ADC
(chapter 13)

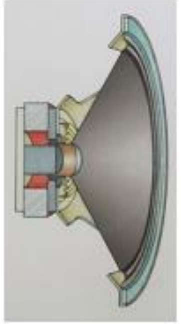
MBSD, 6th Semester

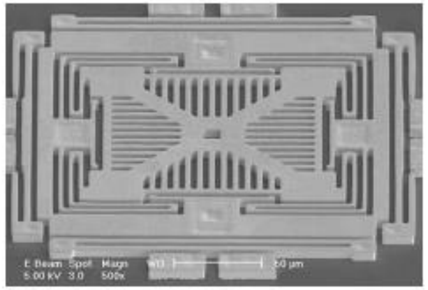
DCSE, UET Peshawar

Bilal Habib

We live in an analog world

- Everything in the physical world is an analog signal
  - Sound, light, temperature, pressure
- Need to convert into electrical signals
  - Transducers: converts one type of energy to another
    - Electro-mechanical, Photonic, Electrical, …
  - Examples
    - Microphone/speaker
    - Thermocouples
    - Accelerometers



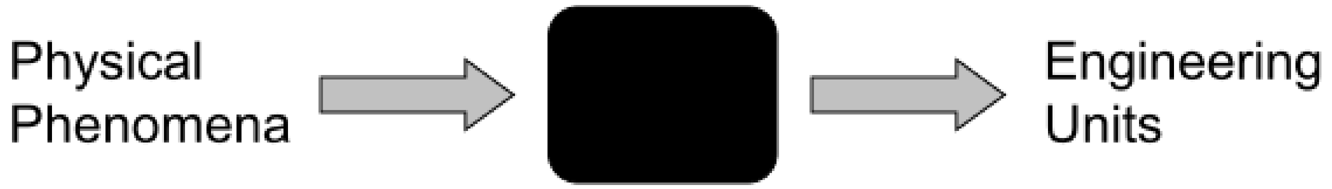


Many other common sensors (some digital)

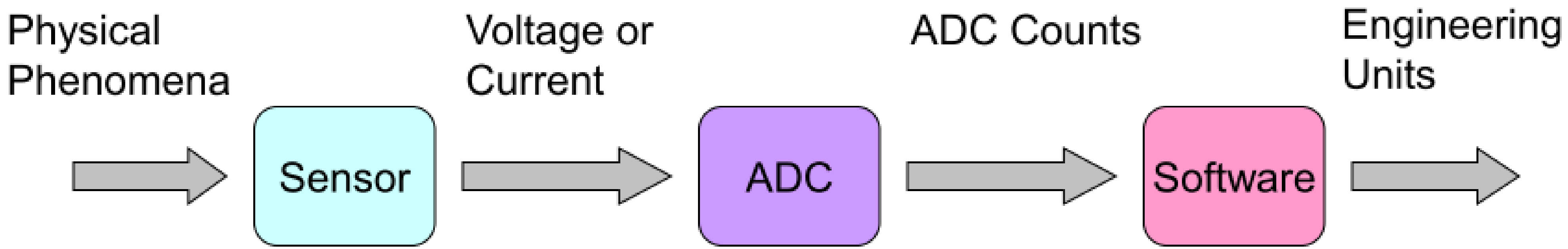
- Force
  - strain gauges - foil, conductive ink
  - conductive rubber
  - rheostatic fluids
    - Piezorestive (needs bridge)
  - piezoelectric films
  - capacitive force
    - Charge source
- Sound
  - Microphones
    - Both current and charge versions
  - Sonar
    - Usually Piezoelectric
- Position
  - microswitches
  - shaft encoders
  - gyros
- Acceleration
  - MEMS
  - Pendulum
- Monitoring
  - Battery-level
    - voltage
  - Motor current
    - Stall/velocity
  - Temperature
    - Voltage/Current Source
- Field
  - Antenna
  - Magnetic
    - Hall effect
    - Flux Gate
- Location
  - Permittivity
  - Dielectric

Going from analog to digital

- What we want



- How we have to get there



INTERFACING TO ADC AND SENSORS

ADC Devices

- ADCs (analog-to-digital converters) are among the most widely used devices for data acquisition
  - A physical quantity, like temperature, pressure, humidity, and velocity, etc., is converted to electrical (voltage, current) signals using a device called a *transducer*, or *sensor*
- We need an analog-to-digital converter to translate the analog signals to digital numbers, so microcontroller can read them

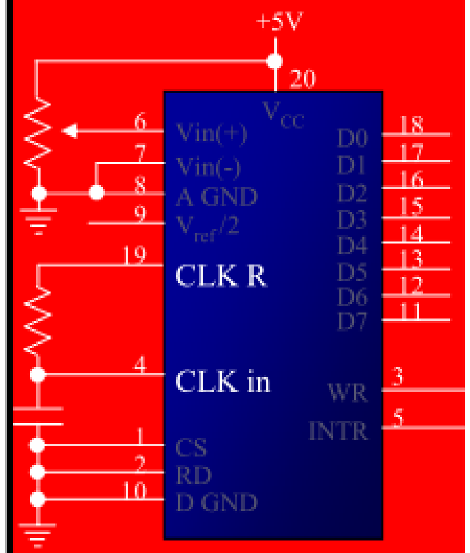
INTERFACING TO ADC AND SENSORS

ADC804 Chip

- ADC804 IC is an analog-to-digital converter
  - It works with +5 volts and has a resolution of 8 bits
  - *Conversion time* is another major factor in judging an ADC
    - Conversion time is defined as the time it takes the ADC to convert the analog input to a digital (binary) number
    - In ADC804 conversion time varies depending on the clocking signals applied to CLK R and CLK IN pins, but it cannot be faster than 110 μs

INTERFACING TO ADC AND SENSORS

ADC804 Chip (cont')



CLK IN and CLK R

- CLK IN is an input pin connected to an external clock source
- To use the internal clock generator (also called self-clocking), CLK IN and CLK R pins are connected to a capacitor and a resistor, and the clock frequency is determined by

$$f = \frac{1}{1.1RC}$$

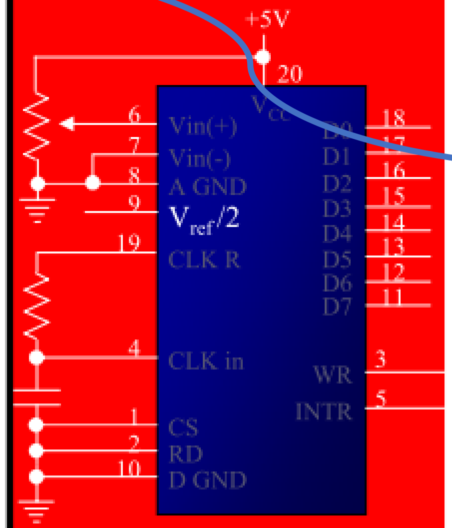
  - Typical values are R = 10K ohms and C = 150 pF
  - We get f = 606 kHz and the conversion time is 110 μs

INTERFACING TO ADC AND SENSORS

ADC804 Chip (cont')

- $V_{ref}/2$
  - It is used for the reference voltage
    - If this pin is open (not connected), the analog input voltage is in the range of 0 to 5 volts (the same as the Vcc pin)
    - If the analog input range needs to be 0 to 4 volts, $V_{ref}/2$ is connected to 2 volts

Vref/2 = 2
=> Vref = 4V



$V_{ref}/2$ Relation to V_{in} Range

$V_{ref}/2$(v)	V_{in}(V)	Step Size (mV)
Not connected*	0 to 5	5/256=19.53
2.0	0 to 4	4/255=15.62
1.5	0 to 3	3/256=11.71
1.28	0 to 2.56	2.56/256=10
1.0	0 to 2	2/256=7.81
0.5	0 to 1	1/256=3.90

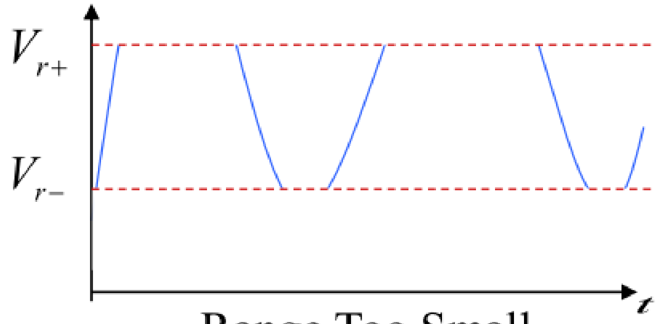
Step size is the smallest change can be discerned by an ADC

Should be 256

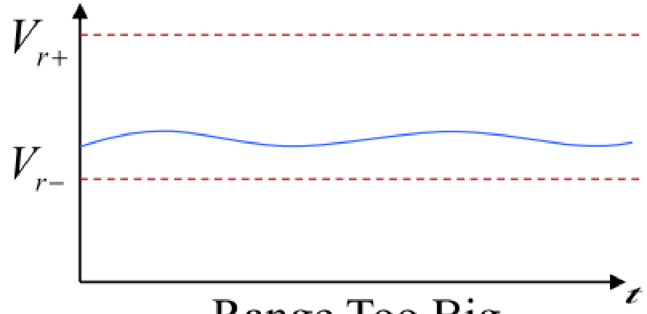
Choosing the range

- Fixed # of bits (e.g. 8-bit ADC)
- Span a particular input voltage range
- What do the sample values represent?
  - Some fraction within the range of values

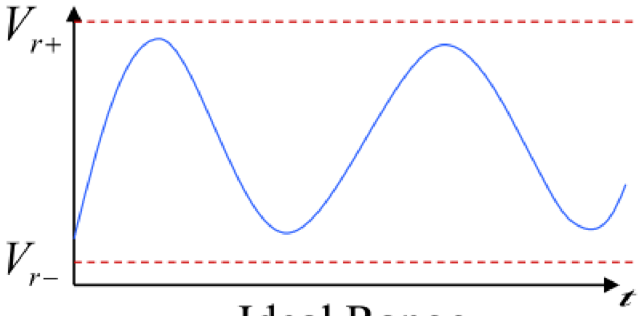
    → ***What range to use?***



Range Too Small



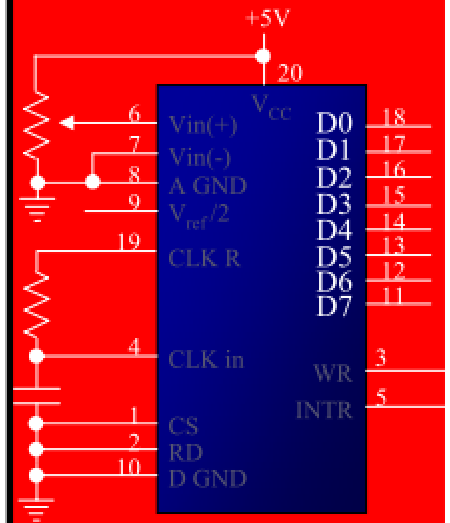
Range Too Big



Ideal Range

INTERFACING TO ADC AND SENSORS

ADC804 Chip (cont')



- D0-D7
  - The digital data output pins
  - These are tri-state buffered
    - The converted data is accessed only when CS = 0 and RD is forced low
  - To calculate the output voltage, use the following formula

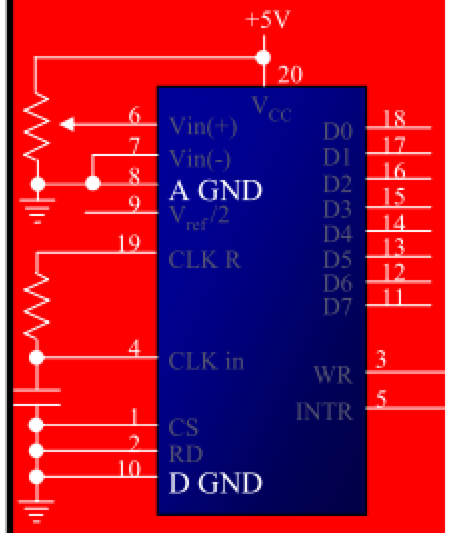
$$D_{out} = \frac{V_{in}}{step\ size}$$

5V/255 = step size

- *Dout* = digital data output (in decimal),
- *Vin* = analog voltage, and
- *step size* (resolution) is the smallest change

INTERFACING TO ADC AND SENSORS

ADC804 Chip (cont')

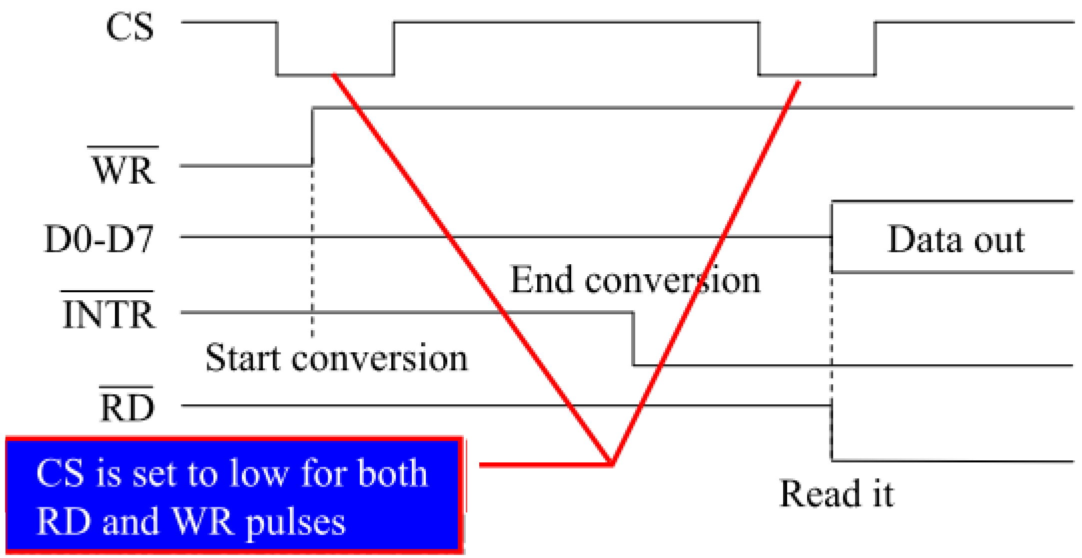


- Analog ground and digital ground
  - Analog ground is connected to the ground of the analog V_{in}
  - Digital ground is connected to the ground of the V_{cc} pin
- To isolate the analog V_{in} signal from transient voltages caused by digital switching of the output D0 – D7
  - This contributes to the accuracy of the digital data output

INTERFACING TO ADC AND SENSORS

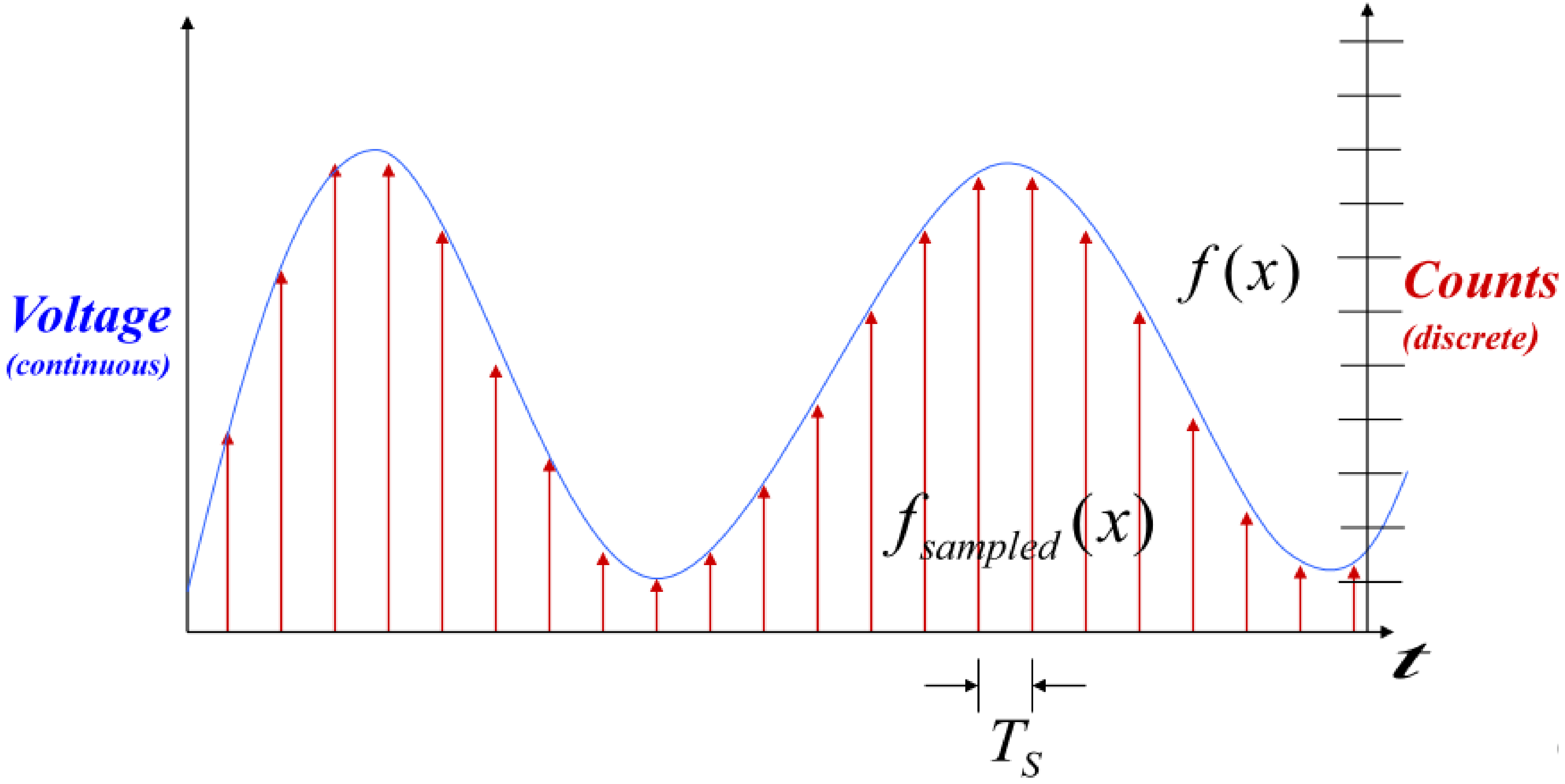
ADC804 Chip (cont')

- The following steps must be followed for data conversion by the ADC804 chip
  - Make CS = 0 and send a low-to-high pulse to pin WR to start conversion
  - Keep monitoring the INTR pin
    - If INTR is low, the conversion is finished
    - If the INTR is high, keep polling until it goes low
  - After the INTR has become low, we make CS = 0 and send a high-to-low pulse to the RD pin to get the data out of the ADC804



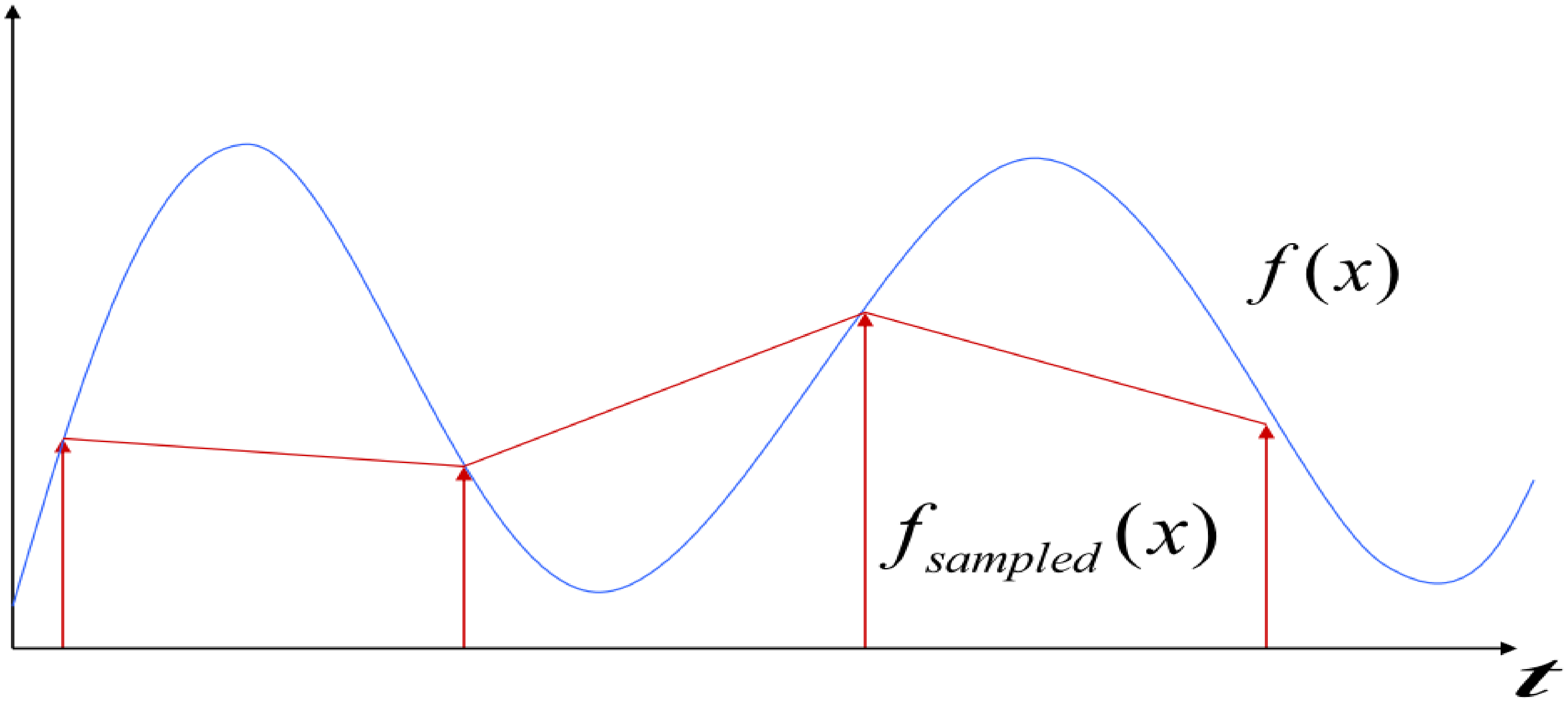
Representing an analog signal digitally

- How do we represent an analog signal (e.g. continuous voltage)?
  - As a time series of discrete values
    → On MCU: read ADC data register (counts) periodically (T_s)



Choosing the sample rate

- What sample rate do we need?
  - Too little: we can't reconstruct the signal we care about
  - Too much: waste computation, energy, resources



Shannon-Nyquist sampling theorem

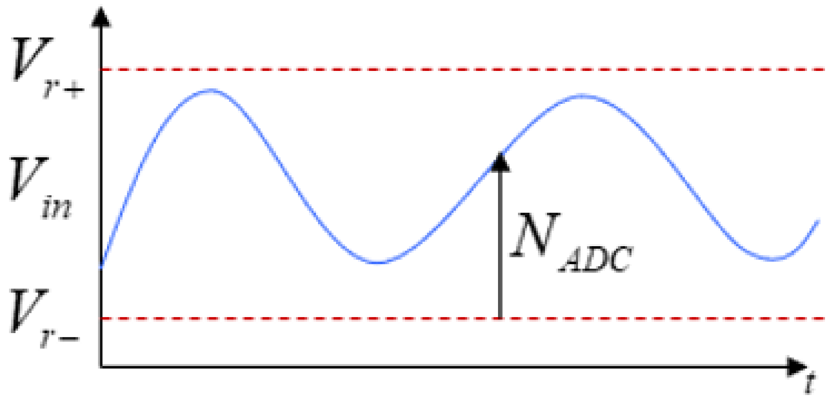
- *If a continuous-time signal* $f(x)$ *contains no frequencies higher than* $f_{\max}$*, it can be completely determined by discrete samples taken at a rate:*

$$f_{\text{samples}} > 2 f_{\max}$$

- Example:
  - Humans can process audio signals 20 Hz – 20 KHz
  - Audio CDs: sampled at 44.1 KHz

Converting between voltages, ADC counts, and engineering units

- Converting: ADC counts ⇔ Voltage



$$Dout = \frac{Vin}{Step\_size}$$

$$Step\_size = \text{Range}/(2^8 - 1)$$

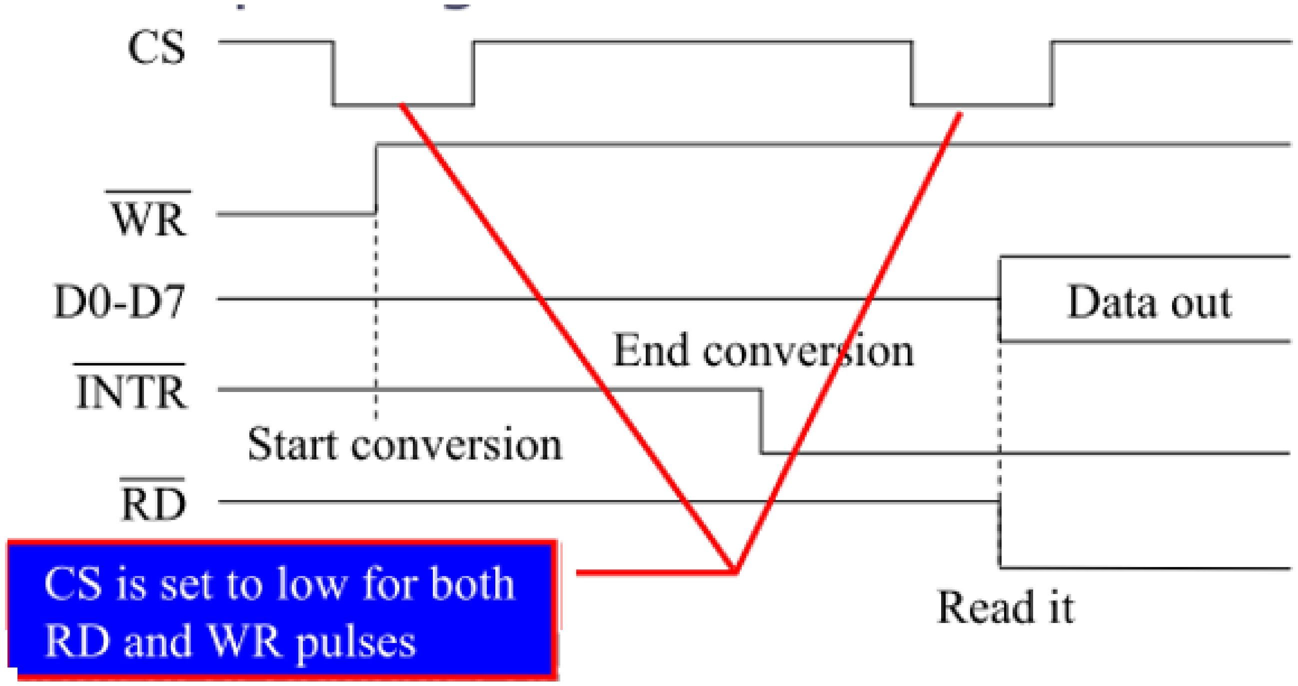
$$Range = V_{r+} - V_{r-}$$

Example:
ADC is 8bit
Dout = N_{ADC} = ?,
if Vin = 2.5V, V_{r+} = 5V, V_{r-} = 0V.

Step size = 5/256=19.53mV

Dout = 2.5V/19.53mV = 128

Interfacing ADC 0804




```
while(1)
{

 RD_n = 1;
 WR_n = 0;
 WR_n = 1;
 while(INTR==1);
 RD_n = 0;
 writecmd(0x2);   // Bring cursor to home
 display_ADC_result();
 delay(100);

}
```

Start conversion

Wait till conversion is done

Interfacing ADC 0804