Beirut Arab University
Faculty of Science
Mathematics and Computer Science Department
Lab Sheet # 1

Course: Computer Organization & Architecture – CMPS 343
Semester: Fall 2022-2023
Instructor: Dr. May Itani
TA: Miss Nour Fatayerji

Part One: Number Systems & Conversions

Converting hexadecimal to binary

Let's say we have a hexadecimal number, D, and we want to convert it to binary. Since $16=2^4$, four digits of a binary number can represent one digit of a hexadecimal number. D=13, and in binary, can be represented as 1101.

Now, to represent a big hexadecimal number, such as 4E9, we first find the binary values of each individual digit.

4 = 0100

E = 1110

9 = 1001

Therefore, 4E9 = 0100 1110 1001 in binary.

Converting binary to hexadecimal

Every 4 binary digits represent a single hexadecimal digit. So if we have a binary number, such as 10110100011, this can be represented as a hexadecimal number, by breaking it into groups of fours. However, the number 10110100011 has 11 digits, which is not a multiple of 4. To fix this, we just add 0s to the front until it is a multiple of 4, so we have 010110100011. Now, breaking this into fours, we get:

0101 = 5

1010 = A

0011 = 3

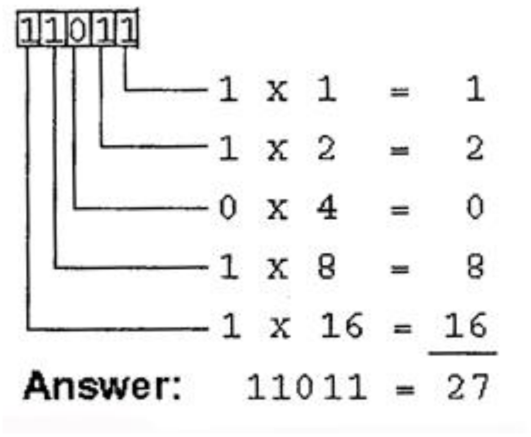
Therefore, the binary number 10110100011 is 5A3 in hexadecimal.

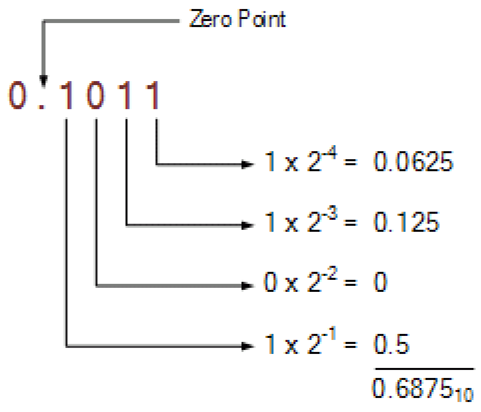
Converting binary to decimal

We multiply each binary digit by its weighted position, and add each of the weighted value together.

Weight value:

2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
32	**16**	**8**	**4**	**2**	**1**	**0.5**	**0.25**	**0.125**

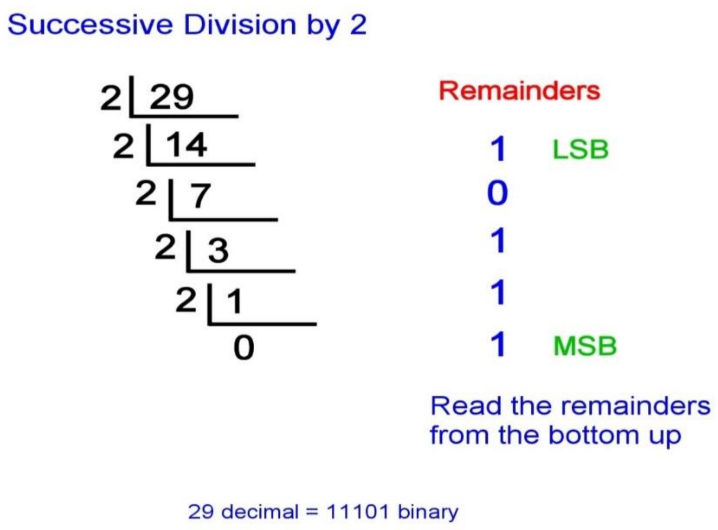




Converting decimal to binary

To convert decimal numbers into binary numbers, repeated division by 2 is needed.

1. Write down the decimal number.
2. Divide the number by 2.
3. Write the result underneath.
4. Write the remainder on the right-hand side. This will be 0 or 1.
5. Divide the result of the division by 2 and again write down the remainder.
6. Continue dividing and writing down remainders until the result of the division is 0.
7. The most significant bit (MSB) is at the bottom of the column of remainders and the least significant bit (LSB) is at the top.
8. Read the series of 1s and 0s on the right from the bottom up. This is the binary equivalent of the decimal number.



We will illustrate the method of converting decimal fractions into binary representation by converting the decimal value .625

Step 1: Begin with the decimal fraction and multiply by 2. The whole number part of the result is the first binary digit to the right of the point.

Because .625 x 2 = 1.25, the first binary digit to the right of the point is a 1. So far, we have .625 = .1??? . . . (base 2).

Step 2: Next we disregard the whole number part of the previous result (the 1 in this case) and multiply by 2 once again. The whole number part of this new result is the *second* binary digit to the right of the point. We will continue this process until we get a zero as our decimal part or until we recognize an infinite repeating pattern.

Because .25 x 2 = 0.50, the second binary digit to the right of the point is a 0. So far, we have .625 = .10?? . . . (base 2).

Step 3: Disregarding the whole number part of the previous result (this result was .50, so there is no whole number part to disregard in this case), we multiply by 2 again. The whole number part of the result is now the next binary digit to the right of the point.

Because .50 x 2 = 1.00, the third binary digit to the right of the point is a 1. So now we have .625 = .101?? . . . (base 2).

Step 4: In fact, we do not need a Step 4. We are finished in Step 3, because we had 0 as the fractional part of our result there.
Hence the representation of .625 = .101 (base 2).

Exercises:

1- **Convert $(37)_{10}$ and $(59)_{10}$ to binary, octal, and hexadecimal.**

2- **Convert the following binary numbers to decimal.**

a) $(100110110)_2$

b) $(1001100.0101)_2$

3- **Convert the following decimal numbers to the stated number system.**

a) $(83.85)_{10} = (?)_8$

b) $(71.82)_{10} = (?)_{16}$

c) $(25.25)_{10} = (?)_2$

4- **Convert the following numbers to the stated number system.**

a) $(BF2.D)_{16} = (?)_8$

b) $(312.20)_4 = (?)_8$

c) $(F13)_{16} = (?)_8$

d) $(8AA)_{16} = (?)_8$

5- **Convert the following numbers from binary to hexadecimal and octal.**

a) 10011011
b) 1001001

6- **Perform the following arithmetic operations.**

a) 10101011 + 11001001
b) 110110 * 1010

Part Two: Logisim

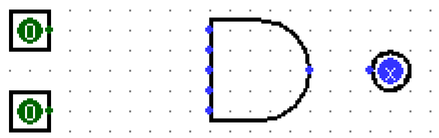
Exercise 1: AND gate

We'll begin by creating a very simple circuit just to get the feel for placing gates and wires.

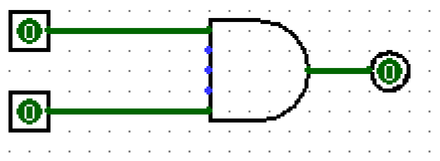
1. Start by clicking the "AND gate" button. This will cause the shadow of an AND gate to follow your cursor around. Click once within the main schematic window to place an AND gate.

2. Click the "Input Pin" button. Now, place two input pins somewhere to the left of your AND gate.

3. Click the "Output Pin" button. Then place an output pin somewhere to the right of your AND gate. Your schematic should look something like this at this point:



4. Click the "Select tool" button. Click and drag to connect the input pins to the left side of the AND gate. This will take several steps, as you can only draw vertical and horizontal wires. Just draw a wire horizontally, release the mouse button, then click and drag down starting from the end of the wire to continue vertically. You can attach the wire to any pin on the AND gate on the left side. Repeat the same procedure to connect the output (right side) of the And Gate to the LED. After completing these steps your schematic should look roughly like this:

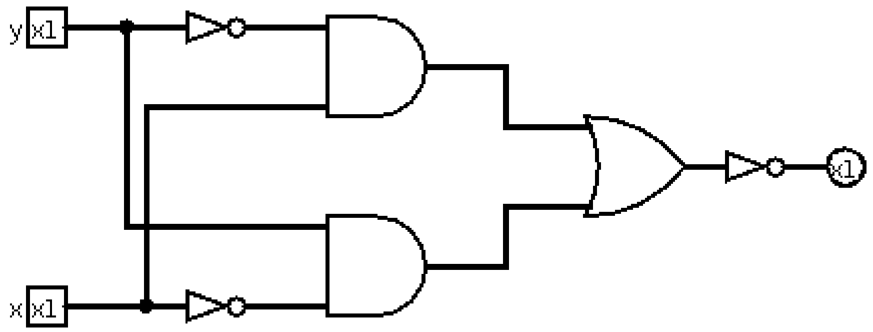


5. Finally, click the "Poke" tool and try clicking on the input pins in your schematic. Observe what happens. Does this match with what you think an AND Gate should do?

<u>Exercise 2: Equality between two 3-bit inputs</u>

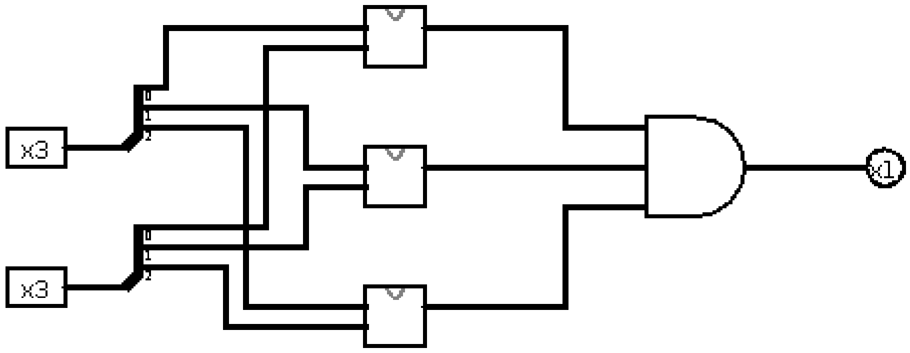
Here we design a simple circuit that checks for equality between two 3-bit inputs. We begin by creating a circuit to determine equality between two single bits by building an XNOR gate. in the following manner:

- Begin by placing two AND gates and an OR gate
- Reduce the number of input bits to 2 for each of the gates via the Attribute Table.
- Place a NOT gate in front of one input of each AND gate as well as the output of the OR gate.
- Connect wires by dragging out from each input/output point as in Figure.
- This creates an XNOR gate, which will output a 1 when both inputs are equal.



- Now that we can check for equality between 1-bit we will make our XNOR gate a sub-circuit. We will then use the XNOR gate to check for equality between the two 3-bit inputs.
- Begin by selecting Project > Add Circuit
- Name your new circuit 'equality.'
- Right-click on equality, and click 'Set as Main Circuit.'
- You can now add your XNOR circuit into the equality circuit by clicking and dropping it as with any other gate.
- We can now finish our equality circuit in a few simple steps.
- Add 3 of your XNOR circuits to the equality circuit.
- Add two inputs we will check for equality.
- Select each input and in the attribute Table change the number of data bits to 3, the outgoing wire will now be a BUS (multiple wires that appear as one wire).
- Place two splitters to check individual bits. The splitter is found in Explorer Plane > Wiring > Splitter
- Select the splitters and change the Bit Width In and the Fan Out to 3 bits (because our inputs are 3-bits wide).
- From the splitter match each of the corresponding bits to an XNOR.
- Drop an AND gate onto the circuit.
- Connect all three XNOR outputs to the AND gate.

- Drop an OUTPUT into the circuit and connect to the AND gate.



This circuit can now check for equality between the two 3-bit inputs. You can alter the inputs to test your circuit by selecting the poke tool (hand shaped tool) and clicking on the individual bits of the inputs. As you change the bit values you will notice the output changing, 1 indicating equality and 0 indicating not equal.