

### LAB3 - Digital System Design

#### הסבר:

הרכיב מקבל 3 משתנים כאשר

- $clk$  הוא השעון.
- $reset$  כאשר הוא 1 הוא מאפס את המכונה.
- $Data_{in}$  הוא רצף מספרים בינאריים.

הרכיב מחזיר משתנה אחד כאשר

- $Data_{out}$  הוא רצף מספרים בינאריים.

הרכיב מבצע פעולות חישבוניות לפי  $Data_{in}$ , המספר הראשון שלא שווה אפס הוא סוג הפעולה החישובית, נשמור המספר הזה ברגיסטר  $opc$ . בוחרים הפעולה לפי הטבלה:

Decimal value	ALUFN	Operation
0	000	ALUout=A
1	001	ALUout=A+B
2	010	ALUout=A-B
3	011	ALUout=A OR B
4	100	ALUout=A AND B
5	101	ALUout=A XOR B

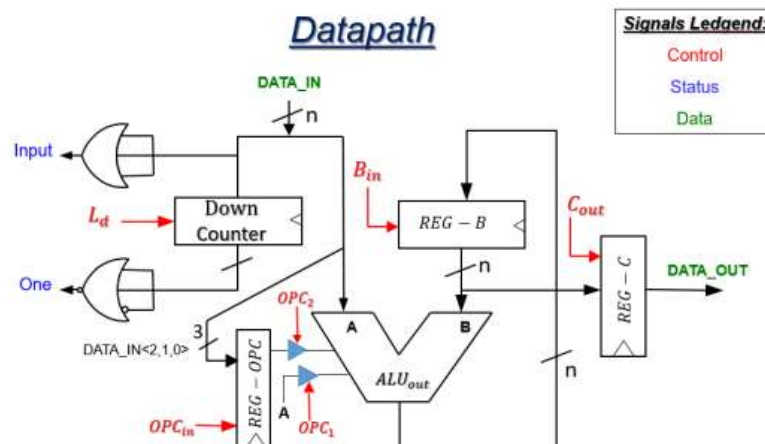
המספר שנכנס אחר כך הוא מספר המספרים הצפויים לקבל מ  $Data_{in}$  שצריך לבצע עליהם הפעולות, נשמור המספר הזה ברגיסטר  $Down - Counter$ .

לא נבצע פעולות על המספר הראשון רק נעביר אותו לרגיסטר B.

עבור כל המספרים האחרים, נבצע הפעולה בין המספר ו-B (ברגיסטר B יהיה התוצאה של הפעולה בין שני המספרים הקודמים) ונשמור התוצאה ברגיסטר B.

אם נשאר מספר אחד שצפוי לקבל התוצאה שברגיסטר B תועבר לרגיסטר C ומכאן ל-  $Data_{out}$ .

מבנה ה-  $Datapath$ : (הסבר על המצבים למטה)



כדי לוודא שהמספרים הנכנסים נשמרים במקום הנכון השתמשנו ברכיב *Control*.

רכיב *Control* הוא מכונת moore שמקבל שני משתנים מ-*Datapath*:

- *Input* אם יש אות כניסה לא שווה 0 אז הוא 1 אחרת 0.
  - *One* הוא 1 אם נשאר מספר אחד שצפוי לקבל אחרת 0 (ערך  $downcounter = 1$ ).
- הרכיב מחזיר 6 משתנים ששולטים על פעולת ה-*Datapath*:
- *Cout* אם הוא 1 אז התוצאה שברגיסטר B תועבר לרגיסטר C ומכאן ל-*Dataout*.
  - *Bin* אם הוא 1 אז התוצאה של הפעולת החישובית תועבר לרגיסטר B.
  - *Ld* אם הוא 1 אז אות הכניסה *Datain* יועבר לרגיסטר *Down – Counter*.
  - *OPC1* אם הוא 1 אז אות הכניסה *Datain* יועבר לרכיב שמבצע הפעולה החישובית, אחרת יועבר  $0^n$ .
  - *OPC2* אם הוא 1 אז נעביר תוכן *reg – opc* ל-*ALU* (מכיל מספר הפעולה של ה-*ALU*) אחרת נעביר  $000 (A = B)$ .
  - *Opcin* אם הוא 1 אז אות הכניסה  $Datain(2 - 0)$  יועבר לרגיסטר *opc*.

הרכיב מחזיר 6 משתנים לפי המצב הנוכחי בו, ולפי שתי הכניסות.

ברכיב *Control* יש 4 מצבים:

- מצב 0: ה-*Datapath* מחכה לקבלת מספר הפעולה החישובית מ-*Datain*.
- מצב-1: ה-*Datapath* מחכה לקבלת מספר המייצג כמות המספרים שצריך לבצע עליהם הפעולה.
- מצב-2: ה-*Datapath* מחכה לקבלת המספר הראשון כדי לשמור אותו ברגיסטר B.
- מצב-3: ה-*Datapath* מחכה לקבלת המספר כדי לעשות פעולה חישובית בינו ובין הרגיסטר B.

טבלת המעבר של המצבים:

(הערך של X יכול להיות 1 או 0, זה אומר שערך המשתנה לא משנה)

Current State	Input	One	Next State	OPCin	OPC2	OPC1	Ld	Bin	Cout
State 0	0	X	State 0	1	0	0	0	0	0
State 0	1	X	State 1	1	0	0	0	0	0
State 1	X	X	State 2	0	0	0	1	0	0
State 2	X	0	State 3	0	0	1	0	1	0
State 2	X	1	State 0	0	0	1	0	1	1
State 3	X	0	State 3	0	1	1	0	1	0
State 3	X	1	State 0	0	1	1	0	1	1

## תוצאות:

יש 3 קבצי testbench :

*controlTB* בודק אם היציאה והמצבים ברכיב *Control* משתנים לפי טבלת המעברים.  
*datapathTB* בודק אם המספרים הנכנסים נשמרים במקומות הנכונים.  
*topTB* בודק כל הרכיבים, אות הכניסה הוא רצף מספרים עשרוניים הנקרא מקובץ inputFile.txt שנמצא בתיקייה העבודה, והתוצאות יהיו רצף מספרים עשרוניים שנשמרו לקובץ outputFile.txt שנמצא באותו תיקייה.

### *controlTB*

ערך של *datain* תמיד 1

כאשר  $OPCin = 1$ , ערך *datain* נשמר ב *op-reg* (אדום)

כאשר  $Ld = 1$ , ערך *datain* נשמר ב- *downcounter* (אדום)

כאשר  $opc1 = 1$ , ערך *datain* נשמר ב- *b-reg* (ירוק)

כאשר  $opc2 = 1$ , ערך סכום B ו *datain* נשמר ב- *b-reg* (כחול)

כאשר  $cout = 1$ , ערך *b-reg* עבר לרגיסטר *c-reg* ול- *Dataout* (צהוב)

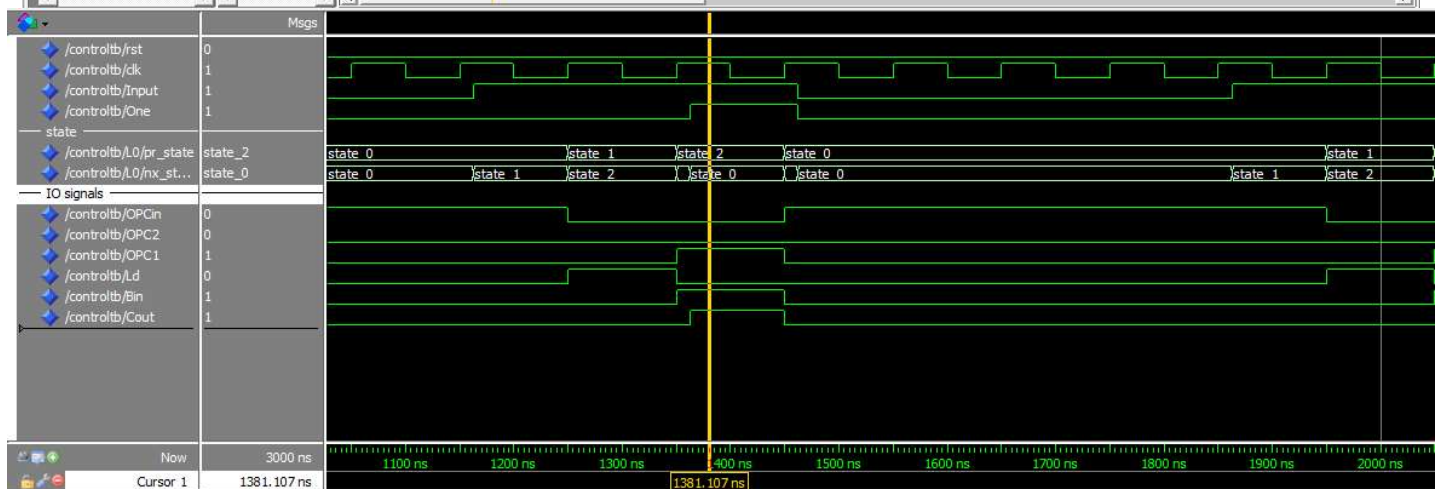
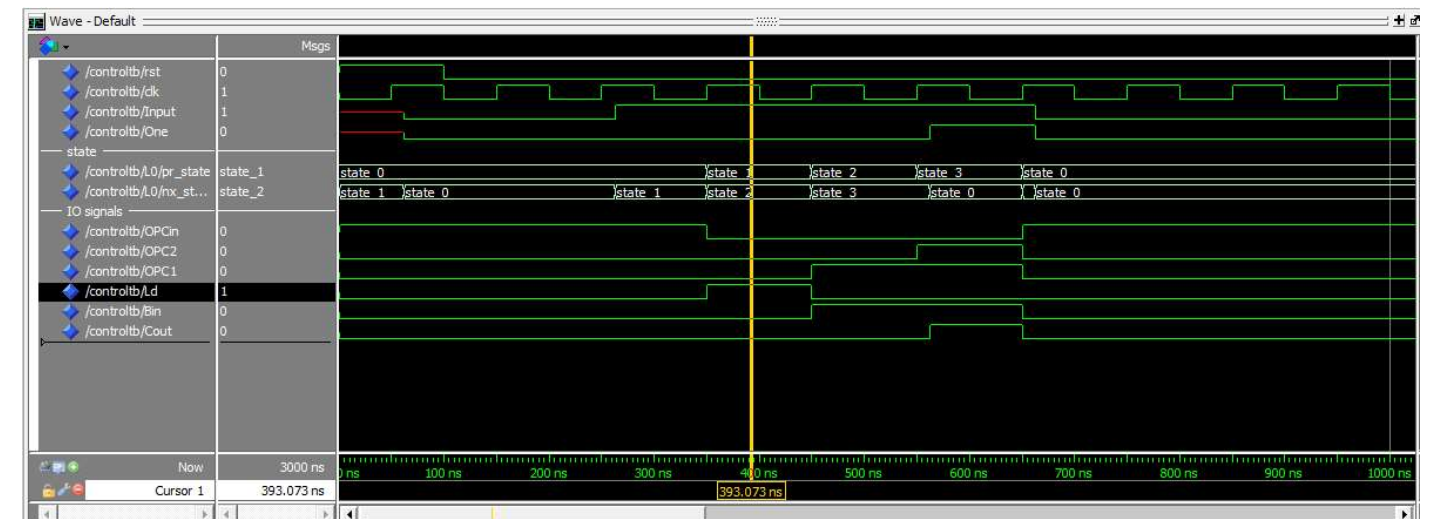
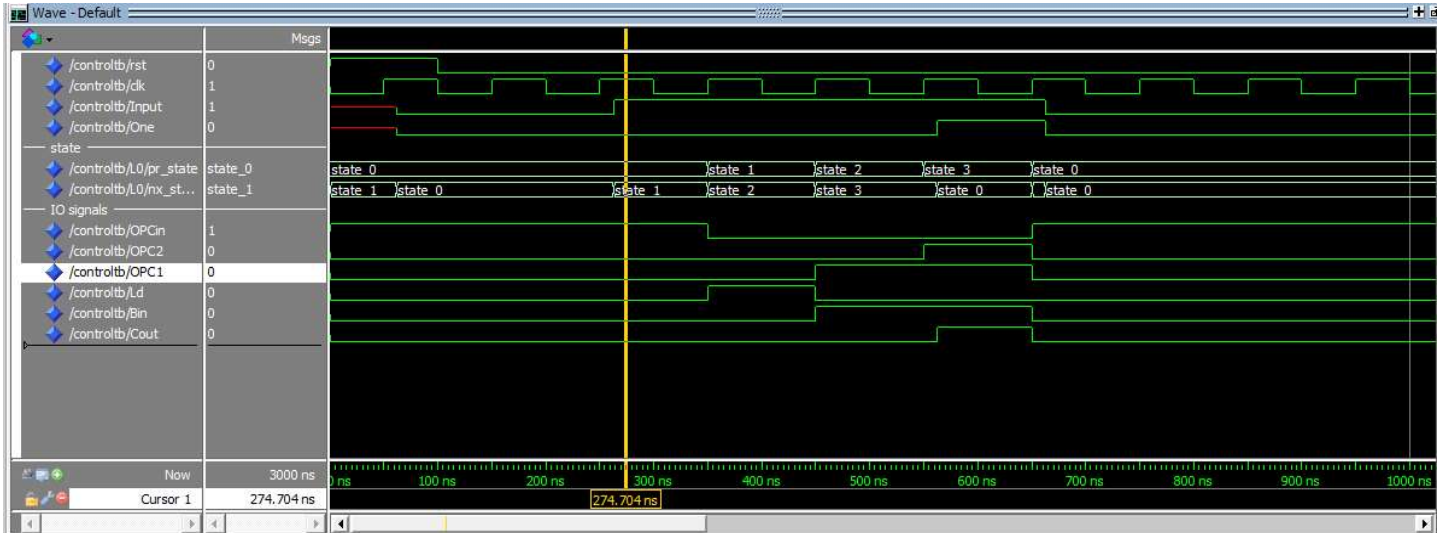


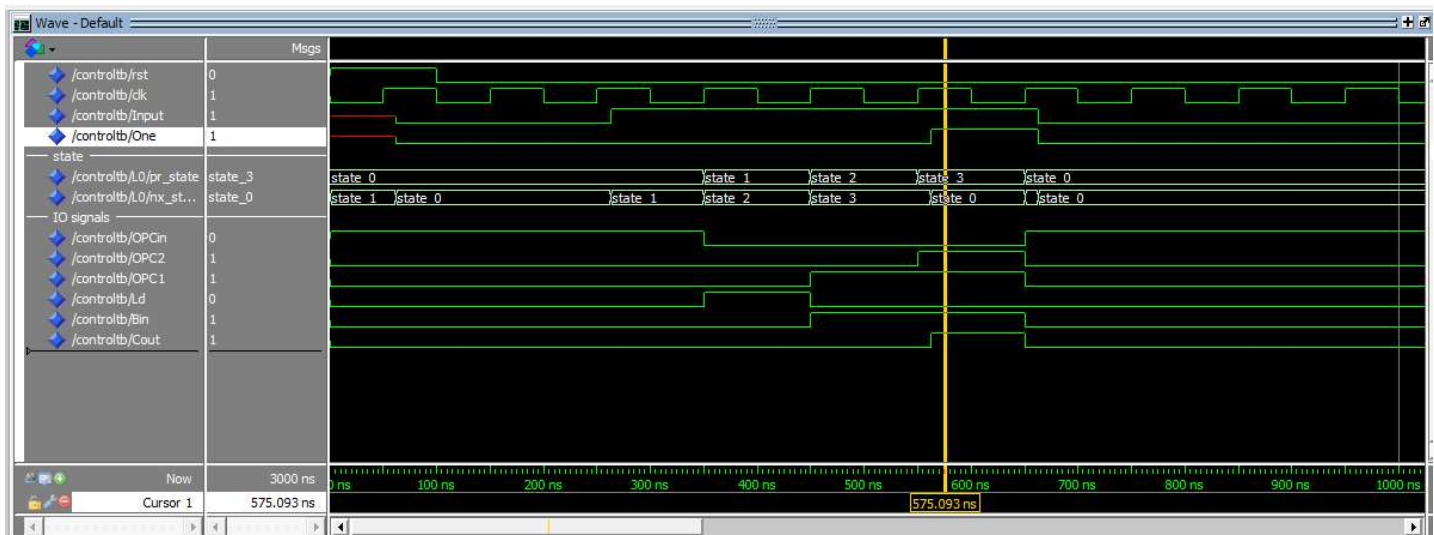
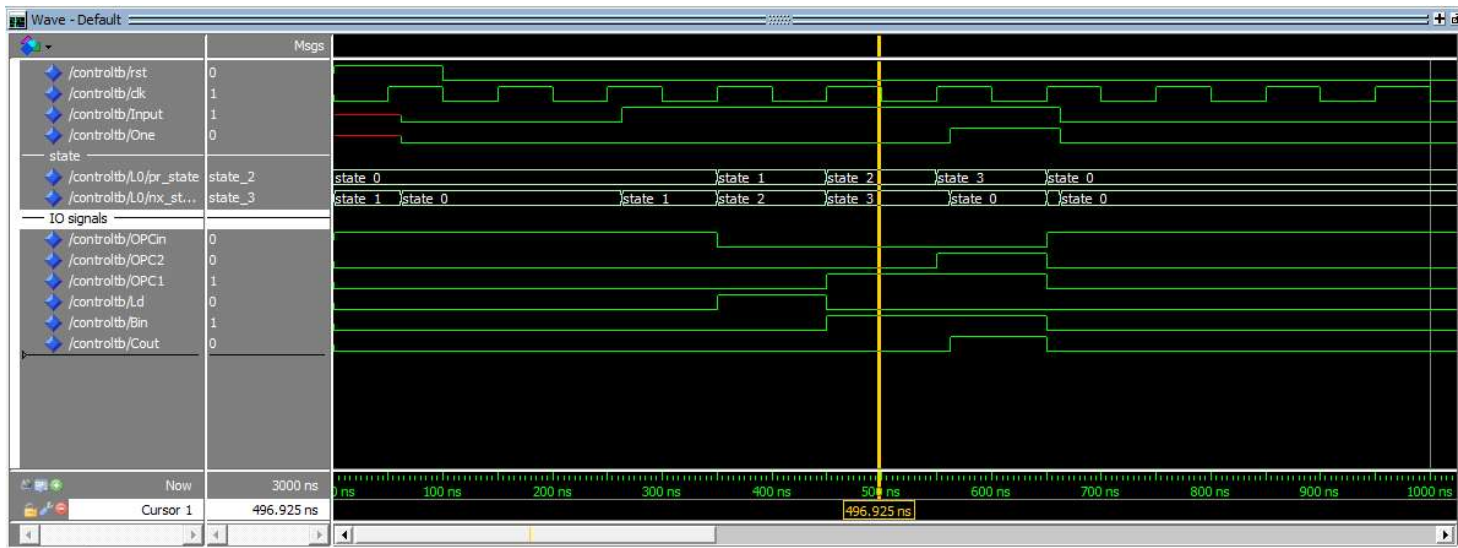
*controlTB*

התמונות לפי הסדר של טבלת המעברים

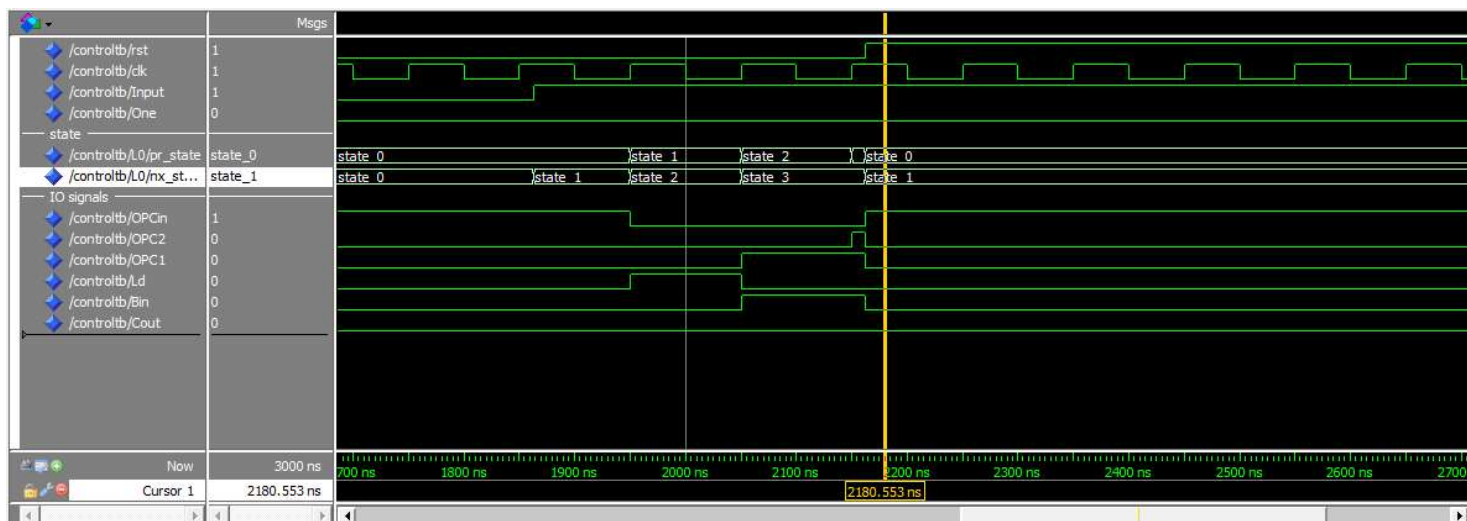
סדר התמונות:

*start : state<sub>0</sub> to state<sub>1</sub> to state<sub>2</sub> to state<sub>0</sub>, state<sub>2</sub> to state<sub>3</sub> to state<sub>0</sub>, state<sub>2</sub> to state<sub>0</sub> with reset*

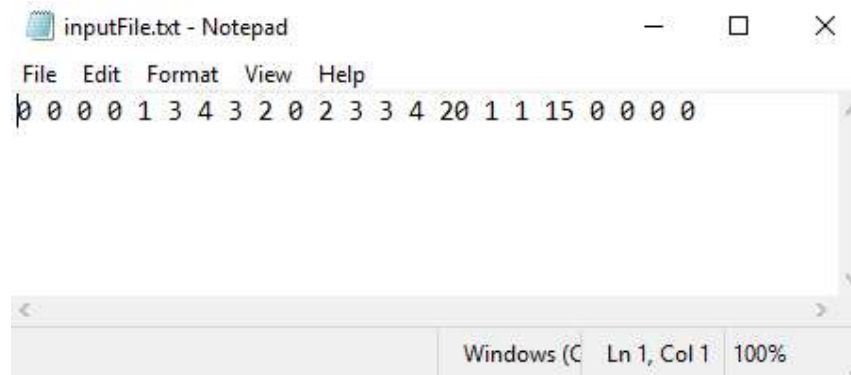




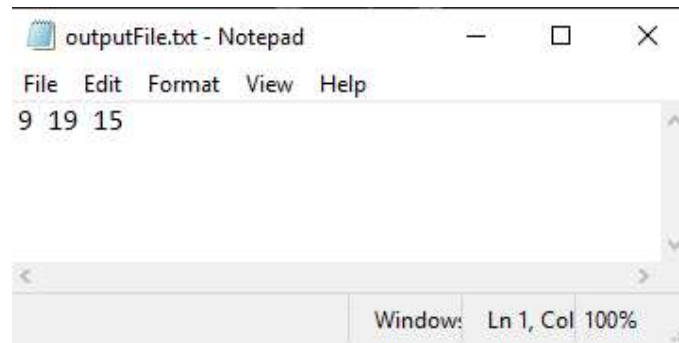
reset



*topTB*



```
inputFile.txt - Notepad
File Edit Format View Help
0 0 0 0 1 3 4 3 2 0 2 3 3 4 20 1 1 15 0 0 0 0
Windows (C) Ln 1, Col 1 100%
```



```
outputFile.txt - Notepad
File Edit Format View Help
9 19 15
Window: Ln 1, Col 100%
```

