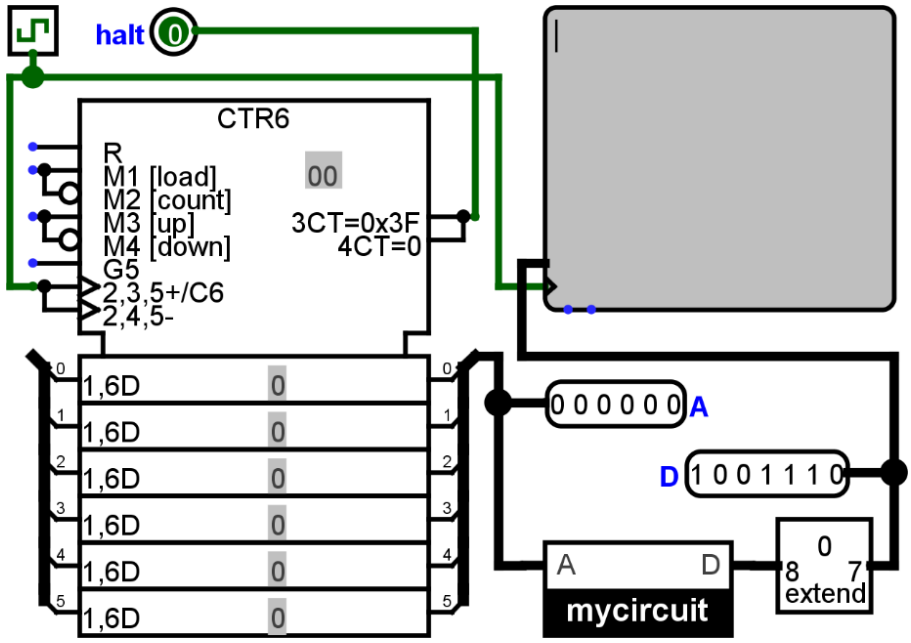
Assignment #3 for 104418019 - ELEC3270: Microprocessors



For each question below, build a sub-circuit ("mycircuit") which implements a 64x8 ROM using the smaller ROMs (found in the Memory library) listed. Each ROM entry show its label (be sure to label each ROM correctly), the address range in the complete ROM (in hexadecimal), and its contents. You can copy these values directly into Logisim one row at a time using copy/paste. You will need to "decode" the input address signal to produce control signals that will enable one three-state buffer ("Controlled Buffer" in the Gates library; set "Data Bits" to 8 to simplify the circuit), and disable all others. These buffers will drive the output bus of the sub-circuit. Use the test design (shown above) to implement your main circuit; it doesn't need to look the same, but it must be connected the same; **don't forget the output pins**. When you run the simulation, you will see a memorable quote appear in the "TTY" (found in the Input/Output library) window over and over again. The "bit extender" can be found in the Wiring library. **When you are finished, ZIP all of your designs into "SID_3" (where SID is your student number), and upload it to Blackboard.**

1. Name this design "SID_3a".

```
ROM0 20-2f = 72 65 73 74 20 6f 66 20 74 68 65 20 77 6f 72 6c
ROM1 00-0f = 41 20 72 65 61 6c 20 66 72 69 65 6e 64 20 77 61
ROM2 30-3f = 64 20 77 61 6c 6b 73 20 6f 75 74 2e 0a 00 00 00
ROM3 10-1f = 6c 6b 73 20 69 6e 20 77 68 65 6e 20 74 68 65 20
```

"A real friend walks in when the rest of the world walks out."

2. Name this design "SID_3b".

```
ROM0 10-17 = 61 67 72 65 65 20 77 69     ROM1 20-27 = 61 6c 77 61 79 73 20 66
ROM2 30-37 = 73 74 20 62 65 20 77 72     ROM3 00-07 = 57 68 65 6e 65 76 65 72
ROM4 28-2f = 65 65 6c 20 49 20 6d 75     ROM5 18-1f = 74 68 20 6d 65 20 49 20
ROM6 38-3f = 6f 6e 67 2e 0a 00 00 00     ROM7 08-0f = 20 70 65 6f 70 6c 65 20
```

"Whenever people agree with me I always feel I must be wrong."

3. The ROMs here also specify which bits they provide for each byte. You therefore need to combine the nybbles from two ROMs to create a single byte. Name this design "SID_3c".

```
ROM0 00-07 (7-4) = 5 6 6 2 7 7 6 7       ROM1 08-0f (7-4) = 2 6 7 7 7 6 6 6
ROM2 18-1f (7-4) = 2 4 7 7 7 6 6 6       ROM3 10-17 (7-4) = 6 7 2 6 7 7 6 7
ROM4 38-3f (3-0) = 9 1 e 3 f 9 a 0       ROM5 18-1f (3-0) = 0 3 5 3 4 f 4 5
ROM6 38-3f (7-4) = 6 6 6 7 3 2 0 0       ROM7 08-0f (3-0) = 0 3 5 3 4 f 4 9
ROM8 28-2f (7-4) = 6 7 6 7 6 7 2 7       ROM9 00-07 (3-0) = 3 5 4 0 1 5 9 3
ROMA 20-27 (3-0) = 3 f 0 8 7 8 f 0       ROMB 30-37 (7-4) = 6 6 2 4 7 6 7 6
ROMC 28-2f (3-0) = 7 5 1 2 4 3 0 4       ROMD 10-17 (3-0) = 5 4 0 9 0 3 f 3
ROME 30-37 (3-0) = 8 5 0 7 5 1 2 4       ROMF 20-27 (7-4) = 7 3 2 2 5 6 6 2
```

"Sed quis custodiet ipsos Custodes? (Who guards the Guardians?)"