EE 231

Exam 4

December 8, 2008

Name: _____

Show all work. Partial credit will be given. No credit will be given if an answer appears with no supporting work.

1. How many address and data lines are needed for the following memory units? How many bytes of memory does each unit hold?

   (a) 256M x 16
   256M = $256 \times 1024 \times 1024 = 2^8 \times 2^{10} \times 2^{10} = 2^{28}$ => 28 address lines, 16 data lines.
   16 bits = 2 bytes, so 256M x 16 = 256M x 2 bytes = 512M bytes

   (b) 4K x 32
   4K = $4 \times 1024 = 2^2 \times 2^{10} = 2^{12}$ => 12 address lines, 32 data lines.
   32 bits = 4 bytes, so 4K x 32 = 4K x 4 bytes = 16K bytes

2. The following two problems deal with the Hamming code for error detection and correction. For (a) and (b), the numbers are of the form $P_1\ P_2\ D_3\ P_4\ D_5\ D_6\ D_7\ P_8\ D_9\ D_{10}\ D_{11}\ D_{12}\ P_{13}$, where P_{13} is the overall parity bit.

   (a) Consider the binary number 01011010_2. Generate the Hamming code for the number which will allow you to correct one-bit errors and detect two-bit errors.

P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8	D_9	D_{10}	D_{11}	D_{12}	P_{13}
X	X	0	X	1	0	1	X	1	0	1	0	X

   $P_1 = \text{XOR}(D_3, D_5, D_7, D_9, D_{11}) = \text{XOR}(0, 1, 1, 1, 1) = 0$ (Even number of 1's)
   $P_2 = \text{XOR}(D_3, D_6, D_7, D_{10}, D_{11}) = \text{XOR}(0, 0, 1, 0, 1) = 0$ (Even number of 1's)
   $P_4 = \text{XOR}(D_5, D_6, D_7, D_{12}) = \text{XOR}(1, 0, 1, 0) = 0$ (Even number of 1's)
   $P_8 = \text{XOR}(D_9, D_{10}, D_{11}, D_{12}) = \text{XOR}(1, 0, 1, 0) = 0$ (Even number of 1's)
   $P_{13} = \text{XOR}(P_1, P_2, D_3, P_4, D_5, D_6, D_7, P_8, D_9, D_{10}, D_{11}, D_{12}$
   $P_{13} = \text{XOR}(0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0) = 0$ (Even number of 1's)

P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8	D_9	D_{10}	D_{11}	D_{12}	P_{13}
0	0	0	0	1	0	1	0	1	0	1	0	0

   Data out is 0 0001 0101 0100.

(b) You read the number 1 0110 0101 1100 from a memory which uses error detection and correction. What was the original 8-bit data word which was written to memory?

P_1	P_2	D_3	P_4	D_5	D_6	D_7	P_8	D_9	D_{10}	D_{11}	D_{12}	P_{13}
1	0	1	1	0	0	1	0	1	1	1	0	0

$C_1 = \text{XOR}(P_1, D_3, D_5, D_7, D_9, D_{11}) = \text{XOR}(1, 1, 0, 1, 1, 1) = 1$ (Odd number of 1's)
$C_2 = \text{XOR}(P_2, D_3, D_6, D_7, D_{10}, D_{11}) = \text{XOR}(0, 1, 0, 1, 1, 1) = 0$ (Even number of 1's)
$C_4 = \text{XOR}(P_4, D_5, D_6, D_7, D_{12}) = \text{XOR}(1, 0, 0, 1, 0) = 0$ (Even number of 1's)
$C_8 = \text{XOR}(P_8, D_9, D_{10}, D_{11}, D_{12}) = \text{XOR}(0, 1, 1, 1, 0) = 1$ (Odd number of 1's)
$C_{13} = \text{XOR}(P_1, P_2, D_3, P_4, D_5, D_6, D_7, P_8, D_9, D_{10}, D_{11}, D_{12}, P_{13})$
$C_{13} = \text{XOR}(1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 0) = 1$ (Odd number of 0's)
$C_8C_4C_2C_1 = 1001_2 = 9_{10}$, and $C_{13} = 1$ => one bit error in bit D_9, so bit D_9 was flipped, and the original data was:

D_3	D_5	D_6	D_7	D_9	D_{10}	D_{11}	D_{12}
1	0	0	1	0	1	1	0

Data was 1001 0110.

(c) How many bits are needed to correct one-bit errors and detect two-bit errors for a 24-bit data word? Explain.

Need parity bits at all powers of two (P_1, P_2, P_8, P_{16}, P_{32}, etc.), with data bits in the other places. Keep adding data bits and parity bits until you get 24 data bits:
P_1 P_2 D_3 P_4 D_5 D_6 D_7 P_8 D_9 D_{10} D_{11} D_{12} D_{13} D_{14} D_{15} P_{16} D_{17} D_{18} D_{19} D_{20} D_{21} D_{22} D_{23} D_{24} D_{25} D_{26} D_{27} D_{28} D_{29}
gives 24 data bits and 5 parity bits. Add one more parity bit for 2-bit error detection, and you need a total of 30 bits – 24 data bits and 6 check bits.

3. Assume that $A = 4'b1011$, $B = 4'b1101$ and $C = 4'b0000$. Show the results of the following Verilog operations:

(a) `A + B` 1011 + 1101 = 11000 (note a carry bit is added)

(b) `A - B` 11110 (note a borrow bit was added)

(c) `A & B` 1011 & 1101 = 1001

(d) `A && B` 1 (both numbers non-zero)

(e) `A | C` 1011 | 0000 = 1011

(f) `A || C` 1 (one of the two numbers non-zero)

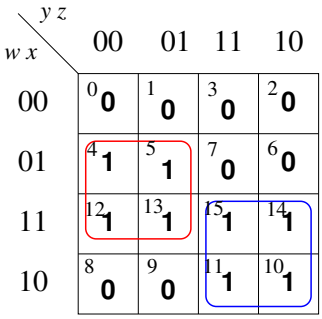
(g) `A >> 2` 1011 >> 2 = 0010 (logical shift – bits 3:2 shifted to 1:0, 0 shifted into 3:2)

(h) `A >>> 2` 1011 >>> 2 = 1110 (arithmetic shift – bits 3:2 shifted to 1:0, MSB shifted into 3:2)

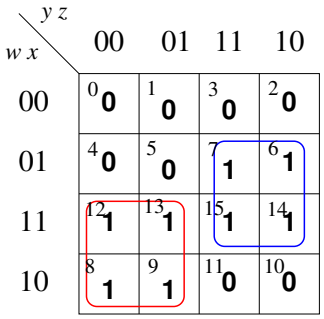
4. Show the PAL fuse map for implementing the following functions. Be sure to explain your work.

   $A(w,x,y,z) = \sum(4,5,10,11,12,13,14,15)$
   $B(w,x,y,z) = \sum(6,7,8,9,12,13,14,15)$
   $C(w,x,y,z) = \sum(0,2,7,8,9,10,12,13,14,15)$

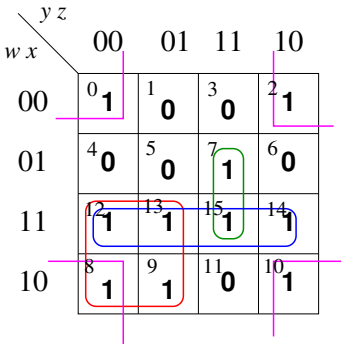
   Solution:



A = xy′ + wy



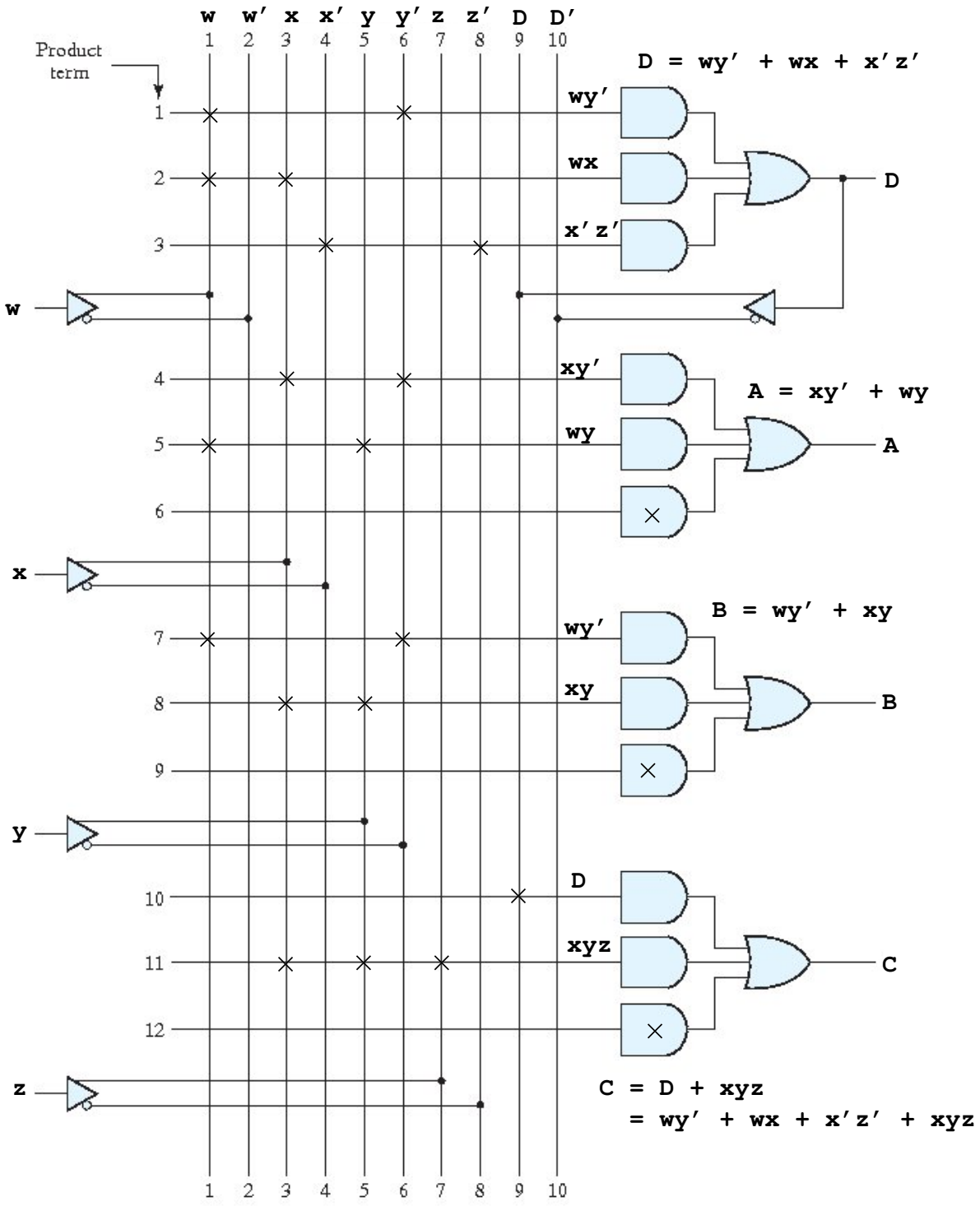
B = wy′ + xy



C = wy′ + wx + x′z′ + xyz

Each output function of the PAL only has three input OR gates, but C needs four terms to be ORed together. Write $C = D + xyz$, where $D = wy' + wx + x'z'$, and feed D back into PAL array to generate C.

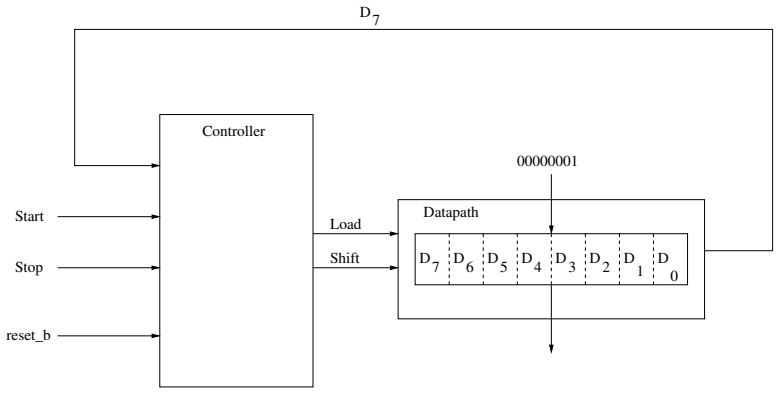
Problem 7.25


w w' x x' y y' z z' D D'
1 2 3 4 5 6 7 8 9 10
Product term
D = wy' + wx + x'z'
wy'
wx
x'z'
D
w
xy'
wy
A = xy' + wy
A
x
B = wy' + xy
wy'
xy
B
y
D
xyz
C
z
C = D + xyz
= wy' + wx + x'z' + xyz

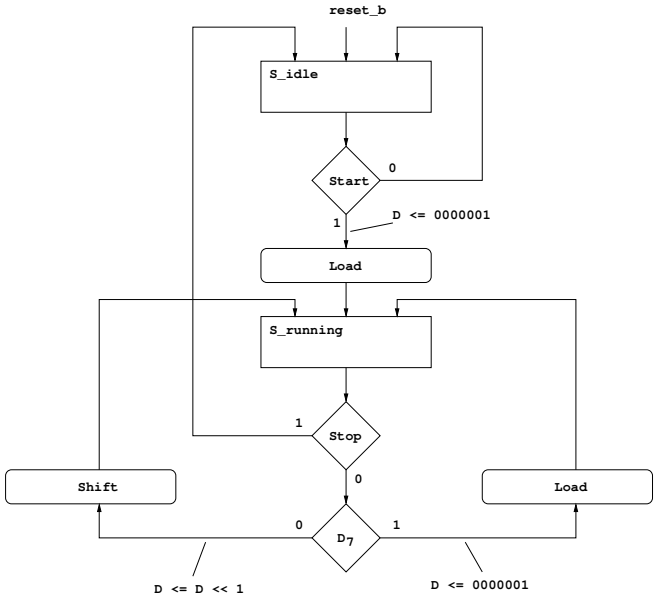
5. The figure below shows the controller and the datapath for a digital circuit. The `Load` signal loads 0000001 into Register D. The `Shift` signal shifts Register D left by 1. The circuit is supposed to do the following: `reset_b` is an asynchronous reset, which puts the system into the `S_idle` state. The system remains in the `S_idle` state until the controller sees the `Start` signal go high. When `Start` goes high, the system loads the 00000001 into Register D and goes to the `S_running` state. After that, the system shifts Register D left until the 1 is in bit D_7, at which point the system will reload D with 00000001. It will continue doing this until `Stop` goes high. When `Stop` goes high, the system returns to the `S_idle` state. The system will generate the following pattern, a ring counter in which a single 1 rotates through the bits:

   00000001, 00000010, 00000100, 00001000, 00010000, 00100000, 01000000, 10000000, 00000001, 00000010,

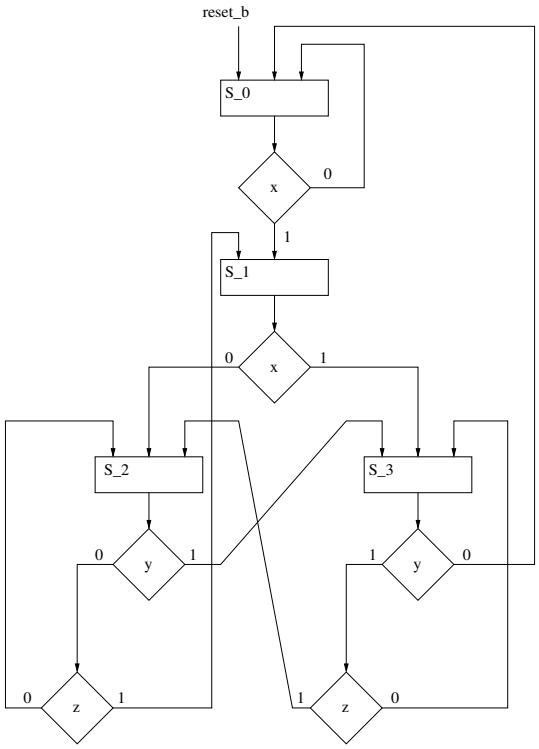
   Draw an ASMD chart for this circuit.



Solution:



6. Consider the following ASMD chart. Draw the state transition diagram for the controller.



Solution: