**Lab 7**

**To Demonstrate the Working of a Digital Comparator**

***Note: You may draw all the logic diagrams with hand and paste the pictures here or on logicly software with your name, roll number & section mentioned in your workspace. Make sure that all of your connections are clearly visible and distinguishable. In logicly, use “text” label to point out/show all your inputs & outputs***

**Tasks**

1. **Construct a logic circuit for a 2 bit magnitude comparator Also write the Boolean expression for output(s). Simulate your circuit in logicly software.**

**Hint: Take 2 bits of each input i.e. A1A0 & B1B0**

2-Bit Magnitude Comparator

1. Truth Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Input** | | | | **Output** | | |
| **A1** | **A0** | **B1** | **B0** | **A<B** | **A=B** | **A>B** |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

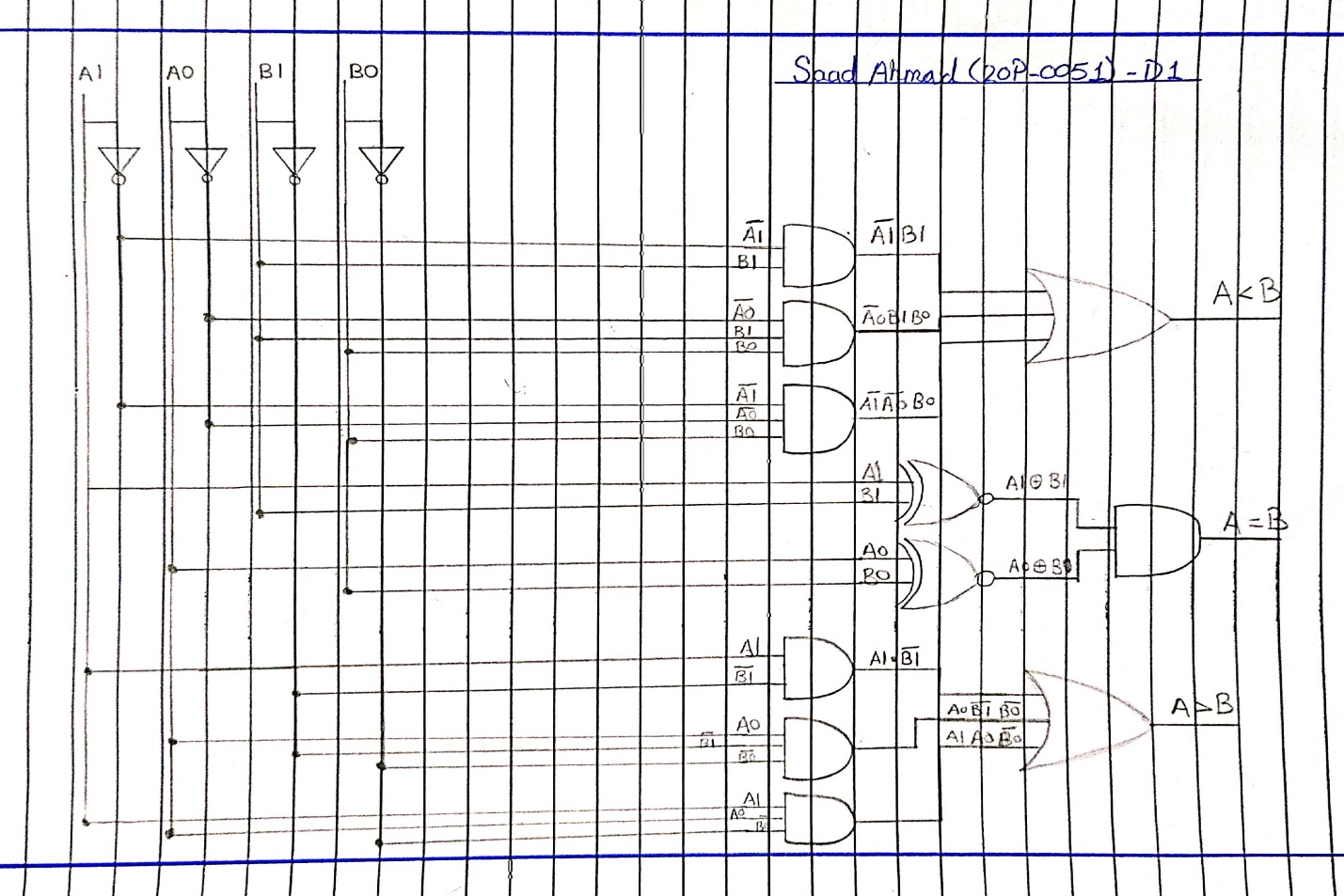
1. Boolean Expression (Simplified)

A<B = A1’B1 + A0’B1B0 + A1’A0’B0

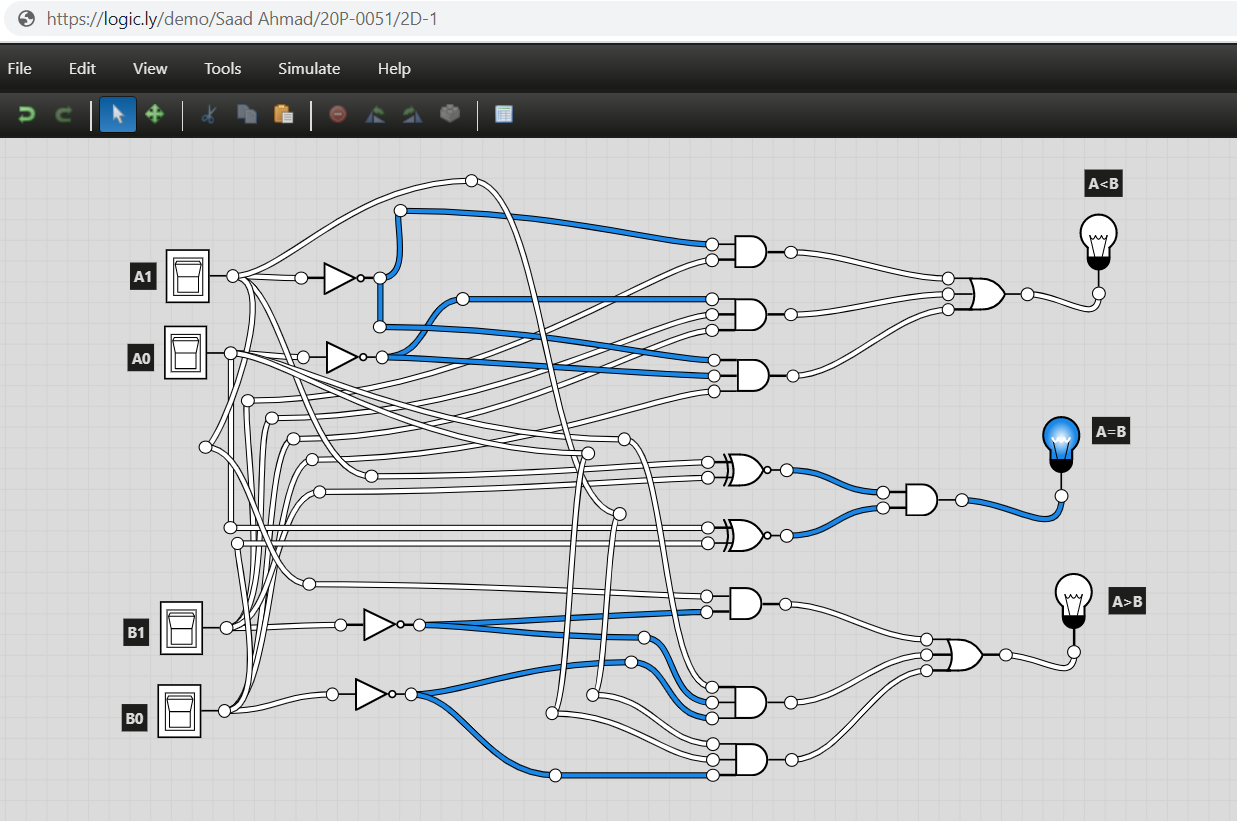
A=B = NOT [(A0 ⊕ B0)] . NOT [(A1 ⊕ B1)]

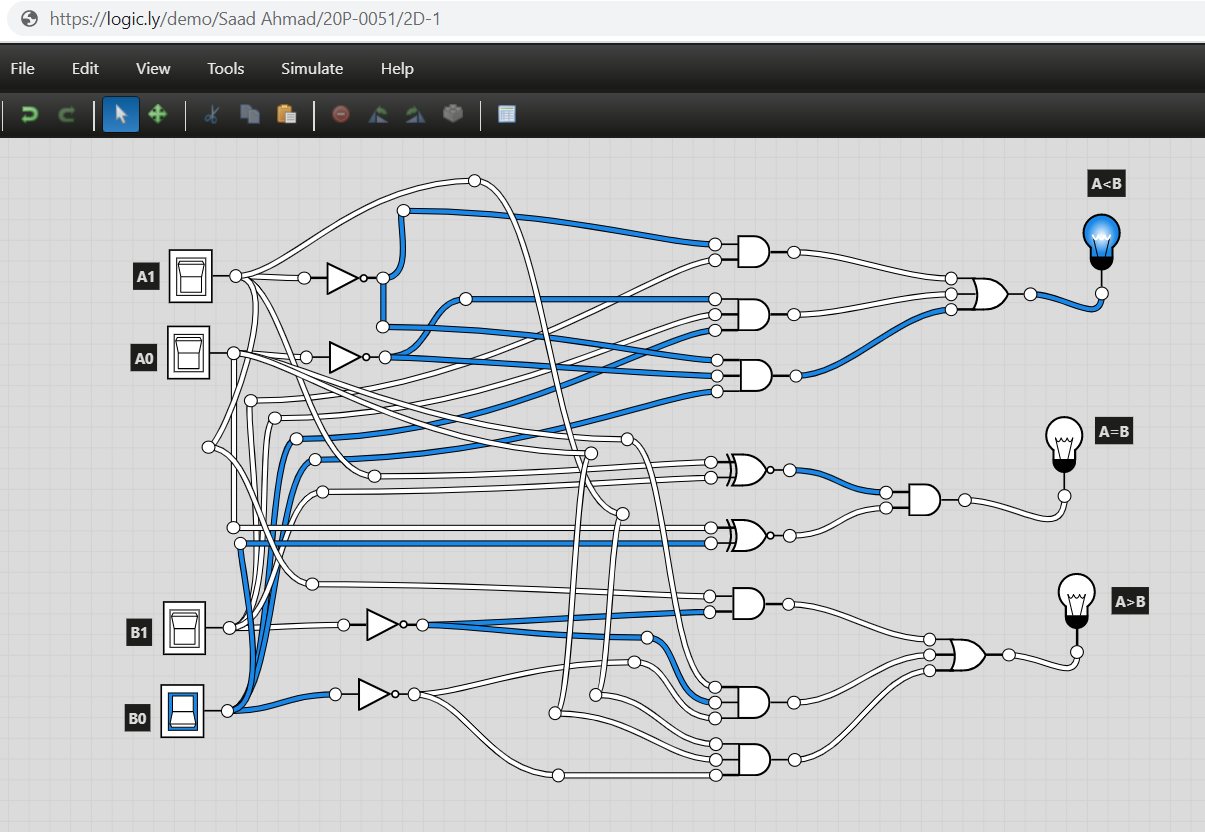
A>B = A1B1’ + A0B1’B0’ + A1A0B0’

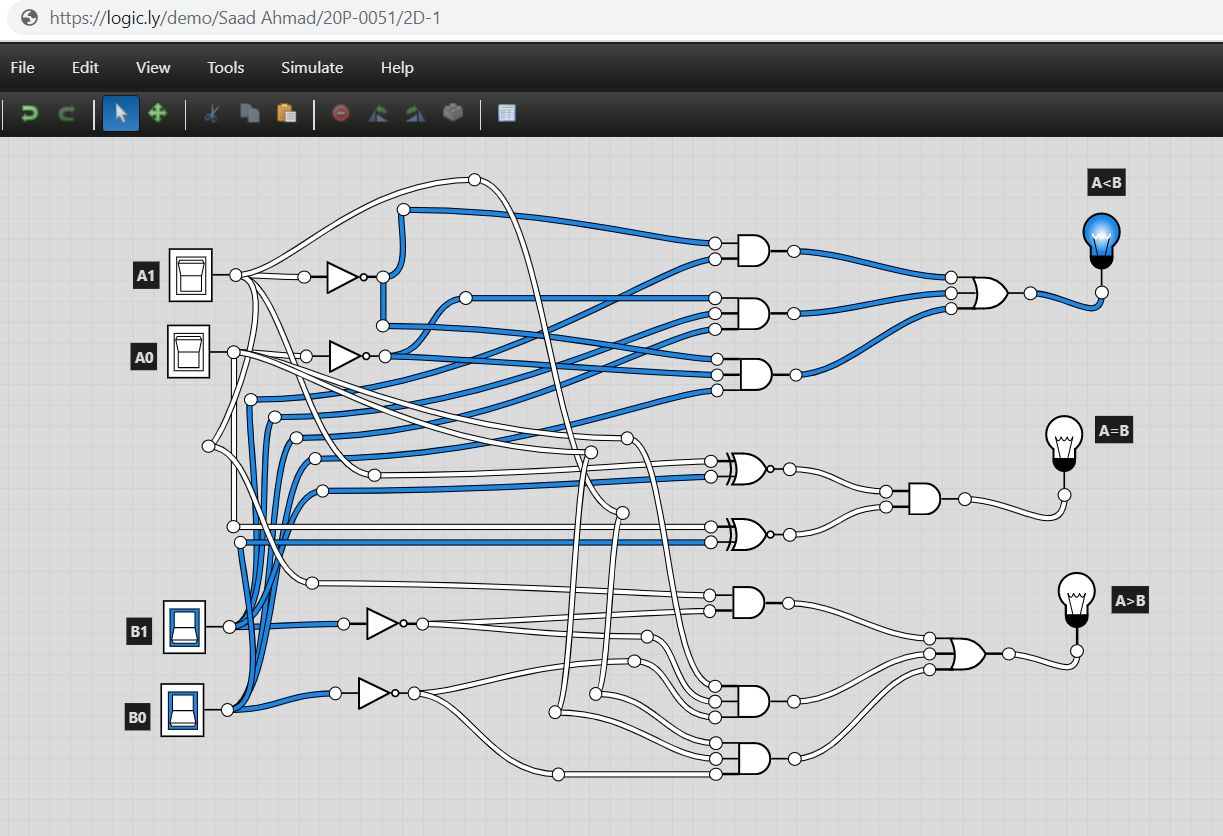
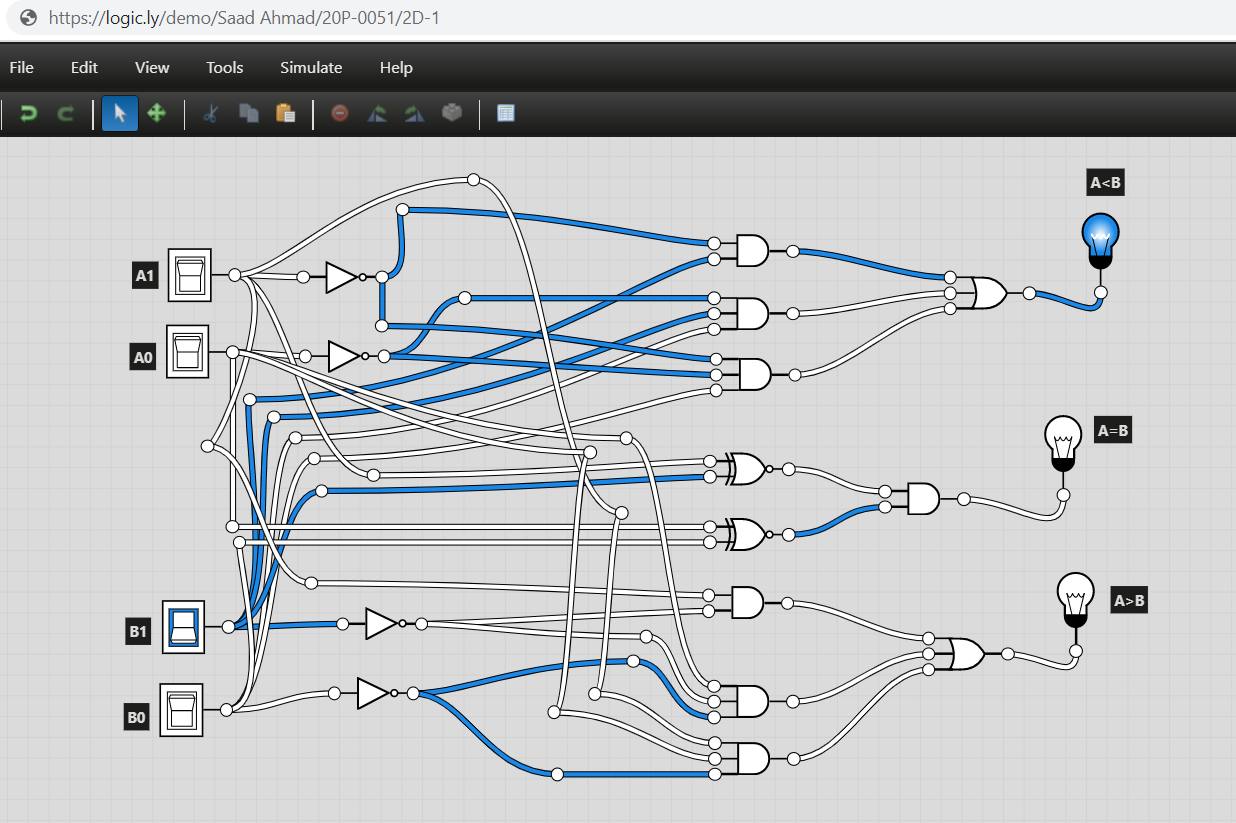
1. Logic Diagram

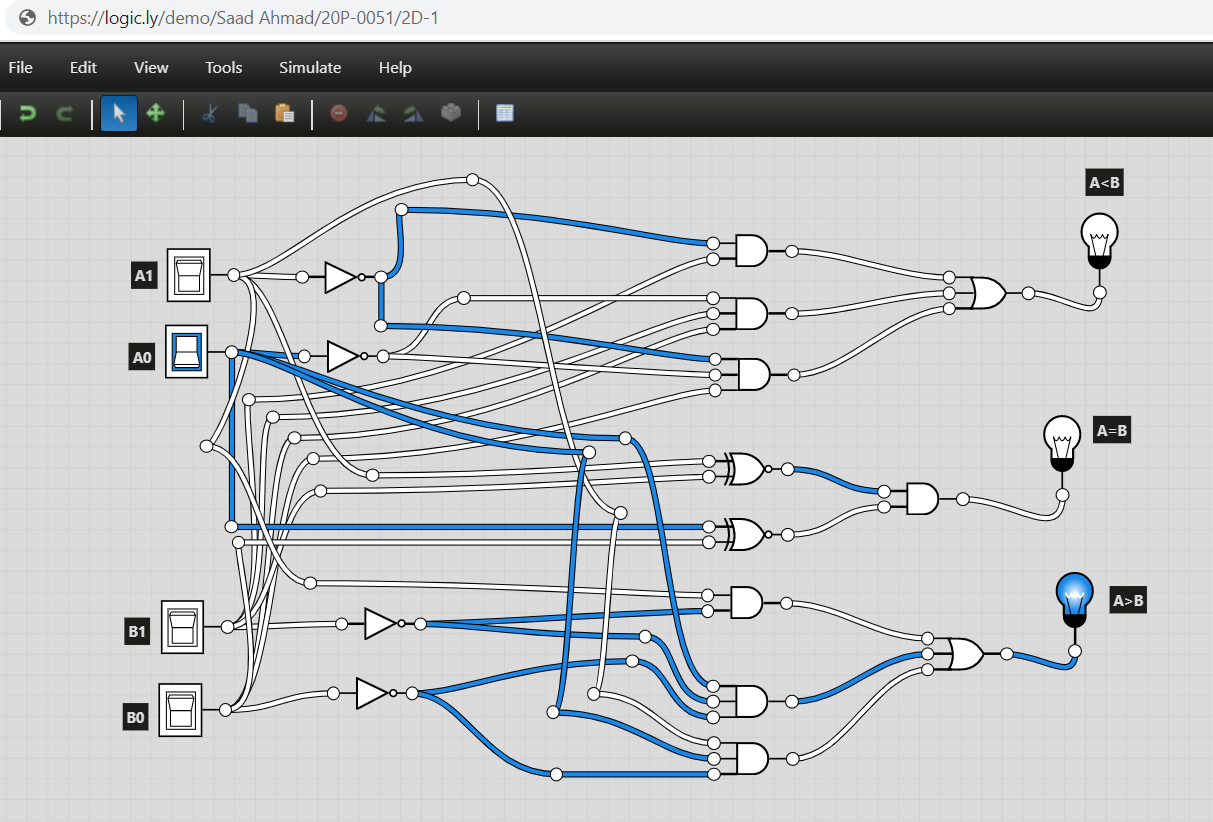


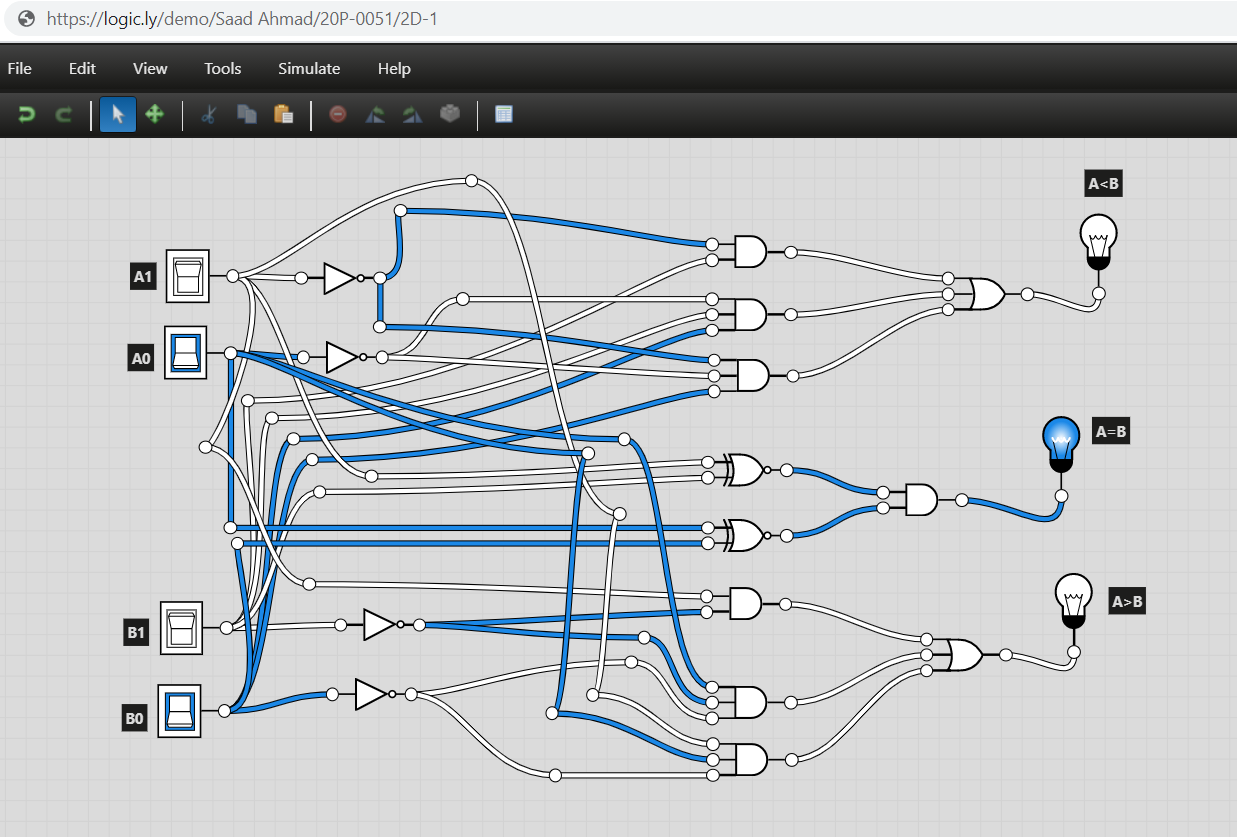
1. Software Simulation (Show here your results for each combination that gives a high output)

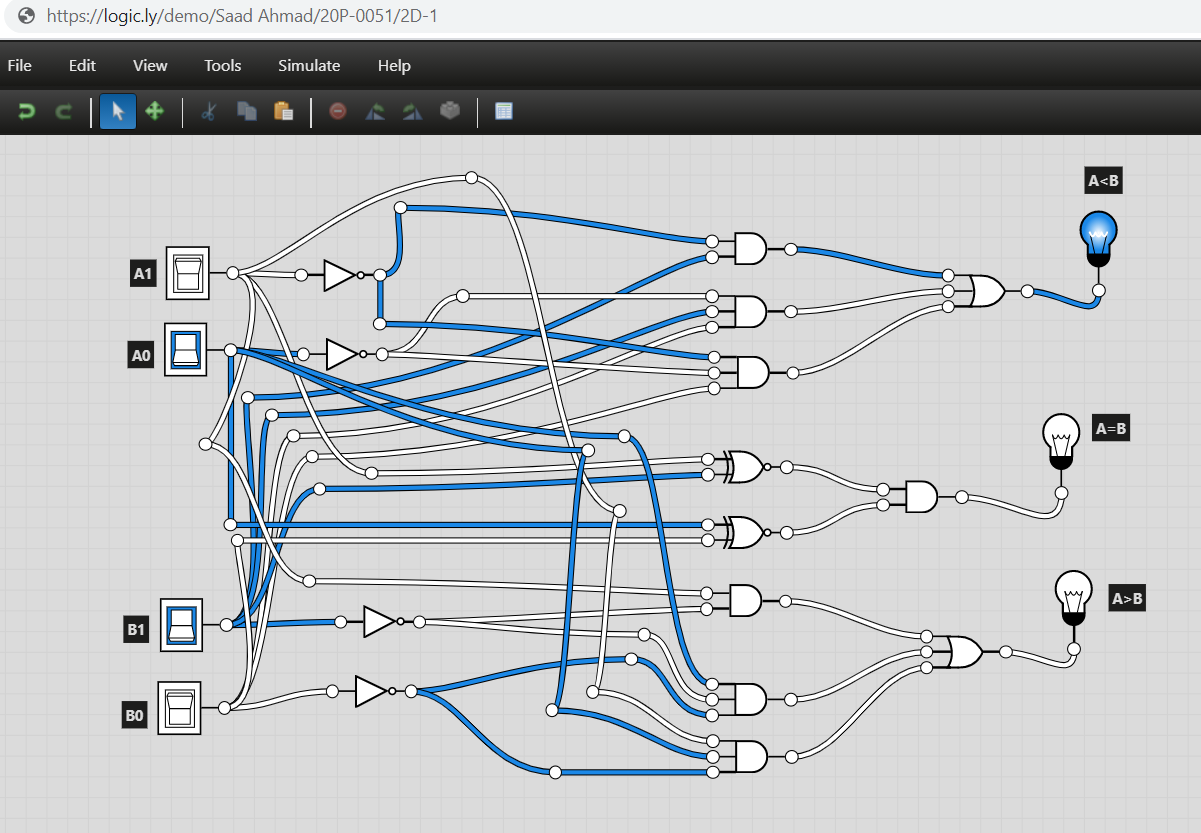


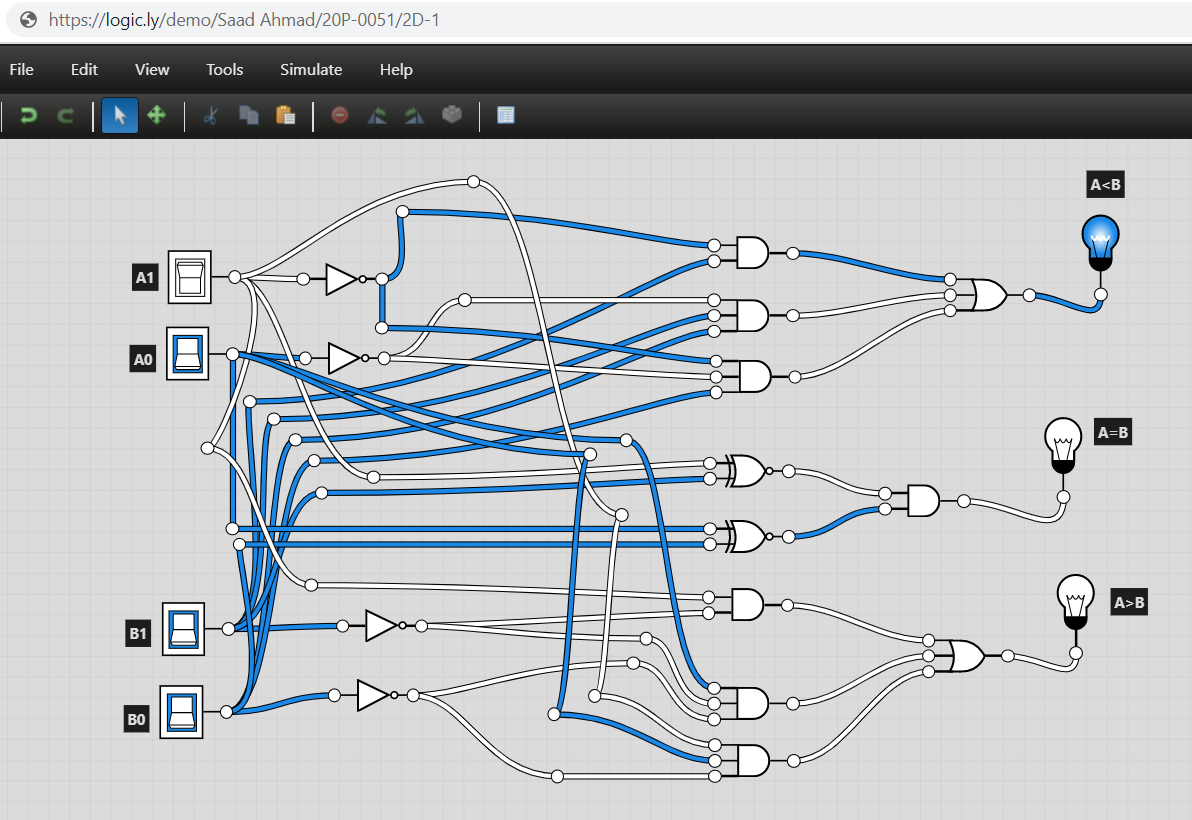


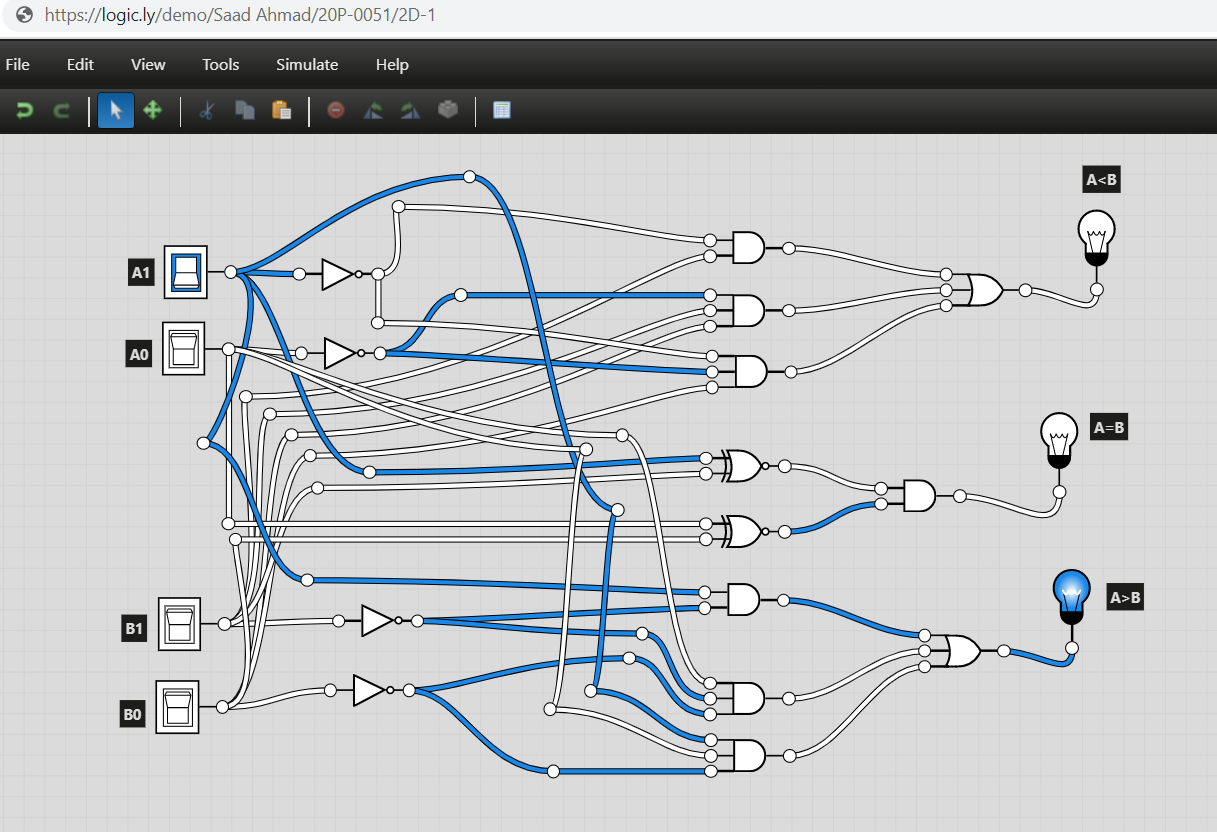


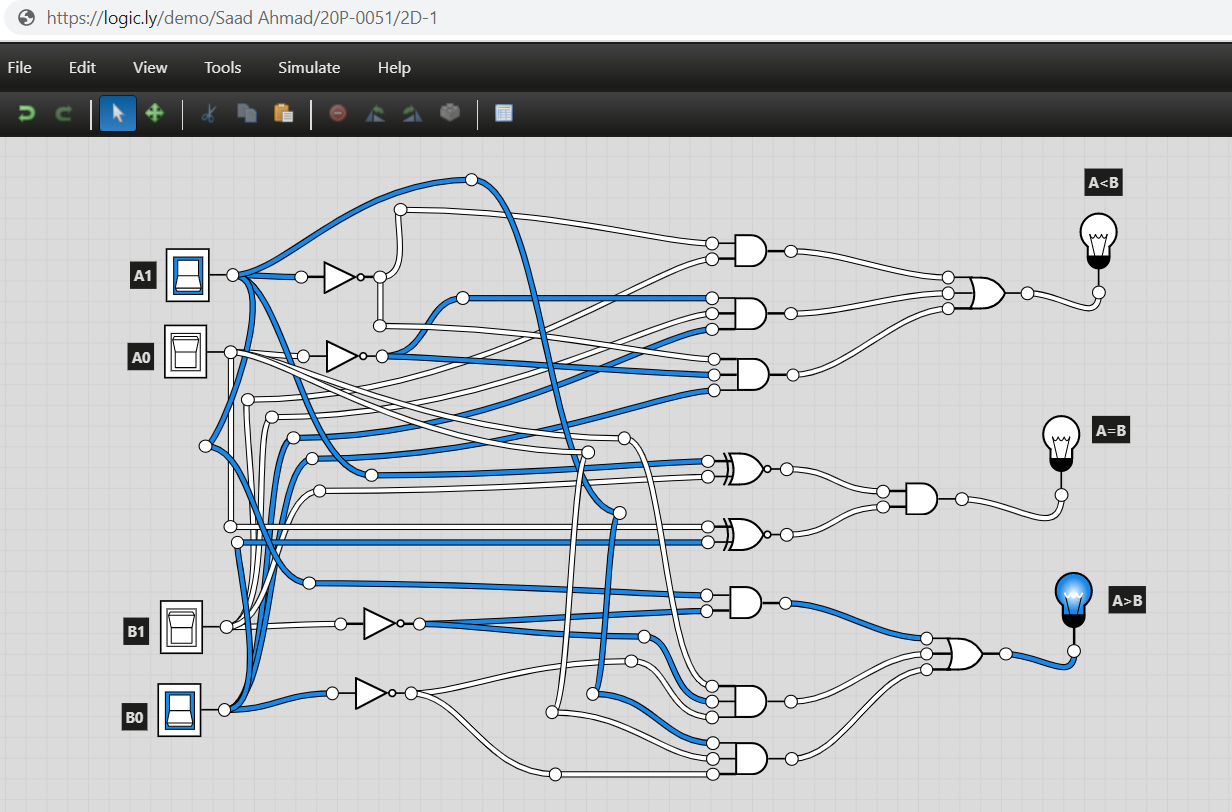


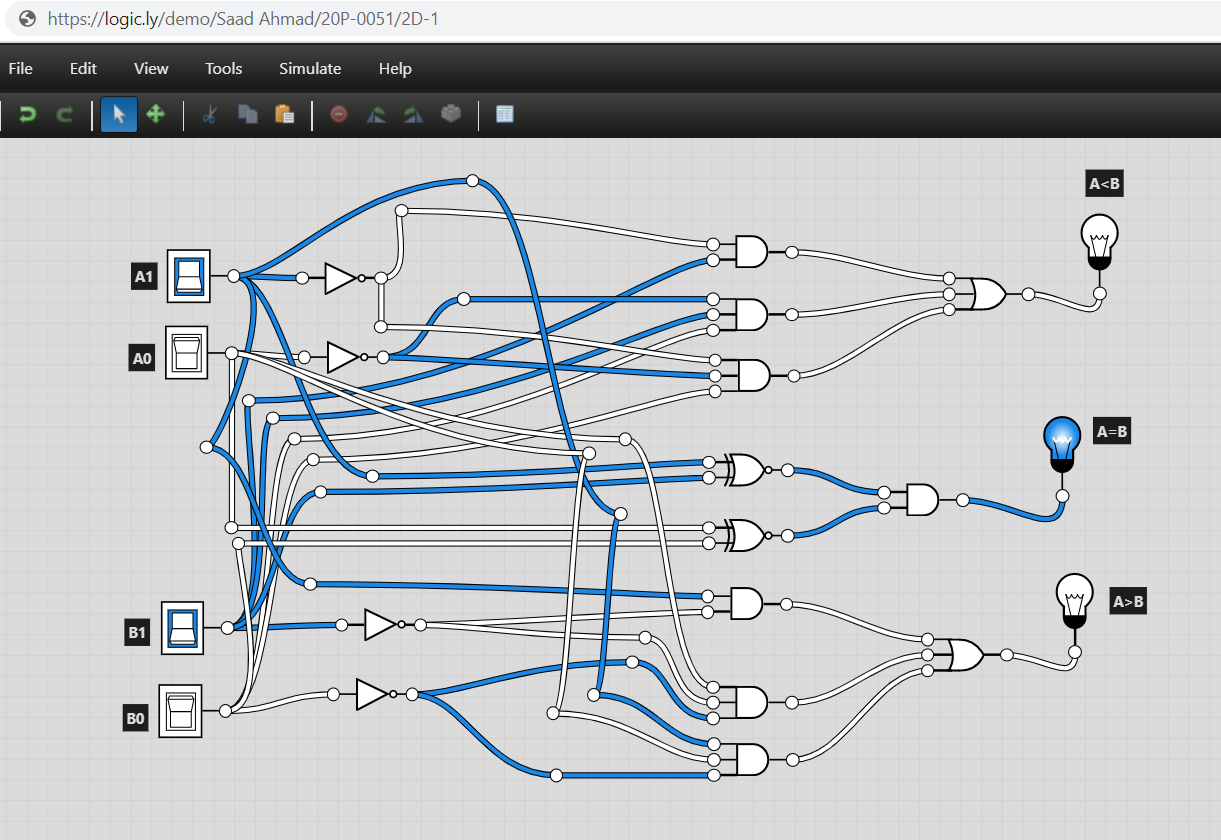




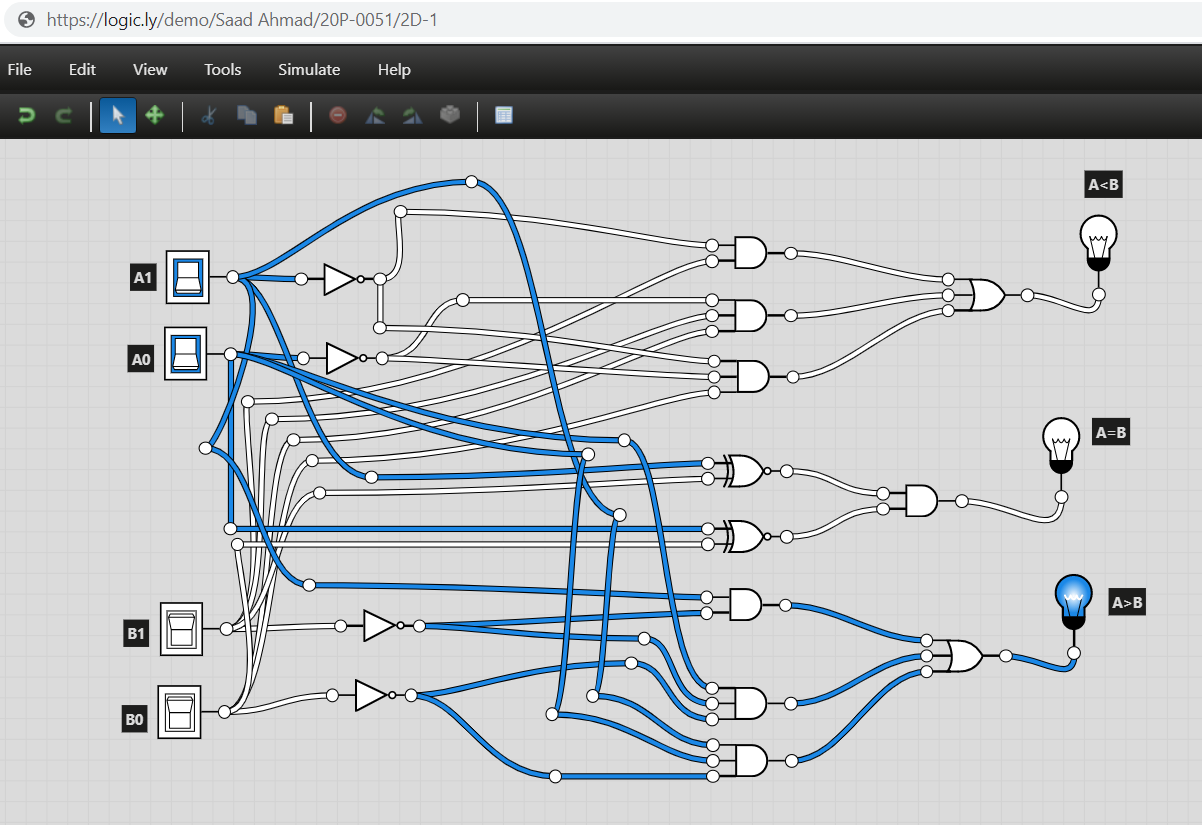


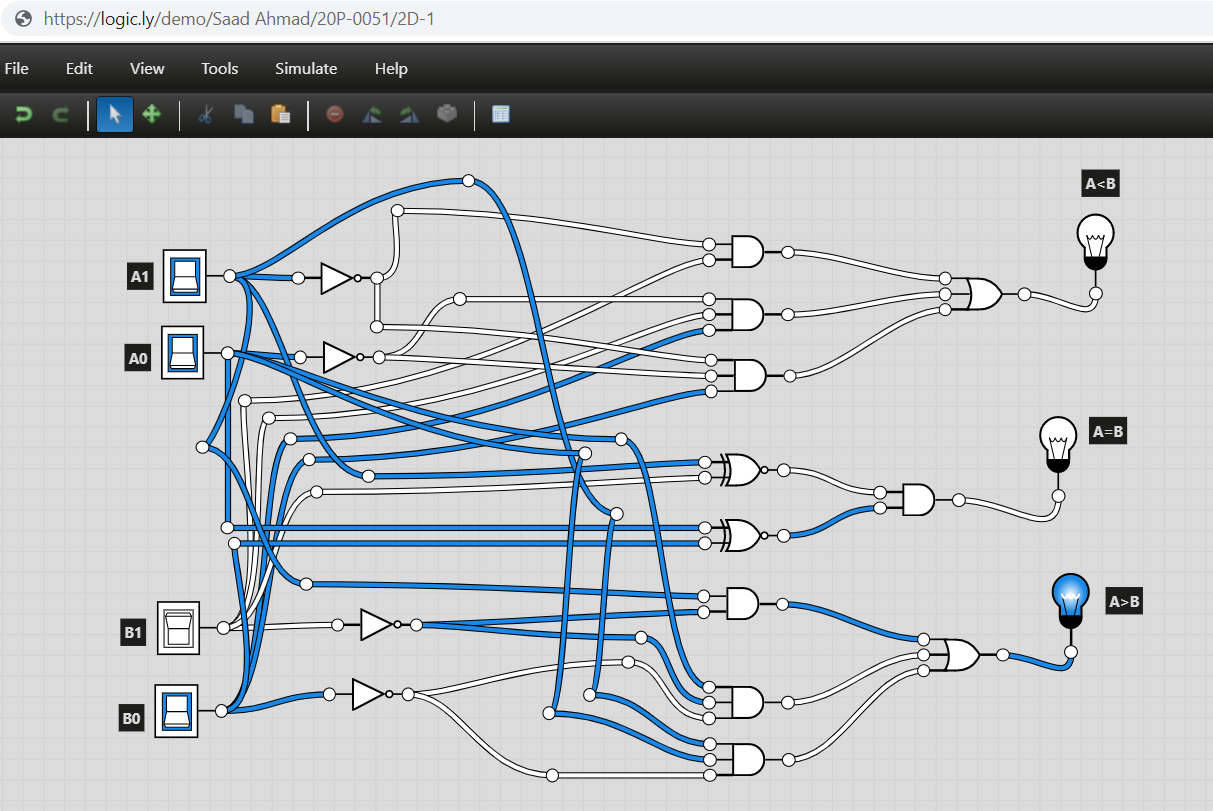


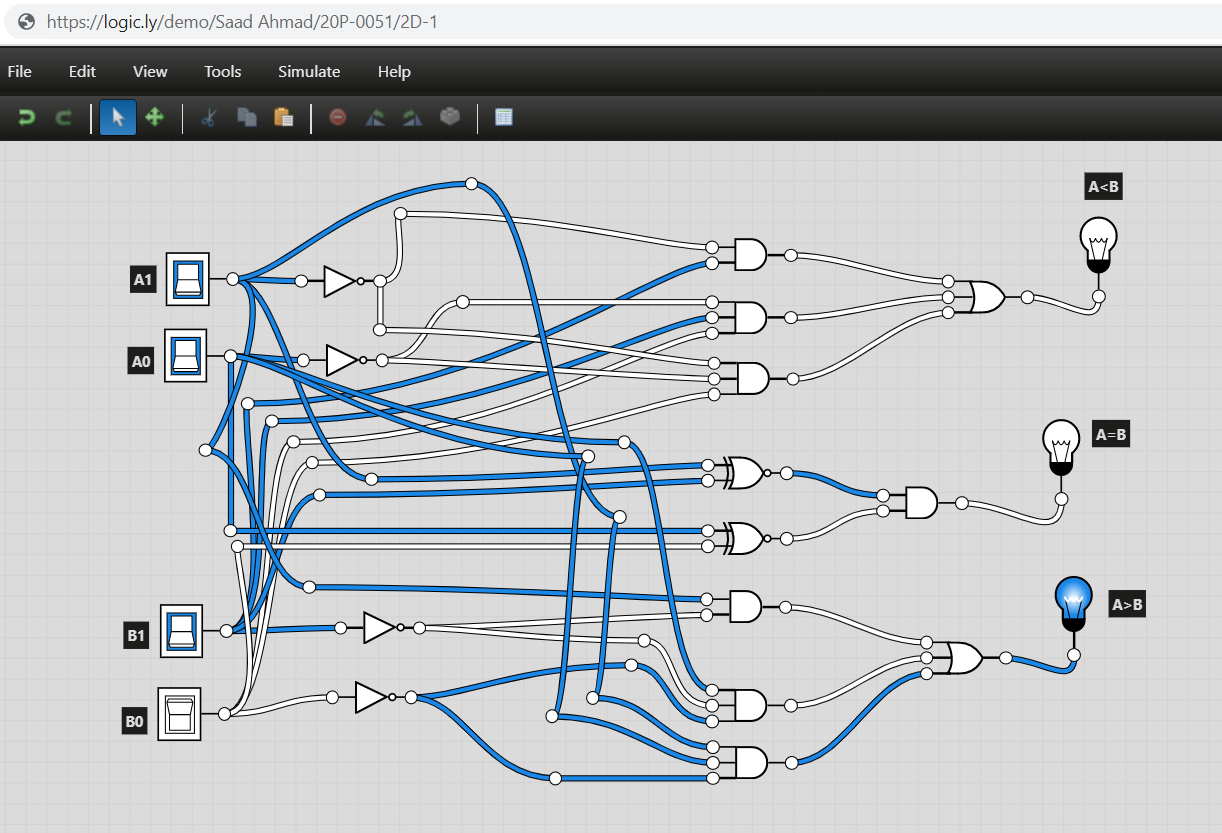


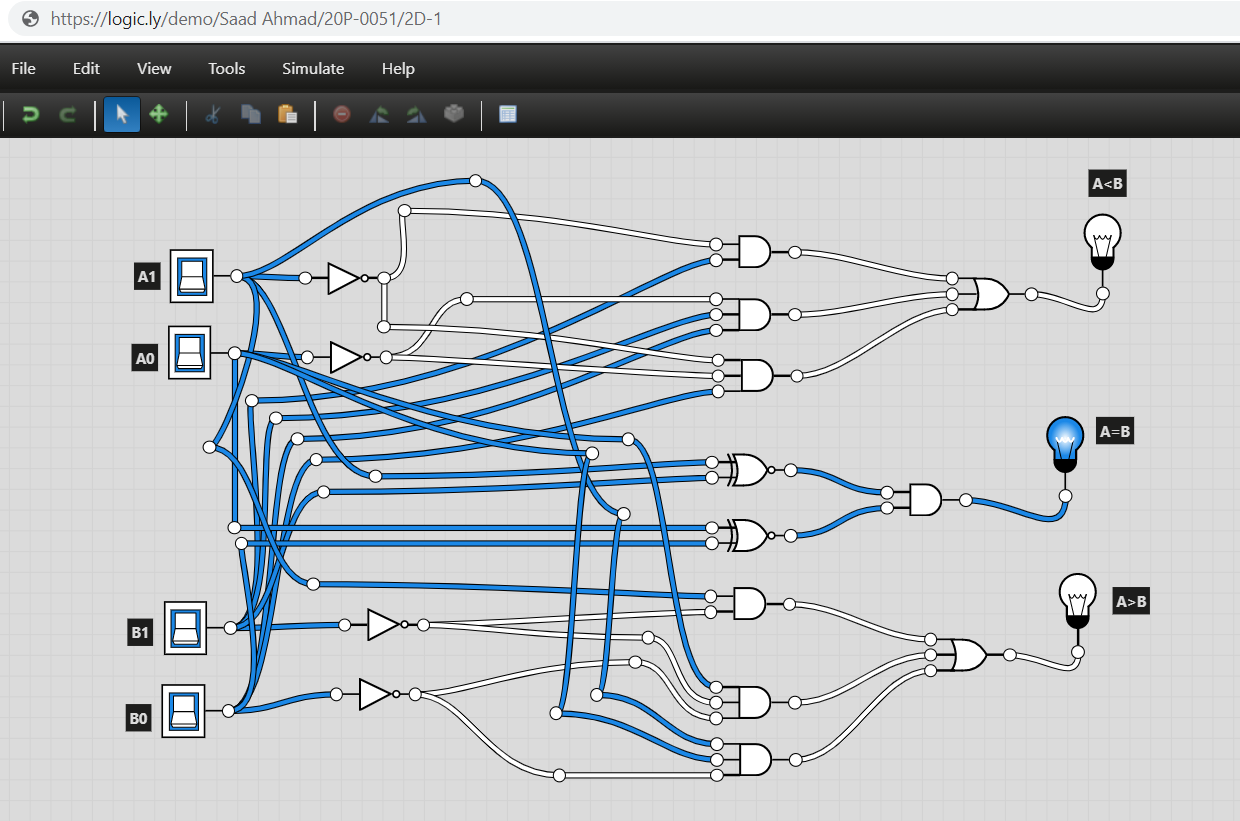












1. **Construct a logic circuit for a 4-bit magnitude comparator Also write the Boolean expression for output(s). Simulate your circuit in logicaly software.**

**You may take help from the logic diagram available on the Internet and compare it with yours for better understanding.**

**The logic circuit should be hand drawn (neatly) with all necessary labels (inputs/outputs).**

4-Bit Magnitude Comparator

1. Truth Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Input** | | | | **Output** | | |
| **A3B3** | **A2B2** | **A1B1** | **A0B0** | **A>B** | **A<B** | **A=B** |
| A3>B3 | x | x | x | 1 | 0 | 0 |
| A3<B3 | x | x | x | 0 | 1 | 0 |
| A3=B3 | A2>B2 | x | x | 1 | 0 | 0 |
| A3=B3 | A2<B2 | x | x | 0 | 1 | 0 |
| A3=B3 | A2=B2 | A1>B1 | x | 1 | 0 | 0 |
| A3=B3 | A2=B2 | A1<B1 | x | 0 | 1 | 0 |
| A3=B3 | A2=B2 | A1=B1 | A0>B0 | 1 | 0 | 0 |
| A3=B3 | A2=B2 | A1=B1 | A0<B0 | 0 | 1 | 0 |
| A3=B3 | A2=B2 | A1=B1 | A0=B0 | 0 | 0 | 1 |

1. Boolean Expression

Supposing Z = NOT[(A ⊕ B)]

A<B : A3’.B3 + Z3.A2’.B2 + Z3.Z2.A1’.B1 + Z3.Z2.Z1.A0’.B0

A=B : Z3.Z2.Z1.Z0

A>B : A3.B3’ + Z3.A2.B2’ + Z3.Z2.A1.B1’ + Z3.Z2.Z1.A0.B0’

1. Logic Diagram



1. Software Simulation (Show here your results for each combination that gives a high output)

