



University of
Nottingham

UK | CHINA | MALAYSIA

Lecture 6

Data conversion: ADC

Mechatronics

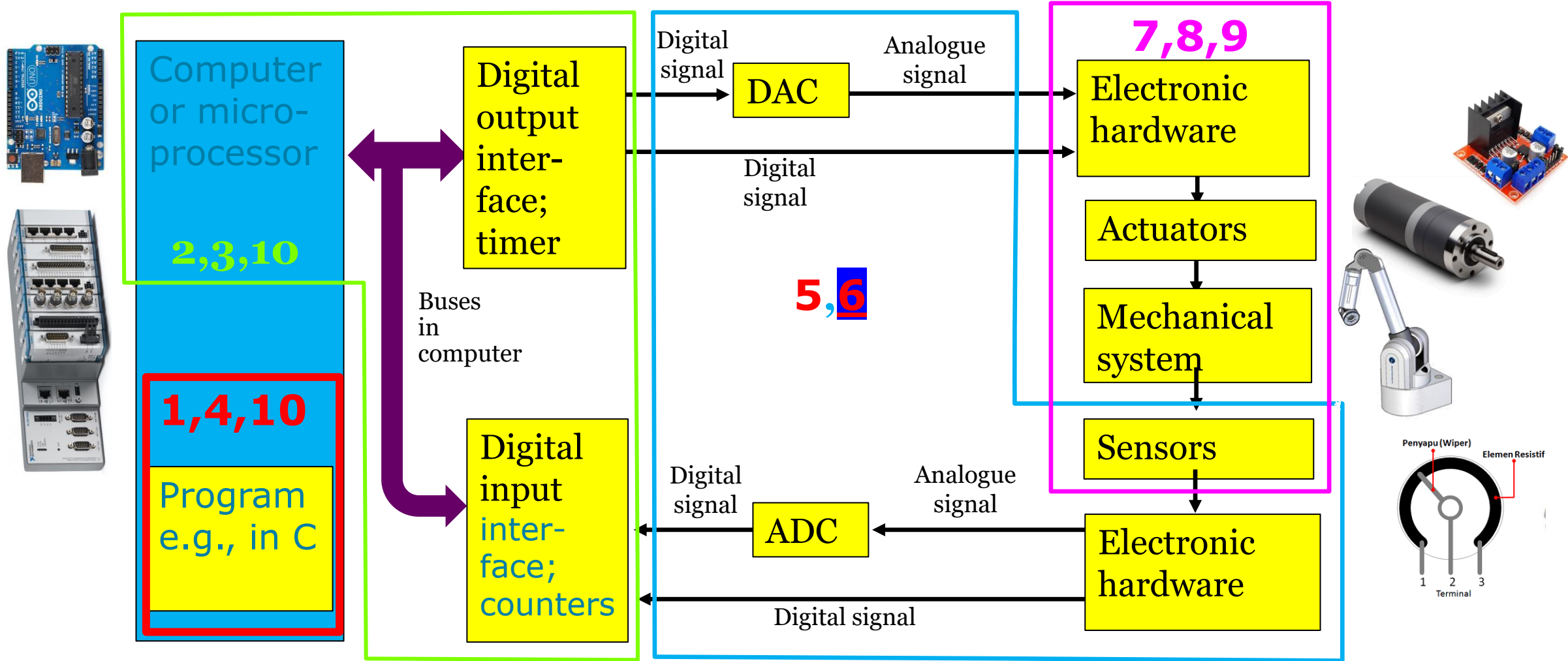
MMME3085

Module Convenor – Abdelkhalick Mohammad



- To revise issues relating to conversion between analogue and digital formats
- To introduce additional data conversion techniques notably:
 - Successive approximation ADCs
 - Flash ADC
 - Delta/Sigma ADCs
- Sampling and Aliasing problem with ADC

A typical Mechatronics System





University of
Nottingham

UK | CHINA | MALAYSIA

Recap



So far, we learned ...

- How to deal with digital signals including train of pulses
 - Generate digital signal
 - Read digital signal
- Timer/Counters as a hardware solution
- Registers in μp
- State Tables
- Finite State Machines
- Interrupt
- DAC and the associated issues with Digital and Analoge signals



University of
Nottingham

UK | CHINA | MALAYSIA

Introduction



- Recall that we often need to collect data from **analogue** devices within a control loop:
 - Position (e.g. Potentiometer or LVDT)
 - Temperature (from thermocouple)
- Need to convert these signals to digital form
- But digital-to-analog conversion typically requires us firstly to be able to convert digital to analogue.



University of
Nottingham

UK | CHINA | MALAYSIA

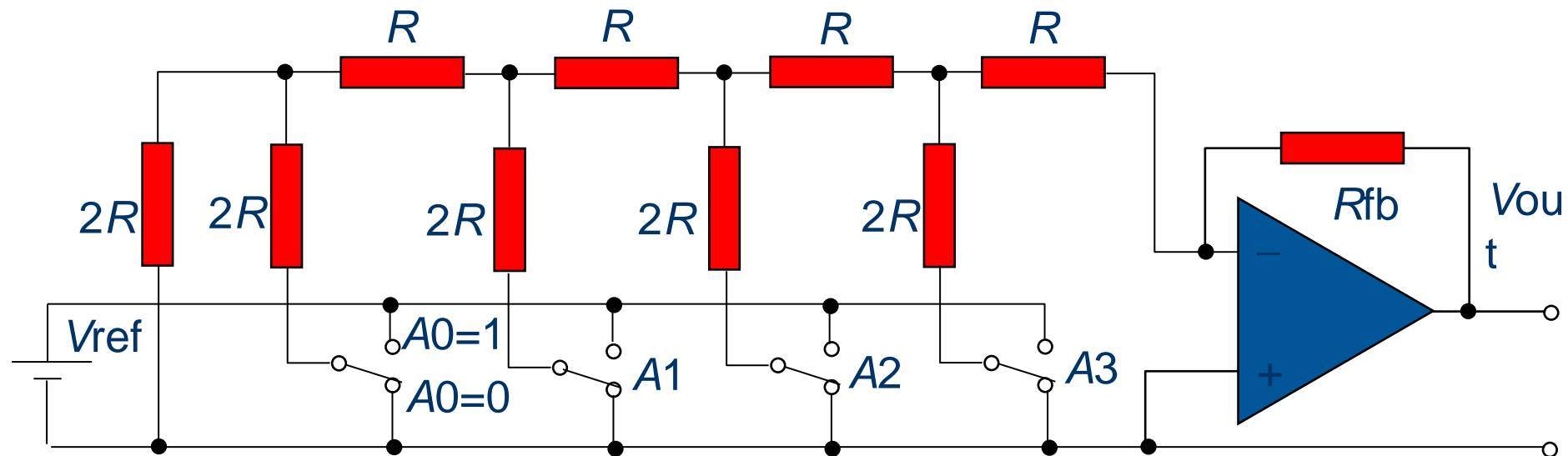
DAC, Revision



- These take **digital data** from a buffer, on a number (e.g. 8) of parallel wires and produces an **analogue output proportional** to the numerical value of the binary data.



R-2R Ladder DAC

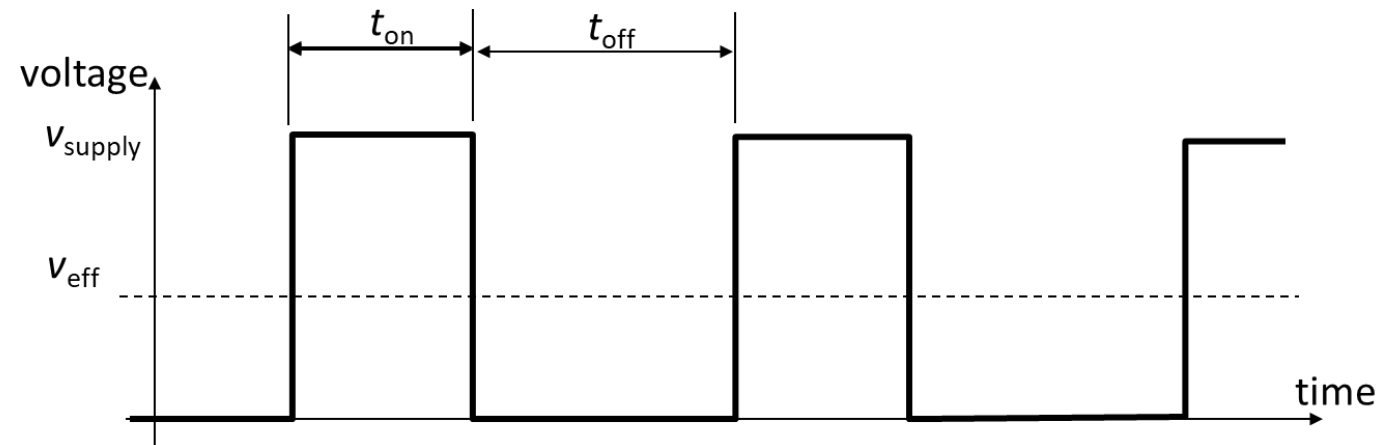


In this case:
$$V_{out} = -V_{ref} \frac{R_{fb}}{4R} \left(\frac{A_0}{8} + \frac{A_1}{4} + \frac{A_2}{2} + \frac{A_3}{1} \right)$$



- Approximation to DAC
- 1-bit DA;
- Train of pulses of constant frequency but varying width
- Vary the “duty cycle” of a pulse train to obtain a varying effective voltage

$$V_{\text{eff}} = V_{\text{supply}} \frac{t_{\text{on}}}{t_{\text{off}} + t_{\text{on}}}$$





University of
Nottingham

UK | CHINA | MALAYSIA

ADC



- Converting digital data to analogue quantity is **straightforward**: just a matter of operating switches in a circuit
- But going the other way needs some “**intelligence**” and “**decision making**”
- Typically make use of the **digital-to-analogue** converters already described by means of a “**trial and error**” comparison between the true signal value and the digital estimate



- These comparisons are essential to the operation of the ADC and can take place:
 - Either one comparison after another
 - Or lots of comparisons at once
- Both of these approaches covered briefly last year – will cover in more depth
- New types: successive approximation, delta-sigma



University of
Nottingham

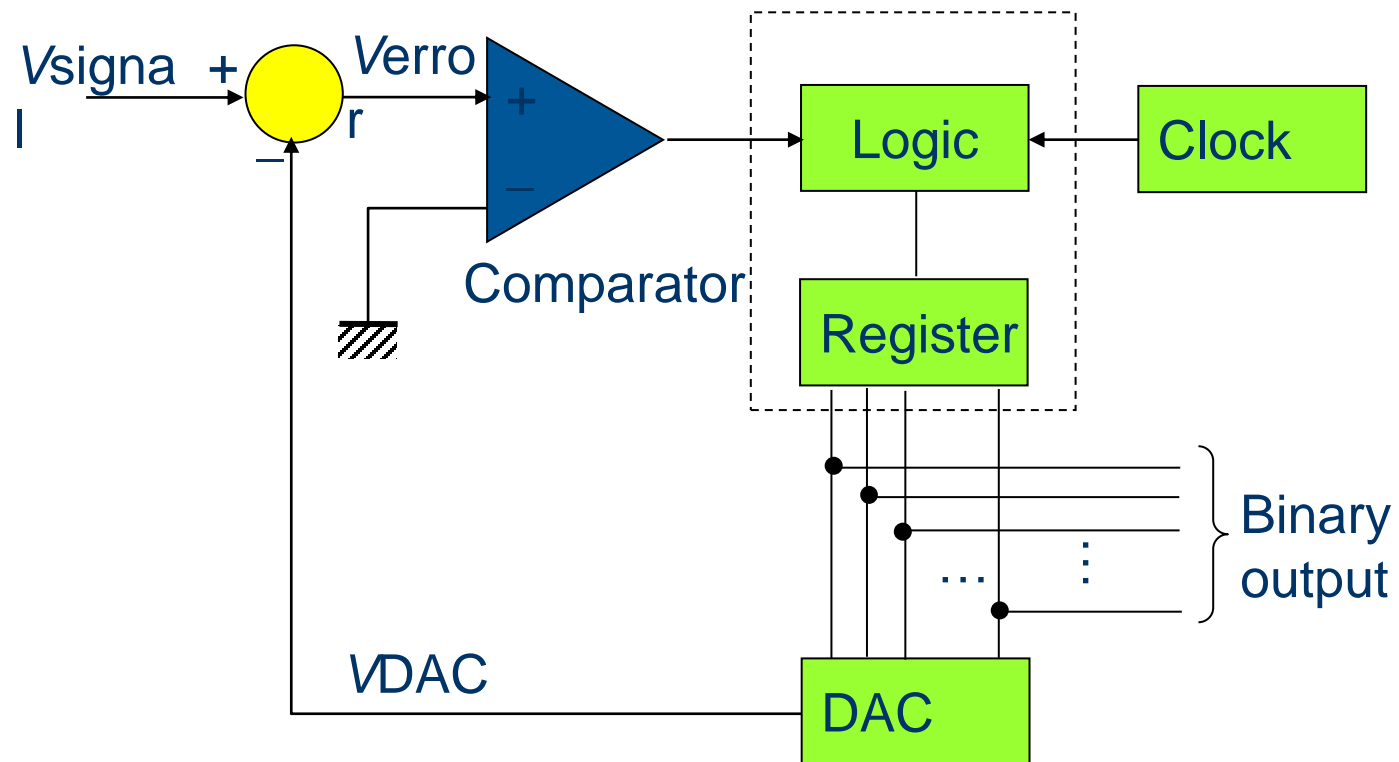
UK | CHINA | MALAYSIA

ADC

Successive Approximation ADC

Successive approximation ADC

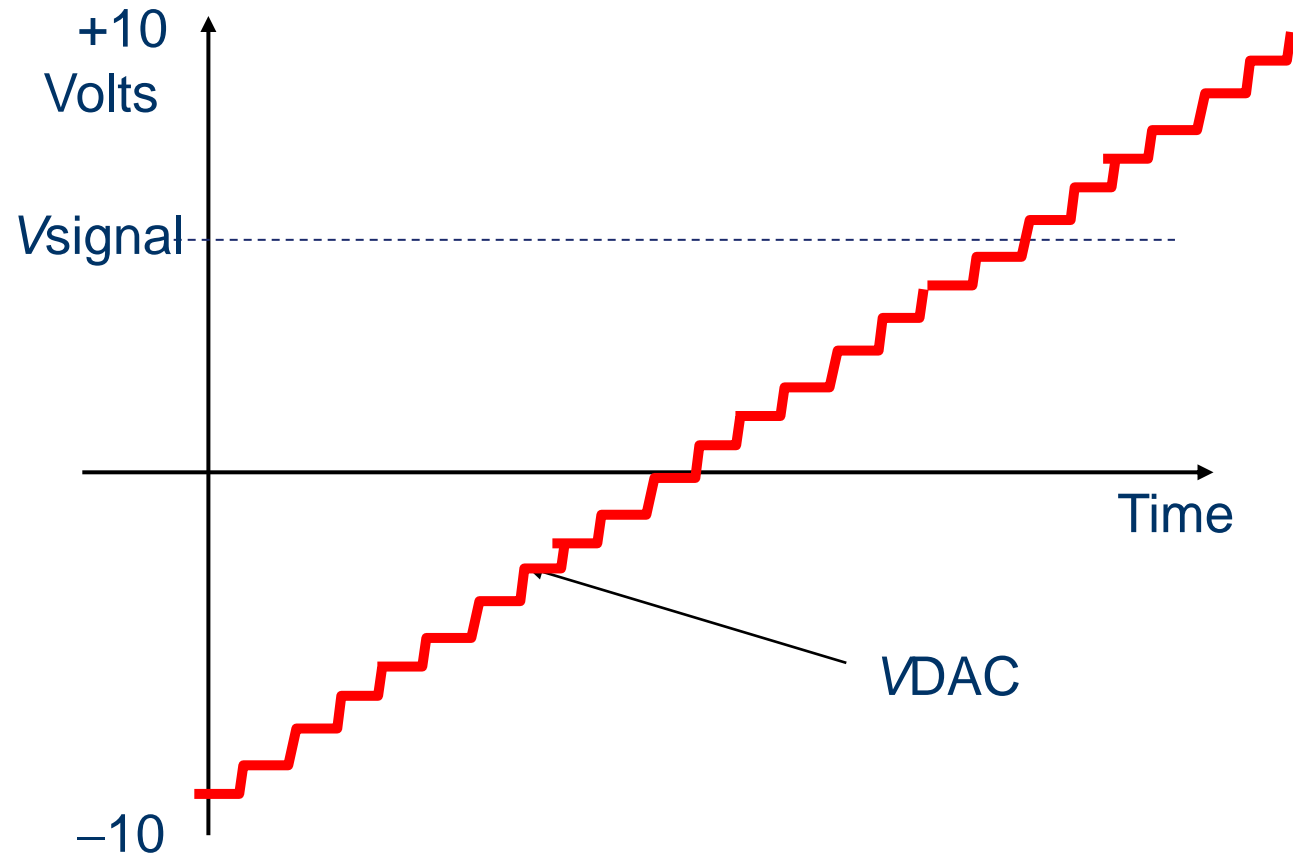
- Compare output of a DAC with the voltage to be digitised



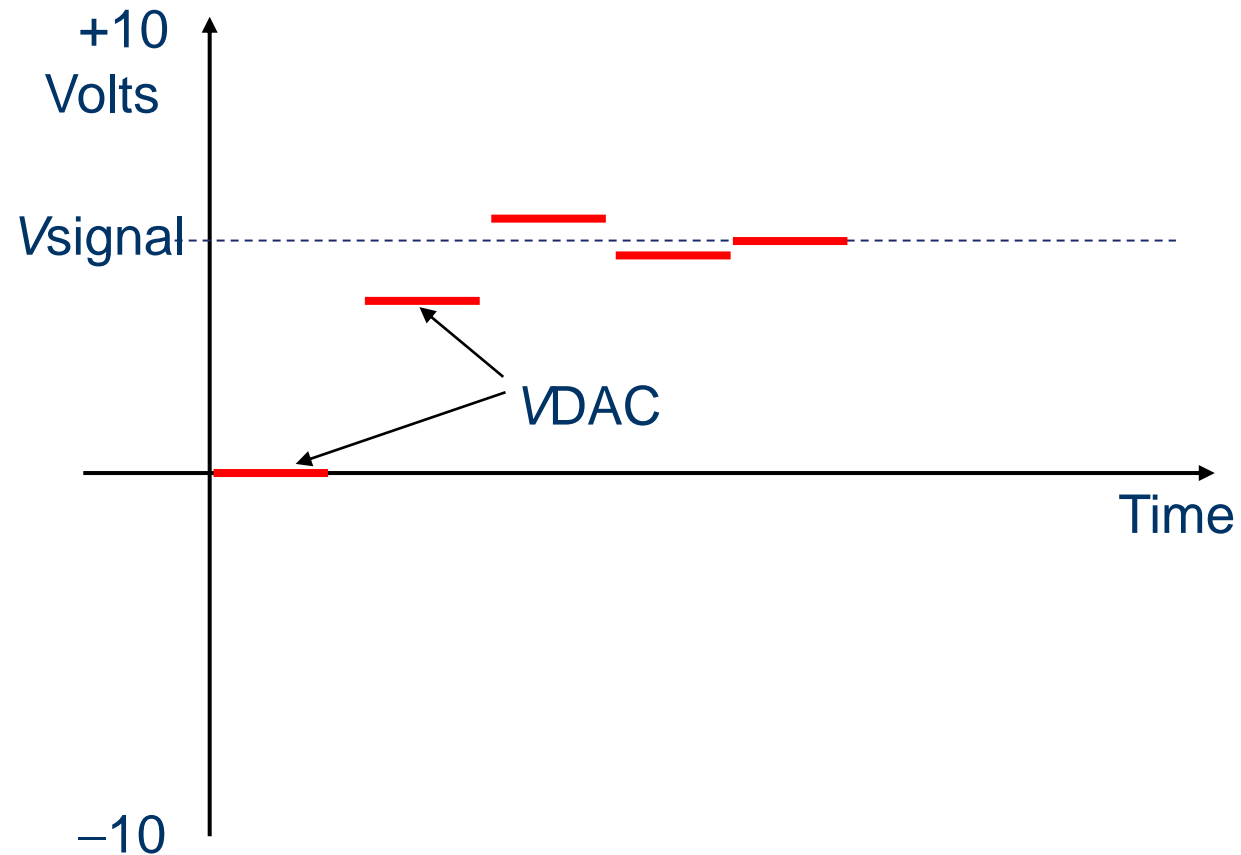


Successive approximation ADC

- Explores every ADC value until finds correct one



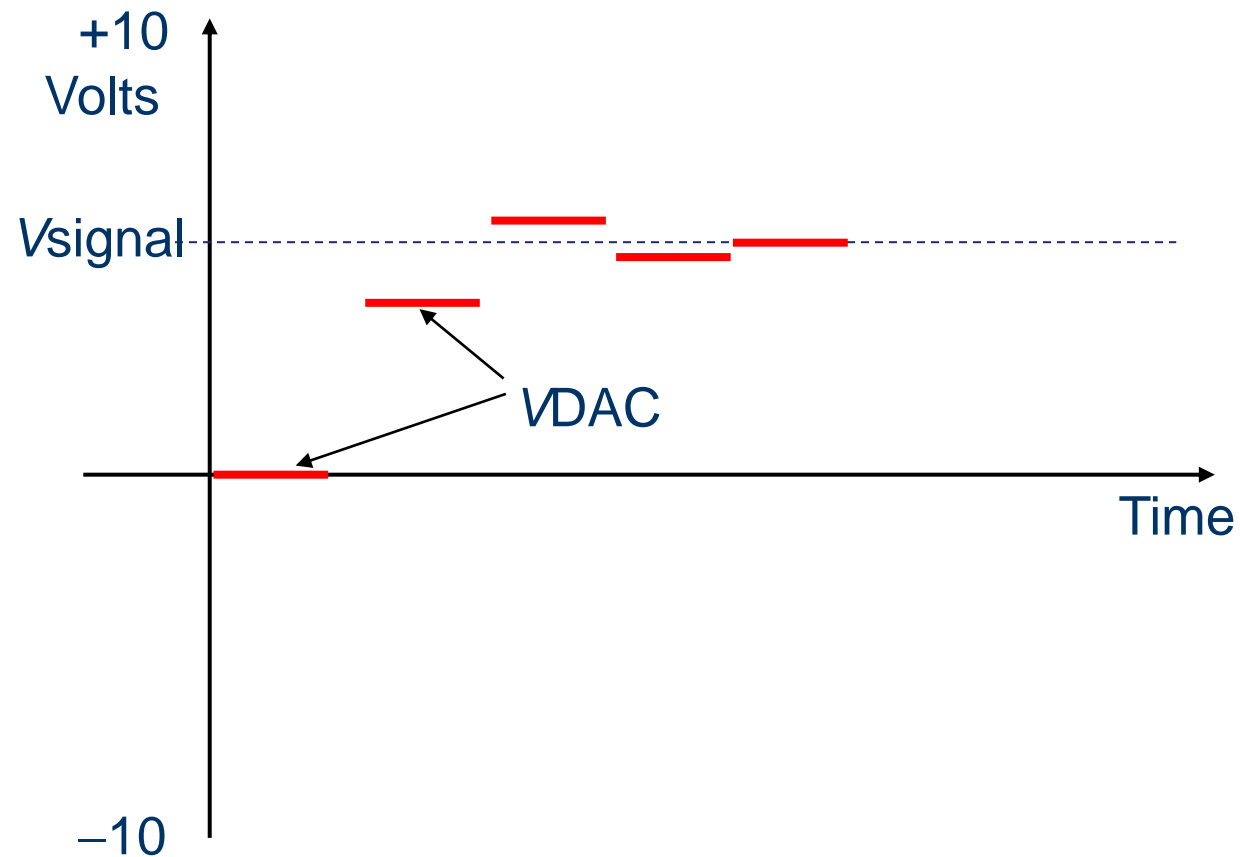
- Much more practical and fast – subdivides range for each iteration





Successive approximation ADC

- Converges rapidly to correct value



Successive approximation ADC

- General purpose, reasonably rapid
- This is the basis of the analogue input on the Arduino



arduino.cc

Successive approximation ADC

- Conversion time on Arduino is 13 clock cycles so around $1 \mu\text{s}$ – fast, but not suitable for above a few hundred kHz



arduino.cc



University of
Nottingham

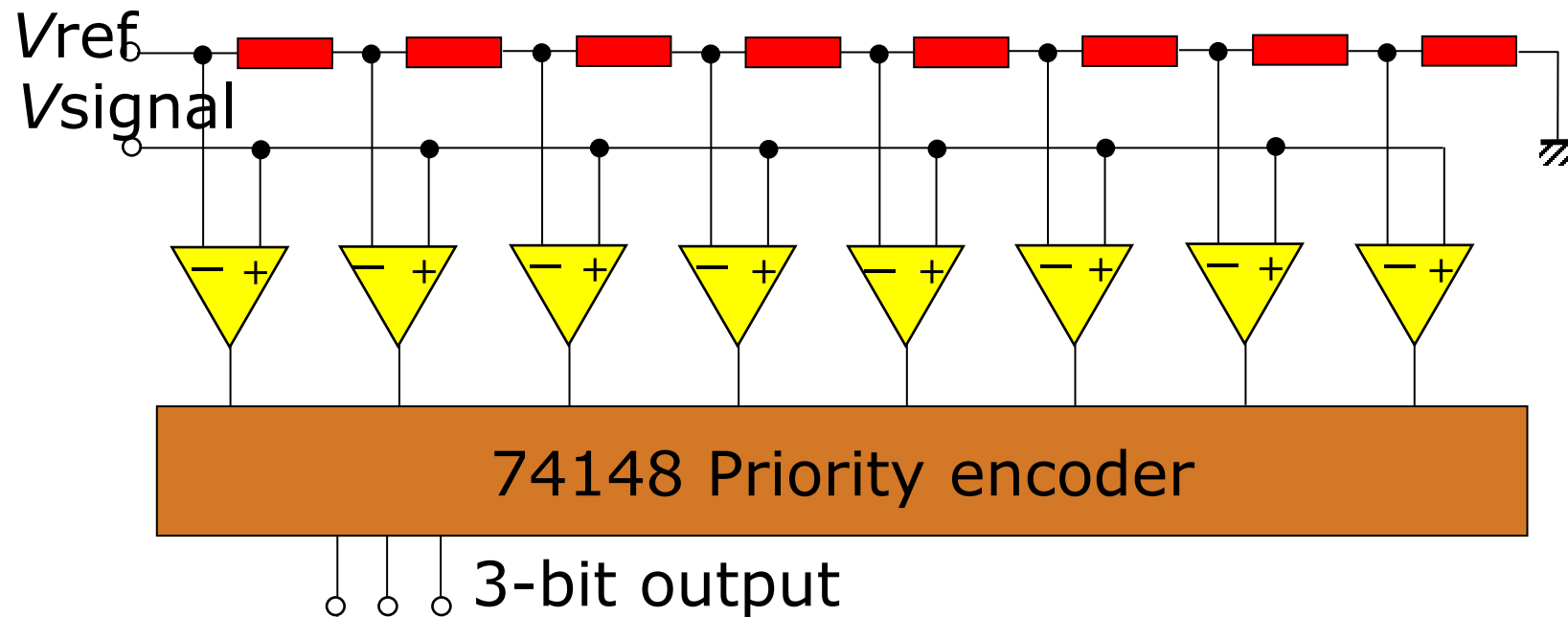
UK | CHINA | MALAYSIA

ADC

Flash ADC

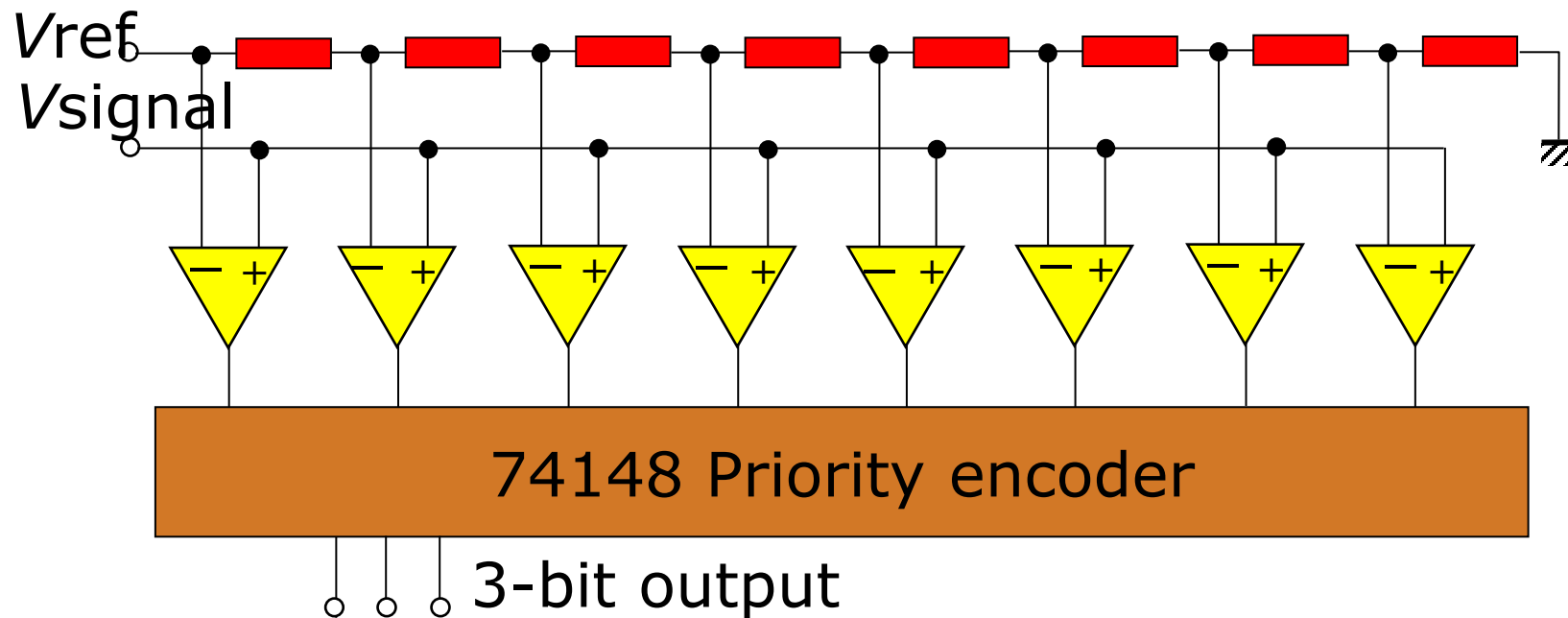
Flash ADC

- Used for VERY rapid signal capture
- High speed at cost of low resolution
- A large number of comparators: 1 per **incr't**



Flash ADC

- Advantage: very fast (conversion time 4-1000 ns i.e. as fast as 1/250 000 000 sec or better)
- Disadvantage: expensive, esp. for high resolu





University of
Nottingham

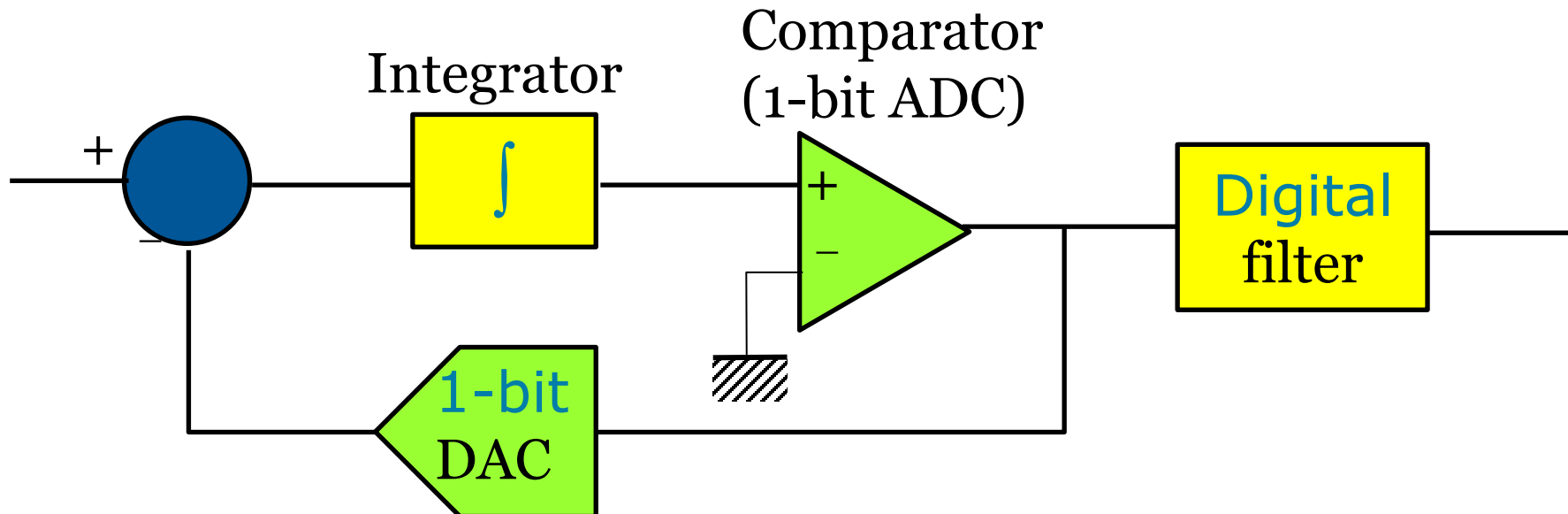
UK | CHINA | MALAYSIA

ADC

Delta-Sigma ADC

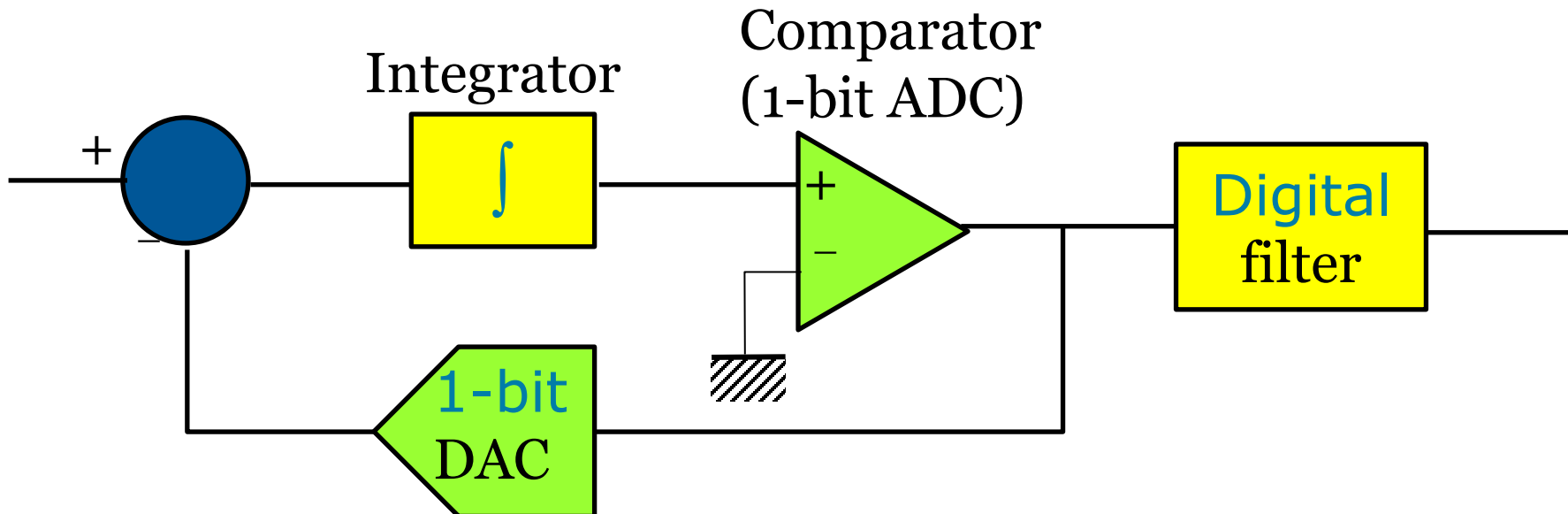
Delta-Sigma ADC (aka Sigma-Delta)

- Used today for high-resolution ADCs
- Variant on “trial and error” approach but using a 1-bit DAC (a bit like PWM)

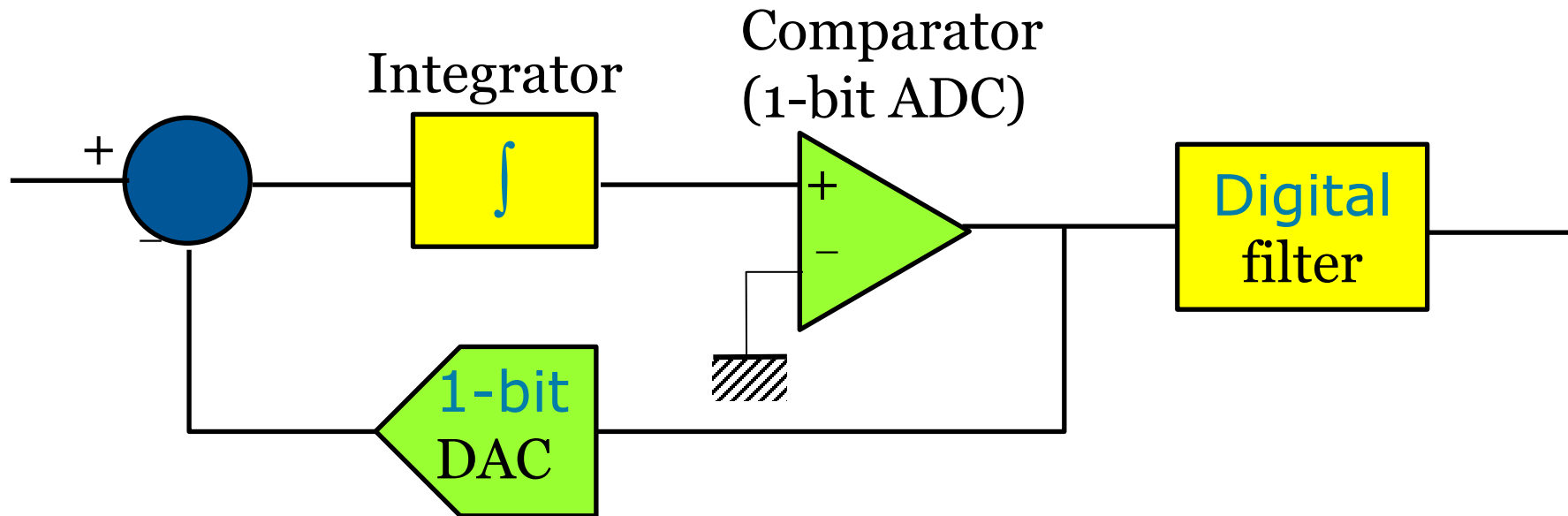


Delta-Sigma ADC (aka Sigma-Delta)

- 1-bit DAC creates stream of hi & lo outputs
- Subtracted from input
- Difference alternates rapidly but is averaged



- Comparator used to determine value to DAC
- Rapidly varying value is smoothed (filtered)
- Output is a smooth, slowly-varying numerical value





Delta-Sigma ADC (aka Sigma-Delta)

- Can be high resolution e.g. 24-bit but not fast
- 24-bit, up to 100 samples per second
- 1 part in 16 million!





University of
Nottingham

UK | CHINA | MALAYSIA

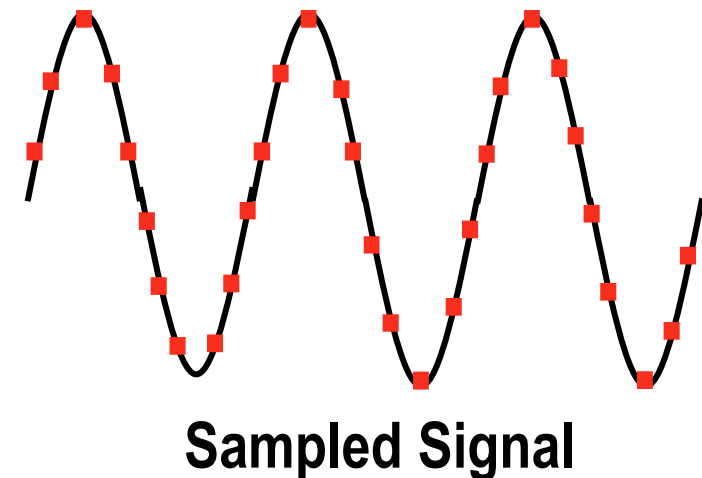
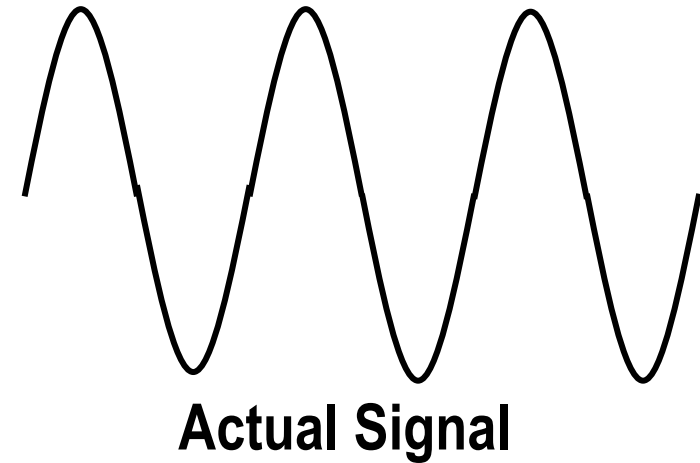
ADC

Sampling and Aliasing



Sampling Considerations

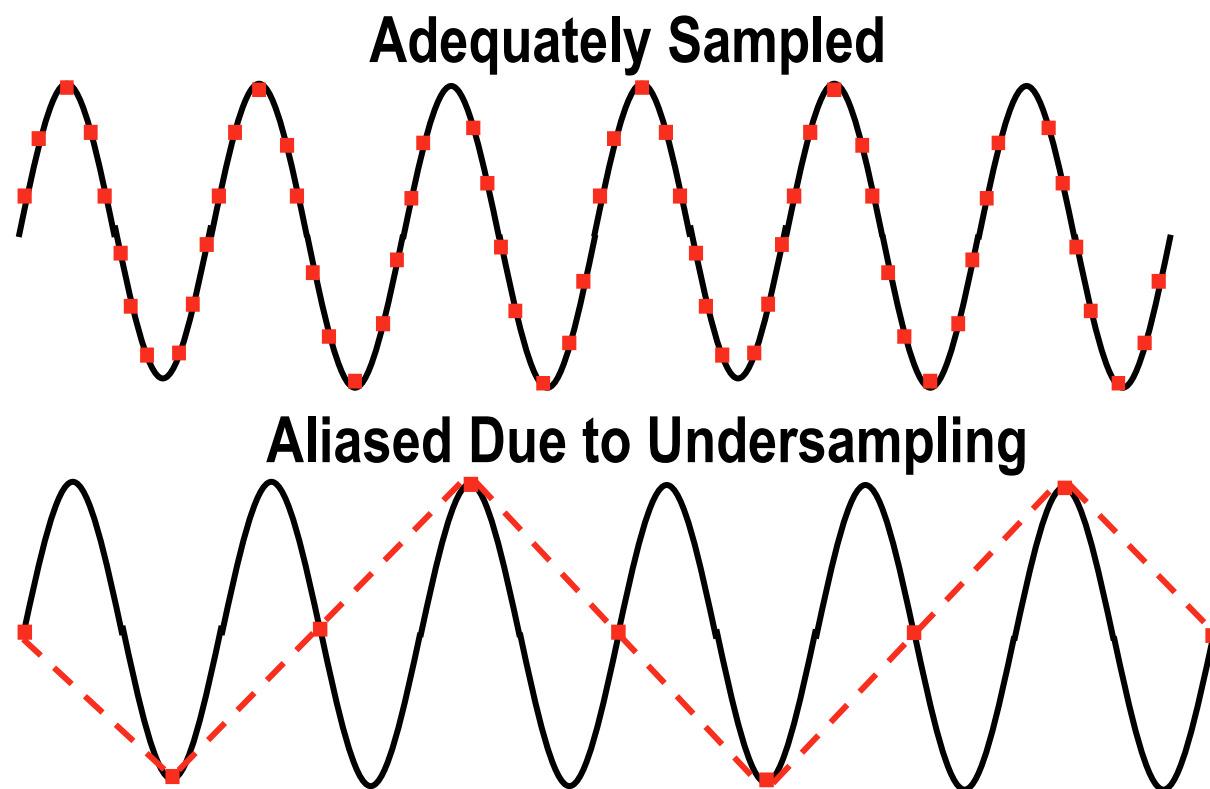
- Actual analog input signal is continuous with respect to time
- Sampled signal is series of discrete samples acquired at a specified sampling rate
- The faster we sample, the more our sampled signal will look like our actual signal
- If not sampled fast enough, a problem known as aliasing will occur





Aliasing

- Sample rate – how often an A/D conversion takes place
- Alias – misrepresentation of a signal

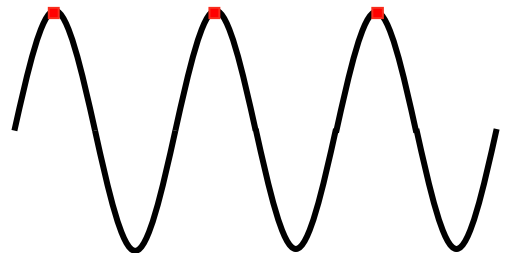


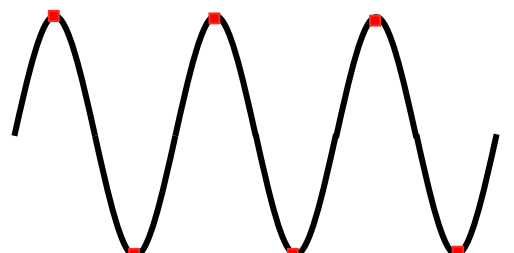

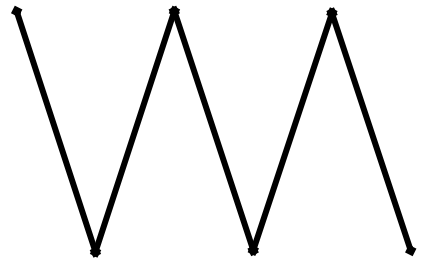
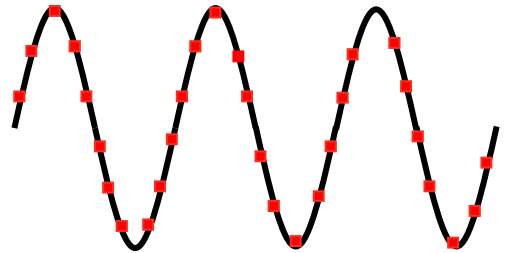

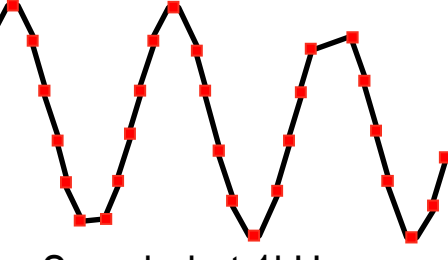




- You must sample at **greater** than **2 times** the **maximum** frequency component of your signal to accurately represent the **FREQUENCY** of your signal.
- **NOTE:** You must sample between 5–10 times greater than the maximum frequency component of your signal to accurately represent the **SHAPE** of your signal.



Nyquist Example

| | | |
|---|--|---|
|  <p>100Hz Sine Wave</p> |   <p>Sampled at 100Hz</p> | <p>Aliased Signal</p> |
|  <p>100Hz Sine Wave</p> |   <p>Sampled at 200Hz</p> | <p>Adequately Sampled for Frequency Only (Same # of cycles)</p> |
|  <p>100Hz Sine Wave</p> |   <p>Sampled at 1kHz</p> | <p>Adequately Sampled for Frequency and Shape</p> |

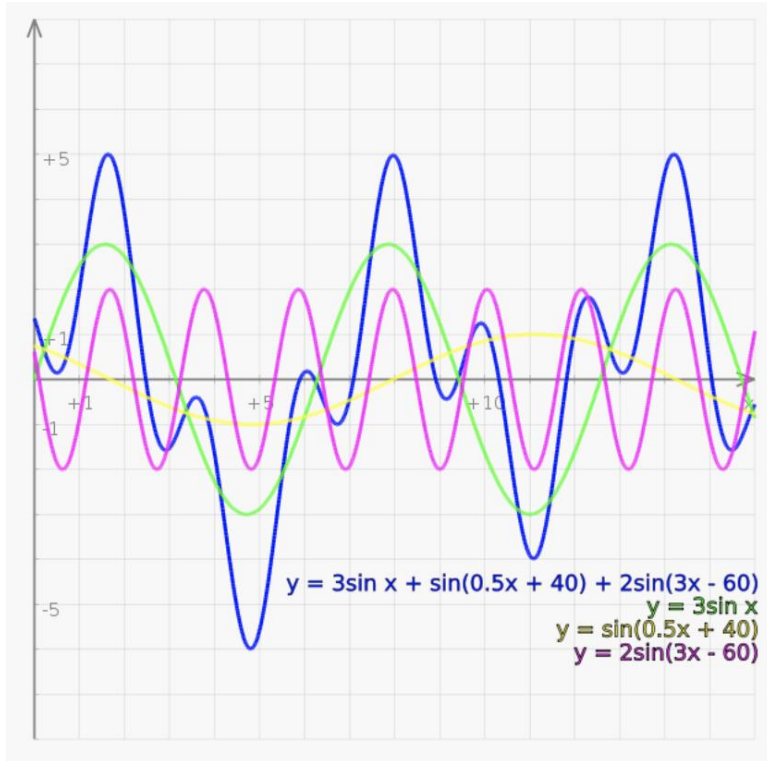


Nyquist Frequency

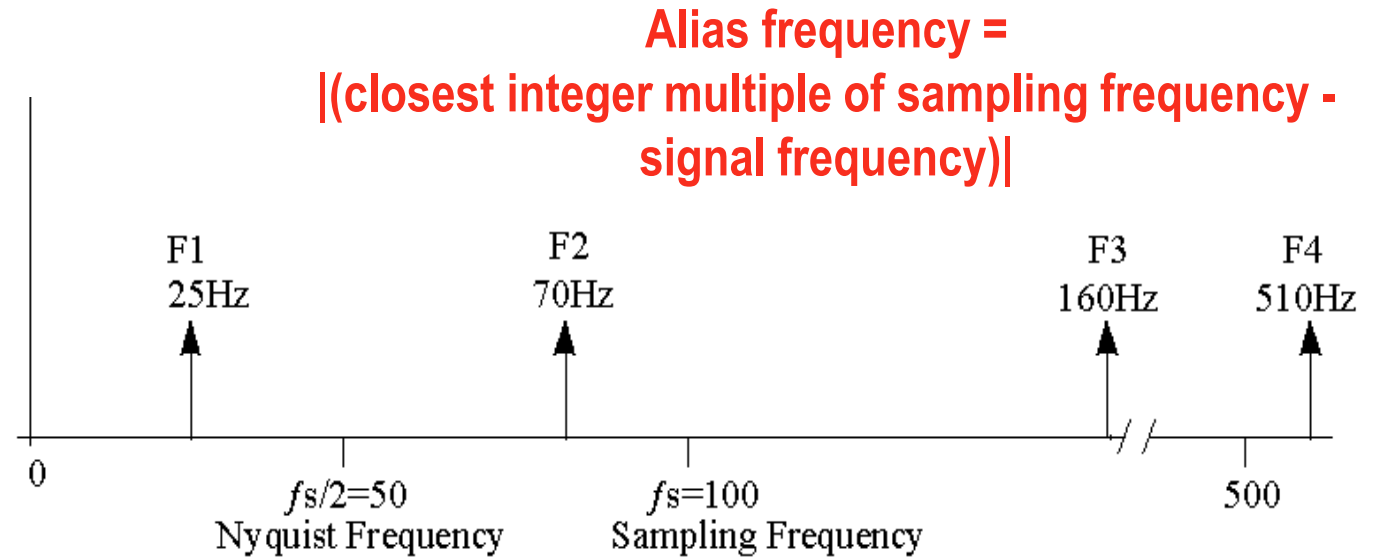
- Half the sampling frequency
- You will only get a proper representation of signals that are equal to or less than your Nyquist Frequency
- Signals above Nyquist Frequency will alias according to the following formula:

$$\text{Alias frequency} = |(\text{closest integer multiple of sampling frequency} - \text{signal frequency})|$$

Aliasing Example

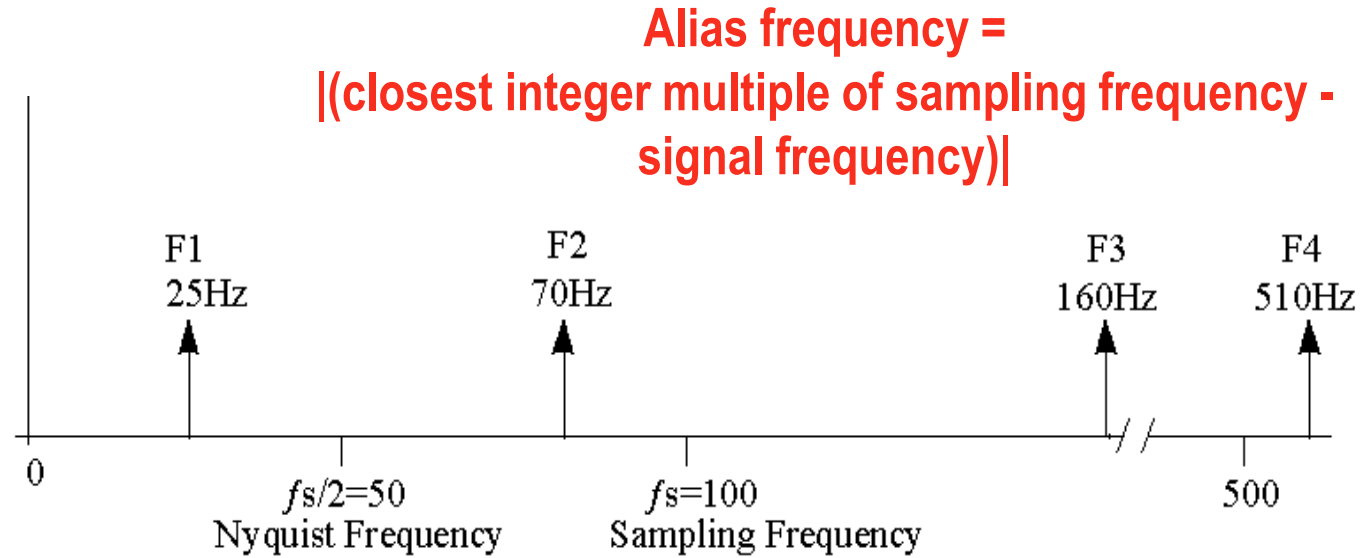


Signals before acquisition

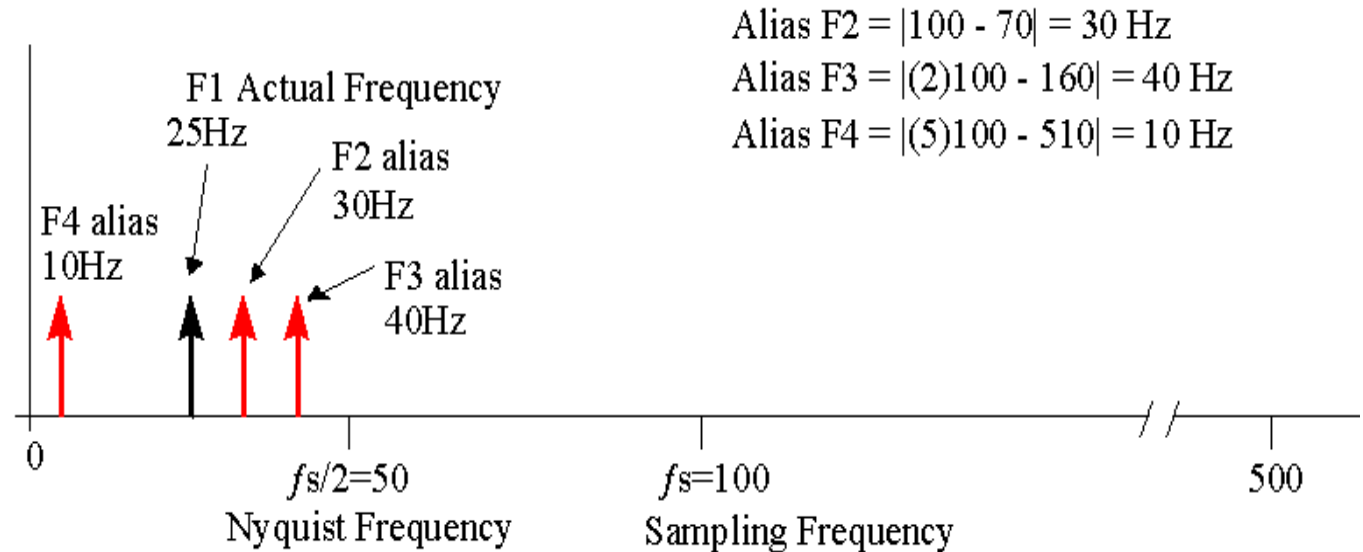


Aliasing Example

Signals before acquisition



Signals after acquisition





What has this got to do with signal capture?

- Need an analogue-to-digital converter (ADC) which is fast enough for the frequency we are trying to capture:
- **MUST** have a sampling rate at least twice as fast as the **highest-frequency component** of the signal
- Or captured signal will be rubbish (aliased)
- May need to filter out HF components to avoid this



University of
Nottingham

UK | CHINA | MALAYSIA

ADC

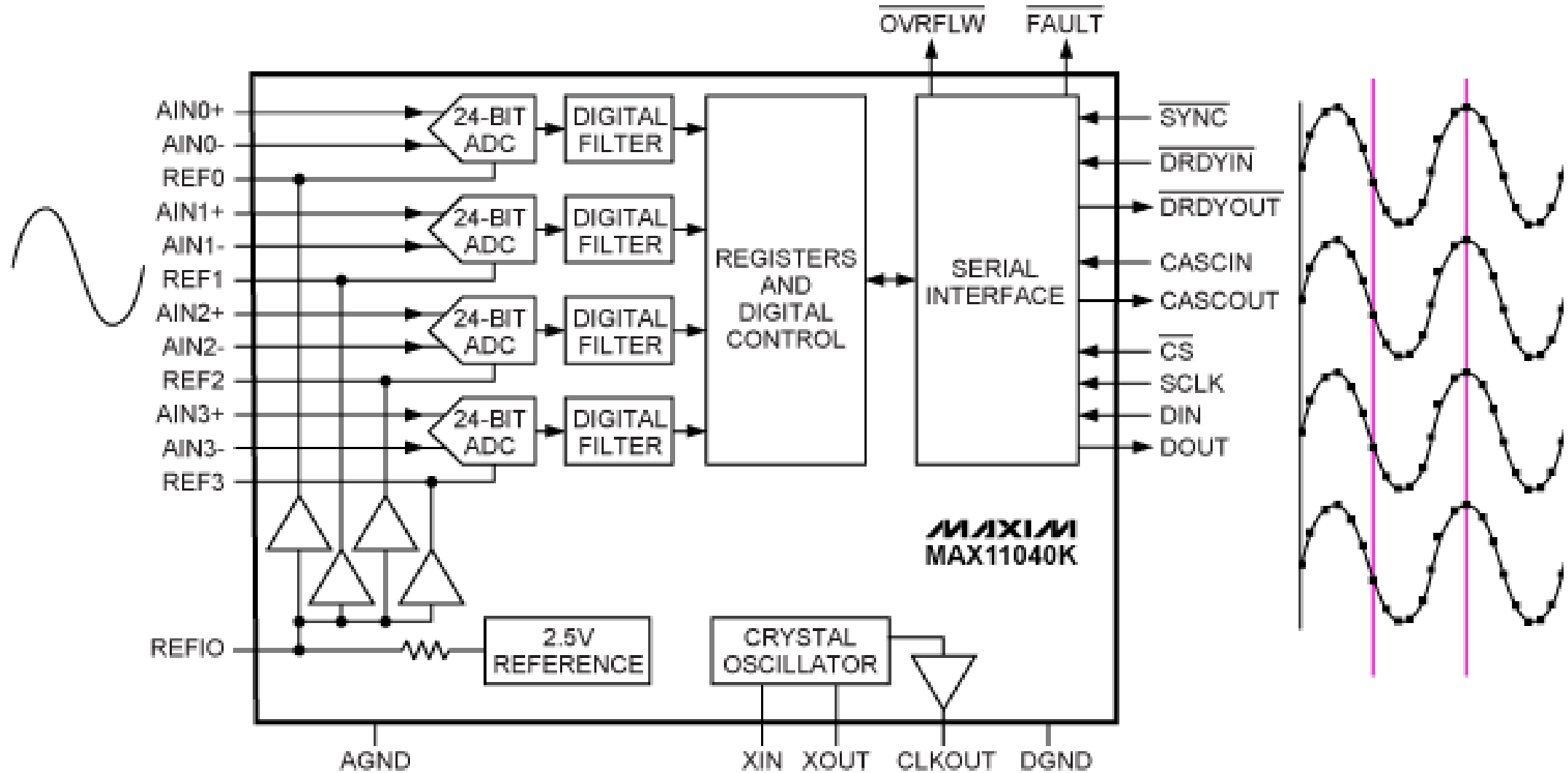
Sample and hold



Analogue signals – more about sampling

- Often we wish to capture analogue signals on **more than one channel**
- Often we are trying to capture at a **very high sample rate**
- Do we:
 - Take a snapshot of all the channels at once (**simultaneous sampling**)?
 - Use a single ADC and use it for each channel in turn (**multiplexing**)?
 - **Sample and hold with multiplexing?**

Simultaneous Sampling

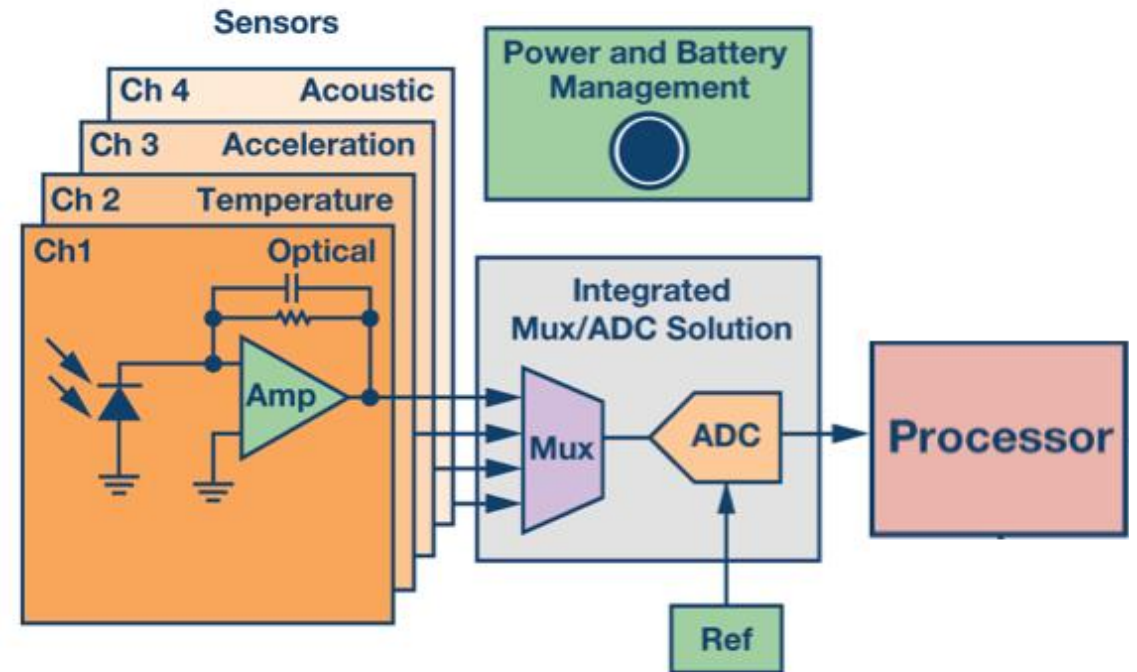
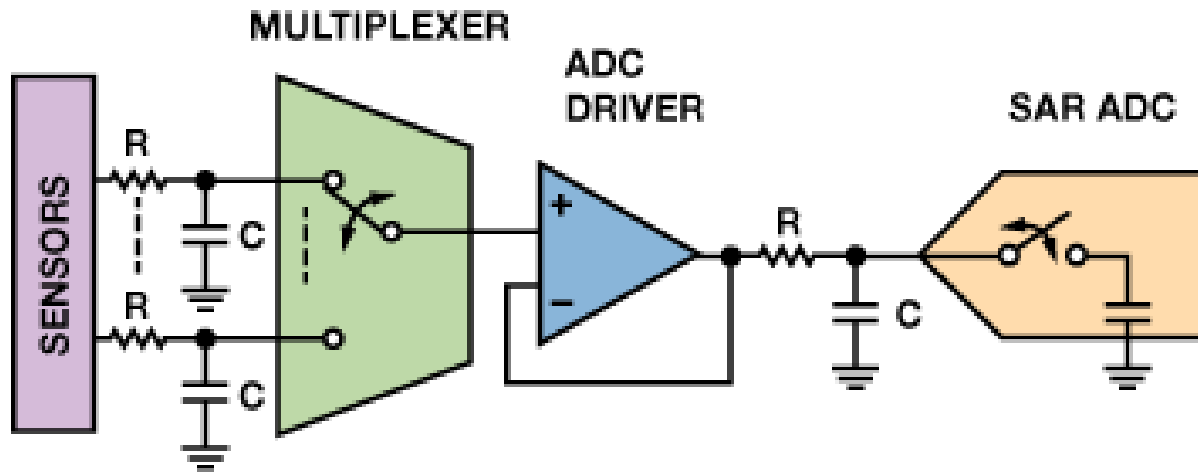




- Often we wish to capture analogue signals on **more than one channel**
- Often we are trying to capture at a **very high sample rate**
- Do we:
 - Take a snapshot of all the channels at once (**simultaneous sampling**)?
 - Use a single ADC and use it for each channel in turn (**multiplexing**)?
 - **Sample and hold with multiplexing?**



Multiplexing

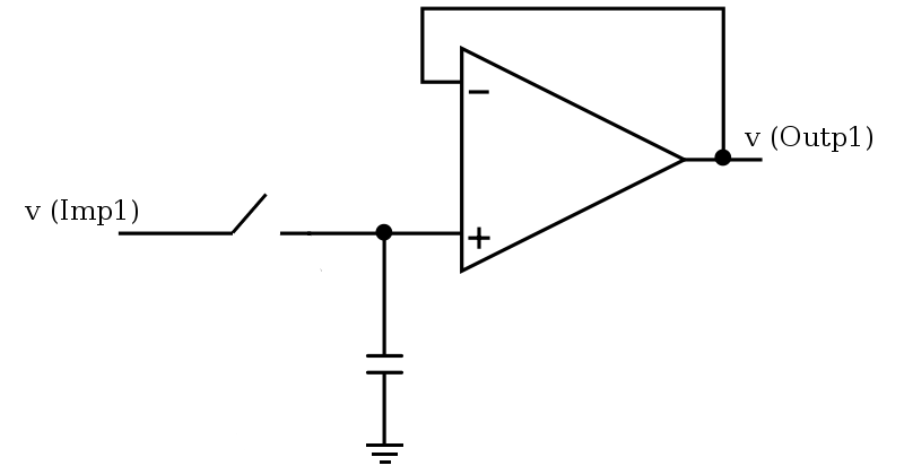




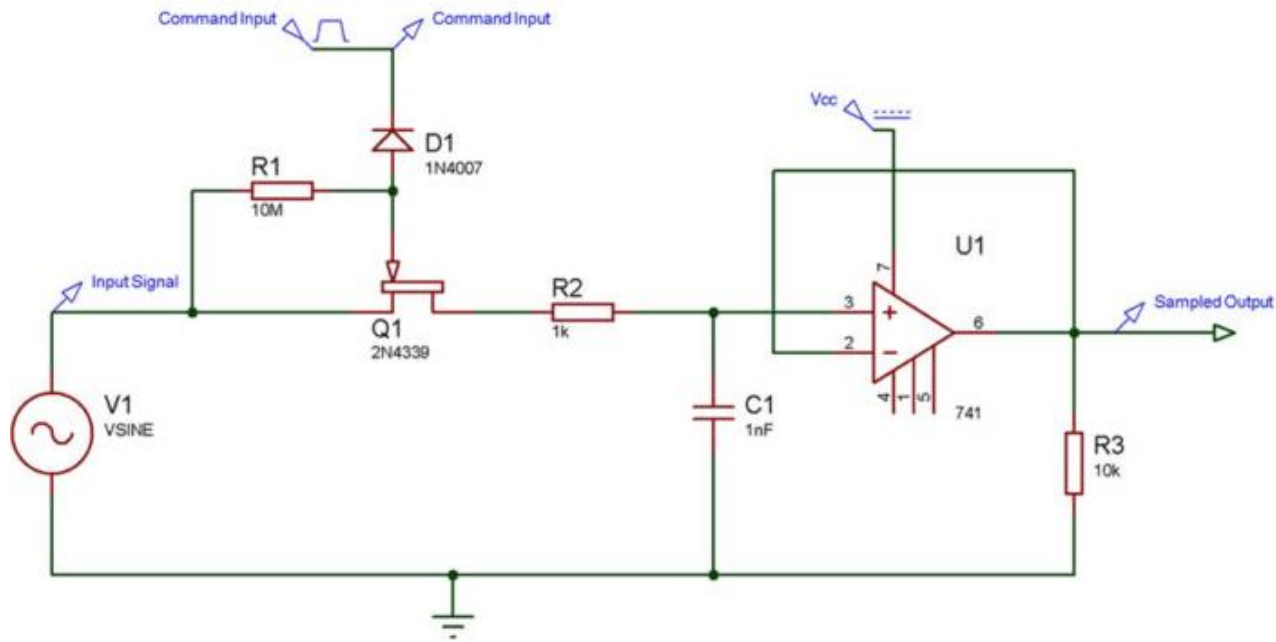
Analogue signals – more about sampling

- Often we wish to capture analogue signals on **more than one channel**
- Often we are trying to capture at a **very high sample rate**
- Do we:
 - Take a snapshot of all the channels at once (**simultaneous sampling**)?
 - Use a single ADC and use it for each channel in turn (**multiplexing**)?
 - **Sample and hold with multiplexing?**

- A sample and hold circuit (S/H) is an analog circuit that is used to sample a continuously changing analog signal and hold it at a constant level for a specified period of time.
- This is typically used in analog-to-digital conversion (ADC), where it is necessary to hold the input signal constant during the ADC conversion process.
- A simple S/H circuit consists of a switch and a capacitor. The switch is controlled by a control signal that tells it to either connect the capacitor to the input signal (sample mode) or disconnect it from the input signal (hold mode).
- When the switch is in sample mode, the capacitor charges to the value of the input signal. When the switch is in hold mode, the capacitor is isolated from the input signal and its voltage remains constant.

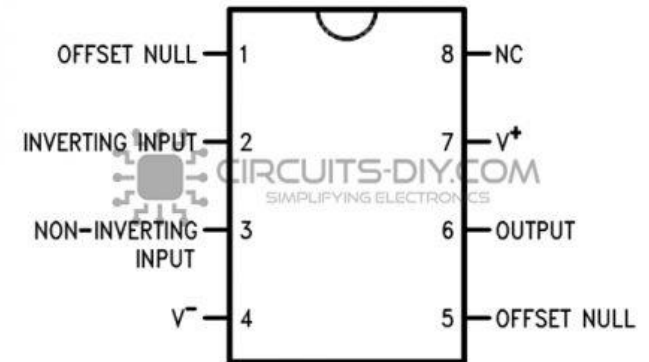


A typical sample and hold IC



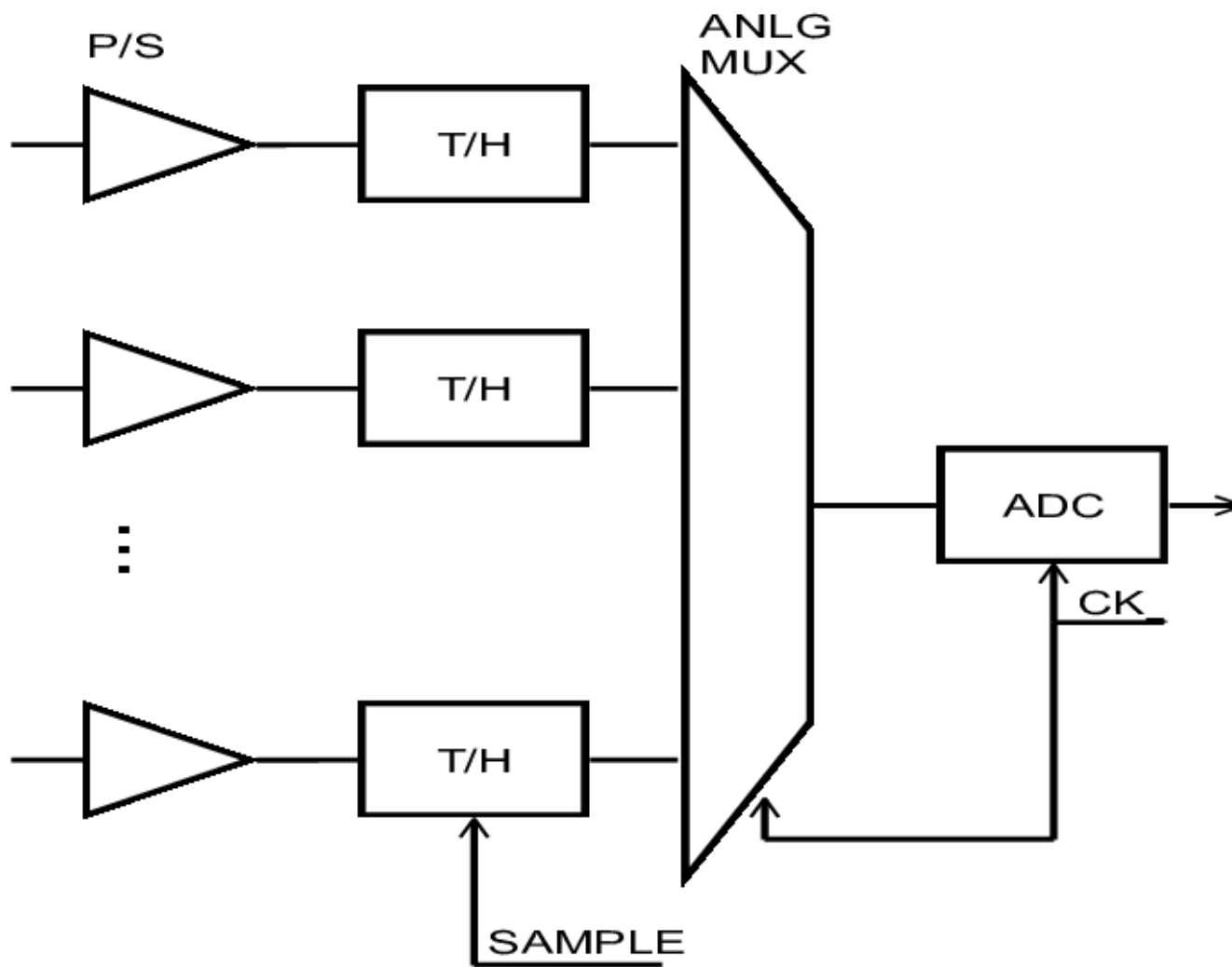
LM741

LM741 Pinout





Sample and hold with Multiplexing





- Simultaneous sampling normally needs **one ADC for each channel** – good but **expensive**
- **Multiplexing** uses “switches” to connect a single ADC between multiple sources - gives **non-simultaneous** samples, so spurious **phase shift** between channels
- **Sample and hold** is sometimes used as a compromise – **stores snapshot** of voltage in analogue device then **digitises** it at leisure



- There is yet **another issue** to consider – what **resolution** do we need, i.e. how many bits of accuracy will our ADC have?
- There is often a **trade-off** between:
 - **Resolution**
 - **Sampling rate**
- We will explore this in the next lecture



- The Atmega microcontrollers (and by extension the Arduinos) have **multiple analogue inputs**
 - Uno: 6 analogue input pins (A0-A5)
 - Mega: 16 analogue input pins (A0-A15)
- Has only one **analogue-to-digital converter (ADC)** – **multiplexed** for different inputs
- The converter is a **successive approximation ADC**



- Simple input using the Arduino software:
`long analogRead(int pin);`
- Value returned is a 10-bit binary number between 0 and 1023
- As in **Lab 1**, need to convert to voltage (by default, in range 0-5 V)

where usually $V_{REF} = 5V$

- Only signals which are **positive** wrt ground



- Revised DAC
- ADC
 - Successive approximation ADC
 - Flash ADC
 - Sigma-Delta ADC
- Sampling and Aliasing
- Sample and hold