CSE 308: Digital System Design | Homework 1

HOMEWORK POLICY

- The purposes of homework include:
  - developing responsible study skills and work habits,
  - practicing a skill or process that students can do independently but not fluently,
  - previewing or preparing for new content, and reflecting or elaborating on information learned in class to deepen students' knowledge.

- Late submission of homework will result in zero score.

- Independently complete homework to the best of your ability.

- If you have any questions related to homework:
  - study with a friend,
  - form a study group in the class,
  - drop your query on Classroom's Stream.

- Collaboration is fine, but it should be you alone who writes up the answers.

- Academic dishonesty will not be tolerated and will result in at least an F for the homework. Your prospectus clearly states "suspension from the university is the expected penalty" for "plagiarism, cheating, and academic integrity issues" and this includes submitting the work of another person as your own or permitting another to submit yours as his/her own.

- Homework will be checked via the quiz policy. There will usually be one problem directly from the homework on the quiz/exam. If you have done all the homework, the quiz/exam should be automatic! If you do not do the homework, you are very unlikely to do well on the quiz/exam, and you should then expect to fail.

Problem 1: Draw the logic circuit described by the Verilog code below.

```
module pie( x, y, a, b, c);
   input a, b, c;             output x, y;
   wire   t1, t2;

   xor x1(t1,a,b);
   not n1(x,t1);
   and a1(t2,x,c);
   or  o1(y,t2,b);

endmodule
```

Problem 2: Draw logic circuit described by the Verilog module `twoterms` below.

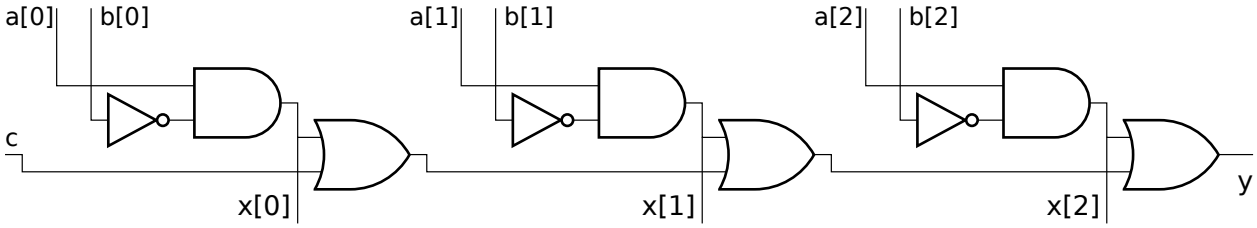
- Show the contents of each instantiated module. (That is, do **not** just show a box labeled `term1100` or `twoterms`.)
- Show using AND, OR, and NOT gates, inferring the correct gate for the Verilog operators used in the `assign` expression.
- To the extent possible, label the diagram using the port names defined by `twoterms` (`x`, `i`, `j`, `k`, and `m`).

```
module term1100(x,a,b,c,d);
   input a, b, c, d;             output x;
   assign x = a && b && !c && !d;
endmodule

module twoterms_bundle(x,a);
   input [3:0] a;             output      x;
   wire        tx101, t100x;
   term1100 t0(tx101, a[2], a[0], a[1], 1'b0);
   term1100 t1(t100x, a[3], a[3], a[1], a[2]);
   or o1(x, tx101, t100x);
endmodule

module twoterms(x,i,j,k,m);
   input i,j,k,m;              output x;
   wire [3:0] bundle;
   assign bundle[3:0] = {i,j,k,m};
   twoterms_bundle t(x,bundle);
endmodule
```

Problem 3: The logic below consists of three repeated parts. Write a Verilog explicit structural description of the logic which consists of two modules, one module, name it `part`, will be for the part that's repeated, the other, name it `whole`, will instantiate `part` three times and interconnect them appropriately. Choose appropriate inputs and outputs for the two modules based on the diagram.



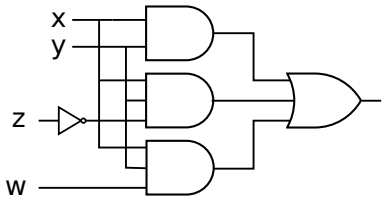
Problem 4: Replace each assign statement below with explicit structural code. Consider each assign statement in isolation (they are not part of the same module). There is no need to show the module declarations.

```
assign x = a & b ? 1 : 0;

assign x = a == b ? 0 : 1;

assign x = a ? b : c;
```

Problem 5: Write a Verilog explicit structural description of the following logic.



Problem 6: Write a Verilog implicit structural description of the following logic.