



概述

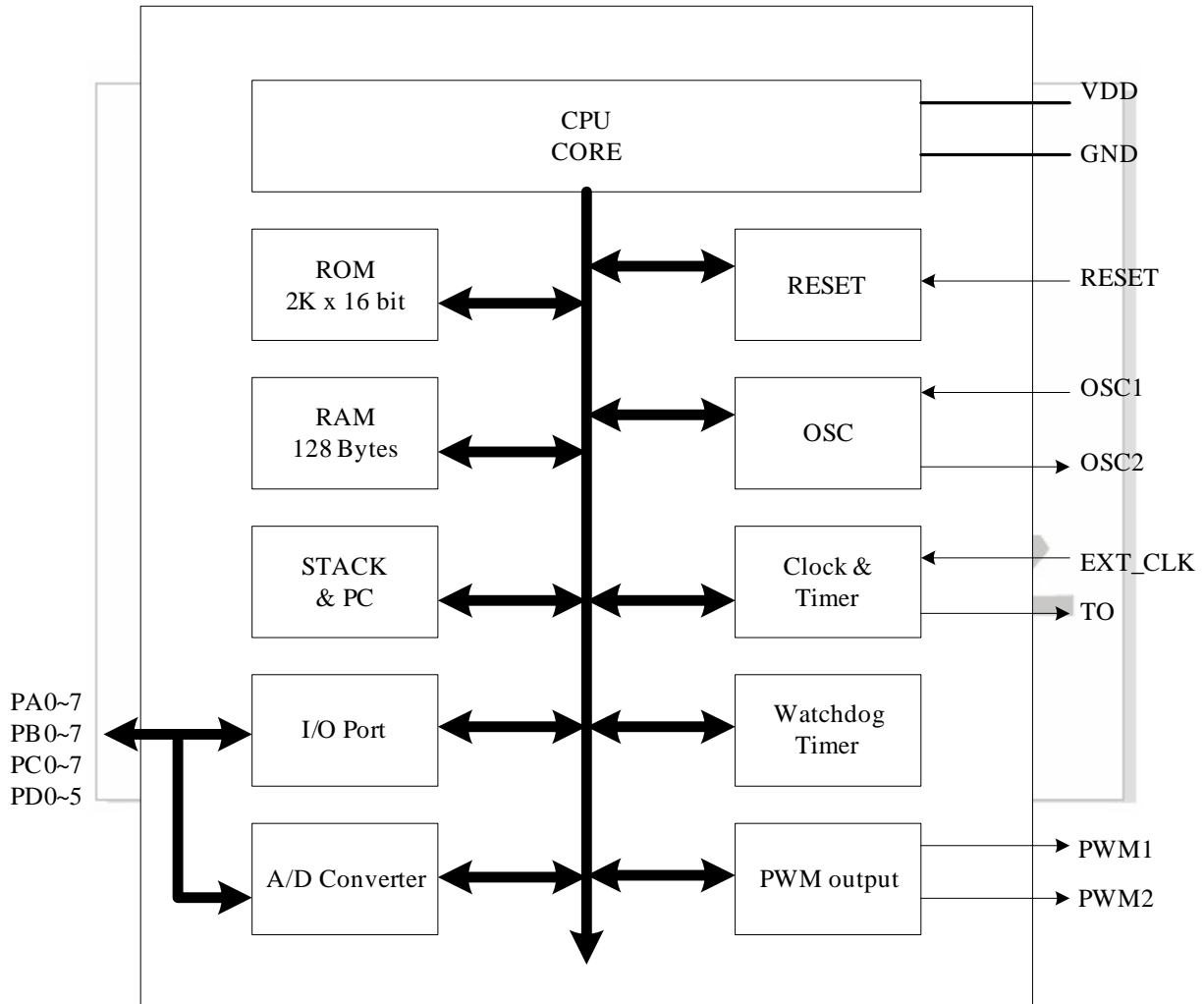
MK7A22P 是带 15 位 (11bit+4bit) A/D 转换器的 RISC 高性能 8 位微控制器。它内部包含 2K 字节的一次性可编程只读存储器、128 字节数据存储器、定时器/计数器、中断、LVR (低电压复位)、I/O 口、比较器和 PWM 输出。

1. 基本特性

- ROM: 2K×16 bits
- RAM: 51×8 bits (特殊寄存器) + 128×8 bits (一般寄存器)
- 堆栈: 8 级
- 一个指令周期由两个系统时钟组成
- 复位模式:
 - (a) 上电复位
 - (b) 低电压复位
 - (c) RESETB/PB2 (如果设置成复位脚位) 输入一个负脉冲
 - (d) 看门狗定时器计数溢出复位
- 双时钟模式
 - 外部 RC 或晶振振荡器
 - 内部 4MHz RC 振荡器
- 定时器/计数器: 5 个累加计数器
 - TM0: 16-bit 定时器 (up-counter)
 - TM1: 8-bit, PWM1 (period) & 定时器
 - TM2: 8-bit, PWM1 (duty) & 定时器
 - TM3: 8-bit, PWM2 (period) & 定时器
 - TM4: 8-bit, PWM2 (duty) & 定时器
 - TO: PWM2 时钟/2 输出
- 看门狗定时器: 芯片内 WDT 是基于一个内部 RC 振荡器 (仅 WDT 使用)。有 8 个周期可供选择。使用者可通过使用预分频器来延长 WDT 溢出周期。
- 中断结果:
 - (a) 外部中断 (PA7~PA0)
 - (b) 内部定时器/结果计数器中断 (TM0~TM4, PWM1, PWM2)
 - (c) ADC 结束转换中断
 - (d) 比较器比较结果激活中断
- I/O 口: 29 脚位
- 4 路比较器设置与 I/O 口分享脚位
- PWM: 两个通道带 8bit×8bit 精度
- ADC: 最多 15-bit 及 12 通道, 至少 12-bit 精度。它可通过寄存器设置在转换模式或比较模式下工作
- 唤醒模式: A 口 (PA7~PA0) 脚位变化唤醒 & WDT 溢出
- 450O: 450KHZ 时钟输出 (可调节)
- CLO: 系统时钟输出
- 不同封装类型:
 - MK7A22PD32C: 600mil 32 pin DIP
 - MK7A22PS32C: 32 pin SOP
 - MK7A22PD28C: 600mil 28 pin DIP
 - MK7A22PS28C: 28 pin SOP
 - MK7A22PD20C: 20 pin DIP
 - MK7A22PS20C: 20 pin SOP
 - MK7A22PS14C: 14 pin SOP

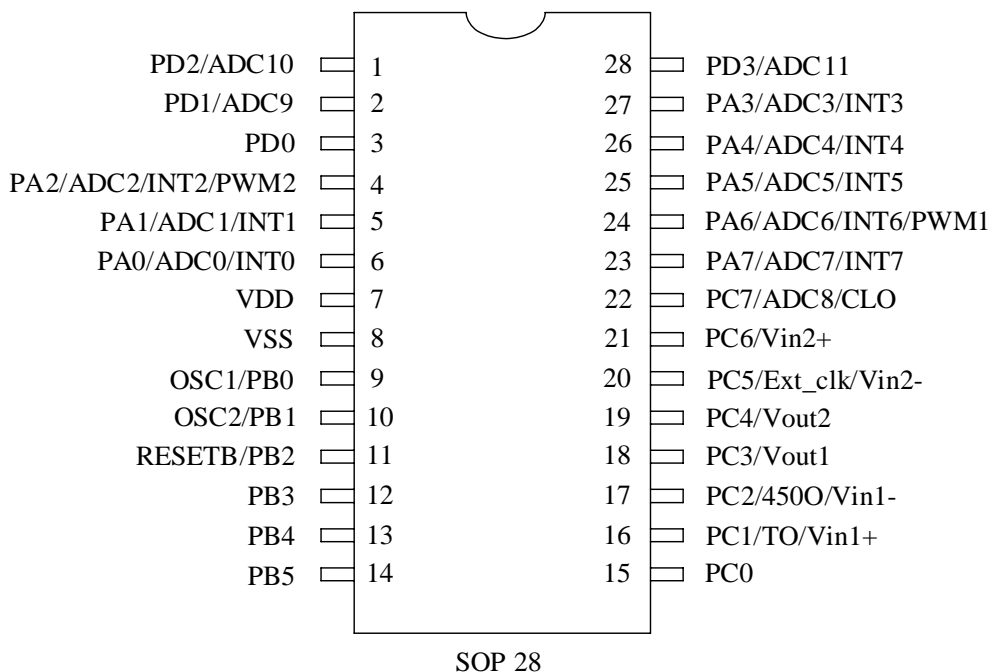
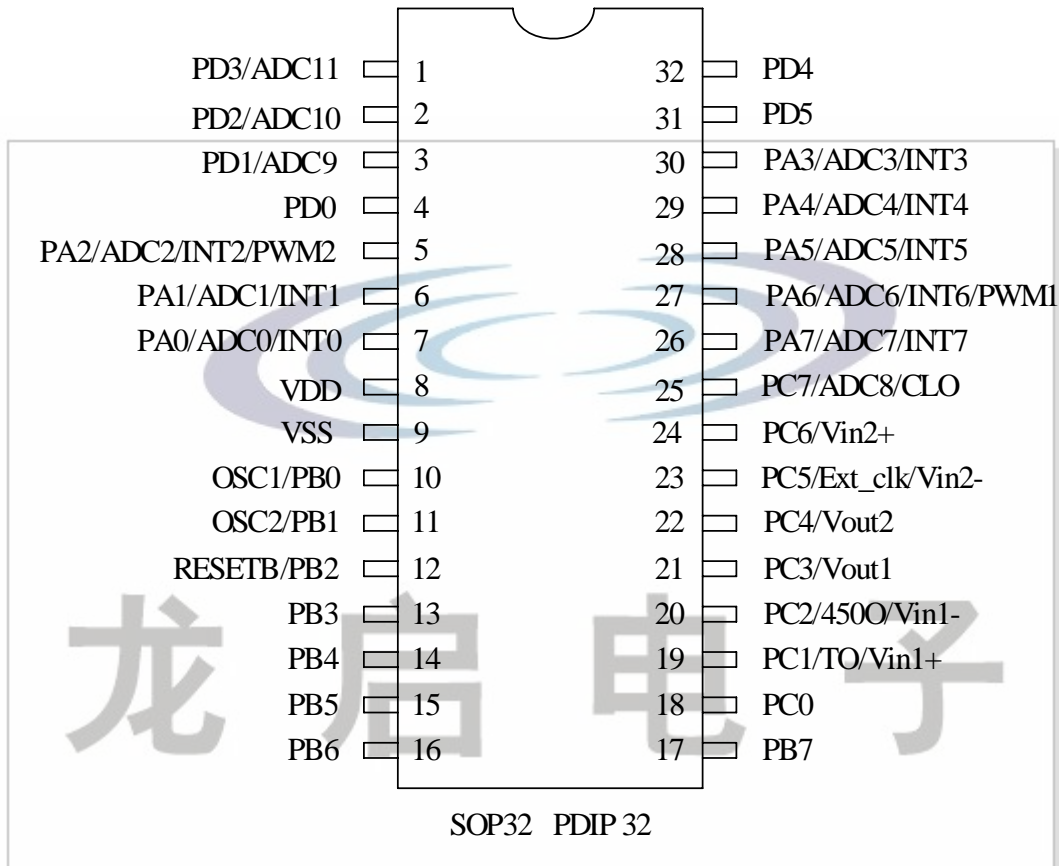


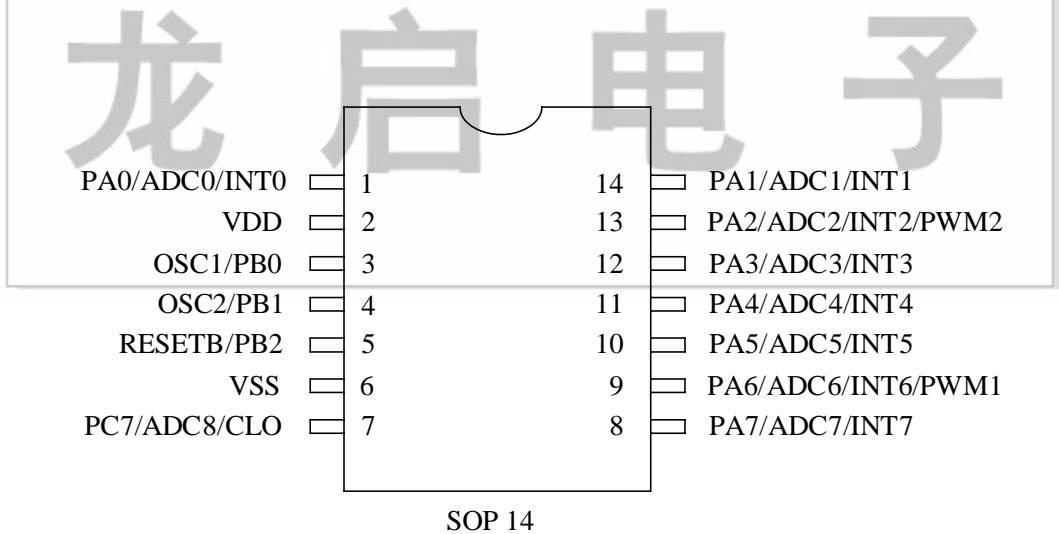
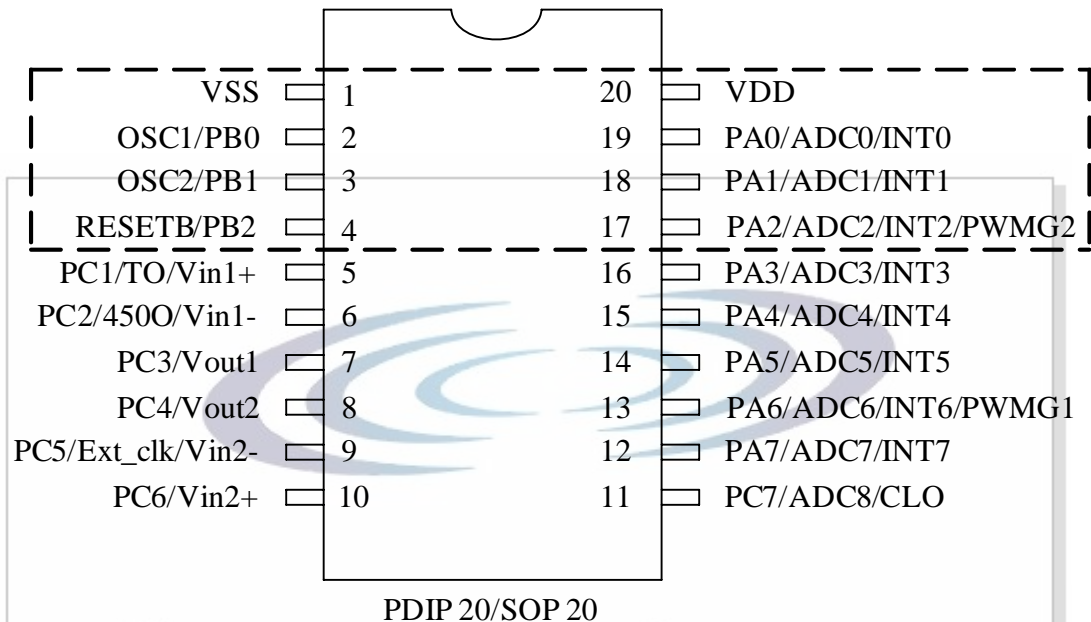
2. 图表





3. 脚位定义 & 管脚分配







4. 脚位说明

脚位名称	I/O	说明
PA0/ADC0/INT0 ~ PA1/ADC1/INT1 PA3/ADC3/INT3 ~ PA5/ADC5/INT5, PA7/ADC7/INT7	I/O	1. 双向I/O口（输入模式下会有上拉电阻） 2. 脚位改变时唤醒（选择） 3. 外部中断输入（选择） 4. A/D转换输入
PA2/ADC2/INT2/PWM2 PA6/ADC6/INT6/PWM1	I/O	1. 双向I/O口（输入模式下会有上拉电阻） 2. 脚位改变时唤醒（选择） 3. 外部中断输入（选择） 4. A/D转换输入 5. PWM1 & PWM2输出
OSC1/PB0	I, I/O	1. 双向I/O口（输入模式下会有上拉电阻） 2. 振荡器输入
OSC2/PB1	O, I/O	1. 双向I/O口（输入模式下会有上拉电阻） 2. 振荡器输出
RESETB/PB2	I	1. 输入口 2. 系统复位脚位
PB3~PB7	I/O	1. PB3~PB7是双向I/O口（输入模式下会有上拉电阻）
PC0	I/O	1. PC0, PC7是双向I/O口（输入模式下会有上拉电阻）
PC1/TO/Vin1+ PC2/500O/Vin1- PC3/Vout1	I/O	1. PC1~PC3是双向I/O口（输入模式下会有上拉电阻） 2. 脚位与比较器1 Vin1+, Vin1-, Vout1共享 3. 如果使用比较器功能，PC1~PC3将不能使用，Vin1+, Vin1-是模拟输入，Vout1是比较器输出，在这个模式下上拉电阻器将会自动禁止 4. TO（PWM2中断/2）与PC1共享 5. 500O（500KHz时钟输出）与PC2共享
PC4/Vout2 PC5/EXT_CLK/Vin2- PC6/Vin2+	I/O	1. PC4~PC6是双向I/O口（输入模式下会有上拉电阻） 2. 脚位与比较器2 Vin2+, Vin2-, Vout2共享 3. 如果使用比较器功能，PC4~PC6将禁止，Vin2+, Vin2-是模拟输入，Vout2是比较器输出，在这个模式下上拉电阻器将会自动禁止 4. EXT_CLK（外部时钟输入）与PC5共享
PC7/ADC8/CLO	I/O	1. 双向I/O口（输入模式下会有上拉电阻） 2. A/D转换器输入 3. CLO（系统时钟输出）与PC7共享



PD0	I/O	1. 双向I/O口（输入模式下会有上拉电阻）
PD1/ADC9~PD3/ADC11	I/O	1. 双向I/O口（输入模式下会有上拉电阻） 2. A/D转换器输入
PD4~PD5	I/O	1. 双向I/O口（输入模式下会有上拉电阻）
VDD	P	电源输入
VSS	P	接地输入

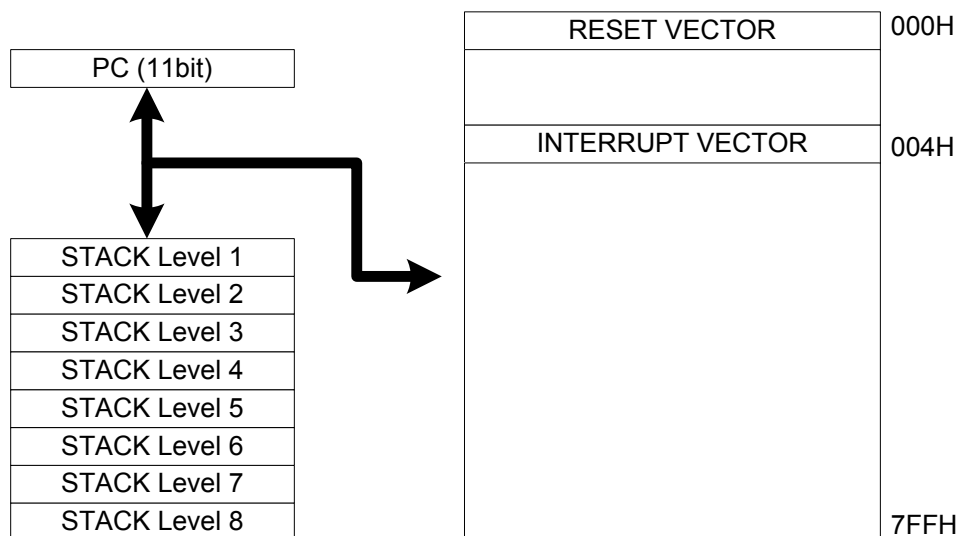
5. 存储器映象

MK7A22P 芯片带有两种存储器，分别是程序存储器（ROM）和数据存储器（RAM）。程序存储器用于存储程序、数据表及中断向量，它是连续的 2048×16bits，不需要转换到 bank。数据存储器是 179(51+128)×8bits，它包括特殊功能寄存器和一般的数据存储器。

5.1 程序存储器（ROM）

指令和数据表存储在程序存储器内。程序存储器只能有一个中断向量存在，那意味着所有发生的中断都将跳到相同的向量。程序会通过中断标记来判断是哪一种中断发生。程序计数器（PC）有 11bit，它能直接寻找所有 1024×16bits 位置地址。查询数据表可以置于程序存储器的任何地方。

RESET 向量位于 000H，中断向量位于 004H。映象图如下所示：

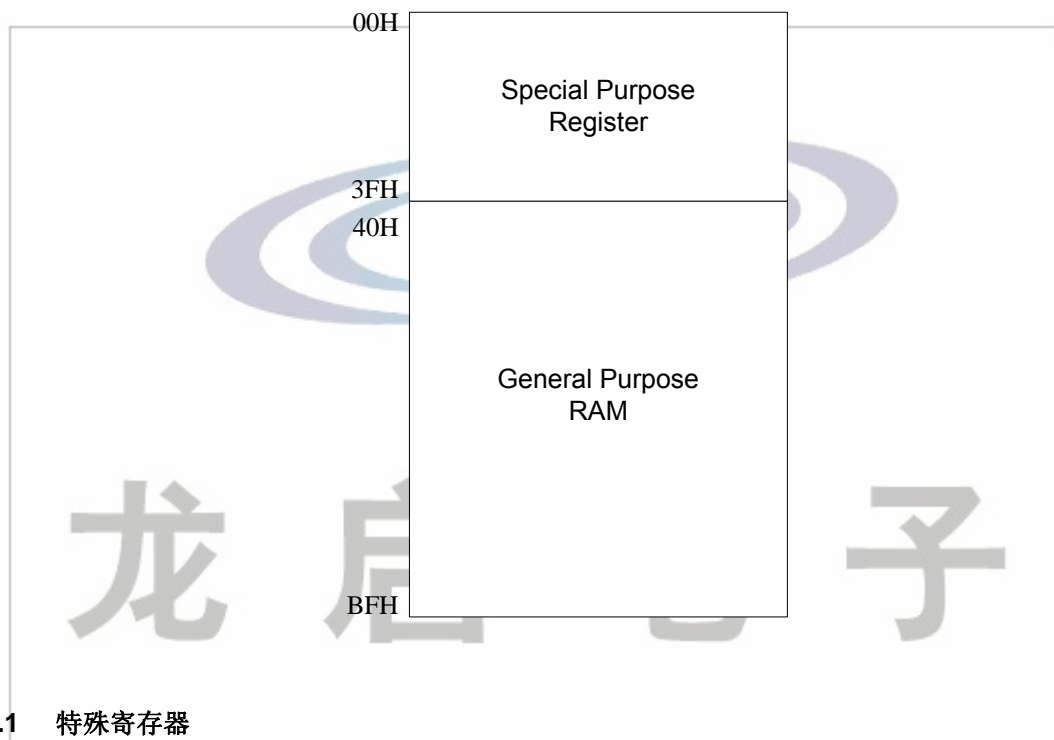




5.2 数据存储器 (RAM)

全部的数据存储器集都是 179×8bits，它们包含两种寄存器组。一种是 128×8bits 的一般存储器，另一种是 51×8bits 的特殊寄存器。特殊寄存器的每一字节都用来存储控制数据和操作数据。

数据存储器映象如下所示：



5.2.1 特殊寄存器

Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG_L		RST_DEF	LV1	LV0	WDTE	CPRT	INRC	FOSC1	FOSC0
CONFIG_H		ADJ6	ADJ5	ADJ4	ADJ3	ADJ2	ADJ1	ADJ0	RTCPN
INDF	\$00	A7	A6	A5	A4	A3	A2	A1	A0
PCL	\$01	A7	A6	A5	A4	A3	A2	A1	A0
PCH	\$02	--	--	--	--	--	A10	A9	A8
STATUS	\$03	--	--	--	\overline{TO}	\overline{PD}	Z	DC	C
FSR	\$04	D7	D6	D5	D4	D3	D2	D1	D0
I/O PAD & Control									
Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_DIR	\$05	IOA7	IOA6	IOA5	IOA4	IOA3	CA2	IOA1	IOA0
PA_DAT	\$06	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
PB_DIR	\$07	IOB7	IOB6	IOB5	IOB4	IOB3	--	IOB1	IOB0
PB_DAT	\$08	DB7	DB6	DB5	DB4	DB3	--	DB1	DB0
PC_DIR	\$09	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0



PC_DAT	\$0A	DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0
PD_DIR	\$0B	--	--	IOD5	IOD4	IOD3	IOD2	IOD1	IOD0
PD_DAT	\$0C	--	--	DD5	DD4	DD3	DD2	DD1	DD0
Timer 0: 16-bit Timer									
Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0_CTL	\$10	TM0_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0
TM0L_LA	\$11	D7	D6	D5	D4	D3	D2	D1	D0
TM0H_LA	\$12	D7	D6	D5	D4	D3	D2	D1	D0
TM0L_CNT	\$13	D7	D6	D5	D4	D3	D2	D1	D0
TM0H_CNT	\$14	D7	D6	D5	D4	D3	D2	D1	D0
Timer 1: 8-bit, PWM (period) & Timer									
Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1_CTL1	\$15	TM1_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0
TM1_CTL2	\$16	MOD	PWM1_OS	**	**	POS3	POS2	POS1	POS0
TM1_LA	\$17	D7	D6	D5	D4	D3	D2	D1	D0
TM1_CNT	\$18	D7	D6	D5	D4	D3	D2	D1	D0
Timer 2: 8-bit, PWM (duty) & Timer									
Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2_CTL	\$19	TM2_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0
TM2_LA	\$1A	D7	D6	D5	D4	D3	D2	D1	D0
TM2_CNT	\$1B	D7	D6	D5	D4	D3	D2	D1	D0
Timer 3: 8 bit, PWM (period) & Timer									
Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3_CTL1	\$1C	TM3_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0
TM3_CTL2	\$1D	MOD	PWM2_OS	**	**	POS3	POS2	POS1	POS0
TM3_LA	\$1E	D7	D6	D5	D4	D3	D2	D1	D0
TM3_CNT	\$1F	D7	D6	D5	D4	D3	D2	D1	D0
Timer 4: 8 bit, PWM (duty) & Timer									
Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM4_CTL	\$20	TM4_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0
TM4_LA	\$21	D7	D6	D5	D4	D3	D2	D1	D0
TM4_CNT	\$22	D7	D6	D5	D4	D3	D2	D1	D0
IRQ									



Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRQM	\$25	INTM	ADCM	PAM	TM4M	TM3M/PWM2M	TM2M	TM1M/PWM1M	TM0M
IRQF	\$26	--	ADCF	PAF	TM4F	TM3F/PWM2F	TM2F	TM1F/PWM1F	TM0F
ADC control									
Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL1	\$29	EN	--	MODE	--	CHSEL3	CHSEL2	CHSEL1	CHSEL0
AD_CTL2	\$2A	RSUT	--	--	--	--	--	CKSEL1	CKSEL0
AD_CTL3	\$2B	--	--	--	--	PISEL3	PISEL2	PISEL1	PISEL0
AD_DATL	\$2C	D7	D6	D5	D4	D3	D2	D1	D0
AD_DATH	\$2D	--	D14	D13	D12	D11	D10	D9	D8
Other									
Name	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMP_CTL	\$2F	--	--	--	--	**	**	CMP2_E	CMP1_E
FREQ_CTL	\$30	S500O	500_PR1	500_PR0	--	--	500O_E	CLO_E	TO_E
PA_PLU	\$31	UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0
PB_PLU	\$32	UB7	UB6	UB5	UB4	UB3	--	UB1	UB0
PC_PLU	\$33	UC7	UC6	UC5	UC4	UC3	UC2	UC1	UC0
PD_PLU	\$34	--	--	UD5	UD4	UD3	UD2	UD1	UD0
IO_CTL	\$35	--	--	--	--	PORTD	PORTC	PORTB	PORTA
PA_INTE	\$3A	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
WDT_CTL	\$3D	WDTEN	--	--	--	--	PRE2	PRE1	PRE0
TAB_BNK	\$3E	--	--	--	--	--	BNK2	BNK1	BNK0
SYS_CTL	\$3F	CLKS	--	--	--	--	--	STPRC	STPOSC

< 注 > “—”: 表示未使用

“**”: 表示 0 (初始值=0)

5.2.2 结构寄存器

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CONFIG_L	RST_DEF	LV1	LV0	WDTE	CPRT	INRC	FOSC1	FOSC0
-	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
CONFIG_H	ADJ6	ADJ5	ADJ4	ADJ3	ADJ2	ADJ1	ADJ0	EXT_CLK

- Bit15~9 (ADJ6~0): 用于校准内部RC振荡器
- Bit8 (EXT_CLK): EXT_CLK 输入
 - 1: EXT_CLK (PC5) 脚位是定时器源输入 & PC5输入
 - 0: EXT_CLK (PC5) 脚位是普通I/O脚位



- Bit7 (RST_DEF): RESETB脚位定义
0: RESETB是普通输入脚位
1: RESETB是系统复位脚位
- Bit6~5 (LV1~0): 设置低电压复位 (LVR) 的复位电压级别

Bit6	Bit5	检测电压
LV1	LV0	
0	0	4V
0	1	未执行
1	0	2.3V
1	1	未使用

< 注 > 掉电电压会受到过程及温度的影响，因此列表中的电压会存在一些误差。

- Bit4 (WDTE): 看门狗定时器使能/禁止
0: WDT 禁止
1: WDT 使能
- Bit3 (CPRT): ROM 密码保护位
0: 开
1: 关
- Bit2~0 (INRC, FOSC1~0): OSC类型及系统时钟选择

Bit2	Bit1	Bit0	OSC 类型	共振频率
INRC	FOSC1	FOSC0		
0	0	0	LS (低速)	系统时钟 =32~200KHz
0	0	1	NS (常速)	系统时钟 =200K~10MHz
0	1	0	HS (高速)	系统时钟 =10~20MHz
0	1	1	双重 RC 时钟: 外部 RC & 内部 RC	系统时钟 =32K ~ 10MHz
1	0	0	LS & 内部 RC	1. 双重时钟模式 LS & 4MHz 2. 初始系统时钟 =4MHz
1	0	1	NS & 内部 RC	1. 双重时钟模式 NS & 4MHz 2. 初始系统时钟 =4MHz
1	1	0	HS & 内部 RC	1. 双重时钟模式 HS & 4MHz 2. 初始系统时钟 =4MHz
1	1	1	内部 RC	1. 系统时钟 =4MHz 2. OSC1 & OSC2 作为I/O口工作



6. 功能描述

此芯片提供许多功能，包括定时器，WDT，PWM，ADC，中断，数据表位置，复位，程序计数器及 STATUS 寄存器。我们将会在下面详细描述。

6.1 I/O口

该芯片有 4 个 I/O 口用于数据输入及输出，使用者可用每个 I/O 口的不同功能来编程。A 口可通过选择寄存器实现外部中断，ADC 模拟输入或 PWM 输出。B 口具有系统复位或外部 RC 振荡器输入功能。C 口带有 ADC 模拟输入，500KHZ 输出，系统时钟输出或 PWM 输出功能。D 口可作为 ADC 模拟输入。

6.1.1 Port A

A 口有 3 个寄存器可设置 8 个 I/O 口，分别是 PA_DIR，PA_DAT，PA_PLU。A 口的每一脚位都可做为外部中断输入或一般 I/O 口。欲了解如何将这脚位设置为外部中断，请参考章节 6.7。PA2 (PWM2)、PA6 (PWM1) 可设置为 PWM 输出。

A. PA_DIR(\$05H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_DIR	IOA7	IOA6	IOA5	IOA4	IOA3	IOA2	IOA1	IOA0

- Bit7~0 (IOA7~0): 定义每一个脚位是输入口还是输出口

0: 输出

1: 输入

B. PA_DAT(\$06H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_DAT	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0

- Bit7~0 (DA7~0): 数据缓冲器

C. PA_PLU(\$31H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PA_PLU	UA7	UA6	UA5	UA4	UA3	UA2	UA1	UA0

- Bit7~0 (UA7~0): 上拉使能/禁止

0: 上拉禁止

1: 上拉使能

6.1.2 B 口

B 口有 3 个寄存器可设置 7 个 I/O 口，分别是 PB_DIR，PB_DAT，PB_PLU。B 口的每一脚位都可设置为一般 I/O 口。当 B 口被设置为输入模式时，使用者可设置上拉。CONFIG_L 寄存器可设置 PB2 (RESETB) 为系统复位 (低电压复位) 或纯输入脚位。通常情况下，PB0 (OSC1) 和 PB1 (OSC2) 是外部振荡器脚位，只有当选择内部 RC 模式，PB0 和 PB1 才是一般 I/O 口。

A. PB_DIR(\$07H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB_DIR	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0



PB_DIR	IOB7	IOB6	IOB5	IOB4	IOB3	--	IOB1	IOB0
--------	------	------	------	------	------	----	------	------

- Bit7~0 (IOB7~0): 定义每一个脚位是输入口还是输出口

0: 输出

1: 输入

< 注 > Bit2: 仅输入及在 CONFIG_L bit7 被设置为 RESET 或输入口。

B. PB_DAT(\$08H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB_DAT	DB7	DB6	DB5	DB4	DB3	--	DB1	DB0

- Bit7~0 (DB7~0): 数据缓冲器

C. PB_PLU(\$32H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PB_PLU	UB7	UB6	UB5	UB4	UB3	--	UB1	UB0

- Bit7~0 (UB7~0): 上拉使能/禁止

0: 上拉禁止

1: 上拉使能

6.1.3 C 口

C 口有 3 个寄存器可设置 8 个 I/O 口，分别是 PC_DIR, PC_DAT, PC_PLU。PC5 (Ext_clk) 可被设置为外部时钟输入。当 PC7~PC0 被设置为输入模式时，使用者可设置上拉。PC5 有多种功能，使用者应首先在 CONFIG_H bit8 定义它。

A. PC_DIR(\$09H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC_DIR	IOC7	IOC6	IOC5	IOC4	IOC3	IOC2	IOC1	IOC0

- Bit7~0: 定义每一个脚位是输入口还是输出口

0: 输出

1: 输入

B. PC_DAT(\$0AH):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC_DAT	DC7	DC6	DC5	DC4	DC3	DC2	DC1	DC0

- Bit7~0 (DC7~0): 数据缓冲器

C. PC_PLU(\$33H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PC_PLU	UC7	UC6	UC5	UC4	UC3	UC2	UC1	UC0

- Bit7~0 (UC7~0): 上拉使能/禁止

0: 上拉禁止

1: 上拉使能



6.1.4 D 口

D口有3个寄存器可设置6个I/O口，分别是PD_DIR, PD_DAT, PD_PLU。A口的每一脚位都可做为一般I/O口或ADC模拟输入。

A. PD_DIR(\$0BH):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PD_DIR	--	--	IOD5	IOD4	IOD3	IOD2	IOD1	IOD0

- Bit6~0 (IOD6~0): 定义每一个脚位是输入口还是输出口

0: 输出

1: 输入

B. PD_DAT(\$0CH):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PD_DAT	--	--	DD5	DD4	DD3	DD2	DD1	DD0

- Bit7~0 (DD7~0): 数据缓冲器

C. PD_PLU(\$34H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PD_PLU	--	--	UD5	UD4	UD3	UD2	UD1	UD0

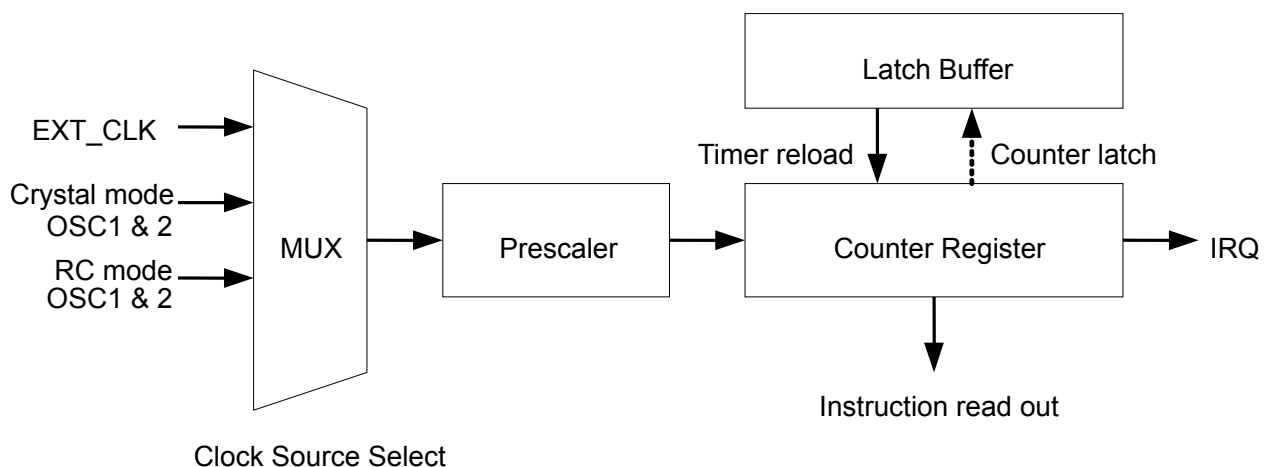
- Bit7~0 (UD7~0): 上拉使能/禁止

0: 上拉禁止

1: 上拉使能

6.2 定时器/结果计数器 (TM1, TM2, TM3)

MK7A22P提供5个累加定时器/计数器和1个看门狗定时器。通过设置每一个定时器控制寄存器，计数器的时钟源可以是INRC, CRYSTAL, 也可以是外部时钟。TM0是16位定时器，其它的是8位定时器。所有定时器带自动重复下载功能，TM1/TM2和TM3/TM4组合后可实现PWM功能。寄存器详细设置及图表如下所示：



这些定时器带多种功能，可作为定时器，PWM输出波形使用。



6.2.1 TM0

TM1 是 16 位定时器/计数器，有 5 个寄存器设置他的属性。

A. TM0_CTL (\$10H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0_CTL	TM0_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0

- Bit7 (TM0_EN): Timer0使能/禁止
0: TM0禁止
1: TM0使能
- Bit6 (WR_CNT): 锁存器数据写到计数器寄存器使能/禁止
0: 锁存器数据写到计数器寄存器禁止
1: 锁存器数据写到计数器寄存器使能

< 注 > 此位只有在新定时器/计数器数据的初始状态下被设置，才能让锁存器数据写到计数器寄存器。当定时器溢出，锁存器数据会自动重复下载到计数器寄存器。使用者不需要再设置一次。

- Bit5~4 (SUR1~0): TM0时钟源选择位

Bit5	Bit4	TM0时钟源
SUR1	SUR0	
0	0	EXT_CLK (PC5)
0	1	晶振模式 OSC1 或 EXT_RC(双重 RC 时钟模式)
1	0	RC 4MHZ 模式
1	1	未使用

< 注 > SUR1~0 定义 TM0 时钟源。

- Bit3 (EDGE): 当定时器作为结果计数器使用时TM0时钟源边沿控制位
0: 外部时钟从低电平到高电平时，定时器加1
1: 外部时钟从高电平到低电平时，定时器加1
- Bit2~0 (PRE2~0): 设置TM0预分频率

Bit2	Bit1	Bit0	TMR0 预分频率
PRE2	PRE1	PRE0	
0	0	0	1:1
0	0	1	1:2
0	1	0	1:4
0	1	1	1:8
1	0	0	1:16
1	0	1	1:32
1	1	0	1:64
1	1	1	1:128



B. TM0L_LA/TM0H_LA and TM0L_CNT/TM0H_CNT Register (\$11H, \$12H, \$13H, \$14H)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM0L_LA	D7	D6	D5	D4	D3	D2	D1	D0
TM0H_LA	D7	D6	D5	D4	D3	D2	D1	D0
TM0L_CNT	D7	D6	D5	D4	D3	D2	D1	D0
TM0H_CNT	D7	D6	D5	D4	D3	D2	D1	D0

< 注 1 > 请不要写 FFh 数据到 TM0L_LA & TM0H_LA 寄存器

< 注 2 > TM0L_CNT & TM0H_CNT 是只读寄存器

6.2.2 TM1 (或 PWM1 period)

TM1 是 8 位定时器/计数器，有 4 个寄存器设置他的属性。TM1 可当作 PWM1 period 使用，与 TM2 一起实现 PWM1 波形。

A.TM1_CTL1 (\$15H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1_CTL1	TM1_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0

- Bit7 (TM1_EN): TM1 (PWM1) 使能/禁止
0: TM1禁止
1: TM1使能
- Bit6 (WR_CNT): 锁存器数据写到计数器寄存器使能/禁止
0: 锁存器数据写到计数器寄存器禁止
1: 锁存器数据写到计数器寄存器使能

< 注 > 此位只有在新定时器/计数器数据的初始状态下被设置，才能让锁存器数据写到计数器寄存器。当定时器溢出，锁存器数据会自动重复下载到计数器寄存器。使用者不需要再设置一次。

- Bit5~4 (SUR1~0): TM1时钟源选择位

Bit5	Bit4	TM1/PWM1 时钟源
SUR1	SUR0	
0	0	EXT_CLK (PC5)
0	1	晶振模式 OSC1 或 EXT_RC(双重 RC 时钟模式)
1	0	RC 4MHZ 模式
1	1	未使用

- Bit3 (EDGE): 当定时器作为结果计数器使用时TM1时钟源边沿控制位
0: 外部时钟从低电平到高电平时，定时器加1
1: 外部时钟从高电平到低电平时，定时器加1
- Bit2~0 (PRE2~0): 设置TM1预分频率



Bit2	Bit1	Bit0	TM1 预分频率
PRE2	PRE1	PRE0	
0	0	0	1:1
0	0	1	1:2
0	1	0	1:4
0	1	1	1:8
1	0	0	1:16
1	0	1	1:32
1	1	0	1:64
1	1	1	1:128

B.TM1_CTL2 (\$16H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1_CTL2	MOD	PWM1_OS	**	**	POS3	POS2	POS1	POS0

- Bit7 (MOD): TM1可在不同模式下工作。定时器, PWM模式设置如下。
0: TM1在定时器模式下工作
1: TM1在PWM模式下工作
- Bit6 (PWM1_OS): 设置PWM1输出比例初始状态
0: 设置初始输出状态是高电平, 当TM2定时器溢出时将变换为低电平
1: 设置初始输出状态是低电平, 当TM2定时器溢出时将变换为高电平
- Bit3~0 (POS3~0): PWM1输出脉冲设置 (只有在TM1工作于PWM模式下时才被激活)

Bit3	Bit2	Bit1	Bit0	PWM1脉冲率
POS3	POS2	POS1	POS0	
0	0	0	0	1:1
0	0	0	1	1:2
0	0	1	0	1:3
.
.
1	1	1	0	1:15
1	1	1	1	1:16

< 注 > 在 PWM 模式下, 1:N 意味着经过 N 个 PWM1 脉冲后中断将发生

C. TM1_LA & TM1_CNT (\$17H, 18H)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM1_LA	D7	D6	D5	D4	D3	D2	D1	D0
TM1_CNT	D7	D6	D5	D4	D3	D2	D1	D0

< 注 > TM1_CNT 寄存器是只读寄存器



6.2.3 TM2 (或 PWM1 duty)

TM2 是 8 位定时器/计数器，有 3 个寄存器设置它的属性。TM2 可当作 PWM1 duty 使用，与 TM1 共同控制实现 PWM1 波形。

A.TM2_CTL (\$19H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2_CTL	TM2_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0

- Bit7 (TM2_EN): TM2使能位

0: TM2禁止

1: TM2使能

< 注 > 当 TM1_CTL2 被设置为 PWM 模式，此位将被自动抑制。TM2 就会成为 PWM1 波形的占空比计数器。

- Bit6 (WR_CNT): 锁存器数据写到计数器寄存器使能/禁止

0: 锁存器数据写到计数器寄存器禁止

1: 锁存器数据写到计数器寄存器使能

< 注 > 此位只有在新定时器计数器数据的初始状态下被设置，才能让锁存器数据写到计数器寄存器。当定时器溢出，锁存器数据会自动重复下载到计数器寄存器。使用者不需要再设置一次。

- Bit5~4 (SUR1~0): TM2时钟源选择位

Bit5	Bit4	TM2时钟源
SUR1	SUR0	
0	0	EXT_CLK (PC5)
0	1	晶振模式 OSC1 或 EXT_RC(双重 RC 模式)
1	0	RC 4MHZ 模式
1	1	未使用

- Bit3 (EDGE): 当定时器作为结果计数器使用时TM2时钟源边沿控制位

0: 外部时钟从低电平到高电平时，定时器加1

1: 外部时钟从高电平到低电平时，定时器加1

- Bit2~0 (PRE2~0): 预分频器分配脚位

Bit2	Bit1	Bit0	TM2预分频率
PRE2	PRE1	PRE0	
0	0	0	1:1
0	0	1	1:2
0	1	0	1:4
0	1	1	1:8
1	0	0	1:16
1	0	1	1:32
1	1	0	1:64
1	1	1	1:128



B. TM2_LA & TM2_CNT (\$1AH,1BH)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM2_LA	D7	D6	D5	D4	D3	D2	D1	D0
TM2_CNT	D7	D6	D5	D4	D3	D2	D1	D0

< 注 > TM2_CNT寄存器为只读寄存器

6.2.4 TM3 (或 PWM2 period)

TM3 是 8 位定时器/计数器，有 4 个寄存器设置它的属性。TM3 可当作 PWM2 period 使用，与 TM4 共同控制实现 PWM2 波形。

A.TM3_CTL1 (\$1CH):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3_CTL1	TM3_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0

- Bit7 (TM3_EN): TM3 (PWM2) 使能/禁止
0: TM3禁止
1: TM3使能
- Bit6 (WR_CNT): 锁存器数据写到计数器寄存器使能/禁止
0: 锁存器数据写到计数器寄存器禁止
1: 锁存器数据写到计数器寄存器使能

< 注 > 此位只有在新定时器计数器数据的初始状态下被设置，才能让锁存器数据写到计数器寄存器。当定时器溢出，锁存器数据会自动重复下载到计数器寄存器。使用者不需要再设置一次。

- Bit5~4 (SUR1~0): TM2时钟源选择位

Bit5	Bit4	TM3/PWM2时钟源
SUR1	SUR0	
0	0	EXT_CLK (PC5)
0	1	晶振模式 OSC1 或 EXT_RC(双重 RC 模式)
1	0	RC 4MHZ 模式
1	1	未使用

- Bit3 (EDGE): 当定时器作为结果计数器使用时TM3时钟源边沿控制位
0: 外部时钟从低电平到高电平时，定时器加1
1: 外部时钟从高电平到低电平时，定时器加1
- Bit2~0 (PRE2~0): 设置TM3预分频率

Bit2	Bit1	Bit0	TM3预分频率
PRE2	PRE1	PRE0	
0	0	0	1:1
0	0	1	1:2



0	1	0	1:4
0	1	1	1:8
1	0	0	1:16
1	0	1	1:32
1	1	0	1:64
1	1	1	1:128

B.TM3_CTL2 (\$1DH):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3_CTL2	MOD	PWM2_OS	**	**	POS3	POS2	POS1	POS0

- Bit7 (MOD): TM3可在不同模式下工作。定时器, PWM模式设置如下。
0: TM3在定时器模式下工作
1: TM3在PWM模式下工作

- Bit6 (PWM2_OS): 设置PWM2输出比例初始状态
0: 设置初始输出状态是高电平, 当TM4定时器溢出时将变换为低电平
1: 设置初始输出状态是低电平, 当TM4定时器溢出时将变换为高电平

< 注 > “**” bits 是为系统保留的, 不能被设置为 “1”。

- Bit3~0 (POS3~0): PWM2输出脉冲设置 (只有在TM3工作于PWM模式下时才被激活)

Bit3	Bit2	Bit1	Bit0	PWM2脉冲率
POS3	POS2	POS1	POS0	
0	0	0	0	1:1
0	0	0	1	1:2
0	0	1	0	1:3
.
.
1	1	1	0	1:15
1	1	1	1	1:16

< 注 1 > 在 PWM 模式下, 1:N 意味着经过 N 个 PWM2 脉冲后中断将发生

C. TM3_LA & TM3_CNT (\$1EH, 1FH)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM3_LA	D7	D6	D5	D4	D3	D2	D1	D0
TM3_CNT	D7	D6	D5	D4	D3	D2	D1	D0

< 注 > TM3_CNT 寄存器是只读寄存器



6.2.5 TM4 (或 PWM2 duty)

TM4 是 8 位定时器/计数器，有 3 个寄存器设置它的属性。TM4 可当作 PWM2 duty 使用，与 TM3 共同控制实现 PWM2 波形。

A.TM4_CTL (\$20H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM4_CTL	TM4_EN	WR_CNT	SUR1	SUR0	EDGE	PRE2	PRE1	PRE0

- Bit7 (TM4_EN): TM4使能位

0: TM4禁止

1: TM4使能

< 注 > 当 TM3_CTL2 被设置为 PWM 模式，此位将被自动抑制。TM4 就会成为 PWM2 波形的占空比计数器。

- Bit6 (WR_CNT): 锁存器数据写到计数器寄存器使能/禁止

0: 锁存器数据写到计数器寄存器禁止

1: 锁存器数据写到计数器寄存器使能

< 注 > 此位只有在新定时器计数器数据的初始状态下被设置，才能让锁存器数据写到计数器寄存器。当定时器溢出，锁存器数据会自动重复下载到计数器寄存器。使用者不需要再设置一次。

- Bit5~4 (SUR1~0): TM4时钟源选择位

Bit5	Bit4	TM4时钟源
SUR1	SUR0	
0	0	EXT_CLK (PC5)
0	1	晶振模式 OSC1 或 EXT_RC(双重 RC 模式)
1	0	RC 4MHZ 模式
1	1	未使用

- Bit3 (EDGE): 当定时器作为结果计数器使用时TM4时钟源边沿控制位

0: 外部时钟从低电平到高电平时，定时器加1

1: 外部时钟从高电平到低电平时，定时器加1

- Bit2~0 (PRE2~0): 预分频器分配脚位

Bit2	Bit1	Bit0	TM4预分频率
PRE2	PRE1	PRE0	
0	0	0	1:1
0	0	1	1:2
0	1	0	1:4
0	1	1	1:8
1	0	0	1:16
1	0	1	1:32
1	1	0	1:64
1	1	1	1:128



B. TM4_LA & TM4_CNT (\$21H,\$22H)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TM4_LA	D7	D6	D5	D4	D3	D2	D1	D0
TM4_CNT	D7	D6	D5	D4	D3	D2	D1	D0

< 注 > TM4_CNT寄存器为只读寄存器

C. CMP_CTL (\$2FH):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMP_CTL	--	--	--	--	**	**	CMP2_E	CMP1_E

- Bit1 (CMP2_E): 比较器2使能/禁止
0: CMP2禁止
1: CMP2使能
- Bit0 (CMP1_E): 比较器1使能/禁止
0: CMP1禁止
1: CMP1使能

< 注 > “**” bits 是为系统保留的，不能被设置为“1”。

龙启电子



6.3 PWM (脉冲宽度调制)

PWM 波形由 TM2 (period) 和 TM3 (duty) 组合控制。通过设置寄存器，这两个定时器可用作一般定时器或 PWM 波形计数器。设置步骤如下例所示。

6.3.1 PWM

如果定时器全为 PWM 使用，则他们都是 8 位可编程占空比和周期。当系统频率为 4MHz 时，每一个都能发生脉冲，脉冲宽度从 0.5us 到 8,192ms。要设置定时器在 PWM 模式下工作，寄存器 TM1_CTL2(\$16)或 TM3_CTL2(\$1D)需设置为 1XX0XXXX (X 指不考虑)。请参考如下编程示例及定时图表：

<Example> 定时器作为PWM工作:

设置PWM period:

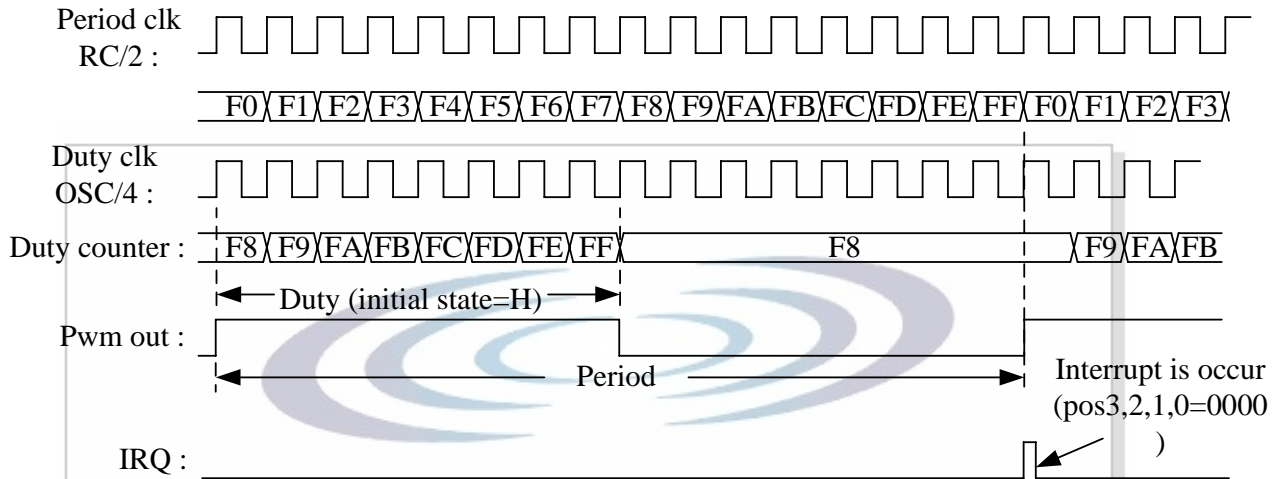
```
MOVLA    b01100001    ; 设置clk_source=RC, prescale=2
MOVAM    TM1_CTL1
MOVLA    b10000000
MOVAM    TM1_CTL2
MOVLA    F0H           ; 设置period计数器数值(累计计数到FFH)
MOVAM    TM1_LA
```

Set PWM duty:

```
MOVLA    b01010010    ; 设置 clk_source=OSC, prescale=4
MOVAM    TM2_CTL
MOVLA    F8H           ; 设置duty计数器数值(累计计数到FFH)
MOVAM    TM2_LA
BS       TM1_CTL1,b7   ; 开始PWM输出
MOVLA    82H           ; 使能INTM & PWM1中断
MOVAM    IRQM
CLR      IRQF          ; 清除中断标记
```




PWM Waveform



C.FREQ_CTL (\$30H):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FREQ_CTL	S500O	500_PR1	500_PR0	--	--	500O_E	CLO_E	TO_E

- Bit7 (S500O): 为500KHz输出选择时钟源
0: 设置500KHz时钟源=INRC (Init)
1: 设置500KHz时钟源=OSC
- Bit6, 5 (S500_PR1, S500_PR0): 预分频分配位

Bit6	Bit5	500Khz预分频率
S500_PR1	S500_PR0	
0	0	1:2
0	1	1:4
1	0	1:8
1	1	1:16

- Bit2 (500O_E): 500khz输出使能 (脚位与PC2共享)
0: 设置此脚位为一般I/O脚位
1: 设置些脚位为500KHZ输出
- Bit1 (CLO_E): 系统时钟输出使能 (脚位与PC7共享)
0: 设置此脚位为一般I/O脚位
1: 设置此脚位为系统时钟输出
- Bit0 (TO_E): 定时器输出 (TO) 使能/禁止 (脚位与PC1共享)
0: 设置此脚位为PC1一般I/O脚位
1: 设置此脚位为TO (定时器输出脚位, 频率是TM3(PWM2)计数器频率/2)

< 注 > TO信号输出前, PC1必须设置为输出口。



6.4 WDT (看门狗定时器)

WDT是防止软体故障及跳过含有不可预知结果的未知页面的定时器。WDT时钟源是一个独立的内部RC振荡器。此定时器会受温度、电压及不同产品批号的影响。

A.WDT_CTL (\$3DH):

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDT_CTL	WDTEN	--	--	--	--	PRE 2	PRE 1	PRE 0

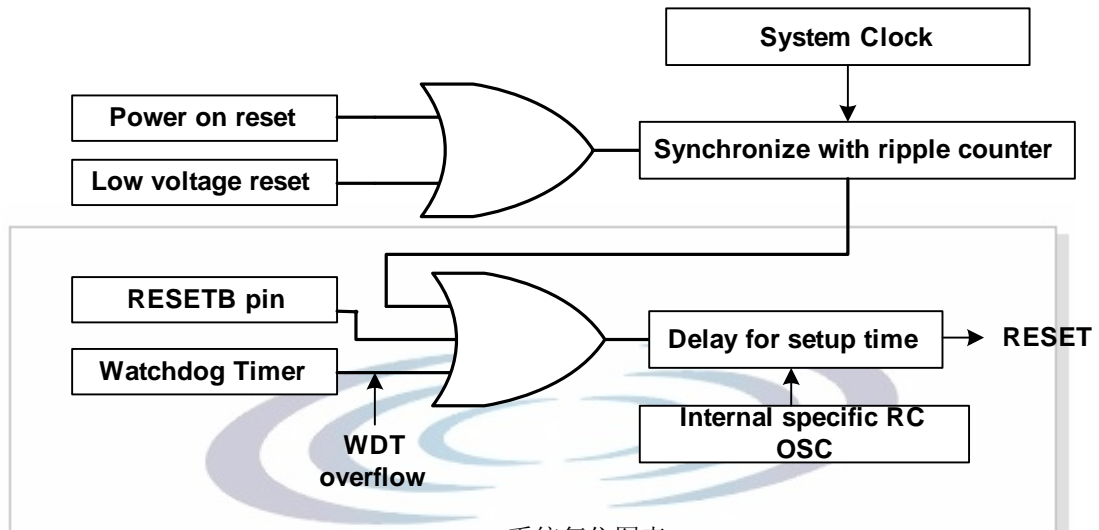
- Bit7 (WDTEN): 看门狗定时器使能位
0: WDT禁止
1: WDT使能
- Bit2~0 (PRE2~0): 设置预分频率。由于是RC OSC, 所有的数据不是精确数据

Bit2	Bit1	Bit0	WDT预分频率
PRE2	PRE1	PRE1	
0	0	0	20mS
0	0	1	40mS
0	1	0	80mS
0	1	1	160mS
1	0	0	320mS
1	0	1	640mS
1	1	0	1.28S
1	1	1	2.56S

6.5 复位

以下列出了4种会引起复位的情况。掉电将会引起MK7A21P复位, 检测电压依照CONFIG寄存器的bit6~bit5。这样能在供电不足的环境下保护芯片, 最后两种情况我们称之为热复位。不同的复位都会影响寄存器和数据存储器。 \overline{TO} 和 \overline{PD} 位用来决定复位的类型。

- (1) 上电复位
- (2) 低电压复位 (LVR)
- (3) RESETB脚位复位 (输入一个负脉冲)
- (4) WDT定时器溢出复位



系统复位图表

< 注 > 看门狗设置时间为大约20ms，由于电源电压、进程及温度差异，在时间设置上会有一些偏差。调整时间通过WDT_CTRL（\$3DH）设置。

不同复位条件下的默认值

Address	Name	Cold Reset	Warm Reset
N/A	Accumulator	xxxx xxxx	pppp pppp
00H	INDF	0000 0000	0000 0000
01H	PCL	0000 0000	0000 0000
02H	PCH	---- -000	---- -000
03H	STATUS	0001 1xxx	0001 1xxx
04H	FSR	Xxxx xxxx	Pppp pppp
05H	PA_DIR	1111 1111	1111 1111
06H	PA_DAT	Xxxx xxxx	Pppp pppp
07H	PB_DIR	1111 1x11	1111 1x11
08H	PB_DAT	Xxxx xxxx	pppp pxpp
09H	PC_DIR	1111 1111	1111 1111
0AH	PC_DAT	Xxxx xxxx	pppp ppxp
0BH	PD_DIR	XX11 1111	XX11 1111
0CH	PD_DAT	Xxxx xxxx	xxpp pppp
10H	TM0_CTL	0000 0000	0000 0000
11H	TM0L_LA (Don't write FF)	0000 0000	0000 0000
12H	TM0H_LA (Don't write FF)	0000 0000	0000 0000
13H	TM0L_DAT (Read only)	1111 1111	1111 1111
14H	TM0H_DAT (Read only)	1111 1111	1111 1111



15H	TM1_CTL1	0000 0000	0000 0000
16H	TM1_CTL2	0000 0000	0000 0000
17H	TM1_LA (Don't write FF)	0000 0000	0000 0000
18H	TM1_DAT (Read only)	1111 1111	1111 1111
19H	TM2_CTL	0000 0000	0000 0000
1AH	TM2_LA (Don't write FF)	0000 0000	0000 0000
1BH	TM2_DAT (Read only)	1111 1111	1111 1111
1CH	TM3_CTL1	0000 0000	0000 0000
1DH	TM3_CTL2	0000 0000	0000 0000
1EH	TM3_LA (Don't write FF)	0000 0000	0000 0000
1FH	TM3_DAT (Read only)	1111 1111	1111 1111
20H	TM4_CTL	0000 0000	0000 0000
21H	TM4_LA (Don't write FF)	0000 0000	0000 0000
22H	TM4_DAT (Read only)	1111 1111	1111 1111
25H	IRQM	0000 0000	0000 0000
26H	IRQF	x000 0000	x000 0000
29H	AD_CTL1	0x0x 0000	0x0x 0000
2AH	AD_CTL2	0xxx xx00	0xxx xx00
2BH	AD_CTL3	xxxx 0000	xxxx 0000
2CH	AD_DATL	0000 0000	0000 0000
2DH	AD_DATH	X000 0000	X000 0000
2FH	CMP_CTL	xxxx xx00	xxxx xx00
30H	FREQ_CTL	000x x000	000x x000
31H	PA_PLU	0000 0000	0000 0000
32H	PB_PLU	0000 0000	0000 0000
33H	PC_PLU	0000 0000	0000 0000
34H	PD_PLU	Xx00 0000	Xx00 0000
35H	IO_CTL	Xxxx 0000	Xxxx 0000
3AH	PA_INTE	0000 0000	0000 0000
3DH	WDT_CTL	1xxx x111	1xxx x111
3EH	TAB_BNK	Xxxx x000	Xxxx x000
3FH	SYS_CTL	0xxx xx00	0xxx xx00

x: 未知的; p: 原来的数值; ? : 依据条件的数值; -: 不执行, 清“0”



6.6 中断

MK7A22P提供7种外部中断（PA0~7），3种内部定时器/结果计数器中断和1种A/D转换器中断。IRQM和IRQF寄存器用来控制或判断所有中断的请求状态。外部中断通过PA0~7的脚位变化触发，相关的中断请求标记

（PAF；IRQF的bit 5）将被设置。通过设置A/D转换器请求标记（ADCF；IRQF的bit 6），A/D转换器中断被初始化，当结束A/D转换，中断发生。

IRQM用来使能/禁止中断，IRQF用来指出是哪一种中断发生。如果特殊IRQM不能先使能则硬体中断将不会发生。但不管IRQM使能或禁止，IRQF都会有状态反应。例如，使用者使能TM1来开始计数，如果IRQM的bit 1使能，当定时器溢出，硬体中断将会发生，IRQF的bit 1将被设置，与此同时，程序将跳到中断向量。使用者应清除中断服务程序中的IRQF，否则中断将不会再次发生。另一种情况是如果IRQM的bit 1禁止，当定时器溢出时，中断将不会产生，但IRQF的bit 1仍被设置。没有硬体中断发生，程序将不会跳到中断向量。

A. IRQM (\$25H)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRQM	INTM	ADCM	PAM	TM4M	TM3M/PWM2M	TM2M	TM1M/PWM1M	TM0M

- Bit7（INTM）：球形使能位

0：禁止所有中断屏蔽

1：使能所有中断屏蔽

< 注 > 当中断正在进行时，INTM将会被设置为“0”以防止其他中断的发生。当中断完成后，IRETI指令将会设置INTM为“1”。

- Bit6（ADCM）：ADC 结束转换（EOC）中断使能

0：禁止中断

1：使能中断

- Bit5（PAM）：PA 中断使能

0：禁止中断

1：使能中断

- Bit4（TM4M）：TM4 中断使能

0：禁止中断

1：使能中断

- Bit3（TM3M/PWM2M）：TM3/PWM2 中断使能

0：禁止中断

1：使能中断

- Bit2（TM2M）：TM2 中断使能

0：禁止中断

1：使能中断



- Bit1 (TM2M/PWM1M): TM1/PWM1 中断使能
0: 禁止中断
1: 使能中断
- Bit0 (TM0M): TM0 中断使能
0: 禁止中断
1: 使能中断

B. IRQF (\$26H)

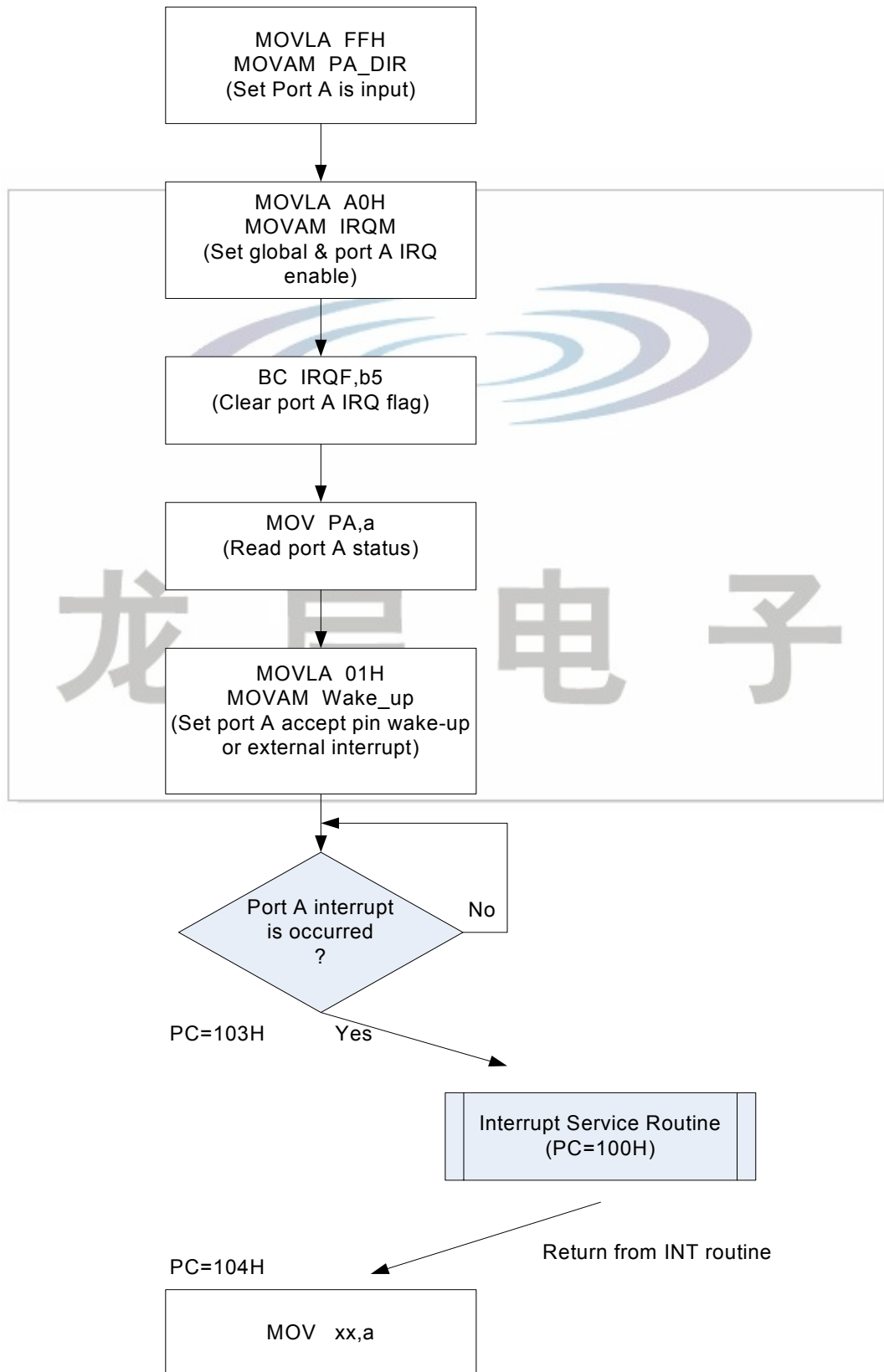
Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRQF	--	ADCF	PAF	TM4F	TM3F/PWM2F	TM2F	TM1F/PWM1F	TM0F

- Bit6 (ADCF): ADC 结束转换中断请求标记
0: 结束转换中断请求关
1: 结束转换中断请求开
- Bit5 (PAF): PA0~7 中断请求标记
0: PA 中断请求关
1: PA 中断请求开
- Bit4 (TM4F): TM4 中断请求标记
0: TM4 中断请求关
1: TM4 中断请求开
- Bit3 (TM3F/PWM2F): TM3/PWM2 中断请求标记
0: TM3/PWM2 溢出中断请求关
1: TM3/PWM2 溢出中断请求开
- Bit2 (TM2F): TM3 中断请求标记
0: TM2 溢出中断请求关
1: TM2 溢出中断请求开
- Bit1 (TM1F/PWM1F): TM1/PWM1 中断标记
0: TM1/PWM1 溢出或 PWM1 中断请求关
1: TM1/PWM1 溢出或 PWM1 中断请求开
- Bit0 (TM0F): TM0 中断请求标记
0: TM0 溢出中断请求关
1: TM0 溢出中断请求开

6.6.1 外部中断/唤醒功能

A口 (PA) 提供外部中断和唤醒功能。当芯片不处于睡眠模式，PA输入信号将做为外部中断工作。当外部中断发生，程序将会跳到004H (中断向量)。如果芯片处于睡眠模式，PA输入信号将做为唤醒功能工作。当唤醒信号输入，芯片将会让系统时钟首先工作，然后等待唤醒定时器 (由WDT_CTL寄存器\$3PH设置) 溢出，再之后，程序将跳到004H。以下流程图说明如何设置A口作为外部中断或唤醒功能工作。

Dj e t f e ^ h % l f 7 _ f k d X c z f d



图表：A端口唤醒和外部中断设置流程



6.7 ADC

MK7A22P提供12个通道和15-bits (11bit + 4bit) 协议A/D转换器。A/D转换器包含5个寄存器，分别是AD_CTL1 (29H)，AD_CTL2 (2AH)，AD_CTL3 (2BH)，AD_DATL (2DH) 及AD_DATH (2EH)。

A. AD_CTL1 (\$29H)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL1	EN	--	MODE	--	CHSEL3	CHSEL2	CHSEL1	CHSEL0

- Bit7 (EN): ADC 使能位

0: ADC 禁止

1: ADC 使能

< 注 > 当结束转换，此位将会自动设置为“0”

- Bit5 (MODE): ADC 模式选择位

0: ADC 通道作为 A/D 转换工作

1: ADC 通道作为比较器工作

< 注 > (a) 如果此位是“1”，Vin 数据将会与 AD_DAT 作比较，结果被存储在 AD_CTL2 Bit7。如果此位是“0”，Vin 被转换成 15-bit 数位数据并保存在 AD_DAT 寄存器。

(b) Vin: ADC 通道的输入电压

- Bit3~0 (CHSEL3~0): ADC 输入通道选择位

Bit3	Bit2	Bit1	Bit0	Input channel
CHSEL3	CHSEL2	CHSEL1	CHSEL0	
0	0	0	0	Channel 0, PA0 pin
0	0	0	1	Channel 1, PA1 pin
0	0	1	0	Channel 2, PA2 pin
0	0	1	1	Channel 3, PA3 pin
0	1	0	0	Channel 4, PA4 pin
0	1	0	1	Channel 5, PA5 pin
0	1	1	0	Channel 6, PA6 pin
0	1	1	1	Channel 7, PA7 pin
1	0	0	0	Channel 8, PC7 pin
1	0	0	1	Channel 9, PD1 pin
1	0	1	0	Channel 10, PD2 pin
1	0	1	1	Channel 11, PD3 pin



B. AD_CTL2 (\$2AH)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL2	RSUT	--	--	--	--	--	CKSEL1	CKSEL0

- Bit7 (RSUT): 比较模式结果位
0: $V_{in} < AD_DAT$
1: $V_{in} \geq AD_DAT$
- Bit1~0 (CKSEL1~0): ADC 转换时钟源选择位

Bit1	Bit0	转换时钟
CKSEL1	CKSEL0	
0	0	系统时钟 X2
0	1	系统时钟 X8
1	0	系统时钟 X32
1	1	系统时钟 X128

< 注 > 转换时钟决定转换率和精度。如果选择的是快速转换时钟，那将减少精度。如果使用者想得到更精确的 A/D 数据，推荐使用慢速。

C. AD_CTL3 (\$2BH)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_CTL3	--	--	--	--	PISEL3	PISEL2	PISEL1	PISEL0

- Bit3~0 (PBSEL3~0): ADC 通道输入模式选择位，设置每个 Port 是模拟输入还是数字输入

Bit3	Bit2	Bit1	Bit0	配置
PISEL3	PISEL2	PISEL1	PISEL0	
0	0	0	0	所有 port 为数字输入
0	0	0	1	AN0
0	0	1	0	AN0~1
0	0	1	1	AN0~2
0	1	0	0	AN0~3
0	1	0	1	AN0~4
0	1	1	0	AN0~5
0	1	1	1	AN0~6
1	0	0	0	AN0~7
1	0	0	1	AN0~8
1	0	1	0	AN0~9
1	0	1	1	AN0~10
1	1	0	0	AN0~11

< 注 > 最小化能耗，所有 I/O 脚在进入睡眠模式前都会被小心控制



D. AD_DATL (2CH)

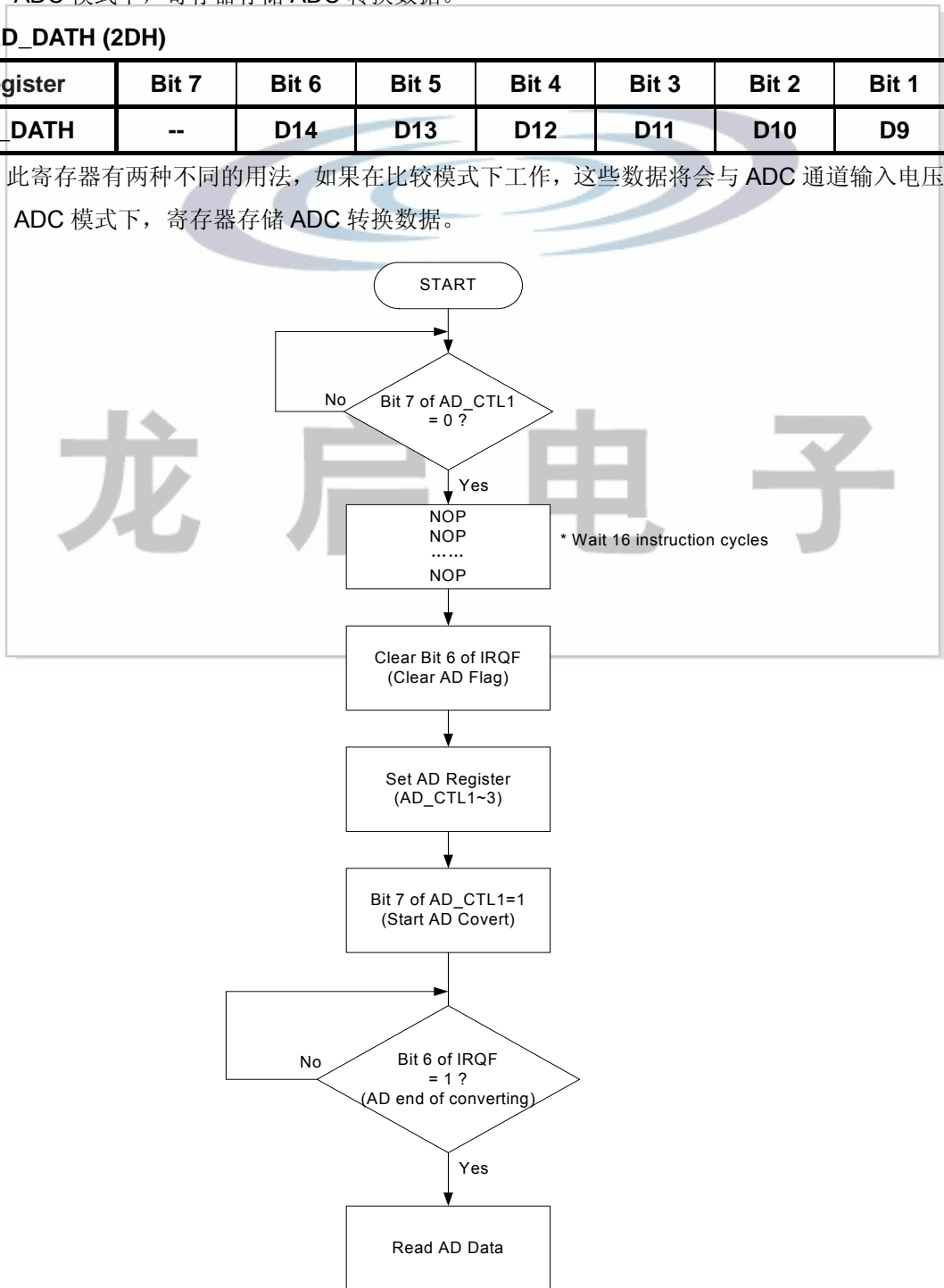
Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_DATL	D7	D6	D5	D4	D3	D2	D1	D0

< 注 > 此寄存器有两种不同的用法，如果在比较模式下工作，这些数据将会与 ADC 通道输入电压相比较。在 ADC 模式下，寄存器存储 ADC 转换数据。

E.AD_DATH (2DH)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AD_DATH	--	D14	D13	D12	D11	D10	D9	D8

< 注 > 此寄存器有两种不同的用法，如果在比较模式下工作，这些数据将会与 ADC 通道输入电压相比较。在 ADC 模式下，寄存器存储 ADC 转换数据。



图表：ADC设置流程



6.8 表格查询功能

MK7A22P提供表格查询功能。查询表格可以置于ROM空间的任何位置。TABRDL指令是读ROM表格的低字节。TABRDH是读高字节。TAB_BNK寄存器用于定义表格位置（3+8=11bits-address bit，2¹¹=2Kbytes-data byte）的高位（MSB）地址。

6.8.1 TAB_BNK (\$3EH)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TAB_BNK	--	--	--	--	--	BANK2	BANK1	BANK0

- Bit2~0 (BANK2~0): 高字节表格位置选择位

Bit2	Bit1	Bit0	BANK 选择
BANK2	BANK1	BANK0	
0	0	0	000 XXXX XXXX 表格位置
0	0	1	001 XXXX XXXX 表格位置
0	1	0	010 XXXX XXXX 表格位置
.	.	.	.
1	1	1	111 XXXX XXXX 表格位置

6.8.2 表格查询举例程序

以下程序是如何编程表格查询功能的例子，TABRDL和TABRDH的结果将会是55H和AAH（地址是0704H）。

```
#DEFINE    TAB_BNK    3EH            ; 定义地址，RAM 的 3EH 定义为 TAB_BNK

BUFA      EQU        43H            ; 定义地址，RAM 的 43H 定义为 BUFA

(address)  ORG        0700H         ; 程序从 ROM 的 0700H 开始
0700H     MOVLA      00H            ; 保存 00H 到 A 寄存器
0701H     DW         1122H         ; 存储 1122H 在 ROM 的 0701H
0702H     DW         3344H         ; 存储 3344H 在 ROM 的 0702H
0703H     DW         6677H         ; 存储 6677H 在 ROM 的 0703H
0704H     DW         55AAH         ; 存储 55AAH 在 ROM 的 0704H

          MOVLA      04H            ; 保存 04H 到 A 寄存器 ( 低位地址 )
          MOVAM     BUFA           ; 保存 A 寄存器的数值到 BUFA
          MOVLA      0FH            ; 保存 0FH 到 A 寄存器 ( 高位地址 )
          MOVAM     TAB_BNK        ; 保存 A 寄存器的数值到 TAB_BNK
          TABRDL    BUFA           ; 查询 TAB_BNK 的低字节数值和 BUFA 指定地址，
                                   保存到 A 寄存器

          TABRDH    BUFA           ; 查询 TAB_BNK 的高字节数值和 BUFA 指定地址，
                                   保存到 A 寄存器
```



6.9 系统控制

MK7A22P提供Auto-Bank功能和双重时钟操作模式。Auto-bank的意思是当程序跳跃，系统将自动保存PC的高字节，以防止进位溢出引起程序计数器错误。双重时钟模式有内部RC和外部晶振时钟源。使用者可以在同一时间使用双重时钟。例如，内部RC（4MHz）常用作系统时钟源，外部晶振（32KHz）用作计数器时钟源。

6.10.1 SYS_CTL (\$3FH)

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SYS_CTL	CLKS	--	--	--	--	--	STPRC	STPOSC

- Bit7 (CLKS): 系统时钟源选择位，只有在双重时钟模式（内部RC 4MHz和晶振）或双重RC模式（内部RC 4MHz和外部RC）下激活

双重时钟模式:

0: 系统时钟是内部RC 4MHz

1: 系统时钟是OSC（双重时钟模式）或外部RC（双重RC模式）

单时钟模式:

不使用

- Bit1 (STPRC): RC（内部RC 4MHz）振荡器控制位

双重时钟模式或双重RC模式:

0: 4MHz RC振荡开

1: 4MHz RC振荡关

其他模式:

不使用

< 注 > 在设置 RC 振荡关之前，首先将 CLKS 转换到 OSC 振荡

- Bit0 (STPOSC): OSC（晶振）振荡器控制位

双重时钟模式或双重RC模式:

0: 外部OSC振荡开（双重时钟模式）或外部RC开（双重RC模式）

1: 外部OSC振荡关（双重时钟模式）或外部RC开（双重RC模式）

其他模式:

不使用

< 注 > 在设置 OSC 振荡关之前，首先将 CLKS 转换到 RC 振荡



6.10 程序计数器 - PC

MK7A22P有一个11-bits程序计数器（PC），包含PCL（8-bits）和PCH（3bits）。PC习惯于被存储在程序通道内。一旦使用者改变PCL的数值，程序将会跳到指示页面。

Ex1: PCH=01H, PCL=02H+10H=12H, 程序将跳到PC=112H

Ex2: PCH=01H, PCL=F0H+30H=20H带进位1, 程序将跳到PC=220H, 但PCH仍旧是01H

- < 注 > (a) 当执行 IRET 和 IRETI, PCH 数据将不会被更新
(b) 当执行 LGOTO, LCALL, JZ, JC 和 RET, PCH 将会在精确的操作后被更新

示例 1:

以下程序说明 PCL 和 PCH 如何在直接精确的操作下工作。

```
#DEFINE PCL 01H ; 定义地址, RAM 的 01H 定义为 PCL
#DEFINE PCH 02H ; 定义地址, RAM 的 02H 定义为 PCH
#DEFINE ADMIN 41H ; 协助 PCL 操作
(address)
1C0H MOVLA HIGH P1 ; 保存 P1 (1C6H) 高字节地址到 A 寄存器
; PC=1C0H, PCL=C0H, PCH=00H
1C1H MOVAM PCH ; 保存 A 寄存器到 PCH (为了避免 PCL 操作的跳跃
; 错误, 首先存储真正的跳跃高字节地址到 PCH)
; PC=1C1H, PC =C1H, PCH=01H
1C2H MOVLA 4BH ; 保存 4BH 到 A 寄存器 (ADDAM PCL, M 的地址
; 是 1C5H, 准备跳到 210H, PCL 增加 210H-1C5H
; =4BH)
; PC=1C2H, PC =C2H, PCH=01H
1C3H MOVAM ADMIN ; 准备 PCL 操作
; PC=1C3H, PCL=C3H, PCH=01H
1C4H DEC ADMIN, a ; ADMIN-1 (真正的跳跃发生在 1C6H, 不在 1C5H,
; 因此 1C6H+(4BH-1H)=210H)
; PC=1C4H, PCL=C4H, PCH=01H
1C5H ADD PCL, M ; PCL 用 A 寄存器增加, 结果存储在 PCL
; PC=1C5H, PCL=C5H, PCH=01H
P1:1C6H NOP ; 跳到 0210H
; PC=1C6H, PCL=C6+4AH=10H 带进位 1, 进位将
; 用 PCH 计数, PCH=01H, 用途 PC 高字节地址将是
; PCH+PCL's 进位=02H。程序将跳到 PC=210H
210H MOVLA 00H ; 用途功能部分
; PC=210H, PCL=10H, PCH=01H
```




示例 2:

以下程序说明 PCL 和 PCH 如何在直接精确的操作下工作。

(address)			
1C0H	MOVLA	03H	; 保存 03H 到 A 寄存器 ; PC=1C0H, PCL=C0H, PCH=00H
1C1H	MOVAM	PCH	; 保存 A 寄存器到 PCH ; PC=1C1H, PCL=C1H, PCH=03H
1C2H	MOVLA	4BH	; 保存 4BH 到 A 寄存器 ; PC=1C2H, PCL=C2H, PCH=03H
1C3H	MOVAM	ADMIN	; 准备 PCL 操作 ; PC=1C3H, PCL=C3H, PCH=01H
1C4H	DEC	ADMIN, a	; ADMIN-1 (真正的跳跃发生在 1C6H, 不在 1C5H, 因此 1C6H+(4BH-1H)=210H) ; PC=1C4H, PCL=C4H, PCH=03H
1C5H	ADD	PCL,M	; PCL 用 A 寄存器增加, 结果存储在 PCL ; PC=1C5H, PCL=C5H, PCH=03H
1C6H	NOP		; 跳到 410H ; PC=1C6H, PCL=C6+4AH=10H 带进位 1, 进位将 用 PCH 计数, PCH=03H, 用途 PC 高字节地址将是 PCH+PCL's 进位=04H。程序将跳到 PC=410H
410H	MOVLA	00H	; 用途功能部分 ; PC=410H, PCL=10H, PCH=03H

示例 3:

以下程序说明 PCL 和 PCH 如何通过 A 寄存器工作。

(address)			
018H	MOVLA	02H	; 保存 02H 到 A 寄存器
019H	MOVAM	PCH	; 保存 A 寄存器到 PCH (用途地址是 200H, 因此存 储 “02H” 到 PCH)
01AH	MOVLA	00H	; 保存 00H 到 A 寄存器
01CH	MOVAM	PCL	; 保存 A 寄存器到 PCL (用途地址是 200H, 因此存 储 “00H” 到 PCL)
01DH	NOP		; 跳到 200H
200H	MOVLA	00H	; 用途功能部分



6.11 STATUS寄存器

STATUS寄存器是一个包含零标记 (Z)，进位标记 (C)，四位进位标记 (DC)，掉电标记 (\overline{PD})，看门狗定时器溢出标记 (\overline{TO}) 的8位寄存器，它用于记录状态信息。

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	--	--	--	\overline{TO}	\overline{PD}	Z	DC	C

- Bit4 (\overline{TO}): 定时器溢出标记位
- Bit3 (\overline{PD}): 掉电标记位

\overline{TO}	\overline{PD}	说明
0	0	在睡眠模式中 WDT 定时器溢出
0	1	在普通模式中 WDT 定时器溢出
1	0	在睡眠模式中给 RESETB 输入一个“低电平”
1	1	上电复位
Unchanged	Unchanged	在普通模式中给 RESETB 输入一个“低电平”

- Bit2 (Z): 零标记位
0: 逻辑操作结果不是0
1: 逻辑操作结果是0
- Bit1 (DC): 四位进位与四位借位标记位

加指令:

0: 无四位进位发生

1: 四位进位发生

减指令:

0: 四位借位发生

1: 无四位借位发生

- Bit0 (C): 进位与借位标记位

加指令:

0: 无进位发生

1: 从 MSB 进位发生

减指令:

0: 从MSB借位发生

1: 无借位发生



SL1 M, m	Rotate left from m to itself m[6:0] → m[7:1] 1 → m[0]	1	None	1110 0111 MMMM MMMM
RR M, a	Rotate right from m to acc 0 → acc[7] m[7:1] → acc[6:0]	1	None	1110 1000 MMMM MMMM
RR M, m	Rotate right from m to itself M[0] → m[7] m[7:1] → m[6:0]	1	None	1110 1001 MMMM MMMM
RRC M, a	Rotate right from m to acc m[0] → c, c → acc[7] m[7:1] → acc[6:0]	1	C	1110 1010 MMMM MMMM
RRC M, m	Rotate right from m to itself m[0] → c, c → m[7] m[7:1] → m[6:0]	1	C	1110 1011 MMMM MMMM
SR0 M, a	Rotate right from m to acc 0 → acc[7] m[7:1] → acc[6:0]	1	None	1110 1100 MMMM MMMM
SR0 M, m	Rotate right from m to itself 0 → m[7] m[7:1] → m[6:0]	1	None	1110 1101 MMMM MMMM
SR1 M, a	Rotate right from m to acc 1 → acc[7] m[7:1] → acc[6:0]	1	None	1110 1110 MMMM MMMM
SR1 M, m	Rotate right from m to itself 1 → m[7] m[7:1] → m[6:0]	1	None	1110 1111 MMMM MMMM
SWAP M, a	m[7:4] → acc[3:0] m[3:0] → acc[7:4]	1	None	1011 1100 MMMM MMMM
SWAP M, m	m[7:4] ↔ m[3:0]	1	None	1011 1101 MMMM MMMM
XOR M, a	(M) xor (acc) → (acc)	1	Z	1011 0110 MMMM MMMM
XOR M, m	(M) xor (acc) → (M)	1	Z	1011 0111 MMMM MMMM
XORLA I	Immediate xor (acc) → (acc)	1	Z	1111 1001 iiiiii iiiiii
MATHEMATICS				
ADD M, a	(M)+(acc) → (acc)	1	C, DC, Z	1010 1010 MMMM MMMM
ADD M, m	(M)+(acc) → (M)	1	C, DC, Z	1010 1011 MMMM MMMM



ADDC M,a	(M)+(acc) + (carry) → (acc)	1	C, DC, Z	1011 1010 MMMM MMMM
ADDC M,m	(M)+(acc) + (carry) → (M)	1	C, DC, Z	1011 1011 MMMM MMMM
ADDLA I	Immediate + (acc) → (acc)	1	C, DC, Z	1111 1010 MMMM MMMM
BC M, bn	Clear bit n of (M)	1	None	1001 1bbb MMMM MMMM
BS M, bn	Set bit n of (M)	1	None	1001 0bbb MMMM MMMM
CLRA	Clear accumulator	1	Z	1010 0010 0000 0000
CLR M	Clear memory M	1	Z	1010 0011 MMMM MMMM
TABRDL M	Read low byte ROM table to (acc) ROM table address={TB_BNK,index of M }	2	None	1101 1000 MMMM MMMM
TABRDH M	Read high byte ROM table to (acc) ROM table address={TB_BNK,index of M }	2	None	1101 1001 MMMM MMMM
DAA M, a	Decimal Adjust M to ACC If ACC[3:0] > 9 or DC=1 Then ACC[3:0] ← ACC[3:0]+6, DC1=DC else ACC[3:0] ← ACC[3:0], DC1=0 If ACC[7:4]+DC1 > 9 or C=1 Then ACC[7:4] ← ACC[7:4]+6+DC1, C=1 else ACC[7:4] ← ACC[7:4]+DC1, C=C	1	C	1101 0110 MMMM MMMM
DAA M, m	Decimal Adjust M to memory If ACC[3:0] > 9 or DC=1 Then M[3:0] ← ACC[3:0]+6, DC1=DC else M[3:0] ← ACC[3:0], DC1=0 If ACC[7:4]+DC1 > 9 or C=1 Then M[7:4] ← ACC[7:4]+6+DC1, C=1 else M[7:4] ← ACC[7:4]+DC1, C=C	1	C	1101 0111 MMMM MMMM
DAS M, a	Decimal Adjust M to ACC If ACC[3:0] > 9 Then ACC[3:0] ← ACC[3:0]-6, Else ACC[3:0] ← ACC[3:0], If ACC[7:4] > 9 or C=1 Then ACC[7:4] ← ACC[7:4]-6, C=1 else ACC[7:4] ← ACC[7:4], C=C	1	C	1101 1110 MMMM MMMM



DAS M, m	Decimal Adjust M to memory If ACC[3:0] > 9 or DC=1 Then M[3:0] ← ACC[3:0]-6, else M[3:0] ← ACC[3:0], DC1=0 If ACC[7:4]+DC1 > 9 or C=1 Then M[7:4] ← ACC[7:4]-6, C=1 else M[7:4] ← ACC[7:4], C=C	1	C	1101 1111 MMMM MMMM
DEC M, a	(M) - 1 → (acc)	1	Z	1010 1100 MMMM MMMM
DEC M, m	(M) - 1 → (M)	1	Z	1010 1101 MMMM MMMM
INC M, a	(M) + 1 → (acc)	1	Z	1011 0000 MMMM MMMM
INC M, m	(M) + 1 → (M)	1	Z	1011 0001 MMMM MMMM
MOVAM m	(acc) → (M)	1	None	1010 0001 MMMM MMMM
MOV M, a	(M) → (acc)	1	Z	1010 0110 MMMM MMMM
MOV M, m	(M) → (M)	1	Z	1010 0111 MMMM MMMM
MOV2 M, a	(M) → (acc)	1	None	1111 0110 MMMM MMMM
MOV2 M, m	(M) → (M)	1	None	1111 0111 MMMM MMMM
MOVLA I	Immediate data → acc	1	None	1111 0000 iiiii iiiii
SUBLA I	(immediate data)-(Acc)→(Acc)	1	C, DC, Z	1111 0100 iiiii iiiii
SUB M, m	(M)-(acc) → (M)	1	C, DC, Z	1011 0101 MMMM MMMM
SUB M, a	(M)-(acc) → (acc)	1	C, DC, Z	1011 0100 MMMM MMMM
OTHER OPERATION				
NOP	No operation	1	None	1111 1111 1111 1111
CLRWDT	Clear watch-dog register	1	$\overline{TO}, \overline{PD}$	1111 1111 1111 0000
RET	Return (for lcall instruction)	2	None	1111 1111 1111 0001
IRETI	Return and enable INTM(for IRQ)	2	None	1111 1111 1111 0010
IRET	Return (for IRQ)	2	None	1111 1111 1111 0011
SLEEP	Enter sleep (saving) mode	1	$\overline{TO}, \overline{PD}$	1111 1111 1111 0100
CONDITION OPERATION				
BTSC M, bn	If (bit n of (M))=0, skip next instruction	1 or 2	None	1000 1bbb MMMM MMMM
BTSS M, bn	If (bit n of (M))=1, skip next instruction	1 or 2	None	1000 0bbb MMMM MMMM
DECSZ M, a	(M) - 1 →(acc), skip if (acc) = 0	1 or 2	None	1010 1110 MMMM MMMM
DECSZ M, m	(M) - 1 → (M), skip if (M) = 0	1 or 2	None	1010 1111 MMMM MMMM
INCSZ M, a	(M) + 1 →(acc), skip if (acc) = 0	1 or 2	None	1011 0010 MMMM MMMM
INCSZ M, m	(M) + 1 → (M), skip if (M) = 0	1 or 2	None	1011 0011 MMMM MMMM
TMSS	If (acc) =0, skip next instruction	1 or 2	None	1011 1000 XXXX XXXX
TMSC M	If (M) = 0, skip next instruction	1 or 2	None	1011 1001 MMMM MMMM



8. 电气特性

8.1 绝对最大额定值

电源电压 Vss-0.3V to Vss+5.5V	存储温度 -50°C to 125°C
输入电压 Vss-0.3V to VDD+0.3V	工作温度 -40°C to 85°C

< 注 > 这里仅强调额定值，超出“绝对最大额定值”指定的范围会对芯片造成严重伤害。在其他条件下（规格书列出的除外），此芯片的功能操作并不意味长期暴露在极端条件下会影响芯片的可靠性。

8.2 直流电特性

龙 启 电 子



Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		VDD	Conditions				
VDD	Operating Voltage	---	f _{sys} =4MHZ	2.0		5.5	V
			f _{sys} =8MHZ	2.5		5.5	
			f _{sys} =12MHZ	3.3		5.5	
I _{DD1}	Operating Current (Crystal OSC)	3.3V	No Load , f _{sys} =4MHZ		0.8		mA
		5V	ADC disable .		1.7		
I _{DD2}	Operating Current (RC OSC)	3.3V	No Load , f _{sys} =4MHZ		0.8		mA
		5V	ADC disable .		1.4		
I _{DD3}	Operating Current (Crystal OSC)	5V	No Load , f _{sys} =8MHZ ADC disable		3		mA
I _{DD4}	Operating Current (Crystal OSC)	5V	No Load , f _{sys} =12MHZ ADC disable		4.2		mA
V _{IH}	Input High Voltage	5V	I/O Port	2		Vdd	V
V _{IL}	Input Low Voltage	5V	I/O Port			0.8	V
I _{STB}	Standby Current	5V	WDT disable			1	μA
			WDT enable			6	
		3V	WDT disable			1	
			WDT enable			2	
I _{IL}	Input LeakageCurrent	5V	Vin=VDD, VSS			1	μA
I _{OH}	I/O Port Driving Current	5.5V	Voh=5V			9.9	mA
			Voh=4.5V			17.6	
			Voh=4V			24.8	
I _{OL}	I/O Port Sink Current	5.5V	Vol=0.5V			24.5	mA
			Vol=0.75V			35.3	
			Vol=1V			43.8	
R _{PH}	Pull-high Resistance	3V		80	100	120	KΩ
		5V		30	50	70	KΩ
V _{AD}	A/D input Voltage	--		0		VDD	V
R _{AD}	Resolution					15	Bits
DNL	A/D Differential Non- Linear	--				4	LSB
INL	A/D Integral Non- Linear	--				6	LSB
I _{ADc}	A/D Operating Current	3.3V			200		uA
		5V			700		



8.3 交流电特性

Symbol	Parameter	Test Conditions		Min	Typ	Max	Unit
		Conditions	VDD				
f _{sys1}	System Clock	LS Crystal mode	5V	32		200	Khz
			3V	32		200	
f _{sys2}	System Clock	NS Crystal mode	5V	0.2		10	Mhz
			3V	0.2		10	
f _{sys3}	System Clock	HS Crystal mode	5V	10		20	Mhz
			3V	10		20	
f _{sys4}	System Clock	RC mode	5V			6	Mhz
			3V			4	
T _{wdt}	Watchdog Timer		5V		20		mS
			3V				
T _{rnt}	Reset Hold Time		5V		20		mS
			3V				
T _{AD}	A/D clock period			3			us
T _{ADC}	A/D Conversion Time				34		t _{AD}
T _{ADCS}	A/D Sampling Time				8		t _{AD}

8.4 外部 RC 表格

R value(R2)	C value(C4)	RC Frequency	R connect to (VDD, OSC1)
4.9M	0.1u (need)	32KHZ	The capacitor is need for stabile frequency
250K	0.1u (suggest)	455KHZ	
116K	0.1u (suggest)	1MHZ	
60K	0.1u (suggest)	2MHZ	
32K	0.1u (suggest)	4MHZ	
18K	0.1u (suggest)	8MHZ	
14K	0.1u (suggest)	10MHZ	

