

# דוח מכין מעבדה 1

מעבדת ארכיטקטורת מעבדים מתקדמת ומאיצי  
חומרה

Liav Ben Or 315909390  
Guy Cohen 207881004

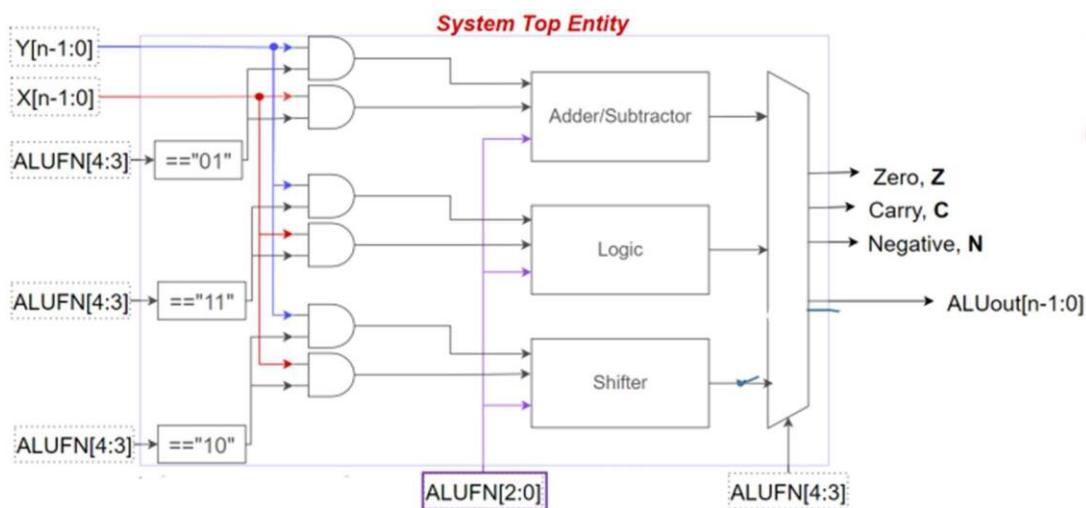
## תוכן עניינים

2-3.....	מבוא
3-4.....	רכיב AdderSub
3-4.....	הסבר תיאורטי
4.....	תוצאות הסימולציה
5-6.....	רכיב Shifter
5-6.....	הסבר תיאורטי
6.....	תוצאות הסימולציה
7.....	רכיב Logic
7.....	הסבר תיאורטי
7.....	תוצאות סימולציה
8.....	רכיב Top
8.....	הסבר תיאורטי
8-9.....	תוצאות סימולציה

## מבוא

במעבדה זו נלמד יכולות בסיסיות בעולם החומרה המקבילית בשפת VHDL. במעבדה זו, נממש מערכת המכילה מספר רכיבים שונים, כאשר כל פעם מודול אחר מתבצע בנפרד לפי בחירת המשתמש.

המערכת שנממש נמצאת באיור הבא:



### חלקי המערכת:

- אות כניסה X.
- אות כניסה Y.
- קו בקרה ALUFN כאשר ביטים 3,4 קובעים את הרכיב הנבחר באופן הבא:
  - 01 הוא רכיב AdderSub.
  - 10 הוא רכיב Shifter.
  - 11 הוא רכיב Logic.

המודולים יכולים לבצע מספר פעולות שונות כתלות בשלושת הביטים הראשונים בקו הבקרה (ביטים 0, 1, 2).

## מוצאי המערכת:

Function Kind	Decimal value	ALUFN	Operation	Note
Arithmetic	8	01000	Res=Y+X	
	9	01001	Res=Y-X	Used also for compare operation
	10	01010	Res=neg(X)	
Shift	16	10000	Res=SHL Y,X(k-1 to 0)	Shift Left Y of q≠X(k-1..0) times Res=Y(n-1-q...0)#(q@0) When $k = \log_2 n$
	17	10001	Res=SHR Y,X(k-1 to 0)	Shift Right Y of q≠X(k-1..0) times Res=(q@0)#Y(n-1...q) When $k = \log_2 n$
Boolean	24	11000	Res=not(Y)	
	25	11001	Res=Y or X	
	26	11010	Res=Y and X	
	27	11011	Res=Y xor X	
	28	11100	Res=Y nor X	
	29	11101	Res=Y nand X	
	30	11111	Res=Y xnor X	

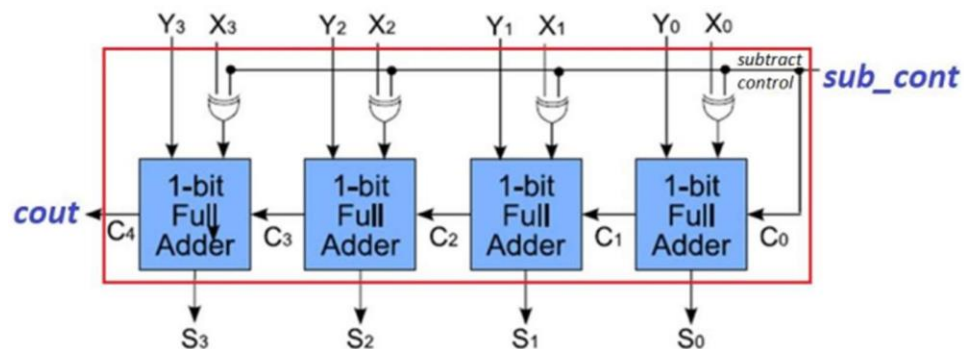
- דגלי בקרה C (Carry), N (Negative), Z (Zero), V (Overflow)
- תוצאת פעולת הרכיב יוצאת בALUOUT

## מודול AdderSub

### תיאור

רכיב זה יופעל כאשר שני הביטים האחרונים ב ALUFN יהיו "01". ויבצע את אחת משלושת הפעולות הבאות בהתאם לכניסת ALUFN באופן הבא:

- "000" - חיבור בין X ל Y
- "001" - חיסור Y בין X
- "010" - NEG(X)



SUB\_CONT אחראי על בחירת אחת מת פעולות הרכיב.

## מימוש VHDL

1. הגדרנו את FA כרכיב
2. יצרנו סיגנלים עבור משתני עזר, למשל וקטור אפסים ואחדות
3. מייצרים את SUB\_CONT על בסיס ביטי הבקרה
4. בוחרים את הוקטור X ע"פ ביטי הבקרה
5. בוחרים את הוקטור Y ע"פ ביטי הבקרה
6. סיגנל המכיל תוצאת XOR בין X לבין SUB\_CONT
7. חיבור FA ראשון
8. חיבור יתר FA
9. הוצאת COUT

## תוצאות הסימולציה



ps delta		/tb/Y	/tb/X	/tb/ALUout	/tb/ALUFN	/tb/cout
0	+5	11111111	11111111	000	11111110	1
50000	+5	11111110	11110101	000	11110011	1
100000	+4	11111101	11101011	001	00010010	1
150000	+4	11111100	11100001	001	00011011	1
200000	+6	11111011	11010111	010	00101001	0
250000	+4	11111010	11001101	010	00110011	0
300000	+4	11111001	11000011	011	00000000	0
350000	+1	11111000	10111001	011	00000000	0
400000	+1	11110111	10101111	100	00000000	0

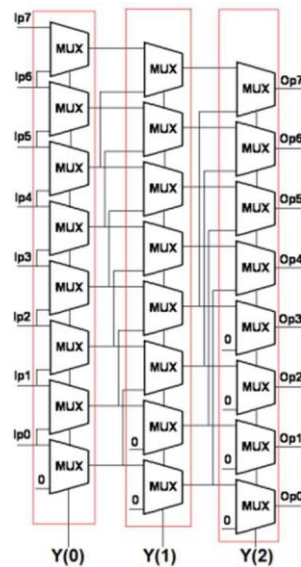
## מודול Shifter

### תיאור

רכיב זה מופעל בהתאם לכניבת ALUFN ומבצע פעולת הזזה המבוססת על BARREL SHIFTER כאשר ביטים 0,1,2 מחליטים אם ההזזה תבצע ימינה או שמאלה, "001", "000" בהתאמה. ניכנס לרכיב זה כאשר שני הביטים השמאליים שך ALUFN יהיו "10".

### מימוש

1. הגדרנו סיגנלים שימושיים כגון MAT השומרת את ההזזות במהלך תהליך זרימת המידע.
2. איתחלנו בשורה הראשונה של המטריצה את המספר המקורי Y לפני ההזזות.
3. בהתאם ל K הביטים הראשונים של X ביצענו הזזות Y ושמרנו בשורות המטריצה.
4. חישוב הCARRYOUT שצריך לצאת והוצאת השורה האחרונה מהמטריצה כתוצאה הסופית.



הערה: כאשר ההזזה מוגדרת ימינה נבצע את שלב 2 בצורה הפוכה ולבסוף לאחר סיום שלב 5 נהפוך את התוצאה הסופית על מנת לקבל הזזות כרצוי.

## תוצאות הסימולציה



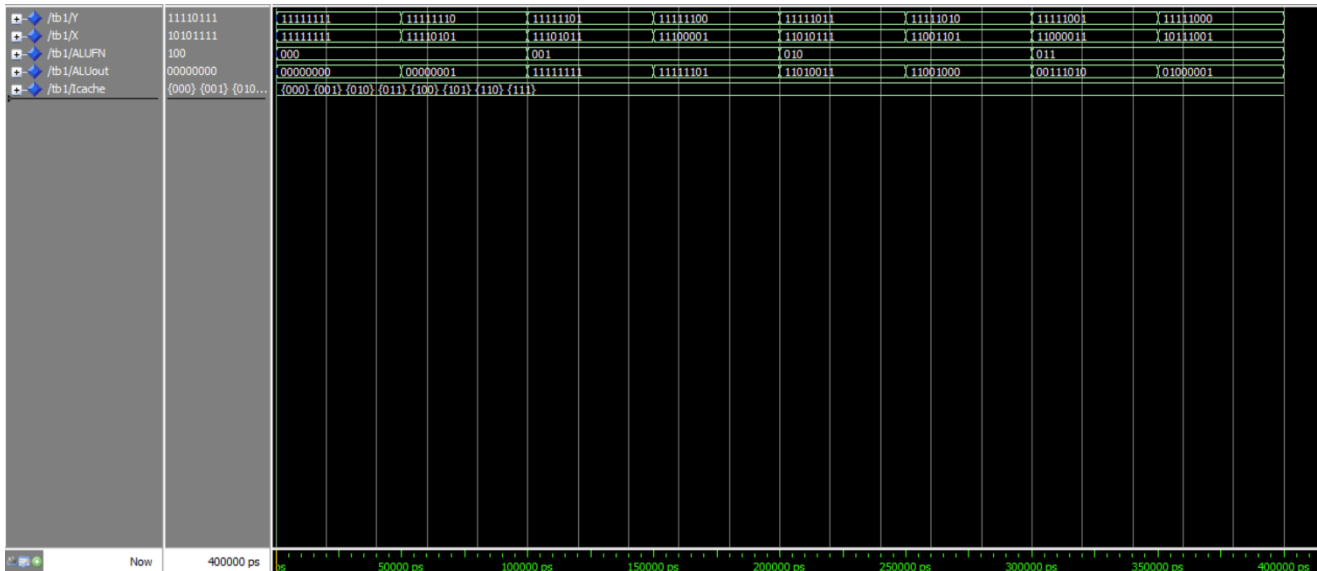
ps	delta	/tb2/Y	/tb2/X	/tb2/ALUout	/tb2/ALUFN	/tb2/cout
0	+6	11111111	11111111	000	10000000	1
50000	+6	11111110	11110101	000	11000000	1
100000	+6	11111101	11101011	001	00011111	1
150000	+6	11111100	11100001	001	01111110	0
200000	+6	11111011	11010111	010	00000000	0
250000	+1	11111010	11001101	010	00000000	0
300000	+1	11111001	11000011	011	00000000	0
350000	+1	11111000	10111001	011	00000000	0
400000	+1	11110111	10101111	100	00000000	0

## רכיב Logic תיאור

ניכנס לרכיב זה כאשר שני הביטים השמאליים שך ALUFN יהיו "11".  
רכיב זה מבצע פעולות לוגיות על וקטורים X ו Y ע"פ כניסת ALUFN:

- NOT(Y) -"000"
- X OR Y -"001"
- Y AND X -"010"
- Y XOR X -"011"
- Y NOR X -"100"
- Y NAND X -"101"
- Y XNOR X -"111"

## תוצאות סימולציה



ps	delta	/tb1/Y	/tb1/X	/tb1/ALUout	/tb1/ALUFN
0	+2	11111111	11111111	000	00000000
50000	+2	11111110	11110101	000	00000001
100000	+2	11111101	11101011	001	11111111
150000	+2	11111100	11100001	001	11111101
200000	+2	11111011	11010111	010	11010011
250000	+2	11111010	11001101	010	11001000
300000	+2	11111001	11000011	011	00111010
350000	+2	11111000	10111001	011	01000001
400000	+2	11110111	10101111	100	00000000

# מודול Top

## תיאור

רכיב זה הוא הרכיב המרכזי של המערכת המכיל את כל הרכיבים המתוארים מעלה כתת רכיבים המרכיבים אותו "השלם גדול מסך חלקיו!". הוא מקבל את הכניסות X, Y, ALUFN ויודע לנתב אותם לרכיב המתאים בהתאם. מימשנו כך שרכיבים שאינם בשימוש יהיו בנתק ורק הרכיב הדרוש יבצע את פעולתו, פלט המערכת ALUOUT יהיה בהתאם. כמו כן למערכת יש דגלי בקרה הנדלקים בהתאם לצורך ע"פ הגדרתם במשימה.

## אופן המימוש

1. הגדרת כל תתי רכיבי המערכת כCOMPONENTS.
2. יצירת סיגנלי עזר לחיבור בין רכיבי המערכת.
3. חילוף שלושת הביטים הראשונים מ ALUFN\_ in וניתובם לרכיב הרלוונטי.
4. חישוב דגלי הבקרה ע"פ הגדרתם במשימה.
5. הוצאת ALU\_OUT בהתאם לרכיב שהופעל.

## תוצאות הסימולציה

ps	delta	/tb3/Y	/tb3/X	/tb3/ALUout	/tb3/ALUFN	/tb3/Cflag	/tb3/Nflag	/tb3/Zflag	/tb3/Vflag
0	+8	11111111	11111111	01000	11111110	1	1	0	0
50000	+7	11111110	11110101	01000	11110011	1	1	0	0
100000	+9	11111101	11101011	01001	00010010	0	1	0	0
150000	+6	11111100	11100001	01001	00011011	0	1	0	0
200000	+8	11111011	11010111	01010	00101001	0	0	0	0
250000	+6	11111010	11001101	01010	00110011	0	0	0	0
300000	+10	11111001	11000011	01000	10111100	1	1	0	0
350000	+8	11111000	10111001	01000	10110001	1	1	0	0
400000	+9	11110111	10101111	01001	01001000	0	1	0	0
450000	+7	11110110	10100101	01001	01010001	0	1	0	0
500000	+9	11110101	10011011	01010	01100101	0	0	0	0
550000	+6	11110100	10010001	01010	01101111	0	0	0	0
600000	+10	11110011	10000111	01000	01111010	0	1	0	1
650000	+9	11110010	01111101	01000	01101111	0	1	0	0
700000	+13	11110001	01110011	01001	01111110	0	1	0	1
750000	+9	11110000	01101001	01001	10000111	1	1	0	0
800000	+6	11101111	01011111	10000	10000000	1	1	0	0
850000	+8	11101110	01010101	10000	11000000	1	1	0	0
900000	+8	11101101	01001011	10001	00011101	0	1	0	0
950000	+7	11101100	01000001	10001	01110110	0	0	0	0
1000000	+7	11101011	00110111	10010	00000000	0	0	1	0
1050000	+1	11101010	00101101	10010	00000000	0	0	1	0
1100000	+8	11101001	00100011	10000	01001000	0	1	0	0
1150000	+8	11101000	00011001	10000	11010000	1	1	0	0
1200000	+8	11100111	00001111	10001	00000001	0	1	0	0
1250000	+6	11100110	00000101	10001	00000111	0	0	0	0
1300000	+7	11100101	11111011	10010	00000000	0	0	1	0
1350000	+1	11100100	11110001	10010	00000000	0	0	1	0
1400000	+4	11100011	11100111	11001	11100111	1	0	0	0
1450000	+4	11100010	11011101	11001	11111111	1	0	0	0
1500000	+4	11100001	11010011	11010	11000001	1	0	0	0
1550000	+4	11100000	11001001	11010	11000000	1	0	0	0
1600000	+4	11011111	10111111	11101	01100000	0	0	0	0
1650000	+4	11011110	10110101	11101	01101011	0	0	0	0
1700000	+4	11011101	10101011	11111	10001001	1	0	0	0
1750000	+4	11011100	10100001	11111	10000010	1	0	0	0
1800000	+4	11011011	10010111	11011	01001100	0	0	0	0
1850000	+4	11011010	10001101	11011	01010111	0	0	0	0



