PREPARATION REPORT LAB1

ADVANCED CPU ARCHITECTURE AND HARDWARE ACCELERATORS LAB

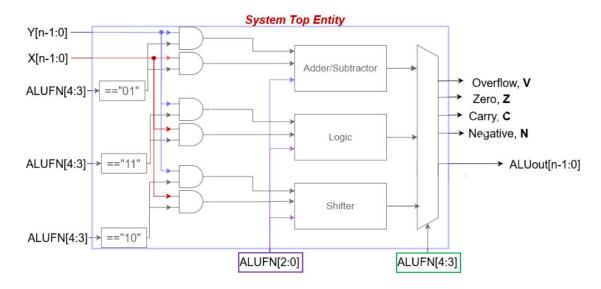
361.1.4693

Nachman Mimoun 321730558 Danel Barsheshet 209471242

תוכן עניינים מבוא 3 AdderSub הסבר תיאורטי 4 4 Shifter 4 הסבר תיאורטי 5 הסבר תיאורטי 5 הסבר תיאורטי 6 הסבר תיאורטי 7 תוצאות סימולציה וסיכום 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:

- X אות כניסה •
- Y אות כניסה
- קו בקרה ALUFN כשהביטים 3 ו-4 קובעים את המודל:
 - ADDERSUBTRACTOR מודל 01
 - LOGIC מודל 11
 - SHIFTER מודל 10

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

:המערכת תפיק 5 מוצאים

- . ערך כל דגל בהתאם לשמו. Overflow V ,Zero Z ,Carry C ,Negative Negative V ,
 - מוצא המערכת ALUout בהתאם למודל.
- .1 יהיה ביצוע ודגל סייבה שכל ברצה ביצוע פעולה שלא פיים ביצוע פעולה בחירת ביצוע ודגל ודגל ודגל וודגל בZ

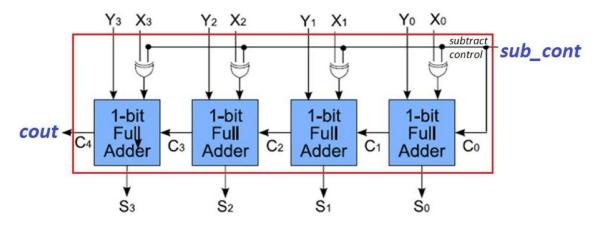
Function	Decimal	ALUFN	Operation	Note
Kind	value			
Arithmetic	8	01000	Res=Y+X	
	9	01 001	Res=Y-X	Used also for compare operation
	10	01 010	Res=neg(X)	
Shift	16	10000	Res=SHL Y,X(k-1 to 0)	Shift Left Y of q≜X(k-10) times
				Res=Y(n-1-q0)#(q@0)
				When $k = log_2 n$
	17	10 001	Res=SHR Y,X(k-1 to 0)	Shift Right Y of q≜X(k-10) times
				Res=(q@0)#Y(n-1q)
				When $k = log_2 n$
Boolean	24	11000	Res=not(Y)	
	25	11 001	Res=Y or X	
	26	11 010	Res=Y and X	
	27	11 011	Res=Y xor X	
	28	11 100	Res=Y nor X	
	29	11 101	Res=Y nand X	
	30	11 1111	Res=Y xnor X	

Table 1: Selected operations

ADDERSUBTRACTOR מודול

הסבר תיאורטי

המודל מבצע ע"פ כניסת הN-1: פעולת חיבור או חיסור בין הסיגנלים Y-1 באורכים שווים באמצעות פעולת מחברי Sub_cont בקרה הב"ל שווה לN-1 מתבצע חיסור, אחרת מתבצע חיבור או פעולת מחברי FA בהתאם לקו בקרה ציים של X ניתן לבצע זאת ע"י המודל הבא:



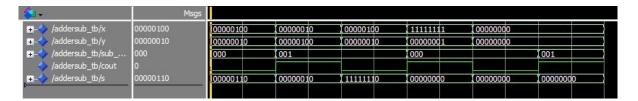
הוא X_temp -ים כך ל-FA-ים כקלטים ל-temp ו- X_temp הביצוע ב VHDL - הביצוע ב VHDL נעשה ע"י הגדרת האותות ע"י הגדרת האותות אווע א עבור היפור היפור עבור אווע ווקטור אחרת. אווע א עבור היבור\חיסור, היפוך עבור עבור NEG האחרת. עבור היבור\חיסור אפסים אחרת. לאחר מכן מבוצע שרשור של ח פעמים FA-ים עם הקלטים המתאימים מתוך עד שרשור של ח

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.
 - חפוך NEG
 - חיסור •
 - חיבור עם נשא
 - חיבור ללא נשא

(תמונה של תוצאות)



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

Shifter מודול

הסבר תיאורטי

"000" שבוע ביטים 0-2 עבור ביטים אווים ל-"000" הזזה מבוססת הדר אווים ל-"000" עבור ביטים ל-"000 אכבות, הבוצע הזזה שמאלה, ועבור "001" תבוצע הזזה ימינה. המימוש נעשה על ידי מעבר ע $K = \log_2 n$ שכבות, עבוצע הזזה שכבה ועבור "001" תבוצע הזזה בהתאם לביט המתאים ב"אם הביט 0, לא תתבצע הזזה ומוצא השכבה יהיה לכניסה. אחרת תבוצע הזזה לפי השכבה עבור שכבה ונבצע הזזה של 2^i ביטים (ראו איור).

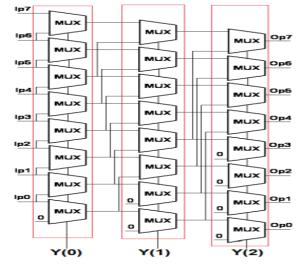


Figure 3: Example of 8-bit Barrel Shifter

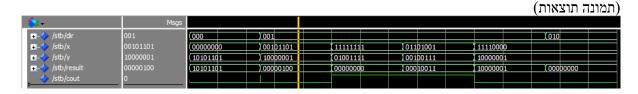
עבור הזזה ימינה, תבוצע היפוך עבור השורה הראשונה והאחרונה.

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)
 - (SHR) הזזה ימינה •



ניתן לראות שלא הופקו שגיאות והתוצאות תקינות.

k=3למשל, במקרה עבור הקלטים למשל, א=10000001, X=00101101 מתבצעת הזזה ימינה (dir=001) כאשר לפיכך למשל, במקרה עבור בא-שהם 101. ולכן צפויה הזזה ימינה של 5 ביטים לY-ללא נשא. לפיכך נבחן את 3 הביטים הנמוכים ביותר ב-Cout=0, וres=00000100.

Logic מודול

הסבר תיאורטי

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

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) \bullet
- X AND Y •
- X XOR Y
- X NOR Y •
- X XNOR Y •
- X NAND Y •

(תמונה סימולציה)

-																					
	01010101	01010101								11111111								11001100			
→ /tb_logic/y	10101010	10101010								11111111								00110011			
/tb_logic/Res	01010101	01010	11111	00000	11111	00000	11111	00000000			шшш		00000000				11111	11001	11111	00000	11111
/tb_logic/ALUFN	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 הופקו שגיאות ולכן התוצאה המתקבלת היא אור וואס והתוצאה המתקבלת היא NOT(Y) (FN=000),

מודול Top

הסבר תיאורטי

מודול זה עוטף 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 הסיגנלים נסווג (High - Z) Z.

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

דגלי המוצא של המודול יהיו כדלקמן:

- (carry = '0' נגדיר Logic מהמודל הנבחר (עבור ה- carry מוצא ה- C- דגל ה- רביה מוצא ה- מוצא ה- מוצא ה-
 - 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).
 - יקבל את הערך '1' כאשר במקרה של חיבור: V יקבל את הערך
 - 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 testbench שניתן לנו. בכל מקרה נשווה בין התוצאה המתקבלת לצפויה.

(תמונת השוואה)

_	Jsers\daneNDesktop	NVADL					_	_	_	_
1	ps		/tb/Y	/tb/X		tb/ALUout				
2	de	elta		/tb.	/ALUFN	/tb/Nfla				
						/tb/C: /tb/				
4										
5									fla	ag n
6 7	50000	+8	111111111	111111111			1	i	0	0
8	100000	+7		11110101			0	1	0	0
9				11101011				i	0	0
	150000	+8		11010111			0	0	0	0
0	200000	+8	111111011				0	0	0	0
2				11001101		101111100	-	-	-	0
	300000 350000	+8	11111001				1	1	0	0
3		+8					-	-	0	
1	400000	+9		10101111		01001000	0	1	0	0
5	450000	+7		10100101			0	0	0	0
	500000			10011011			0	0	0	0
7	550000		11110100							1
3	600000	+8		10000111			0	1	0	-
9	650000			01111101			0	1	0	0
9	700000	+9		01110011			0	1	0	1
1	750000	+9		01101001		10000111	1	1	0	0
2	800000	+9		01011111			1	1	0	0
3	850000	+8		01010101			1	1	0	0
	900000	+7		01001011		00011101	0	1	0	0
5	950000	+8		01000001		01110110	0	0	0	0
	1000000	+9	11101011	00110111	10010	00000000	U	0	1	0
	Jsers\daneNDesktop	NVHDL	\LAB1\Ourlist.le	st						
		NVHDL	\LAB1\Ourlist.lt	st /tb/X	/1	tb/ALUout				
	ps	NVHDL		/tb/X	/1 /ALUFN	tb/ALUout /tb/Nfla	ag			
1	ps			/tb/X				a.c		
2	ps			/tb/X		/tb/Nfla	fla		ag	
1	ps			/tb/X		/tb/Nfla /tb/C: /tb/	f1:	fĺ	ag fla	acr
	ps	elta		/tb/X /tb	/ALUFN	/tb/Nfla /tb/C: /tb,	fla /Z: tb;	fla /Vi		ag 0
1 2 3 4 5 5	ps de	elta	/tb/Y	/tb/X /tb	/ALUFN	/tb/Nfla /tb/C: /tb, /tb, /1	fla /Z: tb;	fla /Vi	fla	
1 2 3 4 5 7	ps de 0 50000	elta +8	/tb/Y	/tb/X /tb.	01000 01000	/tb/Nfla /tb/Ci /tb, /t /t 11111110 11110011	fla /Z: tb;	fla /Vi	fie	ō
1 2 3 4 5 6 7 8	ps de	elta +8 +8	/tb/Y	/tb/X /tb/	01000 01000 01000	/tb/Nf1s /tb/C: /tb/ /tb/ 11111110 11110011 00010010	fla /Z: tb; 1	fla /V: 1	fla 0 0	0
1 2 3 4 5 6 7 8 9	ps de 0 50000 100000	+8 +8 +7	/tb/Y	/tb/X /tb/	01000 01000 01000 01001 01001	/tb/Nf1s /tb/C: /tb, /t 11111110 11110011 00010010 00011011	fla /Z: tb; 1	fla /V: 1 1	0 0 0	0 0
1 2 3 4 5 6 7 8 9 0	ps de 0 50000 100000 150000	+8 +8 +7 +8 +8	/tb/Y	/tb/X /tb, 11111111 11110101 11101011 11100001 11010111	01000 01000 01000 01001 01001 01010	/tb/Nf1a /tb/C: /tb/ /tb/ /tllllll0 11110011 00010010 00011011 00101001	fla /Zi tb; 1 0	fla /V: 1 1	0 0 0 0	0 0 0
1 2 3 4 5 6 7 8 9 0 1	ps de 0 50000 100000 150000 200000	+8 +8 +7 +8 +8 +6	/tb/Y	/tb/X /tb, 11111111 11110101 11101011 11100001 11010111	01000 01000 01000 01001 01001 01010	/tb/Nf1: /tb/C:	fla /Z: tb; 1 0 0	fla /V: 1 1 1 0	0 0 0 0 0	0 0 0 0
1 2 3 4 5 6 7 8 9 0 1 2	ps de 50000 100000 150000 200000 250000	+8 +8 +7 +8 +8	/tb/Y	/tb/X /tb/ 11111111 11110101 11101011 11100001 11010111 11001101	01000 01000 01000 01001 01010 01010 01010	/tb/Nf12 /tb/C: /tb, /till110 11110011 00010010 00011011 00101001 00110011 10111100	fla /Z: tb; 1 0 0	/V: 1 1 1 0 0	0 0 0 0 0	000000
1 2 3 4 5 6 7 8 9 0 1 2 3	ps de 0 50000 100000 200000 250000 350000 350000	+8 +8 +7 +8 +6 +8	/tb/Y	/tb/X /tb/X /tb/ 11111111 11110101 11101011 11100001 110101101	01000 01000 01000 01001 01001 01010 01010 01000	/tb/Nf1a /tb/C: /tb/ 111111001 00010010 00011011 000101001 00110011 10111100	fla /Z: tb/ 1 0 0 0	/V: 1 1 1 0 0	0 0 0 0 0	0000000
1 2 3 4 5 6 7 8 9 0 1 2 3 4	ps de 0 0 0 50000 100000 150000 250000 350000 350000 400000	+8 +8 +7 +8 +6 +8 +9	/tb/Y 1111111 1111110 1111101 1111101 1111101 11111001 11111001	/tb/X /tb/X /tb/X 11111111 11110101 1110001 1100001 11001101 11000001 10111001	01000 01000 01001 01001 01010 01010 01000 01000	/tb/Nf1: /tb/C: /tb/ /tb/ 11111100 1110011 00010010 00011011 00110011 10111100 10110001 010010	fla /Z: tb, 1 0 0 0 1 1	fla /Vi 1 1 1 0 0 1	0 0 0 0 0 0	00000000
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5	ps de 0 0 50000 100000 200000 250000 350000 400000 450000	+8 +8 +7 +8 +6 +8 +9 +9	/tb/Y 1111111 1111110 11111101 11111001 11111001 11111001 11111001 11111001 11111011	/tb/X /tb/X /tb/X /tb/ 11111111 11110101 11100011 11100011 11001011 11000011 1010111001 1010010	01000 01000 01000 01001 01001 01010 01010 01000 01001 01001	/tb/Nf1a /tb/C: /tb/ 11111110 11110011 00010010 00011011 00110001 101111000 10100010 010010	fla /Z: tb/ 1 0 0 0 1 1 0	fla /V: 1 1 1 0 0 1 1	f10 0 0 0 0 0 0 0	0000000
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6	ps de 50000 100000 150000 250000 450000 500000 500000	+8 +8 +7 +8 +6 +8 +9 +9	/tb/Y 11111111 11111100 11111010 11111010 11111010 11111011 11110110	/tb/X /tb, 11111111 11110101 1110001 11100001 11001011 11000011 10101011 1010010	01000 01000 01000 01001 01001 01010 01000 01001 01001 01001	/tb/Nf12 /tb/C: /tb/C: /tb/ 11111110 1110011 00010010 00011001 00101001 10111000 10100001 01010001 0110001 01100100	fla /Z: tb, 1 0 0 0 1 1 0 0	fla /Vf 1 1 1 0 0 1 1 1	f12 0 0 0 0 0 0 0 0	0000000000
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7	ps de 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	+8 +8 +7 +8 +8 +6 +8 +9 +9 +7 +9	/tb/Y 1111111 11111101 11111001 11111001 11111001 11110101 11110101 11110101 11110101 11110101 11110101	/tb/X //tb,	01000 01000 01000 01001 01001 01010 01000 01000 01001 01001 01010	/tb/Nf12 /tb/C:	fla /Zi tb/ 1 0 0 0 1 0 0 0	/V: 1 1 1 0 0 1 1 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	00000000000
123456789012345678	98 de 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	+8 +8 +7 +8 +6 +8 +9 +7 +9 +7 +9	/tb/Y	/tb/X /tb, /tb, /tb, /tb, /tb, /tb, /tb, /tb,	01000 01000 01000 01001 01001 01010 01000 01000 01001 01001 01010 01010	/tb/Nf1a /tb/C: /tb/C: /tb/C: /till11001 11110101 00010010 00011011 00110011 10111100 10110001 010010	fla /Zi tb/ 1 0 0 0 0 1	fla /Vf 1 1 1 0 0 1 1 1 0 0 1	612 0 0 0 0 0 0 0 0 0	0000000000
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9	98 de 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	+8 +8 +7 +8 +6 +8 +9 +7 +9 +6 +8 +10	/tb/Y 11111111 11111101 1111101 1111101 11111010 111110101 111110110	/tb/X /tb,	01000 01000 01001 01001 01001 01010 01000 01001 01001 01010 01010 01010 01010 01010	/tb/Nf1a/ct/Ct/Ct/Ct/Ct/Ct/Ct/Ct/Ct/Ct/Ct/Ct/Ct/Ct	fla /Zi tb) 1 0 0 0 1 1 0 0 0 0 0	/V: 1 1 1 0 0 1 1 1 0 0 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 1 0
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0	98 de 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	+8 +8 +7 +8 +8 +6 +9 +9 +7 +9 +6 +8 +10 +10	/tb/Y 11111111 11111101 11111101 1111101 11111011 11111001 11111011 11110110	/tb/X /tb, /tb, /tb, /tb, /tb, /tb, /tb, /tb,	01000 01000 01001 01001 01001 01010 01000 01001 01001 01010 01010 01000 01000	/tb/Nf1a /tb/Ci /tb/Ci /tb/ 11111110 11111011 00010010 00010010 0010001 10111100 10100001 0100000 0100001 01100101 01100101 01100101 01100101 011011	fla /Zi tb; 1 1 0 0 0 1 1 0 0 0 0 0 0	fle /V/ 1 1 1 0 0 1 1 1 0 0 1 1	610 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 1 0 1
123456789012345678901	ps de	+8 +8 +7 +8 +6 +8 +9 +7 +9 +7 +9 +6 +8 +10 +10 +10	/tb/Y 11111111 11111101 1111101 11111010 11111010 11111010 11111010 11110101 11110101 11110101 11110101 11110010 11110010 11110010 11110010 11110010	/tb/X /tb, /tb, /tb, /tb, /tb, /tb, /tb, /tb,	01000 01000 01001 01001 01001 01010 01000 01001 01010 01010 01010 01000 01001 01001 01001	/tb/Nf1a /tb/C: /tb/C: /tb/C: /tb/C: /tll111100 11110011 00010010 00010011 0010001 0110001 10111000 1010001 01100101 01100101 011011	fla /Z: tb) 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0	/Vi 1 1 1 0 0 1 1 1 1 0 0 1 1 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 1 0 1 0
1234567890123456789012	98 de 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	+8 +8 +7 +8 +8 +6 +8 +9 +7 +9 +6 +8 +10 +10 +10 +9	/tb/Y 11111111 11111101 11111101 11111001 11111001 111110101 11110110	/tb/X /tb, /tb, /tb, /tb, /tb, /tb, /tb, /tb,	01000 01000 01000 01001 01001 01010 01000 01001 01001 01001 01010 01000 01001 01000 01001 01000 01001	/tb/Nf1: /tb/C: /tb/Ci/C: /tb/Ci/Ci/C: /tb/Ci/Ci/Ci/Ci/Ci/Ci/Ci/Ci/Ci/Ci/Ci/Ci/Ci/	flatb/Zitb/ 1 0 0 0 0 1 1 0 0 0 0 0 1 1 1	/V/ 1 1 1 0 0 1 1 1 1 1 1 1 1 1	610 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 1 0 1 0 0
1 2 3 4 5 5 6 7 8 9 9 0 1 1 2 3 3 4 5 6 7 7 8 9 9 0 1 1 1 1 2 3 3 3 4 9 1 1 1 2 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1	ps de	+8 +8 +7 +8 +6 +8 +9 +7 +9 +6 +8 +10 +10 +9 +9	/tb/Y 11111111 11111111 1111110 11111101 11111001 11111001 11111011 11110101 11110101 11110010 11110010 11110010 11110010 11110010 11110010 11110010 11110010 11110010 11110010 11110010 11110010	/tb/X /tb, /tb, /tb, /tb, /tb, /tb, /tb, /tb,	01000 01000 01001 01001 01001 01010 01000 01000 01001 01010 01010 01000 01000 01001 01001 01001 01000 01000 01000	/tb/Nf1a /tb/C: /tb/C: /tb/C: /tb/C: /tll1111001 1001001001 000110011 00110011 10111100 10101001 01010001 01100101 011011	flatb/Zitb/ 1 1 0 0 0 0 1 1 0 0 0 0 1 1 1 1	fla /V/ 1 1 1 0 0 1 1 1 0 0 1 1 1 1 1 1 1 1	f10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2	ps de	+8 +8 +7 +8 +6 +8 +9 +7 +9 +10 +10 +10 +9 +9	/tb/Y 11111111 11111101 11111101 11111011 11111001 11111010 11110101 11110101 11110101 11110101 11110101 11110001 11110001 11110001 11110001 11110001 11110001	/tb/X /tb, /tb, /tb, /tb, /tb, /tb, /tb, /tb,	01000 01000 01000 01001 01001 01001 01000 01001 01001 01010 01010 01010 01000 01001 01001 01001 01000 01001	/tb/Nf1: /tb/C: /tb/Ci/Ci/Ci/Ci/Ci/Ci/Ci/Ci/Ci/Ci/Ci/Ci/Ci/	f1a /Z: tb/ 1 1 0 0 0 0 1 1 0 0 0 0 0 1 1 0 0 0 0	fla /V/ 1 1 1 0 0 1 1 1 1 1 1 1 1	f10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	000000000000000000000000000000000000000
	ps de	+8 +8 +7 +8 +6 +8 +9 +7 +9 +10 +10 +10 +9 +9 +8	/tb/Y 11111111 11111111 1111110 11111101 11111001 11111001 11111011 11110101 11110101 11110010 11110010 11110010 11110010 11110010 11110010 11110010 11110010 11110010 11110010 11110010 11110010	/tb/X /tb, /tb/X /tb, /tb/X /tb, /tb/X /tb, /tb/X /tb, /tb/X /tb, /tb/X	01000 01000 01000 01001 01001 01010 01000 01000 01001 01010 01010 01000 01001 01001 01001 01001 01000 01000 01000 01000 01000	/tb/Nf14 /tb/Ci /tb/Nf14 /tb/Ci /tb/Nf14 /tb/	flatb/Zitb/ 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0	fla /V/ 1 1 1 0 0 1 1 1 1 1 1 1 1 1	fl:	000000000000000000000000000000000000000

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.

ניתן לראות שלא הופקו שגיאות והתוצאות תקינות. Y=11111111, X=11111111 נדרש לבצע NEG(X) נדרש למשל, במקרה עבור הקלטים Y=11111111, X=11111111 נדרש לבצע וניתן לראות שאכן זה מתבצע, הכנסנו 1- ובמוצא קיבלנו 1.

לסיכום, המערכת שפותחה מבצעת באופן תקין את הפעולות השונות במודולים AdderSub, Shifter ו- Logic ו- Dogic בהתאם לאותות הבקרה. המודול העוטף 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