.3V~5.5V	0	—	8000	kHz
t_{WDTOSC}	看门狗振荡器周期	3V	—	45	90	180	μs
		5V	—	32	65	130	μs
t_{WDT1}	看门狗溢出周期 (WDT 振荡)	3V	WDT 无预分频	11	23	46	ms
		5V		8	17	33	ms
t_{WDT2}	看门狗溢出周期(系统时钟)	—	WDT 无预分频	—	1024	—	t_{SYS}
t_{RES}	外部复位低电平脉宽	—	—	1	—	—	μs
t_{SST}	系统启动延时周期	—	HALT 模式唤醒	—	1024	—	t_{SYS}
t_{INT}	中断脉冲宽度	—	—	1	—	—	μs

系统功能说明

指令执行时序

单片机的系统时钟由晶体振荡器或 RC 振荡器产生。该时钟在芯片内部被分成四个互不重叠的时钟周期。一个指令周期包括四个系统时钟周期。

指令的读取和执行是以流水线方式进行的, 这种方式在一个指令周期进行读取指令操作，而在下一个指令周期进行解码与执行该指令。因此，流水线方式使多数指令能在一个周期内执行完成。但如果涉及到的指令要改变程序计数器的值，就需要花两个指令周期来完成这一条指令。



指令执行时序

程序计数器

程序计数器(PC)控制程序存储器 ROM 中指令执行的顺序，它可寻址整个 ROM 的范围。

取得指令码以后，程序计数器会自动加一，指向下一个指令码的地址。但如果执行跳转、条件跳跃、向 PCL 赋值、子程序调用、初始化复位、内部中断、外部中断、子程序返回等操作时，PC 会载入与指令相关的地址而非下一条指令地址。

当遇到条件跳跃指令且符合条件时，当前指令执行过程中读取的下一条指令会被丢弃，取而代之的是一个空指令周期，随后才能取得正确的指令。反之，就会顺序执行下一条指令。

程序计数器的低字节（PCL）是一个可读写的寄存器（06H）。对 PCL 赋值将产生一个短跳转动作，跳转的范围为当前页 256 个地址。

当遇到控制转移指令时，系统也会插入一个空指令周期。

模　　式	程序计数器											
	***11**	***10**	***9**	***8**	***7**	***6**	***5**	***4**	***3**	***2**	***1**	***0**
初始化复位	0	0	0	0	0	0	0	0	0	0	0	0
外部中断	0	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	0	1	0	0	0
定时/计数器 1 溢出	0	0	0	0	0	0	0	0	1	1	0	0
条件跳转	PC+2											
装载 PCL	*11	*10	*9	*8	@7	@6	@5	@4	@3	@2	@1	@0
跳转，子程序调用	#11	#10	#9	#8	#7	#6	#5	#4	#3	#2	#1	#0
从子程序返回	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

程序计数器

注：*11 ~ *0 ： 程序计数器位　　S11~ S0 ： 堆栈寄存器位
　　#11 ~ #0 ： 指令代码位　　@7 ~ @0 ： PCL 位

在线烧写（In System Programming）

在线烧写允许在实际应用电路板上对 HT48FXXE 单片机进行烧写和重复烧写操作，这样节省了开发周期和成本。ISP（在线烧写）使用 3 线接口与 HT48FXXE 单片机进行通讯，对芯片内部的程序存储器和 EEPROM 数据存储器进行重复烧写操作。

引脚名称	功能	说明
PA0	SDATA	串行数据输入/输出
PA4	SCLK	串行时钟输入
RES	RESET	复位
VDD	VDD	电源
VSS	VSS	地

ISP 引脚分配

程序存储器

程序存储器用来存放要执行的指令代码，以及一些数据、表格和中断入口。程序存储器有 4096×15 位，程序存储器空间可以用程序计数器或表格指针进行寻址。

以下列出的程序存储器地址是系统专为特殊用途而保留的：

- 地址 000H

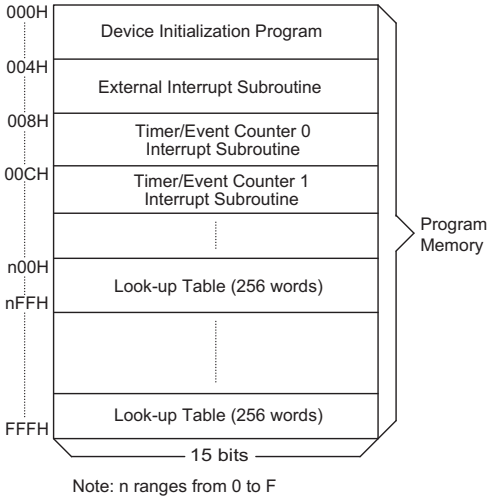
该地址为程序初始化保留。系统复位后，程序总是从 000H 开始执行。

- 地址 004H

该地址为外部中断服务程序保留。当 $\overline{\text{INT}}$ 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。

- 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 溢出，如果中断允许且堆栈未满，则程序会跳转到 008H 地址开始执行。



程序存储器

- 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 溢出，如果中断允许且堆栈未满，则程序会跳转到 00CH 地址开始执行。

- 表格区

ROM 空间的任何地址都可作为查表使用。查表指令“TABRDC [m]”(查当前页表格，1 页=256 个字)和“TABRDL [m]”(查最后页表格)，会把表格内容低字节传送给[m]，而表格内容高字节传送到 TBLH 寄存器(08H)。只有表格内容的低字节被传送到目标地址中，而高字节被传送到表格内容高字节寄存器 TBLH，并且 TBLH 的最高位始终为“0”。表格内容高字节寄存器 TBLH 是只读寄存器。表格指针(TBLP)是可读/写寄存器(07H)，用来指明表格地址。在查表之前，要先将表格地址写入 TBLP 中。如果主程序和中断服务程序(ISR)都用到查表指令，主程序中 TBLH 的值可能会因为 ISR 中执行的查表指令而发生变化，产生错误。也就是说，要避免在主程序和中断服务程序中都使用查表指令。但如果必须这样做的话，我们可以在查表指令前先将中断禁止，在保存了 TBLH 的值后再开放中断以避免发生错误。所有与表格有关的指令都需要两个指令周期的执行时间。这里提到的表格区都可以做为正常的程序存储器来使用。

指令	表格地址											
	***11**	***10**	***9**	***8**	***7**	***6**	***5**	***4**	***3**	***2**	***1**	***0**
TABRDC [m]	P11	P10	P9	P8	@7	@6	@5	@4	@3	@2	@1	@0
TABRDL [m]	1	1	1	1	@7	@6	@5	@4	@3	@2	@1	@0

表格区

注：*11 ~ *0 ：表格地址　　P11~P8 ：当前程序计数器
@7 ~ @0 ：表格指针

堆栈寄存器

堆栈寄存器是特殊的存储器空间，用来保存 PC 的值。系统提供 6 层堆栈，堆栈寄存器既不是数据存储器的一部分，也不是程序存储器的一部分，而且它既不能读出，也不能写入。堆栈的使用是通过堆栈指针（SP）来指示的，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。

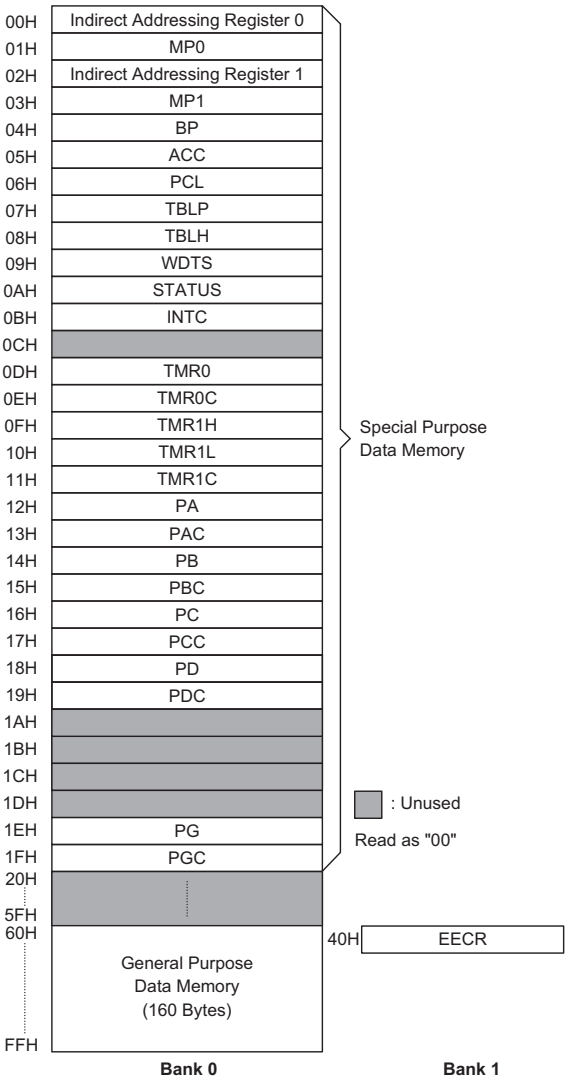
如果堆栈已满，并且发生了非屏蔽中断，那么只有中断请求标志会被记录下来，而中断响应会被禁止，直到堆栈指针发生递减（执行 RET 或 RETI 指令），中断才会被响应。这个功能可以防止堆栈溢出，使得程序员易于使用这种结构。同样，如果堆栈已满，并且发生了子程序调用，那么堆栈会发生溢出，首先进入堆栈的内容将会丢失，只有最后的 6 个返回地址会被保留。

数据存储器

数据存储器由 184×8 位组成，分为两个功能区间：特殊功能寄存器和通用数据存储器（160×8），数据存储器单元大多数是可读/写的，但有些为只读。

60H 之前的未使用空间保留给系统以后扩展使用，读取这些地址返回“00H”。BANK0 的通用数据存储器地址 60H~0FFH 作为数据和控制信息使用。

所有的数据存储器单元都能直接执行算术、逻辑、递增、递减和循环操作。除了一些特殊位外，数据存储器的每一位都可由“SET[m].i”置位或由“CLR[m].i”复位。而且都可以通过间接寻址指针(MP0 或 MP1)进行间接寻址。EEPROM 数据存储器的控制寄存器在 BANK1 的[40H]单元。



数据存储器

间接寻址寄存器

地址00H和02H是作为间接寻址寄存器，没有实际的物理空间。任何对[00H]和[02H]的读/写操作，都会访问由MP0(01H)和MP1(03H)所指向的RAM单元。间接地读取[00H]或[02H]，将会返回00H。而间接地写入[00H]或[02H]，则不会产生任何结果。

间接寻址寄存器之间不支持数据传送功能。间接寻址指针MP0和MP1宽度都是8位，它们与间接寻址寄存器相结合用来间接访问RAM。MP0只能应用于数据存储区的BANK0，而MP1能应用于数据存储区的BANK0和BANK1。

累加器

累加器（ACC）与算术逻辑单元（ALU）紧密联系。它对应于 RAM 的地址 05H，并能与立即数进行操作，存储器间的数据传送都必须经过累加器。

算术逻辑单元

算术逻辑单元（ALU）是执行 8 位算术、逻辑运算的电路，它提供有以下功能：

- 算术运算（ADD，ADC，SUB，SBC，DAA）
- 逻辑运算（AND，OR，XOR，CPL）
- 移位运算（RL，RR，RLC，RRC）
- 递增和递减（INC，DEC）
- 分支判断（SZ，SNZ，SIZ，SDZ…）

ALU 不仅可以储存数据运算的结果，还会改变状态寄存器的值。

状态寄存器

状态寄存器（0AH）由零标志位（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）和看门狗定时器溢出标志位（TO）组成。该寄存器不仅记录状态信息，而且还控制操作顺序。

除了 PDF 和 TO 标志外，状态寄存器的其它位都可以用指令改变。任何对状态寄存器的写操作都不会改变 PDF 和 TO 的值。对状态寄存器的操作可能会导致与预期不一样的结果。TO 标志只受系统上电、看门狗溢出、“CLR WDT” 指令或“HALT” 指令的影响。PDF 标志只受系统上电、“CLR WDT” 指令或“HALT” 指令的影响。

位	符号	功能
0	C	如果在加法运算中结果产生了进位或在减法运算中结果不产生借位，则 C 被置位；反之，C 被清除。它也可被循环移位指令影响。
1	AC	如果在加法运算中低 4 位产生了进位或减法运算中低 4 位不产生借位，则 AC 被置位；反之，AC 被清除。
2	Z	如果算术或逻辑运算的结果为零，则 Z 被置位；反之，Z 被清除。
3	OV	如果运算结果向最高位进位，但最高位并不产生进位输出，则 OV 被置位；反之，OV 被清除。
4	PDF	系统上电或执行“CLR WDT”指令，PDF 被清除； 执行“HALT”指令，PDF 被置位。
5	TO	系统上电、执行“CLR WDT”或“HALT”指令，TO 被清除； WDT 定时溢出，TO 被置位。
6，7	—	未用，读出为“0”

STATUS(0AH) 寄存器

标志位 Z、OV、AC 和 C 反映的是最近一次操作的状态。

在进入中断程序或子程序调用时，状态寄存器不会被自动压入堆栈。如果状态寄存器的内容是重要的，而且子程序会影响状态寄存器的内容，那么程序员必须事先将 STATUS 的值保存好。

中 断

HT48F50E 提供一个外部中断和内部定时/计数器中断。中断控制寄存器（INTC；0BH）包含了中断控制位和中断请求标志，中断控制位用来设置中断允许/禁止。

只要有中断子程序被服务，其余的中断全部都被自动禁止（通过清除 EMI 位），这种做法的目的在于防止中断嵌套。这时如果有其它中断发生，只有中断请求标志会被记录下来。如果在中断服务程序中有另一个中断需要响应，程序员可以置位 EMI 和 INTC 所对应的位，以便进行中断嵌套。如果堆栈已满，则中断并不会被响应，一直到堆栈指针（SP）发生递减后才会响应。如果需要中断立即得到响应，应避免堆栈饱和。

位	符号	功　　能
0	EMI	主中断控制位（1=允许，0=禁止）
1	EEI	外部中断控制位（1=允许，0=禁止）
2	ET0I	定时/计数器 0 中断控制位（1=允许，0=禁止）
3	ET1I	定时/计数器 1 中断控制位（1=允许，0=禁止）
4	EIF	外部中断请求标志位（1=有效， 0=无效）
5	T0F	定时/计数器中断 0 请求位（1=有效， 0=无效）
6	T1F	定时/计数器中断 1 请求位（1=有效， 0=无效）
7	—	未使用位，读出为零

INTC（0BH）寄存器

所有的中断都具有唤醒能力。当有中断被服务，系统会将程序计数器值压入堆栈，然后再跳转至中断服务程序的入口。但这时只有程序计数器的内容被压入堆栈，如果其它寄存器和状态寄存器的内容会被中断程序改变，从而会破坏主程序的控制流程的话，程序员应该事先将这些数据保存起来。

外部中断是由$\overline{\text{INT}}$引脚下降沿信号触发的，其中断请求标志位（EIF；INTC 的第 4 位）会被置位。如果中断允许，且堆栈未满，当发生外部中断时，会产生地址 04H 的子程序调用；而中断请求标志 EIF 和总中断控制位 EMI 会被清除，以禁止其它中断响应。

内部定时/计数器 0 中断是由定时/计数器 0 溢出触发的，其中断请求标志（T0F；INTC 的第 5 位）会被置位。如果中断允许，且堆栈未满，当发生定时/计数器 0 中断时，会产生地址 08H 的子程序调用；而中断请求标志 T0F 和总中断控制位 EMI 会被清除，以禁止其它中断响应。

内部定时/计数器 1 中断是由定时/计数器 1 溢出触发的，其中断请求标志（T1F；INTC 的第 6 位）会被置位。如果中断允许，且堆栈未满，当发生定时/计数器 1 中断时，会产生地址 0CH 的子程序调用；而中断请求标志 T1F 和总中断控制位 EMI 会被清除，以禁止其它中断响应。

在执行中断子程序期间，其它的中断会被屏蔽，直到执行 RETI 指令或 EMI 和相关中断控制位被置位（当然，此时堆栈未满）。如果要从中断子程序返回，只要执行 RET 或 RETI 指令即可。其中，RETI 指令会自动置位 EMI，以允许中断服务，而 RET 则不会。

如果中断在两个连续的 T2 脉冲的上升沿之间发生，且中断响应允许，那么在下一个 T2 脉冲，该中断会被服务。如果同时发生中断请求，其优先级如下表示；也可以通过设定各中断相关的控制位来改变优先级。

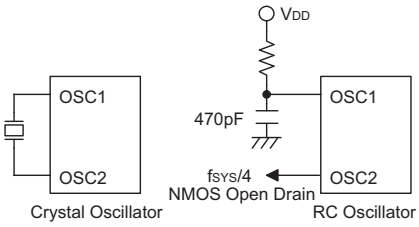
中　断　源	优 先 级	中断
外部中断	1	04H
定时/计数器 0 溢出	2	08H
定时/计数器 1 溢出	3	0CH

一旦中断请求标志（T0F，T1F，EIF）被置位，会一直保留在 INTC 寄存器中，直到中断被响应或用软件指令清除为止。建议不要在中断服务程序中使用“CALL”指令来调用子程序。因为中断随时都可能发生，而且在一些应用中需要立刻给予响应。如果只剩下一层堆栈，而中断使能不能被很好地控制，原先的控制序列很可能因为在中断子程序中执行“CALL”指令而使堆栈溢出，从而发生混乱。

振荡器

HT48F50E 有两种振荡方式，外部 RC 振荡和外部晶体振荡，可以通过掩膜选项设定，不管选用哪一种振荡方式，其信号都可以作为系统时钟。HALT 模式会停止系统振荡器，并忽视任何外部信号以降低功耗。

如果使用 RC 振荡方式，在 OSC1 和 VDD 之间需要一个外部电阻，其阻值范围为 24kΩ至 1MΩ。在 OSC2 端可获得系统时钟四分频信号，可用于同步外部逻辑电路。RC 振荡方式是一种低成本方案，但是振荡频率会随着 VDD、温度和制造漂移而不同。因此，在需要非常精确时基场合，不建议使用 RC 振荡。



系统振荡器

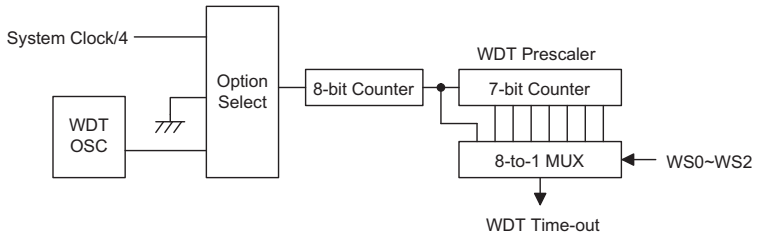
如果选用晶体振荡器，那么在 OSC1 和 OSC2 之间需要连接一个晶体，用来提供晶体振荡器所需要的反馈和相移，无需其他外部元件。另外，在 OSC1 和 OSC2 之间还可以用谐振器代替晶体振荡器产生系统时钟，但是在 OSC1 和 OSC2 需要多连接两个电容。

WDT 振荡器是一个内部 RC 振荡器，并不需要连接任何外部元件。当系统进入暂停模式时，系统时钟会停止，但 WDT 振荡器会继续工作，其振荡周期大约为 65μs/5V。如果要降低功耗，可在掩膜选项中关闭 WDT 振荡器。

看门狗定时器

看门狗定时器的时钟来源有两种：看门狗振荡器或指令时钟（系统时钟 4 分频），由掩膜选项设置。看门狗定时器主要用来防止程序运行故障和程序跳入一死循环而导致不可预测的结果。看门狗定时器可由掩膜选项设置为打开或关闭，如果在关闭状态，所有与 WDT 有关的指令操作都是没有作用的。

如果 WDT 时钟源为内部 WDT 振荡器输出（RC 振荡器周期正常为 65μs/5V），WDT 先经过 256 分频（8 级）产生大约 16.6ms/5V 的溢出周期。溢出周期会因为温度、V_{DD} 以及芯片参数的差异而变化。通过设置 WDT 预分频器，则可得到更长的溢出周期。写数据到 WS2,WS1,WS0 (WDTS 的 2、1、0 位)会产生不同的溢出周期，如果 WS2，WS1，WS0 的值都为 1，其分频系数为 1：128，其最长溢出周期可达到 2.1s/5V。WDT 时钟源除了使用内部 WDT 振荡器输出外，还可以使用指令时钟（系统时钟 4 分频），只是在 HALT 时，WDT 会停止计数而失去保护功能；此时只能靠外部逻辑复位来重新启动系统。WDTS 的高四位及其第 3 位保留给用户定义标志来使用，程序员可以利用这些标志来指示某些特殊的状态。



看门狗定时器

如果系统运用在强干扰的环境中，建议选用内部 RC 振荡器(WDT 振荡器)，因为 HALT 模式会使系统时钟停止，看门狗也就失去了保护的功能。

在正常运行时，WDT 溢出会使系统复位并置位 TO 标志；但在 HALT 模式下，WDT 溢出只产生“热复位”，只有程序计数器 PC 和堆栈指针 SP 被复位。要清除 WDT 的值(包括 WDT 预分频器)可以有三种方法：外部复位(低电平输入到$\overline{\text{RES}}$端)、清除看门狗指令或 HALT 指令。清除看门狗指令有“CLR WDT”和“CLR WDT1”、“CLR WDT2”二组指令。这两组指令中，只能选择其中一组，由掩膜选项决定。如果“CLR WDT”被选择（即 CLR WDT 次数为 1），那么只要执行“CLR WDT” 指令就会清除 WDT。“CLR WDT1”和“CLR WDT2”被选择的情况下（即 CLR WDT 次数为 2），那么二条指令要交替使用才会清除 WDT，否则，WDT 会由于溢出而使系统复位。

WS2	WS1	WS0	分频率
0	0	0	1：1
0	0	1	1：2
0	1	0	1：4
0	1	1	1：8
1	0	0	1：16
1	0	1	1：32
1	1	0	1：64
1	1	1	1：128

WDTS（09H）寄存器

暂停模式

暂停模式是由 HALT 指令来实现的，暂停模式时系统状态如下：

- 系统振荡器停振，但 WDT 振荡器会继续振荡（如果选择 WDT 振荡器）。
- RAM 和寄存器内容保持不变。
- WDT 被清除并重新开始计数（如果 WDT 时钟来源为 WDT 振荡器）。
- 所有输入/输出口都保持其原有状态。
- 置位 PDF 标志，清除 TO 标志。

以下操作可以使系统离开暂停模式：外部复位、中断、PA 口下降沿信号或看门狗定时器溢出。其中，外部复位会使系统初始化，WDT 溢出则会发生“热复位”。通过检测 TO 和 PDF 标志，即可了解系统复位的原因。PDF 标志可由系统上电或执行“CLR WDT”指令清除，由 HALT 指令置位。TO 标志由 WDT 溢出置位，同时产生唤醒，只有程序计数器 PC 和堆栈指针 SP 复位，其它都保持其原有的状态。

PA 口唤醒和中断唤醒可作为正常运行的继续。PA 口的每一位都可以由掩膜选项设置为唤醒功能。如果是由输入/输出口唤醒，程序会从下一条指令开始运行。如果是由中断唤醒，可能会发生两种情况：如果中断禁止或中断允许但堆栈已满，程序将会从下一条指令开始运行；如果中断允许且堆栈未满，则会产生一般的中断响应。如果在进入 HALT 模式之前，中断请求标志位已被置“1”，则中断唤醒功能被禁止。

当发生唤醒，系统需要额外花费 $1024t_{SYS}$(系统时钟周期)的时间，才能重新正常运行，也就是说，唤醒之后会插入一个等待周期。如果唤醒是由中断产生的话，则实际中断子程序的执行会延迟一个以上的周期。如果唤醒导致下一条指令执行，那么在等待周期执行完成之后，会立即执行该指令。

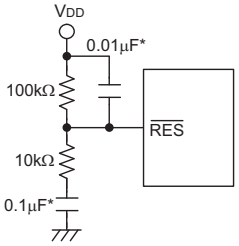
为减小功耗，在进入暂停模式之前，应小心处理所有的输入/输出口状态。

复位

总共有三种方法会产生初始复位：

- 正常运行时由$\overline{\text{RES}}$引脚发生复位。
- 在暂停模式由$\overline{\text{RES}}$引脚发生复位。
- 正常运行时由看门狗定时器溢出发生复位。

暂停模式中的看门狗定时器溢出与其它系统复位状况不同，因为看门狗定时器溢出会执行“热复位”，只有程序计数器 PC 和堆栈指针 SP 被复位，而系统其它部分都保持原有状态。在其它复位状态下，某些寄存器不会改变。在初始复位时，大部分寄存器会复位成初始的状态。通过检测 PDF 和 TO 标志，即可判断出各种不同的复位原因。



复位电路

注：“*”连线应该尽量靠近$\overline{\text{RES}}$引脚，以减小干扰影响

TO	PDF	复位原因
0	0	上电时$\overline{\text{RES}}$发生复位
u	u	正常运行时$\overline{\text{RES}}$发生复位
0	1	暂停模式下$\overline{\text{RES}}$发生复位
1	u	正常运行时 WDT 溢出
1	1	暂停模式下 WDT 溢出

注：“u”表示不变

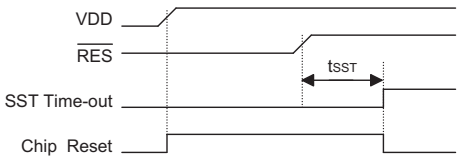
为了保证系统振荡器起振并稳定运行，系统复位（包括上电复位、WDT 溢出或由$\overline{\text{RES}}$端复位）或由暂停状态唤醒时，系统启动定时器（SST）提供了一个额外的延迟时间，共 1024 个系统时钟周期。

系统复位时，SST 会被加在复位延时中；由暂停模式唤醒也会加入 SST 延迟。

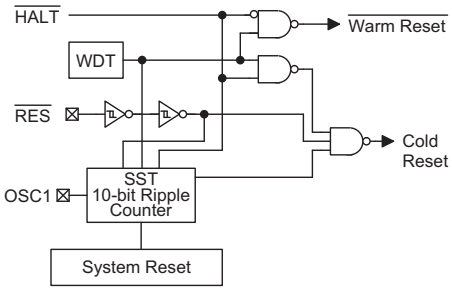
系统复位（包括上电复位、正常运行时 WDT 溢出或由$\overline{\text{RES}}$端复位）需要额外增加一个加载掩膜选项（Option）的时间。

系统复位时各功能单元的状态如下所示：

程序计数器	000H
中断	禁止
预分频器	清除
看门狗定时器	清除，复位后看门狗定时器开始计数
定时/计数器	关闭
输入/输出口	输入模式
堆栈指针	指向堆栈顶端



复位时序



复位电路结构

有关寄存器的状态如下：

寄存器	上电复位	WDT 溢出 （正常运行）	$\overline{\text{RES}}$ 复位 （正常运行）	$\overline{\text{RES}}$ 复位 （暂停模式）	WDT 溢出 （暂停模式）*
MP0	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
MP1	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
BP	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
Prgram Counter	000H	000H	000H	000H	000H
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	xuuu uuuu	xuuu uuuu	xuuu uuuu	xuuu uuuu
WDTS	0000 0111	0000 0111	0000 0111	0000 0111	uuuu uuuu
STATUS	--00 xxxx	--lu uuuu	--uu uuuu	--01 uuuu	--11 uuuu
INTC	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu
TMR0	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMR0C	00-0 1000	00-0 1000	00-0 1000	00-0 1000	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
PB	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PBC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PCC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PD	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PDC	1111 1111	1111 1111	1111 1111	1111 1111	uuuu uuuu
PG	---- ---1	---- ---1	---- ---1	---- ---1	---- ---u
PGC	---- ---1	---- ---1	---- ---1	---- ---1	---- ---u
EECR	1000 ----	1000 ----	1000 ----	1000 ----	uuuu ----

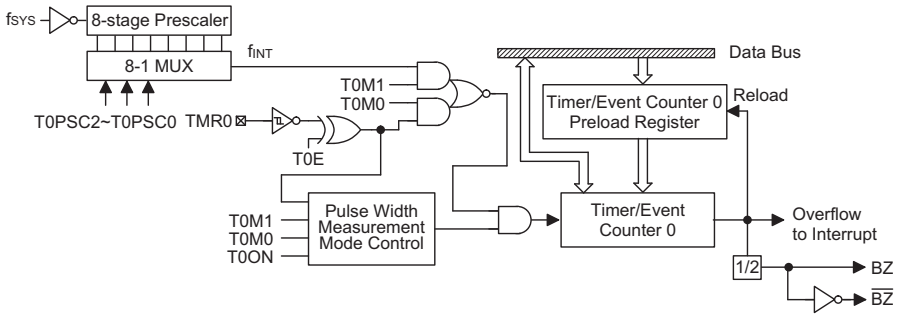
注：“*”表示“热复位”。
“U”表示不变化。
“×”表示不确定。

定时/计数器

HT48F50E 提供两个定时/计数器(TMR0, TMR1)。TMR0 为 8 位可编程向上计数的定时/计数器，时钟来源可以是外部信号输入或系统时钟；TMR1 为 16 位可编程向上计数的定时/计数器，时钟来源可以是外部信号输入或系统时钟四分频。

使用外部时钟输入，允许用户计数外部事件、测量时间间隔或脉冲宽度，或产生一个精确的时基信号。使用内部时钟用于产生精确的时基。

定时/计数器 0 使用外部时钟或内部时钟可以产生 PFD 信号，PFD 信号频率为：f_{INT} / [2×(256-N)]。



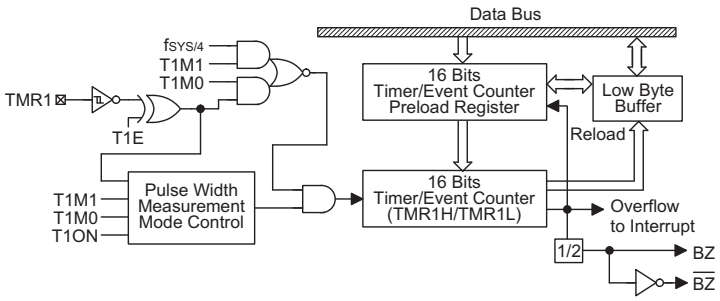
定时/计数器 0

有两个与定时/计数器 0 有关的寄存器，TMR0（[0DH]）和 TMR0C（[0EH]）。TMR0 寄存器有两个物理空间；写入 TMR0 会将初始值装入到定时/计数器 0 的预置寄存器中，而读 TMR0 则会取得定时/计数器 0 的内容。TMR0C 是定时/计数器 0 控制寄存器，用来定义定时/计数器 0 工作模式，计数功能打开或关闭、计数的触发沿。

位	符号	功能
0-2	T0PSC0~ T0PSC2	定义预分频器级数，T0PSC2，T0PSC1，T0PSC0= 000：f_{INT}= fsys/2 001：f_{INT}= fsys/4 010：f_{INT}= fsys/8 011：f_{INT}= fsys/16 100：f_{INT}= fsys/32 101：f_{INT}= fsys/64 110：f_{INT}= fsys/128 111：f_{INT}= fsys/256
3	T0E	定义定时/计数器 TMR0 的有效边沿触发。 在外部事件计数模式(T0M1,T0M0)=(0,1)： 1：下降沿计数； 0：上升沿计数 在脉冲宽度测量模式(T0M1,T0M0)=(1,1)： 1：上升沿开始计数，下降沿停止计数； 0：下降沿开始计数，上升沿停止计数
4	T0ON	打开/关闭定时/计数器 0(1=打开，0=关闭)
5	—	未用，读出为 0
6 7	T0M0 T0M1	定义工作模式(T0M1，T0M0) 01=外部事件计数模式(外部时钟) 10=定时模式(内部时钟) 11=脉冲宽度测量模式 00 =未用

TMR0C（0EH）寄存器

有三个寄存器与定时/计数器 1 相关联，即 TMR1H（[0FH]），TMR1L（[10H]）和 TMR1C（[11H]）。写入 TMR1L 只能将数据写入低字节缓冲器（8 bit）中，但若写入的 TMR1H 则可将数据和低字节缓冲器的内容写入 TMR1H 和 TMR1L 的预置寄存器中。每一次写 TMR1H 都会改变定时/计数器 1 预置寄存器的内容。若读取 TMR1H 则将锁存 TMR1H 的内容并将 TMR1L 传送至低字节缓冲器之中，以避免发生计时错误。然而，若读取 TMR1L，则只读回低字节缓冲器的内容。换言之，定时/计数器的低字节数据并不能直接读取。若欲读取该低字节的数据，必须先读取 TMR1H，以便将定时/计数器 1 的低字节数据传送至低字节缓冲器之中。TMR1C 是定时/计数器 1 控制寄存器，用来定义工作模式、计数功能打开或关闭、计数的触发沿。



定时/计数器 1

位	符号	功能
0-2	—	未用，读出为 0
3	T1E	定义定时/计数器 TMR1 的有效边沿触发。 在外部事件计数模式(T1M1,T1M0)=(0,1)： 1：下降沿计数； 0：上升沿计数 在脉冲宽度测量模式(T1M1,T1M0)=(1,1)： 1：上升沿开始计数，下降沿停止计数； 0：下降沿开始计数，上升沿停止计数
4	T1ON	打开/关闭定时/计数器 1(1=打开，0=关闭)
5	—	未用，读出为 0
6 7	T1M0 T1M1	定义工作模式(T1M1，T1M0) 01=外部事件计数模式(外部时钟) 10=定时模式(内部时钟) 11=脉冲宽度测量模式 00 =未用

TMR1C（11H）寄存器

T0M0，T0M1，T1M0，T1M1 用来定义定时/计数器的工作模式。外部事件计数模式是用来记录外部事件的，其时钟来源为外部引脚输入(TMR0/ TMR1)。定时器模式是一个常用模式，其时钟来源为内部时钟 f_{INT} 或指令时钟。脉宽测量模式可以测量外部引脚(TMR0/ TMR1)高/低电平的脉冲宽度，其时钟来源为内部时钟 f_{INT} 或指令时钟。

无论是定时模式还是外部事件计数模式，一旦开始计数，定时/计数器 0/1 会从寄存器当前值向上计到 0FFH/0FFFFH。一旦发生溢出，定时/计数器 0/1 会从预置寄存器中重新加载初值，并开始计数；同时置位中断请求标志（T0F/ T1F；INTC 的第 5/6 位）。

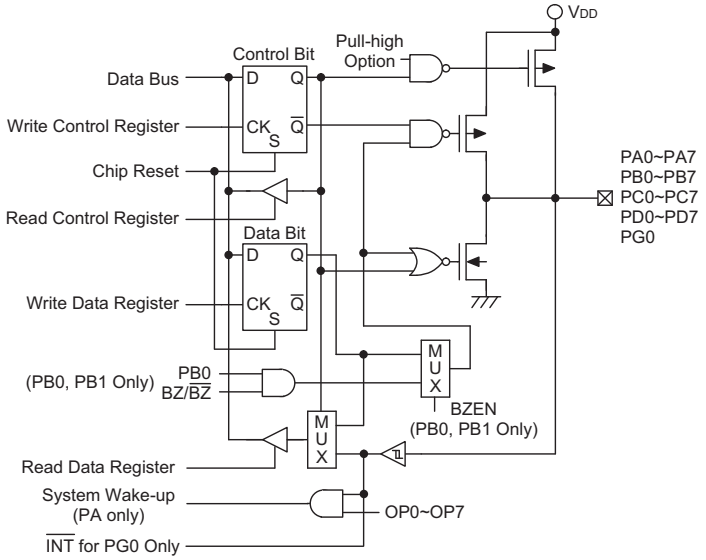
在脉宽测量模式，当 T0ON/T1ON 与 T0E/T1E 是 1 时，只要 TMR0/TMR1 引脚有一个上升沿信号（如果 T0E/T1E 是 0，则为下降沿信号），定时/计数器就会开始计数，直到 TMR0/TMR1 脚电平恢复，同时 T0ON/T1ON 被清零。测量的结果会保存在寄存器中，直到有新的测量开始。换句话说，一次只能测量一个脉冲宽度。重新置位 T0ON/T1ON 后，可以继续测量。注意，在该模式下，定时/计数器 0/1 是跳变触发而不是电平触发。当计数器溢出时，定时/计数器会从预置寄存器中重新加载初值，并置位中断请求标志，这与其它两种模式一样。要启动计数器，只要置位 ON（T0ON/T10N；TMR0C/ TMR1C 的第 4 位）。在脉宽测量模式下，T0ON/T1ON 在测量结束后会被自动清除；但在另外两种模式中，T0ON/T1ON 只能由指令来清除。定时/计数器溢出可以作为唤醒信号。不管是什么模式，只要写 0 到 ET0I/ET1I 即可禁止定时/计数器中断服务。

在定时/计数器停止计数时，写数据到定时/计数器的预置寄存器，同时会将该数据写入到定时/计数器。但如果在定时/计数器运行时这么做，数据只能写入到预置寄存器中，直到发生溢出时才会将数据从预置寄存器加载到定时/计数器寄存器。读取定时/计数器时，计数会被停止，以避免发生错误；计数停止会导致计数错误，程序员必须注意到这一点。

TMR0C 的第 0~2 位用来定义定时/计数器 0 内部时钟预分频级数。定时/计数器 0 的溢出信号可作为 PFD 输出。

输入/输出口

HT48F50E 有 33 个双向输入/输出口，记为 PA、PB、PC、PD 和 PG，其分别对应 RAM 地址[12H]、[14H]、[16H]、[18H]和[1EH]。所有端口都可以进行输入/输出操作。输入时，端口没有锁存功能，输入信号必须在 MOV A，[m]（m=12H、14H、16H、18H 或 1EH）指令的 T2 上升沿到来前准备好；输出时，端口有锁存功能，端口上的数据会保持不变直到执行下一个写入操作。



输入/输出口

每个输入/输出口都有一个控制寄存器（PAC，PBC，PCC，PDC，PGC），用来控制输入/输出状态。利用控制寄存器，可对 CMOS 输出、带或不带上拉电阻的斯密特触发输入通过软件动态地进行改变。作为输入时，对应的控制寄存器应设置为“1”。输入信号来源也取决于控制寄存器，如果控制寄存器的值为“1”，那么读取的是引脚状态；如控制寄存器的值为“0”，则读取的是内部锁存器的值。后者可能会在‘读-修改-写’指令中发生。

作为输出时，只能采用 CMOS 输出。控制寄存器对应 RAM 地址 13H、15H、17H、19H 和 1FH。

系统复位之后，这些输入/输出口会是高电平或浮空状态（由上拉电阻选项决定）。每一个输入/输出锁存位都能用“SET [m].i”或“CLR [m].i”指令置位或清除（m=12H、14H、16H、18H 或 1EH）。

有些指令会先输入数据，然后进行输出操作。例如：“SET [m].i”，“CLR [m].i”，“CPL [m]”，“CPLA[m]”这些指令会先将整个端口状态读入 CPU 中，接着执行所定义的运算（位操作），然后再将结果写入锁存器或累加器中。

PA 的每一个口都具有唤醒系统的能力。PG 的高 7 位没有实际的物理结构；读取这些位会返回“0”，而写入则是一个空操作。

所有的输入/输出口都有上拉电阻选项。一旦选择了上拉电阻选项，输入/输出口就加了上拉电阻。如果不选择上拉电阻，必须注意在输入模式下，输入/输出口会产生浮空状态。

PB0 和 PB1 分别与 BZ 和$\overline{BZ}$共用引脚。如果选择 BZ/$\overline{BZ}$功能，则 PB0/PB1 在输出模式时的输出信号将是由定时/计数器 0 的溢出信号产生的 PFD 信号。而在输入模式始终保持它的原来的功能。一旦选择 BZ/$\overline{BZ}$功能，蜂鸣器的输出信号只受 PB0 数据寄存器控制。

PB0/PB1 的输入/输出功能如下所示：

PB0 输入/输出	I	I	O	O	O	O	O	O	O	O
PB1 输入/输出	I	O	I	I	I	O	O	O	O	O
PB0 模式	×	×	C	B	B	C	B	B	B	B
PB1 模式	×	C	×	×	×	C	C	C	B	B
PB0 数据	×	×	D	0	1	D_0	0	1	0	1
PB1 数据	×	D	×	×	×	D_1	D	D	×	×
PB0 Pad 状态	I	I	D	0	B	D_0	0	B	0	B
PB1 Pad 状态	I	D	I	I	I	D_1	D	D	0	B

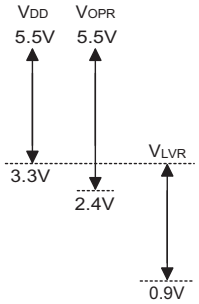
注： “I”输入， “O”输出； “D，D0，D1”数据，
“B”蜂鸣器选项，BZ 或 $\overline{BZ}$； “×”任意值
“C”CMOS 输出

PG0 和 $\overline{INT}$ 共用引脚。

为了避免在浮空状态下功耗太大，建议将未用的或没有连结到外部的输入/输出口由软件指令设置成输出引脚。

低电压复位

为了监控器件的工作电压，HT48F50E 提供低电压复位功能。如果器件的工作电压在 0.9V～V_{LVR} 之间，例如电池电压的变化，那么 LVR 会自动使器件产生内部复位。

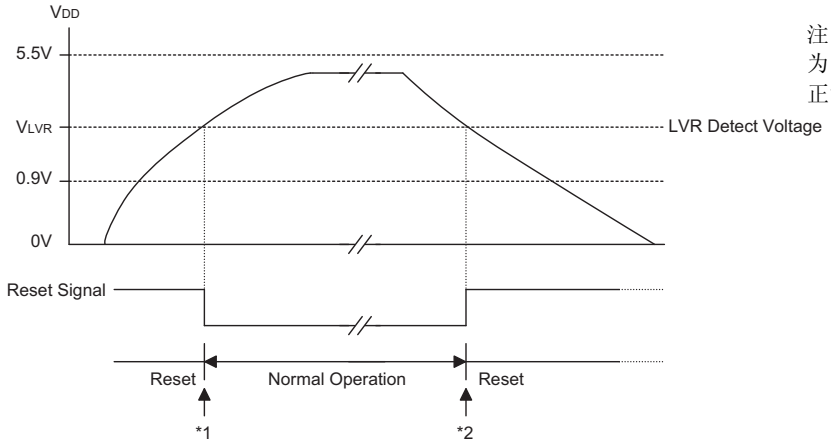


注：V_{OPR} 是在系统时钟为 4MHz 时，使得芯片正常运行的电压值

LVR 功能说明如下：

- 低电压（0.9V～V_{LVR}）的状态必须持续 1ms 以上。如果低电压的状态没有持续 1ms 以上，那么 LVR 会忽视它而不去执行复位功能。
- LVR 通过与外部 $\overline{RES}$ 信号的“或”功能来执行系统复位。

VDD 与 V_{LVR} 之间的关系如下所示：



低电压复位

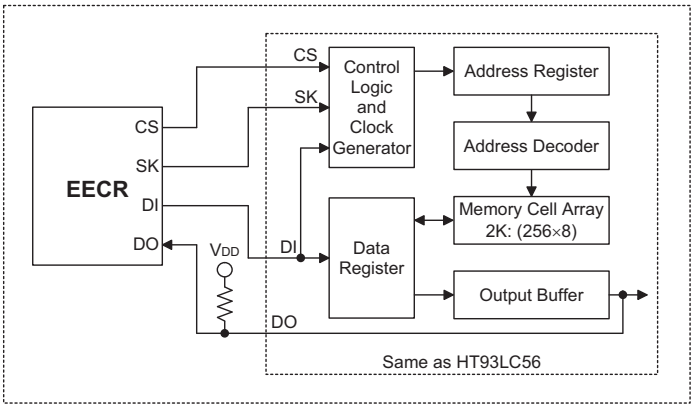
注：*1：要保证系统振荡器起振并稳定运行，在系统进入正常运行以前，SST 提供额外的 1024 个系统时钟周期的延迟。
*2：因为低电压状态必须保持 1ms 以上，因此 1ms 的延迟后进入复位模式。

EEPROM 数据存储器

EEPROM 数据存储器包含 256×8 位，单片机正常工作时可对其进行读/写操作。通过 EECR 寄存器（在 Bank 1 的[40H]）控制 EEPROM 数据存储器的读写，寄存器 EECR 只能使用 MP1 间接寻址。

位	标识	功能
0~3	—	未定义，读出为“0”
4	CS	EEPROM 数据存储器片选
5	SK	EEPROM 数据存储器串口时钟输入
6	DI	EEPROM 数据存储器串口数据输入
7	DO	EEPROM 数据存储器串口数据输出

EECR（40H）寄存器



EEPROM 数据存储器方框图

通过读写 EECR 寄存器，控制 EEPROM 的 CS、SK 和 DI/DO 信号，以软件的方式产生 EEPROM 的操作时序，进而达到读写 EEPROM 的目的。EEPROM 数据寄存器由 256×8 位组成。控制 EEPROM 数据存储器共有 7 条命令：READ、ERASE、WRITE、EWEN、EWDS、ERAL 和 WRAL，这些命令包含 12 个位：1 个起始位，2 个操作码位和 9 个地址位。

存取 EEPROM 之前，应该执行一个初始化程序。其详细过程如下：

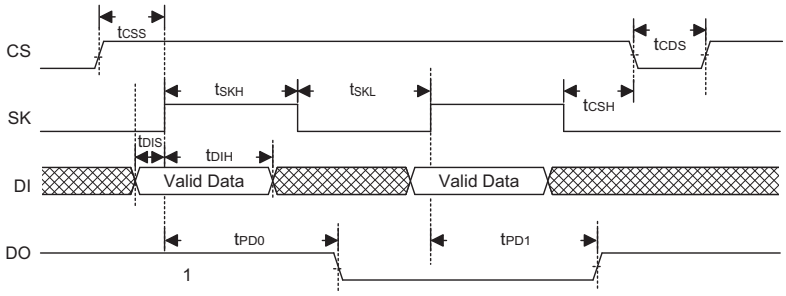
- 执行 EWEN 命令。
- 执行 WRITE 命令（用户需要为初始化程序保留一个未使用空间，其内容将在初始化程序执行后被更改）。
- 执行 EWDS 命令（此步骤可选。若无需立即写数据至 EEPROM，可执行此命令以防误操作）。

下列是汇编语言的范例程序，用户可将其放入单片机上电复位程序处。（注：关于 EWEN , EWDS 和 WRITE 详细程序请参考应用范例）。

```
mov     A, 01h
mov     BP, A          ; 设置指针指向 bank1
mov     A,40h
mov     MP1,A          ; 使 MP1 指向 EECR 地址
call    EWEN           ; 调用 EWEN 命令
mov     A, 0FFh
mov     EEADDR, A
mov     A, 55h
mov     EEDATA, A
call    WRITE          ; 调用 WRITE 命令(向地址 0FFh 中写入数据 55h)
call    EWDS           ; 调用 EWDS 命令(此步骤为可选)
```

通过控制 CS、SK 和 DI，这些指令被输入到 EEPROM 存储器。DI 线上的串行指令数据会在 SK 上升沿写入 EEPROM 数据存储器。执行 READ 命令，DO 作为数据输出；而执行 WRITE 或 ERASE 命令，DO 为 BUSY/READ 状态。当 DO 作为读数据有效或处于 BUSY/READY 状态，CS 必须置高；否则 DO 为高。命令发送完成后，CS 必须马上置为低。上电复位后，系统默认处于 EWDS 状态。执行 ERASE 或者 WRITE 指令之前，需要先执行 EWEN 指令。

下面是 7 条指令的时序图和功能描述：



EECR 交流电气特性 Ta=25℃

符号	参数	Vcc=5V±10%		Vcc=2.2V±10%		单位
		最小	最大	最小	最大	
f_{SK}	时钟频率	0	2	0	1	MHZ
t_{SKH}	SK 高电平时间	250	—	500	—	ns
t_{SKL}	SK 低电平时间	250	—	500	—	ns
t_{CSS}	CS 建立时间	50	—	100	—	ns
t_{CSH}	CS 保持时间	0	—	0	—	ns
t_{CDS}	CS 屏蔽时间	250	—	250	—	ns
t_{DIS}	DI 建立时间	100	—	200	—	ns
t_{DIH}	DI 保持时间	100	—	200	—	ns
t_{PD1}	DO 等待“1”时间	—	250	—	500	ns
t_{PD0}	DO 等待“0”时间	—	250	—	500	ns
t_{SV}	状态有效时间	—	250	—	250	ns
t_{HZ}	DO 释放时间	100	—	200	—	ns
t_{PR1}	每个 WORD 写周期	—	2	—	5	ms
t_{PR2}	每两个 WORD 写周期	—	10	—	10	ms

READ

READ 命令会将指定地址的数据送出到 DO 线上。在 SK 上升沿，DO 线上的数据会改变。8 位数据之前有一个逻辑“0”信号。无论 EWEN 或者 EWDS 命令执行与否，READ 命令都可以有效执行。一个字节数据被读取之后，EEPROM 内地址会自动增加，允许下一个连续的数据被读取，而不需要输入下一数据的地址。地址会循环累加直到 CS 从高电平变为低电平。

EWEN/EWDS

EWEN/EWDS 命令可以使能或禁止擦写动作有效执行。上电复位或者掉电状态，都会使芯片自动进入禁止擦写模式。执行 WRITE、ERASE、WRAL 或 ERAL 命令之前，必须先执行 EWEN 命令使能擦写；否则 ERASE/WRITE 命令无效。执行 EWEN 命令后，擦写使能直到芯片掉电或 EWDS 指令被执行。当处于禁止擦写状态时， EEPROM 存储器内数据无法被改写，这样 EEPROM 内数据处于写保护状态。

ERASE

在擦写使能模式下，ERASE 命令可以擦除指定地址的数据。ERASE 命令码和指定地址送出后，EEPROM 会在 CS 下降沿执行内部擦除动作。内部擦除动作时钟由 EEPROM 时钟发生器提供，并不需要 SK 信号。内部擦除动作期间，当 CS 为高电平，可以检测系统处于 BUSY/READY 状态。DO 线保持低电平直到擦除动作结束；当 DO 恢复到高电平，其它命令可以被执行。

WRITE

在擦写使能模式下，WRITE 命令可以将数据写入 EEPROM 数据存储器的指定地址。WRITE 命令码、指定地址和数据送出后，EEPROM 会在 CS 下降沿进入内部写周期。内部写周期时钟由 EEPROM 时钟发生器提供，并不需要 SK 信号。由于内部写周期包括先擦除再写入的过程，因此在写入数据前不需要执行擦除命令。EEPROM 执行内部写周期，当 CS 为高电平，可以检测系统处于 BUSY/READY 状态。DO 线保持低电平直到内部写周期结束；当 DO 恢复到高电平，其它命令可以被执行。

ERAL

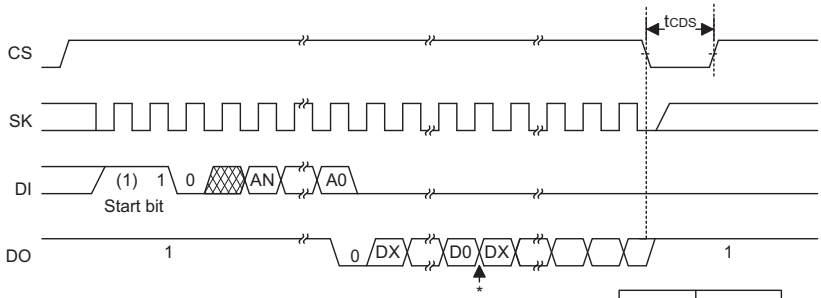
在擦写使能模式下，ERAL 命令可以将所有 256×8 存储器单元擦除为逻辑“1”状态。ERAL 命令送出后，EEPROM 会在 CS 下降沿执行内部擦除动作。内部擦除动作时钟由 EEPROM 时钟发生器提供，并不需要 SK 信号。当 EEPROM 执行内部擦除动作时，若 CS 为高电平，可以检测系统处于 BUSY/READY 状态。DO 线保持低电平直到擦除动作结束；当 DO 恢复到高电平，其它命令可以被执行。

WRAL

在擦写使能模式下，WRAL 命令可以将数据写入所有 256×8 存储器单元。WRAL 命令送出后，EEPROM 会在 CS 下降沿进入内部写周期。内部写周期时钟由 EEPROM 时钟发生器提供，并不需要 SK 信号。EEPROM 执行内部写周期，当 CS 为高电平，可以检测系统处于 BUSY/READY 状态。DO 线保持低电平直到内部写周期结束；当 DO 恢复到高电平，其它命令可以被执行。

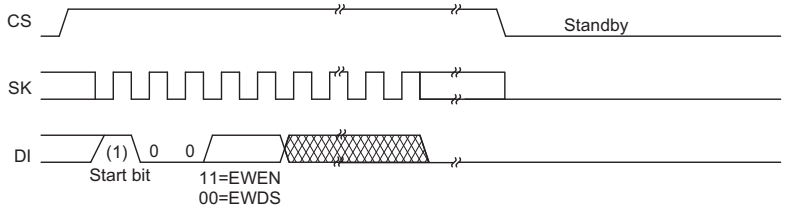
EECR 控制时序图

- **READ**

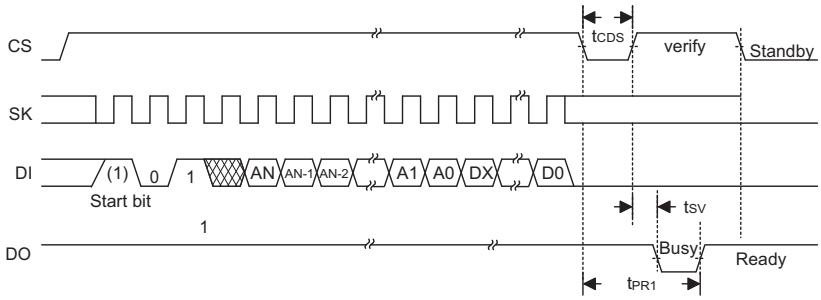


Mode	(X8)
AN	A7
DX	D7

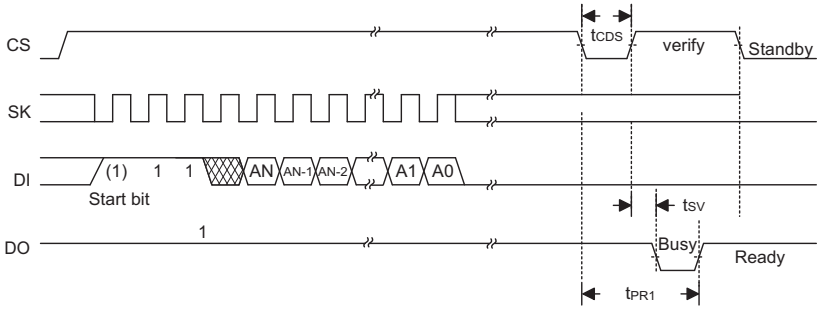
- **EWEN/EWDS**



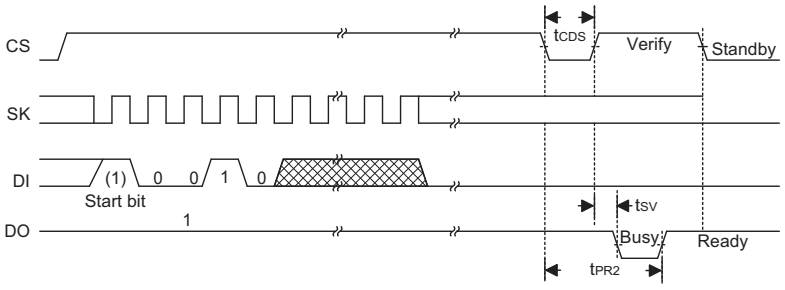
- **WRITE**



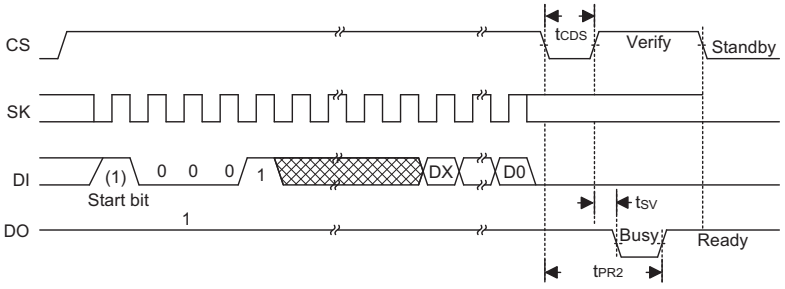
- **ERASE**



- **ERAL**



- **WRAL**



EEPROM 数据存储器指令设置概要

指令	描述	起始位	指令码	地址	数据
READ	Read Data	1	10	X，A7~A0	D7~D0
ERASE	Erase Data	1	11	X，A7~A0	——
WRITE	Write Data	1	01	X，A7~A0	D7~D0
EWEN	Erase/Write Enable	1	00	11XXXXXX	——
EWDS	Erase/Write Disable	1	00	00XXXXXX	——
ERAL	Erase All	1	00	10XXXXXX	——
WRAL	Write All	1	00	01XXXXXX	D7~D0

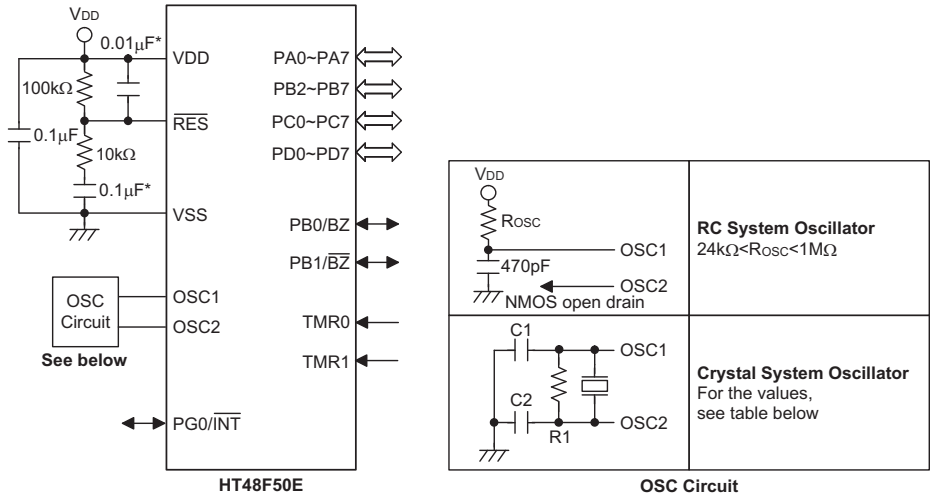
注：“X”表示任意

掩膜选项

下表列出了 HT48F50E 的所有掩膜选项。所有选项必须正确定义，以保证统正常运行。

编号	选　　项
1	WDT 时钟源：WDT 振荡、f_{SYS}/4 或关闭
2	清除看门狗指令条数：1 条或 2 条
3	PA 唤醒功能：打开/关闭
4	PA CMOS/斯密特触发输入
5	PA，PB，PC，PD，PG 上拉电阻：有或无上拉电阻
6	BZ/$\overline{BZ}$ 功能：打开/关闭
7	LVR 功能：打开/关闭
8	系统振荡：RC 振荡/晶体振荡
9	BZ/$\overline{BZ}$ 源：TMR0/TMR1

应用电路



注： 复位电路的电阻和电容选取的原则是使 VDD 保持稳定并在$\overline{RES}$置为高以前把工作电压保持在允许的范围内。

“*”为了避免噪声干扰，连接$\overline{RES}$引脚的线请尽可能地短。

下表所示为不同的晶振选择 R1、C1、C2（仅供参考）

晶体振荡器或谐振器	C1，C2	R1
4MHz 晶振	0pF	10KΩ
4MHz 谐振器	10pF	12KΩ
3.58MHz 晶振	0pF	10KΩ
3.58MHz 谐振器	25pF	10KΩ
2MHz 晶振 & 谐振器	25pF	10KΩ
1MHz 晶振	35pF	27KΩ
480KHz 谐振器	300pF	9.1KΩ
455KHz 谐振器	300pF	10KΩ
429KHz 谐振器	300pF	10KΩ
电阻 R1 的作用是在低电压状态下，确保晶振被关闭。这里的低电压，是指低于 MCU 正常工作的最低电压。注意的是，当启动了 LVR 功能，R1 可以不接。		

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。

指令定义

ADC A, [m] Add data memory and carry to the accumulator
说明： 将指定的数据存储器、累加器以及进位标志位的内容相加，结果存放到累加器。
运算过程： ACC←ACC+[m]+C
影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

ADCM A, [m] Add the accumulator and carry to the accumulator
说明： 将指定的数据存储器、累加器和进位标志位的内容相加，结果存放到指定的数据存储器。
运算过程： [m]←ACC+[m]+C
影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

ADD A, [m] Add data memory to the accumulator
说明： 将指定的数据存储器和累加器内容相加，结果存放到累加器。
运算过程： ACC←ACC+[m]
影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

ADD A, x Add immediate data to the accumulator
说明： 将累加器和立即数内容相加，结果存放到累加器。
运算过程： ACC←ACC+x
影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

ADDM A, [m] Add the accumulator to the data memory
说明： 将指定的数据存储器和累加器内容相加，结果存放到指定的数据存储器。
运算过程： [m]←ACC+[m]
影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

AND A, [m] Logical AND accumulator with data memory
说明： 将累加器中的数据和指定数据存储器内容做逻辑与，结果存放到累加器。
运算过程： ACC←ACC “AND” [m]
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

AND A, x Logical AND immediate data to the accumulator

说明： 将累加器中的数据和立即数内容做逻辑与，结果存放到累加器。
运算过程： ACC←ACC “AND” x
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

ANDM A, [m] Logical AND data memory with the accumulator
说明： 将指定数据存储器内容和累加器中的数据做逻辑与，结果存放到数据存储器。
运算过程： [m]←ACC “AND” [m]
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

CALL addr Subroutine call
说明： 无条件的调用指定地址的子程序，此时程序计数器先加 1 获得下一个要执行的指令地址并压入堆栈，接着载入指定地址并从新地址执行程序。
运算过程： Stack←Program Counter+1
Program Counter ← addr
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

CLR [m] Clear data memory
说明： 将指定数据存储器的内容清零。
运算过程： [m] ←00H
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

CLR [m] . i Clear bit of data memory
说明： 将指定数据存储器的 I 位内容清零。
运算过程： [m].i ←0
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

CLR WDT Clear Watchdog Timer
说明： WDT 计数器、暂停标志位(PDF)和看门狗溢出标志位(TO)清零。
运算过程： WDT ← 00H
PDF & TO ← 0
影响标志位

TO	PDF	OV	Z	AC	C
0	0	—	—	—	—

CLR WDT1 Preclear Watchdog Timer
说明： 必须配合 CLR WDT2 一起使用清除 WDT 计时器。PDF 和 TO 标志位也被清 0。当程序仅执行 CLR WDT1，而没有执行 CLR WDT2 时，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	OV	Z	AC	C
0*	0*	—	—	—	—

CLR WDT2 Preclear Watchdog Timer

说明： 必须配合 CLR WDT1 一起使用清除 WDT 计时器。PDF 和 TO 标志位也被清 0。当程序仅执行 CLR WDT2，而没有执行 CLR WDT1 时，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	OV	Z	AC	C
0*	0*	—	—	—	—

CPL [m] Complement data memory

说明： 将指定数据存储器中的每一位取逻辑反，相当于从 1 变 0 或从 0 变 1。

运算过程： [m] ← [$\overline{m}$]

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

CPLA [m] Complement data memory

说明： 将指定数据存储器中的每一位取逻辑反，相当于从 1 变 0 或从 0 变 1，结果被存放回累加器且数据寄存器的内容保持不变。

运算过程： ACC ← [$\overline{m}$]

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

DAA [m] Decimal-Adjust accumulator for addition

说明 将累加器中的内容转换为 BCD（二进制转成十进制）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”，并且内部进位标志 AC1= $\overline{AC}$，即 AC 求反；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”再加 AC1，并把 C 置位；否则 BCD 调整就执行对原值加 AC1，C 的值保持不变。结果存放到数据存储器中，只有进位标志位(C)受影响。

操作 如果 ACC.3~ACC.0 > 9 或 AC=1
那么 [m].3~[m].0 ←(ACC.3~ACC.0)+6，AC1= $\overline{AC}$
否则 [m].3~[m].0 ←(ACC.3~ACC.0)，AC1=0
并且
如果 ACC.7~ACC.4+AC1 > 9 或 C=1
那么 [m].7~[m].4 ←(ACC.7~ACC.4)+6+ AC1，C=1
否则 [m].7~[m].4 ←(ACC.7~ACC.4)+ AC1，C=C

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

DEC [m] Decrement data memory

说明： 将指定数据存储器的内容减 1。

运算过程： [m] ← [m]-1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

DECA [m] Decrement data memory and place result in the accumulator
说明： 将指定数据存储器的内容减 1，把结果存放回累加器并保持指定数据存储器的内容不变。
运算过程： ACC ← [m]-1
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

HALT Enter power down mode
说明： 此指令终止程序执行并关掉系统时钟，RAM 和寄存器的内容保持原状态，WDT 计数器和预分频器被清“0”，暂停标志位(PDF)被置位 1， WDT 溢出标志位(TO)被清为 0。
运算过程： PC ← PC+1
PDF ← 1
TO ← 0
影响标志位

TO	PDF	OV	Z	AC	C
0	1	—	—	—	—

INC [m] Increment data memory
说明： 将指定数据存储器的内容加 1。
运算过程： [m] ← [m]+1
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

INCA [m] Increment data memory and place result in the accumulator
说明： 将指定数据存储器的内容加 1，结果存放回累加器并保持指定的数据存储器内容不变。
运算过程： ACC ← [m]+1
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

JMP addr Directly jump
说明： 程序计数器的内容无条件地由被指定的地址取代，程序由新的地址继续执行。
运算过程： PC ← addr
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

MOV A, [m] Move data memory to the accumulator
说明： 将指定数据存储器的内容复制到累加器中。
运算过程： ACC ← [m]
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

MOV A, x Move immediate data to the accumulator
说明： 将 8 位立即数载入累加器中。
运算过程： ACC ← x
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

MOV [m], A Move the accumulator data to memory
说明： 将累加器的内容复制到指定的数据存储器内。
运算过程： [m] ← ACC
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

NOP No operation
说明： 空操作，接下来顺序执行下一条指令。
运算过程： PC ← PC+1
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

OR A, [m] Logical OR accumulator with data memory
说明： 将累加器中的数据和指定的数据存储器内容逻辑或，结果存放到累加器。
运算过程： ACC←ACC “OR” [m]
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

OR A, x Logical OR immediate data to the accumulator
说明： 将累加器中的数据和立即数逻辑或，结果存放到累加器。
运算过程： ACC←ACC “OR” x
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

ORM A, [m] Logical OR data memory with accumulator
说明： 将存在指定数据存储器中的数据和累加器逻辑或，结果放到数据存储器。
运算过程： [m]←ACC “OR” [m]
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

RET Return from subroutine
说明： 将堆栈寄存器中的程序计数器值恢复，是一个两周期指令。
运算过程： PC ← Stack
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RET A, x Return and place immediate data in the accumulator
说明： 程序返回，并将立即数送入累加器。
运算过程： PC ← Stack
ACC ← x
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RETI Return from interrupt
说明： 将堆栈寄存器中的程序计数器值恢复且中断功能通过设置 EMI 位重新使能。EMI 是控制中断使能的主控制位。
运算过程： PC ← Stack
EMI ← 1
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RL [m] Rotate data memory left
说明： 将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位。
运算过程： [m].0 ←[m].7, [m].(i+1) ←[m].i; (i=0~6)
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RLA [m] Rotate data memory left and place result in the accumulator
说明： 将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位，结果送到累加器，而指定数据存储器的内容保持不变。
运算过程： ACC.0 ←[m].7, ACC.(i+1) ←[m].i; (i=0~6)
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RLC [m] Rotate data memory left through carry
说明： 将指定数据存储器的内容连同进位标志左移 1 位，第 7 位取代进位标志且原本的进位标志移到第 0 位。
运算过程： [m].(i+1) ←[m].i; (i=0~6)
[m].0 ←C
C ← [m].7
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

RLCA [m] Rotate left through carry and place result in the accumulator
说明： 将指定数据存储器的内容连同进位标志左移 1 位，第 7 位取代进位标志且原本的进位标志移到第 0 位,移位结果送回累加器，但是指定数据寄存器的内容保持不变。
运算过程： ACC.(i+1) ←[m].i; (i=0~6)
ACC.0 ←C
C ← [m].7
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

RR [m] Rotate data memory right

说明： 将指定数据存储器的内容循环右移 1 位且第 0 位移到第 7 位。

运算过程： [m].7 ←[m].0, [m].i ←[m].(i+1)； (i=0~6)

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RRA [m] Rotate right and place result in the accumulator

说明： 将指定数据存储器的内容循环右移 1 位，第 0 位移到第 7 位，移位结果存放到累加器，而指定数据存储器的内容保持不变。

运算过程： ACC.7 ←[m].0, ACC.i ←[m].(i+1)；(i=0~6)

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RRC [m] Rotate data memory right through carry

说明： 将指定数据存储器的内容连同进位标志右移 1 位，第 0 位取代进位标志且原本的进位标志移到第 7 位。

运算过程： [m].i ←[m].(i+1)； (i=0~6)
[m]. 7 ←C
C ← [m].0

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

RRCA [m] Rotate right through carry and place result in the accumulator

说明： 将指定数据存储器的内容连同进位标志右移 1 位，第 0 位取代进位标志且原本的进位标志移到第 7 位,移位结果送回累加器，但是指定数据寄存器的内容保持不变。

运算过程： ACC.i ←[m].(i+1)； (i=0~6)
ACC.7 ←C
C ← [m].0

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

SBC A,[m] Subtract data memory and carry from the accumulator

说明： 将累加器减去指定数据存储器的内容以及进位标志的反，结果存放到累加器。

运算过程： ACC←ACC+[$\overline{m}$]+ C

影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SBCM A,[m] Subtract data memory and carry from the accumulator

说明： 将累加器减去指定数据存储器的内容以及进位标志的反，结果存放到数据存储器。

运算过程： [m]←ACC+[$\overline{m}$]+C

影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SDZ [m] Skip if decrement data memory is 0

说明： 将指定的数据存储器的内容减 1，判断是否为 0，若为 0 则跳过下一条指令，即如果结果为 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(两个指令周期)。否则执行下一条指令(一个指令周期)。

运算过程： 如果[m]-1=0，跳过下一条指令执行再下一条。

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SDZA [m] Decrement data memory and place result in ACC,skip if 0

说明： 将指定数据存储器的内容减 1，判断是否为 0，如果为 0 则跳过下一条指令，此结果会被存放到累加器，但是指定数据存储器的内容不变，由于取得下一个指令时会要求插入一个空指令周期，所以此指令为两个周期的指令。否则执行下一条指令(一个指令周期)。

运算过程： 如果[m]-1=0，跳过下一条指令执行。
ACC ←([m]-1)

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SET [m] Set data memory

说明： 将指定数据存储器的每一位置为 1。

运算过程： [m] ← FFH

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SET [m]. i Set bit of data memory

说明： 将指定数据存储器的第 i 位置为 1。

运算过程： [m].i ←1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SIZ [m] Skip if increment data memory is 0

说明： 将指定的数据存储器的内容加 1，判断是否为 0，若为 0 则跳过下一条指令，由于取得下一个指令时会要求插入一个空指令周期，所以此指令为两个周期的指令。否则执行下一条指令(一个指令周期)。

运算过程： 如果 ([m]+1=0)，跳过下一行指令；[m] ←[m]+1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SIZA Increment data memory and place result in ACC,skip if 0

说明： 将指定数据存储器的内容加 1，判断是否为 0，如果为 0 则跳过下一条指令，此结果会被存放到累加器，但是指定数据存储器的内容不变，由于取得下一个指令时会要求插入一个空指令周期，所以此指令为 2 个周期的指令。否则执行下一条指令(一个指令周期)。

运算过程： 如果[m]+1=0，跳过下一行指令；ACC ←([m]+1)

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SNZ [m]. i Skip if bit I of the data memory is not 0

说明： 判断指定的数据存储器的第 i 位，若不为 0，则程序计数器再加 1，跳过下一行指令，由于取得下一个指令时会要求插入一个空指令周期，所以此指令为两个周期的指令。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

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SUB A, [m] Subtract data memory from the accumulator

说明： 将累加器的内容减去指定的数据存储器的数据，把结果存放到累加器。

运算过程： ACC←ACC+[$\overline{m}$]+1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SUB A, x Subtract immediate data from the accumulator

说明： 将累加器的内容减去立即数，结果存放到累加器。

运算过程： ACC←ACC+$\overline{x}$+1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SUBM A, [m] Subtract data memory from the accumulator

说明： 将累加器的内容减去指定数据存储器的数据，结果存放到指定的数据存储器。

运算过程： [m]←ACC+[$\overline{m}$]+1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SWAP [m] Swap nibbles within the data memory

说明： 将指定数据存储器的低 4 位和高 4 位互相交换。

运算过程： [m].7~[m].4 ↔[m].3~[m].0

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SWAPA [m] Swap data memory and place result in the accumulator

说明： 将指定数据存储器的低 4 位和高 4 位互相交换，再将结果存放到累加器且指定数据寄存器的数据保持不变。

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

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SZ [m] Skip if data memory is 0

说明： 判断指定数据存储器的内容是否为 0，若为 0 则跳过下一条指令，由于取得下一个指令时会要求插入一个空指令周期，所以此指令为两个周期的指令。否则执行下一条指令(一个指令周期)。

运算过程： 如果 [m] = 0, 跳过下一条指令。

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SZA [m] Move data memory to ACC,skip if 0

说明： 将指定数据存储器的内容复制到累加器。并判断指定数据存储器的内容是否为 0，若为 0 则跳过下一条指令，由于取得下一个指令时会要求插入一个空指令周期，所以此指令为两个周期的指令。否则执行下一条指令(一个指令周期)。

运算过程： 如果[m] = 0，跳过下一条指令。

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SZ [m]. i Skip if bit I of the data memory is 0

说明： 判断指定数据存储器的第 i 位是否为 0，若为 0 则跳过下一条指令，由于取得下一个指令时会要求插入一个空指令周期，所以此指令为两个周期的指令。否则执行下一条指令(一个指令周期)。

运算过程： 如果 [m].i = 0，跳过下一条指令。

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

TABRDC [m] Move the ROM code(current page) to TBLH and data memory

说明： 将表格指针 TBLP 所指的程序代码低字节（当前页）移至指定的数据存储器且将高字节移至 TBLH。

运算过程： [m] ←程序代码（低字节）
TBLH←程序代码（高字节）

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

TABRDL [m] Move the ROM code(last page) to TBLH and data memory
说明： 将表格指针 TBLP 所指的程序代码低字节（最后一页）移至指定的数据存储器且将高字节移至 TBLH。
运算过程： [m] ←程序代码（低字节）
TBLH←程序代码（高字节）
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

XOR A, [m] Logical XOR accumulator with data memory
说明： 将累加器的数据和指定的数据存储器内容逻辑异或，结果存放到累加器。
运算过程： ACC←ACC "XOR" [m]
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

XORM A, [m] Logical XOR data memory with accumulator
说明： 将累加器的数据和指定的数据存储器内容逻辑异或，结果放到数据存储器。
运算过程： [m]←ACC "XOR" [m]
影响标志位

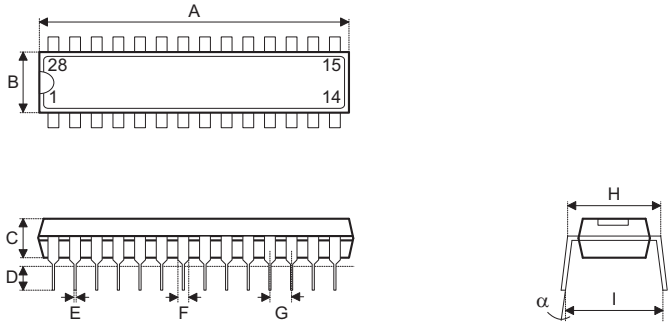
TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

XOR A, x Logical XOR immediate data to the accumulator
说明： 将累加器的数据与立即数逻辑异或，结果存放到累加器。0 标志位受影响。
运算过程： ACC←ACC "XOR" x
影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

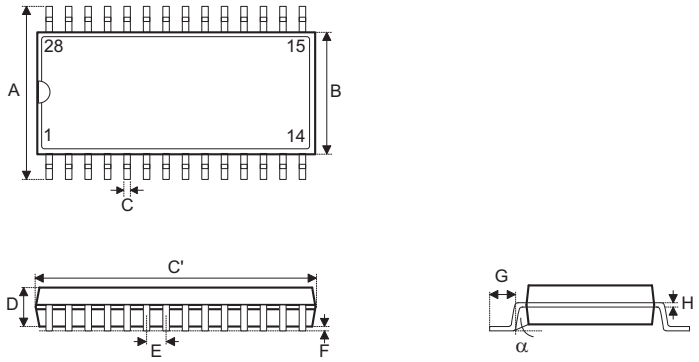
封装信息：

28-pin SKDIP (300mil)外形尺寸



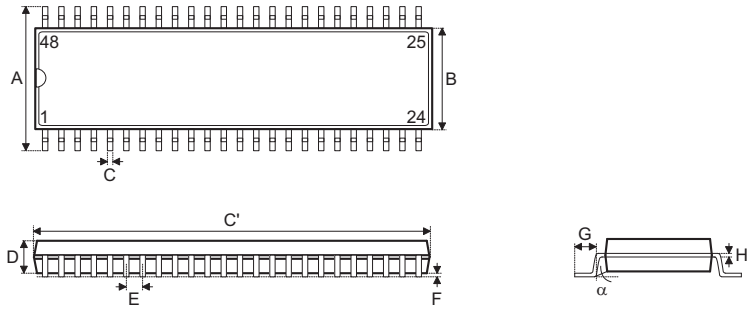
标号	尺寸（mil）		
	最小	典型	最大
A	1375	--	1395
B	278	--	298
C	125	--	135
D	125	--	145
E	16	--	20
F	50	--	70
G	--	100	--
H	295	--	315
I	330	--	375
α	0°	--	15°

28-pin SOP (300mil)外形尺寸



标号	尺寸（mil）		
	最小	典型	最大
A	394	--	419
B	290	--	300
C	14	--	20
C′	697	--	713
D	92	--	104
E	--	50	--
F	4	--	--
G	32	--	38
H	4	--	12
α	0°	--	10°

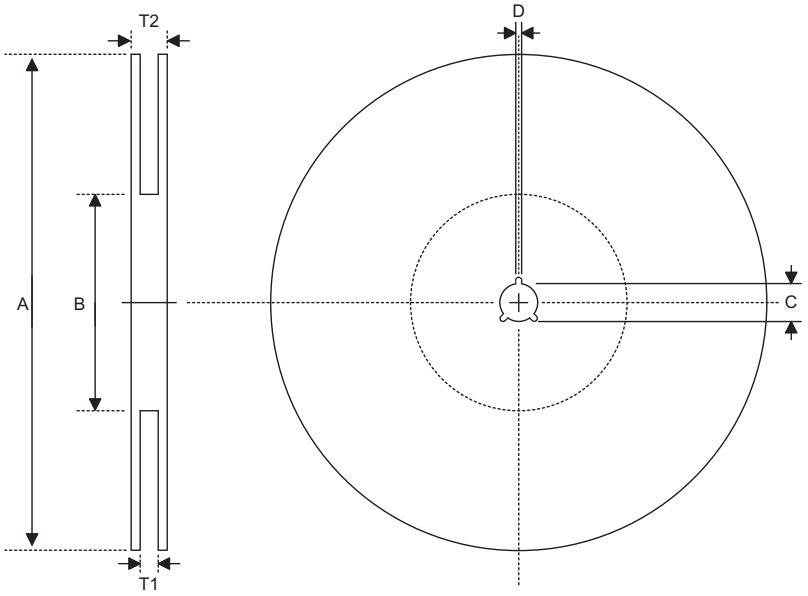
48-pin SOP (300mil)外形尺寸



标号	尺寸（mil）		
	最小	典型	最大
A	395	--	420
B	291	--	299
C	8	--	12
C′	613	--	637
D	85	--	99
E	--	25	--
F	4	--	10
G	25	--	35
H	4	--	12
α	0°	--	8°

包装带和卷轴规格：

卷轴尺寸：



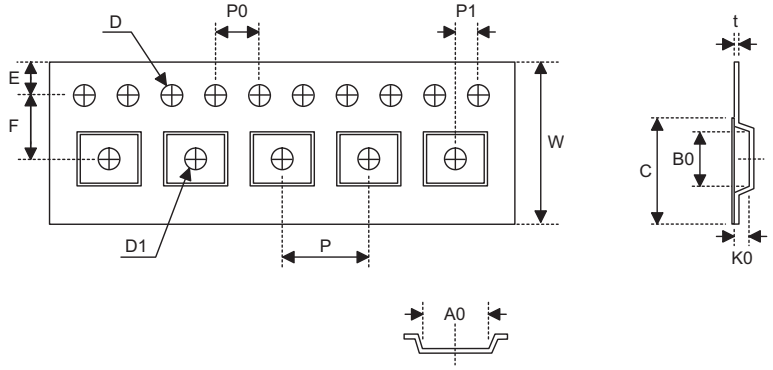
SOP 28W(300mil)

标号	描述	尺寸(mm)
A	卷轴外圈直径	330±1.0
B	卷轴内圈直径	62±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

SSOP 48W

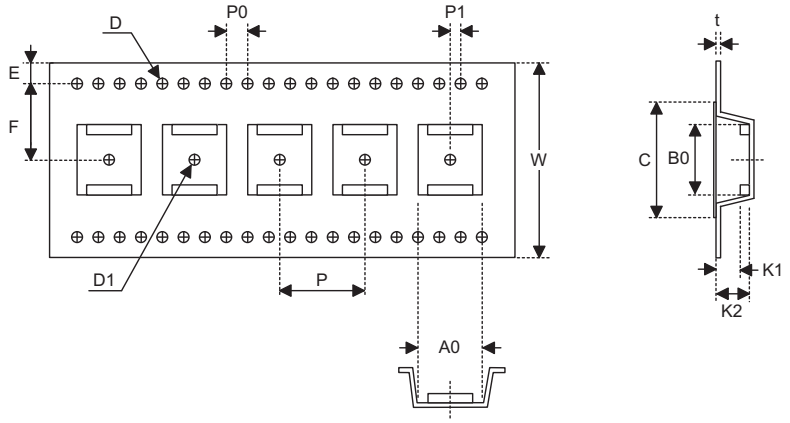
标号	描述	尺寸(mm)
A	卷轴外圈直径	330±1.0
B	卷轴内圈直径	100±0.1
C	轴心直径	13.0+0.5 -0.2
D	缝宽	2.0±0.5
T1	轮缘宽	32.2+0.3 -0.2
T2	卷轴宽	38.2±0.2

运输带尺寸：



SOP 28W(300mil)

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	1.5+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	10.85±0.1
B0	空穴宽	18.34±0.1
K0	空穴深	2.97±0.1
t	传输带厚度	0.35±0.01
C	覆盖带宽度	21.3



SSOP 48W

标号	描述	尺寸(mm)
W	运输带宽	32.0±0.3
P	空穴间距	16.0±0.1
E	穿孔位置	1.75±0.1
F	空穴至穿孔距离（宽度）	14.2±0.1
D	穿孔直径	最小 2.0
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	12.0±0.1
B0	空穴宽	16.20±0.1
K1	空穴深	2.4±0.1
K2	空穴深	3.2±0.1
t	传输带厚度	0.35±0.05
C	覆盖带宽度	25.5

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 889 号 2 号楼 7 楼 200233
电话: 021-6485-5560
传真: 021-6485-0313
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
ISDN: 0755-8615-6181

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

Holmate Semiconductor, 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.holmate.com

Copyright © 2007 by HOLTEK SEMICONDUCTOR INC.

使用指南中所出现的信息在出版当时相信是正确的，然而盛群对于说明书的使用不负任何责任。文中提到的应用目的仅仅是用来做说明，盛群不保证或表示这些没有进一步修改的应用将是适当的，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