盛群知识产权政策

专利权

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

商标权

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

著作权

规格书中所出现的信息在出版当时相信是正确的，然而盛群对于规格内容的使用不负责任。文中提到的应用其目的仅仅是用来做说明，盛群不保证或不表示这些应用没有更深入的修改就能适用，也不推荐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

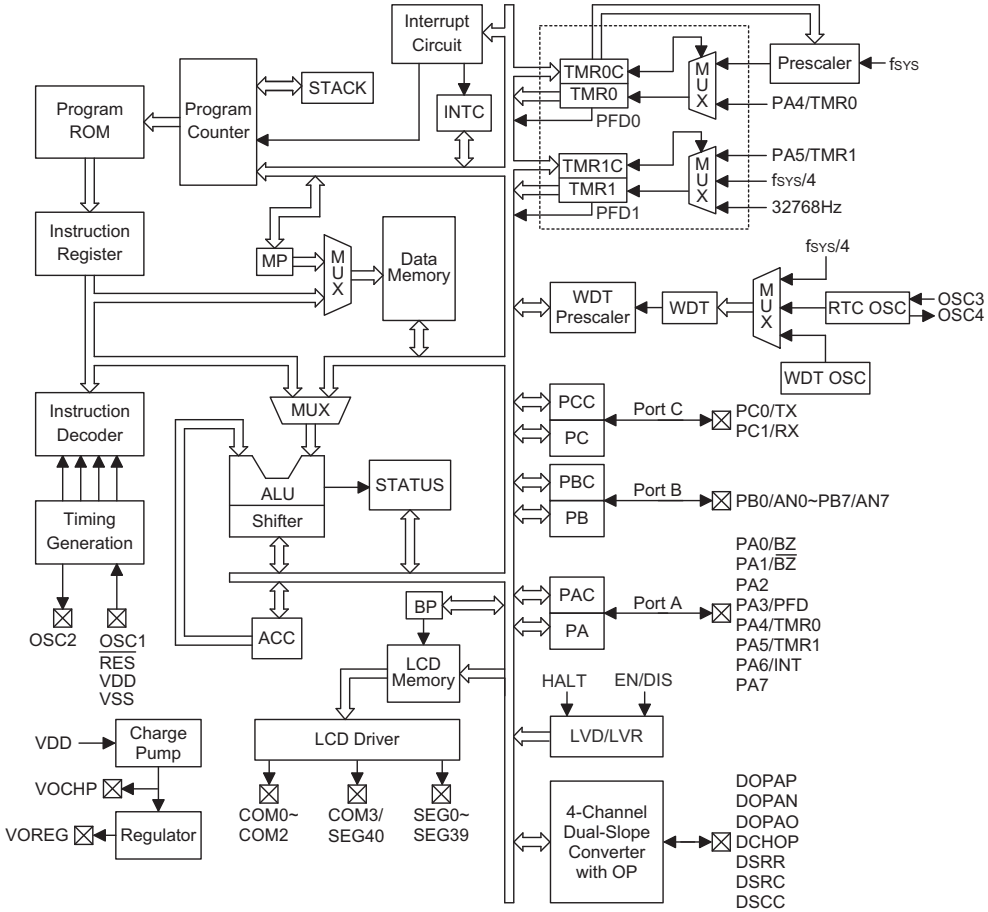
- 工作电压：
  f_{SYS}=4MHz：2.2V～5.5V
  f_{SYS}=8MHz：3.3V～5.5V
- 18 个双向输入/输出口和 2 个 ADC 输入
- 1 个外部中断输入
- 1 个 16 位和 1 个 18 位可编程定时/计数器 ，具有溢出中断和预分频器
- 40×4、41×3、41×2 段的 LCD 驱动可选
- 8K×16 具有部分锁住功能的程序存储器
- 160×8 数据存储器
- 四个带运算放大器的双积分数模转换器
- 看门狗计数器（稳压器输出）
- 蜂鸣器输出
- 32768Hz RTC 振荡
- 内置 RC 或晶体振荡
- HALT 和唤醒功能可降低功耗
- 稳压器（3.3V）和充电泵
- 内置参考电压源（1.5V）
- 16 层硬件堆栈
- 通用异步接收/发送器（UART）
- 位操作指令
- 查表指令，表格内容字长 16 位
- V_{DD}=5V 系统频率为 8MHz 时，指令周期为 0.5μs
- 63 条指令
- 指令执行时间为 1 或 2 个指令周期
- 低电压复位/检测功能
- 100-pin QFP 封装

概述

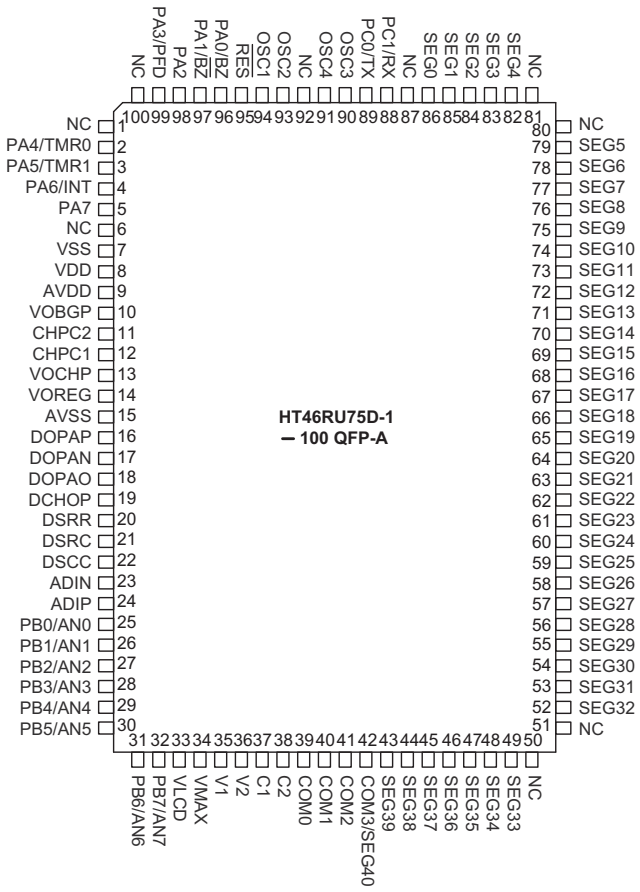
HT46RU75D-1 是 8 位高性能精简指令集单片机，专门为需要 A/D 转换和 LCD 显示的产品而设计。其 AD 模块的输入信号是来自于传感器一类的模拟信号。

因具有低功耗、I/O 使用灵活、计数器、振荡类型选择、双积分 A/D 转换器、LCD 显示、UART 功能、看门狗定时/计数器、暂停和唤醒如此多功能选择及低成本使这款单片机可以广泛应用于需要 A/D 转换和 LCD 显示的产品中，例如电子秤、消费类产品和子系统控制等。

方框图


Interrupt Circuit
Program ROM
Program Counter
STACK
INTC
TMR0C
TMR0
PFD0
TMR1C
TMR1
PFD1
Prescaler
fSYS
PA4/TMR0
PA5/TMR1
fSYS/4
32768Hz
MUX
Instruction Register
MP
Data Memory
WDT Prescaler
WDT
RTC OSC
OSC3
OSC4
WDT OSC
Instruction Decoder
Timing Generation
ALU
Shifter
STATUS
ACC
BP
LCD Memory
LCD Driver
PCC
PC
Port C
PC0/TX
PC1/RX
PBC
PB
Port B
PB0/AN0~PB7/AN7
PAC
PA
Port A
PA0/BZ
PA1/BZ
PA2
PA3/PFD
PA4/TMR0
PA5/TMR1
PA6/INT
PA7
OSC2
OSC1
RES
VDD
VSS
HALT
EN/DIS
LVD/LVR
VDD
Charge Pump
VOCHP
VOREG
Regulator
COM0~ COM2
COM3/ SEG40
SEG0~ SEG39
4-Channel Dual-Slope Converter with OP
DOPAP
DOPAN
DOPAO
DCHOP
DSRR
DSRC
DSCC

引脚图


HT46RU75D-1
100 QFP-A
1 NC
2 PA4/TMR0
3 PA5/TMR1
4 PA6/INT
5 PA7
6 NC
7 VSS
8 VDD
9 AVDD
10 VOBGP
11 CHPC2
12 CHPC1
13 VOCHP
14 VOREG
15 AVSS
16 DOPAP
17 DOPAN
18 DOPAO
19 DCHOP
20 DSRR
21 DSRC
22 DSCC
23 ADIN
24 ADIP
25 PB0/AN0
26 PB1/AN1
27 PB2/AN2
28 PB3/AN3
29 PB4/AN4
30 PB5/AN5
31 PB6/AN6
32 PB7/AN7
33 VLCD
34 VMAX
35 V1
36 V2
37 C1
38 C2
39 COM0
40 COM1
41 COM2
42 COM3/SEG40
43 SEG39
44 SEG38
45 SEG37
46 SEG36
47 SEG35
48 SEG34
49 SEG33
50 NC
51 NC
52 SEG32
53 SEG31
54 SEG30
55 SEG29
56 SEG28
57 SEG27
58 SEG26
59 SEG25
60 SEG24
61 SEG23
62 SEG22
63 SEG21
64 SEG20
65 SEG19
66 SEG18
67 SEG17
68 SEG16
69 SEG15
70 SEG14
71 SEG13
72 SEG12
73 SEG11
74 SEG10
75 SEG9
76 SEG8
77 SEG7
78 SEG6
79 SEG5
80 NC
81 NC
82 SEG4
83 SEG3
84 SEG2
85 SEG1
86 SEG0
87 NC
88 PC1/RX
89 PC0/TX
90 OSC3
91 OSC4
92 NC
93 OSC2
94 OSC1
95 RES
96 PA0/BZ
97 PA1/BZ
98 PA2
99 PA3/PFD
100 NC

引脚说明

引脚名称	输入/输出	掩膜选项	说明
PA0/BZ PA1/$\overline{BZ}$ PA2 PA3/PFD PA4/TMR0 PA5/TMR1 PA6/INT PA7	输入/输出	唤醒功能 上拉电阻 蜂鸣器 PFD 功能	8 位双向输入/输出口。每一位可由掩膜选项设置为唤醒输入。可由软件设置为 CMOS 输出、带或不带上拉电阻(由上拉电阻选项决定：位选择)的斯密特触发输入。BZ/$\overline{BZ}$、PFD、TMR0/TMR1 和 INT 分别与 PA0/1、PA3、PA4/5 和 PA6 共用引脚。
PB0/AN0~ PB7/AN7	输入/输出	上拉电阻	8 位双向输入/输出口。可由软件设置为 CMOS 输出、带或不带上拉电阻（由掩膜选项决定）的斯密特触发输入。PB 和 A/D 输入共用引脚。一旦 A/D 输入引脚由软件指令设置为 A/D 输入，则 I/O 功能和上拉电阻将自动无效。
PC0/TX PC1/RX	输入/输出	上拉电阻	两位双向输入/输出口。可由软件设置为 CMOS 输出、带或不带上拉电阻（由掩膜选项决定）的斯密特触发输入。可由软件设置 PC0 和 PC1 为 I/O 口或 UART TX 输出和 UART RX 输入。
VLCD	—	—	LCD 电源。
VMAX	—	—	IC 最大电压。可与 VDD、VLCD 或 V1 连接
V1,V2,C1,C2	—	—	LCD 电压泵
COM0~COM2 COM3/SEG40	输出	1/2、1/3 或 1/4Duty	COM0~COM3 是 LCD 驱动的 Common 输出。可由配置选项设置 LCD 的占空比。当 1/2 或 1/3duty 被选择，则 COM3/SEG40 口即被设置为 SEG40
SEG0~SEG39	输出	Segment 输出	LCD 驱动的 Segment 输出。
VOBGP	模拟输出	—	Band gap 电压输出(可供内部使用)
VOREG	输出	—	3.3V 稳压输出
VOCHP	输出	—	充电泵输出(需要外接一电容)
CHPC1	—	—	充电泵电容正极
CHPC2	—	—	充电泵电容负极
ADIN ADIP	模拟输出	—	模拟输出连接 OP 外部电阻。ADIN 通过一电阻与 DOPAN 连接。ADIP 通过一电阻与 DOPAP 连接。
DOPAN DOPAP DOPAO DCHOP	模拟输入/ 模拟输出	—	双积分转换器前置放大器 OPA 相关引脚。DOPAN 是 OPA 负输入引脚，DOPAP 是 OPA 正输入引脚，DOPAO 是 OPA 输出引脚，DCHOP 是 OPA 断路器引脚。
DSRR DSRC DSCC	模拟输入/ 模拟输出	—	双积分转换器主要 RC 电路。DSRR 是输入引脚或参考电压引脚，DSRC 是积分器的负极输入， DSCC 是比较器的负极输入。
OSC1 OSC2	输入 输出	晶振或 RC	OSC1、OSC2 连接外部 RC 或晶振，以产生内部系统时钟。如果选择 RC 振荡作为系统时钟，OSC2 是系统时钟四分频的输出口。
OSC3 OSC4	输入 输出	RTC 或系统时钟	OSC3、OSC4 连接 32768Hz 晶振产生一个实时时钟用于定时或作为系统时钟。
$\overline{RES}$	输入	—	斯密特触发复位输入，低电平有效。
VDD	—	—	正电源。
VSS	—	—	负电源。
AVDD	—	—	模拟正电源。
AVSS	—	—	模拟地。

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，ADC 关闭 f_{SYS}=4MHz	—	0.6	1.6	mA
		5V		—	2	4	mA
I_{DD2}	工作电流 (RC 振荡)	3V	无负载，ADC 关闭 f_{SYS}=4MHz	—	0.8	1.5	mA
		5V		—	2.5	4	mA
I_{DD3}	工作电流 (RC 振荡)	3V	无负载，ADC 关闭 f_{SYS}=8MHz	—	2	4	mA
		5V		—	4	8	mA
I_{DD4}	工作电流 (晶体振荡)	5V	无负载，ADC 关闭 f_{SYS}=8MHz	—	4	8	mA
I_{DD5}	工作电流 (RTC 振荡)	3V	无负载，f_{SYS}=32768Hz	—	0.3	0.6	mA
		5V			0.6	1	mA
I_{DD6}	工作电流 (ADC 打开)	5V	V_{REGO}=3.3V,f_{SYS}=4MHz ADC 打开 ADCCLK＝125KHz (其他模拟器件关闭)	—	3	5	mA
I_{STB1}	静态电流 (*f_S=f_{SYS}/4)	3V	无负载，系统 HALT ADC 关闭，LCD 关闭	—	—	1	μA
		5V		—	—	2	μA
I_{STB2}	静态电流 (*f_S=RTC OSC)	3V	无负载，系统 HALT ADC 关闭，LCD 关闭	—	2.5	5	μA
		5V		—	10	20	μA
I_{STB3}	静态电流 (*f_S =WDT OSC)	3V	无负载，系统 HALT ADC 关闭，LCD 关闭	—	2	5	μA
		5V		—	6	10	μA
I_{STB4}	静态电流 (*f_S =RTC OSC)	3V	无负载，系统 HALT，ADC 关闭，LCD 打开 1/2bias，V_{LCD}=V_{DD} （低偏压电流选项）	—	17	30	μA
		5V		—	34	60	μA
I_{STB5}	静态电流 (*f_S =RTC OSC)	3V	无负载，系统 HALT，ADC 关闭，LCD 打开，1/3bias，V_{LCD}=V_{DD} （低偏压电流选项）	—	13	25	μA
		5V		—	28	50	μA

I_{STB6}	静态电流 (*f_S =WDT OSC))	3V	无负载，系统 HALT，ADC 关闭，LCD 打开，1/2bias，$V_{LCD}=V_{DD}$	—	14	25	μA
		5V		—	26	50	μA
I_{STB7}	静态电流 (*f_S =WDT OSC))	3V	无负载，系统 HALT，ADC 关闭，LCD 打开，1/3bias，$V_{LCD}=V_{DD}$	—	10	20	μA
		5V		—	19	40	μA
I_{STB8}	静态电流 (*f_{SYS}=4MHzX'TAL/RC) (WDT 关闭)，f_S=T1	3V	无负载，系统 HALT，LCD 关闭，UART 打开，系统振荡打开	—	—	1200	μA
		5V		—	—	2500	μA
I_{STB9}	静态电流 (*f_{SYS}=8MHzX'TAL/RC) (WDT 关闭)，f_S=T1	3V	无负载，系统 HALT，LCD 关闭，UART 打开，系统振荡打开	—	—	2000	μA
		5V		—	—	4000	μA
V_{IL1}	I/O 口、TMR0、TMR1 及外部中断低电平输入电压	—	—	0	—	0.3 V_{DD}	V
V_{IH1}	I/O 口、TMR0、TMR1 及外部中断高电平输入电压	—	—	0.7 V_{DD}	—	V_{DD}	V
V_{IL2}	低电平输入电压($\overline{RES}$)	—	—	0	—	0.4 V_{DD}	V
V_{IH2}	高电平输入电压($\overline{RES}$)	—	—	0.9 V_{DD}	—	V_{DD}	V
V_{LCD}	LCD 驱动最高电压	—	—	0	—	V_{DD}	V
V_{LVR1}	低电压复位 1	—	LVR 选项=2.2V	1.98	2.1	2.22	V
V_{LVR2}	低电压复位 2	—	LVR 选项=3.3V	2.98	3.15	3.32	V
V_{LVD1}	低电压检测 1	—	LVR 选项=2.2V LVD 选项=LVR+0.2	2.15	2.3	2.45	V
V_{LVD2}	低电压检测 2	—	LVR 选项=3.3V LVD 选项=LVR+0.2	3.2	3.35	3.5	V
I_{OL1}	输入/输出口 Segment 逻辑输出灌电流	3V	V_{OL}=0.1V_{DD}	4	8	—	mA
		5V		10	20	—	mA
I_{OH1}	输入/输出口 Segment 逻辑输出源电流	3V	V_{OH}=0.9V_{DD}	-2	-4	—	mA
		5V		-5	-10	—	mA
I_{OL2}	LCD Common 和 Segment 电流	3V	V_{OL}=0.1V_{DD}	210	420	—	μA
		5V		350	700	—	μA
I_{OH2}	LCD Common 和 Segment 电流	3V	V_{OH}=0.9V_{DD}	-80	-160	—	μA
		5V		-180	-360	—	μA
R_{PH}	输入/输出口带上拉电阻	3V	—	20	60	100	kΩ
		5V	—	10	30	50	kΩ
充电泵和稳压器							
V_{CHPI}	输入电压	—	充电泵打开	2.2	—	3.6	V
			充电泵关闭	3.7	—	5.5	V
V_{REGO}	电压输出	—	无负载	3	3.3	3.6	V

V_{REGDP1}	稳压器输出下降（与无负载时比较）	—	V_{DD} =3.7V~5.5V 电压泵关闭 电流≤10mA	—	100	—	mV
V_{REGDP2}		—	V_{DD} =2.4V~3.6V 电压泵打开 电流≤6mA	—	100	—	mV
双积分 AD 转换器，放大器和带隙							
V_{RFGO}	参考电压发生器输出	—	@3.3V	1.45	1.5	1.55	V
V_{RFGTC}	参考电压发生器温度系数	—	@3.3V	—	50	—	Ppm /C
V_{ICMR}	共模输入范围	—	放大器，无负载	0.2	—	V_{REGO}-1	V
		—	积分器，无负载	1	—	V_{REGO}-0.2	V
V_{ADOFF}	输入偏移范围	—	—	—	500	800	μV

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_{SYS}	系统时钟（RC OSC）	—	2.2V~5.5V	400	—	4000	kHz
	系统时钟（晶体 OSC）	—	2.2V~5.5V	400	—	4000	kHz
		—	3.3V~5.5V	400	—	8000	kHz
f_{INRC}	内部 RC 振荡	3V	—	—	12	—	kHz
		5V		—	15	—	kHz
f_{TIMER}	定时器输入频率 (TMR0/TMR1)	—	2.2V~5.5V	0	—	4000	kHz
t_{WDTOSC}	看门狗振荡器周期	3V	—	45	90	180	μs
		5V	—	32	65	130	μs
t_{RES}	外部复位低电平脉宽	—	—	1	—	—	μs
t_{SST}	系统启动延迟时间	—	上电或从 HALT 状态唤醒	—	1024	—	t_{SYS}
t_{LVR}	复位低电平宽度	—	—	0.25	1	2	ms
t_{INT}	中断脉冲宽度	—	—	1	—	—	μs

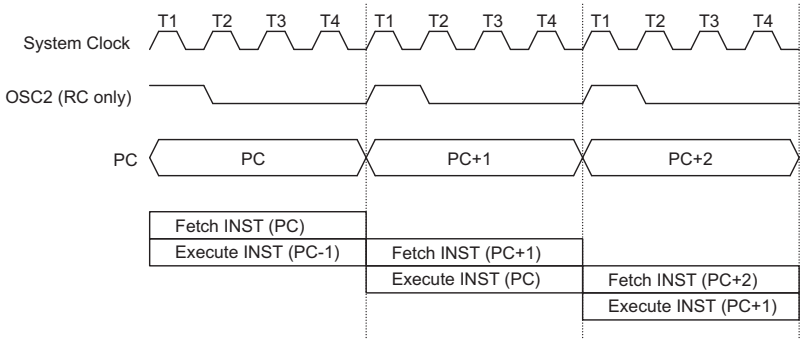
注：t_{SYS} =1/f_{SYS}

系统功能说明

指令执行时序

单片机的系统时钟由一个晶振或一个外部 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

13 位的程序计数器(PC)控制程序存储器 ROM 中指令执行的顺序，它可寻址整个 ROM 范围的 8192 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(程序计数器低字节寄存器)赋值、子程序调用、初始化复位、中断或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												
	***12**	***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	0
外部中断	0	0	0	0	0	0	0	0	0	0	1	0	0
UART 中断	0	0	0	0	0	0	0	0	0	1	0	0	0
定时/计数器 0 溢出	0	0	0	0	0	0	0	0	0	1	1	0	0
定时/计数器 1 溢出	0	0	0	0	0	0	0	0	1	0	0	0	0
ADC 中断	0	0	0	0	0	0	0	0	1	0	1	0	0
RTC 中断	0	0	0	0	0	0	0	0	1	1	0	0	0
条件跳跃	Program Counter+2												
装载 PCL	*12	*11	*10	*9	*8	@7	@6	@5	@4	@3	@2	@1	@0
跳转，子程序调用	#12	#11	#10	#9	#8	#7	#6	#5	#4	#3	#2	#1	#0
子程序返回	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0

程序计数器

注： *12 ~ *0 ： 程序计数器位　　S12~ S0 ： 堆栈寄存器位
#12 ~ #0 ： 指令代码位　　@7 ~ @0 ： PCL 位

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 8192×16 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

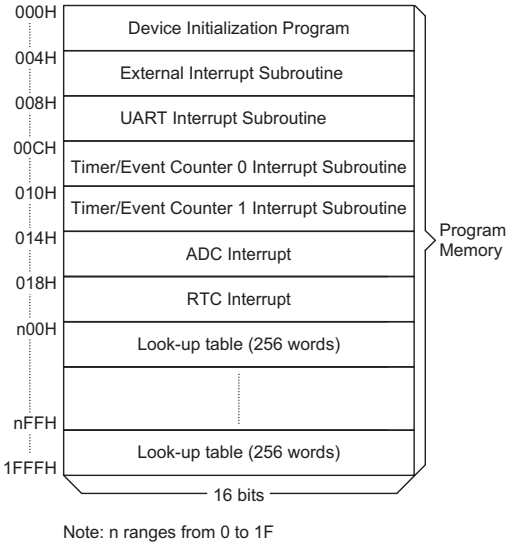
该地址为程序初始化保留。系统复位后，程序总是从 000H 开始执行。

- 地址 004H

该地址为外部中断服务程序保留。当$\overline{INT}$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

此地址保留给 UART 中断服务使用。当中断允许，且堆栈未满，则一旦 UART 的 TX 或 RX 产生中断，程序会从 008H 地址开始执行中断服务程序。



程序存储器

- 地址 00CH

该地址为定时/计数器 0 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 00CH 地址开始执。

- 地址 010H

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 010H 地址开始执行。

- 地址 014H

该地址为 AD 转换中断服务程序保留。当 AD 转换中断发生，如果中断允许且堆栈未满，则程序会跳转到 014H 地址开始执行。

- 地址 018H

该地址为实时时钟中断服务程序保留。当实时时钟中断发生，如果中断允许且堆栈未满，则程序会跳转到 018H 地址开始执行。

- 表格区

ROM 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。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	表格区												
	***12**	***11**	***10**	***9**	***8**	***7**	***6**	***5**	***4**	***3**	***2**	***1**	***0**
TABRDC[m]	P12	P11	P10	P9	P8	@7	@6	@5	@4	@3	@2	@1	@0
TABRDL[m]	1	1	1	1	1	@7	@6	@5	@4	@3	@2	@1	@0

表格区

注：*12 ~ *0 ：表格地址位　　P12 ~ P8 ：当前程序指针位　　@7 ~ @0 ：表格指针位

堆栈寄存器 — STACK

堆栈寄存器是特殊的存储器空间，用来保存 Program Counter 的值。HT46RU75D-1 有 16 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(Program Counter)的值会被压入堆栈；在子程序调用结束或中断响应结束时(执行指令 RET 或 RETI)，堆栈将原先压入堆栈的内容弹出，重新装入程序计数器中。在系统复位后，堆栈指针会指向堆栈顶部。

如果堆栈已满，并且发生了不可屏蔽的中断，那么只有中断请求标志会被记录下来，而中断响应会被抑制，直到堆栈指针(执行 RET 或 RETI 指令)发生递减，中断才会被响应。这个功能可以防止堆栈溢出，使得程序员易于使用这种结构。同样，如果堆栈已满，并且发生了子程序调用，那么堆栈会发生溢出，首先进入堆栈的内容将会丢失，只有最后的 16 个返回地址会被保留。

数据存储器

数据存储器的 Bank 0 由 199×8 位组成，分为两个功能区间：特殊功能寄存器(39×8)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，但有些是只读的。在所有 Bank 中，特殊功能寄存器是重叠的。

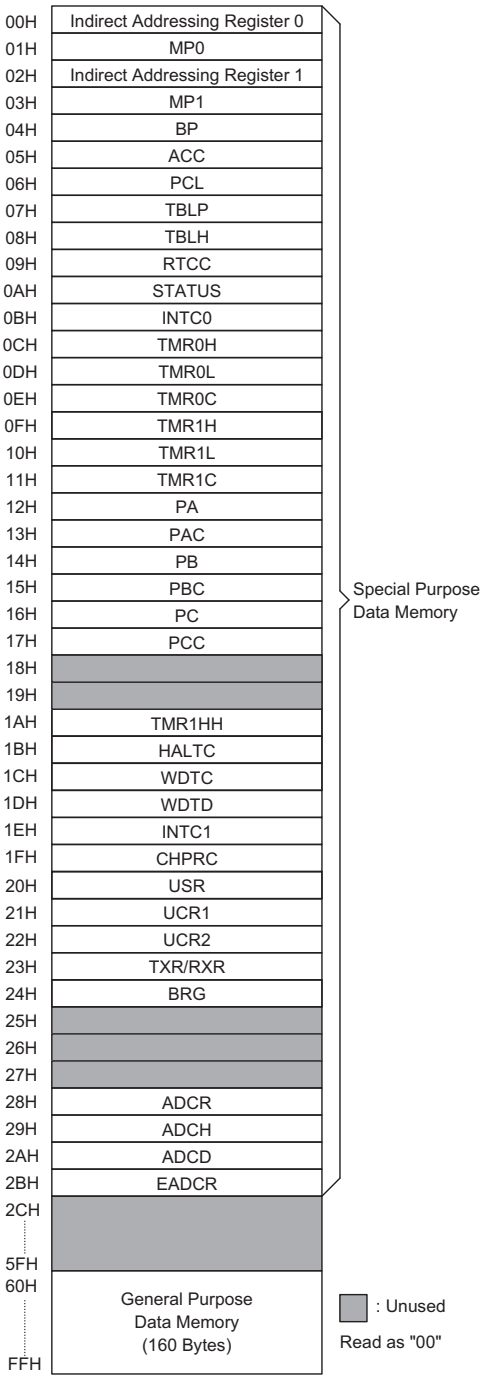
在 60H 之前的空间保留给系统以后扩展使用，读取这些地址的返回值为“00H”。通用数据寄存器地址从 60H 到 FFH，用来存储数据和控制信息。所有的数据存储器单元都能直接执行算术、逻辑、递增、递减和循环操作。除了一些特殊位外，数据存储器的每一位都可由“SET[m].i”置位或由“CLR[m].i”复位。而且都可以通过间接寻址指针(MP0：01H/MP1：03H)进行间接寻址。

Bank1 是 LCD 数据存储区，将 BP 值设为“01H”，便可通过间接寻址指针 MP1 访问 Bank1。当 BP 值为“01H”，通过 MP1 来间接读写地址从 40H 到 68H 的数据存储空间，将会改变 Bank1 中内容。不管 BP 为何值，直接寻址数据寄存器只会改变 Bank0 中内容。

间接寻址寄存器

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]或[02H]的读/写操作，都是访问由 MP0(01H)或 MP1(03H)所指向的 RAM 单元。间接读取地址 00H 或 02H 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，与间接地址寄存器相结合用来访问数据存储器。MP0 只能用于寻址数据存储器，而 MP1 能用于寻址数据存储器和 LCD 显示存储器。



数据存储器

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。

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，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 — STATUS

8 位的状态寄存器(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。

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

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 和 PDF 标志只受看门狗溢出、系统上电、“CLR WDT” 指令或“HALT” 指令的影响。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

HT46RU75D-1 提供一个外部中断、一个 UART 中断、两个内部定时/计数器中断和一个 AD 转换中断。中断控制寄存器 INTC0 和中断控制寄存器 INTC1 都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执行，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、INTC0 或 INTC1 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	总中断控制位(1=允许；0=禁止)
1	EEI	外部中断控制位(1=允许；0=禁止)
2	EURI	UART TX 或 RX 中断控制位（允许=1，禁止=0）
3	ET0I	定时/计数器 0 中断控制位(1=允许；0=禁止)
4	EIF0	外部中断 0 请求标志(1=有；0=无)
5	URF	UART TX 或 RX 中断请求标志(1=有；0=无)
6	T0F	定时/计数器 0 中断请求标志(1=有；0=无)
7	—	只作内部测试用。 使用时必须写入‘0’；否则会发生不可预知的错误

INTC0 （0BH）寄存器

位	符号	功　　能
0	ET1I	定时/计数器 1 中断控制位(1=允许；0=禁止)
1	EADI	A/D 转换中断控制位(1=允许；0=禁止)
2	ERTI	RTC 中断控制位(1=允许；0=禁止)
3, 7	—	未用，读出为“0”
4	T1F	定时/计数器 1 中断请求标志(1=有；0=无)
5	ADF	A/D 转换中断请求标志(1=有；0=无)
6	RTF	RTC 中断请求标志(1=有；0=无)

INTC1 （1EH）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的内容压入堆栈，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 INT 引脚边沿变化触发的(可由掩膜设置为上升沿触发、下降沿触发或两者皆可触发)，它们相关的中断请求标志位(EIF0；INTC0 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 004H 的子程序调用；而中断请求标志 EIF0 和总中断控制位 EMI 会被清除，以禁止其它中断响应。

异步串行中断产生，会置位 UART 中断请求标志（URF，INTC0 的第 5 位），中断请求标志一般是由 UART 发送或接收完成产生的。当中断允许，堆栈未满，并且 URF 已被置位，就会产生 08H 单元的子程序调用。系统将清除中断请求标志位（URF）和 EMI 位，以禁止其他中断的响应。

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；INTC0 的第 6 位)会被置位。如果中断允许，且堆栈未满，当发生定时/计数器中断时，会产生地址 0CH 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。定时/计数器 1 的中断响应过程与定时/计数器 0 相同，只是其中断请求标志为 T1F（INTC1 的第 4 位），中断入口地址为 10H。

A/D 转换中断是由 A/D 转换完成触发的，其中断请求标志(ADF；INTC1 的第 5 位)会被置位。如果中断允许，且堆栈未满，当发生 A/D 转换中断时，会产生地址 14H 的子程序调用；而中断请求标志位 ADF 和总中断控制位 EMI 会被清除，以禁止其它中断响应。

当 RTC 中断请求标志被置起，实时时钟中断将被触发，其中断请求标志(RTF；INTC1 的第 6 位)会被置位。如果中断允许，且堆栈未满，当发生 RTC 中断时，会产生地址 18H 的子程序调用；而中断请求标

志位 RTF 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，其它的中断请求会被屏蔽，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，那么在下两个 T2 脉冲之间，该中断会被服务。如果同时发生中断请求，其优先级如下图所示，这些中断可以通过清除 EMI 位来进行屏蔽。

中断源	优先级	中断向量
外部中断	1	04H
UART 中断	2	08H
定时/计数器 0 中断	3	0CH
定时/计数器 1 中断	4	10H
A/D 转换中断	5	14H
RTC 中断	6	18H

一旦中断请求标志被置位，会一直保留在 INTC0 或 INTC1 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，而且需要立刻给予响应。如果只剩下一层堆栈，而中断不能被很好地控制，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。

振荡电路

HT46RU75D-1 提供三种振荡方式做为系统频率：外部 RC 振荡，外部晶体振荡和 32768Hz 晶体振荡。用户可以通过相关配置选项选择使用。进入省电模式时，系统振荡会停止（32768 除外）以降低功耗。如果选择 32768 晶体振荡做为系统频率，系统进入省电模式后，32768 晶体不会停振，但指令将会停止执行。当 32768 晶体振荡做为计时所用时，系统进入省电模式后，内部计时钟（RTC、时基、WDT）将继续运作。

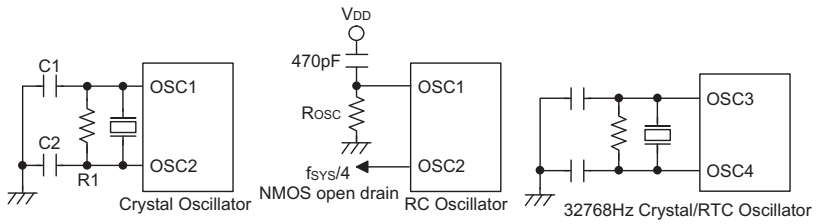
在三种振荡方式中，如果选用外部 RC 振荡方式，在 OSC1 与 VSS 之间需要接一个外部电阻，其阻值为 30kΩ到 750KΩ。OSC2 上拉一电阻到 VDD 可输出系统频率的 4 分频信号，可用于同步外部逻辑。外部 RC 振荡方式提供了一种低成本的振荡方式，但其频率会与 VDD、温度及器件自身参数相关，因此并不适合应用于对时间精度要求高的方案中。

如果选用晶体振荡方式，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的反馈和相移，除此之外，不再需要其它外部元件。另外，在 OSC1 和 OSC2 之间也可使用谐振器来取代晶体振荡器，但是在 OSC1，OSC2 和地之间分别需要 1 个电容。

另外还一个专为实时时钟设计的振荡电路。如果使用 RTC 振荡，那么只要在 OSC3 与 OSC4 之间接一个 32.768kHz 的晶体，不需要其它外部器件。

RTC 振荡器可以通过“QOSC”(RTCC 的第 4 位)设置快速起振。系统上电时默认为开启快速振荡模式，为了降低功耗建议在系统稳定约 2 秒钟后关闭此位。

WDT 振荡是一个独立的内置 RC 振荡电路，不需要外接器件。当系统进入省电模式，系统振荡会停止，但 WDT 振荡会继续作用，其振荡周期一般在 5V 时为 65μs。WDT 振荡器可以由掩膜选项设置为关闭以节省功耗。



系统振荡器

看门狗定时器 — WDT

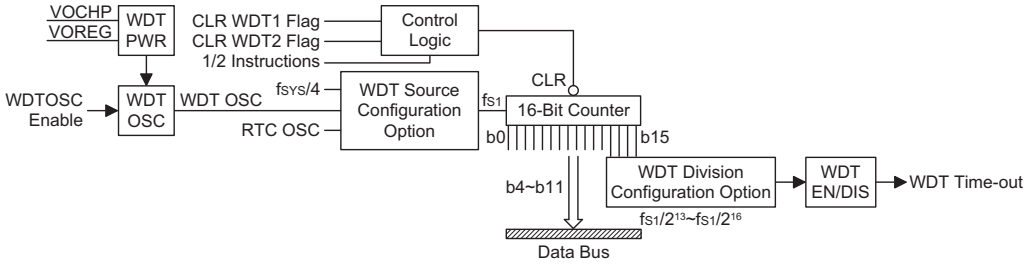
WDT 的时钟来源可由掩膜选项设置为专用 RC 振荡（WDT 振荡器）、指令时钟(系统时钟 4 分频)或实时时钟振荡（RTC 振荡器）。WDT 主要用来防止程序运行故障和程序跳入一死循环而导致不可预测的结果。WDT 可由掩膜选项设置为关闭，如果在关闭状态，WDT 计数器还是会同样工作，只是其计数溢出不会导致器件复位。因此当 WDT 除能时，WDT 计数寄存器还是能被读出来或是被清除。此功能是专门设计，使软件设计者可以通过获取 WDT 的当前频率值来检测当前外部的温度系数是否发生变化，以做为是否调整外部模拟信号量的依据。为了达到降低功耗的目地，可以通过控制寄存器（WDTC：WDTOSC）来选择 WDT 功能的使能或除能。

有两个寄存器与 WDT 相关：WDTC 及 WDTD。其中 WDTC 是控制寄存器，用来选择 WDT 电源的来源及 WDT 功能是否使能。WDTD 是 WDT 计数寄存器。

WDTPWR 位为 WDT 电源选择位，默认的 WDT 电源为 VOCHP。而使用调整器输出电压做为 WDT 电源主要是为了获得一个精确的温度调整系数。当然在此应用前应保证调整器模块是处于工作状态中。WDTOSC 位是用来选择是否使能 WDT OSC (12kHz)。如果没有选用 WDT OSC，那么应该对它除能来降低功耗。

如果 WDT 时钟源为内部 RC 振荡(RC 振荡周期一般在 5V 时为 65μs)，该频率可再经过 2^{12}~2^{15}(由掩膜选项设置)的分频。最小的 WDT 溢出周期大约是 300ms~600ms。但溢出时间会因为温度、VDD 以及芯片参数的变化而变化。如果使用相关的 WDT 掩膜选项，则可以得到更长的溢出周期。如果 WDT 的溢出时间选为 2^{15} 分频，最大的溢出时间可达到 2.3s～4.7s(最大的分频系数为 2^{15}~2^{16})。

如果 WDT 的时钟源为指令时钟，则在 HALT 状态时，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。如果系统运用在强干扰的环境中，建议选用内部 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。



看门狗定时器

在正常运行时，WDT 溢出会使系统复位并置位 TO 标志；但在 HALT 模式或空闲模式下，WDT 溢出只产生“热复位”，只有程序指针 PC 和堆栈指针 SP 被复位。要清除 WDT 的值可以有三种方法：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 果选择“CLR WDT”，那么只要执行“CLR WDT” 指令就会清除 WDT。如果选择“CLR WDT1”和“CLR WDT2”，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。

位	符号	功 能
0~1	WDTPWR0~ WDTPWR1	WDT 电源源选择 01:WDT 电源来自 VOCHP 10:WDT 电源来自于调整器 00/11:保留
2~3	WDTOSC0~ WDTOSC1	WDT振荡器的使能位(WDTOSC1:0)= 01:WDT OSC 关闭 10:WDT OSC 打开 00/11:保留
4~ 7	—	保留

WDTC（1CH）寄存器

注意:如果 WDT 使能及 WDT 时钟来源选为 WDT 时钟，那么 WDTOSC 位的初始值为（1,0），反之被设置为(0,1)。

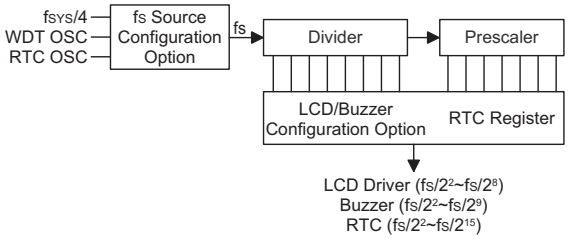
位	符号	功　　能
0~7	WDTD0~ WDTD7	WDT 计数数据寄存器 此寄存器只读，用于温度系数的调整。

WDTD(1DH)寄存器

WDT 时钟(f_{s1})可以通过内部分频以得到一个更久的溢出周期。通过配置选项选择不同的分频系数以得到 2^{13}~2^{16} 之间的 4 个选择。

多功能定时器

系统为 RTC、LCD 和蜂鸣器提供了具有不同溢出周期的多功能定时器。多功能定时器由一个 8 级分配器和一个 7 位预分频器组成。其时钟源可以是 WDT OSC、RTC OSC 或指令时钟（系统时钟四分频）。多功能定时器还为 LCD 驱动电路提供可选择的频率信号（范围从 $f_S/2^2$～$f_S/2^8$），并为蜂鸣器输出电路提供可选择的频率信号（范围从 $f_S/2^2$～$f_S/2^9$），频率由掩膜选项决定。为了正确地显示，建议选择 4kHz 左右的信号作为 LCD 驱动信号。



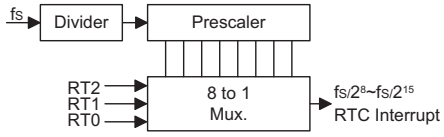
对于升压电路及 ADC 断路器来说，它们的工作频率是独立于多功能定时器的，此部分的频率直接来自于系统频率（RC 或 Crystal）。

实时时钟 — RTC

实时时钟(RTC)是用来提供一个同时基相似的有规律的内部中断。它的溢出周期范围为 $f_S/2^8$～$f_S/2^{15}$，可通过软件编程实现。写数据到 RT2、RT1、RT0（RTCC 的第 2、1、0 位）将产生各种溢出周期。当 RTC 发生溢出，其中断请求标志(RTF；INTC1 第 6 位)被置位，如果中断允许，且堆栈未满，那么就会产生一个地址 18H 的子程序调用。

RT2	RT1	RT0	RTC 实时时钟分频级数
0	0	0	2^8*
0	0	1	2^9*
0	1	0	2^{10}*
0	1	1	2^{11}*
1	0	0	2^{12}
1	0	1	2^{13}
1	1	0	2^{14}
1	1	1	2^{15}

注：“*”不建议使用



实时时钟

蜂鸣器输出

蜂鸣器功能提供一个可变频率输出的功能，以适用于像压电蜂鸣器驱动、或其它需要精确频率输出的场合。BZ 和$\overline{BZ}$是一对和输入/输出引脚 PA0 与 PA1 共用的蜂鸣器输出。可通过掩膜选项选择 PA0 和 PA1 都是普通输入/输出、BZ 和$\overline{BZ}$输出、以及单一 BZ 输出而 PA1 作为普通输入/输出。要注意的是$\overline{BZ}$引脚是 BZ 引脚的反向输出，两者配合可产生更强的输出功率驱动蜂鸣器等器件。

蜂鸣器时钟来源为内部时钟 f_S 经过分频器得到，其分频系数可通过掩膜选项选择为 $f_S/2^2$ 到 $f_S/2^9$ 之间。而 f_S 可通过掩膜选项选择为内部 RC 振荡源或系统时钟四分频。需要注意的是蜂鸣器输出频率是完全由掩膜选项控制，包括其时钟 f_S 来源和分频系数，没有特殊功能寄存器与蜂鸣器频率有关。

如果掩膜选项确定 PA0 和 PA1 是 BZ 和$\overline{BZ}$蜂鸣器输出，必须将 PA 控制寄存器 PAC0 和 PAC1 设为输出。若要蜂鸣器正常工作，需要将 PA0 置为高；若清零，则 PA0 和 PA1 都输出低电平。这个输出数据位被用来作为 BZ/$\overline{BZ}$输出开关控制。注意，PA1 对$\overline{BZ}$输出控制不起作用。

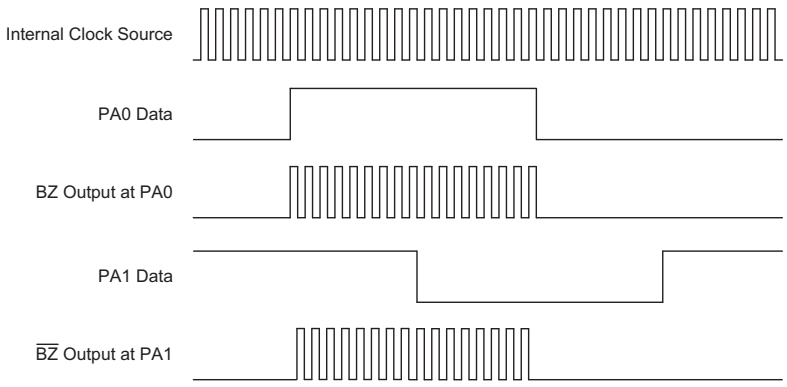
如果掩膜选项确定 PA0 为 BZ 输出，那么 PA1 可被用作普通输入/输出。蜂鸣器正常工作，需要通过 PAC0 将 PA0 设为输出。若 PA 寄存器中 PA0 置为高，则引脚 PA0 可输出蜂鸣器驱动输出；若清为零，则引脚 PA0 保持低电平。这样输出数据位 PA0 被用来作为 BZ 输出开关控制。若掩膜选项中选择为 BZ 输出，而 PA 控制寄存器 PAC0 置为高，则 PA0 可被用作普通的输入口。

注意，若将相应口设为输入，无论掩膜选项中选择如何，都会作为普通输入使用。这样可使得这些引脚既可作为蜂鸣器输出，也可作为输入使用；实际功能可通过程序修改端口控制寄存器完成。

PAC 寄存器 PAC.0	PAC 寄存器 PAC.1	PA 寄存器 PA.0	PA 寄存器 PA.1	输出功能
0	0	0	X	PA0=0，PA1=0
0	0	1	X	PA0=BZ，PA1=$\overline{BZ}$
0	1	0	X	PA0=0，PA1=Input
0	1	1	X	PA0=BZ,PA1=Input
1	0	0	X	PA0=Input,PA1=0
1	1	X	X	PA0=Input,PA1=Input

PA0/PA1 引脚功能

注意：“X”表示任意值



蜂鸣器输出

上图中 PA0 和 PA1 都被选择为 BZ 和$\overline{BZ}$输出，其端口控制寄存器都被设置为输出。PA1 位对蜂鸣器输出不起作用。

暂停模式 — HALT

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 OSC 振荡器继续振荡(如果选择 WDT OSC 振荡或 RTC)。
- 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 OSC 振荡或 RTC 振荡)。
- 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。
- LCD 驱动器仍然运行（如果其时钟来源于 WDT OSC 或 RTC OSC）。

以下操作可以使系统离开暂停：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，但只有程序计数器 Program Counter 和堆栈指针 SP 被复位，其它都保持其原有的状态。

PA 口唤醒和中断唤醒可做为正常运行的继续。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。

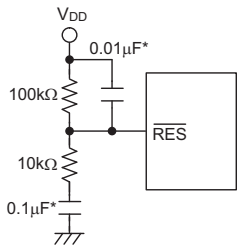
当HALT 指令被执行后，CPU 即将停止运行，与其相关的 OSC 及周边时钟状态可以通过寄存器 HALTC 来设置。对 HALTC 寄存器的操作也仅在系统频率选择 OSC CLK 时才会有效。

注意：当选择 32KHz 做为系统频率时，寄存器 HALTC 无意义。

位	符号	功　　能
0	LCDON	省电模式下 LCD 的状态： 1:不管配置选项如何设置，LCD 模块维持原态（如果 OSCON＝1） 0:LCD 模块状态由 LCD 相关配置选项而定
1	UARTON	在暂停模式下 UART 的状态： 1：UART 模块保持开启（如果 OSCON=1） 0：UART 关闭
2~6	—	未用，读取值为 0
7	OSCON	省电模式下，系统振荡器状态： 0:振荡器停止振荡。所有相关的周边单元停止工作（LCDON 位无效） 1:振荡器维持振荡。（除了 LCDON 定义的 LCD 状态外，其他相关功能部分继续工作）

HALTC(17H)寄存器

复位



复位电路

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

总共有三种方法会产生初始复位：

- 正常运行时由$\overline{RES}$引脚发生复位。
- 在暂停模式由$\overline{RES}$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 Program Counter 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。

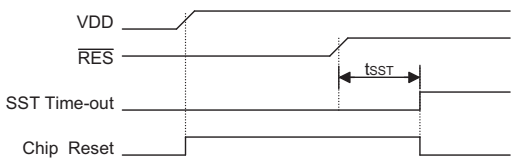
TO	PDF	复位原因
0	0	上电时$\overline{RES}$发生复位
u	u	正常运行时$\overline{RES}$发生复位
0	1	暂停模式下$\overline{RES}$发生复位
1	u	正常运行时 WDT 溢出
1	1	暂停模式下 WDT 溢出

注：“u”表示不变

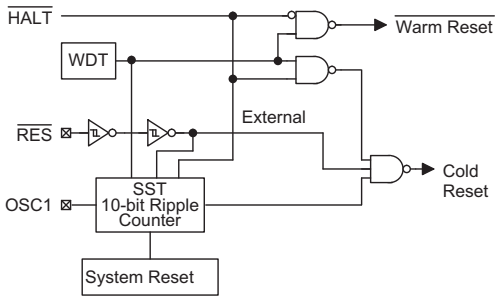
为了保证系统振荡器起振并稳定运行，系统复位(包括上电复位、WDT 溢出或由$\overline{RES}$端复位)或由暂停状态唤醒时，系统启动定时器(SST)提供了一个额外的延迟时间，共 1024 个系统时钟周期。系统复位时，SST 会被加在复位延时中；由暂停模式唤醒也会加入 SST 延迟。系统复位(包括上电复位、正常运行时 WDT 溢出或由$\overline{RES}$端复位)需要额外增加一个加载掩膜选项的时间。

系统复位时各功能单元的状态如下所示：

Program Counter	000H
中断	禁止
预分频器、分配器	清除
WDT	清除，在主系统复位后，WDT 开始计数
定时/计数器	停止
输入/输出口	输入模式
堆栈指针 SP	指向堆栈顶部



复位时序图



复位配置

有关寄存器的状态如下：

寄存器	复位(上电)	WDT 溢出 (正常运作)	$\overline{RES}$复位 (正常运作)	$\overline{RES}$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	---- ---u
ACC	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
Program Counter	0000H	0000H	0000H	0000H	0000H
TBLP	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TBLH	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
RTCC	--00 0111	--00 0111	--00 0111	--00 0111	--uu uuuu
STATUS	--00 xxxx	--lu uuuu	--uu uuuu	--01 uuuu	--11 uuuu
INTC0	-000 0000	-000 0000	-000 0000	-000 0000	-uuu uuuu
TMR0H	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR0L	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR0C	00-0 1000	00-0 1000	00-0 1000	00-0 1000	uu-u uuuu
TMR1H	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR1L	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TMR1C	0000 1---	0000 1---	0000 1---	0000 1---	uuuu 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	---- --11	---- --11	---- --11	---- --11	---- --uu
PCC	---- --11	---- --11	---- --11	---- --11	---- --uu
TMR1HH	---- --xx	---- --uu	---- --uu	---- --uu	---- --uu
HALTC	0--- --00	0--- --00	0--- --00	0--- --00	u--- --uu
WDTC	---- ss01	---- ss01	---- ss01	---- ss01	---- uuuu
WDTD	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000
INTC1	-000 -000	-000 -000	-000 -000	-000 -000	-uuu -uuu
CHPRC	0000 0000	0000 0000	0000 0000	0000 000	uuuu uuuu
USR	0000 1011	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
UCR1	0000 0x00	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
UCR2	0000 0000	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
TXR/RXR	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
BRG	xxxx xxxx	uuuu uuuu	uuuu uuuu	uuuu uuuu	uuuu uuuu
ADCR	-000 x000	-000 x000	-000 x000	-000 x000	-000 x000
ADCH	-000 --00	-000 --00	-000 --00	-000 --00	-000 --00
ADCD	---- -111	---- -111	---- -111	---- -111	---- -uuu
EADCR	0000 0000	0000 0000	0000 0000	0000 0000	uuuu uuuu

注： 1.“*”表示“热复位；

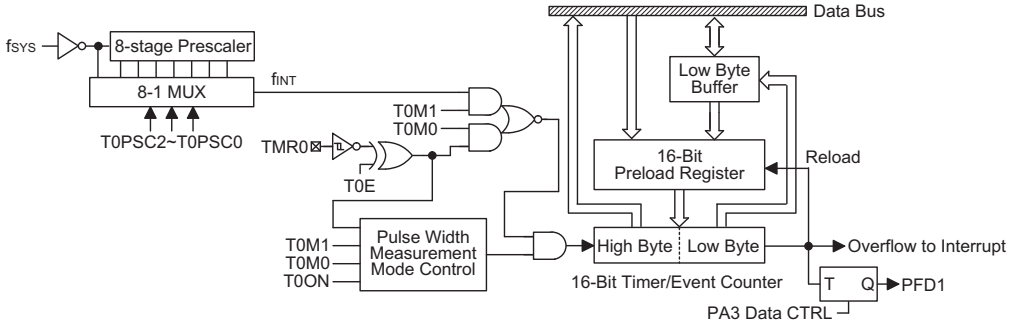
2.“u”表示不变化；

3.“x”表示不确定；

4.“s”要特别留意，它由选项表格决定(详情请看 WDT 部分的内容)。

定时/计数器

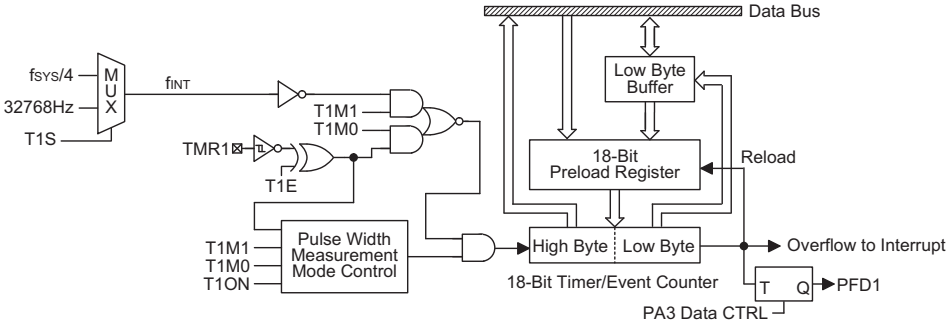
HT46RU75D-1 提供两个定时/计数器。定时/计数器 0 是一个 16 位可编程向上计数器，它的时钟来源可以是外部信号输入或内部时钟，内部时钟源来自于 f_{SYS}。定时/计数器 1 是一个 18 位可编程向上计数器，其时钟来源可以是外部信号输入或内部时钟。其内部时钟源可通过配置选项选择 $f_{SYS}/4$ 或 32768Hz。外部信号输入可以用来计数外部事件、测量时间间隔、测量脉冲宽度或产生一个精确的时基信号。



定时/计数器 0

位	符号	功能
0 1 2	T0PSC0 T0PSC1 T0PSC2	定义预分频器级数，T0PSC2，T0PSC1，T0PSC0= 000：$f_{INT}=f_{SYS}$ 001：$f_{INT}= f_{SYS}/2$ 010：$f_{INT}= f_{SYS}/4$ 011：$f_{INT}= f_{SYS}/8$ 100：$f_{INT}= f_{SYS}/16$ 101：$f_{INT}= f_{SYS}/32$ 110：$f_{INT}= f_{SYS}/64$ 111：$f_{INT}= f_{SYS}/128$
3	T0E	定义定时/计数器 TMR0 的触发方式 在事件计数模式（T0M1，T0M0）=（0，1）： 1：在下降沿计数 0：在上升沿计数 在脉冲宽度测量模式（T0M1，T0M0）=（1，1）： 1：在上升沿开始计数，下降沿停止计数 0：在下降沿开始计数，上升沿停止计数
4	T0ON	打开/关闭定时/计数器(0=关闭，1=打开)
5	—	保留位，读取值为 0
6 7	T0M0 T0M1	定义工作模式(T0M1，T0M0)： 01=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(外部时钟) 00 =未用

TMR0C（0EH）寄存器



定时/计数器 1

位	符号	功能
0	T132KON	定义 32768 振荡器是否起振 0:32768 振荡器停振，如果没有相关功能在动作 1:开机后 32768 一直维持振荡
1~2	—	未用，读取值为 0
3	T1E	定义定时/计数器 TMR1 的触发方式 在事件计数模式（T1M1，T1M0）=（0，1）： 1：在下降沿计数 0：在上升沿计数 在脉冲宽度测量模式（T1M1，T1M0）=（1，1）： 1：在上升沿开始计数，下降沿停止计数 0：在下降沿开始计数，上升沿停止计数
4	T1ON	打开/关闭定时/计数器(0=关闭，1=打开)
5	T1S	TMR1 时钟来源选项(0= f_{SYS}/4，1=32768Hz)
6 7	T1M0 T1M1	定义工作模式(T1M1，T1M0)： 01=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(外部时钟) 00 =未用

TMR1C（11H）寄存器

注： 32768 振荡器是否工作是由 T132KON 位或所有相关的掩膜选项决定的，也就是说任何相关的功能使能的话，32768 振荡就工作，反之如果没有任何相关的功能动作，32768 振荡就被关闭。

有三个与定时/计数器 0 有关的寄存器，TMR0H，TMR0L 和 TMR0C(0EH)。写入 TMR0L 会将初始值装入到内部低字节 8 位缓冲器中，而写 TMR0H 会把指定数据和低字节缓冲器的数据分别写到 TMR0H 和 TMR0L 寄存器中。定时/计数器 0 预制寄存器数据随每次写入 TMR0H 而变化。读 TMR0H 会把 TMR0H 和 TMR0L 的数据分别锁存到指定位置和低字节缓冲器中。而读 TMR0L 只能读取低字节缓冲器的数据。TMR0C 是定时/计数器 0 控制寄存器，用来定义定时/计数器 0 一些选项。有四个与定时/计数器 1 有关的寄存器，TMR1HH、TMR1H、TMR1L 和 TMR1C。写 TMR1L 和 TMR1H 只能将数据写到低字节缓冲器，而写 TMR1HH 会把指定数据和低字节缓冲器的数据分别写到 TMR1HH 和 TMR1H、TMR1L 预置寄存器中。

定时/计数器 1 预置寄存器的内容只有在写入 TMR1HH 时才会被改变，读取 TMR1HH 会把 TMR1HH 的内容送至目标单元而 TMR1H 及 TMR1L 的值被送至低字节缓冲器中。读 TMR1H 及 TMR1L 寄存器会读取低字节缓冲器的值。TMR1C 是定时/计数器 1 控制寄存器，用来定义定时/计数器 1 一些选项。

T0M0、T0M1(TMR0C)和 T1M0、T1M1(TMR1C)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（TMR0、TMR1）引脚输入。定时器模式是一个常用模式，其时钟来源为内部时钟。脉宽测量模式可以测量 TMR0 或 TMR1 引脚高/低电平的脉冲宽度，其时钟来源为内部时钟。

要启动计数器，只要置位计数控制器相应控制位为 1（T0ON:TMR0C 的第 4 位；T1ON:TMR1C 的第

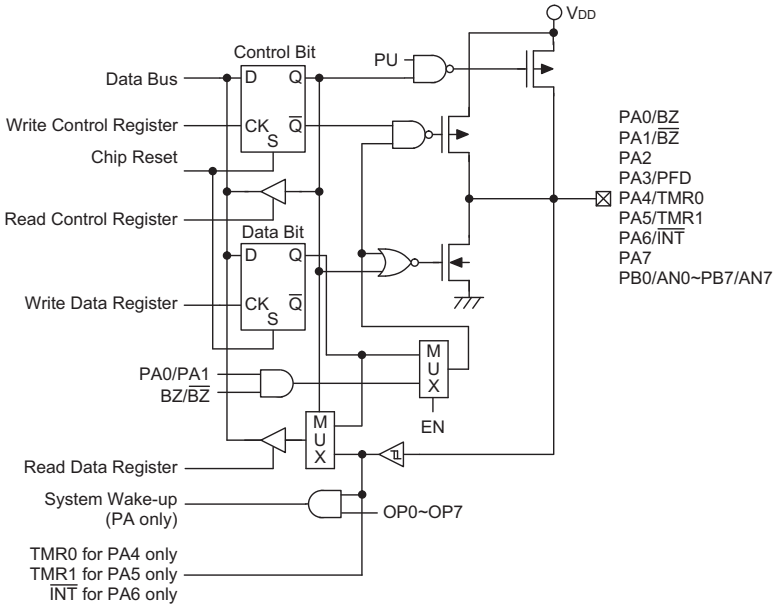
4 位)。在脉宽测量模式下，T0ON/T1ON 在测量结束后会被自动清除；但在另外两种模式中，T0ON/T1ON 只能由指令来清除。定时/计数器溢出可以做为唤醒信号。定时/计数器也能做为 PFD 的时钟源（频率可变），此功能由配置选项来决定，且同一时刻仅有一个定时/计数器被选定为 PFD 的时钟源，设计者可以通过配置选项(PFD0 或 PFD1)进行设置。如果 PA3 做为 PFD 输出，会有两种情况发生：一是选择 PFD0 做为 PFD 输出，另一种是选择 PFD1 做为 PFD 输出。PFD0 和 PFD1 是定时/计数器 0/1 的溢出信号。不管是什么模式，只要写 0 到 ET0I/ET1I 即可禁止定时/计数器 0/1 中断服务。当 PFD 功能被选中后，执行 SET [PA].3 指令即使能 PFD 输出，反之执行 CLR [PA].3 指令即停止 PFD 输出

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中，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。因为 TMR0/TMR1 的初始值不确定，为了保证正确的操作强烈建议先将数据写入 TMR0 和 TMR1 寄存器中，再打开相应的定时/计数器。

TMR0C 的第 0~2 位用来定义内部时钟预分频级数。定时/计数器的溢出信号可用来产生 PFD 信号。

输入/输出口

HT46RU75D-1 有 18 位双向输入/输出口，记为 PA、PB 和 PC，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] ”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 和 PCC)，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。做为输入时，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。

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。

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。

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。

PA0、PA1、PA3、PA4、PA5 和 PA6 分别和 BZ、$\overline{BZ}$、PFD、TMR0、TMR1 和 INT 共用引脚。

PA0、PA1 分别与 BZ、$\overline{BZ}$共用引脚，如果选择 BZ/$\overline{BZ}$功能，则 PA0/PA1 在输出模式时的输出信号将是蜂鸣器信号，而在输入模式始终保持其原来的功能。一旦选择 BZ/$\overline{BZ}$功能，蜂鸣器的输出信号只受 PA0 数据寄存器控制。

PA0/PA1 的输入/输出功能如下所示：

PA0 I/O	I	I	O	O	O	O	O	O	O	O
PA1 I/O	I	O	I	I	I	O	O	O	O	O
PA0 模式	X	X	C	B	B	C	B	B	B	B
PA1 模式	X	C	X	X	X	C	C	C	B	B
PA0 数据	X	X	D	0	1	D_0	0	1	0	1
PA1 数据	X	D	X	X	X	D_1	D	D	X	X
PA0 引脚状态	I	I	D	0	B	D_0	0	B	0	B
PA1 引脚状态	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/$\overline{BZ}$

“X”任意值

“C”CMOS 输出

PA3 与 PFD 共享引脚。如果选为 PFD 功能，则在 PA3 设置为输出的模式下，当定时/计数器溢出信号产生时，将产生 PFD 信号。在输入模式下，该引脚保持其通常功能。一旦选为 PFD 功能，PFD 输出信号仅受 PA3 数据寄存器的控制。写“1”到 PA3 则打开 PFD 输出功能，而写“0”则使 PA3 保持“0”。PA3 输入/输出功能如下表所示。

I/O 模式	**I/P (正常工作状态)**	**O/P (正常工作状态)**	**I/P (PFD)**	**O/P (PFD)**
PA3	逻辑输入	逻辑输出	逻辑输入	PFD(定时器打开)

注：PFD 频率是定时/计数器溢出频率的 1/2。

PFD 控制信号和 PFD 输出频率之间的关系如下表所示。

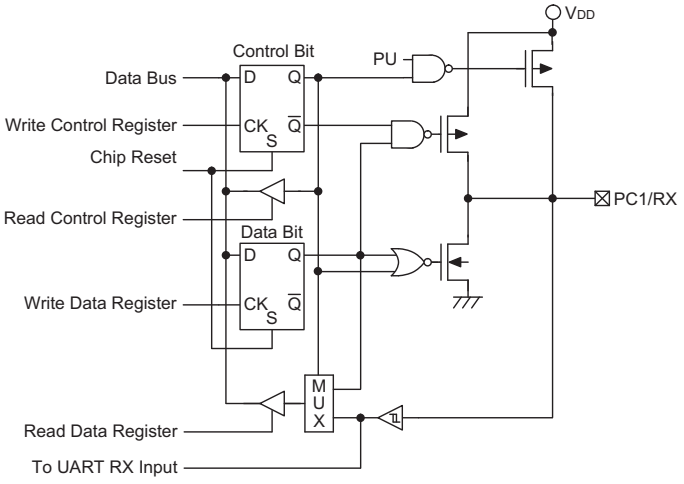
定时器	**定时器 预制寄存器值**	**PA3 数据寄存器**	**PA3 引脚状态**	**PFD 频率**
关闭	X	0	0	X
关闭	X	1	U	X
打开	N	0	0	X
打开	N	1	PFD	$f_{TMR}/[2\times(M-N)]$

注：“X”表示未用；“U”表示未知；“M”表示最大值为 65536；

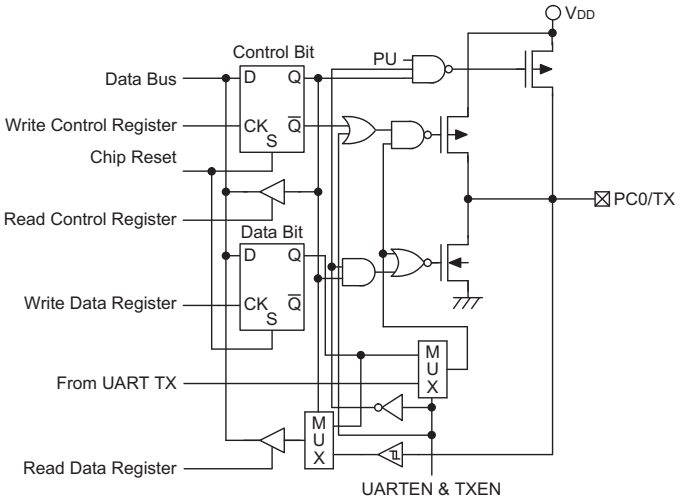
“N”表示定时/计数器的预制寄存器值；

“f_{TMR}”表示定时/计数器的输入时钟频率

建议用软件将未使用和没有外接的输入/输出口设置为输出模式，以防止这些端口在输入浮空时增加系统的功耗。



PC1/RX 结构图

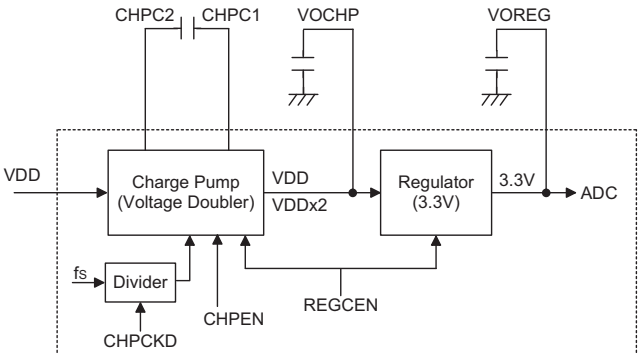


PC0/TX 结构图

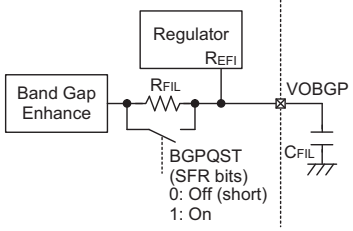
充电泵和稳压器

HT46RU75D-1 提供一个充电泵和稳压器。

充电泵可由程序控制打开或关闭，它由 VDD 作为输入，可输出 2 倍 V_{DD} 电压（即 $2\times V_{DD}$）。稳压器用于产生稳定的 3.3V 电压，作为 AD 转换电路和外部传感器电源，也可供给其它相关电路使用。为确保稳压器输出为稳定的 3.3V，必须要使充电泵输出大于 3.6V。



另外，还提供了一个 1.5V 的 band gap 电压发生器，做为低温条件下的参考电压。这个电压相当于在零校准时引进的一个直流参考电压，R_{FIL} 等效值约为 100 kΩ，C_{FIL} 的推荐值为 10μF。



注： VOBGP 为内部信号，外部仅需要连接器件 C_{FIL}

有一寄存器 CHPRC 与此模块相关。它是充电泵/稳压器控制寄存器，可以控制充电泵和稳压器的打开与关闭，以及设置充电泵时钟分频系数。

位	符号	功能
0	REGCEN	打开或关闭充电泵/稳压器（1=打开；0=关闭）
1	CHPEN	打开或关闭充电泵（1=打开；0=关闭） 注意：若 REGCEN=0，则此位无效
2	BGPQST	Band gap 电压快速起动功能 0：电阻被短路，快速起动 1：电路接入，正常的 RC 模式 为了保证快速起动成功，每次 REGCEN 位从 0 到 1，BGPQST 位也必执行从 0 到 1 的动作（稳压器打开），且 0 状态最少保持 2ms 时间。
3~7	CHPCKD0~ CHPCKD4	充电泵时钟分频，此 5 位可产生 1~32 的分频系数 其等式：充电泵时钟=（f_{SYS} /16）/（CHPCKD+1）

CHPRC（1FH）寄存器

CHPCKD4~0 位用于设置充电泵工作时钟分频。实际频率计算公式为：

实际充电泵时钟频率=（f_{SYS} /16）/（CHPCKD+1）。

建议充电泵工作频率为 20kHz，应用程序需要设置正确值得到预期的频率。例如：若时钟为 4MHz 时，CHPCKD 应为 11；若时钟为 2MHz 时，CHPCKD 应为 5。

REGCEN	CHPEN	电压泵	VOCHP 引脚	稳压器	VOREG 引脚	OPA ADC	描述
0	X	OFF	V_{DD}	OFF	高阻态	关闭	整个模块关闭，OPA/ADC 掉电
1	0	OFF	V_{DD}	ON	3.3V	打开	V_{DD} 高于 3.6V(V_{DD} >3.6V)
1	1	ON	2×V_{DD}	ON	3.3V	打开	V_{DD} 低于 3.6V(V_{DD} =2.2V~3.6V)

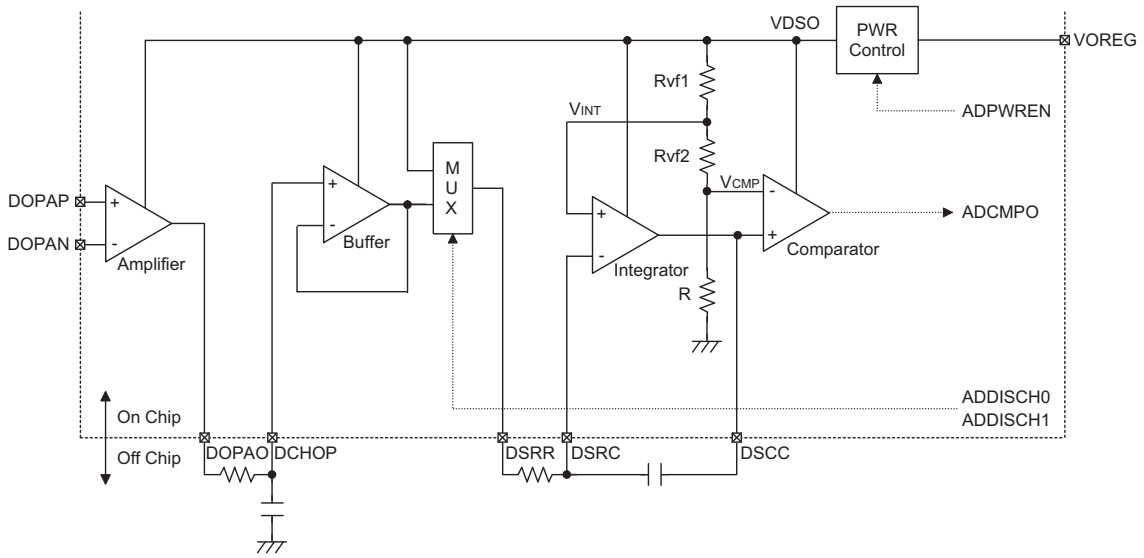
寄存器 CHPRC 中 REGCEN 位是充电泵/稳压器打开/关闭控制位。若关闭，则充电泵和稳压器都会被关闭以降低功耗。当 REGCEN=0，此模块将进入掉电模式并忽略 CHPEN 位，AD 转换和 OPA 也会关闭以降低功耗。

当 REGCEN=1，稳压器将会使能。若 CHPEN=1，充电泵使能并且利用 VDD 作为输入来产生 2 倍的 VDD 电压输出，此输出电压作为稳压器输入。若 CHPEN=0，充电泵除能并且输出电压等于输入电压（VDD）。

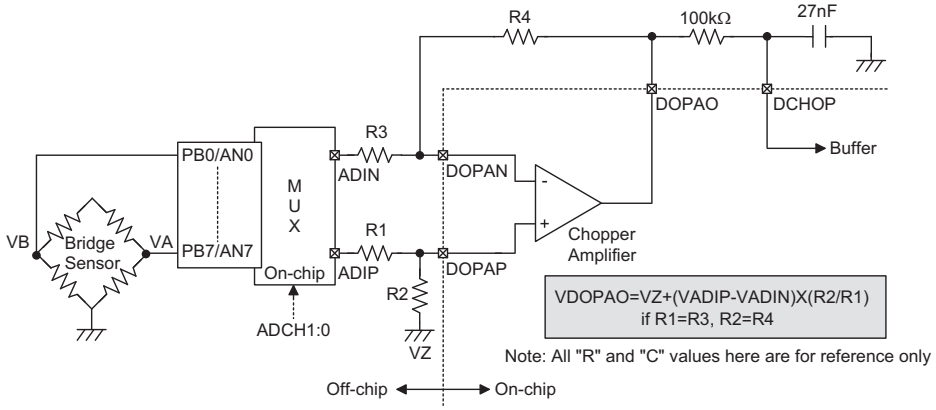
必须注意 V_{DD} 电压，若小于 3.6V，则需将 CHPEN 置 1 使能充电泵；否则需将 CHPEN 清 0。若充电泵除能且 V_{DD} 电压低于 3.6V，则稳压器输出将得不到保证。

AD 转换器 — 双积分

HT46RU75D-1 提供一个双积分 AD 转换器。它包括一个放大器、电压跟随器、积分器和比较器。



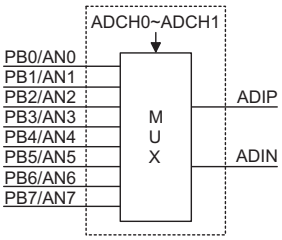
注：V_{INT} 和 V_{CMP} 信号可通过寄存器设置，使其来自不同电阻分压



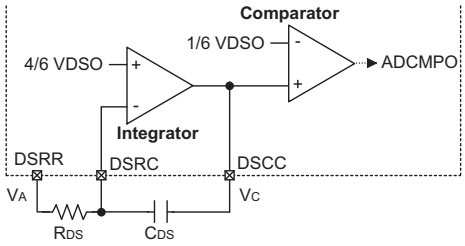
有四个与 AD 转换有关的寄存器，ADCR、 ADCD、ADCH 和 EADCR。ADCR 寄存器是 AD 转换的控制寄存器，用来控制 AD 转换的开/关，断路器的开/关，充电/放电以及用于读取比较器输出状态。ADCD 寄存器用于设置断路器时钟分频。EADCR 寄存器是 AD 双积分自动模式的控制器。ADCH A/D 通道选择寄存器，用来控制 PB 口 A/D 功能的开/关和选择 AD 转换输入的通道。

ADPWREN 位用于控制 AD 转换器开/关。ADCCKEN 位用于控制断路器时钟开/关。当 ADCCKEN=1，断路器时钟将打开，其时钟频率由 ADCD 寄存器设置。AD 转换器包含 OPA、缓冲器、积分器和比较器。而 band gap 电压发生器是一个独立部分，它会自动根据稳压器使能而使能，反之亦然。应用程序可控制各功能打开与关闭，以降低功耗。充放电控制位 ADDISCH1~0 用于控制双积分电路充电与放电。ADCMPO 位是只读位，用于表示比较器输出，而 ADINTM 位用来设置产生 AD 转换中断的 ADCMPO 触发模式。

PB 口可以设置为 A/D 转换输入或作为普通双向 I/O 口。PCR2~0 位用来定义 A/D 输入个数。ADCH1~0 用来定义 ADIP 和 ADIN 输入组合。



下面的描述基于 ADRR0=0。放大器和缓冲器组成双端输入的前置放大器将传感器信号进行放大。



积分器、比较器、DSRR 与 DSRC（R_{ds}）之间电阻、DSRC 与 DSCC（C_{ds}）之间电容组成双积分 AD 转换主要部分。

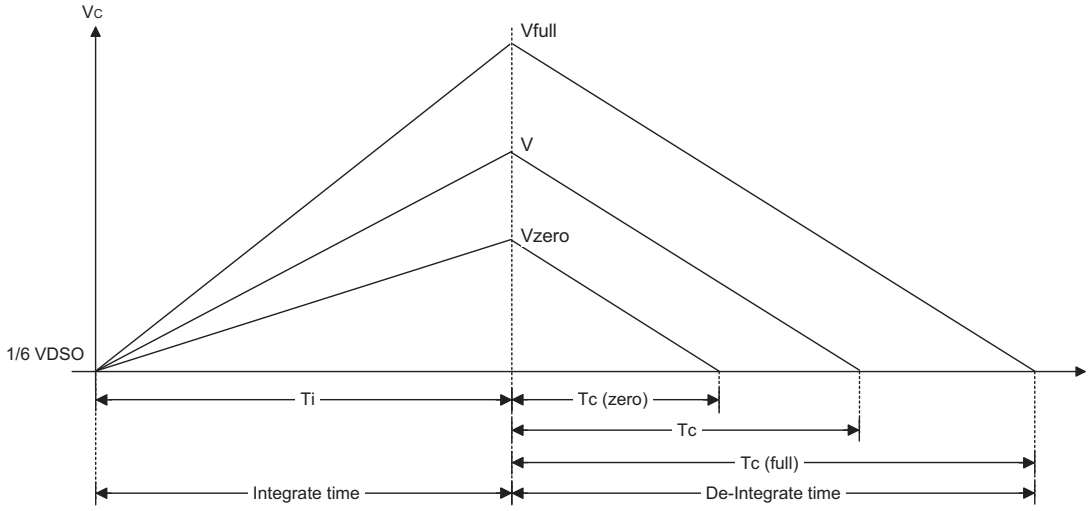
积分器输出电压的上升或下降是由“电流开关”-如下面结构图所示。充放电曲线如下举例所示。

当 VC（DSCC 引脚电压）下降低于 1/6VDSO 时，比较器输出将会由高变低。

在一般的应用中，程序可通过开关控制 AD 转换器进行固定时间（Ti）积分，然后切换到放电模式。当 Vc 降至低于 1/6VDSO（比较器会改变状态）的这段时间记为反向积分时间（Tc）。通过公式 1 可计算出输入电压 V_A。

公式 1：V_A=（1/3）×VDSO×（2-Tc/Ti）。
（ADRR0=0 时）

提示：必须注意选择正确的 R_{DS}，C_{DS} 来确定 T_i，使 V_C 介于 5/6VDSO 和 1/6VDSO 之间（例如：V_{FULL} 不能高于 5/6VDSO，V_{ZERO} 不能低于 1/6VDSO）



位	符号	功能
0 1	ADCH0 ADCH1	OPA 输入通道选择寄存器。这个寄存器可设置输入信号的组合。细节请查看如下表格。
2~3，7	—	保留
4~6	PCR0~PCR2	端口 B：A/D 通道（OPA 输入）/通用输入/输出设置

ADCH(29H)寄存器

ADCH1~ADCH0	输入通道（ADIP/ADIN）	描述
00	AN0/AN1	微分
01	AN2/AN3	微分
10	AN4/AN5	微分
11	AN6/AN7	微分

OPA 输入通道设置

PCR2~PCR0	
000	端口 B A/D 通道—全部关闭
001	保留
010	PB0~PB1 作为 A/D 通道打开
011	保留
100	PB0~PB3 作为 A/D 通道打开
101	保留
110	PB0~PB5 作为 A/D 通道打开
111	PB0~PB7 作为 A/D 通道打开

端口 B 的设置

位	符号	功能
0	ADPWREN	双积分电路电源开关 0：电源关闭 1：电源来自稳压器
1~2	ADDISCH0~ ADDISCH1	设置 AD 转换器充电/放电： 00：保留 01：充电（积分器输入连接至缓冲器输出） 10：放电（积分器输入连接至 VDSO） 11：保留
3	ADCMPO	双积分 AD 转换器---比较器输出 只读，写此位指令被忽略 在放电阶段，当积分器输出电压低于参考电压，ADCMPO 将由高变低
4~5	ADINTM0~ ADINTM1	ADC 积分器中断模式定义，此两位用来定义 ADCMPO 数据中断触发模式： 00:不触发中断 01:上升沿触发中断 10:下降沿触发中断 11: 上升/下降沿触发中断
6	ADCCKEN	AD 转换器断路器时钟开关 0：关闭 1：打开（时钟频率由 ADCD 寄存器设置）
7	ADRR0	AD 转换器电阻选择 0：（V_{INT}，V_{CMP}）=（4/6VOREG，1/6VOREG） 1：（V_{INT}，V_{CMP}）=（4.4/6VOREG，1/6VOREG）

ADCR（18H）寄存器

位	符号	功能
0 1 2	ADCD0 ADCD1 ADCD2	定义断路器时钟（需 ADCCKEN 使能），建议频率为 10kHz 其断路器频率： 0：时钟频率=（f_{SYS}/32）/1 1：时钟频率=（f_{SYS}/32）/2 2：时钟频率=（f_{SYS}/32）/4 3：时钟频率=（f_{SYS}/32）/8 4：时钟频率=（f_{SYS}/32）/16 5：时钟频率=（f_{SYS}/32）/32 6：时钟频率=（f_{SYS}/32）/64 7：时钟频率=（f_{SYS}/32）/128
3~7	—	保留

ADCD（1AH）寄存器

位	符号	功能
0~1	—	保留位，读取值为 0
2	CHGTS	ADC 充电状态定时/计数器 0：定时/计数器 0 1：定时/计数器 1
3	DISTS	ADC 放电状态定时/计数器 0：定时/计数器 0 1：定时/计数器 1
4	CHGCMP	ASTEN 使能状态下，充电定时/计数器(*1)启动是否由输出位（ADCMPO）决定。 0:充电开始，定时/计数器立即开始计时(*3) 1:充电开始后等待 ADCMPO 输出上升沿后定时/计数器开始计时(*3)
5	ASTEN	双积分模式自动开始标志位。当这个功能被使能，当用户设置 ADDISCH 去启动充电模式时，充电时定时/计数器（*1）即会自动开始计时（*3）。这种启动方式也是由 CHGCMP 位来确定。 0:除能　　1:使能自动功能
6	ADISEN	双积分自动放电使能位。当这个功能被使能且 ADDISCH 设置为充电模式，一旦充电计时溢出(*1)，那么充电定时/计数器会自动停止计时(*3)，ADDISCH 将会自动设置为放电模式(*4)且自动放电定时/计数器(*2)会自动开始计时(*3) 0:除能　　1:使能自动功能
7	AENDEN	双积分自动模式结束控制位。当此功能使能且 ADDISCH 设置为放电模式，当检测到 ADCMPO 输出下降时，放电定时/计数器会自动停止计数(*2) 0:除能　　1:使能自动功能

*1：充电定时/计数器为在 CHGTS 位选择的 timer0 或 timer1

*2：放电定时/计数器为在 DISTS 位选择的 timer0 或 timer1

*3：计数器自动开始计数，将 T0ON 或 T1ON 位置 1，计数器自动停止计数，将设置 T0ON 或 T1ON 位为 0

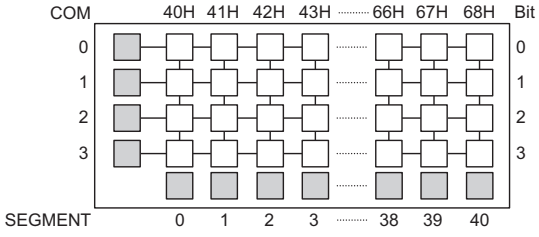
*4：将 ADDISCH 1/0 位设置成 0/1 则将自动进入放电模式

EADCR（1BH）寄存器

LCD 显示存储器

HT46RU75D-1 为 LCD 显示提供一个嵌入式数据存储器区域。这个区域位于第一段数据存储器(RAM Bank 1)的 40H 到 68H 单元。存储器段指针 Bank Pointer(BP；RAM 的 04H 单元)是通用存储器与 LCD 显示存储器之间切换的开关。当 BP 被置“1”，任何数据写入 40H~68H(用 MP1 和 R1 间接寻址访问)将会影响 LCD 的显示。当 BP 被清“0”， 任何数据写入 40H~68H 意味着访问一般意义上的数据存储器。LCD 显示存储器能被读出和写入，但是只能通过间接寻址模式，并使用 MP1 来进行。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，可以控制显示或不显示。下图为显示存储器和 LCD 显示模块之间的映射关系。

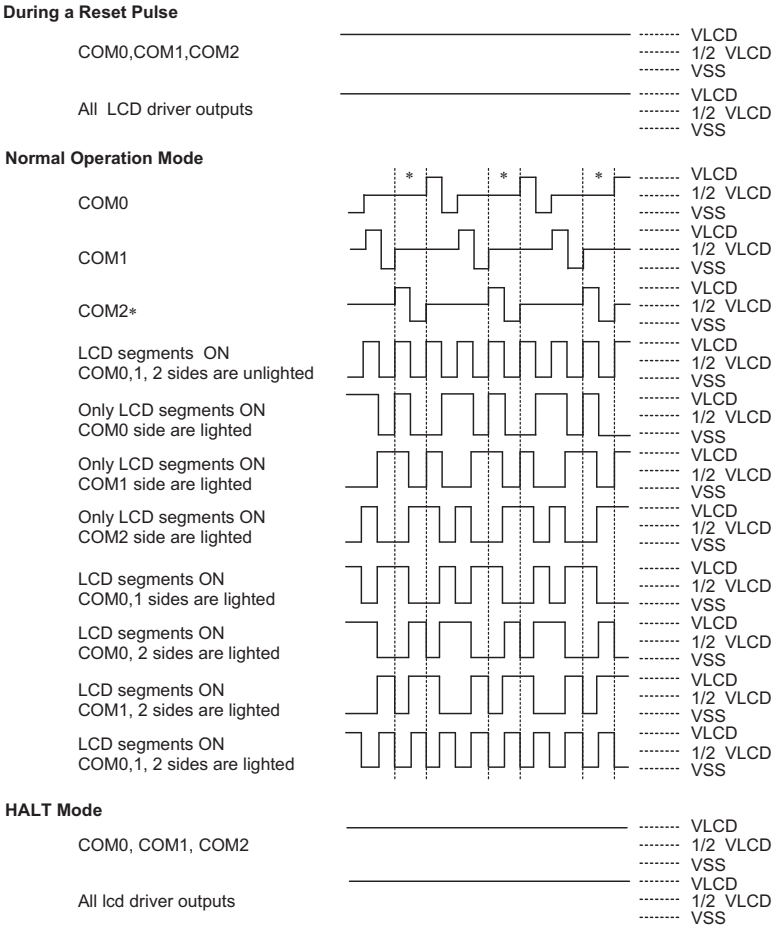
LCD 时钟频率可由掩膜选项选择，其可选择频率范围是 $fs/2^2$~$fs/2^8$。为了达到最好的显示效果请将 LCD 时钟频率尽可能地选择接近 4K。注意：LCD 时钟分频由掩膜选项控制，其频率为内部分频率得到。



显示存储器

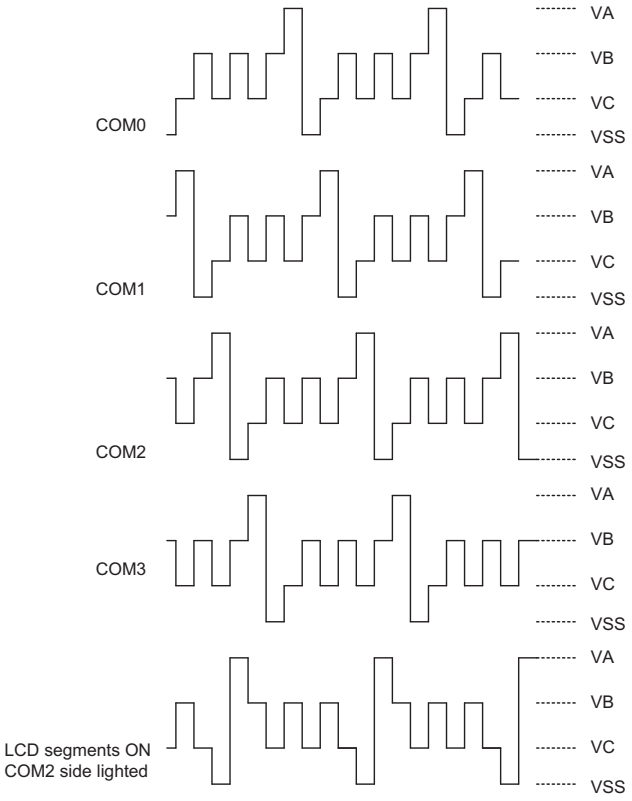
LCD 驱动输出

HT46RU75D-1 LCD 驱动器的输出数目可以由掩膜选项确定为 41×2、41×3、40×4 (即 1/2、1/3 或 1/4 占空比)。



Note: "*" Omit the COM2 signal, if the 1/2 duty LCD is used.

LCD 驱动输出（1/3Duty,1/2 Duty,R Type）



Note: 1/4 duty, 1/3 bias, R type: "VA" VLCD, "VB" 2/3 VLCD, "VC" 1/3 VLCD

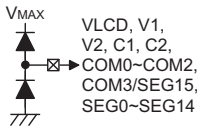
LCD 驱动输出（1/4Duty）

LCD 驱动器偏压产生方式有“R”型和“C”型。如果选为“R”型，不需要外接电容器；如果为“C”型，需要在 C1 和 C2 之间外接一个电容器。LCD 驱动器偏置电压值可以由掩膜选项设置为 1/2 bias 或 1/3 bias。如果选择为 1/2 bias，则引脚 V2 到地需要接一个电容；如果选择为 1/3 bias 的话，则需要两个电容到地分别地接到引脚 V1、V2。接法参考下面的表格：

LCD 偏压方式	R 型偏压		C 型偏压	
LCD 偏压类型	1/2bias	1/3bias	1/2bias	1/3bias
V_{MAX}	如果 $V_{DD}>V_{LCD}$，则 V_{MAX} 接到 V_{DD}，反之 V_{MAX} 接到 V_{LCD}		如果 $V_{DD}>\frac{3}{2}V_{LCD}$，则 V_{MAX} 接到 V_{DD}，反之 V_{MAX} 接到 V1	

LCD Segment 输出和逻辑输出

SEG0～SEG23 可由掩膜选项设置为逻辑输出。一旦 LCD segment 用作逻辑输出，LCD 存储区的相关 segment 地址的 bit0 将出现在这个 segment。SEG0～SEG7 和 SEG8~SEG15 作为逻辑输出时分别是按字节选择的，SEG16~SEG23 作为逻辑输出时是分别按位选择的。



低电压复位/检测功能

系统具有低电压检测(LVD)和低电压复位(LVR)功能，此两种功能可由掩膜选项设置为打开或关闭。如果选择 LVD 功能，用户可以通过 RTCC.3 来打开/关闭低电压检测，通过 RTCC.5 来读取低电压检测的状态。否则，低电压检测无效。

RTCC 寄存器的定义如下表：

位	标号	功能
0~2	RT0~RT2	通过控制 8 选 1 多路器输入来选择实时钟预置寄存器的分频输出比例
3	LVDC	低电压检测打开/关闭(1/0)
4	QOSC	32768Hz 晶振快速起振　0/1：快速/慢速
5	LVDO	低电压检测输出(1/0)，1：检测到低电压，只读
6~7	—	未定义，读出为 0

RTCC（09H）寄存器

LVR 与外部复位信号有相同的功能，都会使芯片复位。但要注意在 HALT 状态下，LVR 及 LVD 功能是没有作用的。

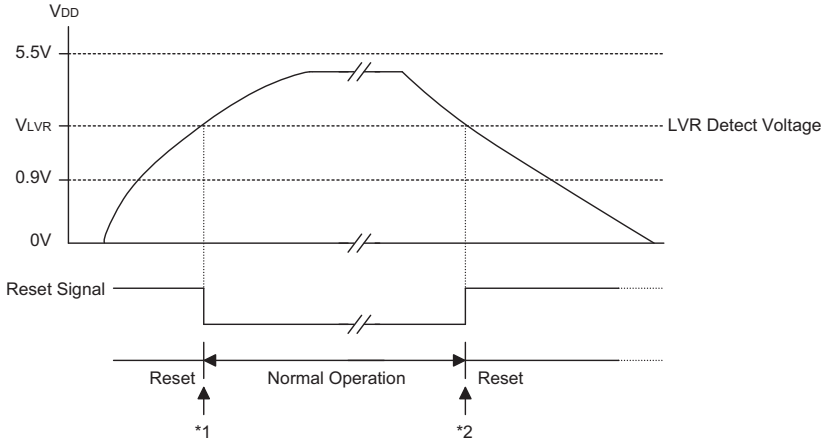
为了监控器件的工作电压，HT46RU75D-1 提供低电压复位功能。如果器件的工作电压在 0.9V～V_{LVR} 之间，例如电池电压的变化，那么 LVR 会自动使器件产生内部复位。

当 LVD 功能启动后，LVD 电压也由掩膜选项来决定。当选择值为 LVR＋0.2 时，实际的 LVD 电压由 LVR 值来决定，且监测的是 VDD 的电压；当选择 LVD 电压值为调整器输出＋0.2 时，实际的 LVD 电压为 VLVD3（请参看 DC 特性表），且监测的是调整器的输入电压。

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”的功能来执行系统复位。

V_{DD} 与 V_{LVR} 之间的关系如下所示：



低电压复位

注：*1：为了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，因此进入复位模式就要有 1ms 的延迟。

异步串行口 — UART

HT46RU75D-1 具有一个全双工的异步串行通信口,可以很方便的与其它具有串行口的芯片通讯。UART 具有许多功能特性，发送或接收串行数据时，将数据组成一个 8 位或 9 位的数据块，连同数据特征位一并传输。当数据过多或数据特征位不正确时，UART 可以检测出错误。UART 功能占用一个内部中断向量，当接收到数据或数据发送结束，触发 UART 中断。

- **UART 特性**

UART 具有以下功能:

- 全双工异步传输
- 可选择 8 位或 9 位字符长度
- 可选择奇校验、偶校验或无校验
- 可选择 1 位或 2 位停止位
- 8 位预分频的波特率发生器
- 奇偶、帧、噪声和过速检测
- 支持中断和地址检测（最后一位=1）
- 独立的发送和接收允许
- 两层 FIFO 接收缓冲器
- 发送和接收中断
- 下列条件可触发中断
  - — 发送完成
  - — 发送寄存器空闲
  - — 接收完成
  - — 过速错误
  - — 地址匹配

- **UART 外部引脚**

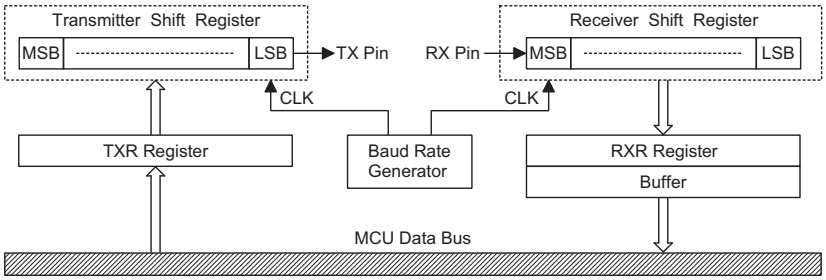
UART 是通过两个外部引脚 TX 和 RX 与外部芯片通讯。TX 引脚是 UART 的发送引脚。当 UCR2 寄存器的 TXEN 位清零，UART 发送功能被禁止，则 TX 引脚可作为普通 IO 口使用。RX 引脚是 UART 的接收引脚。同样的，当 UCR2 寄存器的 RXEN 位清零，UART 接收功能被禁止，则 RX 引脚可作为普通 IO 口使用。若 UARTEN、TXEN 和 RXEN 置位，这些 IO 口将自动转换成相应 TX 输出和 RX 输入，并且 RX 脚的上拉电阻将无效。

- **数据发送**

下图显示了 UART 的整体结构。需要发送的数据首先写入 TXR 寄存器，然后在波特率发生器的控制下将寄存器中数据一位位地移到 TX 引脚上，低位在前。TXR 寄存器被映像到单片机的数据存储器中，而发送移位寄存器没有实际地址，所以发送移位寄存器不可直接操作。

数据在波特率发生器的控制下，低位在前高位在后，从外部引脚 RX 进入接收移位寄存器。当数据接收完成，数据从接收移位寄存器移入可被用户程序操作的 RXR 寄存器中。RXR 寄存器被映像到单片机数据存储器中，而接收移位寄存器没有实际地址，所以接收移位寄存器不可直接操作。

需要注意的是，上述发送寄存器 TXR 和接收寄存器 RXR，其实是共享一个地址的数据寄存器 TXR/RXR 寄存器。



UART 数据发送接收图

- **UART 状态控制寄存器**

有 5 个寄存器与 UART 功能相关。寄存器 USR、UCR1 和 UCR2 全面控制 UART，而寄存器 BRG 控制波特率，发送和接收数据则通过寄存器 TXR/RXR。

- **USR 寄存器**

寄存器 USR 是 UART 的状态寄存器，可以通过程序读取。所有 USR 位是只读的。

详细解释如下：

- TXIF

TXIF 是发送数据寄存器为空标志。若 TXIF=0，数据还没有从缓冲器加载到移位寄存器中；若 TXIF=1，数据已从缓冲器中加载到移位寄存器中。读取 USR 寄存器再写 TXR 寄存器将清除 TXIF。当 TXEN 被置位，由于发送缓冲器未满，TXIF 也会被置位。

- TIDLE

TIDLE 是数据发送完成标志位。若 TIDLE=0，数据传输中。当 TXIF=1 且数据发送完毕或者暂停字被发送时，TIDLE 置位。TIDLE=1，TX 引脚空闲。读取 USR 寄存器再写 TXR 寄存器将清除 TIDLE 位。

- RXIF

RXIF 是接收寄存器状态标志。当 RXIF=0，RXR 寄存器为空；当 RXIF=1，RXR 寄存器接收到新数据。当数据从移位寄存器中加载到 RXR 寄存器中，如果 UCR2 寄存器中的 RIE=1，则会触发中断。当接收数据时检测到一个或多个错误时，相应的标志位 NF、FERR 或 PERR 会在同一周期内置位。读取 USR 寄存器再读 RXR 寄存器，如果 RXR 寄存器中没有新的数据，那么将清除 RXIF 标志。

- RIDLE

RIDLE 是接收状态标志。若 RIDLE=0，正在接收数据；若 RIDLE=1，接收器空闲。在接收到停止位和下一个数据的起始位之间，RIDLE 被置位，表明 UART 空闲。

- OERR

OERR 是过速错误标志，表示接收缓冲器是否溢出。若 OERR=0，没有数据溢出；若 OERR=1，发生了过速错误，它将影响下一组数据的接收。先读取 USR 寄存器再读 RXR 寄存器将清除此标志位。

- FERR

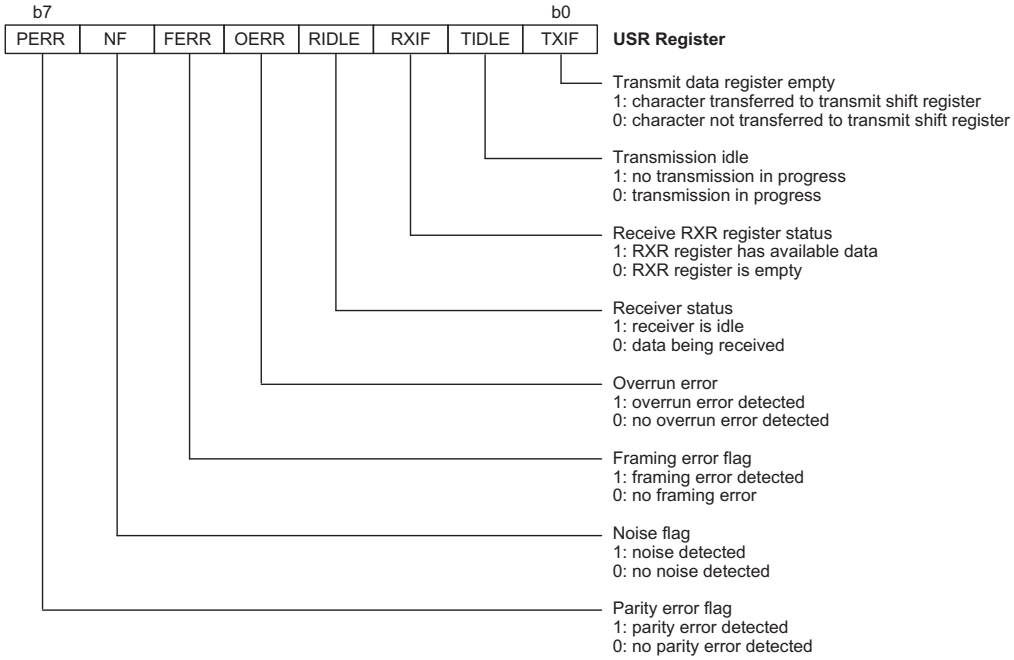
FREE 是帧错误标志位。若 FREE=0，没有帧错误发生；若 FREE=1，当前的数据发生了帧错误。任何复位都会清除该标志位，也可以先读取 USR 寄存器再读 RXR 寄存器来清除此位。

- NF

NF 是噪声干扰标志。若 NF=0，没有受到噪声干扰；若 NF=1，UART 接收数据时受到噪声干扰。它与 RXIF 在同周期内置位，但不会与过速标志位同时置位。先读取 USR 寄存器再读 RXR 寄存器将清除此标志位。

- PERR

PERR 是奇偶校验出错标志。若 PERR=0，奇偶校验正确；若 PERR=1，接收到的数据奇偶校验出错。只有使能了奇偶校验此位才有效。任何复位都会清除该标志位，也可以先读取 USR 寄存器再读 RXR 寄存器来清除此位。



- **UCR1 寄存器**

UCR1 和 UCR2 是 UART 的两个控制寄存器，用来定义各种 UART 功能，例如 UART 的使能与除能、奇偶校验控制和传输数据的长度等等。

详细解释如下：

- TX8

此位只有在传输数据为 9 位的格式中有效，用来存储发送数据的第 9 位。BNO 是用来控制传输位数是 8 位还是 9 位。

- RX8

此位只有在传输数据为 9 位的格式中有效，用来存储接收数据的第 9 位。BNO 是用来控制传输位数是 8 位还是 9 位。

- TXBRK

TXBRK 是暂停字发送控制位。TXBRK=0，没有暂停字要发送，TX 引脚正常操作；TXBRK=1，将会发送暂停字，发送器将发送逻辑 0。若 TXBRK 为高，缓冲器中数据发送完毕后，发送器将至少保持 13 位宽的低电平直至 TXBRK 复位。

- STOP

此位用来设置停止位的长度。STOP=1，有两位停止位；STOP=0，只有一位停止位。

- PRT

奇偶校验选择位。PRT=1，奇校验；PRT=0，偶校验。

- PREN

此位为奇偶校验使能位。PREN=1，使能奇偶校验；PREN=0，除能奇偶校验。

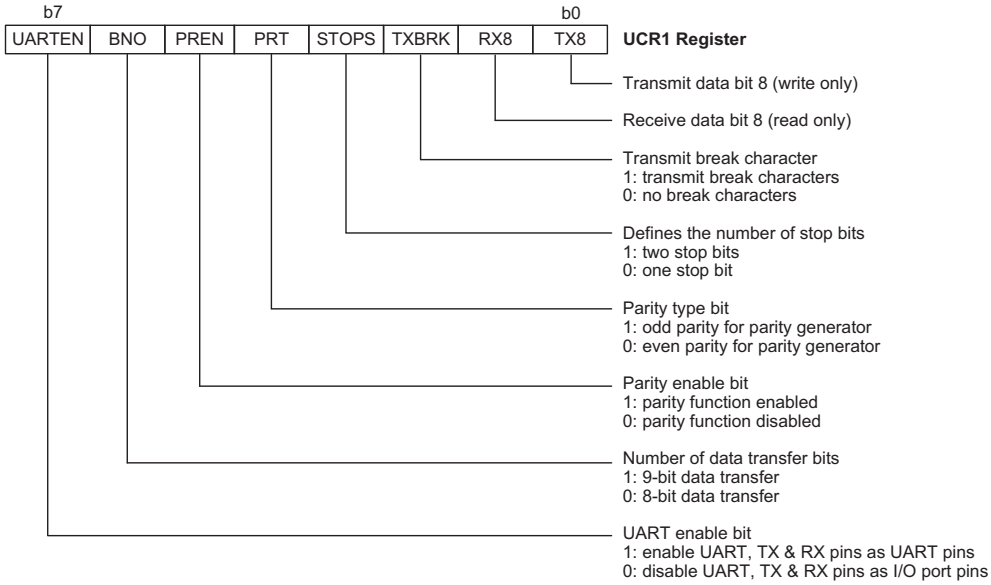
- BNO

BNO 是发送位数控制位。BNO=1，传输数据为 9 位；BNO=0，传输数据为 8 位。若选择了 9 位数据传输格式，RX8 和 TX8 将分别存储接收和发送数据的第 9 位。

- UARTEN

此位为 UART 的使能位。UARTEN=0，UART 除能，RX 和 TX 可用作普通的输入输出口；UARTEN=1，UART 使能，TX 和 RX 将分别由 TXEN 和 RXEN 控制。当 UART 被除能将清除缓冲器，所有缓冲器中的数据将被忽略，另外波特率计数器、错误和状态标志位被复位，TXEN、RXEN、TXBRK、RXIF、OERR、FERR 和 NF 清零而 TIDLE、TXIF 和 RIDLE 置位，UCR1、UCR2 和 BRG 寄存器中的其它位保持不变。若 UART 工作时 UARTEN 清零，所有发送和接收将停止，模块也将复位成上述状态。当 UART 再次使能时，它将

在上次配置下重新工作。



- **UCR2 寄存器**

UCR2 是 UART 的另一个控制寄存器，它的主要功能是使能或除能发送和接收允许以及 UART 的各种中断源。它也可用来控制波特率，使能接收唤醒和地址侦测。

详细解释如下：

- TEIE

此位为发送寄存器为空中断的使能或除能位。若 TEIE=1，当 TXIF 置位时，UART 的中断请求标志置位；若 TEIE=0，UART 中断请求标志不受 TXIF 的影响。

- TIIE

此位为发送器空闲中断的使能或除能位。若 TIIE=1，当 TIDLE 置位时，UART 的中断请求标志置位；若 TIIE=0，UART 中断请求标志不受 TIDLE 的影响。

- RIE

此位为接收中断使能或除能位。若 RIE=1，当 OERR 或 RXIF 置位时，UART 的中断请求标志置位；若 RIE=0，UART 中断请求标志不受 OERR 和 RXIF 影响。

- WAKE

此位为接收唤醒功能的使能和除能位。若 WAKE=1 且在暂停模式下，RX 引脚的下降沿将唤醒单片机。若 WAKE=0 且在暂停模式下，RX 引脚的任何边沿都不能唤醒单片机。

- ADDEN

此位为地址检测使能和除能位。ADDEN=1，地址检测使能，此时数据的第 8 位（BON=0）或第 9 位（BON=1）为高，那么接到的是地址而非数据。若相应的中断使能且接收到的值最高位为 1，那么中断请求标志将会被置位，若最高位为 0，那么将不会产生中断且收到的数据也会被忽略。

- BRGH

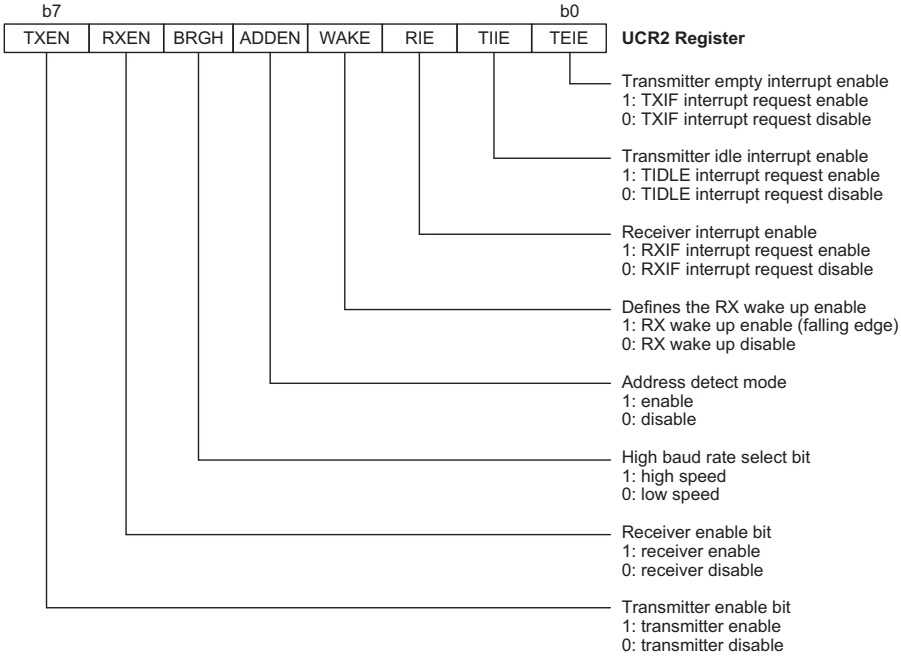
此位为波特率发生器高低速选择位，它和 BRG 寄存器一起控制 UART 的波特率。BRGH=1，为高速模式；BRGH=0，为低速模式。

- RXEN

此位为接收使能位。RXEN=0，接收将被除能，接收器立刻停止工作。另外缓冲器将被复位，此时 RX 引脚可作普通的输入输出端口使用。若 RXEN=1 且 UARTEN=1，则接收将被使能，RX 引脚将由 UART 来控制。在数据传输时清除 RXEN 将中止数据接收且复位接收器，此时 RX 引脚可作为普通输入输出端口使用。

- TXEN

此位为发送使能位。TXEN=0，发送将被除能，发送器立刻停止工作。另外缓冲器将被复位，此时 TX 引脚可作为普通的输入输出端口使用。若 TXEN=1 且 UARTEN=1，则发送将被使能，TX 引脚将由 UART 来控制。在数据传输时清除 TXEN 将中止数据发送且复位发送器，此时 TX 引脚可作为普通的输入输出口使用。



- **波特率发生器**

UART 自身具有一个波特率发生器，通过它可以设定数据传输速率。波特率是由一个独立的内部 8 位计数器产生，它由 BRG 寄存器和 UCR2 寄存器的第 2 位 BRGH 来控制。BRGH 是决定波特率发生器处于高速模式还是低速模式，从而决定计算公式的选用。BRG 寄存器的值 N 可根据下表中的公式计算，N 的范围是 0 到 255。

UCR2 的 BRGH 位	0	1
波特率	$\frac{f_{SYS}}{64(N+1)}$	$\frac{f_{SYS}}{16(N+1)}$

为得到相应的波特率，首先需要设置 BRGH 来选择相应的计算公式从而算出 BRG 的值。由于 BRG 的值不连续，所以实际波特率和理论值之间有一个偏差。下面举例怎样计算 BRG 寄存器中的值 N 和误差。

- 波特率和误差的计算

系统选用 8M 晶振且 BRGH=0，若期望的波特率为 9600，计算它的 BRG 寄存器的值 N，实际波特率和误差。

根据上表，波特率 BR = $\frac{f_{SYS}}{64(N+1)}$

转换后的公式 N = $\frac{f_{SYS}}{BR*64}-1$

代入参数 N = $\frac{8000000}{9600*64}-1$=12.0208

取最接近的值，十进制 12 写入 BRG 寄存器，实际波特率如下

BR = $\frac{8000000}{64(12+1)}=9615$

误差 $\frac{9615-9600}{9600}=0.16\%$

下面两表给出 BRGH 取不同值时的实际波特率和误差。

波特率 K/BPS	BRGH=0											
	f_{SYS}=8MHz			f_{SYS} =7.159MHz			f_{SYS} =4MHz			f_{SYS} =3.579545MHz		
	BRG	Kbaud	Error	BRG	Kbaud	Error	BRG	Kbaud	Error	BRG	Kbaud	Error
0.3	-	-	-	-	-	-	207	0.300	0.00	185	0.300	0.00
1.2	103	1.202	0.16	92	1.203	0.23	51	1.202	0.16	46	1.19	-0.83
2.4	51	2.404	0.16	46	2.38	-0.83	25	2.404	0.16	22	2.432	1.32
4.8	25	4.807	0.16	22	4.863	1.32	12	4.808	0.16	11	4.661	-2.9
9.6	12	9.615	0.16	11	9.322	-2.9	6	8.929	-6.99	5	9.321	-2.9
19.2	6	17.857	-6.99	5	18.64	-2.9	2	20.83	8.51	2	18.643	-2.9
38.4	2	41.667	8.51	2	37.29	-2.9	1	-	-	1	-	-
57.6	1	62.5	8.51	1	55.93	-2.9	0	62.5	8.51	0	55.93	-2.9
115.2	0	125	8.51	0	111.86	-2.9	-	-	-	-	-	-

BRGH=0 时的波特率和误差

波特率 K/BPS	BRGH=1											
	f_{SYS}=8MHz			f_{SYS} =7.159MHz			f_{SYS} =4MHz			f_{SYS} =3.579545MHz		
	BRG	Kbaud	Error	BRG	Kbaud	Error	BRG	Kbaud	Error	BRG	Kbaud	Error
0.3	-	-	-	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	207	1.202	0.16	185	1.203	0.23
2.4	207	2.404	0.16	185	2.405	0.23	103	2.404	0.16	92	2.406	0.23
4.8	103	4.808	0.16	92	4.811	0.23	51	4.808	0.16	46	4.76	-0.83
9.6	51	9.615	0.16	46	9.520	-0.832	25	9.615	0.16	22	9.727	1.32
19.2	25	19.231	0.16	22	19.454	1.32	12	19.231	0.16	11	18.643	-2.9
38.4	12	38.462	0.16	11	37.287	-2.9	6	35.714	-6.99	5	37.286	-2.9
57.6	8	55.556	-3.55	7	55.93	-2.9	3	62.5	8.51	3	55.930	-2.9
115.2	3	125	8.51	3	111.86	-2.9	1	125	8.51	1	111.86	-2.9
250	1	250	0	-	-	-	0	250	0	-	-	-

BRGH=1 时的波特率和误差

- **UART 设置与控制**

  - 简介

    UART 采用标准的不归零码传输数据，这种方法通常被称为 NRZ 法。它由 1 位起初位，8 位或 9 位数据位和 1 位或者两位停止位组成。奇偶校验是由硬件自动完成的，可设置成奇校验、偶校验和无校验三种格式。常用的数据传输格式由 8 位数据位，1 位停止位，无校验组成，用 8，N，1 表示，它是系统上电的默认格式。数据位数、停止位数和奇偶校验由 UCR1 寄存器的 BNO、PRT、PREN 和 STOPS 设定。用于数据发送和接收的波特率由一个内部的 8 位定时器产生，数据传输时低位在前高位在后。尽管 UART 发送器和接收器在功能上相互独立，但它们使用相同的数据传输格式和波特率，在任何情况下，停止位是必须的。

  - UART 的使能和除能

    UART 是由 UCR1 寄存器的 UARTEN 位来使能和除能的。它的发送引脚 TX 和接收引脚 RX 分别与输入/输出引脚复用，UARTEN 的一个基本功能就是控制这两个引脚。若 UARTEN、TXEN 和 RXEN 都为高，则 TX 和 RX 分别为 UART 的发送端口和接收端口，而不能作为普通的输入输出端口使用。若没有数据发送，TX 引脚默认状态为高电平。

    UARTEN 清零将除能 TX 和 RX，使其可作为普通的输入输出端口使用。当 UART 被除能将清除缓冲器，所有缓冲器中的数据将被忽略，另外错误和状态标志位被复位，TXEN、RXEN、TXBRK、RXIF、OERR、FERR 和 NF 清零而 TIDLE、TXIF 和 RIDLE 置位，UCR1、UCR2 和 BRG 寄存器中的其它位保持不变。若 UART 工作时 UARTEN 清零，所有发送和接收将停止，模块也将复位成上述状态。当 UART 再次使能时，它将在上次配置下重新工作。

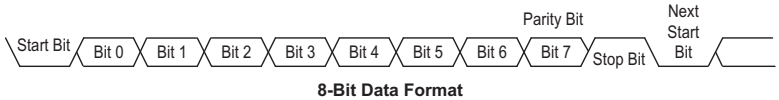
  - 数据位、停止位位数以及奇偶校验的选择

    数据传输格式由数据长度、是否校验、校验类型地址表明位以及停止位长度组成。它们都是由 UCR1 寄存器的各个位控制的。BNO 决定数传输是 8 位还是 9 位；PRT 决定校验类型；PRTEN 决定时是否选择奇偶校验；而 STOPS 决定选用 1 位还是 2 位停止位。下表列出了各种数据传输格式。地址表明位用来确定此帧是否为地址。

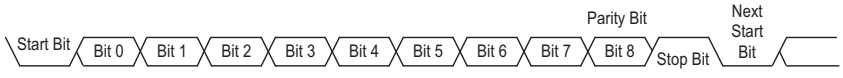
Start Bit	Data Bits	Address Bits	Parity Bits	Stop Bits
8 位数据位				
1	8	0	0	1
1	7	0	1	1
1	7	1[1]	0	1
9 位数据位				
1	9	0	0	1
1	8	0	1	1
1	8	1[1]	0	1

发送和接收数据格式

下图是传输 8 位和 9 位数据的波形。



8-Bit Data Format



9-Bit Data Format

- UART 发送器

UCR1 寄存器的 BNO 位是控制数据传输的长度。BNO=1 其长度为 9 位，第 9 位 MSB 存储在 UCR1 寄存器的 TX8 中。发送器的核心是发送移位寄存器 TSR，它的数据由发送寄存器 TXR 提供，应用程序只须将发送数据写入 TXR 寄存器。上组数据的停止位发出前，TSR 寄存器禁止写入。如果还有新的数据要发送，一旦停止位发出，待发数据将会从 TXR 寄存器加载到 TSR 寄存器。TSR 不像其它寄存器一样映射在数据存储器，所以应用程序不能对其进行读写操作。TXEN=1，发送使能，但若 TXR 寄存器没有数据或者波特率没有设置，发送器将不会工作。先写 TXR 寄存器再 TXEN 也会触发发送。当发送器使能，若 TSR 寄存器为空，数据写入 TXR 寄存器将会直接加载到 TSR 寄存器中。发送器工作时，TXEN 清零，发送器将立刻停止工作并且复位，此时 TX 引脚可作为普通的输入输出口使用。

- 发送数据

当 UART 发送数据时，数据从移位寄存器中移到 TX 引脚上，其低位在前高位在后。在发送模式中，TXR 寄存器在内部总线和发送移位寄存器间形成一个缓冲。如果选择 9 位数据传输格式，最高位 MSB 存储在 UCR1 寄存器的 TX8 中。

发送器初始化可由如下步骤完成：

--正确地设置 BNO、PRT、PREN 和 STOPS 位以确定数据长度、校验类型和停止位长度。

--设置 BRG 寄存器，选择期的波特率。

--置高 TXEN，使引脚作为 UART 的发送端而非普通的输入输出端口。

--读取 USR 寄存器，然后将待发数据写入 TXR 寄存器，此步骤会清除 TXIF 标志位。

--如果要发送多个数据只需重复上一步骤。

当 TXIF=0 时，数据将禁止写入 TXR 寄存器。可以通过以下步骤来清除 TXIF：

1. 读取 USR 寄存器
2. 写 TXR 寄存器

只读标志位 TXIF 由 UART 硬件置位。若 TXIF=1，TXR 寄存器为空，其它数据可以写入而不会覆盖以前的数据。若 TEIE=1，TXIF 标志位会影响中断。

在数据传输时，写 TXR 指令会将待发数据暂存在 TXR 寄存器中，当前数据发送完毕后，待发数据被加载到发送移位寄存器中。当发送器空闲时，写 TXR 指令会将数据直接加载到 TSR 寄存器中，数据传输立刻开始且 TXIF 置位。当一帧数据发送完毕，TIDLE 将被置位。

可以通过以下步骤来清除 TIDLE：

1. 读取 USR 寄存器
2. 写 TXR 寄存器

清除 TXIF 和 TIDLE 软件执行次序相同。

- 发送暂停字

若 TXBRK=1，下一帧将会发送暂停字。它是同一个起始位、13*N（N=1，2……）位逻辑 0 以及停止位组成。置位 TXBRK 将会发送暂停字，而清除 TXBRK 产生停止位，传输暂停字不会产生中断。需要注意思的是，暂停字至少 13 位宽。若 TXBRK 持续为高，那么发送器会一直发送暂停字；当应用程序清除了 TXBRK，发送器将传输最后一帧暂停字再加上一位或者两位停止位。暂停字后的高电平保证下一帧数据起始位的检测。

- **UART 接收器**

  - 简介

    UART 接收器支持 8 位或者 9 位数据接收。若 BNO=1，数据长度为 9 位，而最高位 MSB 存放在 UCR1 寄存器的 RX8 中。接收器的核心是串行移位寄存器 RSR。RX 引脚上的数据送入数据恢复器中，它在 16 倍波特率的频率下工作，而串行移位器工作在正常波特率下。当在 RX 引脚上检测到停止位，数据从 RSR 寄存器中加载到 RXR 寄存器。RX 引脚上的每一位数据会被采样三次以判断其逻辑状态。RSR 不像其它寄存器一样映射在数据存储器，所以应用程序不能对其进行读写操作。

  - 接收数据

    当 UART 接收数据时，数据低位在前高位在后，连续地从 RX 引脚进入。RXR 寄存器在内部总线和接收移位寄存器间形成一个缓冲。RXR 寄存器是一个两层的 FIFO 缓冲器，它能保存两帧数据的同时接收第三帧数据，应用程序必须保证在接收完第三帧前读取 RXR 寄存器，否则忽略第三帧数据并且发生过速错误。

    接收器的初始化可由如下步骤完成：

    --正确地设置 BNO、PRT、PREN 和 STOPS 位以确定数据长度、校验类型和停止位长度。

    --设置 BRG 寄存器，选择期的波特率。

    --置高 TXEN，使引脚作为 UART 的发送端而非普通的输入输出端口。

    此时接收器被使能并检测起始位。

    接收数据将会发生如下事件：

    --当 RXR 寄存器中有一帧以上的数据时，USR 寄存器中的 RXIF 位将会置位。

    --若 RIE=1，数据从 RSR 寄存器加载到 RXR 寄存器中将产生中断。

    --若接收器检没到帧错误、噪声干扰错误、奇偶出错或过速错误，那么相应的错误标志位置位。

    可以通过如下步骤来清除 RXIF：

    1. 读取 USR 寄存器
    2. 读取 RXR 寄存器

  - 接收暂停字

    UART 接收任何暂停字都会当作帧错误处理。接收器只根据 BNO 和 STOPS 位确定一帧数据的长度。若暂停字位数大于 BNO 和 STOPS 位指定的长度（13 位），接收器认为接收已完毕，RXIF 和 FERR 置位，RXR 寄存器清 0，若相应的中断允许且 RIDLE 为高将会产生中断。若暂停字较长，接收器收到起始位、数据位将会置位 FERR 标志，且在下一起始位前必须检测到有效的停止位。暂停字只会被认为包含信息 0 且会置位 FERR 标志。暂停字将会加载到缓冲器中，在接收到停止位前不会再接收数据，没有检测到停止位也会置位只读标志位 RIDLE。

    UART 接收到暂停字会产生以下事件：

    --帧错误标志位 FERR 置位。

    --RXR 寄存器清零。

    ---OERR、NF、PERR、RIDLE 或 RXIF 可能会置位。

  - 空闲状态

    当 UART 接收数据时，即在起初位和停止位之间，USR 寄存器的接收标志位 RIDLE 清零。在停止位和下一帧数据的起始位之间，RIDLE 被置位，表示接收器空闲。

  - 接收中断

    USR 寄存器的只读标志位 RXIF 由接收器的边缘触发置位。若 RIE=1，数据从移位寄存器 RSR 加载到 RXR 寄存器时产生中断，同样地，过速也会产生中断。

- **接收错误处理**

UART 会产生几种接收错误，下面部分将描述各错误以及怎样处理。

- 过速——OERR 标志

RXR 寄存器是一个两层的 FIFO 缓冲器，它能保存两帧数据的同时接收第三帧数据，应用程序必须保证在接收完第三帧前读取 RXR 寄存器，否则发生过速错误。

产生过速错误时将会发生以下事件：

一 USR 寄存器中 OERR 被置位。

一 RXR 寄存器中数据不会丢失。

一 RSR 寄存器数据将会被覆盖。

一若 RIE=1，将会产生中断。

读取 RXR 寄存器后，OERR 标志将在访问 USR 寄存器时被清零。

- 噪声干扰——NF 标志

数据恢复时多次采样可以有效的鉴别出噪声干扰。当检测到数据受到噪声干扰时将会发生以下事件：

一 在 RXIF 上升沿，USR 寄存器中只读标志位 NF 置位。

一 数据从 RSR 寄存器加载到 RXR 寄存器中。

一 不产生中断，此位置位的同时由 RXIF 请求中断。

先读取 USR 寄存器再读取 RXR 寄存器将复位 NF。

- 帧错误——FERR 标志

若在停止位上检测到 0，USR 寄存器中只读标志 FERR 置位。若选择两位停止位，两停止都必须为高，否则将置位 FERR。它同数据一起存储在缓冲器中，可被任何复位清除。

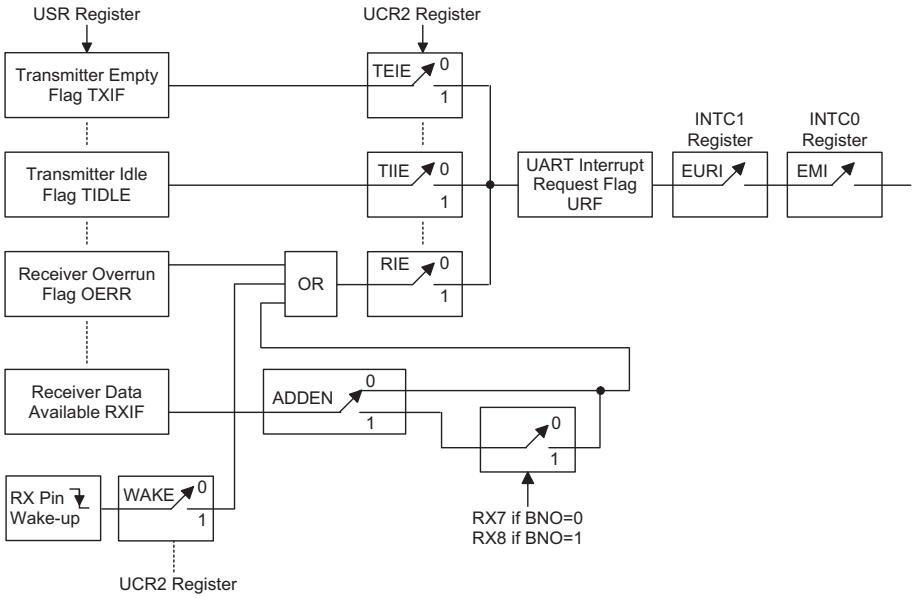
- 奇偶校验错误——PERR 标志

若接收到数据出现奇偶校验错误，USR 寄存器中只读标志 PERR 置位。只有使能了奇偶校验，选择了校验类型，此标志位才有效。它同数据一起存储在缓冲器中，可被任何复位清除。注意，FERR 和 PERR 与相应的数据一起存储有缓冲器中，在读取数据之前必须先访问错误标志位。

- **接收中断图解**

UART 拥有单独的内部中断和独立的中断变量。发送寄存器为空、发送器空闲、接收器数据有效、过速和地址检测和 RX 引脚唤醒都会产生中断。若 UART 中断允许且堆栈未满，程序将会跳转到相应的中断向量执行中断服务程序，而后再返回主程序。其中四种，若 UCR2 寄存器中相应中断允许位被置位，USR 寄存器中标志位将会产生中断。发送器有两个相应的中断允许位而接收器共用一个中断允许位。这些允许位可用于禁止个别的 UART 中断源。

地址检测也是 UART 的中断源，它没有相应的标志位，若 UCR2 寄存器中 ADDEN=1，当检测到地址将会产生 UART 中断。RX 引脚唤醒也可以产生 UART 中断，它没有相应的标志位，当 UXR2 中的 WAKE 和 RIE 位被置位，RX 引脚上有下降沿可以唤醒单片机。应注意，RX 唤醒中断发生时，系统必须延时 1024 个系统时钟才能正常工作。



UART 中断框图

- **地址检测模式**

置位 UCR2 寄存器中的 ADDEN 将启动地址检测模式。若 ADDEN 有效，只有在接收到数据最高位为 1 才会产生中断，中断允许位 EURI 和 EMI 也要使能才会产生中断。地址的最高位为第 9 位（BNO=1）或第 8 位（BNO=0），若此位为高，则接收到的是地址而非数据。只有接收的数据的最后位为高才会产生中断。若 ADDEN 除能，每接收到一个有效数据便会置位 RXIF，而不用考虑数据的最后一位。地址检测和奇偶校验在功能上相互排斥，若地址检测模式使能，必须保证操作的正确，同时必须将奇偶检验使能位清零，除能奇偶校验。

ADDEN	位 9（BNO=1），位 8（BNO=0）	产生 UART 中断
0	0	√
	1	√
1	0	×
	1	√

ADDEN 位功能

- **暂停模式下的 UART 功能**

当 MCU 进入暂停模式，UART 将停止工作。当芯片进入暂停模式，模块的所有时钟关闭。当 UART 传送数据时，MCU 进入暂停模式，发送将停止并且 TX 引脚保持高电平。同样地，当 MCU 接收数据时进入暂停模式，数据接收也会停止。当单片机进入暂停模式，USR、UCR1 和 UCR2、接收/发送寄存器、BRG 寄存器都不会受到影响。

UART 功能中包括了 RX 引脚的唤醒功能，由 UCR2 寄存器中 WAKE 位控制。进入暂停模式前，若该标志位与 UART 允许位 UARTEN、接收器允许位 RXEN 和接收器中断位 RIE 都被置位，则 RX 引脚的下降沿可唤醒单片机。唤醒后系统需延时 1024 个系统时钟才能正常工作，在此期间，RX 引脚上的任何数据将被忽略。

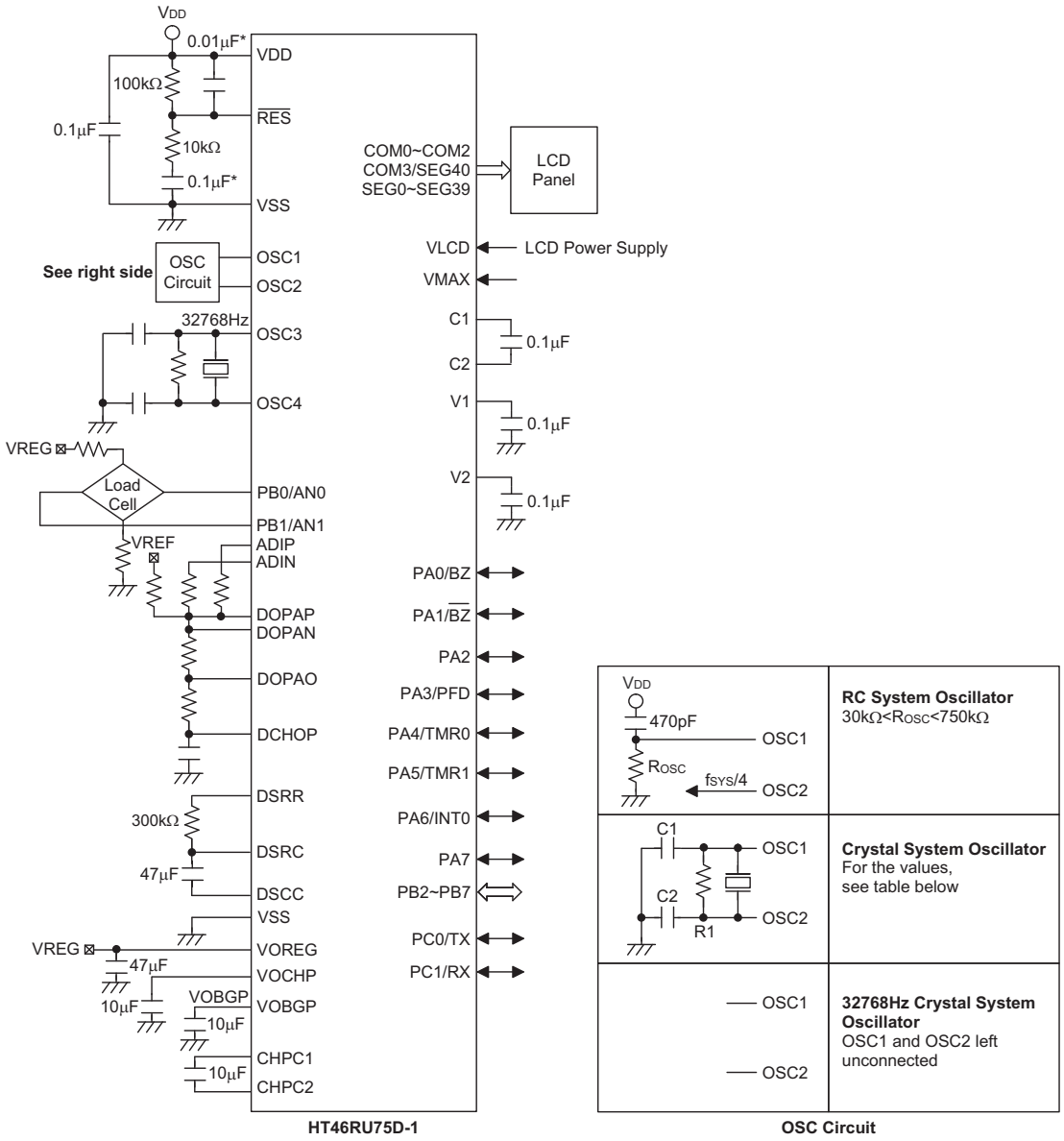
若要唤醒并产生 UART 中断， 全局中断允许位 EMI 和 UART 中断允许 EURI 也必须置位；若这两标志位没有被置位，那么，单片机将可以被唤醒但不会产生中断。同样唤醒后系统需延时 1024 个系统时钟才能正常工作，然后才会产生 UART 中断。

掩膜选项

下表列出了所有掩膜选项。所有选项必须正确定义，以保证统正常运行。

掩膜选项
OSC 选项 有两种选择方式： RC OSC 或晶体 OSC
系统时钟选择：OSC 或 RTC
f_S 时钟源选项 有三种选择方式：WDT、RTC 或 f_{SYS} /4。
WDT 时钟源选项。 有三种选择方式：WDT、RTC 或 f_{SYS} /4。
WDT 打开/关闭选项。 由掩膜选项，WDT 打开或关闭。
WDT 溢出周期选项。 有四种选择的方式：WDT 时钟源的 $2^{13}/f_S$、$2^{14}/f_S$、$2^{15}/f_S$ 或 $2^{16}/f_S$ 分频。
CLR WDT 次数选项。 这个选项定义用指令清除 WDT 的方法。“One time”指用“CLR WDT”指令功能清除 WDT。“Two times”指的是必须要用 CLR WDT1 和 CLR WDT2 二条指令来清除 WDT。
蜂鸣器输出频率选项。 有八种输出频率供选择：$f_S/2^2$ ~ $f_S/2^9$。“f_S”是由掩膜选项确定的时钟源频率。
Wake-up 选项。 这个选项用来设置唤醒功能。外部的输入/输出引脚(仅 PA 具有)的下降沿，具有将系统从 HALT 模式唤醒的能力。
上拉电阻选项。 这个选项用来设置输入/输出口在输入模式时，是否带有内部上拉电阻。按位设置。
输入/输出与其它功能共用引脚选项。 PA0/$\overline{BZ}$、PA1/BZ：PA0 和 PA1 可以设置为一般输入/输出口或蜂鸣器输出。PA3 可以设置为一般输入/输出口或 PFD 输出
PFD 时钟源选择：定时/计数器 0 或定时/计数器 1
LCD COM 选择。 有三种方式可选择：2COM（1/2 duty）、3COM（1/3 duty）或 4COM（1/4 duty）。
LCD 偏压选项。 有二种选择：1/2 bias 或 1/3 bias。
LCD Segment 做为逻辑输出选项。这个选项决定 SEG16~SEG23 是为一般逻辑输出口或 segment 输出（按位选择），及 SEG0~SEG7 和 SEG8~SEG15 是为逻辑输出口或 segment 输出（按字节选择）。
LCD 驱动器时钟源选项。 LCD 驱动器的时钟范围：$f_S/2^2$ ~ $f_S/2^8$。
LCD 在暂停模式下开/关选项。
LCD 偏压类型选项。 这个选项决定偏压类型是电阻型或电容型。
LVR 选项。LVR 打开或关闭。
LVD 选项。LVD 打开或关闭。
LVR 电压选择：2.1，3.15V 或 4.2V
LVD 电压选择：LVR＋0.2 或 regulator+0.2
$\overline{INT}$ 触发方式选项：禁止，上升沿触发，下降沿触发，或两者皆可触发
局部锁选项：0~3 页，4~7 页，8~11 页，….24~27 页，28~30 页，31 页。

应用电路


VDD
0.01μF*
VDD
100kΩ
RES
0.1μF
10kΩ
0.1μF*
VSS
COM0~COM2
COM3/SEG40
SEG0~SEG39
LCD Panel
VLCD
LCD Power Supply
See right side
OSC Circuit
OSC1
OSC2
VMAX
32768Hz
OSC3
C1
0.1μF
C2
OSC4
V1
0.1μF
VREG
V2
Load Cell
PB0/AN0
0.1μF
PB1/AN1
VREF
ADIP
ADIN
PA0/BZ
PA1/BZ
DOPAP
DOPAN
PA2
DOPAO
PA3/PFD
DCHOP
PA4/TMR0
PA5/TMR1
DSRR
300kΩ
PA6/INT0
DSRC
PA7
47μF
DSCC
PB2~PB7
VSS
PC0/TX
VREG
VOREG
PC1/RX
47μF
VOCHP
10μF
VOBGP
VOBGP
10μF
CHPC1
10μF
CHPC2
HT46RU75D-1
VDD
470pF
OSC1
ROSC
fSYS/4
OSC2
RC System Oscillator
30kΩ<ROSC<750kΩ
C1
OSC1
C2
OSC2
R1
Crystal System Oscillator
For the values,
see table below
OSC1
32768Hz Crystal System Oscillator
OSC1 and OSC2 left unconnected
OSC2
OSC Circuit

下表是不同晶体频率时，C1、C2 和 R1 的不同取值。以下数据仅供参考。

晶体或共振器	C1、C2	R1
4MHz 晶体	0pF	10kΩ
4MHz 共振器	10pF	12kΩ
3.58MHz 晶体	0pF	10kΩ
3.58MHz 共振器	25pF	10kΩ
2MHz 晶体和共振器	25pF	10kΩ
1MHz 晶体	35pF	27kΩ
480kHz 共振器	300pF	9.1kΩ
455kHz 共振器	300pF	10kΩ
429kHz 共振器	300pF	10kΩ
R1 的作用是在低电压的时候确保关闭振荡，此低电压值低于单片机的最低工作电压。需要注意的是如果 LVR 使能，可以不加 R1。		

注：电阻和电容值选取的原则是使 VDD 保持稳定并在$\overline{\text{RES}}$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$\overline{\text{RES}}$引脚的线请尽可能地短。

指令集介绍

指令集

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

为了更加容易了解各式各样的指令码，接下来按功能分组介绍它们。

指令周期

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

数据的传送

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

算术运算

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

逻辑和移位运算

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

分支和控制的转换

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

位运算

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

查表运算

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

其它运算

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

指令集摘要

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

表格惯例：

x：立即数

m：数据存储器地址

A：累加器

I：0~7 号位

Addr：程序存储器地址

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；否则不影响 TO 和 PDF 标志位。

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

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

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

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

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

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

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

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

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

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

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

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

AND A, x 累加器与立即数做“与”运算，结果放入累加器

说明： 本指令把累加器值、立即数做逻辑与，结果存放到累加器。

运算过程： ACC←ACC “AND” x

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

ANDM A, [m] 累加器与数据存储器做“与”运算，结果放入数据存储器

说明： 本指令把累加器值、数据存储器值做逻辑与，结果存放到数据存储器。

运算过程： [m]←ACC “AND” [m]

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

CALL addr 子程序调用

说明： 本指令直接调用地址所在处的子程序，此时程序计数器加一，将此程序计数器值存到堆栈寄存器中，再将子程序所在处的地址存放到程序计数器中。

运算过程： Stack←Program Counter+1

Program Counter ← addr

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

CLR [m] 清除数据存储器

说明： 本指令将数据存储器内的数值清零。

运算过程： [m] ←00H

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

CLR [m] . i 将数据存储器的第 i 位清“0”

说明： 本指令将数据存储器内第 i 位值清零。

运算过程： [m].i ←0

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

CLR WDT 清除看门狗定时器

说明： 本指令清除 WDT 计数器(从 0 开始重新计数)，暂停标志位(PDF)和看门狗溢出标志位(TO)也被清零。

运算过程： WDT ← 00H

PDF&TO ← 0

影响标志位

TO	PDF	OV	Z	AC	C
0	0	—	—	—	—

CLR WDT1 预清除看门狗定时器

说明： 必须搭配 CLR WDT2 一起使用，才可清除 WDT 定时/计数器(从 0 开始重新计数)。当程序只执行过该指令，没有执行 CLR WDT2 时，系统只会不会将暂停标志位(PDF)和计数溢出位(TO)清零，PDF 与 TO 保留原状态不变。

运算过程： WDT ← 00H*

PDF&TO ← 0*

影响标志位

TO	PDF	OV	Z	AC	C
0*	0*	—	—	—	—

CLR WDT2 预清除看门狗定时器

说明： 必须搭配 CLR WDT1 一起使用，才可清除 WDT 定时/计数器(从 0 开始重新计数)。当程序只执行过该指令，没有执行 CLR WDT1 时，系统只会不会将暂停标志位(PDF)和计数溢出位(TO)清零，PDF 与 TO 保留原状态不变。

运算过程： WDT ← 00H*

PDF&TO ← 0*

影响标志位

TO	PDF	OV	Z	AC	C
0*	0*	—	—	—	—

CPL [m] 对数据存储器取反，结果放入数据存储器

说明： 本指令是将数据存储器内保存的数值取反。

运算过程： [m] ← $[\overline{m}]$

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

CPLA [m] 对数据存储器取反，结果放入累加器

说明： 本指令是将数据存储器内保存的值取反后，结果存放在累加器中。

运算过程： ACC ← $[\overline{m}]$

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

DAA [m] 将加法运算后放入累加器的值调整为十进制数，并将结果放入数据存储器

说明 本指令将累加器高低四位分别调整为 BCD 码。如果低四位的值大于“9”或 AC=1，那么 BCD 调整就执行对原值加“6”，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] 数据存储器的内容减 1，结果放入数据存储器

说明： 本指令将数据存储器内的数值减一再放回数据存储器。

运算过程： [m] ← [m]-1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

DECA [m] 数据存储器的内容减 1，结果放入累加器

说明： 本指令将存储器内的数值减一，再放到累加器。

运算过程： ACC ← [m]-1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

HALT 进入暂停模式

说明： 本指令终止程序执行并关掉系统时钟，RAM 和寄存器内的数值保持原状态，WDT 计数器清“0”，暂停标志位(PDF)被设为 1， WDT 计数溢出位(TO)被清为 0。

运算过程： Program Counter ← Program Counter+1

PDF ← 1

TO ← 0

影响标志位

TO	PDF	OV	Z	AC	C
0	1	—	—	—	—

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

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

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

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

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

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

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

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

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

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

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

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

NOP 空指令

说明： 本指令不作任何运算，而只将程序计数器加一。

运算过程： Program Counter ← Program Counter+1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

OR A, [m] 累加器与数据存储器做“或”运算，结果放入累加器

说明： 本指令是把累加器、数据存储器值做逻辑或，结果放到累加器。

运算过程： ACC←ACC “OR” [m]

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

OR A, x 累加器与立即数做“或”运算，结果放入累加器

说明： 本指令是把累加器值、立即数做逻辑或，结果放到累加器。

运算过程： ACC←ACC “OR” x

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

ORM A, [m] 累加器与数据存储器做“或”运算，结果放入数据存储器

说明： 本指令是把累加器值、存储器值做逻辑或，结果放到数据存储器。

运算过程： [m]←ACC “OR” [m]

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

RET 从子程序返回

说明： 本指令是将堆栈寄存器中的程序计数器值送回程序计数器。

运算过程： Program Counter ← Stack

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RET A, x 从子程序返回，并将立即数放入累加器

说明： 本指令是将堆栈寄存器中的程序计数器值送回程序计数器，并将立即数送回累加器。

运算过程： Program Counter ← Stack

ACC ← x

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RETI 从中断返回

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

运算过程： Program Counter ← Stack
EMI ← 1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RL [m] 数据存储器左移一位，结果放入数据存储器

说明： 本指令是将数据存储器内的数值左移一位，第 7 位移到第 0 位，结果送回数据存储器。

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

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RLA [m] 数据存储器左移一位，结果放入累加器

说明： 本指令是将存储器内的数值左移一位，第 7 位移到第 0 位，结果送到累加器，而数据存储器内的数值不变。

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

影响标志位

TO	PDF	OV	Z	AC	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

说明： 本指令是将存储器内的数值与进位标志左移一位，第 7 位取代进位标志，进位标志移到第 0 位，结果送回数据存储器。

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

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

RLCA [m] 带进位将数据存储器左移一位，结果放入累加器

说明： 本指令是将存储器内的数值与进位标志左移一位，第七位取代进位标志，进位标志移到第 0 位，结果送回累加器。

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

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

RR [m] 数据存储器右移一位，结果放入数据存储器

说明： 本指令是将存储器内的数值循环右移，第 0 位移到第 7 位，结果送回数据存储器。

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

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

RRA [m] 数据存储器右移一位，结果放入累加器

说明： 本指令是将数据存储器内的数值循环右移，第 0 位移到第 7 位，结果送回累加器，而数据存储器内的数值不变。

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

影响标志位

TO	PDF	OV	Z	AC	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

说明： 本指令是将存储器内的数值加进位标志循环右移，第 0 位取代进位标志，进位标志移到第 7 位，结果送回存储器。

运算过程： [m].i ←[m].(i+1); (i=0~6)

[m]. 7 ←C

C ← [m].0

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

RRCA [m] 带进位将数据存储器右移一位，结果放入累加器

说明： 本指令是将数据存储器内的数值加进位标志循环右移，第 0 位取代进位标志，进位标志移到第 7 位，结果送回累加器，数据存储器内的数值不变。

运算过程： ACC.i ←[m].(i+1); (i=0~6)

ACC.7 ←C

C ← [m].0

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	√

SBC A,[m] 累加器与数据存储器、进位标志相减，结果放入累加器

说明： 本指令是把累加器值减去数据存储器值以及进位标志的取反，结果放到累加器。

运算过程： ACC←ACC+[$\overline{m}$]+ C

影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SBCM A,[m] 累加器与数据存储器、进位标志相减，结果放入数据存储器

说明： 本指令是把累加器值减去数据存储器值以及进位标志取反，结果放到数据存储器。

运算过程： [m]←ACC+[$\overline{m}$]+C

影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SDZ [m] 数据存储器减 1，如果结果为“0”，则跳过下一条指令

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

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

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

运算过程： 如果[m]-1=0，跳过下一条指令执行再下一条。

ACC ←([m]-1)

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SET [m] 置位数据存储器

说明： 本指令是把存储器内的数值每个位置为 1。

运算过程： [m] ← FFH

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SET [m]. i 将数据存储器的第 i 位置“1”

说明： 本指令是把存储器内的数值的第 i 位置为 1。

运算过程： [m].i ←1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SIZ [m] 数据存储器加 1，如果结果为“0”，则跳过下一条指令

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

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

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

运算过程： 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 如果数据存储器的第 i 位不为“0”，则跳过下一条指令

说明： 本指令是判断数据存储器内的数值的第 i 位，若不为 0，则程序计数器再加 1，跳过下一行指令，放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以取得正确的指令(二个指令周期)。否则执行下一条指令(一个指令周期)。

运算过程： 如果 [m].i≠0，跳过下一行指令。

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SUB A, [m] 累加器与数据存储器相减，结果放入累加器

说明： 本指令是把累加器值、数据存储器值相减，结果放到累加器。

运算过程： ACC←ACC+[$\overline{m}$]+1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SUB A, x 累加器与立即数相减，结果放入累加器

说明： 本指令是把累加器值、立即数相减，结果放到累加器。

运算过程： ACC←ACC+$\overline{x}$+1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SUBM A, [m] 累加器与数据存储器相减，结果放入数据存储器

说明： 本指令是把累加器值、存储器值相减，结果放到存储器。

运算过程： [m]←ACC+[$\overline{m}$]+1

影响标志位

TO	PDF	OV	Z	AC	C
—	—	√	√	√	√

SWAP [m] 交换数据存储器的高低字节，结果放入数据存储器

说明： 本指令是将数据存储器的低四位和高四位互换，再将结果送回数据存储器。

运算过程： [m].7~[m].4 ↔[m].3~[m].0

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SWAPA [m] 交换数据存储器的高低字节，结果放入累加器

说明： 本指令是将数据存储器的低四位和高四位互换，再将结果送回累加器。

运算过程： ACC.3~ACC.0← [m].7~[m].4

ACC.7~ACC.4← [m].3~[m].0

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SZ [m] 如果数据存储器为“0”，则跳过下一条指令

说明： 本指令是判断数据存储器内的数值是否为 0，为 0 则跳过下一行指令，即放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以得正确的指令(二个指令周期)。否则执行下一条指令(一个指令周期)。

运算过程： 如果 [m] = 0, 跳过下一行指令。

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SZA [m] 数据存储器送至累加器，如果内容为“0”，则跳过下一条指令

说明： 本指令是判断存储器内的数值是否为 0，若为 0 则跳过下一行指令，即放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以得正确的指令(二个指令周期)。并把存储器内值送到累加器，而存储器的值保持不变。否则执行下一条指令(一个指令周期)。

运算过程： 如果[m] = 0，跳过下一行指令，并 ACC← [m]。

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

SZ [m]. i 如果数据存储器的第 i 位为“0”，则跳过下一条指令

说明： 本指令是判断存储器内第 i 位值是否为 0，若为 0 则跳过下一行指令，即放弃在目前指令执行期间所取得的下一条指令，并插入一个空周期用以得正确的指令(二个指令周期)。否则执行下一条指令(一个指令周期)。

运算过程： 如果 [m].i = 0，跳过下一行指令。

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

TABRDC [m] 读取 ROM 当前页的内容，并送至数据存储器和 TBLH

说明： 本指令是将表格指针指向程序寄存器当前页，将低字节送到存储器，高字节直接送到 TBLH 寄存器内。

运算过程： [m] ←程序存储器低字节
TBLH←程序存储器高字节

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

TABRDL [m] 读取 ROM 最后一页的内容，并送至数据存储器和 TBLH

说明： 本指令是将 TABLE 指针指向程序寄存器最后页，将低字节送到存储器，高字节直接送到 TBLH 寄存器内。

运算过程： [m] ←程序存储器低字节
TBLH←程序存储器高字节

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	—	—	—

XOR A, [m] 累加器与立即数做“异或”运算，结果放入累加器

说明： 本指令是把累加器值、数据存储器值做逻辑异或，结果放到累加器。

运算过程： ACC←ACC “XOR” [m]

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

XORM A, [m] 累加器与数据存储器做“异或”运算，结果放入数据存储器

说明： 本指令是把累加器值、数据存储器值做逻辑异或，结果放到数据存储器。

运算过程： [m]←ACC “XOR” [m]

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

XOR A, x 累加器与数据存储器做“异或”运算，结果放入累加器

说明： 本指令是把累加器值与立即数做逻辑异或，结果放到累加器。

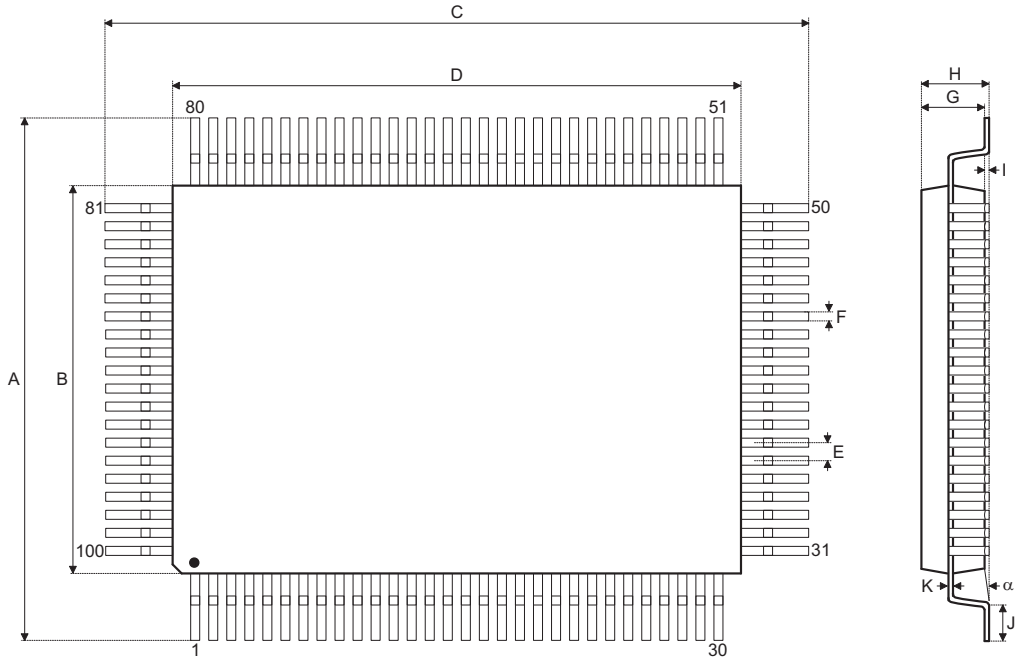
运算过程： ACC←ACC “XOR” x

影响标志位

TO	PDF	OV	Z	AC	C
—	—	—	√	—	—

封装信息

100-pin QFP (14×20) 外形尺寸



符号	尺寸(单位：mil)		
	最小	正常	最大
A	18.5	—	19.2
B	13.9	—	14.1
C	24.5	—	25.2
D	19.9	—	20.1
E	—	0.65	—
F	—	0.3	—
G	2.5	—	3.1
H	—	—	3.4
I	—	0.1	—
J	1	—	1.4
K	0.1	—	0.2
α	0°	—	7°

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

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

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

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