UNIVERSITY OF CALIFORNIA
Department of Electrical Engineering and Computer Sciences
CS150 Fall 2001

Prof. Subramanian

Homework #5

Due: Friday, October 19, 2001

1. Given the following function in sum of products form (not necessarily minimized):
   F(A,B,C,D) = A'BC + AD + AC

   Draw the circuit for the following:

   a. F in minimized product of sums form
   b. F' in minimized sum of products form
   c. F using NOR gates only
   d. F' using NAND gates only
   e. F using a single OR-AND-Invert gate
   f. F' using a single AND-OR-Invert gate

Solution

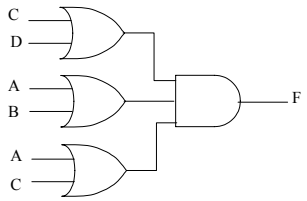
First, construct the K-map with the terms given.

CD \ AB				
	0	0	0	0
	0	0	1	1
	0	1	1	1
	0	1	1	1

a)

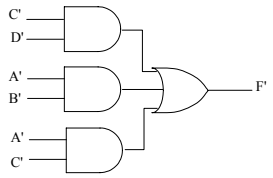
CD \ AB				
	0	0	0	0
	0	0	1	1
	0	1	1	1
	0	1	1	1

F = (C+D)(A+B)(A+C)

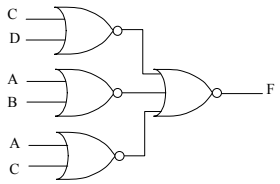


b) Take the inverse of the K-map of a) and group the ones, the circles are the same but now they represent minterms.

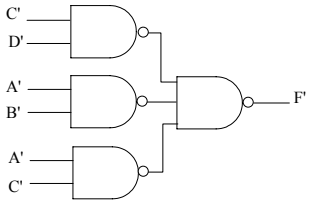
$$F' = C'D'+A'B'+A'C'$$



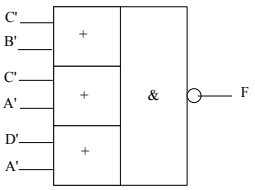
c)



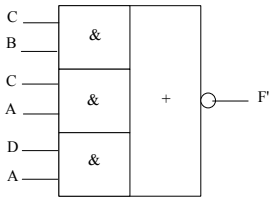
d)



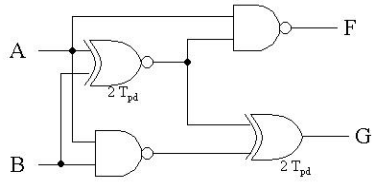
e)

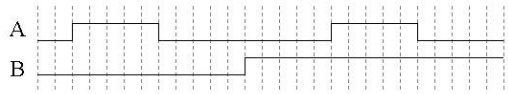


f)

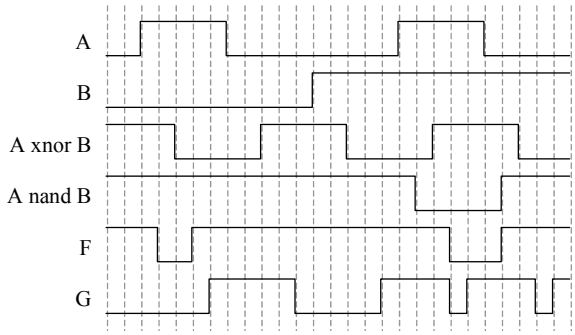


2. Consider the circuit below. Given that XOR/XNOR gates have twice the delay of NAND gates, what is the circuit's output response to the input waveforms in the timing diagram below? (Each 5-time-unit division represents one NAND gate delay.)





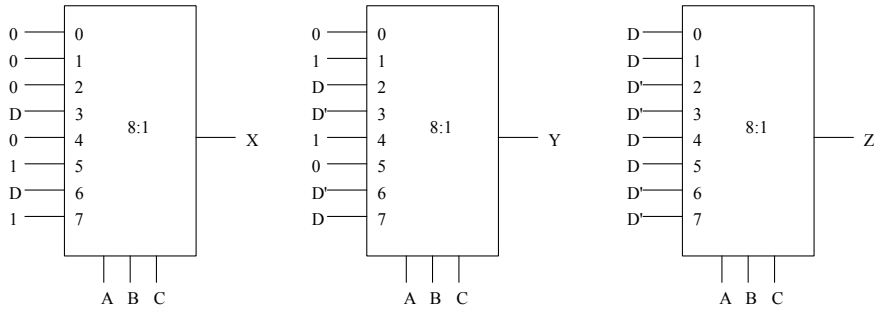
Solution



3. Implement the 2-bit adder function (i.e. 2-bit binary number AB plus 2-bit binary number CD yields a 3-bit result XYZ) using three 8:1 multiplexers. Show your truth table and how you derived the inputs to the multiplexers.

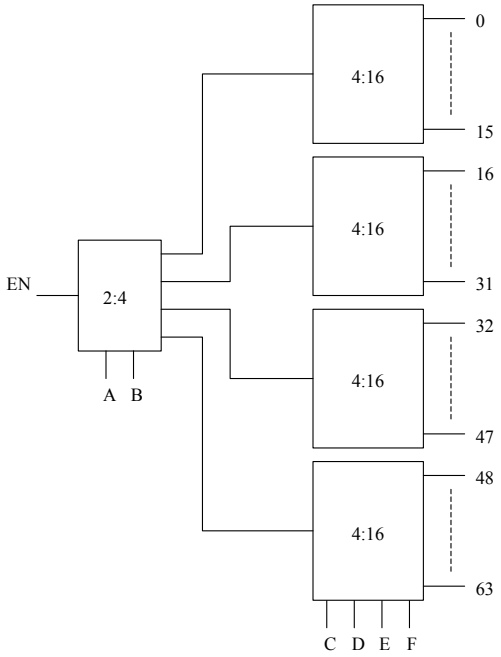
Solution

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



4. Demonstrate how to implement a 6:64 decoder using generic 2:4 and 4:16 decoders.

Solution



5. Implement a 2-bit combinational multiplier. It has two 2-bit inputs and a 4-bit output. The first 2-bit input is represented by the variables A, B; the second 2-bit input is represented by C, D. The outputs are W, X, Y, Z, from the most significant to the least significant.

    a. Complete a truth table that describes the functional behavior of the multiplier.
    b. Find the minimum sum of products forms for the outputs using the K-map method.

Solution

a)

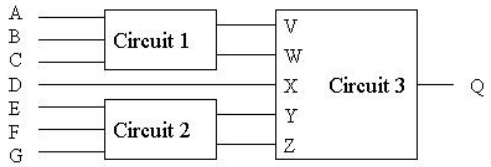
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

b) W = ABCD
X = ACD'+AB'C
Y = A'BC+BCD'+AC'D+AB'D
Z = BD

6. An n-input majority function asserts its output whenever more than half of its inputs are asserted. You are to implement a seven-input majority function, which will assert its output whenever four or more inputs are asserted.

   Don't panic just because this is a seven-variable function. Build it up as a multilevel function whose subfunctions each have less than six variables. As a block diagram, it looks like the figure below. Circuits 1 and 2 tally the number of their inputs that are asserted, providing the count in binary on the outputs (V, Y are the MSBs; W, Z are the LSBs). Based on these second-level inputs, Q determines if more than four or more inputs of the original inputs are 1.

   a. Find the minimized sum of products form for circuit 1 (circuit 2 is identical). The functions V and W should look familiar. What do they implement?
   b. Find the minimum sum of products form for circuit 3. (It would probably be helpful to write out the truth table first.)



Solution

a) V = BC+AC+AB (majority tule)
W = A'B'C + A'BC'+ABC+AB'C' (XOR)

b)

V	W	X	Y	Z	Q
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	1
1	0	0	1	1	1
1	0	1	0	0	0
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	0
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

V'

YZ \ WX			
0	0	0	0
0	0	0	0
0	1	1	0
0	0	1	0

V

0	0	1	0
0	1	1	1
1	1	1	1
1	1	1	1

Q = XYZ + WXY + VY + VXZ + VWZ + VWX