



**8位单片机**  
**GMS87C1102**  
**GMS87C1202**

**使用手册**

**2001年6月版**  
(2001年 10月中译)

## GMS87C1102/GMS87C1202 CMOS 8 位单片机

### 1. 概述

#### 1.1 描述

GMS87C1102和GMS87C1202, 是带2K字节ROM的8位单片微控制器。本芯片可以为许多小型应用提供高度灵活和低成本解决方案。GMS87C1102/GMS87C1202提供如下标准特性：2K字节ROM(OTP)，128字节RAM，8位定时/计数器，8位A/D转换器，10位高速PWM输出，可编程蜂鸣器驱动口(仅限GMS87C1202)，片内振荡和时钟电路。另外，两者还提供节电模式以降低功耗。

本文档仅对GMS87C1202的基本特性作如下说明。

器件名称	OTP ROM 大小	RAM 字节数	I/O	BUZ	INT1	封装类型	操作温度
GMS87C1102	2K	128	11	无	无	16DIP/SOP	-20°C~+85°C
GMS87C1202	2K	128	15	有	有	20DIP/SOP	-20°C~+85°C

#### 1.2 特性

- 2K 片内程序存储器
- 128字节片内RAM
- 最短指令执行时间  
- 8MHz频率 500ns( 2周期 NOP指令)
- 宽操作电压范围2.7~6.0V
- 基本的间隔定时器
- 两个8 位定时/计数器
- 10位高速PWM 输出
- 2个外部中断口  
(GMS87C1102有1个)
- 1个可编程蜂鸣器驱动口  
(仅对GMS87C1202)
- 15个可编程 I/O口  
(GMS87C1102有11个)
- 7个中断源  
(GMS87C1102有6个)
- 8路 8位片内A/D 转换器
- 看门狗定时器
- 振荡器
  - 晶振
  - 陶瓷振荡器
  - 外部振荡电路
  - RC 振荡器
- 节电模式
  - STOP模式
  - 唤醒定时器模式
  - RC-WDT 模式
- 电源失效处理噪声消除电路

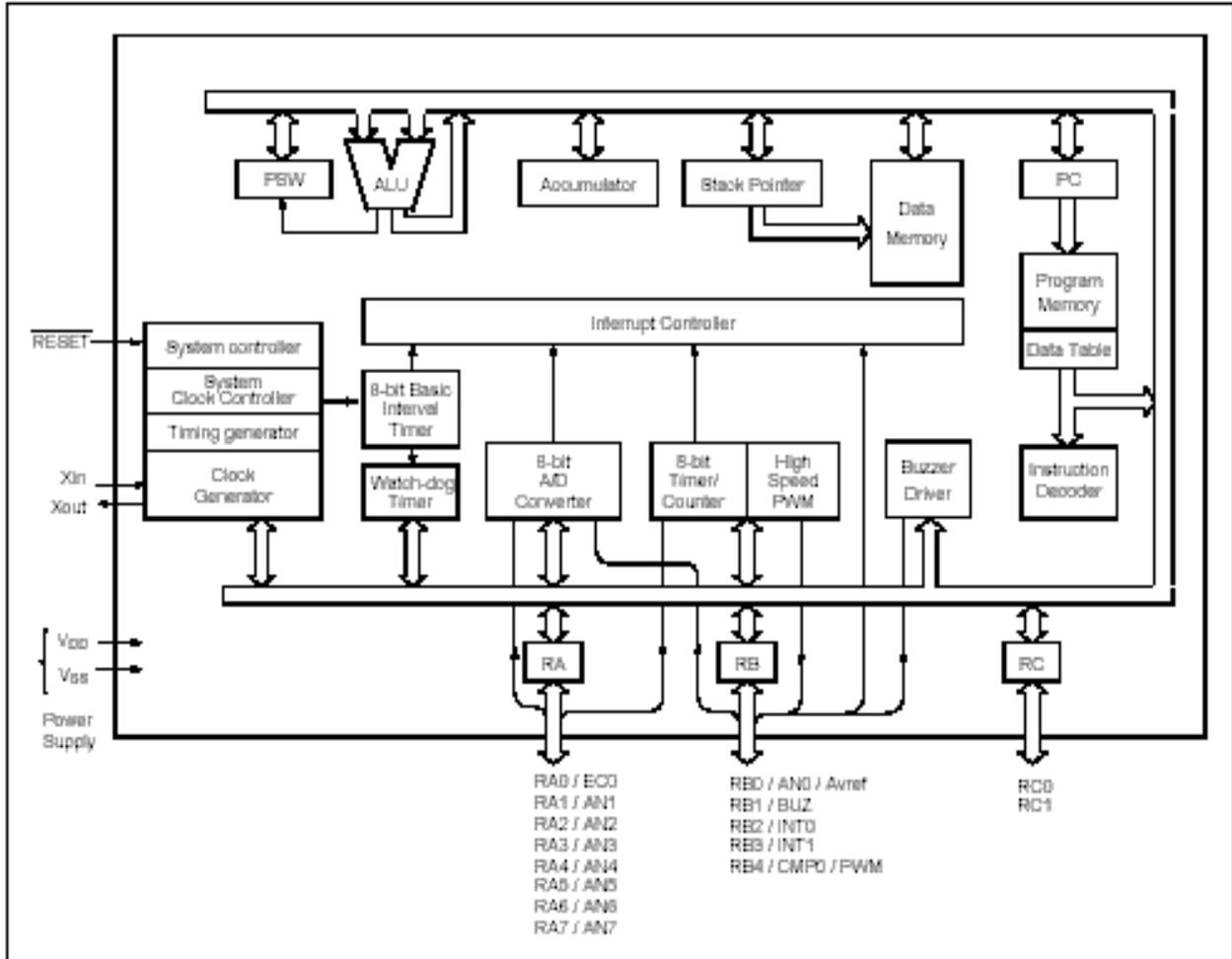
#### 1.3 开发工具

GMS800系列单片机有一套完整的开发工具，包括汇编器，仿真器 CHOICE-Dr.，OTP烧写器 Dr.Writer等。

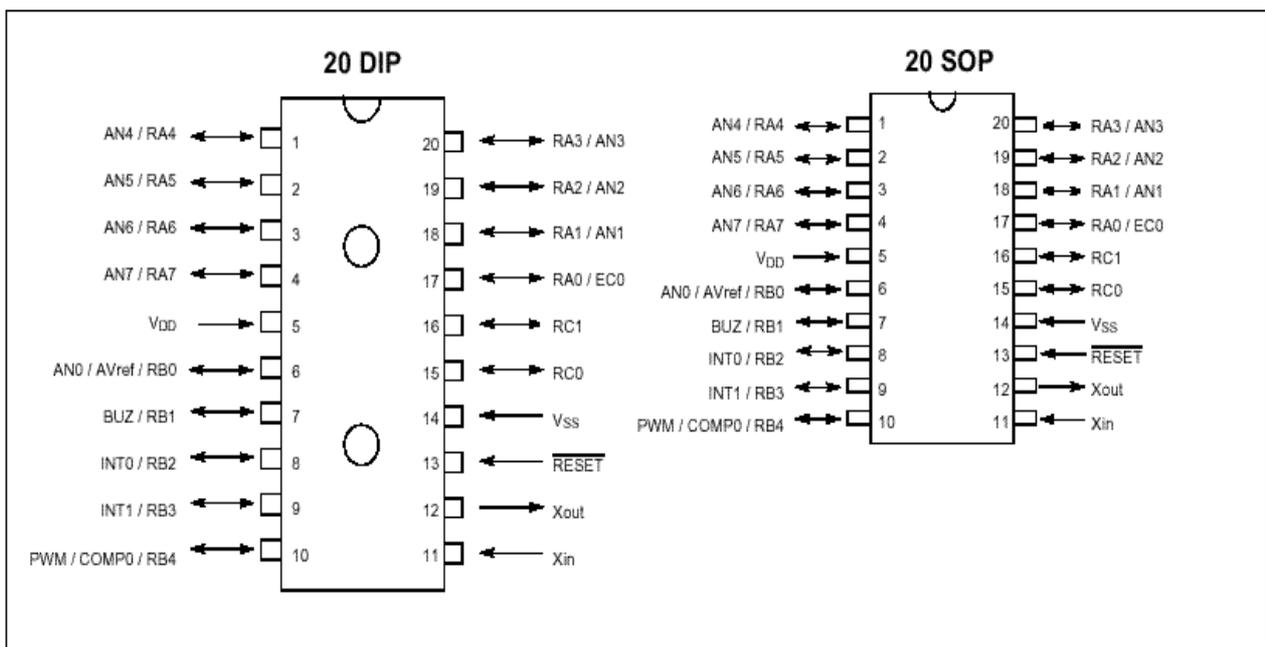
OTP适用于客户需对控制码经常进行更改或升级的器件。OTP单片机用塑膜封装，允许用户对其进行一次编程。

仿真器	CHOICE-Dr.
Assembler	HYNIX Semiconductor Macro Assembler
OTP Writer	Dr. Writer

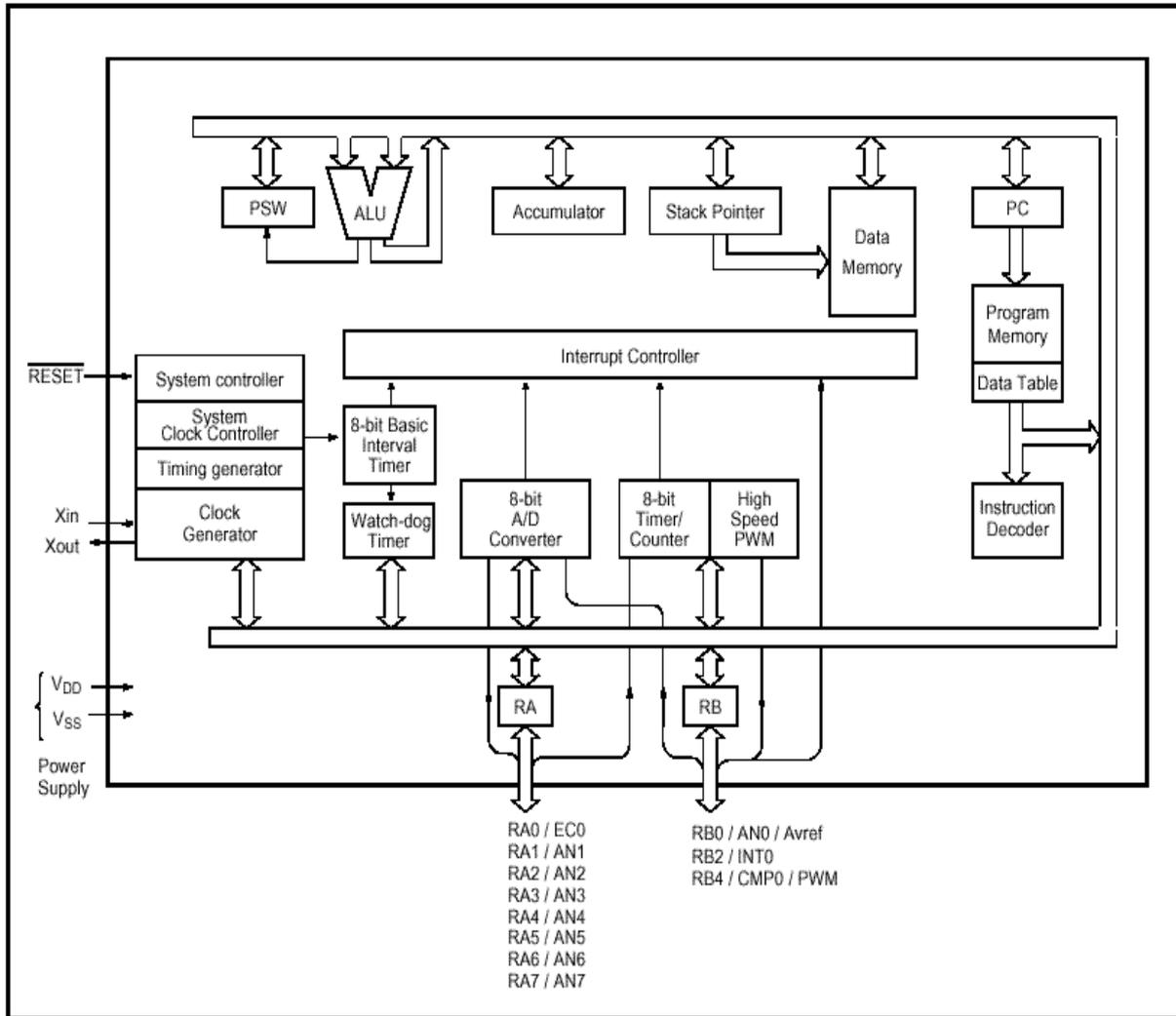
2. 功能框图(GMS87C1202)



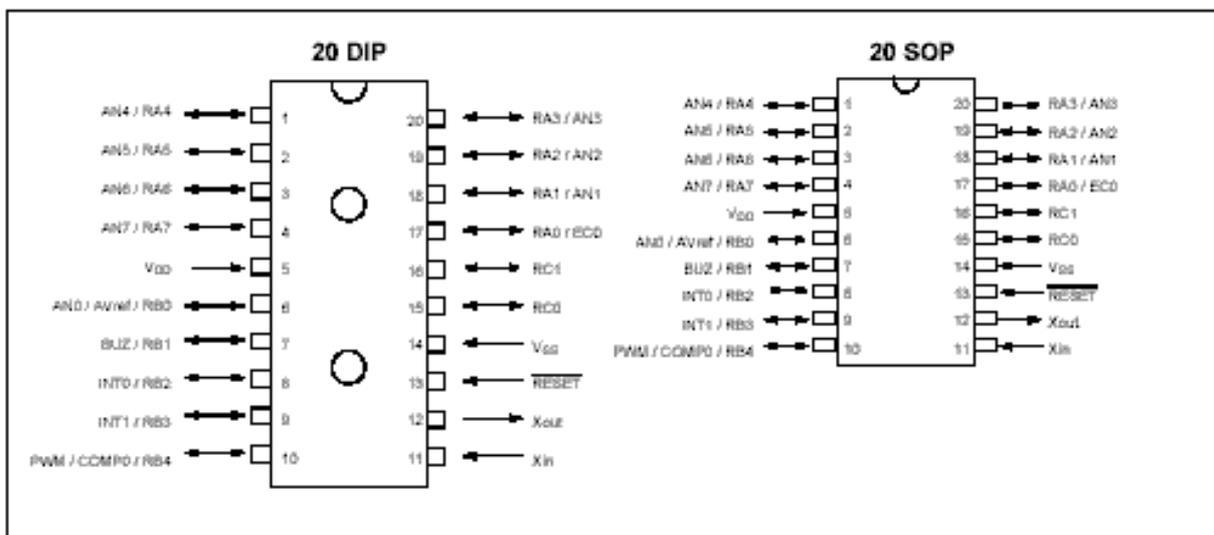
3. 管脚排列(GMS87C1202)



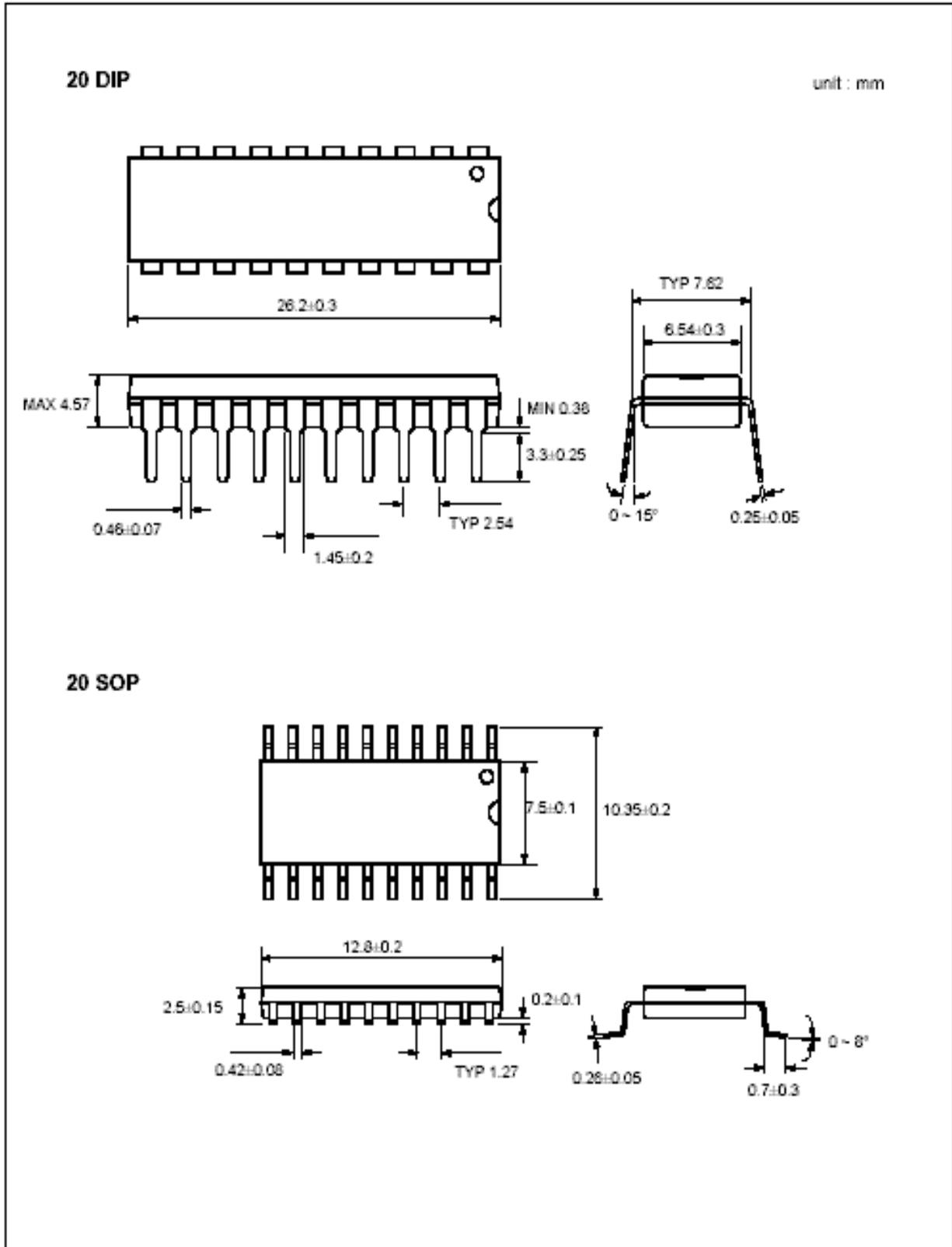
4. 功能框图(GMS87C1102)



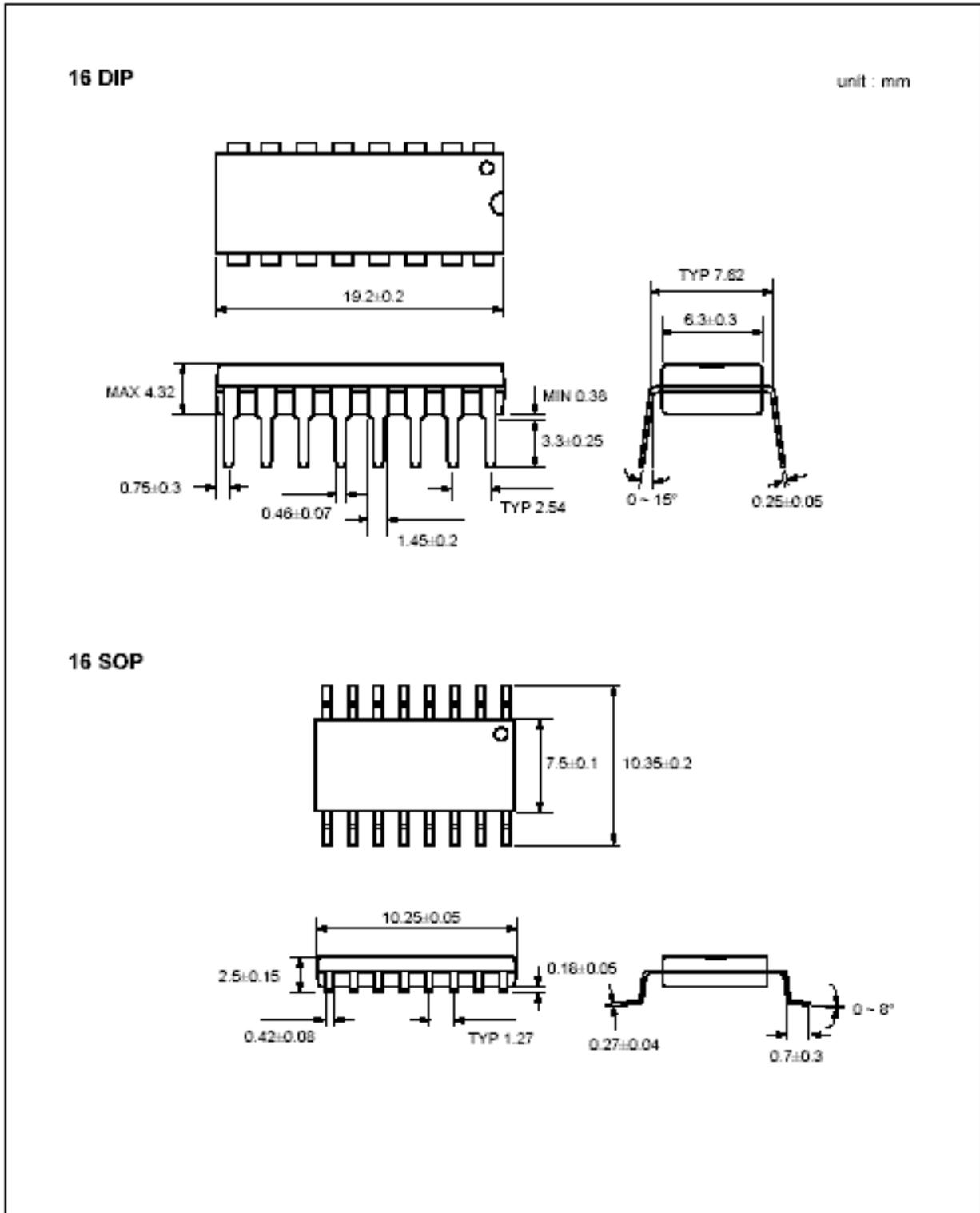
5. 管脚排列(GMS87C1102)



6. 封装尺寸(GMS87C1202)



7. 封装尺寸(GMS87C1102)



## 8. 管脚功能

$V_{DD}$  : 电源

$V_{SS}$  : 地

RESET : 复位

$X_{IN}$  : 反相振荡放大器输入,内部时钟电路输入端。

$X_{OUT}$  : 反相振荡放大器输出。如果使用内部RC振荡,振荡器频率4分频后从  $X_{OUT}$  脚输出。

**RA0~RA7:** RA 是一个8位, CMOS双向 I/O口。将口方向寄存器(RAIO)写入0或1,来决定RA

用作输入或输出。另外,RA还用于几种不同的特殊功能(见表8-1)。

**RB0~RB4:** RB是一个5位, CMOS双向口。将口方向寄存器(RBIO)写入0或1,来决定RB用作输入或输出。RB还用于几种不同的特殊功能(见表8-2)。

**RC0~RC1:** RC是一个2位, CMOS双向口。将口方向寄存器(RCIO)写入0或1,来决定RC用作输入或输出。

管脚	可选功能
RA0	EC0 (事件计数器输入)
RA1	AN1 (模拟输入 1)
RA2	AN2 (模拟输入 2)
RA3	AN3 (模拟输入 3)
RA4	AN4 (模拟输入 4)
RA5	AN5 (模拟输入 5)
RA6	AN6 (模拟输入 6)
RA7	AN7 (模拟输入 7)

表 8-1. RA Port

管脚	可选功能
RB0	AN0 (模拟输入 0)
	Avref(外部模拟参考脚)
RB1	BUZ (蜂鸣器驱动输出口)
RB2	INT0 (外部中断输入 0)
RB3	INT1 (外部中断输入 1)
RB4	PWM (PWM 输出)
	COMP0 (定时器 0 比较输出)

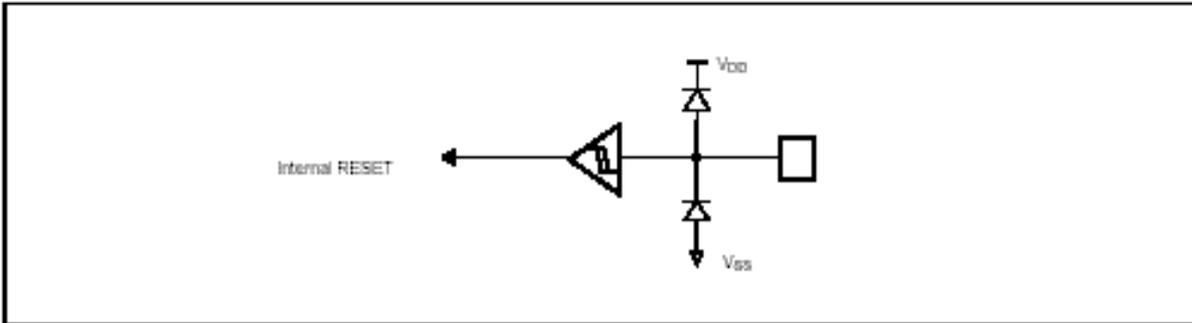
表 8-2. RB Port

管脚名称	管脚号	I/O	功能
$V_{DD}$	5	-	电源
$V_{SS}$	14	-	地
RESET	13	I	复位信号输入
$X_{IN}$	11	I	
$X_{OUT}$	12	O	
RA0(EC0)	17	I/O(输入)	外部事件计数器输入
RA1(AN1)	18	I/O(输入)	模拟输入 1
RA2(AN2)	19	I/O(输入)	模拟输入 2
RA3(AN3)	20	I/O(输入)	模拟输入 3
RA4(AN4)	1	I/O(输入)	模拟输入 4
RA5(AN5)	2	I/O(输入)	模拟输入 5
RA6(AN6)	3	I/O(输入)	模拟输入 6
RA7(AN7)	4	I/O(输入)	模拟输入 7
RB0(Avref/AN0)	6	I/O(输入)	模拟输入 0/模拟参考
RB1(BUZ)	7	I/O(输入)	蜂鸣器驱动输出
RB2(INT0)	8	I/O(输入)	外部中断输入 0
RB3(INT1)	9	I/O(输入)	外部中断输入 1
RB4(PWM/COMP0)	10	I/O(输入/输出)	PWM 输出或定时器 0 比较输出
RC0	15	I/O	2 位通用 I/O 口
RC1	16	I/O	

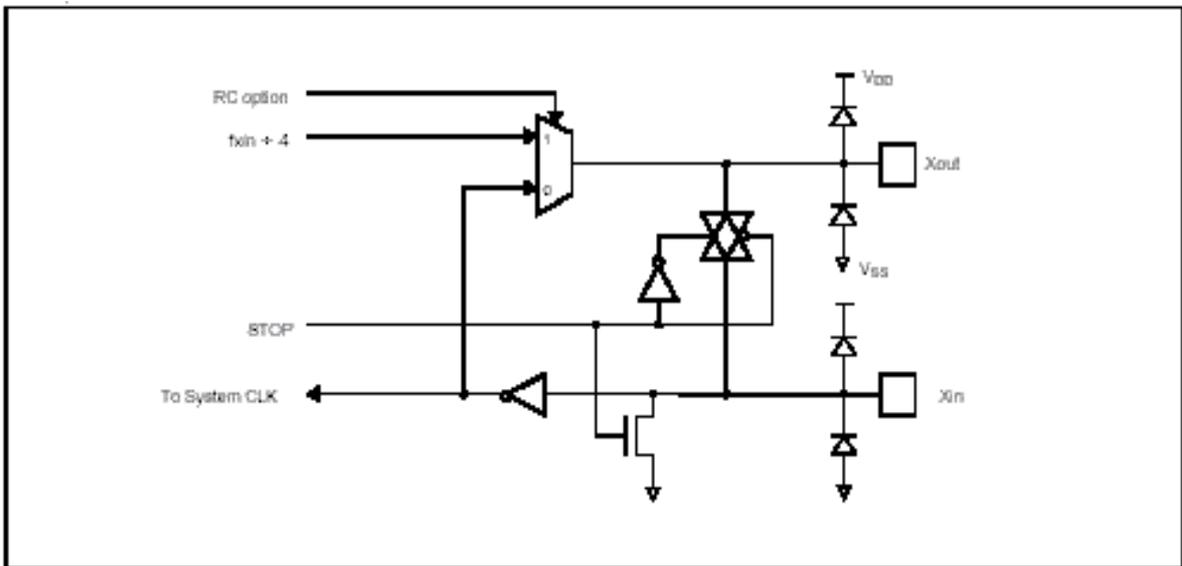
表 8-3. 管脚描述

9. I/O 口的结构

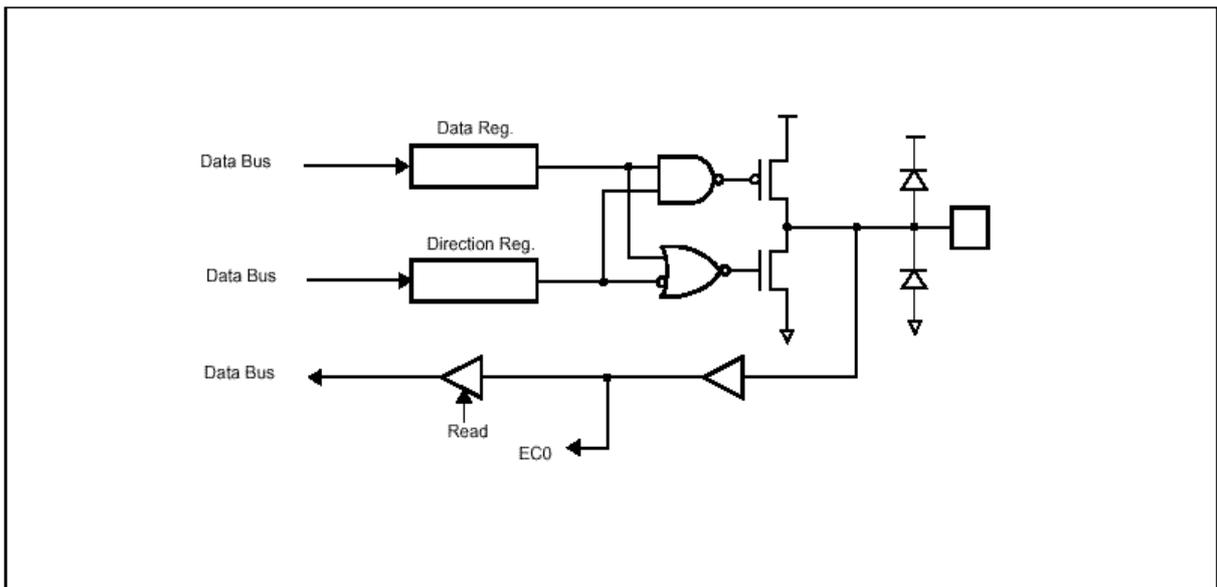
• RESET



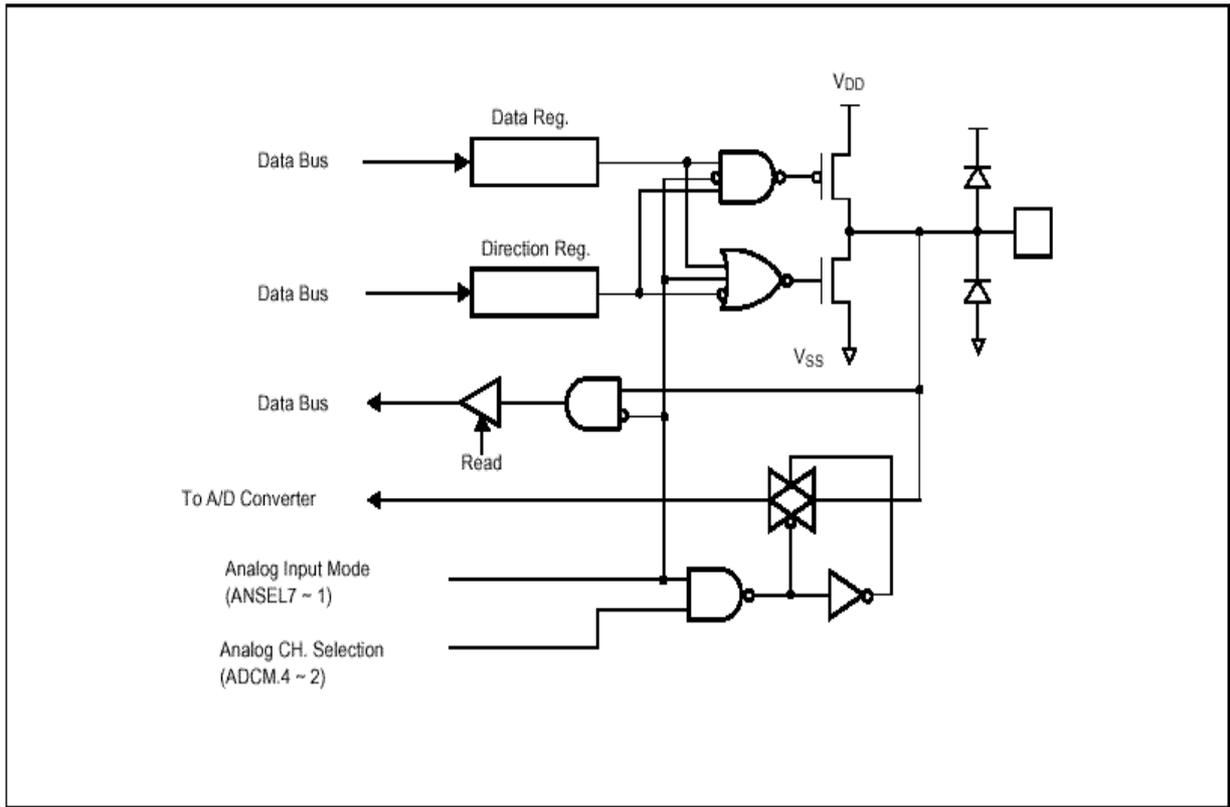
• X<sub>IN</sub>, X<sub>OUT</sub>



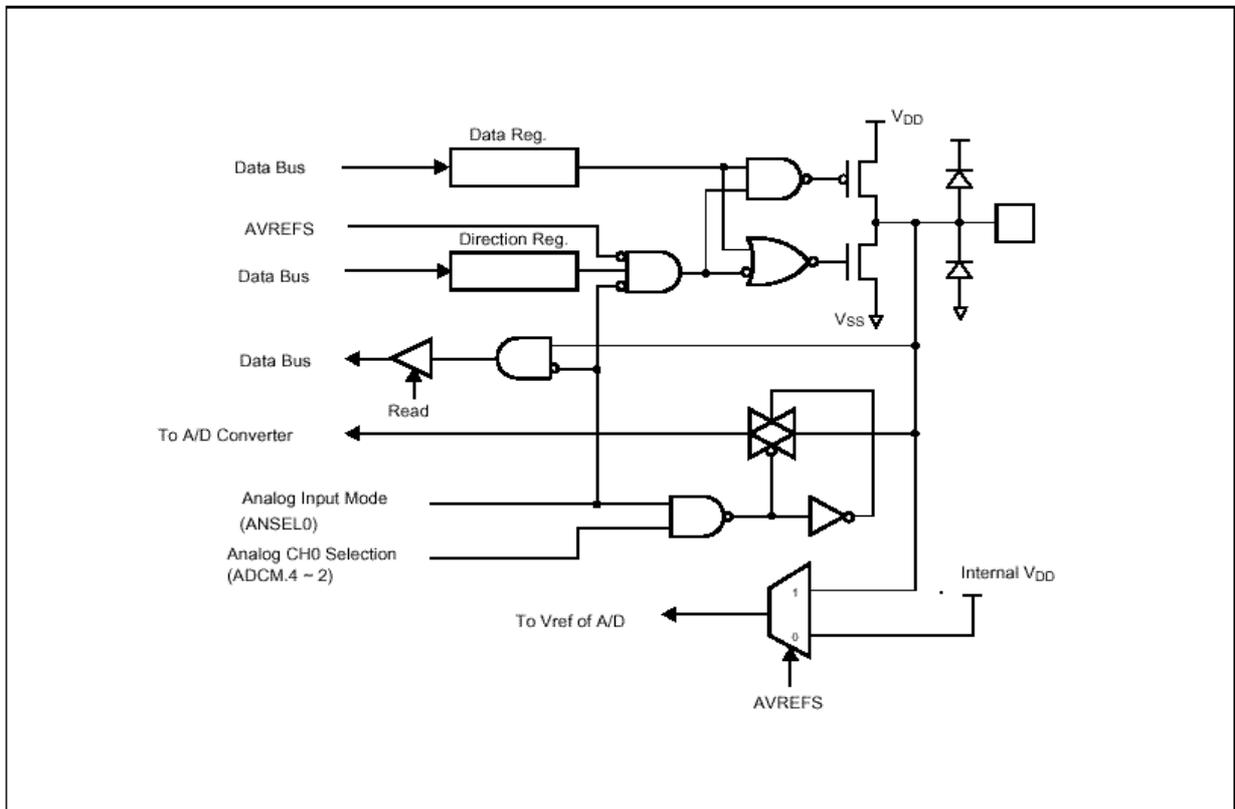
• RA0/EC0



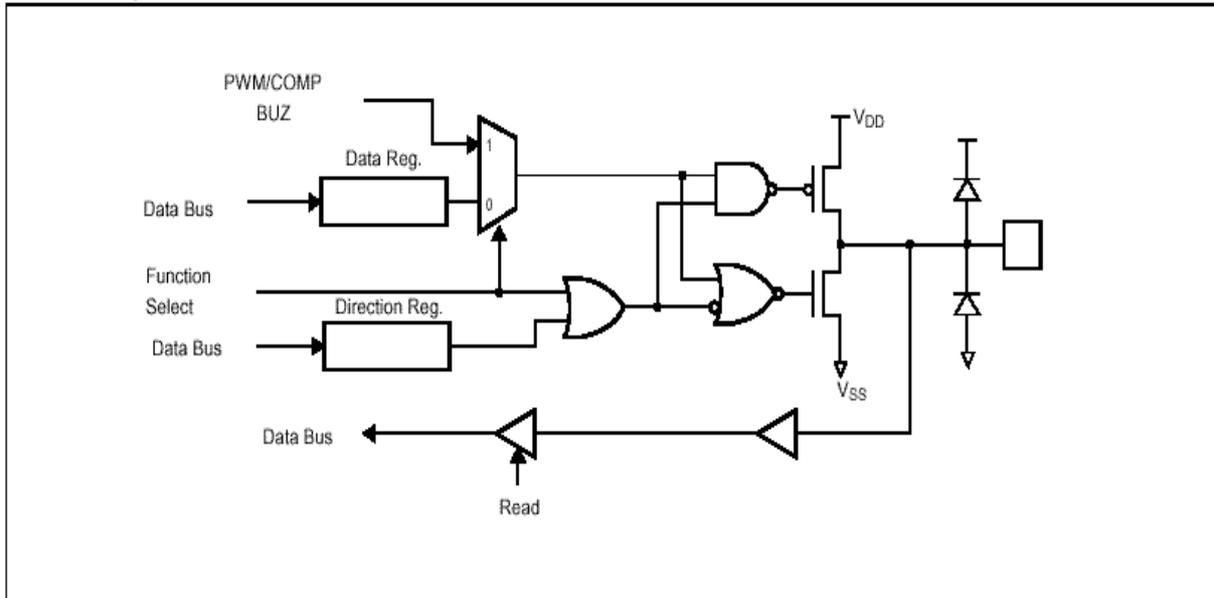
• RA1/AN1~RA7/AN7



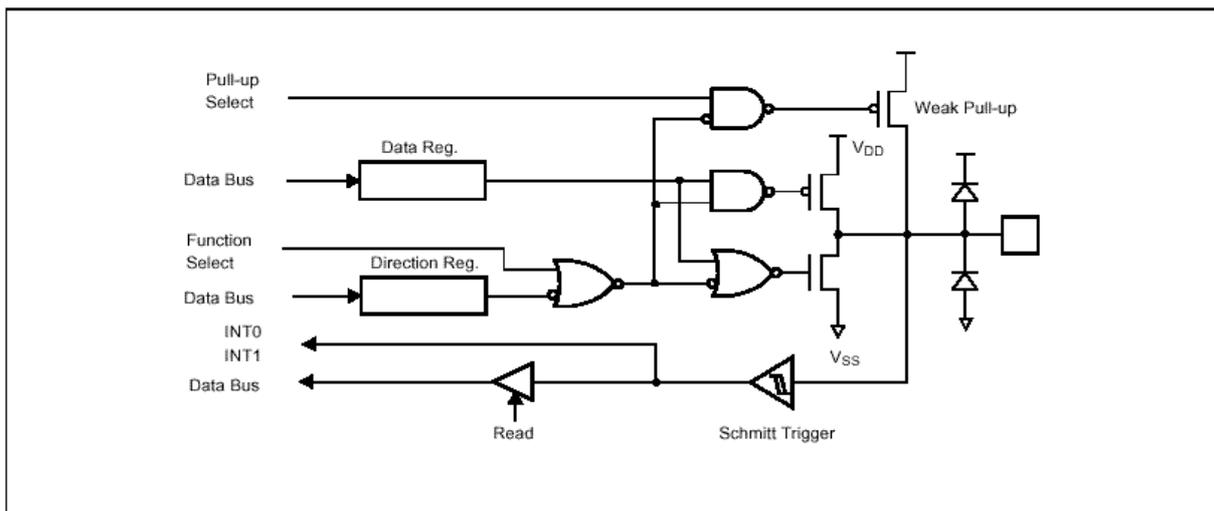
• RB0/AN0/AVref



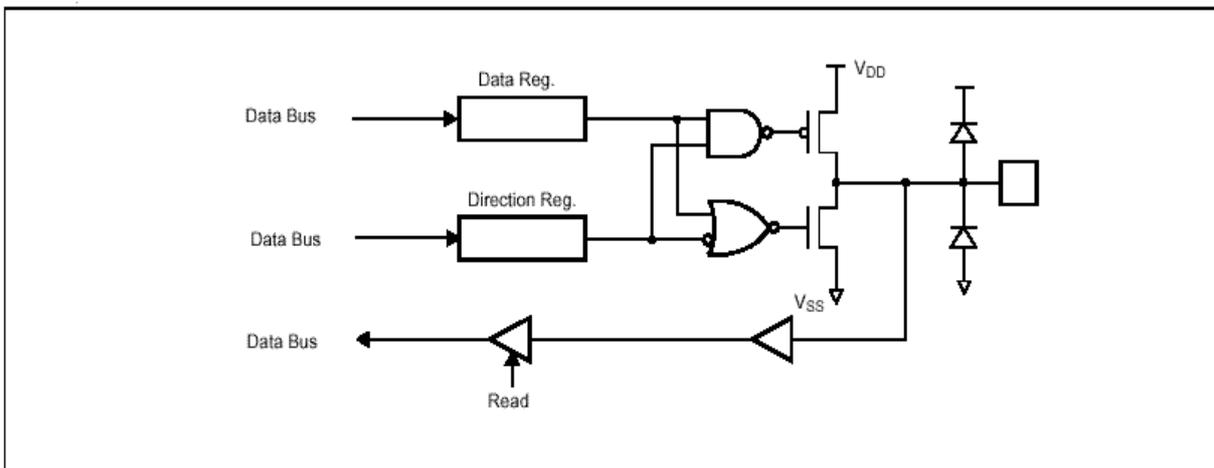
• RB1/BUZ, RB4/PWM0/COMP



• RB2/INT0, RB3/INT1



• RC0, RC1



## 10. 电气特性

## 10.1. 极限参数

电源电压.....	-0.3 ~ +6.5V
储存温度.....	-40°C ~ +125°C
任意脚对地V <sub>SS</sub> 电压.....	-0.3~V <sub>DD</sub> +0.3
V <sub>SS</sub> 脚最大输出电流.....	200mA
V <sub>DD</sub> 脚最大输入电流.....	150mA
最大灌电流(每个I/O口 I <sub>OL</sub> ) .....	25mA
最大拉电流(每个I/O口 I <sub>OH</sub> ) .....	15mA
最大电流(ΣI <sub>OL</sub> ).....	150mA
最大电流(ΣI <sub>OH</sub> ).....	100mA

## 10.2 推荐的操作条件

参数	符号	条件	规格		单位
			最大	最小	
电源电压	V <sub>DD</sub>	F <sub>XIN</sub> =1.2MHz	4.5	6	V
		F <sub>XIN</sub> =4.2MHz	2.7	6	
操作频率	S <sub>XIN</sub>	V <sub>DD</sub> =4.5 6.0V	1	12	MHz
		V <sub>DD</sub> =4.5 6.0V	1	4.2	
操作温度	f <sub>XIN</sub>	GMS87C1102/ GMS87C1202	-20	8.5	°C
	T <sub>CDR</sub>	GMS87C1102E/ GMS87C1202E	-40	125	

## 10.3. DC 电气特性--GMS87C1102, GMS87C1202

参数	符号	管脚	测试条件	规格			单位
				最小	典型	最大	
输入高电压	$V_{IH1}$	XIN, $\overline{\text{RESET}}$	$V_{DD}=4.5\sim 6.0V$	$0.8V_{DD}$		$V_{DD}$	V
	$V_{IH2}$	RB2, RB3		$0.8V_{DD}$		$V_{DD}$	
	$V_{IH3}$	RA, RBO, RB1, RB4, RC		$0.7V_{DD}$		$V_{DD}$	
输入低电压	$V_{IL1}$	XIN, $\overline{\text{RESET}}$	$V_{DD}=4.5\sim 6.0V$	0		$0.2V_{DD}$	V
	$V_{IL2}$	RB2, RB3		0		$0.2V_{DD}$	
	$V_{IL3}$	RA, RBO, RB1, RB4, RC		0		$0.3V_{DD}$	
输出高电压	$V_{OH}$	RA, RB, RC	$V_{DD}=5V, I_{OH}=-2mA$	$V_{DD}-1$			V
输出低电压	$V_{OL}$	RA, RB, RC	$V_{DD}=5V, I_{OL}=10mA$			1	V
输入漏电流	$I_{IL}$	$\overline{\text{RESET}}, RA, RB, RC$	$V_{IN}=V_{SS}\sim V_{DD}$			$-5\sim +5$	V
	$I_{IL}$	XIN				$-5\sim +5$	
输入上拉电流	$I_{PU}$	RB2, RB3 <sup>1</sup>	$V_{DD}=5V, V_{IN}=V_{SS}$	-350	-250	-100	
电源失败检测电压	$V_{PFD}$	$V_{DD}$		2	3.5	4.2	V
正常操作电流	$I_{DD}$	$V_{DD}$	$V_{DD}=5.5V, f_{XIN}=8MHz$		4	6	mA
			$V_{DD}=5.5V, f_{XIN}=12MHz$		6	10	
唤醒定时器模式电流	$I_{WKUP}$	$V_{DD}$	$V_{DD}=5.5V, f_{XIN}=8MHz$		1	1.8	mA
			$V_{DD}=5.5V, f_{XIN}=12MHz$		2	3	
RC-WDT模式电流	$I_{WKUP}$	$V_{DD}$	$V_{DD}=5.5V,$ $f_{XIN}=8MHz/12MHz$		0.3	0.8	mA
STOP模式电流	$I_{STOP}$	$V_{DD}$	$V_{DD}=5.5V$		0.2	2	mA
滞后电压	$V_{T+}\sim V_{T-}$	$\overline{\text{RESET}}, RB2, RB3$		0.5			uA
内部RC振荡周期	$T_{RCWDT}$	$X_{OUT}$	$V_{DD}=5.5V,$ $f_{XIN}=8MHz/12MHz$	80		250	uS
RC振荡器频率(系统时钟)	$f_{RCOSC}$	$X_{OUT}$	$R=20K, C=24pF$	300		550	kHz

1. RCO, RC1, RB3 脚只用于GMS87C1202。

2. 典型值一栏为5V, 25℃下的数据。这些参数未经过测试仅作为设计指南。

3. 当位PUPSELx 被选且设置为输入模式或中断输入功能时该参数无效。

( $V_{DD}=2.7\sim 6.0V, V_{SS}=0V, f_{XIN}=1MHz\sim 4.2MHz, T_A=20^{\circ}C\sim +85^{\circ}C$ )

参数	符号	管脚 <sup>1</sup>	测试条件	规格			单位
				最小	典型 <sup>2</sup>	最大	
输入高电压	$V_{IH1}$	XIN	$V_{DD}=2.7\sim 6.0V$	$0.8V_{DD}$		$V_{DD}$	V
	$V_{IH2}$	RESET		$0.9V_{DD}$		$V_{DD}$	
	$V_{IH3}$	RB2, RB3		$0.8V_{DD}$		$V_{DD}$	
	$V_{IH4}$	RA, RB0, RB1, RB4, RC		$0.7V_{DD}$		$V_{DD}$	
输入低电压	$V_{IL1}$	XIN	$V_{DD}=2.7\sim 6.0V$	0		$0.2V_{DD}$	V
	$V_{IL2}$	RESET		0		$0.1V_{DD}$	
	$V_{IL3}$	RB2, RB3		0		$0.2V_{DD}$	
	$V_{IL4}$	RA, RB0, RB1, RB4, RC		0		$0.3V_{DD}$	
输出高电压	$V_{OH}$	RA, RB, RC	$V_{DD}=3V, I_{OH}=-2mA$	$V_{DD}-1$			V
输出低电压	$V_{OL}$	RA, RB, RC	$V_{DD}=3V, I_{OL}=7mA$			0.8	V
输入漏电流	$I_{IL}$	RESET, RA, RB, RC	$V_{IN}=V_{SS}\sim V_{DD}$			$-5\sim +5$	V
	$I_{IL}$	XIN				$-15\sim +15$	
输入上拉电流	$I_{PU}$	RB2, RB3 <sup>3</sup>	$V_{DD}=3V, V_{IN}=V_{SS}$	-120	-80	-50	$\mu A$
电源失败检测电压	$V_{PFD}$	$V_{DD}$		2	3.5	4	V
正常操作电流	$I_{DD}$	$V_{DD}$	$V_{DD}=3V, f_{XIN}=4MHz$		0.3	1	mA
唤醒定时器模式电流	$I_{WKUP}$	$V_{DD}$	$V_{DD}=3V, f_{XIN}=4MHz$		0.3	1	mA
RC-WDT模式电流	$I_{WKUP}$	$V_{DD}$	$V_{DD}=5.5V, f_{XIN}=4MHz$		0.1	0.6	mA
STOP模式电流	$I_{STOP}$	$V_{DD}$	$V_{DD}=3V$		0.01	1	$\mu A$
滞后电压	$V_{T+} \sim V_{T-}$	RESET, RB2, RB3		0.5			V
内部RC振荡周期	$T_{RCWDT}$	$X_{OUT}$	$V_{DD}=5.5V, f_{XIN}=4MHz$	200		450	$\mu S$
RC振荡器频率(系统时钟)	$f_{RCOSC}$	$X_{OUT}$	$R=20K, C=39pF$	200		400	kHz

1. RCO, RC1, RB3 脚只用于GMS87C1202。

2. 典型值一栏为5V, 25°C下的数据. 这些参数未经过测试仅作为设计指南。

3. 当位PUPSELx 被选且设置为输入模式或中断输入功能时该参数无效。

## 10.4. DC 电气特性-GMS87C1102, GMS87C1202(Extended version)

 $(V_{DD}=2.7\sim 6.0V, V_{SS}=0V, f_{XIN}=1MHz\sim 12MHz, T_A=-40^{\circ}C\sim +125^{\circ}C)$ 

参数	符号	管脚 <sup>1</sup>	测试条件	规格			单位
				最小	典型 <sup>2</sup>	最大	
输入高电压	$V_{IH1}$	XIN, RESET	$V_{DD}=4.5\sim 6.0V$	$0.8V_{DD}$		$V_{DD}$	V
	$V_{IH2}$	RB2, RB3		$0.8V_{DD}$		$V_{DD}$	
	$V_{IH3}$	RA, RBO, RB1, RB4, RC		$0.7V_{DD}$		$V_{DD}$	
输入低电压	$V_{IL1}$	XIN, RESET	$V_{DD}=4.5\sim 6.0V$	0		$0.2V_{DD}$	V
	$V_{IL2}$	RB2, RB3		0		$0.2V_{DD}$	
	$V_{IL3}$	RA, RBO, RB1, RB4, RC		0		$0.3V_{DD}$	
输出高电压	$V_{OH}$	RA, RB, RC	$V_{DD}=5V, I_{OH}=-2mA$	$V_{DD}-1$			V
输出低电压	$V_{OL}$	RA, RB, RC	$V_{DD}=5V, I_{OL}=10mA$			1	V
输入漏电流	$I_{IL}$	RESET, RA, RB, RC	$V_{IN}=V_{SS}\sim V_{DD}$			$-5\sim +5$	V
	$I_{IL}$	XIN				$-15\sim +15$	
输入上拉电流	$I_{PU}$	RB2, RB3 <sup>3</sup>	$V_{DD}=5V, V_{IN}=V_{SS}$	-350	-250	-100	uA
电源失败检测电压	$V_{PFD}$	$V_{DD}$		2	3.5	4.2	V
正常操作电流	$I_{DD}$	$V_{DD}$	$V_{DD}=5.5V, f_{XIN}=8MHz$		4	6	mA
			$V_{DD}=5.5V, f_{XIN}=12MHz$		6	10	mA
唤醒定时器模式电流	$I_{WKUP}$	$V_{DD}$	$V_{DD}=5.5V, f_{XIN}=8MHz$		1	1.8	mA
			$V_{DD}=5.5V, f_{XIN}=12MHz$		2	3	mA
RC-WDT模式电流	$I_{RCWDT}$	$V_{DD}$	$V_{DD}=5.5V,$ $f_{XIN}=8MHz/12MHz$		0.3	0.8	mA
STOP模式电流	$I_{STOP}$	$V_{DD}$	$V_{DD}=5.5V$		0.2	2	uA
滞后电压	$V_{T+} \sim V_{T-}$	RESET, RB2, RB3		0.5			V
内部RC振荡周期	$T_{RCWDT}$	XOUT	$V_{DD}=5V,$ $f_{XIN}=8MHz/12MHz$	80		250	uS
RC振荡器频率(系统时钟)	$f_{RCOSC}$	XOUT	R=20K, C=24pF	300		550	kHz

1. RCO, RC1, RB3脚只用于GMS87C1202。

2. 典型值一栏为5V, 25°C下的数据, 这些参数未经过测试仅作为设计指南。

3. 当位PUPSELx 被选且设置为输入模式或中断输入功能时该参数无效。

( $V_{DD}=2.7\sim 6.0V, V_{SS}=0V, f_{XIN}=1MHz\sim 4MHz, T_A=-40^{\circ}C\sim +125^{\circ}C$ )

参数	符号	管脚 <sup>1</sup>	测试条件	规格			单位
				最小	典型 <sup>2</sup>	最大	
输入高电压	$V_{1H1}$	$X_{IN}$	$V_{DD}=2.7\sim 6.0V$	$0.8V_{DD}$		$V_{DD}$	V
	$V_{1H2}$	RESET		$0.9V_{DD}$		$V_{DD}$	
	$V_{1H3}$	RB2, RB3		$0.8V_{DD}$		$V_{DD}$	
	$V_{1H4}$	RA, RBO, RB1, RB4, RC		$0.7V_{DD}$		$V_{DD}$	
输入低电压	$V_{1L1}$	$X_{IN}$	$V_{DD}=2.7\sim 6.0V$	0		$0.2V_{DD}$	V
	$V_{1L2}$	RESET		0		$0.1V_{DD}$	
	$V_{1L3}$	RB2, RB3		0		$0.2V_{DD}$	
	$V_{1L4}$	RA, RBO, RB1, RB4, RC		0		$0.3V_{DD}$	
输出高电压	$V_{OH}$	RA, RB, RC	$V_{DD}=3V, I_{OH}=-2mA$	$V_{DD}-0.7$			V
输出低电压	$V_{OL}$	RA, RB, RC	$V_{DD}=3V, I_{OL}=7mA$			0.8	V
输入漏电流	$I_{IL}$	RESET, RA, RB, RC	$V_{IN}=V_{SS}\sim V_{DD}$			$-5\sim +5$	V
	$I_{IL}$	$X_{IN}$				$-15\sim +15$	
输入上拉电流	$I_{PU}$	RB2, RB3 <sup>3</sup>	$V_{DD}=3V, V_{IN}=V_{SS}$	-120	-80	-50	uA
电源失败检测电压	$V_{PFD}$	$V_{DD}$		2	3.5	4	V
正常操作电流	$I_{DD}$	$V_{DD}$	$V_{DD}=3V, f_{XIN}=4MHz$		1	3	mA
醒定时器模式电	$I_{WKUP}$	$V_{DD}$	$V_{DD}=3V, f_{XIN}=4MHz$		0.3	1	mA
RC-WDT模式电流	$I_{RCWDT}$	$V_{DD}$	$V_{DD}=3V, f_{XIN}=4MHz$		0.1	0.6	mA
STOP模式电流	$I_{STOP}$	$V_{DD}$	$V_{DD}=3V$		0.01	1	uA
滞后电压	$V_{T+} \sim V_{T-}$	RESET, RB2, RB3		0.5			V
内部RC振荡周期	$T_{RCWDT}$	$X_{OUT}$	$V_{DD}=3V, f_{XIN}=4MHz$	200		450	uS
RC振荡器频率 (系统时钟)	$f_{RCOSC}$	$X_{OUT}$	$R=20K, C=39pF$	200		400	kHz

1. RCO, RC1, RB3脚只用于GMS87C1202。

2. 典型值一栏为5V, 25°C下的数据, 这些参数未经过测试仅作为设计指南。

3. 当位PUPSELx 被选且设置为输入模式或中断输入功能时该参数无效。

## 10.5 A/D 转换特性--GMS87C1102E, GMS87C1202E

( $V_{DD}=0V$ ,  $V_{DD}=3.07V/at$ ,  $f_{XIN}=4MHz$ ,  $V_{DD}=5.12V/$  at  $f_{XIN}=8/12MHz$ ,  $T_A=20^{\circ}C\sim+85^{\circ}C$ )

参数	符号	条件	规格			单位
			最小	典型	最大	
模拟输入电压范围	$V_{AIN}$	AVREF=0	$V_{SS}$	-	$V_{DD}$	V
		AVREF=1	$V_{SS}$	-	$V_{REF}$	
模拟电源电压输入范围	$V_{REF}$	AVREF=1	3	-	$V_{DD}$	V
总体精度	$N_{ACC}$		-	$\pm 1.3$	$\pm 1.5$	LSB
非线性误差	$N_{NLE}$		-	-	$\pm 1.2$	LSB
微分非线性误差	$N_{CNLE}$		-	$\pm 1.0$	$\pm 1.5$	LSB
零偏移误差	$N_{ZOE}$		-	$\pm 0.25$	$\pm 1.5$	LSB
满刻度误差	$N_{FSE}$		-	$\pm 1.0$	$\pm 1.5$	LSB
增益误差	$N_{NLE}$		-	-	$\pm 0.5$	LSB
转换时间	$T_{CONV}$	$f_{XIN}=4MHz$	-	-	$\pm 1.5$	$\mu S$
		$f_{XIN}=8MHz$	-	-	20	
		$f_{XIN}=12MHz$	-	-	7	
$A_{VREF}$ 输入电流	$I_{REF}$	$f_{XIN}=4MHz$	-	0.4	0.6	mA
		$f_{XIN}=8MHz$	-	0.5	0.8	
		$f_{XIN}=12MHz$	-	0.6	1	

## 10.6 A/D 转换特性-GMS87C1102E, GMS87C1202E(Extended version)

(V<sub>DD</sub>=0V, V<sub>DD</sub>=35.12V, at f<sub>XIN</sub>=8MHz/12MHz, T<sub>A</sub>=-40°C~+125 °C)

参数	符号	条件	规格			单位
			最小	典型	最大	
模拟输入电压范围	V <sub>AIN</sub>	AVREF=0	V <sub>SS</sub>	-	V <sub>DD</sub>	V
		AVREF=1	V <sub>SS</sub>	-	V <sub>REF</sub>	
模拟电源电压输入范围	V <sub>REF</sub>	AVREF=1	3	-	V <sub>DD</sub>	V
总体精度	N <sub>ACC</sub>		-	±1.3	±2.0	LSB
非线性误差	N <sub>NLE</sub>		-	-	±2.0	LSB
微分非线性误差	N <sub>CNLE</sub>		-	±1.0	±2.0	LSB
零偏移误差	N <sub>ZOE</sub>		-	±1.0	±2.5	LSB
满刻度误差	N <sub>FSE</sub>		-	±0.25	±1.0	LSB
增益误差	N <sub>NLE</sub>		-	±1.0	±2.0	LSB
转换时间	T <sub>CONV</sub>	f <sub>XIN</sub> =8MHz		-	10	uS
		f <sub>XIN</sub> =12MHz	-	-	7	
AVREF输入电流	I <sub>REF</sub>	f <sub>XIN</sub> =8MHz	-	0.5	0.8	mA
		f <sub>XIN</sub> =12MHz	-	0.6	1	
(V <sub>DD</sub> =0V, V <sub>DD</sub> =3.072V, at f <sub>XIN</sub> =4MHz, T <sub>A</sub> =-40°C~+125 °C)						

参数	符号	条件	规格			单位
			最小	典型	最大	
模拟输入电压范围	V <sub>AIN</sub>	AVREF=0	V <sub>SS</sub>	-	V <sub>DD</sub>	V
		AVREF=1	V <sub>SS</sub>	-	V <sub>REF</sub>	
模拟电源电压输入范围	V <sub>REF</sub>	AVREF=1	3	-	V <sub>DD</sub>	V
总体精度	N <sub>ACC</sub>		-	±1.3	±1.5	LSB
非线性误差	N <sub>NLE</sub>		-	-	±1.2	LSB
微分非线性误差	N <sub>CNLE</sub>		-	±1.0	±1.5	LSB
零偏移误差	N <sub>ZOE</sub>		-	±1.0	±1.5	LSB
满刻度误差	N <sub>FSE</sub>		-	±0.25	±0.5	LSB
增益误差	N <sub>NLE</sub>		-	±1.0	±15	LSB
转换时间	T <sub>CONV</sub>	f <sub>XIN</sub> =4MHz	-	-	20	uS
AVREF输入电流	I <sub>REF</sub>	f <sub>XIN</sub> =4MHz	-	0.4	0.6	mA

## 10.7 AC 特性

( $T_A = -20^{\circ}\text{C} \sim +85^{\circ}\text{C}$ ,  $V_{DD} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$ )

参数	符号	条件	规格			单位
			最小	典型	最大	
操作频率	$f_{CP}$	XIN	1	-	12	MHz
外部时钟脉宽	$t_{CPW}$	XIN	80	-	-	nS
外部时钟跳变时间	$t_{RCP}$ , $t_{FCP}$	XIN	-	-	20	nS
外部输入脉宽	$t_{EPW}$	INT0, INT1, ECO	2	-	-	$t_{SYS}$
外部时钟脉冲跳变时间	$t_{REP}$ , $t_{FEP}$	INT0, INT1, ECO	-	-	20	nS
复位信号输入宽度	$t_{RST}$	RESET	8	-	-	$t_{SYS}$

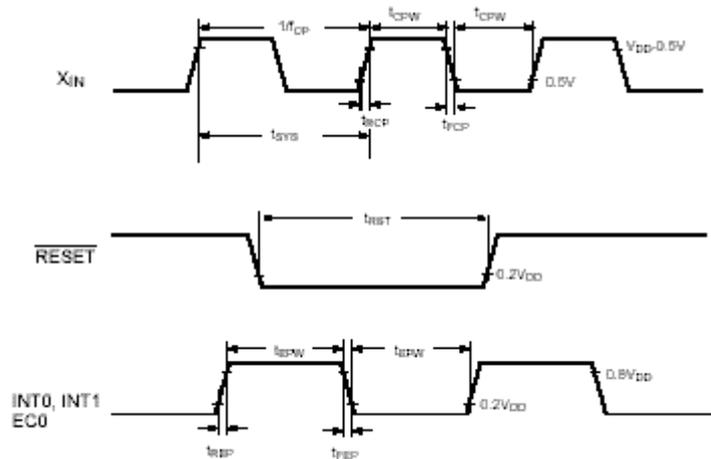


图10-1 时序图

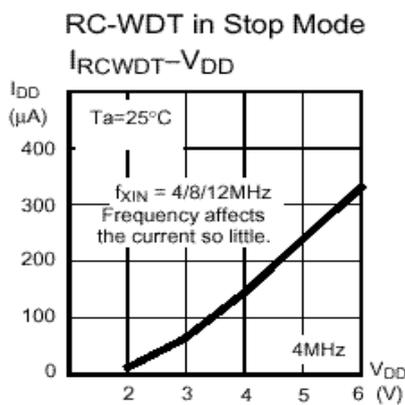
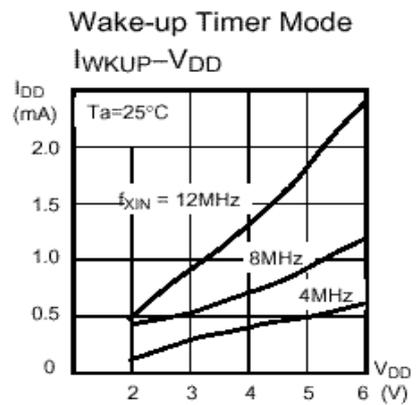
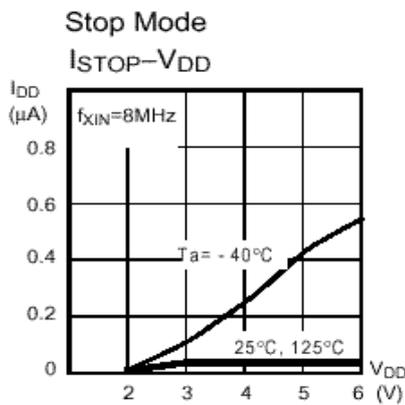
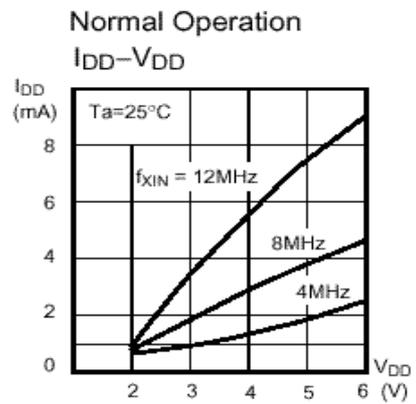
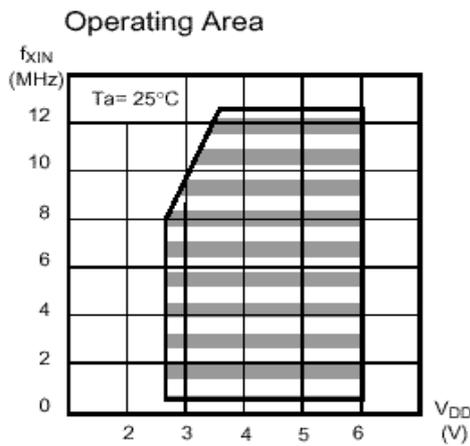
### 10.8 典型特征

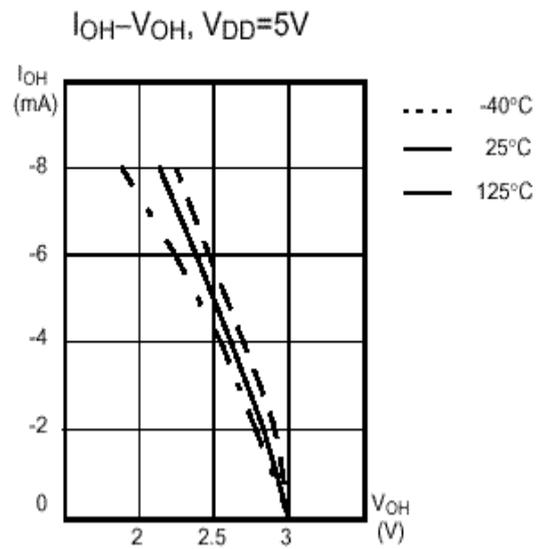
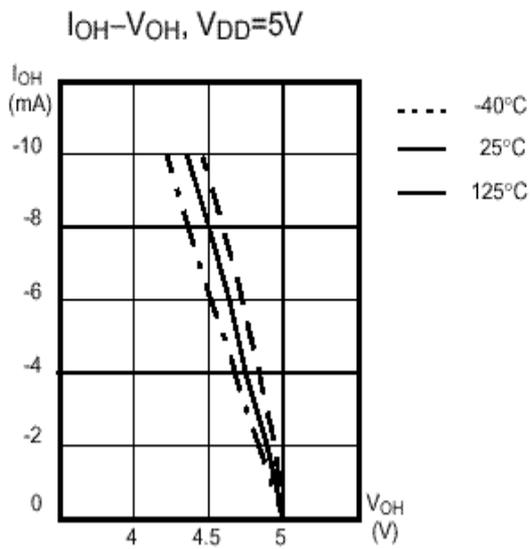
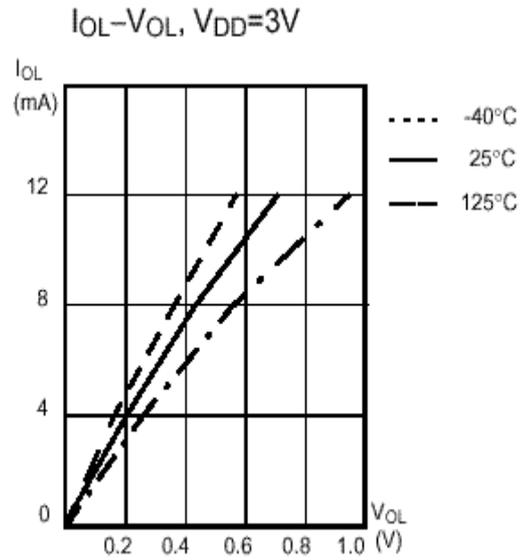
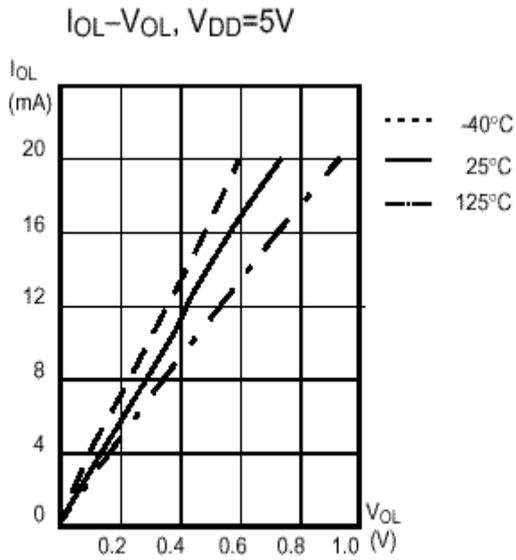
本小节中的图象和表，并未经测试，亦不提供保证，只供设计人员参考。

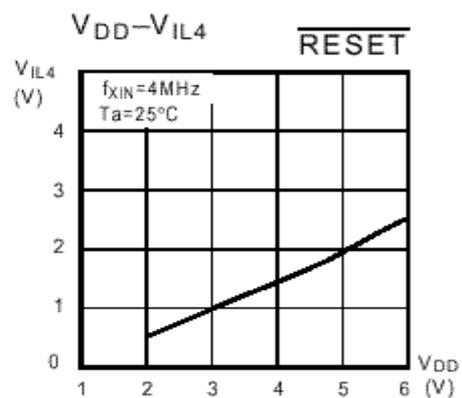
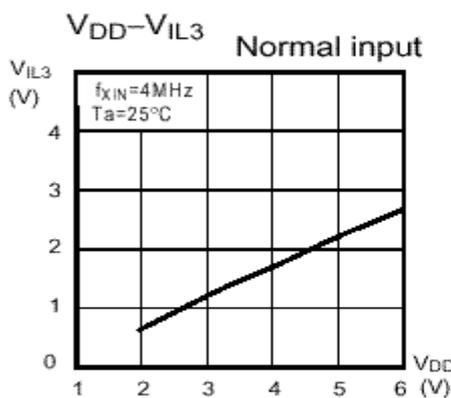
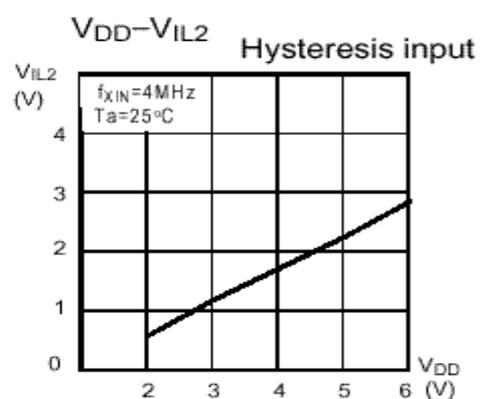
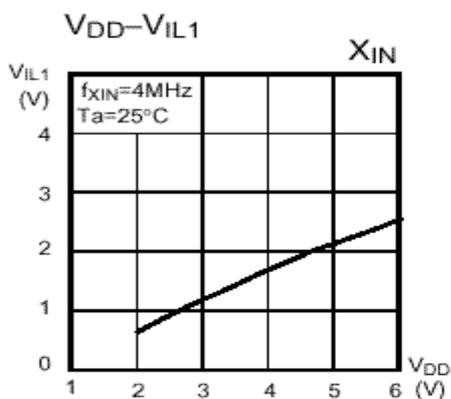
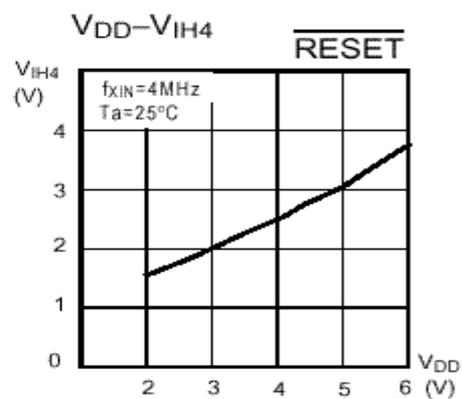
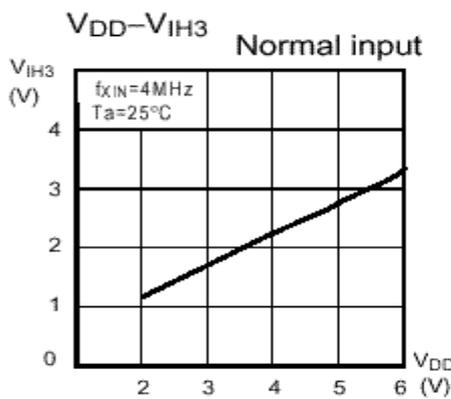
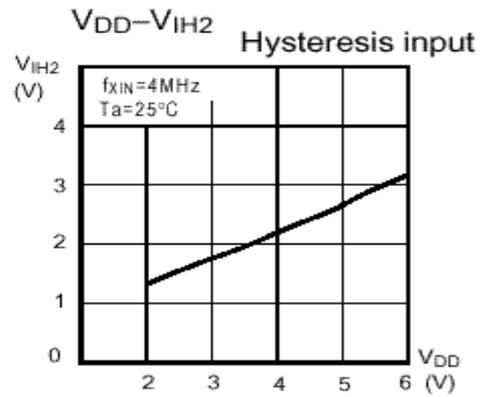
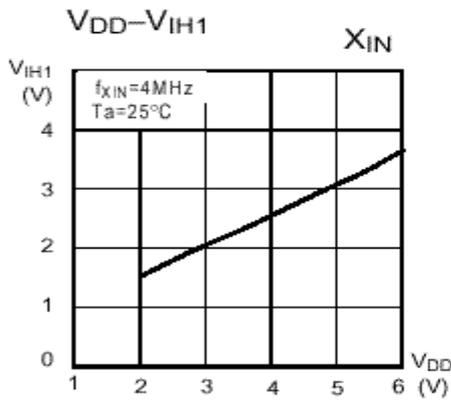
在某些图象和表中，有些数据超出了指定的操作范围 (例如，超出指定的电压范围  $V_{DD}$ )。这只限该信息，而其他器件保证是在指定工作范围内的。

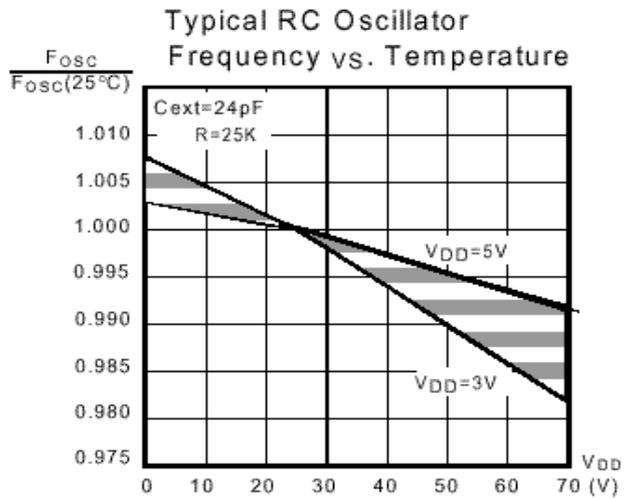
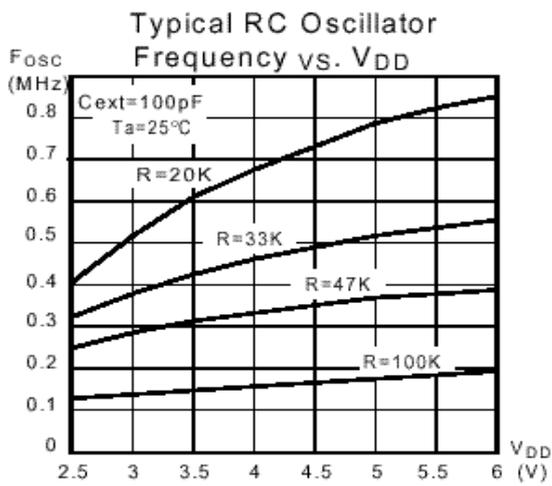
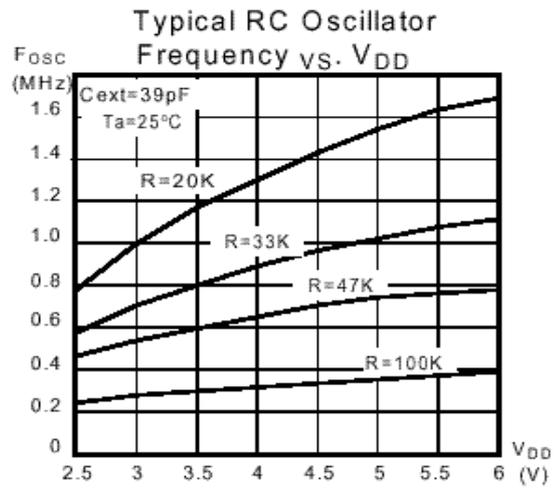
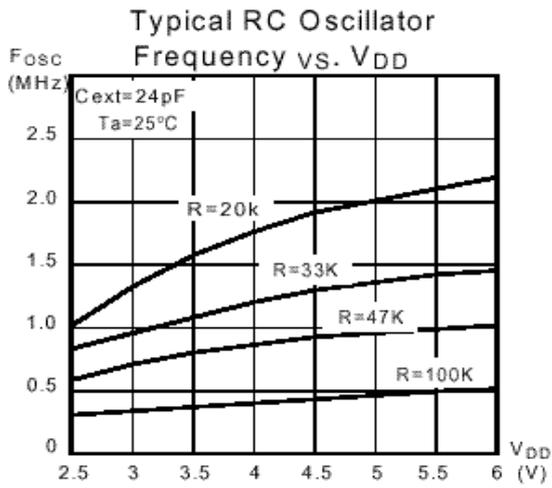
本小节中的数据，是对一个时期内不同批次产品的数据的统计学性概括。

‘典型’表示‘平均值’，而最大或最小分别表示 (平均+3 $\sigma$ )或 (平均-3 $\sigma$ )，其中  $\sigma$ 标准误差。









Cext	Rext	Average	
		Fosc @ 5V,25°C	
24pF	20K	2.02MHz	±14.11%
	33K	1.34MHz	±11.50%
	47K	0.952MHz	±10.30%
	100K	0.48MHz	±9.07%
39pF	20K	1.536MHz	±14.79%
	33K	1.012MHz	±11.67%
	47K	0.72MHz	±10.42%
	100K	0.364MHz	±9.75%
100pF	20K	0.78MHz	±13.53%
	33K	0.512MHz	±10.35%
	47K	0.364MHz	±9.48%
	100K	0.18MHz	±7.34%

表 10-1 RC 振荡器频率

## 11. 存储器结构

GMS87C1202的程序存储器和数据存储器有各自独立的地址空间。程序存储器可读不可写，容量可达2K字节。数据存储器可被读写，包括堆栈区在内共有128字节。

### 11.1 寄存器

GMS87C1202 共有6个寄存器，分别是程序计数器(PC)，累加器(A)，两个变址寄存器(X, Y)，堆栈指针(SP)，程序状态字(PSW)。PC为16位寄存器。

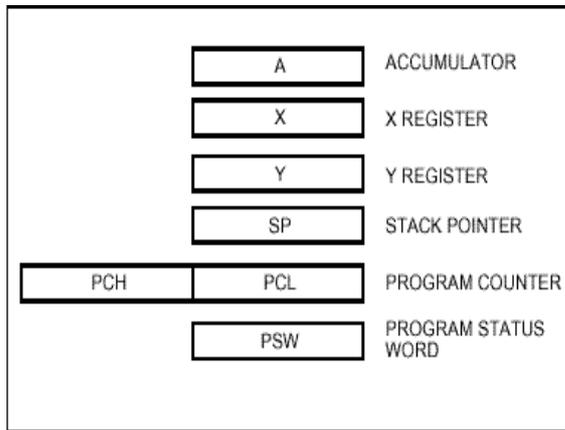


图 11-1 寄存器结构

#### • 累加器

累加器是一个8位通用寄存器，用于对数据的操作。例如，转移，暂存和条件判断，等等。累加器还可和Y寄存器组成一个16位寄存器。

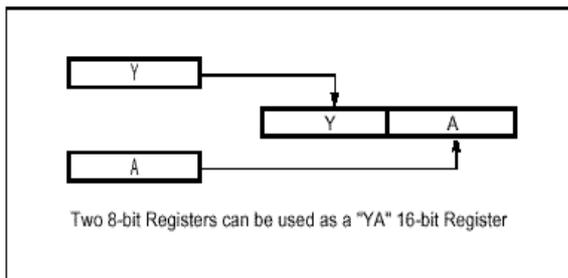


图 11-2 累加器结构

#### • X, Y 寄存器

在用到变址寄存器的寻址方式中，寄存器的内容加上已确定的地址就成为寻址的实际地址。这些寻址方式对访问子程序表和数据表极为有效。

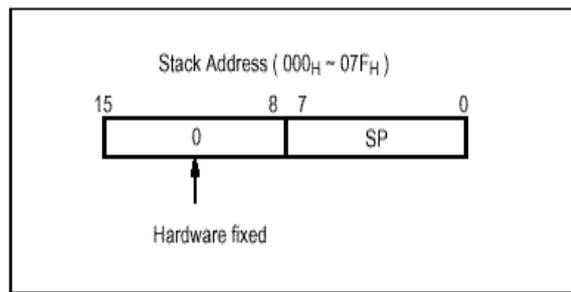
变址寄存器也有加减比较和数据转移功能。它们可作为一个单独的累加器

#### • 堆栈指针

堆栈指针是一个用于中断和子程序调用的8位寄存器。堆栈指针确定堆栈被访问的位置(保存或恢复)。

通常在调用子程序或响应中断时，SP是自动更新的。但是如果堆栈指针超过了分配给堆栈区的地址范围，用户数据可能会因此而丢失。

堆栈可位于内部数据存储区 00<sub>H</sub>~ 7F<sub>H</sub>的任意位置。SP不由硬件初始化，它需要在初始化程序中赋初值(堆栈的起始位置)，通常都将7F<sub>H</sub>为初值。



#### 注：

由于复位(*RESET*)后栈指针之值无定义，故必须通过软件为之初始化。

例如，为初始化SP，

```
LDX #07FH
```

```
TXSP ; SP<-7FH
```

#### • 程序计数器

程序计数器是一个包含两个8位寄存器PCH，和PCL的16位寄存器。计数器指向下一条要执行指令的地址。复位状态时，PC值为复位程序地址(PC<sub>H</sub>: 0FF<sub>H</sub>，PCL: 0FE<sub>H</sub>)。

#### • 程序状态字

程序状态字(PSW)包含几个反映CPU当前状态的位。其中包括负数标志，溢出标志，Break标志，半进位标志(用于BCD码操作)作中断，使能标志，零标志和进位标志。

### 【进位标志 C】

本标志保存算术运算后CPU的ALU的任何进位或借位。它也可被移位或循环移位指令所改变。

### 【零标志 Z】

当算术运算或数据转移的结果为零时本位被置位，否则清零。

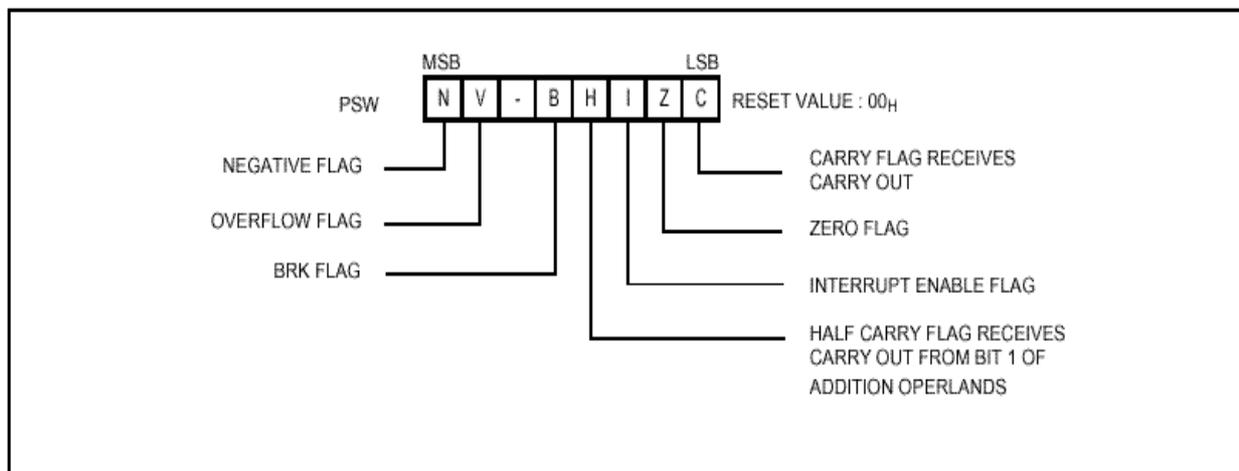


图 11—3 PSW( 程序状态字)寄存器

### [中断禁能标志 I]

本位使能/禁能 所有中断，复位或软件，BRK 指令引起的中断除外。当该位清零时，所有中断被禁能。进入中断后，本位立即被置为0。本位通过EI 指令置位，DI 指令清零。

### [半进位标志 H]

进行操作后，当ALU的位4 无借位而位3 有进位时 H被置位。除了CLR V指令外，其它指令不能将本标志置位或清零。

### [Break 标志 B]

本标志由BRK指令置位，以区别于位于同一个向量地址的TALL 指令。

### [溢出标志 V]

运算结果发生溢出时，本位被置为 1。加减法运算结果超过-127( 7F<sub>H</sub>)+127( 80<sub>H</sub>) 范围时发生溢出。溢出标志由 CLR V指令清零，没有置位指令。执行BIT 指令时，操作数位 6 的值被赋给V。

### [负数标志 N]

本标志的状态与数据的或算术运算结果的符号位(位 7) 一致。执行BIT 指令时，操作数位7的值被赋给此标志。

### 11.2 . 程序存储器

16位的程序计数器可寻址64K 字节，但该器件在物理上只有2K 字节的程序存储空间。访问的地址如果超过FFFF<sub>H</sub>，将会使PC翻转到0000<sub>H</sub>。

图11-5所示为程序存储区最上部分。复位后，CPU从存放在地址 FFFE<sub>H</sub>， FFFF<sub>H</sub> 处的复位向量处开始执行程序。

如图11-5. 所示，程序存储器内的每个区域都被分配了一个固定的位置。程序存储区包含用户程序，页调用( TCALL )区域包含子程序，由于用2字节PCALL指令代替了3 字节CALL 指令，减少了程序字节的长度。如果频繁调用子程序，缩短程序长度会更有用。

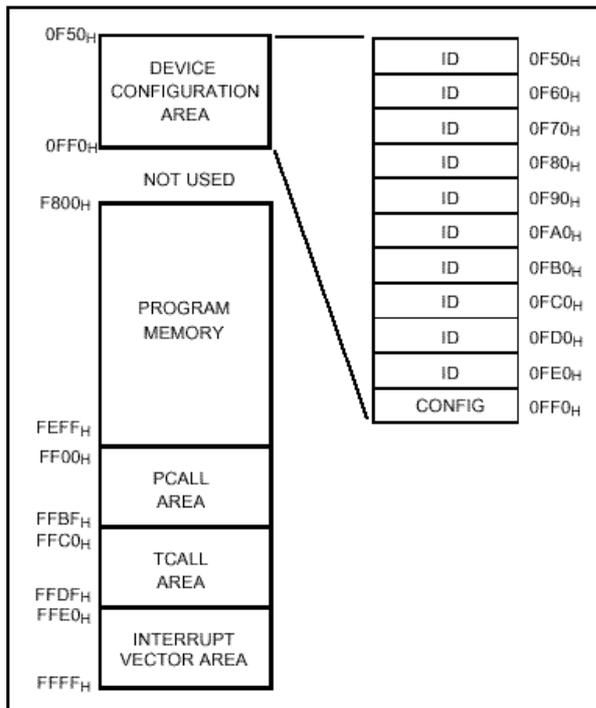


图 11-4 程序存储器图

器件配置区域可编程或保留，以用于选择器件的配置，如，RC器选项。本区域在程序正常执行时不可访问，但在编程/校验时可被读写。

详细请见器件配置区域章节。

表调用( TCALL)使CPU跳到每个TCALL地址，开始服务程序的执行。TCALL地址占用2 个字节宽度：FFC0<sub>H</sub> 用于TCALL15,FFC2<sub>H</sub> 用于TCALL14，等。

中断使CPU跳到特定的地址，开始执行中断服务程序。例如，对外部中断 0，被赋地址FFFA<sub>H</sub>。中断向量占用2 个字节宽：例如，对外部中断 1，为FFF8<sub>H</sub>，外部中断 0，为FFFA<sub>H</sub>，等。

地址	TCALL 名称
FFC0H	TCALL15
FFC2H	TCALL14
FFC4H	TCALL13
FFC6H	TCALL12
FFC8H	TCALL11
FFCAH	TCALL10
FFCCH	TCALL9
FFCEH	TCALL8
FFD0H	TCALL7
FFD2H	TCALL6
FFD4H	TCALL5
FFD6H	TCALL4
FFD8H	TCALL3
FFDAH	TCALL2
FFDCH	TCALL1
FFDEH	TCALL0/BRK <sup>1</sup>

表 11-1 TCALL 向量

从地址FF00<sub>H</sub> 起到地址FFFF<sub>H</sub> 区域，如果其中有闲置没用的，则可用作通常的程序存储区。

地址	向量名称
FFC0H	未使用
FFC2H	未使用
FFC4H	未使用
FFC6H	基本间隔定时器
FFC8H	看门狗定时器
FFCAH	A/D 转换器
FFCCH	未使用
FFCEH	未使用
FFD0H	未使用
FFD2H	未使用
FFD4H	定时器/计数器 1
FFD6H	定时器/计数器 0
FFD8H	外部中断 1
FFDAH	外部中断 0
FFDCH	未使用
FFDEH	复位

表 11-2 中断向量

页调用 (TCALL) 区域包含子程序，由于用 2 字节 PCALL 指令代替了 3 字节 CALL 指令，减少了程序字节的长度。如果频繁调用子程序，缩短程序字节长度会更有用。

表调用 (TCALL) 使 CPU 跳到各个 TCALL 地址，开始执行服务子程序。表调用服务区的各个 TCALL 为 2 个字节宽度：例如，对 TCALL15 是 0FFC0<sub>H</sub>，TCALL14，是 0FFC2<sub>H</sub>，等，如图 11-5 所示。

例：TCALL 的使用

```

LDA    #5
TCALL  0FH
:
:
;
;TABLE CALL ROUTINE
;
FUNC_A: LDA  LRG0
        RET
;
FUNC_B: LDA  LRG1
        RET
;
;TABLE CALL ADD. AREA
;
        ORG  0FFC0H
        DW   FUNC_A
        DW   FUNC_B
    
```

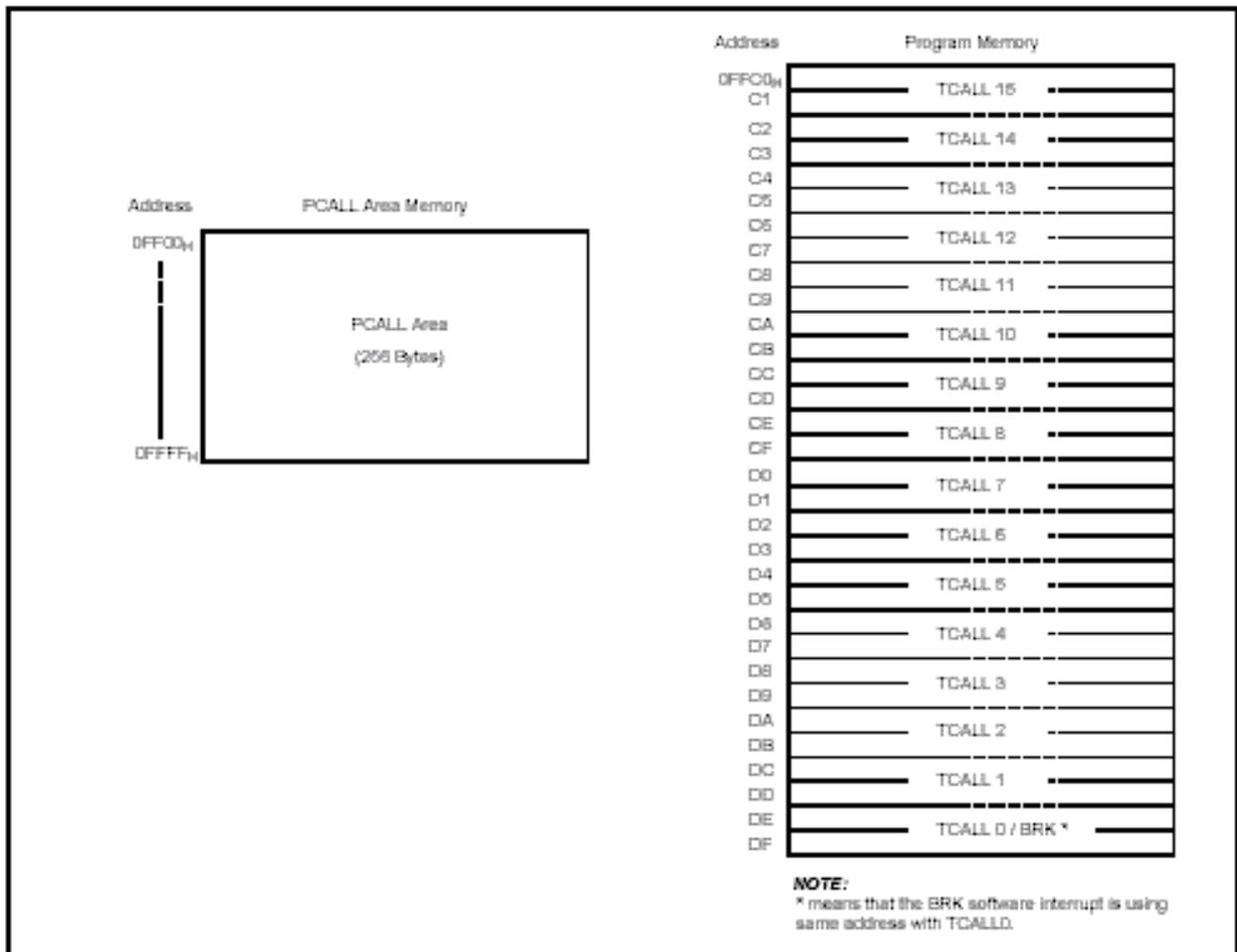
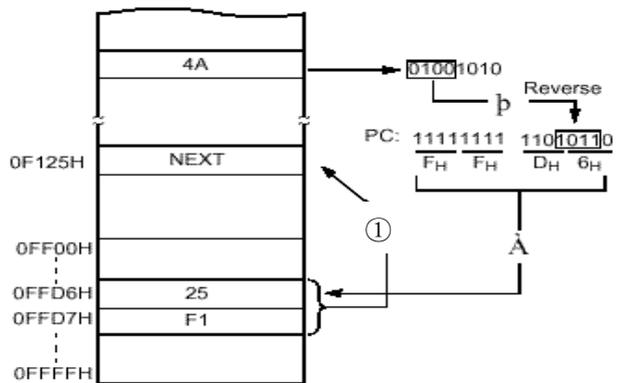
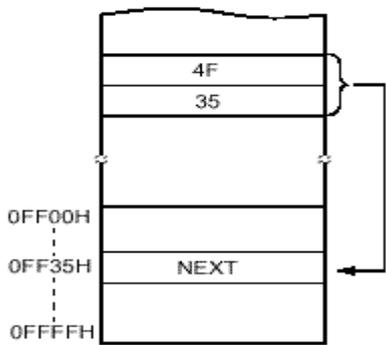


图 11-5 PCALL 和 TCALL 存储区

PCALL → rel  
4F35 PCALL 35H

TCALL → n  
4A TCALL 4



例：向量地址和初始化部分的软件示例.

```

ORG          0FFE0H

DW          NOT_USED      ;(0FFE0)
DW          NOT_USED      ;(0FFE2)
DW          NOT_USED      ;(0FFE4)
DW          BIT_INT       ;(0FFE6) 基本间隔定时器
DW          WDT_INT       ;(0FFE8) 看门狗定时器
DW          AD_INT        ;(0FFEA) A/D
DW          NOT_USED      ;(0FFEC)
DW          NOT_USED      ;(0FFEE)
DW          NOT_USED      ;(0FFF0)
DW          NOT_USED      ;(0FFF2)
DW          TMR1_INT      ;(0FFF4) 定时器1
DW          TMR0_INT      ;(0FFF6) 定时器0
DW          INT1          ;(0FFF8) 外部中断1
DW          INT0          ;(0FFFA) 外部中断0
DW          NOT_USED      ;(0FFFC)
DW          NOT_USED      ;(0FFFE) 复位

ORG          0F800H

;-----
;          主程序
;-----

RESET:      DI            ;Disable All Interrupts
            LDX          #0
RAM_CLR:    LDA          #0 ;Disable All Interrupts
            STA          {X}+
            CMPX         #080H
            BNE          RAM_CLR
;
            LDX          #07FH ;Stack Pointer Initialize
            TXSP
;
            CALL         INITIAL ;
    
```

```

;
LDM          RA, #0           ;Normal Port    A
LDM          RAI0,#1000_0010B ;Normal Port    Direction
LDM          RB, #0           ;Normal Port
LDM          RBIO,#1000_0010B ;Normal Port    Direction
:
:
LDM          PFDR,#0         ;Enable Power Fai
:

```

### 11.3 数据存储

图11-6所示为可访问的内部数据存储空间。数据存储分为两个部分，用户RAM(包括堆栈)和控制寄存器。

内部数据存储区地址宽度为1个字节，这意味着包括堆栈区在内共有128个字节。

由于堆栈指针的值在复位后不定，所以应当通过软件将其初始化为 $00_H$ 到 $7F_H$ 间的某个值。

堆栈区定义在数据存储区内，因此堆栈不能和RAM数据区重叠。例如，假设堆栈指针为 $6F$ ，如果程序访问该地址将堆栈值改变，就有可能发生故障。

控制寄存器供CPU与外围功能用来控制器件,实现所期望的操作。这些寄存器包含控制和状态位,用于中断系统定时/计数器, A/D 转换器和I/O口。控制寄存器位于地址 $0C0_H$ 至 $0FF_H$ 之间。

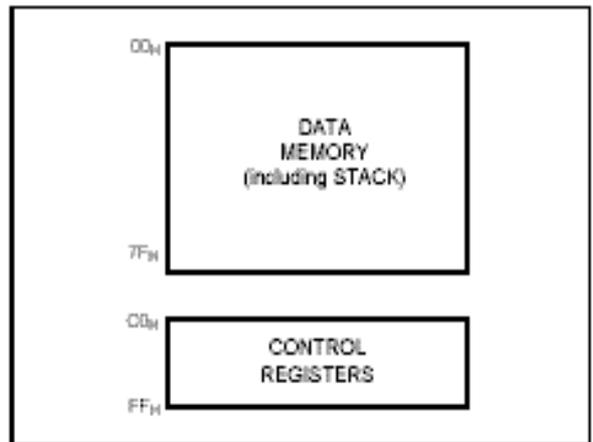


图 11-6 数据存储图

要注意的是，未定义的地址可能在芯片上并不存在。对这些地址行读操作，将会得到随机的数据。写操作则不会产生确定的结果，

更多的细节详见各外围功能章节。

地址	符号	RW	复位值
C0H	RA	R/W	未定义
C1H	RAIO	W	0000_0000
C2H	RB	R/W	未定义
C3H	RBIO	W	---0_0000
C4H	RC	R/W	未定义
C5H	RCIO	W	----_--00
CAH	RAFUNC	W	0000_0000
CBH	RBFUNC	W	---0_0000
CCH	PUPSEL	W	----_--00
D0H	TM0	R/W	--00_--00
D1H	T0	R	0000_0000
D1H	TDR0	W	1111_1111
D1H	CDR0	R	0000_0000
D2H	TM1	R/W	0000_0000
D3H	TDR1	W	1111_1111
D3H	T1PPR	W	1111_1111
D4H	T1	R	0000_0000
D4H	CDR1	R	0000_0000
D4H	T1PDR	R/W	0000_0000
D5H	PWMHR	W	----_0000
DEH	BUR	W	1111_1111
E2H	IENH	R/W	0000_----
E3H	IENL	R/W	000_----
E4H	IRQH	R/W	0000_----
E5H	IRQL	R/W	000_----
E6H	IEDS	R/W	----_0000
EAH	ADCM	R/W	--00_0001
EBH	ADCR	R	未定义
ECH	BITR	R	0000_0000
ECH	CKCTRL	W	-001_0111
EDH	WDTR	R/W	0111_1111
EFH	PFDR	R/W	----_100

表 11—3 控制寄存器的复位值

## 堆栈区

堆栈提供调用子程序或响应中断之前用于保存返回地址的空间。

从当前处理的程序返回时，执行子程序返回指令[RET]，从堆栈中恢复程序计数器的内容，执行中断返回指令[RETI]，恢复程序计数器和标志位的内容。

堆栈中保存/恢复的位置由堆栈指针(SP)决定，SP在保存/恢复时自动减少/增加，这意味着SP的值指向下次要保存的堆栈位置。

地址	读			写	
	定时器模式	捕捉模式	PWM模式	定时器模式	PWM模式
D1H	T0	CDR0	—	TDR0	—
D3H	—			TDR1	T1PPR
D4H	T1	CDR1	T1PDR		T1PDR

表 11—4 位于同一地址的不同寄存器名

**注：**只写寄存器不能通过位操作指令进行访问  
不应使用读-修改-写指令应使用字节操作指令

**例：**写寄存器 CKCTRL

*LDM CKCTRL, #9H ;分频比率 +16*

地址	名称	位7	位6	位5	位4	位3	位2	位1	位0
C0H	RA	RA口数据寄存器							
C1H	RAIO	RA口方向寄存器							
C2H	RB	RA口数据寄存器							
C3H	RBIO	RA口方向寄存器							
C4H	RC	RA口数据寄存器							
C5H	RCIO	RA口方向寄存器							
CAH	RAFUNC	ANSEL7	ANSEL7	ANSEL7	ANSEL7	ANSEL7	ANSEL7	ANSEL7	ANSEL7
CBH	RBFUNC	-	-	-	PWM0	INT1I	INT0I	BUZO	AVREFS
CCH	PUPSEL	-	-	-	-	-	-	PUPSEL1	PUPSEL0
D0H	TM0	-	-	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST
D1H	T0/TDRO/ CDR0	定时器0 寄存器							
D2H	TM1	POL	16BIT	PWME	CAP1	T1CK1	T1CK0	T1CN	T1ST
D3H	TDR1/ T1PPR	定时器数据寄存器 1/PWM周期寄存器 1							
D4H	T1/XDR1/ T1PPR	定时器 1 寄存器/捕捉 数据寄存器 1/PWM占空							
D5H	PWMHR	PWM高位寄存器							
DEH	BUR	BUCK1	BUCK0	BUR5	BUR4	BUR3	BUR2	BUR1	BUR0
E2H	IENH	INT0E	INT1E	T0E	T1E	-	-	-	-
E3H	IENL	ADE	WDTE	BITE	-	-	-	-	-
E4H	IRQH	INT0IF	INT1IF	T0IF	T1IF	-	-	-	-
E5H	IRQL	ADIF	WDTIF	BITIF	-	-	-	-	-
E6H	IEDS	-	-	-	-	IED1H	IED1L	IED0H	IED0L
EAH	ADCM	-	-	ADEN	ADS2	ADS1	ADS0	ADST	ADSF
EBH	ADCR	ADV结果寄存器							
ECH	BITR1	基本间隔定时器寄存器数据寄存器							
ECH	CKCTLR	-	WAKEUP	RCWDT	WDTON	BTCL	BTS2	BTS1	BTS0
EDH	WDTR	WDTCL	7位看门狗计数寄存器						
EFH	PFDR2	-	-	-	-	-	PFDIS	PFDM	PFDS

表 11-5 GMS87C1202 控制寄存器

阴影区域的寄存器不可位寻址，只可由字节操作指令寻址，如，“SET1,CLR1”，故应通过寄存器操作指令，象“LDM dp, #imm”指令访问。

- 1, 寄存器 BITR 和 CKCTLR 位于同一个地址。地址 ECH 被读为 BITR，被写为 CKCTLR。
- 2, 寄存器 PFDR 仅在器件中被处理，在仿真器中则不能。

### 11.4 寻址方式

GMS87C1102/1202使用以下6种寻址方式。

- 寄存器寻址
- 立即数寻址
- 直接页寻址
- 绝对寻址
- 变址寻址
- 寄存器间接寻址

下面分别举例加以说明。

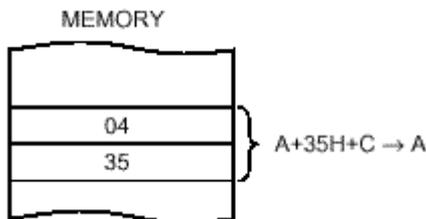
#### (1) 寄存器寻址

寄存器寻址访问 A, X, Y, C 和 PSW。

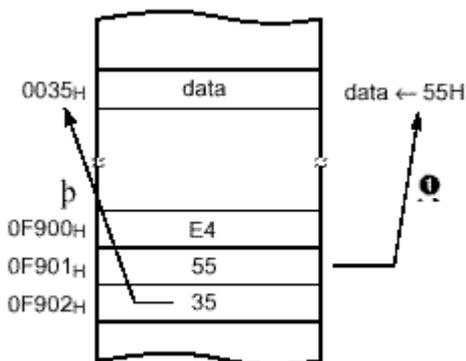
#### (2) 立即数寻址 → #imm

在本方式中，第二字节(操作数)作为一个立即数被访问。

例： 0435 ADC #35H



E45535 LDM 35H #55H

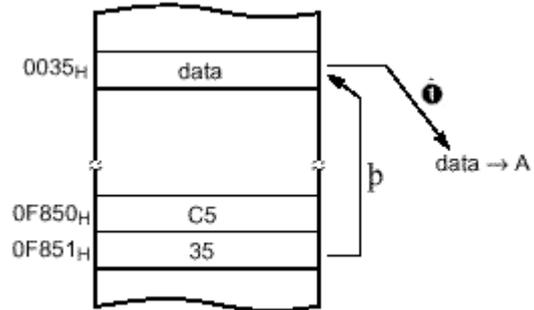


#### (3) 直接页寻址 → dp

在本方式中，地址指定在直接页内。

例:

C535 LDA 35H ; A ← RAM[35H]



#### (4) 绝对寻址 → !abs

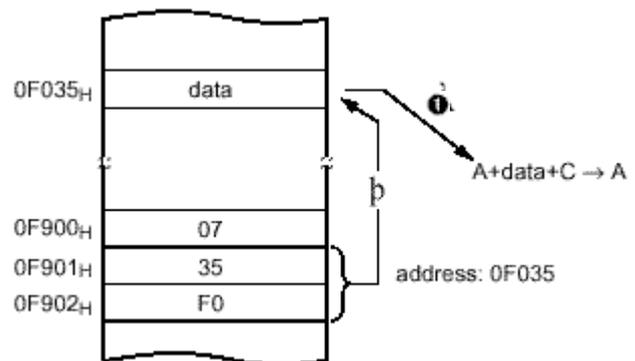
绝对寻址将对应的存储器数据设为数据。即，命令的第二字节(操作数 I), 成为地址低位, 第三字节(操作数 II) 成为地址高位。

三字节指令可对整个存储区进行访问。

ADC, AND, CMP, CMPX, CMPY, EOR, LDA, LDX, LDY, OR, SBC, STA, STX, STY

例:

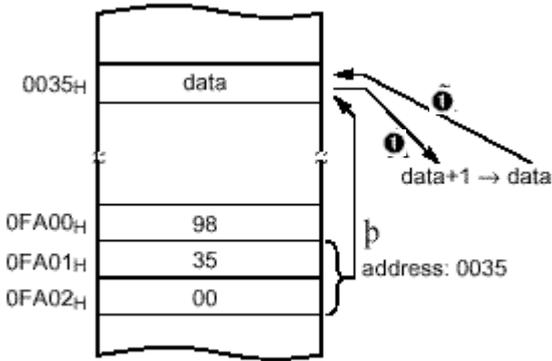
0735F0 ADC !0F035H ; A ← ROM[0F035H]



数据存储区(RAM)内的操作

ASL BIT DEC INC LSR ROL ROR

例: 983500 INC 0035H ; A ← RAM[035H]



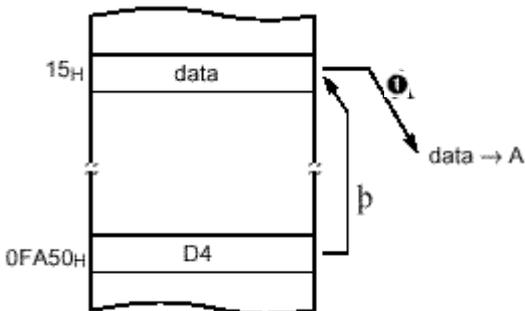
(5) 变址寻址

X 变址直接页寻址(无偏移量) → {X}

在本方式中, 地址由X寄存器指定。

ADC, AND, CMP, EOR, LDA, 0R, SBC, STA, XMA

例: X=15H  
D4 LDA {X} ; ACC ← RAM[X]

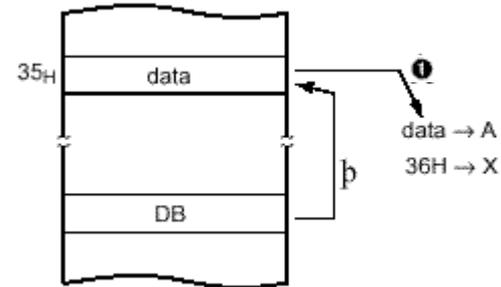


X 变址直接页寻址, 自动增加 → {X}+

在本方式中, 地址由X寄存器指定X 的内容加 1。

LDA, STA

例: X=35H  
DB LDA {X}+

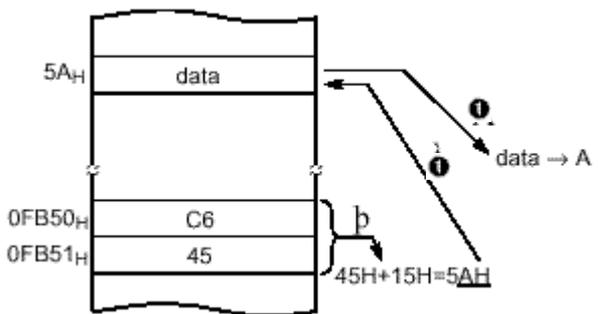


X 变址直接页寻址(8 位偏移量) → dp+X

本地址值为命令第二字节(操作数)加上X寄存器的内容。它位于直接页内的存储单元。

ADC, AND, CMP, EOR, LDA, LDY, OR, SBC, STA, STY, XMA, ASL, DEC, INC, LSR, ROL, ROR

例: X=0 15H  
C645 LDA 45H+X



**Y 变址直接页寻址(8 位偏移量) → dp+Y**

本地址值为命令第二字节(操作数)加上X寄存器的内容,它位于直接页内的存储单元。

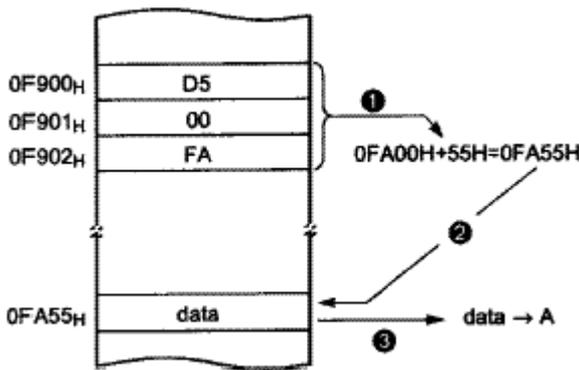
与上面的项相(2)同,用寄存器代替X。

**Y 变址绝对寻址 →!abs+Y**

16位绝对地址值加上Y寄存器值,作为寻址地址。本寻址方式可在整个存储区内实现。

例: Y=55<sub>H</sub>

```
D500FA LDA !0FA00H+Y
```



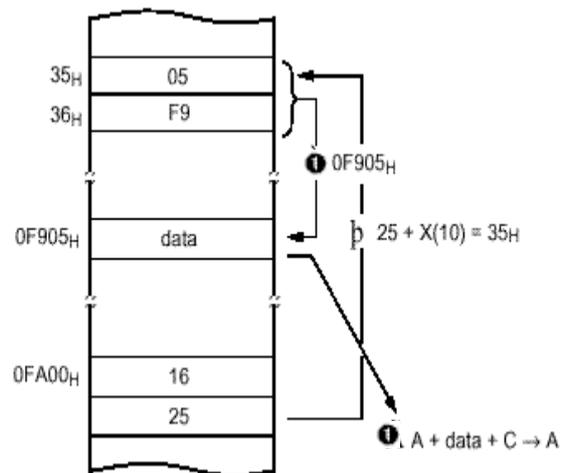
**X 变址间接寻址 →[dp+ X]**

过程存储区数据就象普通数据,由一对16位存储单元指定,该存储单元由数据对[dp+X+1][dp+X]操作数加上直接页中的X寄存器数据确定。

**ADC,AND,CMP,EOR,LDA,OR,SBC,STA**

例: X=10<sub>H</sub>

```
1625 ADC [25H+X]
```



**(6)间接寻址**

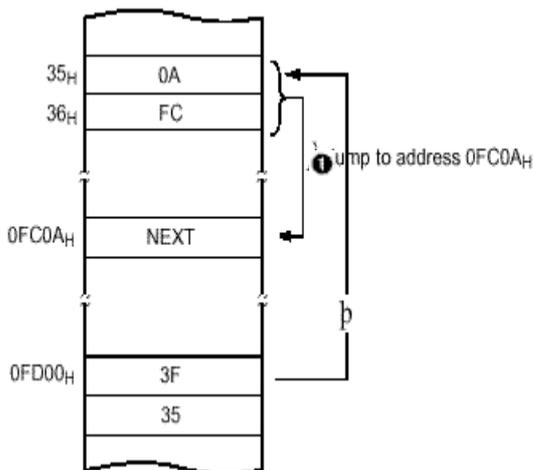
**直接页间接寻址 →[dp]**

设置数据地址,以便用于操作数来完成一条用于设置存储区数据(或一双存储单元)的指令。

变址寄存器 X, Y可用于变址。

**JMP,CALL**

例: 3F35 JMP [35H]



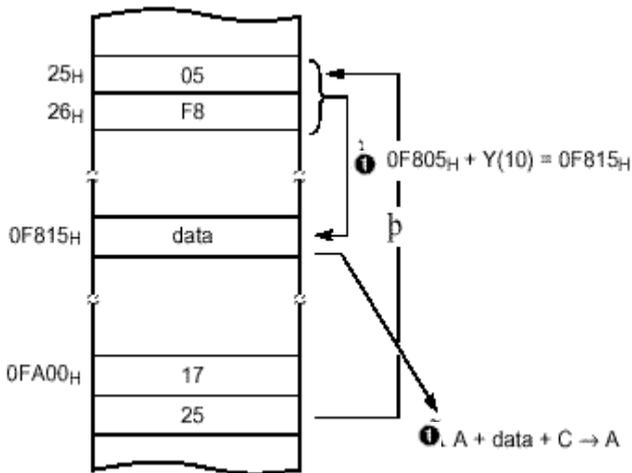
**Y 变址间接寻址  $\rightarrow [dp]+Y$**

象存储区数据那样处理普通数据，数据由一对16位存储单元  $[dp+1][dp]$ 指定，该存储单元由直接页加上Y寄存器数据确定。

ADC, AND, CMP, EOR, LDA, OR, SBC, STA

例:  $Y=10H$

```
1725 ADC [25H]+Y
```

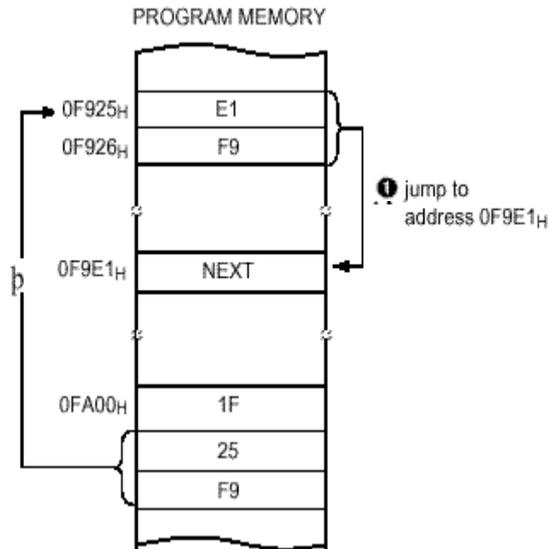


**绝对间接寻址  $\rightarrow [!abs]$**

程序跳转到指定的16位绝对地址处。

JMP

```
例: 1F25F9 JMP [!0F925H]
```



## 12. I/O 口

GMS87C1202有三个I/O口，即 RA, RB和RC。这些口具有多个可选择的功能，用于器件的外围功能。通常在初始复位状态时，所有口都作为通用的输入口。

所有口都有数据方向寄存器，用来将口设为输入或输出。口方向寄存器置 1时，相应的管脚作为输出。相反，向相应位写入0 定义口即为输入。例如，将RA的偶数位设为输出，奇数位设为输入，就在初始化设定时将55H 写入地址C1HRA。

方向寄存器如图12—1所示。

### 12. 1 RA 和RAIO 寄存器

RA 是一个8 位双向I/O 口( 地址C0<sub>H</sub>)，每个口都可通过RAIO寄存器地址C1<sub>H</sub>单独设为输入或输出。RA7~RA1还可作为模拟输入口( AN7~AN1)，RA0 口可作为事件计数器输入口( ECO)。

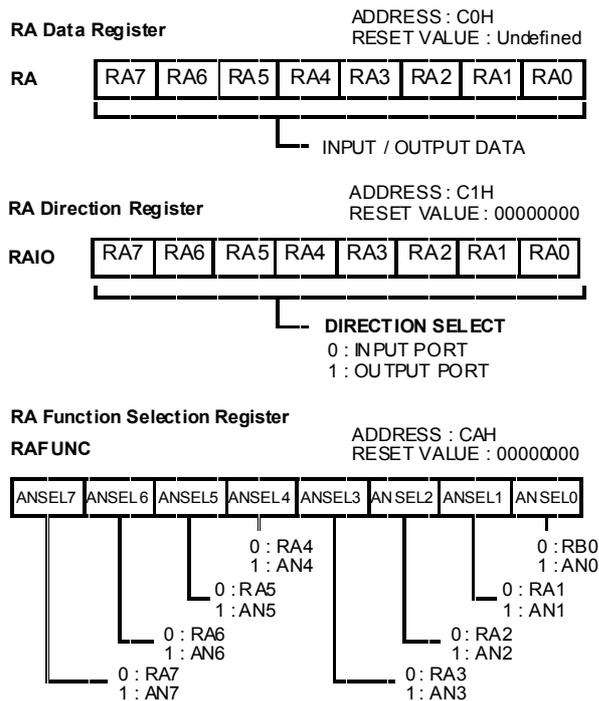


图 12—2 RA口寄存器

控制寄存器RAFUNC( 地址CAH )控制第二功能的选择。复位后该寄存器值为0，口作为通用I/O口。如要选择第二功能，如模拟输入或外部事件计数输入，将RAFUNC相应位置1。不论口方向寄存器RAIO 如何，RAFUNC将被用作功能选择，口可被用作相应的特色( RA0/ECO则由RBFUNC控制)。

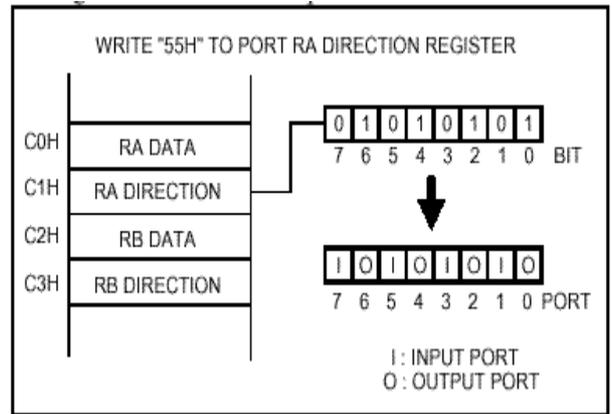


图12—1 方向寄存器

口	RAFUNC. 7 ~0	描述
RA7/AN7	0	RA7 (普通I/O口)
	1	AN7 (ADS2~0=111)
RA6/AN6	0	RA6 (普通I/O口)
	1	AN6 (ADS2~0=110)
RA5/AN5	0	RA5 (普通I/O口)
	1	AN5 (ADS2~0=101)
RA4/AN4	0	RA4 (普通I/O口)
	1	AN4 (ADS2~0=100)
RA3/AN3	0	RA3 (普通I/O口)
	1	AN3 (ADS2~0=011)
RA2/AN2	0	RA2 (普通I/O口)
	1	AN2 (ADS2~0=010)
RA1/AN1	0	RA1 (普通I/O口)
	1	AN1 (ADS2~0=001)
RA0/ECO <sup>1</sup>	0	RA0 (普通I/O口)
	1	ECO (ADS2~0=111)

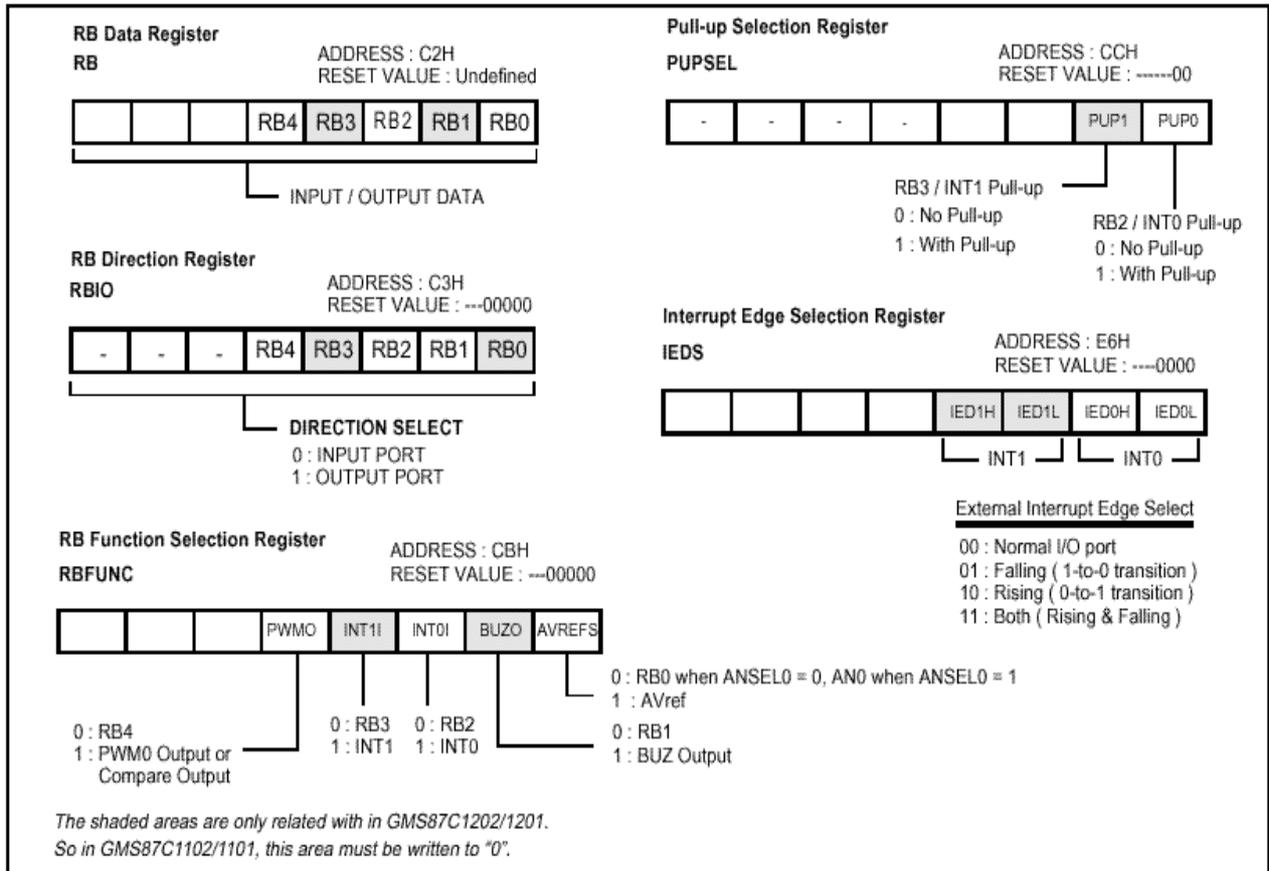
1. 本口不是一个模拟输入口，而是事件计数器时钟源输入口。ECO 通过设置TOCK2 0=111 进行控制。

位RAFUNC. 0 (ANSEL0)控制RBO/ANO/AVref 口( 参看RB口)。

### 12.2 RB 和RBIO 寄存器

RB是一个5位双向I/O口(地址C2<sub>H</sub>)。每个口都可通过RBIO寄存(地址C3<sub>H</sub>)单独设为输入或输出。RB口还可用于几个专门功能。控制寄存器RBFUNC(地址CB<sub>H</sub>)控制第二功能的选择。复位后该寄存器值为0,口作为通用I/O口。如要选择第二功能,如外部中断或定时器比较输出,则应将RBFUNC相应位置1。

RBFUNC 的功能选择与RBIO 寄存器无关。



12-3 RB口寄存器

口	RBFUNC. 4~0	描述
RB4/ PWM0/ COMP0	0	RB4 (普通I/O口)
	1	PWM0输出定时器1比较输出
RB3/INT1	0	RB3 (普通I/O口)
	1	外部中断输入INT1
RB2/INT0	0	RB2 (普通I/O口)
	1	外部中断输入INT0
RB1/BUZ	0	RB1 (普通I/O口)
	1	蜂鸣器输出
RB0/ AN0/ Avref	0	RB0 (普通I/O口)/AN0=(ANSEL0=1)
	1	外部模拟参考电压

1. 当ANSEL=0时,本管脚定义为普通I/O口(RB0)。

当ANSEL=1,且ADS2~0=000时,本脚可作为模拟输入口(AN0)。

2 当本位置1时,口定义为AVref,不可用作模拟输入口AN0和普通I/O口(RB0)。

### 12.3 RC和RCIO寄存器

RC是一个4位双向I/O口(地址C4<sub>H</sub>)。每个口都可通过RBIO寄存器(地址C5<sub>H</sub>)单独设为输入或输出口。

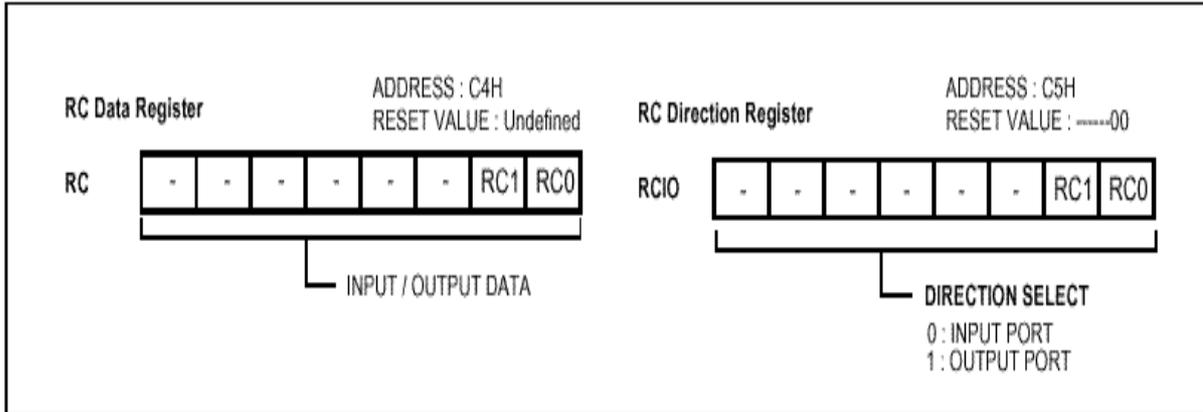


图12-4 RC口寄存器

### 13. 时钟发生器

时钟发生器产生基本的时钟脉冲，给CPU和外围电路提供系统时钟。主系统时钟振荡器通过连接在 $X_{IN}$ 和 $X_{OUT}$ 两脚间的晶振或陶瓷振荡器产生振荡。外部时钟可输入到主系统时钟振荡器。这样，时钟信号从 $X_{IN}$ 脚输入 $X_{OUT}$ 开路。

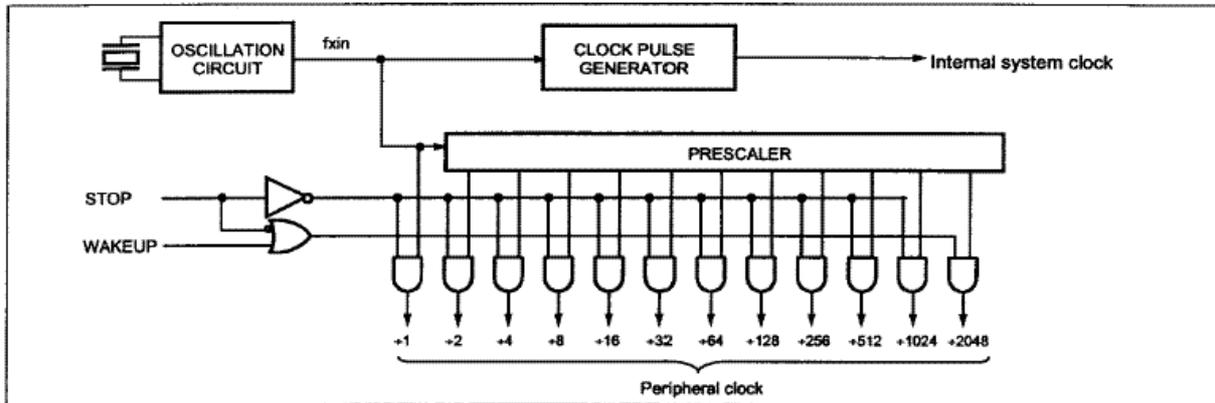


图 13-1 时钟脉冲发生器方框图

#### 13.1 振荡器电路

$X_{IN}$  和  $X_{OUT}$  分别作为输入和输出，反相放大器可设置作为片内振荡器,如图13-2所示。

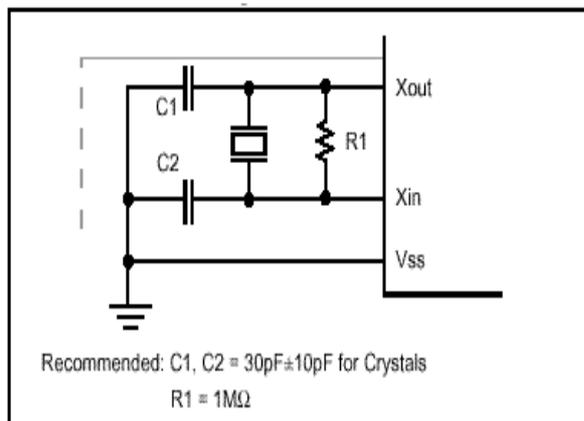


图 13-2 振荡器的连接

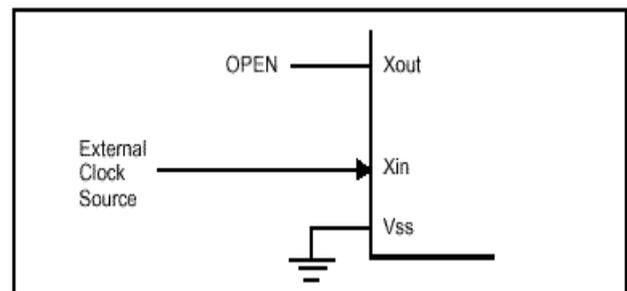


图 13-3 外部时钟的连接

使用外部时钟源驱动器件时， $X_{OUT}$  应当开路用  $X_{IN}$  作为驱动输入，如图13-3。对外部时钟信号无占空比的要求，因为时钟通过触发器二分频输入到内部时钟电路。但高低电平的最小和最大时间须符合器件手册的规定。

振荡器电路设计成可使用陶瓷振荡器或晶振。由于每个晶振和陶瓷振荡器都有各自的特性，用户应当向晶振制造商咨询外部元件的合理使用值。

另外，GMS87C1202还可使用外部RC振荡。这样为对时间无严格要求的应用节约了成本。RC振荡器频率为电源电压，外部电阻（ $R_{ext}$ ），和电容 $C_{ext}$ ，以及操作温度的函数。用户需要考虑由于外部RC元件的误差所造成的影响。图13-4为RC元件与GMS87C1202的连接。

**注：**使用系统时钟振荡器时，布线应避开图13-2所示区域，以防任何来自布线电容干扰。

- 使布线最短。
- 不允许线路穿越其他信号导体
- 不允许线路接近可变高电流。
- 对于  $V_{SS}$ ，要预留振荡电容接地的位置  
凡高电流不接地。
- 不从振荡器取信号。

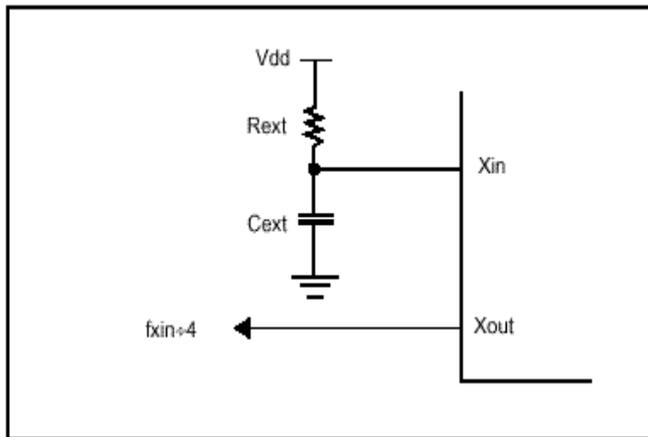


图13-4 RC 振荡器的连接

为设置RC振荡器，应j将CONFIG(0FF0<sub>H</sub>)的RCOPT位编程为1( 详见器件配置区)。

**注:**

使用系统时钟振荡器时，注意布线应避开图13-4所示区域，以防任何来自布线的感应。

- 使布线最短。
- 不允许线路穿越其他信号导体
- 不允许线路接近可变高电流。
- 对于V<sub>SS</sub>，要预留振荡电容接地的位置。且，凡出现高电流处不接地。
- 不从振荡器取信号。

振荡器频率经4分频后从X<sub>OUT</sub>脚输出，可用于测试 或与其它逻辑同步。

## 14. 基本间隔定时器

GMS87C1202带有一个8位基本间隔定时器。该定时器独立运行，不能停止。框图见图14-1。8位基本间隔定时器寄存器(BITR)在每个计数脉冲输入时加1。该计数脉冲经过预分频器分频。由于预分频器的分频比率为8~1024，因此计数速率为振荡频率的1/8~1/1024。当计数值从FF<sub>H</sub>溢出翻转到00<sub>H</sub>，就会产生基本间隔定时器中断。BITF为中断请求标志。

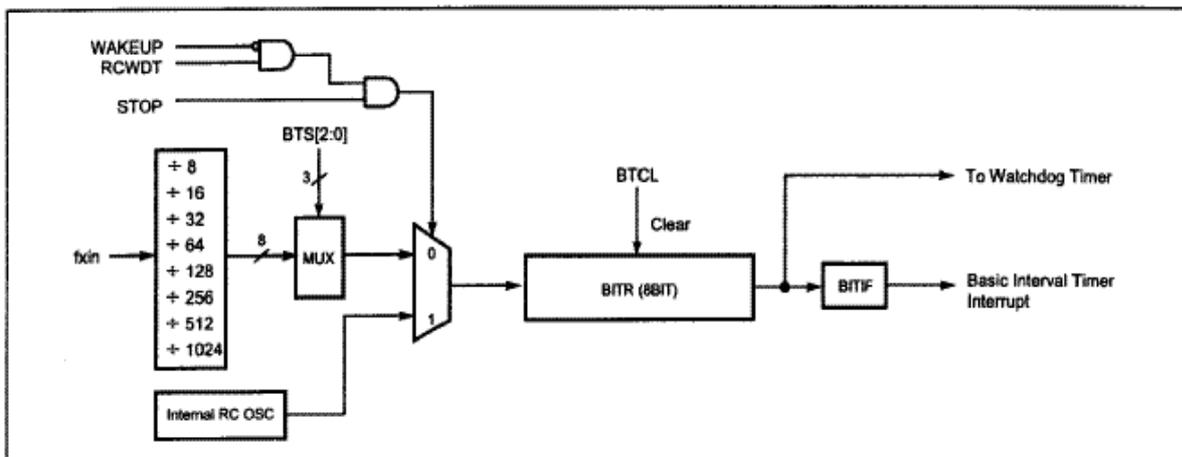


图14-1 基本间隔定时器方框图

将CKCTLR中的位BTCL置1时，BITR寄存器清零，并重新开始计数。位BTCL在一个机器周期后由硬件清零。

如果在将CKCTLR的位WAKEUP置1后执行STOP指令，就进入唤醒定时器模式。在该模式中除振荡器预分频器（仅有fxin=2048）和定时器0外，其它所有的模块都停止运行。

如果将CKCTLR的位RCWDT置1后执行STOP指令，就进入内部RC振荡看门狗定时器模式。在该模式中，除了内部RC振荡器，基本间隔定时器，和看门狗定时器外，其它所有的模块都停止运行。更多细节详见节电模式章节。由位WDTON来决定是作为看门狗定时器，还是通常的7位定时器。

**注：**CKCTLR寄存器内基本间隔定时器的所有控制位都位于同一个地址( EC<sub>H</sub> )。地址EC<sub>H</sub>读的时候作为BITR，写的时候作为CKCTLR，因此CKCTLR不能通过位操作指令访问。

时钟控制寄存器								地址: ECH
								复位值: 00H
								不可位寻址
	-	WAKEUP	RCWDT	WDTON	BTCL	BTS2	BTS1	BTS0
符号	功能描述							基本间隔定时器时钟选项
WAKEUP	1: 使能唤醒定时器 0: 禁用唤醒定时器							000: $fxin \div 8$
RCWDT	1: 使能内部 RC 看门狗定时器 0: 禁用内部 RC 看门狗定时器							001: $fxin \div 16$
WDTON	1: 使能看门狗定时器 0: 作为 7 位定时器							010: $fxin \div 32$
BTCL	1: BITR 清零, BTCL 在一个机器周期后自动变为 0, BITR 继续计数。							011: $fxin \div 64$
								100: $fxin \div 128$
								101: $fxin \div 256$
								110: $fxin \div 512$
								111: $fxin \div 1024$

图 14-2 CKCTRL: 时钟控制寄存器

### 15. 定时器/计数器

GMS87C1202有两个定时/计数器寄存器。每个模块都可产生中断来表明发生了事件(即, 定时器匹配)。

定时器0和1, 可作为两个8位定时/计数器或将其组合成一个16位定时/计数器。

在定时器功能中, 寄存器在每个内部时钟输入时加1。因此, 可将其看作是对内部时钟输入的计数。由于1个时钟周期最短为2个最长包括2048个振荡周期, 因此, 定时器0的计数速率为振荡器频率的1/2 ~ 1/2048。定时器1也可使用同一个时钟源。另外, 定时器1可有更高速的时钟源(1/1 ~ 1/8)。

在计数器功能中, 寄存器在对应外部输入脚 EC0 0到1的跳变上升沿时加 1。

在捕捉功能中, 寄存器的增加与在定时器功能一样。但在外部中断沿输入时, 计数寄存器的值被捕捉到捕捉数据寄存器CDRx 中。

定时器 1还具有PWM功能和比较输出功能。

通过设定定时器模式寄存器TM0 和TM1, 可实现以下 7种操作模式: 8 位定时/计数器, 16位定时/计数器, 8位捕捉, 16位捕捉, 8位比较输出, 16位比较输出, 10位PWM, 如图15-1和表15-1所示。

Timer 0 Mode Register								ADDRESS : 00H
								RESET VALUE : -000000
TM0	-	-	CAP0	T0CK2	T0CK1	T0CK0	T0CN	T0ST
CAP0	Capture mode selection bit				T0CN		Continue control bit	
	0 : Disables Capture 1 : Enables Capture						0 : Stop counting 1 : Start counting continuously	
T0CK[2:0]	Input clock selection				T0ST		Start control bit	
	000 : $fxin \div 2$		100 : $fxin \div 128$				0 : Stop counting 1 : Counter register is cleared and started again	
	001 : $fxin \div 4$		101 : $fxin \div 512$					
	010 : $fxin \div 8$		110 : $fxin \div 2048$					
	011 : $fxin \div 32$		111 : External Event ( ECD )					
Timer 1 Mode Register								ADDRESS : 02H
								RESET VALUE : 00000000
TM1	POL	16BIT	PWME	CAP1	T1CK1	T1CK0	T1CN	T1ST
POL	PWM Output Polarity				T1CK[2:0]		Input clock selection	
	0 : Duty active low 1 : Duty active high				00 : $fxin$		10 : $fxin \div 8$	
16BIT	16-bit mode selection				T1CN		Continue control bit	
	0 : 8-bit mode 1 : 16-bit mode						0 : Stop counting 1 : Start counting continuously	
PWME	PWM enable bit				T1ST		Start control bit	
	0 : Disables PWM 1 : Enables PWM						0 : Stop counting 1 : Counter register is cleared and started again	
CAP1	Capture mode selection bit							
	0 : Disables Capture 1 : Enables Capture							

图 15-1 定时器 0和定时器 1模式寄存器

16BIT	CAP0	CAP1	PWME	T0CK[2:0]	T1CK[1:0]	PWME	定时器0	定时器1
0	0	0	0	xxx	xx	0	8位定时器	8位定时器
0	0	1	0	111	xx	0	8位事件计数器	8位捕捉
0	1	0	0	xxx	xx	1	8位捕捉	8位比较输出
0	0	0	1	xxx	xx	1	8位定时器计数器	10位PWM
1	0	0	0	xxx	11	0	16位定时器	
1	0	0	0	111	11	0	16位事件计数器	
1	1	x	0	xxx	11	0	16位捕捉	
1	0	0	0	xxx	11	1	16位比较输出	

表 15-1 定时器0 和定时器1 的操作模式

- 1, 此位为RB功能寄存器( RBFUNC )的位4。
- 2, X: 其值或0或1, 取决于用户的操作

### 15.1 8位定时器 /计数器模式

GMS87C1202有两个8位定时/计数器，定时器 0和 1，如图15-2。  
 定时或计数功能，由模式寄存器TM0， TM1进行选择，如图15-1和表15-1。设定8位定时/计数器模式时，TM0 中的位CAP0 和TM1中的 16BIT 位需清零( 表15-1)。

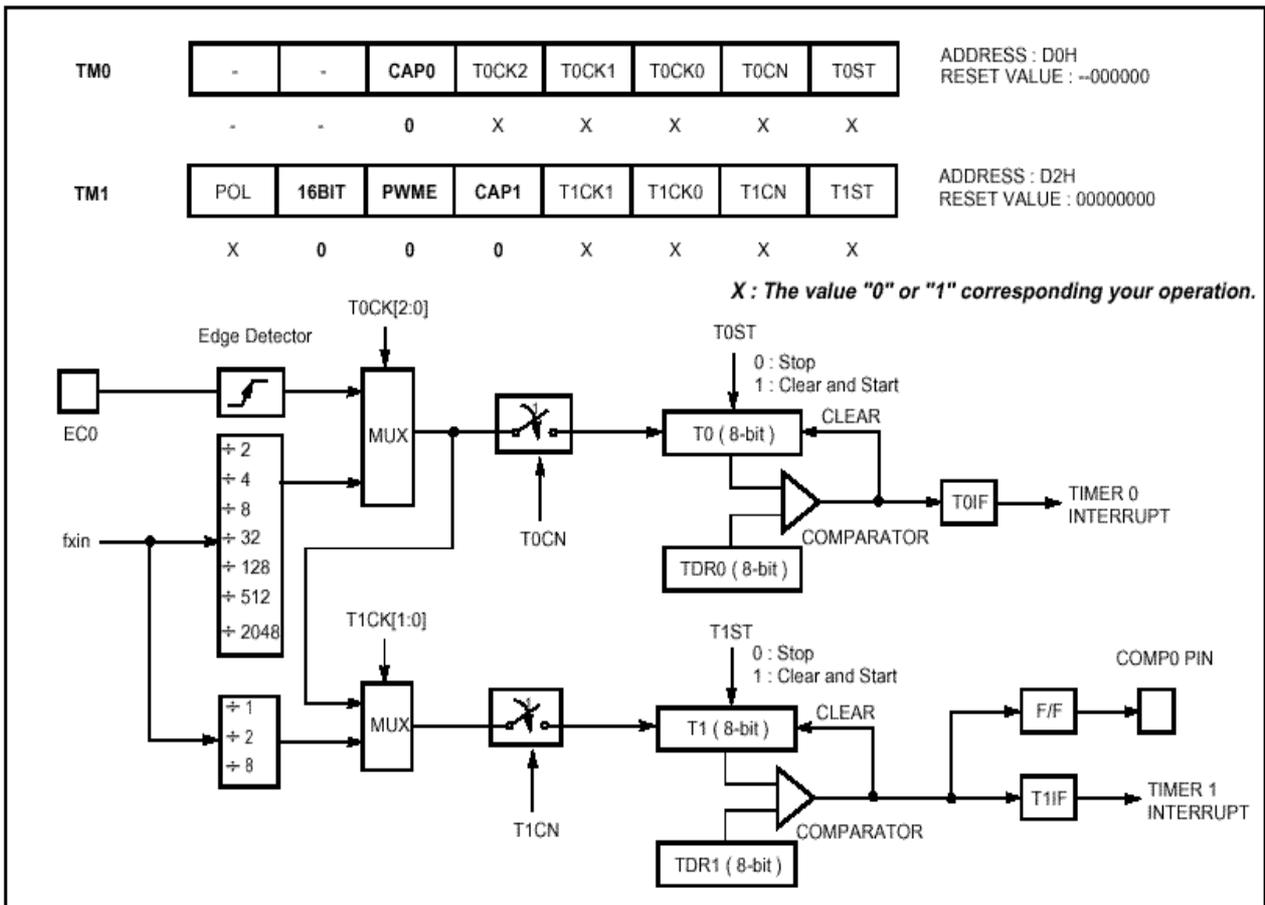


图15-2 8位定时/计数器模式

这两个定时器都有8位计数寄存器和数据寄存器。计数寄存器在每个内部或外部时钟输入时加1。内部时钟有预分频比率选项 2, 4, 8, 32, 128, 512, 2048 (由寄存器TM0中的控制位T0CK2, T0CK1, 和T0CK0选择)和1, 2, 8(由寄存器 TM1中的控制位T1CK1, T1CK0 选择)。在定时器0中, 定时器寄存器T0 从00<sub>H</sub> 开始增加直到与TDR0 一致时复位翻转到00<sub>H</sub>。定时器0的匹配输出产生定时器0中断 (锁存于T0F位)。TDRx 和Tx 寄存器位于同一个地址, 读操作时为Tx, 写操作时为TDRx。

在计数器功能中, 计数器在EC0脚的每个0到1的迁移中(上升沿)加1。为使用计数器功能, RA方向寄存器RAIO中的位RA0应置0。定时器0 可通过EC0 输入用作一个计数器, 定时器1则不能。

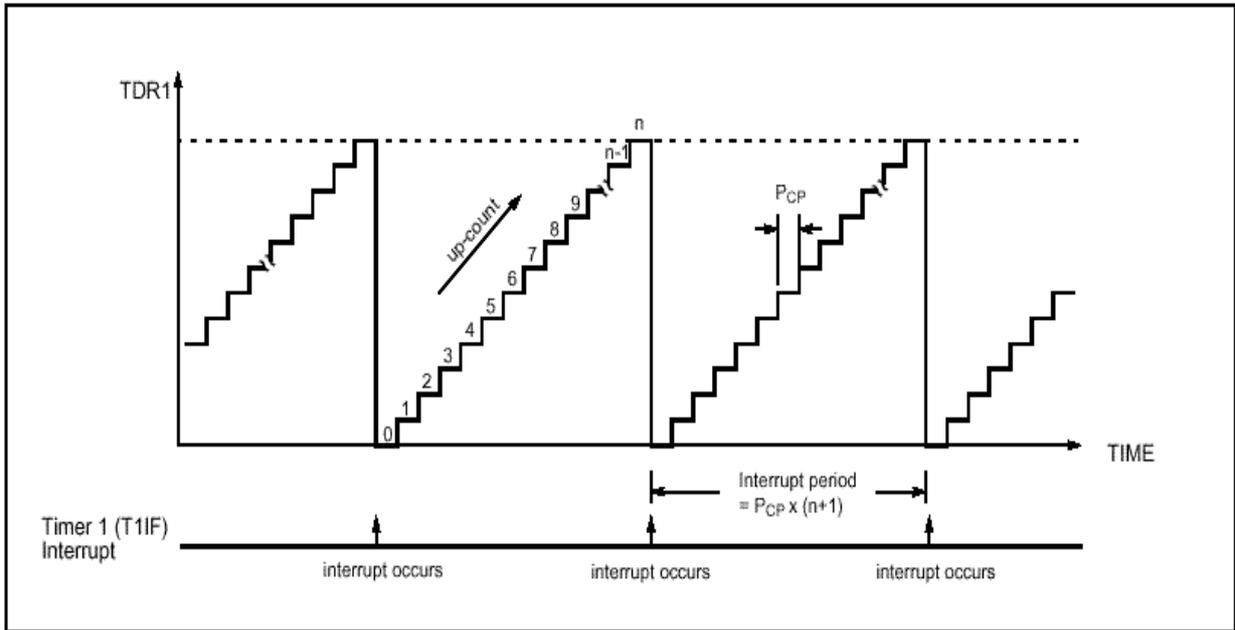


图15-3 定时器数据寄存器计数示例

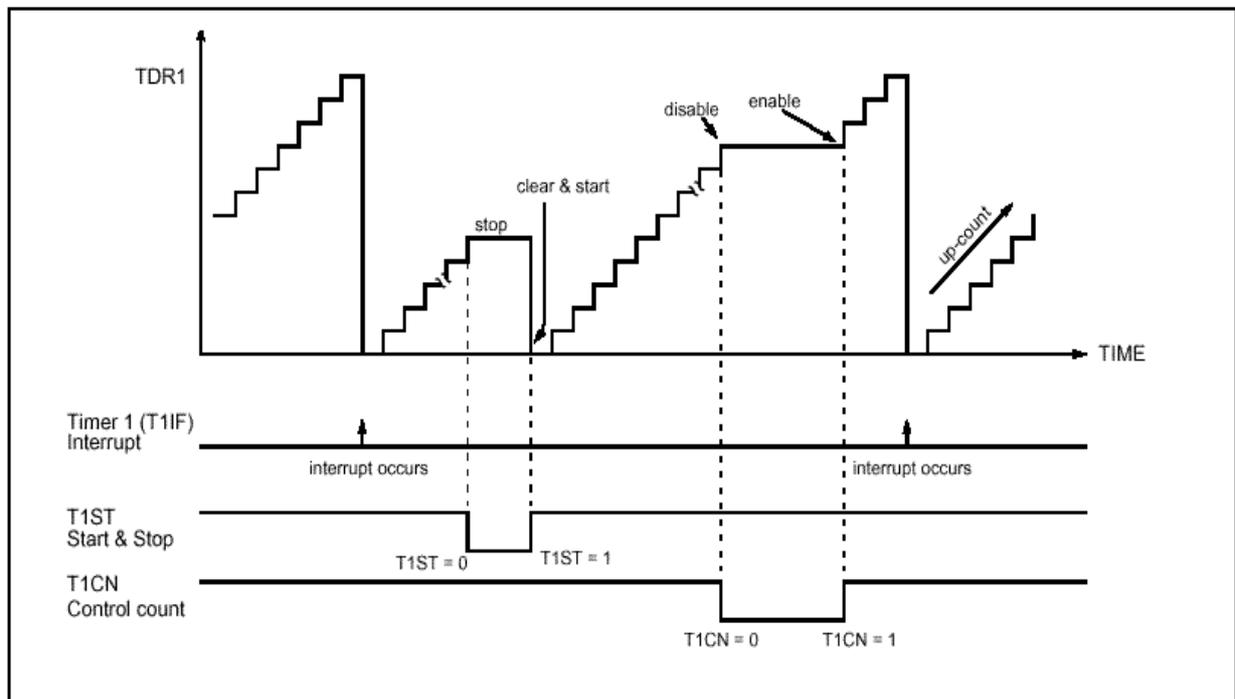


图15-4 定时器计数操作

## 15.2 16 位定时/计数器模式

定时器寄存器以16 位运行。16位定时/计数器寄存器T0，T1 从0000<sub>H</sub> 开始增加直到与TDR0，TDR1一致时，复位翻转为0000<sub>H</sub>。匹配输出产生定时器0 中断，而定时器1 则不中断。

定时器 0 的时钟源，可通过位T0CK2，T0CK1 和T0SL0选择为内部或外部时钟输入。

在 16 位模式中，位T1CK1 和T0SL0，TM1 的16BIT，都应置为1。

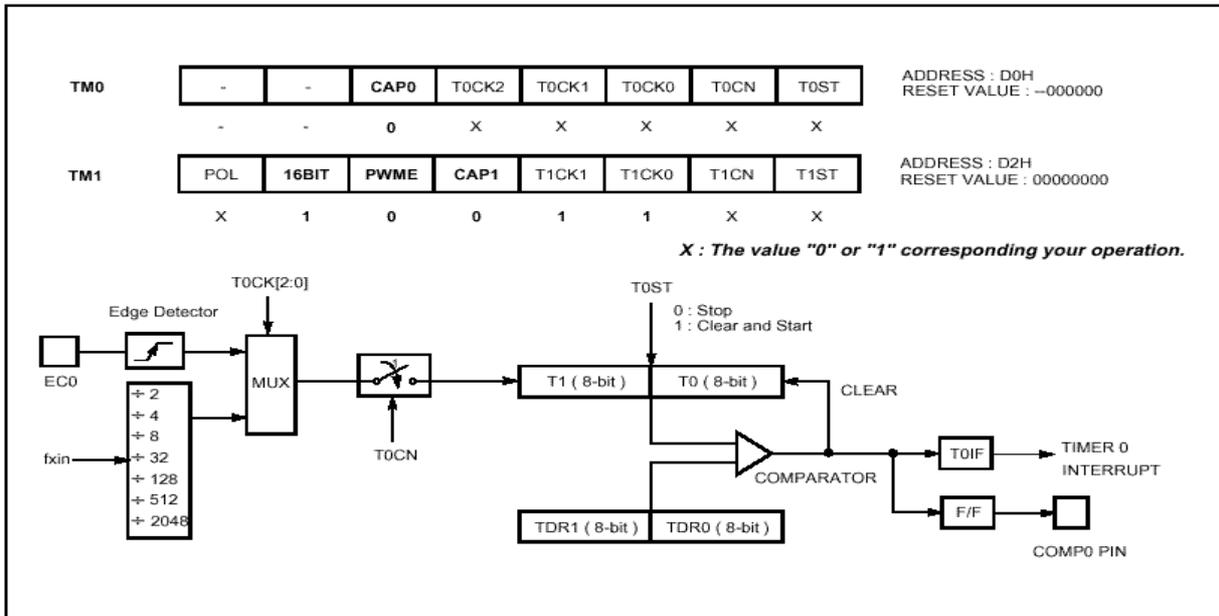


图15—5 16位定时/计数器模式

## 15.3 8位比较输出(16位)

GMS87C1102/1202 具有定时器比较输出功能。为从管脚输出脉冲，定时器匹配可取管脚（COMP0），如图15—2和图 15—5所示。于是。脉冲输出从定时器匹配产生。这些操作是由管脚RB4/COMP0/PWM实现的。

这一管脚输出的，是50：50 空比的方波，输出频率见下式

$$f_{\text{COMP}} = \frac{\text{振荡器频率}}{2 \times \text{预分频值} \times (\text{TDR} + 1)}$$

在这一模式中，RB功能寄存器（RBFUNC）的位PWMO应置为1，且定时器1模式寄存器（TM1）的位PWME则应置为0。

此外，16位比较输出模式亦可使用。

## 15.4 8位捕捉模式

定时器0捕捉模式由定时器模式寄存器TM0的位CAP0设定（而定时器1，则由定时器模式寄存器TM1的位CAP1设定），如图15—6所示。

如上所说，定时器0和定时器1均可用作捕捉模式。

定时/计数寄存器是在回应内部或外部输入时增1的。这一计数功能与通常的计数功能相同，且定时器中断在定时器寄存器T0（T1）增1且达到与TDR0（TDR1）一致时发生。

当捕捉信号的脉宽远大于定时器的最大周期时，上述捕捉模式中的定时器中断非常有用。

例如，在图15—8中，捕捉信号的脉宽大于定时器数据值(  $FF_H$  )的两倍。当外部中断产生时，捕捉值(  $13_H$  )远远小于所需要的值。可通过计数定时器溢出的次数得到正确值。

定时/计数器仍和前面一样工作，但有一个额外的特性，即在外部输入INTx管脚上的边沿到来时，将定时器寄存器( T0 , T1)的当前值，分别捕捉到寄存器CDRx(CDR0,CDR1)中。捕捉完成后，定时器寄存器清零，并由硬件重新启动。

捕捉可有3种触发模式：下降沿，上升沿，和双边沿。可通过中断边沿选择寄存器IEDS进行选择( 参考外部中断章节)。另外，INTx脚的跳变信号可产生中断。

注：CDRx,TDRx 和Tx 位于同一个地址在捕捉模式中读操作是对CDRx 进行，操作写操作则是对TDRx。

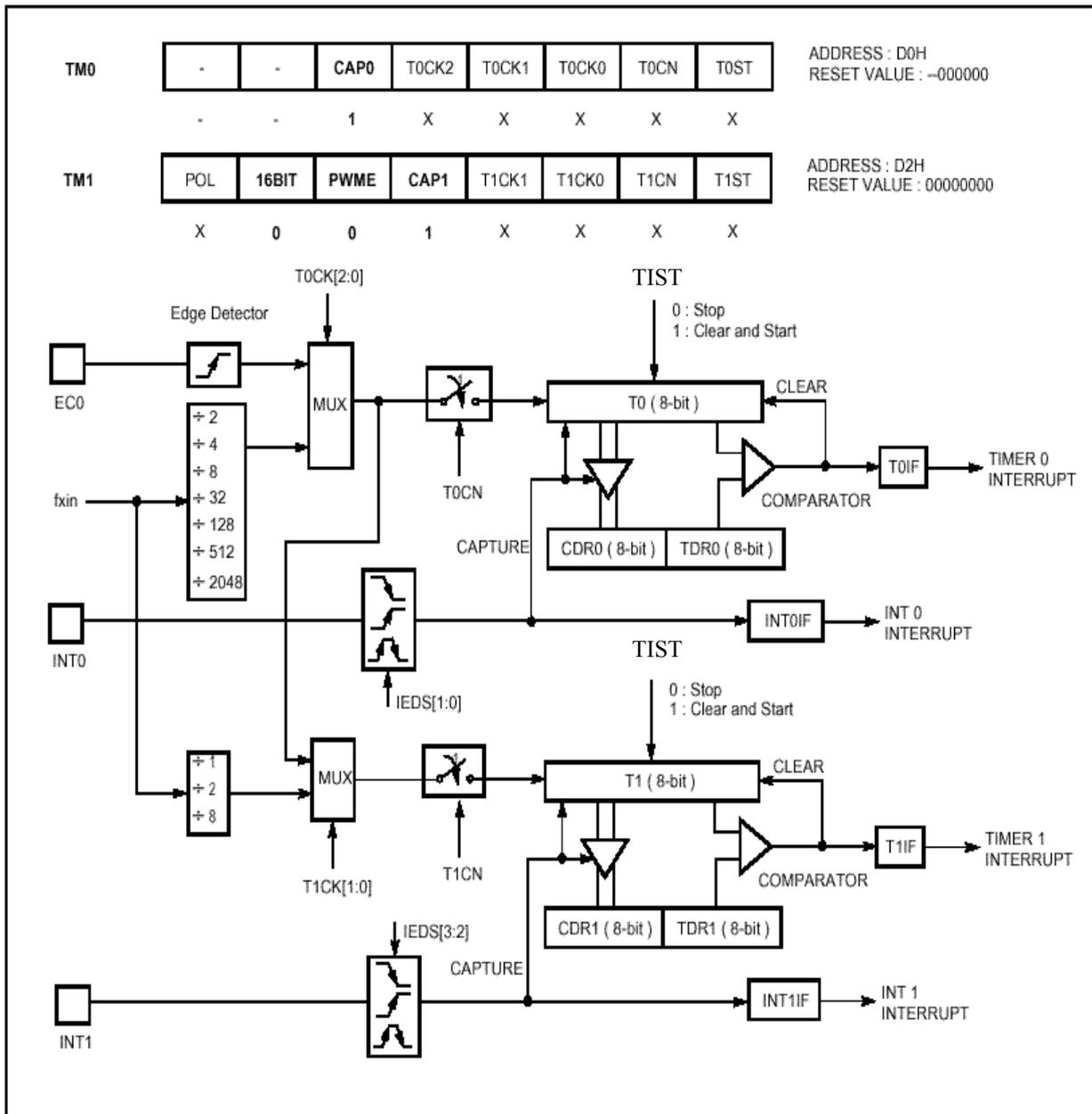


图15—6 8位捕捉模式

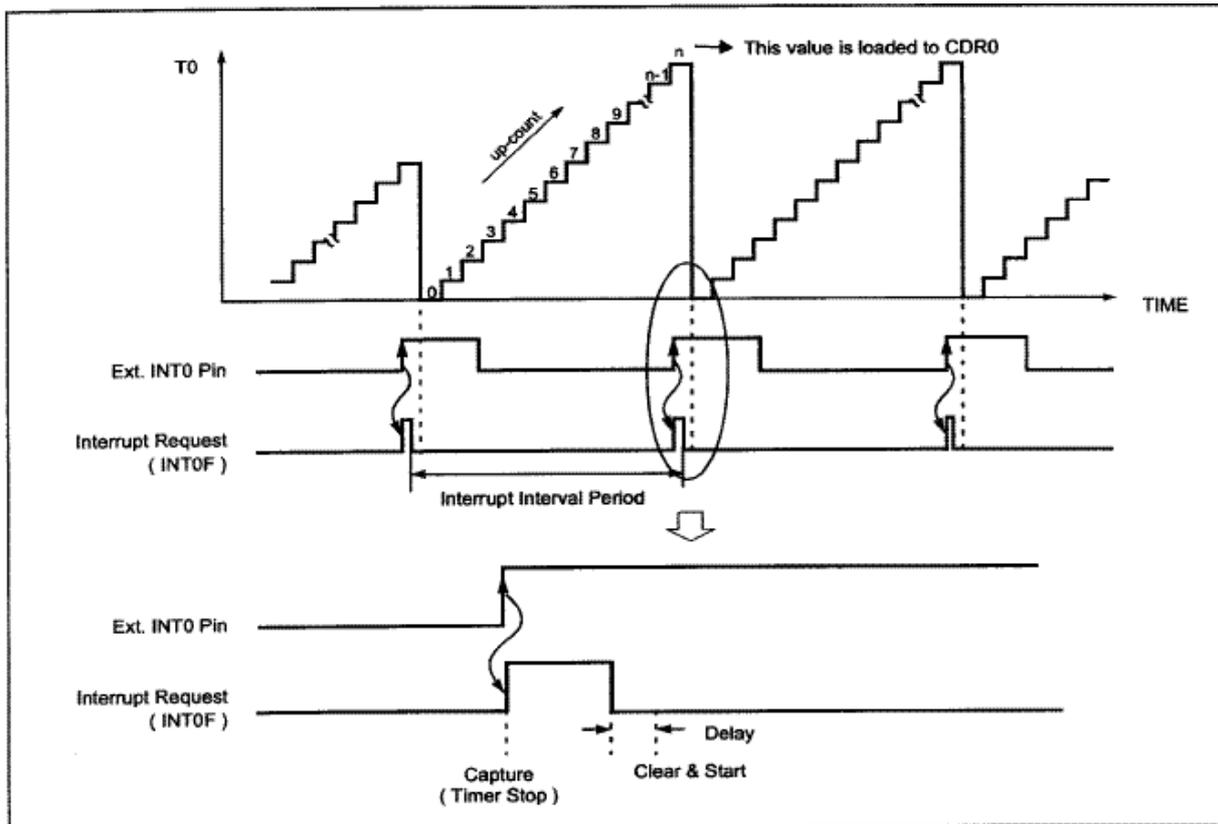


图 15-7 输入捕捉操作

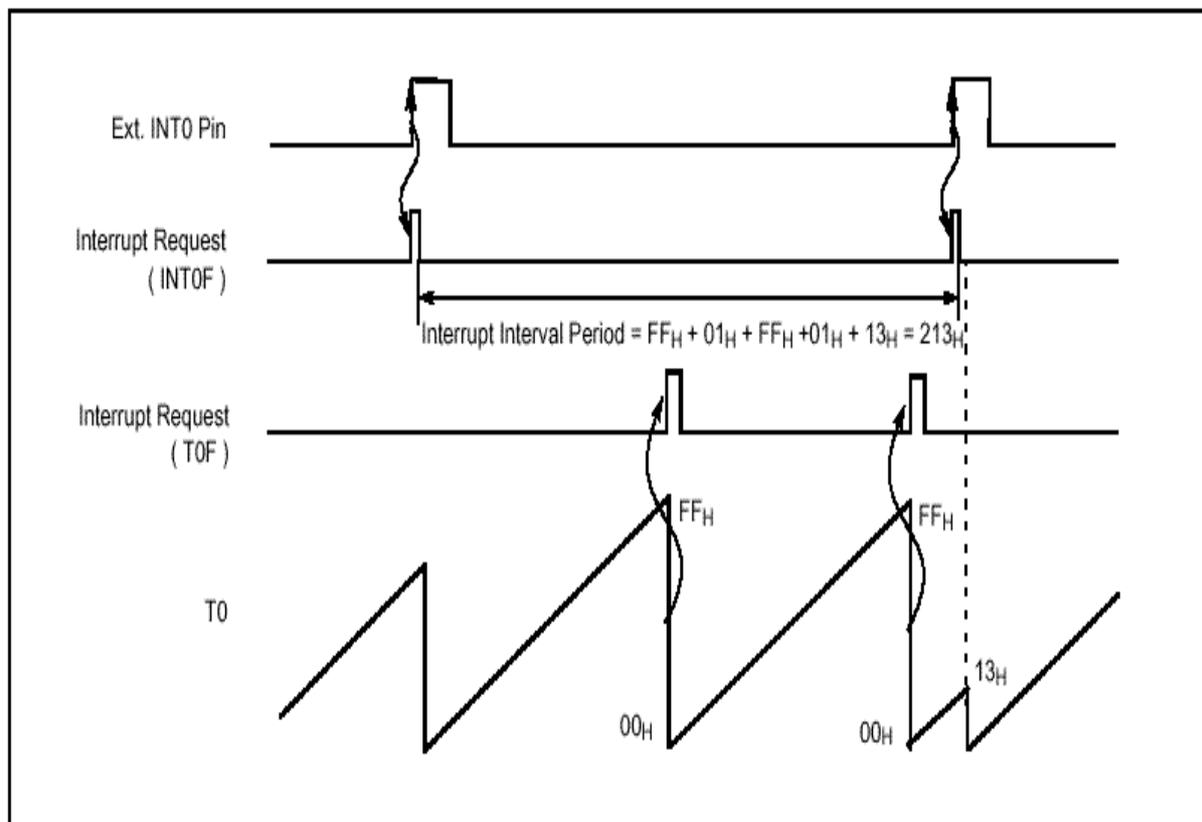


图 15-8 捕捉模式中定时器的多次溢出

### 15.5 16位捕捉模式

除了定时器寄存器以16位运行外，16位捕捉模式和8位捕捉模式相同。

定时器0的时钟源可通过位T0CK2，T0CK1和T0CK0选择为内部或外部时钟。

在16位模式中，位T1CK1，T1CK0和TM1的16BIT 都应置1。

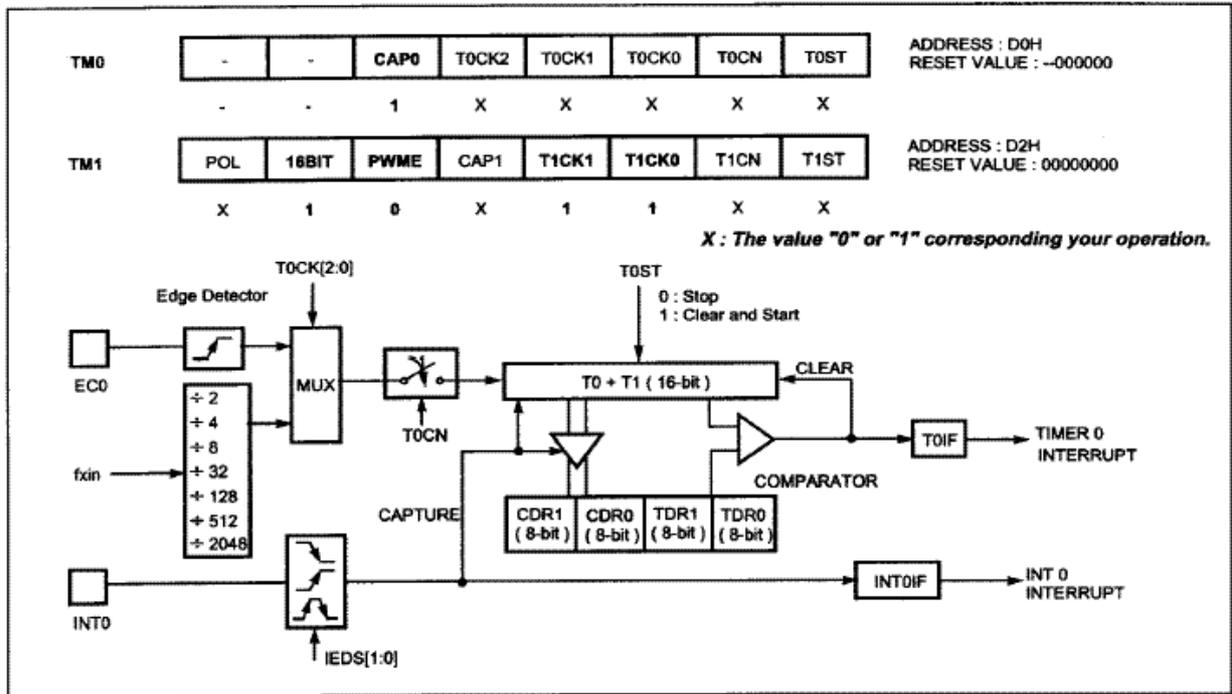


图15—9 16 位捕捉模式

### 15.6 PWM 模式

GMS87C1202具有2个高速PWM脉宽调制功能(PWM,Pulse With Modulation)模块，并与定时器1共享。

在PWM模式中，管脚RB4/COMP0/PWM0可输出直到10位分辨率的PWM信号。通过将RBFUNC寄存器中的PWM0位置1，可将这些管脚定义为PWM输出。

PWM输出的周期由T1PPR( PWM0周期寄存器)和PWM0HR[3:2](PWM0高速寄存器的位3, 2)确定，且PWM输出的占空比由T1PDR(PWM占空比寄存器)和PWM0HR[1:0] (PWM0高速寄存器位1, 0) 确定。

用户将低8位周期值写入T1PPR，并将高2位周期值写入PWM0HR[3:2]。占空比值也以同样方式写入T1PDR和PWM0HR[1:0]。

对于正常PWM输出而言，T1PDR的配置为双缓冲输出。在图15—10中，当周期值与计数值一致时，占空比值从主控寄存器发送到被控器( 即，在下一空比周期之开头)。

$$PWM \text{ 周期} = [PWM0HR[3:2]T1PPR] \times \text{源时钟}$$

$$PWM \text{ 占空比} = [PWM0HR[1:0]T1PDR] \times \text{源时钟}$$

有关频率和分辨率的上面等式表明，频率与分辨率成反比关系见。表15—2说明，要得到较高频率的PWM，就应降低分辨率。

频率	分辨率		
	T1CK1[1:0]=00(125ns)	T1CK1[1:0]=00(125ns)	T1CK1[1:0]=00(125ns)
10位	7.8kHz	3.9kHz	0.98kHz
9位	15.6kHz	7.8kHz	1.95kHz
8位	31.2kHz	15.6kHz	3.90kHz
7位	62.5kHz	31.2kHz	7.81kHz

表15—2 8MHz时PWM频率与分辨率的关系

TM1 的位POL 决定占空比的极性。

如果占空比值设成与周期一致，PWM输出由位POL决定(1：高，0：低)。如果占空比值设成 00<sub>H</sub>，则PWM输出由位POL 决定(1：低，0：高)。

当PWM输出时可改变占空比值。当前周期结束后，就会输出改变后的占空比值。当仅改变周期值时，占空比可保持不变，如图15—12 所示。在变化的频率中，绝对占空比时间不变。但改变的周期必须大于占空比值。

*注：在PWM输出开始命令中，第一个脉冲将被特殊输出。因为，如果用户在定时器操作时写寄存器值，这些寄存器应首先被设置为一定值。为了防止这类操作，用户必须停止PWM定时器时钟，并设置占空和周期寄存器值。*

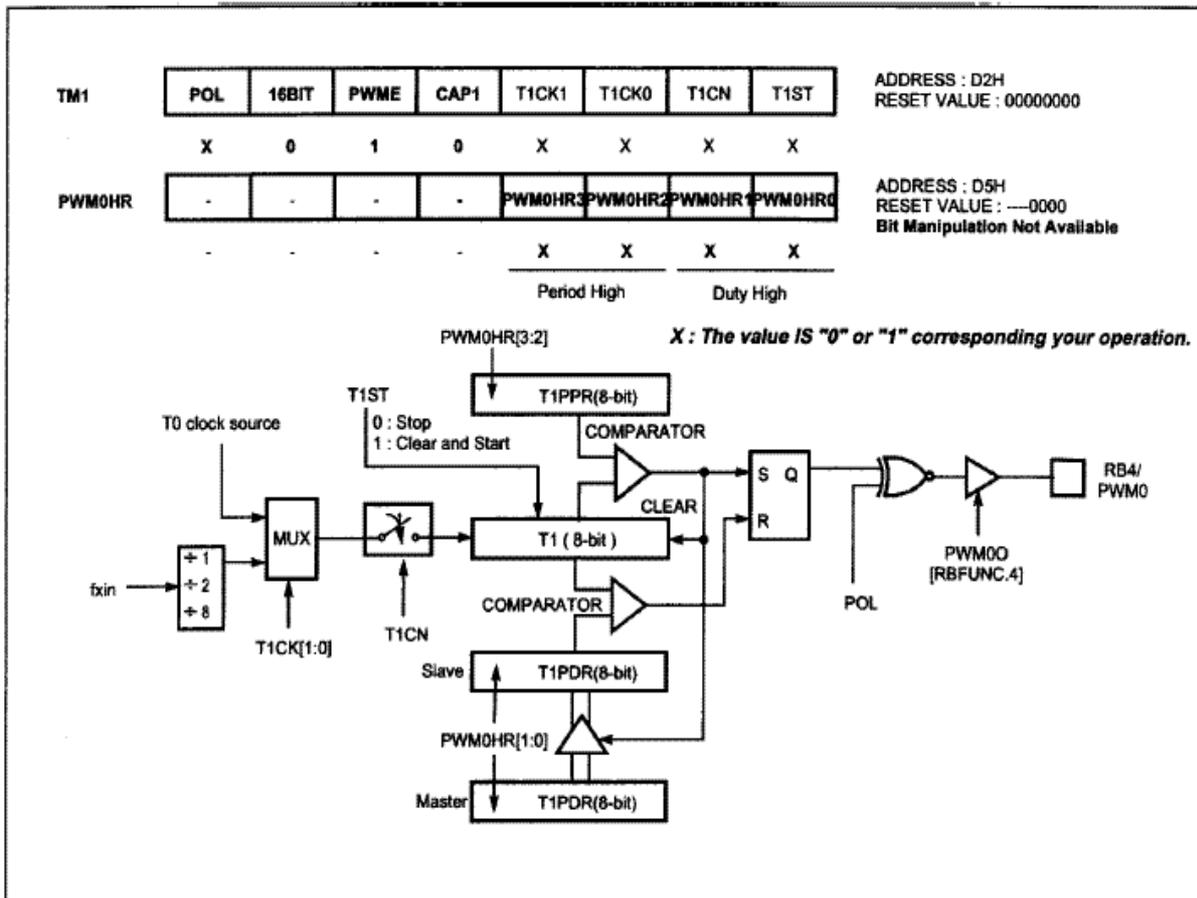


图15-10 PWM 模式

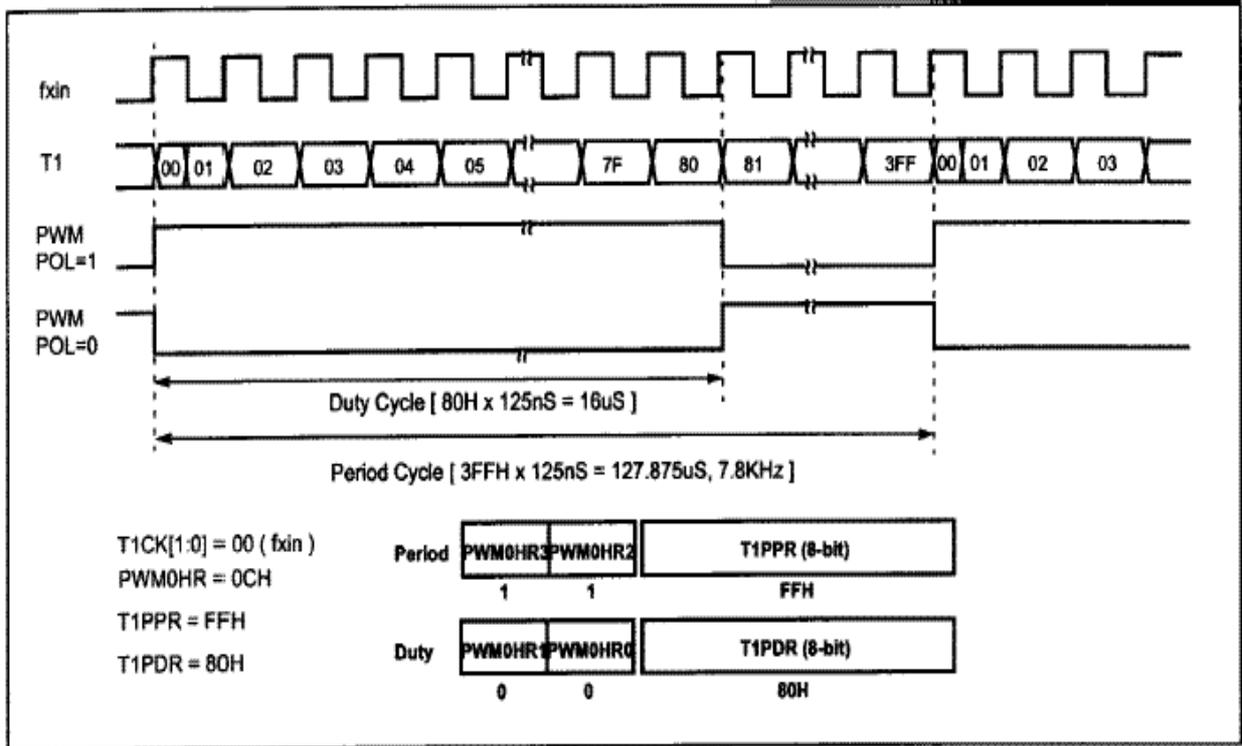


图 15-11 8MHz时的PWM举例

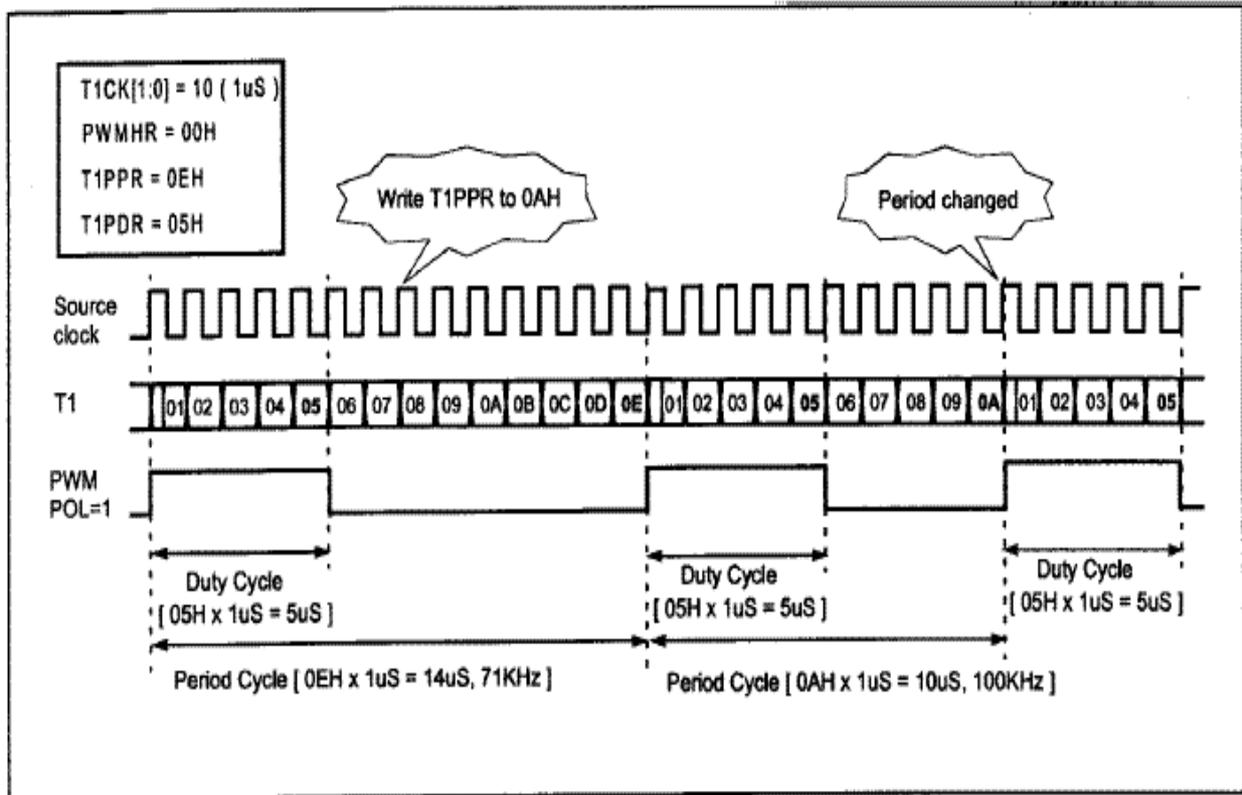


图 15-12 在绝对占空比中改变周期示例 (8MHz时)

### 16. 蜂鸣器输出功能

蜂鸣器驱动器包括6位二进制计数器，蜂鸣器寄存器BUR，和时钟选择器。通过用户编程计数器，可产生宽范围频率的方波（ $f_{in}=4MHz$  时480Hz~250KHz）。

将RBFUNC 的BUZO 位置1， 可将RB1作为蜂鸣器的输出口。

对BUR寄存器的写信号，可将6位蜂鸣器计数器清零，并启动计数。它将从00<sub>H</sub>开始增加，直到它与6位寄存器BUR相一致。

蜂鸣器计数器在溢出之后清零，并计数直到输出占空比为50 的方波。

BUR的位 0至5确定蜂鸣器的驱动输出频率。频率的计算公式如下：

$$f_{BUZ}(Hz) = \frac{\text{振荡器频率}}{2 \times \text{预分频比率} \times (BUR+1)}$$

BUR中的位BUCK1，BUCK0 用于从预分频输出中选择源时钟。

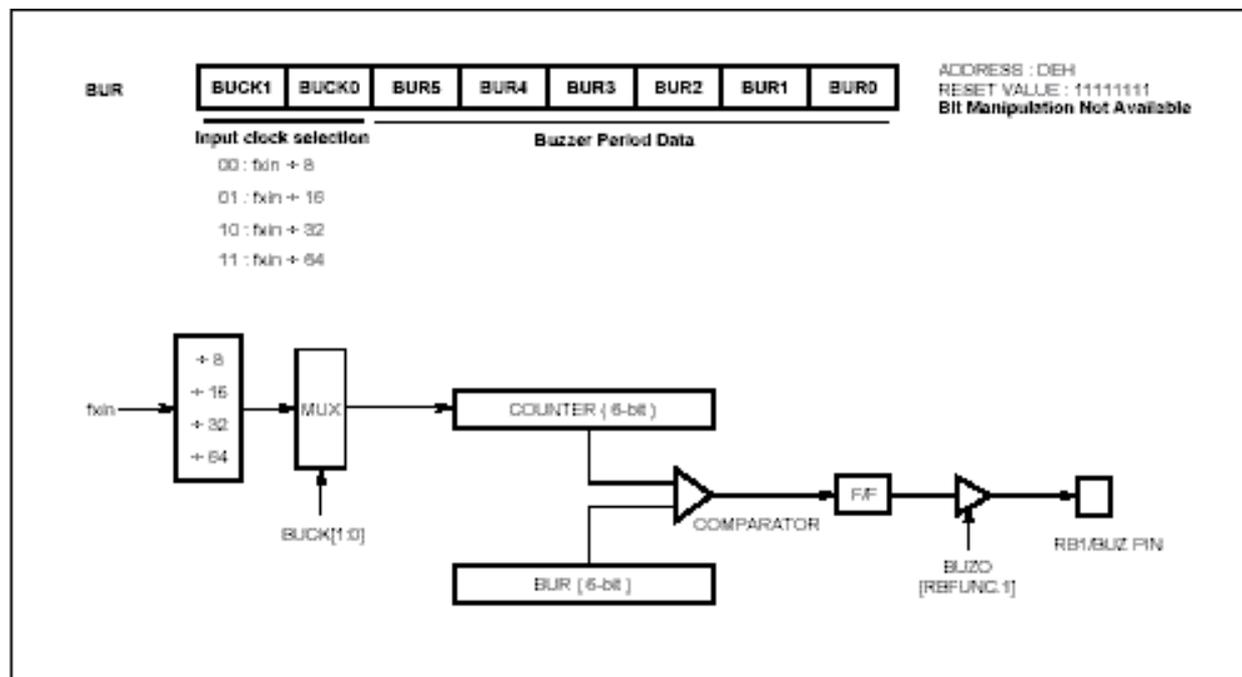


图16—1 蜂鸣器驱动器

## 17. A/D 转换器

A/D 转换器将模拟输入信号转换成对应的8位数字量。A/D 转换器有8个模拟输入端，通过多路开关输入到一个采样/保持器。采样/保持器的输出，作为转换器的输入，经过逐次逼近得到转换结果。

模拟参考电压可通过设定RBFUNC中的位AVRFS 选择  $V_{DD}$  或  $AV_{ref}$ 。如果选择外部模拟参考电压  $AV_{ref}$ ，位ANSEL0不能置位为1，因为该管脚作为A/D转换器的模拟参考输入。

A/D 模块有两个寄存器，控制寄存器ADCM 和A/D，结果寄存器ADCR。ADCM寄存器，如图17—2所示，控制A/D 转换模块的操作。管口可配置为模拟输入或数字I/O口。

为使用模拟输入，可通过将RAFUNC中的ANSEL[7:0]置位，将各个口设置为模拟输入。并通过设置ADS[2:0]，来选择相应的通道进行转换。

当启动位ADST置为1时，启动转换过程。在一个周期之后，ADST由硬件清零。寄存器ADCR 包含了A/D 转换的结果。转换完成时，结果装入ADCR。A/D 转换状态位ADSF被置位，A/D 中断标志被置位，A/D 模块的方框图见图17—1。A/D 转换完成时，ADSF自动被置位，并在转换进行时清零。转换时间大约需要10  $\mu$ s(  $f_{xin}=8$ MHz时)。

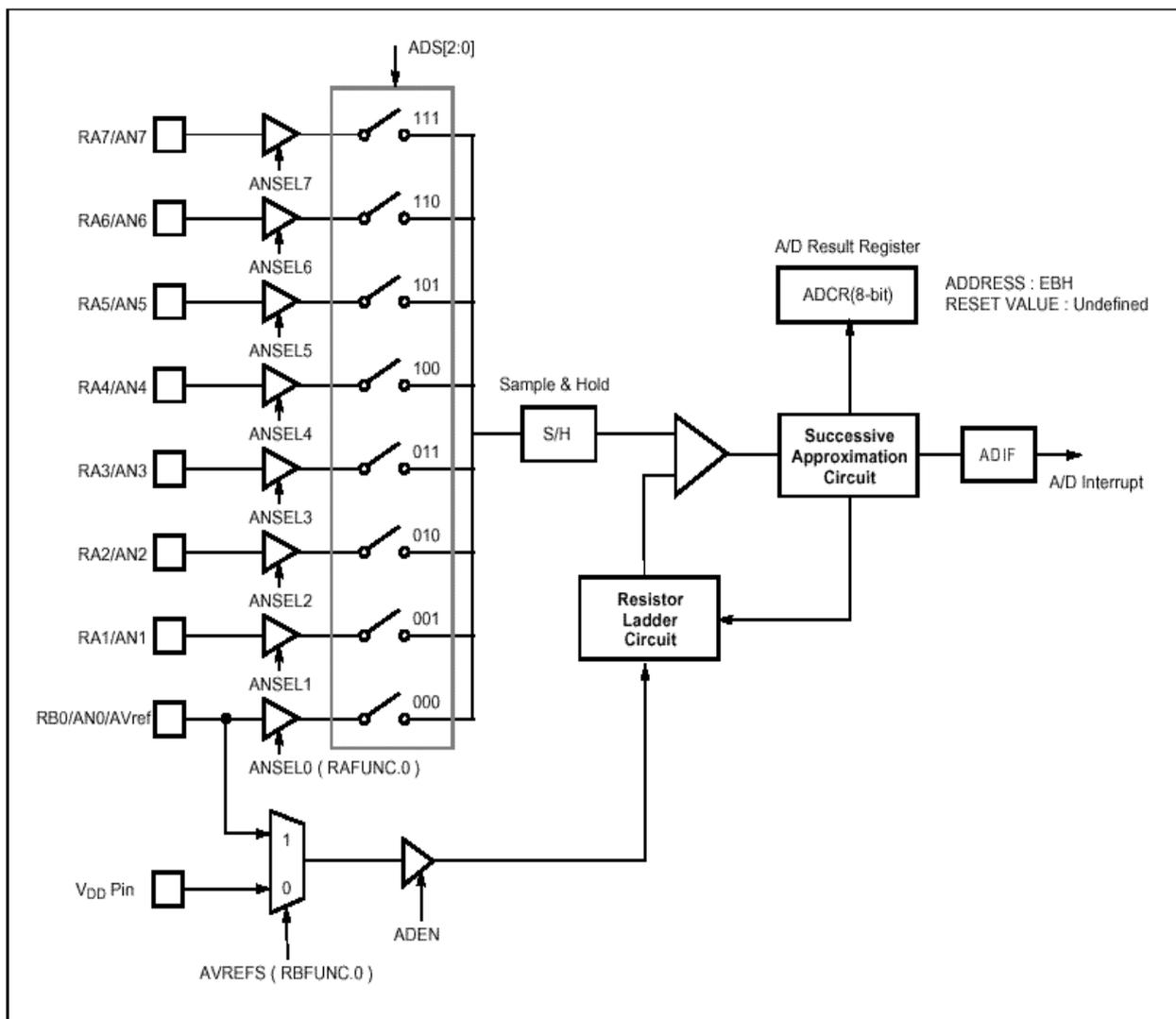


图 17-1 A/D 转换器方框图

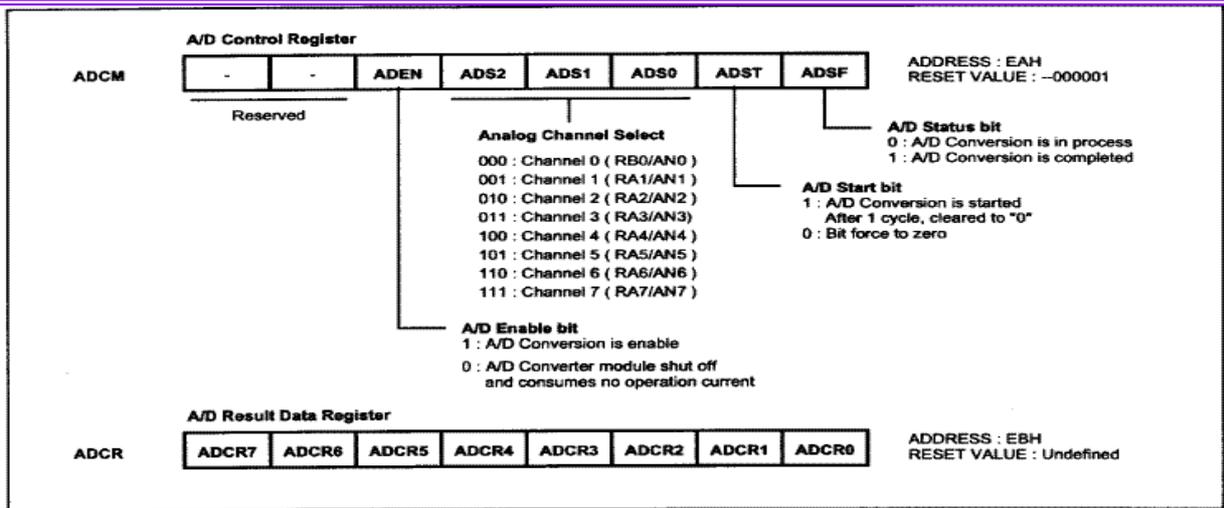


图 17-2 A/D 转换器寄存器

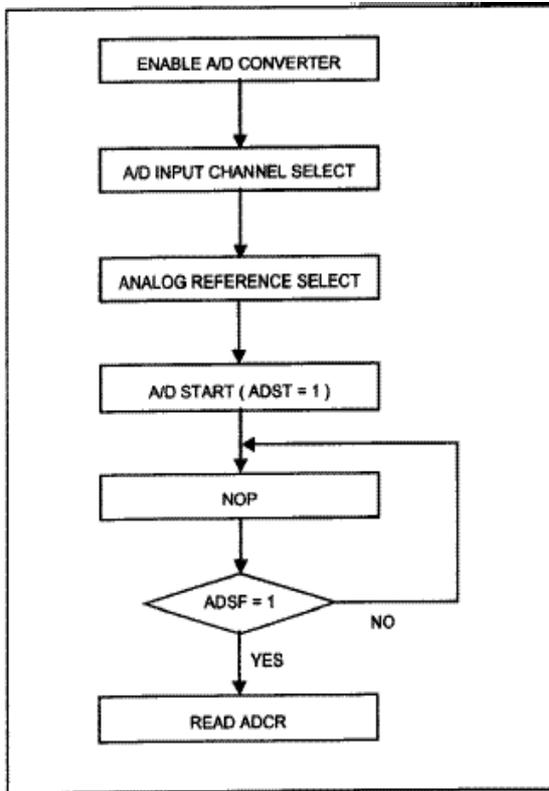


图17-3 A/D 转换器操作流程

**A/D 转换注意事项**

(1) AN0→ AN7 的输入范围

AN0 →AN7 的输入电压应在规定范围内。特别地，如果电压超过  $V_{DD}$  或  $Av_{ref}$ ，或低于  $V_{SS}$ ，即使在极限参数范围内，该路的转换结果将是不确定的。其它路的转换也会受到影响。

(2) 降噪措施

为了保持 8 位的分辨率，必须注意  $AV_{ref}$  和 AN0→AN7 管脚上的噪声。建议在模拟输入管脚连接一个电容以降低噪声，如图 17—4。

(3) 管脚 AN0/RB0 和 AN1/RA1→ AN7/RA7

模拟输入脚 AN0→AN7 也具有和输入/输出口 (RA, RB0) 相同功能。当选择 AN0 →AN7 的任意管脚执行 A/D 转换时，应确保 A/D 转换进行中不执行口输入指令，否则会降低转换的分辨率。

同样地，如果数字脉冲应用于位于进行 A/D 转换的管脚的相邻管脚，则由于噪声的影响，未必能获得理想的 A/D 转换。故，应避免将数字脉冲应用于位于进行 A/D 转换的管脚的相邻管脚，

(4)  $AV_{ref}$  的输入阻抗

$AV_{ref}$  与  $V_{SS}$  之间的串联阻抗大约为 10K。

因此，如果参考电压源的输出阻抗很高，将会导致与  $AV_{ref}$  和  $V_{SS}$  间串联电阻并联，从而出现很大的参考电压误差。

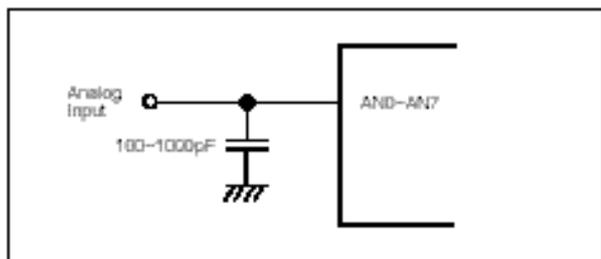


图 17—4 模拟输入脚接电容

18. 中断

GMS87C1202中断电路包括中断使能寄存器(IENH, IENL), 中断请求标志IRQH, IRQL, 中断边沿选择寄存器( IEDS), 优先级电路和主控使能标志(PSW 中的 I 位)。中断电路的配置如图18—2, 中断优先级, 如表18—1所示。

外部中断INT0 和INT1 都为边沿触发(1->0, 0->1 双边沿中断)。

实际上, 是由寄存器IRQH中的标志位INT0IF 和INT1IF产生的。产生外部中断时, 中断标志由硬件清零。

定时器0/1中断由T0IF和T1IF产生。A/D转换中断由ADIF产生。看门狗定时器中断由WDTIF 产生 ( WDTON 位置0 )。基本间隔定时器寄存器的溢出时, 由BITIF 产生中断。中断是由中断主使能标志 I PSW.2, 中断使能寄存器(IENH, IENL)和中断请求位于(IRQH, IRQL)上电复位和软件BRK 中断除外。

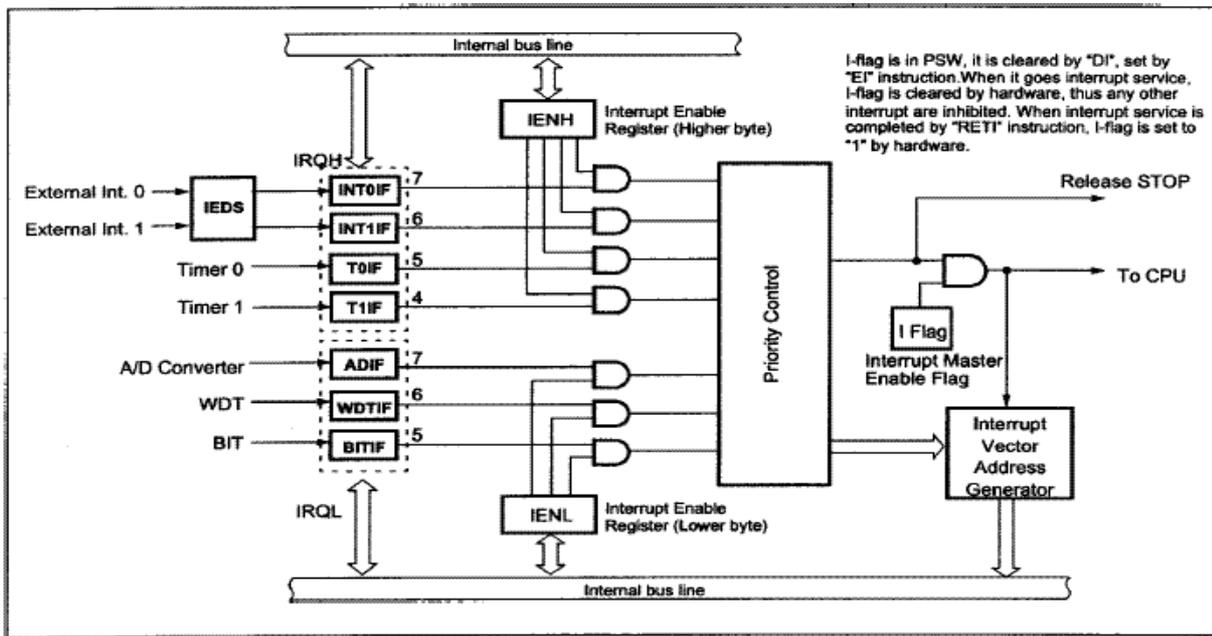


图18—1 中断功能方框图

中断使能寄存器如图18-2所示。寄存器由每一个中断源的中断使能标志组成。这些标志决定是否接受中断。当使能标志为0时, 对应的中断源被禁止。PSW 内有一个主使能位I-flag, 可以禁止所有中断。

当产生中断时, I-flag清零, 并禁止其它所有中断。返回地址和PSW 被压入堆栈。在中断服务程序中可以通过查询中断请求标志位确定中断源。

中断请求标志位必须在重新使能中断前由软件清零, 以避免出现中断。循环中断请求标志可进行读写操作。

复位/中断	符号	优先级	向量地址
硬件复位	RESET	-	FFFE <sub>H</sub>
外部中断0	INT0	1	FFFA <sub>H</sub>
外部中断1	INT1	2	FFF8 <sub>H</sub>
定时器0	Timer0	3	FFF6 <sub>H</sub>
定时器1	Timer1	4	FFE4 <sub>H</sub>
A/D转换器	A/D C	5	FFEA <sub>H</sub>
看门狗定时器	WDT	6	FFE8 <sub>H</sub>
基本间隔定时器	BIT	7	FFE6 <sub>H</sub>

表 18-1 中断优先级

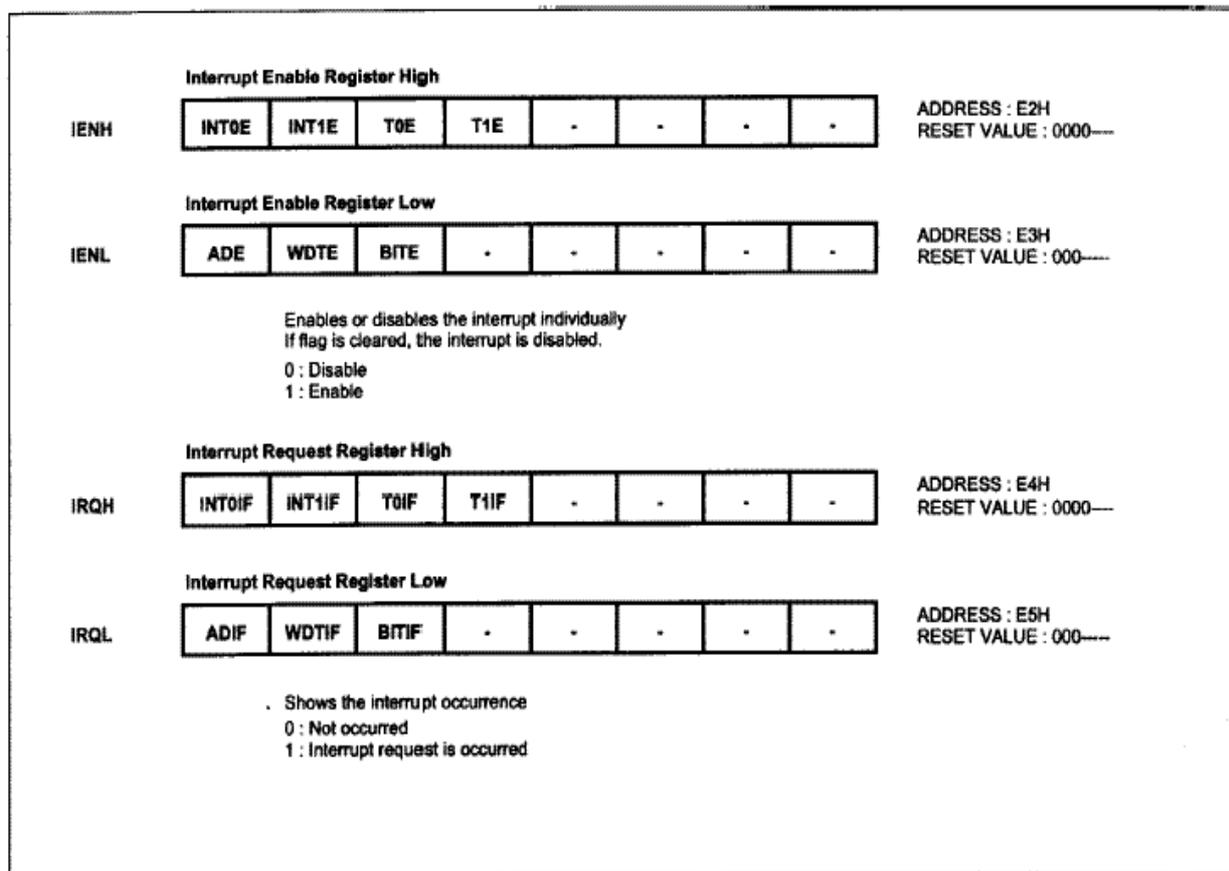


图 18-2 中断使能寄存器和中断请求寄存器

## 18.1 中断序列

一个中断请求一直保持到中断被接受，或通过复位或指令将中断锁存。清零在当前指令执行完成后需要8个  $f_{osc}$  ( $f_{XIN} = 4\text{MHz}$ 时， $2\mu\text{s}$ )响应。中断序列中断服务任务在执行[RETI]指令后终止。

### 中断响应

1. 中断主使能标志(I-flag) 清零,以暂时禁止任何可屏蔽的中断。当响应一个非屏蔽中断时，后续中断响应将被暂时禁止。
2. 被响应的中断源的中断请求标志清零。
3. 程序计数器的内容( 返回地址 )和程序状态字被保存到( 压入)堆栈区，堆栈指针减3。
- 4 中断服务程序的入口地址从中断向量表中读出并装入程序计数器。
5. 执行保存在中断服务程序入口地址处的指令。

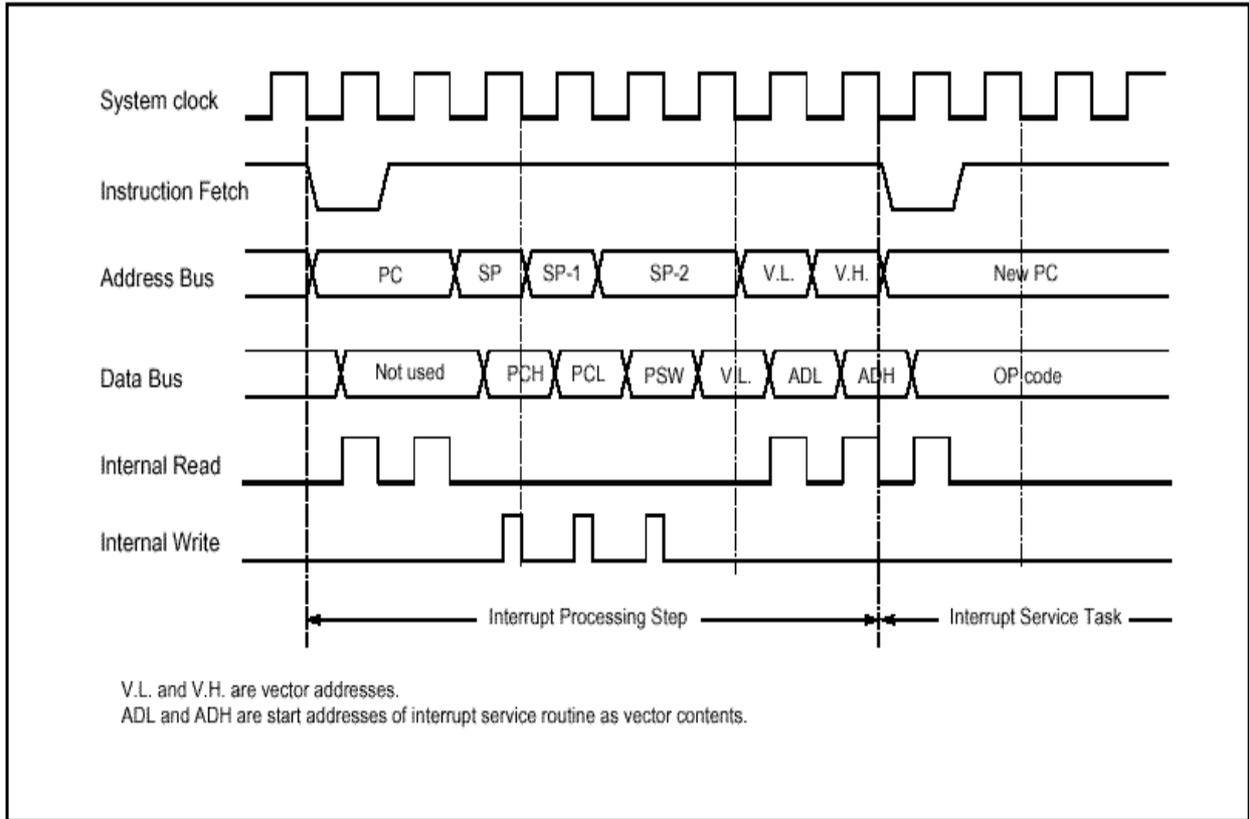
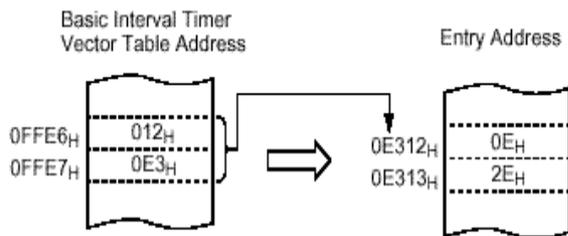


图 18-3 中断响应和中断返回指令时序



Correspondence between vector table address for BIT interrupt and the entry address of the interrupt service program.

即使一个请求的中断优先级高于正在处理的中断，也不会接受中断请求，除非I-flag标志置位。

当需要实现中断嵌套时，I-flag标志应在中断服务程序中由EI指令置位。这种情况下，可通过单独的中断使能标志使能可接受的中断。

**保存/恢复通用寄存器**

在中断响应处理中，程序计数器和程序状态字由堆栈自动保存。但不会保存累加器和其它寄存器。如果有必要，可通过软件保存。多个中断嵌套时，应当避免使用同一个数据存储区保存寄存器。

下面的方法可用于保存/恢复通用寄存器。

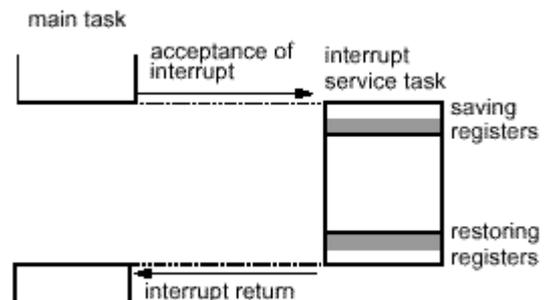
例：使用PUSH和POP指令保存寄存器

```
INTxx:  PUSH  A    ;SAVE ACC.
        PUSH  X    ;SAVE X REG.
        PUSH  Y    ;SAVE Y REG.
```

interrupt processing

```
POP  Y    ;RESTORE Y REG.
POP  X    ;RESTORE X REG.
POP  A    ;RESTORE ACC.
RETI     ;RETURN
```

通用寄存器的保存/恢复使用压入和弹出指令进行。



### 18.2 BRK 中断

BRK 指令可调用软件中断。该中断为最低优先级。

BRK 的中断向量地址与TCALL 0 的向量地址相同(参看程序寄存器章节)。当产生BRK 中断时, PSW 的位B-flag置位, 以使BRK区别于TCALL 0。由B标志决定的每一个步骤, 如图18—4 所示。

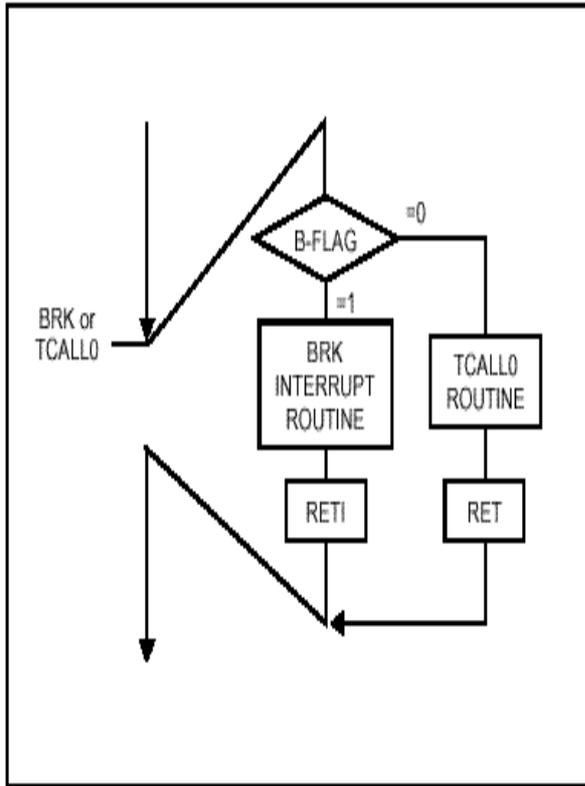


图18—4 BRK、TCALL 0 指令的执行

### 18.3 多中断

如果同时收到两个不同优先级中断请求,则响应较高优先级的中断。如果同时收到同一个优先级的中断请求, 则由硬件内部查询序列决定响应哪一个中断。

不过, 可通过软件实现多中断处理。通常, 当响应一个中断时, I-flag 标志清零, 以禁止其它中断。但用户可在中断服务程序中置位I-flag, 这样即使正在中断处理当中, 也可以响应其它的中断。

例: 尽管定时器1 中断正在处理当中, INTO中断仍得到立即响应。

```
TIMER1: PUSH  A
        PUSH  X
        PUSH  Y
        LDM  IENH, # 80H ; 仅使能INT0
        LDM  IENL, # 0   ; 禁止其它中断
        EI    ; 使能中断
        :
        :
        :
        :
        LDM  IENH, # 0FFH ; 使能所有中断
        LDM  IENL, # 0F0H
        POP  Y
        POP  X
        POP  A
        RETI
```

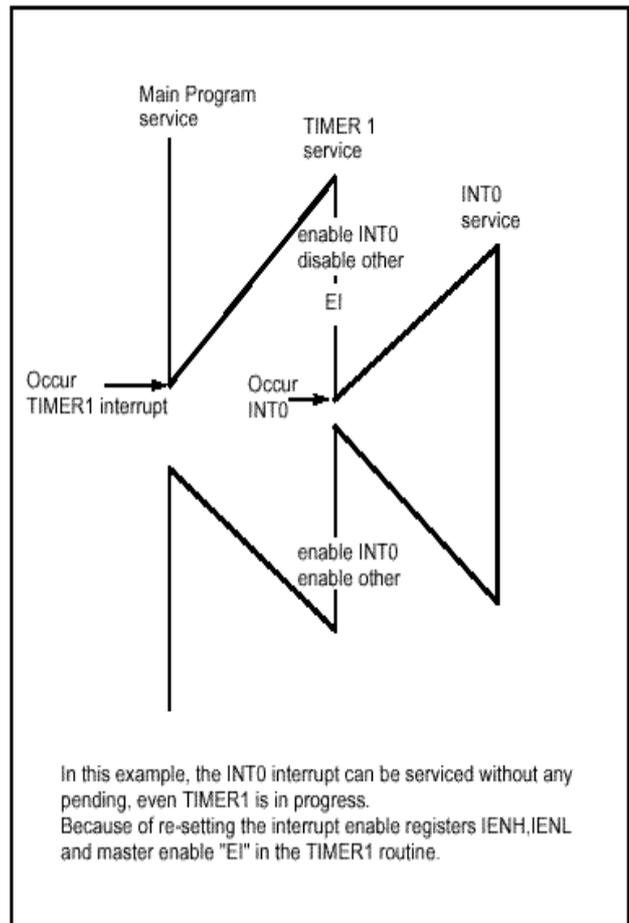


图 18-5 多中断的执行

### 18.4 外部中断

外部中断 INT0 和INT1为边沿触发，可通过边沿选择寄存器IEDS(地址0E6<sub>H</sub>)选择触发，如图18—6所示。

外部中断的边沿有3种触发方式，上升沿，下降沿，和双边沿。

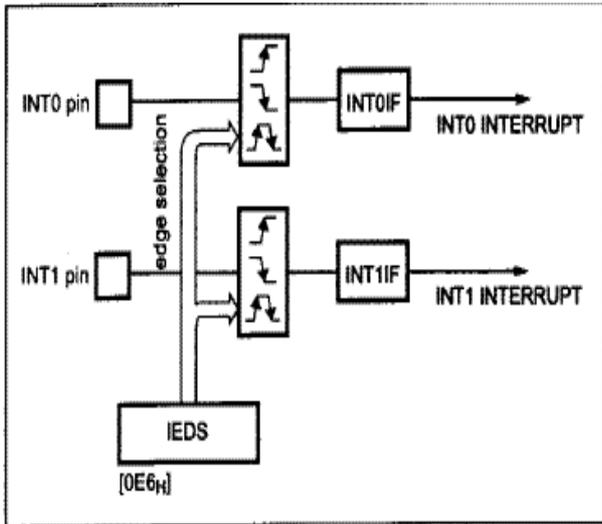


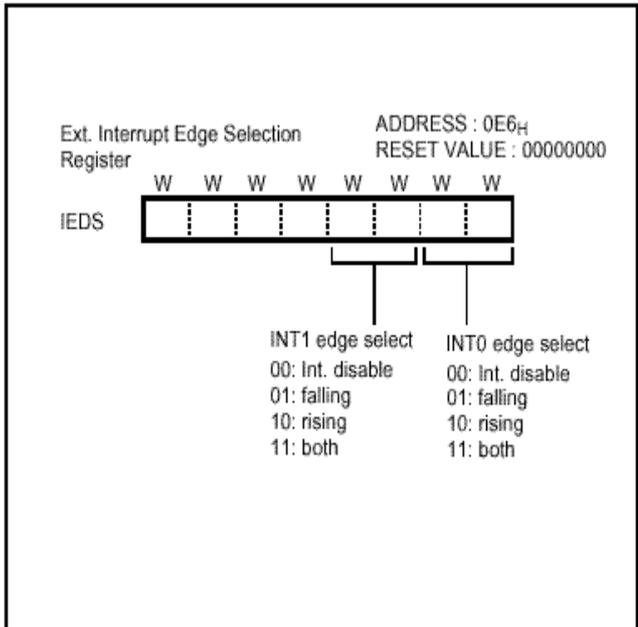
图18—6 外部中断方框图

例：:使用 INT0和 INT1

```

:
:
;****设置 RB2,RB3为输入口
LDM RBIO,
#1111_0011B
;
;
;**** 设置中断口
LDM RBFUNC, #0C0H
;
;
;**** 设置下降沿
LDM IEDS,
#0000_0101B

```



#### 响应时间

INT0 和INT1 的边沿在每个机器周期锁存到 INT0IF 和INT1IF。在下一个机器周期到来之前，不会再由电路查询。如果一个请求被激活并且响应的条件也正确，那么下一条要执行的指令将会是调用所请求的中断服务。程序DIV自身需要12个周期。因此，从外部中断激活到开始执行中断程序的第一条指令之间，至少经过12个机器周期。

下面为中断响应时序。

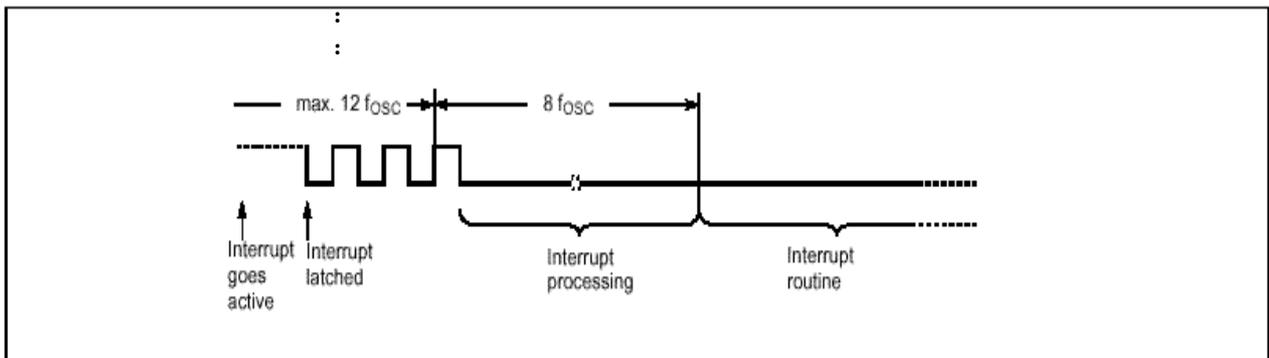


图18—7 中断响应时序框图

### 19. 看门狗定时器

看门狗定时器的目的，是检测由于外部噪声或其它原因引起的程序故障，并使程序回到正常情况下的运行。

看门狗定时器可以有两种类型的时钟源。

一个是不需要任何外部元件的片内RC振荡器。RC振荡器独立于从Xin脚输入的外部时钟电路。这意味着即使停止从Xin输入时钟，看门狗定时器仍然运行。例如，进入STOP模式。

另一种为经过预分频的系统时钟。

看门狗定时器包括7位二进制计数器和看门狗定时器。数据寄存器基本间隔定时器的溢出作为WDT的时钟源。当7位二进制计数器等于WDTR的低7位时产生中断，请求标志根据位WDTON的状态确定用作WDT中断或是CPU复位信号。

*注：由于看门狗定时器是在清基间隔定时器清零之后，位WDTON置位后设为使能，定时器的最大错误取决于基本间隔定时器的预分频设定。*

7位二进制计数器通过置位WDTCL( WDTR.7) 清零，WDTCL在一个机器周期后自动清零。

通过对CKCTLR的位RCWDT置位，使RC振荡看门狗定时器动作，并执行STOP指令如下。

```

:
LDM CKCTLR, #3FH ; 使能RC WDT
LDM WDTR, #0FFH ; 设置WDT 周期
STOP ; 进入STOP 模式
NOP
NOP ; RC WDT 运行
:
    
```

RC振荡周期随温度，V<sub>DD</sub>和制造的变化而不同(大约为120~180 uS)。下面的等式为RC振荡看门狗的溢出时间。

$$T_{RCWDT} = CLK_{RC} \times 2^8 \times [WDTR.6 \sim 0] + (CLK_{RC} \times 2^8) / 2$$

where,  $CLK_{RC} = 120 \sim 180 \mu s$

另外，看门狗定时器还可通过中断WDTIF作为一个7位定时器。看门狗定时器中断的间隔由基本间隔定时器确定算式如下：

$$T_{WDT} = [WDTR.6 \sim 0] \times Interval\ of\ BIT$$

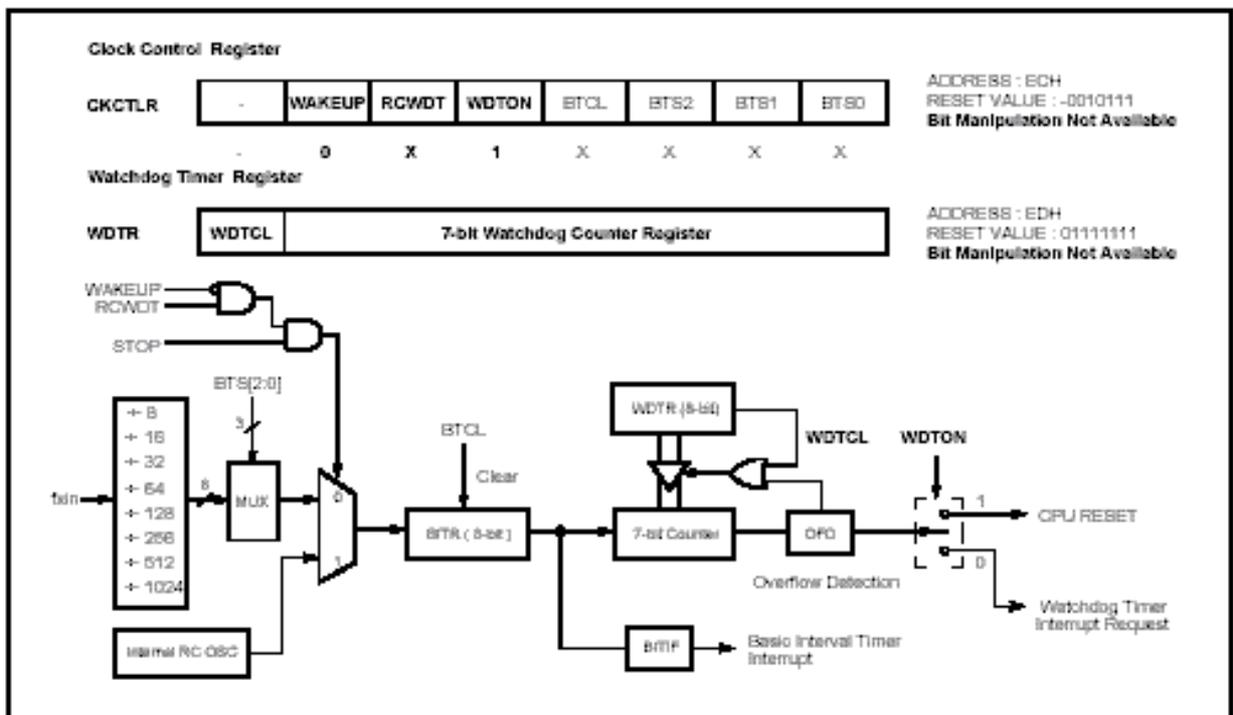


图 19—1 看门狗定时器方框图

## 20. 节电模式

在应用当中，功耗是一个很重要的指标。GMS87C1202 提供三种节电模式。STOP 模式，唤醒定时器模式和内部RC 看门狗定时器 模式。

在设置CKCTLR 中相应位( WAKEUP, RCWDT )后执行STOP 指令，就进入节电模式。

表20—1 列出了每一种节电模式的状态。

**注：执行STOP 指令之前，清零所有中断请求标志。如果在STOP 指令之前将中断标志置位，MCU 在STOP 指令之后仍然继续运行。因此，必须在STOP 令执行之前清除所有中断标志( IRQH, IRQL )。**

外围功能	STOP 模式	唤醒定时器模式	内部RC—WDT 模式
RAM	保持	保持	保持
控制寄存器	保持	保持	保持
IO 口	保持	保持	保持
CPU	停止	停止	停止
定时器0	停止	振荡	停止
振荡	停止	振荡	停止
预分频器	停止	÷2048 only	停止
内部RC振荡器	停止	停止	振荡
输入条件 CKCTLR[6, 5]	00	IX	01
唤醒源	RESET, INT0, INT1	RESET, INT0, INT1, 定时器	RESET, INT0, INT1, RC-WDT

表 20—1 节电模式

### 20.1 STOP 模式

在STOP 模式中，片内振荡器停振。由于时钟冻结，所有的功能都停止。只有片内RAM 和控制寄存器保持。口输出值由各自口数据寄存器和口方向寄存器保持。振荡器以及系统所有操作都停止。

- 系统进入STOP 模式后，将保持RAM, 寄存器和有效锁存保持进入停止模式之前的状态。
- 在STOP 指令启动STOP 模式后，程序计数器停止在下一条要执行的指令地址。

将CKCTLR 的位WAKEUP 和RCWDT 设置成00 后，执行STOP 指令就进入STOP 模式。( 该寄存器应当通过字节指令进行写操作。如果使用位操作指令，例如set1 或clr1 指令将不会得到所期望的操作。)

在STOP 模式操作中， $V_{DD}$  可下降以减少功耗。必须注意的是，在进入STOP 模式之前不要降低  $V_{DD}$ 。在STOP 模式终止之前，应当将  $V_{DD}$  恢复到正常操作电压。

$V_{DD}$  恢复到正常操作电压之前，复位信号无效。必须将信号保持足够长的时间以使振荡器启动并稳定下来。

**注：在STOP 指令之后至少应当有两条NOP 指令。**

```
LDM CKCTLR #0000_1110B
LDM IRQH,#0
LDM IRQL,#0
STOP 在STOP
NOP
NOP
```

在STOP 操作中，与振荡器和内部硬件相关的功率损耗降低了，但与管脚接口相关的功率损耗并不直接由STOP 模式的硬件操作决定( 而是取决于外部电路和程序 )。当输入电平稳定在电源电压( $V_{DD}/V_{SS}$ ) 时，几乎没有电流。但当输入电平高于电源电压(大约0.3~0.5V) 时开始有电流。

### 退出STOP 模式

通过硬件复位和外部中断可退出STOP模式。复位将重新定义所有的控制寄存器但不改变片内RAM。外部中断不改变控制寄存器和片内RAM 的值。

退出STOP模式后，根据标志 I( PSW.2 )的不同而出现两种可能。

如果 I=1, 响应中断；如果I= 0 执行STOP 指令的后一条指令。不进入中断服务程序（见 图20—1）。

由外部中断导致退出STOP 模式时，需要足够的时间使振荡器稳定，以恢复正常操作。图20—2为时序图。当退出STOP模式时，基本间隔定时器被唤醒，从00<sub>H</sub>开始计数，计数到FF<sub>H</sub>后溢出，并启动正常的操作。因此，在STOP指令之前，用户必须设置预分频比率，以保证有足够的时间（超过20ms），使振荡器启动，并稳定下来。由复位导致从 STOP 模式退出如图20—3所示。

### STOP 模式中的最小化电流消耗

STOP 模式设计用于降低功耗。在STOP 模式中，为了将电流消耗降到最低，用户在可能的情况下，应将拉电流或灌电流输出驱动器关闭。I/O口的弱上拉也应关闭。所有的输入应为 V<sub>SS</sub> 或 V<sub>DD</sub>。

输入管脚上的中间电压，将会使输入缓冲器产生相当大的电流。

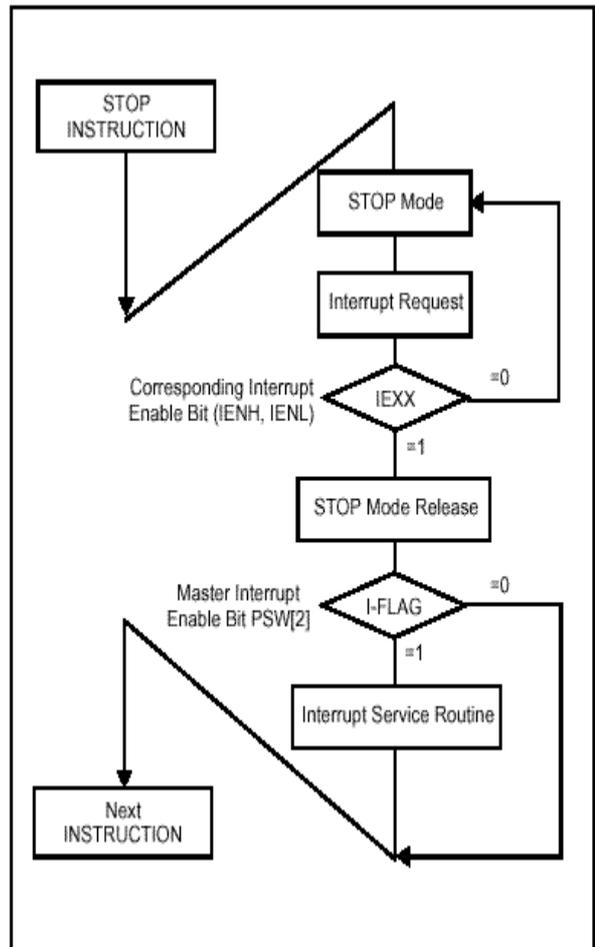


图 20-1 中断导致退出STOP模式流程图

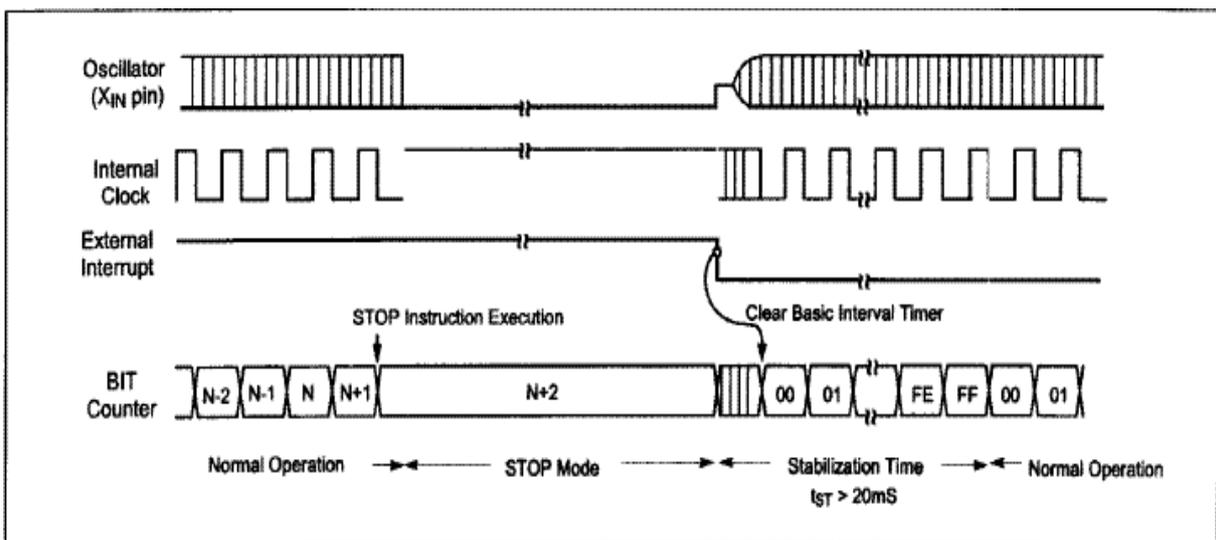


图 20—2 外部中断导致退出STOP 模式时序

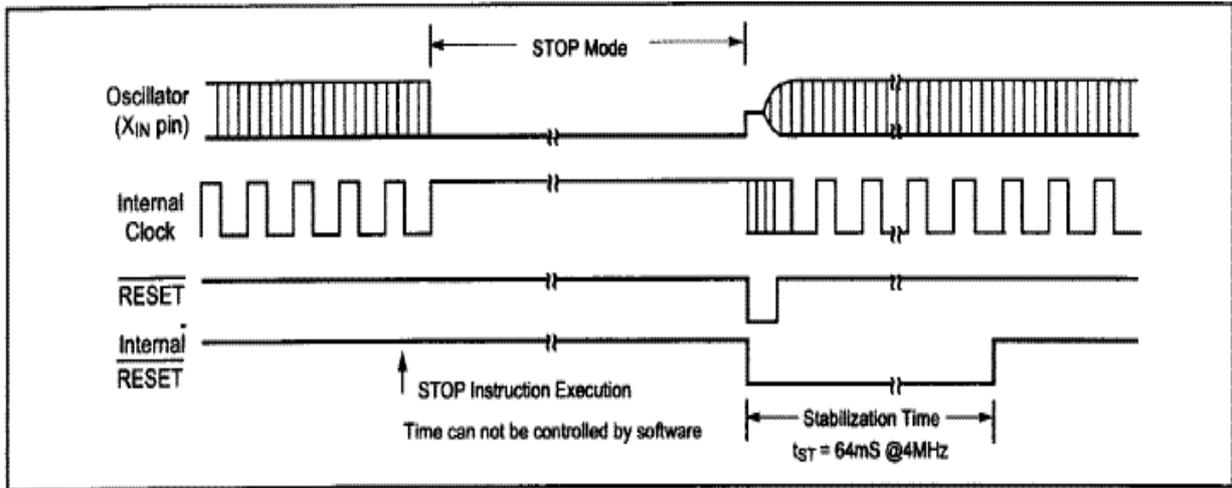


图20—3 复位导致退出STOP 模式时序

## 20.2 唤醒定时器模式

在唤醒定时器模式中，片内振荡器并不停振。除了预分频器(仅为2048)和定时器0外，所有功能都停止工作。但片内RAM和控制寄存器的内容将会保持。管脚的输出电平由各自口的数据寄存器和方向寄存器保持。

在将CKCTLR中WAKEUP置位后，执行STOP指令就进入唤醒定时器模式(该寄存器应当通过字节指令进行写操作。如果使用位操作指令，例如set1或clr1指令，则不会得到所期望的操作)。

**注：**在STOP指令后，至少应写2条NOP指令。

```
例：LDM TDR0, #0FFH
      LDM TM0, #0001_1011B
      LDM CKCTLR, #0100_1110B
      LDM IRQH, #0
      LDM IRQL, #0
      STOP
      NOP
      NOP
```

另外，定时器0的时钟源应选择预分频率为2048。否则唤醒定时器功能将不能工作。定时器0可以和定时器1一起作为16位定时器工作(参考定时器功能)。通过设定定时器数据寄存器TDR0，可得到不同的唤醒周期。

### 退出唤醒定时器模式

通过硬件复位定时器0溢出和外部中断，可退出唤醒定时器模式。复位将重新定义所有的控制寄存器，但不会改变片内RAM。外部中断和定时器0溢出都不改变控制寄存器和片内RAM的值。

如果I—标志=1，将响应中断。如果I—标志=0，则执行STOP令的后一条指令，不进入中断服务程序(见图20—1)。

由外部中断或定时器0溢出导致退出唤醒定时器模式时，不需要稳定振荡时间来恢复正常操作。因为该模式中片内振荡器并不停振，见图20—4。

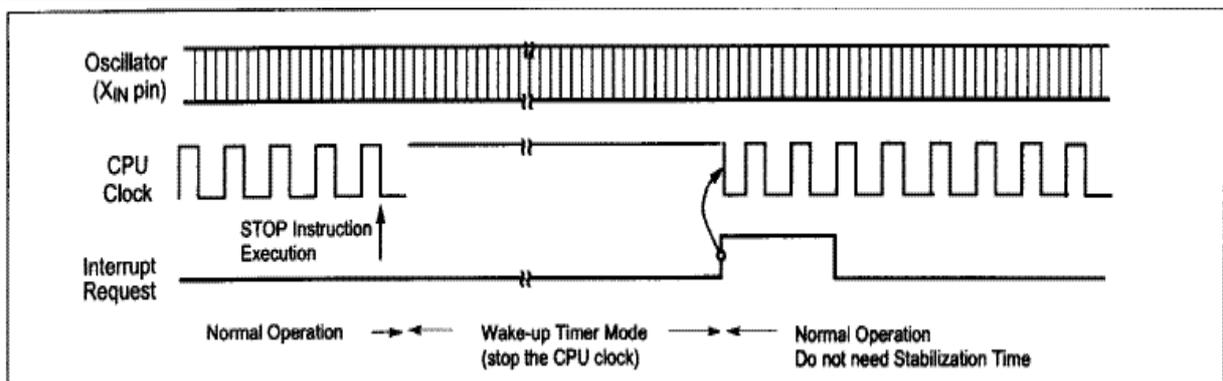


图20—4 外部中断或定时器0 导致退出唤醒定时器模式时序

### 20.3 内部RC-Oscillated WDT(看门狗定时器)模式

在内部RC WDT 模式中，片内振荡器停振。但内部 RC 振荡电路仍然工作。片内RAM 和控制寄存器的内容将会保持。管脚的输出电平由各自口的数据寄存器和方向寄存器保持。

在将CKCTLR 中WAKEUP 和RCWDT 设置成01后执行STOP 指令，就进入内部RC振荡看门狗定时器模式（该寄存器应当通过字节指令进行写操作。如果使用位操作指令，例如set1 或clr1 指令，将不会得到所期望的操作）。

**注：**在STOP 指令后，至少应写2条NOP 指令。

```
例：LDM  TDR0, #0FFH
      LDM  TM0, #0001_1011B
      LDM  CKCTLR, #0100_1110B
      LDM  IRQH, #0
      LDM  IRQL, #0
      STOP
      NOP
      NOP
```

#### 退出RC WDT 模式

通过硬件复位和外部中断，可退出RC WDT模式。复位将重新定义所有的控制寄存器，但不会改变片内RAM。外部中断不改变控制寄存器和片内RAM 的值。

如果I=1，则响应中断。在这种情况下，如果CKCTLR 中的位WDTON为0，且IENH中的位WDTE 为1，器件将会执行看门狗定时器中断服务程序（见图20—5）。如果CKCTLR 中的位WDTON 为1，则器件将产生内部复位信号，并进入复位处理过程（见图20—6）。

如果I=0，则执行STOP指令的后一条指令，而不进入中断服务程序（见图20—1）。

当由外部中断导致退出RC WDT 模式时，需要足够的时间使振荡器稳定以恢复正常操作。图20—5为时序图。当退出RC WDT 模式时，基本间隔定时器被唤醒，从00<sub>H</sub> 开始计数，计数到FF<sub>H</sub> 后溢出并启动正常的操作。因此，在STOP指令之前用户必须设置预分频比率以保证有足够的时间（超过20ms）使振荡器启动并稳定下来。

由复位导致从RC WDT 模式退出，如图20—6所示。

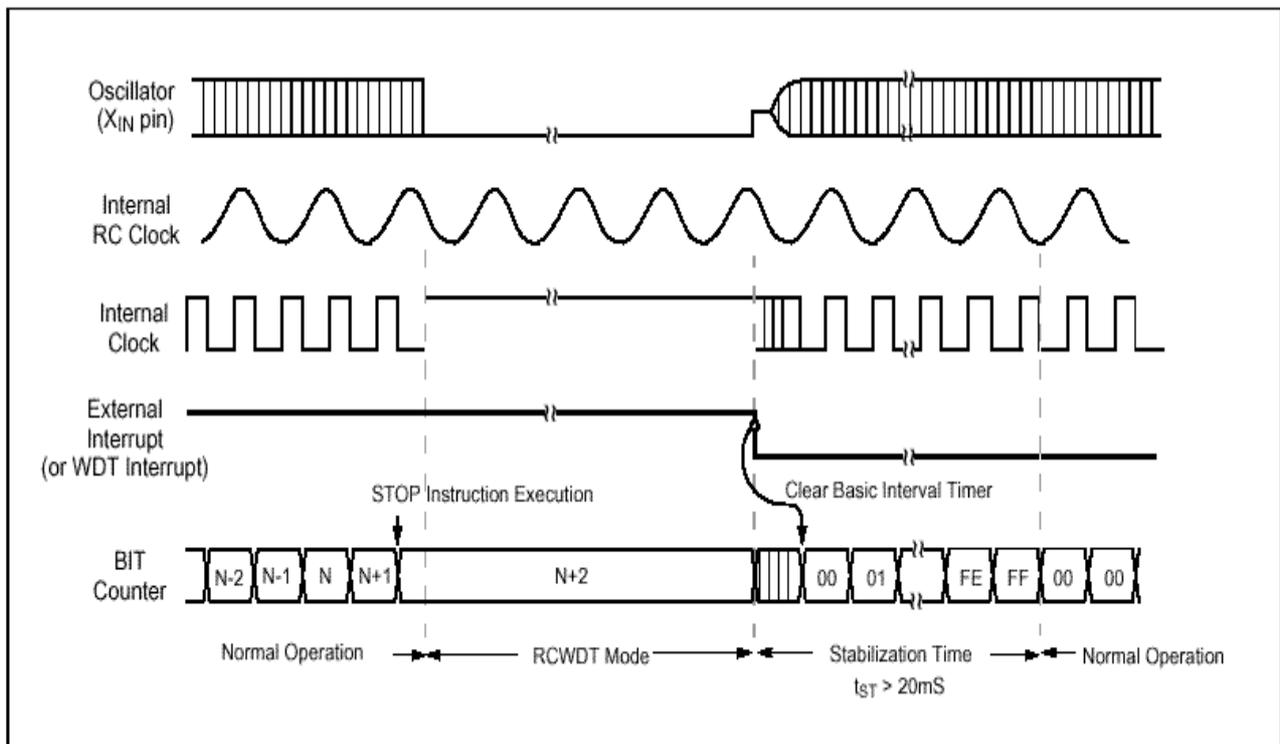


图 20—5 外部中断或WDT 中断导致退出内部RCWDT 模式时序

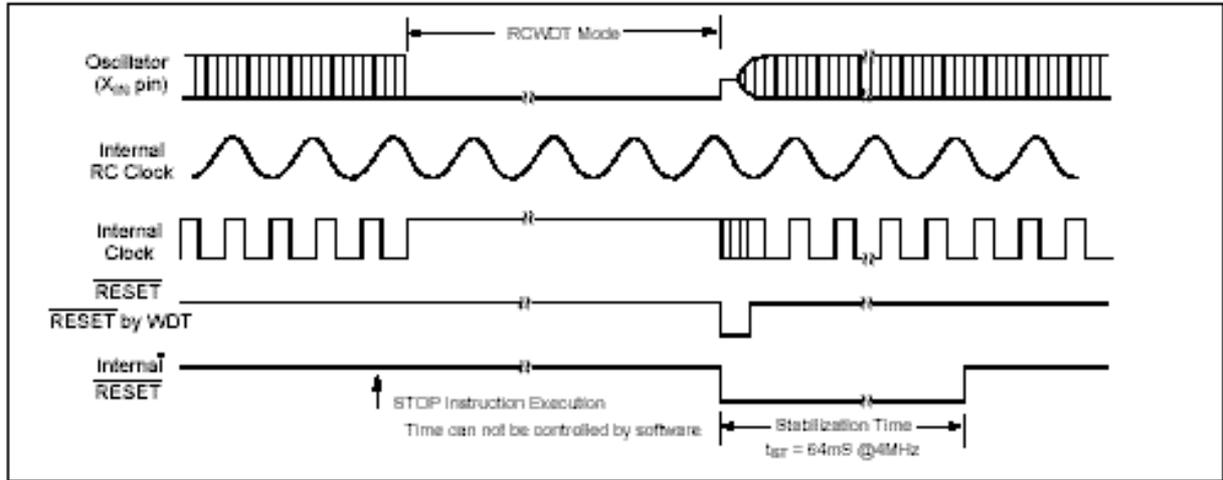


图 20—6 复位导致退出内部RCWDT 模式

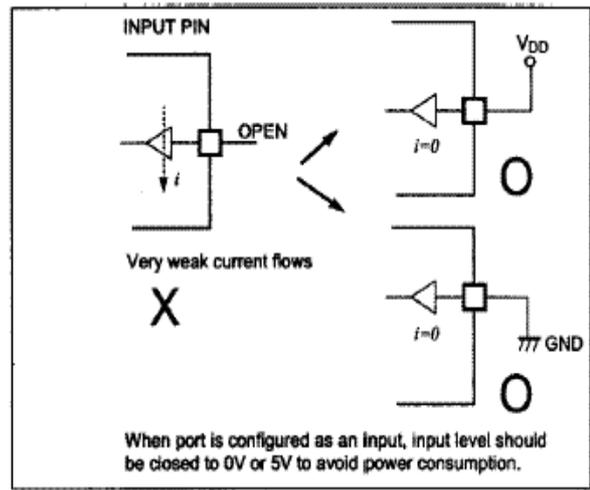
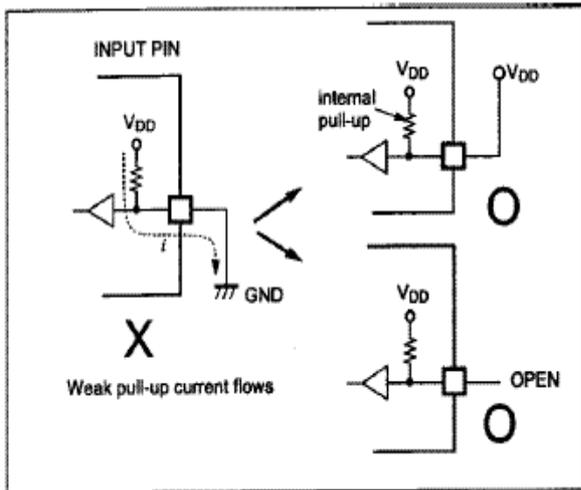


图 20—7 未使用的输入口应用示例

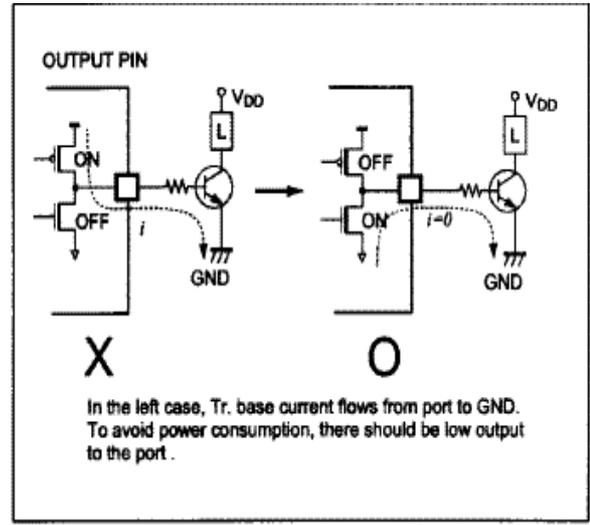
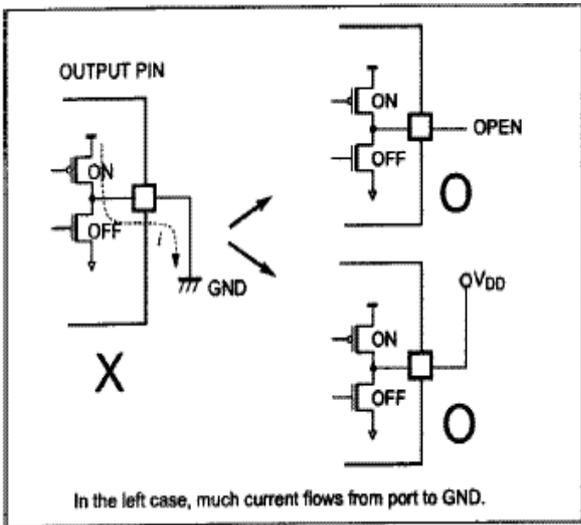


图20—8 未使用的输出口应用示例

## 21. 复位

复位信号从RESET脚输入，该输入为施密特触发。输入在振荡器工作时保持RESET脚低电平至少8个振荡器周期实现复位。复位后，需要64ms(4MHz时)加上7个振荡器周期以启动执行过程，如图21-1。复位不影响片内RAM的状态。故 $V_{DD}$ 上电时，RAM的内容是不确定的。因此，RAM应当在读取或测试之前进行初始化。

每个寄存器的初始化状态见表11-3。

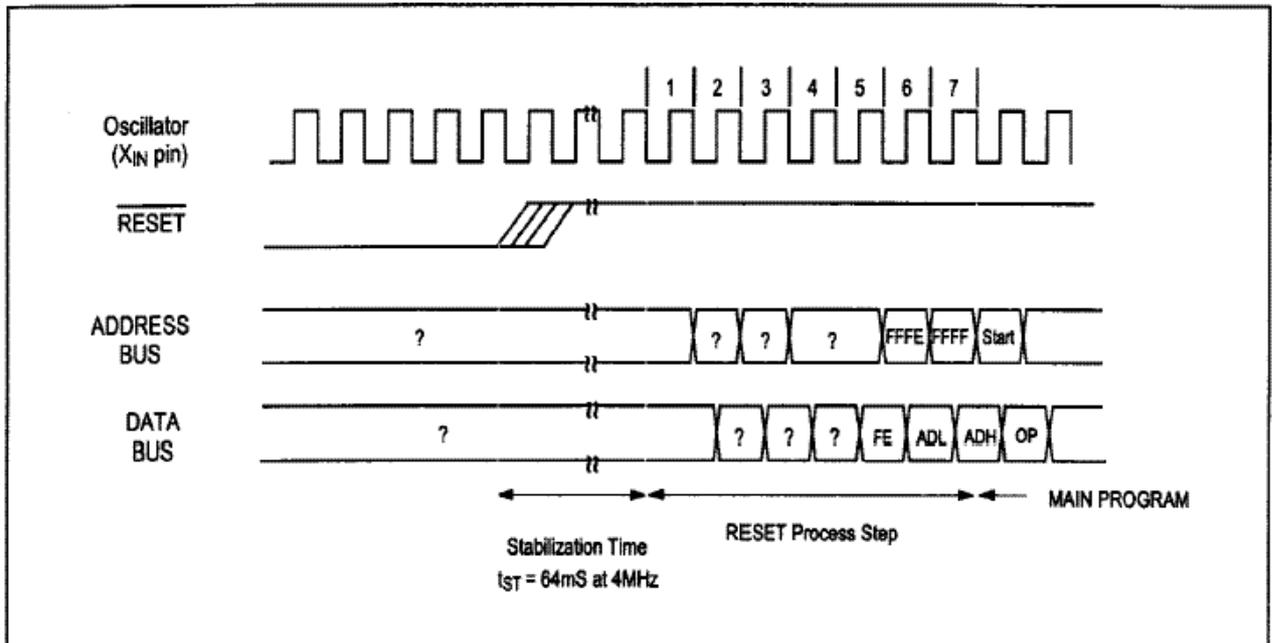


图21-1 复位后的时序

## 22. 电源失效处理

GMS87C1202有一个片内电源失效检测电路用于消除电源噪声。通过配置寄存器，PFDR，可使能或禁止电源失效检测电路。如果 $V_{DD}$ 降到3.0~4.0V超过50ns，电源失效处理将根据PFDR中的位PFS决定是否复位MCU。

由于PFDR寄存器不能由电路仿真器实现，故用户无法对其进行试验。因此在用户编程的最后阶段才进行试验。

*注：在3V操作电压下不能使用电源失效检测，因为此时该功能将总是检测电源为失效状态。*

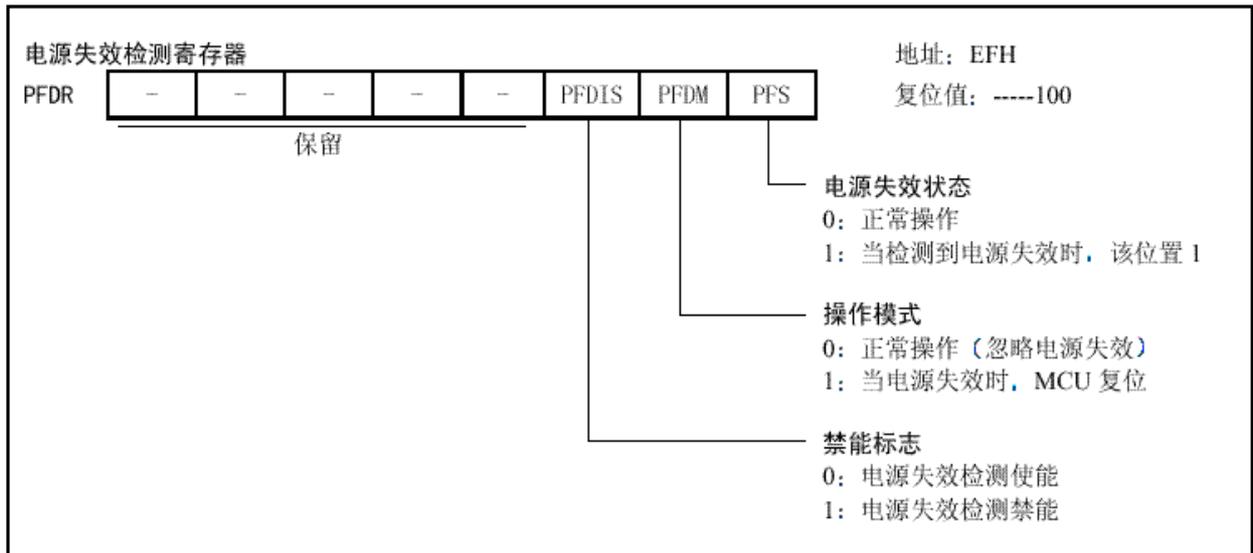


图22-1 电源失效检测寄存器

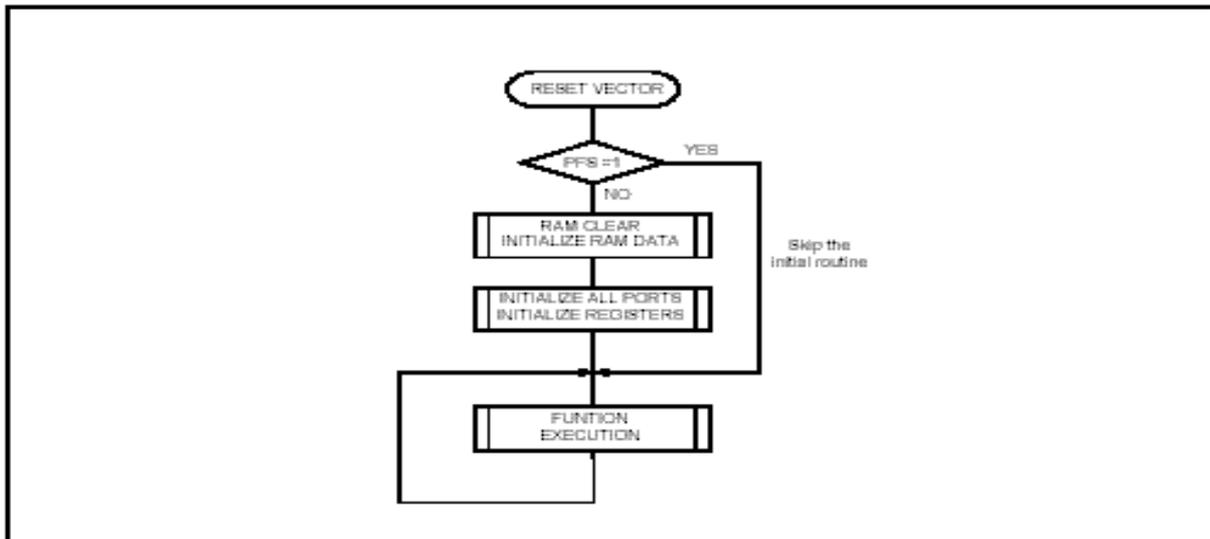


图22-2 电源失效引起的复位软件的例子

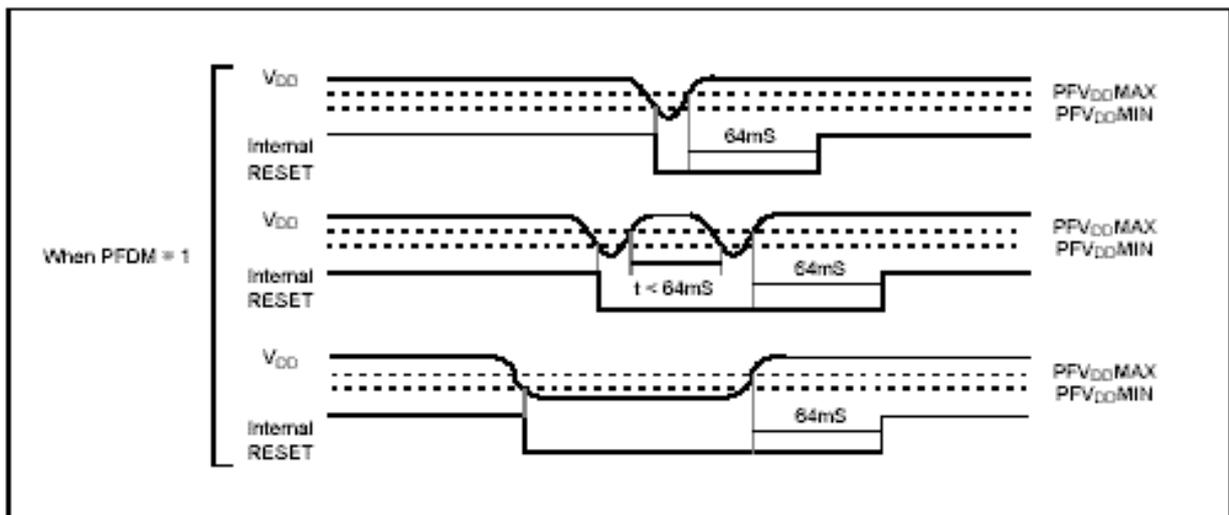


图22-3 电源失效处理状态

## 23. OTP程序设计

GMS87C1102/1202T芯片，为一次性可编程单片机。本芯片具有2K字节的可编程的只读存储区。

为了给OTP芯片编程，用户必须使用支持Hynix半导体公司产品的编程器。

### 23.1 程序存储图

程序存储区由程序结构区和用户程序存储区构成。结构区有( 用户ID和系统结构位 )两部分，如图23—1所示。

器件结构区可编程，亦可保留不编程，以用于选择器件器件结构，如保密位等。

用户ID有10个存储位( 0F50<sub>H</sub>~0FE0<sub>H</sub> )，用户可在其中保存检测数或其他用户识别数。这一区域在正常执行期间不可访问，但在编程和修改程序时可读可写。

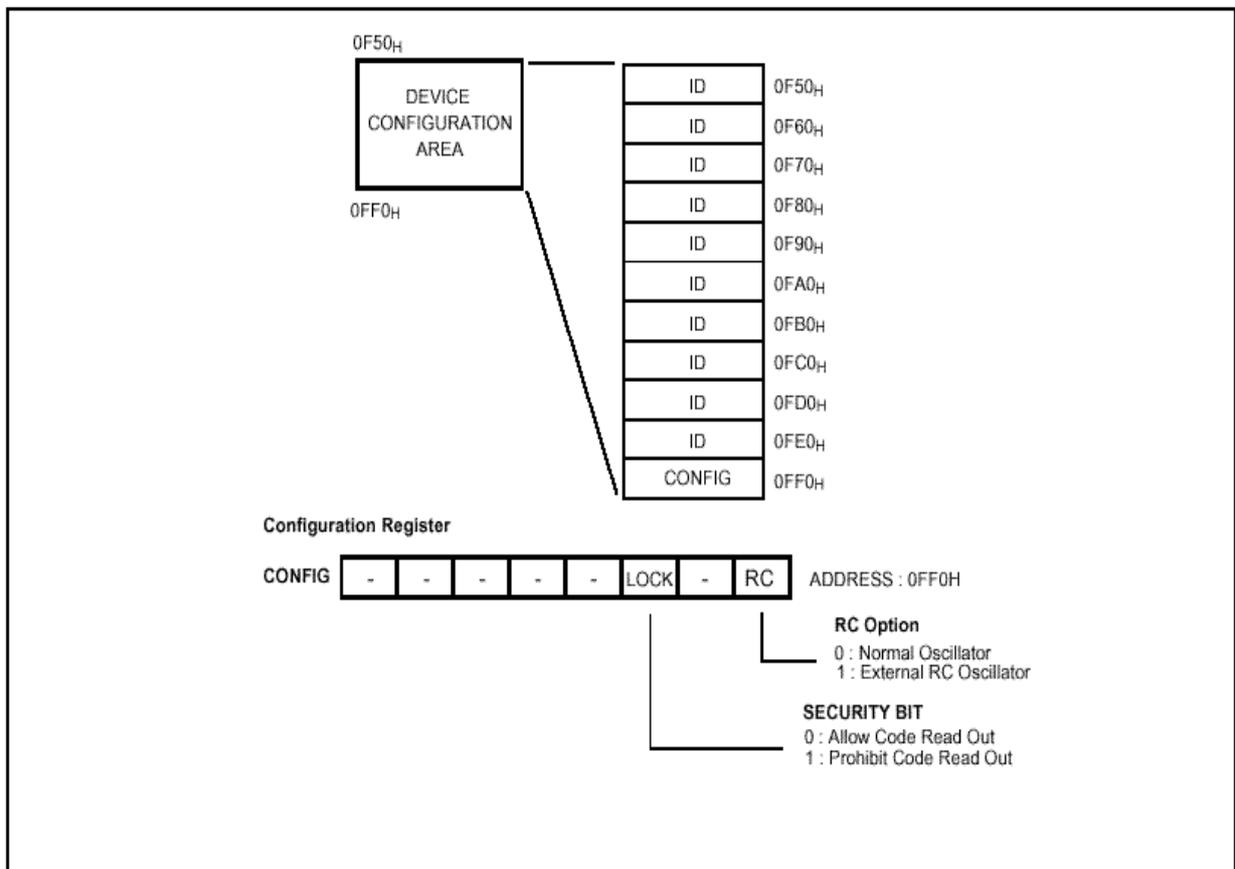


图23—1 器件结构区

密码的定义方法描述如下。

- 1) 在写，修改模式和退出写和修改模式中，为了给保密位编码，在写号H字符之后，用户不可读出程序码。但是，如果尚没退出写和修改模式(保持编程电源VPP=12.75V)，用户可以更改程序码。
- 2) 不必考虑代码的保护密码，用户可读出结构区(用户ID，和结构位)。
- 3) 如果用户了解密码(锁)的状态，用户可以从系统结构位读出密码位，

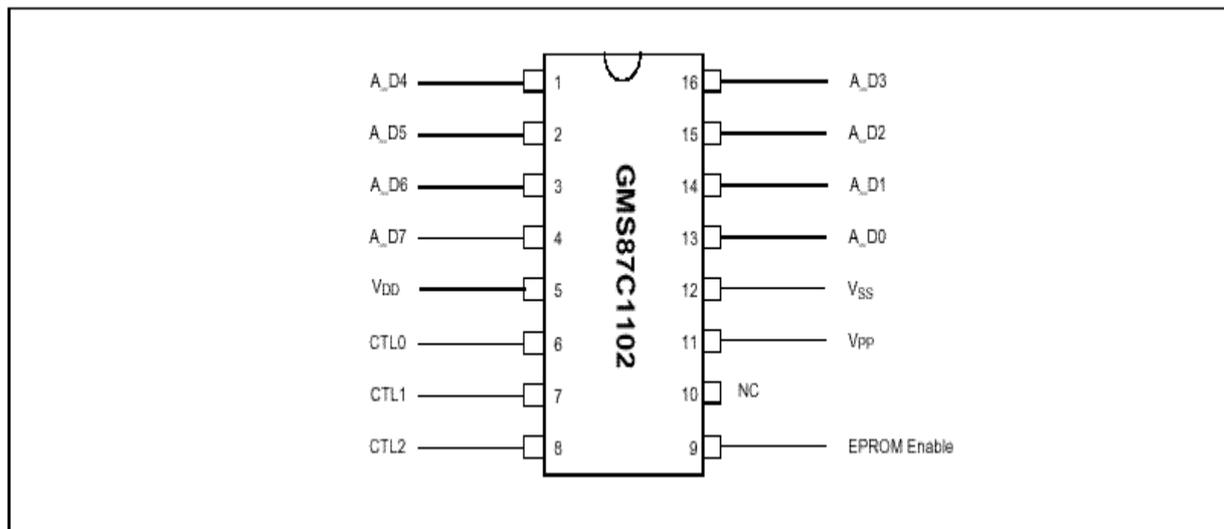


图23—2 管脚定义图

管脚号	用户模式	EPROM模式				
	管脚名称	管脚名称	描述			
1	RA4 (AN4)	A	地址 输入 数据 输入/ 输出	A12	A4	D4
2	RA5 (AN5)	A		A13	A5	D5
3	RA6 (AN6)	A		A14	A6	D6
4	RA7 (AN7)	A		A15	A7	D7
5	V <sub>DD</sub>	V <sub>DD</sub>	连接到V <sub>DD</sub> ( 6.0V)			
6	RB0 (Avref/AN0)	CTL0	读/写 控制 地址/数据 控制			
7	RB2 (INT0)	CTL1				
8	RB4 (PWM/COMP)	CTL2				
9	X <sub>IN</sub>	EPROM 不能	下降沿中的高位活跃，加锁地址			
10	X <sub>OUT</sub>	NC	未连			
11	RESET	V <sub>PP</sub>	编程电源 (0V, 12.75)			
12	V <sub>SS</sub>	V <sub>SS</sub>	连接到V <sub>SS</sub> ( 0V)			
13	RA0 (ECO)	A	地址 输入 数据 输入/ 输出	A8	A0	D0
14	RA1 (AN1)	A		A9	A1	D1
15	RA2 (AN2)	A		A10	A2	D2
16	RA3 (AN3)	A D3		A11	A3	D3

表23—1 EPROM模式中的管脚描述

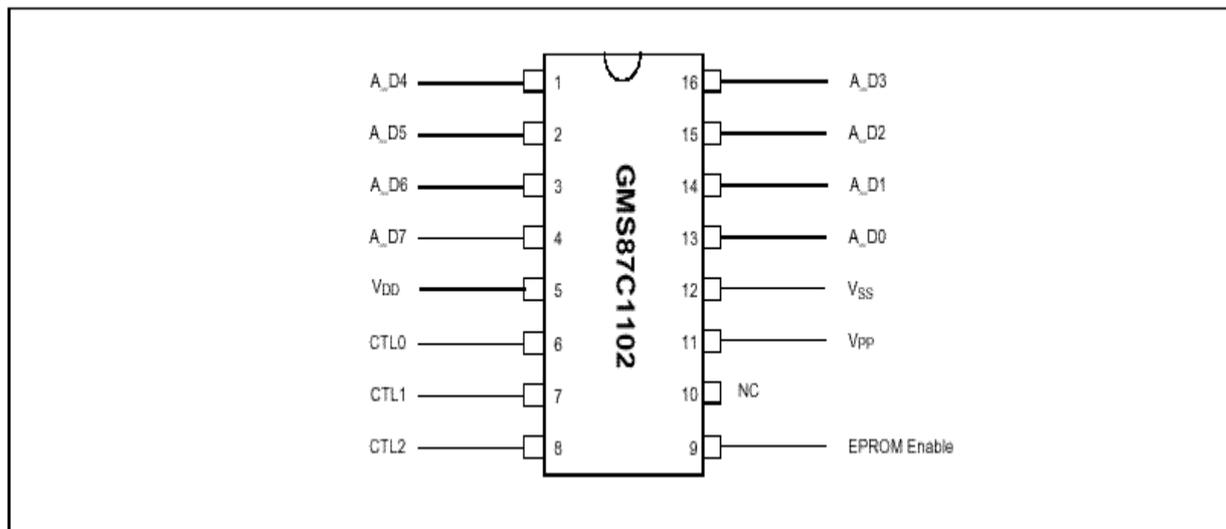


图23—3 管脚定义图

管脚号	用户模式	EPROM模式				
	管脚名称	管脚名称	描述			
1	RA4 (AN4)	A D4	地址 输入 数据 输入/ 输出	A12	A4	D4
2	RA5 (AN5)	A		A13	A5	D5
3	RA6 (AN6)	A		A14	A6	D6
4	RA7 (AN7)	A		A15	A7	D7
5	V <sub>DD</sub>	V <sub>DD</sub>	连接到V <sub>DD</sub> ( 6.0V)			
6	RB0 (Avref/AN0)	CTL0	读/写 控制 地址/数据 控制			
7	RB1 (BUZ)	CTL1				
8	RB2 (INT0)	CTL2				
9	RB3 (INT1)	V <sub>DD</sub>	连接到V <sub>DD</sub> ( 6.0V)			
10	RB4 (PWM/COMP)	V <sub>DD</sub>	连接到V <sub>DD</sub> ( 6.0V)			
11	X <sub>IN</sub>	EPROM 不能	下降沿中的高位活跃，加锁地址			
12	X <sub>OUT</sub>	NC	未连			
13	RESET	V <sub>PP</sub>	编程电源 (0V, 12.75)			
14	V <sub>SS</sub>	V <sub>SS</sub>	连接到V <sub>DD</sub> ( 0V)			
15, 16	RC0, 1	V <sub>DD</sub>	连接到V <sub>DD</sub> ( 6.0V)			
17	RA0 (ECO)	A	地址 输入 数据 输入/ 输出	A8	A0	D0
18	RA1 (AN1)	A		A9	A1	D1
19	RA2 (AN2)	A		A10	A2	D2
20	RA3 (AN3)	A D3		A11	A3	D3

表23—2 EPROM模式中的管脚描述

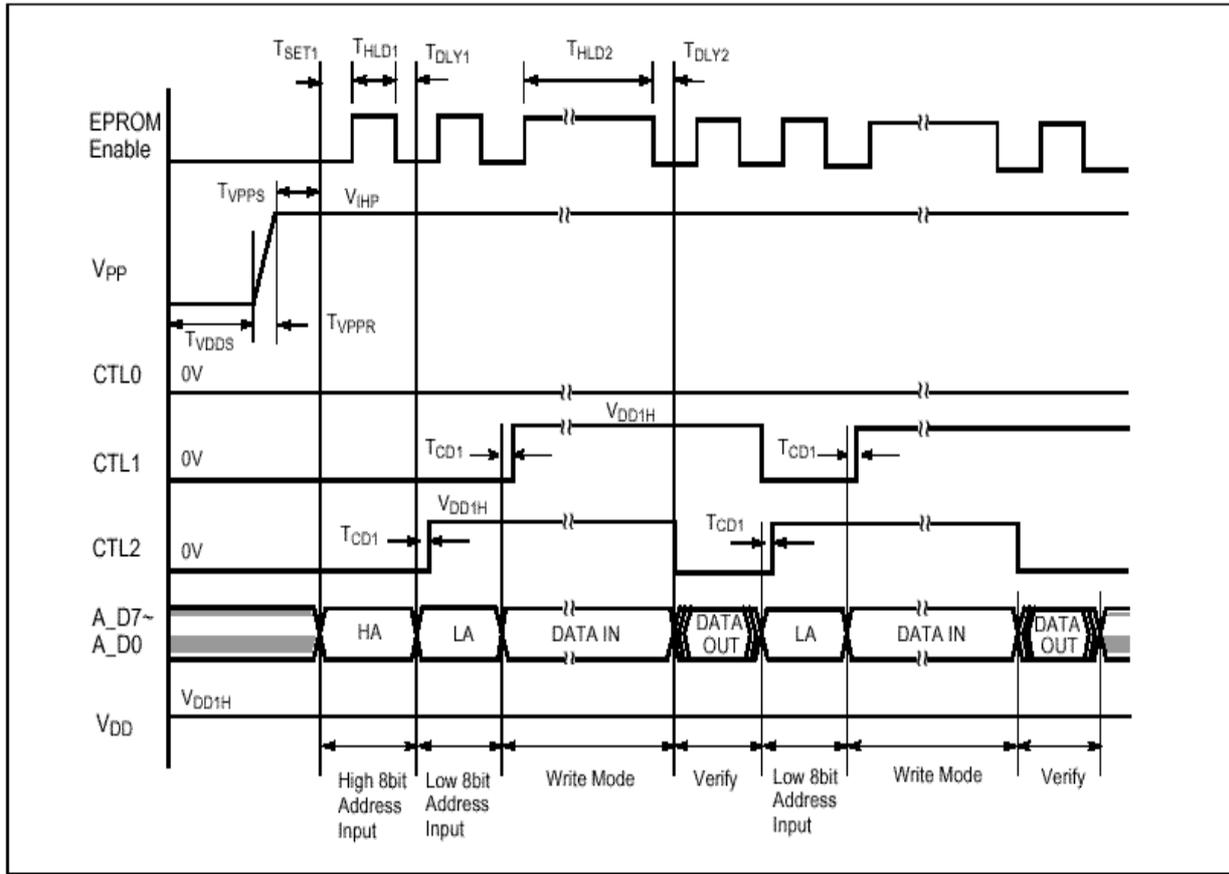


图23-4 编程（写和修改）模式中的时序图

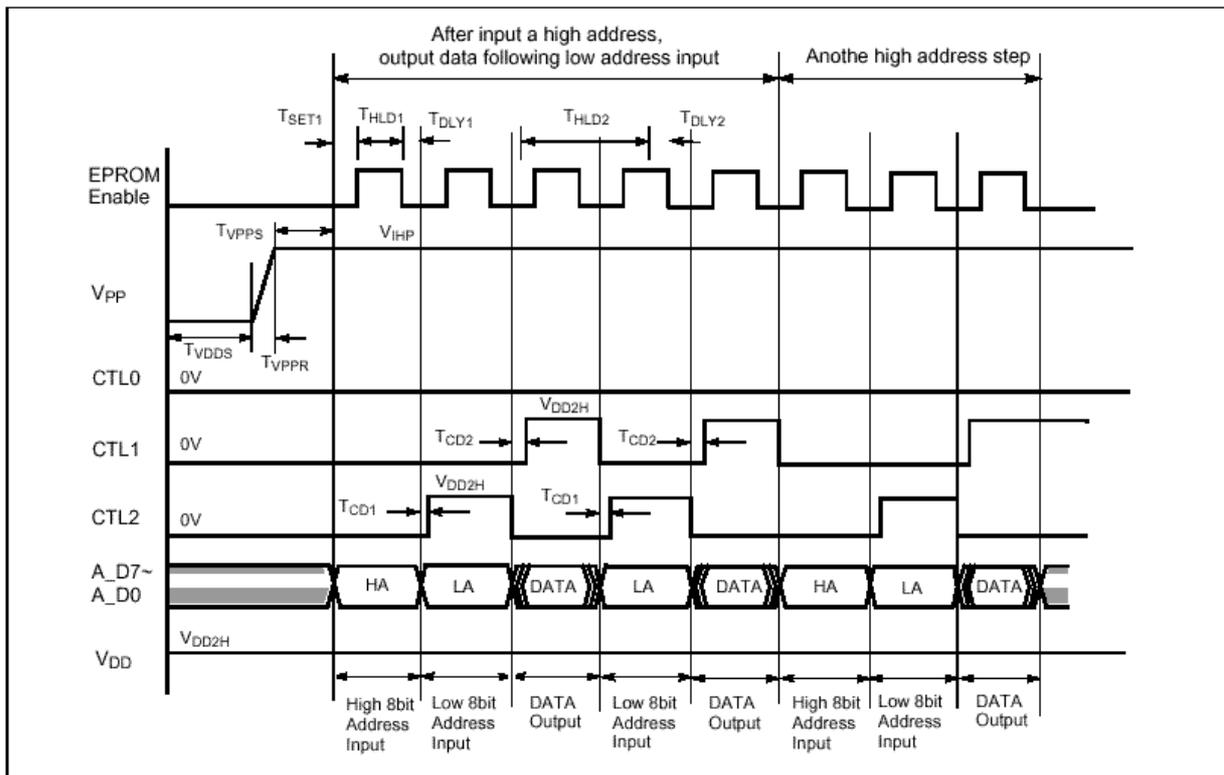


图23-5 读模式中的时序图

参数	符号	最小	典型	最大	单位
编程供电电流	$I_{VPP}$	-	-	50	mA
EPROM模式中供电电流	$I_{VDDP}$	-	-	20	mA
编程期间 $V_{PP}$ 级	$V_{IHP}$	12.0	12.5	13.0	V
程序模式中 $V_{DD}$ 级	$V_{DD1H}$	5	6	6.5	V
读模式中 $V_{DD}$ 级	$V_{DD2H}$	-	2.7	-	V
EPROM模式中CTL2~0高级	$V_{IHC}$	$0.8V_{DD}$	-	-	V
EPROM模式中CTL2~0低级	$V_{ILC}$	-	-	$0.2V_{DD}$	V
EPROM模式中A...D7~A_D0高级	$V_{IHAD}$	$0.9V_{DD}$	-	-	V
EPROM模式中A...D7~A_D0低级	$V_{ILAD}$	-	-	$0.1V_{DD}$	V
$V_{DD}$ 充满时间	$T_{VDDS}$	-	-	1	mS
$V_{PP}$ 安装时间	$T_{VPPR}$	-	-	1	mS
$V_{PP}$ 充满时间	$T_{VPPS}$	1	-	-	mS
数据输入后EPROM不能安装时间	$T_{SET1}$		200		nS
$T_{SET1}$ 后EPROM不能停止时间	$T_{HLD1}$		500		nS
$T_{HLD1}$ 后EPROM不能延迟时间	$T_{DLY1}$		200		nS
写模式中EPROM不能停止时间	$T_{HLD2}$		100		nS
$T_{HLD2}$ 后EPROM不能延迟时间	$T_{DLY2}$		200		nS
低位地址输入和数据输入后CTL2, 1安装时间	$T_{CD1}$		100		nS
读和修改模式中数据输出前CTL1安装时间	$T_{CD2}$		100		nS

表23-3 编程/读模式中的 AC/DC要求

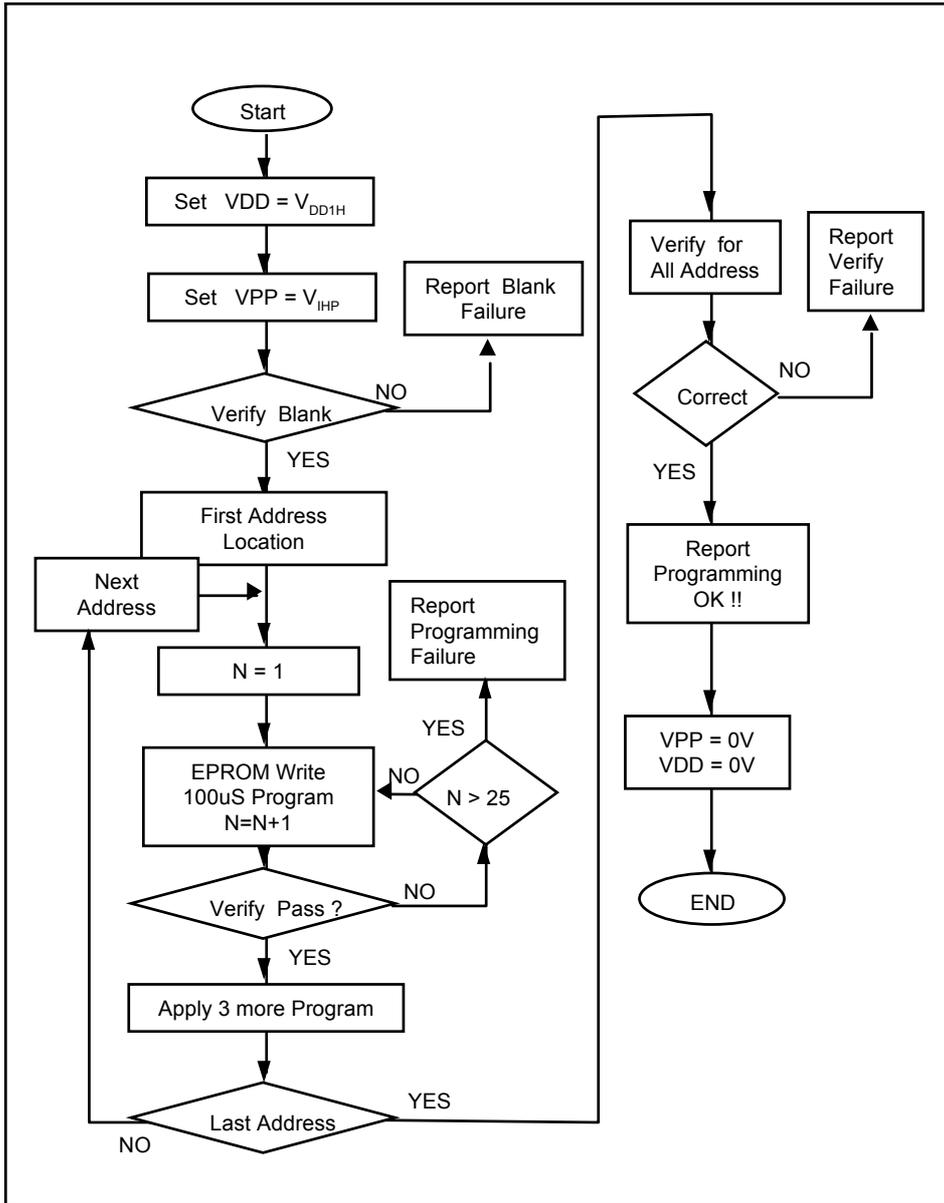


图23-6 编程流程图

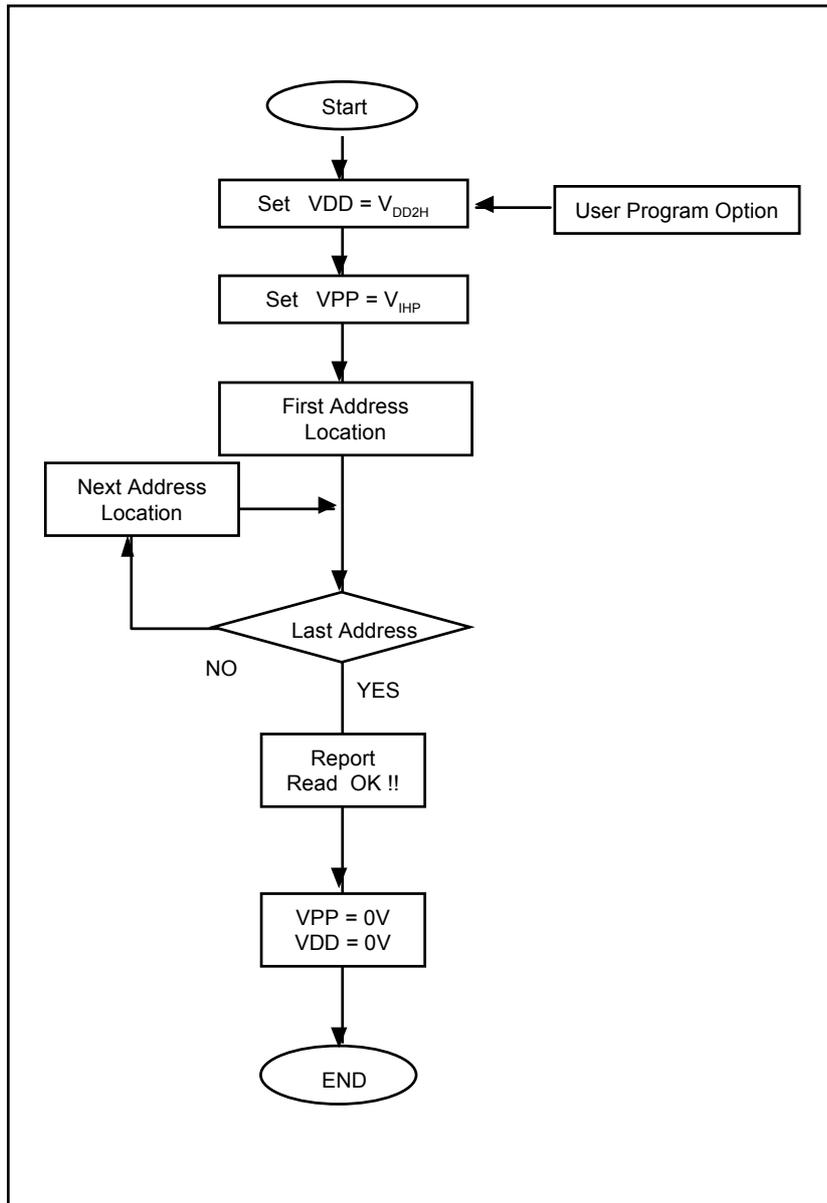


图23-7 读流程图

## A. 指令图

LOW HIGH	0000 00	00001 01	00010 02	00011 03	00100 04	00101 05	00110 06	00111 07	01000 08	01001 09	01010 0A	01011 0B	01100 0C	01101 0D	01110 0E	01111 0F
000	-	SET1 dp.bit	BBS A.bit,rel	BBS dp.bit,rel	ADC #imm	ADC dp	ADC dp+X	ADC !abs	ASL A	ASL dp	TCALL 0	SETA1 .bit	BIT dp	POP A	PUSH A	BRK
001	CLRC				SBC #imm	SBC dp	SBC dp+X	SBC !abs	ROL A	ROL dp	TCALL 2	CLRA1 .bit	COM dp	POP X	PUSH X	BRA rel
010	CLRG				CMP #imm	CMP dp	CMP dp+X	CMP !abs	LSR A	LSR dp	TCALL 4	NOT1 M.bit	TST dp	POP Y	PUSH Y	PCALL Upage
011	DI				OR #imm	OR dp	OR dp+X	OR !abs	ROR A	ROR dp	TCALL 6	OR1 OR1B	CMPX dp	POP PSW	PUSH PSW	RET
100	CLRv				AND #imm	AND dp	AND dp+X	AND !abs	INC A	INC dp	TCALL 8	AND1 AND1B	CMPY dp	CBNE dp+X	TXSP	INC X
101	SETC				EOR #imm	EOR dp	EOR dp+X	EOR !abs	DEC A	DEC dp	TCALL 10	EOR1 EOR1B	DBNE dp	XMA dp+X	TSPX	DEC X
110	SETG				LDA #imm	LDA dp	LDA dp+X	LDA !abs	TXA	LDY dp	TCALL 12	LDC LDCB	LDX dp	LDX dp+Y	XCN	DAS
111	EI				LDM dp,#imm	STA dp	STA dp+X	STA !abs	TAX	STY dp	TCALL 14	STC M.bit	STX dp	STX dp+Y	XAX	STOP

LOW HIGH	10000 10	10001 11	10010 12	10011 13	10100 14	10101 15	10110 16	10111 17	11000 18	11001 19	11010 1A	11011 1B	11100 1C	11101 1D	11110 1E	11111 1F
000	BPL rel	CLR1 dp.bit	BBC A.bit,rel	BBC dp.bit,rel	ADC {X}	ADC !abs+Y	ADC [dp+X]	ADC [dp]+Y	ASL !abs	ASL dp+X	TCALL 1	JMP !abs	BIT !abs	ADDW dp	LDX #imm	JMP [!abs]
001	BVC rel				SBC {X}	SBC !abs+Y	SBC [dp+X]	SBC [dp]+Y	ROL !abs	ROL dp+X	TCALL 3	CALL !abs	TEST !abs	SUBW dp	LDY #imm	JMP [dp]
010	BCC rel				CMP {X}	CMP !abs+Y	CMP [dp+X]	CMP [dp]+Y	LSR !abs	LSR dp+X	TCALL 5	MUL	TCLR1 !abs	CMPW dp	CMPX #imm	CALL [dp]
011	BNE rel				OR {X}	OR !abs+Y	OR [dp+X]	OR [dp]+Y	ROR !abs	ROR dp+X	TCALL 7	DBNE Y	CMPX !abs	LDYA dp	CMPY #imm	RETI
100	BMI rel				AND {X}	AND !abs+Y	AND [dp+X]	AND [dp]+Y	INC !abs	INC dp+X	TCALL 9	DIV	CMPY !abs	INCW dp	INC Y	TAY
101	BVS rel				EOR {X}	EOR !abs+Y	EOR [dp+X]	EOR [dp]+Y	DEC !abs	DEC dp+X	TCALL 11	XMA {X}	XMA dp	DECW dp	DEC Y	TYA
110	BCS rel				LDA {X}	LDA !abs+Y	LDA [dp+X]	LDA [dp]+Y	LDY !abs	LDY dp+X	TCALL 13	LDA {X}+	LDX !abs	STYA dp	XAY	DAA
111	BEQ rel				STA {X}	STA !abs+Y	STA [dp+X]	STA [dp]+Y	STY !abs	STY dp+X	TCALL 15	STA {X}+	STX !abs	CBNE dp	XYX	NOP