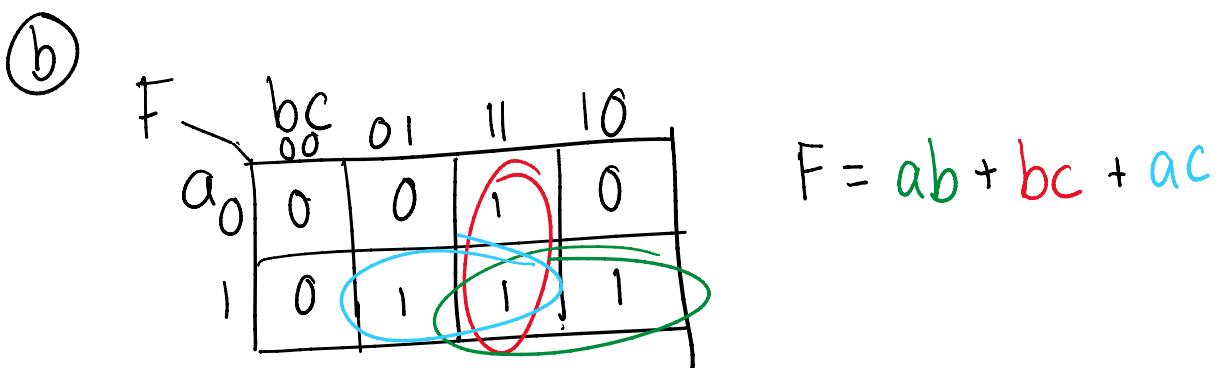


# Digital Design Hwk 3

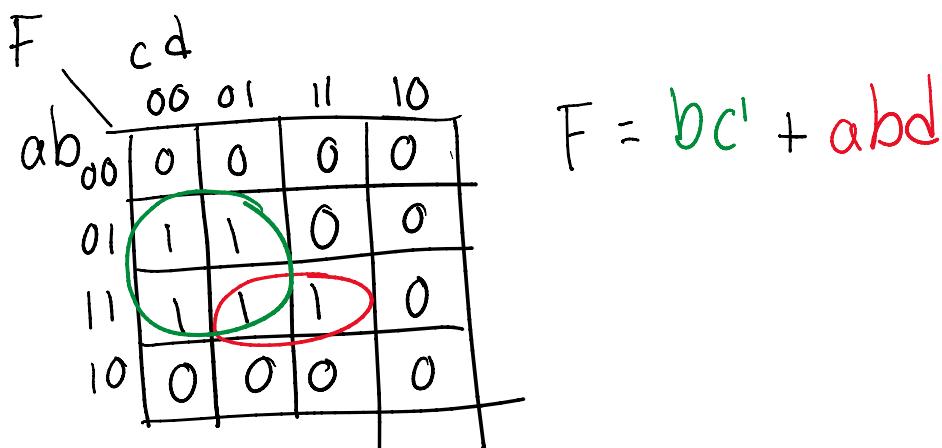
Sunday, October 20, 2019 2:28 PM

- 6.3 Perform two-level logic size optimization for  $F(a,b,c) = ab'c + abc + a'bc + abc'$  using  
 (a) algebraic methods, (b) a K-map. Express the answers in sum-of-products form.

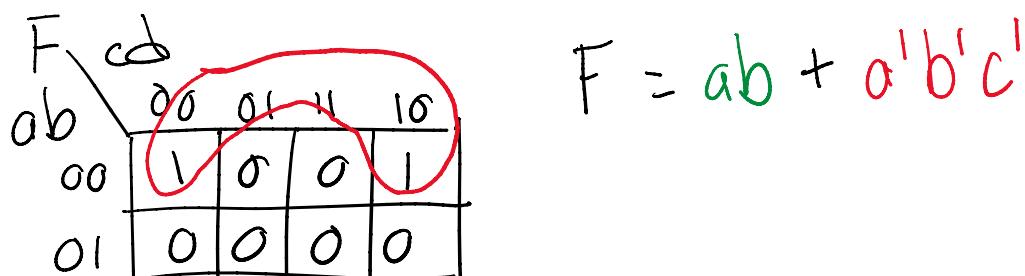
$$\begin{aligned}
 @a \quad F &= ab'c + abc + a'bc + abc' \\
 &= ab'c + abc + abc + a'bc + abc + abc' \\
 &= ac(b' + b) + bc(a + a') + ab(c + c') \\
 &\equiv ac + bc + ab
 \end{aligned}$$



- 6.5 Perform two-level logic size optimization for  $F(a,b,c,d) = a'bc' + abc'd' + abd$  using a K-map.



- 6.6 Perform two-level logic size optimization  $F(a,b,c,d) = ab + a'b'd'$  using a K-map.



	01	00	00	00
	11	11	11	11
	10	00	00	00

6.7 Perform two-level logic size optimization for  $F(a, b, c) = a'b'c + abc$ , assuming input combinations  $a'b'c$  and  $ab'c$  can never occur (those two minterms represent don't cares).

	$b'c$	00	01	11	10
$a$	0	0	1	?	0
	1	0	?	1	0

$F = C$

6.8 Perform two-level logic size optimization for  $F(a, b, c, d) = a'b'c'd + ab'cd'$ , assuming that  $a$  and  $b$  can never both be 1 at the same time, and that  $c$  and  $d$  can never both be 1 at the same time (i.e., there are don't cares).

	$cd$	00	01	11	10
$ab$	00	0	0	?	0
	01	0	1	?	0
	11	?	?	?	?
	10	0	0	?	1

$F = ac + bd$

### SECTION 3.2: SHIFTING ONE BIT—FLIP-FLOPS

3.1 Compute the clock period for the following clock frequencies.

- 50 kHz (early computers)
- 300 MHz (Sony Playstation 2 processor)
- 3.4 GHz (Intel Pentium 4 processor)
- 10 GHz (PCs of the early 2010s)
- 1 THz (1 terahertz) (PC of the future?)

Ⓐ  $\frac{1}{50,000} = 0.00002 \text{ s}$

(a)  $\frac{1}{50,000} = 0.00002 \text{ s}$

(b)  $\frac{1}{300,000,000} = 3.33 \text{ ns}$

(c)  $\frac{1}{3,400,000,000} = 0.294 \text{ ns}$

(d)  $\frac{1}{10,000,000,000} = 0.1 \text{ ns}$

(e)  $\frac{1}{1,000,000,000,000} = 0.001 \text{ ns}$

3.8 Trace the behavior of a level-sensitive SR latch (see Figure 3.16) for the input pattern in Figure 3.95. Assume S1, R1, and Q are initially 0. Complete the timing diagram, assuming logic gates have a tiny but nonzero delay..

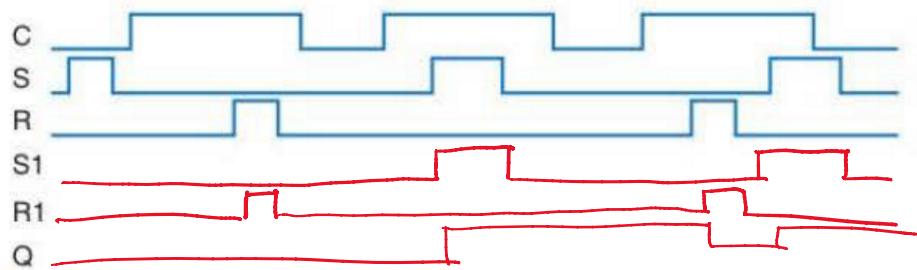


Figure 3.95 SR latch input pattern timing diagram.

Question 8: Simplify the following Boolean expression using K-map.

8.1 List all the primary implicants

8.2 List all the essential primary implicants

8.3 Derive the final optimized Boolean expression

8.1

$$\begin{aligned} O &= A'B'C'D \\ Z &= A'B'C'D' \\ Y &= A'BC'D' \end{aligned}$$

8.2 List all the essential primary implicants

8.3 Derive the final optimized Boolean expression

$$F(A,B,C,D) = \Sigma m(0, 2, 4, 5, 10, 11, 12, 13, 15)$$

F

cd

ab

	00	01	11	10
00	0 1	1 0	3 0	2 1
01	4 1	5 1	7 0	6 0
11	12 1	13 1	15 1	14 0
10	8 0	9 0	11 1	10 1

$$\begin{aligned}4 &= A'BC'D' \\5 &= A'BC'D \\10 &= AB'CD' \\11 &= AB'C'D \\12 &= ABC'D' \\13 &= ABC'D \\15 &= ABCD\end{aligned}$$

(8.2)  $(4, 5, 12, 13) = BC'$

$$(0, 2) = A'B'D'$$

$$(2, 10) = BCD$$

$$(11, 15) = ACD$$

(8.3)  $F = BC' + A'B'D' + A'BD' + ABD$

Question 9: Simplify the following Boolean function.

9.1 Drive the simplified Boolean function using K-map

9.2 Implement the design using NAND gate only based on the result of 9.1

$$F(X,Y,Z) = \Sigma m(1, 2, 3, 4, 5, 7)$$

Q.1

$F$

$X \swarrow Y \searrow Z$

	00	01	11	10
0	0   1   3   2			
1	4   5   7   6   0			

Q.2

$F$  (NAND)

$X \swarrow Y \searrow Z$

	00	01	11	10
0	0   1   3   2			
1	4   5   7   6   0			