UNITED STATES AIR FORCE ACADEMY

ECE 281

Lesson 8 Notes

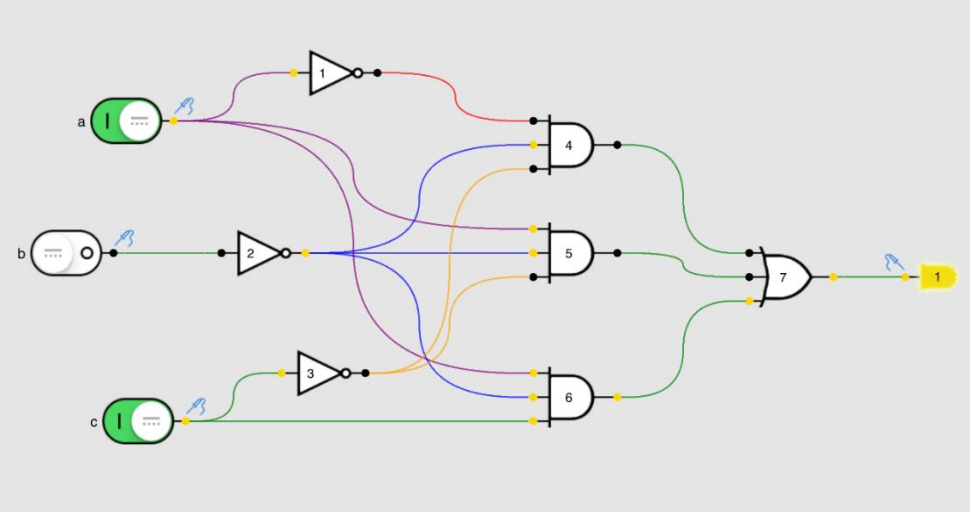
Objectives:

- Based on a problem description, design a solution using Boolean equations to solve the problem
- Utilize Boolean algebra to simplify Boolean equations or to derive a form that yields the most desirable hardware configuration
- Demonstrate the ability to describe a combinational digital system by a truth table, Boolean equation, Sigma/Pi notation, and schematic
- Given any one form, be able to produce the others

Review and Example #1: Recall from lesson 5 where we introduced the circuit below which was essentially the logic representation for the Sum of Products form obtained from the included truth table

Y = A'B'C'+AB'C'+AB'C

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0



Boolean Algebra Simplification:

Y = A'B'C'+AB'C'+AB'C

↓ *Distributive*

Y = A'B'C' + (AB')(C+C')

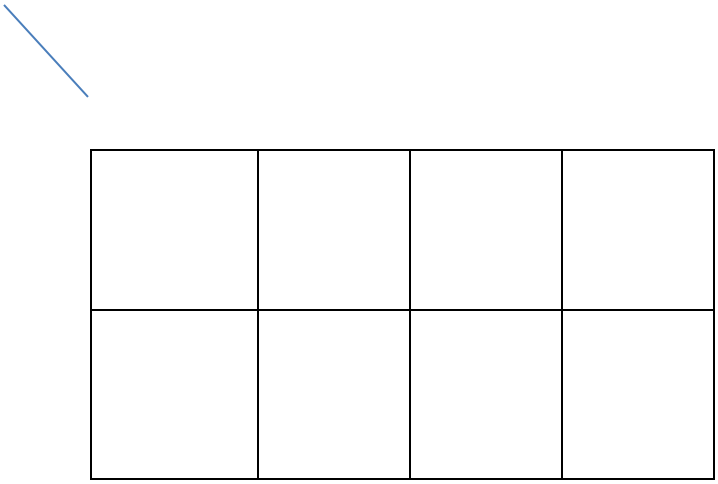
↓ *Complement*

Y = A'B'C' + (AB')(1)

↓ *Identity*

Y = A'B'C' + AB'

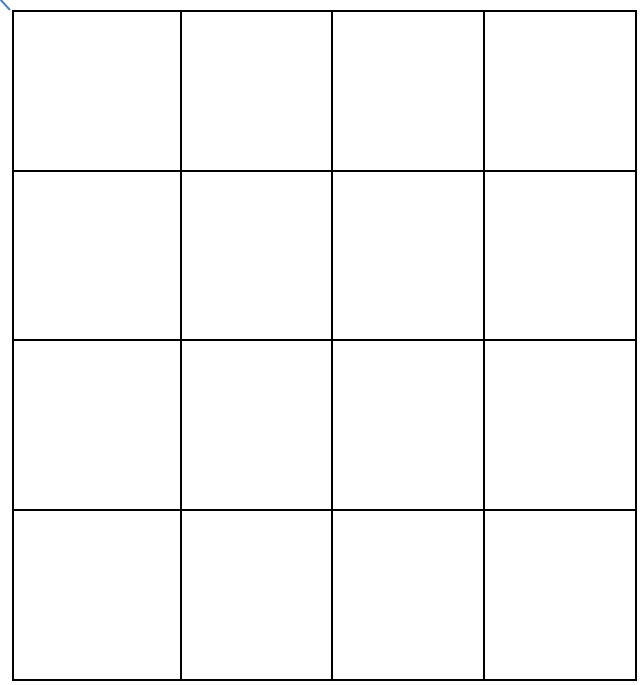
Introduction to K-Maps: We can use another tool known as the K-Map to assist with reducing the logic representations of our systems.



A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

K-Maps: a tool that provides an easy visual way to minimize logic

- Adjacent containers on a K-Map must differ by only one bit



We will limit ourselves to 4-input K-Maps in this class, but you could have larger

What does each container represent?

Prime Implicants:

Rules for K-Maps:

1. Use the fewest circles necessary to cover all the 1's
2. All containers in each circle must contain 1's
3. Each circle must span a rectangular block that is a power of 2 (i.e. 1,2,4,8…) containers in each direction
4. Each circle should be as large as possible
5. A circle may wrap around the edges of the K-map
6. A 1 in a K-map may be circled multiple times if doing so allows for fewer circles to be used
7. Circle X's (don't cares) to make a bigger circle
    a. Not always necessary to circle X's

Examples – Please use the following truth tables and the associated K-Map to find the minimal logic representation.

A	B	C	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

A	B	C	D	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	X
0	1	0	0	X
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	X
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Additional Points:

- K-Maps visualize prime implicants, placing minterms that vary by just one literal next to one another
- K-Maps wrap around from left to right/right to left and top to bottom / bottom to top