

PREPARATION REPORT LAB1

ADVANCED CPU ARCHITECTURE AND HARDWARE ACCELERATORS LAB

361.1.4693

Nachman Mimoun 321730558

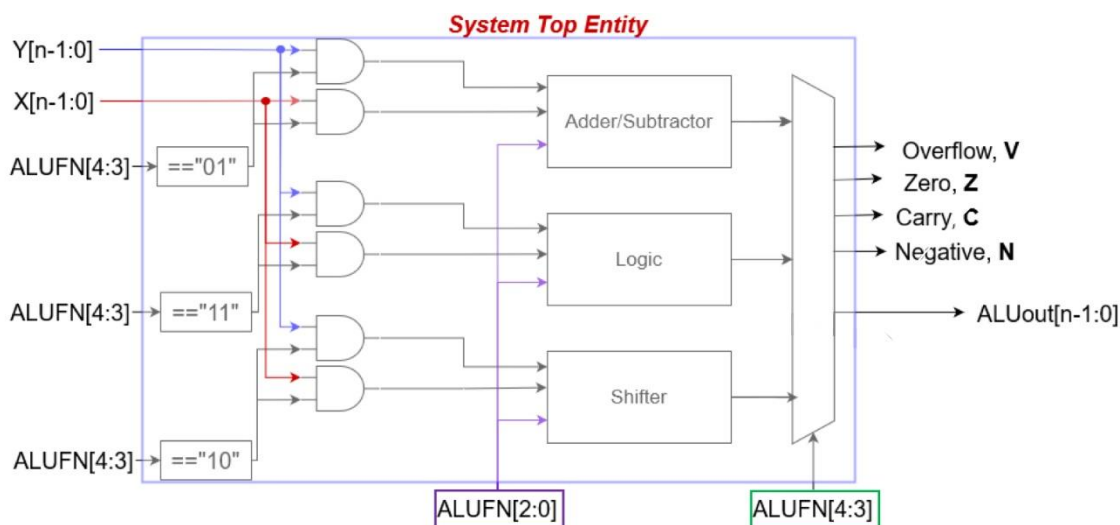
Danel Barsheshet 209471242

תוכן עניינים

2	מבוא
3	מודול AdderSub
3	הסבר תיאורטי
4	תוצאות סימולציה
4	מודול Shifter
4	הסבר תיאורטי
5	תוצאות סימולציה
5	מודול Logic
5	הסבר תיאורטי
5	תוצאות סימולציה
6	מודול Top
6	הסבר תיאורטי
7	תוצאות סימולציה וסיכום

מבוא

במעבדה זו נלמד על יכולות בסיסיות בעולם החומרה המקבילית תוך שימוש בשפת VHDL במסגרת הפרויקט נרצה ליישם מערכת המבצעת מספר מודולים שונים, כאשר כל מודול פועל בנפרד מהאחרים ומופעל באמצעות אות בחירה (chip select) הניתן על ידי המשתמש. להלן המערכת שנרצה לממש:



נסביר על כל חלק במערכת:
המערכת תקבל 3 כניסות:

- אות כניסה X
- אות כניסה Y
- קו בקרה ALUFN כשהביטים 3 ו-4 קובעים את המודל:

01 – מודל ADDERSUBTRACTOR

11 – מודל LOGIC

10 – מודל SHIFTER

כל מודול יכול לפעול במספר מצבים בהתאם לביטים 0, 1 ו-2 של קו הבקרה ALUFN.

המערכת תפיק 5 מוצאים:

- דגלים: Negative – N, Carry – C, Zero – Z, Overflow – V, ערך כל דגל בהתאם לשמו.
 - מוצא המערכת ALUout בהתאם למודל.
- ב – ISA של המערכת, עבור בחירת ביצוע פעולה שלא קיים – נרצה שכל המוצאים יהיו 0 ודגל Z יהיה 1.

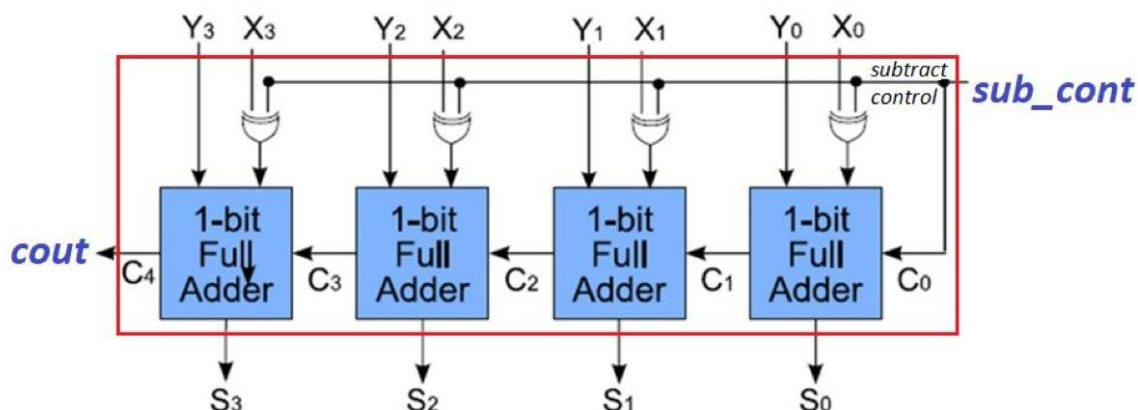
Function Kind	Decimal value	ALUFN	Operation	Note
Arithmetic	8	01000	Res=Y+X	
	9	01001	Res=Y-X	Used also for compare operation
	10	01010	Res=neg(X)	
Shift	16	10000	Res=SHL Y,X(k-1 to 0)	Shift Left Y of $q \triangleq X(k-1 \dots 0)$ times Res=Y(n-1-q...0)#(q@0) When $k = \log_2 n$
	17	10001	Res=SHR Y,X(k-1 to 0)	Shift Right Y of $q \triangleq X(k-1 \dots 0)$ times Res=(q@0)#Y(n-1...q) 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	

Table 1: Selected operations

ADDERSUBTRACTOR מודול

הסבר תיאורטי

המודל מבצע ע"פ כניסת ה-ALUFN: פעולת חיבור או חיסור בין הסיגנלים X ו-Y באורכים שווים באמצעות מחברי FA ובהתאם לקו בקרה sub_cont כאשר הנ"ל שווה ל-1 מתבצע חיסור, אחרת מתבצע חיבור או פעולת NEG המהפכת את הביטים של X ניתן לבצע זאת ע"י המודל הבא:



פעולת החיבור מתבצע כאשר $sub_count = 0$, והחיסור כאשר $sub_count = 1$ וההיפוך כאשר מחסרים את ווקטור האפסים מ-X.

הביצוע ב-VHDL נעשה ע"י הגדרת האותות X_temp ו-Y_temp כקלטות ל-FA-ים כך ש-X_temp הוא תוצאת XOR עבור חיבור/חיסור, היפוך עבור NEG ו-0 אחרת. Y_temp הוא Y עבור חיבור/חיסור ווקטור אפסים אחרת. לאחר מכן מבוצע שרשרת של n פעמים FA-ים עם הקלטים המתאימים מתוך X_temp ו-Y_temp.

תוצאות הסימולציה

נריץ סימולציה הכוללת מקרי בדיקה בסיסיים וקצה, תוך השוואה בין התוצאה המתקבלת לתוצאה הצפויה. במקרה של אי-התאמות תופק הודעת שגיאה המקרים שנבדקו:

- כניסת sub_count לא מוגדרת.
- NEG הפוך
- חיסור
- חיבור עם נשא
- חיבור ללא נשא

(תמונה של תוצאות)

	Msgs	
/addersub_tb/x	00000100	00000100 00000010 00000100 11111111 00000000
/addersub_tb/y	00000010	00000010 00000100 00000010 00000001 00000000
/addersub_tb/sub_...	000	000 001 000 001
/addersub_tb/cout	0	
/addersub_tb/s	00000110	00000110 00000010 11111110 00000000 00000000 00000000

ניתן לראות שלא הופק שגיאות, והתוצאות תואמות את הצפוי. למשל במקרה 3 מתבצע חיסור בין $Y = 00000010$ ו- $X = 00000100$ ומתקבל $RES = 11111110$ שזהו אכן -2 כפי שציפינו.

מודול Shifter

הסבר תיאורטי

מודל זה מבצע בהתאם לכניסת ALUFN הזזה מבוססת Barrel Shifter, עבור ביטים 0-2 השווים ל-"000" תבוצע הזזה שמאלה, ועבור "001" תבוצע הזזה ימינה. המימוש נעשה על ידי מעבר ע $K = \log_2 n$ שכבות, כאשר בכל שכבה מתבצעת הזזה בהתאם לביט המתאים ב: X-אם הביט 0, לא תתבצע הזזה ומוצא השכבה יהיה זהה לכניסה. אחרת תבוצע הזזה לפי השכבה עבור שכבה i נבצע הזזה של 2^i ביטים (ראו איור).

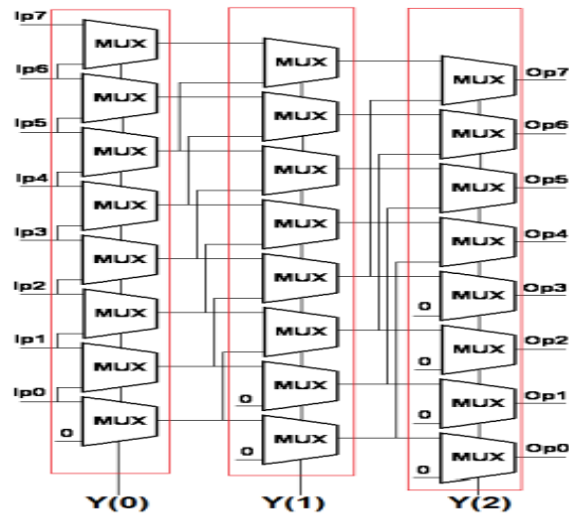


Figure 3: Example of 8-bit Barrel Shifter

עבור הזזה ימינה, תבוצע היפוך עבור השורה הראשונה והאחרונה.
כדי למצוא את ביט הנשא ביציאה, נגדיר סיגנל המכיל את הנשא בכל שלב ונחזיר את ערכו האחרון.

תוצאות הסימולציה

נריץ סימולציה עם מקרי בדיקה בסיסיים וקצה, תוך השוואת התוצאה לתוצאה הצפויה.
מקרי הבדיקה:

- כניסת sub_cont לא מוגדרת
- הזזה שמאלה (SHL)
- הזזה ימינה (SHR)

(תמונה תוצאות)

	Msgs	
/stb/dir	001	(000) 001 11111111 01101001 11110000 010
/stb/x	00101101	00000000 00101101 11111111 01101001 11110000
/stb/y	10000001	10101101 10000001 01001111 00100111 10000001
/stb/result	00000100	10101101 00000100 00000000 00010011 10000001 00000000
/stb/cout	0	

ניתן לראות שלא הופקו שגיאות והתוצאות תקינות.
למשל, במקרה עבור הקלטים $X=00101101$, $Y=10000001$ מתבצעת הזזה ימינה ($dir=001$) כאשר $k=3$.
נבחן את 3 הביטים הנמוכים ביותר ב X -שהם 101. ולכן צפויה הזזה ימינה של 5 ביטים ל Y -ללא נשא. לפיכך
האות המצופה הוא $res=00000100$, ו $cout=0$, וזהו אכן מה שהתקבל.

מודול Logic

הסבר תיאורטי

מודול זה מבצע פעולות לוגיות שונות על האותות X ו Y , בהתאם לכניסת ה-ALUFN הפעולות הלוגיות מיושמות ב- VHDL ומסוגלות להתסנתז בהתאם לביטים 0-2 של ה-ALUFN.

תוצאות הסימולציה

נריץ סימולציה עם מקרי בדיקה בסיסיים וקצה, תוך השוואת התוצאה לזו הצפויה. במקרה של אי-התאמה תופק שגיאה.

מקרי הבדיקה:

- | | |
|----------|---|
| X OR Y | • |
| NOT(Y) | • |
| X AND Y | • |
| X XOR Y | • |
| X NOR Y | • |
| X XNOR Y | • |
| X NAND Y | • |

(תמונה סימולציה)

0	1b010101	01010101																	11111111		11001100
1	1b010101	10101010																			00110011
2	01010101	01010101																	11111111		00000000
3	000	000	001	010	011	100	101	110	111	000	001	010	011	100	101	110	111	000	001	010	011

לא הופקו שגיאות ולכן התוצאות תקינות. למשל, עבור $Y=10101010$, $X=01010101$ מתבצעת פעולת $\text{NOT}(Y)$ (FN=000), והתוצאה המתקבלת היא, $\text{res}=01010101$. כצפוי.

מודול Top

הסבר תיאורטי

מודול זה עוסף את כל המערכת ומקבל את סינגלי הכניסה, מעביר אותם למודול המתאים ומסווג את המוצאים בהתאם.

רק המודול המתאים לפי ערך ה-ALUFN יהיה פעיל, והאחרים ינותקו. לכן תחילה נסווג את הסינגלים הנכנסים לכל מודול להיות האמיתיים רק עבור המודול הרלוונטי, עבור שאר הסינגלים נסווג (High - Z).

לאחר מכן נעביר את הסינגלים דרך המודולים, ונבחר את המוצא להיות המודול הפעיל בהתאם לערך ה-
ALUFN

דגלי המוצא של המודול יהיו כדלקמן:

- דגל ה-C יהיה מוצא ה- carry מהמודל הנבחר (עבור ה- Logic נגדיר '0' = carry)
- דגל ה-Z יקבל את הערך '1' כאשר המוצא הוא וקטור אפסים
- דגל ה-N יהיה הביט העליון (MSB) של התוצאה, כאשר '0' מייצג מספר חיובי ו-'1' מספר שלילי (לפי הגדרת משלים ל-2).
- דגל ה-V יקבל את הערך '1' כאשר במקרה של חיבור:
 1. חיבור X ו-Y חיוביים תניב תוצאה שלילית.
 2. חיבור X ו-Y שליליים תניב תוצאה חיובית.

במקרה של חיסור:

1. חיסור Y חיובי עם X שלילי תניב תוצאה שלילית.
2. חיסור Y שלילי עם X חיובי תניב תוצאה חיובית.

תוצאות הסימולציה

כעת נריץ סימולציה על כלל המערכת, שתכלול מקרי בדיקה במודולים השונים. הסימולציה תתבצע באמצעות קובץ ה testbench- בשם tb_ref1.vhd שניתן לנו. בכל מקרה נשווה בין התוצאה המתקבלת לצפויה.

(תמונת השוואה)

C:\Users\dane\Desktop\HDC\LAB19b_ref1.txt						
	ps	/cb/Y	/cb/X	/cb/ALOut		
	delta		/cb/ALUFN	/cb/Mflag		
				/cb/Cflag		
				/cb/Zflag		
				/cb/Vflag		
0	0	+ 11111111	11111111	01000	11111110	1 1 0 0
7	50000	+ 11111110	11110101	01001	11110011	1 1 0 0
8	100000	+ 11111101	11101011	01001	00010010	0 1 1 0
9	150000	+ 11111100	11100001	01001	00010101	0 1 0 0
10	200000	+ 11111011	11010111	01010	01010010	0 1 0 0
11	250000	+ 11111010	11001011	01010	01010011	0 0 0 0
12	300000	+ 11111001	11000011	01010	01111010	1 1 0 0
13	350000	+ 11111000	11111001	01010	11111001	1 1 0 0
14	400000	+ 11110111	11010111	01001	01010000	0 1 0 0
15	450000	+ 11110110	11000101	01001	01010001	0 1 0 0
16	500000	+ 11110101	11001011	01010	01010010	0 1 0 0
17	550000	+ 11110100	10010001	01010	01101111	0 0 0 0
18	600000	+ 11110011	10000101	01000	01111010	0 1 0 0
19	650000	+10 11110100	01111011	01000	01011110	0 1 0 0
20	700000	+ 11110001	11111011	01001	11111110	0 1 1 0
21	750000	+ 11110000	01010001	01001	10000111	1 1 0 0
22	800000	+ 11110111	01011111	10000	10000000	1 1 0 0
23	850000	+ 11111110	01010101	10000	11000000	1 1 0 0
24	900000	+ 11110101	01010011	10001	01011010	0 1 0 0
25	950000	+ 11110100	01000001	10001	01110110	0 0 1 0
26	1000000	+ 11110111	01011011	10010	00000000	0 0 1 0

C:\Users\dane\Desktop\HDC\LAB19\Tour1.txt						
	ps	/cb/Y	/cb/X	/cb/ALOut		
	delta		/cb/ALUFN	/cb/Mflag		
				/cb/Cflag		
				/cb/Zflag		
				/cb/Vflag		
0	0	+ 11111111	11111111	01000	11111110	1 1 0 0
7	50000	+ 11111110	11110101	01001	11110011	1 1 0 0
8	100000	+ 11111101	11101011	01001	00010010	0 1 1 0
9	150000	+ 11111100	11100001	01001	00010101	0 1 0 0
10	200000	+ 11111011	11010111	01010	01010010	0 1 0 0
11	250000	+ 11111010	11001011	01010	01010011	0 0 0 0
12	300000	+ 11111001	11000011	01010	01111010	1 1 0 0
13	350000	+ 11111000	11111001	01010	11111001	1 1 0 0
14	400000	+ 11110111	11010111	01001	01010000	0 1 0 0
15	450000	+ 11110110	11000101	01001	01010001	0 1 0 0
16	500000	+ 11110101	11001011	01010	01010010	0 1 0 0
17	550000	+ 11110100	10010001	01010	01011011	0 0 0 0
18	600000	+ 11110011	10000101	01000	01111010	0 1 1 0
19	650000	+10 11110100	01010001	01000	01011010	0 1 0 0
20	700000	+10 11110001	01111001	01001	11111110	0 1 1 0
21	750000	+10 11110000	01010001	01000	10000111	1 1 0 0
22	800000	+ 11110111	01011111	10000	10000000	1 1 0 0
23	850000	+ 11110110	01010101	10000	11000000	1 1 0 0
24	900000	+ 11110101	01010011	10001	01011010	0 1 0 0
25	950000	+ 11110100	01000001	10001	01110110	0 0 1 0
26	1000000	+ 11110111	01011011	10010	00000000	0 0 1 0

ניתן לראות שהתקבלו התוצאות הצפויות לאורך כל הבדיקות שבוצעו על המערכת, התוצאות בשני הקבצים זהות, מלבד הבדלים זניחים (delta) שניתן להתעלם מהם.

נריץ גם סימולציה שאנו כתבנו שתכלול בדיקות על כלל המודלים, אשר תרוץ על שכבת ה TOP ונבדוק שאכן המערכת פועלת כראוי.

(תמונה תוצאות)

Timing diagram for the 'b' bus in the 'b' block. The diagram shows the bus activity for various signals: /b_top/? (input), /b_top/A (input), /b_top/ALLPM (input), /b_top/ALLout (input), /b_top/R#ag (input), /b_top/C#ag (input), /b_top/Z#ag (input), and /b_top/I#ag (input). The bus is shown as a series of 1s and 0s, with a vertical yellow line indicating a specific time point.

ניתן לראות שלא הופקו שגיאות והתוצאות תקינות.

למשל, במקרה עבור הקלטים $X=11111111$, $Y=11111111$ ו $ALUFN=01010$ נדרש לבצע $NEG(X)$ וניתן לראות שאכן זה מתבצע. הכנסנו 1- ובמוצא קיבלנו 1.

לסיכום, המערכת שפותחה מבצעת באופן תקין את הפעולות השונות במודולים AdderSub, Shifter ו-Logic בהתאם לאותות הבקרה. המודול העוטר Top מנתב את האותות בהתאם ומפיק את המוצאים הנכונים. תוצאות הסימולציה מראות התאמה בין הפלט בפועל לבין הפלט הצפוי, מה שמעיד על תקינות המימוש של המער