

产品目录.....	3
描述.....	3
订货信息.....	3
特征.....	3
各封装引脚配置.....	4
系统方框图.....	7
管脚描述.....	8
特殊功能寄存器(SFR).....	9
功能描述.....	13
1. 总特征.....	13
1.1 嵌入式程序存储器.....	13
1.2 IO 口.....	13
1.3 指令时钟周期选择.....	13
1.4 时钟输出选择.....	14
1.5 复位.....	14
1.5.1 硬件复位功能.....	14
1.5.2 软件复位功能.....	14
1.5.3 Reset status.....	15
1.5.4 Time Access Key register (TAKEY).....	15
1.5.5 软件复位寄存器(SWRES).....	15
1.5.6 Example of software reset.....	16
1.6 时钟源.....	16
2. 指令设置.....	17
3. 存储器结构.....	21
3.1 程序存储器.....	21
3.2 数据存储器.....	22
3.3 数据内存-低 128 字节(00h 到 7Fh).....	22
3.4 数据存储器-高 128 字节(80h 到 FFh).....	22
3.5 存储器-扩展的 256 字节(\$00 到 \$FF).....	22
4. CPU 结构.....	23
4.1 累加器.....	23
4.2 B 寄存器.....	23
4.3 程序状态字.....	24
4.4 堆栈指针.....	24
4.5 数据指针.....	24
4.6 数据指针 1.....	25
4.7 时钟控制寄存器.....	25
4.8 接口控制寄存器.....	26
5. GPIO 管脚型态.....	27
5.1 寄存器设置.....	27
5.2 烧录软件设置.....	28
6. 定时器 0 和定时器 1.....	29
6.1 定时器/计数器模式控制寄存器(TMOD).....	29
6.2 定时器/计数器控制寄存器(TCON).....	30
6.3 定时器输入频率控制寄存器.....	30
6.4 模式 0 (13 位定时/计数).....	31
6.5 模式 1 (16 位定时/计数).....	31
6.6 模式 2 (8 位自动重载定时/计数).....	33
6.7 模式 3 (两个独立 8 位定时/计数(仅定时器 0)).....	33
7. 定时器 2 以及捕捉/比较单元.....	34

7.1	定时器 2 功能	36
7.1.1	定时器模式	36
7.1.2	外部信号计数模式	37
7.1.3	外部信号定时器模式	37
7.1.4	定时器 2 的重载	38
7.2	比较功能	38
7.2.1	比较模式 0	38
7.2.2	比较模式 1	39
7.3	捕获功能	40
7.3.1	捕捉模式 0	40
7.3.2	捕捉模式 1	40
8.	串行接口	41
8.1	串行接口由以下 4 种模式可以设置	42
8.1.1	模式 0	42
8.1.2	模式 1	43
8.1.3	模式 2	43
8.1.4	模式 3	43
8.2	串行接口的多重机通讯	44
8.3	输入频率控制寄存器	44
8.4	波特率发生器	44
8.4.1	串行接口的模式 1 和 3	44
9.	看门狗定时器	46
10.	中断	50
10.1	优先权配置	54
11.	电源管理单元	56
11.1	待机模式(空闲模式)	56
11.2	停止模式	56
12.	脉宽调制器(PWM)	57
13.	IIC 功能	60
14.	SPI 功能	65
15.	KBI –键盘接口	70
16.	LVI –低压侦测中断	73
17.	12 位模拟数字转换器(ADC)	75
18.	在系统编程(INTERNAL ISP)	81
18.1	ISP 服务程序	81
18.2	锁定位(N)	81
18.3	对 ISP 服务程序编程	82
18.4	启动 ISP 服务程序	82
18.5	ISP 寄存器– TAKEY, IFCON, ISPFAL, ISPFD and ISPFC	83
19.	比较器(COMPARATOR)	88
	工作环境	91
	DC 电气特性	91
	LVI& LVR 电气特性	92
	ADC 电气特性	92
	COMPARATOR 电气特性	92

产品目录

OB38R08A1W20,
 OB38R08A1W16,
 OB38R08A1W14,
 OB38R08A1W10,

描述

原来的 8052 有 12 时钟结构,一个机器周期需要 12 个时钟,大多数指令是一个或两个机器周期.因此,除了乘和除指令,8052 的每个指令使用 12 或 24 个时钟,此外,8052 中的每个周期用了两个记忆提取.在许多情况下,第二个是假的提取,和额外的时钟被浪费了

该 OB38R08A1 是一个快速的单芯片 8 位微控制器内核.这是一个全功能的 8 位嵌入式控制器,执行所有 ASM51 指令,具有与 MCS - 51 相同的指令设置

订货信息

OB38R08A1ihhkL

YWW

i: 工艺标志{ W = 2.4V ~ 5.5V}

hh: 封装脚位

k: 封装形式后缀{as table below }

L: 无铅标志

{无文字即含铅, "P" 即无铅}

Y: 年

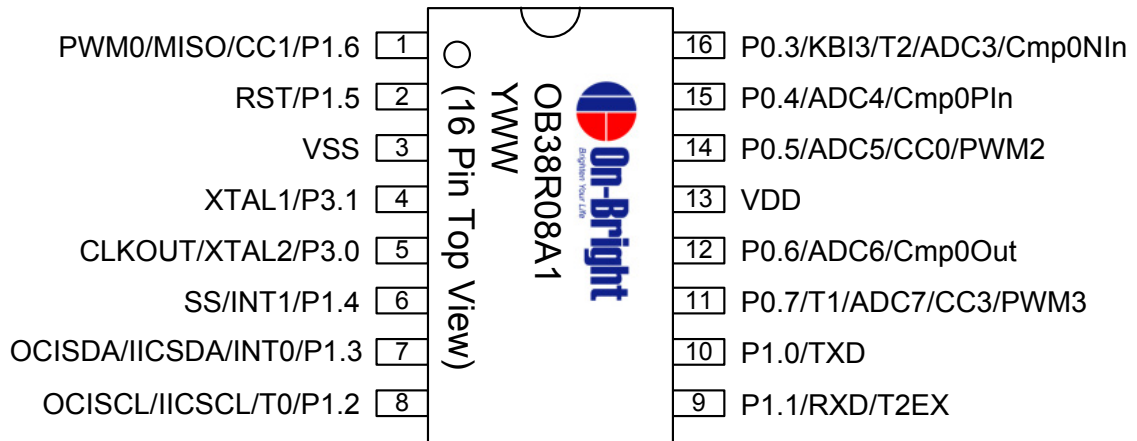
WW: 周(01~52)

特征

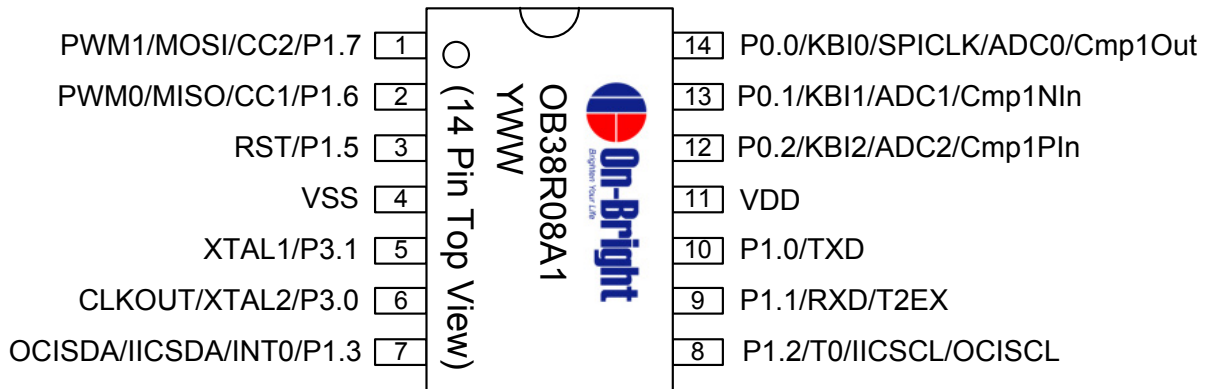
- 工作电压: 2.4V ~ 5.5V
- 高速 1T 架构,最高可达 16MHz
- 1~8T 模式可使用软件编程
- 指令设置兼容 MCS-51
- 内置 16MHz RC 振荡器,及可程序化的分频器
- 8KB 字节的片上可编程存储器中程序存储器
- 512B 字节的标准的 8052 RAM
- 双 16-bit 数据指针 (DPTR0&DPTR1)
- 一个全双工通信的串行接口.附加波特率产生器
- 三个 16-bit 的定时器/计数器 (计时器 0, 1, 2)
- 8 ~18 GPIOs(10L~ 20L 封装脚位), GPIOs 可选择四种型态(准双向口、推挽、开漏、只输入),默认准双向口(上拉)
- 具有四级优先权的外部中断 0&外部中断 1
- 可编程的看门狗定时器 (WDT)
- 一个 IIC 接口 (主/从机模式)
- 一个 SPI 接口 (主/从机模式)
- 4 路 10bit 脉宽调制(PWM)
- 4 路 16bit 比较(PWM)/捕获/重载功能
- 8 路 12bit 模拟数字转换 (ADC)
- 片上内建比较器
- 片上存储器支持 ISP/IAP/ICP 及 EEPROM 功能
- ISP 服务程序存储空间设置为 N*128 byte (N=0 to 8)
- 片上在线仿真功能(ICE)及片上在线调试功能(OCD)
- 键盘接口(KBI) 共 4 个的中断源
- 低电压中断/低电压复位(LVIL/LVR)
- 管脚 ESD 性能超过 4KV
- 增强用户代码保护
- 电源管理单元空闲及掉电模式

Postfix	Package
N	PDIP (300 mil)
S	SOP (300 mil)
B	QFN(3mmx3mm)
O	SOP (150 mil)
G	SSOP (150 mil)
M	MSOP (118 mil)

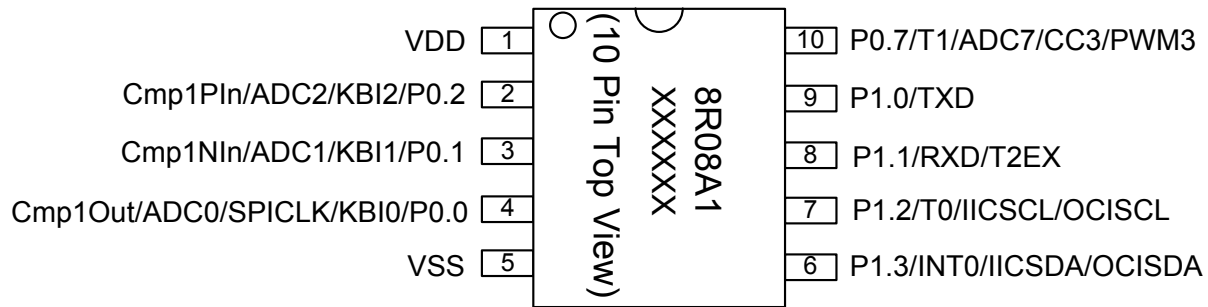
16 Pin SOP



14 Pin SOP



10 Pin MSOP



附注：出厂默认值注意事项

- (1) 管脚RST/P1.5 于出厂时设置为一般双向I/O(P1.5)脚，若使用者需切换为复位脚可于烧录时将此管脚定义为RESET脚
- (2) 为避免偶然的情况下误入ISP烧录状态(参考第 18.4 单元)，在上电时请确保没有连续的脉冲信号在管脚RXD P1.1 及管脚P1.6 必须置高
- (3) OSI_SDA/P1.3 及OCI_SCL/P1.2 于复位期间为ICP烧录功能管脚，复位完成后切换成双向I/O.

管脚描述

20L	20L QFN	16L	14L	10L	Symbol	I/O	Description
1	9	-	14	4	P0.0/KBI0/SPICLK/ADC0/CMP1Out	I/O	P0口的位0 & 键盘接口中断0 & SPI 接口时钟&模数转换通道0 &比较器1输出
2	10	-	1	-	P1.7/CC2/MOSI/PWM1	I/O	P1口的位7 & 计时器2及捕获/比较单元通道2& SPI 接口串行数据线主输出或从输入口 & 宽脉调制通道1
3	11	1	2	-	P1.6/CC1/MISO/PWM0	I/O	P1口的位6 & 计时器2及捕获/比较单元通道1& SPI 接口串行数据线主输入或从输出口 & 宽脉调制通道0
4	12	2	3	-	P1.5/RST	I/O	P1口的位5 & 复位
5	13	3	4	5	VSS	I	供电电源地
6	14	4	5	-	P3.1/XTAL1	I/O	P3口的位1 & 晶振输入&外部振荡器输入
7	15	5	6	-	P3.0/XTAL2/CLKOUT	I/O	P3口的位0 & 晶振输出&时钟输出
8	16	6	-	-	P1.4/INT1/SS	I/O	P1口的位4 & 外部中断1& SPI 接口从机跳线
9	17	7	7	6	P1.3/INT0/IICSDA/OCISDA	I/O	P1口的位3 & 外部中断0 & IIC 串行数据线 & ICE和 ICP 功能的指令及数据输入
10	18	8	8	7	P1.2/T0/IIC SCL/OCISCL	I/O	P1口的位2 & 计时器0外部输入& IIC 串行时钟线 & ICE和 ICP 功能的时钟输入
11	19	9	9	8	P1.1/RXD/T2EX	I/O	P1口的位1 & 串行接口通道接收/发送数据 & 计时器2捕捉触发及捕获触发器
12	20	10	10	9	P1.0/TXD	I/O	P1口的位0 & 串行接口通道数据传输或接收模式0时钟
13	1	11	-	10	P0.7/T1/ADC7/CC3/PWM3	I/O	P0口的位7 & 计时器1外部输入&模数转换通道7 & 计时器2及捕获/比较单元通道3 & 宽脉调制通3
14	2	12	-	-	P0.6/ADC6/CMP0Out	I/O	P0口的位6 & 模数转换通道6 & 比较器0输出
15	3	13	11	1	VDD	I	数位电源电压
16	4	14	-	-	P0.5/ADC5/CC0/PWM2	I/O	P0口的位5 & 模数转换通道5 & 计时器2及捕获/比较单元通道0 & 宽脉调制通道2
17	5	15	-	-	P0.4/ADC4/CMP0PIn	I/O	P0口的位4 & 模数转换通道4 & 比较器0非反向输入
18	6	16	-	-	P0.3/KBI3/T2/ADC3/CMP0NIn	I/O	P0口的位3 & 键盘接口中断3 & 计时器2外部输入时钟 & 模数转换通道3 & 比较器0反向输入
19	7	-	12	2	P0.2/KBI2/ADC2/ CMP1PIn	I/O	P0口的位2 & 键盘接口中断2 & 模数转换通道2 & 比较器1非反向输入
20	8	-	13	3	P0.1/KBI1/ADC1/ CMP1NIn	I/O	P0口的位0 & 键盘接口中断1 & 模数转换通道1 & 比较器1反向输入

特殊功能寄存器(SFR)

特殊功能寄存器分布图如下所示:

Hex\Bin	X000	X001	X010	X011	X100	X101	X110	X111	Bin/Hex
F8	IICS	IICCTL	IICA1	IICA2	IICRWD	IICEBT	CMP0CON	CMP1CON	FF
F0	B	SPIC1	SPIC2	SPITXD	SPIRXD	SPIS	OPPIN	TAKEY	F7
E8				ISPFDH			INTDEG	ADCSH	EF
E0	ACC	ISPF AH	ISPF AL	ISPF DL	ISPF C	ENHIT	LVC	SWRES	E7
D8		PFC ON	P3M0	P3M1					DF
D0	PSW	CCEN2	P0M0	P0M1	P1M0	P1M1			D7
C8	T2CON	CCCON	CRCL	CRCH	TL2	TH2	PWMMDH	PWMMDL	CF
C0	IRCON	CCEN	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3	C7
B8	IEN1	IP1	SRELH		PWMD0H	PWMD0L	PWMD1H	PWMD1L	BF
B0	P3	PWMD2H	PWMD2L	PWMD3H	PWMD3L	PWMC	WDTC	WDTK	B7
A8	IEN0	IP0	SRELL	ADCC1	ADCC2	ADCDH	ADC DL	ADCCS	AF
A0		RSTS							A7
98	SCON	SBUF	IEN2						9F
90	P1	AUX		KBLS	KBE	KBF	KBD	IRCON2	97
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	IFCON	8F
80	P0	SP	DPL	DPH	DPL1	DPH1		PCON	87
Hex\Bin	X000	X001	X010	X011	X100	X101	X110	X111	Bin/Hex

注: OB38R08A1 特殊功能寄存器的重置值在下表描述

寄存器	地址	重置值	描述
SYSTEM			
SP	81h	07h	Stack Pointer
ACC	E0h	00h	Accumulator
PSW	D0h	00h	Program Status Word
B	F0h	00h	B Register
DPL	82h	00h	Data Pointer 0 low byte
DPH	83h	00h	Data Pointer 0 high byte
DPL1	84h	00h	Data Pointer 1 low byte
DPH1	85h	00h	Data Pointer 1 high byte
AUX	91h	00h	Auxiliary register
PCON	87h	00h	Power Control
CKCON	8Eh	10h	Clock control register
INTERRUPT & PRIORITY			
IRCON	C0h	00h	Interrupt Request Control Register
IRCON2	97h	00h	Interrupt Request Control Register 2
IEN0	A8h	00h	Interrupt Enable Register 0

寄存器	地址	重置值	描述
IEN1	B8h	00h	Interrupt Enable Register 1
IEN2	9Ah	00h	Interrupt Enable Register 2
IP0	A9h	00h	Interrupt Priority Register 0
IP1	B9h	00h	Interrupt Priority Register 1
ENHIT	E5h	07h	ENHance Interrupt Type Register
INTDEG	EEh	00h	External Interrupt Deglitch Register
KBI			
KBLS	93h	00h	Keyboard level selector Register
KBE	94h	00h	Keyboard input enable Register
KBF	95h	00h	Keyboard interrupt flag Register
KBD	96h	00h	Keyboard interface De-bounce control register
UART			
PCON	87h	00h	Power Control
AUX	91h	00h	Auxiliary register
SCON	98h	00h	Serial Port, Control Register
SBUF	99h	00h	Serial Port, Data Buffer
SRELL	AAh	00h	Serial Port, Reload Register, low byte
SRELH	BAh	00h	Serial Port, Reload Register, high byte
PFCON	D9h	00h	Peripheral Frequency control register
ADC			
ADCC1	ABh	00h	SADC Control 1 Register
ADCC2	ACH	0Fh	SADC Control 2 Register
ADCDH	ADh	00h	SADC data high byte
ADC DL	A Eh	00h	SADC data low byte
ADCCS	AFh	00h	SADC clock select
ADCSH	EFh	00h	SADC Sample and Hold Time
WDT			
RSTS	A1h	00h	Reset status register
WDTC	B6h	04h	Watchdog timer control register
WDTK	B7h	00h	Watchdog timer refresh key.
TAKEY	F7h	00h	Time Access Key register
PWM			
PWMC	B5h	00h	PWM control register
PWMD0H	BCh	00h	PWM channel 0 data high byte
PWMD0L	BDh	00h	PWM channel 0 data low byte
PWMD1H	BEh	00h	PWM channel 1 data high byte
PWMD1L	BFh	00h	PWM channel 1 data low byte
PWMD2H	B1h	00h	PWM channel 2 data high byte
PWMD2L	B2h	00h	PWM channel 2 data low byte
PWMD3H	B3h	00h	PWM channel 3 data high byte
PWMD3L	B4h	00h	PWM channel 3 data low byte

寄存器	地址	重置值	描述
PWMMDH	CEh	00h	PWM Max Data Register, high byte.
PWMDL	CFh	FFh	PWM Max Data Register, low byte.
TIMER0/TIMER1			
TCON	88h	00h	Timer/Counter Control
TMOD	89h	00h	Timer Mode Control
TL0	8Ah	00h	Timer 0, low byte
TL1	8Bh	00h	Timer 1, low byte
TH0	8Ch	00h	Timer 0, high byte
TH1	8Dh	00h	Timer 1, high byte
PFCN	D9h	00h	Peripheral Frequency control register
PCA(TIMER2)			
CCEN	C1h	00h	Compare/Capture Enable Register
CCL1	C2h	00h	Compare/Capture Register 1, low byte
CCH1	C3h	00h	Compare/Capture Register 1, high byte
CCL2	C4h	00h	Compare/Capture Register 2, low byte
CCH2	C5h	00h	Compare/Capture Register 2, high byte
CCL3	C6h	00h	Compare/Capture Register 3, low byte
CCH3	C7h	00h	Compare/Capture Register 3, high byte
T2CON	C8h	00h	Timer 2 Control
CCCON	C9h	00h	Compare/Capture Control
CRCL	CAh	00h	Compare/Reload/Capture Register, low byte
CRCH	CBh	00h	Compare/Reload/Capture Register, high byte
TL2	CCh	00h	Timer 2, low byte
TH2	CDh	00h	Timer 2, high byte
CCEN2	D1h	00h	Compare/Capture Enable 2 register
GPIO			
P0	80h	FFh	Port 0
P1	90h	FFh	Port 1
P3	B0h	FFh	Port 3
P0M0	D2h	00h	Port 0 output mode 0
P0M1	D3h	00h	Port 0 output mode 1
P1M0	D4h	00h	Port 1 output mode 0
P1M1	D5h	00h	Port 1 output mode 1
P3M0	DAh	00h	Port 3 output mode 0
P3M1	DBh	00h	Port 3 output mode 1
ISP/IAP/EEPROM			
IFCON	8Fh	00h	Interface control register
ISPFAH	E1h	FFh	ISP Address-High register
ISPFAL	E2h	FFh	ISP Address-Low register
ISPFDL	E3h	FFh	ISP Data High register
ISPFDH	E3h	FFh	ISP Data Low register

寄存器	地址	重置值	描述
ISPCF	E4h	00h	ISP control register
TAKEY	F7h	00h	Time Access Key register
LVI/LVR/SOFTRESET			
RSTS	A1h	00h	Reset status register
LVC	E6h	20h	Low voltage control register
SWRES	E7h	00h	Software Reset register
TAKEY	F7h	00h	Time Access Key register
SPI			
AUX	91h	00h	Auxiliary register
SPIC1	F1h	08h	SPI control register 1
SPIC2	F2h	00h	SPI control register 2
SPITXD	F3h	00h	SPI transmit data buffer
SPIRXD	F4h	00h	SPI receive data buffer
SPIS	F5h	40h	SPI status register
IIC			
AUX	91h	00h	Auxiliary register
IICS	F8h	00h	IIC status register
IICCTL	F9h	04h	IIC control register
IICA1	FAh	A0h	IIC channel 1 Address 1 register
IICA2	FBh	60h	IIC channel 1 Address 2 register
IICRWD	FCh	00h	IIC channel 1 Read / Write Data buffer
IICEBT	FDh	00h	IIC Enable Bus Transaction register
OPA			
OPPIN	F6h	00H	Comparator Pin Select register
CMPOCON	FEh	00h	Comparator 0 Control register
CMP1CON	FFh	00h	Comparator 1 Control register

功能描述

1. 总特征

OB38R08A1是一个8位的微处理器,它的所有功能以及特殊功能寄存器（SFR）的详细定义将在以下章节给出.

1.1 嵌入式程序存储器

可通过编程器或在线编程(ISP)将程序加载到8KB的嵌入式可编程存储器中。

1.2 IO口

OB38R08A1有3个I/O 口: Port 0, Port 1 与Port 3. Ports 0, 1 是 8位口而Port 3是2位口. 四种型态: 准双向口(标准 8051输出), 推挽, 开漏, 与仅为输入。在章节5中有详细说明。

当使用片上硬件复位机制时,复位脚可配置为P1.5。

当用户使用内部晶振作为系统时钟时, XTAL2 与 XTAL1可在烧写或用ISP时定义成IO管脚P3.0、P3.1; 当用户使用外部振荡器由XTAL1输入作为系统时钟时, 只有XTAL2可定义成IO管脚P3.0。

所有的P0~P3口都可通过转化速率来降低EMI。另一种降低EMI的方式是在禁止ALE输出, 可通过特殊功能寄存器（SFR）来进行选择控制, 在高静电环境且在人体接触模式中, 其I/O口的ESD可达4KV, 可以保证 OB38R08A1在高静电环境下的品质。

1.3 指令时钟周期选择

传统的52 系列单片机时钟周期是12T, 即12 个振荡器时钟为1个机器周期。OB38R08A1为1T~8T的微控制器, 即机器周期为1个时钟周期~8个时钟周期。换句话说, 执行一条指令可是1个时钟~8个时钟。

符号: CKCON						地址: 8Eh		
7	6	5	4	3	2	1	0	Reset
-	ITS[2:0]			-	-	CLKOUT[1:0]		10H

ITS: 指令时钟周期选择.

ITS [2:0]	指令时钟周期
000	1T 模式
001	2T 模式(默认)
010	3T 模式
011	4T 模式
100	5T 模式
101	6T 模式
110	7T 模式
111	8T 模式

默认为2T模式, 在任何时候,如CKCON [6:4]（地址为8Eh）被改变时, 每条指令并不是都能在一个机器周期内被执行的。所有指令的确切机器周期将在下一章节中给出。

1.4 时钟输出选择

在任何时候CKCON [1:0] (地址为8Eh) 被改变时, OB38R08A1能在P3.0产生时钟输出, 主系统时钟源设置为使用振荡器(时钟由晶振输入脚输入)或片内RC振荡器皆可。

CLKOUT: 时钟输出除频选择.

CKCON [1:0]	Mode.
00	GPIO(默认)
01	主系统时钟频率
10	主系统时钟频率/2
11	主系统时钟频率/4

1.5 复位

1.5.1 硬件复位功能

OB38R08A1提供了片上硬件复位机制, 片上硬件复位的时间长度可以通过编程器或ISP设置。

片上硬件复位的时间长度
25ms (default)
200ms
100ms
50ms
16ms
8ms
4ms

1.5.2 软件复位功能

OB38R08A1提供一种软件复位机制来实现整个芯片的复位.要实现软件复位,于程序中需把3个特殊值 55h,AAh和5Ah按顺序写到TAKEY寄存器来使能软件复位寄存器(SWRES)之写入. 在软件复位寄存器获得可写权后,可以对SWRES寄存器写入FFh. 硬件将解码出复位讯号它是与其它硬件复位讯号作“OR”处理. 软件复位寄存器在软件复位过程的最后会进行自复位.

符号	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
Software Reset function											
RSTS	Reset status register	A1h	-	LVRLP INTF	LVRL PF	PDRF	WDTF	SWRF	LVRF	PORF	00H
TAKEY	Time Access Key register	F7h	TAKEY [7:0]								00H
SWRES	Software Reset register	E7h	SWRES [7:0]								00H

1.5.3 Reset status

符号: RSTS								地址: A1h
7	6	5	4	3	2	1	0	Reset
-	LVRLP INTF	LVRL PF	PDRF	WDTF	SWRF	LVRF	PORF	00H

LVRLPINTF: “内部”低电压复位旗标.

当 MCU 复位信号由“内部”低电压复位产生时, LVRLPINTF 旗标将由硬件置高.此旗标需由软件清零.

LVRLPF: 低电压复位旗标.(低功率)

当 MCU 复位信号由低电压复位(低功率)产生时, LVRLPF 旗标将由硬件置高.此旗标需由软件清零.

PDRF: 复位脚复位旗标.

当 MCU 复位信号由复位脚产生时, PDRF 旗标将由硬件置高.此旗标需由软件清零.

WDTF: 看门狗定时器复位旗标.

当 MCU 复位信号由看门狗产生时, WDTF 旗标将由硬件置高.此旗标需由软件清零.

SWRF: 软件复位旗标.

当 MCU 复位信号由软件复位产生时, SWRF 旗标将由硬件置高.此旗标需由软件清零.

LVRF: 低电压复位旗标.

当 MCU 复位信号由低电压复位产生时, LVRF 旗标将由硬件置高.此旗标需由软件清零.

PORF: 上电复位旗标.

当 MCU 复位信号由上电复位产生时, PORF 旗标将由硬件置高.此旗标需由软件清零.

1.5.4 Time Access Key register (TAKEY)

符号: TAKEY								地址: F7H
7	6	5	4	3	2	1	0	Reset
TAKEY [7:0]								00H

软件复位寄存器(SWRES) 默认为只读; 软件把 3 个特殊值按顺序写到 TAKEY 寄存器来使能软件复位寄存器(SWRES)可写. 它们是

```
MOV TAKEY, #55h
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah
```

1.5.5 软件复位寄存器(SWRES)

符号: SWRES								地址: E7H
7	6	5	4	3	2	1	0	Reset
SWRES [7:0]								00H

SWRES[7:0]: 软件复位寄存器. 在软件复位过程的最后会进行自重置.

SWRES [7:0] = FFh, 产生软件复位.

SWRES [7:0] = 00h ~ FEh, 不产生复位动作.

1.5.6 Example of software reset

```

MOV TAKEY, #55h
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah ; enable SWRES write attribute
MOV SWRES, #0FFh ; software reset MCU
  
```

1.6 时钟源

默认时钟来自于内部16MhzOSC时钟信号，其时钟应用于初始化阶段，其主要的工作是确定时钟源使用的正常运行。

内部时钟源来源于不同分频的内部OSC如表 1-1所示，时钟源可在编程器或ICP中设置。

表 1-1: Selection of clock source

Clock source
external crystal (use XTAL1 and XTAL2 pins)
external crystal/2 (use XTAL1 and XTAL2 pins)
external crystal/4 (use XTAL1 and XTAL2 pins)
external crystal (only use XTAL1, the XTAL2 define as I/O)
16MHz from internal OSC
8MHz from internal OSC
4MHz from internal OSC
2MHz from internal OSC
1MHz from internal OSC

对于来源于内部OSC的频率，这将有些许误差，在应用程序中需要精确的频率时，须注意使用. 最大误差如表 1-2

表 1-2: Temperature with variance

Temperature	Max Variance
25°C	±2%

2. 指令设置

所有OB38R08A1的指令都是兼容的二进制码,且具有标准的8051一样的功能,以下表格列出了以OB38R08A1微控器为核心的指令集的周期的总结,这里的周期表示为机器周期。

表 2-1: Arithmetic operations

符号	描述	代码	字节	周期
ADD A,Rn	Add register to accumulator	28-2F	1	1
ADD A,direct	Add direct byte to accumulator	25	2	2
ADD A,@Ri	Add indirect RAM to accumulator	26-27	1	2
ADD A,#data	Add immediate data to accumulator	24	2	2
ADDC A,Rn	Add register to accumulator with carry flag	38-3F	1	1
ADDC A,direct	Add direct byte to A with carry flag	35	2	2
ADDC A,@Ri	Add indirect RAM to A with carry flag	36-37	1	2
ADDC A,#data	Add immediate data to A with carry flag	34	2	2
SUBB A,Rn	Subtract register from A with borrow	98-9F	1	1
SUBB A,direct	Subtract direct byte from A with borrow	95	2	2
SUBB A,@Ri	Subtract indirect RAM from A with borrow	96-97	1	2
SUBB A,#data	Subtract immediate data from A with borrow	94	2	2
INC A	Increment accumulator	04	1	1
INC Rn	Increment register	08-0F	1	2
INC direct	Increment direct byte	05	2	3
INC @Ri	Increment indirect RAM	06-07	1	3
INC DPTR	Increment data pointer	A3	1	1
DEC A	Decrement accumulator	14	1	1
DEC Rn	Decrement register	18-1F	1	2
DEC direct	Decrement direct byte	15	2	3
DEC @Ri	Decrement indirect RAM	16-17	1	3
MUL AB	Multiply A and B	A4	1	5
DIV	Divide A by B	84	1	5
DA A	Decimal adjust accumulator	D4	1	1

表 2-2: Logic operations

符号	描述	代码	字节	周期
ANL A,Rn	AND register to accumulator	58-5F	1	1
ANL A,direct	AND direct byte to accumulator	55	2	2
ANL A,@Ri	AND indirect RAM to accumulator	56-57	1	2
ANL A,#data	AND immediate data to accumulator	54	2	2
ANL direct,A	AND accumulator to direct byte	52	2	3
ANL direct,#data	AND immediate data to direct byte	53	3	4
ORL A,Rn	OR register to accumulator	48-4F	1	1
ORL A,direct	OR direct byte to accumulator	45	2	2
ORL A,@Ri	OR indirect RAM to accumulator	46-47	1	2
ORL A,#data	OR immediate data to accumulator	44	2	2
ORL direct,A	OR accumulator to direct byte	42	2	3
ORL direct,#data	OR immediate data to direct byte	43	3	4
XRL A,Rn	Exclusive OR register to accumulator	68-6F	1	1
XRL A,direct	Exclusive OR direct byte to accumulator	65	2	2
XRL A,@Ri	Exclusive OR indirect RAM to accumulator	66-67	1	2
XRL A,#data	Exclusive OR immediate data to accumulator	64	2	2
XRL direct,A	Exclusive OR accumulator to direct byte	62	2	3
XRL direct,#data	Exclusive OR immediate data to direct byte	63	3	4
CLR A	Clear accumulator	E4	1	1
CPL A	Complement accumulator	F4	1	1
RL A	Rotate accumulator left	23	1	1
RLC A	Rotate accumulator left through carry	33	1	1
RR A	Rotate accumulator right	03	1	1
RRC A	Rotate accumulator right through carry	13	1	1
SWAP A	Swap nibbles within the accumulator	C4	1	1

表 2-3: Data transfer

符号	描述	代码	字节	周期
MOV A,Rn	Move register to accumulator	E8-EF	1	1
MOV A,direct	Move direct byte to accumulator	E5	2	2
MOV A,@Ri	Move indirect RAM to accumulator	E6-E7	1	2
MOV A,#data	Move immediate data to accumulator	74	2	2
MOV Rn,A	Move accumulator to register	F8-FF	1	2
MOV Rn,direct	Move direct byte to register	A8-AF	2	4
MOV Rn,#data	Move immediate data to register	78-7F	2	2
MOV direct,A	Move accumulator to direct byte	F5	2	3
MOV direct,Rn	Move register to direct byte	88-8F	2	3
MOV direct1,direct2	Move direct byte to direct byte	85	3	4
MOV direct,@Ri	Move indirect RAM to direct byte	86-87	2	4
MOV direct,#data	Move immediate data to direct byte	75	3	3
MOV @Ri,A	Move accumulator to indirect RAM	F6-F7	1	3
MOV @Ri,direct	Move direct byte to indirect RAM	A6-A7	2	5
MOV @Ri,#data	Move immediate data to indirect RAM	76-77	2	3
MOV DPTR,#data16	Load data pointer with a 16-bit constant	90	3	3
MOVC A,@A+DPTR	Move code byte relative to DPTR to accumulator	93	1	3
MOVC A,@A+PC	Move code byte relative to PC to accumulator	83	1	3
MOVX A,@Ri	Move external RAM (8-bit addr.) to A	E2-E3	1	3
MOVX A,@DPTR	Move external RAM (16-bit addr.) to A	E0	1	3
MOVX @Ri,A	Move A to external RAM (8-bit addr.)	F2-F3	1	4
MOVX @DPTR,A	Move A to external RAM (16-bit addr.)	F0	1	4
PUSH direct	Push direct byte onto stack	C0	2	4
POP direct	Pop direct byte from stack	D0	2	3
XCH A,Rn	Exchange register with accumulator	C8-CF	1	2
XCH A,direct	Exchange direct byte with accumulator	C5	2	3
XCH A,@Ri	Exchange indirect RAM with accumulator	C6-C7	1	3
XCHD A,@Ri	Exchange low-order nibble indir. RAM with A	D6-D7	1	3

表 2-4: Program branches

符号	描述	代码	字节	周期
ACALL addr11	Absolute subroutine call	xxx11	2	6
LCALL addr16	Long subroutine call	12	3	6
RET	from subroutine	22	1	4
RETI	from interrupt	32	1	4
AJMP addr11	Absolute jump	xxx01	2	3
LJMP addr16	Long iump	02	3	4
SJMP rel	Short jump (relative addr.)	80	2	3
JMP @A+DPTR	Jump indirect relative to the DPTR	73	1	2
JZ rel	Jump if accumulator is zero	60	2	3
JNZ rel	Jump if accumulator is not zero	70	2	3
JC rel	Jump if carry flag is set	40	2	3
JNC	Jump if carry flag is not set	50	2	3
JB bit,rel	Jump if direct bit is set	20	3	4
JNB bit,rel	Jump if direct bit is not set	30	3	4
JBC bit,direct rel	Jump if direct bit is set and clear bit	10	3	4
CJNE A,direct rel	Compare direct byte to A and jump if not equal	B5	3	4
CJNE A,#data rel	Compare immediate to A and jump if not equal	B4	3	4
CJNE Rn,#data rel	Compare immed. to reg. and jump if not equal	B8-BF	3	4
CJNE @Ri,#data rel	Compare immed. to ind. and jump if not equal	B6-B7	3	4
DJNZ Rn,rel	Decrement register and jump if not zero	D8-DF	2	3
DJNZ direct,rel	Decrement direct byte and jump if not zero	D5	3	4
NOP	No operation	00	1	1

表 2-5: Boolean manipulation

符号	描述	代码	字节	周期
CLR C	Clear carry flag	C3	1	1
CLR bit	Clear direct bit	C2	2	3
SETB C	Set carry flag	D3	1	1
SETB bit	Set direct bit	D2	2	3
CPL C	Complement carry flag	B3	1	1
CPL bit	Complement direct bit	B2	2	3
ANL C,bit	AND direct bit to carry flag	82	2	2
ANL C,/bit	AND complement of direct bit to carry	B0	2	2
ORL C,bit	OR direct bit to carry flag	72	2	2
ORL C,/bit	OR complement of direct bit to carry	A0	2	2
MOV C,bit	Move direct bit to carry flag	A2	2	2
MOV bit,C	Move carry flag to direct bit	92	2	3

3. 存储器结构

OB38R08A1存储器结构和通用的8051结构相同，它们是作为程序存储器的8KB的嵌入式可编程存储器中。

3.1 程序存储器

OB38R08A1有8KB的嵌入式可编程存储器中,如以下图 3-1,可做为通用的程序存储.其它还包括最大为1K的特定ISP服务程序存储空间. 这8K的地址从\$0000到\$1FFF. ISP服务程序的地址从\$1C00到\$1FFF. ISP服务程序空间可以被分割N块128字节(N=0到8). 当N=0时,意味着没有ISP复位程序空间可用. 全部的8KB字节空间都被用来做程序存储. 当N=1时,意味着地址\$1F80到 \$1FFF保留为ISP服务程序空间. 当N=2意味着内存地址\$1F00到\$1FFF保留为ISP服务程序空间等等... 数值N可以通过编程器设置.

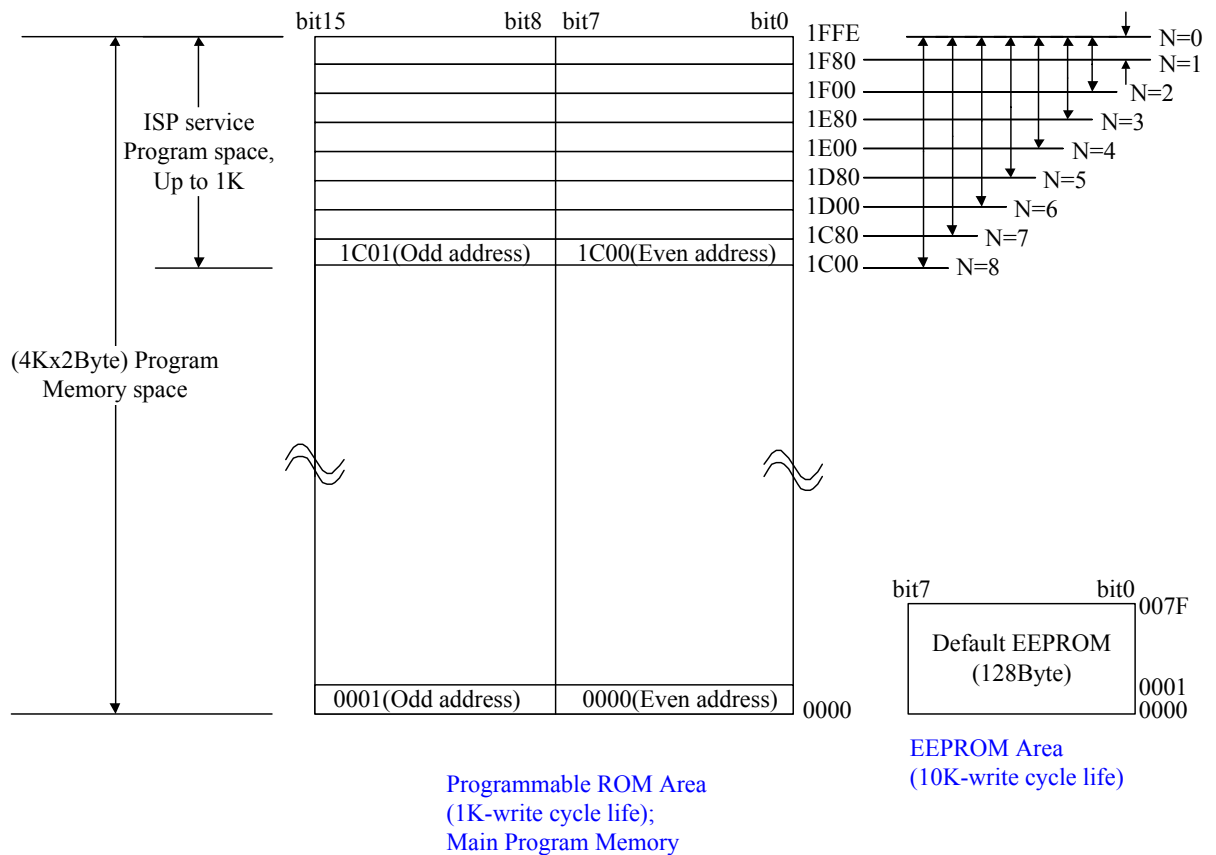


图 3-1: OB38R08A1 ROM

3.2 数据存储

OB38R08A1具有512B Bytes的片上SRAM, 如以下 图 3-2, 其中256 Bytes和通用的8052内部存储器结构一样。

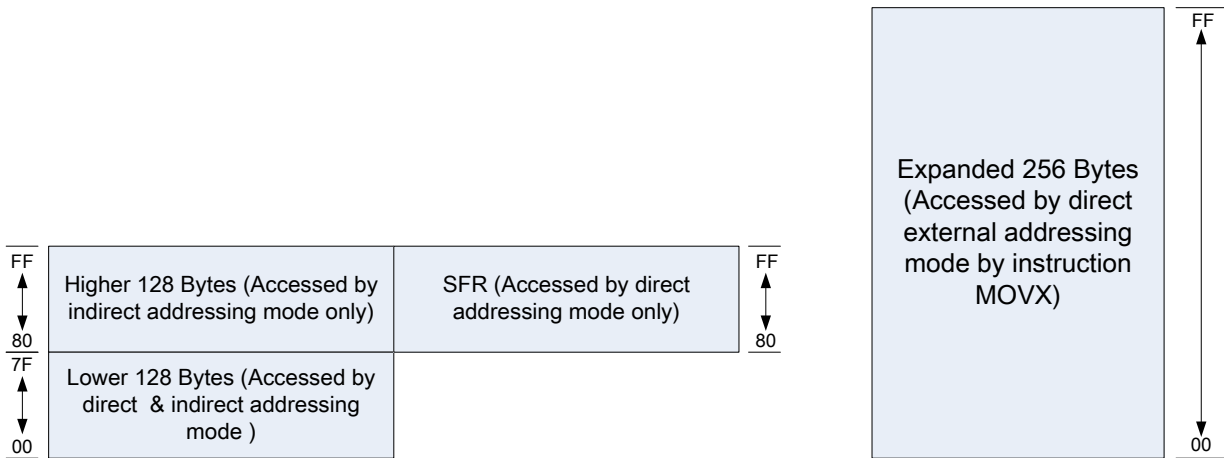


图 3-2: RAM architecture

3.3 数据内存-低 128 字节(00h到 7Fh)

数据存储器从 00h 到 FFh 的地址和在 8052 中的定义是一样的。
 00h 到 7Fh 的地址可通过直接或者间接寻址方式访问。
 00h 到 1Fh 是寄存器的空间,
 20h 到 2Fh 是位寻址空间,
 30h 到 7Fh 是通用的数据存储区。

3.4 数据存储器-高 128 字节(80h到FFh)

80h 到 FFh 的地址仅仅能从间接寻址的方式访问,它是一个数据区。

3.5 存储器-扩展的 256 字节(\$00 到 \$FF)

从00h到FFh是片内扩展的SRAM区域,共256字节, 该空间地址只能通过外部直接寻址的方式进行访问。(利用MOVX指令)

4. CPU结构

OB38R08A1结构由以下四部分组成:

- (1) 控制单元
- (2) 算法-逻辑单元
- (3) 存储器控制单元
- (4) RAM和SFR控制单元

OB38R08A1结构允许接受来自程序存储器的指令并与RAM或SFR做数据处理，以下各段详细叙述了主要功能寄存器。

符号	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
8051 Core											
ACC	Accumulator	E0h	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00H
B	B register	F0h	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00H
PSW	Program status word	D0h	CY	AC	F0	RS[1:0]		OV	PSW.1	P	00H
SP	Stack Pointer	81h	SP[7:0]								07H
DPL	Data pointer low 0	82h	DPL[7:0]								00H
DPH	Data pointer high 0	83h	DPH[7:0]								00H
DPL1	Data pointer low 0	84h	DPL1[7:0]								00H
DPH1	Data pointer high 0	85h	DPH1[7:0]								00H
AUX	Auxiliary register	91h	BRGS	-	-	-	-	-	-	DPS	00H
CKCON	Clock control register	8Eh	-	ITS[2:0]					CLKOUT[1:0]		10H
IFCON	Interface control register	8Fh	-	CDPR	-	-	-	-	-	ISPE	00H

4.1 累加器

ACC是一个累加器，大部分单操作指令的一个操作数取自累加器。

符号: ACC	地址: E0h
7 6 5 4 3 2 1 0 Reset	
ACC.7	ACC.6
ACC.5	ACC.4
ACC.3	ACC.2
ACC.1	ACC.0
00h	

ACC[7:0]: The A (or ACC) register is the standard 8052 accumulator.

4.2 B 寄存器

B寄存器被用于乘法或除法指令，也可作为一般寄存器以存储临时数据。

符号: B	地址: F0h
7 6 5 4 3 2 1 0 Reset	
B.7	B.6
B.5	B.4
B.3	B.2
B.1	B.0
00h	

B[7:0]: The B register is the standard 8052 register that serves as a second accumulator.

4.3 程序状态字

符号: PSW							地址: D0h	
7	6	5	4	3	2	1	0	Reset
CY	AC	F0	RS [1:0]	OV	F1	P		00h

CY: 进位标志位

AC: 为 BCD 辅助进位标志位

F0: 用户设置标志位 0

RS[1:0]	Bank Selected	Location
00	Bank 0	00h – 07h
01	Bank 1	08h – 0Fh
10	Bank 2	10h – 17h
11	Bank 3	18h – 1Fh

OV: 溢出标志位

F1: 用户设置标志位 1

P: 奇偶校验位, 受硬件影响, 显示累加器中的奇偶的 1 位, 即奇偶校验

4.4 堆栈指针

堆栈指针是一个1字节的寄存器,在复位后初始化为07h。此寄存器在执行PUSH和CALL指令之前增值,使得堆栈指针在08h开始执行

符号: SP							地址: 81h	
7	6	5	4	3	2	1	0	Reset
SP [7:0]								07h

SP[7:0]: 堆栈指针储存了暂时寄存器的地址,该地址是堆栈指针的起始位置.换言之,它总是指向堆栈指针的顶端。

4.5 数据指针

数据指针为2字节.低位为DPL.高位为DPH.它可以作为一个2字节的寄存器(MOV DPTR,#data16)来使用,或者作为两个寄存器(例如,MOV DPL,#data8),它通常被用作是运行外部程序或者是数据空间(如,MOVC A,@ A+DPTR 或者各自的MOV A,@ DPTR).

符号: DPL							地址: 82h	
7	6	5	4	3	2	1	0	Reset
DPL [7:0]								00h

DPL[7:0]: Data pointer Low 0

符号: DPH							地址: 83h	
7	6	5	4	3	2	1	0	Reset
DPH [7:0]								00h

DPH [7:0]: Data pointer High 0

4.6 数据指针 1

双数据指针加速了块状数据的运行速度.标准的DPTR 是一个16位的寄存器,是用来寻址外部空间或者外部存储器. 在OB38R08A1中, 标准的数据指针称为DPTR,第二个数据指针称为DPTR1.选择位用来选择使用的数据指针.数据指针的选择位位于AUX寄存器的LSB(DPS).

用户通过触发AUX寄存器中的LSB位进行切换. 所有有关的DPTR指令都将为当前选用的DPTR所操作。

符号: DPL1							地址: 84h	
7	6	5	4	3	2	1	0	Reset
DPL1 [7:0]								00h

DPL1[7:0]: Data pointer Low 1

符号: DPH1							地址: 85h	
7	6	5	4	3	2	1	0	Reset
DPH1 [7:0]								00h

DPH1[7:0]: Data pointer High 1

符号: AUX							地址: 91h	
7	6	5	4	3	2	1	0	Reset
BRGS	-	-	-	-	-	-	DPS	00H

DPS: 数据指针选择位.

DPS = 1 选择数据指针 1.

4.7 时钟控制寄存器

符号: CKCON							地址: 8Eh	
7	6	5	4	3	2	1	0	Reset
-	ITS[2:0]		-	-	CLKOUT[1:0]			10H

ITS[2:0]: 指令时钟周期选择.

ITS [2:0]	指令时钟周期
000	1T 模式
001	2T 模式(默认)
010	3T 模式
011	4T 模式
100	5T 模式
101	6T 模式
110	7T 模式
111	8T 模式

CLKOUT: 时钟输出除频选择.

CKCON [1:0]	Mode.
00	GPIO(默认)
01	主系统时钟频率
10	主系统时钟频率/2
11	主系统时钟频率/4

主系统时钟源设置为使用振荡器(时钟由晶振输入脚输入)或片内RC振荡器皆可

4.8 接口控制寄存器

符号: IFCON							地址: 8Fh	
7	6	5	4	3	2	1	0	Reset
-	CDPR	-	-	-	-	-	ISPE	00H

CDPR: 程序码已加密指示位(仅读)

ISPE: ISP 功能使能位

ISPE = 1, 允许使用 ISP 功能

ISPE = 0, 禁止使用 ISP 功能

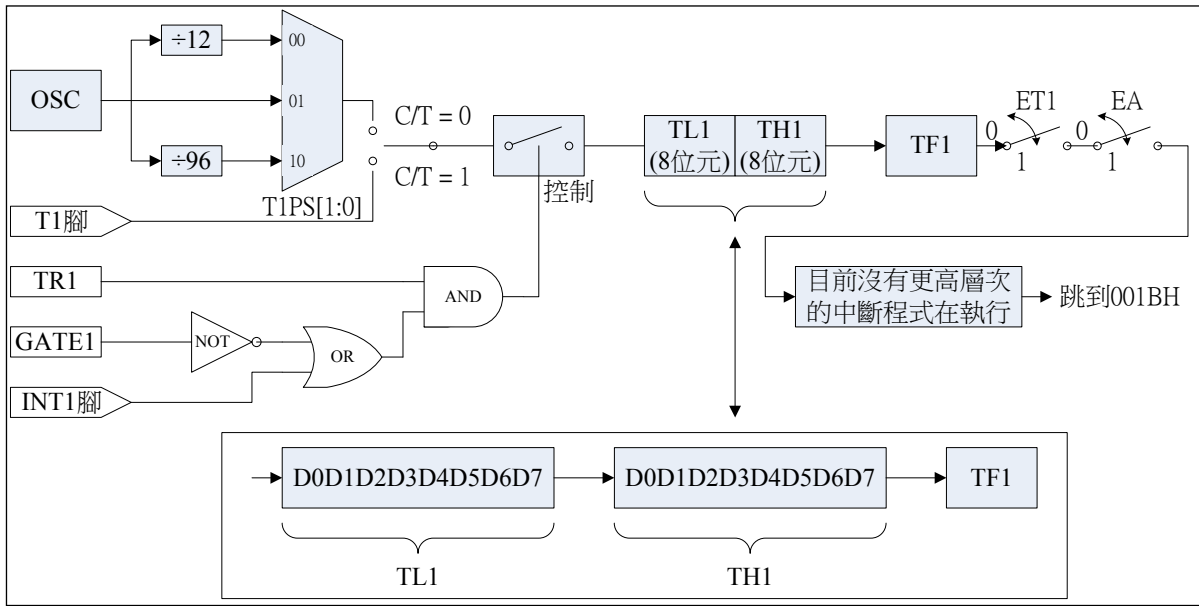


图 6-2: 模式 1 -16 位定时器/计数器操作

6.6 模式 2 (8 位自動重載定時/計數)

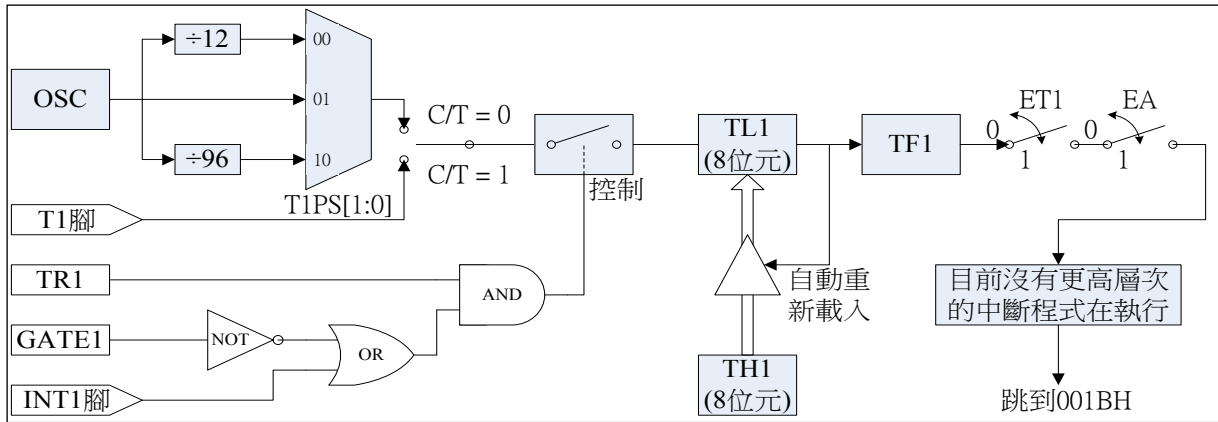


图 6-3: 模式 2 - 8 位自動重載定時/計數

6.7 模式 3 (兩個獨立 8 位定時/計數(僅定時器 0))

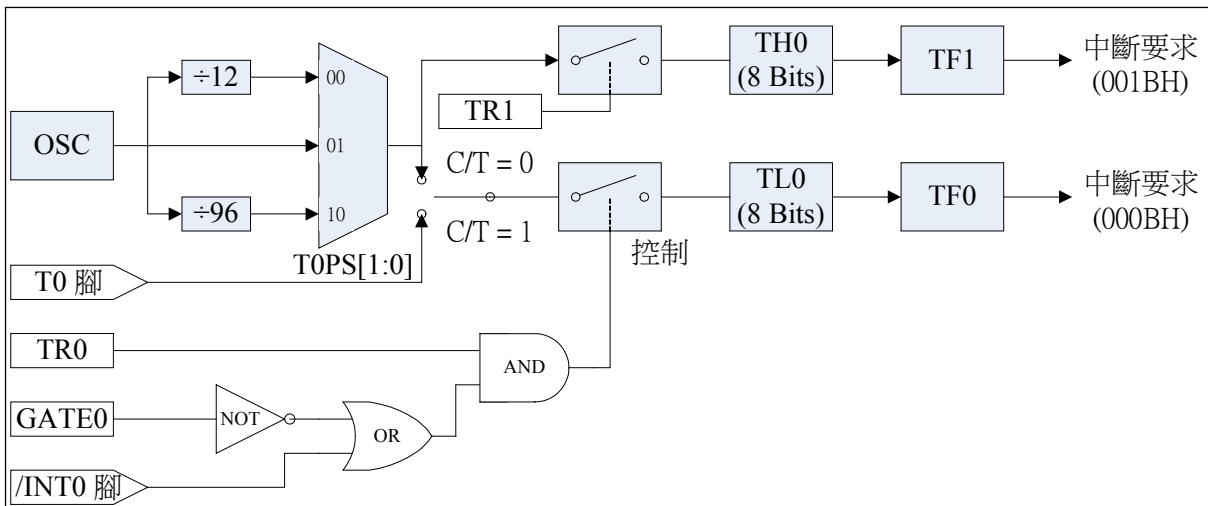


图 6-4: 模式 3 - 兩個獨立 8 位定時/計數(僅定時器 0)

7. 定时器2 以及捕捉/比较单元

定时器2不仅仅是一个16位的定时器，也是一个带有4个通道比较,捕获及重载功能，这是非常相似在其它一些微控制器的可编程计数器阵列（PCA），除脉冲宽度调制（PWM）

符号	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
Timer 2 and Capture Compare Unit											
T2CON	Timer 2 control	C8h	T2PS[2:0]			T2R[1:0]		-	T2I[1:0]		00H
CCCON	Compare/Capture Control	C9h	CCI3	CCI2	CCI1	CCI0	CCF3	CCF2	CCF1	CCF0	00H
CCEN	Compare/Capture Enable register	C1h	-	COCAM1[2:0]			-	COCAM0[2:0]		00H	
CCEN2	Compare/Capture Enable 2 register	D1h	-	COCAM3[2:0]			-	COCAM2[2:0]		00H	
TL2	Timer 2, low byte	CCh	TL2[7:0]								00H
TH2	Timer 2, high byte	CDh	TH2[7:0]								00H
CRCL	Compare/Reload/Capture register, low byte	CAh	CRCL[7:0]								00H
CRCH	Compare/Reload/Capture register, high byte	CBh	CRCH[7:0]								00H
CCL1	Compare/Capture register 1, low byte	C2h	CCL1[7:0]								00H
CCH1	Compare/Capture register 1, high byte	C3h	CCH1[7:0]								00H
CCL2	Compare/Capture register 2, low byte	C4h	CCL2[7:0]								00H
CCH2	Compare/Capture register 2, high byte	C5h	CCH2[7:0]								00H
CCL3	Compare/Capture register 3, low byte	C6h	CCL3[7:0]								00H
CCH3	Compare/Capture register 3, high byte	C7h	CCH3[7:0]								00H

符号: **T2CON**

地址: **C8h**

7	6	5	4	3	2	1	0	Reset
T2PS[2:0]			T2R[1:0]		-	T2I[1:0]		00H

T2PS[2:0]: 分频器选择位:

T2PS = 000 –定时器 2 的时钟为振荡频率.

T2PS = 001 –定时器 2 的时钟为振荡频率的 1/2.

T2PS = 010 –定时器 2 的时钟为振荡频率的 1/4.

T2PS = 011 –定时器 2 的时钟为振荡频率的 1/6.

T2PS = 100 –定时器 2 的时钟为振荡频率的 1/8.
 T2PS = 101 –定时器 2 的时钟为振荡频率的 1/12.
 T2PS = 110 –定时器 2 的时钟为振荡频率的 1/24.

T2R[1:0]: 定时器 2 重载模式选择位

T2R[1:0] = 0X –重载无效
 T2R[1:0] = 10 –模式 0:自动重载
 T2R[1:0] = 11 –模式 1:依 T2EX 脚下降缘重载

T2I[1:0]: 定时器 2 输入选择位

T2I[1:0] = 00 –定时器 2 计数停止
 T2I[1:0] = 01 –输入频率依 T2PS[2:0]分频器选择
 T2I[1:0] = 10 –定时器 2 在 T2 脚的外部信号递增
 T2I[1:0] = 11 –门控定时器 2 可为内部时钟输入

符号: CCCON								地址: C9h	
7	6	5	4	3	2	1	0	Reset	
CCI3	CCI2	CCI1	CCI0	CCF3	CCF2	CCF1	CCF0	00H	

CCI3: 捕捉/比较信道 3 中断功能控制位.

“1” 中断功能使能.

CCI2: 捕捉/比较信道 2 中断功能控制位.

“1” 中断功能使能.

CCI1: 捕捉/比较信道 1 中断功能控制位.

“1” 中断功能使能.

CCI0: 捕捉/比较信道 0 中断功能控制位.

“1” 中断功能使能.

CCF3: 捕捉/比较通道 3 中断旗标位.可由软件清零.

CCF2: 捕捉/比较通道 2 中断旗标位.可由软件清零.

CCF1: 捕捉/比较通道 1 中断旗标位.可由软件清零.

CCF0: 捕捉/比较通道 0 中断旗标位.可由软件清零.

捕捉/比较中断与定时器 2 中断共享同一个中断向量.

符号: CCEN					地址: C1h				
7	6	5	4	3	2	1	0	Reset	
-	COCAM1[2:0]			-	COCAM0[2:0]			00H	

COCAM1[2:0]: 000: 禁止比较/捕获功能

001: 比较功能启动但无输出

010: 比较功能模式 0

011: 比较功能模式 1

- 100: 捕获在 CC1 脚的上升沿
 - 101: 捕获在 CC1 脚的下降沿
 - 110: 捕获在 CC1 脚的上升沿及下降沿
 - 111: 在寄存器 CC1 中捕获写入操作
- COCAM0[2:0]: 000: 禁止比较/捕获功能
- 001: 比较功能启动但无输出
 - 010: 比较功能模式 0
 - 011: 比较功能模式 1
 - 100: 捕获在 CC0 脚的上升沿
 - 101: 捕获在 CC0 脚的下降沿
 - 110: 捕获在 CC0 脚的上升沿及下降沿
 - 111: 在寄存器 CC0 中捕获写入操作

符号: CCEN2				地址: D1h			
7	6	5	4	3	2	1	0
-	COCAM3[2:0]			-	COCAM2[2:0]		Reset
							00H

- COCAM3[2:0]: 000: 禁止比较/捕获功能
- 001: 比较功能启动但无输出
 - 010: 比较功能模式 0
 - 011: 比较功能模式 1
 - 100: 捕获在 CC3 脚的上升沿
 - 101: 捕获在 CC3 脚的下降沿
 - 110: 捕获在 CC3 脚的上升沿及下降沿
 - 111: 在寄存器 CC3 中捕获写入操作
- COCAM2[2:0]: 000: 禁止比较/捕获功能
- 001: 比较功能启动但无输出
 - 010: 比较功能模式 0
 - 011: 比较功能模式 1
 - 100: 捕获在 CC2 脚的上升沿
 - 101: 捕获在 CC2 脚的下降沿
 - 110: 捕获在 CC2 脚的上升沿及下降沿
 - 111: 在寄存器 CC2 中捕获写入操作

7.1 定时器 2 功能

定时器2既可做为定时器，又可做为计数器，或解释为门控定时器(如下解释).

7.1.1 定时器模式

如以下图 7-1，在此模式中，定时器2递增频率依分频器选择决定，而分频器则由特殊寄存器T2CON中的T2PS[2:0]位选择。

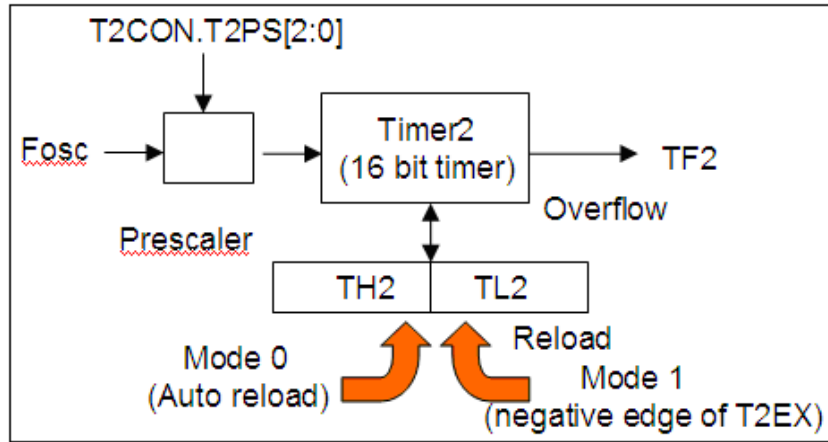


图 7-1: Timer mode and Reload mode function

7.1.2 外部信号计数模式

如以下图 7-2，在此模式中，当然外部信号T2由1到0的跳变时，定时器得到递增，T2输入在每个周期中得以采样，定时器2在跳变检测的一个周期中得到递增。

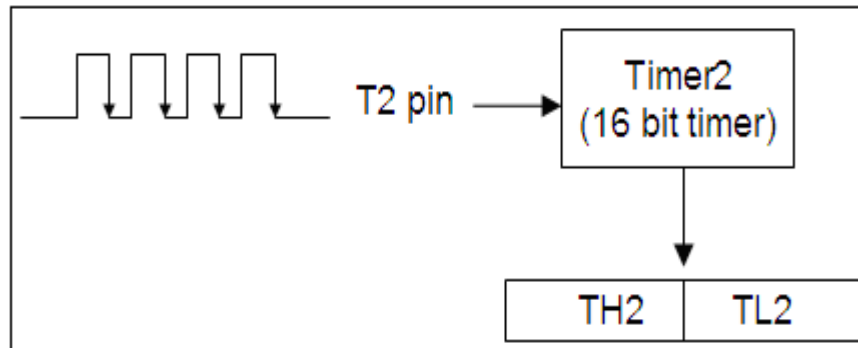


图 7-2: Event counter mode function

7.1.3 外部信号定时器模式

如以下图 7-3，在此模式中，定时器2递增的内部时钟是由外部信号T2来控制的。

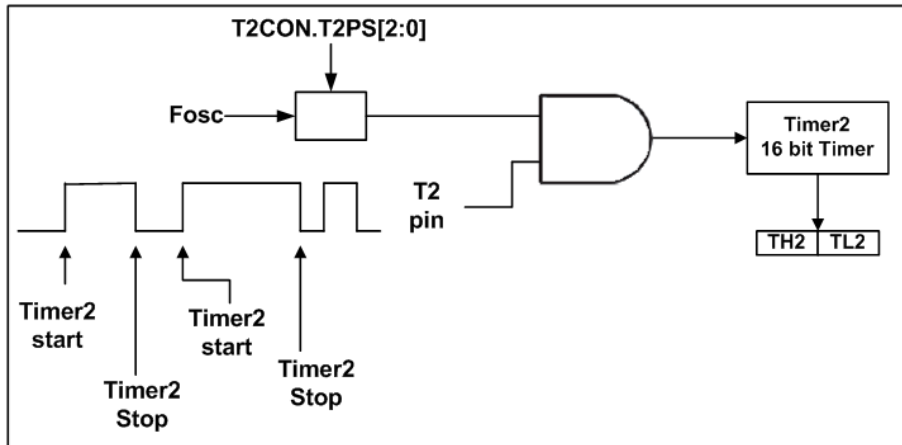


图 7-3: Gated timer mode function

7.1.4 定时器 2 的重载

重载（来自CRC寄存器的16位重载）可在以下两种模式中执行：

- 模式0：重载信号由定时器2溢出产生,即自动重载。
- 模式1：重载信号由相应的输入脚T2EX负跳变产生。

7.2 比较功能

在四个独立的比较器中，任何比较/捕捉寄存器中的值都和定时器寄存器的内容比较，其比较模式0或1由位C0CAMx来选择.在这两种比较模式中，其比较结果在同样的机器周期的P1口的得出使得其内部比较信号被激活。

7.2.1 比较模式 0

在模式0中，当定时器2的值等同与比较寄存器上的值时，其输出信号由低到高的跳变，并在定时器溢出时返回低电位。在此模式中，对端口写入无效，因为无法对来自内部总线及暂存器进行操作。

如以下图 7-4 数字阐述了比较模式0的功能

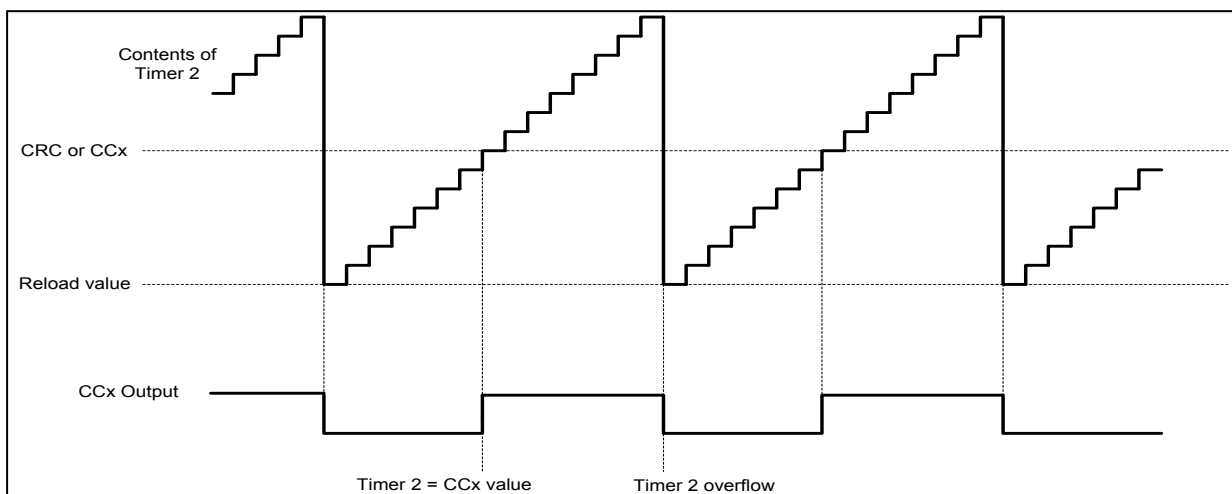


图 7-4: Compare mode 0 function

7.2.2 比较模式 1

在比较模式1中，其输出信号的跳变是由软件来决定的。定时器2的溢出不会导致输出的改变。在此模式中，两种信号的跳变都是可以控制的，如图 7-5显示了在比较模式1中的寄存器/端口结构菜单。

在比较模式1中，其值将首先被写入到“阴影寄存器”中，当比较信号被激活时，其值被传递到输出寄存器

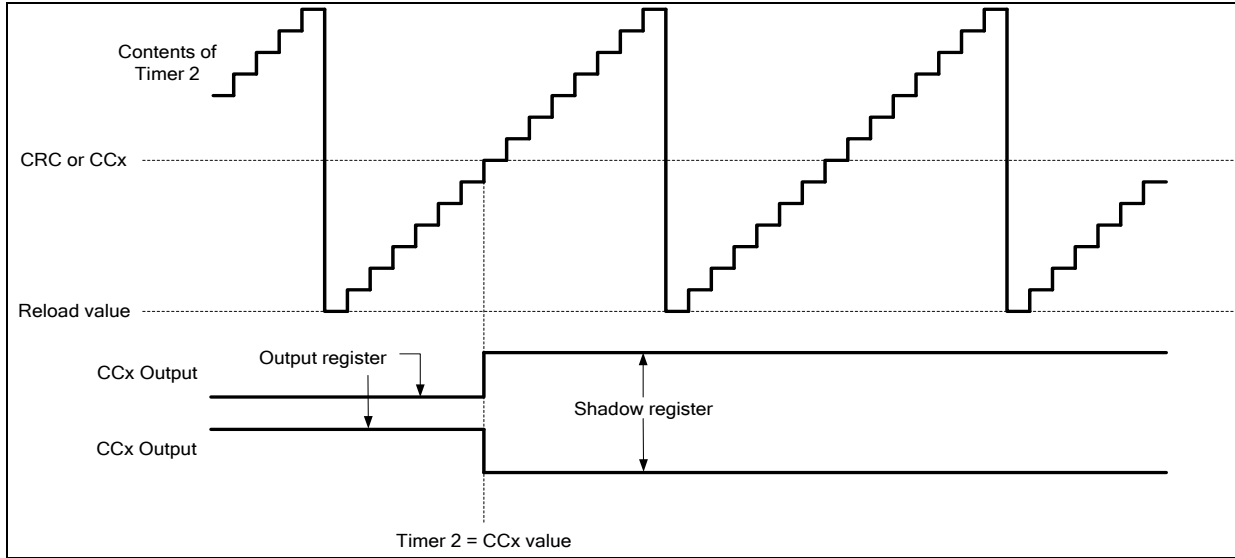


图 7-5: 比较模式 1 功能

7.3 捕获功能

在一个外部事件（模式0）或一个软件写操作（模式1）上，实际的定时器/计数器的值可以被保存在寄存器CCx或CRC中。

7.3.1 捕捉模式 0

如以下图 7-6，在模式0中，定时器2的值的捕捉在以下情况进行：

- (1) 上升沿输入CC0-CC3。
- (2) 下降沿输入CC0-CC3。
- (3) 上升沿及下降沿输入CC0-CC3。

定时器2的内容将被对应的捕捉寄存器锁存。

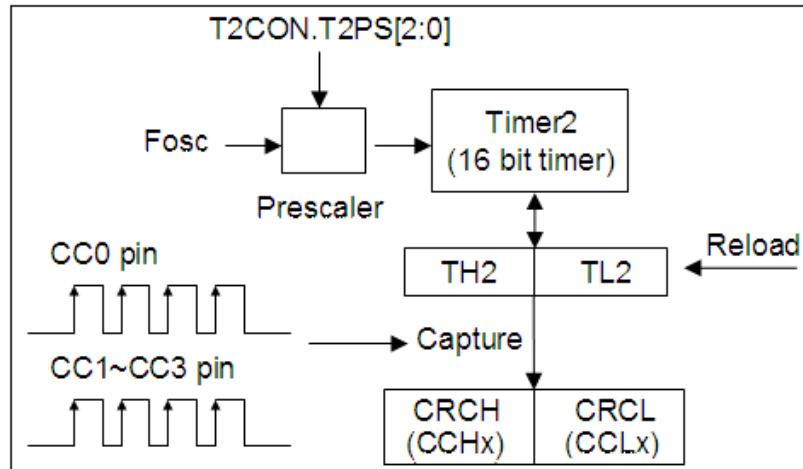


图 7-6: 捕捉模式 0 功能

7.3.2 捕捉模式 1

如以下图 7-7，在模式1中，定时器2的值的捕捉将导致写入到捕捉寄存器中的低字节无任何价值,其捕捉寄存器的写入值与改功能无关，定时器2的内容将被对应的捕捉寄存器锁存。

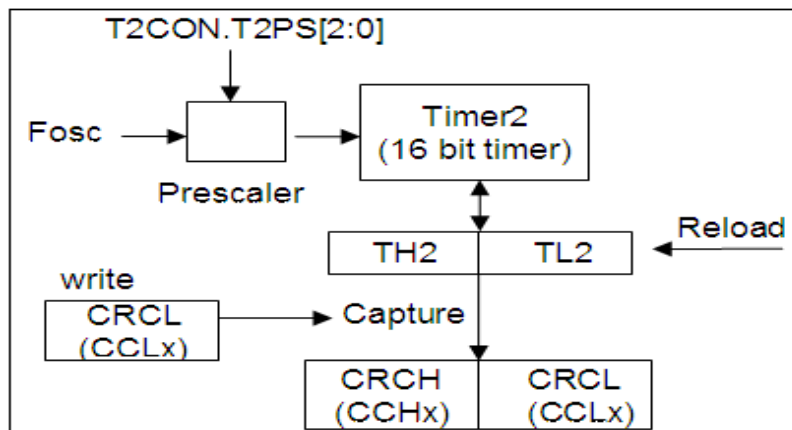


图 7-7: 捕捉模式 1 功能

8. 串行接口

两个串行缓冲器组成两个单独寄存器，即一个传输缓冲区和一个接收缓冲区。

写入数据到特殊功能寄存器（SFR）SBUF并设置这些数据在串行输出缓冲，并开始传输，来自SBUF的读取及从串行接收缓冲区读取数据，串行口可同时传输和接收数据，它也可在接收时缓存1字节，如CPU在第一个字节传输完成之前读取第二个字节，以防接收数据丢失。

符号	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
Serial interface 0 and 1											
PCON	Power control	87H	SMOD	-	-	-	-	-	STOP	IDLE	00H
AUX	Auxiliary register	91h	BRGS	-	-	-	-	-	-	DPS	00H
SCON	Serial Port control register	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SRELL	Serial Port reload register low byte	AAH	SREL.7	SREL.6	SREL.5	SREL.4	SREL.3	SREL.2	SREL.1	SREL.0	00H
SRELH	Serial Port reload register high byte	BAH	-	-	-	-	-	-	SREL.9	SREL.8	00H
SBUF	Serial Port data buffer	99H	SBUF[7:0]								00H
PFCON	Peripheral Frequency control register	D9h	-	-	SRELPS[1:0]		T1PS[1:0]		T0PS[1:0]		00H

符号: **AUX** 地址: **91h**

7	6	5	4	3	2	1	0	Reset
BRGS	-	-	-	-	-	-	DPS	00H

BRGS: BRGS = 0 –波特率产生器使用定时器 1 TH1 寄存器.

BRGS = 1 –波特率产生器使用 SREL 寄存器.

符号: **SCON** 地址: **98h**

7	6	5	4	3	2	1	0	Reset
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00h

SM0,SM1: 串行口模式选择.

SM0	SM1	Mode
0	0	0
0	1	1
1	0	2
1	1	3

在 UART 的 4 种模式中，模式 0~3 稍后解释

SM2: 多处理机通信使能位

REN: 如置位，串行接收使能，软件清除禁止接收

TB8: 在模式 2 和 3 中，发送的第 9 位数据位，置位或清零取决与它执行的功能，如奇偶校验功能，多处理机通信等。

RB8: 在模式 2 和 3 中，RB8 为接收的第 9 位数据位。在模式 1 中，如 SM2=0，RB8 为停止位。在模式 0 中，此位不被使用。须由软件清除。

TI: 发送中断标志位.在完成串行传输后由硬件置位, 须由软件清除。

RI: 接收中断标志。在完成串行传输后由硬件置位, 须由软件清除。

8.1 串行接口由以下 4 种模式可以设置

SM0	SM1	Mode	描述	Board Rate
0	0	0	Shift register	Fosc/12
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	Fosc/32 or Fosc/64
1	1	3	9-bit UART	Variable

这里的Fosc是晶体或振荡器的频率。

8.1.1 模式 0

引脚RXD充当输入和输出。TXD输出时钟。每次发送或接收以LSB最低位作首位, 每次8位。波特率固定为晶体的频率1/12,接收通过以下的在SCON中的设置标志在模式0中初始化: RI = 0及REN = 1。在其它模式中, 当REN=1时, 开始从起始位接收串行数据。



图 8-1: 发送模式 0

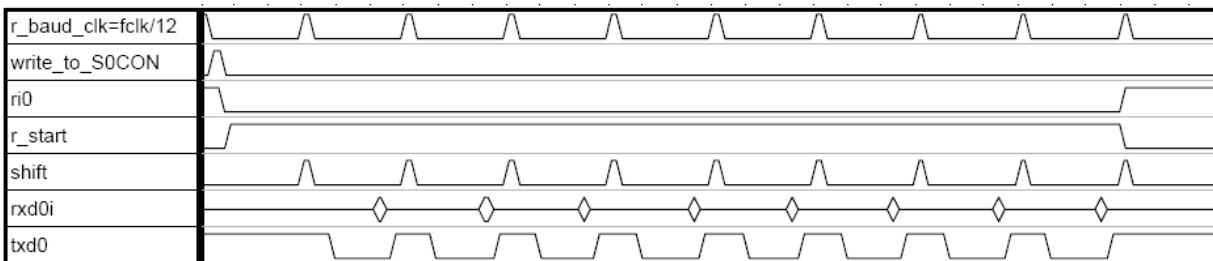


图 8-2: 接收模式 0

8.1.2 模式 1

引脚RXD充当输入，TXD充当串行输出，无任何外部时钟被使用，每次数据为10位：一个起始位(=0)，8个数据位（LSB位在前），及一个停止位(=1)。在接收数据时，起始位将被同步传输，8个数据位可通过SBUF来读取，一个停止位存于特殊功能寄存器SCON的设置标志RB8内，在模式1中，无论是内部的波特率发生器或定时器1可以用来指定波特率。

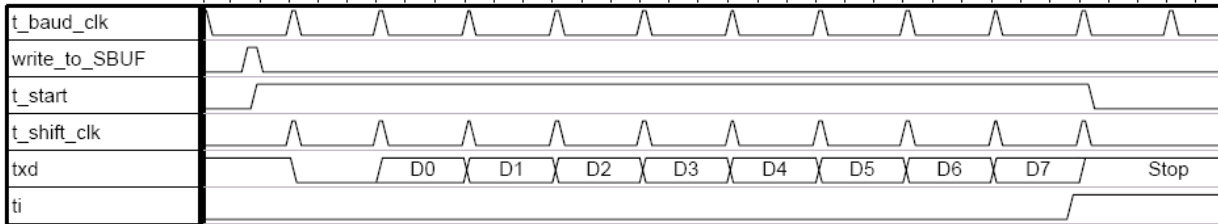


图 8-3: 发送模式 1

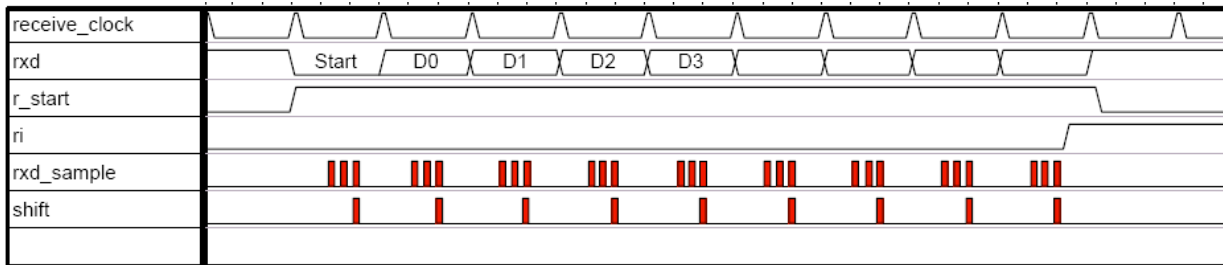


图 8-4: 接收模式 1

8.1.3 模式 2

该模式和模式1类似，但有两点不同。波特率被固定在振荡器的频率的1/32（SMOD=1）或1/64（SMOD=0），且有11位数据被传输或接收：1个起始位（=0），8个数据位（LSB在前），一个可编程的第9位及一个停止位（=1），9位可以用来控制串行接口的奇偶性。在传输中，SCON中的TB8输出第9位，在接收中,SCON中的RB8将被影响。

8.1.4 模式 3

模式2和3的唯一不同之处在于：在模式3中，无论是内部的波特率发生器或定时器1可以用来指定波特率。



图 8-5: 传输模式 2 和模式 3

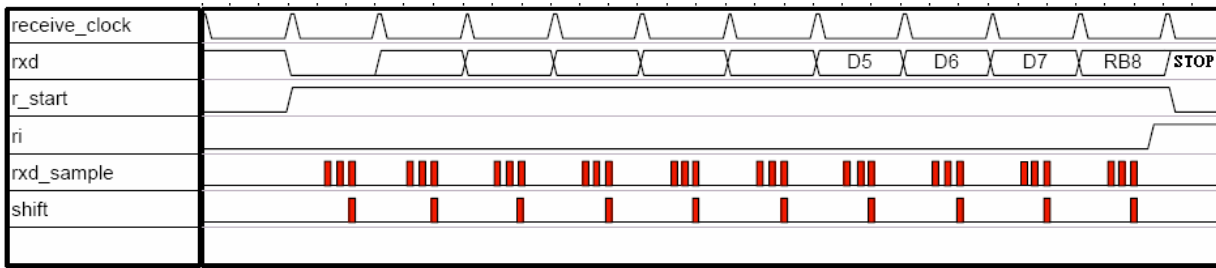


图 8-6: 接收模式 2 和 3 中

8.2 串行接口的多重机通讯

在串行接口的模式 2 和 3, 接收 9 位的功能, 可用于多处理器的通讯。在这种情况下, 从机在 SCON 中的位 SM2 被置位为 1。当主机输出从机的地址时, Bit9 将被置 1, 从而在所有的从机中导致串行口接收中断。从机将接收到的字节和它们的网络地址比较, 如匹配, 其从机将清除 SM2, 并接收其余的信息, 其它的从机将远离 SM2 无效, 并忽略此信息。在解决从机后, 其主机在 Bit9 清 0 时将输出剩余部分的信息, 因此, 没有串口接收中断会产生在未选中的处理机中。

8.3 输入频率控制寄存器

符号: PFCON						地址: D9h	
7	6	5	4	3	2	1	0
-	-	SRELPS[1:0]		T1PS[1:0]		T0PS[1:0]	
							Reset
							00H

SRELPS[1:0]: SREL 分频器选择位

SRELPS[1:0]	Prescaler
00	Fosc/64
01	Fosc /32

T1PS[1:0]: Timer1 分频器选择位

T1PS[1:0]	Prescaler
00	Fosc/12
01	Fosc
10	Fosc/96
11	reserved

8.4 波特率发生器

8.4.1 串行接口的模式 1 和 3

8.4.1.1 当 BRGS = 0 (在 AUX 寄存器):

(1) 当 T1PS[1:0] = 00

$$\text{Baud Rate} = \frac{2^{\text{SMOD}} \times F_{\text{osc}}}{32 \times 12 \times (256 - \text{TH1})}$$

(2) 当 T1PS[1:0] = 01

$$\text{Baud Rate} = \frac{2^{\text{SMOD}} \times F_{\text{osc}}}{32 \times (256 - \text{TH1})}$$

(3) 当 T1PS[1:0] = 10

$$\text{Baud Rate} = \frac{2^{\text{SMOD}} \times F_{\text{osc}}}{32 \times 96 \times (256 - \text{TH1})}$$

8.4.1.2 当 BRGS = 1 (在 AUX 寄存器)

(1) 当 SRELPS[1:0] = 00

$$\text{Baud Rate} = \frac{2^{\text{SMOD}} \times F_{\text{osc}}}{64 \times (2^{10} - \text{SREL})}$$

(2) 当 SRELPS[1:0] = 01

$$\text{Baud Rate} = \frac{2^{\text{SMOD}} \times F_{\text{osc}}}{32 \times (2^{10} - \text{SREL})}$$

9. 看门狗定时器

看门狗定时器是一个8位的计数器并会在计数器溢出时产生复位讯号.WDT在噪音.电源干扰,或断电等导致软件死循环或跑飞程序等情况下非常有用.WDT功能可以帮助拥护的软件从不正常的软件状态中恢复正常.WDT不同于通用的8052的计时器0,1,2.为了防止WDT复位,可以通过软件定时清除WDT计数器.当不可预料的复位发生时,用户应该检查WDTC寄存器的WDTF位.在一外部复位后,该看门狗定时器将无效且所有寄存器都清零.

看门狗定时器是一个自由运行的片上RC振荡器(约20KHz). WDT将保持运行就算是系统时钟被关闭(例如,在睡眠状态).在正常的运行或睡眠状态,一个WDT超时(如使能)将导致MCU复位.WDT在正常状态下可随时被使能或失效.请注意WDTC寄存器的WDTE位.默认的WDT超时时间为接近178.0ms (WDTM [3:0] = 0100b).

WDT有可选择的分频.要选择分频,需要正确设置看门控制寄存器(WDTC)的位3~位0(WDTM [3:0])

$$WDTCLK = \frac{20\text{KHz}}{2^{\text{WDTM}}}$$

$$\text{Watchdog reset time} = \frac{256}{WDTCLK}$$

表 9-1: WDT 超时时间

WDTM [3:0]	Divider (20 KHz RC oscillator in)	Time period @ 20KHz
0000	1	12.8ms
0001	2	25.6ms
0010	4	51.2ms
0011	8	102.4ms
0100	16	204.8ms (default)
0101	32	409.6ms
0110	64	819.2ms
0111	128	1.6384s
1000	256	3.2768s
1001	512	6.5536s
1010	1024	13.10s
1011	2048	26.21s
1100	4096	52.42s
1101	8192	104.85s
1110	16384	209.71s
1111	32768	419.43s

注: RC 振荡器(20 KHz), 大约有 ± 20 % 误差

看门狗一旦开始工作将无法停止.当WDTK寄存器用户可通过对看门狗定时器重置密钥(WDTK)写入55h实现WDT计时器清0.这将会清除8位计数器内容并让计数器重启.看门狗定时器必须定时刷新以防来自可变的重置复位请求信号.

当看门狗定时器溢出时, WDTF标志位将被置1并自动重置复位MCU.该标志位可被软件或外部复位清除.

对WDTE位写入1能使WDT功能使能.在WDTE设为1后.8位的计数器用通过WDTM[3:0]设置好的分频进行计数.它在溢出时将会产生复位讯号.WDTE位在MCU重启时将被自动清0,同时硬件复位或WDT复位.

看门狗一旦开始工作将无法停止.当WDTK寄存器用户可通过对看门狗定时器重置密钥(WDTK)写入55h实现WDT计时器

清0.这将会清除8位计数器内容并让计数器重启.看门狗定时器必须定时刷新以防来自可变的重置复位请求信号.

当看门狗定时器溢出时, WDTF标志位将被置1并自动重置复位MCU.该标志位可被软件或外部复位清除.

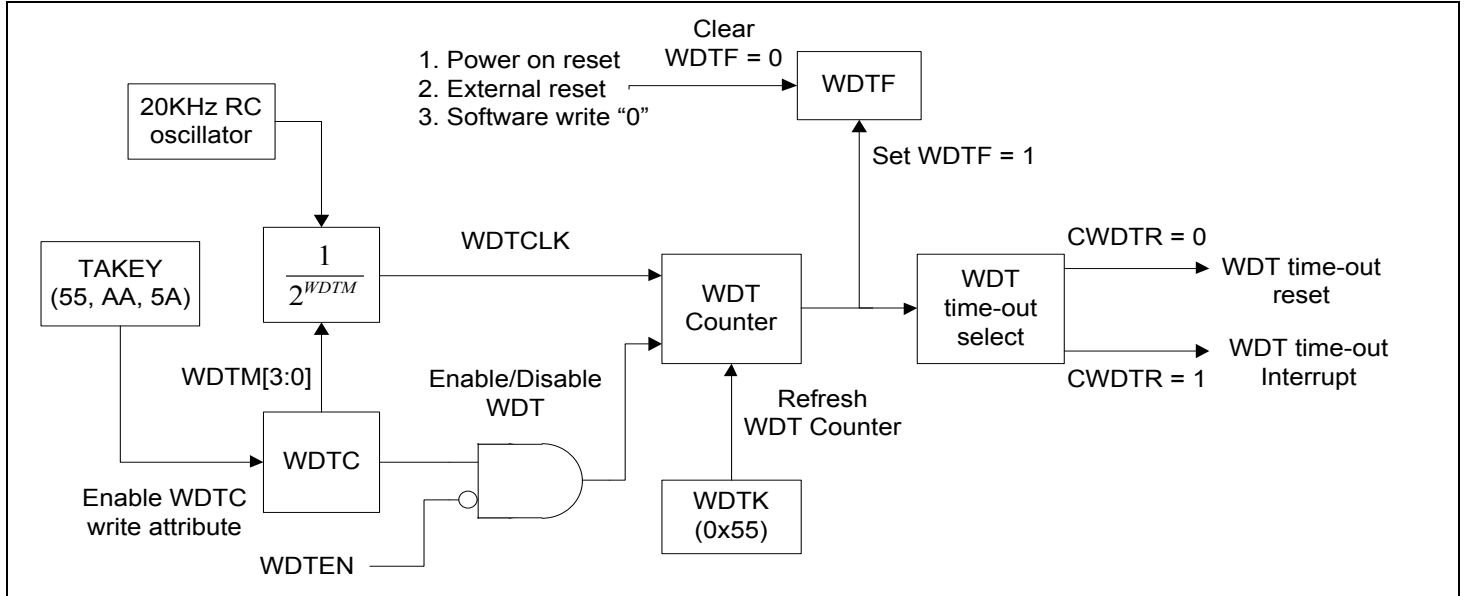


图 9-1: Watchdog 定时器框图

符号	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值	
Watchdog Timer												
TAKEY	Time Access Key register	F7h	TAKEY [7:0]									00H
WDTC	Watchdog timer control register	B6h	-	CWDTR	WDTE	-	WDTM [3:0]				04H	
WDTK	Watchdog timer refresh key	B7h	WDTK[7:0]									00H
RSTS	Reset status register	A1h	-	LVRLPIN TF	LVRLP F	PDRF	WDTF	SWRF	LVRF	PORF	00H	

符号: TAKEY							地址: F7h	
7	6	5	4	3	2	1	0	Reset
TAKEY [7:0]								00H

看门狗控制寄存器(WDTC)默认为仅读;软件需依序于寄存器 TAKEY 写入 55h, AAh 及 5Ah 才能对看门狗控制寄存器(WDTC)执行写入,启动看门狗功能及设定重置时间

```
MOV TAKEY, #55h
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah
```

符号: WDTC							地址: B6h	
7	6	5	4	3	2	1	0	Reset
-	CWDTR	WDTE	-	WDTM [3:0]				04H

CWDTR: 看门狗溢位状态选择(支持休眠唤醒)

0: 看门狗发生溢位时产生重置信号

1: 看门狗发生溢位时产生中断信号

WDTE: 看门狗定时器使能位.

0: 禁能.

1: 使能.

WDTM [3:0]: 看门狗重置信号产生时间选择位.请参考表 9-1所列看门狗重置信号产生时间.

符号: RSTS								地址: A1h
7	6	5	4	3	2	1	0	Reset
-	LVRLP INTF	LVRLP F	PDRF	WDTF	SWRF	LVRF	PORF	00h

WDTF: 看门狗重置旗标.

此旗标于芯片之复位信号是由看门狗重置产生时由硬件自动设置.此旗标可由软件清零

符号: WDTK								地址: B7h
7	6	5	4	3	2	1	0	Reset
WDTK[7:0]								00h

WDTK: 看门狗定时器计数器清零寄存器.

于此寄存器写入 0x55, 看门狗定时器计数器将清零重新计数.

范例 1:

看门狗定时器使能并选择重置信号产生时间为 3.2768s

第一步,先确认烧录时看门狗功能已选择使用

第二步,如下

```
MOV TAKEY, #55h
```

```
MOV TAKEY, #0AAh
```

```
MOV TAKEY, #5Ah ; enable WDTC write attribute.
```

```
MOV WDTC, #28h ; Set WDTM [3:0] = 1000b. Set WDTE =1 to enable WDT  
; function.
```

.
.
.

```
MOV WDTK, #55h ; Clear WDT timer to 0.
```

范例 2:

看门狗定时器使能并选择周期中断信号产生时间为 204.8ms

第一步,先确认烧录时看门狗功能已选择使用

第二步,如下

```
MOV TAKEY, #55h
```

```
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah ; enable WDTC write attribute.
MOV WDTC, #64h ; Set WDTM [3:0] = 0100b. Set WDTE =1 to enable WDT function
; and Set CWDTR =1 to enable period interrupt function
```

10. 中断

OB38R08A1提供14个中断源并带有4级优先权。每一中断源都在特殊寄存器(SFR)中有自身的请求标志，每个中断请求信号通过相应的标志及特殊寄存器（SFR）中的IEN0及IEN1中的使能位来独自允许或禁止。

当中断发生时，CPU将会跳转预先设定的地址，如表 10-1 所示，一旦中断开始执行，就只能被更高优先级的中断终止，其中断服务会被来自指令RETI的返回所终结。当一RETI被执行时且中断发生时，处理器将返回此指令，并执行下一条指令。

当中断条件发生时，该处理器通过设置一个标志位来表示，无论中断使能或禁止，每个中断标志都将在每个机器周期中采样一次，随后采样由硬件来检测，当中断被使能时，且采样指出相应中断时，中断请求标志被设置。在随后的指令周期中，中断将被硬件所确定。从而迫使一个LCALL 指向相应的地址向量。

当中断发生时，中断响应将需要不同的时间，这取决于相关的处理器。如一处理器执行的中断服务程序同等或更优先，新的中断将不会启用，在其它情况下，响应时间将取决于当前的指令。以最快的速度回应一个中断是需要7个机器周期，这包括一个检测中断的机器周期和6周期执行LCALL周期。

表 10-1: 中断向量

	Interrupt Request Flags	Interrupt Vector Address	Interrupt Number *(use Keil C Tool)
1	IE0 – 外部中断0	0003h	0
2	TF0 – 定时器0 中断	000Bh	1
3	IE1 – 外部中断 1	0013h	2
4	TF1 – 定时器1 中断	001Bh	3
5	RI/TI – 串行口中断	0023h	4
6	TF2/EXF2 – 定时器2 中断	002Bh	5
7	PWMIF – PWM 中断	0043h	8
8	SPIIF – SPI 中断	004Bh	9
9	ADCIF – A/D 转换 中断	0053h	10
10	KBIIF – 键盘 中断	005Bh	11
11	LVIIIF – 低压侦测 中断	0063h	12
12	IICIF – IIC 中断	006Bh	13
13	WDTIF – 看门狗 中断	008Bh	17
14	Comparator 中断	0093h	18

*参考Keil C用户指南中的有关中断功能使用说明

符号	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
Interrupt											
IEN0	Interrupt Enable 0 register	A8H	EA	-	ET2	ES	ET1	EX1	ET0	EX0	00H
IEN1	Interrupt Enable 1 register	B8H	EXEN2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPWM	00H
IEN2	Interrupt Enable 2 register	9AH	-	-	-	-	-	ECmpl	EWDT	-	00H
IRCON	Interrupt request register	C0H	EXF2	TF2	IICIF	LVIIIF	KBIIF	ADCIF	SPIIF	PWMI F	00H
IRCON2	Interrupt request register 2	97H	-	-	-	-	-	CmplF	WDTI F	-	00H
IP0	Interrupt priority level 0	A9H	-	-	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	00H
IP1	Interrupt priority level 1	B9H	-	-	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0	00H
ENHIT	ENHance Interrupt Type Register	E5H	-	ENHIT1	-	ENHIT0	-	-	-	-	07H
INTDEG	External Interrupt Deglitch Register	EEh						INT1DEG[1:0]	INT0DEG[1:0]		00H

符号: IEN0

地址: A8h

7	6	5	4	3	2	1	0	Reset
EA	-	ET2	ES0	ET1	EX1	ET0	EX0	00h

- EA: EA=0 –禁能所有中断
EA=1 –使能所有中断
- ET2: ET2=0 –禁能定时器 2 中断
ET2=1 –使能定时器 2 中断
- ES: ES=0 –禁能串行口中断
ES=1 –使能串行口中断
- ET1: ET1=0 –禁能定时器 1 中断
ET1=1 –使能定时器 1 中断
- EX1: EX1=0 –禁能外部中断 1
EX1=1 –使能外部中断 1
- ET0: ET0=0 –禁能定时器 0 中断
ET0=1 –使能定时器 0 中断
- EX0: EX0=0 –禁能外部中断 0
EX0=1 –使能外部中断 0

符号: IEN1							地址: B8h	
7	6	5	4	3	2	1	0	Reset
EXEN2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPWM	00H

EXEN2: 定时器 2 重载中断使能位

EXEN2 = 0 –禁能定时器 2 外部重载中断

EXEN2 = 1 –使能定时器 2 外部重载中断

IEIIC: IIC 中断使能位

IEIIC = 0 –禁能 IIC 中断

IEIIC = 1 –使能 IIC 中断

IELVI: 低压侦测中断使能位

IELVI = 0 –禁能低压侦测中断

IELVI = 1 –使能低压侦测中断

IEKBI: KBI 中断使能位

IEKBI = 0 –禁能 KBI 中断

IEKBI = 1 –使能 KBI 中断

IEADC: A/D 转换中断使能位

IEADC = 0 –禁能 ADC 中断

IEADC = 1 –使能 ADC 中断

IESPI: SPI 中断使能位

IESPI = 0 –禁能 SPI 中断

IESPI = 1 –使能 SPI 中断

IEPWM: PWM 中断使能位

IEPWM = 0 –禁能 PWM 中断

IEPWM = 1 –使能 PWM 中断

符号: IEN2							地址: 9Ah	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	ECmpl	EWDT	-	00H

ECmpl: ECmpl = 0 –禁能比较器中断.

ECmpl = 1 –使能比较器中断(包括比较器 0 及比较器 1).

EWDT: WDT 看门狗中断使能位

EWDT = 0 –禁能 WDT 中断

EWDT = 1 –使能 WDT 中断

符号: IRCON								地址: C0h	
7	6	5	4	3	2	1	0	Reset	
EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMIF	00H	

EXF2: 定时器 2 重载中断旗标位. 必须由软件清零

TF2: 定时器 2 中断旗标位. 必须由软件清零

IICIF: IIC 中断旗标位

LVIIF: 低压侦测中断旗标位

KBIIF: KBI 中断旗标位

ADCIF: A/D 转换中断旗标位

SPIIF: SPI 中断旗标位

PWMIF: PWM 中断旗标位. 必须由软件清零

符号: IRCON2								地址: 97h	
7	6	5	4	3	2	1	0	Reset	
-	-	-	-	-	CmpIF	WDTIF	-	00H	

CmpIF: 比较器中断旗标位,

当比较器中断使能时, 此旗标位将于程序至中断向量执行时自动清零.

当比较器中断禁能时, 此旗标位可由软件清零

WDTIF: 看门狗中断旗标位

符号: ENHIT								地址: E5h	
7	6	5	4	3	2	1	0	Reset	
-	ENHIT1	-	ENHIT0	-	-	-	-	07H	

注: 此功能只支持 H 版本以后的 MCU 设置才有效

ENHIT1: 中断 1 触发边沿控制位。

当 ENHIT1 设置为 0 且 IT1 设置为 1 时, 则开启中断 1 触发选项为下降沿触发

当 ENHIT1 且 IT1 皆设置为 1 时, 则开启中断 1 触发选项为上升沿触发

	ENHIT1=0	ENHIT1=1
IT1=0	INT1 低电平触发	INT1 低电平触发
IT1=1	INT1 下降沿触发	INT1 上升沿触发

ENHIT0: 中断 0 触发边沿控制位。

当 ENHIT0 设置为 0 且 IT0 设置为 1 时, 则开启中断 0 触发选项为下降沿触发

当 ENHIT0 且 IT0 皆设置为 1 时, 则开启中断 0 触发选项为上升沿触发

	ENHIT0=0	ENHIT0=1
IT0=0	INT0低电平触发	INT0 低电平触发
IT0=1	INT0下降沿触发	INT0上升沿触发

符号: INTDEG						Address: EEh		
7	6	5	4	3	2	1	0	Reset
-	-	-	-	INT1DEG[1:0]	INT0DEG[1:0]			00H

INT1DEG[1:0]: 选择 INT1 deglitch 时间.

- 00: no deglitch.
- 01: 5us
- 10: 10us
- 11: 15us

INT0DEG[1:0]: 选择 INT0 deglitch 时间.

- 00: no deglitch.
- 01: 5us
- 10: 10us
- 11: 15us

10.1 优先权配置

所有中断来源相组合于以下组中:

表 10-2: Priority level groups

Groups		
外部中断 0	-	PWM 中断
定时器 0 中断	看门狗中断	SPI 中断
外部中断 1	比较器中断	ADC 中断
定时器 1 中断	-	KBI 中断
串行口中断	-	低压侦测中断
定时器 2 中断	-	IIC 中断

通过对SFR中的IP0或IP1置位或清零，每一组中断都可被独自编程为四个优先级中的一个，如要求相同的优先级被同时接收，一内部轮询序列将依次确定哪个请求被优先服务。

符号: IP0						地址: A9h		
7	6	5	4	3	2	1	0	Reset
-	-	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	00h

符号: IP1						地址: B9h		
----------------	--	--	--	--	--	----------------	--	--

7	6	5	4	3	2	1	0	Reset
-	-	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0	00h


表 10-3: Priority levels

IP1.x	IP0.x	优先级层级
0	0	Level0 (最低)
0	1	Level1
1	0	Level2
1	1	Level3 (最高)

表 10-4: Groups of priority

Bit	Group		
IP1.0, IP0.0	外部中断 0	-	PWM 中断
IP1.1, IP0.1	定时器 0 中断	看门狗中断	SPI 中断
IP1.2, IP0.2	外部中断 1	比较器中断	ADC 中断
IP1.3, IP0.3	定时器 1 中断	-	KBI 中断
IP1.4, IP0.4	串行口 0 中断	-	低压侦测中断
IP1.5, IP0.5	定时器 2 中断	-	IIC 中断

表 10-5: 轮询顺序

中断源	顺序
外部中断 0	 轮询顺序
PWM 中断	
定时器 0 中断	
看门狗中断	
SPI 中断	
外部中断 1	
比较器中断	
ADC 中断	
定时器 1 中断	
KBI 中断	
串行口 0 中断	
低压侦测中断	
定时器 2 中断	
IIC 中断	

11. 电源管理单元

电源管理单元提供两种电源管理模式，即IDLE（待机/空闲）和STOP（停止），并为用户提供省电功能。

符号: PCON								地址: 87h	
7	6	5	4	3	2	1	0	Reset	
SMOD	-	-	-	-	-	STOP	IDLE	00h	

STOP: STOP 模式控制位。设置此位将运行 STOP 模式，STOP 位总是读为 0.

IDLE: IDIE 模式控制位。设置此位将运行 IDLE 模式，IDLE 位总是读为 0

11.1 待机模式(空闲模式)

使用IDLE（待机模式(空闲模式)）模式可通过对PCON寄存器的IDLE位置位。空闲模式停止MCU的时钟源，但外部的时钟源依旧保持，由于此时CPU不工作，电源功耗将被降低。当任何一中断信号或者一复位信号时，CPU将退出待机模式（空闲模式）。

11.2 停止模式

使用STOP（停止模式）模式可通过对PCON寄存器的STOP位置位。在此模式中，所有的中断源将全部被关闭，CPU将从一个无时钟的中断（外部中断0/1，键盘中断及低压侦测、KBI、比较器中断、看门狗中断）或者一个重置（看门狗及低压重置）条件下退出该模式，内部产生的中断（定时器，串行端口...），由于它们需要时钟源而会没有任何动作。

12. 脉宽调制器(PWM)

OB38R08A1提供-四个通道的PWM输出.
 中断值为43h.

符号	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
PWM											
PWMC	PWM Control register	B5h	PWMCS[2:0]			-	PWM3 EN	PWM2 EN	PWM1 EN	PWM0 EN	00H
PWMD0H	PWM 0 Data register high byte	BCh	PWMP 0	-	-	-	-	-	PWMD0[9:8]		00H
PWMD0L	PWM 0 Data register low byte	BDh	PWMD0[7:0]								00H
PWMD1H	PWM 1 Data register high byte	BEh	PWMP 1	-	-	-	-	-	PWMD1[9:8]		00H
PWMD1L	PWM 1 Data register low byte	BFh	PWMD1[7:0]								00H
PWMD2H	PWM 2 Data register high byte	B1h	PWMP 2	-	-	-	-	-	PWMD2[9:8]		00H
PWMD2L	PWM 2 Data register low byte	B2h	PWMD2[7:0]								00H
PWMD3H	PWM 3 Data register high byte	B3h	PWMP 3	-	-	-	-	-	PWMD3[9:8]		00H
PWMD3L	PWM 3 Data register low byte	B4h	PWMD3[7:0]								00H
PWMMDH	PWM Max Data register high byte	CEh	-	-	-	-	-	-	PWMMD[9:8]		00H
PWMMDL	PWM Max Data register low byte	CFh	PWMMD[7:0]								FFH

符号: **PWMC**

地址: **B5h**

7	6	5	4	3	2	1	0	Reset
PWMCS[2:0]			-	PWM3EN	PWM2EN	PWM1EN	PWM0EN	00H

PWMCS[2:0]: PWM 时钟源选择.

PWMCS [2:0]	Mode
000	Fosc
001	Fosc/2
010	Fosc/4
011	Fosc/6
100	Fosc/8
101	Fosc/12
110	Timer 0 overflow
111	Timer 0 external input (P1.2/T0)

PWM3EN PWM 通道 3 使能位.

PWM3EN = 1 – PWM 通道 3 使能.

PWM3EN = 0 – PWM 通道 3 禁能.

PWM2EN PWM 通道 2 使能位.

PWM2EN = 1 – PWM 通道 2 使能.

PWM2EN = 0 – PWM 通道 2 禁能.

PWM1EN: PWM 通道 1 使能位.

PWM1EN = 1 – PWM 通道 1 使能.

PWM1EN = 0 – PWM 通道 1 禁能.

PWM0EN: PWM 通道 0 使能位.

PWM0EN = 1 – PWM 通道 0 使能.

PWM0EN = 0 – PWM 通道 0 禁能.

符号: PWMD0H							地址: BCh	
7	6	5	4	3	2	1	0	Reset
PWMP0	-	-	-	-	-	PWMD0[9:8]		00H

符号: PWMD0L							地址: BDh	
7	6	5	4	3	2	1	0	Reset
PWMD0[7:0]								00h

PWMP0: PWM 通道 0 空闲极性选择.

“0” – PWM 通道 0 于空闲时置低.

“1” – PWM 通道 0 于空闲时置高.

PWMD0[9:0]: PWM 通道 0 数值寄存器.

符号: PWMD1H							地址: BEh	
7	6	5	4	3	2	1	0	Reset
PWMP1	-	-	-	-	-	PWMD1[9:8]		00H

符号: PWMD1L							地址: BFh	
7	6	5	4	3	2	1	0	Reset
PWMD1[7:0]								00H

PWMP1: PWM 通道 1 空闲极性选择.

“0” – PWM 通道 1 于空闲时置低.

“1” – PWM 通道 1 于空闲时置高.

PWMD1[9:0]: PWM 通道 1 数值寄存器.

符号: PWMD2H							地址: B1h	
7	6	5	4	3	2	1	0	Reset
PWMP2	-	-	-	-	-	PWMD2[9:8]		00H

符号: PWMD2L							地址: B2h	
7	6	5	4	3	2	1	0	Reset
PWMD2[7:0]								00H

PWMP2: PWM 通道 2 空闲极性选择.

“0” – PWM 通道 2 于空闲时置低.

“1” – PWM 通道 2 于空闲时置高.

PWMD2[9:0]: PWM 通道 2 数值寄存器.

符号: PWMD3H							地址: B3h	
7	6	5	4	3	2	1	0	Reset
PWMP3	-	-	-	-	-	PWMD3[9:8]		00H

符号: PWMD3L							地址: B4h	
7	6	5	4	3	2	1	0	Reset
PWMD3[7:0]								00H

PWMP3: PWM 通道 3 空闲极性选择.

“0” – PWM 通道 3 于空闲时置低.

“1” – PWM 通道 3 于空闲时置高.

PWMD3[9:0]: PWM 通道 3 数值寄存器.

符号: PWMMDH							地址: CEh	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	PWMMD[9:8]		00H

符号: PWMMDL							地址: CFh	
7	6	5	4	3	2	1	0	Reset
PWMMD[7:0]								FFH

PWMMD[9:0]: PWM 最大数值寄存器.

PWM 从 0000h 计数至最大数值寄存器 PWMMD[9:0]. 当 PWM 计数至与最大数值寄存器数值相等时 PWMMD[9:0] 产生溢出.

PWMPx = 0 & PWMDx = 00h

PWMx _____ Low _____

PWMPx = 0 & PWMDx ≠ 00h

PWMx _____

PWMPx = 1 & PWMDx = 00h

PWMx _____ High _____

PWMPx = 1 & PWMDx ≠ 00h

PWMx _____

$$\text{PWM period} = \frac{\text{PWMMD} + 1}{\text{PWM clock}}$$

$$\text{Leader pulse} = \frac{\text{PWMDx}}{\text{PWM clock}}$$

13. IIC 功能

这些IIC模块使用SCL（时钟）和SDA（数据）线来联系其它的IIC界面，其速度可以由软件设置特殊寄存器（SFR）中的IICBR[2: 0]，从而使其高达到400KBpS（最大值）。IIC模块可以是主机也可以是从机，提供两个中断（RXIF,TXIF），并有两个地址以作数据传输，它将产生开始，在主机模式中，反复启动和停止信号，在从机模式中，它将检测开始，反复启动和停止信号，最大沟通的长度及连接设备的数目被一个最大400pF 的电容器所限制

中断向量为6Bh.

符号	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
IIC function											
IICCTL	IIC control register	F9h	IICEN	MSS	MAS	AB_EN	BF_EN	IICBR[2:0]			04H
IICS	IIC status register	F8h	-	MPIF	LAIF	RXIF	TXIF	RXAK	TXAK	RW or BB	00H
IICA1	IIC Address 1 register	FAh	IICA1[7:1]							MATC H1 or RW1	A0H
IICA2	IIC Address 2 register	FBh	IICA2[7:1]							MATC H2 or RW2	60H
IICRWD	IIC Read/Write register	FCh	IICRWD[7:0]								00H
IICEBT	IIC Enaable Bus Transaction	FDh	FU_EN	-	-	-	-	-	-	-	00H

符号: IICCTL

地址: F9h

7	6	5	4	3	2	1	0	Reset
IICEN	MSS	MAS	AB_EN	BF_EN	IICBR[2:0]			04h

IICEN: IIC 模式使能

IICEN = 1, 使能

IICEN = 0, 禁止

MSS: 主\从机模式选择

MSS = 1, 选择主机模式

MSS = 0, 选择从机模式

软件必须在设置其它寄存器之前置位

MAS: 主模式的地址选择（仅在主机模式中）

MAS = 0, 使用 IICA1

MAS = 1, 使用 IICA2

AB_EN: 仲裁失去了使能位. (仅主机模式)

当 AB_EN 位使能, 硬件将检查仲裁丢失位. 一旦发生丢失仲裁, 硬件将返回到空闲状态. 如果此位被清除, 硬件也不会理会仲裁丢失情况. 当多主机与从机连接时需设置此位. 于单主机与从机时清除此位.

BF_EN: 总线忙碌使能位. (仅主机模式)

当 BF_EN 位使能, 硬件将不会产生开始信号至总在线直到 BF 已置低. 此位若被清零将永远产生开

始信号至总线当 MStart 置高.当多主机与从机连接时需设置此位.于单主机与从机时清除此位.

IICBR[2:0]: 波特率选择 (仅在主机模式) 这里的 Fosc 为外部晶体或振荡器的频率, 系统默认为 Fosc/512 以使用户选择

IICBR[2:0]	Baud rate
000	Fosc/32
001	Fosc/64
010	Fosc/128
011	Fosc/256
100	Fosc/512
101	Fosc/1024
110	Fosc/2048
111	Fosc/4096

符号: IICS

地址: F8H

7	6	5	4	3	2	1	0	Reset
-	MPIF	LAIF	RXIF	TXIF	RXAK	TXAK	RW or BB	00H

MPIF: 停止条件中断旗标

当停止条件发生此位将置高.此位需由软件清零

LAIF: 仲裁失去中断旗标. (只主机模式)

当仲裁失去条件发生此位将置高.此位需由软件清零

RxIF: 数据接收中断标志位, 在 IICRWD (IIC 读写数据缓存) 载入一个新的接收数据时被置位; 软件清零该位后, IIC 中断标志位 (IICIF) 将自动清零。

TxIF: 数据传输中断标志位, 当所有位于转换寄存器中的位数据被传输时, 该位被置位, 来自 IICRWD(IIC 读/写数据缓存)的 8 位数据被下载至转换寄存器中; 软件清零该位后, IIC 中断标志位 (IICIF) 将自动清零。

RxAK: 接收确认位. 置零,这意味着一个确认信号已经收到在完成 8 位数据传输总线上下。

TxAK: 传输确认位. 当收到完整的 8 位数据, 此位将设置(NoAck) 或清除(Ack) 并传输到主机显示接收状态. 事实上, 它是被作为一个字节的传输在第 9 位如图 13-1.

RW or BB: 主机模式:

BB:总线忙碌表示位

当检测到 scl=0 或 sda=0 或总线产生开始信号,此位将置高.当检测到停止信号,此位将清零. 此位可由软件清零以使系统回到就绪状态.

从机模式:

RW:从机模式的读取 (接收) 或写(传输) 在 IIC 总线. 当此位被清除, 表示从机模式在 IIC 总线接收数据.(只从机模式)

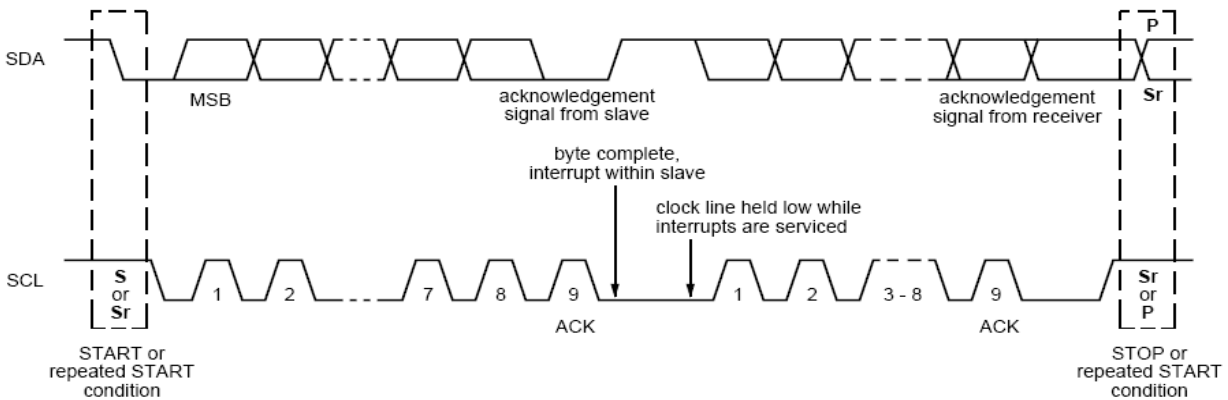
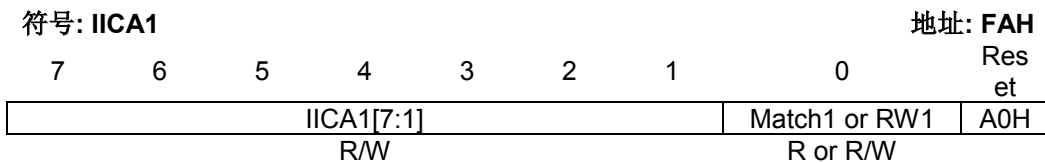


图 13-1: Acknowledgement bit in the 9th bit of a byte transmission



从机模式:

IICA1[7:1]: IIC 地址寄存器

这是为从机模式的第一个 7-bit 的地址，它在一个地址（来自主机模式）接收后被检测

Match1: 当 IICA1 和来自主机模式方的接收地址匹配时，该位将被硬件置位，当 IIC 总线读或写第一笔资料时，该位将被自动清除。

主机模式:

IICA1[7:1]: IIC 地址寄存器

显示要与它通讯的从机的 7 位地址

RW1: 如模块被设置为 Mstart 或 Rstart bit，该位将被作为从机方的 RW 发送，它在 IIC 地址之后的抵 8th 位显示，如图 13-2。它用来告诉从属模式中将来的通信的方向。如设置为 1，模块在主模式中为接收方，如是 0，模块在主模式中为传输方。

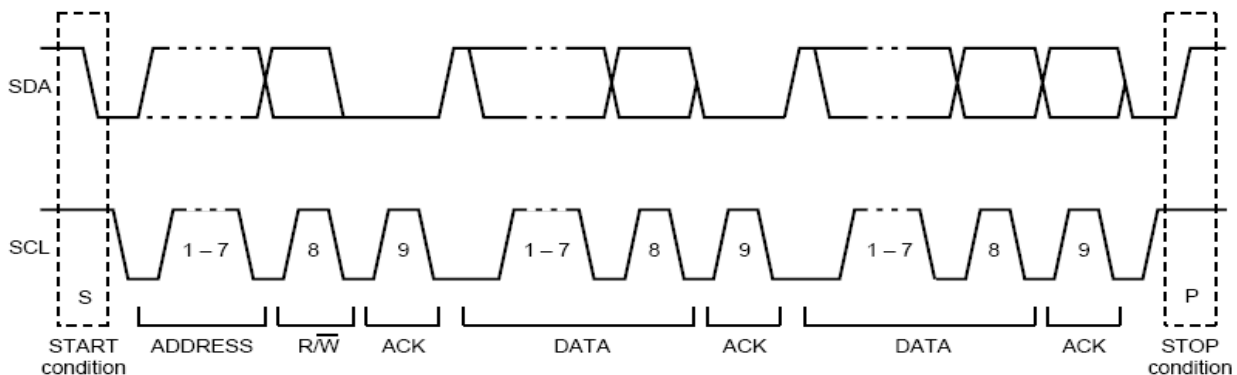


图 13-2: RW bit in the 8th bit after IIC address

符号: IICA2							地址: FBh	
7	6	5	4	3	2	1	0	Reset
IICA2[7:1]							Match2 or RW2	60h
R/W							R or R/W	

从机模式:

IICA2[7:1]: IIC 地址寄存器

这是为从机模式的第二个 7-bit 的地址, 它在一个地址 (来自主机模式) 接收后被检测

Match2: 当 IICA2 和来自主机模式方的接收地址匹配时, 该位将被硬件置位, 当 IIC 总线读或写第一笔资料时, 该位将被自动清除。

主机模式:

IICA2[7:1]: IIC 地址寄存器

显示要与它通讯的从机的 7 位地址

RW2: 如模块被设置为 Mstart 或者 Rstart bit, 该位将被作为从机方的 RW 发送, 它用来告诉从机模式中将来的通信的方向。如设置为 1, 模块在主模式中为接收方, 如是 0, 模块在主模式中为传输方。

符号: IICRWD							地址: FCh	
7	6	5	4	3	2	1	0	Reset
IICRWD[7:0]								00h

IICRWD[7:0]: IIC 读写缓存

在接收 (读) 模式中, 接收的字节存储于此
 在传输模式中, 该字节通过该 SDA 被转换

符号: IICEBT							地址: FDH	
7	6	5	4	3	2	1	0	Reset
FU_EN	-	-	-	-	-	-	-	00H

主机模式:

- 00: 保留
 - 01: IIC 模块将致能以便由 SDA 及 SCL 读写数据.
 - 10: IIC 模块将于 SDA 及 SCL 发出开始信号,再送出储存于 IICA1 或 IICA2(由 MAS 位选择)的地址
 - 11: IIC 模块将于 SDA 及 SCL 发出停止信号.
- FU_EN[7:6] 此两位将自动清零,软件需重复写入.

从机模式:

- 01: FU_EN[7:6] 此两位于从机模式仅能写入 01.其它值是无效的.
- 注意:
 FU_EN[7:6] 当总线空闲时,于读写数据前此两位需先写入 01;否则,SCL 将被锁住 (置低).

FU_EN[7:6] 当读写数据后并收到主机之停止信号时此两位需接着再写入 01.
在传送数据时(从机模式),于此两位写入 01 前待传送数据需先写入 IICRWD.
FU_EN[7:6] 此两位将自动清零,软件需重复写入.

14. SPI功能

串行外围接口（SPI）是一个同步串行接口，允许主机和从机沟通。

中断向量为4Bh.

SPI 使用4个信号

SPI_MOSI: 在主机模式中数据输出，在从机模式中数据输入。

SPI_MISO: 在主模式中数据输入，在从机模式中数据输出，

SPI_SCK: 从机模式时钟输出，以上数据与同步

SPI_SS: 从机模式中输入。

从机设备检测该信号来判断是否被主模式选择。

在主机模式中，它可选择所需的从机设备的任何IO的值为零。如图14-1所示的一个例子将显示主机与从机的4路信号之间的关系

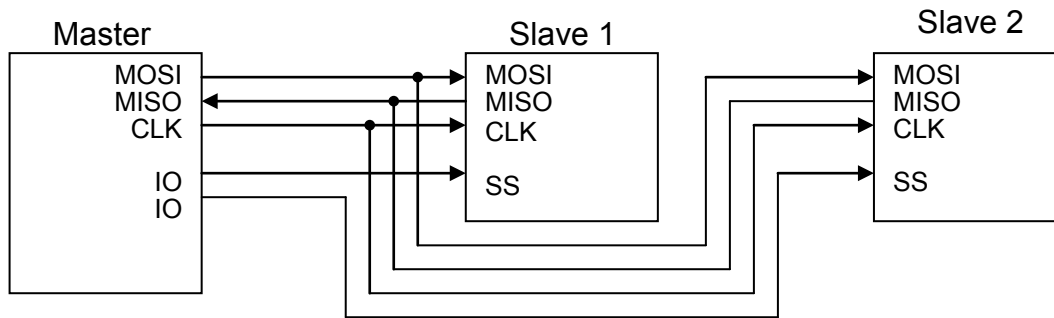


图 14-1: SPI 主主机模式和从机模式之间的信号

这仅是单通道SPI的界面，SPI SFRs 如下所示:

SPI	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
SPI function											
SPIC1	SPI control register 1	F1h	SPIEN	SPIMSS	SPISSP	SPICKP	SPICKE	SPIBR[2:0]			08H
SPIC2	SPI control register 2	F2h	SPIFD	TBC[2:0]			SPIRST	RBC[2:0]			00H
SPIS	SPI status register	F5h	SPIRF	SPIMLS	SPIOV	SPITXIF	SPITDR	SPIRXIF	SPIRDR	SPIRS	40H
SPITXD	SPI transmit data buffer	F3h	SPITXD[7:0]								00H
SPIRXD	SPI receive data buffer	F4h	SPIRXD[7:0]								00H

符号: SPIC1							地址: F1h				
7	6	5	4	3	2	1	0	Reset			
SPIEN	SPIMSS	SPISSP	SPICKP	SPICKE	SPIBR[2:0]			08h			

SPIEN: SPI 模式使能位. “1” SPI 功能使能. “0” SPI 功能禁止.

SPIMSS: 主/从机模式选择位

“1” 主机模式.

“0” 从机模式.

SPISSP: 从机选择的极性 (仅从机模式有效)

“1” -高电平动作.

“0” -低电平动作.

SPICKP: 时钟空闲极性

“1” -空闲时 SCK 为高电平. Ex :



“0” -空闲时 SCK 为低电平. Ex :

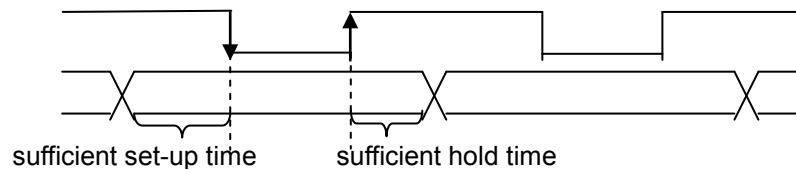


SPICKE: 时钟采样相位选择位.

“1” -数据锁存在上升沿

“0” -数据锁存在下降沿.

*为确保数据锁存的稳定性, 产生的输出数据, 无论对方可锁存稳定的数据是在上升沿或下降沿, 都将作为考虑在以下示例中.



SPIBR[2:0]: SPI 速率选择位 (仅主机模式有效), 这里的 F_{osc} 为外部时钟或振荡器的频率

SPIBR[2:0]	Baud rate
0:0:0	$F_{osc}/4$
0:0:1	$F_{osc}/8$
0:1:0	$F_{osc}/16$
0:1:1	$F_{osc}/32$
1:0:0	$F_{osc}/64$
1:0:1	$F_{osc}/128$
1:1:0	$F_{osc}/256$
1:1:1	$F_{osc}/512$

符号: SPIC2							地址: F2h	
7	6	5	4	3	2	1	0	Reset
SPIFD	TBC[2:0]		SPIRST		RBC[2:0]		00h	

SPIFD: 全双工通信模式使能位.

“1”: 使能全双工模式.

“0”: 禁止全双工模式.

当此位被置位时, TBC[2;0]和 RBC[2;0]将被重置并归零。也就是说, 只有 8 位数据通讯在全双工模式中被允许, 当主机模式通过 MOSI 线传输数据到从机模式中, 意味着带有数据输入和输出的全双工模式在同样的 SCK 时钟中是同步的, 如下所示.

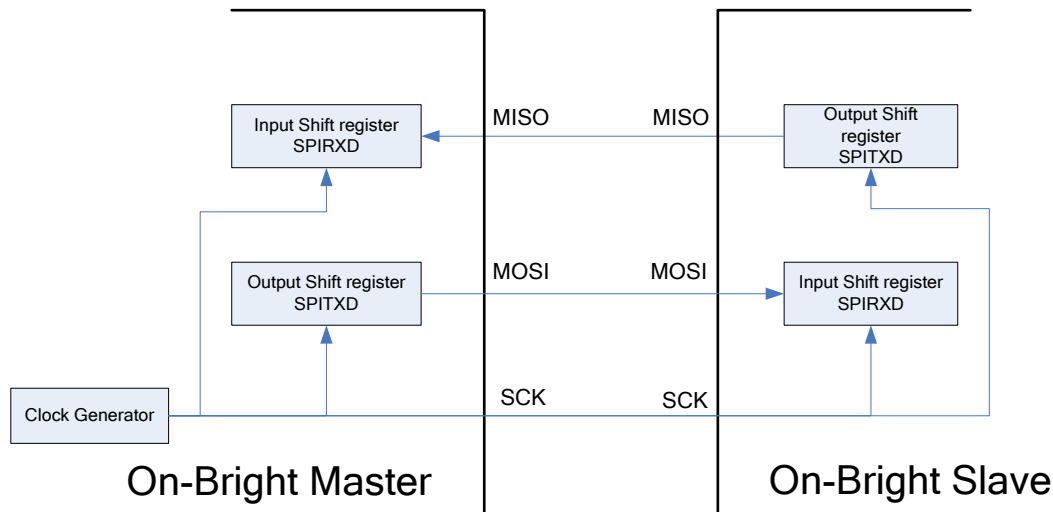


图 14-2: SPI 主机和从机传输方法

TBC[2:0]: SPI 发送数据位, 这里 1-8 位数据除了在全双工模式中是被允许的

TBC[2:0]	Bit counter
0:0:0	8 bits output
0:0:1	1 bit output
0:1:0	2 bits output
0:1:1	3 bits output
1:0:0	4 bits output
1:0:1	5 bits output
1:1:0	6 bits output
1:1:1	7 bits output

SPIRST: SPI 重新致能模式选择位. (仅从机模式有效)

= 0 禁能. SPI 于 SS 脚重新致能时, 传送或接收数据.

于传送或接收数据缓冲寄存器内, 前次传送或接收之数据完全保留. (表示它是有效的)

= 1 使能. SPI 于 SS 脚重新致能时, 传送或接收新数据,

于传送或接收数据缓冲寄存器内, 前次传送或接收之数据全部舍弃. (表示它是无效的)

RBC[2:0]: SPI 接收数据位, 这里 1-8 位数据除了在全双工模式中是被允许的

RBC[2:0]	Bit counter
0:0:0	8 bits input
0:0:1	1 bit input
0:1:0	2 bits input
0:1:1	3 bits input
1:0:0	4 bits input
1:0:1	5 bits input
1:1:0	6 bits input
1:1:1	7 bits input

符号: SPIS							地址: F5H	
7	6	5	4	3	2	1	0	Reset
SPIRF	SPIMLS	SPIOV	SPITXIF	SPITDR	SPIRXIF	SPIRDR	SPIRS	40H

SPIRF: SS 脚被释放旗标指示位.

当 SS 脚被释放且 SPIRST 为"1"时,此位被置高.

SPIMLS: MSB 或者 LSB 输入/输出优先

"1": MSB 输入/输出优先

"0": LSB 输入/输出优先

SPIOV: 溢出标志位

当 SPIRDR 被置位 (SPIRXR 中的一个字节, 但未被移除) 且下一数据也进入时 (这里不具备锁存功能), 该标志将被设置并导致在 SPIRXD 接收的数据被溢出毁坏, 当 SPIRDR 被清除时, 它也被硬件清除。

SPITXIF: 传输中断标志

当 SPITXD 寄存器的数据被下载到转换寄存器时, 该位被置位

SPITDR: 传输数据就绪位

当 MCU 结束对 SPITXD 寄存器读写数据时, MCU 须将改位置位, 以通知 SPI 模块发送数据。在 SPI 模块完成从 SPITXD 或 SPITXD 发送数据到下载到转换寄存器后, 该位将自动被清除。

SPIRXIF: 接受中断标志位

在 SPIRXD 被重载一个新的接收数据后, 该位被置位

SPIRDR: 接收数据就绪位

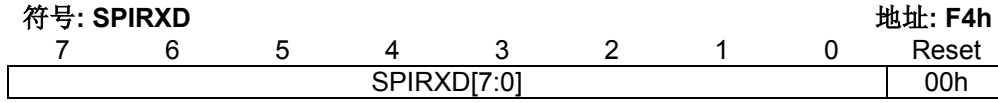
当一字节被接收时, SPIRDR 被设为一个标志去通知 MCU。在获取来自 SPIRXD 寄存器的数据后, MCU 须清除该位。如 SPI 模块向传输方写入新的数据至 SPIRXD 中且在该位清零之前, 该数据将随后被覆盖。

SPIRS: 接收起始位

该位被置位以此通知 SPI 模块来接收位于 SPIRXD 寄存器中的数据

符号: SPITXD							地址: F3h	
7	6	5	4	3	2	1	0	Reset
SPITXD[7:0]								00h

SPITXD[7:0]: 传输数据缓冲



SPIRXD[7:0]: 接收数据缓冲.

15. KBI—键盘接口

键盘接口(KBI) 可被连接一个 4 x n 矩阵键盘或任何普通的设备. 具有4路输入高或低的可编程的中断能力. 该4路接口可通过外部中断从IDLE和STOP模式中退出. 且输出相互彼此独立, 但共享同一中断向量5Bh.

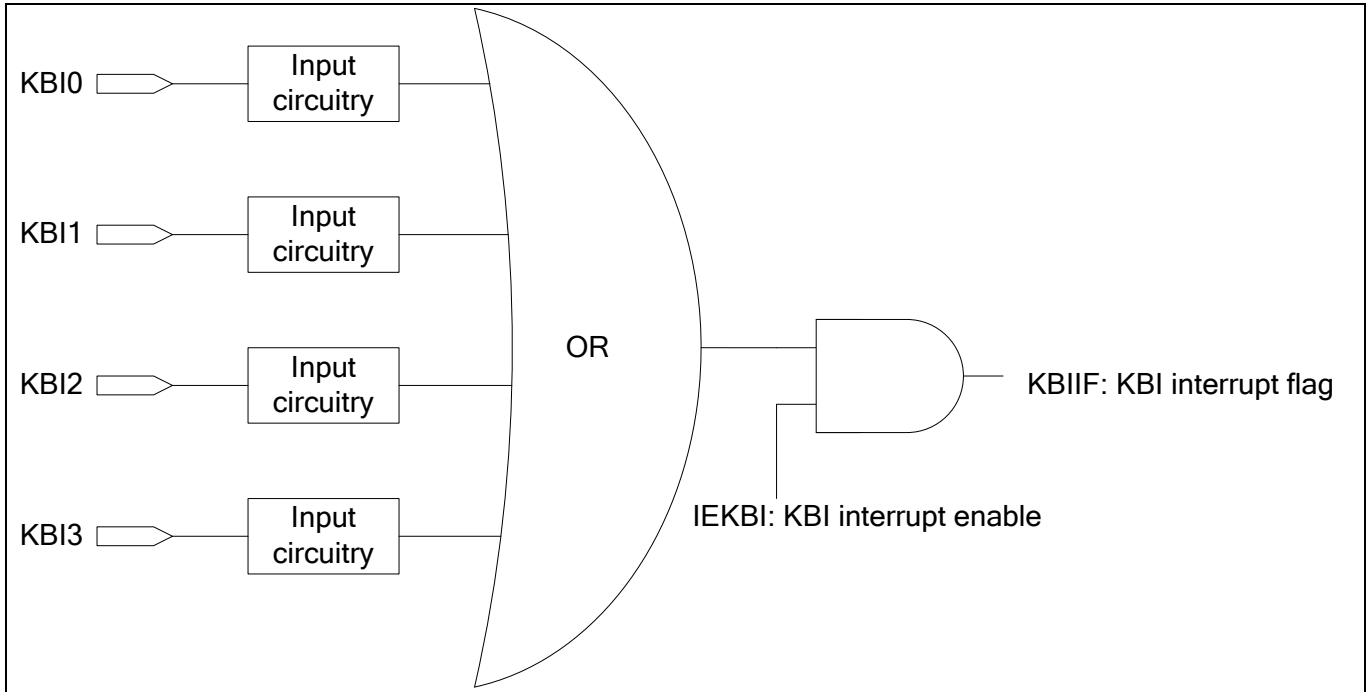


图 15-1: 键盘接口框图

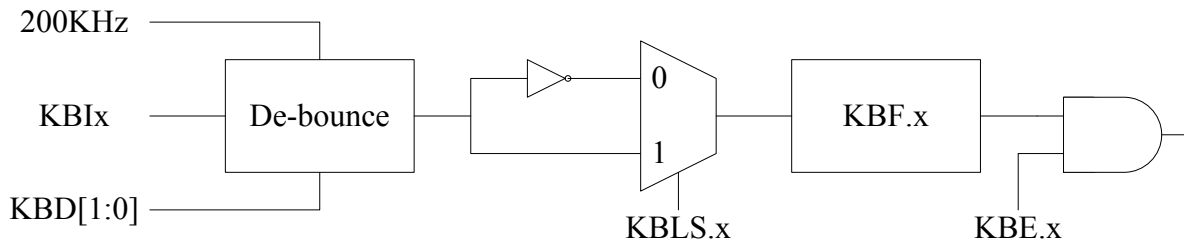


图 15-2: 键盘输入电路

KBI	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
KBI function											
KBLS	KBI level selection	93h	-	-	-	-	KBLS3	KBLS2	KBLS1	KBLS0	00H
KBE	KBI input enable	94h	-	-	-	-	KBE3	KBE2	KBE1	KBE0	00H
KBF	KBI flag	95h	-	-	-	-	KBF3	KBF2	KBF1	KBF0	00H
KBD	KBI De-bounce control register	96h	KBDE N	-	-	-	-	-	KBD1	KBD0	00H

符号: **KBLS**

 地址: **93h**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	KBLS.3	KBLS.2	KBLS.1	KBLS.0	00h

KBLS.3: KBI3 电位选择位

0 : 使能 KBI3 低电平检测.

1 : 使能 KBI3 高电平检测.

KBLS.2: KBI2 电位选择位

0 : 使能 KBI2 低电平检测.

1 : 使能 KBI2 高电平检测.

KBLS.1: KBI1 电位选择位

0 : 使能 KBI1 低电平检测.

1 : 使能 KBI1 高电平检测.

KBLS.0: KBI0 电位选择位

0 : 使能 KBI0 低电平检测.

1 : 使能 KBI0 高电平检测.

 符号: **KBE**

 地址: **94h**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	KBE.3	KBE.2	KBE.1	KBE.0	00h

KBE.3: KBI3 使能位

0 : 使能标准 I/O 口.

1 : 使能 KBF 寄存器中的 KBF.3 位产生一中断请求.

KBE.2: KBI2 使能位

0 : 使能标准 I/O 口.

1 : 使能 KBF 寄存器中的 KBF.2 位产生一中断请求.

KBE.1: KBI1 使能位

0 : 使能标准 I/O 口.

1 : 使能 KBF 寄存器中的 KBF.1 位产生一中断请求.

KBE.0: KBI0 使能位

0 : 使能标准 I/O 口.

1 : 使能 KBF 寄存器中的 KBF.0 位产生一中断请求.

 符号: **KBF**

 地址: **95h**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	KBF.3	KBF.2	KBF.1	KBF.0	00h

KBF.3: KBI3 标志位

当 KBI3 检测到一编程水平线时, 该位由硬件置位.

如 KBE.3 同样被置位时将产生一个 KBI 中断请求, 该位须软件清零.

KBF.2: KBI2 标志位

当 KBI2 检测到一编程水平线时, 该位由硬件置位.

如 KBE.2 同样被置位时将产生一个 KBI 中断请求, 该位须软件清零.

KBF.1: KBI1 标志位

当 KBI1 检测到一编程水平线时, 该位由硬件置位.

如 KBE.1 同样被置位时将产生一个 KBI 中断请求, 该位须软件清零.

KBF.0: KBI0 标志位

当 KBI0 检测到一编程水平线时, 该位由硬件置位.

如 KBE.0 同样被置位时将产生一个 KBI 中断请求, 该位须软件清零.

符号: KBD							地址: 96H	
7	6	5	4	3	2	1	0	Reset
KBDEN	-	-	-	-	-	KBD.1	KBD.0	00H

KBDEN: KBI 去反弹使能位. 默认使能.

KBDEN = 0, 使能去反弹功能. 去反弹时间于 KBD [1:0] 选择.

KBDEN = 1, 禁能去反弹功能. KBI 输入口不需要去反弹机制.

KBD[1:0]: KBI 去反弹时间选择. 当 KBDEN = "0", 默认去反弹时间为 320 ms.

KBD[1:0] = 00, 去反弹时间为 320 ms.

KBD[1:0] = 01, 去反弹时间为 160 ms.

KBD[1:0] = 10, 去反弹时间为 80 ms.

KBD[1:0] = 11, 去反弹时间为 40 ms.

16. LVI – 低压侦测中断

中断向量为 63h.

符号	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
LVI function											
RSTS	Reset status register	A1h	-	LVRLP INTF	LVRLP F	PDR F	WDTF	SWRF	LVRF	PORF	00H
LVC	Low voltage control	E6h	LVI_EN	-	LVRE	LVIF	-	-	LVIS[1:0]		20H

符号: **RSTS** 地址: **A1h**

7	6	5	4	3	2	1	0	Reset
-	LVRLP INTF	LVRLP PF	PDRF	WDTF	SWRF	LVRF	PORF	00H

LVRLPINTF: “内部” 低电压复位旗标.

当 MCU 复位信号由“内部”低电压复位产生时, LVRLPINTF 旗标将由硬件置高.此旗标需由软件清零.

LVRLPF: 低电压复位旗标.(低功率)

当 MCU 复位信号由低电压复位(低功率)产生时, LVRLPF 旗标将由硬件置高.此旗标需由软件清零.

PDRF: 外部复位脚复位旗标.

此旗标于芯片之复位信号是由外部复位脚产生时由硬件自动设置.此旗标可由软件清零

LVRF: 低电压复位旗标.

此旗标于芯片之复位信号是由低电压复位产生时由硬件自动设置.此旗标可由软件清零

PORF: 上电复位旗标.

此旗标于芯片之复位信号是由上电复位产生时由硬件自动设置.此旗标可由软件清零

符号: **LVC** 地址: **E6h**

7	6	5	4	3	2	1	0	Reset
LVI_EN	-	LVRE	LVIF	-	-	LVIS[1:0]		20H

LVI_EN: 低电压中断功能使能位

0: 禁能低电压检测中断功能

1: 使能低电压检测中断功能

LVRE: 外部低电压重置功能使能位.

0: 禁能外部低电压重置功能.

1: 使能外部低电压重置功能.

LVIF: 低电压中断旗标位

LVIS: 低电压中断电压位阶选择:

00: 2.4V

01: 2.6V

10: 3.2V

11: 4.0V

17. 12位模拟数字转换器(ADC)

OB38R08A1提供了8通道加上内部1通道的12位ADC，数字输出采样的模拟信号放入ADCD[11:0]中，而ADC的中断向量则为53h.

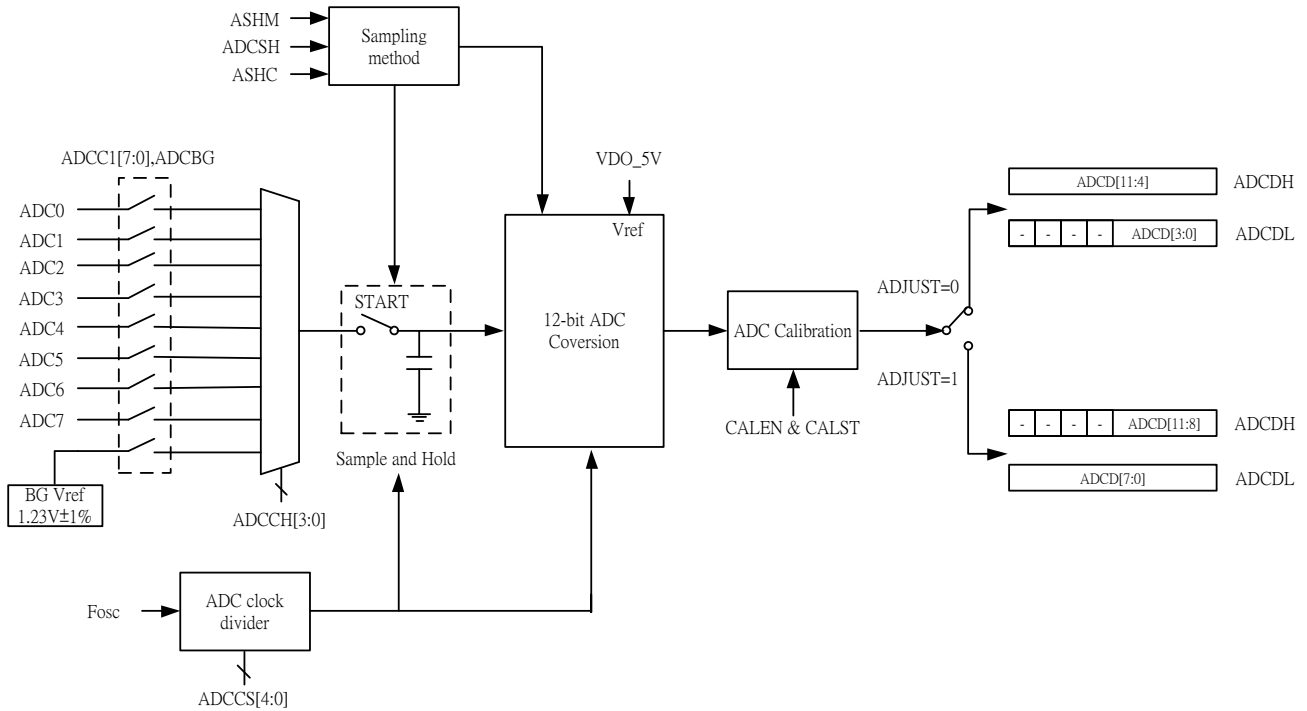


图 17-1: ADC 模拟到数字转换器的操作设置

ADC SFRs 如下所示:

符号	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
ADC											
ADCC1	ADC Control 1	ABH	ADC7EN	ADC6EN	ADC5EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	ADC0EN	00H
ADCC2	ADC Control 2	ACH	START	ADJUST	ASHM	ASHC	ADCCH[3:0]			00H	
ADCDH	ADC Data High	ADH	ADCDH[7:0]								00H
ADCDL	ADC Data Low	AEH	ADCDL[7:0]								00H
ADCCS	ADC Clock Select	AFH	ADCB GE	-	-	ADCCS[4:0]				00H	
ADCSH	ADC Sample and Hold Time	EFH	ADCSH[7:0]								00H

ADC Calibration:

符号	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
ADCCAL	ADC Calibration	FFFFH	-	-	-	-	-	-	CALEN	CALST	02H

符号: **ADCC1**

 地址: **ABh**

7	6	5	4	3	2	1	0	Reset
ADC7EN	ADC6EN	ADC5EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	ADC0EN	00H

ADC7EN: ADC 通道 7 使能位。

ADC7EN = 0 –禁能 ADC 通道 7。

ADC7EN = 1 –使能 ADC 通道 7。

ADC6EN: ADC 通道 6 使能位。

ADC6EN = 0 –禁能 ADC 通道 6。

ADC6EN = 1 –使能 ADC 通道 6。

ADC5EN: ADC 通道 5 使能位。

ADC5EN = 0 –禁能 ADC 通道 5。

ADC5EN = 1 –使能 ADC 通道 5。

ADC4EN: ADC 通道 4 使能位。

ADC4EN = 0 –禁能 ADC 通道 4。

ADC4EN = 1 –使能 ADC 通道 4。

ADC3EN: ADC 通道 3 使能位。

ADC3EN = 0 –禁能 ADC 通道 3。

ADC3EN = 1 –使能 ADC 通道 3。

ADC2EN: ADC 通道 2 使能位。

ADC2EN = 0 –禁能 ADC 通道 2。

ADC2EN = 1 –使能 ADC 通道 2。

ADC1EN: ADC 通道 1 使能位。

ADC1EN = 0 –禁能 ADC 通道 1。

ADC1EN = 1 –使能 ADC 通道 1。

ADC0EN: ADC 通道 0 使能位。

ADC0EN = 0 –禁能 ADC 通道 0。

ADC0EN = 1 –使能 ADC 通道 0。

 符号: **ADCC2**

 地址: **ACH**

7	6	5	4	3	2	1	0	Reset
START	ADJUST	ASHM	ASHC	ADCCH[3:0]				00H

START: 当该位被置位时, ADC 将启动单次转换, 转换完成后自动清零。

ADJUST: ADC 数字输出格式调整。

ADJUST = 0: (默认)

ADC 数字输出高字节 ADCD [11:4] = ADCDH [7:0].

ADC 数字输出低字节 ADCD [3:0] = ADCDL [3:0].

ADJUST = 1:

ADC 数字输出高字节 ADCD [11:8] = ADCDH [3:0].

ADC 数字输出低字节 ADCD [7:0] = ADCDL [7:0].

ASHM: ADC 取样与维持时间模式 选择位:

ASHM=0: ADC 取样时间由硬件控制

ASHM=1: ADC 取样时间由软体控制

ASHC: ADC 取样与保持控制位. 该寄存器动作于ASHM=1.

ASHC=0: 禁能ADC取样.

ASHC=1: 使能ADC 取样.

ADCCH[3:0]: ADC 通道选择.

ADCCH [3:0]	通道
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	Bandgap 内部参考电压
Others	-

ADJUST = 0:

符号: ADCDH							地址: ADh	
7	6	5	4	3	2	1	0	Reset
ADCD[11:4]							00H	

符号: ADCDL							地址: AEh	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	ADCD[3:0]			00H	

ADJUST = 1:

符号: ADCDH							地址: ADh	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	ADCD[11:8]			00H	

符号: ADCDL							地址: AEh	
7	6	5	4	3	2	1	0	Reset
ADCD[7:0]							00H	

ADCD[11:0]: ADC 数字寄存器.

符号: ADCSH							地址: EFh	
7	6	5	4	3	2	1	0	Reset
ADCSH[7:0]							00H	

ADCSH [7:0]: ADC 取样与维持时间寄存器, 作用在于调整取样时间. 而该寄存器动作于 ASHM=0.

符号: **ADCCS**

 地址: **AFh**

7	6	5	4	3	2	1	0	Reset
ADCBGE	-	-	ADCCS[4:0]				00H	00H

ADCBGE: ADC 测量 MCU 内部参考电压使能位。(内部参考电压 1.05V±2%)

0: 禁能。

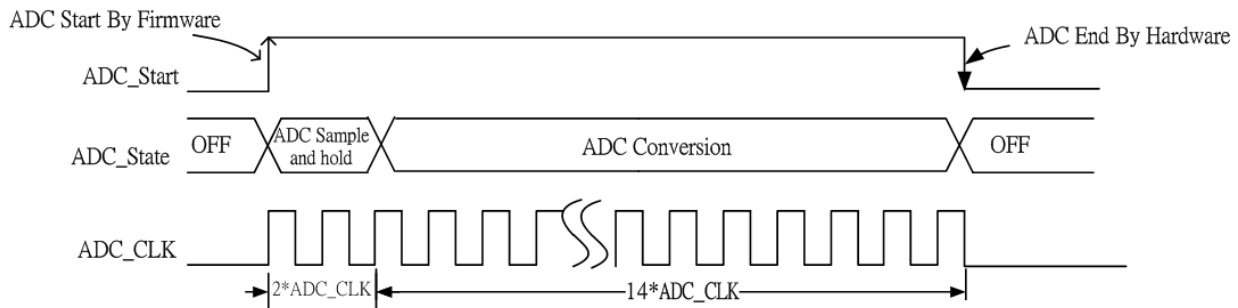
1: 始能。

ADCCS[4:0]: ADC 时钟选择。

$$ADC_Clock = \frac{F_{clk}}{6 \times (ADCCS[4:0] + 1)}$$

If ASHM=0, ADCSH=0x00:

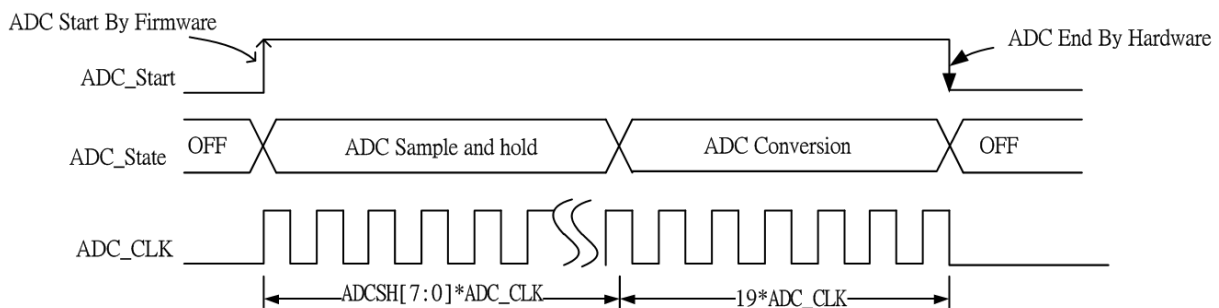
$$ADC_Conversion_Rate = \frac{ADC_Clock}{16}$$



If ASHM=0, ADCSH≠0x00:

$$ADC_Conversion_Rate = \frac{ADC_Clock}{19 + ADCSH[7:0]}$$

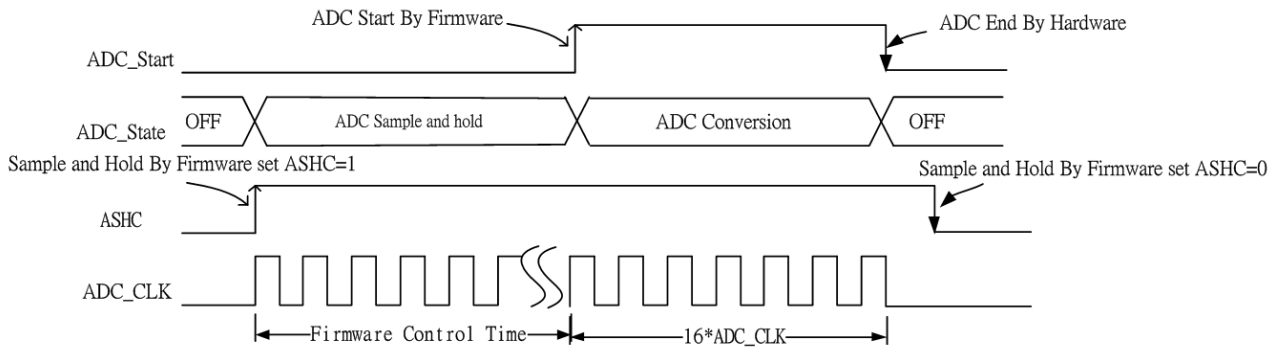
※Sample and hold time by hardware control decisions.



If ASHM=1:

$$ADC_Conversion_Rate = \frac{ADC_Clock}{16} + F/W_Sampling_Time$$

※Sample and hold time by firmware control decisions.



Mnemonic: ADCCAL						Indirect Address: FFFFH		
7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	CALEN	CALST	02H

CALEN: ADC 校正使能位

0: 禁能

1: 始能

※如果 CALEN 和 ADCxEN 都设定为“1”，ADC 根据校准参数执行转换。这将花一些时间转换完成 ADC 转换。

CALST: ADC 校正启动位

0: 校正完成(硬件将自动清除为 0 指示校正已完成)

1: 启动校正(软件设定为 1 开始校正)

MCU 开始 Work 后，仅有初次启动 ADC 转换时，H/W 会先做一次 ADC 校正，校正完后再开始执行 ADC 转换，之后将不再主动做 ADC 转换，除非 MCU 有发生复位。

校正范例程序:

C 语言:

```

unsigned char xdata ADCCAL _at_ 0xFFFF;
void ADC_Calibration(bit bEnable)
{
    if(bEnable==1)
    {
        ADCCAL=0x00; // Set ADCCAL=0x00;
        ADCCAL=0x03; // Set ADCCAL=0x03;
        while((ADCCAL&0x01)!=0x01); // End wait for ADC calibration
    }
}
void main(void)
{
    :
    :
    ADC_Calibration(1); //ADC Calibration
}
    
```

汇编语言:

```
ADCCAL equ 0xFFFF
```

```
:
ADC_Calibration:
  mov DPTR,#ADCCAL
  mov A,#0x00
  movx @DPTR,A      ;set ADCCAL=0x00;
  mov A,#0x03
  movx @DPTR,A      ;set ADCCAL=0x03;
WAIT:
  movx A,@DPTR
  anl A,#0x01
  jnz WATT           ; End wait for ADC calibration
  ret
```

18. 在系统编程(Internal ISP)

OB38R08A1可通过内部硬件电路产生可编程存储器中控制讯号.用户利用可编程存储器中控制寄存器,可编程存储器中地址寄存器和可编程存储器中数据寄存器在不需要将OB38R08A1移出系统的情况下实现ISP功能. OB38R08A1提供了可以实现可编程存储器中编程/芯片擦除/页擦除/保护功能的可编程存储器中控制讯号.用户需要去设计和使用任何OB38R08A1可以输入数据的接口,然后利用ISP服务程序去实现可编程存储器中的编程/芯片擦除/页擦除/保护功能.

18.1 ISP 服务程序

ISP服务程序是由用户自行开发并放置于ISP服务程序区的软件. 用户可依需求决定ISP服务程序的大小. 另外用户需将ISP服务程序烧录至OB38R08A1芯片中才可使用ISP功能.

由于ISP服务程序是由用户自行开发, 它需包括可编程存储器中数据寄存器的烧录及OB38R08A1芯片与上位机的通讯协议. 举例来说, 使用者使用OB38R08A1芯片上之UART接口与上位机作数据之传送及接收, 则OB38R08A1芯片上之ISP服务程序内需有波特率设定, 为避免数据传输错误需有数据总合检查或同位检查或其它错误检查机制.

于正常动作模式或空闲模式可启动 ISP服务程序, 于停止模式则不行.

18.2 锁定位(N)

锁定位N有两个功能:一是配置服务程序的大小,另一个则是通过可编程存储器中擦除功能锁定ISP服务程序的空间.

ISP服务程序地址位从\$1C00到\$1FFF. 可被分为N*128字节(N=0-8). 当N为0时没有ISP功能.所有的8KB字节可编程存储器中都被用来做程序存储.当N为1时,ISP服务程序占用128字节,剩余的7.875K字节可编程存储器中可被用做程序存储.ISP服务程序的最大空间为1K字节(当N为8时). 在这样的配置下,可用的程序存储空间为7K 字节.

在N定下来后,OB38R08A1会从上往下(从最顶端到地址\$1FFF)保存ISP服务程序空间.ISP服务程序的开始地址位于\$1x00 (x为任意数字,以N为准,如表 18-1所示). 请参考章节3.1有关ISP服务程序与程序内存之关系图.

锁定位N功能不同于可编程存储器中保护功能,可编程存储器中擦除功能可以擦除除ISP服务程序空间外的所有可编程存储器中记忆体.如可编程存储器中没有被保护,ISP服务程序的内容一直可被读.如可编程存储器中被保护,所有的可编程存储器中程序空间包括ISP服务程序空间不能被读.

表 18-1 ISP code area.

N	ISP service program address
0	No ISP service program
1	128 bytes (\$1F80h ~ \$1FFFh)
2	256 bytes (\$1F00h ~ \$1FFFh)
3	384 bytes (\$1E80h ~ \$1FFFh)
4	512 bytes (\$1E00h ~ \$1FFFh)
5	640 K bytes (\$1D80h ~ \$1FFFh)
6	768 K bytes (\$1D00h ~ \$1FFFh)
7	896 K bytes (\$1C80h ~ \$1FFFh)
8	1.0 K bytes (\$1C00h ~ \$1FFFh)

ISP 服务程序配置于 N*128 字节 (N= 0 ~ 8)

18.3 对ISP服务程序编程

在锁定位N被设置并ISP服务程序被编程后,ISP服务程序记忆体被自动的保护(锁定).锁定位N有它自己的编程/擦除时序.这不同于可编程存储器中记忆体的编程/擦除时序.因此,被锁定的ISP服务程序不可被可编程存储器中擦除功能所擦除.如客户需要擦除锁定的ISP服务程序,他只可以通过烧写器实现.当OB38R08A1在系统时,客户不可改变ISP服务程序.

18.4 启动ISP服务程序

要启动ISP服务程序就是要在ISP服务程序的开始地址装载程序计数(PC)并执行它.有四种实现的方法:

- (1) 空复位.用首地址为空(\$0000=#0FFH)的硬件复位会在ISP服务程序的开始地址装载PC.硬件复位包括内部(电源或重启)和外部重置端口复位.
- (2) 执行JUMP指令可在ISP服务程序的开始地址装载PC
- (3) 通过硬件设置进入ISP服务程序.用户可通过设置当硬件复位时, P1.6'为低', OB38R08A1 会强行进入ISP服务程序.硬件复位包括内部(电源或重启)和外部键盘复位.在应用系统设计时,用户因该对P1.6 在复位时间内防止OB38R08A1 进入ISP服务程序.
- (4) 通过硬件设置进入ISP服务程序,P1.1 在硬件复位时会侦测到 2 个时钟讯号. 硬件复位包括内部(电源或重启)和外部阻容复位

在硬件复位周期内, 硬件可以侦测P1.6/P1.1状态.如果他们满足了上面的其中一个条件,芯片将会自动跳转到ISP模式.在ISP服务程序被执行后,用户需要对OB38R08A1进行复位,通过硬件复位或WDT,或是'JUMP'至地址\$0000来实现重启程序.

为了用户的不同应用情况,这里有6种不同的进入机制.这个进入方法可以在烧路器或ISP中选取.

- (1) 首地址为空例如\$0000 = 0xFF. 且由内部重置信号触发.
- (2) 首地址为空例如\$0000 = 0xFF. 且由外部重置信号触发.
- (3) P1.6 = 0. 且由内部重置信号触发.
- (4) P1.6 = 0. 且由外部重置信号触发.
- (5) P1.1 输入 2 个时钟. 且由内部重置信号触发.
- (6) P1.1 输入 2 个时钟. 且由外部重置信号触发.

18.5 ISP寄存器– TAKEY, IFCON, ISPF AH, ISPF AL, ISPF D and ISPF C

符号	描述	地址	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
ISP function											
TAKEY	Time Access Key register	F7h	TAKEY [7:0]								00H
IFCON	Interface Control register	8Fh	-	CDPR	-	-	-	-	-	ISPE	00H
ISPF AH	ISP Address – High register	E1h	ISPF AH [7:0]								FFH
ISPF AL	ISP Address – Low register	E2h	ISPF AL [7:0]								FFH
ISPF DH	ISP High Data register	EBh	ISPF DH [7:0]								FFH
ISPF DL	ISP Low Data register	E3h	ISPF DL [7:0]								FFH
ISPF C	ISP Control register	E4h	EMF1	-	EMF3	EMF4	-	ISPF.2	ISPF.1	ISPF.0	00H

地址: **F7H**

符号: **TAKEY**

7	6	5	4	3	2	1	0	Reset
TAKEY [7:0]								00H

ISP使能位(ISPE)默认为只读,软件必须连续的对TAKEY寄存器写三个特定值55h,AAh,和5Ah,使ISPE位可写.这是:

```
MOV TAKEY, #55h
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah
```

地址: **8FH**

符号: **IFCON**

7	6	5	4	3	2	1	0	Reset
-	CDPR	-	-	-	-	-	ISPE	00H

位 0(ISPE) of IFCON 是 ISP 使能位,用户可通过设置 ISPE 位为 1 使能 OB38R08A1 的全部 ISP 功能,通过设置 ISPE 为 0 来禁止全部 ISP 功能.ISPE 的作用像一个加密匙,用户可禁止所有的 ISP 功能以保护软件程序不被意外的擦除.ISP 寄存器 ISPF AH,ISPF AL,ISPF D,与 ISPF C 默认为只读.软件必须将 ISPE 位设为 1 以使上述 4 个寄存器为可写.

地址: **E1H**

符号: **ISPF AH**

7	6	5	4	3	2	1	0	Reset
-	-	ISPF AH5	ISPF AH4	ISPF AH3	ISPF AH2	ISPF AH1	ISPF AH0	FFH

ISPF AH [5:0]: 使用 ISP 功能之地址高字节

地址: **E2H**

符号: **ISPF AL**

7	6	5	4	3	2	1	0	Reset
ISPF AL7	ISPF AL6	ISPF AL5	ISPF AL4	ISPF AL3	ISPF AL2	ISPF AL1	ISPF AL0	FFH

ISPFAL [7:0]: 使用 ISP 功能之地址低字节

ISPFAL 与 ISPFAL 提供了 14 位可编程存储器中记忆体地址用做 ISP 功能.这个可编程存储器中记忆体地址不应该包括 ISP 服务程序空间地址.若 ISPFAL & ISPFAL 寄存器控制的 可编程存储器中记忆体地址覆盖了 ISP 服务程序空间的地址.这个可编程存储器中程序/页擦除等 ISP 功能的执行将会无效.

符号: ISPFDH								地址: EBH
7	6	5	4	3	2	1	0	Reset
ISPFDH7	ISPFDH6	ISPFDH5	ISPFDH4	ISPFDH3	ISPFDH2	ISPFDH1	ISPFDH0	FFH

ISPFDH [7:0]: 使用 ISP 功能之数值.

ISPFDH 寄存器提供 ISP 功能中所需之 8 位数据寄存器.

符号: ISPFDL								地址: E3H
7	6	5	4	3	2	1	0	Reset
ISPFDL7	ISPFDL6	ISPFDL5	ISPFDL4	ISPFDL3	ISPFDL2	ISPFDL1	ISPFDL0	FFH

ISPFDL [7:0]: ISP data for ISP function.

ISPFDL 寄存器提供 ISP 功能中所需之 8 位数据寄存器.

符号: ISPFC								地址: E4H
7	6	5	4	3	2	1	0	Reset
EMF1	-	EMF3	EMF4	-	ISPF[2]	ISPF[1]	ISPF[0]	00H

EMF1: 进入机制 (1) 标志位, 由复位信号清除. (仅读)

EMF3: 进入机制 (3) 标志位, 由复位信号清除. (仅读)

EMF4: 进入机制 (4) 标志位, 由复位信号清除. (仅读)

ISPF [2:0]: ISP 功能选择位.

ISPF[2:0]	ISP function
000	1-byte program ROM
001	2-byte program ROM
010	Chip protect
011	Write EEPROM
100	Read EEPROM
101	Write option
110	Read option
111	reserved

选择的 ISP 功能在软件对 ISPFC 寄存器写入数据时将会执行一次.

范例. 可编程存储器中地址: \$XYMN

将对地址\$XY00 to \$XYFF 执行页抹除

要实现芯片擦除 ISP 功能,OB38R08A1 将擦除除 ISP 服务程序外的所有可编程存储器中程序记忆体. 要实现芯片保护 ISP 功能,OB38R08A1 内容将被读为#00H.

ISPF[2:0]=000 (1-byte program ROM)

EX1: 将对程序记忆体地址\$1004H 执行字节写, 写入#22H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah           ; enable ISPE write attribute
MOV IFCON, #01H          ; enable ISP function
MOV ISPF AH, #10H        ; set flash address-high, 10H
MOV ISPF AL, #04H        ; set flash address-low, 04H
MOV ISPF DL, #22H        ; set flash data to be programmed, data = 22H
MOV ISPF C, #00H         ; start to program #22H to the flash address $1004H
```

ISPF[2:0]=000 (1-byte program ROM)

EX2: 将对程序记忆体地址\$1005H 执行字节写, 写入#33H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah           ; enable ISPE write attribute
MOV IFCON, #01H          ; enable ISP function
MOV ISPF AH, #10H        ; set flash address-high, 10H
MOV ISPF AL, #05H        ; set flash address-low, 05H
MOV ISPF DL, #33H        ; set flash data to be programmed, data = 33H
MOV ISPF C, #00H         ; start to program #33H to the flash address $1005H
```

ISPF[2:0]=001 (2-byte program ROM)

EX3: 将对程序记忆体地址\$1004H 执行字节写, 写入#3322H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah           ; enable ISPE write attribute
MOV IFCON, #01H          ; enable ISP function
MOV ISPF AH, #10H        ; set flash address-high, 10H
MOV ISPF AL, #04H        ; set flash address-low, 04H
MOV ISPF DL, #22H        ; set flash data to be programmed, data = 22H
MOV ISPF DH, #33H        ; set flash data to be programmed, data = 33H
MOV ISPF C, #01H         ; start to program #3322H to the flash address $1004H
```

ISPF[2:0]=011 (EEPROM write)

EX4: 将对 EEPROM 地址\$0004H 执行字节写, 写入#22H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah           ; enable ISPE write attribute
MOV IFCON, #01H          ; enable ISP function
MOV ISPF AH, #00H        ; set flash address-high, 00H
MOV ISPF AL, #04H        ; set flash address-low, 04H
MOV ISPF DL, #22H        ; set flash data to be programmed, data = 22H
MOV ISPF C, #03H         ; start to program #22H to the EEPROM address $0004H
```

ISPF[2:0]=011 (EEPROM write)

EX5: 将对 EEPROM 地址\$0005H 执行字节写，写入#33H

```

MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah           ; enable ISPE write attribute
MOV IFCON, #01H          ; enable  ISP function
MOV ISPFAH, #00H         ; set flash address-high, 00H
MOV ISPFAH, #05H         ; set flash address-low, 05H
MOV ISPFDL, #33H         ; set flash data to be programmed, data = 33H
MOV ISPFC, #03H          ; start to program #33H to the EEPROM address $0005H
  
```

ISPF[2:0]=100 (EEPROM read)

EX6: 将对 EEPROM 地址\$0004H 执行字节读取

```

MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah           ; enable ISPE write attribute
MOV IFCON, #01H          ; enable  ISP function
MOV ISPFAH, #00H         ; set flash address-high, 00H
MOV ISPFAH, #04H         ; set flash address-low, 04H
MOV ISPFDL, #04H         ; start to program read the EEPROM address $0004H to ISPFDL
MOV A, ISPFDL             ; read ISPFDL and store it in Accumulator
  
```

ISPF[2:0]=101 (write option)

EX7: 将对 option 地址\$0004H 执行字节写，写入#22H

```

MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah           ; enable ISPE write attribute
MOV IFCON, #01H          ; enable  ISP function
MOV ISPFAH, #00H         ; set flash address-high, 00H
MOV ISPFAH, #04H         ; set flash address-low, 04H
MOV ISPFDL, #22H         ; set flash data to be programmed, data = 22H
MOV ISPFC, #05H          ; start to program #22H to the flash address $0004H
  
```

ISPF[2:0]=101 (write option)

EX8: 将对 option 地址\$0005H 执行字节写，写入#33H

```

MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah           ; enable ISPE write attribute
MOV IFCON, #01H          ; enable  ISP function
MOV ISPFAH, #00H         ; set flash address-high, 00H
MOV ISPFAH, #05H         ; set flash address-low, 05H
MOV ISPFDL, #33H         ; set flash data to be programmed, data = 33H
MOV ISPFC, #05H          ; start to program #33H to the flash address $0005H
  
```

ISPF[2:0]=110 (read option)

EX9: 将对 option 地址\$0004H 执行字节读取

```

MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah           ; enable ISPE write attribute
MOV IFCON, #01H          ; enable  ISP function
MOV ISPFAH, #00H         ; set flash address-high, 00H
MOV ISPFAH, #04H         ; set flash address-low, 04H
MOV ISPFDL, #06H         ; start to program read the option address $0004H to ISPFDL
MOV A, ISPFDL             ; read ISPFDL and store it in Accumulator
  
```

EX10: 将对程序记忆体地址地址\$1004H 执行字节读取


```
CLR A
MOV DPTR,#1004h      ;
MOVC A,@A+DPTR      ; read the data of address $1004 and store it in Accumulator
```

EX11: 将对程序记忆体地址地址\$1005H 执行字节读取

```
CLR A
MOV DPTR,#1005h      ;
MOVC A,@A+DPTR      ; read the data of address $1005 and store it in Accumulator
```

19. 比较器(Comparator)

OB38R08A1在片内集成了两组比较器.当比较器使用时,比较器的非反向输入脚大于反向输入脚时,比较器输出脚置高,反之比较器输出脚置低.

Comparator SFRs 如下所示:

符号	描述	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	重置值
Comparator											
OpPin	OpCmp Pin Select	F6h	-	Cmp0_En	C0PosVBG	C0PosPad	-	Cmp1_En	C1PosVBG	C1PosPad	00h
Cmp0CON	Comparator_0 control	FEh	Hys0En	Cmp0o	CMF0MS[1:0]		CMF0	Cmp0OutEN	-	-	00h
Cmp1CON	Comparator_1 control	FFh	Hys1En	Cmp1o	CMF1MS[1:0]		CMF1	Cmp1OutEN	-	-	00h

符号: OpPin								地址: F6h	
7	6	5	4	3	2	1	0	Reset	
-	Cmp0_En	C0PosVBG	C0PosPad	-	Cmp1_En	C1PosVBG	C1PosPad	00h	

Cmp0_En : 比较器 0 使能位.

1: 比较器 0 电路使能,并自动将多功能之 I/O P0.3/P0.4/P0.6 切换到比较器 0 相应的信号.

C0PosVBG : 非反向输入端参考内部固定电压(1.23V±1%)使能位

1: 使能

C0PosPad: 非反向输入端电压参考外部输入脚电压使能位

1: 使能

Cmp1_En : 比较器 1 使能位.

1: 比较器 1 电路使能,并自动将多功能之 I/O P0.0/P0.1/P0.2 切换到比较器 1 相应的信号.

C1PosVBG: 非反向输入端参考内部固定电压(1.23V±1%)使能位

1: 使能

C1PosPad: 非反向输入端电压参考外部输入脚电压使能位

1: 使能

CmpxOut、Cmpx_En、CxPosVBG及CxPosPad 设定参考表:

Cmpx_En	CxPosPad	CxPosVBG	CmpxOut_En	Comparator		
				CmpxPin	CmpxNIn	CmpxOUT
0	X	X	X	IO	IO	IO
1	0	1	0	IO	CMP	IO
1	0	1	1	IO	CMP	CMP
1	1	0	0	CMP	CMP	IO
1	1	0	1	CMP	CMP	CMP

符号: Cmp0CON							地址: FEh	
7	6	5	4	3	2	1	0	Reset
Hys0En	Cmp0o	CMF0MS[1:0]		CMF0	Cmp0 OutEN	-	-	00h

Hys0En: 比较器 0 迟滞功能使能位

0: 禁能

1: 使能

Cmp0o: 比较器 0 结果输出位(仅读)

1: 非反向输入端电压高于反向输入端电压

0: 反向输入端电压高于非反向输入端电压

CMF0MS[1:0]: 选择比较器 0 旗标(CMF0)产生之模式

00: 比较器 0 旗标(CMF0)于比较器 0 输出转换时被置高

01: 比较器 0 旗标(CMF0)于比较器 0 输出为上升沿时被置高

10: 比较器 0 旗标(CMF0)于比较器 0 输出为下降沿时被置高

11: 无动作

Cmp0OutEN: 比较器 0 输出使能位

0: 比较器 0 输出至外部管脚

1: 比较器 0 不输出至外部管脚

CMF0: 比较器 0 旗标位

此位依 CMF0MS[1:0]之选择模式条件符合时被置高.此位需由软件清零.

符号: Cmp1CON							地址: FFh	
7	6	5	4	3	2	1	0	Reset
Hys1En	Cmp1o	CMF1MS[1:0]		CMF1	Cmp1 OutEN	-	-	00h

Hys1En: 比较器 1 迟滞功能使能位

0: 禁能

1: 使能

Cmp1o: 比较器 1 结果输出位(仅读)

1: 非反向输入端电压高于反向输入端电压

0: 反向输入端电压高于非反向输入端电压

CMF1MS[1:0]: 选择比较器 1 旗标(CMF1)产生之模式

00: 比较器 1 旗标(CMF1)于比较器 1 输出转换时被置高

01: 比较器 1 旗标(CMF1)于比较器 1 输出为上升沿时被置高

10: 比较器 1 旗标(CMF1)于比较器 1 输出为下降沿时被置高

11: 无动作

CMF1: 比较器 1 旗标位

此位依 CMF1MS[1:0]之选择模式条件符合时被置高.此位需由软件清零.

Cmp1 OutEN: 比较器 1 输出使能位

0: 比较器 1 输出至外部管脚

1: 比较器 1 不输出至外部管脚

工作环境

Symbol	Description	Min.	Typ.	Max.	Unit.	Remarks
TA	Operating temperature	-40	25	85	°C	Ambient temperature under bias
VDD	Supply voltage	2.4	-	5.5	V	

DC 电气特性

TA = -40°C to 85°C, Vcc = 2.4~5.0V

Symbol	Parameter	Valid	Min	Typical	Max	Units	Conditions
VIL1	Input Low-voltage	Port 0,1,3			0.2Vcc	V	
VIL2	Input Low-voltage	XTAL1			0.2Vcc	V	-
VIH1	Input High-voltage	Port 0,1,3	0.8Vcc			V	-
VIH2	Input High-voltage	XTAL1	0.8Vcc			V	-
IOL1	Sink Current (Open drain)	P0.0/P0.1/P0.2/P0.3 P0.4/P0.6/P1.0/P1.1 P1.4	20			mA	VOL=0.45V VCC=5V
			17			mA	VOL=0.45V VCC=3.3V
IOL2	Sink Current (Open drain)	P0.5/P0.7/P1.2/P1.3 P1.5/P1.6/P1.7/P3.0 P3.1	40			mA	VOL=0.45V VCC=5V
			28			mA	VOL=0.45V VCC=3.3V
IOH1	Source Current (Pull-Up)	Port 0,1,3	0.3			mA	VOH=2.6V VCC=5V
			0.09			mA	VOH=2.4V VCC=3.3V
IOH2	Source Current (Push-Pull)	P0.0/P0.1/P0.2/P0.3 P0.4/P0.6/P1.0/P1.1 P1.4	7			mA	VOH=4.6V VCC=5V
			5			mA	VOH=2.9V VCC=3.3V
IOH3	Source Current (Push-Pull)	P0.5/P0.7/P1.2/P1.3 P1.5/P1.6/P1.7/P3.0 P3.1	10			mA	VOH=4.6V VCC=5V
			7			mA	VOH=2.9V VCC=3.3V
CIO	Pin Capacitance	-	-	-	10	pF	Freq= 1MHz, Ta= 25°C
ICC	Power Supply Current	VCC=5V	-	5	6	mA	Active mode, IRC=16MHz 25 °C
			-	4	5	mA	Idle mode, IRC=16MHz 25 °C
			-	2	5	uA	Power down mode 25 °C

LVI& LVR 电气特性

	LVR		
	Min	Typical	Max
2.0V	VIL=1.84V	VIL=2.00V	VIL=2.16V
2.2V	VIL=2.024V	VIL=2.20V	VIL=2.376V
2.4V	VIL=2.208V	VIL=2.40V	VIL=2.592V
3.0V	VIL=2.76V	VIL=3.00V	VIL=3.24V

	LVI		
	Min	Typical	Max
LVIS[1:0] = 00	VIL=2.208V	VIL=2.40V	VIL=2.592V
LVIS[1:0] = 01	VIL=2.392V	VIL=2.60V	VIL=2.808V
LVIS[1:0] = 10	VIL=2.944V	VIL=3.20V	VIL=3.456V
LVIS[1:0] = 11	VIL=3.68V	VIL=4.00V	VIL=4.32V

ADC 电气特性

	Symbol	Test Condition	MIN	TYP	MAX	Unit
Operation	VDD	VDD	3		5.5	V
Resolution					12	bit
Conversion time				16t _{ADC}		us
Sample rate				167k		Hz
Integral Non-Linearity Error	INL		-2		2	LSB
Differential Non-Linearity	DNL		-1		1	LSB
Clock frequency	ADCCLK				4	MHz

Comparator 电气特性

Symbol	Description	Test Condition		MIN	TPY	MAX	Unit
		V5V	Condition				
I _{OP}	Operating current	5	-	-	-	2	mA
-	Power Down Current	5	-	-	-	0.1	nA
-	Offset voltage	5	-	-10	-	+10	mV
V	Input voltage range	-	-	V _{SS}	-	V5V	V
T _p	Propagation delay	5	△ V _{in} =10mV	-	4	6	us

单击下面可查看定价，库存，交付和生命周期等信息

[>>On-Bright](#)