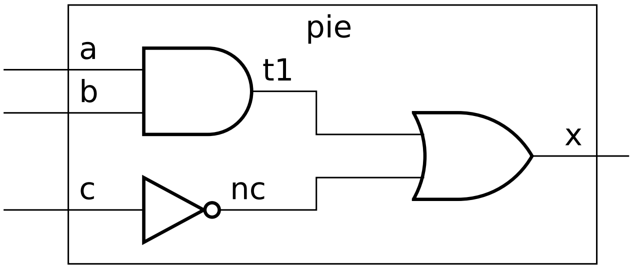
LSU EE 3755 Homework 1 Solution Due: 10 September 2012

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

```
module pie(x,a,b,c);
   input a, b, c;             output x;
   wire   t1, nc;
   and a1(t1,a,b);
   not n1(nc,c);
   or o1(x,t1,nc);
endmodule
```

Solution:



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

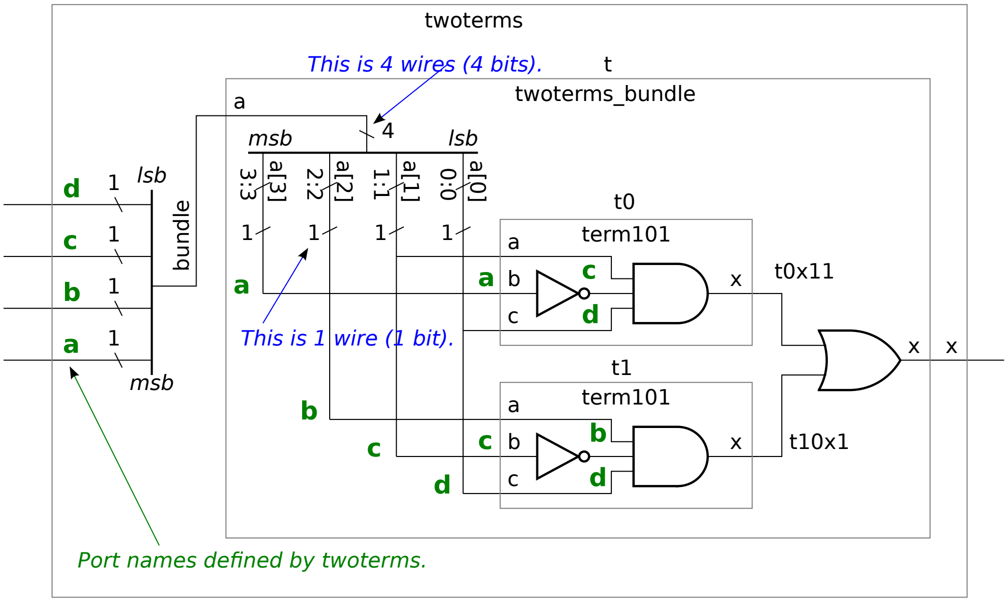
- Show the contents of each instantiated module. (That is, do **not** just show a box labeled `term101` or `twoterms`.)
- Show using AND, OR, and NOT gates, inferring the correct gate for the Verilog operators used in the `assign` expression.
- To the extent possible, label the diagram using the port names defined by `twoterms` (`x`, `a`, `b`, `c`, and `d`).

```
module term101(x,a,b,c);
   input a, b, c;              output x;
   assign x = a && !b && c;
endmodule

module twoterms_bundle(x,a);
   input [3:0] a;              output     x;
   wire        t0x11, t10x1;
   term101 t0(t0x11,a[1],a[3],a[0]);
   term101 t1(t10x1,a[2],a[1],a[0]);
   or o1(x, t0x11, t10x1);
endmodule

module twoterms(x,a,b,c,d);
   input a,b,c,d;               output x;
   wire [3:0] bundle;
   assign bundle[3:0] = {a,b,c,d};
   twoterms_bundle t(x,bundle);
endmodule
```

Solution appears below. The labels defined by module `twoterms` appear in **green bold**. Notice that at the input to `twoterms` the four inputs, `a`, `b`, `c`, and `d` are concatenated into a single wire named `bundle`. This is shown in the diagram using a medium-weight line. The opposite operation is performed by the bit select operator (such as `a[0]`) operating on `a` (a.k.a. `bundle`) in `twoterms_bundle`. This is also shown with a medium-weight line, but this time the input is a 4-bit wire (`a`) and there are four 1-bit outputs.

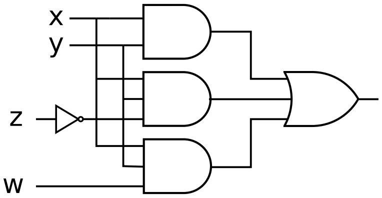


This is 4 wires (4 bits).

This is 1 wire (1 bit).

Port names defined by twoterms.

Problem 3: Write a Verilog explicit structural description of the following logic.



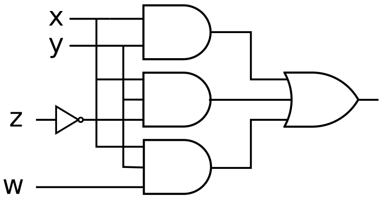
Solution appears below.

```
module problem_3(out,x,y,z,w);
   input x,y,z,w;
   output out;

   and a1(xy,x,y);
   not n1(nz,z);
   and a2(xynz,x,y,nz);
   and a3(xyw,x,y,w);
   or o1(out,xy,xynz,xyw);

endmodule
```

Problem 4: Write a Verilog implicit structural description of the following logic.



Solution appears below.

```
module problem_4(out,x,y,z,w);
   input x,y,z,w;
   output out;

   assign out = x && y || x && y && ~z || x && y && w;

endmodule
```