# 4CS015 Fundamentals of Computing – Workshop #4

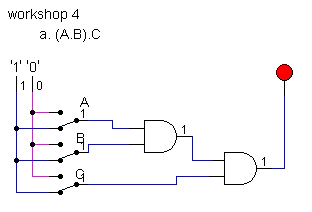
Start by working through the Introduction to Logsim document on CANVAS. Then

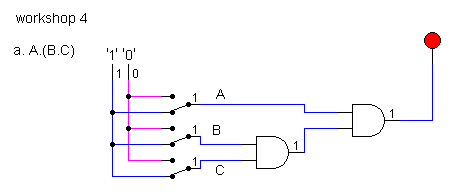
1. Use logsim to build equivalent logic circuits for both sides of the logic equations below. Prove that the expressions are equivalent by completing the truth tables. Insert an image of your circuit under each truth table. ***(10 marks for each correct and complete answer)***

1. (A.B).C = A.(B.C)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | (A.B).C | A.(B.C) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

[Insert your GIF image of your logic circuit here]

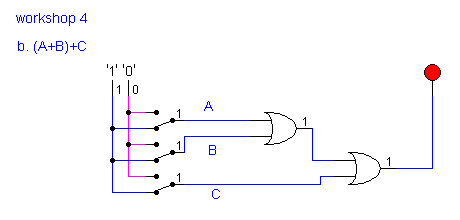


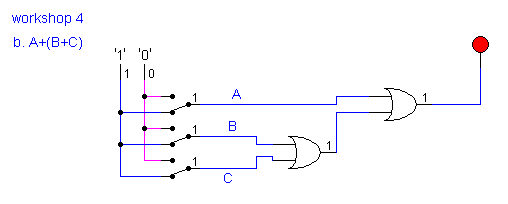


1. (A+B)+C = A+(B+C)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | (A+B)+C | A+(B+C) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

[Insert your GIF image of your logic circuit here]

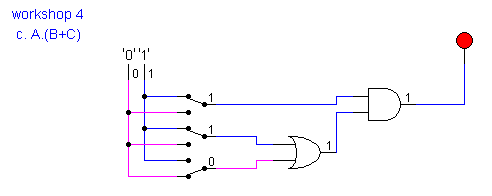


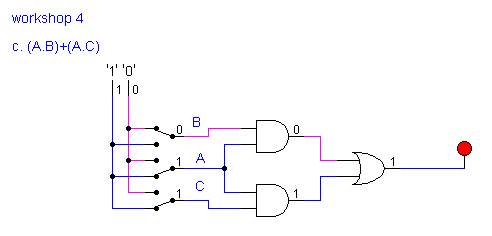


1. A.(B+C) = (A.B)+(A.C)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | A.(B+C) | (A.B)+(A.C) |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

[Insert your GIF image of your logic circuit here]

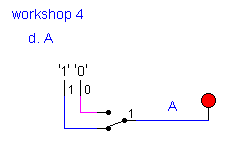


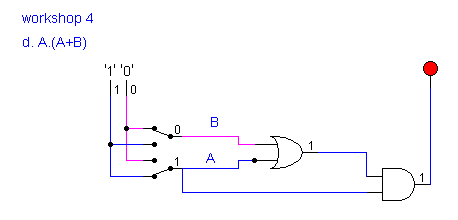


1. A.(A+B) = A

|  |  |  |
| --- | --- | --- |
| A | B | A.(A+B) |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

[Insert your GIF image of your logic circuit here]

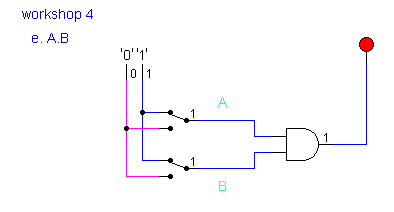


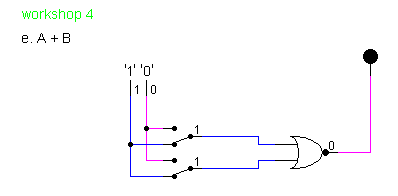


1. A . B = not(A + B)

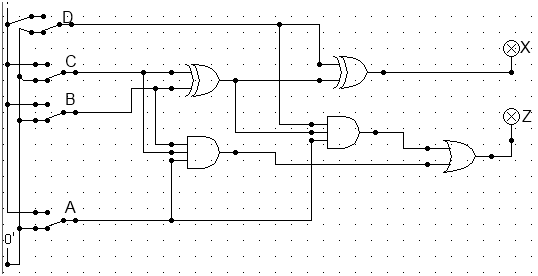
|  |  |  |  |
| --- | --- | --- | --- |
| A | B | A.B | not(A + B) |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 |

[Insert your GIF image of your logic circuit here]





1. Build the following circuit using LogSim and fill in the truth table showing the value of the two outputs for all input conditions.



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D | X | Z |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |

In your own words, describe the function of the above circuit. How did you come to this conclusion? Look very carefully at the truth table and compare it with others that we’ve discussed in the lectures. As a final hint, look at how we turn one mathematical operation in logic into another. ***(25 marks for a correct and complete answer)***

[Insert your answer here]

1. Using the last 4 *unique* digits of your student number as input, e.g. 0791377 would be 9137, design and build a password circuit (a circuit with a single output that lights an LED for your unique input pattern in binary) using LogSim that will give a true output when your number, expressed in a suitable binary form, is present on the inputs and 0 for all other 4 digit numbers between 0000 and 9999. Switches will therefore be needed for all inputs.

To get you started:

* 1. Decide how to represent your 4 digit number in 1’s and 0’s
  2. Put a switch for each binary input.

***(25 marks for a working answer)***

[Insert your GIF image of your logic circuit here]