T: Resistor Ladder Networks

A resistor ladder network is an electrical circuit made from cascaded resistor subcircuits. Two configurations are discussed here:

1. A string resistor ladder, as shown in Figure 1, is a voltage-divider configuration that can be used for analog-to-digital conversion, and
2. An *R*–2*R* ladder, as shown in Figure 2, is a simple and inexpensive way to perform digital-to-analog conversion using cascaded precision resistor networks.

String Resistor Ladder Network (Analog to Digital Conversion, or ADC)

A string of many, often equally-dimensioned, resistors connected between two reference voltages is a resistor string ladder network. The resistors act as voltage dividers between the referenced voltages. Each tap along the string generates a different voltage, which can be compared with another voltage. This is the basic principle of a flash ADC (analog-to-digital converter).

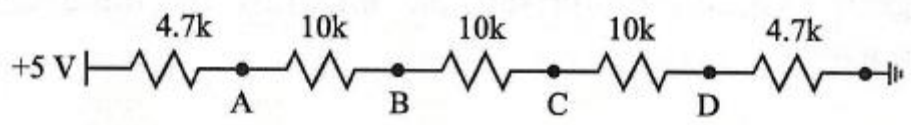

Figure 1. A string resistor ladder network

- Disadvantage: for an *n*-bit ADC, the number of resistors grows exponentially, as 2^n resistors are required, while the *R*–2*R* resistor ladder only increases linearly with the number of bits, as it needs only $2n$ resistors.
- Advantage: higher impedance values can be reached using the same number of components.

R–2*R* Resistor Ladder Network (Digital to Analog Conversion, or DAC

A basic R–2R resistor ladder network is shown in Figure 2. Bits a_{n-1} (most significant bit, MSB) through a_0 (least significant bit, LSB) are driven from digital logic gates. Ideally, the bit inputs are switched between $V = 0$ (logic 0) and V_{ref} (logic 1). The *R*–2*R* network causes these digital bits to be weighted in their contribution to the output voltage V_{out}. Depending on which bits are set to 1 and which to 0, the output voltage (V_{out}) will have a corresponding stepped value between 0 and V_{ref} minus the value of the minimal step, corresponding to bit 0. The actual value of V_{ref} (and the voltage of logic 0) will depend on the type of technology used to generate the digital signals.

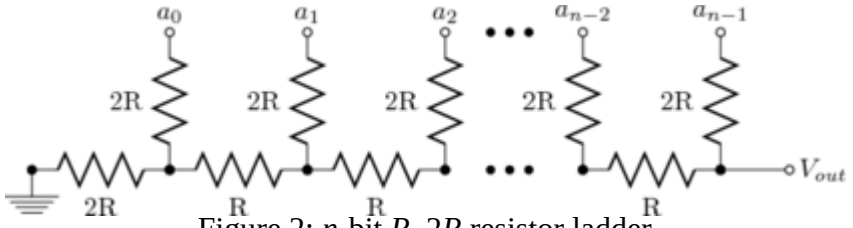

Figure 2: *n*-bit *R*–2*R* resistor ladder

For a digital value VAL, of a R–2R DAC with N bits and $0\text{ V}/V_{ref}$ logic levels, the output voltage V_{out} is:

$$V_o = V_{ref} \times \frac{VAL}{2^N}$$

For example, if $N = 5$ (hence $2^N = 32$) and $V_{ref} = 3.3\text{ V}$ (typical CMOS logic 1 voltage), then V_{out} will vary between 0 volts ($VAL = 0 = 00000_2$) and the maximum ($VAL = 31 = 11111_2$):

$$V_o = 3.3\text{ V} \times \frac{31}{2^5} = 3.196875\text{ V}$$

with steps (corresponding to $VAL = 1 = 00001_2$)

$$\Delta V_o = 3.3\text{ V} \times \frac{1}{2^5} = 0.103125\text{ V}$$

Let's try a quick tutorial analysis of one of the simplest DAC architectures – a 4-bit R-2R resistor ladder network as shown in Figure 3.

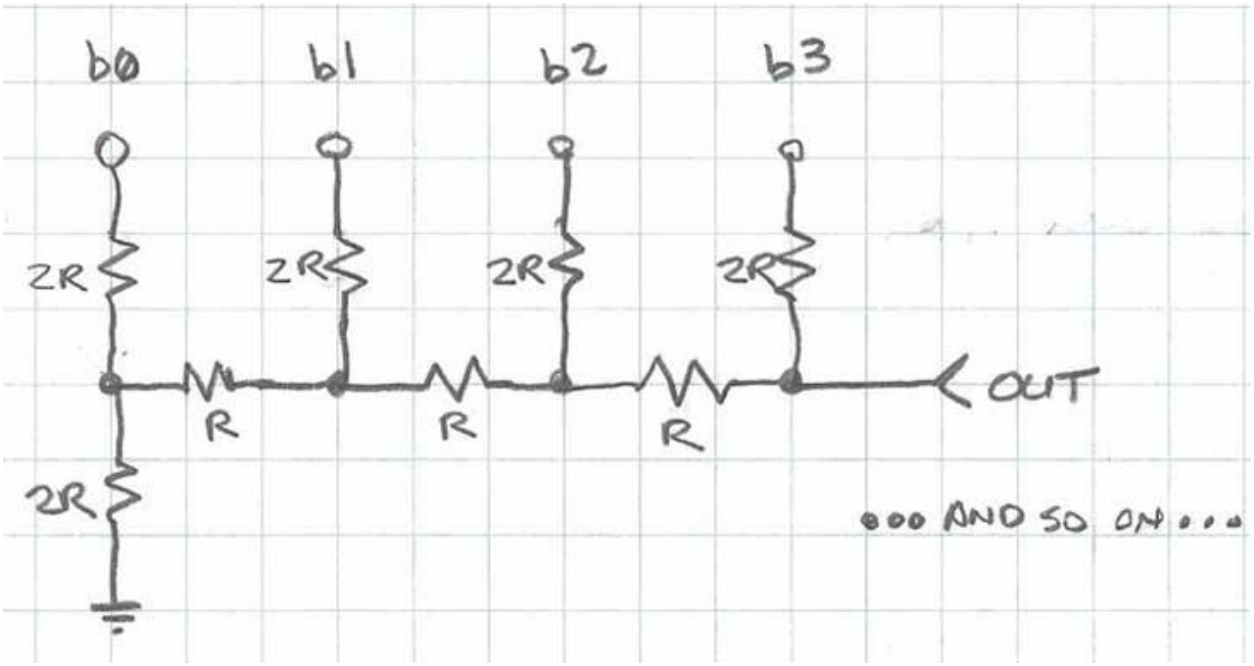

Figure 3: A 4-bit R-2R Network

The R-2R resistor ladder network directly converts a parallel digital symbol/word into an analog voltage. Each digital input (b0, b1, etc.) adds its own weighted contribution to the analog output. This network has some unique and interesting properties.

- Easily scalable to any desired number of bits
- Uses only two values of resistors which make for easy and accurate fabrication and integration
- Output impedance is equal to *R*, regardless of the number of bits, simplifying filtering and further analog signal processing circuit design

How to Analyze the *R*-2*R* Network

Analyzing the *R*-2*R* network brings back memories of the seemingly infinite variety of networks that you're asked to solve during your undergraduate electrical engineering studies. The reality though, is that the analysis of this network and how it works is quite simple. By methodical application of Thevenin Equivalent circuits and Superposition, we can easily show how the *R*-2*R* circuit works.

Let's start off by analyzing the output impedance. Working through the circuit, simplifying it with Thevenin equivalents, makes this process simple. Thevenin says that if your circuit contains linear elements like voltage sources, current sources and resistors, that you can cut your circuit at any point and replace everything on one side of the cut with a voltage source and a single series resistor. The voltage source is the open-circuit voltage at the cut point, and the series resistor is the equivalent open circuit resistance with all voltage sources shorted.

Figure 4 below shows the locations of the "cut lines" we'll use to simplify this circuit to calculate its output impedance. For this analysis, the digital inputs will all be considered shorted to ground.

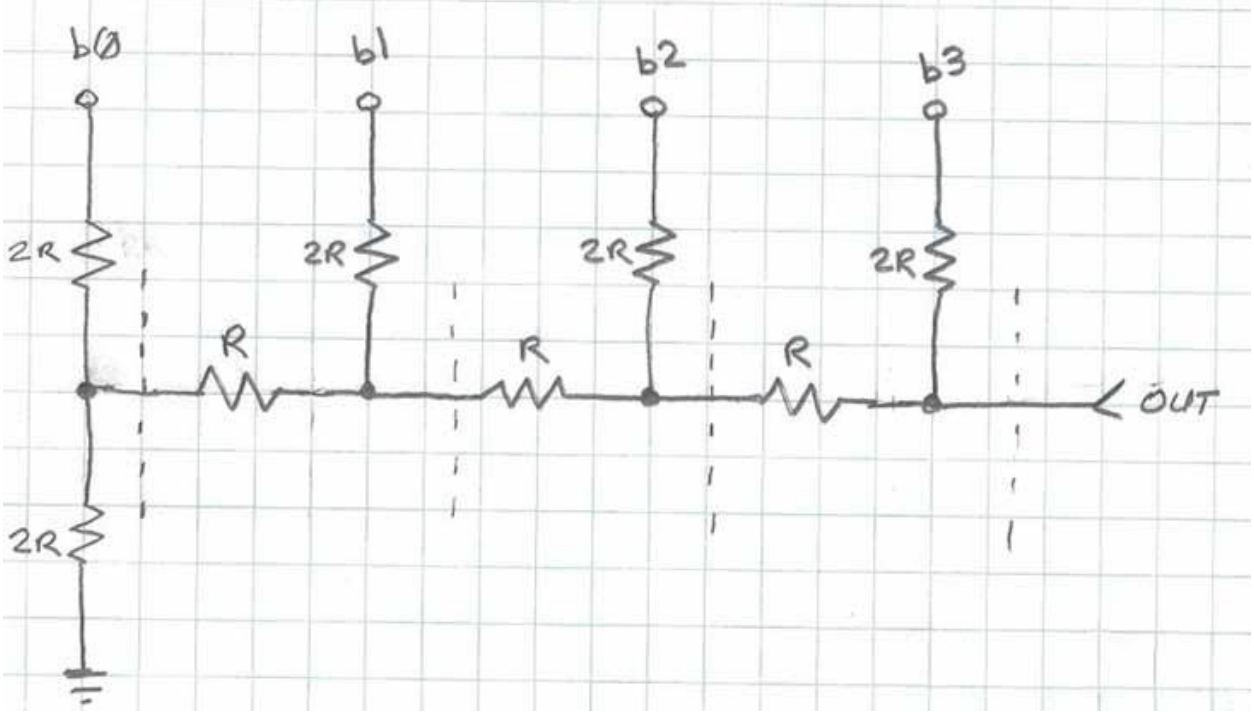

Figure 4: Establishing the cut lines for Thevenin Analysis

The two 2*R* resistors to the left of the first cut line in Figure 4 appear in parallel (when the digital bit b0 is grounded), and can be replaced with a single resistor *R* as shown in Figure 5. The series combination of the two *R* resistors on the left of Figure 5 combine to a single resistor of value 2*R*, which is in parallel with the 2*R* resistor to b1.

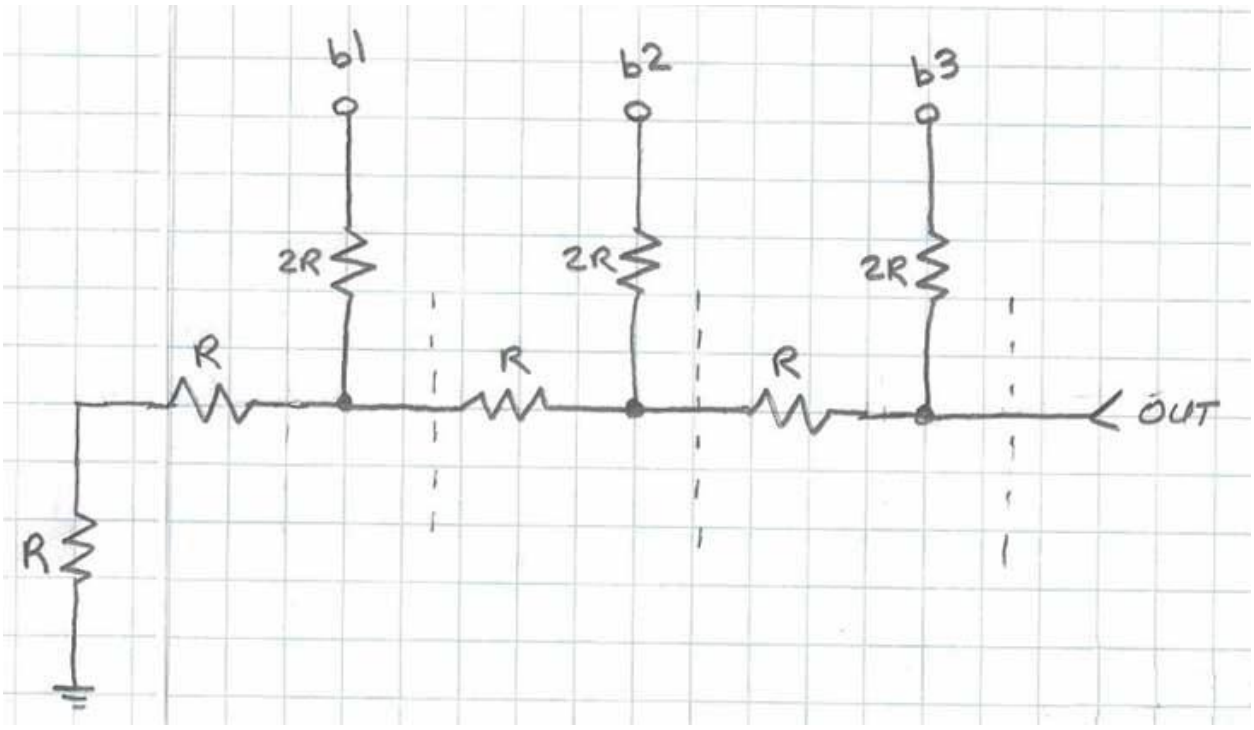

Figure 5: First Thevenin Equivalent

You may notice that this process repeats itself each time we work from left to right, successively replacing combinations of resistors with their equivalents. As one can see in Figure 6, the circuit ultimately simplifies to a single resistor *R*.

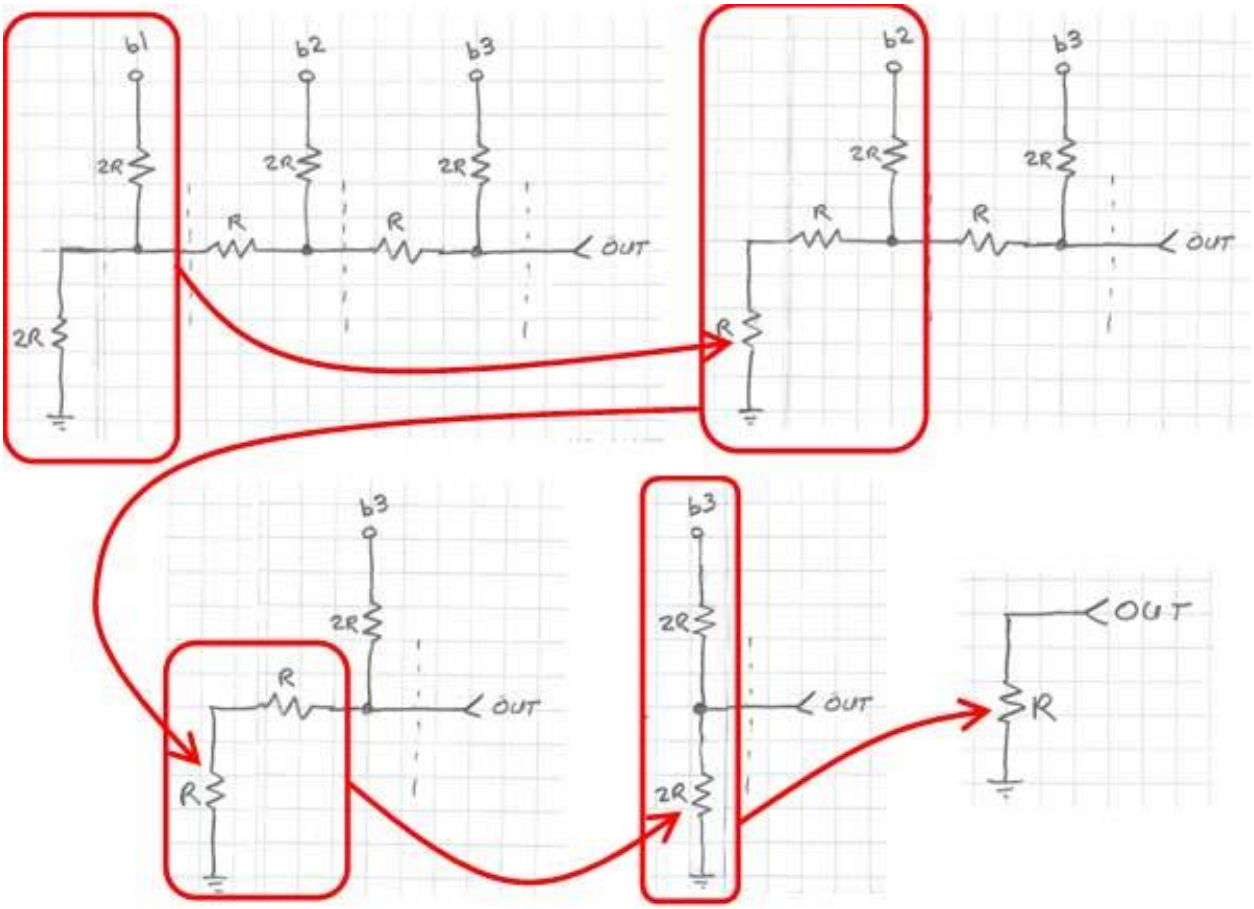

Figure 6: Calculating the equivalent output resistance of the *R*-2*R* network

Thus, the output impedance of the *R*-2*R* resistor network is always equal to *R*, regardless of the size (number of bits) of the network. This simplifies the design of any filtering, amplification or additional analog signal conditioning circuitry that may follow the network.

How to Calculate Analog Voltage Output

Next, we'll look at how to calculate the analog voltage output for a given parallel digital input on the b0, b1, etc. inputs. We'll use the same Thevenin equivalents technique shown above, as well as Superposition. Superposition tells us that if you individually compute the contribution of a given source to the output (with all others voltage sources shorted and current sources opened), you can then sum the results for each of the sources to obtain the final result for the output.

We'll calculate the contribution of two of the bits of our 4-bit *R*-2*R* DAC in Figure 7 to show the process. We'll assume the bits b0 and b2 are logic high, and bits b1 and b3 are logic low (ground).

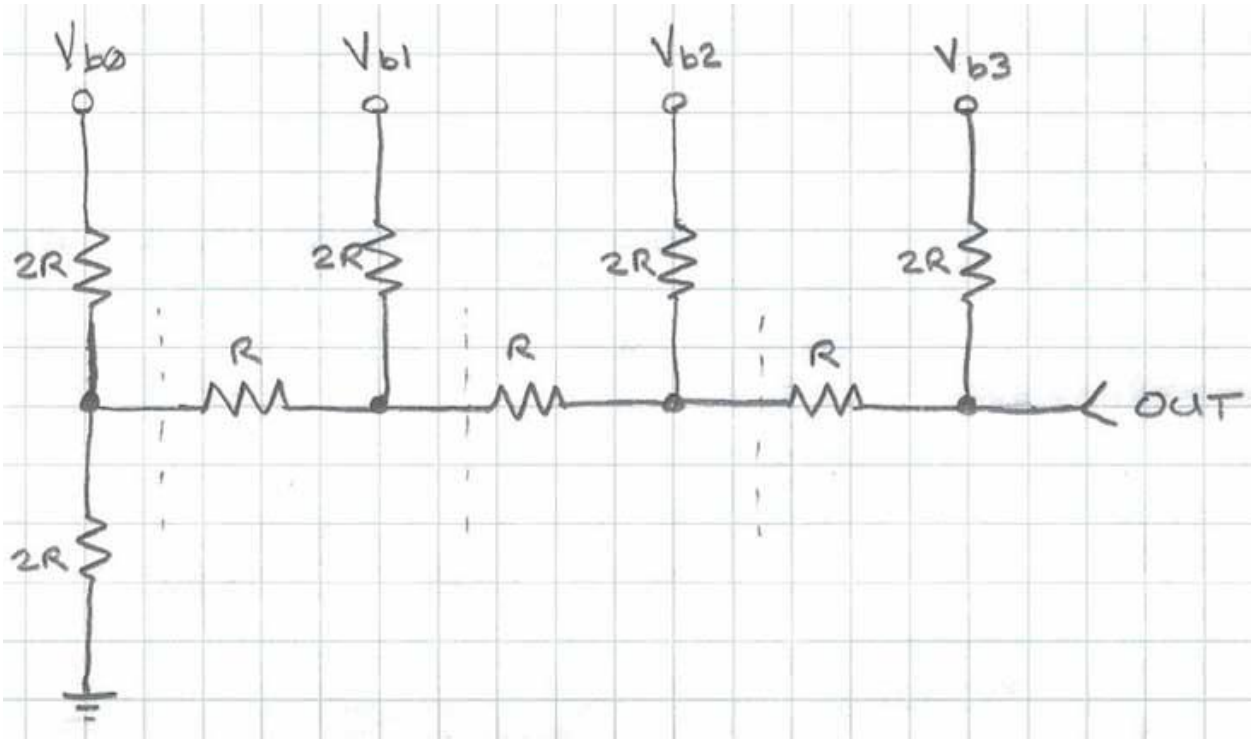

Figure 7: 4-bit *R*-2*R* example

We start by replacing the circuit to the left of the left-most cut-line with its Thevenin equivalent. Figure 8 shows the Thevenin equivalent, which is the series resistor of value *R* (parallel combination of two 2*R* resistors), and the open circuit voltage from the resistor divider (Vb0/2).

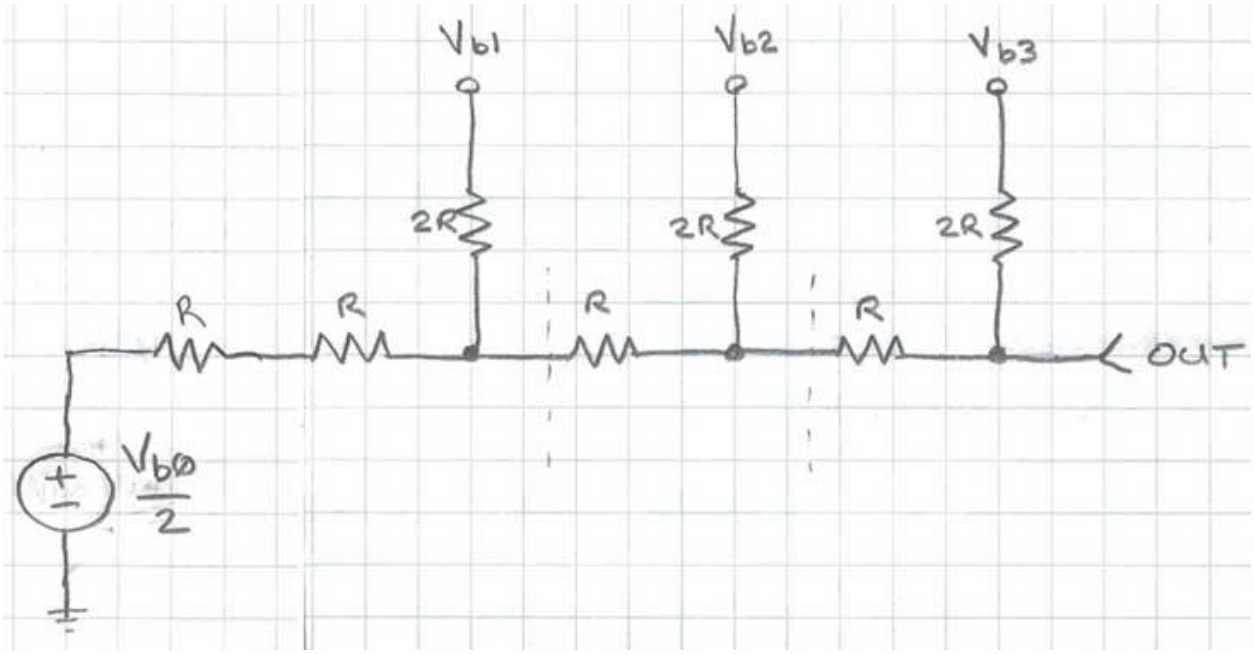

Figure 8: Replacing the first stage with its Thevenin equivalent

The process continues methodically, step by step for each cut-line, substituting the equivalent circuit for each stage, as shown graphically in Figure 9.

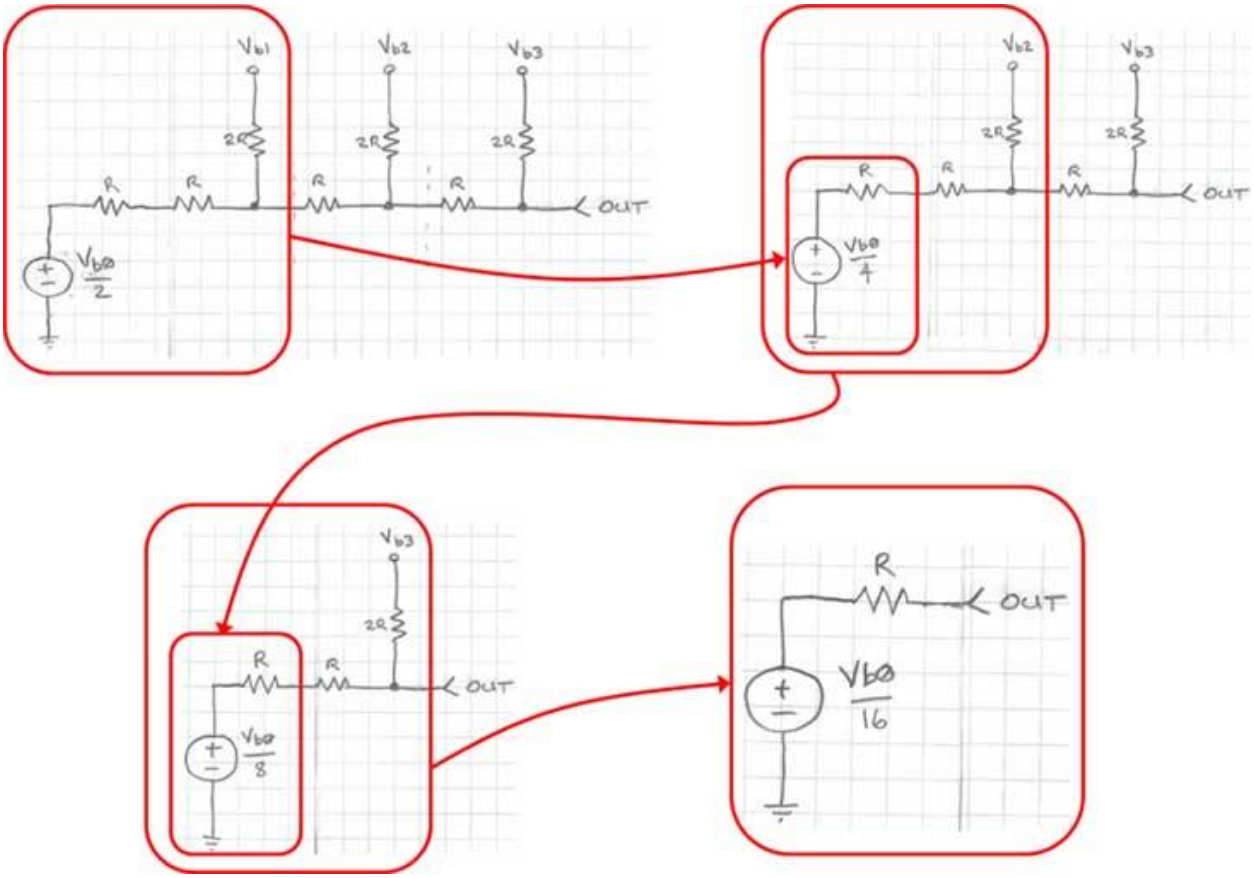

Figure 9: Calculating the contribution of Vb0 to the output

We can see that the voltage contribution from bit b0 is $1/16^{th}$ of the logic high voltage level. Each bit stage that this voltage passes through cuts the contribution by a factor of 2. You may begin to see a theme here…

Next, we'll compute the contribution from bit b2, as shown in Figure 10 below:

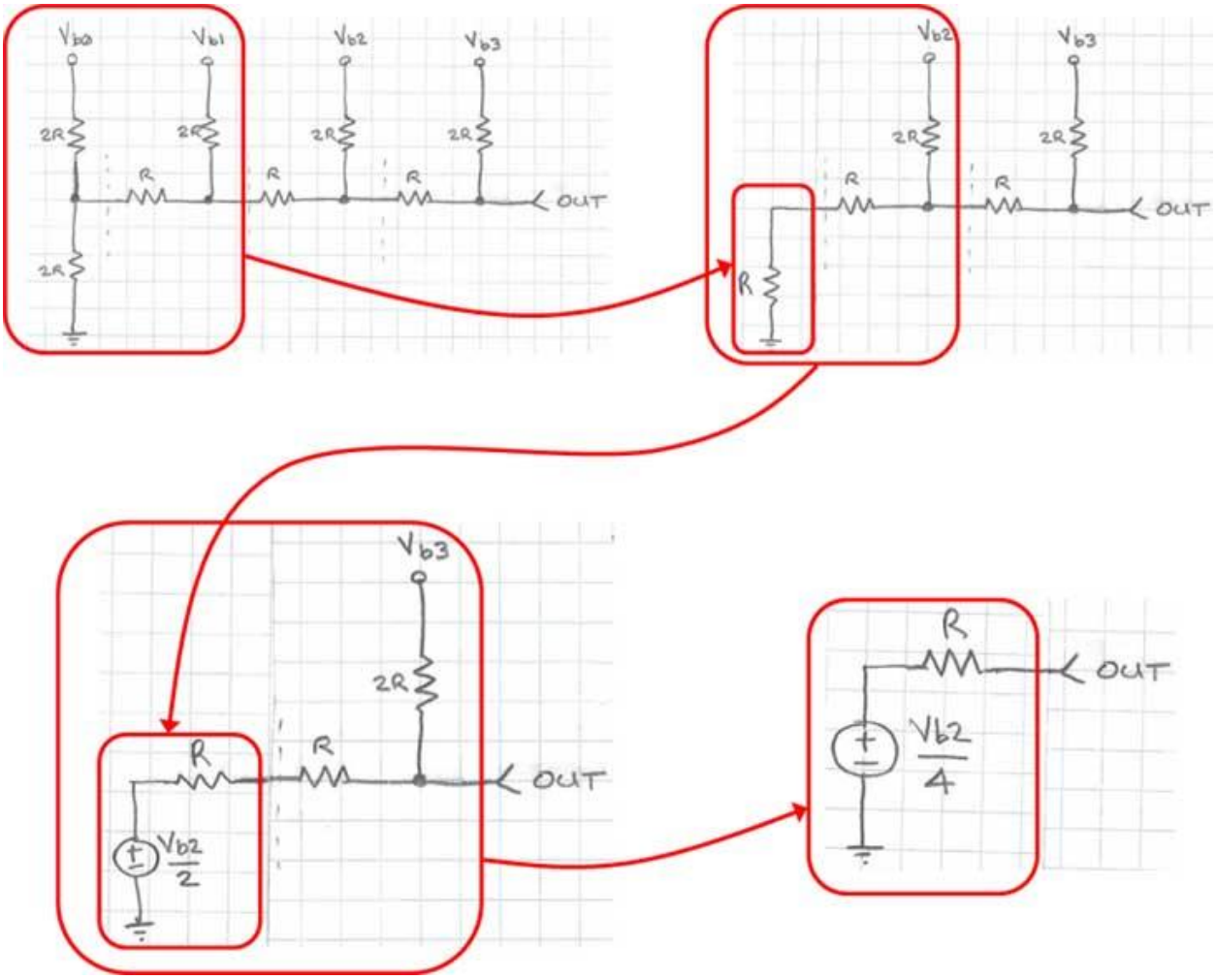

Figure 10: Computing the contribution of bit b2 to the output

From the Thevenin equivalent analysis shown earlier, we know that we can replace any portion of this circuit to the left of any of the cut lines with a resistor of value R, shown as the first step in Figure 10. Next, we follow the same Thevenin equivalent process to the output. As you may have already suspected, the contribution of bit b2 is simply Vb2/4. Thus, the analog output voltage when bits b0 and b2 are equal to logic one is simply given by Vb0/16 + Vb2/4.

In a more general sense, the contribution of each bit to the output is a simple binary weighting function of each bit. As you work back from the MSB to the LSB, the voltage contribution due to each bit is cut in half. Thus, the general form of the equation to calculate the output voltage of the 4-bit *R*-2*R* DAC is:

$$V_{out} = \frac{V_{b0}}{16} + \frac{V_{b1}}{8} + \frac{V_{b2}}{4} + \frac{V_{b3}}{2}$$

The *R*–2*R* resistor ladder-based digital-to-analog converter (DAC) is a simple, effective, accurate, and inexpensive way to create analog voltages from digital values. They are relatively easy to manufacture, since only two resistor values are required (or even one, if *R* is made by placing a

pair of 2*R* in parallel, or if 2*R* is made by placing a pair of *R* in series). It is fast and has fixed output impedance *R*. The *R*–2*R* ladder operates as a string of current dividers, whose output accuracy is solely dependent on how well each resistor is matched to the others. Small inaccuracies in the MSB resistors can entirely overwhelm the contribution of the LSB resistors. This may result in non-monotonic behavior at major crossings, such as from 01111_2 to 10000_2. Depending on the type of logic gates used and design of the logic circuits, there may be transitional voltage spikes at such major crossings even with perfect resistor values. These can be filtered with capacitance at the output node (the consequent reduction in bandwidth may be significant in some applications). Finally, the 2*R* resistance is in series with the digital-output impedance. High-output-impedance gates (e.g., LVDS) may be unsuitable in some cases. For all of the above reasons (and doubtless others), this type of DAC tends to be restricted to a relatively small number of bits; although integrated circuits may push the number of bits to 14 or even more, 8 bits or fewer is more typical.

Accuracy of *R*–2*R* Resistor Ladders

Resistors used with the more significant bits must be proportionally more accurate than those used with the less significant bits; for example, in the *R*–2*R* network discussed above, inaccuracies in the bit-4 (MSB) resistors must be insignificant compared to $R/32$ (i.e., much better than 3%). Further, to avoid problems at the $10000_2 - \text{to} - 01111_2$ transition, the sum of the inaccuracies in the lower bits must be significantly less than $R/32$. The required accuracy doubles with each additional bit: for 8 bits, the accuracy required will be better than 1/256 (0.4%). Within integrated circuits, high-accuracy *R*–2*R* networks may be printed directly onto a single substrate using thin-film technology, ensuring the resistors share similar electrical characteristics. Even so, they must often be laser-trimmed to achieve the required precision. Such on-chip resistor ladders for digital-to-analog converters achieving 16-bit accuracy have been demonstrated. On a printed circuit board, using discrete components, resistors of 1% accuracy would suffice for a 5-bit circuit, however with bit counts beyond this the cost of ever-increasing precision resistors becomes prohibitive. For a 10-bit converter, even using 0.1% precision resistors would not guarantee monotonicity of output. This being said, high resolution *R*-2*R* ladders formed from discrete components are sometimes used, with the nonlinearity being corrected in software (e.g., the Korad KA3005D power supply).