

# Analog Comparator

An analog comparator is available on pins PE2(AIN0), PE3(AIN1)

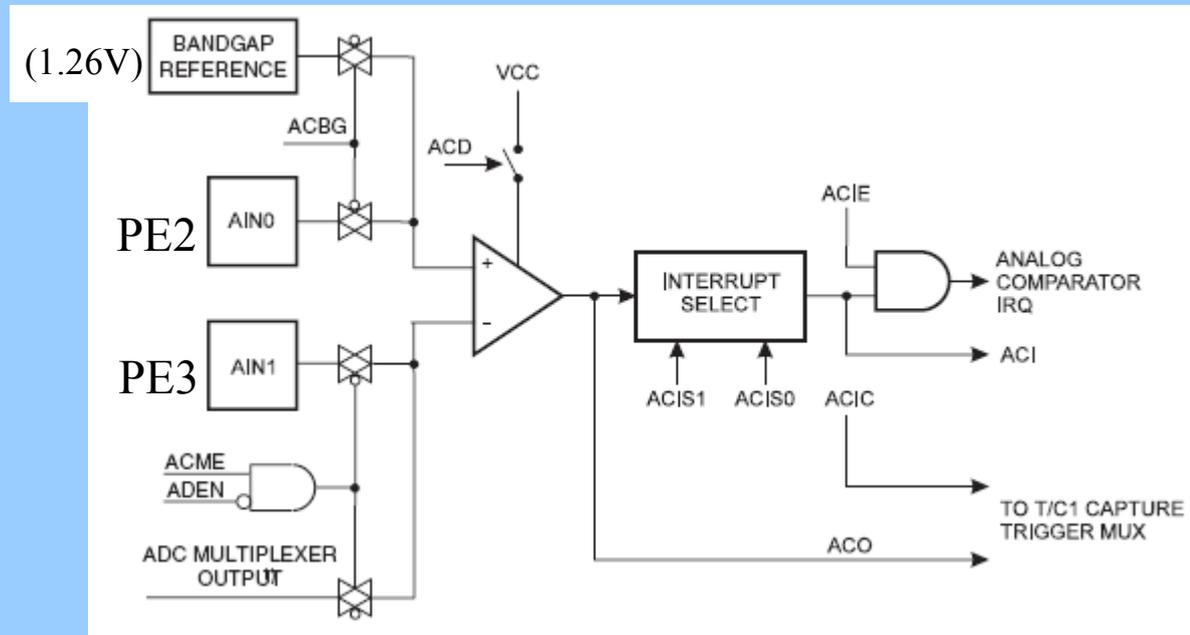
The comparator operates like any other comparator.

-when (+) exceeds (-) input, output goes logic “1”

The comparator's output change can:

-trigger the input capture register (ICR1) on TCNT1

-cause an interrupt triggered on rising AND/OR falling edges



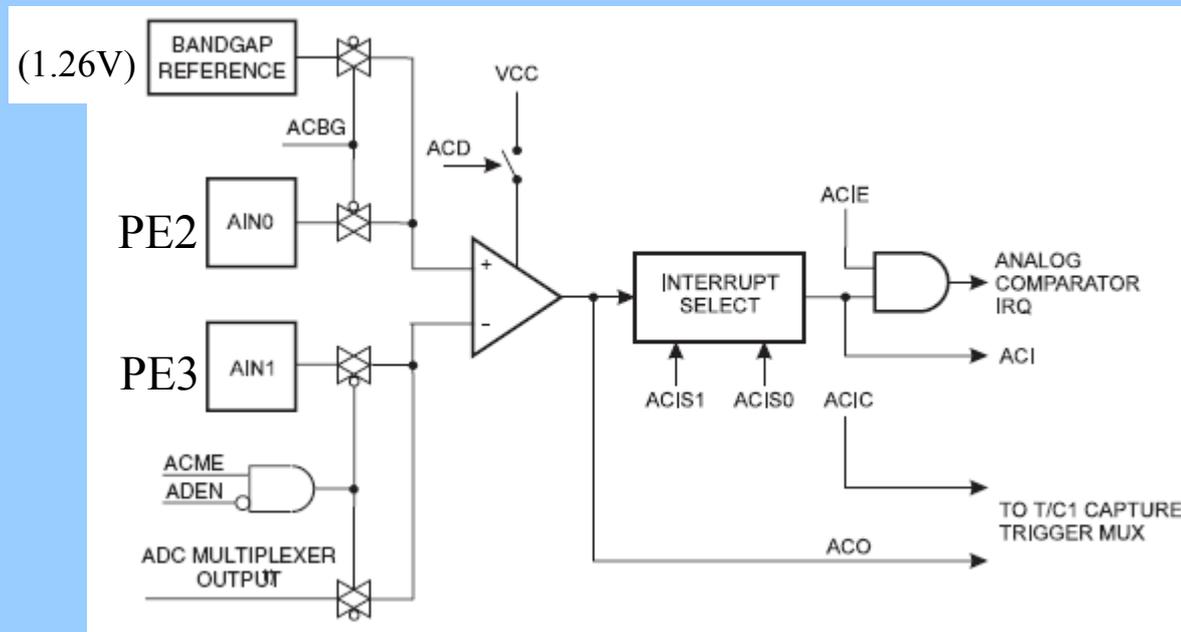
# Analog Comparator

Analog comparator controlled with two registers SFIOR and ACSR

Bit	7	6	5	4	3	2	1	0	
	TSM	-	-	-	ACME	PUD	PSR0	PSR321	SFIOR
Read/Write	R/W	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

*analog comparator mux enable:*

If ADC is off, it allows the comparator to use the ADC input mux. This would allow selecting among many inputs instead of one.



# Analog Comparator

## Analog Comparator Control and Status Register (ACSR)

**analog comparator interrupt mode select:**

- 00 : interrupt on output toggle
- 01 : reserved
- 10 : interrupt on falling edge
- 11 : interrupt on rising edge

Bit	7	6	5	4	3	2	1	0	
	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	N/A	0	0	0	0	0	

**analog comparator disable:** If set, comparator is off

**analog comparator bandgap select:** If set, bandgap voltage is applied to non-inverting comparator input

**analog comparator output:**

**analog comparator interrupt flag:** Set when comparator triggers an interrupt event. Cleared by ISR or writing a logic one to the bit.

**analog comparator interrupt enable:** If global interrupt enable is set, when comparator can trigger an interrupt.

**analog comparator input capture enable:** When set, allows comparator to trigger input capture register

# Bandgap Voltage Reference

An internal bandgap voltage reference is available on the Atmega128.

Output is typically 1.23 Volts. (quite accurate, see datasheet)

The 2.56 volt internal reference is generated from this reference.

It's used by the brownout detection circuitry and optionally by ADC or comparator.

It is only enabled when one of the below is true:

- brownout detection is enabled
- bandgap reference is connected to the analog comparator
- when the ADC is enabled

Bandgap reference takes 70uS max to startup.

- can be a power consumption issue

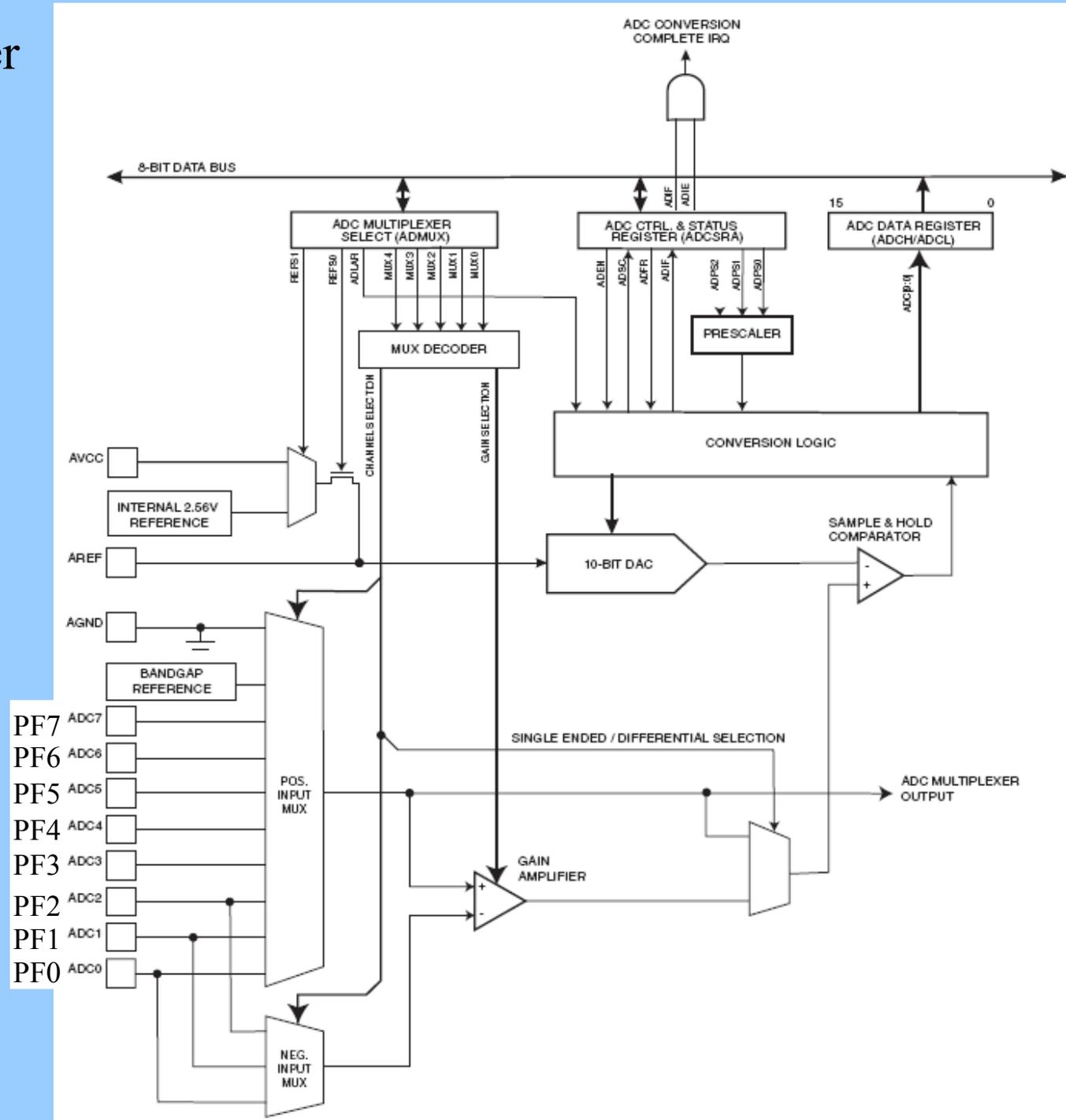
# Analog to Digital Converter (ADC)

## Analog to Digital Converter

- 10-bit successive approximation ADC
- 13-260uS conversion time
- 8 multiplexed single ended input channels
- 7 differential input channels
- 2 differential input channels with optional gain of 10x or 200x
- Ground to Vcc input voltage range
- Reference voltage of 2.56V or external AVcc
- Interrupt on conversion complete

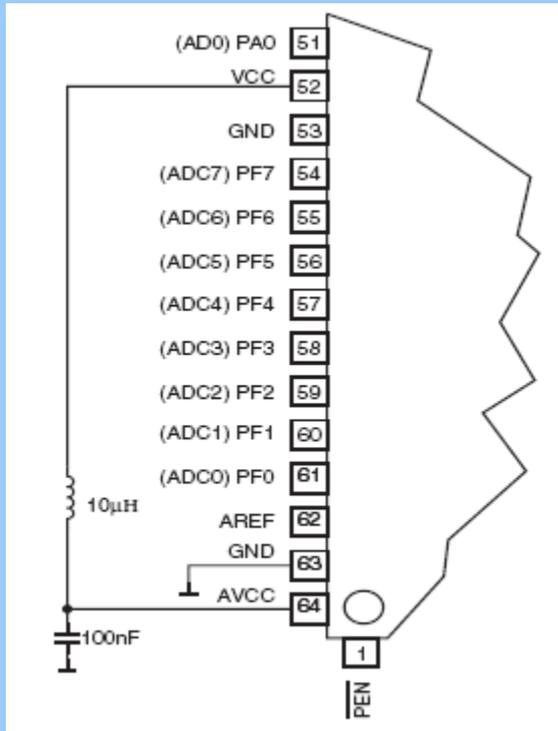
# ADC

## Analog to Digital Converter

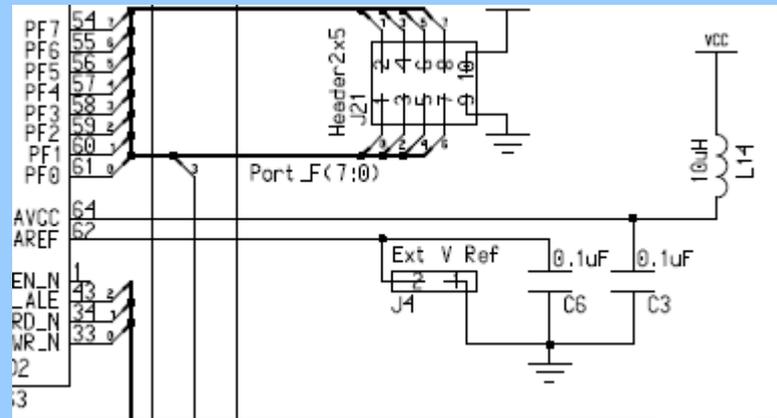


# ADC

## ADC Power Connections



Atmel suggested circuit



Circuit on mega128 board

To keep the digital noise out of the ADC circuitry, it is given its own power supply pin. This pin has extra decoupling circuitry.

AVCC can also be used as the reference voltage to the ADC.

# ADC

## ADC operation

Setup involves:

- choose a reference voltage (usually AVCC)
- select input channel via the input multiplexer
- enable interrupts if necessary
- enable ADC

The reference voltage determines

- the maximum input voltage that should be applied to the ADC input
- the smallest voltage that can be resolved by the ADC

We use AVCC (5V) for the ADC reference.

- maximum input should not exceed 5V (electrical and s/w reasons!)
- input voltage resolution is  $(5/1024) = 0.00488$  volts
- 0 volts gives ADC output of 0x0000
- 5 volts gives ADC output of 0x03FF

# ADC

## ADC operation

### Starting a conversion:

#### -Single conversion mode:

- Write logic one to Start Conversion bit (ADSC) in ADCSRA
- Bit stays one while doing conversion , clears when done

#### -Free Running mode:

- Set free running bit (ADFR) in ADCSRA
- First conversion started by writing logic one to ADSC bit
- Conversions continue regardless of state of ADC interrupt bit (ADIF) is cleared or not.

# ADC

## ADC operation (cont.)

### ADC completion:

- When ADIF bit in ADCSRA goes logic one
- ADIF is cleared by ISR or by writing a one to it.

### Getting the results:

- ADLAR bit in ADCSRA controls presentation of data
- If ADLAR=0 data is right justified in the ADC data registers
- If ADLAR=1 data is left justified

### Data registers

- ADCH and ADCL
- Accessed together as a 16-bit register called ADC (16-bit access)
- Only need 8-bits?...,
  - left justify, read ADCH, and throw away 2 bits of precision

# ADC

ADC clock input

Successive approximation conversion requires an input clock.

Clock is derived from internal CPU clock

-7-bit prescaler provides outputs at

-clk/2, clk/4, clk/8, clk/16, clk/32, clk/64, clk/128

Required clock speed is between 50khz to 200khz depending on resolution.

Normal conversion takes 13 clock cycles.

First conversion takes 25 clock cycles for analog circuitry initialization.

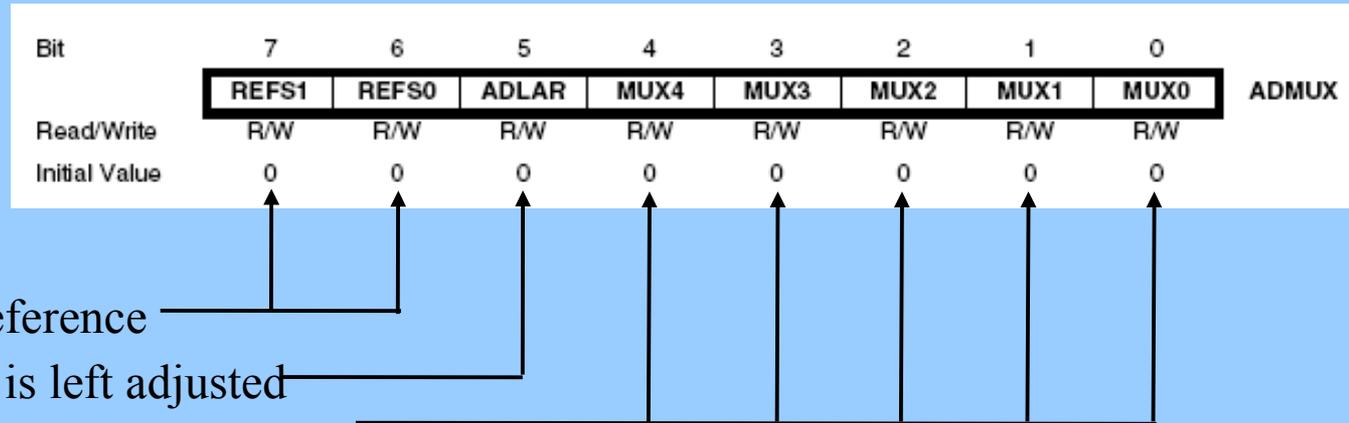
To stay under 200khz, must divide 16Mhz clock by 128, giving 125khz ADC clock.

For a 13 cycle conversion,  $13 * (1/125000) = 104\mu\text{S}$  per conversion

# ADC

## ADC registers

### -ADC Multiplexer Selection Register (ADMUX)



*reference selection*: Selects ADC reference

*ADC left adjust result*: If set, result is left adjusted

*analog channel and gain selection*:

For our project, use AVCC (5V) as the reference, REFS1,0 = 01

# ADC

## ADC registers

### -ADC Multiplexer Selection Register (ADCSRA)

Bit	7	6	5	4	3	2	1	0	ADCSRA
Read/Write	R/W								
Initial Value	0	0	0	0	0	0	0	0	

**ADC Enable:** Enables ADC operation

**ADC Start Conversion:** Set to one to start conversions

**ADC Free Running Select:** When set, operate free running

**ADC Interrupt Flag:** Set, when conversion complete

**ADC Interrupt Enable:** When set, interrupts are enabled

**ADC Prescaler Select:** select clock division ratio



# ADC

## ADC Noise Canceler

- Power-down mode to reduce CPU core digital noise
- MCU Control Register (MCUCR) controls power management
- MCUCR sleep mode bits will cause CPU to enter noise reduction mode
- Stops CPU, I/O clocks, and memory clocks

## To use:

- ADC enabled, single conversion mode, interrupts on
- Enter noise reduction mode via MCUCR register (SM2-0 = 001)
- When conversion is done, interrupt wakes up CPU and  
ISR(ADC\_vect) is executed

Note: Other interrupts can wake up CPU from noise reduction mode before the ADC conversion is completed.