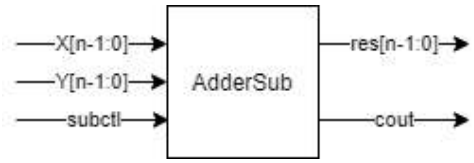
Elias Assaf 315284729
Jameel Nassar 206985152

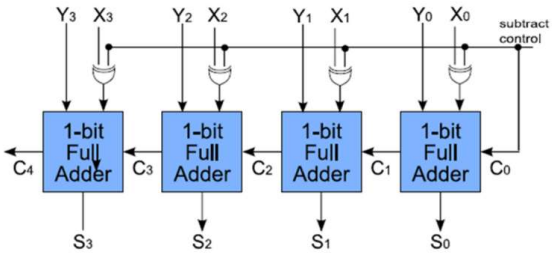
LAB1 - VHDL part1

Adder/Sub

הסבר:

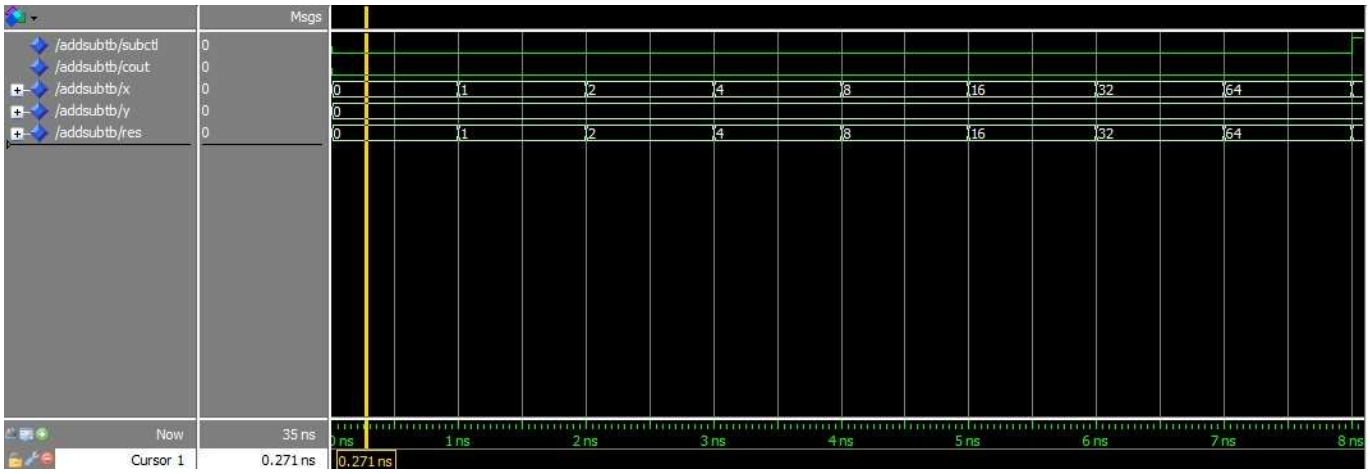


הרכיב מקבל 3 כניסות, X Y המחוברים ועוד כניסה $subctl$ שכובעת אם נבצע סכום או חיסור.
ויש 2 יציאות res שמכילה את תוצאת החיבור\חיסור, $cout$ השארית.
הרכיב עובד כמו $rippleadder$ שבו משרשרים FA של ביט יחיד כמו שמתואר בשרטוט:

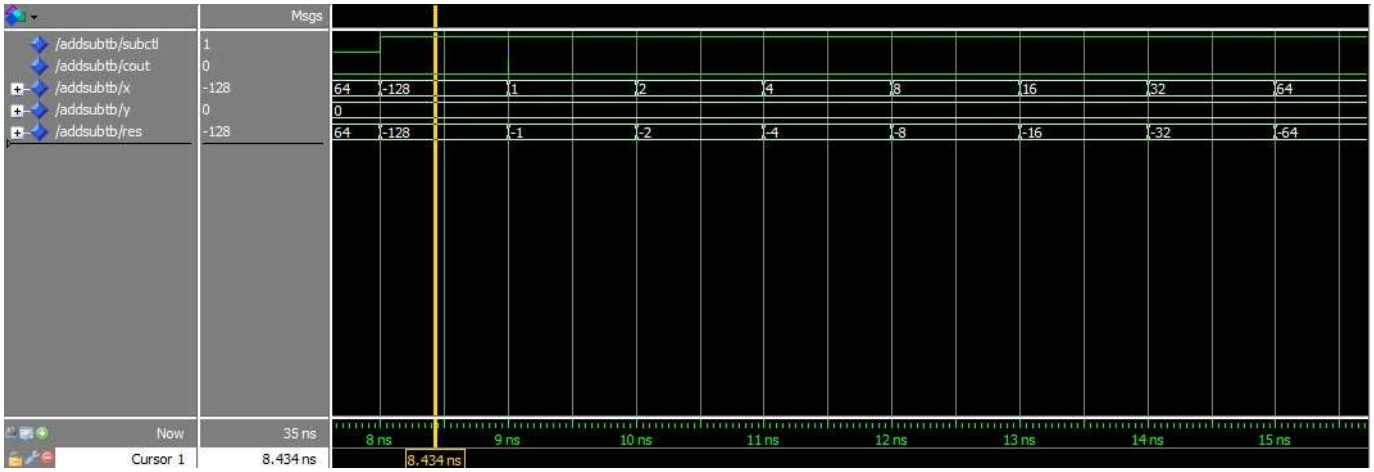


תוצאות של $AdderSub - TB$ **:**

שנינו את ייצוג הוקטורים X, Y ל- $decimal$ לנוחות הקריאה.
בחלק ראשון (1ns - 8ns) ביצענו $X + Y$ עם $Y = 0$ לבדוק האם חיבור כל ביט בודד יעבוד.

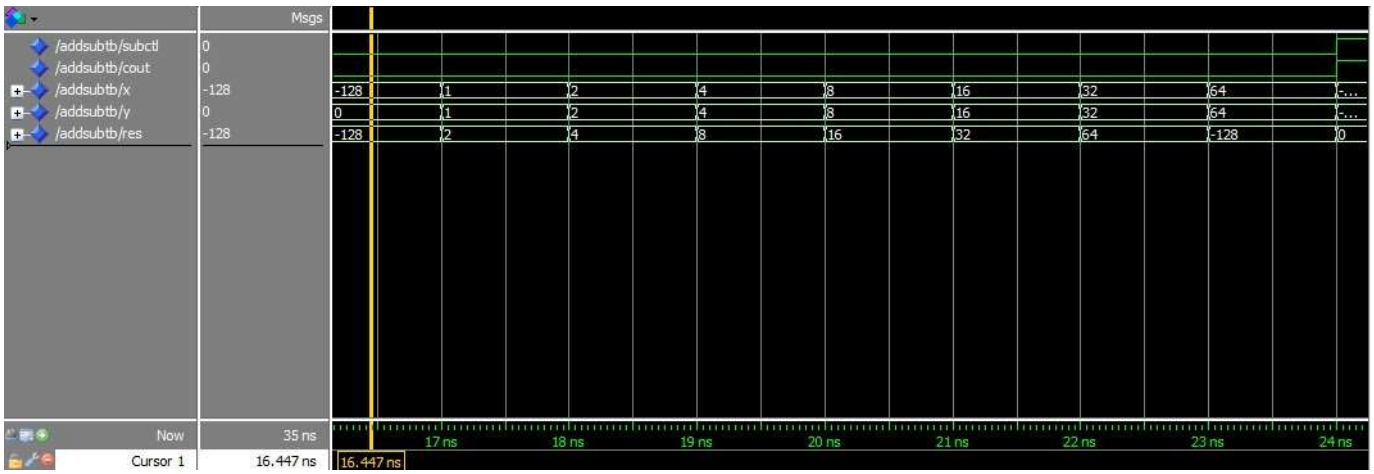


בחלק שני (8ns-16ns) ביצענו $Y - X$ עם $Y = 0$ כדי לבדוק את החיסור ביט לביט.



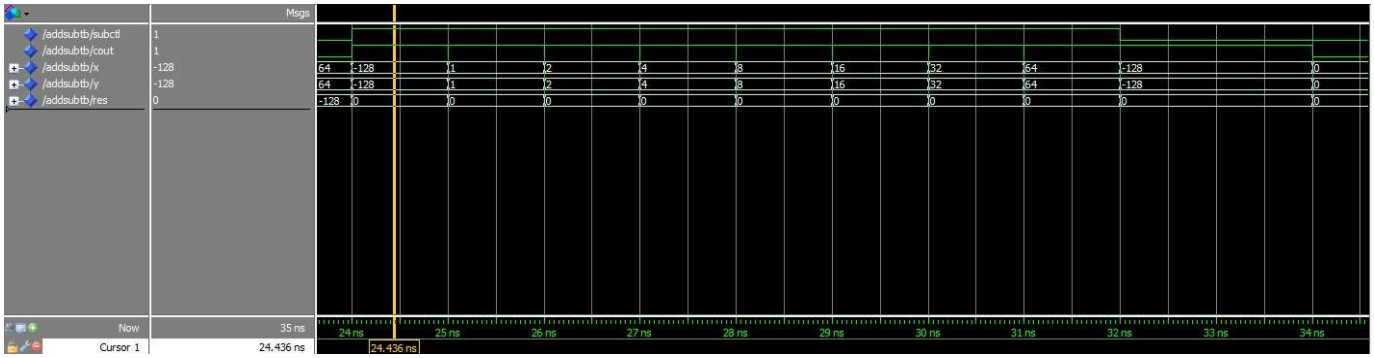
נשים לב ב- 8 ns עליית subctl ל-1.

בחלק שלישי (16ns-24ns) ביצענו $X + Y$ עם $X = Y$ לבדוק חיבור בין 1 ל- 1.



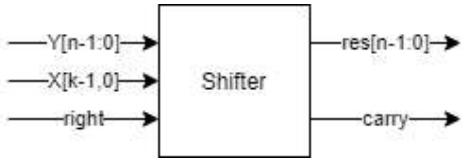
בחלק רבעי (24ns -32ns) ביצענו $Y - X$ עם $X = Y$ תמיד נקבל תוצאה 0.

נשים לב ב- 32 ns ביצענו חיבור בין -128 + -128 שבייצוג בינארי זה $10000000 + 1000000$ נקבל תוצאה 0 עם שארית 1 כמו שרואים בציור.



Shifter

הסבר:



מקבלת וקטור $Y[n-1:0]$ שמבצעים $shift$ עליו, וקטור $X[k-1:0]$ שמתאר מספר ההזזות, ו $right$ ביט המייצג את כיוון הזזה הרצויה.

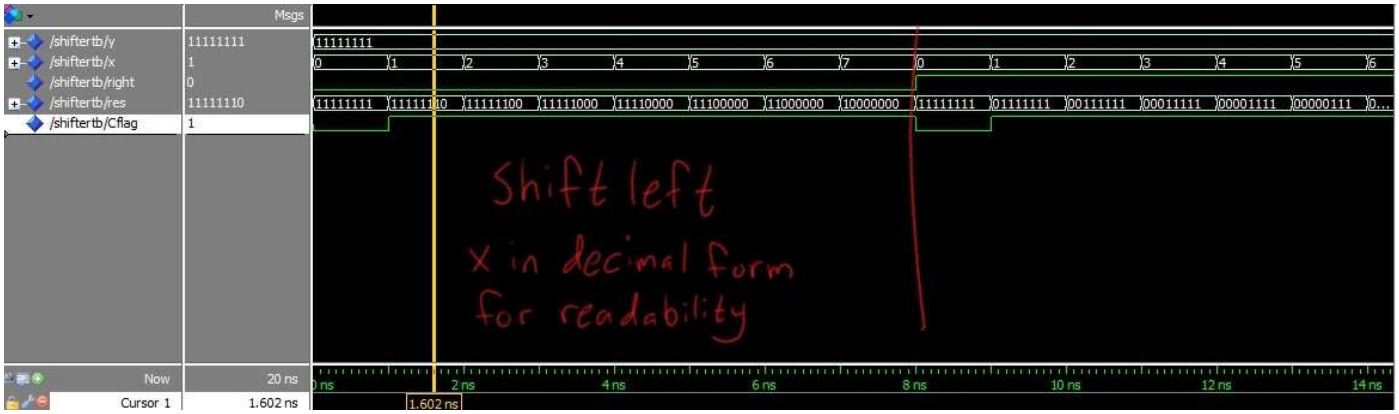
הרכיב עובד בצורה הבא:

מייצרים שני מערכים שמכילים כל הזזות האפשריות לכניסה Y , ונעשה המרה ל- X ליחידות integer, ונשתמש ב if ו- when לבחור את הערך המתאים.

נבחור ה- $carry$ ע"י בדיקת המערך שמכיל ההזזות במקום $X-1$ אחרי המרה למספר (אחד לפני ההזזה הרצויה), ונבחר ביט ה- MSB כאשר מזיזים $left$, LSB כאשר מזיזים $right$.

תוצאות של Shifter

שנינו את ייצוג הוקטורים X ל- $decimal$ לנוחות הקריאה.

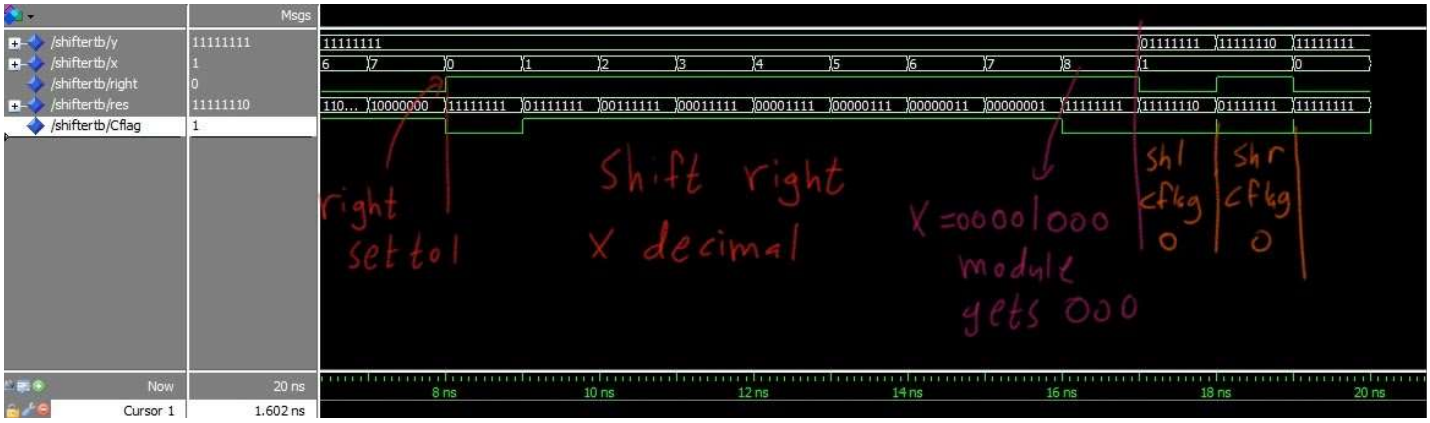


בחלק ראשון (1ns-8ns) ביצענו הזזה לשמאל ל $Y = 11111111$ כאשר הערכים של X עולים מ 0 ל- 7(000-111).

אפשר לראות שהערך של Cflag עולה ל 1 כאשר ביא ה MSB הוא 1.

בחלק השני (8ns-16ns) ביצענו הזזה לימין ל $Y = 11111111$ כאשר הערכים של X עולים מ 0 ל- 7(000-111).

בחלק השלישי (16ns-20ns) בדקנו את ערך ה Cflag כאשר הוא צריך להיות אפס.



Logical TB

ממשנו אותו ב- top module והשתמשנו בפקודות הלוגיות של $STD-LOGIC$.
השתמשנו ב select לבחור את הפקודה המתאימה לפי $ALUFN[1:0]$ (מסבירים על זה בחלק של ה- top)

100	Res=Y or X
101	Res=Y and X
110	Res=Y xor X
111	Res=Y nor X

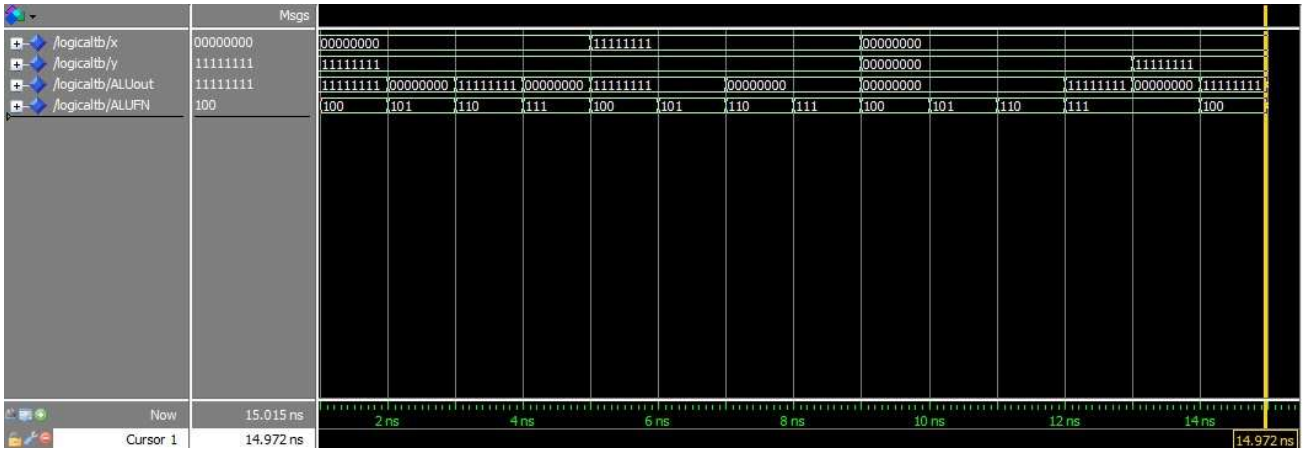
תוצאות:

נשים שלא חיברנו את ה- $flags$ כי רק רוצים לבדוק תוצאות הפעולות, בודקין ה- $flags$ בחלק של top.

בין $0-5ns$ בדקנו את תוצאת עבור $x=0...0,\ y=1...1$

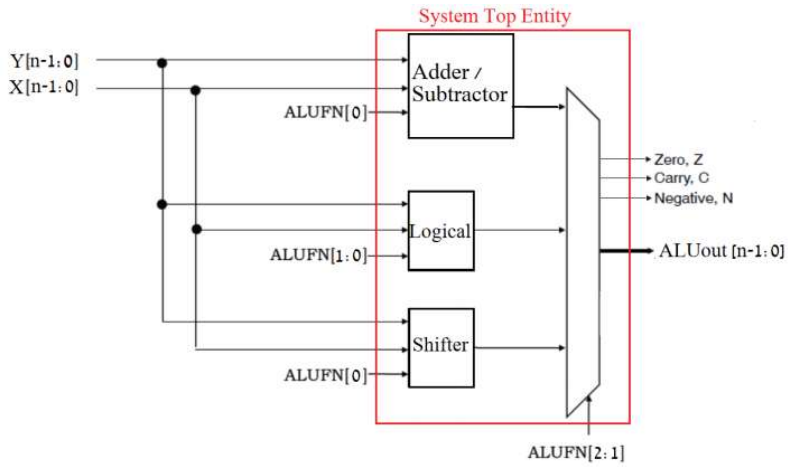
בין $5-9\ ns$ בדקנו את תוצאת עבור $x=1...1,\ y=1...1$

בין $9-13\ ns$ בדקנו את תוצאת עבור $x=0...0,\ y=0...0$



Top

הסבר:



הרכיב מקבל 3 כניסות X,Y,ALUFN, ונקבל 4 יציאות

ALUout - התוצאה

Z - אחד כאשר התוצאה אפס, אחרת אפס.

N - אם המספר שלילי (MSB הוא 1) הוא 1 אחרת 0.

C - השארית לפי הפעולה המתאימה.

נחשב את כל הערכים הרצויים במקביל ושומרים אותם במערך $rows$, הזמן שרוצים להוציא את התוצאה נעשה זה לפי $AFLUFN[2:1]$, בצורה הבא:

00 - נבחר תוצאת ה- $AdderSub$ ו- $ALUFN[0]$ מגדיר אם סוכמים או מחסירים.

01 - נבחר תוצאת ה- $Shifter$ ו- $ALUFN[0]$ מגדיר אם מזיזים שמאל או ימין.

10 - בוחרים בין or/and כאשר $ALUFN[1:0]$ מחליט בין or/and.

11 - בוחרים בין xor/nor כאשר $ALUFN[1:0]$ מחליט בין xor/nor.

תוצאות:

נשים לב עבור פקודת הזזה ימינה נקבל תמיד ש- $Nflag = 0$ כי אם היה ביט במקום ה- $n-1$ אז נדחוף אותו ימינה ונכניס 0 במקומו.

עבור הפעולות הלוגיות תמיד $Cflag = 0$.

עבור הזזה של $y = 11111111,\ x = 00000111$, הערך המקסימלי של x ב- $shift$ נקבל ש- $ALUout = 10000000$ ואז לא מתאפס.