**F.3 Chapter 3 Solutions**

3.1

|  |  |  |
| --- | --- | --- |
|  | N-Type | P-Type |
| Gate=1 | closed | open |
| Gate=0 | open | closed |

3.3 There can be 16 different two input logic functions.

3.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C |  | OUT |
| 0 | 0 | 0 |  | 1 |
| 0 | 0 | 1 |  | 0 |
| 0 | 1 | 0 |  | 1 |
| 0 | 1 | 1 |  | 0 |
| 1 | 0 | 0 |  | 1 |
| 1 | 0 | 1 |  | 0 |
| 1 | 1 | 0 |  | 0 |
| 1 | 1 | 1 |  | 0 |

3.7 There is short circuit (path from Power to Ground) when either A = 1 and B = 0 or A = 0 and B = 1.

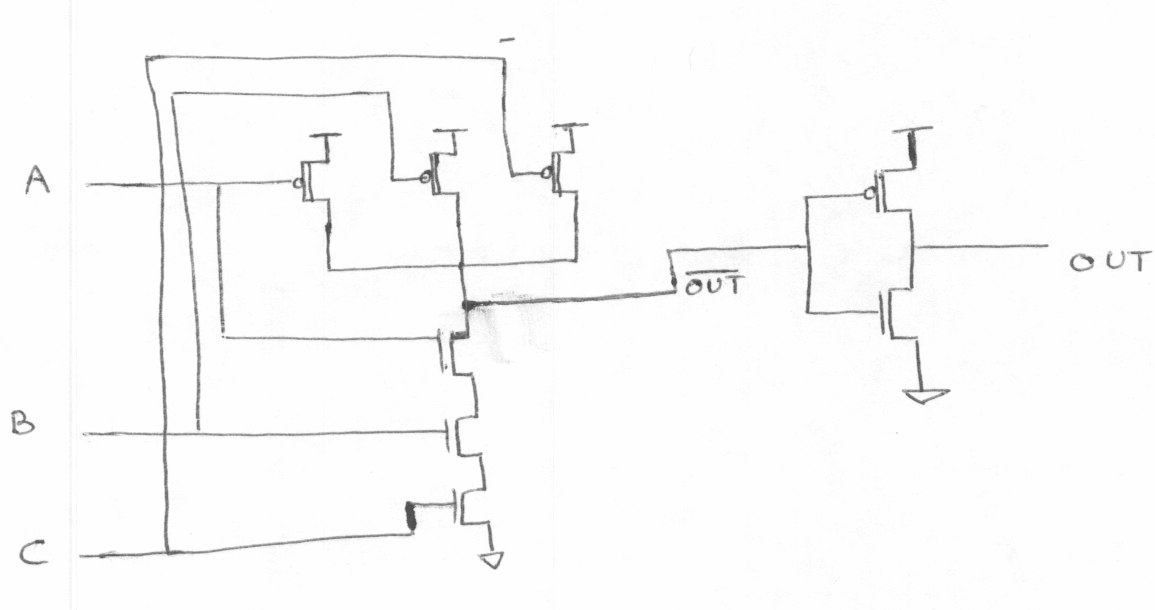
* 1. The circuit is floating when A = 0 and creates a short-circuit when A = 1, which is guaranteed to burn out the power supply.
  2. When A = 0, Out = 3.3 V  
      When A = 1, Out is floating, but power and ground are shorted, which will result in very high current and possible breakdown.
  3. \***NOTE**: This problem was mistakenly included in Chapter 3 when it should belong in Chapter 5. See ERRATA for more information\*  
     OUT = 1 signifies that the instruction is a branch instruction, and the branch will be taken. OUT = 0 means otherwise.

3.15

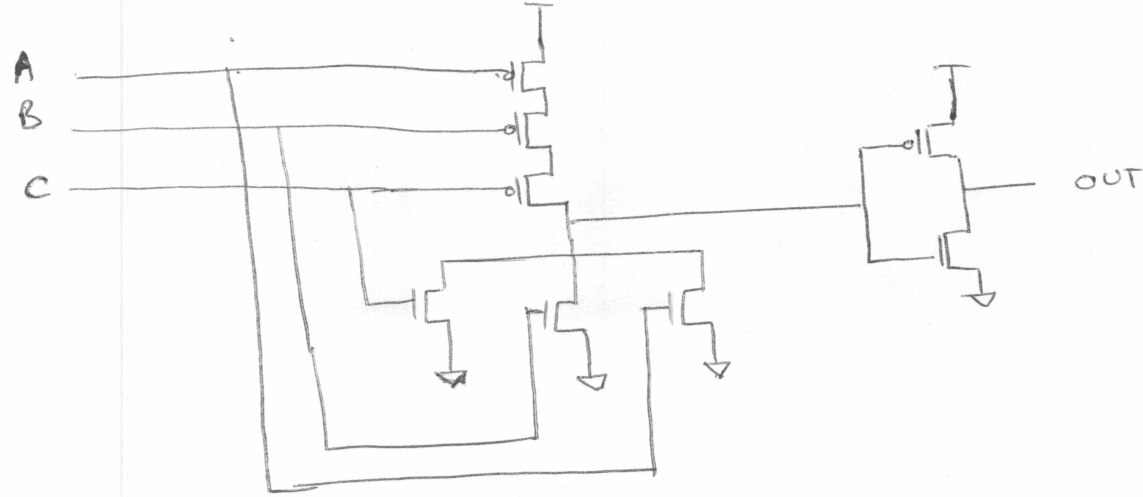
|  |  |  |  |
| --- | --- | --- | --- |
| A | B |  | NOT(NOT(A) OR NOT(B)) |
| 0 | 0 |  | 0 |
| 0 | 1 |  | 0 |
| 1 | 0 |  | 0 |
| 1 | 1 |  | 1 |

AND gate has the same truth table.

3.17 a. Three input And-Gate

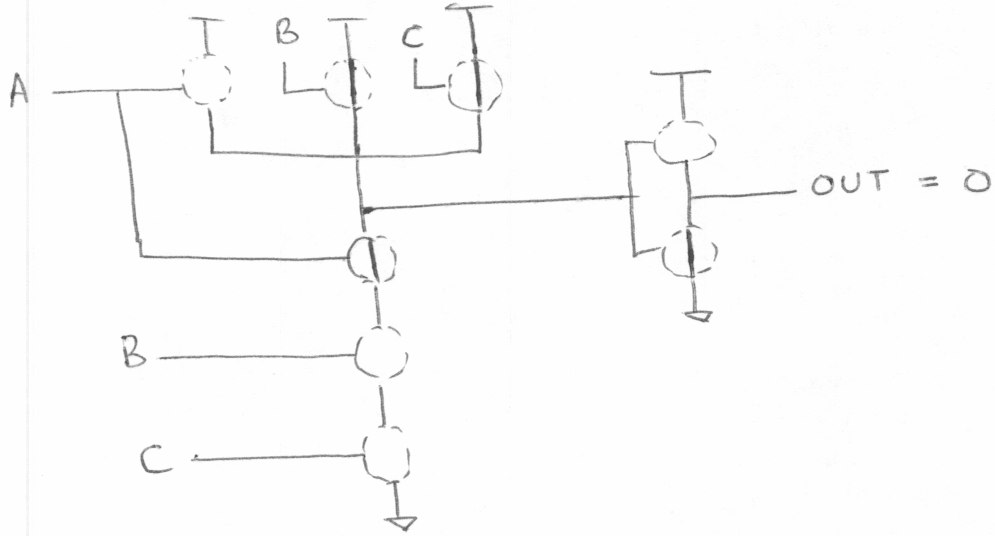


Three input OR-Gate

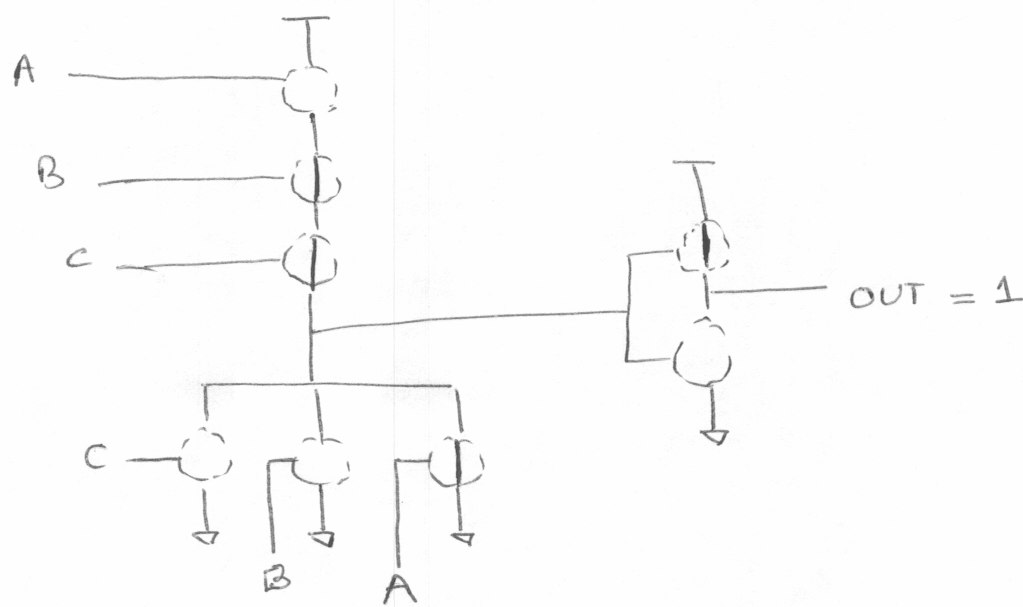


b. (1) A = 1, B = 0, C = 0.

AND Gate

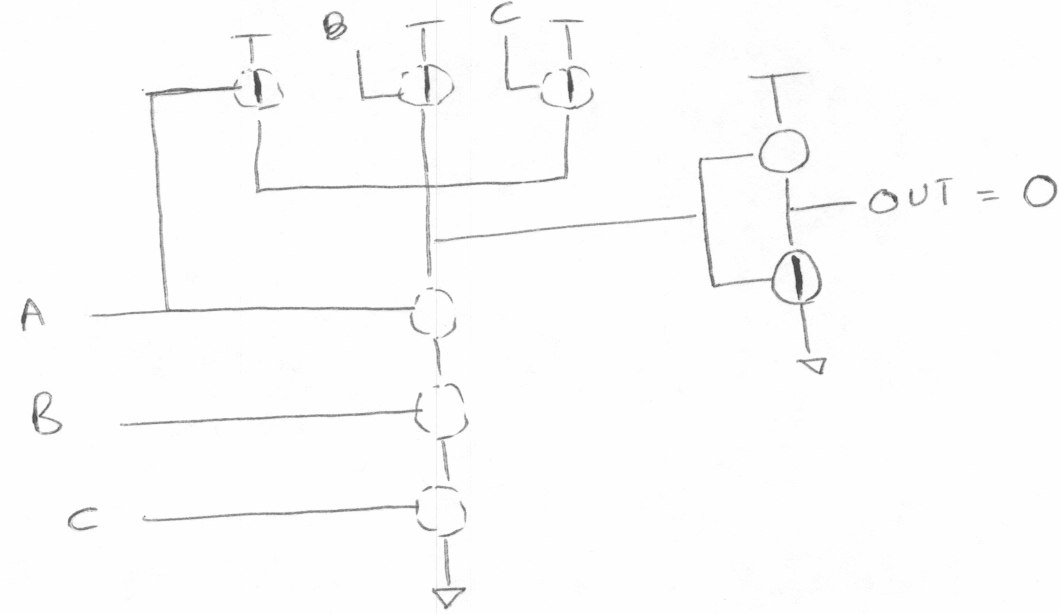


OR Gate

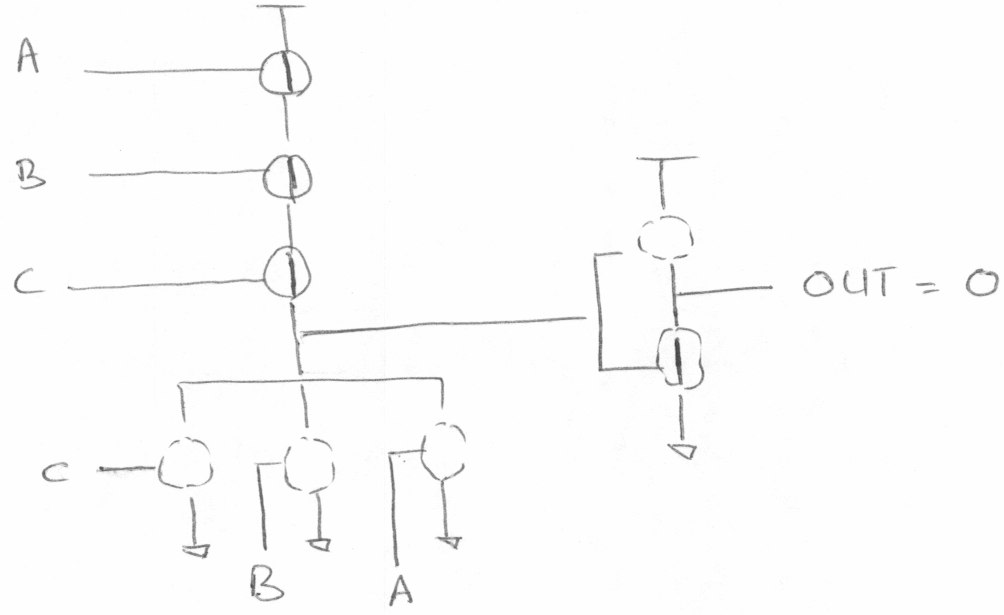


b. (2) A = 0, B = 0, C = 0

AND Gate

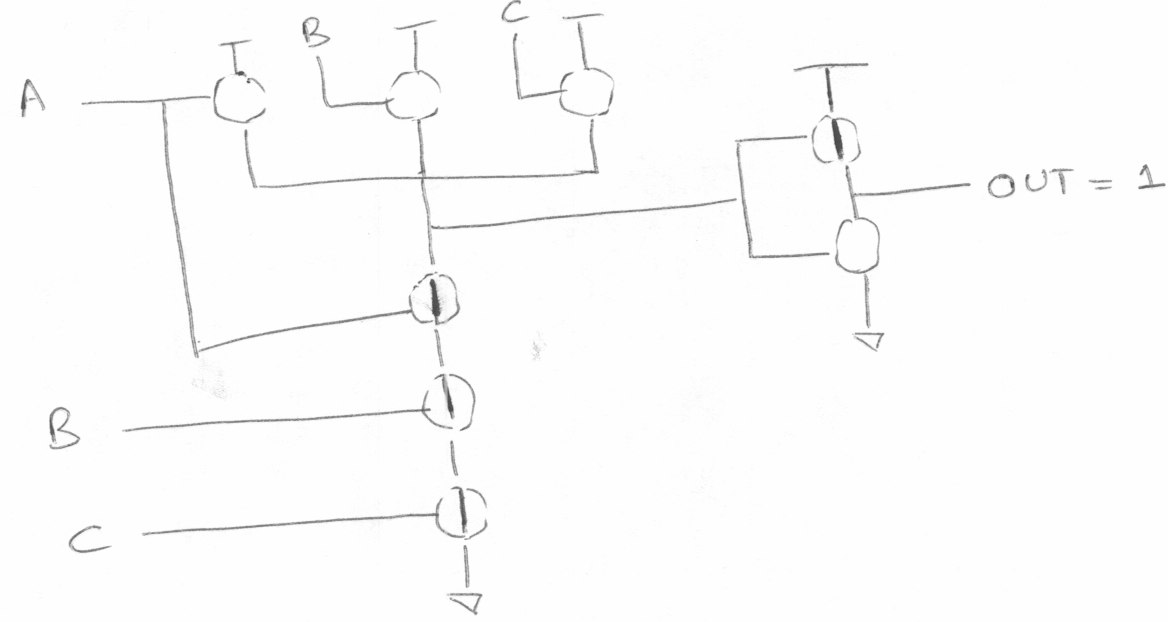


OR Gate

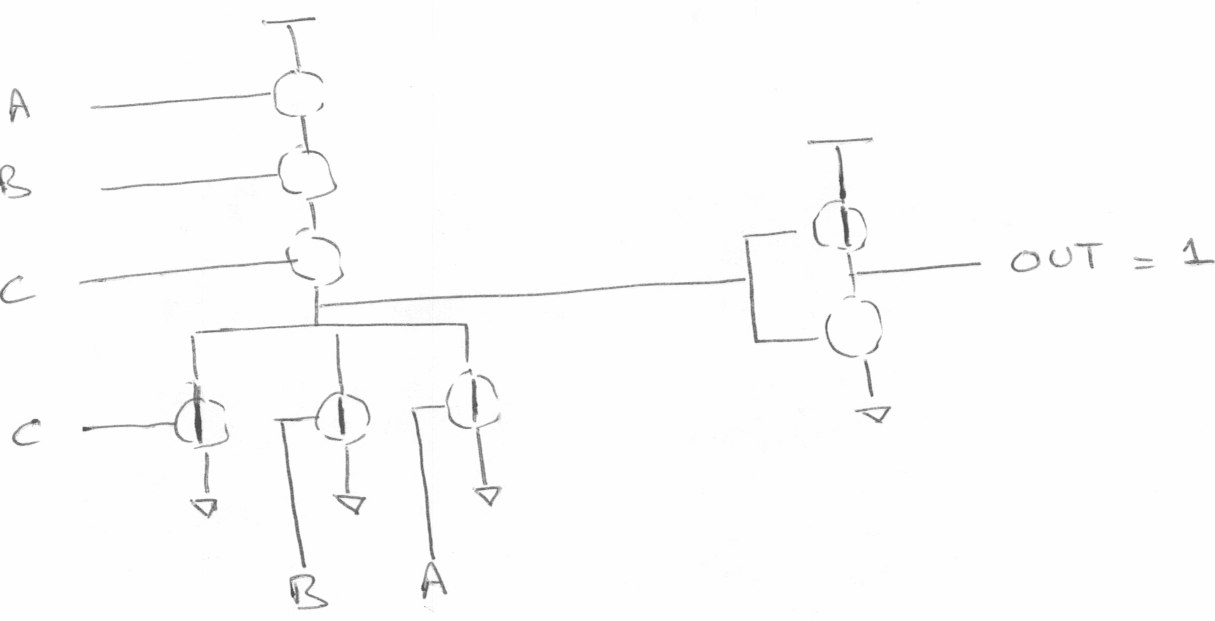


b. (3) A = 1, B = 1, C = 1

AND Gate



OR Gate



3.19 A five input decoder will have 32 output lines.

3.21

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cin | 1 | 1 | 1 | 0 |
| A | 0 | 1 | 1 | 1 |
| B | 1 | 0 | 1 | 1 |
| S | 0 | 0 | 1 | 0 |
| Cout | 1 | 1 | 1 | 1 |

A = 7, B = 11, A + B = 18.

In the above calculation, the result (S) is 2 !! This is because 18 is too large a number to be represented in 4 bits. Hence there is an overflow - Cout[3] = 1.

3.23 (a) The truth table will have 16 rows. Here is the truth table for Z = XOR (A, B, C, D). Any circuit with at least seven input combinations generating 1s at the output will work.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D |  | Z |
| 0 | 0 | 0 | 0 |  | 0 |
| 0 | 0 | 0 | 1 |  | 1 |
| 0 | 0 | 1 | 0 |  | 1 |
| 0 | 0 | 1 | 1 |  | 0 |
| 0 | 1 | 0 | 0 |  | 1 |
| 0 | 1 | 0 | 1 |  | 0 |
| 0 | 1 | 1 | 0 |  | 0 |
| 0 | 1 | 1 | 1 |  | 1 |
| 1 | 0 | 0 | 0 |  | 1 |
| 1 | 0 | 0 | 1 |  | 0 |
| 1 | 0 | 1 | 0 |  | 0 |
| 1 | 0 | 1 | 1 |  | 1 |
| 1 | 1 | 0 | 0 |  | 0 |
| 1 | 1 | 0 | 1 |  | 1 |
| 1 | 1 | 1 | 0 |  | 1 |
| 1 | 1 | 1 | 1 |  | 0 |

Z = XOR (A,B,C,D)

(b)

Z

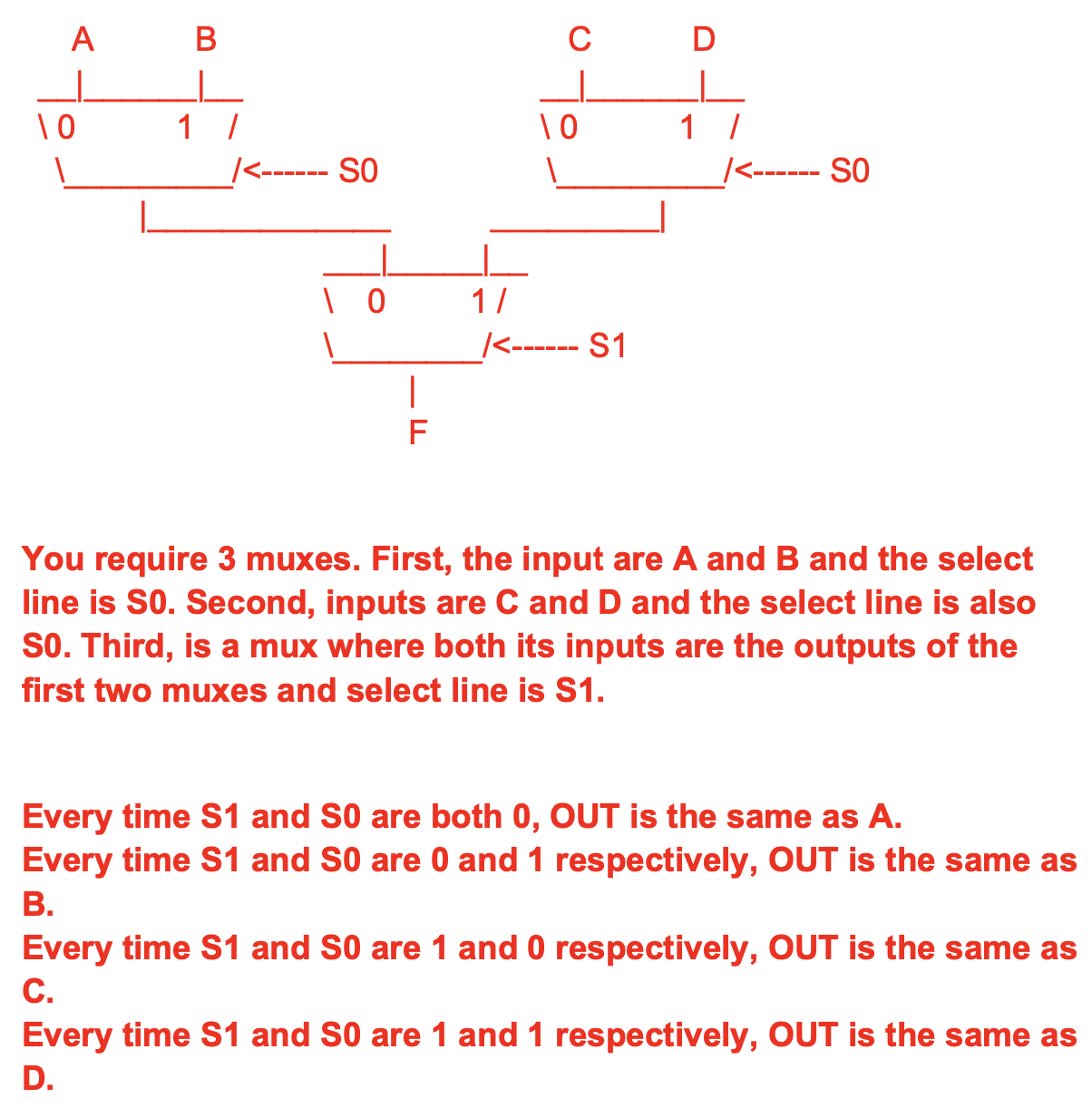
A B C D

3.25 Figure 3.36 is a simple combinational circuit. The output value depends ONLY on the input values as they currently exist. Figure 3.37 is an R-S Latch. This is an example of a logic circuit that can store information. That is, if A, B are both 1, the value of D depends on which of the two (A or B) was 0 most recently.

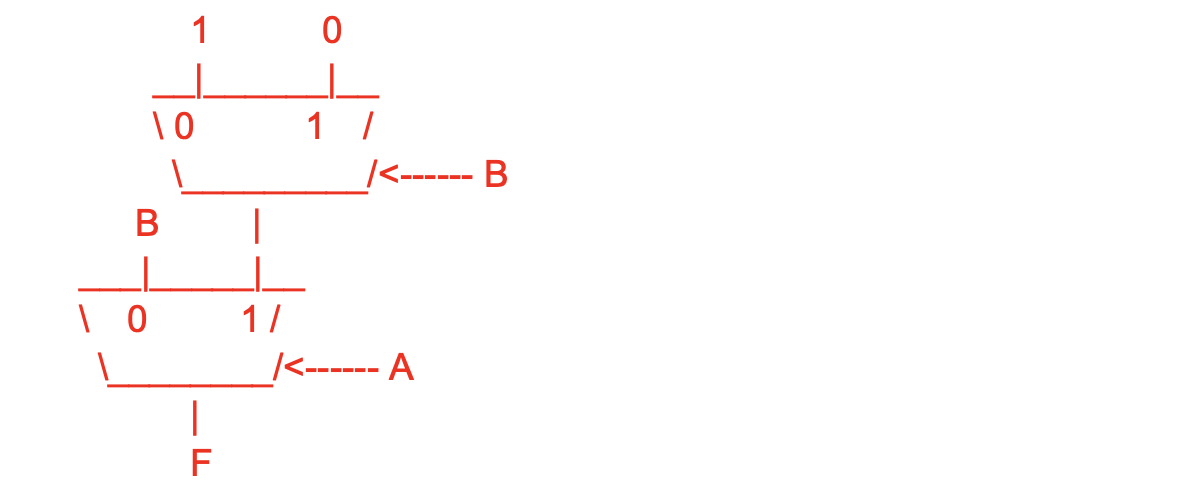
3.27 2 \* 214 = 215 = 32768 nibbles

3.28

A.



B.



3.29

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C |  | Z |
| 0 | 0 | 0 |  | 0 |
| 0 | 0 | 1 |  | 0 |
| 0 | 1 | 0 |  | 0 |
| 0 | 1 | 1 |  | 0 |
| 1 | 0 | 0 |  | 0 |
| 1 | 0 | 1 |  | 0 |
| 1 | 1 | 0 |  | 0 |
| 1 | 1 | 1 |  | 0 |

3.31 (a) 3 gate delays

3.31 (b) 3 gate delays

3.31 (c) 3\*4 = 12 gate delays

3.31 (d) 3\*32 = 96 gate delays

3.32 **For the OR gate generating the Si, connect the 4 outputs: 001, 010, 100, and 111 from the decoder to the OR gate. Connect the remaining 2 inputs of the OR gate to 0.**

**For the OR gate generating the Ci+1, connect the 4 outputs: 011, 101, 110, and 111 from the decoder to the OR gate. Connect the remaining 2 inputs of the OR gate to 0.**

3.33(a) When S=0, Z = A

3.33(b) When S=1, Z retains its previous value. 3.33(c) Yes; the circuit is a storage element.

3.35 No. The original value cannot be recovered once a new value is written into a register.

3.36 a)**ABGEL**

00010

01001

10100

11010

**b) G = AB' , L = A'B , E = A'B' + AB**

c) **Y = G[3] + E[3]G[2] + E[3]E[2]G[1] + E[3]E[2]E[1]G[0]**

3.37. 8 \* (2^3) = 64 bytes

3.39.(a) To read the 4th memory location, A[1,0] = 11, WE = 0

3.39.(b) A total of 6 address lines are required for a memory with 60 locations. The addressability of the memory will remain unchanged.

3.39.(c) A program counter of width 6 can address 2^6 = 64 locations. So without changing the width of the program counter, 64-60 = 4 more locations can be added to the memory.

3.41 Total bits of storage = 2^22 \* 3 = 12582912

3.43 There are a total of four possible states in this lock. Any other state can be expressed as one of states A, B, C or D. For example, the state performed one correct followed by one incorrect operation is nothing but state A as the incorrect operation reset the lock.

* 1. No. An arc is needed between the two states.
     1. Game in Progress:

Texas \* Oklahoma

Fouls:4 Fouls: 4

73 68

First Half 7:38

Shot Clock : 14

* + 1. Texas Win:

Texas \* Oklahoma

Fouls:10 Fouls: 10

85 70

Second Half 0:00

Shot Clock : 0

* + 1. Oklahoma Win:

Texas \* Oklahoma

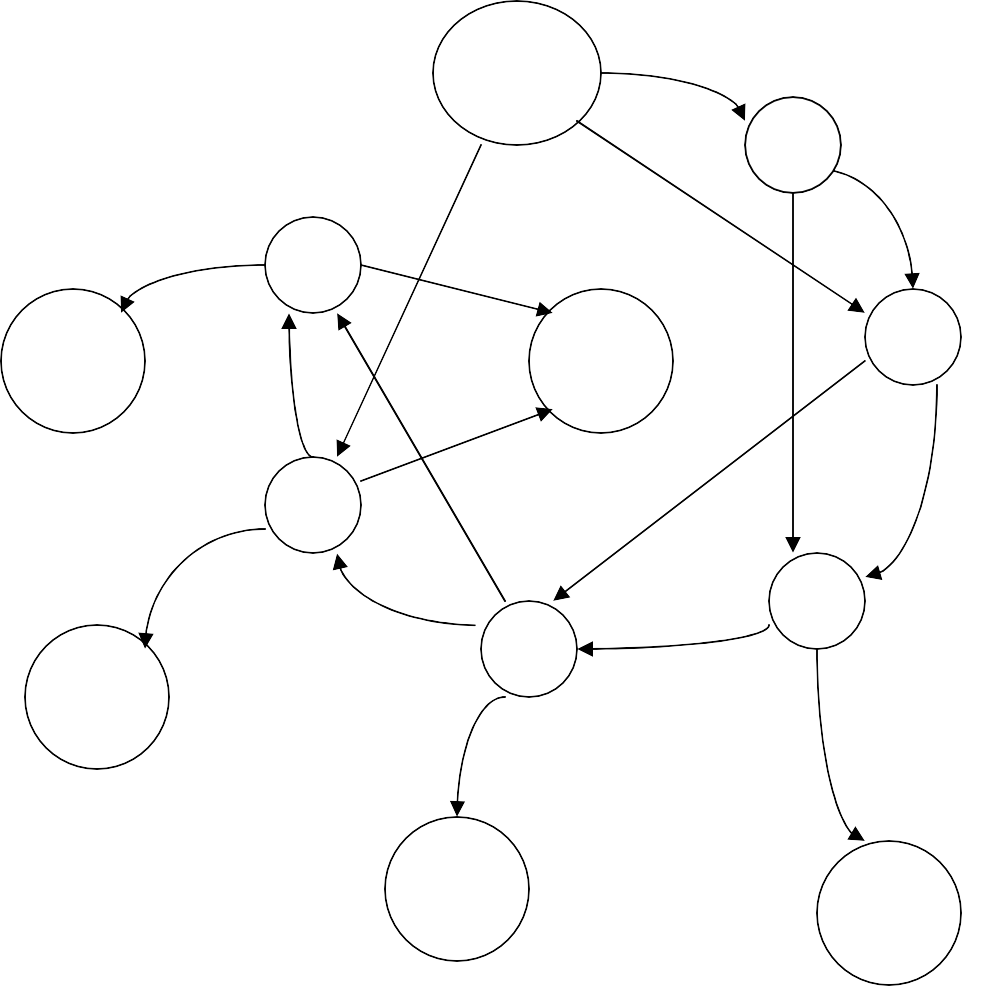
Fouls:10 Fouls: 10

81 90

First Half 7:38

Shot Clock : 0

3.47



No coins

N

Q

5

D/Q

Q

N

30

N

D

D

35+

ch

N

D

35

10

Q

D

Q

25

N

D

D

15

35+

ch

N

20

N

Q Q

35+

ch

35+

ch

3.49

a)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S1 | S0 | X |  | D1 | D0 | Z |
| 0 | 0 | 0 |  | 0 | 0 | 0 |
| 0 | 0 | 1 |  | 0 | 0 | 0 |
| 0 | 1 | 0 |  | 0 | 0 | 1 |
| 0 | 1 | 1 |  | 1 | 0 | 1 |
| 1 | 0 | 0 |  | 1 | 1 | 1 |
| 1 | 0 | 1 |  | 1 | 1 | 1 |
| 1 | 1 | 0 |  | 1 | 0 | 1 |
| 1 | 1 | 1 |  | 1 | 0 | 1 |

b)

0/1

0

00

01

0/1

1

11

10

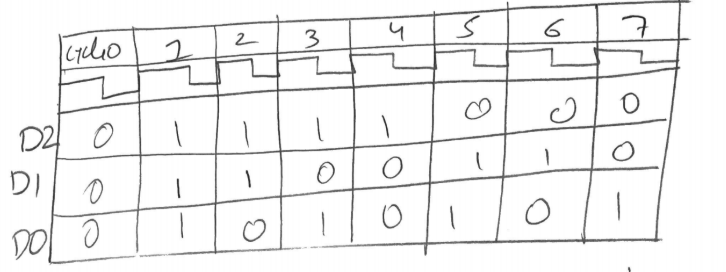
0/1

3.50

**NOT: A NAND A => NOT A  
AND: (A NAND B) NAND (A NAND B) = NOT (A NAND B) => A AND B**

3.51 Flip-Flop

3.52 A=1,B=0

3.53   
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
 The circuit is a decrementing 3-bit counter.

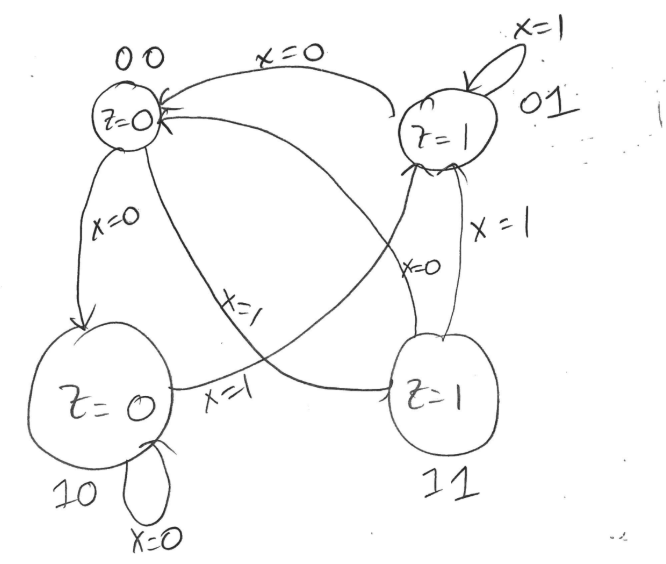
A close up of text on a whiteboard

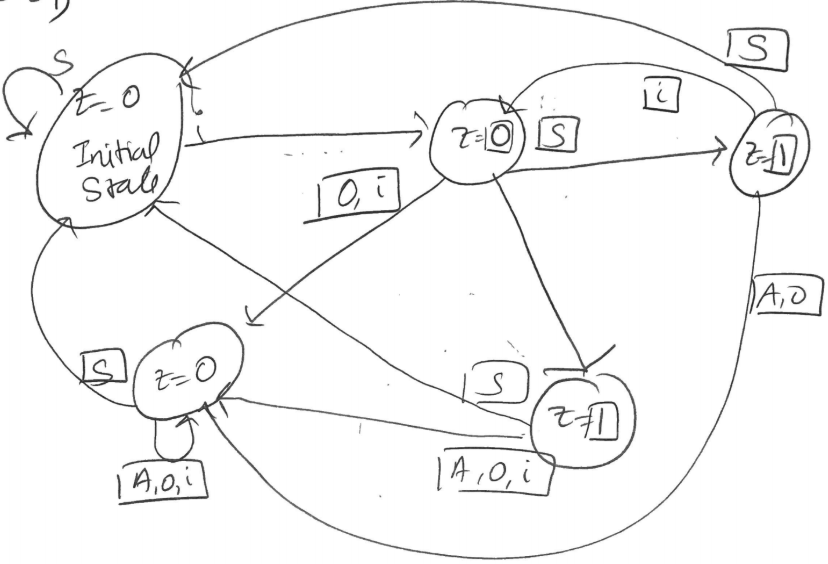
Description automatically generated3.55

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | b | c |  | Y | Z |
| 0 | 0 | 0 |  | 0 | 0 |
| 0 | 0 | 1 |  | 1 | 0 |
| 0 | 1 | 0 |  | 1 | 1 |
| 0 | 1 | 1 |  | 0 | 1 |
| 1 | 0 | 0 |  | 1 | 1 |
| 1 | 0 | 1 |  | 0 | 1 |
| 1 | 1 | 0 |  | 0 | 0 |
| 1 | 1 | 1 |  | 1 | 0 |

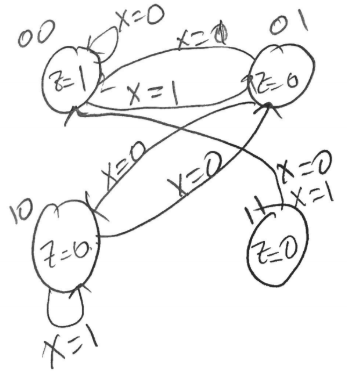
3.57





3.59

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S1 | S0 | X |  | Z | S1’ | S0’ |
| 0 | 0 | 0 |  | 1 | 0 | 0 |
| 0 | 0 | 1 |  | 1 | 0 | 1 |
| 0 | 1 | 0 |  | 0 | 1 | 0 |
| 0 | 1 | 1 |  | 0 | 0 | 0 |
| 1 | 0 | 0 |  | 0 | 0 | 1 |
| 1 | 0 | 1 |  | 0 | 1 | 0 |
| 1 | 1 | 0 |  | 0 | 0 | 0 |
| 1 | 1 | 1 |  | 0 | 0 | 0 |

3.61