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: .000100110101011110011… 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.

סדרי יצק"ל
במ"ח: 3+

שאלה 1

- קיום יחידה - קיים איבר $e \in S$ כך שלכל $x \in S$ $x*e=e*x=x$
- קיבוציות - לכל $x,y,z \in S$ מתקיים $(x*y)*z = x*(y*z)$
- חילופיות - לכל $x,y \in S$ מתקיים $x*y = y*x$
- חסמאות (סגירות?) לכל $x,y \in S$ $x*y \in S$
- פילוג של $*$ מימין ל-$+$ - לכל $x,y,z \in S$ $(x+y)*z=(x*z)+(y*z)$

נסתכל על האופרציות שמוגדרות ע"י הטבלאות

$\cdot$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

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

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

עבור $+$ איבר היחידה הוא 2 אכן $0 = 2+0$, $1 = 2+1$
$1 - 2 = 2+2$

חילופיות: ראינו שהטבלאות סימטריות מכאן שלשתי האופרציות חילופיים.

סגירות: ראינו שבטבלאות מופיעים רק איברים מ-$\{0,1,2\}$ מכאן שהקבוצה סגורה תחת הפעולות.

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

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

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

מכאן שהפעולות $\cdot$ ו-$+$ מקיימות קיבוציות ופילוג זה על זה.

קיום מנגדים: אין מנגדים עבור איבר מהקבוצה בי"ס ל- +, ובי"ס ל- ·. למשל, עבור 2 עבור x: $2 \cdot x \neq 0$. לכן ל-2 אין מנגדים בי"ס ל- ·. ועבור 0 עבור x $x + 0 \neq 0$ לכן ל-0 אין מנגדים בי"ס ל- +.

שאלה 2

① צ"ל: $x \cdot (y+x) = x$

הוכחה:

$$x(x+y) = (0+x)(x+y) = (0+x)x + (0 \cdot y) = x + 0 = x$$

(ב) (ב) (ב) (ב) (ב)

(ב) כאשר שדה $x \in \mathbb{R}$ $0 \cdot x = 0$:

$$0 \cdot x = (0 + x')x = 0 \cdot x + x'x = 0 \cdot x + 0 = 0 \cdot x$$

(ב) (ב) (ב) (ב) (ב)

② צ"ל: $y + x = (y + x') + x$

הוכחה:

$$y + x = (y + x) \cdot (x' + x) = (y + x')$$

(ב) (ב) (ב) (ב)

③ צ"ל: $y \cdot x + y \oplus x = y + x$

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

לכן נרים לפרטים ש: $y \cdot x + (x \cdot y + y \cdot x) = y + x$

$$= (x \cdot y + y \cdot x) + y \cdot x = y \cdot x + (x \cdot y + y \cdot x) = y + x$$

(ב) (ב) (ב) (ב)

$$y + x = x + y$$

(ב) (ב) (ב) (ב)

④ ראינו ב $x'+y'+z' = x \oplus y + y \oplus z + x'y'z'$

הוכחה:

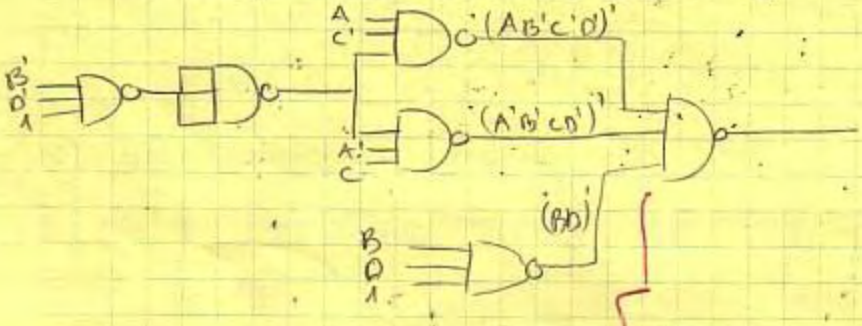
$$x \oplus y + y \oplus z + x'y'z' = xy' + x'y + yz' + y'z + x'y'z' =$$
$$= (x+z)y' + (x'+z')y + x'y'z' =$$
$$= ((x+z) + x'z')y' + (x'+z')y =$$
$$= (x+z+x')(x+z+z')y' + (x'+z')y =$$
$$= y' + (x'+z')y = (y'+x'+z')(y'+y) = x'+y'+z'$$

(יש!)

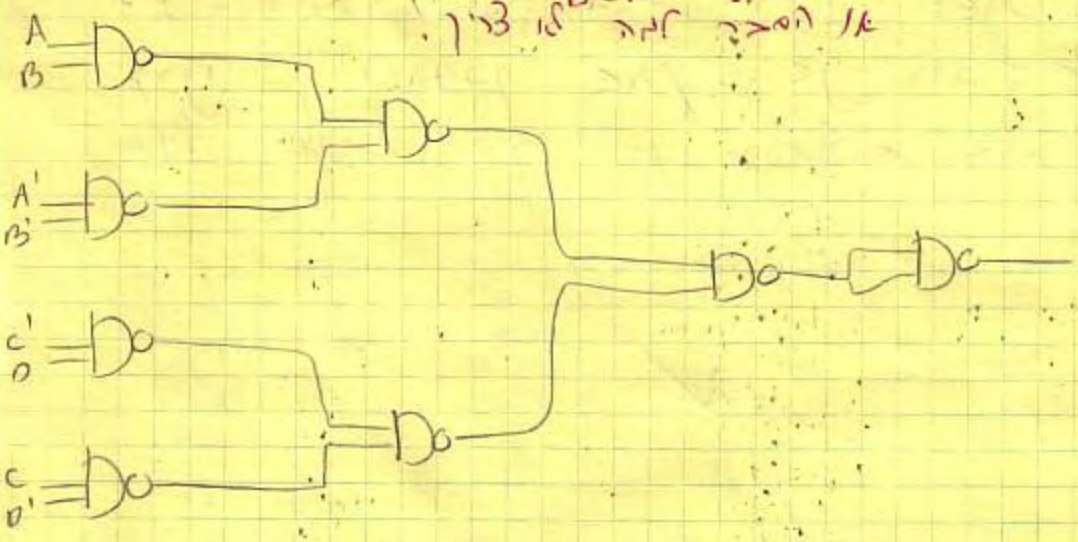
✓

שאלה 4

① $BD + BCD + AB'C'D + A'B'CD'$



✓



? BCD - אין הסבר למה צריך

✓

שאלה 3

נתונה 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 שבהן הפונקציה מקבלת $f=1$ ונקשר ביניהם באמצעות OR.

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

נרשום את ה maxterms שבהן הפונקציה מקבלת $f=0$ ונקשר ביניהם באמצעות AND.

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

② מלית: מרוב שהוא שלם ומסוים לפתרון! לא נמצא ולקבל מי שהוא בשל ערך, עד שנכנעת. ניצה ל שלה מלית אפשרי ②

בגדול היית אמקלוגי היו לעשות מפה קרנו להוכיח שוויון. אין פורמלים. סמוכים בשלה אולי יתן לכן. הצדקות בלבי שבשי.

✓

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 (O1,O2)=(I1,I2).
אם C=1 אזי היציאות הן (O2,O1)=(I1,I2).

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.

שאלה 1

יהיו $a = 01101110$, $b = 10110111$, $c = 00001000$ מספרים בשיטת המשלים לשתיים כאשר סיבית הסימן באותן 1, המערך באותן 3 והמנטיסה באותן 4.

* נחשב את $(a+b)+c$

$a-b$ אותו מערך ⇐ פשוט נהפוך את המנטיסות

$$a - b = 01101110 + 01001001 = 10110111$$

$$(a+b)+c = (a+(-b))+c = 1 \times 2^{1-3} \times 2^{-2} = 1/4$$

נשאר סה"כ. זה קרה אחרי

$$\begin{array}{r} 10111 \\ 00100 \\ \hline 11011 \end{array} \quad \begin{array}{r} 11011 \\ 10111 \\ \hline 00010 \end{array}$$

* נחשב את $a+(b+c)$

$c-q$ אחריהם שונים ⇐ יש להציג את המערך של c $q-011$ ל"י, בכדי שנוכל את הסיבית הנוספת. (ראו)

$$\begin{array}{l} 00000:100:0 \\ 00001:010:0 \\ 00010:110:0 \\ 00100:001:0 \\ 01000:101:0 \\ 10000:011:0 \end{array}$$

עכשיו אפשר לחבר את המנטיסות החפשיות של c וq:

$$\begin{array}{r} 10000 \\ 01011 \\ \hline 11011 \end{array} \quad \begin{array}{r} 00001 \\ 00100 \\ \hline 00101 \end{array} \quad \begin{array}{r} 00000 \\ 00010 \\ \hline 00010 \end{array} \quad \begin{array}{r} 00000 \\ 00001 \\ \hline 00001 \end{array} = c - q$$

$$10011 \xrightarrow{\text{משלים-2}} 11101$$

אבל בדיוק שלנו היא 4 סיביות את $a+c = 10110111$

עכשיו נחבר לכך את b. המערכים שווים אך כך צריך לחבר מנטיסות:

$$a+(b+c) = 1 \times 2^{6-3} \times 2^{-4} = 1/2$$

$$\begin{array}{r} 10110 \\ 00100 \\ \hline 10000 \end{array} \quad \begin{array}{r} 10111 \\ 01100 \\ \hline 00011 \end{array} = a - (b+c)$$

נשאר סה"כ. זה קרה אחרי

שאלה 2

המספר הכי קרוב לאפס שאינו אפס מתקבל כאשר הערך המעריך הוא $e = e_{min} - 1$ והמנטיסה שווה לאפס. כלומר המספר הוא

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

המספר הכי גדול מתקבל כאשר הערך המעריך $e < e_{max} - 1$ $e > e_{min}$ והמנטיסה שווה לאפס. כלומר המספר הוא

$$1.f \times 2^{e}$$

א. אלגוריתם ה single המנטיסה באורך 23, המעריך באורך 8.

לכן המספר הכי קטן הוא

והמספר הכי גדול הוא

ב. אלגוריתם ה double המנטיסה באורך 52 והמעריך באורך 11

לכן המספר הכי קטן הוא

והמספר הכי גדול

שאלה 3

א. טבלת האמת של פונקציית הסכום והנשא:

a	b	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

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

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

ב. נבנה טבלת אמת עבור 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	1	1	0
0	1	0	1	0	0	1
0	1	1	0	0	1	1
0	1	1	1	1	0	1
1	0	0	0	1	0	1
1	0	0	1	1	1	1
1	0	1	0	0	0	1
1	0	1	1	0	1	1
1	1	0	0	0	1	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	1

ג. נבנה טבלת אמת עבור הפונקציה הבאה:

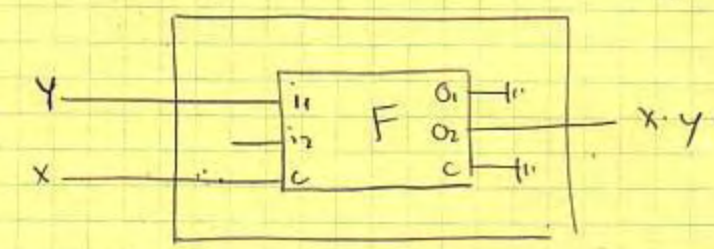
לפונקציה 3 כניסות a_3, a_2, a_1 ושמונה יציאות $b_7, \dots, b_0$

הפלט $b_j = 1$ כאשר j הוא המספר העשרוני המיוצג ע"י $a_1a_2a_3$ של $a_1a_2a_3$ ואחרת $b_j = 0$.

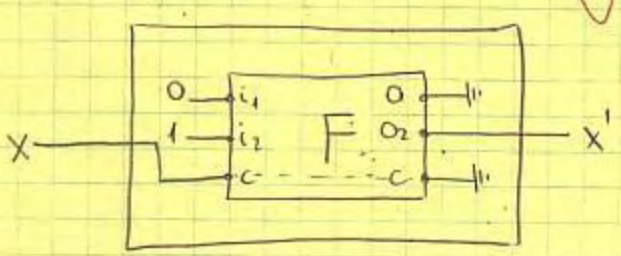
a_1	a_2	a_3	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7
0	0	0	1							
0	0	1		1						
0	1	0			1					
0	1	1				1				
1	0	0					1			
1	0	1						1		
1	1	0							1	
1	1	1								1

שאלה 4

AND מימוש:

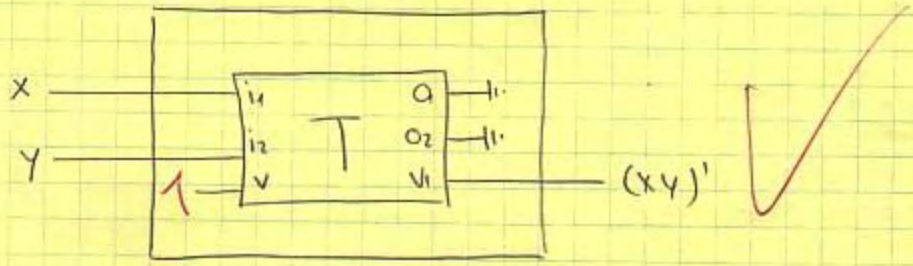


NOT מימוש:



שאלה 5

הוכיחו ש {NAND} היא אם שמה
(אחת שער NAND מצמצות שער טופלי והקבוע
1 ואחרון נקרא ש {Toffoli} 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) = (0,1,4,5)
2. F(A,B,C,D) = (0,1,2,3,4,10,11)
3. F(w,x,y,z) = (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

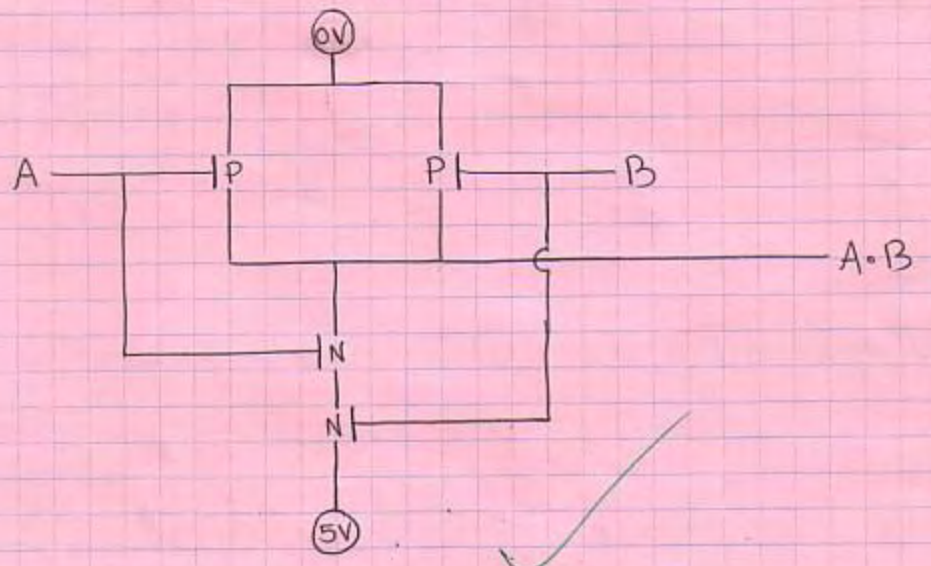
ממשו את הפונקציות הבאות תוך שימוש בשערים האדישים d (כלומר כאשר d=1 הצירוף אדיש).
הנח שנתונות שתי הצורות של כל משתנה (הרגיל והמשלים).
אנא פרטו את הדרך וכתבו את הביטוי בנוסף למעגל.

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 ולוגיקה ספרתית - תרגול 4

שאלה 1

שער AND מאתגי, n, p:



אם A=0V או B=0V אז מתג ה- p יתחבר ונקבל 0V מיציאה. אם A=5V ו B=5V אז יתחבר שני מתגי ה- N ונקבל 5V מיציאה

שאלה 2

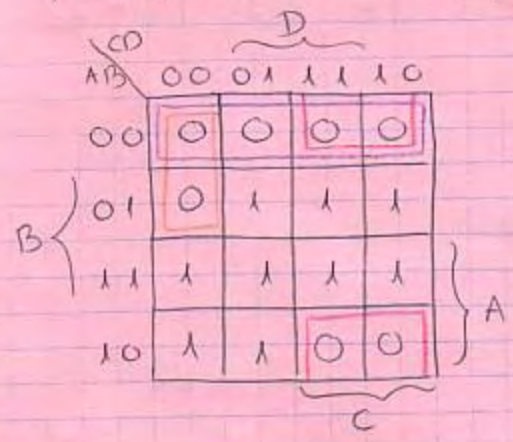
① נציג מפת קרנו של $f(x,y,z) = \Pi(0,1,4,5) = \Sigma(2,3,6,7)$

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

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

② נמצא את הדרך של הפונקציה

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

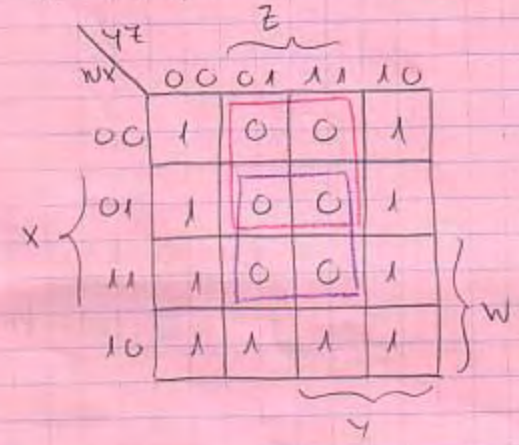


$f' = A'B' + A'C'D' + B'C$

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

③ נמצא את הדרך של הפונקציה

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



$f' = zx + z'w$

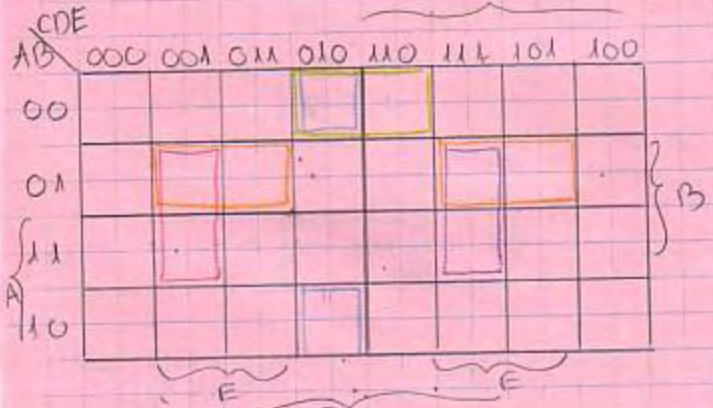
$\Rightarrow f = (w+z')(x'+z')$ ✓

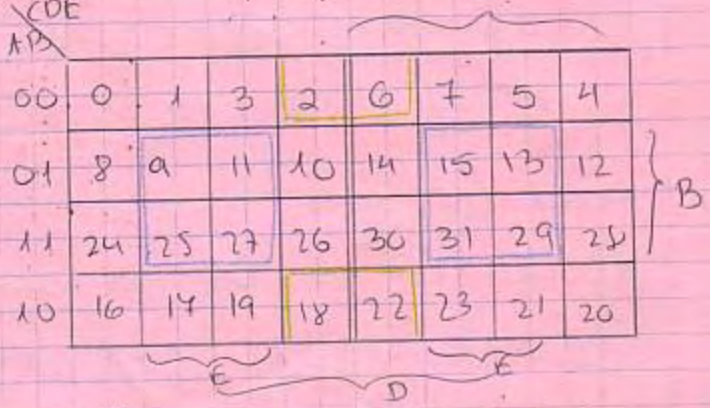
שאלה 3

נמצא את הדרך של הפונקציות:

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

$BE + B'DE'$





המינטרמים בשני השלישים הם אלה שלא משותפים לשתי הפונקציות:

$\Sigma(27, 22, 29)$ ✓

שאלה 4

① בדיק כיתוב ראשון פונק' מלכה 2 שער NOR בפל' 3 כניסות הינו:

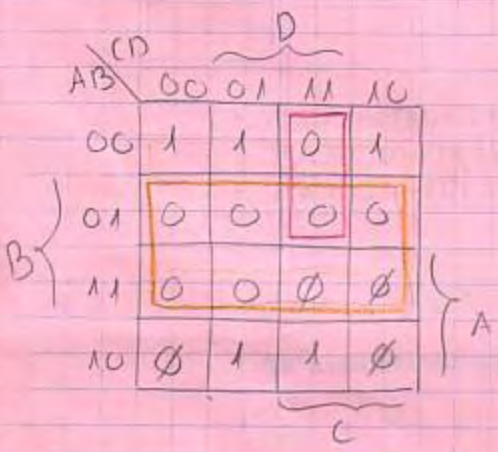
משר $F = NOR(x, y, NOR(z, w, v))$

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

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

נכדה להמיר את F ל: סכום מכפלות משר כ-2 מהמרכלות אבר אורך (מלבן עם 8 ריבועים) ובאזור מרכזה 3 אינדיקס (מלבן עם 2 ריבועים).

נבנה מפת קרנו של הפונקציה:



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

$\Downarrow$

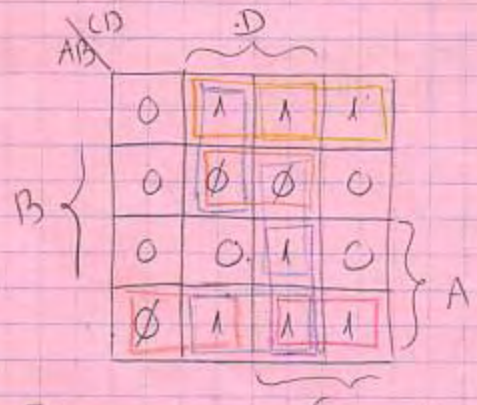
$F = (B + B + A'CD)' =$

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

$= NOR(B, B, NOR(A, C', D'))$ ✓

② נבנה מפת קרנו של הפונקציה $F = B'D' + B'C + ABCD$

עם הדיסיוזיות האדישות $d = A'BD + AB'C'D'$



משמע נוכל את הפונקציה ע"י שער NAND בעל 3 כניסות ש"ח לפחות 7 אלמנטים עם שיהי 2.

מדעים (3-)

(1-) אובדן לא נכון כיוון שהשתמש בשער יותר

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

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

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

מיתוג ומערכות ספרתיות תרגיל 5 (תרגיל גדול).

משחק החיים מוגדר כך:

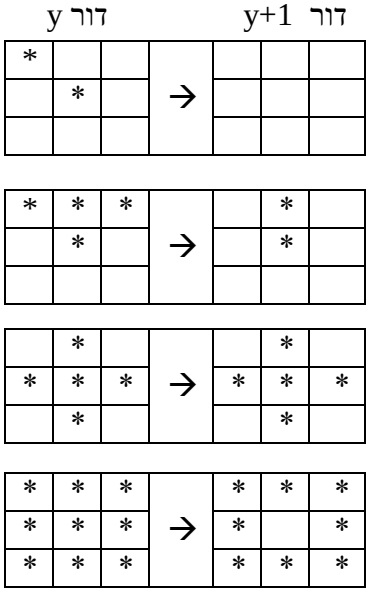
משחק החיים

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



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

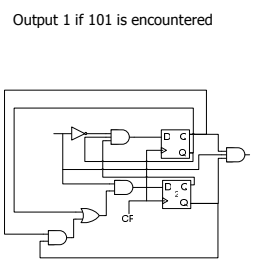
שימו לב שבכל דופק שעון נקראת סיבית אחת והמספר הוא אוסף כל הסיביות אשר נקראו עד כה.

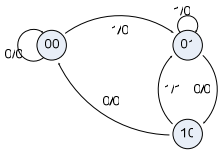
הדרכה:

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

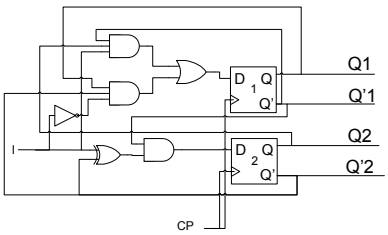
Output 1 if 101 is encountered

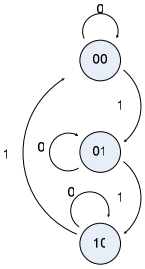




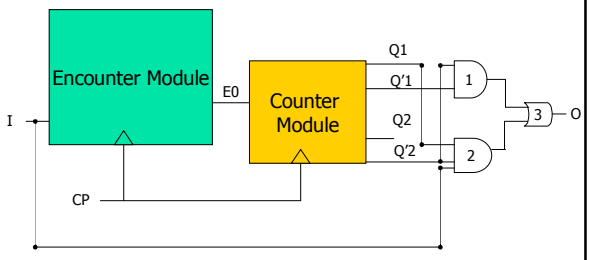
Solution to Exercise 6 Q2 Counter Module

A mod 3 counter for the number of appearances

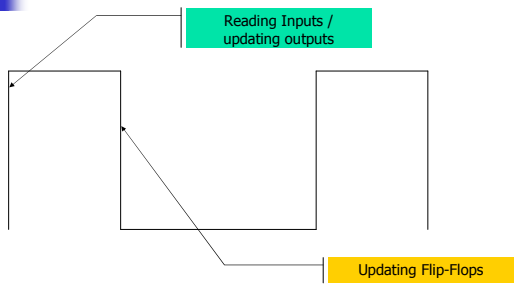




The Complete Circuit Q2



How does the solution work?



How does the solution work?

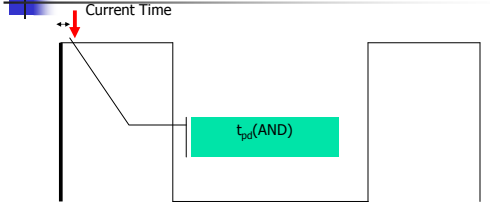
Suppose 2 101 were encountered. The Counter module is In state 10. Suppose that now 10 were encountered, The encounter module is in state 10

How does the solution work?

Current Time

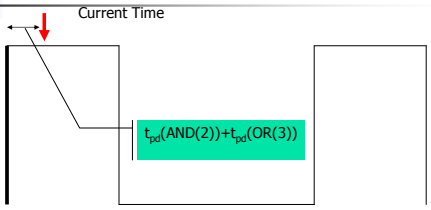
I =1	Counter Module = 10	Encounter Module = 10	E0=0
And(1) = 0	And(2) = 0	O = 0	

How does the solution work?



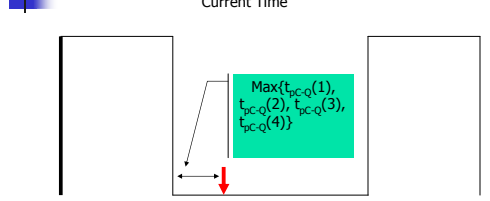
I =1	Counter Module = 10	Encounter Module = 10	E0=1
And(1) = 0	And(2) = 0	O = 0	

How does the solution work?



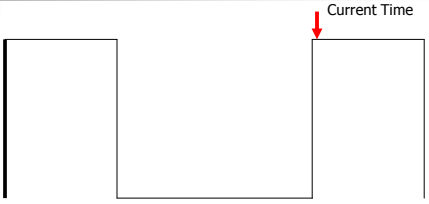
I =1	Counter Module = 10	Encounter Module = 10	E0=1
And(1) = 0	And(2) = 1	O = 1	

How does the solution work?



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

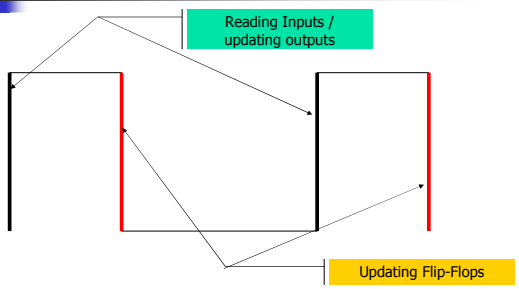
How does the solution work?



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

Solution to Question 2 Exercise 7

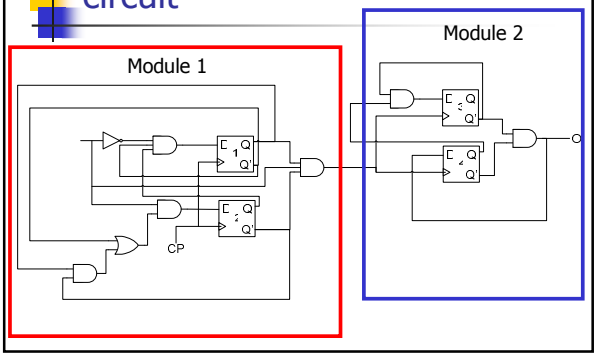
Events in synchronized system



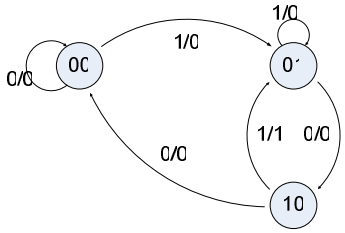
The requirements:

- 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

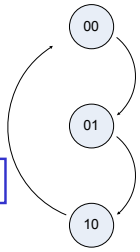


1. Analyze the state machine of module 1



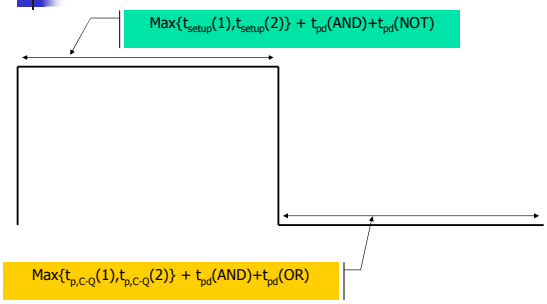
1. Analyze the state machine of module 2

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

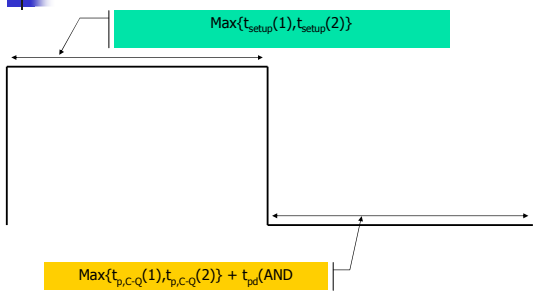


The update of the FF is when the clock
Of the FF goes from 1 to 0.

Set the timing of module 1 when it is independent



Set the timing of module 2 when it is independent



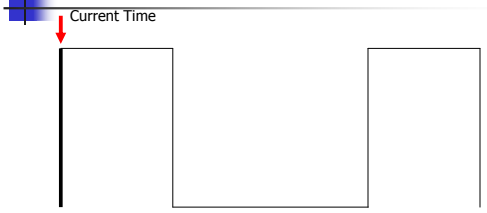
The "0" of a circuit:

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

Examine the "Clock" of module 2.

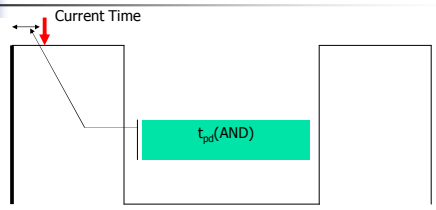
1. Suppose that module 1 is in state 10.
2. Suppose that now the input is 1

The clock of module 2



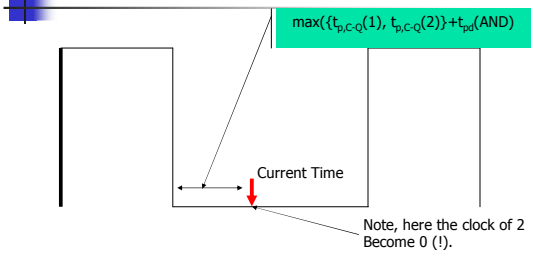
I =1	Encounter Module = 10	Module 1 output = 0

The clock of module 2



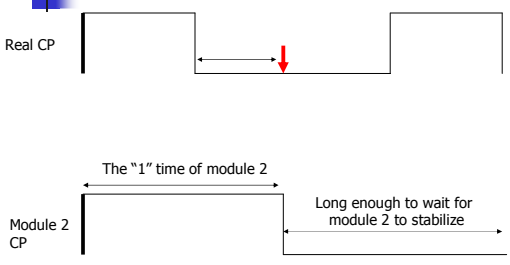
I =1	Encounter Module = 10	Module 1 output = 1 The Clock of module 2

The clock of module 2

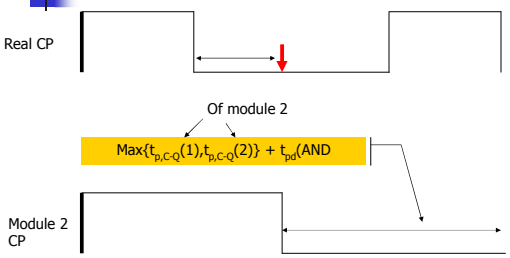


I =1	Encounter Module = 01	Module 1 output = 0 The Clock of module 2

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

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

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({$t_{p,C-Q}(1)$, $t_{p,C-Q}(2)$}+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:

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

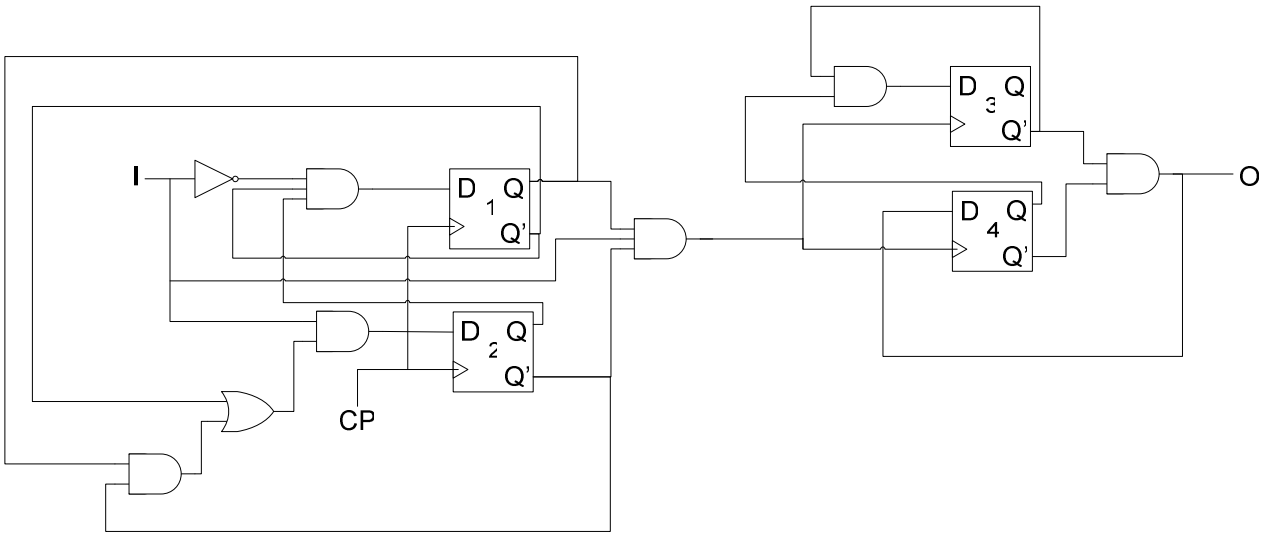
Of module 1 Of module 2

מיתוג ומערכות סיפרתיות תרגיל 7

אנא ציינו login על התרגיל.

במעגלים הבאים נכנסת **<u>מימין</u>** 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

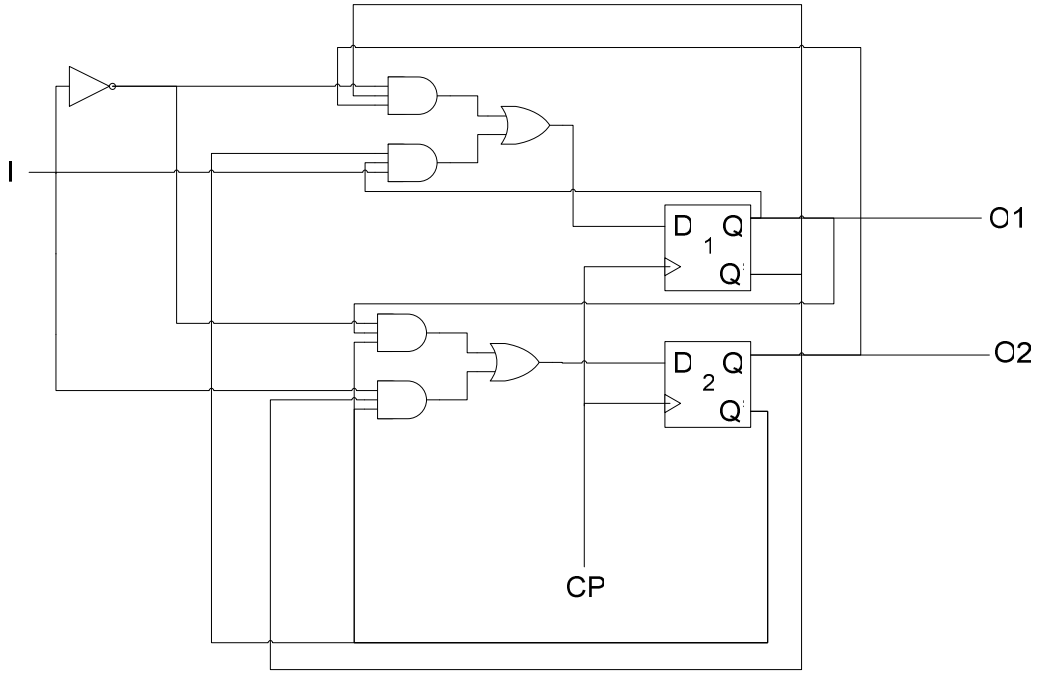
TSetup(FF)	0.05

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:

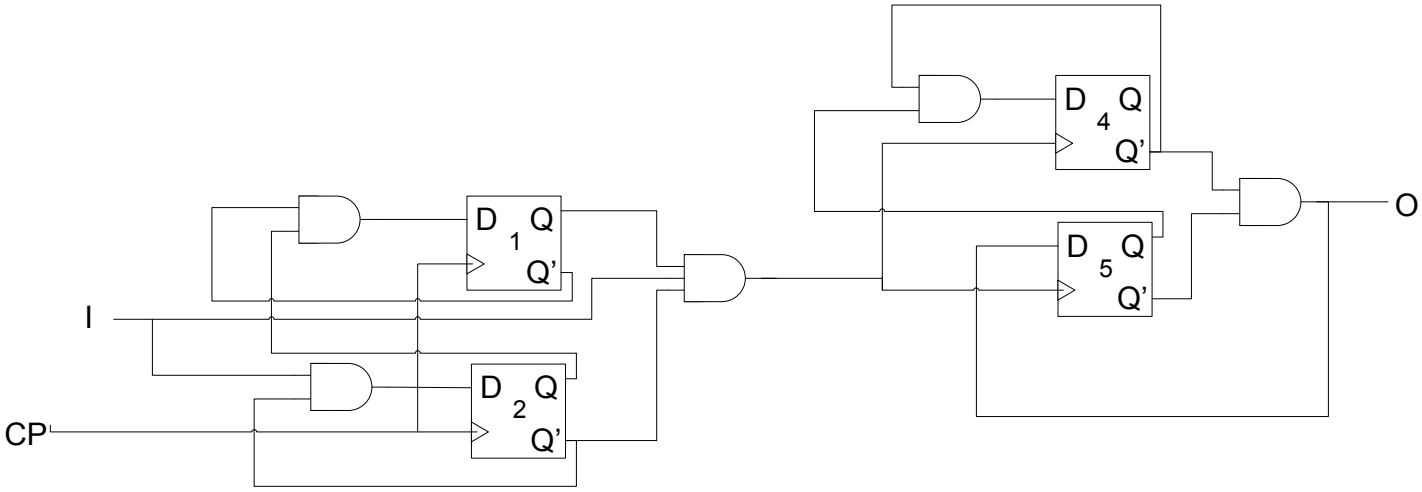
הפלט של המעגל (O_1O_0)	המספר הנקרא (עשרוני)	המחרוזת שנקרא	מספר הדפקים שעברו
00	0	0	0
01	1	01	1
10	2	010	2
10	5	0101	3

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:

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

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	

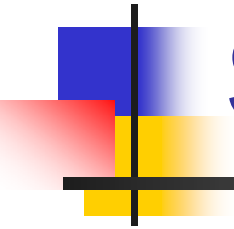
האלגוריתם בצעד 3 הינו דוגמא בלבד ולכן חסרים בו כמה שלבים.

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

קוד ASCII לאותיות הרלבנטיות לתרגיל

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

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

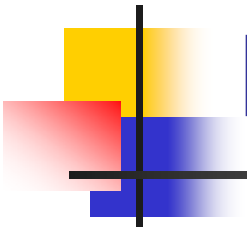
   (I==4):
   1. Reg[Dest] ← Op(Dig1,Dig2)
   2. I = 0

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

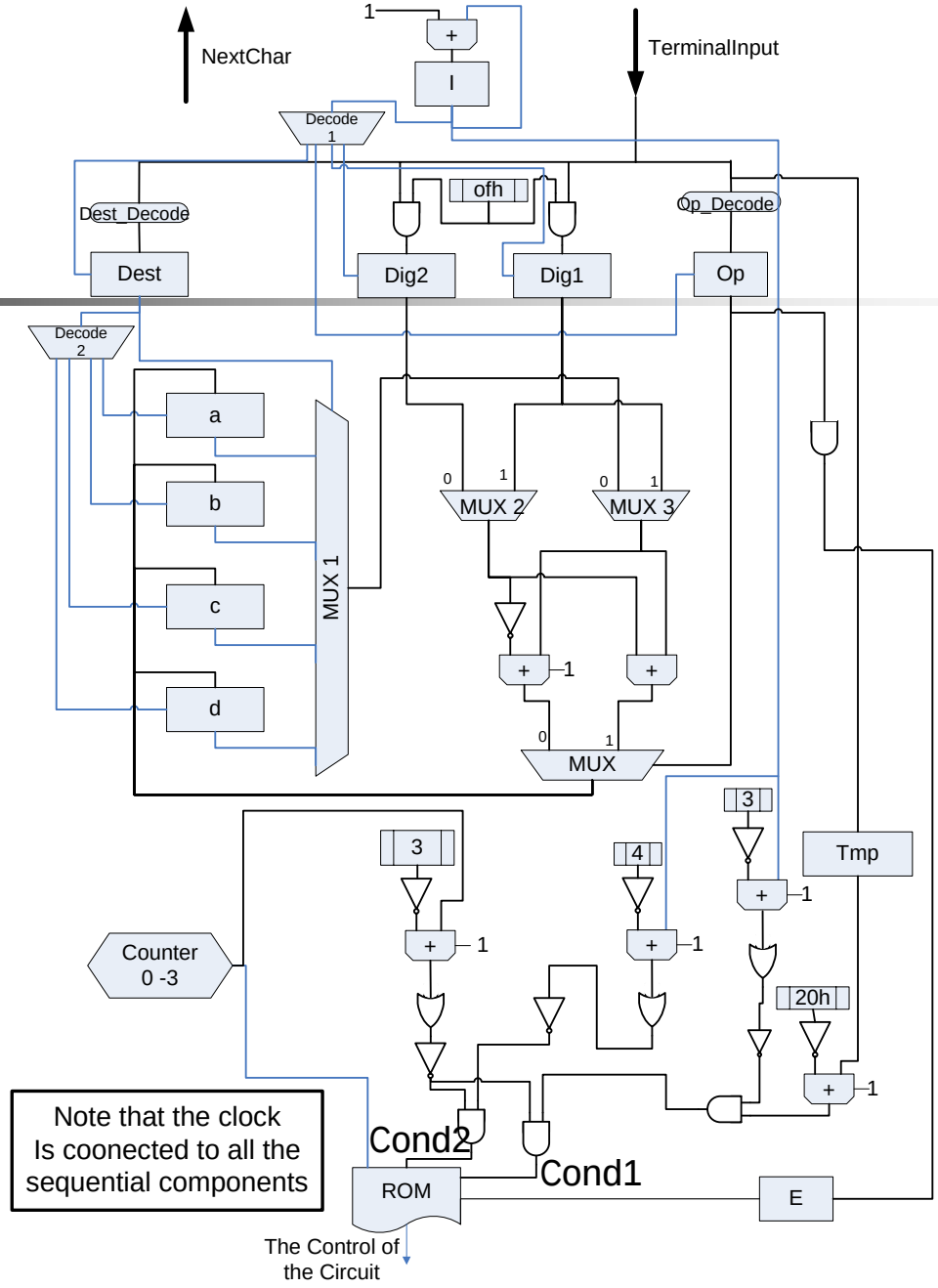
   (I==4):
   1. Reg[Dest] ← Op(Dig1,Dig2)
   2. I = 0

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


NextChar
TerminalInput
1
+
I
Decode 1
ofh
Dest_Decode
Op_Decode
Dest
Dig2
Dig1
Op
Decode 2
a
b
c
d
MUX 1
MUX 2
MUX 3
MUX
0
1
3
4
Tmp
Counter 0 -3
20h
Cond2
Cond1
ROM
E
Note that the clock Is coonected to all the sequential components
The Control of the Circuit

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