COMPUTER ENGINEERING
MAHIDOL UNIVERSITY

EGCI330: Microprocessor and Interfacing

Analog Signal Handler in AVR328P

COMPUTER ENGINEERING
MAHIDOL UNIVERSITY

Outline

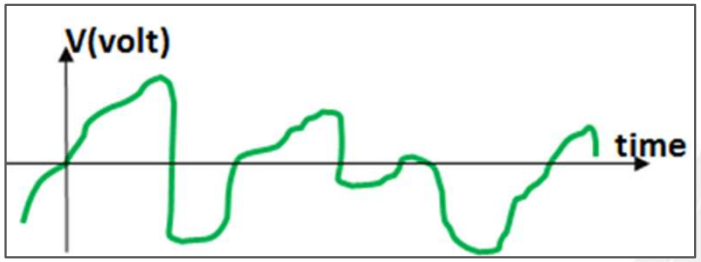
- Timer/Counter Concept
- AVR 328P Timer/Counter Interfaces
- AVR 328P Timer/Counter Registers
  - Prescaler
  - Timer0,2
  - Timer1
  - Counter

COMPUTER ENGINEERING
MAHIDOL UNIVERSITY

Analog Vs. Digital Signal

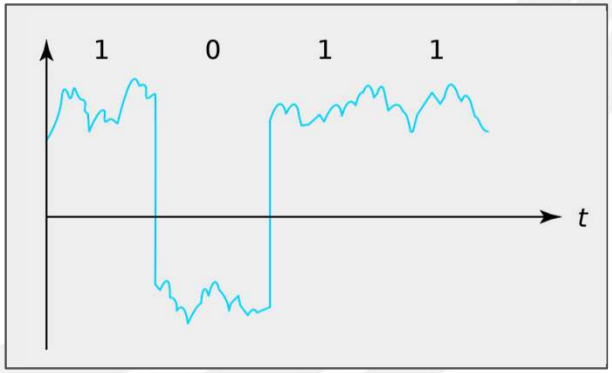
Analog Signal

Analog output is typical of most transducers and sensors. In order to use the power of digital electronics with the real world, one must convert from analog to digital and vice versa.



Digital Signal

A digital signal is a signal that represents data as a sequence of discrete values.
Simple digital signals represent information in discrete bands of analog levels which in most digital circuits, the signal can have two possible valid values.



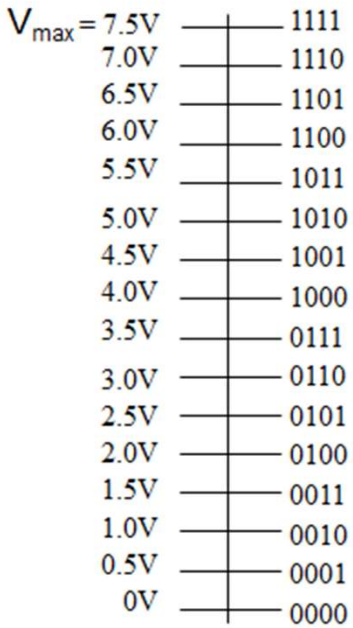
COMPUTER ENGINEERING
MAHIDOL UNIVERSITY

Analog Vs. Digital Signal

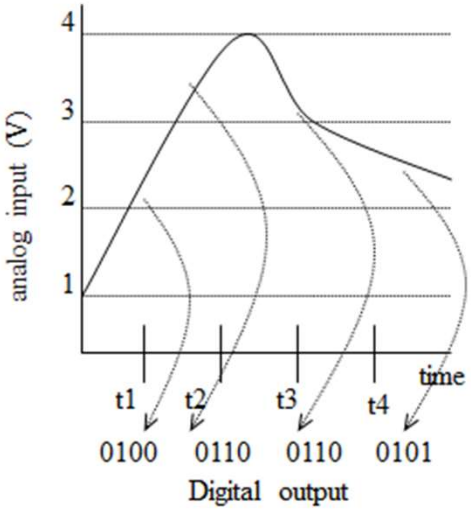
Examples of A/D Applications

- Microphones - take your voice varying pressure waves in the air and convert them into varying electrical signals
- Thermocouple – temperature measuring device converts thermal energy to electric energy
- Digital Multimeters

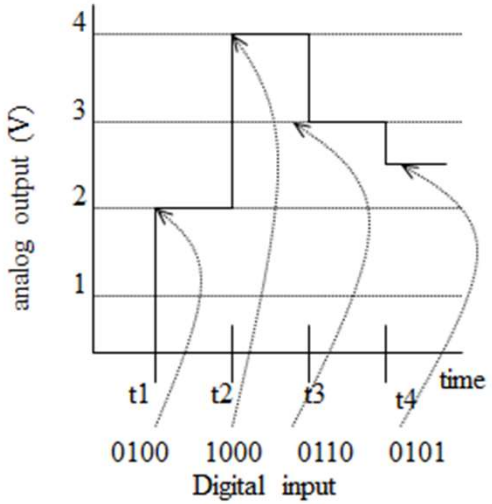
A/D and D/A converters



proportionality



analog to digital

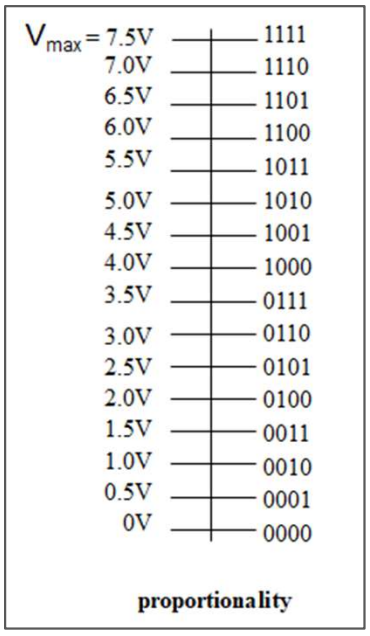


digital to analog

A/D and D/A converters

Relationship Between Analog and Digital Values

An ideal A/D converts an analog voltage to a linearly proportional digital representation.



The A/D has two sides: analog and digital

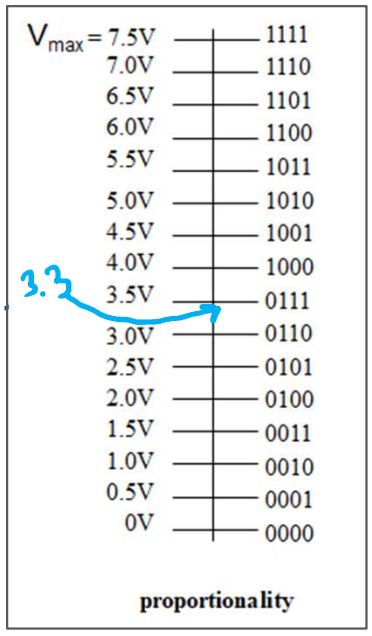
Analog	Digital
V_{min}	0..000
:	:
V_{max}	1..111

- If input voltage > Vmax the digital output is 1..111.
- Similarly if input < Vmin the digital output is 0..000.

A/D and D/A converters

Relationship Between Analog and Digital Values

Let the A/D converter has n-bit digital output and let ***A*** be Analog Value and ***D*** be the equivalent Digital Number.

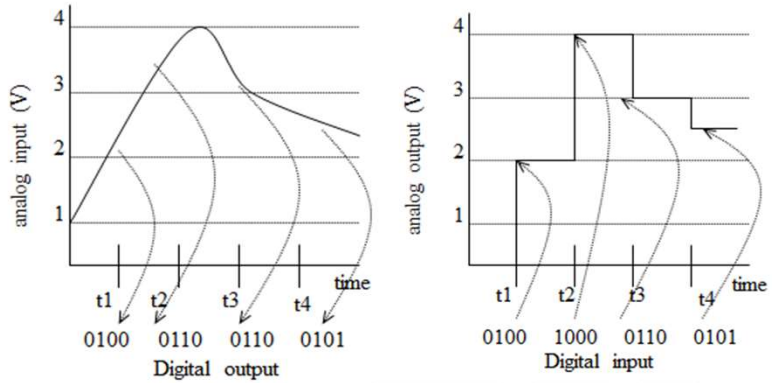


Then,

Analog	Digital
Vmin	0..000
A	D
Vmax	1..111

$$D = \frac{A - V_{min}}{V_{max} - V_{min}}(2^n - 1)$$

$$\frac{3.3 - 0}{7.5 - 0}(2^4 - 1) \approx 6.6 \approx 7 = 0111$$



COMPUTER ENGINEERING
MAHIDOL UNIVERSITY

A/D and D/A converters

Definitions:

- **Offset**: minimum analog value Vmin
- **Span (or Range)**: is the difference between maximum and minimum analog values Vmax - Vmin
- **Step Size (or Resolution, Q):** smallest analog change resulting from changing one bit in the digital number, or the analog difference between two consecutive digital numbers:

$$Q = \frac{V_{\max} - V_{\min}}{2^n - 1}$$

A/D and D/A converters

Example

Given an 4-bit A/D converter having an analog input that ranges from 0V to 7.5V. What is the resolution of this A/D converter?

$$Q = \frac{V_{\max} - V_{\min}}{2^n - 1} = \frac{7.5 - 0}{2^4 - 1} = 0.5\text{V}$$

Digital to Analog Convertor

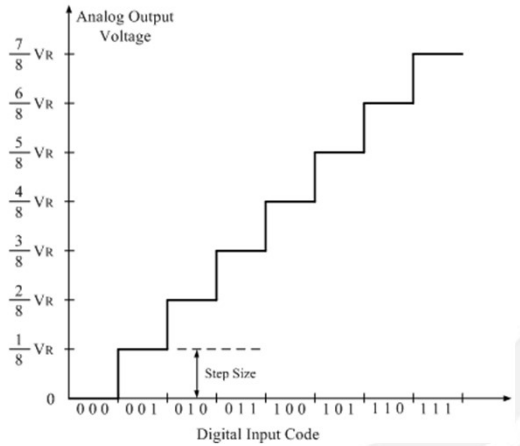
Digital to Analog Convertor (DAC)

In an electronic circuit, a combination of high voltage (+5V) and low voltage (0V) is usually used to represent a binary number. For example, a binary number 1010 is represented by

MSB LSB

Weighting	2^3	2^2	2^1	2^0
Binary Digit	1	0	1	0
State	+5V	0V	+5V	0V

+3.3V



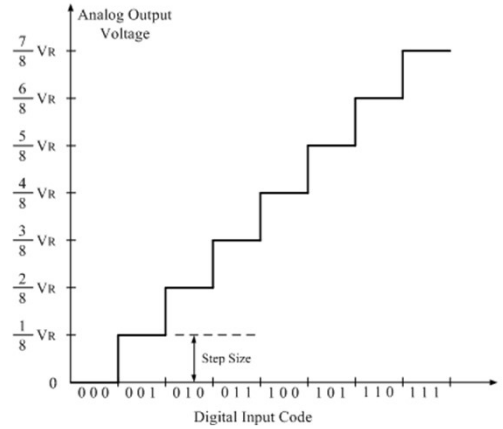
DAC are electronic circuits that convert digital, (usually binary) signals (for example, 1000100) to analog electrical quantities (usually voltage) directly related to the digitally encoded input number.

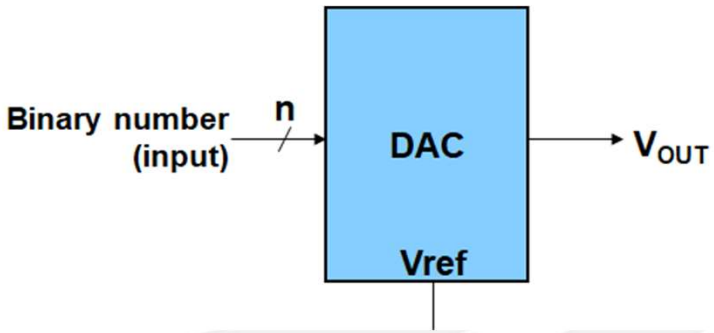
Digital to Analog Convertor

Digital to Analog Convertor (DAC)

$$\text{Step size} = \frac{V_{REF}}{\text{Num of steps}}$$

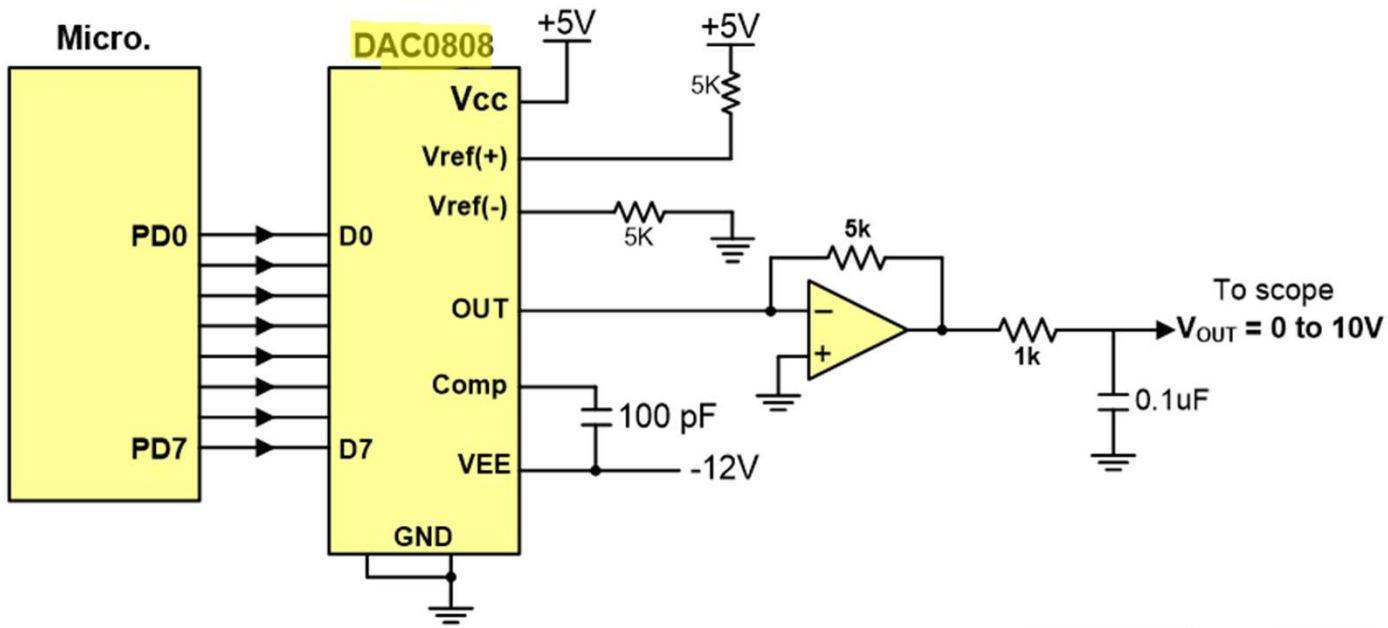
$$V_{OUT} = \text{num} \times \text{step size}$$

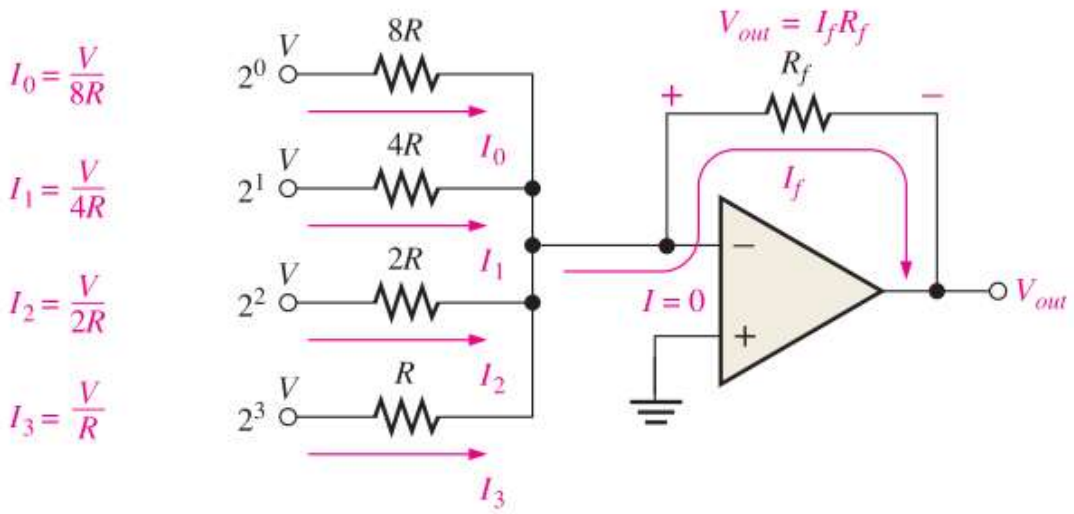




Digital to Analog Convertor

Connecting a DAC to the microcontroller

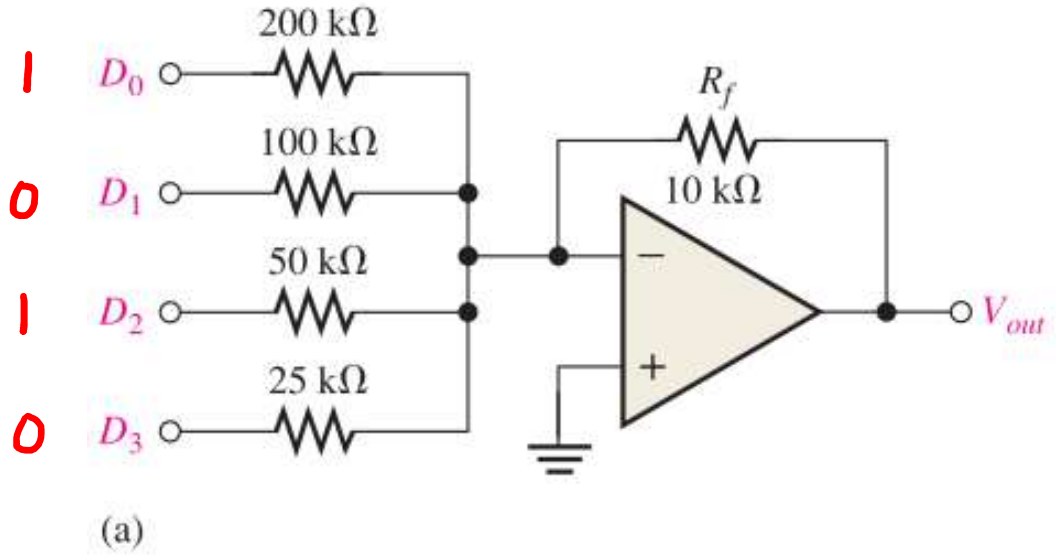




$$\frac{V}{R} + \frac{V}{2R} + \frac{V}{4R} + \frac{V}{8R} = -\frac{V_{out}}{R_f}$$

Floyd, Thomas L. *Digital Fundamentals, 10/e*. Pearson Education India, 2010.

HIGH = 5V



(a)

$$\frac{5}{50k} + \frac{5}{200k} = \frac{-V_{out}}{10k} \Rightarrow V_{out} = -1.25\,V$$

Floyd, Thomas L. *Digital Fundamentals, 10/e*. Pearson Education India, 2010.

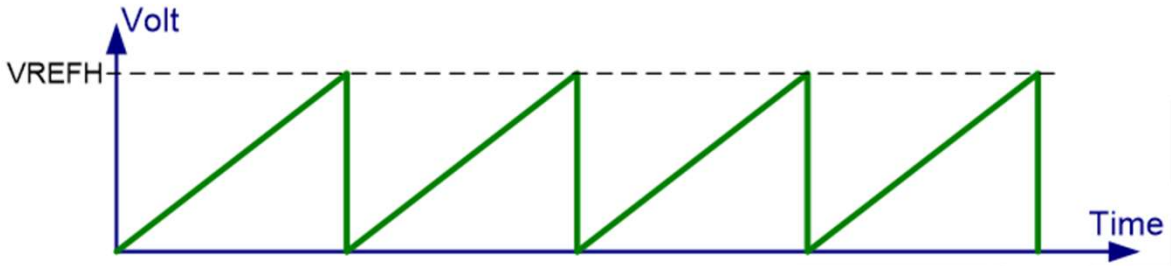
COMPUTER ENGINEERING
MAHIDOL UNIVERSITY

Digital to Analog Convertor

Generating a saw-tooth wave using DAC

```
#include <avr/io.h>

int main (void)
{
     unsigned char i = 0; //define a counter
     DDRD = 0xFF; //make Port D an output
     while (1) //do forever
     {
        PORTD = i;//copy i into PORTD to be converted
        i++;//increment the counter
     }
}
```



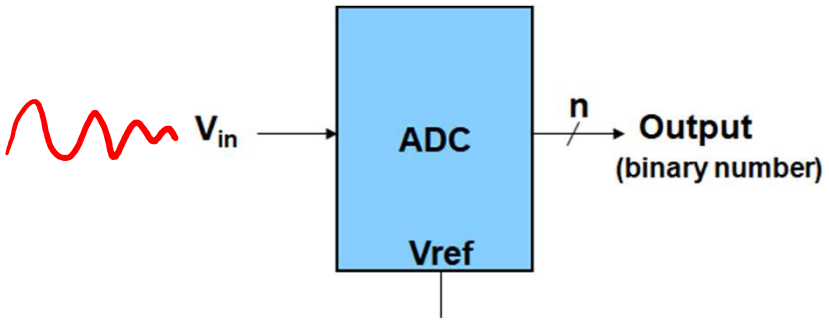
Analog to Digital Convertor

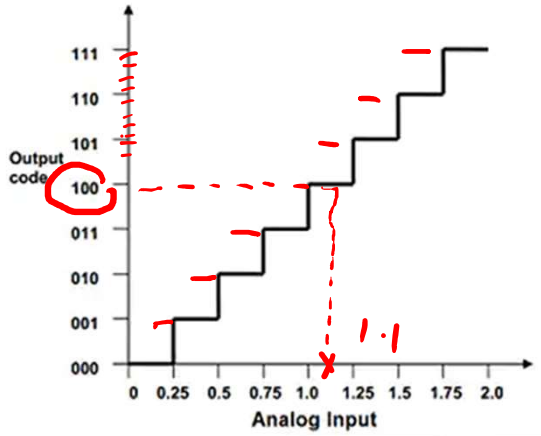
Analog to Digital Convertor (ADC)

Converts analog signals into binary words

$$stepSize = \frac{Vref}{numofsteps}$$

$$output = \left\lfloor \frac{Vin}{stepSize} \right\rfloor$$

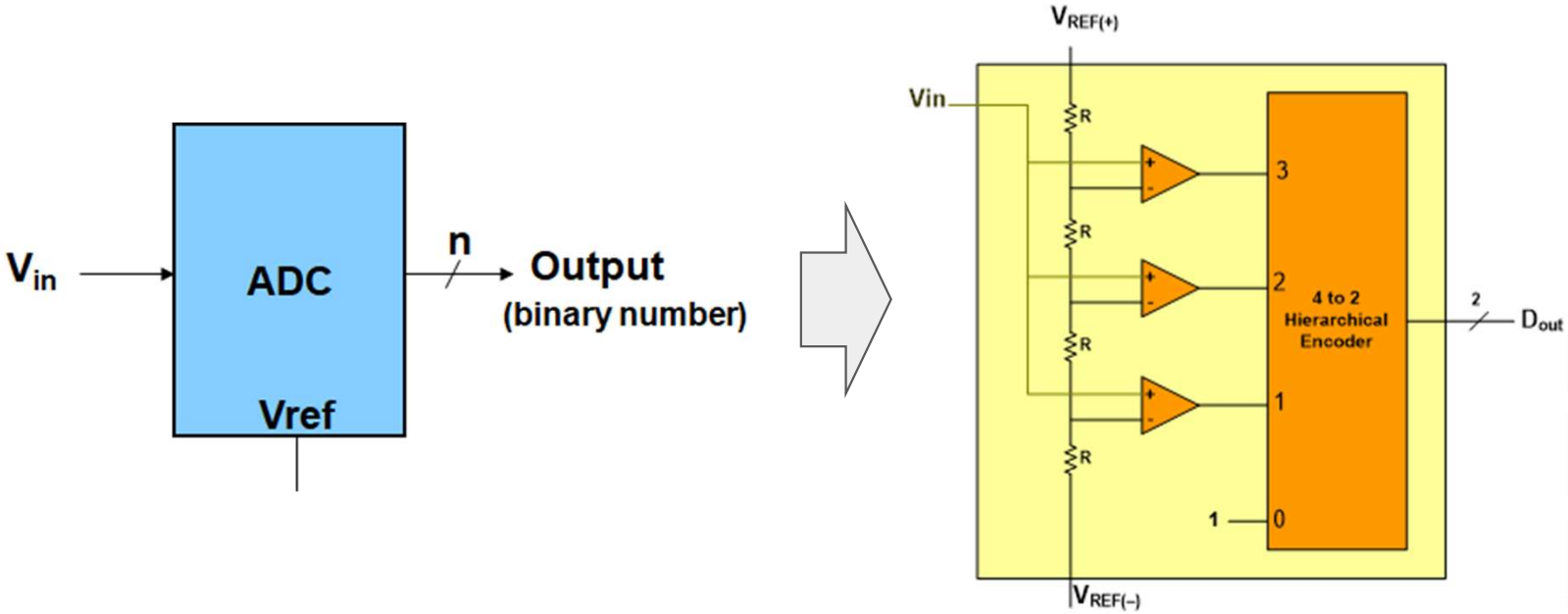




Analog to Digital Convertor

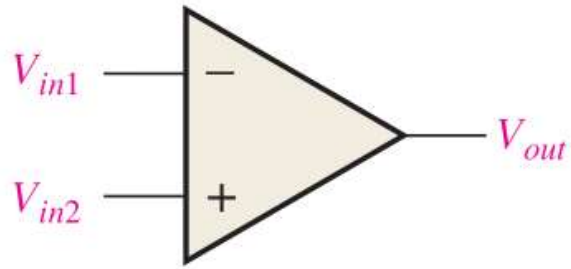
Analog to Digital Convertor (ADC)

Converts analog signals into binary words



Op amp as comparator

- $V_{out} = 1$ if $V_+ > V_-$
- $V_{out} = 0$ if $V_+ < V_-$



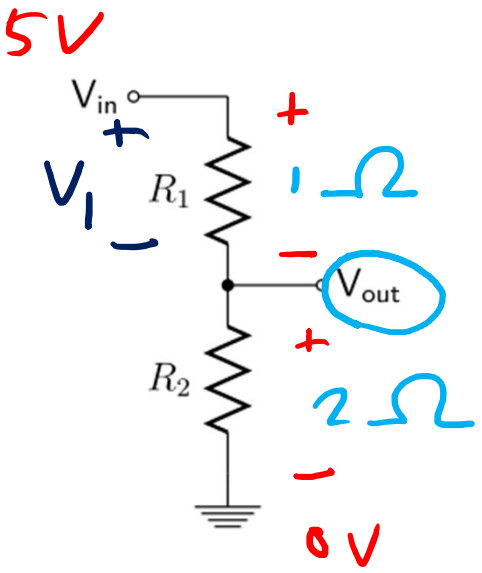
Floyd, Thomas L. *Digital Fundamentals, 10/e*. Pearson Education India, 2010.

Voltage divider

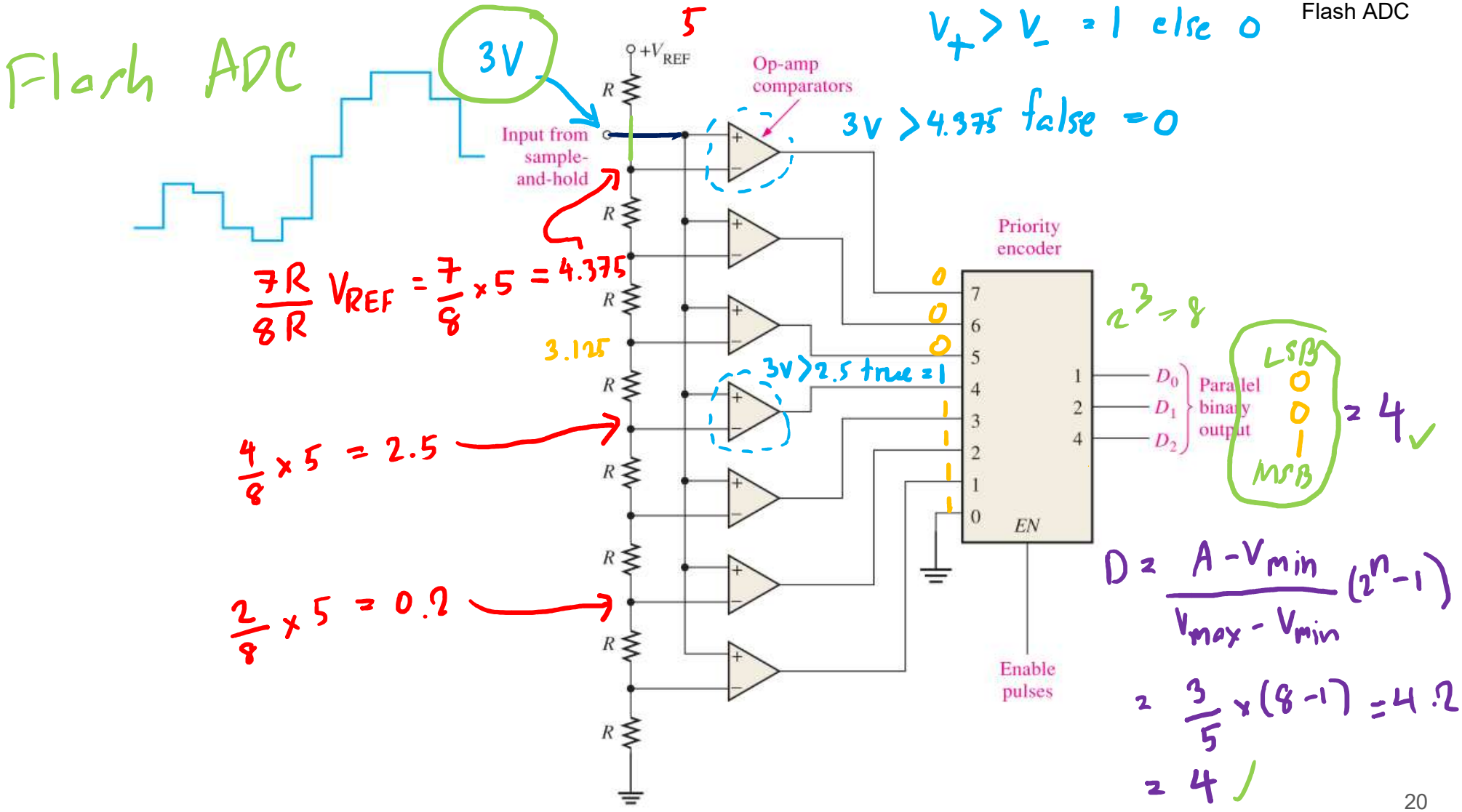
$$V_1 = \frac{R_1}{R_1 + R_2} = \frac{1\,\Omega}{3\,\Omega} \times 5$$

$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in}$$

$$\frac{2\,\Omega}{3\,\Omega} \times 5 =$$



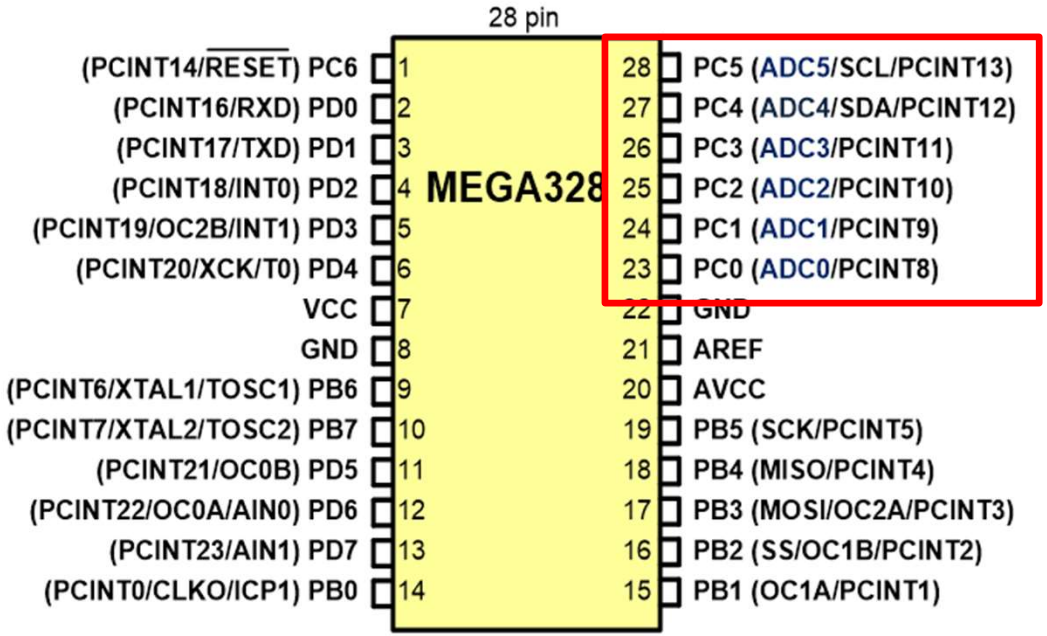
Floyd, Thomas L. *Digital Fundamentals, 10/e*. Pearson Education India, 2010.



Floyd, Thomas L. *Digital Fundamentals, 10/e*. Pearson Education India, 2010.

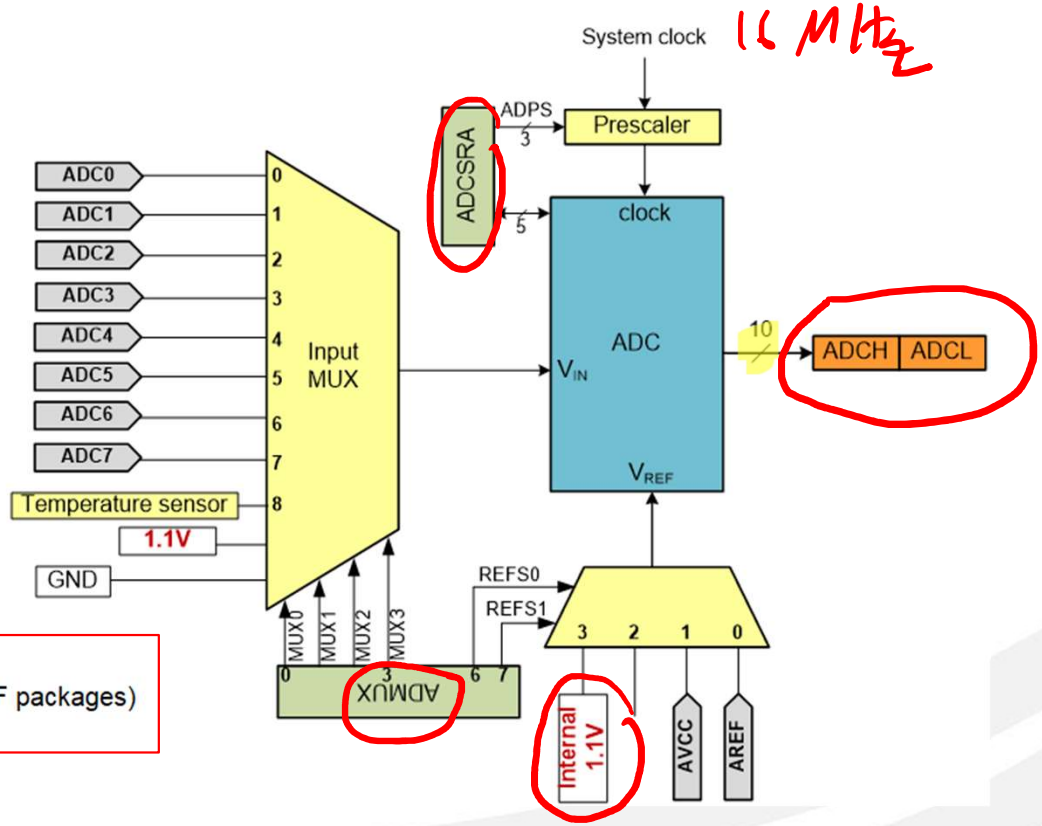
Analog to Digital Convertor

ADC in AVR328P



Analog to Digital Convertor

ADC in AVR328P



6-channel 10-bit A/D converter
(8-channels in TQFP and QFN/MLF packages)

Analog to Digital Convertor

ADMUX Register

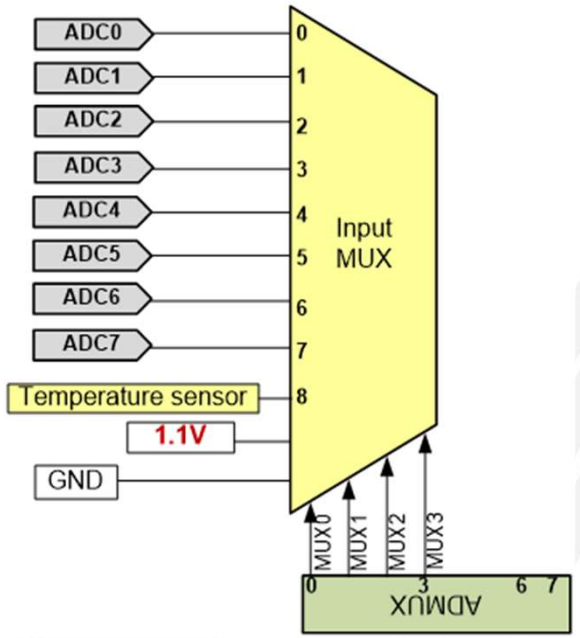
- MUX0-MUX3: input select
- ADLAR:
  - 0: right adjust the result
  - 1: left adjust the result
- REFS1-REFS0: Vref selection

Analog to Digital Convertor

ADMUX Register

D7							D0
REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0

- **MUX0-MUX3: input select**
- ADLAR:
  - 0: right adjust the result
  - 1: left adjust the result
- REFS1-REFS0: Vref selection

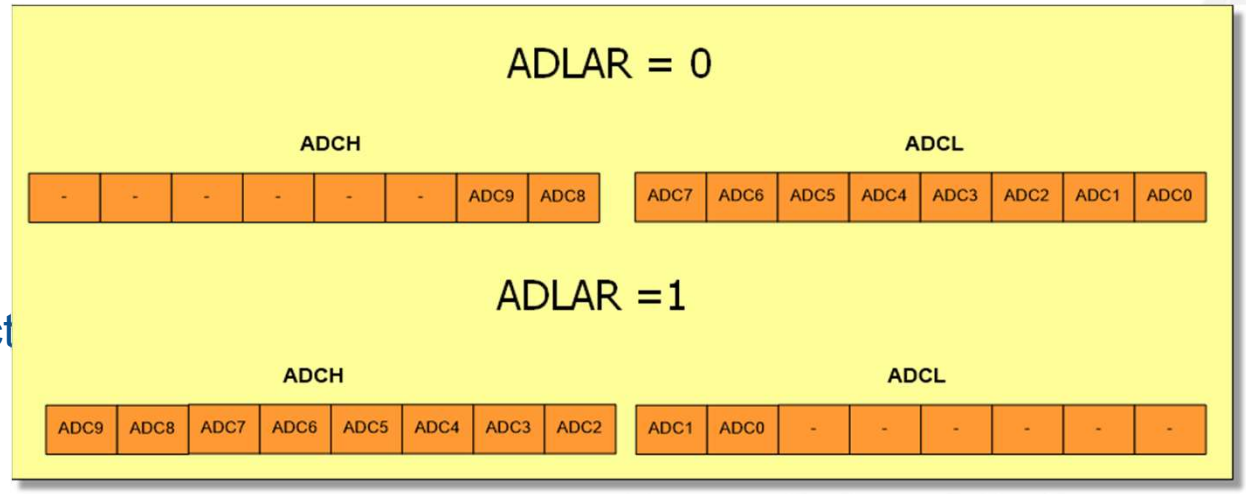


Analog to Digital Convertor

ADMUX Register

D7							D0
REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0

- MUX0-MUX1: input select
- **ADLAR:**
  - **0: right adjust the result**
  - **1: left adjust the result**
- REFS1-REFS0: Vref select

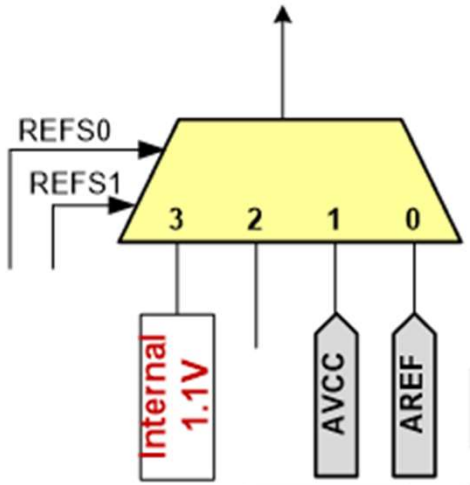


Analog to Digital Convertor

ADMUX Register

D7							D0
REFS1	REFS0	ADLAR	-	MUX3	MUX2	MUX1	MUX0

- MUX0-MUX1: input select
- ADLAR:
  - 0: right adjust the result
  - 1: left adjust the result
- **REFS1-REFS0: V-ref selection**



Analog to Digital Convertor

ADCSA Register

ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0

ADEN- Bit7 ADC Enable
This bit enables or disables the ADC. Writing this bit to one will enable and writing this bit to zero will disable the ADC even while a conversion is in progress.

ADSC- Bit6 ADC Start Conversion
To start each coversion you have to write this bit to one.

ADATE- Bit5 ADC Auto Trigger Enable
Auto Triggering of the ADC is enabled when you write this bit to one.

ADIF- Bit4 ADC Interrupt Flag
This bit is set when an ADC conversion completes and the Data Registers are updated

ADIE- Bit3 ADC Interrupt Enable
Writing this bit to one enables the ADC Conversion Complete Interrupt.

ADPS2:0- Bit2:0 ADC Prescaler Select Bits
These bits determine the division factor between the XTAL frequency and the input clock to the ADC.

Analog to Digital Convertor

ADC Prescaler

- PreScaler Bits let us change the clock frequency of ADC
- The frequency of ADC should not be more than 200 KHz
- Conversion time is longer in the first conversion

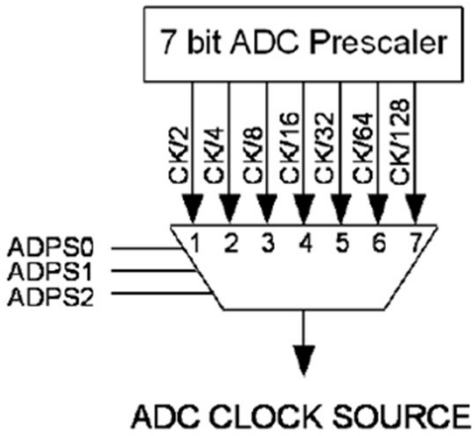


Table 13-3: V_{ref} source selection table

Condition	Sample and Hold Time (Cycles)	Conversion Time (Cycles)
First Conversion	14.5	25
Normal Conversion, Single ended	1.5	13
Normal Conversion, Differential	2	13.5
Auto trigger conversion	1.5 / 2.5	13/14

Analog to Digital Convertor

Steps in programming ADC

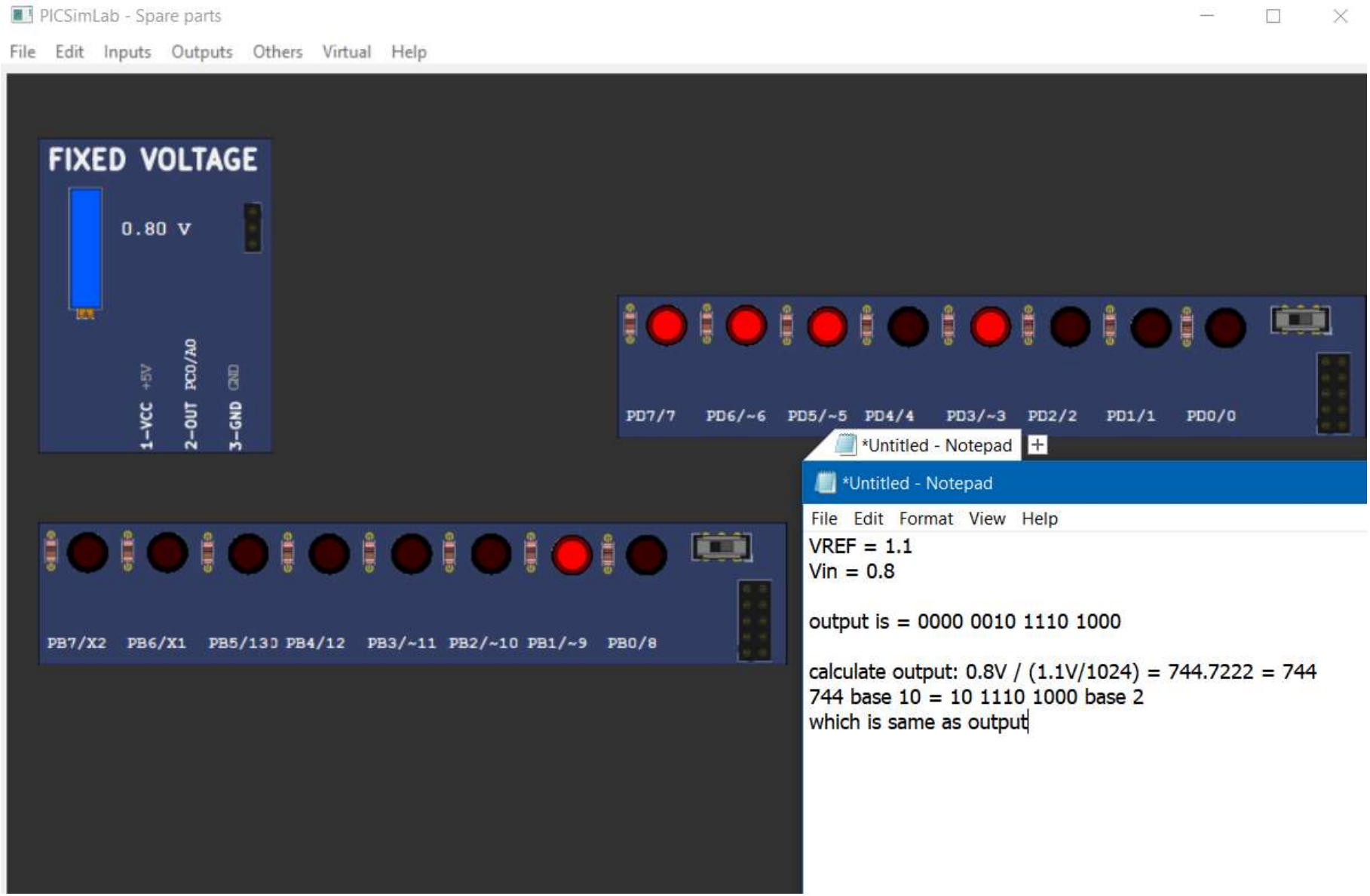
1. Make the pin for the selected ADC channel an input pin.
2. Turn on the ADC module
3. Select the conversion speed (prescalling)
4. Select voltage reference and ADC input channels.
5. Activate the start conversion bit by writing a one to the ADSC bit of ADCSRA.
6. Wait for the conversion to be completed by polling the ADIF bit in the ADCSRA register.
7. After the ADIF bit has gone HIGH, read the ADCL and ADCH registers to get the digital data output.

Analog to Digital Convertor

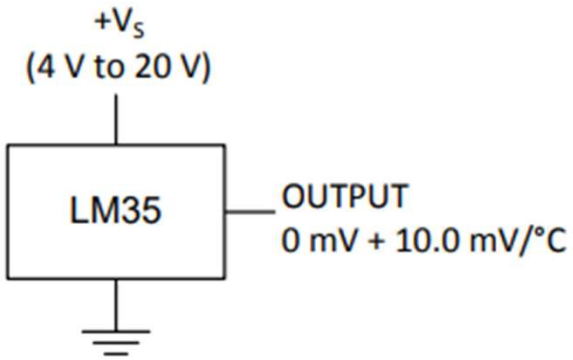
Steps in programming ADC

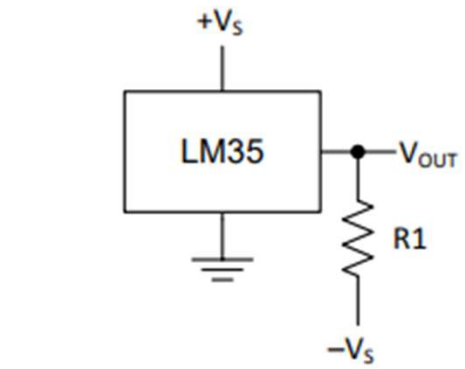
8. If you want to read the selected channel again, go back to step 5.
9. If you want to select another Vref source or input channel, go back to step 4.

Examples

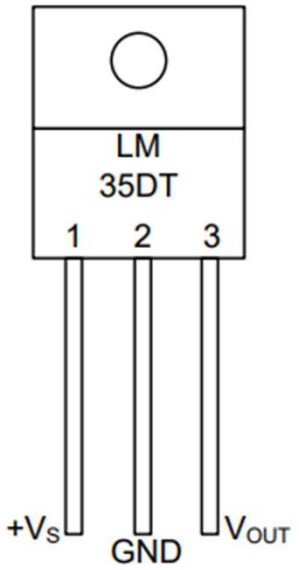

PICSimLab - Spare parts
File Edit Inputs Outputs Others Virtual Help
FIXED VOLTAGE
0.80 V
1-VCC +5V
2-OUT PCD/A0
3-GND GND
PD7/7 PD6/~6 PD5/~5 PD4/4 PD3/~3 PD2/2 PD1/1 PD0/0
PB7/X2 PB6/X1 PB5/13 PB4/12 PB3/~11 PB2/~10 PB1/~9 PB0/8
*Untitled - Notepad
*Untitled - Notepad
File Edit Format View Help
VREF = 1.1
Vin = 0.8
output is = 0000 0010 1110 1000
calculate output: 0.8V / (1.1V/1024) = 744.7222 = 744
744 base 10 = 10 1110 1000 base 2
which is same as output

LM35 temperature sensor

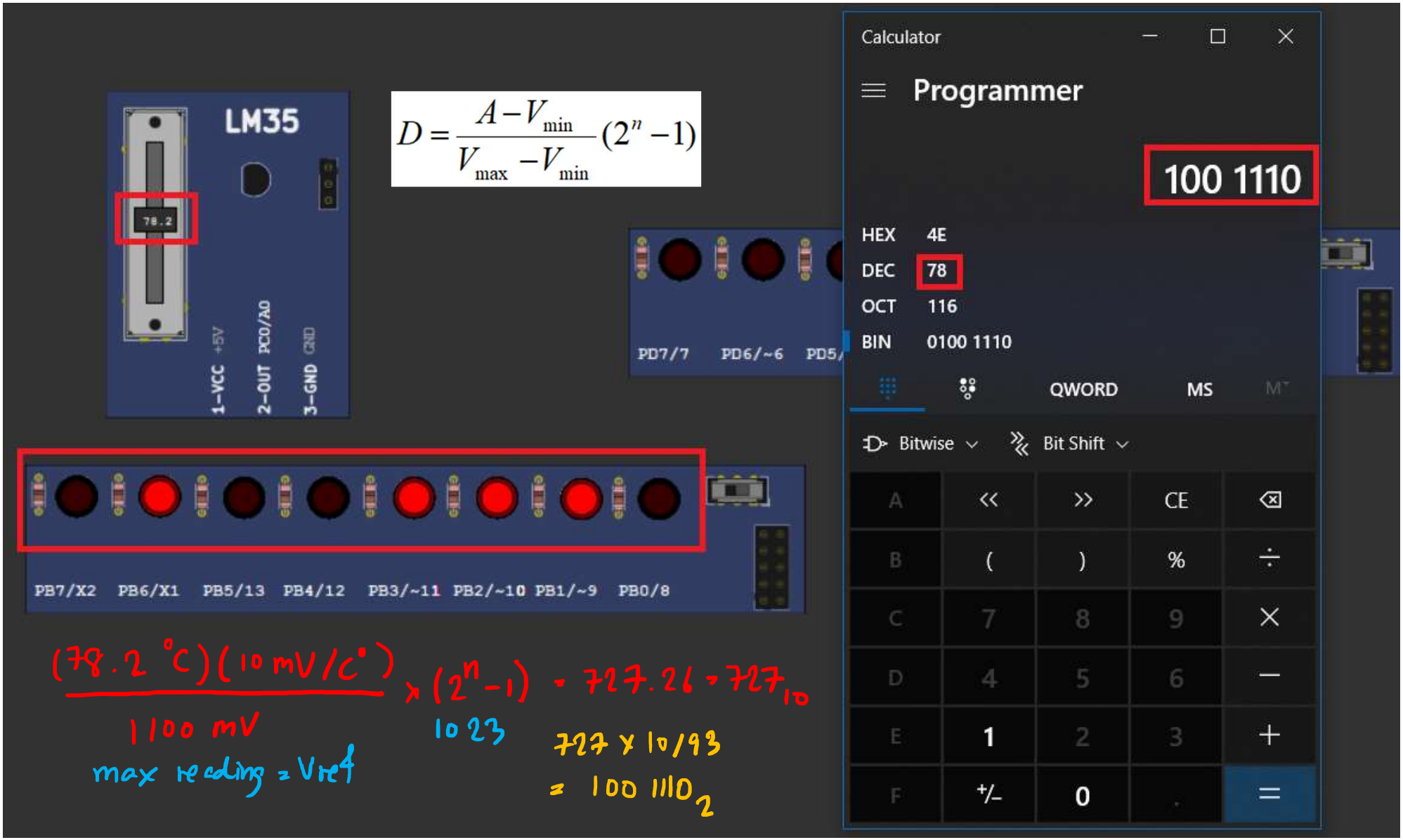


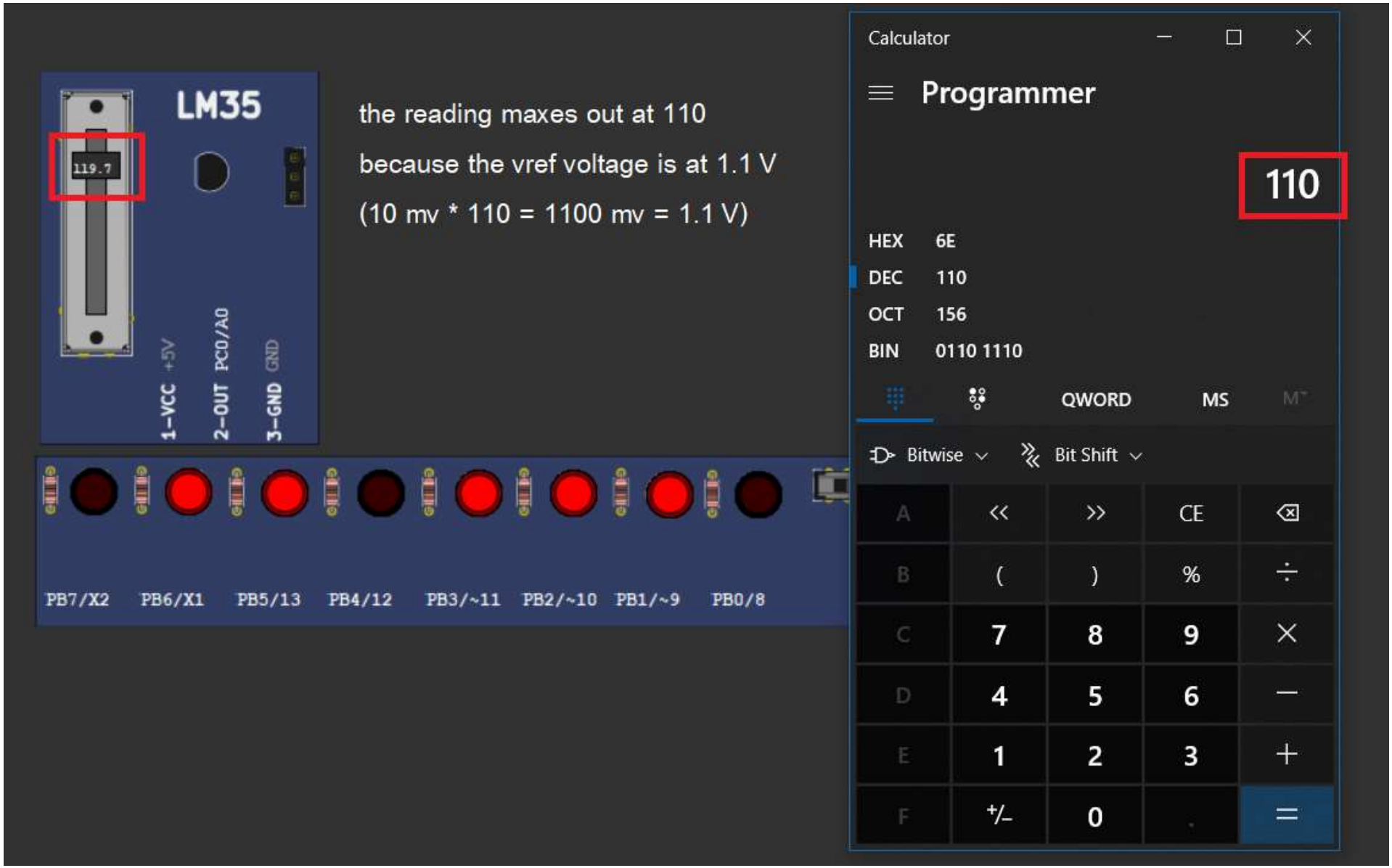


Choose R_1 = –V_S / 50 µA
V_{OUT} = 1500 mV at 150°C
V_{OUT} = 250 mV at 25°C
V_{OUT} = –550 mV at –55°C



https://www.ti.com/lit/ds/symlink/lm35.pdf


LM35
78.2
1-VCC +5V
2-OUT PC0/A0
3-GND GND
PD7/7 PD6/~6 PD5/
PB7/X2 PB6/X1 PB5/13 PB4/12 PB3/~11 PB2/~10 PB1/~9 PB0/8
Calculator
Programmer
100 1110
HEX 4E
DEC 78
OCT 116
BIN 0100 1110
QWORD MS
Bitwise Bit Shift
A << >> CE
B ( ) % ÷
C 7 8 9 ×
D 4 5 6 −
E 1 2 3 +
F +/- 0 . =
$D = \frac{A - V_{min}}{V_{max} - V_{min}}(2^n - 1)$
$\frac{(78.2\ ^\circ C)(10\ mV/C^\circ)}{1100\ mV} \times (2^n - 1) = 727.26 \approx 727_{10}$
max reading = Vref
1023
727 x 1023
$= 100\ 1110_2$


LM35
119.7
1-VCC +5V
2-OUT PC0/A0
3-GND GND
PB7/X2 PB6/X1 PB5/13 PB4/12 PB3/~11 PB2/~10 PB1/~9 PB0/8
the reading maxes out at 110
because the vref voltage is at 1.1 V
(10 mv * 110 = 1100 mv = 1.1 V)
Calculator
Programmer
110
HEX 6E
DEC 110
OCT 156
BIN 0110 1110
QWORD
MS
Bitwise
Bit Shift
CE

Exercise 12

- Modify the LM35 example to use interrupt instead of polling
- Expected output is the same as the slides 34-35, but do not use polling in the code.

Analog to Digital Convertor

Steps in programming ADC

This program gets data from channel 0 (ADC0) of ;ADC and displays the result on Port B and Port D.

```
#include <avr/io.h>
#define F_CPU 16000000UL
#include <util/delay.h>

int main (void)
{
 DDRB = 0xFF;//make Port B an output
 DDRD = 0xFF; //make Port D an output

 ADCSRA= 0x87;//make ADC enable and select ck/128
 ADMUX= 0xC8;//1.1V Vref, temp. sensor, right-justified

 while(1)
{
  ADCSRA |= (1<<ADSC);//start conversion

  while((ADCSRA&(1<<ADIF))==0);//wait for conversion to finish

  ADCSRA |= (1<<ADIF);

  PORTD = ADCL;//give the low byte to PORTD
  PORTB = ADCH;//give the high byte to PORTB

  _delay_ms(100);
 }
}
```

COMPUTER ENGINEERING
MAHIDOL UNIVERSITY

Analog Comparator

ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	TCCR1B

ICNC1: Input Capture Noise Canceller
0:disabled
1:Enabled (captures after 4 successive equal valued samples)

ICSES1: Input Capture Edge Select
0: Falling edge
1: Rising edge

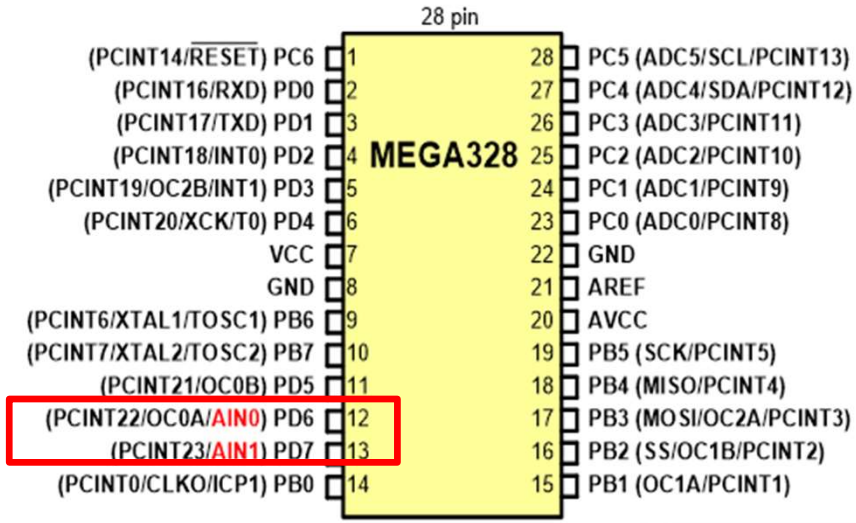
7	6	5	4	3	2	1	0	
ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	

ACIC: Analog Comparator Input Capture Enable
0: ICP1 provides the capture signal
1: analog comparator is connected to the capturer

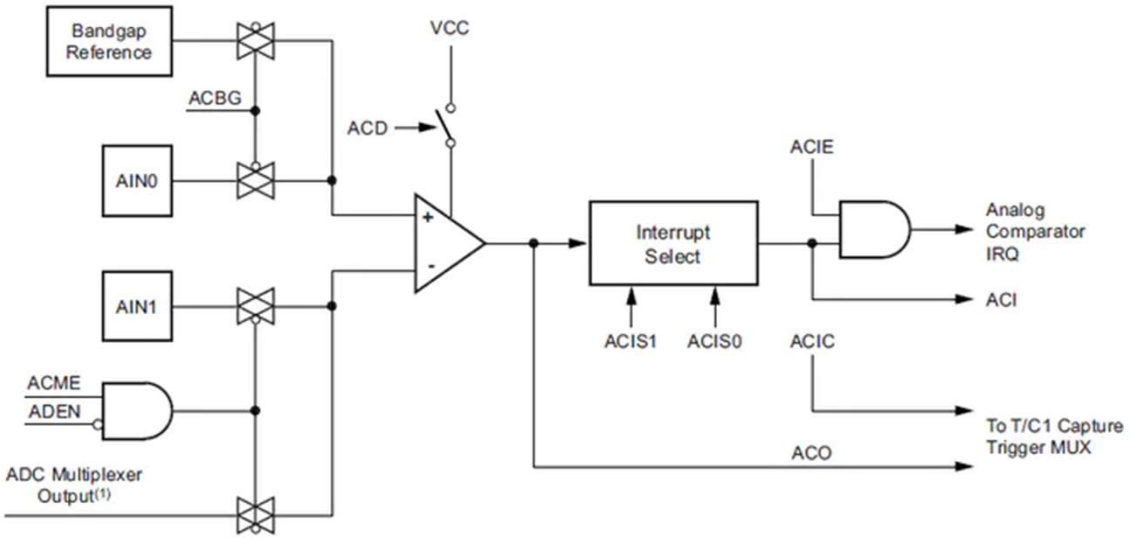
Analog Comparator

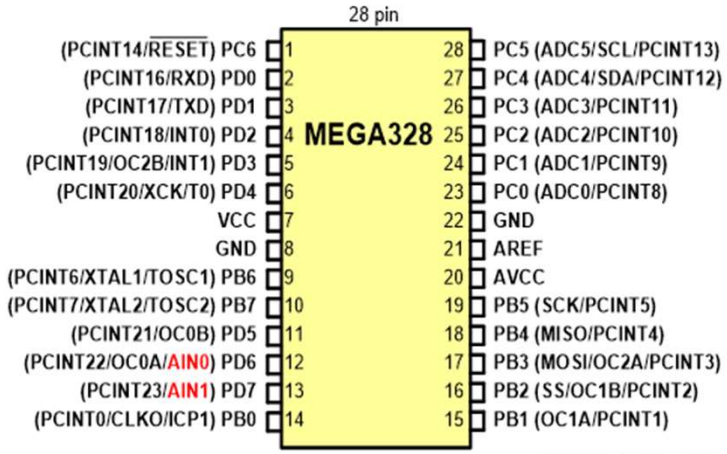
The analog comparator compares the input values on the positive pin AIN0 and negative pin AIN1.

When the voltage on the positive pin AIN0 is higher than the voltage on the negative pin AIN1, the analog comparator output, ACO, is set

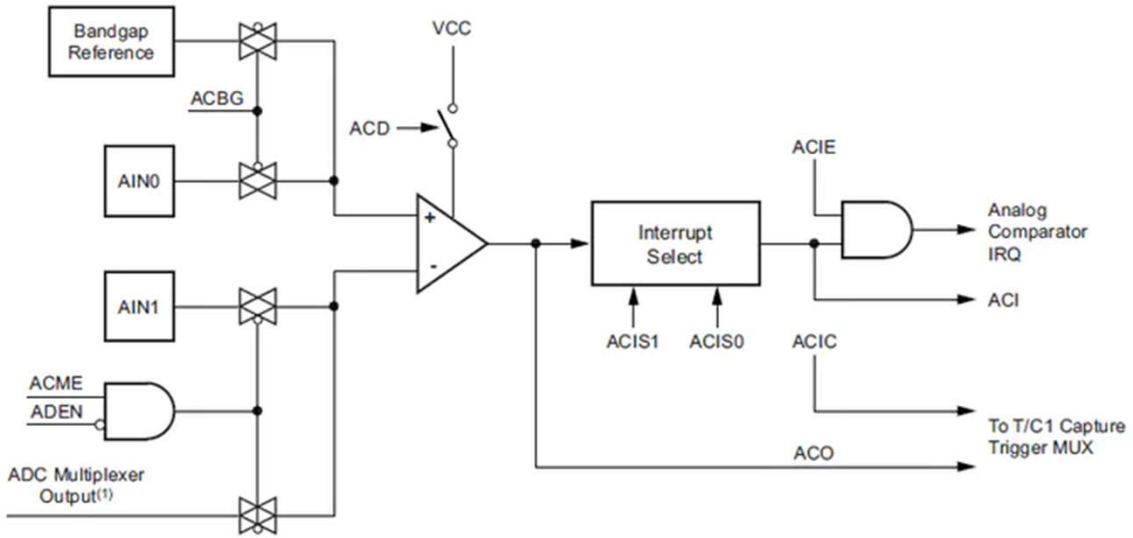


Analog Comparator





Analog Comparator



The comparator's output can be set to trigger the Timer/Counter1 input capture function.

In addition, the comparator can trigger a separate interrupt, exclusive to the analog comparator. The user can select interrupt triggering on comparator output rise, fall, or toggle."

COMPUTER ENGINEERING
MAHIDOL UNIVERSITY

Analog Comparator

ACSR – Analog Comparator Control and Status Register

Bit	7	6	5	4	3	2	1	0	
0x30 (0x50)	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	

- ACD - Analog Comparator Disable: when this bit is set by writing 1, the analog comparator is switched off.
- ACBG - Analog Comparator Bandgap Select: When this bit is set, a fixed bandgap reference voltage replaces the positive input to the analog comparator. When this bit is cleared, AIN0 is applied to the positive input of the analog comparator.
- ACO - Analog Comparator Output: The output of the analog comparator is synchronized and then directly connected to ACO. The synchronization introduces a delay of 1 - 2 clock cycles.

COMPUTER ENGINEERING
MAHIDOL UNIVERSITY

Analog Comparator

ACSR – Analog Comparator Control and Status Register

Bit	7	6	5	4	3	2	1	0	
0x30 (0x50)	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	

- ACI - Analog Comparator Interrupt Flag: This bit is set by hardware when a comparator output event triggers the interrupt mode defined by ACIS1 and ACIS0. The analog comparator interrupt routine is executed if the ACIE bit is set and the I-bit in SREG is set. ACI is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logic one to the flag.
- ACIE - Analog Comparator Interrupt Enable: When the ACIE bit is written logic one and the I-bit in the status register is set, the analog comparator interrupt is activated. When written logic zero, the interrupt is disabled.
- ACIC - Analog Comparator Input Capture Enable: When written logic one, this bit enables the input capture function in Timer/Counter1 to be triggered by the analog comparator.

Analog Comparator

ACSR – Analog Comparator Control and Status Register

Bit	7	6	5	4	3	2	1	0	
0x30 (0x50)	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	

Table 22-2. ACIS1/ACIS0 Settings

ACIS1	ACIS0	Interrupt Mode
0	0	Comparator interrupt on output toggle.
0	1	Reserved
1	0	Comparator interrupt on falling output edge.
1	1	Comparator interrupt on rising output edge.

Analog Comparator

ADCSRB – ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
(0x7B)	–	ACME	–	–	–	ADTS2	ADTS1	ADTS0	ADCSRB
Read/Write	R	R/W	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- ACME - Analog Comparator Multiplexer Enable:
  When this bit is written logic one and the ADC is switched off (ADEN in ADCSRA is zero), the ADC multiplexer selects the negative input to the Analog Comparator. When this bit is written logic zero, AIN1 is applied to the negative input of the Analog Comparator.

ACME	ADEN	MUX2..0	Analog Comparator Negative Input
0	x	xxx	AIN1
1	1	xxx	AIN1
1	0	000	ADC0
1	0	001	ADC1
1	0	010	ADC2
1	0	011	ADC3
1	0	100	ADC4
1	0	101	ADC5
1	0	110	ADC6
1	0	111	ADC7

Conclusion

Q/A