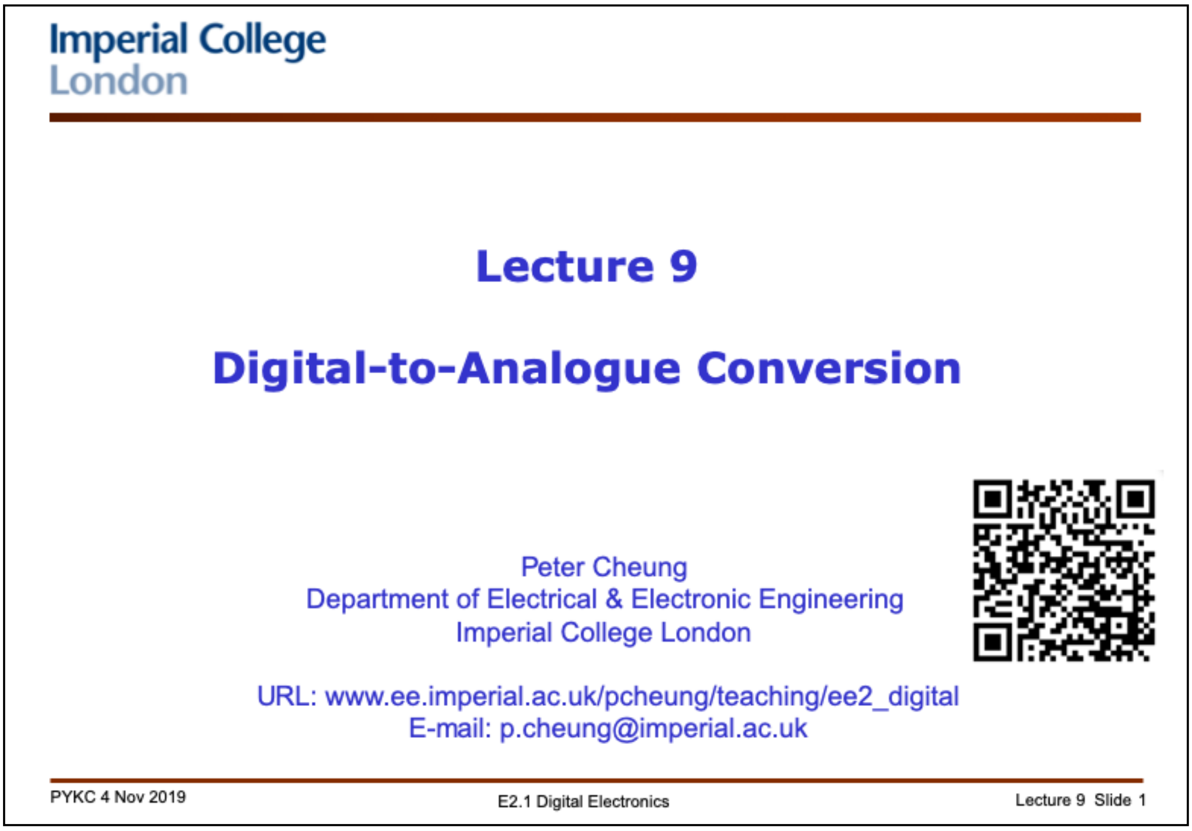


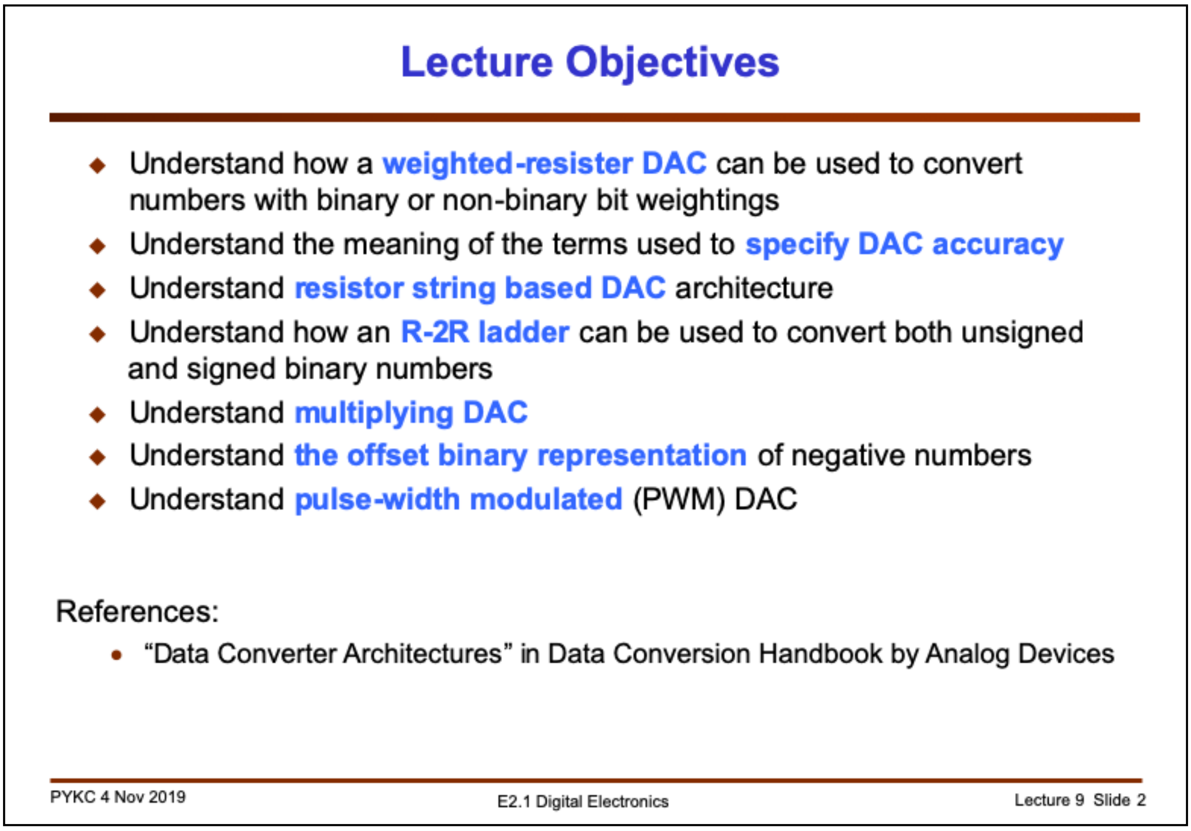
Imperial College London

Lecture 9

Digital-to-Analogue Conversion

Peter Cheung
Department of Electrical & Electronic Engineering
Imperial College London

URL: www.ee.imperial.ac.uk/pcheung/teaching/ee2_digital
E-mail: p.cheung@imperial.ac.uk



Lecture Objectives

- Understand how a **weighted-resister DAC** can be used to convert numbers with binary or non-binary bit weightings
- Understand the meaning of the terms used to **specify DAC accuracy**
- Understand **resistor string based DAC** architecture
- Understand how an **R-2R ladder** can be used to convert both unsigned and signed binary numbers
- Understand **multiplying DAC**
- Understand **the offset binary representation** of negative numbers
- Understand **pulse-width modulated** (PWM) DAC

References:

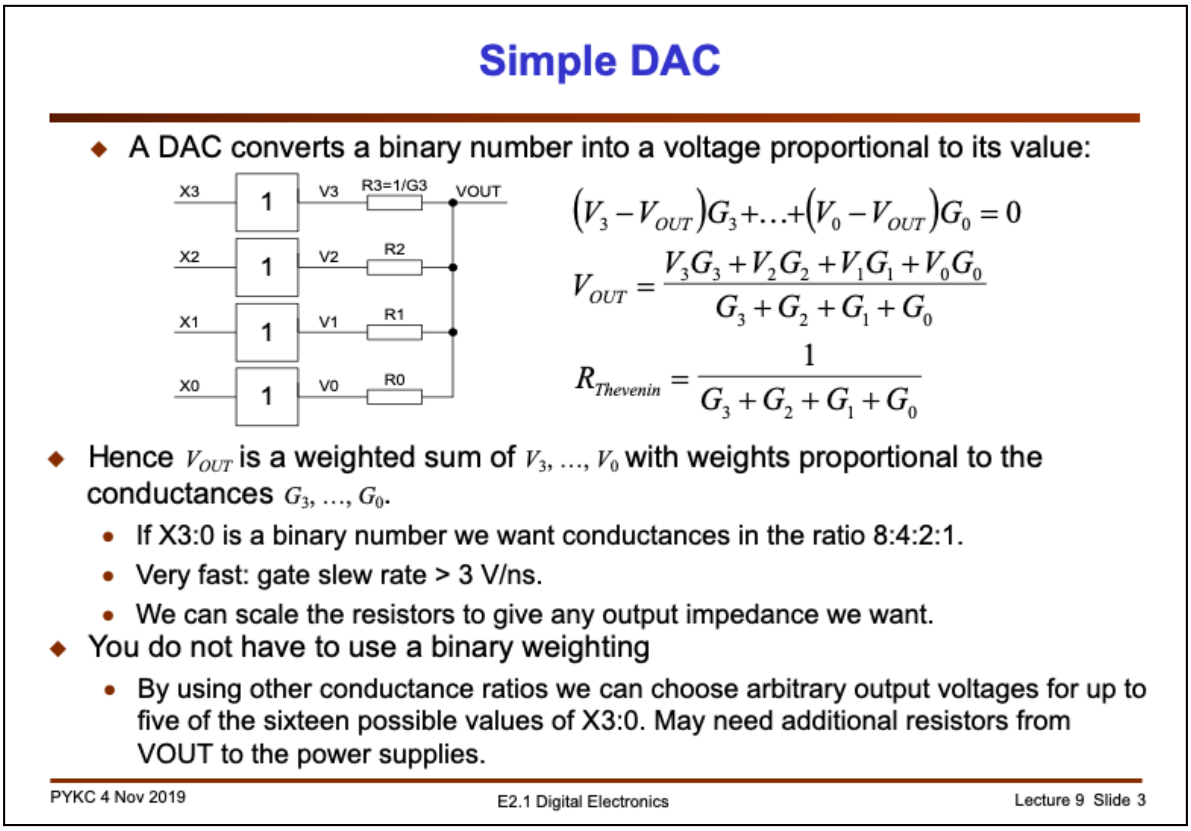
- “Data Converter Architectures” in Data Conversion Handbook by Analog Devices

PYKC 4 Nov 2019 E2.1 Digital Electronics Lecture 9 Slide 2

Although digital technology dominates modern electronic systems, the physical world remains mostly analogue in nature. The most important components that link the analogue world to digital systems are analogue-to-digital and digital-to-analogue converters (ADCs and DACs).

In the next two lectures, we will consider how these converters work, their limitations and how to read their data sheets. Designing ADC and DAC requires both knowledge of analogue and digital designs. We are only interested in examining the basic principles of these converters and learn how to use them. We will NOT consider how they are designed. Detail ADC/DAC designs at transistor level will be considered in 3rd and 4th years on other course modules.

Analog Devices is a US company that has the largest range of converter products. They publish an excellent handbook which is available through the course webpage. Relevant to this lecture is the chapter on “Chapter 3: Data Converter Architectures”.



Simple DAC

- A DAC converts a binary number into a voltage proportional to its value:

X3
1
V3
R3=1/G3
VOUT
X2
1
V2
R2
X1
1
V1
R1
X0
1
V0
R0

$$(V_3 - V_{OUT})G_3 + \ldots + (V_0 - V_{OUT})G_0 = 0$$

$$V_{OUT} = \frac{V_3G_3 + V_2G_2 + V_1G_1 + V_0G_0}{G_3 + G_2 + G_1 + G_0}$$

$$R_{Thevenin} = \frac{1}{G_3 + G_2 + G_1 + G_0}$$

- Hence V_{OUT} is a weighted sum of $V_3, \ldots, V_0$ with weights proportional to the conductances $G_3, \ldots, G_0$.
  - If X3:0 is a binary number we want conductances in the ratio 8:4:2:1.
  - Very fast: gate slew rate > 3 V/ns.
  - We can scale the resistors to give any output impedance we want.
- You do not have to use a binary weighting
  - By using other conductance ratios we can choose arbitrary output voltages for up to five of the sixteen possible values of X3:0. May need additional resistors from VOUT to the power supplies.

PYKC 4 Nov 2019 E2.1 Digital Electronics Lecture 9 Slide 3

The simplest DAC can be constructed using a number of resistors with binary weighted values. X[3:0] is the 4-bit digital value to be converter to an analogue voltage Vout. The 4-bit number is used as input to buffer circuits (the rectangular blocks labelled ‘1’). The outputs of the four buffers are V[3:0] respectively.

Using Kirchhoff current law, the current at node Vout sums to zero, and this gives the first equation. (G_0 is $1/R_0$ etc.) Rearranging the equation produces the equation for Vout.

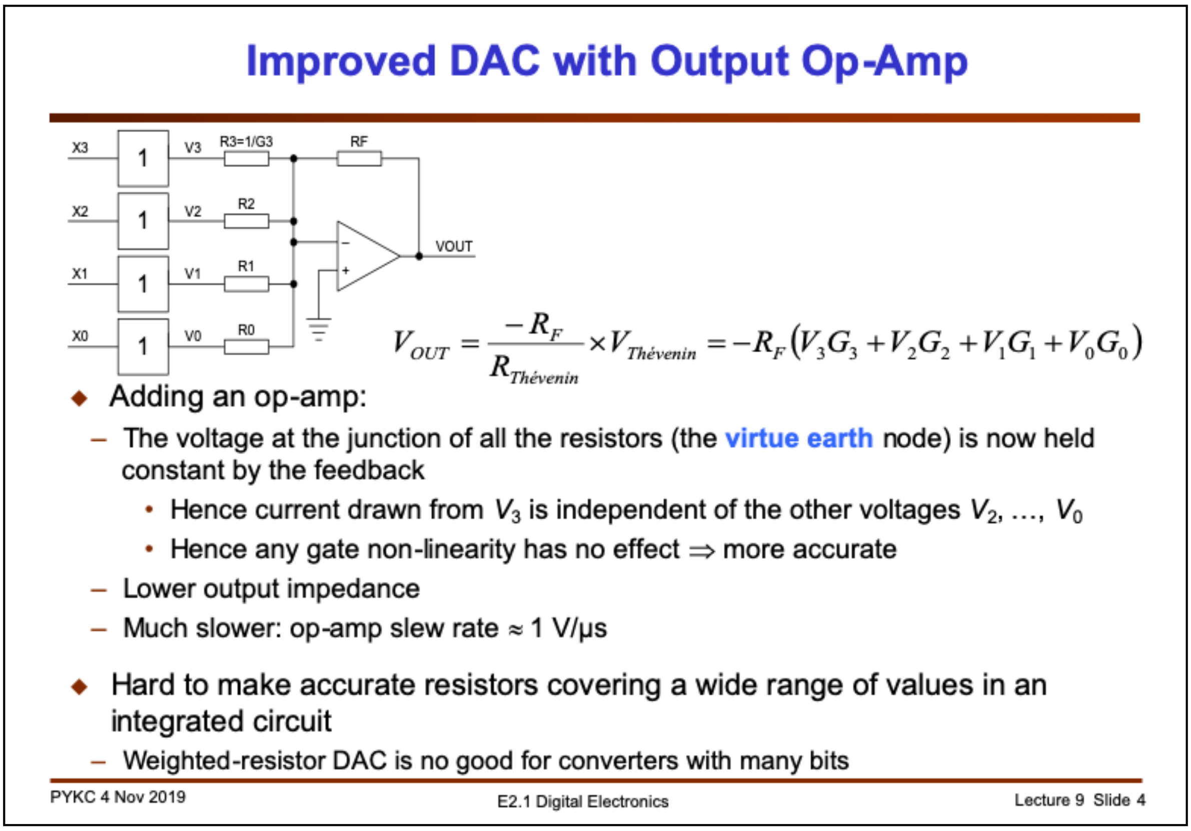
The digital value X[3:0] can therefore be converted to an analogue voltage in the correct **binary weighting** if G3:G2:G1:G0 have the ratio of 8:4:2:1.

Since the digital buffer is very fast and the resistor network has no (or negligible) capacitance or inductance, this DAC can be very fast. However, this DAC has two problems:

1. The output impedance of the DAC is the Thevenin equivalent circuit resistance. Choosing too high a resistance value results in the DAC having a high output impedance; choosing too low a resistance value draws lots of current from the buffers and is inefficient on power.

2. It requires very large resistance ratio if the number of bits of X is large. For example, for a 10-bit DAC, the ratio is 1024:1. Such a DAC is difficult and expensive to manufacture.

Instead of only using binary weighting, it is possible for you to choose five arbitrary Vout values. If you add another resistor R4 connecting from Vout to the power supply, and set X[3:0] to 0000, 0001, 0010, 0100 and 1000, you can easily work out the required value of the resistances in order to give you the five arbitrary voltages.



Improved DAC with Output Op-Amp

$$V_{OUT} = \frac{-R_F}{R_{Th\acute{e}venin}} \times V_{Th\acute{e}venin} = -R_F\left(V_3G_3 + V_2G_2 + V_1G_1 + V_0G_0\right)$$

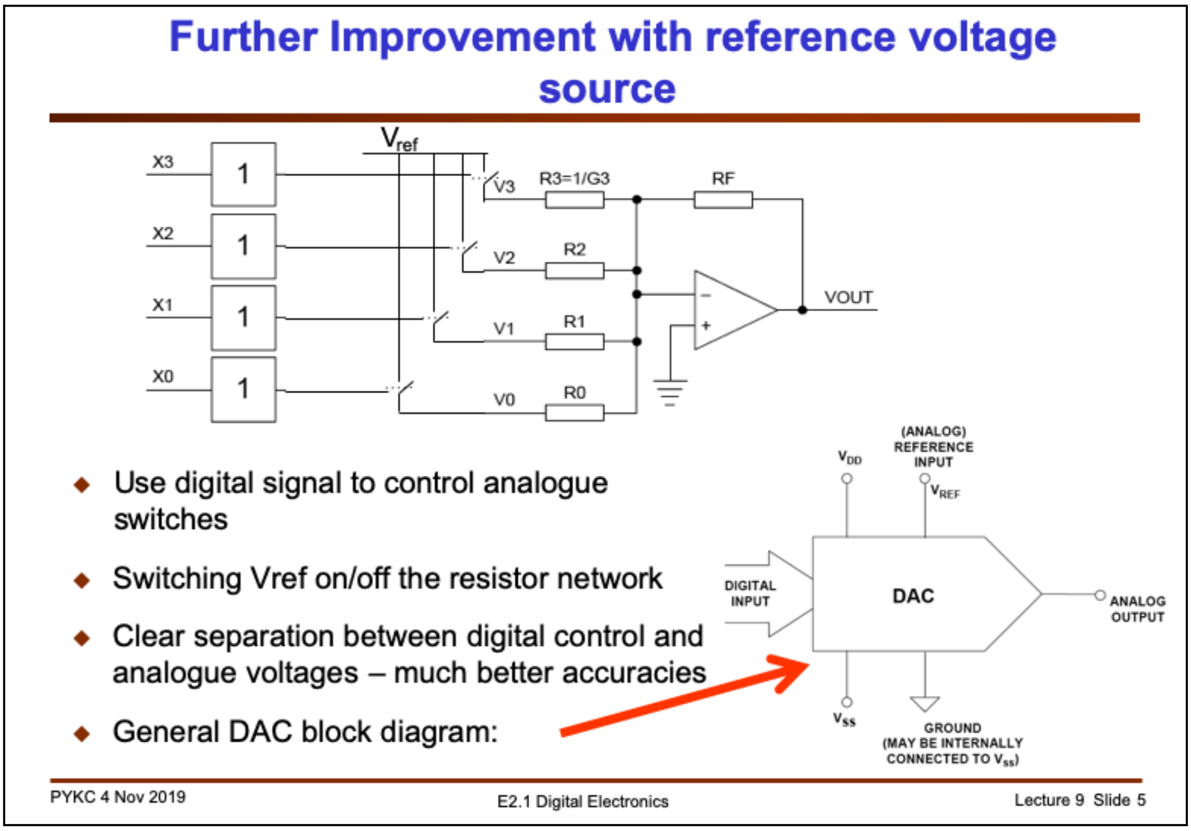
- Adding an op-amp:
  - The voltage at the junction of all the resistors (the **virtue earth** node) is now held constant by the feedback
    - Hence current drawn from V_3 is independent of the other voltages $V_2, \ldots, V_0$
    - Hence any gate non-linearity has no effect ⇒ more accurate
  - Lower output impedance
  - Much slower: op-amp slew rate ≈ 1 V/μs
- Hard to make accurate resistors covering a wide range of values in an integrated circuit
  - Weighted-resistor DAC is no good for converters with many bits

PYKC 4 Nov 2019 E2.1 Digital Electronics Lecture 9 Slide 4

The high output impedance of the previous circuit can be circumvented using an operational amplifier. Shown here is a summing amplifier. Vout is given by this simple linear equation. The output impedance is that of the op amp and is very low.

Unfortunately the output voltage of this circuit cannot change very fast. It is limited by the **slew rate** of the op amp. (Slew rate is a measure of how fast the output voltage can change, and it is in units of V/sec.)

Making binary weighted resistors is still difficult and expensive of the number of bits in the DAC is high.

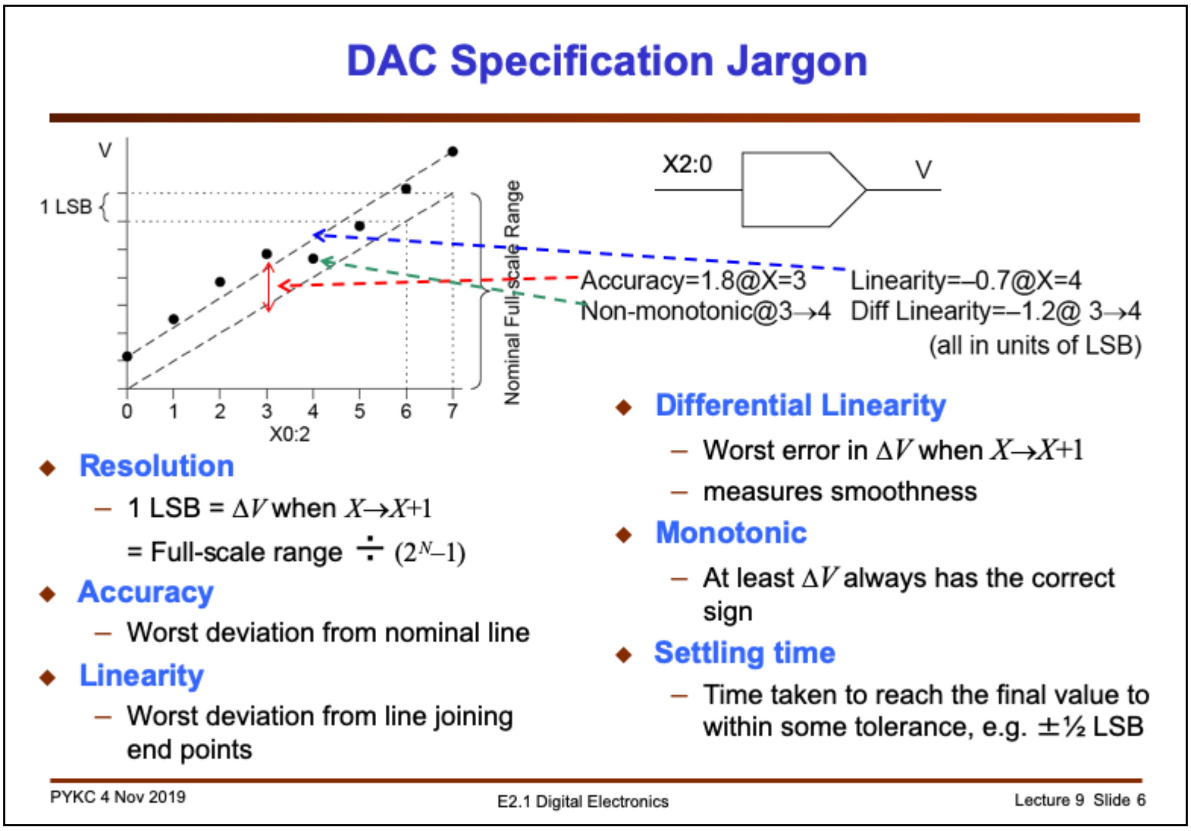


Further Improvement with reference voltage source

- Use digital signal to control analogue switches
- Switching Vref on/off the resistor network
- Clear separation between digital control and analogue voltages – much better accuracies
- General DAC block diagram:

PYKC 4 Nov 2019 E2.1 Digital Electronics Lecture 9 Slide 5

Instead of driving the resistor network directly from the digital output, which is not very accurate, most DAC actually use the digital signal to control electronic switches which switch in or out a reference voltage Vref. This reference voltage can be made very accurate, thus providing accurate output voltage values.



DAC Specification Jargon

Accuracy=1.8@X=3
Non-monotonic@3→4
Linearity=–0.7@X=4
Diff Linearity=–1.2@ 3→4
(all in units of LSB)

- **Resolution**
  - 1 LSB = ΔV when $X \to X+1$
    = Full-scale range ÷ (2^N-1)
- **Accuracy**
  - Worst deviation from nominal line
- **Linearity**
  - Worst deviation from line joining end points
- **Differential Linearity**
  - Worst error in ΔV when $X \to X+1$
  - measures smoothness
- **Monotonic**
  - At least ΔV always has the correct sign
- **Settling time**
  - Time taken to reach the final value to within some tolerance, e.g. ±½ LSB

PYKC 4 Nov 2019 E2.1 Digital Electronics Lecture 9 Slide 6

Here are the important specifications found in a datasheet that defines the performance of a DAC. Here we use the line from full range value to the origin as reference. We will express all voltage in terms of the delta-v corresponding to one LSB.

Resolution – the voltage step equivalent to one least significant bit (1 LSB) of the digital input. Assuming that the input is an N-bit number, then resolution of the DAC is the same as: **(full-scale voltage) / (2^N-1)**.

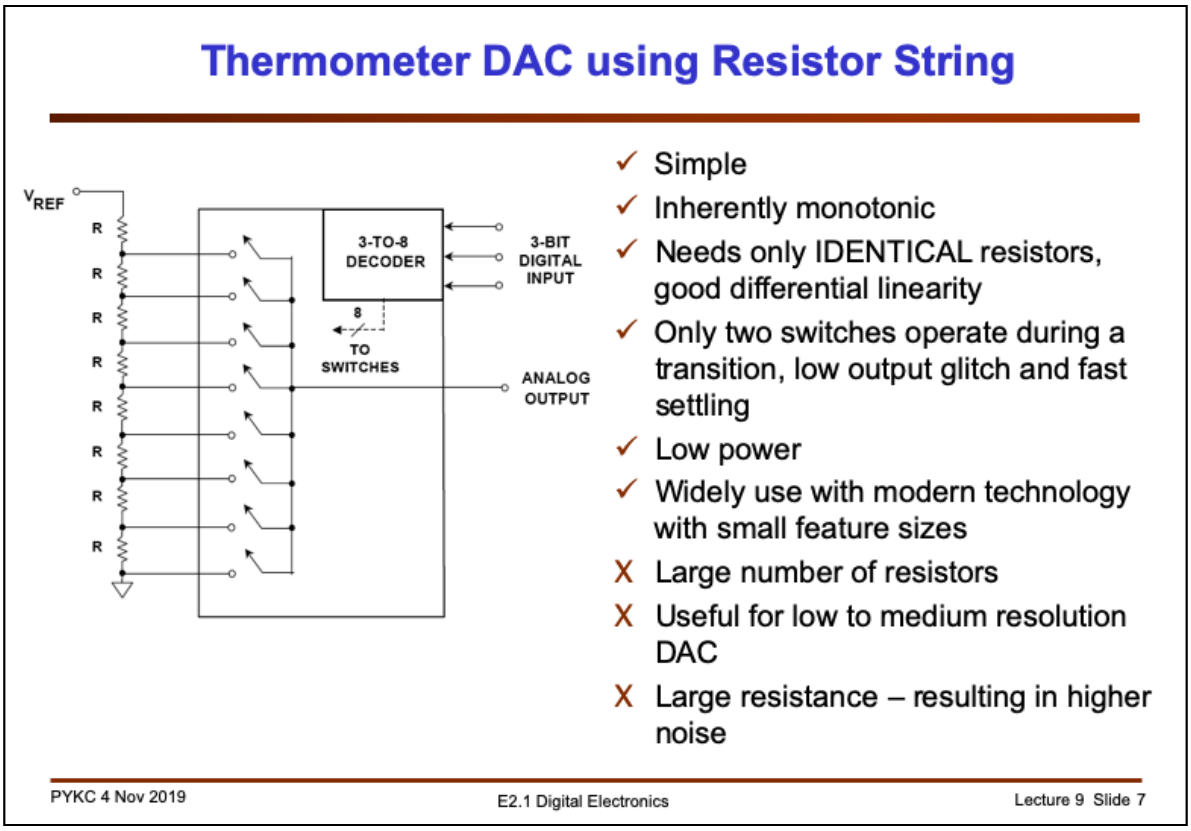
Accuracy – maximum error as compared to the perfect reference line (red).

Linearity – Instead of using the reference line, we can join to max-point with the min-point to form another straight line. Linearity is the maximum deviation from this new line.

Differential Linearity – Worse case error as you step from X to X+1 for all values of X.

Monotonic DAC – One that always goes up as the input number X[3:0] increases.

Settling time – Time taken to reach final value within ±1 LSB as input changes.



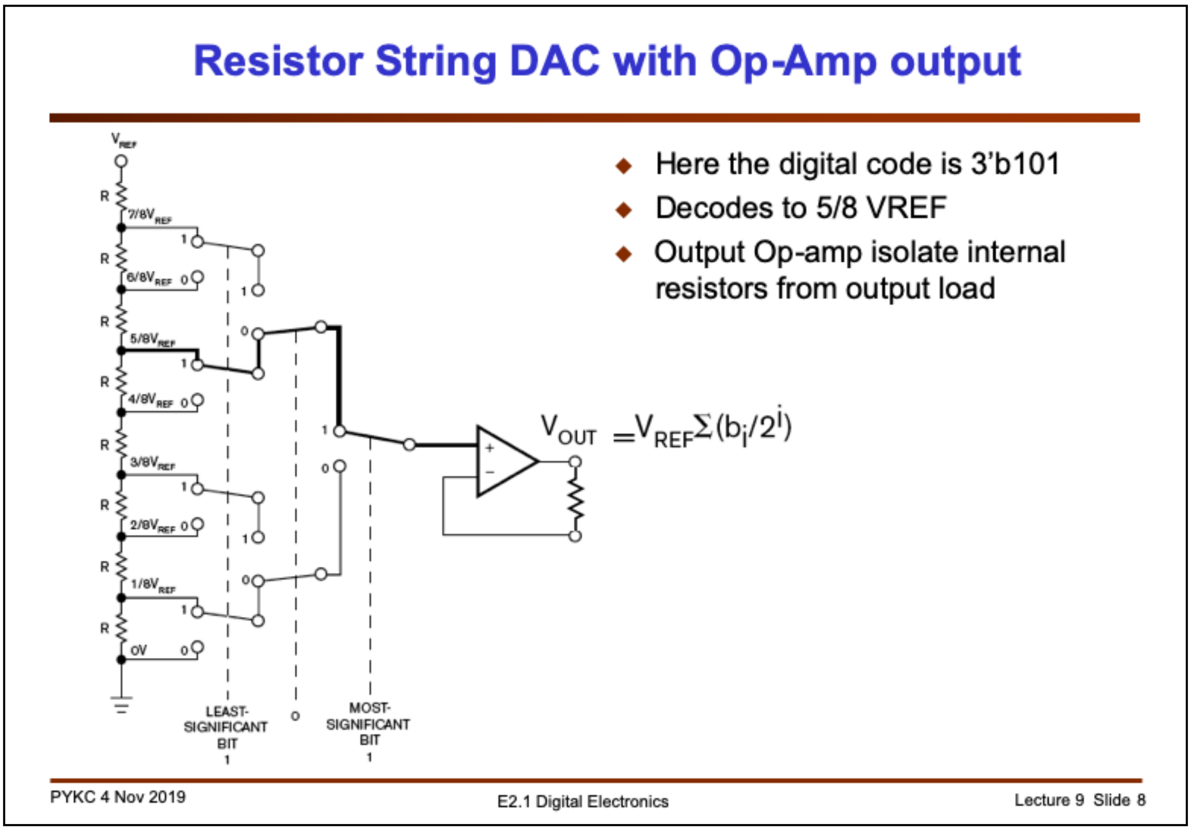
Thermometer DAC using Resistor String

- ✓ Simple
- ✓ Inherently monotonic
- ✓ Needs only IDENTICAL resistors, good differential linearity
- ✓ Only two switches operate during a transition, low output glitch and fast settling
- ✓ Low power
- ✓ Widely use with modern technology with small feature sizes
- X Large number of resistors
- X Useful for low to medium resolution DAC
- X Large resistance – resulting in higher noise

Instead of using binary weighted resistor network, we could use a series string of identical resistors as shown here. With this architecture, Vref to 0 is divided into 8 equal steps (including 0 value). The 3-bit digital input is decoded into 8 possible binary one-hot codes. For example, 000 results in the lowest switch being connected and 111 will switch the upper most switch on.

This DAC has the advantages listed here:

- It is simple, uses only one resistor value R everywhere, therefore easy to manufacture using semiconductor process.
- Only operating two switches at anyone time, so the glitches are smaller.
- It is low power and inherently monotonic.



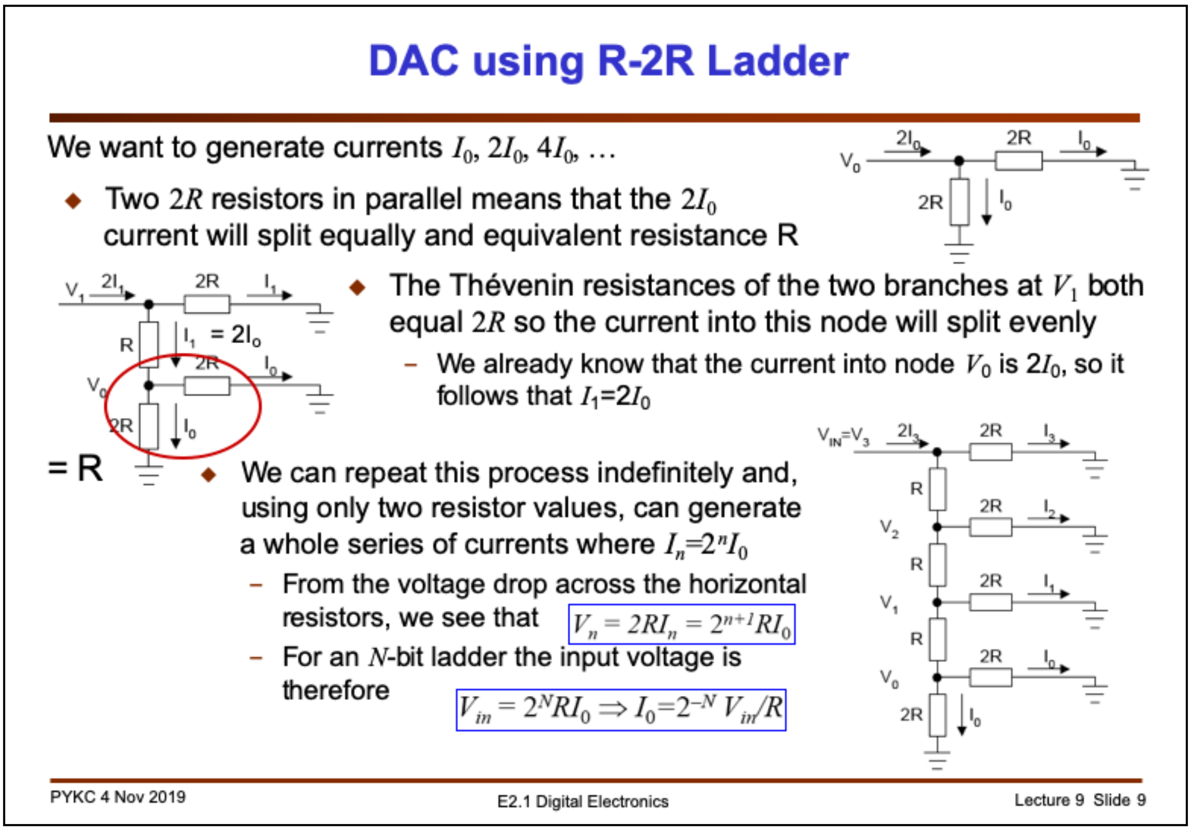
Resistor String DAC with Op-Amp output

- Here the digital code is 3'b101
- Decodes to 5/8 VREF
- Output Op-amp isolate internal resistors from output load

$V_{OUT} = V_{REF}\Sigma(b_i/2^i)$

Instead of using a large number of switches, we can also use switches arranged in a tree structure as shown here. Here is an example showing the decoding of the digital value 3'b0101. Decoding is implicitly performed via the control of the switches using the three digital bits. The output op amp provides buffering of the DAC voltage.

In this example, the 3'b101 digital value selects the 5/8 Vref tap of the resistor string to route to the op amp.



DAC using R-2R Ladder

We want to generate currents I_0, $2I_0$, $4I_0$, …

- Two $2R$ resistors in parallel means that the $2I_0$ current will split equally and equivalent resistance R
- The Thévenin resistances of the two branches at V_1 both equal $2R$ so the current into this node will split evenly
  - We already know that the current into node V_0 is $2I_0$, so it follows that I_1=$2I_0$

= R

- We can repeat this process indefinitely and, using only two resistor values, can generate a whole series of currents where I_n=2^nI_0
  - From the voltage drop across the horizontal resistors, we see that $\boxed{V_n = 2RI_n = 2^{n+1}RI_0}$
  - For an N-bit ladder the input voltage is therefore $\boxed{V_{in} = 2^N RI_0 \Rightarrow I_0 = 2^{-N} V_{in}/R}$

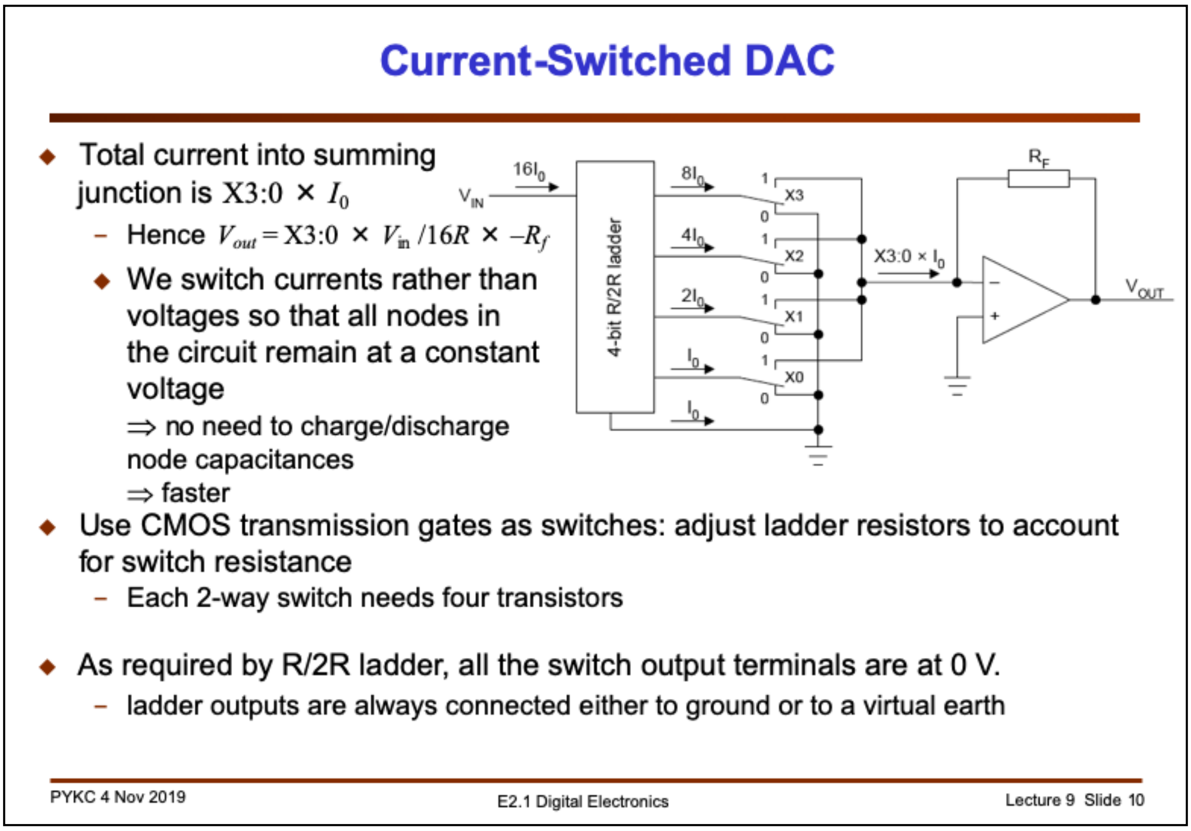
PYKC 4 Nov 2019 E2.1 Digital Electronics Lecture 9 Slide 9

String resistor network is good for, say, up to 10-bit DAC (requiring 1024 identical resistors). If you want a 16-bit DAC, you would need 65536 resistors! That is obviously not practical or too expensive. A better solution is to use R-2R Ladder network.

This circuit is very clever. The basic idea is to produce current `Io, 2Io, 4Io` etc, using only identical resistors connected in a special way.

The best way to understand the working of this R-2R network is to consider just two resistors both with values 2R. If the current flowing through each resistor is `Io`, then the total current at node Vo must be `I1 = 2Io`. The Thevenin equivalent resistance of these two resistor is 2R || 2R = R. Now we add an extra resistor R in series with these two 2R network. Together they form a resistance 2R. If we add the next step of the ladder as shown here, the total current at V1 is `2I1 = 4 Io`.

As you can see, adding each extra step of the ladder doubles the current. If the voltage drop across the horizontal resistors therefore also increases in ratios of 2 for each step.



Current-Switched DAC

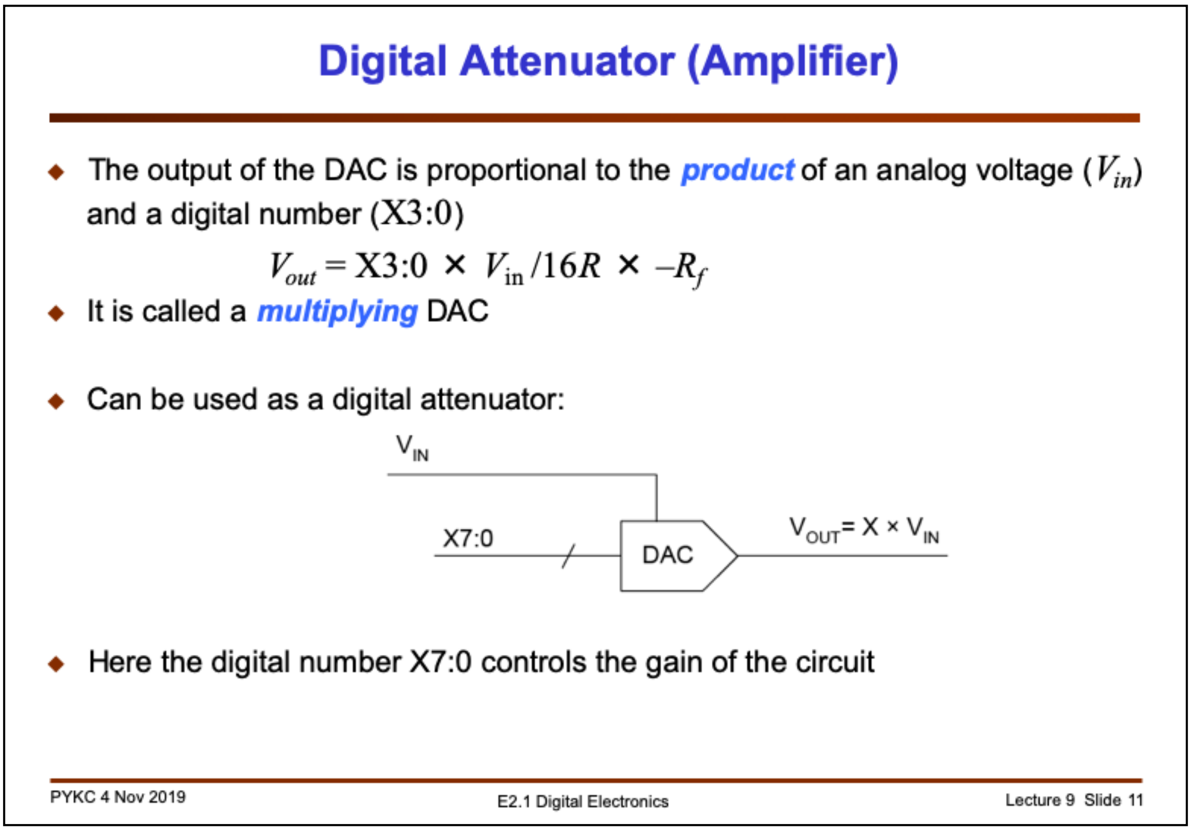
- Total current into summing junction is X3:0 × I_0
  - Hence $V_{out} = \mathrm{X3{:}0} \times V_{in}/16R \times -R_f$
  - We switch currents rather than voltages so that all nodes in the circuit remain at a constant voltage
    ⇒ no need to charge/discharge node capacitances
    ⇒ faster

16I$_0$
V$_{IN}$
4-bit R/2R ladder
8I$_0$
4I$_0$
2I$_0$
I$_0$
I$_0$
1
0
X3
1
0
X2
1
0
X1
1
0
X0
X3:0 × I$_0$
R$_F$
–
+
V$_{OUT}$

- Use CMOS transmission gates as switches: adjust ladder resistors to account for switch resistance
  - Each 2-way switch needs four transistors
- As required by R/2R ladder, all the switch output terminals are at 0 V.
  - ladder outputs are always connected either to ground or to a virtual earth

For a practical DAC circuit, the R/2R ladder network is connected to the virtual earth of the op amp as shown here. The current is either sent to the virtual earth node if the digital value is ‘1’, or switched to earth if it is ‘0’. In that way , the output voltage Vout is a converter analogue value of X[3:0].

Note that we switch current from one branch to another branch. It is known as **current steering**. Current steering is much faster than turning the current on and off.



Digital Attenuator (Amplifier)

- The output of the DAC is proportional to the ***product*** of an analog voltage (V_{in}) and a digital number ($X3{:}0$)

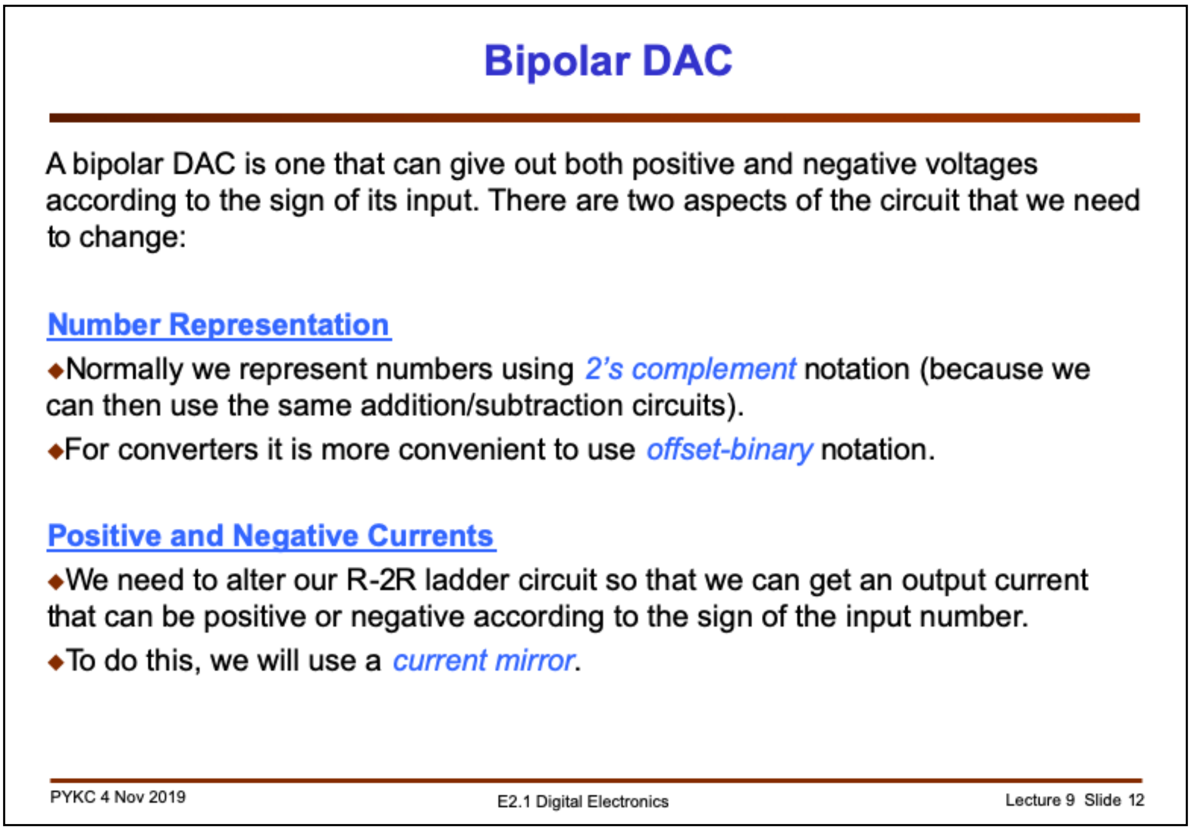
$$V_{out} = X3{:}0 \times V_{\mathrm{in}}/16R \times -R_f$$

- It is called a ***multiplying*** DAC
- Can be used as a digital attenuator:

V_{IN}
X7:0
DAC
$V_{OUT} = X \times V_{IN}$

- Here the digital number X7:0 controls the gain of the circuit

Instead of using Vref, a fixed reference voltage, we could use an analogue input Vin (such as an audio signal), and then use the DAC as a digitally control amplifier or attenuator. This is also known as a multiplying DAC. The output is X multiplied by Vin.



Bipolar DAC

A bipolar DAC is one that can give out both positive and negative voltages according to the sign of its input. There are two aspects of the circuit that we need to change:

Number Representation

- Normally we represent numbers using *2's complement* notation (because we can then use the same addition/subtraction circuits).
- For converters it is more convenient to use *offset-binary* notation.

Positive and Negative Currents

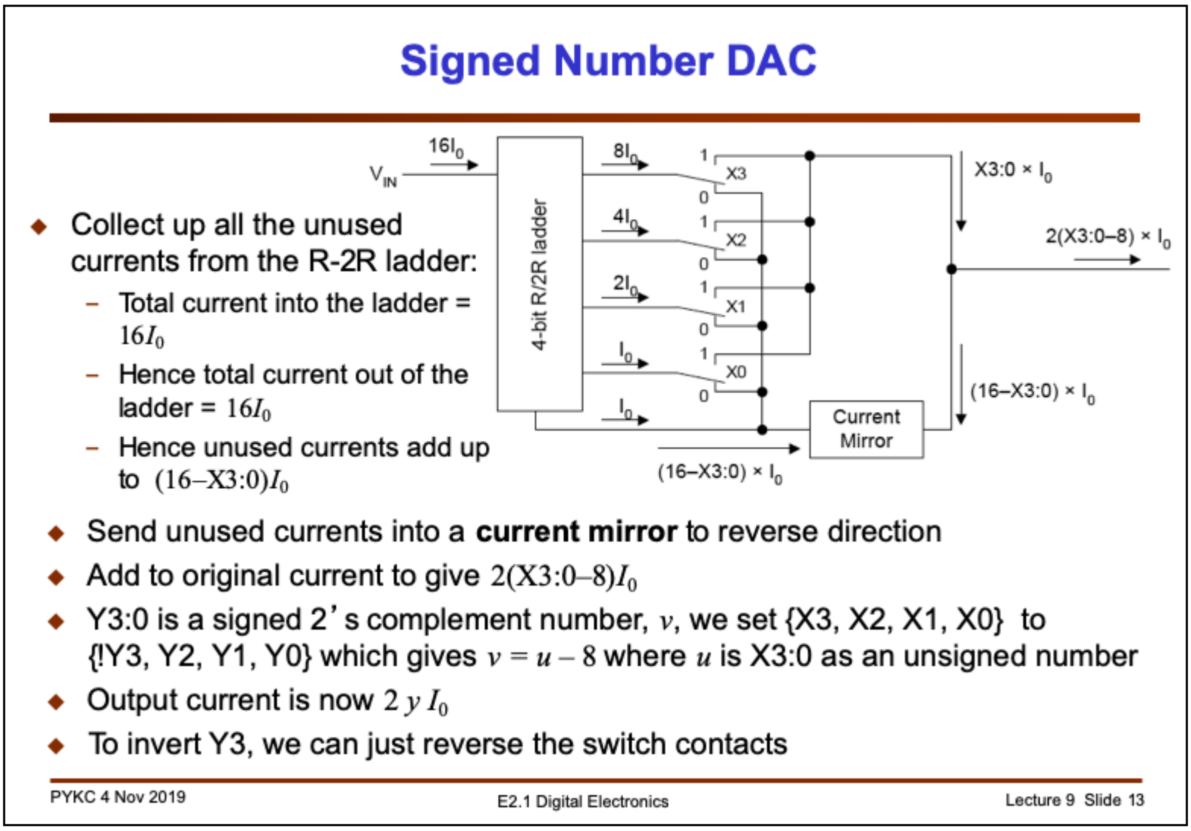
- We need to alter our R-2R ladder circuit so that we can get an output current that can be positive or negative according to the sign of the input number.
- To do this, we will use a *current mirror*.

You are familiar with 2's complement notation. You may not know about offset-binary numbers.

What it means is that you use zero to represent the most negative value instead of a negative number. For example, for a range from -512 to 511, use the range 0 to 1023 by adding to your number an offset of 512:

Yoffset = X + 512

If you need to produce a DAC with negative voltage or current for bipolar digital input values, you need an analogue component known as a current mirror. You don't need to know exactly how this could be implemented. It is sufficient to understand that a current mirror simply mirrors the current on one branch of the circuit to a second branch of the circuit as shown in the next slide.



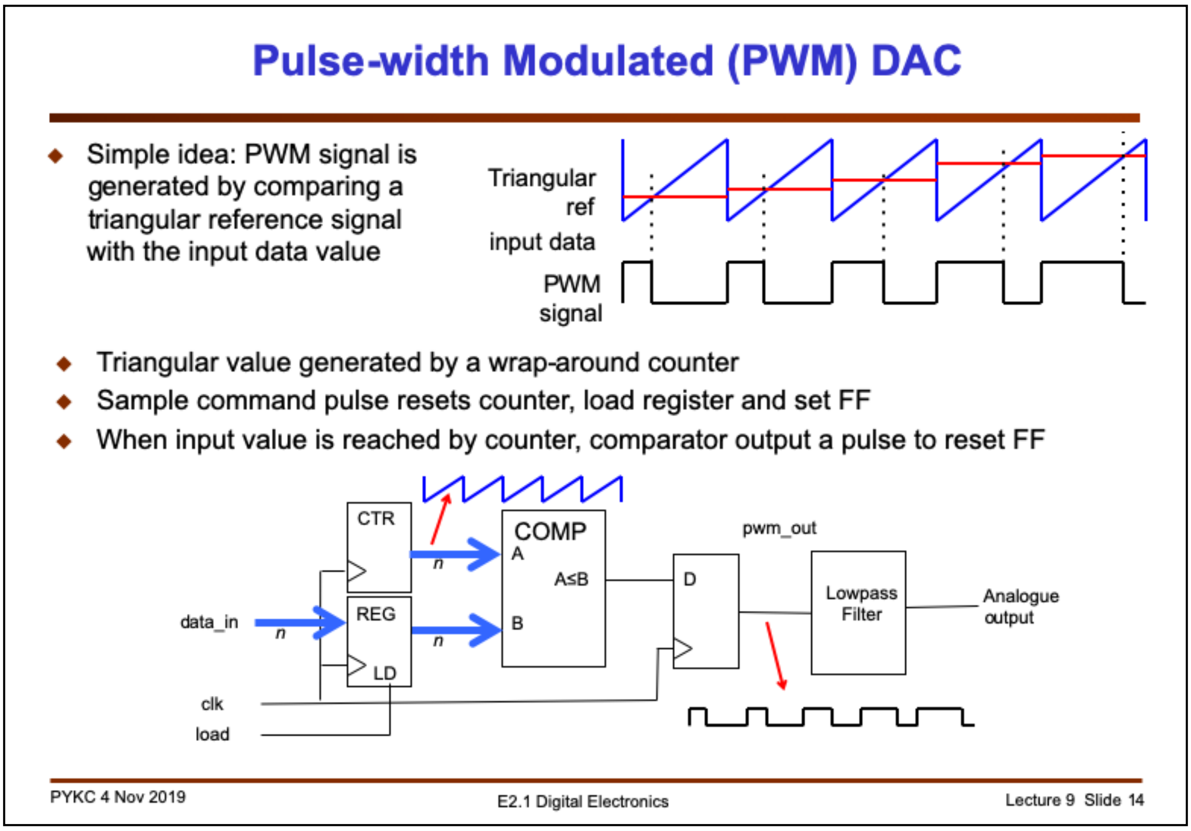
Signed Number DAC

- Collect up all the unused currents from the R-2R ladder:
  - Total current into the ladder = $16I_0$
  - Hence total current out of the ladder = $16I_0$
  - Hence unused currents add up to $(16–X3:0)I_0$
- Send unused currents into a **current mirror** to reverse direction
- Add to original current to give $2(X3:0–8)I_0$
- Y3:0 is a signed 2's complement number, v, we set {X3, X2, X1, X0} to {!Y3, Y2, Y1, Y0} which gives $v = u – 8$ where u is X3:0 as an unsigned number
- Output current is now $2\,y\,I_0$
- To invert Y3, we can just reverse the switch contacts

PYKC 4 Nov 2019 E2.1 Digital Electronics Lecture 9 Slide 13

Assuming that you have the current mirror component available, you can connect this as shown here.

Y[3:0] is a 2's complement number that we want to convert. X3 is made to be ~Y3. The output current is now bipolar. If X[3:0] = 4'b0000, then the output current is -16 Io (i.e Y[3:0] = -8). If X[3:0] = 4'b1111, then output current is 14 Io (i.e. Y[3:0] = 7).



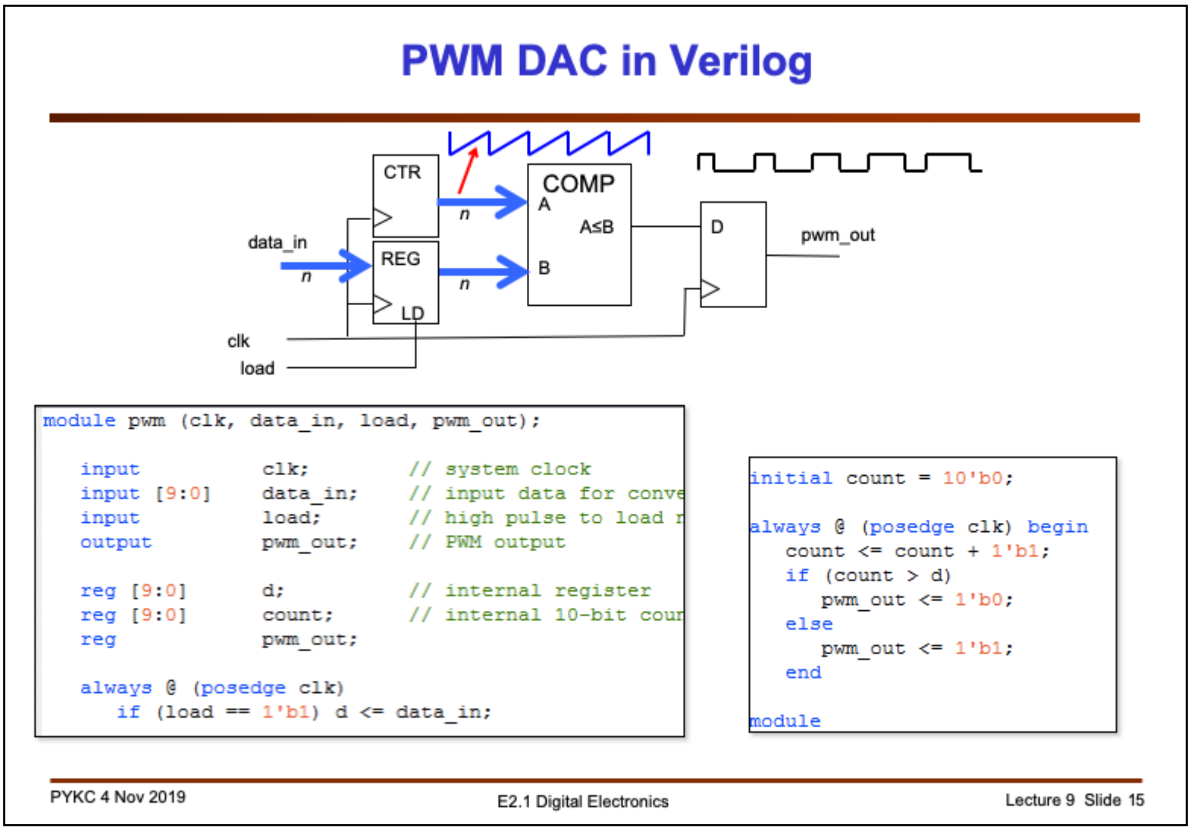
Pulse-width Modulated (PWM) DAC

- Simple idea: PWM signal is generated by comparing a triangular reference signal with the input data value

- Triangular value generated by a wrap-around counter
- Sample command pulse resets counter, load register and set FF
- When input value is reached by counter, comparator output a pulse to reset FF

Instead of using analogue resistor network, it is possible to build a simple DAC using only digital components.

Here is a circuit schematic for a pulse-width modulated DAC. Here the counter is used to produce a count value A that ramps up linearly in a sawtooth manner. The digital value we want to convert to analogue value is data_in, which is stored as B in the input register. A digital comparator circuit compares this input data with the counter value (which is ramping up). While A is less than B, the output of the comparator is high. As soon as A exceeds B, the output goes low. In this way, the pulse width is proportional to the value of B (or data_in) in a linear manner. Passing this PWM signal through a lowpass filter will give an analogue output which is linearly related to data_in.



PWM DAC in Verilog

CTR
COMP
A
A≤B
B
D
pwm_out
data_in
n
REG
LD
n
n
clk
load

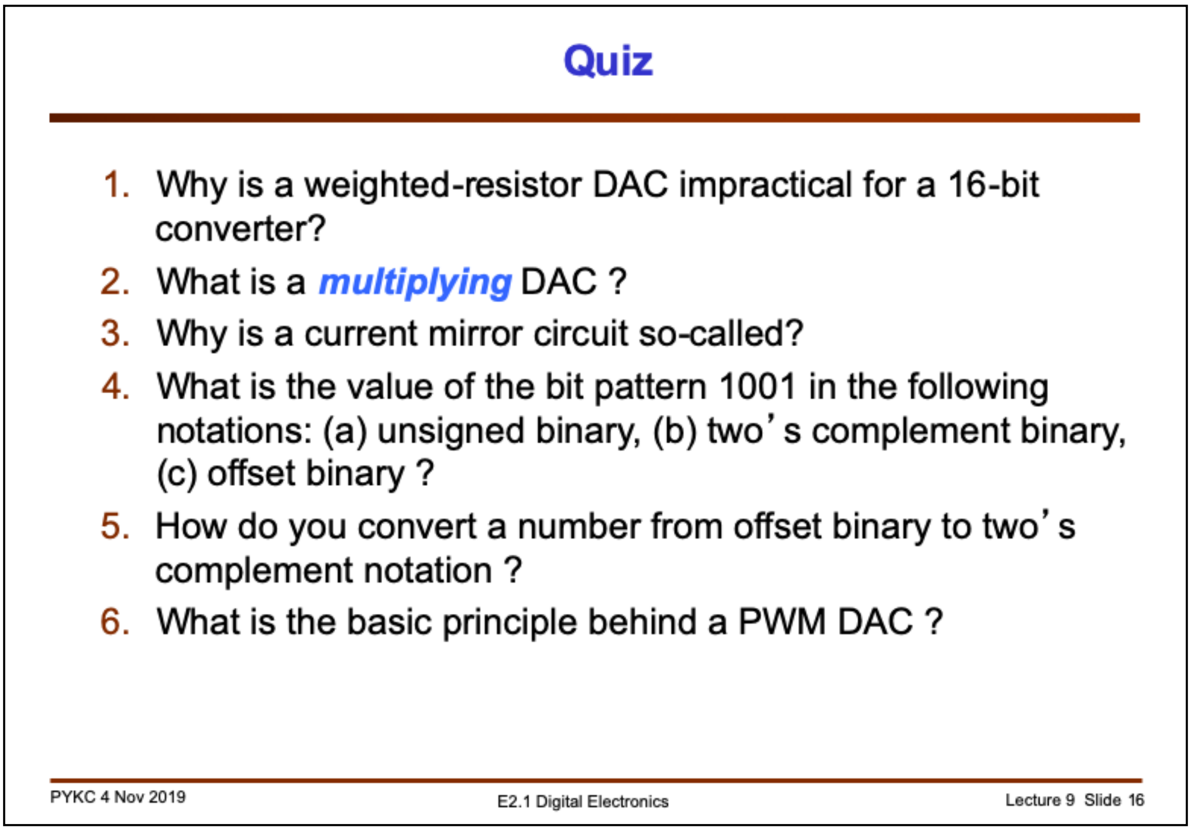
```verilog
module pwm (clk, data_in, load, pwm_out);

   input          clk;        // system clock
   input [9:0]    data_in;    // input data for conve
   input          load;       // high pulse to load r
   output         pwm_out;    // PWM output

   reg [9:0]      d;          // internal register
   reg [9:0]      count;      // internal 10-bit cour
   reg            pwm_out;

   always @ (posedge clk)
      if (load == 1'b1) d <= data_in;
```

```verilog
initial count = 10'b0;

always @ (posedge clk) begin
   count <= count + 1'b1;
   if (count > d)
      pwm_out <= 1'b0;
   else
      pwm_out <= 1'b1;
   end

module
```

PYKC 4 Nov 2019 E2.1 Digital Electronics Lecture 9 Slide 15

Implementing a PWM DAC is extremely simple in Verilog. Although this is not specified as an exercise in the experiment, I suggest that you should try this out for yourself.

Here is the Verilog code.



Quiz

1. Why is a weighted-resistor DAC impractical for a 16-bit converter?
2. What is a ***multiplying*** DAC ?
3. Why is a current mirror circuit so-called?
4. What is the value of the bit pattern 1001 in the following notations: (a) unsigned binary, (b) two’s complement binary, (c) offset binary ?
5. How do you convert a number from offset binary to two’s complement notation ?
6. What is the basic principle behind a PWM DAC ?