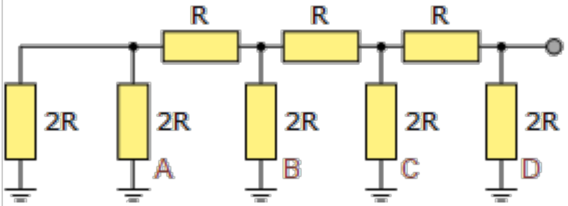
ElectronicsTutorials



R-2R DAC

R-2R Digital-to-Analogue Converter, or DAC, is a data converter which use two precision resistor to convert a digital binary number into an analogue output signal proportional to the value of the digital number

Compared to the R-2R DAC, the binary weighted digital-to-analogue converter has an analogue output voltage which is the weighted sum of the individual inputs. Thus it requires a large range of precision resistors within its ladder network, making its design both expensive and impractical for most DAC’s requiring lower levels of resolution.

As the binary weighted DAC is based on a closed-loop inverting operational amplifier using summing amplifier topology, this type of data converter configuration may work well for a D/A converter of a few bits of resolution. But a much simpler approach is using a R-2R resistive ladder network to construct a **R-2R Digital-to-Analogue Converter** requires only two precision resistances.

The R-2R resistive ladder network uses just two resistive values. One resistor has the base value “R”, and the second resistor has twice the value of the first resistor, “2R”, no matter how many bits are used to make up the ladder network.

So for example, we could just use a standard 1kΩ resistor for the base resistor “R”, and therefore a 2kΩ resistor for “2R” (or multiples thereof as the base value of R is not too critical). Thus the resistive value of 2R will always be twice the value of the base resistor, R. That is 2R = 2*R. This means that it is much easier for us to maintain the required accuracy of the resistors along the ladder network compared to the previous weighted resistor DAC. But what is a “R-2R resistive ladder network” anyway.

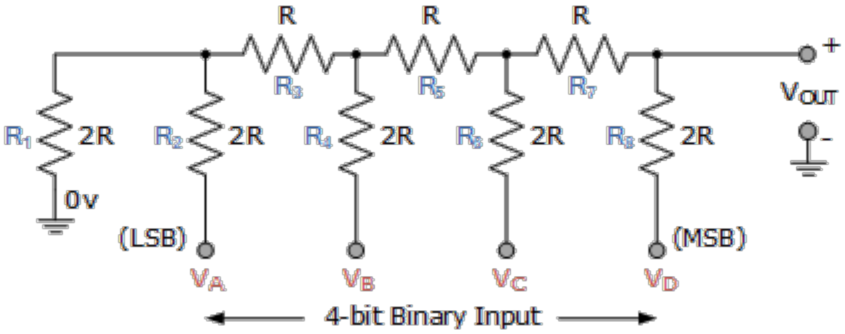
R-2R Resistive Ladder Network

As its name implies, the “ladder” description comes from the ladder-like configuration of the resistors used within the network. A R-2R resistive ladder network provides a simple means of converting digital voltage signals into an equivalent analogue output.

Input voltages are applied to the ladder network at various points along its length and the more input points the better the resolution of the R-2R ladder. The output signal as a result of all these input voltage points is taken from the end of the ladder which is used to drive the inverting input of an operational amplifier.

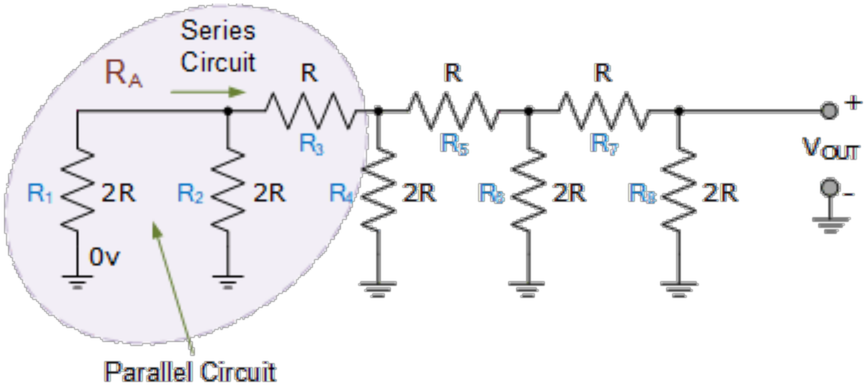
Then a R-2R resistive ladder network is nothing more than long strings of parallel and series connected resistors acting as interconnected voltage dividers along its length, and whose output voltage depends soley on the interaction of the input voltages with each other. Consider the basic 4-bit R-2R ladder network (4-bits because it has four input points) below.

4-bit R-2R Resistive Ladder Network



This 4-bit resistive ladder circuit may look complicated, but its all about connecting resistors together in parallel and series combinations and working back to the input source using simple circuit laws to find the proportional value of the output. Lets assume all the binary inputs are grounded at 0 volts, that is: $V_A = V_B = V_C = V_D = 0V$ (LOW). The binary code corresponding to these four inputs will therefore be: **0000**.

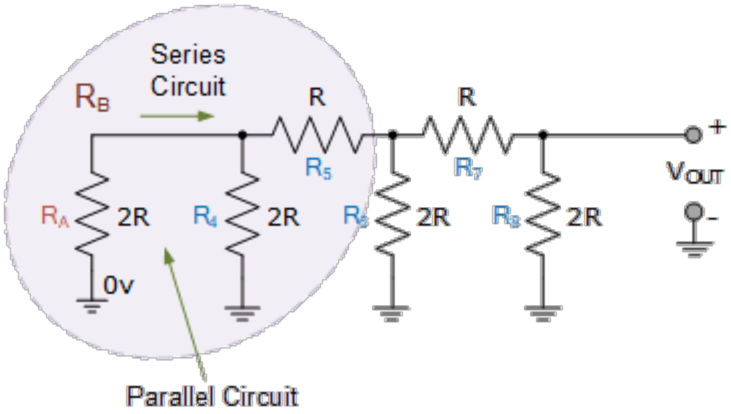
Starting from the left hand side and using the simplified equation for two parallel resistors and series resistors, we can find the equivalent resistance of the ladder network as:



Resistors R_1 and R_2 are in "parallel" with each other but in "series" with resistor R_3. Then we can find the equivalent resistance of these three resistors and call it R_A for simplicity (or any other form of identification you want).

$$R_A = R_3 + \frac{R_1 \times R_2}{R_1 + R_2} = R + \frac{2R \times 2R}{2R + 2R} = R + R = 2R$$

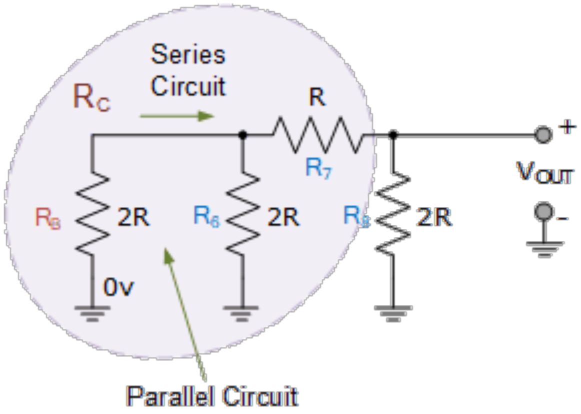
Then R_A is equivalent to "2R". Now we can see that the equivalent resistance "R_A" is in parallel with R_4 with the parallel combination in series with R_5.



Again we can find the equivalent resistance of this combination and call it R_B.

$$R_B = R_5 + \frac{R_A \times R_4}{R_A + R_4} = R + \frac{2R \times 2R}{2R + 2R} = R + R = 2R$$

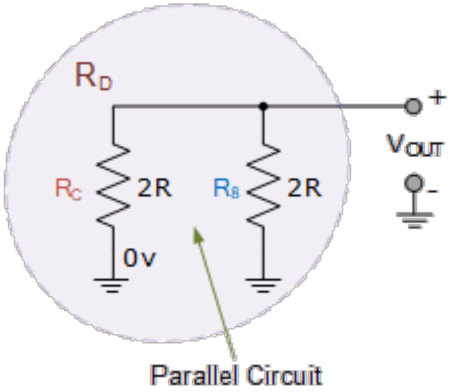
So R_B combination is equivalent to “2R”. Hopefully we can see that this equivalent resistance R_B is in parallel with R_6 with the parallel combination in series with R_7 as shown.



As before we find the equivalent resistance and call it R_C.

$$R_C = R_7 + \frac{R_B \times R_6}{R_B + R_6} = R + \frac{2R \times 2R}{2R + 2R} = R + R = 2R$$

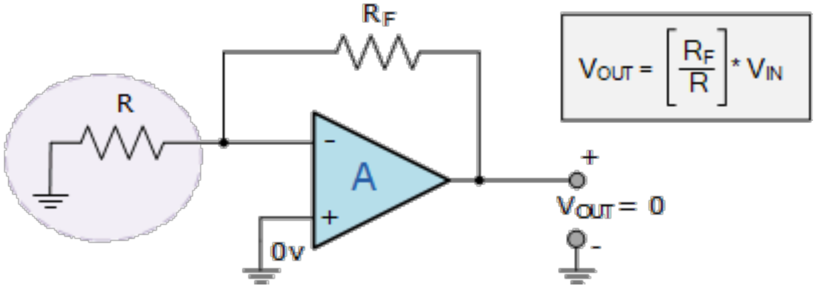
Again, resistor combination R_C is equivalent to “2R” which is in parallel with R_8 as shown.



As we have shown above, when two equal resistor values are paralled together, the resulting value is one-half, so 2R in parallel with 2R equals an equivalent resistance of R. So the whole 4-bit R-2R resistive ladder network comprising of individual resistors connected together in parallel and series combinations has an equivalent resistance (R_{EQ}) of “R” when a binary code of “0000” is applied to its four inputs.

Therefore with a binary code of “0000” applied as inputs, our basic 4-bit R-2R digital-to-analogue converter circuit would look something like this:

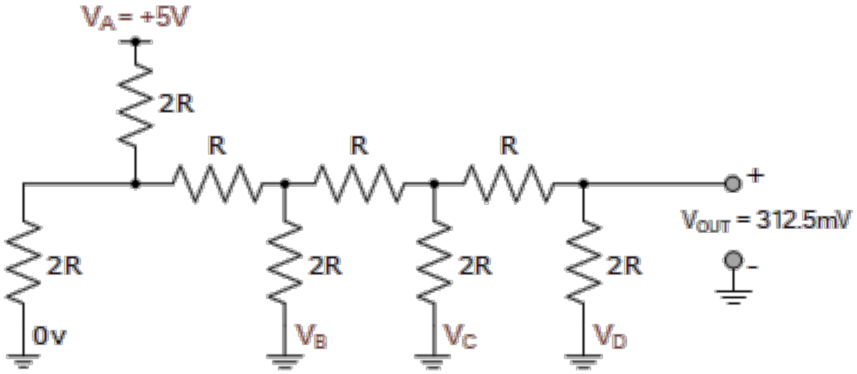
R-2R DAC Circuit with Four Zero (LOW) Inputs



The output voltage for an inverting operational amplifier is given as: $(R_F/R_{IN})*V_{IN}$. If we make R_F equal to R, that is $R_F = R = 1$, and as R is terminated to ground (0V), then there is no V_{IN} voltage value, ($V_{IN} = 0$) so the output voltage would be: (1/1)*0 = 0 volts. So for a 4-bit R-2R DAC with four grounded inputs (LOW), the output voltage will be “zero” volts, thus a 4-bit digital input of **0000** produces an analogue output of 0 volts.

So what happens now if we connect input bit V_A HIGH to +5 volts. What would be the equivalent resistive value of the R-2R ladder network and the output voltage from the op-amp.

R-2R DAC with Input V_A

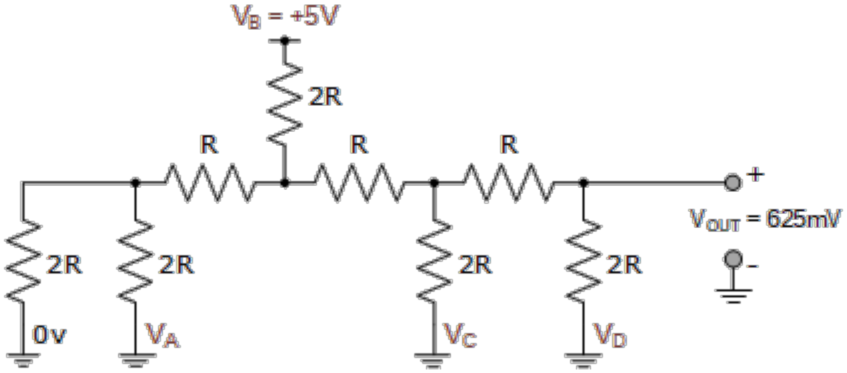


Input V_A is HIGH and logic level “1” and all the other inputs grounded at logic level “0”. As the R/2R ladder network is a linear circuit we can find Thevenin’s equivalent resistance using the same parallel and series resistance calculations as above to calculate the expected output voltage. The output voltage, V_{OUT} is therefore calculated at 312.5 milli-volts (312.5 mV).

As we have a 4-bit R-2R resistive ladder network, this 312.5 mV voltage change is one-sixteenth the value of the +5V input (5/0.3125 = 16) voltage so is classed as the Least Significant Bit, (LSB). Being the least significant bit, input V_A will therefore determine the "resolution" of our simple 4-bit digital-to-analogue converter, as the smallest voltage change in the analogue output corresponds to a single step change of the digital inputs. Thus for our 4-bit DAC this will be 312.5mV (1/16th) for a +5V input.

Now lets see what happens to the output voltage if we connect input bit V_B HIGH to +5 volts.

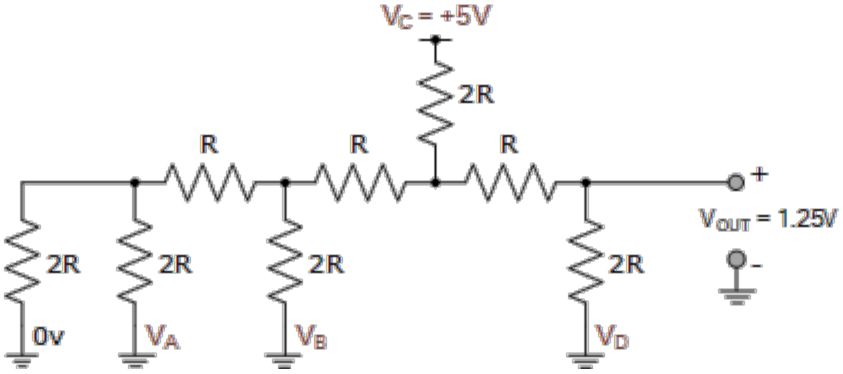
R-2R DAC with Input V_B



With input V_B HIGH and logic level "1" and all the other inputs grounded at logic level "0", the output voltage, V_{OUT} is calculated at 625mV, and which is one-eighth (1/8th) the value of the +5V input (5/0.625 = 8) voltage. We can also see that it is double the output voltage when only input bit V_A was applied, and we would expect this as its the 2_{nd} bit (input) so has double the weighting of the 1_{st} bit.

Now lets see what happens to the output voltage if we connect input bit V_C HIGH to +5 volts.

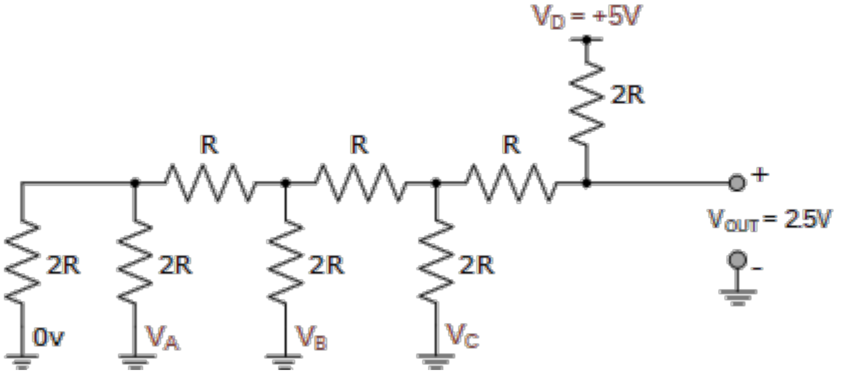
R-2R DAC with Input V_C



With input V_C HIGH and logic level “1” and the other input bits at logic level “0”, the output voltage, V_{OUT} is calculated at 1.25 volts, and which is one-quarter (1/4) the value of the +5V input (5/1.25 = 4) voltage. Again we can see that this voltage is double the output of input bit V_B but also 4 times the value of bit V_A. This is because input V_C is the 3_{rd} bit so has double the weighting of the 2_{nd} bit and four times the weighting of the 1_{st} bit.

Finally lets see what happens to the output voltage if we connect input V_D HIGH to +5 volts.

R-2R DAC with Input V_D



With only input V_D HIGH and logic level “1” and the other inputs at logic level “0”, the output voltage, V_{OUT} is calculated at 2.5 volts. This is on-half (1/2) the value of the +5V input (5/2.5 = 2) voltage. Again we can see that this voltage is double the output of input bit V_C, 4 times the value of bit V_B and 8 times the value of input bit V_A as it is the 4_{th} bit and therefore classed as the Most Significant Bit, (MSB).

Then we can see that if input V_A represents the LSB and therefore controls the DAC’s resolution, and input V_B is double V_A, input V_C is four times greater than V_A, and input V_D is eight times greater than V_A, we can obtain a relationship for the analogue output voltage of our 4-bit digital-to-analogue converter with the following equation:

Digital-to-Analogue Output Voltage Equation

$$V_{OUT} = \frac{V_A + 2V_B + 4V_C + 8V_D}{16}$$

Where the denominator value of 16 corresponds to the 16 (2^4) possible combinations of inputs to the 4-bit R-2R ladder network of the DAC.

We can expand this equation further to obtain a generalised R-2R DAC equation for any number of digital inputs for a R-2R D/A converter as the weighting of each input bit will always be referenced to the least significant bit (LSB), giving us a generalised equation of:

Generalised R-2R DAC Equation

$$V_{OUT} = \frac{V_A + 2V_B + 4V_C + 8V_D + 16V_E + 32V_F + \ldots \text{etc}}{2^n}$$

Where: “n” represents the number of digital inputs within the R-2R resistive ladder network of the DAC producing a resolution of: $V_{LSB} = V_{IN}/2^n$.

Clearly then input bit V_A when HIGH will cause the smallest change in the output voltage, while input bit V_D when HIGH will cause the greatest change in the output voltage. The expected output voltage is therefore calculated by summing the effect of all the individual input bits which are connected HIGH.

Ideally, the ladder network should produce a linear relationship between the input voltages and the analogue output as each input will have a step increase equal to the LSB, we can create a table of expected output voltage values for all 16 combinations of the 4 inputs with +5V representing a logic “1” condition as shown.

4-bit R-2R D/A Converter Voltage Output

Digital Inputs				V_{OUT} Expression	V_{OUT}
D	C	B	A	$(8*V_D + 4*V_C + 2*V_B + 1*V_A)/2^4$	in Volts

0	0	0	0	(0*5 + 0*5 + 0*5 + 0*5)/16	0
0	0	0	1	(0*5 + 0*5 + 0*5 + 1*5)/16	0.3125
0	0	1	0	(0*5 + 0*5 + 2*5 + 0*5)/16	0.6250
0	0	1	1	(0*5 + 0*5 + 2*5 + 1*5)/16	0.9375
0	1	0	0	(0*5 + 4*5 + 0*5 + 0*5)/16	1.2500
0	1	0	1	(0*5 + 4*5 + 0*5 + 1*5)/16	1.5625
0	1	1	0	(0*5 + 4*5 + 2*5 + 0*5)/16	1.8750
0	1	1	1	(0*5 + 4*5 + 2*5 + 1*5)/16	2.1875
1	0	0	0	(8*5 + 0*5 + 0*5 + 0*5)/16	2.5000
1	0	0	1	(8*5 + 0*5 + 0*5 + 1*5)/16	2.8125
1	0	1	0	(8*5 + 0*5 + 2*5 + 0*5)/16	3.1250
1	0	1	1	(8*5 + 0*5 + 2*5 + 1*5)/16	3.4375
1	1	0	0	(8*5 + 4*5 + 0*5 + 0*5)/16	3.7500
1	1	0	1	(8*5 + 4*5 + 0*5 + 1*5)/16	4.0625
1	1	1	0	(8*5 + 4*5 + 2*5 + 0*5)/16	4.3750
1	1	1	1	(8*5 + 4*5 + 2*5 + 1*5)/16	4.6875

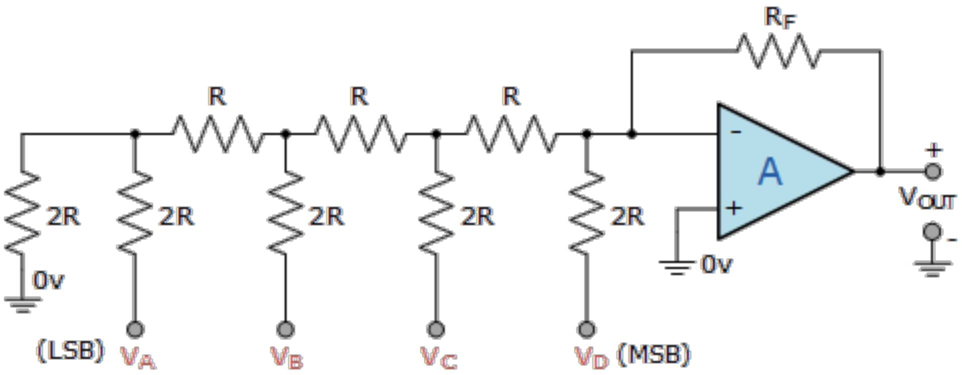
You may have noticed that the full-scale analogue output voltage for a binary code of **1111** never reaches the same value as the digital input voltage (+5V) as it is always less by the equivalent of one LSB bit, (312.5mV in this example).

However, the higher the number of digital input bits (resolution) the nearer the analogue output voltage reaches full-scale when all the input bits are HIGH. Likewise when all the input bits are LOW, the resulting lower resolution of LSB makes V_{OUT} closer to zero volts.

R-2R Digital-to-Analogue Converter

Now that we understand what a *R-2R resistive ladder network* is and how it works, we can use it to produce a R-2R Digital-to-Analogue Converter. Again using our 4-bit R-2R resistive ladder network from above and adding it to an inverting operational amplifier circuit, we can create a simple R-2R digital-to-analogue converter of:

R-2R Digital-to-Analogue Converter



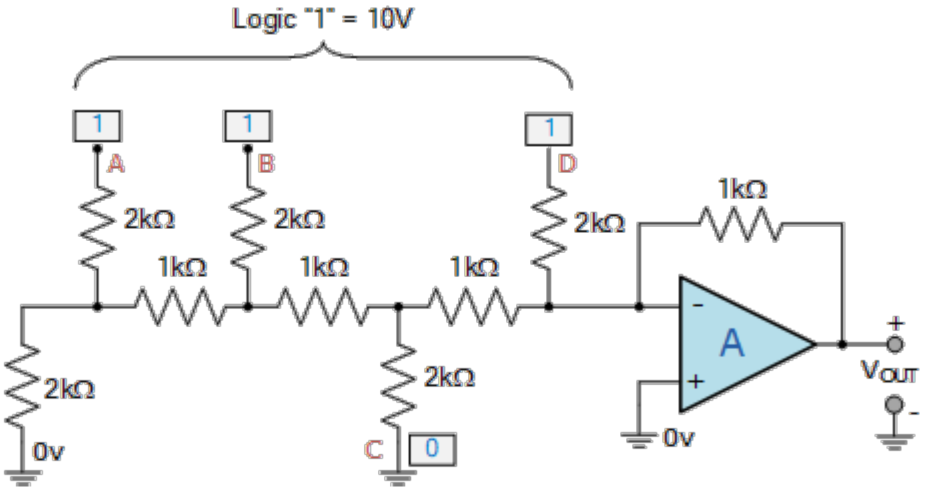
The digital logic circuit used to drive the D/A converter can be generated by combinational or sequential logic circuits, data registers, counters or simply switches. The interfacing of a R-2R D/A converter of "n"-bits will depend upon its application. All-in-one boards such as the Arduino or Raspberry Pi have *digital-to-analogue converters* built-in so make interfacing and programming much easier. There are many popular DAC's available such as the 8-bit DAC0808.

R-2R D/A Converter Example No1

A 4-bit R-2R digital-to-analogue converter is constructed to control the speed of a small DC motor using the output from a digital logic circuit. If the logic circuit uses 10 volt CMOS devices, calculate the analogue output voltage from the DAC when the input code is hexadecimal number "B". Also determine the resolution of the DAC.

1). The hexadecimal letter "B" is equal to the number eleven in decimal. The decimal number eleven is equal to the binary code "1011" in binary. That is: $B_{16} = 1011_2$. Thus for our 4-bit binary number of 1011_2, input bit D = 1, bit C = 0, bit B = 1 and bit A = 1.

If we assume that feedback resistor R_F is equal to "R", then our R-2R D/A converter circuit will look like:



The digital logic circuit uses 10 volt CMOS devices, so the input voltage to the R-2R network will be 10 volts. Also being a 4-bit ladder DAC, there will be 2^4 possible input combinations, so using our equation from above, the ouput voltage for a binary code of 1011_2 is calculated as:

$$V_{OUT} = \frac{1V_A + 2V_B + 4V_C + 8V_D}{2^n}$$

$$V_{OUT} = \frac{1\times10 + 2\times10 + 4\times0 + 8\times10}{16}$$

$$V_{OUT} = \frac{110}{16} = 6.875 \text{ Volts}$$

Therefore the analogue output voltage used to control the DC motor when the input code is 1011_2 is calculated as: -6.875 volts. Note that the output voltage is negative due to the inverting input of the operational amplifier.

2). The resolution of the converter will be equal to the value of the least significant bit (LSB) which is given as:

$$\text{Resolution} = V_{(LSB)} = \frac{V_{IN}}{2^n}$$

$$\therefore\ \text{Resolution} = \frac{10}{16} = 0.625\ \text{Volts}$$

Then the smallest step change of the analogue output voltage, V_{OUT} for a 1-bit LSB change of the digital input of this 4-bit R-2R digital-to-analogue converter example is: 0.625 volts. That is the output voltage changes in steps or increments of 0.625 volts and not as a straight linear value.

4-bit Binary Counting R-2R DAC

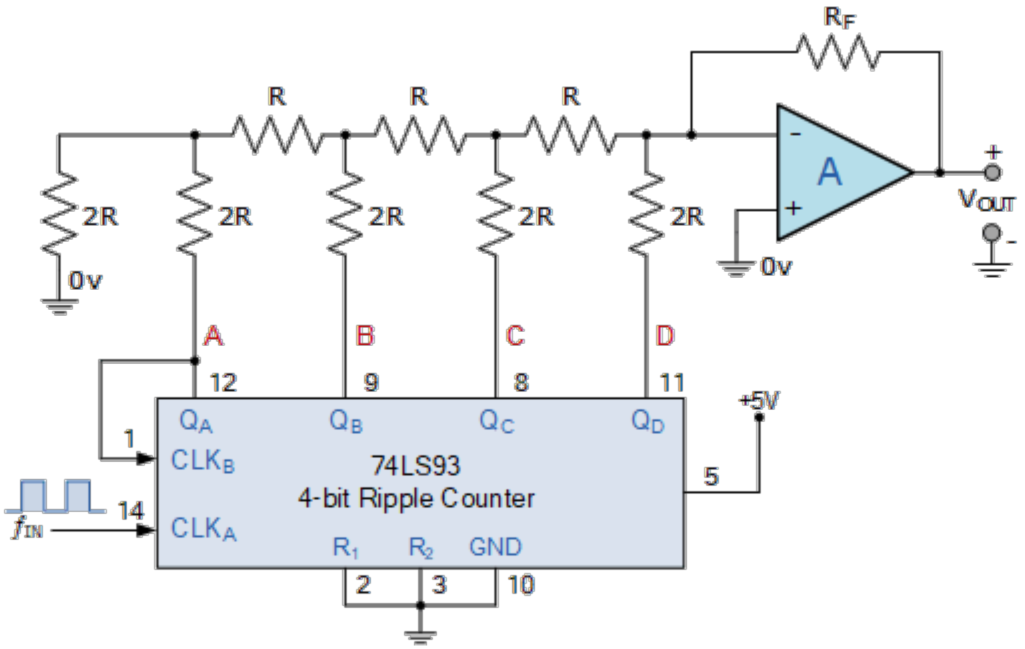
Hopefully by now we understand that we can make a R-2R ladder DAC using just two resistor values, one the base value "R" and the other twice or double the value being "2R". In our simple example above we have made a 4-bit R-2R DAC with four input data lines, A, B, C, and D giving us 16 (2^4) different input combinations from "0000" to "1111".

The binary code for these four digital input lines can be generated in many different ways, using micro-controllers, digital circuits, mechanical or solid state switches. But one interesting option is to use a 4-bit binary counter such as the 74LS93.

The **74LS93** is a 4-bit J-K ripple counter which can be configured to count-up from 0000_2 to 1111_2 (MOD-16) and reset back to zero (0000) again by the application of a single external clock signal. The 74LS93 is an asynchronous counter commonly called a "ripple" counter because of the way that the internal J-K bistables respond to the clock or timing input producing a 4-bit binary output.

The frequency (or period) of this external clock or timing pulse is divided by a factor of 2, 4, 8, and 16 by the counters output lines as the clock pulse appears to ripple through the four J-K flip-flops producing the required 4-bit output count sequence from 0000_2 to 1111_2.

4-bit Binary Counting R-2R DAC



Note that to count upwards from 0000 to 1111, the external CLK_B input must be connected to the Q_A (pin-12) output and the input timing pulses are applied to input CLK_A (pin-14).

This simple 4-bit asynchronous up counter built around the 74LS93 binary ripple counter as the same counting sequence given in the above table. On the application of a clock pulse the outputs: Q_A, Q_B, Q_C, and Q_D change by one step.

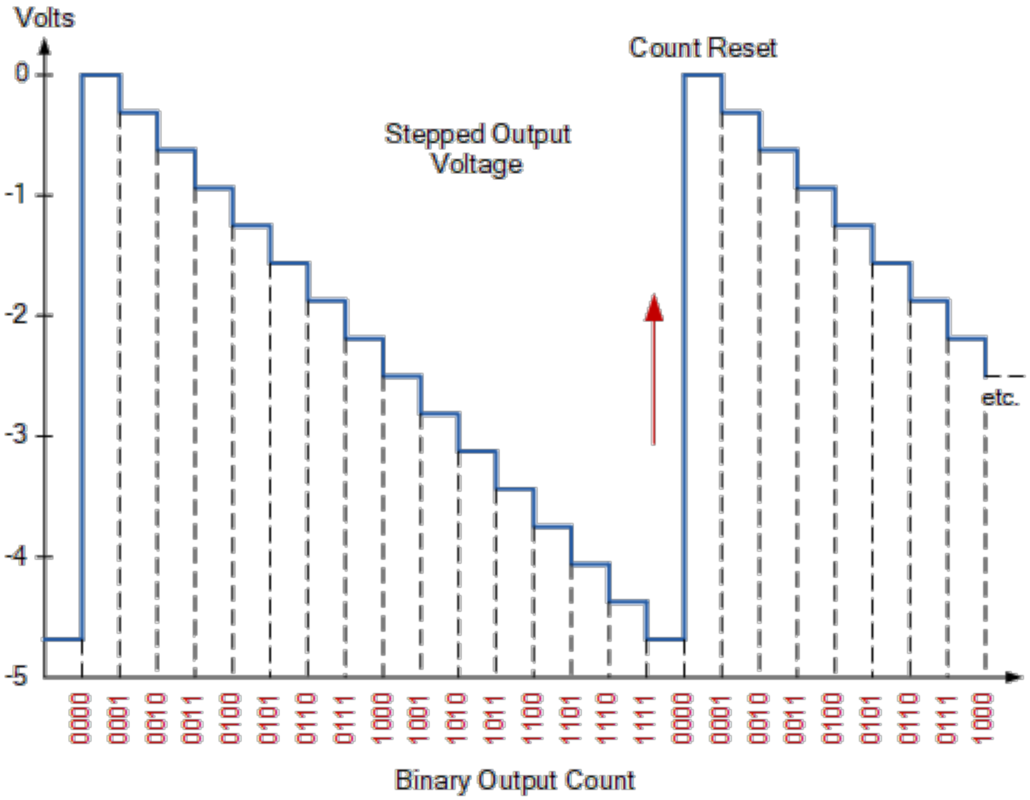
The input of the operational amplifier detects this step change and outputs a negative voltage (inverting op-amp) relative to the binary code at the R-2R ladder inputs. The output voltage value for each step will correspond to that given in the table above.

The ripple counter will count up in sequence with the four outputs producing an output sequence of binary values upto the 15th clock pulse where the outputs are set to 1111_2 (decimal 15) producing the maximum negative output voltage of the digital-to-analogue converter.

On the 16th pulse the counters output sequence is reset and the count returns back to 0000, which resets the op-amps output back to zero volts. The application of the next clock pulse begins a new counting cycle from zero to $V_{OUT(max)}$.

We can show the output sequence for this simple 4-bit binary asynchronous counting R-2R D/A converter in the following timing diagram.

4-bit R-2R DAC Timing Diagram



Clearly then, the output voltage of the operational amplifier varies from zero volts to its maximum negative voltage as the ripple counter counts from 0000_2 to 1111_2 respectively. This simple circuit could be used to vary the brightness of a lamp connected to the op-amps output, or continually vary the speed of a DC motor from slow to fast, and back to slow again at a rate determined by the clock period.

Here the ripple counter and R-2R DAC are configured for 4-bit operation but using commonly available binary ripple counters such as the CMOS 4024 7-bit (÷128), the CMOS 4040 12-bit (÷4096) or the larger CMOS 4060 14-bit (÷16,384) counter and adding more input resistors to the R-2R ladder network such as those available from Bournes, the resolution (LSB) of the circuit can be greatly lowered producing a smoother output signal from the *R-2R digital-to-analogue converter*.

Read more Tutorials inCombinational Logic

- 1. Combinational Logic Circuits
- 2. The Multiplexer
- 3. The Demultiplexer
- 4. Priority Encoder

5 Comments

Join the conversation

Error! Please fill all fields.

Your Name

Email Address

Write your comment here

□ Notify me of follow-up comments by email.

Submit

- *M . Nitish Kumar*

  Ossum

  Posted on June 14th 2024 | 6:25 am
  Reply
- *Philippe*

  Thank you very much for these explanations. I love the step-by-step calculations !

Posted on January 20th 2024 | 10:42 am
Reply

- *James*

  What’s the reasoning behind inverting the output voltage?

  Posted on October 22nd 2023 | 7:25 pm
  Reply

  - *Wayne Storr*

    There is no reasoning. The DAC examples given are based around an inverting op-amp for ease of explanation, but you could also use a non-inverting amplifier or a second inverting amplifier. The choice is yours.

    Posted on October 23rd 2023 | 9:50 am
    Reply

- *Bhskhyathi*

  Tnx for helping me❤️

  Posted on June 24th 2023 | 5:28 pm
  Reply

Close