1 Digital Systems small exercise 1

Submission date: 16/3/2006.

All questions have an equal weight.

1.1 Question 1

Convert

1. $(1729)_{10} \rightarrow ()_8$
2. $(8EA2)_{16} \rightarrow ()_{10}$
3. $(12.31)_{10} \rightarrow ()_2$
4. $(B.AB)_{13} \rightarrow ()_{10}$

1.2 Question 2

Calculate the 2 complements of:

1. $(28A)_{16}$
2. $(82)_9$

1.3 Question 3

For binary numbers with 4 bits. Determine when there is an overflow and when the calculation can be carrying legally. Use the 2 complement. (The numbers are in base 10).

1. 3+7
2. 2-3
3. 4-7
4. -1-2
5. -5-7

1.4 Question 4

Calculate (use the 2 complement where necessary). Show the details of your calculation.

1. $(B.AB)_{16} + (A.BF)_{16}$
2. $(28A)_{16} - (82)_{16}$
3. $(28C.43)_{16} - (17.12)_{15}$

1.5 Question 5

The 9 complement is defined as follows:

Let $x = x_1x_2...x_m$ be a number in base 10. The 9 complement of the number is $(9-x_1)(9-x_2)...(9-x_m)$. That is, the digit in the i-th place is obtained by replacing it with 9-digit. For example the 9-complement of the number 23 is 76.

Set the codes for digits 0..9 with weights 3,3,2,1 and 4,4,3,-2 such that the 9 complement of the digit will be obtained by switching all the 1 with 0 and all the 0 with 1.

DLOCS Ex1 –

I. Question 1

1. 1729/8 = 216 mod **1**
   216/8 = 27 mod **0**
   27/8 = 3 mod **3**
   3/8 = 0 mod **3**
   $(1729)_{10} = (3301)_8$

2. 8EA2 = $(1000\ 1110\ 1010\ 0010)_2$ =
   $2^{15} + 2^{11} + 2^{10} + 2^9 + 2^7 + 2^5 + 2^1 = 36514$
   (Converting to Binary base was not necessary, could've used powers of 16 instead)

3. $(12.31)_{10} = (12)_{10}.(31)_{10}$
   Integer part first: $(12)_{10} = (1100)_2$. And now to the fraction part:
   We'll add fractions of the type 2^{-n} as long as their sum does not exceed the required 0.31:
   $0.31 = 2^{-2} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8} + 2^{-10} + 2^{-12} + 2^{-13} + 2^{-14} + 2^{-19} + \ldots..$
   The above is accurate up to about a millionth. Theoretically we could go on forever always improving the accuracy (exponentially) up to any requested ε.
   And thus the fraction part approaches: $(.0100111101011100001)_2$.
   And so: $(12.31)_{10} = (1100.0100111101011100001)_2$

4. $(B.AB)_{13} = (11*13^0 + 10*13^{-1} + 11*13^{-2})_{10} = (11 + \frac{141}{169})_{10}$

II. Question 2 (I hope I understood this question correctly)

1. 2-complement of $(28A)_{16}$ using 12-bits:
   Conversion to binary using groups of 4-bits (each hex digit = 4 bits), we get: 0010 1000 1010.
   In order to find the complement we invert the digits and add 1: 1101 0111 0101 + 1 = 1101 0111 0110.
   And so, the 2-complement representation of -28A using 12-bits is **1101 0111 0110.**
   Of course this could be verified using normal addition and expecting a zero (last carry ignored).
2. 2-complement of $(82)_9$ using 8-bits:
   Firstly we shall convert it to binary. To make things simple, I'll use the decimal base as an intermediate mediator. $(82)_9 = (8*9^1 + 2*9^0)_{10} = (74)_{10}$. Using the iterative algorithm as in 1.1 we get $(74)_{10} = (01001010)_2$. As above, we shall invert the digits and add 1 in order to find the representation of $(-82)_9$ using 2's complement and 8-bits: $\mathbf{not}(01001010)_2 + 1 = (10110101)_2 + (1)_2 = \mathbf{(10110110)}_2$

III. Question 3

* Using 2's complement and 4-bits we may represent all integers between -8 and 7 (2^4 altogether).

1. 3+7 > 7 and thus *overflow*. Alternatively: 0011+0111 = 1010 which is negative.

2. 2-3 = -1 and is within the legal range. Alternatively: 0010+1101=1111 which is the correct representation of -1 (dec) using 2's complement.
3. 4-7 = -3 and is within the legal range: 0100+1101=1101 which represents -3 using 2's complement.
4. -1-2 = -3 and is within the legal range: 1111+1110=11101 (last carry ignored) = 1101 = -3 as above.
5. -5-7 < -8 and thus *overflow*: 1101+1001=10110 (last carry ignored) = 0110 = $(6)_{10}$?! Error…

IV. Question 4

1. (B.AB) + (A.BF) in hex. After conversion to binary we get:
   1011.10101011 +
   1010.10111111 =
   10110.01101010
   Back to hexadecimal we get: 16.6A

2. (28A) – (82) in hex. We'll convert to 12-bit binary and use 2's complement:
   $(28A)_{16} = (001010001010)_2$
   $(82)_{16} = (000010000010)_2$
   $(-82)_{16} = (111101111110)_2$
   Binary addition of (28A) and (-82):
   001010001010 +
   111101111110 =
   001000001000 (last carry ignored)
   Back to hexadecimal we get: **(208)**$_{16}$

3. $(28C.43)_{16}$ - $(17.12)_{15}$. We'll use 20-bits to represent these numbers (12 for integer part, 8 for fraction part).
   The hexadecimal number in binary is simply (using groups of 4-bits) 001010001100.01000011
   The quindecimal number converted to decimal is: $1*15^1 + 7*15^0 + 1*15^{-1} + 2*15^{-2} = 4967/225 \sim 23.075555....$,
   Now converting the integer part to binary we get 10111, and converting the fraction part to binary (as in 1.3) we get: .00010011010101110011… and overall: 000000010111.00010011 (up to 2^8 precision) when represented in 20-bits. Using 2's complement we get: 111111101000.11101101.

   <u>Performing addition:</u>

0010 1000 1100 . 0100 0011 +	$(28C.43)_{16}$
1111 1110 1000 . 1110 1101 = (we have a carry from the fraction part)	- $(17.12)_{15}$
0010 0111 0110 . 0011 0000 = **(276.30)**$_{16}$	(last carry ignored)

V. Question 5

Digit	Weight A (3,3,2,1)	Weight B (4,4,3,-2)
0	0000	0000
1	0001	0011
2	0010	1001
3	1000	0010
4	1001	1000
5	0110	0111
6	0111	1101
7	1101	0110
8	1110	1100
9	1111	1111

Explanation: Suppose you wish to find the 9-complement of 7 using Weight A and according to the guidelines. Look at the table in the matching position; 1101 is the code. Its easy to see that according to the weights 1101 = 7 (1*3 + 1*3 + 0*2 + 1*1). Now, perform NOT on all digits and you get: 0010, which according to weight A is 2 (0*3 + 0*3 + 1*2 + 0*1) and indeed (9-7) = 2 as the 9-complement suggests.

One interesting thing to notice is that in fact the upper part of the table is a mirror image of its lower part with all digits reversed (with respect to the matching weight). This does not have to be so, however, since some digits could have more than a single representation using these weights (since 4 bits would allow up to 16 unique codes, but in this example only 10 are being used and there are also weight similarities between the bits in both cases).

1 Digital Systems small exercise 2

Submission date: 26/3/2006.

1.1 Question 1 (10%)

Which of the following 6 basic rules

1. Existance of identity (neutral element).
2. Associative
3. Commutative
4. Closure
5. Distributive
6. Existance of a complement that is, if I is the identity of operator *, the if X*Y = I then Y is X complement.

does not apply for the operands:

+	0	1	2
0	0	0	0
1	0	1	1
2	0	1	2

·	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

1.2 Question 2 (20%)

Prove the following equalities using the boolean axioms only:

1. $x \cdot (x + y) = x$
2. $x + (x'y) = x + y$

3. $X' + Y' = X \oplus Y + X'Y'$

   Where $\oplus$ is the xor operator:

A	B	$\oplus$
0	0	0
0	1	1
1	0	1
1	1	0

4. Find a similar expression for $X' + Y' + Z'$

1.3 Question 3 (20%)

Let F_1

x	y	z	F_1
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

1. Express F_1 as a canonical products and canonical sums.

2. Express the function as a sum of products with the minimal number of terms possible.

3. Express the function as a products of sums with the minimal number of products possible.

1.4 Question 4 - (50%)

Implement the following funcions using NAND (and followed by not) alone. Assume that you can use the literal, its complement (its negation), and the values 0,1. Nand(a,b,c) has the following formula (a*b*c)'.

1. BD + BCD + AB'C'D' + A'B'CD' Use up to 6 gates with 3 entries each

2. (AB + A'B')(CD' + C'D) Use gates with 2 entries each.

תרגיל קטן 2 במתמ"ד (103)

סדרי יצק"ין
בונוס: +3

שאלה 1

- קיום יחידה - קיים איבר $e\in S$ כך שלכל $x\in S$ $e*x=x*e=x$
- קיבוציות - לכל $x,y,z\in S$ מתקיים $x*(y*z)=(x*y)*z$
- חילופיות - לכל $x,y\in S$ מתקיים $x*y=y*x$
- סגירות (סגירות?) לכל $x,y\in S$ $x*y\in S$
- פילוג של * ביחס ל-· - לכל $x,y,z\in S$ $x*(y\cdot z)=(x*y)\cdot(x*z)$
- קיום משלים - לכל $x\in S$ קיים $x'\in S$ כך ש

נסתכל על האופרנדים שמוגדרים ע"י הטבלאות

+	0	1	2
0	0	0	0
1	0	1	1
2	0	1	2

·	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

קיום יחידה: עבור האופרנד · איבר היחידה הוא 0
אכן $0\cdot 0=0$, $0\cdot 1=1$, $0\cdot 2=2$
עבור + איבר היחידה הוא 2 אכן $0+2=0$, $1+2=1$
$2+2=2$

חילופיות: מאחר שהטבלאות סימטריות ברור ששני האופרנדים חילופיים.

סגירות: מאחר שבטבלאות מופיעים רק איברים מ-$\{0,1,2\}$ ברור שהקבוצה חסומה תחת הפעולות.

קיבוציות ופילוג: נשים לב ש- $x\cdot y=\max\{x,y\}$
$x+y=\min\{x,y\}$

מאחר ש $\min\{\min\{x,y\},z\}=\min\{x,\min\{y,z\}\}$
$\max\{\max\{x,y\},z\}=\max\{x,\max\{y,z\}\}$
$\max\{x,\min\{y,z\}\}=\min\{\max\{x,y\},\max\{x,z\}\}$
$\min\{x,\max\{y,z\}\}=\max\{\min\{x,y\},\min\{x,z\}\}$

ברור ששני האופרטורים קיבוציים וכן פילוגיים אחד ביחס לשני.

קיום משלים: אין משלים לכל איבר מקבוצה ביחס ל- +, וביחס ל- ·. למשל, עבור 2 לכל x: $x\cdot 2 = 2 \neq 0$. לכן ל-2 אין משלים ביחס ל- ·. ועבור 0 לכל x: $0+x = 0 \neq 2$ לכן ל-0 אין משלים ביחס ל- +.

שאלה 2

(1) צ"ל: $x\cdot(x+y) = x$

הוכחה:

$$x(x+y) \underset{\text{הגדרת } 0}{=} (x+0)(x+y) \underset{\text{פילוג}}{=} x+(0\cdot y) \underset{\text{חילוף}}{=} x+(y\cdot 0) =$$

$$\underset{\text{חיל אורים}}{=} x+0 \underset{\text{הגדרת } 0}{=} x$$

(11)

כאה ש לכל $x\in B$: $x\cdot 0 = 0$

$$x\cdot 0 \underset{\text{הגדרת } 0}{=} 0 + x\cdot 0 \underset{\text{הגדרת משלים}}{=} xx' + x\cdot 0 \underset{\text{פילוג}}{=} x(x'+0) \underset{\text{הגדרת } 0}{=} xx' \underset{\text{הגדרת משלים}}{=} 0$$

(11)

(2) צ"ל: $x+(x'y) = x+y$

הוכחה:

$$x+(x'y) \underset{\text{פילוג}}{=} \underbrace{(x+x')}_{1}\cdot(x+y) \underset{\text{הגדרת משלים}}{=} x+y$$

(11)

(3) צ"ל: $x'+y' = x\oplus y + x'y'$

הוכחה: נשים לב שמתקיים $x\oplus y = x'y + y'x$

לכן צריך להראות ש: $x'+y' = (x'y+y'x) + x'y'$

$$(x'y+y'x)+x'y' \underset{\text{קיבוץ}}{=} x'y+(y'x+x'y') \underset{\text{חילוף}}{=} x'y+(x'y'+y'x) =$$

$$= (x'y+x'y')+y'x \underset{\text{פילוג}}{=} x'(y+y')+y'x \underset{\text{הגדרת משלים}}{=} x'+y'x =$$

$$\underset{\text{חילוף}}{=} x'+xy' \underset{\text{משפט קודם}}{=} x'+y'$$

(11)

(4) נוסף ב $x'+y'+z' = x\oplus y + y\oplus z + x'y'z'$

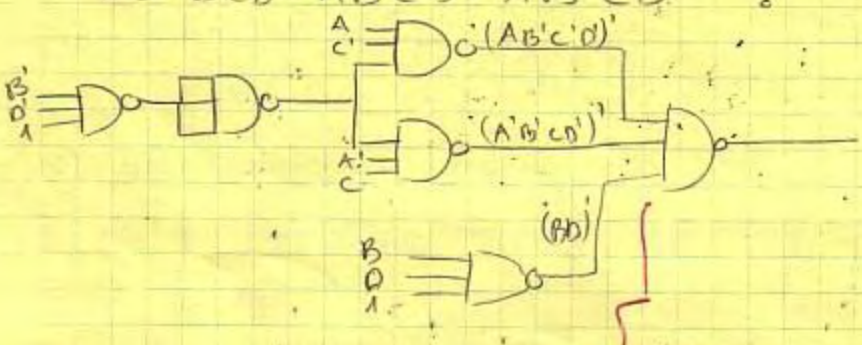
הוכחה:

$$x\oplus y + y\oplus z + x'y'z' = xy' + x'y + y'z + yz' +$$
$$+ x'y'z' = (x+z)y' + (x'+z')y + x'y'z' =$$
$$= ((x+z) + x'z')y' + (x'+z')y =$$
$$= (x+z+x')(x+z+z')y' + (x'+z')y =$$
$$= y' + (x'+z')y = (y'+x'+z')(y'+y) = x'+y'+z'$$

(11)

$\boxed{\text{שאלה 4}}$

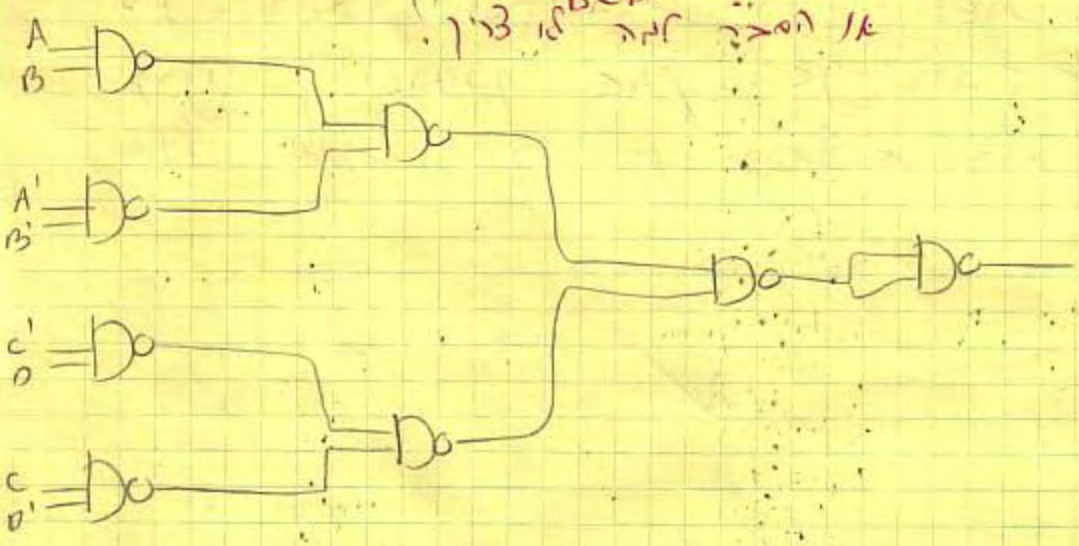
(1) $BD + BCD + AB'C'D + A'B'CD'$:



? - BCD

אין הסבר למה לא בדיק?

(2)



שאלה 3

תרגום הפונקציה הבוליאנית:

x	0	0	0	0	1	1	1	1
y	0	0	1	1	0	0	1	1
z	0	1	0	1	0	1	0	1
f	0	1	1	0	1	0	0	1

(1) נרשום את ה minterms של הקומבינציות בהן $f=1$ ונעשה OR ביניהם אל

$$f = x'y'z + x'yz' + xy'z' + xyz$$

נרשום את ה maxterms של הקומבינציות בהן $f=0$ ונעשה AND ביניהם אל

$$f = (x+y+z)(x+y'+z')(x'+y+z')(x'+y'+z)$$

(2) מילת: בערך קומבינציה שלם בנסיון לפתור לדמות ולקבל מ שהו בשל ערך, עד שנרגלת. ניגה ל שלה בלתי אפשר (:)

בדרך היותר אפקטיבית היו לעשות מפה
הרגו המחוברים שאין. שני הבתים. קומבינציה
בשלה אחרי מלך לכן. הקומבינציה בלבי אפשרי.

מיתוג ומערכות סיפרתיות תרגיל 3

אנא ציינו login על התרגיל.

שאלה 1

יהיו a=01101101 , b=11101101, c=00010000 מספרים בשיטת הנקודה הצפה. כאשר סיבית הסימן בעלת אורך של ביט אחד, האקספוננט בעל אורך של 3 סיביות והמנטיסה בעלת אורך של 4 סיביות.
חשב את (a+b)+c ואת a+(b+c). הסבר את התוצאה.

שאלה 2

בתקן IEEE754 (ראה מצגת לפרטים), מהו המספר הקרוב ביותר לאפס (שאינו אפס) אשר ניתן ליצוג ב double וב single? מהו המספר הגדול ביותר?

שאלה 3

א. כתבו את טבלת האמת של הפונקציה הבוליאנית אשר מתארת את הסכום והנשא של 2 מספרים בינאריים בעלי סיבית אחת. כלומר לפונקציה ישנן שתי כניסות ושתי יציאות. הכניסות של הפונקציה הן המספרים הבינאריים בעלי סיבית אחת a,b. היציאות הן המספרים בעלי סיבית אחת s,c (אשר מיצגים נשא וסכום בהתאמה). לאחר מכן כתבו את הפונקציה כסכום מכפלות ומכפלת סכומים.

ב. כתבו מעגל זהה למספרים בעלי 2 סיביות (שימו לב שהיציאה s היא בעלת 2 סיביות בעוד היציאה c הינה בעלת סיבית אחת).

ג. כתבו את טבלת האמת של הפונקציה אשר עונה לתיאור הבא: לפונקציה 3 כניסות $a_1a_2a_3$ ושמונה יציאות $b_0 \ldots b_7$. הפלט של הפונקציה הוא $b_j=1$ כאשר j הוא המספר העשרוני בייצוג BCD של סיביות $a_1a_2a_3$ ו- $b_i = 0$ עבור $i \neq j$.

שאלה 4

שער Fredkin הינו שער החלפה מותנה. כלומר השער מוגדר כך:

לשער 3 כניסות (C,I1,I2) ו3 יציאות (C,O1,O2). היציאה C היא הכניסה C.
אם C=0 אזי היציאות הן (I1,I2)=(O1,O2).
אם C=1 אזי היציאות הן (I1,I2)=(O2,O1).

הוכיחו ש {Fredkin,0,1} הינה מערכת שלמה. כלומר עליכם לממש AND ו NOT (או OR ו NOT) על ידי שערי Fredkin והערכים 0,1.

שאלה 5

שער Toffoli הינו שער שלילה מותנה. כלומר השער מוגדר כך:

לשער 3 כניסות (I1,I2,V) ו3 יציאות (O1,O2,V1). היציאות O1, O2 הן הכניסות I2 ו I1 בהתאמה.
אם I1=I2=1 אזי V1 = V'.
אחרת, V1=V. הוכיחו ש {Toffoli,1} הינה מערכת שלמה. כלומר עליכם לממש AND ו NOT (או OR ו NOT) על ידי שערי Toffoli והערך 1.

מיתוג ומערכות ספרתיות - תרגיל 3

שאלה 1

יהיו $a = 01\underline{10}1101$, $b = 1\underline{110}1101$, $c = 0\underline{001}0000$ מספרים בשיטת הנקודה הצפה ראשי סיבית הסימן באורך 1, המעריך באורך 3 והמנטיסה באורך 4.

* נחשב את $(a+b)+c$

ל-a,b אותו מעריך ⇐ פשוט נחבר את המנטיסות

$$-\begin{matrix}a\\b\end{matrix} = -\begin{matrix}1101\\1101\end{matrix} = +\begin{matrix}1101\\0011\\\hline 1\,0000\end{matrix} \Rightarrow (a+b)+c = c = 1\times 2^{1-3} = 1\times 2^{-2} = 1/4$$

נשא סופי. התעלם ממנו

* נחשב את $a+(b+c)$

ל-b,c מעריכים שונים ⇐ יש להשוות את המעריך של c ל-b11 ע"י הזזה ימינה (ונזכור את הסיבית הבולטת. (ראית)

0:001:00000
0:010:10000
0:011:01000
0:100:00100
0:101:00010
0:110:00001

עכשיו אפשר לחבר את המנטיסות החדשות של b,c:

$$-\begin{matrix}c\\b\end{matrix} = -\begin{matrix}00001\\11010\end{matrix} = +\begin{matrix}000001\\100110\\\hline 100111\end{matrix} \xrightarrow[\text{משלים ל-2}]{} 11001$$

אבל הדיוק שלנו הוא 4 סביות אל $b+c = 11101100$

עכשיו נחבר רק את a. המעריכים שווים אל כך צריך לחבר מנטיסות:

$$-\begin{matrix}a\\b+c\end{matrix} = -\begin{matrix}1101\\1100\end{matrix} = +\begin{matrix}01101\\10100\\\hline 1\,00001\end{matrix} \Rightarrow a+(b+c) = 2^{-4}\times 2^{6-3} = 1/2$$

נשא סופי. התעלם ממנו

התוצאות שונות במיוחד הנורמלי בגלל לא מנטיסה של 4 סביות - איבדנו דיוק במעבר מ- 11001 ל- 1100

שאלה 2

המספר הכי קרוב לאפס שאינו אפס מתקבל כאשר המעריך הוא $e = e_{min} - 1$ והמנטיסה שונה מאפס. אז המספר הוא

$$0.f \times 2^{e_{min}}$$

המספר הכי גדול מתקבל כאשר המעריך $e_{min} < e < e_{max}$ והמנטיסה שונה מאפס. אז המספר הוא $1.f \times 2^{e}$

ב IEEE754 ב single המנטיסה באורך 23, המעריך באורך 8.

לכן המספר הכי קטן הוא

$$0\ \underbrace{0\ldots0}_{8\text{ סיביות}}\ \underbrace{0\ldots01}_{23\text{ סיביות}}$$

והמספר הכי גדול הוא

$$0\ \underbrace{01\ldots1}_{7\text{ סיביות}}\ \underbrace{1\ldots1}_{23\text{ סיביות}}$$

ב double המנטיסה באורך 52 והמעריך באורך 11

לכן המספר הכי קטן הוא

$$0\ \underbrace{0\ldots0}_{11\text{ סיביות}}\ \underbrace{0\ldots01}_{52\text{ סיביות}}$$

והמספר הכי גדול

$$0\ \underbrace{01\ldots1}_{10\text{ סיביות}}\ \underbrace{1\ldots1}_{52\text{ סיביות}}$$

שאלה 3

א) טבלת האמת של פונקציית הסכום והנשא:

a	b	s	c
0	0	0	0
0	1	$\boxed{1}$	0
1	0	$\boxed{1}$	0
1	1	0	$\boxed{1}$

$$s(a,b) = a'b + b'a = (a+b)(a'+b')$$

$$c(a,b) = ab = (a+b)(a+b')(a'+b)$$

ב) נכתוב טבלת אמת לבור קלט בן 2 סיביות:

a_1	a_2	b_1	b_2	S_1	S_2	C
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	1	0	0
0	0	1	1	1	1	0
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	1	0
1	0	1	0	0	0	1
1	0	1	1	0	1	1
1	1	0	0	1	1	0
1	1	0	1	0	0	1
1	1	1	0	0	1	1
1	1	1	1	1	0	1

ג) נכתוב טבלת אמת לבור הפונקציה הבאה:

לפונקציה 3 כניסות a_1, a_2, a_3 ושמונה יציאות $b_0, \dots, b_7$

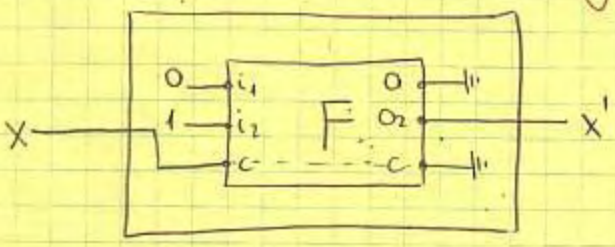
הפלט $b_j = 1$ כאשר j הוא המספר המשרוני ביצוג BCD

של $a_1 a_2 a_3$ ואחרת $b_i = 0$.

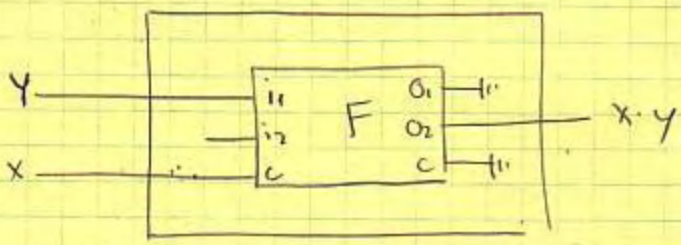
a_1	a_2	a_3	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7
0	0	0	1							
0	0	1		1						
0	1	0			1					
0	1	1				1				
1	0	0					1			
1	0	1						1		
1	1	0							1	
1	1	1								1

שאלה 4

נממש NOT:

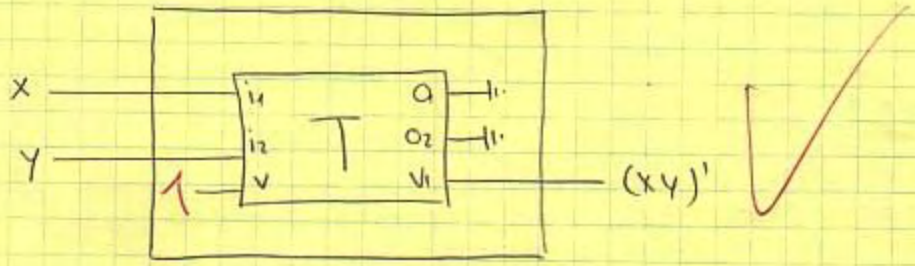


נממש AND:



שאלה 5

הובחנו ש {NAND} היא אס' שמה
נאמש שער NAND מאמצות שער טופולי והקבוע
1 ומכאן יבנה ש {Toffoli, 1} שמה



מיתוג ומערכות סיפרתיות תרגיל 4

אנא ציינו login על התרגיל.
לכל השאלות משקל זהה.

שאלה 1

בנו את השער AND ממתגים p ו n

שאלה 2

פשט את הביטויים הבאים (שימו לב, הביטויים בצורת מכפלת סכומים):
על ידי מפות קרנו

1. $F(x,y,z) = \Pi(0,1,4,5)$
2. $F(A,B,C,D) = \Pi(0,1,2,3,4,10,11)$
3. $F(w,x,y,z) = \Pi(1,3,5,7,13,15)$

שאלה 3

הביטוי הבוליאני BE+B'DE' התקבל מהביטוי A'BE+BCDE+BC'D'E+A'B'DE'+B'C'DE'.
האם ישנם ביטויים אדישים? אם כן, מהם?

שאלה 4

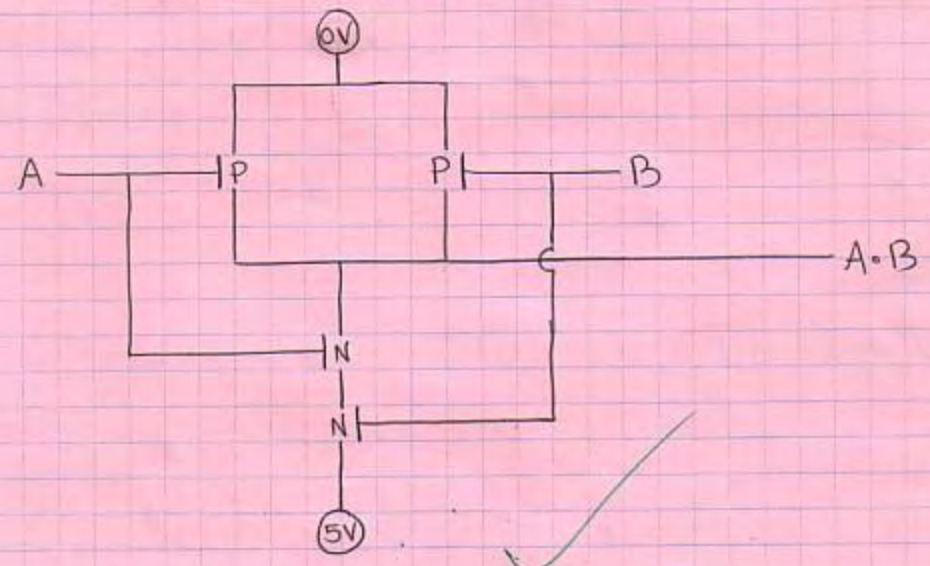
ממשו את הפונקציות הבאות תוך שימוש בשערים האדישים d (כלומר כאשר d=1 הצירוף אדיש).
הנח שנתונות שתי הצורות של כל משתנה (הרגיל והמשלים).
אנא פרטו את הדרך וכתבו את הביטוי בנוסף למעגל.

ממשו בעזרת שני שערי NOR בעלי 3 כניסות בלבד.	1. F = A'B'C' + AB'D + A'B'CD' d = ABC + AB'D'
בעזרת שערי NAND (אין הגבלה על המספר) בעלי 3 כניסות.	2. B'D + B'C + ABCD d = A'BD + AB'C'D'

מיתוג ולוגיקות סדרתיות - תרגול 4

שאלה 1

שער AND מתוך n, p :



אם A=0V או B=0V אז מתגי ה-p יתחברו ונקבל 0V בידיאה. אם גם A=5V ו-B=5V אז יתחברו שני מתגי ה-N ונקבל 5V בידיאה

שאלה 2

① נצייר מפת קרנו של $f(x,y,z) = \Pi(0,1,4,5) = \Sigma(2,3,6,7)$

x \ yz	00	01	11	10
0	0	0	1	1
1	0	0	1	1

$\Rightarrow f(x,y,z) = y$

② נצייר מפת קרנו של הפונקציה

$f(A,B,C,D) = \Pi(0,1,2,3,4,10,11) = \Sigma(5,6,7,8,9,12,13,14,15)$

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	1	1	1	1
10	1	1	0	0

$f' = A'B' + A'C'D' + B'C$

$\Rightarrow f = (A+B)(A+C+D)(B+C')$

③ נצייר מפת קרנו של הפונקציה

$f(W,X,Y,Z) = \Pi(1,3,5,7,13,15) = \Sigma(0,2,4,6,8,9,10,11,12,14)$

WX \ YZ	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	0	0	1
10	1	1	1	1

$f' = W'Z + XZ$

$\Rightarrow f = (W+Z')(X'+Z')$

שאלה 3

נצייר מפת קרנו של הפונקציות:

$A'BE + BCDE + BC'D'E + A'B'DE' + B'C'DE'$

AB \ CDE	000	001	011	010	110	111	101	100
00								
01								
11								
10								

$BE + B'DE'$

AB \ CDE	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

המינים הבודדים הם אלה שלא משותפים לשתי המפות:

$\Sigma(27, 22, 29)$

שאלה 4

(1) הדרך היחידה לממש פונק' בעלות 2 שערי NOR בעלי 3 כניסות היא:

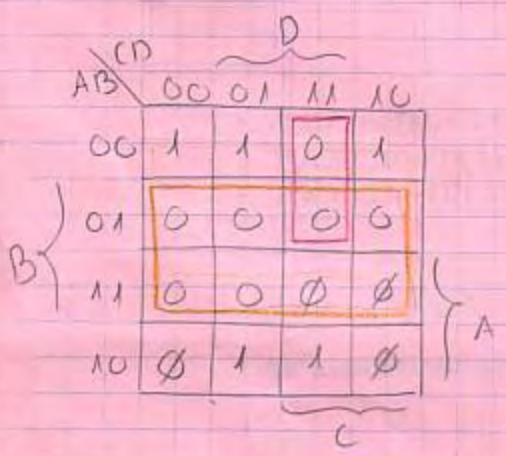
משר $F = NOR(x, y, NOR(z, w, v))$

$x, y, z, w, v \in \{A, B, C, D, A', B', C', D'\}$

$F = (x + y + (z + w + v)')' = (x + y + z'w'v')'$

נרצה להביע את F' ע"י סכום מכפלות. משר ב-2 מהמכפלות איבר אחד (מלבן עם 8 ריבועים) ובעוד מכפלה 3 איברים (מלבן עם 2 ריבועים).

נצייר מפת קרנו של הפונק' הנ"ל:



$F' = B + B + A'CD$

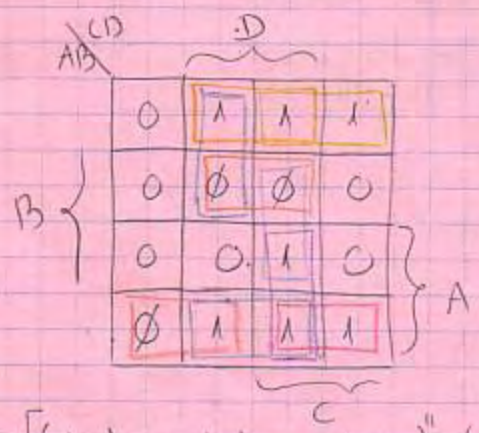
$\Downarrow$

$F = (B + B + A'CD)' =$

$= (B + B + (A + C' + D')')' =$

$= NOR(B, B, NOR(A, C', D'))$

(2) נצייר מפת קרנו של הפונק' הנ"ל $F = B'D' + B'C + ABCD$ עם הביטויים האדישים $d = A'BD + AB'C'D'$



בשביל לממש את הפונק' ע"י שערי NAND בעלי 3 כניסות יש לבקש 3 מלבנים עם שטח 2.

מדשלים (-3)

(-1) אפשר למצוא ביטוי הפוך יותר.

$F = [(A'B'D + A'B'C + A'BD)'' + (AB'C' + AB'D + AB'C)'' + (A'C'D + BCD + ACD)'']'' =$

$= NAND([(A'B'D)'(A'B'C)'(A'BD)']', [(AB'C')'(AB'D)'(AB'C)']', [(A'C'D)'(BCD)'(ACD)']') =$

$= NAND(NAND(NAND(A', B', D), NAND(A', B', C), NAND(A', B, D)), NAND(NAND(A, B', C'),$

$NAND(A, B', D), NAND(A, B', C)), NAND(NAND(A', C', D), NAND(BCD), NAND(ACD)))$

מיתוג ומערכות ספרתיות תרגיל 5 (תרגיל גדול).

משחק החיים מוגדר כך:

משחק החיים

תהי מטריצה nXm. כל תא יכול להיות חי או מת.
לכל תא (חוץ מהקצוות) ישנם 8 שכנים

1	2	3
8	*	4
7	6	5

בהינתן דור y התא בדור y+1 יהיה חי אם מספר השכנים שלו נע בין min ל max. אם מספר השכנים של התא קטן ממש מ min התא ימות מבדידות, אם מספר השכנים של התא גדול ממש מ max התא ימות מחנק.

דוגמא (* תסמן תא חי): כאן min=3 ו max = 6

דור y → y+1 דור

*			→			
	*					

*	*	*	→		*	
	*				*	

	*		→		*	
*	*	*		*	*	*
	*				*	

*	*	*	→	*	*	*
*	*	*		*		*
*	*	*		*	*	*

תרגיל

העבודה בתרגיל זה הינה בזוגות.
ממשו מעגל בעזרת תוכנת TKGate אשר מקבל לוח בגודל 16x14 בדור y ומחזיר לוח בדור y+1.
עליכם להגיש אלקטרונית לאתר הקורס את הקובץ v. של המעגל יחד עם קובץ README אשר מפרט את האלגוריתם הכללי ותיאור קצר של כל מודול. כמו כן, עליכם להגיש תדפיס של כל מודול לתא ברוס 2-.
במודול הראשי צריכה להיות אפשרות להגדיר את מצב הלוח (על ידי DIP switches) ולראות את התוצאה (על ידי Led Bars). כמו כן, מספר השכנים המינימאלי ומספר השכנים המקסימאלי (כך שהתא יחיה בדור

הבא) הוא חלק מהפרמטרים של המעגל. אנא הקפידו על תיעוד של כל מודול (גם במעגל על ידי Comment).

הדרכה

עברו על ה Tutorial אשר מופיע באתר.
על מנת לממש את המעגל אנו נזדקק לכמה מודולי עזר:

ייצוג המטריצה

המטריצה תיוצג על ידי 16 מחרוזות בנות 14 סיביות כל אחת. אולם על מנת לטפל במקרי הקצה נשנה את הקלט במעט (כיצד?) כך שאותם מודולים יטפלו בתא ללא תלות במיקומו (כלומר בקצה או במרכז המפה).

מודולי עזר

1. בנו מודול בשם Comparator אשר מקבל 2 מספרים בני 4 סיביות, B,A ומחזיר 0 אם A>B ו 1 אם A≤B.
2. בנו מודול בשם Subtractor אשר מקבל 2 מספרים בני 4 סיביות A,B ומחזיר את A-B.
3. בנו מודול בשם Decode16 אשר מקבל מספר i בן 4 סיביות ומחזיר מחרוזת בת 16 סיביות כאשר הסיבית ה i בעלת ערך 1 וכל השאר בעלי ערך 0.
לדוגמא: אם i=5 המודול יחזיר את המחרוזת הבאה 0000100000000000.
רמז: השתמשו בשני Decode 1-8.

מודולים

1. Adjacent_Alive: מקבל מחרוזת בת 16 סיביות ומספר i בעל 4 סיביות ומחזיר 1 אם המקום ה i חי ו 0 אם לא.
2. Adjacent_Line_Count: מקבל מחרוזת בת 16 סיביות, מספר i בעל 4 סיביות וסיבית Center ומחזיר את מספר השכנים מספר בעל 4 סיביות) שיש למקום ה i. כלומר אם Center=1 השכנים ייספרו במקומות ה i-1,i+1 ואם Center=0 השכנים ייספרו במקומות i-1,i,i+1.
3. Adjacent_Count: מקבל 3 שורות בנות 16 סיביות כל אחת
LineUp,LineCenter,LineDown ומספר i בעל 4 סיביות. מחזיר את מספר השכנים (מספר בעל 4 ספרות) שיש לתא בשורה LineCenter במקום ה i.
4. Cell_Next_Generation: מקבל מקבל 3 שורות בנות 16 סיביות כל אחת
LineUp,LineCenter,LineDown, מספר i בעל 4 סיביות ומספר data בעל 8 סיביות.
מחזיר 1 אם התא חי בדור הבא ו 0 אם לא (ב data אנו מקודדים את min ו max – שני מספרים בעלי 4 סיביות כל אחד).
5. Line_Next_Generation: מקבל מקבל 3 שורות בנות 16 סיביות כל אחת
LineUp,LineCenter,LineDown , ומספר data בעל 8 סיביות.
מחזיר את הדור הבא של השורה LineCenter.

6. Matrix_Next_Generation: מקבל 16 שורות של 14 סיביות ומחזיר את הדור הבא של כל השורות.

שימו לב: אין חובה לממש דווקא את המבנה המתואר כאן.

מיתוג ומערכות סיפרתיות תרגיל 6

אנא ציינו login על התרגיל.

שאלה 1 (30%)

סרטט דיאגרמה לוגית של דלגלג מסוג JK בעזרת שערי NOR ושערי AND. אפשר העברת הדלגלג למצב set (Q = 1 ו Q' = 0) ולמצב clear (Q=0 ו Q' = 1) באופן א-סינכרוני (ללא תלות בשעון).
כלומר הוסף 2 כניסות אחת לביצוע set ואחת לביצוע clear.

בשאלות הבאות בכל דופק שעון נכנסת סיבית חדשה לקלט I. אתם רשאים להשתמש בכל סוג דלגלג.

שאלה 2 (30%)

עצבו מעגל אשר ערך היציאה שלו הוא 1 אם מספר ה 1 שנקראו (מקלט I) עד כה הוא זוגי.

שאלה 3 (30%)

עצבו מעגל אשר ערך היציאה שלו הוא 1 אם מספר ההופעות של המחרוזת 101 מתחלק ב 3.
שימו לב במחרוזת 10101 ישנן 2 הופעות.

שאלה 4 (10%)

עצבו מעגל אשר לו 2 סיביות יציאה : O_0 ו O_1. אשר מגדירות את המספר הבינארי O_1O_0.
הערך של היציאות האלו הוא השארית של החלוקה ב 3 של המספר שנקרא עד עכשיו.
לדוגמא:

הפלט של המעגל (O_1O_0)	המספר הנקרא (עשרוני)	המחרוזת שנקרא	מספר הדפקים שעברו
00	0	0	0
01	1	01	1
10	2	010	2
10	5	0101	3

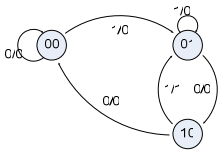
שימו לב שבכל דופק שעון נקראת סיבית אחת והמספר הוא אוסף כל הסיביות אשר נקראו עד כה.

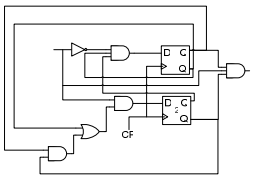
הדרכה:

1. בהינתן מחרוזת של סיביות, מה ערך המספר לאחר שאנו משרשרים לו סיבית אחת מימין.
2. ענו על סעיף 1 רק הפעם כתבו את המספר העשרוני לאחר חלוקה ב 3 (כלומר תוצאת החילוק והשארית). שימו לב לשארית.

Solution to Exercise 6 Q2 Encounter module

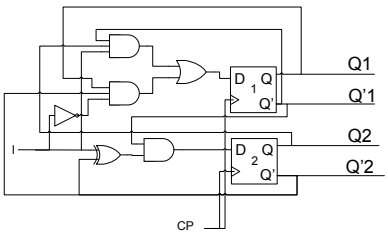
Output 1 if 101 is encountered

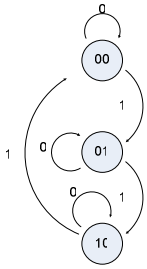




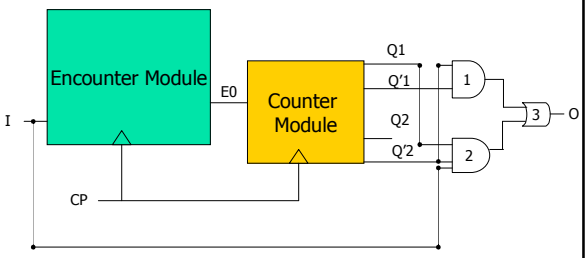
Solution to Exercise 6 Q2 Counter Module

A mod 3 counter for the number of appearances

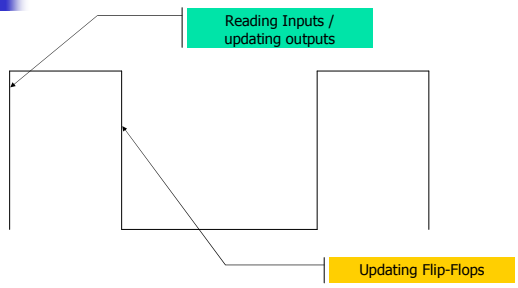




The Complete Circuit Q2



How does the solution work?



How does the solution work?

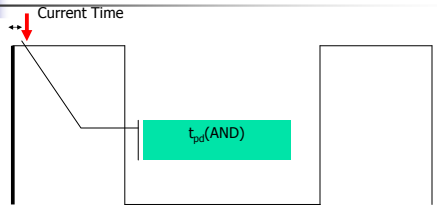
Suppose 2 101 were encountered. The Counter module is In state 10. Suppose that now 10 were encountered, The encounter module is in state 10

How does the solution work?

Current Time

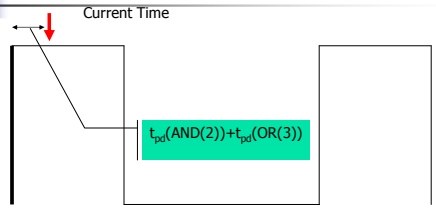
I =1	Counter Module = 10	Encounter Module = 10	E0=0
And(1) = 0	And(2) = 0	O = 0	

How does the solution work?



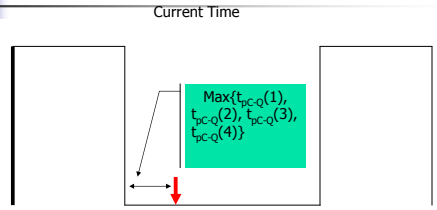
I =1	Counter Module = 10	Encounter Module = 10	E0=1
And(1) = 0	And(2) = 0	O = 0	

How does the solution work?



I =1	Counter Module = 10	Encounter Module = 10	E0=1
And(1) = 0	And(2) = 1	O = 1	

How does the solution work?



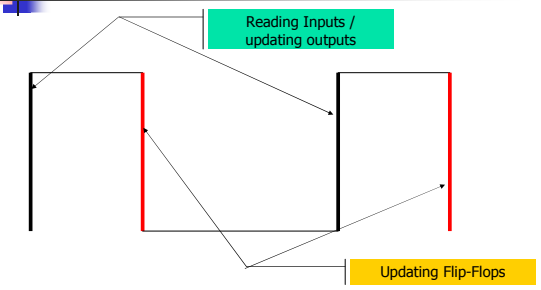
I =1	Counter Module = 00	Encounter Module = 01	E0=0
And(1) = 1	And(2) = 0	O = 1	

How does the solution work?

I =0	Counter Module = 00	Encounter Module = 01	E0=0
And(1) = 1	And(2) = 0	O = 1	

Solution to Question 2 Exercise 7

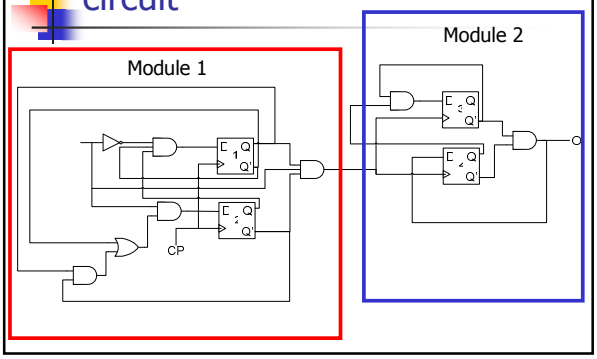
Events in synchronized system



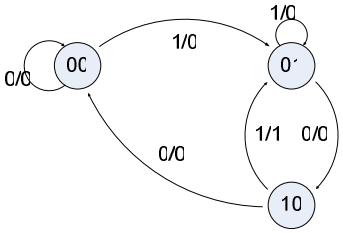
The requirements:

- All the inputs to the FF must be stable at least t_{setup} before the update of the FF
- All the outputs of the circuit must be stable on their right value, as close as possible to the updating of inputs.

Analyze the timing of the circuit

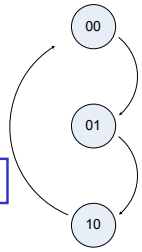


1. Analyze the state machine of module 1



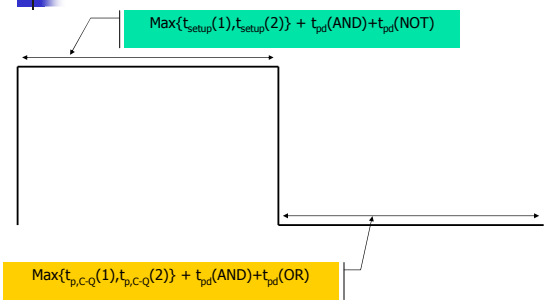
1. Analyze the state machine of module 2

Note: Module 2 is a counter.
It progress each update of the FF
To the next state.

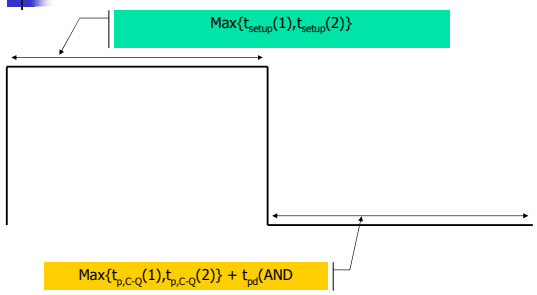


The update of the FF is when the clock
Of the FF goes from 1 to 0.

Set the timing of module 1 when it is independent



Set the timing of module 2 when it is independent



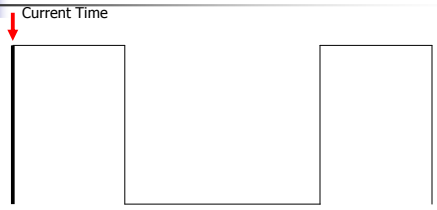
The "0" of a circuit:

The time it take the circuit output to stabilize from the negtive pulse of the clock.

Examine the "Clock" of module 2.

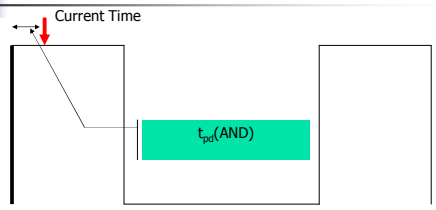
1. Suppose that module 1 is in state 10.
2. Suppose that now the input is 1

The clock of module 2



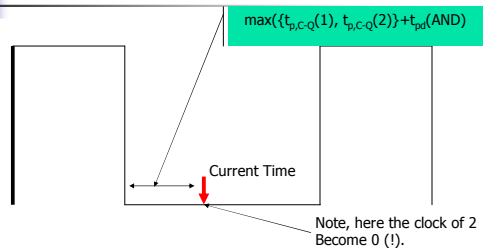
I =1	Encounter Module = 10	Module 1 output = 0

The clock of module 2



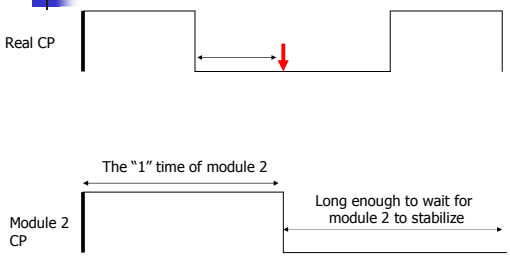
I =1	Encounter Module = 10	Module 1 output = 1 The Clock of module 2

The clock of module 2

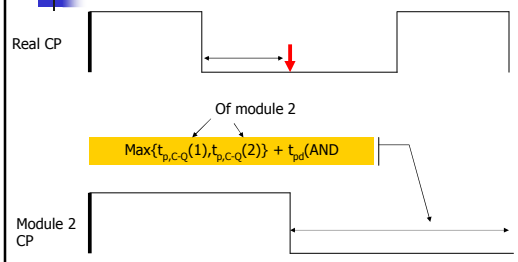


I =1	Encounter Module = 01	Module 1 output = 0 The Clock of module 2

The time "0" of module 2 should be



The time "0" of module 2 should be



The real time "0" of module 2

Of module 1

$$\text{Max}\{t_{p,C\text{-}Q}(1),t_{p,C\text{-}Q}(2)\} + t_{pd}(\text{AND})+$$

$$\text{Max}\{t_{p,C\text{-}Q}(1),t_{p,C\text{-}Q}(2)\} + t_{pd}(\text{AND})$$

Of module 2

What about time "1" of module 2?

- Required Time "1" of module 2 is t_{setup}.
- Time "1" of the clock of module "2" is:
  - Time "1" of module 1 + max({$t_{p,C\text{-}Q}$(1), $t_{p,C\text{-}Q}$(2)}+t_{pd}(AND)

Of module 1

Therefore, no problem with time 1.

The timing of the total circuit

- Time "1" : The max over time "1" of all the FF:

$$\text{Max}\{t_{setup}(1),t_{setup}(2)\} + t_{pd}(\text{AND})+t_{pd}(\text{NOT})$$

- Time "0": The max over all time "0" of all the FF:

$$\text{Max}\{t_{p,C\text{-}Q}(1),t_{p,C\text{-}Q}(2)\} + 2*t_{pd}(\text{AND})+ \text{Max}\{t_{p,C\text{-}Q}(1),t_{p,C\text{-}Q}(2)\}$$

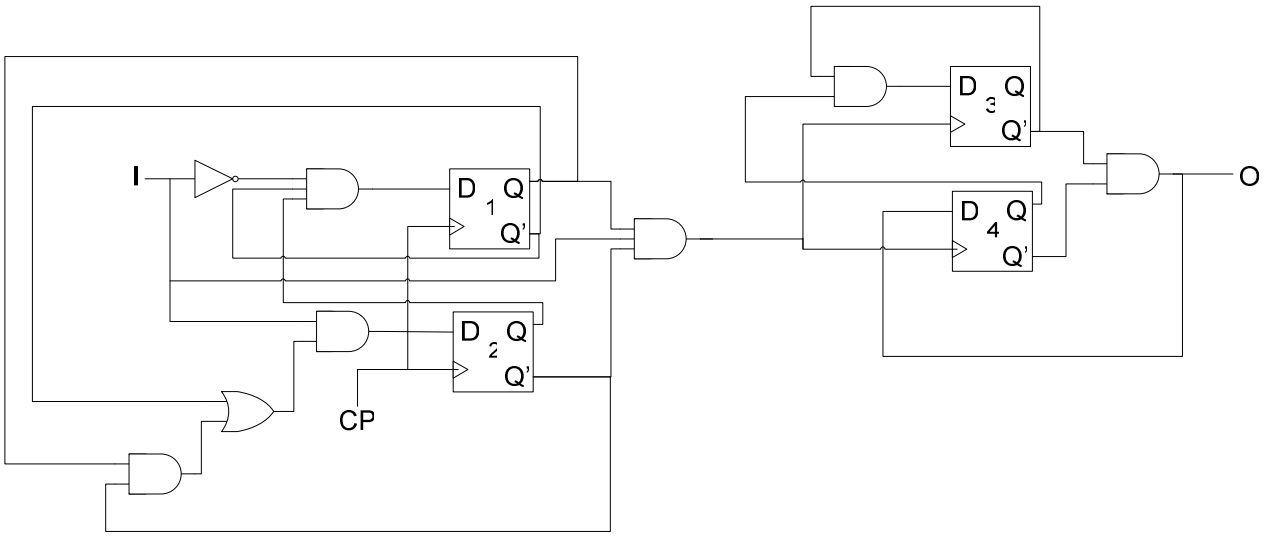
Of module 1 Of module 2

מיתוג ומערכות סיפרתיות תרגיל 7

אנא ציינו login על התרגיל.

במעגלים הבאים נכנסת **מימין** סיבית חדשה לקלט I בכל דופק שעון חיובי.

שאלה 1 (20%)



ערך היציאה ,O, של המעגל בשרטוט הוא 1 אם מספר ההופעות של המחרוזת 101 מתחלק ב 3.
שימו לב במחרוזת 10101 ישנן 2 הופעות. (שאלה 3, תרגיל 6).
במעגל 2 מודולים.

1. ציינו לאיזה מודול שייך כל דלגלג. (רמז, התחילו מהיציאה לכיוון הכניסה).
2. שרטטו את דיאגרמת המצבים של כל מודול
3. הסבירו מה התפקיד של כל מודול.

שאלה 2 (30%)

הניחו כי

פרמטר	ערך
Tpd(AND)	1
Tpd(OR)	2
Tpd(NOT)	0.5
Tcd(AND)	0.2
Tcd(OR)	0.1
Tcd(NOT)	0.05
Tp,C-Q(FF)	1.5
Tc,C-Q(FF)	0.1
Thold(FF)	0.02

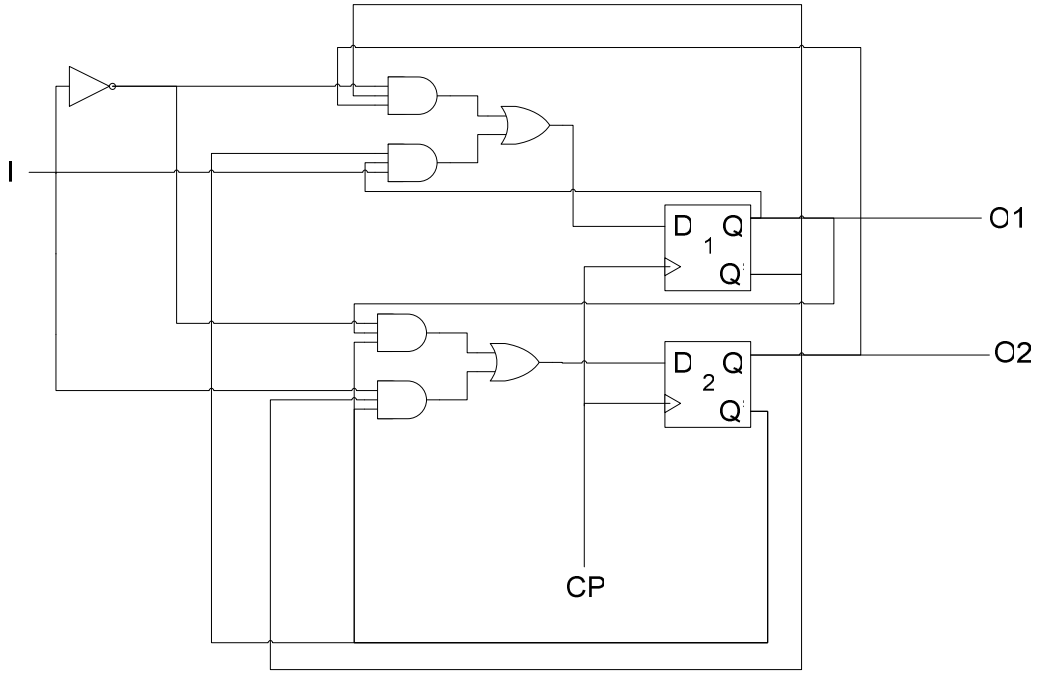
TSetup(FF)	0.05

כאשר FF הוא דלגלג כלשהוא.

ענו על השאלות הבאות עבור המעגל בשאלה 1:

1. מהו זמן המחזור המינימלי? מהו זמן 0 המינימלי? מהו זמן 1 המינימלי?
2. מהוא ה Tpd של המעגל?
3. האם המעגל יפעל כמתוכנן אם ימומש בעזרת הרכיבים שלעיל ? נמקו.

שאלה 3 (30%)



למעגל בשרטוט 2 סיביות יציאה : O_2 ו O_1. אשר מגדירות את המספר הבינארי O_2O_1.
הערך של היציאות האלו הוא השארית של החלוקה ב 3 של המספר שנקרא עד עכשיו.
לדוגמא:

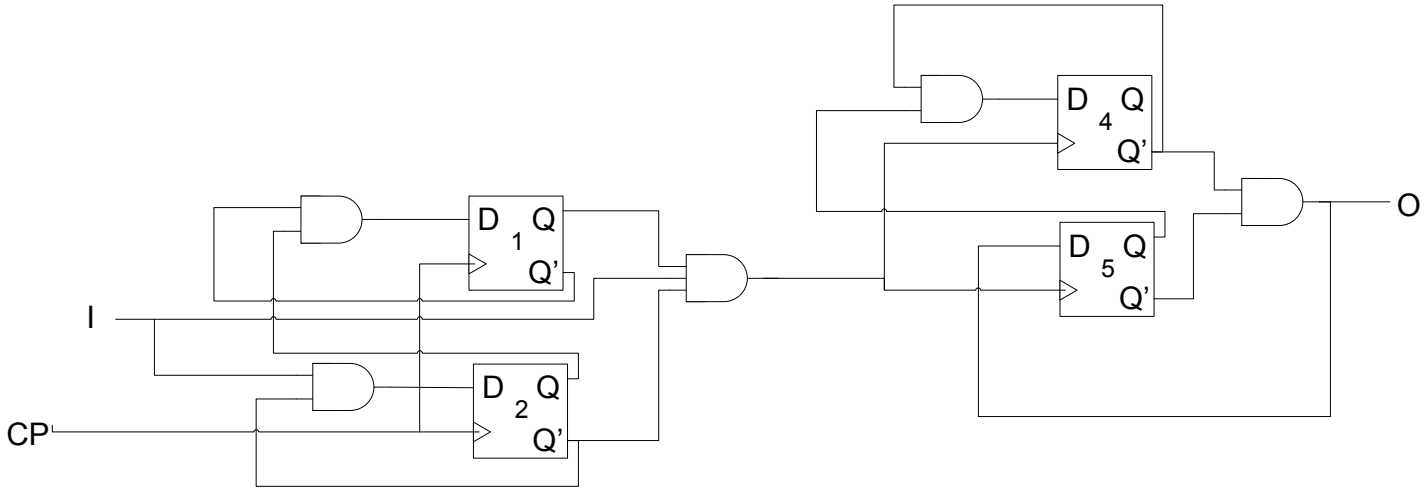
הפלט של המעגל (O_1O_0)	המספר הנקרא (עשרוני)	המחרוזת שנקרא	מספר הדפקים שעברו
00	0	0	0
01	1	01	1
10	2	010	2
10	5	0101	3

שימו לב שבכל דופק שעון נקראת סיבית אחת והמספר הוא אוסף כל הסיביות אשר נקראו עד כה.
(שאלה 4, תרגיל 6).

תחת ההנחות על הרכיבים שבשאלה 2.
ענו על השאלות הבאות עבור המעגל בשרטוט:

1. מהו זמן המחזור המינימלי? מהו זמן 0 המינימלי? מהו זמן 1 המינימלי?
2. מהוא ה Tpd של המעגל?
3. האם המעגל יפעל כמתוכנן אם ימומש בעזרת הרכיבים שלעיל ? נמקו.

שאלה 4 (10%)



מעגל זה מתוכנן גם הוא לפתור את שאלה 3, תרגיל 6.
במעגל 2 מודולים.

1. ציינו לאיזה מודול שייך כל דלגלג. (רמז, התחילו מהיציאה לכיוון הכניסה).
2. שרטטו את מכונת המצבים של כל מודול
3. הסבירו מה התפקיד של כל מודול.
4. מעגל זה אינו מחזיר תשובה נכונה תמיד. תנו דוגמא לסדרה בה טועה המעגל. תנו דוגמה לסדרה בה המעגל מוציא תשובה נכונה.

שאלה 5 (10%)

תחת ההנחות על הרכיבים שבשאלה 2.
ענו על השאלות הבאות עבור המעגל בשאלה 4:

4. מהו זמן המחזור המינימלי? מהו זמן 0 המינימלי? מהו זמן 1 המינימלי?

5. מהוא ה Tpd של המעגל?
6. האם המעגל יפעל כמתוכנן אם ימומש בעזרת הרכיבים שלעיל ? נמקו.

מיתוג ומערכות סיפרתיות תרגיל 8

אנא ציינו login על התרגיל כמו כן, אנא שמרו על עותק מפתרונכם.
אפשר להגיש את התרגיל בזוגות.

בתרגיל זה נממש מעגל חשמלי אשר מדמה מחשבון פשוט.
למחשבון ישנם 4 תאי זיכרון בשם a..d ו4 סוגי פעולות.
לפעולה החלקים הבאים: Operation Name []Destination,Operand1,[Operand2]
כאשר [Operand2] עשוי לא להופיע.

הפעולות הן:

קלט	משמעות
a[]regname,5,7[]	Regname = 5+7
a[]regname,6[]	Regname= regname+6
s[]regname,7,6[]	regname = 7-6
s[]regname,6[]	regname=regname-6
e	עצירה

כאשר [] מציין את התו רווח (20h = 20 בקידוד בבסיס 16) ו [regname] הוא אחד התווים a..d .
שימו לב, בקלט ישנה לכל היותר ספרה אחת, אך התוצאה יכולה להיות גדולה מ10 או קטנה מ 10-.

הקלט נקרא הקלט נקרא כקוד ascii (בעל 8 סיביות, ראו טבלה מצורפת בסוף התרגיל) בדופק חיובי ממקור קלט חיצוני בעל 8 סיביות בשם TerminalInput כאשר הסיבית nextchar דלוקה.
הערך של TermianlInput הוא 00h כאשר nextchar כבויה. כלומר עליכם להדליק ולכבות את סיבית nextchar כאשר אתם מוכנים לקלוט את האות הבאה.

הנחות

1. הוראות מגיעות ברצף עד להופעת e. (שימו לב שהרווח בסוף הוראה היא חלק מההוראה).
2. ההוראות חוקיות. (לא צריך לטפל במקרי קצה)
3. אין צורך להתחשב בתזמון. כלומר זמן המעבר מ 0 ל 1 (ומ1 ל 0) הוא מידי הן בשעון והן ברכיבים הצירופיים. עדכון הדלגלג מתרחש מייד עם מעבר הדופק.

הוראות הגשה:

1. פרט למעגל עליכם להגיש את האלגוריתמים, דיאגרמות המצבים וטבלת ה ROM אשר שמשו אתכם בבנייתו (צעדים 1 ו 3 בהדרכה).
2. הנכם רשאים לשרטט קו בודד ולציין עליו את רוחבו (כמו ב TKGate). כמו כן, הנכם רשאים לחבר קוים בעלי מספר סיביות שונה, אך הקפידו לציין אילו סיביות מחוברות לאן.
3. הנכם רשאים להשתמש בMUX, DeMuX, Decoder , Encoder , Comparator, Full Adder ו ROM. כאשר הנכם משתמשים ב ROM רשמו את הטבלה הצרובה בו. אם הנכם משתמשים במעגלים אחרים אנא ציינו את מימושכם.

הדרכה:

זוהי הדרכה בלבד, ואינכם חייבים לממש את האלגוריתם הכתוב.

1. השלימו את האלגוריתם הבא:

1. Read next character into Input
2. Select on I
   a. I = 0 : Operation ← Decode_Operation(Input)
   b. …..
   c. …..
3. ….

כאשר Decode_Operaion הינו מודולים (צירופיים) אשר ממירים את קוד ASCII לערך הרצוי במעגל.

2. שימו לב, מכיוון שנבצע פעולות חשבוניות על הקלט רצוי לשמור אותו בפורמט בינארי. אולם הקלט הוא בפורמט ASCII, כיצד תמיר בין הפורמטים? (שימו לב, הקלט הוא של סיבית בודדת).
3. כתבו את האוגרים והדלגלגים אשר תשתמשו בהם.
4. המירו את האלגוריתם בשלב 1 לאלגוריתם המתאים למעגל.

0. Set NextChar = 1 , Input Load = 1	Now Terminalline holds the next character, nc
1. Set NextChar = 0 , Input Load = 0	Now TerminalLine=0., Input hold nc.
2. When I=0 : Load of register [Operation] = 1	
3. When I=0 : Operation = Decode_Operation(Input)	
4. When I=0 : Load of register [Operation] = 0	

האלגוריתם בצעד 3 הינו דוגמא בלבד ולכן חסרים בו כמה שלבים.

5. שרטטו את דיאגרמת המצבים של האלגוריתם.
6. שרטטו את המעגל.

קוד ASCII לאותיות הרלבנטיות לתרגיל

קוד Ascii (בבסיס 16)	האות
61h	a
62h	b
63h	c
64h	d
65h	E
73h	s
30h	0
31h	1
32h	2
33h	3
34h	4
35h	5
36h	6
37h	7
38h	8

9	39h
רווח	20h

Solution to Exercise 8

The algorithm

1. I == 0 : Op ← Decode_Op(TerminalInput)
   I == 1 : Dest ← Decode_Dest(TerminalInput)
   I == 2 : Dig1 ← Decode_Digit(TerminalInput)
   I == 3 : Dig2 ← Decode_Digit(TerminalInput)
2. If Op == ‘e’ E ← 0
3. Tmp ←TerminalInput, I = I+1
4. (I == 3) && Tmp == 20h (space):
   1. Reg[Dest] ← Op(Reg[Dest],Dig1)
   2. I = 0

   (I==4):
   1. Reg[Dest] ← Op(Dig1,Dig2)
   2. I = 0

The algorithm

Execute if Condition

1. I == 0 : Op ← Decode_Op(TerminalInput)
   I == 1 : Dest ← Decode_Dest(TerminalInput)
   I == 2 : Dig1 ← Decode_Digit(TerminalInput)
   I == 3 : Dig2 ← Decode_Digit(TerminalInput)
2. If Op == ‘e’ E ← 0
3. Tmp ←TerminalInput, I = I+1
4. (I == 3) && Tmp == 20h (space):
   1. Reg[Dest] ← Op(Reg[Dest],Dig1)
   2. I = 0

   (I==4):
   1. Reg[Dest] ← Op(Dig1,Dig2)
   2. I = 0

The algorithm

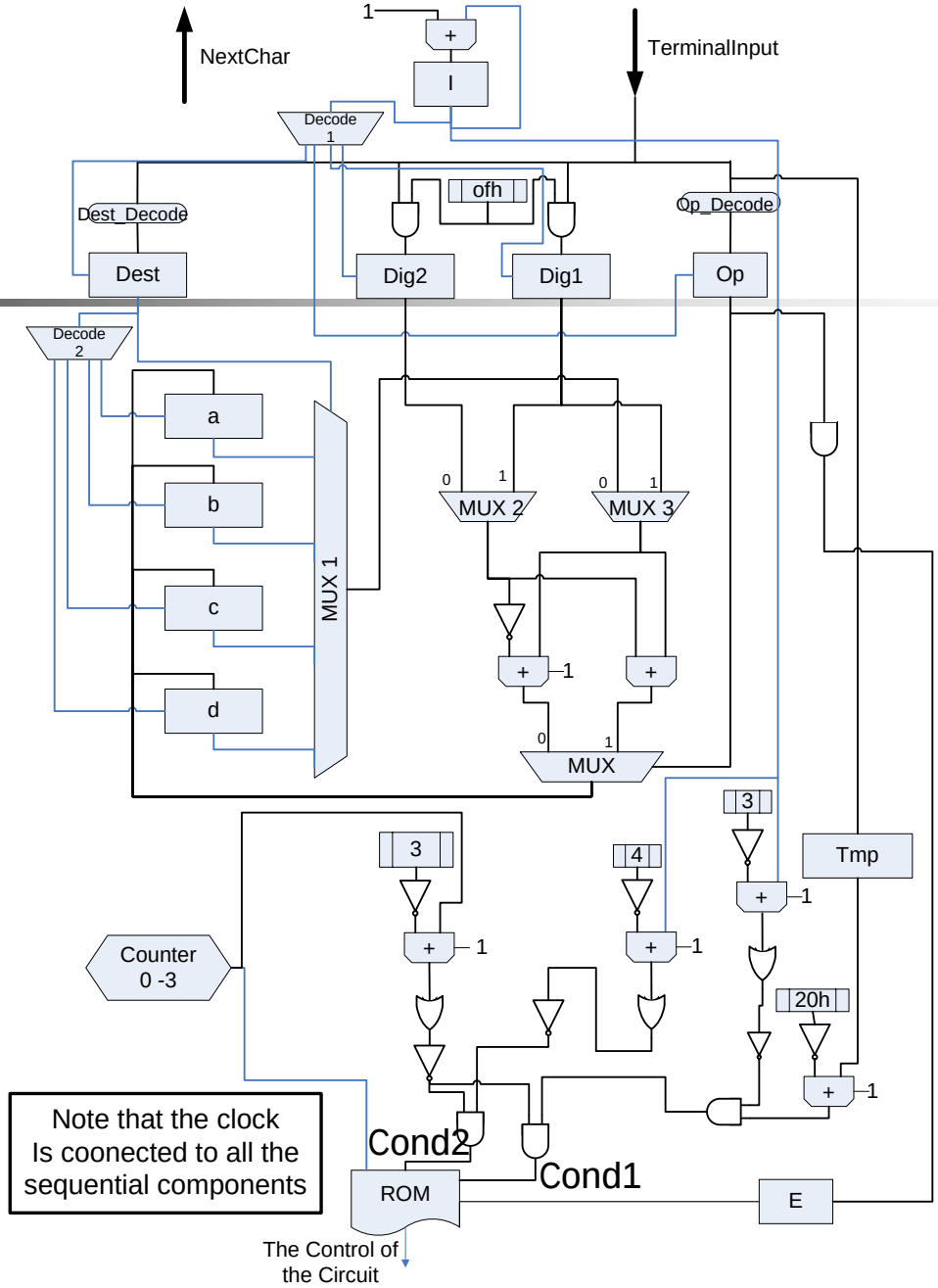
1. I == 0 : Op ← Decode_Op(TerminalInput)
   I == 1 : Dest ← Decode_Dest(TerminalInput)
   I == 2 : Dig1 ← Decode_Digit(TerminalInput)
   I == 3 : Dig2 ← Decode_Digit(TerminalInput)
2. If Op == ‘e’ E ← 0
3. Tmp ←TerminalInput, I = I+1
4. (I == 3) && Tmp == 20h (space):
   1. Reg[Dest] ← Op(Reg[Dest],Dig1)
   2. I = 0

   (I==4):
   1. Reg[Dest] ← Op(Dig1,Dig2)
   2. I = 0

Put result in the reg Written in dest

Register sizes:

- Op – 2 bit
- Dest – 2 bit
- Dig1 – 8 bits
- Dig2 – 8 bits
- a to d – 8 bits
- Tmp – 8 bits
- E = 1



Note that the clock
Is coonected to all the
sequential components

Control of the circuit

- NextChar
- Decode1Enable
- Decode2Enable
- Mux2
- Mux3
- Iwrite
- Tmpwrite

The control of the circuit

Input				Output						
E	Step	Cond 1	Cond 2	nxtchar	decode1 enable	decode2 enable	MUX2	MUX3	Iwrite	tmpwrite
1	0	0	0	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
1	2	0	0	1	0	0	0	0	1	1
1	3	1	0	0	0	1	1	0	0	0
1	3	0	1	0	0	1	0	1	0	0

Op_Decode

Input	Output
01100001 (61h = a)	00
01110011 (73h = s)	01
01100101 (65h = e)	11

Dest_Decode

Input	Output
01100001 (61h = a)	00
01110010 (62h = b)	01
01100011 (63h = c)	10
01100100 (64h = d)	11

מיתוג ומערכות סיפרתיות תרגיל 9 – חלק א

אנא ציינו login על התרגיל
אפשר להגיש את התרגיל בשלישיות.
עדכונים והערות יופיעו בראש האתר.

בתרגיל זה נממש ב tkgate מעגל אשר מקבל פקודות מהמשתמש ומבצע אותן (להדרכה על זיכרון, דלגלגים וקלט מהמשתמש ב tkgate ראו בסוף התרגיל).

נכתוב 4 תוכנות קצרות על ידי השפה אשר תוגדר בחלק הבא:

1. בנו תוכנה (מאוסף הפעולות של המעבד) אשר מדפיסה מספר גדול מ 0 (וקטן מ 65000).
   כלומר התוכנה היא :
   a. Set $1 = MyNumnber
   b. Print The number on the terminal.

2. בחרו מספר בין 30000- ו 30000 וכתבו תוכנה אשר מדפיסה אותו (כולל הסימן – אם הוא שלילי).
3. כתבו תוכנה אשר קולטת מהטרמינל מספר חיובי (בין 0 ל 65000) וממירה אותו למספר בינארי. קריאת המספר נגמרת כאשר התו 0b (enter) נקרא.
4. כתבו תוכנה אשר קולטת מהטרמינל מספר (בין 30000 to 30000-) וממירה אותו למספר בינארי.
   אם המספר שלילי התו – (קוד 2D) יהיה לפני המספר. קריאת המספר נגמרת כאשר התו 0b (enter) נקרא.

הפעולות של המעבד:

על המעבד לבצע את פעולות הבאות:

Micro instruction	Meaning	Comments
	Arithmetic / logic operations	
$3 = $2 + $1	Set register $3 with $2 + $1	
$3 = $2 - $1	Set register $3 with $2 - $1	
$3 = mod($2,$1)	Set register $3 with b such that $1*a + b = $2 (b < a)	Use the tkgate division
$3 = $2/$1	Set register $3 with a such that $1*a + b = $2 (b < a)	Use the tkgate division
$3 = $2 * $1	Set register $3 with $2 * $1	Use tkgate multiplier
$3 = AND($2,$1)	Set register $3 with And($2,$1)	
$3 = NOT ($1)	Set register $3 with not ($1)	
$3 = OR($2,$1)	Set register $3 with Or($2,$1)	
$3 = H($1)	Set register $3 with upper 8 bits of $1	For example (in hexadecimal) if $1 = 0f0bh then $3 = 000fh. Implement using

		the shift in the ALU
$3 = L($1)	Set register $3 with lower 8 bits of $1	
lShift $1,$2	Shift left $1 in $2 bits (the lower 8 bits)	
rShift $1,$2	Shift right $1 in $2 bits (the lower 8 bits)	
Set-on-less $1,$2,$3	$1 = 1 if $2< $3 $1=0 otherwise	
$1 = VAL	Put the VAL in register $1	
Memory Access operations		
Load $1,Addr($2)	Load into $1 the value in Addr+$2	
Store $1,Addr($2)	Store in $1 the value in Addr+$2	
Push $1	Push $1 to the stack	
Push PC	Push the current value of PC to the stack. (note that the current value of PC already points to the next instruction).	
Pop $1	Pop the stack into register $1.	
I/O operations		
Read $1	Read a character from the terminal into the lower 8 bits of register $1	See the section on tty in order to see how to implement it
Print $1	Print the character in the lower 8 bits of $1	See the section on tty in order to see how to implement it
Program control		
J Addr	Jump to address Addr	
Return $1	Jump to address $1	
Jne $1,$2,Addr	Jump to Addr if $1 is different than $2	
Je $1,$2,Addr	Jump to Addr if $1 equals $2	

כאשר $x הוא סימון לרגיסטר מספר x. הכוונה בסימון $3=$2+$1 היא לכך שניתן לחבר כל זוג רגיסטרים ולהציב את התוצאה ברגיסטר שלישי.

מספר הרגיסטרים של המעבד לא יעלה על 7 (רגיסטר 0 הוא 0 קבוע ולא ניתן לכתוב עליו).
מלבד רגיסטרים אלו, ניתן להשתמש ברגיסטרים אחרים (למצביע של המחסנית, ל PC , IR , MAR וכו'), אך רגיסטרים אלו לא יכללו בחישוב של הפעולות האריתמטיות.
כל הרגיסטרים הם בעלי 16 סיביות בדיוק (גם ה SP, PC וה IR) – אין להשתמש ברגיסטרים בעלי פחות או יותר סיביות.

אלגוריתם הפעולה של המעבד.

המעבר עובד לפי האלגוריתם הבא:

1. CAR = 0 : טען את הפעולה הבאה לתוך IR. PC++, CAR++
2. CAR = 1: הצב ב CAR את הערך המתאים לפעולה שנמצאת בתוך ה IR.
3. CAR = ...: בצע את הפעולה. (בסופה הצב CAR=0).

שלבים 1 ו 2 נקראים שלב ה fetch .

המחסנית

על מנת לממש קריאה לפרוצדורות אנו נזדקק למחסנית. מנגנון הקריאה עצמו יוסבר בחלק ב' של התרגיל.
על מנת לממש מחסנית אנו נחזיק רגיסטר מיוחד בשם SP. ברגיסטר זה תוחזק הכתובת בזיכרון של ראש המחסנית הנוכחי. על כן נגדיר 2 פעולות : Push ו Pop.
הפעולה Push תמומש כך:

a. push PC (or $1) - דחוף את PC למחסנית. כלומר:
   i. SP = SP+1
   ii. Load PC (or $1),SP (שים את הערך שבPC בכתובת שב SP).
b. pop $1 – הוצא את הערך בראש המחסנית והצב אותו ברגיסטר.
   i. Store $1,SP
   ii. SP = SP-1.

הדרכה

1. קראו את ההדרכה על רגיסטרים, RAM ו TTY באתר של TKGate. כמו כן, מצורף באתר קובץ דוגמא לעבודה עם RAM. על מנת להבין טוב יותר את העבודה עם TTY, חברו switches לכל כניסה ו Led לכל יציאה שהיא ביט בודד, ו DIP Switch ו Led Bar לכניסות שהן 8 ביטים. זכרו שההדפסה היא בקוד ascii, ולכן, קודים שהם לא מספרים או ספרות לא יודפסו. נסו לדוגמה את 61h – האות a.
2. ממשו את המעגל על הנייר כשלב ראשון. ממשו את המעגל ב tkgate רק לאחר שממשתם את **כל** ההוראות על הנייר.
3. בנו ALU בעל הפעולות הבאות:
   c. חיבור
   d. חיסור
   e. כפל (הנכם רשאים להשתמש במעגל הכפל של tkgate).
   f. שארית בחילוק (הנכם רשאים להשתמש במעגל החילוק של tkgate).
   g. חילוק (הנכם רשאים להשתמש במעגל החילוק של tkgate).

יש לאפשר ביצוע הסטה ימינה או שמאלה של התוצאה של ה ALU במספר סיביות שמתקבל (השתמשו באיבר ההסטה של tkgate).

4. בנו טבלה של מבנה ההוראות לדוגמא:

סוג ההוראה	תת-סוג ההוראה	אופרנד 1	....

5. כתבו טבלה בה מתואר לכל פעולה מהו אוסף מיקרו-פעולות הנדרש כדי לממש אותה (זכרו את מיקרו הפעולה fetch).
6.
7. בנו את המעגל ללא ה control. כלומר חברו את רכיבי המערכת (לדוגמא הרגיסטרים, RAM, TTY , CAR,IR) בעזרת ה MUX וה decoder ההכרחיים.
8. מכיוון שכל פעולה יכולה להכיל כמה מיקרו פעולות, אנו זקוקים לטבלה אשר תתאם בין ההוראה לבין המיקרו הוראות.
בנו את טבלת ה CAR, כלומר לכל פעולה מהוא ה CAR הרלוונטי.

נניח להוראה אריתמטית דרושה מיקרו פעולה אחת (פרט ל fetch) ולload דרושות 2 מיקרו פעולות. אזי טבלה אפשרית היא:

CAR	Instruction
00010	Arithmetic operation
00011	Load
00100	

בטבלה זו
9. בנו טבלת אשר מתאמת בין ה CAR לcontrol :
לדוגמא,

Input	Outputs		
CAR	PC Load	IR load	Ect..
00000	1	1	עוד סיביות control.

כלומר ב CAR 00000 (דהיינו, fetch), עלינו לטעון את ה PC ואת ה IR.

מכיוון שעליכם להמיר את הטבלה לקוד הקסדצימלי מומלץ לסמן כל 4 ביטים בטבלה בצבע (או במסגרת) על מנת להאיץ את ההמרה .

10. המירו את טבלה 6 לקוד הקסדצימאלי, כתבו את קובץ הזיכרון המתאים, וחברו את ה control של המעגל.
11. כאשר נממש תוכנה נכתוב ראשית את אוסף ההוראות. לאחר מכן נארגן בטבלה אצ הערכים המתאימים ולבסוף נמיר לקוד בקסדצימאלי.
לדוגמא, התוכנה הבאה מציבה את הערכים 1 ו a0 ב $1 וב $2(בהתאמה) ומציבה ב $5 את הסכום שלהם.

Instruction	Type (2 bits)		Instruction (4 bits)				Empty (1bit)	Operand 1 (3bits)			Operand 2 (3 bits)			Operand 3 (3 bits)			Value (16bits)
$1 = 0001h	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1	0001h
	3				c				0				1				0001
$2 = 00a0h	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	0	00a0h
	3				c				0				2				00a0
$5 = $2+$1	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	00000h
	0							0				5				5	0000h

ולכן הקובץ הזיכרון של התוכנה יראה כך:

0/ 3c010001
1/ 3c0200a0
2/ 00550000

שימו לב מספר הביטים ביציאה הוא 32, מספר הביטים בכניסה (אי אפשר לראות זאת ישירות) הוא 16.

Debug

מציאת שגיאות במעגל גדול אינה פשוטה. על מנת לפשט את התהליך עליכם לתת שמות עם משמעות לכניסות ולרכיבים השונים. לחיצה ממושכת על קו תוך כדי ריצה, תחשוף את ערכו בקידוד בינארי (אם הוא ביט בודד) או בהקסדצימאלי.
כך תוכלו לבדוק כל תו בקרה רלבנטי.
זכרו להתקדם בצעדים בודדים (הסימן של K עם השעון עליו).
על מנת לבדוק האם המעגל תקין לאחר השלמתו, כתבו תוכנה קצרה (בת שורה או שתיים) שבודקת כל פעולה.
לדוגמא :

1. $1 = 0f0fh
2. $2 = 10
3. Store $1,100($2)

על מנת לבדוק את תוכן הזיכרון (ועל מנת לקבוע אותו) הינכם יכולים לטעון או לכתוב אותו לקובץ. לחצו על הכפתור הימני לבחרו באופציה dump (או load).

הערות על המימוש

1. הגודל המכסימלי של ROM הוא 32bit. אם יש צורך ביותר מ 32bit השתמשו ב2 (או יותר) ROM.
2. בגרסא 1.8.5 של tkgate צריך להוסיף buffer (שער הזהות) בין ה רגיסטר לבין כניסת הכתובת של ה RAM.
3. לאתחול המערכת הוסף switch ליציאות ה clear של הרגיסטר אשר יאתחל את כל הרגיסטרים ל 0. כלומר כל עוד ה Switch במצב off הרגיסטרים יקבעו בערך 0 . כאשר הswitch יעבור למצב on המעגל יתחיל לעבוד.
4. אין לשים ROM או RAM בתוך מודול (שימו אותו ב main, וחברו כניסה לתוך המודול).
5. הקלט נקרא ונכתב בקוד בעל 8 סיביות. אולם מכיוון שהרגיסטרים הינם בעלי 16 סיביות, נתייחס רק ל 8 סיביות התחתונות.
6. שימו לב כי סיבית ה DSR דולקת כאשר מתקבלת האות הבאה מהמשתמש. אולם אם הפעולה המתבצעת אינה קריאה של אות, סיבית הבקרה תתאפס והאות הנוכחית לא תקרא. על מנת לפתור את הבעיה הזו, בנו את הבקרה כך שקלט 0 לא יאפס את הDSR אלא ישאיר אותה בערך הנוכחי שלה. בסוף קריאת האות, אפסו את הDSR.
7. אם אין אף אות (כלומר RTS=0) קראו את התו המיוחד ff00.

8. על מנת שהכתיבה ל RAM תעבוד עלינו לבצע את השלבים הבאים:
   a. חשב את הכתובת הרצויה
   b. טען את הכתובת לאוגר הכתובת
   c. כתוב את המידע לאוגר.

שימו לב, חובה לשים שער זהות בין האוגר של הכתובת לבין כניסת ה RAM. (ראו דוגמא מצורפת).

9. כוונו את השעון להיות בעל cycle של כ 2000ns.

מבנה קבצי הזיכרון

לקובץ זיכרון (של RAM או של ROM יש את המבנה הבא:

Address/ data data data

כלומר בזיכרון בו רוחב הכתובת הוא 4 ורוחב הפלט (כלומר המילה בזיכרון) הוא 8 הקוד הבא

a/ 0f 0a 01
e/ 01
10/ 3f

יציב במקום a את הערך 0f , במקום b את הערך 0a וכן הלאה.

קבצים אלא ניתן לטעון תוך כדי ריצה (על ידי בחירת הזיכרון ובחירת load לאחר לחיצה על הכפתור הימני), או לטעון מראש על ידי בחירת הקובץ בחלון ה properties של הזיכרון.

הוראות הגשה.

1. יש להגיש עותק אלקטרוני של קובץ המעגל בצירוף README וכל קבצי הזיכרון הרלוונטיים. קראו לקבצי הזיכרון של תוכנות program1 program2 ו program3 1,2 ו 3.
2. יש להגיש עותק מודפס של כל המודולים אשר השתמשתם בהם לפיתרון יש להגיש

מידע נוסף :

לקריאה נוספת פנו ל:

- Digital logic and computer design , Mano.
  (הספר באנגלית בלבד)
  - Chapter 10 Control logic design
  - Chapter 11 Computer design
  - Chapter 12 – microcomputer system design
- Computer organization and design, chapter 5 patterson and hennessy.