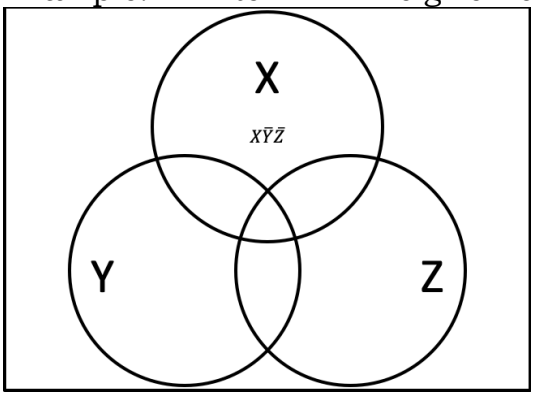
Cpr E 281 HW05
ELECTRICAL AND COMPUTER ENGINEERING
IOWA STATE UNIVERSITY

Karnaugh Maps
Assigned: Week 5
Due Date: Sep. 21, 2020

P1 (15 points): Minimization with Venn Diagrams

a) On a 3 variable Venn Diagram, show the location of all 8 min-terms. Example: min-term $X\bar{Y}\bar{Z}$ is given below.



b) Show a separate Venn diagram for each product term in the function $f = x\bar{y}z + xy + \bar{x}z$.

c) Use the Venn diagrams from (b) to find the minimal SOP form of f.

P2 (20 points): Given the two functions $F = a\bar{b}c$ and $G = a\bar{b}c + a\bar{c}\bar{d} + a\bar{c}d$:

a) Create a minimal cost circuit that implements both F and G using only NOT, AND, and OR gates. Show that the minimal cost circuit has a cost of 14 (cost = # of gates + # of inputs).

b) Implement the circuit that you made in part a using only (a minimal number of) NAND gates. Your circuit should only use 6 NAND gates.

P3 (15 points): Use a K-map to derive the simplest SOP expressions for the given shorthand expressions.

a) $F_1(X,Y,Z) = \sum m(1,4,5) + D(0,2)$

b) $F_2(W,X,Y,Z) = \sum m(1,3) + D(0,2,5,7,8,9,10,11)$

c) $F_3(W,X,Y,Z) = \sum m(1,3,4,6,9,11,12,14) + D(0,2,5,7,8,10,13,15)$

P4 (15 points): Use a K-map to derive the simplest POS expressions for the given shorthand POS expressions.

a) $G_1(x,y,z) = M_4 + D(0,1,2,5)$

b) $G_2(w,x,y,z) = \prod M(6,12) + D(0,4,5,7,8)$

c) $G_3(w,x,y,z) = \prod M(3,4,5,7,8,11,12,13,14) + D(0,1,2,6,9,10,15)$

P5 (15 points): Given the function $P(a,b,c,d) = \prod M(0,7,10,11) + D(1,3,4,6,8,9,13,14,15)$:

a) Derive a simplest SOP expression for P using a K-map.

b) Derive a simplest POS expression for P using a K-map.

c) Draw a circuit which implements P but uses only NAND gates.

Karnaugh Maps
Assigned: Week 5
Due Date: Sep. 21, 2020

P6 (20 Points) Consider the following Structural Verilog code:

```
module structural_code (A,B,C,D,F);
      input A,B,C,D;
      output F;

      not(notA, A);
      not(notB, B);
      not(notC, C);
      not(notD, D);

      and(v1, A, B);
      and(ABD, v1, D);

      and(w1, A, notC);
      and(AnCnD, w1, notD);

      and(x1, notA, notC);
      and(nAnCnD, x1, notD);

      and(y1, notB, C);
      and(nBCnD, y1, notD);

      or(z1, ABD, AnCnD);
      or(z2, z1, nAnCnD);
      or(F, z2, nBCnD);

endmodule
```

a) Write the expression that is implemented by the Structural Verilog code
b) Simplify your expression from part (a) using a k-map
c) Write out your expression using **Behavioral** Verilog

```
module behavioral_code (A,B,C,D,F);
      input A,B,C,D;
      output F;

      assign F =                                          ;

endmodule
```