

# AS128A用户手册

深圳市全智芯科技有限公司

Shenzhen Aschip Tech Co.,Ltd

产品目录 .....	3
描述 .....	3
特征 .....	3
各封装引脚配置 .....	4
管脚描述 .....	7
特殊功能寄存器 (SFR) .....	9
功能描述 .....	15
1. 总特征 .....	15
1.1 嵌入式程序存储器 .....	15
1.2 IO 口 .....	15
1.3 指令时钟周期选择 .....	15
1.4 时钟输出选择 .....	15
1.5 复位 .....	16
1.5.1 硬件复位功能 .....	16
1.5.2 软件复位功能 .....	16
1.5.3 Reset status .....	16
1.5.4 Time Access Key register (TAKEY) .....	17
1.5.5 软件复位寄存器 (SWRES) .....	17
1.5.6 Example of software reset .....	17
1.6 时钟源 .....	18
2. 指令设置 .....	19
3. 存储器结构 .....	23
3.1 程序存储器 .....	23
3.2 数据存储器 .....	24
3.3 数据内存-低 128 字节 (00h 到 7Fh) .....	24
3.4 数据存储器-高 128 字节 (80h 到 FFh) .....	24
3.5 存储器-扩展的 1024 字节 (\$00 到 \$ 3FF) .....	24
4. CPU 结构 .....	26
4.1 累加器 .....	26
4.2 B 寄存器 .....	26
4.3 程序状态字 .....	27
4.4 堆栈指针 .....	27
4.5 数据指针 .....	27
4.6 时钟控制寄存器 .....	28
4.7 接口控制寄存器 .....	28
5. GPIO 管脚型态 .....	29
6. 定时器 0、定时器 1、定时器 3 .....	31
6.1 定时器/计数器模式控制寄存器 .....	31
6.2 定时器/计数器控制寄存器 (TCON0) .....	32
6.3 增强中断型态寄存器 (ENHIT) .....	34
6.4 外部中断 deglitch 寄存器 (INTDEG) .....	34
6.5 定时器输入频率控制寄存器 .....	35
6.6 模式 0 (13 位定时/计数) .....	35
6.7 模式 1 (16 位定时/计数) .....	36
6.8 模式 2 (8 位自动重载定时/计数) .....	36
6.9 模式 3 (两个独立 8 位定时/计数 (仅定时器 0)) .....	37
7. 定时器 2 以及捕捉/比较单元 (CCU) .....	38
7.1 定时器 2 功能 .....	41
7.1.1 定时器模式 (Timer mode) .....	41

7.1.2	外部信号计数模式 (Event counter mode)	42
7.1.3	外部信号定时器模式 (Gated timer mode)	42
7.1.4	定时器 2 的重载	42
7.2	比较功能	43
7.2.1	比较模式 0	43
7.2.2	比较模式 1	43
7.3	捕获功能	44
7.3.1	捕捉模式 0	44
7.3.2	捕捉模式 1	44
8.	串行接口	45
8.1	串行接口由以下 4 种模式可以设置	46
8.1.1	模式 0	46
8.1.2	模式 1	47
8.1.3	模式 2	47
8.1.4	模式 3	47
8.2	串行接口的多重机通讯	48
8.3	输入频率控制寄存器	48
8.4	波特率发生器	48
8.4.1	串行接口的模式 1 和 3	48
9.	看门狗定时器	50
9.1	看门狗复位定时器	52
9.2	看门狗中断定时器	53
10.	中断	55
10.1	优先权配置	58
11.	电源管理单元	60
11.1	待机模式 (空闲模式)	60
11.2	停止模式	60
12.	IIC 功能	61
12.1	IIC interface 0	61
12.2	IIC interface 1	65
13.	12 位模拟数字转换器 (ADC)	69
14.	LVI & LVR - 低压侦测中断及低压侦测复位	73
15.	触摸按键单元	75
15.1	触摸按键简介	75
15.2	触摸按键特性	75
15.3	寄存器设定	75
16.	脉宽调制器 (PWM)	86
17.	LED 驱动器	92
17.1	LED Driver 简介	92
17.2	LED Driver 特性	92
17.3	LED 应用说明	93
17.3.1	LED 模块设定方式	93
17.3.2	SFR LEDCLK 设定说明	93
17.4	寄存器设定	94
17.5	Display RAM Map	98
17.6	LED 应用电路参考	99
18.	EEPROM	101
	工作环境	104
	DC 电气特性	104
	ADC CHARACTERISTICS	105

## 产品目录

AS128A-28E  
AS128A-20E  
AS128A-16E

## 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

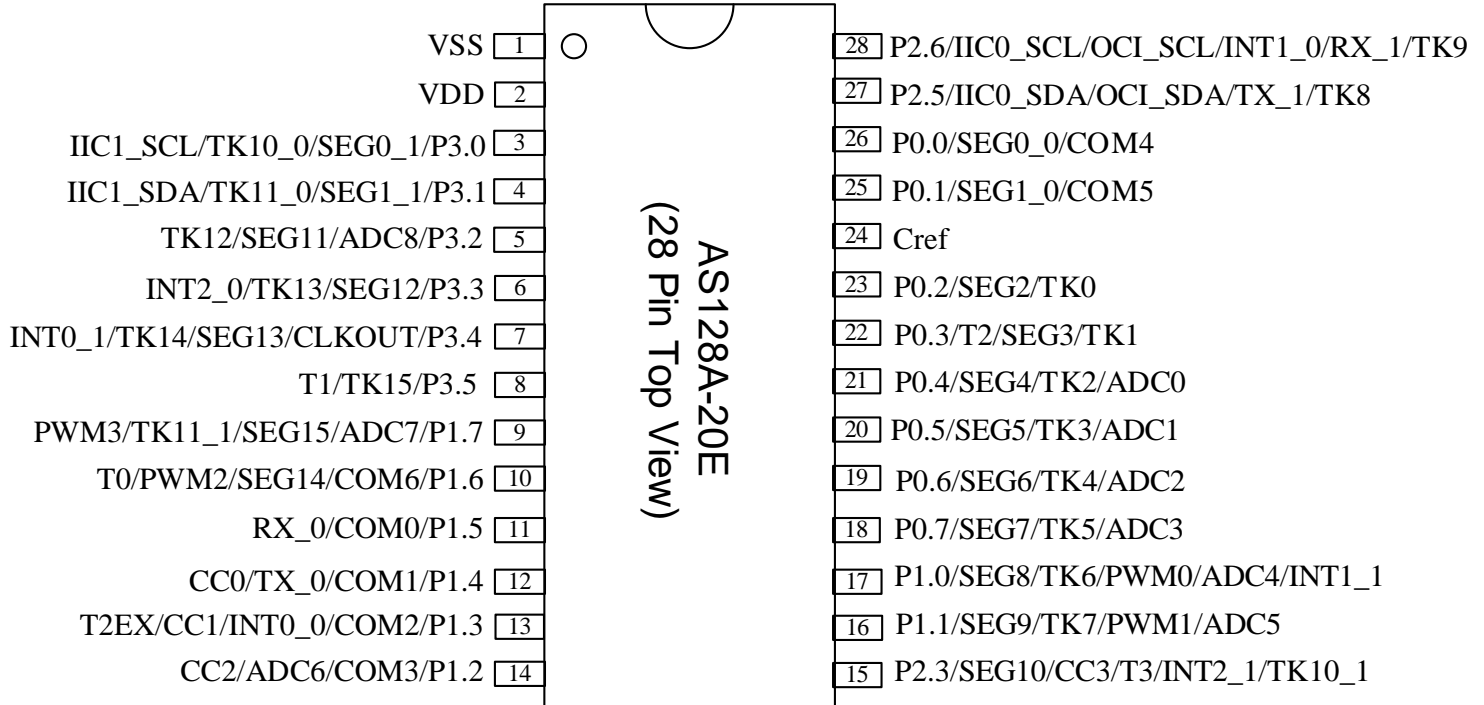
该 AS128A 是一个快速的单芯片 8 位微控制器内核.这是一个全功能的 8 位嵌入式控制器,执行所有 ASM51 指令,具有与 MCS - 51 相同的指令设置

## 特征

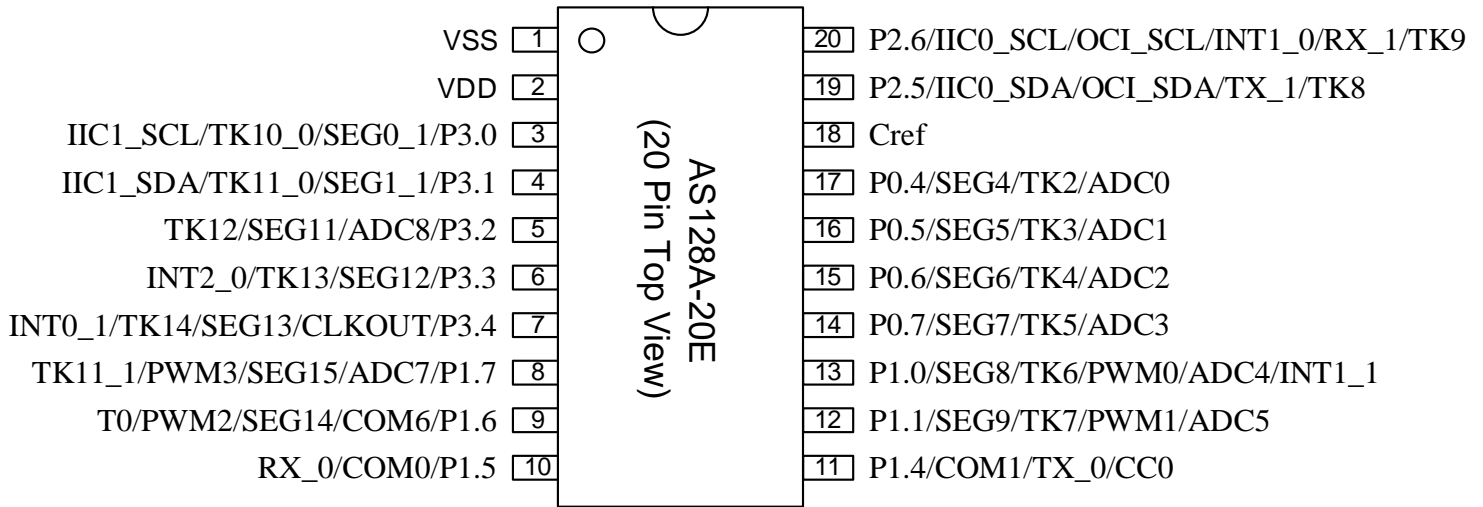
工作电压: 2.4V ~ 5.5V  
 高速 1T 架构,最高可达 16MHz  
 1~8T 模式可使用软件编程  
 指令设置兼容 MCS-51  
 内置 16MHz RC 振荡器,及可程序化的分频器  
 16KB 字节的片上可编程存储器中程序存储器  
 256B 字节的标准的 8052 RAM  
 1024B 字节 SRAM  
 一个全双工通信的串行接口.附加波特率产生器  
 集成 16 路触摸按键  
 低功耗节电模式唤醒  
 4 个 16-bit 的定时器/计数器 (计时器 0, 1, 2, 3)  
 25 GPIOs, 可选择四种型态(准双向口、推挽、开漏、只输入),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 &外部中断 2  
 可编程的看门狗定时器 (WDT)  
 2 个 IIC 接口 (主/从机模式)  
 4 路具有死区时间 10bit 脉宽调制(PWM)  
 4 路 16bit 比较(PWM)/捕获/重载功能  
 9 路 12bit 模拟数字转换 (ADC)  
 LED 驱动: 7 个 大电流 COM 口, 16 个 SEG 口  
 片上存储器支持 ICP 及 EEPROM 功能  
 片上在线仿真功能(ICE)及片上在线调试功能(OCD)  
 低电压中断/低电压复位(LVVI/LVR)  
 3 个外部中断 INTX3 支持 上升沿/下降沿 侦测  
 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

## 各封装引脚配置

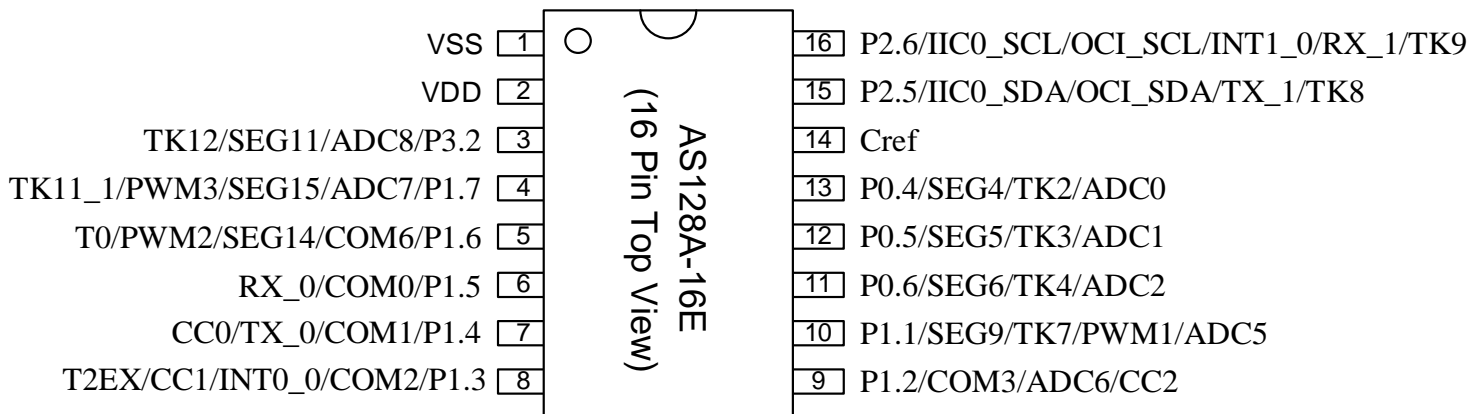
### 28 Pin SOP 300mil/ TSSOP 173mil



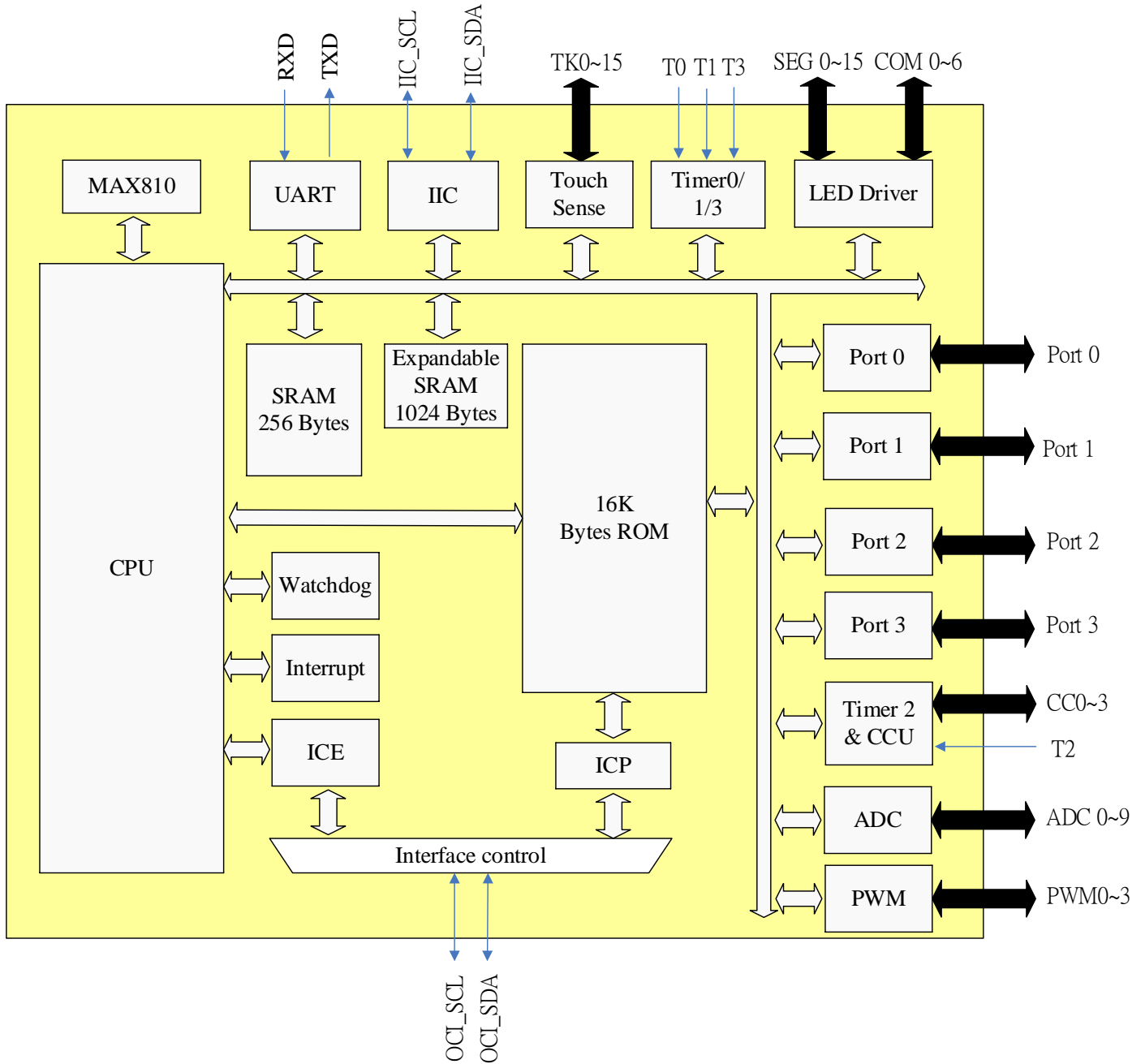
**20 Pin SOP 300mil / TSSOP 173mil**



**16 Pin SOP 150mil**



系统方框图



## 管脚描述

28L	20L	16L	Symbol	I/O	Description
1	1	1	VSS	I	Ground
2	2	2	VDD	I	Power supply
3	3		P3.0/SEG0_1/TK10_0/IIC1_SCL	I/O	Bit 0 of port 3 & LED driver segment 0_1 & Touch key 10_0 & IIC1 SCL
4	4		P3.1/SEG1_1/TK11_0/IIC1_SDA	I/O	Bit 1 of port 3 & LED driver segment 1_1 & Touch key 11_0 & IIC1 SDA
5	5	3	P3.2/ADC8/SEG11/TK12	I/O	Bit 2 of port 3 & ADC input channel 8 & LED driver segment 11 & Touch key 12
6	6		P3.3/SEG12/TK13/INT2_0	I/O	Bit 3 of port 3 & LED driver segment 12 & Touch key 13 & External interrupt 2_0
7	7		P3.4/CLKOUT/SEG13/TK14/INT0_1	I/O	Bit 4 of port 3 & Clock output & LED driver segment 13 & Touch key 14 & External interrupt 0_1
8			P3.5/TK15/T1	I/O	Bit 5 of port 3 & Touch key 15 & Timer 1 external input
9	8	4	P1.7/ADC7/SEG15/PWM3/TK11_1	I/O	Bit 7 of port 1 & ADC input channel 7 & LED driver segment 15 & PWM channel 3 & Touch key 11_1
10	9	5	P1.6/SEG14/PWM2/T0/COM6	I/O	Bit 6 of port 1 & LED driver segment 14 & PWM Channel 2 & Timer 0 external input & LED driver common 6
11	10	6	P1.5/COM0/RX_0	I/O	Bit 5 of port 1 & LED driver common 0 & Serial interface channel_0 receive data
12	11	7	P1.4/COM1/TX_0/CC0	I/O	Bit 4 of port 1 & LED driver common 1 & Serial interface channel_0 transmit data & Timer 2 compare/ capture channel 0
13		8	P1.3/COM2/INT0_0/CC1/T2EX	I/O	Bit 3 of port 1 & LED driver common 2 & External interrupt 0_0 & Timer 2 compare/ capture channel 1 & Timer 2 capture trigger
14		9	P1.2/COM3/ADC6/CC2	I/O	Bit 2 of port 1 & LED driver common 3 & ADC input channel 6 & Timer 2 compare/ capture channel 2
15			P2.3/SEG10/CC3/T3/INT2_1/TK10_1	I/O	Bit 3 of port 2 & LED driver segment 10 & Timer 2 compare/capture channel 3 & Timer 3 external input & External interrupt 2_1 & Touch key 10_1
16	12	10	P1.1/SEG9/TK7/PWM1/ADC5	I/O	Bit 1 of port 1 & LED driver segment 9 & Touch key 7 & PWM channel 1 & ADC input channel 5
17	13		P1.0/SEG8/TK6/PWM0/ADC4/ INT1_1	I/O	Bit 0 of port 1 & LED driver segment 8 & Touch key 6 & PWM channel 0 & ADC input channel 4 & External interrupt 1_1



18	14		P0.7/SEG7/TK5/ADC3	I/O	Bit 7 of port 0 & LED driver segment 7 & Touch key 5 & ADC input channel 3
19	15	11	P0.6/SEG6/TK4/ADC2	I/O	Bit 6 of port 0 & LED driver segment 6 & Touch key 4 & ADC input channel 2
20	16	12	P0.5/SEG5/TK3/ADC1	I/O	Bit 5 of port 0 & LED driver segment 5 & Touch key 3 & ADC input channel 1
21	17	13	P0.4/SEG4/TK2/ADC0	I/O	Bit 4 of port 0 & LED driver segment 4 & Touch key 2 & ADC input channel 0
22			P0.3/SEG3/TK1/T2	I/O	Bit 3 of port 0 & LED driver segment 3 & Touch key 1 & Timer 2 external input
23			P0.2/SEG2/TK0	I/O	Bit 2 of port 0 & LED driver segment 2 & Touch key 0
24	18	14	CREF	I/O	Touch key external capacitor
25			P0.1/SEG1_0/COM5	I/O	Bit 1 of port 0 & LED driver segment 1_0 & LED driver common 5
26			P0.0/SEG0_0/COM4	I/O	Bit 0 of port 0 & LED driver segment 0_0 & LED driver common 4
27	19	15	P2.5/IIC0_SDA/OCI_SDA/TX_1/TK8	I/O	Bit 5 of port 2 & IIC0 SDA & On-Chip Instrumentation Data I/O pin of ICE and ICP Functions & Serial interface channel_1 transmit data & Touch key 8
28	20	16	P2.6/IIC0_SCL/OCI_SCL/INT1_0/RX_1/TK9	I/O	Bit 6 of port 2 & IIC SCL & On-Chip Instrumentation Clock I/O pin of ICE and ICP Functions & External interrupt 1_0 & Serial interface channel_1 receive data & Touch key 9

## 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
<b>F8</b>	IIC0S	IIC0CTL	IIC0A1	IIC0A2	IIC0RWD	IIC0EBT	TKSW	WDTIC	<b>FF</b>
<b>F0</b>	B	IIC1A2		TKEN1	TKSTATUS1	PDGPEN		TAKEY	<b>F7</b>
<b>E8</b>	IIC1S	IIC1CTL	IIC1A1		IIC1RWD	IIC1EBT	INTDEG	ADCSH	<b>EF</b>
<b>E0</b>	ACC	ISPF AH	ISPF AL	ISPF DL	ISPF C	ENHIT	LVC	SWRES	<b>E7</b>
<b>D8</b>	TCON1	PFC ON	P3M0	P3M1	TKFC4	PWMC1	PWMDT0	PWMDT1	<b>DF</b>
<b>D0</b>	PSW	CCEN2	P0M0	P0M1	P1M0	P1M1	P2M0	P2M1	<b>D7</b>
<b>C8</b>	T2CON	CCCON	CRCL	CRCH	TL2	TH2	PWMDL	PWMDH	<b>CF</b>
<b>C0</b>	IRCON	CCEN	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3	<b>C7</b>
<b>B8</b>	IEN1	IP1	SRELH	TKSTATUS0	PWMD0L	PWMD0H	PWMD1L	PWMD1H	<b>BF</b>
<b>B0</b>	P3	PWMD2L	PWMD2H	PWMD3L	PWMD3H	PWMC	WDTRC	WDTK	<b>B7</b>
<b>A8</b>	IEN0	IP0	SRELL	ADCC1	ADCC2	ADCDH	ADCDL	ADCCS	<b>AF</b>
<b>A0</b>	P2	RSTS							<b>A7</b>
<b>98</b>	SCON	SBUF	IEN2	TKCON		TKCHN	TKCDL	TKCDH	<b>9F</b>
<b>90</b>	P1	AUX	T3MOD	TKEN0	TL3	TH3		IRCON2	<b>97</b>
<b>88</b>	TCON0	TMOD	TL0	TL1	TH0	TH1	CKCON	IFCON	<b>8F</b>
<b>80</b>	P0	SP	DPL	DPH			RCON	PCON	<b>87</b>
<b>Hex\Bin</b>	<b>X000</b>	<b>X001</b>	<b>X010</b>	<b>X011</b>	<b>X100</b>	<b>X101</b>	<b>X110</b>	<b>X111</b>	<b>Bin/Hex</b>

## Xdata寄存器映射(External data memory):

### Indirect Access Mode

HexBin	X000	X001	X010	X011	X100	X101	X110	X111	Bin/Hex
FFE8		TKDC	TKRUNTIME	GPUCNT	SKUCNT	TKPSSR			FFE8
FFE0	GKWCNT	SKWCNT	TKRVC	TKFC0	TKFC1	TKFC2	TKFC3		FFE0
FFD8									FFD8
FFD0									FFD0
FFC8	GP4S1	GP4S0	GP5S1	GP5S0	GP6S1	GP6S0	GP7S1	GP7S0	FFC8
FFC0	GP0S1	GP0S0	GP1S1	GP1S0	GP2S1	GP2S0	GP3S1	GP3S0	FFC0
...									...
FF20	LEDCLK	LEDSC0	LEDSC1						FF20
FF18	COM5_CH	COM5_CL	COM6_CH	COM6_CL	COMEN	SEGEN0	SEGEN1	LEDCON	FF18
FF10	COM1_CH	COM1_CL	COM2_CH	COM2_CL	COM3_CH	COM3_CL	COM4_CH	COM4_CL	FF10
FF08	COM4_AH	COM4_AL	COM5_AH	COM5_AL	COM6_AH	COM6_AL	COM0_CH	COM0_CL	FF08
FF00	COM0_AH	COM0_AL	COM1_AH	COM1_AL	COM2_AH	COM2_AL	COM3_AH	COM3_AL	FF00

注：AS128A 特殊功能寄存器的重置值在下表描述

Register	Location	Reset value	Description
<b>SYSTEM</b>			
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 low byte
DPH	83h	00h	Data Pointer high byte
AUX	91h	00h	Auxiliary register
PCON	87h	00h	Power Control
RCON	86h	00h	Internal RAM Control Register
CKCON	8Eh	10h	Clock control register
<b>INTERRUPT &amp; PRIORITY</b>			
IRCON	C0h	00h	Interrupt Request Control Register
IRCON2	97h	00h	Interrupt Request Control Register 2
IEN0	A8h	00h	Interrupt Enable Register 0
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

Register	Location	Reset value	Description
<b>UART</b>			
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
<b>WDT</b>			
RSTS	A1h	00h	Reset status register
WDTRC	B6h	04H	Watchdog Timer Reset Control
WDTIC	FFh	00H	Watchdog Timer Interrupt Control
WDTK	B7h	00h	Watchdog timer refresh key.
TAKEY	F7h	00h	Time Access Key register
<b>PWM</b>			
PWMC	B5h	00H	PWM Control Register
PWMD0H	BDh	00H	PWM 0 Data High Register
PWMD0L	BCh	00H	PWM 0 Data Low Register
PWMD1H	BFh	00H	PWM 1 Data High Register
PWMD1L	BEh	00H	PWM 1 Data Low Register
PWMD2H	B2h	00H	PWM 2 Data High Register
PWMD2L	B1h	00H	PWM 2 Data Low Register
PWMD3H	B4h	00H	PWM 3 Data High Register
PWMD3L	B3h	00H	PWM 3 Data Low Register
PWMDMH	CFh	00H	PWM Max Data High Register
PWMDML	CEh	FFH	PWM Max Data Low Register
PWMC1	DDh	30H	PWM Control 1
PWMDT0	DEh	00H	PWM 0 Dead Time
PWMDT1	DFh	00H	PWM 1 Dead Time
<b>TIMER0/TIMER1/TIMER3</b>			
TCON0	88h	00h	Timer/Counter Control 0
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
TCON1	D8h	00h	Timer/Counter Control 1
T3MOD	92h	00h	Timer 3 Mode Control
TL3	94h	00h	Timer 3, low byte
TH3	95h	00h	Timer 3, high byte
PFCON	D9h	00h	Peripheral Frequency control register
<b>PCA(TIMER2)</b>			

Register	Location	Reset value	Description
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
<b>GPIO</b>			
P0	80h	User define	Port 0
P1	90h	User define	Port 1
P2	A0h	User define	Port 2
P3	B0h	User define	Port 3
P0M0	D2h	User define	Port 0 output mode 0
P0M1	D3h	User define	Port 0 output mode 1
P1M0	D4h	User define	Port 1 output mode 0
P1M1	D5h	User define	Port 1 output mode 1
P2M0	D6h	User define	Port 2 output mode 0
P2M1	D7h	User define	Port 2 output mode 1
P3M0	DAh	User define	Port 3 output mode 0
P3M1	DBh	User define	Port 3 output mode 1
<b>EEPROM</b>			
IFCON	8Fh	00h	Interface control register
ISPF AH	E1h	FFh	EEPROM Address-High register
ISPF AL	E2h	FFh	EEPROM Address-Low register
ISPF DL	E3h	FFh	EEPROM Data High register
ISPF C	E4h	00h	EEPROM control register
TAKEY	F7h	00h	Time Access Key register
<b>TOUCH KEY</b>			
TKEN0	93h	00h	Touch Key Enable 0
TKEN1	F3h	00h	Touch Key Enable 1
TKCON	9Bh	00h	Touch Key Control
TKRUNTIME	FFE AH	00h	Touch Key Running Time
TKCHN	9Dh	00h	Touch Key Channel Number
TKCDL	9Eh	00h	Touch Key Capture Data Low Byte

Register	Location	Reset value	Description
TKCDH	9Fh	00h	Touch Key Capture Data High Byte
TKSW	FEh	10h	Touch Key Switch
TKSTATUS0	BBh	00h	Touch Key Status 0
TKSTATUS1	F4h	00h	Touch Key Status 1
TKPSSR	FFEDh	07h	Touch Key Sampling Rate
GKWKCNT	FFE0h	02h	Group Key Wake-up Counter
SKWKCNT	FFE1h	02h	Touch Key Trigger Counter
TKRVC	FFE2h	6Bh	Touch Key Reference Voltage Control
TKFC0	FFE3h	20h	Touch Key Frequency Control 0
TKFC1	FFE4h	00h	Touch Key Frequency Control 1
TKFC2	FFE5h	3Fh	Touch Key Frequency Control 2
TKFC3	FFE6h	3Fh	Touch Key Frequency Control 3
TKFC4	DCh	3Fh	Touch Key Frequency Control 4
TKDC	FFE9h	00h	Touch Key Debounce Control
GP0S0	FFC1h	00h	Group 0 Key Selection 0
GP0S1	FFC0h	00h	Group 0 Key Selection 1
GP1S0	FFC3h	00h	Group 1 Key Selection 0
GP1S1	FFC2h	00h	Group 1 Key Selection 1
GP2S0	FFC5h	00h	Group 2 Key Selection 0
GP2S1	FFC4h	00h	Group 2 Key Selection 1
GP3S0	FFC7h	00h	Group 3 Key Selection 0
GP3S1	FFC6h	00h	Group 3 Key Selection 1
GP4S0	FFC9h	00h	Group 4 Key Selection 0
GP4S1	FFC8h	00h	Group 4 Key Selection 1
GP5S0	FFCBh	00h	Group 5 Key Selection 0
GP5S1	FFCAh	00h	Group 5 Key Selection 1
GP6S0	FFDDh	00h	Group 6 Key Selection 0
GP6S1	FFCCh	00h	Group 6 Key Selection 1
GP7S0	FFCFh	00h	Group 7 Key Selection 0
GP7S1	FFCEh	00h	Group 7 Key Selection 1
<b>LVI/LVR/SOFTRESET</b>			
RSTS	A1h	00h	Reset status register
LVC	E6h	60h	Low voltage control register
SWRES	E7h	00h	Software Reset register
TAKEY	F7h	00h	Time Access Key register
<b>IIC0</b>			
IIC0S	F8h	00h	IIC0 status register
IIC0CTL	F9h	03h	IIC0 control register
IIC0A1	FAh	A0h	IIC0 Address 1 register
IIC0A2	FBh	60h	IIC0 Address 2 register
IIC0RWD	FCh	00h	IIC0 Read / Write Data buffer

Register	Location	Reset value	Description
IIC0EBT	FDh	00h	IIC0 Enable Bus Transaction register
<b>IIC1</b>			
IIC1S	E8h	00h	IIC1 status register
IIC1CTL	E9h	04h	IIC1 control register
IIC1A1	EAh	A0h	IIC1 Address 1 register
IIC1A2	F1h	60h	IIC1 Address 2 register
IIC1RWD	ECh	00h	IIC1 Read / Write Data buffer
IIC1EBT	EDh	00h	IIC1 Enable Bus Transaction register
<b>ADC</b>			
ADCC1	ABh	00h	SADC Control 1 Register
ADCC2	ACh	0Fh	SADC Control 2 Register
ADCDH	ADh	00h	SADC data high byte
ADCDL	A Eh	00h	SADC data low byte
ADCCS	AFh	00h	SADC clock select
ADCSH	EFh	00h	SADC Sample and Hold Time
<b>LED</b>			
COMEN	FF1Ch	00h	LED COM Enable Register.
SEGEN0	FF1Dh	00h	LED SEG Enable 0 Register.
SEGEN1	FF1Eh	00h	LED SEG Enable 1 Register.
LEDCON	FF1Fh	00h	LED Control Register.
LEDCLK	FF20h	00h	LED Clock Register.
LEDCS0	FF21h	33h	LED current limiting Selection 0
LEDCS1	FF22h	33h	LED current limiting Selection 1

## 功能描述

### 1. 总特征

AS128A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

可通过编程器或将程序加载到16KB的嵌入式可编程存储器中。

#### 1.2 IO 口

AS128A有4个I/O 口: Port 0~Port 3.四种型态: 准双向口(标准 8051输出口), 推挽, 开漏, 与仅为输入。在章节5中有详细说明。

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 AS128A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。AS128A 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	
-	ITS[2:0]			-	-	CLKOUT[1:0]		
							Reset	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, AS128A 能在P3.4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

AS128A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

AS128A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	ZPLV RF	LVRL PINTF	WDT WF	TKWF	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	
ZPLV RF	LVRLP INTF	WDT WF	TKWF	WDTR F	SWRF	LVRF	PORF	00H	

ZPLVRF: 零功率低压复位旗标

当通过 ZPLOVR 信号复位 MCU 时,硬件会将 ZPLVR 旗标设置为 1.该旗标可以由软件清

除. (VIL: 0.7V)

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.

WDTRF: 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16MHz OSC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

<b>Clock source</b>
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

<b>Temperature</b>	<b>Max Variance</b>
25°C	±2%

## 2. 指令设置

所有AS128A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AS128A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

AS128A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16KB的嵌入式可编程存储器中。

#### 3.1 程序存储器

AS128A有16KB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。

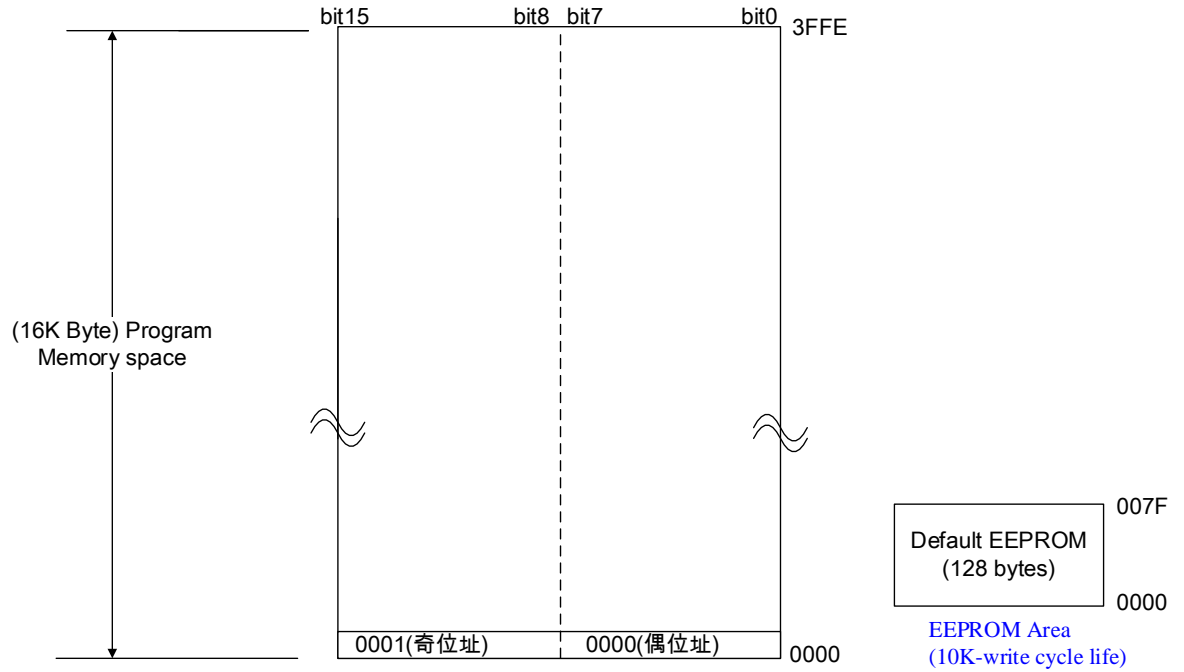


图 3-1: AS128A ROM



### 3.2 数据存储

AS128A具有256 Bytes + 1024 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。

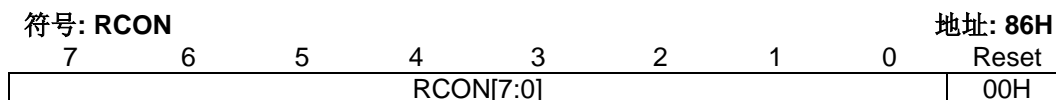
### 3.4 数据存储-高 128 字节(80h 到 FFh)

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 1024 字节(\$00 到 \$ 3FF)

从00h到17Fh是片内扩展的SRAM区域,共1024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)

指令MOVX @Ri, i = 0, 1的地址空间由特殊功能寄存器86h RCON (内部RAM控制寄存器) 的RCON [3: 0]确定。默认值. RCON [3: 0]的设置为00h (第0页)。 一页数据RAM为256字节。



RCON[7:0]: 外部 RAM 页选择

RCON[7:0]	Page Selection	Location
00H	0	00 – FFH
01H	1	100 –1FFH
02H	2	200 –2FFH
03H	3	300 –3FFH

## 4. CPU结构

AS128A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

AS128A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
<b>8051 Core</b>											
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		RS[1:0]		OV		P	00H
SP	Stack Pointer	81h	SP[7:0]								07H
DPL	Data pointer low	82h	DPL[7:0]								00H
DPH	Data pointer high	83h	DPH[7:0]								00H
AUX	Auxiliary register	91h	BRGS	TK11S	SICS[1:0]		TK10S	SEG1 S	SEG0 S	-	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: <b>ACC</b>										地址: <b>E0h</b>
	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]: A (或 ACC) 寄存器是标准的 8052 累加器。

### 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>B</b>										地址: <b>F0h</b>
	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]: B 寄存器是用作第二个累加器的标准 8052 寄存器。

### 4.3 程序状态字

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

CY: 进位标志位

AC: 为 BCD 辅助进位标志位

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

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

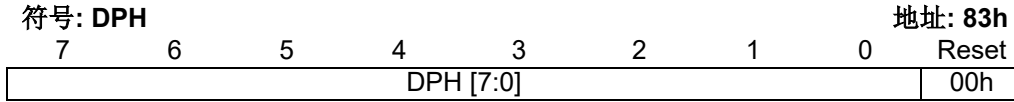
SP[7:0]: 堆栈指针储存了暂时寄存器的地址,该地址是堆栈指针的起始位置.换言之,它总是指向堆栈指针的顶端。

### 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).

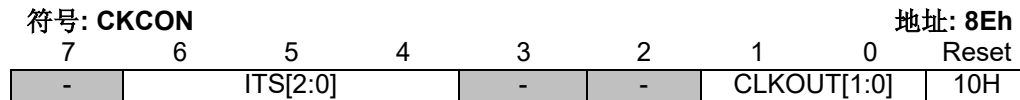
符号: DPL							地址: 82h	
7	6	5	4	3	2	1	0	Reset
DPL [7:0]								00h

DPL[7:0]: Data pointer Low



DPH [7:0]: Data pointer High

#### 4.6 时钟控制寄存器



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

#### 4.7 接口控制寄存器



CDPR: 程序码已加密指示位(仅读)

ISPE: EEPROM 功能使能位

ISPE = 1, 允许使用 EEPROM 功能

ISPE = 0, 禁止使用 EEPROM 功能

## 5. GPIO管脚型态

AS128A有4个I/O口: Port 0~ Port 3. 他们是: 准双向口 (标准的8051端口输出),推挽电路,开漏, 与只输入. 两种寄存器的配置为每个端口的每个位选择输出方式.AS128A的所有端口可以通过软件配置四种型号的一种.如下表显示:

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
I/O port function register											
P0M0	Port 0 output mode 0	D2h	P0M0[7:0]							~OP17	
P0M1	Port 0 output mode 1	D3h	P0M1[7:0]							~OP18	
P1M0	Port 1 output mode 0	D4h	P1M0[7:0]							~OP19	
P1M1	Port 1 output mode 1	D5h	P1M1[7:0]							~OP1A	
P2M0	Port 2 Output Mode 0	D6H	-	P2M0[6:5]	-	P2M0[3]	-	-	-	-	~OP1B
P2M1	Port 2 Output Mode 1	D7H	-	P2M1[6:5]	-	P2M0[3]	-	-	-	-	~OP1C
P3M0	Port 3 Output Mode 0	DAH	-	-	P3M0[5:0]					~OP1D	
P3M1	Port 3 Output Mode 1	DBH	-	-	P3M1[5:0]					~OP1E	

PxM1.y	PxM0.y	Port output mode
0	0	准双向口 (标准的8051端口输出)
0	1	推挽电路
1	0	只输入 (high-impedance)
1	1	开漏

一般的应用,每个管脚都可独立的置高或置低.如下表显示:

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
Ports											
Port 3	Port 3	B0h	-	-	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	~OP22
Port 2	Port 2	A0h	-	P2.6	P2.5	-	P2.3	-	-	-	~OP21
Port 1	Port 1	90h	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	~OP20
Port 0	Port 0	80h	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	~OP1F

符号: **P0** Address: **80h**

7	6	5	4	3	2	1	0	Reset
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	~OP1F

P0.7~ 0: Port0 [7] ~ Port0 [0]

符号: **P1** Address: **90h**

7	6	5	4	3	2	1	0	Reset
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	~OP20

P1.7~ 0: Port1 [7] ~ Port1 [0]

符号: **P2** Address: **A0h**

7	6	5	4	3	2	1	0	Reset
-	P2.6	P2.5	-	P2.3	-	-	-	~OP21

P2.6~ 3: Port2 [6] ~ Port2 [3]

符号: **P3** Address: **B0h**

7	6	5	4	3	2	1	0	Reset
-	-	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	~OP22

P3.5~ 0: Port3 [5] ~ Port3 [0]

## 6. 定时器0、定时器1、定时器3

AS128A有三个16bit的定时/计数寄存器: 定时器0, 定时器1, 定时器3. 所有这都可以被设置为定时或计数操作.

在计时的模式中, 定时器0、定时器1、定时器3寄存器的递增频率, 可由寄存器PFCON中选择为与振荡器频率相同或为振荡器频率的1/12或为振荡器频率的1/96.

在计数的模式中, 当检测到相应的输入脚T0/T1有下降沿产生时, 寄存器便得到递增, 由于它需要2个机器周期来识别由1到0的跳变, 其最大输入的计数频率为振荡器频率的1/2, 从而使得这里没有限制的占空比, 以确定适当的识别为0或1的状态, 因此, 一个输入信号至少要稳定在1个机器周期.

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 0 and 1											
TL0	Timer 0, low byte	8Ah				TL0[7:0]					00H
TH0	Timer 0, high byte	8Ch				TH0[7:0]					00H
TL1	Timer 1, lowbyte	8Bh				TL1[7:0]					00H
TH1	Timer 1, high byte	8Dh				TH1[7:0]					00H
TL3	Timer 3 Low Byte	94H				TL3[7:0]					00H
TH3	Timer 3 High Byte	95H				TH3[7:0]					00H
TMOD	Timer Mode Control	89h	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
T3MOD	Timer 3 Mode Control	92H	-	-	-	-	GATE	C/T	M1	M0	00H
TCON0	Timer/Counter Control 0	88h	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
TCON1	Timer/Counter Control 1	D8H	IT0S	IT1S	TF3	TR3	IT2S	-	IE2	IT2	00H
ENHIT	ENHance Interrupt Type Register	E5h	-	-	ENHIT2[1:0]		ENHIT1[1:0]		ENHIT0[1:0]		00H
INTDEG	External Interrupt Deglitch register	EEh	-	-	INT2DEG[1:0]		INT1DEG[1:0]		INT0DEG[1:0]		00H
PFCON	Peripheral Frequency control register	D9h	-	-	T3PS[1:0]		T1PS[1:0]		T0PS[1:0]		00H

### 6.1 定时器/计数器模式控制寄存器

符号: TMOD							地址: 89h		
7	6	5	4	3	2	1	0	Reset	
GATE	C/T	M1	M0	GATE	C/T	M1	M0	00h	
Timer 1				Timer 0					

**GATE:** 该位被置位时为门控时, 仅当'INT0 或者 INT1'脚为高时, 且'TRx'控制位被置位 (参考 TCON 寄存器) 时使能, 计数器在每个 T0 或 T1 输入脚处于下减沿触发时得到加强.

**C/T:** 门控定时器或计数器选择器. 该位被置位时用作计数器功能, 该位被清零时用作定时器功能.

**M[1:0]:** 定时/计数器 0 或定时/计数器 1 的选择模式



M1	M0	Mode	Function
0	0	Mode0	13bit计数器/定时器, 包含TL0/TL1寄存器的低5位及TH0/TL1寄存器的全部8位, 其TL0/TL1寄存器的高3位可设置为0.
0	1	Mode1	16 位计数器/定时器.
1	0	Mode2	8 位自动重载的计数器/定时器,自动重载的值保留在TH0 和TH1.同时TL0 或者TL1 在每个机器周期内都会递增. 当溢出时, 将THx 存放的值装入TLx.
1	1	Mode3	如定时器1 的M1 和M2 位被设置为1, 定时器2 停止计数。如定时器0 的M1 和M0 位被设置为1, 定时器0 作为两个独立的8 位定时器/计数器.

符号: T3MOD							地址: 92h	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	GATE	C/T	M1	M0	00h
Timer 3								

**GATE:** 该位被置位时为门控时, 仅当‘INT2 脚为高时, 且‘TR3’控制位被置位 (参考 TCON1 寄存器) 时使能, 计数器在每个 T3 输入脚处于下减沿触发时得到加强。

**C/T:** 门控定时器或计数器选择器。该位被置位时用作计数器功能, 该位被清零时用作定时器功能。

**M[1:0]:** 定时/计数器 3 的选择模式

M1	M0	Mode	Function
0	0	Mode0	13bit计数器/定时器, 包含TL3寄存器的低5位及TH3寄存器的全部8位, 其TL3寄存器的高3位可设置为0.
0	1	Mode1	16 位计数器/定时器.
1	0	Mode2	8 位自动重载的计数器/定时器,自动重载的值保留在TH0 和TH1.同时TL0 或者TL1 在每个机器周期内都会递增. 当溢出时, 将TH3存放的值装入TL3.
1	1	-	保留

## 6.2 定时器/计数器控制寄存器(TCON0)

符号: TCON0							地址: 88h	
7	6	5	4	3	2	1	0	Reset
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00h

**TF1:** 定时器 1 溢出标志。定时器/计数器溢出时由硬件置位。中断执行时硬件清零, 或由软件清零

- TR1: 定时器 1 运行控制位。 如置位/清零, 关闭定时器/计数器 1。
- TF0: 定时器 0 溢出标志。定时器/计数器溢出时由硬件置位。中断执行时硬件自动清零, 或由软件清零
- TR0: 定时器 0 运行控制位。如置位/清零, 关闭定时器/计数器 0
- IE1: 中断 1 边沿标志。当检测到外部中断 1 边沿/低电平时由硬件置位该标志。中断处理时由硬件清零, 或通过软件清零
- IT1: 中断 1 类型控制位。  
IT1 = 0: INT1 选择电平触发。(高电平或低电平取决于 ENHIT1)  
IT1 = 1: INT1 选择边沿触发。(下降沿或上升沿, 或两个边沿均取决于 ENHIT1)。
- IE0: 中断 0 边沿标志。当检测到外部中断 0 边沿/低电平时由硬件置位该标志。中断处理时由硬件清零, 或通过软件清零
- IT0: 中断 0 类型控制位。  
IT0 = 0: INT0 选择电平触发。(高电平或低电平取决于 ENHIT0)  
IT0 = 1: INT0 选择边沿触发。(下降沿或上升沿, 或两个边沿均取决于 ENHIT0)。

符号: TCON1							地址: 88h	
7	6	5	4	3	2	1	0	Reset
IT0S	IT1S	TF3	TR3	IT2S	-	IE2	IT2	00h

- IT0S: 外部中断 0 引脚交换。  
0: P1.3  
1: P3.4
- IT1S: 外部中断 1 引脚交换。  
0: P2.6  
1: P1.0
- TF3: 定时器 3 溢出标志。定时器/计数器溢出时由硬件置位。中断执行时硬件清零, 或由软件清零。
- TR3: 定时器 3 运行控制位。 如置位/清零, 关闭定时器/计数器 3。
- IT2S: 外部中断 2 引脚交换。  
0: P3.3  
1: P2.3
- IE2: 中断 2 边沿标志。当检测到外部中断 2 边沿/电平时由硬件置位该标志。中断处理时由硬件清零。
- IT2: 中断 2 类型控制位。  
IT2=0: INT2 选择电平方式触发 (高或低电平依据 ENHIT2 设定)  
IT2=1: INT2 选择边缘方式触发 (下降或上升缘触发或两者兼具, 依据 ENHIT2 设定)

### 6.3 增强中断型态寄存器(ENHIT)

符号: ENHIT				地址: E5h			
7	6	5	4	3	2	1	0
-	-	ENHIT2[1:0]		ENHIT1[1:0]		ENHIT0[1:0]	
Reset 00h							

	ENHIT0[1:0]=00	ENHIT0[1:0]=01	ENHIT0[1:0]=10	ENHIT0[1:0]=11
IT0=0	INT0 Low level trigger	INT0 High level trigger	--	--
IT0=1	INT0 Falling edge	INT0 Rising trigger	INT0 Both falling and rising	--

	ENHIT1[1:0]=00	ENHIT1[1:0]=01	ENHIT1[1:0]=10	ENHIT1[1:0]=11
IT1=0	INT1 Low level trigger	INT1 High level trigger	--	--
IT1=1	INT1 Falling edge	INT1 Rising trigger	INT1 Both falling and rising	--

	ENHIT2[1:0]=00	ENHIT2[1:0]=01	ENHIT2[1:0]=10	ENHIT2[1:0]=11
IT2=0	INT2 Low level trigger	INT2 High level trigger	--	--
IT2=1	INT2 Falling edge	INT2 Rising trigger	INT2 Both falling and rising	--

### 6.4 外部中断 deglitch 寄存器(INTDEG)

符号: INTDEG				地址: EEh			
7	6	5	4	3	2	1	0
-	-	INT2DEG[1:0]		INT1DEG[1:0]		INT0DEG[1:0]	
Reset 00H							

INT2DEG[1:0] 选择 INT2 deglitch 时间.

00: no deglitch.

01: 5us

10: 10us

11: 15us

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

### 6.5 定时器输入频率控制寄存器

符号: PFCON							地址: D9h	
7	6	5	4	3	2	1	0	Reset
-	-	T3PS[1:0]		T1PS[1:0]		T0PS[1:0]		00H

T3PS[1:0]: 定时器 3 分频器选择位

T3PS[1:0]	分频器
00	Fosc/12
01	Fosc
10	Fosc/96
11	reserved

T1PS[1:0]: 定时器 1 分频器选择位

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

T0PS[1:0]: 定时器 0 分频器选择位

T0PS[1:0]	分频器
00	Fosc/12
01	Fosc
10	Fosc/96
11	reserved

### 6.6 模式 0 (13 位定时/计数)

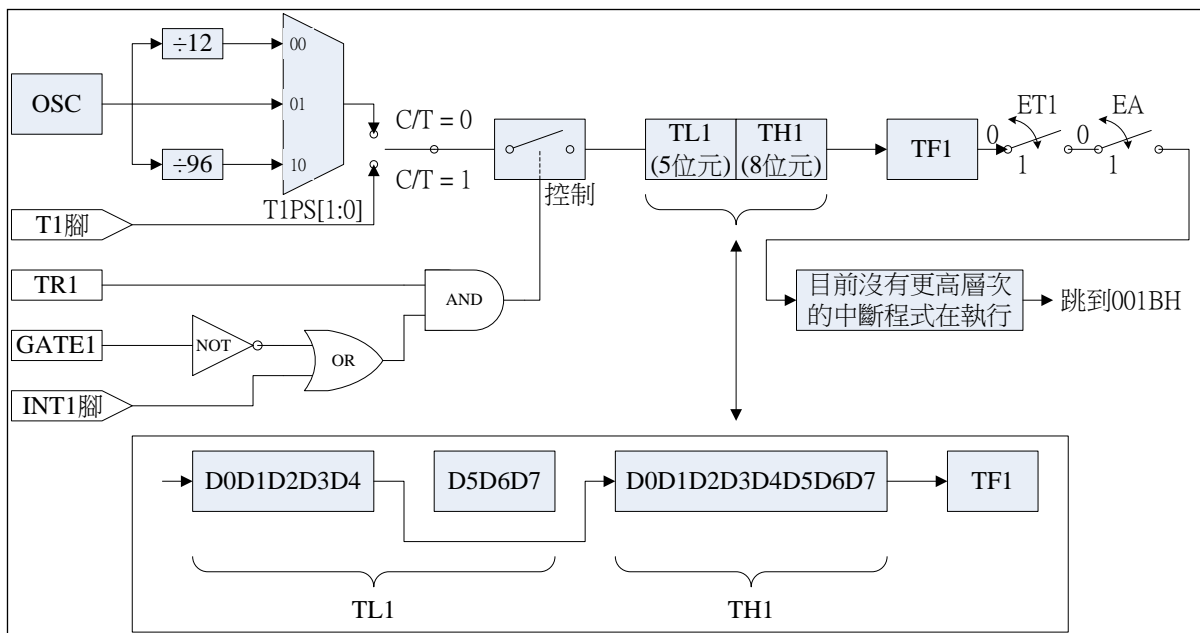


图 6-1: 模式 0-13 位定时器/计数器操作

### 6.7 模式 1 (16 位定时/计数)

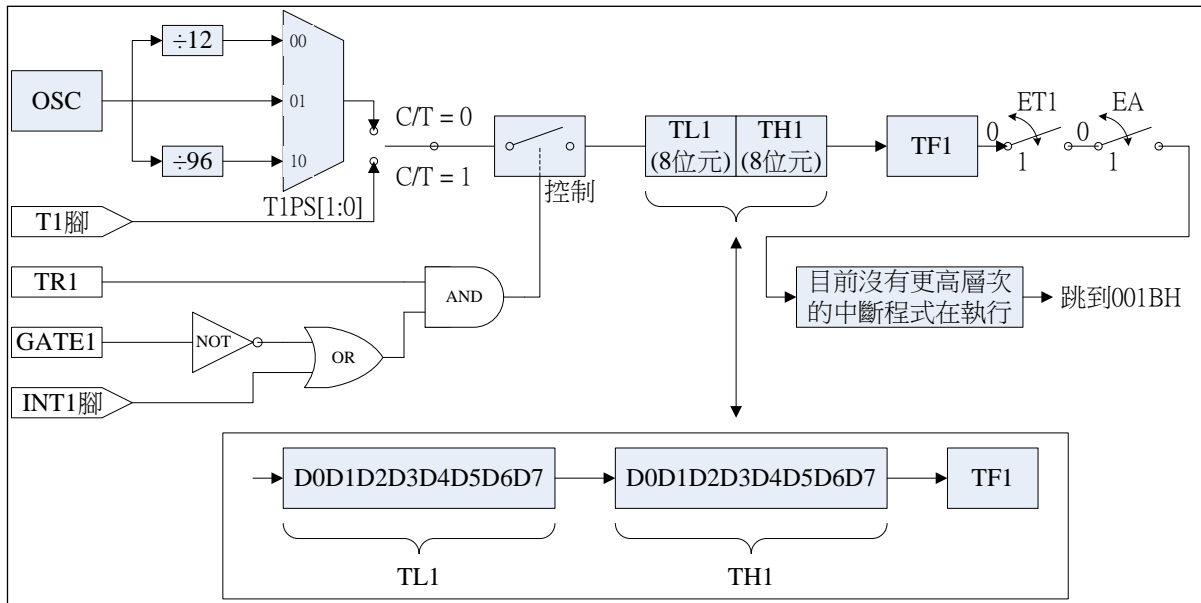


图 6-2: 模式 1-16 位定时器/计数器操作

### 6.8 模式 2 (8 位自动重载定时/计数)

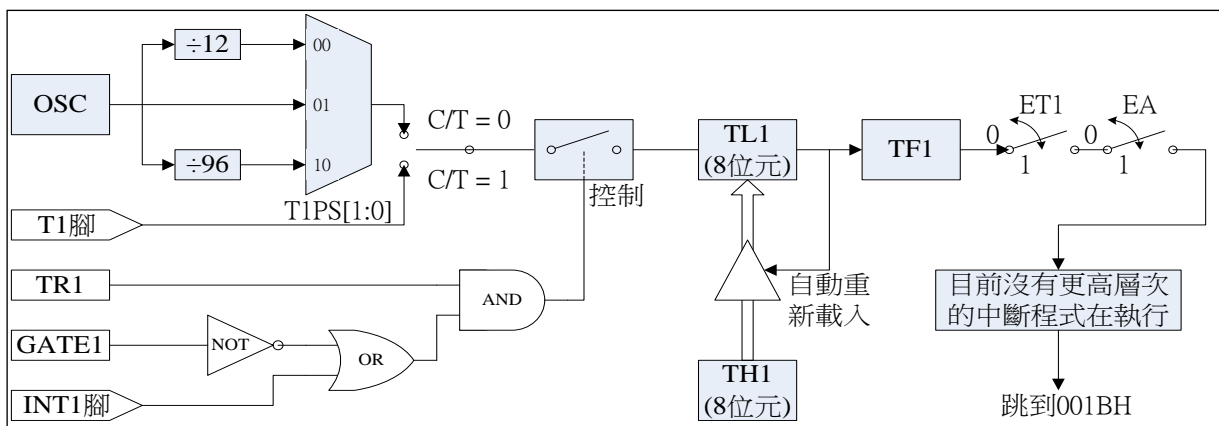


图 6-3: 模式 2-8 位自动重载定时/计数

6.9 模式 3 (两个独立 8 位定时/计数(仅定时器 0))

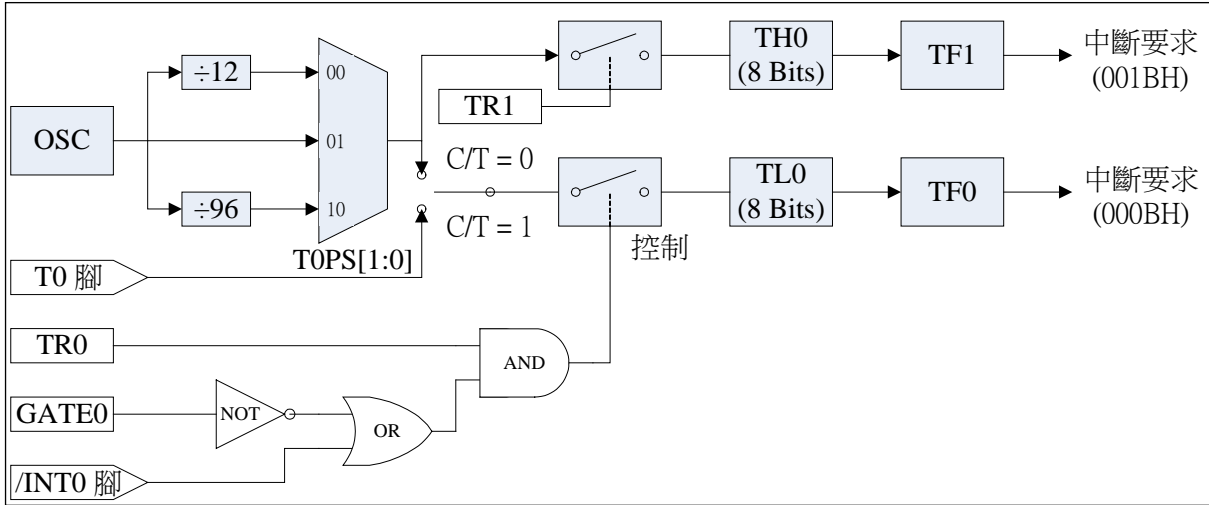
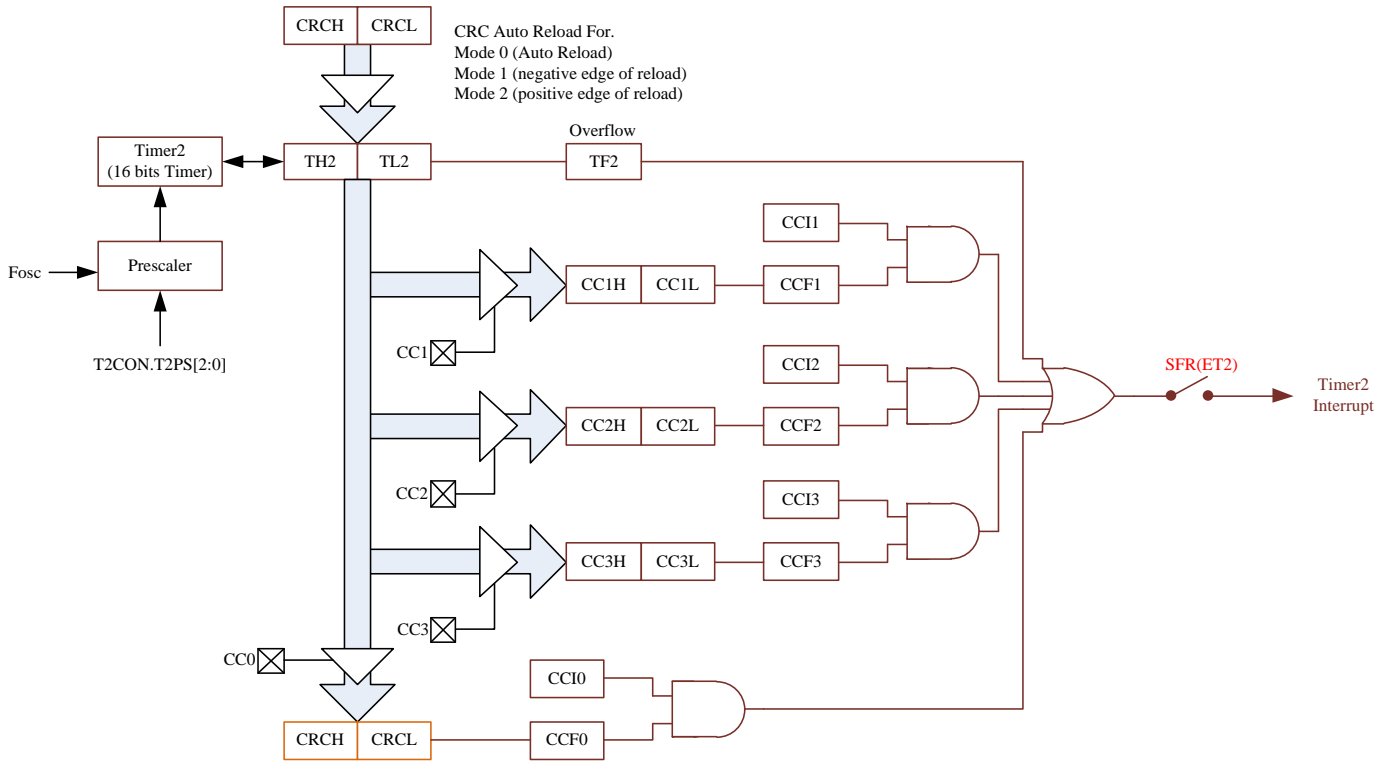


图 6-4: 模式 3 -两个独立 8 位定时/计数(仅定时器 0)

## 7. 定时器2 以及捕捉/比较单元(CCU)

定时器2不仅仅是一个16位的定时器，也是一个带有比较，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
<b>Timer 2 and Capture Compare Unit</b>											
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] = 00 – 重载关闭。
- T2R[1:0] = 01 – 模式 2: 依 T2EX 脚上升缘重载。
- 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	Reset
-	COCAM3[2:0]		-	COCAM2[2:0]				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(Timer mode)

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。

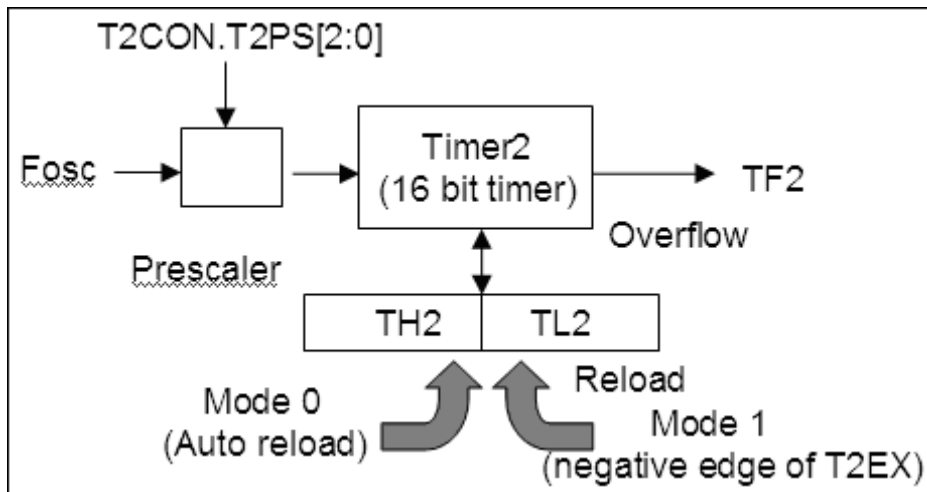


图 7-1: Timer mode and Reload mode function

### 7.1.2 外部信号计数模式(Event counter mode)

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。

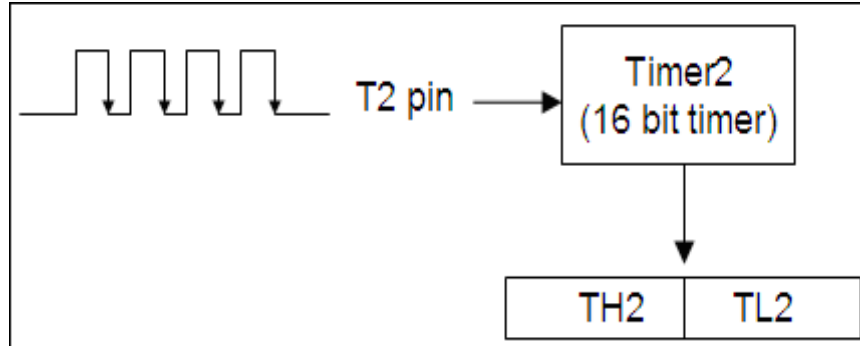


图 7-2: Event counter mode function

### 7.1.3 外部信号定时器模式(Gated timer mode)

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。

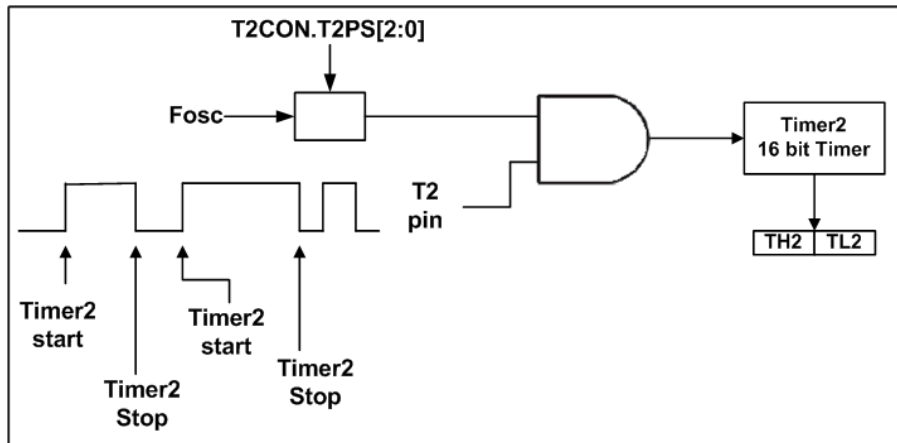


图 7-3: Gated timer mode function

### 7.1.4 定时器 2 的重载

重载（来自CRC寄存器的16位重载，此时CRC寄存器做为重载寄存器，CRC的内容值将会重载至TH2与TL2）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

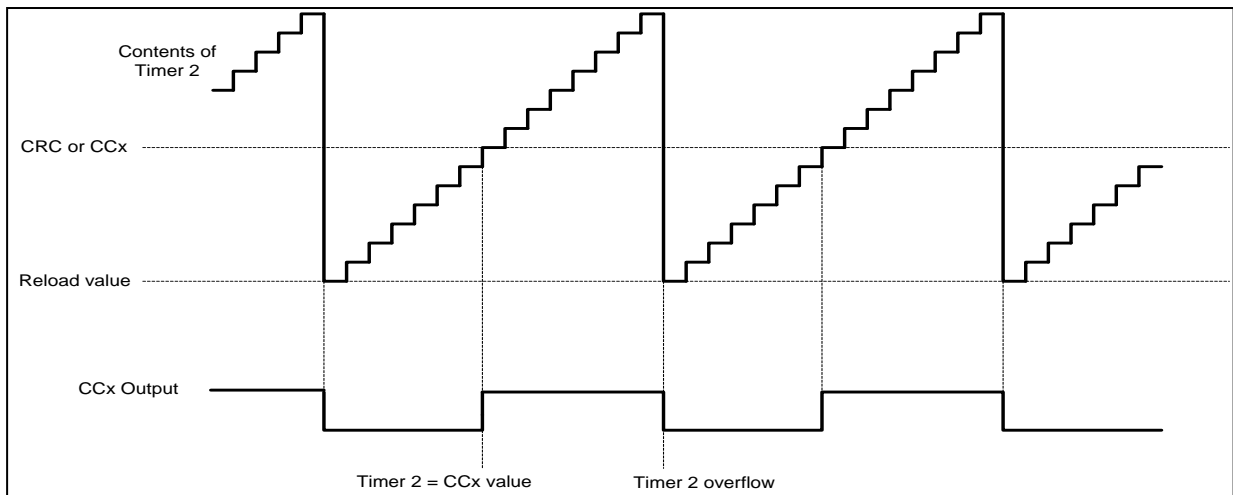


图 7-4: Compare mode 0 function

### 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

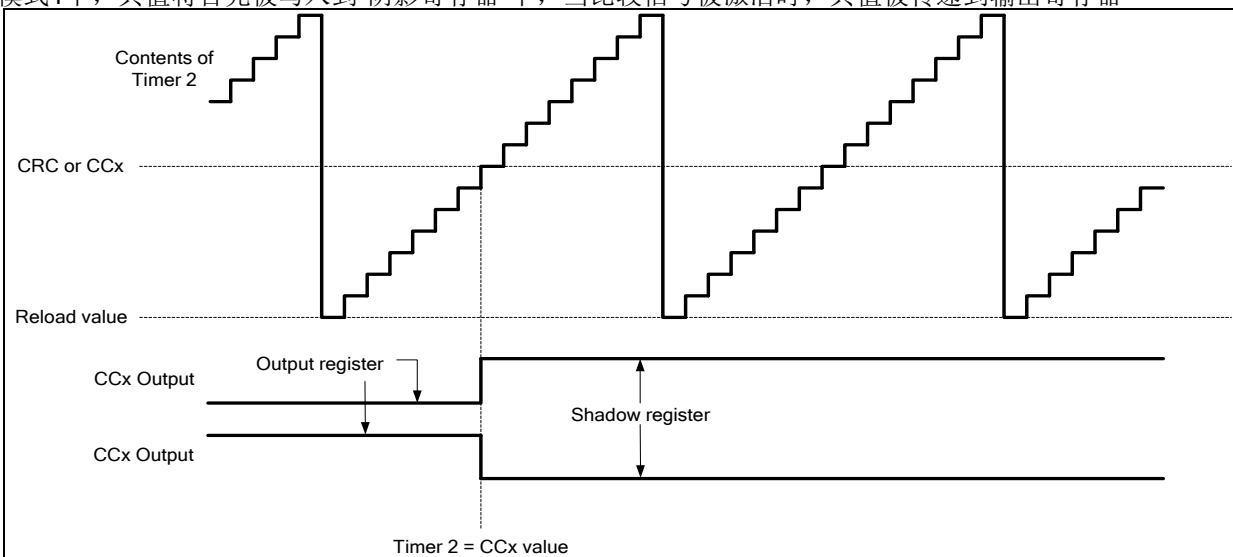


图 7-5: 比较模式 1 功能

### 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。

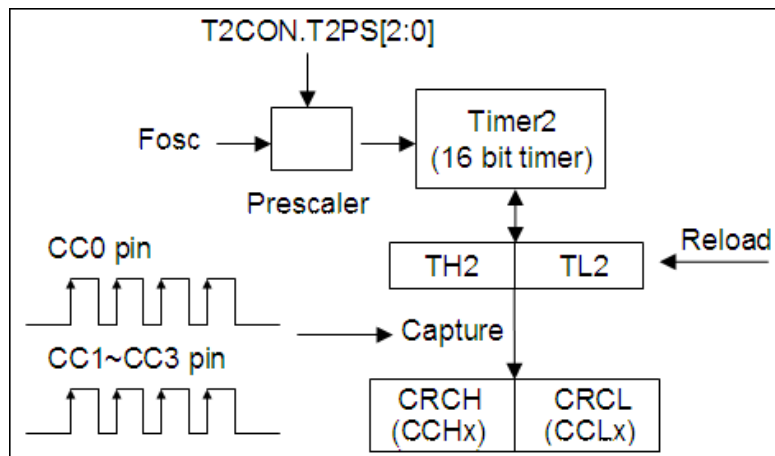


图 7-6: 捕捉模式 0 功能

#### 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。

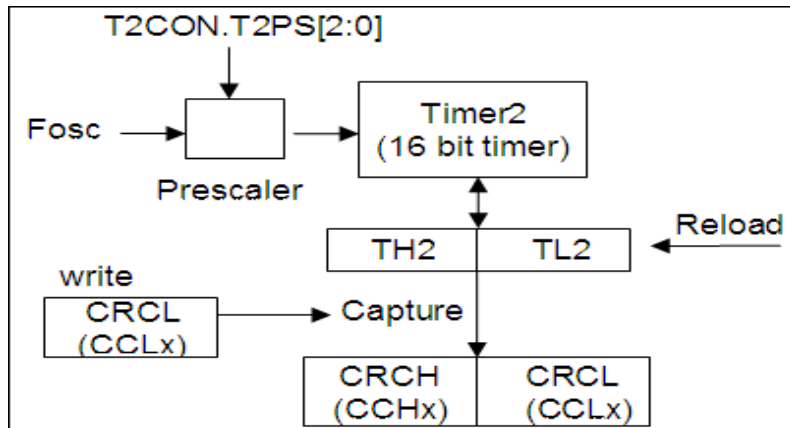


图 7-7: 捕捉模式 1 功能

## 8. 串行接口

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											
PCON	Power control	87H	SMOD	-	-	-	-	-	STOP	IDLE	00H
AUX	Auxiliary register	91h	BRGS	TK11 S	SICS[1:0]		TK10 S	SEG1 S	SEG0 S	-	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	-	-	T3PS[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	TK11S	SICS[1:0]		TK10S	SEG1S	SEG0S	-	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。

SICS[1:0]: 串行接口通道选择控制

SICS[1:0]	Channel	Note
00	Port 1	RXD_0,TXD_0
01	Port 2	RXD_1,TXD_1
10	保留	-
11	保留	-

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，置位或清零取决于它执行的功能，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。

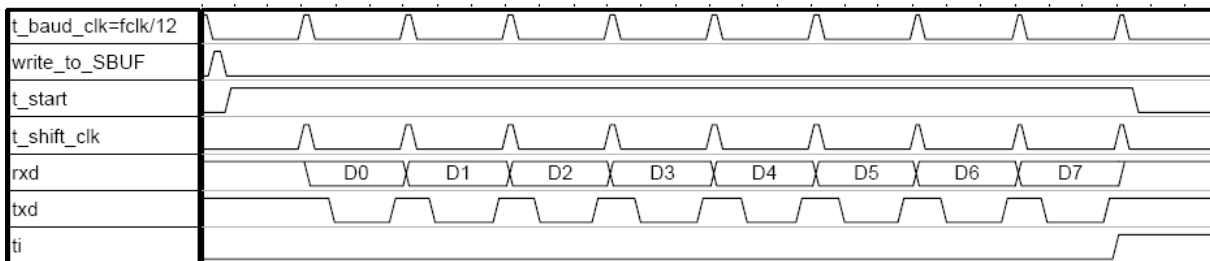


图 8-1: 发送模式 0

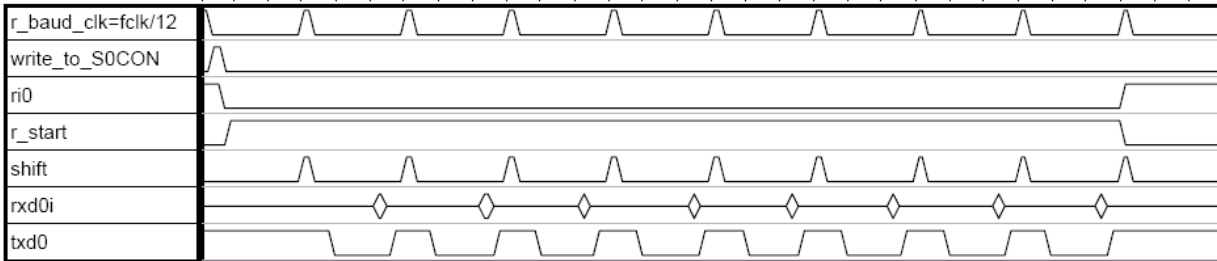


图 8-2: 接收模式 0

### 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。

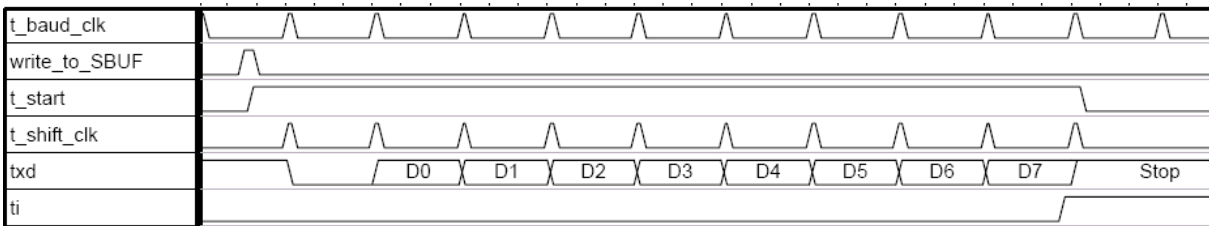


图 8-3: 发送模式 1

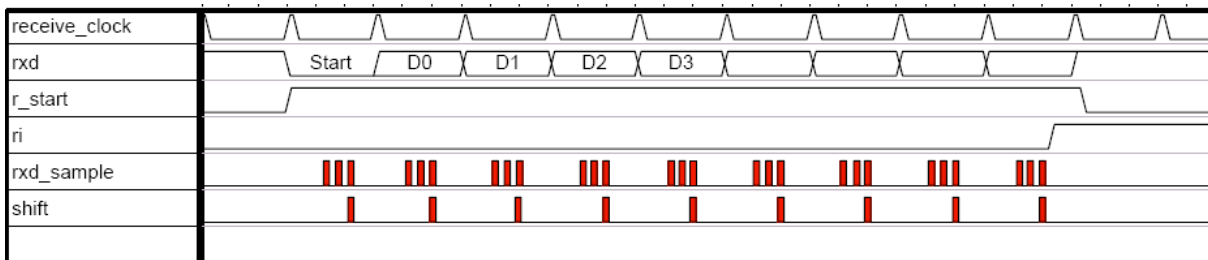


图 8-4: 接收模式 1

### 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。



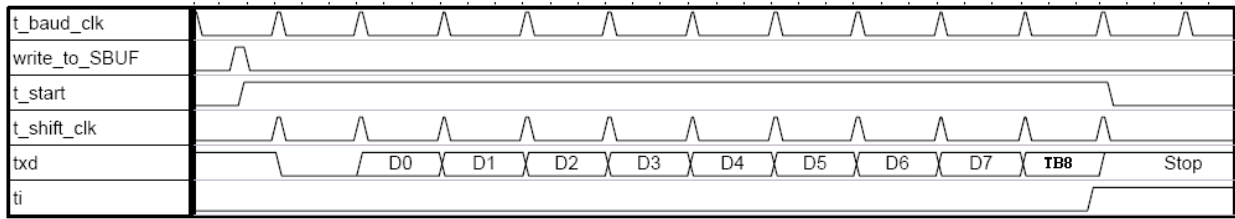


图 8-5: 传输模式 2 和模式 3

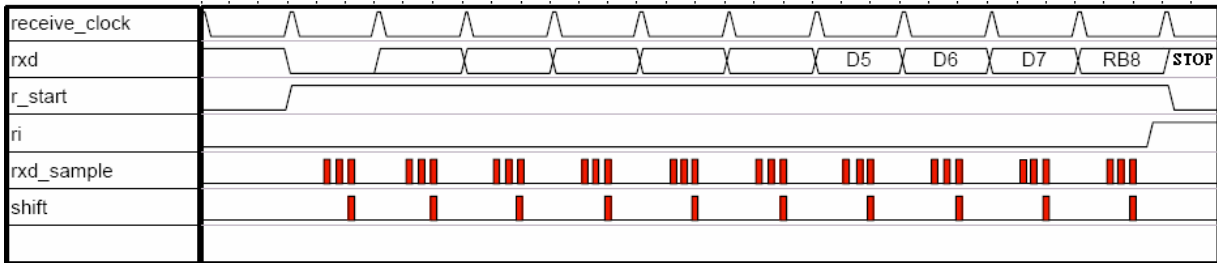


图 8-6: 接收模式 2 和 3 中

## 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
-	-	T3PS[1:0]		T1PS[1:0]		T0PS[1:0]	
							Reset
							00H

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)

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

## 9. 看门狗定时器

看门狗定时器区分为看门狗复位定时器与看门狗中断定时器，两种定时器共享同一个 23KHz 晶振源，但可以独立运作，因此可同时启用看门狗复位与看门狗中断。并会在计数器溢出时产生复位或中断。

$$WDTRCLK = \frac{23\text{KHz}}{2^{WDTRM}} \quad \text{Watchdog reset time} = \frac{256}{WDTRCLK}$$

$$WDTICKL = \frac{23\text{KHz}}{2^{WDTIM}} \quad \text{Watchdog Interrupt time} = \frac{256}{WDTICKL}$$

表 9- 1: WDT time-out period

WDTRM [3:0]	Divider (23 KHz RC oscillator in)	Reset Time period @ 23KHz	WDTIM [3:0]	Divider (23 KHz RC oscillator in)	Interrupt Time period @ 23KHz
0000	1	11.1ms	0000	1	11.1ms
0001	2	22.2ms	0001	2	22.2ms
0010	4	44.5ms	0010	4	44.5ms
0011	8	89.0ms	0011	8	89.0ms
0100	16	178.0ms (default)	0100	16	178.0ms
0101	32	356.1ms	0101	32	356.1ms
0110	64	712.3ms	0110	64	712.3ms
0111	128	1.4246s	0111	128	1.4246s
1000	256	2.8493s	1000	256	2.8493s
1001	512	5.6987s	1001	512	5.6987s
1010	1024	11.397s	1010	1024	11.397s
1011	2048	22.795s	1011	2048	22.795s
1100	4096	45.590s	1100	4096	45.590s
1101	8192	91.180s	1101	8192	91.180s
1110	16384	182.36s	1110	16384	182.36s
1111	32768	364.72s	1111	32768	364.72s

注: RC 振荡器(23 KHz), 大约有 ± 20 % 误差

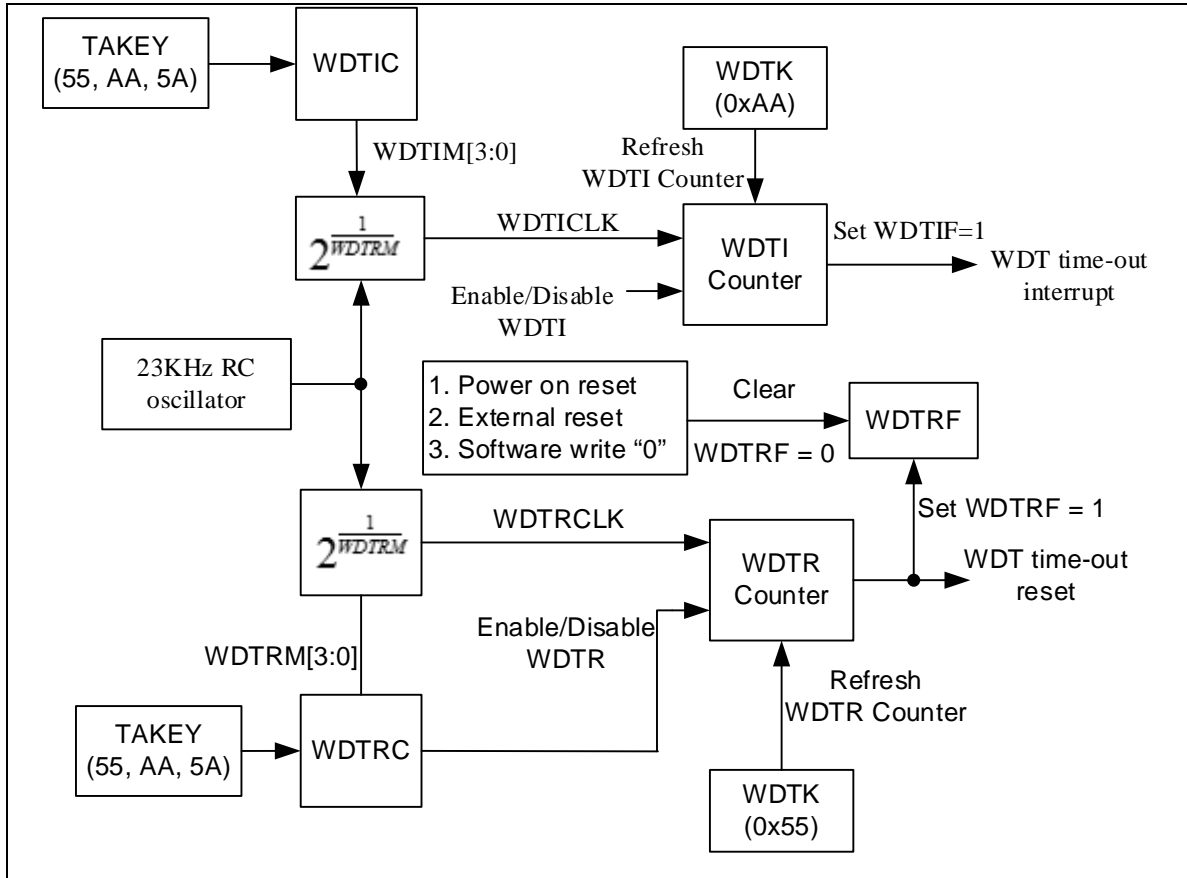
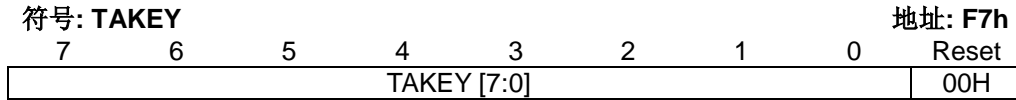


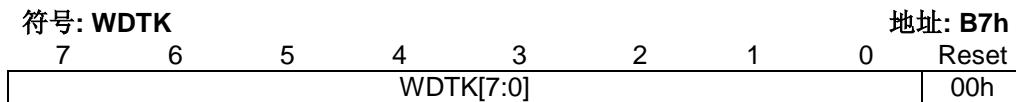
图 9- 1: Watchdog 定时器框图

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
WDTRC	Watchdog timer reset control register	B6h	-	-	WDTRE	-	WDTRM[3:0]				04H
WDTIC	Watchdog timer interrupt control register	FFh	-	-	WDTIE	WDTIOF	WDTIM[3:0]				00H
WDTK	Watchdog timer refresh key	B7h	WDTK[7:0]								00H
RSTS	Reset status register	A1h	ZPLVRF	LVRPINTF	WDTWTF	TKWF	WDTRF	SWRF	LVRF	PORF	00H



看门狗控制寄存器(WDTRC、WDTIC)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
```



WDTK: 看门狗定时器计数器清零寄存器.

于此寄存器写入 0x55 或 0xAA，看门狗定时器计数器将清零重新计数.

### 9.1 看门狗复位定时器

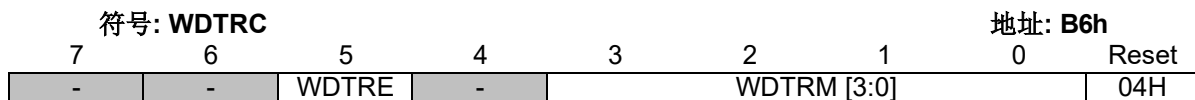
WDT复位在噪音、电源干扰，或断电等导致软件死循环或跑飞程序等情况下非常有用。WDT复位功能可以帮助拥护的软件从不正常的软件状态中恢复正常。WDT不同于计时器0、1、2、3。为了防止WDT复位，可以通过软件定时清除WDT计数器。当不可预料的复位发生时，用户应该检查RSTS寄存器的WDTRF位。在外部复位后，该看门狗定时器将关闭且所有寄存器都清零。

看门狗定时器是一个自由运行的片上RC振荡器 (约23 KHz)。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关闭。

对WDTRE位写入1，能使WDT功能使能。在WDTRE设为1后，8位的计数器用通过WDTRM[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。WDTRE位在MCU重启时将被自动清0，同样在硬件复位或WDT复位WDTRE也将被自动清0。如图9-1所示。

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复位计时器清0，这将会清除8位计数器内容并让计数器重启。看门狗复位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，WDTRF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。



WDTRE: 看门狗复位定时器使能位.

0: 禁能.

1: 使能.

WDTRM [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: <b>RSTS</b>							地址: <b>A1h</b>	
7	6	5	4	3	2	1	0	Reset
ZPLVRF	LVRLPINTF	WDTWF	TKWF	WDTRF	SWRF	LVRF	PORF	00h

WDTWF: 看门狗中断掉电唤醒旗标.

此旗标可由软件清零

WDTRF: 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

范例:

看门狗定时器使能并选择重置信号产生时间为 2.8493s

第一步,先确认刻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 WDTRC write attribute.

MOV WDTRC, #28h ; Set WDTRM [3:0] = 1000b. Set WDTRE =1 to enable WDT function.

...

...

...

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

## 9.2 看门狗中断定时器

看门狗中断定时器无论在正常的运行或睡眠状态, 都可以保持运行, 最常应用于睡眠模式下唤醒MCU。

对WDTIE位写入1, 能使WDT中断功能使能. 在WDTIE设为1后, 8位的计数器用通过WDTIM[3:0]设置好的分频进行计数. 它在溢出时将会产生中断讯号. 如图9-1所示。

当WDTK寄存器用户可通过对看门狗定时器重置密钥(WDTK)写入AAh可实现WDT中断计时器清0, 这将会清除8位计数器内容并让计数器重启。

符号: <b>WDTIC</b>							地址: <b>FFh</b>	
7	6	5	4	3	2	1	0	Reset
-	-	WDTIE	WDTIOF	WDTIM [3:0]				00H

WDTIE: 看门狗中断定时器使能位.

0: 禁能.

1: 使能.

WDTIOF: 看门狗中断定时器溢出标志由硬件在看门狗中断定时器溢出时设置。该标志可以由软件清除

WDTRM [3:0]: 看门狗中断信号产生时间选择位.请参考表 9-1 所列看门狗中断信号产生时间.

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

EWDT: WDT 看门狗中断使能位

EWDT = 0 –禁能WDT 中断

EWDT = 1 –使能WDT 中断，使能后当看门狗中断发生时，程序会跳跃到看门狗中断向量地址

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

WDTIF: 看门狗中断旗标位.进入中断向量时，硬件自动清零.

范例:

看门狗定时器使能并选择周期中断信号产生时间为 178.0ms

第一步,先确认刻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 WDTIC write attribute.

MOV WDTIC, #24h ; Set WDTIM [3:0] = 0100b. Set WDTIE =1 to enable WDT interrupt function

## 10. 中断

AS128A提供15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、IEN1及IEN2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

	<b>Interrupt Request Flags</b>	<b>Interrupt Vector Address</b>	<b>Interrupt Number *(use Keil C Tool)</b>
1	IE0 – External interrupt 0	0003h	0
2	TF0 – Timer 0 interrupt	000Bh	1
3	IE1 – External interrupt 1	0013h	2
4	TF1 – Timer 1 interrupt	001Bh	3
5	RI/TI – Serial channel interrupt	0023h	4
6	TF2/EXF2 – Timer 2 interrupt	002Bh	5
7	IE2 - External interrupt 2	0033h	6
8	PWMIF – PWM interrupt	0043h	8
9	IICIF1 – IIC 1 interrupt	004Bh	9
10	ADCIF – A/D converter interrupt	0053h	10
11	LVIIIF – Low Voltage Interrupt	0063h	12
12	IICIF0 – IIC 0 interrupt	006Bh	13
13	TF3 - Timer 3 interrupt	0073h	14
14	WDTIF – Watchdog interrupt	008Bh	17
15	TKIF – Touch interrupt	009Bh	19



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	ET3	ET2	ES	ET1	EX1	ET0	EX0	00H
IEN1	Interrupt Enable 1 register	B8H	EXEN2	EX2	IEIIC0	IELVI	-	IEADC	IEIIC1	IEPWM	00H
IEN2	Interrupt Enable 2 register	9AH	-	-	-	-	IETK	-	EWD T	-	00H
IRCON	Interrupt request register	C0H	EXF2	TF2	IICIF0	LVIIIF	-	ADCF	IICIF1	PWMIF	00H
IRCON2	Interrupt request register 2	97H	-	-	-	-	TKIF	-	WDTIF	-	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

符号: **IEN0** 地址: **A8h**

7	6	5	4	3	2	1	0	Reset
EA	ET3	ET2	ES	ET1	EX1	ET0	EX0	00h

- EA: EA=0 –禁能所有中断。  
EA=1 –使能所有中断。
- ET3: ET3=0 –禁能定时器 3 中断。  
ET3=1 –使能定时器 3 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	EX2	IEIIC0	IELVI	-	IEADC	IEIIC1	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
- EX2:** 外部中断 2 使能位  
 EX2=0 –禁能外部中断 2。  
 EX2=1 –使能外部中断 2。
- IEIIC0:** IIC0 中断使能位  
 IEIIC0 = 0 –禁能 IIC0 中断  
 IEIIC0 = 1 –使能 IIC0 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
- IEADC:** A/D 转换中断使能位  
 IEADC = 0 –禁能 ADC 中断  
 IEADC = 1 –使能 ADC 中断
- IEIIC1:** IIC1 中断使能位  
 IEIIC1 = 0 –禁能 IIC1 中断  
 IEIIC1 = 1 –使能 IIC1 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
-	-	-	-	IETK	-	EWDT	-	00H

- IETK:** 触摸按键中断使能位  
 IETK = 0 –禁能触摸按键中断  
 IETK = 1 –使能触摸按键中断
- 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	IICIF0	LVIIF	-	ADCIF	IICIF1	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
- IICIF0:** IIC0 中断旗标位. 进入中断向量时, 硬件自动清零.

LVIIF: 低压侦测中断旗标位.进入中断向量时, 硬件自动清零.  
ADCIF: A/D 转换中断旗标位. 进入中断向量时, 硬件自动清零.  
IICIF1: IIC1 中断旗标位. 进入中断向量时, 硬件自动清零.  
PWMIF: PWM 中断旗标位. 进入中断向量时, 硬件自动清零.

符号: <b>IRCON2</b>								地址: <b>97h</b>
7	6	5	4	3	2	1	0	Reset
-	-	-	-	TKIF	-	WDTIF	-	00H

TKIF: 触摸按键中断旗标位.必须由软件清零.  
WDTIF: 看门狗中断旗标位.进入中断向量时, 硬件自动清零.

### 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	IIC1 中断
外部中断 1	定时器 3 中断	ADC 中断
定时器 1 中断	外部中断 2	-
串行口中断	-	LVI 低压侦测中断
触摸按键中断	定时器 2 中断	IIC0 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: <b>IP0</b>								地址: <b>A9h</b>
7	6	5	4	3	2	1	0	Reset
-	-	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	00h

符号: <b>IP1</b>								地址: <b>B9h</b>
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	优先权层级
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	IIC1 中断
IP1.2, IP0.2	外部中断 1	定时器 3 中断	ADC 中断
IP1.3, IP0.3	定时器 1 中断	外部中断 2	-
IP1.4, IP0.4	串行口中断	-	LVI 低压侦测中断
IP1.5, IP0.5	触摸按键中断	定时器 2 中断	IIC0 中断

表 10- 5: Polling sequence

中断源	顺序
外部中断 0	↓
PWM 中断	
定时器 0 中断	
看门狗中断	
IIC1 中断	
外部中断 1	
定时器 3 中断	
ADC 中断	
定时器 1 中断	
外部中断 2	
串行口中断	
LVI 低压侦测中断	
触摸按键中断	
定时器 2 中断	
IIC0 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CPU的时钟源，但外部周边的时钟源依旧保持，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0/1/2、低压侦测中断、看门狗中断、触摸按键中断）或者一个复位（看门狗复位及低压复位）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. 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 的电容所限制

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	
IIC0 function												
IIC0CTL	IIC0 control register	F9h	IIC0EN	MSS0	MAS0	AB0_EN	BF0_EN	-	IIC0BR[1:0]		03H	
IIC0S	IIC0 status register	F8h	-	MPIF0	LAIF0	RXIF0	TXIF0	RXAK0	TXAK0	RW or BB	00H	
IIC0A1	IIC0 Address 1 register	FAh	IIC0A1[7:1]							MATCH1or RW1		A0H
IIC0A2	IIC0 Address 2 register	FBh	IIC0A2[7:1]							MATCH2 or RW2		60H
IIC0RWD	IIC0 Read/Write register	FCh	IIC0RWD[7:0]								00H	
IIC0EBT	IIC0 Enaable Bus Transaction	FDh	FU0_EN	-	-	-	-	-	-	-	00H	
IIC1 function												
IIC1CTL	IIC1 control register	E9h	IIC1EN	MSS1	MAS1	AB1_EN	BF1_EN	IIC1BR[2:0]			04H	
IIC1S	IIC1 status register	E8h	-	MPIF1	LAIF1	RXIF1	TXIF1	RXAK1	TXAK1	RW or BB	00H	
IIC1A1	IIC1 Address 1 register	EAh	IIC1A1[7:1]							MATCH1or RW1		A0H
IIC1A2	IIC1 Address 2 register	F1h	IIC1A2[7:1]							MATCH2 or RW2		60H
IIC1RWD	IIC1 Read/Write register	ECh	IIC1RWD[7:0]								00H	
IIC1EBT	IIC1 Enaable Bus Transaction	EDh	FU1_EN	-	-	-	-	-	-	-	00H	

### 12.1 IIC interface 0

符号: IIC0CTL								地址: F9h	
7	6	5	4	3	2	1	0	Reset	
IIC0EN	MSS0	MAS0	AB0_EN	BF0_EN	-	IIC0BR[1:0]		03h	

IIC0EN: IIC0 模式使能

IIC0EN = 1, 使能

IIC0EN = 0, 禁止

MSS0: 主/从机模式选择

MSS0 = 1, 选择主机模式

MSS0 = 0, 选择从机模式

软件必须在设置其它寄存器之前置位

MAS0: 主模式的地址选择 (仅在主机模式中)

MAS0 = 0, 使用 IIC0A1

MAS0 = 1, 使用 IIC0A2

AB0\_EN: 仲裁失去了使能位. (仅主机模式)

当 AB0\_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.

BF0\_EN: 总线忙碌使能位. (仅主机模式)

当 BF0\_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 FU0\_EN[1:0] 设 10. 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.

IIC0BR[1:0]: 波特率选择 (仅在主机模式) 这里的 Fosc 为系统的频率, 系统默认为 Fosc / (256+5)

以便用户选择

IICBR[1:0]	Clock	Note
00	Fosc / (32+5)	-
01	Fosc / (64+5)	-
10	Fosc / (128+5)	-
11	Fosc / (256+5)	Default

符号: IIC0S

地址: F8H

7	6	5	4	3	2	1	0	Reset
-	MPIF0	LAIF0	RXIF0	TXIF0	RXAK0	TXAK0	RW or BB	00H

MPIF0: 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

LAIF0: 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

RxIF0: 数据接收中断标志位, 在 IIC0RWD (IIC0 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, IIC0 中断标志位 (IICIF0) 将自动清零。

TxIF0: 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 IIC0RWD (IIC0 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, IIC0 中断标志位 (IICIF0) 将自动清零。

RxAK0: 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 位数据传输总线上后。

TxAK0: 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 12-1.

RW or BB: 主机模式:

BB: 总线忙碌表示位

当侦测到 SCL=0 或 SDA=0 或总线产生开始信号, 此位将置高. 当侦测到停止信号, 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)

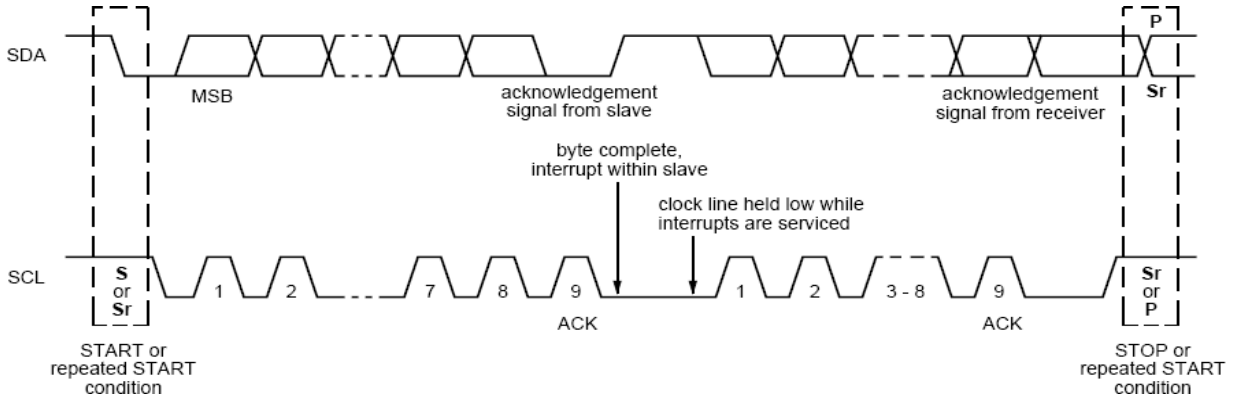
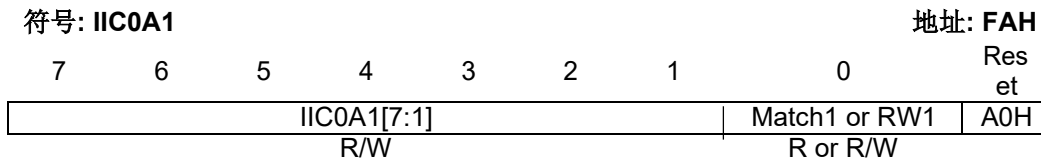


图 12-1: Acknowledgement bit in the 9th bit of a byte transmission



**从机模式:**

**IIC0A1[7:1]:** IIC0 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 **IIC0A1** 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:**

**IIC0A1[7:1]:** IIC0 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 **FU0\_EN[1:0]** 设 10, 该位将被作为从机方的 **RW** 发送, 它在 IIC 地址之后的抵 8<sup>th</sup> 位显示, 如图 12-2. 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。



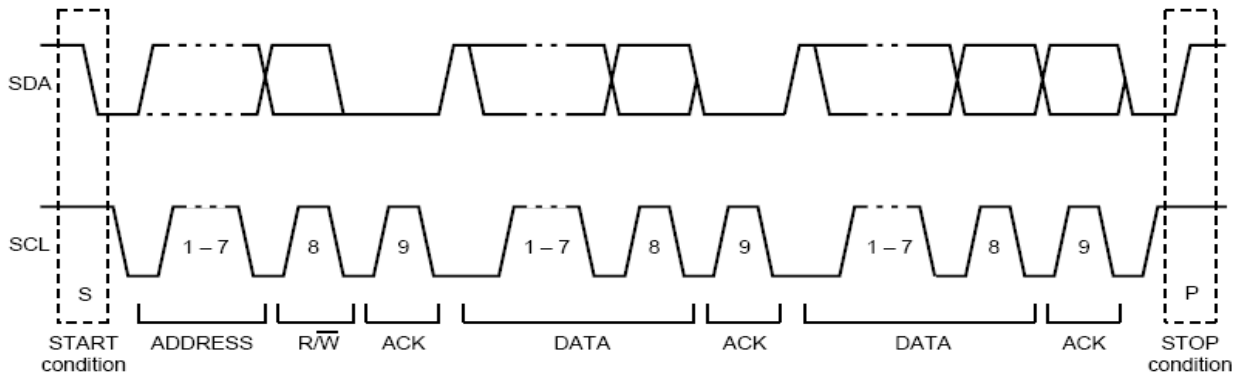


图 12-2: RW bit in the 8th bit after IIC address

符号: IIC0A2							地址: FBh	
7	6	5	4	3	2	1	0	Reset
IIC0A2[7:1]							Match2 or RW2	60h
R/W							R or R/W	

**从机模式:**

IIC0A2[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:**

IIC0A2[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 FU0\_EN[1:0] 设 10, 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: IIC0RWD							地址: FCh	
7	6	5	4	3	2	1	0	Reset
IIC0RWD[7:0]							00h	

IIC0RWD[7:0]: IIC0 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: IIC0EBT							地址: FDH	
7	6	5	4	3	2	1	0	Reset
FU0_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 IIC0A1 或 IIC0A2(由 MAS0 位选择)的地址
- 11: IIC 模块将于 SDA 及 SCL 发出停止信号.  
FU0\_EN[1:0] 此两位将自动清零,软件需重复写入.

**从机模式:**

- 01: FU0\_EN[1:0] 此两位于从机模式仅能写入 01.其它值是无效的.  
注意:  
FU0\_EN[1:0] 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).  
FU0\_EN[1:0] 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 IIC0RWD.  
FU0\_EN[1:0] 此两位将自动清零,软件需重复写入.

**12.2 IIC interface 1**

符号: IIC1CTL							地址: E9h
7	6	5	4	3	2	1	0
IIC1EN	MSS1	MAS1	AB1_EN	BF1_EN	IIC1BR[2:0]		Reset
							04h

IIC1EN: IIC1 模式使能

IIC1EN = 1, 使能

IIC1EN = 0, 禁止

MSS1: 主\从机模式选择

MSS1 = 1, 选择主机模式

MSS1 = 0, 选择从机模式

软件必须在设置其它寄存器之前置位

MAS1: 主模式的地址选择 (仅在主机模式中)

MAS1 = 0, 使用 IIC1A1

MAS1 = 1, 使用 IIC1A2

AB1\_EN: 仲裁失去了使能位. (仅主机模式)

当 AB1\_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.

**BF1\_EN:** 总线忙碌使能位. (仅主机模式)

当 **BF1\_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 **FU1\_EN[1:0]**设 10.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.

**IIC1BR[2:0]:** 波特率选择 (仅在主机模式) 这里的  $F_{osc}$  为系统的频率, 系统默认为  $F_{osc}/(128+5)$

以便用户选择

<b>IIC1BR[2:0]</b>	<b>Baud rate</b>
000	$F_{osc}/(8+5)$
001	$F_{osc}/(16+5)$
010	$F_{osc}/(32+5)$
011	$F_{osc}/(64+5)$
100	$F_{osc}/(128+5)$
101	$F_{osc}/(256+5)$
110	$F_{osc}/(512+5)$
111	$F_{osc}/(1024+5)$

符号: **IIC1S**

地址: **E8H**

7	6	5	4	3	2	1	0	Reset
-	MPIF1	LAIF1	RXIF1	TXIF1	RXAK1	TXAK1	RW or BB	00H

**MPIF1:** 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

**LAIF1:** 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

**RxIF1:** 数据接收中断标志位, 在 **IIC1RWD** (**IIC1** 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, **IIC1** 中断标志位 (**IICIF1**) 将自动清零。

**TxIF1:** 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 **IIC1RWD**(**IIC1** 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, **IIC1** 中断标志位 (**IICIF1**) 将自动清零。

**RxAK1:** 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后.

**TxAK1:** 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 12-1.

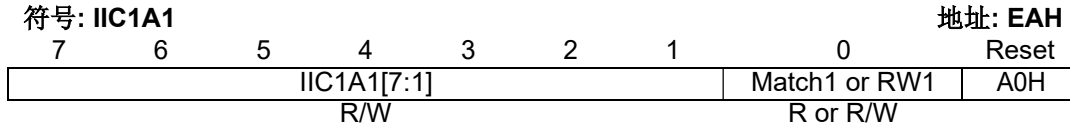
**RW or BB: 主机模式:**

**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)



**从机模式:**

**IIC1A1[7:1]:** IIC1 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

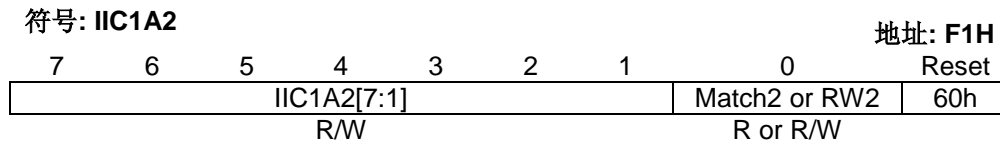
**Match1:** 当 **IIC1A1** 和来自主机模式方的接收地址匹配时，该位将被硬件置位，当 IIC 总线读或写第一笔资料时，该位将被自动清除。

**主机模式:**

**IIC1A1[7:1]:** IIC1 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 **FU1\_EN[1:0]** 设 10，该位将被作为从机方的 RW 发送，它在 IIC 地址之后的第 8<sup>th</sup> 位显示，如图 12-2。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。



**从机模式:**

**IIC1A2[7:1]:** IIC1 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

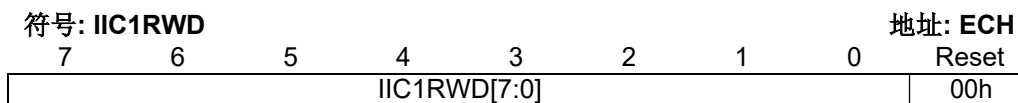
**Match2:** 当 **IIC1A2** 和来自主机模式方的接收地址匹配时，该位将被硬件置位，当 IIC 总线读或写第一笔资料时，该位将被自动清除。

**主机模式:**

**IIC1A2[7:1]:** IIC1 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 **FU1\_EN[1:0]** 设 10，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。



**IIC1RWD[7:0]:** IIC1 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: IIC1EBT							地址: EDH	
7	6	5	4	3	2	1	0	Reset
FU1_EN	-	-	-	-	-	-	-	00H

**主机模式:**

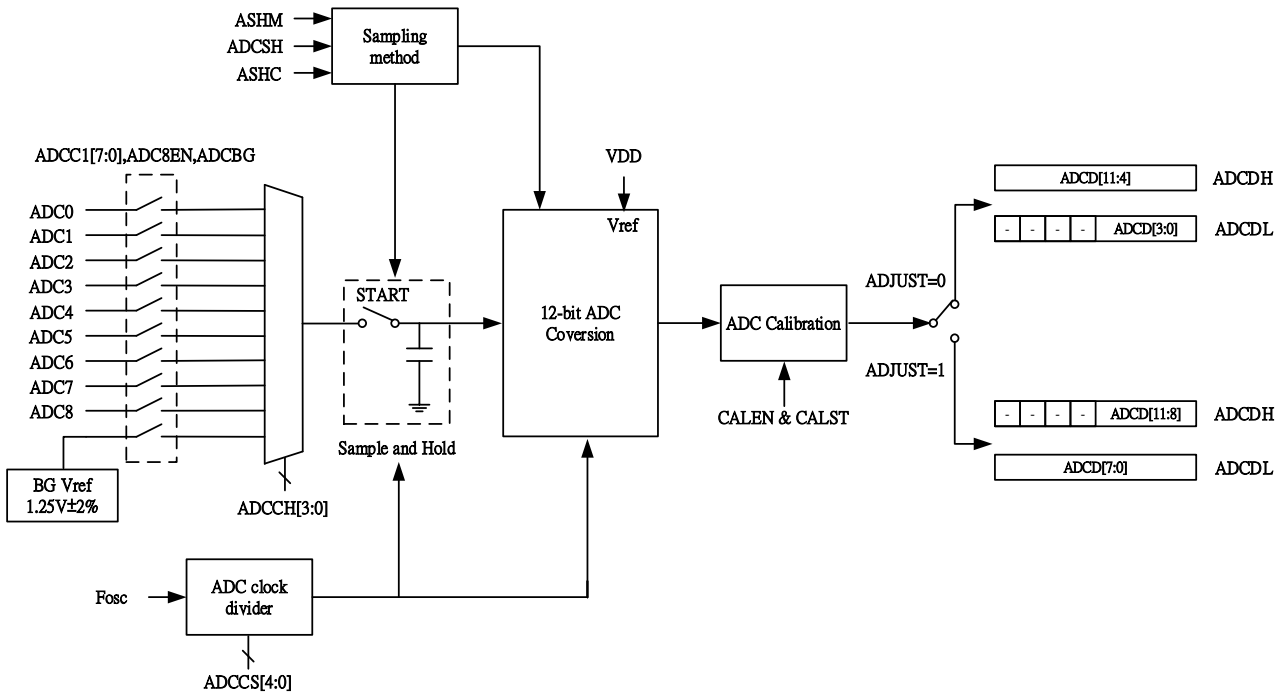
- 00: 保留
- 01: IIC 模块将致能以便由 SDA 及 SCL 读写数据.
- 10: IIC 模块将于 SDA 及 SCL 发出开始信号,再送出储存于 IIC1A1 或 IIC1A2(由 MAS1 位选择)的地址
- 11: IIC 模块将于 SDA 及 SCL 发出停止信号.  
FU1\_EN[1:0] 此两位将自动清零,软件需重复写入.

**从机模式:**

- 01: FU1\_EN[1:0] 此两位于从机模式仅能写入 01.其它值是无效的.  
注意:  
FU1\_EN[1:0] 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).  
FU1\_EN[1:0] 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 IIC1RWD.  
FU1\_EN[1:0] 此两位将自动清零,软件需重复写入.

### 13. 12位模拟数字转换器(ADC)

AS128A提供了外部9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。AS128A提供片内参考电压用于ADC模块自校验功能。



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	ADCCCH[3:0]			00H	
ADCDH	ADC Data High	ADH	ADCDH[7:0]								00H
ADC DL	ADC Data Low	AEH	ADC DL[7:0]								00H
ADCCS	ADC Clock Select	AFH	ADCB GE	ADC8E N	-	ADCCS[4:0]				00H	
ADCSH	ADC Sample and Hold Time	EFH	ADCSH[7:0]								00H

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: <b>ADCC2</b>							地址: <b>ACh</b>	
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	8
others	内部参考电压

ADJUST = 0:

符号: ADCDH 地址: ADh

7	6	5	4	3	2	1	0	Reset
ADCDC[11:4]								00H

符号: ADCDL 地址: AEh

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

ADJUST = 1:

符号: ADCDH 地址: ADh

7	6	5	4	3	2	1	0	Reset
-	-	-	-	ADCDC[11:8]				00H

符号: ADCDL 地址: AEh

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

ADCDC[11:0]: ADC 数字寄存器.

符号: ADCSH 地址: EFh

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

ADCDC [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	ADC8EN	-	ADCCS[4:0]				00H	

ADCBGE: ADC 测量 MCU 内部参考电压使能位。(内部参考电压 1.25V±2%)

0: 禁能。

1: 始能。

ADC8EN: ADC 通道 8 使能位。

0: 禁能 ADC 通道 8。



1: 使能 ADC 通道 8。

ADCCS[4:0]: ADC 时钟选择.

$$ADC\_Clock = \frac{Fclk}{6 \times (ADCCS[4:0] + 1)}$$

<p>ASHM=0, ADCSH=0x00</p>	<p><math>ADC\_Conversion\_Rate = \frac{ADC\_Clock}{16}</math></p>
<p>ASHM=0, ADCSH≠ 0x00</p>	<p><math>ADC\_Conversion\_Rate = \frac{ADC\_Clock}{19 + ADCSH[7:0]}</math></p> <p>※Sample and hold time by hardware control decisions.</p>
<p>ASHM=1</p>	<p><math>ADC\_Conversion\_Rate = \frac{ADC\_Clock}{16} + F/W\_Sampling\_Time</math></p> <p>※Sample and hold time by firmware control decisions.</p>

## 14. LVI & LVR – 低压侦测中断及低压侦测复位

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	ZPLV RF	LVRL PINTF	WDT WF	TKWF	WDTF	SWRF	LVRF	PORF	00H
LVC	Low voltage control	E6h	LVI_EN	ILVRE	LVRE	-	-	LVIS[2:0]			60H

符号: **RSTS** 地址: **A1h**

7	6	5	4	3	2	1	0	Reset
ZPLV RF	LVRLP INTF	WDT WF	TKWF	WDTF	SWRF	LVRF	PORF	00H

**ZPLVRF:** 零功率低压复位旗标

当通过 ZPLOVR 信号复位 MCU 时,硬件会将 ZPLVR 旗标设置为 1.该旗标可以由软件清除. (VIL: 0.7V)

**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.

**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.

符号: **LVC** 地址: **E6h**

7	6	5	4	3	2	1	0	Reset
LVI_EN	ILVRE	LVRE	-	-	LVIS[2:0]			60H

**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

**ILVRE:** 内部低功耗低压复位使能位

0: 禁能内部低功耗低电压重置功能.

1: 使能内部低功耗低电压重置功能.

**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.

**LVIS:** 低电压中断电压位阶选择:

LVIS[2:0]	Voltage Level (V)
000	2.8
001	3



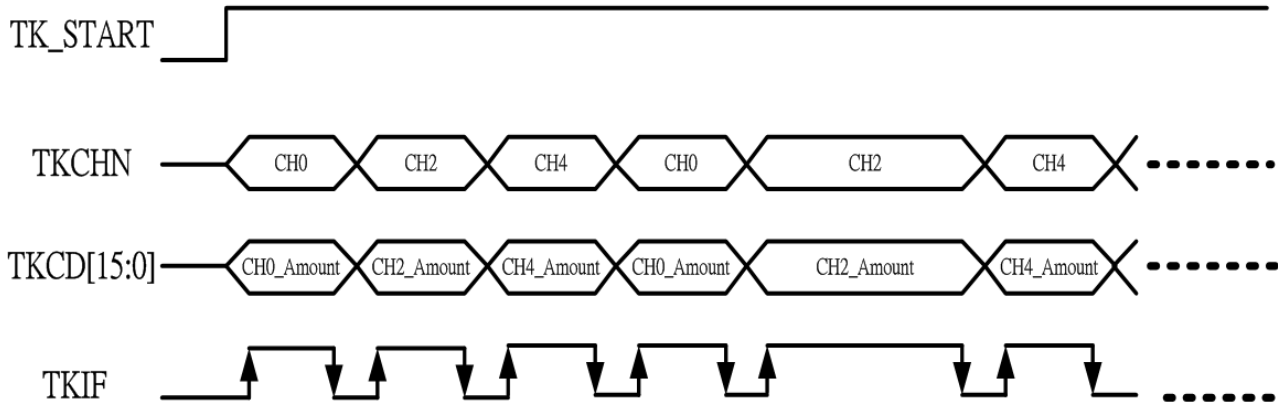
010	3.2
011	3.4
100	3.6
101	3.8
110	4.0
111	4.2

## 15. 触摸按键单元

### 15.1 触摸按键简介

触摸按键AS128A 内建 一组触摸模块，仅需在Cref上外挂10nf的电容，触摸引脚可与LED SEG 脚位复用，以分时多任务方式，同时运行触摸与LED，并可由软件简易的设定即可动作。

Enable Touch Key Channel 0、2、4



TKIF is set to high level by hardware.

TKIF is cleared to low level by firmware.

### 15.2 触摸按键特性

- 最多 16 通道触摸击键。
- 低成本的周边零件。
- 电容式触摸按键。
- 可以用来取代任何传统的机械式按键。
- 藉由软件设定，LED 与 触摸按键可分时多任务运行。

### 15.3 寄存器设定

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
Capacitor Sensor											
AUX	Auxiliary register	91H	BRG S	TK11 S	SICS[1:0]		TK10 S	SEG1 S	SEG0 S	-	00H
IEN2	Interrupt Enable 2 register	9AH	-	-	-	-	IETK	-	EWD T	-	00H
IRCON2	Interrupt request register 2	97H	-	-	-	-	TKIF	-	WDT IF	-	00H
RSTS	Reset Status	A1H	ZPLV RF	LVRL PINT F	WDT WF	TKW F	WDT RF	SWR F	LVRF	PORF	00H

TKEN0	Touch Key Enable 0	93H	TK7 EN	TK6 EN	TK5 EN	TK4 EN	TK3 EN	TK2 EN	TK1 EN	TK0EN	00H
TKEN1	Touch Key Enable 1	F3H	TK15 EN	TK14 EN	TK13 EN	TK12 EN	TK11 EN	TK10 EN	TK9 EN	TK8EN	00H
TKCON	Touch Key Control Reg.	9BH	TK_S TART	PDIB E	-	-	TK_SPEED[3:0]			00H	
TKSW	Touch Key Switch Reg.	FEH	TKSI F	LEDS IF	-	NSP	-	-	DTS[1:0]		10H
TKRUNTIME	Touch Key Running Time	FFEAH	TKRUNTIME[7:0]								00H
TKCHN	Touch Key Channel Number Reg.	9DH	-	-	-	TKCHN[4:0]				00H	
TKCDL	Touch Key Capture Data Low-byte Reg.	9EH	TKCD[7:0]								00H
TKCDH	Touch Key Capture Data Hi-byte Reg.	9FH	TKCD[15:8]								00H
TKSTATUS0	Touch Key Status 0	BBH	TKSTATUS0[7:0]								00H
TKSTATUS1	Touch Key Status 1	F4H	TKSTATUS1[7:0]								00H
TKPSSR	Touch Key Samping Rate	FFEDH	TKPSSR[7:0]								07H
GKWKCNT	Group Key Wake-up Counter	FFE0H	GKWKCNT[7:0]								02H
SKWKCNT	Single Key Wake-up Counter	FFE1H	SKWKCNT[7:0]								<b>02H</b>
TKRVC	Touch Key Reference Voltage Control	FFE2H	-	BLDO E	BLDO S	VTK[2:0]		CMPRVS[1:0]		6BH	
TKFC0	Touch Key Frequency Control 0	FFE3H	TKCS	TK_PS[1:0]		FHE	FHP[3:0]			20H	
TKFC1	Touch Key Frequency Control 1	FFE4H	SFC	-	-	-	FSS[3:0]			00H	
TKFC2	Touch Key Frequency Control 2	FFE5H	-	-	TKHF[5:0]					3FH	
TKFC3	Touch Key Frequency Control 3	FFE6H	-	-	TKLF[5:0]					3FH	
TKFC4	Touch Key Frequency Control 4	DCH	-	-	TKTF[5:0]					3FH	
TKDC	Touch Key Debounce Control	FFE9H	-	-	-	-	-	CMPD[2:0]		00H	
PDGPEN	Power-down Group Key	F5H	GP7E	GP6E	GP5E	GP4E	GP3E	GP2E	GP1E	GP0E	00H

	Enable										
GPUCNT	Touch Group Update Count	FFEBH	GPUCNT[7:0]								00H
SKUCNT	Touch Single Key Update Count	FFECH	SKUCNT[7:0]								01H
GP0S0	Group 0 Key Selection 0	FFC1H	TK7E	TK6E	TK5E	TK4E	TK3E	TK2E	TK1E	TK0E	00H
GP0S1	Group 0 Key Selection 1	FFC0H	TK15E	TK14E	TK13E	TK12E	TK11E	TK10E	TK9E	TK8E	00H
GP1S0	Group 1 Key Selection 0	FFC3H	TK7E	TK6E	TK5E	TK4E	TK3E	TK2E	TK1E	TK0E	00H
GP1S1	Group 1 Key Selection 1	FFC2H	TK15E	TK14E	TK13E	TK12E	TK11E	TK10E	TK9E	TK8E	00H
GP2S0	Group 2 Key Selection 0	FFC5H	TK7E	TK6E	TK5E	TK4E	TK3E	TK2E	TK1E	TK0E	00H
GP2S1	Group 2 Key Selection 1	FFC4H	TK15E	TK14E	TK13E	TK12E	TK11E	TK10E	TK9E	TK8E	00H
GP3S0	Group 3 Key Selection 0	FFC7H	TK7E	TK6E	TK5E	TK4E	TK3E	TK2E	TK1E	TK0E	00H
GP3S1	Group 3 Key Selection 1	FFC6H	TK15E	TK14E	TK13E	TK12E	TK11E	TK10E	TK9E	TK8E	00H
GP4S0	Group 4 Key Selection 0	FFC9H	TK7E	TK6E	TK5E	TK4E	TK3E	TK2E	TK1E	TK0E	00H
GP4S1	Group 4 Key Selection 1	FFC8H	TK15E	TK14E	TK13E	TK12E	TK11E	TK10E	TK9E	TK8E	00H
GP5S0	Group 5 Key Selection 0	FFCBH	TK7E	TK6E	TK5E	TK4E	TK3E	TK2E	TK1E	TK0E	00H
GP5S1	Group 5 Key Selection 1	FFCAH	TK15E	TK14E	TK13E	TK12E	TK11E	TK10E	TK9E	TK8E	00H
GP6S0	Group 6 Key Selection 0	FFDDH	TK7E	TK6E	TK5E	TK4E	TK3E	TK2E	TK1E	TK0E	00H
GP6S1	Group 6 Key Selection 1	FFCCH	TK15E	TK14E	TK13E	TK12E	TK11E	TK10E	TK9E	TK8E	00H
GP7S0	Group 7 Key Selection 0	FFCFH	TK7E	TK6E	TK5E	TK4E	TK3E	TK2E	TK1E	TK0E	00H
GP7S1	Group 7 Key Selection 1	FFCEH	TK15E	TK14E	TK13E	TK12E	TK11E	TK10E	TK9E	TK8E	00H

符号: AUX

地址: 91h

7	6	5	4	3	2	1	0	Reset
BRGS	TK11S	SICS[1:0]	TK10S	SEG1S	SEG0S	-	00H	

TK11S: Touch key 11 pin selection

0: P3.1

1: P1.7

TK10S: Touch key 10 pin selection

0: P3.0

1: P2.3

符号: <b>IEN2</b>							地址: <b>9Ah</b>	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	IETK	-	EWDT	-	00H

IETK: 触摸按键中断使能位。  
IETK = 0 – 禁能触摸按键中断。  
IETK = 1 – 使能触摸按键中断。

符号: <b>IRCON2</b>							地址: <b>97h</b>	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	TKIF	-	WDTIF	-	00H

TKIF: 触摸按键中断旗标, 必须以软件清除为 0。

符号: <b>RSTS</b>							地址: <b>A1h</b>	
7	6	5	4	3	2	1	0	Reset
-	LVRLP INTF	WDTWF	TKWF	WDTRF	SWRF	LVRF	PORF	00H

TKWF: 触摸中断掉电唤醒标志, 必须以软件清除为 0。

符号: <b>TKEN0</b>							地址: <b>93h</b>	
7	6	5	4	3	2	1	0	Reset
TK7EN	TK6EN	TK5EN	TK4EN	TK3EN	TK2EN	TK1EN	TK0EN	00H

TK7EN: 触摸按键通道 7 使能位。  
TK7EN = 1 – 使能触摸按键通道 7。  
TK6EN: 触摸按键通道 6 使能位。  
TK6EN = 1 – 使能触摸按键通道 6。  
TK5EN: 触摸按键通道 5 使能位。  
TK5EN = 1 – 使能触摸按键通道 5。  
TK4EN: 触摸按键通道 4 使能位。  
TK4EN = 1 – 使能触摸按键通道 4。  
TK3EN: 触摸按键通道 3 使能位。  
TK3EN = 1 – 使能触摸按键通道 3。  
TK2EN: 触摸按键通道 2 使能位。  
TK2EN = 1 – 使能触摸按键通道 2。  
TK1EN: 触摸按键通道 1 使能位。  
TK1EN = 1 – 使能触摸按键通道 1。  
TK0EN: 触摸按键通道 0 使能位。  
TK0EN = 1 – 使能触摸按键通道 0。

符号: **TKEN1**

地址: **F3h**

7	6	5	4	3	2	1	0	Reset
TK15EN	TK14EN	TK13EN	TK12EN	TK11EN	TK10EN	TK9EN	TK8EN	00H

TK15EN: 触摸按键通道 15 使能位。

TK15EN = 1 – 使能触摸按键通道 15。

TK14EN: 触摸按键通道 14 使能位。

TK14EN = 1 – 使能触摸按键通道 14。

TK13EN: 触摸按键通道 13 使能位。

TK13EN = 1 – 使能触摸按键通道 13。

TK12EN: 触摸按键通道 12 使能位。

TK12EN = 1 – 使能触摸按键通道 12。

TK11EN: 触摸按键通道 11 使能位。

TK11EN = 1 – 使能触摸按键通道 11。

TK10EN: 触摸按键通道 10 使能位。

TK10EN = 1 – 使能触摸按键通道 10。

TK9EN: 触摸按键通道 9 使能位。

TK9EN = 1 – 使能触摸按键通道 9。

TK8EN: 触摸按键通道 8 使能位。

TK8EN = 8 – 使能触摸按键通道 8。

※ 注意:TKEN 的控制 IO 组态功能需在 TK\_START=1 之状况下, 硬件才会设置 Touch 功能的 IO 组态。若 TK\_START=0, 就算 TKxEN=0xFF, 其 IO 组态则切回 GPIO PxM0 和 PxM1 的组态与 Px 设定的输出值。

※ “TKENx” SFR 只影响 Single key 的 enable / disable, 与 group key 的 enable/disable 无关

符号: **TKCON**

地址: **9Bh**

7	6	5	4	3	2	1	0	Reset
TK_ST ART	PDIBE	-	-	TK_SPEED[3:0]			00H	

TK\_START: 0 = 禁能触摸按键。

1 = 使能触摸按键。

PDIBE: 设置为仅 GPIO 输入模式时, 在掉电模式下触摸键的控制 IO 输入缓冲区。

0: 禁能输入缓冲区。

1: 使能输入缓冲区。

※ 用意是在 power down mode 下, touch key IO mode 选 input only mode 时, 将 input buffer 关闭, 这样就不会造成多耗电现象

TK\_SPEED[3:0]: 选择触摸传感器的充电速度。

TK\_SPEED="0000" Charge Level 0 (最快)

TK\_SPEED="0001" Charge Level 1

⋮

TK\_SPEED="1111" Charge Level 15 (最慢)



符号: **TKSW**

地址: **FEH**

7	6	5	4	3	2	1	0	Reset
TKSIF	LEDSIF	-	NSP	-	-	DTS[1:0]		10H

TKSIF: 当触摸功能开始运行时, 该标志由硬件置位。  
必须由软件清除。

LEDSIF: 当 LED 功能开始工作且触摸功能打开时, 该标志由硬件置位。  
必须由软件清除。

NSP: 当前尚未扫描的触摸键的端口状态 (开始触摸键)  
0—浮接。

1—由 GPIO 端口设置决定。

DTS[1:0]: 放电时间选择。

DTS [1:0]	Clock ( $F_{osc}$ )
00	64
01	96
10	128
11	192

符号: **TKRUNTIME**

地址: **FFEAH**

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

TKRUNTIME[7:0]: 触摸按键运行时间。

只有在 LED 驱动和触摸按键功能使能后该设置有效。此值决定触摸按键运行时间。

触摸按键运行时间 = TKRUNTIME[7:0] / LED 时钟。

※ 如果 TKRUNTIME = "0x00", LED 与触摸按键分享 I/O 是禁能。

符号: **TKCHN**

地址: **9Dh**

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

TKCHN[4:0]: 该寄存器指示计数器的扫描通道 (只读)。

※ 16~23(0x10~0x17)为指示 Group 0~7

符号: **TKCDL**

地址: **9Eh**

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

TKCD[7:0]: 该寄存器指示触摸按键撷取数据低字节的内容 (只读)。

符号: <b>TKCDH</b>							地址: <b>9Fh</b>	
7	6	5	4	3	2	1	0	Reset
TKCD[15:8]								00H

TKCD[15:8]: 该寄存器指示触摸按钮数据高字节的内容 (只读)。

符号: <b>TKSTATUS0</b>							地址: <b>BBh</b>	
7	6	5	4	3	2	1	0	Reset
TKSTATUS0[7:0]								00H

TKSTATUS0[7:0]: 记录触摸键通道 0~7 状态。

符号: <b>TKSTATUS1</b>							地址: <b>F4h</b>	
7	6	5	4	3	2	1	0	Reset
TKSTATUS1[7:0]								00H

TKSTATUS1[7:0]: 记录触摸键通道 8~15 状态。

符号: <b>TKPSSR</b>							地址: <b>FFEDh</b>	
7	6	5	4	3	2	1	0	Reset
TKPSSR [7:0]								07H

TKPSSR [7:0]: 此设置用于省电模式下触摸键的采样率。  
采样率=23KHz/[256\*( TKPSSR [7:0]+1 )]

符号: <b>GKWCNT</b>							地址: <b>FFE0h</b>	
7	6	5	4	3	2	1	0	Reset
GKWCNT [7:0]								02H

GKWCNT [7:0]: 此设置用于在省电模式下组触摸键达到触发值后唤醒。

- ※ Power down mode 下供硬件判断唤醒触发次数用
- ※ 真正唤醒是在第(数值+1)次

符号: <b>SKWCNT</b>							地址: <b>FFE1h</b>	
7	6	5	4	3	2	1	0	Reset
SKWCNT [7:0]								02H

SKWCNT [7:0]: 此设置用于在省电模式下触摸键达到触发值后唤醒。

- ※ Power down mode 下供硬件判断唤醒触发次数用
- ※ 真正唤醒是在第(数值+1)次

符号: **TKRVC**

地址: **FFE2h**

7	6	5	4	3	2	1	0	Reset
-	BLDOE	BLDOS	VTK[2:0]			CMPRVS[1:0]		6BH

BLDOE 触摸备用 LDO

0: 禁能

1: 使能

BLDOS: 触摸备用 LDO 电压选择

BLDOS	工作范围	条件
0	4.2~5.5	Applicable VTK=01b,10b,11b
1	3.6~5.5	Applicable VTK=10b,11b

VTK[2:0]: 触摸按键 LDO 电压选择。

VTK[2:0]	电压	工作范围
000	2.85	4.5~5.5
001	2.4	4.0~5.5
010	1.8	3.2~5.5
011	1.2	2.4~5.5
Other	Reserved	

CMPRVS[1:0]: 触摸比较器参考电压选择

CMPRVS [1:0]	Voltage	Note
00	$7/8 * VTK$	不适用 2.4V, 2.85V, 1.8V
01	$6/8 * VTK$	不适用 2.4V, 2.85V
10	$5/8 * VTK$	
11	$1/2 * VTK$	-

※ 比较器是 1.5V 电源, 当 V+ 脚 > 1.5V, 则比较器无法正常进行比较

符号: **TKFC0**

地址: **FFE3H**

7	6	5	4	3	2	1	0	Reset
TKCS	TK_PS[1:0]		FHE	FHP[3:0]				20H

TKCS: 触摸时钟源选择

0: MCU HIRC 16MHz

1: Touch HIRC 16MHz

TK\_PS[1:0]: 触摸计数器时钟源选择

TK_PS [1:0]	Clock (MHz)
00	16

01	32
10	PHY clock (sensor clock)
11	PHY clock (sensor clock)

FHE: 触摸跳频使能位

0: 禁能

1: 使能

※ If TKCS = 0, 跳频将不会执行。

FHP[3:0]: 触摸跳频周期

$$F_{\text{interval}} = (\text{FHI}[3:0] + 1) * 1 \text{ PHY clock (sensor clock)}$$

※ 指定多久跳频一次

符号: <b>TKFC1</b>							地址: <b>FFE4H</b>	
7	6	5	4	3	2	1	0	Reset
SFC	-	-	-	FSS[3:0]				00H

SFC: 触摸启动频率控制

0: 使用 TKHF 设置作为起始频率

1: 将上一个频道的最后一个频率设置为起始频率

※ 控制每个 channel 开始跳频的起点 (重新从 TKHF 设定的频率开始跳或继续不停跳频)

FSS[3:0]: 触摸频率步进选择

频率步进范围是 1 到 16

※ 指定一次跳频要跳几阶

符号: <b>TKFC4</b>							地址: <b>DCH</b>	
7	6	5	4	3	2	1	0	Reset
-	-	TKTF[5:0]						3FH

TKTF[5:0]: 触摸目标频率 (读/写)

※当 TKTF = 0x3F 时, Touch HIRC 的目标约为 16MHz。当 TKTF = 0x00 时, 约为 6MHz。

※ 读值会因 MCU 与 TK HIRC 间的频率不同, 造成读出来的值已是跳频若干次后的值

※ 手动指定频率设定条件: (TKHF = TKLF) + (SFC=1)

符号: <b>TKFC2</b>							地址: <b>FFE5H</b>	
7	6	5	4	3	2	1	0	Reset
-	-	TKHF[5:0]						3FH

TKHF[5:0]: 触摸高频

※ 初始频率为 16MHz

※ 若 TKHF < TKLF, 跳频将出现不可预期现象

符号: **TKFC3** 地址: **FFE6H**

7	6	5	4	3	2	1	0	Reset
-	-	TKLF[5:0]						3FH

TKLF[5:0]: 触摸低频  
 ※ 初始频率为 16MHz (3FH)  
 ※ 若 TKHF<TKLF, 跳频将出现不可预期现象  
 ※ 若 TKLF=00H, 在粗调 5%后, 最低频率约 6MHz

符号: **TKDC** 地址: **FFE9H**

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

CMPD[2:0]: 触摸比较器输出抖动时间选择

CMPD [2:0]	Clock (F <sub>osc</sub> )
000	1
001	2
010	4
011	6
100	8
101	10
110	12
111	14

※FOSC 的来源可能是 MCU HIRC 或 TK HIRC (包括跳频)。  
 ※时钟的滤波长度最好是 PHY 时钟的 1.1~1.5 倍, 最长不超过 2 倍。  
 如果最快的可调PHY时钟=8MHz (125ns), 则 CMPD 建议选择 000b (1 个时钟= 62.5ns)。

符号: **PDGPEN** 地址: **F5H**

7	6	5	4	3	2	1	0	Reset
GP7E	GP6E	GP5E	GP4E	GP3E	GP2E	GP1E	GP0E	00H

GPxE: 触摸组 0~7 启用。  
 0: 禁能  
 1: 使能  
 ※ 在 normal mode 若要使用 touch 群组, NGPEN 要设为 1

符号: **GPUCNT** 地址: **FFEBH**

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

GPUCNT[7:0]: 触摸组更新计数  
 ※ 指定每隔几次 group key 扫描后同时更新 group baseline/threshold, 若设为 0, 则表示每次扫描就会跟著更新 baseline  
 ※ Group key 扫描时间仍由 TKPSSR 来决定

- ※ 計算 baseline 最終是為了求出 threshold
- ※ 硬體計數是從 0 開始起算，即如果設 3，是在第 4 次才會進行 scan + update

符号: GPxS0、GPxS1								地址: FFC0H~FFCFH
7	6	5	4	3	2	1	0	Reset
TK7E	TK6E	TK5E	TK4E	TK3E	TK2E	TK1E	TK0E	00H
TK15E	TK14E	TK13E	TK12E	TK11E	TK10E	TK9E	TK8E	00H

GPxS0[7:0]: 触摸组 0~7 键选择

GPxS1[7:0]: ※ 每個 group 都可指定最多 16 key

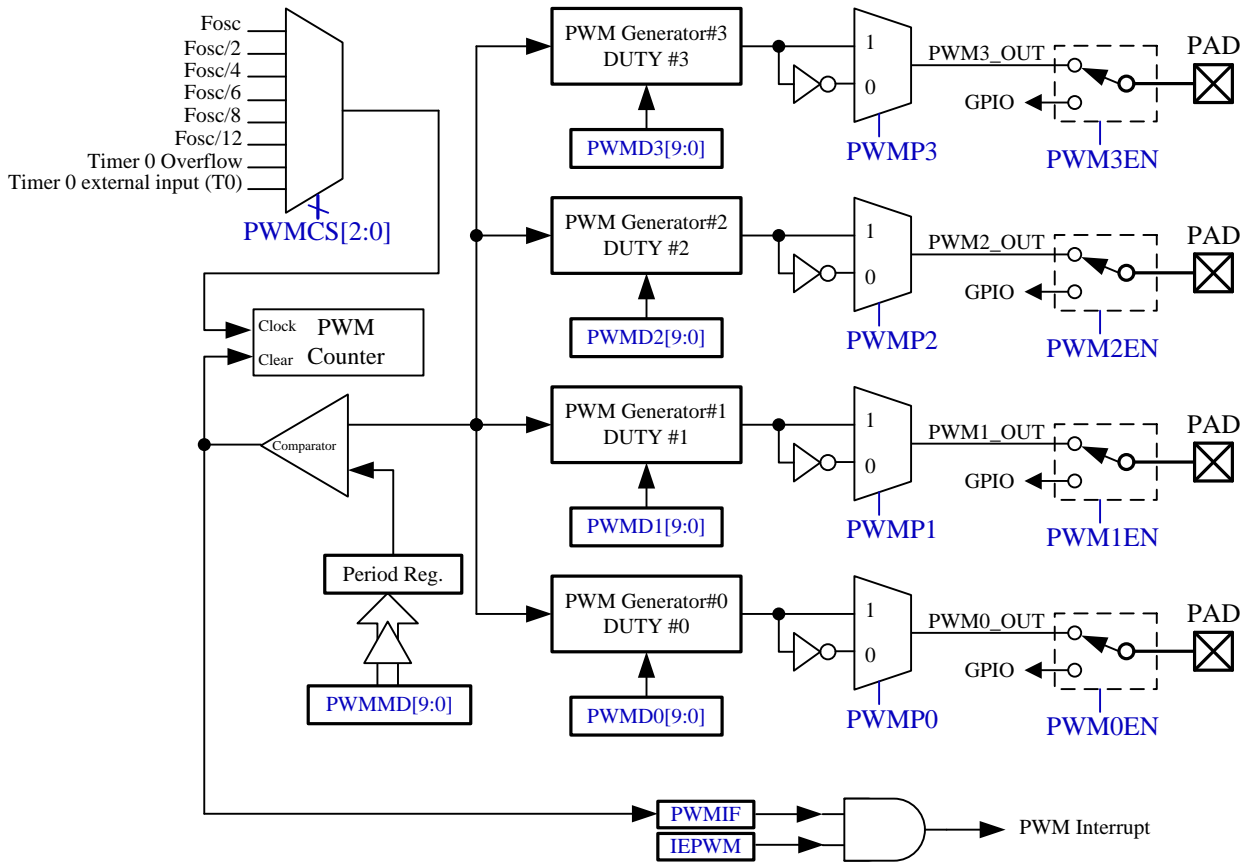
※ 8 個群組在 TKCHN 中顯示代碼為 16~23

※ 硬體只專注每個群組中設定的 key 進行掃描，所以 8 個群組中若有重覆設定的 key，硬體在個別群組中還是會進行掃描；未來應用若有某個 key 在 power down mode 需要較高反應速度時，或許可利用此特性

※ 在 normal mode, group key 的行為就跟 single key 一樣，硬體並不會再像 power down mode 一樣自動更新 threshold/baseline

## 16. 脉宽调制器(PWM)

AS128A提供4个通道的PWM输出.



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
<b>PWM</b>											
PWMC0	PWM Control 0 register	B5h	PWMCS[2:0]			PWMO MS	PWM3E N	PWM2E N	PWM1E N	PWM0E N	00H
PWMD0H	PWM 0 Data register high byte	BDh	-	-	-	-	-	-	PWMD0[9:8]		00H
PWMD0L	PWM 0 Data register low byte	BCh	PWMD0[7:0]								00H
PWMD1H	PWM 1 Data register high byte	BFh	-	-	-	-	-	-	PWMD1[9:8]		00H
PWMD1L	PWM 1 Data register low byte	BEh	PWMD1[7:0]								00H
PWMD2H	PWM 2 Data register high byte	B2h	-	-	-	-	-	-	PWMD2[9:8]		00H
PWMD2L	PWM 2 Data register low byte	B1h	PWMD2[7:0]								00H
PWMD3H	PWM 3 Data register high byte	B4h	-	-	-	-	-	-	PWMD3[9:8]		00H
PWMD3L	PWM 3 Data register low byte	B3h	PWMD3[7:0]								00H

PWMMDH	PWM Max Data register high byte	CFh	-	-	-	-	-	-	PWMMD[9:8]	00H
PWMMDL	PWM Max Data register low byte	CEh	PWMMD[7:0]							FFH
PWMC1	PWM Control 1	DDH	UDIS	CLS[2:0]		PWM3 PS	PWM2 PS	PWM1 PS	PWM0 PS	30H
PWMDT0	PWM 0 Dead Time	DEH	DTP0[1:0]		DT0[5:0]				00H	
PWMDT1	PWM 1 Dead Time	DFH	DTP1[1:0]		DT1[5:0]				00H	

符号: **PWMC0**

地址: **B5h**

7	6	5	4	3	2	1	0	Reset
PWMCS[2:0]			PWMOMS	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 (P1.6/T0)

PWMOMS: PWM 输出模式选择。

PWMOMS = 1 – 互补模式。

- Channel 0~3 将分为 2 组 (通道 0~1 和 2~3)
- 占空比由 PWMD0, PWMD2 决定。 (PWMD1, PWMD3 无效)
- 死区时间功能启用。

PWMOMS = 0 – 独立模式。

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: <b>PWMD0H</b>							地址: <b>BDh</b>	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	PWMD0[9:8]		00H

符号: <b>PWMD0L</b>							地址: <b>BCh</b>	
7	6	5	4	3	2	1	0	Reset
PWMD0[7:0]							00h	

PWMD0[9:0]: PWM 通道 0 数值寄存器.

符号: <b>PWMD1H</b>							地址: <b>BFh</b>	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	PWMD1[9:8]		00H

符号: <b>PWMD1L</b>							地址: <b>BEh</b>	
7	6	5	4	3	2	1	0	Reset
PWMD1[7:0]							00H	

PWMD1[9:0]: PWM 通道 1 数值寄存器.

符号: <b>PWMD2H</b>							地址: <b>B2h</b>	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	PWMD2[9:8]		00H

符号: <b>PWMD2L</b>							地址: <b>B1h</b>	
7	6	5	4	3	2	1	0	Reset
PWMD2[7:0]							00H	

PWMD2[9:0]: PWM 通道 2 数值寄存器.

符号: <b>PWMD3H</b>							地址: <b>B4h</b>	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	PWMD3[9:8]		00H

符号: <b>PWMD3L</b>							地址: <b>B3h</b>	
7	6	5	4	3	2	1	0	Reset
PWMD3[7:0]							00H	

PWMD3[9:0]: PWM 通道 3 数值寄存器.

符号: <b>PWMMDH</b>							地址: <b>CFh</b>	
7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	PWMMD[9:8]		00H

符号: <b>PWMMDL</b>							地址: <b>CEh</b>	
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.

符号: <b>PWMC1</b>							地址: <b>DDh</b>	
7	6	5	4	3	2	1	0	Reset
UDIS	CLS[2:0]			PWM3PS	PWM2PS	PWM1PS	PWM0PS	30H

UDIS: PWM 更新禁止位

0: 允许从占空比和周期缓冲寄存器更新

1: 禁止从占空比和周期缓冲寄存器更新

※ UDIS 位影响所有占空比缓冲寄存器 (PWMD0x, PWMD1x, PWMD2x, PWMD3x) 和 PWM 周期缓冲寄存器 (PWMMDx)。当 UDIS = 1 时, 占空比变化或周期值变化无效。

CLS[2:0]: 限流选择(VCC=5V, Vo=2V)

CLS[2:0]	Current (mA)
000	4
001	6
010	10
011	14 (default)
100	18
101	22
110	26
111	30

※PWM 电流限制必须有效, 并且 G3CS 位必须设置为 1。

※为了使 PWM 使用电流限制, 请注意 LED 的 SEG8、9、14 和 15 的电路设计。

※PWM 电流限制是与 LED 组 3 相同的电流限制电路。当同时在组 3 中打开 PWM 和 LED 时, 根据 SFR-LEDCS1.7 (G3CS) 的设置进行选择 (0: LED, 1: PWM)。限流令。

※PWM 输出低电平为 GPIO 模式灌电流= 50mA@1.5V, 而输出高电平为限流 LED 模式的源电流。

PWM3PS: PWM 通道 3 极性选择

0: 低电平有效

1: 高电平有效

PWM2PS: PWM 通道 2 极性选择

0: 低电平有效

1: 高电平有效

PWM1PS: PWM 通道 1 极性选择

0: 低电平有效

1: 高电平有效

PWM0PS: PWM 通道 0 极性选择

0: 低电平有效

1: 高电平有效

符号: <b>PWMDT0</b>							地址: <b>DEH</b>	
7	6	5	4	3	2	1	0	Reset
DTP0[1:0]			DT0[5:0]				00H	

DTP0[1:0]: 通道 0/1 的死区时间预分频器

DTP0 [1:0]	Mode
00	Fosc
01	Fosc/8
10	Fosc/16
11	Fosc/32

DT0[5:0]: PWM 通道 0/1 死区时间值

※死区时间:

$$\text{Dead Time} = \frac{\text{DT0}[5:0] + 1}{\text{DTP0 clock}}$$

※增加死区时间将缩短活动时间，延长非活动时间并导致占空比发生变化。

※ DT0 设 0x3F 会因加 1 而造成值变成 0x00，即关闭 Dead time 功能

符号: <b>PWMDT1</b>							地址: <b>DFH</b>	
7	6	5	4	3	2	1	0	Reset
DTP1[1:0]			DT1[5:0]				00H	

DTP1[1:0]: 通道 2/3 的死区时间预分频器

DTP1 [1:0]	Mode
00	Fosc
01	Fosc /8
10	Fosc /16
11	Fosc/32

DT1[5:0]: PWM 通道 2/3 死区时间值

※死区时间:

$$\text{Dead Time} = \frac{\text{DT1}[5:0] + 1}{\text{DTP1 clock}}$$

※增加死区时间将缩短活动时间，延长非活动时间并导致占空比发生变化。

※ DT1 设 0x3F 会因加 1 而造成值变成 0x00，即关闭 Dead time 功能

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

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

PWM0(Active Hi)  
PWM1(Active Hi)

PWM Without Dead Time



PWM With Dead Time



PWM0(Active Low)  
PWM1(Active Hi)

PWM Without Dead Time

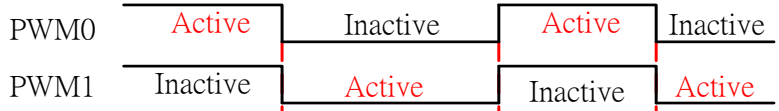


PWM With Dead Time

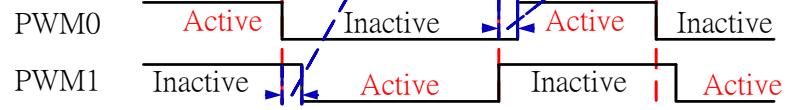


PWM0(Active Hi)  
PWM1(Active Low)

PWM Without Dead Time

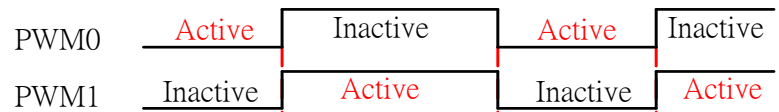


PWM With Dead Time

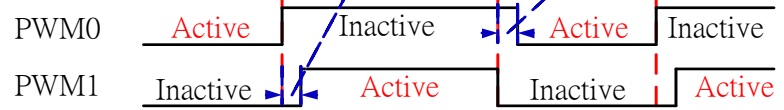


PWM0(Active Low)  
PWM1(Active Hi)

PWM Without Dead Time



PWM With Dead Time



## 17. LED 驱动器

### 17.1 LED Driver 简介

AS128A 内建 一组LED 驱动器，不需外加额外的硬件，就可直接推动LED，LED驱动引脚均可藉由软件设定选为 SEG & COM ， AS128A 内含16KB可编程存储器，256B+1024B SRAM 数据储存器等，且以 AS128A 为母体，给出 1/8 ~ 8/8 (full) duty 的 LED 驱动方法，及共阴、共阳以及阴阳同时驱动方式。

### 17.2 LED Driver 特性

- LED 驱动器包含一个控制器，一个占空比发生器及 7 个 COM 输出引脚和 16 个 Segment 输出引脚。
- 支持可调 1/8-8/8 占空比电压驱动方式。
- 通过 LEDMODE 位对驱动模式进行选择(共阴、共阳以及阴阳同时)。
- 控制器由显示数据 RAM 存储区和一个占空比发生器组成。
- 在上电复位、低电压复位、软件复位或看门狗复位期间，LED 被关闭。

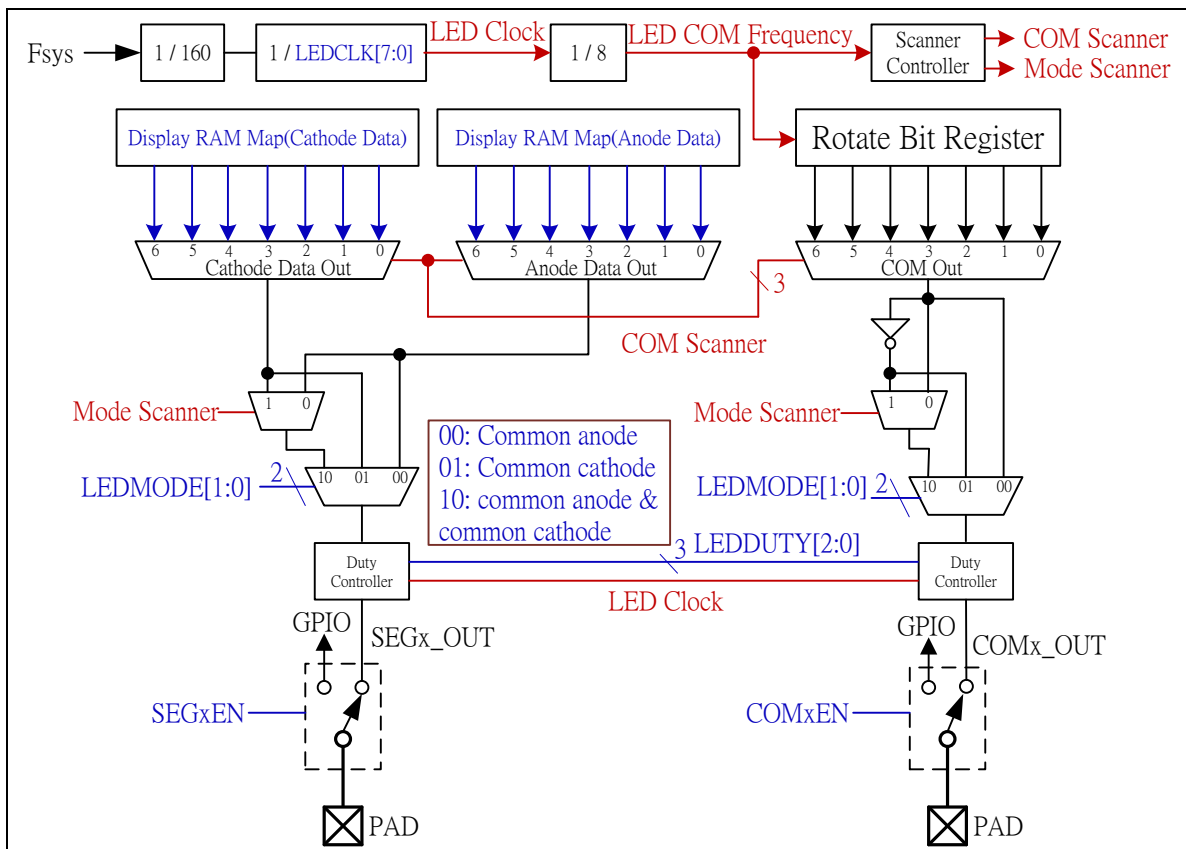


图 16 - 1: LED 寄存器工作状态

## 17.3 LED 应用说明

### 17.3.1 LED 模块设定方式

1. 先确定需使用到几个LED COM与SEG。
2. 决定LED扫描时间，设定LEDCLK值。
3. 设定SFR的TKRUNTIME=0。
4. 设定每一列的COM所对应之SEG要显示的Table表。
5. 设定LED 显示模式(共阴、共阳、阴阳同时)、显示Duty、启动LED。

### 17.3.2 SFR LEDCLK 设定说明

1. LEDCLK设定时，需先确认COM开启数量，且决定每秒钟要扫描LED几次，一般设定扫描至少30次以上，人眼才不会感觉有闪烁现象。
2. 范例: 当LED使用到6个COM，且预设LED每秒要扫描120次，算法如下:
3. 参阅Datasheet 公式:

$$\text{LED Clock} = \text{Fsys} / [160 * (\text{LEDCLK}[7:0])]$$

$$\text{LED COM Frequency} = \text{LED Clock} / 8$$

$$\text{LED Frame Rate} = \text{LED COM Frequency} / (\text{COM enable Number})$$

$$\text{LED Frame Time} = (\text{COM enable Number}) / \text{LED COM Frequency} \quad \square$$

■ 由于LED Frame Rate=120 Hz。 □

■ LED COM Frequency=120\*6=720 Hz。 □

■ LED Clock=LED COM Frequency\*8=720 \*8=5.76KHz □

■ LEDCLK[7:0]=Fosc/(LED Clock\*160) =16MHZ/(160\*5.76KHZ) =17 □

■ 故LEDCLK须设定17，此时LED Frame Rate才会为120 Hz”

17.4 寄存器设定

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
LED Driver											
AUX	Auxiliary register	91H	BRGS	TK11S	SICS[1:0]		TK10S	SEG1 S	SEG0 S	-	00H

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
LED Driver											
COMEN	COM Enable	FF1CH	-	COM6 EN	COM5 EN	COM4 EN	COM3 EN	COM2 EN	COM1 EN	COM0 EN	00H
SEGEN0	Segment Enable 0	FF1DH	SEG7 EN	SEG6 EN	SEG5 EN	SEG4 EN	SEG3 EN	SEG2 EN	SEG1EN	SEG0EN	00H
SEGEN1	Segment Enable 1	FF1EH	SEG1 5EN	SEG1 4EN	SEG1 3EN	SEG1 2EN	SEG1 1EN	SEG1 0EN	SEG9EN	SEG8EN	00H
LEDCON	LED Control	FF1FH	LEDS	DT1	LEDMODE[1:0]		DT0	LEDDUTY[2:0]			00H
LEDCLK	LED Clock control	FF20H	LEDCLK[7:0]								00H
LEDCS0	LED current limiting Selection 0	FF21H	-	G1CLS[2:0]			-	G0CLS[2:0]			33H
LEDCS1	LED current limiting Selection 1	FF22H	G3CS	G3CLS[2:0]			-	G2CLS[2:0]			33H

符号: **AUX** 地址: **91h**

7	6	5	4	3	2	1	0	Reset
BRGS	-	SICS[1:0]		-	SEG1S	SEG0S	-	00H

SEG1S: Segment 1 引脚选择

0: P0.1

1: P3.1

SEG0S: Segment 0 引脚选择

0: P0.0

1: P3.0

符号: **COMEN** 地址: **0xFF1C**

7	6	5	4	3	2	1	0	Reset
-	COM6EN	COM5EN	COM4EN	COM3EN	COM2EN	COM1EN	COM0EN	00H

COM6EN: 1: 使能 COM6。

0: 禁能 COM6。

COM5EN: 1: 使能 COM5。

0: 禁能 COM5。

COM4EN: 1: 使能 COM4。

0: 禁能 COM4。

- COM3EN: 1: 使能 COM3。  
0: 禁能 COM3。
- COM2EN: 1: 使能 COM2。  
0: 禁能 COM2。
- COM1EN: 1: 使能 COM1。  
0: 禁能 COM1。
- COM0EN: 1: 使能 COM0。  
0: 禁能 COM0。

符号: **SEGEN0** 地址: **0xFF1D**

7	6	5	4	3	2	1	0	Reset
SEG7EN	SEG6EN	SEG5EN	SEG4EN	SEG3EN	SEG2EN	SEG1EN	SEG0EN	00H

- SEG7EN: 1: 使能 SEG7。  
0: 禁能 SEG7。
- SEG6EN: 1: 使能 SEG6。  
0: 禁能 SEG6。
- SEG5EN: 1: 使能 SEG5。  
0: 禁能 SEG5。
- SEG4EN: 1: 使能 SEG4。  
0: 禁能 SEG4。
- SEG3EN: 1: 使能 SEG3。  
0: 禁能 SEG3。
- SEG2EN: 1: 使能 SEG2。  
0: 禁能 SEG2。
- SEG1EN: 1: 使能 SEG1。  
0: 禁能 SEG1。
- SEG0EN: 1: 使能 SEG0。  
0: 禁能 SEG0。

符号: **SEGEN1** 地址: **0xFF1E**

7	6	5	4	3	2	1	0	Reset
SEG15EN	SEG14EN	SEG13EN	SEG12EN	SEG11EN	SEG10EN	SEG9EN	SEG8EN	00H

- SEG15EN: 1: 使能 SEG15。  
0: 禁能 SEG15。
- SEG14EN: 1: 使能 SEG14。  
0: 禁能 SEG14。
- SEG13EN: 1: 使能 SEG13。  
0: 禁能 SEG13。
- SEG12EN: 1: 使能 SEG12。  
0: 禁能 SEG12。
- SEG11EN: 1: 使能 SEG11。  
0: 禁能 SEG11。
- SEG10EN: 1: 使能 SEG10。  
0: 禁能 SEG10。



- SEG9EN: 1: 使能 SEG9。  
0: 禁能 SEG9。  
SEG8EN: 1: 使能 SEG8。  
0: 禁能 SEG8。

※注意:COMEN、SEGEN0、SEGEN1 的控制 IO 组态功能需在 LEDS=1 之状况下, 硬件才会设置 LED 功能的 IO 组态。若 LEDS =0, 就算 COMEN、SEGEN0、SEGEN1 都为 0xFF, 其 IO 组态则切回 GPIO PxM0 和 PxM1 的组态与 Px 设定的输出值。

符号: LEDCON						地址: 0xFF1F	
7	6	5	4	3	2	1	0
LEDS	DT1	LEDMODE[1:0]		DT0	LEDDUTY[2:0]		Reset
							00H

LEDS: LED 驱动器使能位。

- 1: 使能。  
0: 禁能。

DT[1:0]: LED 死区时间选择

DT[1:0]	Condition
00	8/Fosc
01	16/Fosc
10	32/Fosc
11	4/Fosc

※在死区时间内, COM&SEG 被拉高。

LEDMODE[1:0]: LED 模式选择位。

- 00: 共阳驱动模式。  
01: 共阴驱动模式。  
10: 共阳 & 共阴驱动模式。(不支持同时使用触摸按键)  
11: 保留。

LEDDUTY [2:0]: LED duty 选择位。

- 000: 8/8 duty  
001: 1/8 duty  
010: 2/8 duty  
011: 3/8 duty  
100: 4/8 duty  
101: 5/8 duty  
110: 6/8 duty  
111: 7/8 duty

符号: LEDCLK							地址: 0xFF20	
7	6	5	4	3	2	1	0	
LEDCLK[7:0]							Reset	
							00H	

LEDCLK: LED 时钟选择:

$$\text{LED Clock} = \text{Fsys} / (160 * \text{LEDCLK}[7:0])$$

$$\text{LED COM Frequency} = \text{LED Clock} / 8$$

$$\text{LED Frame Rata} = \text{LED COM Frequency} / (\text{COM enable Number})$$

LED Frame Time=(COM enable Number) / LED COM Frequency

符号: LEDCS0~1

地址: FF21~FF22H

7	6	5	4	3	2	1	0	Reset
-	G1CLS [2:0]			-	G0CLS[2:0]			33H
G3CS		G3CLS [2:0]			-	G2CLS [2:0]		33H

GxCLS[2:0]: Segment group 0~4 限流选择(VCC=5V, Vo=2V)

GxCLS[2:0]	Current (mA)
000	4
001	6
010	10
011	14 (default)
100	18
101	22
110	26
111	30

Group	Segment	GPIO
0	0_0, 1_0, 2~4	P0.0~0.4
1	5~7, 10	P0.5~0.7, P2.3
2	0_1, 1_1, 11~13	P3.0~3.4
3	8, 9, 14, 15	P1.0, 1.1, 1.6, 1.7

G3CS: Group 3 源电流限制设置

0: 应用 LED 电流限制选择。

1: 应用 PWM 电流限制选择。

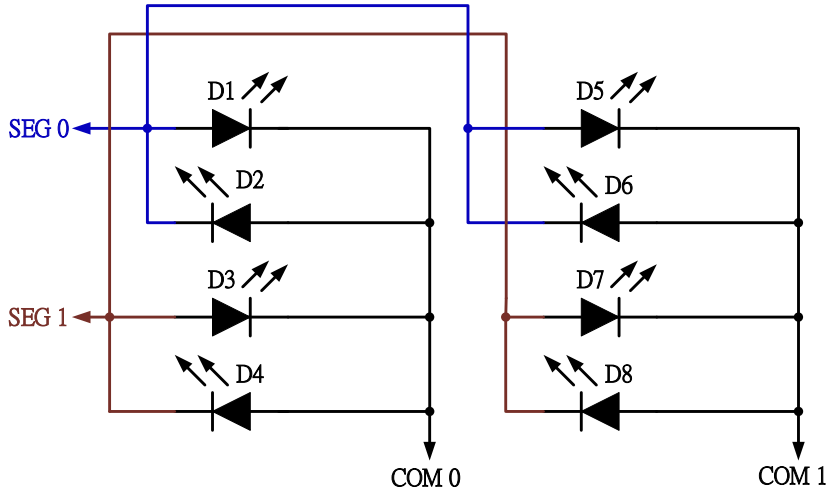
## 17.5 Display RAM Map

### LED RAM Map:

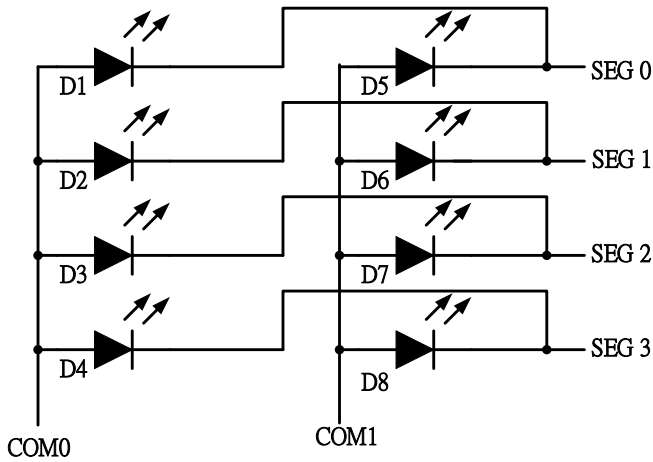
Description	Indirect	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	RST
COM0_AH	0xFF00	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM0_AL	0xFF01	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM1_AH	0xFF02	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM1_AL	0xFF03	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM2_AH	0xFF04	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM2_AL	0xFF05	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM3_AH	0xFF06	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM3_AL	0xFF07	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM4_AH	0xFF08	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM4_AL	0xFF09	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM5_AH	0xFF0A	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM5_AL	0xFF0B	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM6_AH	0xFF0C	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM6_AL	0xFF0D	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM0_CH	0xFF0E	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM0_CL	0xFF0F	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM1_CH	0xFF10	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM1_CL	0xFF11	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM2_CH	0xFF12	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM2_CL	0xFF13	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM3_CH	0xFF14	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM3_CL	0xFF15	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM4_CH	0xFF16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM4_CL	0xFF17	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM5_CH	0xFF18	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM5_CL	0xFF19	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM6_CH	0xFF1A	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM6_CL	0xFF1B	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00

## 17.6 LED 应用电路参考

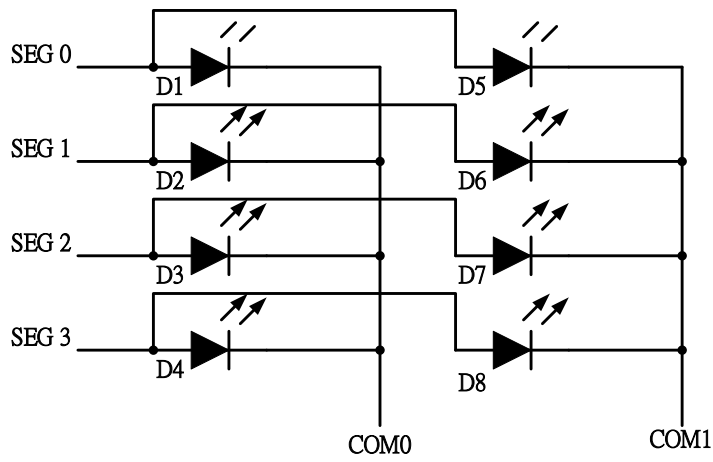
### 17.6.1 LED 共阴同时驱动电路



### 17.6.2 LED 共阳驱动电路

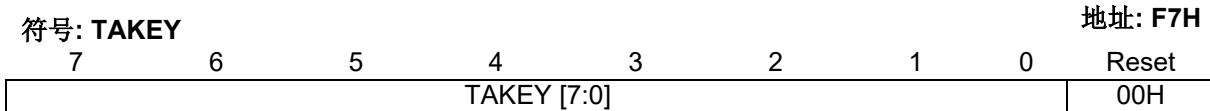


### 17.6.3 LED 共阴驱动电路



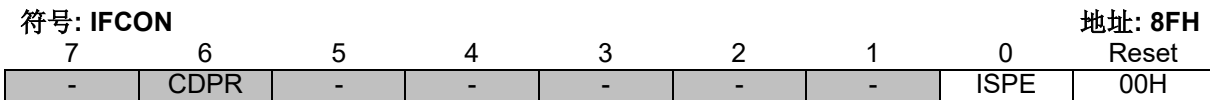
## 18. EEPROM

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
EEPROM function											
TAKEY	Time Access Key register	F7h	TAKEY [7:0]								00H
IFCON	Interface Control register	8Fh	-	CDPR	-	-	-	-	-	ISPE	00H
ISPF AH	EEPROM Address – High register	E1h	ISPF AH [7:0]								FFH
ISPF AL	EEPROM Address – Low register	E2h	ISPF AL [7:0]								FFH
ISPF DL	EEPROM Low Data register	E3h	ISPF DL [7:0]								FFH
ISPF C	EEPROM Control register	E4h	-	-	-	-	-	ISPF.2	ISPF.1	ISPF.0	00H

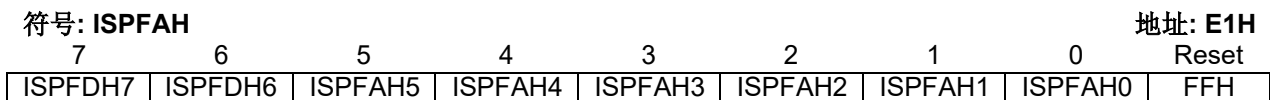


EEPROM使能位(ISPE)默认为只读,软件必须连续的对TAKEY寄存器写三个特定值55h,AAH,和5Ah,使ISPE位可写.这是:

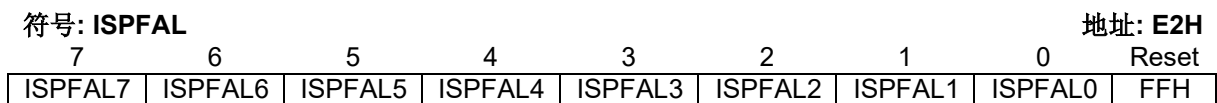
```
MOV TAKEY, #55h
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah
```



位 0(ISPE) of IFCON 是 EEPROM 使能位,用户可通过设置 ISPE 位为 1 使能 AS128A 的 EEPROM 功能,通过设置 ISPE 为 0 来禁止 EEPROM 功能.ISPE 的作用像一个加密匙,用户可禁止 EEPROM 功能以保护 EEPROM 不被意外的擦除. EEPROM 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.



ISPF AH [7:0]: 使用 EEPROM 功能之地址高字节



ISPF AL [7:0]: 使用 EEPROM 功能之地址低字节

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]: 使用 EEPROM 功能之数值.

ISPFDL 寄存器提供 EEPROM 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
-	-	-	-	-	ISPF[2]	ISPF[1]	ISPF[0]	00H

ISPF [2:0]: EEPROM 功能选择位.

ISPF[2:0]	EEPROM function
000	reserved
001	reserved
010	reserved
011	Write EEPROM
100	Read EEPROM
101	reserved
110	Read option
111	reserved

选择的 EEPROM 功能在软件对 ISPFC 寄存器写入数据时将会执行一次.

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

EX1: 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 EEPROM 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)

EX2: 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 EEPROM function
MOV ISPF AH, #00H         ; set flash address-high, 00H
MOV ISPF AL, #05H         ; set flash address-low, 05H
MOV ISPF DL, #33H         ; set flash data to be programmed, data = 33H
MOV ISPF C, #03H          ; start to program #33H to the EEPROM address $0005H

```

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

EX3: 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 EEPROM function
MOV ISPF AH, #00H    ; set flash address-high, 00H
MOV ISPF AL, #04H    ; set flash address-low, 04H
MOV ISPF C, #04H     ; start to program read the EEPROM address $0004H to ISPF DL
MOV A, ISPF DL       ; read ISPF DL and store it in Accumulator
```

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

EX4: 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 EEPROM function
MOV ISPF AH, #00H    ; set flash address-high, 00H
MOV ISPF AL, #04H    ; set flash address-low, 04H
MOV ISPF C, #06H     ; start to program read the option address $0004H to ISPF DL
MOV A, ISPF DL       ; read ISPF DL and store it in Accumulator
```

EX5: 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
```

EX6: 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
```



## 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	VCC	Min	Typical	Max	Units	Conditions
VIL1	Input Low-voltage	Port 0,1,2,3				0.2V <sub>cc</sub>	V	
VIH1	Input High-voltage	Port 0,1,2,3		0.8V <sub>cc</sub>			V	
V <sub>hys</sub>	Hysteresis voltage	Port 0,1,2,3			0.2		V	VCC=5V
V <sub>hys</sub>	Hysteresis voltage	I2C			0.7		V	VCC=5V
IOL1	Sink Current	Port 0,1,2,3	5	40			mA	VOL=0.45V
			3		30		mA	
IOL2	Sink Current	COM 0~6	5	64	80	96	mA	VOL=0.45V
			3		70		mA	
IOL3	Sink Current	Segment 0~15, PWM 0~3	5		4		mA	VOL=1.5V
			5		6		mA	
			5		10		mA	
			5		14		mA	
			5		18		mA	
			5		22		mA	
			5		26		mA	
IOH1	Source Current (Pull-Up)	Port 0,1,2,3	5	0.36			mA	VOH=2.6V
			3	0.07			mA	VOH=2.4V
IOH2	Source Current (Push-Pull)	Port 0,1,2,3	5	15			mA	VOH=4.5V
			3		6		mA	VOH=2.6V
IOH3	Source Current	Segment 0~15, PWM 0~3	5		4		mA	VOH=3.5V
			5		6		mA	
			5		10		mA	
			5		14		mA	
			5		18		mA	
			5		22		mA	
			5		26		mA	
IOH4	Source Current	COM 0~6	5		60		mA	VOH=4.5V

			3		30		mA	
ICC0	Power Supply Current		5		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
ICC1	Power Supply Current		5		10		uA	Power down mode, Touch on(8Hz, 16 touch key)

### ADC Characteristics

	Symbol	Test Condition	MIN	TYP	MAX	Unit
Operation	VDD	VDD	3		5.5	V
Bandgap input		±2%		1.25		V
Resolution					12	bit
Conversion time				16		T <sub>sample_rate</sub>
Sample rate				167		KHz
Integral Non-Linearity Error	INL		-2		2	LSB
Differential Non-Linearity	DNL		-1		1	LSB
Clock frequency	ADCCLK				2.66	MHz