CSEE 3827: Fundamentals of Computer Systems

Midterm Exam
March 11, 2009

SOLUTIONS

Read all of the following information before starting the exam:

- Be sure to write your name on each page of the exam.
- Use the exam itself for your solutions (no blue books or spare sheets of paper). You may use the backside of pages if you need more space.
- Show your work in order to earn partial credit.
- You may use your textbook and class notes, but *absolutely no electronic devices (laptops, cell phones, etc.)*
- Good luck!

Problem	**Point Value**	**Points Earned**
1	6	
2	12	
3	10	
4	10	
5	20	
Total	58	

1. Each of the following arithmetic operations is correct in at least one number system (e.g., base n). Determine possible bases of the numbers in each operation.

   (a) $41/3 = 13$

   $41_8/3_8 = 13_8$

   (b) $33/3 = 11$

   $33_{n\geq4}/3_{n\geq4} = 11_{n\geq4}$

   (c) $\sqrt{41} = 5$

   $\sqrt{41_6} = 5_6$

2. A self-dual logic function is a function F such that $F = dual(F)$. Which of the following functions are self-dual?

(a) $F = X$

$dual(F) = X \Rightarrow$ self-dual.

(b) $F = \Sigma_{X,Y,Z}(0, 3, 5, 6)$

$F = \overline{X}\,\overline{Y}\,\overline{Z} + \overline{X}YZ + Z\overline{Y}Z + ZY\overline{Z}$
$dual(F) = (\overline{X} + \overline{Y} + \overline{Z})(\overline{X} + Y + Z)(Z + \overline{Y} + Z)(Z + Y + \overline{Z})$
$dual(F) = \Pi_{X,Y,Z}(7, 4, 2, 1) \Rightarrow$ self-dual.

(c) $F = X \cdot \overline{Y} + \overline{X} \cdot Y$

$F = \Sigma_{X,Y}(1, 2)$
$dual(F) = (X + \overline{Y})(\overline{X} + Y)$
$F = \Pi_{X,Y}(1, 2) \Rightarrow$ not self-dual.

(d) $F = W \cdot (X \oplus Y \oplus Z) + \overline{W} \cdot \overline{(X \oplus Y \oplus Z)}$

From part (c) we know that $dual(X \oplus Y) = X \overline{\oplus} Y$.
$dual(F) = (W + dual(X \oplus Y \oplus Z))(\overline{W} + dual(\overline{(X \oplus Y \oplus Z)}))$
$dual(F) = (W + (X \overline{\oplus} Y \overline{\oplus} Z))(\overline{W} + (\overline{(X \overline{\oplus} Y \overline{\oplus} Z)}))$
When $W = 1, X = 1, Y = 1, Z = 1$: $F = 1$ but $dual(F) = 0 \Rightarrow$ not self-dual.

(e) A function F of 7 variables such that $F = 1$ if and only if 4 or more of the variables are 1.

Consider a smaller version of the same problem: A function G of 3 variables such that $G = 1$ if and only if 2 or more of the variables are 1.
$G = ab + bc + ac$ (G has one term for each combination of two variables.)

- If *any* two vars are 1, some term will be 1 so $G = 1$.
- If *no* two vars are 1, all terms will be 0 so $G = 0$.

$dual(G) = (a + b)(b + c)(a + c)$

- If *any* two vars are 1, all terms will be 1, so $dual(G) = 1$.
- If *no* two vars are 1, some term will be 0 so $dual(G) = 0$.

Larger problem sizes follow the same pattern, $\Rightarrow$ self-dual.

(f) A function F of 10 variables such that $F = 1$ if and only if 5 or more of the variables are 1.

Again, consider a smaller version of the function: A function G of 4 variables such that $G = 1$ if and only if 2 or more of the variables are 1. $G = ab + bc + cd + da + ac + bd$ (As before, G has one term for each combination of two variables)

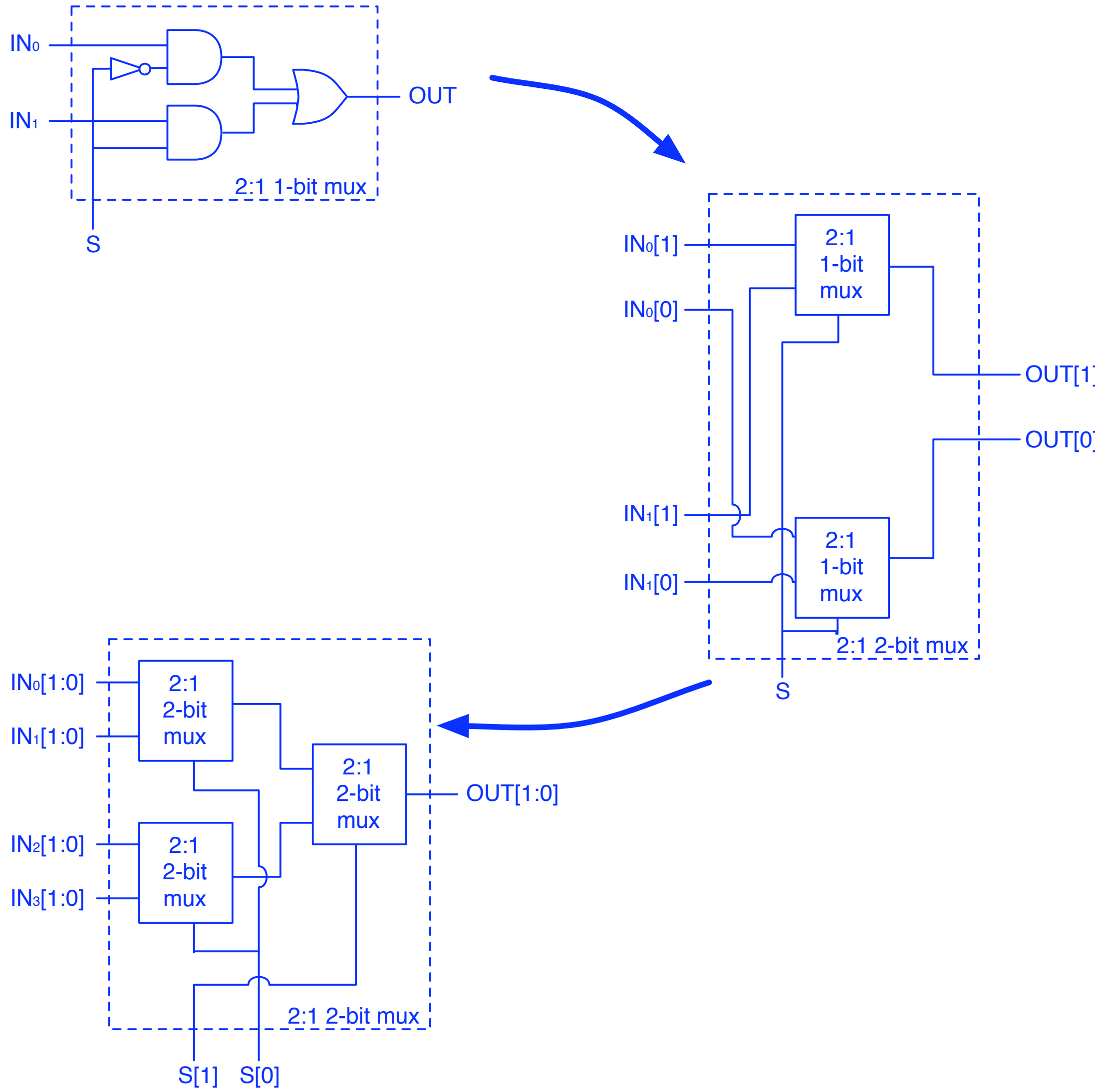
- If *any* two vars are 1, some term will be 1 so $G = 1$.
- If *no* two vars are 1, all terms will be 0 so $G = 0$.

$dual(G) = (a + b)(b + c)(c + d)(d + a)(a + c)(b + d)$ Consider the case when *only* two vars are 1. (e.g., $a = 1, b = 1, c = 0, d = 0$)

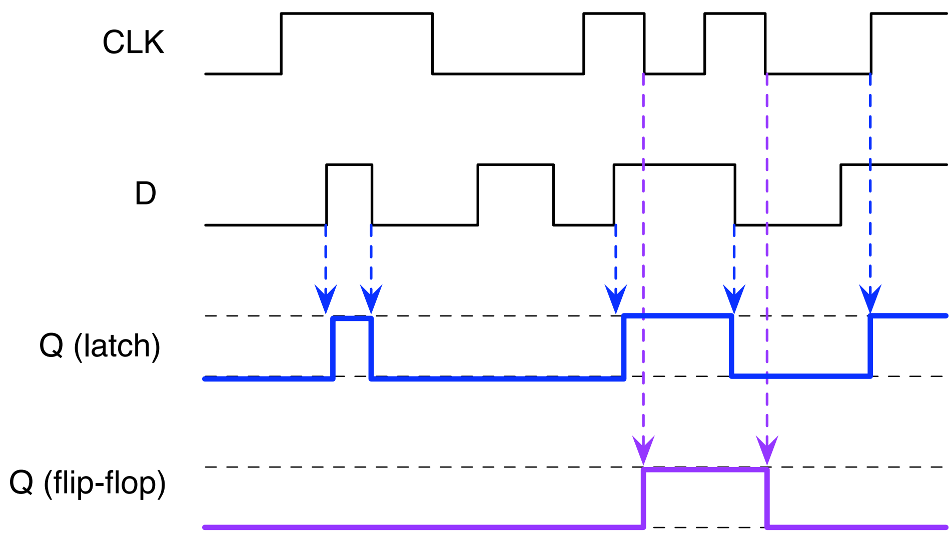
- By definition, $G = 1$
- The term corresponding to the *other* two variables (e.g., $(c + d)$) will be 0, so $dual(G) = 0$.

Larger problem sizes follow the same pattern, $\Rightarrow$ not self-dual.

3. Draw a schematic implementing a 4:1 2-bit multiplexer.



4. Given the input waveforms shown here, sketch the outputs of a D latch and a D falling edge triggered flip-flop. Assume that all wires and gates have no propagation delay.

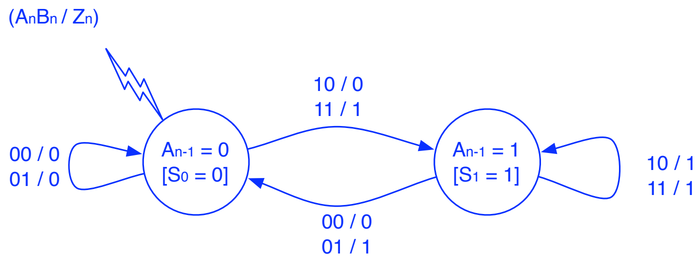


5. Design a finite state machine that takes two inputs, A and B, and generates one output, Z. The output in cycle n, Z_n, is either the boolean AND or the boolean OR of the corresponding A_n and the previous input A_{n-1}, depending on the other input, B_n:

$$Z_n = A_n \cdot A_{n-1} \quad \text{if } B_n = 0$$
$$Z_n = A_n + A_{n-1} \quad \text{if } B_n = 1$$

After designing, sketch an implementation of the finite state machine using D flip-flops.

(a) State diagram:

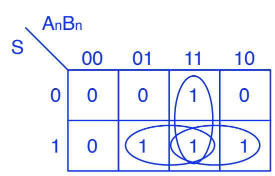


(b) State table:

S	A_n	B_n	S^+	Z_n
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

(c) Next state and output logic:

- Next state logic: $D = S^+ = A_n$
- Output logic:



$$Z_n = A_nB_n + A_nS + B_nS$$

(d) Circuit diagram: