BEE 271 Spring 2017
Homework 4 answers

Please answer the following questions. Each is worth 8 points.

1. Write a Verilog module using a casex statement that takes a 4-bit value as input and outputs the number of leading 1's, e.g., 0111 → 0, 1011 → 1, 1100 → 2, etc.

```
module CountLeadingOnes( input [ 3:0 ] v, output reg [ 2:0 ] p );
   always @( * )
      casex ( v )
         4'b0xxx: p = 0;
         4'b10xx: p = 1;
         4'b110x: p = 2;
         4'b1110: p = 3;
         4'b1111: p = 4;
      endcase
endmodule
```

2. Create a module in Verilog that adds a 4-bit ***unsigned*** binary number to an 8-bit ***unsigned*** binary number, producing a 10-bit ***unsigned*** result using the + operator.

```
module AddU( input [ 3:0 ] A, input [ 7:0 ] B, output [ 9:0 ] sum );

   // Verilog pads high-order bits with zeros.

   assign sum = A + B;
endmodule
```

3. Create a module in Verilog that adds a 4-bit ***signed*** binary number to an 8-bit ***unsigned*** binary number, producing a 10-bit ***signed*** result using the + operator.

Two possible approaches using explicit sign extension or the new signed keyword:

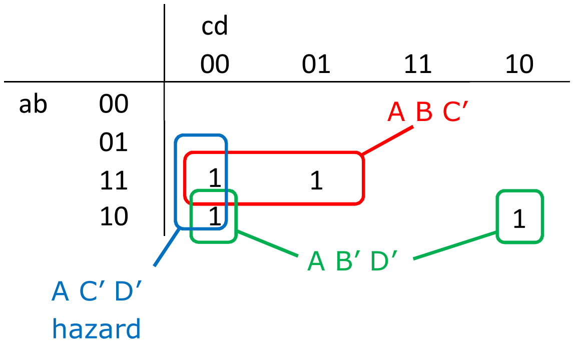
```
module AddS1( input [ 3:0 ] A, input [ 7:0 ] B, output [ 9:0 ] sum );

   // Sign-extend A, replicating the sign bit A[ 3 ] through 6
   // positions.

   assign sum = { { 6{ A[ 3 ] } }, A } + B;
endmodule


module AddS2( input signed [ 3:0 ] A, input [ 7:0 ] B,
      output signed [ 9:0 ] sum );

   // Using the signed keyword.

   assign sum = A + B;
endmodule
```

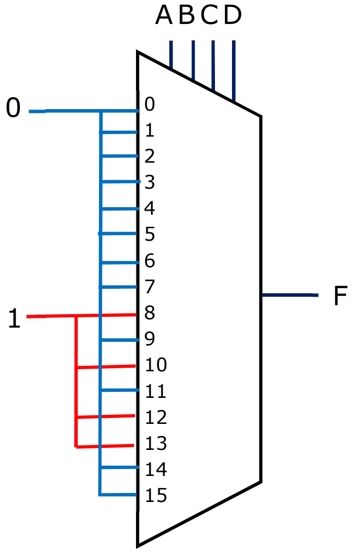
4. What is Shannon's expansion? What is a cofactor?

If $f = f(x1, x2, x3, \ldots, xn)$ then $f = x1' f_{x1'} + x1 f_{x1}$
where $f_{x1'}$ and f_{x1} are cofactors of f with respect to x1′ and x1 such that
$f_{x1'} = f(0, x2, x3, \ldots, xn)$ and $f_{x1} = f(1, x2, x3, \ldots, xn)$

5. Implement F = A B C′ + A B′ D′ using a 16:1 multiplexer.

Plotting this on a Karnaugh map, we can see that F = Σ m(8, 10, 12, 13), meaning this function can be implemented by tying those inputs to 1 and the others to 0.





6. Implement your 16:1 multiplexer solution to F = A B C′ + A B′ D′ as directly as you can in Verilog using a case statement and the concatenation operator.

```
module solution( input A, B, C, D, output reg f );

   always @( * )
      case ( { A, B, C, D } )
         8, 10, 12, 13: f = 1;
         default: f = 0;
      endcase

endmodule
```

7. Use Shannon's expansion to implement F = A B C′ + A B′ D′ using a 2:1 multiplexer, two NOR gates and one inverters. Show your steps.

$F = A\,B\,C' + A\,B'\,D'$

$F_{A'} = 0$

$F_A = B\,C' + B'\,D'$

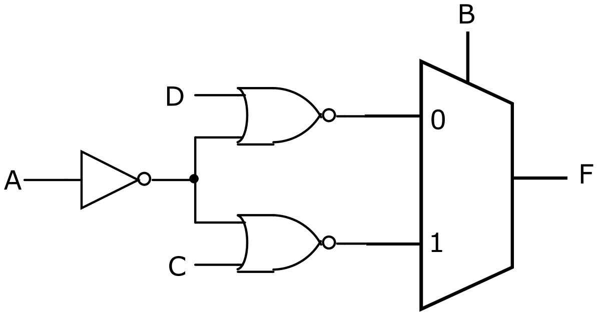
$$\boxed{\begin{array}{l} F_{B'} = A\,D' = (\,A' + D\,)' \\ F_B = A\,C' = (\,A' + C\,)' \end{array}}$$

$F_{C'} = A\,B + A\,B'\,D'$

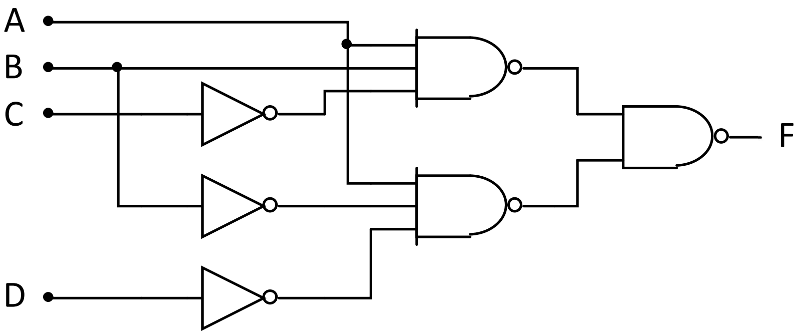
$F_C = A\,B'\,D'$

$F_{D'} = A\,B\,C' + A\,B'$

$F_D = A\,B\,C'$



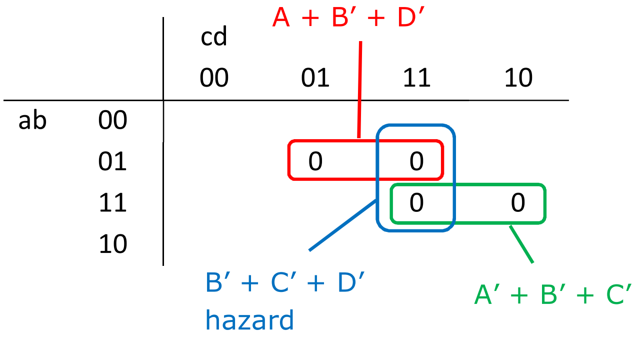
8. Draw a schematic that implements F = A B C′ + A B′ D′ using NAND gates and inverters. Will it have a hazard? If there is a hazard, will it be a static 1 or a static 0 hazard and how could you fix the formula and your circuit to eliminate the hazard?



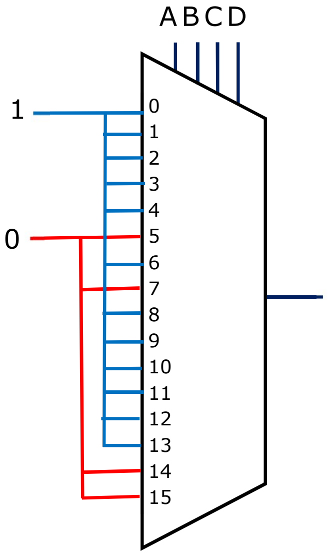
Yes, it has a static 1 hazard, shown on the Karnaugh map in problem 5. To fix it, add a A C′ D′ term to join the two adjacent 1's at 1000 and 1100 into a single implicant.

9. Implement F = (A + B′ + D′) (A′ + B′ + C′) using a 16:1 multiplexer.

Plotting this on a Karnaugh map, we can see that F = Π M(5, 7, 14, 15), meaning this function can be implemented by tying those inputs to 0 and the others to 1.



ab \ cd	00	01	11	10
00				
01		0	0	
11			0	0
10				



10. Use Shannon’s expansion to implement F = (A + B′ + D′) (A′ + B′ + C′) using a 2:1 multiplexer and two NAND gates. Show your steps.

$F = (A + B' + D')(A' + B' + C')$

$F_{A'} = B' + D' = (B D)'$

$F_{A} = B' + C' = (B C)'$

$F_{B'} = 1$

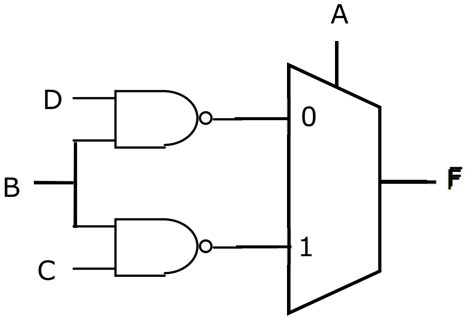
$F_{B} = (A + D')(A' + C')$

$F_{C'} = (A + B' + D')(A' + B')$

$F_{C} = (A + B' + D')$

$F_{D'} = (A + B')(A' + B' + C')$

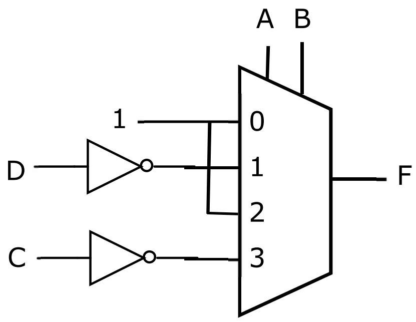
$F_{D} = A' + B' + C'$



11. Use Shannon's expansion to implement $F = (A + B' + D') (A' + B' + C')$ using a 4:1 multiplexer and two inverters. Show your steps.

To solve this, you'll need to try all the possibilities of two inputs at a time as selection lines to the 4:1 multiplexer until you find the right one. Only AB is the right solution.

$F = (A + B' + D') (A' + B' + C')$



$F_{A'} = B' + D' = (B D)'$

$F_{A} = B' + C' = (B C)'$

$F_{A'B'} = 1$

$F_{A'B} = D'$

$F_{AB'} = 1$

$F_{AB} = C'$

$F_{A'C'} = (B D)'$

$F_{A'C} = 0$

$F_{AC'} = 1$

$F_{AC} = B'$

$F_{A'D'} = 1$

$F_{A'D} = B'$

$F_{AD'} = (B C)'$

$F_{AD} = (B C)'$

$F_{B'} = 1$

$F_{B} = (A + D') (A' + C')$

$F_{B'C'} = 1$

$F_{B'C} = 1$

$F_{BC'} = A + D'$

$F_{B'C} = (A + D') A' = A' D'$

$F_{B'D'} = A' + C'$

$F_{B'D} = 1$

$F_{BD'} = A' + C'$

$F_{B'D} = A (A' + C') = A C'$

$F_{C'} = A + B' + D'$

$F_{C} = (A + B' + D') (A' + B')$

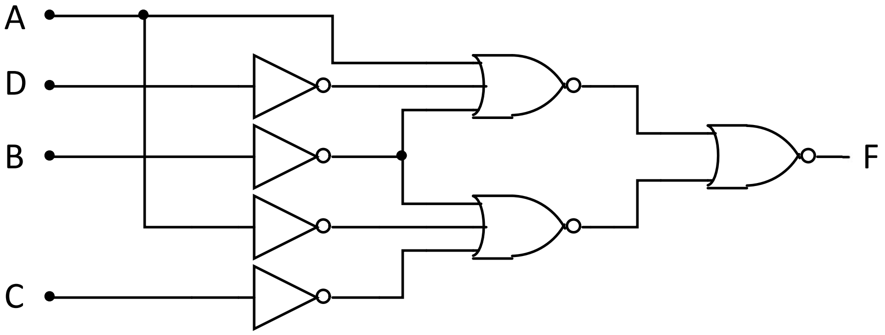
$F_{C'D} = A + B'$

$F_{C'D'} = 1$

$F_{CD} = A + B' + D'$

$F_{CD'} = 1$

12. Draw a schematic that implements $F = (A + B' + D') (A' + B' + C')$ using NOR gates and inverters. Will it have a hazard? If there is a hazard, will it be a static 1 or a static 0 hazard and how could you fix the formula and your circuit to eliminate the hazard?



Yes, it has a static 0 hazard, shown on the Karnaugh map in problem 9. To fix it, add a (B′ + C′ + D′) term to join the two adjacent 0’s at 0111 and 1111 into a single implicant.