

HW3: Combinational Logic and Components

1. **Turn K-maps back into a truth table:** For each of the following, find the truth tables corresponding to the functions defined by the K-maps:

(a)

		A	
		0	1
BC	00	1	1
	01	1	0
	11	1	1
	10	0	0

(b)

		AB			
		00	01	11	10
CD	00	1	0	0	1
	01	1	0	0	1
	11	1	1	1	1
	10	0	1	0	0

2. Obtain an MSOP and an MPOS for each of the following functions defined by the K-maps:

(a)

		A	
		0	1
BC	00	1	0
	01	1	0
	11	1	1
	10	0	1

(b)

		AB			
		00	01	11	10
CD	00	0	0	1	1
	01	1	1	1	0
	11	0	0	1	0
	10	0	0	1	0

3. Simplifying K-Maps with Don't Care values: For each of the following K-maps:

- Find the MSOP expression (show groupings)
- Find the MPOS expression (show groupings)
- Are your solutions unique, or are there other minimum expressions?
- Does the MPOS = MSOP?

(a)

		ab			
		00	01	11	10
cd	00	X	0	0	1
	01	1	0	0	X
	11	0	X	0	1
	10	0	0	0	1

(b)

		ab			
		00	01	11	10
cd	00	1	X	0	1
	01	1	1	1	0
	11	0	0	X	0
	10	X	0	1	1

4. Adder design:

(a) Determine the complete truth table for a 2-bit adder (4 inputs A_1 , A_0 , B_1 , B_0), and derive equations for both the sum (S_1 , S_0) and the carry (C_{out}) outputs (no need to simplify).

(b) Design a 1-bit full adder (3 inputs A , B , C_{in}) (create a truth table, and show the equations) and use this component to create the equivalent of a 2-bit adder by chaining two components together. The overall circuit you create should have inputs (A_1 , A_0 , B_1 , B_0 , C_{in}), and outputs (S_1 , S_0 , C_{out}).