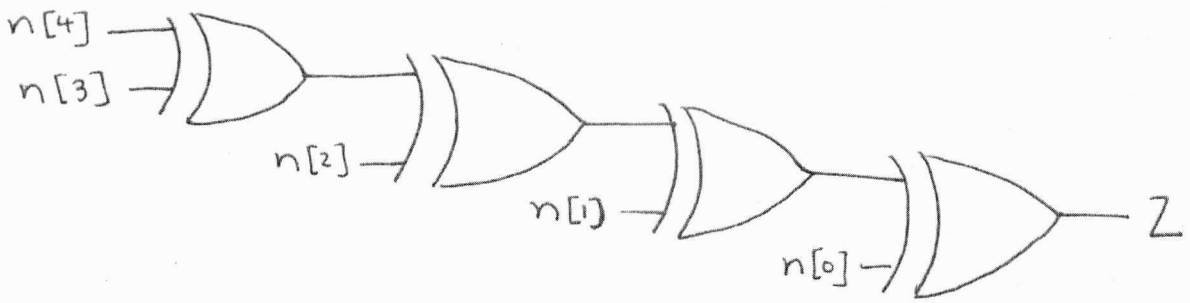
EE 306 Fall 2008

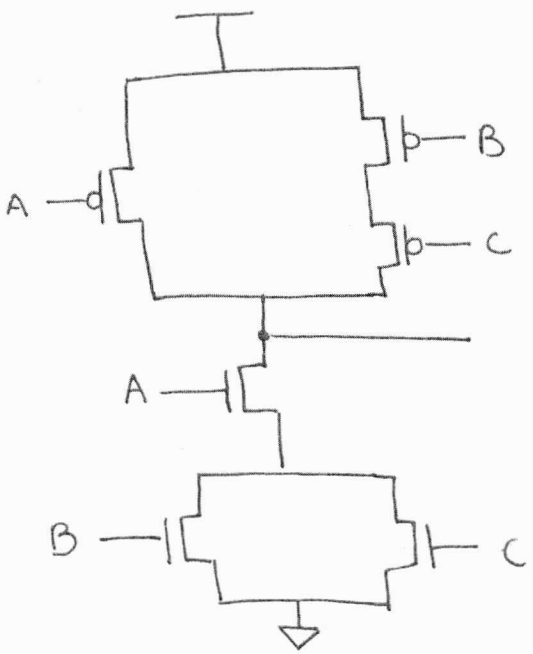
Problem Set 2 Solutions

1. A series of XOR gates can be used to test for an odd number of 1s.



Also refer to problem 14 in this problem set.

2. NOT(A AND (B OR C))



3. (NOT (B) AND C) OR A

ALTERNATIVE SOLUTION

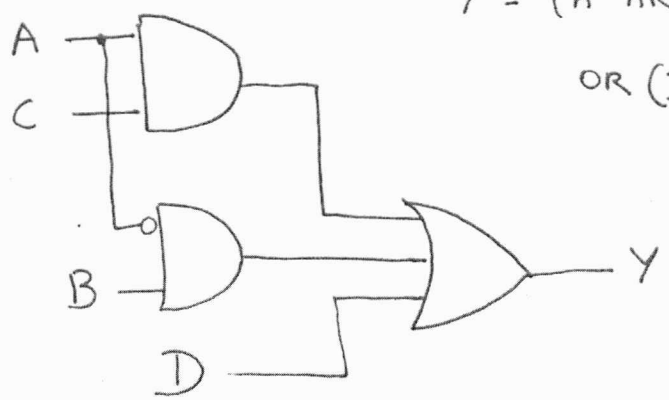
((NOT A) AND (NOT B) AND C) OR (A AND (NOT B) AND (NOT C))

OR (A AND (NOT B) AND C) OR (A AND B AND (NOT C))

OR (A AND B AND C)

This can be figured by drawing out a truth table for the transistor ~~dio~~ circuit.

4. ~~A=~~



Y = (A AND C) OR (NOT A AND B) OR (D)

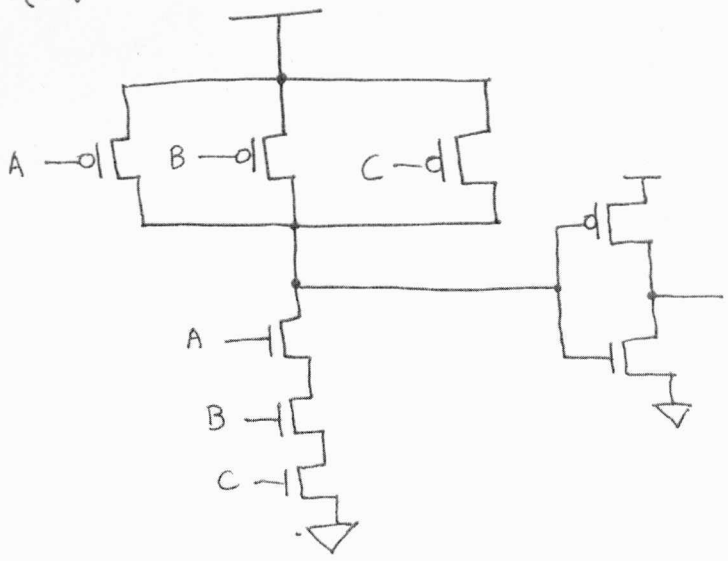
5.

A	B	C	OUT
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

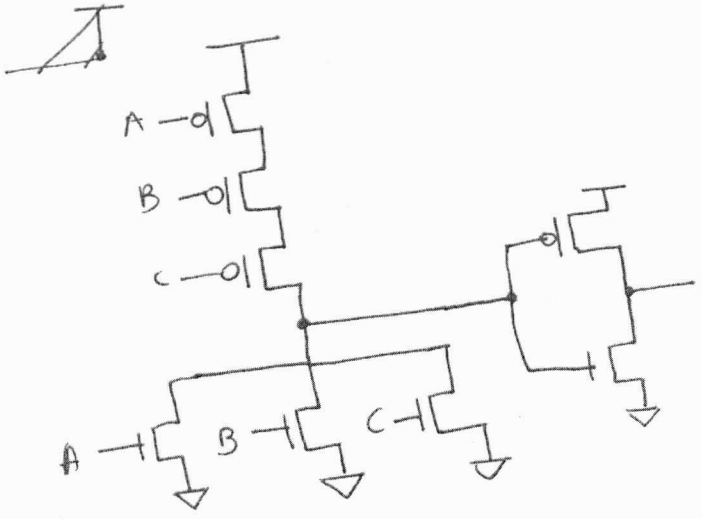
A 3-input OR gate has the same truth table.

6.

(a)



Three-input AND gate.



Three-input OR gate

(b) Refer to fig 3.5 (d) and fig. 3.6 (b).

7.

a) $2^5 = 32$ output lines

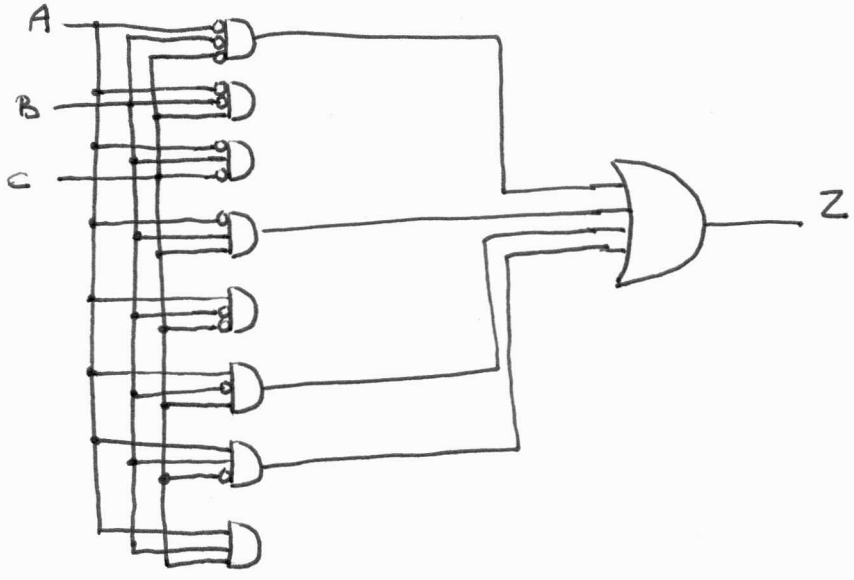
b) $2^8 = 256$ output lines

c) 2^n output lines

8. Truth table is:

A	B	C	Z
0	0	0	**1**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

So, there are 4 rows where Z is equal to 1, which means that we will need one 4 input OR gate. Each row can be represented with one 3 input AND gate. The logical equation for this gate level logic circuit is: A'B'C'+A'BC+AB'C+ABC'.



9.

A) The output of the circuit when select line S is 0 is equal to the input A, that is for A=1, output is 1 and for A=0 output is 0.

B) If the S switches from 0 to 1, the output depends on the last propagated value of A, i.e. if the previous output was 0 the output is going to stay 0, and if the previous output was 1, the new output is going to be 1.

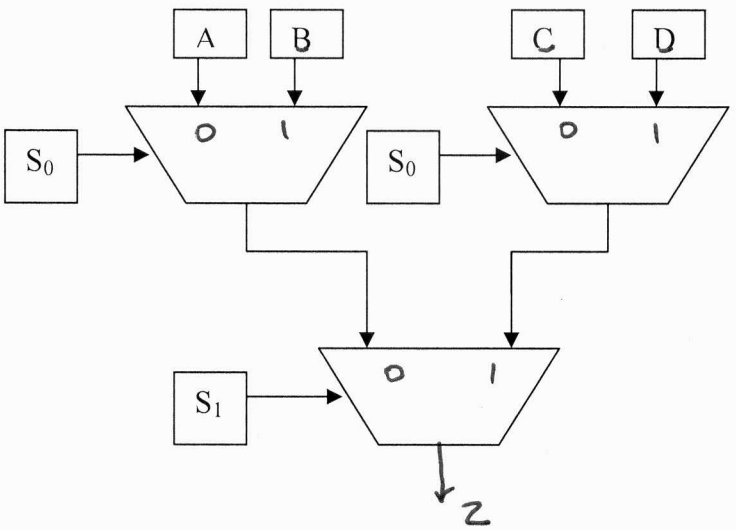
c) This is a storage element, since it stores the value that A had in the moment when S changed from 0 to 1.

10.

For implementation of a 4-to-1 mux we need to understand what is the 4-to-1 mux doing. The idea is that with 2 selection bits we can distinguish 4 different inputs in the multiplexor and by assigning the values to the selection bits we are building a programmable switch that connects output to one of the 4 inputs. The short table for representing this is:

S1	S0	Z
0	0	**A**
0	1	**B**
1	0	**C**
1	1	**D**

If we look at the table more closely, we can see that selection bit S1 makes the choice will one of the signals in the group (A,B) be on the output or a signal from group (C,D). In each of this groups signal S0 makes the precise decision which one will be used. This is presented on the figure



So, the output values are:

S1	S0	A	B	C	D	OUT
0	0	0	0	0	0	**0**
0	0	0	0	0	1	**0**
0	0	0	0	1	0	**0**
0	0	0	0	1	1	**0**
0	0	0	1	0	0	**0**
0	0	0	1	0	1	**0**
0	0	0	1	1	0	**0**
0	0	0	1	1	1	**0**
0	0	1	0	0	0	**1**
0	0	1	0	0	1	**1**
0	0	1	0	1	0	**1**
0	0	1	0	1	1	**1**
0	0	1	1	0	0	**1**
0	0	1	1	0	1	**1**
0	0	1	1	1	0	**1**

S1	S0	A	B	C	D	OUT
1	0	0	0	0	0	**0**
1	0	0	0	0	1	**0**
1	0	0	0	1	0	**1**
1	0	0	0	1	1	**1**
1	0	0	1	0	0	**0**
1	0	0	1	0	1	**0**
1	0	0	1	1	0	**1**
1	0	0	1	1	1	**1**
1	0	1	0	0	0	**0**
1	0	1	0	0	1	**0**
1	0	1	0	1	0	**1**
1	0	1	0	1	1	**1**
1	0	1	1	0	0	**0**
1	0	1	1	0	1	**0**
1	0	1	1	1	0	**1**

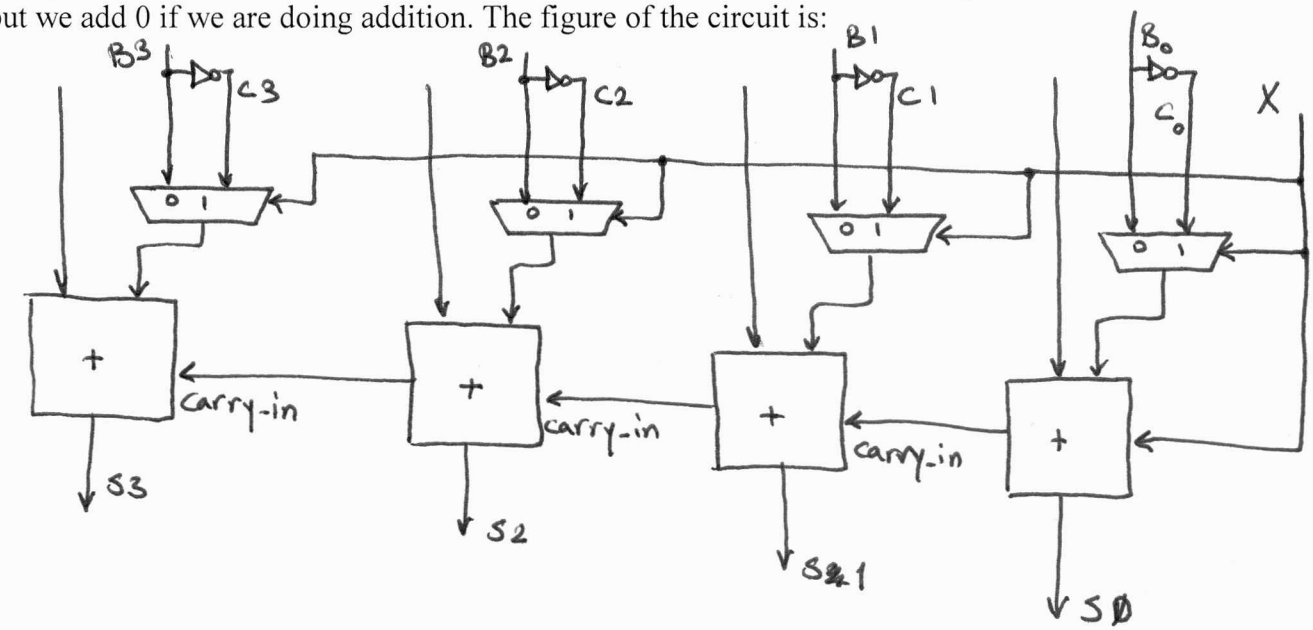
0	0	1	1	1	1	**1**
0	1	0	0	0	0	**0**
0	1	0	0	0	1	**0**
0	1	0	0	1	0	**0**
0	1	0	0	1	1	**0**
0	1	0	1	0	0	**1**
0	1	0	1	0	1	**1**
0	1	0	1	1	0	**1**
0	1	0	1	1	1	**1**
0	1	1	0	0	0	**0**
0	1	1	0	0	1	**0**
0	1	1	0	1	0	**0**
0	1	1	0	1	1	**0**
0	1	1	1	0	0	**1**
0	1	1	1	0	1	**1**
0	1	1	1	1	0	**1**
0	1	1	1	1	1	**1**

1	0	1	1	1	1	**1**
1	1	0	0	0	0	**0**
1	1	0	0	0	1	**1**
1	1	0	0	1	0	**0**
1	1	0	0	1	1	**1**
1	1	0	1	0	0	**0**
1	1	0	1	0	1	**1**
1	1	0	1	1	0	**0**
1	1	0	1	1	1	**1**
1	1	1	0	0	0	**0**
1	1	1	0	0	1	**1**
1	1	1	0	1	0	**0**
1	1	1	0	1	1	**1**
1	1	1	1	0	0	**0**
1	1	1	1	0	1	**1**
1	1	1	1	1	0	**0**
1	1	1	1	1	1	**1**

11.

a) The value of X controls will we do A+B or A+C. If the input of X is 0 then we will add A+B, if it is equal to1 we will add A+C.

b) What we have is circuit that does only addition, if we want it to do subtraction we need to convert our number B into a 2's complement negative number. The way we do it is by taking the inputs for B, invert it, and put it back into C. What we are still missing is the 1 that we need to add to flipped B to make it a 2's complement, however we can use X as an input into the Cin bit of the circuit, so we actually added 1 if we are doing subtraction but we add 0 if we are doing addition. The figure of the circuit is:



12.

a) The propagation of mux is three gate delays since the longest path is inverting S bit, doing S' AND A, and in the end we have an OR gate.

b) The propagation delay for one full adder is 3 gate delays, so for 4 bit adder it is 12.

c) we can reduce it to 3 gate delays by grouping the terms ((A and B) and (C and D)) and E.

13.

a)

A	B	C	D	E	Z
0	0	0	0	0	**0**
0	0	0	0	1	**1**
0	0	0	1	0	**1**
0	0	0	1	1	**0**
0	0	1	0	0	**1**
0	0	1	0	1	**0**
0	0	1	1	0	**0**
0	0	1	1	1	**1**
0	1	0	0	0	**1**
0	1	0	0	1	**0**
0	1	0	1	0	**0**
0	1	0	1	1	**1**
0	1	1	0	0	**0**
0	1	1	0	1	**1**
0	1	1	1	0	**1**
0	1	1	1	1	**1**
1	0	0	0	0	**1**
1	0	0	0	1	**0**
1	0	0	1	0	**0**
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	**0**
1	1	0	0	0	**0**
1	1	0	0	1	**1**
1	1	0	1	0	**1**
1	1	0	1	1	**0**
1	1	1	0	0	**1**
1	1	1	0	1	**0**
1	1	1	1	0	**0**
1	1	1	1	1	**1**

b) All the combinations that give the output 1 have an odd number of 1s and all the combinations that give output 0 have an even number of 1s.