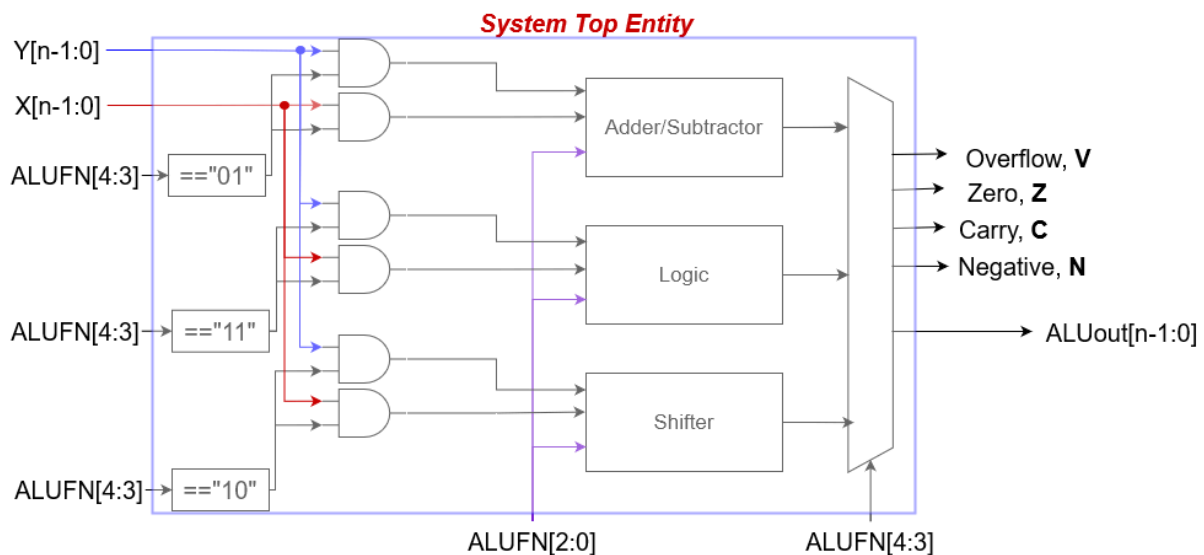


## Advanced CPU architecture and hardware accelerators lab

**Prepared by: Yarin Oziel- [REDACTED], Itay Kandil- [REDACTED]**

## הקדמה

במעבדה זו נלמד לכתוב קוד מקבילי בשפת תיאור חומרה VHDL. נממש ALU בעל כמה מצבים ע"פ הארכיטקטורה הבאה:



## כניסות ומוצאים של המערכת:

- אות כניסה X
- אות כניסה Y
- קו בקרה ALUFN כך שביטים 3,4 קובעים את המודול:  
 - 01 עבור AdderSub  
 - 10 עבור Logic  
 - 11 עבור Shifter  
 ביטים 0,1,2 אחראים לקנפג את המודולים במערכת.

- ALUout אות מוצא המערכת
- דגלי סטטוס: N(Negative), Z(Zero), C(Carry), V(Overflow).

ניתן לקבל מידע נוסף דרך ה ISA של המערכת המתוארת בטבלה הבאה:

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 $q \triangleq X(k-1 \dots 0)$ times Res=Y(n-1-q...0)#(q@0) <b>When <math>k = \log_2 n</math></b>
	17	10001	Res=SHR Y,X(k-1 to 0)	Shift Right Y of $q \triangleq X(k-1 \dots 0)$ times Res=(q@0)#Y(n-1...q) <b>When <math>k = \log_2 n</math></b>
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	

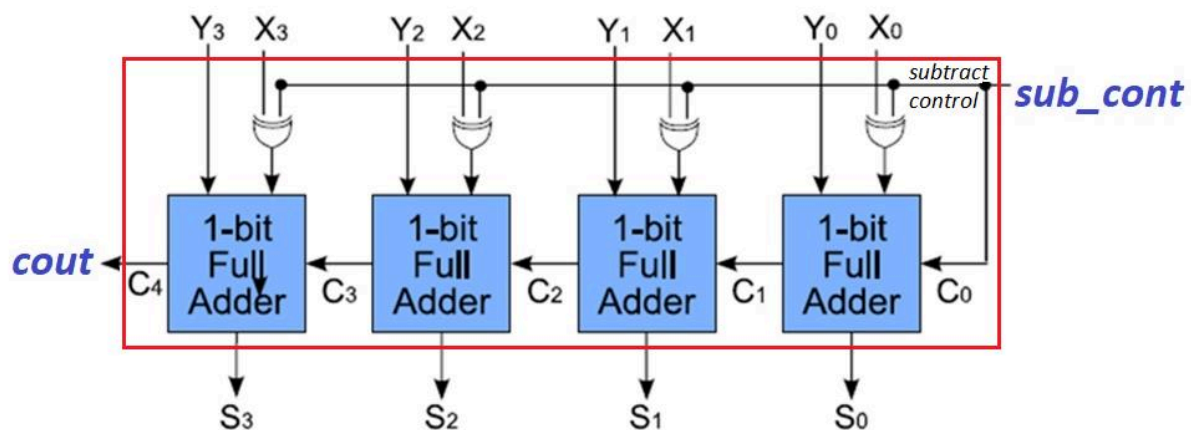
בעמודים הבאים נראה פירוט של כל מודול ביחד עם סימולציות ב MODEL-SIM

## מודול AdderSub

מודול זה מתוכנן כ ripple-adder על גבי FA, כך שהוא מממש 3 מצבי עבודה:

1. חיבור  $(Y+X)$ : מתקבל כאשר סיגנל  $sub\_cont='0'$  מוגדר על ביטים 0,1,2 ('000').
2. חיסור  $(Y-X)$ : מתקבל כאשר סיגנל  $sub\_cont='1'$  מוגדר על ביטים 0,1,2 ('001').
3. שלילי  $(-X)$ : מתקבל כאשר סיגנל  $sub\_cont='1'$  מוגדר על ביטים 0,1,2 ('010').

המודול בנוי תוך שימוש בFA לפי הארכיטקטורה הבאה:



כך שבמוצא נקבל את התוצאה ואת הדגלים התואמים לסטטוס הפעולה.

## סימולציה

נבדוק את תקינות המודול לפי כמה מקרי קצה, הכנסנו בדיקה בכל מקרה כך שבמידה ולא תהיה תואמת לפונקציה הבולאנית המתארת את המעגל נקבל שגיאה בסימולציה.  
בדקנו את מקרי הקצה: חיבור, חיבור עם carry, חיסור ו negative.

Waveform 80ns (20ns per test, Hexadecimal Radix):

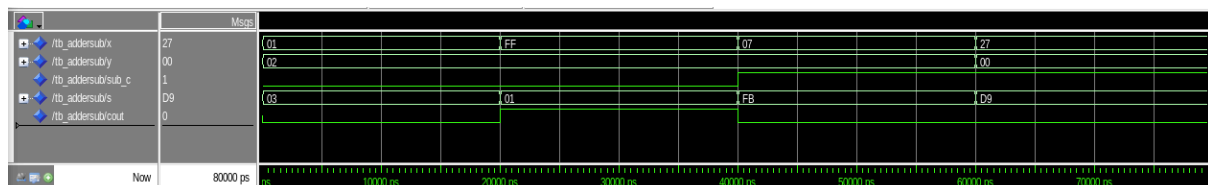


Table:

ps	delta	/tb_addersub/x	/tb_addersub/y	/tb_addersub/sub_c	/tb_addersub/cout
0	+3	00000001	00000010	0	00000011
20000	+10	11111111	00000010	0	00000001
40000	+9	00000111	00000010	1	11111011
60000	+3	00100111	00000000	1	11011001

קיבלנו תוצאות תקינות וללא שגיאות.

## מודול Logic

מודול זה מתוכנן באופן דומה ל-MUX כך שבהינתן סיגנל ALUFN בביטים 0,1,2 הוא בורר מבין 7 מצבים ומבצע פעולות על 1 או 2 אופרנדים.

המצבים התואמים ל-ALUFN[2:0] עבור כניסות X ו-Y:

Not(Y): 000  
Y OR X: 001  
Y AND X: 010  
Y XOR X: 011  
Y NOR X: 100  
Y NAND X: 101  
Y XNOR X: 111

## סימולציה

נבדוק את תקינות המודול לפי כמה מקרי קצה, הכנסנו בדיקה בכל מקרה כך שבמידה ולא תהיה תואמת לפונקציה הבולאנית המתארת את המעגל נקבל שגיאה בסימולציה.  
בדקנו את מקרי הקצה: .NOT, OR, AND, XOR, NOR, NAND, XNOR.

Waveform 160ns (20ns per test, Hexadecimal Radix):

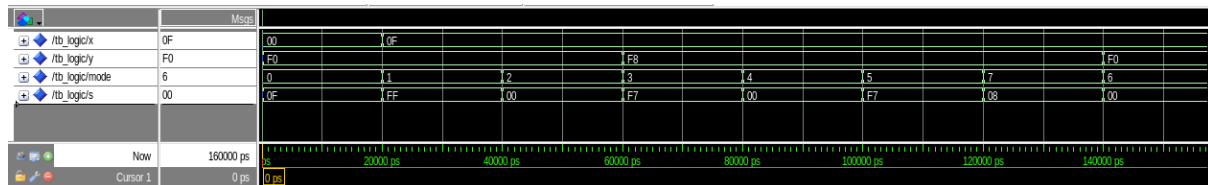


Table:

ps delta		/tb_logic/x	/tb_logic/y	/tb_logic/s	/tb_logic/mode
0	+2	00000000	11110000	000	00001111
20000	+2	00001111	11110000	001	11111111
40000	+2	00001111	11110000	010	00000000
60000	+2	00001111	11111000	011	11110111
80000	+2	00001111	11111000	100	00000000
100000	+2	00001111	11111000	101	11110111
120000	+2	00001111	11111000	111	00001000
140000	+2	00001111	11110000	110	00000000

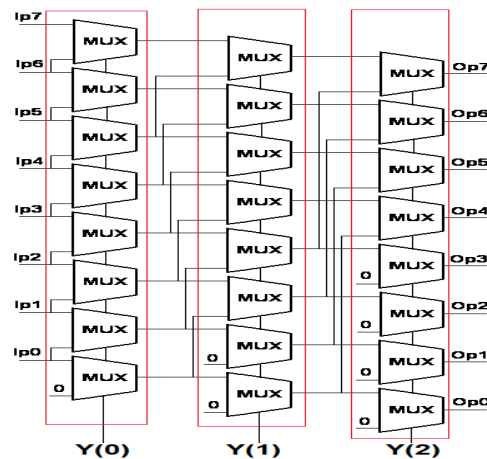
קיבלנו תוצאות תקינות וללא שגיאות.

## מודול Shifter

מודול זה מבצע הזזות לכניסה Y לפי X פעמים מבוסס על n-bit barrel-shifter. סיגנל בקרה ALUFN קובע בביטים 0,1,2 את סוג ההזזה (שמאלה: 000, ימינה: 001).

מימוש המודול מבוצע ע"י מטריצה דו-מימדית  $((n+1) \cdot \log(n))$ . תחילה אם ההזזה נקבע שמאלה נעביר את Y כפי שהוא לשלב הראשון במטריצה, אחרת נכניס את Y הפוך (INVERTED) כך שנוכל לממש הזזה ימינה בעזרת מנגנון הזזה שמאלה. לאחר מכן נבצע הזזה לפי 3 הביטים הראשונים של X ונכניס לכל שלב במטריצה. נשמור את ה carry ז"א הביט האחרון שנדחף מ Y ולבסוף נעביר את התוצאה למוצא המודול (במידה וההזזה ימינה נהפוך את הפלט כדי לקבל את התוצאה הרצויה).

המודול מחקה את הארכיטקטורה הבאה:



## סימולציה

נבדוק את תקינות המודול לפי כמה מקרי קצה, הכנסנו בדיקה בכל מקרה כך שבמידה ולא תהיה תואמת לפונקציה הבולאנית המתארת את המעגל נקבל שגיאה בסימולציה.

בדקנו את מקרי הקצה: ההזזה שמאלה, ההזזה שמאלה עם carry, ההזזה ימינה, ההזזה ימינה עם carry והזזות לא מוגדרות (הכנסת קלטים לא מוגדרים לDir).

Waveform 200ns (20ns per test, Hexadecimal Radix):

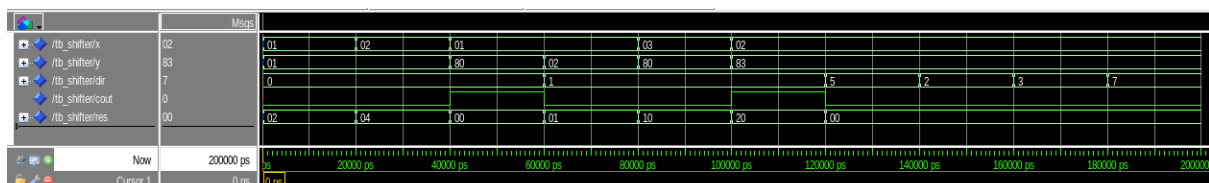


Table:

ps	delta	/tb_shifter/x	/tb_shifter/y	/tb_shifter/res	/tb_shifter/dir	/tb_shifter/cout
0	+6	00000001	00000001	000 0	00000010	
20000	+5	00000010	00000001	000 0	00000100	
40000	+6	00000001	10000000	000 1	00000000	
60000	+6	00000001	00000010	001 0	00000001	
80000	+6	00000011	10000000	001 0	00010000	
100000	+6	00000010	10000011	001 1	00100000	
120000	+2	00000010	10000011	101 0	00000000	
140000	+1	00000010	10000011	010 0	00000000	
160000	+1	00000010	10000011	011 0	00000000	
180000	+1	00000010	10000011	111 0	00000000	

קיבלנו תוצאות תקינות וללא שגיאות.

## מודול Top

מודול זה מתפקד בתור מעטפת וגישור עבור המודולים לעיל ונותנת ממשק לכניסות ומוצאי המערכת. כאשר נבחר במודול מסוים, השאר יקבלו כניסת 0 ומוצא 0 לפי בחירה של ALUFN והדגלים הרלוונטים ינווטו לכניסה המתאימה.

## סימולציה

נבדוק את תקינות המודול לפי כמה מקרי קצה, הכנסנו בדיקה בכל מקרה כך שבמידה ולא תהיה תואמת לפונקציה הבולאנית המתארת את המעגל נקבל שגיאה בסימולציה.

בדקנו את כל הפונקציונליות במערכת ז"א: כל המודולים, בדיקה תקינות של הדגלים והמוצאים.

Waveform 90ns (10ns per test, Hexadecimal Radix):

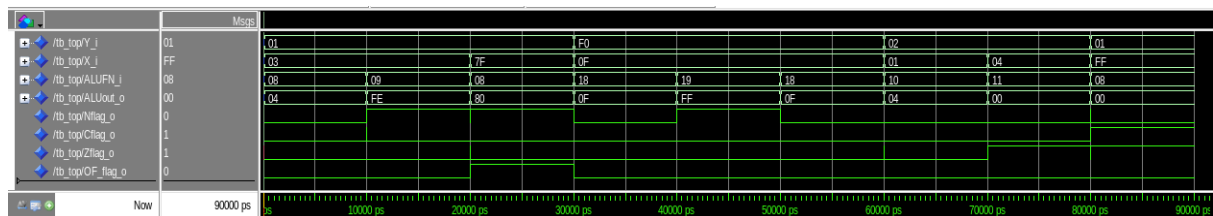


Table:

ps	delta	/tb_top/Y_i	/tb_top/X_i	/tb_top/ALUFN_i	/tb_top/ALUOut_o	/tb_top/Nflag_o	/tb_top/Cflag_o	/tb_top/Zflag_o	/tb_top/OF_flag_o
0	+8	00000001	00000011	01000	00000100	0 0 0	0		
10000	+12	00000001	00000011	01001	11111110	1 0 0	0		
20000	+13	00000001	01111111	01000	10000000	1 0 0	1		
30000	+5	11110000	00001111	11000	00001111	0 0 0	0		
40000	+4	11110000	00001111	11001	11111111	1 0 0	0		
50000	+4	11110000	00001111	11000	00001111	0 0 0	0		
60000	+9	00000010	00000001	10000	00000100	0 0 0	0		
70000	+8	00000010	00000100	10001	00000000	0 0 1	0		
80000	+13	00000001	11111111	01000	00000000	0 1 1	0		

כל הבדיקות יצאו תקינות, בנוסף מול בדיקת tb\_ref1, tb\_ref2 בבדיקת השוואה קיבלנו תוצאות תקינות.