MEGAWIN
MAKE YOU WIN

8051 内核-单片机

MG87FEL2051_4051_6051
规格书

版本: **A1.0**

2014/03 version A1.0

特性

- 80C51 中央处理单元
- MG87FE/L2051 2KB 闪存，4051 4KB 闪存，6051 6KB 闪存
- 工作电压为：E 型 4.5V~5.5V，L 型 2.4V~3.6V
- 工作频率：12T 模式下最高为 48MHz，6T 模式下最高为 24MHz
  - 外部晶体振荡器模式
  - 内部 RC 振荡器模式，温度在 -40~85℃范围内频漂+/-4%，并有 6 种频率可供选择

	内部振荡器频率
1	6MHz
2	11.059MHz
3	12MHz
4	22.118MHz
5	24MHz
6	24.576MHz

- ISP 存储空间能自定义大小，可以为 0.5K/1K/1.5K~3.5K
- IAP 容量：1KB IAP 存储空间
- 片内有 256 字节的静态数据存储器 RAM
- 基于 Flash 访问的代码安全机制，数据加密以及程序存储空间加锁 3 种代码安全机制
- 两个 16 位定时/计数器
- PWM-Timer 可做 PWM 发生器或普通 8 位定时器，并可选择 INT3 中断向量
- 7 个中断源，4 级优先级中断功能
- 增强型的 UART，提供帧错误检测和硬件地址识别
- 15 位的看门狗定时器，能 8 位预分频，CPU 一次性启用或上电一次性启用，启用后软件无法停止
- 有空闲模式 IDLE 和掉电模式 Power-down 两种电源控制模式，空闲模式可由任意一个中断源唤醒，掉电模式下可由 /INT0（P3.2）、/INT1（P3.3）、/INT2（P4.3）、/INT3（P4.2）及其它 IO 口皆可唤醒
- 17 个 IO 口：P1[7:0]、P3[7,5:0]、P4.2/INT3/XTAL2，P4.3/INT2/XTAL1
- 内建模拟比较器 INT2 中断向量选择，输入口 AIN0（V+）/P1.0 和 AIN1(V-)/P1.1，由 P3.6 输出（内部 CPU 可读）
- 封装类型：PDIP-20，SOP-20

型 号	封装类型	说 明
MG87Fxy051AE20	PDIP-20	x = E:5.0V , L:3.3V y = 2, 4, 6 . 2051/4051/6051
MG87Fxy051AS20	SOP-20	

目录

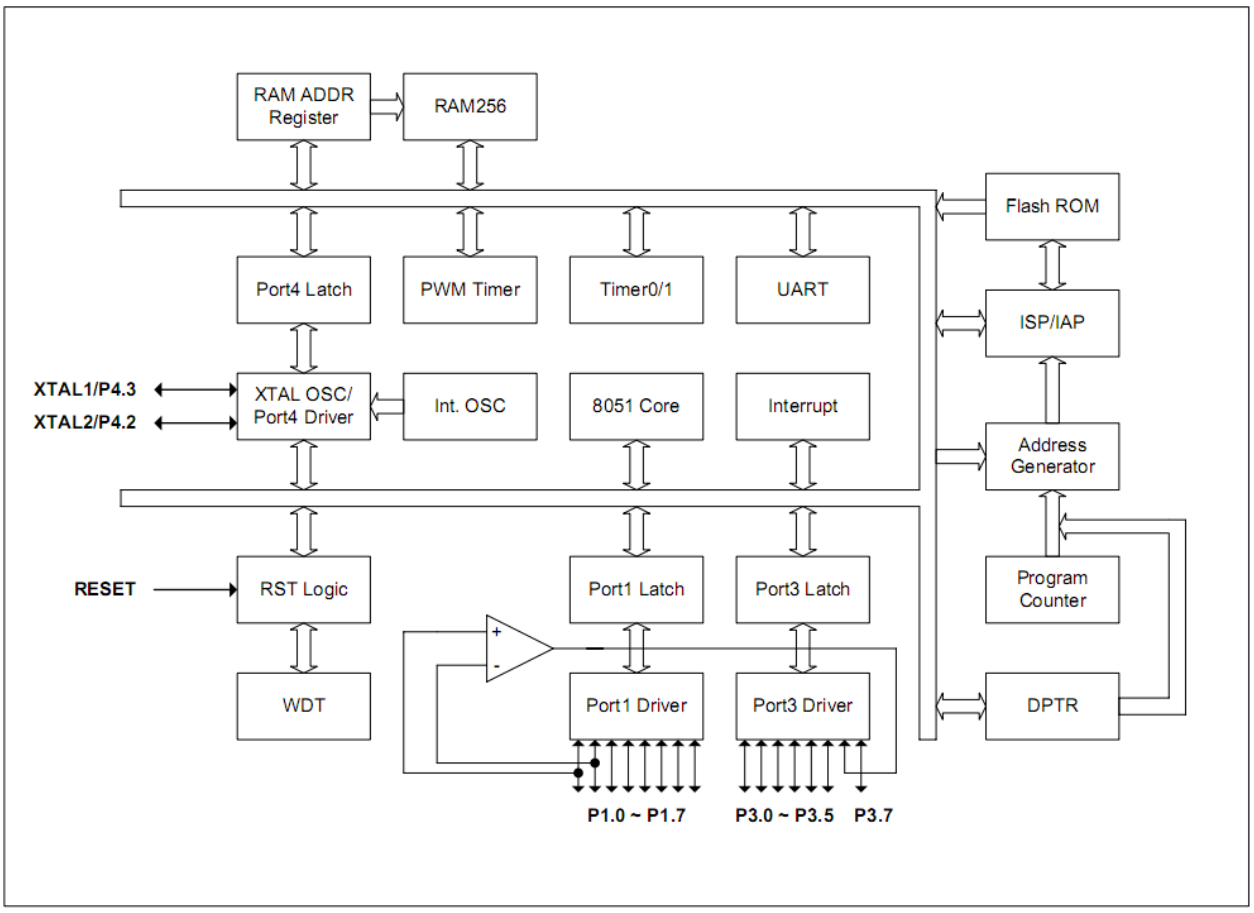
特性 3
目录 5
1. 概述 7
2. 方框图 8
3. 引脚定义 9
3.1. 封装 9
3.2. 引脚描述（PDIP-20、SOP-20） 10
4. 8051 CPU 功能说明 11
4.1. CPU 寄存器 11
4.2. CPU 时序 12
4.3. 寻址方式 12
5. 存储器结构 14
5.1. 片上闪存 14
5.2. 片上数据存储器 RAM 15
6. 特殊功能寄存器（SFR） 16
6.1. SFR 映射图 16
6.2. SFR 各位分配 17
7. I/O 口 19
7.1. I/O 结构 19
7.2. 通用 GPIO 口 P1/3/4 结构 19
7.3. P1 口寄存器 19
7.4. P3 口寄存器 19
7.5. P4 口寄存器 20
7.6. GPIO 示例代码 21
8. 中断 22
8.1. 中断结构 22
8.2. 中断寄存器 22
9. 定时器/计数器 26
9.1. Timer0 与 Timer1 26
9.1.1. 模式 0 26
9.1.2. 模式 1 26
9.1.3. 模式 2 27
9.1.4. 模式 3 27
9.1.5. Timer0/1 寄存器 27
9.2. 定时器 0/1 示例代码 29
9.3. PWM-Timer（PWM 定时器） 33
9.3.1. PWM 定时器结构 33
9.3.2. PWM 定时器寄存器 34
9.4. PWM 示例代码 36
10. 串口 UART 37
10.1. UART 结构 37
10.2. UART 寄存器 38
10.3. 13.5 串行口示例代码 40

11. 模拟比较器 42
11.1. 模拟比较器结构 42
11.2. 模拟比较器寄存器 42
12. 看门狗定时器（WDT） 44
12.1. WDT 结构 44
12.2. WDT 寄存器 44
12.3. WDT 示例代码 46
13. 复位 47
13.1. 复位源 47
14. 电源管理 48
14.1. 省电模式 48
14.2. 空闲 IDLE 模式 48
14.3. 掉电模式 48
14.4. 中断唤醒 48
14.5. 复位唤醒 49
14.6. 普通 I/O GPIO 口唤醒 49
14.7. 电源控制寄存器 49
15. 系统时钟 51
15.1. 时钟结构 51
15.2. 时钟寄存器 51
16. 在线编程（ISP） 53
17. 应用程序可编程（IAP） 55
17.1. IAP 示例代码 56
18. 附加的特殊功能寄存器 57
19. 硬件选项设置 58
20. 最大绝对额定参数 59
21. 电气特性 60
21.1. 直流特性 60
22. 封装尺寸 61
23. 指令介绍 63
24. 修订历史 67

1. 概述

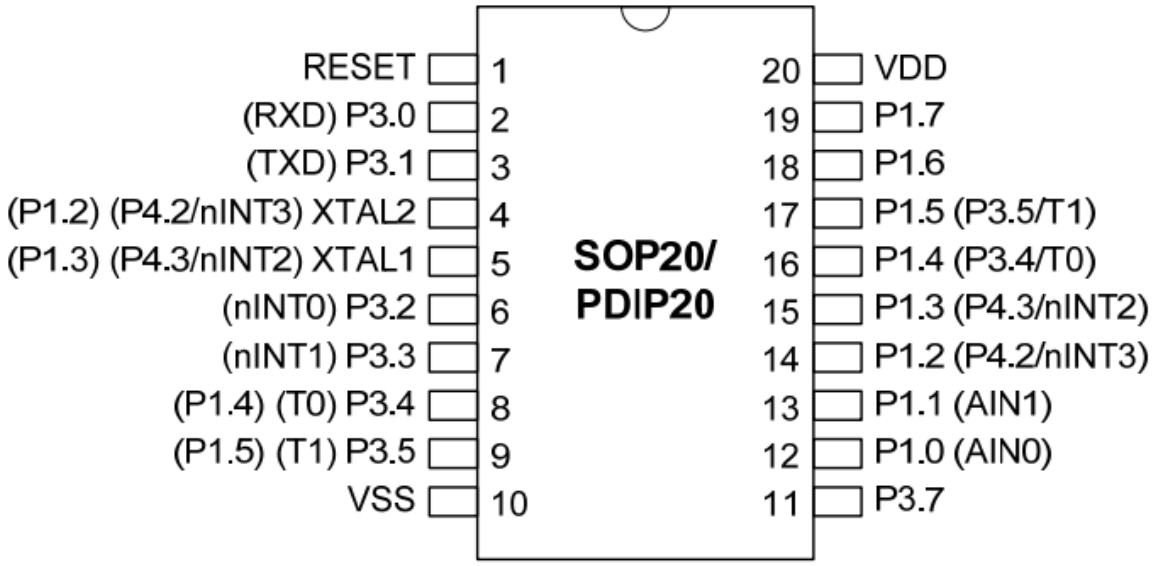
MG87FE/L2051/4051/6051 是一颗 8 位的单片机，它的指令集完全兼容工业标准的 80C51 系列的单片机。分别有 2K/4K/6K 字节的闪存程序存储空间和 256 字节的数据存储空间，为芯片提供了广泛的应用领域。在线编程功能使用户更方便更新程序和数据。芯片的一个机器周期可以为 6 个或者 12 个时钟周期。芯片有一个 8 位 IO 口(P1)、一个 7 位 IO 口（P30~P35，P37），2 个 16 位的定时器/计数器，一个 8 位 PWM，7 个 4 级优先级的中断源，一个增强型的 UART，一个精准模拟比较器，外挂晶体振荡器（XTAL2/P42、XTAL1/P43）以及一个高精度的内部 RC 振荡器。

2. 方框图


RAM ADDR Register
RAM256
Flash ROM
Port4 Latch
PWM Timer
Timer0/1
UART
ISP/IAP
XTAL1/P4.3
XTAL2/P4.2
XTAL OSC/ Port4 Driver
Int. OSC
8051 Core
Interrupt
Address Generator
RESET
RST Logic
Port1 Latch
Port3 Latch
Program Counter
WDT
Port1 Driver
Port3 Driver
DPTR
P1.0 ~ P1.7
P3.0 ~ P3.5
P3.7

3. 引脚定义

3.1. 封装


RESET 1
(RXD) P3.0 2
(TXD) P3.1 3
(P1.2) (P4.2/nINT3) XTAL2 4
(P1.3) (P4.3/nINT2) XTAL1 5
(nINT0) P3.2 6
(nINT1) P3.3 7
(P1.4) (T0) P3.4 8
(P1.5) (T1) P3.5 9
VSS 10
SOP20/
PDIP20
20 VDD
19 P1.7
18 P1.6
17 P1.5 (P3.5/T1)
16 P1.4 (P3.4/T0)
15 P1.3 (P4.3/nINT2)
14 P1.2 (P4.2/nINT3)
13 P1.1 (AIN1)
12 P1.0 (AIN0)
11 P3.7

3.2. 引脚描述（PDIP-20、SOP-20）

引脚名称	引脚编号	类型	描述
P1.0~P1.7	12~19	I/O	P1：通用内部弱上拉 IO 口。当由“0”写为“1”时，开启时的两个周期为 PMOS 强输出，然后保持弱上拉输出高电平。 P1.0 可作比较器正向输入端 P1.1 可作比较器负向输入端 P1.2 可与 P4.2/INT3 互换 P1.3 可与 P4.3/INT2 互换 P1.4 可与 P3.4/T0 互换 P1.5 可与 P3.5/T1 互换
P3.0~P3.7	2~3,6~9,11	I/O	P1：通用内部弱上拉 IO 口。当由“0”写为“1”时，开启时的两个周期为 PMOS 强输出，然后保持弱上拉输出高电平。 MG87FE/L2051/4051/6051 的 P3 口同样具有特殊功能。 P3.4 可与 P1.4 互换 P3.5 可与 P1.5 互换
RESET	1	I	复位时 RESET 口至少保持两个机器周期的高电平,内置下拉电阻
XTAL1/P4.3	5	I/O	外部晶振输入脚 XTAL1 可作 P4.3/INT2 口 P4.3/INT2 可与 P1.3 互换
XTAL2/P4.2	4	O	外部晶振输出脚 XTAL2 可作 P4.2/INT3 口 P4.2/INT3 可与 P1.2 互换
VDD	20	P	电源脚
VSS	10	G	地线

4. 8051 CPU 功能说明

4.1. CPU 寄存器

PSW：程序状态字

地址 D0H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
CY	AC	F0	RS1	RS0	OV	F1	P

CY：进位标志。有进位/借位时 CY=1，否则 CY=0。

AC：半进位标志。当 D3 位向 D4 位产生进位/借位时 AC=1，否则 AC=0，常用于十进制调整运算中。

F0：用户可设定的标志位 0，可置位/复位，也可供测试。

RS1、RS0：四个通用寄存器组的选择位，该两位的四种组合状态用来选择 0~3 寄存器组。见下表：

RS1	RS0	工作寄存器组
0	0	0 组（00-07H）
0	1	1 组（08-0FH）
1	0	2 组（10-17H）
1	1	3 组（18-1FH）

OV：溢出标志。当带符号数运算结果超出 -128~+127 范围时 OV=1，否则 OV=0。当无符号数乘法结果超过 255 时，或当无符号数除法的除数为 0 时，OV=1，否则 OV=0。

F1：用户可设定的标志 1，可置位/复位，也可供测试。

P：奇偶校验标志。每条指令执行完，若 A 中 1 的个数为奇数时 P=1，否则 P=0，即奇偶校验方式。

SP：堆栈指针

地址 81H，读/写，复位初始值 0000-0111B

7	6	5	4	3	2	1	0
SP[7]	SP[6]	SP[5]	SP[4]	SP[3]	SP[2]	SP[1]	SP[0]

DPL：数据指针低字节

地址 82H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
DPL[7]	DPL[6]	DPL[5]	DPL[4]	DPL[3]	DPL[2]	DPL[1]	DPL[0]

DPH：数据指针高字节

地址 83H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
DPH[7]	DPH[6]	DPH[5]	DPH[4]	DPH[3]	DPH[2]	DPH[1]	DPH[0]

B：B 寄存器

地址 F0H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
B[7]	B[6]	B[5]	B[4]	B[3]	B[2]	B[1]	B[0]

4.2. CPU 时序

最快可以达到一个机器周期执行一条指令。MG87FE/L2051/4051/6051 一些指令只需一个机器周期便可完成，而另一些指令执行完成也只需要 2 个或 4 个机器周期。一个机器周期含 12 个或 6 个时钟周期。如果是 12MHz 的系统时钟，则一个机器周期是 1 微秒或 0.5 微秒。

一个机器周期分为 S1、S2、……S6 六个状态，每个状态又分为两拍，称为 P1 和 P2。每一拍是一个时钟周期。执行一个单机器周期的指令是在锁存操作码的 S1 状态开始，然后在同一机器周期的 S4 状态取操作数，最后在这个机器周期的 S6 状态执行完这条指令。

MOVX 指令在 MG87FE/L2051/4051/6051 芯片中是无效的，因为芯片没有片上外部 RAM 和外部存取总线。

4.3. 寻址方式

直接寻址

在指令中直接给出操作数地址的就属于直接寻址。此时，指令中的操作数部分就是操作数的地址。例如指令：

```
MOV     A,4FH                ;(A)←(4FH)
```

可用于直接寻址的空间是，内部数据 RAM 的低 128 字节及特殊功能寄存器 SFRs。

寄存器间接寻址

由指令中指出某一个寄存器的内容作为操作数的地址。内部 RAM 和外部 RAM 都能通过间接寻址方式进行访问。使用当前工作寄存器组中的 R0 或 R1 存放操作数单元的地址指针（8 位地址），在执行 PUSH（压栈）和 POP（出栈）指令时采用堆栈指针 SP 作寄存间接寻址。而如果地址是 16 位时就只能使用 DPTR 数据指针作间接寻址了。例如指令：

```
MOV     A,@R0                ;(A) ←((R0))
```

寄存器寻址(REG)

寄存器寻址就是以通过寄存器的内容作为操作数。在指令的助记符号中直接以寄存器的名字来表示操作数的地址。例如指令：

```
MOV     A,R0            ;(A) ←(R0)
ADD A,R0                ;(A) ←(Acc)+(R0)
```

能用于这种寻址方式的寄存器还有 ACC、B、DPTR、AB(双字节)和 CY(位累加器)。

变址寻址

以某个寄存器的内容作为基本地址，然后在这个基本地址基础上加上地址偏移量才是真正的操作数地址。例如指令：

```
MOVC    A,@A+DPTR            ;(A) ←((A)+(DPTR))
```

不论用 DPTR 或是 PC 作为基址指针，变址寻址方式都只适用于 8051 的程序存储器，通常用于读取数据表。

立即寻址

指令中地址码部分给出的就是操作数。即取出指令的同时立即得到了操作数。例如指令：

```
MOV     A,#4FH               ;(A) ←4FH
```

相对寻址

相对寻址时，由程序计数器 PC 提供的基地址与指令中提供的偏移量 rel 相加，得到操作数的地址。这时指出的

地址是操作数与现行指令的相对位置。例如指令：

```
SJMP    rel                     ;PC←(PC)+2+rel
```

位寻址

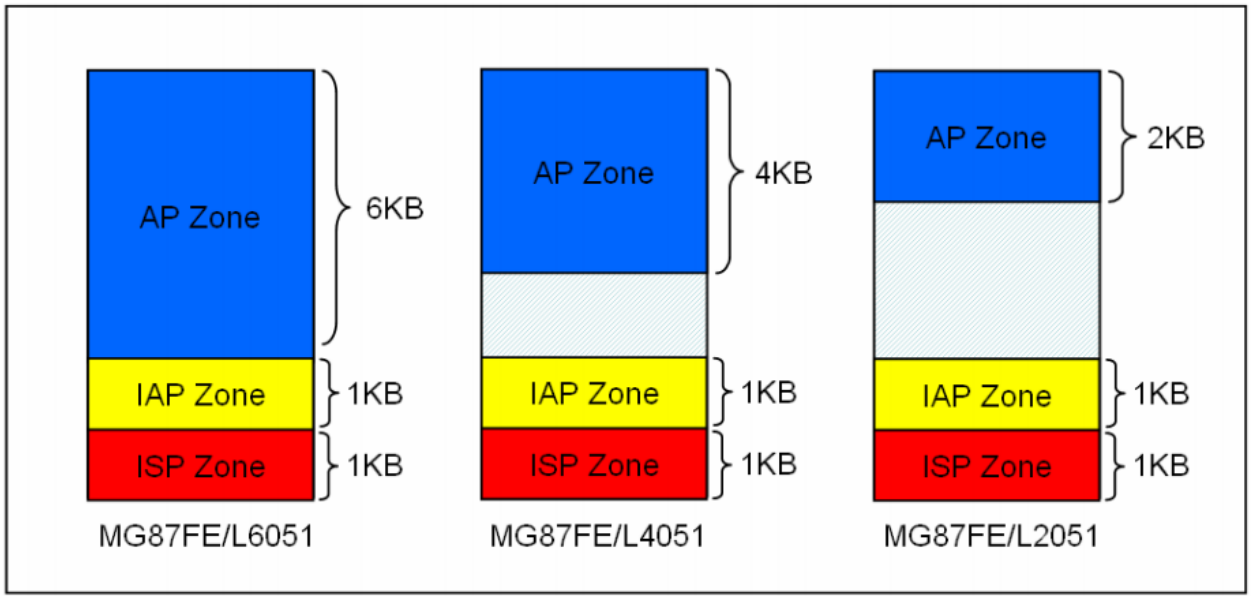
操作数是二进制数的某一位，其位地址出现在指令中，例如指令：

```
SETB    bit                     ;(bit) ←1
```

5. 存储器结构

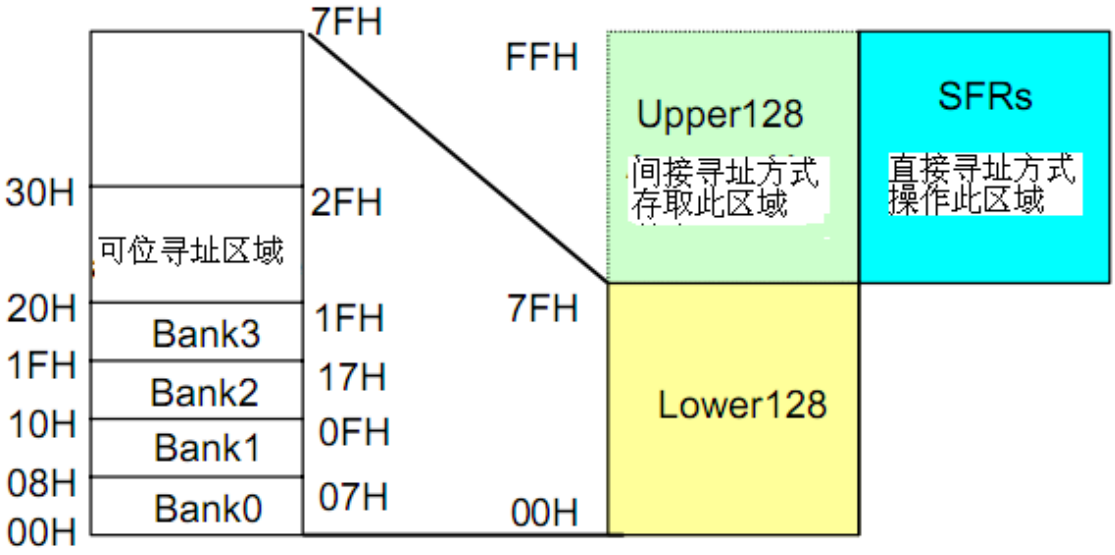
MG87FE/L2051/4051/6051 的程序存储器跟数据存储器是分开编址的。片上数据存储器地址是 8 位的能被 8 位的 CPU 快速存取及操作。MG87FE/L2051/4051/6051 的程序存储器只能读不能写。

5.1. 片上闪存



MG87FE/L2051/4051/6051 闪存分三个区，第一个区是供用户存储应用程序的叫 AP-memory，第二个区是供 CPU 存储用户数据的叫 IAP-memory（相当于 EEPROM），第三个区是专供放置在线编程 ISP 启动程序的叫 ISP-memory。

5.2. 片上数据存储器 RAM



MG87FE/L2051/4051/6051 内部数据存储器映射为三个独立的数据段，分别是低 128 字节，高 128 字节及 128 字节特殊功能寄存器 SFR。

6.2.1 低 128 字节 RAM：（地址 0x00~0x7F）用直接或间接寻址方式存取。

6.2.2 高 128 字节 RAM：（地址 0x80~0xFF）只能用间接寻址方式（使用 R0 或 R1）存取。

6.2.3 特殊功能寄存器 SFR：（地址 0x80~0xFF）只能用直接寻址方式存取。

6. 特殊功能寄存器（SFR）

6.1. SFR 映射图

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8H			CCAP0H 00000000						FFH
F0H	B 00000000								F7H
E8H	P4 XXXX11XX		CCAP0L 00000000						EFH
E0H		WDTCR 0X000000	IFD 11111111	IFADRH 00000000	IFADRL 00000000	IFMT XXXX0000	SCMD XXXXXXXX	ISPCR 0000XXXX	E7H
D8H	CCON 00XXXXXX	CMOD 00000000							DFH
D0H	PSW 00000000						P3WKPE 0X000000	P1WKPE 00000000	D7H
C8H									CFH
C0H	XICON 00000000							CKCON XXXXX000	C7H
B8H	IPL XXX00000	SADEN 00000000						CKCON2 XX001010	BFH
B0H	P3 11111111							IPH 00X00000	B7H
A8H	IE 0XX00000	SADDR 00000000							AFH
A0H			AUXR1 00000000						A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111			TSTWD 0X000000				ACSR 0XX00000	97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR 00000000	CKCON3 XXXXXX0X	8FH
80H		SP 00000111	DPL 00000000	DPH 00000000				PCON 00010000	87H
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	

6.2. SFR 各位分配

SYMBOL	DESCRIPTION	ADDRESS	BIT ADDRESS AND SYMBOL							LSB	INITIAL VALUE
SP	Stack Pointer	81H									00000111B
DPL	Data Pointer Low	82H									00000000B
DPH	Data Pointer High	83H									00000000B
PCON	Power Control	87H	SMOD	SMOD0	PWMEN	POF	GF1	GF0	PD	IDL	00010000B
TCON	Timer Control	88H	9FH	9EH	9DH	9CH	9BH	9AH	89H	88H	00000000B
			TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	
TMOD	Timer Mode	89H	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00000000B
TL0	Timer Low 0	8AH									00000000B
TL1	Timer Low 1	8BH									00000000B
TH0	Timer High 0	8CH									00000000B
TH1	Timer High 1	8DH									00000000B
AUXR	Auxiliary	8EH	INT3H	INT2H	P15FS	P14FS	P13FS	P12FS	P11PU	P10PU	00000000B
CKCON3	Clock Control 3	8FH	-	-	-	-	-	-	PWDEX	EN6TR	xxxxxx0xB
P1	Port 1	90H	97H	96H	95H	94H	93H	92H	91H	90H	11111111B
ACSR	Analog Comp. Reg.	97H	ACIDX	-	-	ACF	ACEN	ACM2	ACM1	ACM0	0xx00000B
SCON	Serial Control	98H	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	00000000B
			SM0 /FE	SM1	SM2	REN	TB8	RB8	TI	RI	
SBUF	Serial Buffer	99H									xxxxxxxxB
AUXR1	Auxiliary 1	A2H	P14FD	*	*	*	GF2	*	*	*	0**0000000**B
IE	Interrupt Enable	A8H	AFH	AEH	ADH	ACH	ABH	AAH	A9H	A8H	0xx00000B
			EA	EAC	-	ES	ET1	EX1	ET0	EX0	
SADDR	Slave Address	A9H									00000000B
P3	Port 3	B0H	B7H	B6H	B5H	B4H	B3H	B2H	B1H	B0H	1x111111B
					T1	T0	INT1	INT0	TXD	RXD	
IPH	Interrupt Priority High	B7H	PX3H/ PTCH	PX2H/ PACH	-	PSH	PT1H	PX1H	PT0H	PX0H	00x00000B
IPL	Interrupt Priority Low	B8H	BFH	BEH	BDH	BCH	BBH	BAH	B9H	B8H	xxx00000B
			-	PAC	-	PS	PT1	PX1	PT0	PX0	
SADEN	Slave Address Mask	B9H									00000000B
CKCON2	Clock Control 2	BFH	OSCDR	EN6TR	XCKS5	XCKS4	XCKS3	XCKS2	XCKS1	XCKS0	xx001010B
XICON	Ext. Interrupt Control	C0H	C7H	C6H	C5H	C4H	C3H	C2H	C1H	C0H	00000000B
			PX3/ PTC	EX3	IE3	IT3	PX2	EX2	IE2	IT2	
CKCON	Clock Control	C7H	-	-	-	-	-	SCKS2	SCKS1	SCKS0	xxxxx000B
PSW	Program Status Word	D0H	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H	00000000B
			CY	AC	F0	RS1	RS0	OV	-	P	
P3WKPE	P3 Wakeup Enable	D6H	P37WE	-	P35WE	P34WE	P33WE	P32WE	P31WE	P30WE	0x000000B
P1WKPE	P1 Wakeup Enable	D7H	P17WE	P16WE	P15WE	P14WE	P13WE	P12WE	P11WE	P10WE	00000000B
CCON	Counter Control Reg.	D8H	CF	CR	-	-	-	-	-	-	00xxxxxxB
CMOD	Counter Mode Reg.	D9H	CIDL	POS2	POS1	POS0	CPS2	CPS1	CPS0	ECF	00000000B
WDTCR	Watch-dog-timer	E1H	WRF	-	ENW	CLW	WIDL	PS2	PS1	PS0	0x000000B
	Control register										
IFD	ISP Flash data	E2H									11111111B
IFADRH	ISP Flash address High	E3H									00000000B
IFADRL	ISP Flash Address Low	E4H									00000000B
IFMT	ISP Mode Table	E5H	-	-	-	-	MS3	MS2	MS1	MS0	xxxx0000B
IAPLB	IAP Low Boundary	Note 1									11111111B
SCMD	ISP Serial Command	E6H									xxxxxxxxB
ISPCR	ISP Control Register	E7H	ISPEN	BS	SRST	CFAIL	-	-	-	-	0000xxxxB
P4	Port 4	E8H	-	-	-	-	EBH	EAH	-	-	xxxx11xxB
CCAP0L		EAH									00000000B
B	B Register	F0H	F7H	F6H	F5H	F4H	F3H	F2H	F1H	F0H	00000000B
CCAP0H		FAH									00000000B

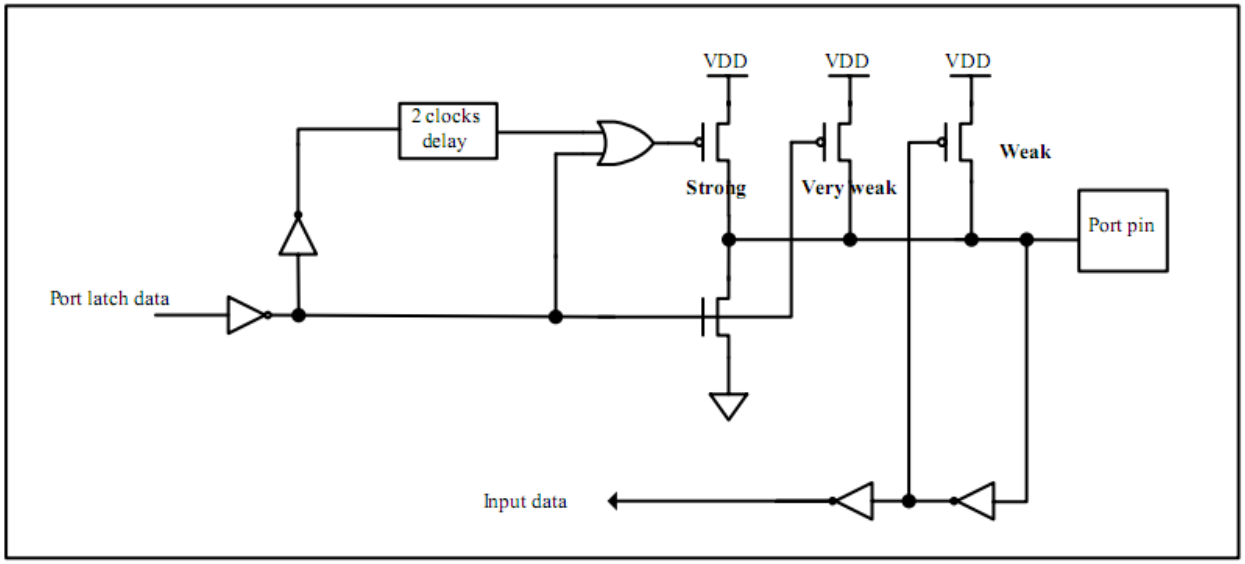
Note1: The registers are addressed by IFMT and SCMD. Please refer the IFMT register description for more detail information.

*: 细节请参考 MG87FE/L6051_4051_2051 V0.10.pdf 应用文件说明 (**建议用户将这些位，保持为 0**)

7. I/O 口

7.1. I/O 结构

7.2. 通用 GPIO 口 P1/3/4 结构



默认状况 P10/P11 是没有使能内部上拉的。

7.3. P1 口寄存器

P1：P1 口寄存器

地址 90H，读/写，复位初始值 1111-1111B

7	6	5	4	3	2	1	0
P17	P16	P15	P14	P13	P12	P11	P10

P17~P10：能被 CPU 置 1 或清 0，或者跟 PWM 定时器绑定在一起作为 PWM 输出。

7.4. P3 口寄存器

P3：P3 口寄存器

地址 B0H，读/写，复位初始值 1X11-1111B

7	6	5	4	3	2	1	0
P37	P36	P35	P34	P33	P32	P31	P30

P37,P35~P30 只能被 CPU 置 1 或清 0，CPU 读到的 P36 口仅仅是模拟比较器的输出。

7.5. P4 口寄存器

P4：P4 口寄存器

地址 E8H，读/写，复位初始值 XXXX-11XXB

7	6	5	4	3	2	1	0
-	-	-	-	P43	P42	-	-

P43 ~P42 只能被 CPU 置 1 或清 0。

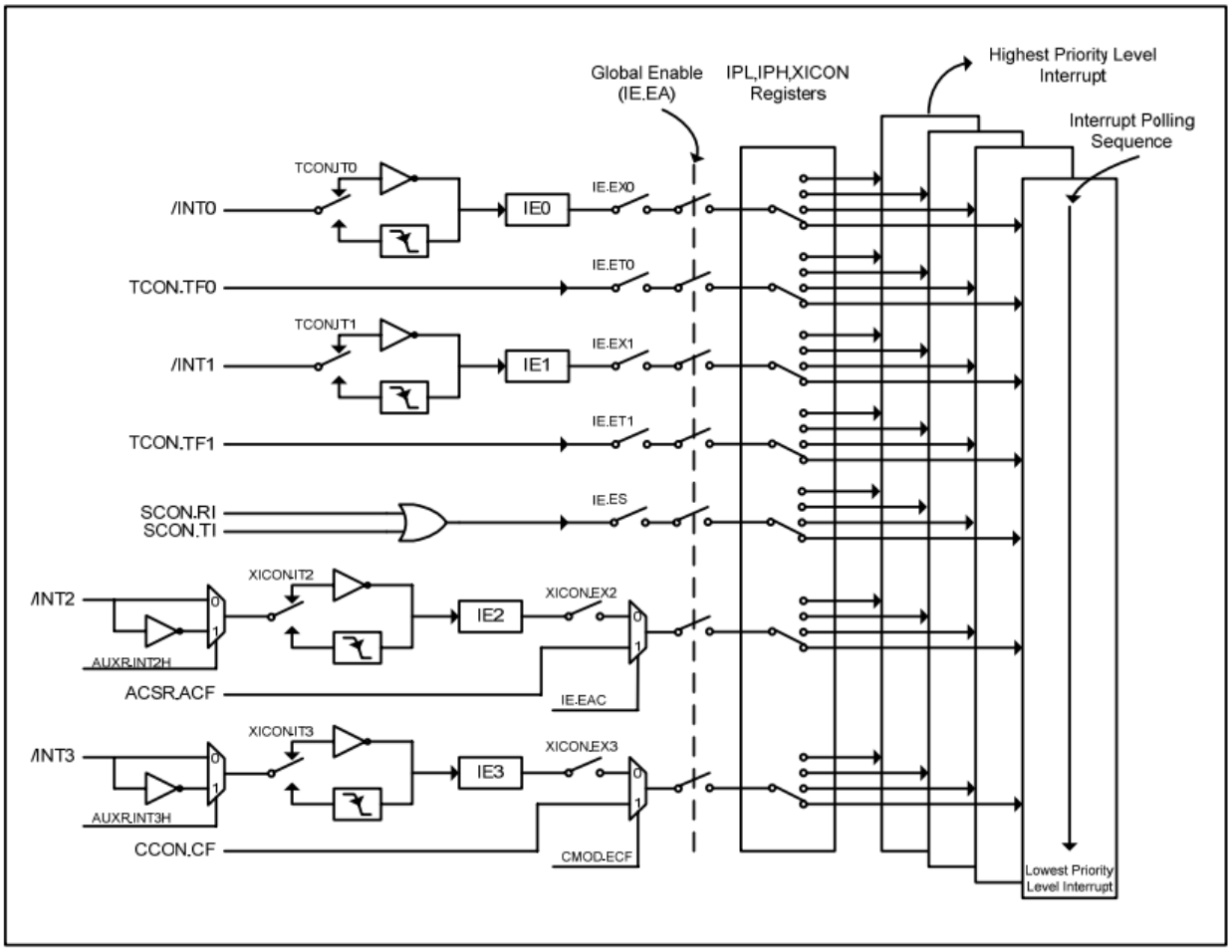
7.6. GPIO 示例代码

(1). 功能需求: 设置 P1.0

汇编语言代码范例:	
SETB P1.0	; 设置 P1.0 为“1”
C 语言代码范例:	
P10 = 1;	//设置 P1.0 为“1”

8. 中断

8.1. 中断结构



8.2. 中断寄存器

IE：中断使能寄存器

地址 A8H，读/写，复位初始值 00X0-0000B

7	6	5	4	3	2	1	0
EA	EAC	--	ES	ET1	EX1	ET0	EX0

EA: 全局中断使能位

0: 禁止所有中断

1: 使能中断

EAC:模拟比较器中断使能位

0: 禁止模拟比较器中断，外部中断 2（INT2）使用中断向量（33H）

1: 使能模拟比较器中断，模拟比较器占用中断 2（INT2）中断向量（33H），这时 IE2 仍然保持它原有的功能但不管 EX2 被置位与否都不产生中断。

ES: 串口中断使能位

0: 禁止

1: 使能

ET1: 定时器 1 中断使能位

0: 禁止

1: 使能

EX1: 外部中断使能位

0: 禁止

1: 使能

ET0: 定时器 0 中断使能位

0: 禁止

1: 使能

EX0: 外部中断 0 使能位

0: 禁止

1: 使能

XICON：外部中断控制寄存器

地址 C0H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
PX3/PTC	EX3	IE3	IT3	PX2	EX2	IE2	IT2

PX3: 外部中断 3 优先级低位，与 IPH 中的 PX3H 一起使用。当 ECF=1 时为 PTC 功能，ECF=0 时为 PX3 功能

PTC: PWM 定时器中断优先级低位，与 IPH 中的 PTCH 一起使用。当 ECF=1 时为 PTC 功能，ECF=0 时为 PX3 功能

EX3: 外部中断 3 使能位

0: 禁止

1: 使能，当 CMOD.ECF 被使能时此功能必须被设定

IE3: 外部中断 3 中断标志位。外部中断 3 边沿触发时，有硬件置位，中断处理时由硬件清零

IT3: 外部中断 3 触发类型控制位

0: 低电平触发。如果 AUXR.INT3H=1 则为高电平触发

1: 下降沿触发。如果 AUXR.INT3H=1 则为上升沿触发

PX2: 外部中断 2 优先级低位,与 IPH 中的 PX2H 一起使用

EX2: 外部中断 2 使能位

0: 禁止

1: 使能

IE2: 外部中断 2 中断标志位。外部中断 2 边沿触发时，有硬件置位，中断处理时由硬件清零

IT2: 外部中断 2 触发类型控制位

0: 低电平触发。如果 AUXR.INT2H=1 则为高电平触发

1: 下降沿触发。如果 AUXR.INT2H=1 则为上升沿触发

IPL：中断优先级低位

地址 B8H，读/写，复位初始值 X0X0-0000B

7	6	5	4	3	2	1	0
-	PAC	--	PS	PT1	PX1	PT0	PX0

IPH：中断优先级高位

地址 B7H，读/写，复位初始值 00X0-0000B

7	6	5	4	3	2	1	0
PX3H/PTCH	PX2H/PACH	--	PSH	PT1H	PX1H	PT0H	PX0H

IPL(或 XICON)和 IPH 一起决定了 4 级优先级，见下表

IPH.x, IPL.x/XICON.x	优先级
1,1	1(最高)
1,0	2
0,1	3
0,0	4

MG87FE/L2051/4051/6051 提供了 8 个中断源。每个中断源都可以通过特殊寄存器 **IE/XICON** 中的位来使能和禁止。该寄存器有一个 **EA** 位，清零它可以立刻禁止所有中断。

每个中断源都有两个对应的位来定义它的优先级。一个在 **IPH**，另一个在 **IPL/XICON**。处理高优先级中断时，不会响应低优先级的中断请求。如果两个不同优先级的中断同时发出请求，高优先级的中断请求将会被响应。如果相同优先级的中断同时发出请求，则由内部优先级来决定哪个中断会被响应。下表说明了内部优先级和中断向量地址。

中断源	中断向量地址	内部优先级
外部中断 0	03H	1(最高)
定时器 0	0BH	2
外部中断 1	13H	3
定时器 1	1BH	4
串口	23H	5
--------	2BH	-
外部中断 2/比较器	33H	6
外部中断 3/PWM 定时器	3BH	7

外部中断**/INT0**、**/INT1**、**/INT2** 和**/INT3** 分别通过 **TCON** 的 **IT0**、**IT1**、**XICON 的 IT2/ IT3** 可以设置成电平触发或边沿触发。实际产生的中断标志位是 **TCON** 的 **IE0**，**IE1**,**XICON** 的 **IE2** 和 **IE3**。产生外部中断时，如果是边沿触发，进入中断服务程序后由硬件清除中断标志位，如果中断是电平触发，由外部请求源而不是由片内硬件控制请求标志。

定时 0 和定时器 1 中断由 TF0 和 TF1(分别由各自的定时/计数寄存器控制，定时器 0 工作在模式 3 时除外)产生。当产生定时器中断时，进入中断服务程序后由片内硬件清除标志位。

串口中断由 RI 和 TI 的逻辑或产生。进入中断服务程序后，这些标志均不能被硬件清除。实际上，中断服务程序通常需要确定是由 R I 还是 TI 产生的中断，然后由软件清除中断标志。

外部中断 2（INT2）与模拟比较器共享同一个中断向量 33H。IE.EAC=1 时中断向量 33H 为模拟比较器使用，且进入中断时 ACSR.ACF 月 IE2 不会被清 0。IE.EAC=0 时中断向量 33H 为外部中断 2（INT2）使用，当 EX2 使能并进入中断时 XICON.IE2 被清 0。

外部中断 3（INT3）与 PWM 定时器共享同一个中断向量 3BH。CMOD.ECF=1 时中断向量 3BH 为 PWM 定时器使用，且进入中断时 CCON.CF 与 IE3 不会被清 0。CMOD.ECF=0 时中断向量 3BH 为外部中断 3（INT3）使用，当 EX3 使能并进入中断时 XICON.IE3 被清 0。

所有这些产生中断的位都可通过软件置位或清零,与通过硬件置位或清零的效果相同。简而言之，中断可由软件产生，推迟或取消。

硬件如何响应中断

每一个机器周期的 S5P2 时间采样中断请求标志，采样被保持到下一个 S5P2 时期。如果一个中断请求标志在第一个 S5P2 时间被置位，在第二个 S5P2 时间（采样周期）将被硬件发现，这时若没有下面任何一个阻止条件则中断系统将产生一个硬件调用 LCALL 转去执行相应的中断服务程序。

阻止条件如下：

1. CPU 正在处理同级或更高级的中断；
2. 现行机器周期不是所执行的最后一个机器周期；
3. 正在执行的是 RETI 或是访问 IE 或 IP 的指令。

上叙任何一个条件都会阻止硬件去响应发生的中断请求。条件 2 保证了在进入其它中断服务程序前执行完当前指令。条件 3 保证了执行完 RETI 指令或是访问 IE 或 IP 的指令后在进入其它中断向量前能执行至少一条或更多条的指令。

每个机器周期硬件都会重复查询中断请求标志，查询到的值都是前一个 S5P2 时间的状态。值得注意的是如果一个被触发的中断标志由于上面的阻止条件发生了而没得到响应，并且此标志没有持续到阻止条件消失，这样的中断请求将得不到响应。也就是说，中断处理本身不能锁存中断，譬如外部电平中断若在电平出现时被屏蔽，而在中断识别之前电平消失，它被完全忽略。也就是每个查询都是新状态。

9. 定时器/计数器

MG87FE/L2051/4051/6051 提供了两个定时器/计数器 T0 和 T1。每个都可以用来作为定时器或计数器。

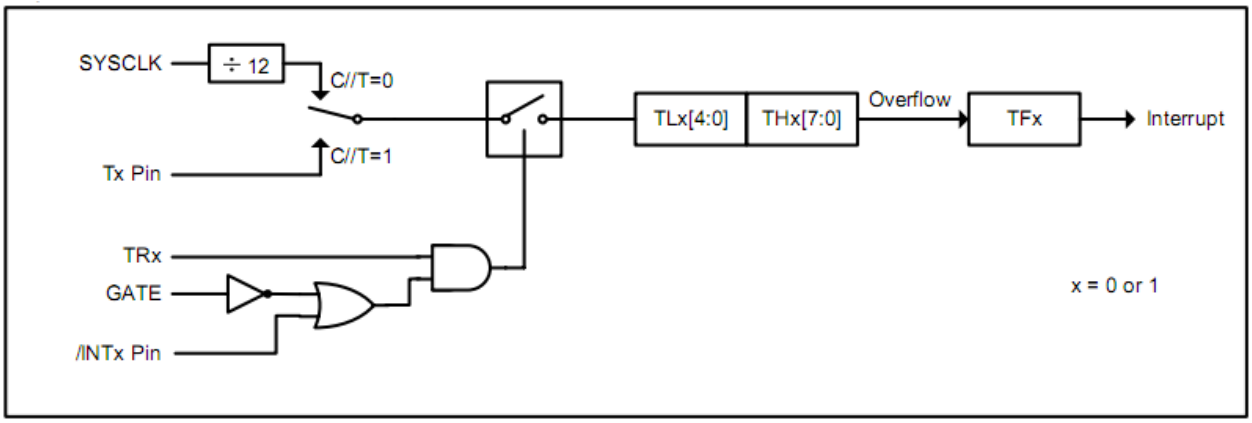
当 T0/T1 用作定时器时，用来触发定时器的时间单位是机器周期。一个机器周期等于 12 或 6 个振荡周期，这取决于用户配置这个芯片是 12T 模式，还是 6T 模式.

当 T0/T1 用作计数器时，计数事件是 T0/T1 对应引脚的“高到低的电平变化”。在这个模式，芯片每个机器周期对 T0/T1 引脚进行采样。一旦结果是从 1 到 0，芯片就对计数器计 1。要注意的是：作为计数器操作时，T0/T1 引脚的高脉冲或者低脉冲的宽度必须大于一个机器周期。

9.1. Timer0 与 Timer1

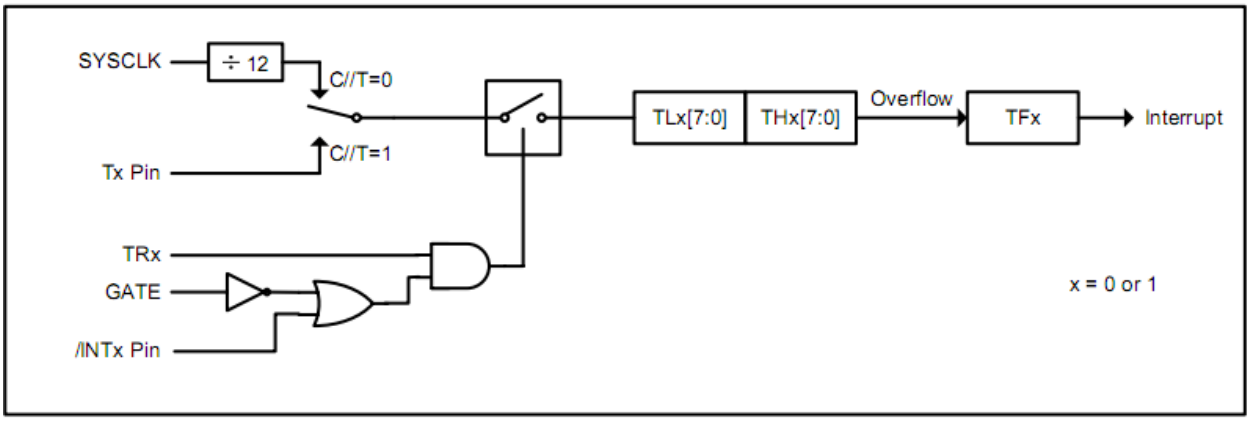
9.1.1. 模式 0

Timer 的寄存器被定义成 13 位寄存器。当寄存器由全 1 变成全 0 时(当寄存器上溢时)，Timer 的中断标志位 TFx 将被置 1。当 TRx=1 并且 GATE=0 或者 INTx=1 时，Timer 被使能。T0 和 T1 的模式 0 操作是一样的。



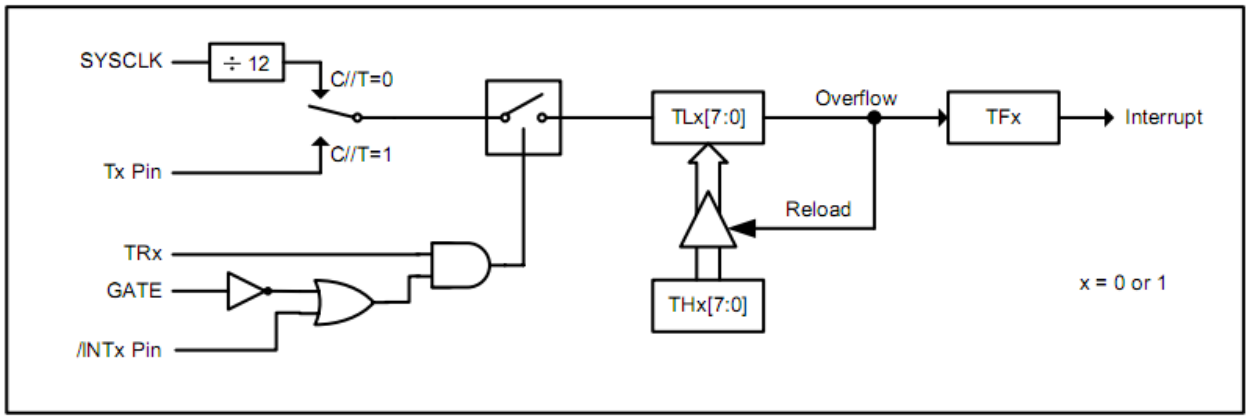
9.1.2. 模式 1

模式 1 与模式 0 相比，除了是 16 位外，其它的都是相同的。



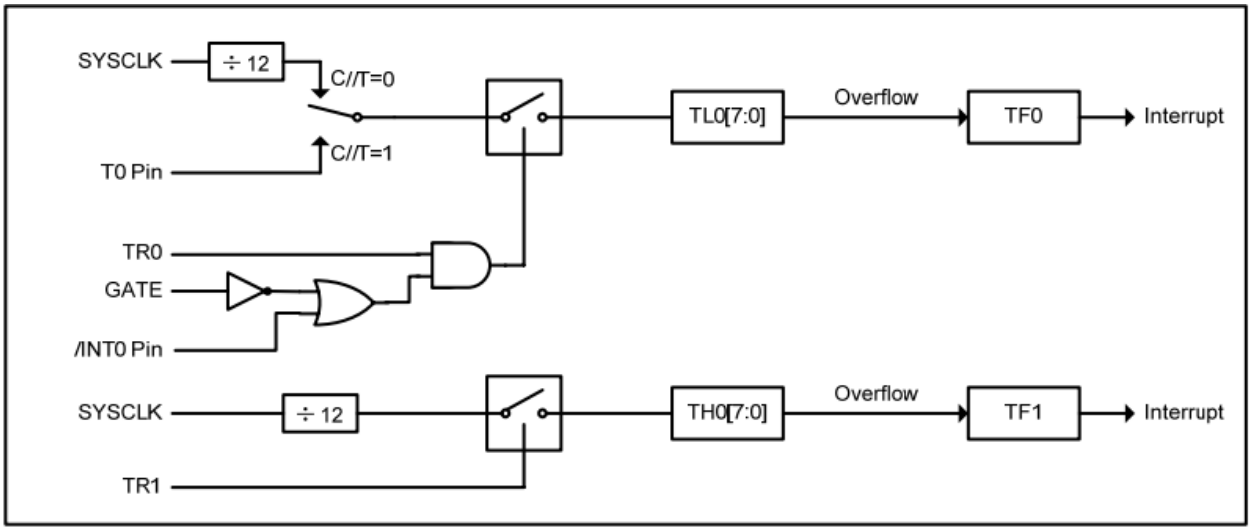
9.1.3. 模式 2

定时器被设置为可自动重载的 8 位计数器，当 TLx 上溢时，在置 TFx 位的同时，自动将 THx 的值重载到 TLx 中，THx 的值不发生改变。定时器 0 和定时 1 在模式 2 的操作是相同的。



9.1.4. 模式 3

定时器 1 在模式 3 下停止工作。定时器 0 在模式 3 下被设置为 2 个独立的 8 位计数器，TL0 可作定时器/计数器，占用定时器 0 的控制位如 C/T，GATE，TR0，INT0 及 TF0。TH0 只能作定时器用，占用定时器 1 的资源 TR1 和 TF1，TH0 控制着定时器的中断。



9.1.5. Timer0/1 寄存器

TMOD：定时器/计数器模式控制寄存器

地址 89H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
GATE	C/T	M1	M0	GATE	C/T	M1	M0
|←---------- Timer1 ----------→				|←---------- Timer0 ----------→			

GATE: 0: 只要 TRx 置 1，Timer x 即使能

1: 必须 TRx 置 1，且/INTx 为高，Timer x 才使能

C/T: 0: 作为定时器

1: 作为计数器

M1,M0 模式选择

0,0: 作为 13 位定时/计数器

0,1: 作为 16 位定时/计数器

1,0: 作为 8 位自动重载定时/计数器，重载值存于 THx

1,1: 对于 T0, TL0 是一个 8 位定时/计数器, TH0 是一个 8 位定时器 T1 被停止

TCON：定时器/寄数器控制寄存器

地址 88H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1: T1 溢出标志位当 T1 溢出时，该位会自动置 1. 当执行 T1 溢出中断时，该位自动清零.

TR1: 0: 停止 T1

1: 开始 T1

TF0: T0 溢出标志位当 T0 溢出时，该位会自动置 1. 当执行 T0 溢出中断时，该位自动清零.

TR0: 0: 停止 T0

1: 开始 T0

IE1: 外部中断 1 标志当外部中断 1 产生时，该位会自动置 1. 当执行外部中断 1 时，该位自动清零.

IT1: 0: 引脚 EX1 低电平，产生外部中断 0

1: 引脚 EX1 下降沿，产生外部中断 0

IE0: 外部中断 0 标志当外部中断 0 产生时，该位会自动置 1. 当执行外部中断 0 时，该位自动清零.

IT0: 0: 引脚 EX0 低电平，产生外部中断 0

1: 引脚 EX0 下降沿，产生外部中断 0

9.2. 定时器 0/1 示例代码

(1). 功能需求: 定时器 T0 以 10KHz 的频率唤醒空闲模式，SYSCLK = 12MHz 晶振

汇编语言代码范例:

```
T0M0            EQU     01h
T0M1            EQU     02h
PT0     EQU     02h
PT0H            EQU     02h
IDL     EQU     01h

    ORG     0000h
    JMP     main

    ORG     0000Bh
time0_isr:
    to do…
    RETI

main:                                                   ;
    MOV     TH0,#(256-100)              ; 设置定时器 0 溢出率为 = SYSCLK x 100
    MOV     TL0,#(256-100)              ;
    ANL     TMOD,#0F0h                  ; 设置定时器为模式 2
    ORL     TMOD,#T0M1                  ;
    CLR     TF0                                 ; 清定时器 0 标志位

    ORL     IP,#PT0                             ; 选择定时器 0 中断优先级
    ORL     IPH,#PT0H                   ;

    SETB    ET0                                 ; 使能定时器 0 中断
    SETB    EA                                  ; 使能全局中断

    SETB    TR0                                 ; 启动定时器 0 运行

    ORL     PCON,#IDL                   ; 设置 MCU 进入空闲模式
    JMP     $
```

C 语言代码范例:

```
#define     T0M0            0x01
#define     T0M1            0x02
#define     PT0         0x02
#define     PT0H            0x02
#define     IDL         0x01

void time0_isr(void) interrupt 1
```

```
{
    To do…
}
void main(void)
{
    TH0 = TL0 = (256-100);                  //设置定时器 0 溢出率为  = SYSCLK x 100
    TMOD &= 0xF0;                           // S 设置定时器为模式 2
    TMOD |= T0M1;
    TF0 = 0;                                //清定时器 0 标志位

    IP |= PT0;                              //选择定时器 0 中断优先级
    IPH |= PT0H;
    ET0 = 1;                                //使能定时器 0 中断
    EA = 1;                                 //使能全局中断

    TR0 = 1;                                //启动定时器 0 运行
    PCON =IDL;                              //设置 MCU 进入空闲模式
    while(1);
}
```

(2). 功能需求: 选择定时器 0 时钟源为 SYSCLK (使能 T0X12)

汇编语言代码范例:

```
T0M0            EQU     01h
T0M1            EQU     02h
PT0         EQU         02h
PT0H            EQU     02h
T0X12           EQU     80h

    ORG     0000h
    JMP     main

    ORG     0000Bh
time0_isr:
    to do…
    RETI

main:
    ORL     AUXR, #T0X12                ; 选择定时器 0 时钟源为 SYSCLK
    CLR     TF0                         ; 清定时器 0 标志位

    ORL     IP,#PT0                     ; 选择定时器 0 中断优先级
    ORL     IPH,#PT0H                   ;

    SETB    ET0                         ; 使能定时器 0 中断
    SETB    EA                          ; 使能全局中断

    MOV     TH0, #(256 - 240)           ;中断间隔 20us
    MOV     TL0, #(256 - 240)           ;

    ANL     TMOD,#0F0h                  ; 设置定时器 0 为模式 2
    ORL     TMOD,#T0M1                  ;


    SETB    TR0                         ; 启动定时器 0
    JMP     $
```

C 语言代码范例:

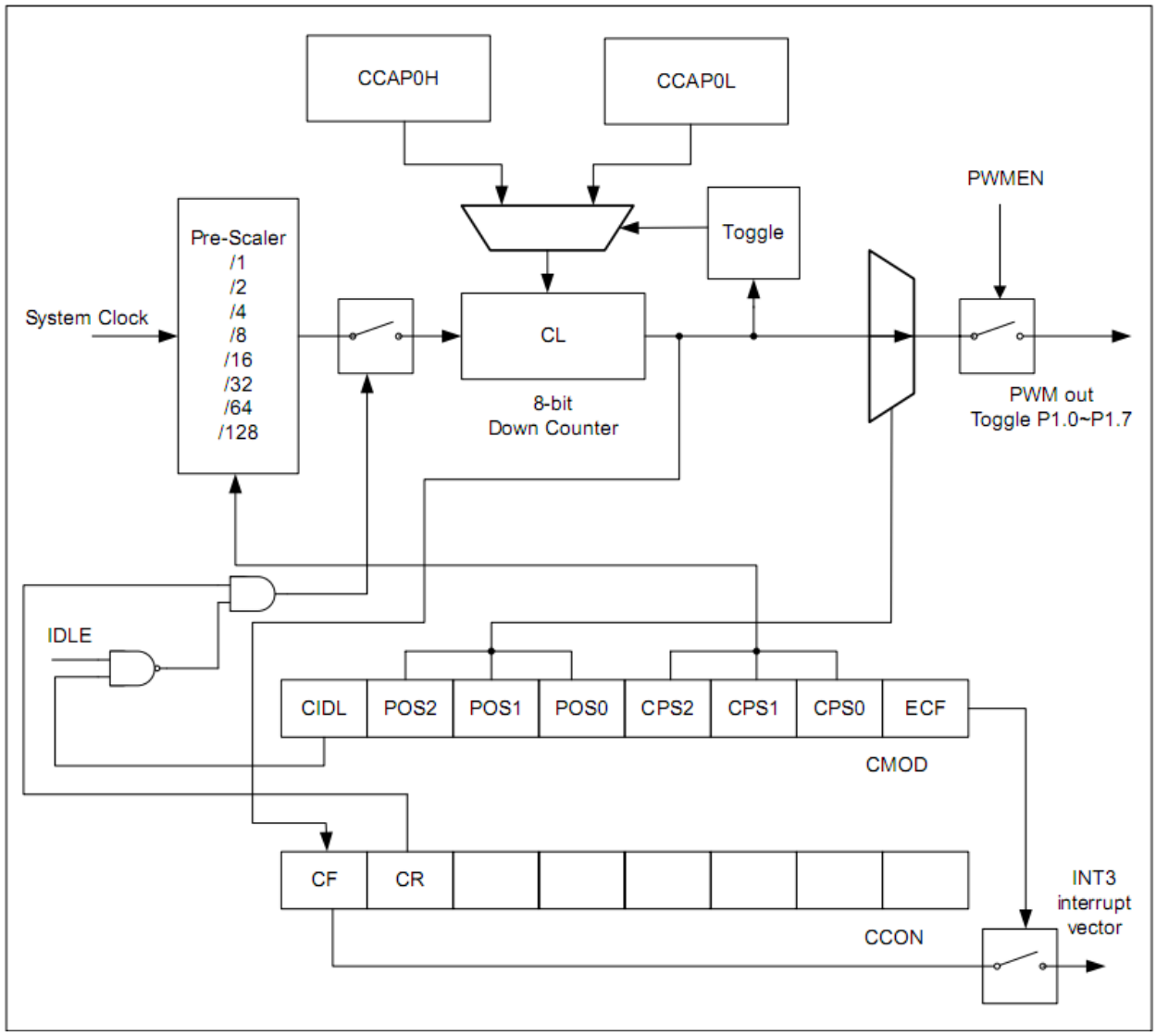
```
#define     T0M0            0x01
#define     T0M1            0x02
#define     PT0         0x02
#define     PT0H            0x02
#define     T0X12           0x80

    AUXR |= T0X12                       //选择定时器 0 时钟源为 SYSCLK
```

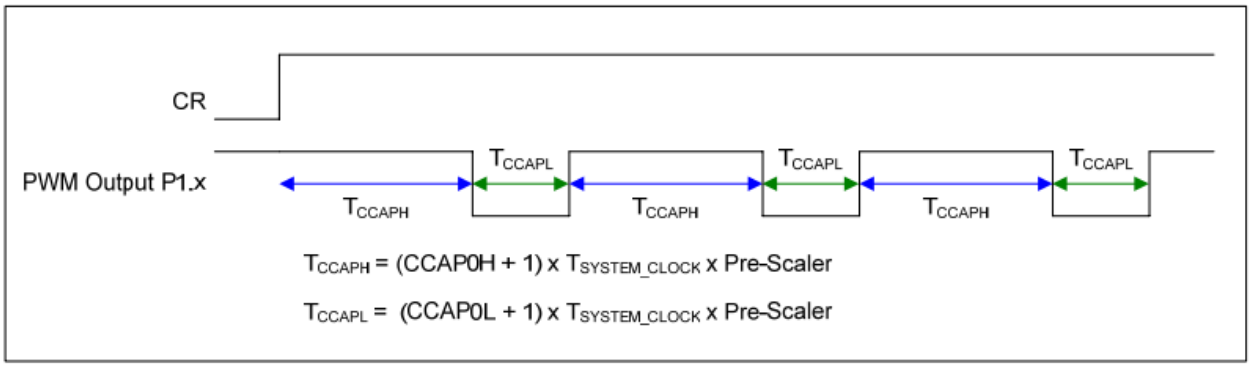
```
TF0 = 0;

IP |= PT0;                                  //选择定时器 0 中断优先级
IPH |= PT0H;

ET0 = 1;                                    //使能定时器 0 中断
EA = 1;                                     //使能全局中断

TH0 = TL0 = (256 - 240);

TMOD &= 0xF0;                               //设置定时器 0 为模式 2
TMOD |= T0M1;

TR0 = 1;                                    //启动定时器 0
```

9.3. PWM-Timer（PWM 定时器）

一个为 PWM 发生器设计的专用 8 位定时器。

9.3.1. PWM 定时器结构





9.3.2. PWM 定时器寄存器

CMOD：PWM 定时器模式寄存器

地址 D9H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
CIDL	POS2	POS1	POS0	CPS2	CPS1	CPS0	ECF

CIDL: CPU 空闲模式（IDLE）计数控制

0：IDLE 时 PWM 定时器继续计数

1：IDLE 时 PWM 定时器停止计数

POS[2:0]: PWM 输出口选择

POS[2:0]	PWMEN	PWM输出口
0 0 0	1	P1.0
0 0 1	1	P1.1
0 1 0	1	P1.2
0 1 1	1	P1.3
1 0 0	1	P1.4
1 0 1	1	P1.5
1 1 0	1	P1.6
1 1 1	1	P1.7
X X X	0	Disabled

CPS[2:0]: PWM 计数器分频选择

CPS[2:0]	分 频
0 0 0	1
0 0 1	2
0 1 0	4
0 1 1	8
1 0 0	16
1 0 1	32
1 1 0	64
1 1 1	128

ECF: 使能 PWM 定时器向下溢出时中断

0：禁止 CCON.CF 位产生中断

1：使能 CCON.CF 位产生中断

CCON：PWM 定时器控制寄存器

地址 D8H，读/写，复位初始值 00XX-XXXXB

7	6	5	4	3	2	1	0
CF	CR	-	-	-	-	-	-

CF: PWM 定时器下溢出标志

0： 只能通过软件清 0

1： 当 PWM 计数器向下溢出时硬件置 1。若 EMOD.ECF=1 时，将产生一个中断请求。可以由硬件或

软件置 1。

CR：　PWM 定时器启动控制位

0：停止

1：启动

CACP0L：PWM 占空比低位寄存器

地址 EAH，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
b7	b6	b5	b4	b3	b2	b1	b0

CACP0H：PWM 占空比高位寄存器

地址 FAH，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
b7	b6	b5	b4	b3	b2	b1	b0

9.4. PWM 示例代码

(1). 功能需求: 设置 P1.7 输出占空比 75% 的波形

汇编语言代码范例

```
PWM_P17             EQU         070h
PWM_Pre_scale_DIV_8         EQU         06h
PWMEN     EQU               020h
CF          EQU         080h
CR          EQU         040h

    ORG     0000h
    JMP     main

main:                                           ;

    MOV     CACP0H,#192                     ; 设置占空比为 75%
    MOV     CACP0L,#(255-192)                   ;
    MOV     CMOD,#PWM_P17                   ; 设置 P1.7 输出 PWM
    ORL     CMOD,#PWM_Pre_scale_DIV_8       ; 设置 PWM 时钟源 = 系统时钟 / 8
    ANL     CCON,#(0FFh - CF)                   ;

    ORL     PCON,#PWNEN                     ; 使能 PWM 由 I/O 输出
    ORL     CCON,#CR                        ; 启动  PWM

    JMP     $
```

C 语言代码范例:

```
#define     PWM_P17             0x70
#define     PWM_Pre_scale_DIV_8         0x06
#define     PWMEN               0x20
#define     CF                  0x80
#define     CR                  0x40

void main(void)
{
    CCAP0H = 192;                   //设置占空比为 75%
    CCAP0L = (255-192);                 //
    CMOD |= PWM_P17;                    //设置 P1.7 输出 PWM
    CMOD |= PWM_Pre_scale_DIV_8;        //设置 PWM 时钟源 = 系统时钟 / 8
    CCON &= ~CF;                        //

    PCON |= PWMEN;                      //使能 PWM 由  I/O 输出
    CCON |= CR;                         //启动  PWM

    while(1);
}
```

10. 串口 UART

MG87FE/L2051/4051/6051 的串口支持全双工传输，它可以同时进行发射和接收。串口接收和发射共享相同的特殊功能寄存器 SBUF，但实际上在芯片内部是有两个不同的 SBUF，一个用于发射，另外一个用于接收。串行可以在 4 个不同的模式下工作。

10.1. UART 结构

模式 0

串行数据通过 RXD（P3.0）输入/TXD（P3.1）输出，8 位串行数据均从最低位开始接收/发射。波特率固定为系统时钟的 12 分频。

$$\text{Baud Rate in Mode 0} = \frac{F_{SYSCLK}}{12}$$

模式 1

10 位的串口数据通过 RXD（P3.0）输入/TXD（P3.1）输出，每帧数据包括一个起始位（0）、8 个数据位和一个停止位（1）。对于一次接收，停止位的数据会保存在 SCON 的 RB8 中。波特率是可变的。

$$\text{Baud Rate in Mode 1} = \frac{2^{SMOD}}{32} \times \text{(timer1 overflow rate)}$$

模式 2

11 位的串口数据通过 RXD（P3.0）输入/TXD（P3.1）输出，每帧数据包包括一个起始位（0）、8 个数据位、可编程的第 9 位和一个停止位（1），发射的第 9 个数据存放在 SCON 的 TB8 位中，接收的第 9 个数据存放在 SCON 的 RB8。波特率固定为系统时钟的 32 分频或者 64 分频。

$$\text{Baud Rate in Mode 2} = \frac{2^{SMOD}}{64} \times F_{SYSCLK}$$

模式 3

其工作方式与模式 2 相同，但其波特率是可变的。

$$\text{Baud Rate in Mode 3} = \frac{2^{SMOD}}{32} \times \text{(timer1 overflow rate)}$$

在上面的四种模式中，只要有指令将 SBUF 作为目标操作对象，发射动作就会启动，在模式 0 中，当 RI

为“0”且 REN 为“1”时，接收动作就会被启动，对于其它模式，接收动作只有在 REN 为“1”且检测到起始信号（RXD 发生负跳变）时才会启动。

自动地址识别

自动地址识别功能是通过硬件比较电路让串口确认某些地址的串行位流。这个功能使得软件不需要检查每一个传入的地址。要启用此功能只需要设置 SCON 中的 SM2 位。在模式 2 和模式 3 中，当收到字节包含“Given”地址或“广播”地址时，串口接收中断标志位 RI 将自动置“1”。这两种模式中，要求获得第九位是“1”，表明收到的字节是一个地址，而不是数据。在模式 1 中，如果 SM2 被使能且一个有效的停止位后跟随的 8 位数据是“Given”地址或“广播”地址时，串口接收中断标志位 RI 将自动置“1”。模式 0 中 SM2 无意义。

帧错误检测

对于一帧数据，如果丢失位停止位，SCON 中的 FE 位将被置位。FE 与 SM0 公用 SCON 的第 7 位，当 SMOD0（PCON.6）为“1”时，SCON.7 为 FE 功能，否则为 SM0 功能。当用于 FE 功能时，标志必须由软件清除。

10.2. UART 寄存器

SCON：串口控制寄存器

地址 98H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI

FE： 帧错误位。当检测到一个无效停止位时通过 UART 接收器设置该位，但它必须由软件清零，要使该位有效 PCON 寄存器中的 SMOD0 位必须置 1

SM0： 和 SM1 定义串口操作模式，要使该位有效 PCON 寄存器中的 SMOD0 必须置 0

SM1： 和 SM0 定义串行口操作模式，见下表：

SM0	SM1	UART 模式	波特率
0	0	模式 0, 同步移位寄存器	osc/12
0	1	模式 1 ,8 位 UART	可变,取决于 Timer1 溢出
1	0	模式 2, 9 位 UART	fosc /64 或 fosc /32
1	1	模式 3, 9 位 UART	可变,取决于 Timer1 溢出

SM2： 在模式 2 和 3 中多处理机通信使能位。在模式 2 或 3 中，若 SM2=1 且接收到的第 9 位数据 RB8 是 0，则 RI 接收中断标志不会被激活。在模式 1 中，若 SM2=1 且没有接收到有效的停止位，则 RI 不会被激活。在模式 0 中 SM2 必须是 0。

REN： 允许接收位。由软件置位或清除，REN=1 时允许接收，REN=0 时禁止接收

TB8： 模式 2 和 3 中发送的第 9 位数据，可以按需要由软件置位或清除

RB8： 模式 2 和 3 中已接收的第 9 位数据，在模式 1 中，或 SM2=0 RB8 是已接收的停止位。在模式 0 中 RB8 未用

TI： 发送中断标志。 模式 0 中，在发送完第 8 位数据时由硬件置位。其它模式中，在发送停止位之初由硬件置位。在任何模式中都必须由软件来清除 TI

RI： 接收中断标志。模式 0 中，接收第 8 位结束时由硬件置位。其它模式中在接收停止位的中间时刻由硬件置位.在任何模式(SM2 所述情况除外)必须由软件清除

SBUF：串口发送/接收数据寄存器

地址 99H，读/写，复位初始值 XXXX-XXXXB

7	6	5	4	3	2	1	0
b7	b6	b5	b4	b3	b2	b1	b0

SADDR：地址寄存器

地址 A9H，读/写，复位初始值 0000-0000B

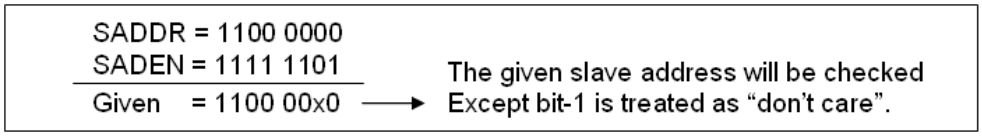
7	6	5	4	3	2	1	0
b7	b6	b5	b4	b3	b2	b1	b0

SADEN：地址屏蔽寄存器

地址 B9H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
b7	b6	b5	b4	b3	b2	b1	b0

当地址自动识别功能启用后，可用 SADDR 和 SADEN 来预置地址，事实上，SADEN 是 SADDR 的“屏蔽”寄存器，如下图所示



每个从对象的广播地址为 SADDR 和 SADEN 进行逻辑“或”的结果，结果中为“0”的位将被忽略。在系统复位后，SADDR 和 SADEN 都被初始化为 0，从而忽略“Given”地址的全部地址位和“广播”地址的全部地址位而导致自动地址识别功能无效。

10.3. 串行口示例代码

(1). 功能需求: 串行口输入 RI 唤醒空闲模式

汇编语言代码范例:

```
PS          EQU     10h
PSH         EQU     10h



    ORG     00023h
uart_ri_idle_isr:
    JB      RI,RI_ISR                   ; 判断是否串行输入中断
    JB          TI,TI_ISR                                   ; 判断是否串行发送中断
RETI                                    ; 中断返回

RI_ISR:
;   Process
    CLR     RI                          ; 清除 RI 标志
    RETI                                ; 中断返回

TI_ISR:
;   Process
    CLR     TI                          ; 清除 TI 标志
    RETI                                ; 中断返回

main:
    CLR     TI                          ; 清除 TI 标志
    CLR     RI                          ; 清除 RI 标志
    SETB    SM1                         ;
    SETB    REN                         ;8 位的模式 2，接收使能

    CALL    UART_Baud_Rate_Setting      ;参考 获得更多信息

    MOV     IP,#PSL                     ; 选择串行口中断优先级
    MOV     IPH,#PSH                    ;

    SETB    ES                          ; 使能串行口中断
    SETB    EA                          ; 使能全局中断

    ORL     PCON,#IDL;                  ; 设置 MCU 进入空闲模式
```

C 语言代码范例:

```
#define     PS          0x10
#define     PSH         0x10
```

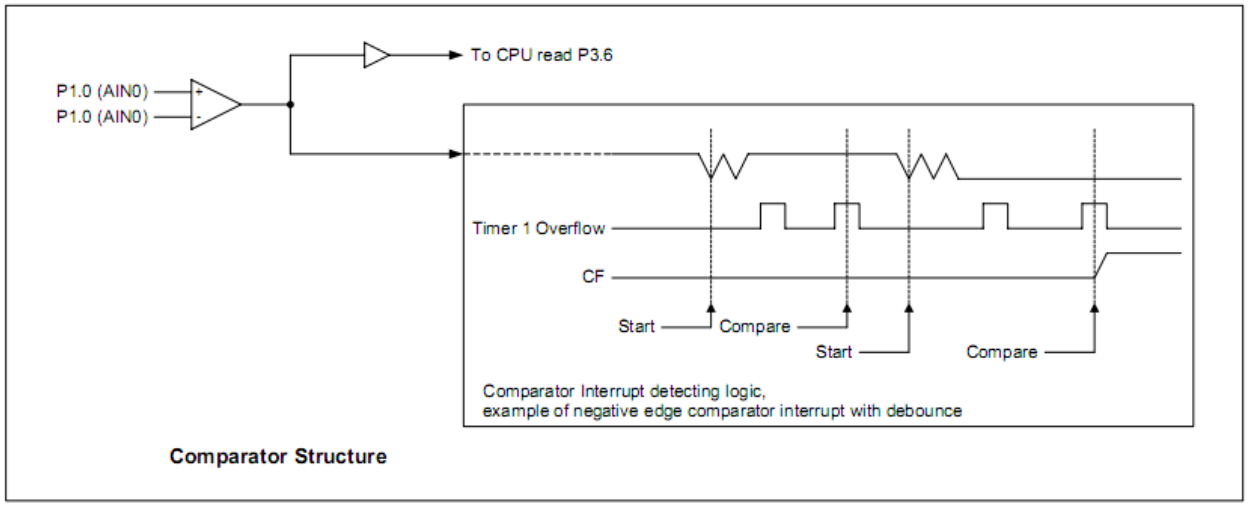
```
void uart_ri_idle_isr(void) interrupt 4
{    if(RI)
     {
          RI=0;
          // to do ...
     }

     if(TI)
     {
          TI=0;
          // to do ...
     }
}

void main(void)
{
     TI = RI = 0;
     SM1 = REN = 1;                                   // 8 位的模式 2，接收使能

     UART_Baud_Rate_Setting()                  //参考 “ 获得更多信息

     IP = PSL;                                          //选择串行口中断优先级
     IPH = PSH;                                        //

     ES = 1;                                             // 使能串行口中断
     EA = 1;                                             //使能全局中断

     PCON |= IDL;                                 //设置 MCU 进入空闲模式
}
```

11. 模拟比较器

MG87FE/L2051/4051/6051 提供用户一个模拟比较器，当正向输入端 AIN0（P1.0）电压大于负向输入端 AIN1（P1.1）时比较器输出逻辑高否则输出逻辑低。ACSR.ACEN 位使能比较器。第一次使能比较器，它的输出在 10 微秒内是不稳定的，所以在这段时间不要使能比较器中断，并且在使能比较器中断前清除比较器中断标志。

通过 ACSR 的 ACM 可以方便的设定比较器的中断模式。当 ACM 所设定的中断条件发生时比较器中断标志 ACSR.ACF 会被置 1，中断标志 ACF 可以被软件查询，可以用来去产生一个中断，但只能用软件清除。模拟比较器在 CPU 空闲模式 IDLE 或掉电模式下无法工作。

11.1. 模拟比较器结构



每个机器周期的 S4 时刻比较器采样输出，这样的比较器非常适合处理慢速的模拟信号。三种延迟模式可以达到滤波的作用。延迟模式下，定时器 1 产生采样时间，当相应的转换事件发生时，比较器再一次采样时需等待两个定时器溢出。如果两次采样的结果相同则把 ACF 置 1，否则忽略它。滤波器可以通过定时器的溢出进行调整。因为定时器 1 一直不停的在运行，而延迟须等到两个溢出以保证采样点至少有一个溢出周期，所以在初始化边沿比较中断事件后，中断将发生在以后的第 1 个或第 2 个溢出周期之间。

11.2. 模拟比较器寄存器

ACSR：模拟比较器控制、状态寄存器

地址 97H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
ACIDX	-	-	ACF	ACEN	ACM2	ACM1	ACM0

ACIDX: CPU 空闲模式（IDLE）比较器开关

0：IDLE 时比较器停止工作

1：IDLE 时比较器继续工作

ACF: 模拟比较器中断标志

0：只能软件清 0

1： ACEN=1 且 ACM[2:0]所设定的比较中断发生时自动置 1。中断能被 IE 的 Bit6[EAC]使能或禁止。

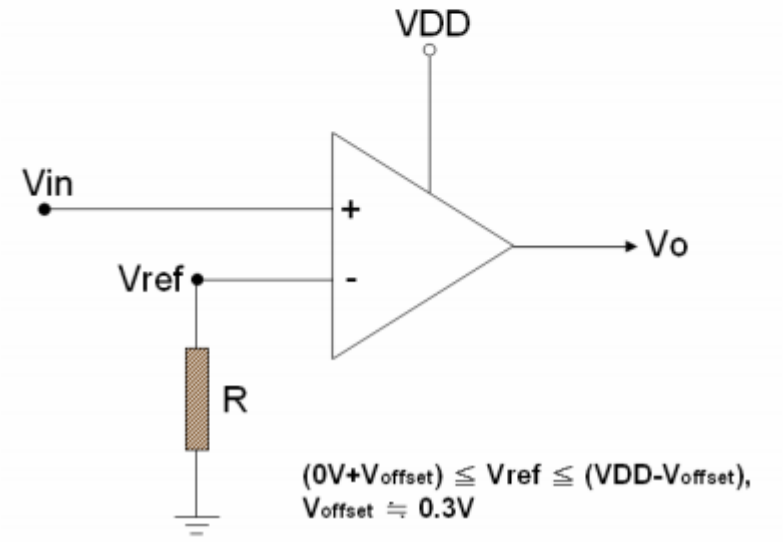
ACEN： 模拟比较器使能

0：禁止。清 0 此位将使比较器强制输出低电平并且防止 ACF 产生中断事件

1：使能。

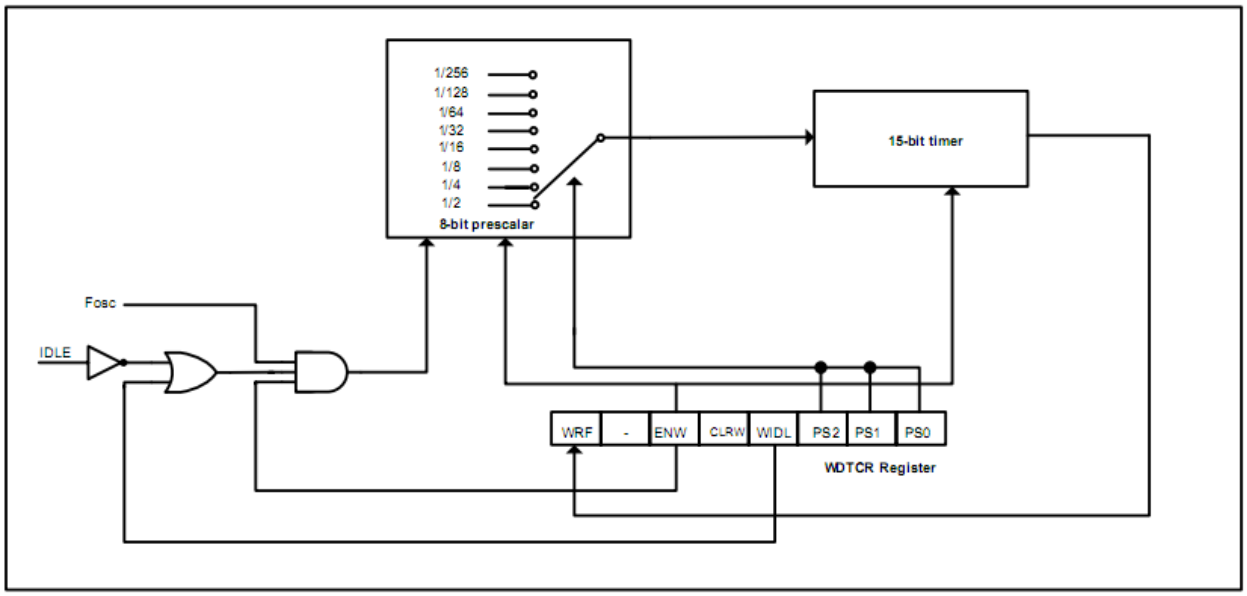
ACM2~ACM0：比较器中断模式选择

ACM[2:0]	Interrupt Mode
0 0 0	Negative (Low) level
0 0 1	Positive edge
0 1 0	Toggle with de-bounce
0 1 1	Positive edge with de-bounce
1 0 0	Negative edge
1 0 1	Toggle
1 1 0	Negative edge with de-bounce
1 1 1	Positive (High) level



12. 看门狗定时器（WDT）

12.1. WDT 结构



12.2. WDT 寄存器

WDTCR：看门狗控制寄存器

地址 E1H，读/写，复位初始值 0X00-0000B

7	6	5	4	3	2	1	0
WRF	-	ENW	CLRW	WIDL	PS2	PS1	PS0

WRF: 当 WDT 溢出时，该位被置 1。 由软件清零

ENW: 看门狗使能位。 上电后，该位由 **HWENW** 设定

1: 使能, (注意: 使能后软件不能关闭)

0: 禁止

CLRW: 对该位置 1， 将清零 WDT 计数器

WIDL: 上电后，该位由 **HWIDL** 设定

0: 在 IDLE 模式下停止 WDT 计数

1: 在 IDLE 模式下继续 WDT 计数

PS2,PS1,PS0： 设置看门狗计数器预分频。上电后，该位由 **HWPS2:0** 设定

PS[2:0]	Pre-scalar Value
0 0 0	2
0 0 1	4
0 1 0	8
0 1 1	16
1 0 0	32
1 0 1	64
1 1 0	128
1 1 1	256

12.3. WDT 示例代码

(1)功能需求: 使能 WDT 并且选择 WDT 预分频为 1/32

汇编语言代码范例:

```
PS0             EQU     01h
PS1             EQU     02h
PS2             EQU     04h
WIDL            EQU     08h
CLRW                    EQU             10h
ENW             EQU     20h
WRF             EQU     80h

    ANL     WDTCR,#(0FFh - WRF)                 ; 清除 WRF 标志(写“0”)
    MOV     WDTCR,#(ENW + CLRW + PS2)   ; 使能 WDT 并且选择 WDT 预分频为 1/32
```

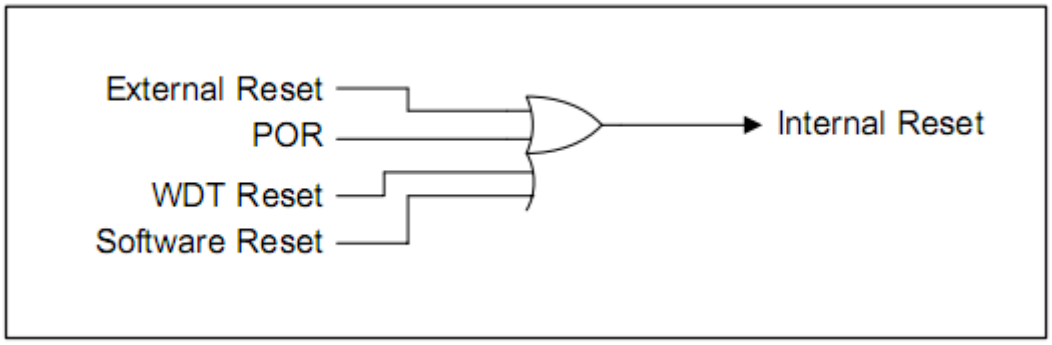
C 语言代码范例:

```
#define     PS0     0x01
#define     PS1     0x02
#define     PS2     0x04
#define     WIDL    0x08
#define     CLRW            0x10
#define     ENW     0x20
#define     WRF     0x80

    WDTCR &= ~WRF;                              //清除 WRF 标志(写“0”)
    WDTCR = (ENW | CLRW | PS2);         //使能 WDT 并且选择 WDT 预分频为 1/32
                                        // PS[2:0]   | WDT  预分频器选项
                                        //     0     | 1/2
                                        //     1     | 1/4
                                        //     2     | 1/8
                                        //     3     | 1/16
                                        //     4     | 1/32
                                        //     5     | 1/64
                                        //     6     | 1/128
                                        //     7     | 1/256
```

13. 复位

复位期间，所有的寄存器被初始为默认值，I/O 端口弱上拉到 VDD，程序从复位入口地址、0000H、或者是硬件选项设定的 ISP 起始地址开始执行。MG87FE/L2051/4051/6051 总共有四种复位源：外部复位、上电复位、WDT 复位及软件复位。

13.1. 复位源



14. 电源管理

MG87FE/L2051/4051/6051 支持两种省电模式：空闲 IDLE 模式和掉电模式，通过 PCON 寄存来控制。

14.1. 省电模式

14.2. 空闲 IDLE 模式

可以通过软件的方式置 PCON.IDL 位，使设备进入空闲模式。在空闲模式下，系统不会给 CPU 提供时钟，CPU 状态、RAM、SP、PC、PSW、ACC 被保护起来。I/O 端口也保持当前逻辑。有两种方式是设备从空闲模式唤醒，首先，将“复位”脚连接到高电平来产生一个内部硬件复位可以唤醒空闲模式中的设备，其次任何处于激活状态的中断源都将会清除 PCON.0 而使设备终止空闲模式，并同时进入中断服务程序，只有在中断返回后才会开始执行进入空闲模式指令之后的程序。空闲模式下定时器 0、定时器 1、PWM 和串口等硬件仍然处于工作状态，模拟比较器将停止运行。

P1.0 和 P1.1 在没有外部上拉的情况下应该清 0，有外部上拉的情况下置 1，或者在使能了 AUXR.P10PU 与 AUXR.P11PU 情况下置 1。

14.3. 掉电模式

可以通过软件的方式置 PCON.PD 位，使设备进入掉电模式。在掉电模式下，振荡器被停止，Flash 存储器掉电以节约电能，只有上电电路继续刷新电源，在减少 VDD 的时候 RAM 的内容仍然会被保持，但特殊功能寄存器 SFR 的内容就不一定能保持住。外部复位、上电复位、外部中断或使能的唤醒口（通用 IO 口 GPIO）能使系统退出掉电模式。

系统复位或刚退出掉电模式后至少要等 4 微秒后才能进入或再次进入掉电模式。

14.4. 中断唤醒

外部中断/INT0（P3.2）、/INT1（P3.3）、/INT2（P4.3）、/INT3（P4.2）四个外部中断可以使系统退出掉电模式，但这些中断必须在进入掉电模式前使能并配置成同一级别。

若/INT2（P4.3）中断向量被模拟比较器占用且使能/INT2 中断（XICON.EX2=1）时，P4.3 输入低电平同样能唤醒系统。同样的若/INT3（P4.2）中断向量被 PWM 占用且使能/INT3 中断（XICON.EX3=1）时，P4.2 输入低电平也同样能唤醒系统。

有两种不同的唤醒方式。CKCON3.PWDEX=0 时，唤醒由内部时钟控制，中断口的下降沿系统退出掉电模式，振荡器开始振荡，内部时钟开始计数，但 CPU 要等到内部时钟计数满时才开始执行指令，计数溢出后，中断服务程序开始工作。为了避免中断被重复触发，中断服务程序在返回前应该被禁止，中断口低电平应保持足够长的时间以等系统稳定。CKCON3.PWDEX=1 时，唤醒由中断控制，与前种方式不同的是 CPU、中断服务程序是在中断口出现上升沿电平后才开始工作。

14.5. 复位唤醒

外部复位脚唤醒有点类似于 PWDEX=0 时的外部中断复位，复位脚 RST 有上升沿电平时系统退出掉电模式，振荡器开始振荡，内部时钟开始计数，但 CPU 要等到内部时钟计数满时才开始执行指令。复位脚 RST 必须保持足够长时间的高电平以保证系统完全复位，复位脚 RST 变成低电平时便开始执行程序。

值得指出的是当 IDLE 模式被硬件复位唤醒时，前两个机器周期（内部复位没有取得控制权）程序正常从进入 IDLE 模式的后一条指令执行。这时内部硬件是禁止访问内部 RAM 的，但访问 I/O 端口没有被禁止，为了保证不可预料的写 I/O 口，在进入 IDLE 指令后不要放置写 I/O 口或外部存储器的指令（最好加两到三个 NOP 指令）。

14.6. 普通 I/O GPIO 口唤醒

如果 P1WKPE 或 P3WKPE 寄存器中设置好了，P1、P3 口都具有唤醒的能力。P3.2/INT3 或 P3.3/INT1 的中断被禁止但使能了 P3WKPE 寄存器相应位，P3.2、P3.3 仍然具有唤醒功能。但是 P4.2/INT3 与 P4.3/INT2 只能使能了中断才能唤醒系统。

GPIO 脚唤醒有点类似于 PWDEX=0 时的外部中断复位，GPIO 脚电平下降沿时系统被唤醒，振荡器开始振荡，内部时钟开始计数，但 CPU 要等到内部时钟计数满时才开始执行指令，计数溢出后，不发生中断，CPU 开始从使其进入掉电模式的下一条指令开始执行程序。也就是说，能唤醒的 GPIO 口只有唤醒功能而不会产生中断。

14.7. 电源控制寄存器

PCON：电源控制寄存器

地址 87H，读/写，复位初始值 000X-0000B

7	6	5	4	3	2	1	0
SMOD	SMOD0	PWMEN	POF	GF1	GF0	PD	IDL

SMOD: 串行口波特率加倍位

0：禁止

1：使能，适用于方式 1、2、3。

PWMEN: PWM 使能位

0: PWM-Timer 作定时器用

1: PWM-Timer 作 PWM，通过 POS[2:0]选择触发输出口

POF: 上电标志

0： 必须由软件清除

1： 仅仅在上电复位时会置“1”，其它复位动作都不会影响该位，这个数据位必须由软件清除。

GF1、GF0：两个通用标志位

PD： 掉电（Power-Down）控制位

0：任何一个退出 Power-down 事件发生时自动清 0

1：置 1 则进入掉电模式

IDL： 空闲模式（IDLE 模式）控制位

0：任何一个退出 IDLE 事件发生时自动清 0

1：置 1 则进入 IDLE 模式

P1WKPE：P1 口唤醒控制寄存器

地址 D7H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
P17WKP	P16WKP	P15WKP	P14WKP	P13WKP	P12WKP	P11WKP	P10WKP

P1nWKP: P1 口唤醒设置位

0：禁止唤醒功能

1：使能唤醒功能，当对应的 IO 口出现下降沿时系统退出 Power-down 与 IDLE 模式

P3WKPE：P3 口唤醒控制寄存器

地址 D6H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
P37WKP	-	P35WKP	P34WKP	P33WKP	P32WKP	P31WKP	P30WKP

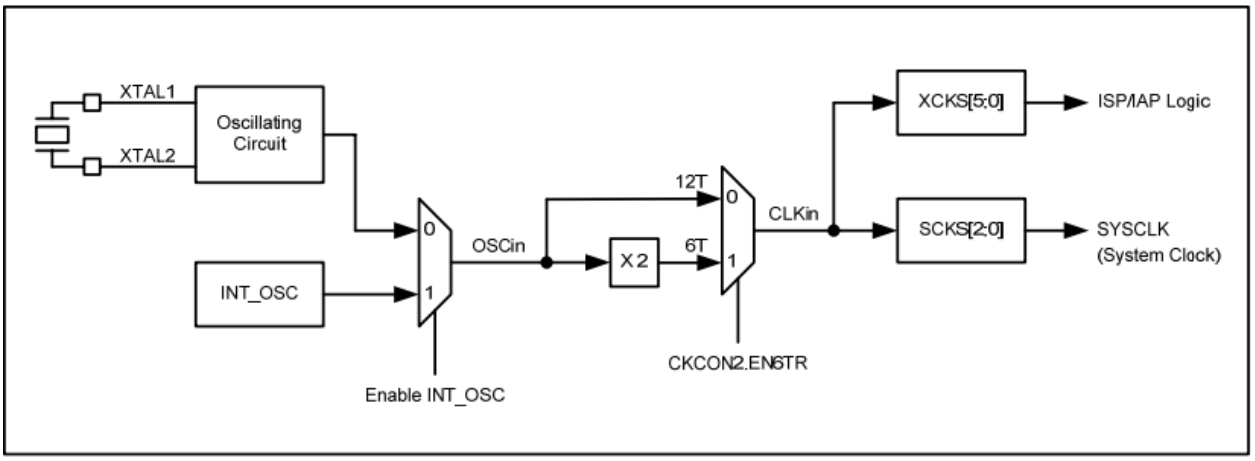
P3nWKP: P3 口唤醒设置位(P3.6 没有唤醒功能)

0：禁止唤醒功能

1：使能唤醒功能，当对应的 IO 口出现下降沿时系统退出 Power-down 与 IDLE 模式

15. 系统时钟

15.1. 时钟结构



15.2. 时钟寄存器

CKCON：时钟控制寄存器

地址 C7H，读/写，复位初始值 XXXX-X000B

7	6	5	4	3	2	1	0
-	-	-	-	-	SCKS2	SCKS1	SCKS0

SCKS2~SCKS0: 可编程系统时钟选择

SCKS[2:0]	系统时钟（F_{SYSCLK}）
0 0 0	CLKin
0 0 1	CLKin /2
0 1 0	CLKin /4
0 1 1	CLKin /8
1 0 0	CLKin /16
1 0 1	CLKin /32
1 1 0	CLKin /64
1 1 1	CLKin /128

CKCON2：时钟控制寄存器 2

地址 BFH，读/写，复位初始值 XX00-1010B

7	6	5	4	3	2	1	0
OSCDR	EN6TR	XCKS5	XCKS4	XCKS3	XCKS2	XCKS1	XCKS0

OSCDR: 晶振驱动位，默认值为硬件选项设置，可通过软件读/写它

0：晶振增益可以足够大到 48MHz

1：减小晶振增益以降低 EMI(建议当方案不需要高频时钟时需使能)

EN6TR：12T/6T 控制位，默认值为硬件选项设置，可通过软件读/写它，读写此位将相应的影响

CKCON3.EN6TR 位，它们的功能一样的。

0：MCU 工作在 12T 模式下

1：MCU 工作在 6T 模式下

XCKS5~XCKS0：为 ISP/IAP 的编程时基设置晶振频率值

XTAL@12T	XTAL@6T	XCKS5~0
1MHz	0.5MHz	000000B
2MHz	1MHz	000001B
3MHz	2MHz	000010B
4MHz	3MHz	000011B
… …	… …	… …
45MHz	22.5MHz	101100B
46MHz	23MHz	101101B
47MHz	23.5MHz	101110B
48MHz	24MHz	101111B

XCKS 的默认值为 001010B(XTAL=11MHz@12T)

CKCON3：时钟控制寄存器 3

地址 8FH，读/写，复位初始值 XXXX-XX0XB

7	6	5	4	3	2	1	0
-	-	-	-	-	-	PWDEX	EN6TR

PWDX: 掉电模式(Power-down)退出方式设置位

0：内部时钟控制

1：外部控制

EN6TR：12T/6T 控制位，默认值为硬件选项设置，可通过软件读/写它，读写此位将相应的影响 CKCON2.EN6TR 位，它们的功能一样的。

0：MCU 工作在 12T 模式下

1：MCU 工作在 6T 模式下

16. 在线编程（ISP）

IFD：ISP/IAP Flash 数据寄存器

地址 E2H，读/写，复位初始值 1111-1111B

7	6	5	4	3	2	1	0
数据							

IFD 为 ISP/IAP 操作的数据寄存器，ISP/IAP 进行读写操作时，IFD 作为数据缓冲区。当用于访问 IAPLB 时，IFD 为 IAPLB 的值

IFADRH：ISP/IAP 地址的高字节

地址 E3H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
高位地址							

IFADRH 存放 ISP/IAP 操作的目标地址的高位

IFADRL：ISP/IAP 地址的低字节

地址 E4H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
低位地址							

IFADRL 存放 ISP/IAP 操作的目标地址的低位，在进行页擦除时，IFADRL 的值被忽略

IFMT：模式表寄存器

地址 E5H，读/写，复位初始值 XXXX-0000B

7	6	5	4	3	2	1	0
保留(00000)					模式选择		

Bit[2:0]	模式
0 0 0	空闲状态
0 0 1	读 Flash 数据
0 1 0	写 Flash 数据
0 1 1	擦除 Flash 数据页
1 0 0	设置 IAPLB
1 0 1	读取 IAPLB

IAPLB：IAP 低边界寄存器

地址为间接地址，读/写，复位初始值 1111-1111B

7	6	5	4	3	2	1	0
IAPLB 数据							

IAPLB 用于定义 IAP 空间的低边界限，由于 Flash 的页面大小为 512 字节，所以 IAPLB 的值必须为偶数

读 IAPLB 的方法：

```
IFMT = 0x05;
ISPCR = 0x80;
```

SCMD = 0x46;

SCMD = 0xB9;

//此时 IFD 中保存的即为 IAPLB 的值

设置 IAPLB 的方法：

IFD = ??; //将 IAPLB 的预设值写入 IFD 中

IFMT = 0x04;

ISPCR = 0x80;

SCMD = 0x46;

SCMD = 0xB9;

IAP 区域由 IAPLB 和 ISP 起始地址共同决定

IAP 低边界 = IAPLB * 256

IAP 高边界 = ISP 起始地址 – 1

例如：如果 IAPLB=0x12，ISP 的起始地址是 0x1C00，则 IAP 存储器的范围位于 0x1200~0x1BFF。

需要注意的是 IAPLB 的值不能大于 ISP 的起始地址

SCMD：ISP 顺序命令寄存器

地址为 E6H，读/写，复位初始值 XXXX-XXXXB

7	6	5	4	3	2	1	0
命令							

ISP/IAP/IAPLB 的操作都需要用 SCMD 寄存器来触发，当 ISPCR.7 为“1”且 SCMD 顺序写入命令“0x46 0xB9”时，ISP 操作被触发。

ISPCR：ISP 控制寄存器

地址为 E7H，读/写，复位初始值 0000-XXXXB

7	6	5	4	3	2	1	0
ISPEN	SWBS	SWRST	CFAIL	-			

ISPEN：ISP/IAP 使能位

0：全局禁用 ISP/IAP 编程/擦除/读功能

1：使能 ISP/IAP 编程/擦除/读功能

SWBS：软件引导选择位

0：复位后从 AP 区域启动

1：复位后从 ISP 区域启动

SWRST：软件复位触发控制位

0：无操作

1：产生软件复位时，启动后硬件自动清除它

CFAIL：ISP/IAP 命令是否执行失败

0：ISP/IAP 操作成功

1：ISP/IAP 操作失败

MG87FE/L2051/4051/6051 不用 IDLE 模式去完成 ISP 操作，而是建立一个 CPU 等待以释放 Flash 存储器供给 ISP 控制电路使用。一旦 ISP 执行完成，CPU 将被唤醒并开始执行调用 ISP 指令的下一条指令，整个 ISP 操作期间中断服务程序被封锁。

ISP 控制电路有一个内部定时器产生时序去控制，可以通过 CKCON2.XCKS[5:0]去设定以得到精确的擦除/编程时序。

17. 应用程序可编程（IAP）

MG87FE/L2051/4051/6051 可编程序存储器（AP-memory）大小被限制为 2K/2051，4K/4051，6K/6051。IAPLB 与 ISP 起始地址之间的 Flash 可以定义为数据闪存 IAP，可供应用程序通过 ISP 操作存取，IAP 可以通过 IAPLB 来改变其大小。

当 MCU 从应用程序区 AP 启动时，应用程序只能存取 IAP 区域而 AP 区域及 ISP 区域因受保护而无法访问。

当 MCU 从 ISP 启动时，AP 区域与 IAP 区域都能被 ISP 完全存取。

17.1. IAP 示例代码

如上所述，所有的ISP模式也可应用于IAP操作，这些模式的示范代码如下显示

触发“页面擦除模式” 的演示代码

```
ORL CKCON2,#011h; CKCON2[5:0]=011h, 假设 MCU 运行在 11.0592MHz
MOV ISPCR,#10000000b ; ISPCR.7=1, 启用 ISP
MOV IFMT,#03h    ; 选择页擦除模式
MOV IFADRH,??    ; 填写 [IFADRH,IFADRL] 为页地址
MOV IFADRL,??    ; ！这个页面必须在 IAP 存储区
MOV    SCMD,#46h          ; 触发的 ISP 处理

MOV SCMD,#0B9h

;现在单片机将会停止运行，直到处理完成
```

触发”编程模式”的演示代码

```
ORL CKCON2,#011h; CKCON2[5:0]=011h, 假设 MCU 运行在 11.0592MHz
MOV ISPCR,#10000000b ; ISPCR.7=1, 启用 ISP
MOV IFMT,#02h    ; 选择 编程 模式

MOV IFADRH,??    ; 填写 [IFADRH,IFADRL] 为字节地址
MOV IFADRL,??    ; ！这个字节地址必须在 IAP 存储区
MOV IFD,??   ; 填写 IFD 的数据进行编程
MOV    SCMD,#46h          ; 触发的 ISP 处理
MOV    SCMD,#0B9h         ;

; 现在单片机将会停止运行，直到处理完成
```

触发”**读模式**”的演示代码

```
ORL CKCON2,#011h; CKCON2[5:0]=011h, 假设 MCU 运行在 11.0592MHz
MOV ISPCR,#10000000b ; ISPCR.7=1, 启用 ISP
MOV IFMT,#01h    ; 选择 读 模式

MOV IFADRH,??    ; 填写 [IFADRH,IFADRL] 为字节地址
MOV IFADRL,??    ; ！这个字节地址必须在 IAP 存储区
MOV     SCMD,#46h          ;  触发的 ISP 处理
MOV     SCMD,#0B9h

; 现在单片机将会停止运行，直到处理完成
MOV A,IFD    ;现在，读出的数据在 IFD 寄存器了
...
...
```

18. 附加的特殊功能寄存器

AUXR：附加控制寄存器

地址为 8EH，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
INT3H	INT2H	P15FS	P14FS	P13FS	P12FS	P11PU	P10PU

INT3H：INT3 触发方式

0：INT3 低电平触发或 P4.2 下降沿触发

1：INT3 高电平触发或 P4.2 上升沿触发

INT2H：INT2 触发方式

0：INT2 低电平触发或 P4.2 下降沿触发

1：INT2 高电平触发或 P4.2 上升沿触发

P15FS：P1.5 交换使能

0：P1.5 与 P3.5 保持原有的功能

1：P1.5 作 P3.5/INT1，P3.5 作 P1.5

P14FS：P1.4 交换使能

0：P1.4 与 P3.4 保持原有的功能

1：P1.4 作 P3.4/INT0，P3.4 作 1.4

P13FS：P1.3 交换使能

0：P1.3 与 P4.3 保持原有的功能

1：P1.3 作 P4.3/INT2，P4.3 作 P1.4（如果 MCU 使用内部振荡器 XTAL1 为 GPIO 时）

P12FS：P1.2 交换使能

0：P1.2 与 P4.2 保持原有的功能

1：P1.2 作 P4.2/INT3，P4.2 作 P1.2（如果 MCU 使用内部振荡器 XTAL2 为 GPIO 时）

P11PU：P1.1 上拉电阻使能

0：P1.1 为开漏状态无上拉

1：P1.1 为开漏上拉

P10PU：P1.0 上拉电阻使能

0：P1.0 为开漏状态无上拉

1：P1.0 为开漏上拉

P1.1、P1.0 默认是高阻输入和 N-MOS 无上拉输出模式。能在 AUXR 中个别的设置 P1.1/P1.0 是否上拉。如果 P1.1、P1.0 用作 GPIO 口且 P11PU/P10PU 被使能的情况，在掉电模式时，若没有外部上拉 CPU 不能驱动低电平。

AUXR1：附加控制寄存器 1

地址为 A2H，读/写，复位初始值 0000-0000B

7	6	5	4	3	2	1	0
P14FD	*	*	*	GF2	*	*	*

P14FD：P1.4 快速驱动

0：正常驱动

1：快速驱动

GF2：通用标志 2

*：细节请参考 MG87FE/L6051_4051_2051 V0.10.pdf 应用文件说明 **(建议用户将这些位，保持为 0)**

19. 硬件选项设置

LOCK	对 ROM 区域加锁，外部工具不能对它进行读、写和擦除动作
HWBS	上电时，如果 ISP 空间被配置，是否首先执行 ISP 代码
HWBS2	如果 ISP 空间被配置，则在上电和按复位键复位时强制从 ISP 代码开始执行
EN6T	MCU 工作 6T/12T 模式选择，使能为 6T 模式
OSCDN	用于设置晶振增益的驱动能力，当使能后可以减小 EMI，同时会减少耗电

20. 最大绝对额定参数

MG87FE/L2051/4051/6051（5V 应用）

参数	额定值	单位
环境温度偏差	-55 ~ +125	°C
存储温度	-65 ~ +150	°C
IO 口和复位脚的对地电压	-0.5 ~ VDD+0.5	V
VDD 脚的对地电压	-0.5 ~ +6.0	V
芯片总电流	400	mA
IO 口的最大吸收电流	40	mA

MG87FE/L2051/4051/6051（3.3V 应用）

参数	额定值	单位
环境温度偏差	-55 ~ +125	°C
存储温度	-65 ~ +150	°C
IO 口和复位脚的对地电压	-0.3 ~ VDD+0.3	V
VDD 脚的对地电压	-0.3 ~ +4.2	V
芯片总电流	400	mA
IO 口的最大吸收电流	40	mA

注意：实际参数超过上述各项“绝对最大额定值”可能会对设备造成永久性损坏。这些参数是一个设备进行正常功能操作的最大额定值，任何超过上述各项的条件都不被建议，否则可能会影响设备运行的稳定性。

21. 电气特性

21.1. 直流特性

VDD=5.0V，VSS=0V，TA=25°C，12T 模式

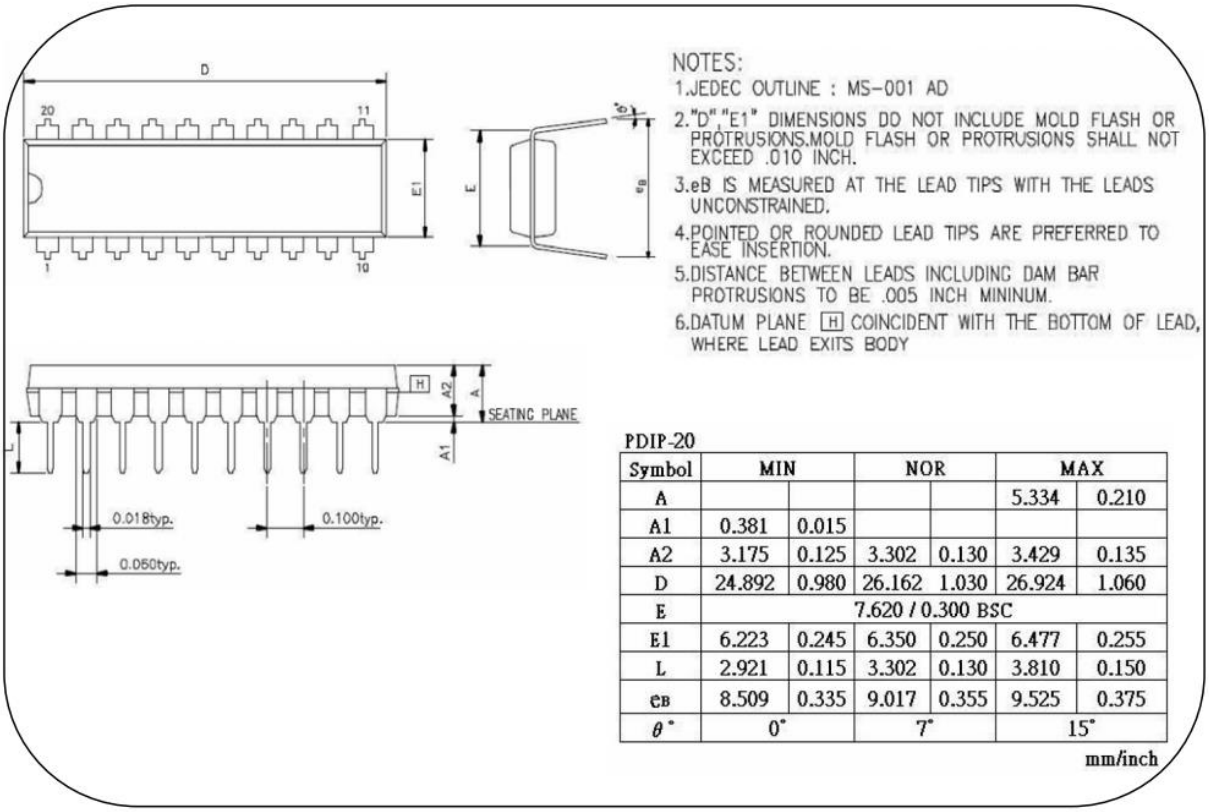
符号	参数	条件	范围			单位
			最小	标称	最大	
V_{IH1}	输入高电压（P1，P3，P4）		2.0			V
V_{IH2}	输入高电压（复位脚 RESET）		3.5			V
V_{IL1}	输入低电压（P1，P3，P4）				0.8	V
V_{IL2}	输入低电压（复位脚）				1.6	V
I_{IH}	输入高的漏电流（P1，P3，P4）	V_{PIN}=VDD		0	10	uA
I_{IL}	输入低的电流（P1，P3，P4）	V_{PIN}=0.4V		20	50	uA
I_{H2L}	输入下降沿的跳变电流（P1，P3，P4）	V_{PIN}=1.8V		250	500	uA
I_{OH1}	输出高的电流（P1，P3，P4）	V_{PIN}=2.4V	150	220		uA
I_{OL1}	输出低的电流（P1，P3，P4）	V_{PIN}=0.4V	12			mA
I_{OP}	工作电流	Fosc=12MHz		8	16	mA
		Fosc=24MHz		10	20	
I_{IDLE}	空闲模式电流	Fosc=12MHz		4	8	mA
		Fosc=24MHz		5	10	
I_{PD}	掉电模式电流			0.1	10	uA
R_{RST}	复位脚上的内部下拉电阻			100		Kohm

VDD=3.3V，VSS=0V，TA=25°C，12T 模式

符号	参数	条件	范围			单位
			最小	标称	最大	
V_{IH1}	输入高电压（P1，P3，P4）		2.0			V
V_{IH2}	输入高电压（复位脚 RESET）		2.8			V
V_{IL1}	输入低电压（P1，P3，P4）				0.8	V
V_{IL2}	输入低电压（复位脚）				1.5	V
I_{IH}	输入高的漏电流（P1，P3，P4）	V_{PIN}=VDD		0	10	uA
I_{IL}	输入低的电流（P1，P3，P4）	V_{PIN}=0.4V		7	30	uA
I_{H2L}	输入下降沿的跳变电流（P1，P3，P4）	V_{PIN}=1.8V		100	250	uA
I_{OH1}	输出高的电流（P1，P3，P4）	V_{PIN}=2.4V	40	70		uA
I_{OL1}	输出低的电流（P1，P3，P4）	V_{PIN}=0.4V	8			mA
I_{OP}	工作电流	Fosc=12MHz		6	12	mA
		Fosc=24MHz		8	16	
I_{IDLE}	空闲模式电流	Fosc=12MHz		2	4	mA
		Fosc=24MHz		2.5	5	
I_{PD}	掉电模式电流			0.1	50	uA
R_{RST}	复位脚上的内部下拉电阻			200		Kohm

22. 封装尺寸

PDIP-20



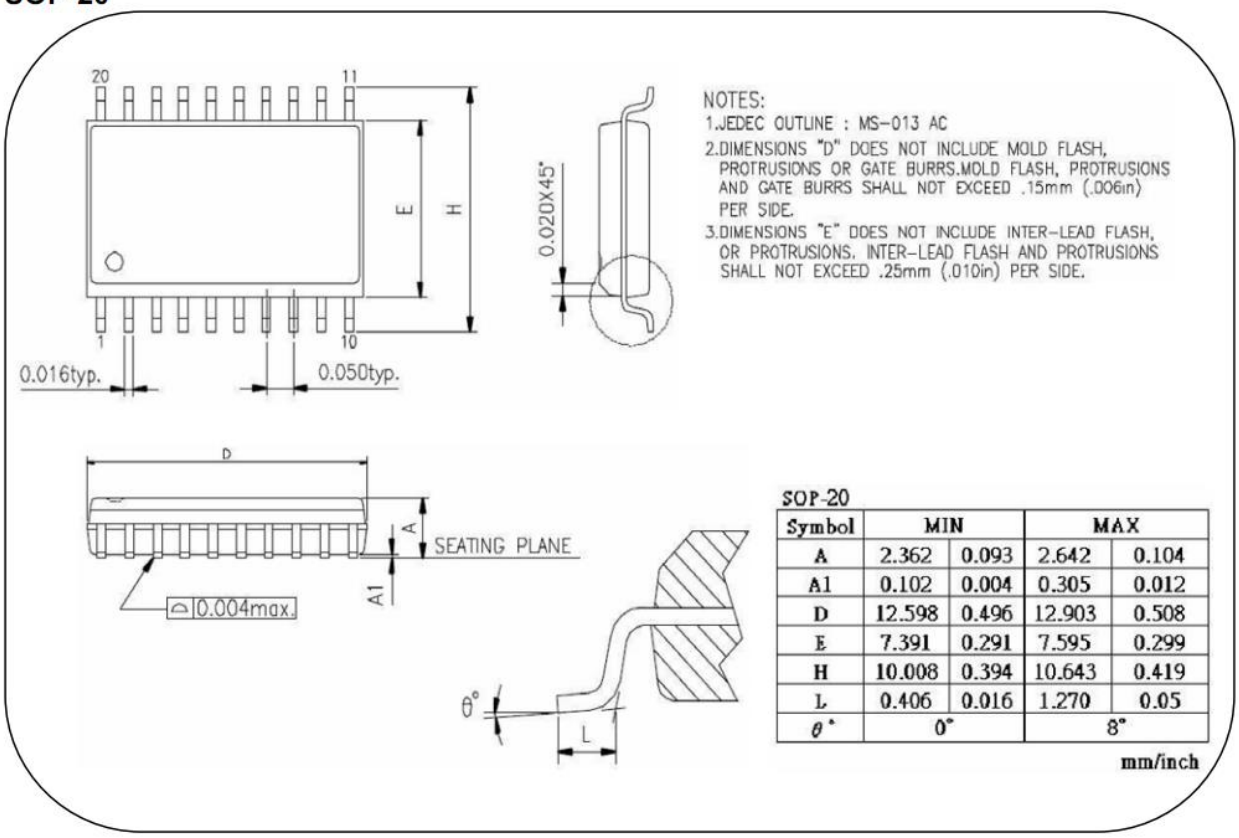
NOTES:
1. JEDEC OUTLINE : MS-001 AD
2. "D","E1" DIMENSIONS DO NOT INCLUDE MOLD FLASH OR PROTRUSIONS. MOLD FLASH OR PROTRUSIONS SHALL NOT EXCEED .010 INCH.
3. eB IS MEASURED AT THE LEAD TIPS WITH THE LEADS UNCONSTRAINED.
4. POINTED OR ROUNDED LEAD TIPS ARE PREFERRED TO EASE INSERTION.
5. DISTANCE BETWEEN LEADS INCLUDING DAM BAR PROTRUSIONS TO BE .005 INCH MININUM.
6. DATUM PLANE [H] COINCIDENT WITH THE BOTTOM OF LEAD, WHERE LEAD EXITS BODY

PDIP-20

Symbol	MIN		NOR		MAX	
A					5.334	0.210
A1	0.381	0.015				
A2	3.175	0.125	3.302	0.130	3.429	0.135
D	24.892	0.980	26.162	1.030	26.924	1.060
E	7.620 / 0.300 BSC					
E1	6.223	0.245	6.350	0.250	6.477	0.255
L	2.921	0.115	3.302	0.130	3.810	0.150
eB	8.509	0.335	9.017	0.355	9.525	0.375
θ°	0°		7°		15°	

mm/inch

SOP-20



SOP-20

Symbol	MIN		MAX	
A	2.362	0.093	2.642	0.104
A1	0.102	0.004	0.305	0.012
D	12.598	0.496	12.903	0.508
E	7.391	0.291	7.595	0.299
H	10.008	0.394	10.643	0.419
L	0.406	0.016	1.270	0.05
θ°	0°		8°	

mm/inch

23. 指令介绍

Rn	暂存器 R0~R7
direct	8 位内部存储器，包括 1. 内部存储器(00~7F)的地址 2. 特殊功能寄存器(80~FF)的地址，如 P0,PSW,TMOD,…等
@Ri	由寄存器 R0 或 R1 所索引的内部 RAM 数据
#data	8 位常数
#data16	16 位常数
Addr16	16 位的目的地址，可使跳转指令跳转 64K
Addr11	11 位的目的地址，可使跳转指令跳转 2K
rel	有正负号的 8 位地址偏移量，用于相对地址的跳转
bit	1 个 bit: 指所有可以位寻址的的位
A	累加器 Acc
C 或 CY	进位标志
AC	辅助进位标志
Bb	指定位 B0~B7
D	半位组(4bit)
F0	旗号 0
I	中断
PC	程序计数器
SP	堆栈
B	寄存器 B
DPTR	程序数据地址寄存器
@	间接寻址符号
$	程序计数器当前的值
reg	寄存器

* 6T 模式: 1 个指令周期= 6 个时钟周期

* 12T 模式: 1 个指令周期=12 个时钟周期

数据传送			
助记符	描述	字节数	指令周期
MOV A, Rn	Acc ← Rn	1	1
MOV A, direct	Acc ← direct	2	1
MOV A, @Ri	Acc ← Ri	1	1
MOV A, #data	Acc ← data	2	1
MOV Rn,A	Rn ← Acc	1	1
MOV Rn,direct	Rn ← direct	2	2
MOV Rn,#data	Rn ← data	2	1
MOV direct,A	direct ← Acc	2	1
MOV direct,Rn	direct ← Rn	2	2
MOV direct,direct	direct ← direct	3	2
MOV direct,@Ri	direct ← Ri	2	2

MOV direct,#data	direct ← data	3	2
MOV @Ri,A	Ri ← Acc	1	1
MOV @Ri,direct	Ri ← direct	2	2
MOV @Ri,#data	Ri ← data	2	1
MOV DPTR,#data16	DPTR ← 16bit data	3	1
MOVC A,@A+DPTR	Acc ← (A+DPTR)地址所指的数据	1	2
MOVC A,@A+PC	Acc ← (A+PC)地址所指的数据	1	2
MOVX A,@Ri	Acc←外部 data	1	2
MOVX A,@DPTR	Acc←外部 16bit data	1	2
MOVX @Ri,A	外部 data← Acc	1	2
MOVX @DPTR,A	外部 16bit data← Acc	1	2
PUSH direct	堆栈 ← direct	2	2
POP direct	direct ← 堆栈	2	2
XCH A,Rn	A 和 Rn 互换	1	1
XCH A,direct	A 和 direct 互换	2	1
XCH A,@Ri	A 和 Ri 互换	1	1
XCHD A,@Ri	A 和 Ri 的低四互换	1	1
算术运算			
ADD A,Rn	Acc ← Acc+Rn	1	1
ADD A,direct	Acc ← Acc+direct	2	1
ADD A,@Ri	Acc ← Acc+Ri	1	1
ADD A,#data	Acc ← Acc+data	2	1
ADDC A,Rn	Acc ← Acc+Rn+C	1	1
ADDC A,direct	Acc ← Acc+direct+C	2	1
ADDC A,@Ri	Acc ← Acc+Ri+C	1	1
ADDC A,#data	Acc ← Acc+data+C	2	1
SUBB A,Rn	Acc ← Acc-Rn-C	1	1
SUBB A,direct	Acc ← Acc-direct-C	2	1
SUBB A,@Ri	Acc ← Acc-Ri-C	1	1
SUBB A,#data	Acc ← Acc-data-C	2	1
INC A	Acc ← Acc +1	1	1
INC Rn	Rn ← Rn +1	1	1
INC direct	direct ← direct +1	2	1
INC @Ri	Ri ← Ri +1	1	1
INC DPTR	DPTR ← DPTR +1	1	2
DEC A	Acc ← Acc - 1	1	1
DEC Rn	Rn ← Rn -1	1	1
DEC direct	direct ← direct -1	2	1
DEC @Ri	Ri ← Ri -1	1	1
MUL AB	两数相乘，结果高八位存入 B,低八位存入 A	1	4
DIV AB	Acc 除以 B，商存入 Acc,余数存入 B	1	4
DA A	Acc 作十进制调整	1	1

逻辑运算			
ANL A,Rn	Acc ← Acc and Rn	1	1
ANL A,direct	Acc ← Acc and direct	2	1
ANL A,@Ri	Acc ← Acc and Ri	1	1
ANL A,#data	Acc ← Acc and data	2	1
ANL direct,A	Direct ← direct and Acc	2	1
ANL direct,#data	Direct ← direct and data	3	2
ORL A,Rn	Acc ← Acc or Rn	1	1
ORL A,direct	Acc ← Acc or direct	2	1
ORL A,@Ri	Acc ← Acc or Ri	1	1
ORL A,#data	Acc ← Acc or data	2	1
ORL direct,A	Direct ← direct or Acc	2	1
ORL direct,#data	Direct ← direct or data	3	2
XRL A,Rn	Acc ← Acc xor Rn	1	1
XRL A,direct	Acc ← Acc xor direct	2	1
XRL A,@Ri	Acc ← Acc xor Ri	1	1
XRL A,#data	Acc ← Acc xor data	2	1
XRL direct,A	Direct ← direct xor Acc	2	1
XRL direct,#data	Direct ← direct xor data	3	2
CLR A	清除累加器 Acc	1	1
CPL A	累加器反相	1	1
RL A	累加器向左旋转	1	1
RLC A	累加器和 C 左旋	1	1
RR A	累加器向右旋转	1	1
RRC A	累加器和 C 右旋	1	1
SWAP A	累加器的高低四位互换	1	1
位逻辑运算			
CLR C	清除进位标记	1	1
CLR bit	清除直接位	2	1
SETB C	设定进位标记	1	1
SETB bit	设定直接位	2	1
CPL C	进位标记取反	1	1
CPL bit	直接位取反	2	1
ANL C,bit	C ← C and bit	2	2
ANL C,/bit	C ← C and bit(反相)	2	2
ORL C,bit	C ← C or bit	2	2
ORL C,/bit	C ← C or bit(反相)	2	2
MOV C,bit	C ← bit	2	1
MOV bit,C	bit ← bit	2	2
位逻辑跳转			
JC rel	如果 C=1 跳转到 rel	2	2
JNC rel	如果 C=0 跳转到 rel	2	2
JB bit,rel	如果 bit=1 跳转到 rel	3	2

JNB bit,rel	如果 bit=0 跳转到 rel	3	2
JBC bit,rel	如果 bit=1 跳转到 rel,并且清除 bit	3	2
程序跳转			
ACALL addr11	绝对式子程序调用	3	2
LCALL addr16	远程子程序调用	3	2
RET	从子程序返回	1	2
RETI	从中断返回	1	2
AJMP addr11	绝对式跳转	2	2
LJMP addr16	远程跳转	3	2
SJMP rel	短程跳转	2	2
JMP @A+DPTR	间接跳转	1	2
JZ rel	如果 Acc=0 则跳到 rel	2	2
JNZ rel	如果 Acc≠0 则跳到 rel	2	2
CJNE A,direct,rel	如果 Acc≠direct 则跳到 rel	3	2
CJNE A,#data,rel	如果 Acc≠data 则跳到 rel	3	2
CJNE Rn,#data,rel	如果 Rn≠data 则跳到 rel	3	2
CJNE @Ri,#data,rel	如果 Ri≠data 则跳到 rel	3	2
DJNZ Rn,rel	如果(Rn-1)≠0 则跳到 rel	2	2
DJNZ direct,rel	如果(direct-1)≠0 则跳到 rel	3	2
NOP	无动作	1	1

24. 修订历史

版本	描述	日期
V1.00	创建文档	2009/02/14
V1.01	1. 修改时钟结构图	2009/04/17
	2. CUS-Timer -> PWM-Timer	
V1.02	修改 AUXR1 初始值以及添加应用说明	2010/09/14
A1.0	重新编排版本	2014/03/10

免责声明

在此，笙泉（Megawin）代表“*Megawin Technology Co., Ltd.*”

生命支援

此产品并不是为医疗、救生或维持生命而设计的，并且当设备系统出现故障时，并不能合理地预示是否会对人身造成伤害。因此，当客户使用或出售用于上述应用的产品时，需要客户自己承担这样做的风险，笙泉公司并不会对不当地使用或出售我公司的产品而造成的任何损害进行赔偿。

更改权

笙泉保留产品的如下更改权，其中包括电路、标准单元、与/或软件 - 在此为提高设计的与/或性能的描述或内容。当产品在大批量生产时，有关变动将通过工程变更通知（ECN）进行通知。