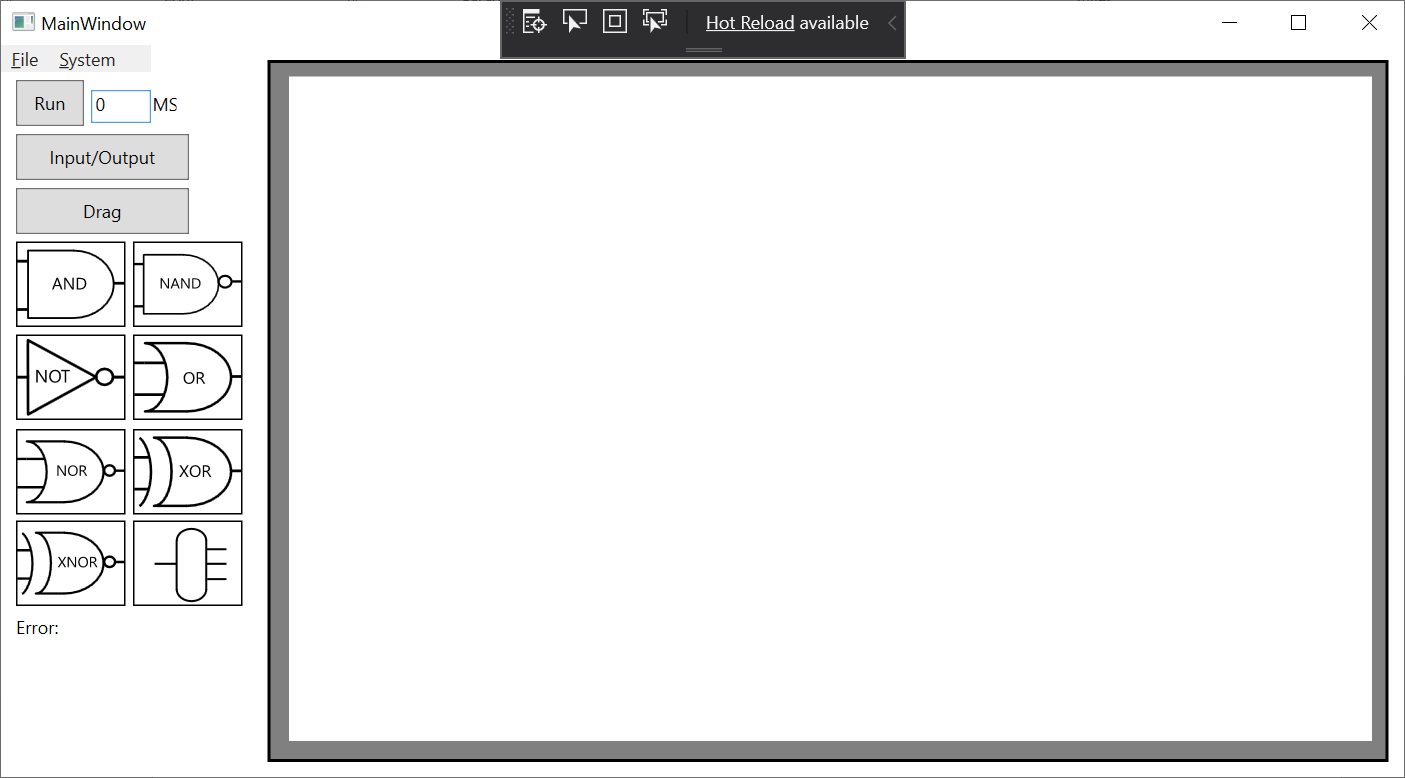
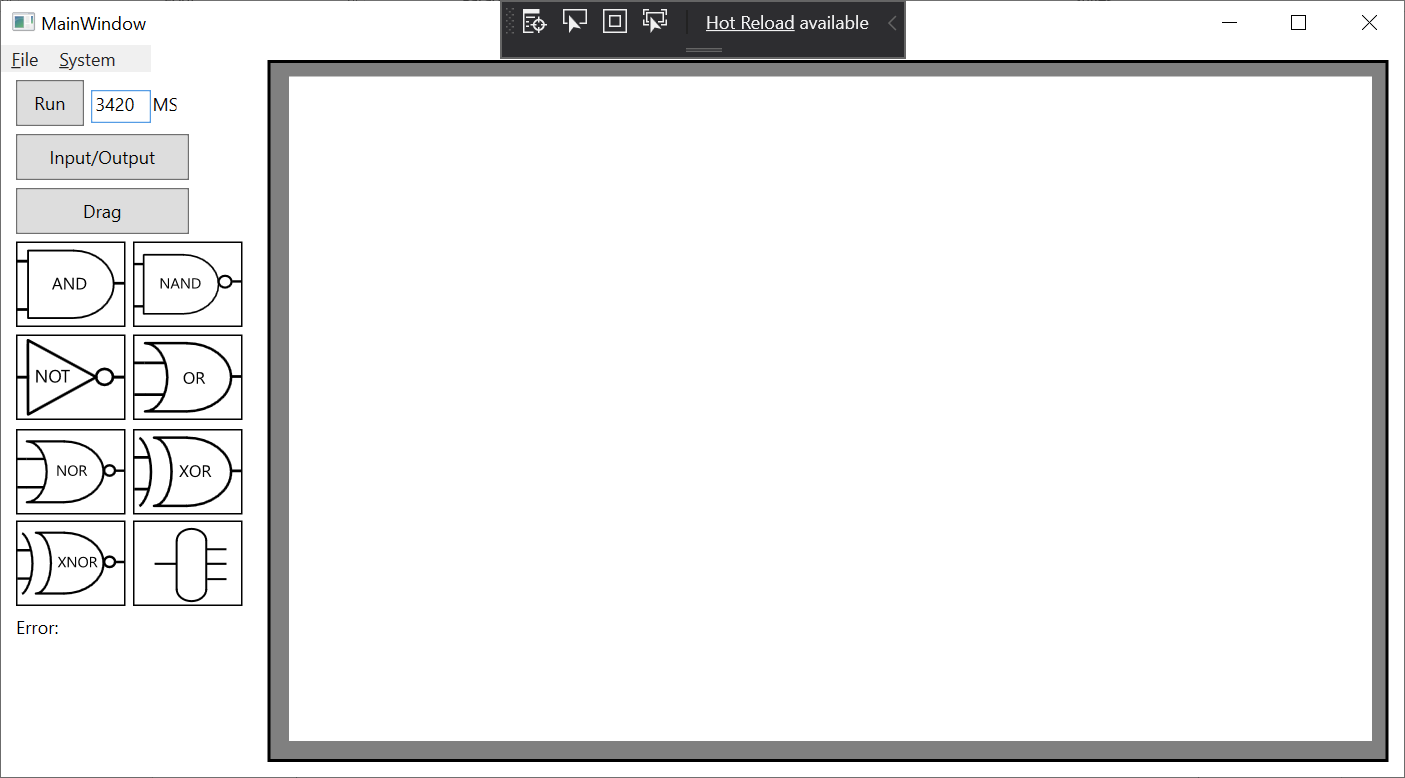
Testing

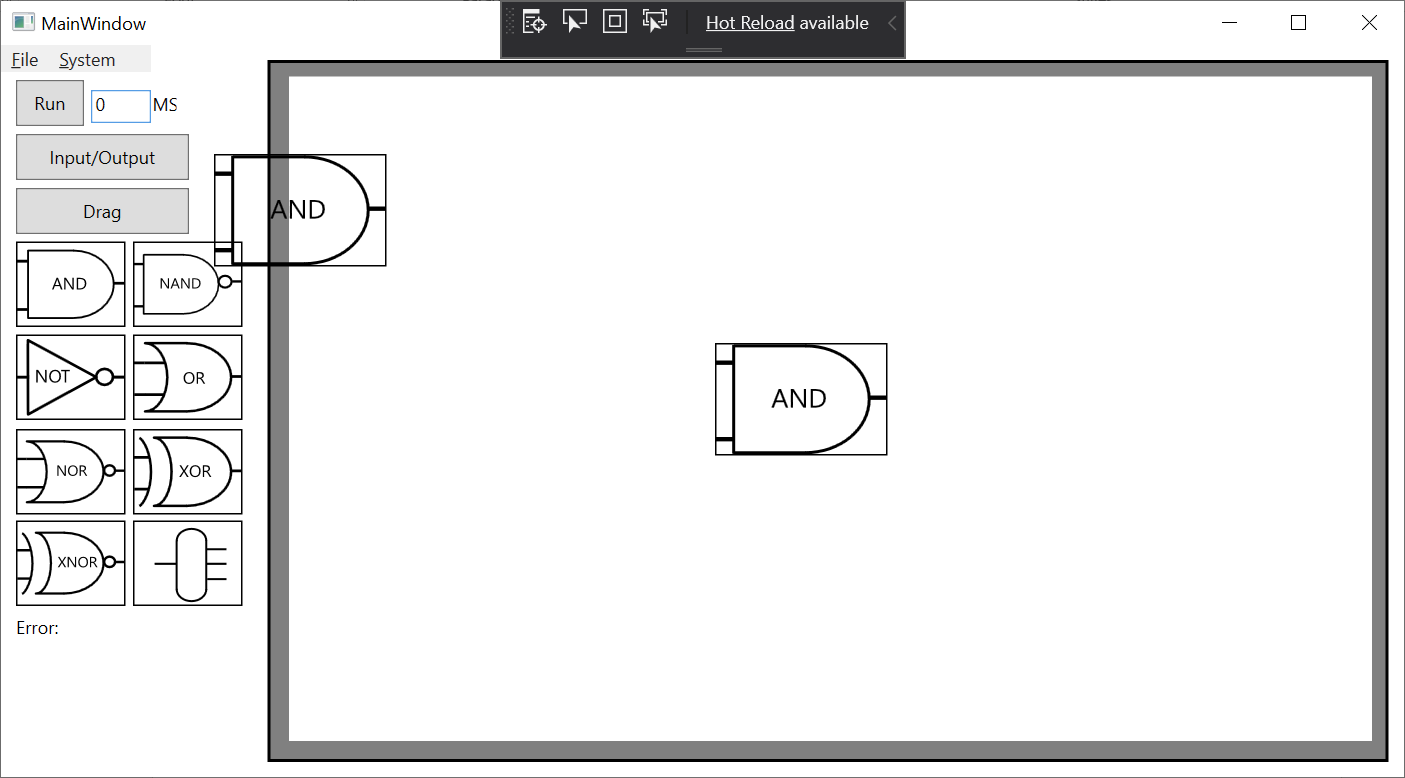
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No | Description | Expected | Material Reference | Notes |
| 1 | Simulator speed input box. Test its string input with multiple char values and see if it returns an exception error. Due to how the boxed is coded it will change to accept an invalid char but will reject it and change back to the old input. | Only integer value in the box. | 1A | The boxed is programmed in a way so that if an invalid char is inputted it will go back to the previous valid input. |
| 2 | Test the Border for adding a new gate to the canvas.  I’m going to drag and drop a rectangle in the middle of the canvas then drag and drop another off the canvas. Then going to move the canvas over to where the second rectangle would be added. | When the rectangle is added off screen it should added to the canvas. | 2A |  |
| 3 | Testing the border of the canvas to make sure it is aligned correctly. | Will remove if the cursor is not on the canvas. | Tested as close as I could to each corner going around to the 4 quadrants and see if it will do the expected result. Passed the test. | Each quadrant of the corner was top left, top right, bottom left, bottom right. |
| 4 | Testing overlap of gates. Will add 1 gate to the canvas. Will create a new one and see if it gets deleted when there is an overlap. | If any corners overlap it will get deleted. | 3A | Did it for each corner(bottom right with top left, top right with bottom left, top left with bottom right, bottom left with top right) |
| 5 | Gate Calculation | Each gate will have the correct output for their gate type | 4A |  |
| 6A | Lines with input and output. Testing to see if that they are correctly added and that line is added to port 2 instead of port 1. | The line output port should be 1 and input port is 1. When adding the input and output buttons they should not overlap the line and for the different types of gate should change their set up. | 5A | For the transformer with 3 output they overlap but not to the level where it’s restricted. |
| 6B | The lines should be able to be removed without effecting the gates. | Just line is removed and the input buttons are in tack. | 5B |  |
| 6C | When all inputs are full(lines or input buttons) | You shouldn’t be able to add another line. | The line is removed. |  |
| 6D | When all outputs are full | The line shouldn’t even be created in the first place | The line doesn’t even get added to the canvas |  |
| 6E | When only parts of the inputs are full | If the port is full it should be added to the next available port. | 5C |  |
| 6F | When the gates are moved | All Inputs outputs and lines should track the movement of the gate when in movement and when placed down. | 5D  Tracks gate when moving. | The label for the line (which is 0) should also move directly in the middle of the line which it does. |
| 7 | When a gate is already placed and is dragged over another rectangle. | It should return to the old position of the rectangle and so should it’s input and output. | 6A |  |
| 8 | System clean up | Any Gates, Lines, input and output classes should be removed if they’re not being used anymore | 7A | Found a bug in the code when testing a partially hard system clean up where I was removing the object before decreasing all the other variables by 1. This worked for all values in the list which weren’t directly above it. So it’s not imidcally noticeable but if you keep making changes everything becomes out of line and will end in an out of range error. |

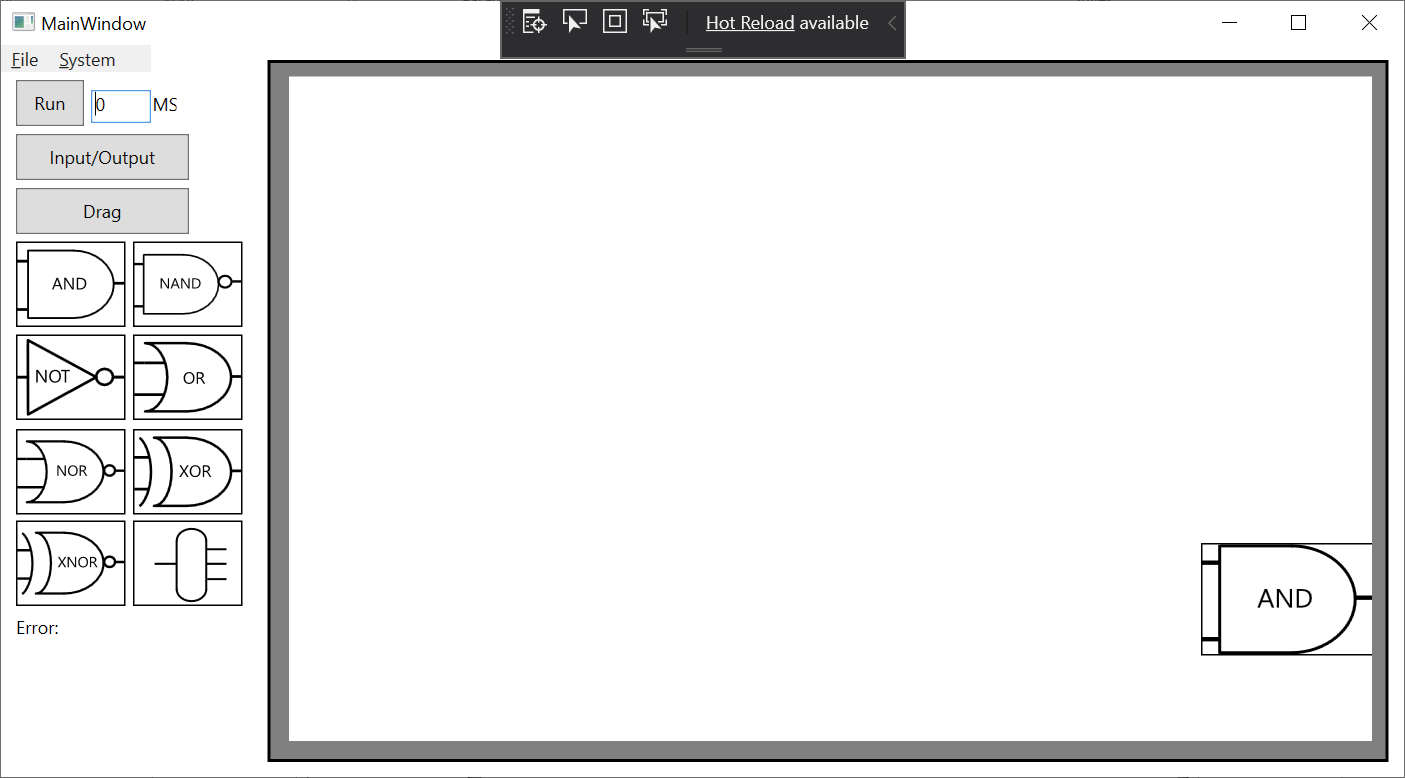
Reference

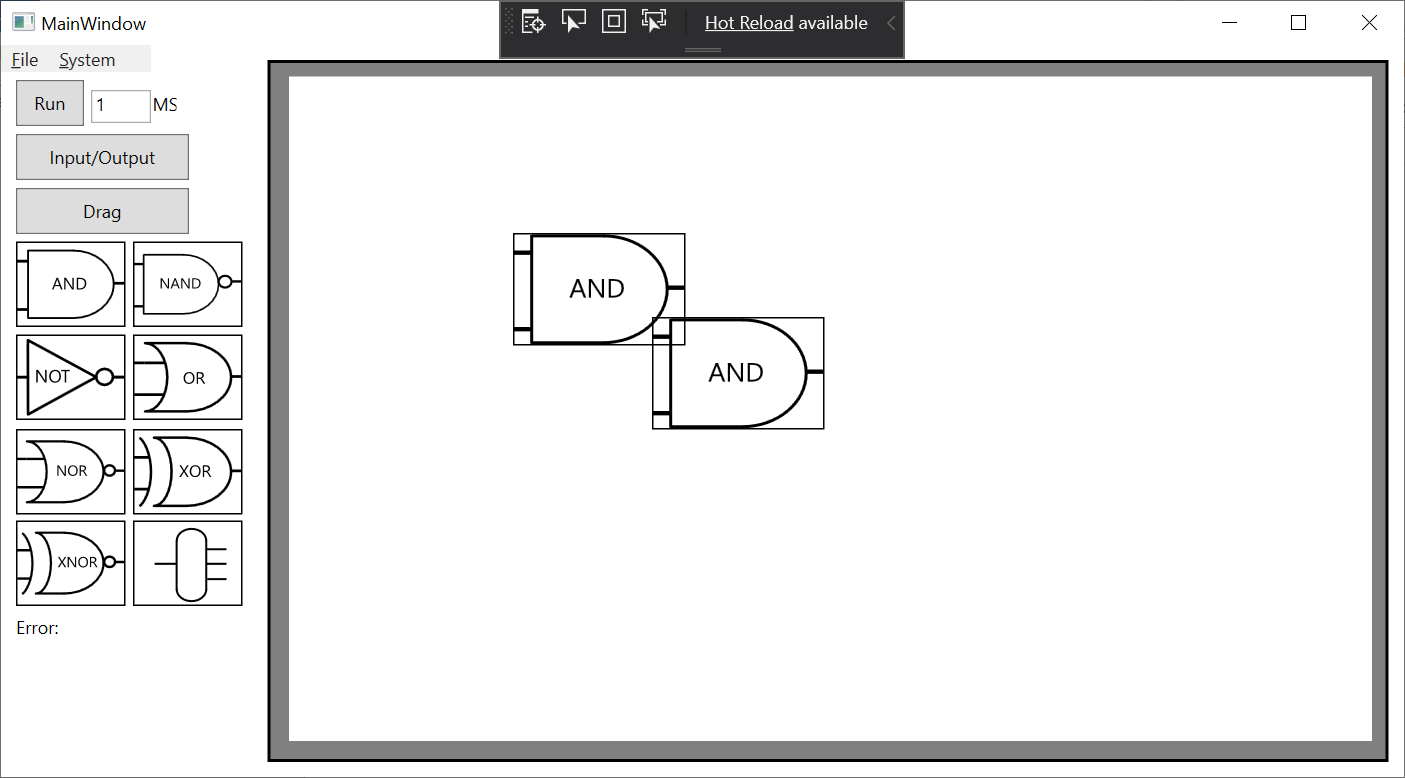
1A. 

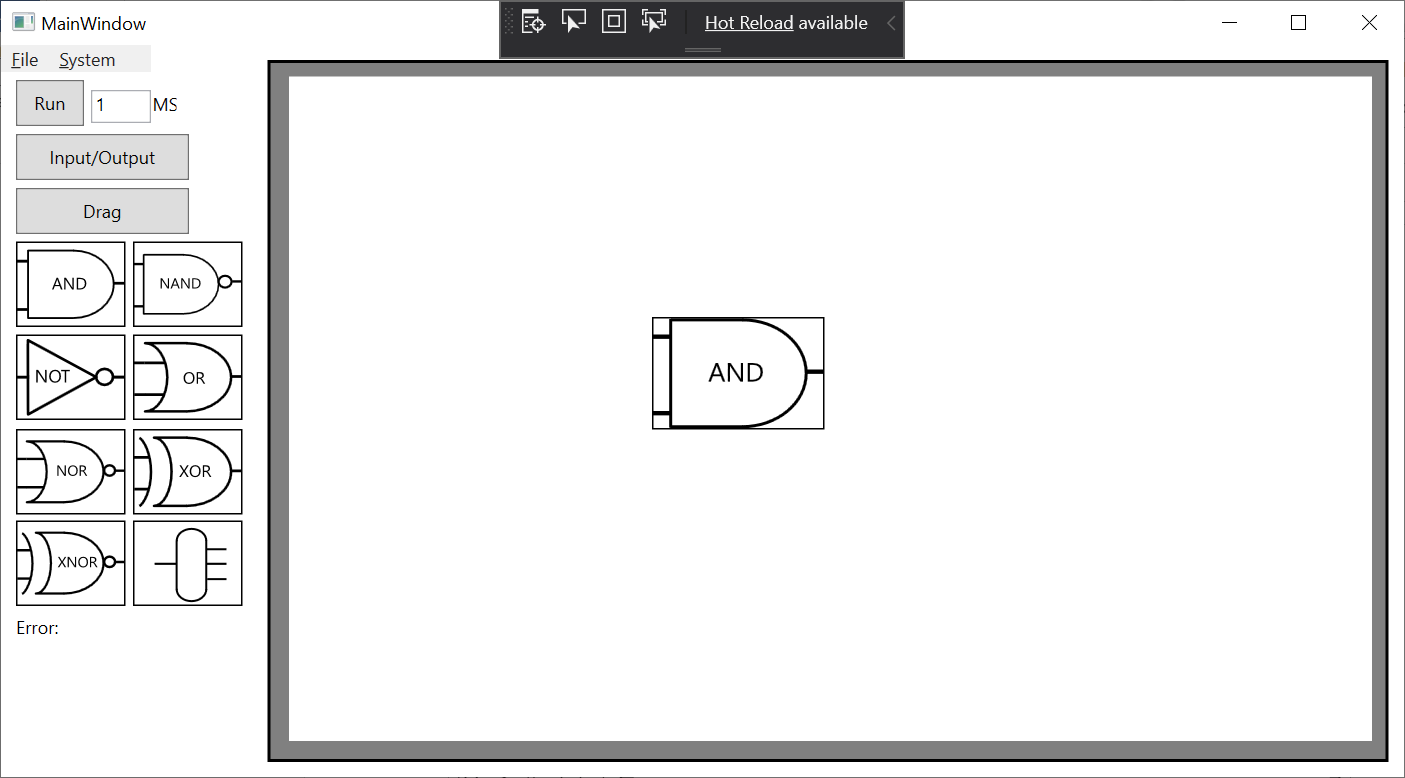


2A.

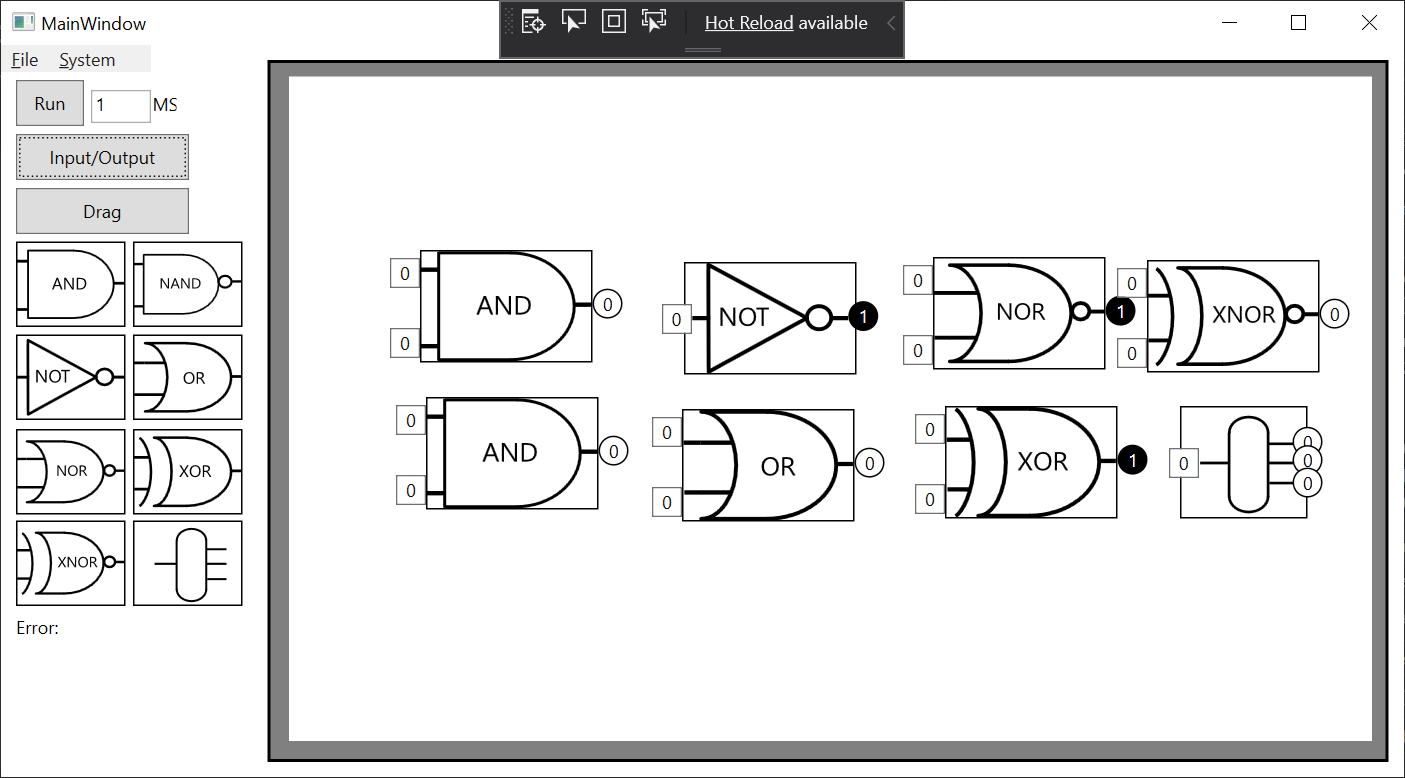


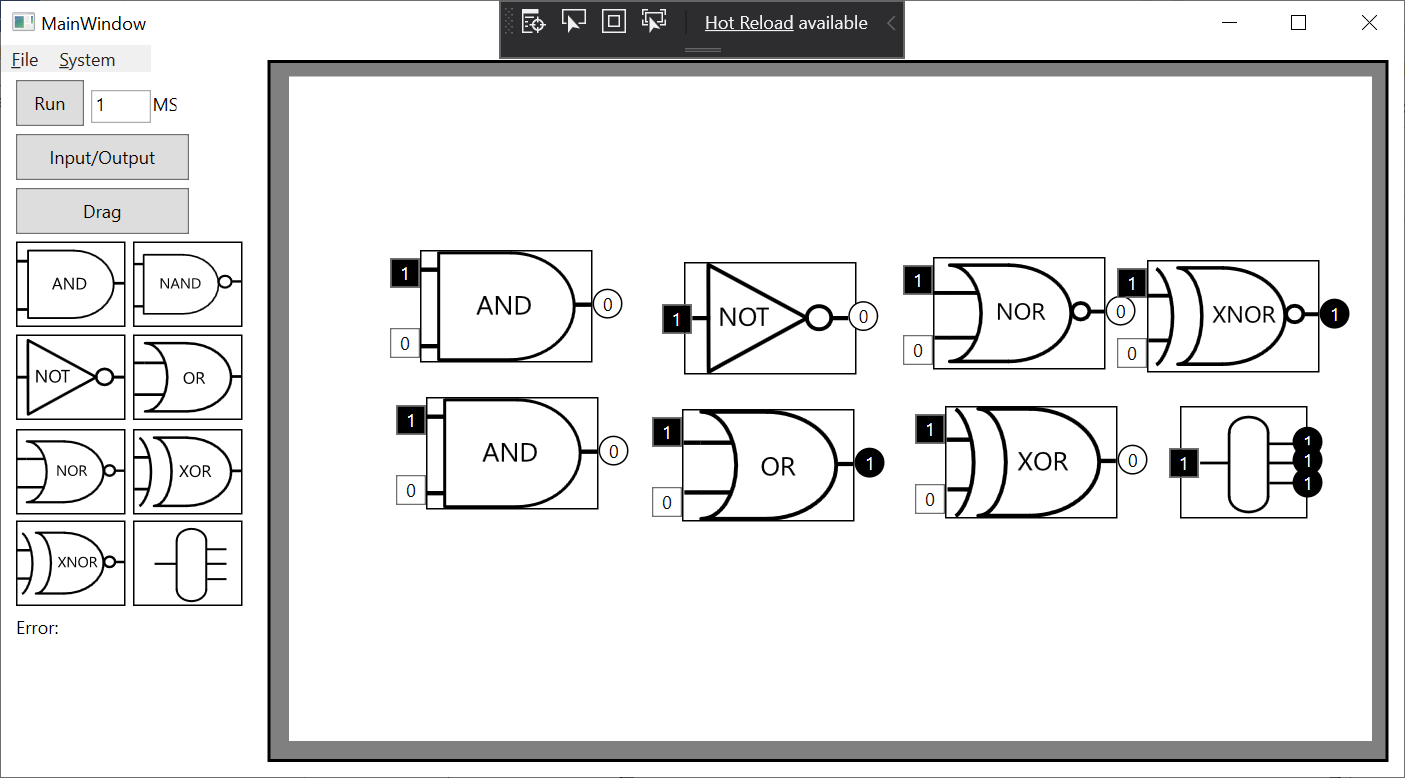


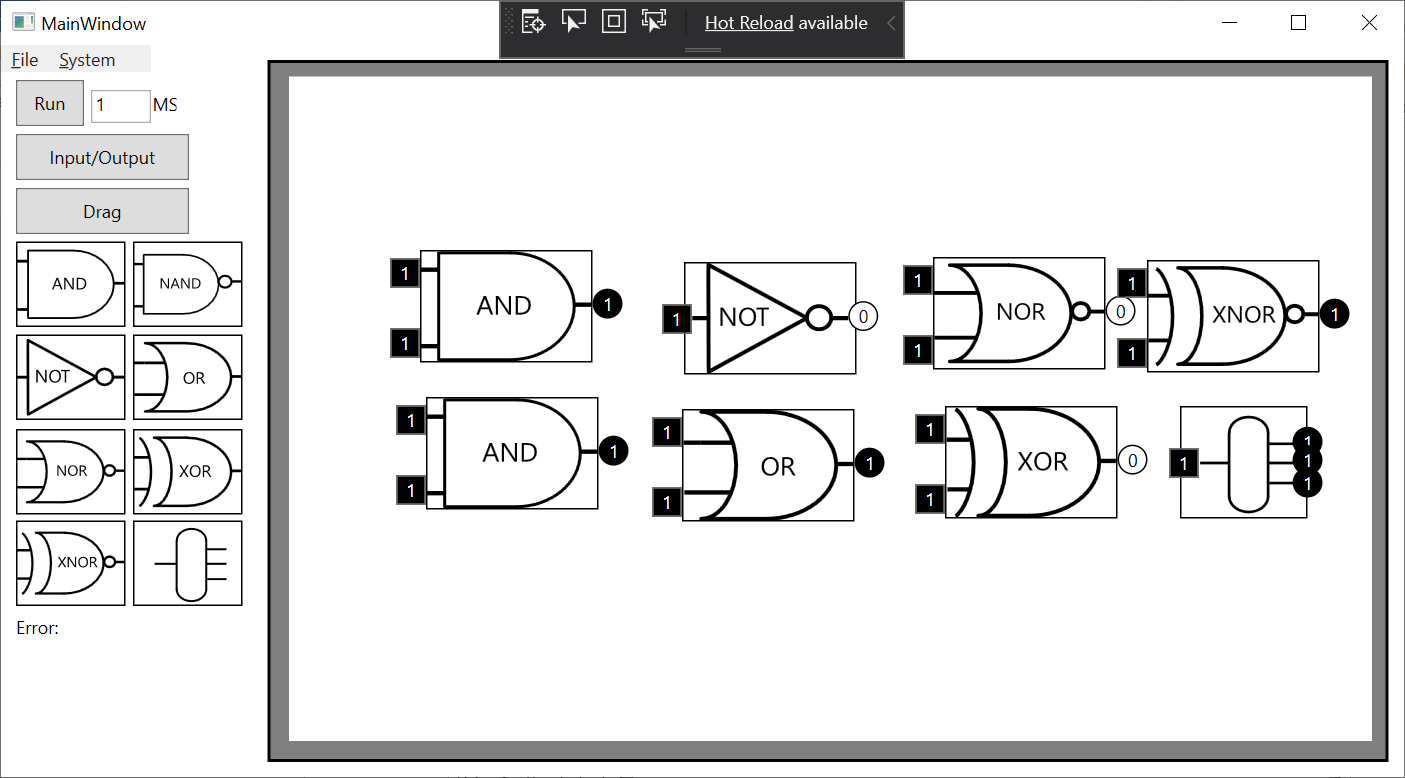
3A. 

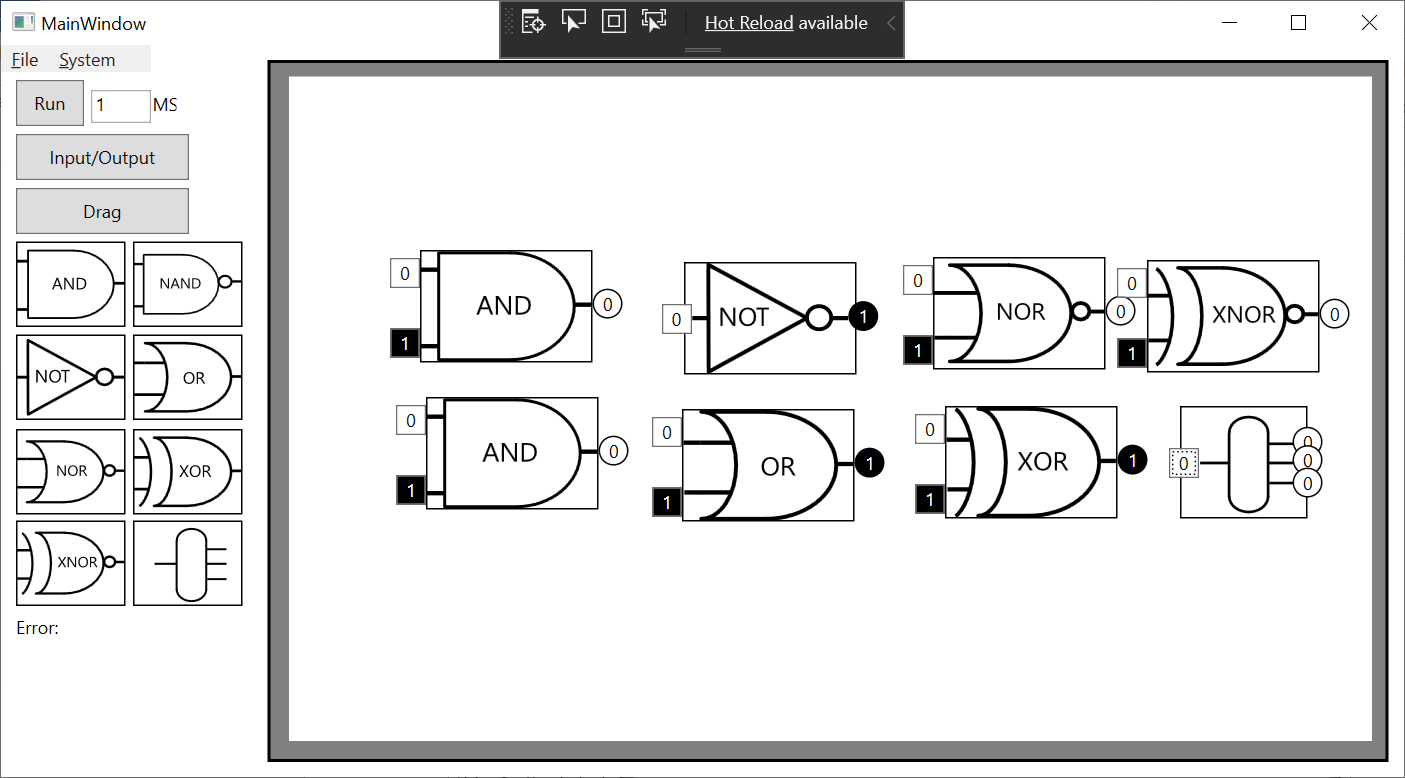


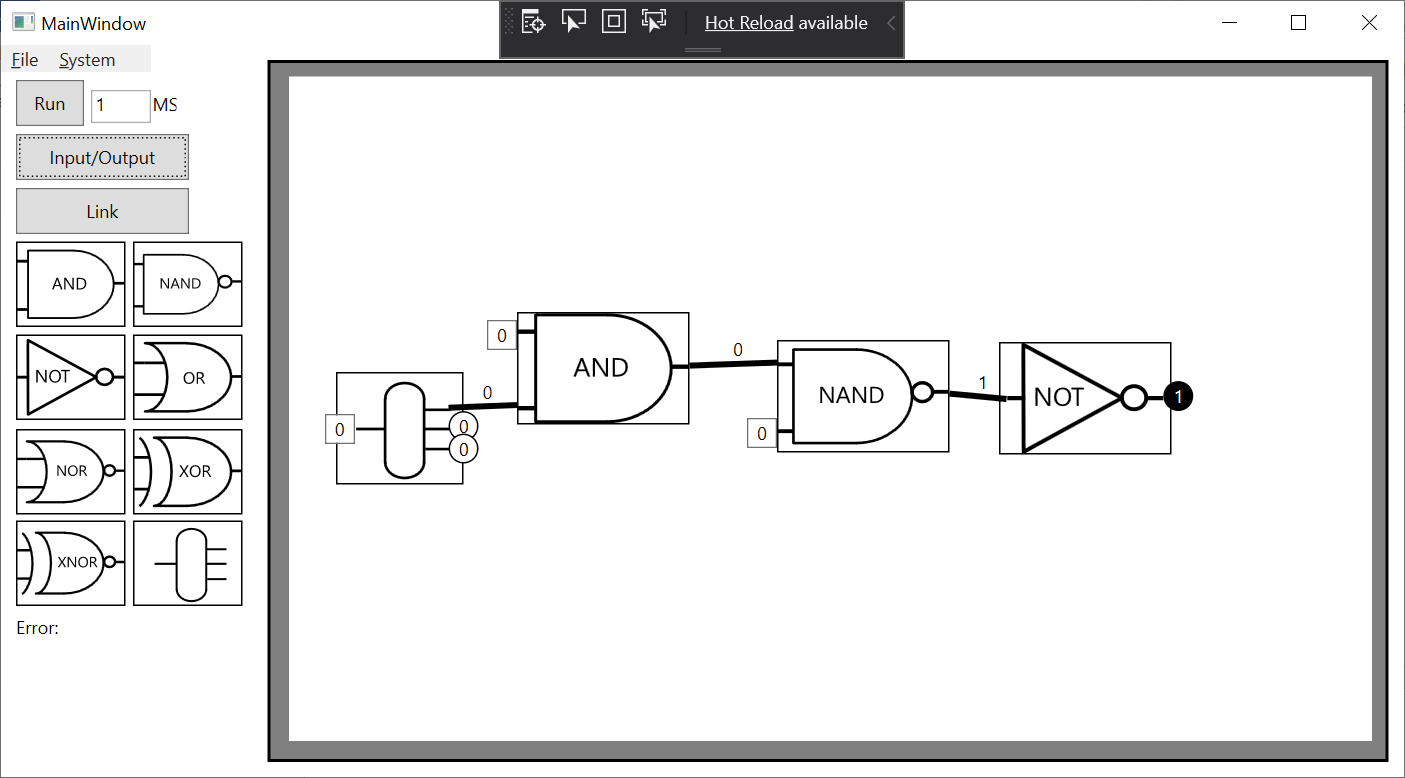
4A.



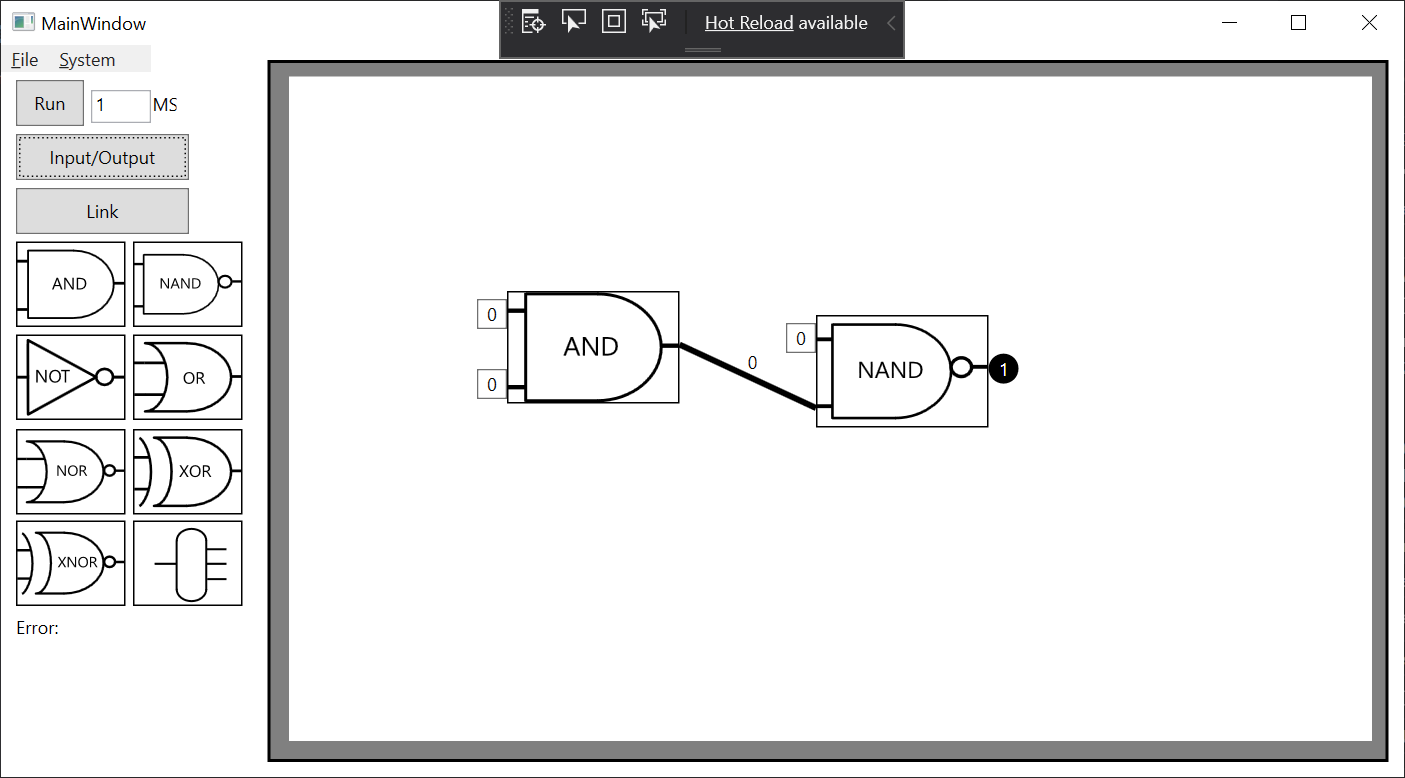


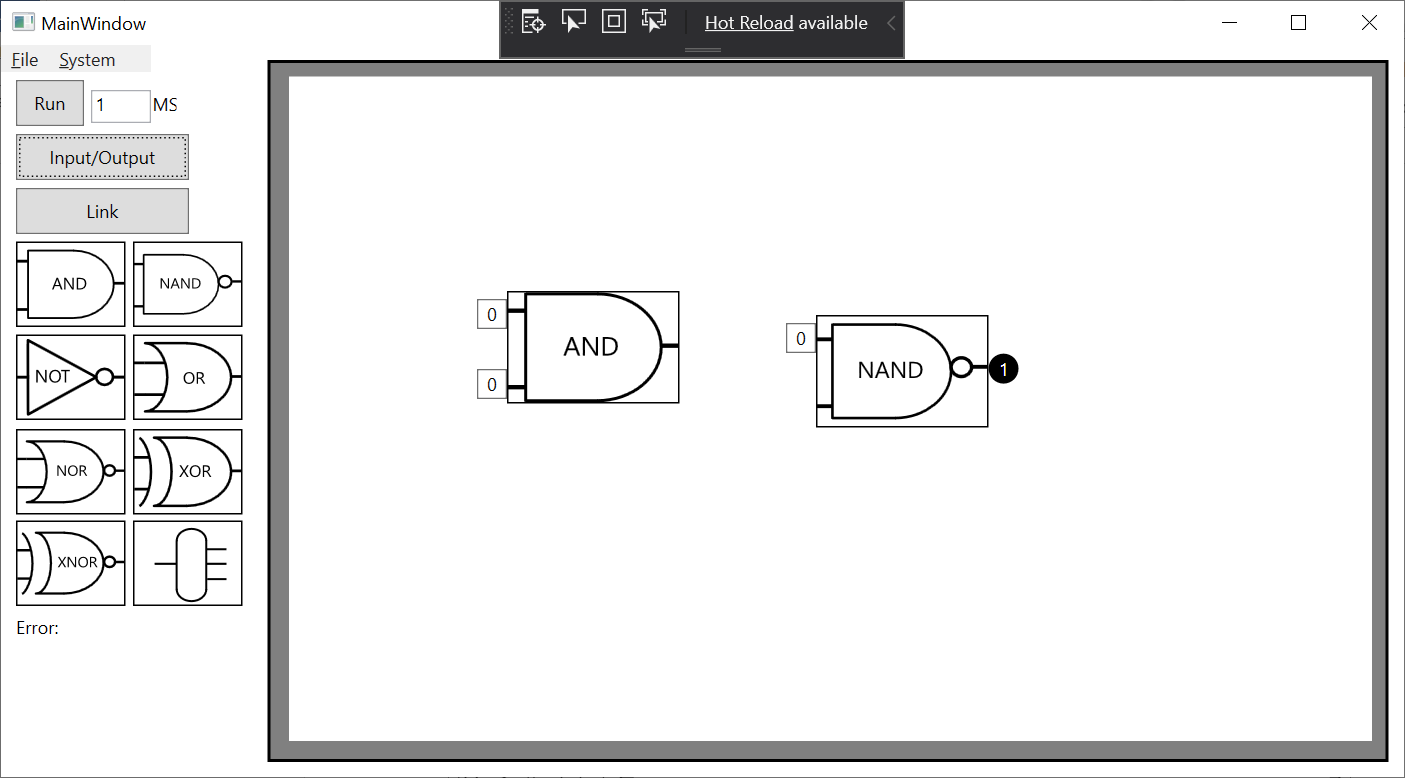




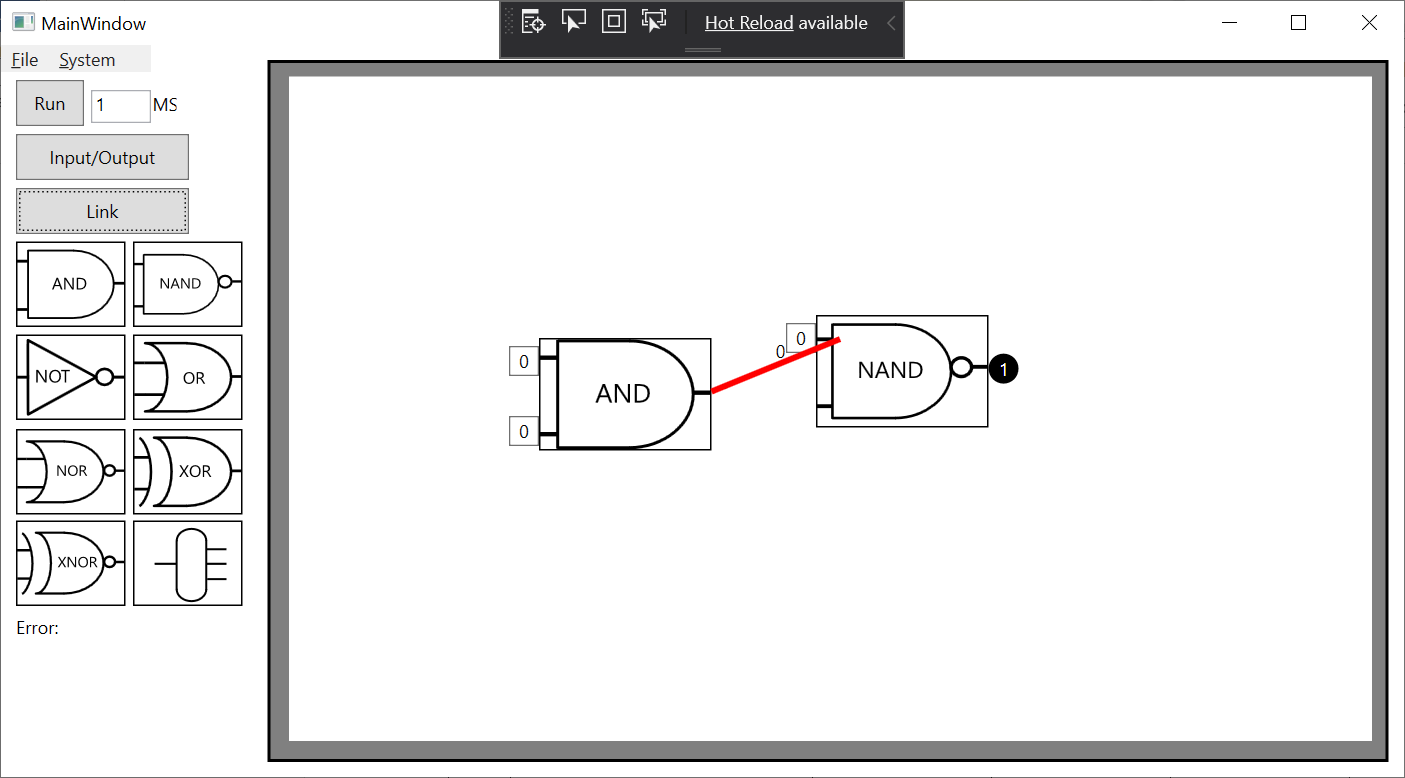
5A. 

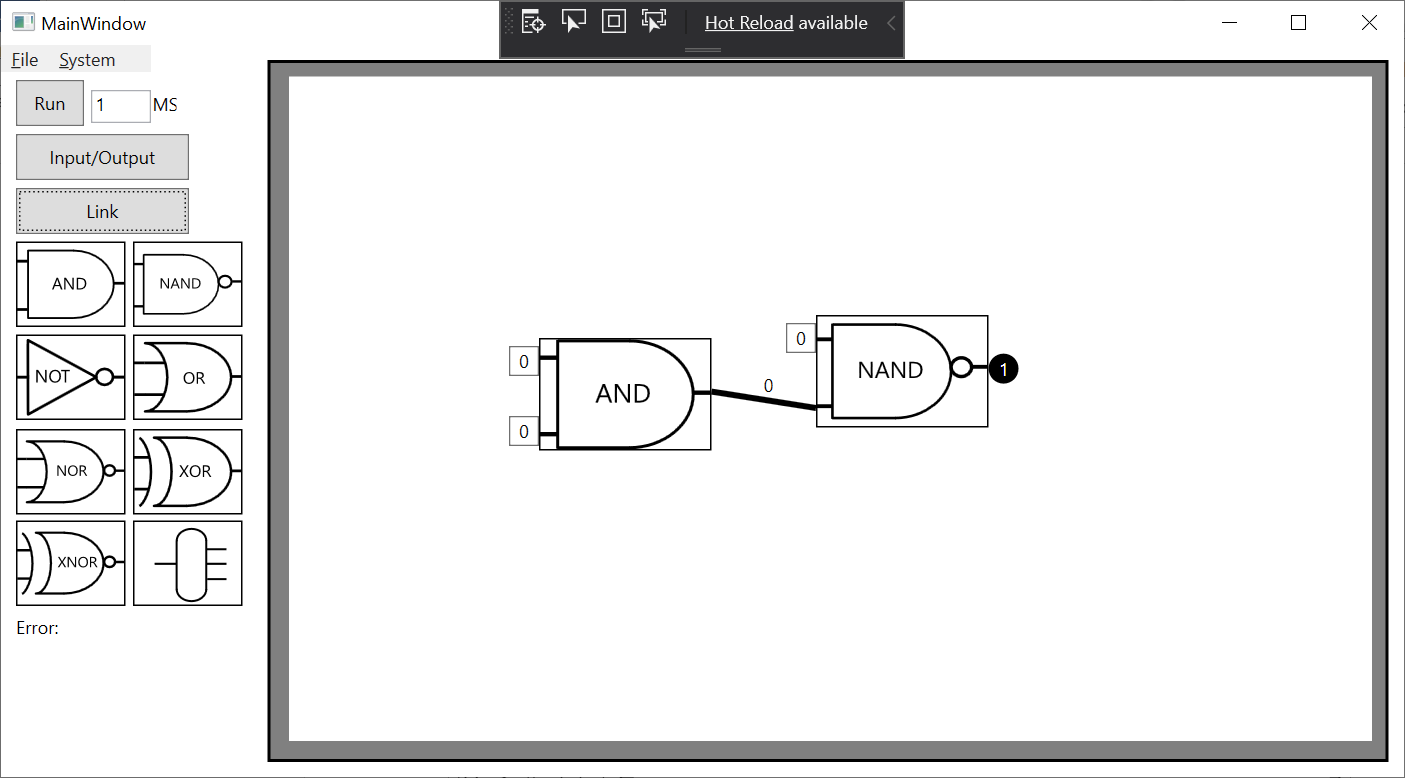
5B.



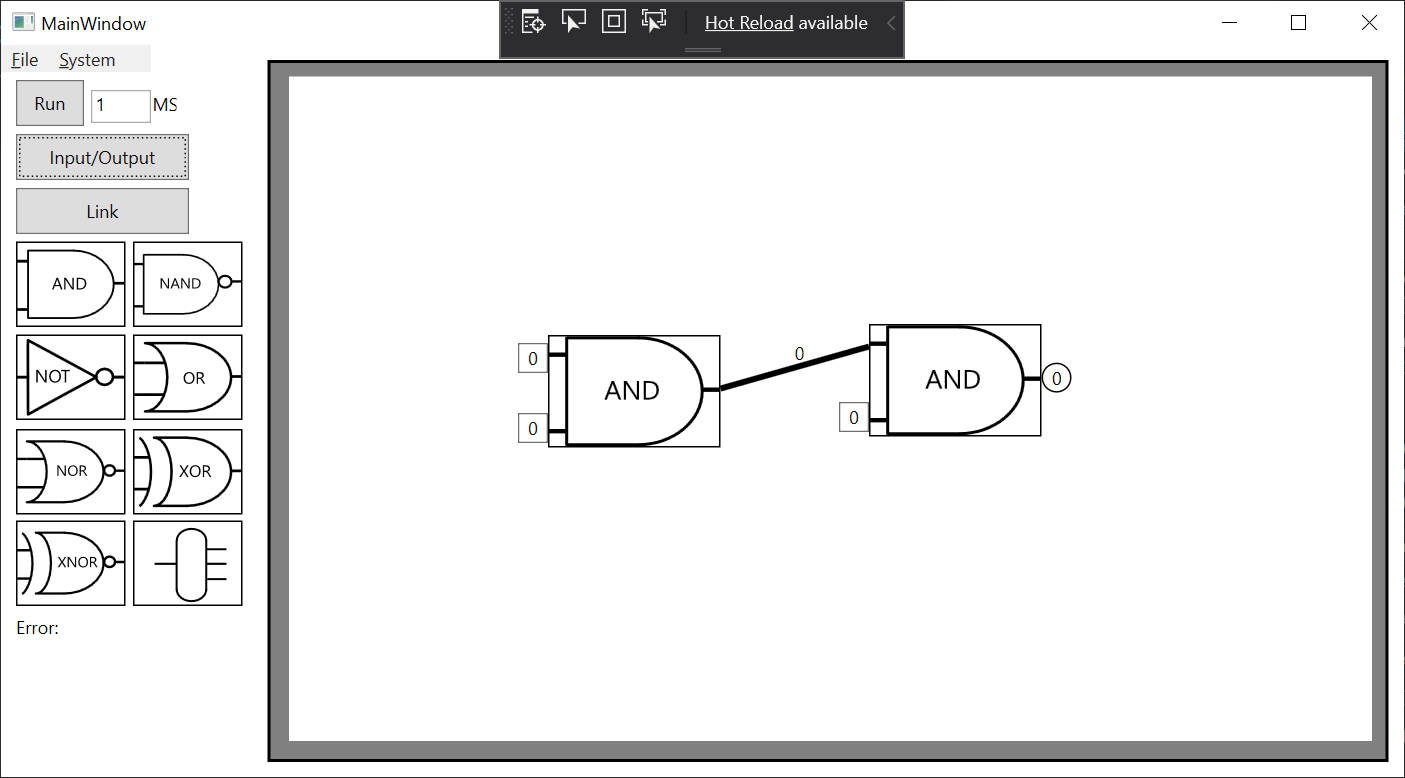


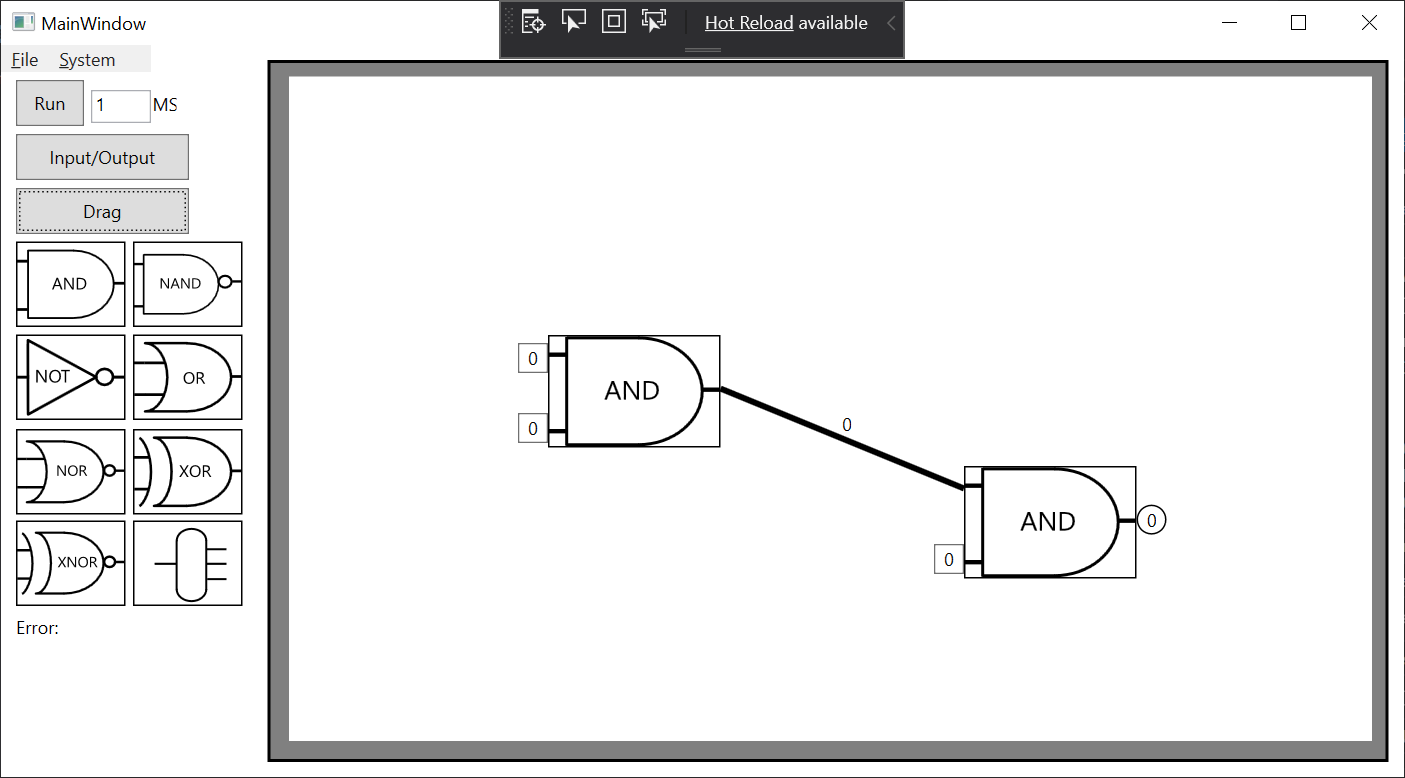
5C.



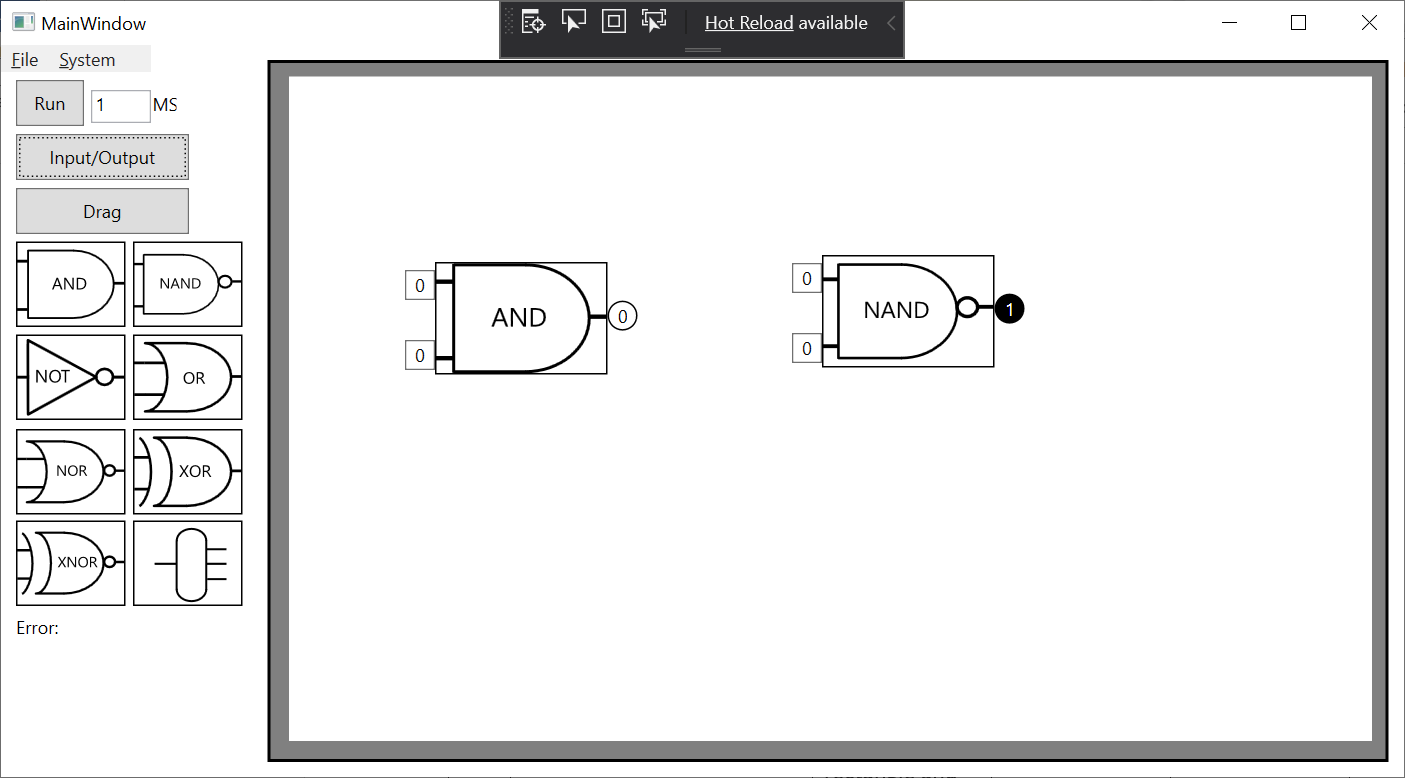


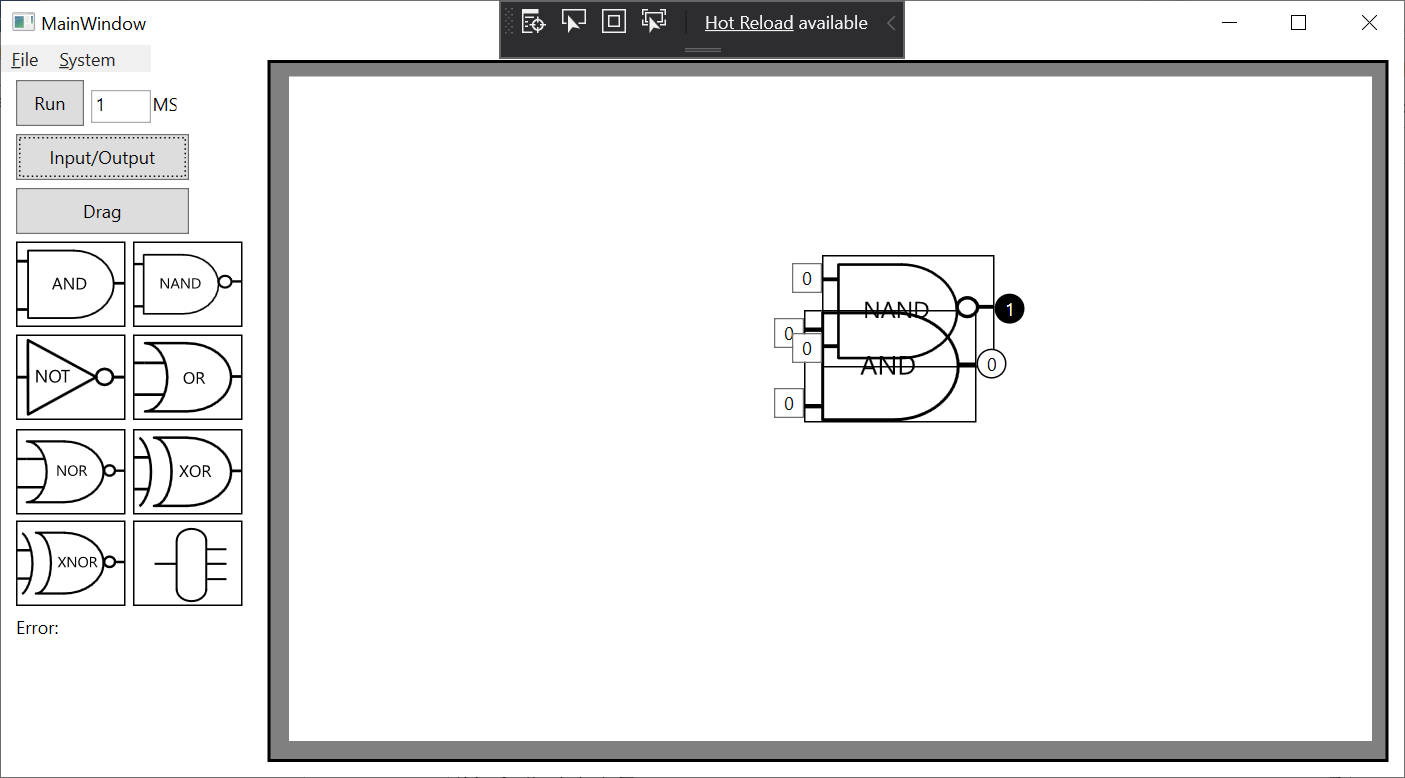
5D.

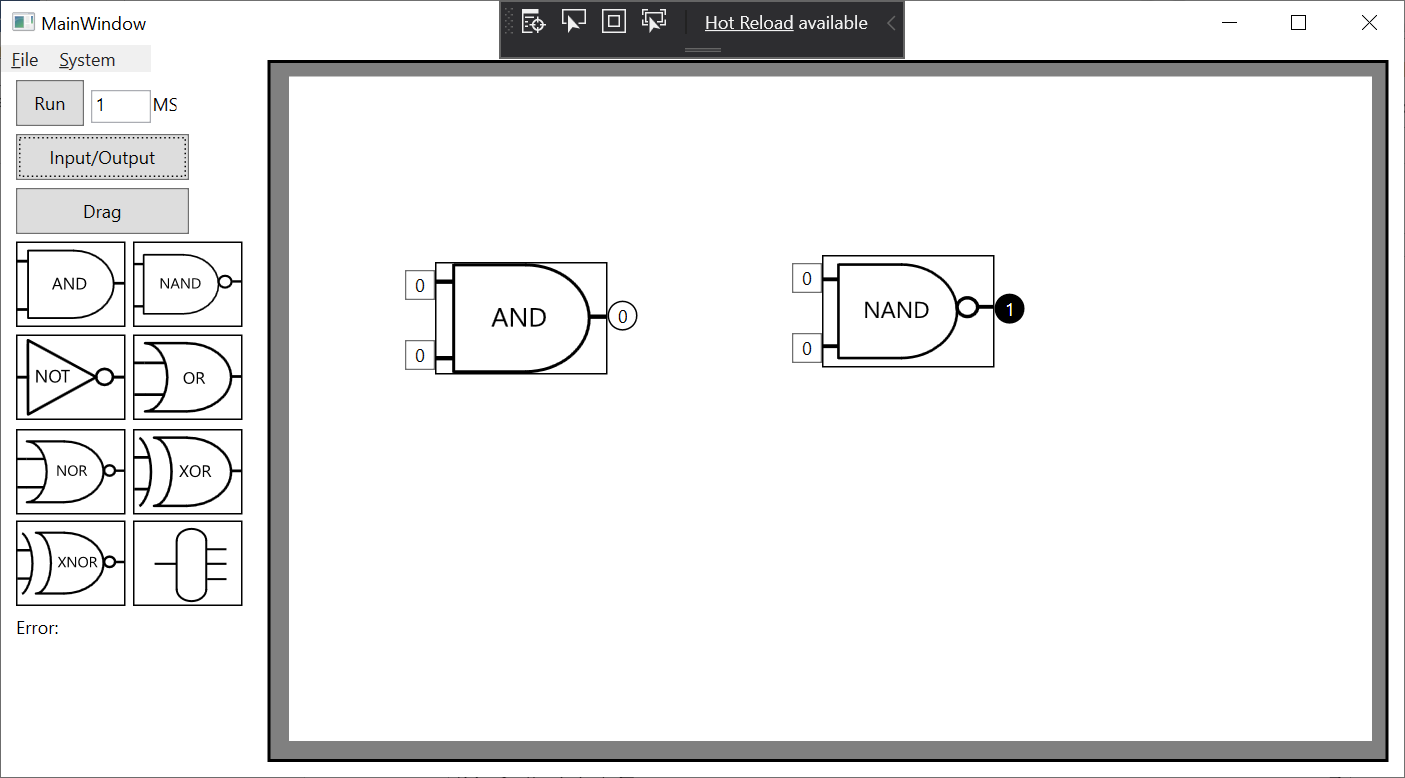




6A.







7A.