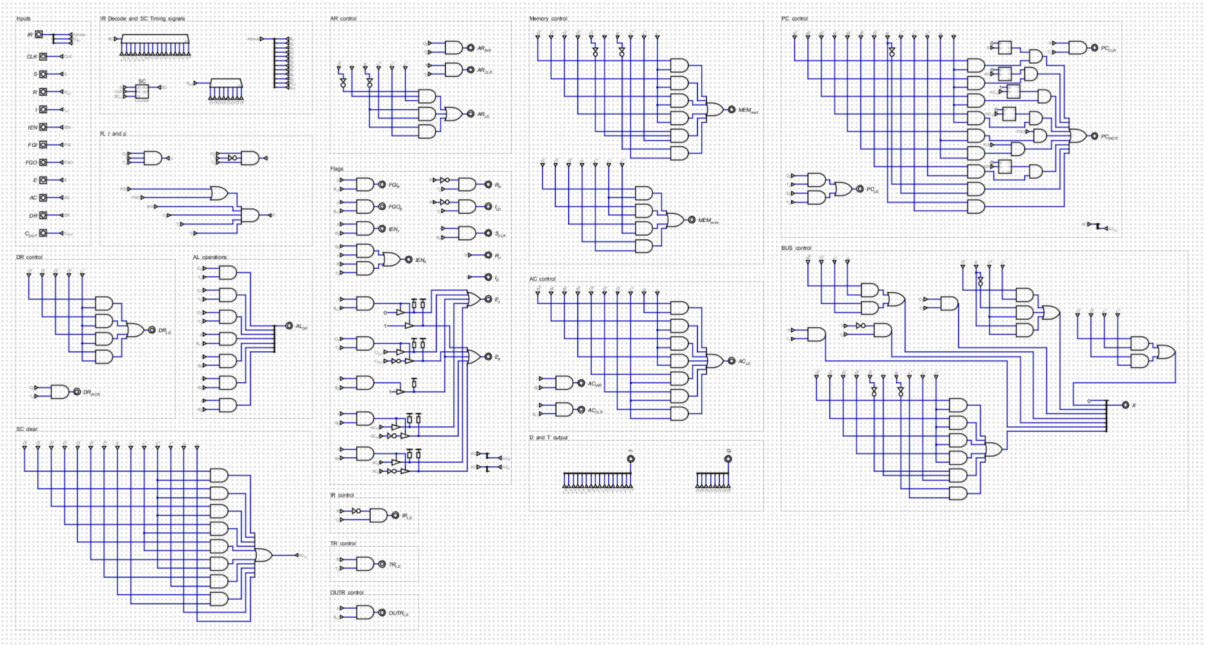
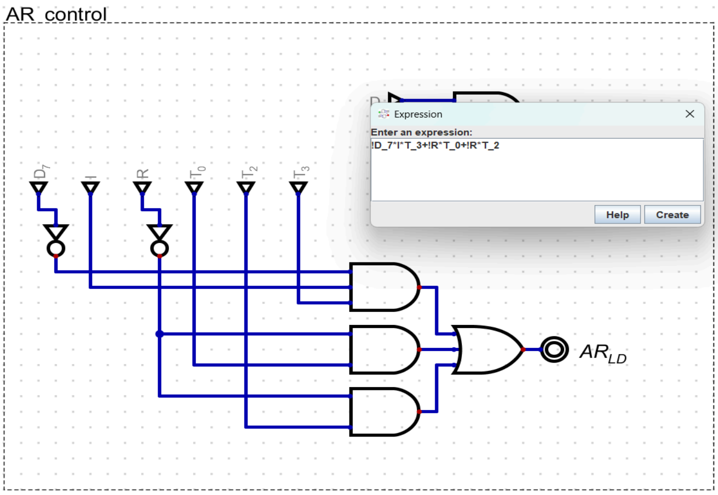
Design Assumptions and Design Approach



- Inputs are on the left top side, every input connected to a tunnel.
- Decoding operations and sequence counter is right next to inputs.
- Below that there are R, r and p being calculated.
- DR control, AR control, Memory control, AC control, IR control, TR control, OUTR control, SC clear are designed by the expression button simply entering the logic expressions.

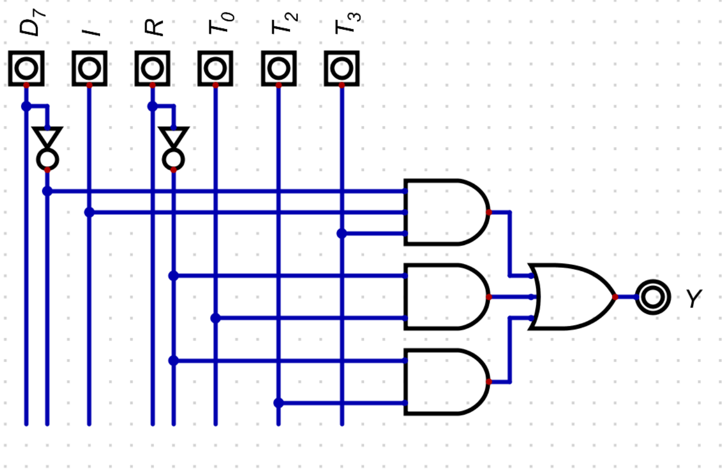
This is the logical expression to obtain AR_LD.



Enter an expression:

```
!D_7*I*T_3+!R*T_0+!R*T_2
```

And this is the output. I simply changed inputs with tunnels and the output name and deleted the unnecessary wires.



- AL operations are concerned with which operation is being executed in the accumulator.

- The outputs for D and T are simply the reverse decoding of D and T.

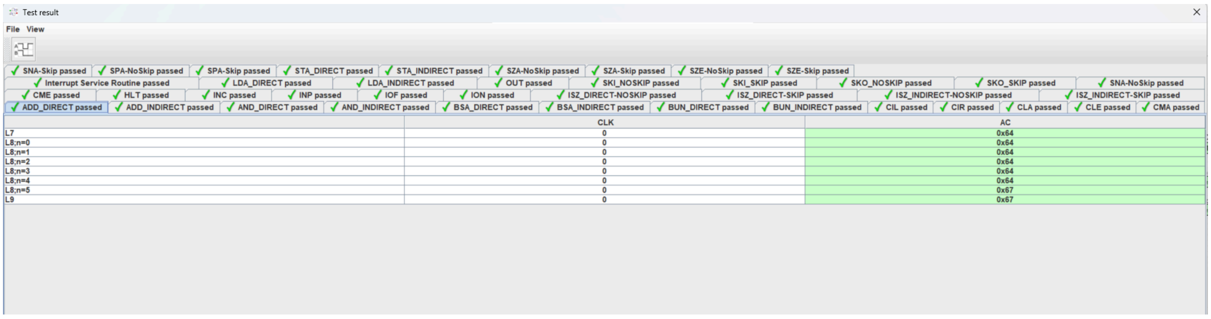
- Flags are just the same but E_K and E_J were challenging due to issues with the three-state buffer.

- PC control is similar to others; however, in some cases, it includes an 'if' statement, so it requires specific control. (I initially tried using a three-state buffer, but encountered the same issue with the E flag and resolved it using a comparator and AND gates. This was before I learned about using pull-down resistors. With three-state buffers and pull-down resistors, implementation should be easier, but I didn't modify my initial implementation after discovering pull-down resistors.)

- Bus control checks when a register accesses the bus and sends a signal to that wire.

Summary of Results

The control unit successfully passes all the tests and it successfully executes all the instructions.



✓ SNA-Skip passed ✓ SPA-NoSkip passed ✓ SPA-Skip passed ✓ STA_DIRECT passed ✓ STA_INDIRECT passed ✓ SZA-NoSkip passed ✓ SZA-Skip passed ✓ SZE-NoSkip passed ✓ SZE-Skip passed

✓ Interrupt Service Routine passed ✓ LDA_DIRECT passed ✓ LDA_INDIRECT passed ✓ OUT passed ✓ SKI_NOSKIP passed ✓ SKI_SKIP passed ✓ SKO_NOSKIP passed ✓ SKO_SKIP passed ✓ SNA-NoSkip passed

✓ CME passed ✓ HLT passed ✓ INC passed ✓ INP passed ✓ IOF passed ✓ ION passed ✓ ISZ_DIRECT-NOSKIP passed ✓ ISZ_DIRECT-SKIP passed ✓ ISZ_INDIRECT-NOSKIP passed ✓ ISZ_INDIRECT-SKIP passed

✓ ADD_DIRECT passed ✓ ADD_INDIRECT passed ✓ AND_DIRECT passed ✓ AND_INDIRECT passed ✓ BSA_DIRECT passed ✓ BSA_INDIRECT passed ✓ BUN_DIRECT passed ✓ BUN_INDIRECT passed ✓ CIL passed ✓ CIR passed ✓ CLA passed ✓ CLE passed ✓ CMA passed

	CLK	AC
L7	0	0x64
L8;n=0	0	0x64
L8;n=1	0	0x64
L8;n=2	0	0x64
L8;n=3	0	0x64
L8;n=4	0	0x64
L8;n=5	0	0x67
L9	0	0x67

Lessons Learned

1. I got used to making tunnels that make building circuits way easier and make them look way better and less complicated.
2. I learned the difference between decoding and splitting the bits of an input that has multiple bits.
3. I learned how to use the expression button that makes logic gates according to the expression you give. (Thanks to my friend Çağlar Kabaca for suggesting this.)
4. I learned how to build control bits.
5. I learned that I need to use a pull-down resistor when the output of the three-state buffers goes into a logic gate. If I don't use it, it kind of gets stuck on that input even if the selector bit is 0.
6. I learned how to change the output order of an included custom circuit and how to change its size.
7. I overall learned how to design a basic computer, how the instructions work, how we decode them, how we control the bus, how we do the timings and logical operations, how to use test cases and debug our design.
8. Most importantly, I literally memorized how a JK flip flop works after hours of tinkering with the E flip flop and making truth tables.