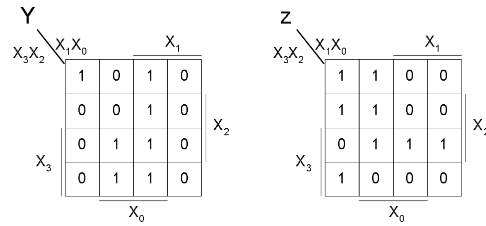
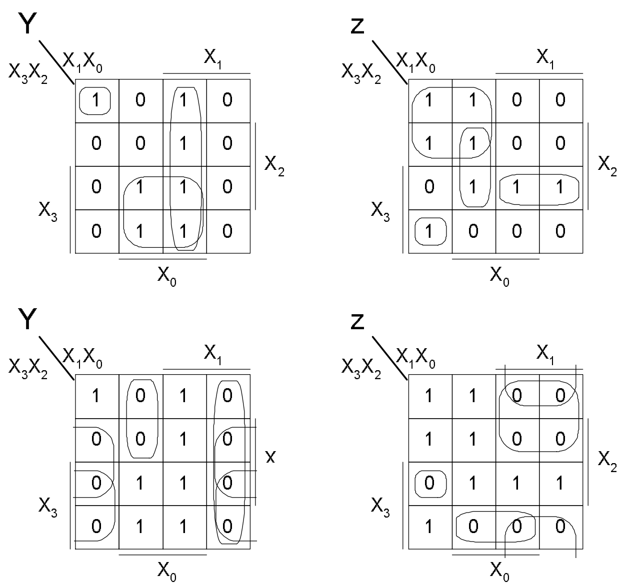
CS320 Fall 2014

Homework#2 Solutions

Problem 1: Find the minimal sum of products and products of sums for the functions shown in the k-maps. Implement Y using a NAND-NAND network, and implement Z using NOR-NOR network.

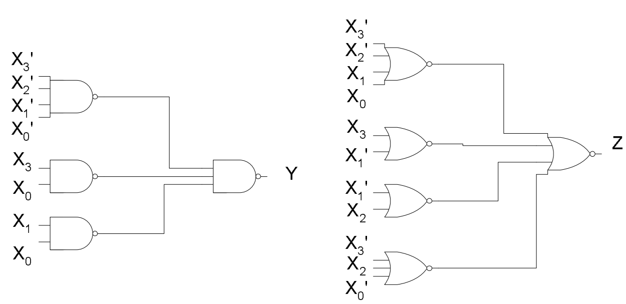


Solution:



NAND-NAND for Y: Sum of products
$Y = x_3'x_2'x_1'x_0' + x_3x_0 + x_1x_0$

NOR-NOR for X: Product of sums
$Z = (x_3' + x_2' + x_1 + x_0)(x_3 + x_1')(x_2 + x_1')(x_3' + x_2 + x_0'))$

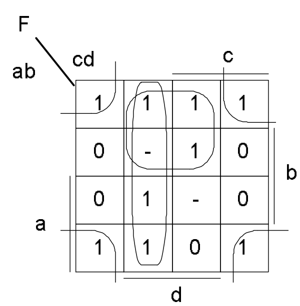


Problem 2: A logic circuit realizes the function F(a,b,c,d) = a'b'+a'cd+ac'd+ab'd'. Assuming that a=c never occurs when b = d = 1, find a simplified expression and circuit for F. (Hint: When can Don't-Cares be used?)

Solution:

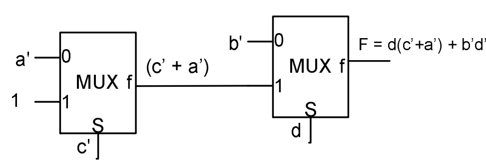
Begin by creating a table of the function F, and then determining the Don't cares (F'). This step can be skipped, if one can go straight to k-maps.

a	b	c	d	F	F'
0	0	0	0	1	1
0	0	0	1	1	1
0	0	1	0	1	1
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	1	0	-
0	1	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	1	1
1	1	1	0	0	0
1	1	1	1	0	-



$F' = b'd' + c'd + a'd$

One implementation of this Function using 2-input Muxes:



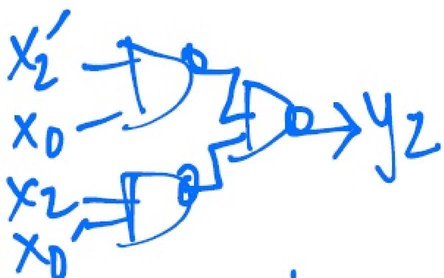
Problem 3: Design a black box that has a 3-bit input x, and a 3-bit output y. The function of the unit is y = (5x) % 8. Create the truth table and implement with the minimal gate logic in NAND-NAND and NOR-NOR for each output.

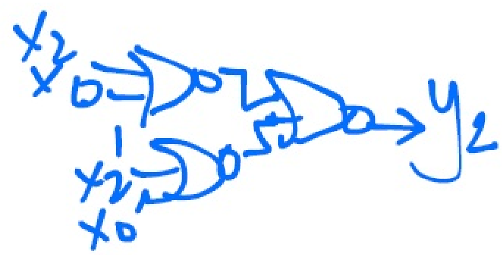
Solution:

Decimal x	(5x) mod 8 y	x_2	x_1	x_0	y_2	y_1	y_0
0	0	0	0	0	0	0	0
1	5	0	0	1	1	0	1
2	2	0	1	0	0	1	0
3	7	0	1	1	1	1	1
4	4	1	0	0	1	0	0
5	1	1	0	1	0	0	1
6	6	1	1	0	1	1	0
7	3	1	1	1	0	1	1

From the K-maps shown below we obtain:

Minterms and maxterms require same number of gates for y_2
$y_2 = x_2' x_0 + x_2 x_0'$
$y_2 = (x_2 + x_0)(x_2' + x_0')$
$y_1 = x_1$
$y_0 = x_0$





$x_1 = y_1$
$x_0 = y_0$

Problem 4: Design a system that takes a two digit base 3 number as input and produces the corresponding binary value (Z). (Example input: $12_3 = 0101_2$). Create the truth table and implement the 2 most significant bits using the minimal number of 2-input multiplexors (selector is 1-bit).
Solution:

Since the largest value of the input $22_3 = 8_{10}$, we need 3 bits to represent output.

a_1	a_0	b_1	b_0	y_3	y_2	y_1	y_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	0
0	0	1	1	-	-	-	-
0	1	0	0	0	0	1	1
0	1	0	1	0	1	0	0
0	1	1	0	0	1	0	1
0	1	1	1	-	-	-	-
1	0	0	0	0	1	1	0
1	0	0	1	0	1	1	1
1	0	1	0	1	0	0	0
1	0	1	1	-	-	-	-
1	1	0	0	-	-	-	-
1	1	0	1	-	-	-	-
1	1	1	0	-	-	-	-
1	1	1	1	-	-	-	-

From the K-maps shown below we obtain:

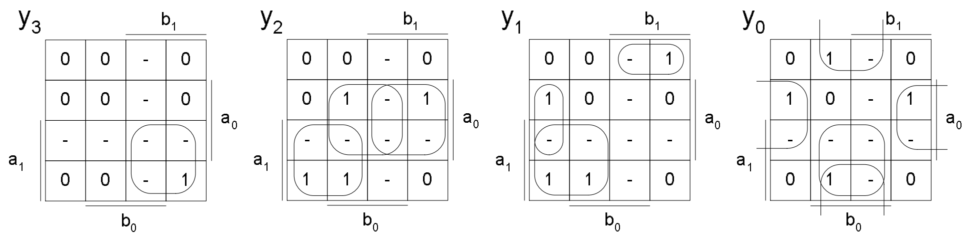
$y_3 = a_1 b_1$

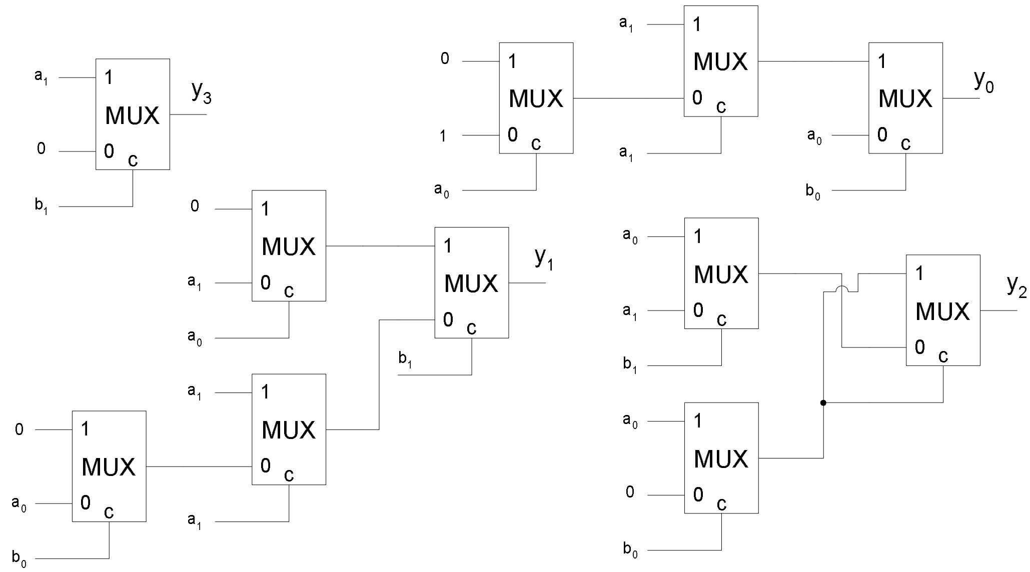
$y_2 = a_1\overline{b_1} + a_0 b_1 + a_0 b_0 = (a_1\overline{b_1} + a_0 b_1) + a_0 b_0$

$y_1 = a_1\overline{b_1} + a_0\overline{b_1}\,\overline{b_0} + a1\overline{a_0}b_1$
$= (a_1 + a_0\overline{b_0})\overline{b_1} + a1\overline{a_0}b_1$

$y_0 = a_1 b_0 + \overline{a_0} b_0 + a_0\overline{b_0} = (a_1 + \overline{a_0})b_0 + a_0\overline{b_0}$

Problem 5: Given two inputs $x = (x_1, x_0)$ and $y = (y_1, y_0)$, the output is the absolute value of their difference: $z = |2y - x|$. Use multiplexers to implement the function.





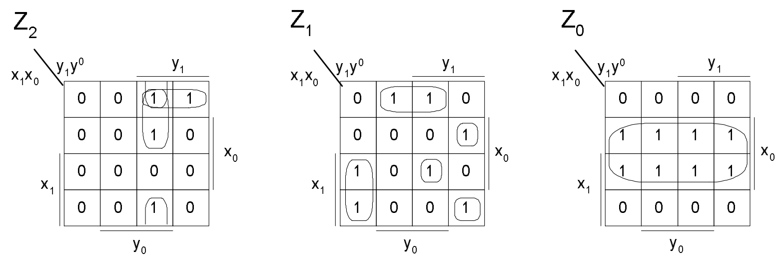
Solution:

Largest number for Y is 3, and smallest x is 0, therefore $|2 * 3 - 0| = 6$. The represent 6 we need at least 3 bits.

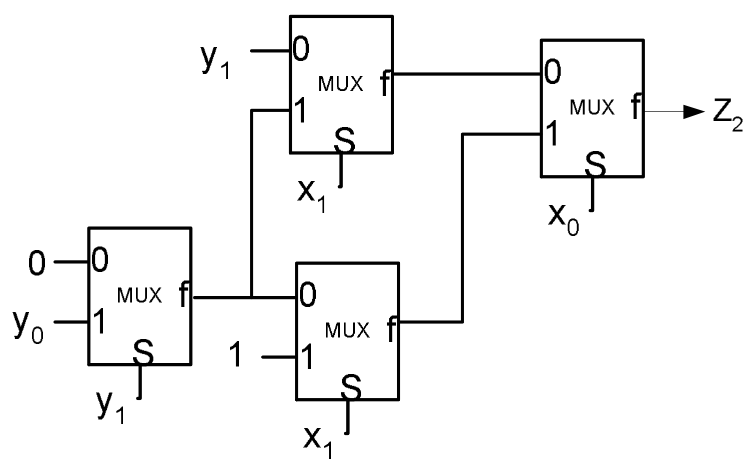
x_1	x_0	y_1	y_0	Z (decimal)	Z_2	Z_1	Z_0
0	0	0	0	0	0	0	0
0	0	0	1	2	0	1	0
0	0	1	0	4	1	0	0
0	0	1	1	6	1	1	0
0	1	0	0	1	0	0	1
0	1	0	1	1	0	0	1
0	1	1	0	3	0	1	1
0	1	1	1	5	1	0	1
1	0	0	0	2	0	1	0
1	0	0	1	0	0	0	0
1	0	1	0	2	0	1	0
1	0	1	1	4	1	0	0
1	1	0	0	3	0	1	1
1	1	0	1	1	0	0	1
1	1	1	0	1	0	0	1
1	1	1	1	3	0	1	1

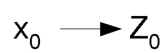
$Z_2 = x_0'y_1y_0 + x_1'y_1y_0 + x_1'x_0'y_1$
$Z_1 = x_1'x_0'y_0 + x_1y_1'y_0' + x_1x_0y_1y_0 + x_1'x_0y_1y_0' + x_1x_0'y_1y_0'$
$Z_0 = x_0$

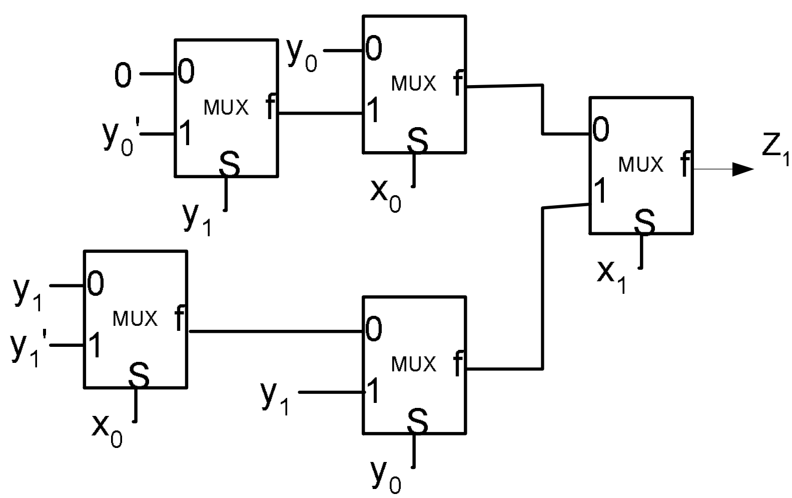
Z_2 :



$x_0' y_1 y_0 + x_1' y_1 y_0 + x_1' x_0' y_1$
$x_0(x_1' y_1 y_0) + x_0'(y_1 y_0 + x_1' y_1)$
$x_1' y_1 y_0 \Rightarrow x_1(1) + x_1'(y_1 y_0)$
$y_1 y_0 + x_1' y_1 \Rightarrow x_1(y_1 y_0) + x_1'(y_1 + y_1 y_0) = x_1(y_1 y_0) + x_1'(y_1)$
$Z_1:$
$x_1' x_0' y_0 + x_1 y_1' y_0' + x_1 x_0 y_1 y_0 + x_1' x_0 y_1 y_0' + x_1 x_0' y_1 y_0'$
$x_1(x_0 y_1' y_0' + y_1 d + x_0' y_1 y_0') + x_1'(x_0' y_0 + x_0 y_1 y_0')$
$x_0 y_1' y_0' + y_1 y_0 + x_0' y_1 y_0' \Rightarrow y_0(y_1) + y_0'(x_0 y_1' + x_0' y_1)$
$x_0 y_1' + x_0' y_1 \Rightarrow x_0(y_1') + x_0'(y_1)$







Problem 6: A circuit will either subtract X from Y or Y from X, depending on the value of A. If A = 1, the output should be $|X - Y|$, and if A = 0, the output should be $|Y - X|$. Design a combinational system for the output using as few multiplexers as possible.

Solution:

A	X_1	X_0	Y_1	Y_0	$\|X-Y\|$ or $\|Y-X\|$	Z_1	Z_0
0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	1
0	0	0	1	0	2	1	0
0	0	0	1	1	3	1	1
0	0	1	0	0	1	1	0
0	0	1	0	1	0	0	0
0	0	1	1	0	1	0	1
0	0	1	1	1	2	1	0
0	1	0	0	0	1	0	1
0	1	0	0	1	1	0	1
0	1	0	1	0	0	0	0
0	1	0	1	1	1	0	1
0	1	1	0	0	3	1	1
0	1	1	0	1	2	1	0
0	1	1	1	0	1	0	1
0	1	1	1	1	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	1	1	0	1
1	0	0	1	0	2	1	0
1	0	0	1	1	3	1	1
1	0	1	0	0	1	0	1
1	0	1	0	1	0	0	0
1	0	1	1	0	1	0	1
1	0	1	1	1	2	1	0
1	1	0	0	0	2	1	0
1	1	0	0	1	1	0	1
1	1	0	1	0	0	0	0
1	1	0	1	1	1	0	1
1	1	1	0	0	3	1	1
1	1	1	0	1	2	1	0
1	1	1	1	0	1	0	1
1	1	1	1	1	0	0	0

The output is the same wether or not A is 1 or 0. For this reason, the output is not dependant on A. Therefore, we can disregard A in the rest of the design.

$z_1 = x_1x_0y_1' + x_1'y_1y_0 + x_1'x_0'y_1 + y_1'y_0'x_0$

$z_0 = x_1y_1'y_0 + x_0y_1y_0' + x_0'y_0$

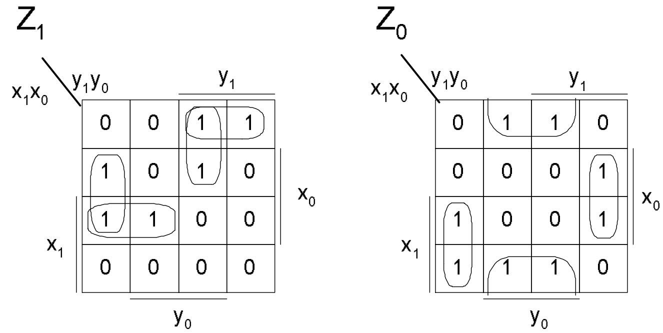
z_1 :

$x_1(x_0y_1') + x_1'(y_1y_0 + x_0'y_1)$

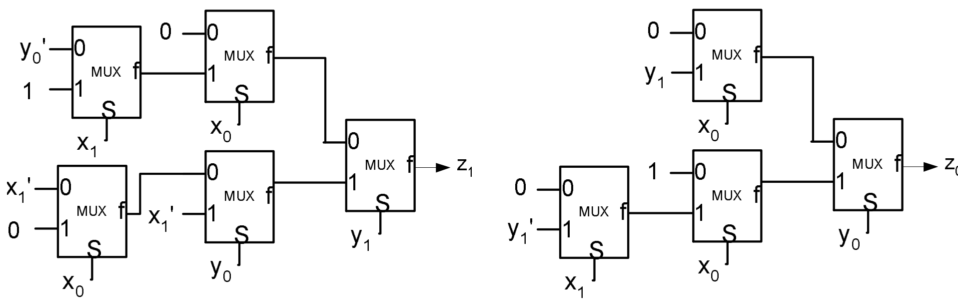
$y_1y_0 + x_0'y_1 \Rightarrow y_1(y_0 + x_0') + y_1'(0)$

z_0 :

$y_0(x_1y_1' + x_0') + y_0'(x_0y_1)$



$x_1y_1' + x_0' \Rightarrow x_0'(1) + x_0(x_1y_1')$

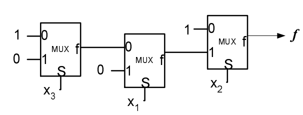


Problem 7: Implement the following expression using multiplexers (Only uncomplemented variables are given.): $f = x_1'x_2' + x_3'x_2x_1' + x_1x_2'$

Solution:

$f = x_2'(x_1' + x_1) + x_3'x_2x_1'$
$f = x_2' + x_3'x_2x_1'$

$f = x_2'(1) + x_2(x_3'x_1')$



Problem 8: Implement the following function using (a) an 8-input multiplexor (3-selector), (b) a 4-input multiplexor (2-selector) and NOR gates

$$Z = a'b'c'd + a'bc'd' + a'b'cd + ab'c'd + abcd + abcd'$$

Solution:

(a):

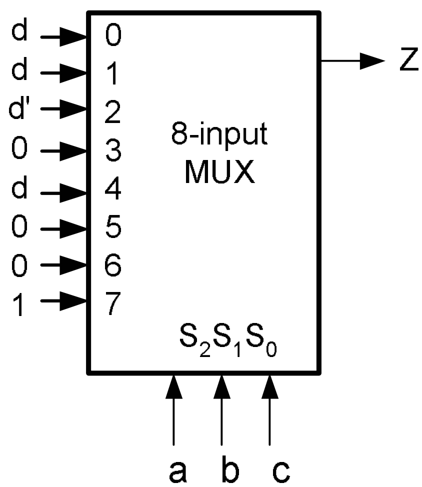
Select 3 variables to be the selector variables: abc

$Z = a'b'c'(d) + a'b'c(d) + a'bc'(d') + a'bc(0) + ab'c'(d) + ab'c(0) + abc'(0) + abc(d + d')$

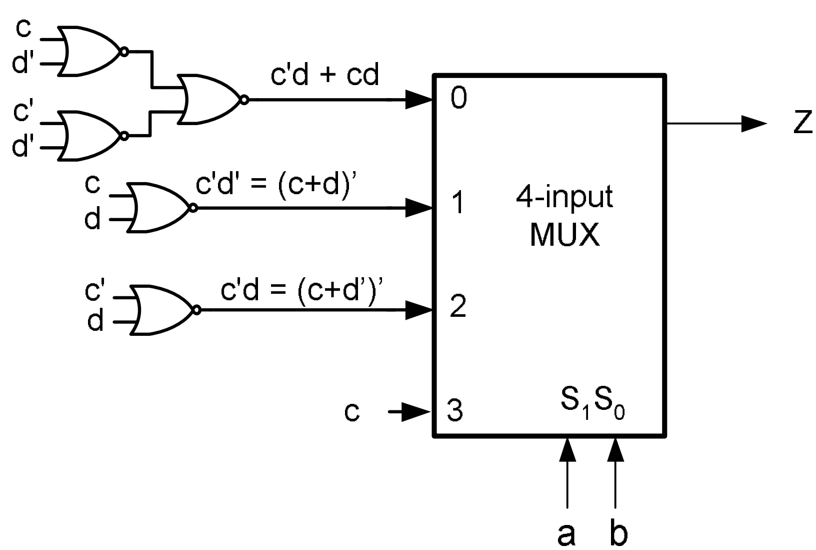
(b):

Select 2 variables to be the selector variables: ab

$Z = a'b'(c'd + cd) + a'b(c'd') + ab'(c'd) + ab(cd + cd')$
$Z = a'b'(c'd + cd) + a'b(c'd') + ab'(c'd) + ab(c)(d + d')$
$Z = a'b'(c'd + cd) + a'b(c'd') + ab'(c'd) + ab(c)$


d
d
d'
0
d
0
0
1
0
1
2
3
4
5
6
7
8-input
MUX
S2S1S0
Z
a
b
c

(a)


c
d'
c'
d'
c'd + cd
c
d
c'd' = (c+d)'
c'
d
c'd = (c+d')'
c
0
1
2
3
4-input
MUX
S1S0
Z
a
b

(b)