

XMUT202 Digital Electronics

Analog to digital conversion

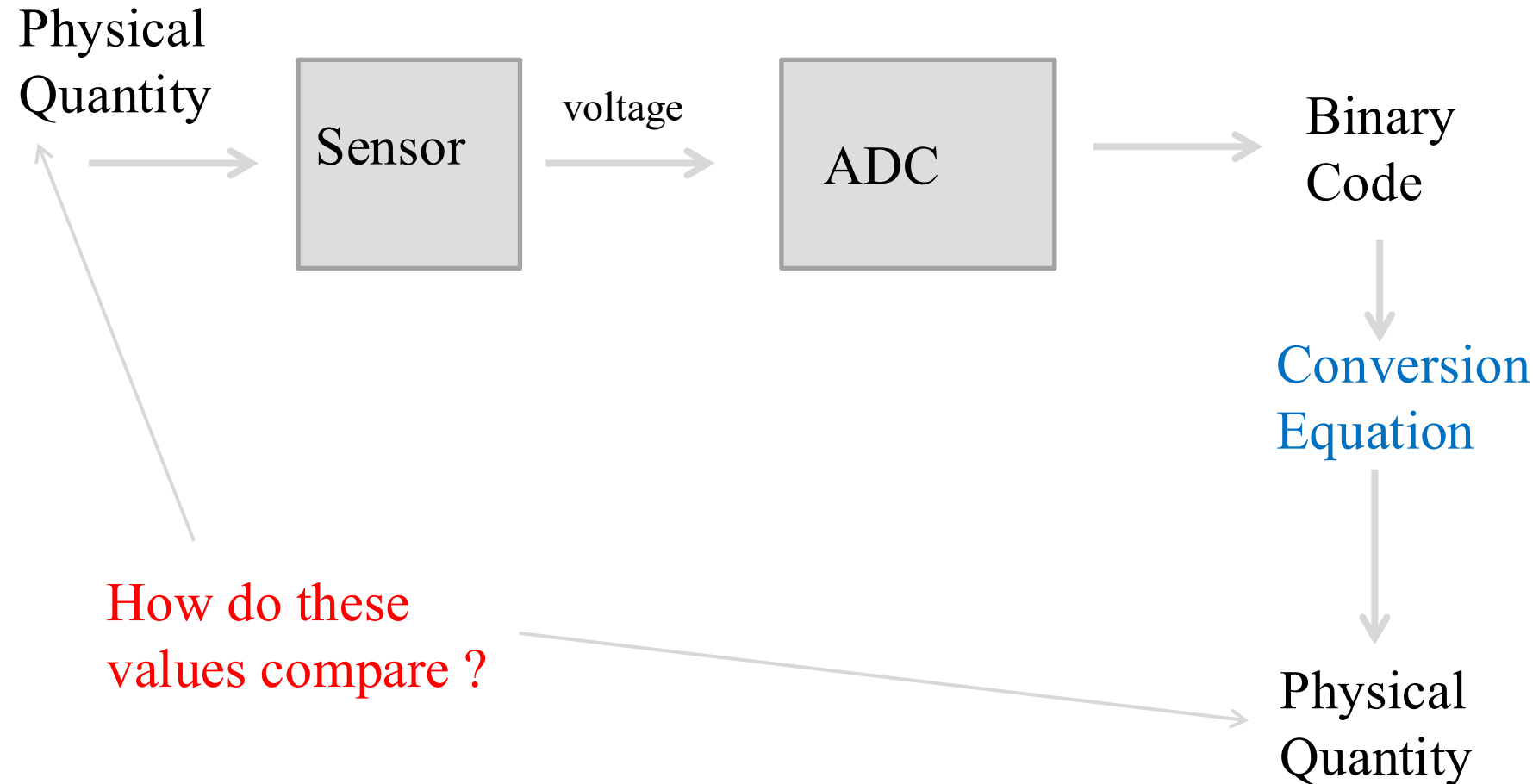
Week 16 Lecture

Agatha Rachmat

School of Engineering and Computer Science

Victoria University of Wellington

Our picture of the analog to digital conversion process

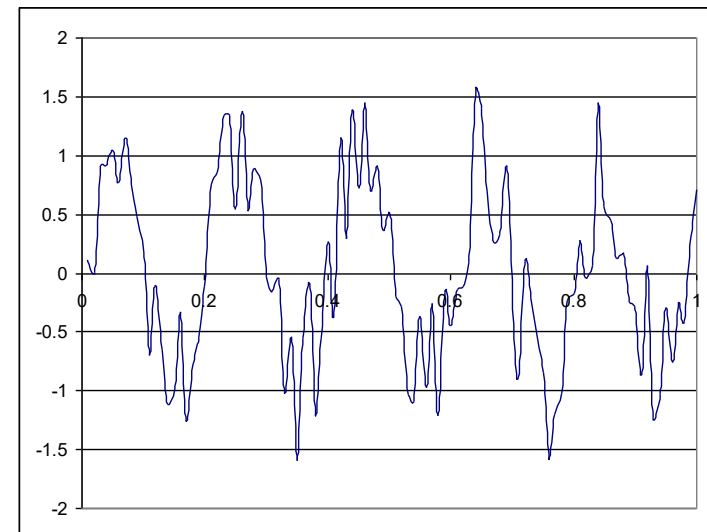
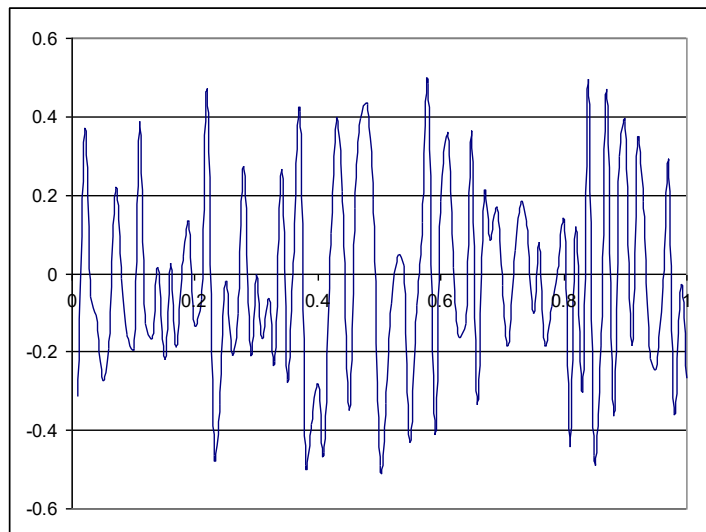
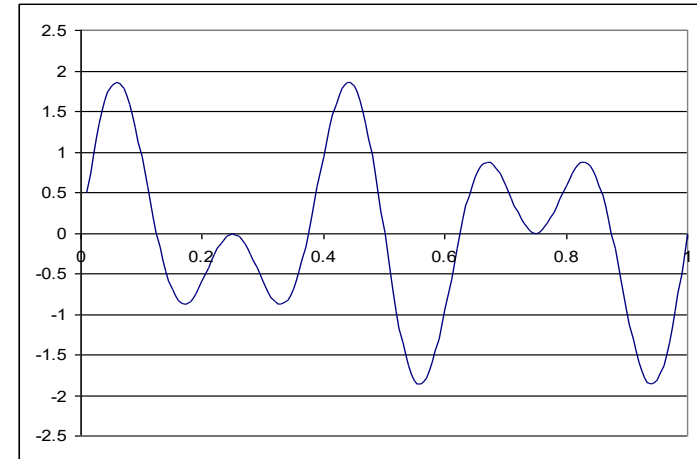
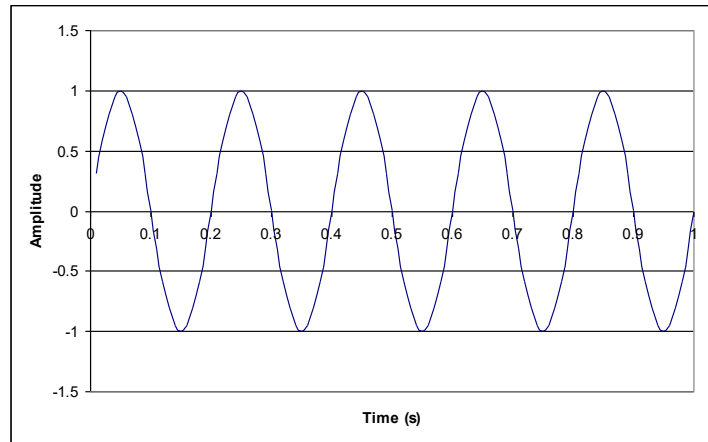


So:

- We are interested in the sampling (measuring) of analog signals by digital system
- Analog signals vary continuously in both amplitude and time.
- In order to sample such a signal we must take a “snapshot” at a certain instant in time.
- Taking this “snapshot” will have errors and uncertainties in both amplitude and time – how big are the errors?
- For this process to be performed by a computerised system the analog signal need to be converted into a digital number – use binary code for the computer.

Some different analog waveforms:

Periodic or not!



Convert analog waveform into a digital format.

To do this we must

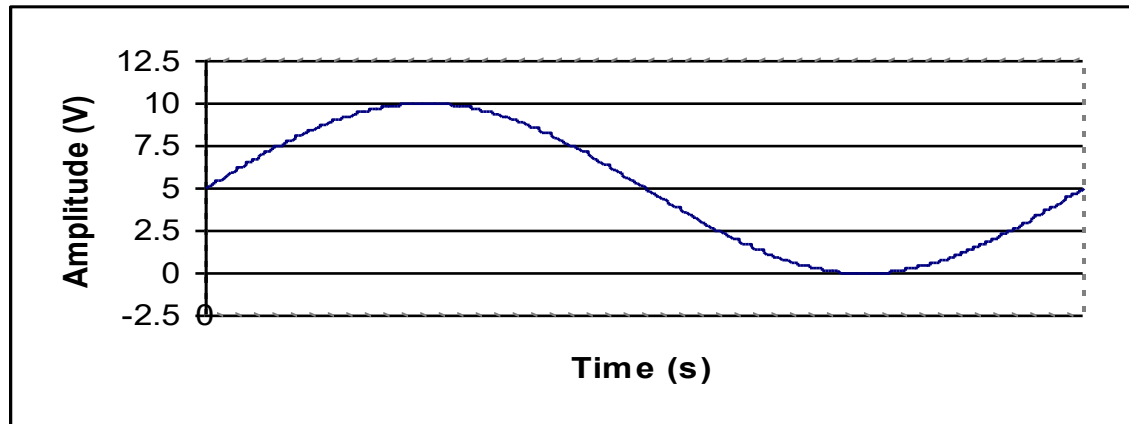
- Sample the analog waveform at certain time intervals
- Convert the amplitude of the signal at that time into a digital (binary) signal.
- Provide a time stamp for the time of sampling
- Convert the digital signal into a decimal number to make it comprehensible to the human operator
- Compare it to the original analog signal and see how well it represents this signal

Understand the limitations of the A to D process

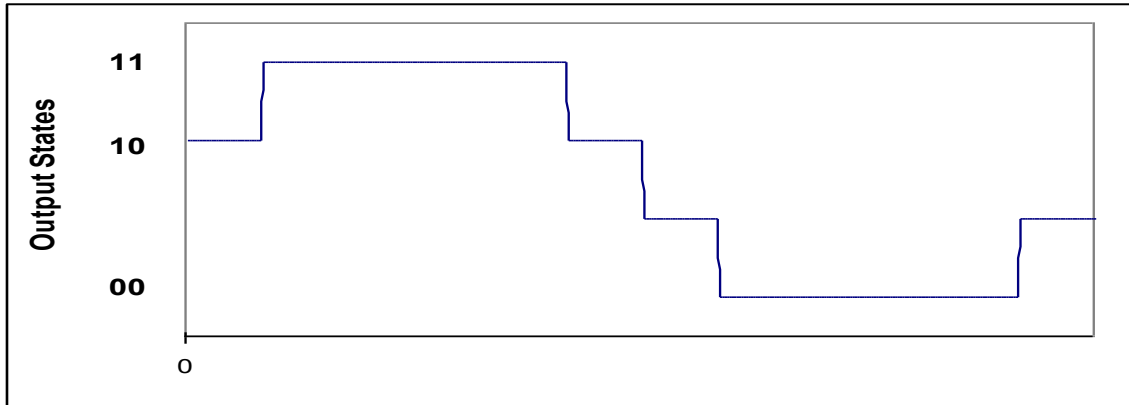
We can expect to have uncertainty (error) in both the voltage and time measurement of the ADC process. These can seriously influence the validity of our results. If we understand the ADC process we can minimise these errors – sample our signal with the best precision we can.

Will first look at a *conceptual ADC process* – then look at the construction and operation of some real ADCs.

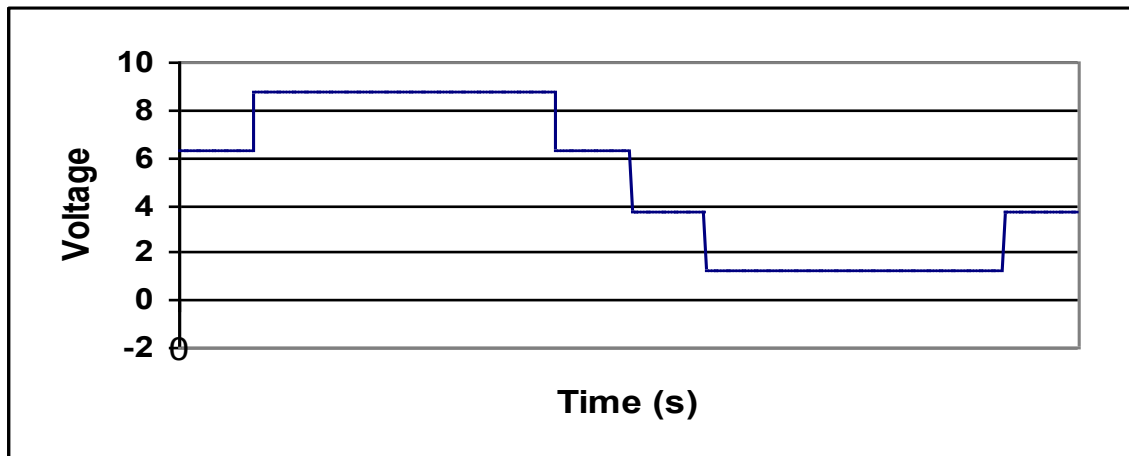
Analog Input



Digital (Binary)
Conversion



Digital (Decimal)
Output



Conceptual ADC (Analog-to-Digital Converter) Process

An **ADC (Analog-to-Digital Converter)** is a circuit that converts a **continuous analog signal (like voltage or temperature)** into a **digital binary number** that a microcontroller or computer can understand.

Basic Idea

An ADC does three main things:

- 1.Samples** the analog signal
- 2.Quantises** it into discrete levels
- 3.Encodes** it into binary form

Simple Real-Life Analogy

Think of an ADC like a thermometer with fixed markings:

- Analog temperature = smooth real value (e.g., 27.3°C)
- ADC = rounds it to nearest mark (e.g., 27°C or 28°C)
- Output = number printed (binary form in electronics)

Conceptual ADC (Analog-to-Digital Converter) Process

It is clear that this digital output is only a representation of the analog input. We can have an error of ± 1.25 V in the decimal output = \pm half of a voltage step.

1. Digital output is only a representation of the analog input

An ADC cannot produce the exact analog value. Instead, it:

- Divides the input range into fixed steps (levels)
- “Rounds” the real voltage to the nearest step
- Outputs a digital code representing that step

So the digital output is an **approximation**, not the exact value.

2. Why an error happens

Because the ADC has fixed steps, any value between two levels must be rounded.

Example (simple idea):

- Suppose step size = 2.5 V
- Levels: 0 V, 2.5 V, 5 V, 7.5 V, ...

Now:

- 3.0 V \rightarrow rounded to 2.5 V or 5 V (depends on design)
- Either way \rightarrow error is introduced

Conceptual ADC (Analog-to-Digital Converter) Process

To improve the ADC representation, you **increase the number of bits (resolution)**. This adds more output states (levels), making the digital approximation closer to the analog input.

1. Key Idea

An ADC with **N bits** has: 2^N *levels*

So if you **add 1 more bit**, you **double the number of levels**.

Example: Adding One More Bit

Original system (3-bit ADC)

- Bits = 3
- Levels: $2^3 = 8$ *levels*
- Step size = large
- Error = large ($\pm 1/2$ LSB)

Improved system (4-bit ADC)

Now we add **one more output bit**:

- Bits = 4
- Levels: $2^4 = 16$ *levels*

So the voltage range is divided more finely.

Conceptual ADC (Analog-to-Digital Converter) Process

To improve the ADC representation, you **increase the number of bits (resolution)**. This adds more output states (levels), making the digital approximation closer to the analog input.

What improves?

(A) Smaller step size

If full-scale voltage = V_{REF} :

- 3bit step: $\frac{V_{REF}}{8}$

- 4bit step: $\frac{V_{REF}}{16}$

So step size becomes **half as big**.

(B) Smaller quantization error

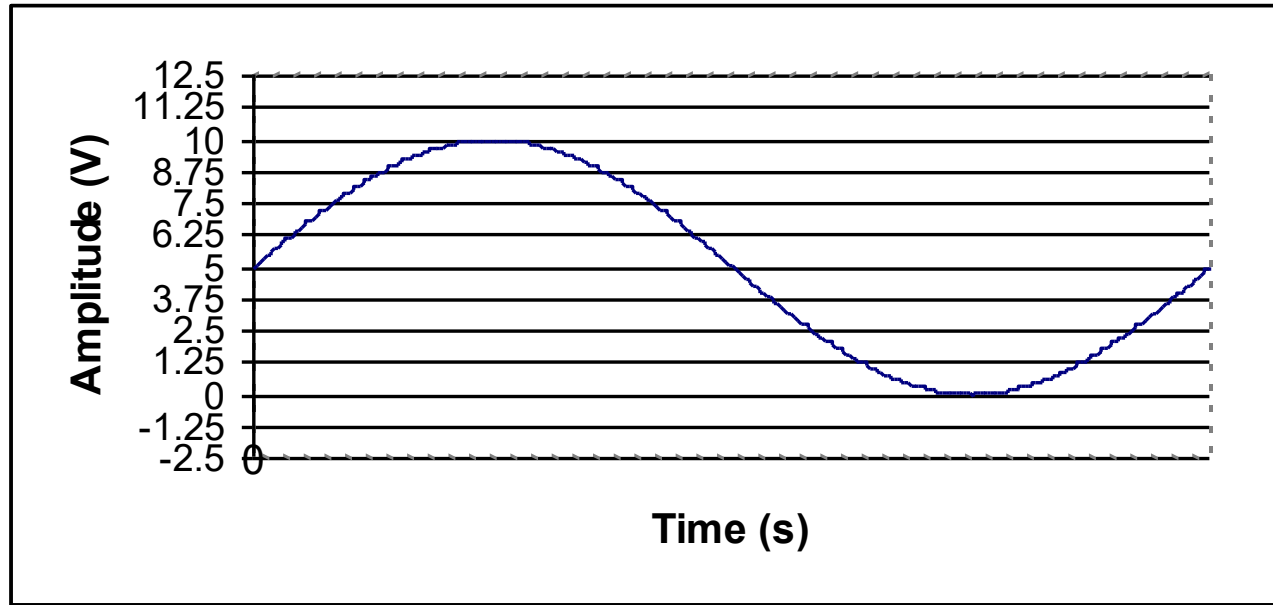
Error is always:

$$\pm \frac{1}{2} \text{ LSB}$$

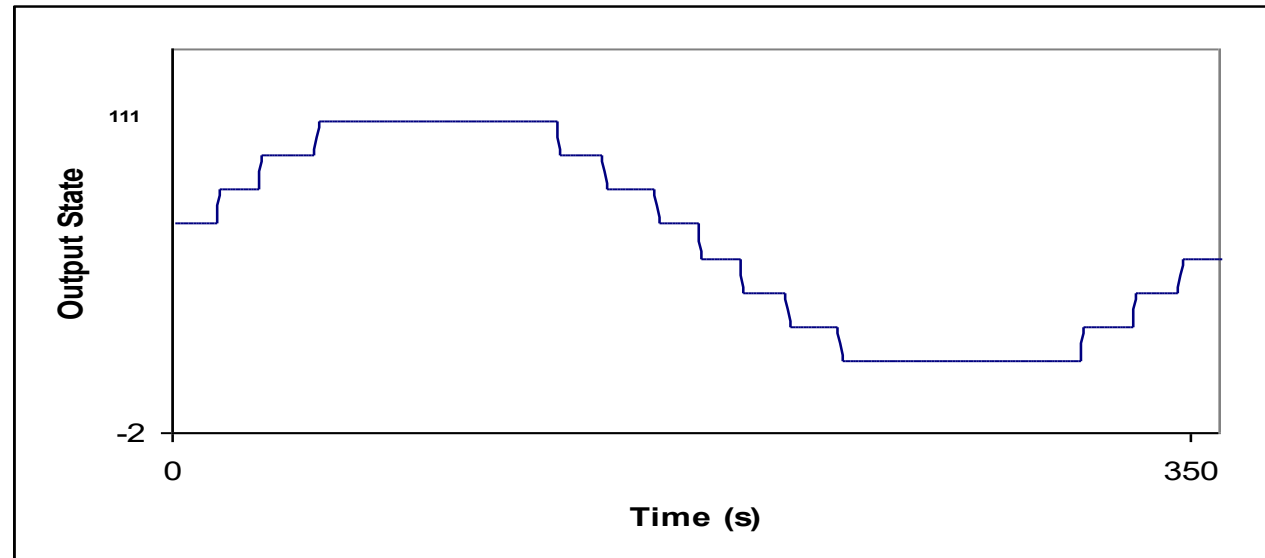
So when LSB becomes smaller:

- Error also becomes smaller
- Output is more accurate

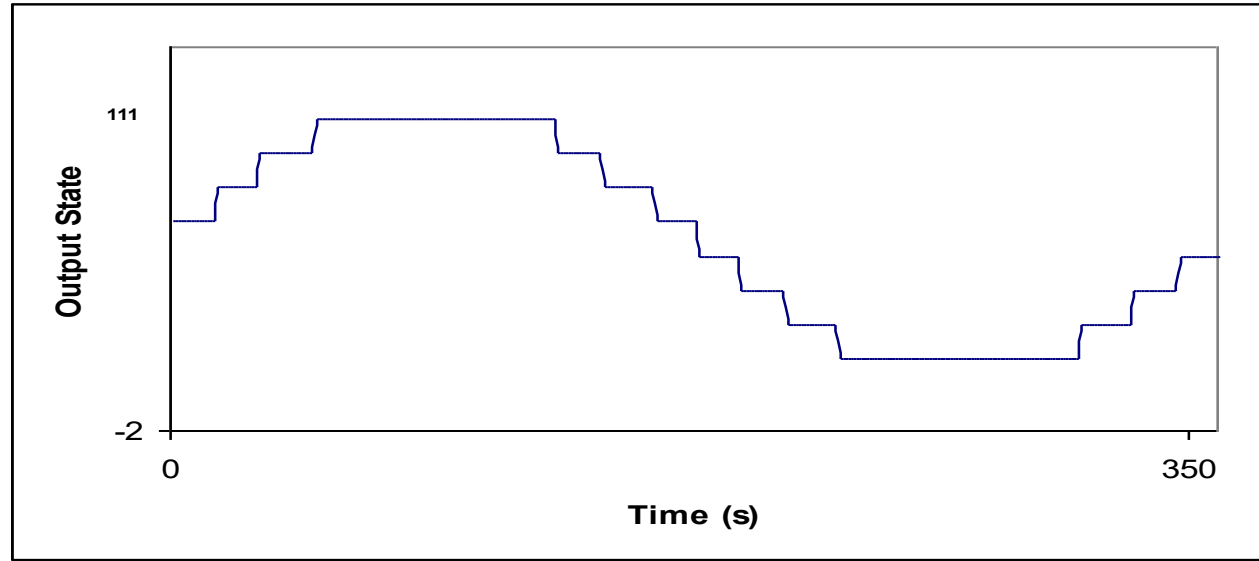
Analog Input



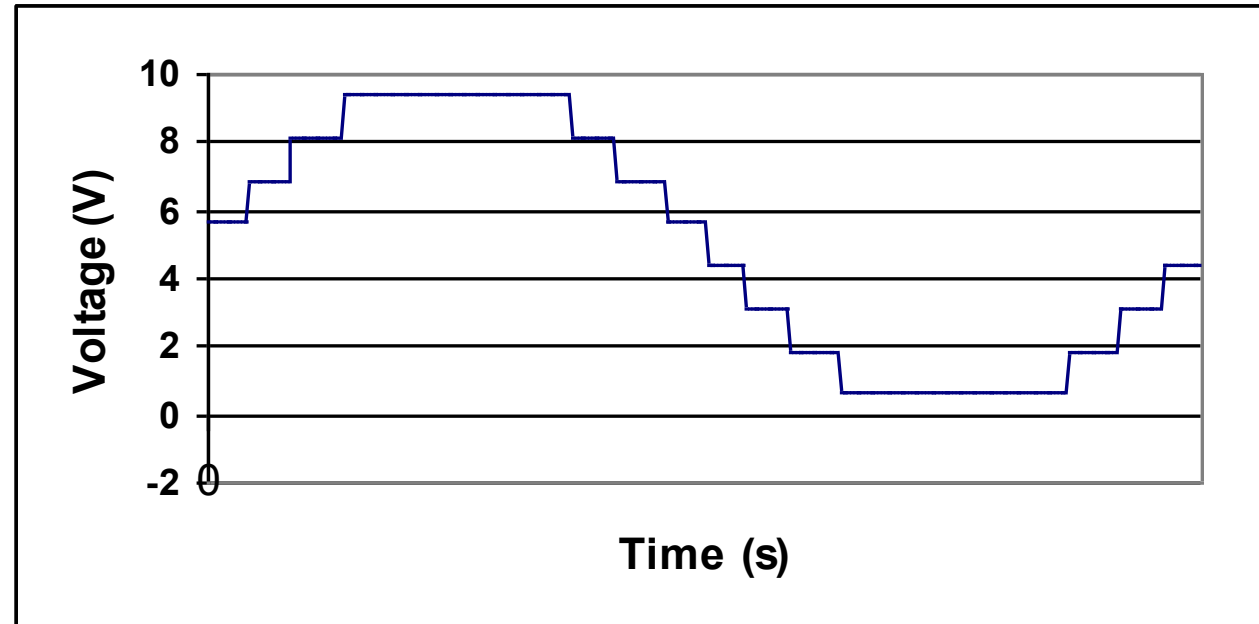
Digital (Binary)
Output



Digital (Binary)
Output



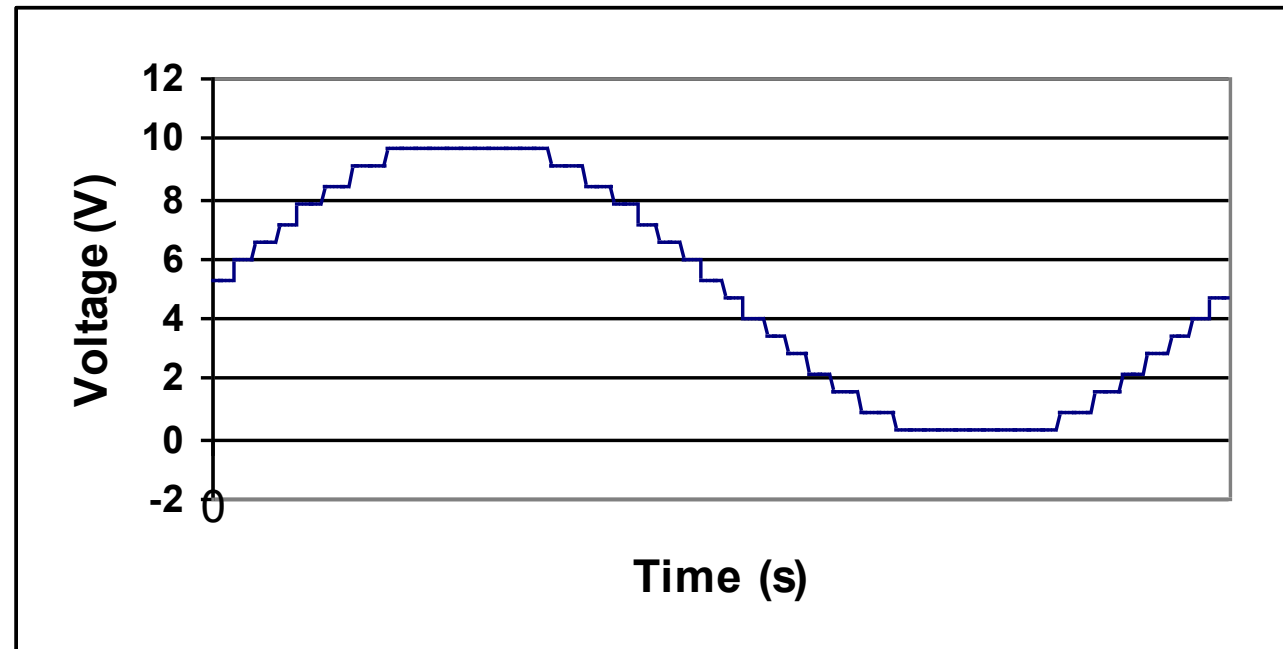
Digital (Decimal)
Output



ADC (Analog-to-Digital Converter) Process

A 3-bit converter now gives us a much better approximation of the analog input with the error (uncertainty) in the decimal output considerably smaller.

This digital output will be further improved in the case of a four bit converter



ADC (Analog-to-Digital Converter) Process

The number of different binary numbers that can represent the analog input = $2^{\text{no of bit}}$ in each case.

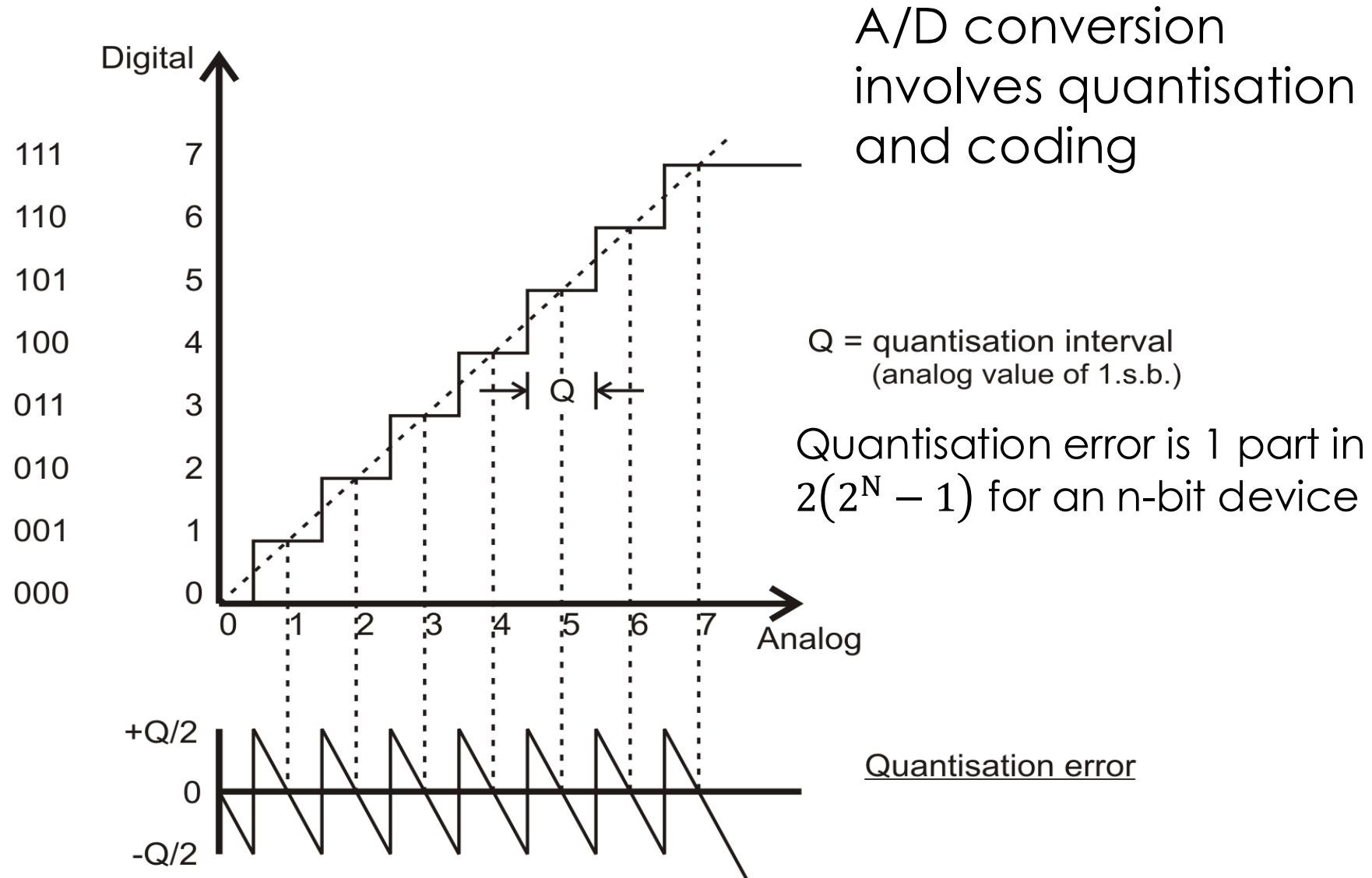
The number of steps between the levels would thus be $2^{\text{no of bit}} - 1$.

The precision with which the input signal can be represented increases as the number of bits increases.

However, the use of more bits would also increase the complexity of the circuitry

If the input signal is higher than half the input range, the MSB will remain turned on.

Quantisation is the process of converting a continuous analog signal into a set of discrete output levels.



Practical embodiment of an A to D.

- Compare the incoming voltage to certain, set reference levels
- Set the binary output state according to this comparison.

Use a very handy electronic device called a comparator

The Comparator

- Based on an operational amplifier
- Forms a neat interface between the analog and digital worlds
- It takes an analog signal at its input and compares it to a reference analog signal.
- Depending on the input, the comparator outputs either LO or HI (binary).

What is a Comparator?

A **comparator** is an electronic circuit that compares two voltages and tells which one is larger.

Basic idea:

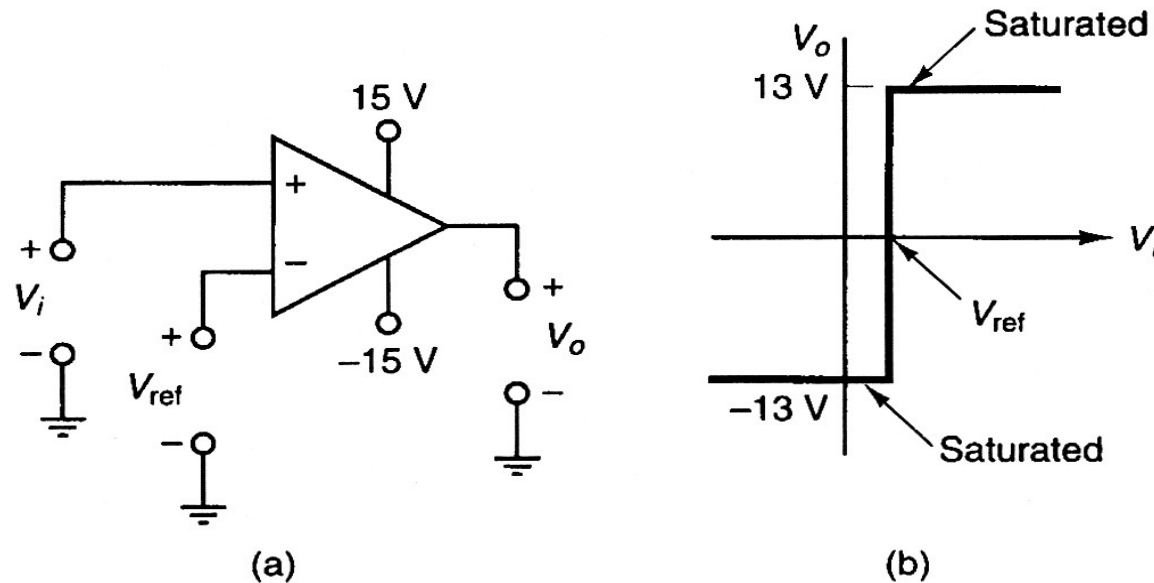
It has two inputs:

- V_{in}) unknown analog signal)
- V_{ref}) reference voltage)

The Comparator

Input signal	Output signal
$V_{in} < V_{ref}$	LO
$V_{in} > V_{ref}$	HI

It can be described as a single-bit analog-to-digital converter !



Voltage divider

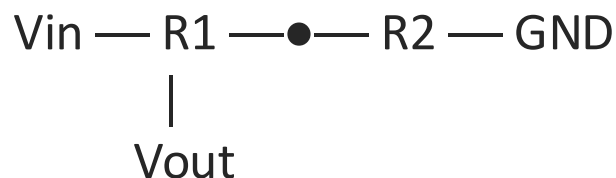
- When a voltage is applied to a series combination of resistances, a fraction of voltage appears across each of the resistance.
- A series circuit acts as a voltage divider. The voltage divider is an important application of series circuits.
- A circuit consisting of a series string of resistors connected to a voltage source acts as a voltage divider.

What is a Voltage Divider?

A **voltage divider** is a simple circuit that splits a higher voltage into smaller voltages using **two resistors in series**.

It is widely used in electronics (including ADC reference circuits) to generate specific voltage levels.

Basic Circuit

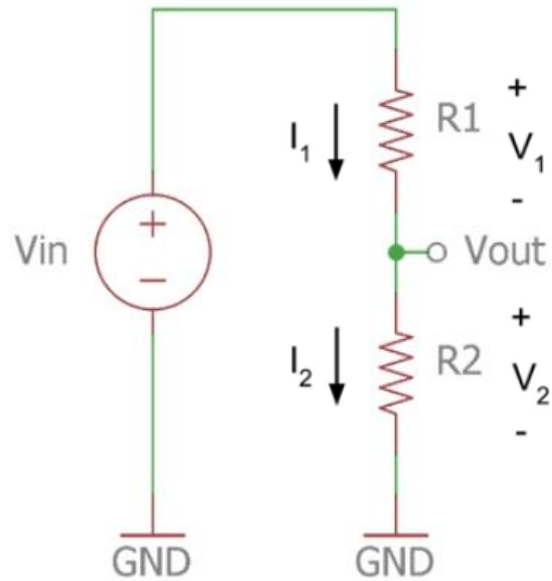


• V_{in} = input voltage

• $R1, R2$ = resistors

• V_{out} = output voltage (taken at the middle point)

How to calculate voltage in voltage divider?



$$V_{out} = V_2$$

$$V_{out} = I_2 \times R_2$$

$$I_2 = I_1 = I$$

$$V_{out} = I \times R_2$$

$$I = \frac{V_{out}}{R_2}$$

$$V_{in} = I \times R$$

$$R = R_1 + R_2$$

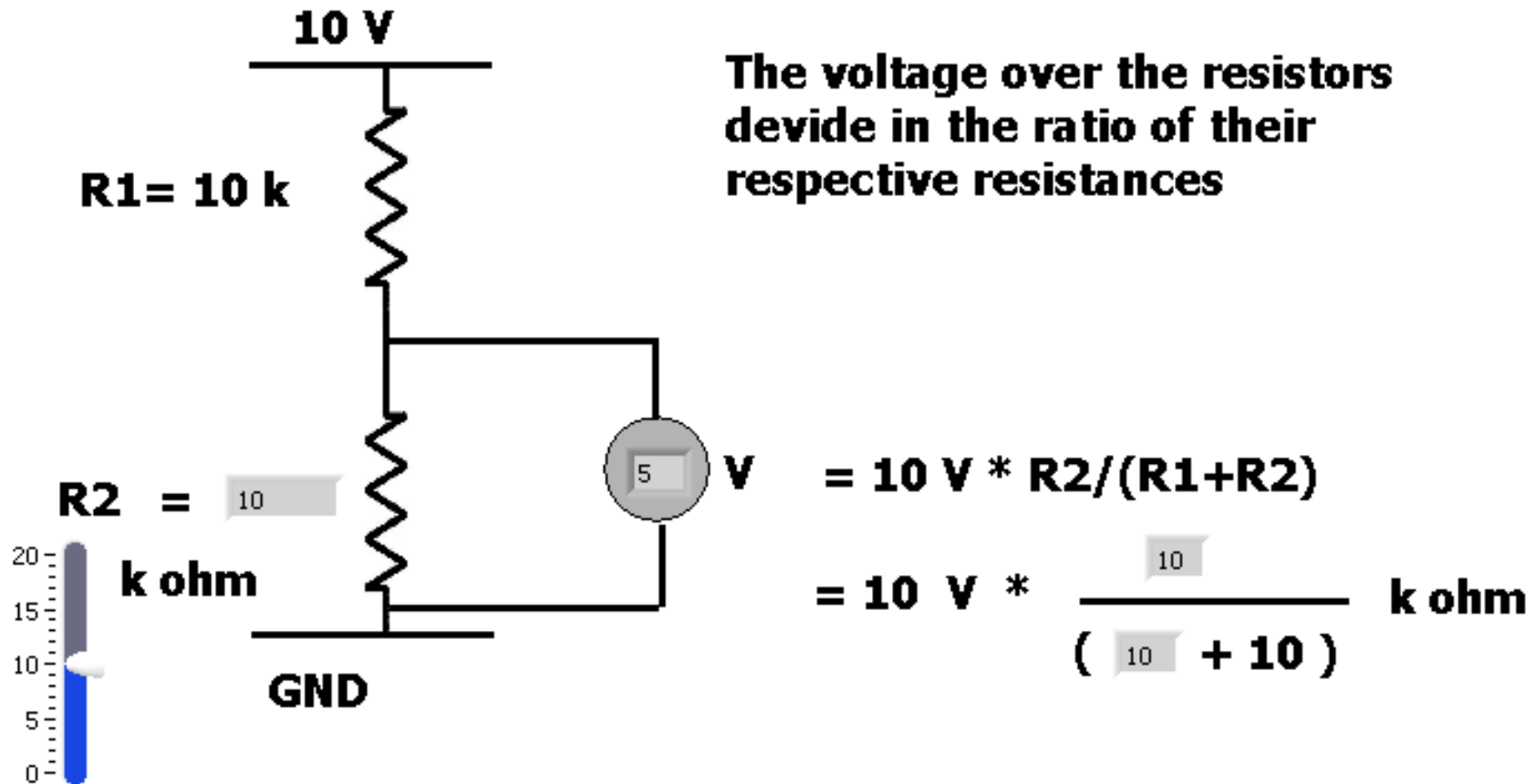
$$V_{in} = I \times (R_1 + R_2)$$

$$I = \frac{V_{in}}{(R_1 + R_2)}$$

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

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

The resistor chain - A simple method to create a series of reference voltages.



**The voltage over the resistors
divide in the ratio of their
respective resistances**

$$\begin{aligned} &= 10 \text{ V} * R2 / (R1 + R2) \\ &= 10 \text{ V} * \frac{10}{(10 + 10)} \text{ k ohm} \end{aligned}$$

The operation of many types of A/D are based on or use the comparator.

We will look at three types:

(1) The flash converter

(2) The ramp/staircase converter

(3) The successive approximation (SAC) converter

FLASH Converter

- Compare the incoming analogue voltage to reference voltage levels and decide which reference levels does best represent it.
- Convert this reference level to a binary code, using a predetermined number of bits.
- Convert the binary code to a decimal number to make it easy for the operator to interact.

Operation of a 3-bit parallel (flash) A/D

Assume an input range of 0 – 7 V

Each “step” is 1 V as created by resistor chain voltage divider

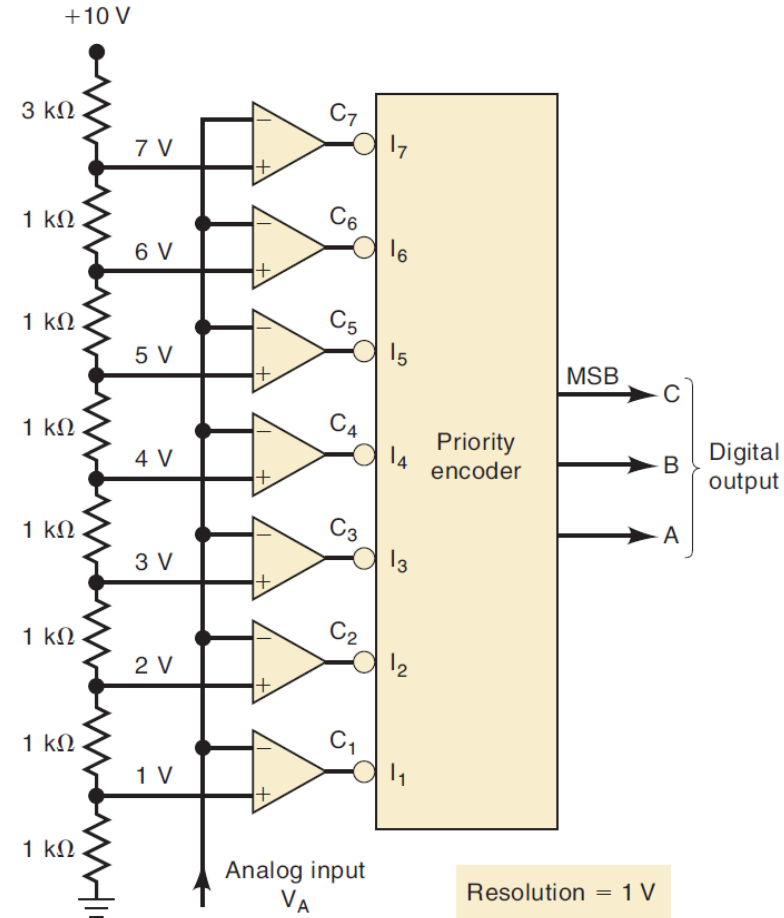
To represent 8 steps we will need 3 bits – 8 possible OP states.

To determine which is the highest active level we need 7 comparators

Thus need $2^n - 1$ comparators

Need 2^n resistors

Need priority encoder to code from 7 IPs to 3-bit code



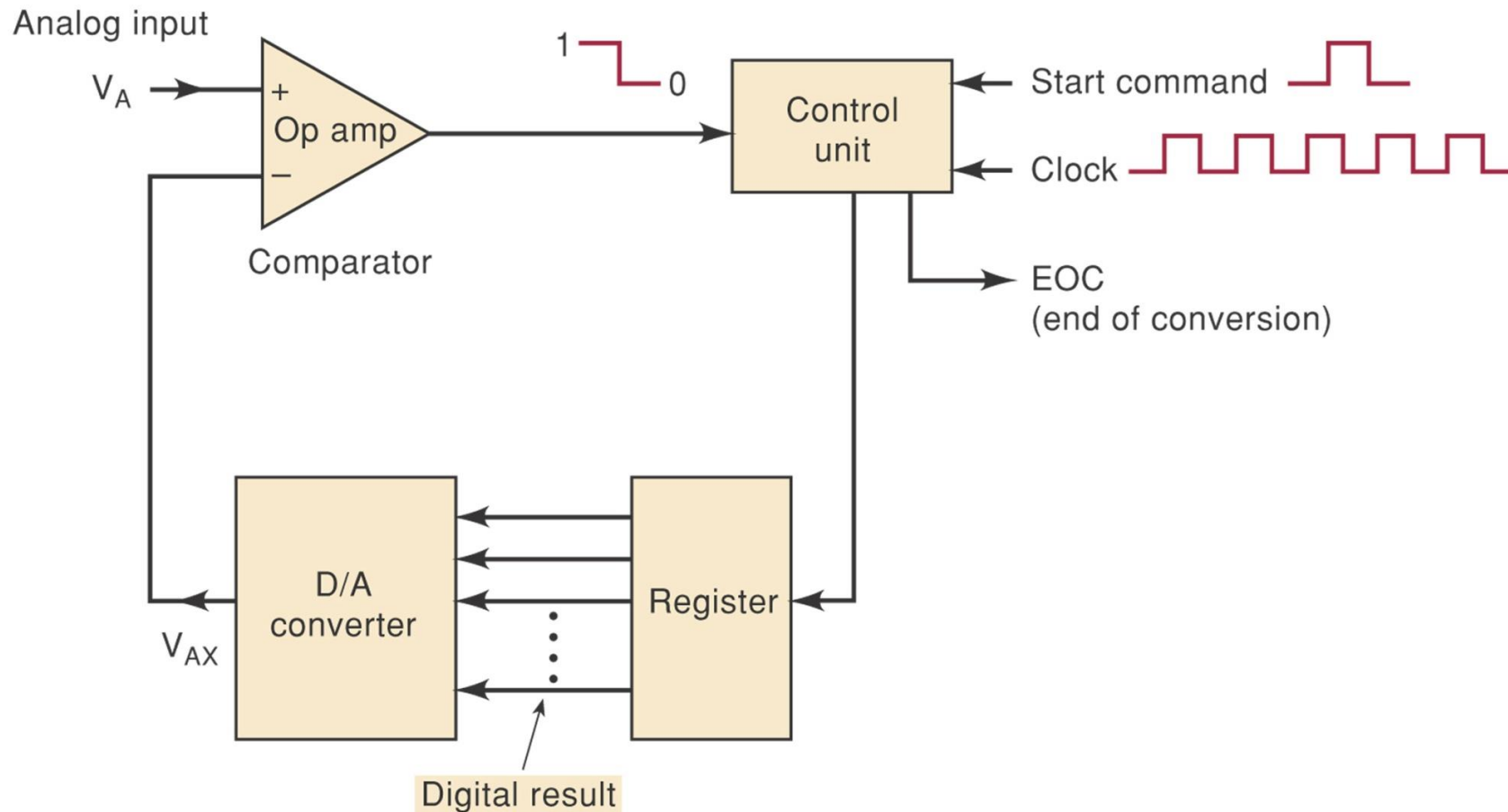
Analog in	Comparator outputs							Digital outputs		
V_A	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C	B	A
0–1 V	1	1	1	1	1	1	1	0	0	0
1–2 V	0	1	1	1	1	1	1	0	0	1
2–3 V	0	0	1	1	1	1	1	0	1	0
3–4 V	0	0	0	1	1	1	1	0	1	1
4–5 V	0	0	0	0	1	1	1	1	0	0
5–6 V	0	0	0	0	0	1	1	1	0	1
6–7 V	0	0	0	0	0	0	1	1	1	0
> 7 V	0	0	0	0	0	0	0	1	1	1

The parallel or flash A/D converter

- Very fast converter
- Compares the analogue voltage to all reference levels simultaneously. No clock signal is used, so the conversion is continuous.
- We would need a comparator for every step (reference levels – 1). Thus to output n bits we would need $2^n - 1$ comparators. 10 bit flash ADC requires 1023 comparators!
- Use resistor based voltage divider
- Use decoder to change comparator output into binary number
- Very expensive due to high number of comparators
- Used only in high end (expensive) A/D's where speed is important

Feedback ADCs

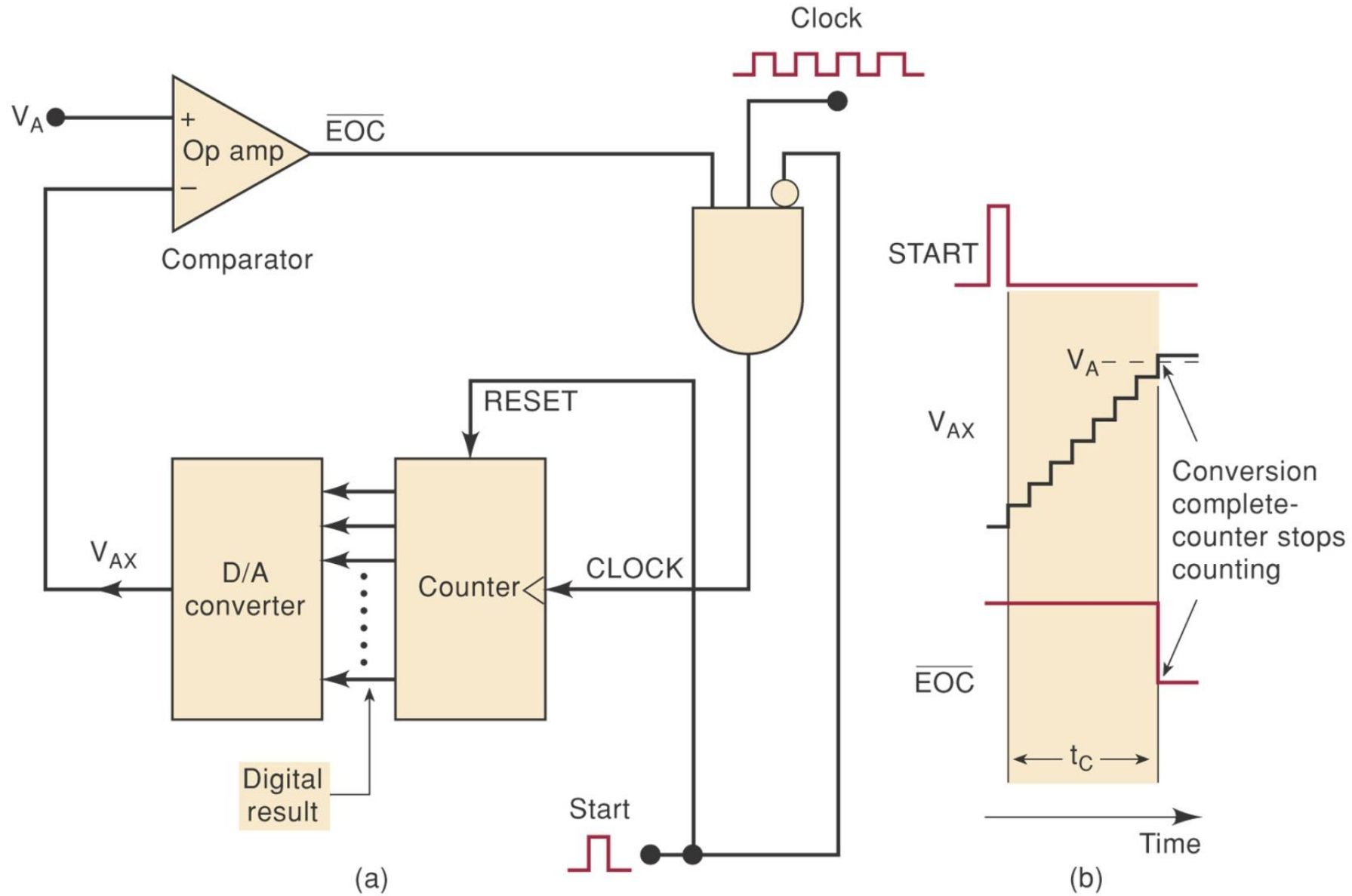
- The ADC uses a DAC and a comparator.
- A guess is applied to the DAC and the resulting output is compared with the input.



The ramp/staircase/counter ADC

- Uses a binary up counter
- Feeds counter into digital to analog converter
- DAC generates an analog voltage representing the binary count of the counter
- This increasing analog voltage is compared by comparator to the analog voltage to be measured (V_{in}) and the comparator will change state when $V_{DAC} > V_{in}$
- This comparator transition will stop the counter and also signal “End Of Conversion” The counter value representing the binary conversion of the analog voltage.

The ramp/staircase/counter ADC



Problem:

Assume the following values for the digital-ramp ADC, clock frequency = 1 MHz; $V_T = 0.1$ mV; DAC has F.S. output = 10.23 V and a 10-bit input. Determine the following values.

- (a) The digital equivalent obtained for $V_A = 3.728$ V
- (b) The conversion time
- (c) The resolution of this converter

Solution:

(a) The DAC has a 10-bit input and a 10.23-V_{F.S.} output. Thus, the number of total possible steps is $2^{10} - 1 = 1023$, and so the step size is $10.23 \text{ V} / 1023 = 10 \text{ mV}$

This means that V_{AX} increases in steps of 10 mV as the counter counts up from 0.

Because $V_A = 3.278 \text{ V}$ and $V_T = 0.1 \text{ mV}$, V_{AX} must reach 3.7281 V for more before the comparator switches LOW. This will require

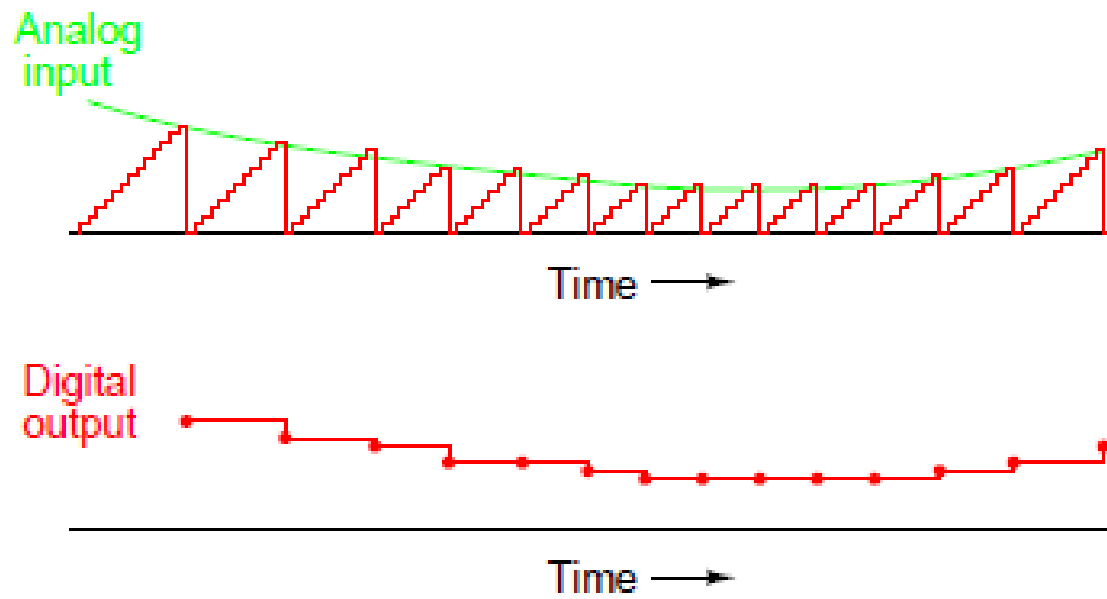
$$3.7281 \text{ V} / 10 \text{ mV} = 372.81 = 373 \text{ steps}$$

At the end of the conversion, then, the counter will hold the binary equivalent of 373, which is 0101110101. This is the desired digital equivalent of $V_A = 3.728 \text{ V}$, as produced by this ADC.

Solution:

(b) 373 steps were required to complete the conversion. Therefore, 373 clock pulses occurred at the rate of one per microsecond. This gives a total conversion time of 373 μs

(c) The resolution of this converter is equal to the step size of the DAC, which is 10 mV. Expressed as a percentage, it is $1/1023 * 100 \approx 0.1\%$.

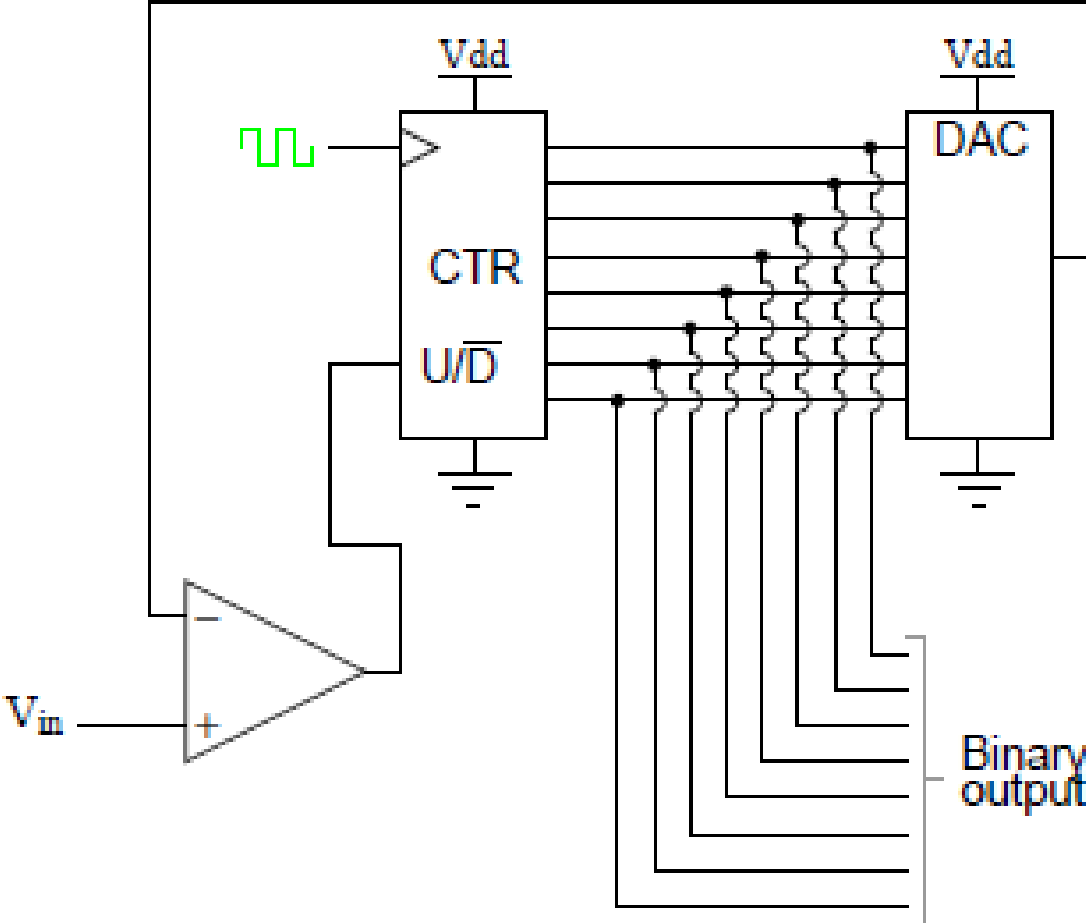


Operation simple and easy to construct but:

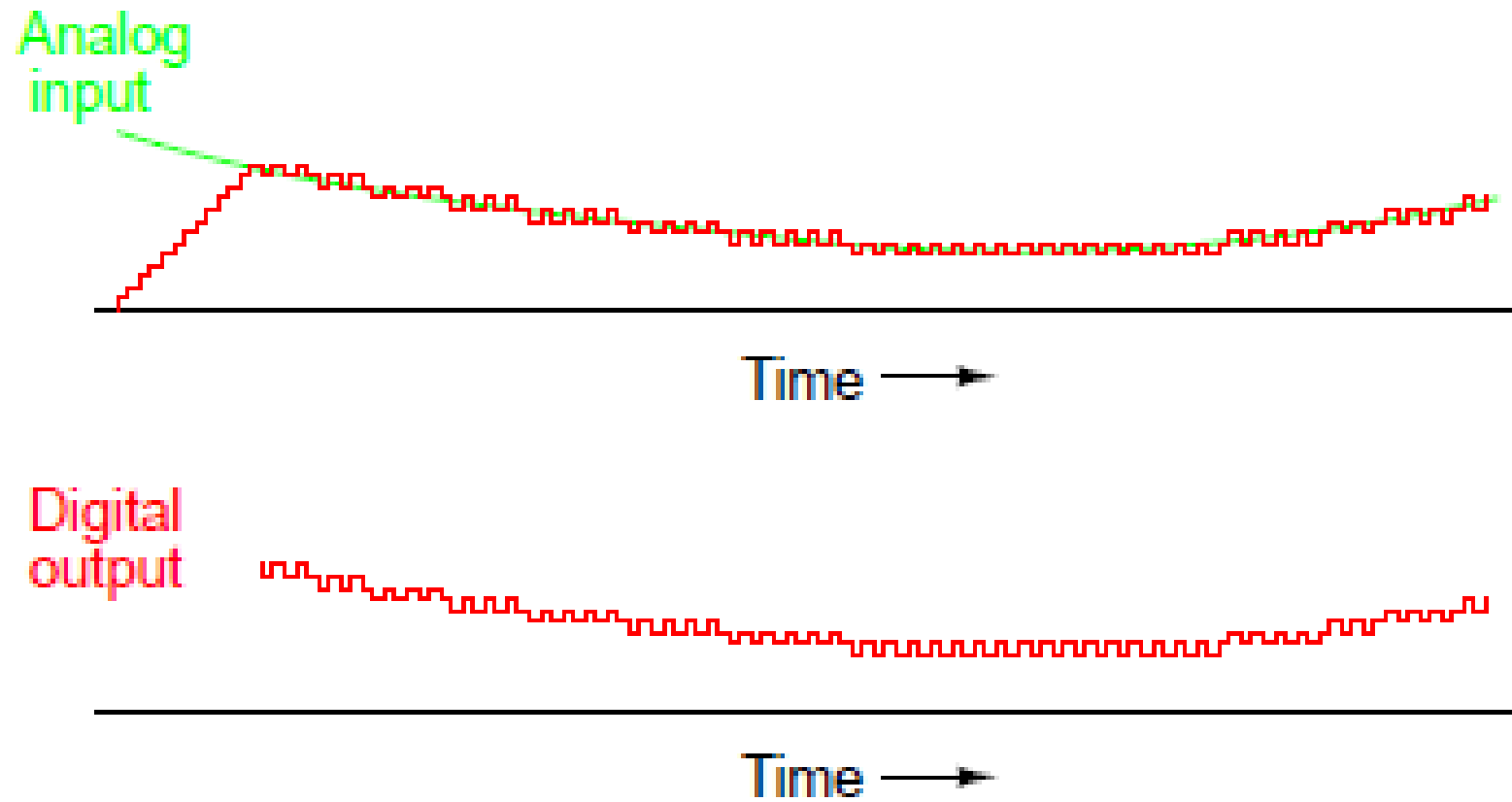
Time to perform a conversion is not constant – depends on the magnitude of V_{in} – large voltage will take more clock cycles to convert.

Start counter at zero every time – makes it a relatively slow conversion process.

The tracking ADC is an improvement on the staircase ADC



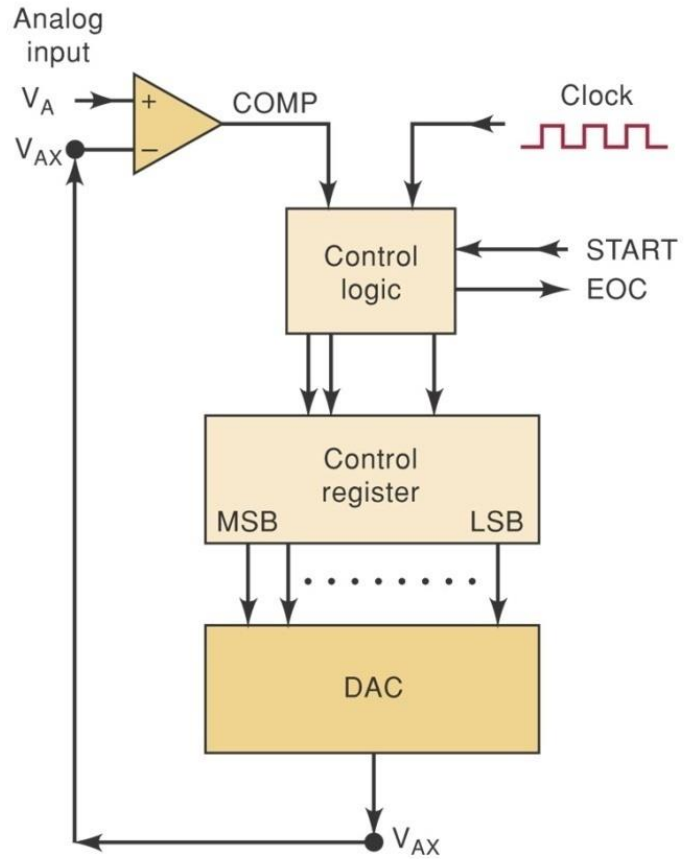
The tracking ADC



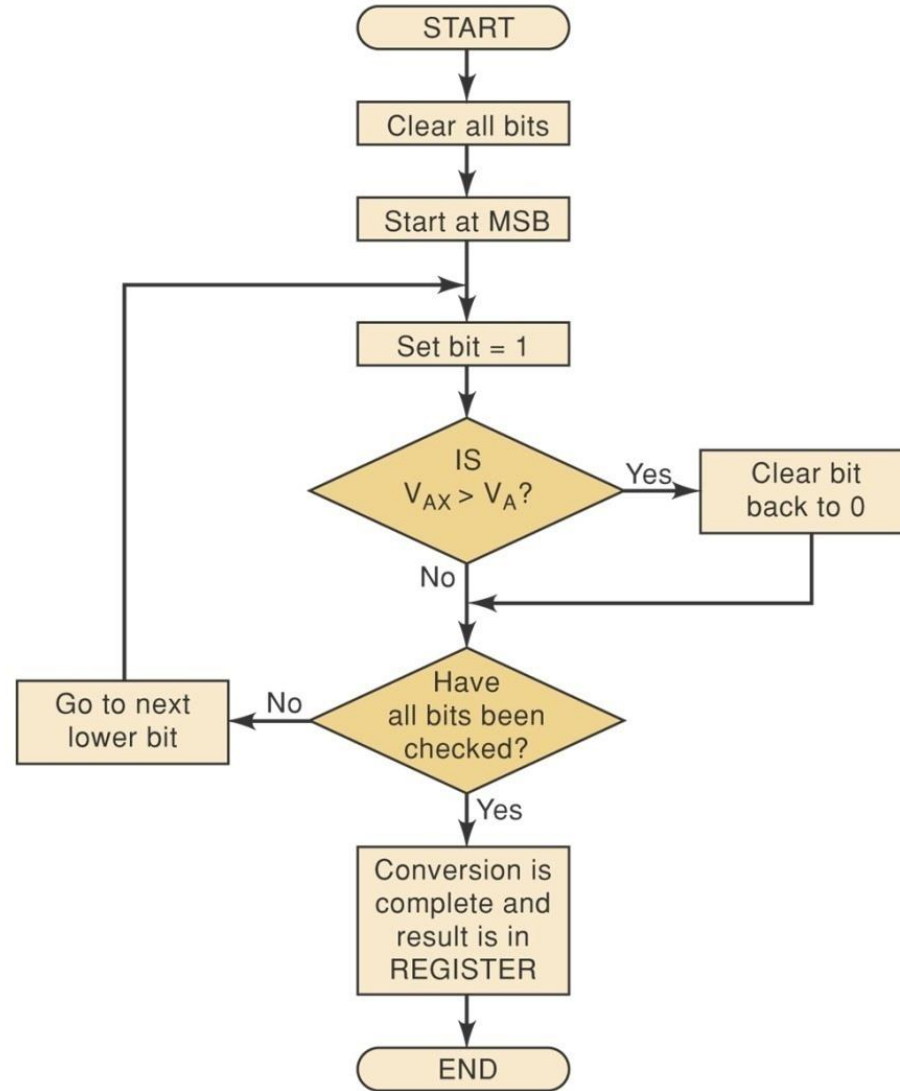
The successive approximation A/D

- One of the most popular types of ADC's
- Binary Search algorithm
- Relatively fast but affordable
- Has a fixed conversion time.
- Uses only one comparator but generates different reference signals for this comparator
- Uses a digital to analog converter (!) to generate a reference signal.
- Digital code is generated by a successive approximation register.
- Operation takes place based on an internal clock cycle

Successive-approximation ADC: (a) simplified block diagram; (b) flowchart of operation.

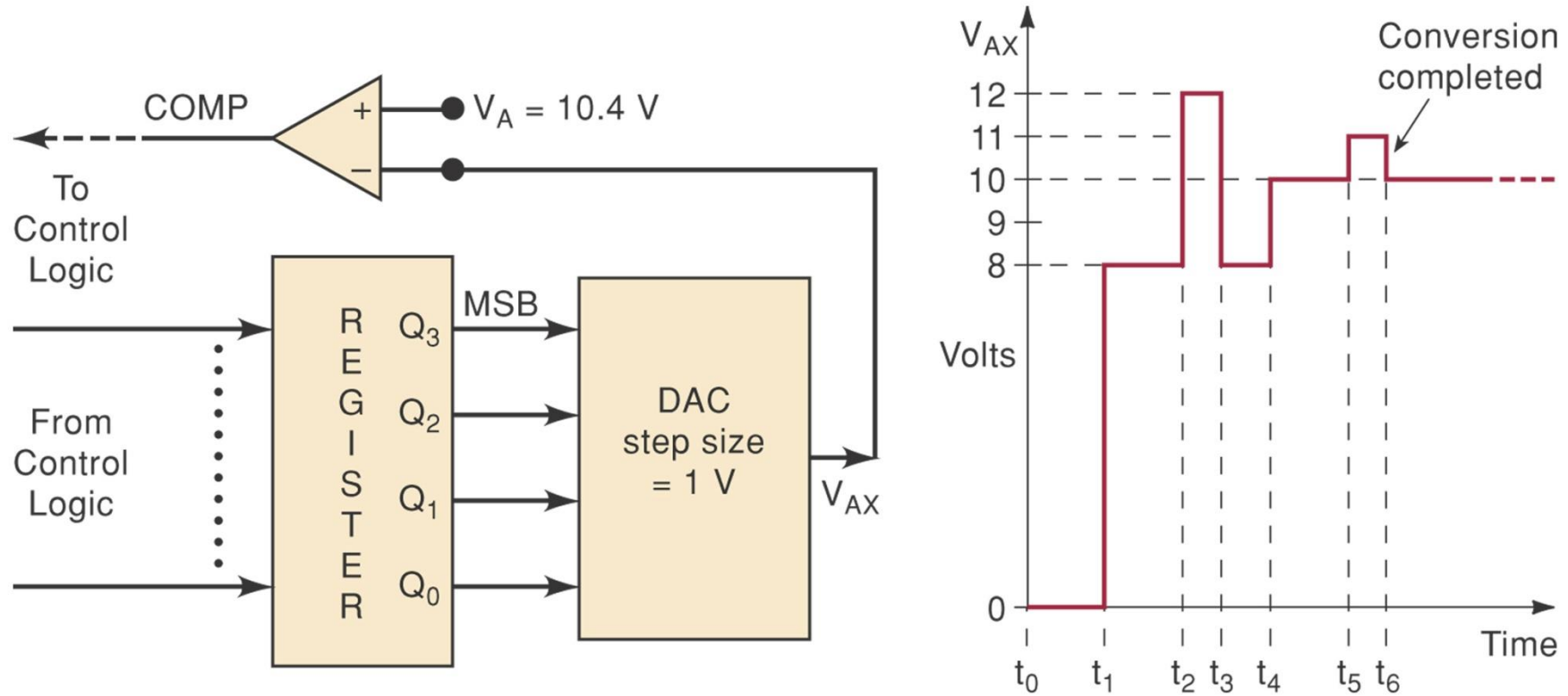


(a)



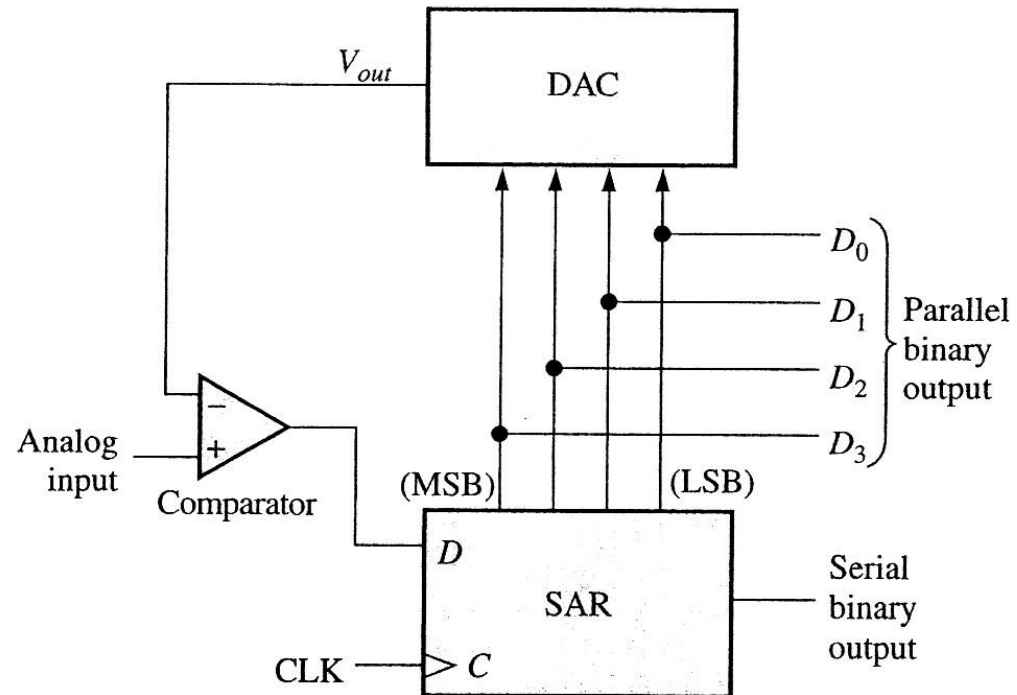
(b)

Illustration of four-bit SAC operation using a DAC step size of 1 V and $V_A = 10.4$ V.



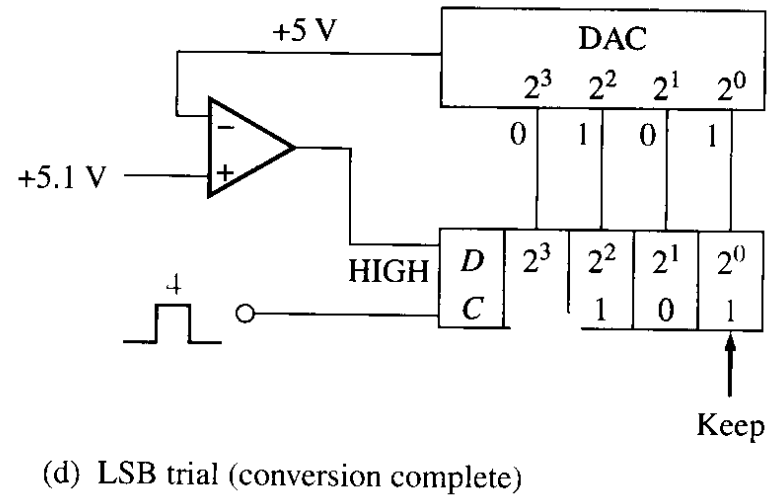
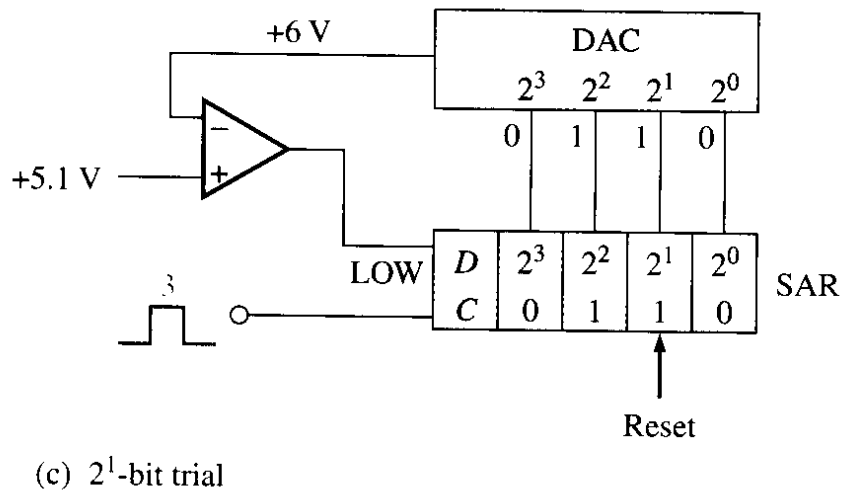
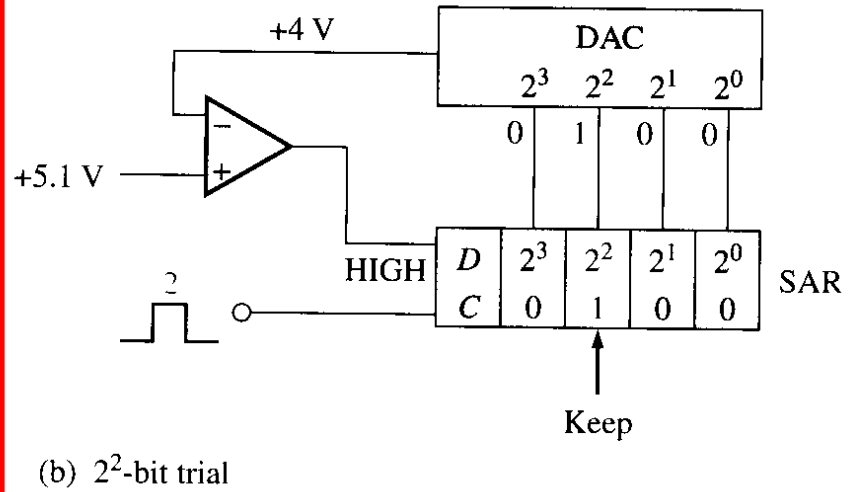
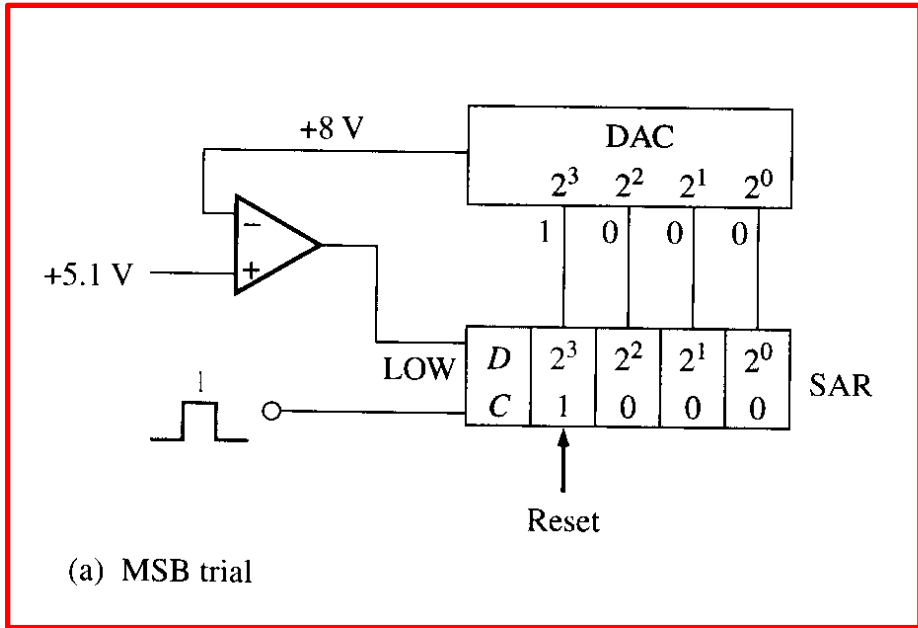
Detailed Operation of a 4-bit successive approximation A/D

- Assume an A/D with a full scale input of 16 V
- Feed an analog input signal of 5.1 V into the A/D
- Represent the signal with a 4-bit binary code.



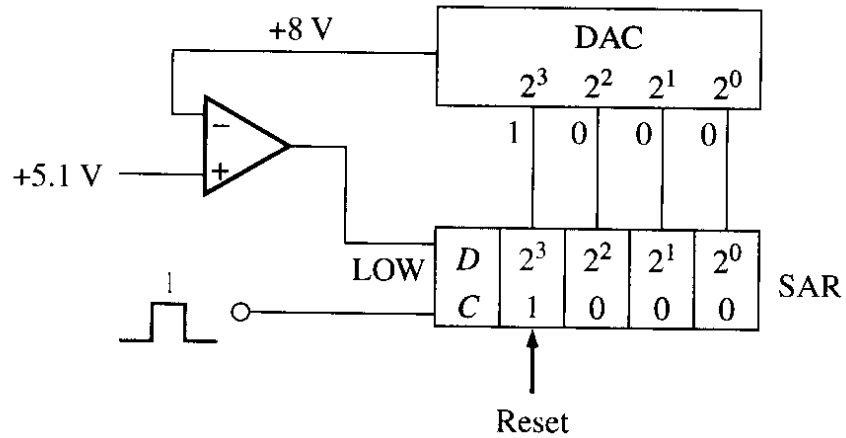
- On the first approximation cycle the MSB (the 2^3 bit) in the successive approximation register (SAR) goes HI and the binary code 1000 is put out to the digital to analog converter. This code will generate an analog output of 8 V from the DAC and this 8 V (half of full scale input) is now put out to the reference input of the comparator.
- The comparator compares the analog input signal (taken to be 5.1 V in this case) to this reference and the output state of the comparator is determined by this comparison.
- If the analog input voltage is lower than the reference (as it is in this case) the output of the comparator will be low and the MSB in the SAR will reset to a zero.

Comparing the MSB

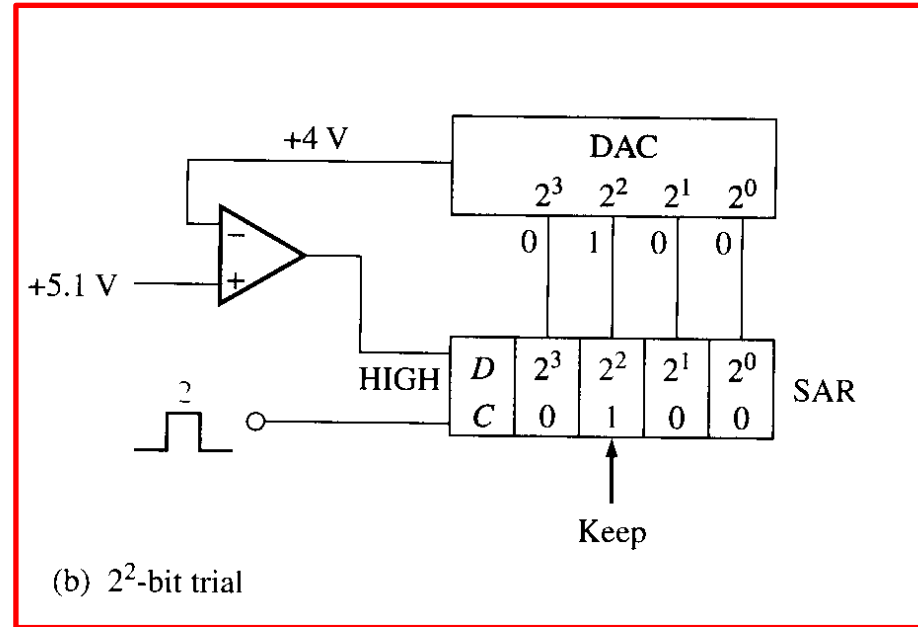


Comparing the second MSB

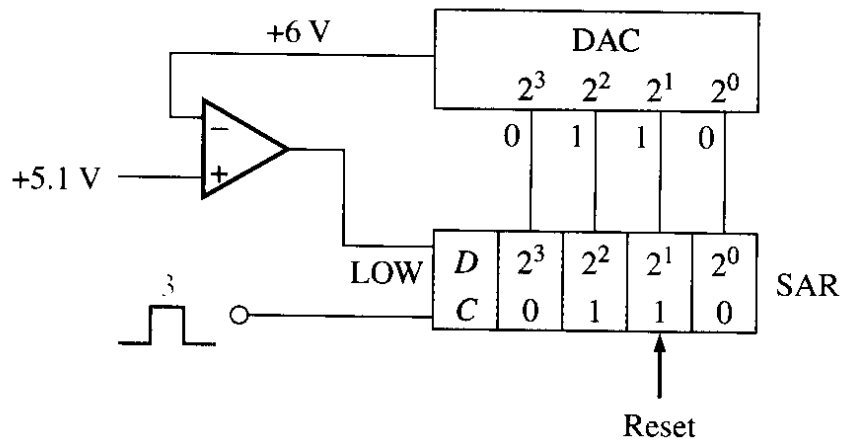
- The second MSB now goes HI and the code 0100 is put out to the DAC. This will produce an output of 4V from the DAC which now becomes the new reference signal of the comparator.
- The analog input signal is again compared to the reference and in this case it exceeds the reference.
- The output of the comparator will thus go HI and the value of “1” for the 2^2 bit is fixed in the SAR.



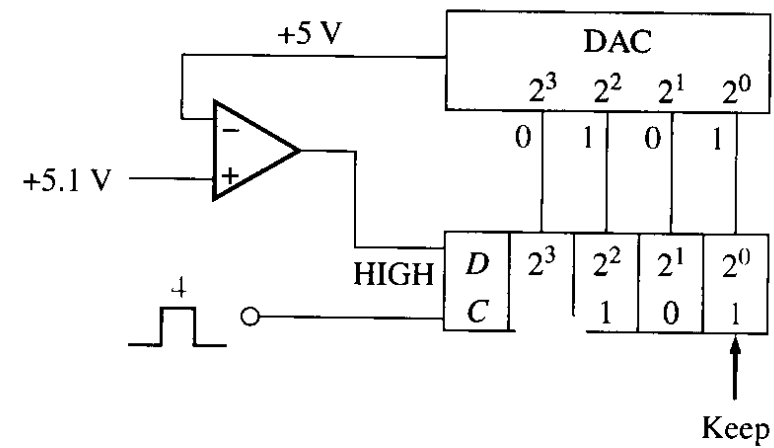
(a) MSB trial



(b) 2²-bit trial



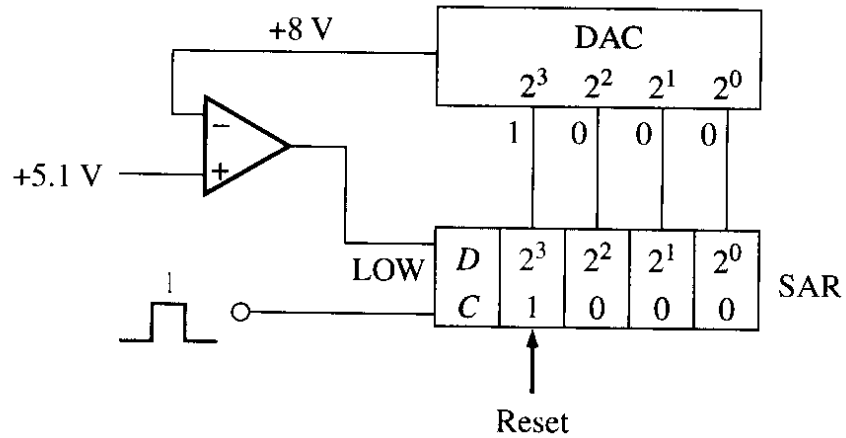
(c) 2¹-bit trial



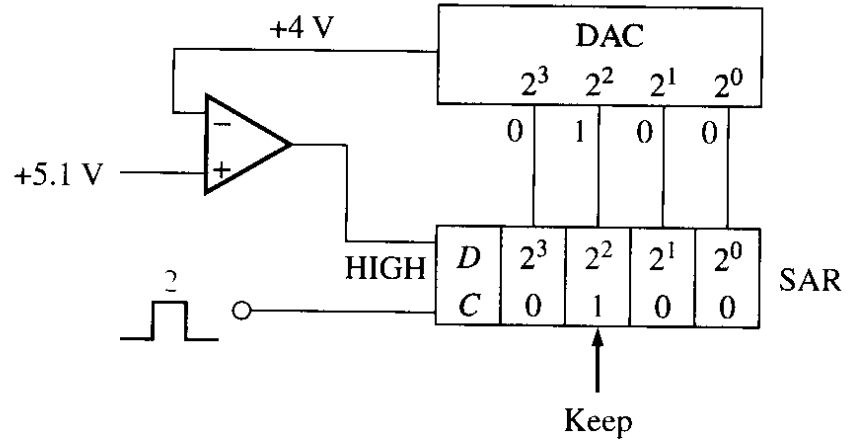
(d) LSB trial (conversion complete)

Comparing the third MSB

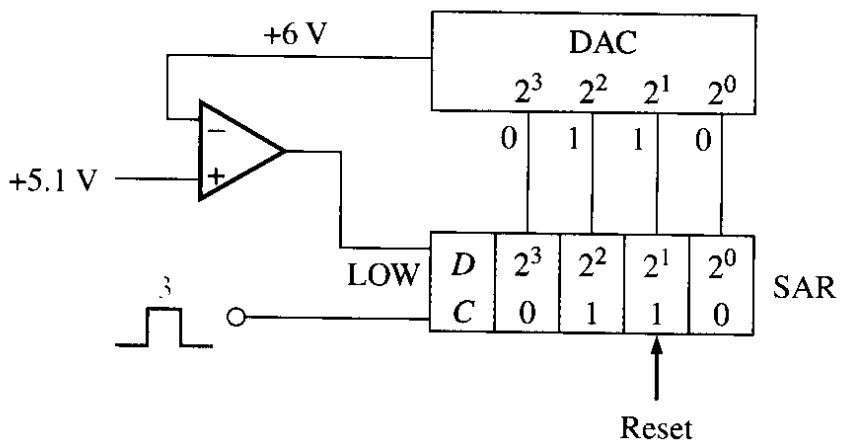
- The third MSB (the 2^1 bit) now goes HI and the code 0110 is put out to the DAC. (Remember the 2^2 bit is now fixed HI). This will produce a 6 volt output which is the new reference signal.
- The input signal is now lower than the reference and the output from the comparator will be LO. This will reset the 2^1 bit to zero again.



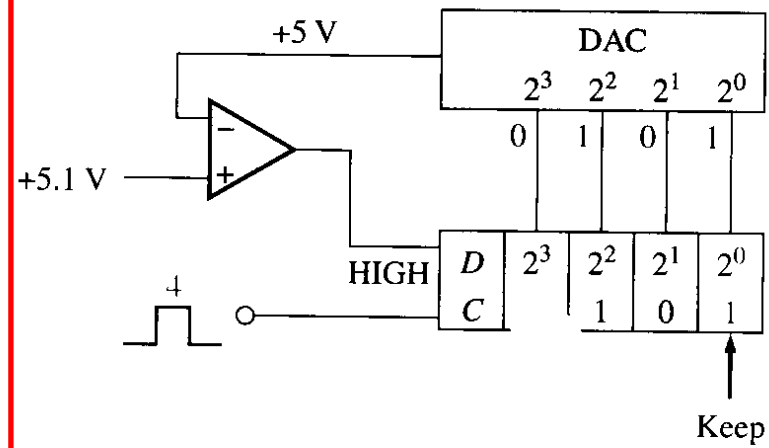
(a) MSB trial



(b) 2²-bit trial



(c) 2¹-bit trial



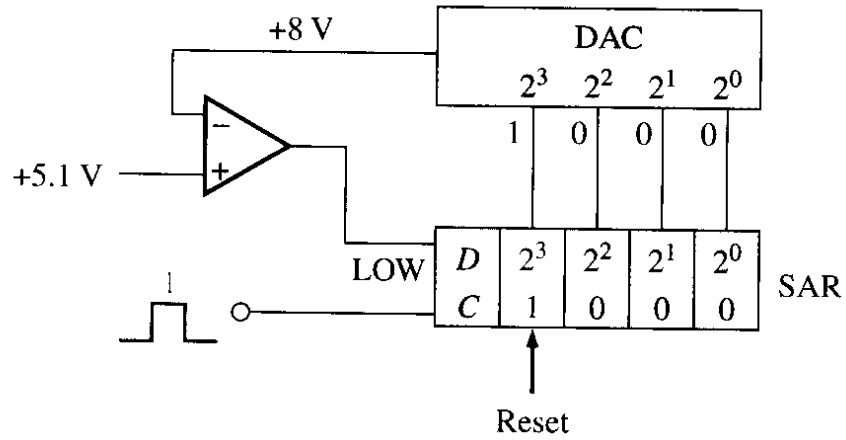
(d) LSB trial (conversion complete)

Comparing the LSB

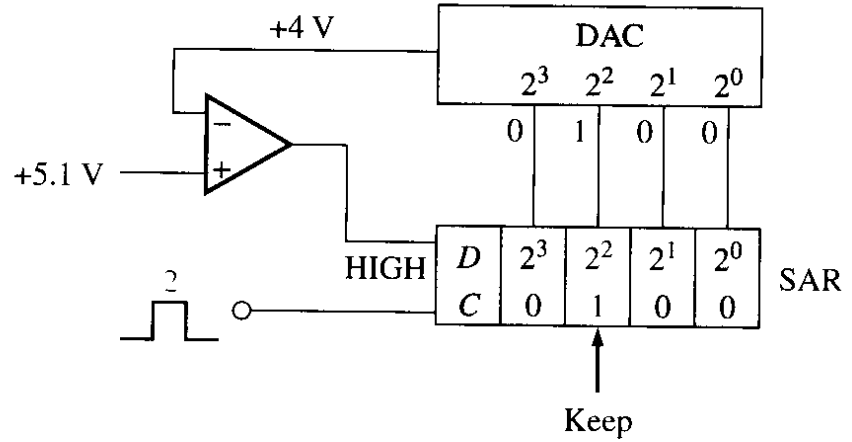
* On the fourth comparison cycle the LSB (2^0 bit) will go HI and the code 0101 put out to the DAC. A reference voltage of 5 V will thus be put on the comparator.

* The input analog voltage is now higher than the reference voltage and the output of the comparator will go HI. This will fix the value of the LSB as “1” in the SAR.

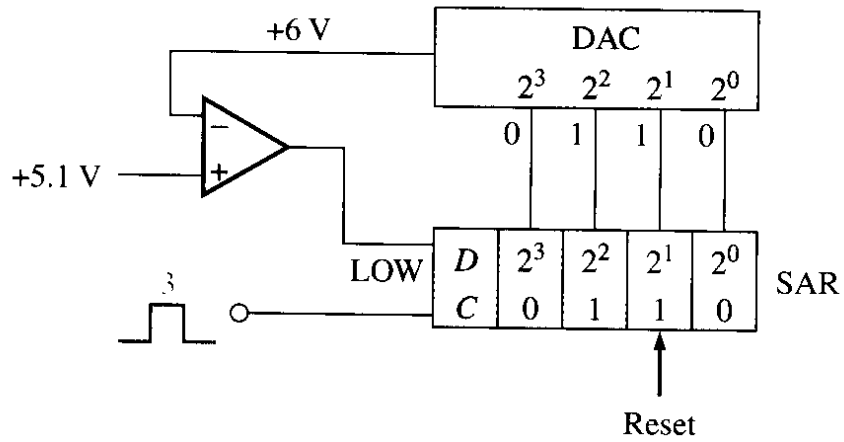
* The conversion process is now complete and the binary number stored in the SAR (0101) can now be read by a microprocessor as representative of the 5.1 V analog input.



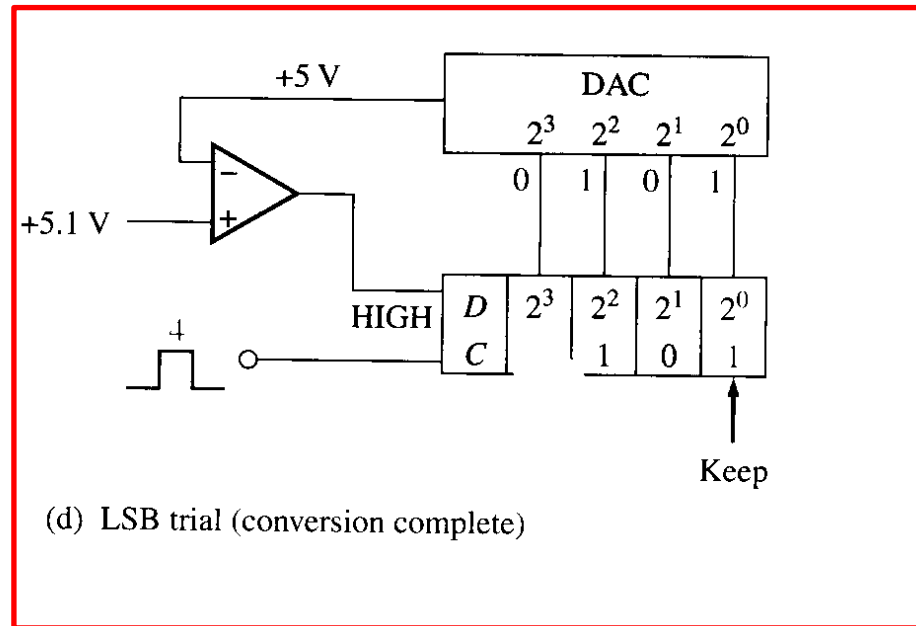
(a) MSB trial



(b) 2^2 -bit trial

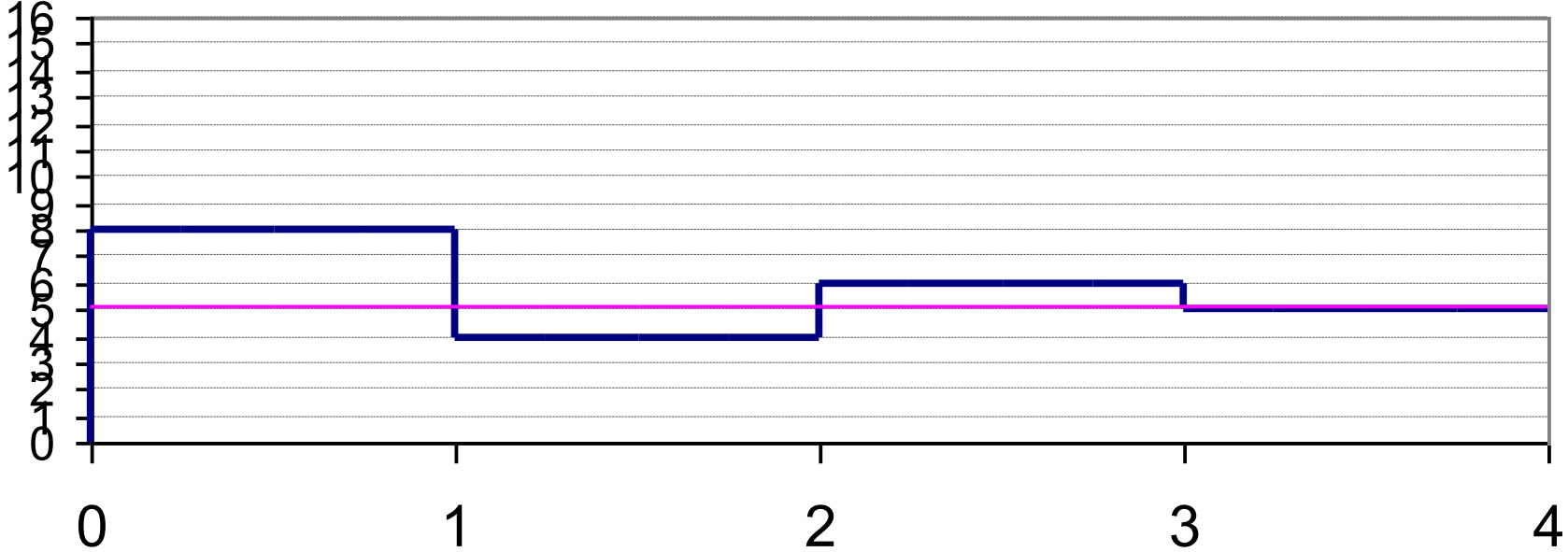


(c) 2^1 -bit trial



(d) LSB trial (conversion complete)

Analog Voltage



— Reference Voltage — Analog Input Voltage

Characteristic of the SAC process:

- The number of comparison steps that need to be done is always equal to the number of output bits in the converter. More bits will thus lead to better accuracy but also a slower conversion process.
- The time to perform a conversion process will always be the same for a particular converter
- This conversion speed is relatively fast compared to some other types of converters (eg. ramp converters) but not as fast as the high priced flash converters. Moderately priced.

Problem:

An eight-bit SAC has a resolution of 20 mV. What will its digital output be for an analog input of 2.17 V?

Given

- ADC resolution = **20 mV per bit = 0.02 V**
- Analog input voltage = **2.17 V**

Step 1: Calculate the digital count

$$\begin{aligned}\text{Digital Output} &= \frac{\text{Input Voltage}}{\text{Resolution}} \\ &= \frac{2.17}{0.02} = 108.5\end{aligned}$$

Since an ADC outputs an integer code, the value is usually rounded to the nearest count:

$$108.5 \approx 109$$

Step 2: Convert to 8-bit binary

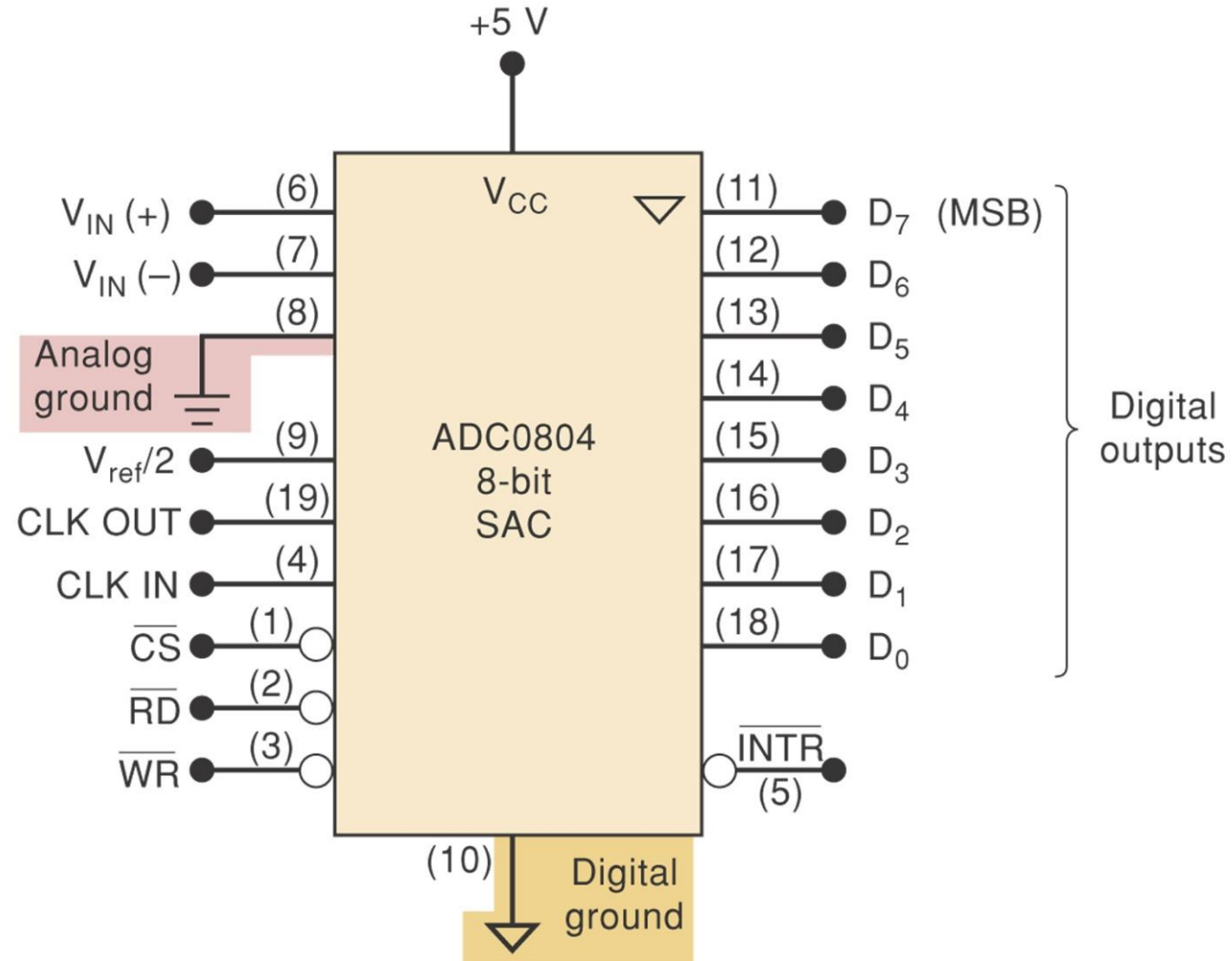
$$109_{10} = 01101101_2$$

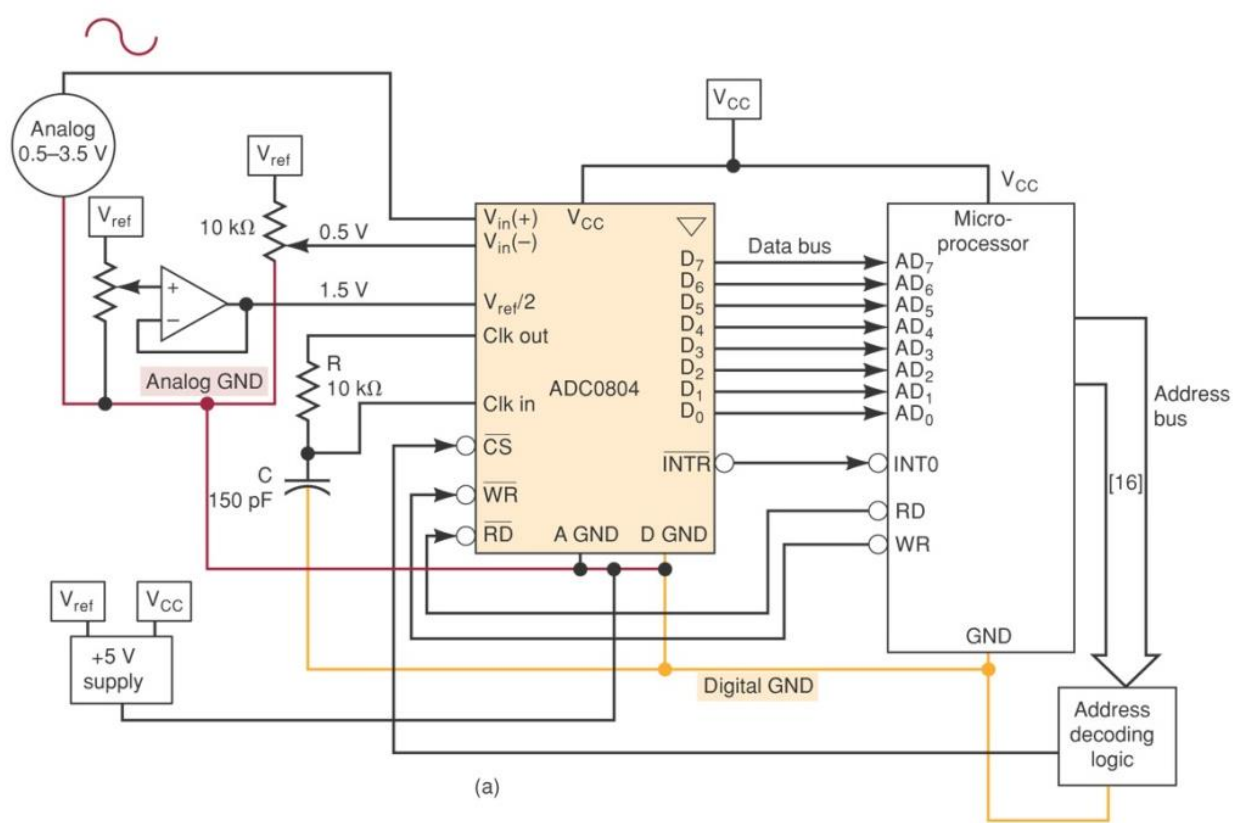
Answer

- Digital output (decimal): **109**
- Digital output (8-bit binary): **01101101**

$$108_{10} = 01101100_2.$$

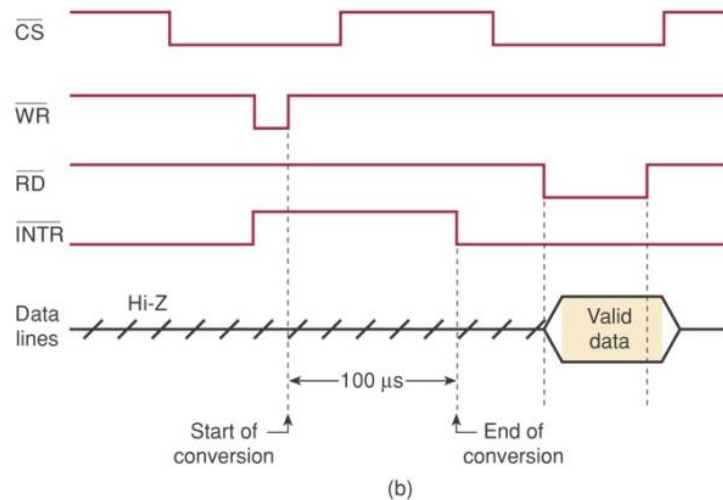
ADC0804 eight-bit successive-approximation ADC with tristate outputs. The numbers in parentheses are in the IC's pin number.





(a) An application of an ADC0804;
 (b) typical timing signals during data acquisition.

An interrupt is used to signal the microprocessor when a conversion is complete



Other A/D Conversion Methods

- There are many other methods of A/D conversion. Each has pros and cons:
 - Dual slope integrating ADC
 - Voltage to frequency ADC, VCO
 - Sigma/delta modulation, PWM
- The method used will depend on the application

Multiplexing

- Multiple analog signals can be converted through time sharing of an ADC
- The process is illustrated in the next slide.
 - The multiplexing clock controls the rate at which the analog signals are switched to the ADC
 - CMOS semiconductor switches can be used to reduce switch delay time
- The ADC0808 can multiplex eight different analog inputs to one ADC

Multiplexing

Multiplexing refers to using a **multiplexer (MUX)** to allow multiple input signals to share a single ADC.

How it works

A multiplexer acts like an electronic switch:

- Several analog input channels are connected to the multiplexer.
- The microcontroller selects one channel at a time.
- The selected signal is routed to the ADC.
- The ADC converts that signal to a digital value.
- The microcontroller then switches to the next channel and repeats the process.

Multiplexing

Example

Suppose you have:

- 1 ADC
- 4 sensors (temperature, light, pressure, and humidity)

Instead of needing 4 separate ADCs, a 4-to-1 multiplexer lets the microcontroller read:

- 1.Sensor 1 → ADC conversion
 - 2.Sensor 2 → ADC conversion
 - 3.Sensor 3 → ADC conversion
 - 4.Sensor 4 → ADC conversion
- very quickly, one after another.

Multiplexing

Why is multiplexing used

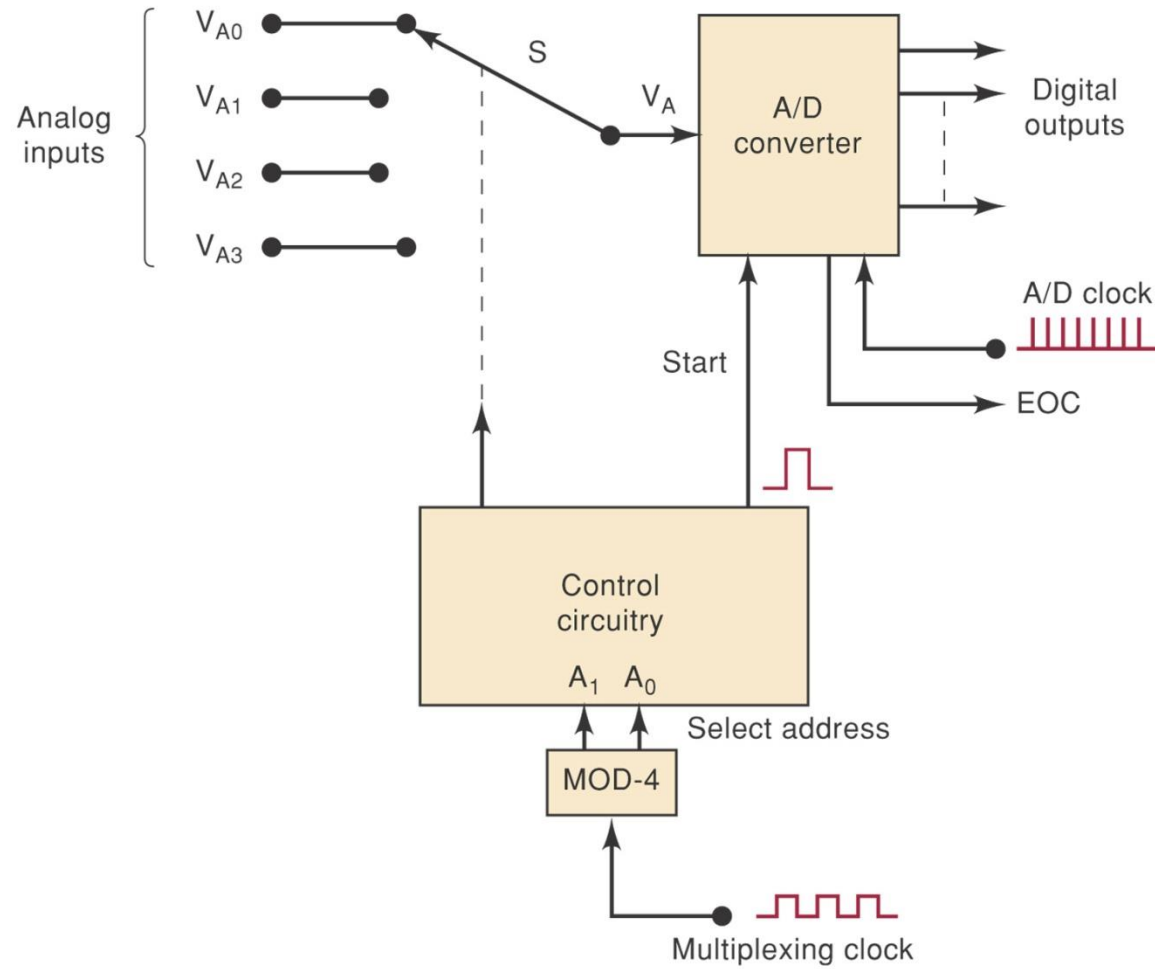
- Reduces hardware cost
- Saves ADC resources
- Allows many sensors to be connected to a microcontroller with limited ADC channels
- Simplifies circuit design

In microcontrollers

Many microcontrollers have a built-in analog multiplexer connected to the ADC. For example, if a microcontroller has 8 analog input pins but only one ADC core, an internal multiplexer selects which pin is connected to the ADC at any moment.



Conversion of four analog inputs by multiplexing through one ADC.

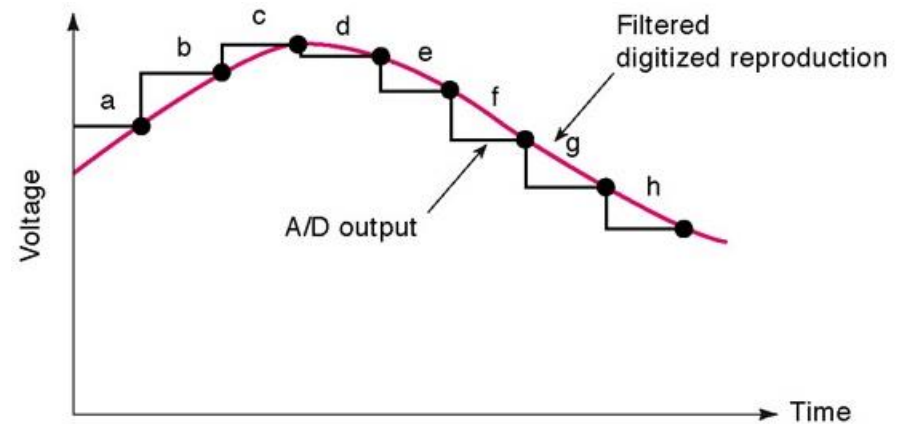
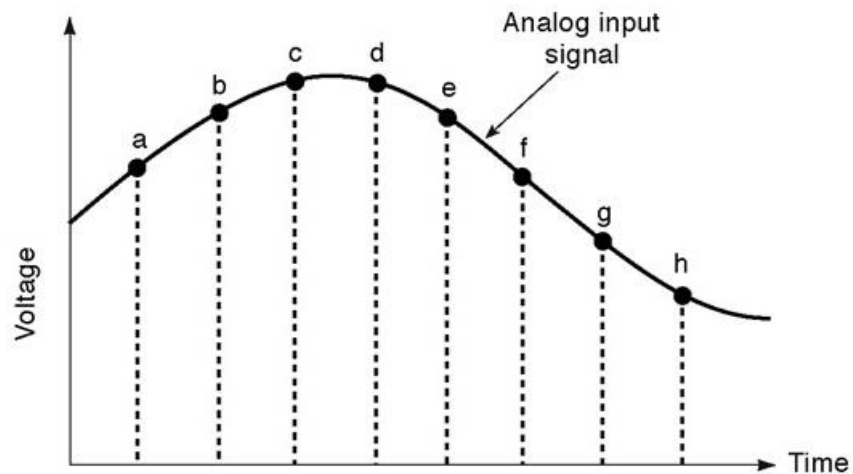


Digital Storage Oscilloscope

- Makes use of A/D converters
- Advantages of the DSO over the CRO
 - Waveform storage
 - Stored waveform display for comparison
 - Store and display waveforms before the trigger point
 - Print waveforms or transfer to a PC

Data Acquisition

- Digitizing analog data and transferring to memory is data acquisition
- Acquiring a single data point value is sampling
- Reconstructing a digitized signal



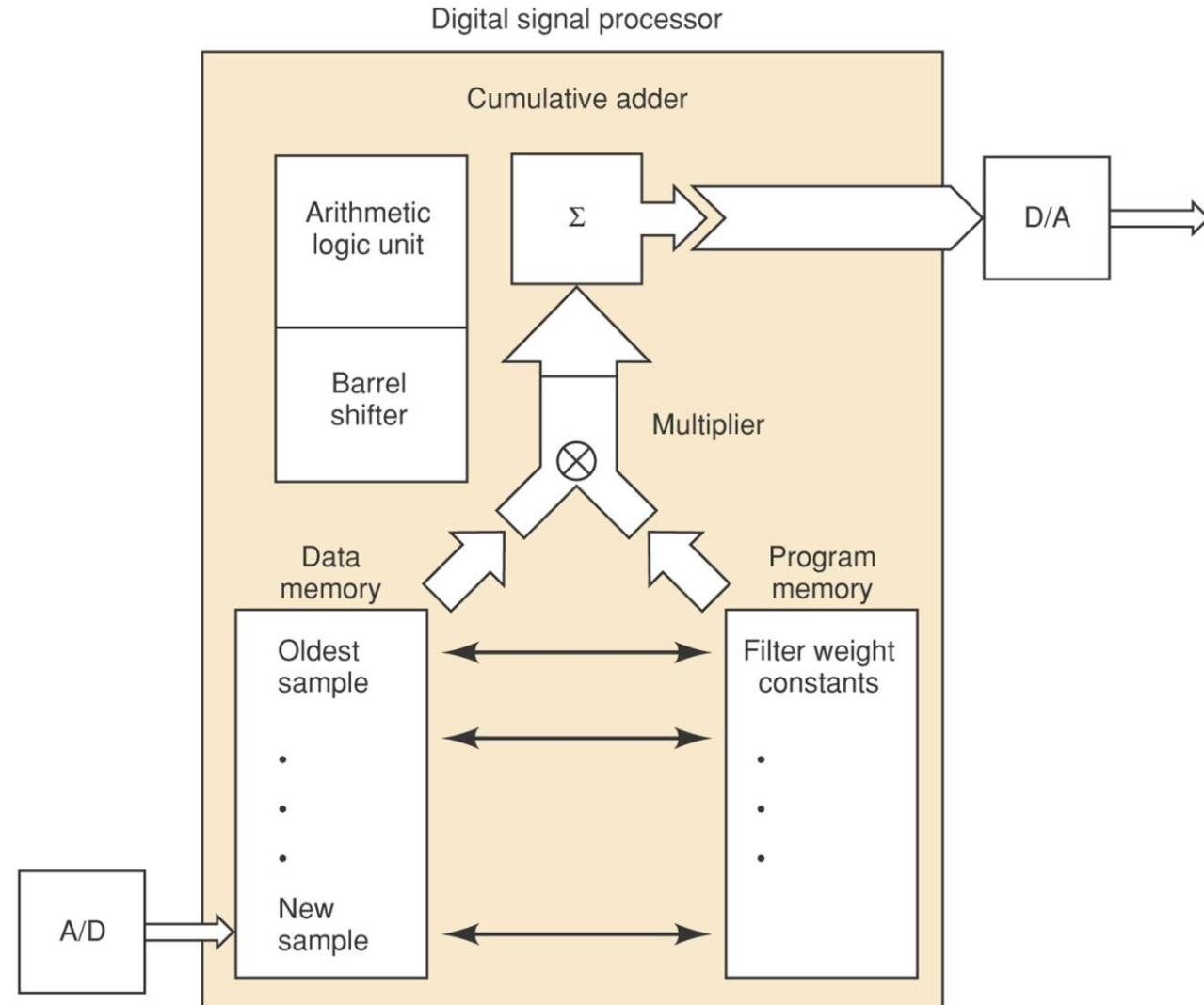
Digital Signal Processing

- Specialized microprocessor optimized for repetitive calculations on streams of digitized data
- DSP is used frequently in filtering and conditioning of analog signals
 - Perform the same function as analog filters but allow greater flexibility
 - Also music generation, effects pedals!

Digital Signal Processing

- Digital filtering process
 - Read the newest sample from A/D
 - Replace the oldest sample with the new one
 - Multiply each of the 256 samples by corresponding weight constant
 - Add all products
 - Output the resulting sum of products to the D/A

Digital signal processor architecture.



D-A Oversampling

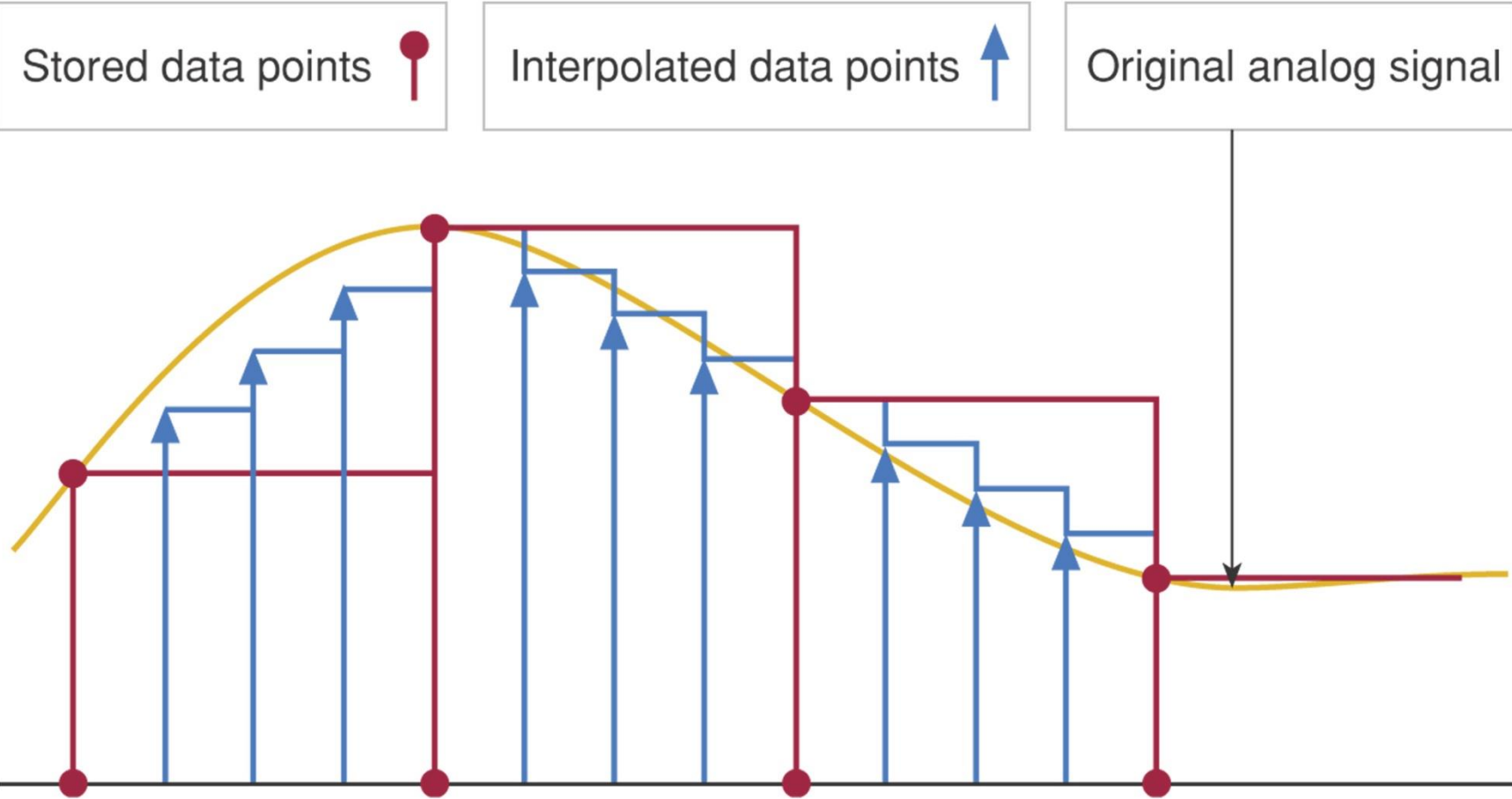
Use interpolation to generate intermediate samples.

Output sample rate is increased.

Can use a simpler LP reconstruction filter as sampling Clock much greater than signal frequencies.

Extreme case is the 1-bit Audio DAC.

Inserting an interpolated data point into a digital signal to reduce harmonic content close to the frequencies of interest.



Audio CODEC

Stereo Audio A-D and D-A all in one package.

