Assignment 4

Jonathon Mesery

1. a)

W	X	Y	Z	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

b)

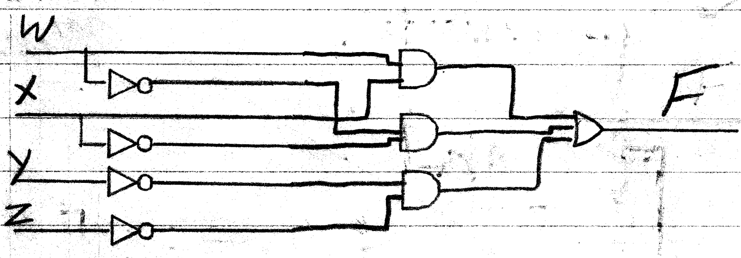
wx \ yz	00	01	11	10	
00	1	1	1	1	← $\overline{w}\,\overline{x}$
01	1	0	0	0	
11	1	1	1	1	← wx
10	1	0	0	0	

↖ $\overline{y}\,\overline{z}$

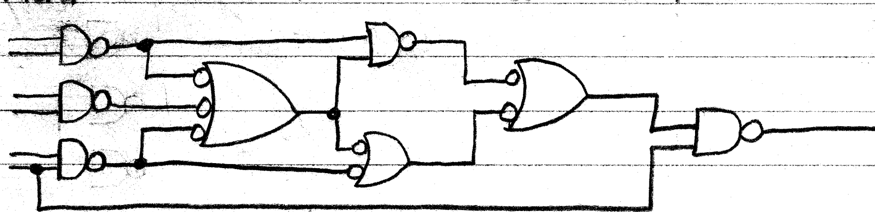
c) See Above

d) $F = wx + \overline{w}\,\overline{x} + \overline{y}\,\overline{z}$

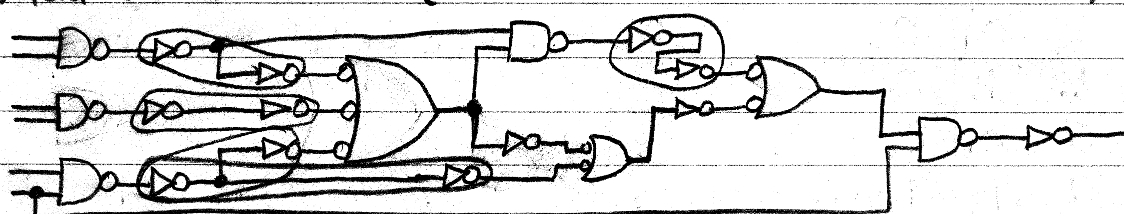
1e)



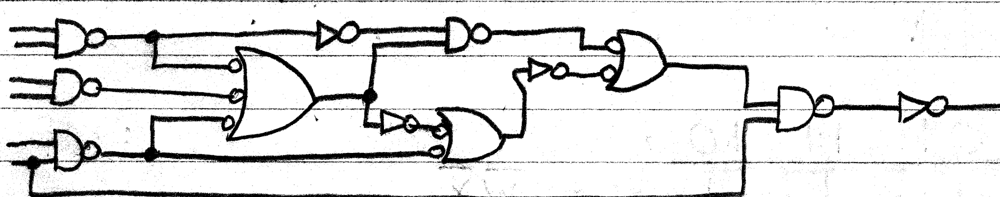
2 Add Bubbles



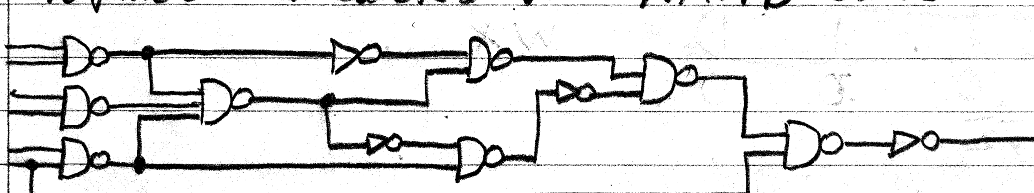
Add NOT Gates (Circled NOT Gates cancel)



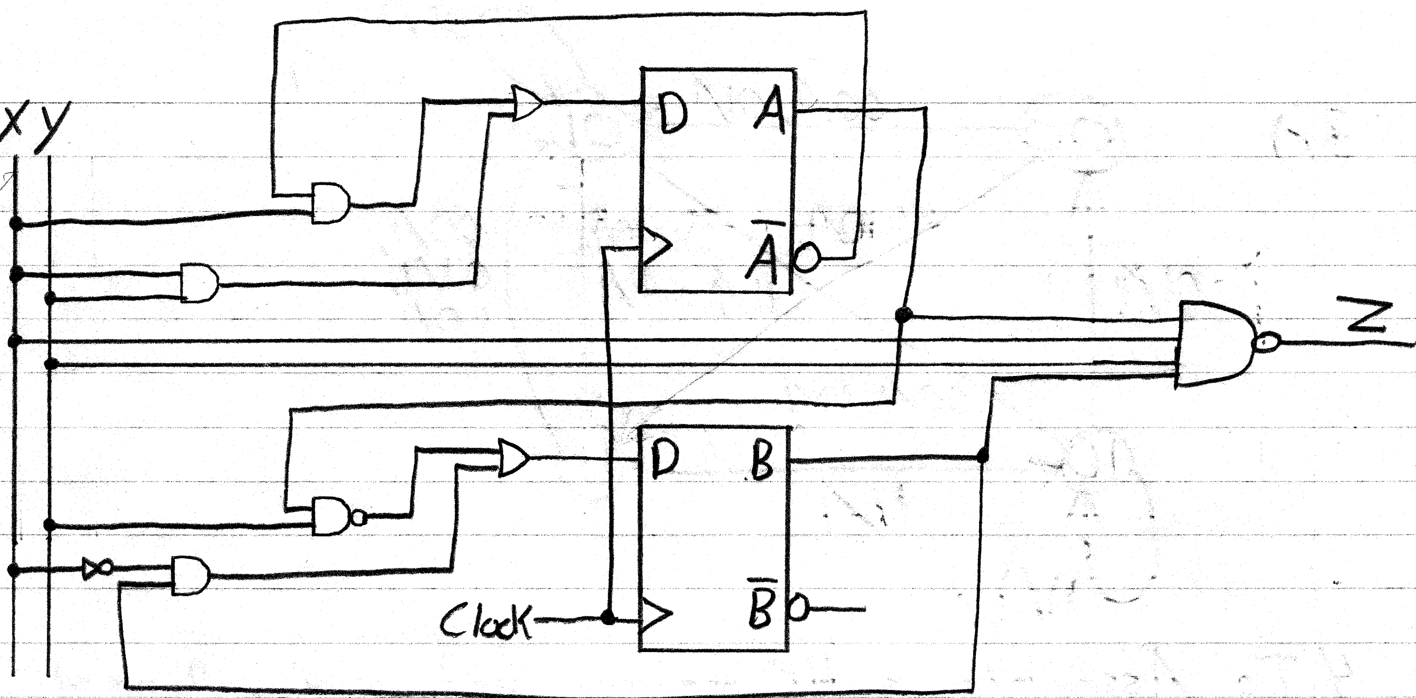
Eliminated NOT Gates



Replace OR Gatetcs with NAND Gates



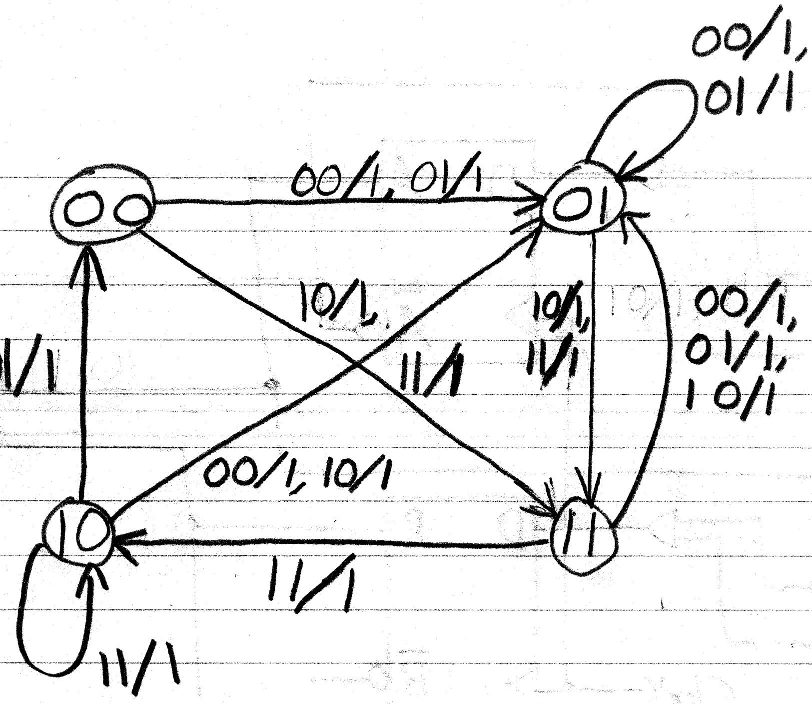
3a)



b)

A(t)	B(t)	x	y	A(t+1)	B(t+1)	Z
0	0	0	0	0	1	1
0	0	0	1	0	1	1
0	0	1	0	1	1	1
0	0	1	1	1	1	1
0	1	0	0	0	1	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
0	1	1	1	1	1	1
1	0	0	0	0	1	1
1	0	0	1	0	0	1
1	0	1	0	0	1	1
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	0	1	1
1	1	1	0	0	1	1
1	1	1	1	1	0	0

3c)



4 To Assemble a 4K memory unit you will need 128 16x16 chips. Math: a 4k chip is also 16 x 2048 therefore 2048/16 = 128 16 row chips.
7 bits will be reserved for the chip which will go into a 7:128 decoder. Since each chip is 16 rows we'll need another 4 bits to get the row in the selected address, thus the total address length is 11 bits, this is also found $\log_2(2048) = 11$

5 Suppose you write 2 instructions and the second one uses data related to the First one. Suppose also they are in a pipeline but the First instruction takes longer than the second, and as said the second uses data related to the first. IF the First instruction isn't done the second may do something wrong. IF you put NOP (or multiple) between the two It would add clock cycles inbetween and the second instruction would enter the pipeline later and the First instruction would have time to Finish

6 The main problem with direct-mapped cache is when a block of main memory is repeatedly accessed leading to a cycle of eviction and replacement of the same cache block. A fully associative cache will have no mapping so any memory block can go in any cache block. When something in the cache must be evicted a replacment policy must be invoked. A set-associative cache behaves like a hash table. The cache blocks are grouped into sets and memory blocks are mapped to specific sets. When a cache block set fills up and something needs to be evicted a replacement policy is used. Some replacement policies include Least Recently Used, FIFO Queue, and random replacement.

7 a) $EAT = (H_c \cdot A_c) + ((1-H_c) \cdot ((H_{mm} \cdot A_{mm} \cdot 2) + ((1-H_{mm}) \cdot A_{vm})))$

$2us = 2000ns$

b) $EAT = (0.96 \cdot 4) + (0.04 \cdot ((0.92 \cdot 8 \cdot 2) + (0.08 \cdot 2000)))$

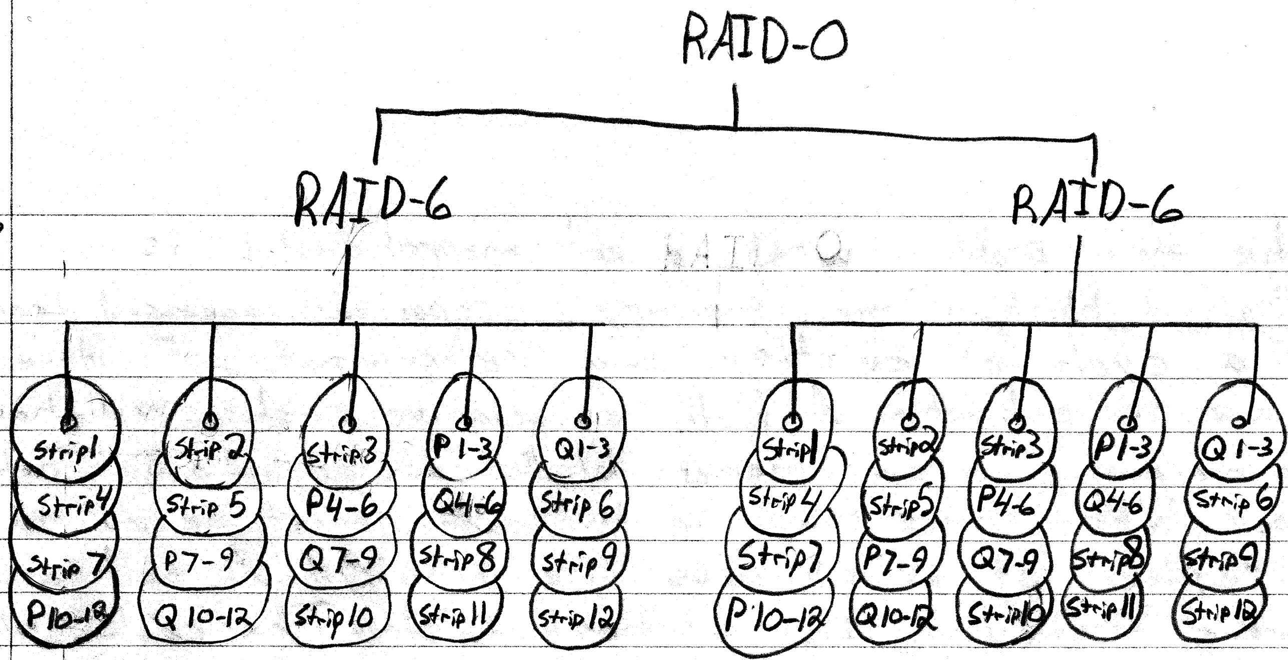
$= 3.84 + (0.04 \cdot (14.72 + 160)$

$= 3.84 + (0.04 \cdot 174.72)$

$= 3.84 + 6.9888$

$= 10.8288\ ns$

8



P: Parity
Q: Reed-Solomon