ADC / DAC

ECET 209 – Introduction to Microcontrollers

Overview

- Review Analog and Digital Converters
- Preview Lab #9
- Do a Group Activity

Digital vs. Analog

- Most values are analog in nature
  - Temperature
  - Speed
  - Position
  - Etc

- Digital systems are used to process information

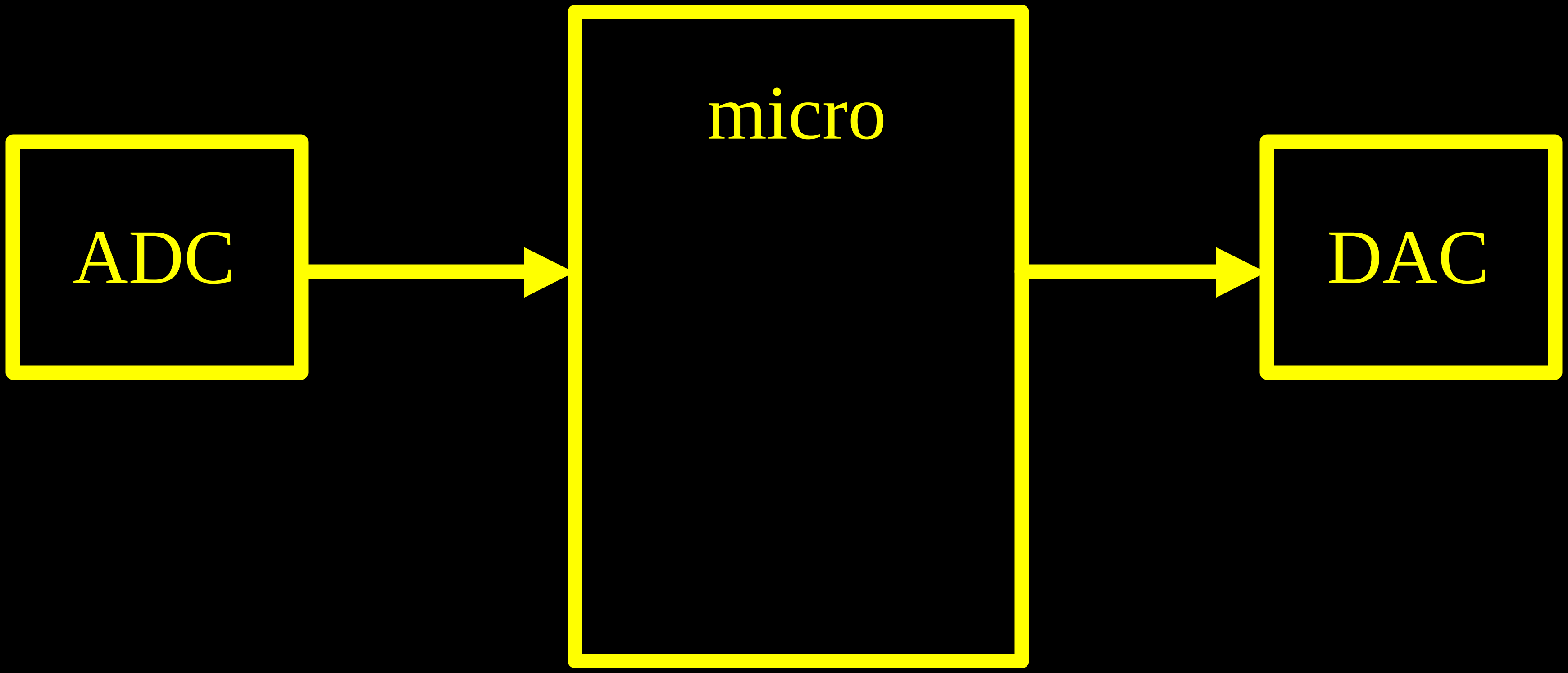
Digital vs. Analog

- Digital systems are used to process info
  - Somehow need to convert an analog value into a digital value
  - Perform digital operations on the data
  - Somehow need to convert the digital results back into an analog quantity

Digital vs. Analog

- Digital systems are used to process info
- A couple Major Examples
  - CD’s
  - MP3’s
  - DVD’s

Digital vs. Analog



ADC & DAC

- ADC – Analog to Digital Converter
- DAC – Digital to Analog Converter

- ADC and DAC operations are required at the input and output from a digital processing system that deals with analog quantities.

Digital Displays

- Digital displays are easier to read (by consumers)

- Accuracy of digital displays depends on the accuracy of the A/D conversion

Analog/Digital Characteristics

- Performance Criteria
  - Resolution
  - Sampling Rate
  - Speed
  - Linearity

Analog Interface Module

- Lab Hardware board for Analog applications
  - 8-bit Digital to Analog converter (AD558)
  - 10-bit Analog to Digital converter (AD573)
  - 0 to 10 volts
  - Pot to adjust the input analog voltage
  - Analog meter to display the analog input or output
  - Selector switches for input and the meter

Digital to Analog Converter

- Takes 8-bit value and converts it to an analog voltage
- A digital value of 0x00 yields 0 volts out
- A digital value of 0xFF yields 10 volts out

Digital to Analog Converter

- Takes 8-bit value and converts it to an analog voltage
- A digital value of 0x00 yields 0 volts out
- A digital value of 0xFF yields 10 volts out

resolution = Vout_max / 2^^{8} - 1

Digital to Analog Converter

- Takes 8-bit value and converts it to an analog voltage
- A digital value of 0x00 yields 0 volts out
- A digital value of 0xFF yields 10 volts out

resolution = Vout_max / 2^^{8} - 1

resolution = 10 volts / 255

Digital to Analog Converter

- Takes 8-bit value and converts it to an analog voltage
- A digital value of 0x00 yields 0 volts out
- A digital value of 0xFF yields 10 volts out

$$\text{resolution} = \text{Vout\_max} / 2\wedge^{8} - 1$$

$$0.0392157 = 10 \text{ volts} / 255$$

Digital to Analog Converter

- Takes 8-bit value and converts it to an analog voltage
- A digital value of 0x00 yields 0 volts out
- A digital value of 0xFF yields 10 volts out

resolution = Vout_max / 2^^{8} - 1

0.0392157 = 10 volts / 255

Vout = resolution * digital input

Analog to Digital Converter

- Takes an analog voltage and converts it to a digital value
- The AD573 is a 10-bit converter

- An analog value of 0 volts will produce a digital output of 0x000
- An analog value of 10 volts will produce a digital output of 0x3FF

Analog to Digital Converter

- Resolution of the ADC is:

  resolution = Vin_max / $2^{\wedge 10}$ – 1

  resolution = 10 / 1023

  0.009775 = 10 / 1023

  or approximately

  10mV per bit when using 10-bits of accuracy

Analog to Digital Converter

- Resolution is a function of the number of data bits in the result

- What if we only wanted 8-bits of data?

Analog to Digital Converter

- Resolution of the ADC is:

  resolution = Vin_max / $2^{\wedge 8}$ – 1

  resolution = 10 / 255

  0.03921 = 10 / 255

  or approximately

  40mV per bit when using 8-bits of accuracy

8-bits vs. 10-bits

- The number of data bits is determined by the resolution requirements of the application
- The resolution has a direct impact on the accuracy of the system

10mV   or   40mV

8-bits vs. 10-bits

- A system with 10-bits of data vs. a system with only 8-bits of data is 4 times more resolution.

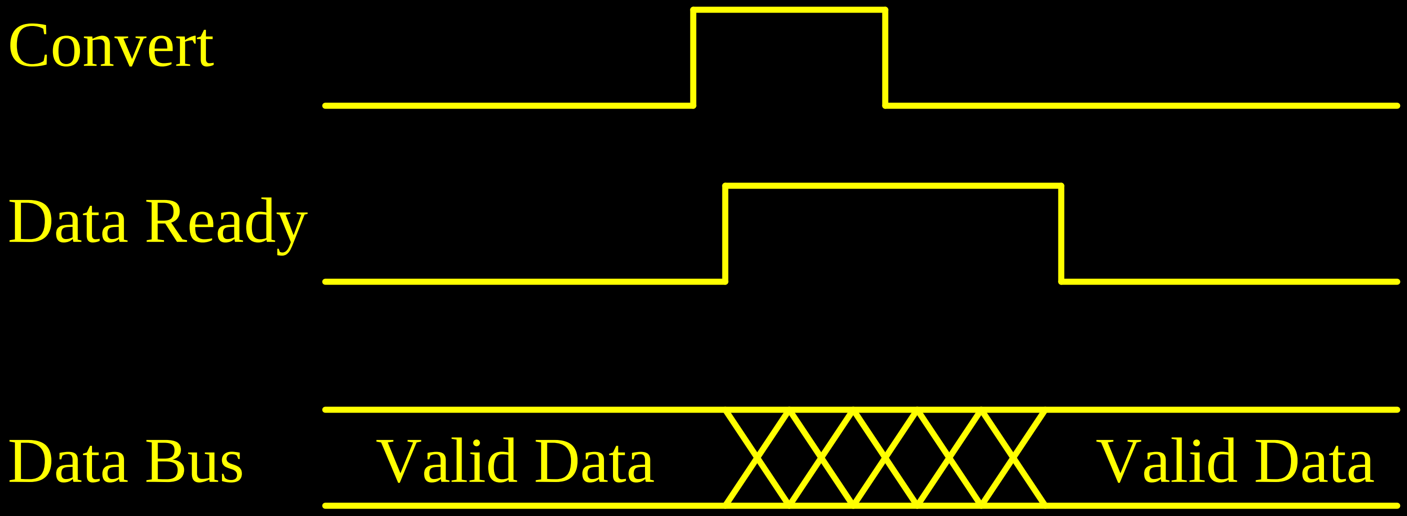
- Remember, $2^{\wedge\text{power}}$

- In this case, the power is 2 and therefore the difference is a factor of 4.

Using the ADC

- *Most* ADC’s require some type of “kick” to start the next conversion
- A *flag* will indicate the state or status of the current conversion
- The new data can be read only when the conversion has been completed

Timing Diagram for ADC



Convert

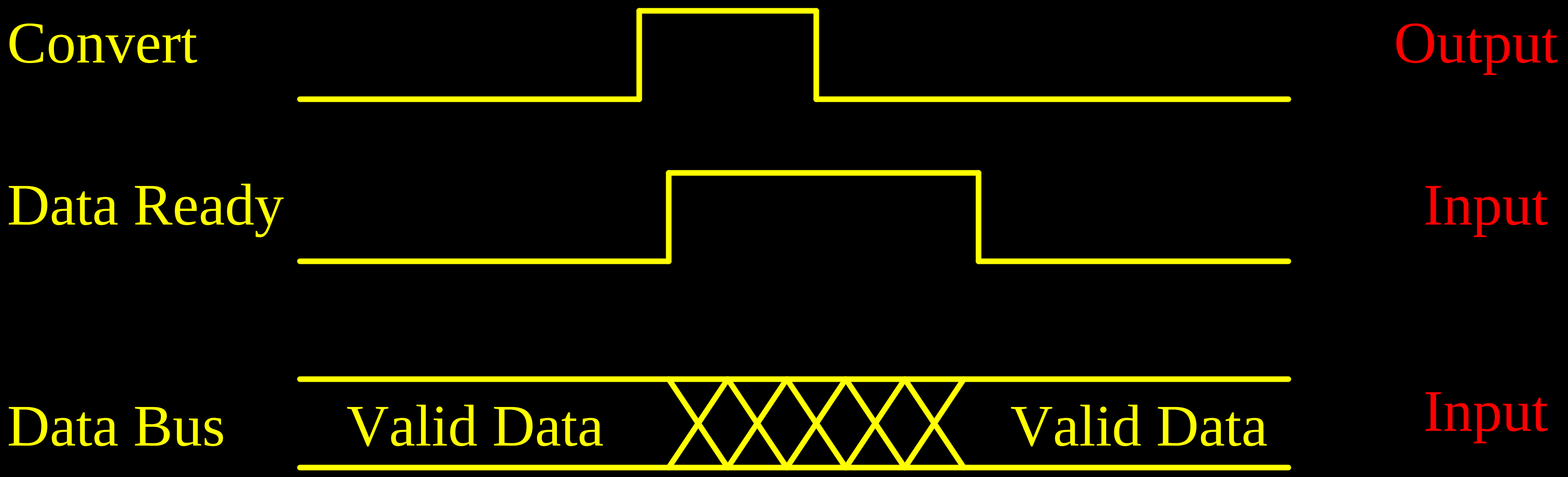
Data Ready

Data Bus

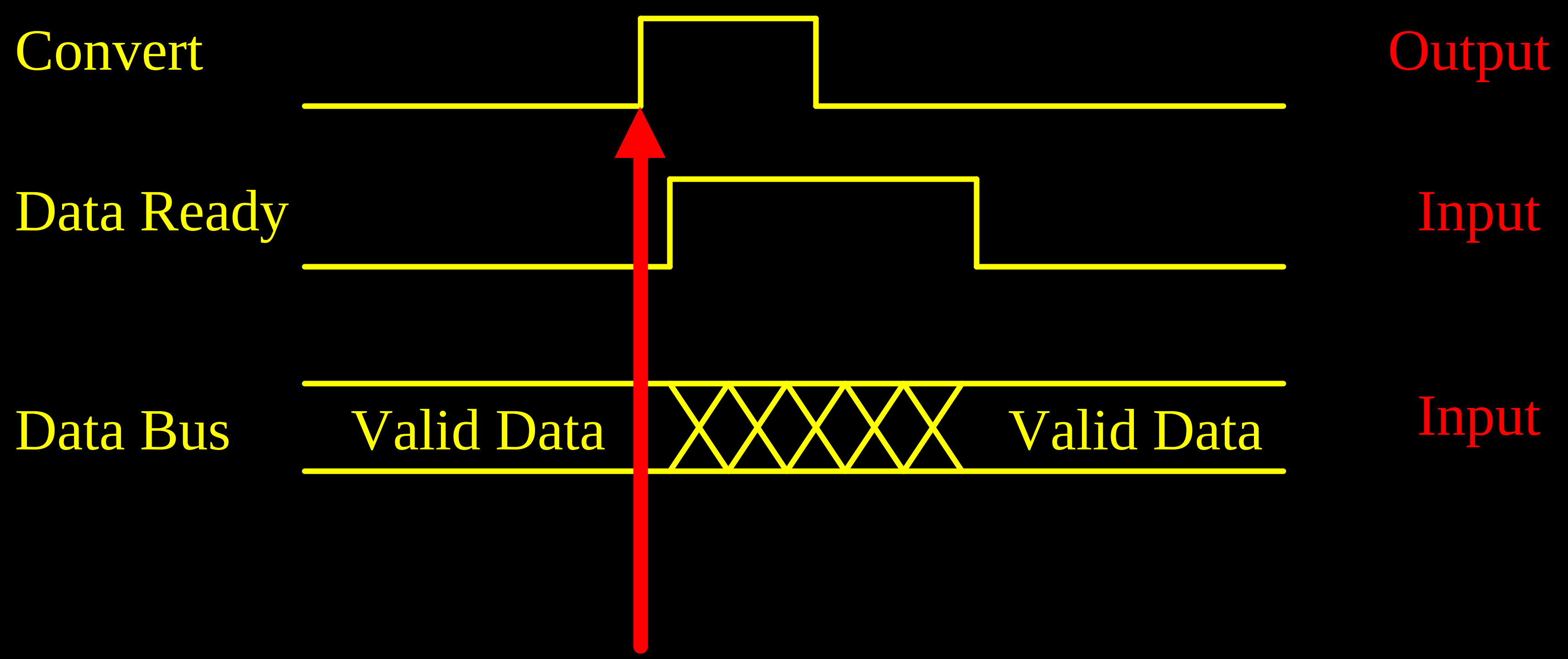
Valid Data

Valid Data

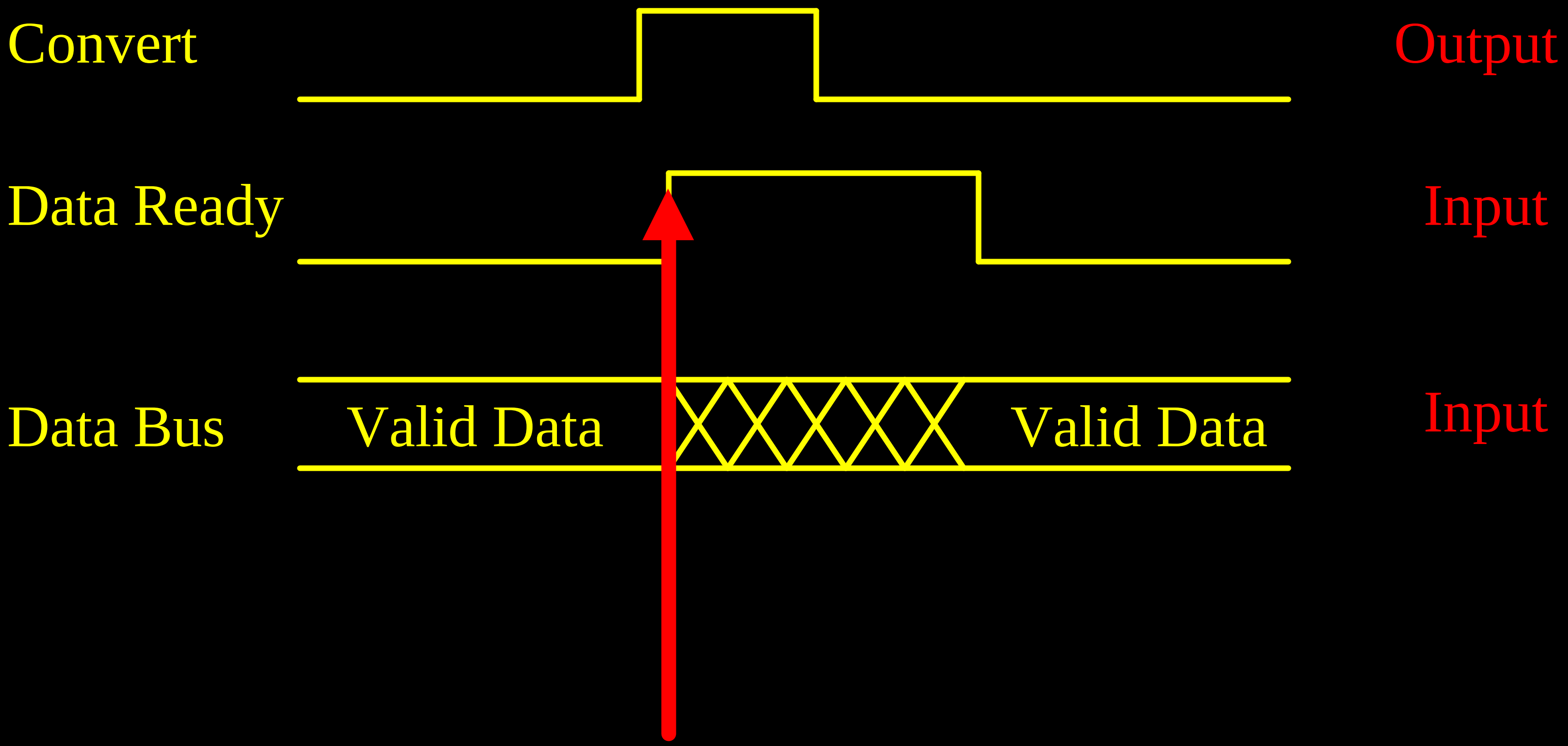
Timing Diagram for ADC



Timing Diagram for ADC



Timing Diagram for ADC



Convert — Output

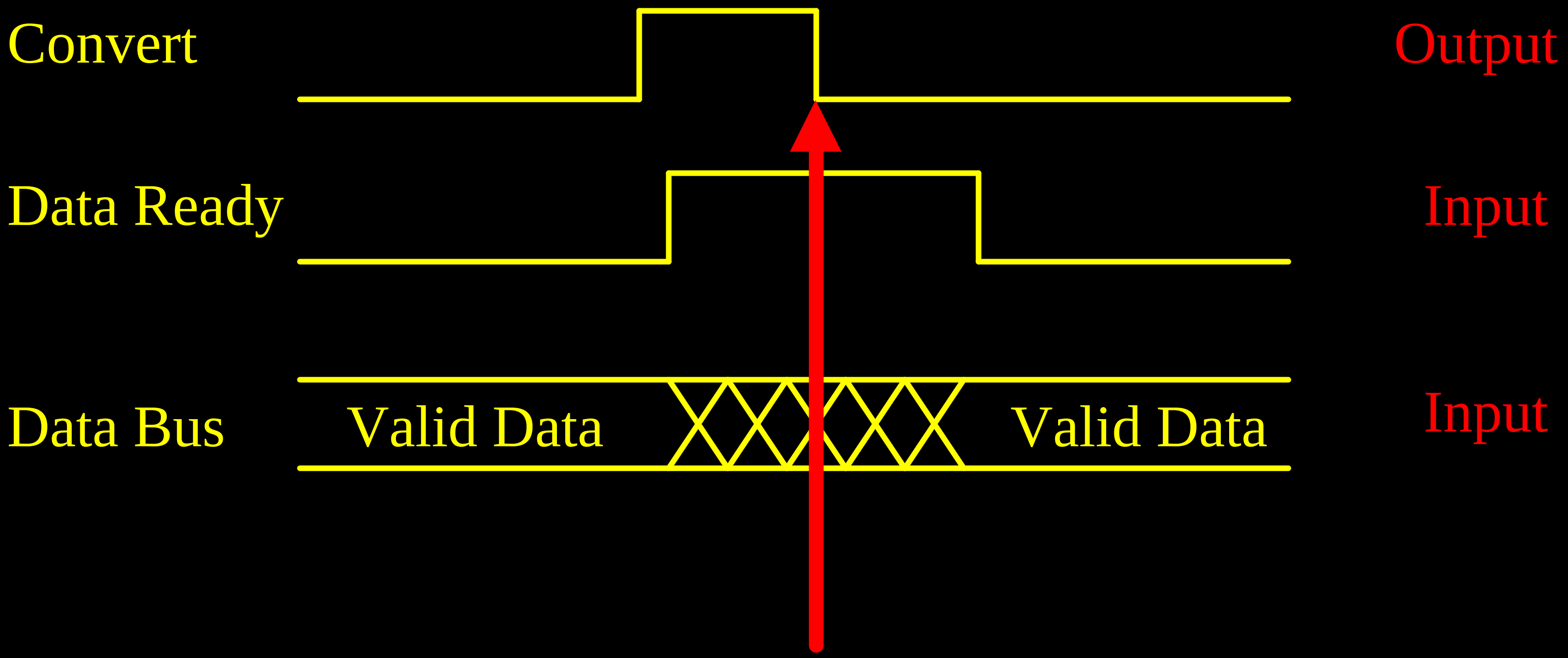
Data Ready — Input

Data Bus — Input

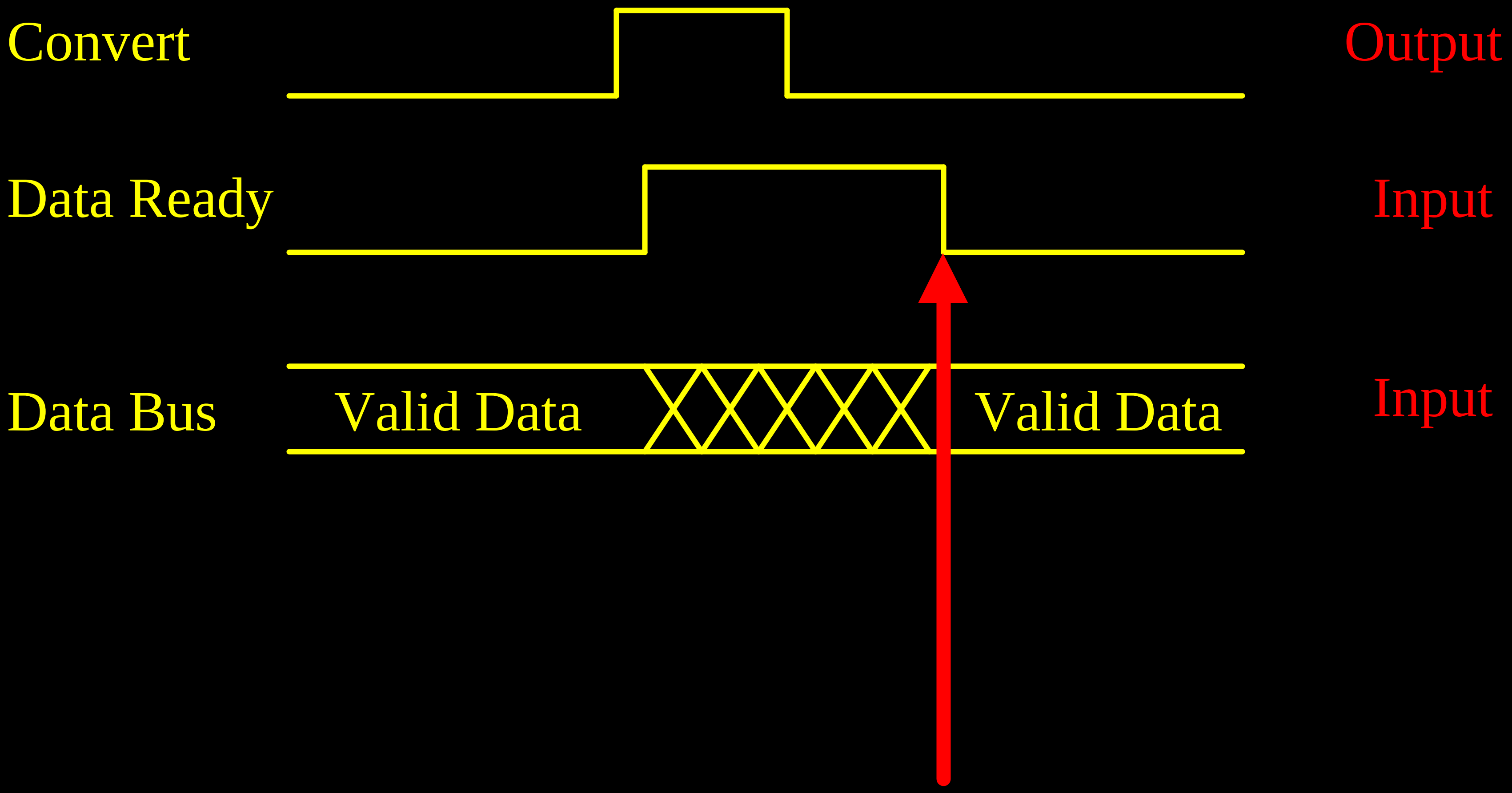
Valid Data

Valid Data

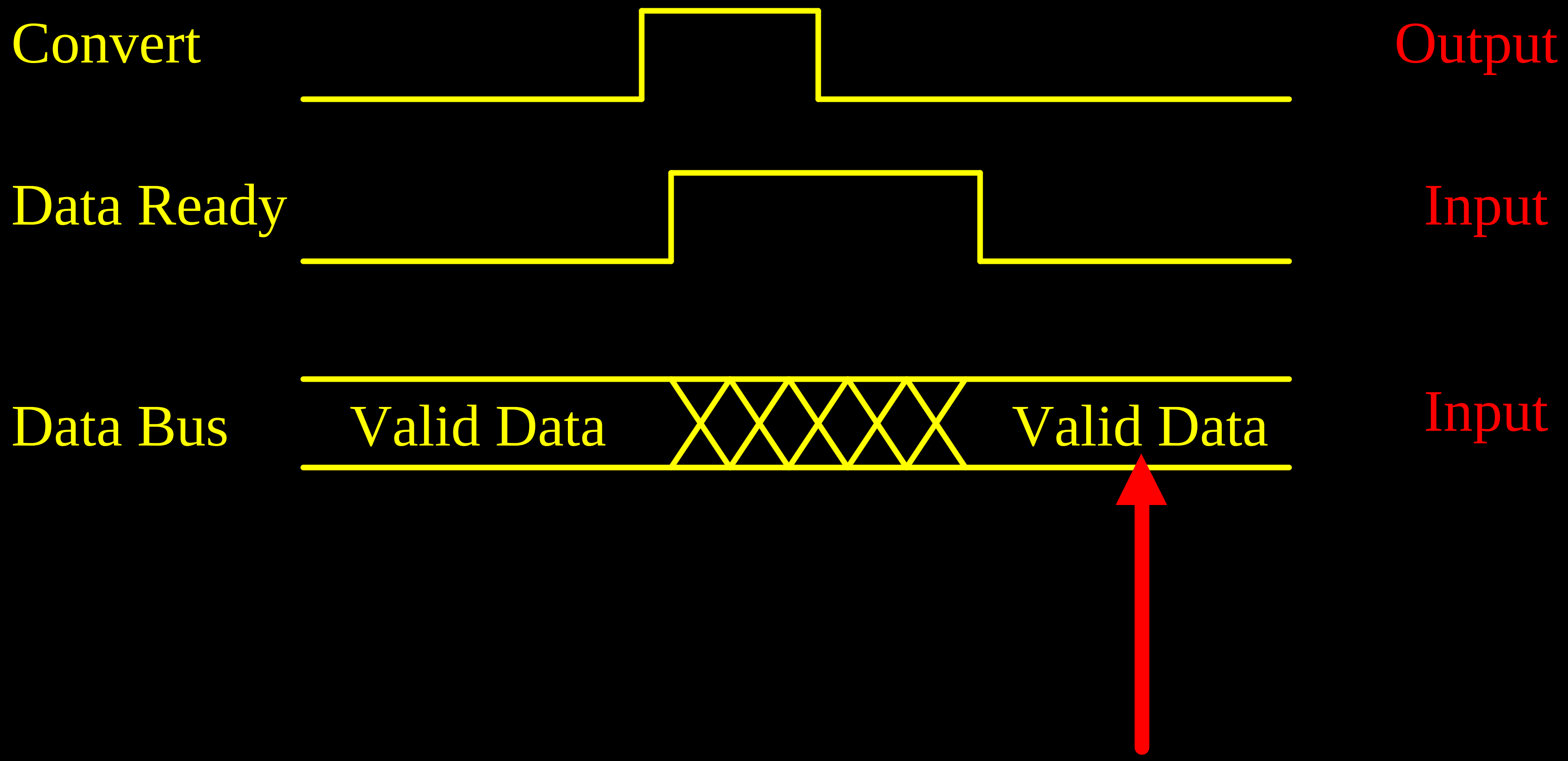
Timing Diagram for ADC



Timing Diagram for ADC



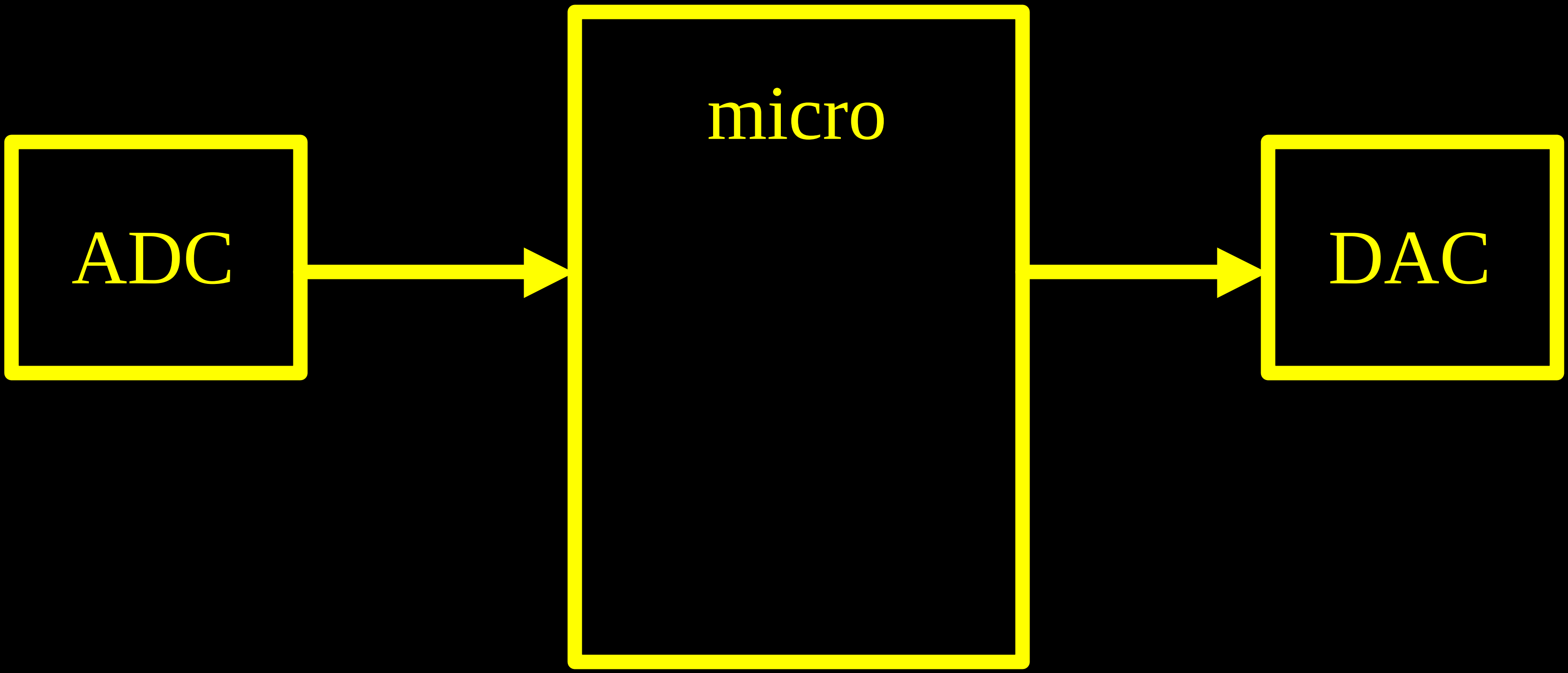
Timing Diagram for ADC



Solution Steps to use the ADC

1. Set the convert line high
2. Wait for Data Ready to go high
3. Clear the convert line
4. Wait for the Ready Line to go low
5. Read the result of the conversion

Analog and Digital Converters

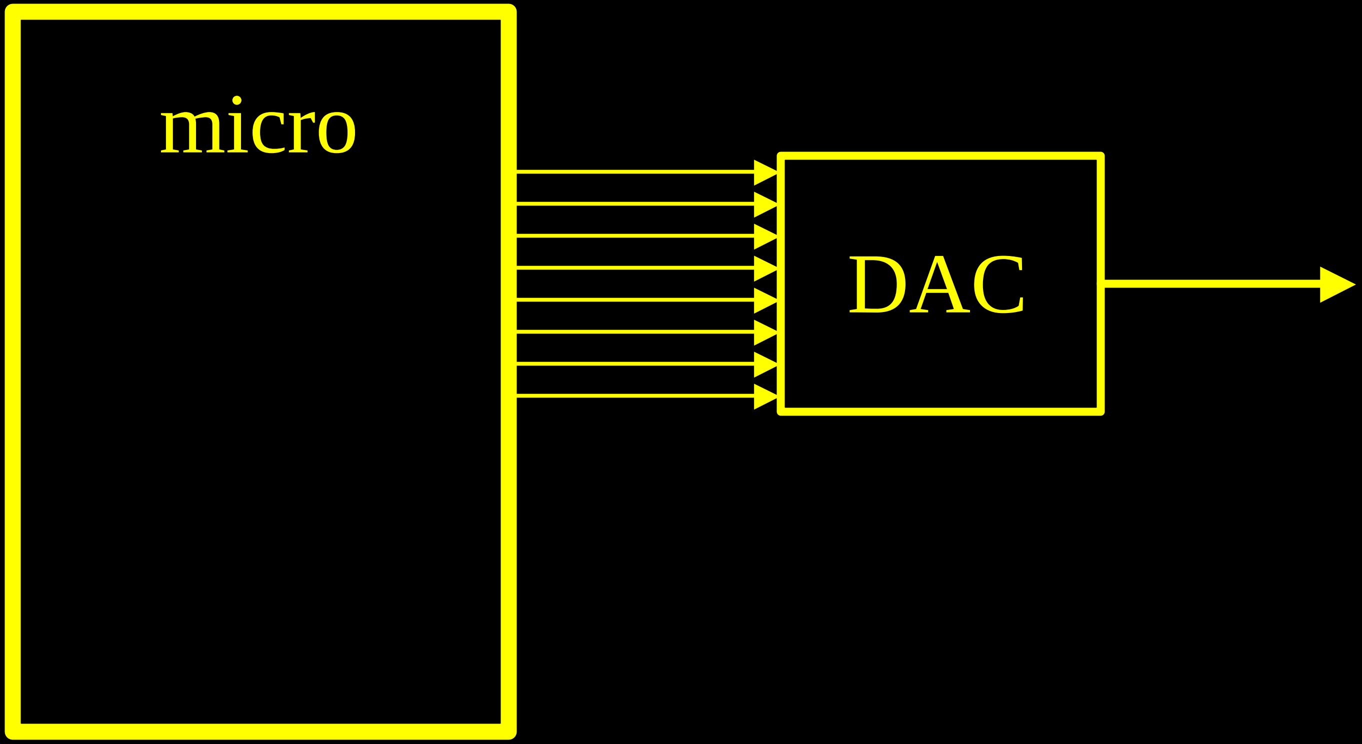


ADC

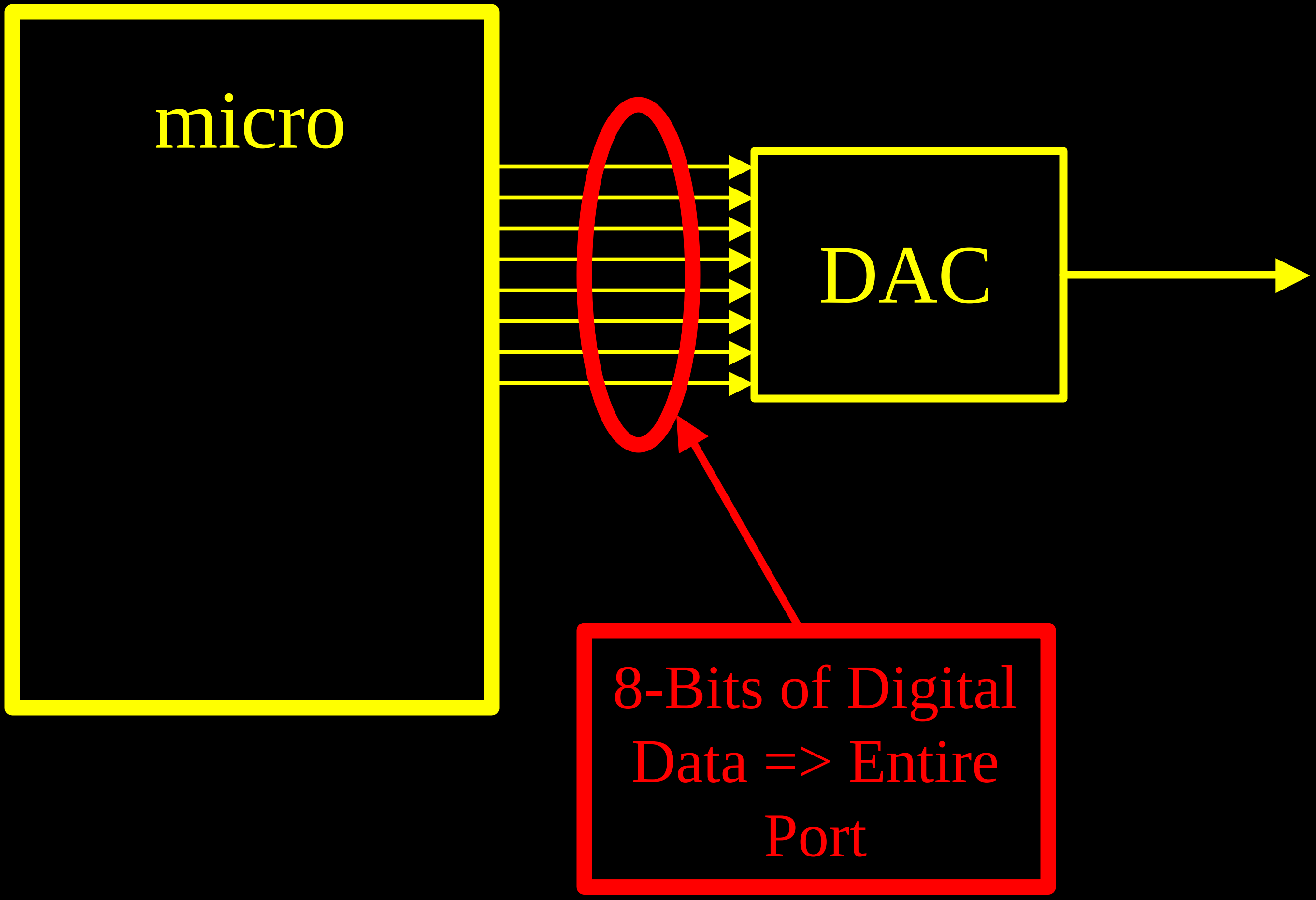
micro

DAC

Digital to Analog Converter

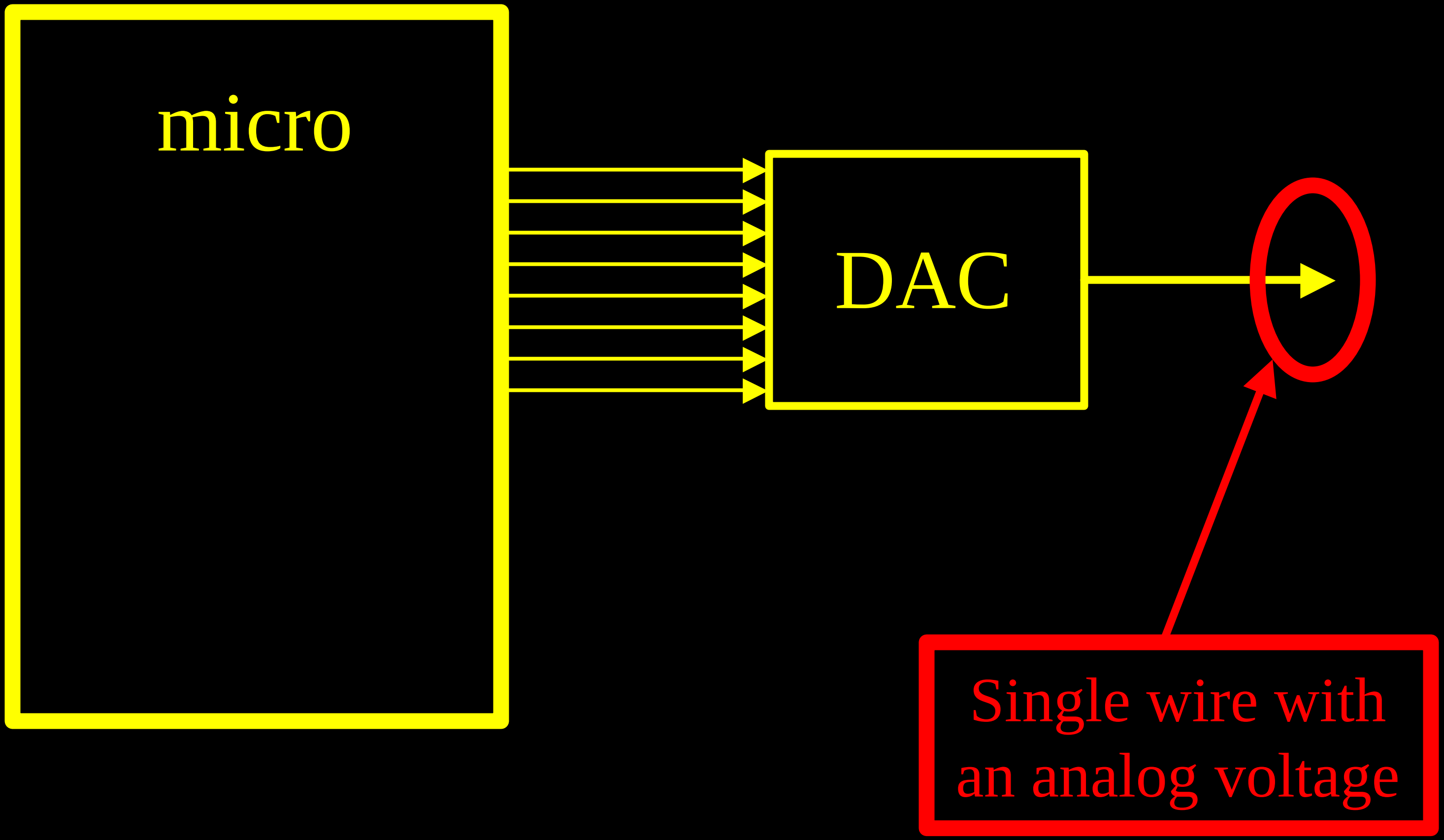


Digital to Analog Converter



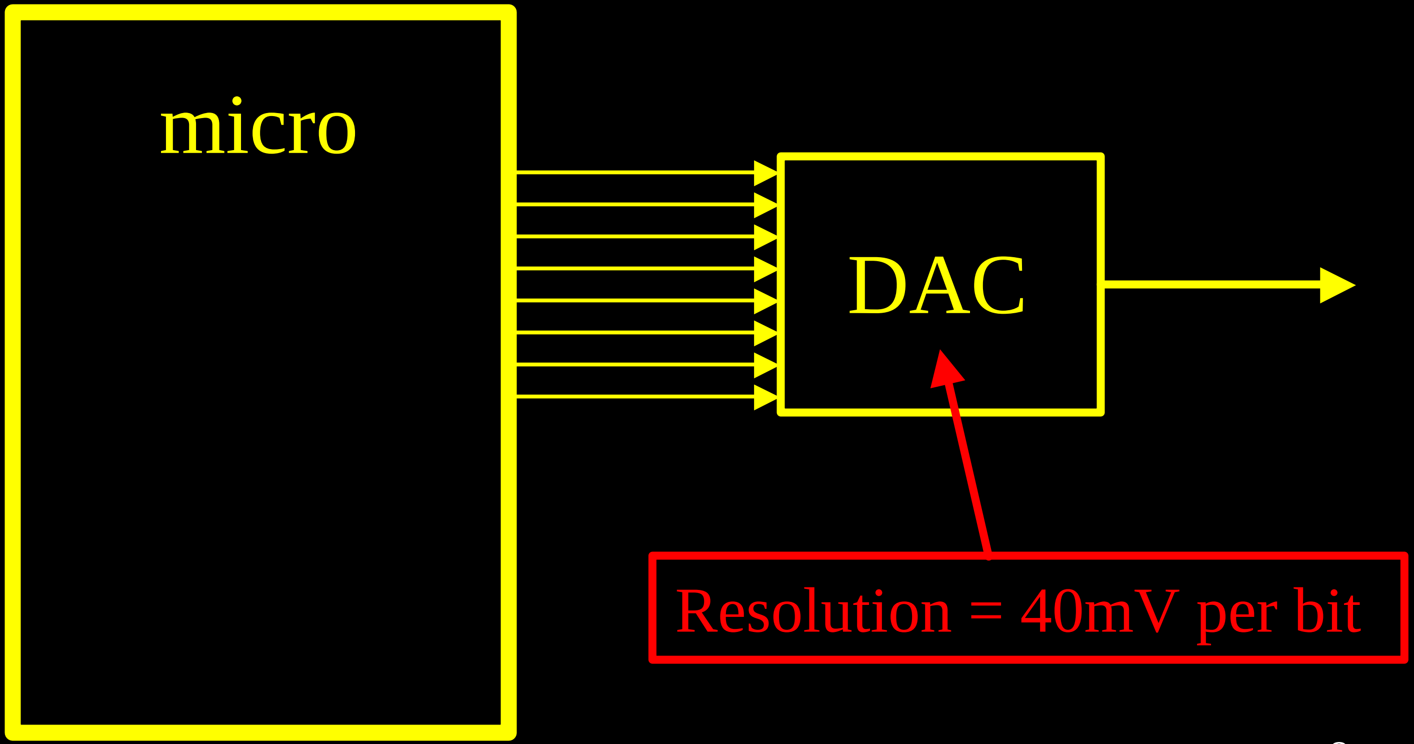
8-Bits of Digital Data => Entire Port

Digital to Analog Converter



Single wire with an analog voltage

Digital to Analog Converter



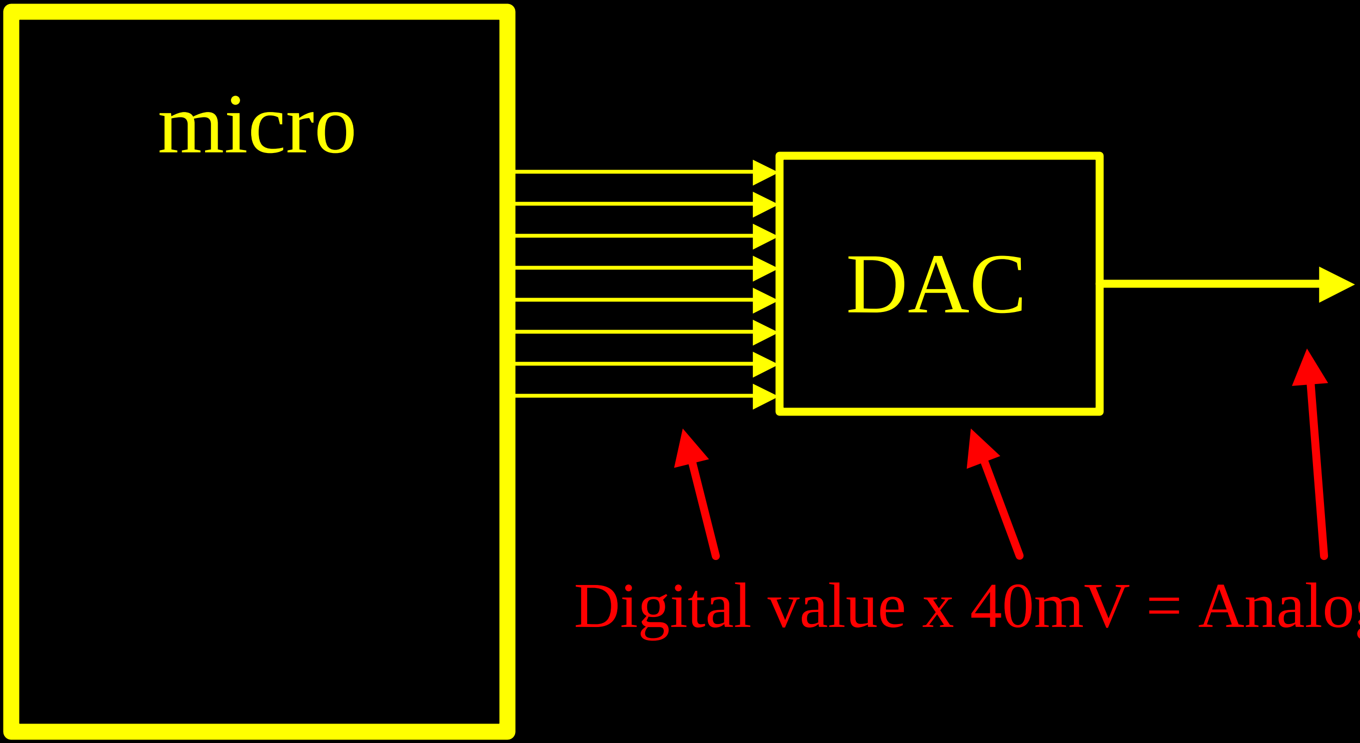
micro

DAC

Resolution = 40mV per bit

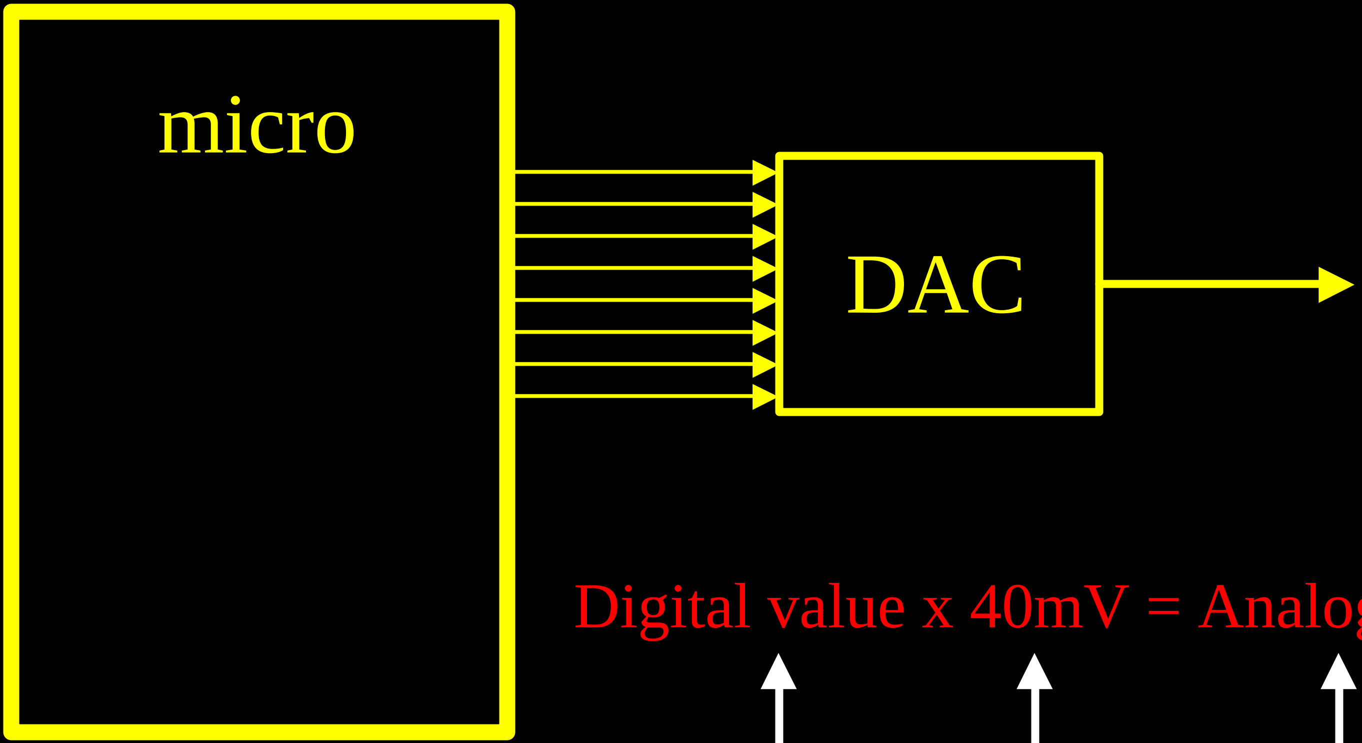
resolution = Vout_max / 2^^{8} - 1

Digital to Analog Converter



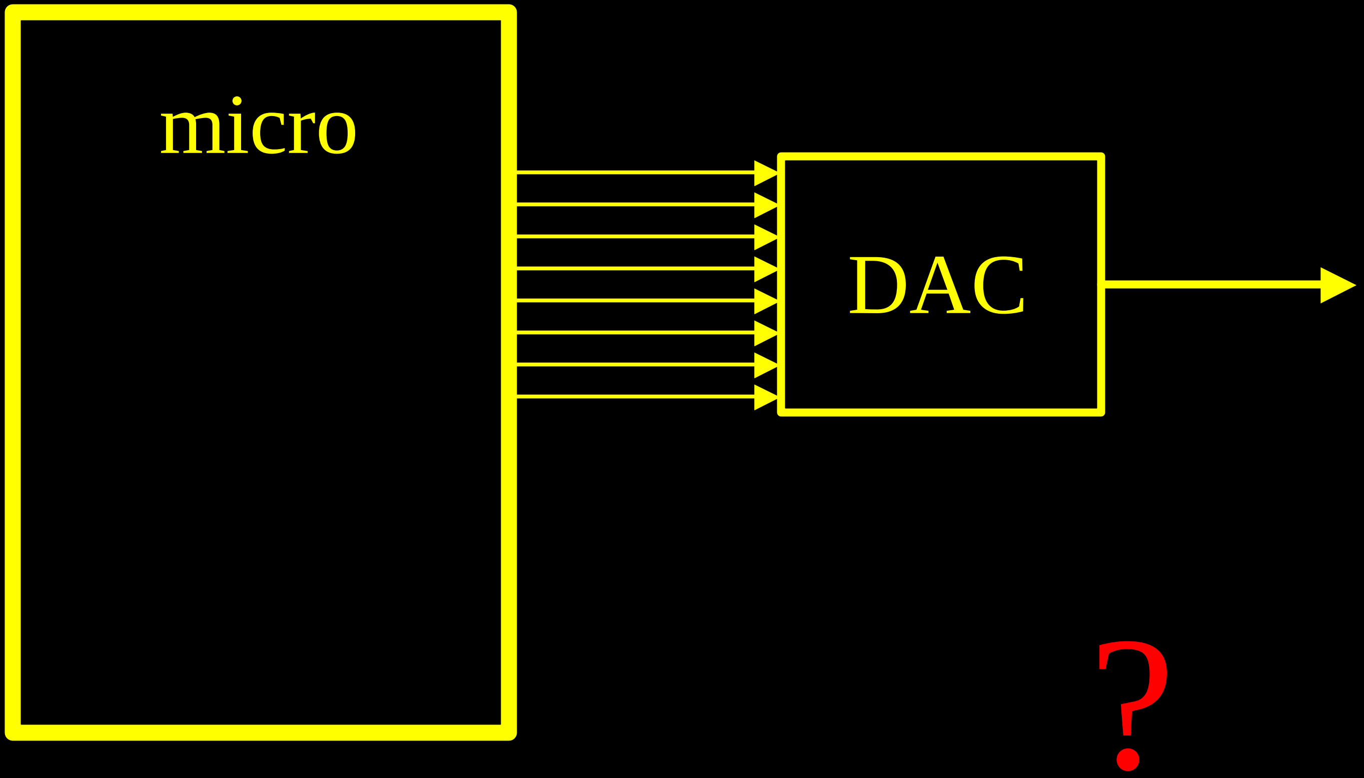
Digital value x 40mV = Analog Out

Digital to Analog Converter

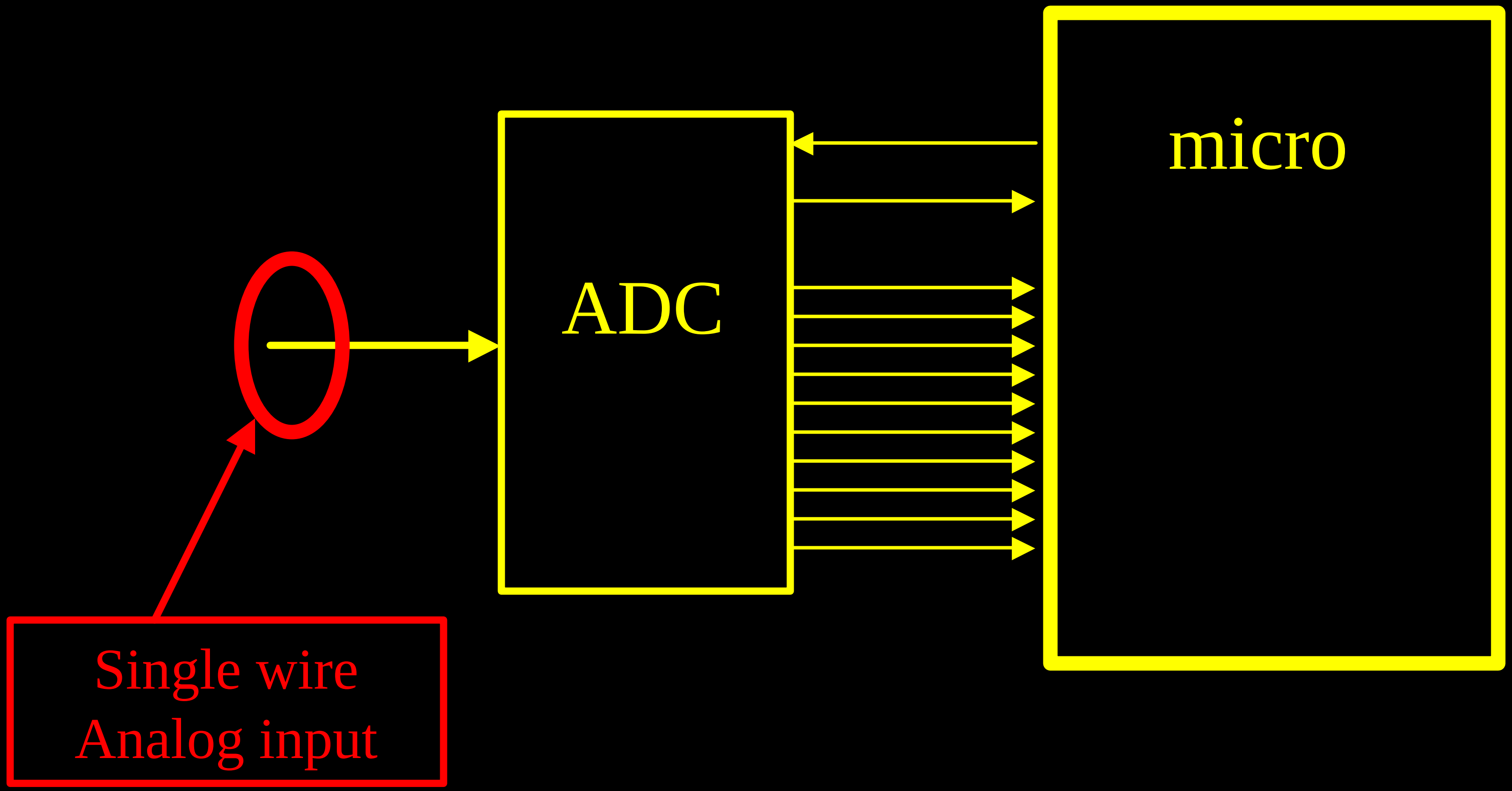


Digital value x 40mV = Analog Out

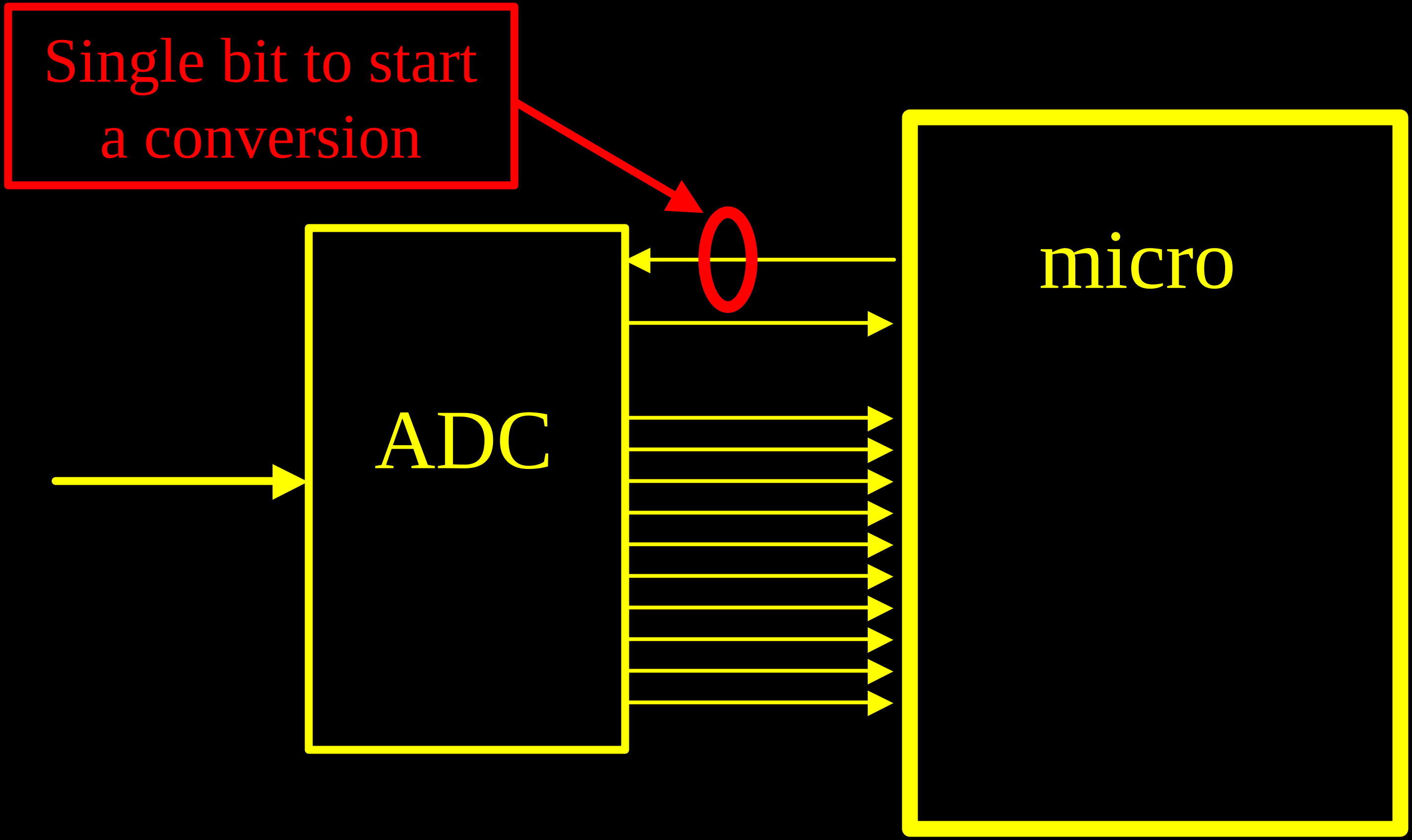
Digital to Analog Converter



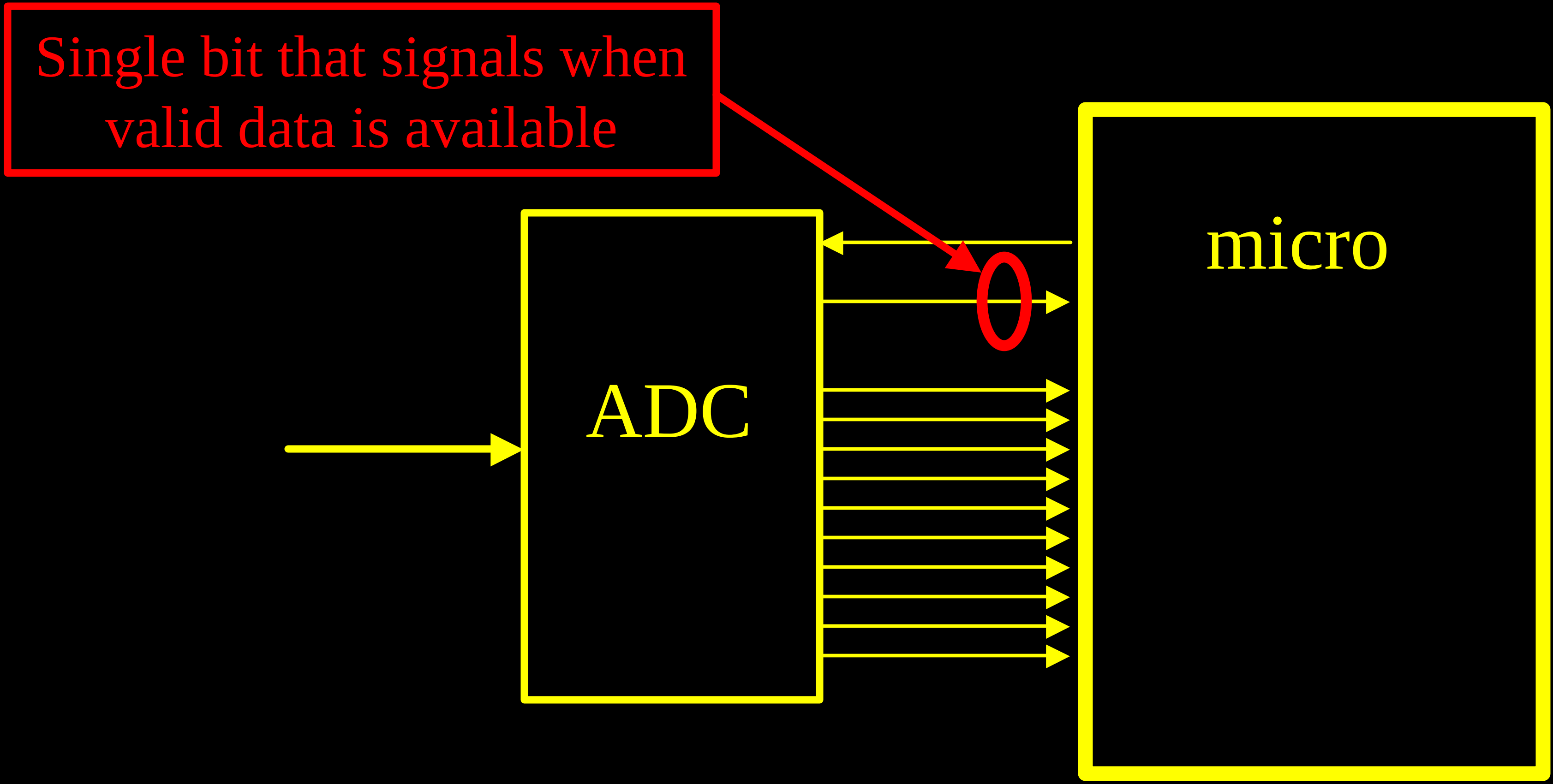
Analog to Digital Converter



Analog to Digital Converter

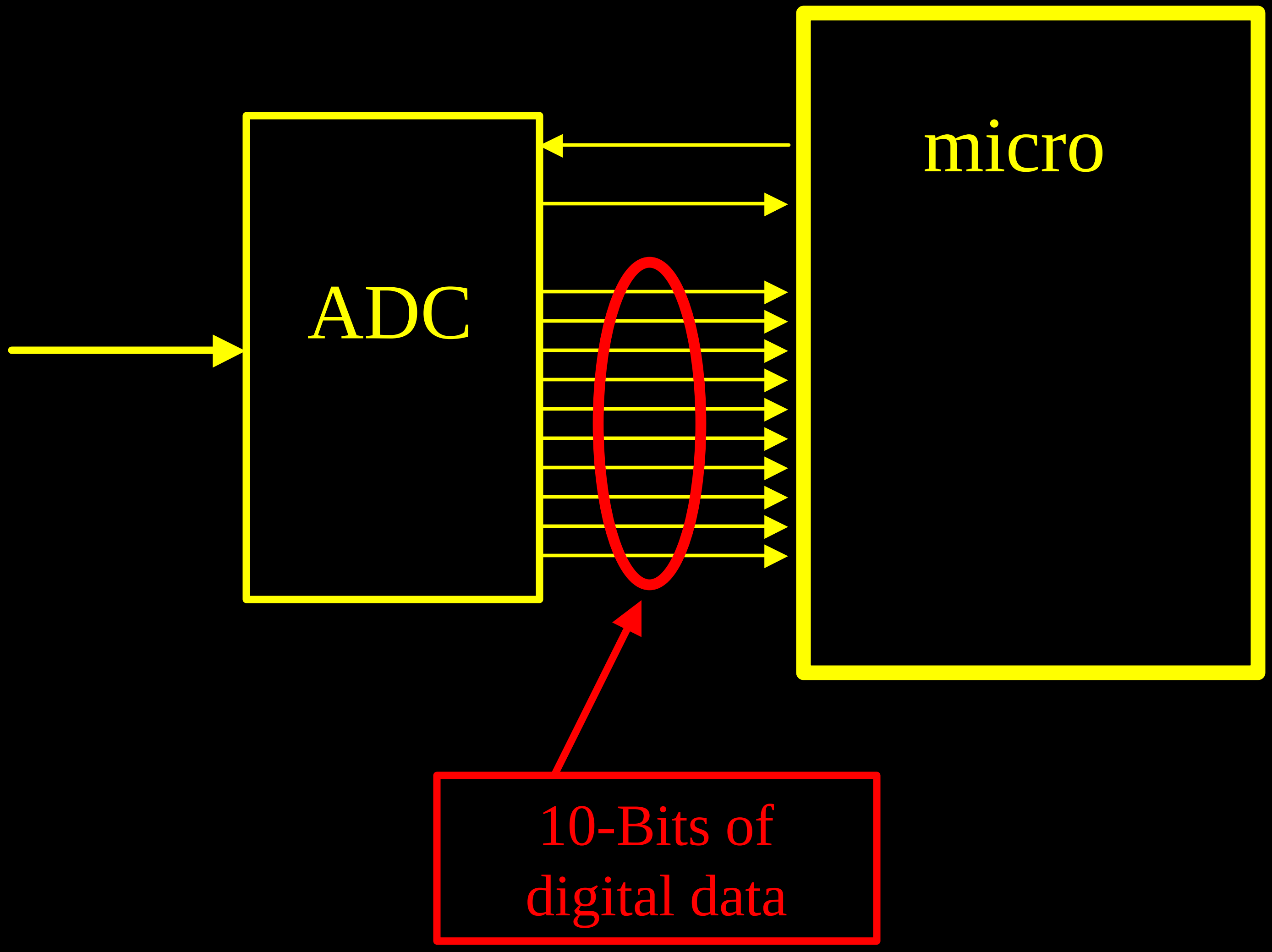


Analog to Digital Converter

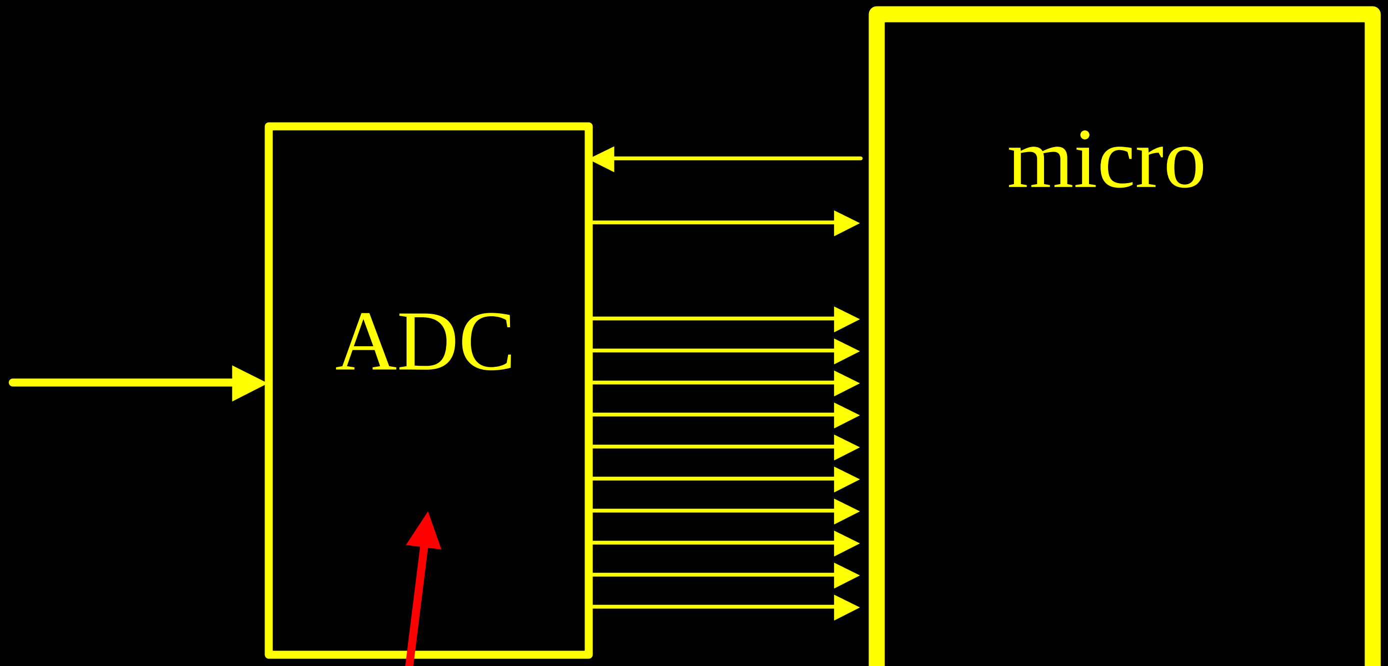


Single bit that signals when valid data is available

Analog to Digital Converter

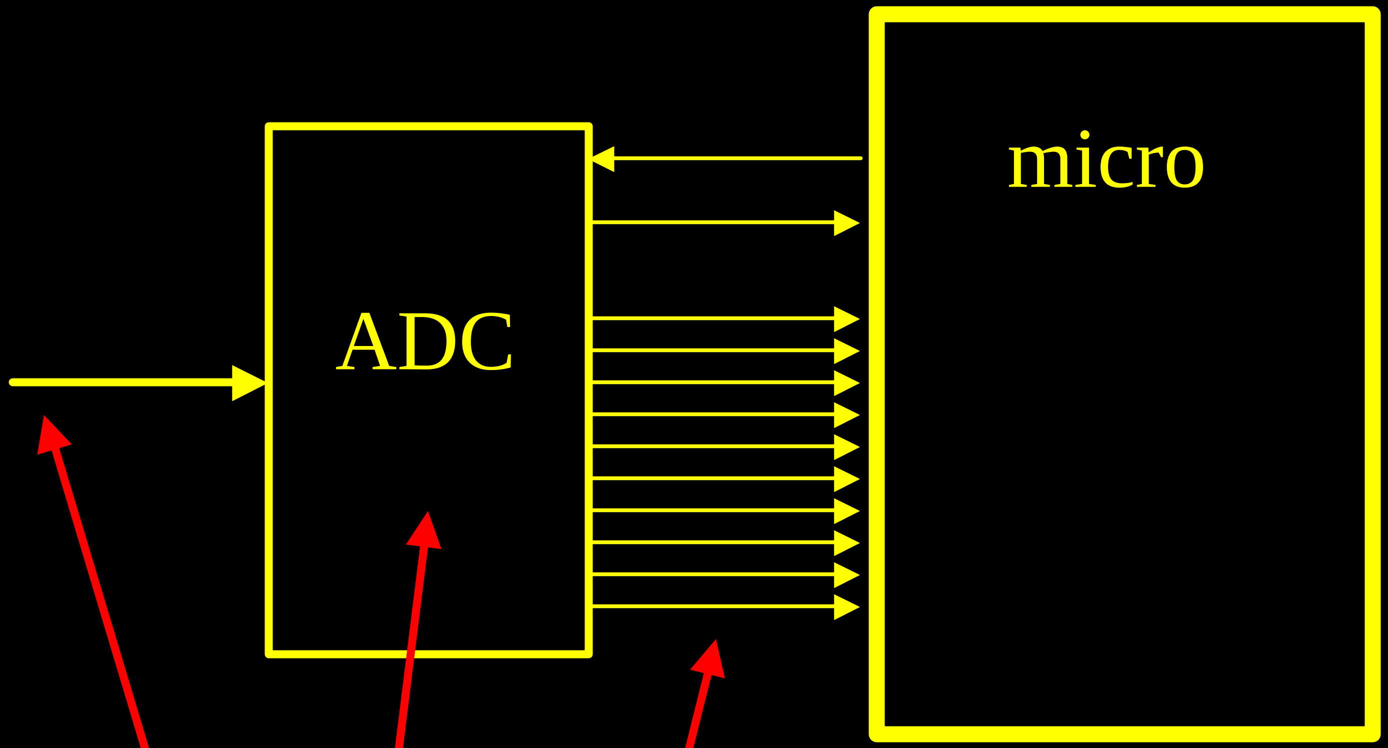


Analog to Digital Converter



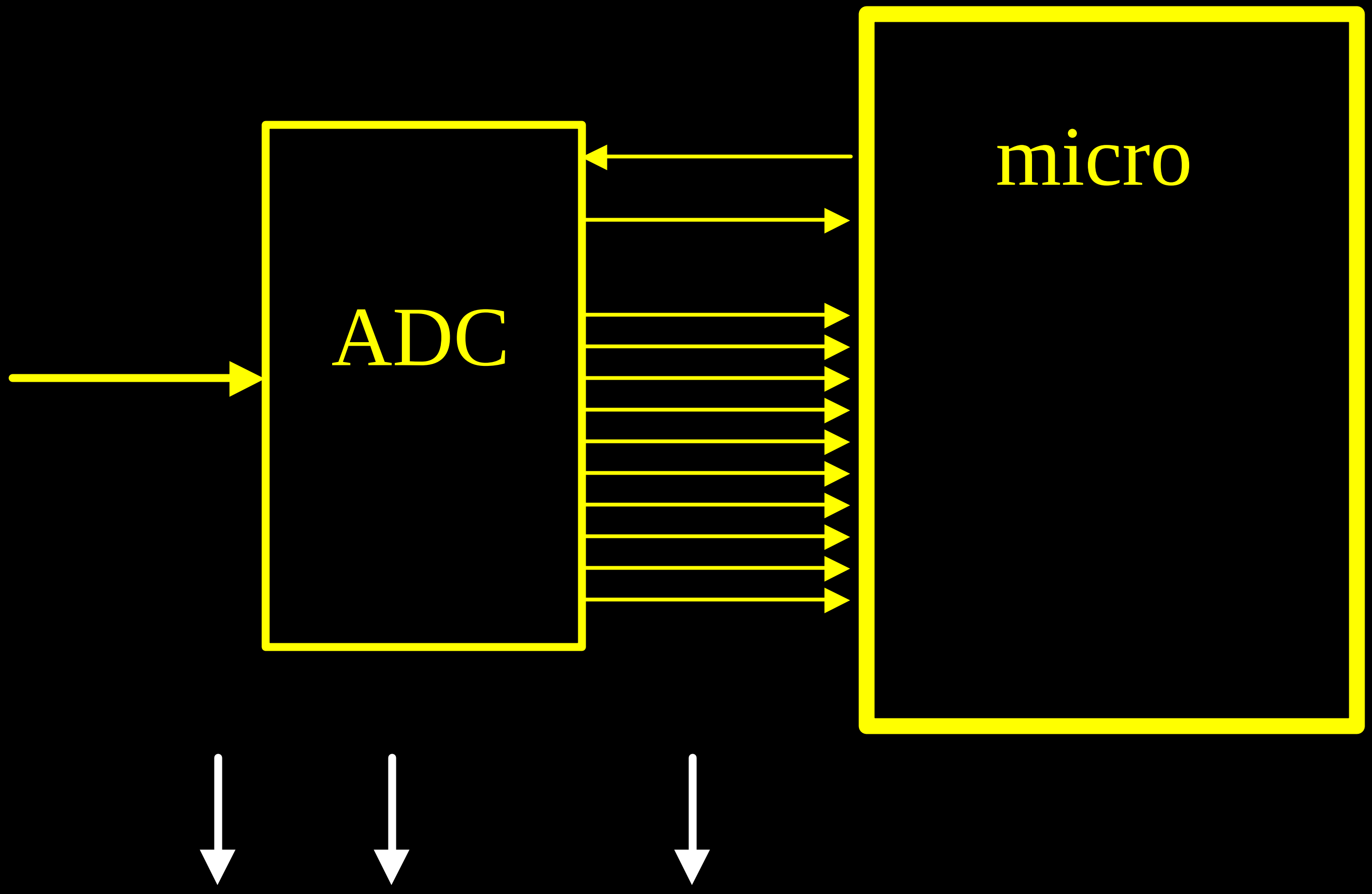
Resolution = 10mV per bit with 10-Bits of data

Analog to Digital Converter



Vin / 10mV = digital output

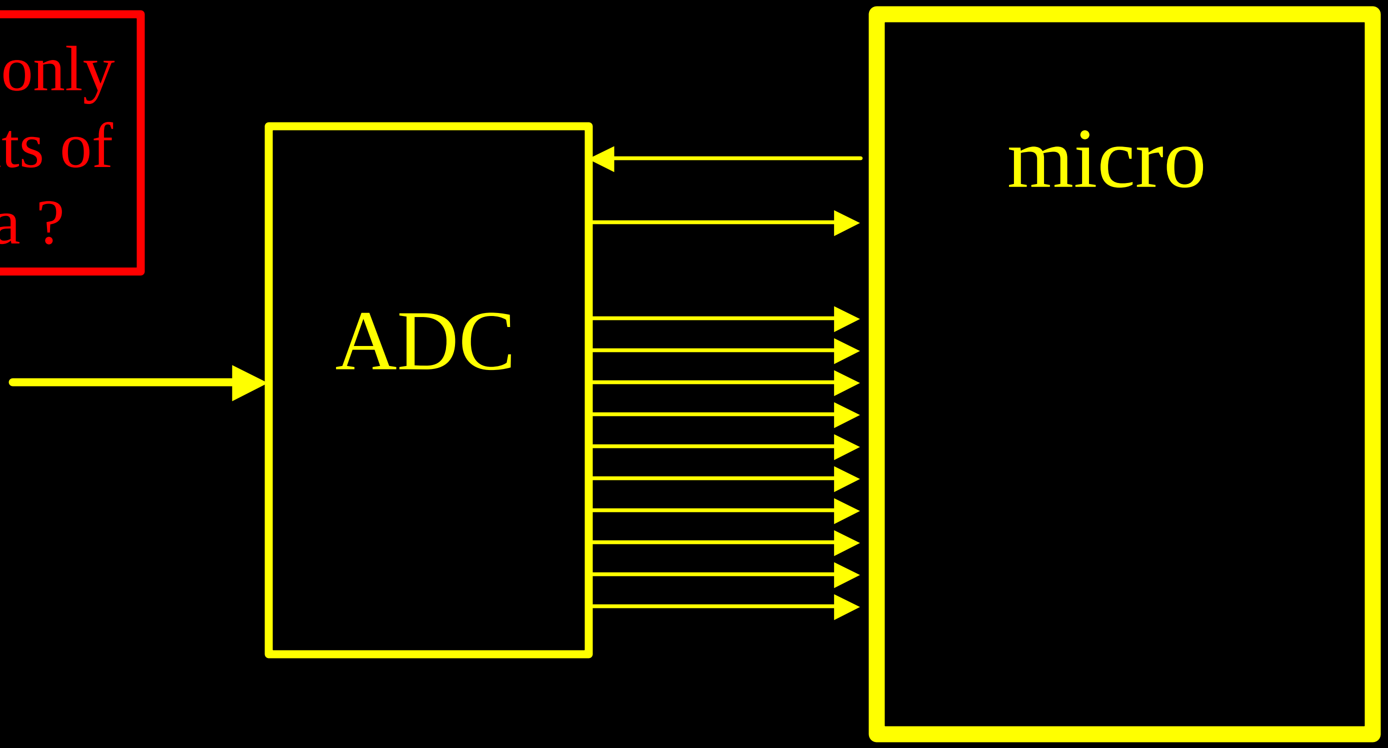
Analog to Digital Converter



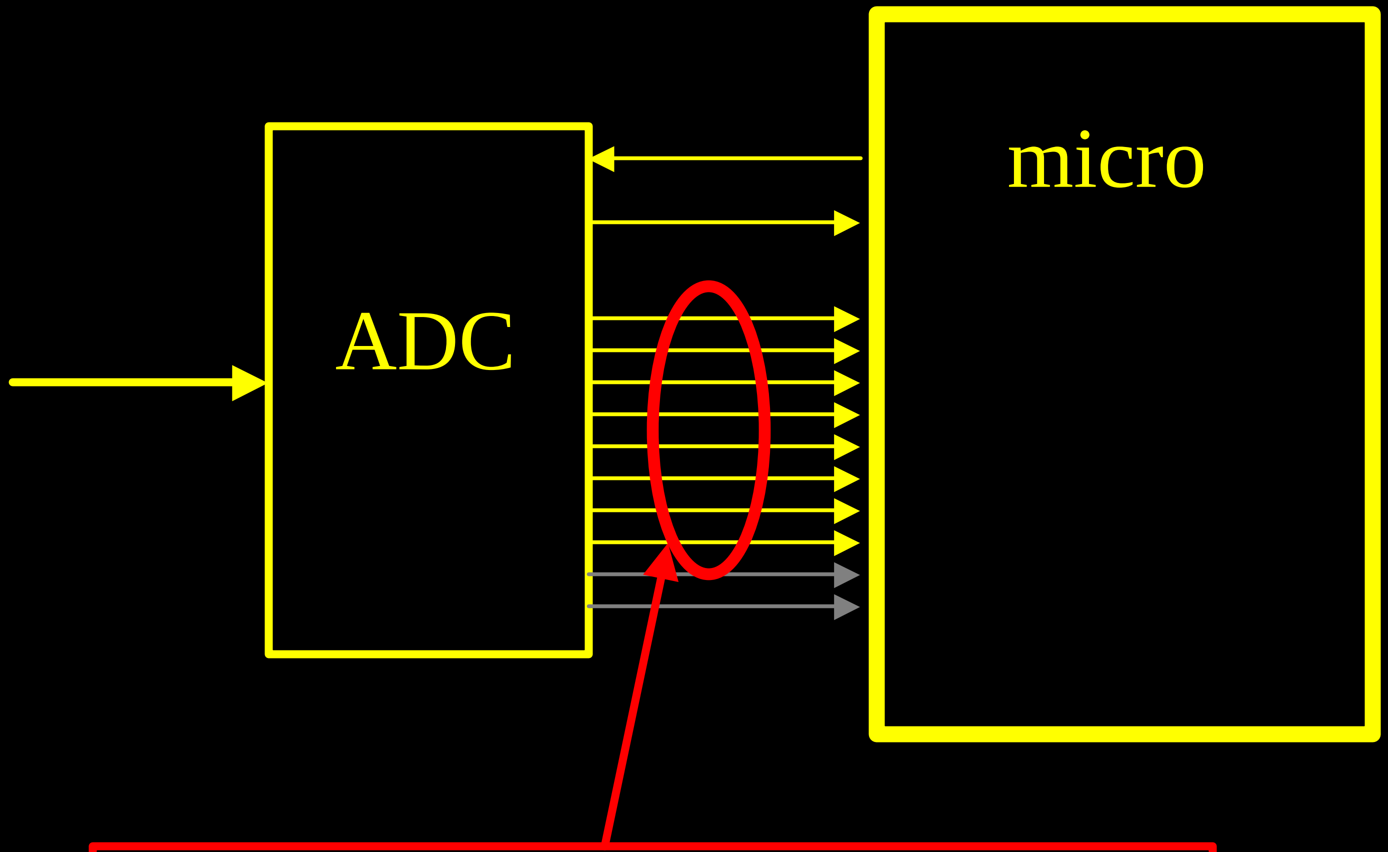
Vin / 10mV = digital output

Analog to Digital Converter

What if, we only wanted 8-Bits of digital data ?

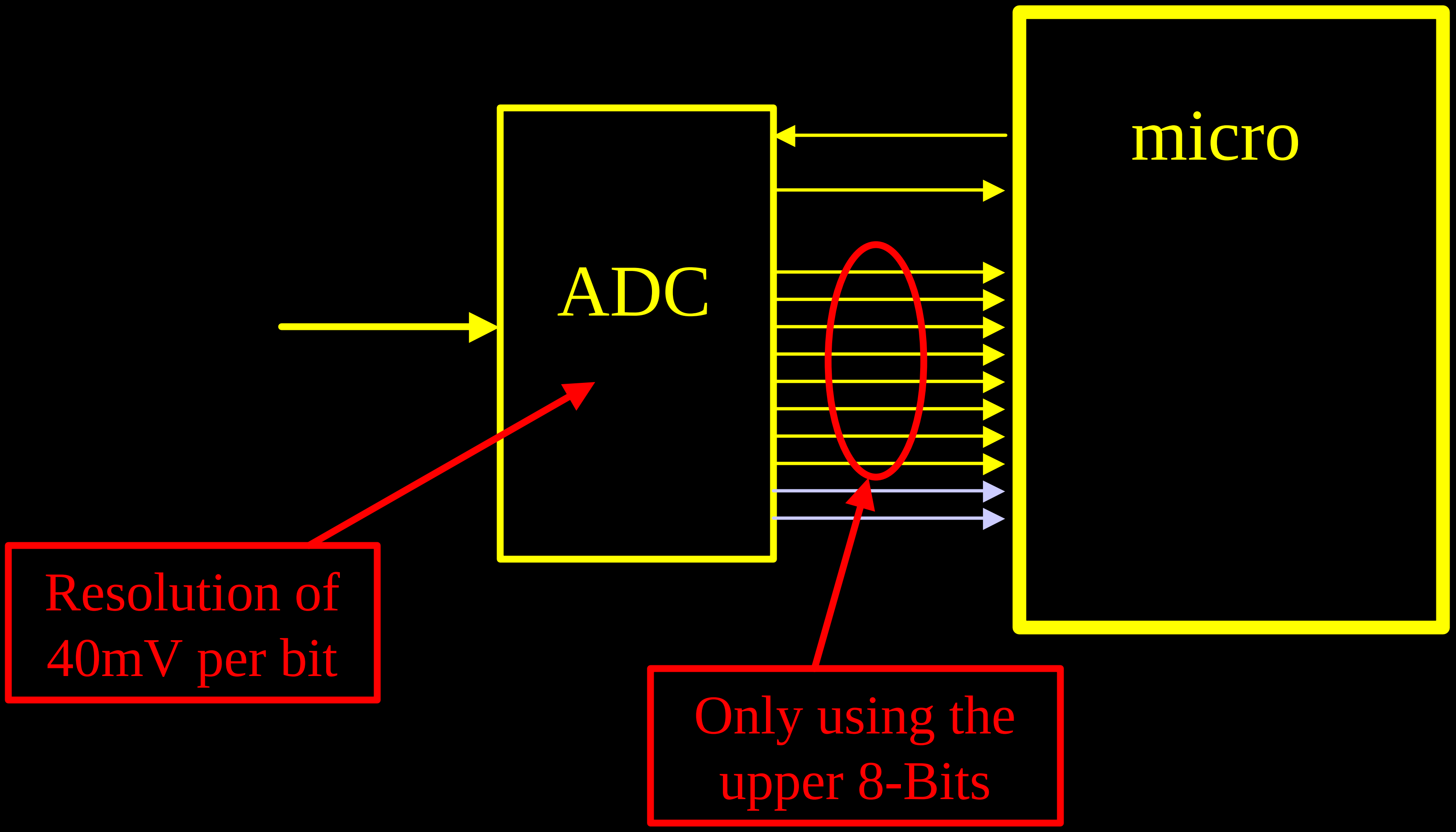


Analog to Digital Converter



Use only upper 8-Bits of digital data

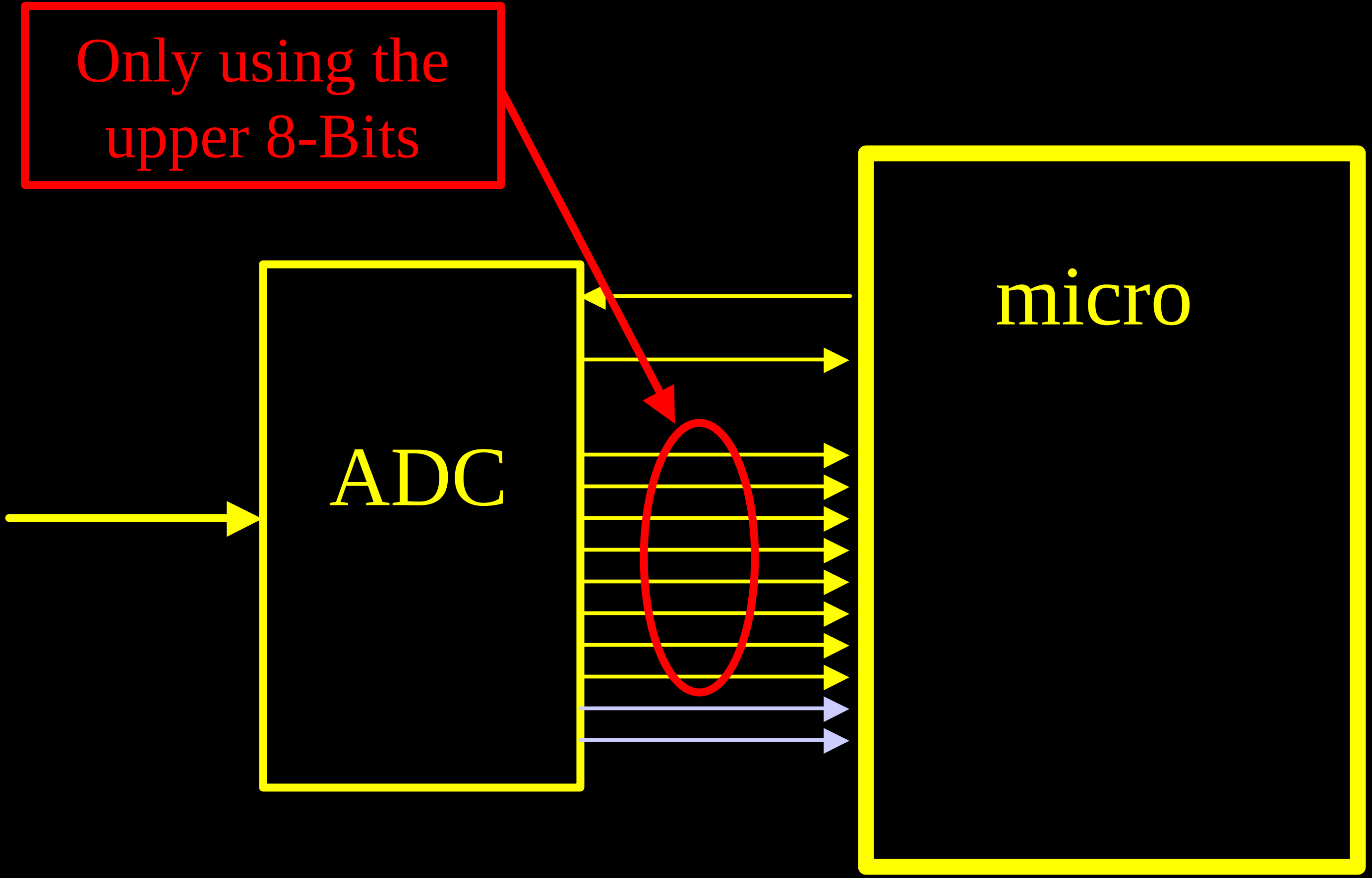
Analog to Digital Converter



Resolution of 40mV per bit

Only using the upper 8-Bits

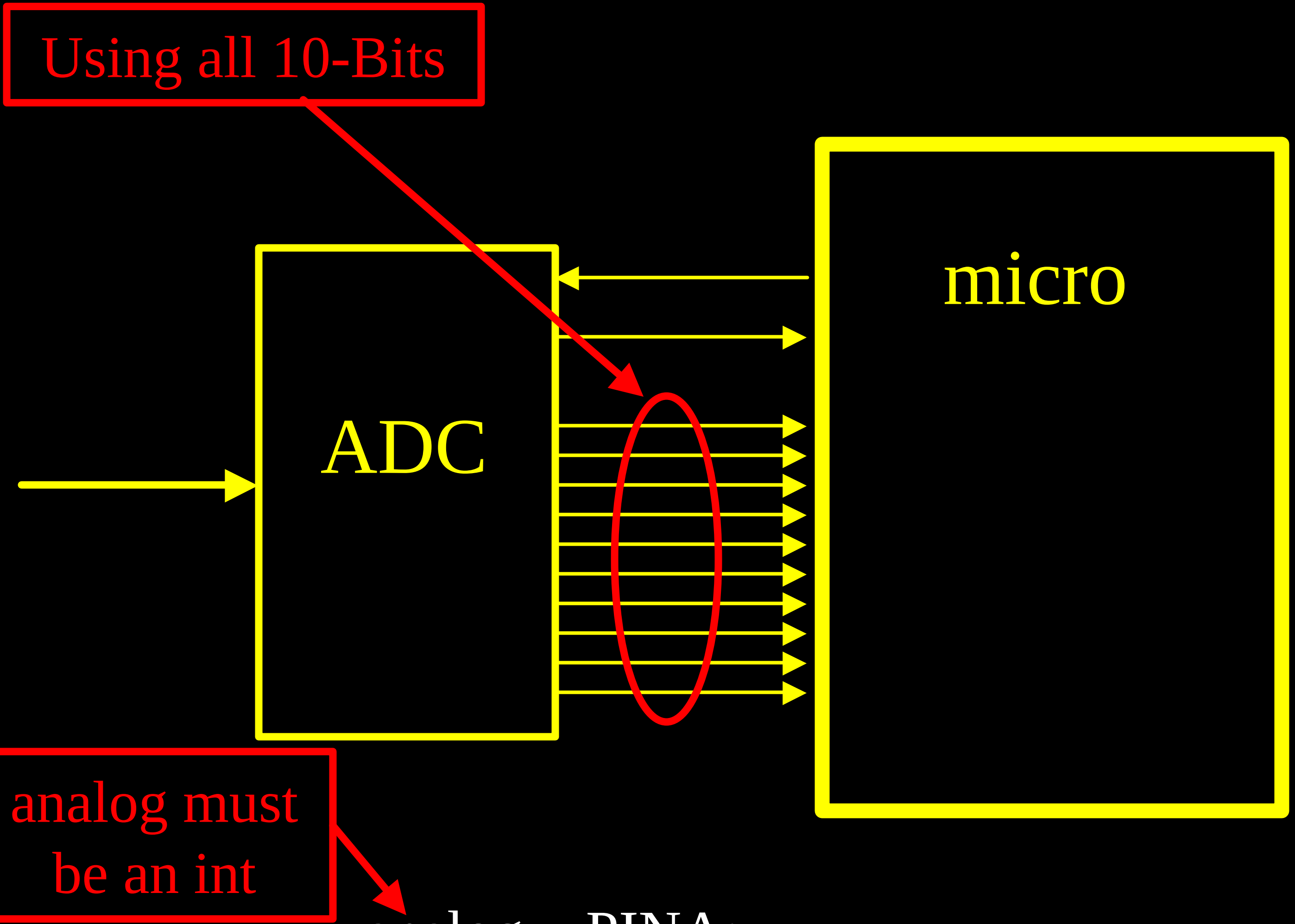
Reading the ADC



*Assuming ADC is connected to PORTA

analog = PINA;

Reading the ADC

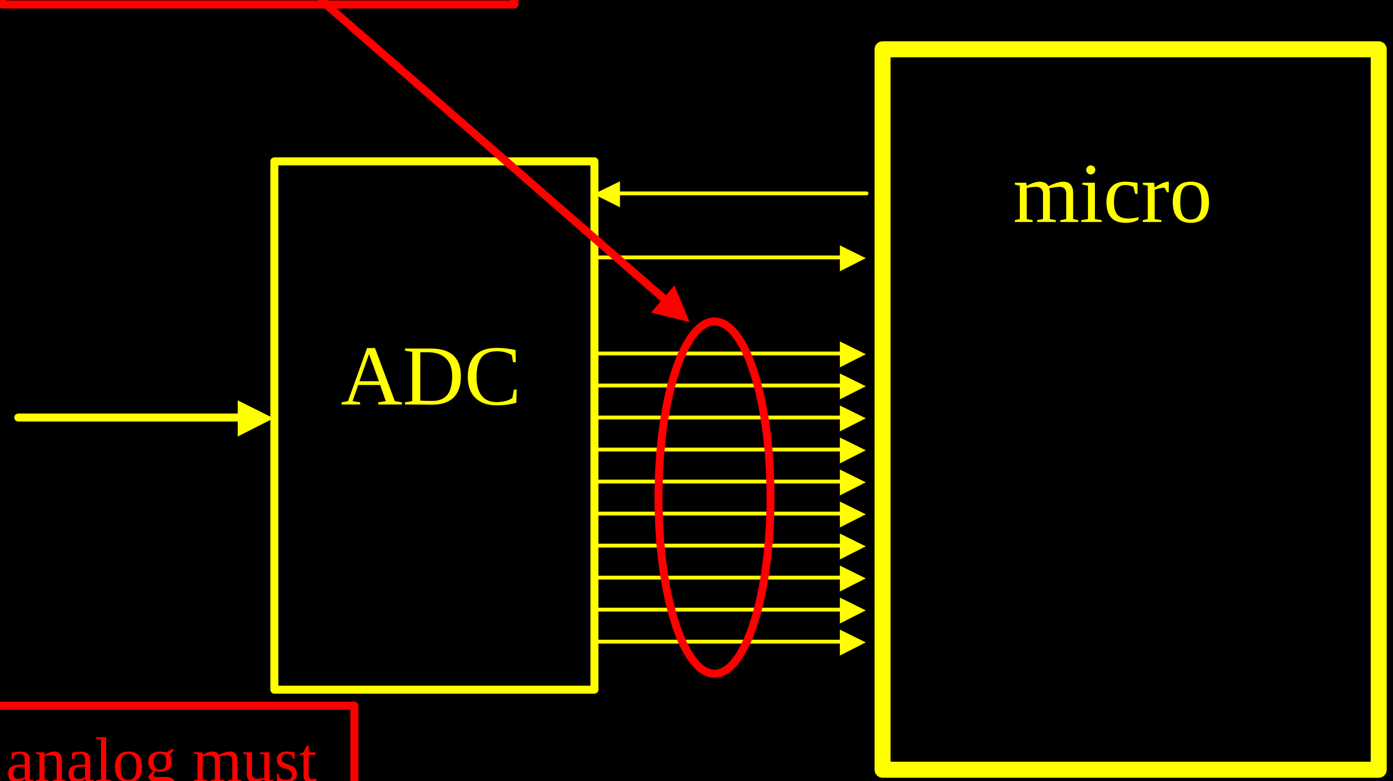


*Assuming upper 8-bits of the ADC is connected to PORTA & the lower 2-Bits are connected to PORTB bits 0 & 1

```
analog = PINA;
analog = (analog << 2) | (PINB & 0x03);
```

Reading the ADC

Using all 10-Bits



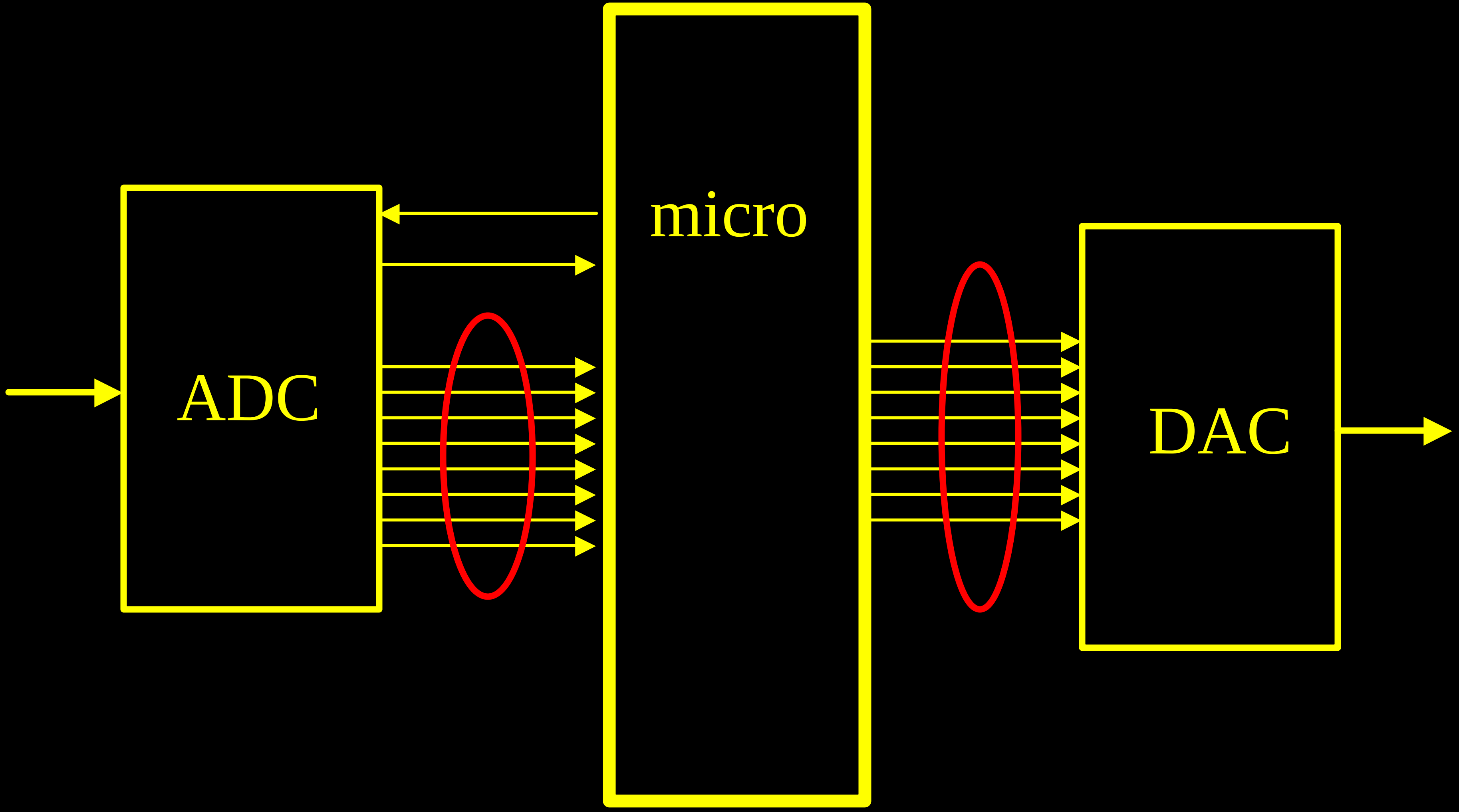
*Assuming upper 8-bits of the ADC is connected to PORTA & the lower 2-Bits are connected to PORTB bits 0 & 1

analog must be an int

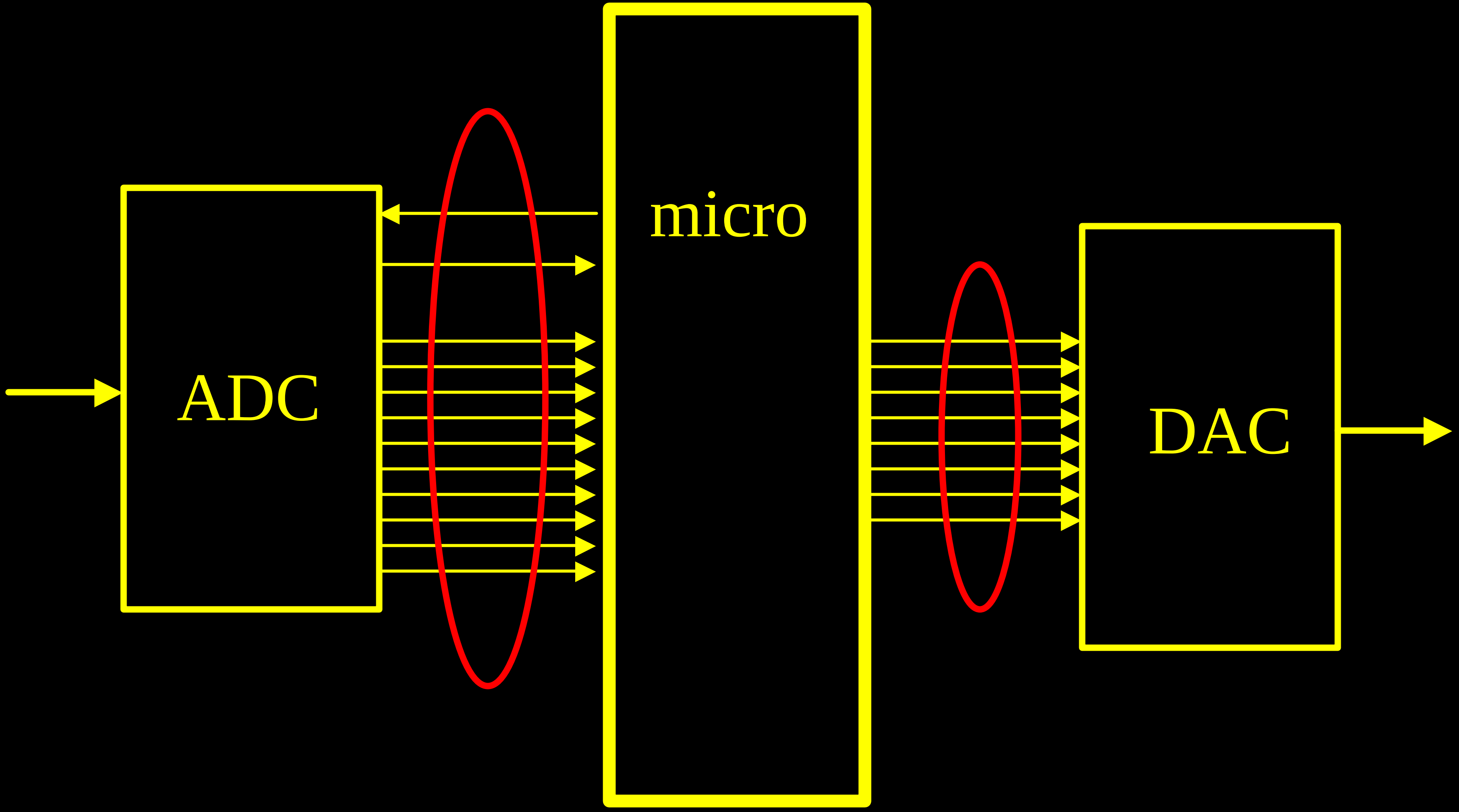
```
analog = PINA;
analog = (analog << 2) | (PINB & 0x03);
```

?

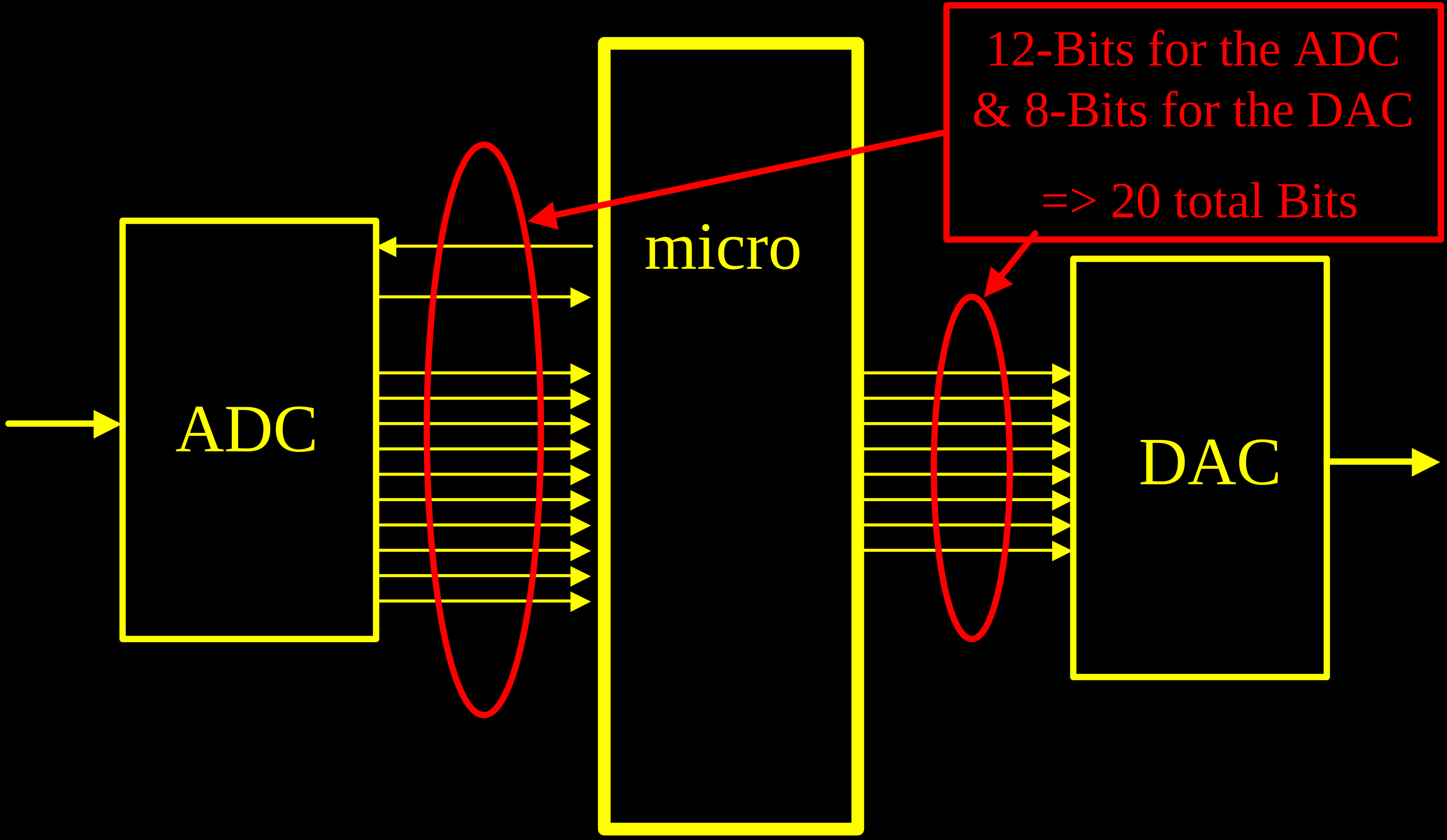
Analog and Digital Converters



Analog and Digital Converter Input and Output Requirements



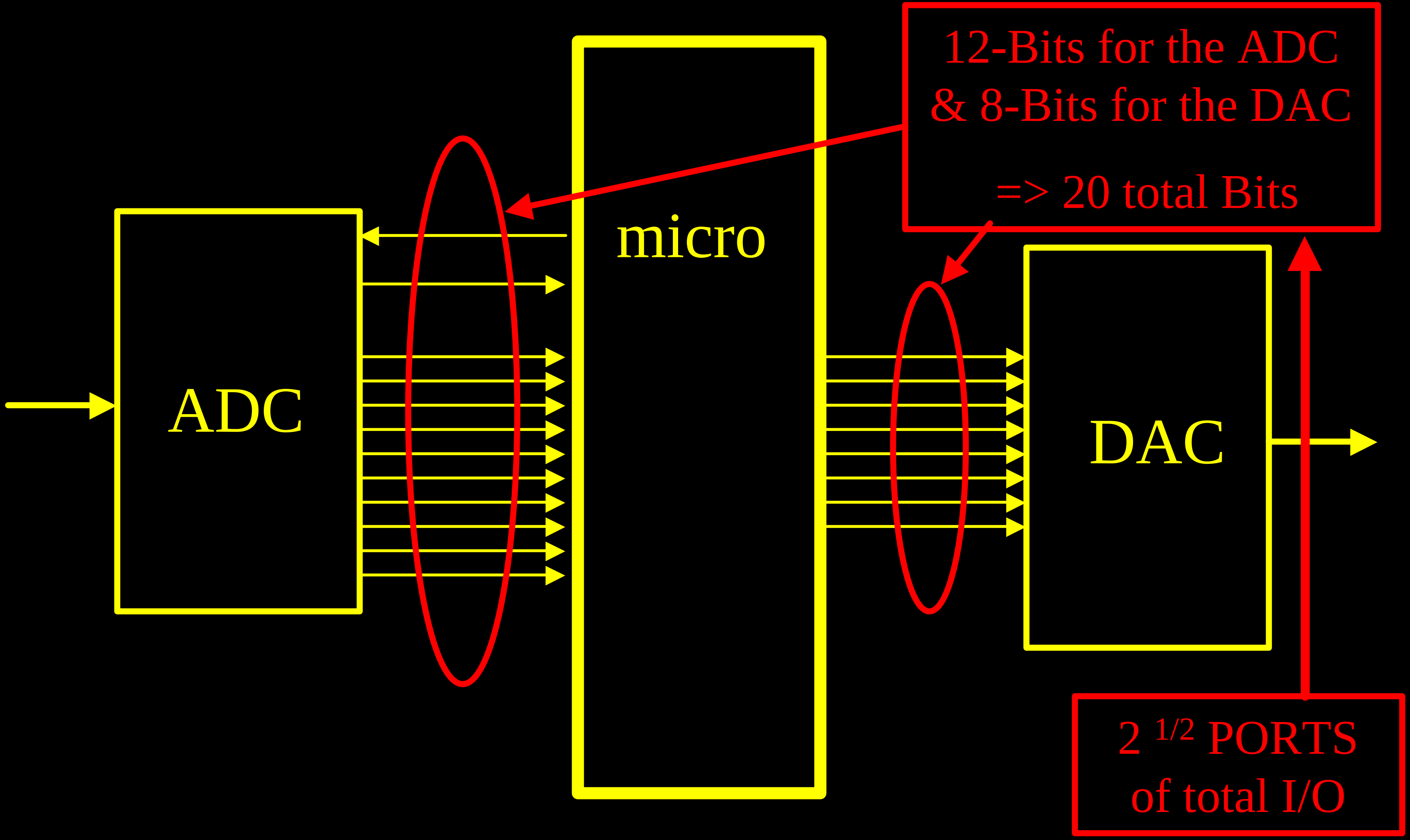
Analog and Digital Converter Input and Output Requirements



12-Bits for the ADC
& 8-Bits for the DAC

=> 20 total Bits

Analog and Digital Converter Input and Output Requirements



How can we reduce the number of I/O port pins required?

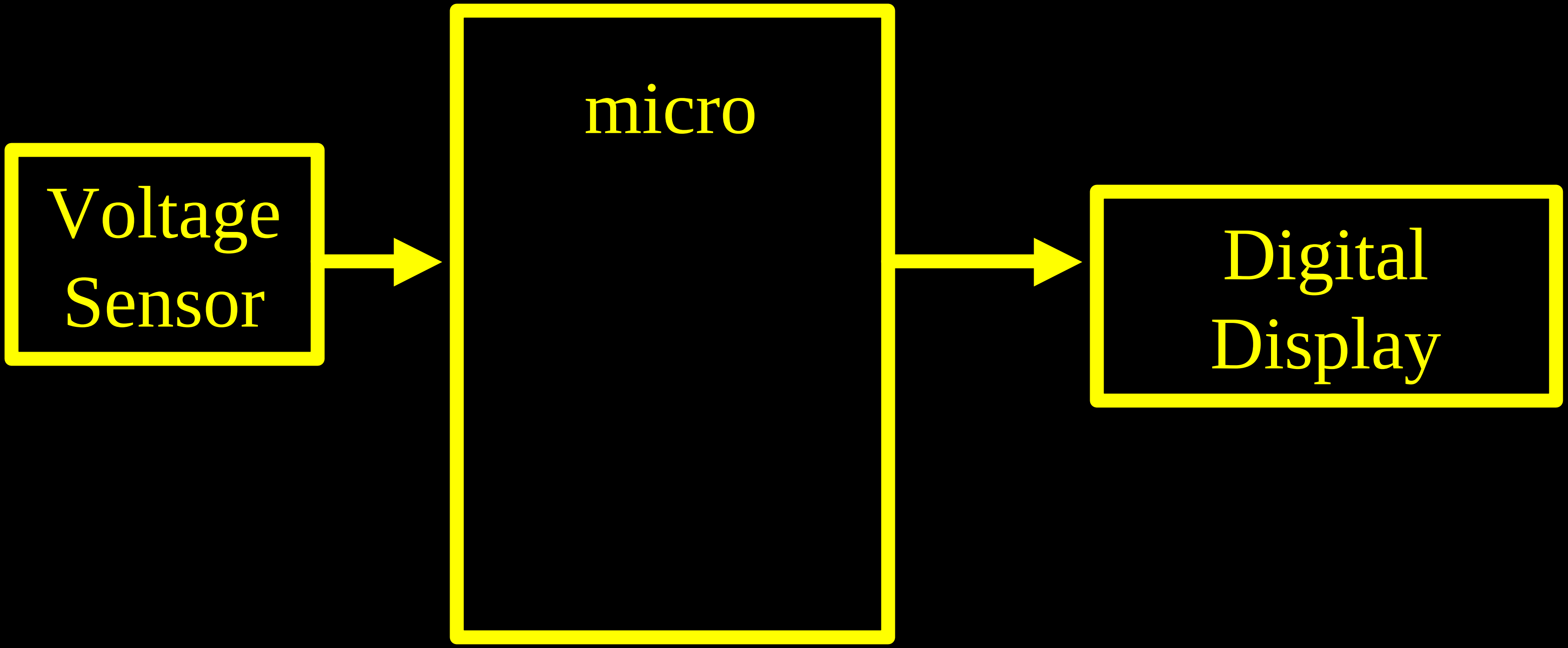
More on Monday

Lab 9

- Create a Digital Voltmeter!!

- That’s pretty straight forward
- Any questions?????

Digital Voltmeter

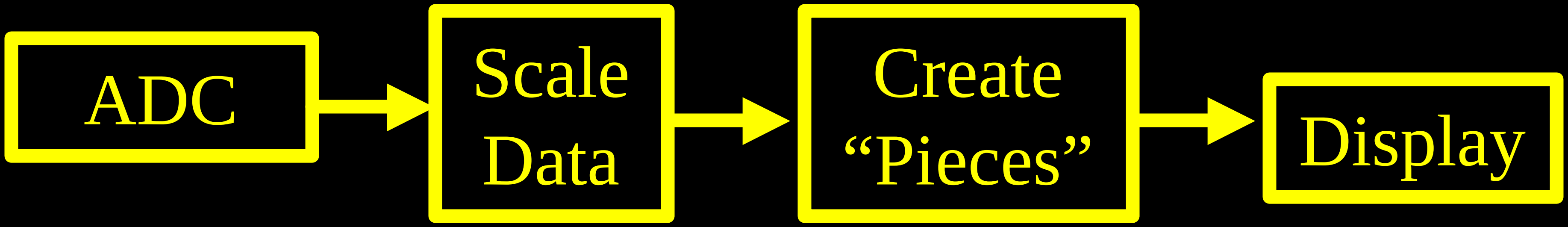


Voltage Sensor

micro

Digital Display

Digital Voltmeter


ADC

Scale Data

Create “Pieces”

Display

Voltmeter

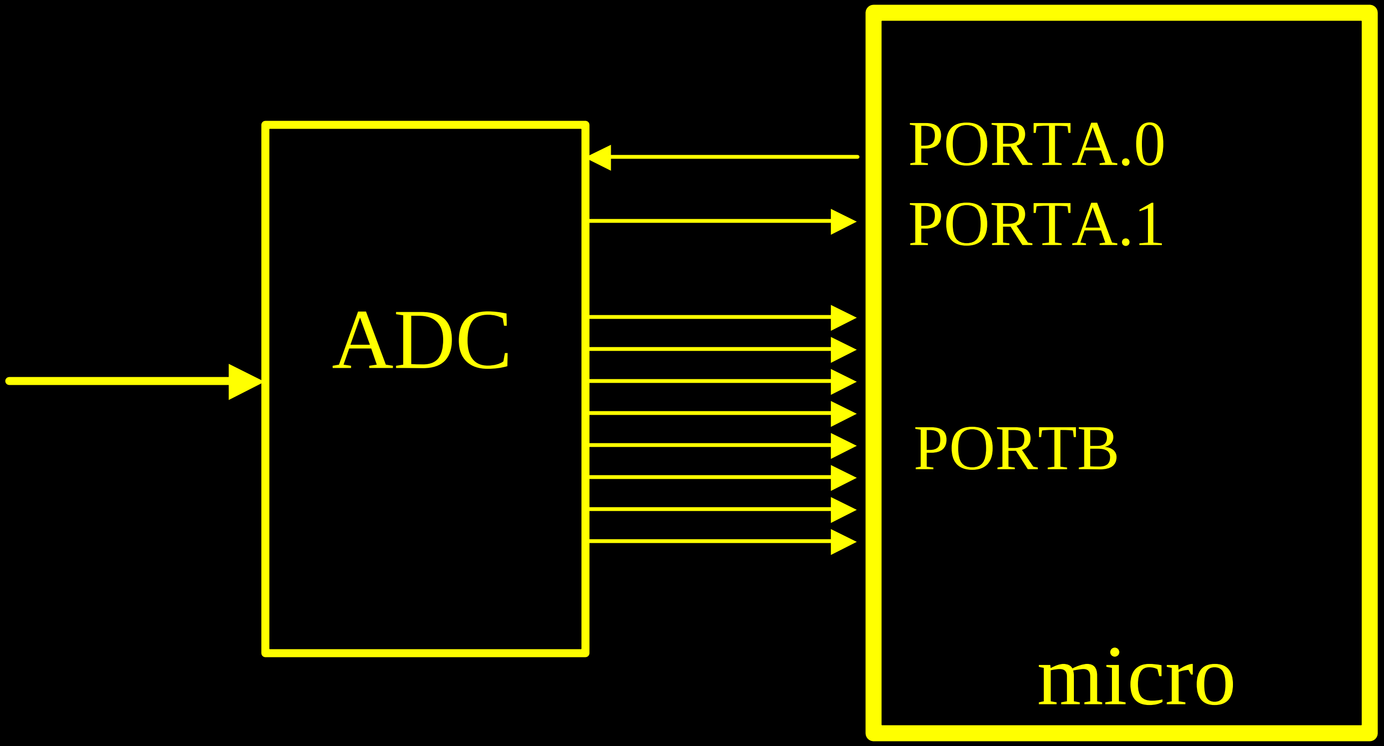
- Break the project into pieces
  - Read the ADC and print to the terminal
  - Scale the data and print to the terminal
  - Separate the voltage into the BCD components
  - Display the BCD components on the 7-Segs

Solution Steps to use the ADC

1. Set the convert line high
2. Wait for Data Ready to go high
3. Clear the convert line
4. Wait for the Ready Line to go low
5. Read the result of the conversion

Group Activity

Given this ADC I/O Diagram:



And These Solution Steps

1. Set the convert line high
2. Wait for Data Ready to go high
3. Clear the convert line
4. Wait for the Ready Line to go low
5. Read the result of the conversion

Provide the Following:

- Initialization Code for the I/O Ports
- Draw the software flowchart
- Write the C code to read the ADC

More on Monday