



低电压 R-F 型 8-Bit 单片机
HT47C07L/HT47C08L

版本 : V1.10 日期 : 2014-12-23

www.holtek.com

目 录

特性	4
概述	4
选型表	5
方框图	5
引脚图	6
引脚说明	7
极限参数	9
直流电气特性	9
交流电气特性	10
上电复位特性	10
系统结构	11
指令执行时序	11
程序计数器 – PC	11
堆栈寄存器 – STACK	12
算术逻辑单元 – ALU	12
程序存储器	13
数据存储器 – RAM	14
特殊功能寄存器	16
间接寻址寄存器 – IAR0, IAR1	16
累加器 – ACC	16
状态寄存器 – STATUS	16
振荡电路	17
看门狗定时器 – WDT	17
多功能计时器	18
时基 – TB	18
复位和初始化	19
输入 / 输出	22
输入 / 输出寄存器列表	23
定时 / 计数器	25
RC 型 A/D 转换器	27
暂停模式 – HALT	35
中断	35
LCD 液晶显示存储器	37
液晶显示驱动输出	38
低电压检测 – LVD	40
掩膜选择	40

应用电路	41
指令集	42
简介	42
指令周期	42
数据的传送	42
算术运算	42
逻辑和移位运算	42
分支和控制转换	43
位运算	43
查表运算	43
其它运算	43
指令集概要	44
惯例	44
指令定义	47
封装信息	59
48-pin LQFP (7mm×7mm) 外形尺寸	60

特性

- 工作电压：1.2V~2.2V
- 多达 21 个双向输入 / 输出口
- 内置 32kHz~128kHz RC 振荡器
- 多达两个 RC 型 A/D 转换通道
- 看门狗定时器
- 多达 2K×16 程序存储器 ROM
- 多达 96×8 数据存储器 RAM
- 时基功能
- 一组蜂鸣器输出 (BZ, $\overline{\text{BZ}}$)
- 一个 EL 载波输出
- 一个多达 21×3 段的液晶显示驱动电路
- HALT 和唤醒功能可降低功耗
- 4 层硬件堆栈
- 低电压检测功能
- 位操作指令
- 查表指令，表格内容字长 16 位
- 当系统时钟为 128kHz 时，指令周期为 31μs
- 63 条指令
- 指令执行时间为 1 或 2 个指令周期
- 封装类型：48-pin LQFP

概述

HT47C07L/HT47C08L 是 8 位高性能精简指令集单片机。专门为有模拟信号输入（如传感器信号）的产品而设计。单一周期指令和流水线结构使它适用于高速应用。

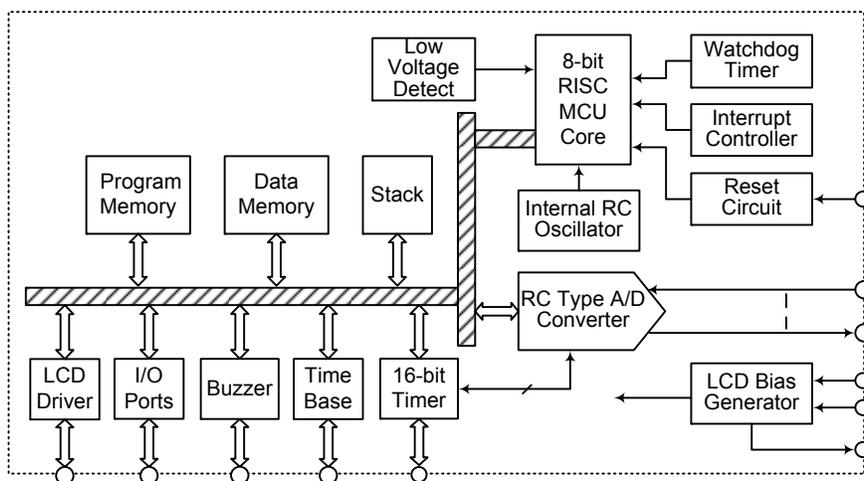
低功耗、I/O 使用灵活、定时 / 计数器、振荡类型选择、RC 型 A/D 转换器、LCD 驱动、暂停和唤醒功能，使这款单片机可以广泛应用于电阻到频率转换的应用，例如传感器信号处理、遥控测量，特别是当作临床体温计单片机使用。

选型表

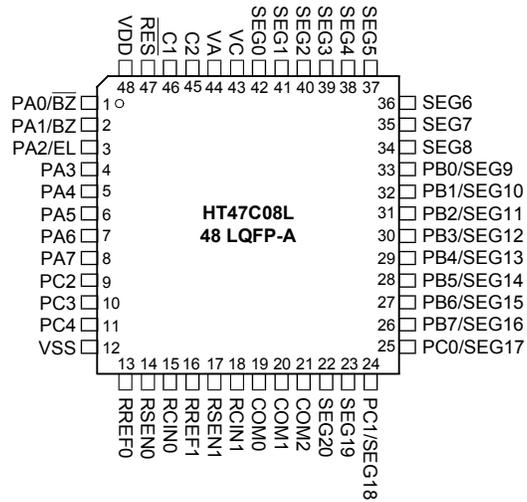
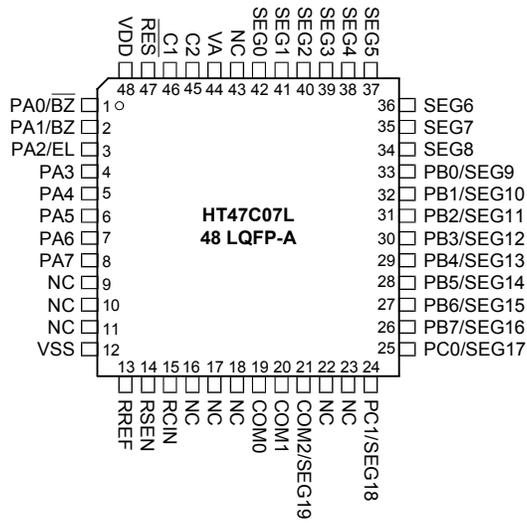
对此系列的芯片而言，大多数的特性参数都是一样的。主要差异在于程序存储器的容量，I/O 数量，RC 型 A/D 转换通道数和 LCD 驱动输出数。下表列出了各单片机的主要特性。

型号	ROM	RAM	I/O	EL 载波输出	RC 型 A/D	16-bit Timer	LCD 驱动	时基	堆栈	封装形式
HT47C07L	1K×16	48×8	18	1	1	1	19×3 或 20×2	1	4	48 LQFP
HT47C08L	2K×16	96×8	21	1	2	1	21×3	1	4	48 LQFP

方框图



引脚图



引脚说明

HT47C07L

引脚名称	输入 / 输出	掩膜选项	功能说明
RES	输入	—	斯密特触发复位输入，低电平有效。
PA0/BZ PA1/BZ	输入 / 输出 输入 / 输出	唤醒功能 上拉电阻 BZ 或 BZ	2 位双向输入 / 输出口，每个位都具有唤醒输入。PA0 和 PA1 分别与 BZ 和 BZ 共用引脚。一旦 PA0 和 PA1 选择为蜂鸣器输出，那么输出信号来自内部蜂鸣器时钟发生器。软件指令决定引脚是 CMOS 输出或斯密特触发输入，可由掩膜选项决定是否有上拉电阻。
PA2/EL	输入 / 输出	唤醒功能 上拉电阻 EL	1 位双向输入 / 输出口，具有唤醒输入。PA2 和 EL 载波输出共用引脚。一旦 PA2 选为 EL 载波输出，那么输出信号来自内部 EL 载波时钟发生器。软件指令决定引脚是 CMOS 输出或斯密特触发输入，可由掩膜选项决定是否有上拉电阻。
PA3~PA7	输入 / 输出	唤醒功能 上拉电阻	5 位双向输入 / 输出口，每个位都具有唤醒输入。软件指令决定引脚是 CMOS 输出或斯密特触发输入，可由掩膜选项决定是否有上拉电阻。
PB0/SEG9~ PB7/SEG16	输入 / 输出	I/O 或 SEG	8 位双向输入 / 输出口。软件指令决定引脚是 CMOS 输出或斯密特触发输入，可由掩膜选项决定是否有上拉电阻。这些引脚都可作为 LCD 面板 Segment 驱动输出，可由掩膜选项选择作为输入 / 输出口或 Segment 驱动输出。
PC0/SEG17~ PC1/SEG18	输入 / 输出	I/O 或 SEG	2 位双向输入 / 输出口。软件指令决定引脚是 CMOS 输出或斯密特触发输入，可由掩膜选项决定是否有上拉电阻。这些引脚都可作为 LCD 面板 Segment 驱动输出，可由掩膜选项选择作为输入 / 输出口或 Segment 驱动输出。
SEG0~SEG8	输出	—	LCD 面板 Segment 驱动输出。
COM0~COM1 COM2/SEG19	输出	1/2 或 1/3 Duty	COM0~COM1 是 LCD 的 common 输出。 LCD 的 1/3 占空比的掩膜选项决定 COM2/SEG19 引脚是作为 LCD 面板的 SEG19 驱动输出还是作为 COM2 驱动输出。
VA,C1,C2	—	—	VA: LCD 电源电压，需在 VA 与 VSS 之间接入一个电容。 C1, C2: VA 的转换引脚，需在 C1 与 C2 之间接入一个电容。
RCIN	输入	—	RC 型 A/D 转换 RC 振荡输入引脚。
RREF	输出	—	RC 型 A/D 转换参考电阻连接引脚。
RSEN	输出	—	RC 型 A/D 转换传感器电阻连接引脚。
VDD	电源	—	电源正极。
VSS	电源	—	电源负极，地。

HT47C08L

引脚名称	输入 / 输出	掩膜选项	功能说明
RES	输入	—	斯密特触发复位输入，低电平有效。
PA0/BZ PA1/ \overline{BZ}	输入 / 输出 输入 / 输出	唤醒功能 上拉电阻 BZ 或 \overline{BZ}	2 位双向输入 / 输出口，每个位都具有唤醒输入。PA0 和 PA1 分别与 BZ 和 \overline{BZ} 共用引脚。一旦 PA0 和 PA1 选择为蜂鸣器输出，那么输出信号来自内部蜂鸣器时钟发生器。软件指令决定引脚是 CMOS 输出或斯密特触发输入，可由掩膜选项决定是否有上拉电阻。
PA2/EL	输入 / 输出	唤醒功能 上拉电阻 EL	1 位双向输入 / 输出口，具有唤醒输入。PA2 和 EL 载波输出共用引脚。一旦 PA2 选为 EL 载波输出，那么输出信号来自内部 EL 载波时钟发生器。软件指令决定引脚是 CMOS 输出或斯密特触发输入，可由掩膜选项决定是否有上拉电阻。
PA3~PA7	输入 / 输出	唤醒功能 上拉电阻	5 位双向输入 / 输出口，每个位都具有唤醒输入。软件指令决定引脚是 CMOS 输出或斯密特触发输入，可由掩膜选项决定是否有上拉电阻。
PC2~PC4	输入 / 输出	—	3 位双向输入 / 输出口。软件指令决定引脚是 CMOS 输出或斯密特触发输入，可由掩膜选项决定是否有上拉电阻。
PB0/SEG9~ PB7/SEG16	输入 / 输出	I/O 或 SEG	8 位双向输入 / 输出口。软件指令决定引脚是 CMOS 输出或斯密特触发输入，可由掩膜选项决定是否有上拉电阻。这些引脚都可作为 LCD 面板 Segment 驱动输出，可由掩膜选项选择作为输入 / 输出口或 Segment 驱动输出。
PC0/SEG17~ PC1/SEG18	输入 / 输出	I/O 或 SEG	2 位双向输入 / 输出口。软件指令决定引脚是 CMOS 输出或斯密特触发输入，可由掩膜选项决定是否有上拉电阻。这些引脚都可作为 LCD 面板 Segment 驱动输出，可由掩膜选项选择作为输入 / 输出口或 Segment 驱动输出。
SEG0~SEG8 SEG19~20	输出	—	LCD 面板 Segment 驱动输出。
COM0~COM2	输出	—	COM0~COM2 是 LCD 的 common 输出。
VA,VC,C1,C2	—	—	VA, VC: LCD 电源电压，需分别在 VA 与 VSS、VC 与 VSS 之间接入一个电容。 C1, C2: VA 和 VC 的转换引脚，需在 C1 与 C2 之间接入一个电容。
RCIN0	输入	—	RC 型 A/D 转换 RC 振荡输入引脚 0。
RREF0	输出	—	RC 型 A/D 转换参考电阻连接引脚 0。
RSEN0	输出	—	RC 型 A/D 转换传感器电阻连接引脚 0。
RCIN1	输入	—	RC 型 A/D 转换 RC 振荡输入引脚 1。
RREF1	输出	—	RC 型 A/D 转换参考电阻连接引脚 1。
RSEN1	输出	—	RC 型 A/D 转换传感器电阻连接引脚 1。
VDD	电源	—	电源正极。
VSS	电源	—	电源负极，地。

极限参数

电源供应电压	$V_{SS}-0.3V \sim V_{SS}+2.5V$
端口输入电压	$V_{SS}-0.3V \sim V_{DD}+0.3V$
储存温度	$-50^{\circ}C \sim 125^{\circ}C$
工作温度	$-10^{\circ}C \sim 50^{\circ}C$

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°C

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	—	—	1.2	—	2.2	V
V _A	LCD 电压	—	1/2 bias	—	V _{DD} ×2	—	V
V _B	LCD 电压	—	1/2 bias	—	V _{DD}	—	V
V _{LVD1}	低电压检测电压	—	掩膜选项	1.25	1.3	1.35	V
V _{LVD2}	低电压检测电压	—	掩膜选项	1.22	1.27	1.32	V
V _{LVD3}	低电压检测电压	—	掩膜选项	1.20	1.25	1.30	V
I _{DD1}	工作电流	1.5V	无负载, f _{SYS} =32kHz, A/D 关闭, LVD 关闭	—	—	5	μA
I _{DD2}	工作电流	1.5V	无负载, f _{SYS} =32kHz, A/D 打开, LVD 关闭 *R=30kΩ, *C=2200pF	—	—	45	μA
I _{DD3}	工作电流	1.5V	无负载, f _{SYS} =128kHz, A/D 关闭, LVD 关闭	—	—	20	μA
I _{DD4}	工作电流	1.5V	无负载, f _{SYS} =128kHz, A/D 打开, LVD 关闭 *R=30kΩ, *C=2200pF	—	—	60	μA
I _{LVD}	LVD 电流	1.5V	LVD 打开	—	50	100	μA
I _{STB1}	静态电流 (LVD 关闭, LCD 关闭, 系统振荡器关闭)	1.5V	无负载, 系统 HALT, A/D 关闭, LVD 关闭	—	0.1	1	μA
I _{STB2}	静态电流 (LCD 打开)	1.5V	无负载, 系统 HALT, f _{SYS} =32kHz, A/D 关闭, LVD 关闭	—	—	3	μA
I _{STB3}	静态电流 (LCD 打开)	1.5V	无负载, 系统 HALT, f _{SYS} =128kHz, A/D 关闭, LVD 关闭	—	—	13	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	—	—	0.8V _{DD}	—	V _{DD}	V

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
V _{IL2}	低电平输入电压 ($\overline{\text{RES}}$)	—	—	0	—	0.4V _{DD}	V
V _{IH2}	高电平输入电压 ($\overline{\text{RES}}$)	—	—	0.9V _{DD}	—	V _{DD}	V
I _{OL1}	输入 / 输出口灌电流 (PA, PB, PC)	1.5V	V _{OL} =0.1V _{DD}	0.5	0.8	—	mA
I _{OH1}	输入 / 输出口源电流 (PA, PB, PC)	1.5V	V _{OH} =0.9V _{DD}	-0.3	-0.6	—	mA
I _{OL2}	输入 / 输出口灌电流 (RREFn, RSEn)	1.5V	V _{OL} =0.1V _{DD}	4	7	—	mA
I _{OH2}	输入 / 输出口源电流 (RREFn, RSEn)	1.5V	V _{OH} =0.9V _{DD}	-3	-5	—	mA
I _{OL3}	Common 和 Segment 灌电流	1.5V	V _{OL} =0.1V _{DD}	50	100	—	μA
I _{OH3}	Common 和 Segment 源电流	1.5V	V _{OH} =0.9V _{DD}	-50	-100	—	μA
R _{PH}	输入 / 输出口上拉电阻	1.5V	—	75	150	300	kΩ

注: *R 代表 RC 型 A/D 转换器的电阻

*C 代表 RC 型 A/D 转换器的电容

交流电气特性

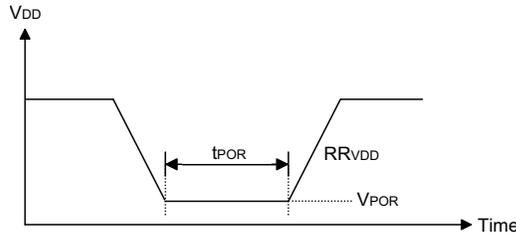
T_a=25°C

符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
f _{LIRC2}	LIRC2 振荡频率	1.5V	—	102.4	128	153.6	kHz
t _{ONLIRC}	LIRC2 振荡器稳定时间	1.5V	—	—	—	1	ms
f _{SYS}	系统时钟	1.5V	—	25	—	154	kHz
t _{RES}	外部复位低电平脉宽	1.5V	—	100	—	—	μs
f _{AD}	RC 型 A/D 转换频率	1.5V	—	—	—	50	kHz
f _{LCD}	LCD 驱动时钟频率	—	由掩膜选项选择	—	4	—	kHz

上电复位特性

T_a=25°C

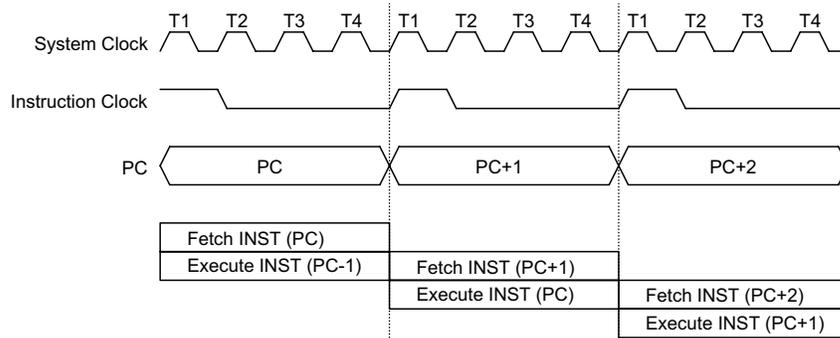
符号	参数	测试条件		最小	典型	最大	单位
		V _{DD}	条件				
V _{POR}	上电复位电压	—	—	—	—	100	mV
RR _{VDD}	上电复位电压速率	—	—	0.035	—	—	V/ms
t _{POR}	V _{DD} 保持为 V _{POR} 的最小时间	—	—	1	—	—	ms



系统结构

指令执行时序

该系列单片机的系统时钟由需要外接电阻的内置 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 – PC

10 位和 11 位的程序计数器 (PC) 控制程序存储器 ROM 中指令执行的顺序，它们可分别寻址 HT47C07L 和 HT47C08L 整个 ROM 范围的 1024 个地址和 2048 个地址。

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										
	*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
定时 / 计数器中断	0	0	0	0	0	0	0	0	1	0	0
时基中断	0	0	0	0	0	0	0	1	0	0	0
条件跳跃	PC+2										
装载 PCL	@10	*9	*8	@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：*10~*0：程序计数器位

#10~#0：指令代码位

S10~S0：堆栈寄存器位

@7~@0：PCL 位

对于 HT47C07L，程序计数器为 10 位，即 bit9~bit0；

对于 HT47C08L，程序计数器为 11 位，即 bit10~bit0。

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。HT47C07L/HT47C08L 有 4 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“CALL”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 4 个返回地址会被保留。

算术逻辑单元 – ALU

- 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, 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

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。

程序存储器

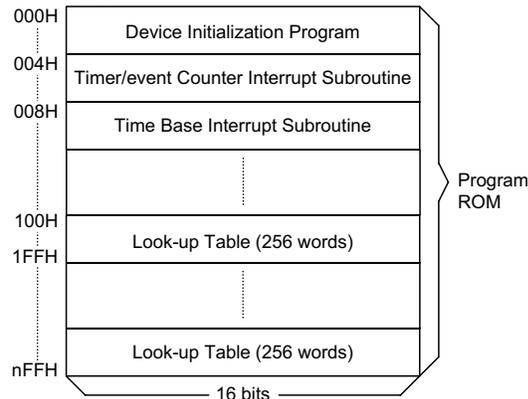
程序存储器用来存放要执行的指令代码，以及一些数据、表格和中断入口。HT47C07L 的程序存储器有 1024×16 位，HT47C08L 的程序存储器有 2048×16 位，程序存储器空间可以用程序计数器或表格指针进行寻址。

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
该地址为定时 / 计数器中断服务程序保留。当定时 / 计数器 A 或 B 发生溢出，如果中断允许且堆栈未滿，则程序会跳转到 004H 地址开始执行。
- 地址 008H
该地址为时基中断服务程序保留。当时基发生溢出，如果中断允许且堆栈未滿，则程序会跳转到 008H 地址开始执行。

● 表格区

程序存储器空间的任何地址都可做为查表使用。查表指令“TABRDC [m]”（查当前页表格，1 页 = 256 个字）和“TABRDL [m]”（查最后一页表格），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。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。TBLH 寄存器只可读，不可写。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。



注：对于 HT47C07L，n 为 3；对于 HT47C08L，n 为 7。

程序存储器

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

表格区

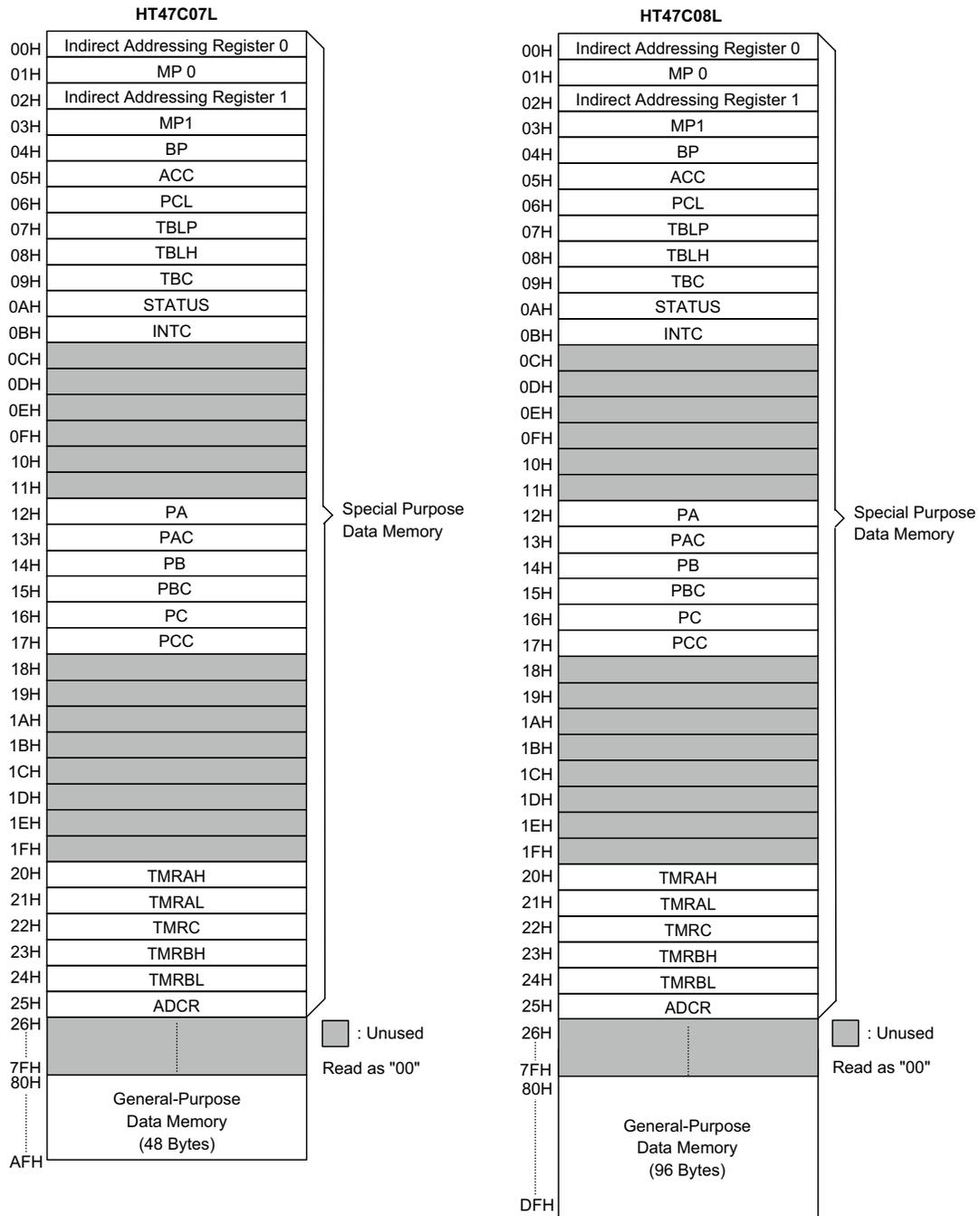
注：*10~*0：表格地址位 P10~P8：当前程序计数器位 @7~@0：表格指针位
对于 HT47C07L，表格地址为 10 位；
对于 HT47C08L，表格地址为 11 位。

数据存储器 – RAM

数据存储器分为两个功能区间：特殊功能寄存器和通用数据存储器，数据存储器单元大多数是可读 / 写的，但有些只读的。

特殊功能寄存器包括间接寻址寄存器 0(IAR0)，间接寻址指针寄存器 0(MP0)，间接寻址寄存器 1(IAR1)，间接寻址指针寄存器 1(MP1)，存储器段指针寄存器 (BP)，累加器 (ACC)，程序计数器低字节寄存器 (PCL)，表格指针寄存器 (TBLP)，表格内容高字节寄存器 (TBLH)，时基控制寄存器 (TBC)，状态标志寄存器 (STATUS)，中断控制寄存器 (INTC)，输入 / 输出寄存器 (PA, PB, PC)，输入 / 输出控制寄存器 (PAC, PBC, PCC)，定时 / 计数器 A 高、低位字节寄存器 (TMRAH, TMRAL)，定时 / 计数器控制寄存器 (TMRC)。定时 / 计数器 B 高、低位字节寄存器 (TMRBH, TMRBL)，RC 振荡型 A/D 转换控制寄存器 (ADCR)。其余在 80H 之前的空间保留给系统以后扩展使用，读取这些地址的返回值为“00H”。HT47C07L 的通用数据寄存器地址从 80H 到 AFH，HT47C08L 的通用数据寄存器地址从 80H 到 DFH，用来存储数据和控制信息。

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通过“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; 01H, MP1; 03H) 进行间接寻址。



数据存储区

特殊功能寄存器

大部分特殊功能寄存器的细节将在相关功能章节描述，但有几个寄存器需在此章节单独描述。

间接寻址寄存器 – IAR0, IAR1

地址 00H 和 02H 是间接寻址寄存器，并无实际的物理区存在。任何对 [00H] 和 [02H] 的读 / 写操作，都是访问由 MP0(01H) 和 MP1(03H) 所指向的 RAM 单元。间接读取 00H 或 02H 地址得到的值为 00H，间接写入此地址，不会产生任何操作。

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 是 8 位寄存器。MP0 只能用于数据存储器，而 MP1 能用于数据存储器 and LCD 显示存储器。

累加器 – ACC

累加器 (ACC) 与算术逻辑单元 (ALU) 有密切关系。它对应于 RAM 地址 05H，做为运算的立即数据。存储器之间的数据传送必须经过累加器。

状态寄存器 – STATUS

这 8 位的状态寄存器由零标志位 (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。这些算术 / 逻辑操作和系统运行标志位是用来记录单片机的运行状态。

除了 PDF 和 TO 标志外，状态寄存器中的位像其它大部分寄存器一样可以被改变。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。

- C: 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 也会被带进位的移位指令所影响。
- AC: 当低半字节加法运算的结果产生进位，或低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。
- Z: 当算术或逻辑运算结果是零时，Z 被置位，否则 Z 被清零。
- OV: 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。
- PDF: 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。
- TO: 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。

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。

STATUS 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	TO	PDF	OV	Z	AC	C
R/W	—	—	R	R	R/W	R/W	R/W	R/W
POR	—	—	0	0	×	×	×	×

“×”为未知

- Bit 7~6 未使用，读为“0”
- Bit 5 **TO**: 看门狗溢出标志位
0: 系统上电或执行“CLR WDT”或“HALT”指令后
1: 看门狗溢出发生
- Bit 4 **PDF**: 暂停标志位
0: 系统上电或执行“CLR WDT”指令后
1: 执行“HALT”指令
- Bit 3 **OV**: 溢出标志位
0: 无溢出
1: 运算结果高两位的进位状态异或结果为 1
- Bit 2 **Z**: 零标志位
0: 算术或逻辑运算结果不为 0
1: 算术或逻辑运算结果为 0
- Bit 1 **AC**: 辅助进位标志位
0: 无辅助进位
1: 在加法运算中低四位产生了向高四位进位，或减法运算中低四位不发生从高四位借位
- Bit 0 **C**: 进位标志位
0: 无进位
1: 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发生借位
C 也受循环移位指令的影响。

振荡电路

内部 RC 振荡器是一个完全集成的系统振荡器，无需外部元件。内部 RC 振荡器具有三种固定的频率：32kHz，64kHz，128kHz，可由掩膜选项选择。振荡器的频率会随着电源电压，温度和芯片自身的参数发生变化。

看门狗定时器 – WDT

WDT 的时钟来源 (f_s) 是系统时钟 f_{SYS} 。WDT 是用来防止程序的不正常运行或是跳到未知或不希望去的地址，而导致不可预见的结果。WDT 可以被掩膜选择禁止。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。

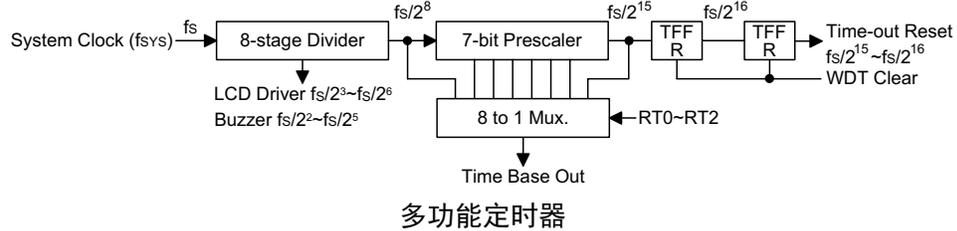
如果看门狗定时器功能使能，在 HALT 状态下，由于暂停模式振荡器还会继续工作，WDT 会继续计数直到溢出使系统唤醒。

看门狗定时器溢出在正常操作时不仅启动系统复位 (Chip Reset)，并置位状态位 TO。在暂停模式中，这溢出会启动热复位 (Warm Reset)，但只有程序计数器和状态指针会复位为零。要清除 WDT 的值可以有三种方法：外部复位 (低电平输入到 RES 端)、清除看门狗指令或 HALT 指令。清除看门狗指令为一条“CLR WDT”指令，只要执行“CLR WDT”指令就会清除 WDT。WDT 会由于溢出而使系统复位。

看门狗定时器的溢出时间周期为 $f_s/2^{15} \sim f_s/2^{16}$ 。执行“CLR WDT”指令只能清除 WDT 的最后两级锁存器。

多功能计时器

该系列单片机为 WDT 和时基提供了具有不同溢出周期的多功能定时器。此多功能定时器由一个八阶除法器及一个七位预分频器所组成，使用的时钟源来自系统时钟。多功能定时器为 LCD 驱动电路提供可选择的频率信号（范围从 $f_s/2^3 \sim f_s/2^6$ ），并为蜂鸣器输出电路提供可选择的频率信号（范围从 $f_s/2^2 \sim f_s/2^5$ ），频率由掩膜选择。为了正确地显示，建议选择 4kHz 作为 LCD 驱动信号。



时基 – TB

时基用来提供一个有规律的内部中断。溢出周期的范围为 $f_s/2^8 \sim f_s/2^{15}$ ，可通过软件编程实现。写数据到 RT2, RT1, RT0（TBC 的第 2, 1, 0 位；09H）将产生各种溢出周期。如果时基溢出产生，系统置位相关的中断请求标志位（TBF；INTC 第 5 位）。如果中断允许，堆栈未滿，那么就产生一个地址在 08H 的子程序调用。在 HALT 状态时，如果 f_{OSC} 打开，时基仍旧工作而且可以唤醒系统。如果在进入 HALT 之前，TBF 被置位，则唤醒功能无效。

TBC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	RT2	RT1	RT0
R/W	—	—	—	—	—	R/W	R/W	R/W
POR	—	—	—	—	—	1	1	1

Bit 7~3 未使用，读为“0”

Bit 2~0 **RT2~RT0**: 时基分频级数

- 000: 2^8
- 001: 2^9
- 010: 2^{10}
- 011: 2^{11}
- 100: 2^{12}
- 101: 2^{13}
- 110: 2^{14}
- 111: 2^{15}

复位和初始化

总共有 3 种方法会产生系统复位，如下所示：

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

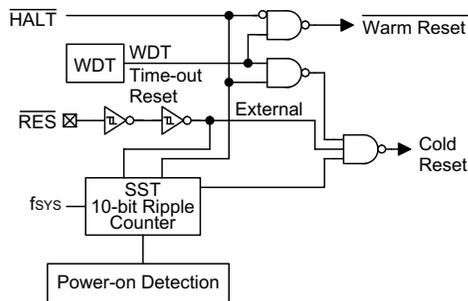
暂停模式中的看门狗定时器溢出与其它系统复位状况不同，因为看门狗定时器溢出会执行热复位，用来重新设置程序计数器和堆栈指针，并保持其它电路原有的状态。少数寄存器在其它复位状态皆不会改变，大部分寄存器一旦符合复位条件时，会复位成初始的状态。通过检测 PDF 和 TO 这两个标志，程序即可区别出各种不同的系统复位。

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

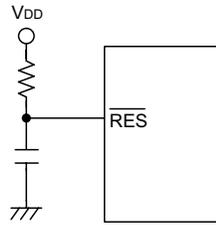
注：“u”表示不变

系统复位时各功能单元的状态如下所示：

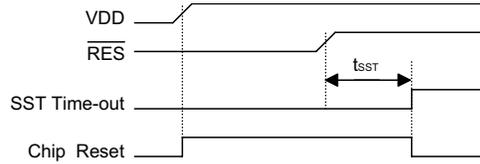
项目	复位后情况
程序计数器 PC	000H
中断	禁止
预分频器、除法器	清除
看门狗定时器、时基	清除、复位后看门狗开始计数
定时 / 计数器	关闭
输入 / 输出口	输入模式
堆栈指针 SP	指向堆栈的顶端



复位框图



复位电路



复位时序

有关寄存器的状态如下：

HT47C07L

寄存器	复位 (上电)	WDT 溢出 (正常运作)	$\overline{\text{RES}}$ 复位 (正常运作)	$\overline{\text{RES}}$ 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	---- ---0	---- ---0	---- ---0	---- ---0	---- ---0
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBC	---- -111	---- -111	---- -111	---- -111	---- -uuu
STATUS	--00 xxxx	--1u uuuu	--uu uuuu	--01 uuuu	--11 uuuu
INTC	--00 -000	--00 -000	--00 -000	--00 -000	--uu -uuu
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	---- --11	---- --11	---- --11	---- --11	---- --uu
PCC	---- --11	---- --11	---- --11	---- --11	---- --uu
TMRAH	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMRAL	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMRC	-000 1---	-000 1---	-000 1---	-000 1---	-uuu u---
TMRBH	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMRBL	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
ADCR	-00x 0000	-00x 0000	-00x 0000	-00x 0000	-uuu uuuu

注：“u”表示不改变
 “x”表示未知
 “-”表示未定义

HT47C08L

寄存器	复位 (上电)	WDT 溢出 (正常运作)	RES 复位 (正常运作)	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	---- --0	---- --0	---- --0	---- --0	---- --0
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
PCL	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBC	---- -111	---- -111	---- -111	---- -111	---- -uuu
STATUS	--00 xxxx	--1u uuuu	--uu uuuu	--01 uuuu	--11 uuuu
INTC	--00 -000	--00 -000	--00 -000	--00 -000	--uu -uuu
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	---1 1111	---1 1111	---1 1111	---1 1111	---u uuuu
PCC	---1 1111	---1 1111	---1 1111	---1 1111	---u uuuu
TMRAH	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMRAL	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMRC	-000 1---	-000 1---	-000 1---	-000 1---	-uuu u---
TMRBH	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
TMRBL	xxxx xxxx	xxxx xxxx	xxxx xxxx	xxxx xxxx	uuuu uuuu
ADCR	000x 0000	000x 0000	000x 0000	000x 0000	uuuu uuuu

注：“u”表示不改变
“x”表示未知
“-”表示未定义

输入 / 输出口

单片机有多个双向输入 / 输出口，标号为 PA, PB 和 PC，对应 RAM 中的 [12H]、[14H] 和 [16H]。所有的输入 / 输出口都可以作为输入和输出，就输入而言，这些输入 / 输出口并不具有锁存功能，也就是说，所有输入在 MOV A, [m] 指令 (m=12H, 14H 或 16H) 的 T2 上升沿准备好。就输出而言，所有数据被锁存住，而且不受任何影响，直到输出锁存被写入新的值为止。

每个 I/O 口都有其自己的控制寄存器 (PAC, PBC, PCC)，用来控制输入 / 输出的设置。使用控制寄存器，可对 CMOS 输出或带上拉电阻的斯密特触发输入在软件下动态地进行改变。要设置为输入功能，相应的控制寄存器必须写 1，同时输入 / 输出口会被自动加上上拉电阻。信号源的输入也取决于控制寄存器。如果控制寄存器的某位值为 1，那么输入信号是读取这个引脚的状态，但是如果控制寄存器的某位值为 0，那么锁存器的内容将会被送到内部总线。后者可以在“读-修改-写”指令中发生。对于输出功能，只能设置为 CMOS 输出。这些控制寄存器是对应于 RAM 的 13H、15H 和 17H 地址。

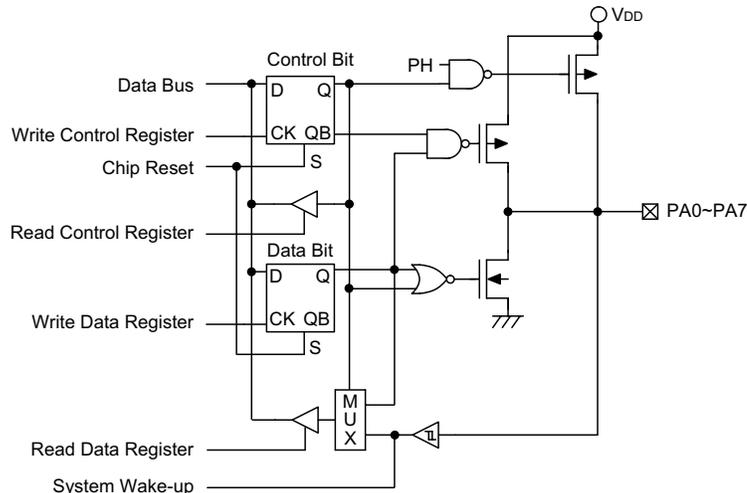
系统复位后，除了 PC0/PC1 为浮接，其余输入 / 输出口都为高电平 (带上拉电阻)。每一个输入 / 输出锁存位都可以用 SET [m].i 或 CLR [m].i 指令置位或清零 (m=12H, 14H 或 16H)。有些指令会先输入数据，然后才输出运行结果。例如，“SET [m].i”和“CPLA[m]”指令，先将整个端口状态读入 CPU 中，接着执行所定义的运算，最后再将执行的结果写入锁存或是累加器中。

PA 的每个口都具有唤醒系统的能力。

PA0/PA1 分别与 BZ/BZ 共用引脚。如果选择蜂鸣器功能，则 PA0/PA1 在输出模式时的输出信号为蜂鸣器 BZ/BZ 信号，在输入模式始终保持它原来的功能。一旦选择 BZ/BZ，4kHz 蜂鸣器输出信号由 PA0/PA1 的数据控制。PA0/BZ 和 PA1/BZ 的真值表如下所示：

PA1 数据寄存器	PA0 数据寄存器	PA1, PA0 引脚功能
0 (CLR PA.1)	0 (CLR PA.0)	PA0=BZ, PA1=BZ
1 (SET PA.1)	0 (CLR PA.0)	PA0=BZ, PA1=0
X	1 (SET PA.0)	PA0=0, PA1=0

掩膜选项

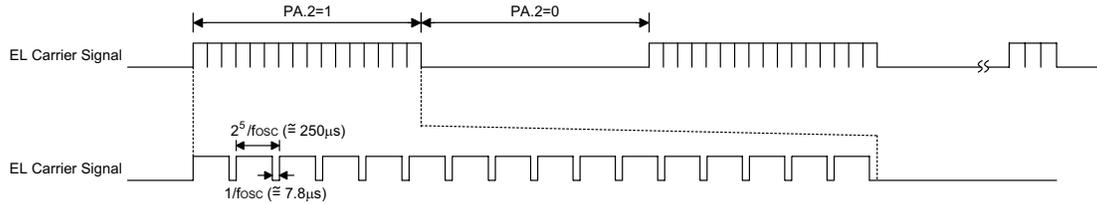


注：框图中没有显示 BZ 和 EL 模式功能

输入 / 输出口

PA2 与 EL 载波信号共用引脚。如果选定为 EL 载波输出，则 PA2 输出模式时的输出信号为 EL 载波信号，输入模式保持原有的功能。由 PA2 数据寄存器控制 EL 载波信号（输出模式）输出。PA2/EL 的真值表如下所示：

PA2 数据寄存器	PA2 引脚功能
0 (CLR PA.2)	PA2=0
1 (SET PA.2)	PA2=EL 载波输出



EL 时序 ($f_{osc}=128kHz$)

输入 / 输出寄存器列表

HT47C07L

寄存器名称	位							
	7	6	5	4	3	2	1	0
PA	D7	D6	D5	D4	D3	D2	D1	D0
PAC	D7	D6	D5	D4	D3	D2	D1	D0
PB	D7	D6	D5	D4	D3	D2	D1	D0
PBC	D7	D6	D5	D4	D3	D2	D1	D0
PC	—	—	—	—	—	—	D1	D0
PCC	—	—	—	—	—	—	D1	D0

HT47C08L

寄存器名称	位							
	7	6	5	4	3	2	1	0
PA	D7	D6	D5	D4	D3	D2	D1	D0
PAC	D7	D6	D5	D4	D3	D2	D1	D0
PB	D7	D6	D5	D4	D3	D2	D1	D0
PBC	D7	D6	D5	D4	D3	D2	D1	D0
PC	—	—	—	D4	D3	D2	D1	D0
PCC	—	—	—	D4	D3	D2	D1	D0

PAC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7~0 PA 口 bit 7~bit 0 输入 / 输出控制位
 0: 输出
 1: 输入

PBC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	1	1	1	1	1	1	1	1

Bit 7~0 PB 口 bit 7~bit 0 输入 / 输出控制位
 0: 输出
 1: 输入

PCC 寄存器
• HT47C07L

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	D1	D0
R/W	—	—	—	—	—	—	R/W	R/W
POR	—	—	—	—	—	—	1	1

Bit 7~2 未使用，读为“0”
 Bit 1~0 PC 口 bit 1~bit 0 输入 / 输出控制位
 0: 输出
 1: 输入

• HT47C08L

Bit	7	6	5	4	3	2	1	0
Name	—	—	—	D4	D3	D2	D1	D0
R/W	—	—	—	R/W	R/W	R/W	R/W	R/W
POR	—	—	—	1	1	1	1	1

Bit 7~5 未使用，读为“0”
 Bit 4~0 PC 口 bit 4~bit 0 输入 / 输出控制位
 0: 输出
 1: 输入

定时 / 计数器

单片机提供一个 16 位定时 / 计数器或一个 RC 型 A/D 转换器。ADC/TM 位 (ADCR 寄存器的第 1 位) 用来决定定时器 A 和定时器 B 是用作 16 位的定时 / 计数器还是用作 RC 型 A/D 转换器。

当 ADC/TM 为“0”时, TMRAL、TMRAH、TMRBL、TMRBH 组成了 16 位的定时 / 计数器。TMRBL 和 TMRBH 组成一个预置寄存器, 分别用来存放定时 / 计数器初始值的低字节和高字节。

当定时 / 计数器的时钟来源来自系统时钟 (f_{SYS}) 或外部时钟。外部时钟输入允许用户去计算外部事件, 测量外部 RC 型的 A/D 时钟, 测量时间长度或脉宽、或产生一个精确的时基信号。

总共有六个与定时 / 计数器工作模式有关的寄存器, 分别是 TMRAH([20H])、TMRAL([21H])、TMRC([22H])、TMRBH([23H])、TMRBL([24H]) 和 ADCR([25H])。若写入 TMRBL 只能将数据写入低字节内部缓冲器中, 但若写入的是 TMRBH 则可将数据和低字节内部缓冲器的内容同时写入定时 / 计数器预置寄存器 (16 位) 之中。改变定时 / 计数器预置寄存器的内容, 只可被写入 TMRBH 之动作改变, 但若写入 TMRBL 则可维持定时 / 计数器预置寄存器的内容不受改变。

同样的, 若读取 TMRAH 则可将 TMRAL 传送至低字节内部缓冲器之中, 以避免发生计时错误。然而, 若读取 TMRAL, 则只读回低字节内部缓冲器的内容。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 TMRAH, 以便将定时 / 计数器的低字节数据传送至内部低字节缓冲器之中。

TMRC 为定时 / 计数器控制寄存器, 用来定义定时 / 计数器的某些选项。定时 / 计数器的控制寄存器可以定义定时 / 计数器的工作模式、计数的允许或禁止以及计数的触发沿。写入定时器 B 就可以将定时 / 计数器的初始值放到预置寄存器中, 而读取定时器 A 就可以得到定时 / 计数器的内容。定时器 B 是定时 / 计数器的初始值预置寄存器。

TM0 和 TM1 用来定义操作模式。计数器模式是用来计数外部事件, 这表示时钟来源为外部引脚的输入。定时器模式作为普通定时器使用, 其时钟来源为内部时钟 (f_{SYS})。最后, 脉冲宽度测量模式能够对外部信号的高电平或低电平的持续时间进行计数, 计数的时钟来源为系统时钟 (f_{SYS})。

在事件计数、A/D 时钟或内部定时器模式下, 一旦定时 / 计数器开始计数即从定时 / 计数器的现行内容 (TMRAH 和 TMRAL) 开始计数至 FFFFH。若发生溢出, 计数器即从定时 / 计数器预置寄存器 (TMRBH 和 TMRBL) 重新装入加载值, 并同时置位中断请求标志 (TF; INTC 的第 4 位)。

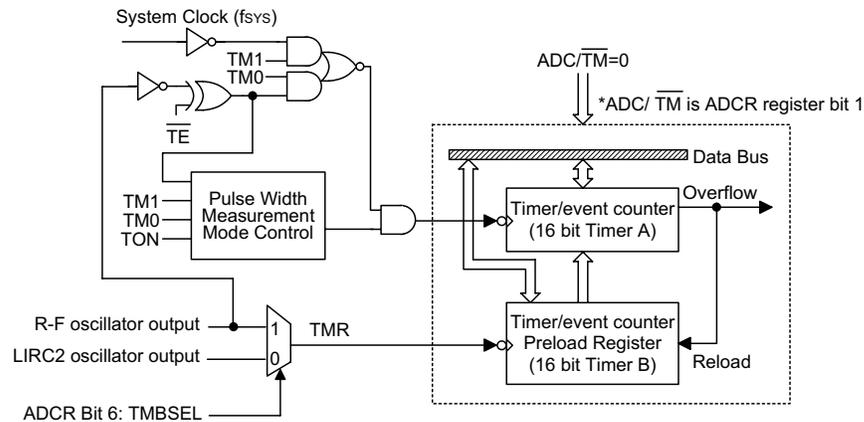
在脉冲宽度测量模式下, 当 TON 和 TE 位的值都为 1 时, 一旦 RCINn 收到由低电平到高电平 (如果 TE 位的值为“0”, 则为由高电平到低电平) 的转变信号, 计数器就会开始计数, 直到 A/D 时钟回到原来的电平为止, 并且会将 TON 位清零。测量的结果会依然存放在定时 / 计数器之中, 也就是说一次只能计数一次脉冲的宽度。而当 TON 位重新置位为“1”, 只要收到跳变脉冲, 测量就会再次执行下去。在脉冲测量模式中, 定时 / 计数器并不会根据逻辑电压来计数, 其根据的标准为信号的转变沿。一旦发生计数溢出, 计数器会从定时 / 计数器加载寄存器重新装入初值, 同时还会发出中断请求, 这种情况和定时和计数这两个模式一样。

若欲启动计数器运行，只要将定时器启动位 (TON: TMRC 的第 4 位) 的值设为“1”即可。在脉冲宽度测量模式中，TON 位在测量周期完成后，会自动被清除。但在其它两种模式中，TON 位只可以用指令清除。

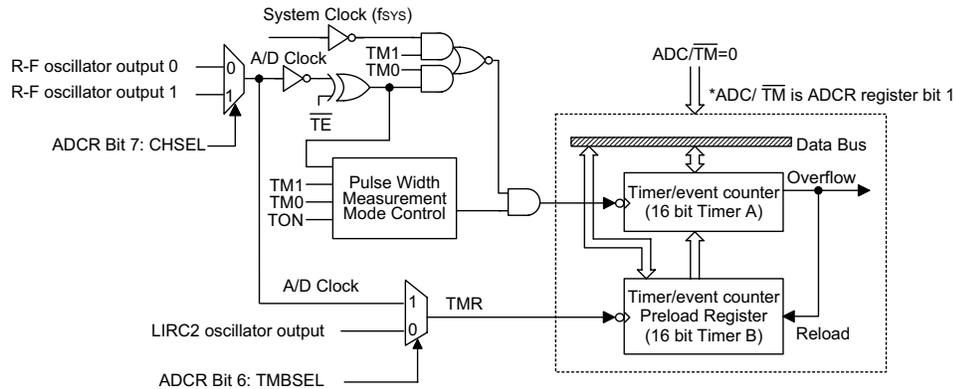
若在定时 / 计数器关闭的情况下，将数据写入定时 / 计数器的预置寄存器同时也会将该数据重新载入定时 / 计数器之中。但若定时 / 计数器已经开启，写入定时 / 计数器的数据只会保存在定时 / 计数器的预置寄存器中。这时定时 / 计数器并不会马上被改变而会继续计数下去，直到发生溢出为止，此时再由预置寄存器装入新的初始值。

一旦定时 / 计数器 (读取 TMRAH) 的数据被读取，会将时钟禁止，以避免发生错误。将时钟禁止可能会导致计数错误，所以程序编写工程师必须考虑清楚才行。

我们强烈建议在打开定时 / 计数器前先将要加载的数据写入到 TMRBL、TMRBH、TMRAL 和 TMRAH 中去，因为在系统初始化后，TMRBL、TMRBH、TMRAL 和 TMRAH 的值是未知的。



定时 / 计数器 - HT47C07L



定时 / 计数器 - HT47C08L

TMRC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	TM1	TM0	TON	TE	—	—	—
R/W	—	R/W	R/W	R/W	R/W	—	—	—
POR	—	0	0	0	1	—	—	—

Bit 7 未使用，读为“0”

Bit 6~5 **TM1~TM0**：定义操作方式

00：未定义

01：外部事件计数器模式（外部时钟：A/D 时钟来自 RCINn）

10：定时器模式（内部时钟： f_{SYS} ）

11：脉冲宽度测量模式（RCINn, f_{SYS} ）

Bit 4 **TON**：使能 / 除能定时器计数

0：除能

1：使能

Bit 3 **TE**：定义定时 / 计数器 TMR 作用沿

外部事件计数器模式

0：上升沿计数

1：下降沿计数

脉冲宽度测量模式

0：下降沿开始计数，上升沿停止计数

1：上升沿开始计数，下降沿停止计数

Bit 2~0 未使用，读为“0”

RC 型 A/D 转换器

该系列单片机有一个 RC 型的 A/D 转换器，包含两个可编程 16 位向上计数的计数器，计数器 A 的时钟来源可以是系统时钟，计数器 B 的时钟来源是外部 RC 振荡器或内部 RC 振荡器，可以通过 ADCR 寄存器的 TMBSEL 位选择。当 ADC/TM 位为“1”时（寄存器 ADCR 的第 1 位），TMRAL、TMRAH、TMRBL、TMRBH 组成了 A/D 转换器。

对于 HT47C07L，A/D 转换定时器 B 的时钟来源可以来自 R-F 振荡器或内部 LIRC2 振荡器。定时器 A 的时钟来源为系统时钟（设置 TM1，TM0=1，0）。对于 HT47C08L，A/D 转换定时器 B 的时钟来源可以来自 R-F 振荡器 0、R-F 振荡器 1 或内部 LIRC2 振荡器。定时器 A 的时钟来源为系统时钟（设置 TM1，TM0=1，0）。

总共有六个与 A/D 转换器有关的寄存器，分别是 TMRAH、TMRAL、TMRC、TMRBH、TMRBL 和 ADCR。内部定时器时钟输入到 TMRAH 和 TMRAL 中，A/D 时钟输入到 TMRBH 和 TMRBL 中。OVB/OVA 位（ADCR 寄存器的第 0 位）用来设置定时器 A 或定时器 B 溢出作为定时 / 计数器中断信号。在 A/D 转换模式下，当定时器 A 或定时器 B 溢出时 TON 位被清除并且计数器停止计数。写入 TMRAH/TMRBH 就是对定时器 A / 定时器 B 设置初值，读取 TMRAH/TMRBH 就是读取定时器 A / 定时器 B 的内容，写入 TMRAL/TMRBL 只能将数据写入内部缓冲器的低位字节，但若写入的是 TMRAH/TMRBH 则可将数据和低字节内部缓冲器的内容写入定时器 A / 定时器 B (16 位) 之中。改变定时 A / 定时器 B 的内容，只可被写入 TMRAH/TMRBH 之动作改变，但若写入 TMRAL/TMRBL 则可维持定时 A / 定时器 B 的内容不改变。

若读取 TMRAH/TMRBH 则可将 TMRAL/TMRBL 传送至低字节内部缓冲器之中，以避免发生计时错误。然而，若读取 TMRAL/TMRBL，则只读回低字节内部缓冲器的内容。换言之，定时器 A/定时器 B 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 TMRAH/TMRBH，以便将定时/计数器的低字节数据传送至内部低字节缓冲器之中。

寄存器 ADCR 的第 2 位用来决定选取哪一组电阻、电容来组成 TMRBH 和 TMRBL 的振荡输入。

寄存器 TMRC 的 TM0 和 TM1 用来决定定时器 A 的时钟来源。建议定时器 A 的时钟来源使用系统时钟。

当 TON 位 (TMRC 的第 4 位) 置为“1”时，定时器 A 和定时器 B 就开始计数，直到定时器 A 或定时器 B 发生溢出，此时，定时/计数器便置位中断请求标志 (TE; INTC 的第 4 位)，同时计数器 A 和计数器 B 停止计数并 TON 位被清为“0”。

当 TON 位置为“1”时，那么 TMRBL、TMRBH、TMRAL 和 TMRAH 不能进行读写操作。只有在定时/计数器关闭并且使用“MOV”指令时，才能对这四个寄存器进行读写操作。

下例为定时/计数器的定时模式 (禁止中断)：

```

clr tmrc
clr adcr.1                ; 设置为定时/计数器模式
clr intc.4                ; 清除定时/计数器的中断请求标志位
mov a, low (65536-1000)   ; 置定时器初值
mov tmrbl, a              ; 计数 1000 然后定时器溢出
mov a, high (65536-1000)
mov tmrbh, a
mov a, 01010000b          ; 定时器时钟来源为 fsys 并且允许定时器计数
mov tmrc, a
p10:
clr wdt

```

系统时钟振荡器是单片机系统时钟的振荡信号，功耗比其它振荡器小。系统时钟也是参考时钟，可以计算实时时间，以减少功耗。然而，在不同的工作电压或环境温度下，系统时钟是不准确的，因此需要先校准。在这种情况下，LIRC2 振荡器比系统时钟振荡器更准确，所以 LIRC2 可用于校准系统时钟振荡器。校准可以通过以下步骤实现。

- 步骤 1：选择系统时钟作为 TMRA 的时钟源，选择 LIRC2 作为 TMRB 的时钟源
- 步骤 2：赋值给 TMRAL 和 TMRAH 寄存器
- 步骤 3：选择 TMRA 作为定时/计数器溢出中断源，然后激活 TMRA 和 TMRB 的定时器模式
- 步骤 4：一旦 TMRA 溢出生成，会自动停止 TMRA 和 TMRB 计数，然后计算出 TMRA 和 TMRB 的方差
- 步骤 5：系统时钟可以通过应用程序计算出的方差进行校准
为了减少功耗，建议在系统时钟脉宽测量时打开 LIRC2，完成测量时关闭 LIRC2。

R-F 振荡器用于时钟振荡，其频率由外部电阻决定。以 R-F 振荡器方框图为例，参考电阻提供给 R-F 振荡器产生了一个输出时钟频率作为参考频率。热敏电阻 NIC 提供给 R-F 振荡器产生了一个输出时钟频率，然后将该频率与参考频率对比以测量温度。温度的测量可以通过以下步骤实现。

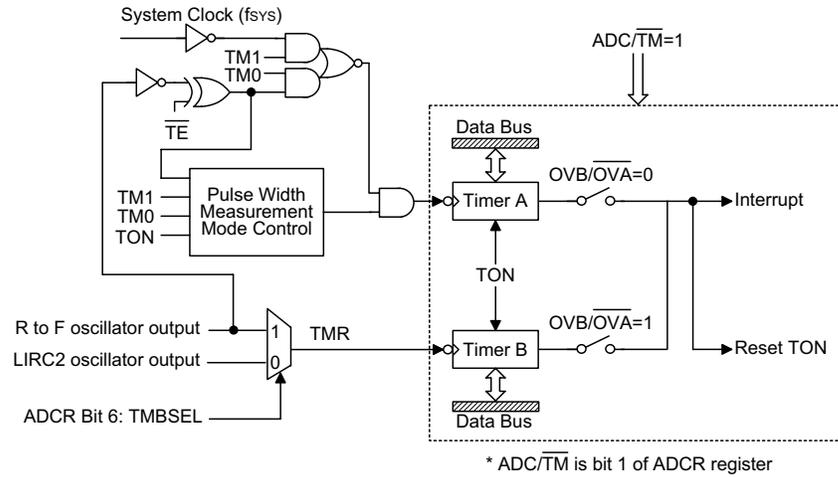
- 步骤 1: 选择系统时钟作为 TMRA 的时钟源，选择 R-F 振荡器作为 TMRB 的时钟源
- 步骤 2: RSEL 控制位清 0，选择 RREF 路径作为 R-F 振荡器的输入 RC 路径
- 步骤 3: 赋值给 TMRAL 和 TMRAH 寄存器
- 步骤 4: 选择 TMRA 作为定时 / 计数器溢出中断源，然后激活 TMRA 和 TMRB 的定时器模式
- 步骤 5: 一旦 TMRA 溢出发生，会自动停止 TMRA 和 TMRB 计数，然后计算出 TMRA 和 TMRB 的方差，该方差称为方差 1
- 步骤 6: RSEL 控制位置 1，选择 RSEN 路径作为 R-F 振荡器的输入 RC 路径，然后重复步骤 4~ 步骤 5，得到的方差称为方差 2
- 步骤 7: 通过比较应用程序中的方差 1 与方差 2，可以计算出 R-F 振荡器中 RSEN 路径与 RREF 路径的输出频率差，并且可以从该频率差计算得到 NIC 温度

R-F 振荡器 0 和 R-F 振荡器 1 打开 / 关闭控制 – HT47C07

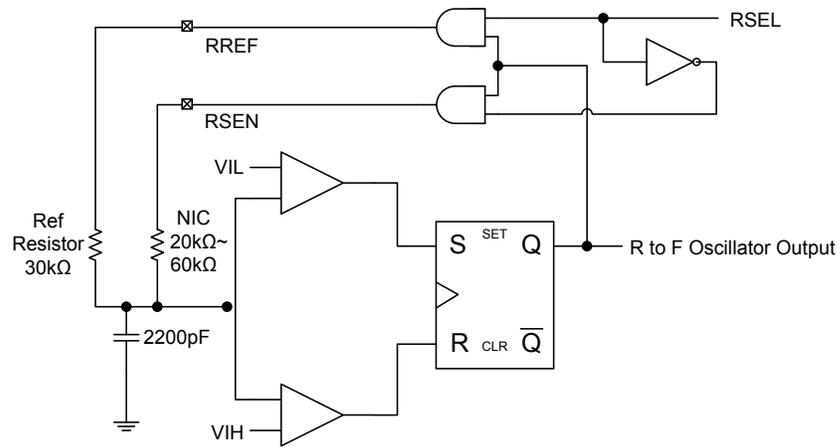
	TM1	TM0	ADC/TM	TON	R-F OSC0	备注
TON 关闭	无关	无关	无关	0	关闭	—
定时器模式	1	0	0	1	关闭	—
事件计数器模式	0	1	1	1	打开	在 TON 设为 1 后，需接着把 TE 设为 1
PWM 模式	1	1	1	1	打开	—
UNUSED	0	0	无关	1	关闭	—
事件计数器模式或 PWM 模式	无关	1	0	1	打开	—

R-F 振荡器 0 和 R-F 振荡器 1 打开 / 关闭控制 – HT47C08L

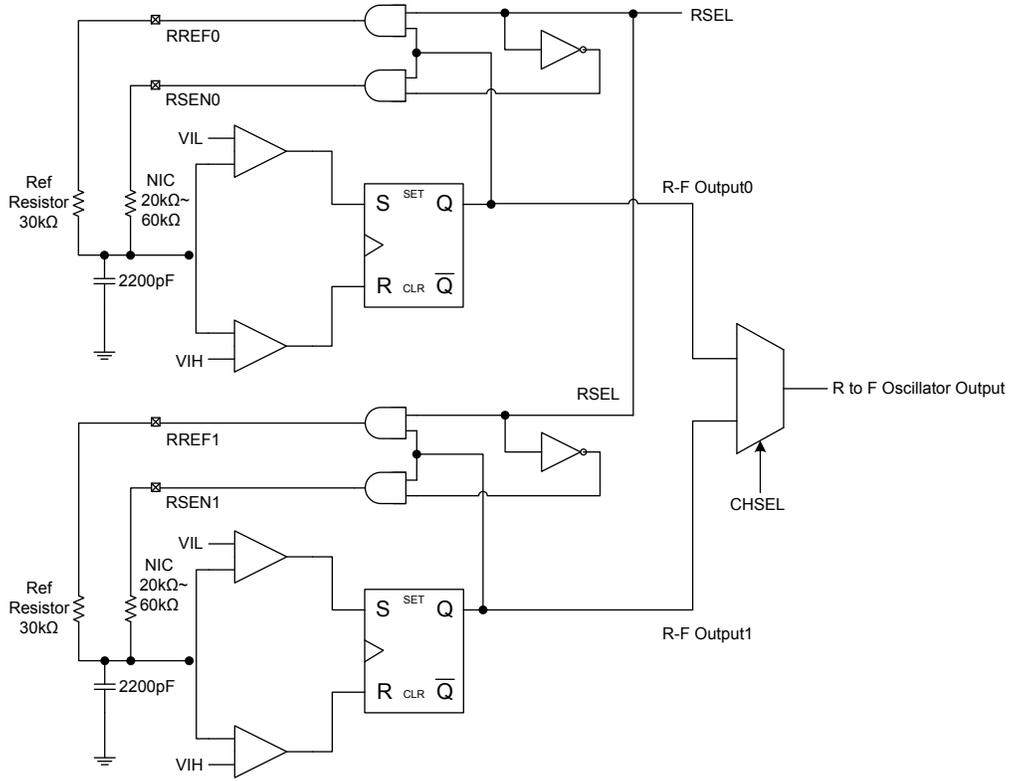
	TM1	TM0	ADC/TM	CHSEL	TON	R-F OSC0	R-F OSC1	备注
TON 关闭	无关	无关	无关	无关	0	关闭	关闭	—
定时器模式	1	0	0	无关	1	关闭	关闭	—
事件计数器模式	0	1	1	0	1	打开	关闭	在 TON 设为 1 后，需接着把 TE 设为 1
事件计数器模式	0	1	1	1	1	关闭	打开	在 TON 设为 1 后，需接着把 TE 设为 1
PWM 模式	1	1	1	1	1	关闭	打开	—
PWM 模式	1	1	1	0	1	打开	关闭	—
UNUSED	0	0	无关	无关	1	关闭	关闭	—
事件计数器模式或 PWM 模式	无关	1	0	0	1	打开	关闭	—
事件计数器模式或 PWM 模式	无关	1	0	1	1	关闭	打开	—



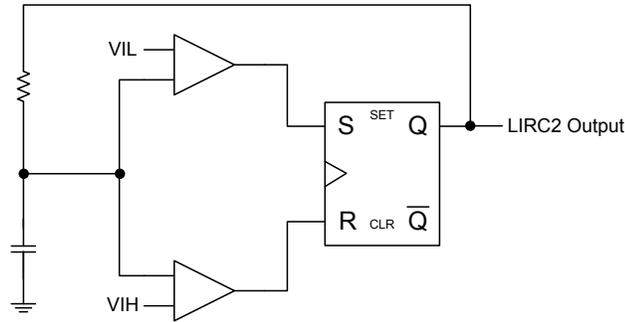
RC 型 A/D 转换器



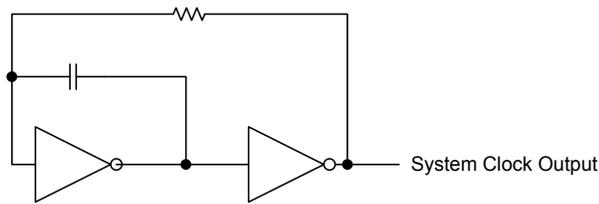
R-F 振荡器 - HT47C07L



R-F 振荡器 - HT47C08L



LIRC2 振荡器



系统时钟振荡器

ADCR 寄存器

• HT47C07L

Bit	7	6	5	4	3	2	1	0
Name	—	TMBSEL	LIRC2ON	BLF	BON	RSEL	ADC/TM	OVB/OVA
R/W	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	—	0	0	0	0	0	0	0

- Bit 7 未使用，读为“0”
- Bit 6 **TMBSEL**: 定时器 B 时钟源选择位
0: LIRC2 振荡器
1: R-F 振荡器
- Bit 5 **LIRC2ON**: LIRC2 振荡器使能控制位
0: 除能
1: 使能
- Bit 4 **BLF**: 低电压标志位
0: 电池电压正常
1: 电池为低电压
- Bit 3 **BON**: 低电压检测使能 / 除能
0: 除能
1: 使能
- Bit 2 **RSEL**: A/D 转换工作模式选择位
0: RREF~CREF 振荡 (参考电阻和参考电容)
1: RSEN~CREF 振荡 (电阻型传感器和参考电容)
- Bit 1 **ADC/TM**: 设定 16 位定时 / 计数器或 RC 型 A/D 转换器使能
0: 16 位定时 / 计数器使能
1: RC 型 A/D 转换器使能

Bit 0 **OVB/OVA**: 选择定时 / 计数器中断源
 0: 定时器 A 溢出
 1: 定时器 B 溢出
 在 RC 型 A/D 转换模式下, 该位用来定义定时 / 计数器中断来自定时器 A 溢出或定时器 B 溢出。在定时 / 计数器模式下, 该位空缺。

● HT47C08L

Bit	7	6	5	4	3	2	1	0
Name	CHSEL	TMBSEL	LIRC2ON	BLF	BON	RSEL	ADC/TM	OVB/OVA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR	0	0	0	0	0	0	0	0

Bit 7 **CHSEL**: 选择 R-F 振荡器输出的输入通道
 0: R-F 输出 0
 1: R-F 输出 1

Bit 6 **TMBSEL**: 定时器 B 时钟源选择位
 0: LIRC2 振荡器
 1: R-F 振荡器

Bit 5 **LIRC2ON**: LIRC2 振荡器使能控制位
 0: 除能
 1: 使能

Bit 4 **BLF**: 低电压标志位
 0: 电池电压正常
 1: 电池为低电压

Bit 3 **BON**: 低电压检测使能 / 除能
 0: 除能
 1: 使能

Bit 2 **RSEL**: A/D 转换工作模式选择位
 0: RREF~CREF 振荡 (参考电阻和参考电容)
 1: RSEN~CREF 振荡 (电阻型传感器和参考电容)

Bit 1 **ADC/TM**: 设定 16 位定时 / 计数器或 RC 型 A/D 转换器允许
 0: 16 位定时 / 计数器允许
 1: RC 型 A/D 转换器允许

Bit 0 **OVB/OVA**: 选择定时 / 计数器中断源
 0: 定时器 A 溢出
 1: 定时器 B 溢出
 在 RC 型 A/D 转换模式下, 该位用来定义定时 / 计数器中断来自定时器 A 溢出或定时器 B 溢出。在定时 / 计数器模式下, 该位空缺。

下例是 RC 型 AD 转换模式（定时器 A 溢出）：

```

clr tmrc
clr adcr.1 ; 设置定时器模式
clr intc.4 ; 清除定时 / 计时器中断请求标志位
mov a, low (65536-1000) ; 置 TIMERA 初值
mov tmral, a ; 计数 1000 后溢出
mov a, high (65536-1000)
mov tmrah, a
mov a, 00000010b ; RREF~CREF, 设置 RC 型 ADC 模式, 设置 TIMERA 溢出
mov adcr, a
mov a, 00h ; 置 TIMERB 初值
mov tmrbl, a
mov a, 00h
mov tmrbh, a
mov a, 01010000b ; TIMERA 的时钟来源为 fsys 并且允许计数
mov tmrc, a
p10:
clr wdt
snz intc.4 ; 判断定时 / 计数器中断请求标志位
jmp p10
clr intc.4 ; 清除定时 / 计数器中断请求标志位
; 程序继续

```

下例是 RC 型 AD 转换模式（定时器 B 溢出）：

```

clr tmrc
clr adcr.1 ; 设置定时器模式
clr intc.4 ; 清除定时 / 计数器中断请求标志位
mov a, 00h ; 置 TIMERA 初值
mov tmral, a
mov a, 00h
mov tmrah, a
mov a, 00000011b ; RREF~CREF, 设置 RC 型 ADC 模式, 设置 TIMERB 溢出
mov adcr, a
mov a, low (65536-1000) ; 置 TIMERB 初值
mov tmrbl, a ; 计数 1000 后溢出
mov a, high (65536-1000)
mov tmrbh, a
mov a, 00110000b ; TIMERA 的时钟来源为 fsys 并且允许计数
mov tmrc, a
p10:
clr wdt
snz intc.4 ; 判断定时 / 计数器中断请求标志位
jmp p10
clr intc.4 ; 清除定时 / 计数器中断请求标志位
; 程序继续

```

暂停模式 – HALT

暂停模式是由 HALT 指令来实现的，具有下列功能：

- f_{osc} 和 f_{sys} 工作还是停止取决于暂停模式下振荡器 / LCD 功能打开 / 关闭，但 T1 被关闭
- 不改变 RAM 和寄存器的内容
- 如果暂停模式下振荡器打开，看门狗定时器清除并重新计数
- 所有输入 / 输出口都维持其原有状态
- 置位 PDF 标志并且清除 TO 标志
- 可由暂停模式下 LCD 功能使能 / 除能选择液晶显示驱动电路仍继续运作还是关闭
- 时基停止还是保持工作取决于暂停模式下振荡器关闭 / 打开

PA 口唤醒和中断这两种唤醒方式可以视为正常运行的继续，PA 口上每个位在掩膜选择时都可以单独选择，用来唤醒系统。一旦从输入 / 输出口启动唤醒之后，程序即从下一条指令重新开始运行。但如果是从中断被唤醒的话，此时可能会发生两种情况。如果相关中断都被禁止，或该中断被允许且堆栈已用满的话，程序会从下一条指令重新开始运行。但如果该中断被允许，但堆栈尚未用满，则会产生一般中断响应。

当进入“HALT”状态以前某个中断请求位被置位，那么系统不能用这个中断来唤醒。

如果唤醒是由于中断响应的话，实际中断子程序的执行会延时大约一个以上的周期。如果唤醒事件导致 HALT 下一条指令执行，会立即执行下一条指令。另外，为减少电源损耗，在进入暂停模式之前，应小心处理所有的输入 / 输出管脚状态。

中断

单片机提供了一个内部定时器 / 计数器中断和一个内部时基中断。中断控制寄存器 (INTC; 0BH) 包含了中断控制位，用来控制中断允许 / 禁止以及中断请求标志。

一旦有中断子程序被服务，其余的中断全部被禁止 (通过清除 EMI 位)，这种做法的目的在于防止中断嵌套。在执行中断服务子程序期间，可能会发生其它的中断请求，但只有中断请求标志会被记录下来。如果来实现中断嵌套，使用者可以在中断子程序中置位 EMI 及 INTC 中对应的中断控制位。如果堆栈已满，即使有关的中断允许，该中断申请并不会被响应，一直到堆栈指针 (SP) 发生递减才会响应。如果需要立即服务，应避免让堆栈发生饱和。

当有中断服务请求，会将程序计数器值压栈，然后再转至中断服务程序的入口。但要记住这时只有程序计数器的内容被压栈，如果寄存器和状态寄存器的内容被中断程序改变，而导致该中断服务程序会破坏预期控制序列的话，使用者应该事先将这些数据储存起来。

内部定时 / 计数器中断是由置位定时 / 计数器中断申请标志 (TF; INTC 的第 4 位) 的方式来启动的，而该中断申请标志是由定时器 A 或定时器 B 溢出产生的。如果中断允许，而此时堆栈尚未用满，而且 TF 也已置位的话，会调用地址 04H 子程序。此时不仅其相关的中断申请标志 (TF) 会被清除，而且 EMI 位也会被清除，以禁止其他中断。

时基中断是由置位时基中断申请标志位 (TBF; INTC 的第 5 位) 的方式来启动的, 而该中断申请标志是由时基时钟产生的。如果中断允许, 而此时堆栈尚未用满, 而且 TBF 也已置位的话, 会调用地址 08H 子程序。此时不仅其相关的中断申请标志 (TBF) 会被清除, 而且 EMI 位也会被清除, 用以禁止任何进一步的中断。

执行中断子程序期间, 其他的中断响应会被暂停, 直到执行 RETI 指令或是将 EMI 位和其相关中断控制位设为 “1” (此时堆栈尚未用满) 为止。若欲从此中断子程序返回, 只要执行 RET 或 RETI 指令即可。其中, RETI 会设定 EMI 位, 用以允许中断服务, 而 RET 则否。

如果在两个连续的 T2 脉冲上升沿之间发生中断, 则在这两个 T2 后面的 T2 脉冲该中断会被服务。而如果同时发生中断申请, 其顺序会依照下表所显示, 这个顺序可以通过设定各中断相关的控制位来改变顺序。

中断源	优先级	中断向量
定时 / 计数器中断	1	04H
时基中断	2	08H

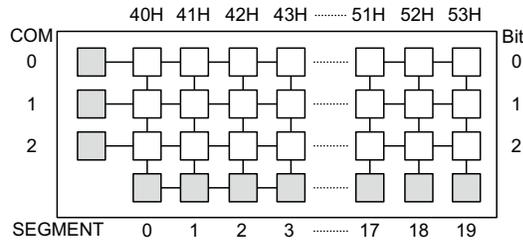
INTC 寄存器

Bit	7	6	5	4	3	2	1	0
Name	—	—	TBF	TF	—	ETBI	ETI	EMI
R/W	—	—	R/W	R/W	—	R/W	R/W	R/W
POR	—	—	0	0	—	0	0	0

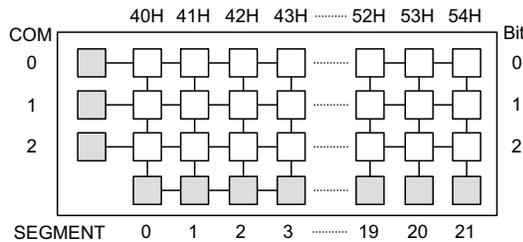
- Bit 7~6 未使用, 读为 “0”
- Bit 5 **TBF**: 时基中断请求标志位
0: 无请求
1: 中断请求
- Bit 4 **TF**: 定时 / 计数器中断请求标志位
0: 无请求
1: 中断请求
- Bit 3 未使用, 读为 “0”
- Bit 2 **ETBI**: 时基中断控制位
0: 除能
1: 使能
- Bit 1 **ETI**: 定时 / 计数器中断控制位
0: 除能
1: 使能
- Bit 0 **EMI**: 总中断控制位
0: 除能
1: 使能

LCD 液晶显示存储器

单片机为液晶显示提供一块嵌入式的数据存储区。它的存储区域位于 RAM 的 BANK1，存储器段指针 (BP; RAM 的 04H 单元) 是在 RAM 和 LCD 显示存储器之间的切换开关。当 BP 被置“01H”，任何数据写入 40H~53H 或 40H~54H 将会影响 LCD 的显示。当 BP 被清“00H”，任何数据写入 40H~53H 或 40H~54H 意味着访问通用数据存储器。LCD 只能通过 MP1 进行间接寻址来访问。当数据被写入显示数据区域，这些数据自动地被 LCD 驱动器读取来产生相应的 LCD 驱动信号。把“1”或“0”写入显示存储器的相应位，来控制显示或不显示。图示表达了显示存储器和单片机上 LCD 显示模式之间的对应显示关系。



显示存储器 (Bank 1) – HT47C07L



显示存储器 (Bank 1) – HT47C08L

液晶显示驱动输出

HT47C07L 的 LCD 液晶显示驱动器数，可通过掩膜选择为 20×2 或 19×3 （1/2 或 1/3 占空比）。HT47C08L 的 LCD 液晶显示驱动器数，可通过掩膜选择为 21×3 （1/3 占空比）。LCD 的驱动偏压方式只有电容式偏压。在 C1 和 C2 引脚之间要接上一个电容。如果选择了 1/2 偏压，则要在 VA 脚和地之间接上一个电容。

注意，对于 HT47C08L，如果选择了 1/3 占空比，则 LCD 驱动器偏压将设置为 1/2。建议 LCD 偏压发生器使用 $0.1\mu\text{F}$ 的电容。LCD 偏压类型、偏压值和电容连接之间的关系如下表所示。

偏压类型	偏压值	C1/C2	VA 引脚	VC 引脚
C	1/2	$0.1\mu\text{F}$	$0.1\mu\text{F}$	x

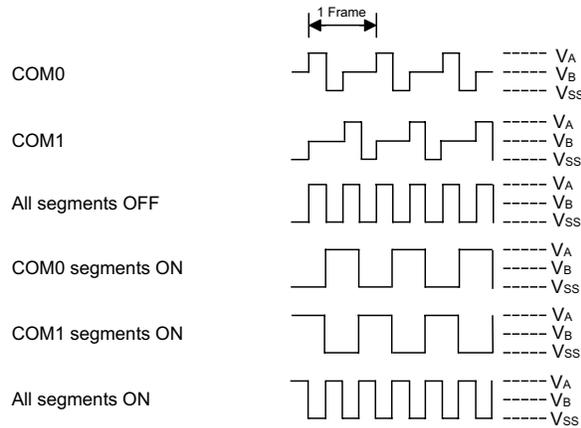
LCD 驱动器需要一个时钟源来驱动，建议使用频率约 4kHz 的时钟。由于 LCD 驱动器时钟来自系统时钟，可以通过掩膜选项选择分频比，以获得合适的 LCD 驱动时钟频率。

时钟源频率	LCD 时钟分频比	LCD 驱动器时钟频率
128kHz	$1/2^5$	4kHz
64kHz	$1/2^4$	4kHz
32kHz	$1/2^3$	4kHz

During Reset or LCD Function is switched off



LCD Function in Normal Operation Mode



Note: $V_A=V_{DD} \times 2$, $V_B=V_{DD}$

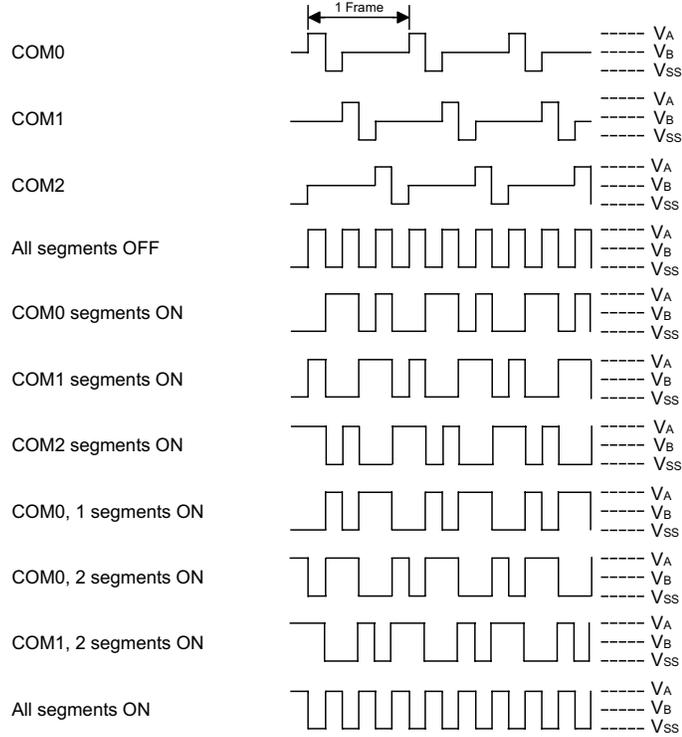
注: $V_A=V_{DD} \times 2$, $V_B=V_{DD}$

LCD 驱动输出 (1/2 Duty, 1/2 Bias)

During Reset or LCD Function is switched off



LCD Function in Normal Operation Mode



注: $V_A = V_{DD} \times 2$, $V_B = V_{DD}$

LCD 驱动输出 (1/3 Duty, 1/2 Bias)

低电压检测 – LVD

单片机为电池应用系统提供了一个低电压检测功能。如果 LVD 打开，当电池电压低于指定电压值时，低电压标志位被置位（BLF；ADCR 的第 4 位）。指定的电压值可以通过掩膜选项设置为 $1.3V \pm 0.05V$ 、 $1.27V \pm 0.05V$ 或 $1.25V \pm 0.05V$ 。低电压检测电路关闭或打开，可通过对 BON（ADCR 第 3 位）写“1”或“0”来控制。当 BON 为“0”时，BLF 无效。检测完电压之后，把 BON 设置为“0”，防止 LVD 消耗直流电流。

掩膜选择

下表列出了单片机的掩膜选择，这些选择必须设定清楚，以确保系统运作正常。

编号	掩膜选择
1	内部 RC 振荡器频率选择： 32kHz, 64kHz 或 128kHz
2	LVD 电压选择： 1.25V, 1.27V 或 1.3V
3	看门狗定时器功能： 使能或除能
4	蜂鸣器输出频率选择： $f_{SYS}/2^2$, $f_{SYS}/2^3$, $f_{SYS}/2^4$ 或 $f_{SYS}/2^5$
5	确定 PA0 和 PA1 的输出功能： 一般输入 / 输出功能或蜂鸣器输出（PA0: BZ; PA1: \overline{BZ} ）
6	确定 PA2 输出功能： 一般输入 / 输出功能或 EL 载波输出
7	当 CPU 处于 HALT 状态时，振荡器 / LCD 关闭或打开： 关闭或打开
8	PA0~PA7 上拉电阻功能： 使能或除能
9	PB0~PB7 上拉电阻功能： 使能或除能
10	PC0~PC1（HT47C07L）/PC0~PC4（HT47C08L）上拉电阻功能： 使能或除能
11	LCD segment 选择（HT47C07L 和 HT47C08L 的 SEG9~SEG18）： I/O 功能或 LCD segment
12	LCD 占空比选择 ^(*) ： 对于 HT47C07L: 1/2 占空比（2 common）或 1/3 占空比（3 common） 对于 HT47C08L: 1/3 占空比（3 common, 1/2 bias）。
13	LCD 驱动时钟选择： $f_{SYS}/2^3$, $f_{SYS}/2^4$, $f_{SYS}/2^5$ 或 $f_{SYS}/2^6$

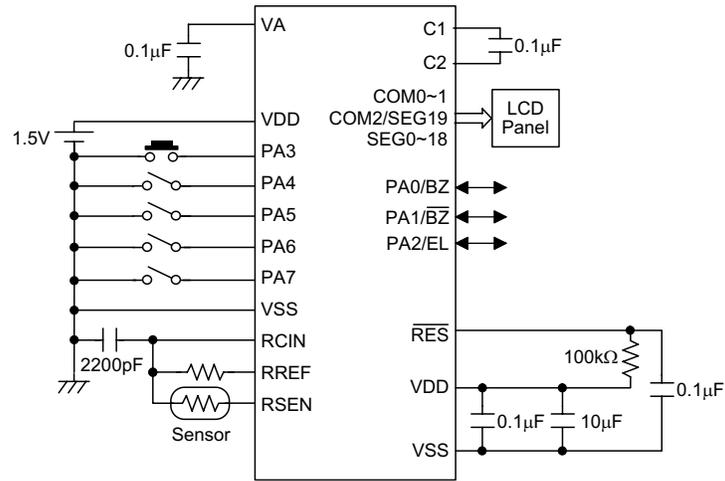
注：对于 HT47C07L:

如果选择了 1/2 占空比，COM2/SEG19 作为 LCD segment 输出脚 SEG19 使用；

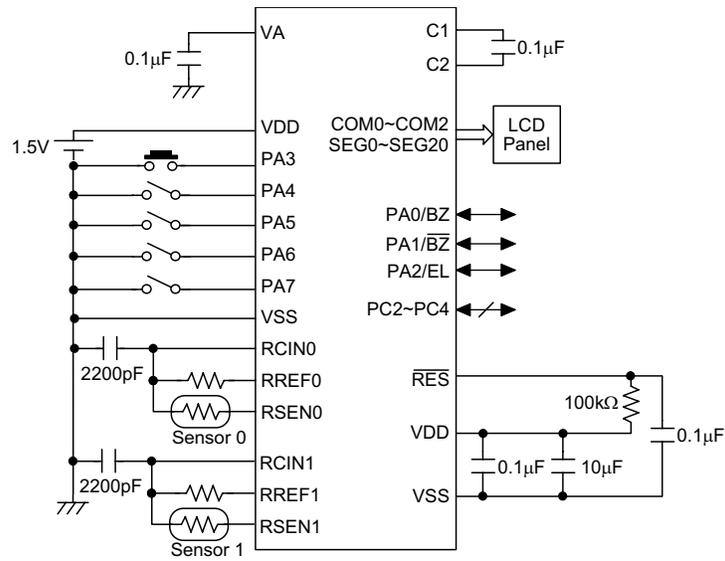
如果选择了 1/3 占空比，COM2/SEG19 作为 LCD common 输出脚 COM2 使用。

应用电路

HT47C07L



HT47C08L



指令集

简介

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 HOLTEK 单片机中，提供了丰富且灵活的指令，共超过六十条，程序设计者可以事半功倍地实现它们的应用。

为了更加容易理解各种各样的指令码，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 0.5 μ s 中执行完成，而分支或调用操作则将在 1 μ s 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”指令。对于跳转指令必须注意的是，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输入端口接收数据或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必需具备的能力，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。不同的移位指令可满足不同的应用需要。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，或是内部数据位的值。

位运算

提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设置尤其有用，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建立一个表格作为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。

指令集概要

下表中说明了按功能分类的指令集，用户可以将该表作为基本的指令参考。

惯例

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

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 ^注	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 ^注	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 ^注	Z, C, AC, OV
SBC A,[m]	ACC 与数据存储器、进位标志的反相减，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 ^注	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 ^注	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 ^注	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 ^注	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 ^注	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 ^注	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 ^注	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 ^注	Z

助记符	说明	指令周期	影响标志位
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 ^注	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 ^注	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 ^注	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 ^注	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 ^注	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 ^注	无
SET [m].i	置位数据存储器的位	1 ^注	无
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 ^注	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 ^注	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 ^注	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 ^注	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 ^注	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 ^注	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 ^注	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 ^注	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			
TABRD [m]	读取特定页的 ROM 内容, 并送至数据存储器 and TBLH	2 ^注	无
TABRDC [m]	读取当前页的 ROM 内容, 并送至数据存储器 and TBLH	2 ^注	无
TABRDL [m]	读取最后一页的 ROM 内容, 并送至数据存储器 and TBLH	2 ^注	无
其它指令			
NOP	空指令	1	无
CLR [m]	清除数据存储器	1 ^注	无
SET [m]	置位数据存储器	1 ^注	无
CLR WDT	清除看门狗定时器	1	TO, PDF

助记符	说明	指令周期	影响标志位
CLR WDT1	预清除看门狗定时器	1	TO, PDF
CLR WDT2	预清除看门狗定时器	1	TO, PDF
SWAP [m]	交换数据存储器的高低字节，结果放入数据存储器	1 ^注	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: 1. 对跳转指令而言，如果比较的结果牵涉到跳转即需 2 个周期，如果没有发生跳转，则只需一个周期。
 2. 任何指令若要改变 PCL 的内容将需要 2 个周期来执行。
 3. 对于“CLR WDT1”或“CLR WDT2”指令而言，TO 和 PDF 标志位也许会受执行结果影响，“CLR WDT1”和“CLR WDT2”被连续地执行后，TO 和 PDF 标志位会被清除，否则 TO 和 PDF 标志位保持不变

指令定义

ADC A, [m]	Add Data Memory to ACC with Carry
指令说明	将指定的数据存储器、累加器内容以及进位标志相加，结果存放到累加器。
功能表示	$ACC \leftarrow ACC + [m] + C$
影响标志位	OV、Z、AC、C
ADCM A, [m]	Add ACC to Data Memory with Carry
指令说明	将指定的数据存储器、累加器内容和进位标志位相加，结果存放到指定的数据存储器。
功能表示	$[m] \leftarrow ACC + [m] + C$
影响标志位	OV、Z、AC、C
ADD A, [m]	Add Data Memory to ACC
指令说明	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。
功能表示	$ACC \leftarrow ACC + [m]$
影响标志位	OV、Z、AC、C
ADD A, x	Add immediate data to ACC
指令说明	将累加器和立即数相加，结果存放到累加器。
功能表示	$ACC \leftarrow ACC + x$
影响标志位	OV、Z、AC、C
ADDM A, [m]	Add ACC to Data Memory
指令说明	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。
功能表示	$[m] \leftarrow ACC + [m]$
影响标志位	OV、Z、AC、C
AND A, [m]	Logical AND Data Memory to ACC
指令说明	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。
功能表示	$ACC \leftarrow ACC \text{ "AND" } [m]$
影响标志位	Z

AND A, x	Logical AND immediate data to ACC
指令说明	将累加器中的数据和立即数做逻辑与，结果存放到累加器。
功能表示	$ACC \leftarrow ACC \text{ "AND" } x$
影响标志位	Z
ANDM A, [m]	Logical AND ACC to Data Memory
指令说明	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。
功能表示	$[m] \leftarrow ACC \text{ "AND" } [m]$
影响标志位	Z
CALL addr	Subroutine call
指令说明	无条件地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继续执行程序，由于此指令需要额外的运算，所以为一个 2 周期的指令。
功能表示	$Stack \leftarrow Program\ Counter + 1$ $Program\ Counter \leftarrow addr$
影响标志位	无
CLR [m]	Clear Data Memory
指令说明	将指定数据存储器的内容清零。
功能表示	$[m] \leftarrow 00H$
影响标志位	无
CLR [m].i	Clear bit of Data Memory
指令说明	将指定数据存储器的 i 位内容清零。
功能表示	$[m].i \leftarrow 0$
影响标志位	无
CLR WDT	Clear Watchdog Timer
指令说明	WDT 计数器、暂停标志位 PDF 和看门狗溢出标志位 TO 清零。
功能表示	WDT cleared $TO \ \& \ PDF \leftarrow 0$
影响标志位	TO、PDF

CLR WDT1	Preclear Watchdog Timer
指令说明	PDF 和 TO 标志位都被清 0。必须配合 CLR WDT2 一起使用清除 WDT 计时器。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。
功能表示	WDT \leftarrow 00H TO & PDF \leftarrow 0
影响标志位	TO、PDF
CLR WDT2	Preclear Watchdog Timer
指令说明	PDF 和 TO 标志位都被清 0。必须配合 CLR WDT1 一起使用清除 WDT 计时器。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。
功能表示	WDT \leftarrow 00H TO & PDF \leftarrow 0
影响标志位	TO、PDF
CPL [m]	Complement Data Memory
指令说明	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 0 变 1。
功能表示	[m] \leftarrow $\overline{[m]}$
影响标志位	Z
CPLA [m]	Complement Data Memory with result in ACC
指令说明	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 0 变 1，而结果被储存回累加器且数据存储器中的内容不变。
功能表示	ACC \leftarrow $\overline{[m]}$
影响标志位	Z

<p>DAA [m] 指令说明</p> <p>功能表示</p> <p>影响标志位</p>	<p>Decimal-Adjust ACC for addition with result in Data Memory 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”，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”。BCD 转换实质上是根据累加器和标志位执行 00H，06H，60H 或 66H 的加法运算，结果存放于数据存储器。只有进位标志位 C 受影响，用来指示原始 BCD 的和是否大于 100，并可以进行双精度十进制数的加法运算。</p> <p>[m] ← ACC + 00H 或 [m] ← ACC + 06H 或 [m] ← ACC + 60H 或 [m] ← ACC + 66H</p> <p>C</p>
<p>DEC [m] 指令说明 功能表示 影响标志位</p>	<p>Decrement Data Memory 将指定数据存储器内容减 1。</p> <p>[m] ← [m] - 1</p> <p>Z</p>
<p>DECA [m] 指令说明 功能表示 影响标志位</p>	<p>Decrement Data Memory with result in ACC 将指定数据存储器内容减 1，把结果存放回累加器并保持指定数据存储器内容不变。</p> <p>ACC ← [m] - 1</p> <p>Z</p>
<p>HALT 指令说明 功能表示 影响标志位</p>	<p>Enter power down mode 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 0。</p> <p>TO ← 0 PDF ← 1</p> <p>TO、PDF</p>
<p>INC [m] 指令说明 功能表示 影响标志位</p>	<p>Increment Data Memory 将指定数据存储器内容加 1。</p> <p>[m] ← [m] + 1</p> <p>Z</p>

INCA [m] 指令说明	Increment Data Memory with result in ACC 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。
功能表示	$ACC \leftarrow [m] + 1$
影响标志位	Z
JMP addr 指令说明	Jump unconditionally 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。当新的地址被加载时，必须插入一个空指令周期，所以此指令为 2 个周期的指令。
功能表示	$Program\ Counter \leftarrow addr$
影响标志位	无
MOV A, [m] 指令说明	Move Data Memory to ACC 将指定数据存储器的内容复制到累加器。
功能表示	$ACC \leftarrow [m]$
影响标志位	无
MOV A, x 指令说明	Move immediate data to ACC 将 8 位立即数载入累加器。
功能表示	$ACC \leftarrow x$
影响标志位	无
MOV [m], A 指令说明	Move ACC to Data Memory 将累加器的内容复制到指定的数据存储器。
功能表示	$[m] \leftarrow ACC$
影响标志位	无
NOP 指令说明	No operation 空操作，接下来顺序执行下一条指令。
功能表示	$PC \leftarrow PC+1$
影响标志位	无
OR A, [m] 指令说明	Logical OR Data Memory to ACC 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。
功能表示	$ACC \leftarrow ACC \text{ "OR" } [m]$
影响标志位	Z

ORA, x	Logical OR immediate data to ACC
指令说明	将累加器中的数据和立即数逻辑或，结果存放到累加器。
功能表示	$ACC \leftarrow ACC \text{ "OR" } x$
影响标志位	Z
ORM A, [m]	Logical OR ACC to Data Memory
指令说明	将存在指定数据存储器的数据和累加器逻辑或，结果放到数据存储器。
功能表示	$[m] \leftarrow ACC \text{ "OR" } [m]$
影响标志位	Z
RET	Return from subroutine
指令说明	将堆栈寄存器中的程序计数器值恢复，程序由取回的地址继续执行。
功能表示	$Program Counter \leftarrow Stack$
影响标志位	无
RET A, x	Return from subroutine and load immediate data to ACC
指令说明	将堆栈寄存器中的程序计数器值恢复且累加器载入指定的立即数，程序由取回的地址继续执行。
功能表示	$Program Counter \leftarrow Stack$ $ACC \leftarrow x$
影响标志位	无
RETI	Return from interrupt
指令说明	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。如果在执行 RETI 指令之前还有中断未被相应，则这个中断将在返回主程序之前被相应。
功能表示	$Program Counter \leftarrow Stack$ $EMI \leftarrow 1$
影响标志位	无
RL [m]	Rotate Data Memory left
指令说明	将指定数据存储器的内容左移 1 位，且第 7 位移到第 0 位。
功能表示	$[m].(i+1) \leftarrow [m].i \ (i=0\sim6)$ $[m].0 \leftarrow [m].7$
影响标志位	无

RLA [m]	Rotate Data Memory left with result in ACC
指令说明	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。
功能表示	$ACC.(i+1) \leftarrow [m].i \ (i=0\sim6)$ $ACC.0 \leftarrow [m].7$
影响标志位	无
RLC [m]	Rotate Data Memory Left through Carry
指令说明	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。
功能表示	$[m].(i+1) \leftarrow [m].i \ (i=0\sim6)$ $[m].0 \leftarrow C$ $C \leftarrow [m].7$
影响标志位	C
RLC A [m]	Rotate Data Memory left through Carry with result in ACC
指令说明	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。
功能表示	$ACC.(i+1) \leftarrow [m].i \ (i=0\sim6)$ $ACC.0 \leftarrow C$ $C \leftarrow [m].7$
影响标志位	C
RR [m]	Rotate Data Memory right
指令说明	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。
功能表示	$[m].i \leftarrow [m].(i+1) \ (i=0\sim6)$ $[m].7 \leftarrow [m].0$
影响标志位	无
RRA [m]	Rotate Data Memory right with result in ACC
指令说明	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。
功能表示	$ACC.i \leftarrow [m].(i+1) \ (i=0\sim6)$ $ACC.7 \leftarrow [m].0$
影响标志位	无

RRC [m]	Rotate Data Memory right through Carry
指令说明	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。
功能表示	$[m].i \leftarrow [m].(i+1) (i=0\sim6)$ $[m].7 \leftarrow C$ $C \leftarrow [m].0$
影响标志位	C
RRCA [m]	Rotate Data Memory right through Carry with result in ACC
指令说明	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。
功能表示	$ACC.i \leftarrow [m].(i+1) (i=0\sim6)$ $ACC.7 \leftarrow C$ $C \leftarrow [m].0$
影响标志位	C
SBC A, [m]	Subtract Data Memory from ACC with Carry
指令说明	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。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
功能表示	$ACC \leftarrow ACC - [m] - \bar{C}$
影响标志位	OV、Z、AC、C、SC、CZ
SBCM A, [m]	Subtract Data Memory from ACC with Carry and result in Data Memory
指令说明	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。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
功能表示	$[m] \leftarrow ACC - [m] - \bar{C}$
影响标志位	OV、Z、AC、C、SC、CZ
SDZ [m]	Skip if Decrement Data Memory is 0
指令说明	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，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。如果结果不为 0，则程序继续执行下一条指令。
功能表示	$[m] \leftarrow [m] - 1$ ，如果 $[m]=0$ 跳过下一条指令执行
影响标志位	无

SDZA [m] 指令说明	Decrement data memory and place result in ACC,skip if 0 将指定数据存储器内容减 1，判断是否为 0，如果为 0 则跳过下一条指令，此结果将存放到累加器，但指定数据存储器内容不变。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。如果结果不为 0，则程序继续执行下一条指令。
功能表示	ACC ← [m]-1，如果 ACC=0 跳过下一条指令执行
影响标志位	无
SET [m] 指令说明	Set Data Memory 将指定数据存储器的每一位设置为 1。
功能表示	[m] ← FFH
影响标志位	无
SET [m].i 指令说明	Set bit of Data Memory 将指定数据存储器的第 i 位置位为 1。
功能表示	[m].i ← 1
影响标志位	无
SIZ [m] 指令说明	Skip if increment Data Memory is 0 将指定的数据存储器内容加 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 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
功能表示	[m] ← [m]+1，如果 [m]=0 跳过下一条指令执行
影响标志位	无
SIZA [m] 指令说明	Skip if increment Data Memory is zero with result in ACC 将指定数据存储器内容加 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。如果结果不为 0，则程序继续执行下一条指令。
功能表示	ACC ← [m]+1，如果 ACC=0 跳过下一条指令执行
影响标志位	无

SNZ [m].i	Skip if bit i of Data Memory is not 0
指令说明	判断指定数据存储器的第 i 位，若不为 0，则程序跳过下一条指令执行。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。如果结果为 0，则程序继续执行下一条指令。
功能表示	如果 [m].i≠0，跳过下一条指令执行
影响标志位	无
SUB A, [m]	Subtract Data Memory from ACC
指令说明	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。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
功能表示	$ACC \leftarrow ACC - [m]$
影响标志位	OV、Z、AC、C、SC、CZ
SUBM A, [m]	Subtract Data Memory from ACC with result in Data Memory
指令说明	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。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
功能表示	$[m] \leftarrow ACC - [m]$
影响标志位	OV、Z、AC、C、SC、CZ
SUB A, x	Subtract immediate Data from ACC
指令说明	将累加器的内容减去立即数，结果存放到累加器。如果结果为负，C 标志位清除为 0，反之结果为正或 0，C 标志位设置为 1。
功能表示	$ACC \leftarrow ACC - x$
影响标志位	OV、Z、AC、C、SC、CZ
SWAP [m]	Swap nibbles of Data Memory
指令说明	将指定数据存储器的低 4 位和高 4 位互相交换。
功能表示	$[m].3\sim[m].0 \leftrightarrow [m].7\sim[m].4$
影响标志位	无
SWAPA [m]	Swap nibbles of Data Memory with result in ACC
指令说明	将指定数据存储器的低 4 位与高 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。
功能表示	$ACC.3\sim ACC.0 \leftarrow [m].7\sim[m].4$ $ACC.7\sim ACC.4 \leftarrow [m].3\sim[m].0$
影响标志位	无

SZ [m] 指令说明	Skip if Data Memory is 0 判断指定数据存储器的内容是否为 0，若为 0，则程序跳下一条指令执行。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。如果结果不为 0，则程序继续执行下一条指令。
功能表示	如果 [m]=0, 跳过下一条指令执行
影响标志位	无
SZA [m] 指令说明	Skip if Data Memory is 0 with data movement to ACC 将指定数据存储器内容复制到累加器，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 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
功能表示	ACC ← [m], 如果 [m]=0, 跳过下一条指令执行
影响标志位	无
SZ [m].i 指令说明	Skip if bit i of Data Memory is 0 判断指定数据存储器的第 i 位是否为 0，若为 0，则跳下一条指令。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 2 个周期的指令。如果结果不为 0，则程序继续执行下一条指令。
功能表示	如果 [m].i=0, 跳过下一条指令执行
影响标志位	无
TABRD [m] 指令说明	Read table (specific page) to TBLH and Data Memory 将表格指针对 TBHP 和 TBLP 所指的程序代码低字节 (指定页) 移至指定数据存储器且将高字节移至 TBLH。
功能表示	[m] ← 程序代码 (低字节) TBLH ← 程序代码 (高字节)
影响标志位	无
TABRDC [m] 指令说明	Read table (current page) to TBLH and Data Memory 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。
功能表示	[m] ← 程序代码 (低字节) TBLH ← 程序代码 (高字节)
影响标志位	无

TABRDL [m]	Read table (last page) to TBLH and Data Memory
指令说明	将表格指针 TBLP 所指的程序代码低字节 (最后一页) 移至指定数据存储器且将高字节移至 TBLH。
功能表示	[m] ← 程序代码 (低字节) TBLH ← 程序代码 (高字节)
影响标志位	无
XOR A, [m]	Logical XOR Data Memory to ACC
指令说明	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。
功能表示	ACC ← ACC “XOR” [m]
影响标志位	Z
XORM A, [m]	Logical XOR ACC to Data Memory
指令说明	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。
功能表示	[m] ← ACC “XOR” [m]
影响标志位	Z
XOR A, x	Logical XOR immediate data to ACC
指令说明	将累加器的数据与立即数逻辑异或, 结果存放到累加器。
功能表示	ACC ← ACC “XOR” x
影响标志位	Z

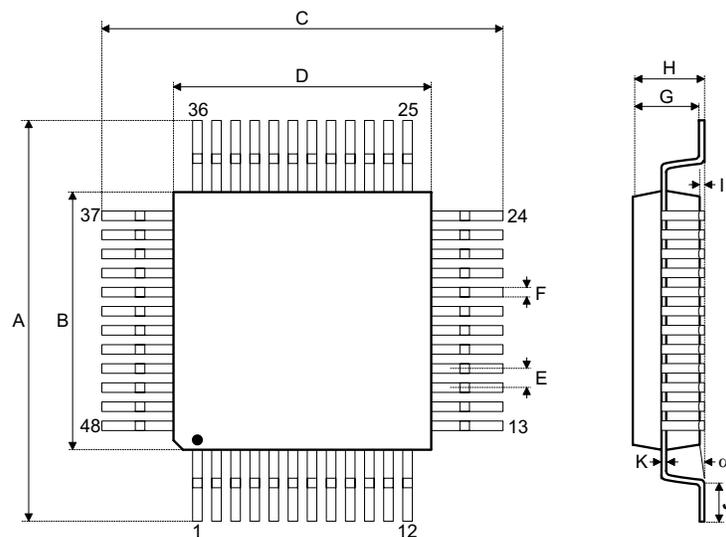
封装信息

请注意，这里提供的封装信息仅作为参考。由于这个信息经常更新，提醒用户咨询 [Holtek 网站](#) 以获取最新版本的封装信息。

封装信息的相关内容如下所示，点击可链接至 Holtek 网站相关信息页面。

- 封装信息（包括外形尺寸、包装带和卷轴规格）
- 封装材料信息
- 纸箱信息

48-pin LQFP (7mm×7mm) 外形尺寸



符号	尺寸 (inch)		
	最小值	典型值	最大值
A	—	0.354 BSC	—
B	—	0.276 BSC	—
C	—	0.354 BSC	—
D	—	0.276 BSC	—
E	—	0.0197 BSC	—
F	0.007	0.009	0.011
G	0.053	0.055	0.057
H	—	—	0.063
I	0.002	—	0.006
J	0.018	0.024	0.030
K	0.004	—	0.008
α	0°	—	7°

符号	尺寸 (单位: mm)		
	最小	典型	最大
A	—	9.0 BSC	—
B	—	7.0 BSC	—
C	—	9.0 BSC	—
D	—	7.0 BSC	—
E	—	0.5 BSC	—
F	0.17	0.22	0.27
G	1.35	1.40	1.45
H	—	—	1.60
I	0.05	—	0.15
J	0.45	0.60	0.75
K	0.09	—	0.20
α	0°	—	7°

Copyright© 2014 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、维生从机或系统中做为关键从机。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>.