FreeRange

Digital Design
Foundation Modeling
Solutions Manual

Chapter 1 Exercises (Answers)

1) List and briefly describe the basic definition of digital design.

ANS: Digital design is the act of creating a digital circuit to solve a given problem.

2) Briefly explain why there is no good off-the-shelf textbook for digital design courses.

ANS: Digital design textbooks have not kept up with current technology. They teach concepts that primarily support the easy generation of exam problems. The main problem, however, is that they teach little or no actual digital design concepts. But they do primarily teach stuff that you'll forget ten minutes after you're tested on it.

3) List a few websites where you can purchase inexpensive digital design texts.

ANS: ebay.com, addall.com, alibris.com

4) Briefly describe the main goals of Digital Design Foundation Modeling.

ANS: To provide an approach to teach and describe digital circuits in a uniform and simple manner. DDFM leverages the fact that individual digital circuits are relatively simple to understand at the operational level. DDFM also presents a "structured" approach to digital design in that any digital circuit can be constructed from a set of relatively simple digital modules, which we refer to as digital design foundation modules.

5) Briefly describe the three main types of design.

ANS: Brute Force Design (BFD), Iterative Modular Design (IMD), and Modular Design (MD).

6) Briefly describe the four ways you can control a digital circuit.

ANS: 1) no control, 2) external control, 3) internal control, and 4) circuit control.

Chapter 2 Exercises (Answers)

1) The analog world we live in has many people who seem to thrive on the use of digital photography. Practically everyone it has a digital camera, or has the equivalent on his or her cell phone or computer. A conversion from analog to digital occurs somewhere in the camera. Where exactly does this analog-to-digital (ADC) occur? Explain as best you can.

ANS: The conversion occurs at the light sensor. The photons cause charge to collect, the collected charge has a voltage potential, which is an analog value. That voltage is converted to a digital value, which is what the camera uses for its processing. These digital values (representing light intensities of red, blue, and green light) are converted back to analog so they can be displayed on the visual display device (LED, OLED, etc.).

2) Although the dimmer effectively provides what a continuous range of light frequencies between the ON and OFF limit, how can it possibly still be digital in nature? Explain as best you can.

ANS: The dimmer works by changing the duty cycle of a pulse width modulated signal. This means that if the light is half bright, the light is only on 50% of the time. You don't notice this because it happens to fast for the human visual system (HVS) to detect.

3) In reference to analog and digital cameras, describe the difference between analog zoom and digital zoom.

ANS: The analog zoom is a true zoom and is based on optics in place before the sensor. Digital zoom is a form of digital image processing and happens after the image has already been converted from analog to digital.

4) There are analog computers out there. Briefly describe what an analog computer entails. Feel free to look this up online.

ANS: Analog computers use a continuous range of physical attributes rather than the discrete 1's & 0's of digital computers. Stuff like slide rules and old clocks are forms of analog computers.

Chapter 3 Exercises (Answers)

1) Briefly explain the general purpose for a model.

ANS: The general purpose of a model is to describe something.

2) Is there one correct model for anything? Briefly explain your answer.

ANS: There is no one correct model for anything in general. The best model for anything is the model that transfers the most amount of useful information in the fastest manner.

3) Briefly describe the attributes of the "best" model for anything.

ANS: The best model for anything is the model that provides the most useful information for your purposes. If you're hoping for low-level details and get a model with high-level details, it will not be a good model for you.

4) List some of the pros and cons of not having stringent rules regarding basic black box modeling techniques.

ANS: Not having rules allows you the freedom to express models in a way that deem the most clear and the most informative. Having rules prevents you the freedom of expression and causes your soul to slowly die until you want nothing more than to become an academic administrator.

5) One of the themes of this chapter is the hierarchical design approach. Would it be possible to have too many levels for a given design? Explain your answer without being too verbose

ANS: You can have too many levels for a design. The idea is to present designs in an understandable manner; having an inappropriate number of levels reduces the understandability of the model. This means that too many levels or too few levels are equally as problematic in the context of understanding.

Chapter 3 Exercises (Answers)

1) Briefly explain the general purpose for a model.

ANS: The general purpose of a model is to describe something.

2) Is there one correct model for anything? Briefly explain your answer.

ANS: There is no one correct model for anything in general. The best model for anything is the model that transfers the most amount of useful information in the fastest manner.

3) Briefly describe the attributes of the "best" model for anything.

ANS: The best model for anything is the model that provides the most useful information for your purposes. If you're hoping for low-level details and get a model with high-level details, it will not be a good model for you.

4) List some of the pros and cons of not having stringent rules regarding basic black box modeling techniques.

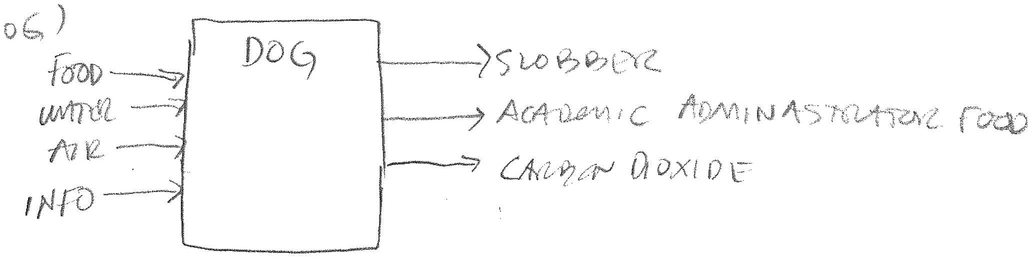
ANS: Not having rules allows you the freedom to express models in a way that deem the most clear and the most informative. Having rules prevents you the freedom of expression and causes your soul to slowly die until you want nothing more than to become an academic administrator.

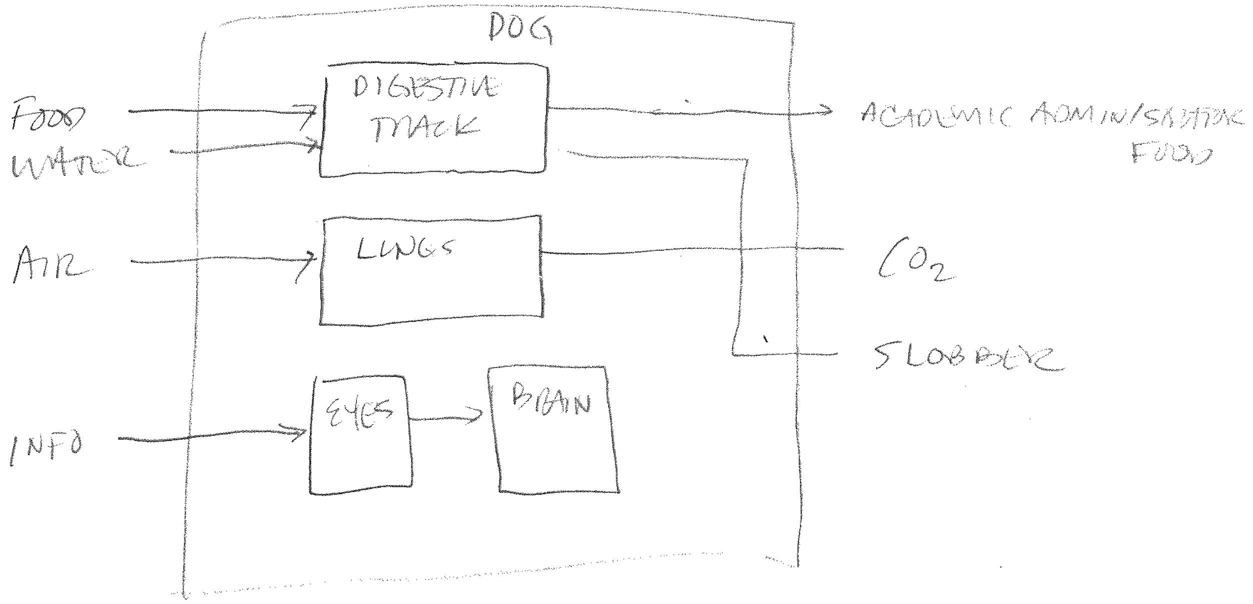
5) One of the themes of this chapter is the hierarchical design approach. Would it be possible to have too many levels for a given design? Explain your answer without being too verbose

ANS: You can have too many levels for a design. The idea is to present designs in an understandable manner; having an inappropriate number of levels reduces the understandability of the model. This means that too many levels or too few levels are equally as problematic in the context of understanding.

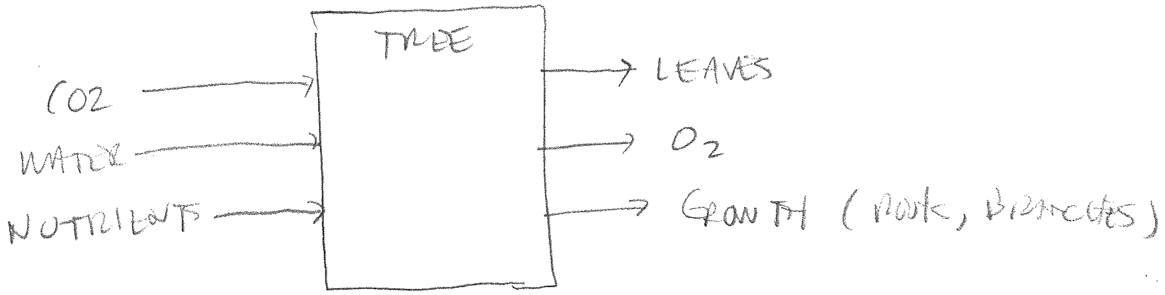
CHAPTER 3 EXERCISES

1) (DOG)

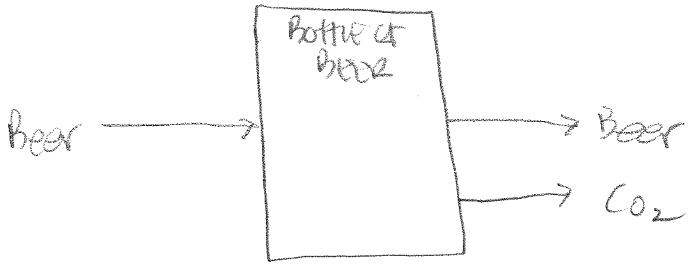




(TREE)

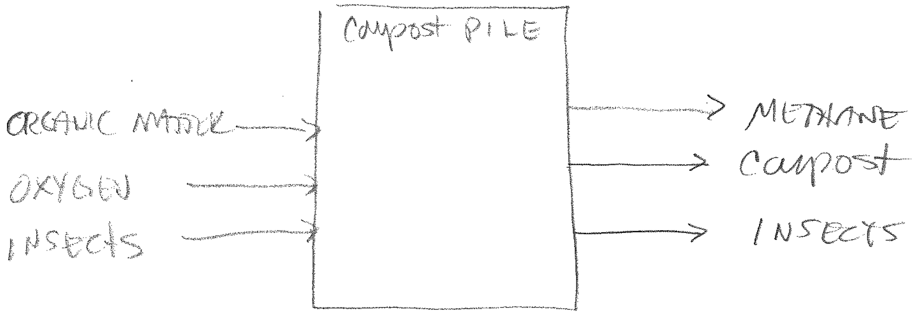


(Beer)

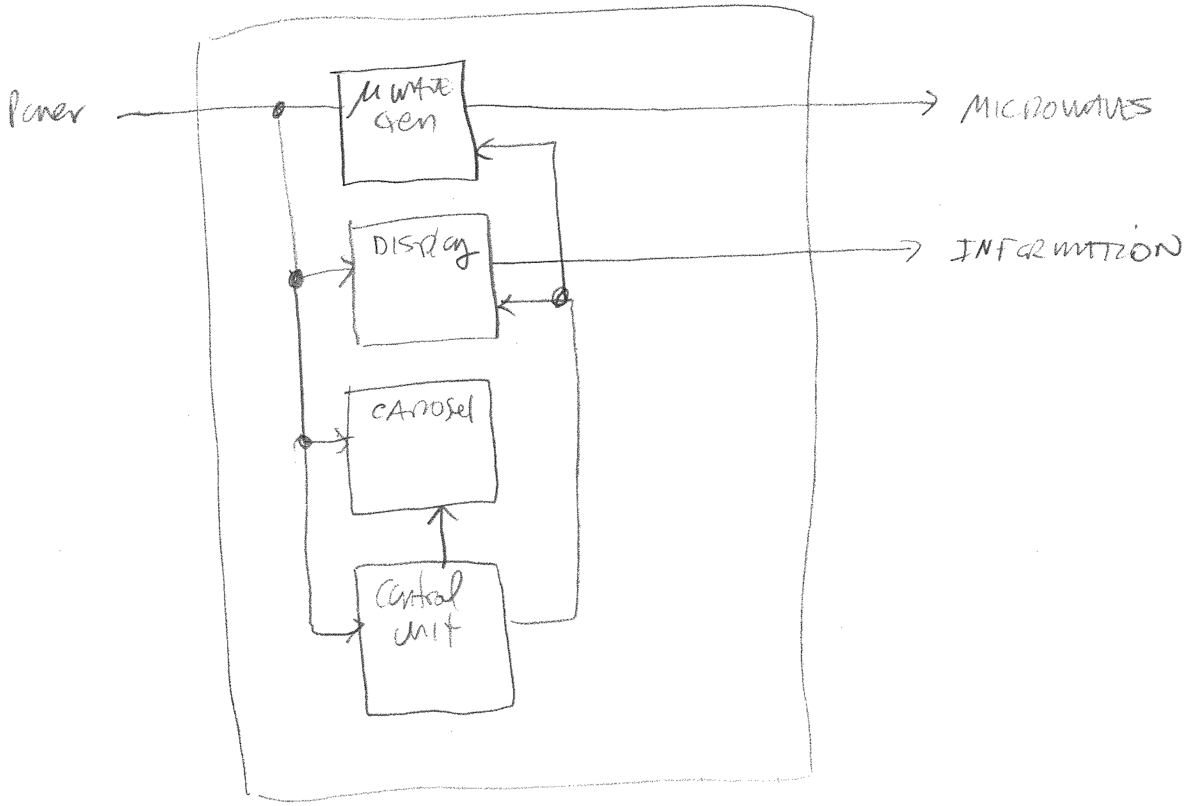


(BEST FRIEND) See DOG

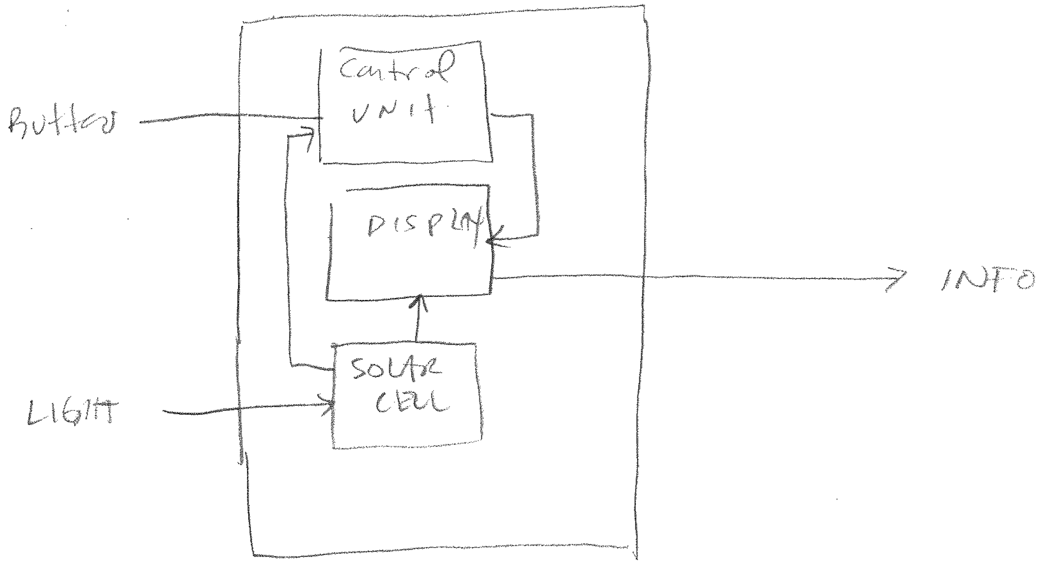
COMPOST PILE



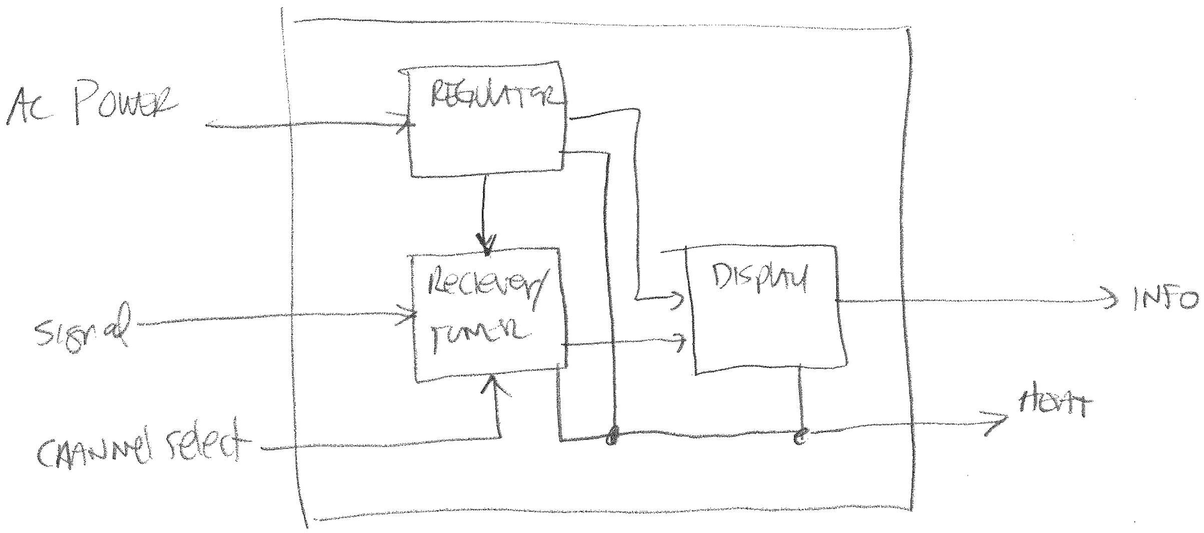
2) MICROWAVE OVEN



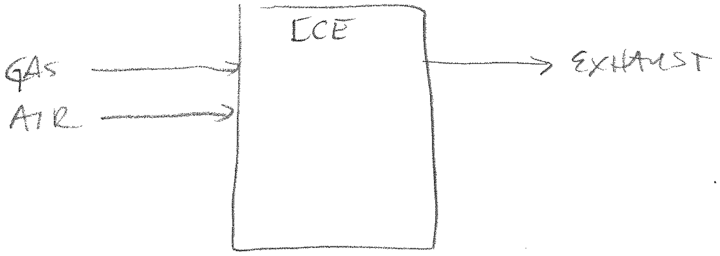
b) (calculater)

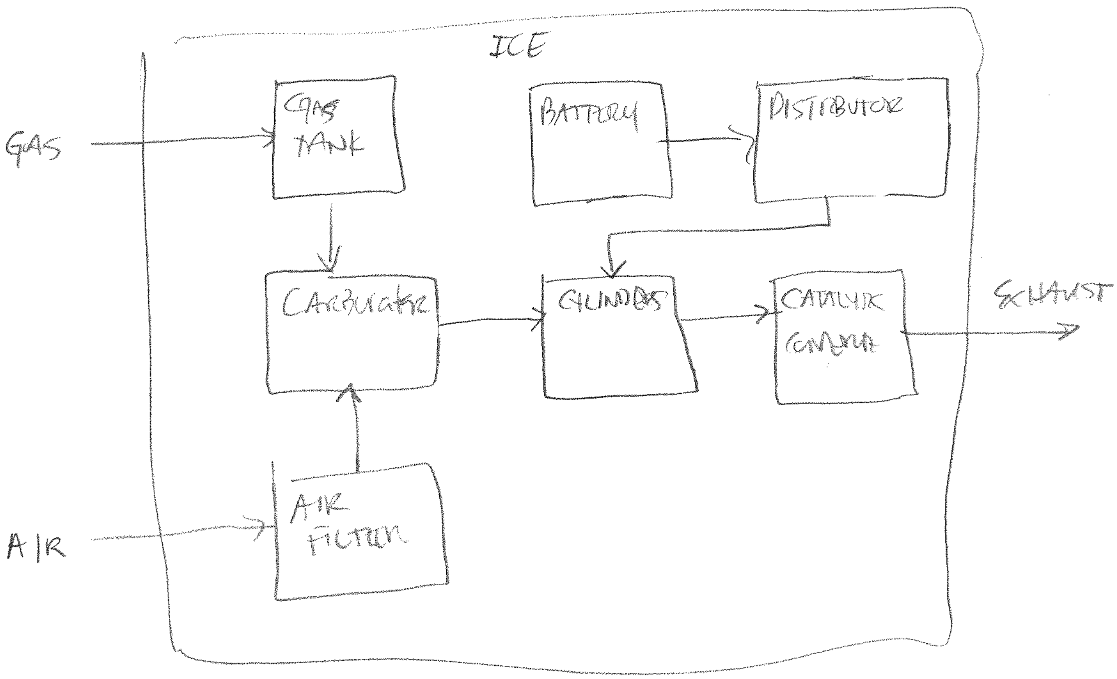


c) (TELEVISION)

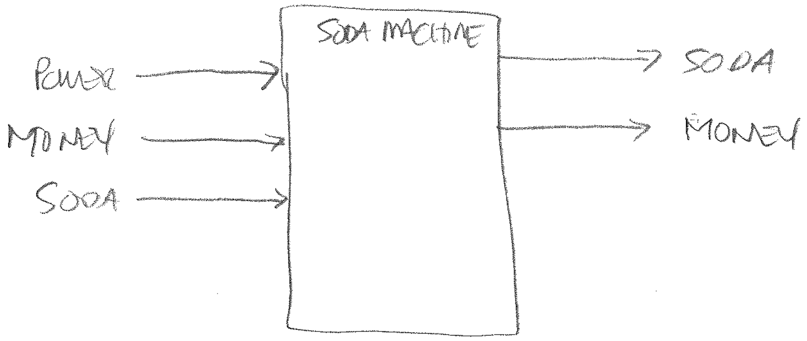


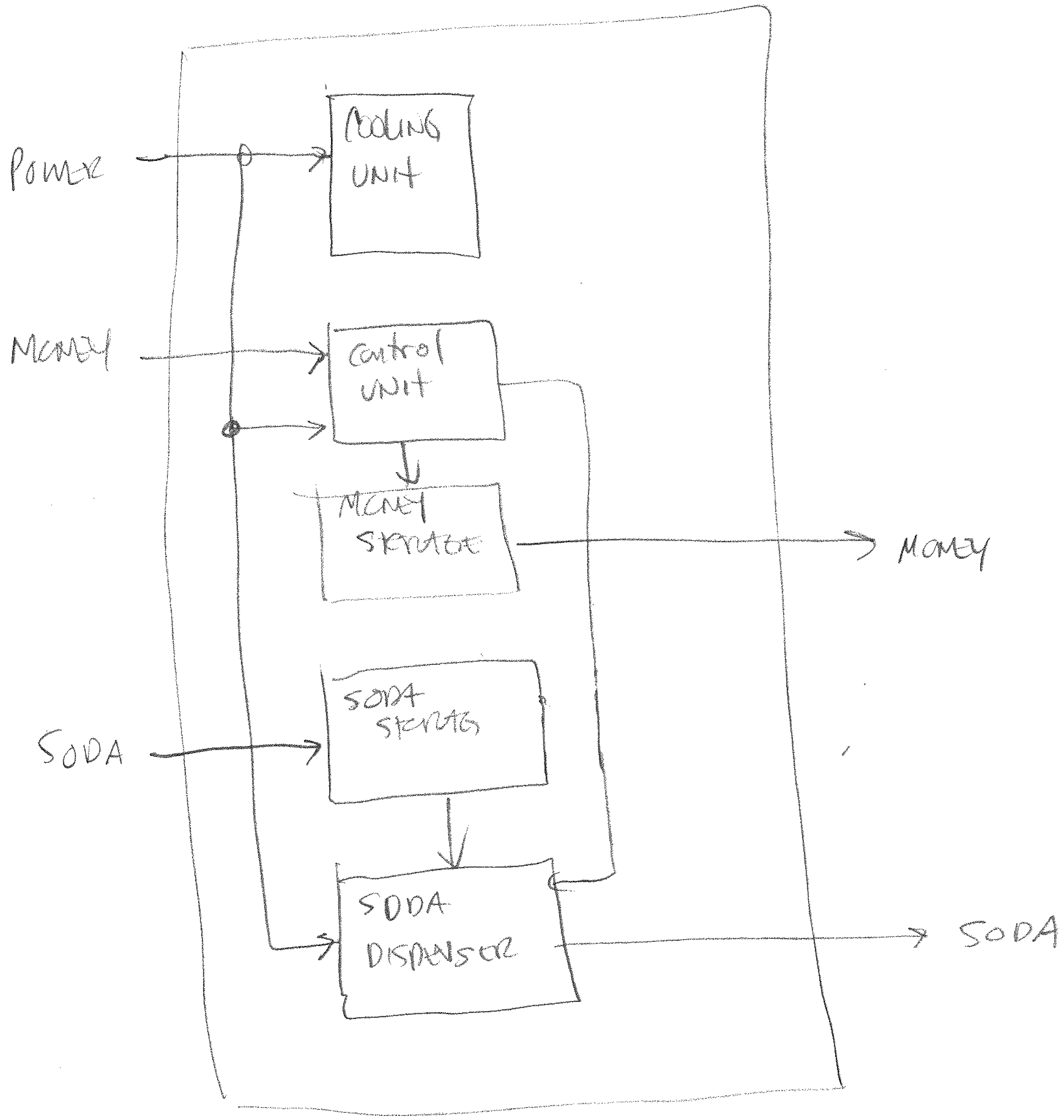
3) a) INTERNAL COMBUSTION ENGINE (ICE)





b) SODA DISPENSING MACHINE





SOLUTIONS CHAP 4

1) HEX IS USED AS A SHORTHAND NOTATION FOR BINARY

2) a) $235500000 = \underline{235.5M}$

b) $45 \times 10^{-4} = 4.5 \times 10^{-3} = \underline{4.5m}$

c) $241.3 \times 10^{8} = 24.13 \times 10^{9} = \underline{24.13G}$

d) $-33.8 \times 10^{-4} = -3.38 \times 10^{-3} = \underline{-3.38m}$

e) $0.00303 \times 10^{-4} = 303 \times 10^{-9} = \underline{303n}$

f) $0.146 \times 10^{8} = 14.6 \times 10^{6} = \underline{14.6M}$

g) $0.0000000253 \times 10^{4} = 253 \times 10^{-6} = \underline{253\mu}$

h) $8.355 \times 10^{7} = 83.55 \times 10^{6} = \underline{83.55M}$

3) a) $235500000 = 235 \times 10^{6} > 23.55 \times 10^{6}$

LARGER

b) $4.5m = 4.5 \times 10^{-3}$ $45 \times 10^{-4} = 4.5 \times 10^{-3}$

EQUAL

c) $241.3M = 241.3 \times 10^{6}$ $241.3 \times 10^{8} = 2.41 \times 10^{6}$

LARGER

d) -33.8×10^{-6} -33.81×10^{-6}

LARGER

e) $47.4 \times 10^{-6} = 47.4\mu$ $4.74 \times 10^{8} =$

LARGER

4)
- CARRY 153 Lb ROCK
- CARRY 153 OZ of WATER

} SORT OF THE SAME AS STORAGE UNARY

5) a) ||| OR ● ● ●

b) ~~||||~~ ~~||||~~ ~~||||~~ | OR ●●●●● ●●●●● ●●●●● ●

c) YEAH, RIGHT!

6) 4 ⇒ $2^4 = 16$

8 ⇒ $2^8 = 256$

12 ⇒ $2^{12} = 4096$

7) a) 7 ⇒ 0111_2 → 7_{16}

b) 9 ⇒ 1001_2 ⇒ 9_{16}

c) 14 ⇒ 1110_2 ⇒ E_{16}

d) 2 ⇒ 0010_2 ⇒ 2_{16}

e) 15 ⇒ 1111_2 ⇒ F_{16}

8) BECAUSE DIGITAL CIRCUITS ARE implemented WITH TRANSISTORS THAT ARE (Generally speaking) on OR OFF.

9) TOUGH PROBLEM to WORD

Ex: 3 bits ⇒ RANGE = $0 \rightarrow 2^3 - 1$ ⇒ $0 \rightarrow 7$

MIDDLE 2 NUMBER IN RANGE = 3, 4

Closed form formula = $\underline{2^{x-1} - 1}$, $\underline{2^{x-1}}$

Test it: 4 bits ⇒ 0 – 15 $2^{x-1} - 1 = 7$

$2^{x-1} = 8$

10)

A_5	B_3	C_2	D_1	#
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	3
0	1	0	1	4
0	1	1	0	5
0	1	1	1	6
1	0	0	0	5
1	0	0	1	6
1	0	1	0	7
1	0	1	1	8
1	1	0	0	8
1	1	0	1	9
1	1	1	0	10
1	1	1	1	11

$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$

11) NUM Bits = $\lceil \log_2 X \rceil$ where X is THE NUMBER

a) $\lceil \log_2 3 \rceil = 2$

b) 6

c) 8

d) 8

e) 12

f) 8

12) UNIQUE #'s $= 2^X$

a) $2^6 = 64$ $[0, 63]$

b) $2^{10} = 1024$ $[0, 1023]$

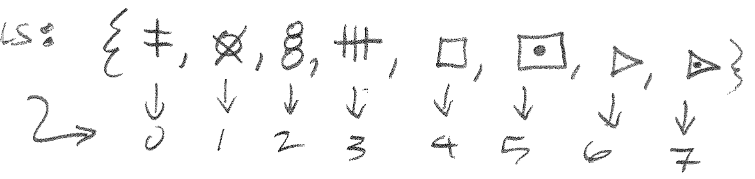
c) $2^8 = 256$ $[0, 255]$

e) $2^9 = 512$ $[0, 511]$

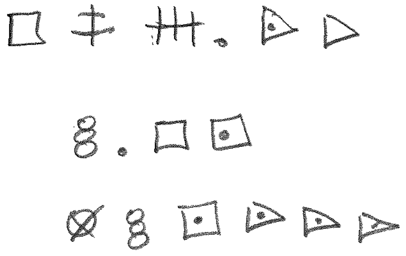
CHAPTER 4 DESIGN problems

1) ORDERED SET OF SYMBOLS:

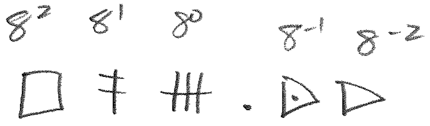
DECIMAL EQ



Example #'s



Example conversion to Decimal



Solution: $4\times 8^2 + 0\times 8^1 + 3\times 8^0 + 7\times 8^{-1} + 6\times 8^{-2}$

CHAPTER 5 EXERCISES SOLUTIONS

1) IT WORKS BASED ON POWERS OF 2; GROUPS OF 5 would not work.

2) a) 0111 1001 0001 BCD to Decimal

7 9 1 => 791

b) $0001\ 0000\ 0011\ 0110_{BCD}$ => 1036

1 0 3 6

c) 4377 to BCD

0100 0011 0111 0111 BCD

d) 70023 to BCD

0111 0000 0000 0010 0011 BCD

e) $4AC_{16}$ to Decimal

$4 \times 16^2 + 10 \times 16^1 + 12 \times 16^0 = ?$

f) $782B_{16}$ to Decimal

$7 \times 16^3 + 8 \times 16^2 + 2 \times 16^1 \div 11 \times 16^0 = ?$

g) 10110_2 to Decimal

$2^4 + 2^2 + 2^1 = ?$

H) 101111_2 to Decimal

$2^5 + 2^3 + 2^2 + 2^1 + 2^0 = ?$

3) a) $11011011_2 * 8 = 11011011000_2$

b) $10110110 \div 16 = 1011.0110_2$

c) $3AB_{16} \times 8$

$0011\ 1010\ 1011_2 * 8 = 0011\ 1010\ 1011\ 000$

$= 1\ D\ 5\ 8_{16}$

d) $4A7F_{16} \div 32$

$0100\ 1010\ 0111\ 1111_2 \Rightarrow 010|0\ 1010\ 0111\ 111|1_2$

$253.F8_{16}$

4) 145.801 , 8 is THE LARGEST value, so THE min RADIX IS 9

5) BA.12 , B is Associated with 11 (eleven) so The min is 12

6) 100110110.11_2

$15B.B_{16}$ ← Greater

0001|0101|1011.1011

$1\ 3\ 5\ .11_{16}$

7)

111 ✓	111	ANS
000 ✓	110	
110 ✓	100	
011	000	
001	001	
100 ✓	011	

8)

0011	1111	CAN'T BE DONE
0110	1110	
✓1100	1100	
0111		
✓1111		
✓1110 ✓		
0001		

9) 00 11 ⟺ 11 00 = A DISTANCE OF 4

10) CROSS out 1101

CROSS out 10111

11) LEFT COLUMN : ADD 0111

RIGHT COLUMN : ADD 1110

12) RIGHT-MOST AND SECOND FROM LEFT
ARE NOT UDCs

13) a) 3-bit

001
010
100

b) 6-bit

000001
000010
000100
001000
010000
100000

c) 8-bit

0000 0001
0000 0010
0000 0100
0000 1000
0001 0000
0010 0000
0100 0000
1000 0000

14) 0X001
0X002
0X004
0X008
0X010
0X020
0X040
0X080
0X100
0X200
0X400
0X800

14) 16-Bit one Hot code

CODE (BINARY	Hex Equivalent
0000 0000 0000 0001	0001
0000 0000 0000 0010	0002
0000 0000 0000 0100	0004
TOO BORING TO WRITE	0008
	0010
	0020
	0040
	0080
	0100
	0200
	0400
	0800
	1000
	2000
	4000
	8000

15) $3FD1_{16}$ $256 = 16^2$

ANS: $3F.D1_{16}$ Move Decimal place LEFT 2 spots

16) 101101000100111_2 $32 = 2^5$

ANS: 1011 01000 1.00111

17) $A473.1_{16}$

$A47310_{16}$

18) $B321.A2_{16} \times 4096$ $4096 = 16^3$

ANS $B321A20_{16}$

19) $110\ 110.10_2 \times 64$ $\qquad 64 = 2^6$

ANS: $110\ 110.100000. = 110110100000_2$

20) $1\ 10.1001_2 \quad \times 256$ $\qquad 256 = 2^8$

ANS $110\ 1001\ 0000_2$

CHAPTER 5 DESIGN PROBLEM SOLUTIONS

1)

00000
00010
00110
00100
0ϕ100
01000

2)

NUM	BCD	UDC
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	11ϕ0
6	0110	1111
7	0111	1011
8	1000	1001
9	1001	1000

3)

NUM	BIN	UDC
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	
10	1010	
11	1011	
12	1100	
13	1101	
14	1110	
15	1111	

4) 00000011 would be a GOOD CODE FOR A NOISY ENVIRONMENT
00001100
00110000
11000000

5) 00000011 * MANY POSSIBLE SOLUTIONS
00001100
00011000
00110000
01100000
11000000

CHAPTER 6 EXERCISES SOLNS

1) THE TRUTH TABLE FORMS THE BASIS OF ITERATIVE DESIGN

2) WE START WITH BFD BECAUSE we need to learn to CRAWL BEFORE WE LEARN to RUN.

3) IT'S "Brute Force" BECAUSE we must GRIND out A SOLUTION on A Low Level. THE NOTION OF BRUTE FORCE REFERS to how inefficient this STYLE OF DESIGN IS.

4) I CAN'T THINK OF ANY THING better. IF THERE IS A better APPROACH, I WOULD USE it. LET ME KNOW IF YOU THINK OF SOMETHING better.

5) BECAUSE WITH 4-variables, I'M WORKING WITH A 16-Entry truth TABLE. IT GETS Really UGLY AFTER THAT. For Example, A 5-VARIABLE TRUTH TABLE HAS 32 ROWS. 6-vars = 64 ROWS.

CHAP 6 EXERCISE SOLUTIONS 2/5

6) (a)

$$F = \overline{B2}\,\overline{B1}\,B0 + \overline{B2}\,B1\,\overline{B0} + B2\,\overline{B1}\,\overline{B0} \quad \boxed{\text{Sop}}$$

$$\overline{F} = \overline{B2}\,\overline{B1}\,\overline{B0} + \overline{B2}\,B1\,B0 + B2\,\overline{B1}\,B0 + B2\,B1\,\overline{B0} + B2\,B1\,B0$$

$$F = \overline{(\overline{B2}\,\overline{B1}\,\overline{B0}) + (\overline{B2}\,B1\,B0) + (B2\,\overline{B1}\,B0) + (B2\,B1\,\overline{B0}) + (B2\,B1\,B0)}$$

$$\text{Pos} \rightarrow F = (B2+B1+B0)\cdot(B2+\overline{B1}+\overline{B0})\cdot(\overline{B2}+B1+\overline{B0})\cdot(\overline{B2}+\overline{B1}+B0)\cdot(\overline{B2}+\overline{B1}+\overline{B0})$$

6 (b)

$$F = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,B\,\overline{C} + \overline{A}\,B\,C + A\,\overline{B}\,C \quad \boxed{\text{Sop}}$$

$$\overline{F} = \overline{A}\,\overline{B}\,C + A\,\overline{B}\,\overline{C} + A\,B\,\overline{C} + A\,B\,C$$

$$F = (A+B+\overline{C})\cdot(\overline{A}+\overline{B}+C)\cdot(A+\overline{B}+\overline{C})\cdot(\overline{A}+\overline{B}+\overline{C}) \quad \boxed{\text{POS}}$$

6 (c)

$$F = \overline{X}YZ + X\overline{Y}\,\overline{Z} + X\overline{Y}Z \quad \boxed{\text{Sop}} \nearrow$$

$$\overline{F} = \overline{X}\,\overline{Y}\,\overline{Z} + \overline{X}\,\overline{Y}Z + \overline{X}Y\overline{Z} + XY\overline{Z} + XYZ$$

$$F = (X+Y+Z)\cdot(X+Y+\overline{Z})\cdot(X+\overline{Y}+Z)(\overline{X}+\overline{Y}+Z)(\overline{X}+\overline{Y}+\overline{Z}) \quad \boxed{\text{POS}} \nearrow$$

6. (d)

$$F1 = \overline{T}\,\overline{U}\,\overline{V} + \overline{T}\,\overline{U}V + T\overline{U}V \quad \boxed{\text{Sop}}$$

$$\overline{F1} = \overline{T}U\overline{V} + \overline{T}UV + T\overline{U}\,\overline{V} + TU\overline{V} + TUV$$

$$F1 = (T+\overline{U}+V)\,(T+\overline{U}+\overline{V})\,(\overline{T}+U+V)(\overline{T}+\overline{U}+V)(\overline{T}+\overline{U}+\overline{V})$$

$$F2 = \overline{T}U\overline{V} + \overline{T}UV + TU\overline{V} + TUV \quad \boxed{\text{Sop}}$$

$$\overline{F2} = \overline{T}\,\overline{U}\,\overline{V} + \overline{T}\,\overline{U}V + T\overline{U}\,\overline{V} + T\overline{U}V$$

$$F2 = (T+U+V)\cdot(T+U+\overline{V})(\overline{T}+U+V)(\overline{T}+U+\overline{V}) \quad \boxed{\text{POS}}$$

7) a) $F = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + AB\bar{C}$

A	B	C	F
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

$\bar{F} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

$\boxed{F = (A+B+\bar{C})(A+\bar{B}+C)(\bar{A}+B+C)(\bar{A}+\bar{B}+\bar{C})}$

b) $F(A,B,C) = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + ABC$

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$\bar{F} = \bar{A}\bar{B}\bar{C} + \bar{A}BC + AB\bar{C}$

$\boxed{F = (A+B+C)(A+\bar{B}+\bar{C})(\bar{A}+\bar{B}+C)}$

c) $F(X,Y,Z) = \bar{X}\bar{Y}Z + X\bar{Y}\bar{Z} + XY\bar{Z} + XYZ$

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

$\bar{F} = \bar{X}\bar{Y}\bar{Z} + \bar{X}Y\bar{Z} + \bar{X}YZ + X\bar{Y}Z$

$\boxed{F = (X+Y+Z)(X+\bar{Y}+Z)(X+\bar{Y}+\bar{Z})(\bar{X}+Y+\bar{Z})}$

8a) $F(RST) = (\bar{R}+\bar{S}+\bar{T})(\bar{R}+S+\bar{T})(\bar{R}+S+T)(R+\bar{S}+T)(R+S+T)$

R	S	T	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

$\bar{F} = RST + R\bar{S}T + R\bar{S}\bar{T} + \bar{R}S\bar{T} + \bar{R}\bar{S}\bar{T}$

$F = \bar{R}\bar{S}T + \bar{R}ST + RS\bar{T}$

8 b) $F(A,B,C) = (A+B+C)(A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+\bar{B}+C)$

$\bar{F} = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + AB\bar{C}$

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$F = \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + ABC$

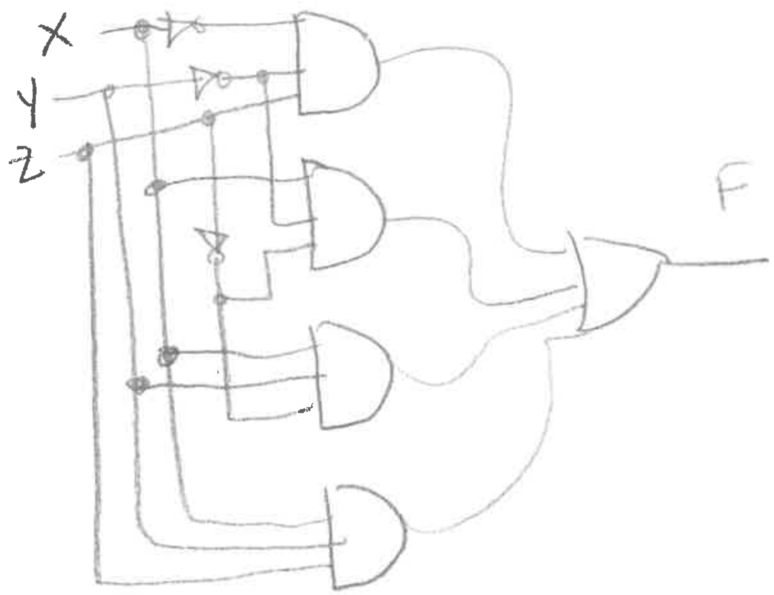
8 c) $F(X,Y,Z) = (\bar{X}+\bar{Y}+Z)(\bar{X}+Y+\bar{Z})(X+\bar{Y}+Z)(\bar{X}+\bar{Y}+\bar{Z})$

$\bar{F} = \underset{110}{XY\bar{Z}} + \underset{101}{X\bar{Y}Z} + \underset{010}{\bar{Y}Y\bar{Z}} + \underset{111}{XYZ}$

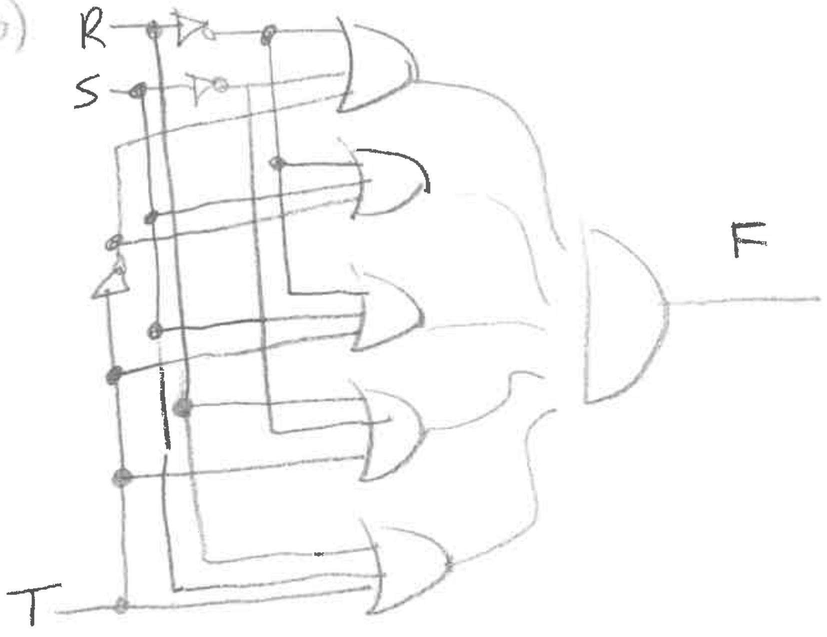
X	Y	Z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

$F = \bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}\bar{Z}$

9 a)



9 b)



CHAPTER 6 EXERCISE SOLUTIONS

10) $F(A,B,C) = \bar{A}B + AC$

11) $\bar{A}\bar{C}\bar{D} + B\bar{D} + A\bar{B}C = F(A,B,C,D)$

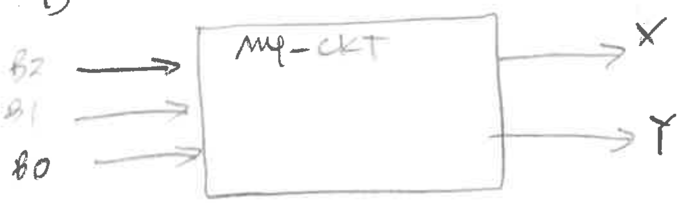
12) $\bar{A}B + AC = F(A,B,C)$

13) $F = AB + \bar{A}\bar{C}$

14) THE TRUTH TABLE would HAVE 64 ROWS. THIS IS BECAUSE THERE ARE 6 inputs, WHICH MEANS THERE ARE 2^6 POSSIBLE UNIQUE COMBINATIONS OF THE 6 inputs.

CHAPTER 6 DESIGN PROBLEMS SOLNS

1)



B2	B1	B0	X	Y
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	1
1	1	1	0	1

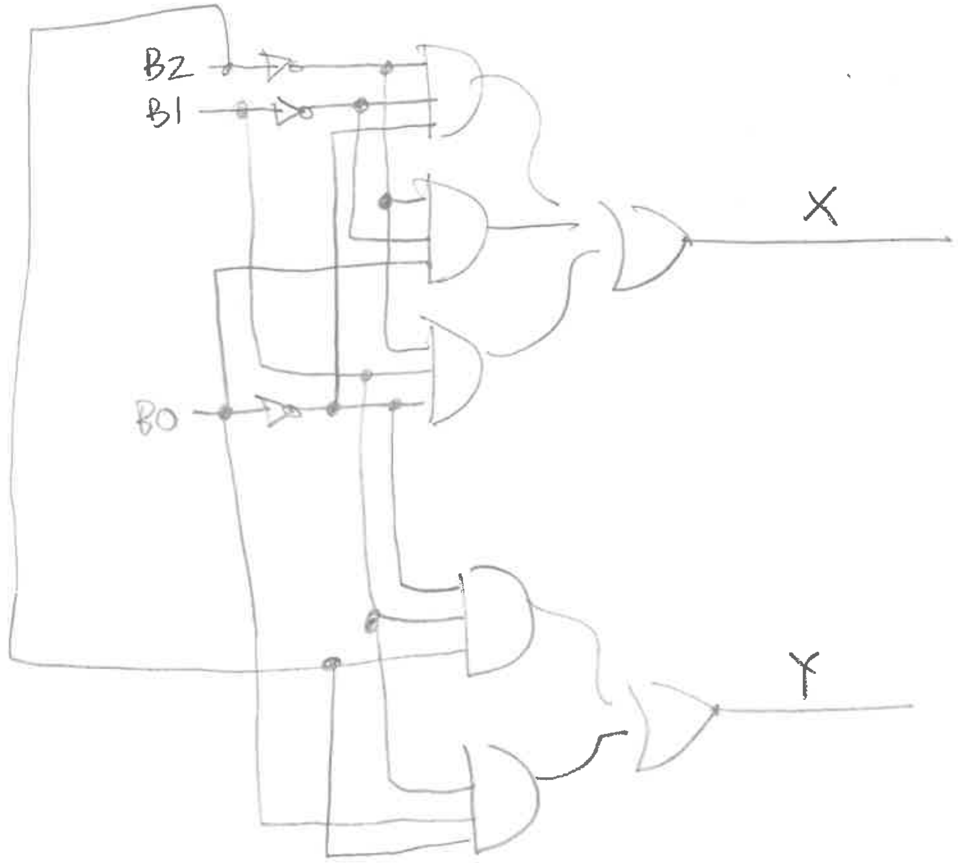
B2 = MSB B0 = LSB

X => INPUTS < 3

Y => input > 5

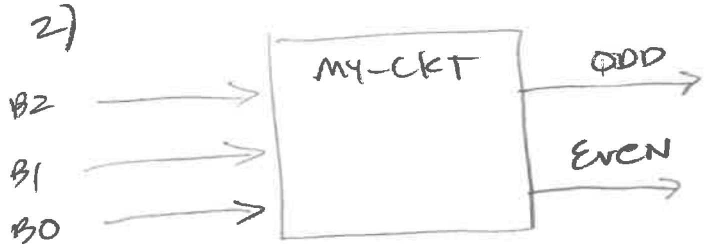
$$X = \overline{B2}\cdot\overline{B1}\cdot\overline{B0} + \overline{B2}\cdot\overline{B1}\cdot B0 + \overline{B2}\cdot B1\cdot\overline{B0}$$

$$Y = B2\cdot B1\cdot\overline{B0} + B2\cdot B1\cdot B0$$



UGLY!

2)



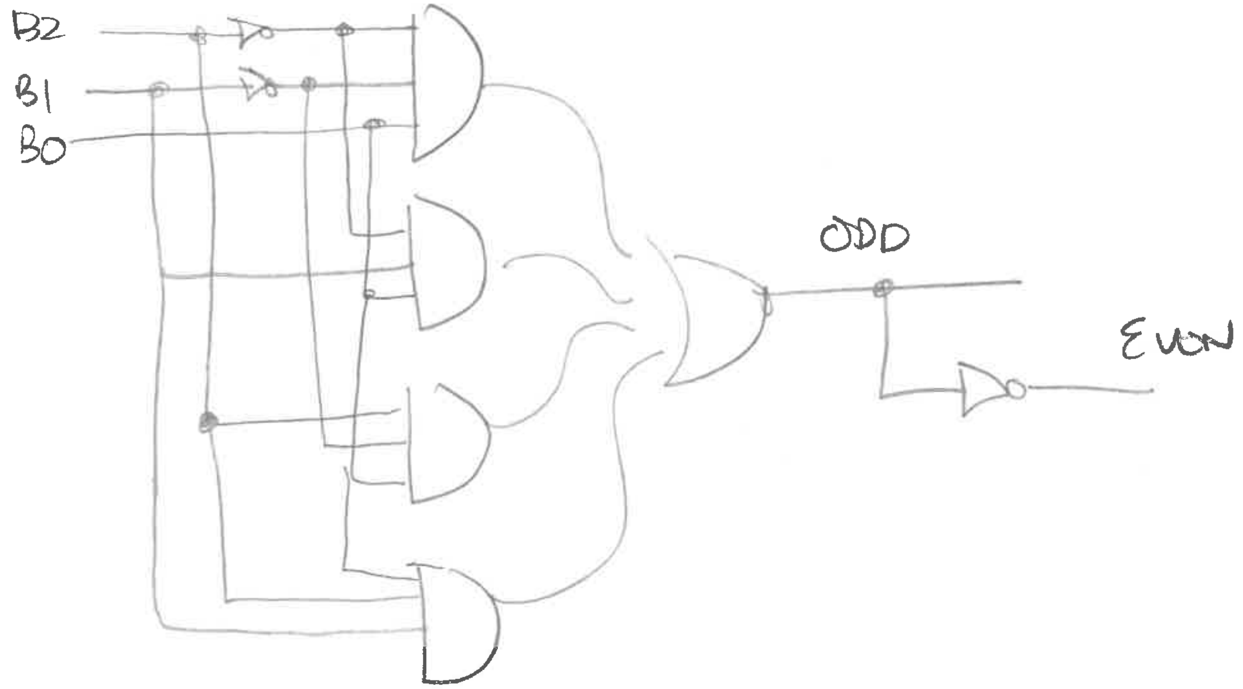
B2 = MSB
B0 = LSB

B2	B1	B0	ODD	Even
0	0	0	0	1
0	0	1	1	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	0	1
1	1	1	1	0

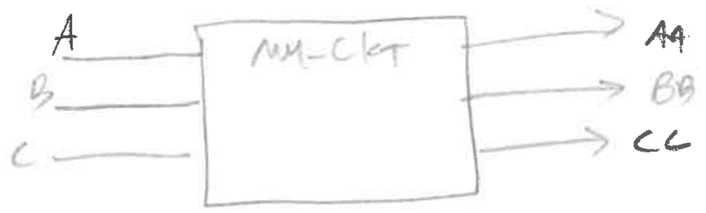
ODD

$$ODD = \overline{B2}\,\overline{B1}\,B0 + \overline{B2}\,B1\,B0 + B2\,\overline{B1}\,B0 + B2\,B1\,B0$$

$$EVEN = \overline{ODD}$$



3)



A	B	C	AA	BB	CC
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	0	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	0	1

A = MSB

B = LSB

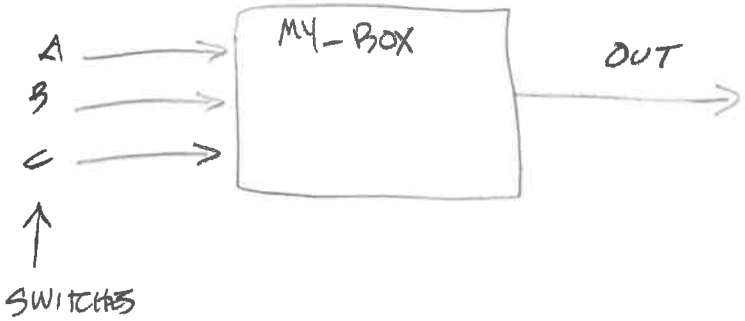
$$AA = \bar{A}B\bar{C} + AB\bar{C} + A\bar{B}\bar{C} + A\bar{B}C$$

$$BB = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C$$

$$CC = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC$$

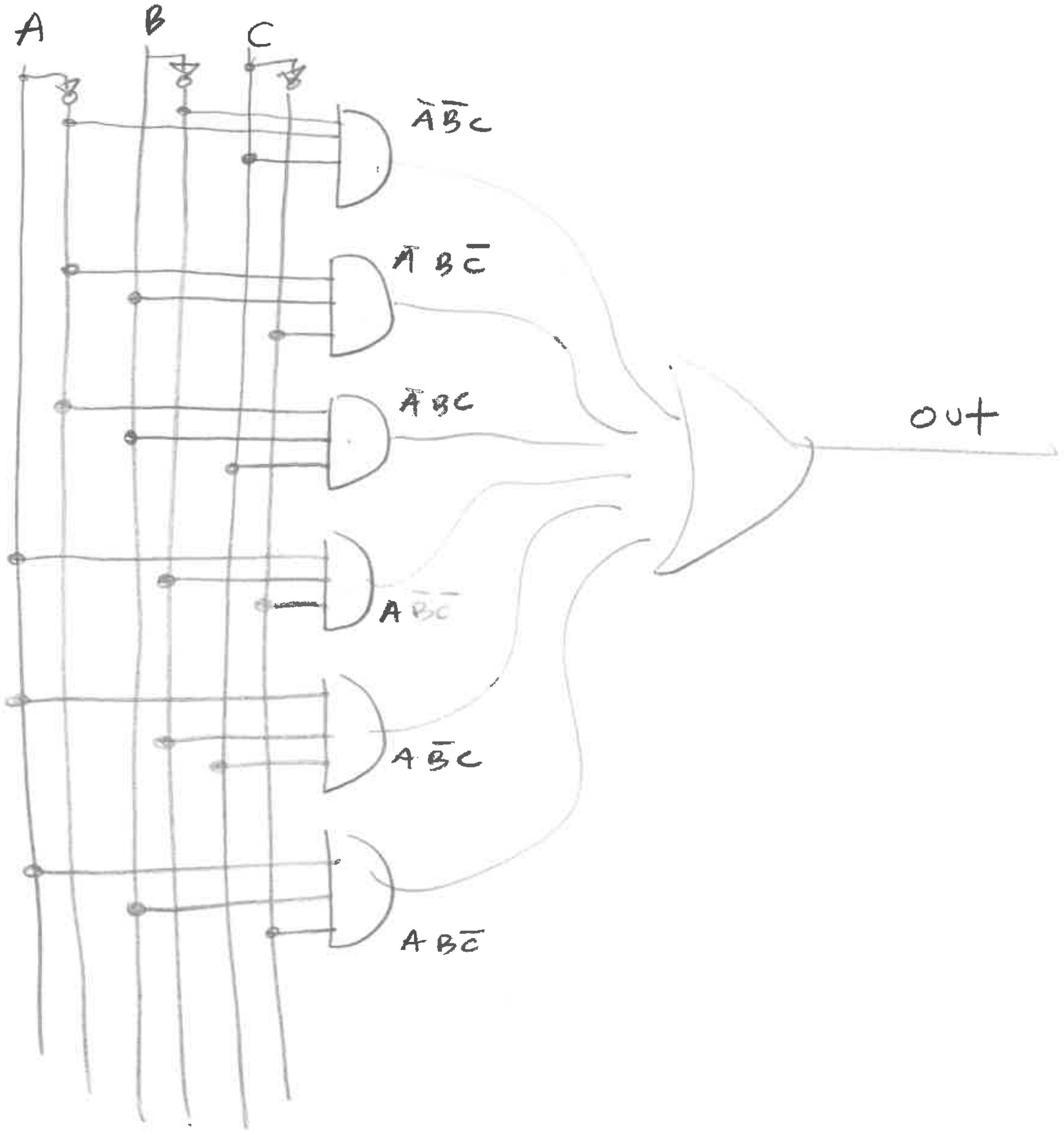
I simply don't feel like drawing it! way too BIG

4)

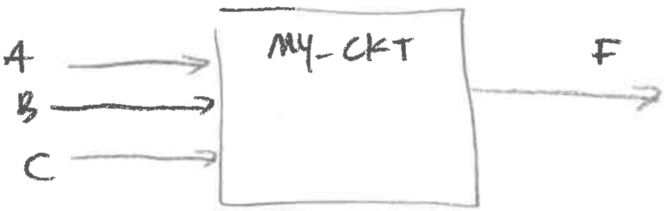


A	B	C	out
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$$OUT = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C}$$



5)

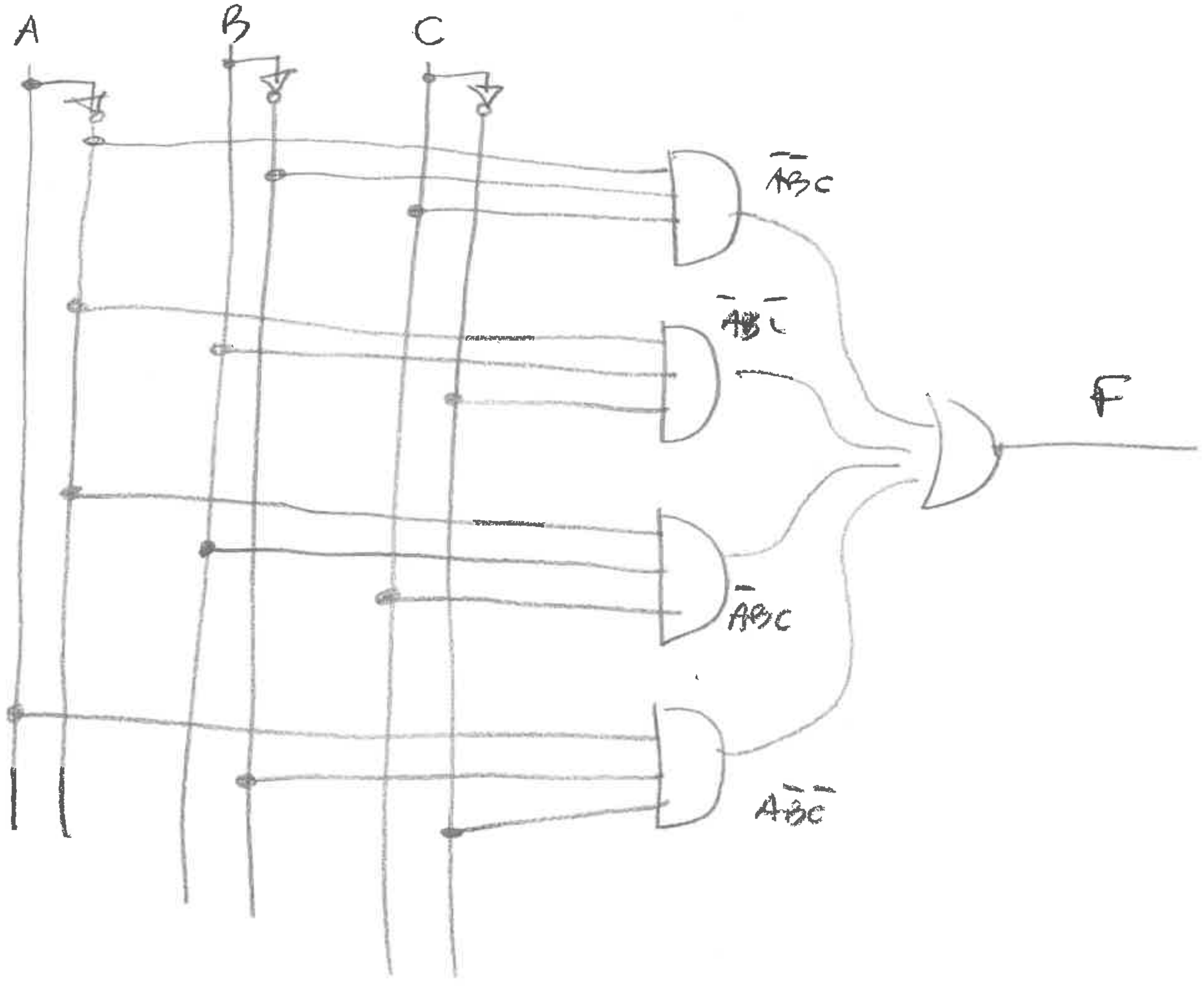


A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

A = MSB
C = LSB

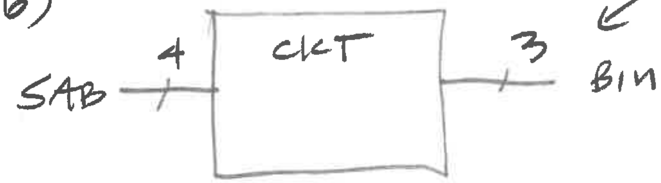
$F \leq 4$
$F \neq 0$

$$F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C}$$



CHAPTER 6

6)



must be 3-bit output Because maximum 4-bit storage unary value = 4

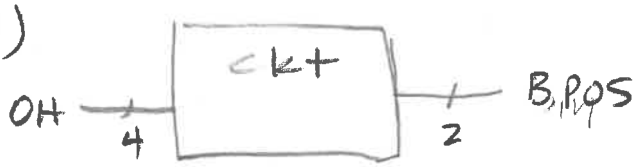
SAB				BINARY		
A	B	C	D	F_2	F_1	F_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	1	0	1	0
0	1	1	1	0	1	1
1	1	1	1	1	0	0

$$F_2 = A \cdot B \cdot C \cdot D$$

$$F_1 = \bar{A} \cdot \bar{B} \cdot C \cdot D + \bar{A} \cdot B \cdot C \cdot D$$

$$F_0 = \bar{A} \cdot B \cdot C \cdot D + \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot D$$

7)



OH				BPOS	
A	B	C	D	F_1	F_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

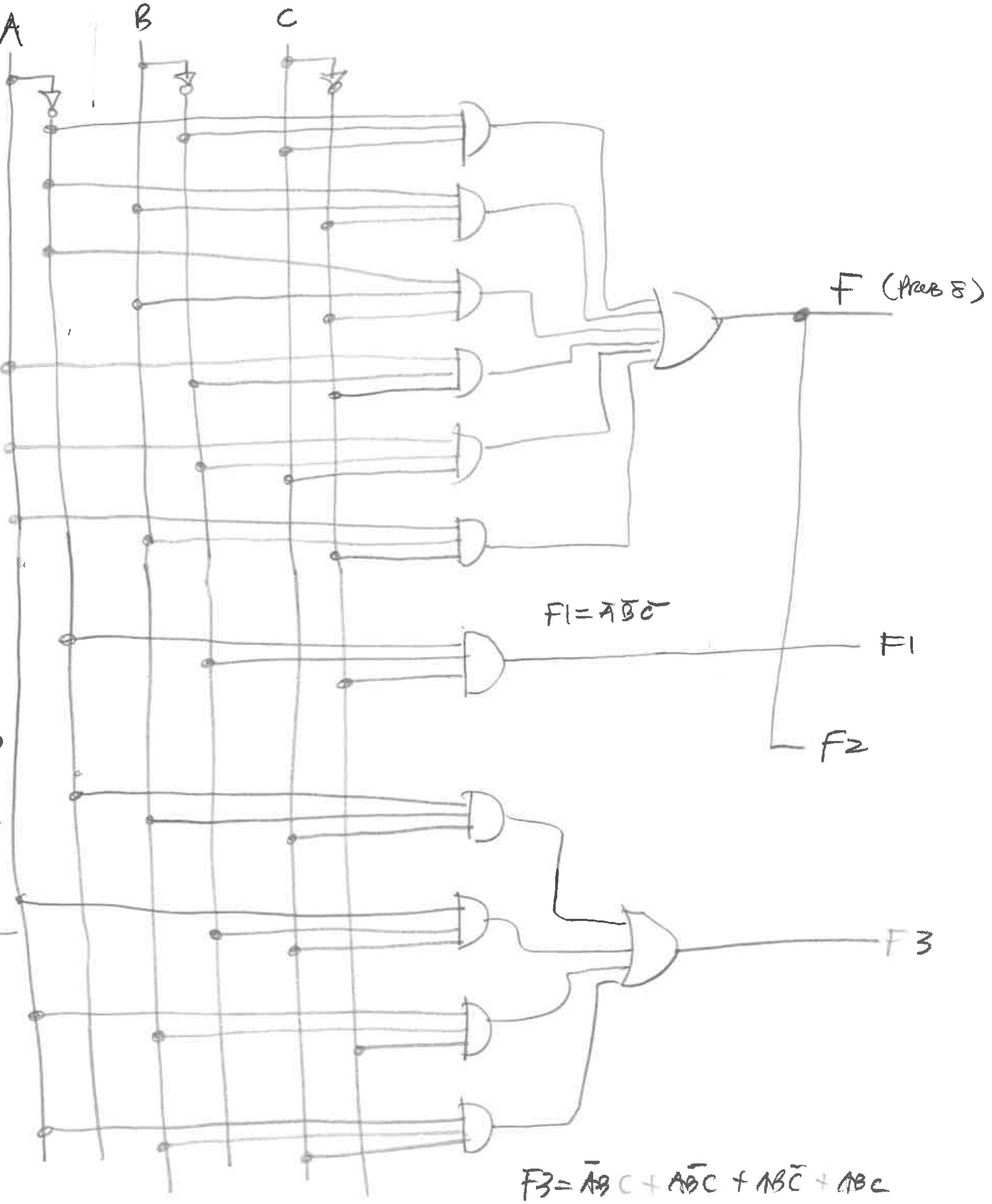
$$F_1 = \bar{A}B\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D}$$

$$F_0 = \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D}$$

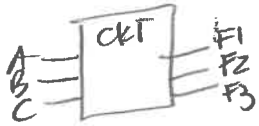
8)

A	B	C	F
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	0

$F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C}$



9)

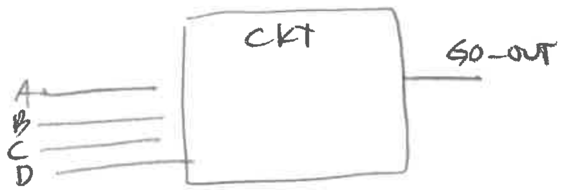


A	B	C	F1	F2	F3
0	0	0	1	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	0	0	1

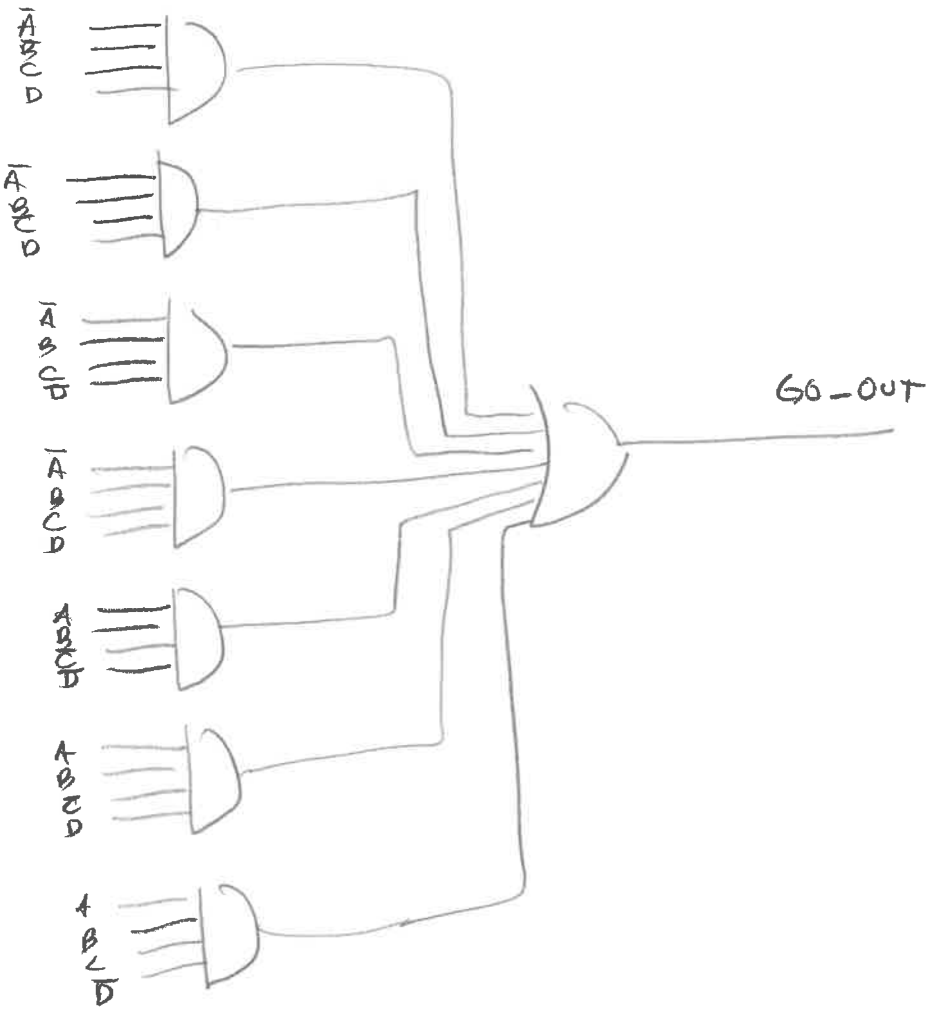
$F3 = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

10)

A	B	C	D	GOOUT
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	D
1	0	1	0	D
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	D

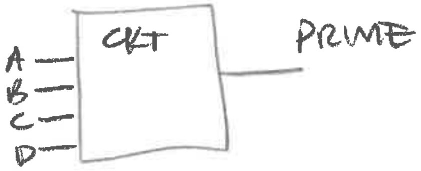


$$F = \overline{A}\,\overline{B}CD + \overline{A}B\overline{C}D + \overline{A}BC\overline{D} + \overline{A}BCD + AB\overline{C}\,\overline{D} + AB\overline{C}D + ABC\overline{D}$$


$\overline{A}$ $\overline{B}$ C D
$\overline{A}$ B $\overline{C}$ D
$\overline{A}$ B C $\overline{D}$
$\overline{A}$ B C D
A B $\overline{C}$ $\overline{D}$
A B $\overline{C}$ D
A B C $\overline{D}$
GO-OUT

11)

NUM ABCD	PRIME	P12
0000	—	0
0001	1	0
0010	1	0
0011	1	1
0100	0	1
0101	1	1
0110	0	1
0111	1	1
1000	0	1
1001	0	1
1010	0	1
1011	1	1
1100	0	0
1101	1	0
1110	0	0
1111	0	0



A = MSB
D = LSB

$F = \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D$

$\bar{A}BCD + A\bar{B}CD + AB\bar{C}D$

12)

A
B
C
D
CKT
P_12

$F = \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD$

$+ A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + A\bar{B}CD$

I REFUSE TO DRAW CKT

13)

SW	A	B	C	F
0	0	0	0	1
	0	0	1	1
	0	1	0	1
	0	1	1	1
	1	0	0	0
	1	0	1	0
	1	1	0	0
	1	1	1	0
1	0	0	0	1
	0	0	1	1
	0	1	0	1
	0	1	1	0
	1	0	0	0
	1	0	1	0
	1	1	0	0
	1	1	1	0

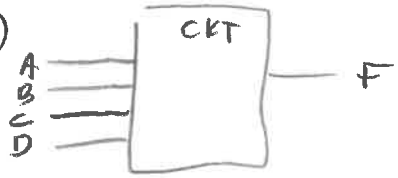
A = MSB
C = LSB

$F = \overline{SW}\,\bar{A}\bar{B}\bar{C} + \overline{SW}\,\bar{A}\bar{B}C + \overline{SW}\,\bar{A}B\bar{C}$

$+ \overline{SW}\,\bar{A}BC + SW\,\bar{A}\bar{B}\bar{C} + SW\,\bar{A}\bar{B}C$

$+ SW\,\bar{A}B\bar{C}$

14)



A	B	C	D	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

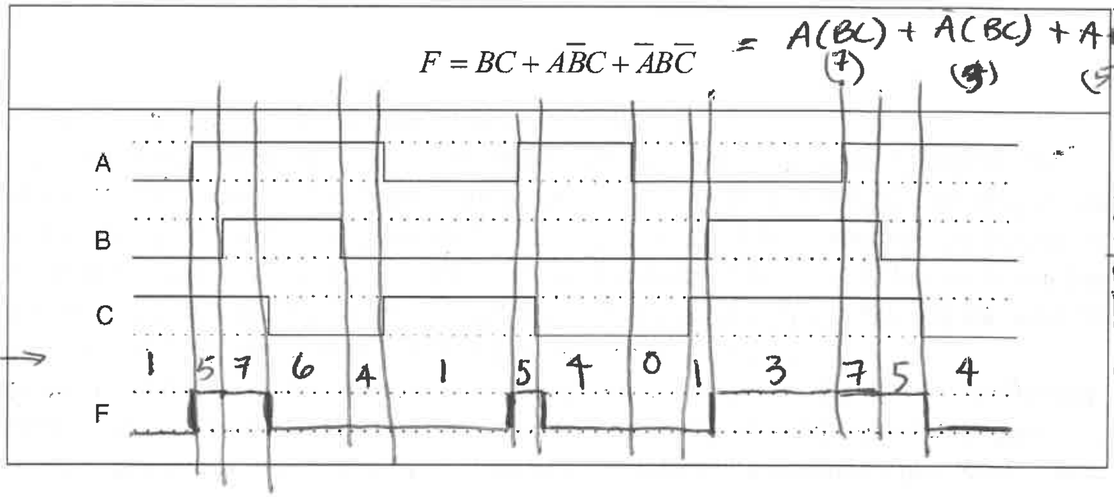
$$F = \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + A\bar{B}CD + AB\bar{C}\bar{D} +$$
$$AB\bar{C}D + ABC\bar{D} + ABCD$$

Chapter Exercises

1) Using the following Boolean equation to complete the accompanying timing diagram.

$F = BC + A\overline{B}C + \overline{A}B\overline{C}$ $= A(BC) + \overline{A}(BC) + A\overline{B}C + \overline{A}B\overline{C}$

(7) (3) (5) (2)



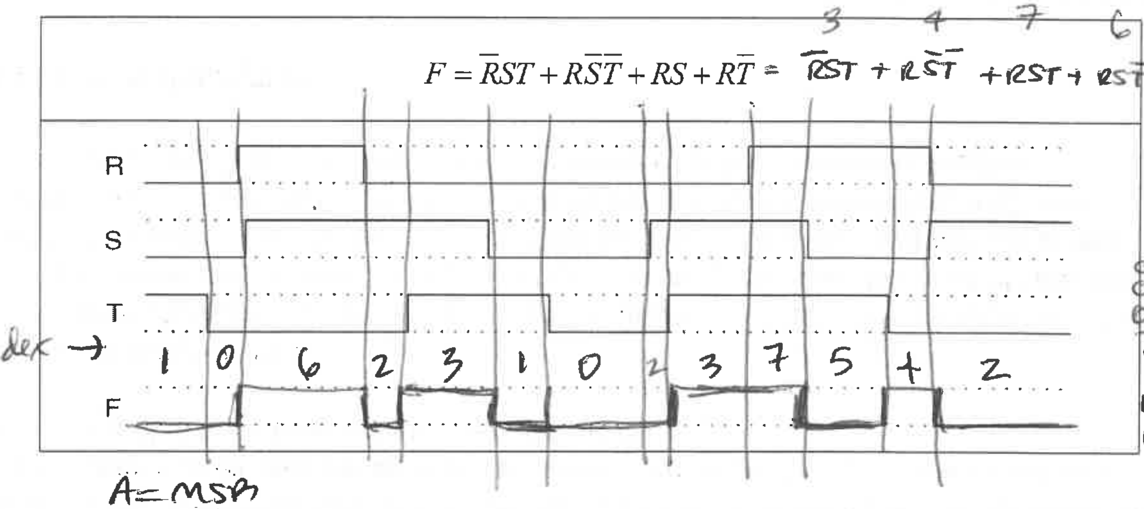
ABC	F
000	0
001	0
010	1
011	1
100	0
101	1
110	0
111	1

A=MSB

2) Using the following Boolean equation to complete the accompanying timing diagram.

3 4 7 6 6

$F = \overline{R}ST + R\overline{S}\overline{T} + RS + R\overline{T} = \overline{R}ST + R\overline{S}\overline{T} + RST + RS\overline{T} + RS\overline{T}$
$+ R\overline{S}\overline{T} > 4$



RST	F
000	0
001	0
010	0
011	1
100	1
101	0
110	1
111	1

A=MSB

3) Does the timing diagram listed below completely define a function? Why or why not? If it does, write both SOP and POS equations that describes the function and provide a circuit diagram in both SOP and POS form that could be used to implement the circuit.

A B C	F
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	0
1 0 1	1
1 1 0	1
1 1 1	1

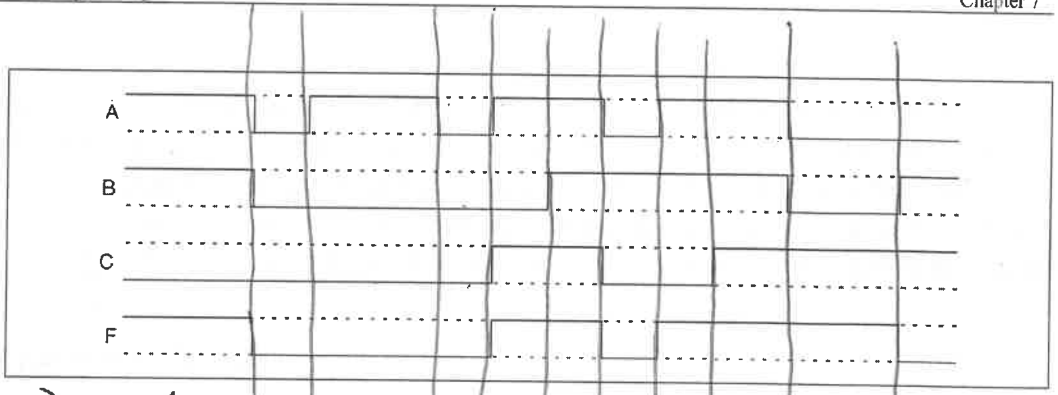
$F = \overline{A}\overline{B}C + A\overline{B}C + AB\overline{C} + ABC$ (SOP)

* TOO LONG TO DRAW CKT DIAGRAM

$\overline{F} = \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\overline{C}$

$F = (A+B+C)(A+\overline{B}+C)(A+\overline{B}+\overline{C})(\overline{A}+B+C)$ POS

A=MSB



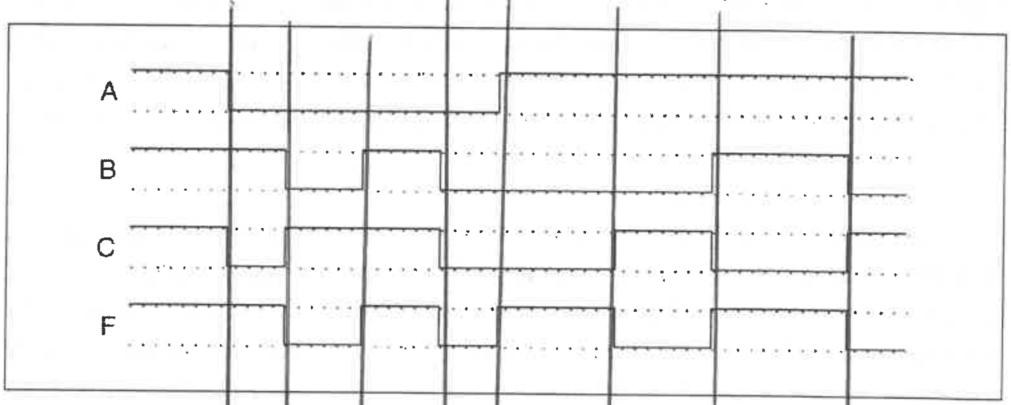
index → 6 | 0 | 4 | 0 | 5 | 7 | 2 | 6 | 7 | 1 | 3

A=MCB

4) The following timing diagram may completely model a function.

- If the timing diagram defines a function, draw a circuit diagram for the function in reduced form. You Don't Know This Yet
- If the timing diagram does not define a function, explicitly describe why it does not.

A	BC	F
0	00	0
0	01	0
0	10	1
0	11	1
1	00	1
1	01	0
1	10	1
1	11	1



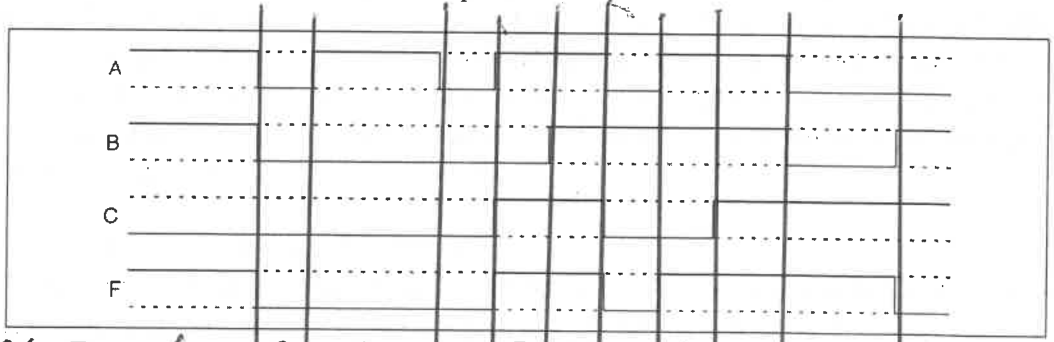
7 | 2 | 1 | 3 | 0 | 4 | 5 | 6 | 5

$F = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + AB\bar{C} + ABC$

TOO LONG TO DRAW CKT

5) Consider the previous problem… can you safely state which of the inputs variables is the MSB or LSB? Be sure to provide a complete explanation. NO, You can't. But… AS LONG AS You ARE consistent, IT DOES not matter

6) Does the timing diagram listed below completely define a function? Why or why not? If it does, write both SOP and POS equations that describes the function and provide a circuit diagram in both SOP and POS form that could be used to implement the circuit.



ABC	F
000	0
001	1
010	0
011	0
100	0
101	1
110	1
111	1

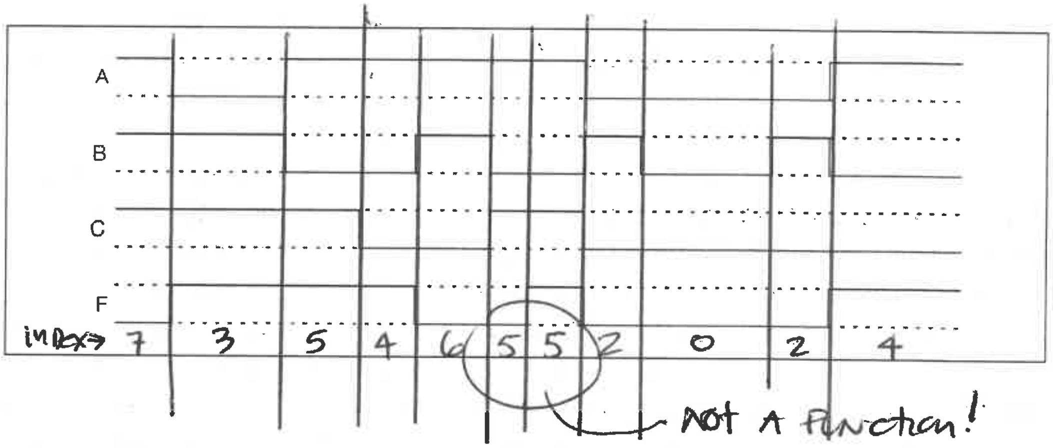
INDEX → 6 | 0 | 4 | 0 | 5 | 7 | 2 | 6 | 7 | 1 | 3

A=MSB

$F = \bar{A}\bar{B}C + A\bar{B}C + AB\bar{C} + ABC$ (SOP) ≋ $F = (A+B+C)(A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+B+C)$ POS

7) Consider the previous problem… how does the ordering of the labels of A, B, and C change the outcome of the problem? Be sure to provide a complete explanation. THE ORDER DOES not matter so long as you are consistent in considering positions.

8) Does the timing diagram listed below completely define a function? Why or why not? If it does, write both SOP and POS equations that describes the function and provide a circuit diagram in both SOP and POS form that could be used to implement the circuit.

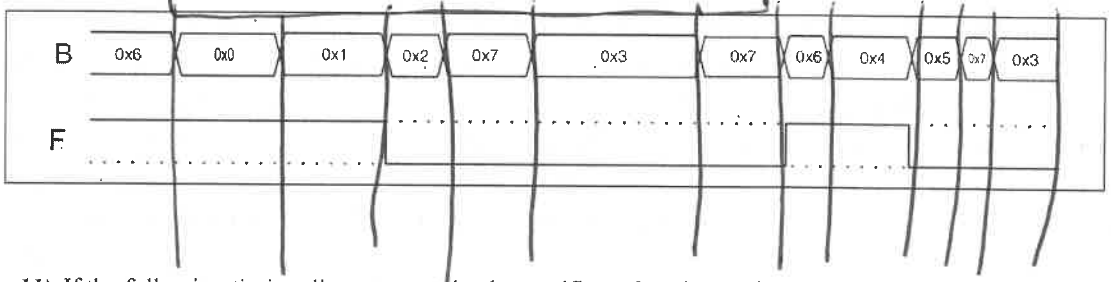


not A function!

9) In your own words, under what conditions could the timing diagram of the previous problem ever be used in a real circuit setting? A CIRCUIT THAT DOES not REQUIRE A STABLE output.

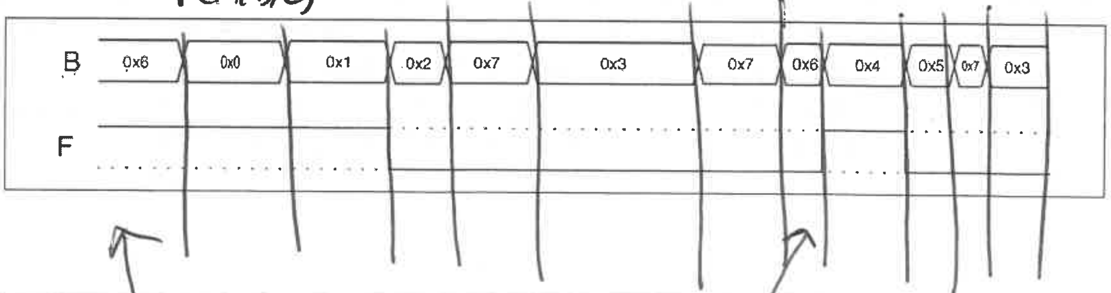
10) If the following ~~timing diagram completely specifies a function,~~ write a Boolean expression for that function. $\boxed{F(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + AB\bar{C}}$

ABC	F
000	1
001	1
010	0
011	0
100	1
101	0
110	1
111	0



11) If the following timing diagram completely specifies a function, write a Boolean expression for that function. F(A,B,C) =

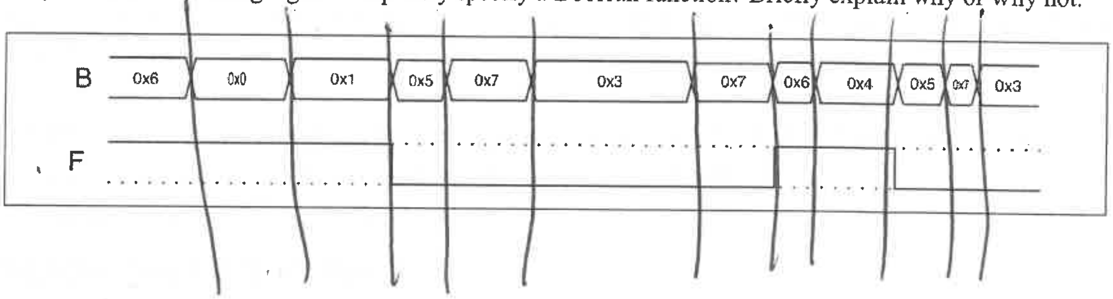
~~ABC~~	~~F~~
~~000~~	~~1~~
~~001~~	~~1~~
~~010~~	~~0~~
~~011~~	~~0~~
~~100~~	
~~101~~	
~~110~~	
~~111~~	~~0~~



not A Function!

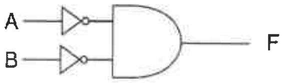
"010" is missing: not a function

12) Does the following signal completely specify a Boolean function? Briefly explain why or why not.

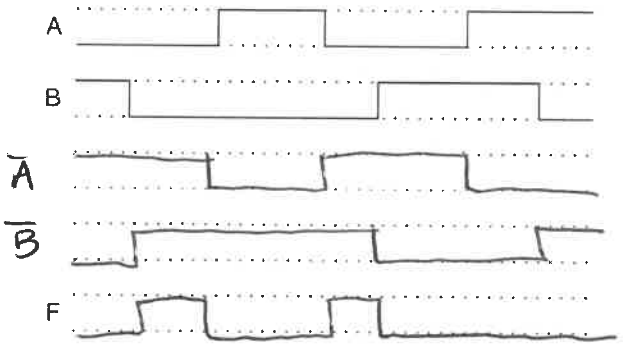


ABC	F
000	1
001	1
010	?
011	0
100	1
101	0
110	1
111	0

13) Complete the following timing diagram for the F output based on the given circuit.



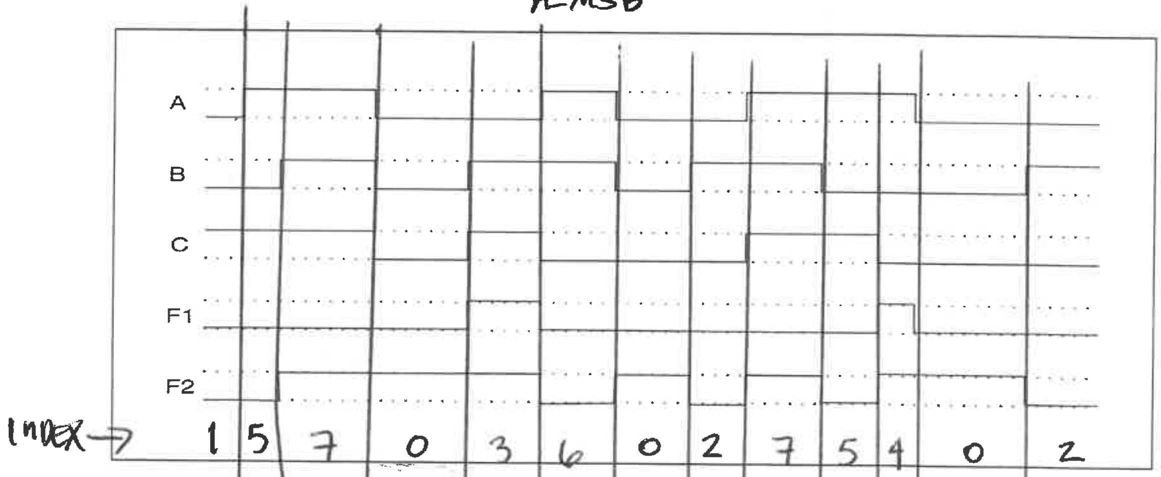
ABC	F1	F2
000	0	1
001	0	0
010	0	0
011	1	1
100	1	1
101	0	0
110	0	0
111	0	1



✓

14) For this problem, consider the input variables to be A, B, and C and the outputs to be F1 and F2. The timing diagram below completely described functions F1 and F2. Write a Boolean expressions that describe F1 and F2

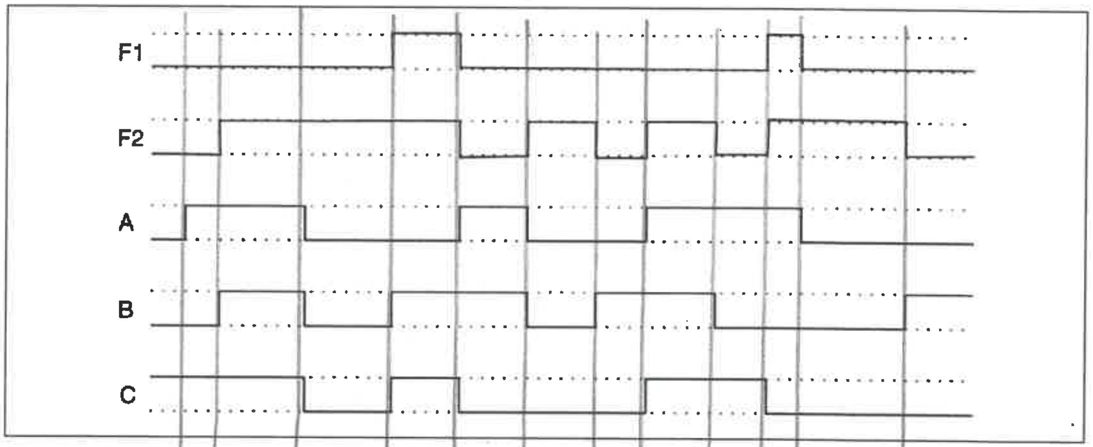
A=MSB



$$\boxed{F1 = A\bar{B}\bar{C} + \bar{A}BC}$$

$$\boxed{F2 = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + ABC}$$

15) For this problem, consider the input variables to be A, B, and C and the outputs to be F1 and F2. The timing diagram below completely described functions F1 and F2. Write a Boolean expressions that describe F1 and F2. $F1(ABC) = \Sigma(3,4) = \bar{A}BC + A\bar{B}\bar{C}$



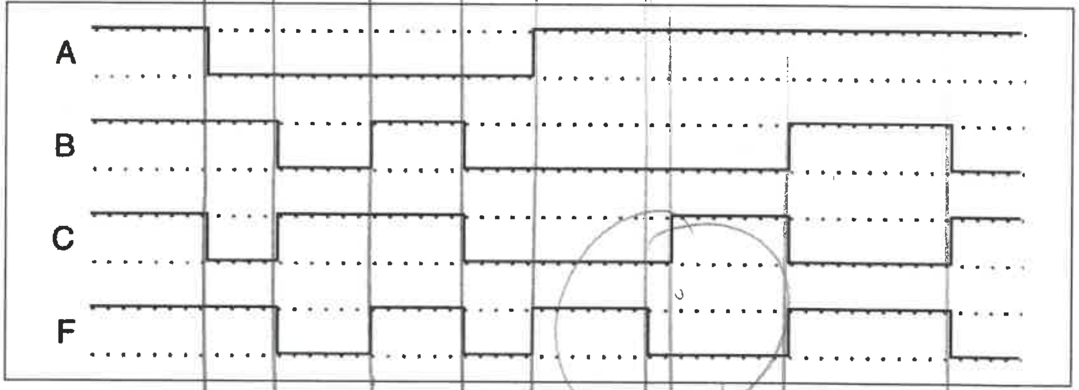
F

index → 1 5 7 0 3 6 0 2 7 5 4 0 2

$F2(ABC) = (0,3,4,7,$
$= \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + ABC$

16) The following timing diagram may completely model a function.

- If the timing diagram defines a function, draw a circuit diagram for the function in reduced form.
- If the timing diagram does not define a function, explicitly describe why it does not.

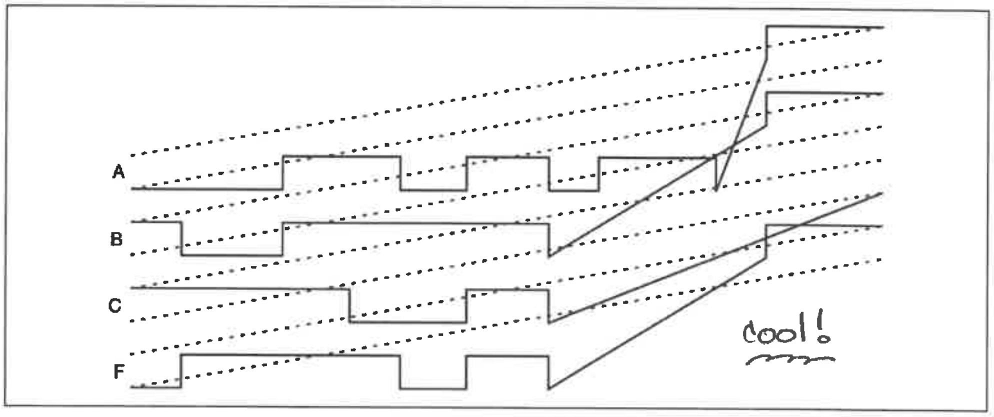


7 2 1 3 0 4

NOT A FUNCTION!

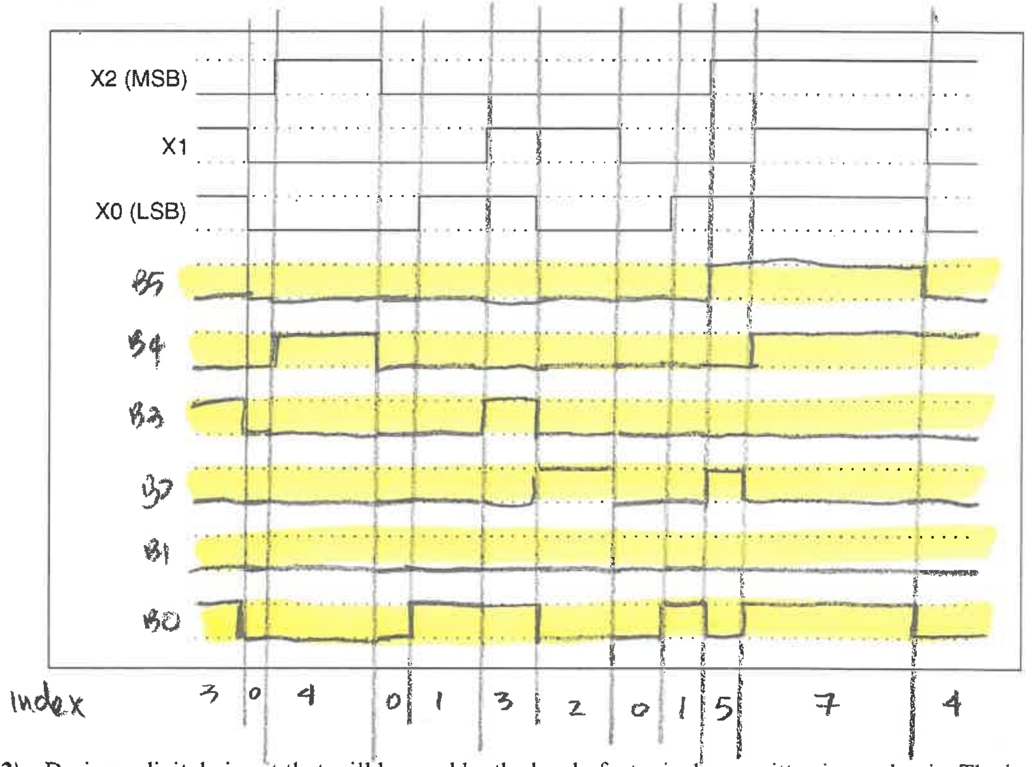
output changed without input change

17) For those aspiring digital designers on drugs, state whether the timing diagram listed below completely defines a function. Why or why not? Does anyone really freaking care?



Design Problems

1) Design a circuit whose output represents a square of the input. For this problem, describe your design using SOP or POS equations. Also, waste yet even more time by completing the timing diagram listed below.



2) Design a digital circuit that will be used by the head of a typical committee in academia. The input labeled "A-HOLE" is the head of the committee; the other two committee members are labeled "GOOD1" and "GOOD2". Being a typical head of a committee, the chairman of the committee has commissioned you to build this circuit in order to better serve himself. The committee has a set of switches that are used for a "secret" vote. Your mission is to modify the circuit inputs such that there is always a majority in any way the head of the committee votes. Provide a truth table and equations for your circuit; also, complete the following timing diagram in order to prove that you may know what you're doing.

See NEXT PAGE

X2	X1	X0	B5	B4	B3	B2	B1	B0
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	0	0
1	0	1	0	1	1	0	0	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0	1

$B5 = X2\,X1\,\overline{X0} + X2\,X1\,X0$

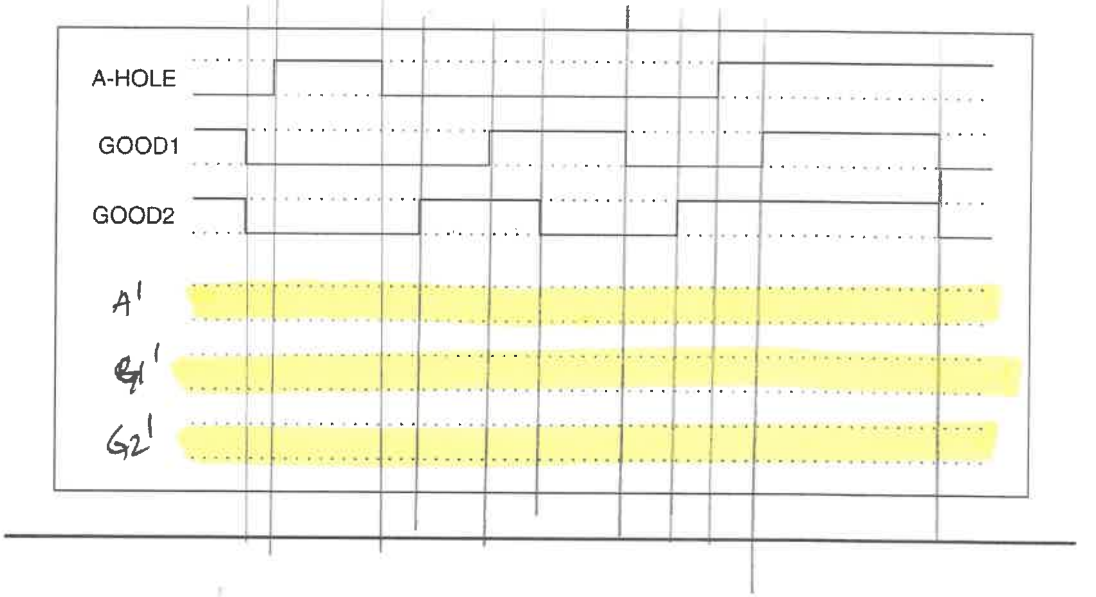
$B4 = X2\,\overline{X1}\,\overline{X0} + X2\,\overline{X1}\,X0 + X2\,X1\,X0$

$B3 = \overline{X2}\,X1\,X0 + X2\,\overline{X1}\,X0$

$B2 = \overline{X2}\,X1\,\overline{X0} + X2\,X1\,\overline{X0}$

$B1 = 0$

$B0 = X2\,X1\,X0 + X2\,X1\,X0 +$
$X2\,X1\,X0 + X2\,X1\,X0$



A	G1	G2	A'	G1'	G2'
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	1	1	0
1	1	1	1	1	1

two ZERO = majority

two one's = majority

$$\boxed{A' = A}$$

$$\boxed{G1' = \overline{A}\,G1\,\overline{G2} + A\,G1\,\overline{G2} + A\,G1\,G2}$$

$$\overline{G2'} = \overline{A}\,\overline{G1}\,\overline{G2} + \overline{A}\,G1\,\overline{G2} + A\,G1\,\overline{G2}$$

$$\boxed{G2' = (A + G1 + G2)(A + \overline{G1} + G2)(\overline{A} + \overline{G1} + G2)}$$

CHAPTER 8 EXERCISES

1) LEAVING INPUTS FLOATING HAS 2 MAIN ISSUES:

a) NOT Connecting THEM Could AFFECT CIRCUIT operation

b) It will MAKE people wonder IF You simply FORGOT TO complete THE circuit OR INTENTIONALLY DID NOT CONNECT inputs

2) A 10-Bit RCA would HAVE 2 10-bit INputs. SO THERE would be 10+10 = 20 TOTAL INPUTS. SO A TRUTH TABLE would HAVE 2^{20} ROWS

3) Look AHEAD MEANS THAT You Don't HAVE to wait FOR THE CARRY TO RIPPLE THROUGH ALL THE BITS until THE ANSWER IS "READY".

4) THE CARRY FROM LOWER BITS "works its way", OR RIPPLES, to THE upper Bits AS NEEDED.

Chapter Exercises

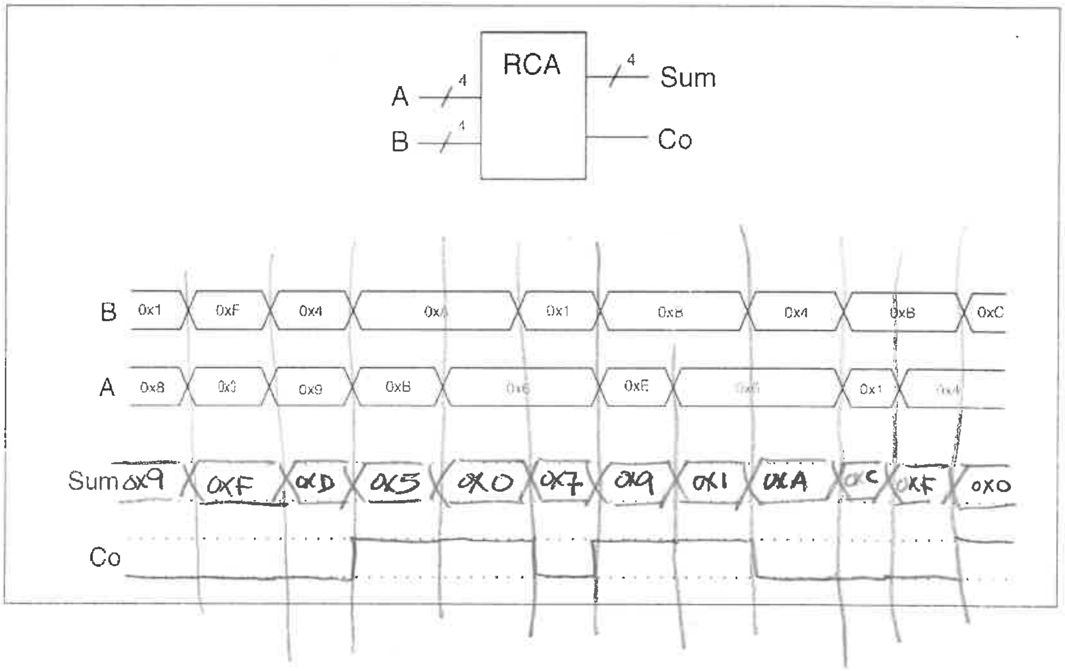
1) Briefly describe why we should always connect all unused input signals to either power or ground in all digital designs. In other words, why do we not what to "leave inputs hanging" or "leave inputs floating".

2) If you were to design a 10-bit RCA using the BFD approach, briefly explain how many rows with the associated truth table have?

3) There are adders out there that fall into the category of "look ahead carry" adders. Briefly explain why these would output a result faster than a RCA.

4) In your own words, briefly explain how the RCA got its name.

5) Complete the timing diagram shown below considering the given schematic symbol.

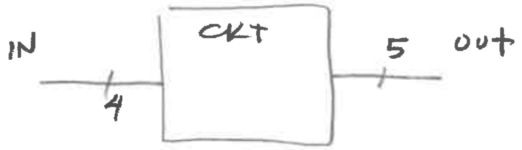


1011
1110
1001

1010
1011
0101

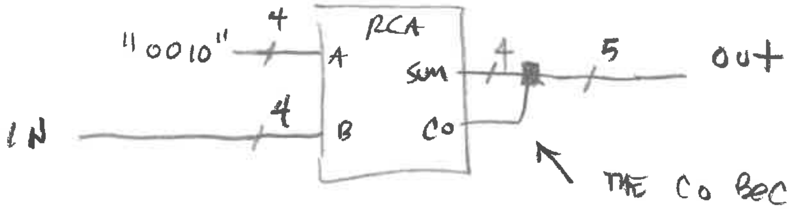
CHAPTER 8 DESIGN Problem

1)



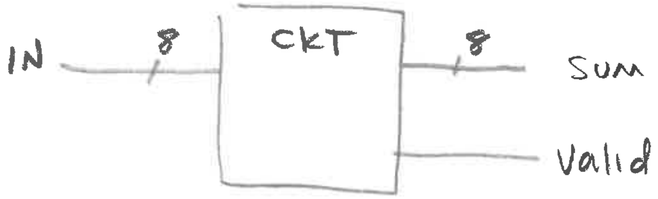
TO ENSURE THE output is ALWAYS valid, THE WIDTH OF THE output must Be 1 Bit WIDER THAN INPUT

no control

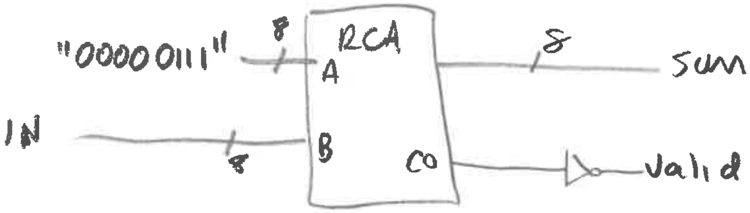


THE Co Becomes THE MSB

2)

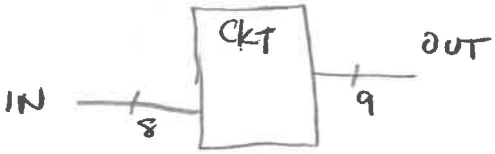


NO Control



when there is a carry out The sum output is not valid; otherwise it is

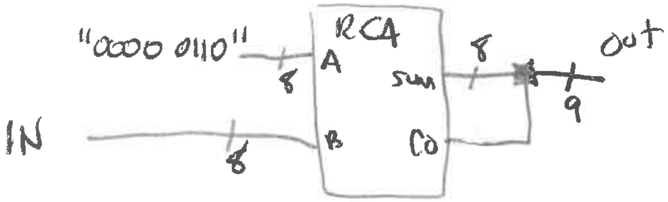
3)



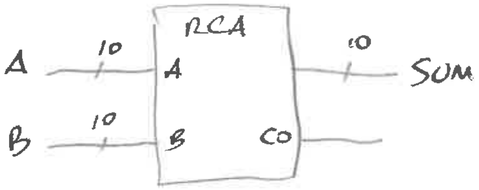
TO ENSURE sum output is Valid it must Be 9-bits wide.

THE Co Becomes THE MSB of THE 9-bit outputs

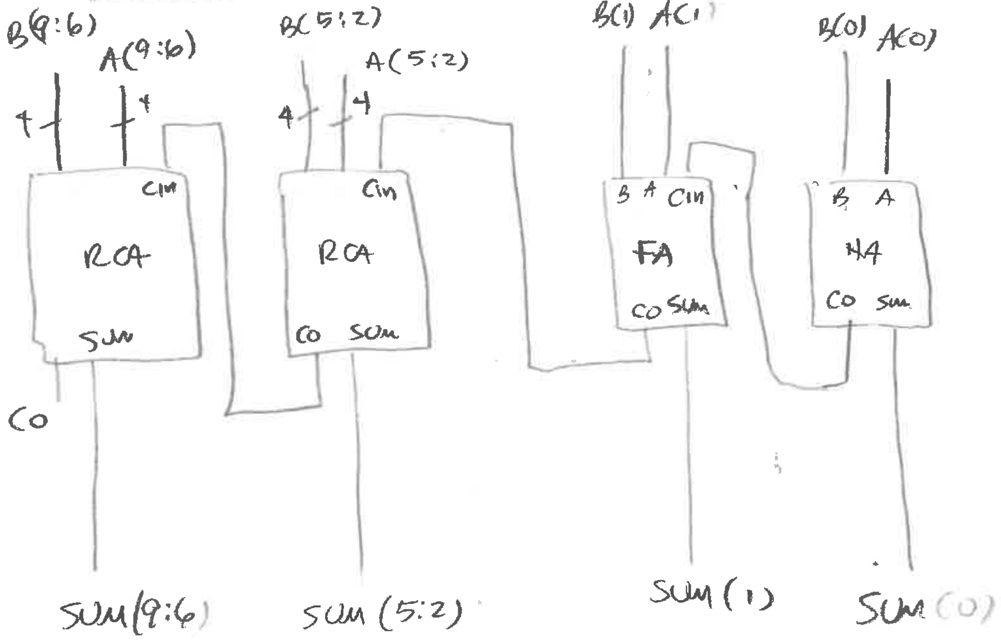
NO Control



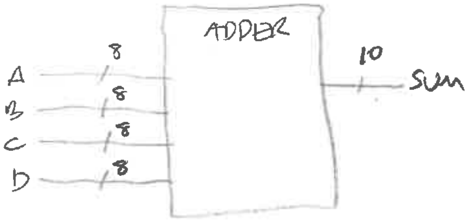
4)



No control



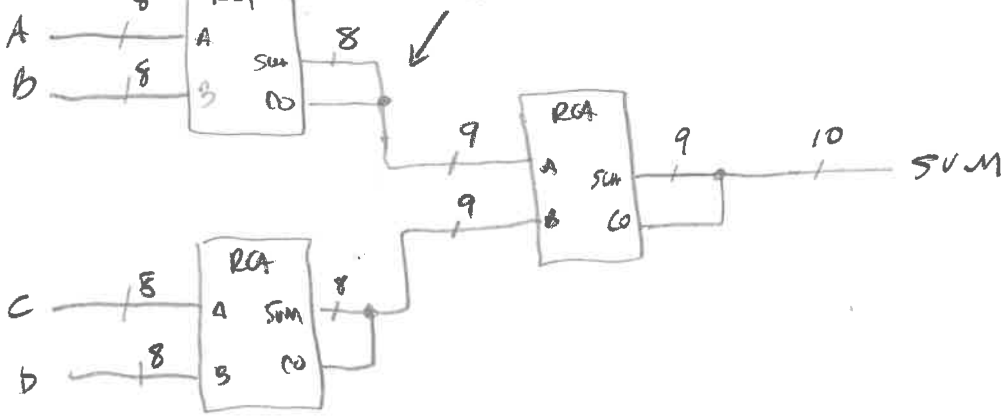
5)



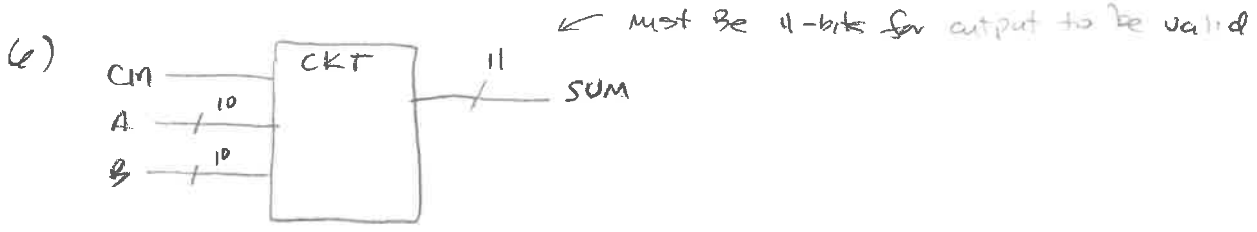
OUTPUT WIDTH MUST BE 10-bits in ORDER FOR THE OUTPUT TO BE Valid.

NO control

COs Become MSBs of SUM output

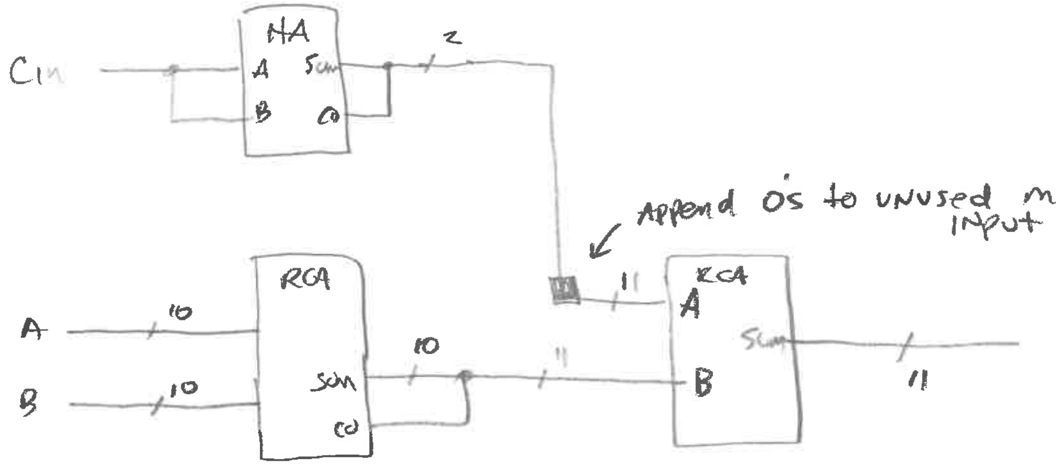


6)



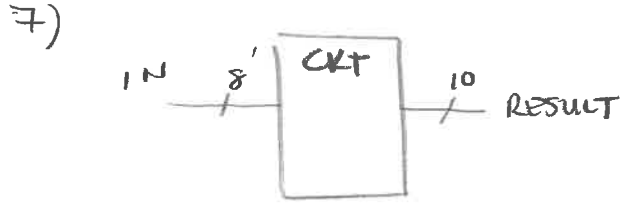
← Must Be 11-bits for output to be valid

NO control

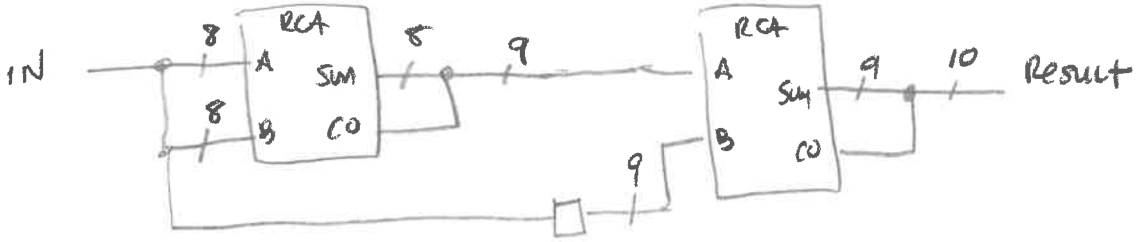


Append 0's to unused most significant input Bits

7)

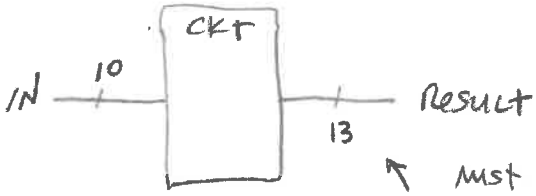


NO control



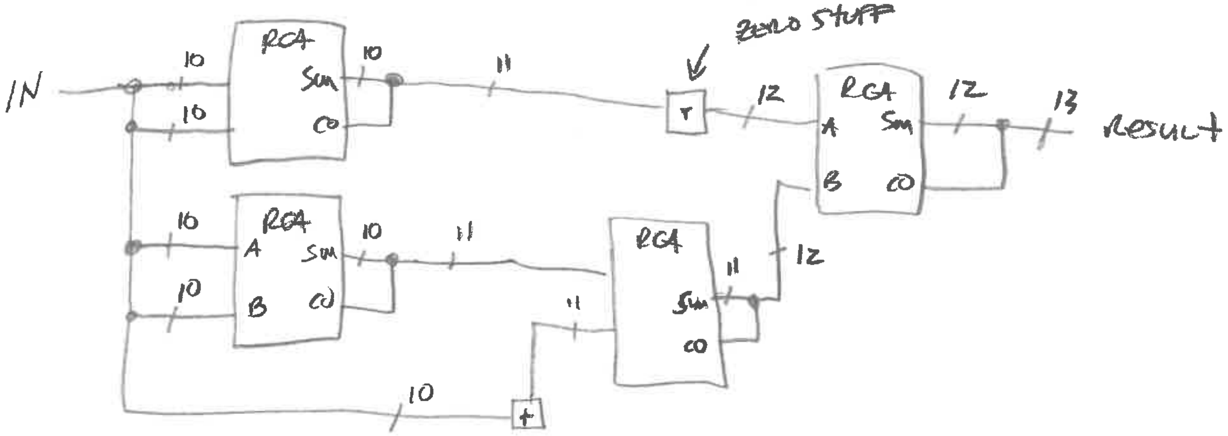
Append a '0' to MSB

8)



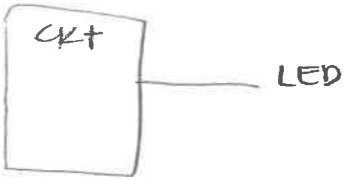
NO Control

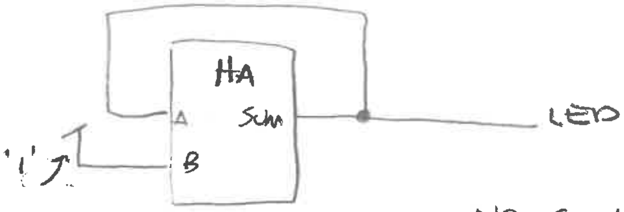
↖ Must Be 3 bits wider for valid result



↑ Append A MSB OF '0' (Zero "STUFFING")

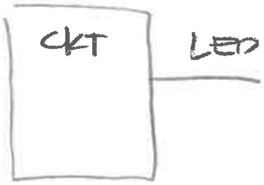
9)

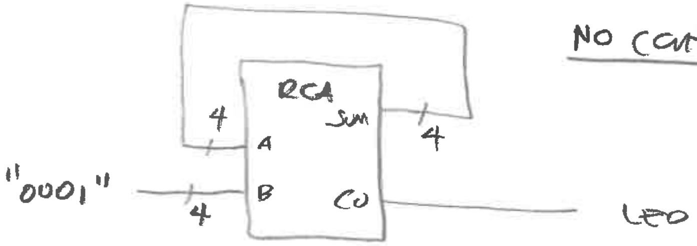




NO Control

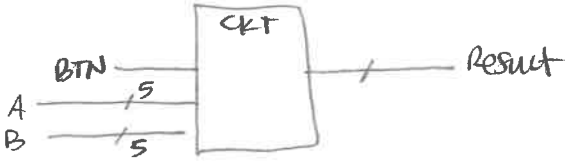
10)



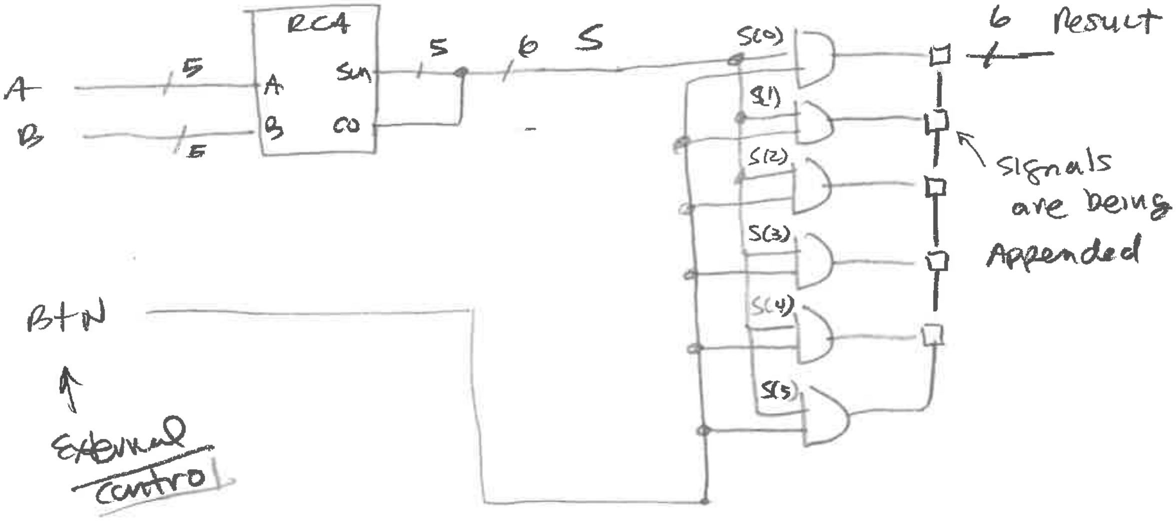


NO control

11)



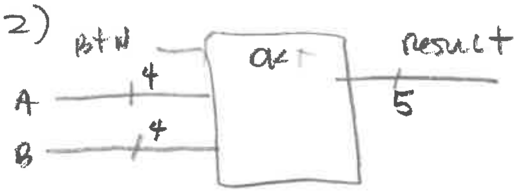
EXTERNAL control



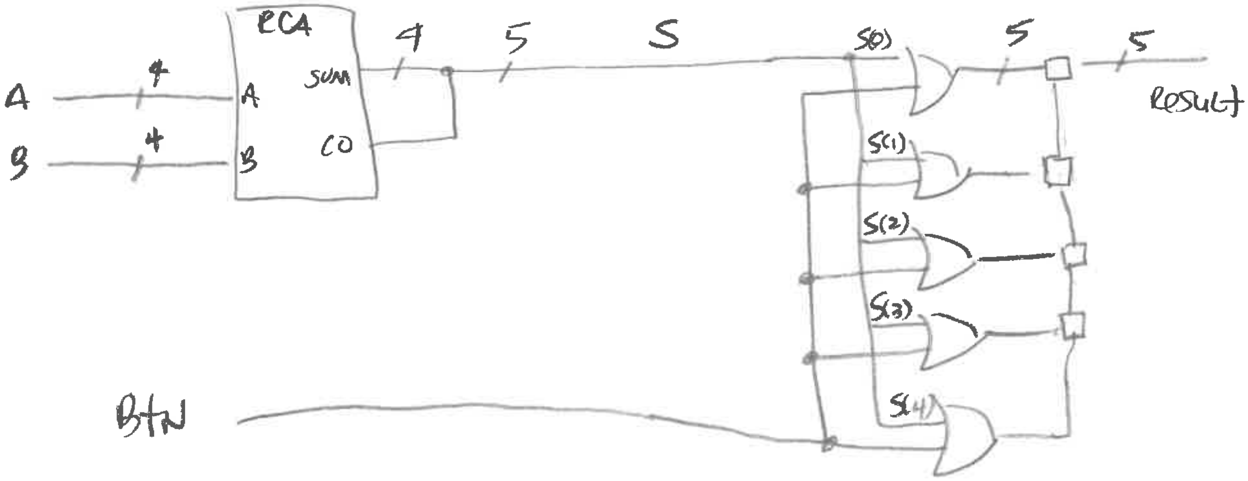
signals are being Appended

External control

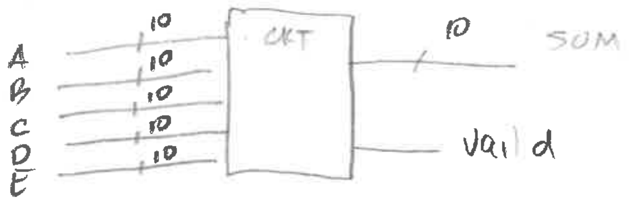
12)



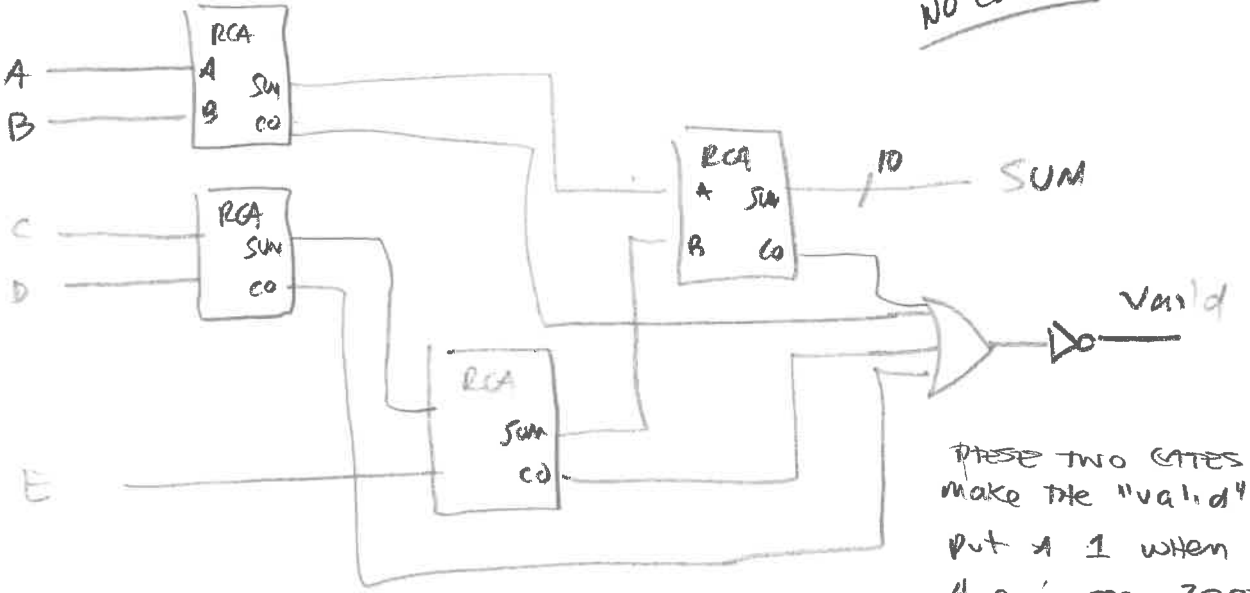
EXTERNAL control



13)

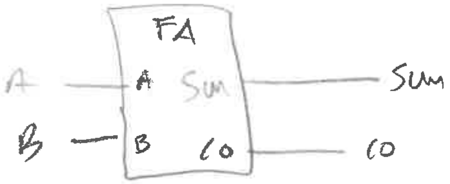


No control



these two gates make the "valid" output a 1 when all 4 co's are zero

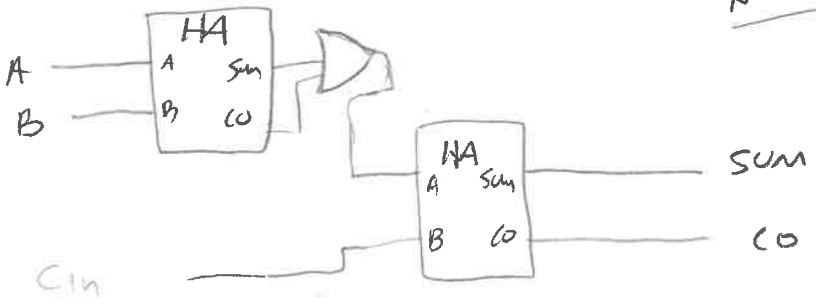
14)



A	B	Sum	Co
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Cin	A	B	Sum	Co
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

No control



CHAPTER 9 Exercises

1) If you did not list all independent variables, you would not know the proper form of standard SOP & POS forms

2) a) one form uses less hardware to implement a circuit thus saving money & power, & board space

b) You may have a bunch of one-type of logic chip you need to use

c) Your Boss told you to do it one way or another

3) a) $\bar{F} = \overline{B2}\,\overline{B1}\,\overline{B0} + \overline{B2}B1B0 + B2\overline{B1}B0 + B2B1\overline{B0} + B2B1B0$

$F = (B2+B1+B0)(B2+\overline{B1}+\overline{B0})(\overline{B2}+B1+B0)(\overline{B2}+\overline{B1}+B0)(\overline{B2}+\overline{B1}+\overline{B0})$

b) $\bar{F} = \bar{A}\bar{B}C + A\bar{B}\bar{C} + AB\bar{C} + ABC$

$F = (A+B+\bar{C})(\bar{A}+B+C)(\bar{A}+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$

c) $\bar{F} = \bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}Y\bar{Z} + XY\bar{Z} + XYZ$

$F = (X+Y+Z)(X+Y+\bar{Z})(X+\bar{Y}+Z)(\bar{X}+\bar{Y}+Z)(\bar{X}+\bar{Y}+\bar{Z})$

d) $\overline{F1} = \bar{t}u\bar{v} + \bar{t}uv + t\bar{u}\bar{v} + t\bar{u}v + tuv$

$F1 = (t+\bar{u}+v)(t+\bar{u}+\bar{v})(\bar{t}+u+v)(\bar{t}+u+\bar{v})(\bar{t}+\bar{u}+\bar{v})$

$\overline{F2} = \bar{t}\bar{u}\bar{v} + \bar{t}\bar{u}v + t\bar{u}\bar{v} + t\bar{u}v$

$F2 = (t+u+v)(t+u+\bar{v})(\bar{t}+u+v)(\bar{t}+u+\bar{v})$

4)

A B C	(a)	(b)	(c)
X Y Z			
0 0 0	1	0	0
0 0 1	0	1	1
0 1 0	0	1	0
0 1 1	1	0	0
1 0 0	0	1	1
1 0 1	1	1	0
1 1 0	1	0	1
1 1 1	0	1	1

5

RST			
ABC			
XYZ	a	b	c
0 0 0	0	0	1
0 0 1	0	1	1
0 1 0	1	0	0
0 1 1	1	0	1
1 0 0	0	1	1
1 0 1	0	1	0
1 1 0	1	0	0
1 1 1	0	1	0

a) $\bar F = \bar A\bar B C + \bar A B\bar C + A\bar B\bar C + ABC$

$F = (A+B+\bar C)(A+\bar B+C)(\bar A+B+C)(\bar A+\bar B+\bar C)$

b) $\bar F = \bar A\bar B\bar C + \bar A BC + AB\bar C$

$F = (A+B+C)(A+\bar B+\bar C)(\bar A+\bar B+C)$

c) $\bar F = \bar X\bar Y\bar Z + \bar X Y\bar Z + \bar X YZ + X\bar Y Z$

$F = (X+Y+Z)(X+\bar Y+Z)(X+\bar Y+\bar Z)(\bar X+Y+\bar Z)$

5) a) $\bar F = \underset{7}{RST} + \underset{5}{R\bar S T} + \underset{4}{R\bar S\bar T} + \underset{2}{\bar R S\bar T} + \underset{0}{\bar R\bar S\bar T}$ ← THESE ARE THE ROWS WITH 0's

$F = \bar R S\bar T + \bar R S T + RS\bar T$

b) $\bar F = \underset{0}{\bar A\bar B\bar C} + \underset{2}{\bar A B\bar C} + \underset{3}{\bar A BC} + \underset{6}{AB\bar C}$

$F = \bar A\bar B C + A\bar B\bar C + A\bar B C + ABC$

c) $\bar F = \underset{6}{XY\bar Z} + \underset{5}{X\bar Y Z} + \underset{2}{\bar X Y\bar Z} + \underset{7}{XYZ}$

$F = \bar X\bar Y\bar Z + \bar X\bar Y Z + \bar X YZ + X\bar Y\bar Z$

6) A) $F = A\bar{B} + \bar{A}\bar{C}$

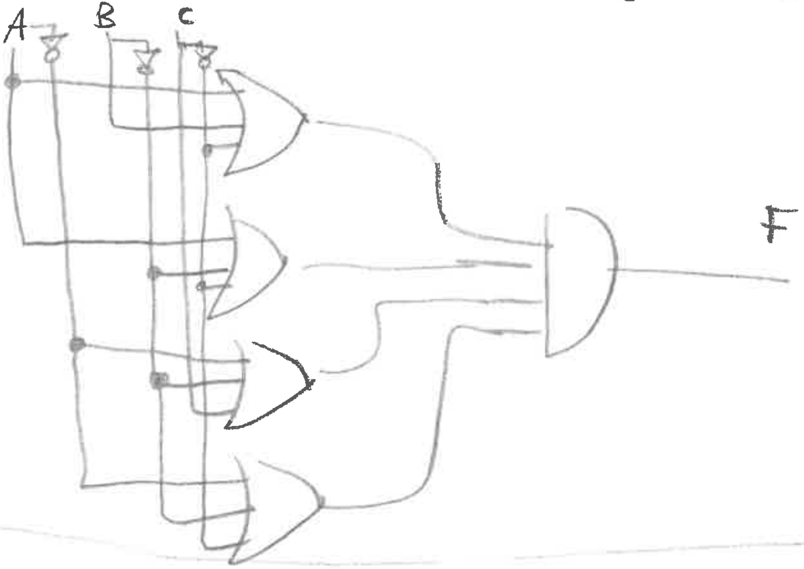
$= A\bar{B}(C+\bar{C}) + \bar{A}\bar{C}(B+\bar{B})$

$A\bar{B}C + A\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}\bar{C} = \Sigma(0,2,4,5)$

$= \Pi(1,3,6,7)$

$\bar{F} = \bar{A}\bar{B}C + \bar{A}BC + AB\bar{C} + ABC$

$F = (A+B+\bar{C})(A+\bar{B}+\bar{C})(\bar{A}+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$



b) $F = (A+\bar{C})(A+B+\bar{C})(\bar{B}+\bar{C})$

$\bar{F} = \bar{A}C + \bar{A}\bar{B}C + BC$

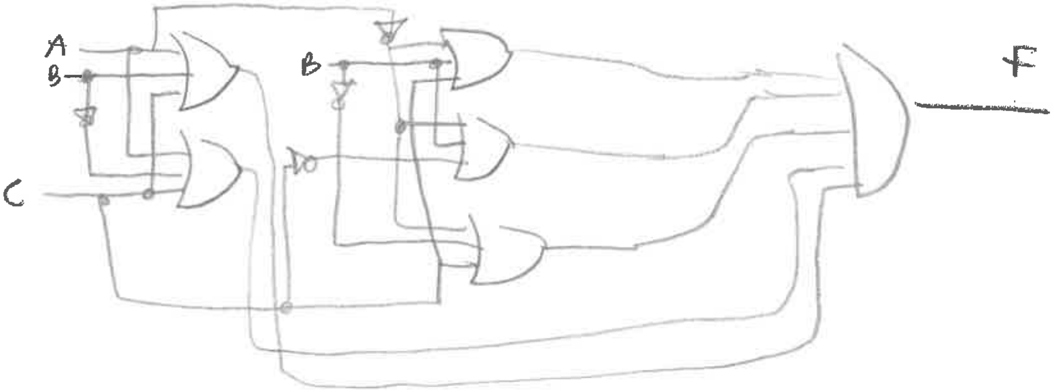
$\bar{F} = \bar{A}C(B+\bar{B}) + ABC + BC(A+\bar{A})$

$= \bar{A}BC + \bar{A}\bar{B}C + \cancel{ABC} + ABC + \cancel{\bar{A}BC} = \Sigma(1,3,7)$

$\Pi(0,2,4,5,6)$

$\bar{F} = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C}$

$F = (A+B+C)(A+\bar{B}+C)(\bar{A}+B+C)(\bar{A}+B+\bar{C})(\bar{A}+\bar{B}+C)$

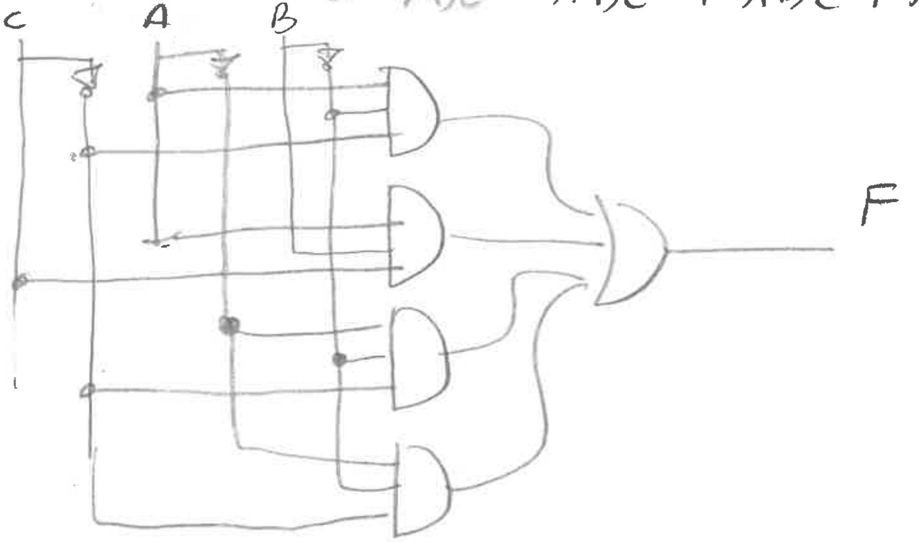


7) $\bar{F} = (A+\bar{B})(\bar{A}+\bar{C})$

$F = \bar{A}B + AC$

$= \bar{A}B(C+\bar{C})\ AC(B+\bar{B})$; Expand to Stardard SOP

$= \bar{A}BC + \bar{A}B\bar{C} + ABC + A\bar{B}C$



8) $\bar{F}(RST) = RST + R\bar{S}T + R\bar{S}\bar{T} + \bar{R}S\bar{T} + \bar{R}\bar{S}\bar{T}$

$\bar{F} = \sum(0, 2, 4, 5, 7)$

$F = \sum(1, 3, 6)$

$\bar{F} = \prod(0, 2, 4, 5, 7)$

9) a) $\bar{F} = RST + R\bar{S}T + R\bar{S}\bar{T} + \bar{R}S\bar{T} + \bar{R}\bar{S}\bar{T}$

$F = \prod(0, 2, 4, 5, 7)$

$F = \sum(1, 3, 6)$

b) $\bar{F} = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + AB\bar{C}$

$F = \prod(0, 2, 3, 6,)$ $\quad F \sum(1, 4, 5, 7)$

c) $\bar{F} = xy\bar{z} + x\bar{y}z + \bar{x}y\bar{z} + xyz$

$F = \prod(2, 5, 6, 7)$ $\quad F = \sum(0, 1, 3, 4)$

3 4 5 6

(10) $F3 = \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C}$ (SOP)

$F3 = \Pi(0,1,2,7)$

$\overline{F3} = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + ABC$

$F3 = (A+B+C)(A+B+\bar{C})(A+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$ (POS)

$F4 = \Sigma(0,1,3,5)$

$= \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C$ (SOP)

$F4 = \Pi(2,4,6,7)$

$\overline{F4} = \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C} + ABC$

$F4 = (A+\bar{B}+C)(\bar{A}+B+C)(\bar{A}+\bar{B}+C)(\bar{A}+\bar{B}+\bar{C})$ POS

$F5 = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}BC\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + AB\bar{C}\bar{D} + AB\bar{C}D$ (SOP)

$F5 = \Pi(1,3,5,7,10,11,14,15)$

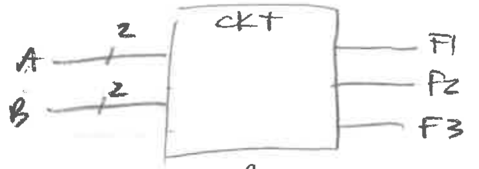
$\overline{F5} = \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BCD + A\bar{B}C\bar{D} + A\bar{B}CD + ABC\bar{D} + ABCD$

$F5 = (A+B+C+\bar{D})(\bar{A}+B+C+\bar{D})(A+\bar{B}+C+\bar{D})(A+\bar{B}+\bar{C}+\bar{D})(\bar{A}+B+\bar{C}+D)(\bar{A}+B+\bar{C}+\bar{D})$

$(\bar{A}+\bar{B}+\bar{C}+D)(\bar{A}+\bar{B}+\bar{C}+\bar{D})$ POS

CHAPTER 9 DESIGN Problems

1)



Bin ↙ ↙ SA Bin

A	B	F1	F2	F3
00	00	0	1	0
00	01	1	0	0
00	10	1	0	0
00	11	1	0	0
01	00	1	0	0
01	01	0	1	0
01	10	1	1	1
01	11	1	0	0
10	00	1	0	0
10	01	1	1	1
10	10	0	1	0
10	11	1	0	0
11	00	1	0	0
11	01	1	0	0
11	10	1	0	0
11	11	0	1	0

N

$F3 = F1 \cdot F2$

$\overline{F1} = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}D + A\bar{B}C\bar{D} + ABCD$

$F1 = (A+B+C+D)(A+\bar{B}+C+\bar{D})(\bar{A}+B+\bar{C}+D)(\bar{A}+\bar{B}+\bar{C}+\bar{D})$

POS IS A SHORTER EQUATION

$F2 = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + A\bar{B}\bar{C}D$

$+ A\bar{B}C\bar{D} + ABCD$

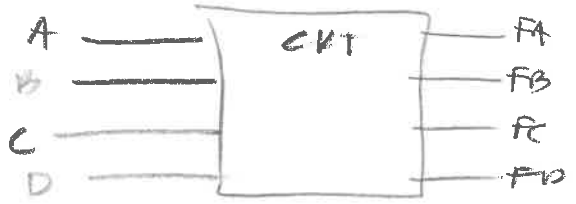
Sop iis SHORTER (SO I wont do POS)

$F3 = \bar{A}BC\bar{D} + A\bar{B}\bar{C}D$ (SOP) (Much SHORTER)

NO control

2)

A	B	C	D	FA	FB	fc	FD
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	1	0	1
0	1	1	0	0	0	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	0	0	0
1	0	0	1	1	0	0	0
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	0
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	0
1	1	1	0	1	0	1	0
1	1	1	1	1	0	1	0



$FA = \Pi(0,1,2,3,4,5,6,7,8)$

$FB = \Pi(0,1,2,3,4,6,8,9,10,11,12,13,14,15)$

$FC = \Pi(0,1,2,3,4,5,6,7,8,9,12,13)$

$FD = \Pi(0,1,2,3,4,6,7,8,9,10,11,12,13,14,15)$

No control

THERE'S NO WAY I'M DRAWING THE CIRCUITS!

CHAPTER 10 EXERCISES

1) THEY ONLY HAVE 2 INPUTS BECAUSE THAT IS HOW THEY ARE DEFINED.

2) THEY ARE NOT FUNCTIONALLY COMPLETE BECAUSE THEY CAN'T BE CONFIGURED TO PERFORM A COMPLIMENT FUNCTION

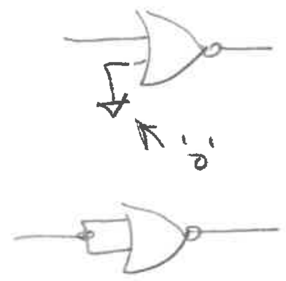
3) NO; THE CAN'T PERFORM AND & OR FUNCTIONS, THOUGH THEY CAN PERFORM COMPLIMENT FUNCTIONS

4) NOR

A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0

INVERTER →

inverter →

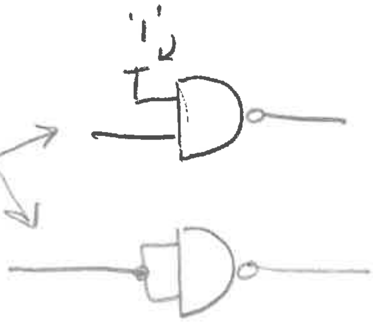


5) NAND

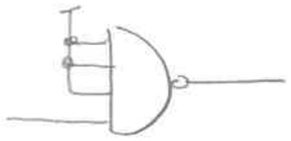
A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

inverter

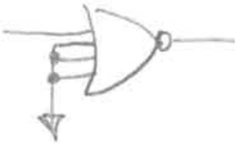
inverter



6)



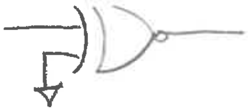
7)



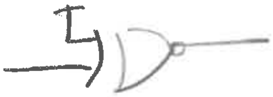
8)

	A	B	$\overline{A\oplus B}$
	0	0	1
x	0	1	0
	1	0	0
	1	1	1

x)



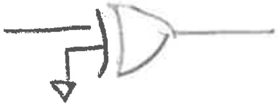
inverter



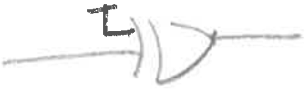
BUFFER

9)

A	B	$A+B$
0	0	0
0	1	1
1	0	1
1	1	0



BUFFER



INVERTER

10) $F1 = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + AB\bar{C} + ABC + \bar{A}B\bar{C}$

$= \bar{A}\bar{B}(C+\bar{C}) + AB(C+\bar{C}) + \bar{A}B\bar{C}$

$= \bar{A}\bar{B} + AB + \bar{A}B\bar{C}$

$= \overline{A\oplus B} + \bar{A}B\bar{C}$

$F2 = ABC + \bar{A}BC + A\bar{B}C$

$= C(\bar{A}B + A\bar{B}) + ABC$

$= C(A\oplus B) + ABC$

$F3 = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C$

$= \bar{A}B(C+\bar{C}) + A\bar{B}(C+\bar{C})$

$= \bar{A}B + A\bar{B}$

$= A\oplus B$

11) XOR & XNOR GATES ARE ONLY DEFINED FOR 2 inputs. AND-TYPE & OR TYPE GATES ARE DEFINED FOR 2 OR more inputs

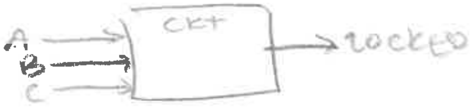
AND GATE: OUTPUT IS HIGH ONLY WHEN ALL INPUTS ARE HIGH. OTHERWISE OUTPUT IS LOW .

OR GATE : OUTPUT IS LOW ONLY WHEN all inputs ARE LOW
OTHERWISE OUTPUT IS HIGH

Chapter 10 Design Problems

1)

A	B	C	LOCKED
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



$$\text{LOCKED} = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}C + ABC$$

$$= \bar{A}\bar{C}(B+\bar{B}) + AC(B+\bar{B})$$

$$= \bar{A}\bar{C} + AC$$

$$= \overline{A \oplus C} \quad (\text{or } A \odot C)$$

No control

2)

A	B	C	WAT
0	0	0	0
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

$$\text{WAT} = \bar{A}BC + A\bar{B}C + AB\bar{C}$$

$$= \bar{A}BC + A(\bar{B}C + B\bar{C})$$

$$= \bar{A}BC + A(B \oplus C)$$

No control

23286078.1 00010937 0-7
EOLK4052 5038 D7 20170118 PG 1 OF 1
018 142 020 1

1) a) $F(A,B,C) = \bar A\bar B\bar C + \bar A\bar B C + A\bar B\bar C + AB\bar C$ AND/OR

$\overline{\overline{\bar A\bar B\bar C + \bar A\bar B C + A\bar B\bar C + AB\bar C}}$

$\overline{(\overline{\bar A\bar B\bar C})(\overline{\bar A\bar B C})(\overline{A\bar B\bar C})(\overline{AB\bar C})}$ NAND/NAND

$\overline{(A+B+C)(A+B+\bar C)(\bar A+B+C)(\bar A+\bar B+C)}$ OR/NAND

$\overline{(A+B+C)}+\overline{(A+B+\bar C)}+\overline{(\bar A+B+C)}+\overline{(\bar A+\bar B+C)} =$ NOR/OR

c $F(ABC)\ \Sigma(0,1,4,6) = \Pi(2,3,5,7)$

$\bar F = \bar AB\bar C + \bar ABC + A\bar BC + ABC$

$F = (A+\bar B+C)(A+\bar B+\bar C)(\bar A+B+\bar C)(\bar A+\bar B+\bar C)$ OR/AND

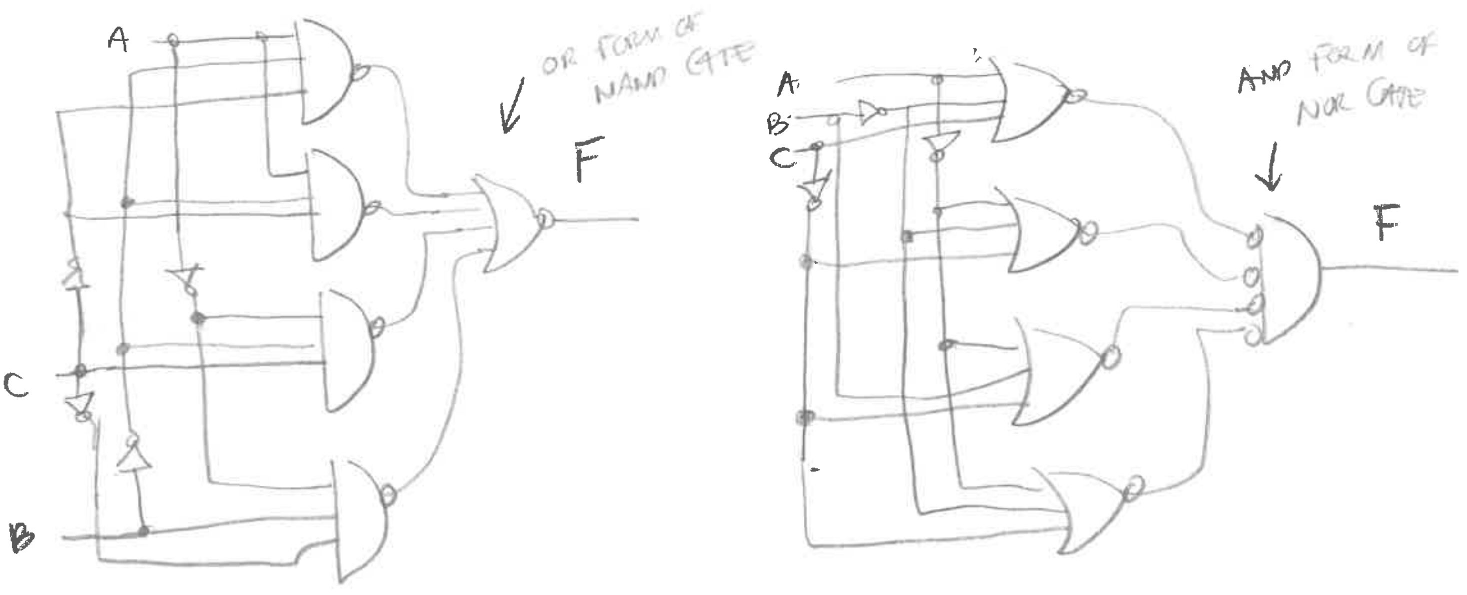
$F = \overline{\overline{(A+\bar B+C)(A+\bar B+\bar C)(\bar A+B+\bar C)(\bar A+\bar B+\bar C)}}$

$F = \overline{\overline{(A+\bar B+C)} + \overline{(A+\bar B+\bar C)} + \overline{(\bar A+B+\bar C)}\ \overline{(\bar A+\bar B+\bar C)}}$ NOR/NOR

$F = \overline{(\bar AB\bar C) + (\bar ABC) + (A\bar BC) + (ABC)}$ AND/NOR

$F = (\overline{\bar AB\bar C})(\overline{\bar ABC})(\overline{A\bar BC})(\overline{ABC}$ NAND/AND

b.)



2) SHOW all FOUR AND/OR Related Form OF THE FOLLOWING EQUATION:

$$F(A,B,C,D) = A\bar{B}C + \bar{B}D + \bar{A}BD$$

$A\bar{B}C + \bar{B}D + \bar{A}BD =$ AND/OR

$\overline{\overline{A\bar{B}C + \bar{B}D + \bar{A}BD}} =$

$\overline{\overline{(A\bar{B}C)}\;\overline{(\bar{B}D)}\;\overline{(\bar{A}BD)}} =$ NAND/NAND

$\overline{(\bar{A}+B+\bar{C})(B+\bar{D})(A+\bar{B}+\bar{D})}$ OR/NAND

$\overline{(\bar{A}+B+\bar{C})} + \overline{(B+\bar{D})} + \overline{(A+\bar{B}+\bar{D})} =$ NOR/OR

3) SHOW all FOUR OR/AND RELATED FORMS FOR THE FOLLOWING EQUATION.

$$F(A,B,C,D) = (B+C+\bar{D})(\bar{A}+\bar{C})(A+\bar{B}+\bar{D})$$

$(B+C+\bar{D})(\bar{A}+\bar{C})(A+\bar{B}+\bar{D})$ OR/AND

$\overline{\overline{(B+C+\bar{D})(\bar{A}+\bar{C})(A+\bar{B}+\bar{D})}}$

$\overline{\overline{(B+C+\bar{D})} + \overline{(\bar{A}+\bar{C})} + \overline{(A+\bar{B}+\bar{D})}}$ NOR/NOR

$\overline{(\bar{B}\bar{C}D) + (AC) + (\bar{A}BD)}$ AND/NOR

$\overline{(\bar{B}\bar{C}D)}\;\overline{(AC)}\;\overline{(\bar{A}BD)}$ NAND/AND

4) SHOW all FOUR AND/OR Related FORMS OF THE FOLLOWING EQ

$$F(A,B,C,D) = \overline{(A+\bar{B})(\bar{A}+\bar{C})(\bar{B}+C+\bar{D})}$$

Problem is Stated in OR/NAND Form

$\overline{(A+\bar{B})} + \overline{(\bar{A}+\bar{C})} + \overline{(\bar{B}+C+\bar{D})}$ NOR/OR

$(\bar{A}B)(AC)(B\bar{C}D)$ AND/OR

$\overline{\overline{(\bar{A}B)}\;\overline{(AC)}\;\overline{B\bar{C}D}}$ NAND/NAND

5) SHOW all FOUR OR/AND TYPE FORMS OF THE FOLLOWING EQUATION

$F(A,B,C,D) = \overline{(\bar{B} D)}\ \overline{(A \bar{D})}\ \overline{(B C \bar{D})}$ $\boxed{\text{NAND / AND}}$

PROBLEM GIVEN IN NAND/AND FORM

|

$(B+\bar{D})(\bar{A}+D)(\bar{B}+\bar{C}+D)$ $\boxed{\text{OR / AND}}$

$\overline{\overline{(B+\bar{D})} + \overline{(\bar{A}+D)} + \overline{\bar{B}+\bar{C}+D}}$ $\boxed{\text{NOR / NOR}}$

$\overline{(\bar{B}D) + (A\bar{D}) + (BC\bar{D})}$ $\boxed{\text{AND / NOR}}$

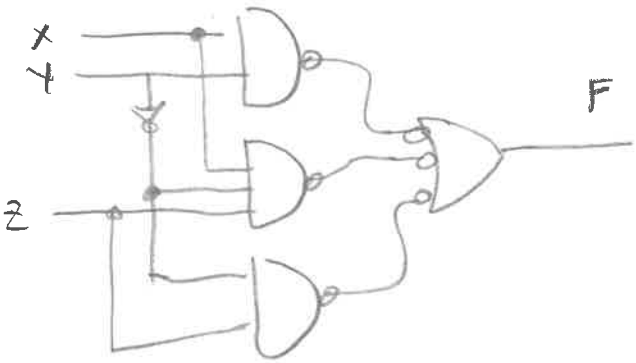
6) DRAW A CIRCUIT FOR THE FOLLOWING EQUATIONS USING only NAND GATES & INVERTERS

a) $F(X,Y,Z) = XY + X\bar{Y}Z + \bar{Y}Z$

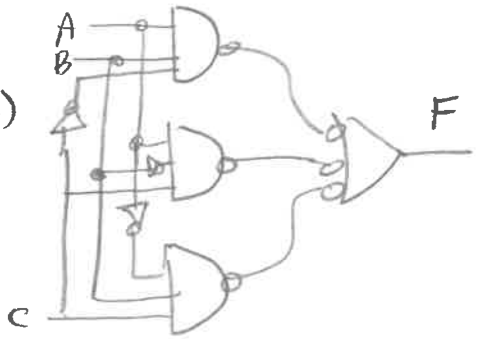
b) $F(A,B,C) = B\bar{C} + A\bar{B}C + \bar{A}C$

c) $F(A,B,C) = AB\bar{C} + A\bar{B}C + \bar{A}BC$

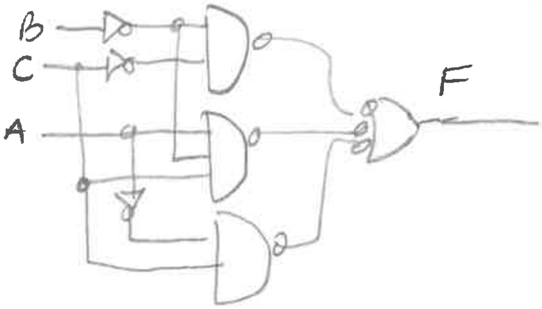
a)



c)



b)



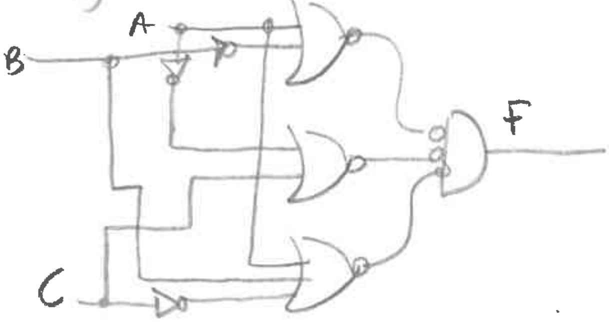
7) DRAW A CIRCUIT FOR FOLLOWING EQUATIONS USING only NOR GATES & INVERTERS

a) $F(A,B,C) = (A+\overline{B})(\overline{A}+C)(A+B+\overline{C})$

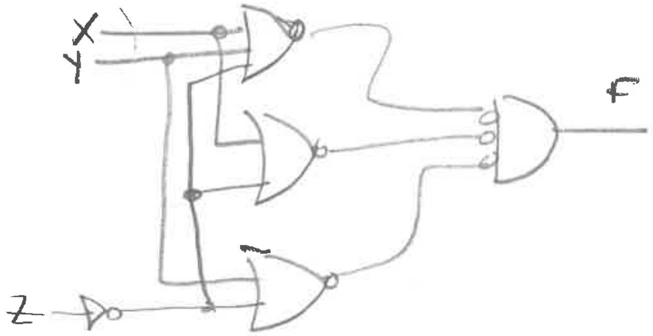
b) $F(X,Y,Z) = (X+Y+\overline{Z})(X+\overline{Z})(Y+\overline{Z})$

c) $F(A,B,C) = (\overline{A})(A+B)(A+\overline{B}+\overline{C})$

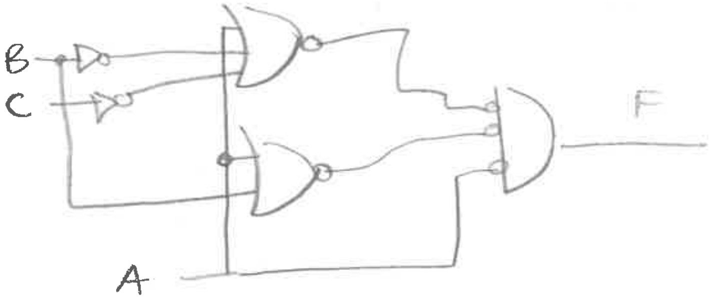
a)



b)

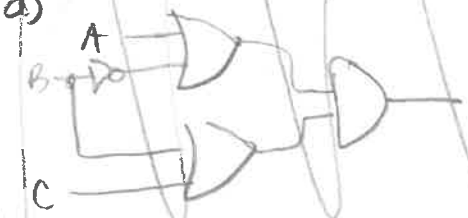


c)

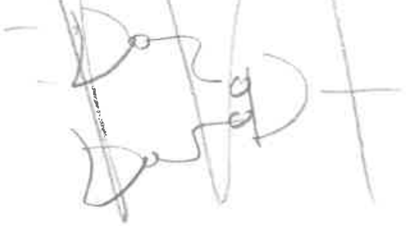


8) SHOW THE FOUR COMMON BOOLEAN EQUATION FORMS OF THE FOLLOWING CIRCUITS

a)



b) FIND SAME



9) Done!

8) a) $F(A,B,C,D)\quad \bar A \bar C \bar D + BD + A\bar B C$ (AND/OR)

$$\overline{\overline{\bar A \bar C D + BD + A \bar B C}}$$

$$\overline{\overline{(\bar A \bar C \bar D)}\;\overline{(BD)}\;\overline{(A\bar B C)}}$$ NAND/NAND

$$\overline{(A+C+D)(\bar B+\bar D)(\bar A+B+\bar C)}$$ OR/NAND

$$\overline{(A+C+D)} + \overline{(\bar B+\bar D)} + \overline{(\bar A+B+\bar C)}$$ NOR/OR

b) $F(A,B,C) = \bar A B + AC$ (AND/OR)

$$= \overline{\overline{\bar A B + AC}}$$

$$= \overline{\overline{(\bar A B)}\;\overline{(AC)}}$$ NAND/NAND

$$= \overline{(A+\bar B)(\bar A+\bar C)}$$ OR/NAND

c) $F(A,B,C,D) = AB + CD$ AND/OR

$$= \overline{\overline{AB + CD}}$$

$$= \overline{\overline{(AB)}\;\overline{(CD)}}$$ NAND/NAND

$$= \overline{(\bar A+\bar B)(\bar C+\bar D)}$$ OR/NAND

$$= \overline{(\bar A+\bar B)} + \overline{(\bar C+\bar D)}$$ NOR/OR

6

9) a) $F(A,B,C) = (A+\bar{B})(\bar{A}+\bar{C})$ OR/AND

$$\overline{\overline{(A+\bar{B})(\bar{A}+\bar{C})}}$$

$$\overline{\overline{(A+\bar{B})} + \overline{(\bar{A}+\bar{C})}}$$ NOR/NOR

$$\overline{(\bar{A}B) + (AC)}$$ AND/NOR

$$\overline{(\bar{A}B)}\,\overline{(AC)}$$ NAND/AND

b) $F = (A+\bar{C})(A+B+\bar{C})(\bar{B}+\bar{C})$ OR/AND

$$\overline{\overline{(A+\bar{C})(A+B+\bar{C})(\bar{B}+\bar{C})}}$$

$$\overline{\overline{(A+\bar{C})} + \overline{(A+B+\bar{C})}\;\overline{(\bar{B}+\bar{C})}}$$ NOR/NOR

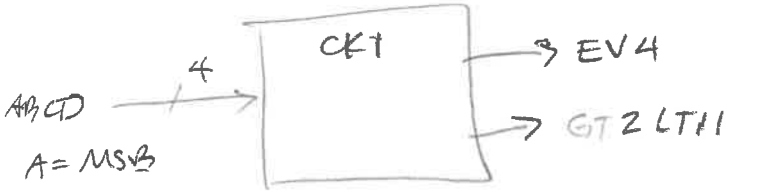
$$\overline{(\bar{A}C) + (\bar{A}\bar{B}C) + (BC)}$$ AND/NOR

$$\overline{(\bar{A}C)}\;\overline{(\bar{A}\bar{B}C)}\;\overline{(BC)}$$ NAND/AND

CHAPTER 11 Design Problems

1)

A	B	C	D	EV4	GT2LT11
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	1
0	1	0	0	1	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0



NO_control

a) $$EV4 = \bar{A}B\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + AB\bar{C}\bar{D}$$

$$= \overline{\overline{(\bar{A}B\bar{C}\bar{D})}\;\overline{(A\bar{B}\bar{C}\bar{D})}\;\overline{(AB\bar{C}\bar{D})}}$$ NAND GATES

$$GT2LT11 = \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D}$$

$$= \overline{\overline{(\bar{A}\bar{B}CD)}\;\overline{(\bar{A}B\bar{C}\bar{D})}\;\overline{(\bar{A}B\bar{C}D)}\;\overline{(\bar{A}BC\bar{D})}\;\overline{(\bar{A}BCD)}\;\overline{(A\bar{B}\bar{C}\bar{D})}\;\overline{(A\bar{B}\bar{C}D)}\;\overline{(A\bar{B}C\bar{D})}}$$

NAND/NAND

b) $$\overline{EV4} = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BC\bar{D} + \bar{A}BCD + A\bar{B}\bar{C}D + A\bar{B}C\bar{D}$$
$$+ A\bar{B}CD \quad + AB\bar{C}D \quad + ABC\bar{D} \quad + ABCD$$

$$EV4 = \overline{(A+B+C+D)}\,\overline{(A+B+C+\bar{D})}\,(A+B+\bar{C}+D)(A+B+\bar{C}+\bar{D})(\bar{A}+B+\bar{C}+D)$$
$$(A+\bar{B}+C+\bar{D})(A+B+\bar{C}+D)\,\overline{(A+\bar{B}+\bar{C}+\bar{D})}(\bar{A}+B+C+\bar{D})(\bar{A}+B+\bar{C}+\bar{D})$$
$$(\bar{A}+\bar{B}+C+\bar{D})(\bar{A}+\bar{B}+\bar{C}+D)(\bar{A}+\bar{B}+C+D)$$ NOR/NOR

$$GT2LT11 = (\bar{A}\bar{B}\bar{C}\bar{D})(\bar{A}\bar{B}\bar{C}D)(\bar{A}\bar{B}C\bar{D})(AB\bar{C}D)(AB\bar{C}\bar{D})(AB\bar{C}D)(ABC\bar{D})(ABCD)$$

$$= (A+B+C+D)(A+B+C+\bar{D})(A+B+\bar{C}+D)(\bar{A}+\bar{B}+C+\bar{D})(\bar{A}+\bar{B}+C+D)$$
$$(\bar{A}+\bar{B}+C+\bar{D})(\bar{A}+\bar{B}+\bar{C}+D)(\bar{A}+\bar{B}+\bar{C}+\bar{D})$$ NOR/NOR

CHAPTER 12 Exercises

1)

# Bits	UNSIGNED BINARY RANGE	SIGNED BINARY RANGE (RC)
4	0 - 15	[-8, 7]
6	0 - 63	[-32, 31]
8	0 - 255	[-128, 127]
10	0 - 1023	[-512, 511]
11	0 - 2047	[-1024, 1023]
12	0 - 4095	[-2048, 2047)
14	0 - 16383	[-8192, 8191]
15	0 - 32767	[-16384, 16383]
16	0 - 65535	[32768, 32767]

2) a) $1110\ 1110_2$ or $0000\ 0010_2$

$0001\ 0010_2$

b) $1000\ 1101_2$ or $0111\ 0111_2$

0111 0011

c) $1110\ 1110_2$ or $0001\ 0011_2$

0001 0010

3) a) $1110\ 0001_2$ SM ⇒ 01100001_2 = MAG

$1001\ 1101_{ONC}$ ⇒ $0110\ 0010_2$ = MAG

$1001\ 1100_{RC}$ ⇒ 01100100_2 = MAG

Largest magnitude

b) 10011110_{SM} ⇒ 01100001_2

$1000\ 1101_{ONC}$ ⇒ $0111\ 0010_2$

Largest ↗

$1001\ 1111_{RC}$ 01100001_2

4) a) $B4_{16}$ = $1011\ 0100_{SM}$ ⇒ $0011\ 0100_2$ (MAG)

CC_{16} = $1100\ 1100_{DRC}$ ⇒ $0011\ 0011_2$ (MAG)

$D1_{16}$ = $1101\ 0001_{RC}$ ⇒ $0010\ 1111_2$ (MAG

b) $F3_{16}$ = $1111\ 0011_{SM}$ ⇒ $0111\ 0011_2$ (MAG)

EC_{16} = $1110\ 1100_{DRC}$ ⇒ $0001\ 0011_2$ (MAG)

DD_{16} = $1101\ 1101_{RC}$ ⇒ $0010\ 0011_2$ (MAG)

5.) a) $BC_{16} = 1011\ 1100_2$ ⇒ $0011\ 1100_2$ SM

$= 3\times16^1 + 12 = \underline{\underline{-60}}$

$1011\ 1100_2$ ⇒ $0100\ 0011_2$ DRC

$4\times16^1 + 3 = \underline{\underline{-67}}$

$1011\ 1100_2$ ⇒ $0100\ 0100_2$ RC

$4\times16^1 + 4 = \underline{\underline{-68}}$

b) $4A_{16} = 0100\ 1010_2 = 4\times16 + 10 = \underline{\underline{74}}$ (NOT too EXCITING)

$0100\ 1010_2 = 4\times16 + 10 = \underline{\underline{74}}$

$0100\ 1010_2 = 4\times16 + 10 = \underline{\underline{74}}$

c) $D2_{16} = 1101\ 0010_2 = 0101\ 0010_2$

$= 5\times16 + 2 = \underline{\underline{-82}}$

$1101\ 0010_2 = 0010\ 1101_2 =$

$= 2\times16 + 13 = \underline{\underline{-45}}$

$1101\ 0010_2$

$0010\ 1110_2$ ⇒ $2\times16 + 14 = \underline{\underline{-46}}$

6) a) BC_{16} = 1011 1100

SM = − 60 $-(3 \times 16 + 12)$

DRC = 0100|0011 = − 67

RC = 01 00|0100 = − 68

b) $4A_{16}$ = 0100 1010.

SM = 74 $(4 \times 16 + 10)$

DRC = 74

RC = 74

c) $D2_{16}$ = 1101 0010

SM = − 82

DRC = 0010 1101 = − 45

RC 0010|11 10 = − 46

7) a) $A7_{16}$ = $0A7_{16}$

b) $4A_{16}$ = $04A_{16}$

c) $C4_{16}$ = $0C4_{16}$

} Zero Extension

8) a) AF_{16} ⇒ $0AF_{16}$

b) $4A_{16}$ ⇒ $04A_{16}$

c) $C4_{16}$ ⇒ $0C4_{16}$

9) a) $A7_{16}$ ⇒ $FA7_{16}$ SIGN Extension

b) $4A_{16}$ ⇒ $04A_{16}$ ZERO Extend

c) $C4_{16}$ ⇒ $FC4_{16}$ SIGN Extend

d) 02_{16} ⇒ 002_{16} zero Extend

10) a) DE_{16} => FDE_{16} Sign Ext

b) $3F_{16}$ ⇒ $03F_{16}$ zero Extend

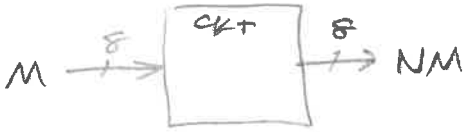
c) $C4_{16}$ ⇒ $FC4_{16}$ Sign Ext

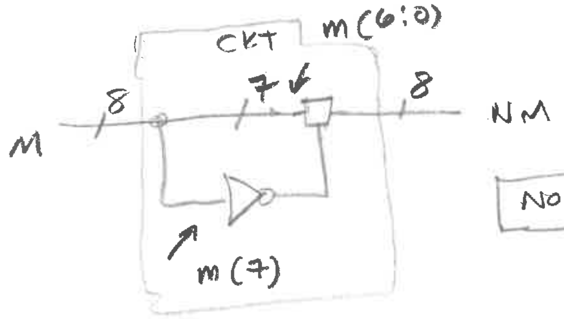
d) 99_{16} = $F99_{16}$ Sign EXT

~~11)~~ .

CHAPTER 12 DESIGN Problems

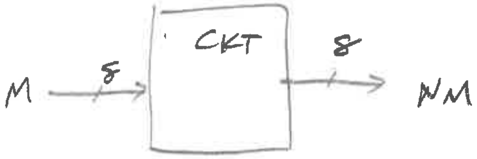
1)



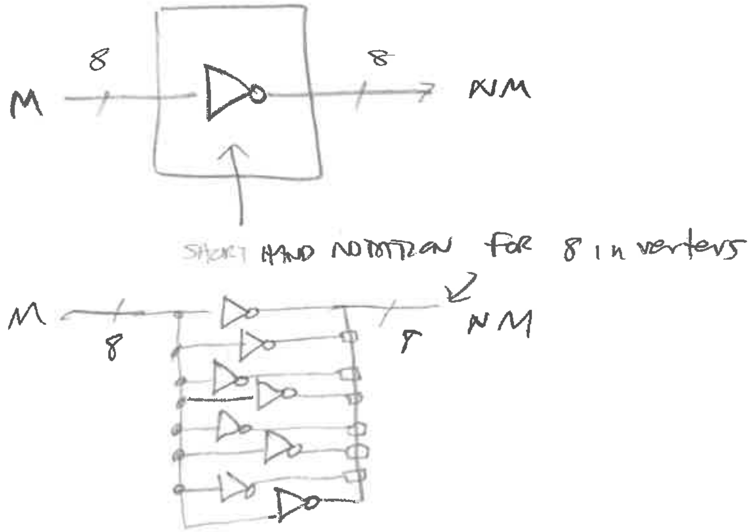


No control

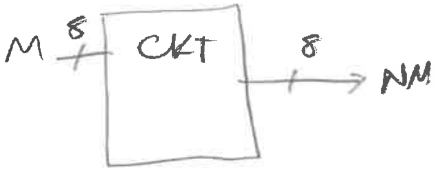
2)

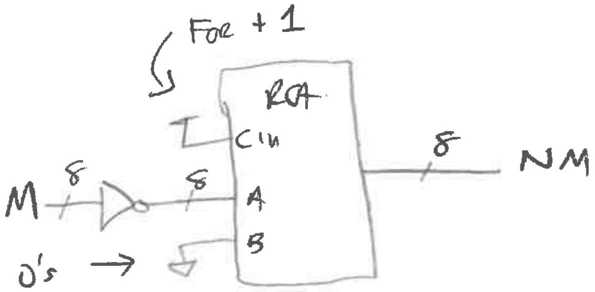


No control



3)

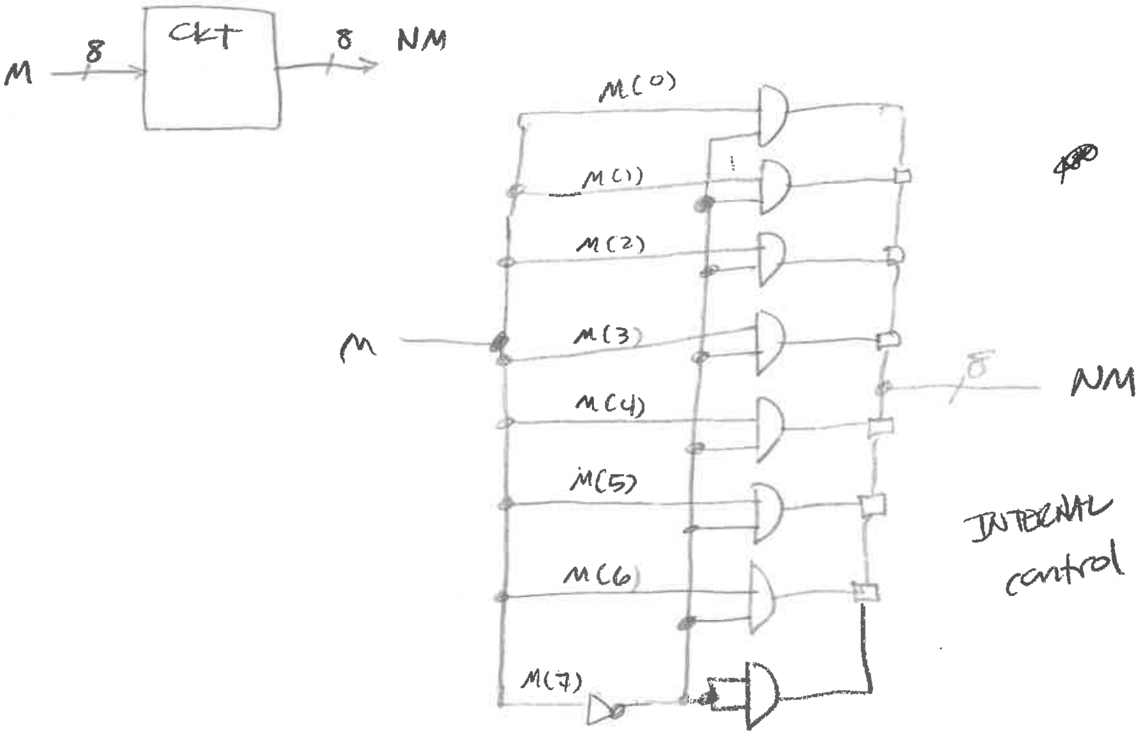




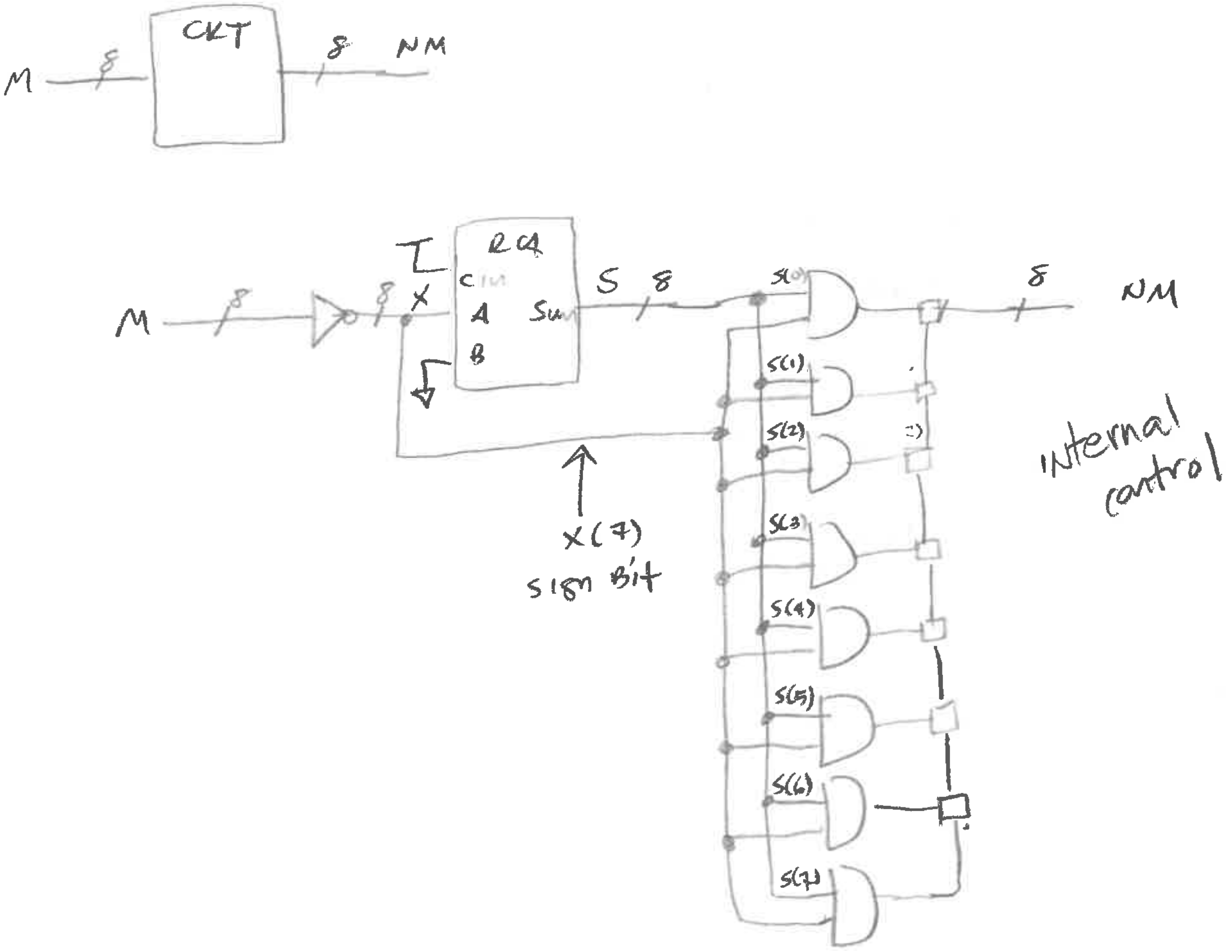
No control

2's comp = 1's comp + 1

4)


CKT
M
8
8
NM
M(0)
M(1)
M(2)
M(3)
M(4)
M(5)
M(6)
M(7)
M
8
NM
INTERNAL control

5)


CKT
M
8
8
NM
M
8
8
X
1
RCA
Cin
A
B
Sum
S
8
8
NM
S(0)
S(1)
S(2)
S(3)
S(4)
S(5)
S(6)
S(7)
X(7)
sign Bit
internal control

CHAPTER 13 | EXERCISE SOLUTIONS

1) THE ONLY WAY A NUMBER CAN INCREASE IN MAGNITUDE IS IF TWO NUMBERS OF THE SAME SIGN ARE ADDED. SO IF TWO NUMBERS FROM A FIXED RANGE ARE ADDED AND THEIR SIGNS ARE NOT THE SAME, THE MAGNITUDE OF THE NUMBER will BECOME Smaller AND WILL THUS THE RESULT will ALWAYS FIT into THE RANGE associated WITH THE TWO NUMBERS BEING ADDED

2) IN TERMS OF THIS CHAPTER, OVERFLOW IS A MANUFACTURED TERM Associated WITH THE SIGN BITS OF BOTH THE TWO OPERANDS AND THE THE RESULT. OVERFLOW IS used to DICERN THE Validity OF THE RESULT. CARRY-OUT IS A SINGLE BIT ASSOCIATED WITH THE MSB'S OF THE TWO OPERANDS. THE CARRY-OUT ALSO CAN SERVE AS A "BORROW" FOR Subtraction operations. IN OTHER WORDS, CARRY OUT AND OVERFLOW ARE Different.

3) BOTH UNDERFLOW AND OVERFLOW Represent NUMBERS THAT HAVE Exceeded THE BOUNDARIES OF A GIVEN RANGE. UNDERFLOW INDICATES THAT YOU HAVE Exceeded THE LOWER BOUNDS OF THE RANGE, OR OVERFLOWN THE LOWER BOUNDARY.

#4

FIXED WIDTH IN HARDWARE MEANS THAT THE HARDWARE DOES MATH ASSOCIATED WITH A GIVEN NUMBER OF BIT (THE WIDTH), AND THAT VALUE DOES NOT (AND CAN NOT) CHANGE. SO IF YOU ADD 1+1 OR 1000 + 1000, THE operation is DONE USING THE SAME NUMBER OF BITS

#5

```
a    001100          b)   0111              c)   100101
   + 000011               001110               + 101010
     001111             + 000111              (1)001111
  no CARRY : valid        010101              CARRY : invalid
                        NO CARRY: valid

        1 1                    11
d)    001000          e)     000100
    + 111100               + 101111
   (1)000100               (0)110011
   CARRY: invalid          NO CARRY: valid
```

#6

```
      1 1 1
a)    001100 - 000111            b)  100101  -  001000
    + 111001       ↓   2's comp          111000        ↓    2's
   (1)000101     111001               (1)011101     111000   comp
   CARRY: valid                       CARRY: valid

c)  111010 - 111100  2's comp     d)       1
   +000100   000100                   010001  -  011011
  (0)111110                          + 100101       ↓
  NO CARRY: NOT VALID                (0)110110    100101
                                     NO CARRY: NOT Valid

      1   1
e)    010010 - 000110
      111010      ↓
   (1)001100    111010
   CARRY: valid
```

#7

```
a)     01001010
      +00010000
      ---------
     0 01011010
     NO CARRY: valid

        1 111
b)     11110000
      +00010001
      ---------
      100000001
      CARRY: INVALID

        11   1
c)     11100100
      +00100101
      ---------
      100001001
      CARRY: INVALID

         1
d)     01000000
      +01110000
      ---------
      010110000
      NO CARRY: valid

        1111
e)     01001000
      +01111111
      ---------
      011000111
      NO CARRY: valid
```

#8

```
a) 01000001 - 00111100
   11000100      ↓        2's
   --------   11000100    Comp
  100000101
   CARRY: Valid

b) 11000000 - 01001110
   10110010      ↓
   --------
  101110010   10110010
   CARRY: Valid

      11
c)  00100101 - 10001110
    01110010       ↓
    --------   01110010
   010010111
    NO CARRY: INVALID

d) 10000001 - 11000010
   00111110       ↓
   --------
  010111111   00111110
   NO CARRY: INVALID

e) 11010011 - 11111100
   00000100   00000100
   --------
  011010111
   NO CARRY: INVALID
```

#9

```
a)   0111
     00011
   + 00111
   -------
     01010

(+ NUM) + (+ NUM) =
  (+ RESULT) ⇒ valid

     111
b)   01110
   + 00011
   -------
     10001

(+ NUM) + (+ NUM)
 = (- RESULT) ⇒ NOT
                 VALID

c)   01001
   + 00100
   -------
     01101

     VALID

     11 1
d)   01010
     00111
 !   -----
     10001
   Not valid

     1 11
e)   01011
     01001
     -----
     10100
   Not valid

f) 00011 - 00111
   USE INDIRECT Subtraction
   By ADDITION

        00111 x -1 =

    11  11001
   00011        (+ NUM) + (-NUM)
   11001         = valid
   -----           result!
   11100
```

#9

g) $$\begin{array}{rll} & 01110 - 00011 & \\ + & 11101 \quad \downarrow & \swarrow \text{2's comp} \\ \hline & 1\underbrace{01011}\quad 11101 & \end{array}$$

Valid: (+ NUM) + (− NUM)

h) $$\begin{array}{rl} & 01001 - 00100 \\ + & 11100 \quad \downarrow \\ \hline & 1\underbrace{00101} \quad 11100 \end{array}$$

Valid!

i) $$\begin{array}{rl} & \overset{1}{0}\overset{1}{0}110 - 10100 \\ + & 01100 \quad \downarrow \\ \hline & \underbrace{10010} \quad 01100 \end{array}$$

(+ NUM) + (+ NUM) = − Result

NOT Valid

J) $$\begin{array}{rl} & 00111 - 11100 \\ + & 00100 \quad \downarrow \\ \hline & 01011 \quad 00100 \end{array}$$

Valid

k) $$\begin{array}{rl} & \overset{1}{0}1010 - 11000 \\ + & 01000 \quad \downarrow \; 01000 \\ \hline & \underbrace{10010} \end{array}$$

NOT Valid

l) $$\begin{array}{rl} & 01\overset{1}{0}10 - 11110 \\ + & 00010 \quad \downarrow \\ \hline & \underbrace{01100} \quad 00010 \end{array}$$

VALID

m) $$\begin{array}{rl} & 0\overset{1}{1}\overset{1}{1}\overset{1}{1}0 - 11001 \\ & 00111 \quad \downarrow \\ \hline & 10101 \quad 00111 \end{array}$$

NOT Valid

#10

a) $$\begin{array}{rl} & 10111 \\ + & 01000 \\ \hline & 11111 \end{array}$$

Valid

b) $$\begin{array}{rl} & \overset{1}{1}\overset{1}{1}\overset{1}{0}\overset{1}{0}1 \\ + & 01111 \\ \hline & 1\,\underbrace{01000} \end{array}$$

valid

c) $$\begin{array}{rl} & 1\overset{1}{1}\overset{1}{1}01 \\ & 00100 \\ \hline & 1\,\underbrace{00001} \end{array}$$

Valid

d) $$\begin{array}{rll} & \overset{1}{1}\overset{1}{1}010 - 01010 & \\ & 10110 \quad \downarrow & \text{2's comp} \\ \hline & 1\,\underbrace{10000} \quad 10110 & \end{array}$$

Valid

e) $$\begin{array}{rl} & \overset{1}{1}\overset{1}{1}\overset{1}{1}01 - 00100 \\ & 11100 \quad \downarrow \\ \hline & \underbrace{11001} \quad 11100 \end{array}$$

Valid

$\boxed{\#10}$

f) $$\begin{array}{r} {}^{0\;1}\quad\; \\ 11010 \\ +10010 \\ \hline 1\;\underbrace{01100}_{\text{invalid}} \end{array} \qquad \begin{array}{c} -\,01110 \\ \downarrow \\ 10010 \end{array}$$

g) $$\begin{array}{r} 10100 \\ +10010 \\ \hline 1\;\underbrace{00110}_{\text{Not valid}} \end{array} \qquad \begin{array}{c} -\,01110 \\ \downarrow \\ 10010 \end{array}$$

h) $$\begin{array}{r} {}^{1\,1\,1\,1\,1} \\ 11111 \\ +10111 \\ \hline 1\;\underbrace{10110}_{\text{valid}} \end{array} \qquad \begin{array}{c} -\,01001 \\ \downarrow \\ 10111 \end{array}$$

i) $$\begin{array}{r} {}^{1\,1\,1\,1} \\ 10111 \\ +01001 \\ \hline 1\;\underbrace{00000}_{\text{valid}} \end{array} \qquad \begin{array}{c} -\,10111 \\ \downarrow \\ 01001 \end{array}$$

j) $$\begin{array}{r} {}^{1\;\;11}\quad \\ 11101 \\ 00110 \\ \hline 1\;\underbrace{00011}_{\text{valid}} \end{array} \qquad \begin{array}{c} -\,11010 \\ \downarrow \\ 00110 \end{array}$$

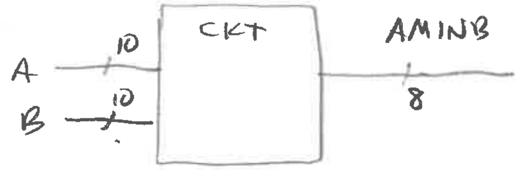
k) $$\begin{array}{r} 11000 \\ 00010 \\ \hline \underbrace{11010}_{\text{valid}} \end{array} \qquad \begin{array}{c} -\,11110 \\ \downarrow \\ 00010 \end{array}$$

11) 1) complement all bits; ADD 1

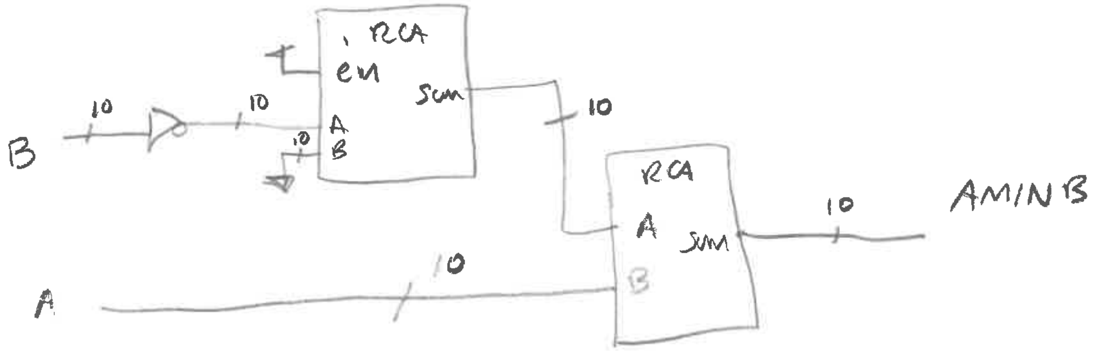
SCAN BITS FROM RIGHT TO LEFT; DON'T CHANGE ANY BITS UNTIL YOU ENCOUNTER THE FIRST 1; DON'T CHANGE THE 1 BUT TOGGLE ALL THE OTHER BITS ENCOUNTERED AS YOU SCAN RIGHT TO LEFT.

CHAPTER 13 DESIGN Problems

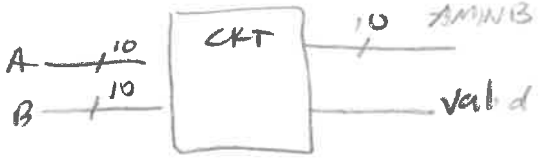
1) DESIGN A CIRCUIT THAT ALWAYS DOES THE FOLLOWING operation: A−B. CONSIDER BOTH A & B TO Be 10-bit BINARY NUMBERS IN RC FORMAT. ASSUME RESULT WILL ALWAYS Be valid.

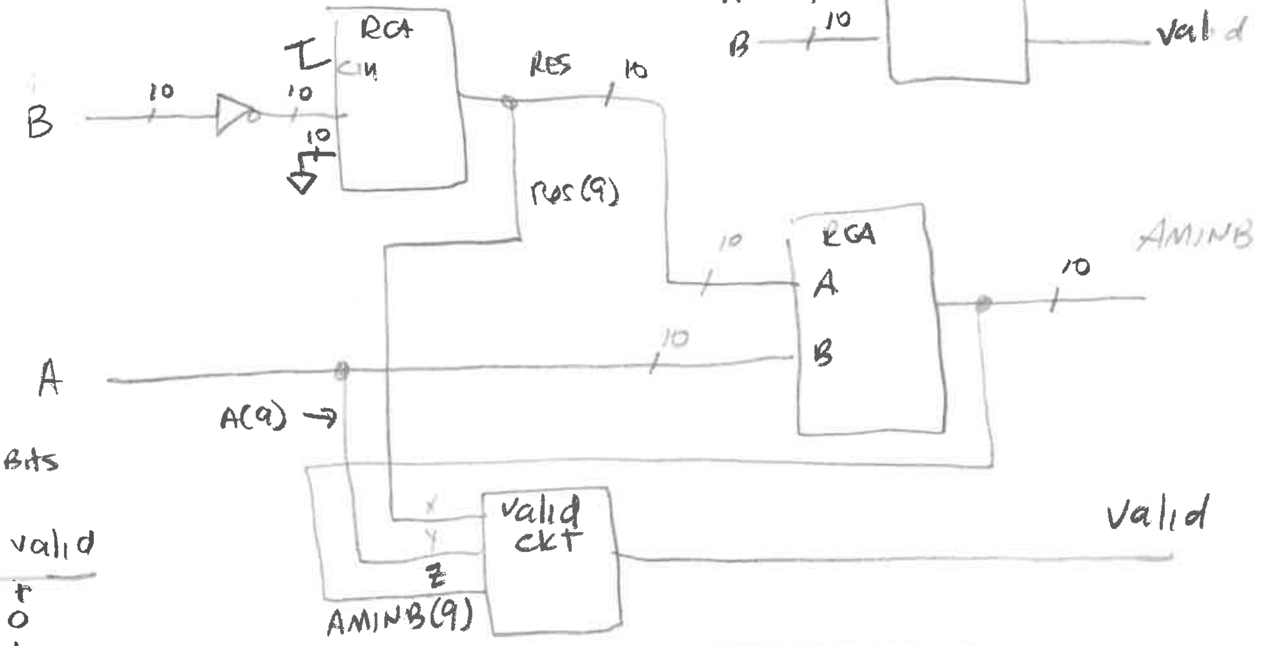


NO CONTROL



2) Repeat previous Problem but include A output "valid" that is A '1' WHEN THE RESULT OF THE SUBTRACTION operation IS valid





No control

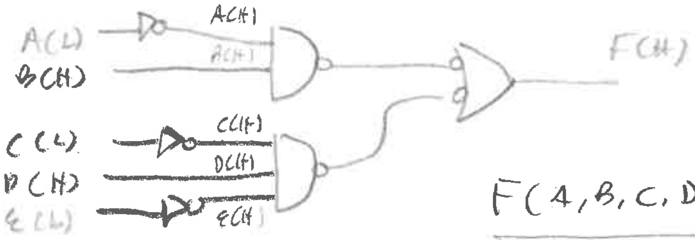
SIGN Bits

X	Y	Z	valid
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

← TRUTH TABLE FOR valid ckt

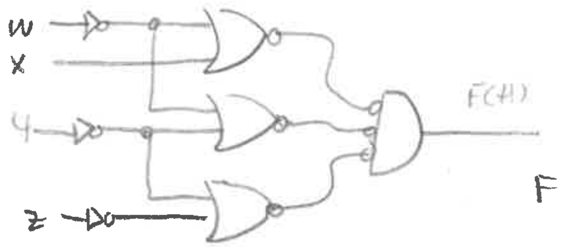
CHAPTER 14 EXERCISES

1) FIRST, REDRAW CIRCUIT:



$F(A,B,C,D,E)(H) = AB + CDE$

2) REDRAW



CKT IS IN NOR/NOR FORM

$F(W,X,Y,Z) = (\bar{W}+X)(\bar{W}+\bar{Y})(\bar{Y}+\bar{Z})$

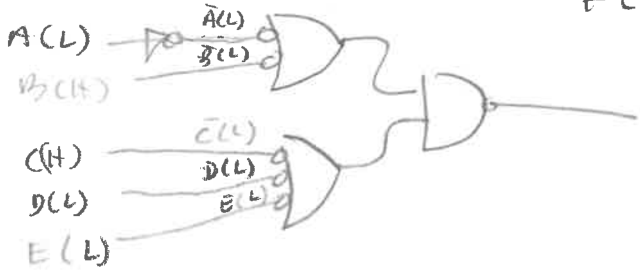
$\bar{F} = W\bar{X} + WY + YZ$ ← where 0's are in truth table

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

$$F = \bar{W}\bar{X}\bar{Y}\bar{Z} + \bar{W}\bar{X}\bar{Y}Z + \bar{W}\bar{X}Y\bar{Z} + \bar{W}X\bar{Y}\bar{Z} + \bar{W}X\bar{Y}Z + \bar{W}XY\bar{Z} + WX\bar{Y}\bar{Z} + WX\bar{Y}Z \qquad \boxed{\text{AND/OR}}$$

$$F = \overline{\overline{(\bar{W}\bar{X}\bar{Y}\bar{Z})}\,\overline{(\bar{W}\bar{X}\bar{Y}Z)}\cdot\overline{(\bar{W}\bar{X}Y\bar{Z})}\;\overline{(\bar{W}X\bar{Y}\bar{Z})}\cdot\overline{(\bar{W}X\bar{Y}Z)}\,\overline{(\bar{W}XY\bar{Z})}\cdot\overline{(WX\bar{Y}\bar{Z})}\cdot\overline{(WX\bar{Y}Z)}}$$

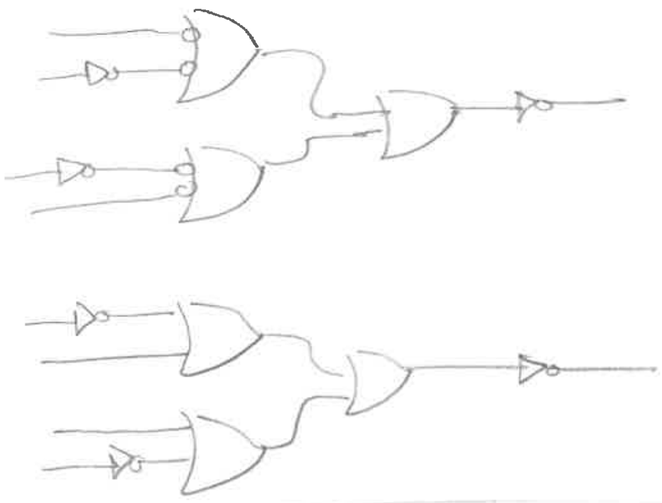
3) REDRAW



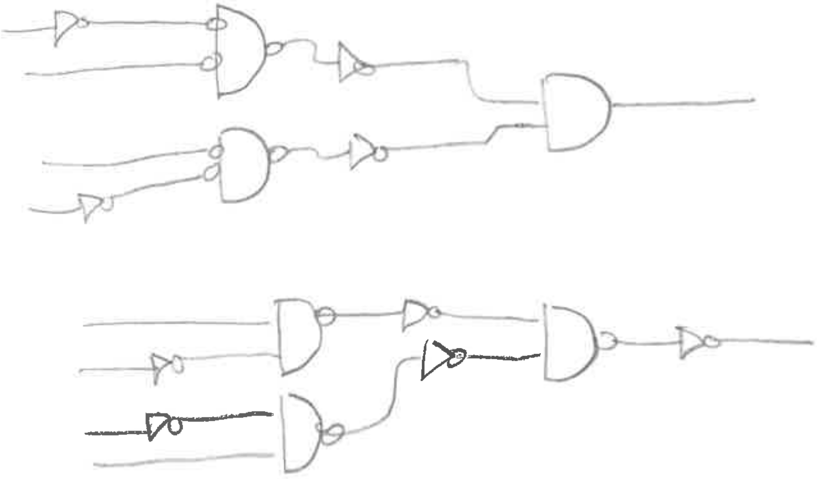
$F(A,B,C,D,E)(L) =$

$(\bar{A} + \bar{B})(\bar{C} + D + E)(L)$

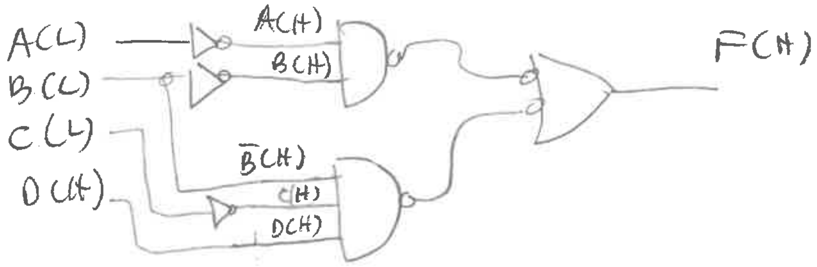
4) REDRAW



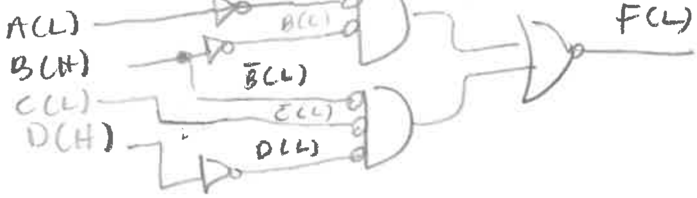
5) REDRAW



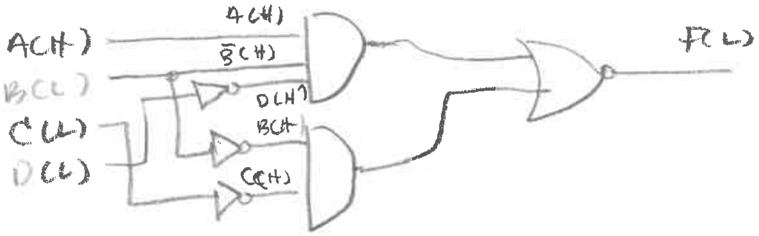
6) $F(H) = (AB + \bar{B}CD)(H)$



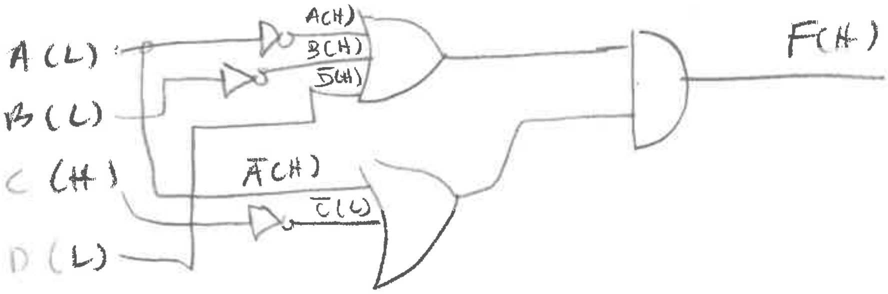
7) $F(L) = (\bar{A}B + \bar{B}CD)(L)$



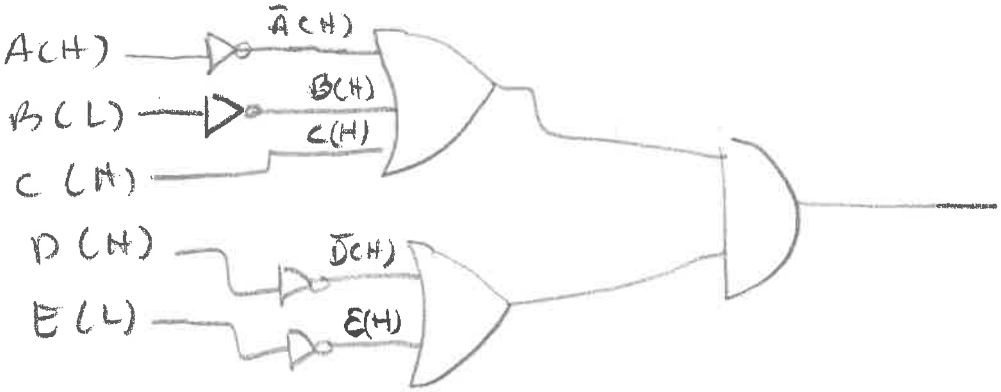
8) $F(A, B, C, D)(L) = (A\bar{B}D + BC)(L)$



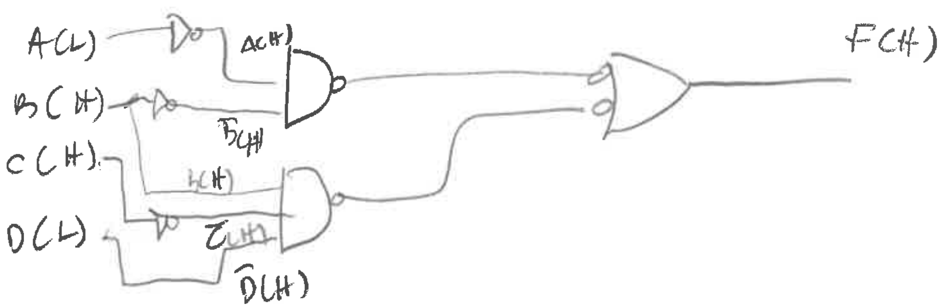
9) $F(A, B, C, D)(H) = [(A+B+\bar{D})(\bar{A}+\bar{C})](H)$



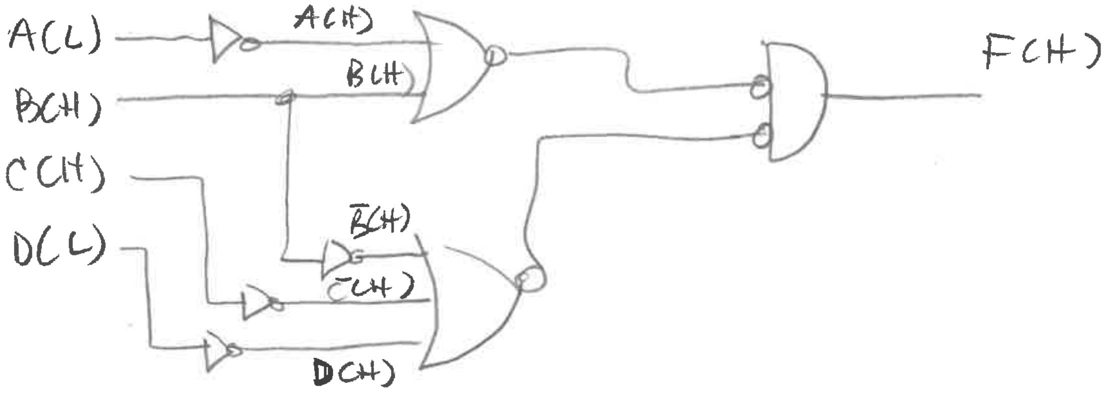
10) $F(A, B, C, D, E)(L) = [(\bar{A}+B+C)(\bar{D}+E)](L)$



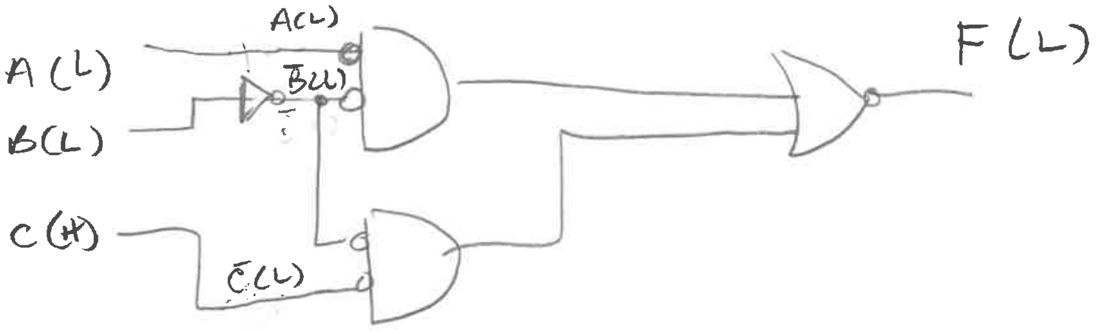
11) $F(H) = (A \cdot \overline{B} + B\overline{C}\overline{D})(H)$



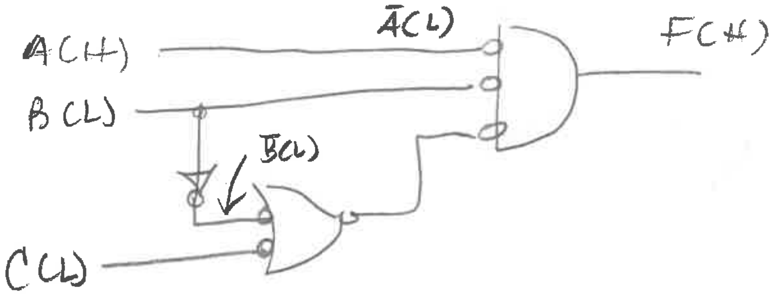
12) $F(H) = (A+B)\cdot(\overline{B}+\overline{C}+D)(H)$



13) $F(L) = [(A \cdot \overline{B}) + (\overline{B} \cdot \overline{C})](L)$

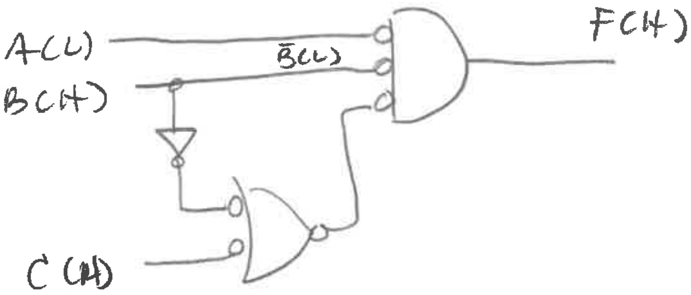


14) a)



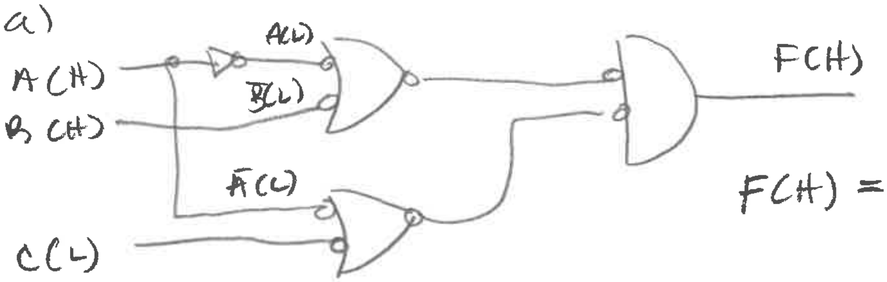
$$F(H) = \bar{A} \cdot B \cdot (\bar{B} + C)$$

b) Same circuit

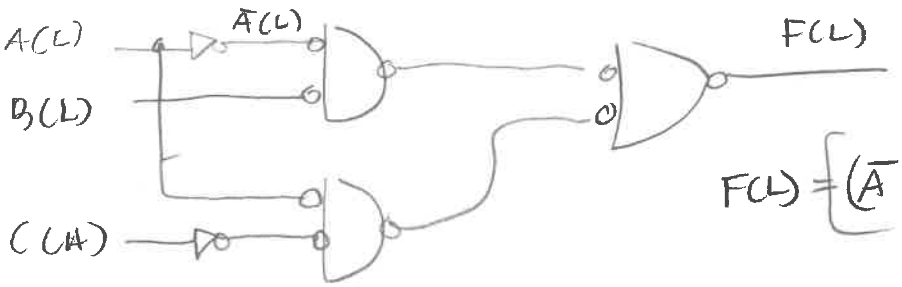


$$F(H) = A \cdot \bar{B} \cdot (B + \bar{C})$$

15) a)

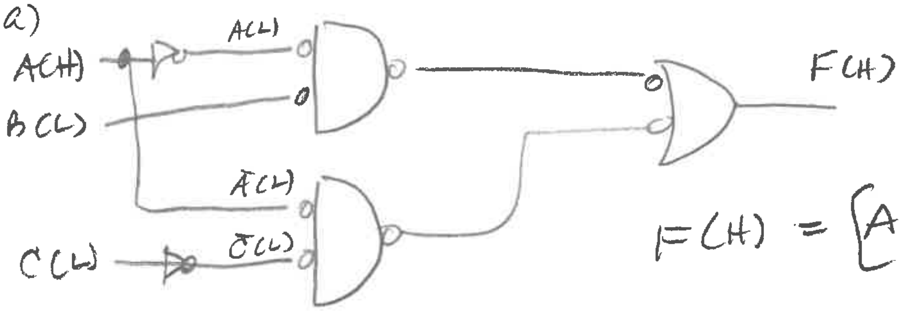


$$F(H) = [(A+B)(\bar{A}+C)](H)$$



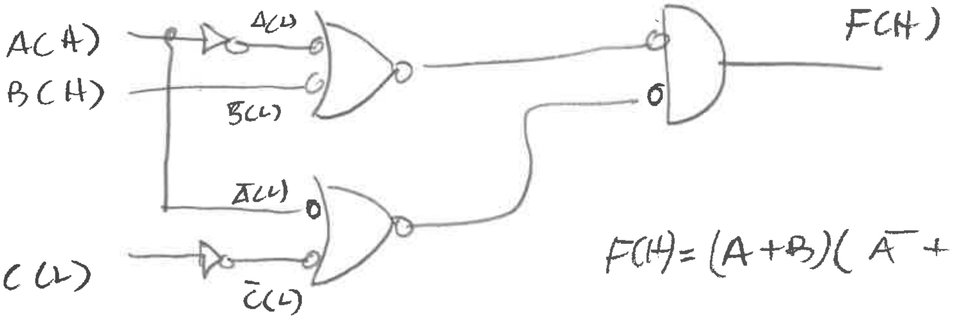
$$F(L) = [(\bar{A}B) + (AC)](L)$$

16) a)



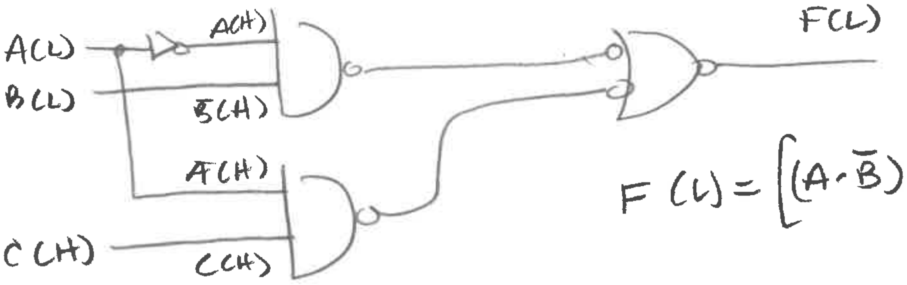
$F(H) = [AB + \bar{A}\bar{C}](H)$

b)



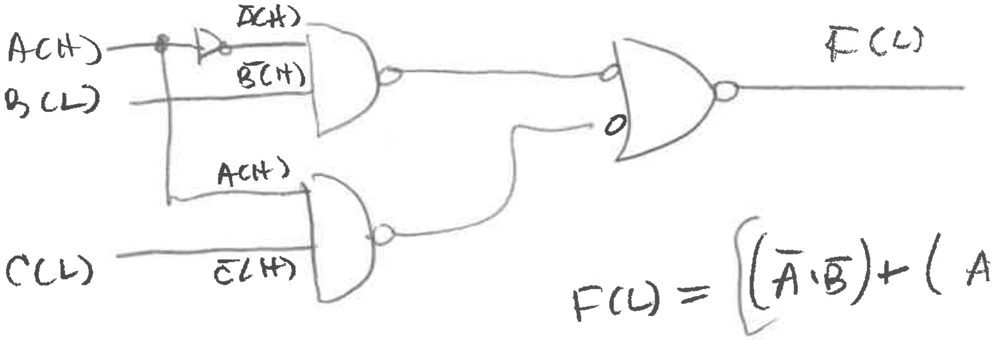
$F(H) = (A+B)(\bar{A}+\bar{C})](H)$

17) a)



$F(L) = [(A\cdot\bar{B}) + (\bar{A}\cdot C)](L)$

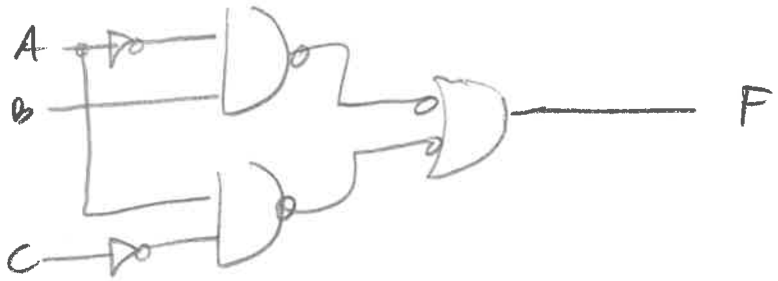
b)



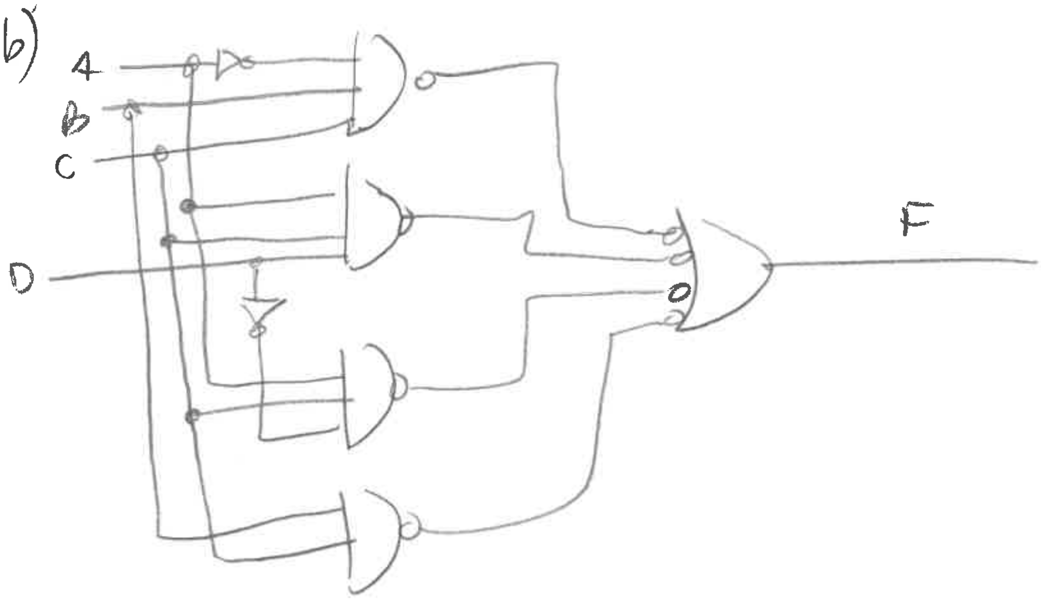
$F(L) = [(\bar{A}\cdot\bar{B}) + (A\cdot\bar{C}]](L)$

18)

a)

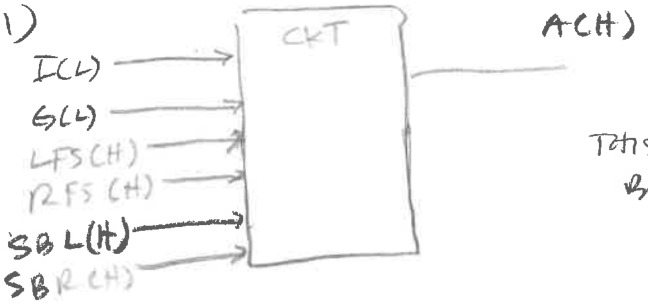


b)



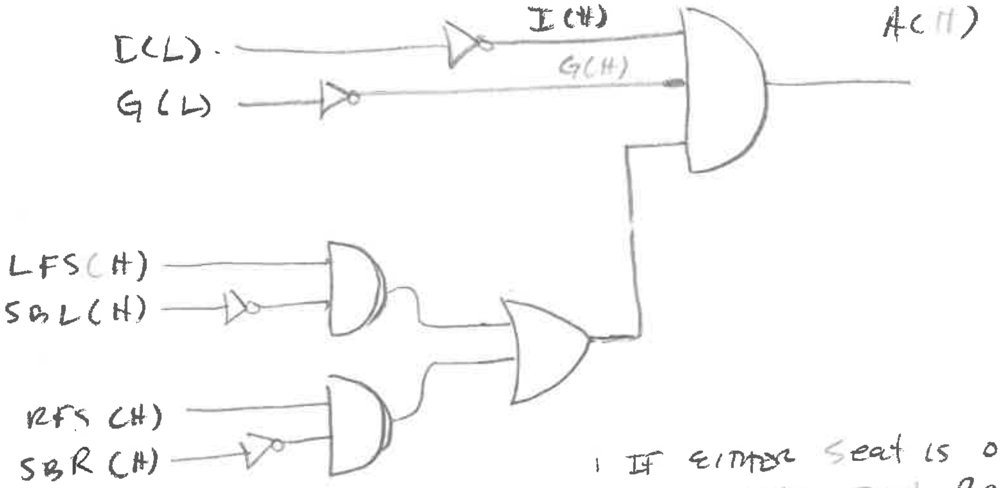
CHAPTER 14 Design Problems

1)



This could be 64 Row Truth Table, But Let's instead Reason it out using Logic

ALARM will Sound if IGNITION is ON, THE CAR IS IN Gear, AND occupied seats DO NOT HAVE SEAT BELTS ON.



If either Seat is occupied But the Seat Belt is not on, then the car should not start

No control!

2)

A	B	C	D

A (L)
B (L)
C (L)
D (L)

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

TWO OR more ADJACENT Slots mean two zero next to each other

1	1	0	1
1	1	0	0
0	0	0	0
1	1	1	1

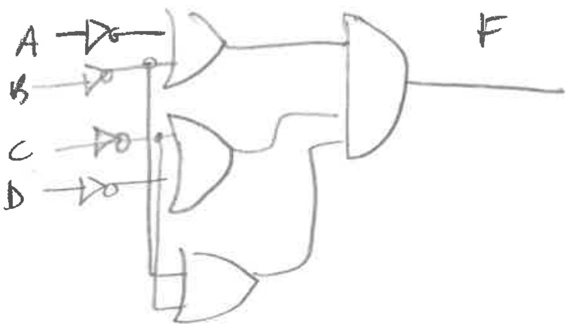
$F = \bar{A}\bar{C} + A\bar{B} + \bar{B}\bar{C}$

$\bar{F} = AB + CD + BC$

$\bar{F} = \overline{AB}CD + \bar{A}BC\bar{D} + \bar{A}BCD + A\bar{B}CD + AB\bar{C}\bar{D} + AB\bar{C}D + ABC\bar{D} + ABCD$

$\bar{F} = AB + CD + BC$ (PLUG into reduction SOFTWARE)

$F = (\bar{A}+\bar{B})(\bar{C}+\bar{D})(\bar{B}+\bar{C})$



No control

CHAPTER 15 EXERCISE

1) THE INTERFACE OF A MODULE IS THE SET OF MODULES INPUTS AND OUTPUTS

2) ITERATIVE MODULAR DESIGN IS A SPECIAL FORM OF MODULAR DESIGN WHERE THE SAME SET OF MODULES ARE USED IN AN ITERATIVE MANNER IN THE DESIGN

3) SELF-COMMENTING MAKES THE MODULES EASIER FOR HUMANS TO UNDERSTAND. SPECIFICALLY, IT HELPS WITH THE FOLLOW

a) implementation of NEW DESIGNS

b) UNDERSTANDING OF EXISTING DESIGNS

c) Simulation of CURRENT DESIGNS

4) MODULAR DESIGN DOES NOT RELY ON A TRUTH-TABLE SO YOU CAN DESIGN CIRCUITS WITH A GREATER NUMBER OF INPUTS.

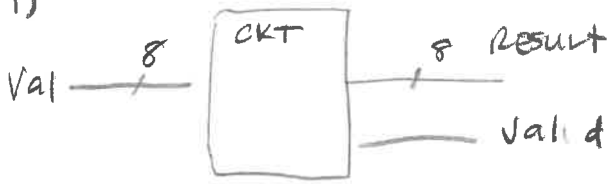
5) EVERYONE KNOWS (OR SHOULD KNOW) HOW all THE FOUNDATION MODULES OPERATE, THEREFORE THERE IS NO NEED TO RESPECIFY THEIR DESIGN AND OPERATION

6) USING EXISTING MODULES (FOUNDATION MODULES) IS MORE EFFICIENT THAN CREATING NEW MODULES BECAUSE THE NEW MODULES MUST BE DESIGNED AND VERIFIED BEFORE THEY CAN BE USED

7) BOTH IMD & BFD ARE LIMITED. BFD IS LIMITED BY THE NUMBER OF INPUTS. IMD IS LIMITED to A Relatively Small Set OF CIRCUITS. BOTH IMD & BFD ARE inefficient Based on THE NOTION THAT THERE ARE MORE Efficiant WAYS to Design CIRCUITS.

CHAPTER 15 DESIGN Problems

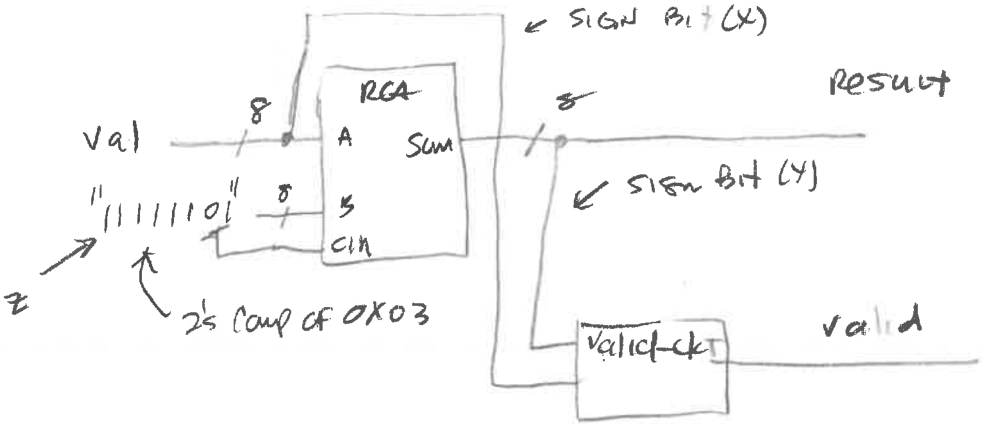
1)



Z	X	Y	
0	0	0	–
0	0	1	–
0	1	0	–
0	1	1	–
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

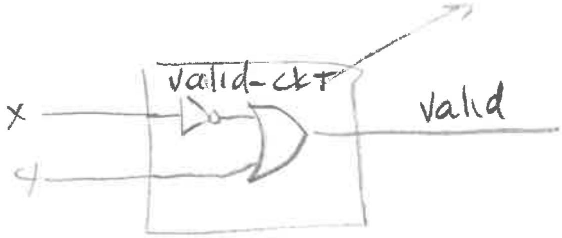
Z is always 1

← not Valid



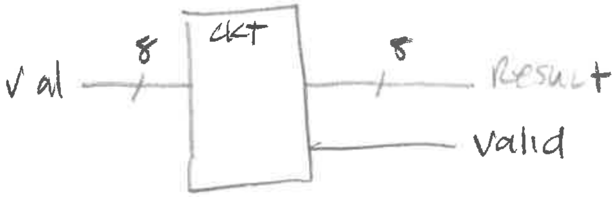
$$\overline{F} = X \cdot \overline{Y}$$

$$F = (\overline{X} + Y)$$



No control

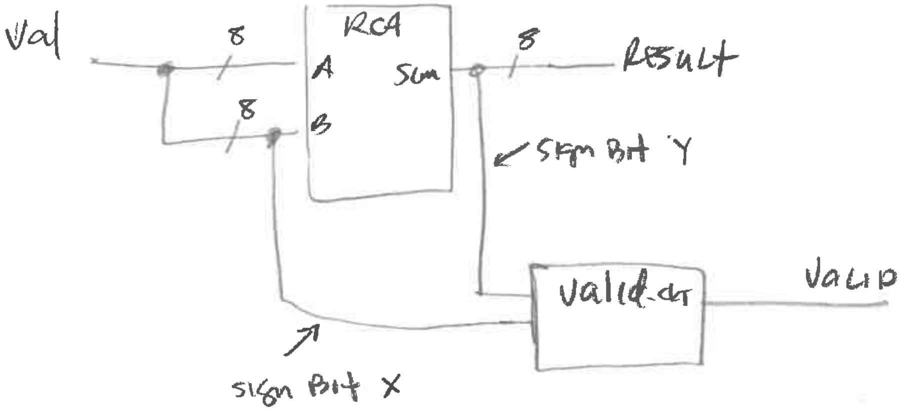
2)

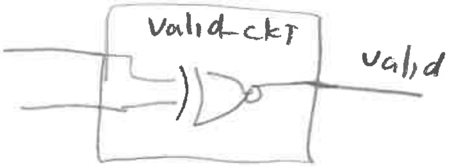


Valid_ckt

X	Y	Valid
0	0	1
0	1	0
1	0	0
1	1	1

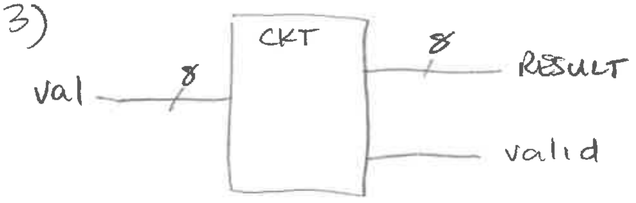
$$\text{Valid} = \overline{A \oplus B}$$



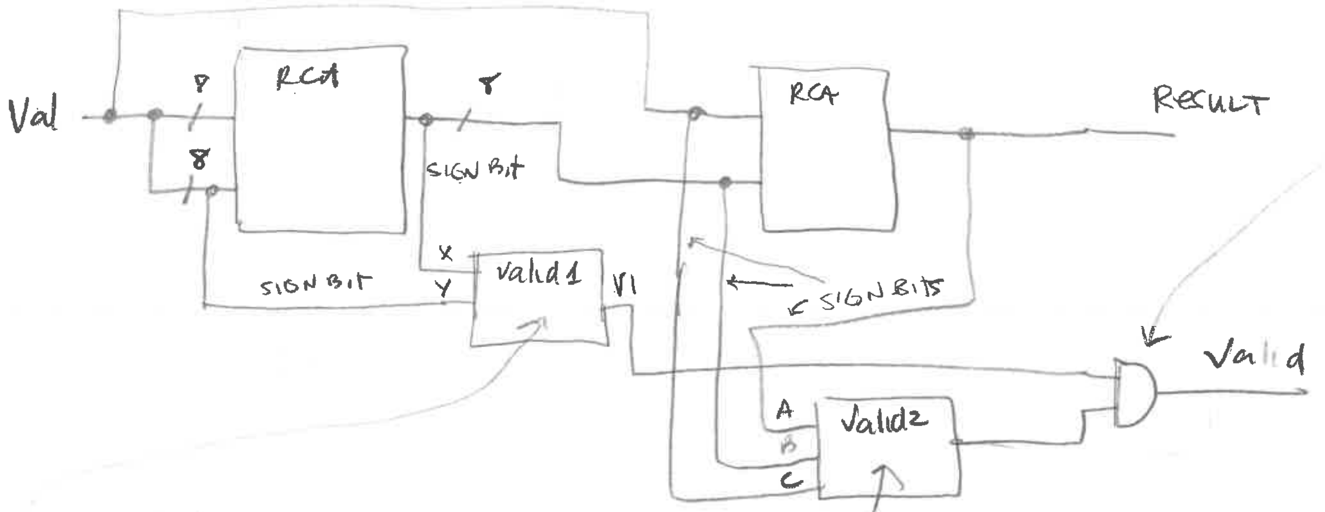


NO control

3)



THE PROBLEM REQUIRES 2 RCAS. THE FINAL RESULT IS Valid only when THE RESULT OF BOTH ADDITIONS (valid1 & valid2) are valid



valid2

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

$\overline{\text{VaLID2}} = \bar{A}\bar{B}C + AB\bar{C}$

$\text{Valid2} = (A+B+\bar{C})(\bar{A}+\bar{B}+C)$

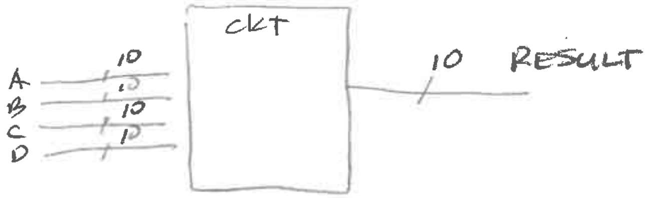
Valid1

X	Y	V1
0	0	1
0	1	0
1	0	0
1	1	1

$\text{Valid1} = \overline{X \oplus Y}$

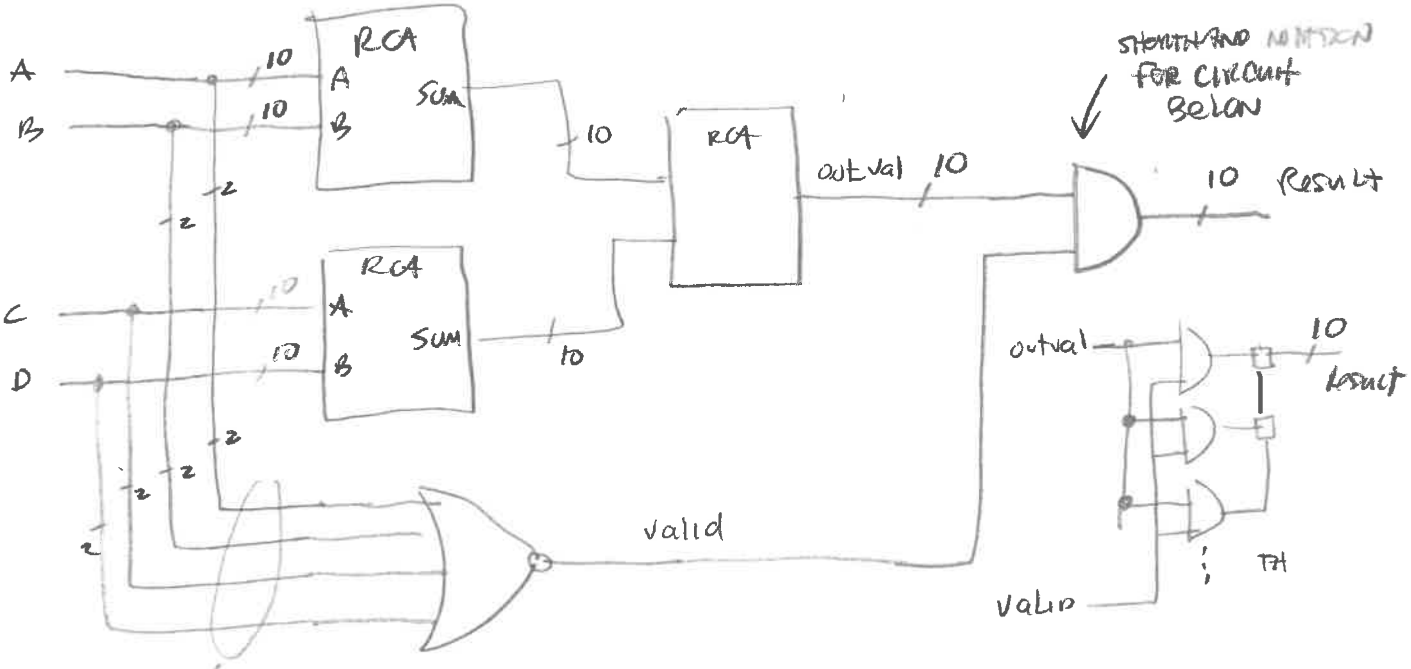
NO CONTROL

4)



THE issue is THAT IF all 4 input values are less than 256, THE 10-bit output will ALWAYS Be valid.

A 10-BIT value will Be less than 256 is THE TWO MSB's are BOTH zero

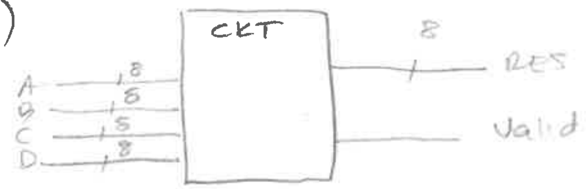


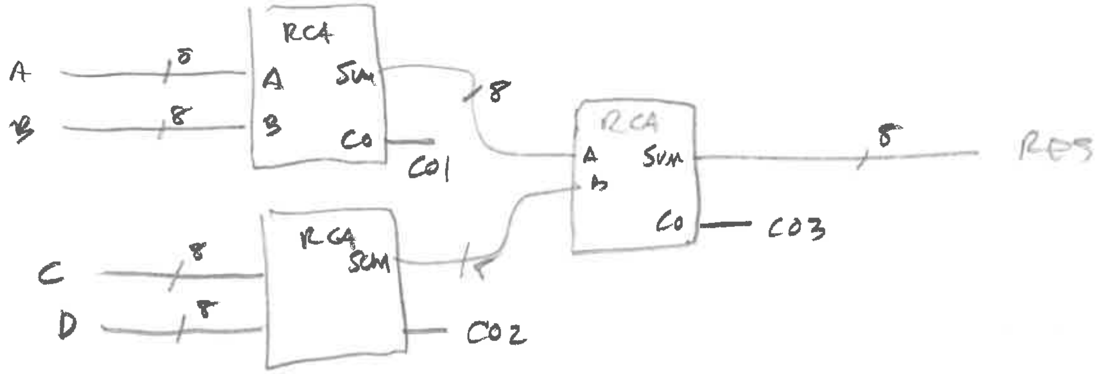
SHORTHAND NOTATION FOR CIRCUIT BELOW

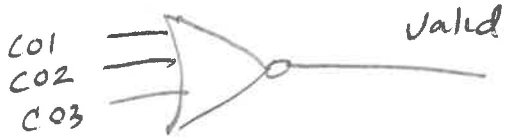
TWO MSB's OF EACH OF A, B, C, D } IF ALL OF THE inputs ARE ZERO then the output of this 8-input NOR GATE is 1

ENTERNAL control

5)



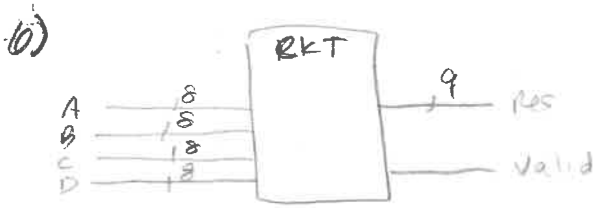


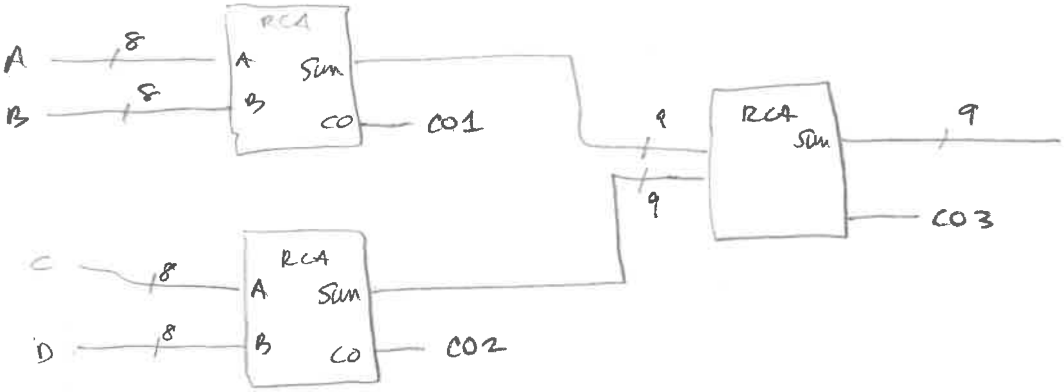


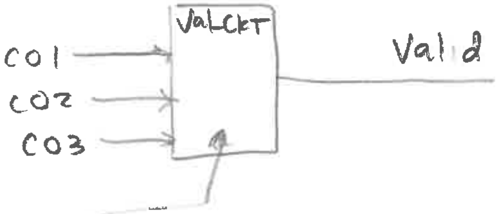
↑ THE CO's from all 3 RCAs must Be '0' in order for the answer to Be valid

NO control

6)







CO1	CO2	CO3	Valid
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

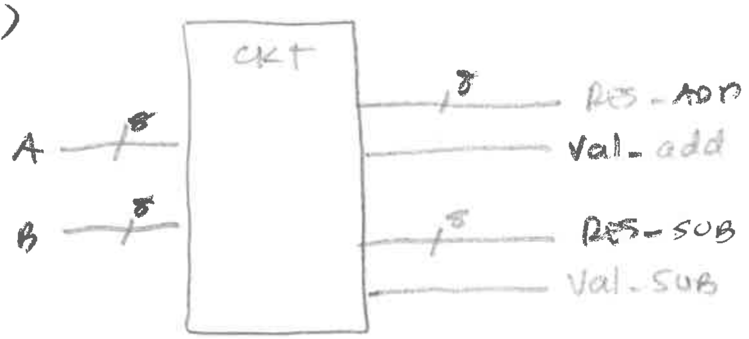
$$\overline{Valid} = \overline{C01}\,C02\,C03 + C01\,C02\,\overline{C03} + C01\,C02\,C03$$

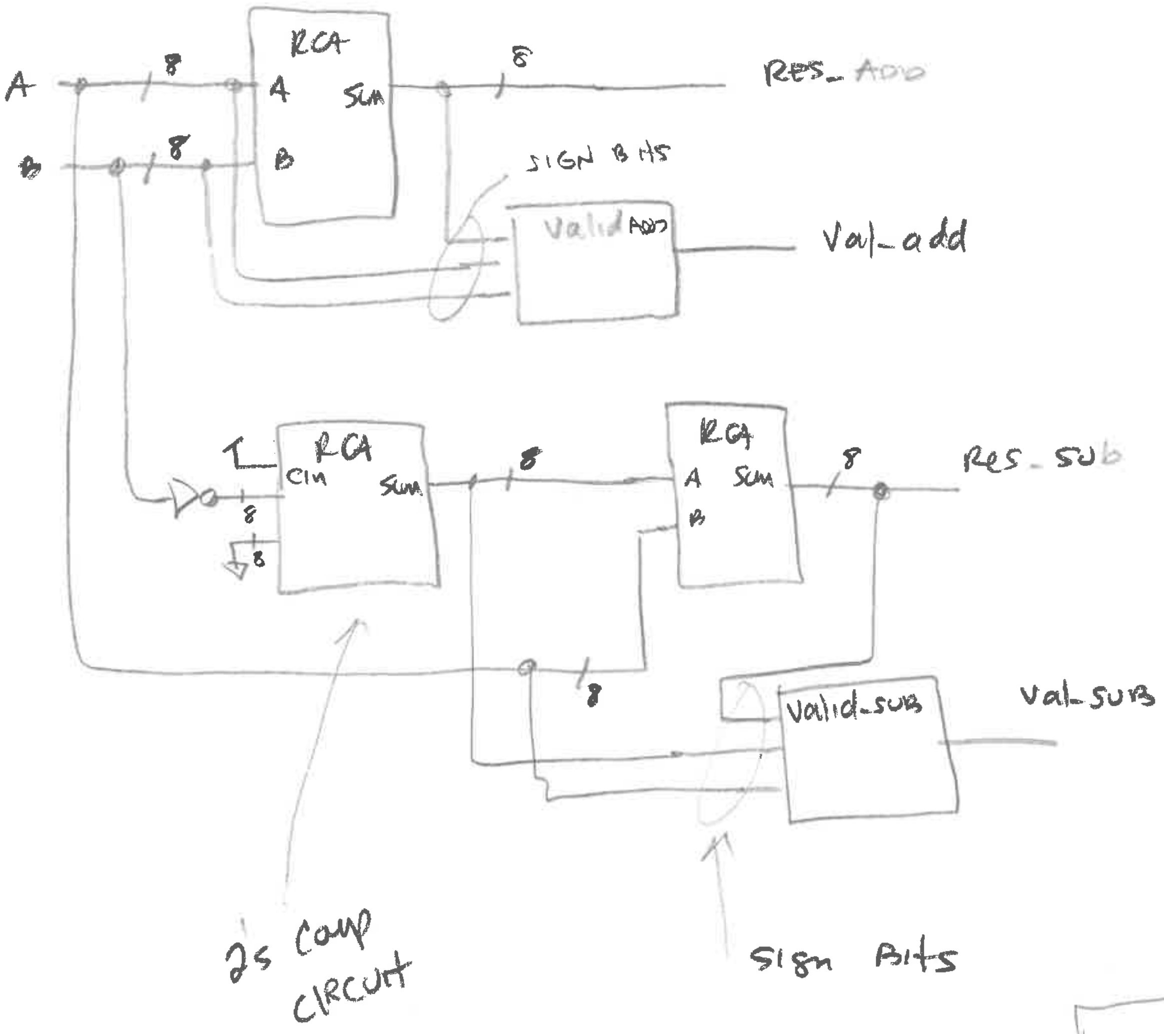
$$Valid = (C01 + \overline{C02} + \overline{C03})(\overline{C01} + \overline{C02} + C03)(\overline{C01} + \overline{C02} + \overline{C03}$$

↑ let me know if you don't agree with this logic

NO control

7)

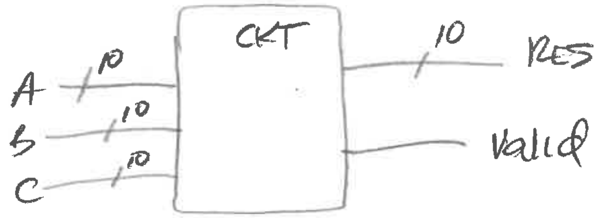




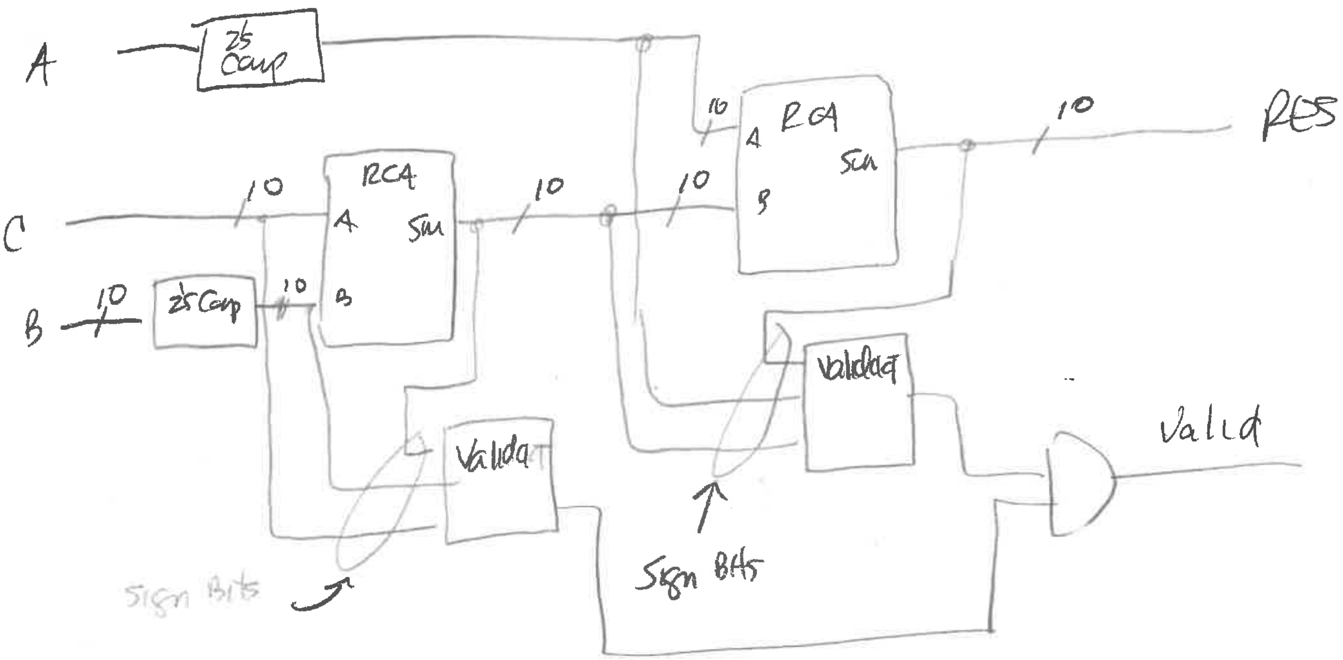
NO CONTROL

Valid_sub & Valid_add ckts are
TYPICAL 3-input Validity checking
circuts FOR NUMBERS in RC FORMAT

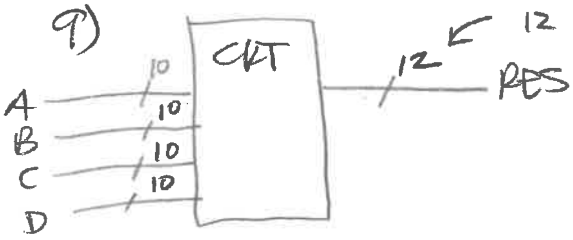
8)



No control

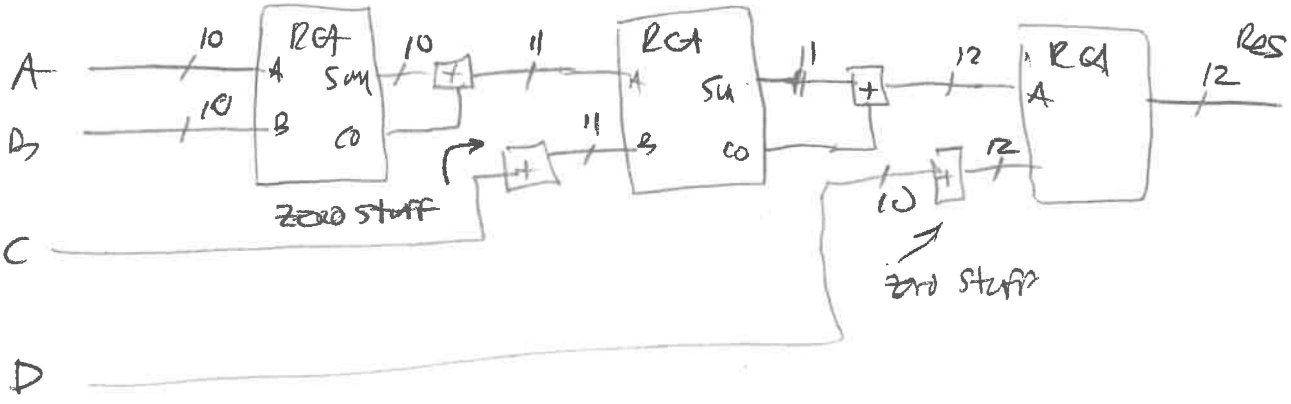


9)

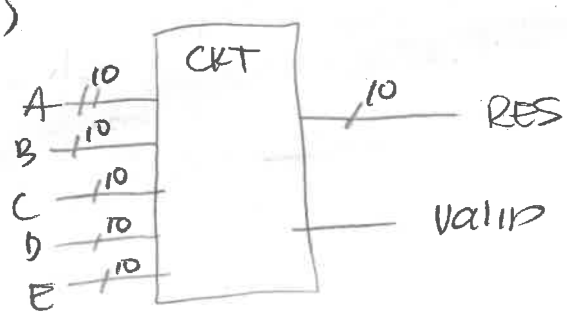


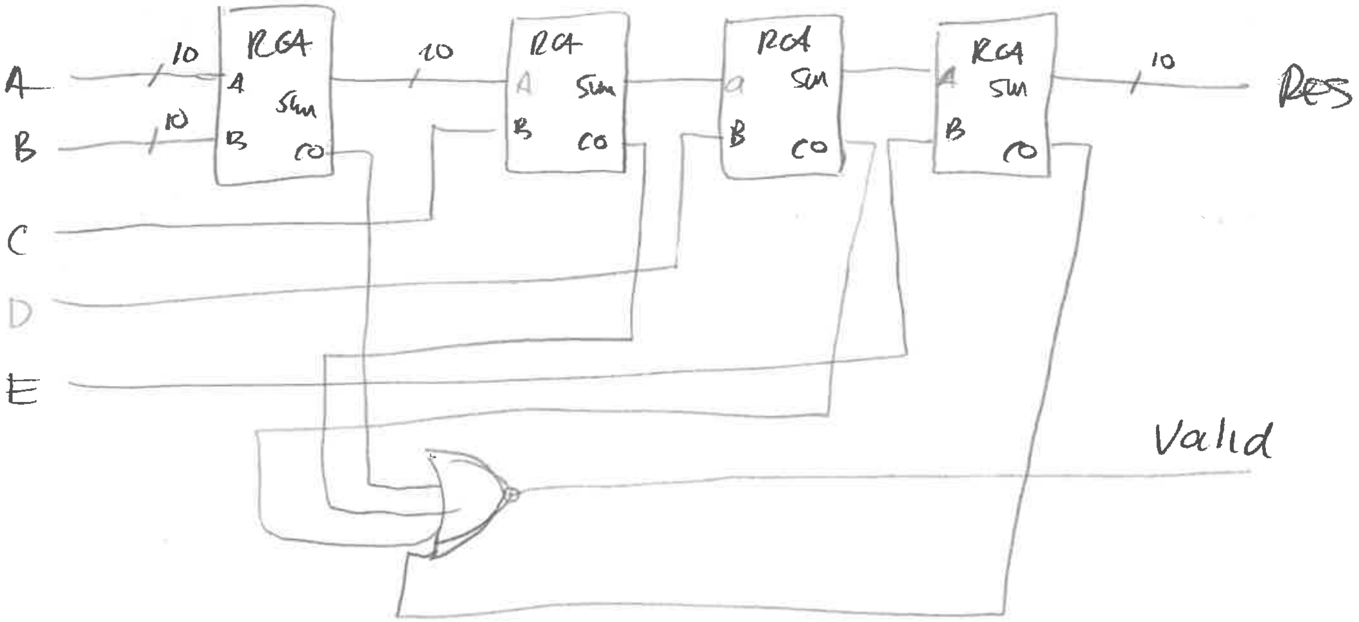
12 Bits are needed for Valid result

No control



10)


CKT
A
10
B
10
C
10
D
10
E
10
10
RES
valid


A
10
B
10
C
D
E
RCA
A
B
Sum
CO
10
RCA
A
B
Sum
CO
RCA
a
B
Sum
CO
RCA
A
B
Sum
CO
10
Res
Valid

CHAPTER 16 EXERCISES

1) a) $F = A\bar{B} + \bar{A}\bar{C}$

$= A\bar{B}(C+\bar{C}) + \bar{A}\bar{C}(B+\bar{B})$

$= \underset{5}{A\bar{B}C} + \underset{4}{A\bar{B}\bar{C}} + \underset{2}{\bar{A}B\bar{C}} + \underset{0}{\bar{A}\bar{B}\bar{C}}$

A	B	C	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

b) $F = \bar{A}\bar{B}D + \bar{A}BD + B\bar{C}D + A\bar{B}D$

$F = \bar{A}\bar{B}D(C+\bar{C}) + \bar{A}BD(C+\bar{C}) + B\bar{C}D(A+\bar{A}) + A\bar{B}D(C+\bar{C})$

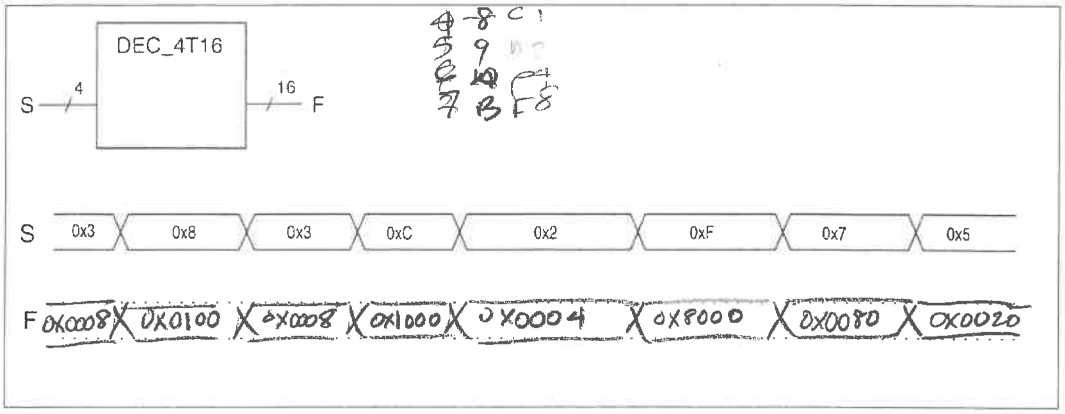
$= \underset{(3)}{\bar{A}\bar{B}CD} + \underset{(1)}{\bar{A}\bar{B}\bar{C}D} + \underset{(7)}{\bar{A}BCD} + \underset{(5)}{\bar{A}B\bar{C}D} + \underset{13}{AB\bar{C}D} + \underset{(5)}{\bar{A}B\bar{C}D} +$

$\underset{(11)}{A\bar{B}CD} + \underset{(9)}{A\bar{B}\bar{C}D}$

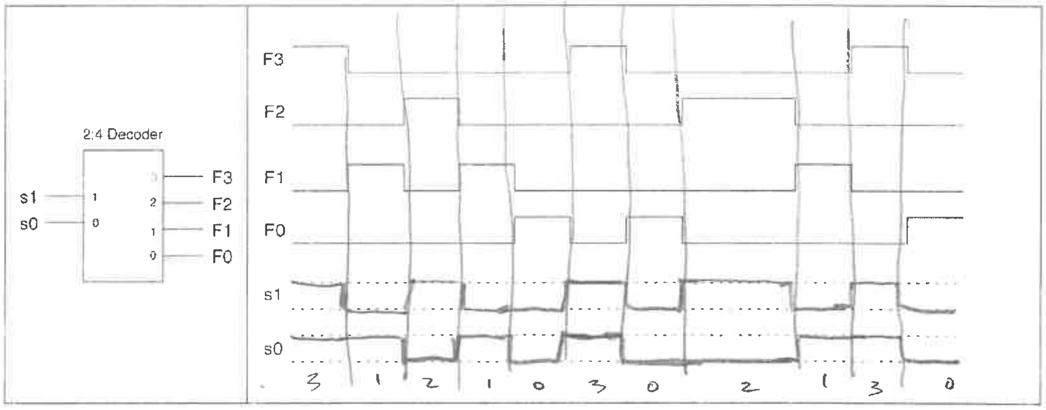
A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

← There's a table; TRANSLATE TO DECODER

4) Use the following black box model for a standard 2:4 decoder to complete the following timing diagram.



5) Based on the standard 2:4 Decoder shown below, complete the following timing diagram by entering the values for signals s1 and s2 that would generate the listed output waveforms. Assume that propagation delays are negligible. Be sure to annotate you solution to this problem.



0 ⇒ 0001
1 ⇒ 0010
2 ⇒ 0100
3 ⇒ 1000

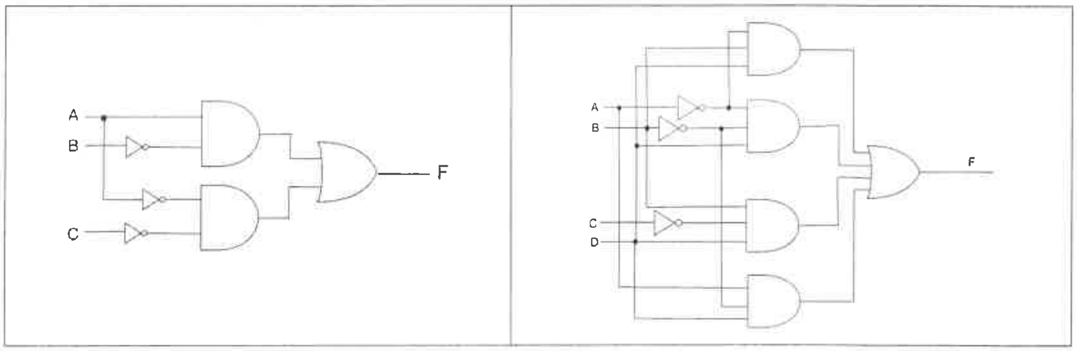
6) STANDARD DECODERS HAVE SPECIFIC OUTPUT FORMS (ONE-HOT output) AND THE classic $n : 2^n$ input/output Relationship.

Generic Decoders HAVE NO constraints on THE NUMBER OF INPUTS OR outputs, AND THERE IS NO GIVEN RELATION Between THE NUMBER OF INPUTS AND outputs AS there is in STANDARD DECODERS.

3

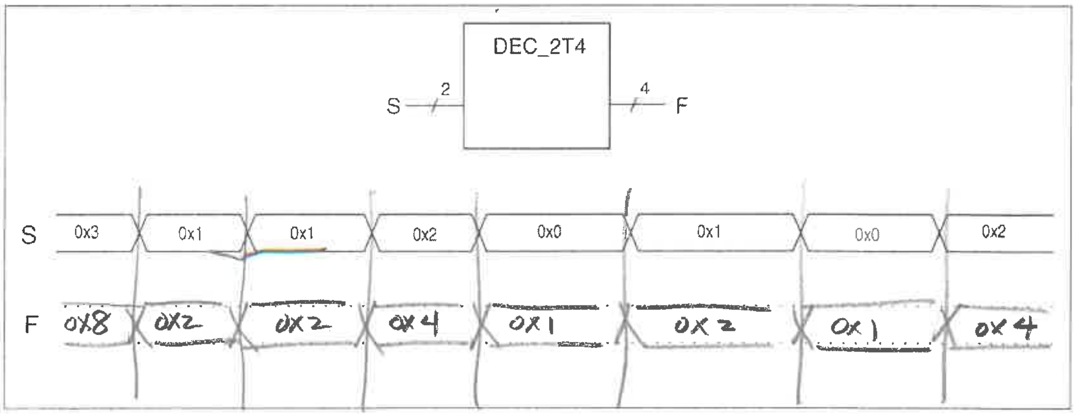
Chapter Exercises

1) Implement the following functions using a generic decoder.

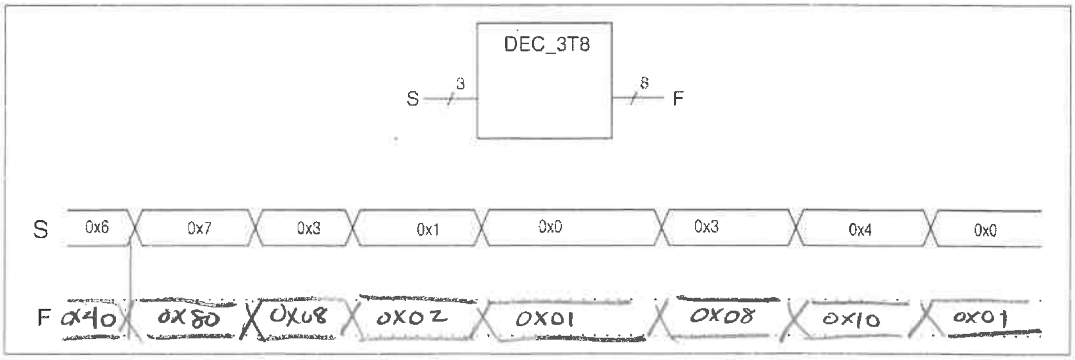


(a) (b)

2) Use the following black box model for a standard 2:4 decoder to complete the following timing diagram.

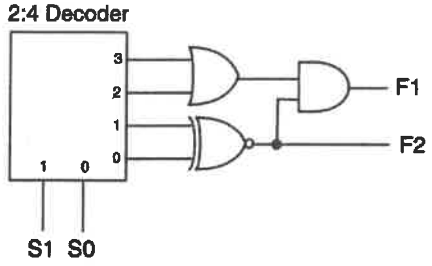


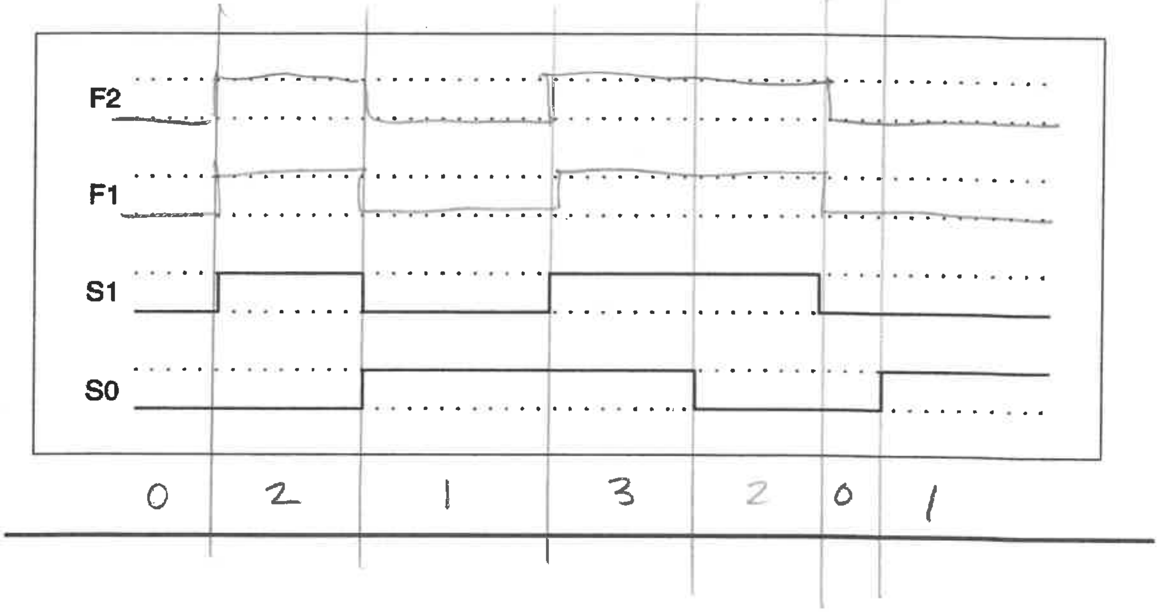
3) Use the following black box model for a standard 2:4 decoder to complete the following timing diagram



4

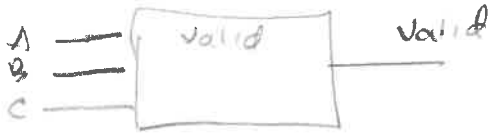
7) Use the schematic diagram to complete the F2 and F1 outputs of the provided timing diagram. Consider the decoder to be a standard 2:4 decoder. Assume that propagation delays are too small to worry about.





CHAPTER 16 DESIGN PROBLEM

1) If sign bits of two input operands ARE THE SAME, AND THE RESULT SIGN BIT IS Different, THE ANSWER IS not valid



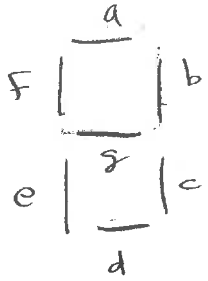
A & B = input sign bits
C = Result sign Bit

A	B	C	Valid
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

TRANSLATE TABLE TO DECODER

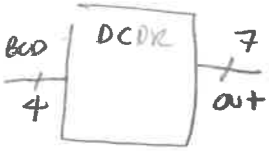
NO Control

2)



Dec	BCD	a	b	c	d	e	f	g
0	0000	1	1	1	1	1	1	0
1	0001	0	1	1	0	0	0	0
2	0010	1	1	0	1	1	0	1
3	0011	1	1	1	1	0	0	1
4	0000	0	1	1	0	0	1	1
5	0101	1	0	1	1	0	1	1
6	0110	0	0	1	1	1	1	1
7	0111	1	1	1	0	0	0	0
8	1000	1	1	1	1	1	1	1
9	1001	1	1	1	0	0	1	1
10								
11								

NOT DEFINED FOR NUMBERS AFTER 9



out = abcdefg

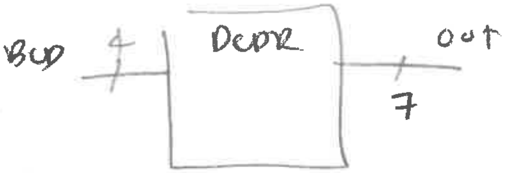
NO Control

3)

Dec	BCD	a	b	c	d	e	f	g
0	0000	1	1	1	1	1	1	0
1	0001	0	0	0	0	1	1	0
2	0010	1	0	1	1	0	1	1
3	0011	1	0	0	1	1	1	1
4	0100	0	1	0	0	1	1	1
5	0101	1	1	0	1	1	0	1
6	0110	0	1	1	1	1	0	1
7	0111	1	0	0	1	1	0	0
8	1000	1	1	1	1	1	1	1
9	1001	1	1	0	0	1	1	1
	1 1 0							
	1 1 1							

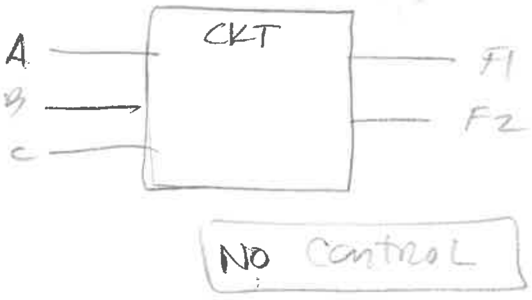
not defined

NO Control



out = abcdefg

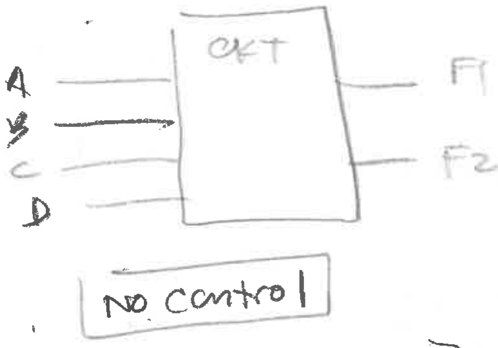
4)



A	B	C	F1	F2
0	0	0	1	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	1
1	1	1	0	1

TRANSLATE TABLE TO DECODER

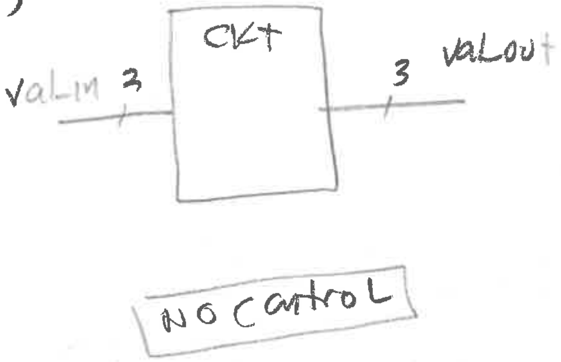
5)



→ TRANSLATE TABLE to DECODER

A	B	C	D	FEVN	FODD
0	0	0	0	1	0
0	0	0	1	0	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	1	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	1
1	1	0	0	0	0
1	1	0	1	0	1
1	1	1	0	0	0
1	1	1	1	0	1

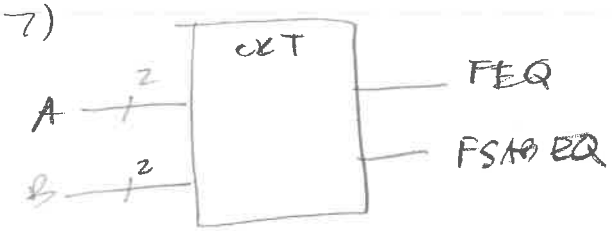
6)



Val_in			valout		
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	0	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	0	0	0
1	1	1	0	0	1

TRANSLATE TABLE TO DECODER

7)

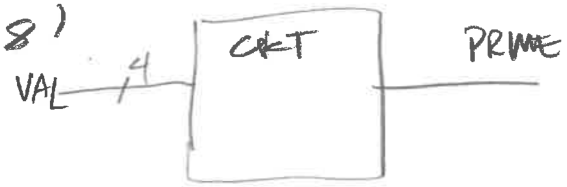


NO Control

A	B	FEQ	FSABEQ
0 0	0 0	0	1
0 0	0 1	1	0
0 0	1 0	1	0
0 0	1 1	1	0
0 1	0 0	0	0
0 1	0 1	1	1
0 1	1 0	0	1
0 1	1 1	0	0
1 0	0 0	0	0
1 0	0 1	0	1
1 0	1 0	1	1
1 0	1 1	0	0
1 1	0 0	0	0
1 1	0 1	0	0
1 1	1 0	0	0
1 1	1 1	1	1

TRANSLATE TABLE to DECODER

8)



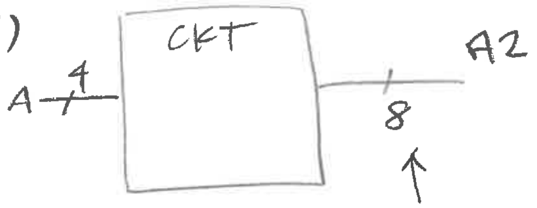
NO Control

Dec	Val	PRIME
0	0 0 0 0	– ← Don't care
1	0 0 0 1	1
2	0 0 1 0	1
3	0 0 1 1	1
4	0 1 0 0	0
5	0 1 0 1	1
6	0 1 1 0	0
7	0 1 1 1	1
8	1 0 0 0	0
9	1 0 0 1	0
10	1 0 1 0	0
11	1 0 1 1	1
12	1 1 0 0	0
13	1 1 0 1	1
14	1 1 1 0	0
15	1 1 1 1	0

TRANSLATE TABLE to DECODER

4.

9)



Largest input value
15   15 x 15 = 225

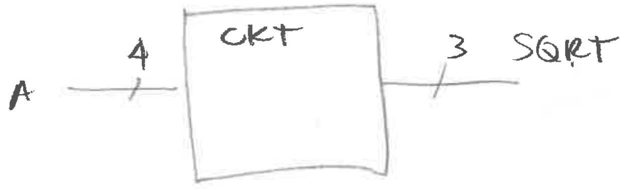
$\lceil \log_2 225 \rceil = 8$

No control

A	A2								
	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	
0 0 0 0	0	0	0	0	0	0	0	0	0
0 0 0 1	0	0	0	0	0	0	0	1	1
0 0 1 0	0	0	0	0	0	1	0	0	4
0 0 1 1	0	0	0	0	1	0	0	1	9
0 1 0 0	0	0	0	1	0	0	0	0	16
0 1 0 1	0	0	0	1	1	0	0	1	25
0 1 1 0	0	0	1	0	0	1	0	0	36
0 1 1 1	0	0	1	1	0	0	0	1	49
1 0 0 0	0	1	0	0	0	0	0	0	64
1 0 0 1	0	1	0	1	0	0	0	1	81
1 0 1 0	0	1	1	0	0	1	0	0	100
1 0 1 1	0	1	1	1	1	0	0	1	121
1 1 0 0	1	0	0	1	0	0	0	0	144
1 1 0 1	1	0	1	0	1	0	0	1	169
1 1 1 0	1	1	0	0	0	1	0	0	196
1 1 1 1	1	1	1	0	0	0	0	1	225

TRANSLATE TABLE
into Decoder

10)

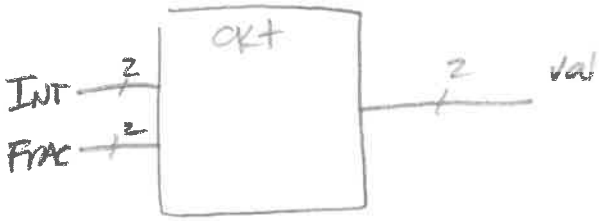


No Control

TRASLATE TABLE
into Decoder

A	SQRT
0 0 0 0	0 0 0
0 0 0 1	0 0 1
0 0 1 0	0 0 1
0 0 1 1	0 1 0
0 1 0 0	0 1 0
0 1 0 1	0 1 0
0 1 1 0	0 1 0
0 1 1 1	0 1 1
1 0 0 0	0 1 1
1 0 0 1	0 1 1
1 0 1 0	0 1 1
1 0 1 1	0 1 1
1 1 0 0	0 1 1
1 1 0 1	1 0 0
1 1 1 0	1 0 0
1 1 1 1	1 0 0

11)

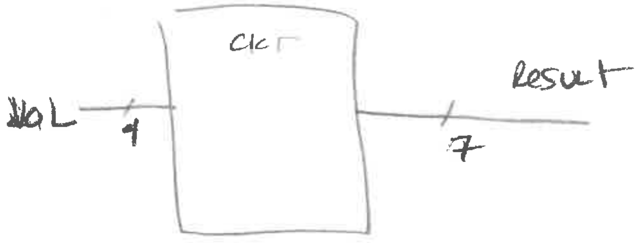


NO control

TRANSLATE TABLE to DECODER

int		FRAC		val		
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	0	1
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	1
1	0	1	1	0	1	1
1	1	0	0	0	1	1
1	1	0	1	0	1	1
1	1	1	0	1	0	0
1	1	1	1	1	0	0

(2)



val	DEC	Result
0 0 0 0	0	000 0000
0 0 0 1	1	000 0001
0 0 1 0	2	000 0100
0 0 1 1	3	000 1001
0 1 0 0	4	001 0000
0 1 0 1	5	001 1001
0 1 1 0	6	010 0100
0 1 1 1	7	011 0001
1 0 0 0	-8	100 0000
1 0 0 1	-7	011 0001
1 0 1 0	-6	010 0100
1 0 1 1	-5	001 1001
1 1 0 0	-4	001 0000
1 1 0 1	-3	000 1001
1 1 1 0	-2	000 0100
1 1 1 1	-1	000 0001

Lagest value for 4-bit RC input is $(-8)^2 = 64$

$2^6 = 64 \Rightarrow [0, 63]$

You need 7 bits

TRANSLATE TABLE into DECODER

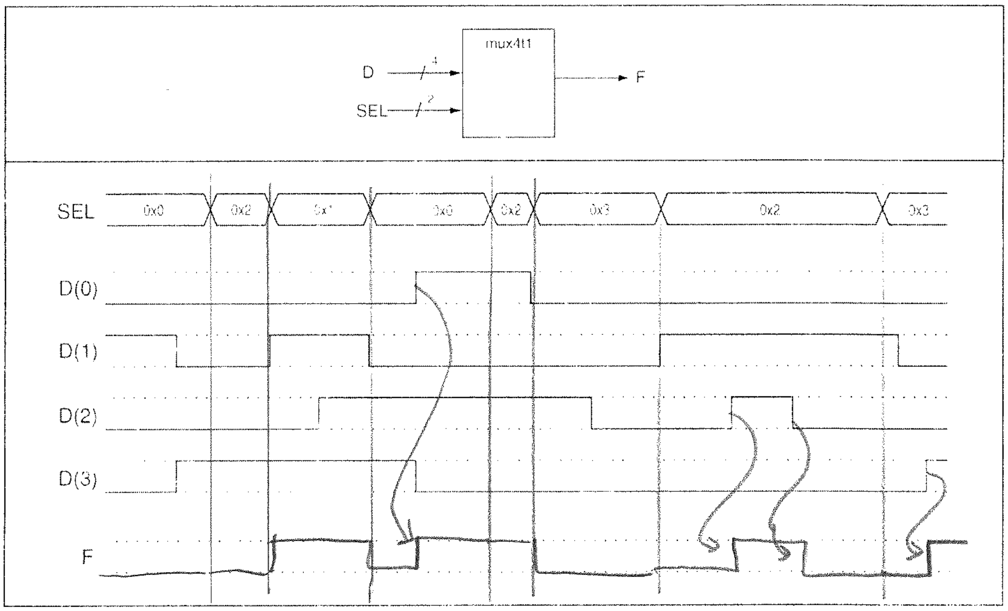
NO CONTROL

Chapter Exercises

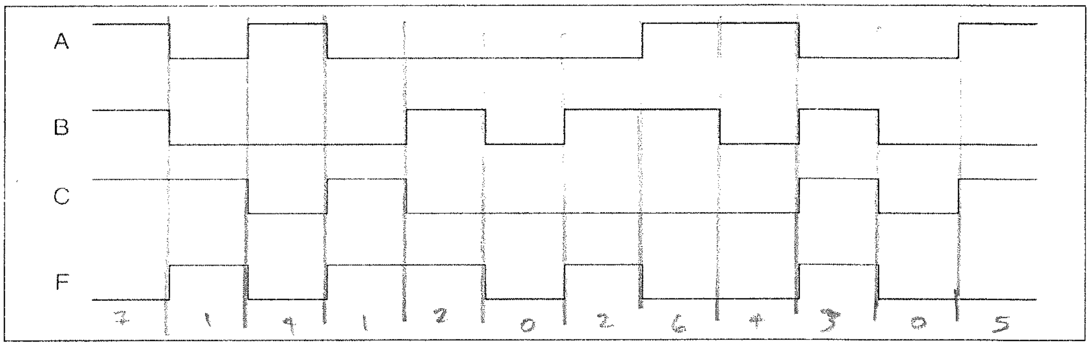
CHAPTER 17 EXERCISES

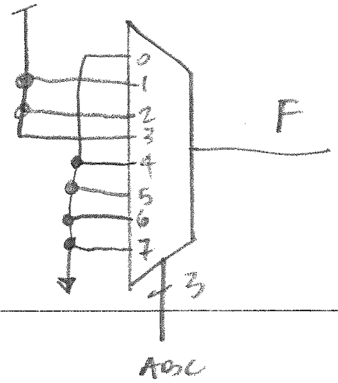
1) Briefly describe the special relationship between a MUX and a standard decoder. A MUX ESSENTIALLY HAS A DECODER EMBEDDED IN IT.

2) Use the following block diagram to complete the provided timing diagram. For this problem, consider the block diagram to represent a basic 4:1 MUX.



3) The following timing diagram completely defines a function F(A,B,C) that has been implemented on an 8:1 MUX. The control variables are A, B, and C (A is the most significant bit and C is the least significant bit) and the output is F. Write an expression for this function in reduced NAND/NAND form. Assume propagation delays are negligible.





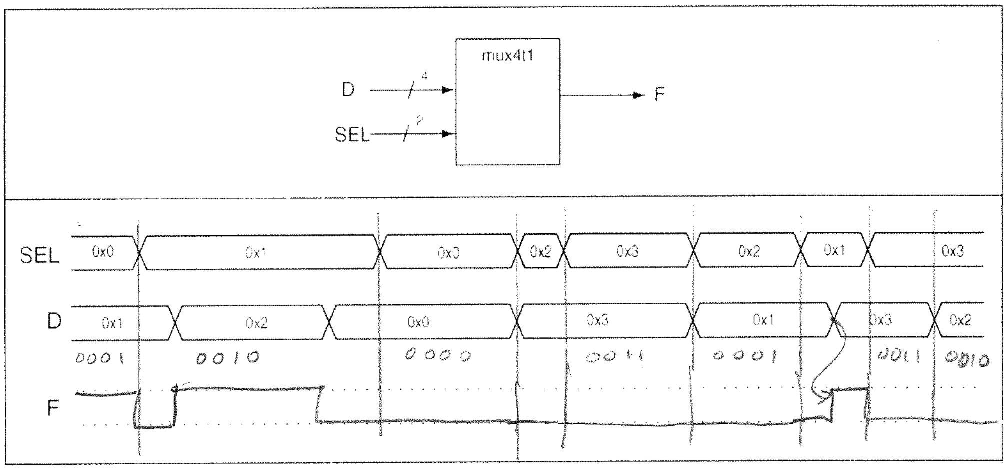
$$F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC$$

$$F = \overline{\overline{\bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC}}$$

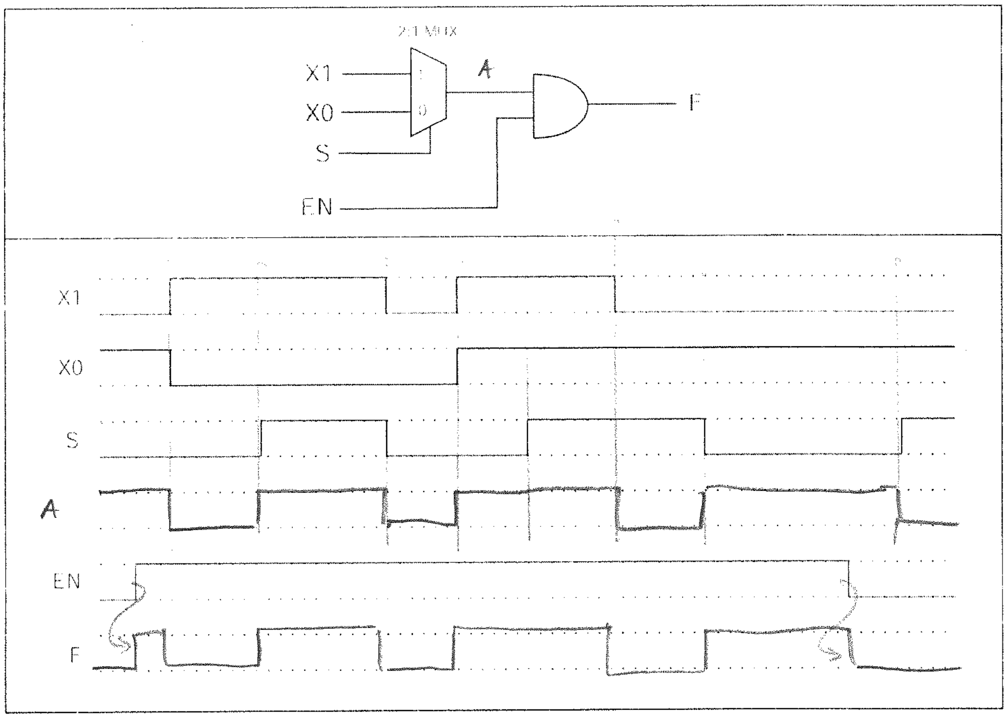
$$F = \overline{(\overline{\bar{A}\bar{B}C})(\overline{\bar{A}B\bar{C}})(\overline{\bar{A}BC})}$$

$\boxed{\text{NAND/NAND}}$

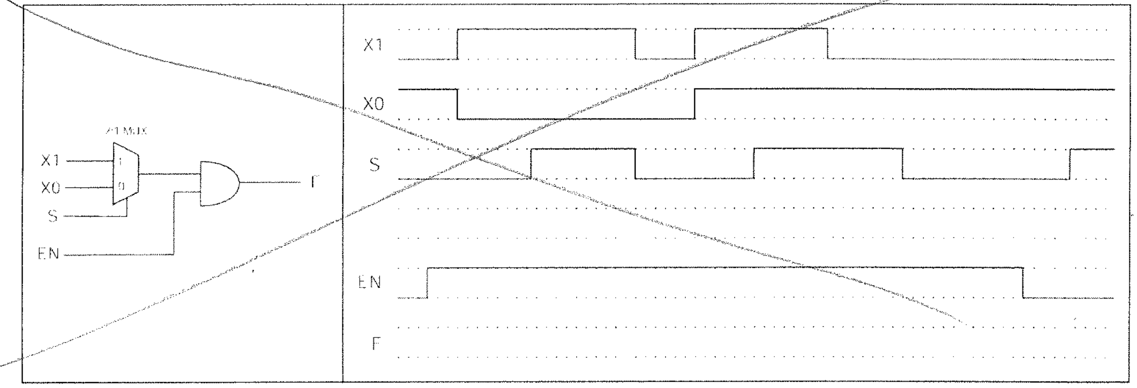
4) Use the following block diagram to complete the provided timing diagram. For this problem, consider the block diagram to represent a basic 4:1 MUX.



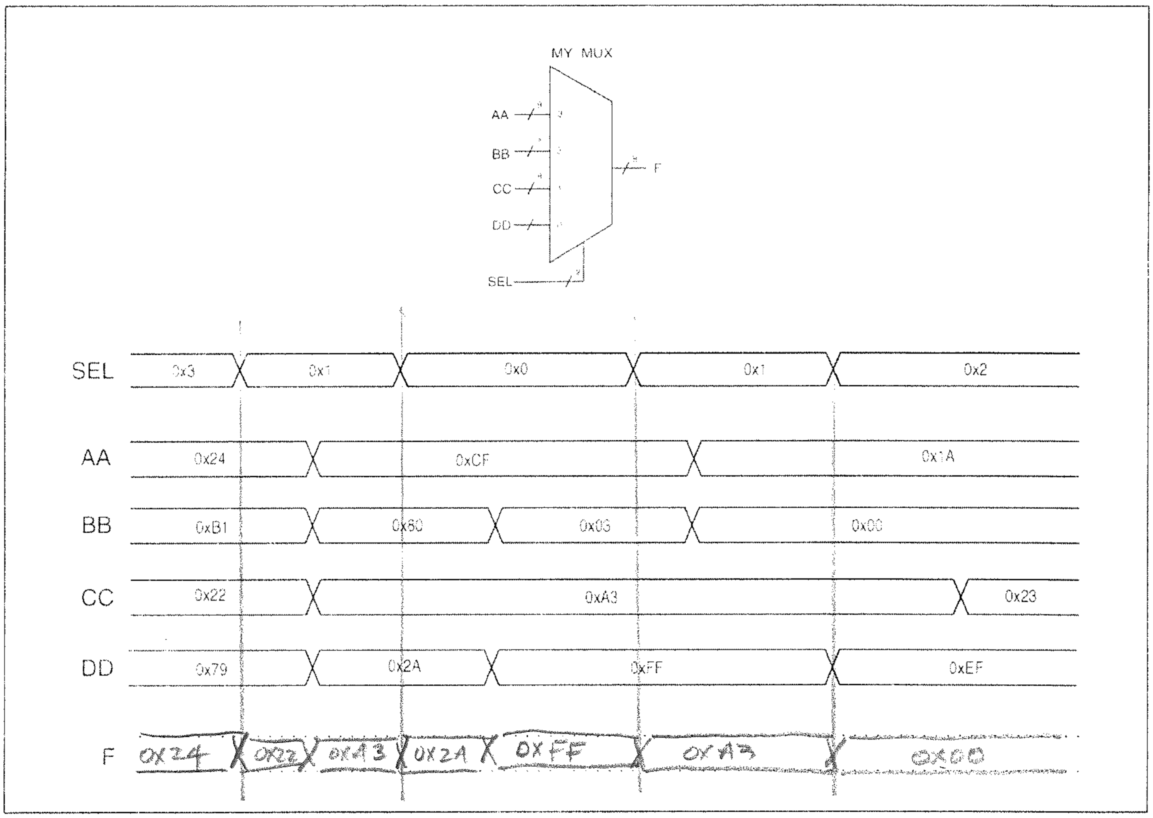
5) Use the listed circuit to complete signal F in the following timing diagram.



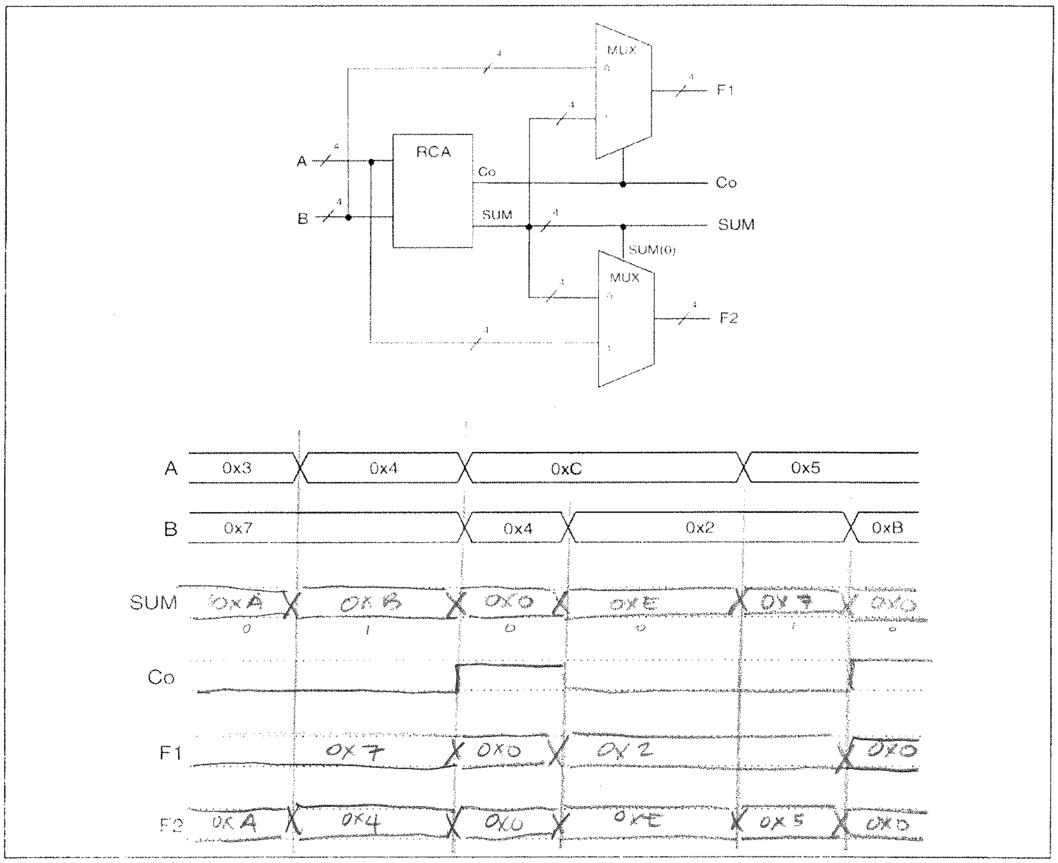
6) Use the listed circuit to fill signal F in the following timing diagram.



6) Using the following diagram of a 4:1 MUX, complete the provided timing diagram. Also provide a VHDL model that implements the 4:1 MUX.

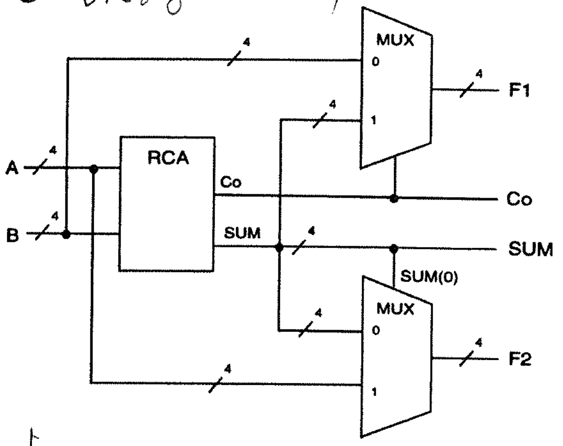


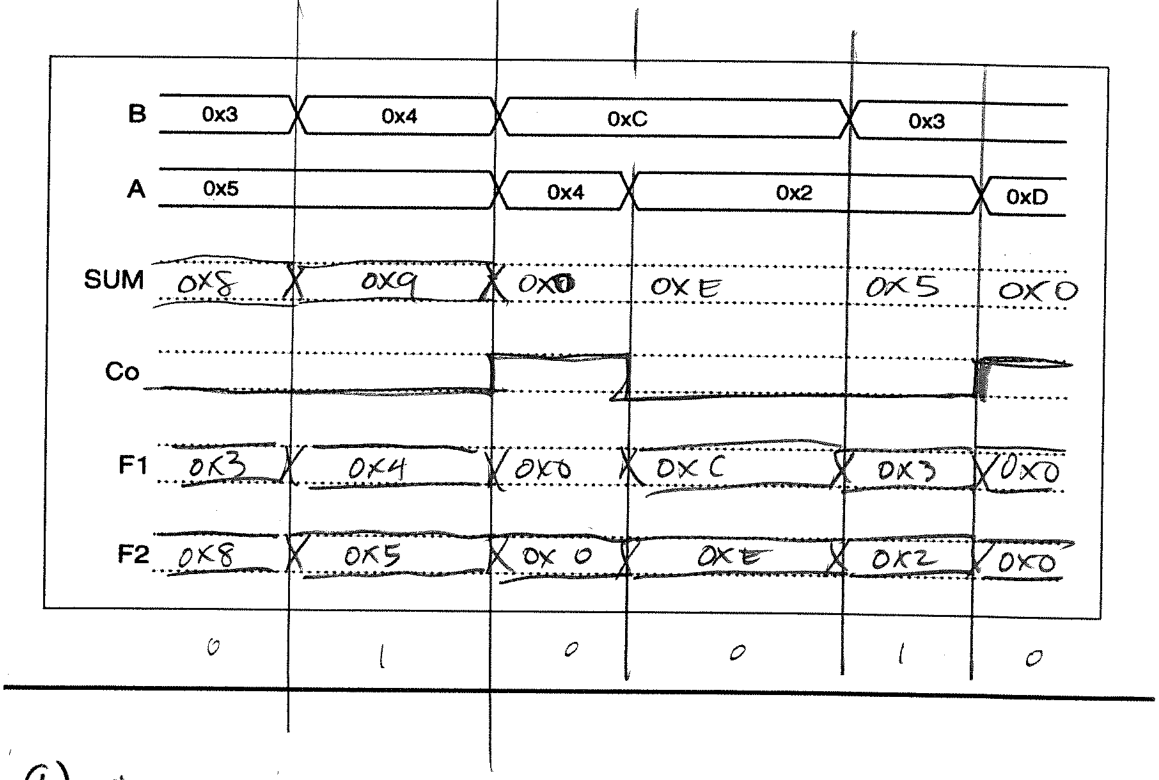
7 8) Use the following circuit diagram to complete the empty rows on the accompanying timing diagram. Use bus notation for all bundles (Co is the only non-bundle signal; 0x indicates hexadecimal).



9 Use the following circuit diagram to complete the empty rows on the accompanying timing diagram. Use bus notation for all bundles. ASSME UNSiged BiNARY

8)



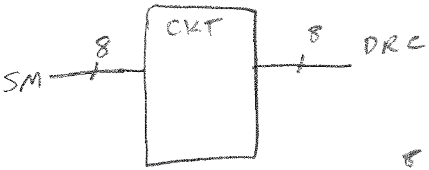


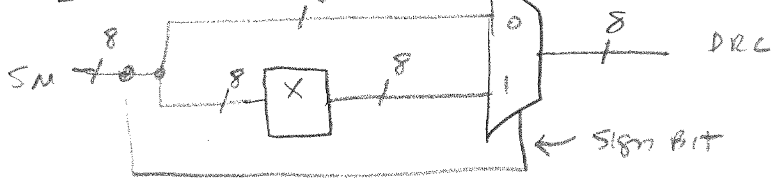
9) THE INTERNALS OF A MUX INCLUDES A STANDARD DECODER; IT'S THE DEVICE THAT SELECTS WHICH INPUT APPEARS ON THE OUTPUT

CHAPTER 17 DESIGN PROBLEMS

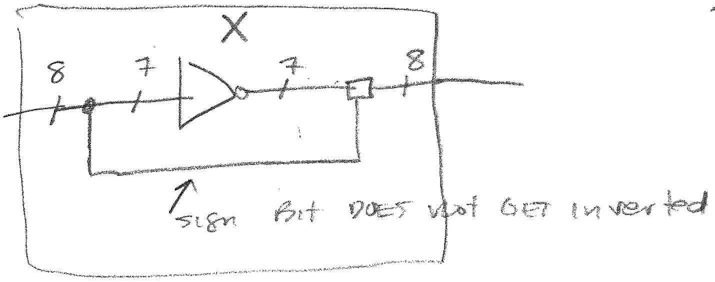
1)

SM → DRC





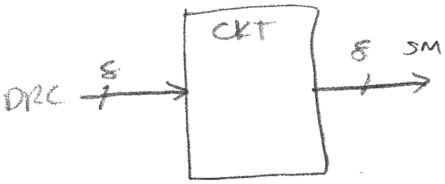
INTERNAL control



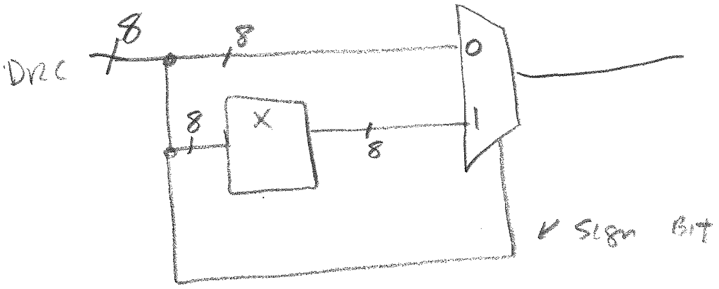
2) DRC → SM

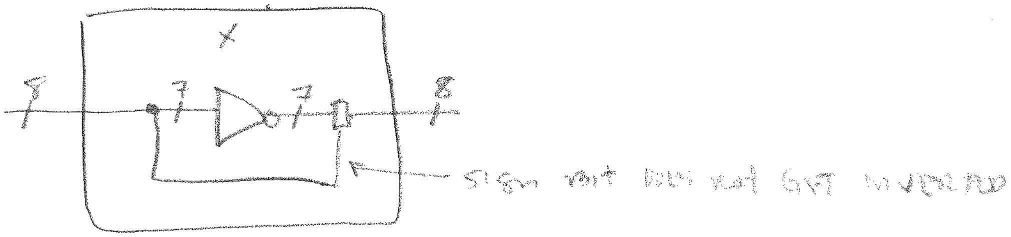
00111 11000

-7

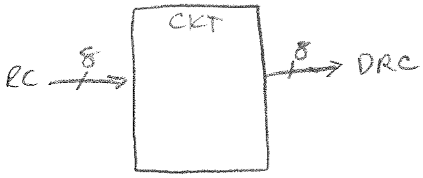


INTERNAL control

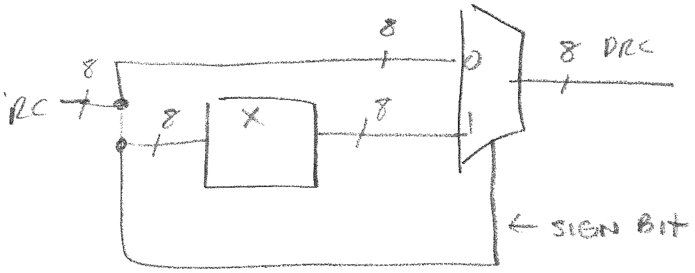




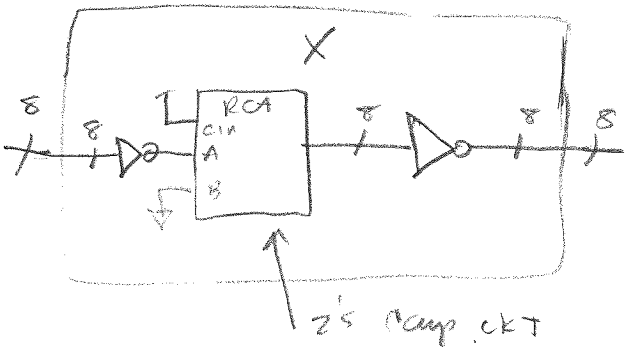
3)



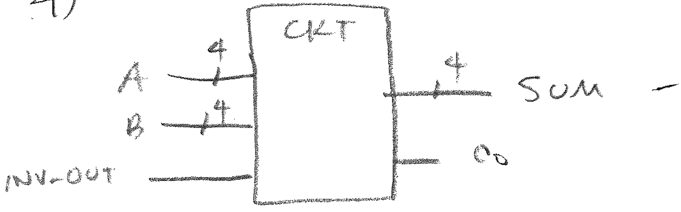
	RC	DRC
2	0010	0010
-2	1110	1101



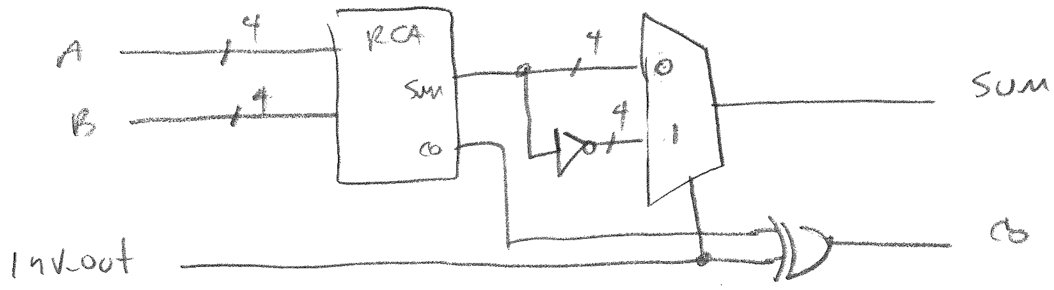
INTERNAL control



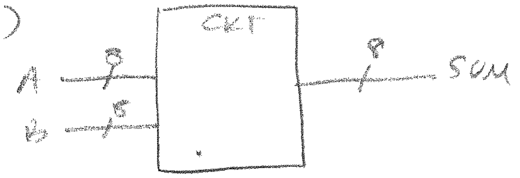
4)



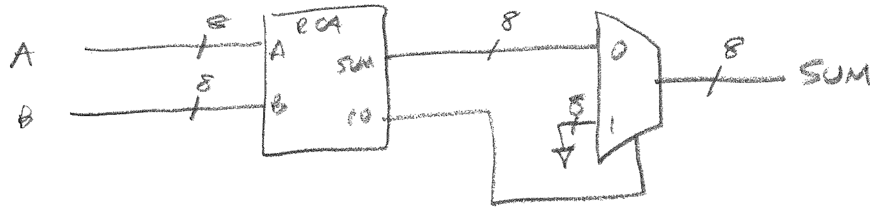
EXTERNAL Control



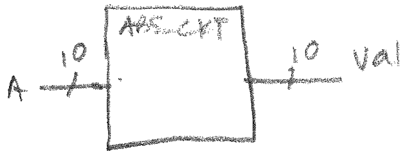
5)

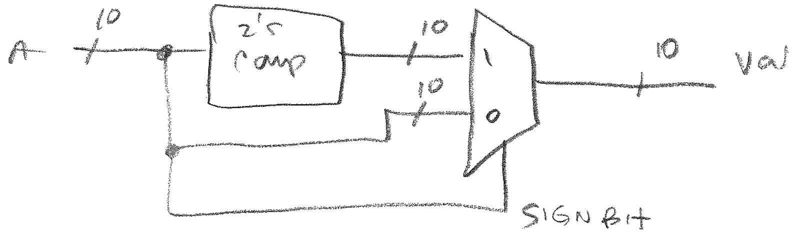


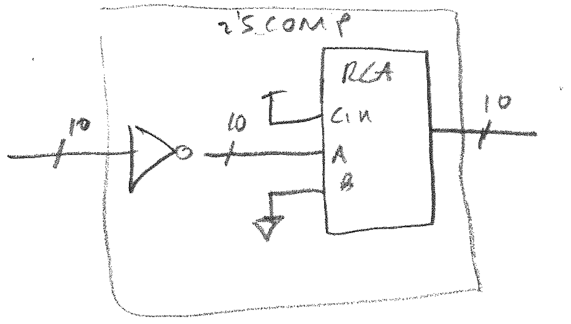
INTERNAL CONTROL



6)

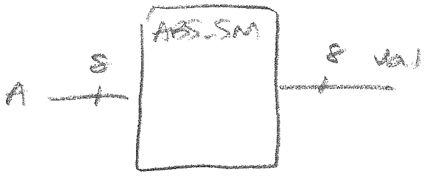


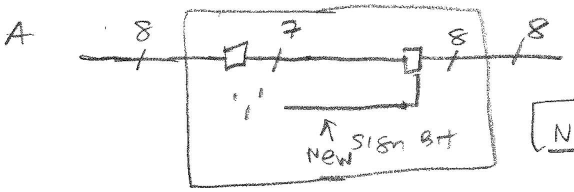




INTERNAL CONTROL

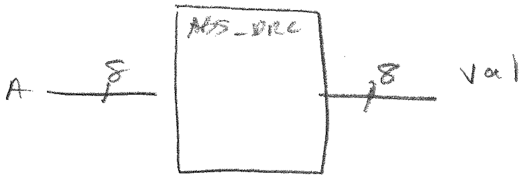
7)



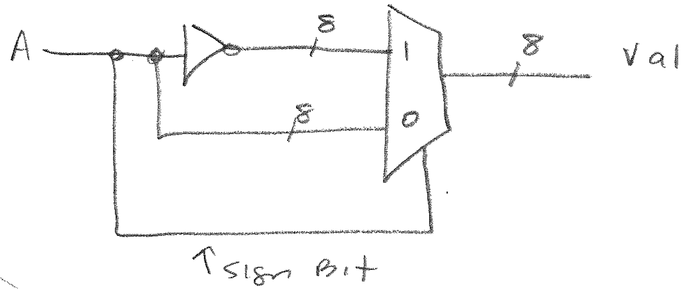


No control

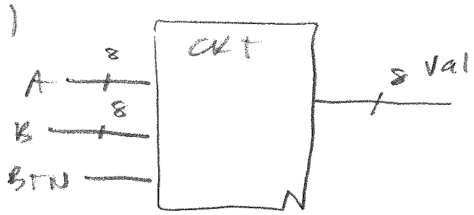
8)



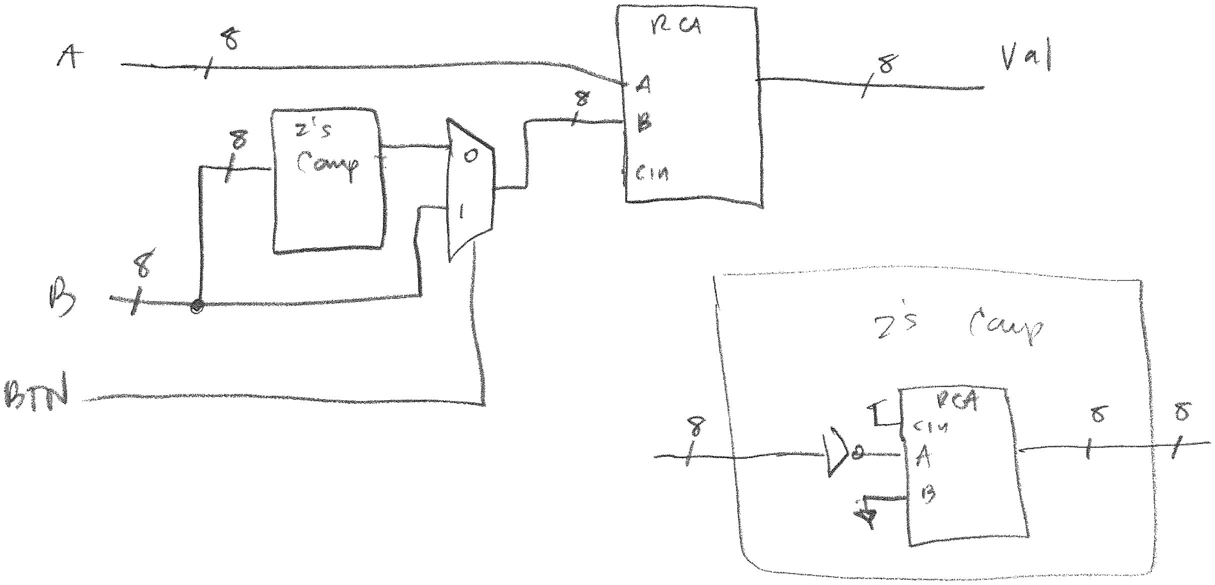
INTERNAL Control



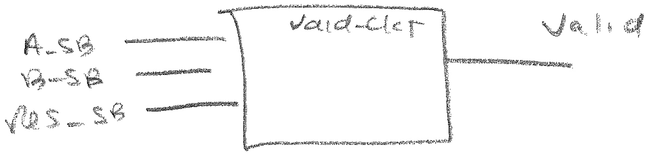
9)



External Control



10) output is Not valid when The SIGN Bit of The Two operands Are The Same But different Than The output of The Result.



No control

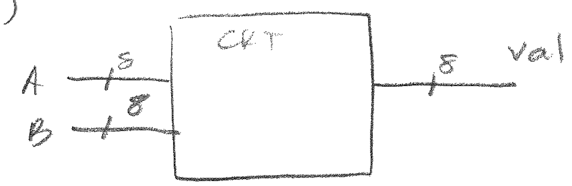
A_SB	B_SB	Res_SB	Valid
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

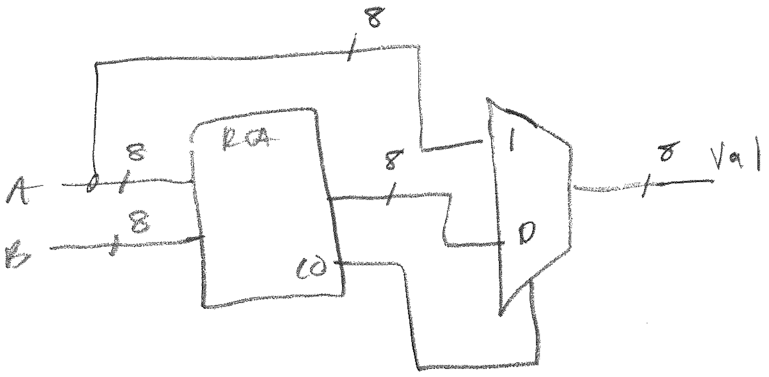
Great looking Decoder

$$\overline{Valid} = \overline{A\_SB} \cdot \overline{B\_SB} \cdot Res\_SB + A\_SB \cdot B\_SB \cdot \overline{Res\_SB}$$

$$valid = (A\_SB + B\_SB + \overline{Res\_SB})(\overline{A\_SB} + \overline{B\_SB} + Res\_SB)$$

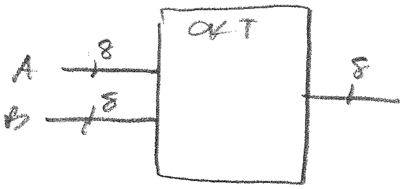
11)



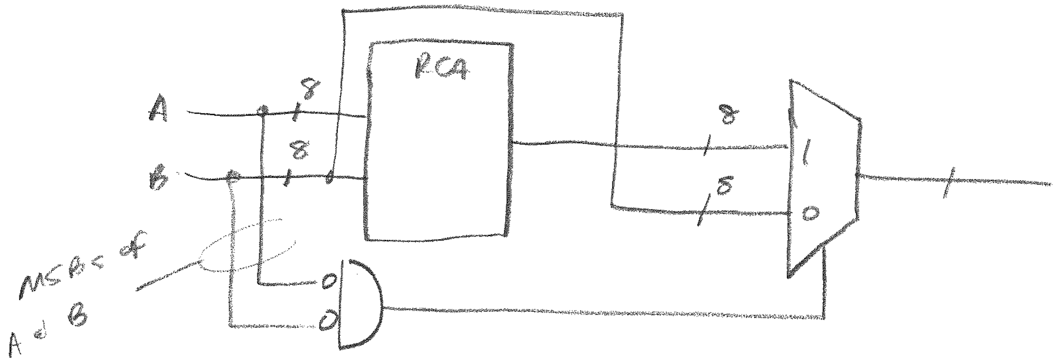


internal control

12)

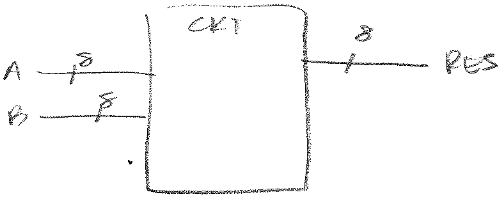


internal control

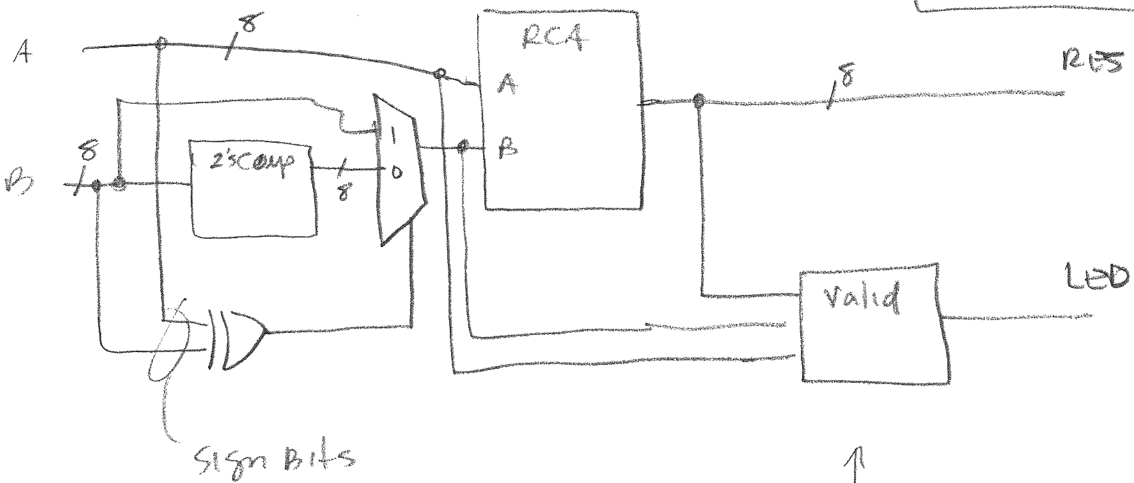


↳ IF MSB OF 8-bit UNSIGNED BINARY NUMBER IS SET, THE NUMBER IS GREATER THAN 127

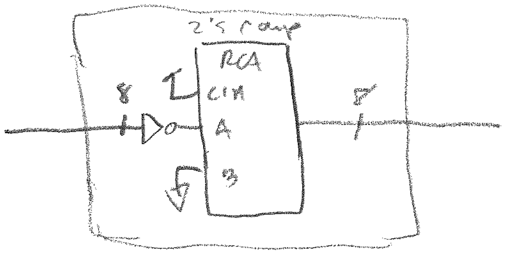
13)



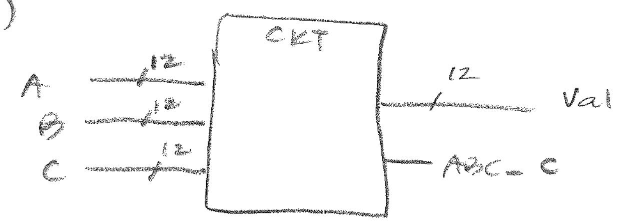
Internal Control



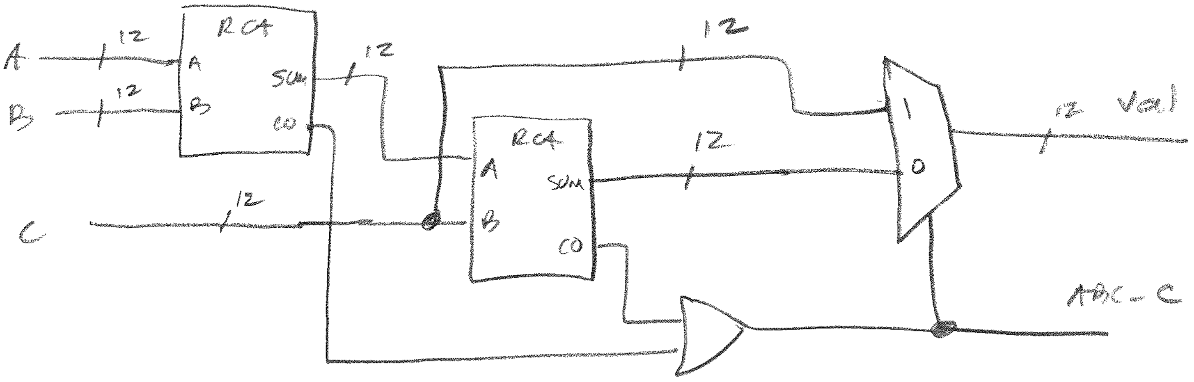
see problem 10



14)

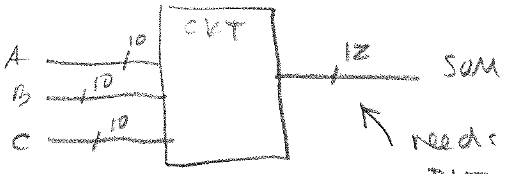


internal control



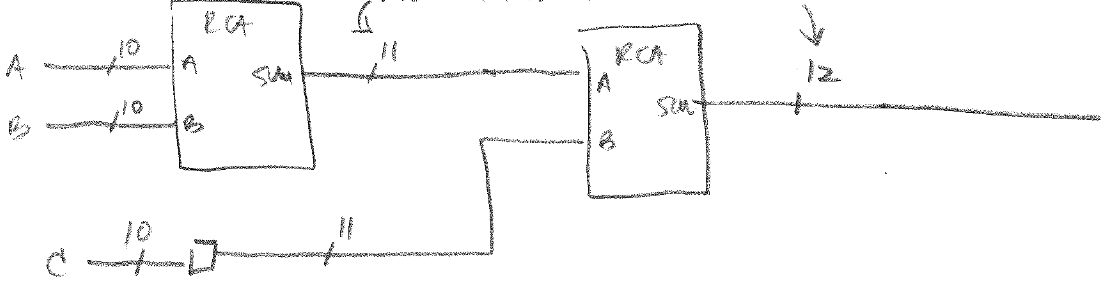
ABC_C = 0 => A+B+C
ABC_C = 1 => C

15)



needs to be 12 bits so THAT THE SUM will ALWAYS Be Valid

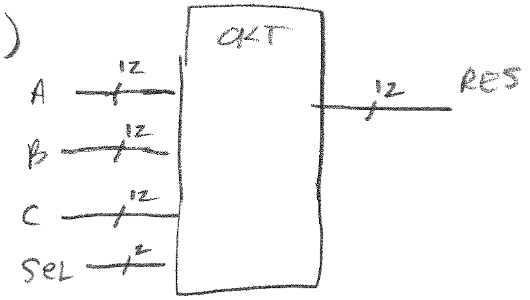
Needs extra bit for valid sum



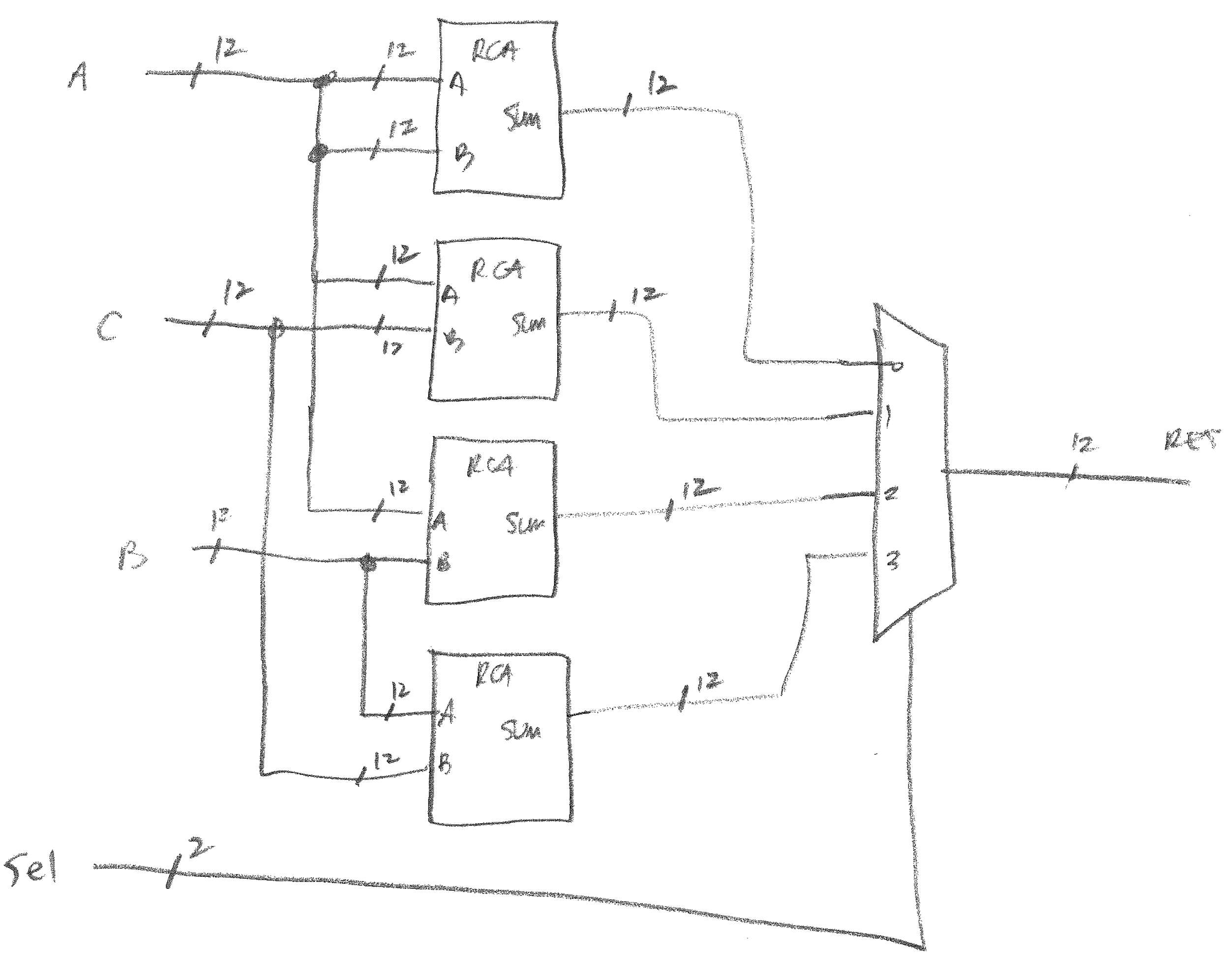
Append '0' to MSB

NO control

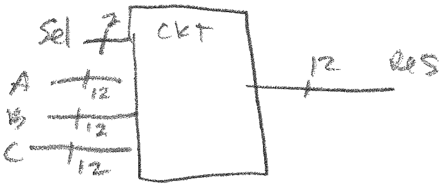
16)

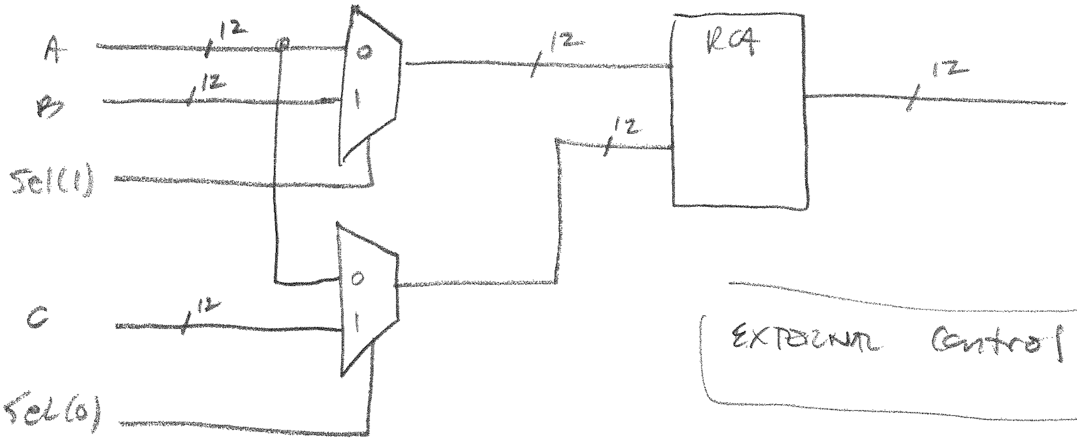


EXTERNAL Control



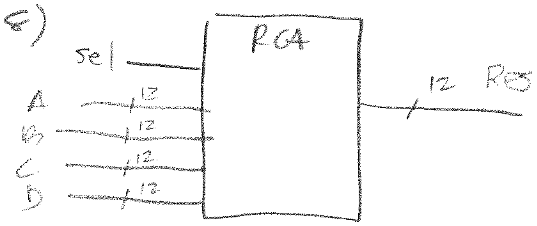
17)



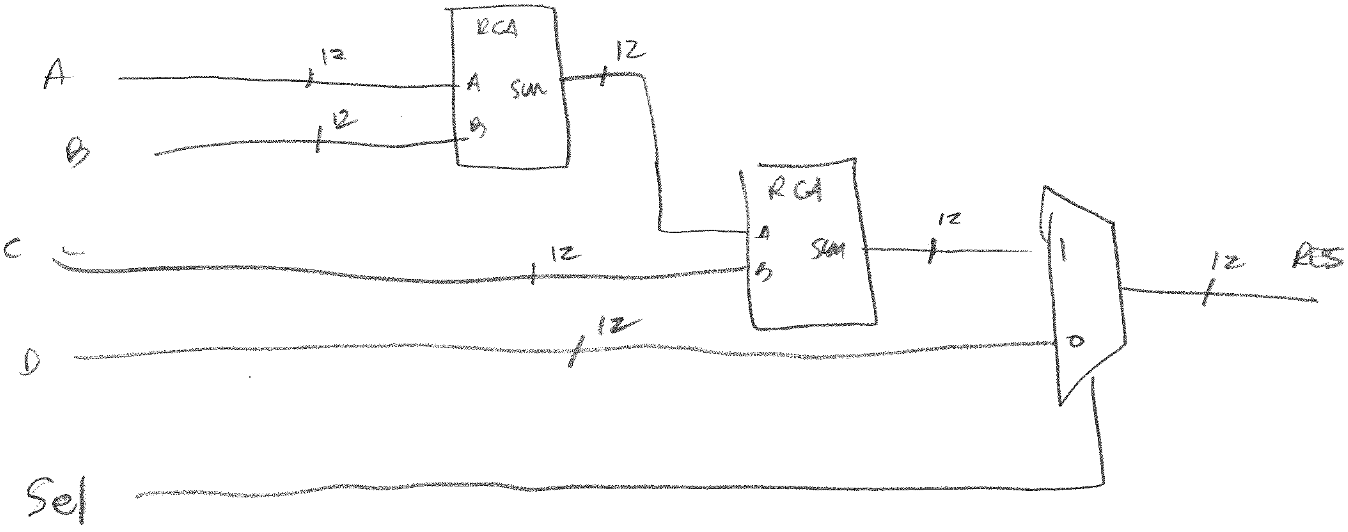


EXTERNAL control

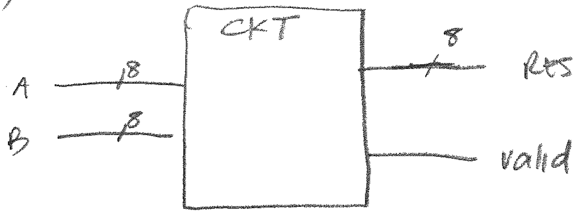
18)



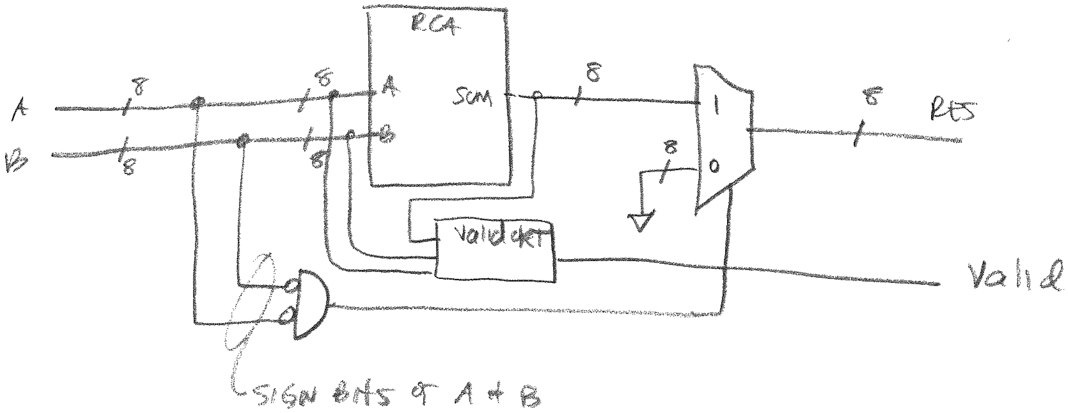
EXTERNAL control



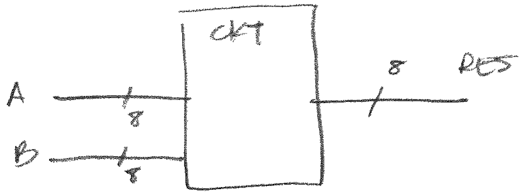
19)



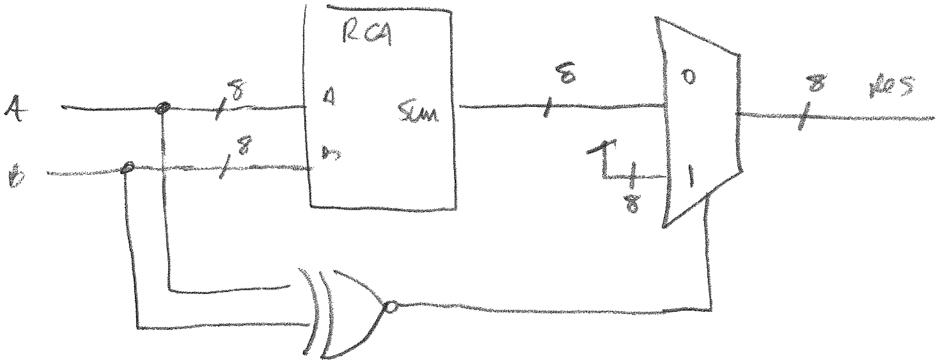
INTERNAL Control



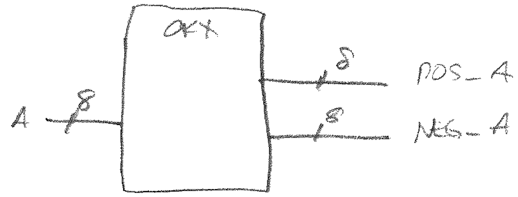
20)



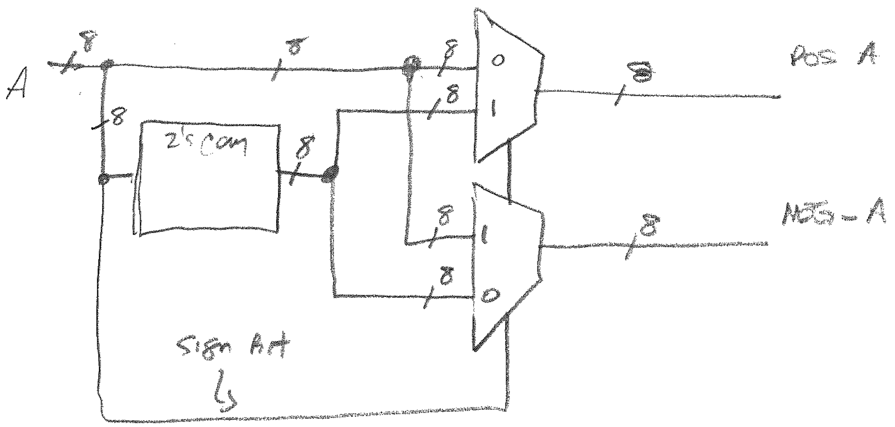
INTERNAL Control



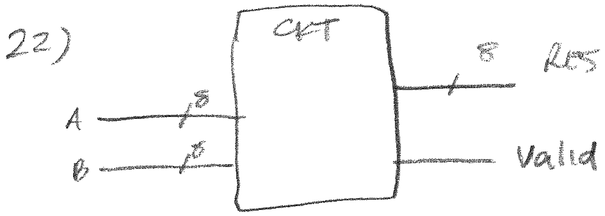
21)



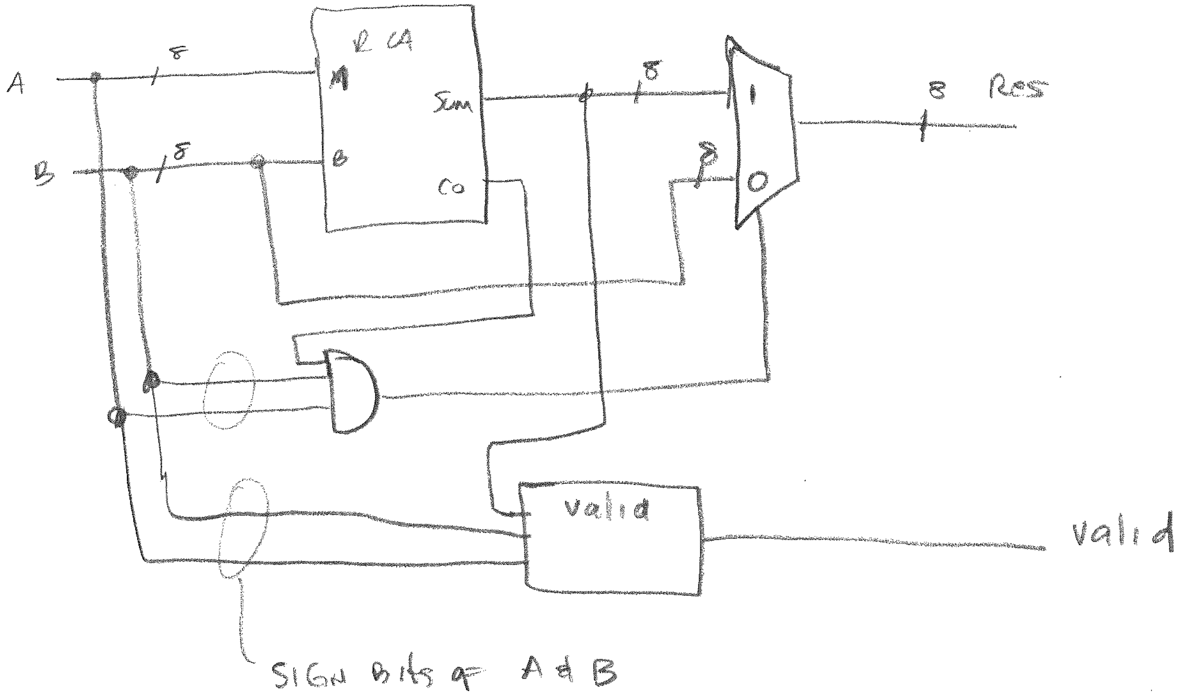
INTERNAL control



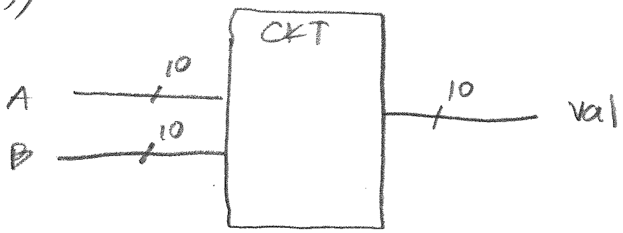
22)



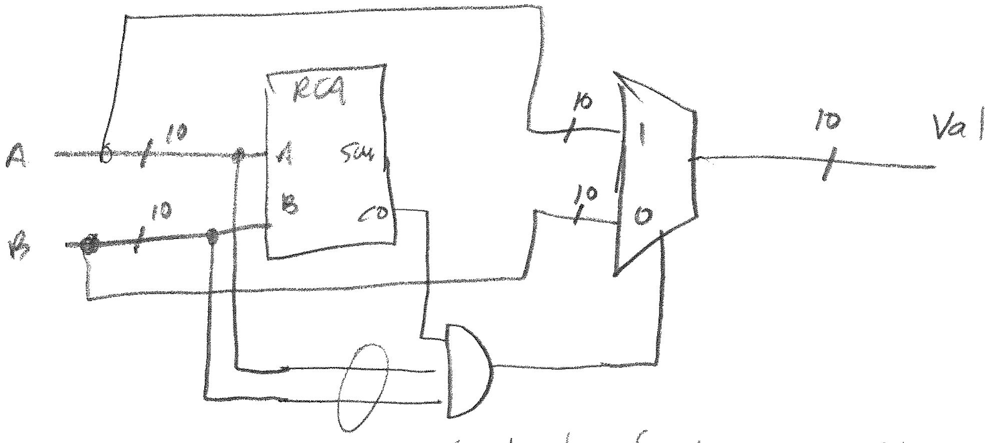
Internal Control



23)

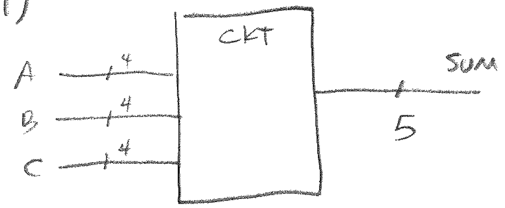


INTERNAL Control



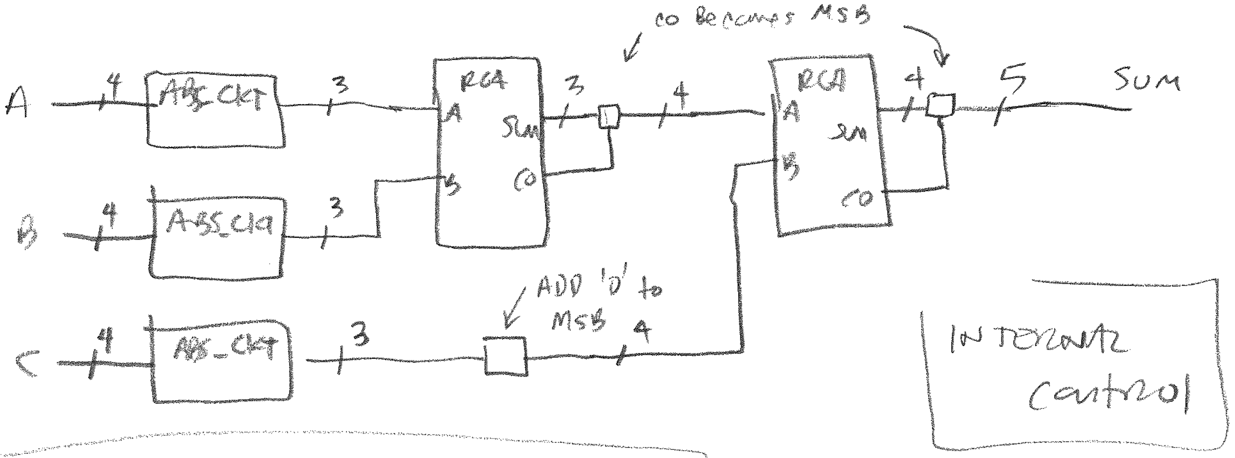
LSBs (indicate if values are odd or even

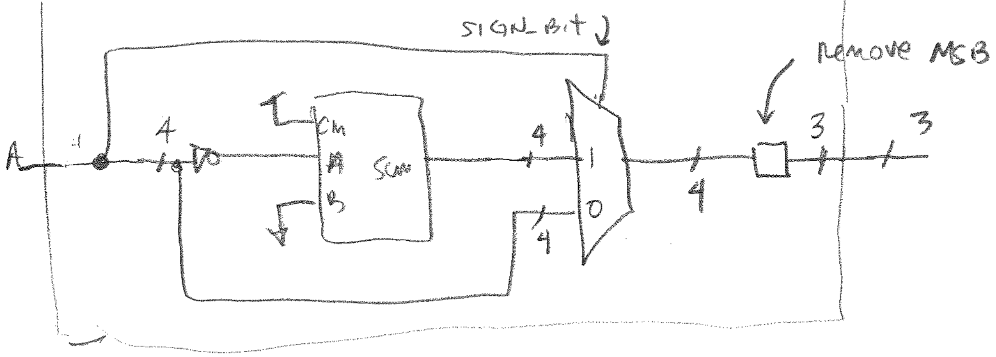
24)



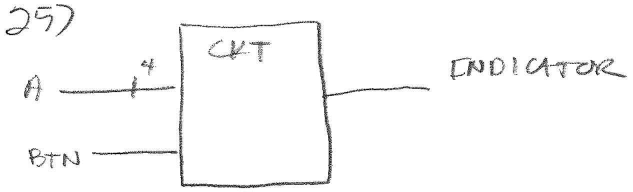
# of Bits on output = $\lceil \log_2 (7+7+7) \rceil$

$= \underline{\underline{5}}$ which is sufficient for UNSIGNED BINARY output

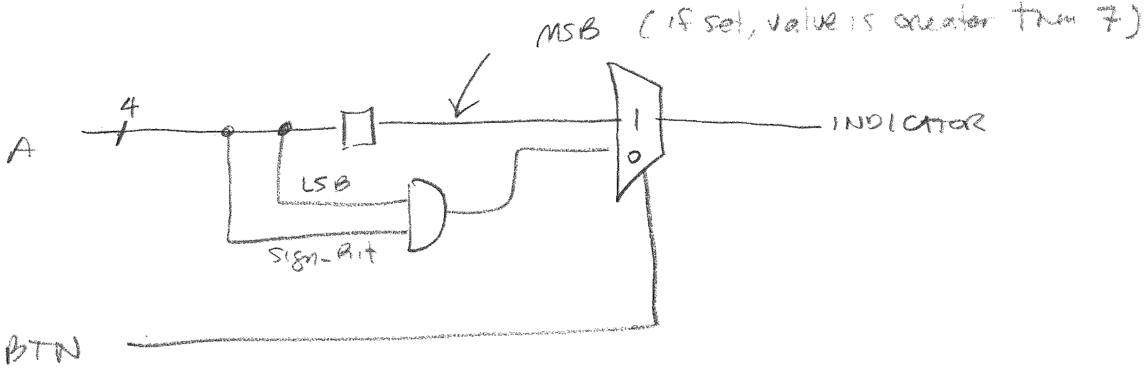




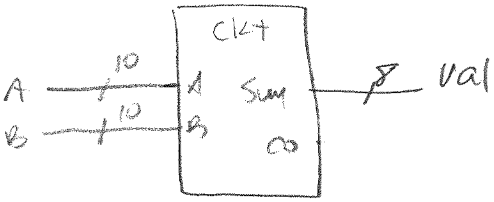
25)



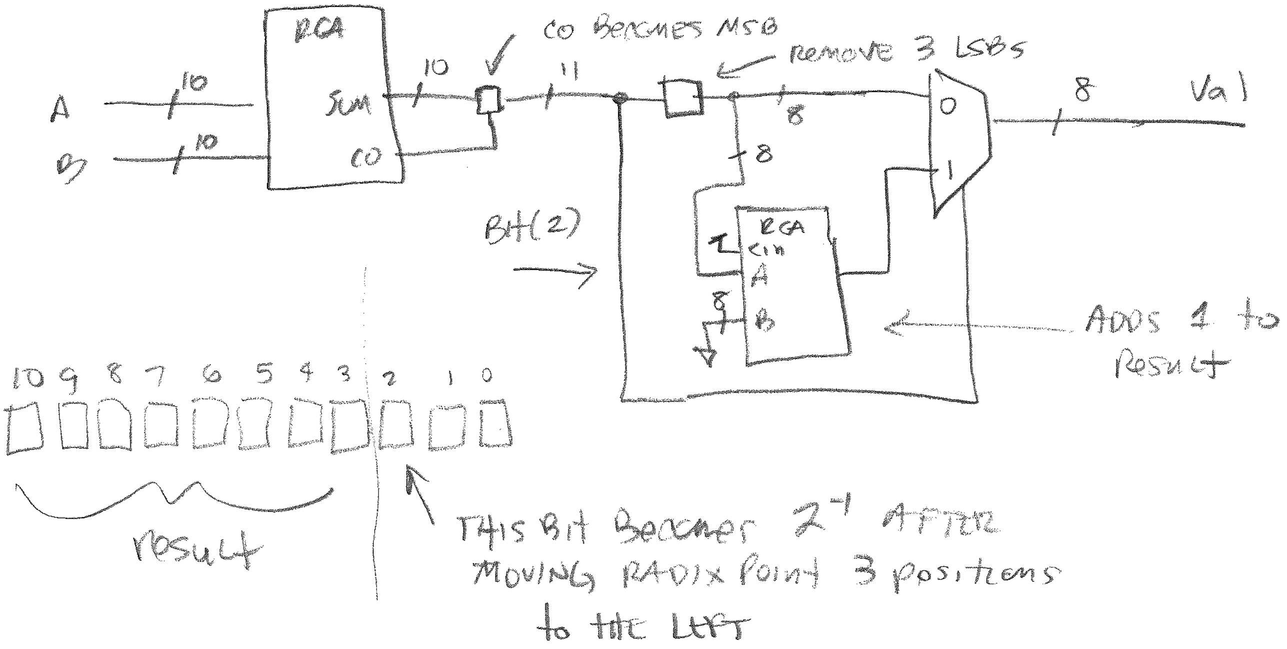
External Control



26)

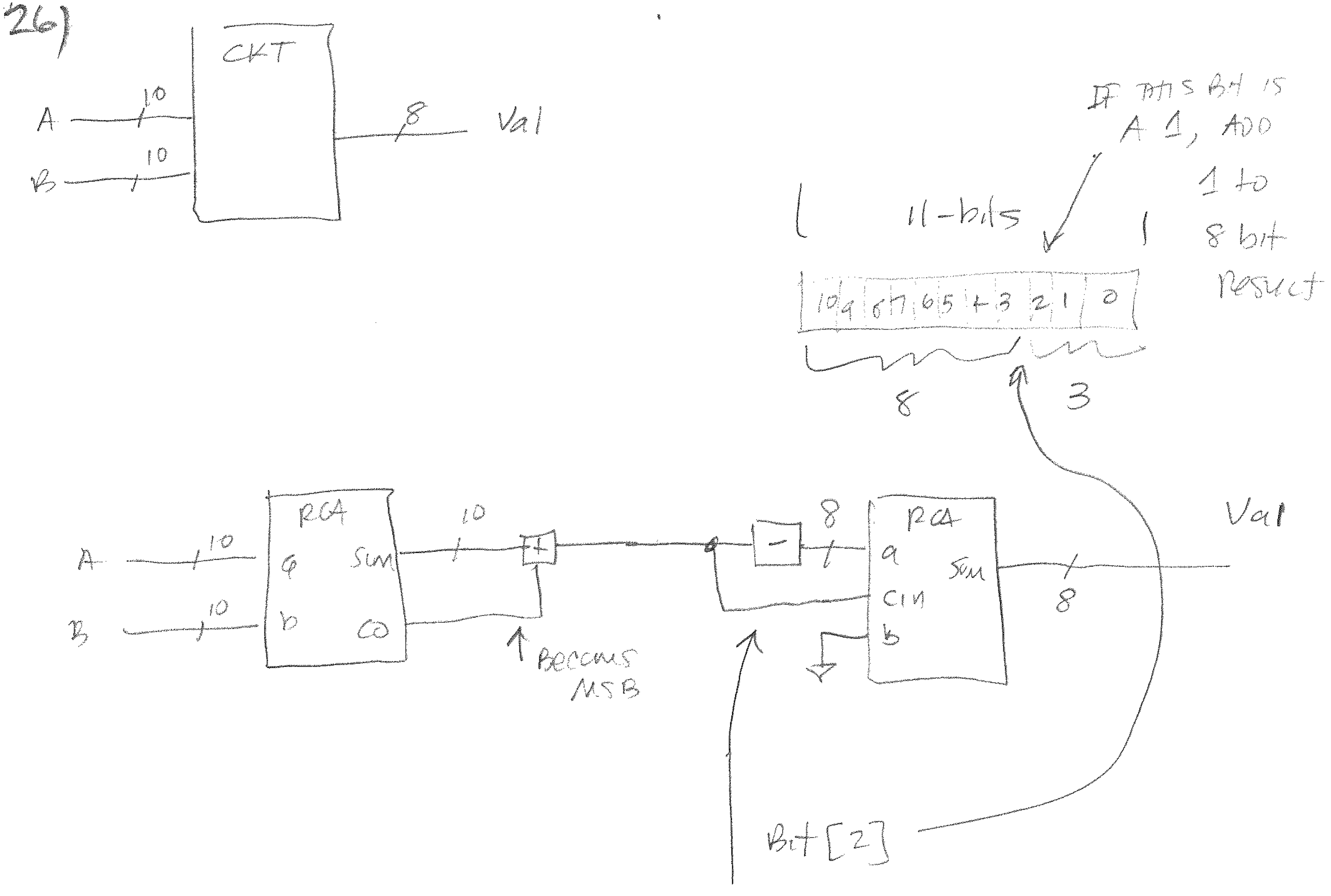


Internal control



ALTERNATIVE SOLUTION

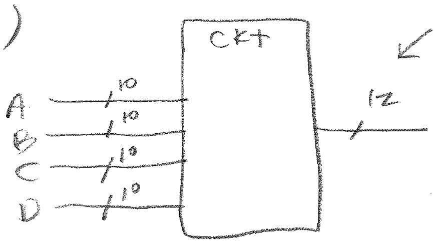
26)



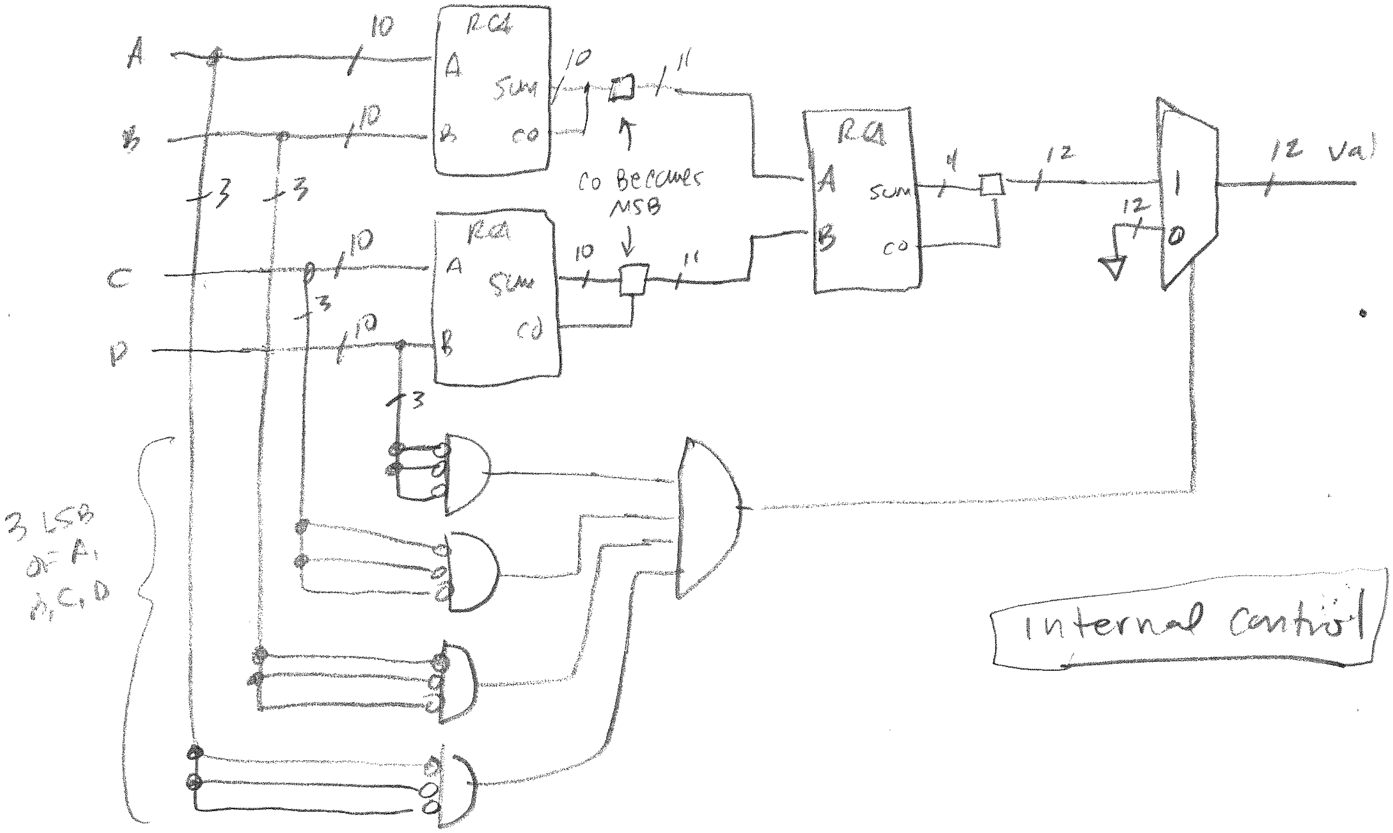
IF THIS BIT IS A 1, ADD 1 to 8 bit Result

No control

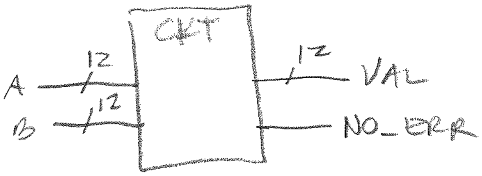
27)



RESULT MUST HAVE 12 bits to Be correct in worst CASE

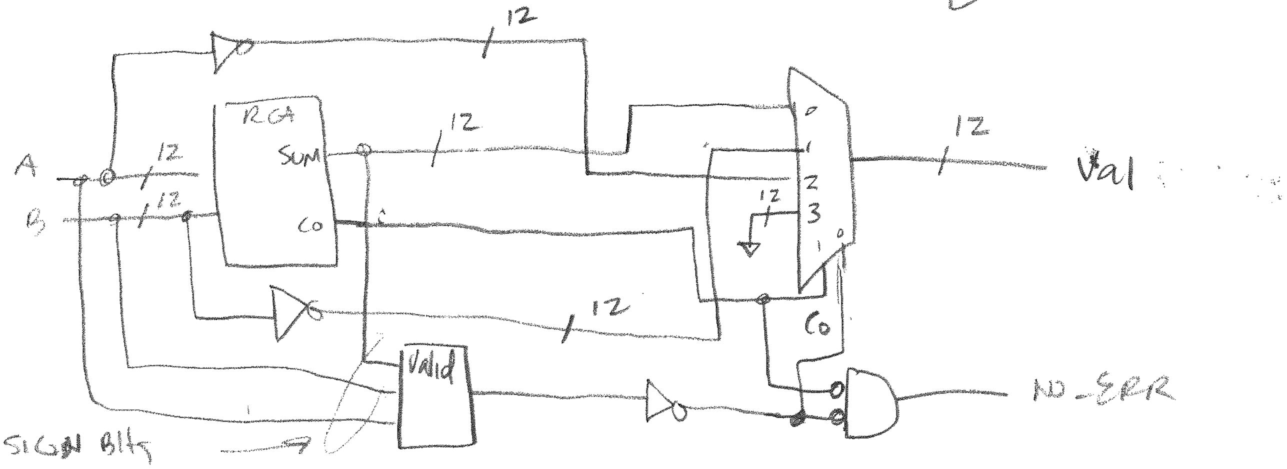


28)

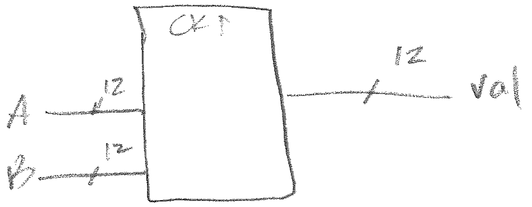


C	Valid	output
0	0	A+B
0	1	$\overline{B}$
1	0	$\overline{A}$
1	1	∅

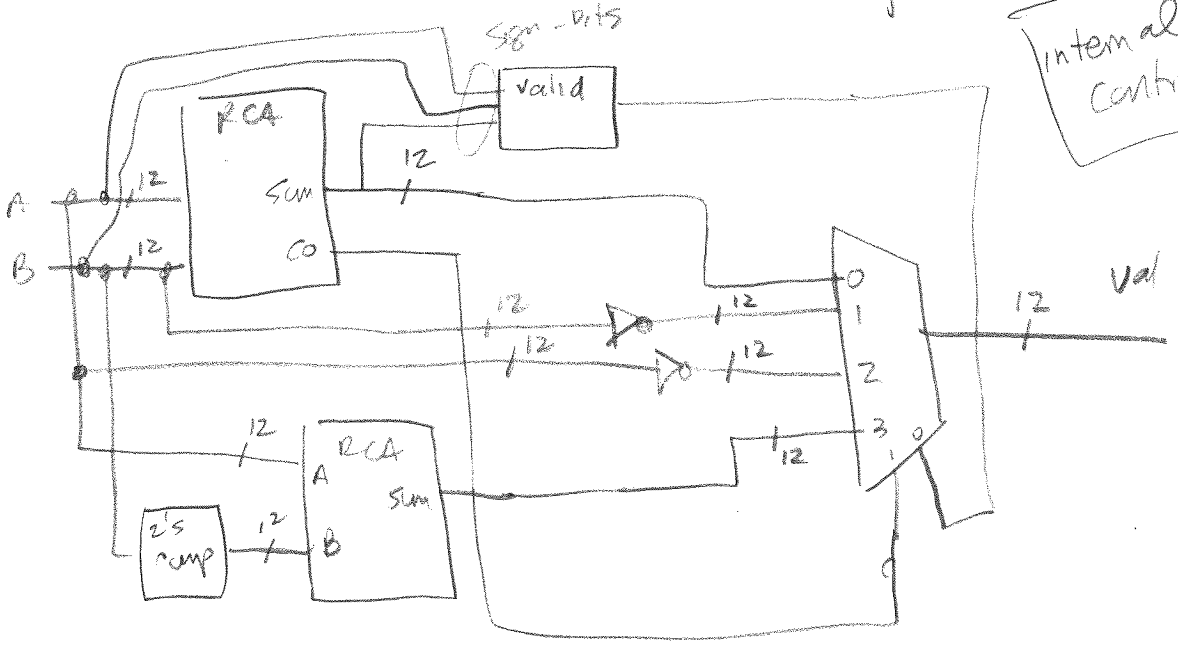
internal control



29)

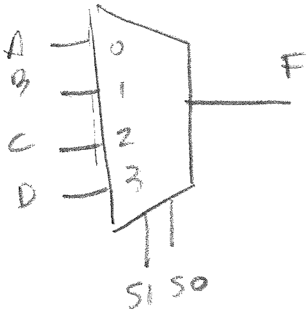


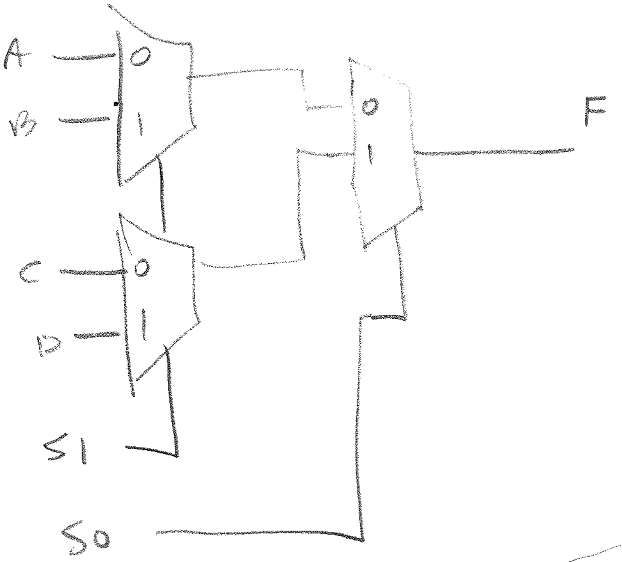
C	OF	operation
0	0	A+B
0	1	$\overline{B}$
1	0	$\overline{A}$
1	1	A - B



Internal control

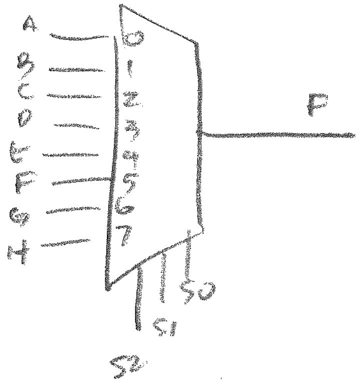
30)



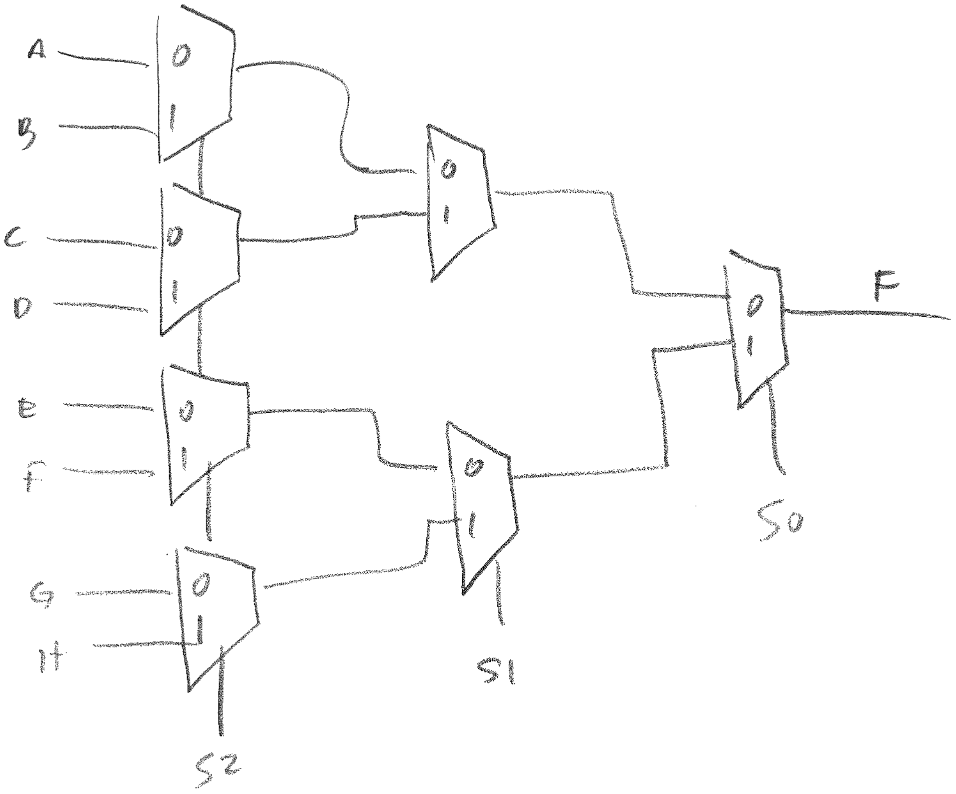


External control

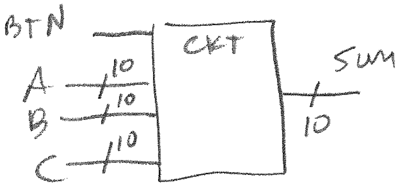
31)



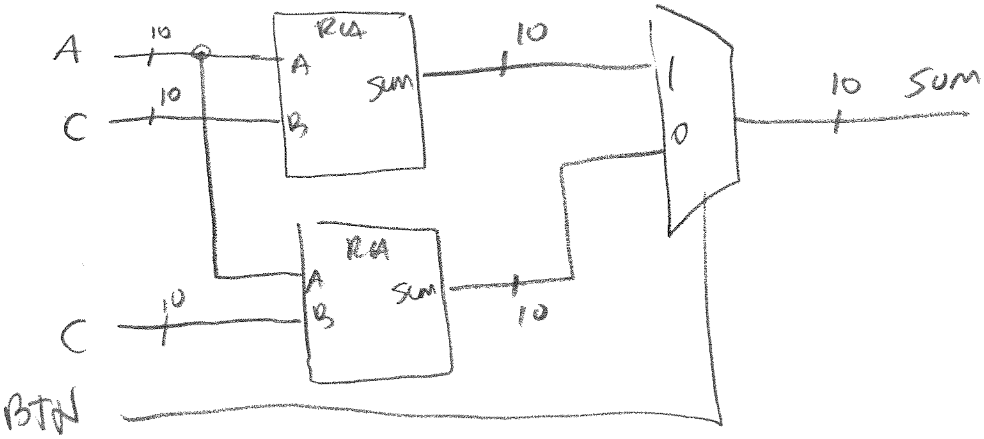
EXTERNAL Control



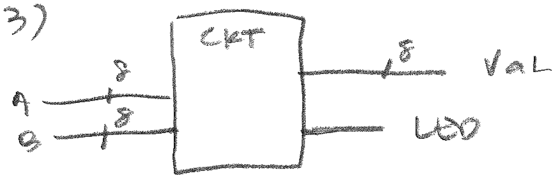
32)



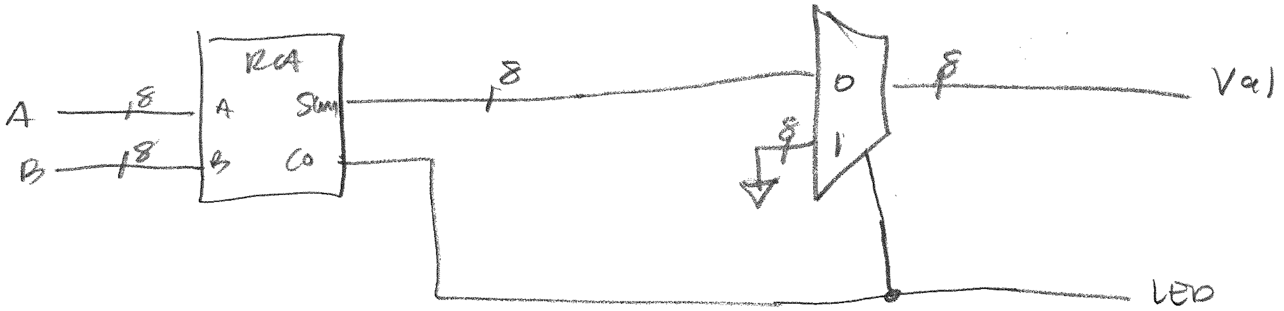
EXTERNAL Control



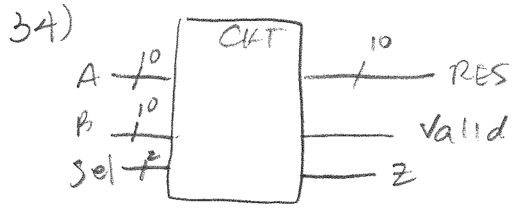
33)



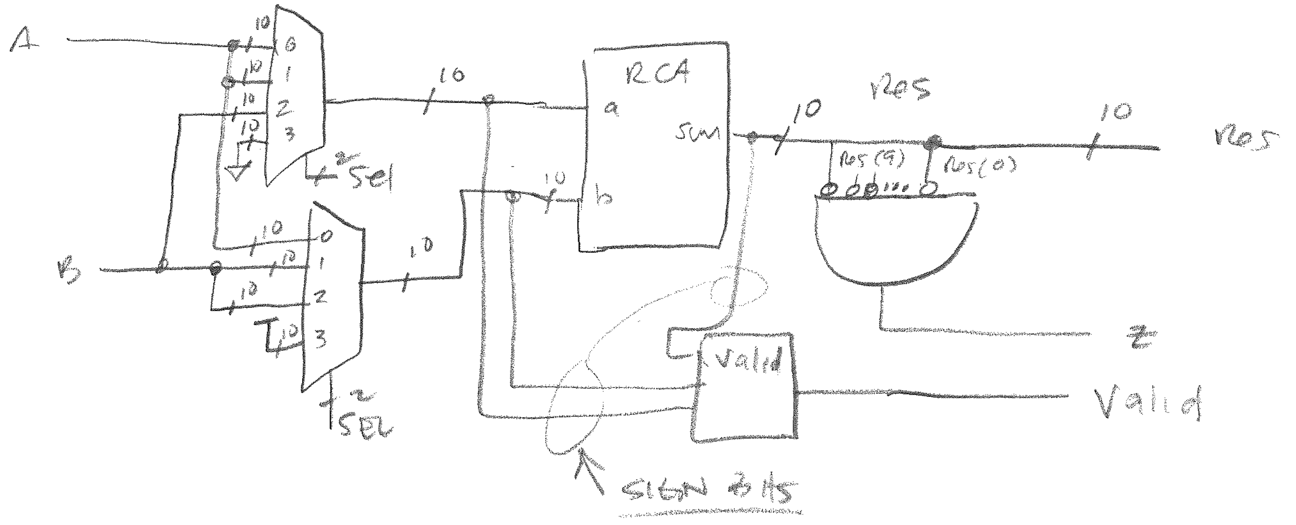
Internal control



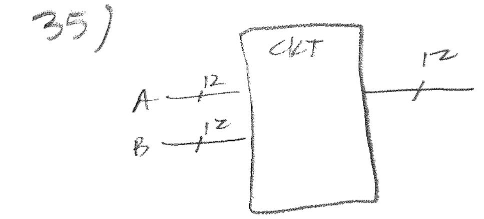
34)



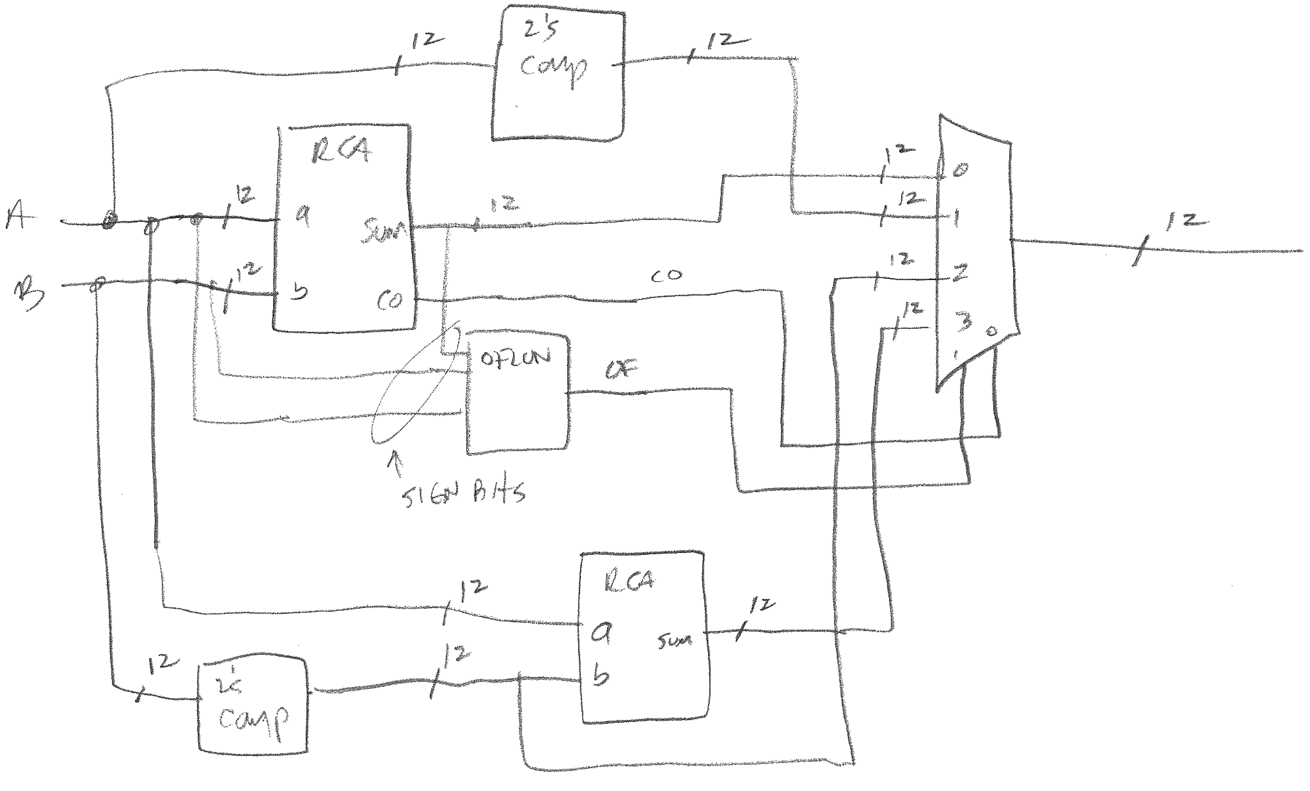
External controlled



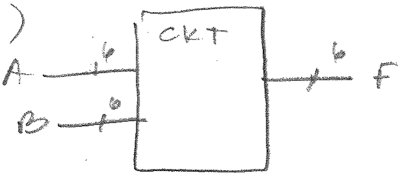
35)



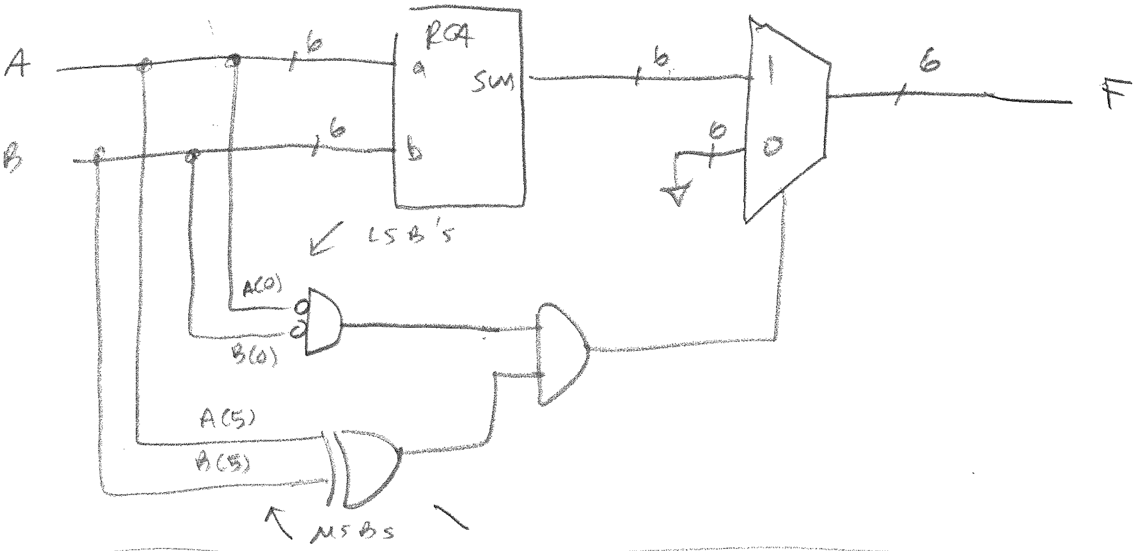
Internal controlled



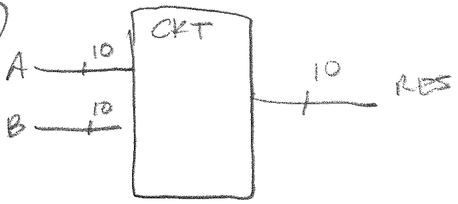
36)



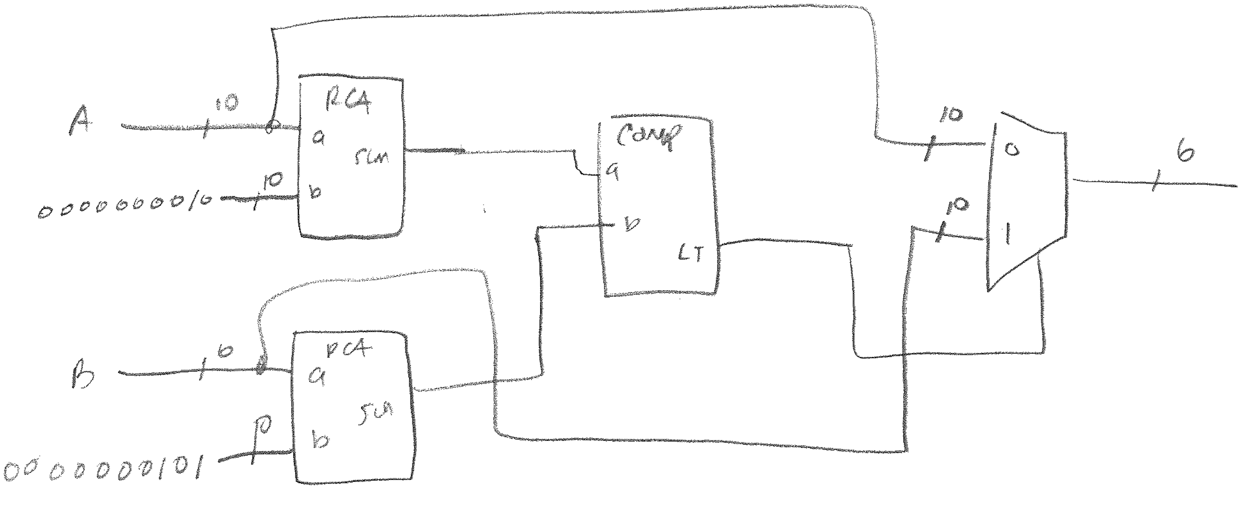
internal control



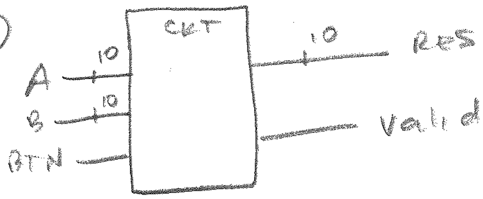
37)

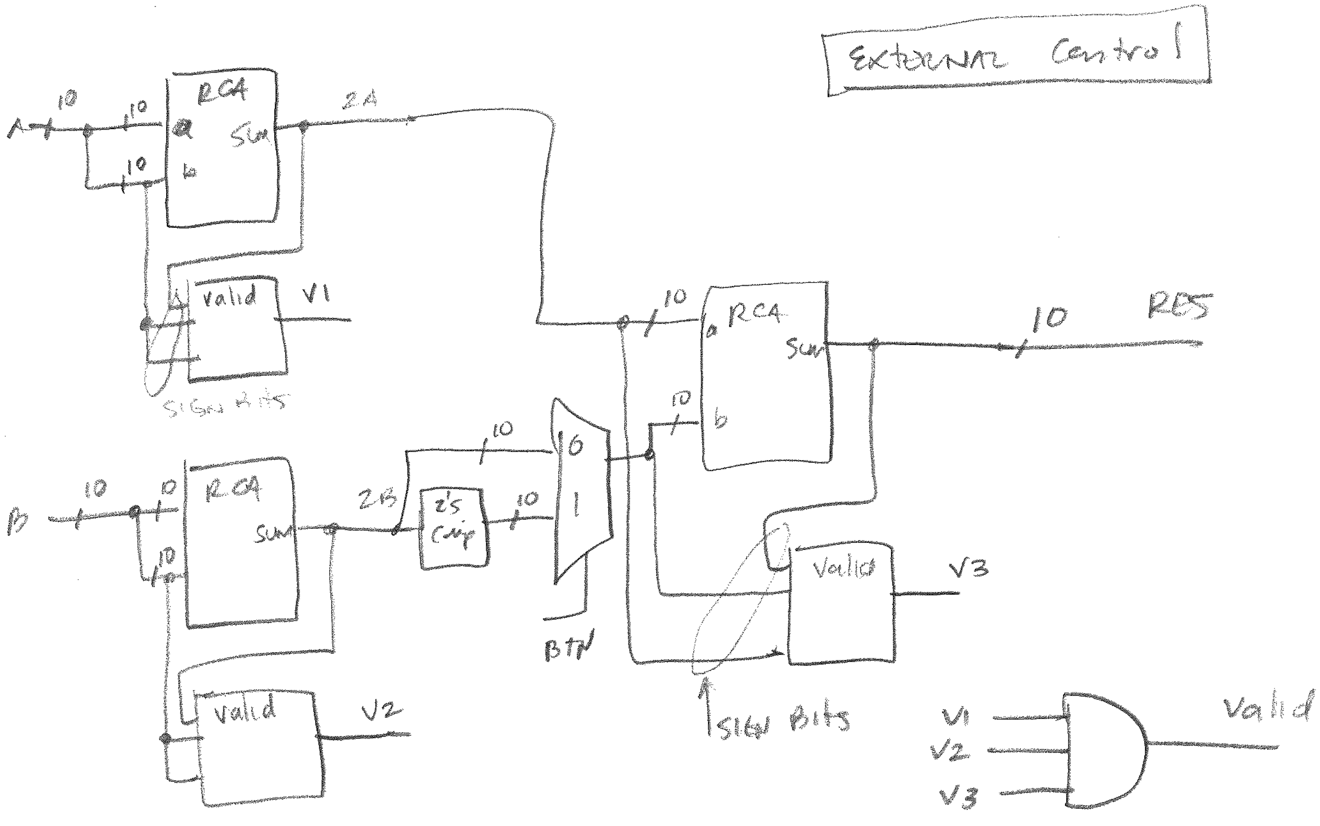


internal control

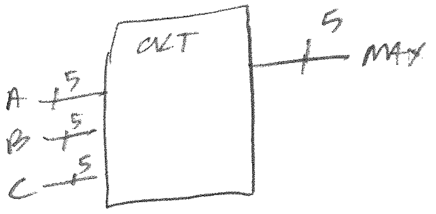


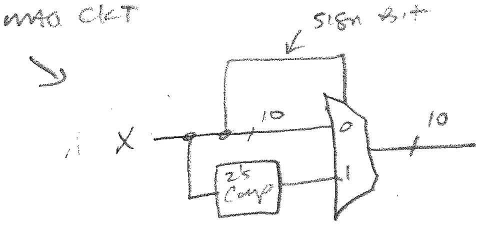
38)

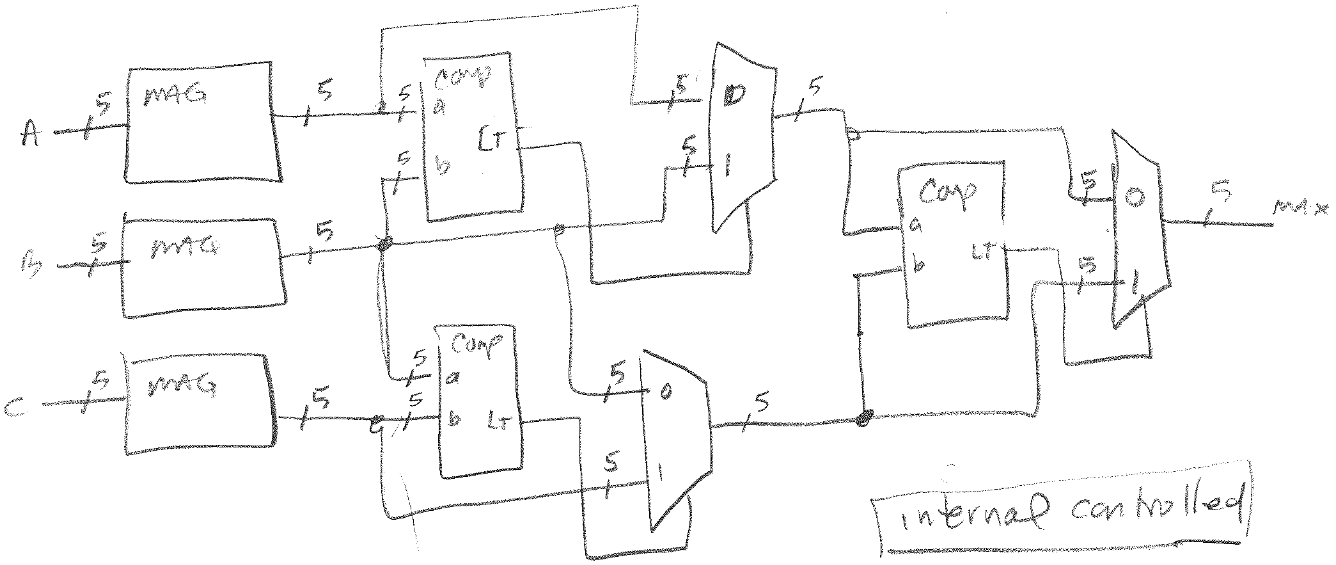

CKT
A
10
B
10
BTN
10
RES
valid


External Control
A
10
10
10
RCA
a
b
Sum
2A
Valid
V1
SIGN BITS
B
10
10
10
RCA
Sum
2B
2's Cmp
10
10
0
1
BTN
Valid
V2
10
10
RCA
a
b
Sum
10
RES
Valid
V3
SIGN BITS
V1
V2
V3
Valid

39)

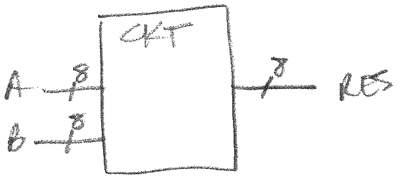




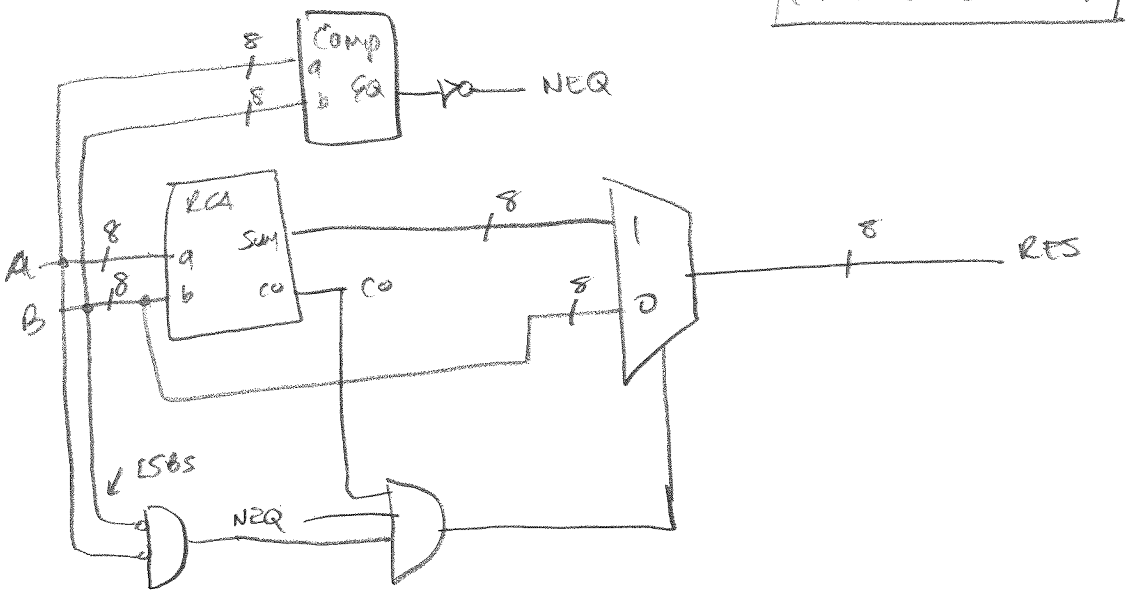


internal controlled

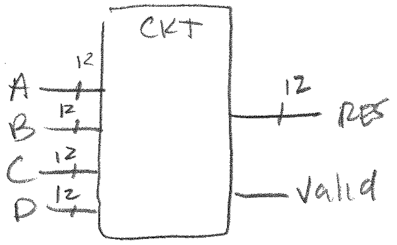
40)



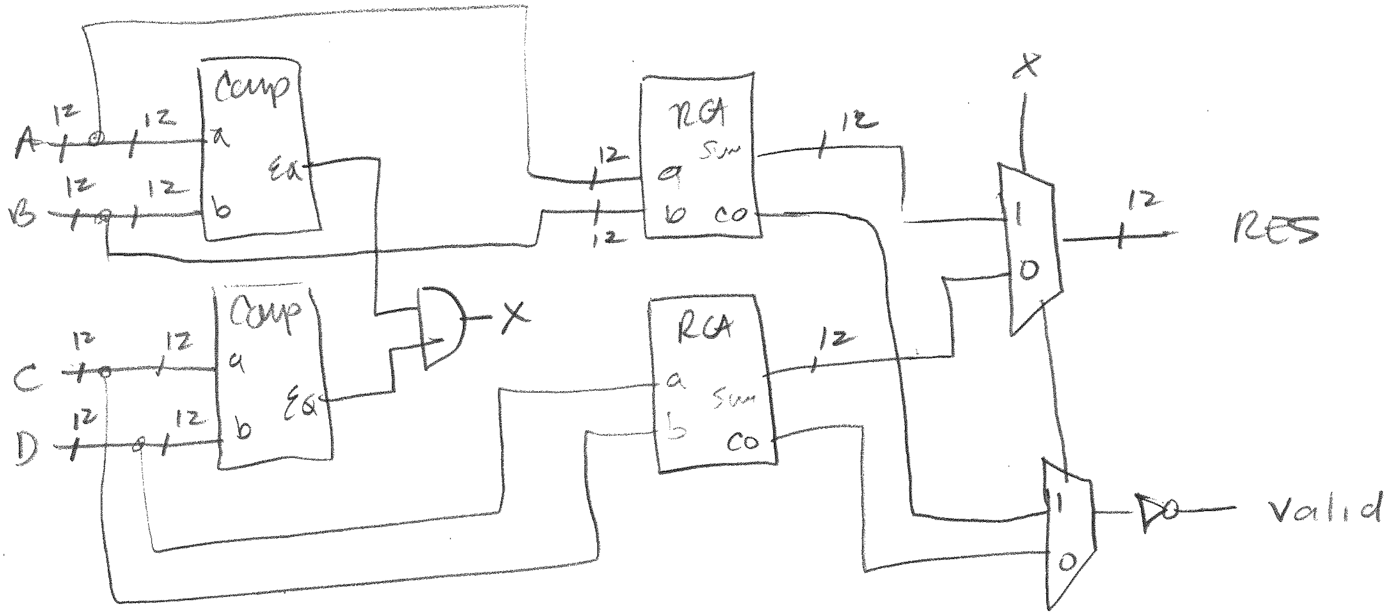
internal control



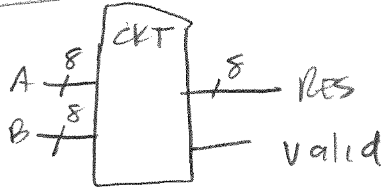
41)



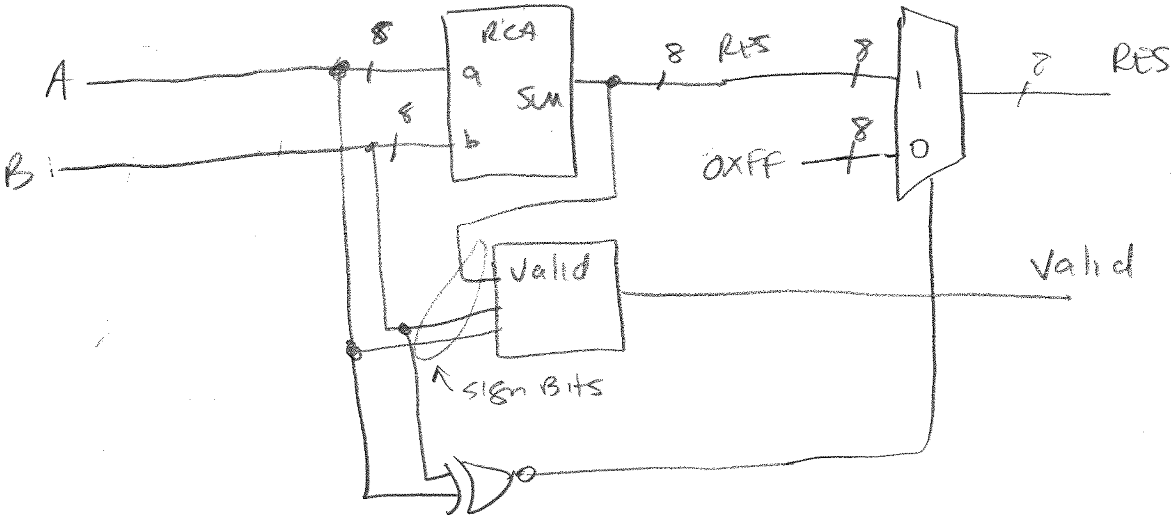
INTERNAL CONTROLLED



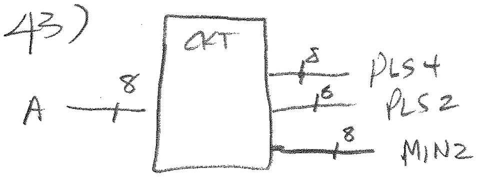
42)



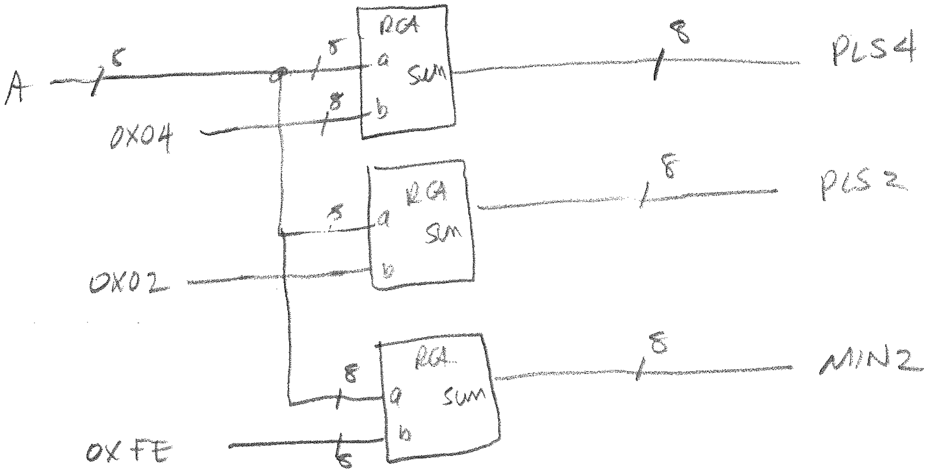
INTERNAL Controlled



43)

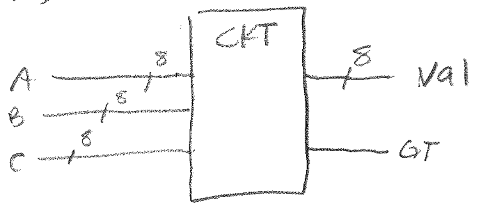


0000 0010 0x02
1111 1110 0xFE

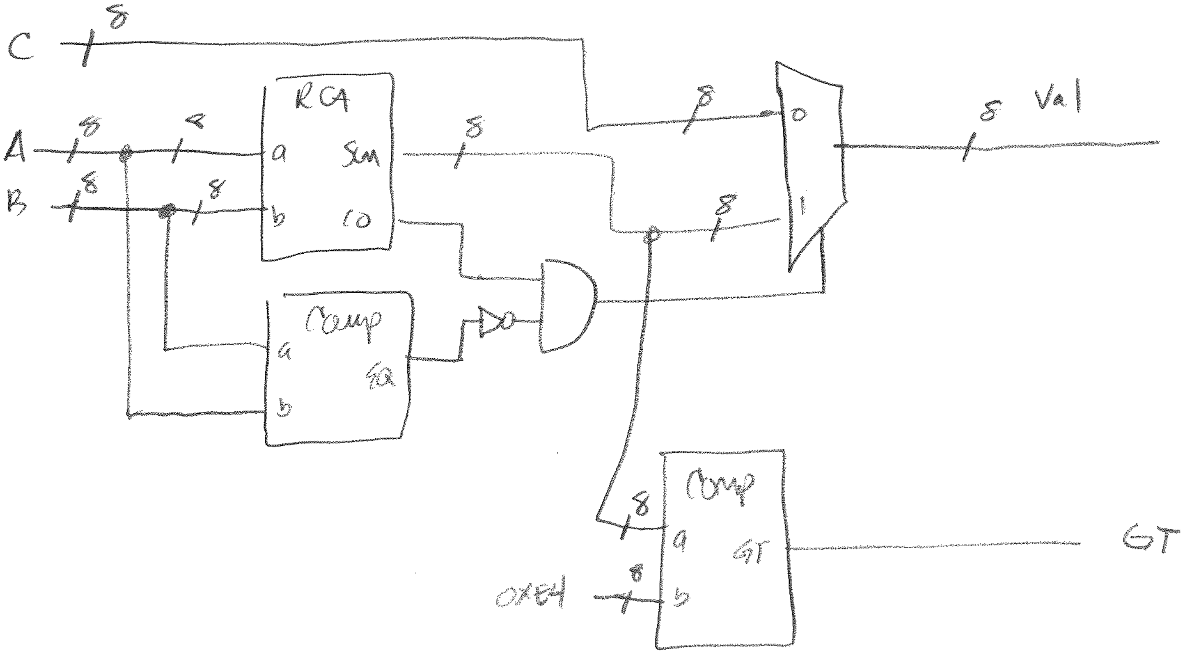


NotControlled

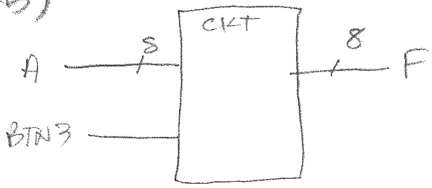
44)



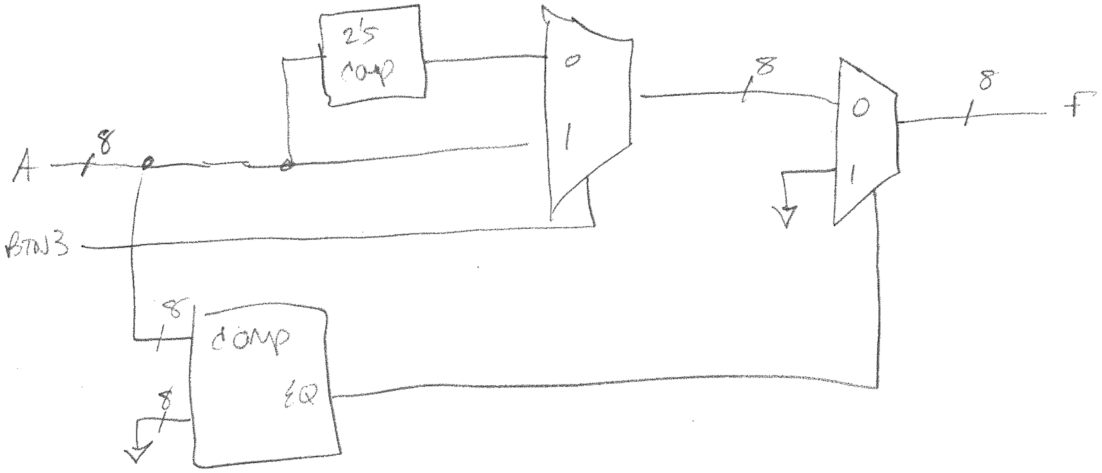
INTERNAL CONTROL



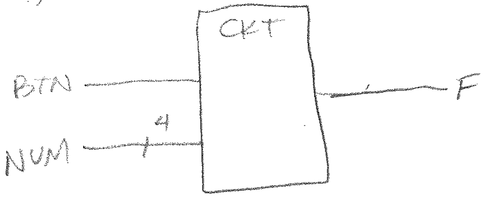
45)



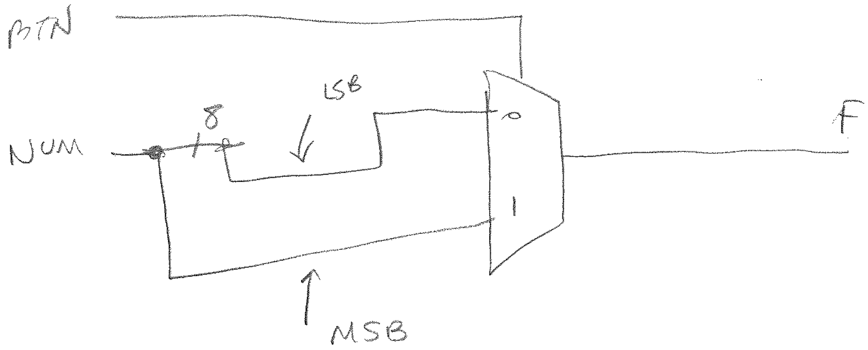
EXTERNAL &
INTERNAL controlled



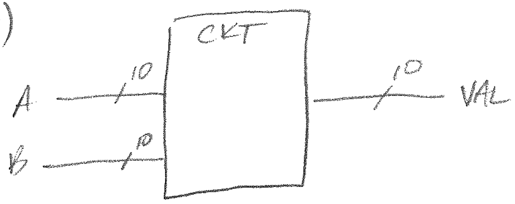
46)



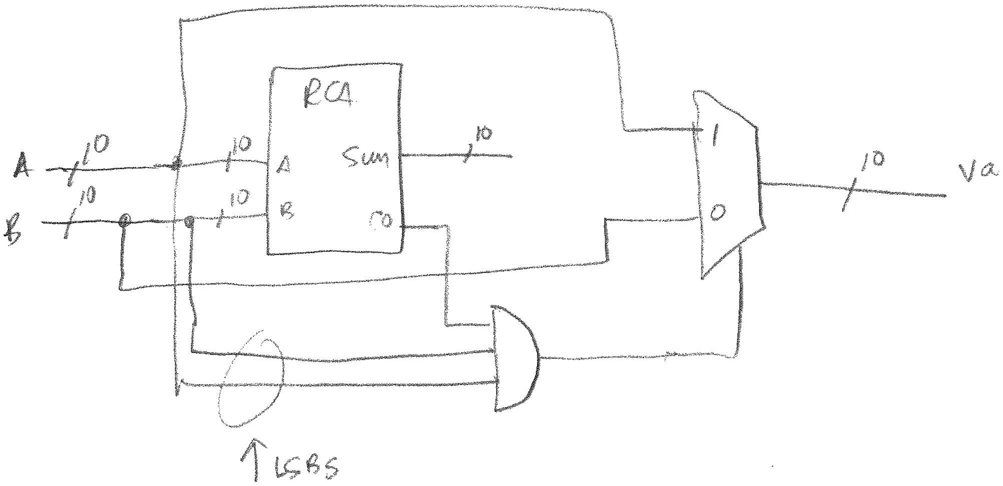
EXTERNAL CONTROL



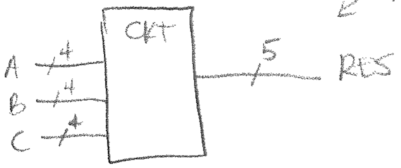
47)



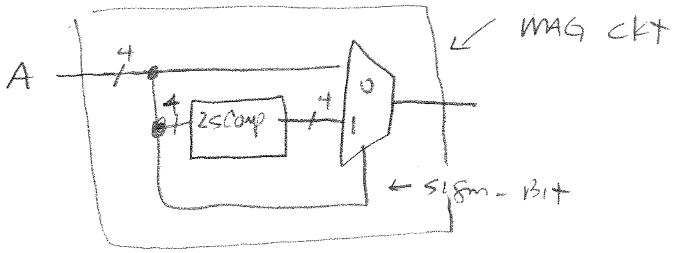
INTERNAL CONTROL



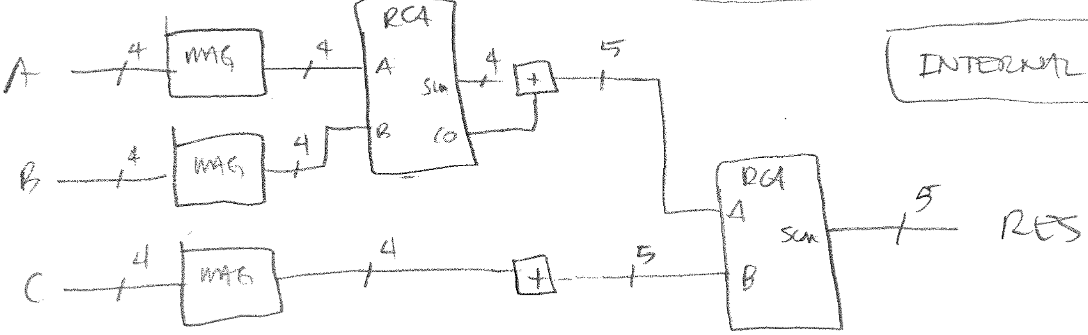
48)



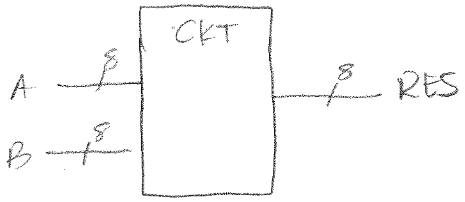
← NEEDS 5 Bit for max Val = 8 + 8 + 8 = 24



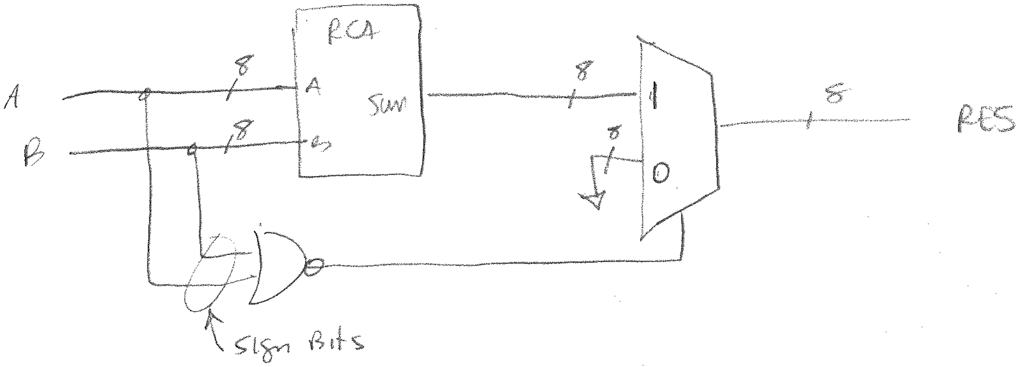
INTERNAL control



49)



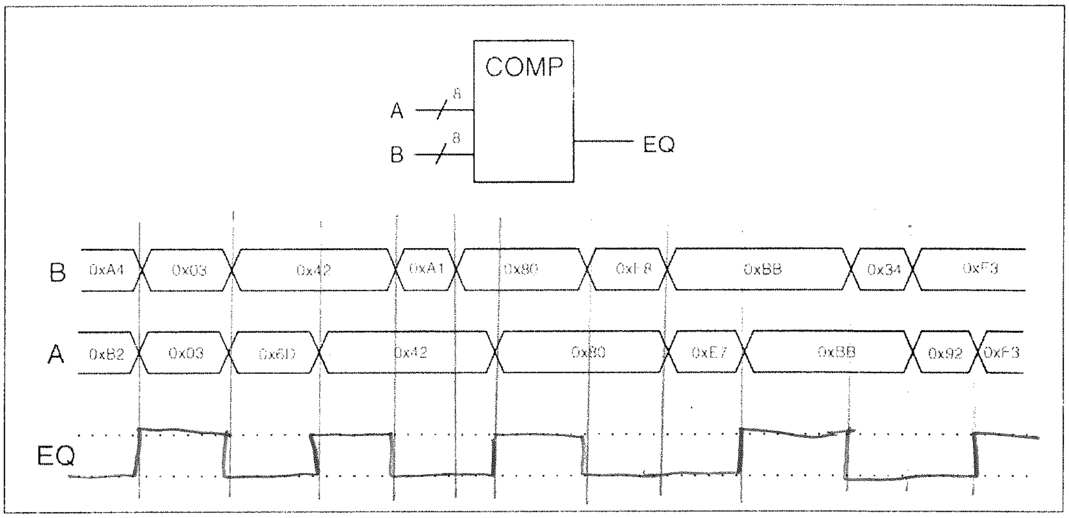
INTORNAL control



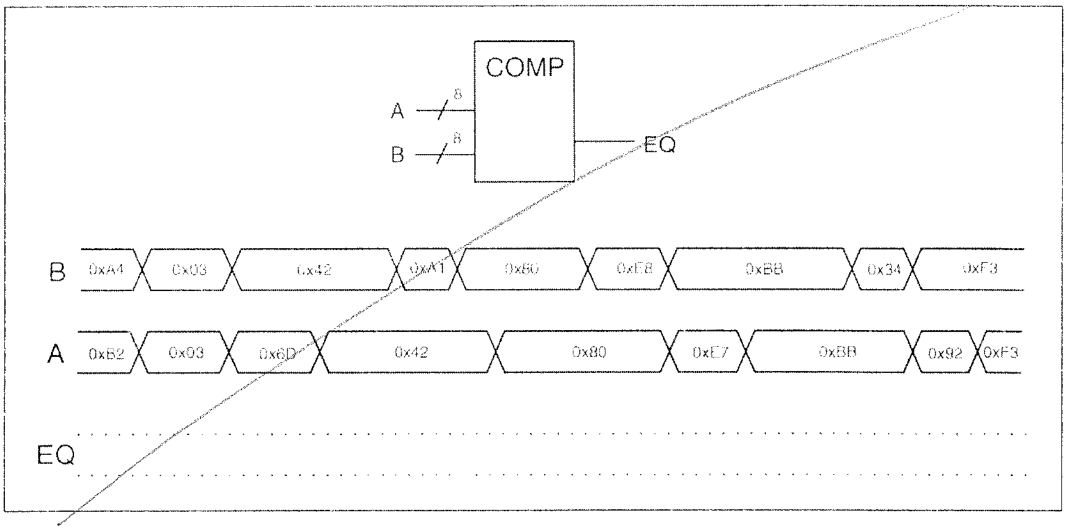
Chapter Exercises

chapter 18

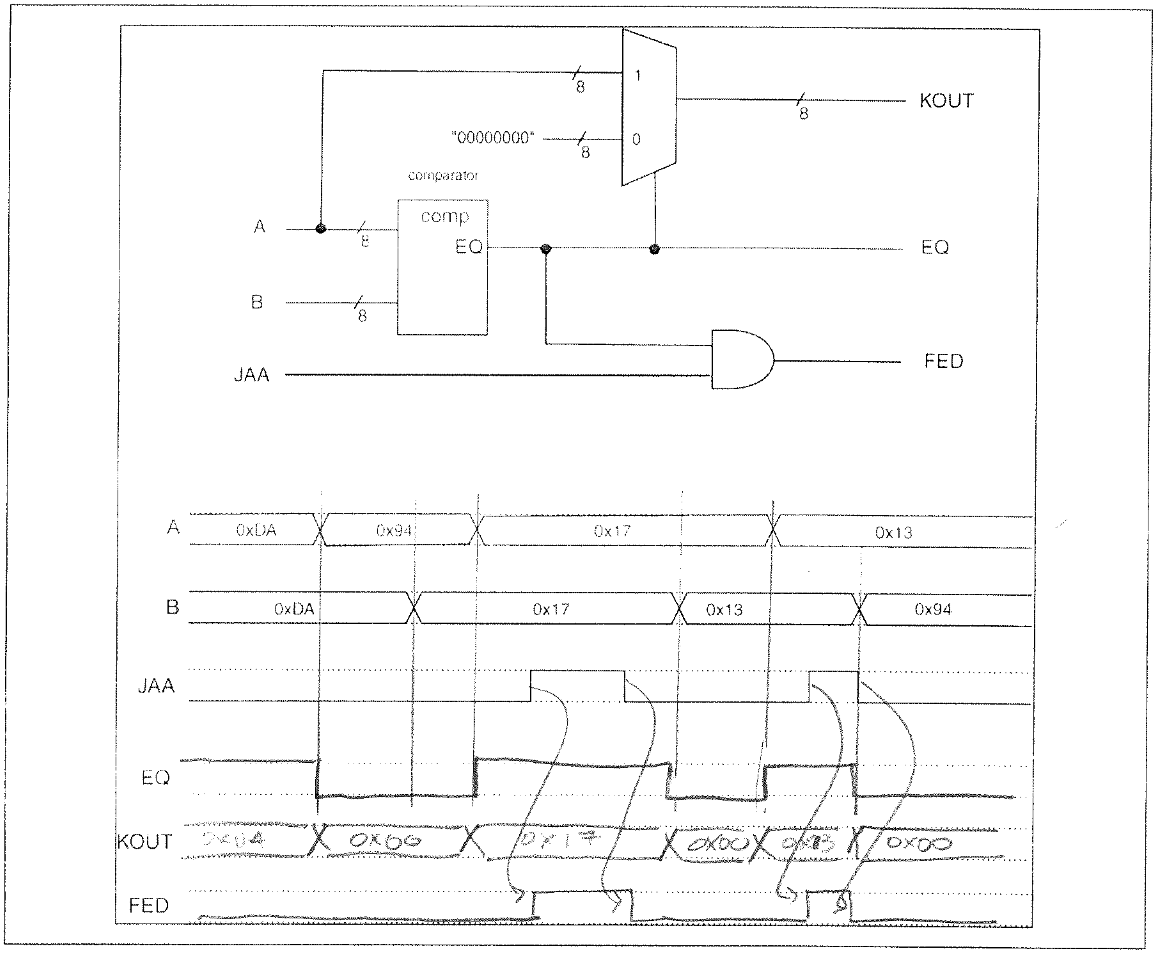
1) Complete the timing diagram shown below considering the given schematic symbol.



2) Complete the timing diagram shown below considering the given schematic symbol

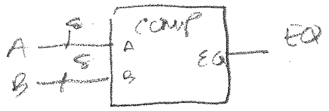


2) Use the following circuit to complete the unlisted signals in the timing diagram. For this problem, assume there are no propagation delays.

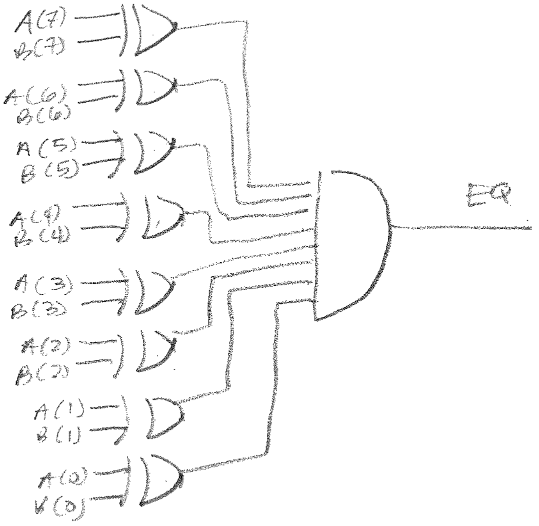


CHAPTER 18 DESIGN PROBLEM SOLUTIONS

1)

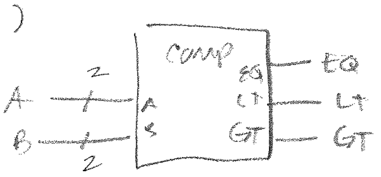


A



No control

2)

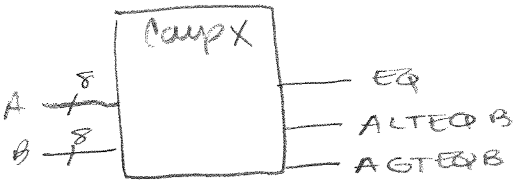


A1	A0	B1	B0	EQ	LT	GT
0	0	0	0	1	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	1	0	0
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	1	0	0
1	0	1	1	0	1	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	1	0	0

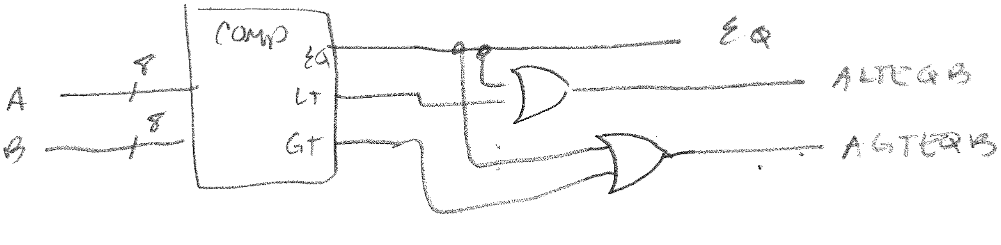
↙ This is sufficient to define a decoder

No control

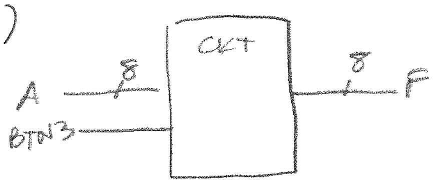
3)



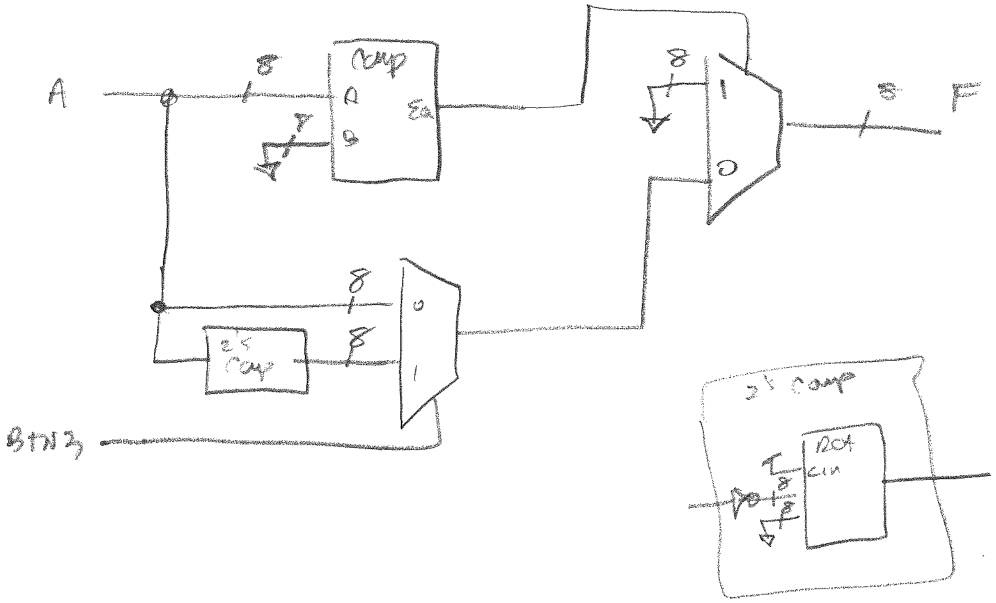
No control



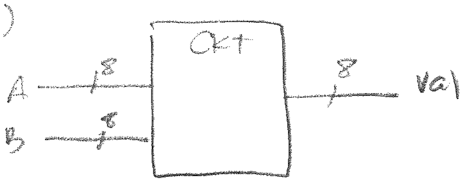
4)



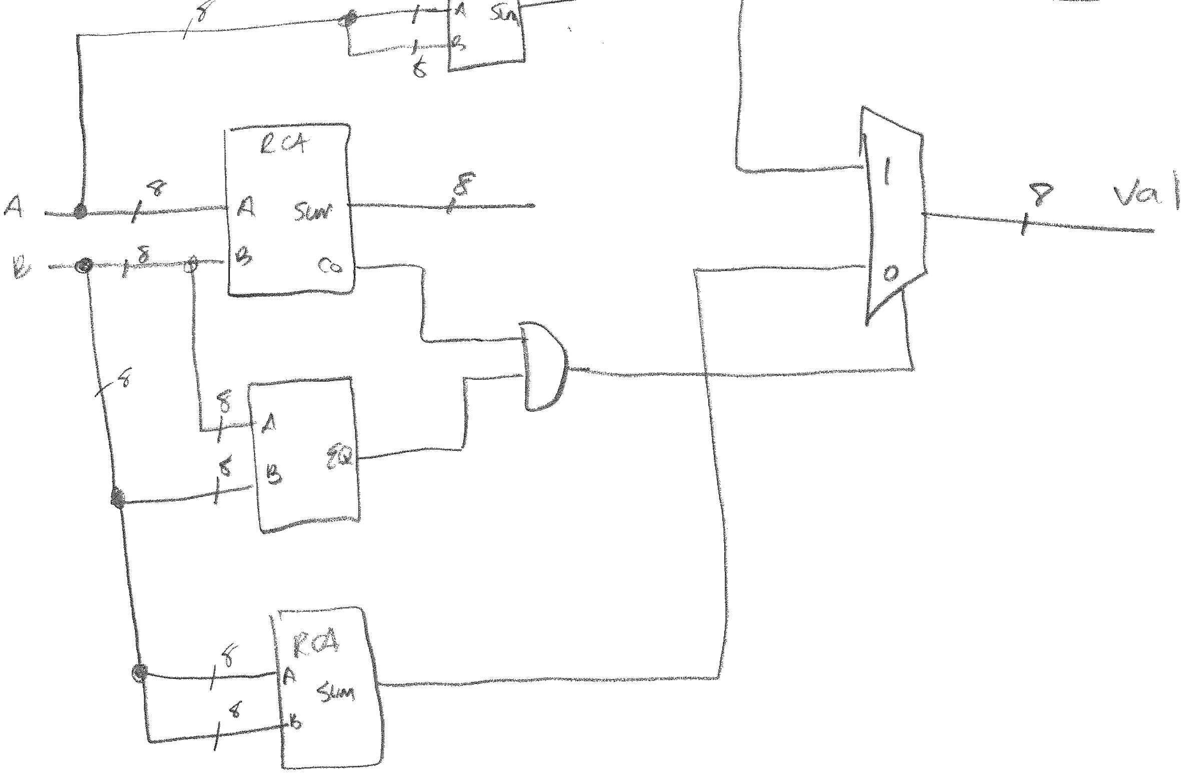
External & internal control



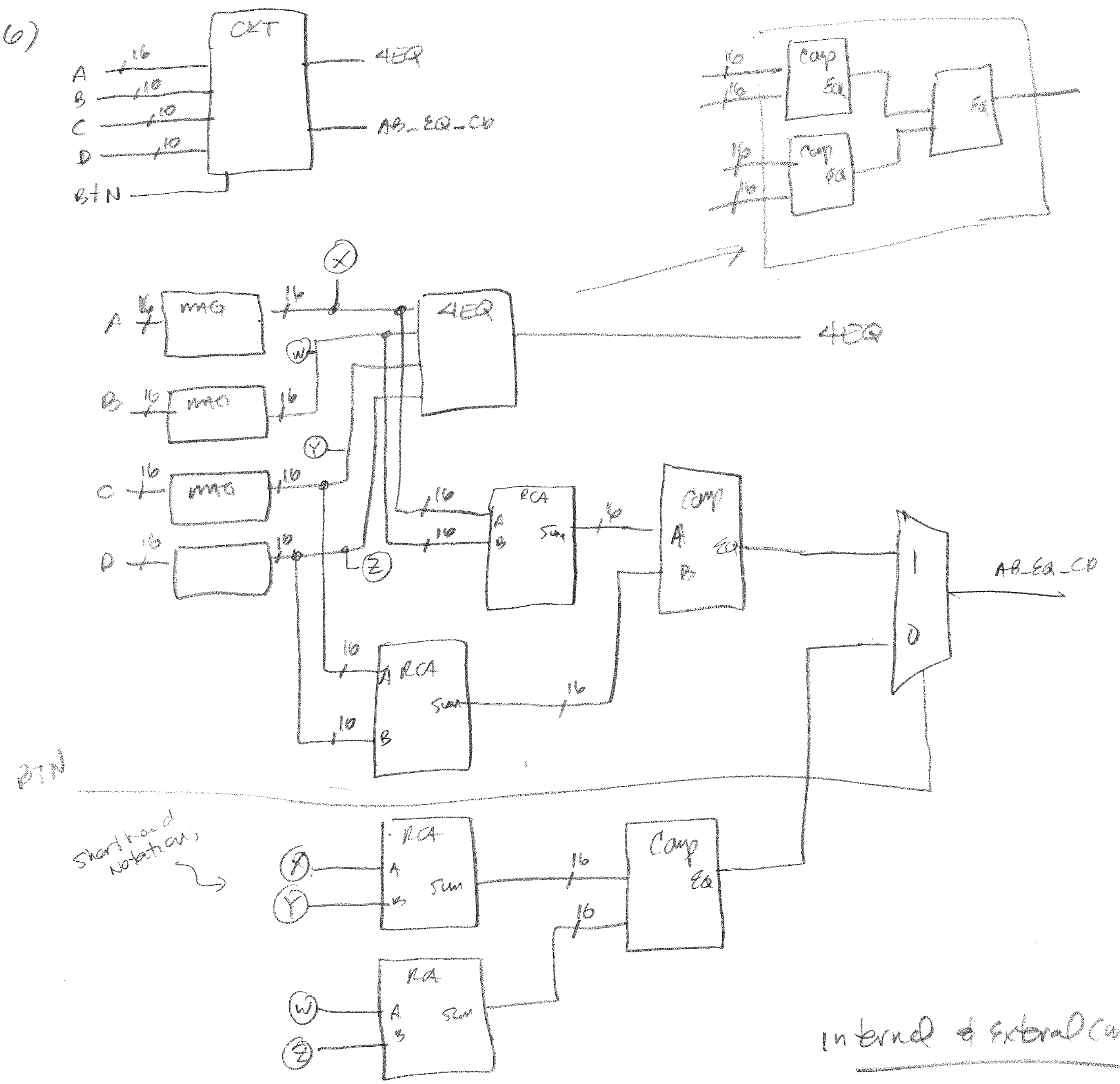
5)



internal control

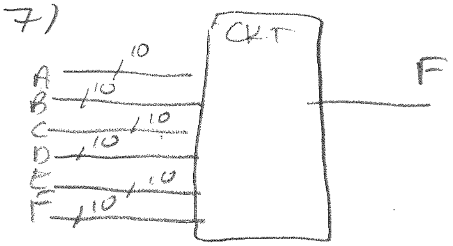


6)



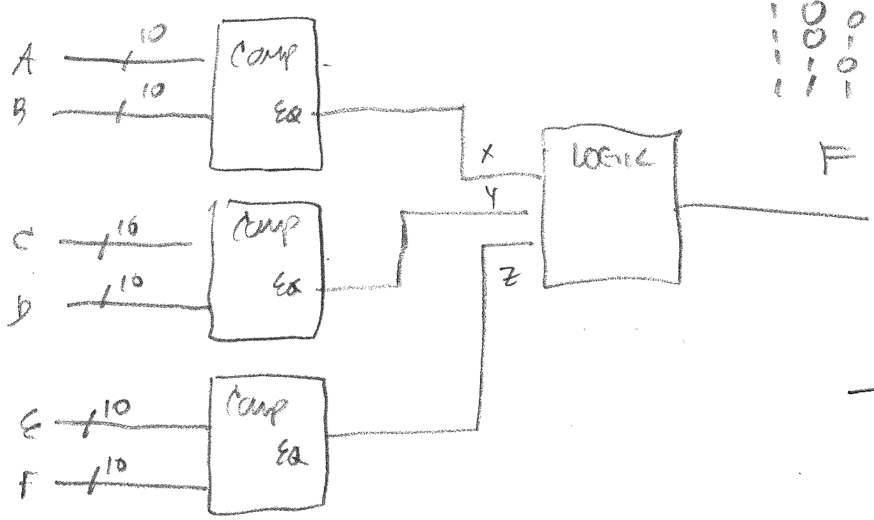
internal & external control

7)



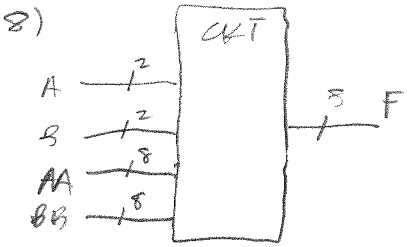
X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

TABLE DEFINES A DECODER

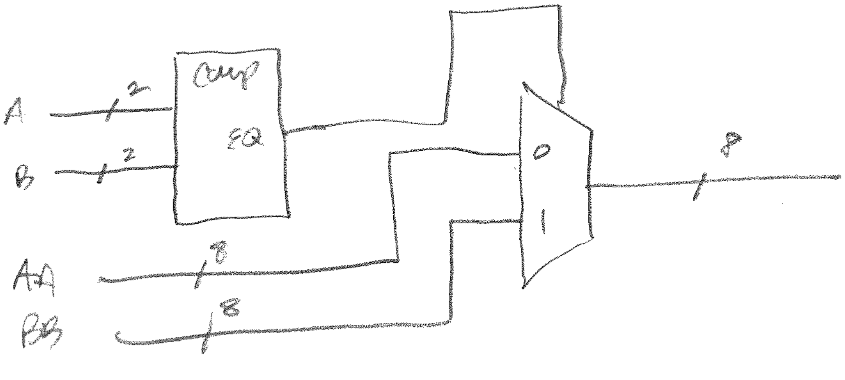


NO CONTROL

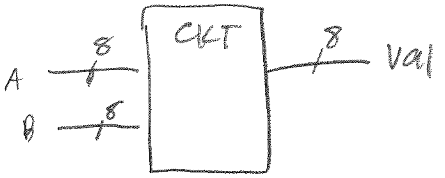
8)

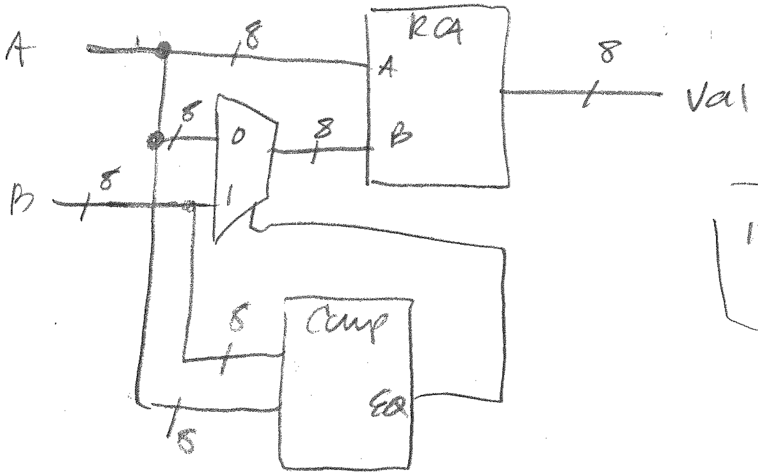


INTERNAL CONTROL



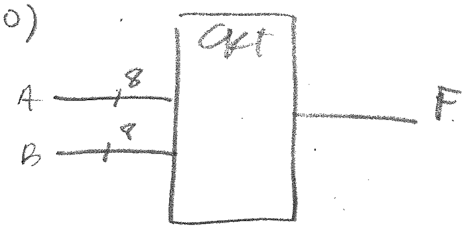
9)



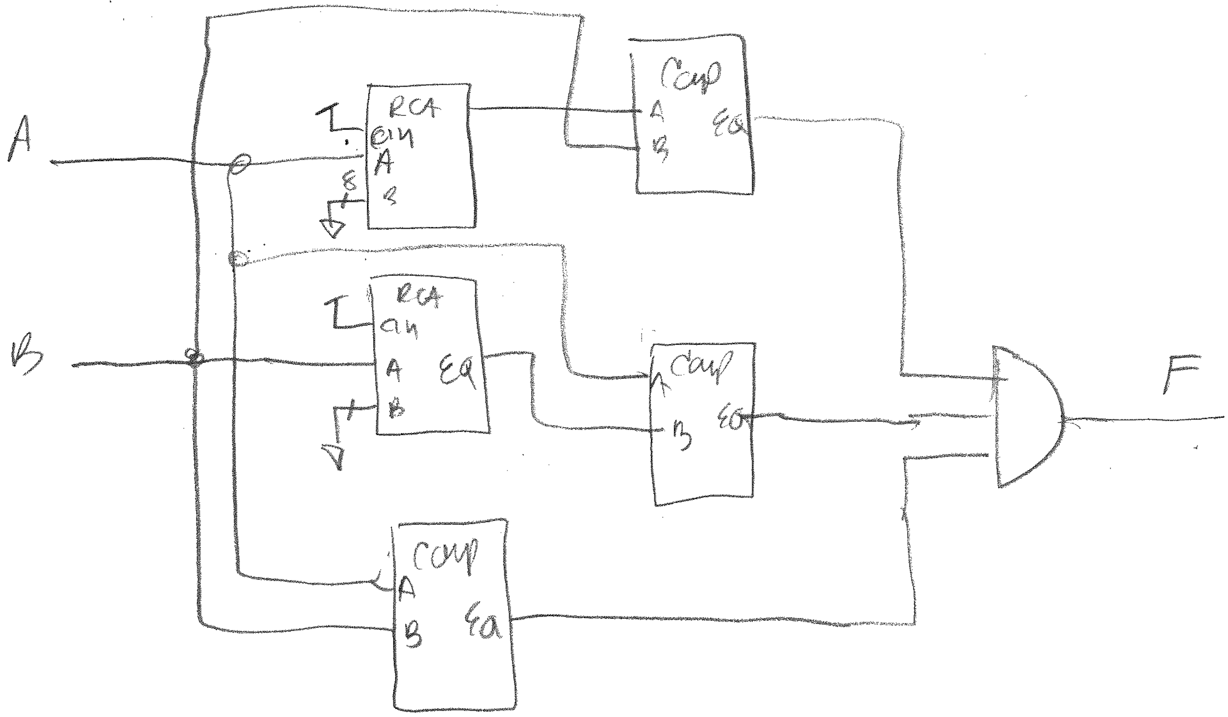


internal control

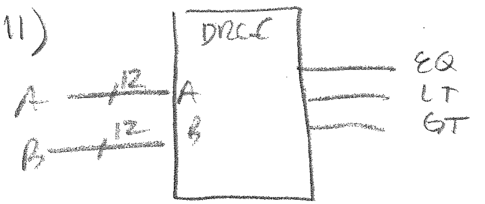
10)



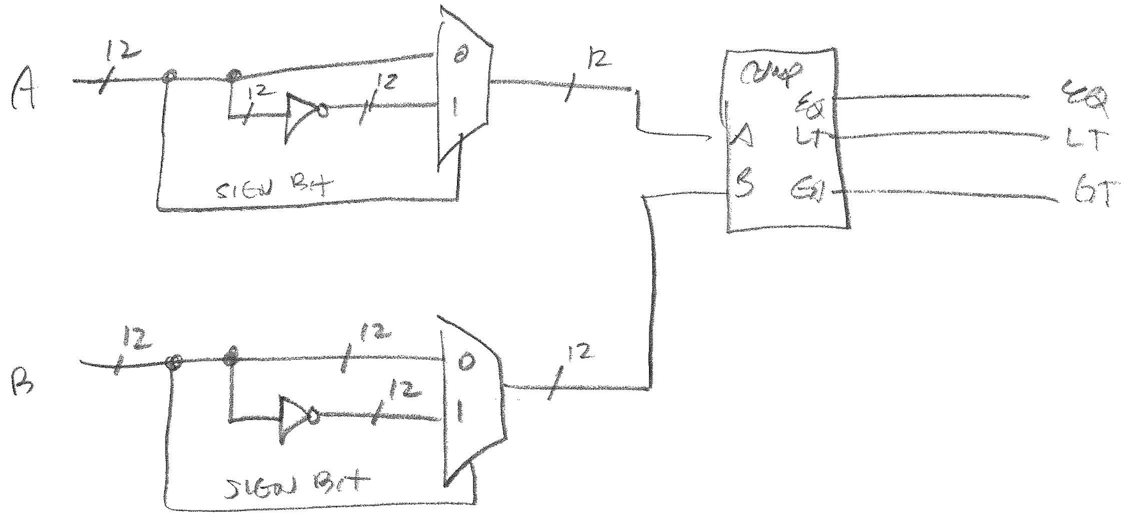
No Control



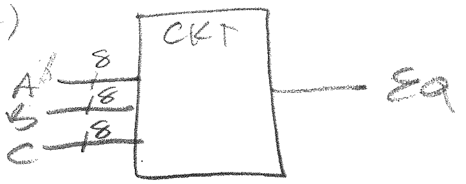
11)



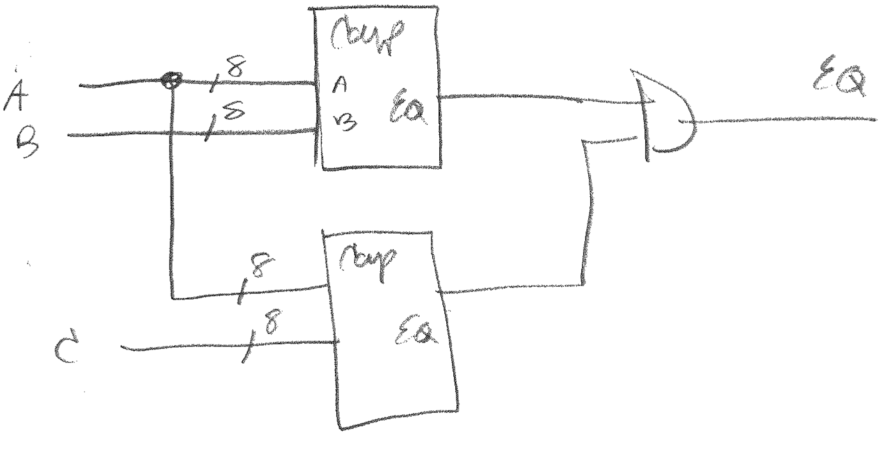
internal control



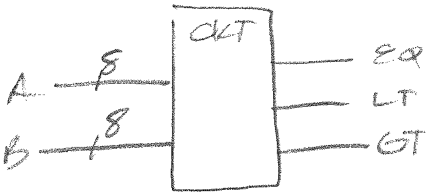
12)



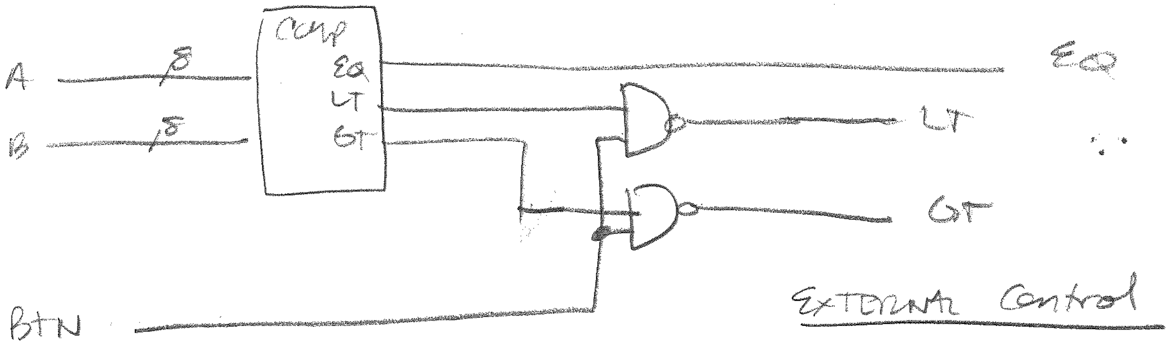
No control



13)

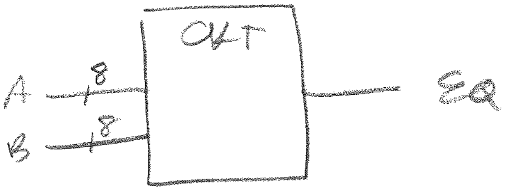


A	B	$\overline{A \oplus B}$
0	0	1
0	1	1
1	0	1
1	1	0

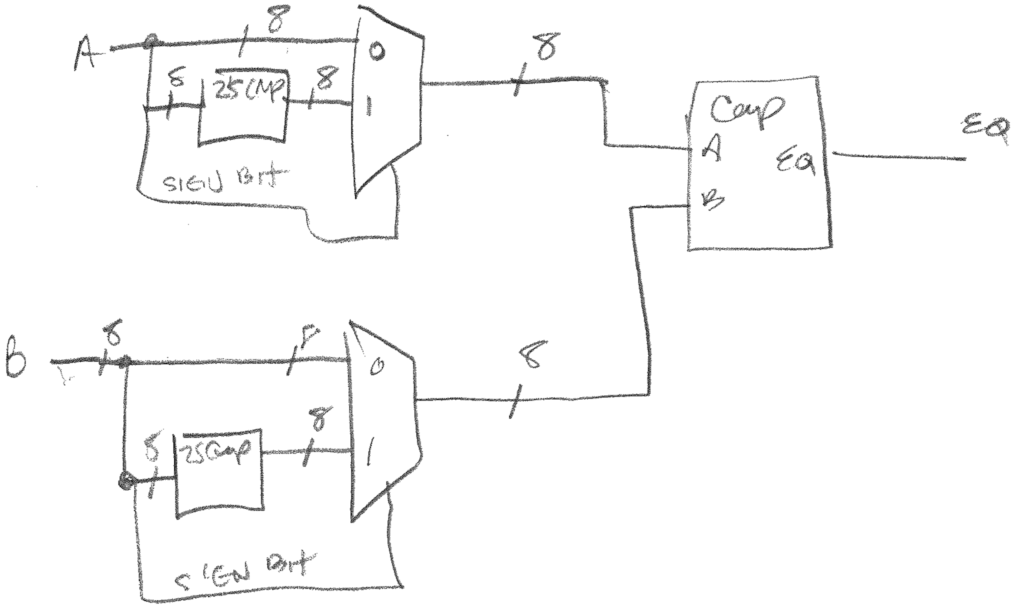


EXTERNAL Control

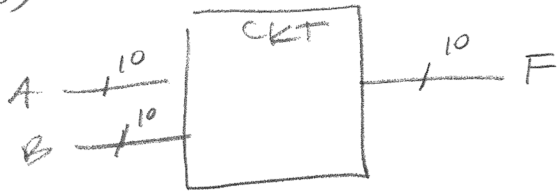
14)



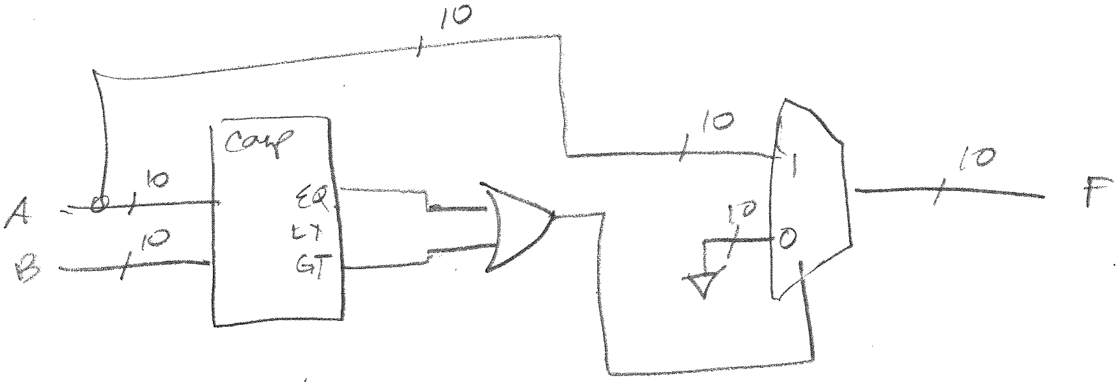
interal control



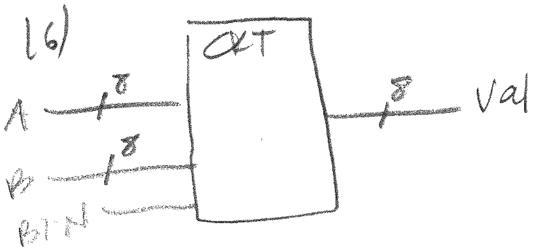
15)

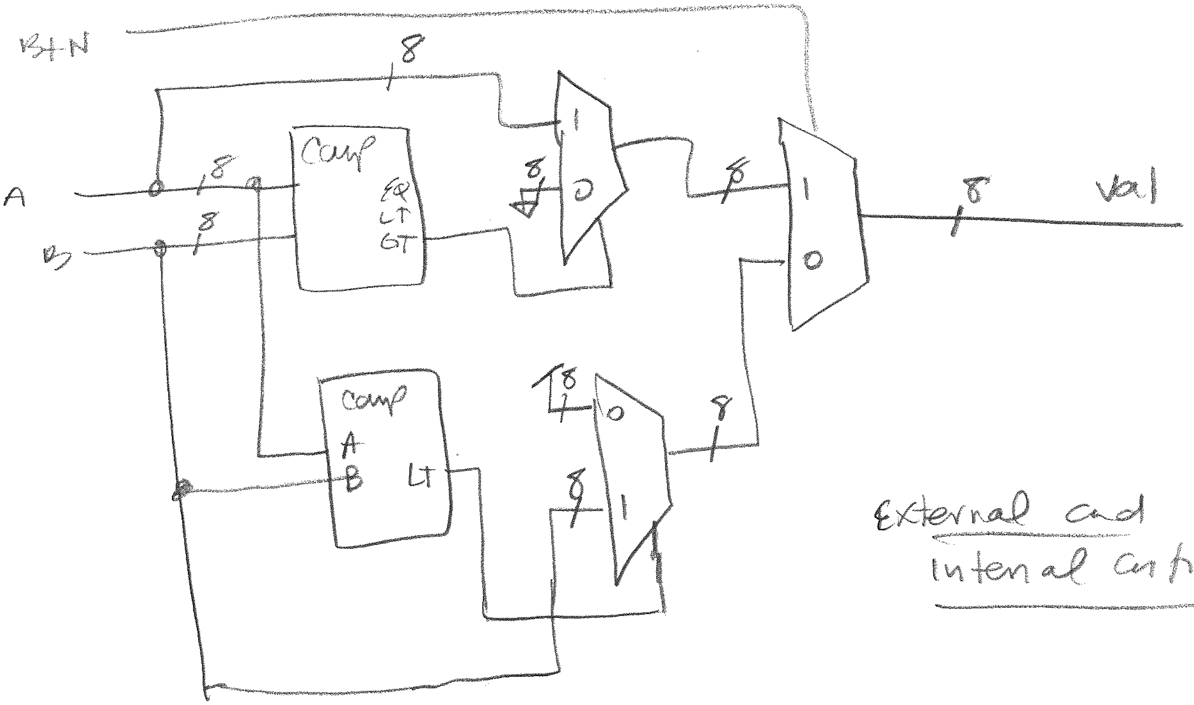


interal control



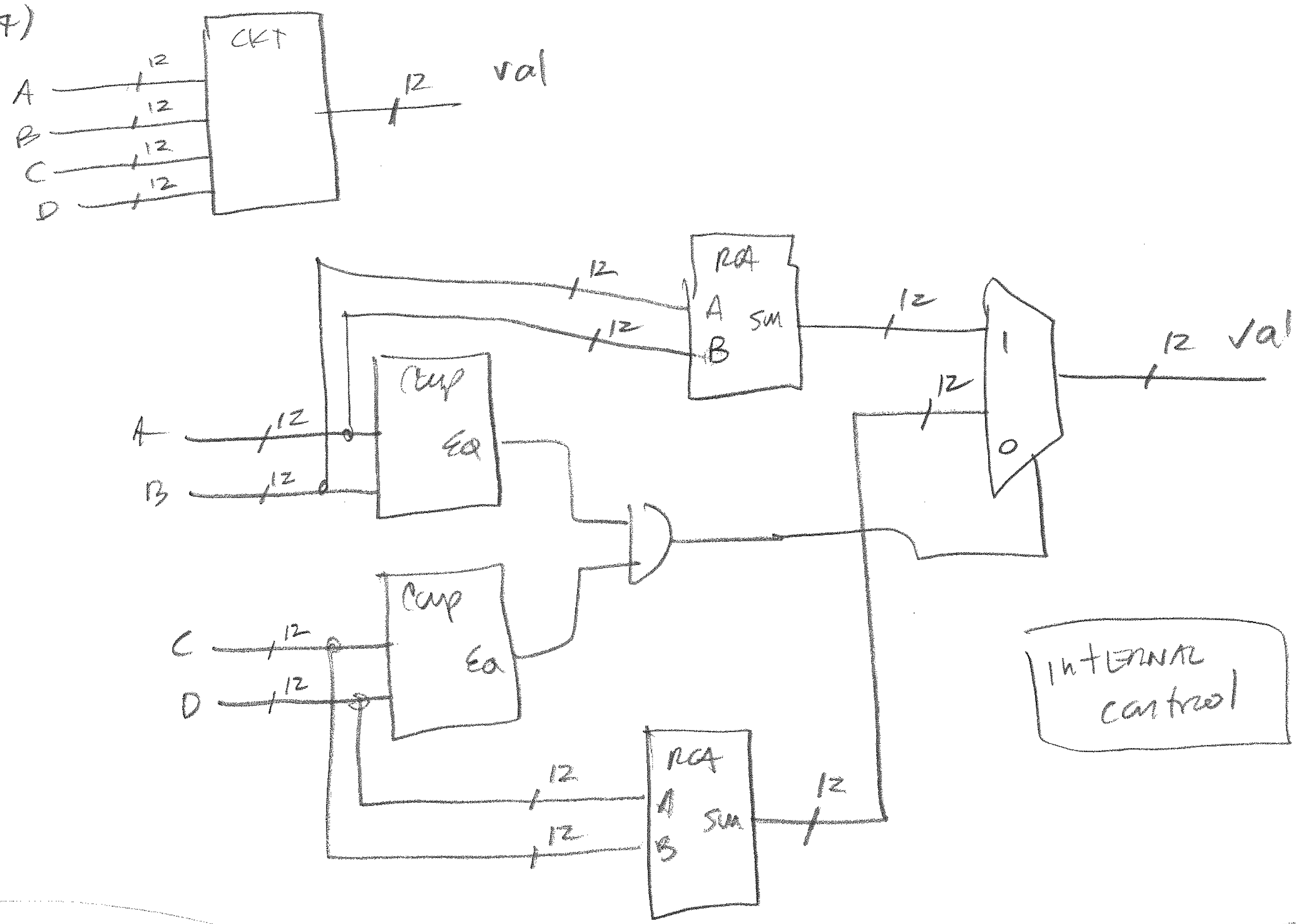
16)



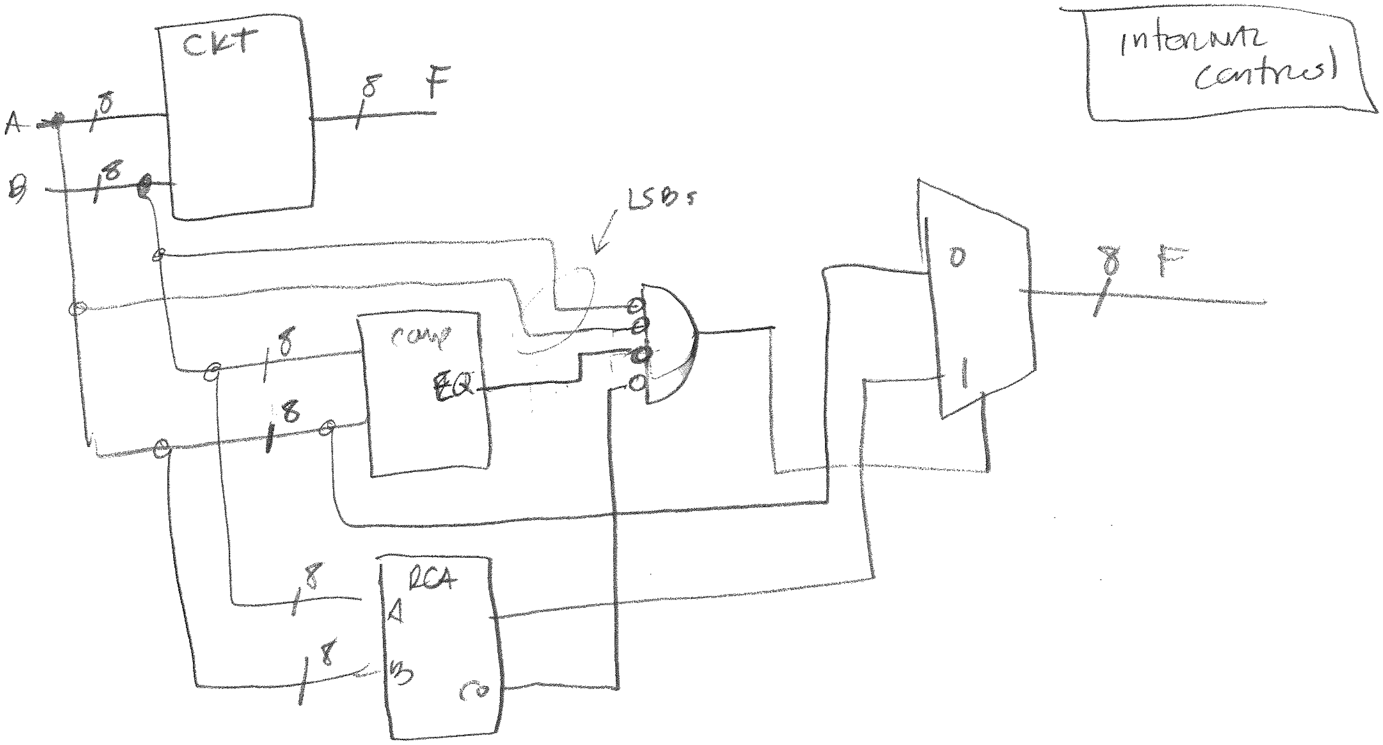


External and internal control

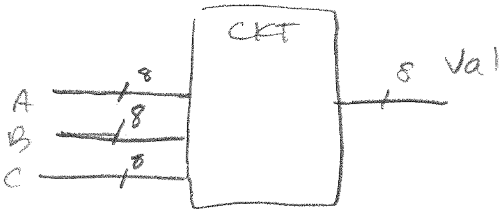
17)


CKT
A
B
C
D
12
12
12
12
12
val
RCA
A
SUM
B
12
12
12
12
1
0
12
val
Cmp
EQ
A
12
B
12
Cmp
EQ
C
12
D
12
RCA
A
SUM
B
12
12
12
internal control

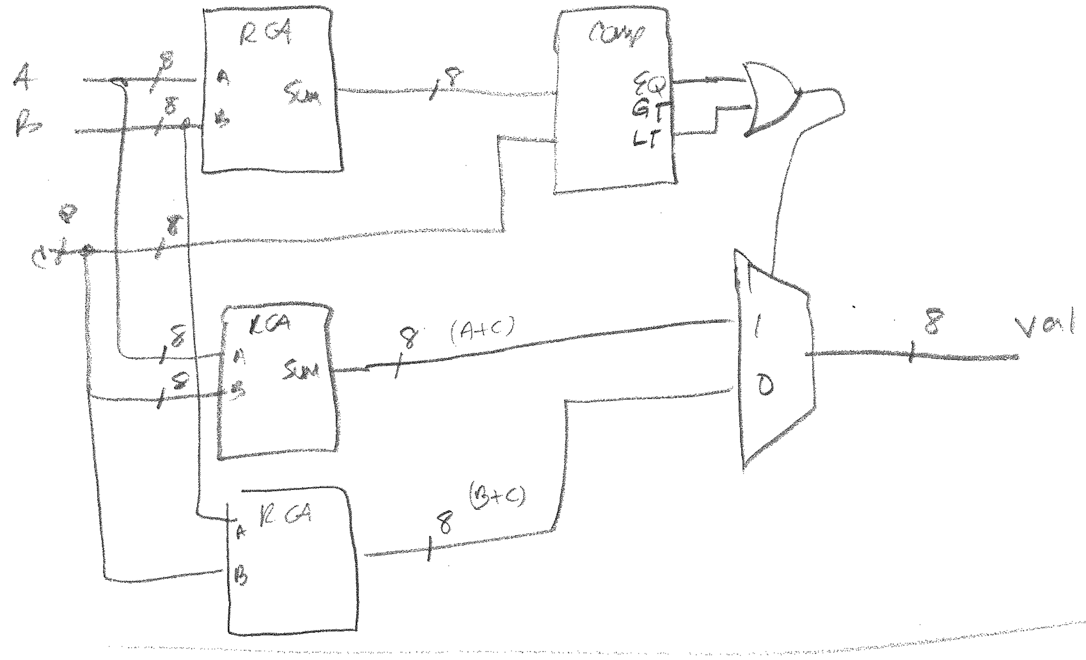
18)


CKT
A
8
B
8
8
F
internal control
LSBs
0
1
8
F
comp
EQ
8
8
RCA
A
B
CO
8
8

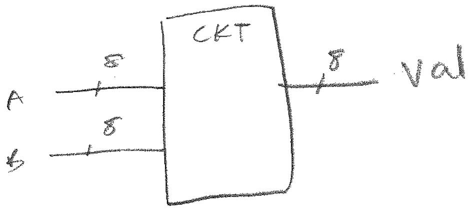
19)



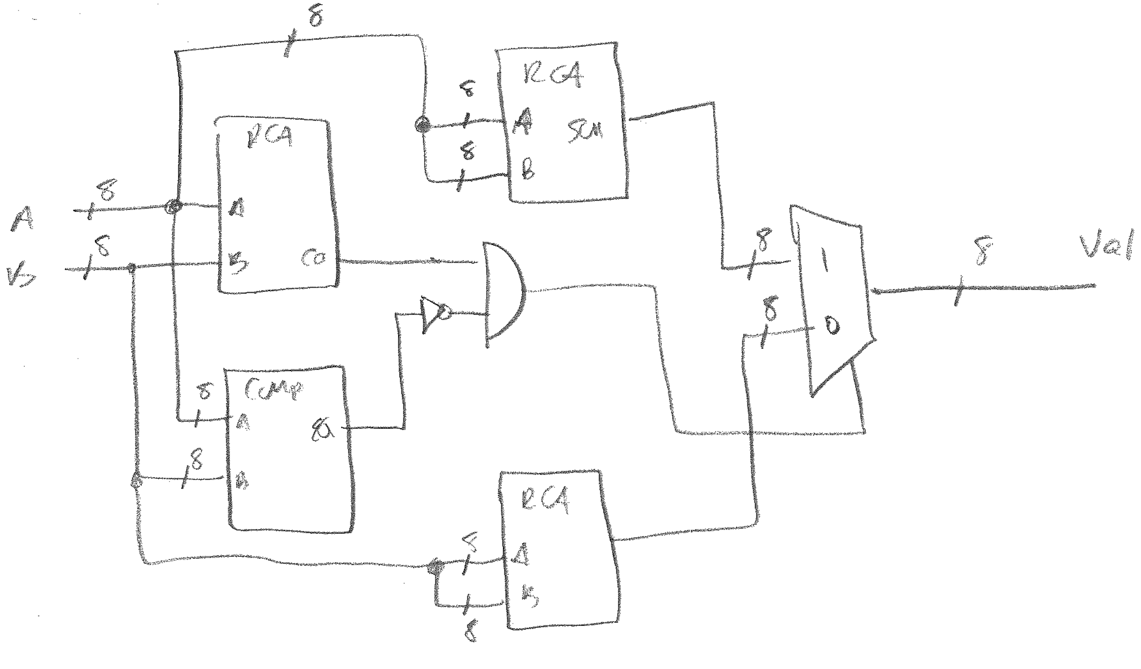
Internal Control



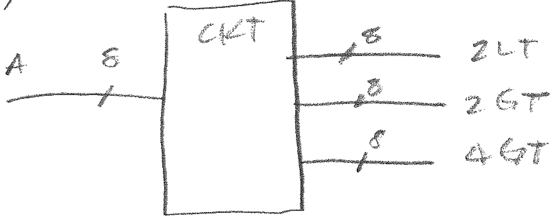
20)



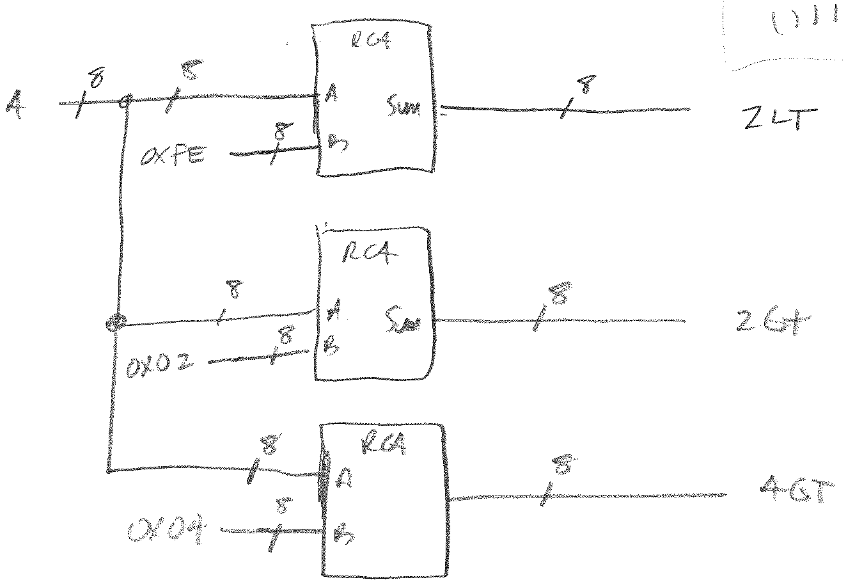
Internal Control



21)

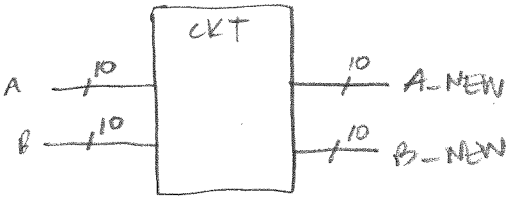


0000 0010 = 2
1111 1110 = -2
FE =

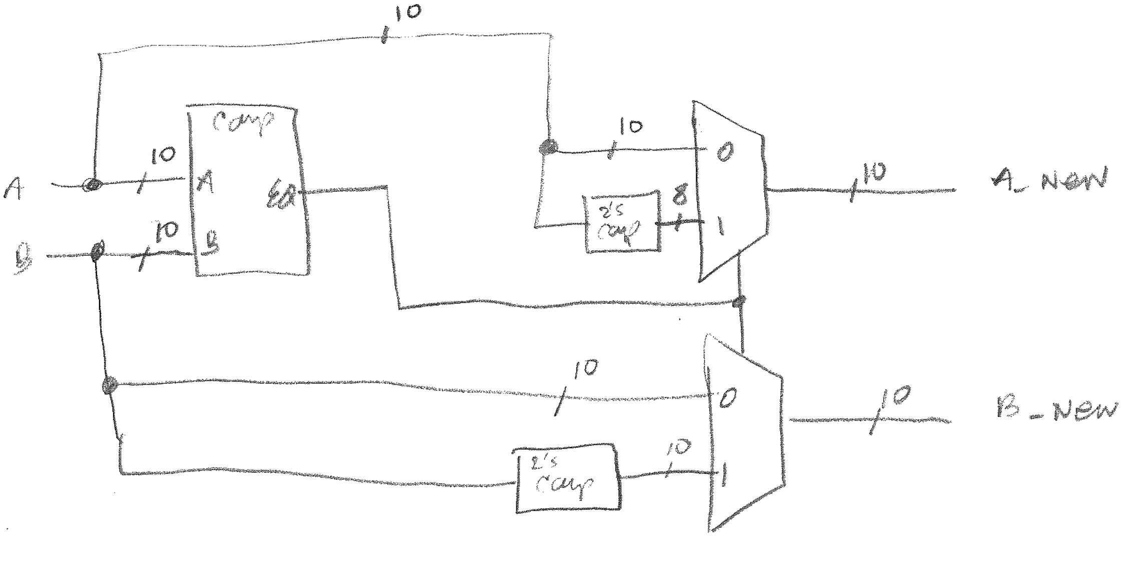


No control

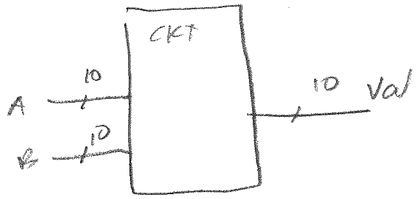
22)



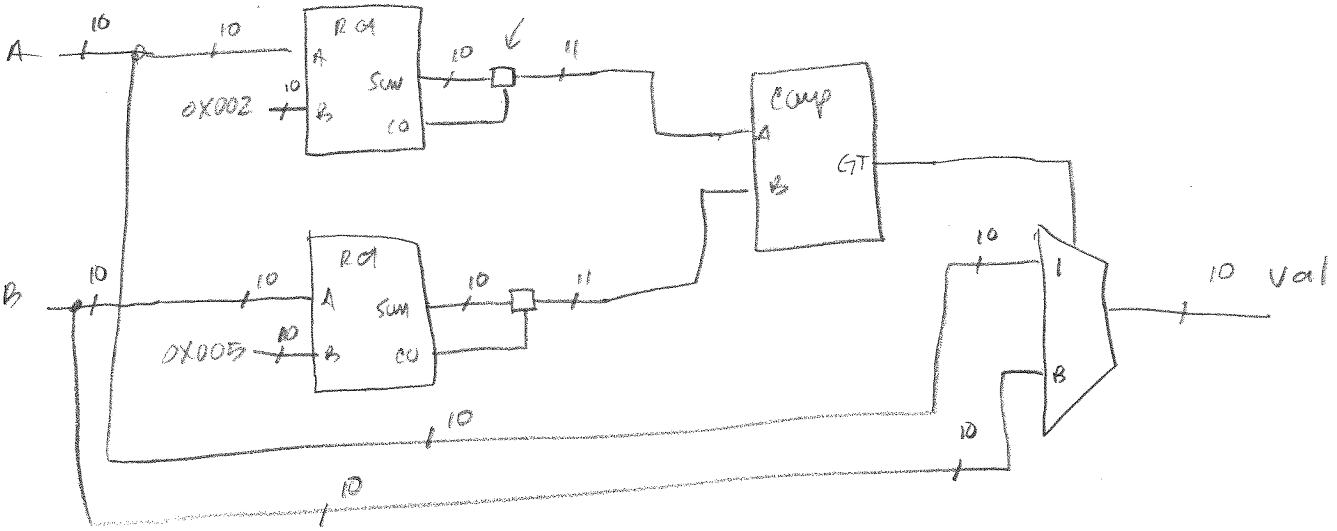
Internal control



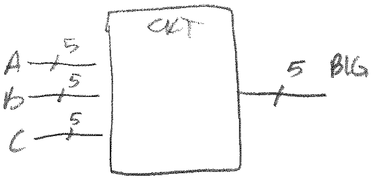
23)

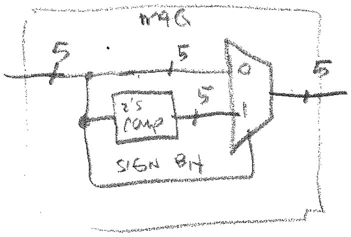


Internal
Control

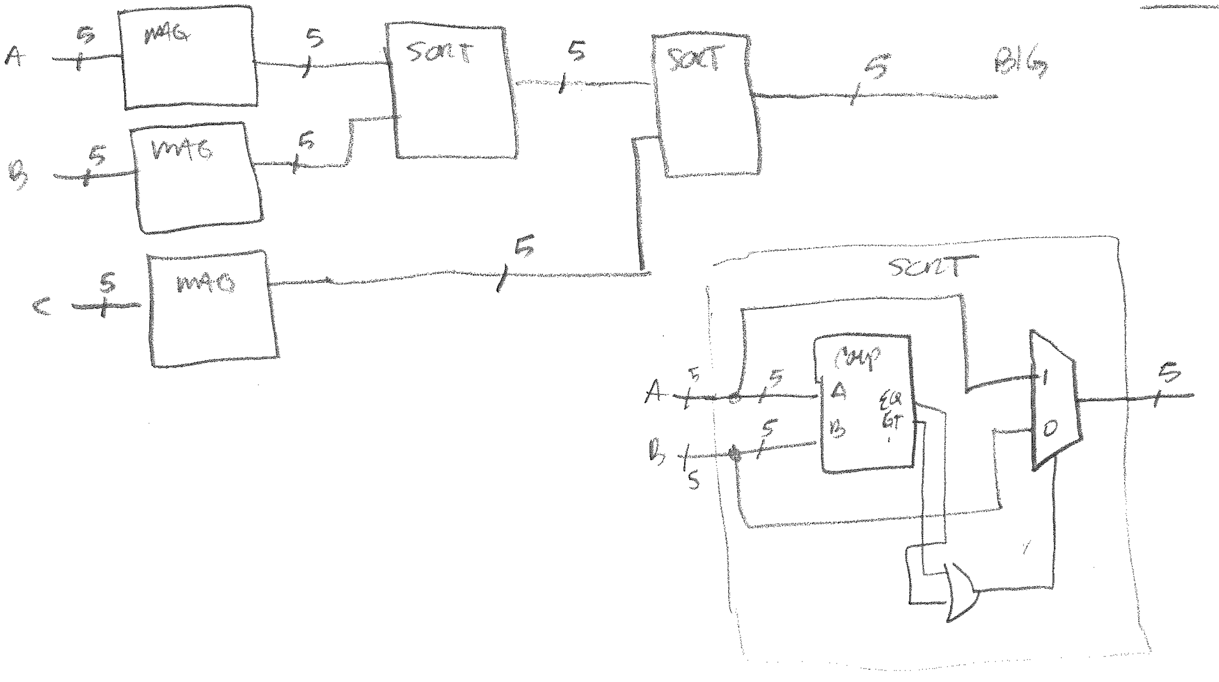


24)

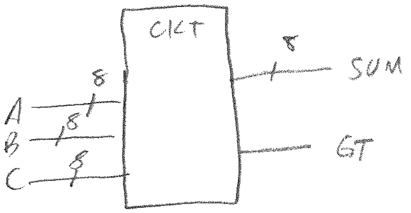




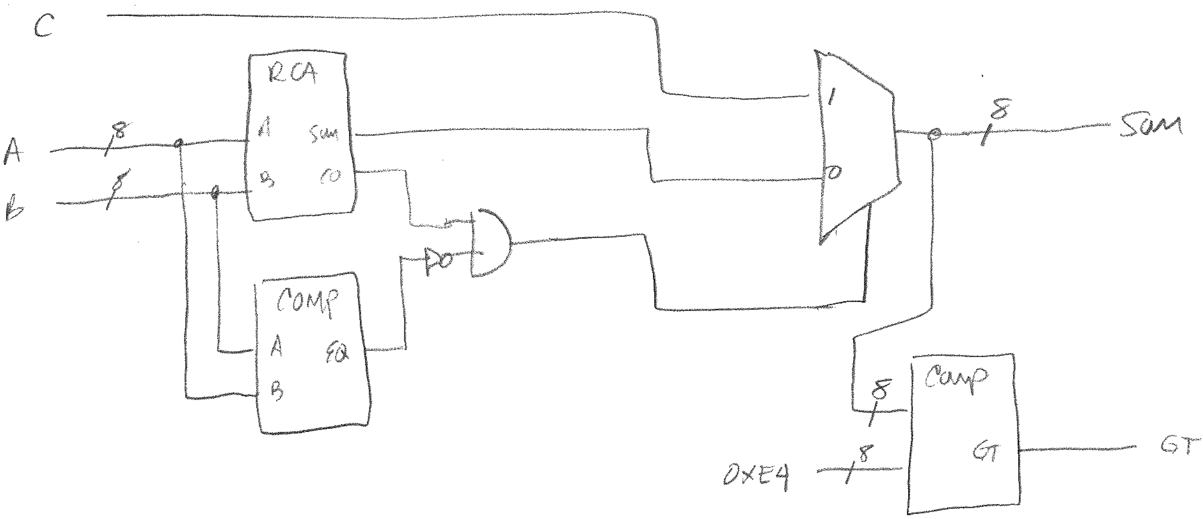
Internal
Control



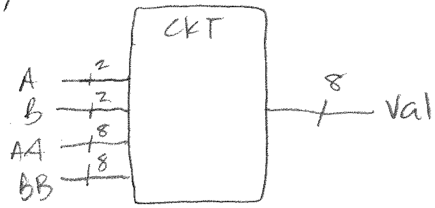
25)



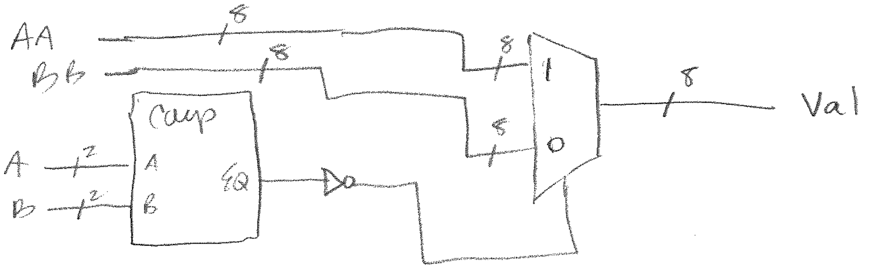
INTERNAL CONTROL



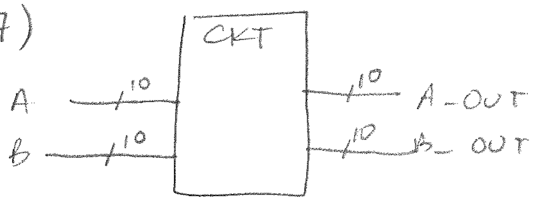
26)



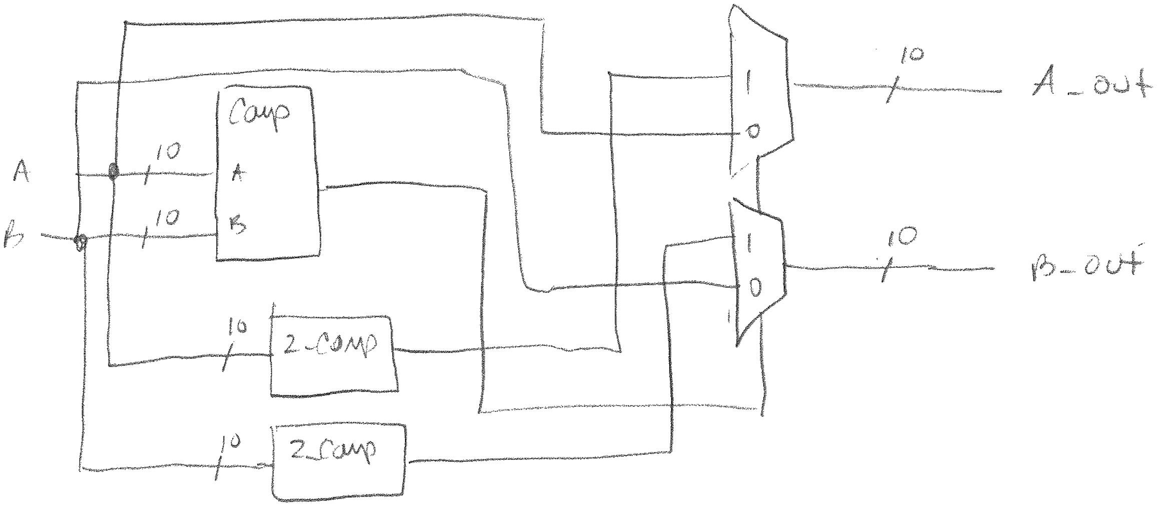
INTERNAL CONTROL



27)



INTERNAL CONTROL



28)

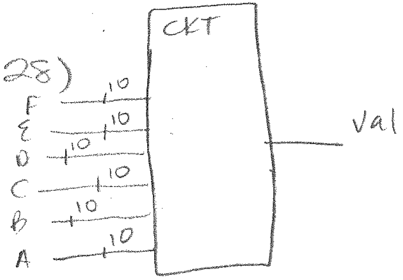
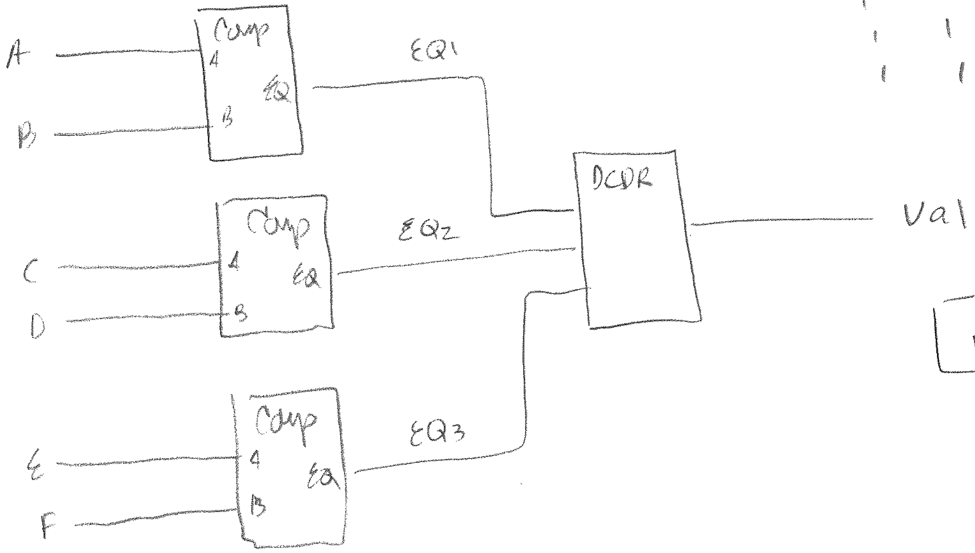


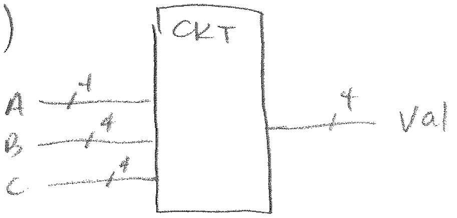
TABLE DEFINING DECODER

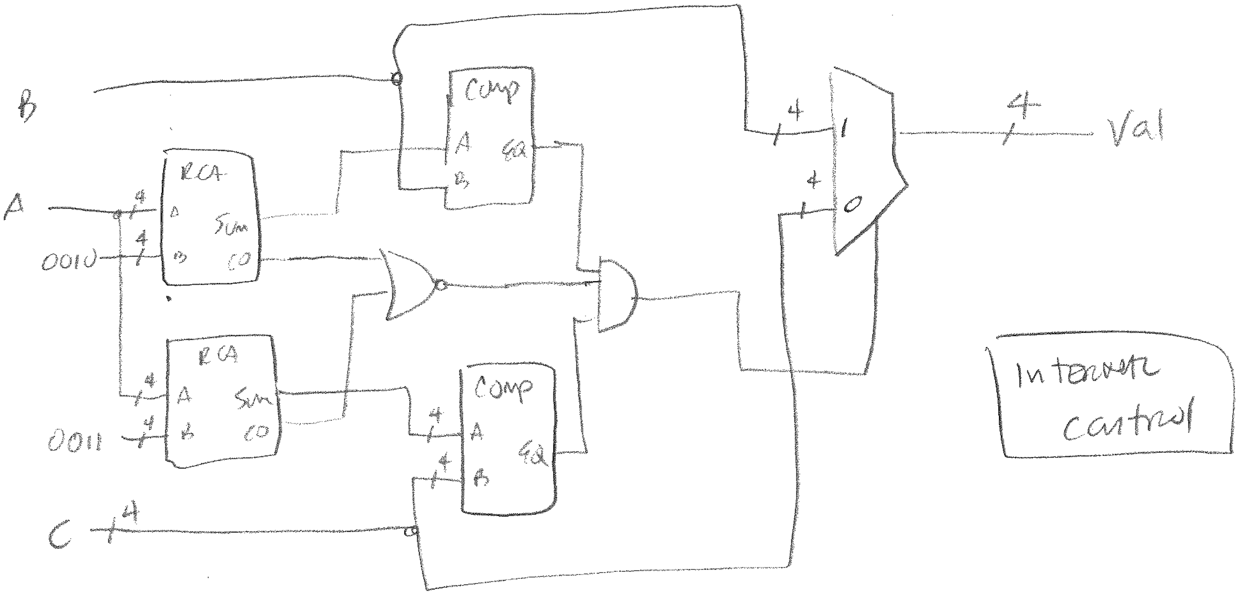
EQ3	EQ2	EQ1	Val
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0



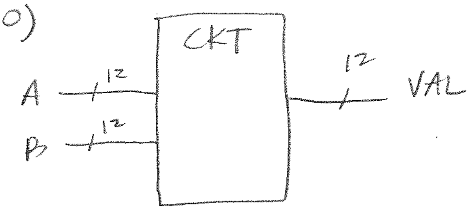
No control

29)

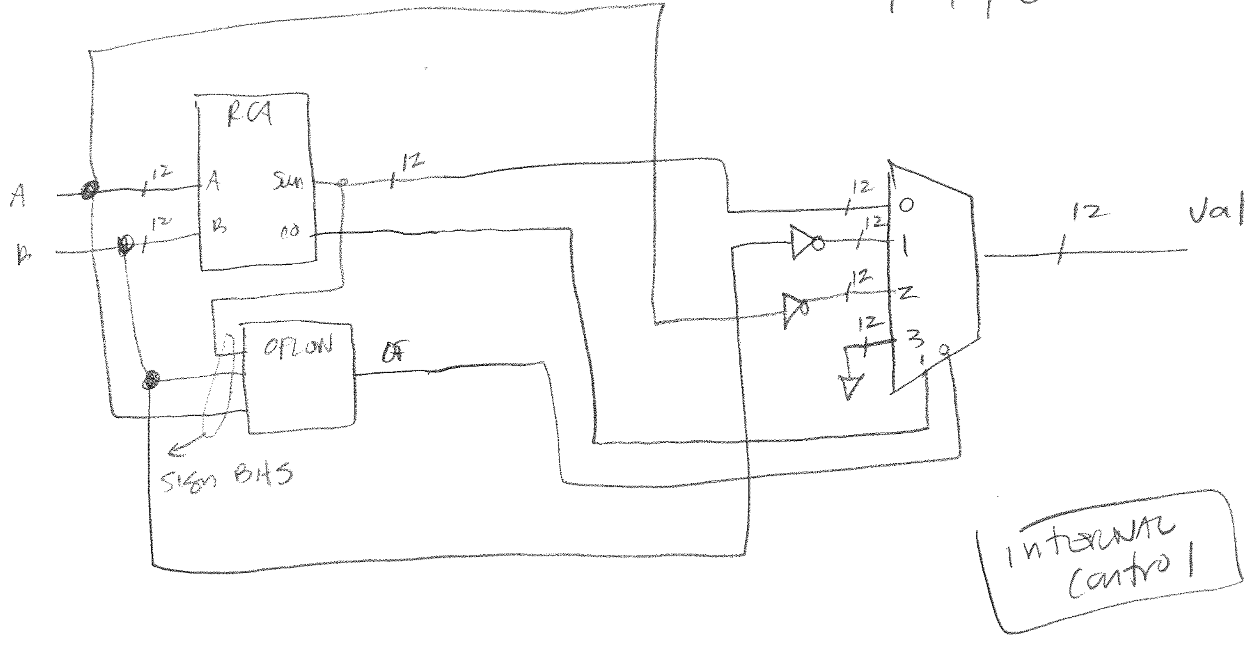




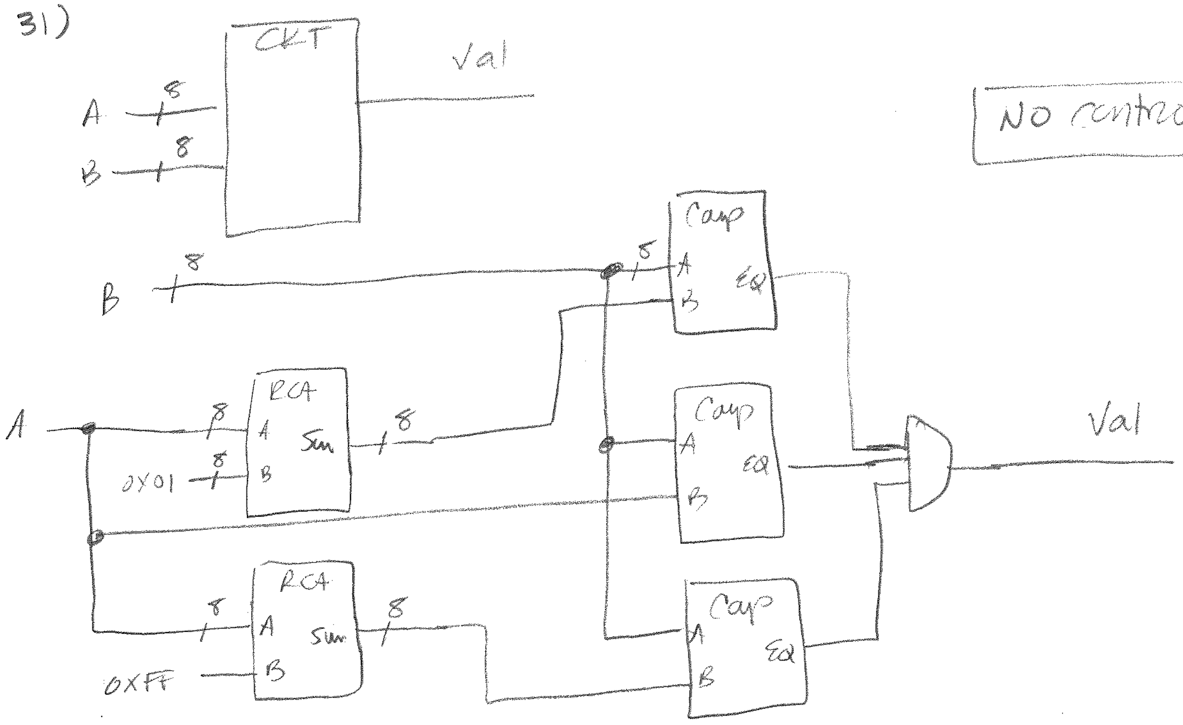
30)



Co	OF	
0	0	A+B
0	1	!B
1	0	!A
1	1	0

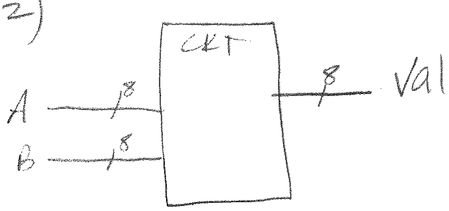


31)

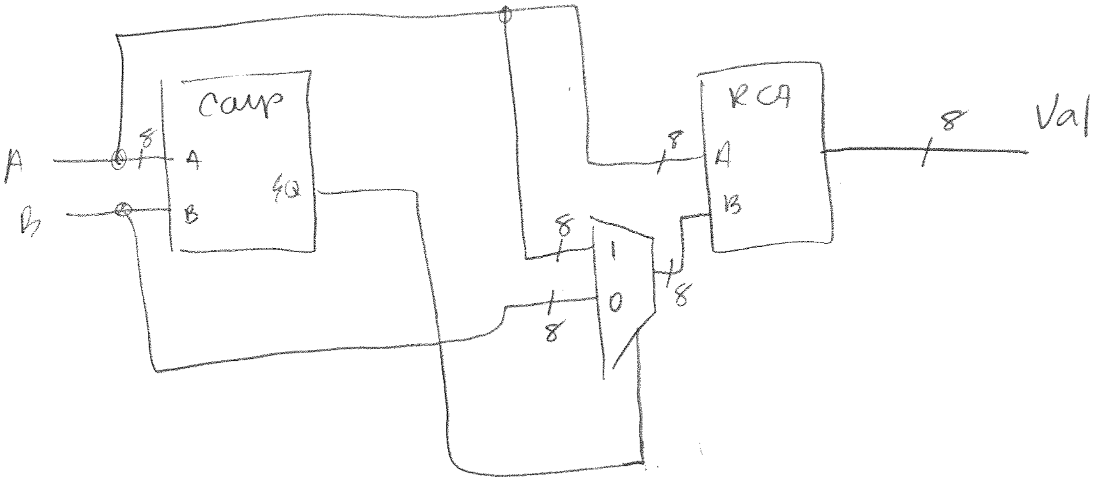


No control

32)



Internal Control

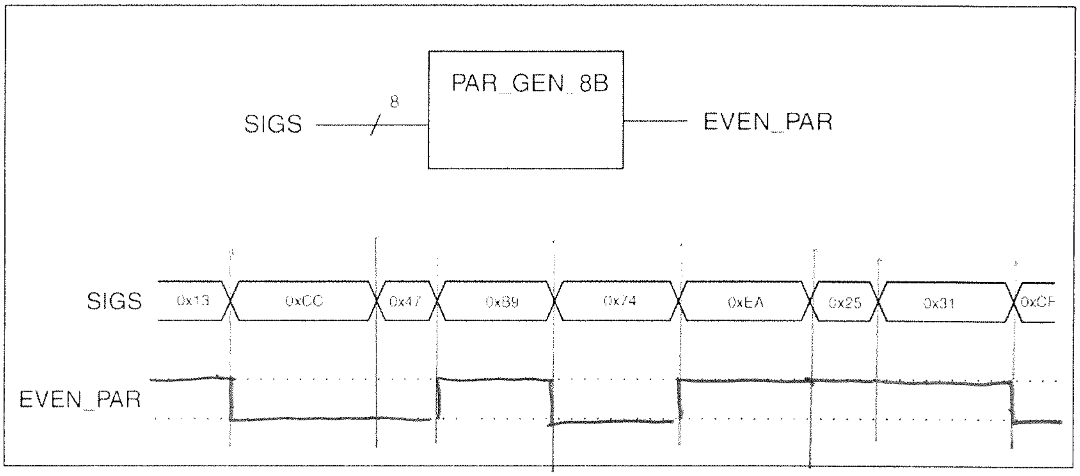


Chapter Exercises

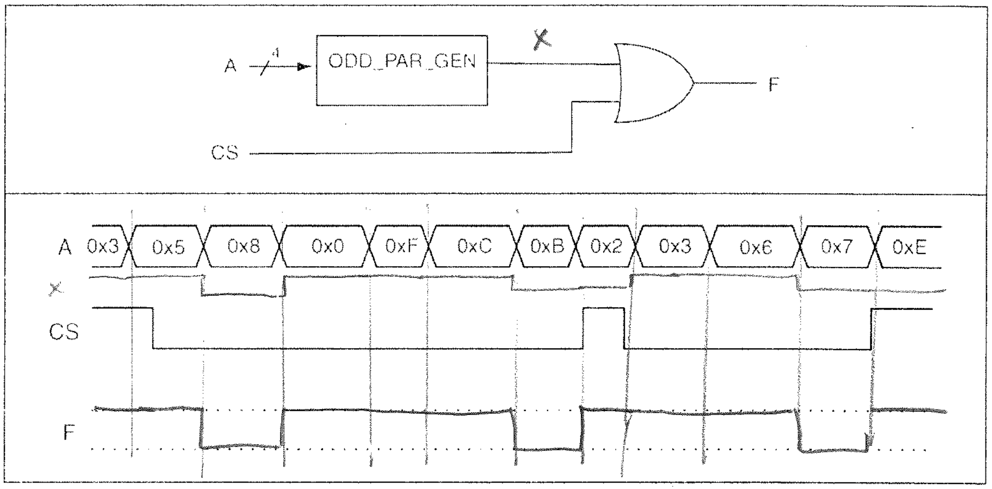
Chapter 19 Solutions

1

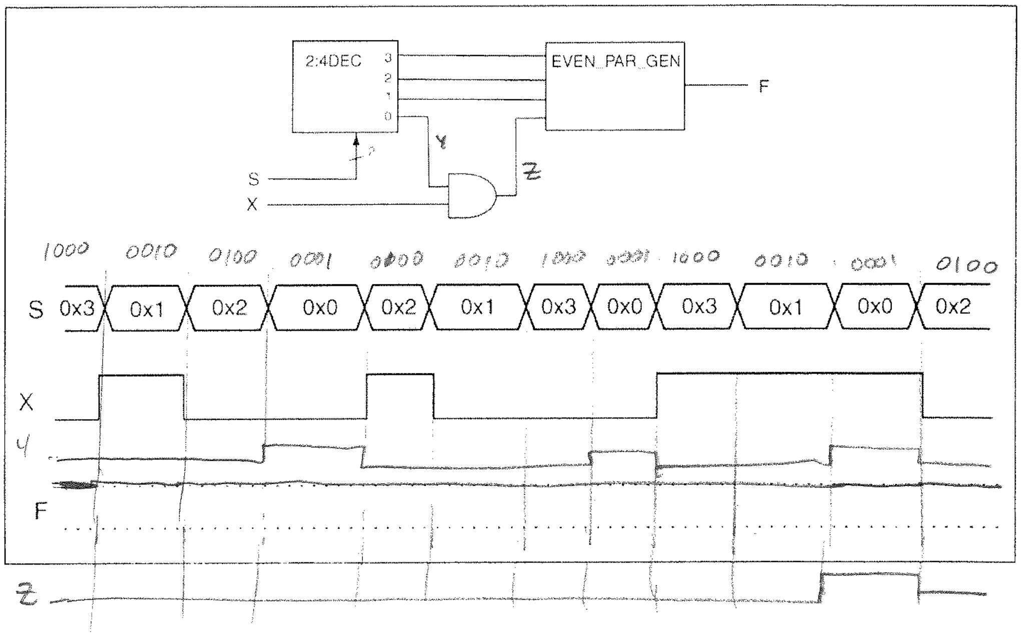
1) Complete the timing diagram shown below considering the given schematic symbol. Consider the circuit to generate even parity for the eight input bits.



2) Use the following circuit to complete the listed timing diagram.



3) Use the following circuit to complete the listed timing diagram



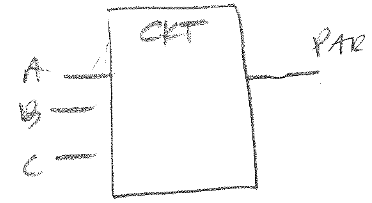
4) Yes; Parity has to do with the number of set bits in a binary value; odd vs. even numbers are a separate issue.

5) Yes; parity has nothing to do with the meaning of the bits, only with the number of set bits in the binary value.

6) Not necessarily. Parity of the result is based on the location of 1's in the items being added. You can't make a general statement such as this and expect it to be always true.

CHAPTER 19 DESIGN PROBLEMS

1)



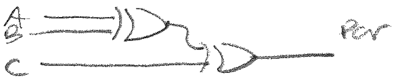
A	B	C	PAR
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$PAR = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

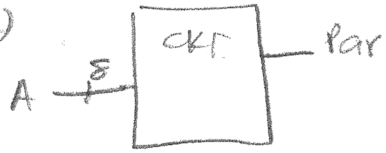
$= \bar{A}(\bar{B}C + B\bar{C}) + A(\bar{B}\bar{C} + BC)$

$= \bar{A}(B \oplus C) + A\overline{(B \oplus C)}$

$PAR = A \oplus B \oplus C$

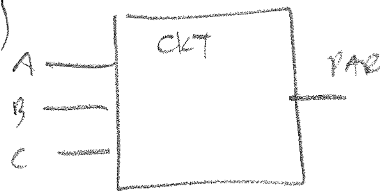


2)



$PAR = A(7) \oplus A(6) \oplus A(5) \oplus A(4) \oplus A(3) \oplus A(2) + A(1) \oplus A(0)$

3)



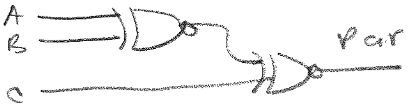
A	B	C	PAR
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

$PAR = \bar{A}\bar{B}\bar{C} + \bar{A}BC + A\bar{B}C + AB\bar{C}$

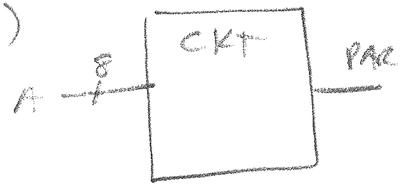
$= \bar{A}(\bar{B}\bar{C} + BC) + A(\bar{B}C + B\bar{C})$

$= \bar{A}(\overline{B \oplus C}) + A(B \oplus C)$

$= A \odot B \odot C$

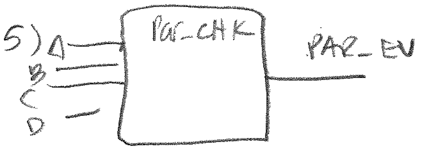


4)



$PAR = A(7) \odot A(6) \odot A(5) \odot A(4) \odot A(3) \odot A(2) \odot A(1) \odot A(0)$

5)



A	B	C	D	Par_ev
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

$Par\_ev = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + \bar{A}BC\bar{D}$

$+ A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + AB\bar{C}\bar{D} + ABCD$

$= \bar{A}\bar{B}(\bar{C}\bar{D} + CD) + \bar{A}B(\bar{C}D + C\bar{D}) +$

$A\bar{B}(\bar{C}D + C\bar{D}) + AB(\bar{C}\bar{D} + CD)$

$= \bar{A}\bar{B}(\overline{C \oplus D}) + \bar{A}B(C \oplus D) +$

$A\bar{B}(C \oplus D) + AB(\overline{C \oplus D})$

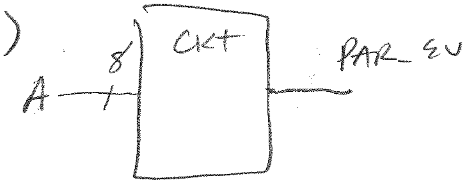
$= (\bar{A}\bar{B} + AB)(\overline{C \oplus D}) +$

$(\bar{A}B + A\bar{B})(C \oplus D)$

$= (\overline{A \oplus B})(\overline{C \oplus D})(A \oplus B)(C \oplus D)$

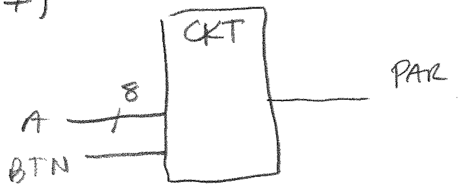
$= \overline{A \oplus B \oplus C \oplus D}$

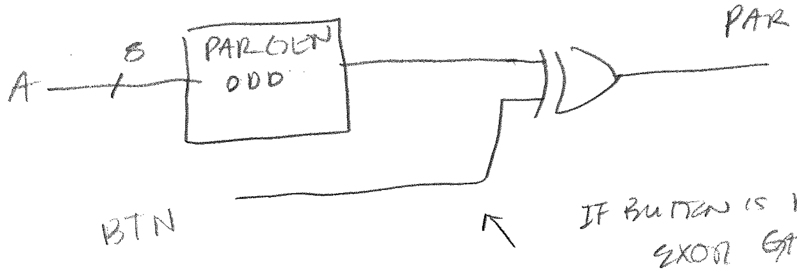
6)



$$PAR\_EV = \overline{A(7) \oplus A(6) \oplus A(5) \oplus A(4) \oplus A(3) \oplus A(2) \oplus A(1) \oplus A(0)}$$

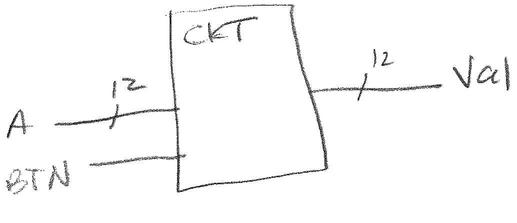
7)



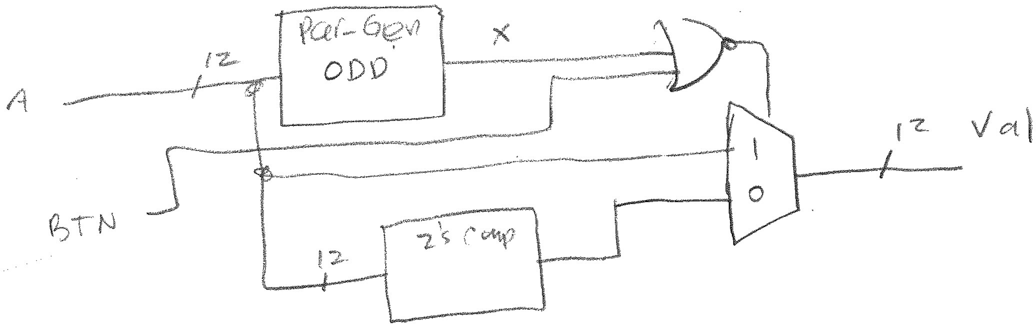


IF BUTTON IS PRESSED, THEN THIS
EXOR GATE BECOMES AN INVERT

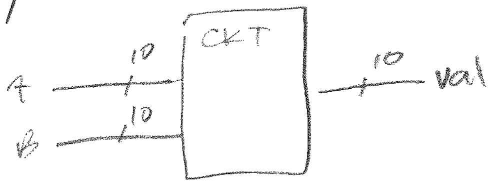
8)

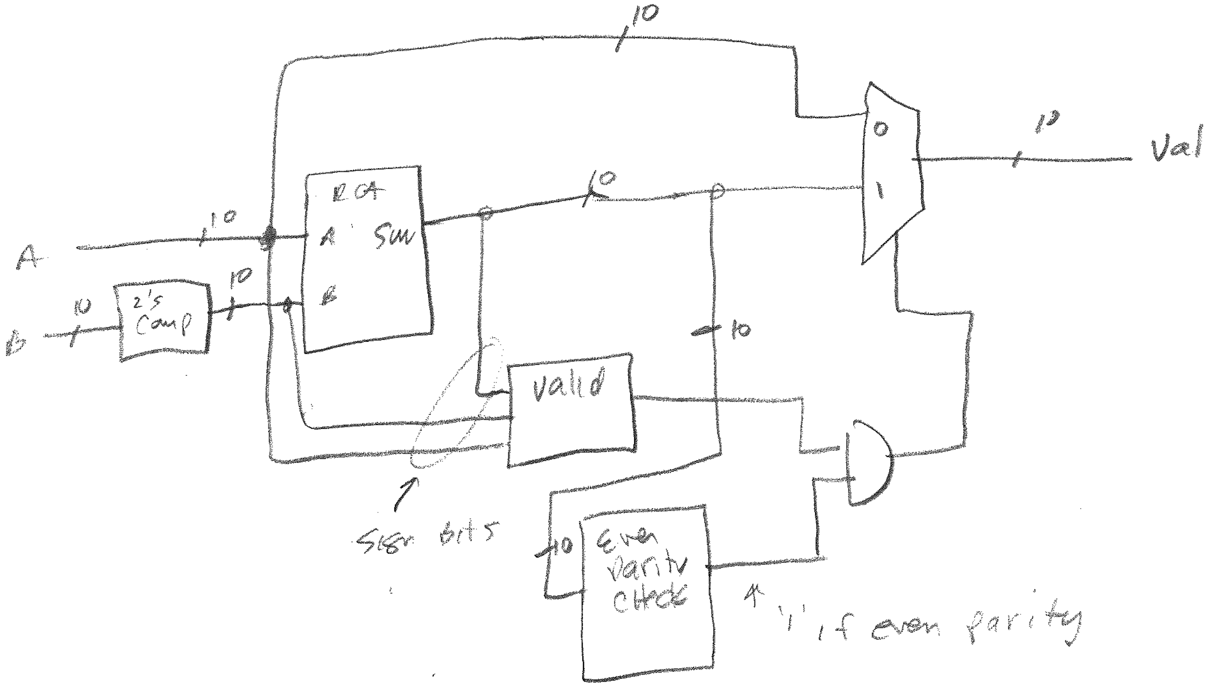


X	BTN	
0	0	1
0	1	0
1	0	0
1	1	0

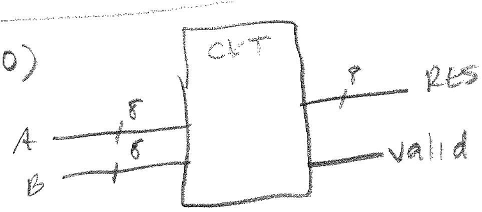


9)



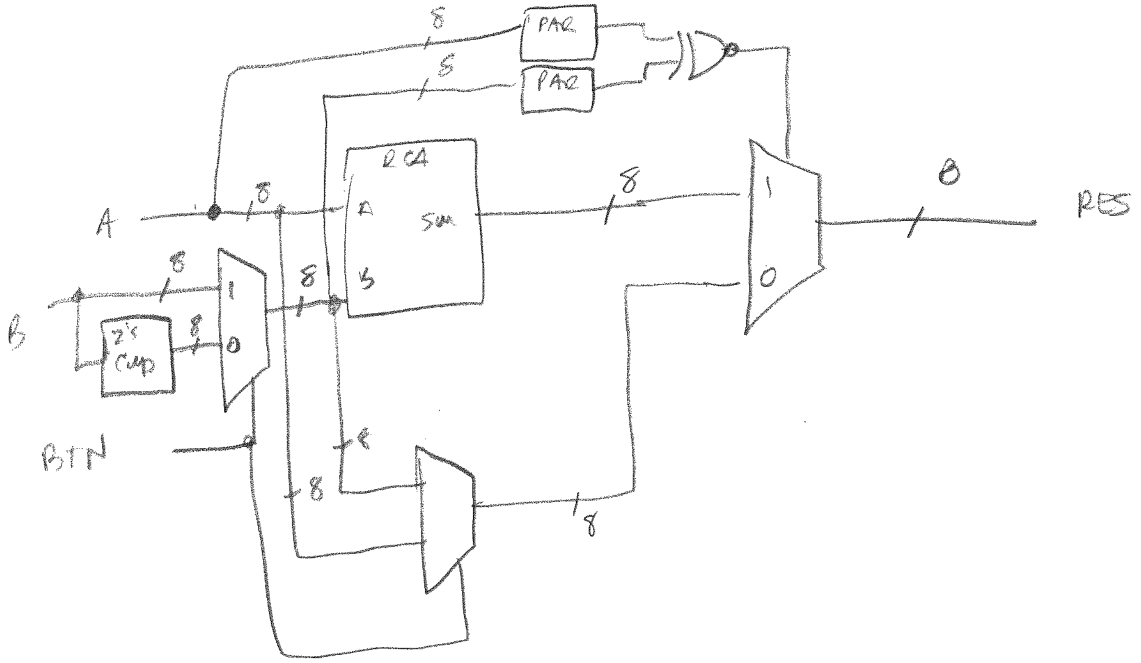


10)

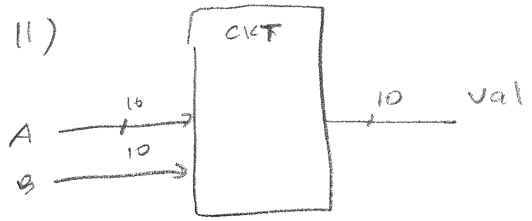


But	SAME PAR	out
0	0	B
0	1	A - B
1	0	A
1	1	A + B

too hard to reason out... make a table!



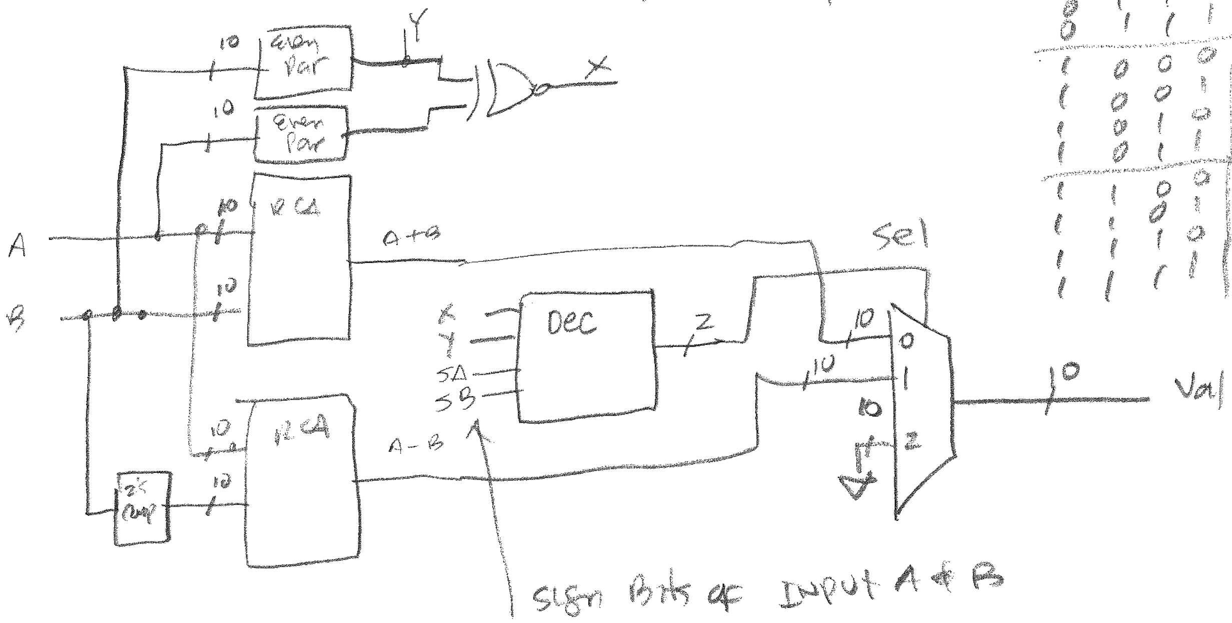
11)



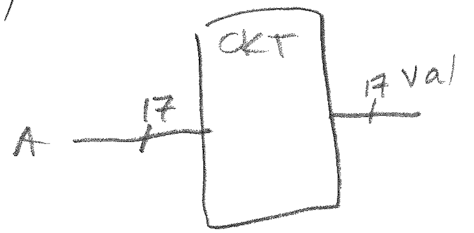
Sign	Par	
Both +	Both ODD	A - B
DIFF	Both Even	A + B
Else		Ø

SFD!

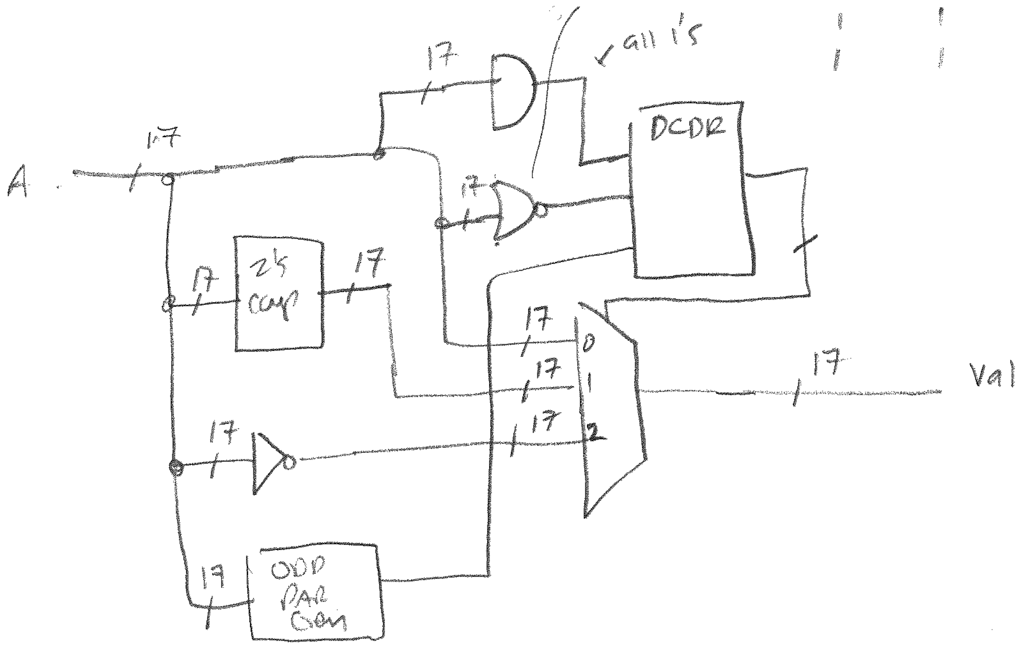
SA	SB	Y	X		Sel
0	0	0	0	Ø	10
0	0	0	1	A - B	01
0	0	1	0	Ø	10
0	0	1	1	Ø	10
0	1	0	0	Ø	10
0	1	0	1	Ø	10
0	1	1	0	Ø	10
0	1	1	1	A+B	00
1	0	0	0	Ø	10
1	0	0	1	Ø	10
1	0	1	0	Ø	10
1	0	1	1	A+B	00
1	1	0	0	Ø	10
1	1	0	1	Ø	10
1	1	1	0	Ø	10
1	1	1	1	Ø	10



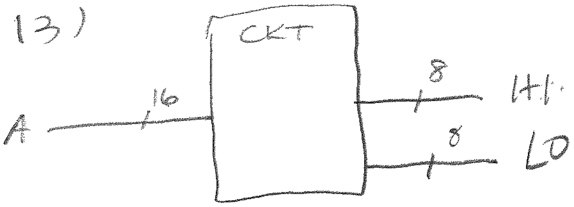
12)

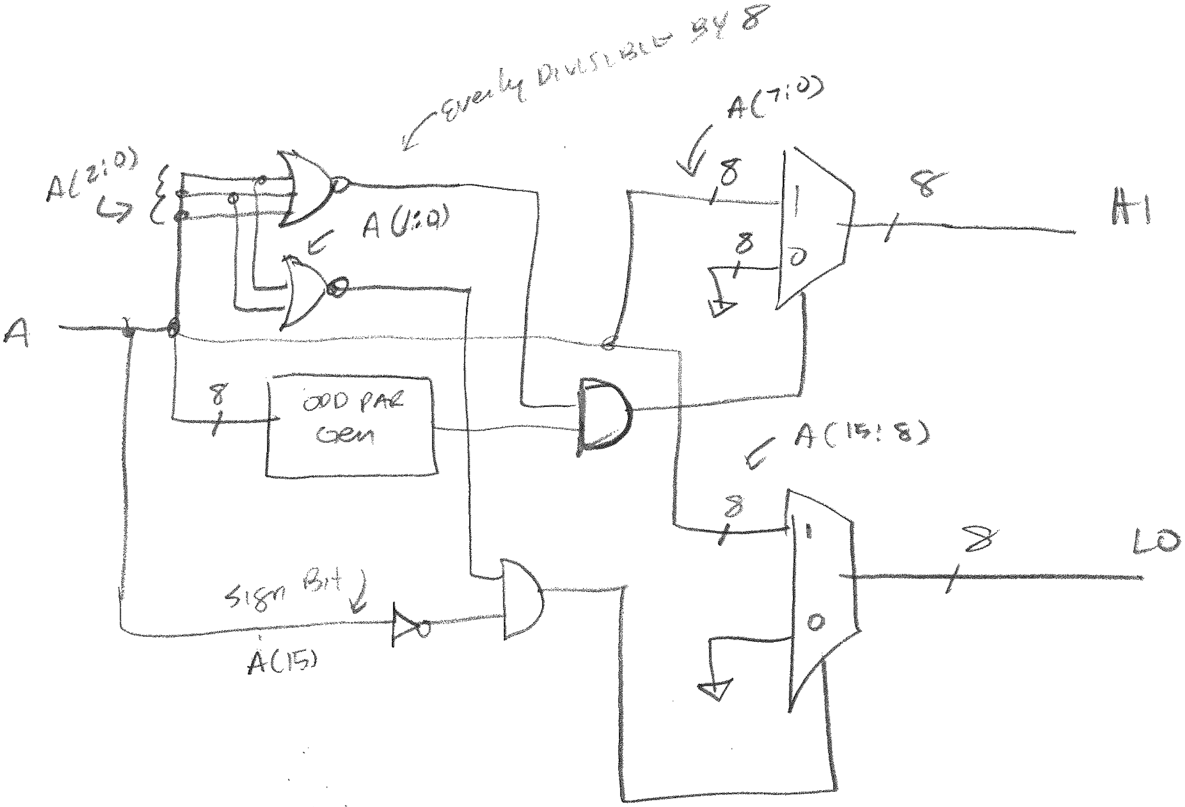


ODD Par	all 1's	all 0's		Sel
0	0	0	-A	01
0	0	1	!A	10
0	1	0	-A	01
0	1	1	!A	10
0	0	0	A	00
1	0	1	A	00
1	1	0	!A	10
1	1	1	!A	10
1	1			



13)


CKT
A
16
8
HI
8
LO


Evenly DIVISIBLE BY 8
A(7:0)
A(2:0)
A(1:0)
8
1
8
0
8
HI
A
8
ODD PAR
Gen
A(15:8)
8
1
8
LO
0
Sign Bit
A(15)

CHAPTER 20 EXERCISES

1) STATE REFERS TO THE VALUE THAT A CIRCUITS MEMORY ELEMENTS ARE CURRENTLY STORING

2) FEEDBACK FROM THE CIRCUIT'S OUTPUT TO THE CIRCUIT'S INPUT

3) THE FORBIDDEN STATE IS FORBIDDEN ~~BECAUSE~~ THE CROSS COUPLED CELL'S OUTPUT WILL NOT BE COMPLIMENTARY

$$\left.\begin{matrix} Q=1 \\ \overline{Q}=0 \end{matrix}\right\}$$ complimentary

$$\left.\begin{matrix} Q=0 \\ \overline{Q}=0 \end{matrix}\right\}$$ not complimentary

4) YOUR CIRCUIT WILL NOT EXPLODE; YOUR CELL STARTS ACTING IN STRANGE WAYS NOT DEFINED BY THE CELL DEFINITION

5) verb: to SET is to make a value '1'

Exaple: THE SIGNAL "SETS" THE CELL

NOUN: DESCRIBES THE CURRENT VALUE OF SOMETHING AS BEING '1'

Exaple: 'THE CELL'S OUTPUT IS "SET"

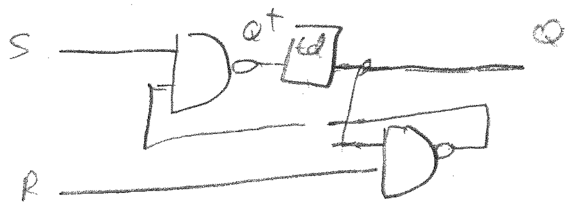
6) VERB: TO CLEAR is to make a value '0'

Example: THE SIGNAL "Clears" THE CELL

NOUN: DESCRIBES THE CURRENT value OF SOMETHING AS BEING '0'

Exaple: THE cell's output is "cleared"

7)



A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

S	R	Q	Q+	
0	0	0	1	} FORBIDDEN
0	0	1	1	
0	1	0	1	} SET
0	1	1	1	
1	0	0	0	} RESET
1	0	1	0	
1	1	0	0	} HOLD
1	1	1	1	

8) "Reset" is generally THE SAME WORD AS Clear

9) TIMING Diagrams SHOW MANY Different STATES. The only WAY you COULD DESCRIBE OR SHOW A NEXT state IS TO DECLARE ONE Displayed state as THE present state

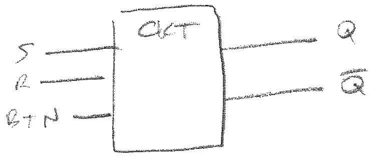
10) STATE DIAGRAMS ARE GOOD IF THEY QUICKLY TRANSFER INFORMATION to THE READER. making state diagrams NEAT, USING self COMMENTING LABELS, AND FOLLOWING ACCEPTED conventions will make state diagrams better.

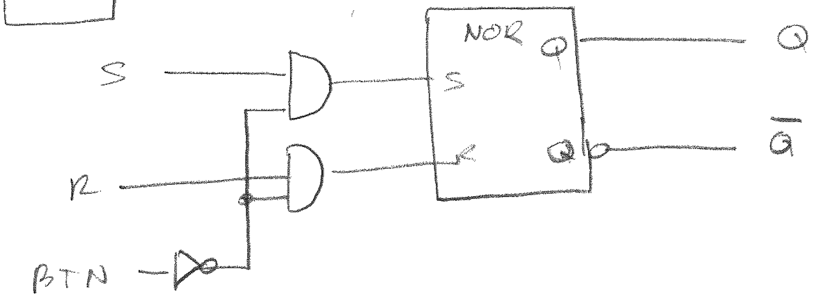
11) FEEDBACK FROM THE OUTPUT TO THE INPUT GIVES THE CIRCUIT MEMORY.

12) COMBINATORIAL CIRCUITS HAVE NO MEMORY CAPABILITIES AND THUS HAVE NO STATE.

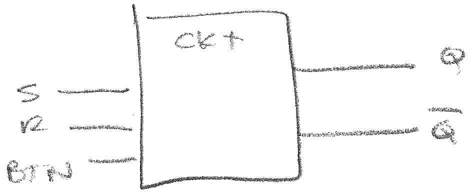
CHAPTER 20 DESIGN PROBLEMS

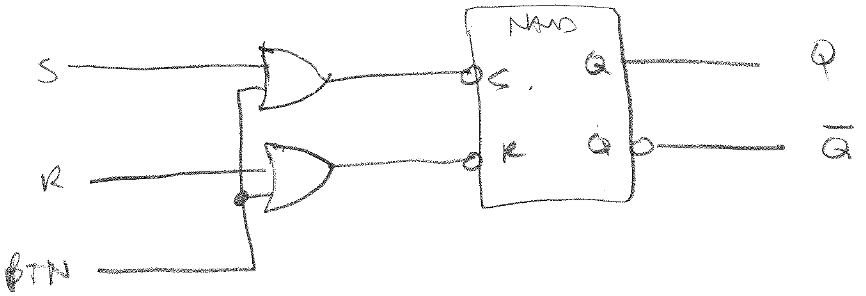
1)



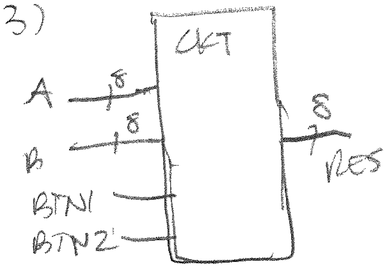


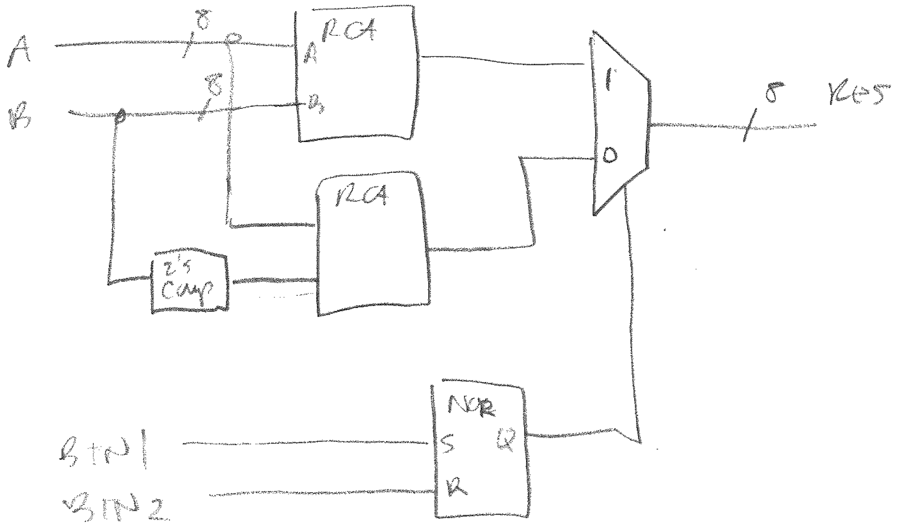
2)





3)



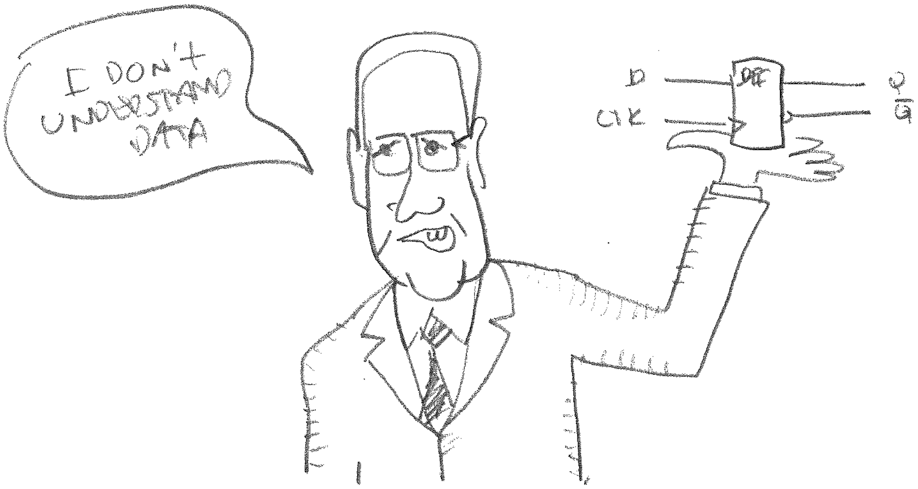


Chapter 21 Exercises

1) A latch is level sensitive while a flip-flop is edge sensitive. Sensitivity refers to what controls the flip flop's outputs. The out put of latches can change any time the inputs change while the output of flip flops can only change on an active clock edge

2) Asynchronous inputs can only change the flip flop's outputs any time they are asserted, while synchronous inputs can only change the flip flops' outputs at the same instance as the active clock edge

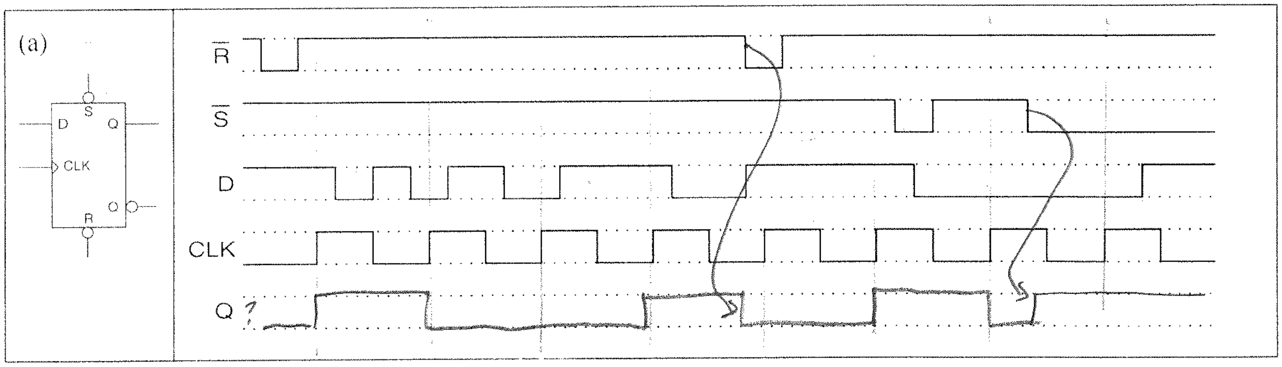
3) The D in D flip-flop stands for "Data"

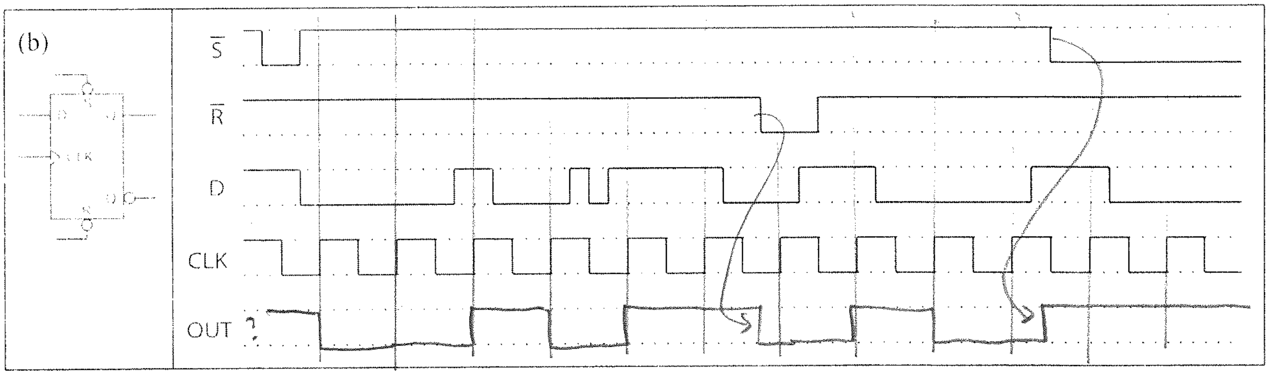


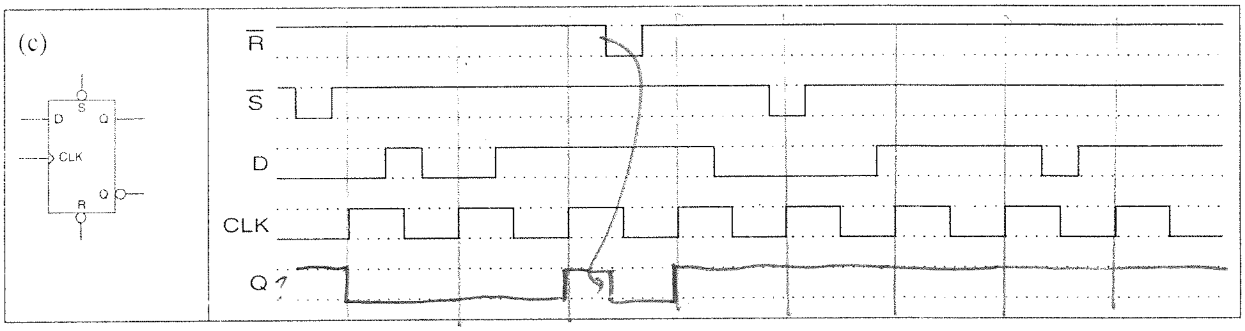
2

Chapter Exercises

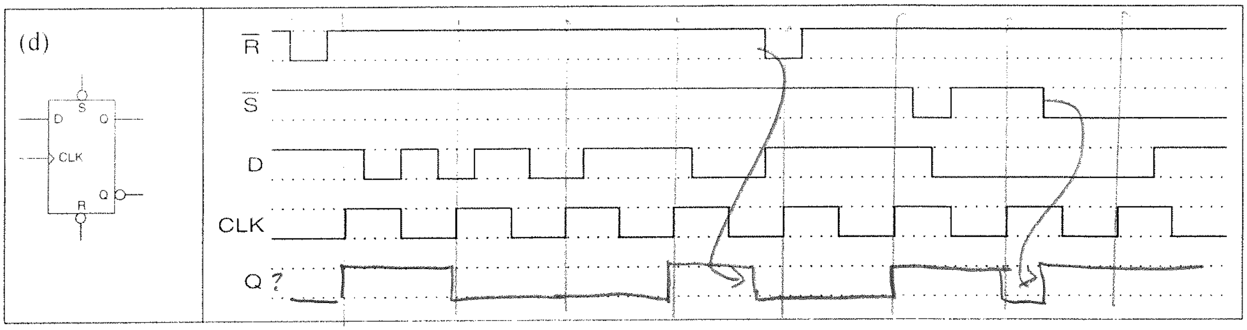
4) Provide the Q output (sometimes labeled as OUTPUT) signal using the associated flip-flops listed below. Consider all S and R inputs to be asynchronous. The asynchronous inputs take precedence over the synchronous inputs. Assume that propagation delays are negligent.



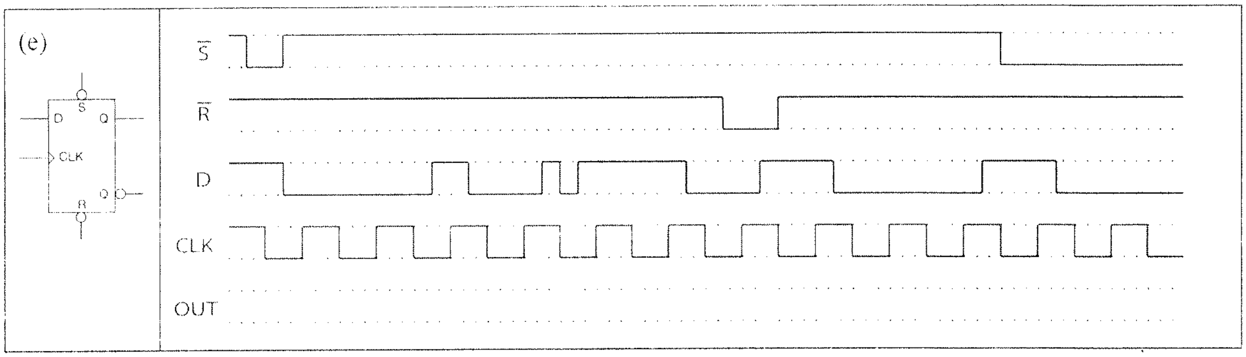




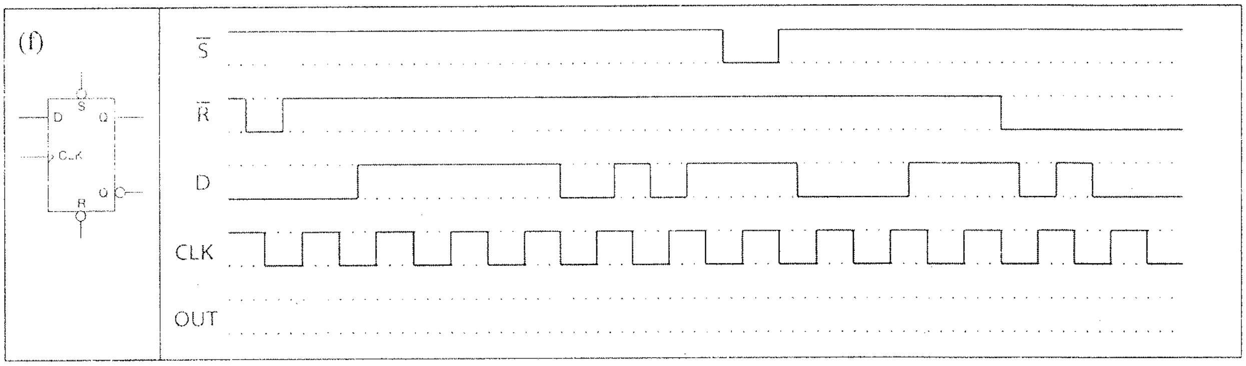
(d)



(e)



(f)

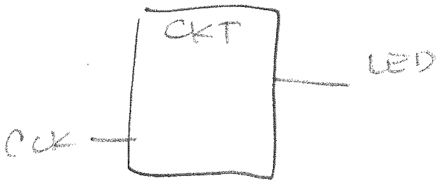


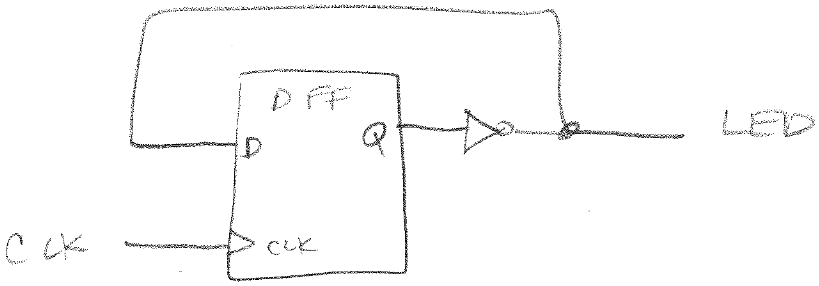
6) YOU MUST FIRST DIE BEFORE YOU CAN DESCRIBE THE KARMIC POTENTIAL OF D FFs. SO, ASK AN ACADEMIC ADMINISTRATOR... THEY HAVE NO SOULS AND THEREFORE MUST BE SOME TYPE ON NON-LIVING ENTITY.

7) IF YOU SQUINT YOUR BRAIN JUST RIGHT, THE ANSWER TO THIS QUESTION WILL BECOME CLEAR.

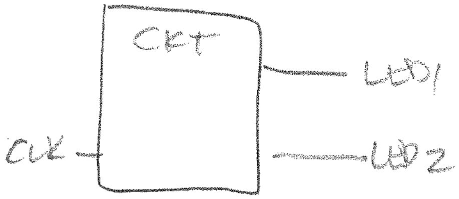
CHAPTER 21 DESIGN PROBLEMS

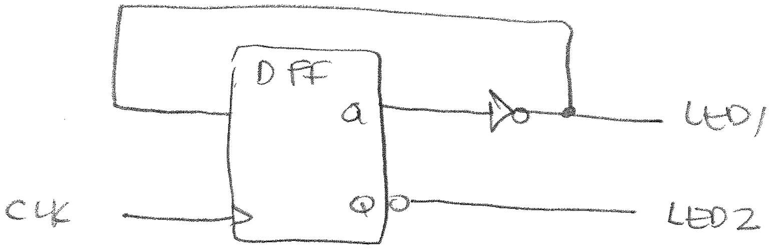
1)



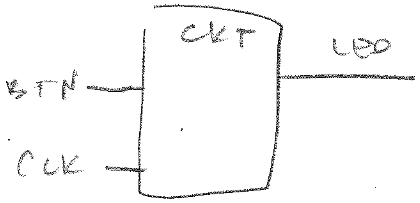


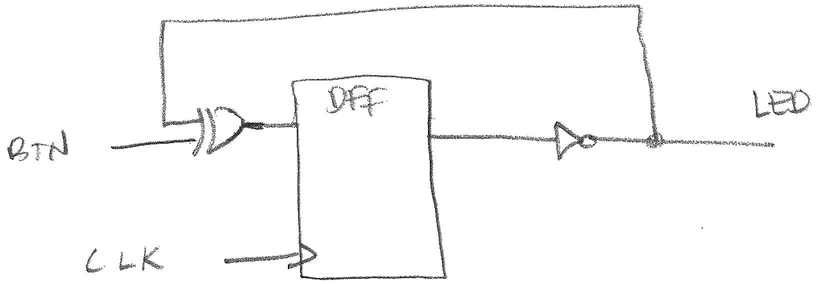
2)



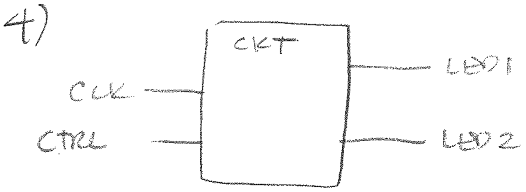


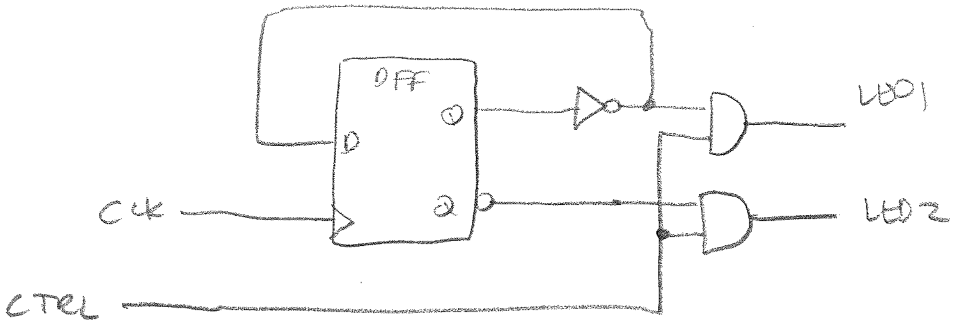
3)



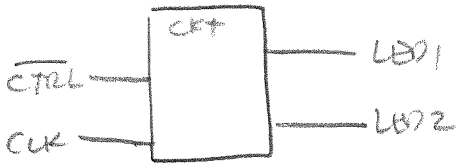


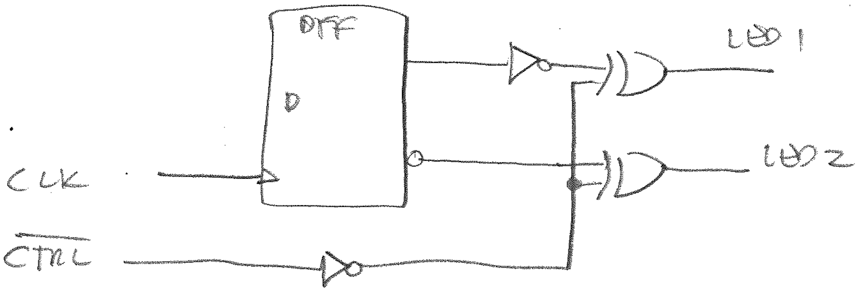
4)


CKT
CLK
CTRL
LED1
LED2

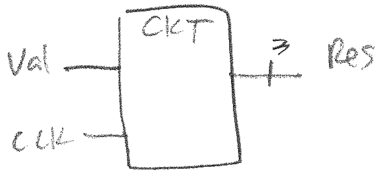

DFF
D
Q
Q
CLK
CTRL
LED1
LED2

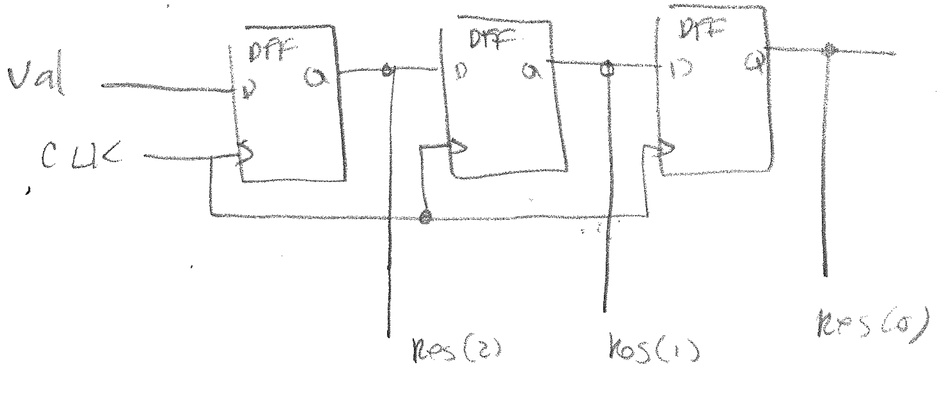
5)


CKT
$\overline{CTRL}$
CLK
LED1
LED2

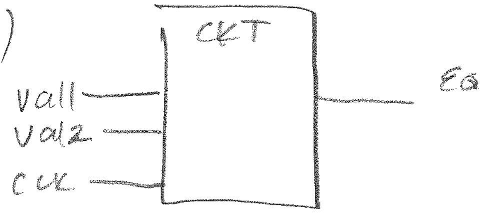

DFF
D
CLK
$\overline{CTRL}$
LED1
LED2

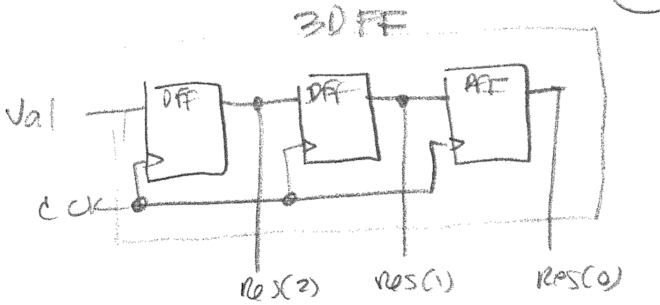
6)

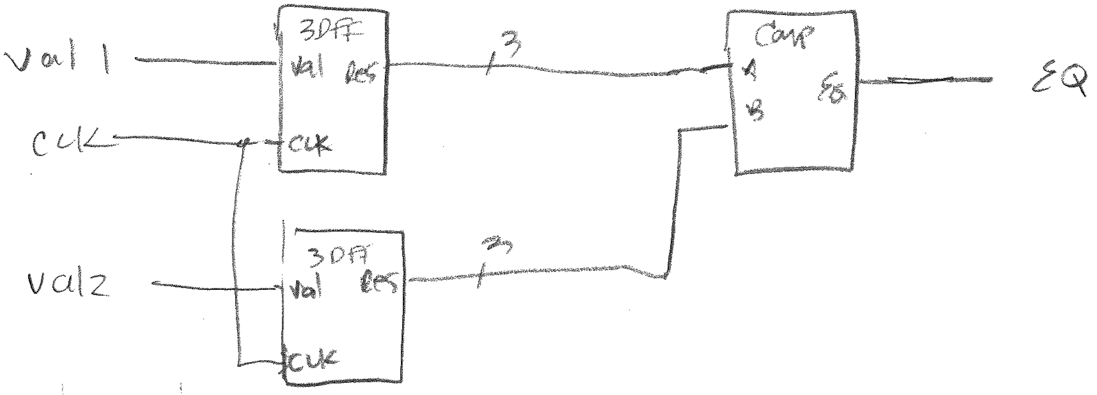

CKT
Val
CLK
3
Res


DFF
D
Q
DFF
D
Q
DFF
D
Q
Val
CLK
Res(2)
Res(1)
Res(0)

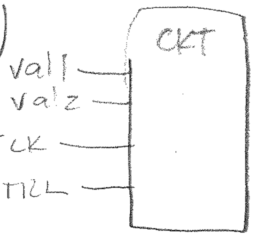
7)

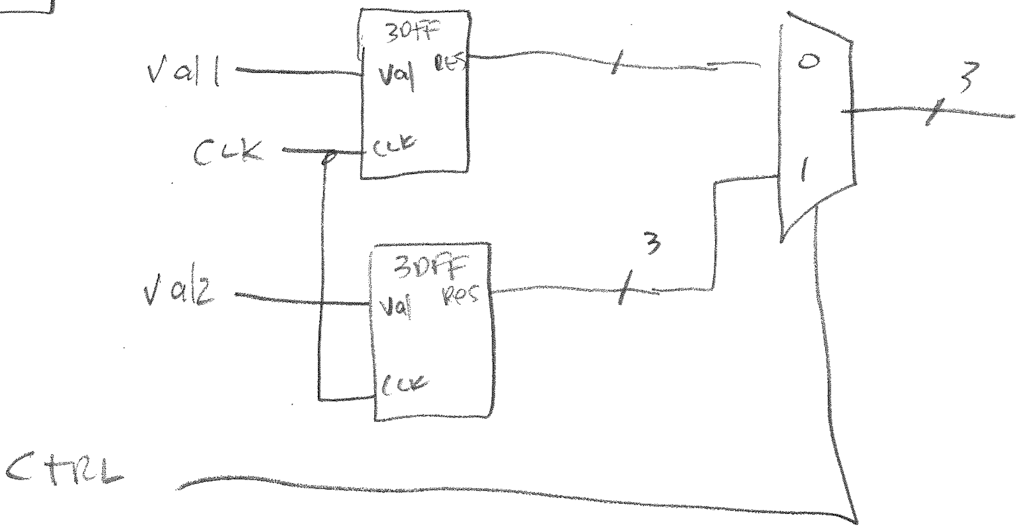

CKT
val1
val2
CLK
EQ


3DFF
Val
CLK
DFF
DFF
DFF
Res(2)
Res(1)
Res(0)


Val1
CLK
Val2
3DFF
Val
Res
CLK
3DFF
Val
Res
CLK
3
3
Comp
A
B
EQ
EQ

8)

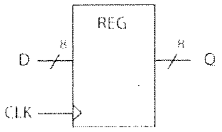

CKT
val1
val2
CLK
CTRL

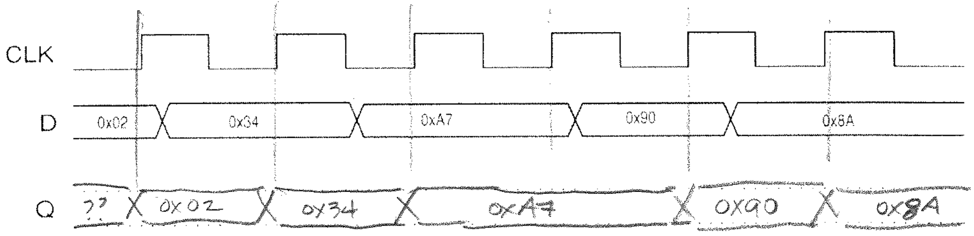

Val1
CLK
Val2
CTRL
3DFF
Val
Res
CLK
3DFF
Val
Res
CLK
3
0
1
3

1

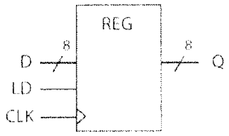
Chapter Exercises

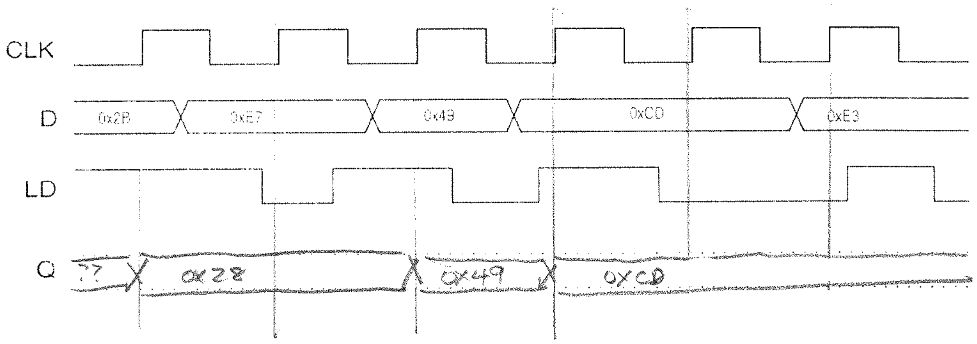
1) Using the block diagram on the right to complete the timing diagram provided below. Consider the register to be rising-edge triggered and ignore all propagation delay issues.





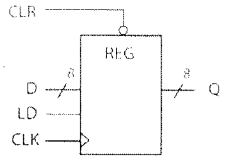
2) Using the block diagram on the right to complete the timing diagram provided below. The LD input must be asserted in order for the register to load the input signal. Consider the register to be rising-edge triggered and ignore all propagation delay issues.

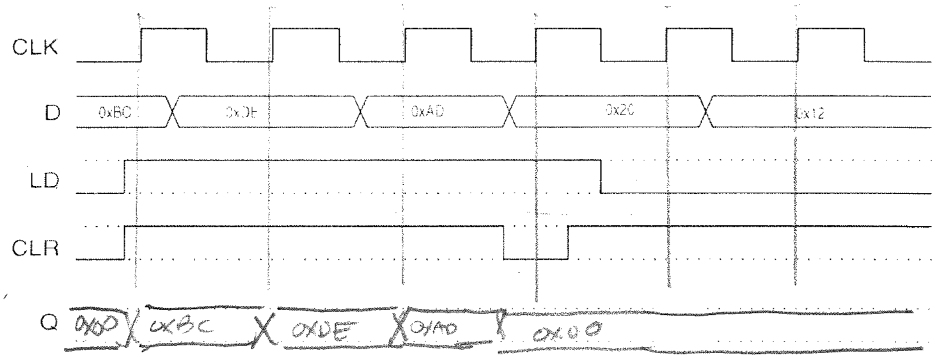




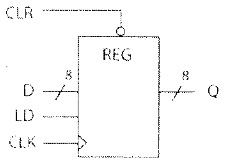
2

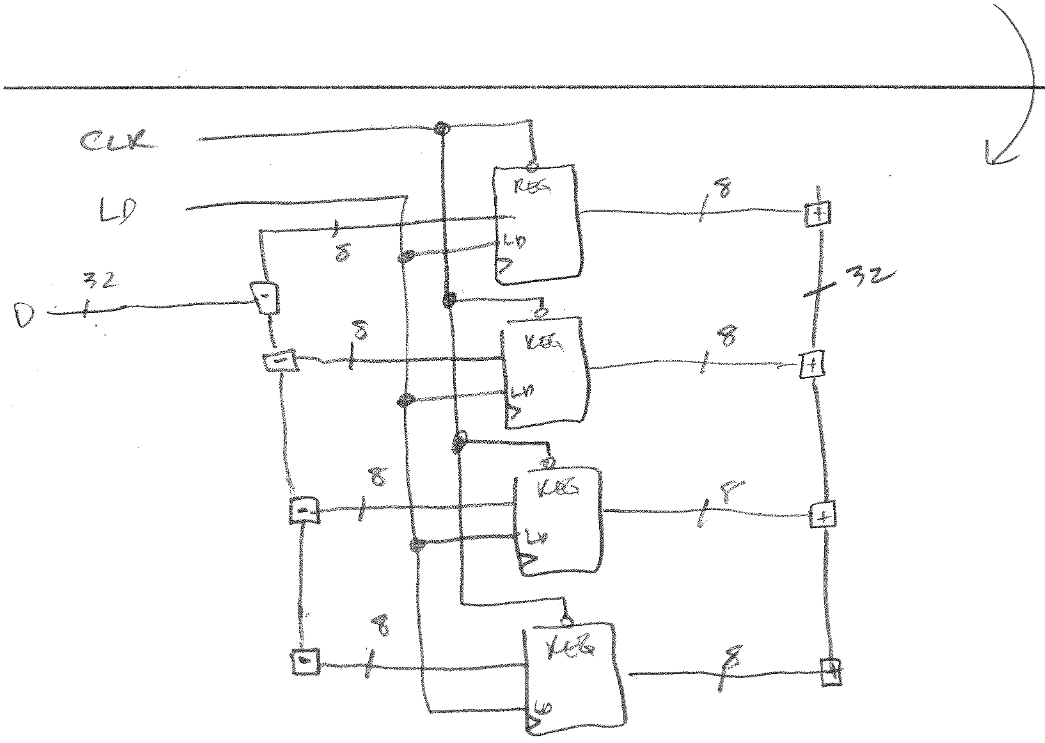
3) Using the block diagram on the right to complete the timing diagram provided below. The LD input must be asserted in order for the register to load the input signal. The CLR input is an asynchronous input that clears the register when asserted and has a higher precedence than the LD input. Consider the register to be rising-edge triggered and ignore all propagation delay issues.





4) Using the block diagram on the right, provide a schematic diagram detailing how you would use this device to create a 32-bit register with all the same features listed on the 16-bit device.





5) THE REGISTER IS A COMPONENT BECAUSE THE ACCUMULATOR value REQUIRES TEMPORARY STORAGE IN ORDER TO DO THE ACCUMULATION OR RUNNING TOTAL, OF THE values BEING ADDED

6) WITHOUT A REGISTER, THE RCA's output (THE RESULT) WOULD IMMEDIATELY Be ADDED AGAIN to THE RCA. THE OUTPUT would change AS FAST AS THE NEW INPUT COULD PROPAGATE THROUGH THE RCA

7) WHEN You Accumulate A SET OF values, You Generally START A ZERO; THE CLEAR control input on THE REGISTER PROVIDES THAT ZERO.

8) FOUR = 2^2 ; THE TOTAL NUMBER OF Bits would then Be

$$2^2 \cdot 2^{20} = 2^{22}$$

so... <u>22</u> Bits

9) $16 = 2^4$

$2^4 \cdot 2^{18}$ is MAX value so

2^{22} , <u>22 Bits</u>

10) 13 ⇒ its Greater than 8
But less than 16, which is 2^4

so $2^4 \cdot 2^{11} = 2^{15}$

<u>15 Bits</u> REQUIRED;
14 Bits' is too few

11) 17 is greater than 16 But less than 32

So... $2^5 \cdot 2^7 = 2^{12}$

<u>12 Bits REQUIRED</u>

12) 13 bit output ADDING 5 Bit values

$$13 = \log_2 [\text{value} \cdot 2^5]$$

$$2^{13} = \text{value} \cdot 2^5$$

$$\text{value} = \frac{2^{13}}{2^5} = \underline{\underline{2^8}} = 256 \text{ values}$$

13) 20 bit output TWO 8-bit NUMBERS

$$20 = \log_2 [\text{vale} \cdot 2^8]$$

$$2^{20} = \text{value} \cdot 2^8$$

$$\text{value} = 2^{20}/2^8 = \underline{\underline{2^{12} = 4096 \text{ NUMBERS}}}$$

14) 39-bit output

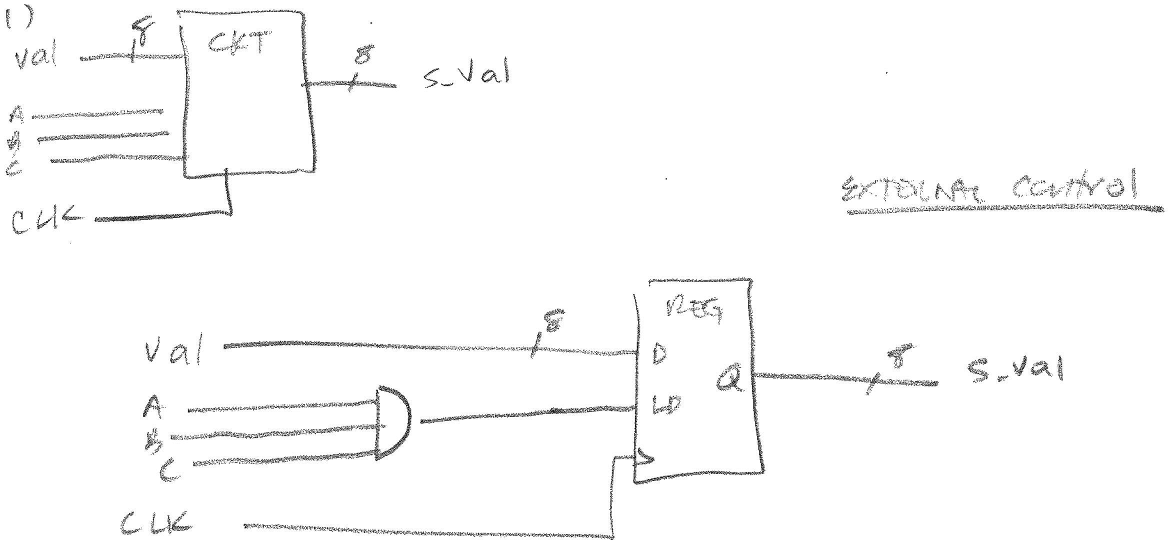
$$39 = \log_2 \left[\text{value} \cdot 2^{16} \right]$$

$$2^{39} = \text{value} \cdot 2^{16}$$

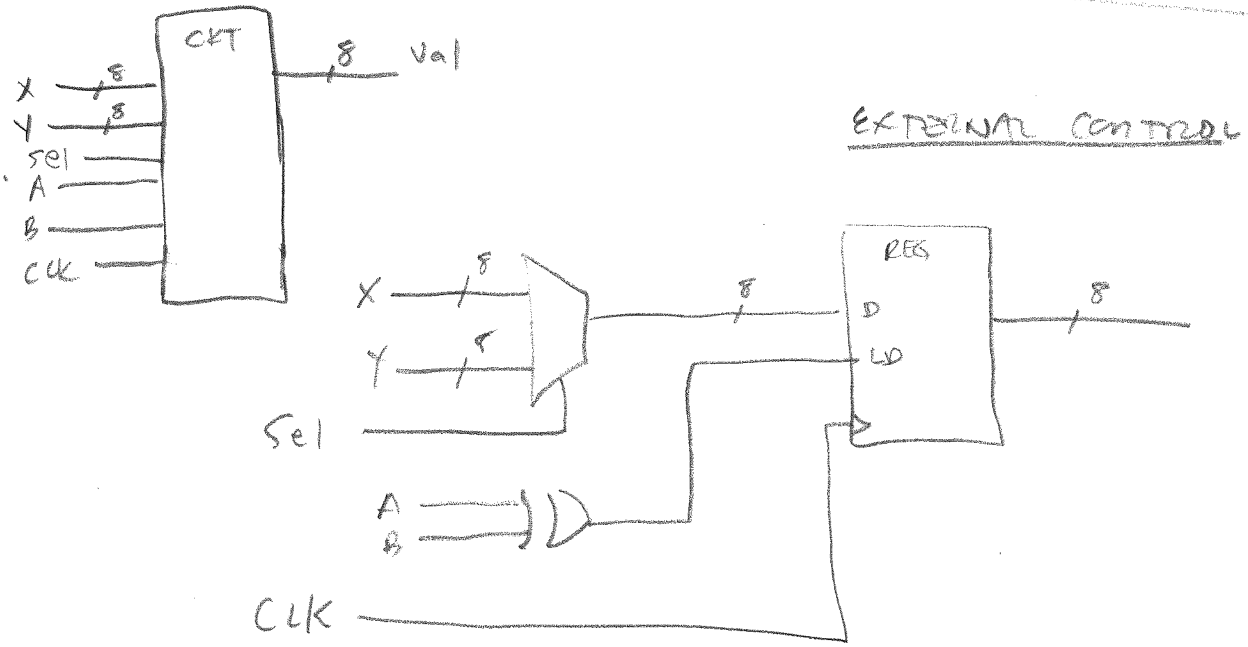
$$\text{value} = \frac{2^{39}}{2^{16}} = \underline{\underline{2^{25} \text{ values}}}$$

CHAPTER 22 DESIGN PROBLEMS

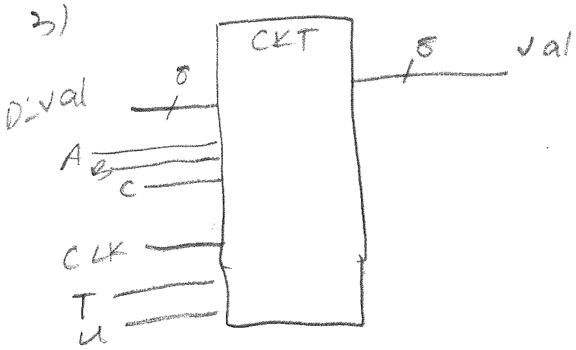
1)



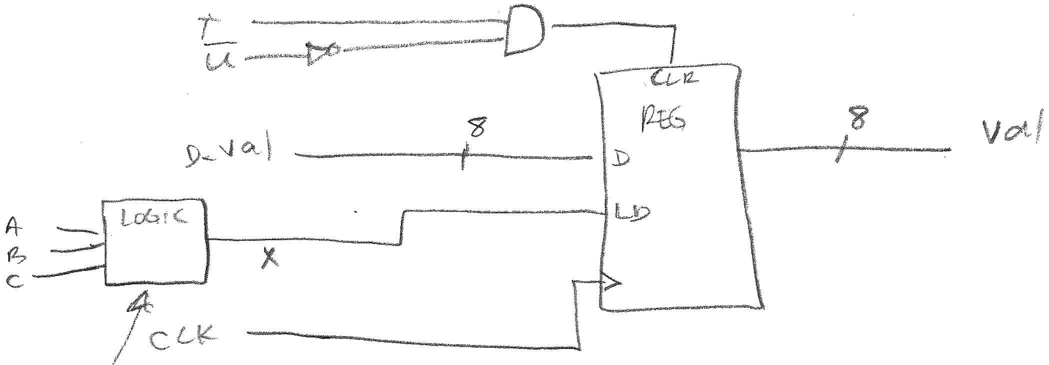
2)



3)



EXTERNAL CONTROL



LOGIC : $X = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$

Chapter 23 Exercises (Answers)

1) Briefly explain the general purpose of a state diagram.

A state diagram provides a quick visual description of a FSM.

2) Briefly explain why do individual states in state diagrams have unique, self-commenting labels.

The states in FSMs have unique labels to differentiate them from other states. The states in FSM use self-commenting labels to help humans more quickly understand the operation of the FSM.

3) Briefly explain why we typically omit clock signals from state diagrams.

We omit clock signals from state diagrams to make them easier to understand. The assumption made by state diagrams is that all transitions occur synchronized to a clock edge. Transitions that are not synchronized with a clock edge must have other information included so that the human reader knows how they operate.

4) Briefly explain why we label unconditional transfers with some type of "don't care" symbol.

The don't care signal alerts the human reader of the FSM that a given transition happens unconditionally, meaning the given transition is not dependent on any external or internal signal.

5) Briefly explain why PS/NS tables don't include clock signals.

PS/NS tables don't include clock signals for two reasons. First, it would be really had to do so, and if you figured out a cool way, the PS/NS table would become harder to understand. Second, most transitions in FSM are synchronous, so the clock signal in the PS/NS table is implied.

6) Briefly explain how we represent asynchronous signals in state diagrams.

Asynchronous signal are singly directed arrows that are directed to a state in the state diagram but do not emanate from a state in the state diagram These arrows effectively come out of "nowhere", which differentiates them from synchronous transitions.

7) Briefly explain the main function of an FSM' next-state decoder.

The next-state decoder provides the logic that determines the next state of the FSM. Inputs to the NS decoder are typically the current state of the FSM and external inputs. Outputs of the NS decoder provide the excitation inputs for the state registers.

8) Briefly explain the main function of an FSM's output decoder.

The output decoder provides the logic that determines the exact form of the FSM's outputs. The output decoder typically has inputs of the state of the FSM and external inputs. Output decoders can implement two different types of outputs: Mealy-type outputs and Moore-type outputs.

9) Briefly explain the main purpose of an FSM's state registers.

The FSM's state registers are the FSM's memory and thus hold the "state" of the FSM. Thus, the state of the FSM is officially the values being stored in the state registers.

10) Briefly explain the different between Moore and Mealy-type outputs on FSMs.

Moore-type outputs are a function of the FSM's state only. Mealy-type outputs are a function of the FSM's state and at least one external input.

11) Briefly describe why it is most convenient to not place Mealy-type outputs in the state bubbles.

We place Mealy-type outputs outside of the bubble because they are a function of both state and external inputs. Because of this, we place the Mealy-type outputs next to the external input(s) that determine the conditions associated with the FSM's state transitions.

12) Briefly describe with it is most convenient to place Moore-type outputs in the state bubbles.

We place the Moore-type outputs in the state bubble because Moore-type outputs are strictly a functions of the state of the FSM.

13) Briefly explain what is meant by the term "unused state" in an FSM.

An unused state in a FSM is a state that could be represented by the FSM's state registers but is not included in the FSM's state diagram.

14) Briefly explain what is meant by the term "hang state" and how an FSM can end up in a hang state.

A hang state is a state in a FSM is an unused state that is not necessarily include in the FSM's state diagram. Hang states have not have a designed method for returning to any of the used states in the FSM.

15) Briefly explain the difference between a hang state and an unused state in an FSM.

An unused state may or may not be able to transition back to a used state in the FSM's state diagram. A hang state is not able to purposely transition back to a used state in the FSM's state diagram.

16) Briefly explain the main strategy behind designing FSM to be self-correcting.

A FSM that is self correctly has all unused states (if any) purposely directed back to a state currently used in the FSM's state diagram. In this way, if the FSM finds itself in a used state, it will transition back to a used state.

17) Briefly explain why some FSM designs inherently do not have hang states.

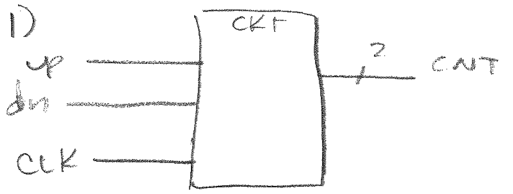
Some FSMs do not have inherently have hang states because there are no unused states in the FSM. This means that every possible combination of the FSM's state variables represent a valid state in the FSM.

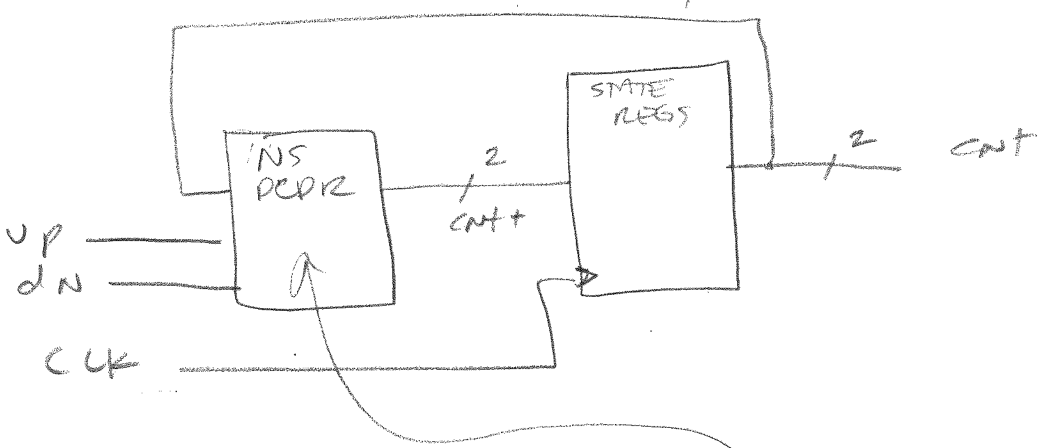
---

18) IF THE CONDITIONS WERE NOT MUTUALLY EXCLUSIVE, THE SAME SET OF CONDITIONS COULD BE ASSOCIATED WITH TWO DIFFERENT TRANSITIONS; IN THIS CASE, THE FSM WOULD NOT KNOW THE CORRECT TRANSITION TO TAKE.

CHAPTER 23 DESIGN PROBLEMS

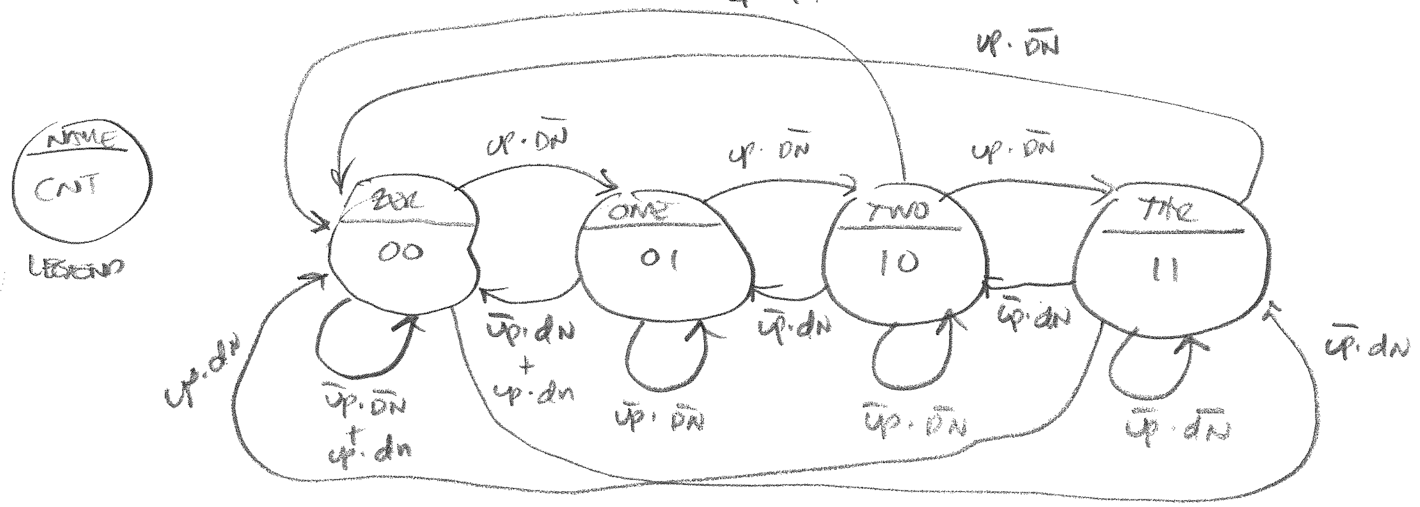
1)



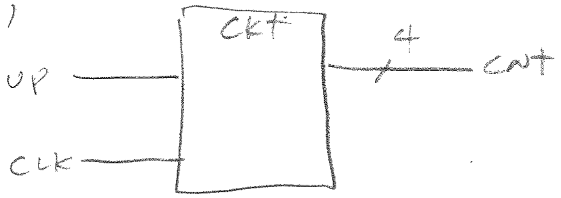


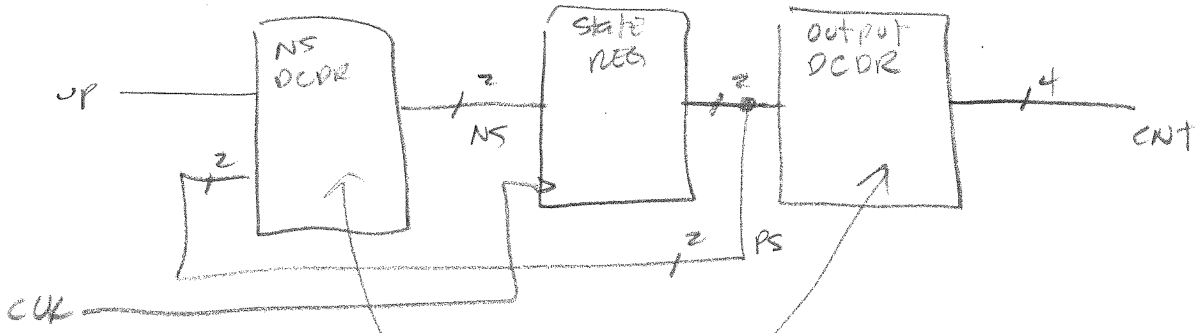
up	dn	CNT		CNT+	
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	1	1
0	1	0	1	0	0
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	1
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

TABLE DEFINES DECODER



21)





NS Decoder

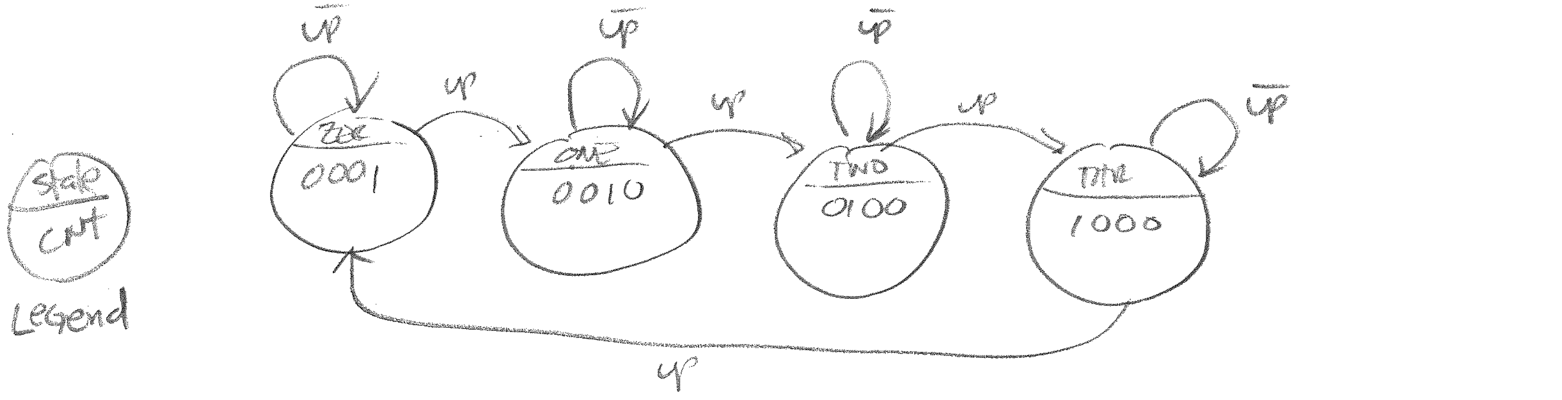
UP	PS	NS
0	0 0	00
0	0 1	01
0	1 0	10
0	1 1	11
1	0 0	01
1	0 1	10
1	1 0	11
1	1 1	00

Defines Decoder

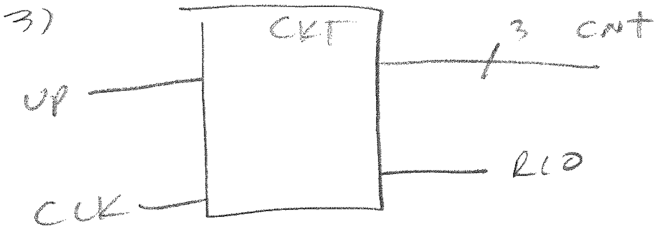
Output Decoder

PS	CNT
0 0	0001
0 1	0010
1 0	0100
1 1	1000

which is of course a standard decoder

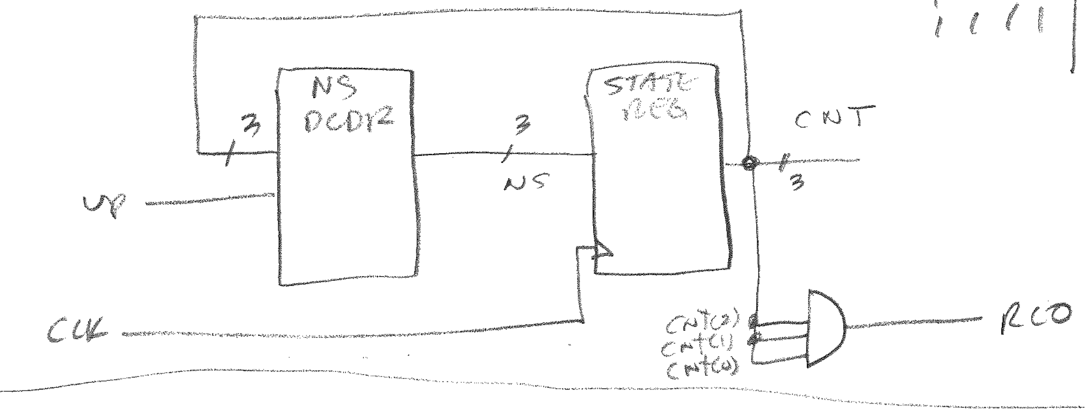


3)

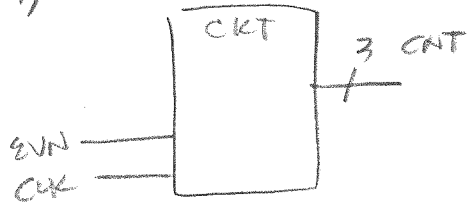


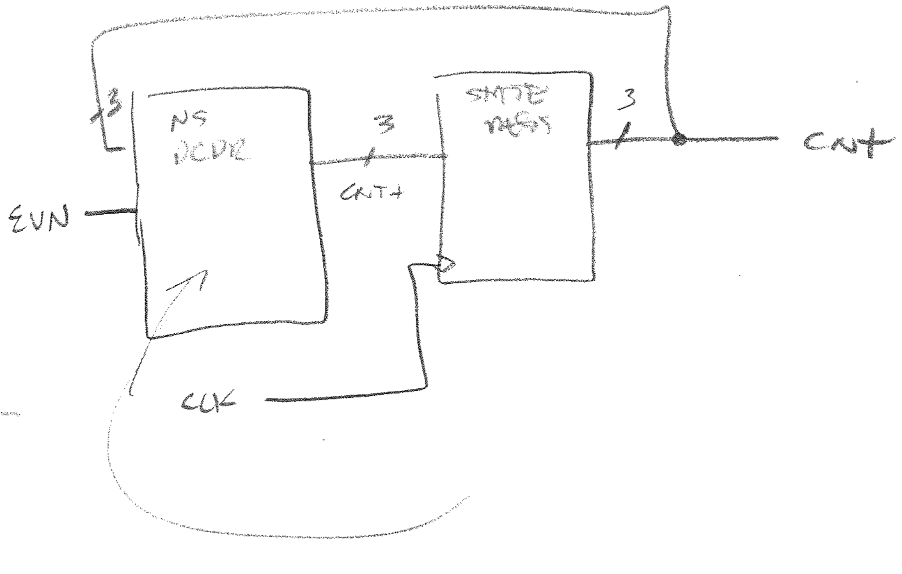
up	CNT			NS
0	0	0	0	000
0	0	0	1	001
0	0	1	0	010
0	0	1	1	011
0	1	0	0	100
0	1	0	1	101
0	1	1	0	110
0	1	1	1	111
1	0	0	0	001
1	0	0	1	010
1	0	1	0	011
1	0	1	1	100
1	1	0	0	101
1	1	0	1	110
1	1	1	0	111
1	1	1	1	000

Defines NS Decoder



4)

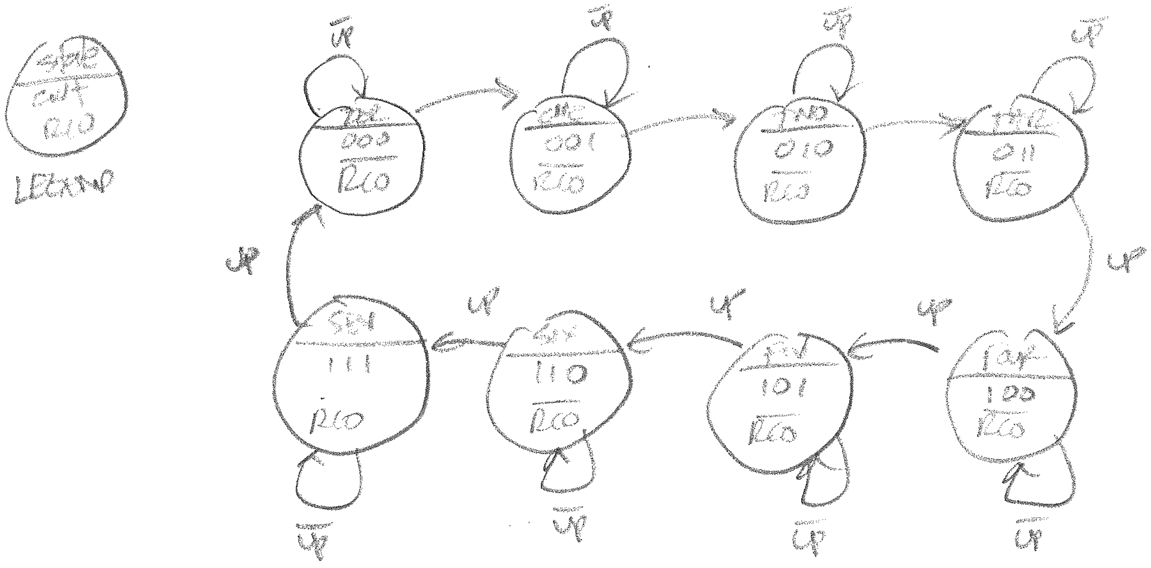




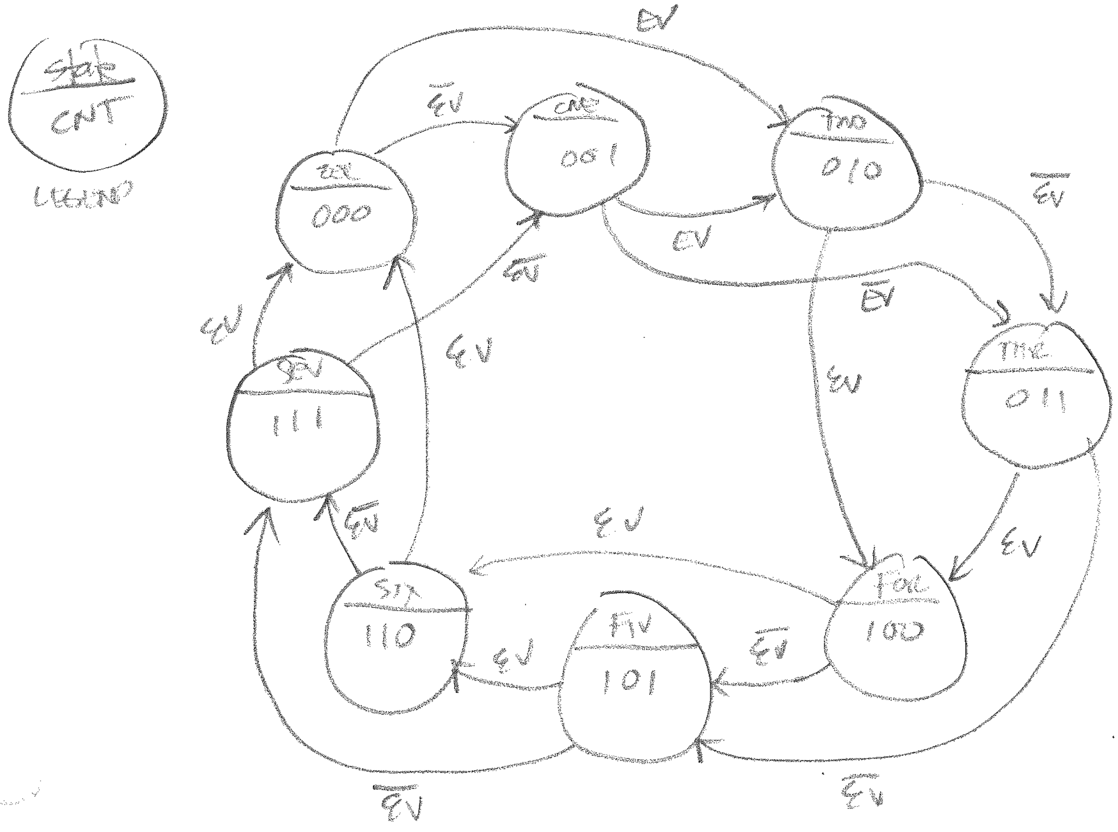
EVN	CNT			CNT+
0	0	0	0	001
0	0	0	1	011
0	0	1	0	011
0	0	1	1	101
0	1	0	0	101
0	1	0	1	111
0	1	1	0	111
0	1	1	1	001
1	0	0	0	010
1	0	0	1	010
1	0	1	0	100
1	0	1	1	100
1	1	0	0	110
1	1	0	1	110
1	1	1	0	000
1	1	1	1	000

← Defines NS Decoder

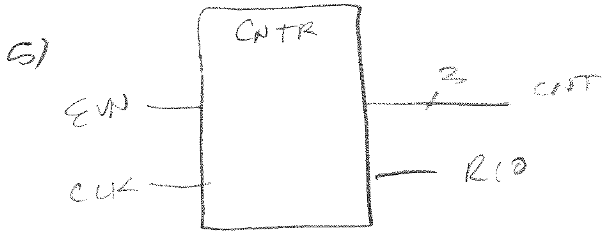
3) State diagram



4) State Diagram

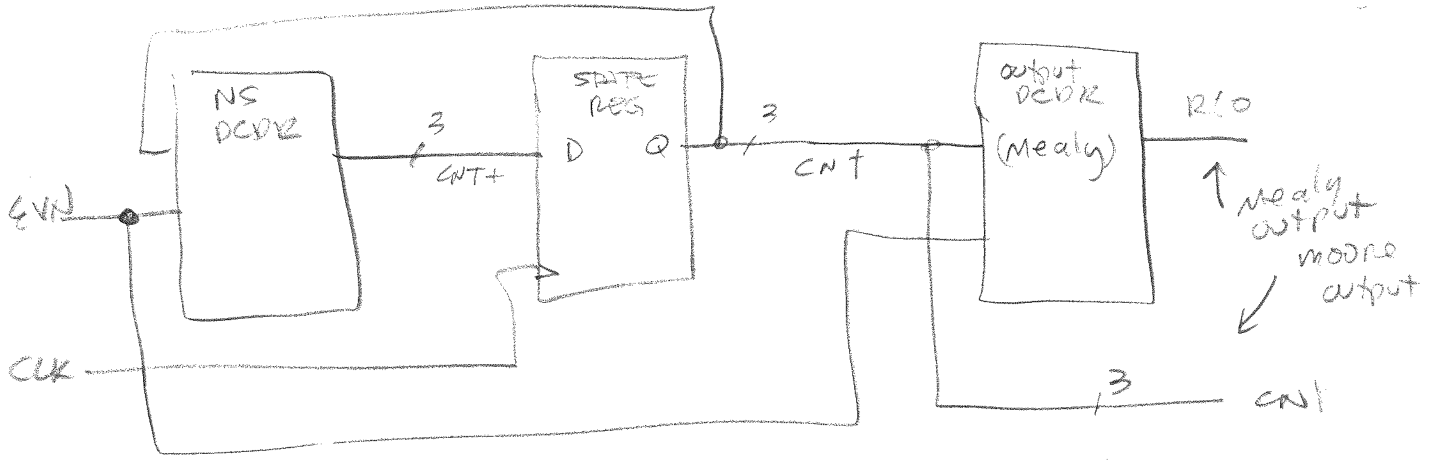


5)

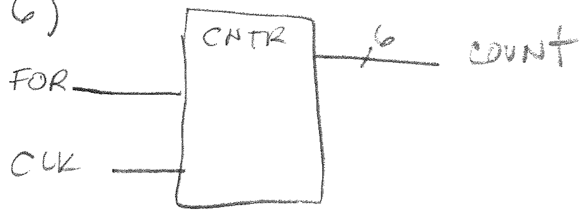


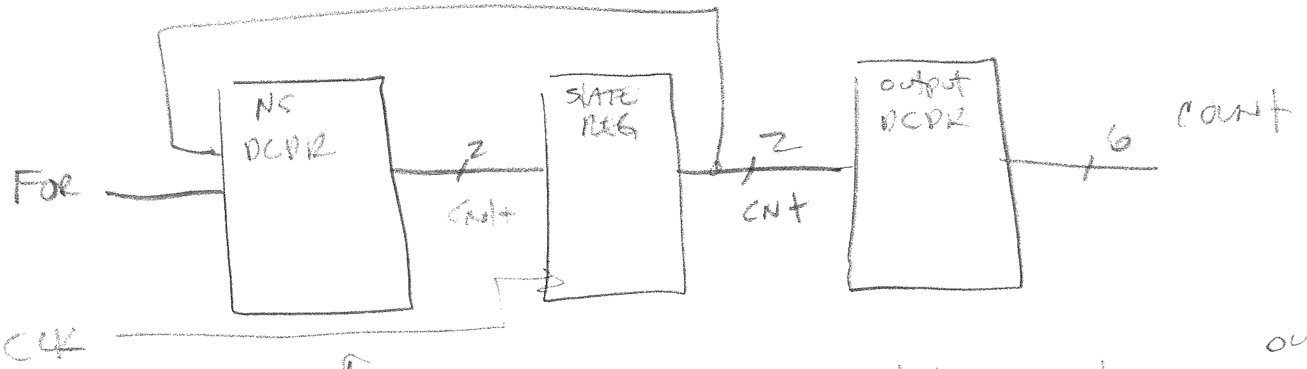
EVN	CNT	CNT+	RCO
0	000	001	0
0	001	011	0
0	010	011	0
0	011	101	0
0	100	101	1
0	101	111	1
0	110	111	0
0	111	001	0
1	000	010	0
1	001	010	0
1	010	100	0
1	011	100	0
1	100	110	0
1	101	110	1
1	110	000	1
1	111	000	0

Defines both NS decoder and output decoder



6)





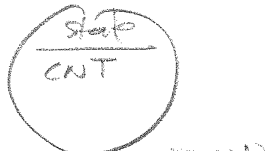
FOR	CNT	CNT+
0	00	00
0	01	01
0	10	10
0	11	11
1	00	01
1	01	10
1	10	11
1	11	00

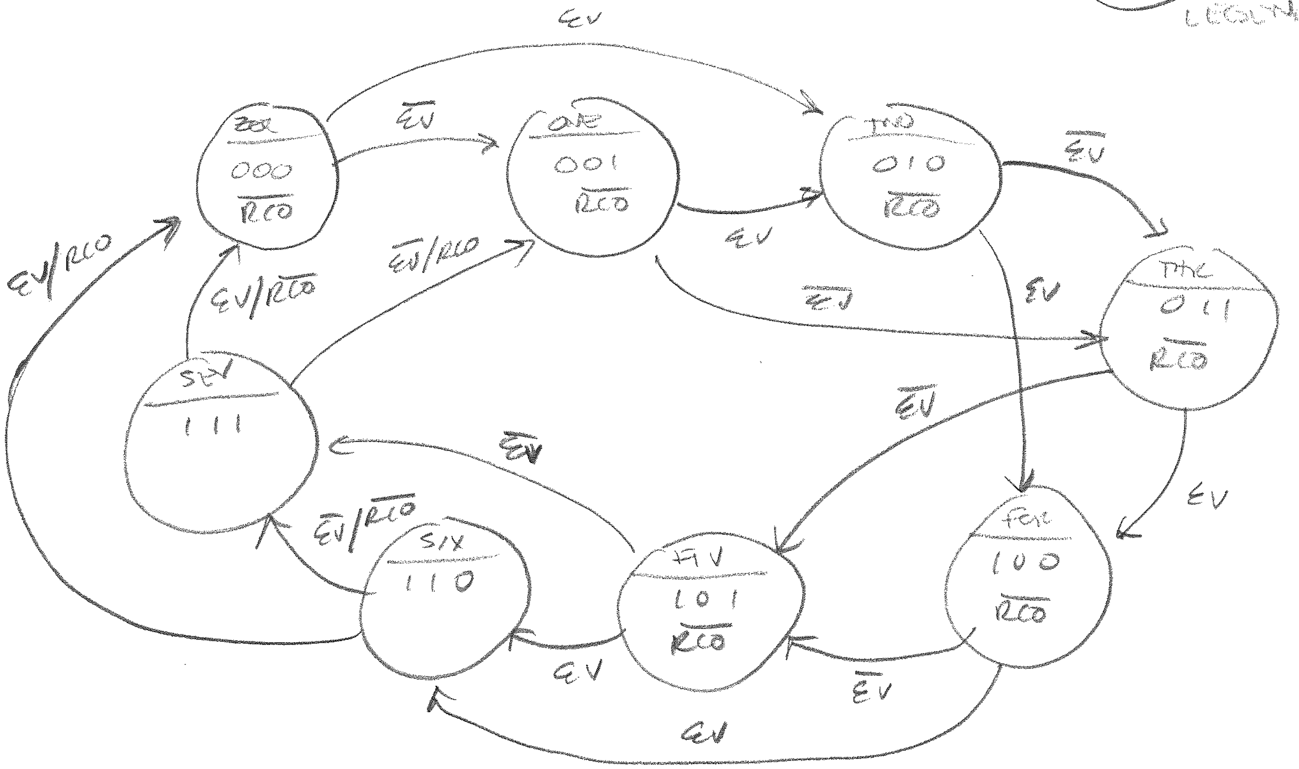
NS Decoder

CNT	COUNT
00	100001
01	110010
10	011000
11	101010

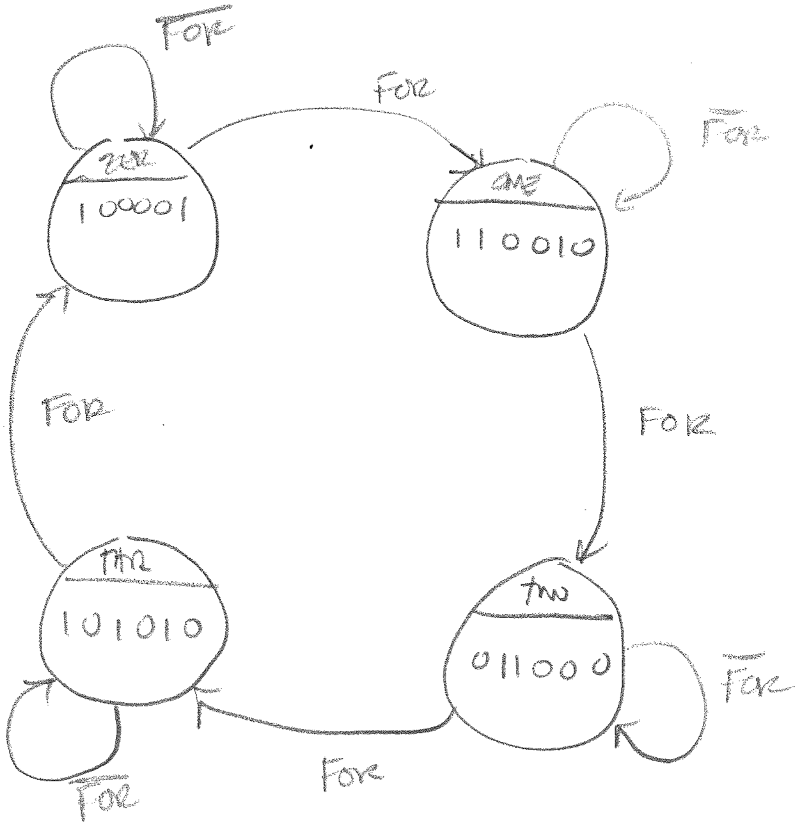
output Decoder definition

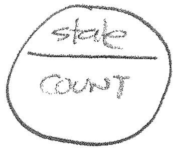
5) STATE DIAGRAM



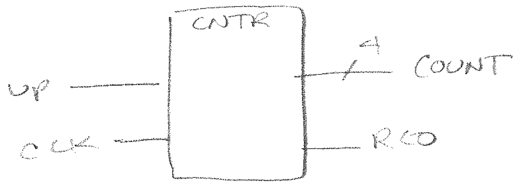


6)

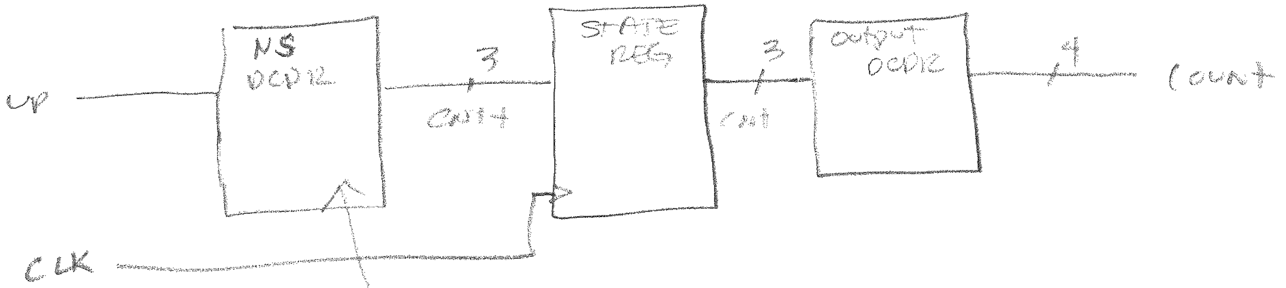




7)



count sequence has 6 unique values, so we need 3-bits of storage in the state registers, we'll use 0→5 to represent the six count values



up	cnt	cnt+
0	000	000
0	001	001
0	010	010
0	011	011
0	100	100
0	101	101
0	110	000
0	111	000
1	000	001
1	001	010
1	010	011
1	011	100
1	100	101
1	101	110
1	110	000
1	111	000

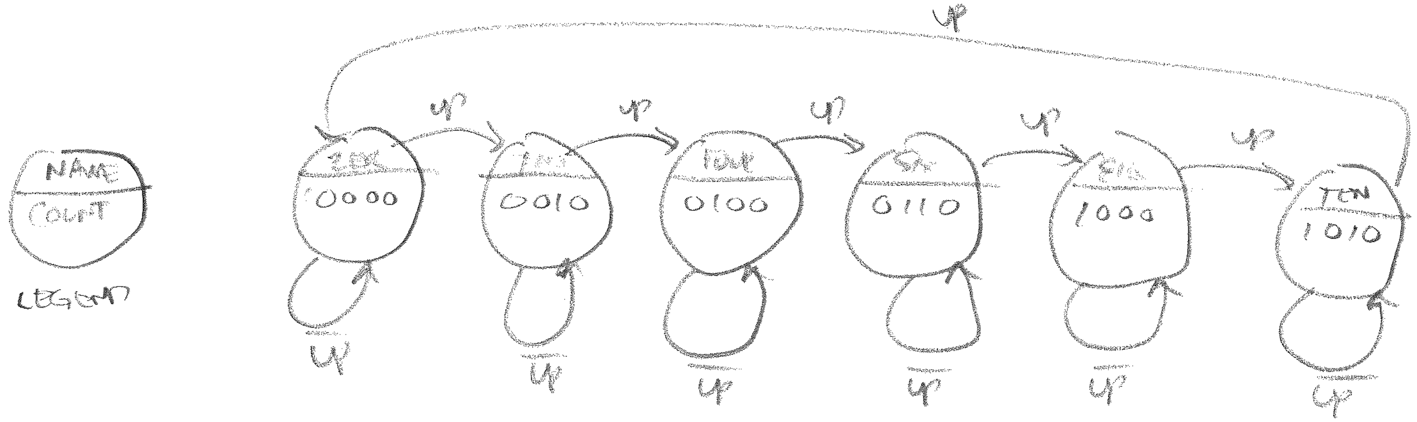
Defines NS Decoder

} For self correction

} for self correction

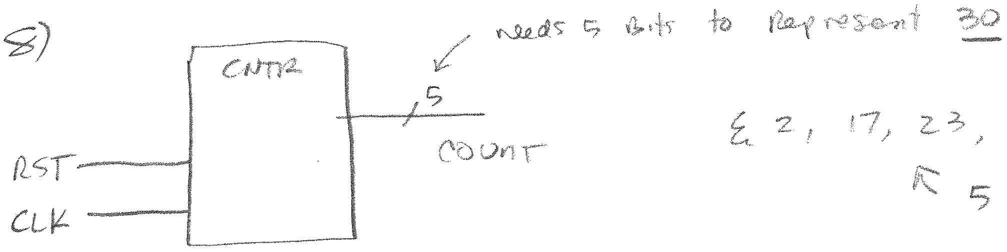
CNT	COUNT
000	0000
001	0010
010	0100
011	0110
100	1000
101	1010
110	0000
111	0000

Defines output Decoder



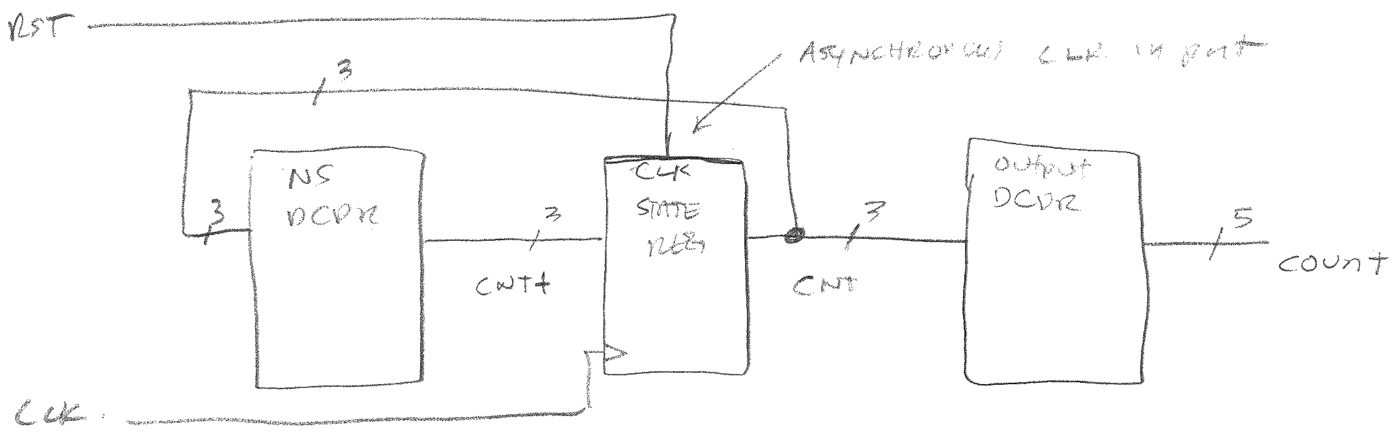
state diagram

8)



{ 2, 17, 23, 11, 30 }

↖ 5 value so we need 3 Bits to Represent



NS DECODER

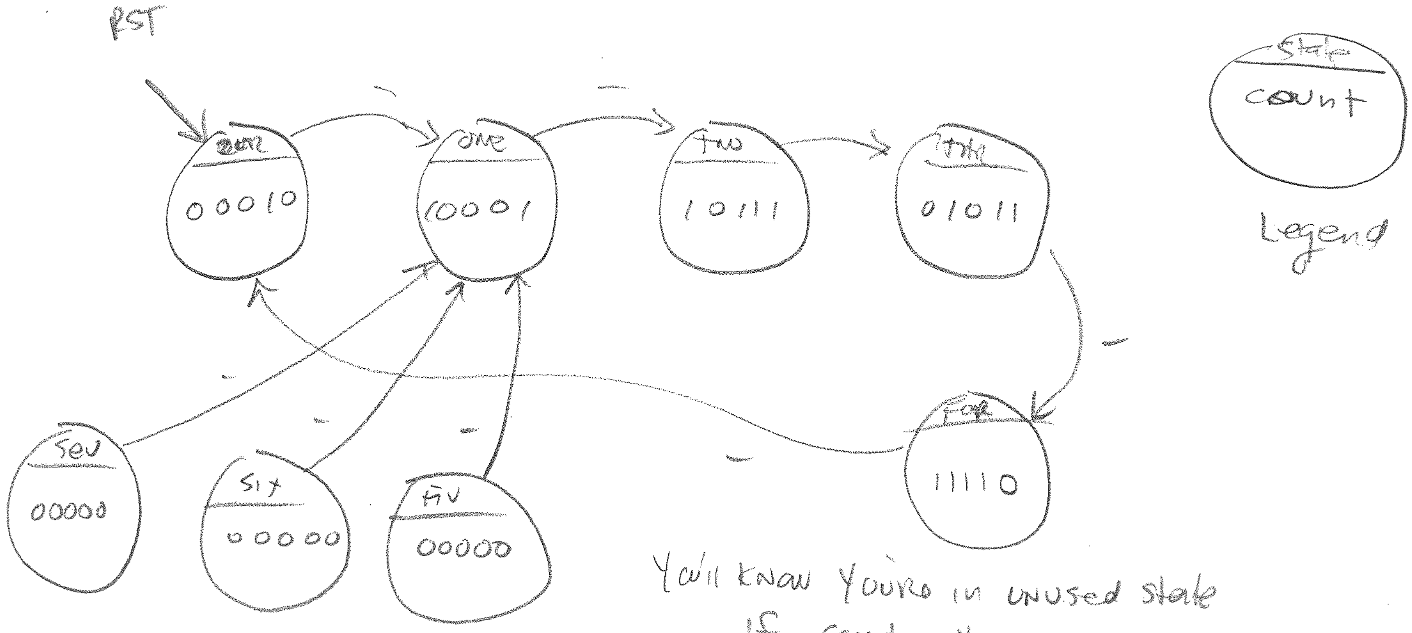
CNT	CNT+
000	001
001	010
010	011
011	100
100	101
101	010
110	010
111	010

} self-correction

output Decoder Definition

Cnt	count
000	00010
001	10001
010	10111
011	01011
100	11110
101	00000
110	00000
111	00000

} UNused states



You'll know You're in UNused state if count = "00000"

CHAPTER 24 EXERCISES

1) a) A PERIODIC signal REPEATS itself AFTER A GIVEN Period OF TIME

1) b) PERIOD IS THE AMOUNT OF TIME REQUIRED FOR A PERIODIC SIGNAL TO REPEAT IT SELF (MEASURED IN SECONDS

c) THE FREQUENCY IS THE NUMBER OF TIMES A PERIODIC signal REPEATS itself in A GIVEN AMOUNT OF TIME

d) THE DUTY CYCLE IS THE PERCENTAGE OF A PERIOD THAT A PERIODIC SIGNAL IS IN THE HIGH STATE

2) NON PERIODIC signal DO NOT HAVE constant PERIODS SO THE NOTION OF DUTY CYCLE MAKES NO SENSE.

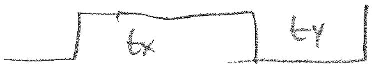
3) $t_x = 30ns \quad t_y = 25ns$

Period $= \underline{55ns}$

Frequency $= 1/55_{ns} \quad = 0.018 \times 10^{9}\ s^{-1} = \underline{18.2\ MHz}$

Duty cycle $= \dfrac{30ns}{55ns} = \underline{54.5\%}$

4)



$$\frac{t_x}{t_x + t_y} = DC$$

$$t_x = DC\,(t_x + t_y)$$

$$\frac{t_x}{DC} - t_x = t_y$$

$$\frac{14ns}{.7} - 14ns = t_y = \underline{6ns}$$

$$Freq = \frac{1}{6ns} = 0.167 \times 10^{-9}$$

$$= 167\ MHz$$

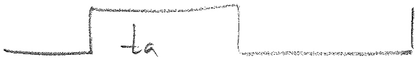
5) FREQ = 50 MHZ

PERIOD = $1/50 \times 10^6$ = 0.020×10^{-6} s = $\underline{20\text{ ns}}$

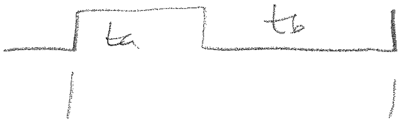
DC = 0.4

t_a = 0.40 × 20ns

= $\underline{8\text{ns}}$



6) DC = 20%



t_b = 20ns

$$\frac{t_a}{t_a+t_b} = 0.2$$

$$t_a = 0.2(t_a+t_b)$$

$$\frac{t_a}{.2} - t_a = t_b$$

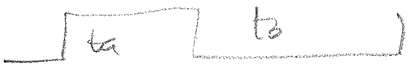
$$4t_a = t_b$$

$$t_a = \frac{t_b}{4} = \frac{20\text{ ns}}{4} = \underline{5\text{ns}}$$

Period = 25ns

FREQ = 40 MHZ

7)



DC = .4

$$\frac{t_a}{t_a+t_b} = 0.4$$

$$t_a = 0.4(t_a+t_b)$$

$$t_a - 0.4t_a = t_b \Rightarrow 20\text{ns} - 0.4(20\text{ns}) = t_b$$

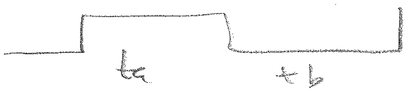
$$t_b = 12\text{ns}$$

Period = 32ns

FREQ = 31.3 MHZ

DC = 0.40

8)



DC= 80%

$\frac{ta}{ta + tb} = DC$

$ta = DC(ta + tb)$

$ta - DC\,ta = tb$

$ta \; .8\,ta = tb$

$.2\,ta = 60\,ns$

$ta = 300\,ns$

Period= 360 ns

FREQ = 2.78 MHZ

9)

$t_{SU} = 6ns$

$t_{FF} = 20ns$

$t_{NS} = 2(10ns) = 20ns$

$20ns + 20ns + 6ns = 46ns$

min PERIOD = 46ns + 0.1(46ns)

min Period = 50.1ns

MAX FREQ = 19.96 MHZ

(10) 46ns + 20ns = 76 ns = min PERIOD

= 13.2 MHZ

11) period = 1/F = 50 ns

min Period = $2(t_g) + t_{su} + t_{FF}$ = 18ns + 8ns + 17ns

= 43 ns

=

50 ns - 43ns = Safety margin

= 7ns

(2) min period = $t_{FF} + 2(t_g) + t_{su}$

= 20ns + 16us + 5us

= 41us

min period = 41us + 0.2 × 41ns

= 41_{ns} + 8.2ns = 49.2us

MAX FREQ = 20.3 MHz

Chapter ~~26~~ 25 Exercise Solutions

1) What is the minimum number of states in a state diagram you would need to obtain a 7/17 duty cycle on an external blinking LED? Briefly explain the reasoning behind your answer.

Answer: Since both numbers are prime number, you would need a state diagram with 17 states.

2) Briefly describe an application where a sequence detector would be useful.

Answer: digital combination locks, automatic resets for circuits

3) Briefly describe the operational difference between a FSM with a Moore-type output and a functionally equivalent FSM with a Mealy-type output. Consider both FSMs to have equivalent clock frequencies.

Answer: Mealy-type FSM typically have fewer states in the associated state diagram. FSMs with Mealy-type outputs can respond faster to given circuit conditions (referring to the fact that Mealy-type outputs are a function of both state and the external inputs).

4) Briefly describe two advantages to using a FSM exclusively Mealy-type outputs over a functionally equivalent FSM with exclusively Moore-type outputs.

Answer: FSM with Mealy-type outputs will naturally have less states than the equivalent Moore-type FSM, which may lead to less storage requirements in the state registers. The Mealy-type outputs can respond quicker to external outputs because the outputs can change immediately. An equivalent Moore-type output has to wait until the next clock cycle to respond.

5) We often consider FSMs as "reacting". In the context of controlling a digital circuit, briefly describe what we mean by "reacting". Be sure to describe what the FSM is reacting to and what the ramifications of these reactions do in a holistic view of the FSM.

Answer: We typically use FSMs as circuits that control other circuit. When we say "react", we mean that the output of a Mealy-type FSM can change immediately rather than having to wait for the next FSM active clock edge as is required by a Moore-type output.

6) Briefly explain why it is that FSMs with Mealy-type outputs can react faster than an equivalent FSM with Moore-type outputs.

Answer: We can think of Mealy-type outputs as reacting faster because they are a function of external inputs (meaning they can change when the external input changes). Moore-type outputs must wait for the next active clock edge before they can respond.

25.8 Chapter Exercises

1) What is the minimum number of states in a state diagram you would need to obtain a 7/17 duty cycle on an external blinking LED? Briefly explain the reasoning behind your answer.

2) Briefly describe an application where a sequence detector would be useful.

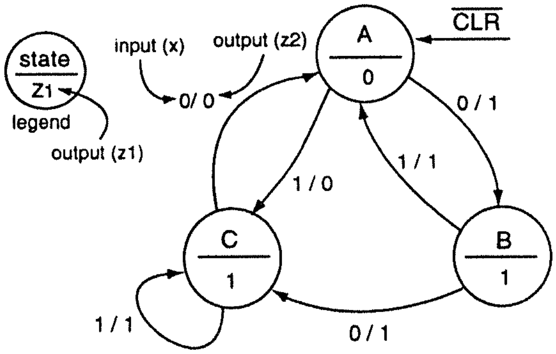
3) Briefly describe the operational difference between a FSM with a Moore-type output and a functionally equivalent FSM with a Mealy-type output. Consider both FSMs to have equivalent clock frequencies.

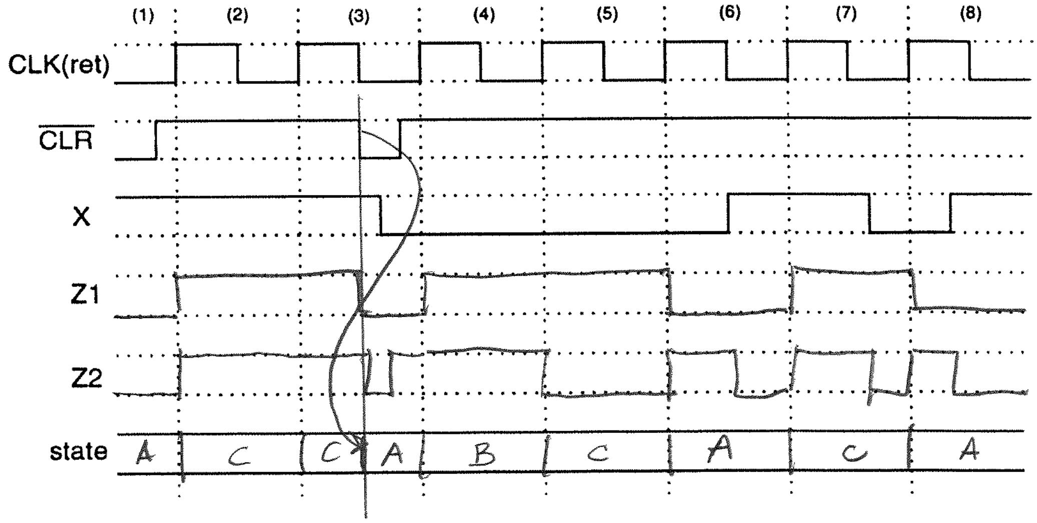
4) Briefly describe two advantages to using a FSM exclusively Mealy-type outputs over an functionally equivalent FSM with exclusively Moore-type outputs.

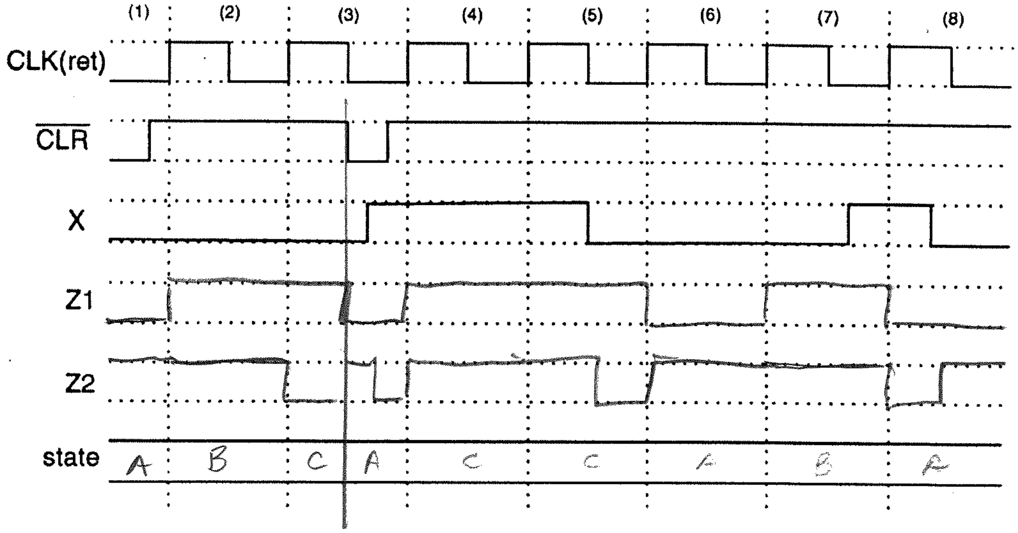
5) We often consider FSMs as "reacting". In the context of controlling a digital circuit, briefly describe what we mean by "reacting". Be sure to describe what the FSM is reacting to and what the ramifications of these reactions do in a holistic view of the FSM.

6) Briefly explain why it is that FSMs with Mealy-type outputs can react faster than an equivalent FSM with Moore-type outputs.

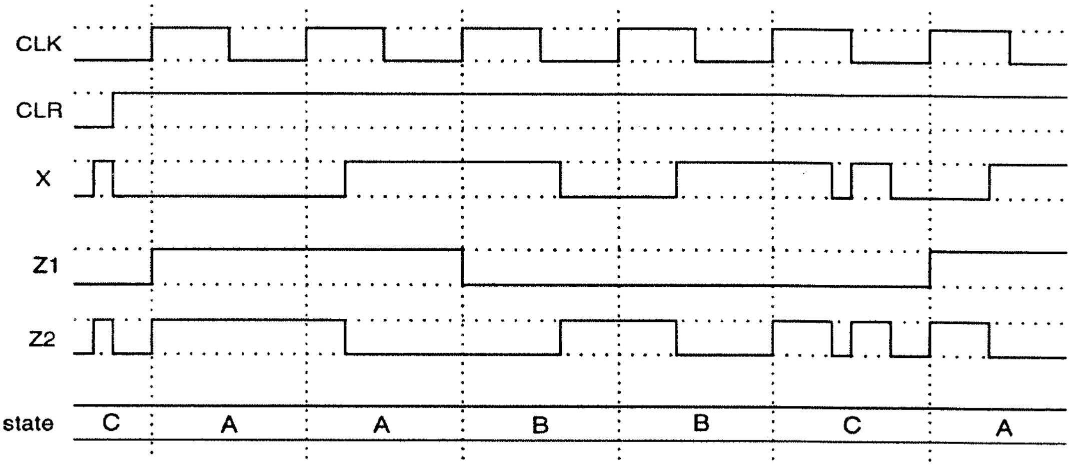
7) Use the following state diagram to complete the two timing diagram provided below. Show how the inputs affect the state transitions and outputs Z by filling in the "state" and "Z" lines in the timing diagram. Assume all setup and hold times are met and that propagation delay times are negligible. Assume state transitions occur on the rising edge of the clock signal. Assume CLR is an asynchronous, active low input.



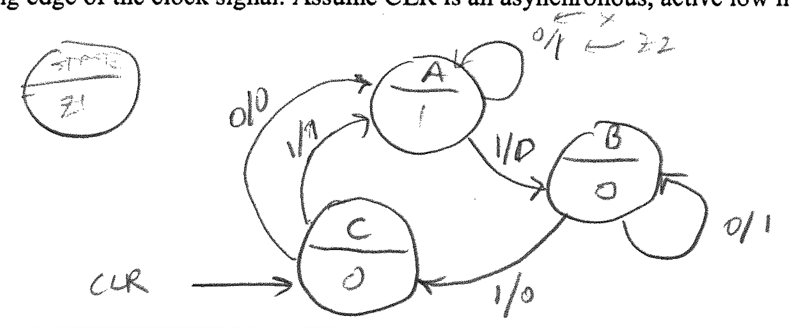




8) The following timing diagram completely specifies an FSM. Use the following timing diagram generate the state diagram that would generate the listed timing diagram. For this problem, assume the CLR input to be an asynchronous active low input that places the FSM into the appropriate state. Assume all setup and hold times have been met and that propagation delay times are negligible. Assume state transitions occur on the rising edge of the clock signal.

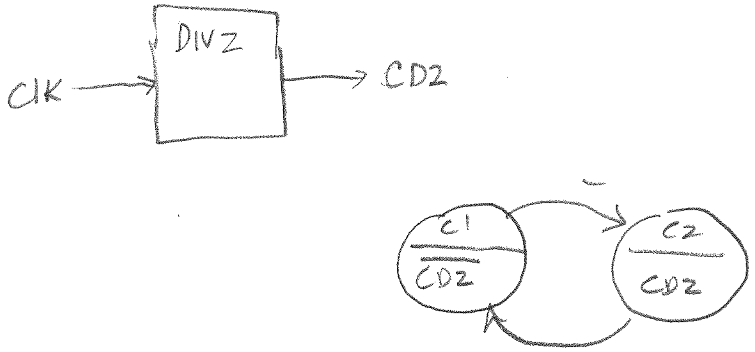


9) Use the following state diagram to complete the timing diagram provided below. Show how the inputs affect the state transitions and outputs Z by filling in the "state" and "Z" lines in the timing diagram. Assume all setup and hold times have been met and that propagation delay times are negligible. Assume state transitions occur on the rising edge of the clock signal. Assume CLR is an asynchronous, active low input.



CHAPTER DESIGN PROBLEM SOLUTIONS

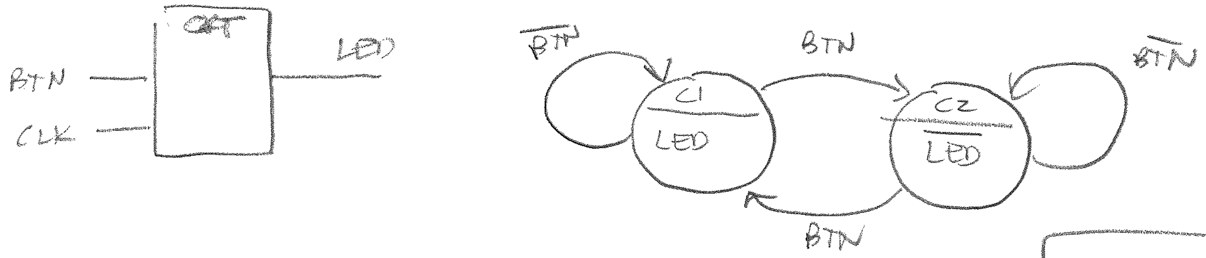
1)



CIRCUIT CONTROLLED

(FSM)

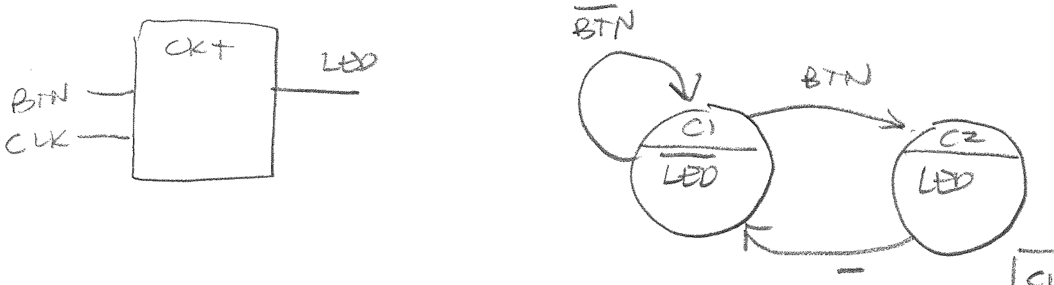
2)



CIRCUIT controlled

(FSM)

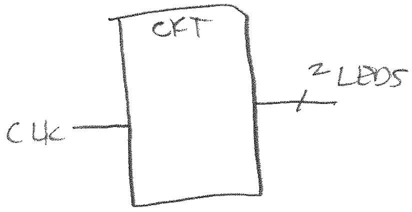
3)



CIRCUIT controlled

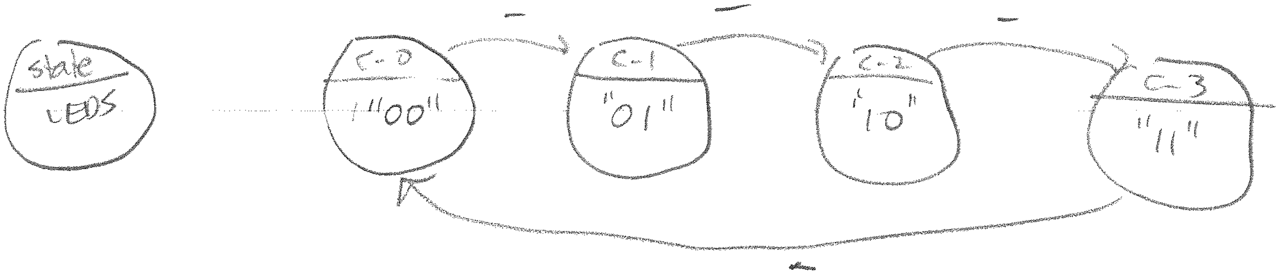
(FSM)

4)

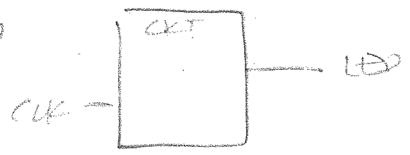


CIRCUIT controlled

(FSM)



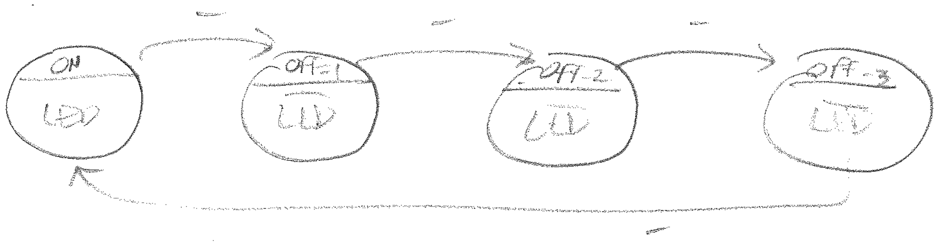
5)



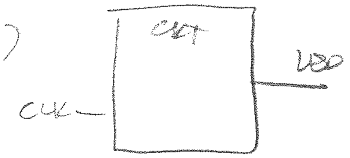
system clock = 4 x 40kHz =
160 KHz

CIRCUIT Controlled

MOORE outputs

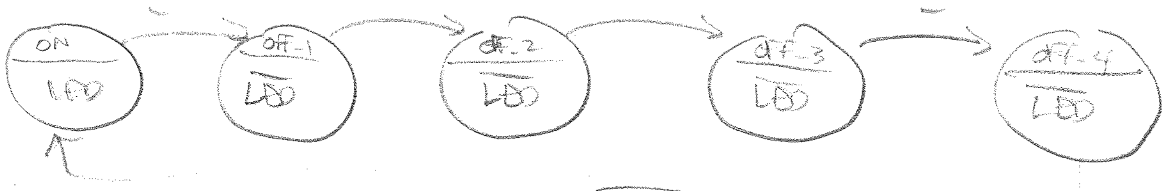


6)



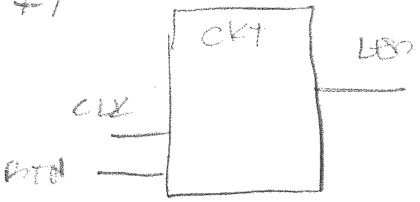
system clock = 5 x 100 Hz
= 500 Hz

MOORE outputs

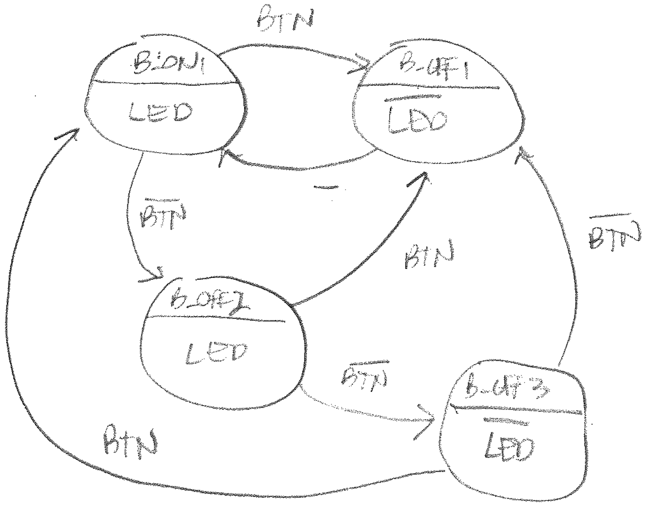


CIRCUIT controlled

7)



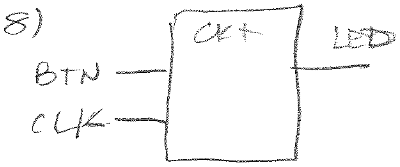
CIRCUIT & External Control



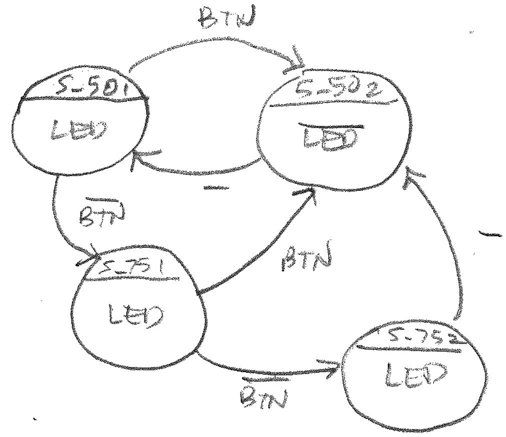
MOORE outputs

Could be done with more states

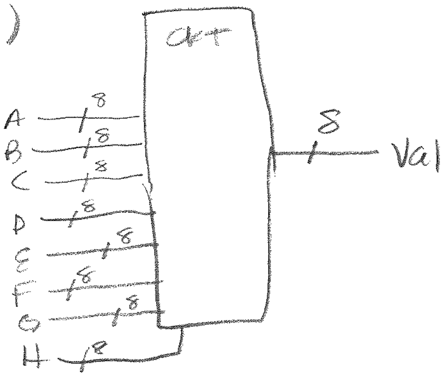
8)

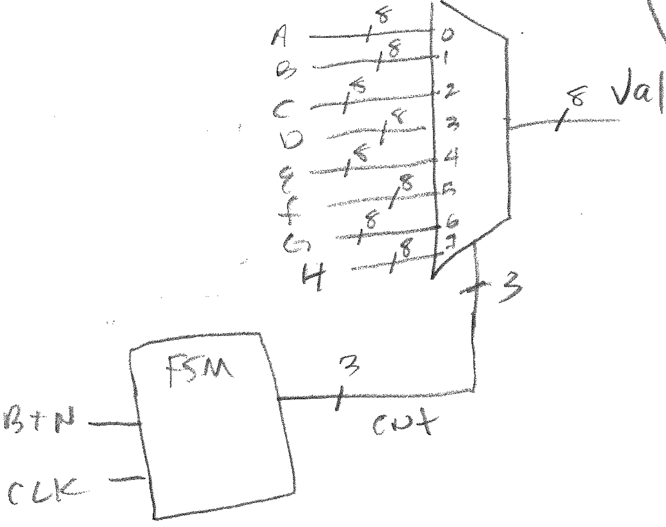


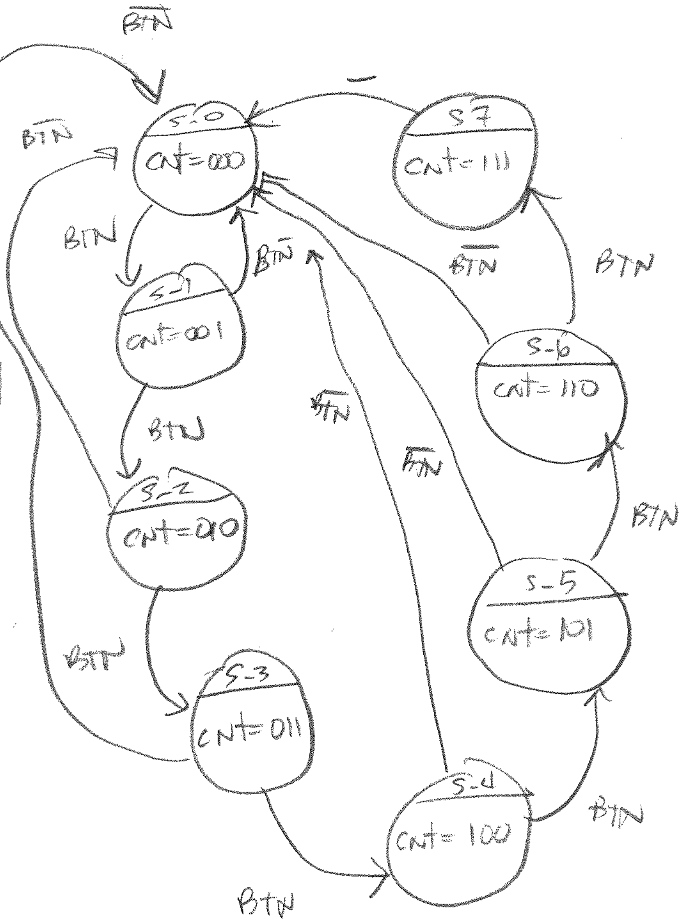
circuit controlled



9)

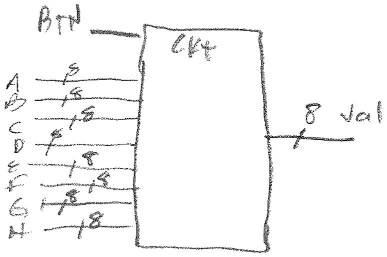


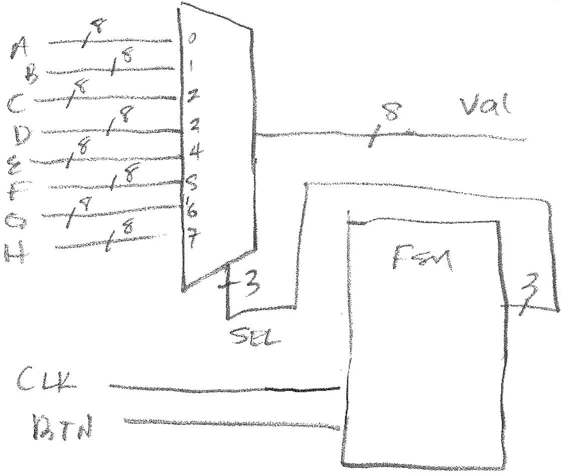


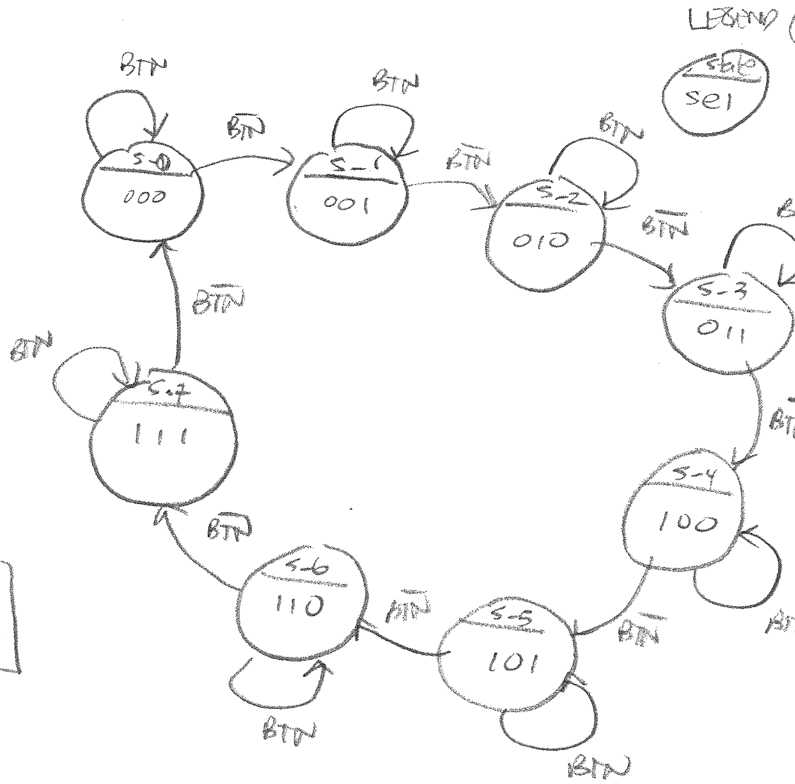


CIRCUIT control

10)

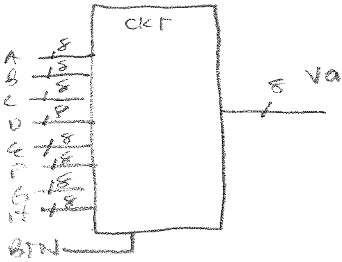


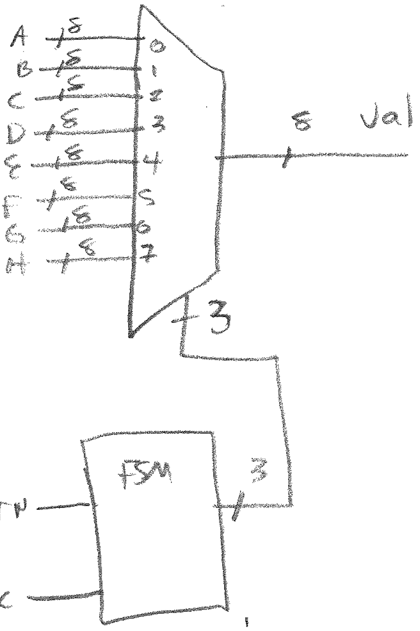


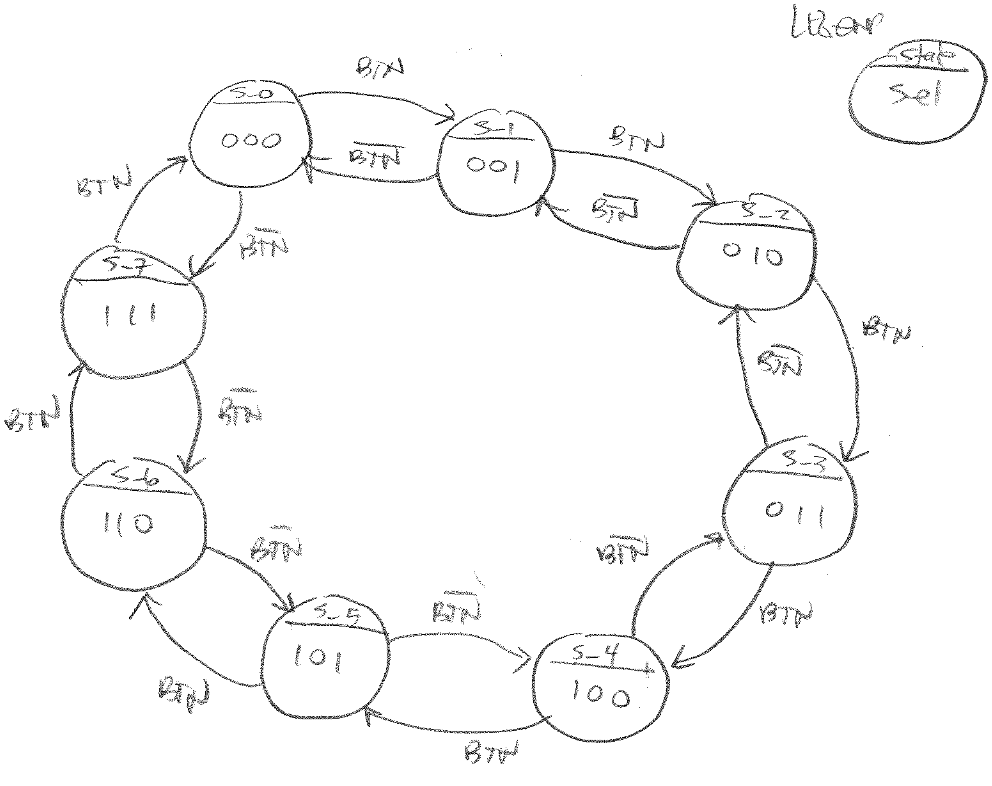


circuit control

11)



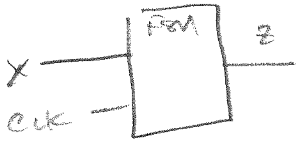


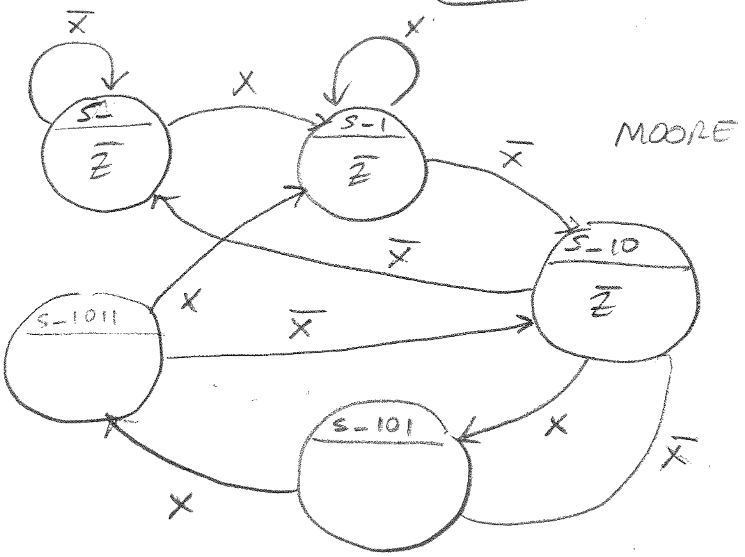


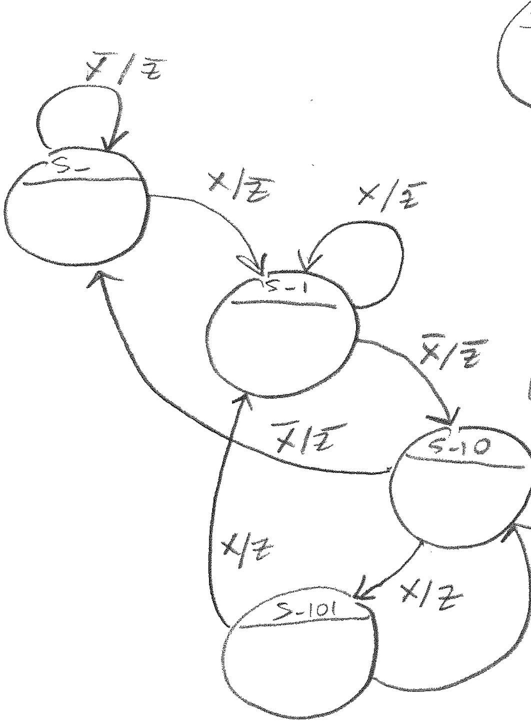
circuit control

circuit control

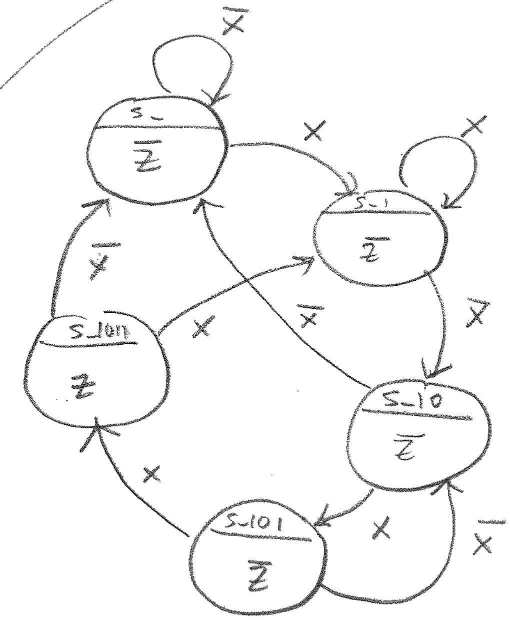
12) SEQ = 1011, NON RESETTING



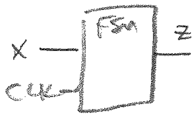


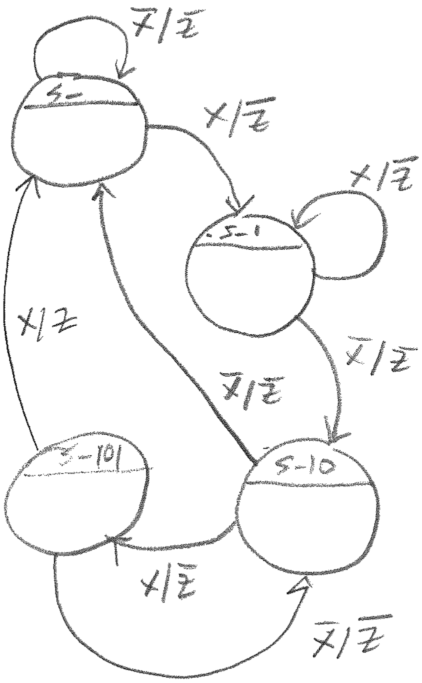

$\overline{X}/\overline{Z}$
S_
$X/\overline{Z}$
$X/\overline{Z}$
S_1
$\overline{X}/\overline{Z}$
$\overline{X}/\overline{Z}$
S_10
X/Z
$X/\overline{Z}$
$\overline{X}/\overline{Z}$
S_101

MEALY



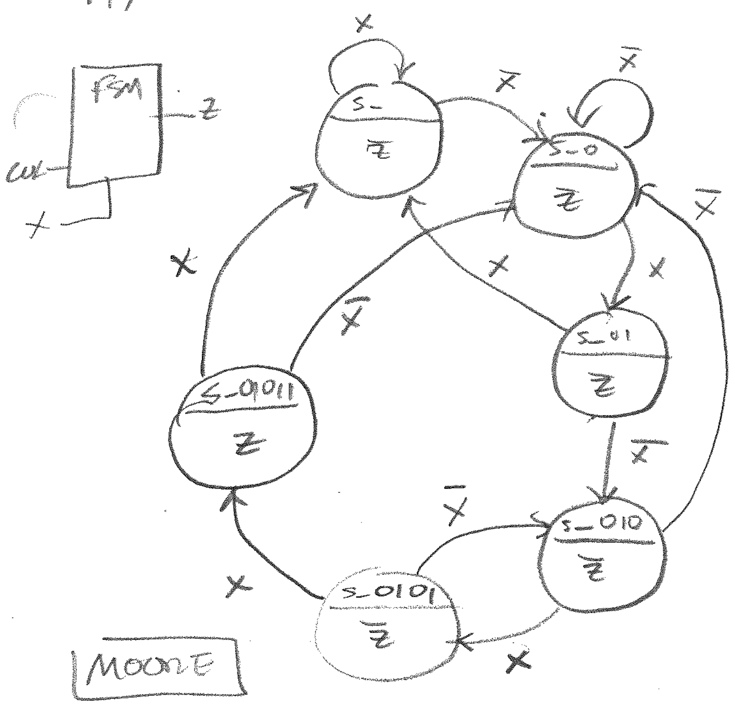
13) SEQ=1011 Resetting




$\overline{X}/\overline{Z}$
S_
$X/\overline{Z}$
$X/\overline{Z}$
S_1
$\overline{X}/\overline{Z}$
X/Z
$\overline{X}/\overline{Z}$
S_101
S_10
$X/\overline{Z}$
$\overline{X}/\overline{Z}$

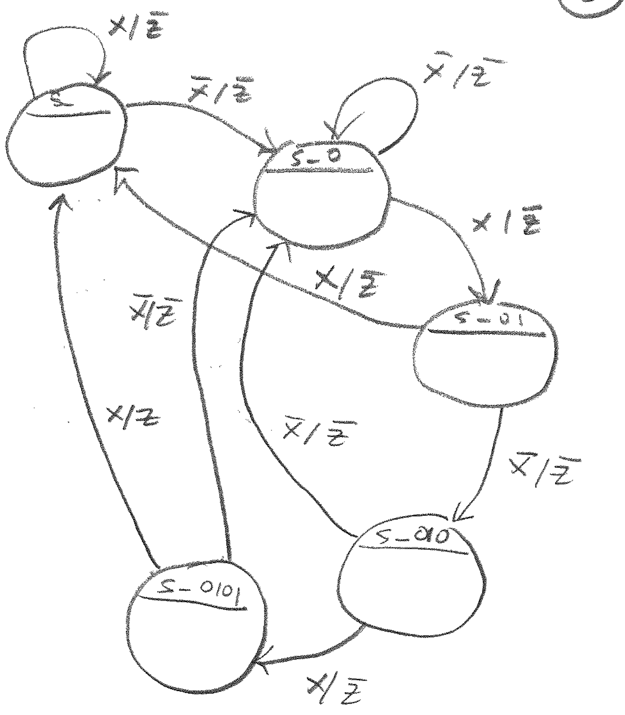
circuit control

14) SEQ= 01011 Non-Reset



MOORE

Circuit control

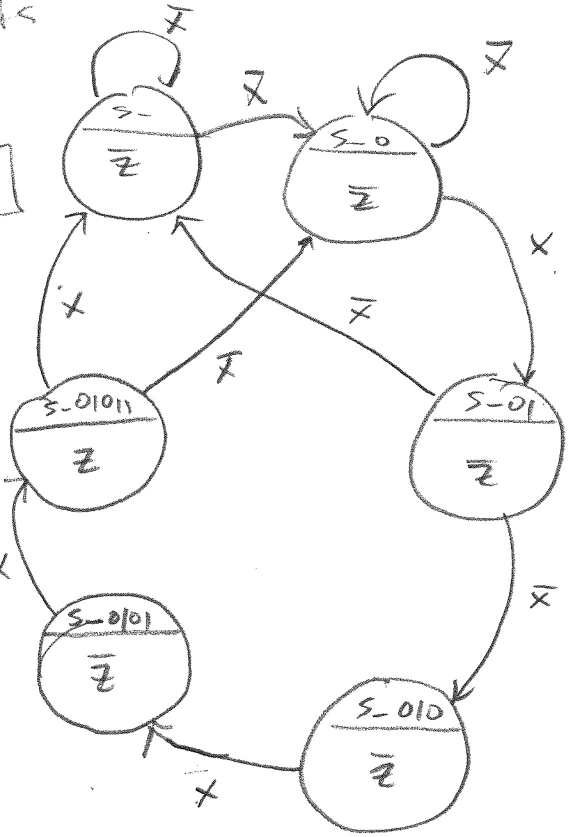

S_
S_0
S_01
S_010
S_0101
X/$\overline{Z}$
$\overline{X}$/$\overline{Z}$
$\overline{X}$/$\overline{Z}$
X/$\overline{Z}$
X/$\overline{Z}$
$\overline{X}$/$\overline{Z}$
$\overline{X}$/$\overline{Z}$
X/$\overline{Z}$
$\overline{X}$/$\overline{Z}$
X/Z

MEALY

Reset

15)

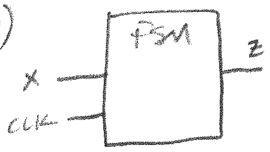
mealy

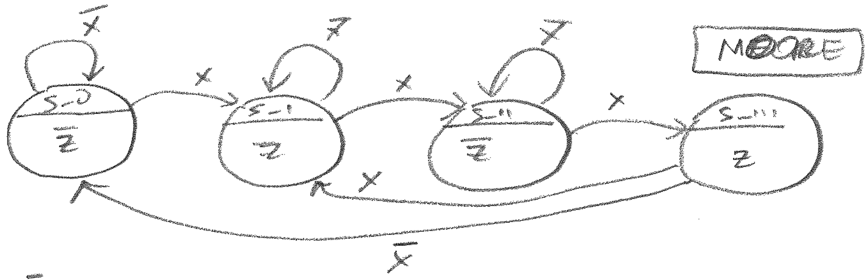


SAME AS ABOVE

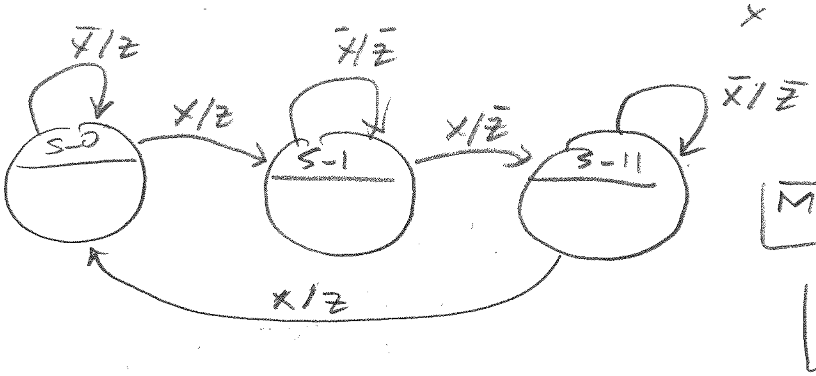
Circuit control

16)





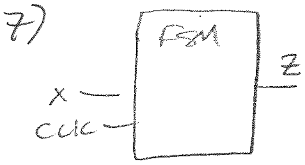
MOORE



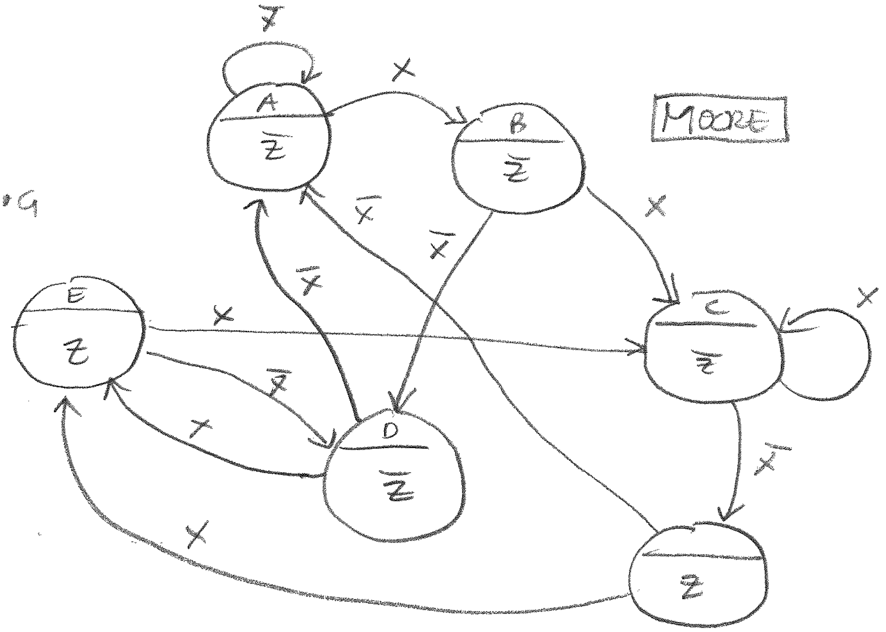
MEALY

circuit control

17)

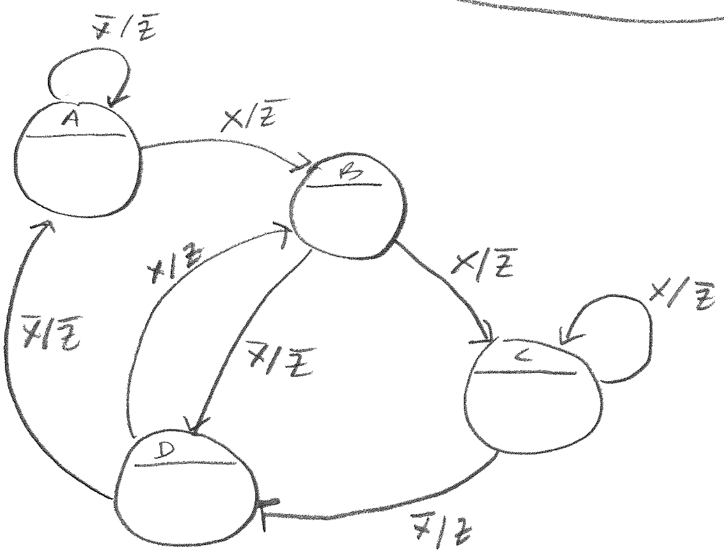


Seqs: 101 Non Repeating
110



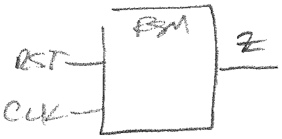
MOORE

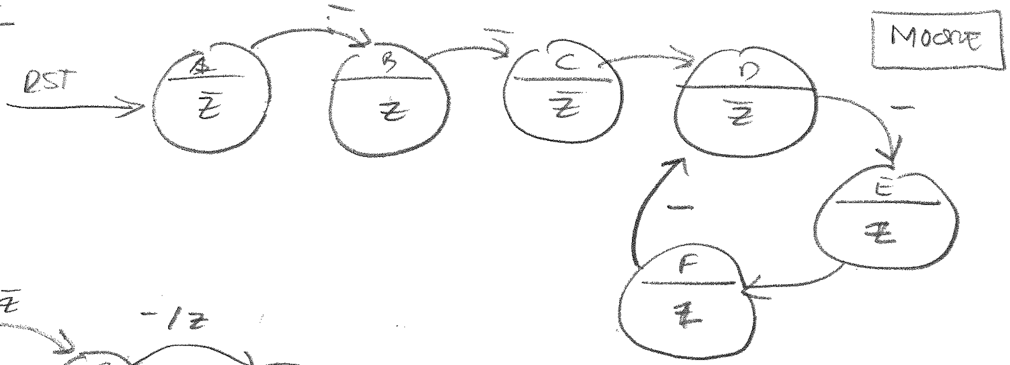
circuit control

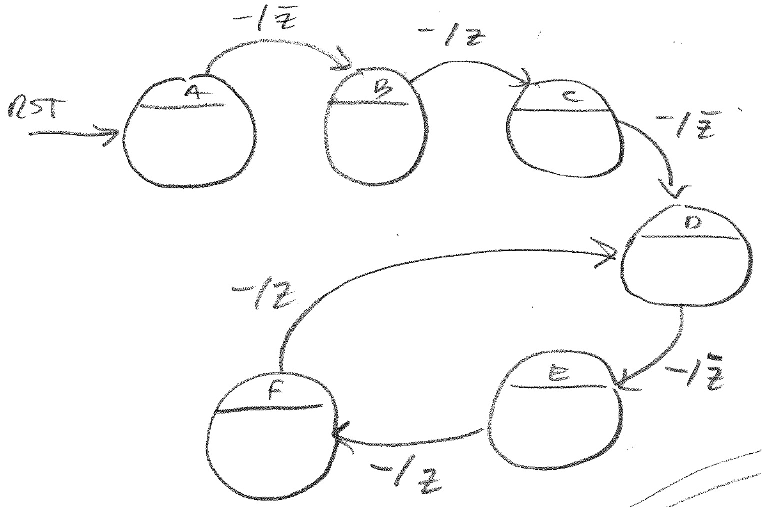


MEALY

18)





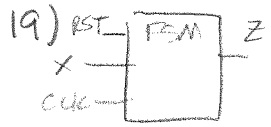


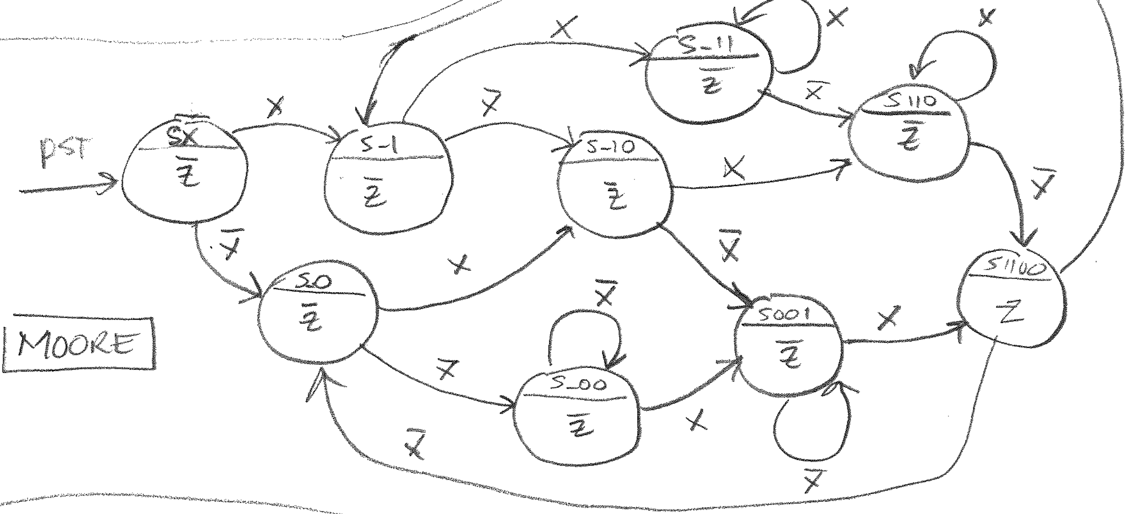
MEALY

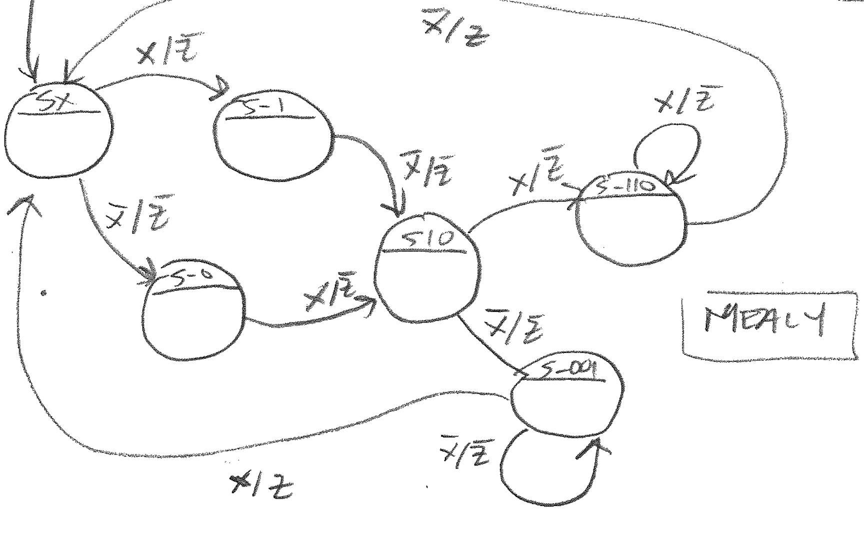
circuit & external control

RST would be asynchronous input on state registers

19)





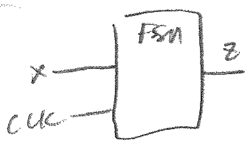


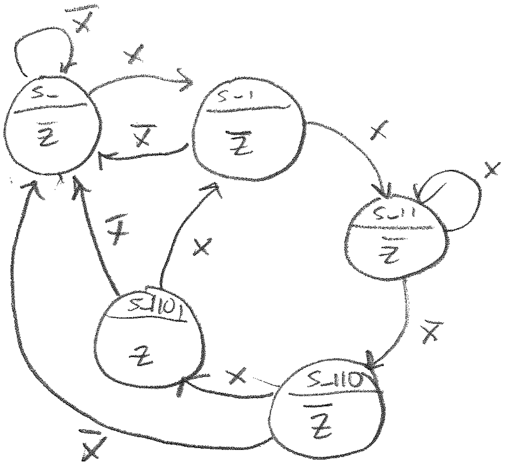
circuit & external control

RST would be asynchronous input on state registers

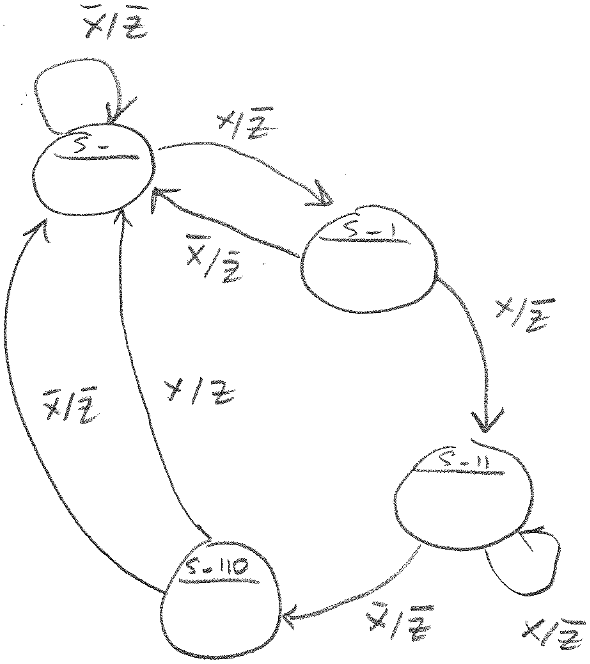
20) SEQ = 1101
Resetting

Circuit Control



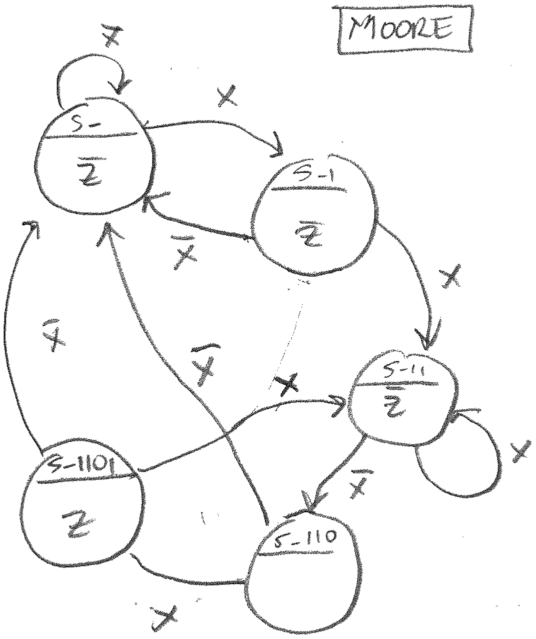


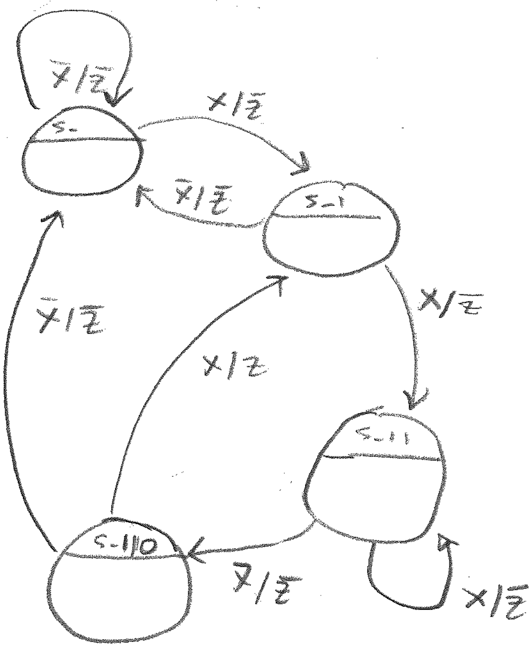
MOORE



MEALY

21) SEQ = 1101
NON-Resetting

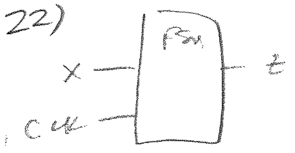




Circuit Control

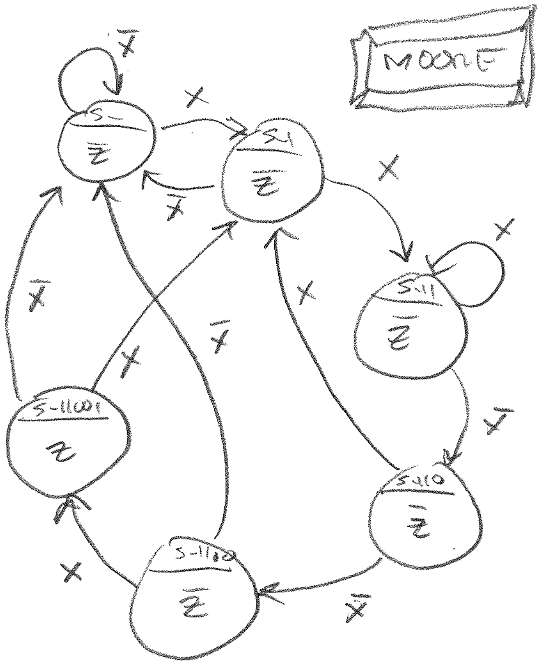
SEQ = 11001

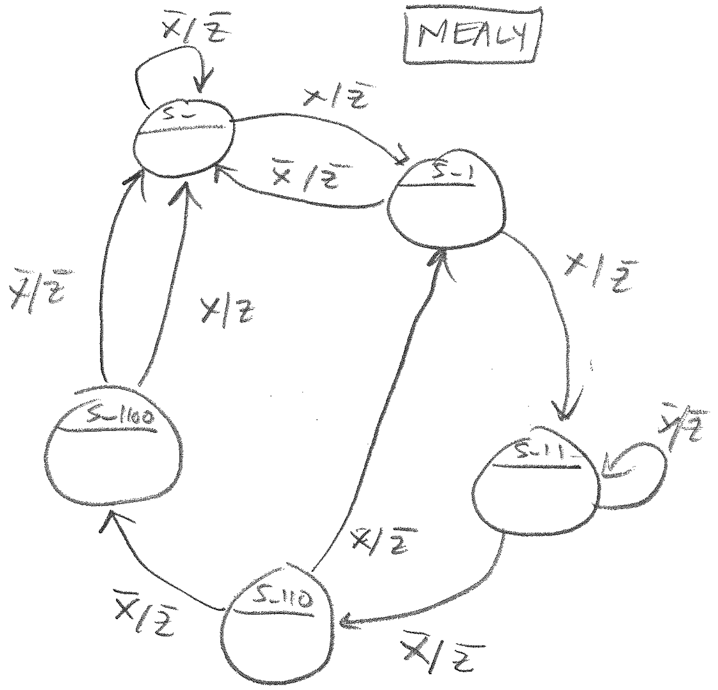
22)



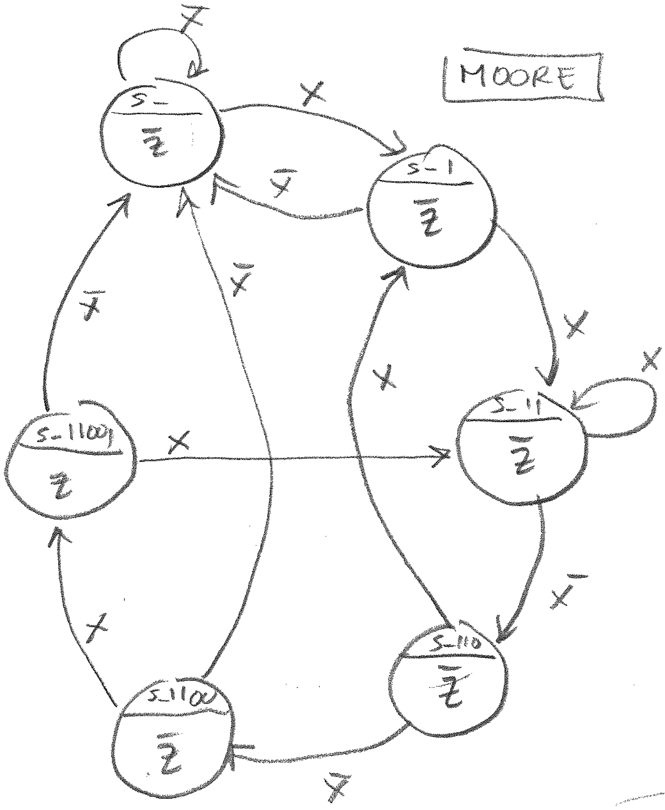
Resetting

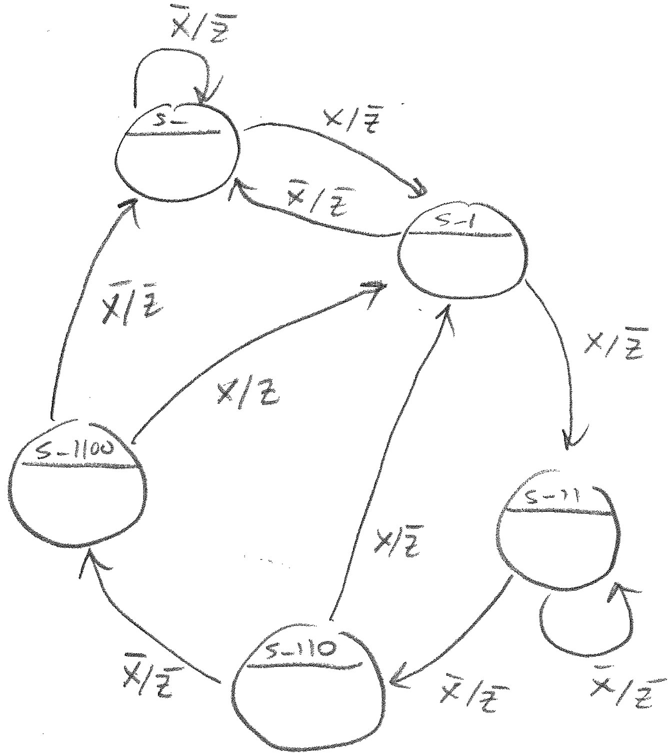
CIRCUIT control





23) NON-resetting



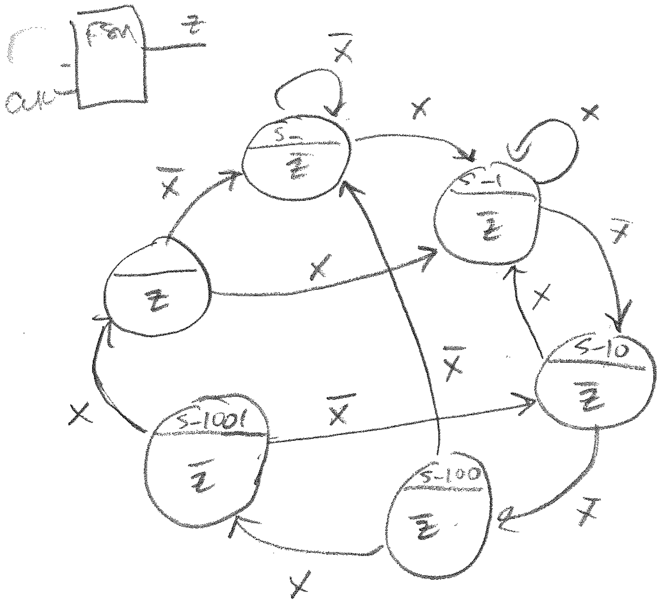


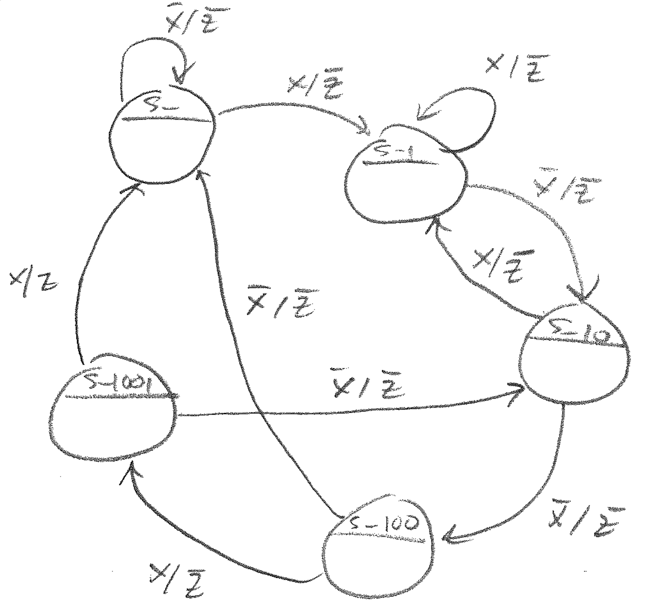
CIRCUIT control

110010

24) 10011 Resetting

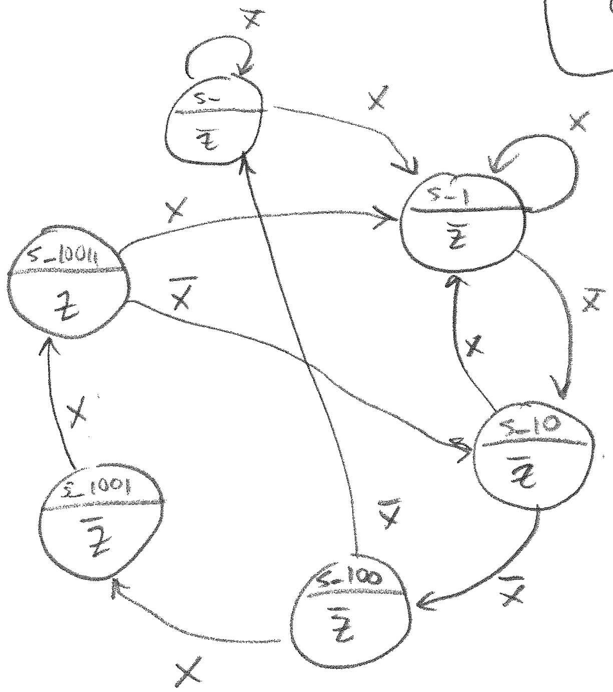
Circuit Control

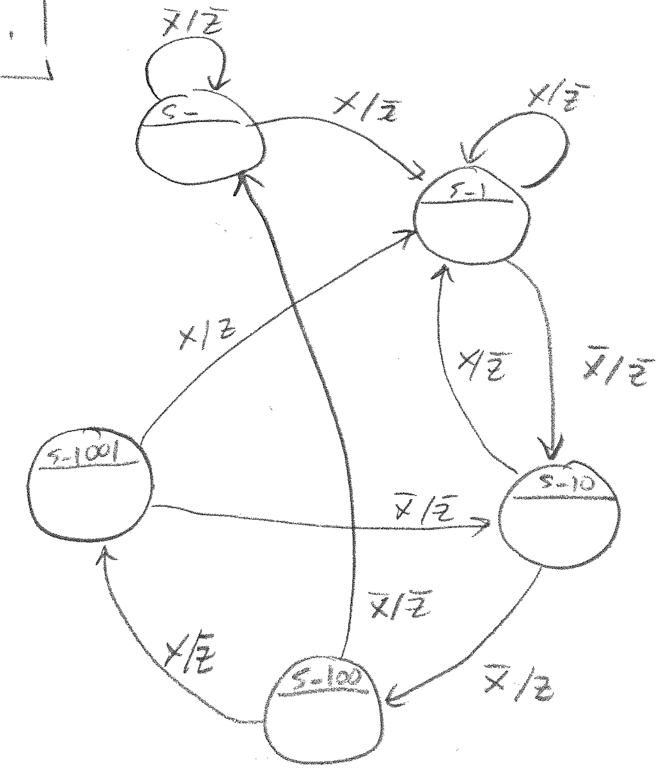



$\overline{X}/\overline{Z}$
S_
X/$\overline{Z}$
X/$\overline{Z}$
S_1
$\overline{X}/\overline{Z}$
X/$\overline{Z}$
X/Z
$\overline{X}/\overline{Z}$
S_10
S_1001
$\overline{X}/\overline{Z}$
S_100
$\overline{X}/\overline{Z}$
X/$\overline{Z}$

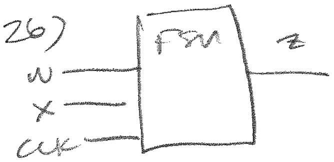
25) Non Resetting

Circuit Control.

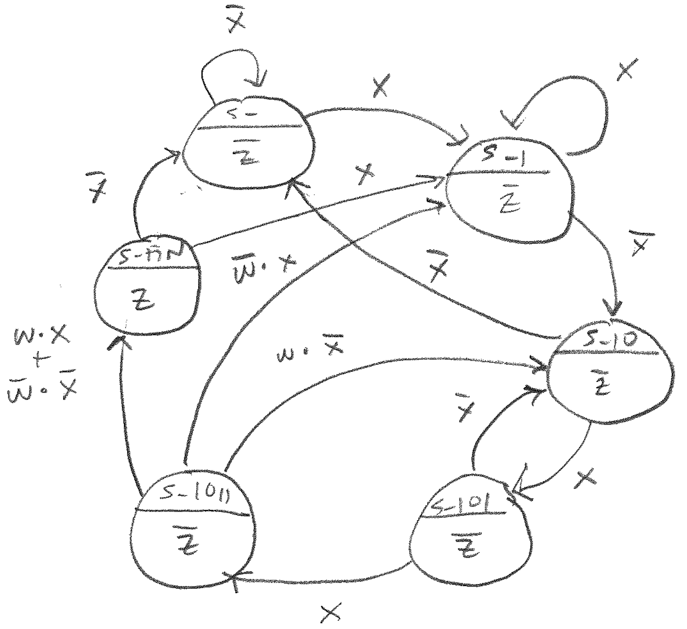



$\overline{X}/\overline{Z}$
S_
X/$\overline{Z}$
X/$\overline{Z}$
S_1
X/Z
X/$\overline{Z}$
$\overline{X}/\overline{Z}$
S_1001
$\overline{X}/\overline{Z}$
S_10
$\overline{X}/\overline{Z}$
X/$\overline{Z}$
S_100
$\overline{X}$/Z

26)



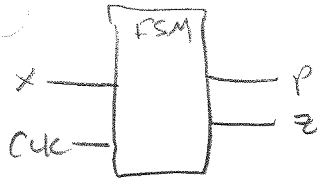
resetting
Moore



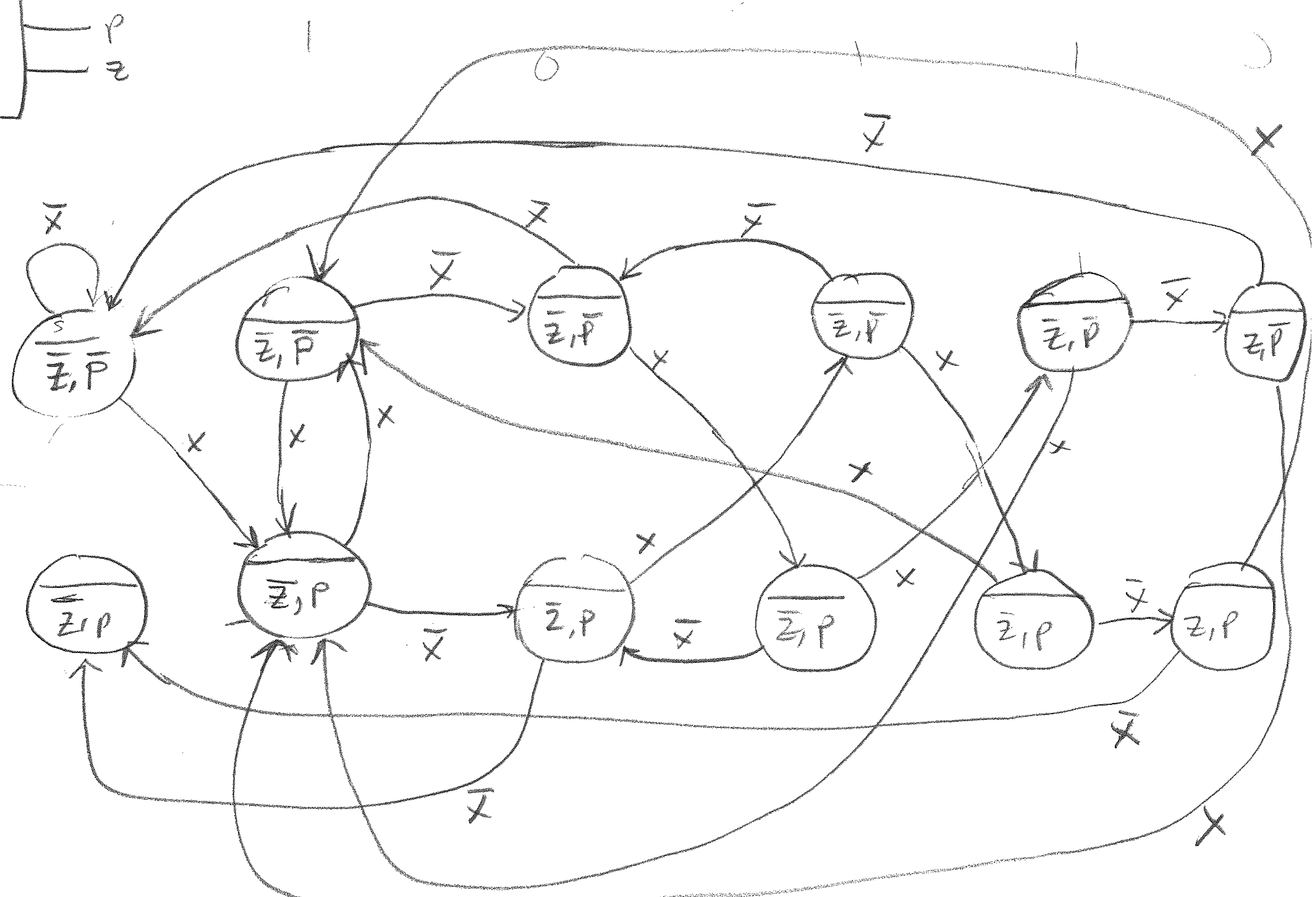
CIRCUIT
control

27) 10110

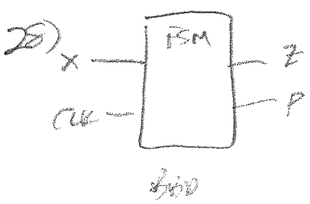
non-resetting



CIRCUIT control

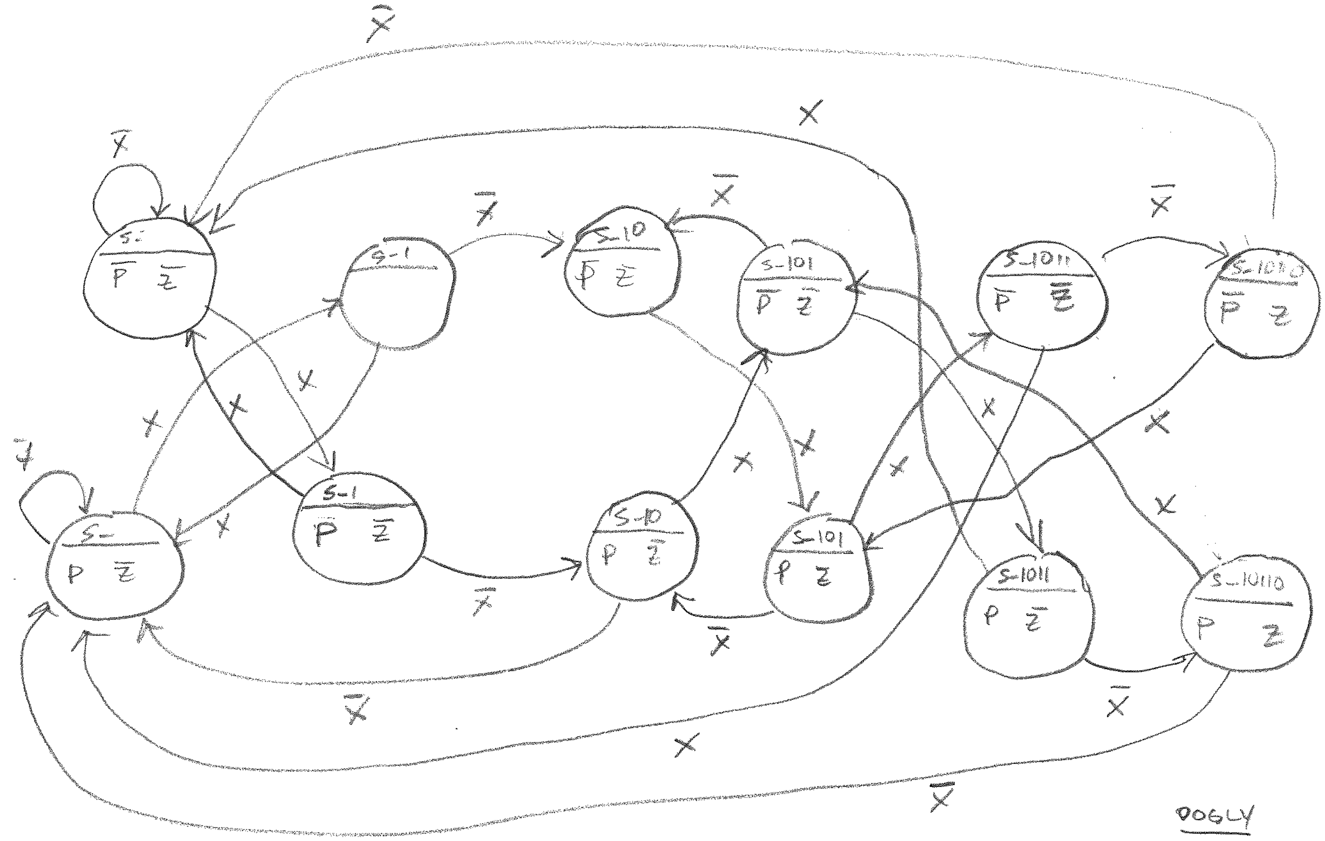


28)



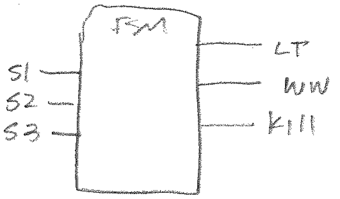
10110

circuit : control



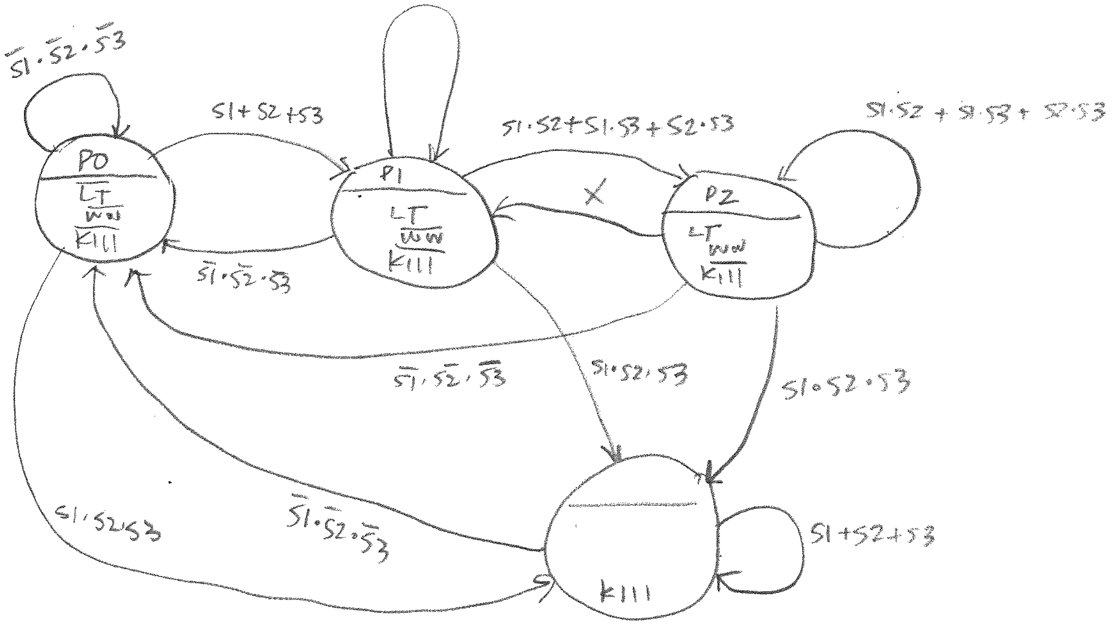
DOGLY

29)



CIRCUIT CONTROL

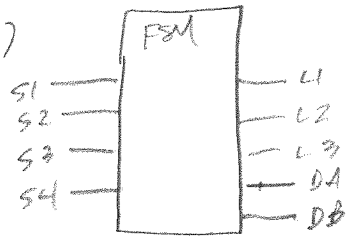
$\overline{S1}\cdot\overline{S2}\cdot S3 + \overline{S1}\cdot S2\cdot\overline{S3} + S1\cdot\overline{S2}\cdot\overline{S3} = X$

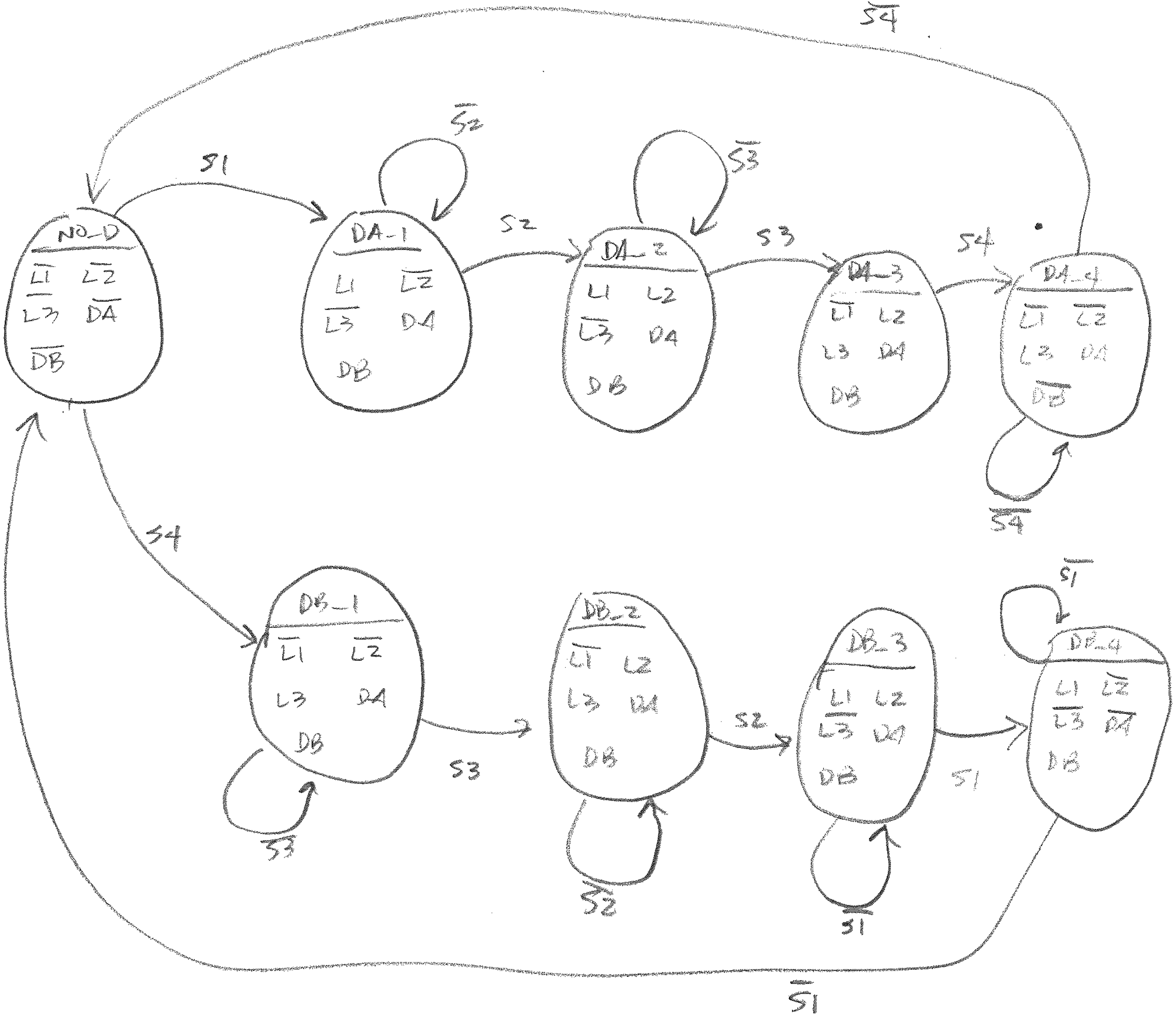

$\overline{S1}\cdot\overline{S2}\cdot\overline{S3}$
S1+S2+S3
S1·S2+S1·S3+S2·S3
S1·S2 + S1·S3 + S2·S3
P0
$\overline{LT}$
$\overline{WW}$
$\overline{KIll}$
P1
LT
$\overline{WW}$
$\overline{KIll}$
P2
LT
WW
$\overline{KIll}$
X
$\overline{S1}\cdot\overline{S2}\cdot\overline{S3}$
$\overline{S1}\cdot\overline{S2}\cdot\overline{S3}$
S1·S2·S3
S1·S2·S3
S1·S2·S3
$\overline{S1}\cdot\overline{S2}\cdot\overline{S3}$
KIll
S1+S2+S3

S1	S2	S3		
0	0	0	0	✓
0	0	1	1	
0	1	0	1	
0	1	1	0	
1	0	0	1	
1	0	1	0	
1	1	0	0	
1	1	1	0	✓

307

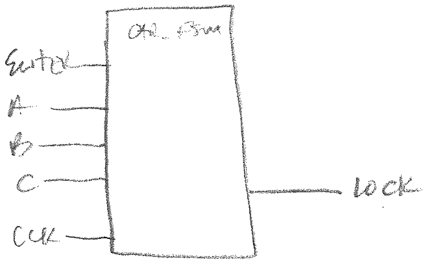
circuit Control





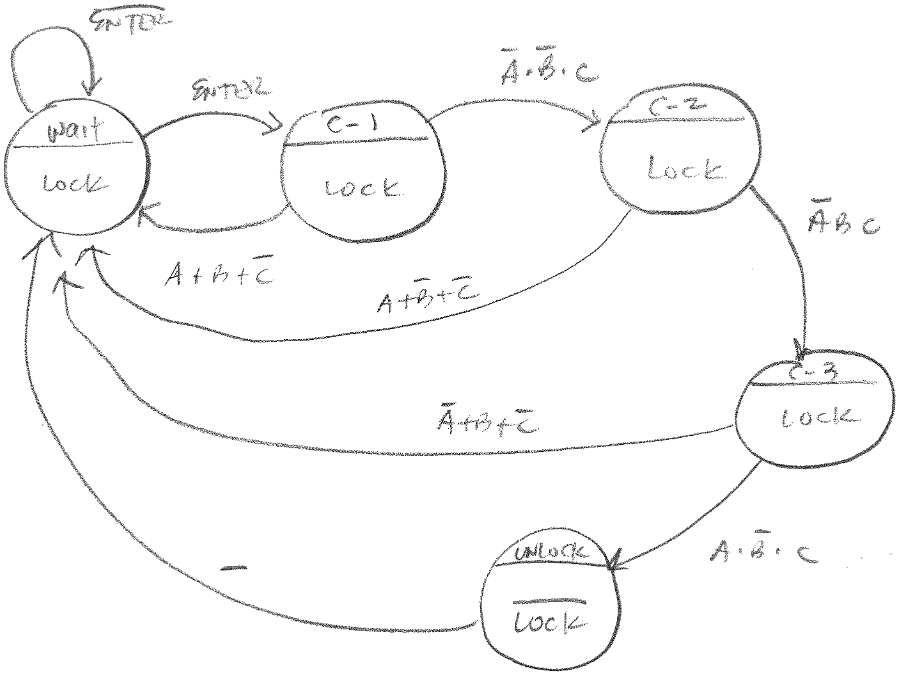
NO_D
$\overline{L1}$ $\overline{L2}$ L3 $\overline{DA}$ $\overline{DB}$
S1
DA_1
L1 $\overline{L2}$ $\overline{L3}$ DA DB
$\overline{S2}$
S2
DA_2
L1 L2 $\overline{L3}$ DA DB
$\overline{S3}$
S3
DA_3
$\overline{L1}$ L2 L3 DA DB
S4
DA_4
$\overline{L1}$ $\overline{L2}$ L3 DA $\overline{DB}$
$\overline{S4}$
$\overline{S4}$
S4
DB_1
$\overline{L1}$ $\overline{L2}$ L3 DA DB
$\overline{S3}$
S3
DB_2
$\overline{L1}$ L2 L3 DA DB
$\overline{S2}$
S2
DB_3
L1 L2 $\overline{L3}$ DA DB
$\overline{S1}$
S1
DB_4
L1 $\overline{L2}$ $\overline{L3}$ $\overline{DA}$ DB
$\overline{S1}$
$\overline{S1}$

31)

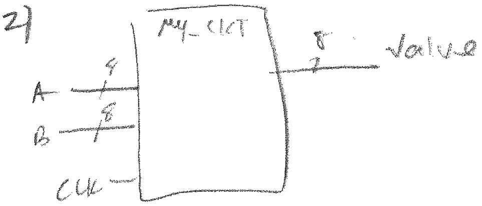


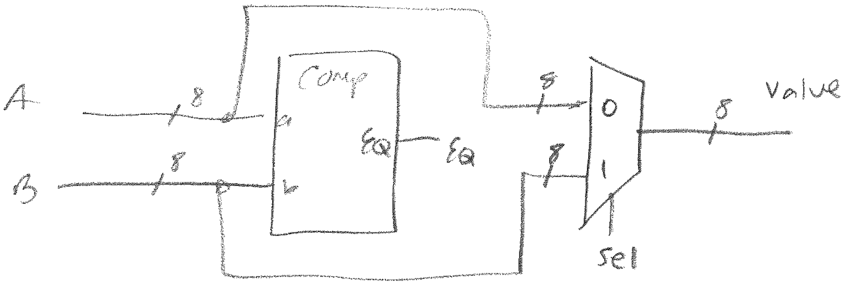
lock=1 : lock on
lock=0 : lock off

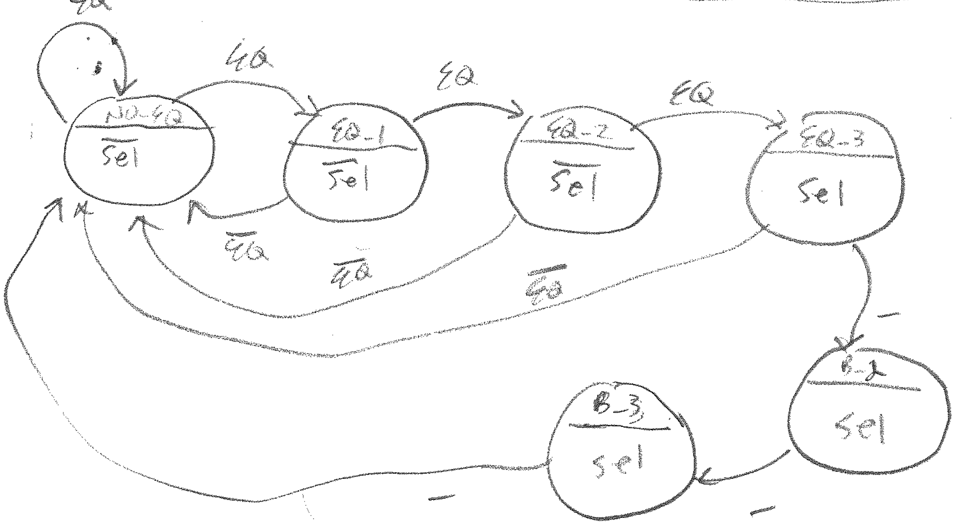
circuit control

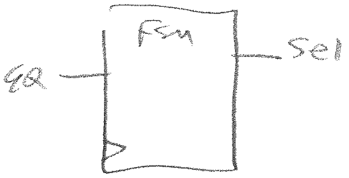


32)





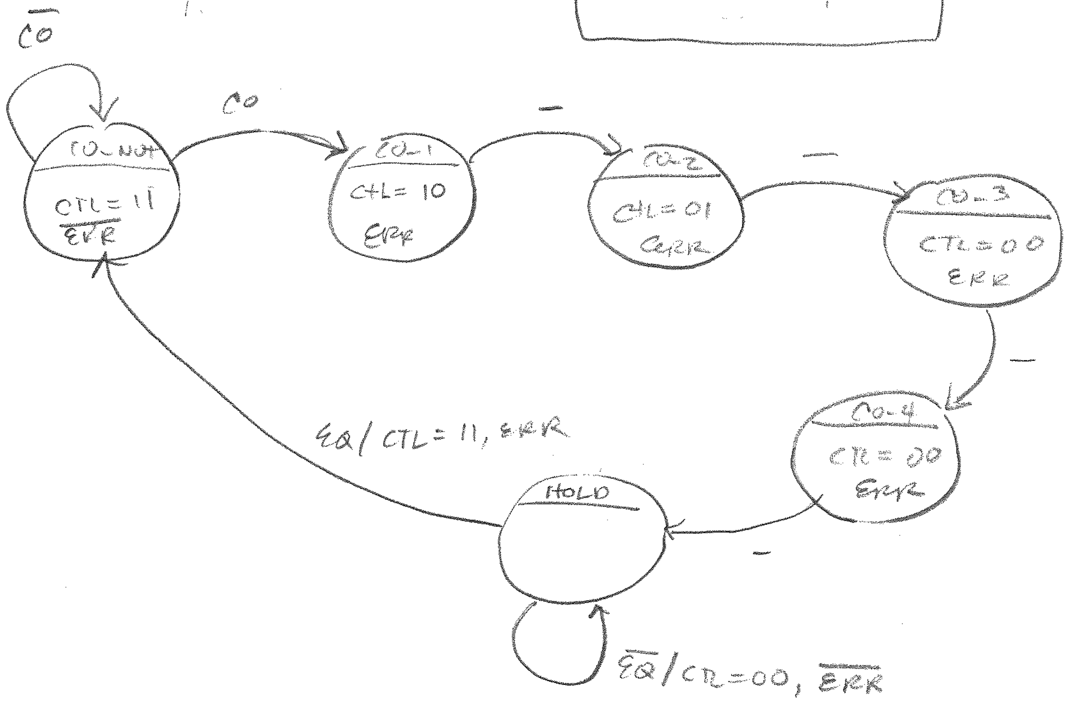




circuit control

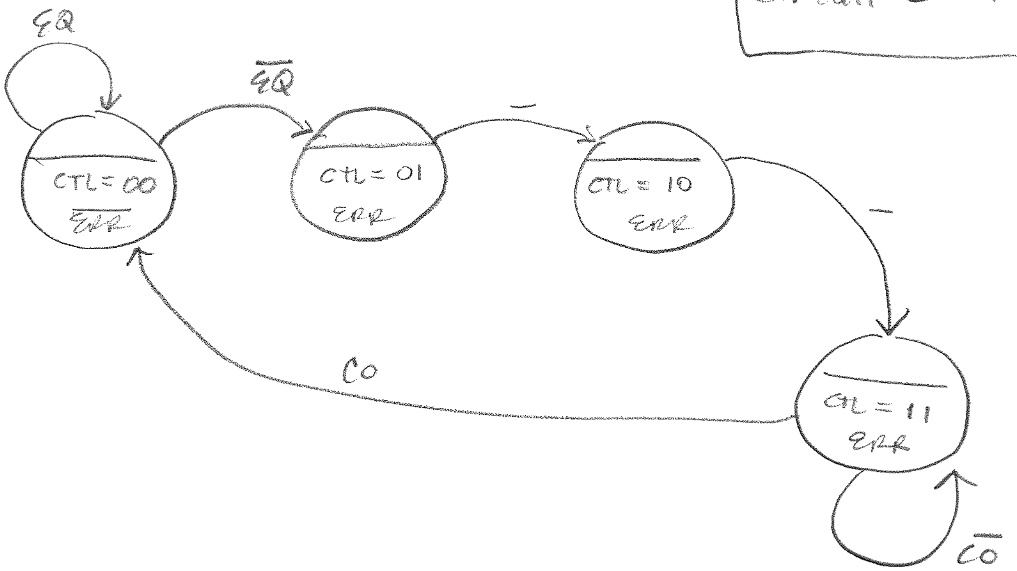
33)

Circuit Control

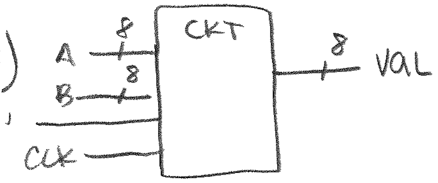


34)

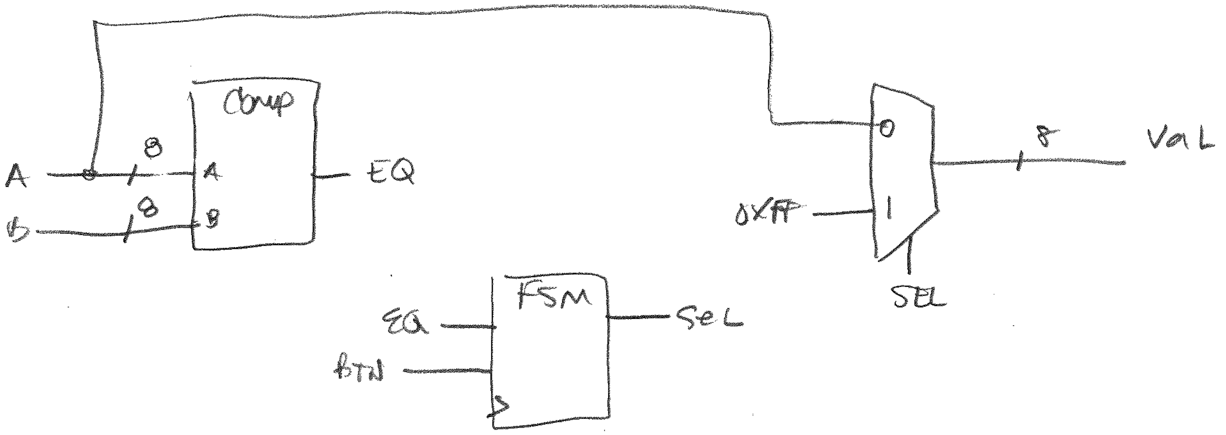
Circuit Control.

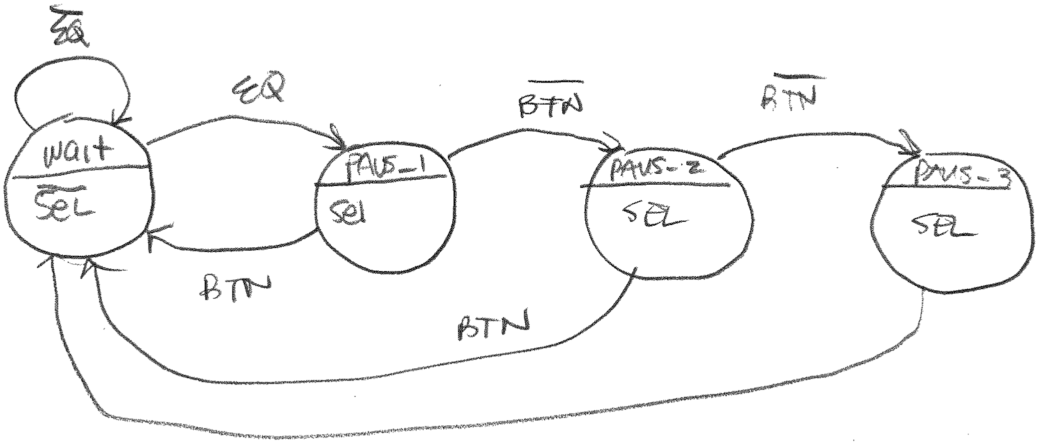


35)

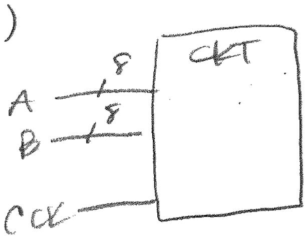


circuit control

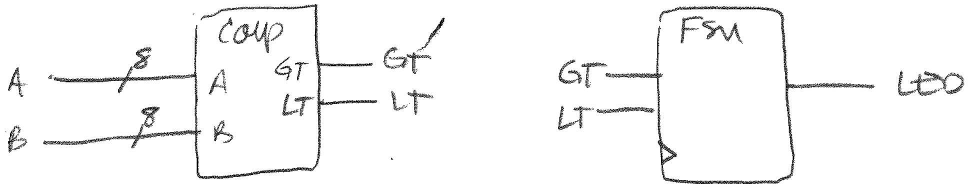


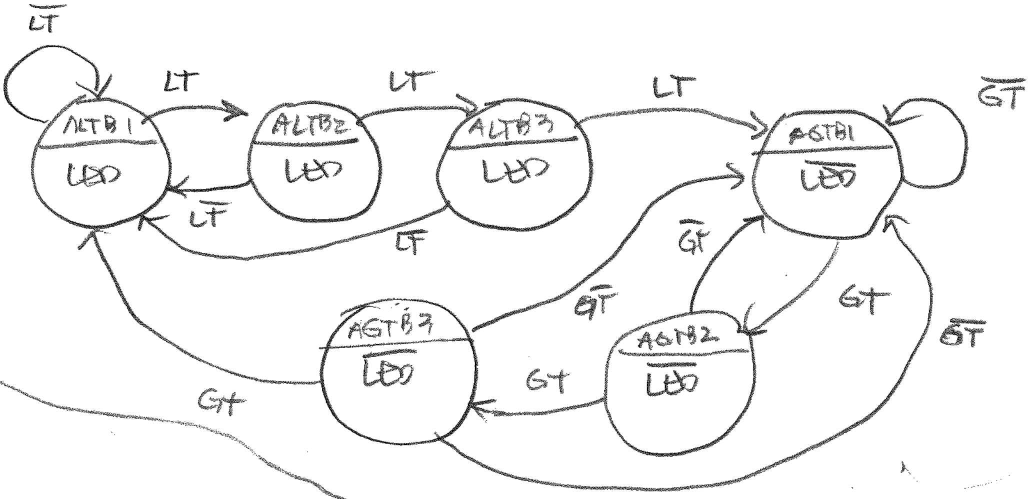


36)

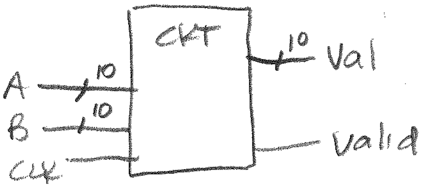


Circuit Control

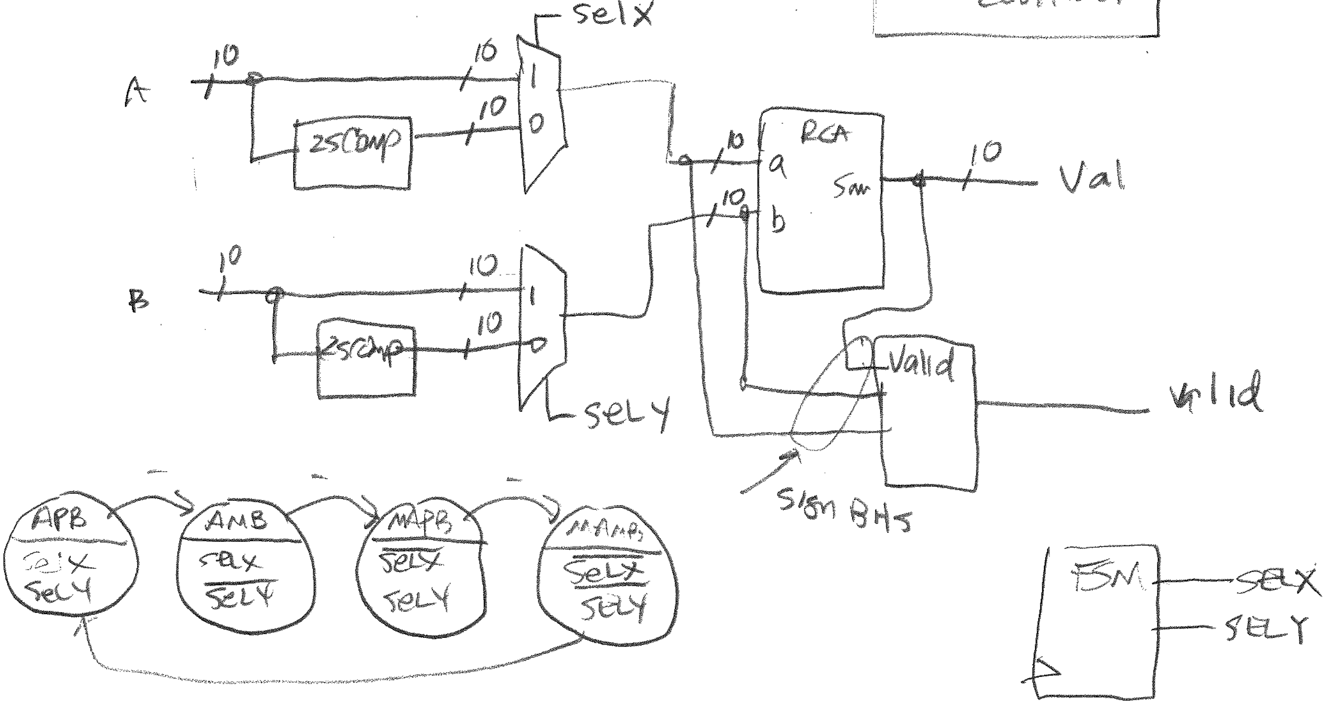




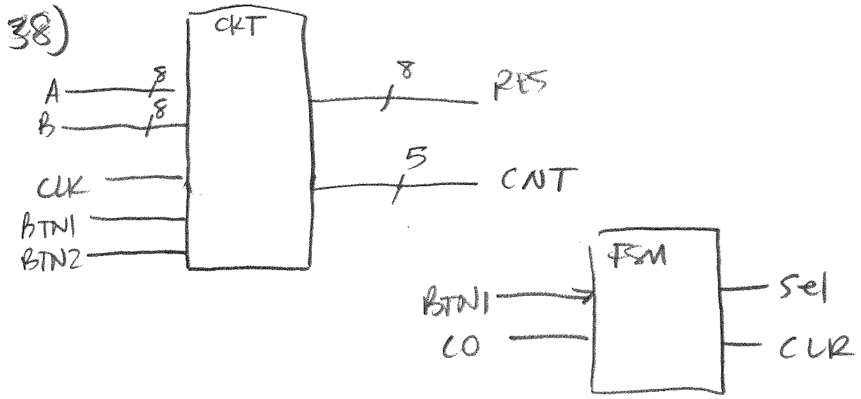
37)



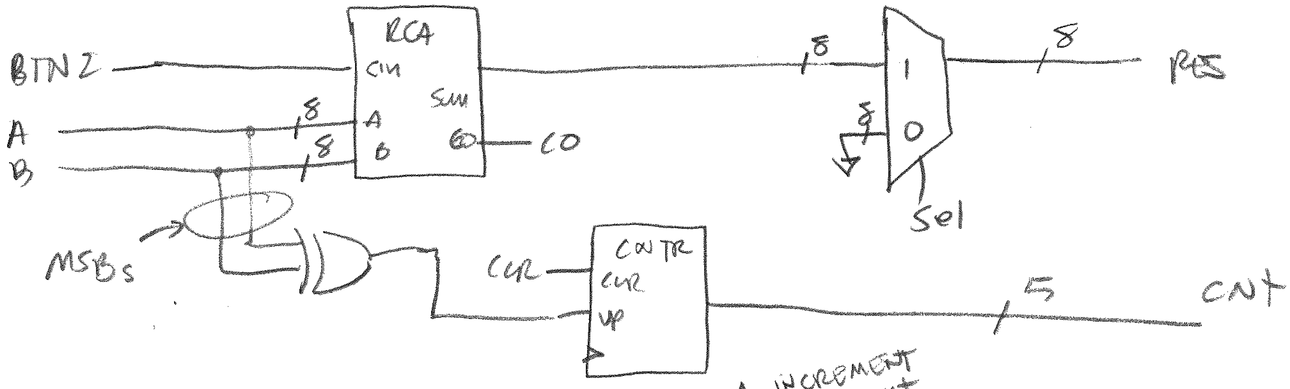
Circuit control



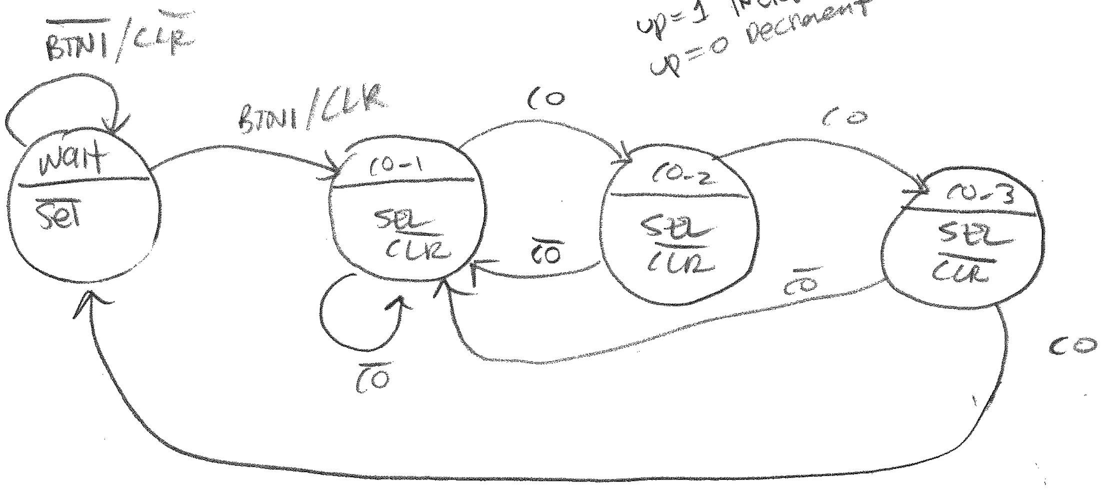
38)



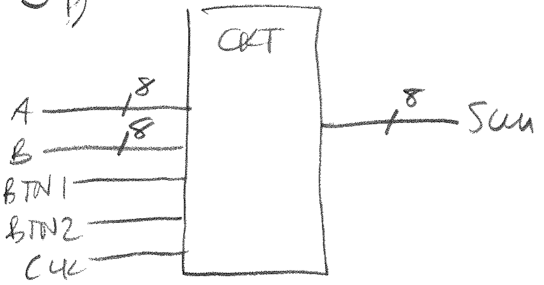
CIRCUIT, CONTROL



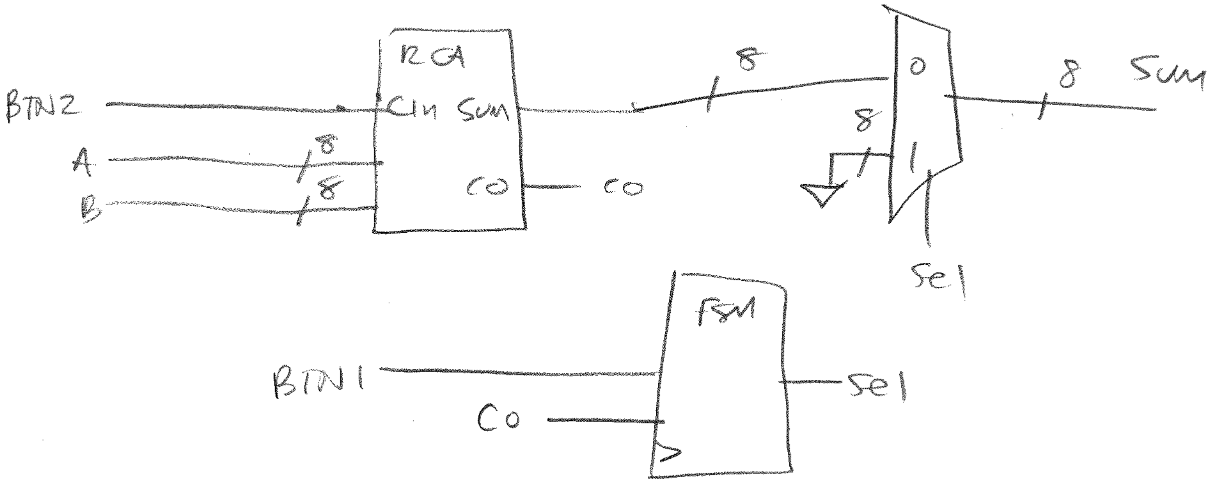
up=1 INCREMENT
up=0 Decrement

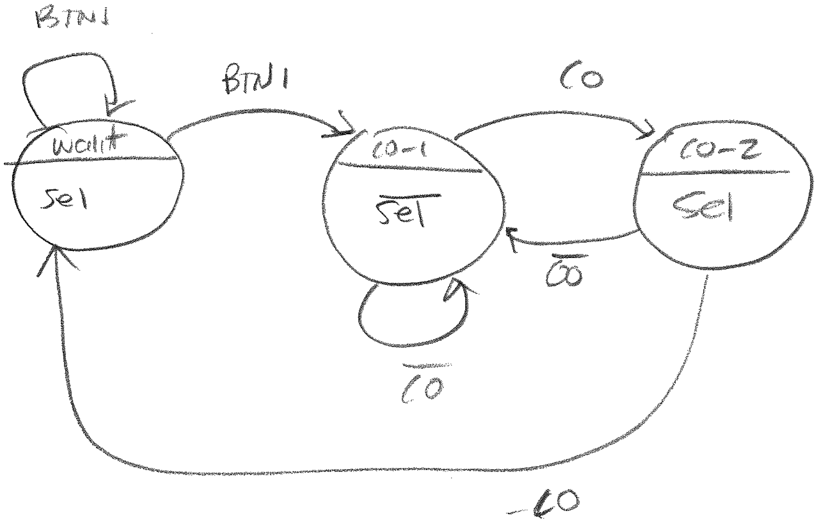


39)



Circuit controlled

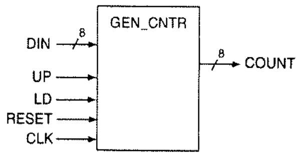


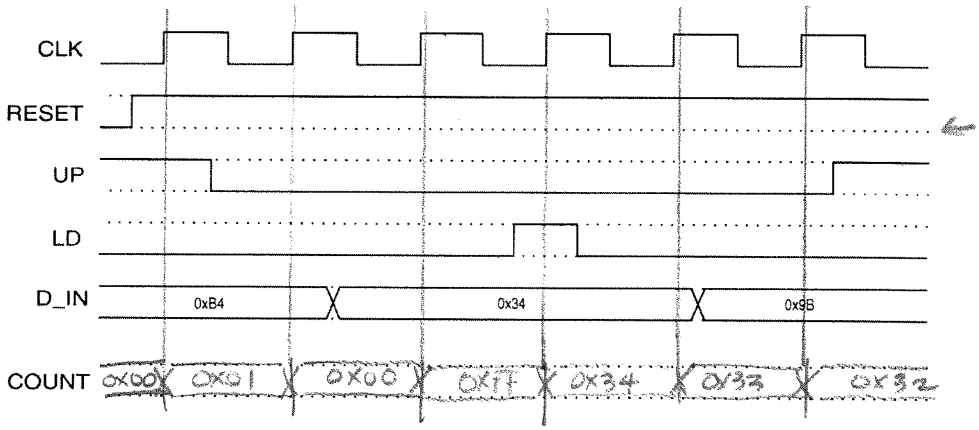


26.7 Chapter Exercises

1) The block diagram on the right shows a model of an 8-bit counter. Use the following assumptions in order to complete the following timing diagram. Assume propagation delays are negligent.

- The LD input enables the DIN loading into the counter
- The RESET input is an asynchronous and active low used to reset the counter
- The COUNT output shows the current value stored by the counter
- The counter counts up when the UP input is asserted (active high) or down otherwise. All count operations are synchronous.





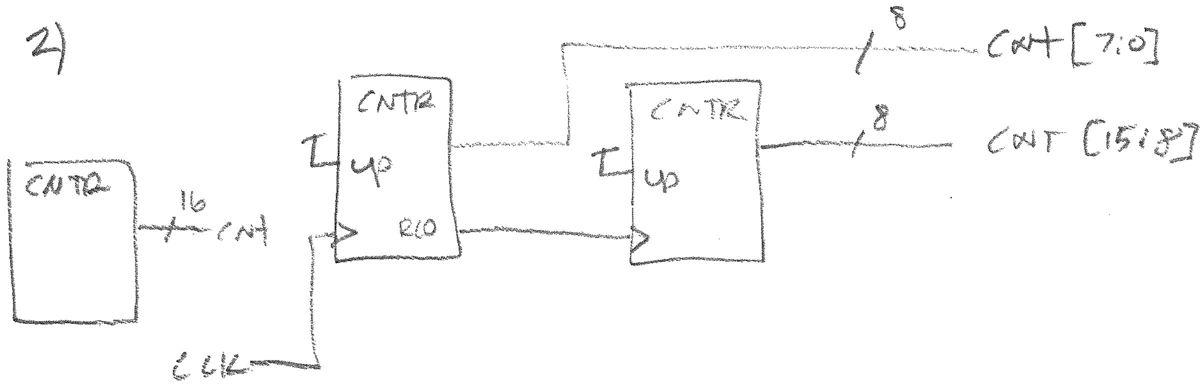
← SOLN

2) Show a schematic that uses two standard 8-bit up counters to implement a 16-bit up counter.

3) In your own words, describe how it is that a counter can replace an accumulator in certain circumstances.

4) Briefly describe the difference between the RCO when the counter is counting up verse counting down.

2)

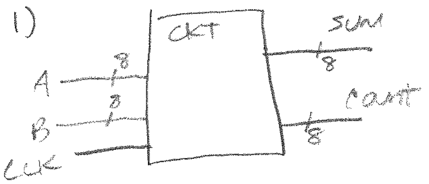


3) A COUNTER CAN REPLACE AN ACCUMULATOR WHEN THE ACCUMULATOR IS ACCUMULATING 1's OR 0's. IN THIS CASE, THE COUNTER SIMPLY INCREMENTS (ADDS 1).

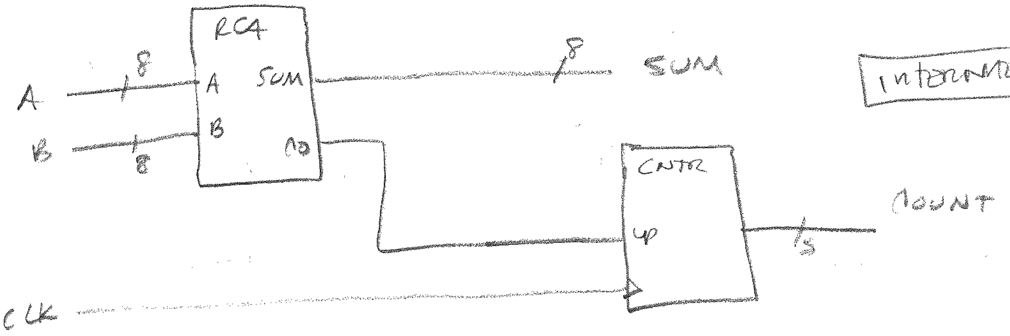
4) WHEN A COUNTER IS COUNTING UP, RCO INDICATES WHEN THE COUNTER'S out put IS AT IT'S MAXIMUM value. WHEN A COUNTER IS COUNTING DOWN, THE RCO INDICATES WHEN THE COUNTER IS AT ITS MINIMUM value.

CHAP ~~10~~ 24 DESIGN PROBLEM SOLNS

1)

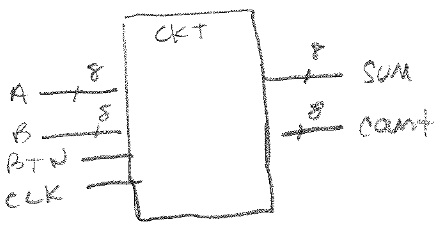


NOTES: up = 1: counter increments
up = 0: counter holds

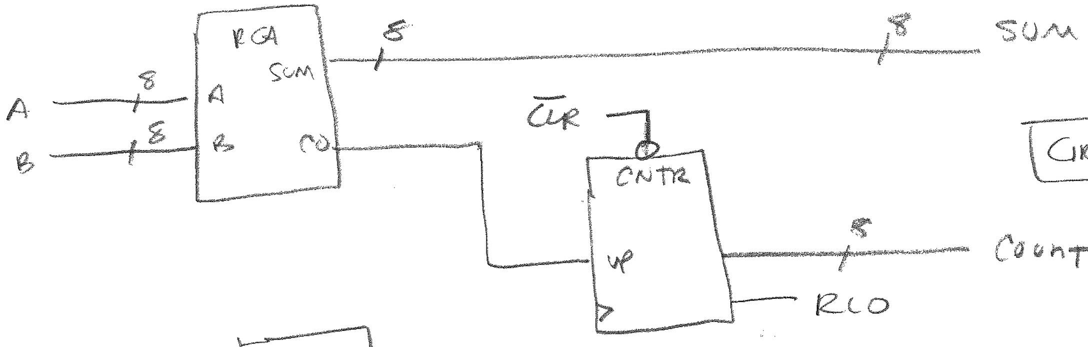


internal control

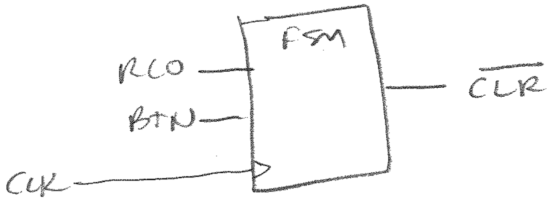
2)

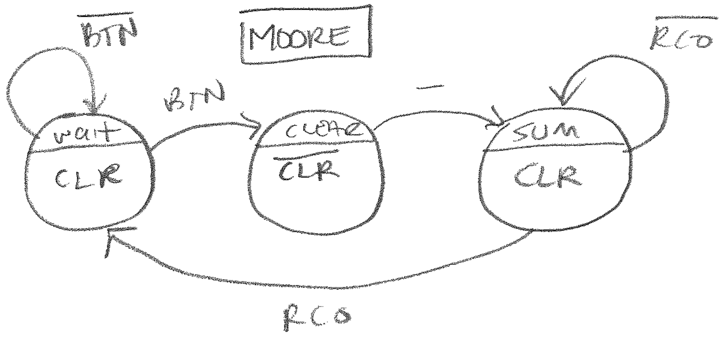


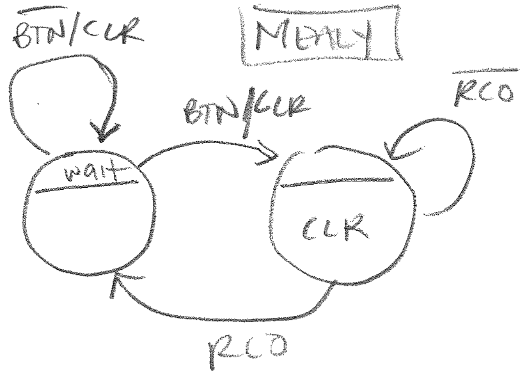
NOTES: up = 1 counter increments
up = 0 counter Holds



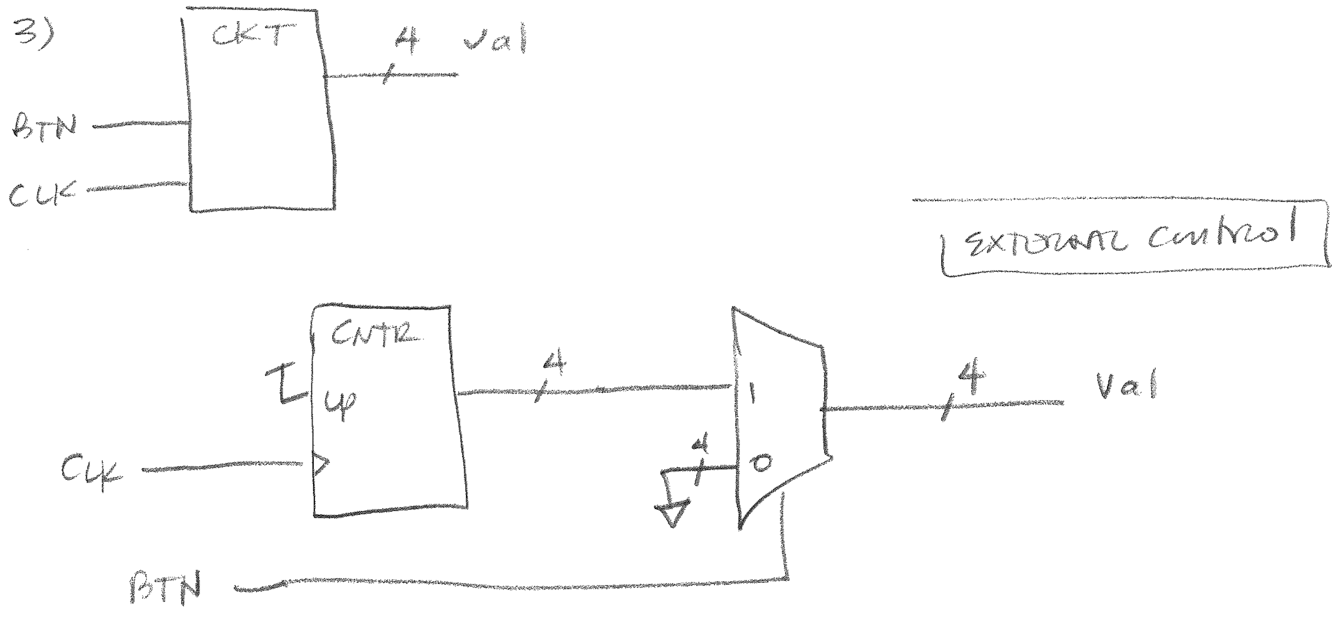
Circuit controlled



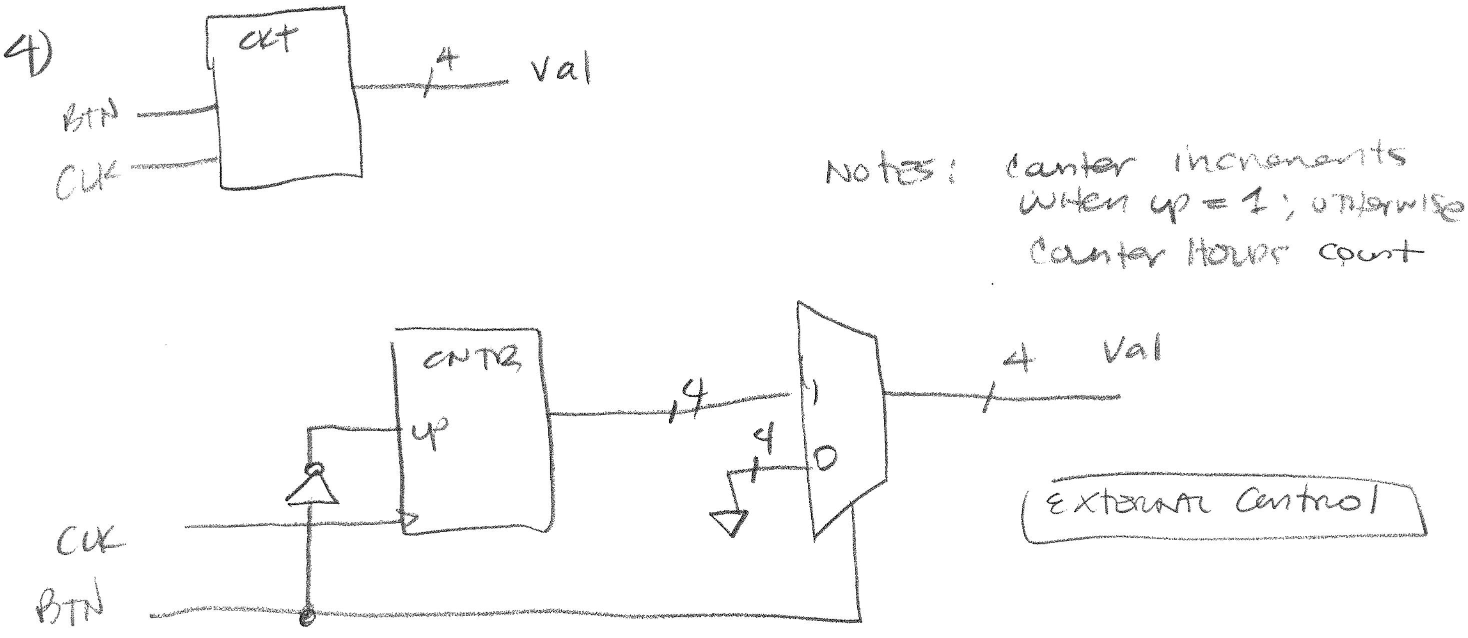




3)

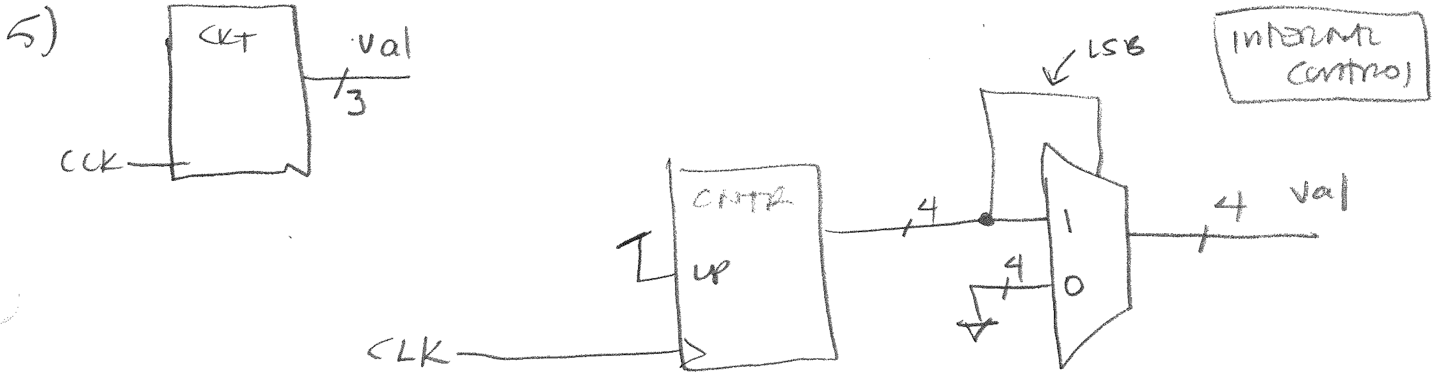


4)

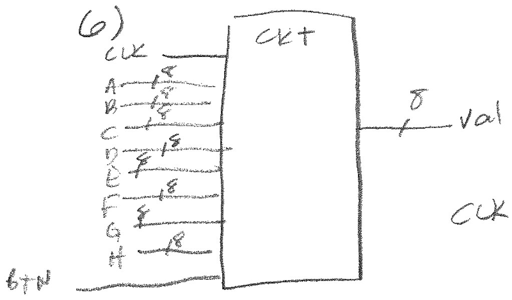


Notes: counter increments when up = 1; otherwise counter holds count

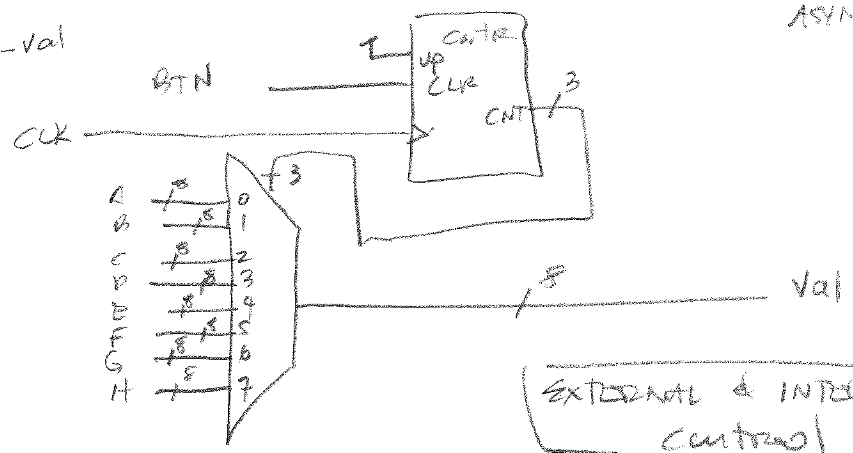
5)



6)

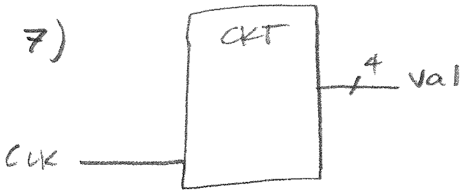


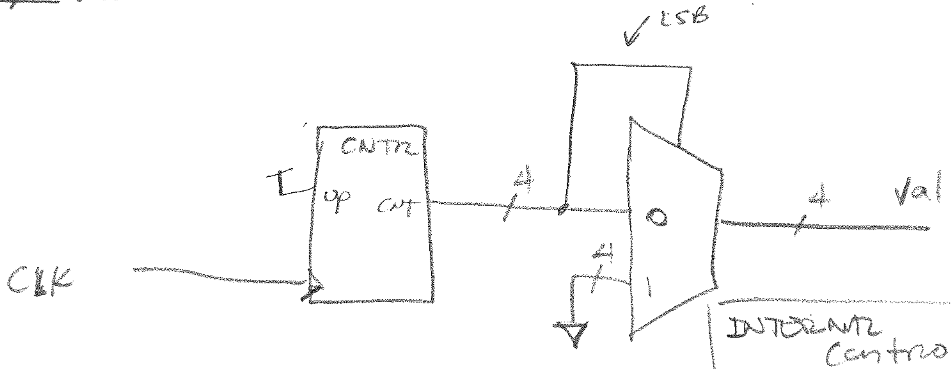
CLR signal is ASYNCHRONOUS



EXTERNAL & INTERNAL control

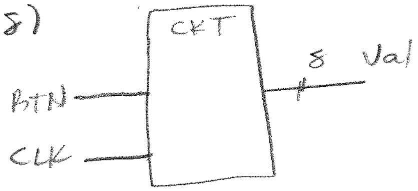
7)



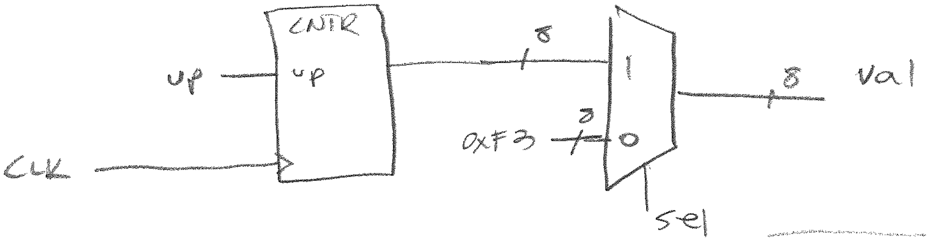


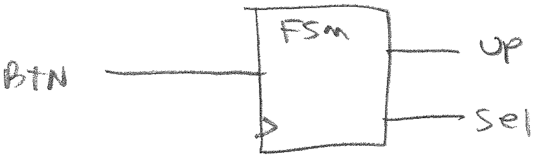
INTERNAL control

8)



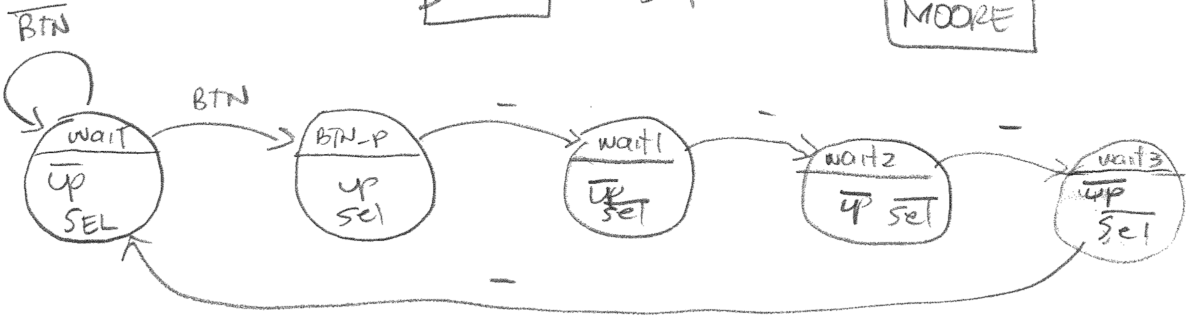
up = 1 ⇒ INCREMENTS up = 0 ⇒ HOLDS



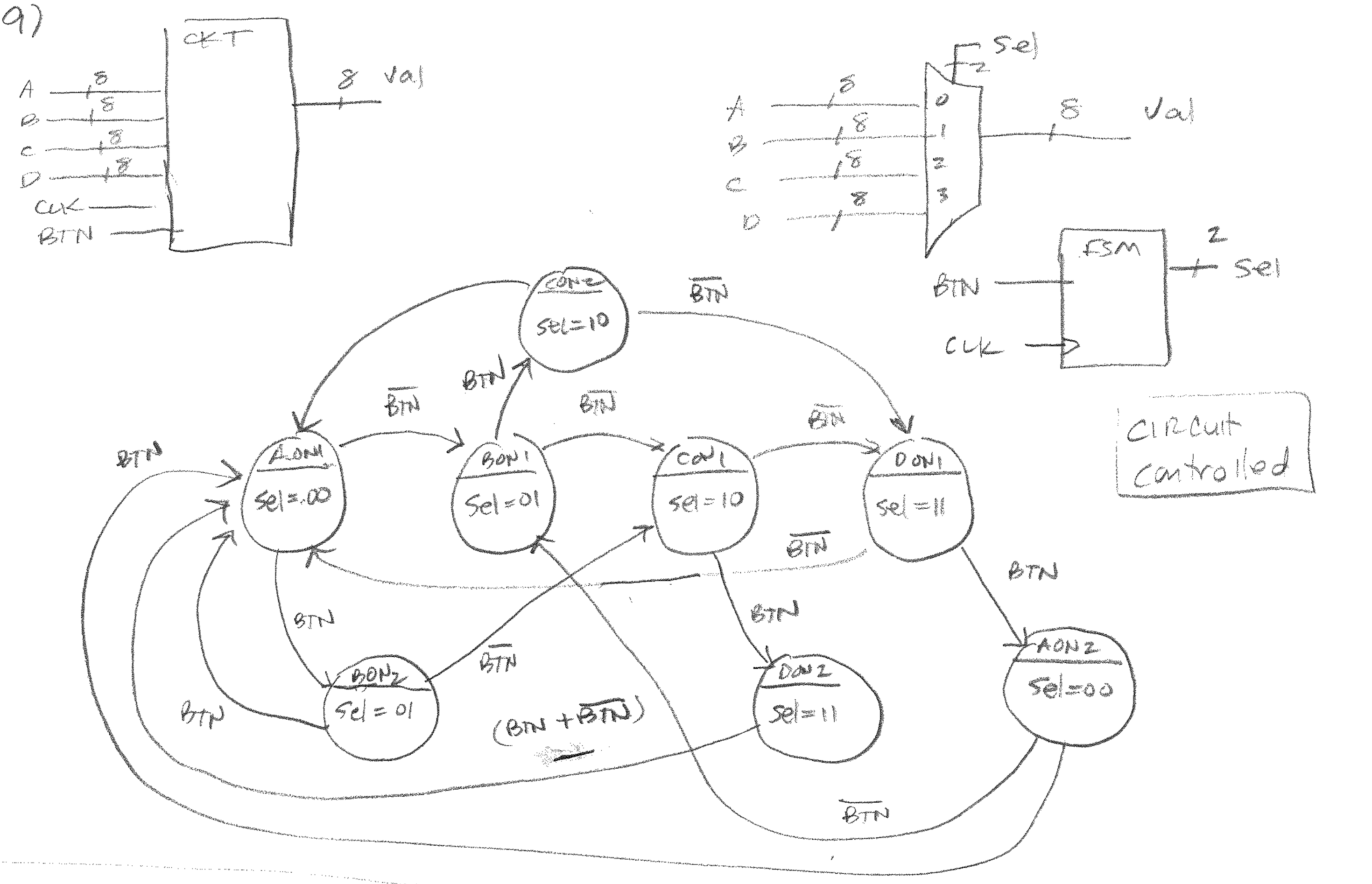


circuit controlled

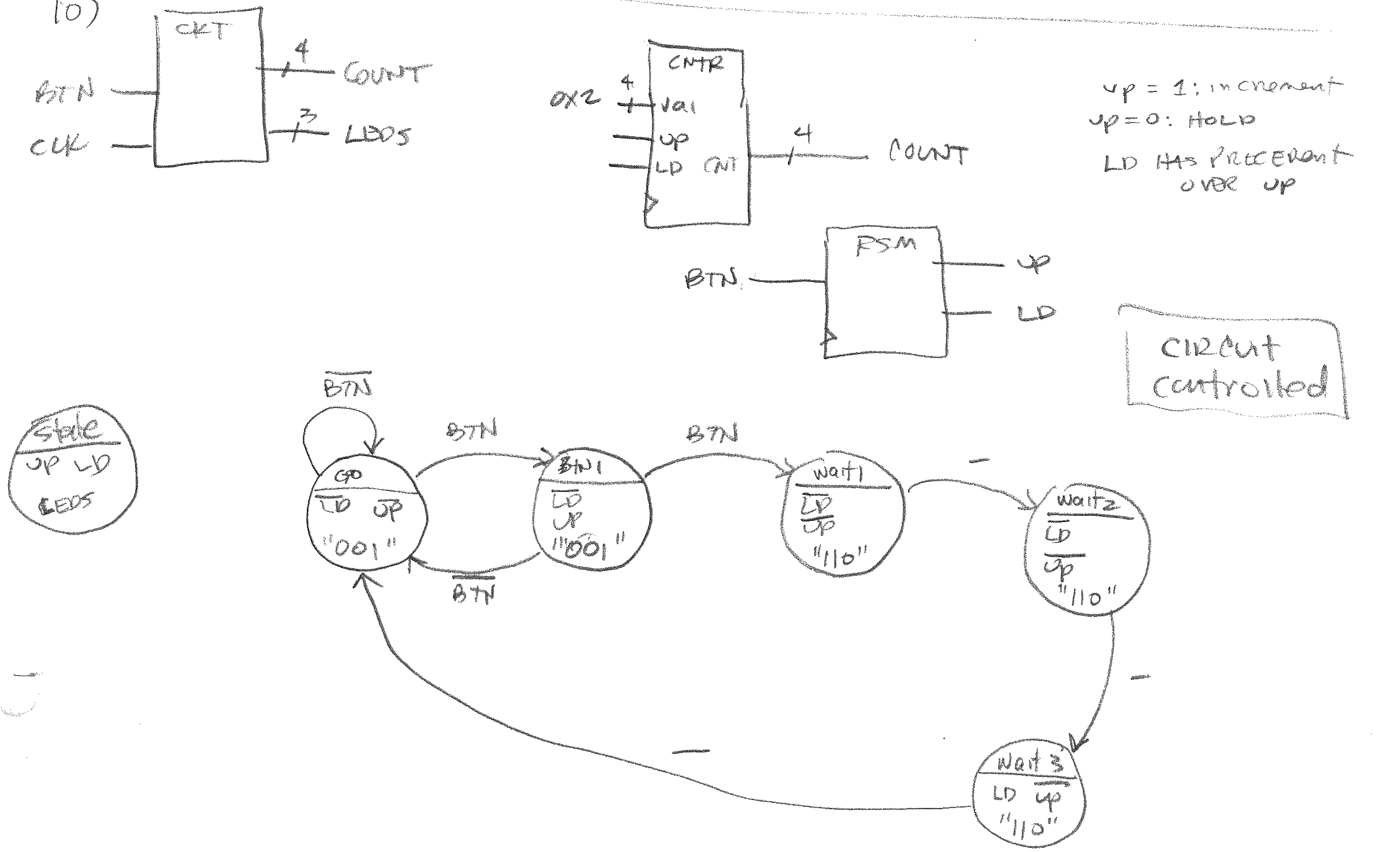
MOORE



9)

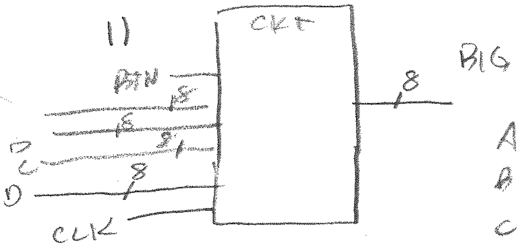


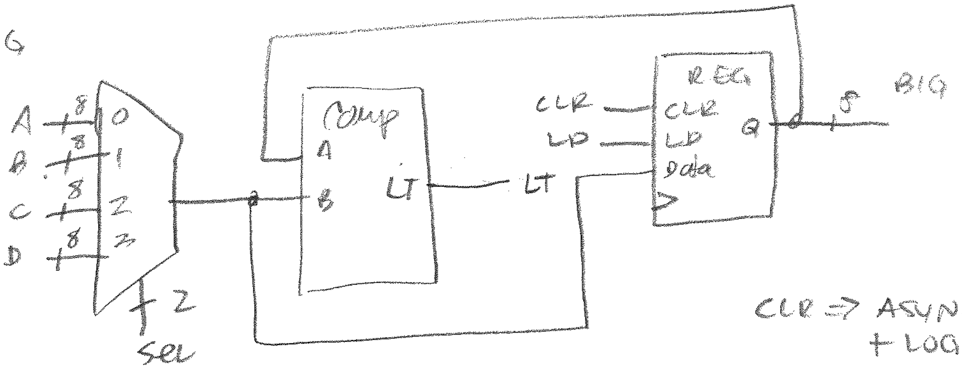
10)



up = 1: increment
up = 0: HOLD
LD HAS PRECEDENT OVER UP

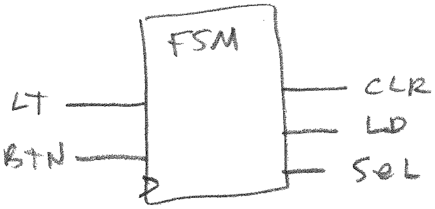
11)

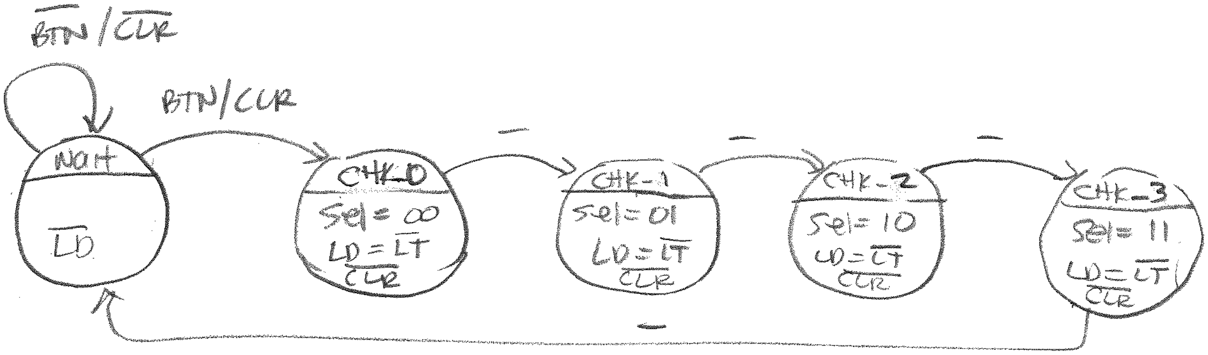




CLR => ASYNC + LOGIC

we LOADED THE REGISTER WITH "0" FOR GENERICITY! THIS SIMPLIFIES THE STATE DIAGRAM





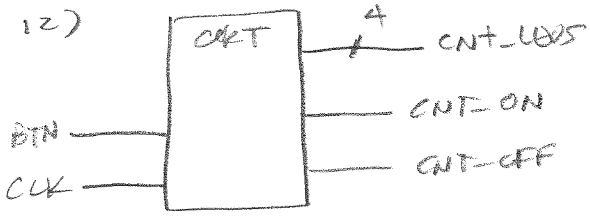
NOTE: Sel is NOT LISTED IN Wait state BECAUSE ITS A "Don't CARE" IN THAT state

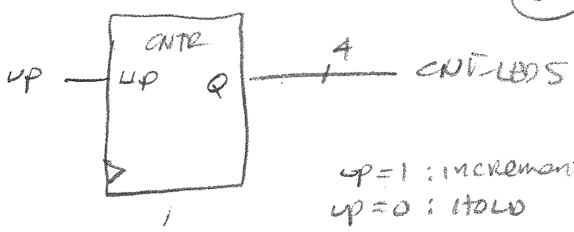
CLR is officially A MEALY-TYPE output AS INDICATED BY HOW IT IS LISTED IN Wait state. NOTE THAT IN OTHER states we LIST IT as we would A MOORE-TYPE output (IN THE STATE BUBBLE) TO make THE STATE DIAGRAM NEATER.

NOTE: clock signals control signals, AND status signals are not LISTED IN ORDER to make circuit schematic more neater.

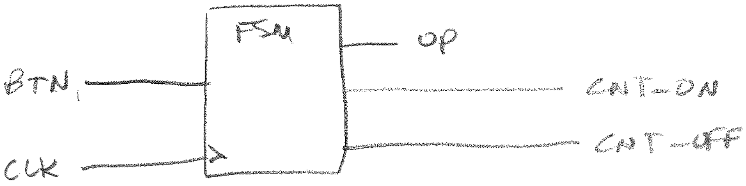
CIRCUIT controlled

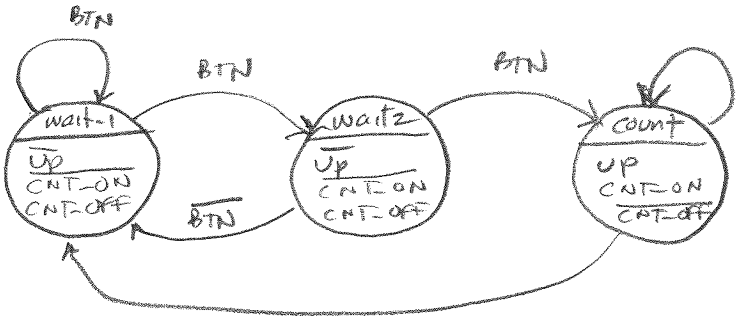
12)





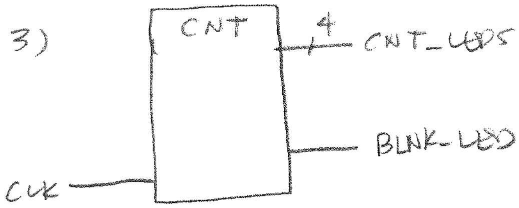
up=1 : increment
up=0 : HOLD

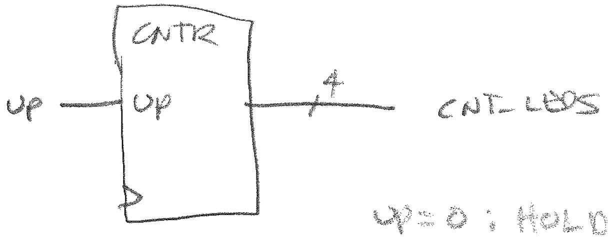




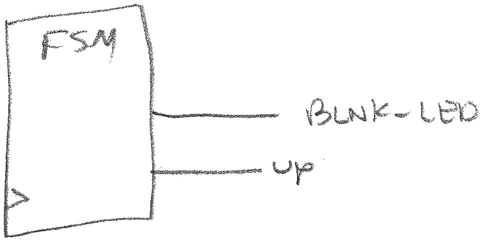
circuit controlled

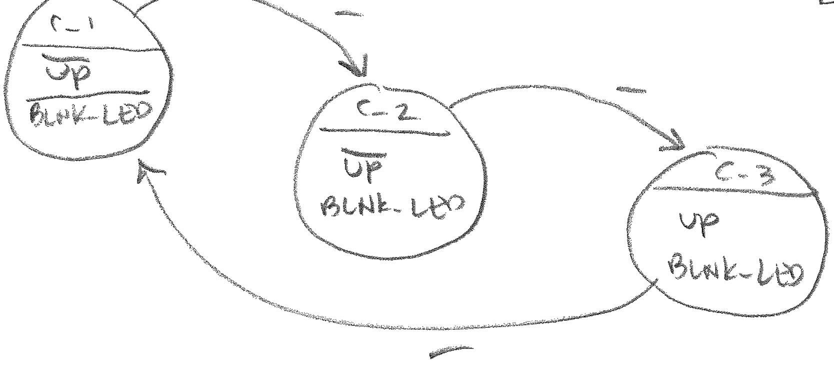
13)





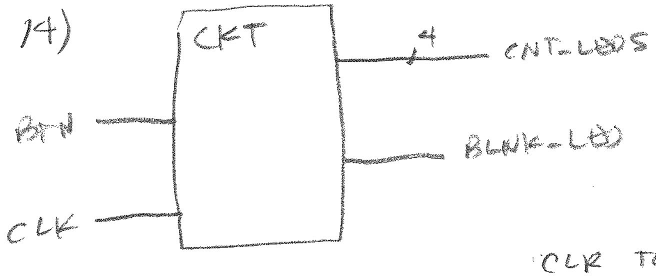
up=0 : HOLD
up=1 : increment

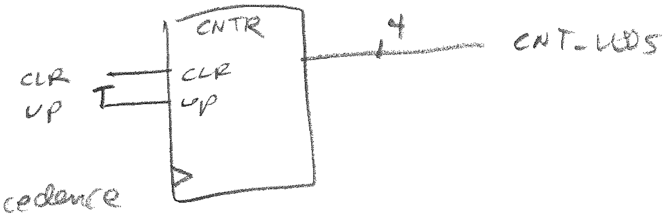




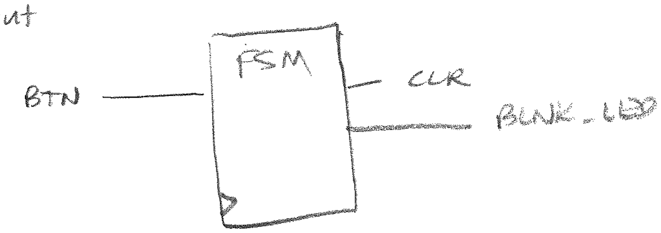
circuit controlled

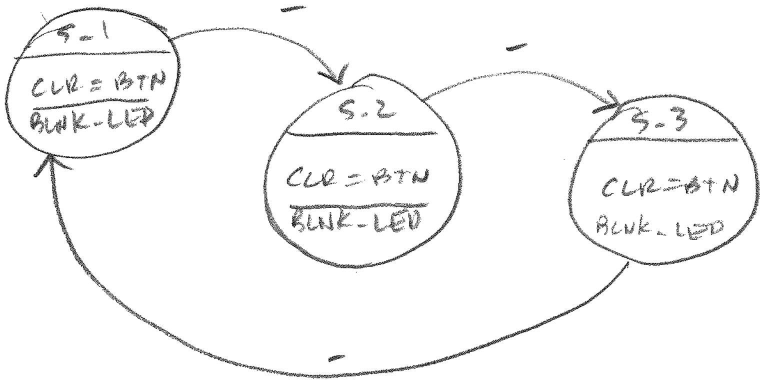
14)





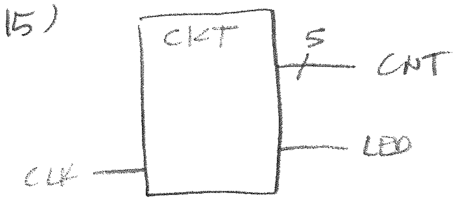
CLR takes precedence over up input

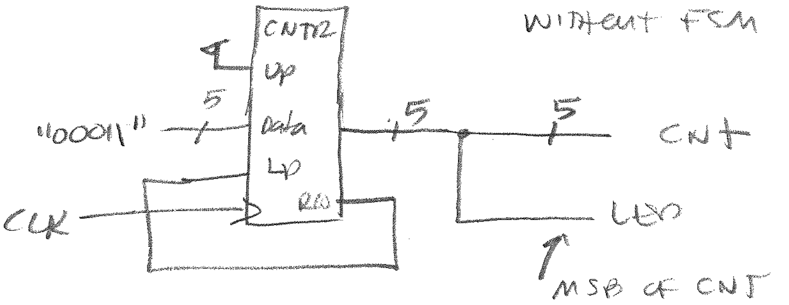


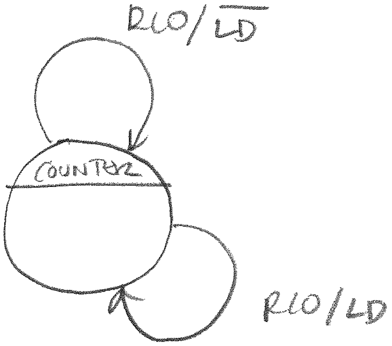


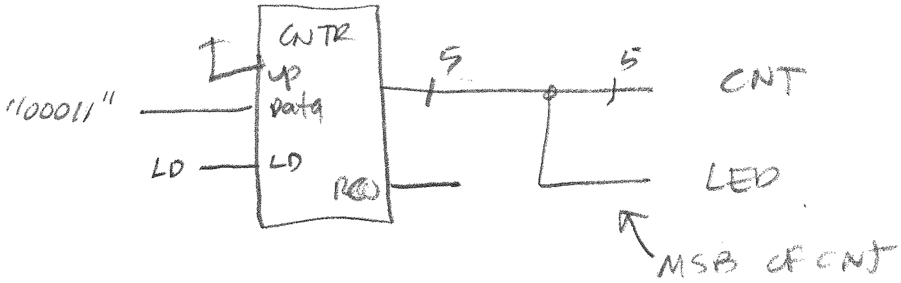
circuit controlled

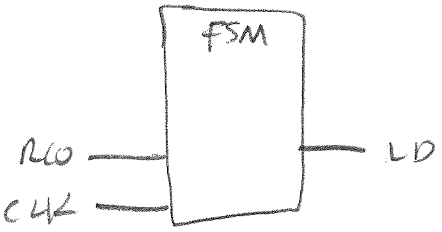
15)





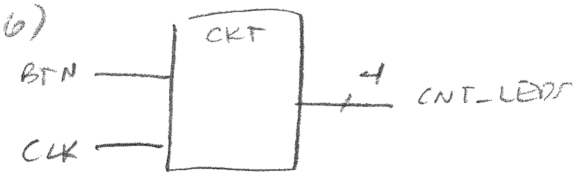


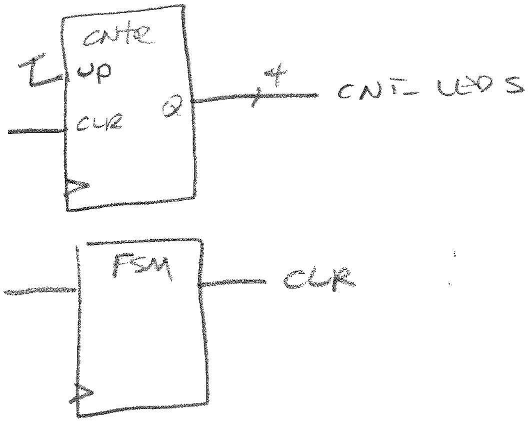


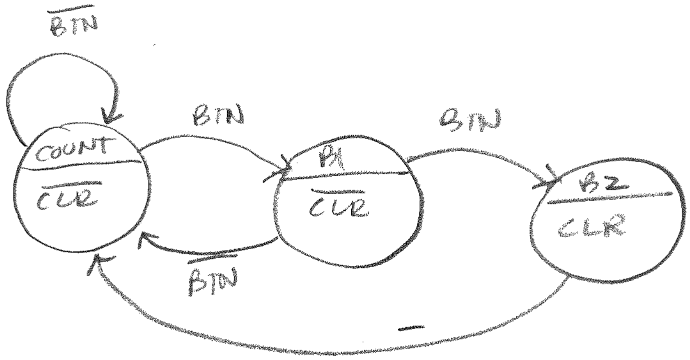


circuit controlled

16)



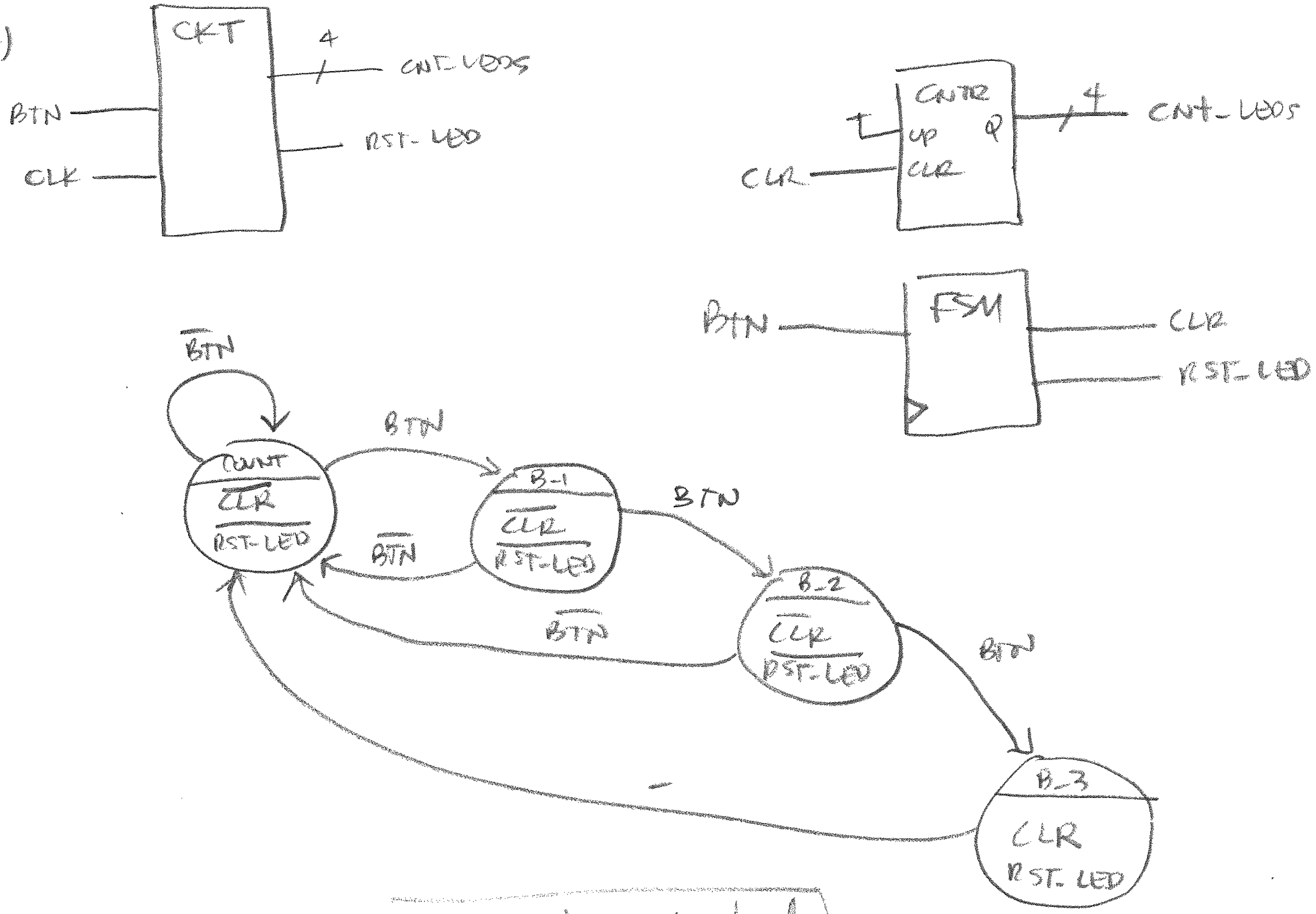




CLR input to counter has higher precedence than up input

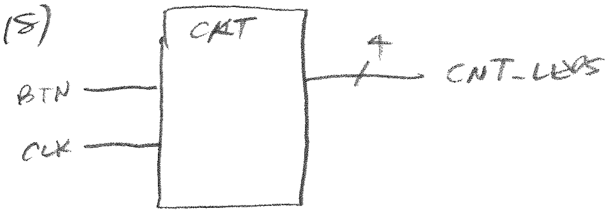
Circuit Controlled

17)

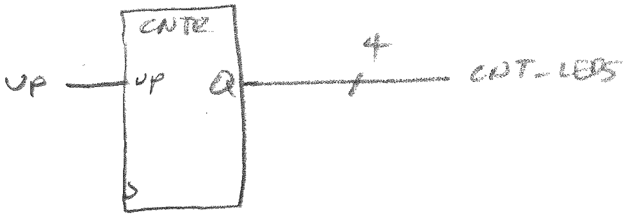


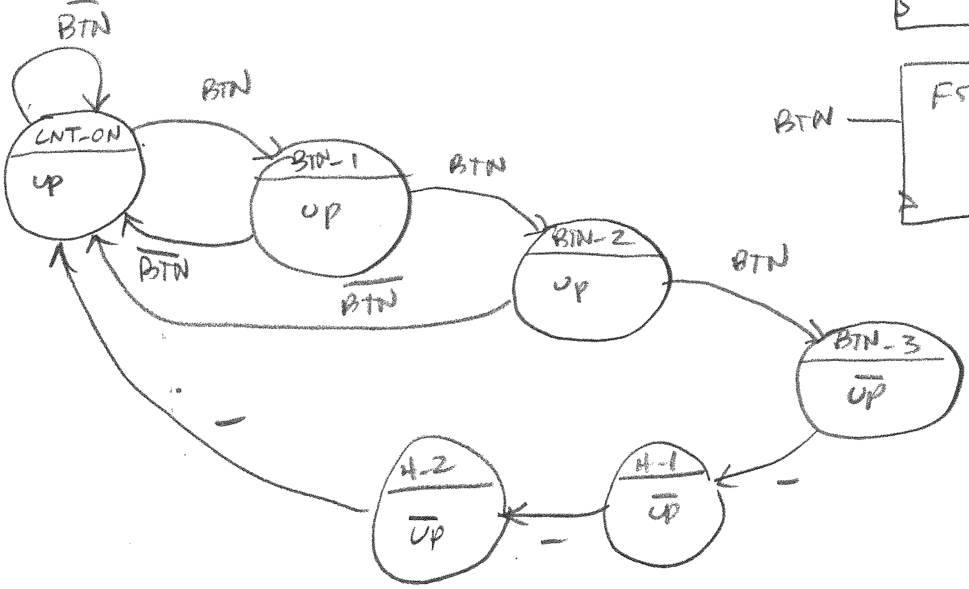
Circuit Controlled

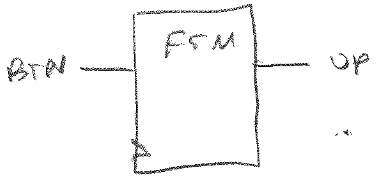
18)



up=1 : increment } COUNT
up=0 : HOLD

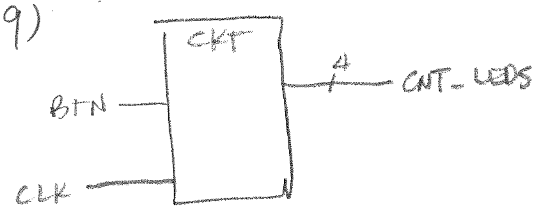




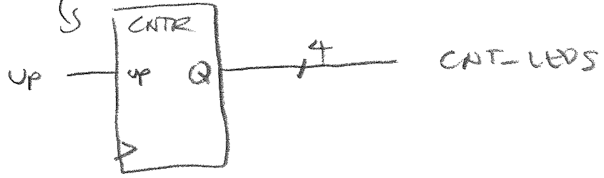


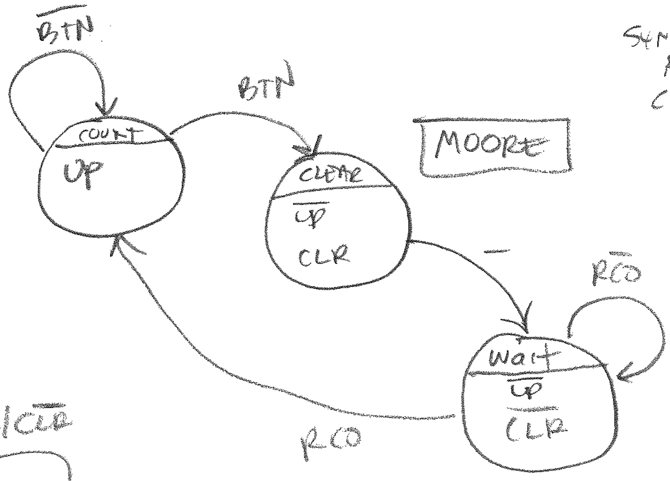
circuit controlled

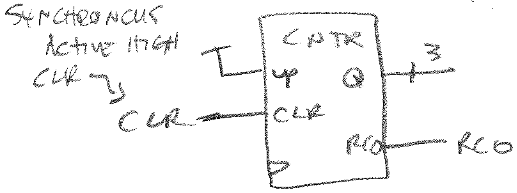
19)

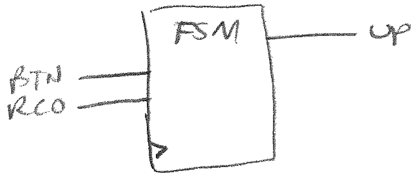


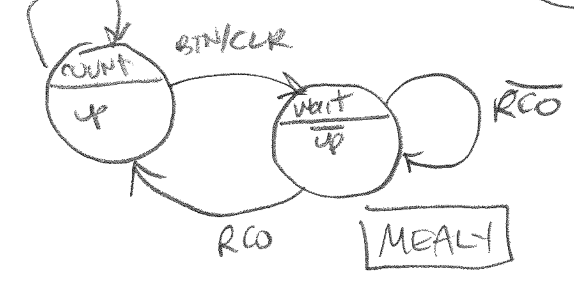
up=1 : increment
up=0 : HOLD





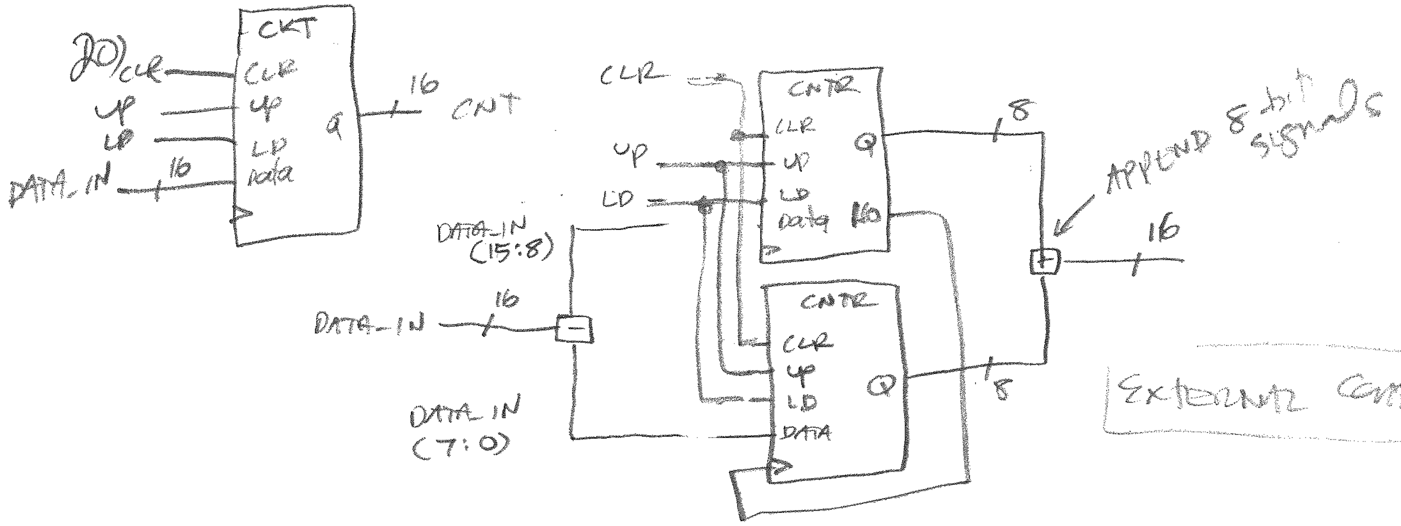






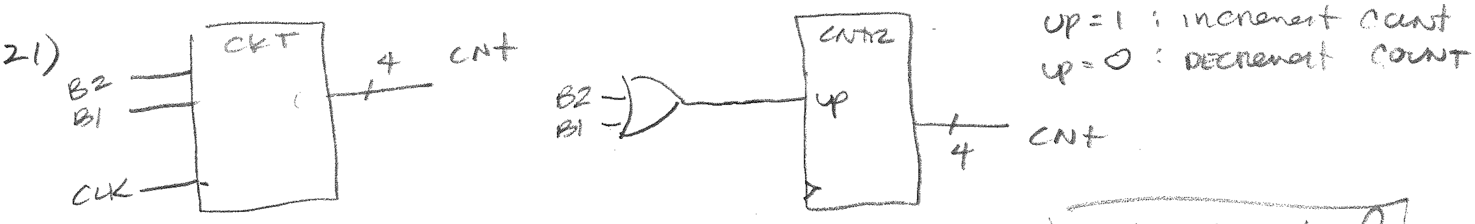
circuit controlled

20)



External control

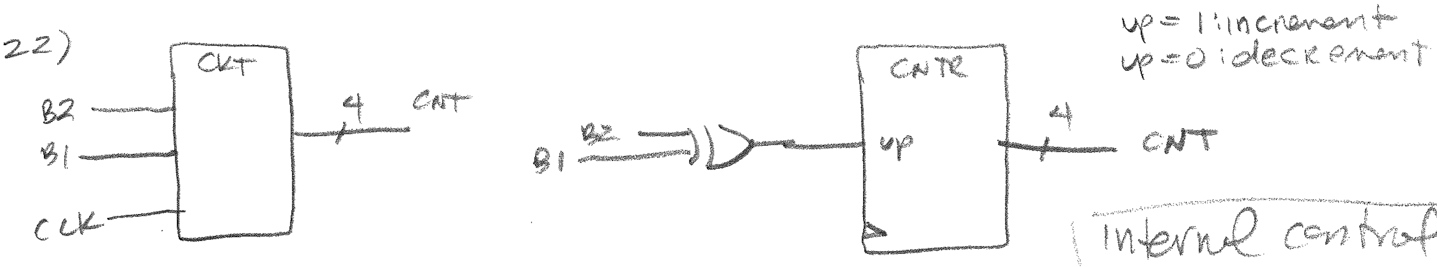
21)



up = 1 : increment count
up = 0 : DECREMENT COUNT

Internal control

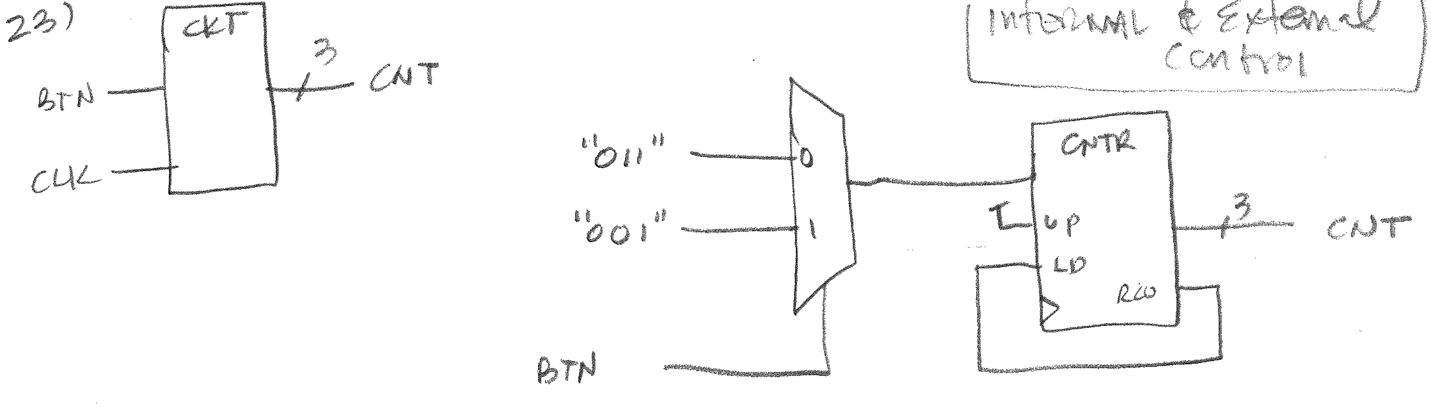
22)



up = 1 : increment
up = 0 : decrement

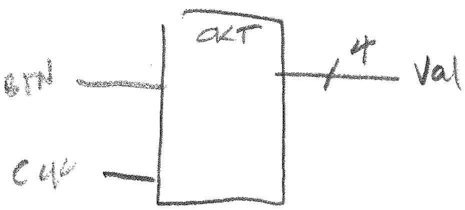
Internal control

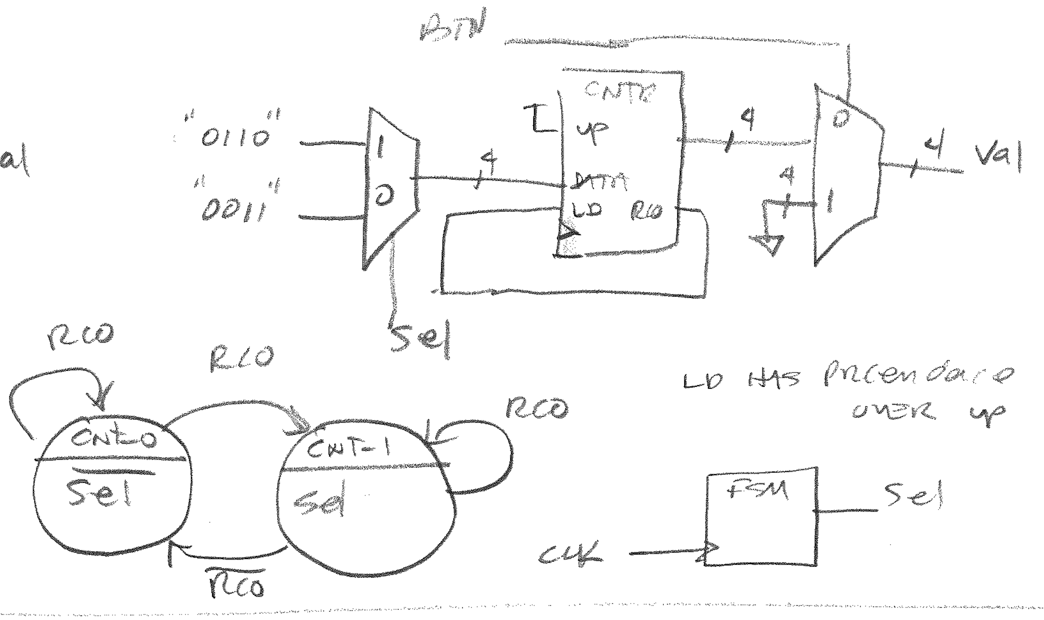
23)



INTERNAL & External control

24) Count 6-15
3-15

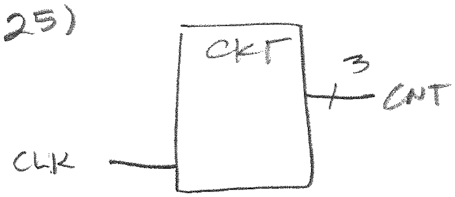


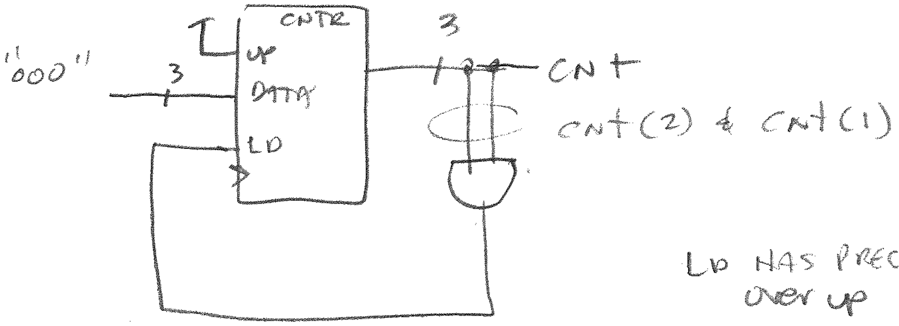


circuit AND internal control

LD HAS PRECEDENCE OVER up

25)



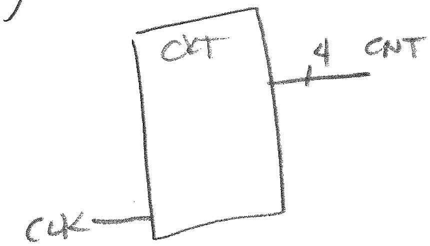


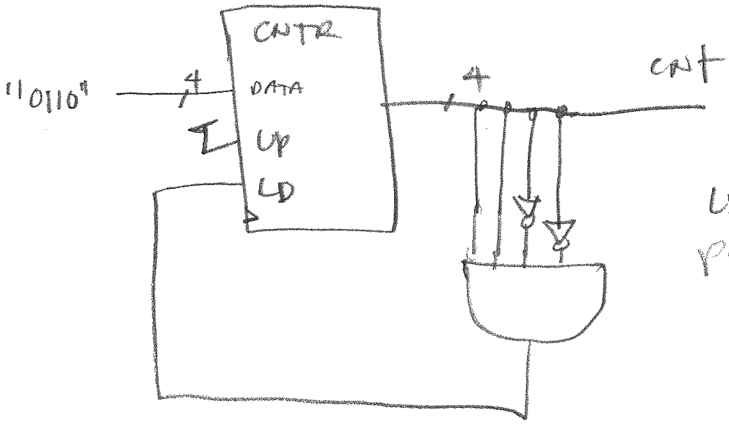
internal control

LD HAS PRECEDENCE over up

26)

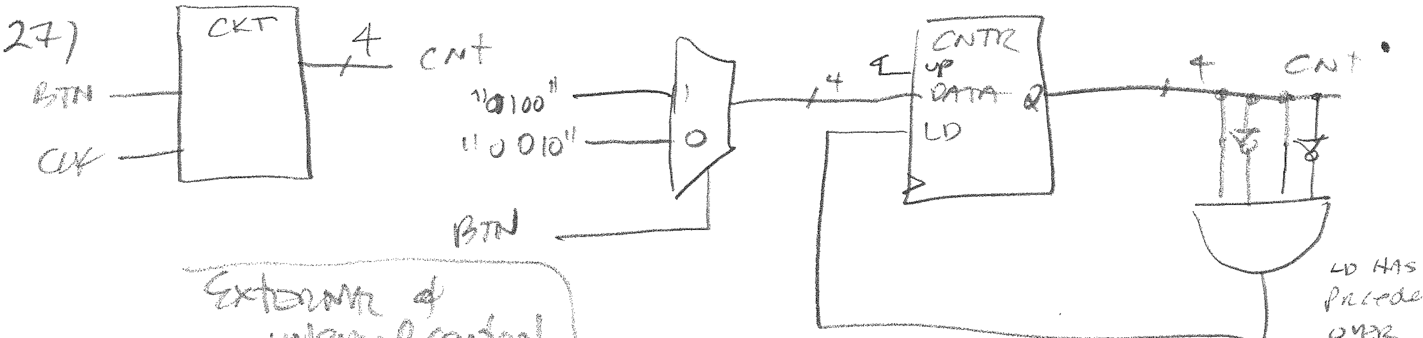
internal control





LD HAS PRECEDENCE Over up

27)



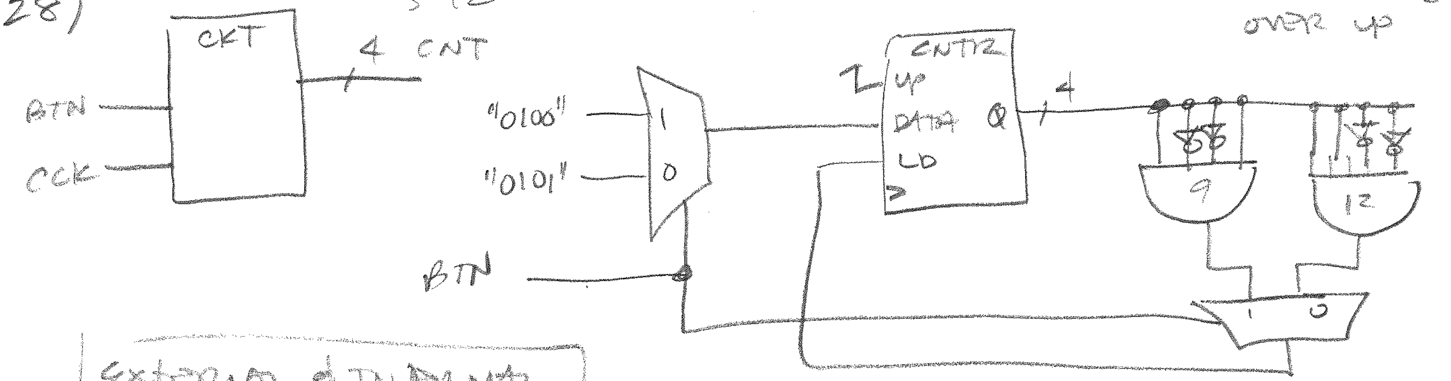
External & internal control

LD HAS precedence over up

28)

4-9
5-12

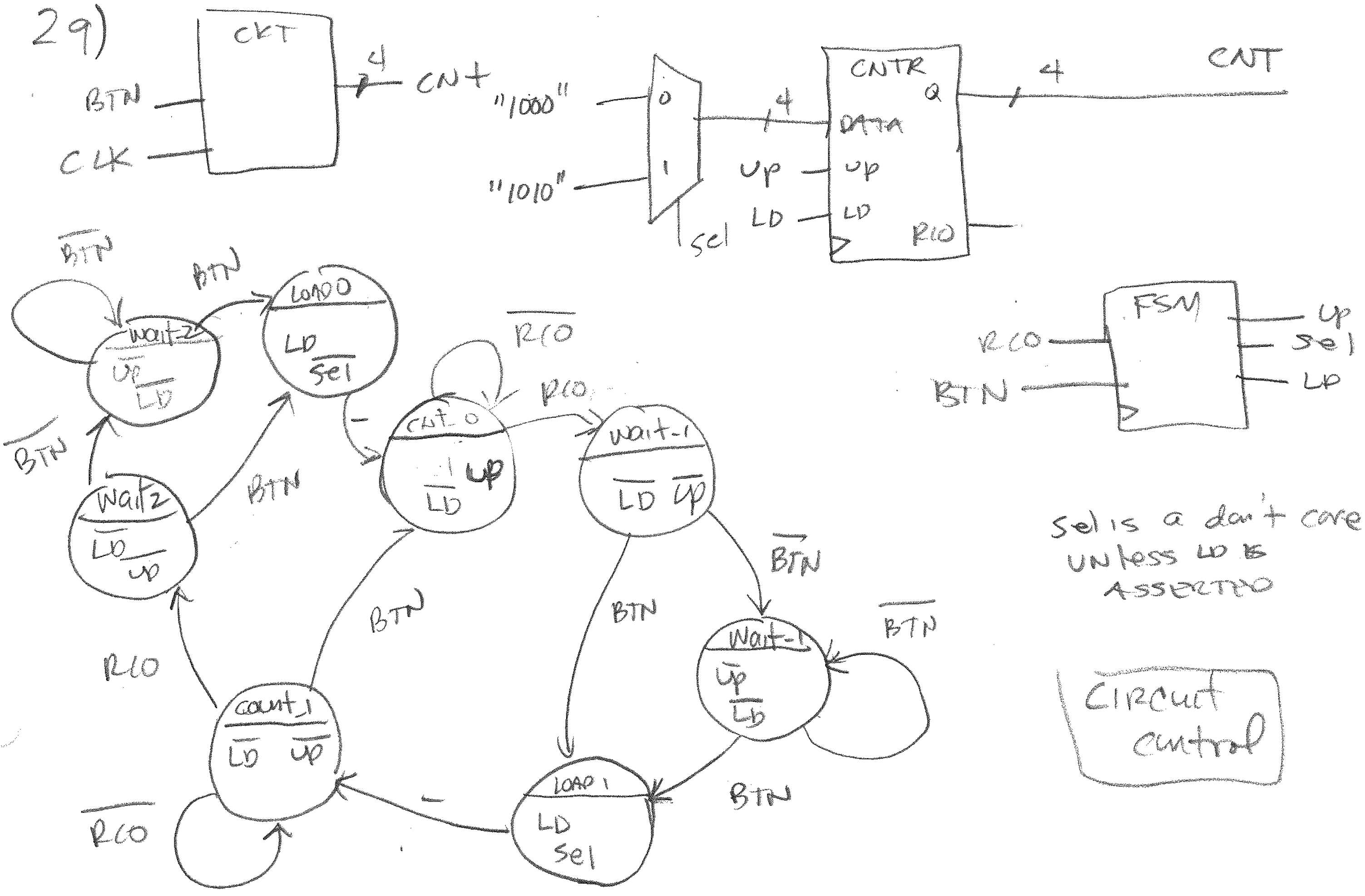
LD HAS precedence over up



External & INTERNAL control

8→15 or 10→15

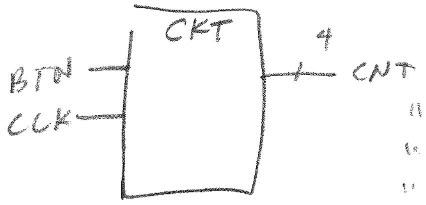
29)



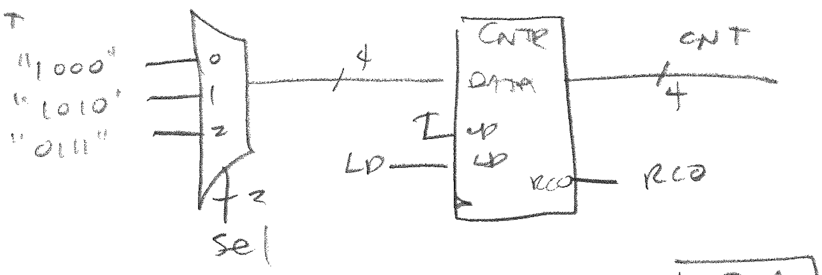
sel is a don't care unless LD is ASSERTED

Circuit control

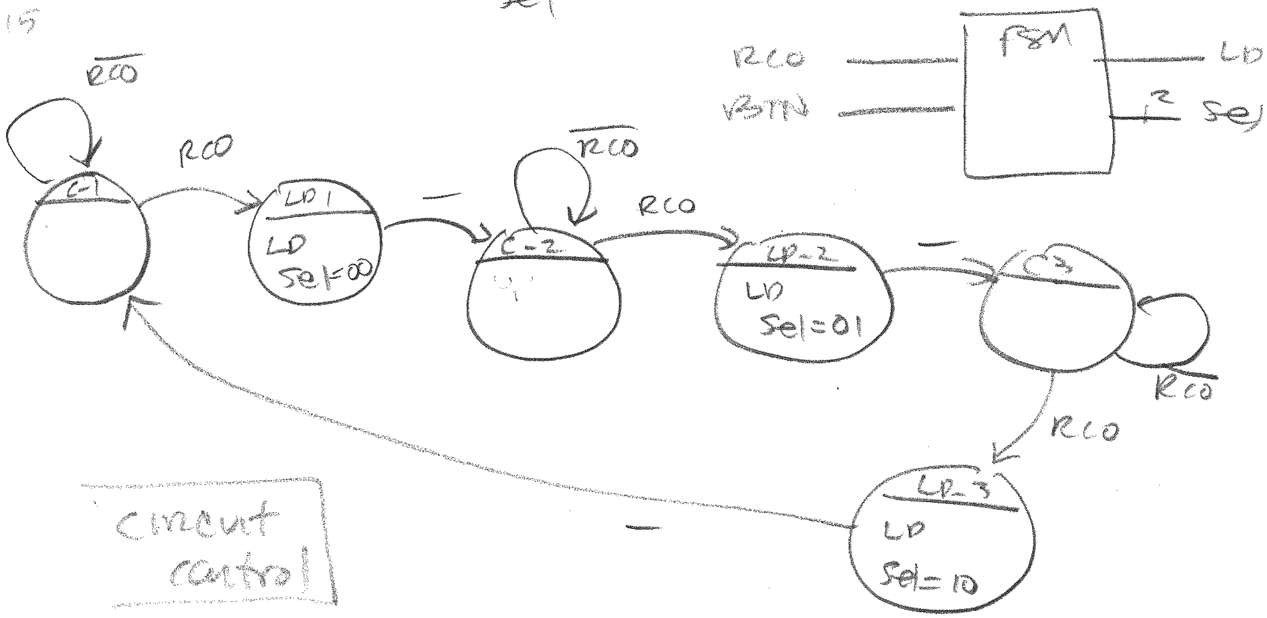
30)



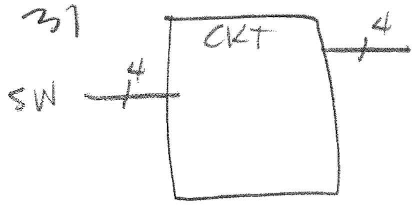
8 – 15
10 – 15
7 – 15

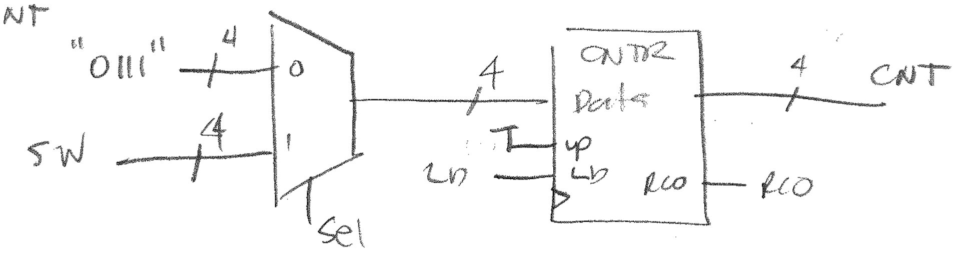


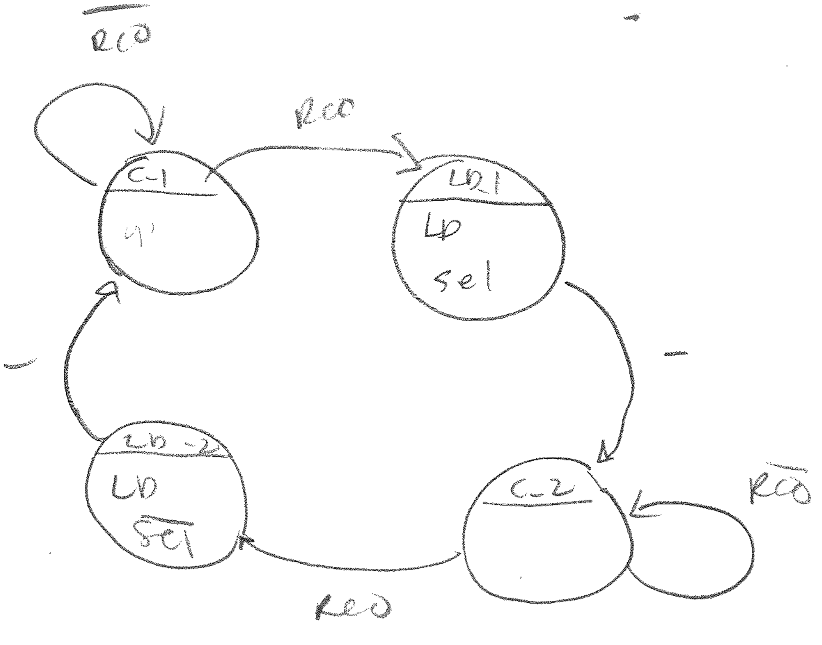
LD TAKES PRECEDENCE OVER UP



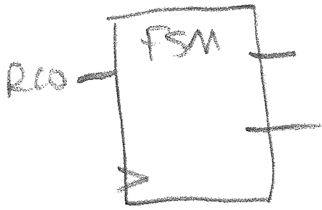
31)





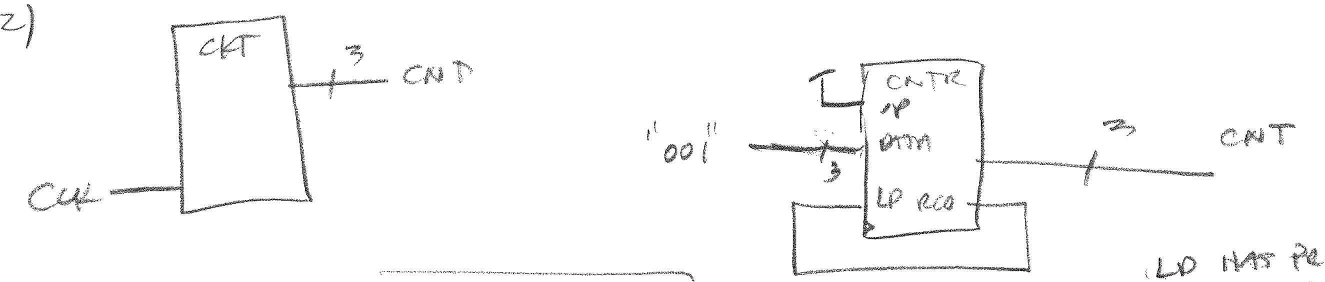


LD TAKES PRECEDENCE OVER UP



circuit controlled

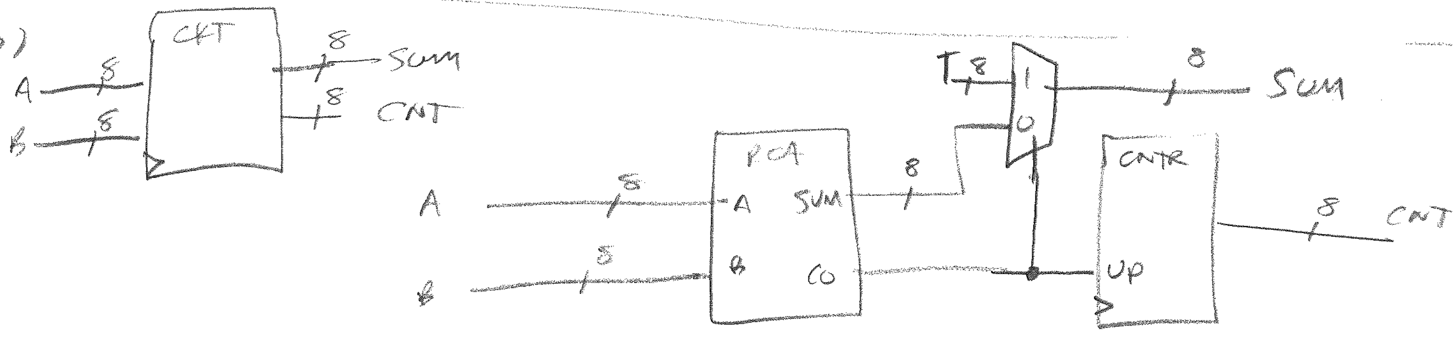
32)



LD HAS PRECEDENCE OVER UP

internal control

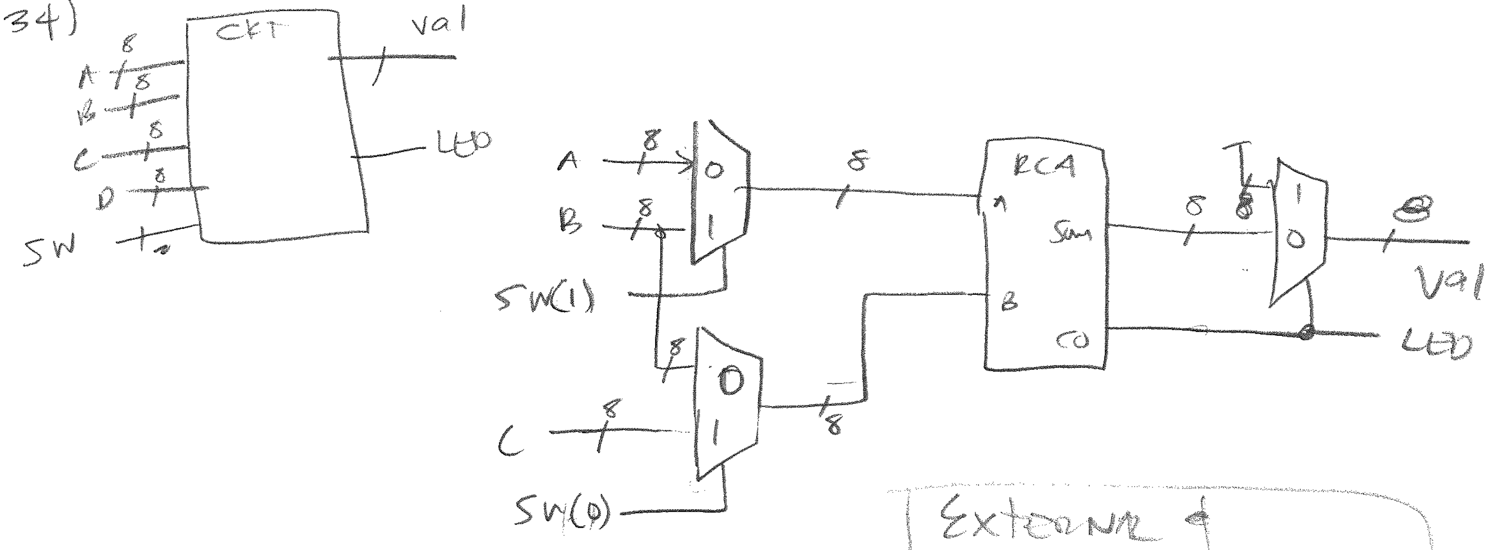
33)



up=1 ⇒ increment
up=0 ⇒ HOLD

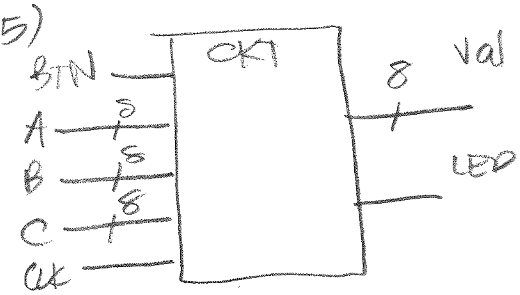
internal control

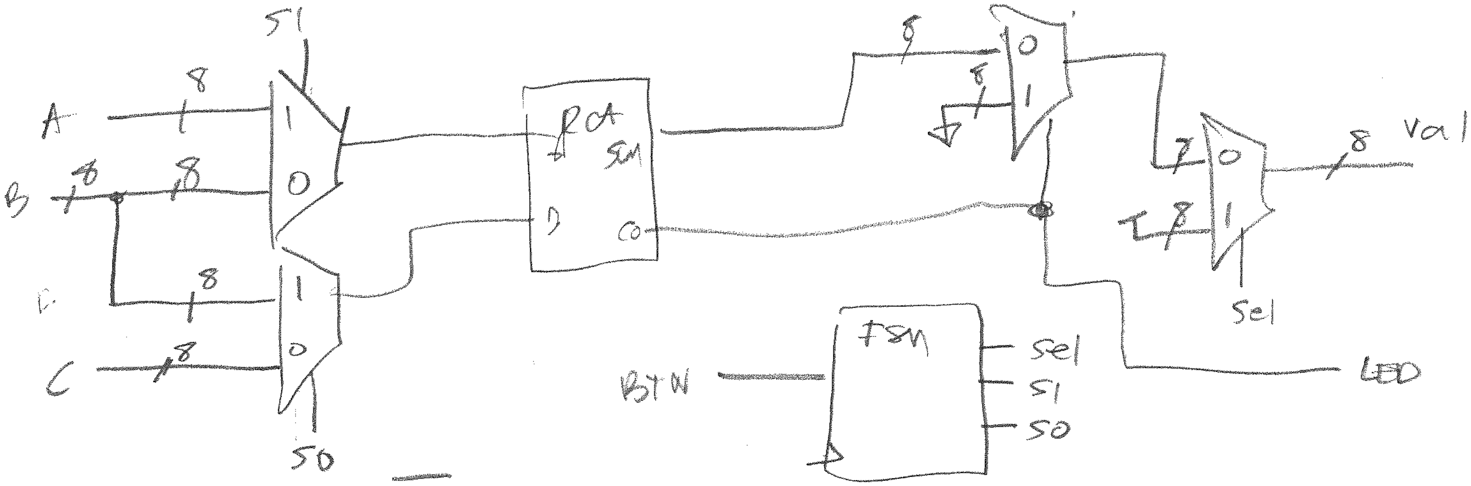
34)

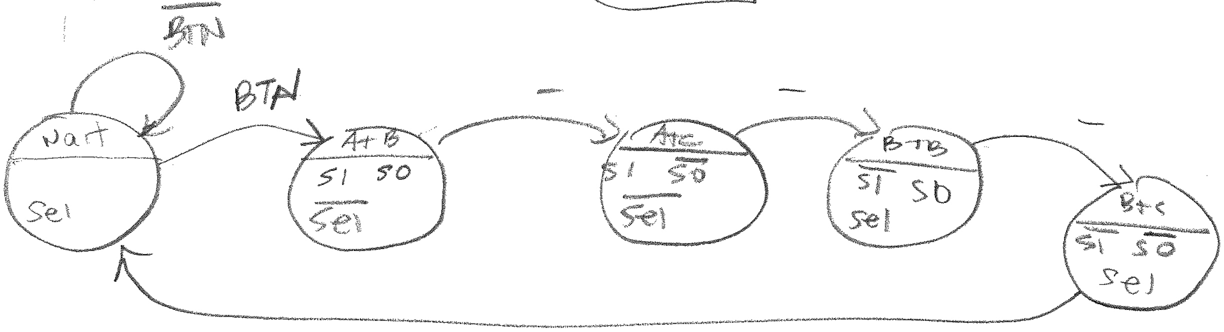


External & internal control

35)

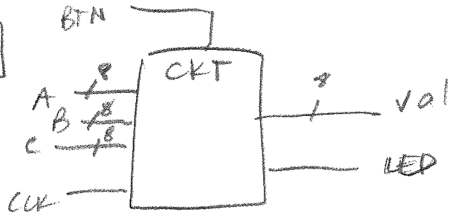


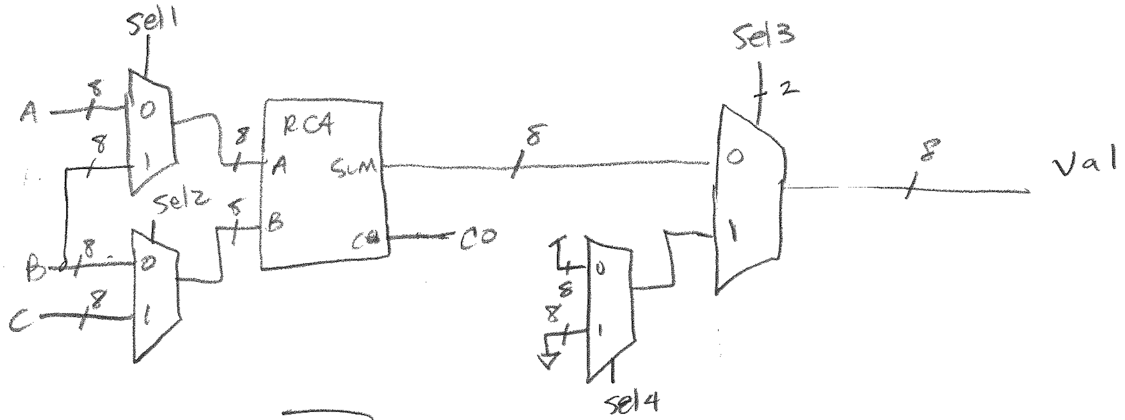


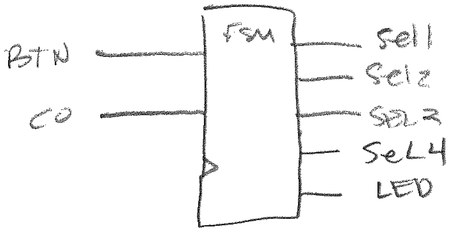


circuit controlled

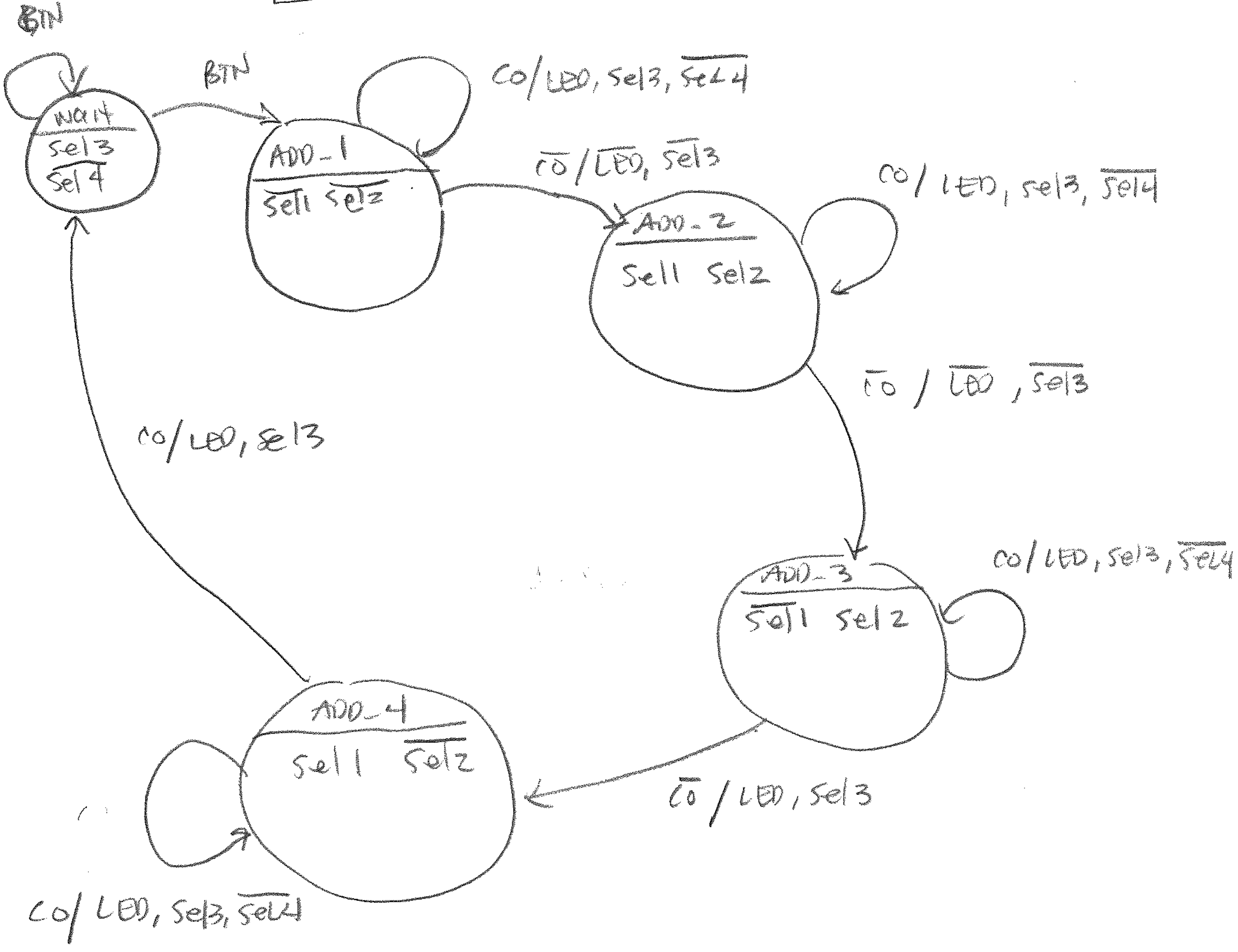
36



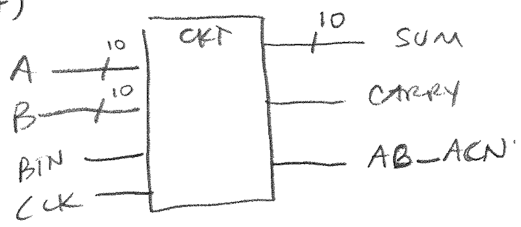




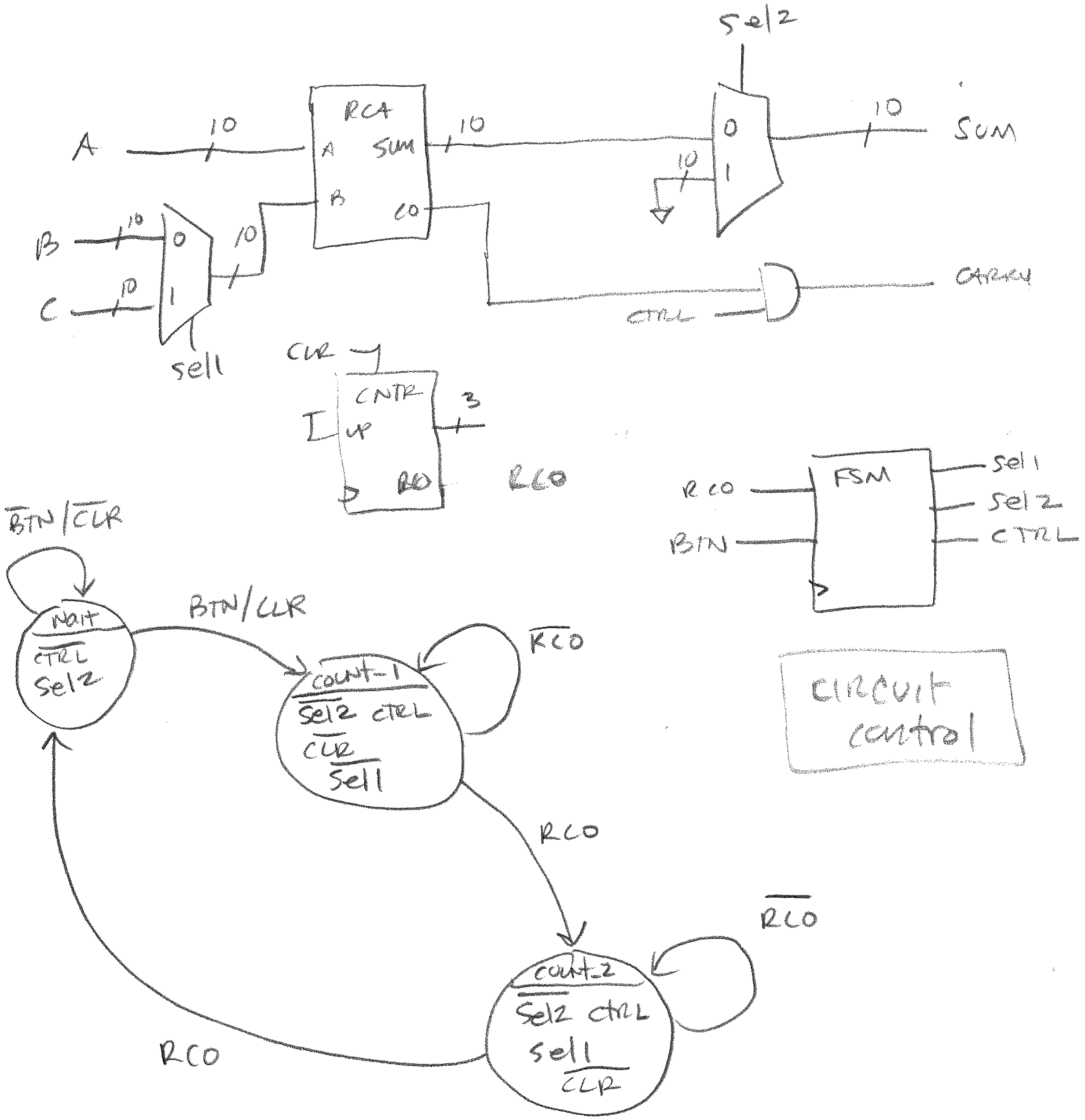
circuit controlled


$\overline{BTN}$
BTN
WAIT
Sel3
$\overline{Sel4}$
CO / LED, Sel3, $\overline{Sel4}$
ADD_1
$\overline{Sel1}$ $\overline{Sel2}$
$\overline{CO}$ / $\overline{LED}$, $\overline{Sel3}$
CO / LED, Sel3, $\overline{Sel4}$
ADD_2
Sel1 Sel2
$\overline{CO}$ / $\overline{LED}$, $\overline{Sel3}$
CO / LED, Sel3
ADD_3
$\overline{Sel1}$ Sel2
CO / LED, Sel3, $\overline{Sel4}$
ADD_4
Sel1 $\overline{Sel2}$
$\overline{CO}$ / LED, Sel3
CO / LED, Sel3, $\overline{Sel4}$

37)

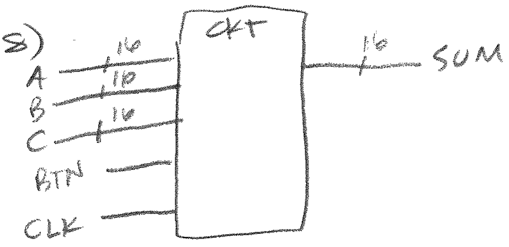


CKT
A
10
B
10
BTN
CLK
10
SUM
CARRY
AB_ACN

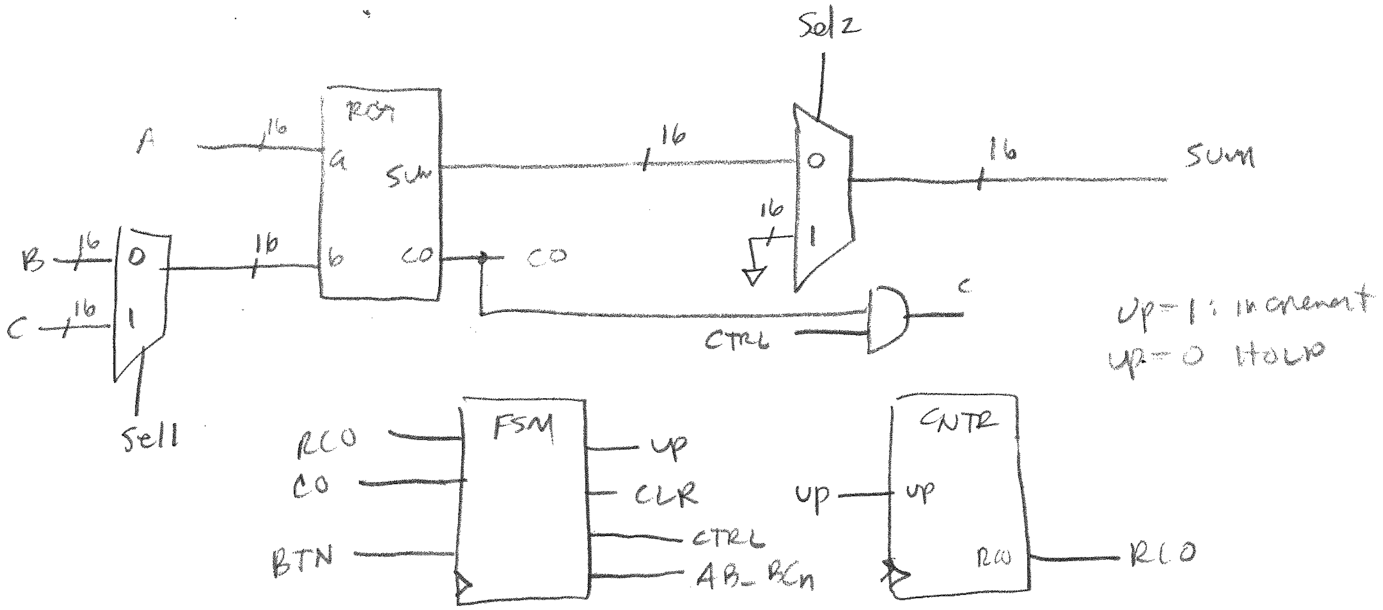


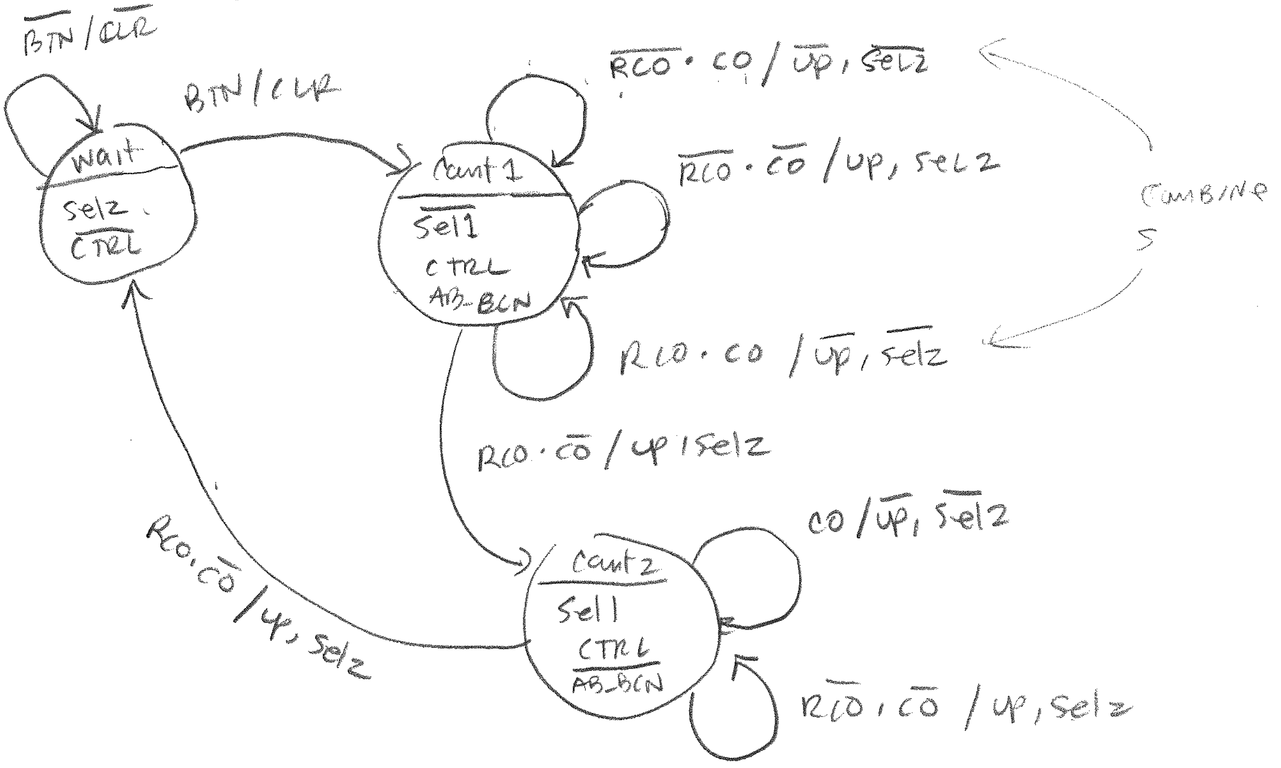
sel2
RCA
A
10
A
SUM
10
0
1
10
10
SUM
B
10
0
10
B
CO
C
10
1
CARRY
CTRL
sel1
CLR
CNTR
1
up
3
RCO
RCO
RCO
FSM
sel1
sel2
CTRL
BTN
BTN/CLR
BTN/CLR
wait
CTRL
sel2
count_1
sel2 ctrl
CLR
sel1
RCO
RCO
count_2
sel2 ctrl
sel1
CLR
RCO
RCO
circuit control

38)

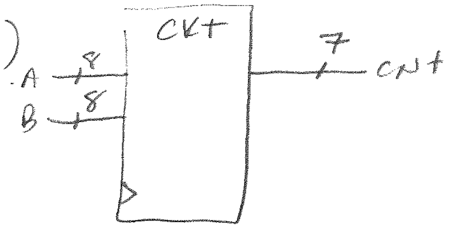


Circuit controlled



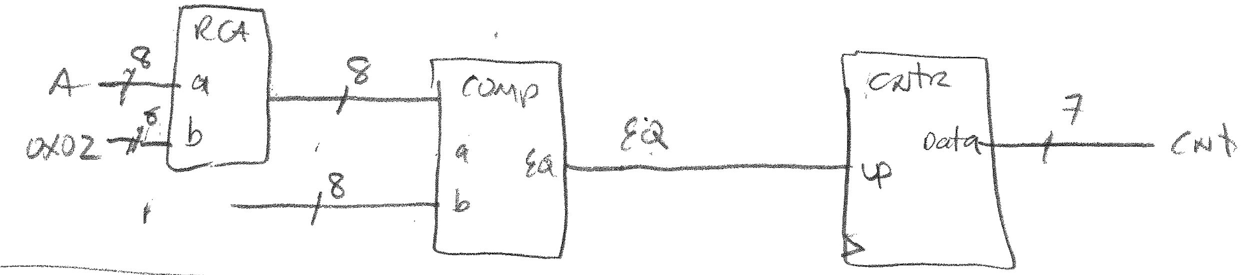

$\overline{BTN}$ / $\overline{CLR}$
BTN / CLR
Wait
Sel2
$\overline{CTRL}$
$\overline{RCO}$ · CO / $\overline{up}$, $\overline{Sel2}$
$\overline{RCO}$ · $\overline{CO}$ / up, Sel2
RCO · CO / $\overline{up}$, Sel2
Combine
Count 1
$\overline{Sel1}$
CTRL
AB_BCN
RCO · $\overline{CO}$ / up, Sel2
CO / $\overline{up}$, $\overline{Sel2}$
Count 2
Sel1
CTRL
$\overline{AB\_BCN}$
$\overline{RCO}$ · $\overline{CO}$ / up, Sel2
RCO · $\overline{CO}$ / up, Sel2

39)

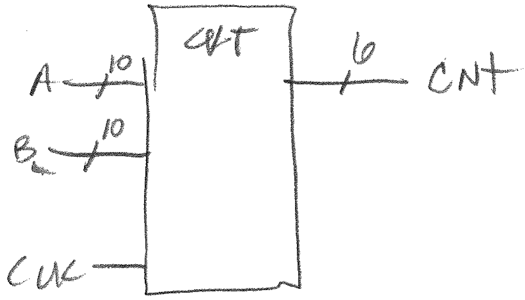


INTERNAL control

up=1: increment
up=0: HOLD

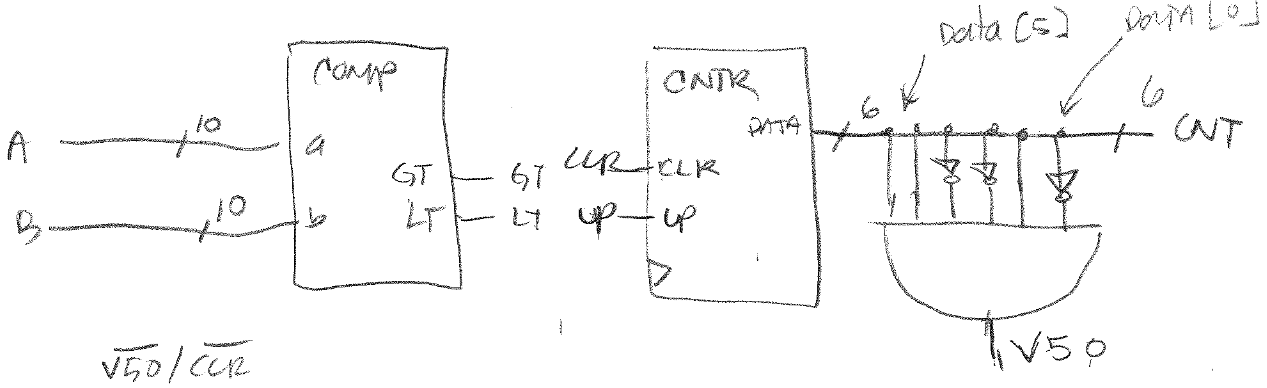


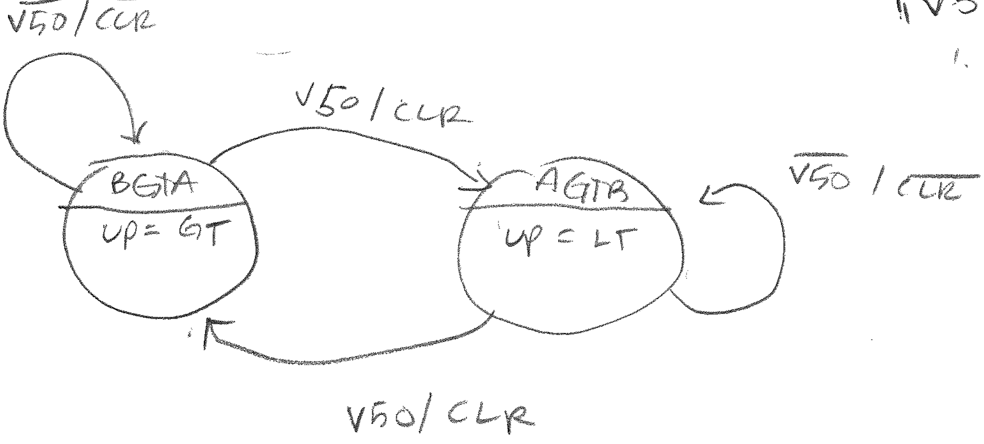
40)



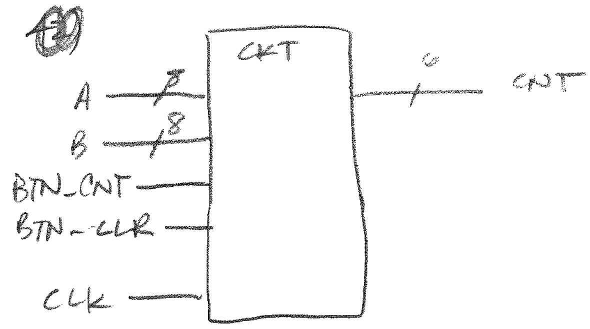
Circuit controlled

32 00110010



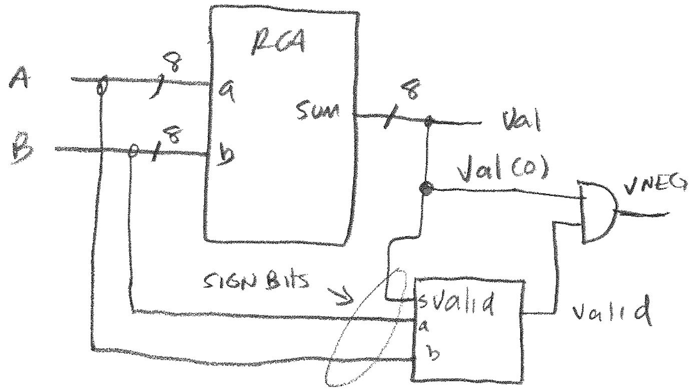

$\overline{V50}/\overline{CLR}$
BGTA
up=GT
V50/CLR
AGTB
up=LT
$\overline{V50}/\overline{CLR}$
V50/CLR

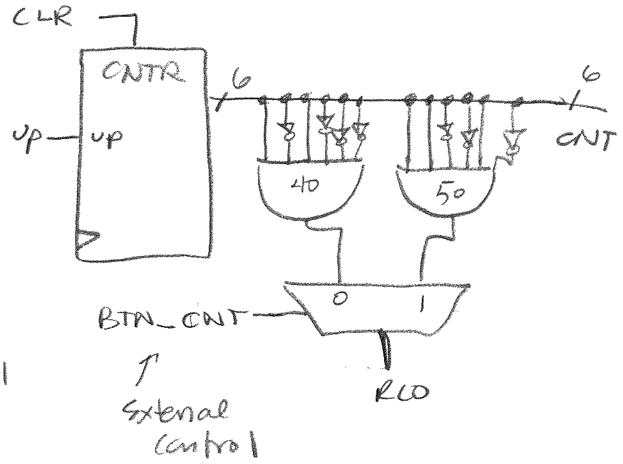
41)

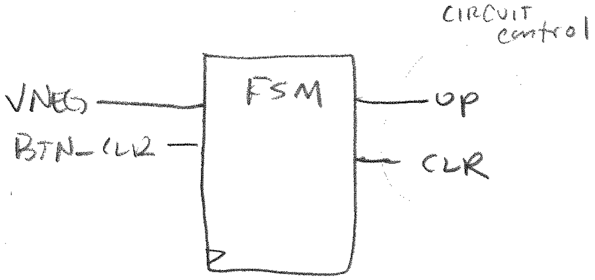


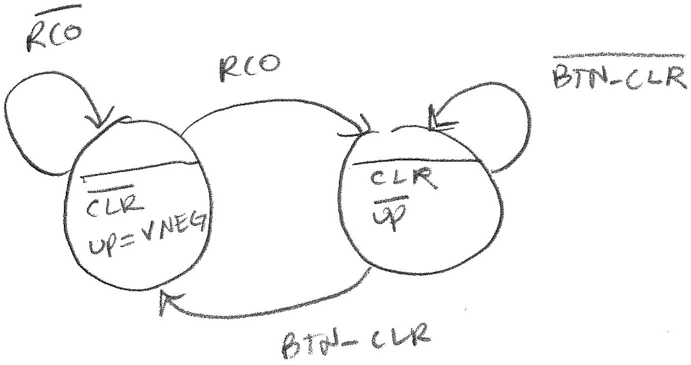
40 = 0x28 = 101000
50 = 0x32 = 110010

up = 1 : increment
up = 0 : HOLD







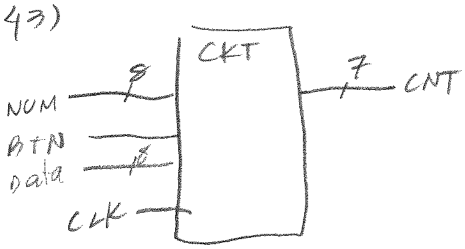


CIRCUIT HAS EXTERNAL AND CIRCUIT (FSM) CONTROL

DECODER spec

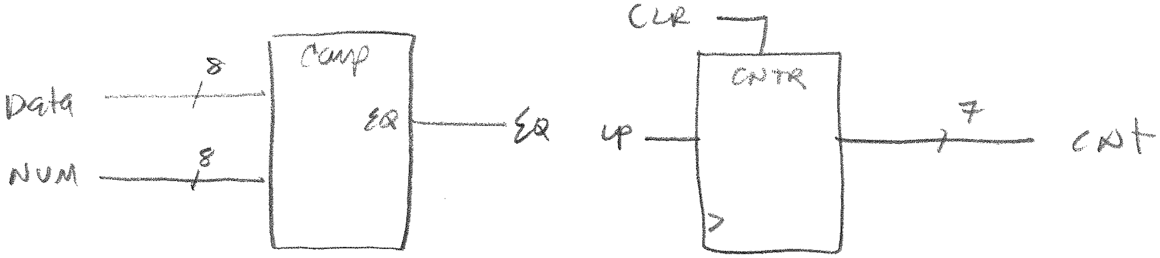
a	b	s	Valid
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

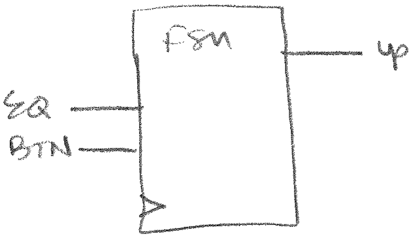
43)

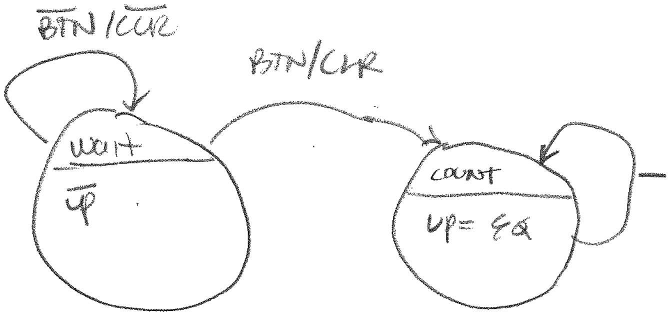


Data is incoming DATA

NUM is the value we check for



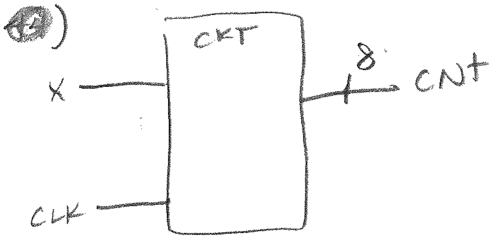




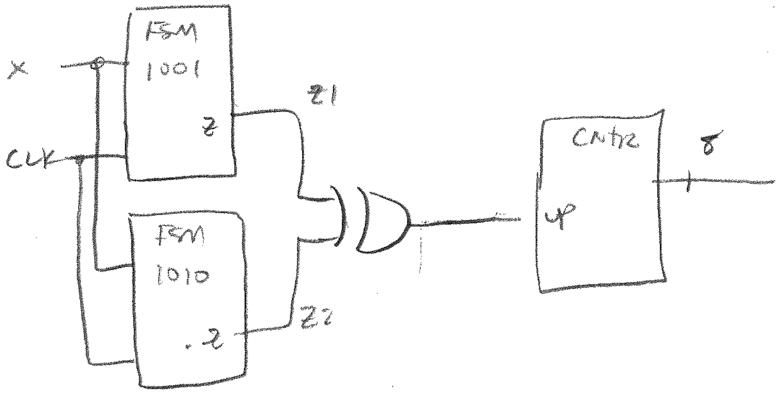
circuit has circuit control (FSM)

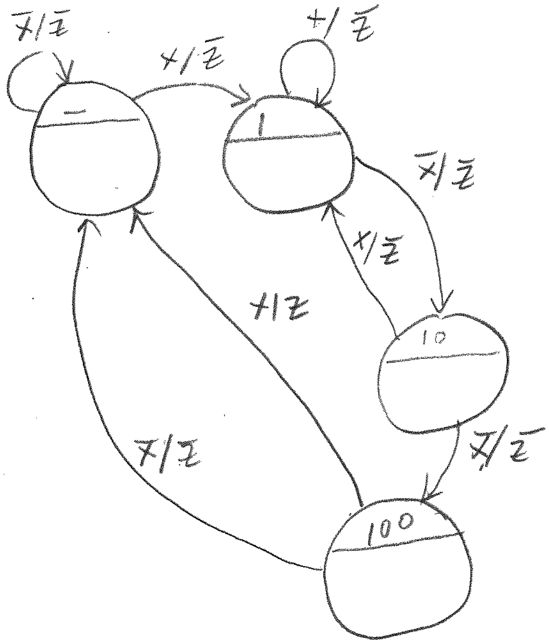
4.3)

[illegible])



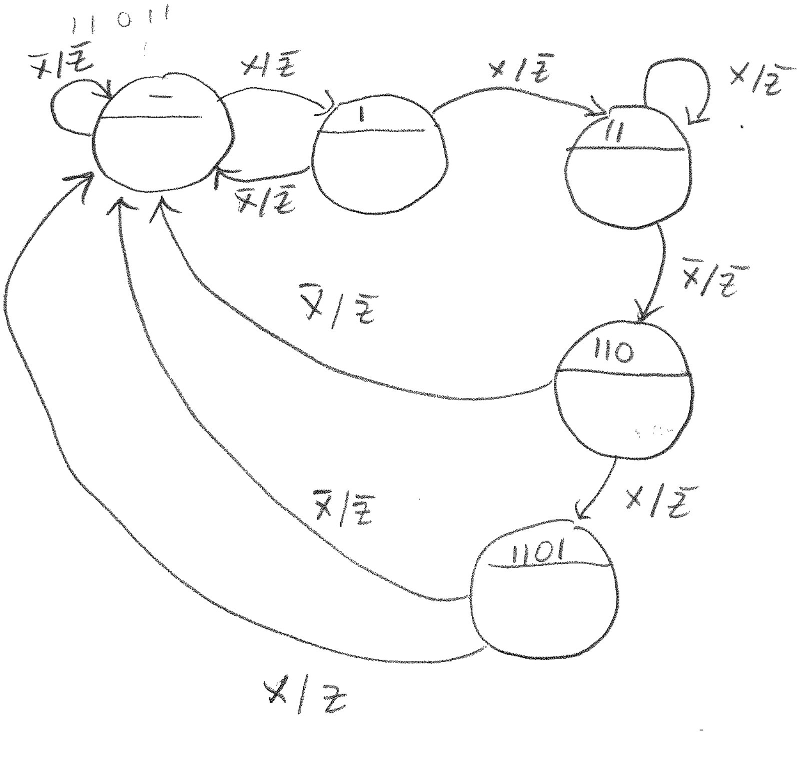
up = 1 : increment
up = 0 : Decrement





FSM: 1001

circuit HAS [illegible]
circuit AND [illegible]
internal control

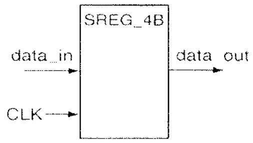


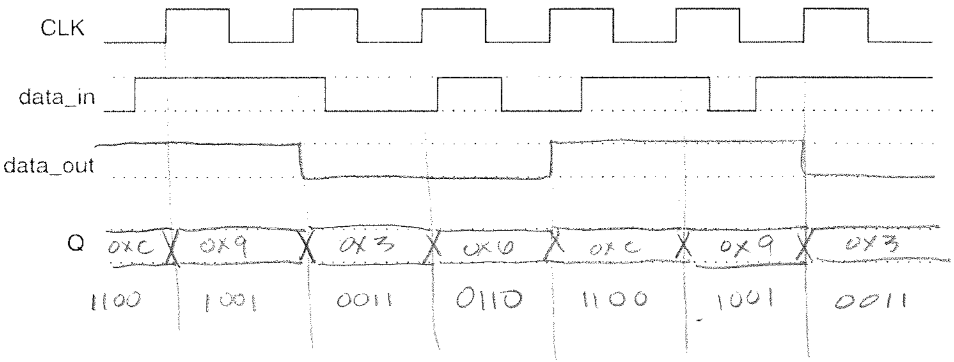
FSM: 11011

Chapter Exercises

CHAPTER ~~20~~ 27

1) Use the block diagram on the right to complete the timing diagram below. Consider the circuit to be a 4-bit shift register (shifts from right-to-left) that is active on the rising-edge triggered of the clock signal. Consider the line labeled "Q" to represent the 4-bit value stored by the shift register and the "data_out" output to represent the value of the highest order bit stored by the shift register. Assume the initial value stored by the shift register is 0xC. Ignore all propagation delay issues with this circuit





2) The block diagram on the right shows a model of a universal shift register; use this model to complete the timing diagram listed below. Consider the following:

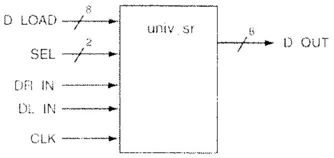
SEL = "00": hold

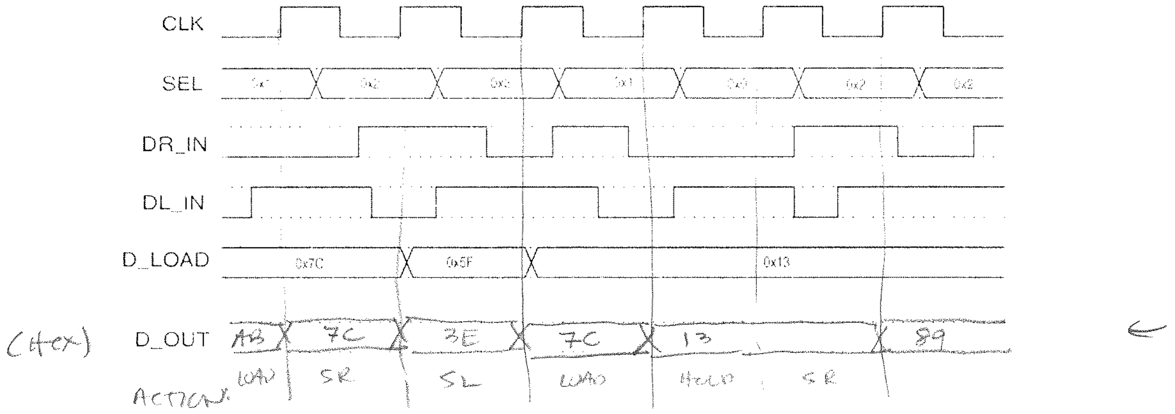
SEL = "01": parallel load of D_LOAD data

SEL = "10": right shift; DL_IN input on left

SEL = "11": left shift: DR_IN input on right

- The rising edge of the CLK signal synchronizes all shift register operations
- Propagation delays are negligent.
- Initial D_OUT value is 0xAB





0011 | 1100

1000 | 1001 | 0011

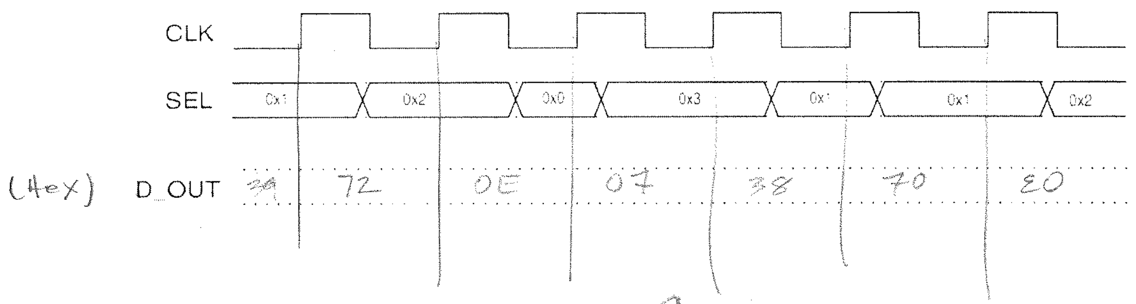
3) Complete the following timing diagram. The SEL inputs are the control inputs to an 8-bit universal shift register. Assume that all operations are synchronized with the rising edge of the clock signal. Assume that propagation delays are negligent. Be sure to state any other assumptions you need to make in order to complete this problem. Assume the 0x39 is the initial value stored by the shift register. Assume "D_OUT" is an 8-bit output representing the value stored by the shift register.

SEL = "00": rotate right

SEL = "01": rotate left

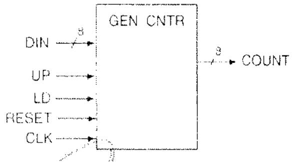
SEL = "10": divide by 8 (bit stuff 0's)

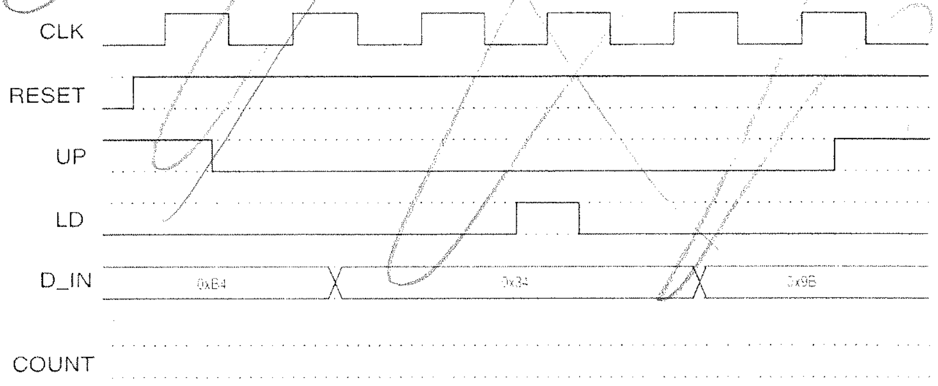
SEL = "11": multiply by 8 (bit stuff 0's)



4) The block diagram on the right shows a model of an 8-bit counter. Use the following assumptions in order to complete the following timing diagram. Assume propagation delays are negligent.

- The LD input enables the DIN loading into the counter
- The RESET input is an asynchronous and active low used to reset the counter
- The COUNT output shows the current value stored by the counter
- The counter counts up when the UP input is asserted (active high) or down otherwise. All count operations are synchronous.





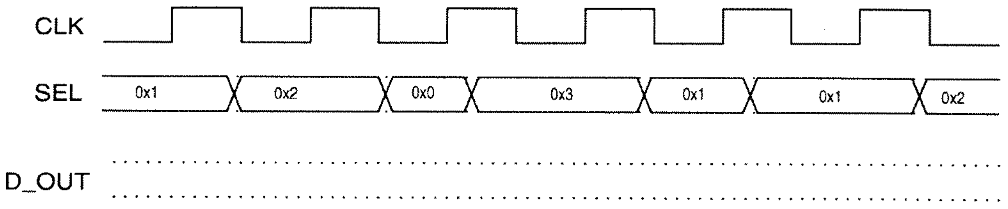
3) Complete the following timing diagram. The SEL inputs are the control inputs to an 8-bit universal shift register. Assume that all operations are synchronized with the rising edge of the clock signal. Assume that propagation delays are negligent. Be sure to state any other assumptions you need to make in order to complete this problem. Assume the 0x39 is the initial value stored by the shift register. Assume "D_OUT" is an 8-bit output representing the value stored by the shift register.

SEL = "00": rotate right

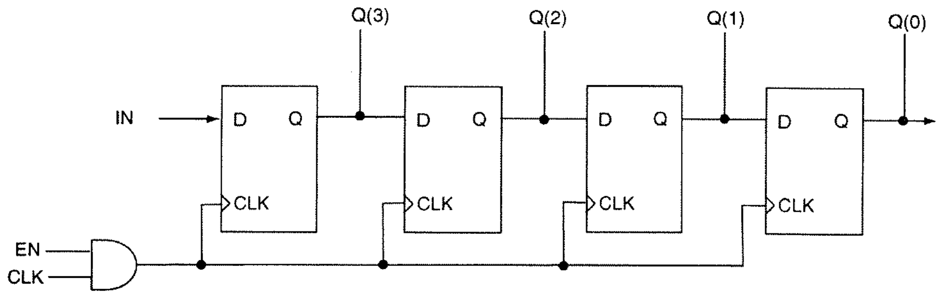
SEL = "01": rotate left

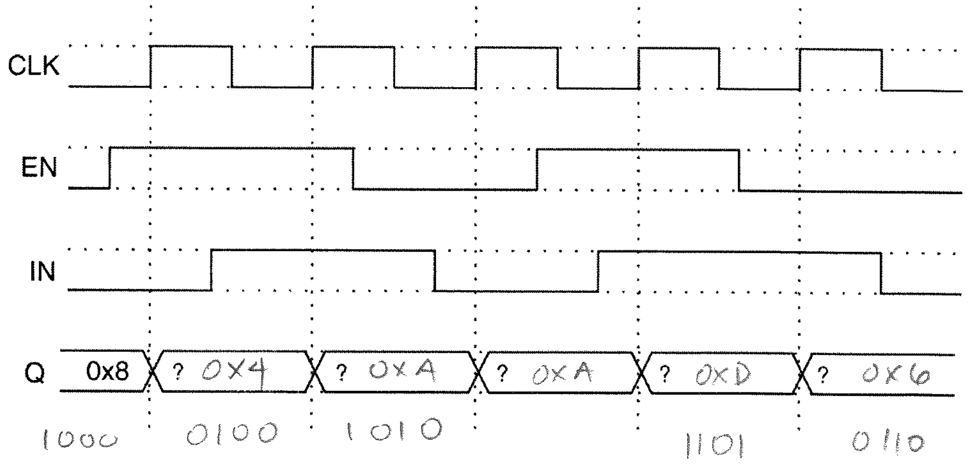
SEL = "10": divide by 8 (bit stuff 0's)

SEL = "11": multiply by 8 (bit stuff 0's)

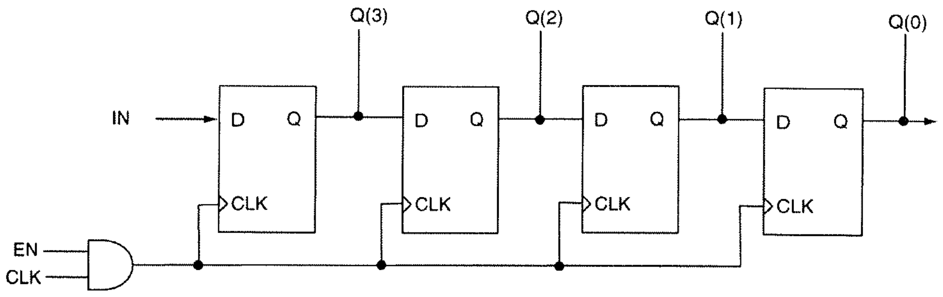


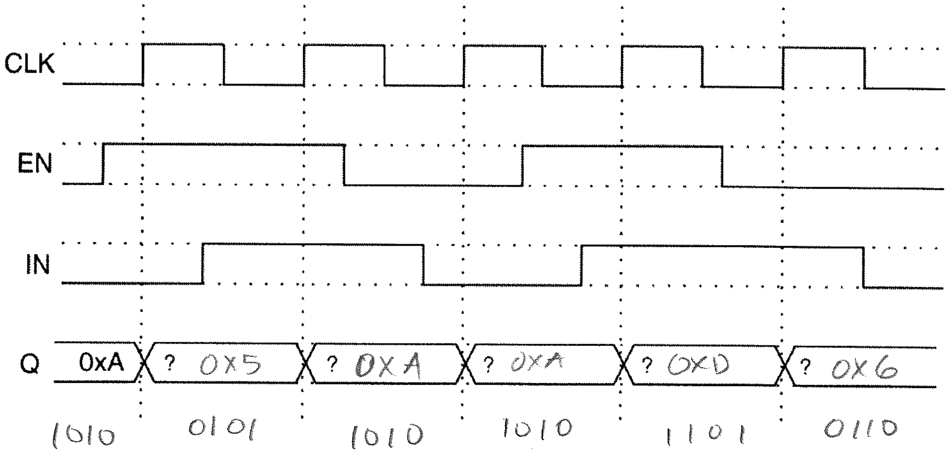
4) Use the schematic diagram to complete the **Q** output. The **Q** output is a 4-bit bundle; the starting state of Q is listed in the timing diagram as a hex value (4-bits).. Assume that propagation delays are negligent.





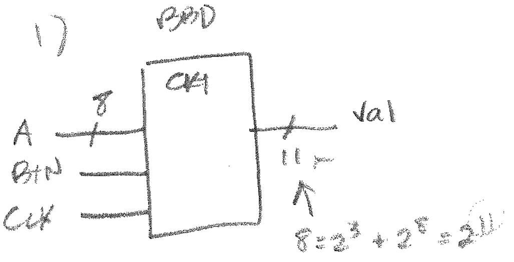
5) Use the schematic diagram to complete the Q output. The Q output is a 4-bit bundle; the starting state of Q is listed in the timing diagram as a hex value (4-bits).

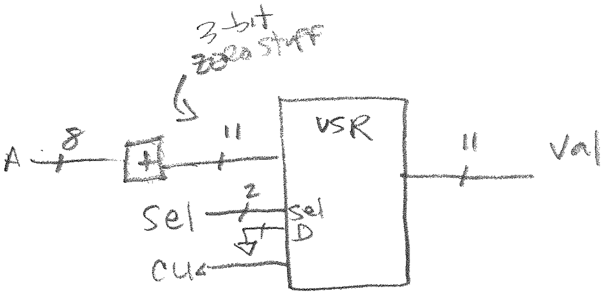




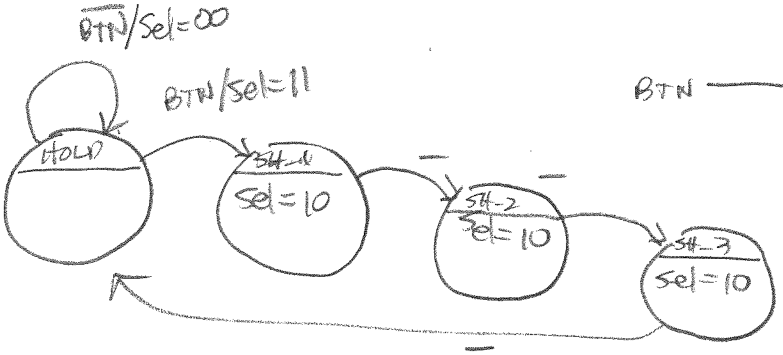
DESIGN PROBLEM SOLUTIONS CHAP 28

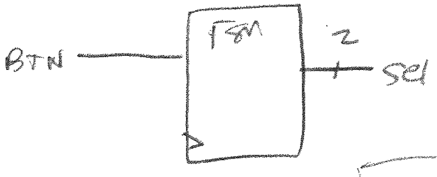
1)





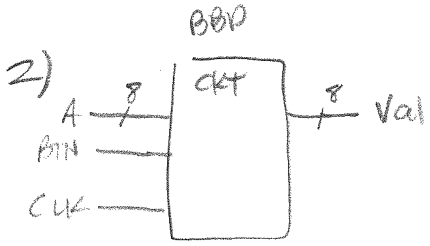
Sel	ACTION
00	HOLD
01	SH RIG
10	SH LEFT
11	LOAD

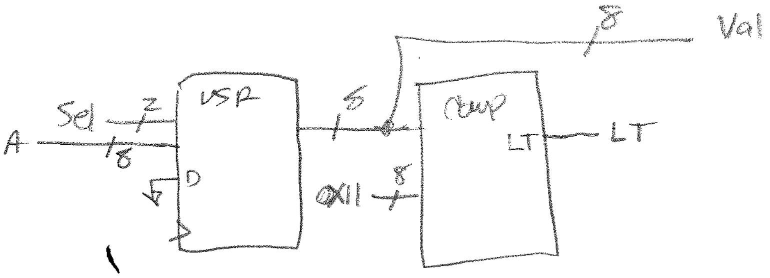


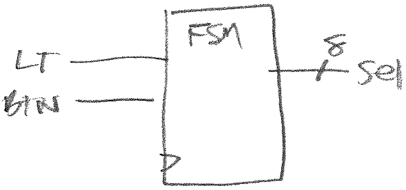


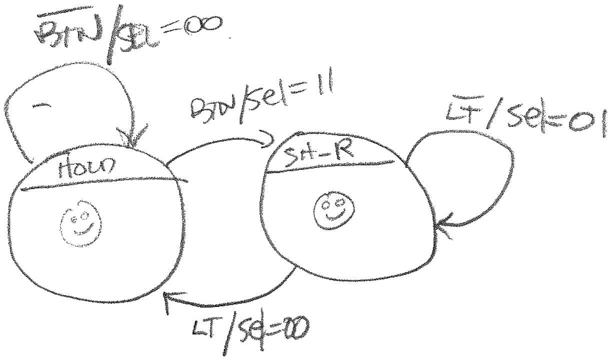
circuit controlled

2)





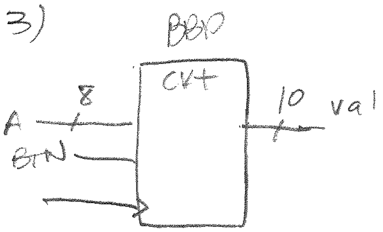


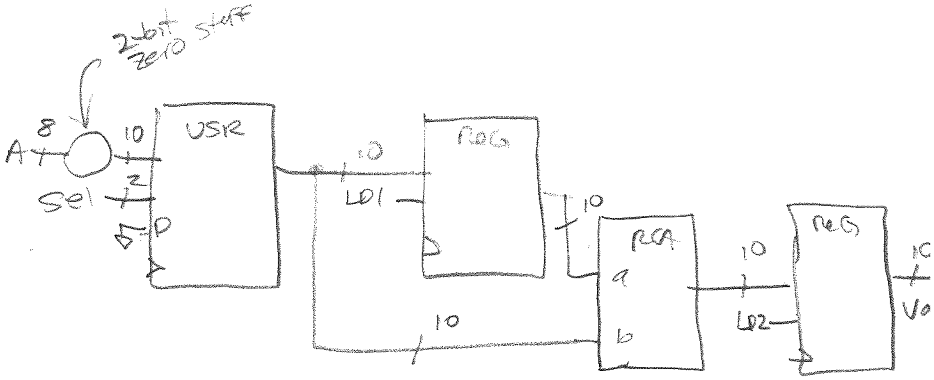


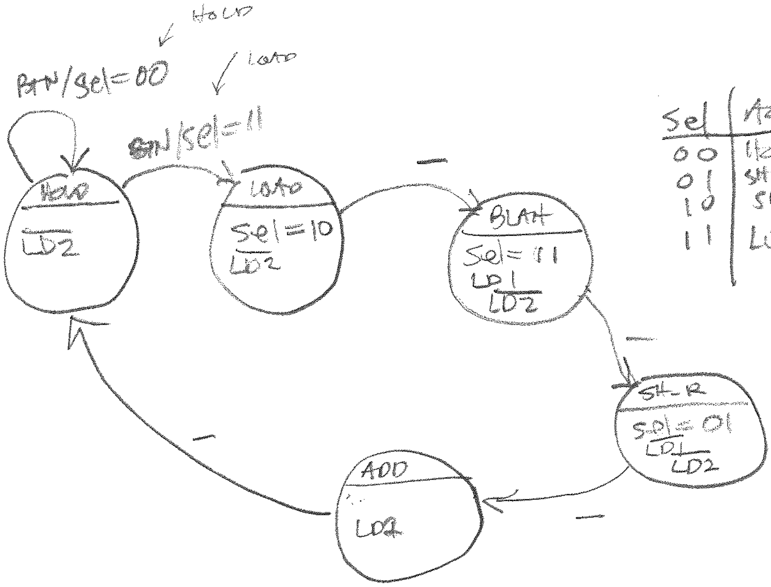
Sel	ACTION
00	HOLD
01	SH RIG
10	SH LEF
11	LOAD

circuit controlled

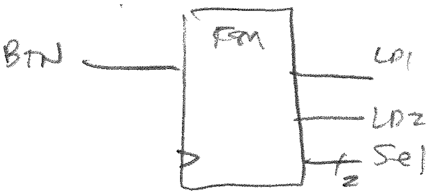
3)







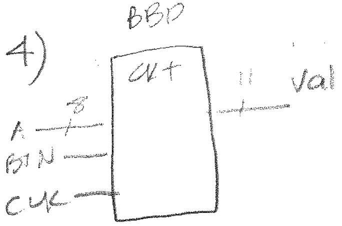
Sel	ACTION
00	HOLD
01	SH RIG
10	SH LEF
11	LOAD

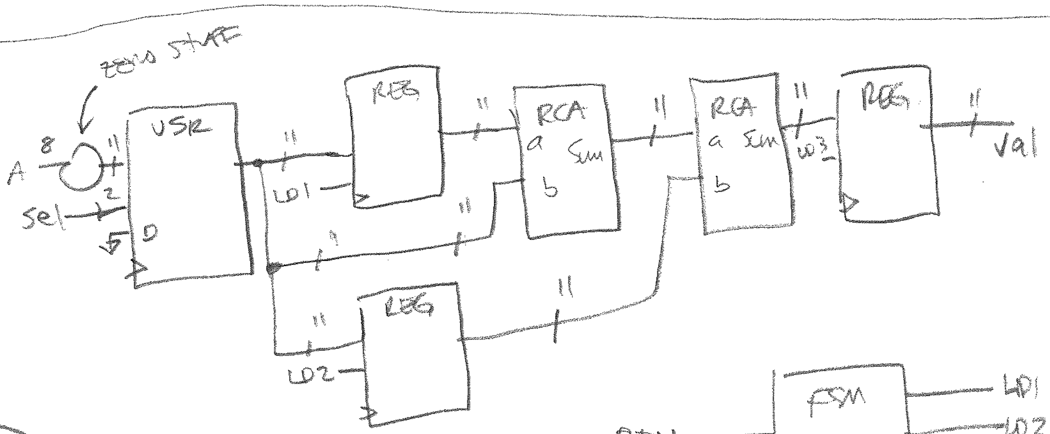


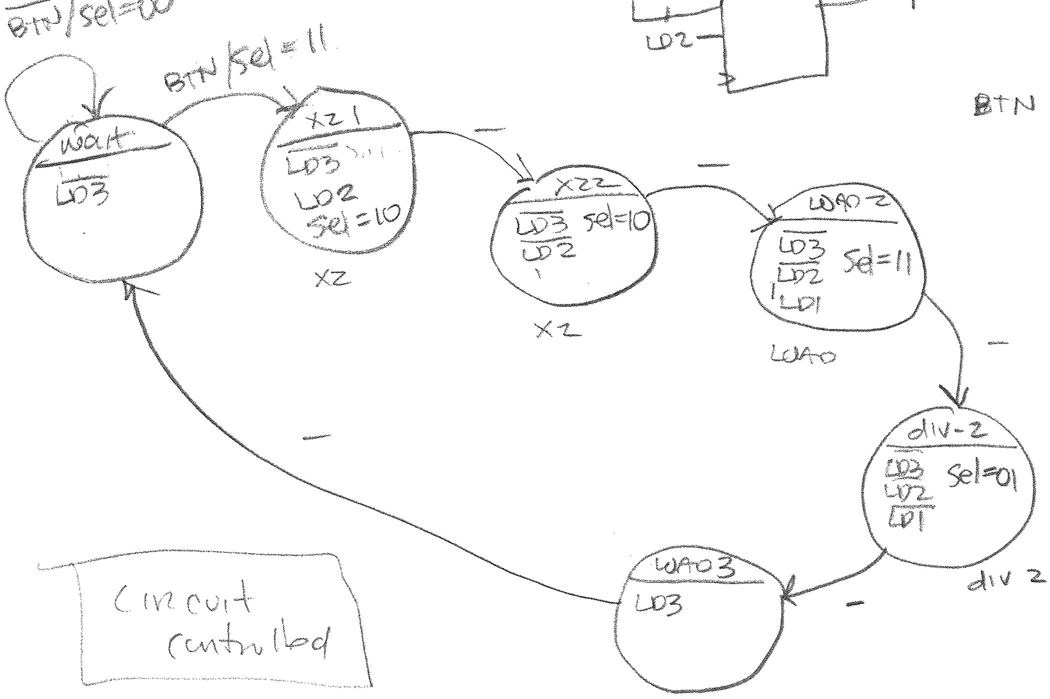
circuit controlled

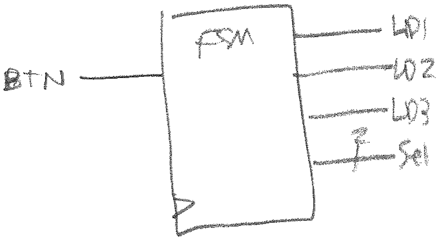
ASSUME A val DOES NOT CHANGE QUICKLY.

4)





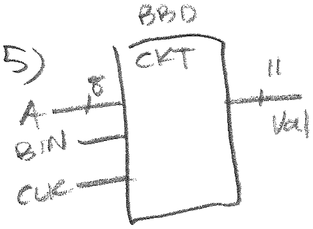


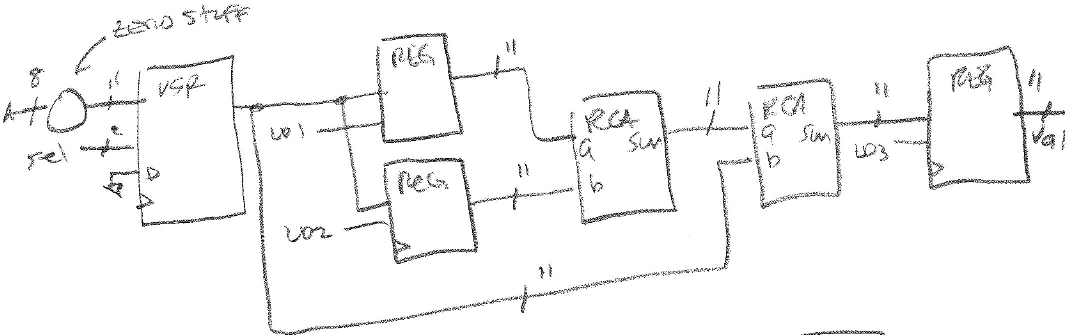


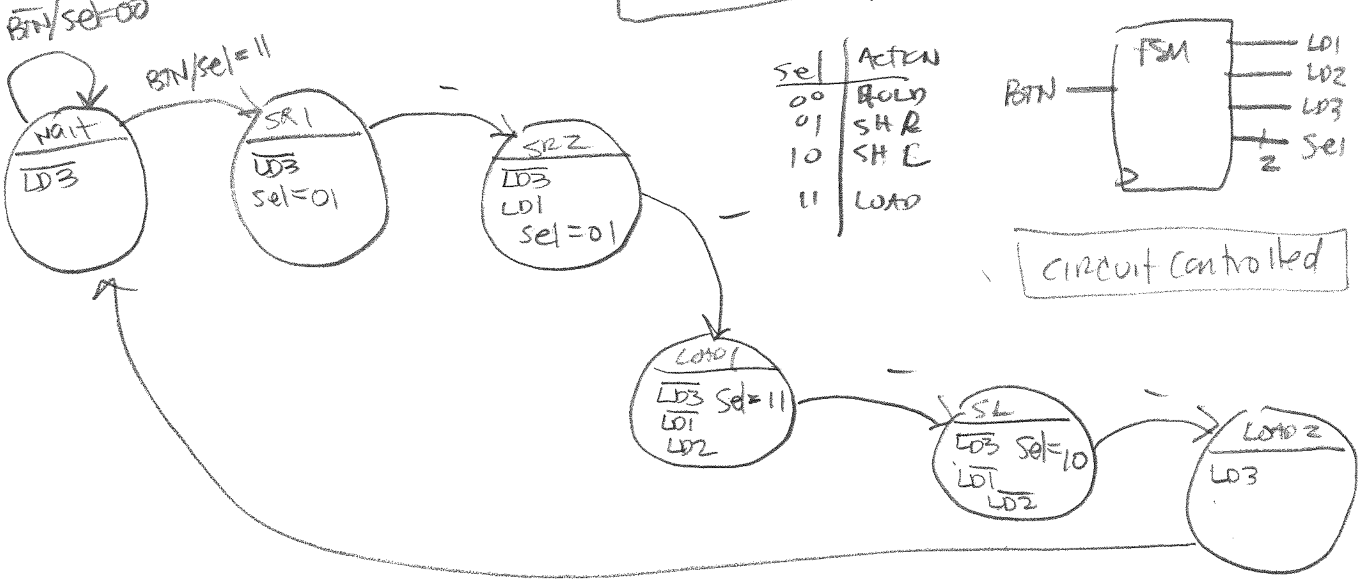
Sel	ACTION
00	HOLD
01	SH RIG
10	SH LEF
11	LOAD

circuit controlled

5)



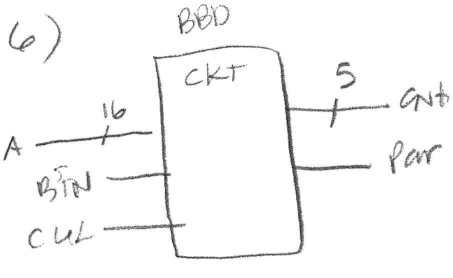


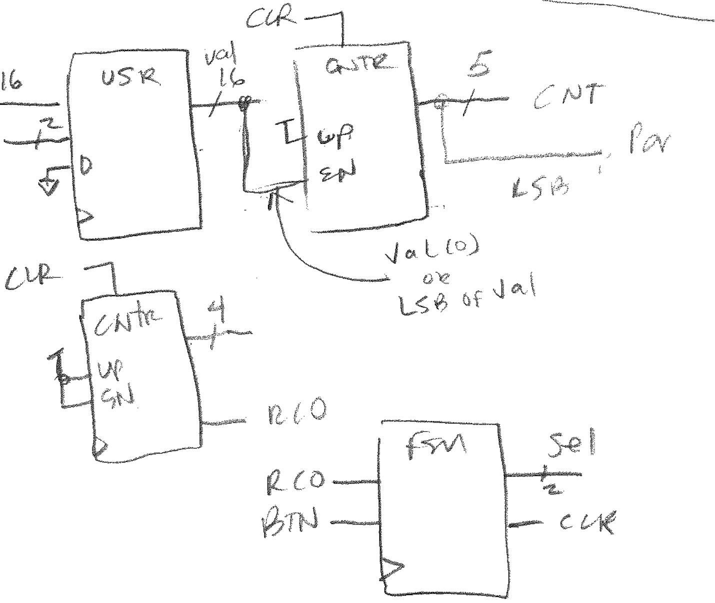


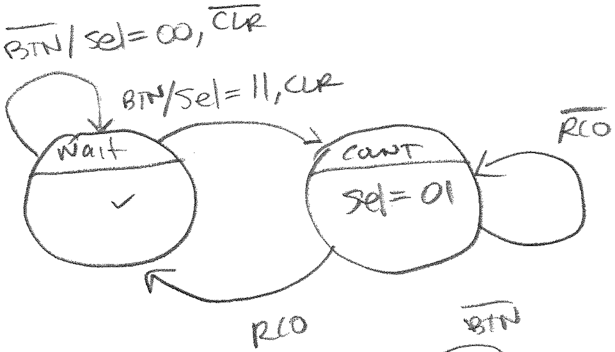
Sel	Action
00	HOLD
01	SH R
10	SH L
11	LOAD

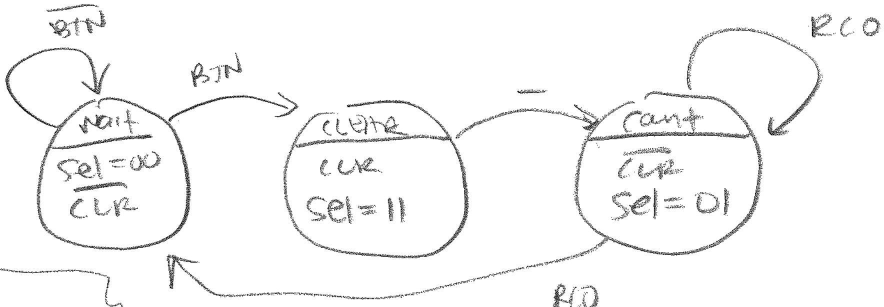
circuit controlled

6)





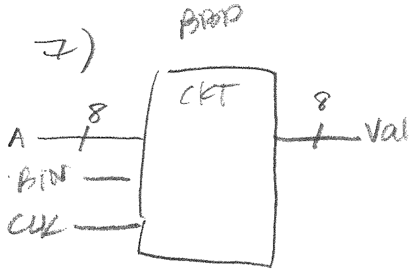


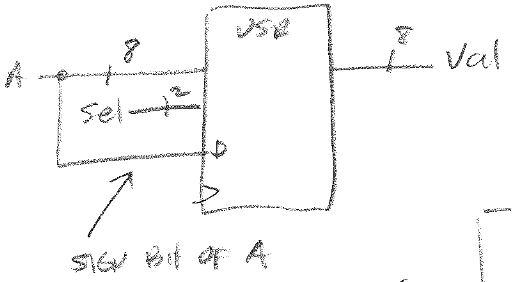


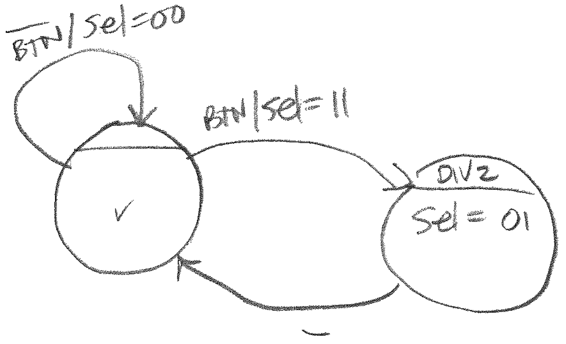
Sel	Action
00	HOLD
01	SH R
10	SH L
11	LOAD

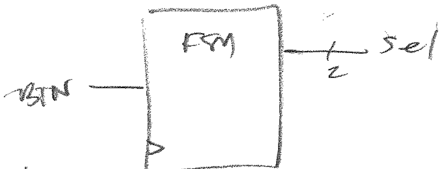
circuit controlled

7)





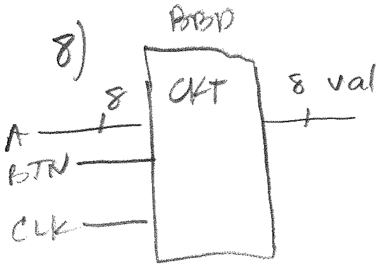


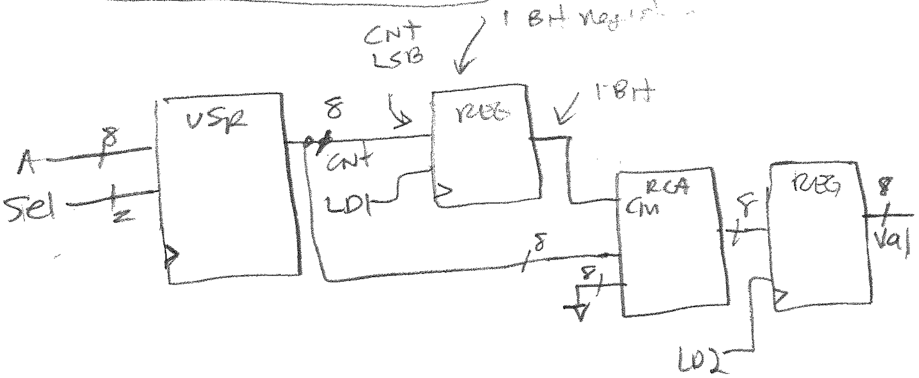


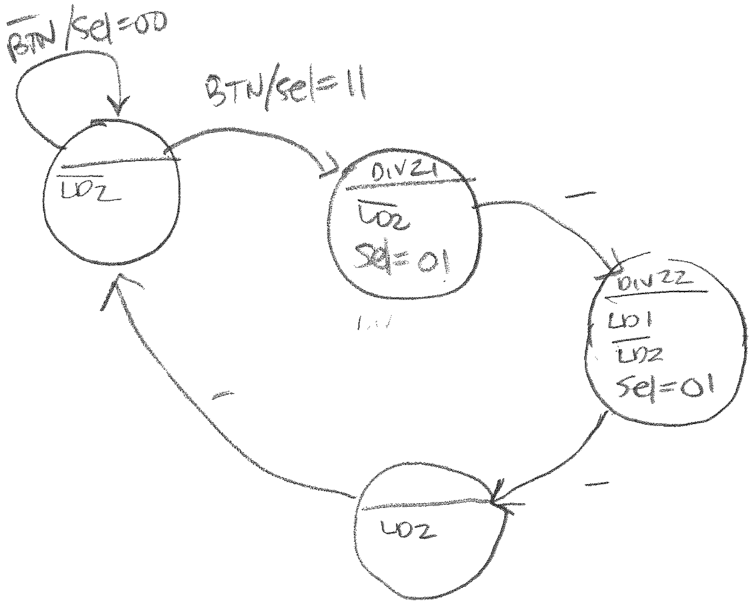
Sel	ACTION
00	HOLD
01	SH RIG
10	SH LEF
11	LOAD

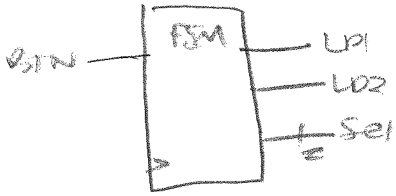
circuit controlled

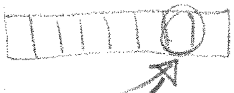
8)









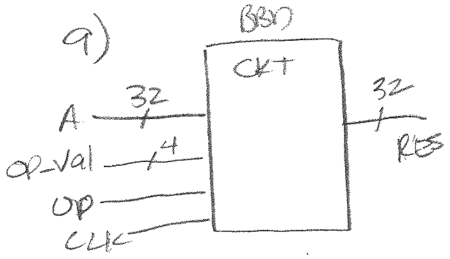


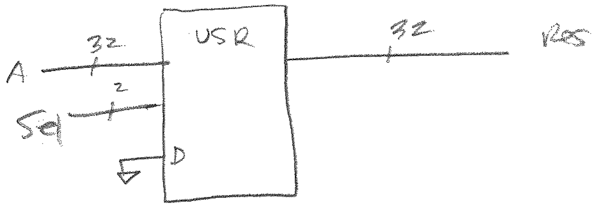
So... If THIS BIT IS '1', THE CIRCUIT ROUNDS UP BY ADDING '1' to THE SHIFTED RESULT

Sel	Action
00	HOLD
01	SHR
10	SH L
11	LOAD

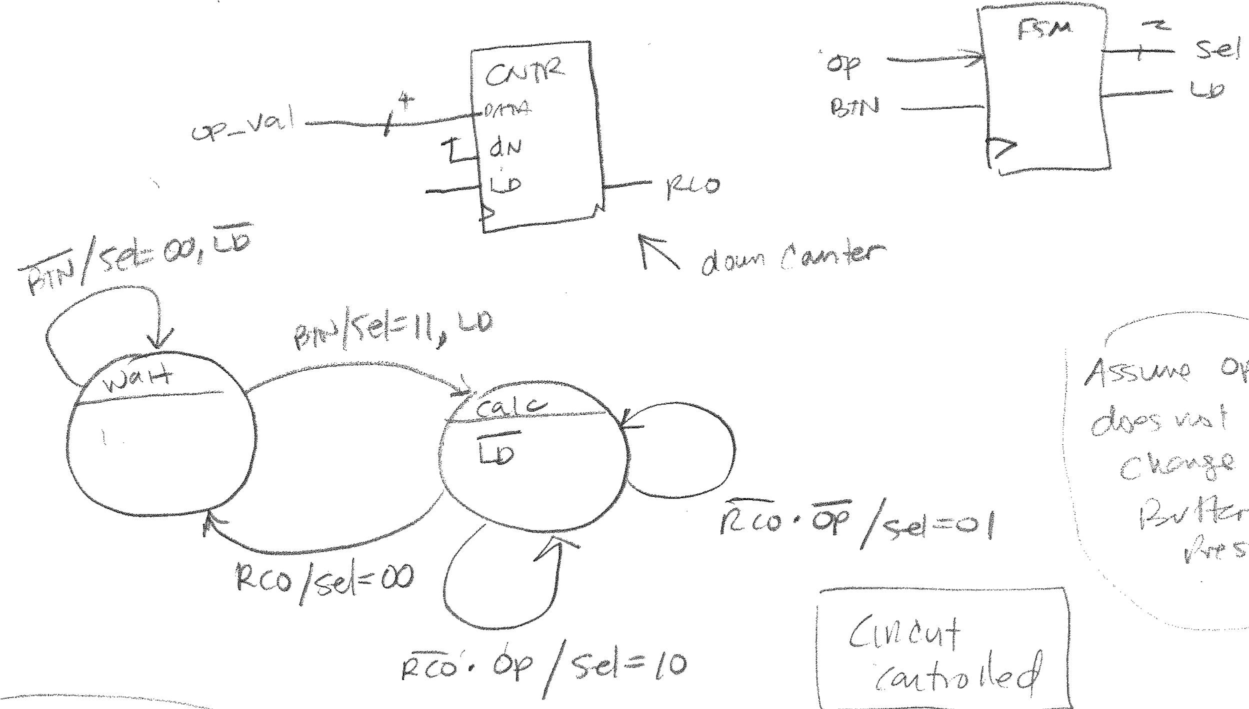
circuit controlled

9)





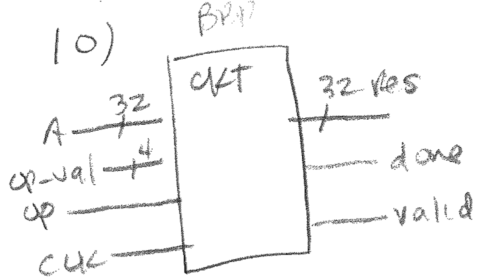
Sel	Action
00	HOLD
01	SH R
10	SH L
11	LOAD

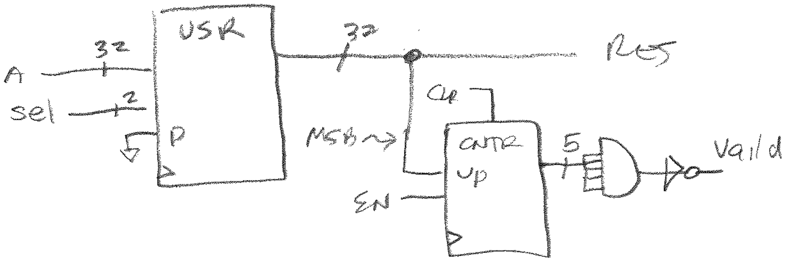


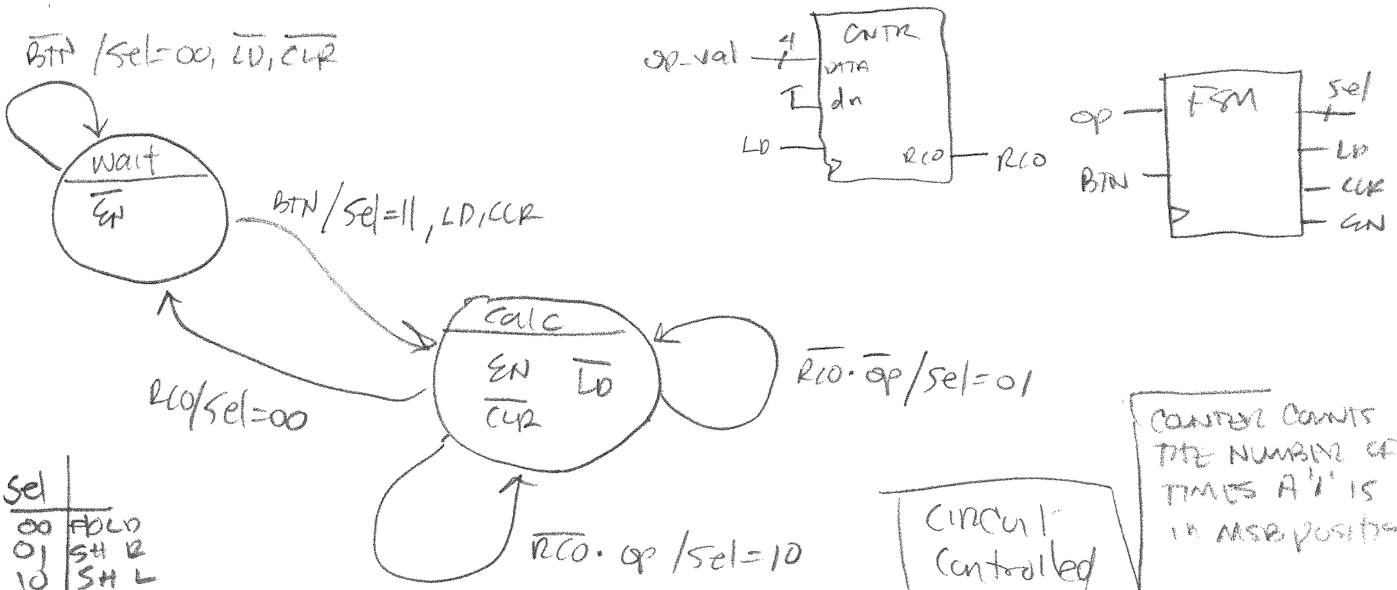
Assume op does not change AFTER Button Press

Circuit controlled

10)





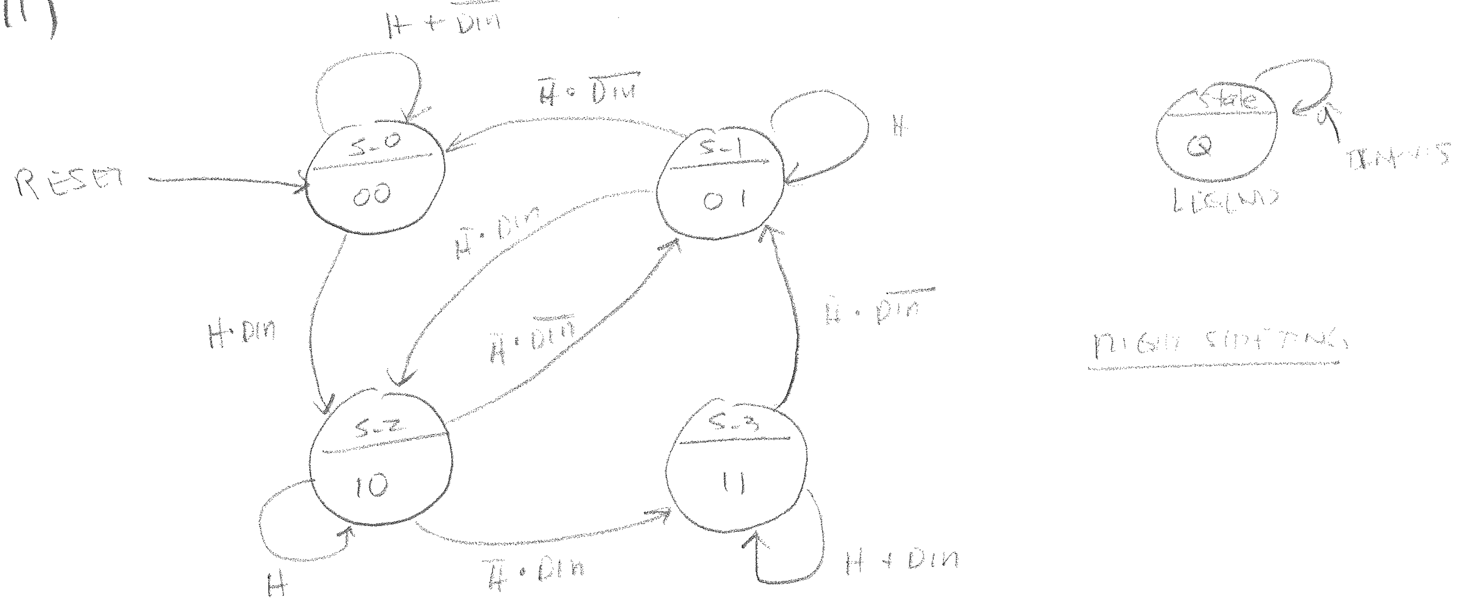


Sel	
00	HOLD
01	SH R
10	SH L
11	LOAD

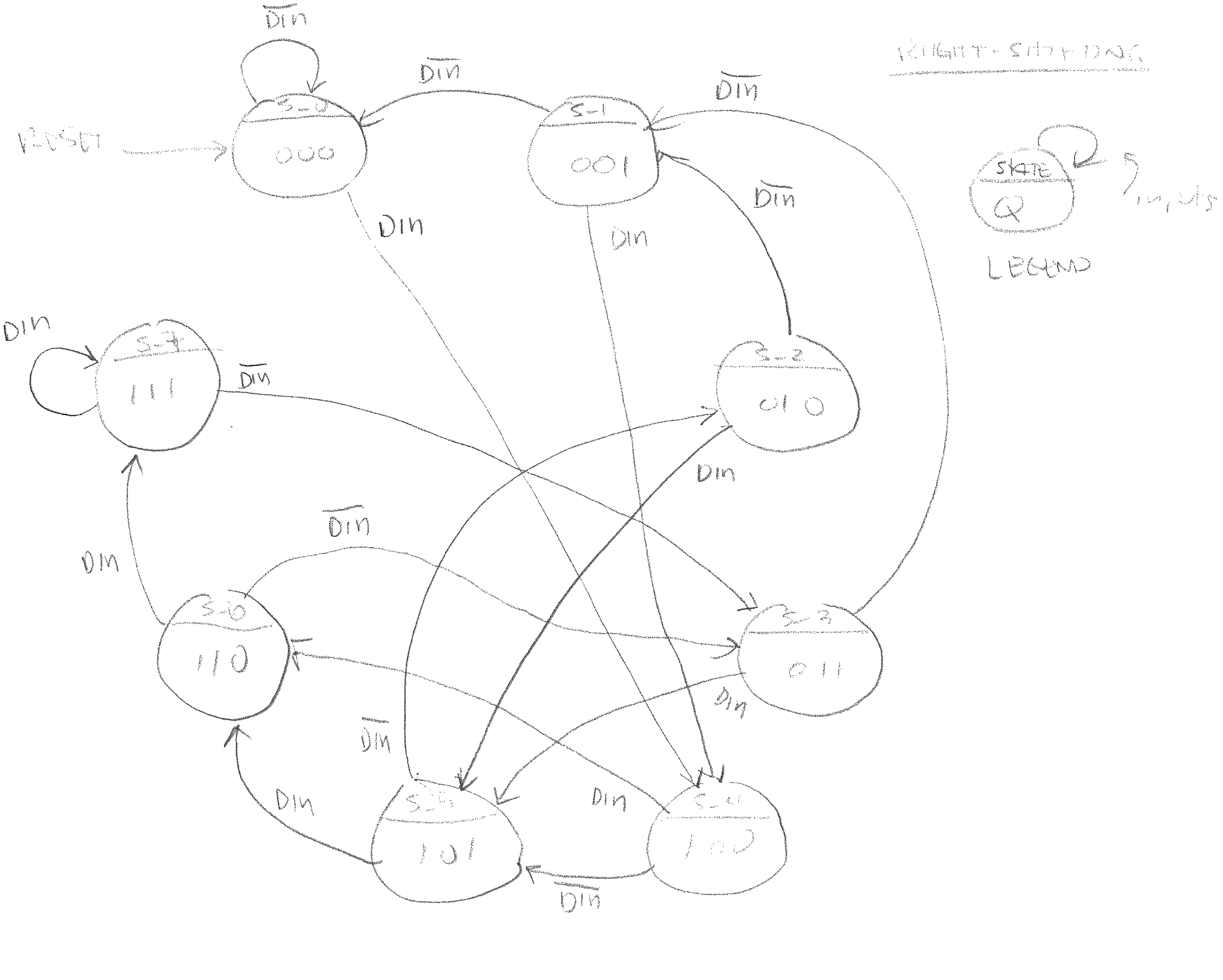
Circuit Controlled

COUNTER COUNTS THE NUMBER OF TIMES A '1' IS IN MSB POSITION.

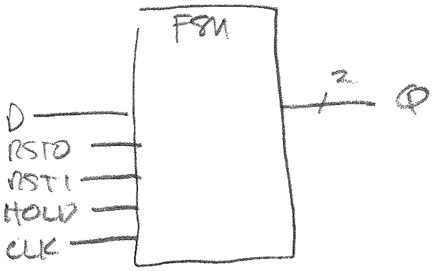
11)

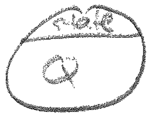

$H + \overline{DIN}$
$\overline{H} \cdot \overline{DIN}$
H
RESET
S_0
00
S_1
01
$\overline{H} \cdot DIN$
H·DIN
$\overline{H} \cdot \overline{DIN}$
$\overline{H} \cdot \overline{DIN}$
S_2
10
S_3
11
H
$\overline{H} \cdot DIN$
H + DIN
State
Q
INPUTS
LEGEND
RIGHT SHIFTING

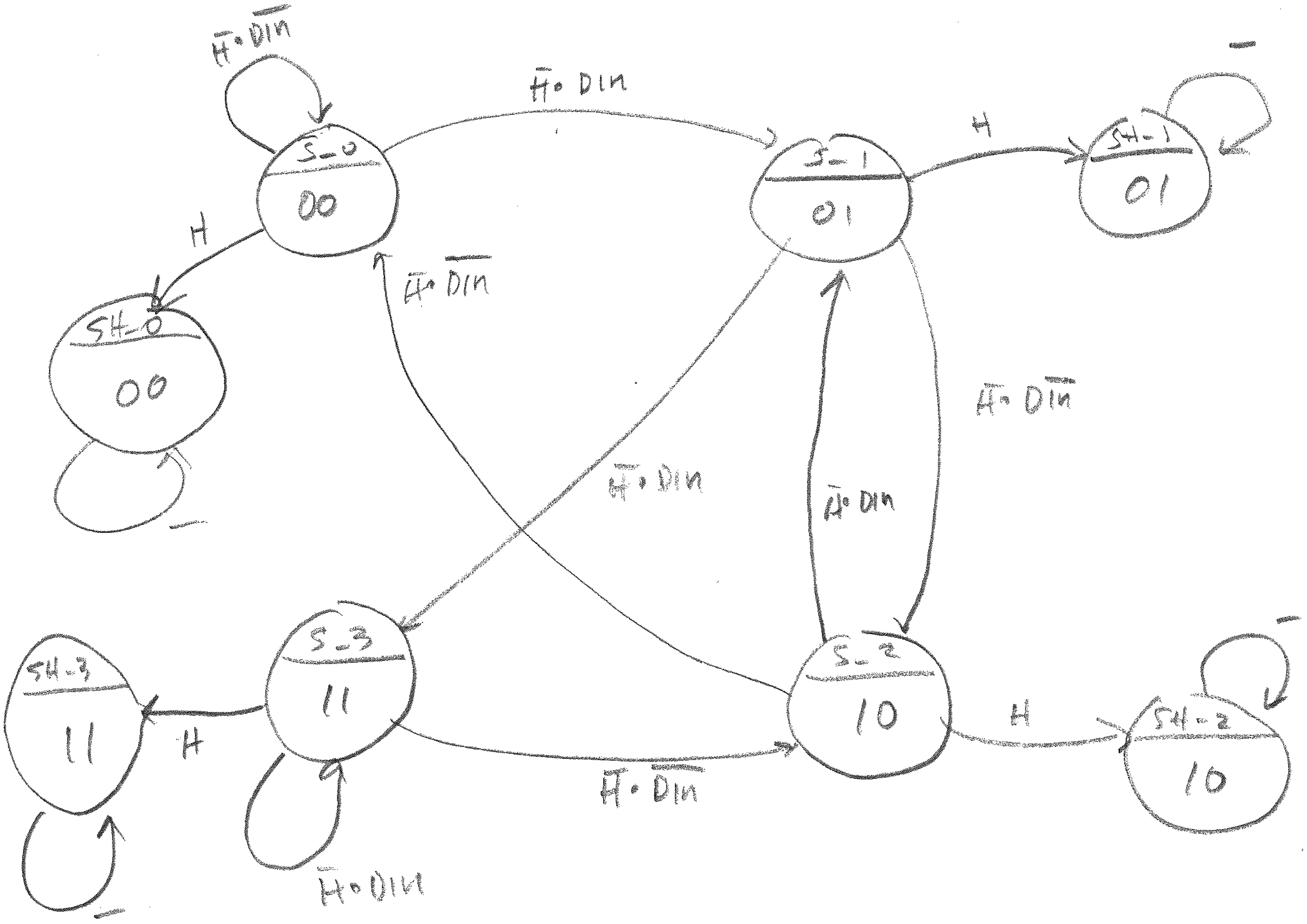
12)


$\overline{DIN}$
$\overline{DIN}$
$\overline{DIN}$
RIGHT-SHIFTING
RESET
S_0
000
S_1
001
$\overline{DIN}$
DIN
DIN
STATE
Q
INPUTS
LEGEND
DIN
S_7
111
$\overline{DIN}$
S_2
010
DIN
$\overline{DIN}$
DIN
S_6
110
S_3
011
DIN
$\overline{DIN}$
DIN
DIN
S_5
101
S_4
100
$\overline{DIN}$

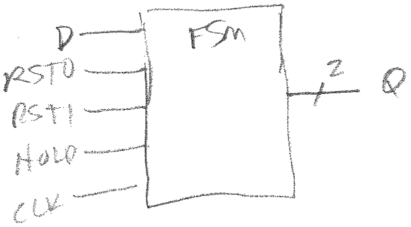
13)


FSM
D
RST0
RST1
HOLD
CLK
2
Q

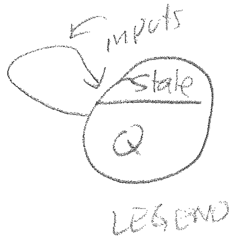

state
Q

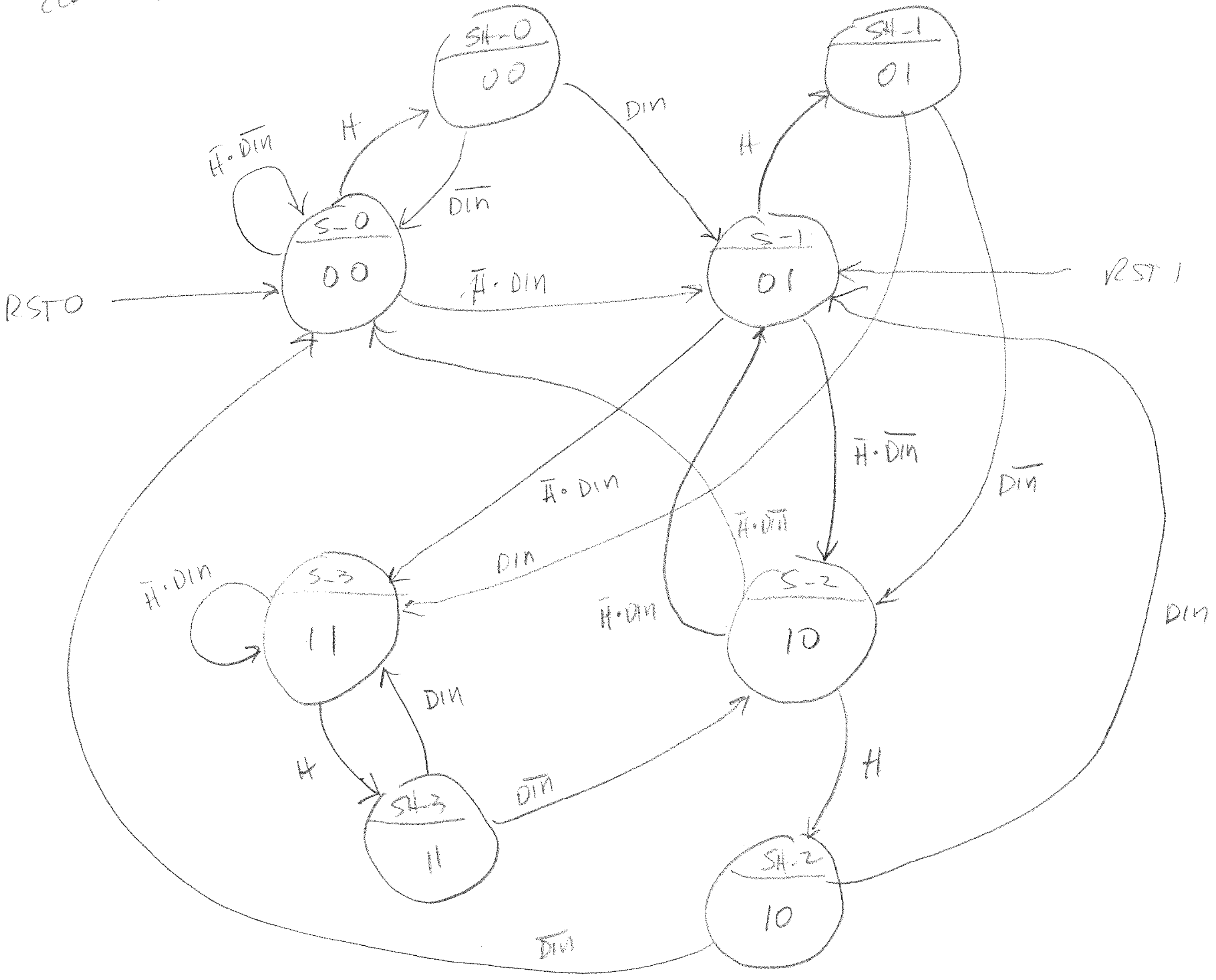

$\overline{H}\cdot\overline{DIN}$
$\overline{H}\cdot DIN$
S_0
00
H
SH_0
00
S_1
01
H
SH_1
01
$\overline{H}\cdot\overline{DIN}$
$\overline{H}\cdot DIN$
$\overline{H}\cdot\overline{DIN}$
$\overline{H}\cdot DIN$
S_3
11
SH_3
11
H
$\overline{H}\cdot DIN$
$\overline{H}\cdot\overline{DIN}$
S_2
10
H
SH_2
10

14)

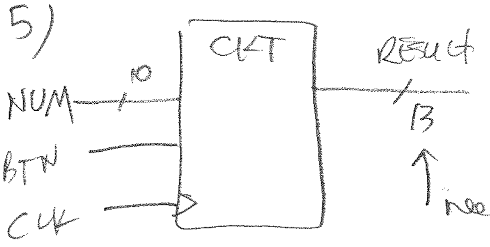


SHIFTS LEFT



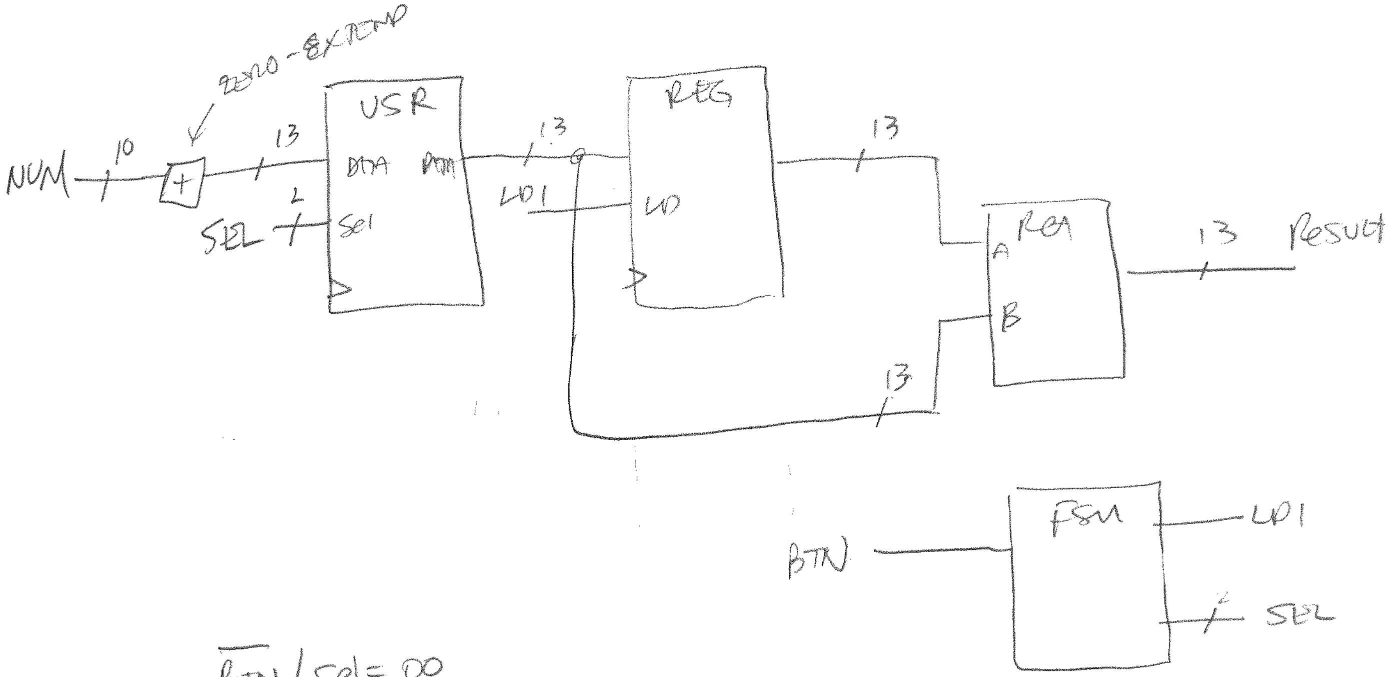


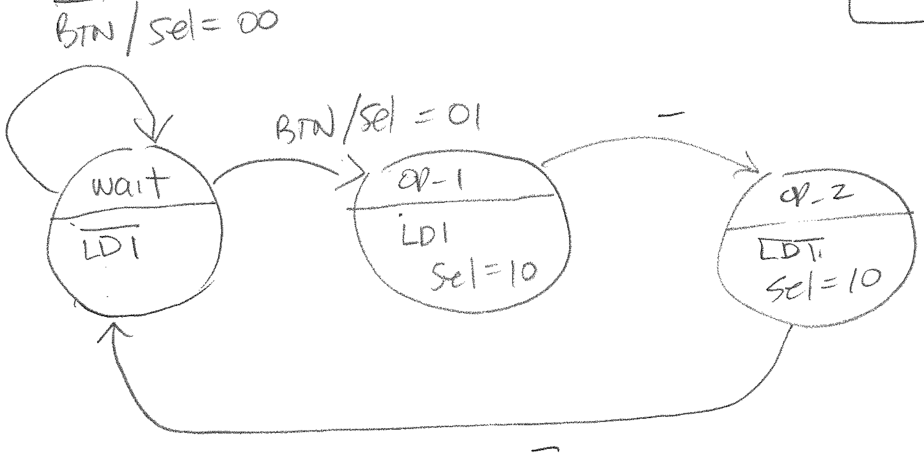
15)



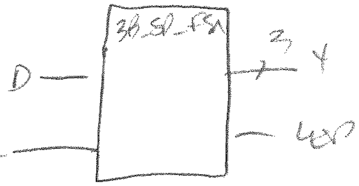
Sel	
00	HOLD
01	LOAD
10	SHL
11	SHR

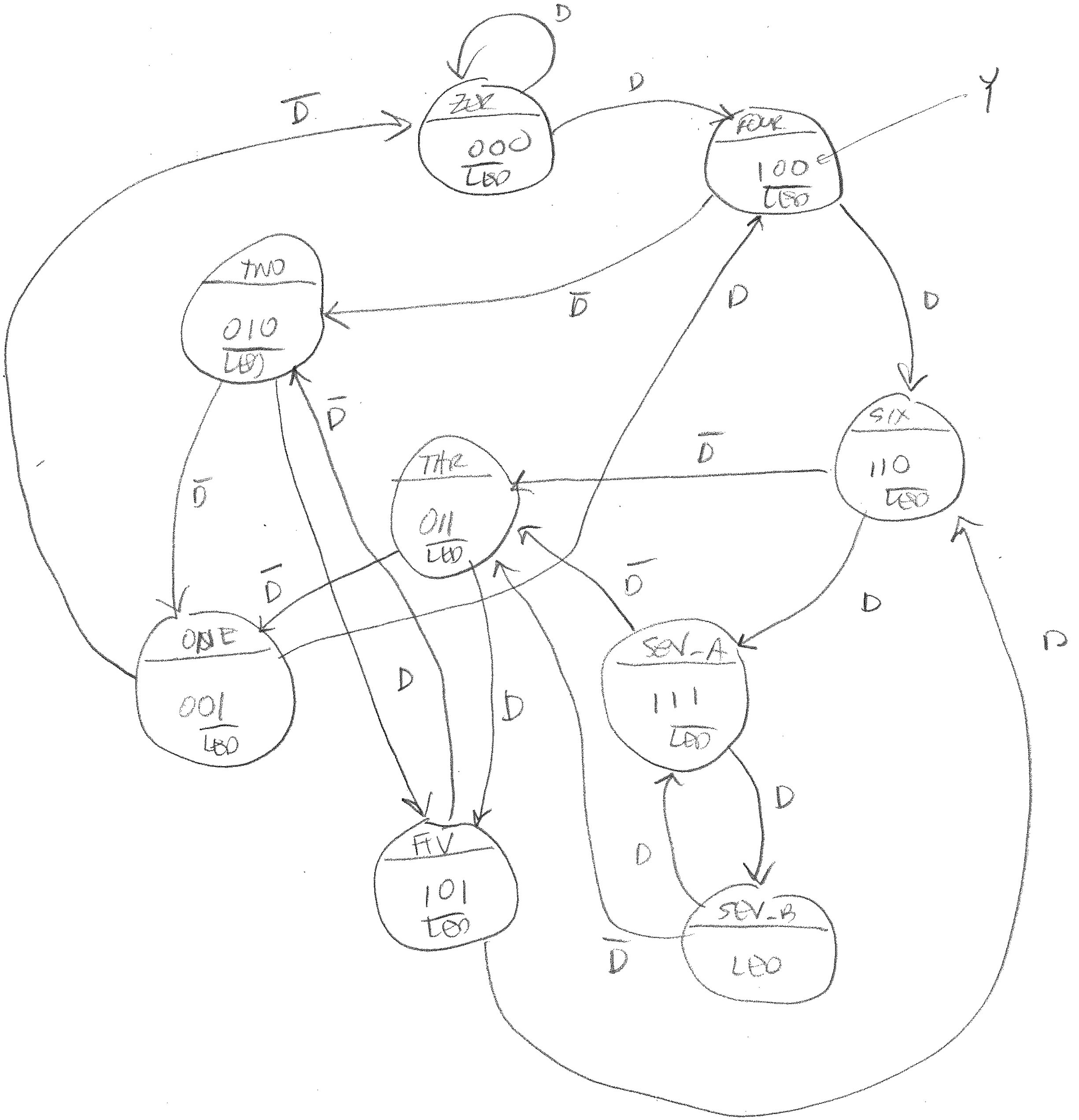
↑ needs extra 3 bits to handle 5x



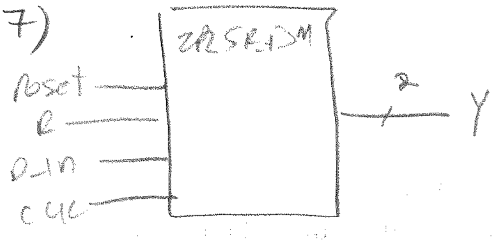


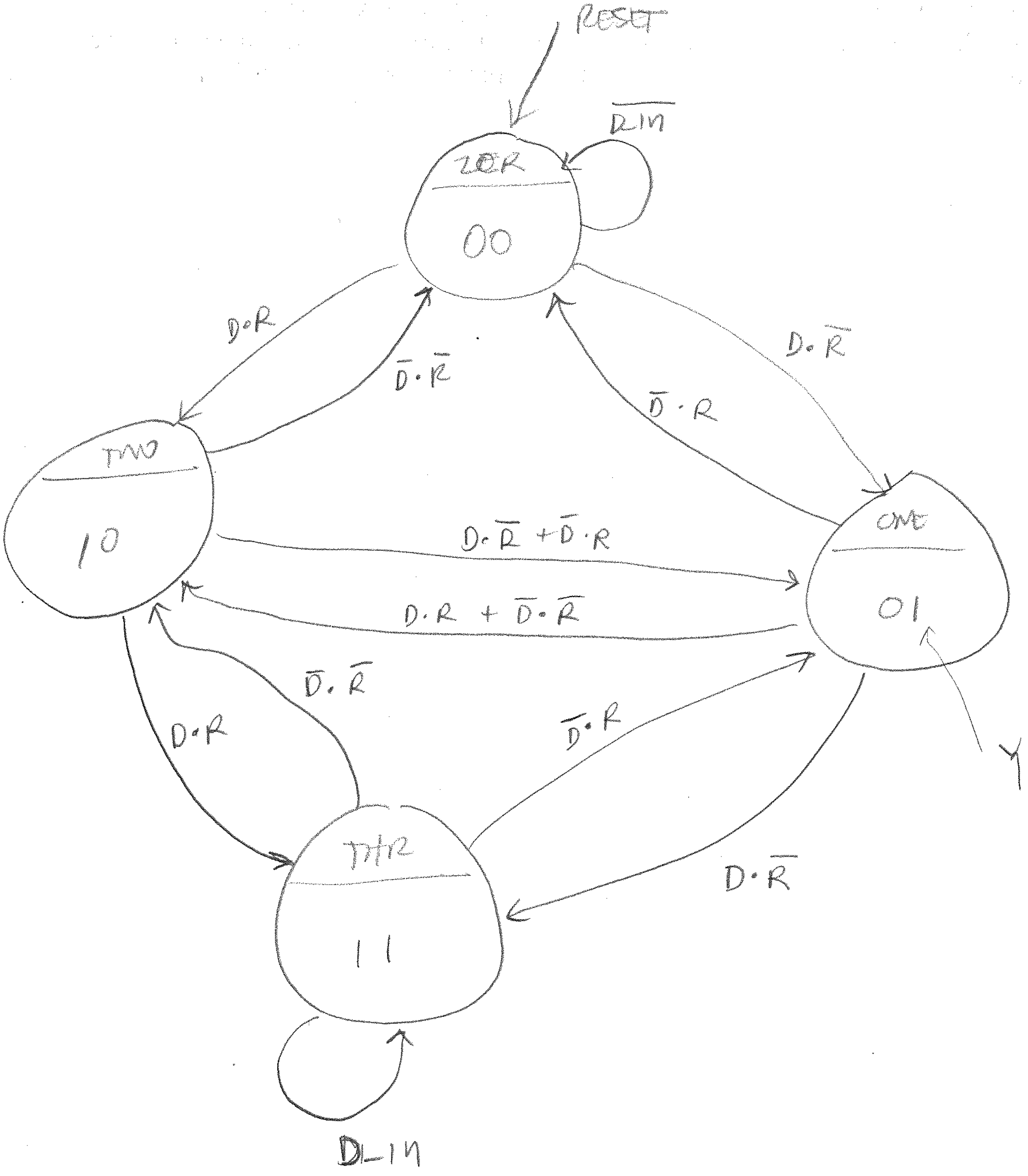
16)


3B_SD_FSM
D
CLK
3
Y
LED

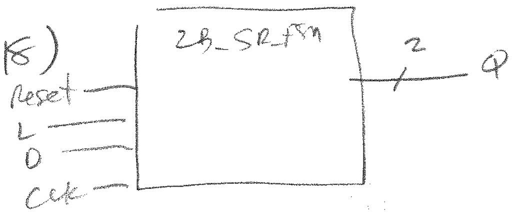

$\overline{D}$
$\overline{D}$
D
Y
ZER
000
LED
FOUR
100
LED
TWO
010
LED
$\overline{D}$
D
D
$\overline{D}$
$\overline{D}$
SIX
110
LED
$\overline{D}$
THR
011
LED
$\overline{D}$
$\overline{D}$
D
D
ONE
001
LED
D
D
SEV_A
111
LED
D
D
FIV
101
LED
SEV_B
LED
$\overline{D}$

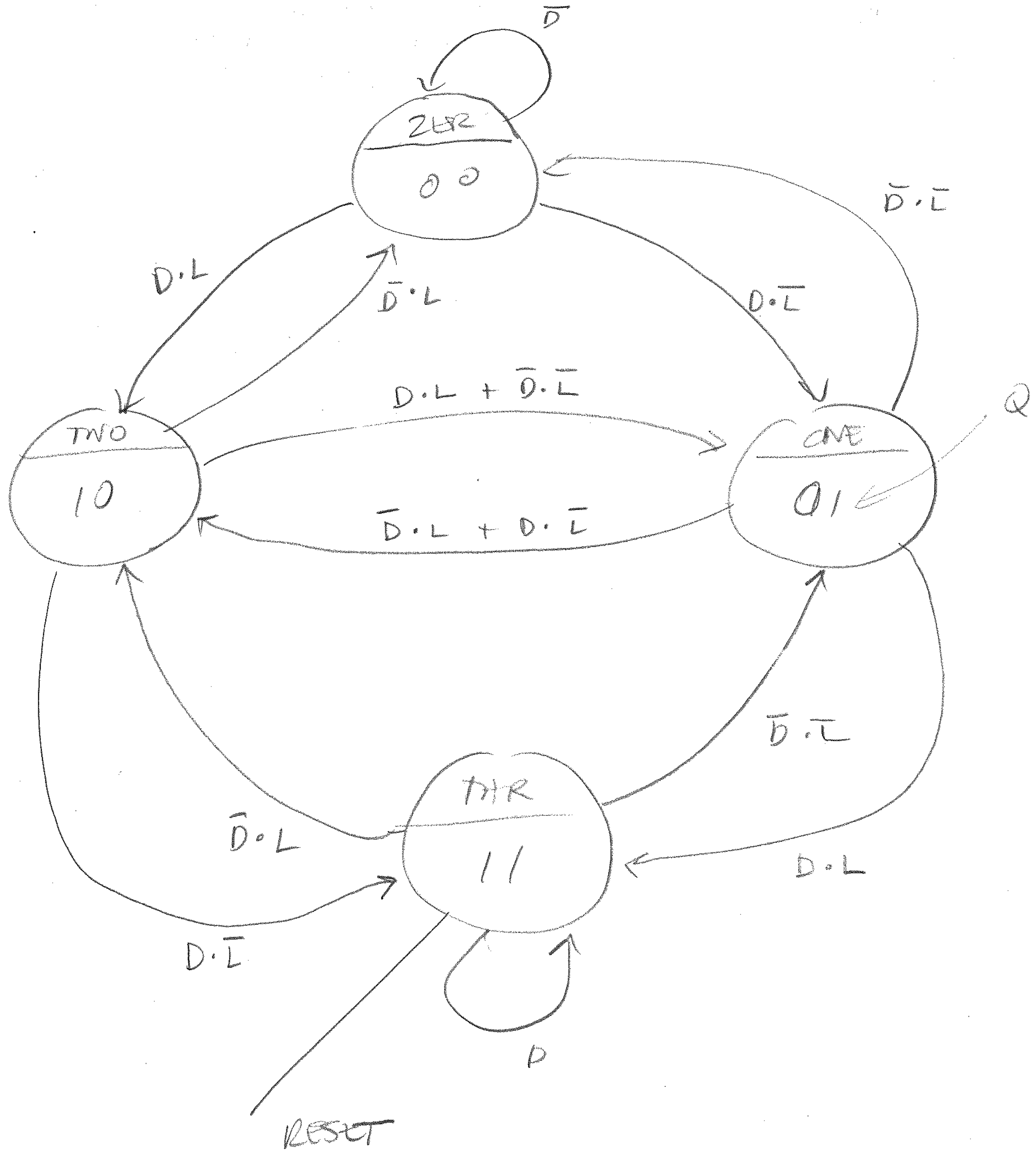
17)


ZRSR_DM
Reset
R
D_in
CLK
2
Y

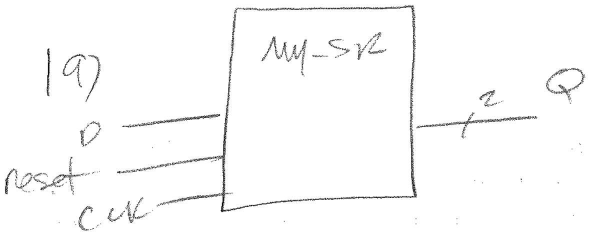

RESET
$\overline{D\_in}$
ZER
00
$D \cdot R$
$\overline{D} \cdot \overline{R}$
$D \cdot \overline{R}$
$\overline{D} \cdot R$
TWO
10
$D \cdot \overline{R} + \overline{D} \cdot R$
$D \cdot R + \overline{D} \cdot \overline{R}$
ONE
01
Y
$\overline{D} \cdot \overline{R}$
$D \cdot R$
$\overline{D} \cdot R$
THR
11
$D \cdot \overline{R}$
D_in

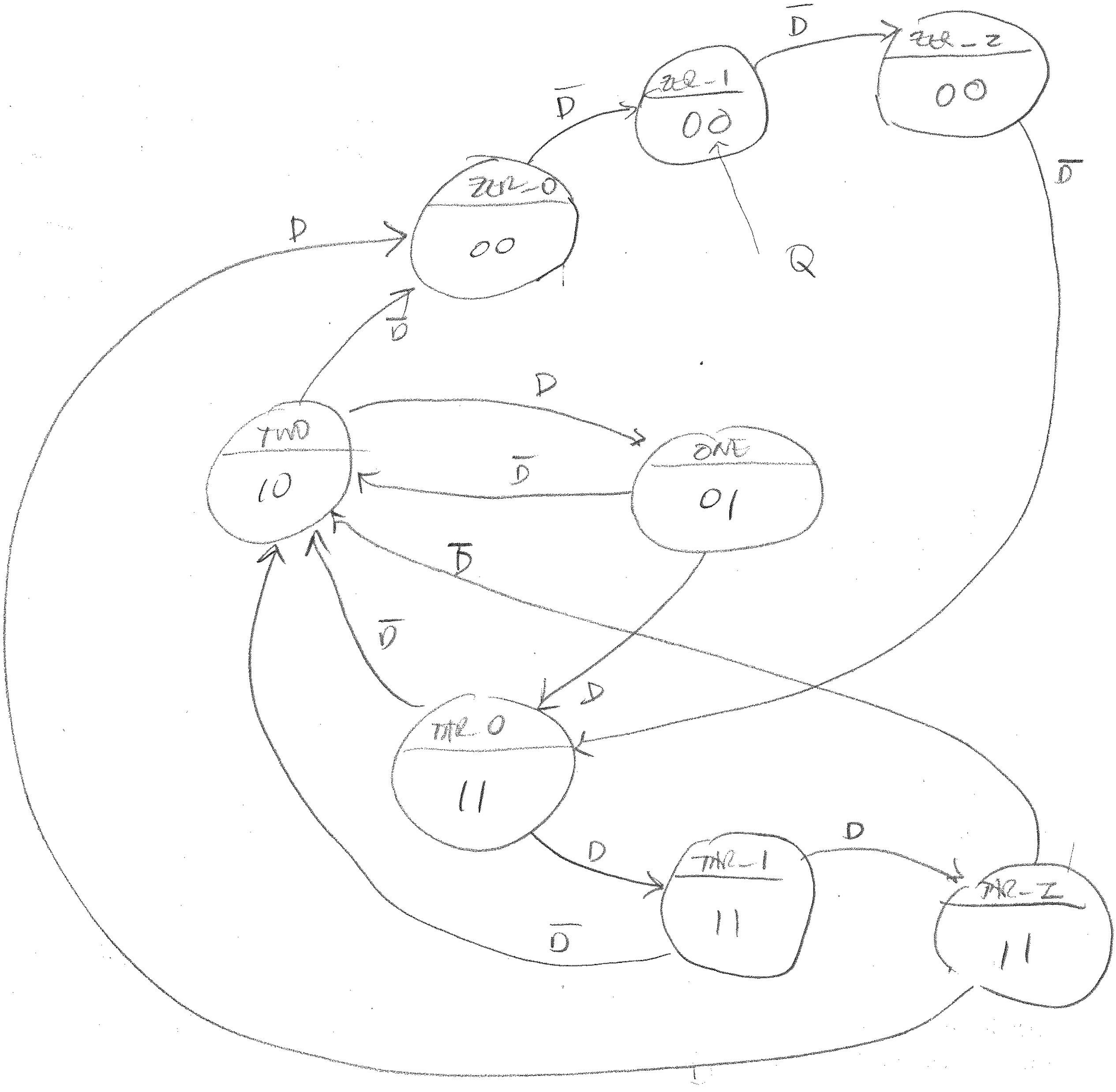
18)


2B_SR_fsm
Reset
L
D
Clk
2
Q


ZER
00
$\overline{D}$
$\overline{D}\cdot\overline{L}$
$D\cdot L$
$\overline{D}\cdot L$
$D\cdot\overline{L}$
TWO
10
$D\cdot L + \overline{D}\cdot\overline{L}$
$\overline{D}\cdot L + D\cdot\overline{L}$
ONE
01
Q
$\overline{D}\cdot\overline{L}$
THR
11
$\overline{D}\cdot L$
$D\cdot L$
$D\cdot\overline{L}$
D
RESET

19)


MY_SR
D
reset
CLK
2
Q


ZER_0
00
ZER_1
00
ZER_2
00
TWO
10
ONE
01
TAR_0
11
TAR_1
11
TAR_2
11
Q
D
$\overline{D}$