



## 内容

### 一、ATmega单片机概述

- 1、AVR单片机简介（主要特性、选型）
- 2、ATmega128单片机引脚

### 二、ATmega128单片机系统结构

- 1、ATmega128的CPU内核
- 2、ATmega128 存储器
- 3、主要时钟系统
- 4、系统控制和复位
- 5、ATmega128 的中断向量
- 6、I/O端口
- 7、定时器/ 计数器（T/C）
- 8、模数转换器 A/D



# 1、AVR单片机简介

## ■ ATMEL公司介绍

- 是世界上著名的高性能、低功耗、非易失性存储器和数字集成电路的一流半导体制造公司。
- 1997年，ATMEL公司出于市场需求，推出了全新配置的精简指令集RISC单片机高速8位单片机，简称为AVR。
- 广泛应用于计算机外设、工业实时控制，仪器仪表、通信设备、家用电器等各个领域。



## 1.1 AVR单片机主要特性

### ■ 衡量单片机性能的重要指标

- 高可靠性、功能强、高速度、低功耗、低价位

### AVR单片机

- 1) 有**高可靠性**为后盾。
- 2) 在软/硬件开销、速度、性能和成本多方面取得优化平衡，是性价比的单片机。
- 3) 内嵌高质量的**Flash**程序存储器，擦写方便，便于产品的调试、开发、生产、更新。
- 4) I/O端口资源灵活、功能强大。
- 5) 单片机内具备多种独立的时钟分频器。





## 1.1 AVR单片机主要特性

- 6) 高波特率的可靠通信
- 7) 包括多种电路，可增强嵌入式系统的可靠性。
  - 电路：自动上电复位、看门狗、掉电检测
  - 多个复位源等
- 8) 具有多种省电休眠模式、宽电压运行（2.7~5V），抗干扰能力强，可降低一般8位机中的软件抗干扰设计的工作量和硬件的使用量。
- 9) 集成多种器件和多种功能。



## 1.2 AVR系列单片机的选型

AVR单片机有3个档次：

- 低档Tiny系列单片机， 20脚
  - Tiny 11/12/13/15/26/28
  - AT89C1051, AT89C1052
- 中档(标准) AT90S系列单片机， 40脚
  - AT90S1200/2313/8515/8535
  - AT89C51
- 高档ATmega系列单片机                      64脚
  - ATmega8/16/32/64/128
  - 存储容量为8/16/32/64/128KB
  - ATmega8515/8535
- **ATmega128单片机——基于AVR低功耗CMOS 8位微控制器，近1MIPS/MHz。**



## ATmega128产品特点

### 1) RISC结构

- 指令条数 133条指令 – 大多数可以在一个时钟周期内完成
- 存储器 32 x 8位 通用工作寄存器 + 外设控制寄存器
- 工作频率 工作于16 MHz 时性能高达16 MIPS
- 片内乘法器 2个 只需两个时钟周期的硬件乘法器

### 2) 非易失存储器

- 片上Flash 128K Bytes,  $10^4$ 次擦写
- EEPROM 4K Bytes,  $10^5$ 次擦写
- 内部SRAM 4K
- 可选外部存储器 64K
- 可编程安全锁、片内可编程SPI接口

### 3) JTAG接口

- 边界扫描特性, 扩展芯片调试支持, 通过其进行可编程Flash、EEPROM



## ATmega128产品特点

### 4) 外设特性

- 定时器/计数器 (2个8位)
- 扩展定时器/计数器 (2个16位)
- 实时时钟计数器 (1个)
- PWM通道
- 输出比较调制器
- 8通道10位ADC:
  - 8个单端通道、7个差分通道、2个可调增益的差分通道
- 面向字节的双线接口
- 可编程的连续串口UART 2个
- 主/从SPI串口、可编程看门狗、片内模拟比较器





## ATmega128产品特点

### 5) 特殊微控制器特性

- 复位、中断源、省电模式等

### 6) I/O和封装

- 53个可编程I/O口线
- 64引脚TQFP与64引脚MLF封装

### 7) 工作电压

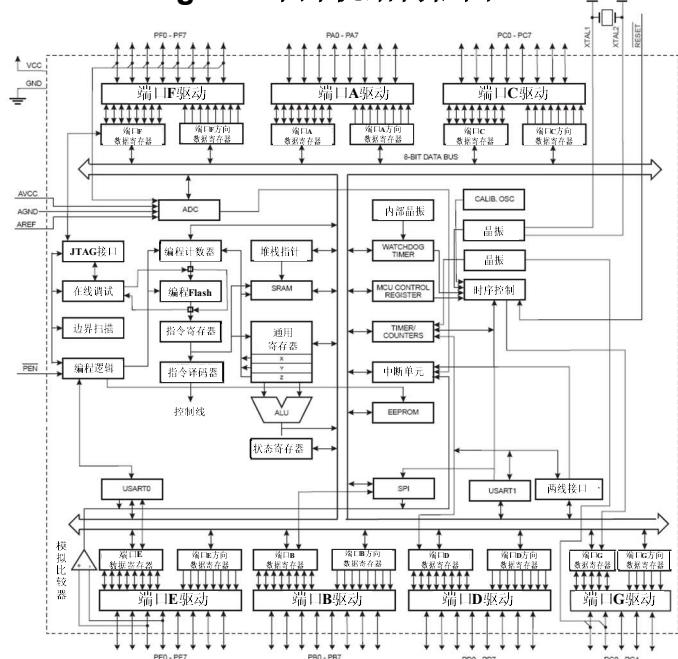
- 2.7~5.5V (ATmega128L)
- 4.5~5.5V (ATmega128)

### 8) 速度级别

- 0~8MHZ (ATmega128L)
- 0~16MHZ (ATmega128)



## ATmega128单片机结构框图





## 128的引脚说明:

除端口外的引脚说明: 1) ~8)

端口引脚说明: 9) ~11)

- 1) **VCC**: 数字电路的电源
- 2) **GND**: 地
- 3) **RESET**: 复位输入引脚
- 4) **XTAL1**: 反向振荡放大器及片内时钟操作电路的输入
- 5) **XTAL2**: 反向振荡放大器的输出
- 6) **AVCC**: **AVCC**为端口F及ADC的电源
- 7) **AREF**: 为ADC的模拟基准输入引脚
- 8) **PEN**: 为SPI串行下载的使能引脚



## 128的引脚说明:

——端口引脚说明: 9) ~11)

- 9) 8位双向I/O端口
  - 端口A(PA7~PA0)
  - 端口B(PB7~PB0)
  - 端口C(PC7~PC0)
  - 端口D(PD7~PD0)
  - 端口E(PE7~PE0)
- 10) 8位端口F(PF7~PF0)
  - 为ADC的模拟输入引脚或作为8位双向I/O端口。
  - 可以作为JTAG接口
- 11) 5位双向I/O端口G(PG4~PG0)



## ATmega128单片机的系统结构 主要内容

- 1、ATmega128的CPU内核
- 2、ATmega128 存储器
- 3、主要时钟系统
- 4、系统控制和复位
- 5、中断向量
- 6、I/O端口
- 7、定时器/ 计数器 (T/C)
- 8、模数转换器 A/D





# 1、ATmega128的CPU内核

## 1.1 AVR CPU内核的结构

- AVR 采用Harvard 结构，具有独立的数据和程序总线。

## 1.2 状态寄存器

AVR 中断寄存器 – SREG

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

Bit 7 – I: 全局中断使能

## 1.3 通用寄存器结构

## 1.4 X、Y、Z寄存器

## 1.5 堆栈指针

## 1.6 复位和中断处理



# 2、ATmega128 存储器

## 2.1 系统内可编程的Flash 程序存储器

- 具有128K字节的在线编程Flash

## 2.2 SRAM数据存储器（静态随机存储器）

- 还可以访问直到64K的外部数据SRAM

## 2.3 EEPROM数据存储器

- 包含4K字节的EEPROM，其寿命为100,000 次

## 2.4 I/O存储器

- 所有I/O 和外设都被放置在I/O 空间。在32个通用工作寄存器和I/O之间传输数据。

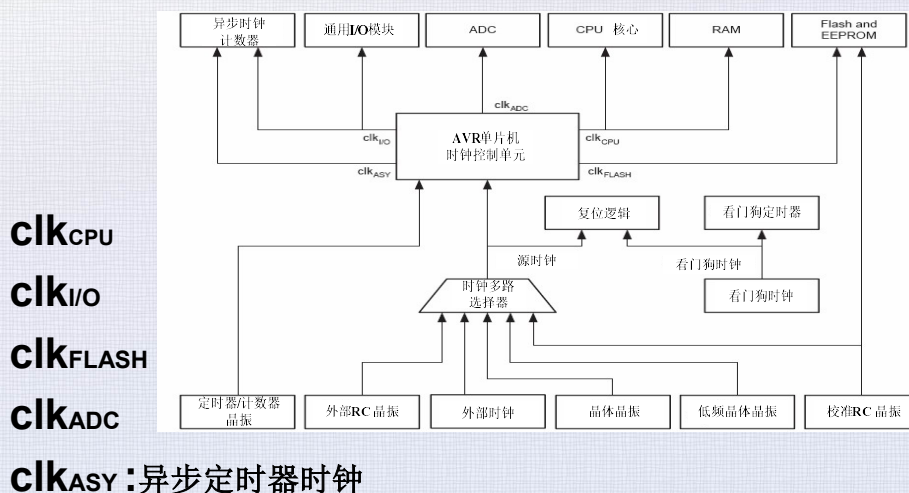
## 2.5 外部存储器接口

- 适合于与存储器器件互连，如外部SRAM和Flash， LCD， A/D， D/A， 等等。





### 3、AVR的主要时钟系统



### 4、系统控制和复位

- 复位时所有的I/O 寄存器都被设置为初始值，程序从复位向量处开始执行。
- 复位源生效时I/O 端口立即复位为初始值，不需要任何时钟的辅助。
- ATmega128 有5个复位源：
  1. 上电复位
    - 当电源电压低于上电复位门限 ( $V_{POT}$ ) 时，MCU 复位。
  2. 外部复位
    - 当引脚 RESET 上的低电平持续时间大于最小脉冲宽度时MCU 复位。
  3. 看门狗复位
    - 当看门狗使能并且看门狗定时器超时复位发生。
  4. 掉电检测复位
    - 当掉电检测复位功能使能，且电源电压低于掉电检测复位门限 ( $V_{BOT}$ ) 时MCU 即复位。
  5. JTAG AVR 复位
    - 当复位寄存器为1 时MCU即复位。



## 5、ATmega128 的中断向量1

向量号	程序地址	中断源	中断定义
1	\$0000	RESET	复位
2	\$0002	INT0	外部中断请求0
3	\$0004	INT1	外部中断请求1
4	\$0006	INT2	外部中断请求2
5	\$0008	INT3	外部中断请求3
6	\$000A	INT4	外部中断请求4
7	\$000C	INT5	外部中断请求5
8	\$000E	INT6	外部中断请求6
9	\$0010	INT7	外部中断请求7
10	\$0012	TIMER2 COMP	T/C 2 比较匹配
11	\$0014	TIMER2 OVF	T/C 2 溢出



## ATmega128 的中断向量2

向量号	程序地址	中断源	中断定义
12	\$0016	TIMER1 CAPT	T/C 1 捕捉事件
13	\$0018	TIMER1 COMPA	T/C 1 比较匹配 A
14	\$001A	TIMER1 COMPB	T/C 1 比较匹配B
15	\$001C	TIMER1 OVF	T/C 1 溢出
16	\$001E	TIMER0 COMP	T/C 0 比较匹配
17	\$0020	TIMER0 OVF	T/C 0 溢出
18	\$0022	SPI, STC	SPI 串行传输结束
19	\$0024	USART0, RX	USART0, Rx 结束
20	\$0026	USART0, UDRE	USART0 数据寄存器空
21	\$0028	USART0, TX	USART0, Tx 结束
22	\$002A	ADC	ADC 转换结束
23	\$002C	EE READY	EEPROM 就绪





## ATmega128 的中断向量3

向量号	程序地址	中断源	中断定义
24	\$002E	ANALOG COMP	模拟比较器
25	\$0030 <sup>(3)</sup>	TIMER1 COMPC	T/C 1 比较匹配C
26	\$0032 <sup>(3)</sup>	TIMER3 CAPT	T/C 3 捕捉事件
27	\$0034 <sup>(3)</sup>	TIMER3 COMPA	T/C 3 比较匹配 A
28	\$0036 <sup>(3)</sup>	TIMER3 COMPB	T/C 3 比较匹配B
29	\$0038 <sup>(3)</sup>	TIMER3 COMPC	T/C 3 比较匹配C
30	\$003A <sup>(3)</sup>	TIMER3 OVF	T/C 3 溢出
31	\$003C <sup>(3)</sup>	USART1, RX	USART1, Rx 结束
32	\$003E <sup>(3)</sup>	USART1, UDRE	USART1 数据寄存器空
33	\$0040 <sup>(3)</sup>	USART1, TX	USART1, Tx 结束
34	\$0042 <sup>(3)</sup>	TWI	两线串行接口
35	\$0044 <sup>(3)</sup>	SPM READY	保存程序存储器内容就绪



## 6、I/O端口

- 所有AVR I/O端口都具有真正的读-修改-写功能。

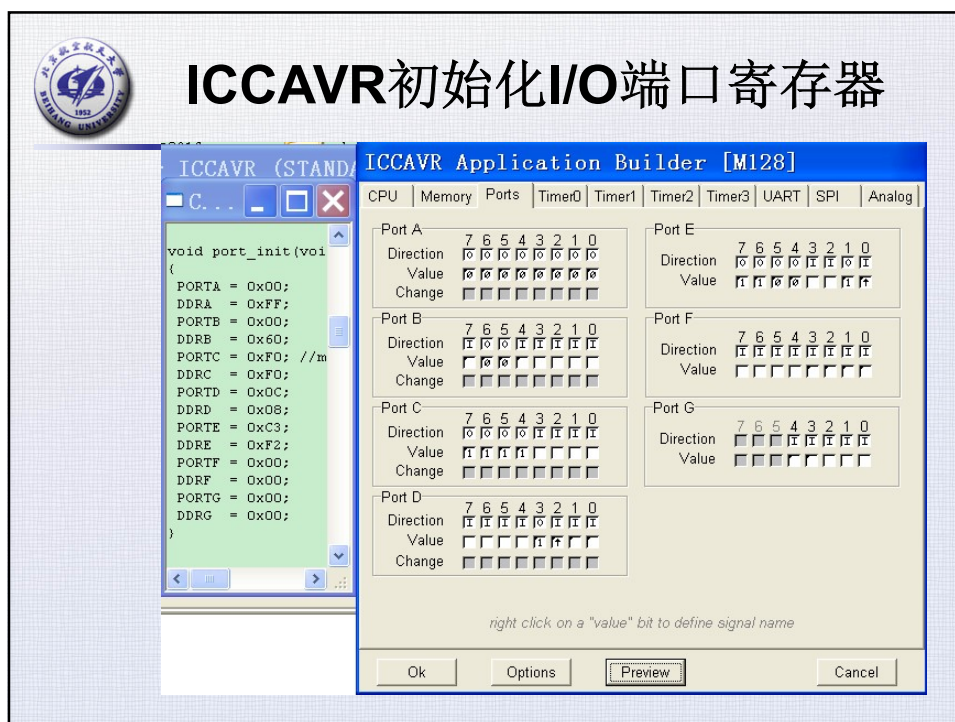
3个8位寄存器用于控制I/O端口

- 方向控制寄存器DDR<sub>x</sub> { =1输出  
=0输入
- 数据寄存器PORT<sub>x</sub> 写
- 输入引脚寄存器PIN<sub>x</sub> 只读

所有的寄存器和位以通用格式表示：  
小写的“x”表示端口序号，  
小写的“n”代表位的序号。

PORTB<sub>3</sub> 表示端口B的第3位

```
void port_init(void)
{
    PORTA = 0x00;
    DDRA  = 0xFF;
    PORTB = 0x00;
    DDRB  = 0x60;
    PORTC = 0xF0; //
    DDRC  = 0xF0;
    PORTD = 0x0C;
    DDRD  = 0x08;
    PORTE = 0xC3;
    DDRE  = 0xF2;
    PORTF = 0x00;
    DDRF  = 0x00;
    PORTG = 0x00;
    DDRG  = 0x00;
}
```



**7、定时器/计数器（T/C）**

**7.1 8位T/ C0**

**7.2 8位T/ C2**

**7.3 16位 T/ C1和T/ C3**

**7.4 定时器/计数器的预分频器**

**中断屏蔽寄存器(TIMSK)**

7	6	5	4	3	2	1	0
T/C2		T/C1				T/C0	

**扩展中断屏蔽寄存器(ETIMSK)**

7	6	5	4	3	2	1	0
				T/C3		T/C3	T/C1

**中断标志寄存器(TIFR)**

7	6	5	4	3	2	1	0
T/C2		T/C1				T/C0	

**扩展中断标志寄存器(ETIFR)**

7	6	5	4	3	2	1	0
				T/C3		T/C3	T/C1





## 7.1 8位T/ C0

■ T/C0 是一个通用的，单通道8 位定时器/计数器模块。其主要特点如下：

1. 单通道计数器
2. 比较匹配发生时，清除定时器(自动加载)
3. 无毛刺的相位修正PWM
4. 频率发生器
5. 10 位时钟预分频器
6. 溢出和比较匹配中断源(TOV0 和OCF0)
7. 允许外部32kHz 晶振作为时钟



### 7.1.1 T/ C0的工作模式

#### (1) 普通模式——计数模式

- 计数器不停累加。计到最大值后(TOP=0xFF)，返回到最小值0x00重新开始。

#### (2) CTC模式——定时器模式

- 波形发生器的频率

$$f_{OCn} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRn)}$$

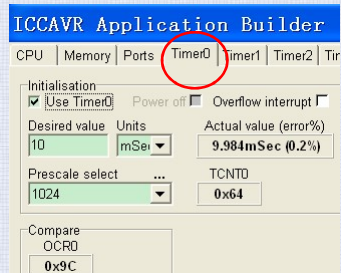
变量N代表分频因子(1、8、32、64、128、256 或1024)

#### (3) 快速PWM模式

- 产生并输出高频的PWM 波形

#### (4) 相位修正PWM模式

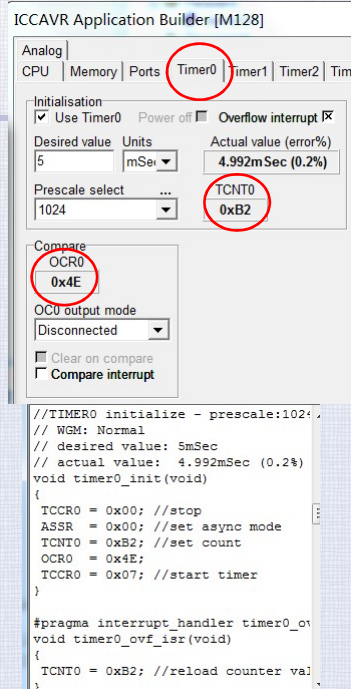
- 提供高精度相位修正PWM波形。





## 7.1.2 与T/C0相关的8位寄存器说明

- **8位控制寄存器 (TCCR0)**
  - 工作模式、匹配输出模式、时钟选择
- **8位计数寄存器 (TCNT0)**
- **8位输出比较寄存器 (OCR0)**
- **8位中断屏蔽寄存器 (TIMSK)**
  - **Bit 1 – OCIE0**: T/C0 输出比较匹配中断使能
  - **Bit 0 – TOIE0**: T/C0 溢出中断使能
- **8位中断标志寄存器 (TIFR)**
  - **Bit 1 – OCF0**: 输出比较标志0 (T/C0与OCR0的值匹配时, OCF0置位)
  - **Bit 0 – TOV0**: T/C0 溢出标志



## 7.2 8位T/ C2

- **T/C2 是一个通用单通道8 位定时/ 计数器，其主要特点如下：**
  1. 单通道计数器
  2. 比较匹配时，清零定时器 ( 自动重载)
  3. 无干扰脉冲, 相位正确的脉宽调制器 (PWM)
  4. 频率发生器
  5. 10 位时钟预分频器
  6. 溢出与比较匹配中断源(TOV2 与OCF2)
  7. 外部事件计数器





## 7.2.1 T/C2的工作模式

### (1) 普通模式——计数模式

- 计数器不停累加。计到最大值后(TOP=0xFF), 返回到最小值0x00重新开始。

### (2) CTC模式——定时器模式

- 波形发生器的频率

$$f_{OCn} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot (1 + OCRn)}$$

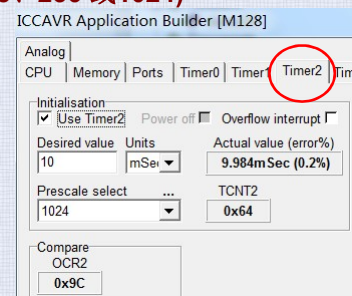
变量N代表分频因子(1、8、32、64、128、256 或1024)

### (3) 快速PWM模式

- 产生并输出高频的PWM 波形

### (4) 相位修正PWM模式

- 提供高精度相位修正PWM波形。



## 7.2.2 与T/C2相关的8位寄存器说明

### ■ 8位控制寄存器 (TCCR2)

- 工作模式、匹配输出模式、时钟选择

### ■ 8位计数寄存器 (TCNT2)

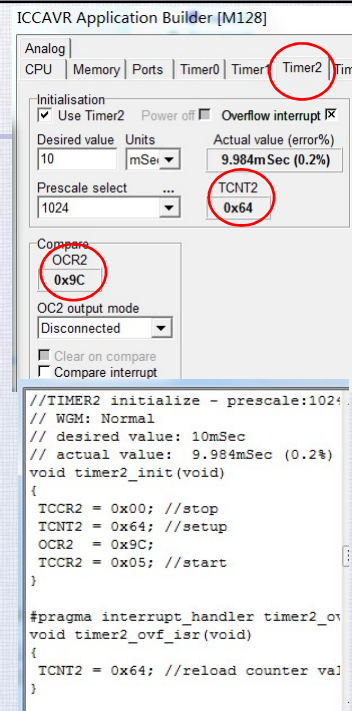
### ■ 8位输出比较寄存器 (OCR2)

### ■ 8位中断屏蔽寄存器 (TIMSK)

- Bit 7 – OCIE2: T/C2 输出比较匹配中断使能
- Bit 6 – TOIE2: T/C2 溢出中断使能

### ■ 8位中断标志寄存器 (TIFR)

- Bit 7 – OCF2: 输出比较标志0 (T/C2与OCR2的值匹配时, OCF2置位)
- Bit 6 – TOV2: T/C2 溢出标志





## 7.3 16位 T/ C1和T/ C3——实现D/A

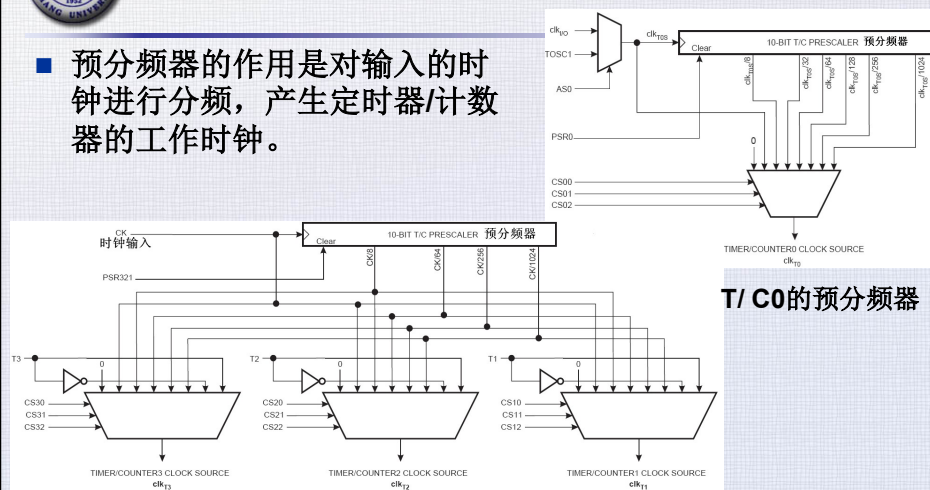
■ 16位的T/C 可以实现精确的程序定时(事件管理)、波形产生和信号测量。其主要特点如下

1. 真正的16 位设计( 即允许16 位的PWM)
2. 3 个独立的输出比较单元
3. 双缓冲的输出比较寄存器
4. 一个输入比较单元
5. 输入捕捉噪声抑制器
6. 比较匹配发生时清除寄存器( 自动重载)
7. 无毛刺的相位修正PWM，可变的PWM 周期
8. 频率发生器
9. 外部事件计数器
10. 10 个独立的中断源
  - TOV1、OCF1A、OCF1B、OCF1C、ICF1
  - TOV3、OCF3A、OCF3B、OCF3C、ICF3



## 7.4 定时器/计数器的预分频器

■ 预分频器的作用是对输入的时钟进行分频，产生定时器/计数器的工作时钟。



T/C1、T/C2和T/C3的的预分频器





## 内容

- 1、ATmega128的CPU内核
- 2、ATmega128 存储器
- 3、主要时钟系统
- 4、系统控制和复位
- 5、ATmega128 的中断向量
- 6、I/O端口
- 7、定时器/ 计数器 (T/C)
- 8、模数转换器 A/D



## 8、模数转换器 A/D

- 8.1 模数转换器特点
- 8.2 相关寄存器
- 8.3 操作（启动一次转换）
- 8.4 预分频器
- 8.5 ADC转换结果处理



## 8.1 模数转换器特点

1. **10 位逐次逼近型精度**
2. **0.5 LSB 的非线性度,  $\pm 2$  LSB 的绝对精度**
3. **13 - 260  $\mu$ s 的转换时间**
4. **最高分辨率时采样率高达15kSPS**
5. **8 路复用的单端输入通道**
6. **7 路差分输入通道**
7. **2 路可选增益为10x 与200x 的差分输入通道**
8. **可选的左对齐ADC 读数**
9. **0 - VCC 的 ADC 输入电压范围**
10. **可选的2.56V ADC 参考电压**
11. **连续转换或单次转换模式**
12. **ADC 转换结束中断**
13. **基于睡眠模式的噪声抑制器**



## 8.2 相关寄存器

- **ADC多工选择寄存器ADMUX**
- **ADC控制和状态寄存器ADCSRA**
- **ADC数据寄存器（ADCL和ADCH）**

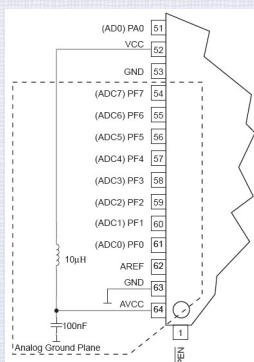


Table 97. ADC 参考电压选择

REFS1	REFS0	参考电压选择
0	0	AREF, 内部 Vref 关闭
0	1	AVCC, AREF 引脚外加滤波电容
1	0	保留
1	1	2.56V 的片内基准电压源, AREF 引脚外加滤波电容





## 8.2.1 ADC多工选择寄存器ADMUX

Bit	7	6	5	4	3	2	1	0
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
初始值	0	0	0	0	0	0	0	0

Bit7:6 – REFS1:0\_参考电压选择。

Bit5–ADLAR\_ADC 转换结果左对齐(=1)

Bits4:0 – MUX4:0\_模拟通道与增益选择位

Table 97. ADC 参考电压选择

REFS1	REFS0	参考电压选择
0	0	AREF, 内部 Vref 关闭
0	1	AVCC, AREF 引脚外加滤波电容
1	0	保留
1	1	2.56V 的片内基准电压源, AREF 引脚外加滤波电容

Table 98. 输入通道与增益选择

MUX4:0	单端输入	正差分输入	负差分输入	增益
00000	ADC0	N/A		
00001	ADC1			
00010	ADC2			
00011	ADC3			
00100	ADC4			
00101	ADC5			
00110	ADC6			
00111	ADC7			
01000		ADC0	ADC0	10x
01001		ADC1	ADC0	10x
01010		ADC0	ADC0	200x
01011		ADC1	ADC0	200x
01100		ADC2	ADC2	10x
01101		ADC3	ADC2	10x
01110		ADC2	ADC2	200x
01111		ADC3	ADC2	200x
10000		ADC0	ADC1	1x



## 8.2.2 控制和状态寄存器ADCSRA

- Bit 7 – ADEN\_ADC 使能
- Bit 6 – ADSC\_ADC 开始转换
- Bit 5 – ADFR\_ADC 连续转换选择(=1)
- Bit 4 – ADIF\_ADC 中断标志
- Bit 3 – ADIE\_ADC 中断使能
- Bits 2:0 – ADPS2:0\_ADC 预分频器选择位

Table 99. ADC 预分频选择

ADPS2	ADPS1	ADPS0	分频因子
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
读/写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	



### 8.2.3 数据寄存器（ADCL和ADCH）

**ADLAR = 0（右对齐）：**

15	14	13	12	11	10	9	8	
-	-	-	-	-	-	ADC9	ADC8	ADCH
ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0	ADCL
7	6	5	4	3	2	1	0	

**ADLAR = 1（左对齐）：**

15	14	13	12	11	10	9	8	
ADC9	ADC8	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADCH
ADC1	ADC0	-	-	-	-	-	-	ADCL
7	6	5	4	3	2	1	0	



## 8.3 操作（启动一次转换）

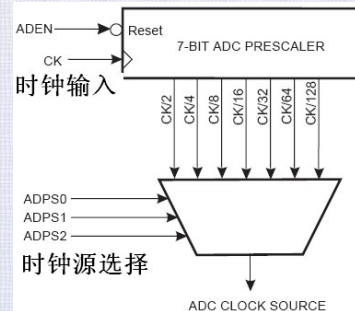
- **ADC 转换**,最小值代表**GND**，最大值代表**AREF**引脚上的电压再减去**1 LSB**。
- 设置**ADCSRA**寄存器的**ADEN**即可启动(使能)**ADC**。
- 默认情况下转换结果为右对齐，但可通过设置**ADMUX**寄存器的**ADLAR**变为左对齐。
- **ADC**转换结束可以触发中断。
- **启动一次转换**
  - 向**ADC**启动转换位**ADSC**位写"1"可以启动单次转换。在转换过程中此位保持为高，直到转换结束，然后被硬件清零。





## 8.4 预分频器

- 逐次逼近电路需要一个从**50 kHz**到**200 kHz**的输入时钟以获得最大精度。
- **ADC**模块包括一个预分频器，它可以由任何超过**100 kHz**的**CPU**时钟来产生可接受的**ADC**时钟。
- 预分频器通过**ADCSRA**寄存器的**ADPS**进行设置。



**ADC预分频器**



## 8.5 ADC转换结果处理

- 转换结束后，**ADIF**为高。
- 单次转换的结果 
$$ADC = \frac{V_{IN} * 1024}{V_{REF}}$$
  
 $V_{IN}$ 为被选中引脚的输入电压， $V_{REF}$ 为参考电压。**0x000**代表模拟地电平，**0x3FF**代表所选参考电压的数值减去**1LSB**。
- 使用差分通道的结果 
$$ADC = \frac{(V_{POS} - V_{NEG}) * GAIN * 512}{V_{REF}}$$
  
 $V_{POS}$ 为输入引脚正电压， $V_{NEG}$ 为输入引脚负电压，**GAIN**为选定的增益因子，且 $V_{REF}$ 为参考电压。结果用**2**的补码形式表示，从**0x200 (-512d)**到**0x1FF (+511d)**。



## ADC转换的实例

**ADMUX = 0xE0 ; // 1110,0000**

//片内2.56V基准源，通道为ADC0，结果左对齐。

**ADCSRA=0x87; //使能ADC，ADC分频系数是128**

**ADCSRA|=(1<<ADSC); //启动单次转换**

**While(!ADCSRA&(1<<ADIF)); //等待转换结束**

**ADCSRA|=(1<<ADIF); //清除ADIF位**

Bit	7	6	5	4	3	2	1	0	
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0	ADMUX
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	

43/45



## ADC转换的实例

**ADMUX = 0xE0 ; // 1110,0000**

//片内2.56V基准源，通道为ADC0，结果左对齐。

**ADCSRA=0x87; //使能ADC，ADC分频系数是128**

**ADCSRA|=(1<<ADSC); //启动单次转换**

**While(!ADCSRA&(1<<ADIF)); //等待转换结束**

**ADCSRA|=(1<<ADIF); //清除ADIF位**

Bit	7	6	5	4	3	2	1	0	
	ADEN	ADSC	ADFR	ADIF	ADIE	ADPS2	ADPS1	ADPS0	ADCSRA
读 / 写	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
初始值	0	0	0	0	0	0	0	0	
	ADC 使能	启动 ADC	连续转换 (=1)	转换 结束 标志	中断 使能	预分频器 $2^7=128$			