



EECE256 Assignment 7

1. How many address lines and data lines are needed for the following memory?

	Address	Data
a) 16K x 8	14	8
b) 256K x 64	18	64
c) 2G x 16	31	16
d) 20 x 4	5	4

2. Show the Parity bits for a 16-bit and 32-bit number. What is the parity value for the 8-bit binary number 10110101? Place the bits into the appropriate places in the parity+data word.

$P_1 P_2 D_{15} P_4 D_{14} D_{13} D_{12} P_8 D_{11} D_{10} D_9 D_8 D_7 D_6 D_5 P_{16} D_4 D_3 D_2 D_1 D_0$

$P_1 P_2 D_{31} P_4 D_{30} D_{29} D_{28} P_8 D_{27} D_{26} D_{25} D_{24} D_{23} D_{22} D_{21} P_{16} D_{20} D_{19} D_{18} D_{17} D_{16} D_{15} D_{14} D_{13} D_{12} D_{11} D_{10} D_9 D_8 D_7 D_6 P_{32} D_5 D_4 D_3 D_2 D_1 D_0$

$P_1 P_2 1 P_4 011 P_8 0101$

$P_1 = \text{XOR}(3,5,7,9,11) = 1$

$P_2 = \text{XOR}(3,6,7,10,11) = 1$

$P_3 = \text{XOR}(5,6,7,12) = 1$

$P_4 = \text{XOR}(9,10,11,12) = 0$

111101100101



3. Create the truth table for a ROM that implements the Boolean functions:

$$A(x,y,z) = \sum(0,3,6)$$

$$B(x,y,z) = \sum(1,2,3,6)$$

$$C(x,y,z) = \sum(1,5)$$

$$D(x,y,z) = \sum(0,1,5,6,7)$$

How big should the ROM be? What is the memory content at addresses 0 and 5?

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

Address 0 = 1101

Address 5 = 0011



4. Derive the PLA programming table for a combinational circuit that squares a 4-bit number, and minimize the number of product terms. What is the width of the output?

WX yZ	WXYZ	$S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0$
0000	0	00000000
0001	1	00000001
0010	4	00000100
0011	9	00001001
0100	16	00010000
0101	25	00010001
0110	36	00100100
0111	49	00110001
1000	64	01000000
1001	81	01010001
1010	100	01100100
1011	121	01110001
1100	144	10010000
1101	169	10100001
1110	196	11000100
1111	225	11100001

$S_0 = Z, S_1 = 0$
 $S_2 = \Sigma(2, 6, 10, 14)$
 $S_3 = \Sigma(3, 5, 11, 13)$
 $S_4 = \Sigma(4, 5, 7, 9, 11, 12)$
 $S_5 = \Sigma(6, 7, 10, 11, 13, 15)$
 $S_6 = \Sigma(8, 9, 10, 11, 14, 15)$
 $S_7 = \Sigma(12, 13, 14, 15)$



K Map

yz

wx	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

$S_2 = yz'$
 $S_2' = y' + z$

yz

wx	00	01	11	10
00	0	0	1	0
01	0	1	0	0
11	0	0	0	0
10	0	0	1	0

$S_3 = xy'z + x'yz$
 $S_3' = z' + xy + x'y'$

yz

wx	00	01	11	10
00	0	0	0	0
01	1	1	0	0
11	1	0	0	0
10	0	1	0	0

$S_4 = wxz + wx'z + xy$
 $S_4' = wx' + x'z + yz' + w$

yz

wx	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	0	0	0	0
10	0	0	1	1

$S_5 = xz + wx'y$
 $S_5' = y'z' + wx' + xz + x'y'$

yz

wx	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	1	1
10	1	1	1	1

$S_6 = wy + wx'$
 $S_6' = w' + xy'$

yz

wx	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	0	0

$S_7 = wx$
 $S_7' = x' + w'$

* look for common terms, & pick the min. ^(total # of terms)
 in this case all functions are true ~~together~~
 or S_4 can be complemented for the same # of terms

Product Term	Inputs		Outputs					
	w	x y z	S_2	S_3	S_4	S_5	S_6	S_7
yz'	-	- 1 0	1	-	1	-	-	-
$x'y'z$	-	1 0 1	-	1	-	-	-	-
$x'y'z$	-	0 1 1	-	1	-	-	-	-
$w'x'$	0	0 - -	-	-	1	-	-	-
$x'z'$	-	0 - 0	-	-	1	-	-	-
xy	1	1 1 1	-	-	1	1	-	-
wxy	-	1 - 1	-	-	1	-	-	-
xz	1	0 1 -	-	-	-	1	-	-
$wx'y$	1	0 1 -	-	-	-	1	-	-
wxy	1	0 - -	-	-	-	1	-	-
wx	1	1 - -	-	-	-	1	-	1

It was possible to implement the square function with a PAL, but not very easy. Think of how easy this is to implement using a ROM. Selection of appropriate device types is important!