Jetic Gū

Columbia College

Please remember to write your name and student number.

You must complete the following assignment and submit a PDF of relevant questions. Handwritten submissions and proprietary formats (e.g. Pages or MS Word) will not be accepted. You will also need to upload LogicWork circuit design file. Then upload a single ZIP file to Moodle.

Submission File structure:

```
submission.zip
    — circuit1-1.cct
    — circuit1-2.cct
    — circuit1-3.cct
    — circuit2-1.cct
    — circuit2-2.cct
    — circuit2-3.cct
    — circuit3.cct
    — circuit4.cct
    — circuit5.cct
    — circuit6.cct
    — lib.clf
```

All circuit files are 1pt each.

Lab 2

1. Save the library and circuit files we created in class containing the following designs in the final ZIP file:
    1. 3-to-8 Decoder implemented using the 2-to-4 Decoder (`circuit1-1.cct`);
    2. 4-to-2 priority encoder with validity bit (`circuit1-2.cct`);
    3. 4 channel 4bit Multiplexer implemented using the 4 channel 1bit Multiplexers (`circuit1-3.cct`);
    4. Include the above designs in your library file (`lib.clt`), I must be able to use these components in your library file, or 50% of the points will be lost.
2. Save the library and circuit files we created in class containing the following designs in the final ZIP file:
    1. 4-bit binary adder (`circuit2-1.cct`);
    2. 4-bit binary adder-subtractor with XOR and Adder (`circuit2-2.cct`);
    3. 4-bit binary plus 1 incrementer (`circuit2-3.cct`);
    4. Include the above designs in your library file (`lib.clt`), I must be able to use these components in your library file, or 50% of the points will be lost.

3. Implement the following Boolean function with an 1bit 4-to-1 multiplexer and optionally, a single inverter:

   $$F(A, B, C) = \Sigma m(2,4,6)$$

   You must ONLY use value-fixing and the components stated above. Save the circuit as `circuit3.cct` .

4. Implement the following Boolean function with a 3-to-8 decoder:

   $$F(A, B, C) = \Sigma m(1,2,3,6)$$

   You must ONLY use value-fixing and the components stated above. Save the circuit as `circuit4.cct` .

5. A parity checker is a component that verifies the parity of the entire input. For example, a 9bit even parity checker would be able to tell that 100010001 contains error, while 000110000 does not. Given 9bit input $A_8 \ldots A_0$, design an even parity checker that outputs 0 when no error is found, and 1 when there is. You should store the implemented component as a `9bit Parity Checker` in your library, and circuit as `circuit5.cct` .

6. Implement an unsigned selective 2s complementer. It should take 4bits $A = A_3 \ldots A_0$ and C as input, $D = D_3 \ldots D_0$ as output. It should output A when $C = 0$, the unsigned 2s complement of A when $C = 1$. You should store the component as a `4bit 2s Comp` in your library, and circuit as `circuit6.cct` .