

四位的微控制器 W541C250



1. W541C250单片机介绍

在引脚分配和IC的功能上，W541C250与W741C250完全兼容。它是一个带有LCD驱动的微功耗、高性能的四位微控制器；它包含有一个四位的ALU、两个八位定时器、一个分频器、一个24×4的LCD驱动、五个四位的I/O端口（包含一个可直接驱动LED的输出端口）；它具有五个中断源和八级堆栈；W541C250有两种省电的工作模式，即HOLD模式、STOP模式，这样可以把功耗降到最小。

W541C250非常适合于遥控器、多I/O产品、键盘控制器、时钟、语音合成，手柄游戏等产品。

2. 主要特性

- 工作电压: 2.4V 到 5.5V (LCD 驱动电压: 3.0V, or 4.5V)
- 操作频率可达4 MHz
- 振荡器可选为晶振或RC振荡电路
- 振荡器可以为选择高频（400KHZ到4MHZ）或低频时钟（32.768KHZ）
- 存储器(ROM)
 - 2048 × 16位的程序存储器(包含 2K × 4 位的查表区)
 - 128 × 4位的数据存储器RAM(包含16工作寄存器)
 - 24 × 4 LCD 数据存储器RAM
- 21 输入/输出脚
 - 单向输入口: 2个端口/8个脚
 - 双向输入/输出口: 2个端口/8个脚
 - 单向输出口: 1个端口/4个脚 (具有很大的灌电流能力, 可用来驱动LED)
 - MFP输出脚: 1个脚 (MFP)
- 省电模式
 - HOLD模式: 除主振和辅助振振荡器外, 其余部分停止工作
 - STOP模式: 所有部分都停止工作, 包括主振但不包括辅振
- 五个中断类型
 - 四个内部中断(分频器0, 分频器1, 定时器0, 定时器1)
 - 一个外部中断(RC端和 \overline{INT} 脚)
- LCD 驱动输出
 - 24 段 × 4 公共端
 - 可设为静态, 1/2 占空比(1/2偏压), 1/3占空比(1/2或1/3偏压), 1/4占空比 (1/3偏压) 等驱动模式
 - LCD驱动输出脚也可以输出电平, 它由代码选项寄存器(code option)控制



- MFP 输出脚
 - MFP的输出端口可以通过软件选择调制频率或单一频率
 - MFP的输出频率可由定时器1确定
- 内置一个14位的分频器
- 两个八位的向下定时计数器
 - 定时器0: 两个内部时钟频率可选择($F_{osc}/4$ 或 $F_{osc}/1024$)
 - 定时器1: 包含有自动重装载功能; 时钟源可以是两个内部时钟频率 (F_{osc} 或 $F_{osc}/64$)之一或外部RC.0口的下降沿; 控制MFP脚的输出频率
- 内含18/14-位的看门狗
 - 可以通过命令允许或禁止看门狗
- 丰富的指令集:116 条指令
- 8 级堆栈
- 1 μ S 指令周期(4 MHz 工作频率)
- 64 管脚的QFP封装

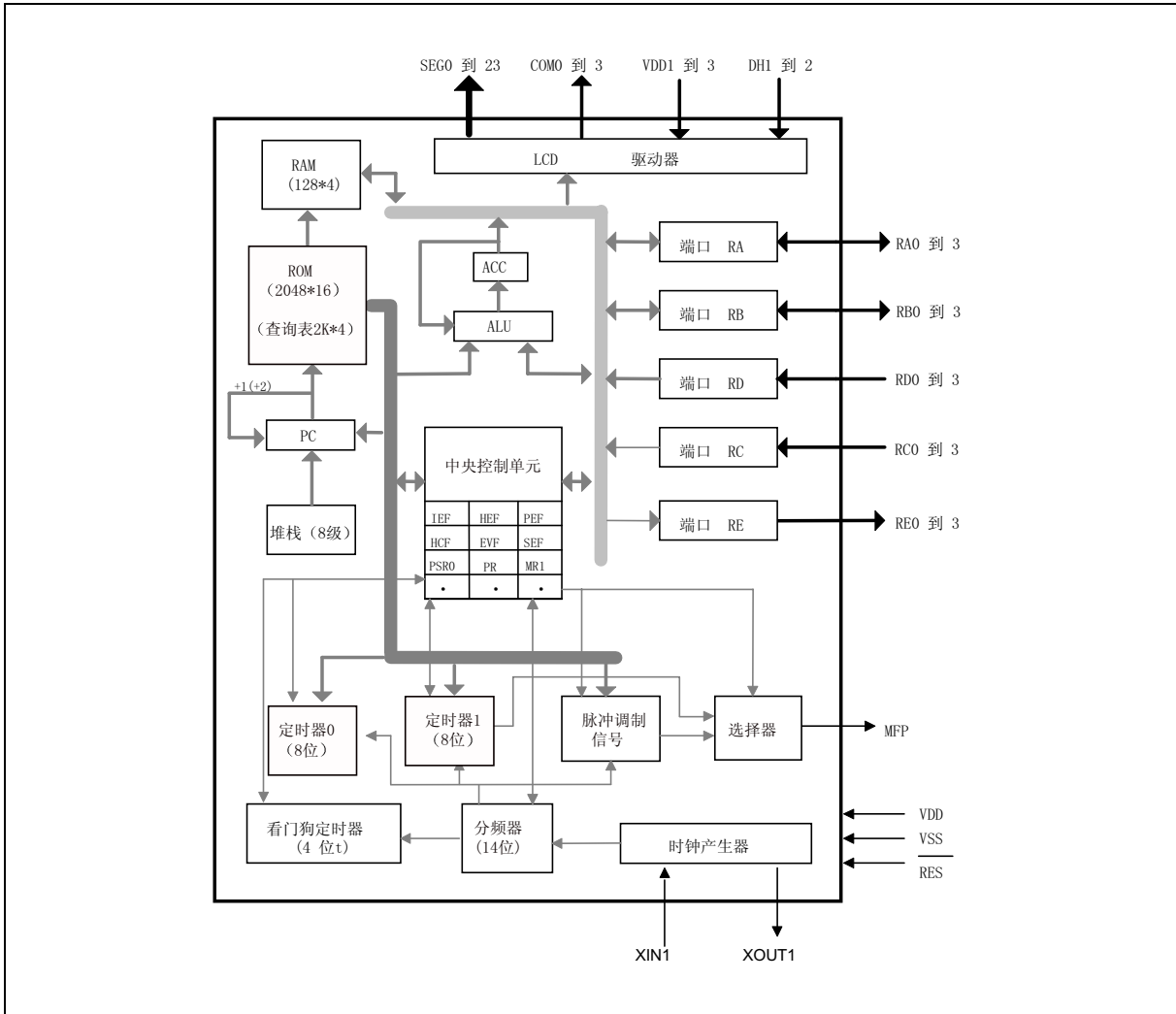
3. 开发工具支持

软件方面, 相关的软件包括: 汇编程序, 程序编辑器, LCD编辑器、LCD仿真器等部分; 运用它可以编辑、仿真源代码程序和LCD; 汇编程序可以把文本程序编译生成用户实际应用需要的.OBJ文件。

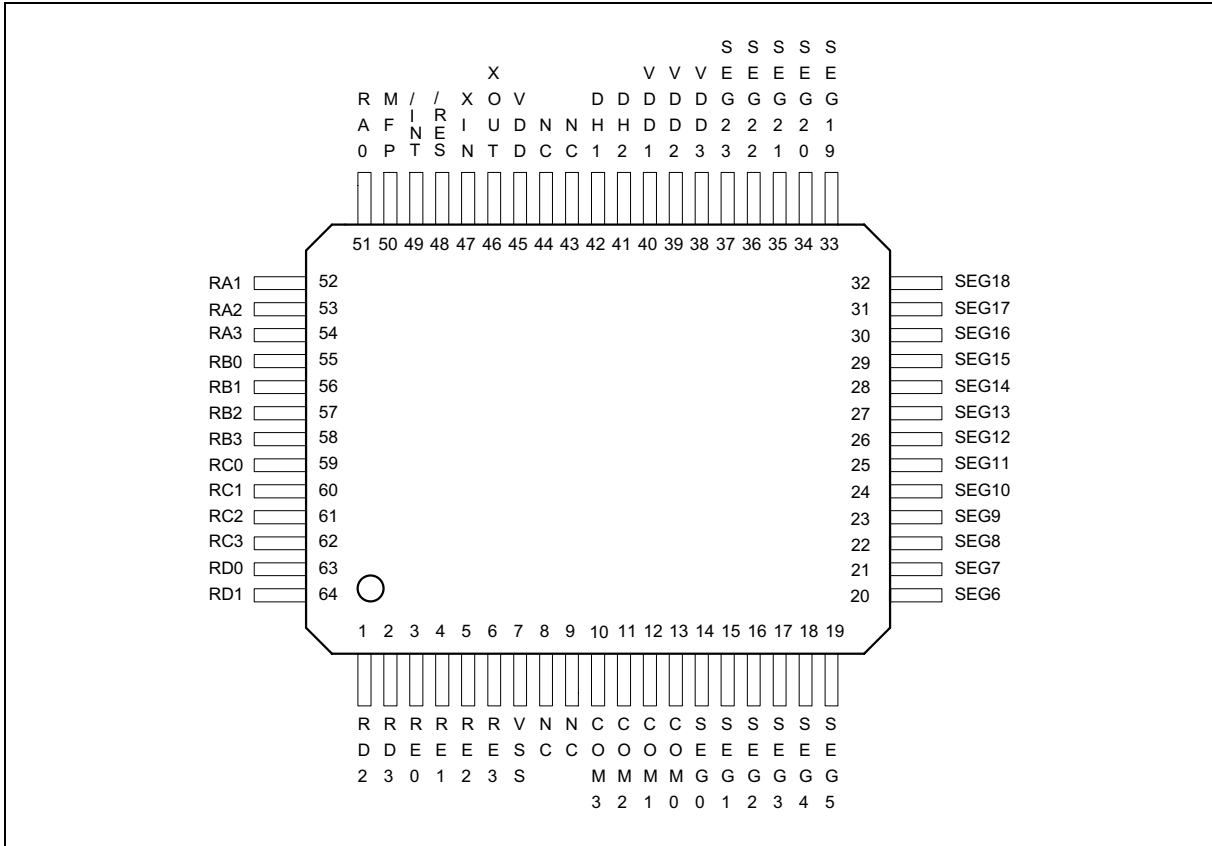
硬件方面, 我们提供ICE、烧录器、带Flash的芯片; Winbond ICE是用来进行调试和验证程序的; 烧录器内置于仿真上, 可以用它将.obj程序烧入带有EPROM的芯片中, 这样用户可在样板来做进一步验证实际的工作情况。

4. W541C250的系统结构

4.1 方框图



4.2 管脚图



4.3 管脚说明图

管脚	输入/出	作 用																									
XIN	输入	振荡器的输入脚。 由code option 控制将其连接到晶振或电阻器上用来产生系统时钟。																									
XOUT	输出	振荡器的输出脚。内置20PF的电容。 由code option 控制将其连接到晶振或电阻器上用来产生系统时钟。																									
RA0-RA3	输入/出	输入/输出口。 输入/输出模式由端口模式寄存器1 (PM1) 控制。																									
RB0-RB3	输入/出	输入/输出口。 输入/输出模式由端口模式寄存器2 (PM2) 控制。																									
RC0-RC3	输入	4 位的输入口。 每个管脚都具有独立的中断功能。																									
RD0-RD3	输入	4 位的输入口。																									
RE0-RE3	输出	输出端口。 提供大灌电流可以驱动LED。																									
MFP	输出	输出端口。 输出调制频率或者电平, 或者通过MR1确定的定时器1的时钟输出。																									
$\overline{\text{RES}}$	输入	带上拉电阻的系统复位脚。																									
$\overline{\text{INT}}$	输入	外部中断, 内部有上拉电阻。																									
SEG0-SEG23	输出	LCD 段输出。 能够用作直流输出。																									
COM0-COM3	输出	LCD 信号公用端输出。 <table border="1" data-bbox="565 1339 1154 1545"> <thead> <tr> <th></th> <th>静态</th> <th>1/2占空比</th> <th>1/3占空比</th> <th>1/4占空比</th> </tr> </thead> <tbody> <tr> <td>COM0</td> <td>用</td> <td>用</td> <td>用</td> <td>用</td> </tr> <tr> <td>COM1</td> <td>不用</td> <td>用</td> <td>用</td> <td>用</td> </tr> <tr> <td>COM2</td> <td>不用</td> <td>不用</td> <td>用</td> <td>用</td> </tr> <tr> <td>COM3</td> <td>不用</td> <td>不用</td> <td>不用</td> <td>用</td> </tr> </tbody> </table> LCD 的输出频率可调, 由code option 控制。		静态	1/2占空比	1/3占空比	1/4占空比	COM0	用	用	用	用	COM1	不用	用	用	用	COM2	不用	不用	用	用	COM3	不用	不用	不用	用
	静态	1/2占空比	1/3占空比	1/4占空比																							
COM0	用	用	用	用																							
COM1	不用	用	用	用																							
COM2	不用	不用	用	用																							
COM3	不用	不用	不用	用																							
DH1, DH2	输入	倍值电容的连接端。																									
VDD1, VDD2, VDD3	输入	提供电源, 参见功能描述。																									
VDD	输入	连接电源的正极 (+)。																									
VSS	输入	连接电源的负极 (-)。																									

5. 功能说明

5.1 程序计数器(PC)

程序计数器为11位二进制计数器（PC0至PC10）；它的寻址范围是 2048×16 ，可以访问所有片内ROM的内容即程序指令；当要执行跳转指令、调用子程序、进入中断服务程序和初始化指令时，这些指令的相应地址存储到程序计数器中。格式如下表：

项 目	地 址	中断优先级
初始化	000H	-
INT 0 (分频器0)	004H	1 st
INT 1 (定时器0)	008H	2 nd
INT 2 (RC口)	00CH	3 rd
INT 3 (分频器1)	014H	4 th
INT 4 (定时器1)	020H	5 th
JP 指令	XXXH	-
调用子程序	XXXH	-

5.2 堆栈寄存器(STACK)

堆栈结构是11位×8级（先进后出），执行调子程序或中断时，程序计数器(PC)自动压入栈顶；调子程序或中断执行完毕，必需执行RTN指令，以将栈顶的内容弹出到程序计数器(PC)中；当压入栈顶中的内容超过8级时，第1级的内容将掉失，也就是说堆栈只有8层。

5.3 程序存储器（ROM）

程序存储器ROM用于存储程序代码和查表数据；ROM的前面3/4部分（000至5FF）只能存储指令代码，后面的1/4（600至7FF的2048×4位）可以存储指令代码或查询表，每个查询表的元素均为四位，所以可查询的元素地址达2048个；TABL和TABH两个寄存器可以存储查询的地址，由MOV TABH, R和MOV TABL, R指令控制；执行MOVC R后，由TABH, TABL和ACC确定的查询表的地址，并且将其内含值送到数据RAM，详细参阅指令表，程序存储器的结构如图1示。

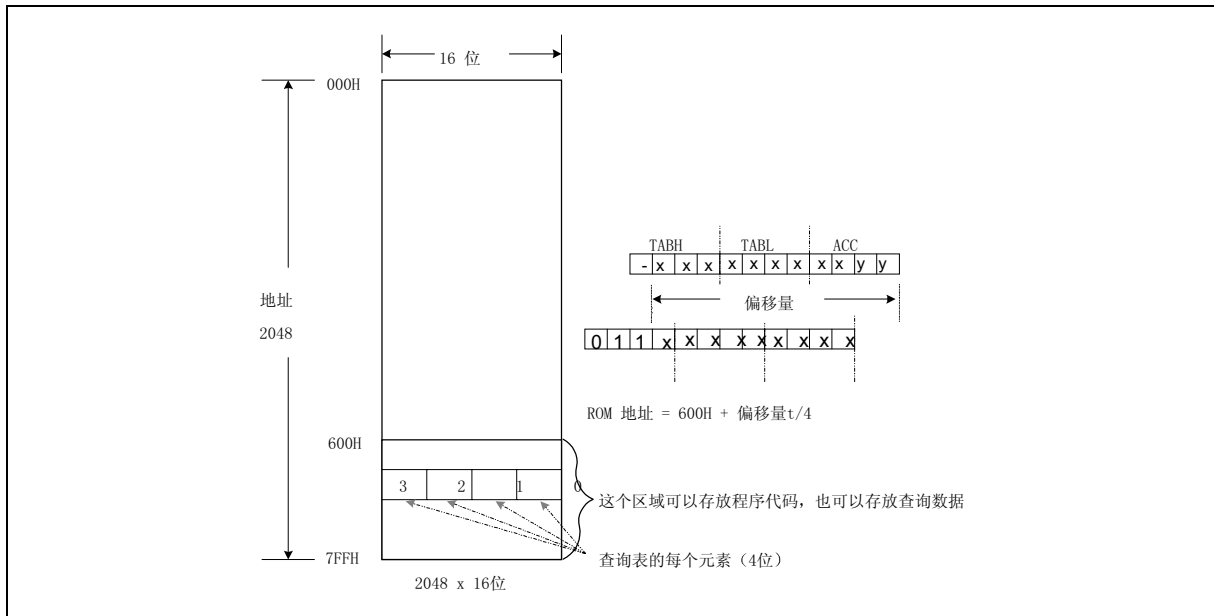


图 1. 程序存储器结构

5.4 数据存储器 (RAM)

5.4.1 结构

静态数据存储器 (RAM) 用于存储数据, 其结构为128×4位, 数据存储器可以直接寻址, 可以间接寻址, 其结构如图2所示。数据存储器的前16个地址 (00至0FH) 为工作寄存器; 他存储空间为通用存储空间, 但不能进行立即数寻址操作; 据存储的定位和间接寻址模式的页 (PAGE) 寄存器的关系将在下面介绍。

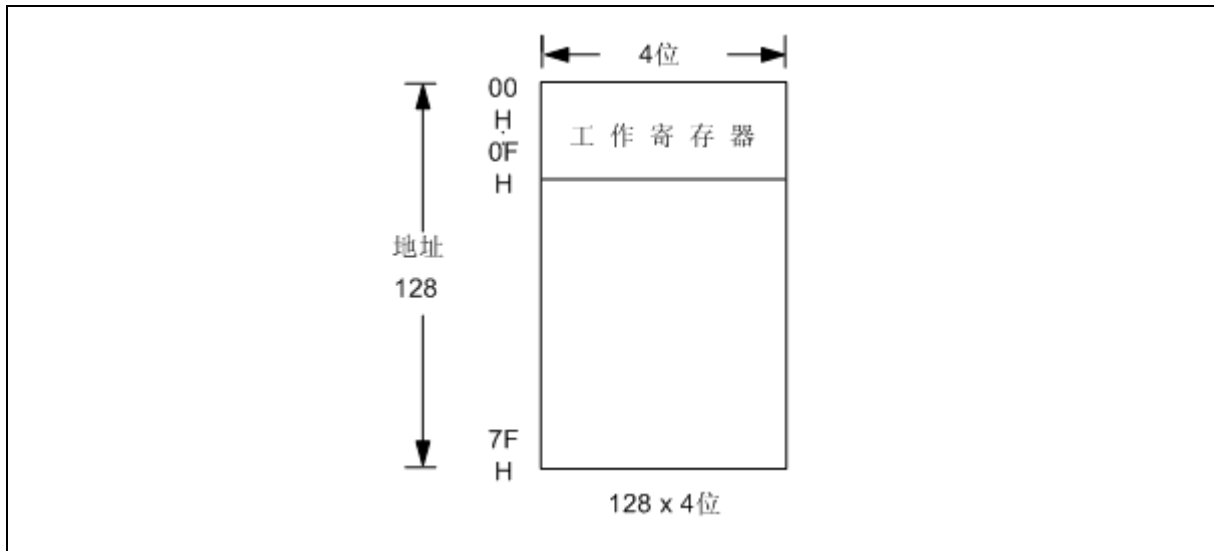


图 2.数据存储器结构

5.4.2 页寄存器(PAGE)

页寄存器为4位二进制寄存器，位定义如下：

	3	2	1	0
PAGE	R/W	R/W	R/W	R/W

注: R/W 表示可读可写

PAGE.3保留

PAGE.2, PAGE.1, PAGE.0 间接寻址模式预选位：

000 = 页 0 (00H-0FH)	001 = 页 1 (10H-1FH)
010 = 页 2 (20H-2FH)	011 = 页 3 (30H-3FH)
100 = 页 4 (40H-4FH)	101 = 页 5 (50H-5FH)
110 = 页 6 (60H-6FH)	111 = 页 7 (70H-7FH)

5.5 累加器(ACC)

累加器 (ACC)为4位寄存器，用于暂存ALU的运算结果；在存储器、I/O口、寄存器间间接传送的数据。

5.6 算术逻辑单元 (ALU)

该行路完成算术和逻辑运算，ALU具有如下功能：

- 逻辑运算指令: ANL, XRL, ORL

- 控制转移指令: JB0, JB1, JB2, JB3, JNZ, JZ, JC, JNC, DSKZ, DSKNZ, SKB0, SKB1, SKB2, SKB3
- 移位操作指令: SHRC, RRC, SHLC, RLC
- 二进加/减指令: ADC, SBC, ADD, SUB, ADU, DEC, INC

以上任意指令完成后, CF和ZF的状态存储于内部寄存器, 执行MOV R, CF可读出CF, MOV CF, R可以存储CF。

5.7 时钟发生器

在option code 的选项中可以选外部连接的晶振或RC振荡电路来产生系统时钟。W541C250使用晶振选项时, XIN和XOUT需与晶体或陶瓷滤波器相联接, 如果需要精确频率时, 还必需在XIN和VSS间接一个匹配电容, 选择代码(option code)可以选择高频时钟(对应于400KHZ至4MHZ)或选择低频时钟(对应于32KHZ); 应用RC振荡时, XIN和XOUT之间需接一个电阻, 阻值范围为20K到1.6K, 如图3所示; 产生时钟频率的高低必需与时钟频率高低选项设置一致; 一个机器周期有4个系统时钟周期, 如果系统时钟为4MHZ时则机器周期为1us。

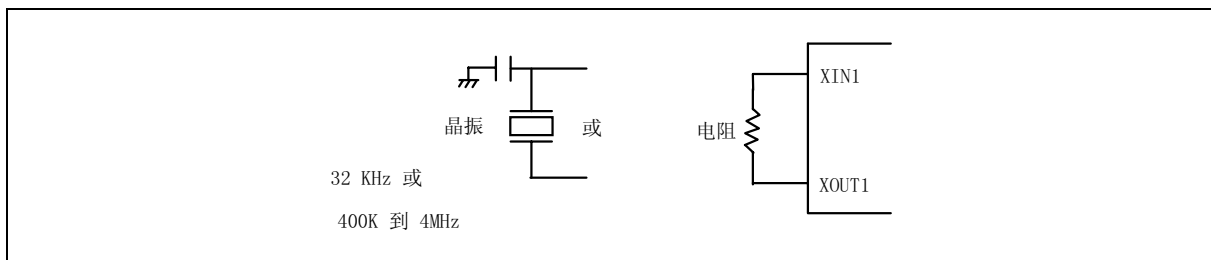


图3. 振荡器的配置

5.8 分频器0

分频器为14位二进制向上计数器, 用于产生周期的中断如图4。系统启动后, 分频器每一个系统时钟(Fosc)都自动加1; 溢出时, 分频器0的事件标志为1(EVF.0=1), 如果分频器0中断允许标志位为1(IEF.0=1), 将执行中断; 如果退出HOLD允许标志为1(HEF.0=1), 则中止HOLD状态; 执行CLR DIVR0可以重置分频器0的后四阶。

5.9 看门狗 (WDT)

看门狗定时器(WDT)为4位向上计数器, 用在程序出现莫名其妙的错误时, 保证程序的正常运行。WDT的相应选择(option code为1时, WDT开始工作; 若WDT溢出, 系统将复位; 初始复位时, WDT的输入时钟为Fosc/1024; 执行SET PMF, #08H(或CLR PMF, #08H), 可将WDT的输入时钟切换为Fosc/16384(或Fosc/1024); 执行CLR WDT, 可将WDT的内容复位; 正常操作时, 应用程序在WDT溢出前, 需先使WDT复位; WDT的溢出表明操作超出控制范围, 芯片需复位。系统时钟为32KHZ, 且WDT时钟输入为Fosc/16348时, WDT的溢出周期为500ms; WDT相应的选择位为0时, 将禁止WDT的功能, 分频器0和看门狗定时器的结构如图5所示。

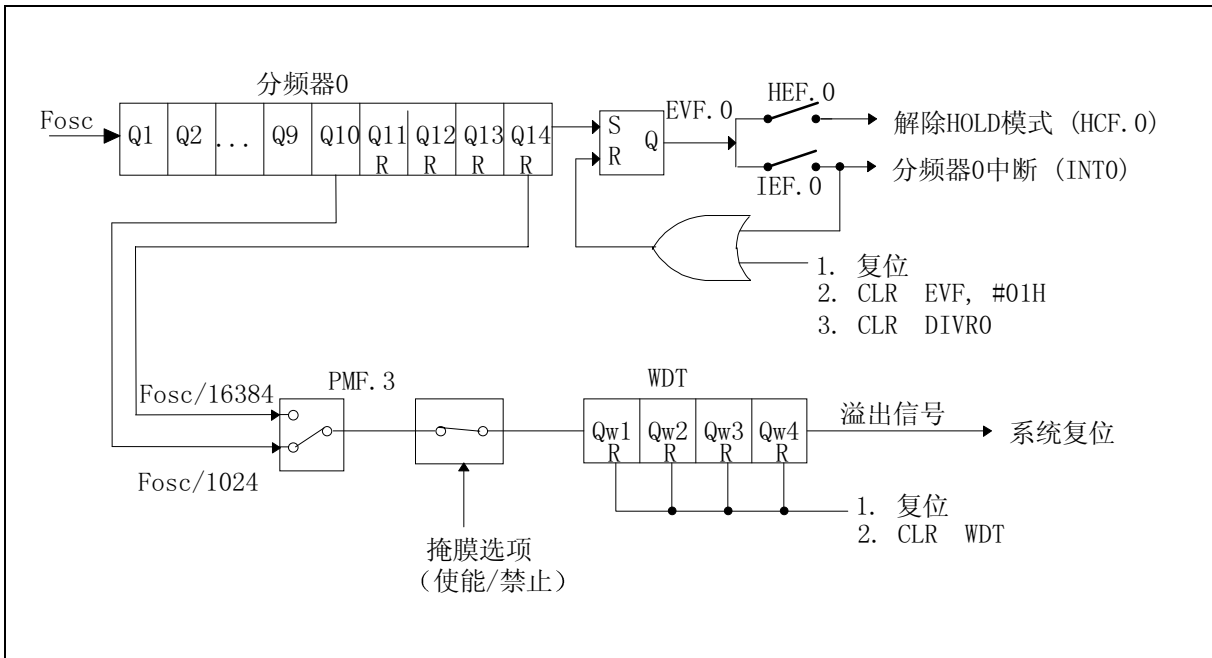


图4.分频器和看门狗定时器的结构

5.10.1 参数标志Parameter Flag (PMF)

PMF为一个四位二进制寄存器 (PMF.0 to PMF.3).; PMF由 SET PMF, #I 或CLR PMF, #I 指令控制, 位定义如下:

	3	2	1	0
PMF	W	-	-	-

注: W 表示只可写不可读

PMF 0, PMF 1, PMF 2 保留

- PMF 3 = 0 看门狗定时器的基频为Fosc/1024.
- = 1 看门狗定时器的基频为Fosc/16384

5.10 定时器0 Timer0 (TM0)

定时器0 (TM0)是一个可编程8位二进制向下计数器; 执行MOV TM0L (TM0H), R 指令可将特定值载入TM0, 该指令执行后, 则TM0 将停止向下计数(若TM0 正向下计数), MR0.3重置为0, 且TM0被设定; 若MR0.3设置为1, 则复位事件标志位为0 (EVF.1), TM0开始计数; 减为FFH时, 定时器0停止计数, 并且产生溢出信号(EVF.1=1); 如果定时器0中断允许标志为(IEF.1=1)1, 则执行中断; 若退出保持标志为1 (HEF.1=1)则中止HOLD状态; 设置MR0.0为1或将MR0.0重置为0, 可以将定时器0时钟输

入设置为 $F_{osc}/1024$ 或 $F_{osc}4$ ，定时器的初始设置为 $F_{osc}/4$ ；定时器0的结构如图6示。

若定时器0的时钟输入为 $F_{osc}/4$ ：

$$\text{所得定时器0的间隔} = (\text{预置值} + 1) \times 4 \times 1/F_{osc}$$

若定时器0的时钟输入为 $F_{osc}/1024$ ：

$$\text{所得定时器0的间隔} = (\text{预置值} + 1) \times 1024 \times 1/F_{osc}$$

预置值：定时器0预置值的十进制数据

F_{osc} ：时钟晶振频率

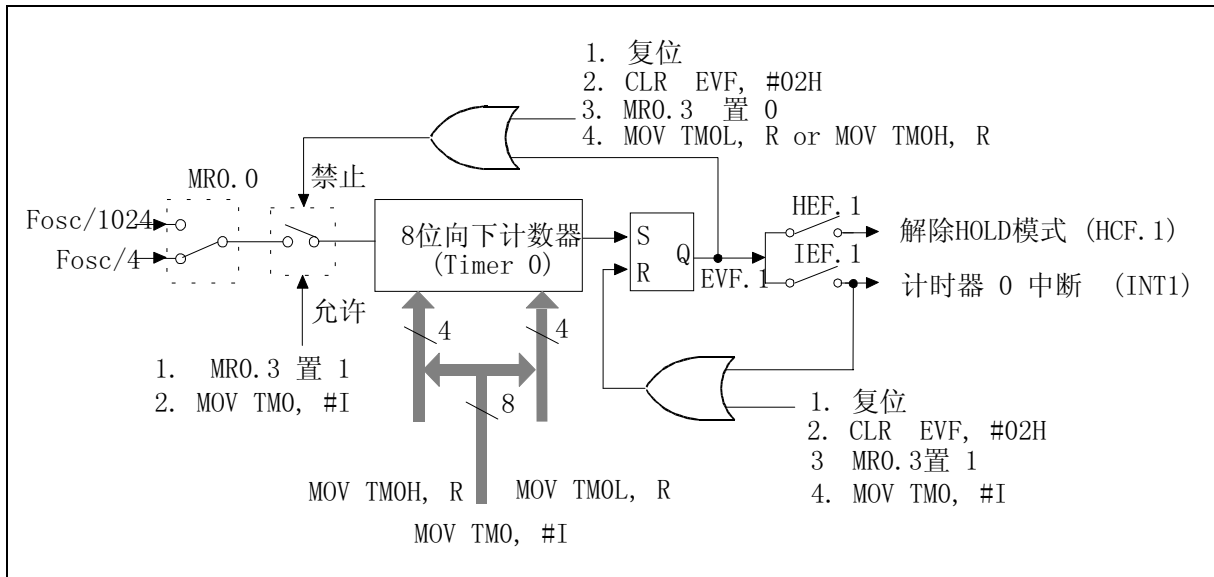


图6. 定时器0的结构

5.10.1 模式寄存器0 Mode Register0(MR0)

模式寄存器0是4位二进制寄存器（MR0.0至MR0.3）。MR0用于控制定时器0，位定义如下：

	3	2	1	0
MR0	W	-	-	W

注：W 表示只可写不可读

MR0 0 = 0 定时器0的内置基本频率为 $F_{osc}/4$ 。

= 1 定时器0的内置基本频率为 $F_{osc}/1024$ 。

MR0 1 , MR0 2 保留

MR0 3 = 0 定时器零停止向下计数
 = 1 定时器零开始向下计数

5.11 定时器1 timer1(TM1)

定时器1 (TM1)为可编程8位二进制向下计数器.结构如图7所示。定时器1可记录外部事件或将特定频率输出到MFP脚。定时器1的输入时钟可以是以下三个中的一个：Fosc/64，Fosc，或RC.0输入脚获得的外部时钟，是通过设置模式寄存器1(MR1)的位0和位1选定的。初始复位时，定时器1的输入时钟为Fosc；若选择外部时钟作为定时器1的时钟源，每个RC.0的下降沿使定时器1的值减1；执行MOV TM1L, R或MOV TM1H, R指令，将R中的数据载入自动重载缓冲器；同时，禁止TM1下行计数（即MR1.3重置为0）；若设置MR1的第3位（MR1.3=1），自动重载缓冲器的内容载入TM1，定时器1开始向下计数，且设置事件标志位7（EVF.7=0）。若在计时时执行MOV TM1, #I指令硬件完成以下动作：EVF.7复位；MR1.3复位为0；自动重载缓冲器刷新；TM1的刷新；MR1.3置1；TM1开始向下计时。定时器减为FF时，将溢出（EVF.7=1）；且自动将该特定的数据重新载入缓冲器，定时器1又开始向下计数。此时，若中断允许标志位为1（IEF.7=1），将产生中断，若退出HOLD允许标志7设置1（HEF.7=1），终止HOLD状态。MR1.2位控制定时器1的特定频率送到MFP输出脚，MR1.3位控制定时器的停止或重新开始计数。

定时器1的时钟输入为FT时：

$$\text{所得的定时器1间隔} = (\text{预置值} + 1) / FT$$

$$\text{MFP输出脚所需的率} = FT \div (\text{预置值} + 1) \div 2 \text{ (HZ)}$$

预置值：定时器1预置值的十进制数

Fosc: 时钟晶振频率

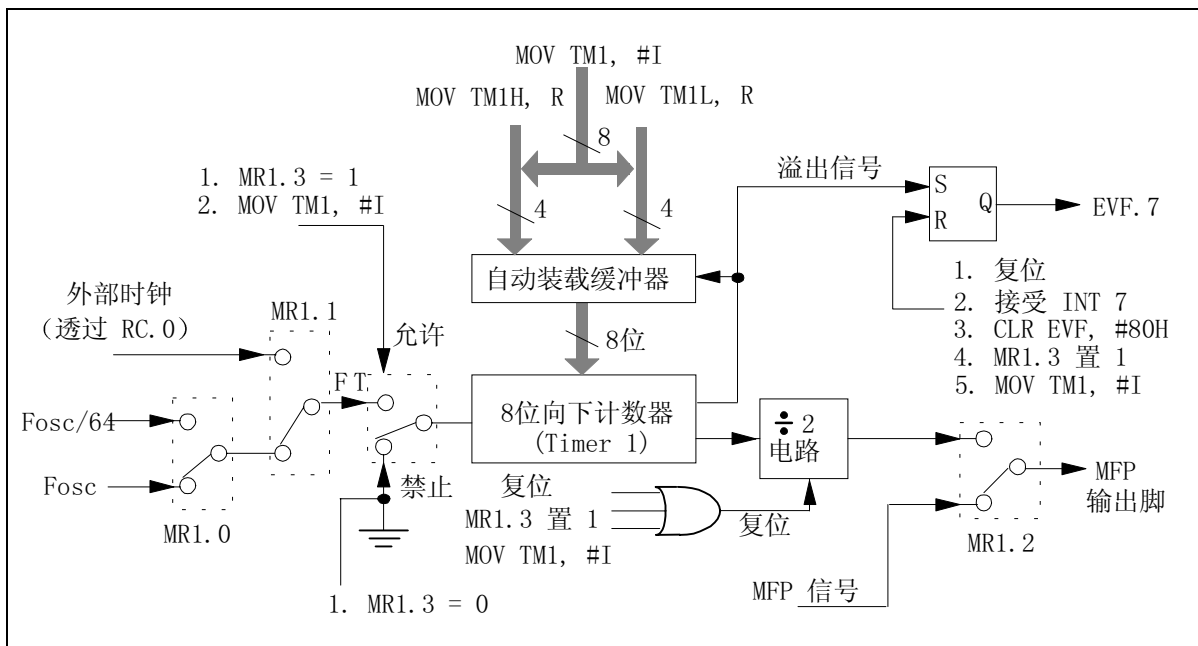


图7 定时器1的结构图

例如：FT为32768HZ,根据TM1的预置值，MFP脚将输出一个单一音调信号，该信号的频率范围是64HZ至16384HZ。音调频率与TM1预置值之间的关系如下表所示。

		3rd octave		4th octave		5th octave	
		Tone frequency	TM1 preset value & MFP frequency	Tone frequency	TM1 preset value & MFP frequency	Tone frequency	TM1 preset value & MFP frequency
T O N E	C	130.81	7CH 131.07	261.63	3EH 260.06	523.25	1EH 528.51
	C#	138.59	75H 138.84	277.18	3AH 277.69	554.37	1CH 564.96
	D	146.83	6FH 146.28	293.66	37H 292.57	587.33	1BH 585.14
	D#	155.56	68H 156.03	311.13	34H 309.13	622.25	19H 630.15
	E	164.81	62H 165.49	329.63	31H 327.68	659.26	18H 655.36
	F	174.61	5DH 174.30	349.23	2EH 372.36	698.46	16H 712.34
	F#	185.00	58H 184.09	369.99	2BH 390.09	739.99	15H 744.72
	G	196.00	53H 195.04	392.00	29H 420.10	783.99	14H 780.19
	G#	207.65	4EH 207.39	415.30	26H 443.81	830.61	13H 819.20
	A	220.00	49H 221.40	440.00	24H 442.81	880.00	12H 862.84
	A#	233.08	45H 234.05	466.16	22H 468.11	932.23	11H 910.22
	B	246.94	41H 248.24	493.88	20H 496.48	987.77	10H 963.76

注：中央音调为 A4 (440 Hz).

5.11.1 模式寄存器 1 Mode Register1(MR1)

MR1是4位二进制寄存器（MR1.0至MR1.3）。MR1用于控制定时器1，位定义如下：

	3	2	1	0
MR1	W	W	W	W

注：W 表示只可写不可读

MR1 0 = 0 定时器1的内置基本频率为Fosc

= 1 定时器1的内置基本频率为Fosc/64

MR1 1 = 0 定时器1的内置基本频率源为内部时钟

= 1 定时器1的内置基本频率源为从 RC.0输入的外部时钟

MR1 2 = 0 MFP发生器的设定波形从MFP输入脚

= 1 定时器1的设定频率为从MFP输出脚

MR1 3 = 0 定时器1停止向下计数

= 1 定时器开始向下计数

5.12 输入/输出口RA, RB

RA口有RA.0 到 RA.3脚, RB口有 RB.0 到 RB.3脚。在初始复位时, 输入/输出口RA和RB口均处于输入状态; RA口和RB口作为输出口时, 由PM0寄存器选择CMOS或NMOS开漏(open drain)输出模式; 当RA/RB为NMOS输出类型时, 不能使其悬浮, 并需将其设置成低; 否则, 会产生漏电流。PM1和PM2寄存器设定RA口或RB口的各个脚的输入/输出模式, MOVA R, RA或MOVA R, RB指令完成输入功能; MOV RA, R或MOV RB, R完成输出功能。详细参阅指令性表和图8。

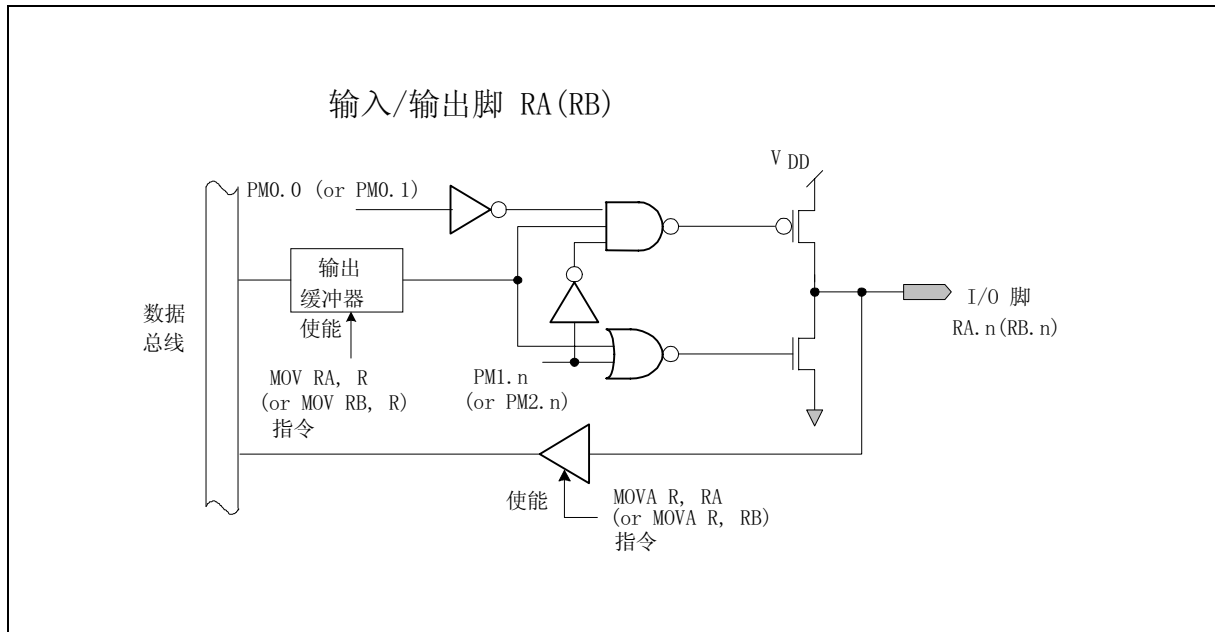


图8. 输入/输出脚的结构

5.13 端口模式0寄存器 (PM0)

端口模式0寄存器为4位二进制寄存器 (PM0.0 to PM0.3). PM0 决定输入或输出口的构成, 由MOV PM0, #1指令控制, 位定义如下:

	3	2	1	0
PM0	w	w	w	w

注: W 表示只可写不可读

- PM0 0 = 0 RA为CMOS 输出
- = 1 RA 口NMOS 开漏极输出(open drain)
- PM0 1 = 0 RB为CMOS 输出;
- = 1 RB 口NMOS开漏极输出
- PM0 2 = 0 禁止RC口的上拉电阻
- = 1 允许RC口的上拉电阻
- PM0 3 = 0 禁止RD口的上拉电阻
- = 1 允许RD口的上拉电阻

5.14 端口模式寄存器1 (PM1)

端口模式1 寄存器为4位二进制寄存器 (PM1.0 到 PM1.3). PM1控制RA口的输入和输出模式, 由MOV PM1, #1指令控制, 位定义如下:

	3	2	1	0
PM1	w	w	w	w

注: W 表示只可写不可读

PM1	0 = 0	RA.0 作为输出脚
	= 1	RA.0 作为输入脚
PM1	1 = 0	RA.1 作为输出脚
	= 1	RA.1 作为输入脚
PM1	2 = 0	RA.2 作为输出脚
	= 1	RA.2 作为输入脚
PM1	3 = 0	RA.3 作为输出脚
	= 1	RA.3 作为输入脚

初始复位时, RA口为输入模式 (PM1 = 1111B).

5.15 端口模式寄存器2 (PM2)

端口模式2寄存器为4位二进制寄存器 (PM2.0 到 PM2.3). PM2决定RB口的输入/输出模式, 由MOV PM2, #1指令控制, 位定义如下:

	3	2	1	0
PM2	w	w	w	w

注: W表示只可写不可读.

PM2	0 = 0	RB.0 作为输出脚
	= 1	RB.0 作为输入脚
PM2	1 = 0	RB.1作为输出脚
	= 1	RB.1作为输入脚
PM2	2 = 0	RB.2作为输出脚
	= 1	RB.2作为输入脚
PM2	3 = 0	RB.3作为输出脚
	= 1	RB.3作为输入脚

初始复位时, RB 口作为输入模式 (PM2 = 1111B)。

5.16 输入口RC, RD

RC口有RC.0到RC.3脚；RD口有RD.0到RD.3。RC口和RD口的每个脚均与一个可选的上拉电阻相联，由端口模式0寄存器（PM0）选定；设置RC口的相应位PEF，HEF，和IEF，RC设定的脚位上的信号改变都将退出HOLD模式或执行中断子程序。端口状态寄存器0（PSR0）记录RC口的状态；PSR0可以以指令MOVA R, PSR0读出，和CLR PSR0指令清除，详细参阅图9和指令表。

RD口只用作输入端口，且不提供中断和HOLD模式的退出。

5.17 端口状态寄存器 (PSR0)

端口状态寄存器0为4位二进制寄存器 (PSR0.0 到 PSR0.3)； PSR0 可以以指令MOVA R, PSR0读出，和CLR PSR0指令清除，位定义如下：

	3	2	1	0
PSR0	R	R	R	R

注: R表示只读

PSR0 0 = 1 RC.0信号改变

PSR0 1 = 1 RC.1信号改变

PSR0 2 = 1 RC.2信号改变

PSR0 3 = 1 RC.3信号改变

5.18 端口允许标志 (PEF)

端口允许标志为四位二进制寄存器 (PEF.0 到 PEF.3).；RC口在退出HOLD模式或执行中断前，需先设置PEF，MOV PEG, #1指令完成对PEF的设置，位定义如下：

	3	2	1	0
PEF	w	w	w	w

注: W表示只可写不可读.

PEF 0 = 1 允许RC.0的信号改变控制退出HOLD模式或中断的执行

=0 禁止RC.0的信号改变控制退出HOLD模式或中断的执行

PEF 1 = 1 允许RC.1的信号改变控制退出HOLD模式或中断的执行

=0 禁止RC.1的信号改变控制退出HOLD模式或中断的执行

PEF 2 = 1 允许RC.2的信号改变控制退出HOLD模式或中断的执行

=0 禁止RC.2的信号改变控制退出HOLD模式或中断的执行

PEF 3 = 1 允许RC.3的信号改变控制退出HOLD模式或中断的执行

=0 禁止RC.3的信号改变控制退出HOLD模式或中断的执行

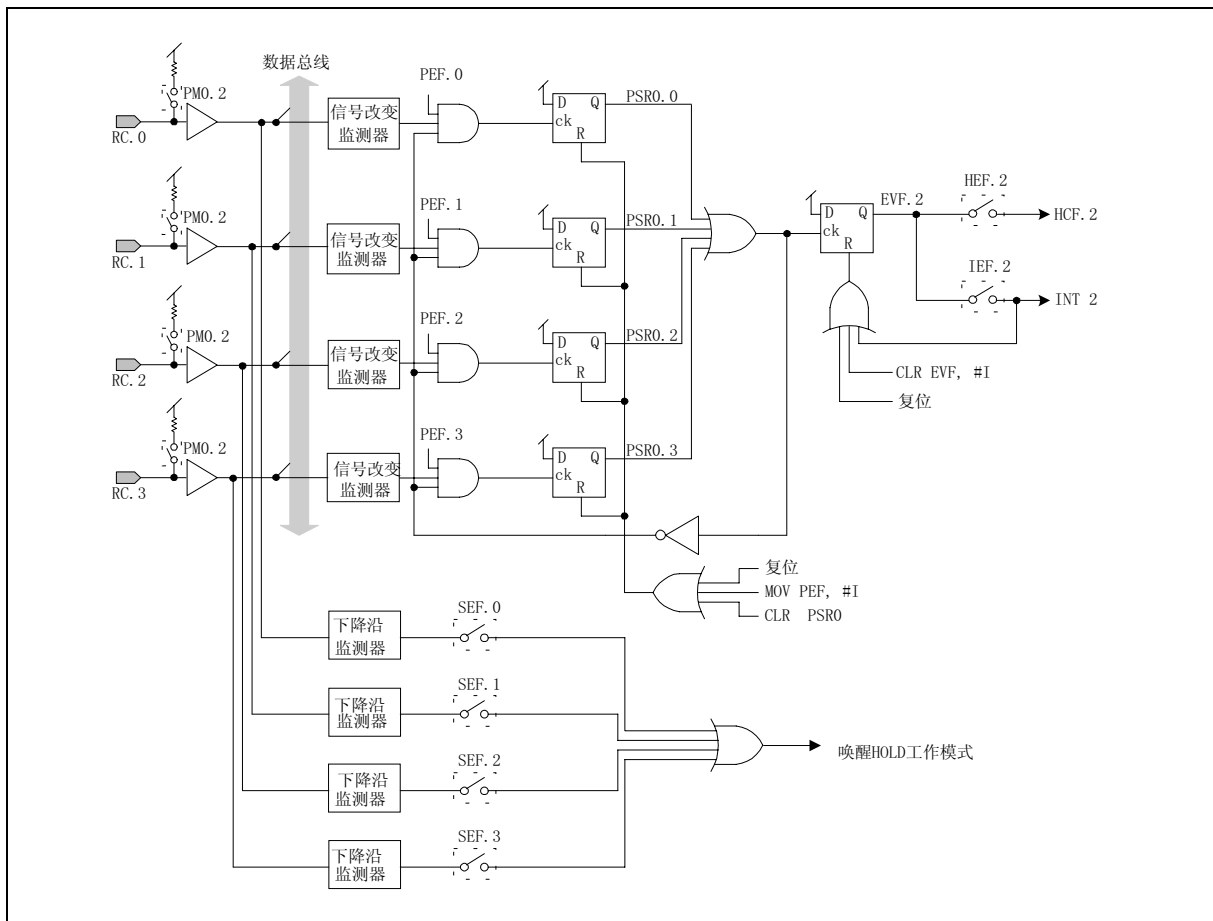


图 9. 输入RC口的结构

5.19 端口RE

执行MOV RE, R指令后, RAM中的数据输出到RE, 它可以提供的较大灌电流直接驱动LED。

5.20 MFP 输出(MFP)

MFP输出脚可以输出定时器1的时钟或调制频率; 脚的输出由模式寄存器1 (MR1) 决定。MR1的结构如图6示。MR1的第2位为0时, MFP输出是在以直流、4096HZ、2048HZ的任意一个为载波, 在载波上调制128HZ, 64HZ, 8HZ, 4HZ, 1HZ (设系统时钟为32.768KHZ) 中的任意一个频率的信号。MOV MFP, #1指令确定调制频率输出合成。寄存器MFP的8位数据说明的含义和MFP的输出脚说明如下:

(Fosc = 32.768 KHz)

W541C250



R7 R6	R 5	R 4	R 3	R 2	R 1	R 0	FUNCTION
0 0	0	0	0	0	0	0	低水平
	0	0	0	0	0	1	128 Hz
	0	0	0	0	1	0	64 Hz
	0	0	0	1	0	0	8 Hz
	0	0	1	0	0	0	4 Hz
	0	1	0	0	0	0	2 Hz
	1	0	0	0	0	0	1 Hz
0 1	0	0	0	0	0	0	高水平
	0	0	0	0	0	1	128 Hz
	0	0	0	0	1	0	64 Hz
	0	0	0	1	0	0	8 Hz
	0	0	1	0	0	0	4 Hz
	0	1	0	0	0	0	2 Hz
	1	0	0	0	0	0	1 Hz
1 0	0	0	0	0	0	0	2048 Hz
	0	0	0	0	0	1	2048 Hz * 128 Hz
	0	0	0	0	1	0	2048 Hz * 64 Hz
	0	0	0	1	0	0	2048 Hz * 8 Hz
	0	0	1	0	0	0	2048 Hz * 4 Hz
	0	1	0	0	0	0	2048 Hz * 2 Hz
	1	0	0	0	0	0	2048 Hz * 1 Hz
1 1	0	0	0	0	0	0	4096 Hz
	0	0	0	0	0	1	4096 Hz * 128 Hz
	0	0	0	0	1	0	4096 Hz * 64 Hz
	0	0	0	1	0	0	4096 Hz * 8 Hz
	0	0	1	0	0	0	4096 Hz * 4 Hz
	0	1	0	0	0	0	4096 Hz * 2 Hz
	1	0	0	0	0	0	4096 Hz * 1 Hz

5.21 中断

W541C250有四个内部中断源（分频器0，分频器1，定时器0，定时器1）和两个外部中断源（RC口和INT）。中断的向量地址存放于程序存储器（ROM）的004H至020H间。IEF，PEF，和EVF控制中断。当EVF被硬件设置为1且其对应的IEF和PEF被软件设置时，将产生中断。产生中断后，在执行EN INT 或MOV IEF, #1指令前，硬件自动禁止所有中断，DIS INT指令可以禁止中断。HOLD模式时产生的中断，将暂时退出HOLD模式，执行中断子程序，执行完中断子程序的RTN指令后，uC重新进入HOLD模式。流程图如图11，控制图如下。

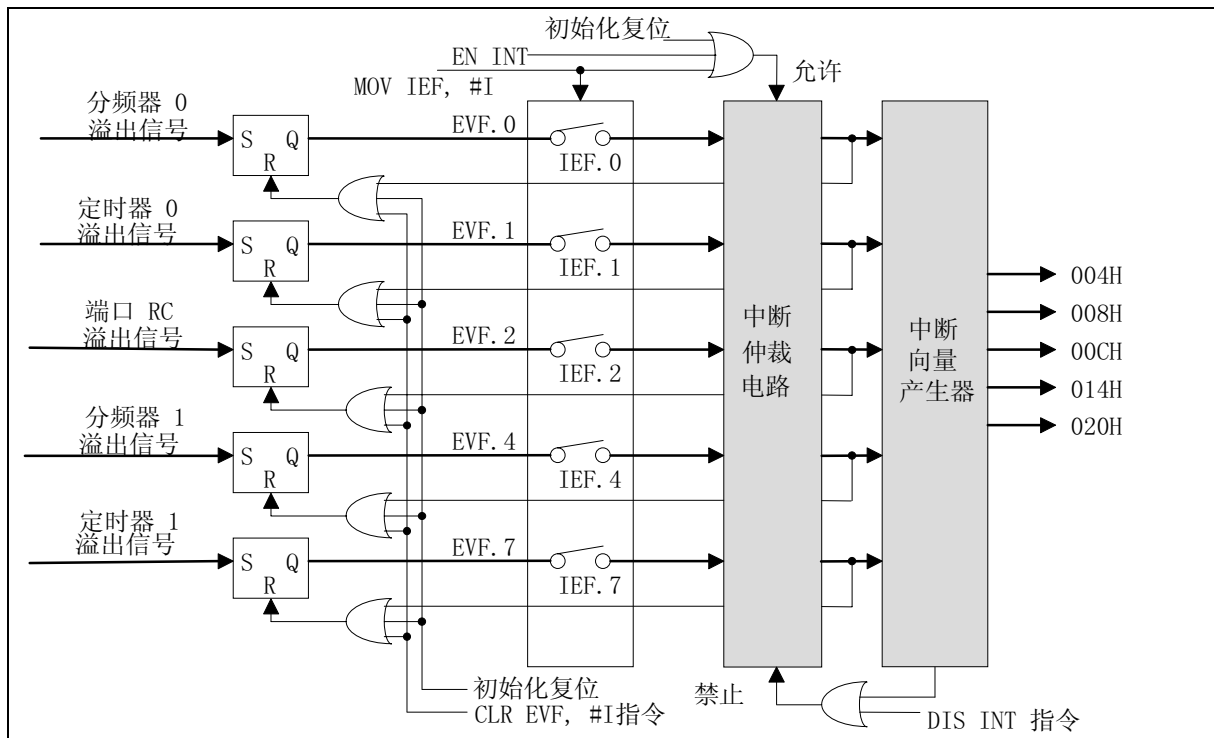


图 10. 中断事件控制标志位

5.22 中断允许标志 (IEF)

中断允许标志为8位二进制寄存器 (IEF.0 到 IEF.7)。这些位控制中断的条件，控制指令为MOV IEF, #1。接收到其中的一个中断时，相应地事件标志位复位，其他位不受影响。中断子程序中，禁止这些中断，直到再次执行任务MOV IEF, #1或EN INT指令。因而，允许这些中断，必须再次执行MOV IEF, #1或EN INT。另外，DIS INT指令将可禁止这些中断。位定义如下：

	7	6	5	4	3	2	1	0
IEF	w	-	-	w	-	w	w	w

注: W 表示只可写

IEF.0 = 1 分频器0溢出产生中断0



- IEF.1 = 1 定时器0溢出产生中断1
 IEF.2 = 1 RC口信号改变产生中断2
 IEF.3 保留
 IEF.4 = 1 接收INT脚的下将沿产生中断4
 IEF.5 IEF.6 保留
 IEF.7 = 1 定时器1溢出产生中断7

5.23 STOP模式

STOP模式时，uC的所有操作均停止。执行STOP指令后，uC进入STOP模式；RC口出现下跳沿的信号，时退出STOP模式。在有效的信号被接收之后，uC退出STOP模式紧跟着执行下面的指令（如果相应的IEF位和PEF位被置位（set），程序将进入中断服务程序）。为防止在退出STOP模式时产生错误，必须在STOP指令之后加上空操作指令（NOP）。

5.24 RC口唤醒STOP的使能标志 (SEF)

RC口唤醒STOP的使能标志是四位的二进制寄存器 (SEF.0 到 SEF.3)。必需对SEF的相应位进行设置之后，才能使RC口唤醒STOP状态。寄存器的内容是通过MOV SEF, #1指令来控制的，位定义如下：

	3	2	1	0
SEF	W	W	W	W

注: W 表示只可写不可读

- SEF 0 = 1 RC.0口信号的下降导致STOP模式的退出使能
 SEF 1 = 1 RC.1口信号的下降导致STOP模式的退出使能
 SEF 2 = 1 RC.2口信号的下降导致STOP模式的退出使能
 SEF 3 = 1 RC.3口信号的下降导致STOP模式的退出使能

5.25 HOLD模式

HOLD模式时，UC的所有操作均停止（晶振，定时器，分频器，和LCD驱动器除外）。执行HOLD指令 (HOLD)后，uC进入HOLD状态。HOLD模式有五退出模式：定时器0，定时器1，分频器0，分频器1，RC口，芯片进入HOLD模式前，HEF，PEF，和IEF标志需设定HOLD模式退出的条件，详细参阅指令设置表和下图。

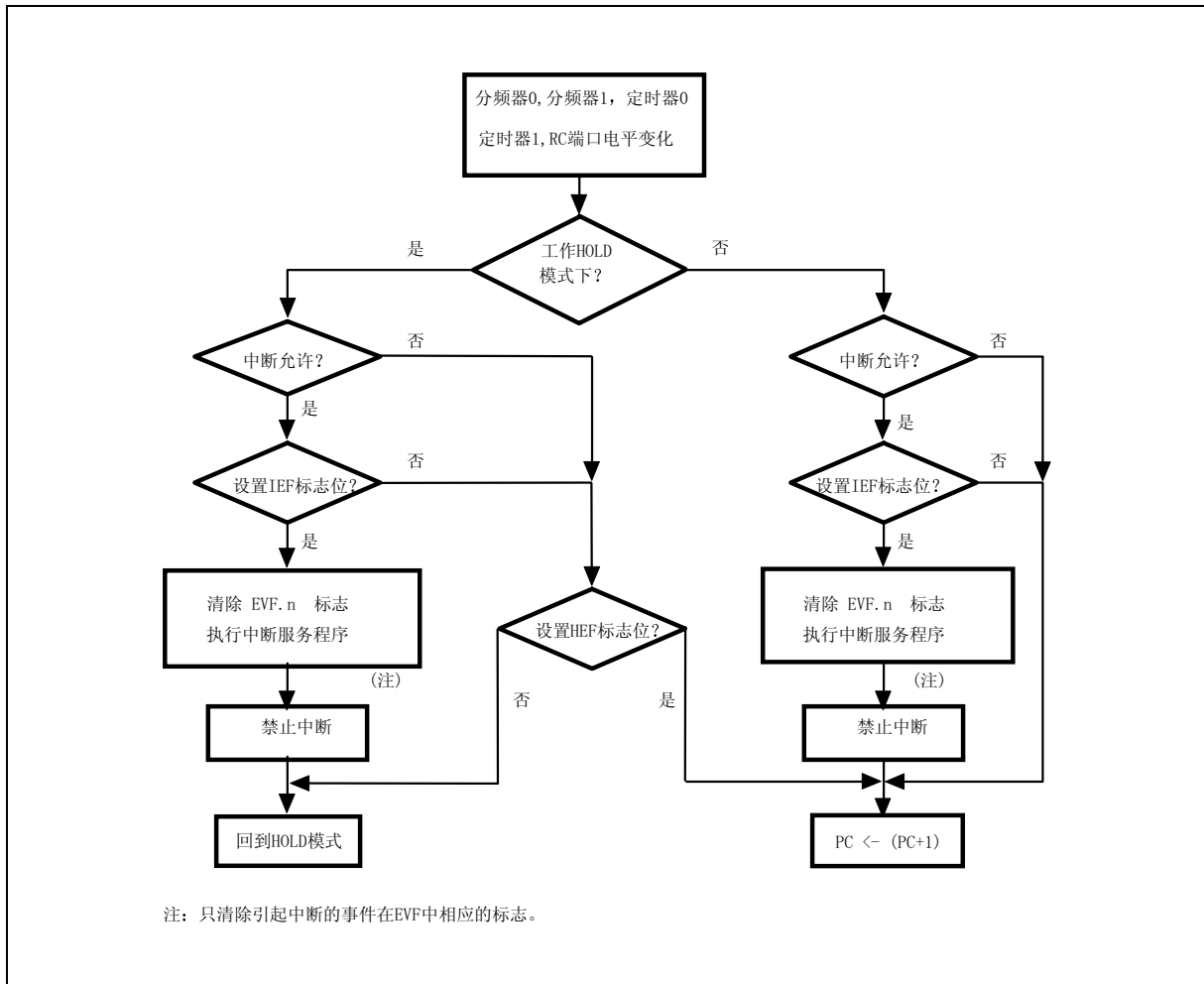


图11. HOLD模式和中断流程图

5.26 HOLD模式退出允许标志 (HEF)

HOLD模式退出允许标志为8位二进制寄存器(HEF.0 to HEF.7)。HEF控制HOLD模式退出条件，MOV HEF, #1指令完成控制。位定义如下：

	7	6	5	4	3	2	1	0
HEF	W	-	-	W	-	W	W	W

注: W 表示只可写不可读

- HEF.0 = 1 分频器0溢出导致HOLD模式退出使能
- HEF.1 = 1 定时器0溢出导致HOLD模式退出使能
- HEF.2 = 1 RC口信号改变导致HOLD模式退出使能



- HEF.3 保留
 HEF.4 = 1 分频器1溢出导致HOLD模式退出使能
 HEF.5 & HEF.6 保留
 HEF.7 = 1 定时器1溢出导致HOLD模式退出使能

5.27 HOLD模式退出的条件标志 (HCF)

HOLD模式退出条件标志为8位二进制寄存器 (HCF0 到 HCF7).指明退出HOLD模式的中断源，由硬件置位。MOV R, HCFL和MOVA R, HCFH可以读出HCF。HCF的任意位为一时，退出HOLD模式，同时，HOLD指令无效，CLR EVF, #1 (EVF. N=0) 指令使HCF复位。EVF和HEF复位后，HCF的相应位同时复位。位定义如下：

	7	6	5	4	3	2	1	0
HCF	—	—	R	R	—	R	R	R

注: R 表示只读

- HCF.0 = 1 分频器0的溢出使得HOLD模式退出。
 HCF.1= 1 定时器0的溢出使得HOLD模式退出。
 HCF.2=2 RC口信号改变使得HOLD模式退出。
 HCF.3 保留。
 HCF.4 = 1 $\overline{\text{INT}}$ 脚的下降沿使得HOLD模式的退出。
 HCF.5 = 1 定时器1的溢出使得HOLD模式的退出。
 HCF.6 & HCF.7 保留

5.28 事件标志 (EVF)

事件标志为8位二进制寄存器，由硬件设置，CLR EVF, #1指令或中断产生将使其复位。位定义如下：

	7	6	5	4	3	2	1	0
EVF	R	—	—	R	—	R	R	R

注: R 表示只读

- EVF.0 = 1 分频器0溢出
 EVF.1 = 1 定时器0溢出
 EVF.2 = 1 RC口信号改变
 EVF.3 保留
 EVF.4 = 1 $\overline{\text{INT}}$ 脚的下降沿



EVF.5 & EVF.6 保留.

EVF.7 = 1 定时器1溢出.

5.29 复位功能

W541C250 可以以上电复位或外部 $\overline{\text{RES}}$ 脚使其复位。复位后，W541C250 的初始状态如下

程序计数器 (PC)	000H
TM0, TM1	复位
MR0, MR1, PM0, PAGE, PMF 寄存器	复位
PM1, PM2 寄存器	设置 (1111B)
PSR0 寄存器	复位
IEF, HEF, PEF, SEF, HCF, EVF 标志	复位
定时器0的输入时钟	Fosc/4
定时器1的输入时钟	Fosc
MFP 输出	低
输入/输出口 RA, RB	输入模式
输出口 RE	高
RA & RB 口输出型式	CMOS 型
RC & RD 口上拉电阻	禁止
看门狗定时器的输入时钟	Fosc/1024
LCD 显示	关闭
段输出模式	LCD 驱动输出

5.30 LCD 控制器/驱动器

W541C250可以直接驱动24段输出脚和4个通用输出脚的LCD，合计24×4点。选择代码可以有5种LCD驱动模式供选择：静态、1/2偏压1/2占空比、1/2偏压1/3占空比、1/3偏压1/3占空比、1/3偏压1/4占空比（参阅图13）。LCD的交流频率可以设置为Fw/64,Fw/128,Fw/256,或Fw/512；另外，在option code中可以设置LCD驱动输出脚（段0至段23）的其中四个作为直流输出口。LCD的交流频率（FLCD）的结构如下图所示：

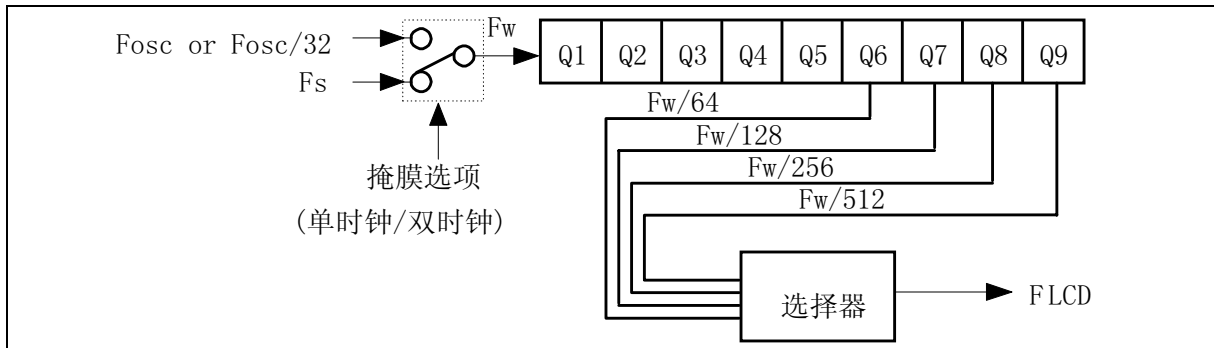


图12. LCD 交流频率 (FLCD) 线路图

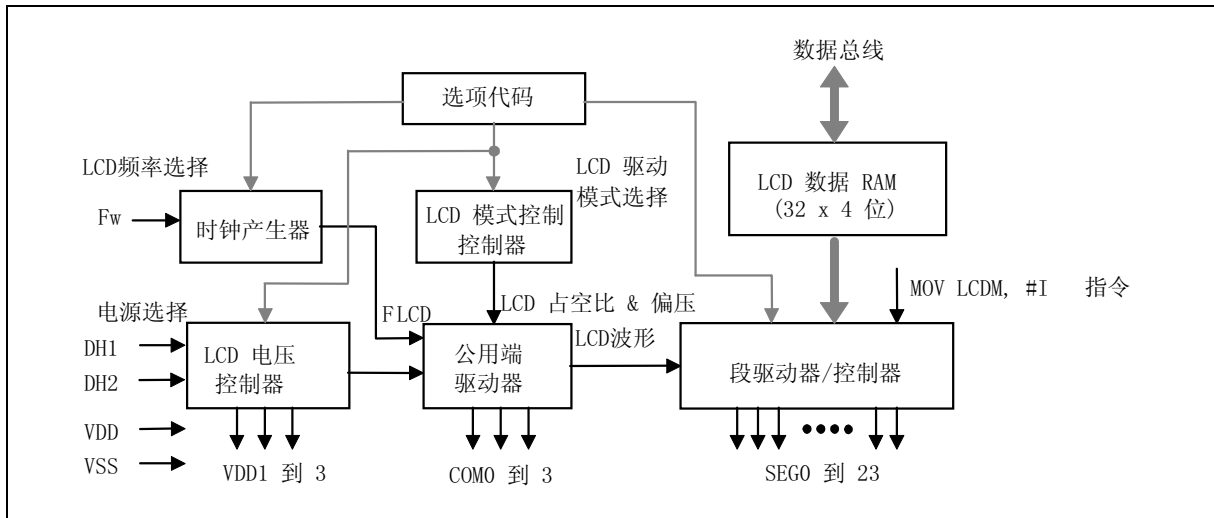


图 13. LCD 驱动器/控制器线路图

Fw = 32.768 KHz时, LCD频率如下表示:

LCD 频率	静态	1/2占空比	1/3 占空比	1/4 占空比
Fw/512 (64 Hz)	64	32	21	16
Fw/256 (128 Hz)	128	64	43	32
Fw/128 (256 Hz)	256	128	85	64
Fw/64 (512 Hz)	512	256	171	128

与24LCD驱动输出脚相对应, 有24个LCD数据RAM段 (LCDR00至LCDR1F), MOV LCDR, #1; MOV WR, LCDR; MOVLCDR, WR; 和MOVLCDR, ACC等指令均可读写LCD数据RAM; LCD数据RAM中的数据由直接存储器存取方式传送到段输出脚, 不需编程控制。LCD数据RAM的位置为1时, 打开LCD, 位置为0, 则关闭。通过存储器直接将LCD数据RAM的内容 (LCDR) 映射到段0至段23的输出口, LCD数据

RAM和段/通用脚之间的关系如下所示:

		COM3	COM2	COM1	COM0
LCD 数 RAM	输出脚	位3	位2	位1	位0
LCDR00	SEG0	0/1	0/1	0/1	0/1
LCDR01	SEG1	0/1	0/1	0/1	0/1
.
.
LCDR16	SEG22	0/1	0/1	0/1	0/1
LCDR17	SEG23	0/1	0/1	0/1	0/1

LCDON指令打开LCD显示(即使在HOLD模式), LCDOFF指令关闭LCD显示。初始复位时, LCD各段全亮。初始复位结束后, LCD显示自动关闭.显示LCD必须执行LCDON指令。当LCD驱动输出脚作为直流输出时(由选择代码选择), 可以执行MOV LCDM, #1指令选择CMOS输出模式或NMOS输出模式。LCD数据RAM和段/常用脚之间的关系如下表示。直接存储器存取可以将LCDR00中的数据传送到相应的段输出口(SEG3和SEG0), 其他的LCD数据处RAM段可以作为常用字数据RAM来存取数据。

LCD 数据 RAM	输出脚	位 3	位 2	位 1	位 0
LCDR00	SEG3-SEG0	SEG3	SEG2	SEG1	SEG0
LCDR03-LCDR01	-	-	-	-	-
LCDR04	SEG7-SEG4	SEG7	SEG6	SEG5	SEG4
LCDR07-LCDR05	-	-	-	-	-
.
.
LCDR17	SEG23-SEG20	SEG23	SEG22	SEG21	SEG20

LCD驱动模式和可获得者的LCD段之间的关系如下示:

LCD 驱动模式	可获得的LCD段最大数量	相联接的电压=3.0V
静态	24 (COM1)	VDD3, VDD2 与VDD1相联
1/2 偏压 1/2 占空比	48 (COM1-COM2)	VDD3 与VDD2相联
1/2 偏压 1/3 占空比	48 (COM1-COM3)	VDD3 与 VDD2相联系
1/3 偏压 1/3 占空比	72 (COM1-COM3)	-
1/3 偏压 1/4 占空比	96 (COM1-COM4)	-

LCD 输出模式型标志(LCDM)



LCD 输出模式形式为8位二进制寄存器。这些位用于控制LCD输出的脚位结构；当LCD输出脚由选择代码设置为直流输出模式时，这些输出脚的结构（段0至段23）可以选择CMOS或NMOS形式，由MOV LCDM，#1指令控制，位定义如下：

	7	6	5	4	3	2	1	0
LCDM	-	-	w	w	w	w	w	w

注: W表示只可写不可读.

LCDM.0 = 0 SEG0 至SEG3 为CMOS输出型模式

= 1 SEG0 至SEG3 为NMOS 输出模式

LCDM.1 = 0 SEG4 至SEG7 为CMOS 输出模式

= 1 SEG4 至SEG7 为NMOS 输出模式

LCDM.2 = 0 SEG8 至 SEG11 为CMOS 输出模式

= 1 SEG8 至 SEG11为 NMOS 输出模式

LCDM.3 = 0 SEG12至 SEG15 为CMOS 输出模式

= 1 SEG12至 SEG15 为NMOS 输出模式

LCDM.4 = 0 SEG16 至 SEG19 为CMOS 输出模式.

= 1 SEG16 至 SEG19 为NMOS 输出模式

LCDM.5 = 0 SEG20 至 SEG23 为 CMOS输出模式.

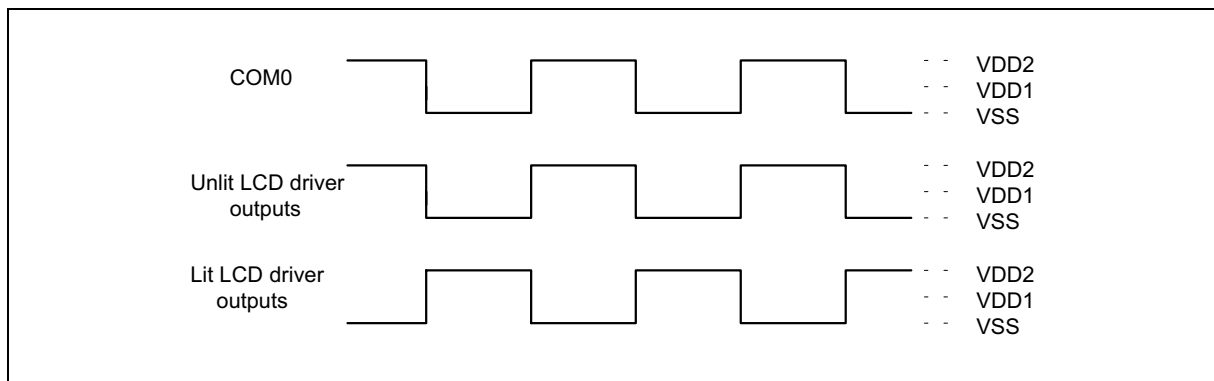
= 1 SEG20 至 SEG23 为NMOS 输出模式

LCDM.6, LCDM.7 保留

LCD驱动模式的输出波形如下：

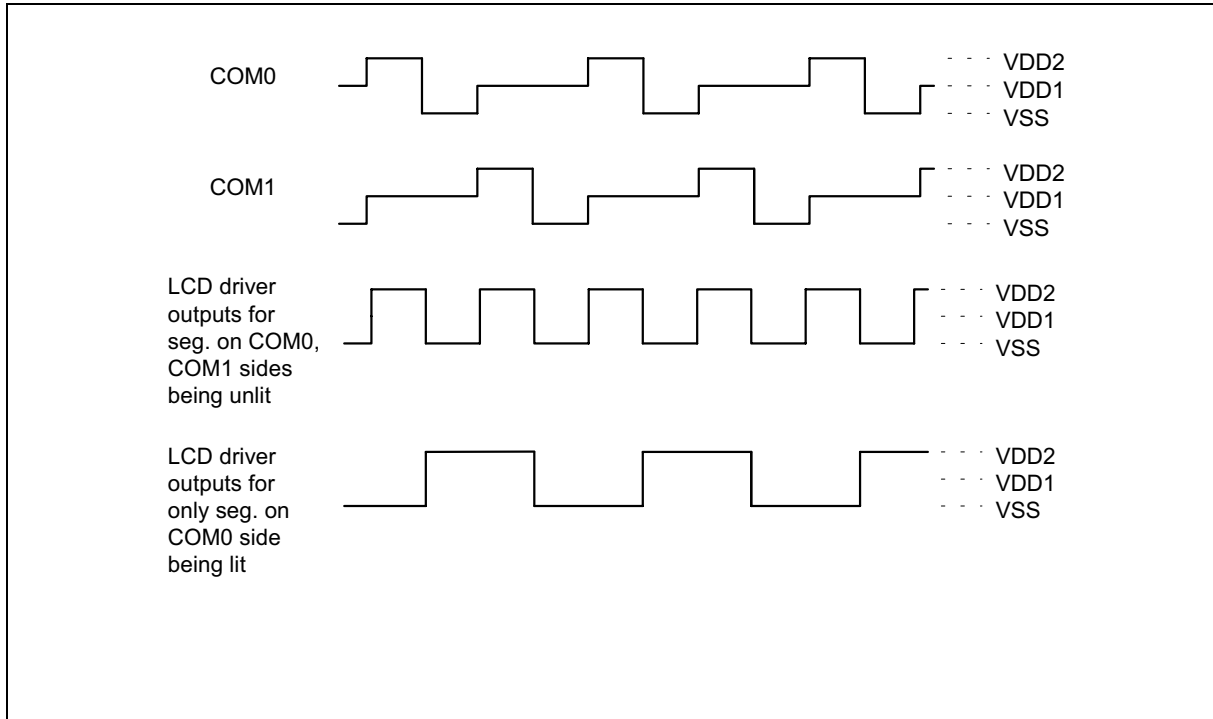
静态点亮系统 (举例)

正常工作模式



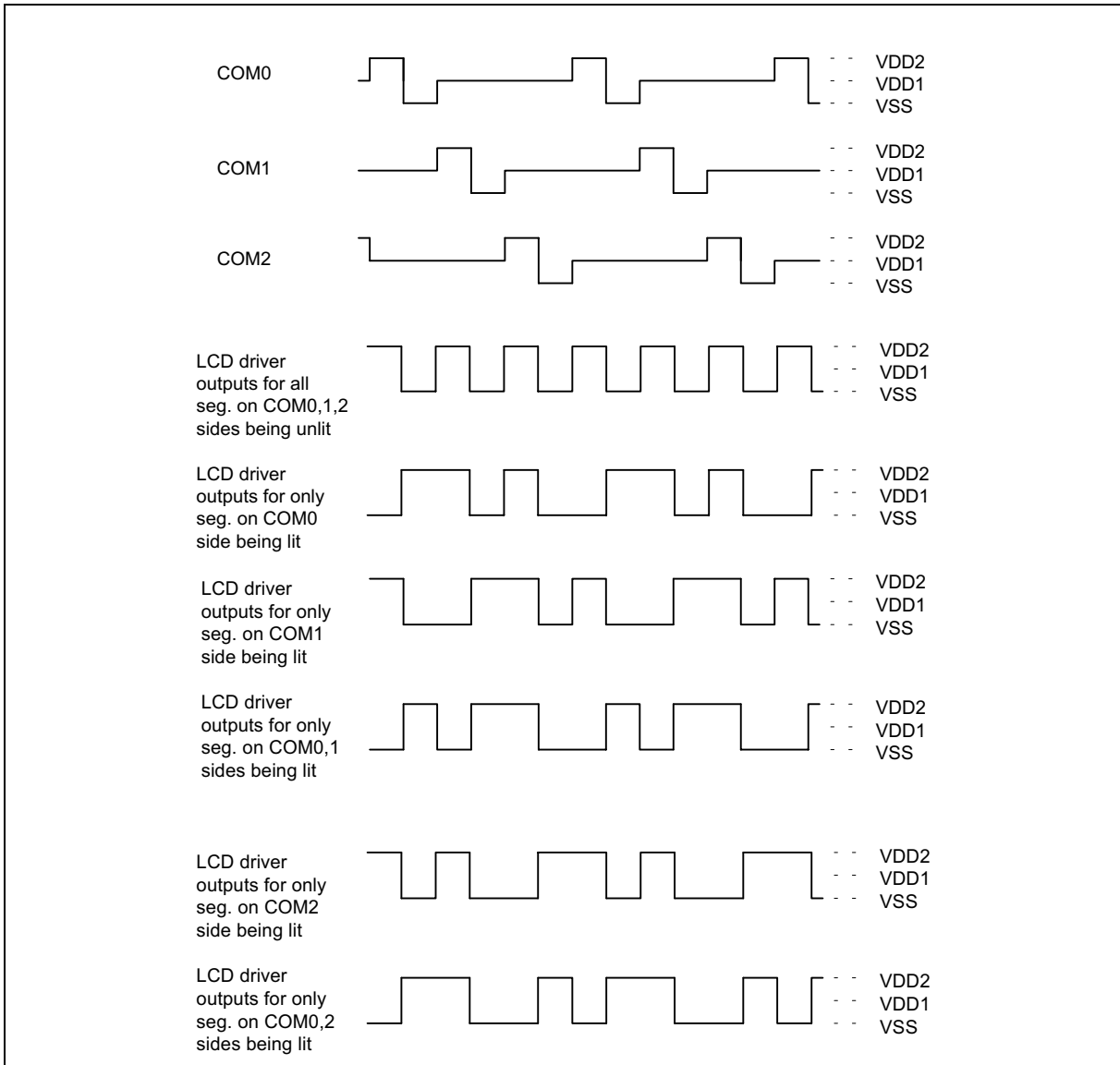
1/2 偏压 1/2 占空比点亮系统 (举例)

正常工作模式



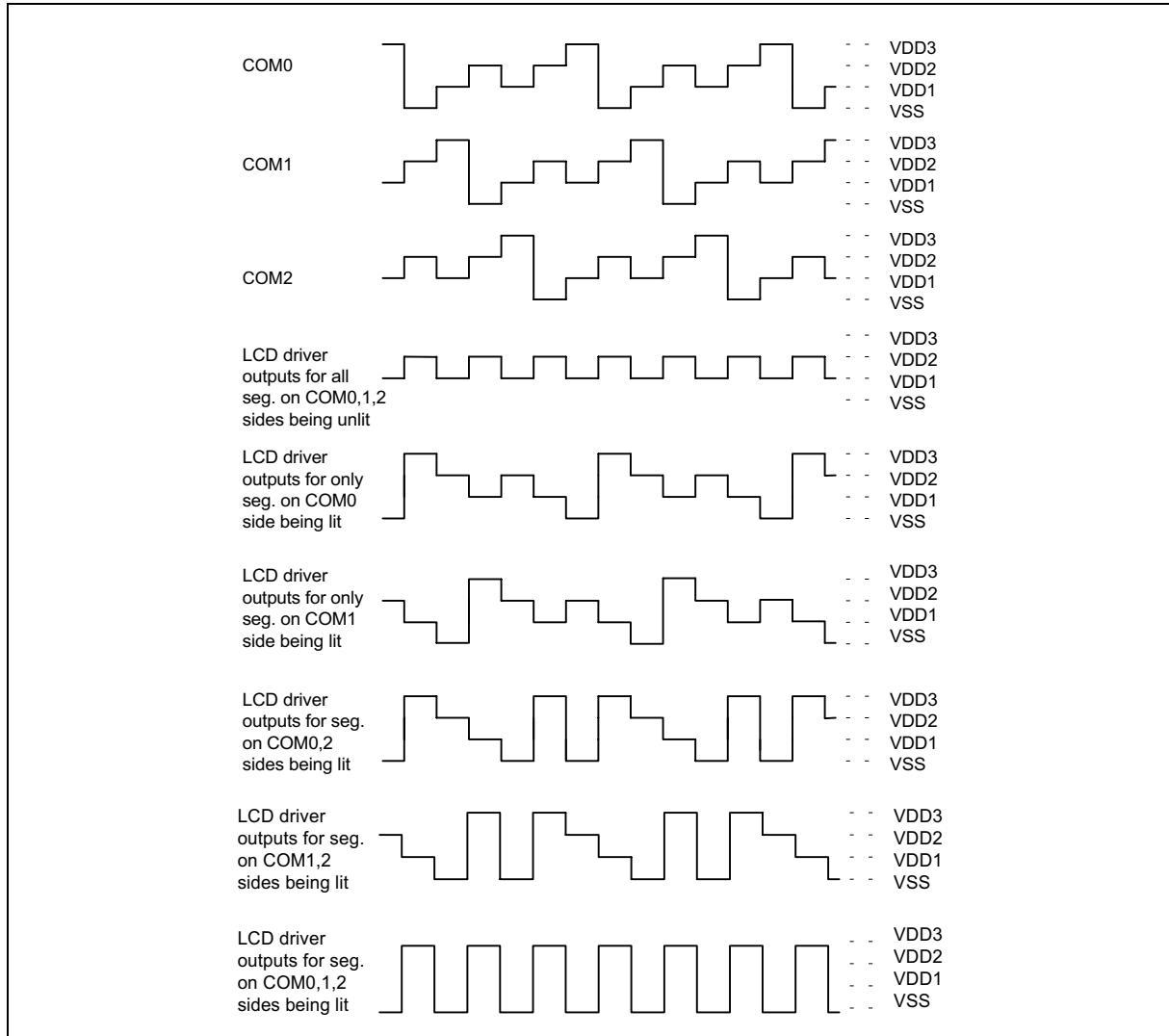
1/2 偏压 1/3 占空比点亮系统(举例)

正常工作模式



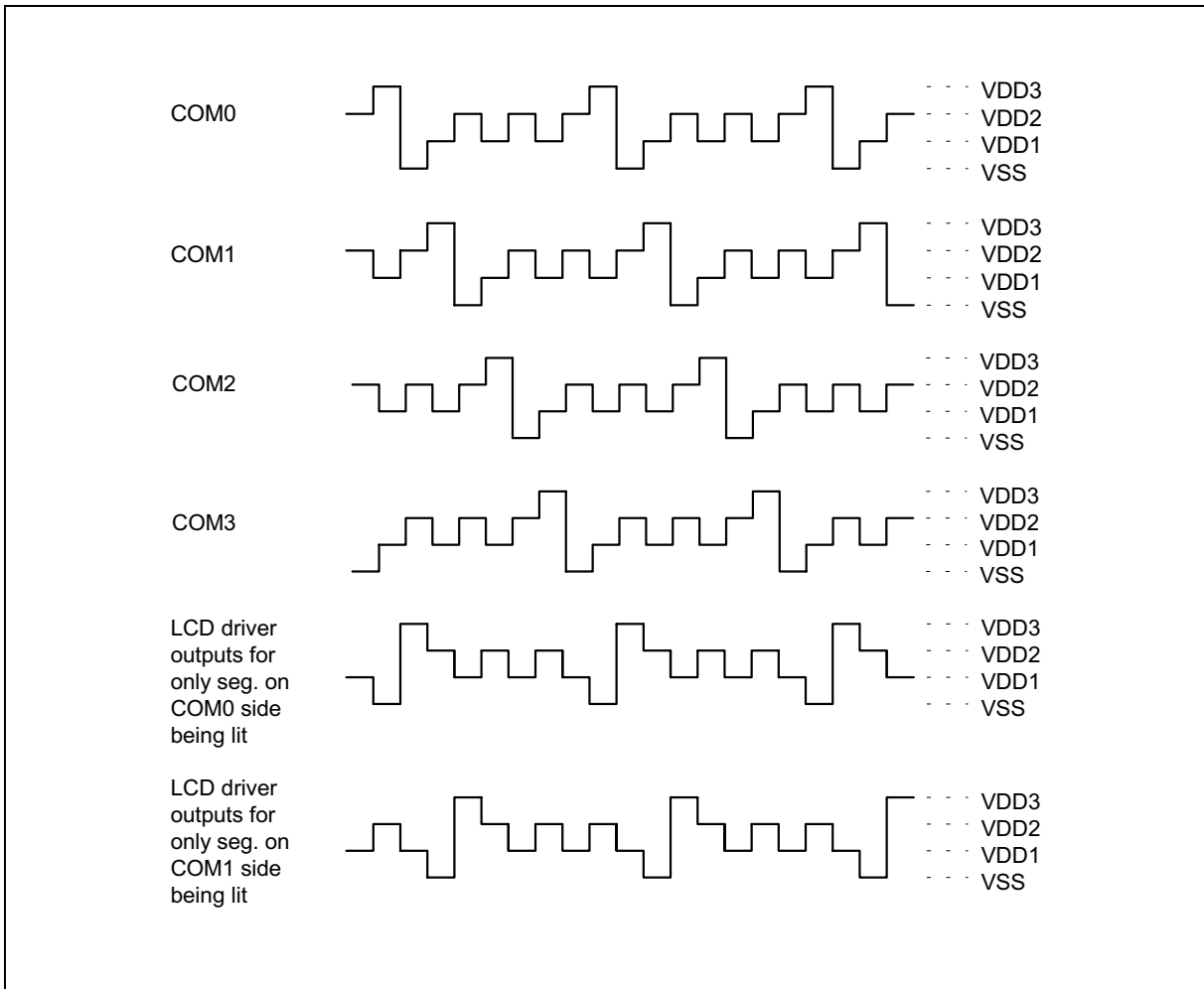
1/3 偏压 1/3 占空比点亮(举例)

正常工作模式

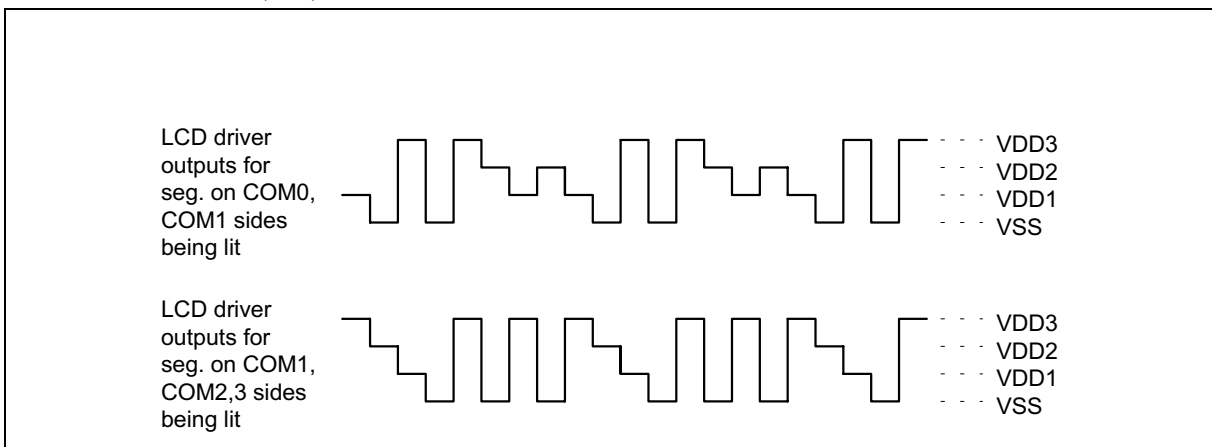


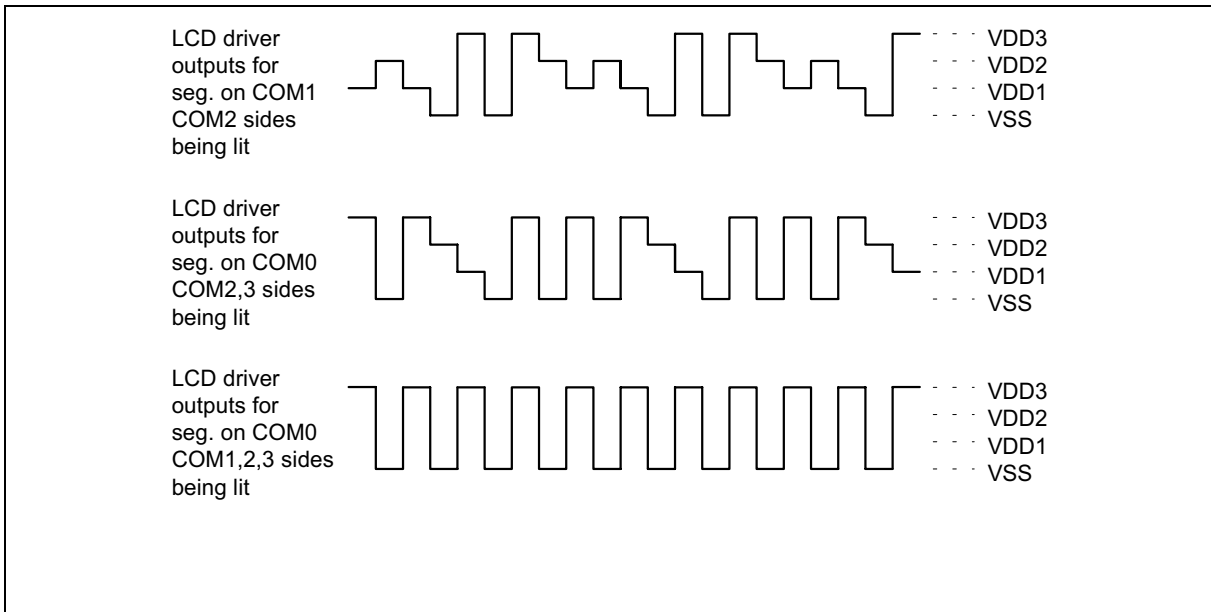
1/3 偏压 1/4 占空比点亮模式 (举例)

正常工作模式

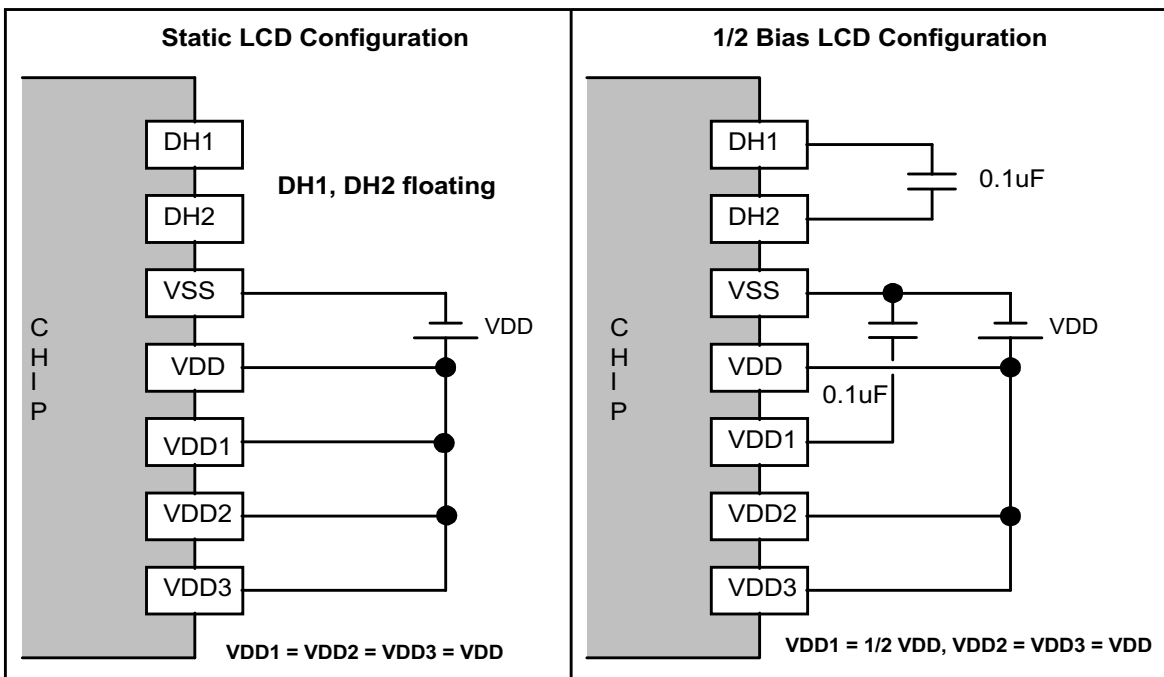


1/3 偏压 1/4 占空比点亮 (举例)- 连续的正常操作模式

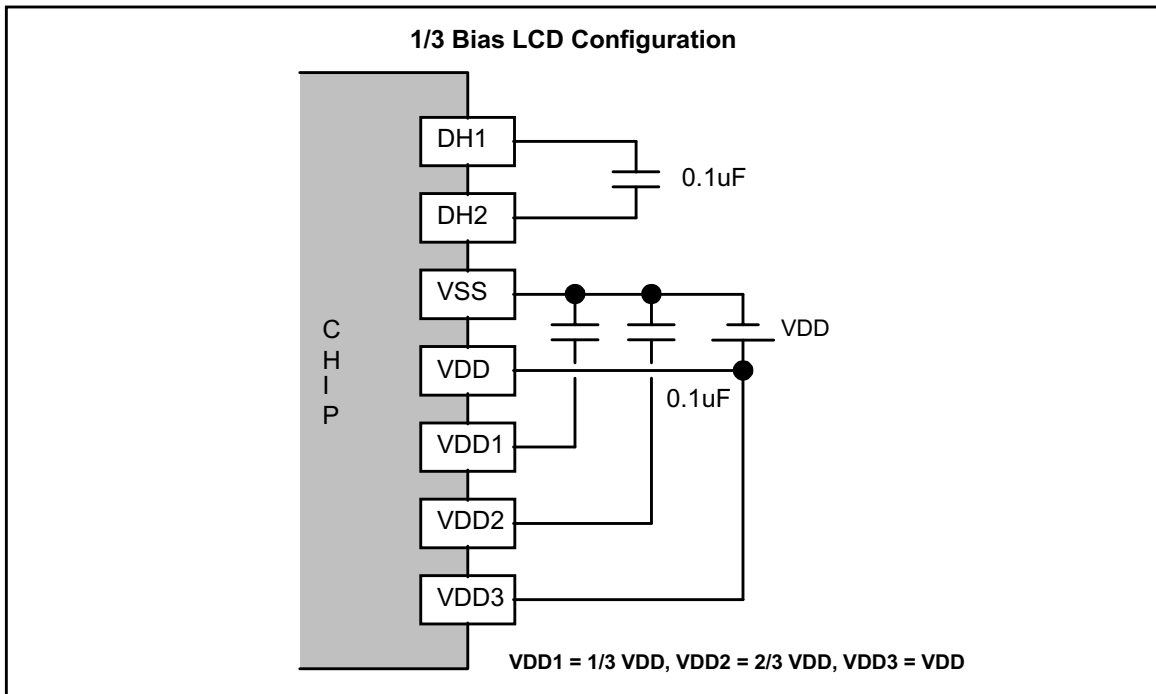
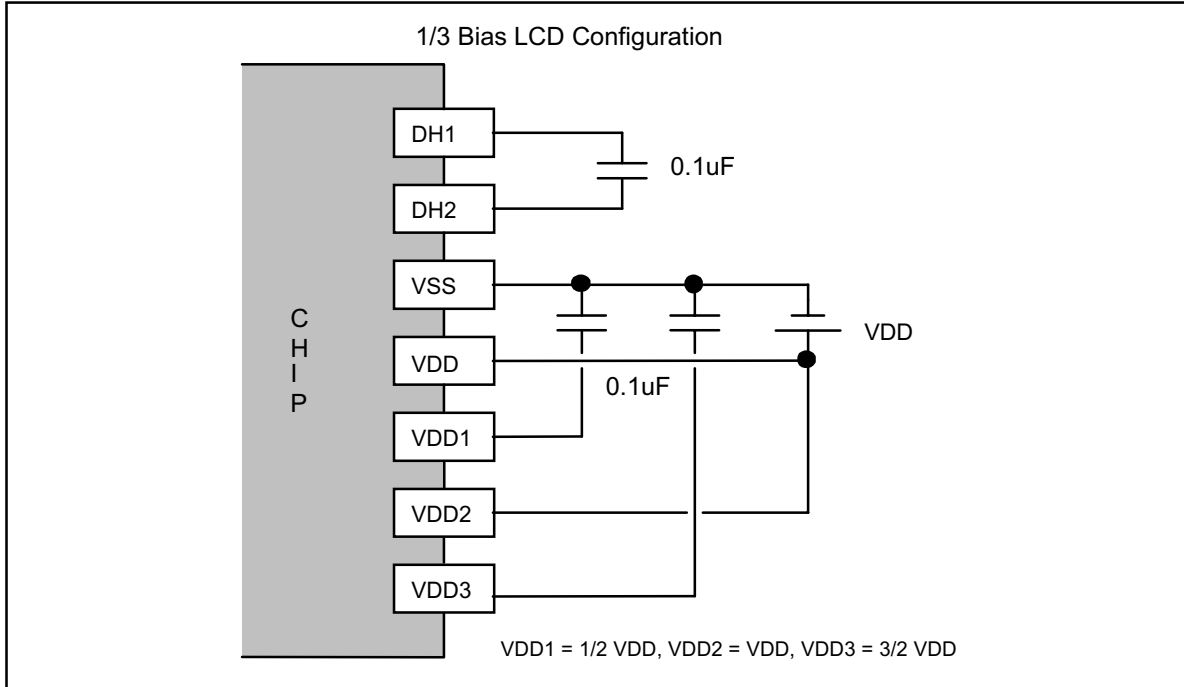




由掩膜模式决定的不同LCD驱动模式下电源连接模式，显示如下：



LCD 配置, 续



6. 最大极限值的范围

参数	范围	单位
电压输入值	-0.3 to +7.0	V
实际的输入/输出电压	-0.3 to +7.0	V
耗电功率	120	mW
工作环境的温度	0 to +70	°C
温度极限值	-55 to +150	°C

注:表中列举的极限值是在有可能对器件的可靠性或者对器件出现损坏的条件下得出来的。

7. 直流特性

(V_{DD}-V_{SS} = 3.0V, F_m = 4.19 MHz, F_s = 32.768 KHz, T_a = 25° C, LCD on; unless otherwise specified)

参数	符号	条件	最小值	典型值	最大值	单位
工作电压	V _{DD}	-	2.4	-	5.5	V
工作电流 (晶体型式)	I _{OP1}	无负载 (Ext-V)	-	8	20	μA
工作电流 (RC 型式)	I _{OP2}	无负载 (Ext-V)	-	35	65	μA
HOLD电流 (晶体型式)	I _{HM1}	HOLD模式无负载 (Ext-V)	-	4	6	μA
HOLD电流 (RC 型式)	I _{HM2}	HOLD模式无负载, (Ext-V)	-	16	40	μA
STOP电流 (晶体型式)	I _{SM1}	STOP模式无负载 (Ext-V)	-	0.1	2	μA
STOP电流 (RC 型式)	I _{SM2}	STOP模式无负载 (Ext-V)	-	0.1	2	μA
输入低电压	V _{IL}	-	V _{SS}	-	0.3 V _{DD}	V

DC 特性,续

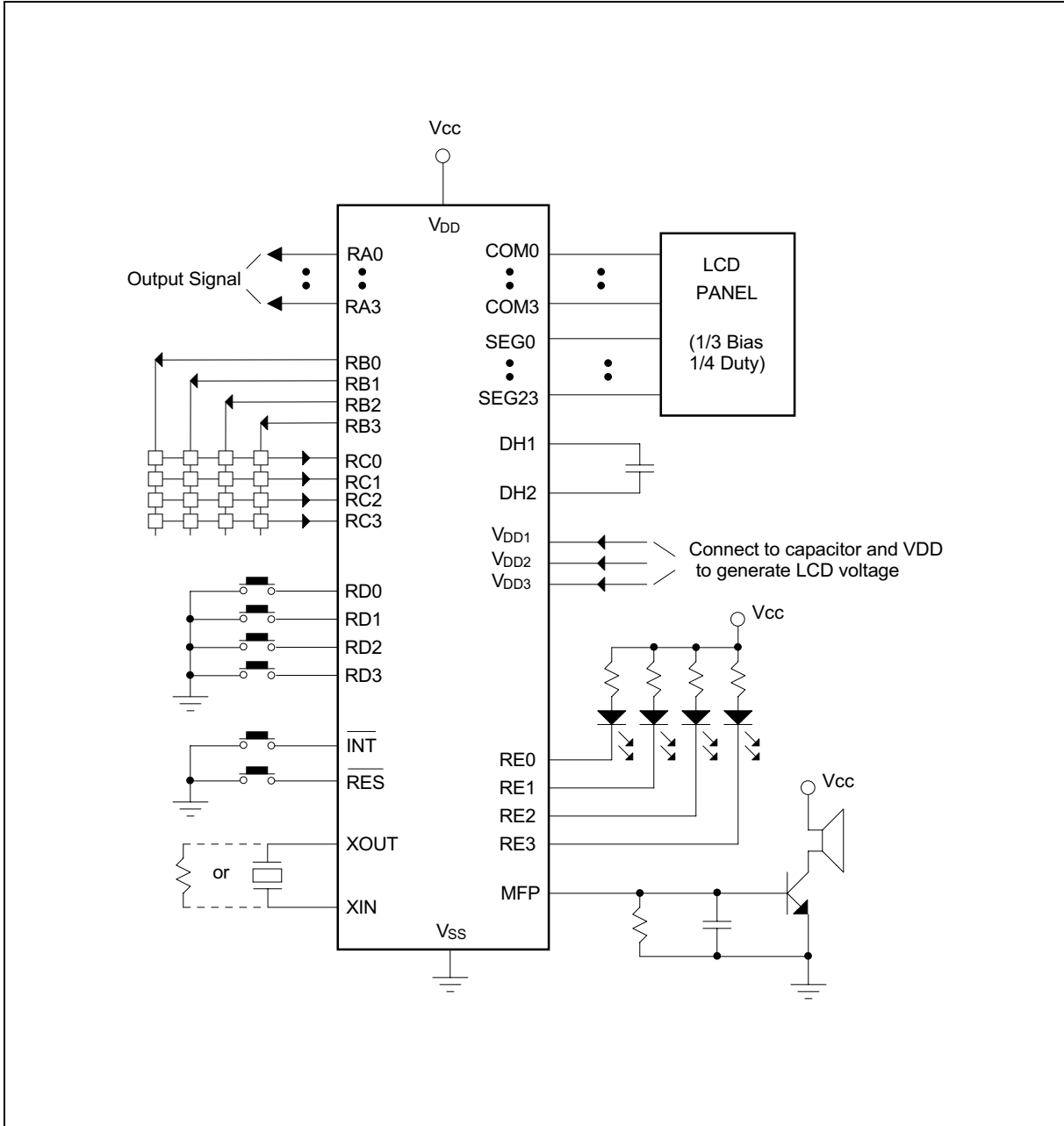
参数	符号	条件	最小值	典型值	最大值	单位
输入高电压	V _{IH}	-	0.7 V _{DD}	-	V _{DD}	V
MFP 输出低电压	V _{ML}	I _{OL} = 3.5 mA	-	-	0.4	V
MFP 输出高电压	V _{MH}	I _{OH} = 3.5 mA	2.4	-	-	V
RA, RB 口输出低电压	V _{ABL}	I _{OL} = 2.0 mA	-	-	0.4	V
RA, RB 输出高电压	V _{ABH}	I _{OH} = 2.0 mA	2.4	-	-	V
LCD 提供的电流	I _{LCD}	All Seg. ON	-	-	10	μA
SEG0-SEG23 吸收电流(作为LCD输出脚)	I _{OL1}	V _{OL} = 0.4V V _{LCD} = 0.0V	0.4	-	-	μA
SEG0-SEG23 驱动电流 (作为 LCD 输出脚)	I _{OH1}	V _{OH} = 2.4V V _{LCD} = 3.0V	0.3	-	-	μA
Segment 输出低电压 (作为直流输出脚)	V _{SL}	I _{OL} = 0.6 mA	-	-	0.4	V
Segment 输出高电压 (作为直流输出脚)	V _{SH}	I _{OH} = 3 μA	2.4	-	-	V
RE 口吸收电流	I _{EL}	V _{OL} = 0.9V	9	-	-	mA
RE 源电流	I _{EH}	V _{OH} = 2.4V	0.4	1.2	-	mA
输入端口上拉电阻	R _{CD}	Port RC, RD	100	350	1000	KΩ
中断上拉电阻	R _{INT}	-	50	250	1000	KΩ
RES上拉电阻	R _{RES}	-	20	100	500	KΩ

8. 交流特性

(V_{DD}-V_{SS} = 3.0V, T_A = 25° C,另有说明除外)

参数	符号	条件	最小值	典型值	最大值	单位
工作频率	FOSC	RC 型	-	-	4000	KHz
		晶体型1 (选择低速度型式)	-	32.768	-	
		晶体型2 (选择高速度型式)	400	-	4190	
RC晶振的电压下降产生的频率偏差	$\frac{\Delta f}{f}$	$\frac{f(3V) - f(2.4V)}{f(3V)}$	-	-	10	%
指令周期	T _I	一个机器周期	-	4/FOSC	-	S
复位有效宽度	T _{RAW}	FOSC = 32.768 KHz	1	-	-	μS
中断有效宽度	T _{IAW}	FOSC = 32.768 KHz	1	-	-	μS

9. 典型的电路图



10. 指令表

符号的描述

ACC:	累加器
ACC.n:	累加器的第N位
WR:	工作寄存器
PAGE:	页寄存器
MR0:	模式寄存器0
MR1:	模式寄存器 1
PM0:	端口寄存器 0
PM1:	端口寄存器 1
PM2:	端口寄存器2
PSR0:	端口状态寄存器 0
R:	存储器 (RAM) 的地址
LCDR:	LCD 数据存储器的地址
R.n:	寻址RAM中的第N位
I:	常量
L:	转移的地址
CF:	进位标志
ZF:	零位标志
PC:	程序计数器
TM0:	定时器0
TM1:	定时器1
IEF.n:	中断标志的第N位
HCF.n:	退出HOLD模式的条件标志位
HEF.n:	HOLD模式的第N位
PEF.n:	端口使能标志的第N位
EVF.n:	状态标志的第N位
! =:	不等于



&: 与
^: 或
EX: 异或
←: 转换方向; 结果

[PAGE*10H+(): PAGE(位2, 位1, 位0)*10H+()地址的内容

[P(): 端口P()的内容

指令表1

助记符		功能	影响标志位	周期
算术类				
ADD	R, ACC	$ACC \leftarrow (R) + (ACC)$	ZF, CF	1
ADD	WR, #I	$ACC \leftarrow (WR) + I$	ZF, CF	1
ADDR	R, ACC	$ACC, R \leftarrow (R) + (ACC)$	ZF, CF	1
ADDR	WR, #I	$ACC, WR \leftarrow (WR) + I$	ZF, CF	1
ADC	R, ACC	$ACC \leftarrow (R) + (ACC) + (CF)$	ZF, CF	1
ADC	WR, #I	$ACC \leftarrow (WR) + I + (CF)$	ZF, CF	1
ADCR	R, ACC	$ACC, R \leftarrow (R) + (ACC) + (CF)$	ZF, CF	1
ADCR	WR, #I	$ACC, WR \leftarrow (WR) + I + (CF)$	ZF, CF	1
ADU	R, ACC	$ACC \leftarrow (R) + (ACC)$	ZF	1
ADU	WR, #I	$ACC \leftarrow (WR) + I$	ZF	1
ADUR	R, ACC	$ACC, R \leftarrow (R) + (ACC)$	ZF	1
ADUR	WR, #I	$ACC, W R \leftarrow (WR) + I$	ZF	1
SUB	R, ACC	$ACC \leftarrow (R) - (ACC)$	ZF, CF	1
SUB	WR, #I	$ACC \leftarrow (WR) - I$	ZF, CF	1
SUBR	R, ACC	$ACC, R \leftarrow (R) - (ACC)$	ZF, CF	1
SUBR	WR, #I	$ACC, WR \leftarrow (WR) - I$	ZF, CF	1
SBC	R, ACC	$ACC \leftarrow (R) - (ACC) - (CF)$	ZF, CF	1
SBC	WR, #I	$ACC \leftarrow (WR) - I - (CF)$	ZF, CF	1
SBCR	R, ACC	$ACC, R \leftarrow (R) - (ACC) - (CF)$	ZF, CF	1
SBCR	WR, #I	$ACC, WR \leftarrow (WR) - I - (CF)$	ZF, CF	1
INC	R	$ACC, R \leftarrow (R) + 1$	ZF, CF	1
DEC	R	$ACC, R \leftarrow (R) - 1$	ZF, CF	1

指令表1, 续

助记符	功能	影响标志位	周期
逻辑指令			
ANL R, ACC	ACC←(R) & (ACC)	ZF	1
ANL WR, #I	ACC←(WR) & I	ZF	1
ANLR R, ACC	ACC, R←(R) & (ACC)	ZF	1
ANLR WR, #I	ACC, WR←(WR) & I	ZF	1
ORL R, ACC	ACC←(R) ^ (ACC)	ZF	1
ORL WR, #I	ACC←(WR) ^ I	ZF	1
ORLR R, ACC	ACC, R←(R) ^ (ACC)	ZF	1
ORLR WR, #I	ACC, WR←(WR) ^ I	ZF	1
XRL R, ACC	ACC←(R) EX (ACC)	ZF	1
XRL WR, #I	ACC←(WR) EX I	ZF	1
XRLR R, ACC	ACC, R←(R) EX (ACC)	ZF	1
XRLR WR, #I	ACC, WR←(WR) EX I	ZF	1
跳转指令			
JMP L	PC10-PC0←L10-L0		1
JB0 L	PC10-PC0←L10-L0; if ACC.0 = "1"		1
JB1 L	PC10-PC0←L10-L0; if ACC.1 = "1"		1
JB2 L	PC10-PC0←L10-L0; if ACC.2 = "1"		1
JB3 L	PC10-PC0←L10-L0; if ACC.3 = "1"		1
JZ L	PC10-PC0←L10-L0; if ACC = 0		1
JNZ L	PC10-PC0←L10-L0; if ACC != 0		1
JC L	PC10-PC0←L10-L0; if CF = "1"		1
JNC L	PC10-PC0←L10-L0; if CF != "1"		1
DSKZ R	ACC, R←(R) - 1; 若ACC = 0则跳转	ZF, CF	1
DSKNZ R	ACC, R←(R) - 1; 若 ACC != 0则跳转	ZF, CF	1
SKB0 R	若R.0 = "1"则跳转		1
SKB1 R	若 R.1 = "1"则跳转		1
SKB2 R	若R.2 = "1"则跳转		1
SKB3 R	若R.3 = "1"则跳转		1

指令表 1, 续

助记符	功能	影响标志位	周期
数据传送			
MOV WR, R	WR←(R)		1
MOV R, WR	R←(WR)		1
MOVA WR, R	ACC, WR←(R)	ZF	1
MOVA R, WR	ACC, R←(WR)	ZF	1
MOV R, ACC	R←(ACC)		1
MOV ACC, R	ACC←(R)	ZF	1
MOV R, #I	R←I		1
MOV WR, @R	WR←[PR (bit2, bit1, bit0) × 10H + (R)]		2
MOV @R, WR	[PR (bit2, bit1, bit0) × 10H + (R)]←WR		2
MOV TABH, R	TAB High addresss ← R		1
MOV TABL, R	TAB Low addresss ← R		1
MOVC R	R←[TAB × 10H + (ACC)]		2
MOVC WR, #I	WR ← [(I6 ~ I0) × 10H + (ACC)]		2
输入 & 输出			
MOVA R, RA	ACC, R←[RA]	ZF	1
MOVA R, RB	ACC, R←[RB]	ZF	1
MOVA R, RC	ACC, R←[RC]	ZF	1
MOVA R, RD	ACC, R←[RD]	ZF	1
MOV RA, R	[RA]←(R)		1
MOV RB, R	[RB]←(R)		1
MOV RE, R	[RE]←(R)		1
MOV MFP, #I	[MFP]← I		1
标志 & 寄存器			
MOVA R, PAGE	ACC, R←PAGE (Page Register)	ZF	1
MOV PAGE, R	PAGE←(R)		1
MOV PAGE, #I	PAGE←I		1
MOV MR0, #I	MR0←I		1
MOV MR1, #I	MR1←I		1

指令表1, 续

助记符	功能	影响的标志位	周期
MOVA R, CF	ACC.0, R.0←CF	ZF	1
MOV CF, R	CF←(R.0)	CF	1
MOVA R, HCF0	ACC, R←HCF0–HCF3	ZF	1
MOVA R, HCF4	ACC, R←HCF4–HCF7	ZF	1
CLR PMF, #I	若 In = 1, 则清除参数标志		1
SET PMF, #I	若 In = 1, 则设置参数标志		1
MOV PM0, #I	口模式 0← I		1
MOV PM1, #I	口模式 1← I		1
MOV PM2, #I	口模式2← I		1
CLR EVF, #I	若 In = 1, 则清除事件标志		1
MOV PEF, #I	设置/重置口允许标志		1
MOV IEF, #I	设置/重置中断允许标志		1
MOV HEF, #I	设置/重置退出HOLD模式允许标志		1
MOV SEF, #I	设置/重置STOP模式的唤醒标允许标志		1
MOVA R, PSR0	ACC, R←口状态 AA0		1
CLR PSR0	清除端口状态寄存器		1
SET CF	设置进位标志	CF	1
CLR CF	清除进位标志	CF	1
CLR DIVR0	清除分频器的后四位		1
CLR WDT	清除看门狗定时器		1
移位指令			
SHRC R	ACC.n, R.n←(R.n+1); ACC.3, R.3←0; CF←R.0	ZF, CF	1
RRC R	ACC.n, R.n←(R.n+1); ACC.3, R.3←CF; CF←R.0	ZF, CF	1
SHLC R	ACC.n, R.n←(R.n-1); ACC.0, R.0←0; CF←R.3	ZF, CF	1
RLC R	ACC.n, R.n←(R.n-1); ACC.0, R.0←CF; CF←R.3	ZF, CF	1

指令表 1, 续

助记符	功能	影响标志位	周期
LCD			
MOV LCDR, #I	LCDR← I		1
MOV WR, LCDR	WR←(LCDR)		1
MOV LCDR, WR	LCDR←(WR)		1
MOV LCDR, ACC	LCDR←(ACC)		1
MOV LCDM, #I	选择 LCD 输出模式		1
LCDON	打开LCD		1
LCDOFF	关闭LCD		1
定时器			
MOV TM0H, R	定时器0高AA ← R		1
MOV TM0L, R	定时器0低AAL ← R		1
MOV TM0, #I	设置定时器0		1
MOV TM1H, R	定时器1 高AA ← R		1
MOV TM1L, R	定时器1 低AA← R		1
MOV TM1, #I	设置定时器1		1
调子程序			
CALL L	STACK ← (PC)+1; PC10-PC0 ← L10-L0		1
RTN	(PC)←STACK		1
其他			
HOLD	进入HOLD模式		1
STOP	进入STOP模式		1
NOP	无操作		1
EN INT	允许中断		1
DIS INT	禁止中断		1

指令表2

ADC R, ACC	将R与ACC及CF相加
操作: 说明: 影响的标志位	$ACC \leftarrow (R) + (ACC) + (CF)$ R6至R0寻址的数据存储器地内容与ACC和CF二进制相加, 结果置于ACC CF & ZF
ADC WR, #I	立即数与WR和CF相加
操作 说明 影响标志位	$ACC \leftarrow (WR) + I + (CF)$ 工作寄存器的内容与立即数和CF相加, 结果送到ACC CF & ZF
ADCR R, ACC	R与ACC和CF相加
操作: 说明: 影响的标志位:	$ACC, R \leftarrow (R) + (ACC) + (CF)$ R6 至 R0寻址的数据存储器的内容与ACC和CF二进制相加,结果置于ACC和数据存储器。 CF & ZF
ADCR WR, #I	立即数与WR 和 CF相加
操作: 说明: 影响的标志位:	$ACC, WR \leftarrow (WR) + I + (CF)$ 工作寄存器的内容与立即数及CF二进制相加, 结果置于ACC和WR。 CF & ZF
ADD R, ACC	R与ACC相加
操作: 说明: 影响的标志位:	$ACC \leftarrow (R) + (ACC)$ R0至R6寻址的数据存储器的内容于ACC相加, 结果置于ACC CF & ZF
ADD WR, #I	WR与立即数相加
操作: 说明 影响的标志位	$ACC \leftarrow (WR) + I$ 工作寄存器的内容 (WR) 和立即数相加, 结果置于ACC。 CF & ZF

指令表 2, 续

ADDR R, ACC	R 与ACC相加
操作: 说明: 影响的标志位:	ACC, R \leftarrow (R) + (ACC) R6至R0寻址的数据存储器的内容与ACC二进制相加, 结果置于ACC和数据存储器 CF & ZF
ADDR WR, #I	立即数与WR相加
操作: 说明: 影响标志位:	ACC, WR \leftarrow (WR) + I 工作寄存器 (WR) 与立即数二进制相, 结果置于ACC和CF & ZF
ADU R, ACC	R 与ACC 相加, 进位标志不改变
操作: 说明: 影响标志位:	ACC \leftarrow (R) + (ACC) R0至R6寻址的数据存储器的内容与ACC二进制相加, 结果置于ACC ZF
ADU WR, #I	立即数与WR相加, 进位标志不改变
操作: 说明: 影响标志位:	ACC \leftarrow (WR) + I 工作寄存器的内容和立即数相加, 结果置于ACC。 ZF
ADUR R, ACC	R 与 ACC 相加, 进位标志不改变
操作: 说明: 影响标志位:	ACC, R \leftarrow (R) + (ACC) R6至R0寻址的数据存储器的内容与ACC二进制相加, 结果置于ACC和数据存储器。 ZF

续指令表2, 续

ANL R, ACC	R与ACC直接相与
操作: 说明: 影响的标志位:	ACC ← (R) & (ACC) R6至R0寻址的数据存储器的内容与ACC的内容相与, 结果置于ACC。 ZF
ANL WR, #I	立即数与WR相与
操作 说明 影响的标志位	ACC ← (WR) & I 工作寄存器的内容与立即数相与, 结果置于ACC。 ZF
ANLR R, ACC	R与ACC直接相与
操作 说明 影响的标志位	ACC, R ← (R) & (ACC) R6 至R0 寻址的数据存储器的内容与ACC相与, 结果置于ACC和数据存储器。 ZF
ANLR WR, #I	立即数与WR相与
操作: 说明: 影响的标志位:	ACC, WR ← (WR) & I 工作寄存器 (WR)的内容与立即数相与, 结果置于ACC和工作寄存WR。 ZF
CALL L	调子程序
操作: 说明:	STACK ← (PC)+1; PC10-PC0 ← L10-L0 下一个程序计数器 (PC10至PC0) 存储于堆栈寄存器, 程序存储器存放直接地址 (L10至L0)。 调用子程序

指令表2, 续

CLR	CF	清除CF																
操作:		清除CF																
说明:		进位标志置0.																
影响的标志位:		CF																
CLR	DIVR0	分频器0的后四位复位																
操作:		分频器0的后四位复位																
说明:		执行该指令性时, 分频器 (14位) 的后四位复位。																
CLR	EVF, #I	清除事件标志																
操作:		清除事件标志																
说明:		可以控制I7至I0规定的条件对应的条件。																
		<table border="1"> <thead> <tr> <th>I0 to I8</th> <th>操作描述</th> </tr> </thead> <tbody> <tr> <td>I0 = 1</td> <td>清除分频器0的溢出事件标志位.</td> </tr> <tr> <td>I1 = 1</td> <td>清除定时器0的溢出事件标志位.</td> </tr> <tr> <td>I2 = 1</td> <td>清除RC端口变化事件标志位。</td> </tr> <tr> <td>I3 = 1</td> <td>保留</td> </tr> <tr> <td>I4 = 1</td> <td>清除INT脚上的下降沿事件标志位.</td> </tr> <tr> <td>I5, I6</td> <td>保留</td> </tr> <tr> <td>I7 = 1</td> <td>清除定时器1的溢出事件标志位.</td> </tr> </tbody> </table>	I0 to I8	操作描述	I0 = 1	清除分频器0的溢出事件标志位.	I1 = 1	清除定时器0的溢出事件标志位.	I2 = 1	清除RC端口变化事件标志位。	I3 = 1	保留	I4 = 1	清除INT脚上的下降沿事件标志位.	I5, I6	保留	I7 = 1	清除定时器1的溢出事件标志位.
I0 to I8	操作描述																	
I0 = 1	清除分频器0的溢出事件标志位.																	
I1 = 1	清除定时器0的溢出事件标志位.																	
I2 = 1	清除RC端口变化事件标志位。																	
I3 = 1	保留																	
I4 = 1	清除INT脚上的下降沿事件标志位.																	
I5, I6	保留																	
I7 = 1	清除定时器1的溢出事件标志位.																	

指令表2, 续

CLR	PMF, #I	清除参数标志
操作:		清除参数标志
说明:		各个标志的说明: I0, I1, I2: 保留 I3 = 1: 看门狗定时器的输入时钟为Fosc/1024.
CLR	PSR0	清除端口状态寄存器0 (RC口信号改变标志)
操作:		清除端口状态寄存器0 (RC口信号改变标志)
说明:		执行改指令时, 清除RC口信号改变标志
CLR	WDT	看门狗定时器的后四位复位
操作:		看门狗定时器的后四位复位
说明:		执行该指令后, 看门定时器的后四位复位
DEC	R	R的内容减1
操作:		ACC, R ← (R) - 1
说明:		数据存储器内容减1, 结果置于ACC和数据存储器
影响标志位:		CF & ZF
DIS	INT	禁止中断
操作:		禁止中断
说明:		中断被禁止

指令表 2, 续

DSKNZ	R	R的内容减1, 若ACC != 0 则跳转
操作:		ACC, R \leftarrow (R) - 1; PC \leftarrow (PC) + 2 if ACC != 0
说明:		数据存储器的内容减1, 结果置于ACC和数据存储器; 若ACC != 0, 程序计数器的内容加2, 产生跳转
影响的标志位:		CF & ZF
DSKZ	R	R的内容减1, 若ACC=0, 则跳转
操作:		ACC, R \leftarrow (R) - 1; PC \leftarrow (PC) + 2 if ACC = 0
说明:		数据存储器的内容减1, 结果置于ACC和数据存储器; 若ACC=0, 则程序计数器加2, 产生跳转
影响的标志位:		CF & ZF
EN	INT	中断允许
操作:		中断允许
说明:		允许中断功能
HOLD		进入HOLD状态
操作:		进入HOLD功能
说明:		下面两个条件退出HOLD状态. (1) 中断产生 (2) 遇到HEF规定的HOLD模式退出条件 IHOLD模式时, 若中断产生将退出HOLD模式, 而执行中断子程序, 执行完中断子程序的RTN指令后, UC将再次进入HOLD模式

指令表 2, 续

INC	R	R的内容加1
操作:		ACC, R \leftarrow (R) + 1
说明:		数据存储器的内容加1, 结果置于ACC和数据存储器 CF & ZF
影响的标志位:		
JB0	L	若ACC的第0位是1, 则产生跳转
操作:		PC10-PC0 \leftarrow L10-L0; if ACC.0 = "1"
说明:		若ACC的第0位为1, 则L10至L0规定的替换程序计数器的PC10至PC0, 且跳转, 若ACC的第0位为0, 则程序计数器加1。
JB1	L	ACC的第1位为1, 则跳转
操作:		PC10-PC0 \leftarrow L10-L0; if ACC.1 = "1"
说明:		若ACC的第1位为1, 则L10至L0规定的替换程序计数器的PC10至PC0, 且跳转, 若ACC的第0位为0, 则程序计数器加1。
JB2	L	ACC的第2位为1, 则跳转
操作:		PC10-PC0 \leftarrow L10-L0; if ACC.2 = "1"
说明:		若ACC的第2位为1, 则L10至L0规定的替换程序计数器的PC10至PC0, 且跳转, 若ACC的第0位为0, 则程序计数器加1。

指令表 2, 续

JB3	L	ACC的第1位为1, 则跳转
操作:	PC10-PC0 ← L10-L0; if ACC.3 = "1"	
说明:	若ACC的第3位为1, 则L10至L0规定的的数据替换程序计数器的PC10至PC0, 且跳转, 若ACC的第0位为0, 则程序计数器加1。	
JC	L	CF为1, 则跳转
操作:	PC10-PC0 ← L10-L0; if CF = "1"	
说明:	若CF为1, 则L10至L0规定的的数据替换程序计数器的PC10至PC0, 且跳转, 若CF为0, 则程序计数器加1。	
JMP	L	无条件跳转
操作:	PC10-PC0 ← L10-L0	
说明:	L10至L0规定的的数据替换程序计数器的PC10至PC0, 且跳转	
JNC	L	若CF不为1, 则跳转
操作:	PC10-PC0 ← L10-L0; if CF = "0"	
说明:	若CF为0, 则L10至L0规定的的数据替换程序计数器的PC10至PC0, 且跳转, 若CF为1, 则程序计数器加1。	
JNZ	L	ACC不为0, 则跳转
操作:	PC10-PC0 ← L10-L0; if ACC ≠ 0	
说明:	若ACC不为0, 则L10至L0规定的的数据替换程序计数器的PC10至PC0, 且跳转, 若ACC为0, 则程序计数器加1。	

指令表 2, 续

JZ	L	ACC为0, 则跳转
操作:		$PC_{10}-PC_0 \leftarrow L_{10}-L_0$; if $ACC = 0$
说明:		若ACC为0, 则L10至L0规定的數據替换程序计数器的PC10至PC0, 且跳转, 若ACC不为0, 则程序计数器加1。
LCDON		打开LCD
操作:		打开LCD
说明:		打开LCD显示
LCDOFF		关闭LCD
操作:		关闭LCD
说明:		关闭LCD显示
MOV	ACC, R	R的内容送到ACC
操作:		$ACC \leftarrow (R)$
说明:		R6至R0地址的数据存储器的内容送到ACC
MOV	CF, R	将R.0送到CF
操作:		$CF \leftarrow (R.0)$
说明:		以R6至R0地址的数据存储器的第0位送到CF



指令表 2, 续

MOV HEF, #I	设置重置HOLD模式退出允许标志																	
操作:	控制HOLD模式退出允许标志																	
说明:	<table border="1"> <thead> <tr> <th>I0 to I7</th> <th>操作描述</th> </tr> </thead> <tbody> <tr> <td>I0 = 1</td> <td>允许分频器0溢出解除HOLD模式.</td> </tr> <tr> <td>I1 = 1</td> <td>允许定时器0溢出解除HOLD模式.</td> </tr> <tr> <td>I2 = 1</td> <td>允许RC端口信号变化解除HOLD模式.</td> </tr> <tr> <td>I3 = 1</td> <td>保留.</td> </tr> <tr> <td>I4 = 1</td> <td>允许INT脚上的下降沿解除HOLD模式.</td> </tr> <tr> <td>I5 & I6</td> <td>保留.</td> </tr> <tr> <td>I7 = 1</td> <td>允许定时器1溢出解除HOLD模式.</td> </tr> </tbody> </table>		I0 to I7	操作描述	I0 = 1	允许分频器0溢出解除HOLD模式.	I1 = 1	允许定时器0溢出解除HOLD模式.	I2 = 1	允许RC端口信号变化解除HOLD模式.	I3 = 1	保留.	I4 = 1	允许INT脚上的下降沿解除HOLD模式.	I5 & I6	保留.	I7 = 1	允许定时器1溢出解除HOLD模式.
I0 to I7	操作描述																	
I0 = 1	允许分频器0溢出解除HOLD模式.																	
I1 = 1	允许定时器0溢出解除HOLD模式.																	
I2 = 1	允许RC端口信号变化解除HOLD模式.																	
I3 = 1	保留.																	
I4 = 1	允许INT脚上的下降沿解除HOLD模式.																	
I5 & I6	保留.																	
I7 = 1	允许定时器1溢出解除HOLD模式.																	

指令表 2, 续

MOV	IEF, #I	设置/重置中断允许标志																
操作:		中断允许标志的控制																
说明:		控制I7-I0规定的的数据所相应的中断允许标志																
		<table border="1"> <thead> <tr> <th>I0 to I5</th> <th>操作描述</th> </tr> </thead> <tbody> <tr> <td>I0 = 1</td> <td>允许(分频器0溢出) 中断0标志.</td> </tr> <tr> <td>I1 = 1</td> <td>允许(定时器0溢出) 中断1标志.</td> </tr> <tr> <td>I2 = 1</td> <td>允许 (RC端口信号变化) 中断2标志.</td> </tr> <tr> <td>I3 = 1</td> <td>保留.</td> </tr> <tr> <td>I4 = 1</td> <td>允许(INT脚上的下降沿) 中断4标志.</td> </tr> <tr> <td>I5 & I6</td> <td>保留.</td> </tr> <tr> <td>I7 = 1</td> <td>允许(定时器1溢出) 中断7标志.</td> </tr> </tbody> </table>	I0 to I5	操作描述	I0 = 1	允许(分频器0溢出) 中断0标志.	I1 = 1	允许(定时器0溢出) 中断1标志.	I2 = 1	允许 (RC端口信号变化) 中断2标志.	I3 = 1	保留.	I4 = 1	允许(INT脚上的下降沿) 中断4标志.	I5 & I6	保留.	I7 = 1	允许(定时器1溢出) 中断7标志.
I0 to I5	操作描述																	
I0 = 1	允许(分频器0溢出) 中断0标志.																	
I1 = 1	允许(定时器0溢出) 中断1标志.																	
I2 = 1	允许 (RC端口信号变化) 中断2标志.																	
I3 = 1	保留.																	
I4 = 1	允许(INT脚上的下降沿) 中断4标志.																	
I5 & I6	保留.																	
I7 = 1	允许(定时器1溢出) 中断7标志.																	
MOV	LCDM, #I	选择LCD输出模式型式																
操作:		选择LCD输出模式型式																
说明:		LCD输出脚设置为直流输出模式时, 用户可以选择CMOS或NMOS输出型式 I0-I5=>CMOS型式; I0-I5=1>NMOS型式																
MOV	LCDR, ACC	ACC 的内容传送到LCDR																
操作:		LCDR ← (ACC)																
说明:		ACC的内容传送到LCD 数据RAM (LCDR) , LCDR由D4至D0取址																



指令表 2, 续

MOV LCDR, WR	WR 的内容送到LCDR																													
操作:	LCDR ← (WR)																													
说明:	WR的内容传送到LCD 数据RAM (LCDR) , LCDR由D4至D0取址																													
MOV LCDR, #I	将立即数送到LCDR																													
操作:	LCDR ← I																													
说明:	立即数送到LCD 数据RAM (LCDR) , LCDR由D4至D0取址																													
MOV MFP, #I	调制输出频率																													
操作:	[MFP] ← I																													
说明:	<p>I如果MR1的第2位为0, 由I7 至 I0 控制的输出波形由MFP脚输出, 输出波形与立即数的关系如下:</p> <table border="1" style="margin-left: 20px;"> <tr> <td>I5~I0</td> <td>I0 = 1</td> <td>I1 = 1</td> <td>I2 = 1</td> <td>I3 = 1</td> <td>I4 = 1</td> <td>I5 = 1</td> </tr> <tr> <td>Signal</td> <td>$\frac{Fosc}{256}$</td> <td>$\frac{Fosc}{512}$</td> <td>$\frac{Fosc}{4096}$</td> <td>$\frac{Fosc}{8192}$</td> <td>$\frac{Fosc}{16384}$</td> <td>$\frac{Fosc}{32768}$</td> </tr> </table> <table border="1" style="margin-left: 20px;"> <tr> <td>I7</td> <td>I6</td> <td>Signal</td> </tr> <tr> <td>0</td> <td>0</td> <td>Low</td> </tr> <tr> <td>0</td> <td>1</td> <td>High</td> </tr> <tr> <td>1</td> <td>0</td> <td>Fosc/16</td> </tr> <tr> <td>1</td> <td>1</td> <td>Fosc/8</td> </tr> </table>	I5~I0	I0 = 1	I1 = 1	I2 = 1	I3 = 1	I4 = 1	I5 = 1	Signal	$\frac{Fosc}{256}$	$\frac{Fosc}{512}$	$\frac{Fosc}{4096}$	$\frac{Fosc}{8192}$	$\frac{Fosc}{16384}$	$\frac{Fosc}{32768}$	I7	I6	Signal	0	0	Low	0	1	High	1	0	Fosc/16	1	1	Fosc/8
I5~I0	I0 = 1	I1 = 1	I2 = 1	I3 = 1	I4 = 1	I5 = 1																								
Signal	$\frac{Fosc}{256}$	$\frac{Fosc}{512}$	$\frac{Fosc}{4096}$	$\frac{Fosc}{8192}$	$\frac{Fosc}{16384}$	$\frac{Fosc}{32768}$																								
I7	I6	Signal																												
0	0	Low																												
0	1	High																												
1	0	Fosc/16																												
1	1	Fosc/8																												

指令表 2, 续

MOV	MR0, #I	将立即数传送到模式寄存器0 (MR0)								
操作:		$MR0 \leftarrow I$								
说明:		立即数被传送到模式寄存器0. 模式寄存器0位描述如下:								
		<table border="1"> <tr> <td>bit 0</td> <td>= 0 定时器0的内置基本频率为 $F_{osc}/4$. = 1 定时器0的内置基本频率为 $F_{osc}/1024$.</td> </tr> <tr> <td>bit 1</td> <td>保留</td> </tr> <tr> <td>bit 2</td> <td>保留</td> </tr> <tr> <td>bit 3</td> <td>= 0 定时器 0 停止向下计时. = 1 定时器 0 启动向下计时.</td> </tr> </table>	bit 0	= 0 定时器0的内置基本频率为 $F_{osc}/4$. = 1 定时器0的内置基本频率为 $F_{osc}/1024$.	bit 1	保留	bit 2	保留	bit 3	= 0 定时器 0 停止向下计时. = 1 定时器 0 启动向下计时.
bit 0	= 0 定时器0的内置基本频率为 $F_{osc}/4$. = 1 定时器0的内置基本频率为 $F_{osc}/1024$.									
bit 1	保留									
bit 2	保留									
bit 3	= 0 定时器 0 停止向下计时. = 1 定时器 0 启动向下计时.									
MOV	MR1, #I	立即数传送至模式寄存器1 (MR1)								
操作:		$MR1 \leftarrow I$								
说明:		立即数被传送到模式寄存器0. 模式寄存器0位描述如下:								
		<table border="1"> <tr> <td>bit0</td> <td>= 0 定时器1的内置基本频率为 F_{osc}. = 1 定时器1的内置基本频率为 $F_{osc}/64$.</td> </tr> <tr> <td>bit1</td> <td>= 0 定时器1的内置基本频率源为内部时钟 = 1 定时器1的内置基本频率源为透过RC.0的外部时钟.</td> </tr> <tr> <td>bit2</td> <td>= 0 把MFP信号产生器产生的信号送到MFP输出脚 = 1 把定时器1产生的信号送到MFP输出脚</td> </tr> <tr> <td>bit3</td> <td>= 0 定时器 1 停止向下计时. = 1 定时器 1 启动向下计时.</td> </tr> </table>	bit0	= 0 定时器1的内置基本频率为 F_{osc} . = 1 定时器1的内置基本频率为 $F_{osc}/64$.	bit1	= 0 定时器1的内置基本频率源为内部时钟 = 1 定时器1的内置基本频率源为透过RC.0的外部时钟.	bit2	= 0 把MFP信号产生器产生的信号送到MFP输出脚 = 1 把定时器1产生的信号送到MFP输出脚	bit3	= 0 定时器 1 停止向下计时. = 1 定时器 1 启动向下计时.
bit0	= 0 定时器1的内置基本频率为 F_{osc} . = 1 定时器1的内置基本频率为 $F_{osc}/64$.									
bit1	= 0 定时器1的内置基本频率源为内部时钟 = 1 定时器1的内置基本频率源为透过RC.0的外部时钟.									
bit2	= 0 把MFP信号产生器产生的信号送到MFP输出脚 = 1 把定时器1产生的信号送到MFP输出脚									
bit3	= 0 定时器 1 停止向下计时. = 1 定时器 1 启动向下计时.									



指令表 2, 续

MOV PAGE, #I	将立即数传送到页寄存器																																				
操作:	Page Register ← I																																				
说明:	<p>将立即数传送到页寄存器</p> <p>位3 被保留, 位0, 位1, 位2 间接寻址模式的预选位</p> <table border="1"> <thead> <tr> <th>位2</th> <th>位1</th> <th>位0</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>= 页 0 (00H 到 0FH)</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>= 页 1 (10H 到 1FH)</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>= 页 2 (20H 到 2FH)</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>= 页 3 (30H 到 3FH)</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>= 页 4 (40H 到 4FH)</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>= 页 5 (50H 到 5FH)</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>= 页 6 (60H 到 6FH)</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>= 页 7 (70H 到 7FH)</td> </tr> </tbody> </table>	位2	位1	位0		0	0	0	= 页 0 (00H 到 0FH)	0	0	1	= 页 1 (10H 到 1FH)	0	1	0	= 页 2 (20H 到 2FH)	0	1	1	= 页 3 (30H 到 3FH)	1	0	0	= 页 4 (40H 到 4FH)	1	0	1	= 页 5 (50H 到 5FH)	1	1	0	= 页 6 (60H 到 6FH)	1	1	1	= 页 7 (70H 到 7FH)
位2	位1	位0																																			
0	0	0	= 页 0 (00H 到 0FH)																																		
0	0	1	= 页 1 (10H 到 1FH)																																		
0	1	0	= 页 2 (20H 到 2FH)																																		
0	1	1	= 页 3 (30H 到 3FH)																																		
1	0	0	= 页 4 (40H 到 4FH)																																		
1	0	1	= 页 5 (50H 到 5FH)																																		
1	1	0	= 页 6 (60H 到 6FH)																																		
1	1	1	= 页 7 (70H 到 7FH)																																		
MOV PEF, #I	设置/重置端口使能标志																																				
操作:	控制端口的使能标志																																				
说明:	<p>在相应位被置1以后, RC口信号的改变可以导致HOLD模式的推出或者中断的产生</p> <table border="1"> <thead> <tr> <th>I0 to I3</th> <th>RC端口信号的改变</th> </tr> </thead> <tbody> <tr> <td>I0 = 1</td> <td>RC0</td> </tr> <tr> <td>I1 = 1</td> <td>RC1</td> </tr> <tr> <td>I2 = 1</td> <td>RC2</td> </tr> <tr> <td>I3 = 1</td> <td>RC3</td> </tr> </tbody> </table>	I0 to I3	RC端口信号的改变	I0 = 1	RC0	I1 = 1	RC1	I2 = 1	RC2	I3 = 1	RC3																										
I0 to I3	RC端口信号的改变																																				
I0 = 1	RC0																																				
I1 = 1	RC1																																				
I2 = 1	RC2																																				
I3 = 1	RC3																																				

指令表 2,续

MOV PM0, #I	设置/重置端口模式0寄存器
操作:	设置/重置端口模式0寄存器
说明:	<p>I0 = 0: RA 端口是CMOS类型; I0 = 1: RA 端口 NMOS 类型.</p> <p>I1 = 0: RB 端口是CMOS 类型; I1 = 1: RB 端口 NMOS 类型.</p> <p>I2 = 0: RC 端口上拉电阻无效</p> <p>I2 = 1: RC 端口上拉电阻有效</p> <p>I3 = 0: RD; 端口上拉电阻无效</p> <p>I3 = 1: RD. 端口上拉电阻有效</p>
MOV PM1, #I	RA 端口的输入/输出控制
操作:	控制四位RA口的输入/输出形式
说明:	<p>I0 = 0: RA.0 是输出脚; I0 = 1: RA.0 是输入脚</p> <p>I1 = 0: RA.1 是输出脚; I1 = 1: RA.1是输入脚</p> <p>I2 = 0: RA.2是输出脚; I2 = 1: RA.2是输入脚</p> <p>I3 = 0: RA.3是输出脚; I3 = 1: RA.3是输入脚</p> <p>RA的缺省值为输入模式 (PM 1= 1111B).</p>
MOV PM2, #I	RB 口输入/输出独立控制
操作:	RB口的四个输入/输出脚独立控制
说明:	<p>I0 = 0: RB.0为输出脚; I0 = 1: RB.0 为输入脚</p> <p>I1 = 0: RB.1为输出脚; I1 = 1: RB.1为输入脚</p> <p>I2 = 0: RB.2为输出脚; I2 = 1: RB.2为输入脚</p> <p>I3 = 0: RB.3为输出脚; I3 = 1: RB.3为输入脚</p> <p>RB 口的缺省值是输入模式 (PM2 = 1111B).</p>

指令表 2,续

MOV R, ACC	ACC内容传送到R
操作:	$R \leftarrow (ACC)$
说明:	ACC的内容传送到R6至R0去取址的数据存储器
MOVA R, RA	输入RA口数据传送到ACC & R
操作:	$ACC, R \leftarrow [RA]$
说明:	输入口RA的内容传送到R0至R6取址的数据存储器和ACC
影响标志位:	ZF
MOVA R, RB	输入RA口数据传送到ACC & R
操作:	$ACC, R \leftarrow [RB]$
说明:	输入口RB的内容传送到R0至R6取址的数据存储器和ACC
影响标志位:	ZF
MOVA R, RC	输入RC口数据传送到ACC & R
操作:	$ACC, R \leftarrow [RC]$
说明:	输入口RC的内容传送到R0至R6取址的数据存储器和ACC
影响标志位:	ZF
MOVA R, RD	输入RA口数据传送到ACC & R
操作:	$ACC, R \leftarrow [RD]$
说明:	输入口RD的内容传送到R0至R6取址的数据存储器和ACC
影响标志位:	ZF

指令表2, 续

MOV R, WR	WR的内容送到 R
操作:	$R \leftarrow (WR)$
说明:	WR的内容送到R6至R0.取址的数据存储器
MOV R, #I	立即数送到R
操作:	$R \leftarrow I$
说明:	立即数送到R6至R0取址的数据存储器
MOV RA, R	输出R的内容到RA口
操作:	$[RA] \leftarrow (R)$
说明:	数据存储器中的数据输出到RA口
MOV RB, R	输出R的内容到RB口
操作:	$[RB] \leftarrow (R)$
说明:	数据存储器中的数据输出到RB口
MOV RE, R	输出R的内容到RE口
操作:	$[RE] \leftarrow (R)$
说明:	数据存储器中的数据输出到RA口



指令表2, 续

<p>MOV SEF, #I</p> <p>操作:</p> <p>说明:</p>	<p>设置/重置RC口的唤起STOP模式允许标志</p> <p>设置/重置RC口的唤起STOP模式允许标志 I定义的数据唤起STOP模式, RC口的下降沿信号独立定义</p> <table border="1"> <tr> <td>I0 to I3</td> <td>RC端口上的下降沿</td> </tr> <tr> <td>I0 = 1</td> <td>RC0</td> </tr> <tr> <td>I1 = 1</td> <td>RC1</td> </tr> <tr> <td>I2 = 1</td> <td>RC2</td> </tr> <tr> <td>I3 = 1</td> <td>RC3</td> </tr> </table>	I0 to I3	RC端口上的下降沿	I0 = 1	RC0	I1 = 1	RC1	I2 = 1	RC2	I3 = 1	RC3
I0 to I3	RC端口上的下降沿										
I0 = 1	RC0										
I1 = 1	RC1										
I2 = 1	RC2										
I3 = 1	RC3										
<p>MOV TM0, #I</p> <p>操作:</p> <p>说明:</p>	<p>设置定时器0</p> <p>设置定时器0</p> <p>I7 至 I0 定义的数据传送到定时器0, 启动定时器</p>										
<p>MOV TM0L, R</p> <p>操作:</p> <p>说明:</p>	<p>将R的内容送到TM0L</p> <p>TM0L ← (R)</p> <p>R6 to R0 取址的数据存储器的内容送到TM0L.</p>										

指令表 2, 续

MOV TM0H, R	将R的内容送到TM0H
操作: 说明:	TM0H ← (R) R6 to R0 取址的数据存储器的内容送到TM0H
MOV TM1, #I	设置定时器1
操作: 说明:	设置定时器1 I7 至 I0 定义的数据传送到定时器1, 启动定时器
MOV TM1L, R	将R的内容送到TM1L
操作: 说明:	TM1L ← (R) R6 to R0 取址的数据存储器的内容送到TM1L.
MOV TM1H, R	将R的内容送到TM1H
操作: 说明:	TM1H ← (R) R6 to R0 取址的数据存储器的内容送到TM1H
MOV TABL, R	将R的内容送入TABL
操作: 说明:	TABL ← (R) R6 至 R0 取址的数据存储器器的内容送入TABL.
MOV TABH, R	将R的内容送入TABH
操作: 说明:	TABH ← (R) R6 至 R0 取址的数据存储器器的内容送入TABL.

指令表, 续

MOV WR, LCDR	将 LCDR中的内容送入到 WR
操作:	$WR \leftarrow (LCDR)$
说明:	D4至D0寻址的LCD数据RAM的内容送到WR
MOV WR, R	R的内容送到WR
操作:	$WR \leftarrow (R)$
说明:	R6 至 R0 寻址的数据存储器器的内容送到WR。
MOV WR, @R	间接传送R 至 WR
操作:	$WR \leftarrow [PR (\text{位}2, \text{位}1, \text{位}0) \times 10H + (R)]$
说明:	$[PR (\text{位}2, \text{位}1, \text{位}0) \times 10H + (R)]$ 的数据存储器的地址送到
MOV @R, WR	间接传送WR 到 R
操作:	$[PR (\text{bit}2, \text{bit}1, \text{bit}0) \times 10H + (R)] \leftarrow WR$
说明:	WR的内容送到 $[PR (\text{bit}2, \text{bit}1, \text{bit}0) \times 10H + (R)]$ 寻址的数据存储器。
MOV PAGE, R	R的内容送到页寄存器
操作:	$PR \leftarrow (R)$
说明:	R6 至 R0 寻址的数据存储器器的内容送到页寄存器 PR

指令表 2, 续

MOVA R, CF	CF的内容送到 ACC.0 & R.0								
操作:	ACC.0, R.0 ← (CF)								
说明:	CF的内容送到R6至R0寻址的数据存储器的位0, 其他的数据存储器各位和ACC设为0								
影响的标志位:	ZF								
MOVA R, HCFH	HCF4-7 送到ACC & R								
操作:	ACC, R ← HCF4-7								
说明:	HCF第4位至第7位 (HCF4至HCF7) 的内容送到R6至R0寻址的数据存储器。执行该指令后ACC的内容和位定义如下:								
影响的标志位:	<table border="1"> <tr> <td>Bit 0</td> <td>INT脚上的下降沿解除HOLD模式标志位.</td> </tr> <tr> <td>Bit 1</td> <td>定时器1溢出解除HOLD模式标志位.</td> </tr> <tr> <td>Bit 2</td> <td>保留.</td> </tr> <tr> <td>Bit 3</td> <td>保留.</td> </tr> </table>	Bit 0	INT脚上的下降沿解除HOLD模式标志位.	Bit 1	定时器1溢出解除HOLD模式标志位.	Bit 2	保留.	Bit 3	保留.
Bit 0	INT脚上的下降沿解除HOLD模式标志位.								
Bit 1	定时器1溢出解除HOLD模式标志位.								
Bit 2	保留.								
Bit 3	保留.								
	ZF								
MOVA R, HCFL	HCF0-3 送到 ACC & R								
操作:	ACC, R ← HCF0-3								
说明:	HCF第0位至第3位 (HCF0至HCF3) 送到R6至R0寻址的数据存储器和ACC。该指令后, ACC的内容和位定义如下								
影响的标志位:	<table border="1"> <tr> <td>Bit 0</td> <td>分频器0溢出解除HOLD模式标志位.</td> </tr> <tr> <td>Bit 1</td> <td>定时器0溢出解除HOLD模式标志位.</td> </tr> <tr> <td>Bit 2</td> <td>RC端口上的信号变化解除HOLD模式标志位.</td> </tr> <tr> <td>Bit 3</td> <td>保留.</td> </tr> </table>	Bit 0	分频器0溢出解除HOLD模式标志位.	Bit 1	定时器0溢出解除HOLD模式标志位.	Bit 2	RC端口上的信号变化解除HOLD模式标志位.	Bit 3	保留.
Bit 0	分频器0溢出解除HOLD模式标志位.								
Bit 1	定时器0溢出解除HOLD模式标志位.								
Bit 2	RC端口上的信号变化解除HOLD模式标志位.								
Bit 3	保留.								
影响的标志位:	ZF								

指令表 2续

MOVA R, PAGE	页寄存器的内容送到ACC & R
操作:	ACC, R ← (页寄存器)
说明:	页寄存器的内容送到R6至R0寻址的数据处理存储器和ACC
影响的标志位	ZF
MOVA R, PSR0	端口状态寄存器的内容送到ACC & R
操作:	ACC, R ← RC信号改变标志(PSR0)
说明:	RC口信号改变标致的内容送到R6至R0寻址的数据存储器和ACC, 当RC口的任意脚信号改变时, 相应的信号改变标志位摄制为1, 否则, 为0。
影响的标志位	ZF
MOVA R, WR	WR 的内容送到ACC & R
操作:	ACC, R ← (WR)
说明:	WR的内容送到ACC和R6至R0寻址的数据存储器
影响的标志位	ZF
MOVA WR, R	R的内容送到ACC & WR
操作:	ACC, WR ← (R)
说明:	R6至R0寻址的数据存储器的内容送到WR和ACC。
影响的标志位:	ZF

指令表2, 续

MOVC R	通过查询表和ACC寻址的内容送WR
操作:	$R \leftarrow [((\text{TABH}) \times 100\text{H} + (\text{TABL})) \times 10\text{H} + \text{ACC}]$
说明:	由TABH, TABL 和ACC 寻址的查询ROM的内容送到R.
MOVC WR, #I	将由立即数和ACC寻址的查询表ROM的内容送到WR
操作:	$\text{WR} \leftarrow [(I6 \sim I0) \times 10\text{H} + (\text{ACC})]$
说明:	I6至I0寻址的查询表ROM内容和ACC送到工作寄存器WR
NOP	空操作
操作:	空操作
ORL R, ACC	R与ACC相或
操作:	$\text{ACC} \leftarrow (R) \wedge (\text{ACC})$
说明:	R6至R0寻址的数据存储器的内容与ACC相或, 结果置于ACC。
影响的标志位	ZF
ORL WR, #I	WR与立即数相或
操作:	$\text{ACC} \leftarrow (\text{WR}) \wedge I$
说明:	工作寄存器的内容与立即数相或, 结果置于ACC。
影响的标志位	ZF

指令表 2, 续

ORLR	R, ACC	R与ACC相或
操作:		$ACC, R \leftarrow (R) \wedge (ACC)$
说明:		R6至R0寻址的数据存储器的内容与ACC相或, 结果置于ACC和数据存储器。
影响的标志位		ZF
ORLR	WR, #	WR与立即数相或
操作:		$ACC, WR \leftarrow (WR) \wedge I$
说明:		工作寄存器的内容与立即数相或, 结果置于ACC和WR。
影响的标志位		ZF
RLC	R	左移R和CF
操作:		$ACC.n, R.n \leftarrow R.n-1; ACC.0, R.0 \leftarrow CF; CF \leftarrow R.3$
说明:		ACC的内容和R6至R0寻址的数据存储器左移一位, 第3位移至CF, CF移到第0位(LSB), 相同的内容送至ACC
影响的标志位		CF & ZF
RRC	R	右移R和CF
操作:		$ACC.n, R.n \leftarrow R.n+1; ACC.3, R.3 \leftarrow CF; CF \leftarrow R.0$
说明:		ACC的内容和R6至R0寻址的数据存储器右移一位, 第0位移至CF, CF移到第3位(MSB), 相同的内容送至ACC
影响的标志位		CF & ZF

指令表 2, 续

RTN	子程序返回
操作:	$(PC) \leftarrow \text{STACK}$
说明:	程序计数器 (PC10 至PC0) 从堆栈中还原, 此时, 从子程序返回到主程序。
SBC R, ACC	R 减进位和ACC
操作:	$ACC \leftarrow (R) - (ACC) - (CF)$
说明:	R6 至R0 寻址的数据存储器减去ACC和CF, 结果存于ACC。
影响的标志位	CF & ZF
SBC WR, #I	WR减立即数和进位位
操作:	$ACC \leftarrow (WR) - I - (CF)$
说明:	WR减立即数和CF, 结果置于ACC
影响的标志位	CF & ZF
SBCR R, ACC	R 减ACC和进位位
操作:	$ACC, R \leftarrow (R) - (ACC) - (CF)$
说明:	R6 至R0 寻址的数据存储器减去ACC和CF, 结果存于ACC和数据存储器
影响的标志位	CF & ZF

指令表2, 续

SBCR	WR, #I	WR减立即数和进位位
操作:		$ACC, R \leftarrow (WR) - I - (CF)$
说明:		WR减立即数和CF, 结果置于ACC和WR
影响的标志位		CF & ZF
SET	CF	设置CF
操作:		设置CF
说明:		设置进位标志为1
影响的标志位		CF
SET	PMF, #I	设置参数标志
操作:		设置参数标志
说明:		位定义如下: I0: 保留 I1=1: 定时器0的输入时钟为Fosc/1024 I2=1: 允许看门狗定时器 I3 = 1: 看门狗定时器的输入时钟为Fosc/16384.
SHLC	R	左移R和CF且LSB = 0
操作:		$ACC.n, R.n \leftarrow R.n-1; ACC.0, R.0 \leftarrow 0; CF \leftarrow R.3$
说明:		ACC和R6至R0寻址的数据存储器的地址左移一位, 但第3位移至CF, 第0位以0替代, 相同的结果置于ACC.
影响的标志位		CF & ZF

指令表 2, 续

SHRC	R	R和CF右移且 MSB = 0
操作:		$ACC.n, R.n \leftarrow R.n+1; ACC.3, R.3 \leftarrow 0; CF \leftarrow R.0$
说明:		ACC和R6至R0寻址的数据存储器的地址右移一位, 但第0位移至CF, 第3位以0替代, 相同的结果置于ACC.
影响的标志位		CF & ZF
SKB0	R	若R的第0位为1, 则跳转
操作:		$PC \leftarrow (PC) + 2; \text{if } R.0 = "1"$
说明:		R的第0位为1时, 程序计数加班2, 跳转。R的第0位不为1时, 程序计数器(PC)不增加。
SKB1	R	若R的第1位为1, 则跳转
操作:		$PC \leftarrow (PC) + 2; \text{if } R.1 = "1"$
说明:		R的第1位为1时, 程序计数加班2, 跳转。R的第1位不为1时, 程序计数器(PC)不增加。
SKB2	R	若R的第2位为1, 则跳转
操作:		$PC \leftarrow (PC) + 2; \text{if } R.2 = "1"$
说明:		R的第2位为1时, 程序计数加班2, 跳转。R的第2位不为1时, 程序计数器(PC)不增加。

指令表 2, 续

SKB3	R	若R的第3位为1, 则跳转
操作:		$PC \leftarrow (PC) + 2; \text{ if } R.3 = "1"$
说明:		R的第3位为1时, 程序计数加班2, 跳转。R的第3位不为1时, 程序计数器(PC)不增加。
STOP		进入STOP模式
操作:		STOP晶振
说明:		芯片进入STOP模式, 当/INT脚信号变低或接受RC的下降沿信号时, 芯片被唤醒, 继续执行下一条指令。 . When the falling edge signal of RC port is accepted, the μC will wake up and execute the next instruction.
SUB	R, ACC	R减去ACC
操作:		$ACC \leftarrow (R) - (ACC)$
说明:		R6 至R0寻址的数据存储器的内容减去ACC, 结果置于ACC。 CF & ZF
SUB	WR, #I	WR减去立即数
操作:		$ACC \leftarrow (WR) - I$
说明:		WR的内容减立即数, 结果置于ACC
影响的标志位		CF & ZF

指令表 2, 续

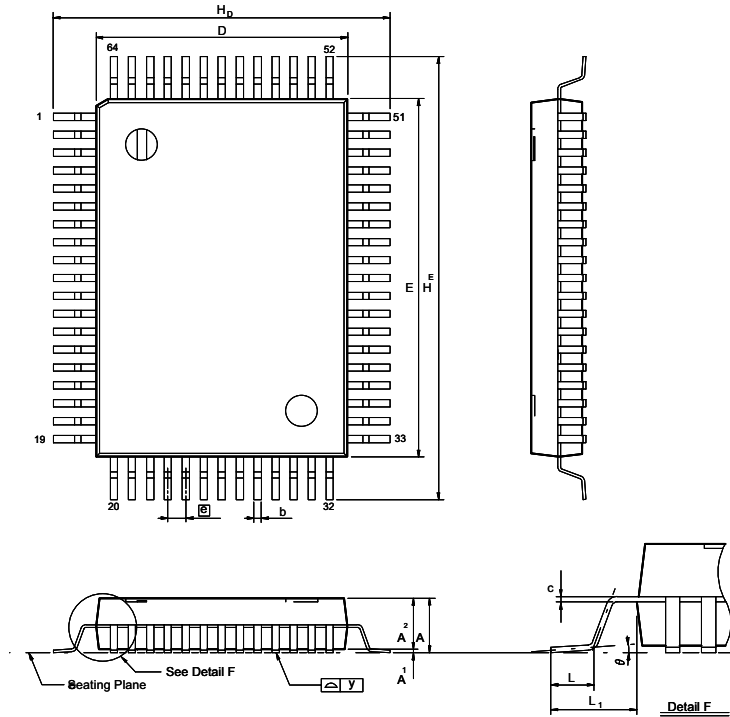
SUBR	R, ACC	R减去ACC
操作:		$ACC, R \leftarrow (R) - (ACC)$
说明:		R6 至R0寻址的数据存储器的内容减去ACC, 结果置于ACC和数据存储器。
影响的标志位		CF & ZF
SUBR	WR, #I	WR减去立即数
操作:		$ACC, WR \leftarrow (WR) - I$
说明:		WR的内容减立即数, 结果置于ACC和WR。
影响的标志位		CF & ZF
XRL	R, ACC	R与ACC相异
操作:		$ACC \leftarrow (R) \text{ EX } (ACC)$
说明:		R6 至R0寻址的数据存储器的内容与ACC相异, 结果置于ACC。
影响的标志位		ZF
XRL	WR, #I	WR与立即数相异
操作:		$ACC \leftarrow (WR) \text{ EX } I$
说明:		工作寄存器的内容与立即数相异, 结果置于ACC。
影响的标志位		ZF

指令表 2, 续

XRLR	R, ACC	R与ACC相异
操作:		ACC, R ← (R) EX (ACC)
说明:		R6 至R0寻址的数据存储器的内容与ACC相异, 结果置于ACC和数据存储器。
影响的标志位		ZF
XRLR	WR, #I	WR与立即数相或
操作:		ACC, WR ← (WR) EX I
说明:		工作寄存器的内容与立即数相异, 结果置于ACC和WR。
影响的标志位		ZF

封装尺寸

64-Lead QFP (14 × 20 × 2.75mm footprint 4.8mm)



Symbol	Dimension in inches			Dimension in mm		
	Min.	Nom	Max.	Min.	Nom	Max.
A	—	—	0.130	—	—	3.30
A ₁	0.004	—	—	0.10	—	—
A ₂	0.107	0.112	0.117	2.73	2.85	2.97
b	0.014	0.016	0.020	0.35	0.40	0.50
c	0.004	0.006	0.010	0.10	0.15	0.25
D	0.546	0.551	0.556	13.87	14.00	14.13
E	0.782	0.787	0.792	19.87	20.00	20.13
⊖	0.033	0.039	0.045	0.85	1.00	1.15
H _D	0.728	0.740	0.752	18.49	18.80	19.00
H _E	0.964	0.976	0.988	24.49	24.80	25.10
L	0.039	0.047	0.055	1.00	1.20	1.40
L ₁	0.087	0.094	0.103	2.21	2.40	2.62
y	—	—	0.004	—	—	0.10
θ	0°	—	12°	0°	—	12°