

LAB 4

ADVANCED CPU ARCHITECTURE AND HARDWARE ACCELERATORS LAB

Ido Ron: 322384330

Binyamin Oumanski: 322688946

מטרת המעבדה

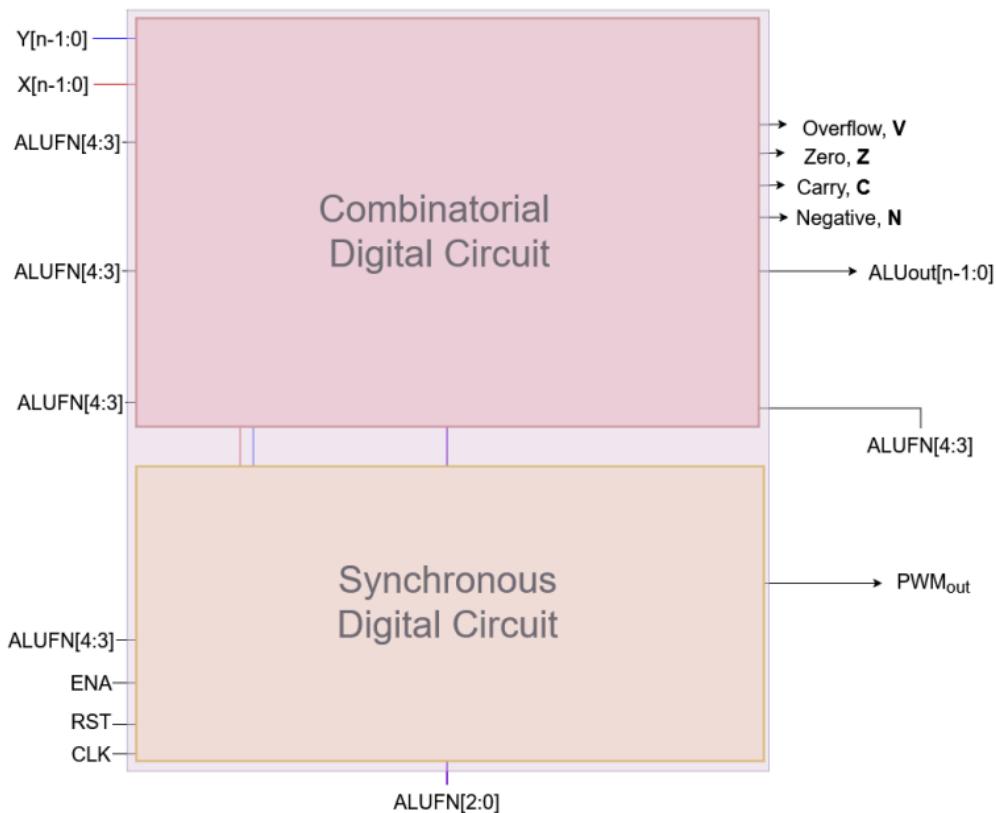
במעבדה זו נלמד לעבוד בתוכנת Quartus כך שביצע סינטזה של קוד VHDL ונctrוב אותו על FPGA מודגם DE2 - 115.

פירוט המערכת

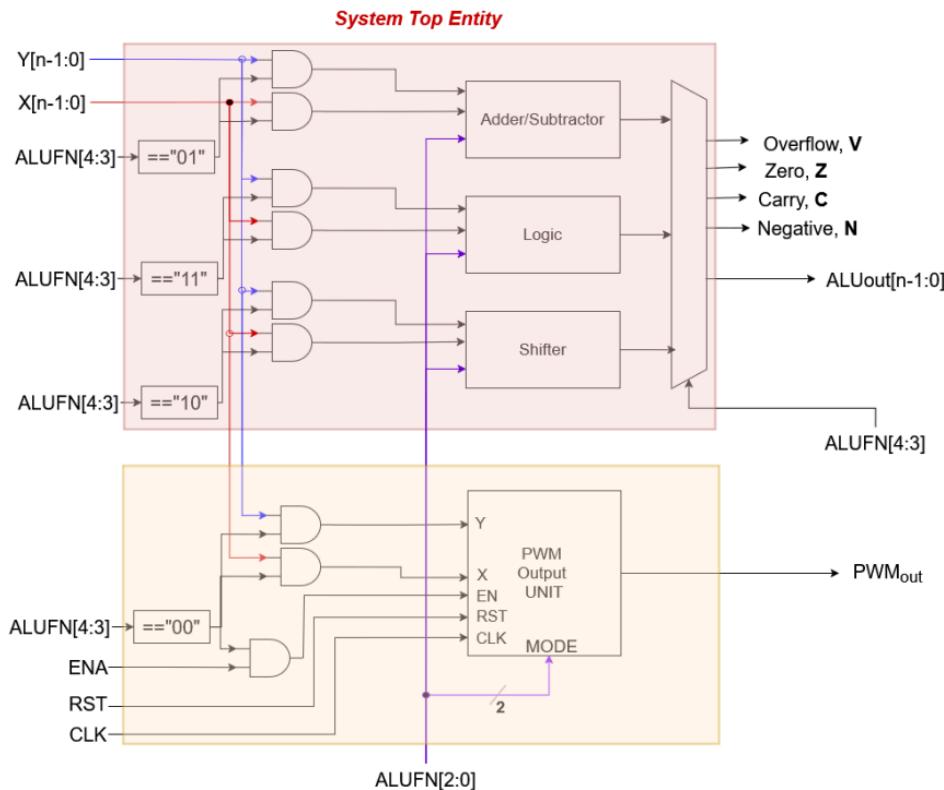
Digital System

הרכיב הדיגיטלי מורכב משני תתי רכיבים: CDC, SDC. הוא מקבל כניסה וקטרוי X - Y עליהם יבוצעו הפעולות, ALUFN שמאגדיר את הפעולה הנדרשת, שעון, RST לאיפוס המערכת, ENA שמאפשר את ריצת גל ה- PW. המוצאים יהיו FLAG תוצאה ה- ALU והגל .PWNOUT

System Top Entity



רכיב ה- **CDC** (הוורד באיר) כולל בתוכו את תת-רכיבים שבנינו במעבדה 1: ADDERSUB, LOGIC ו-SHIFTER שזה בעצם ה-ALU. השינוי היחיד הוא שהוספנו את פעולה SWAP בתוך רכיב ה-ALU.



רכיב ה- **SDC** (הצהוב באIOR הקודם) בעצם מוציא גל PWM שתלי' במנועה שמתאפס בהגעתו ערך ז' ובעל 3 מצבים:

- מצב 0: המוצא יהיה 1 כאשר המוניה בין X ו- Z, יהיה 0 אחרת.
- מצב 1: המוצא יהיה 0 כאשר המוניה בין X ו-Z, יהיה 1 אחרת.
- מצב 2: בהגעת המוניה לערך X ערך המוצא יתהפך (מ-0 ל-1 ולהפך).

Figure 4: Digital System subparts architecture

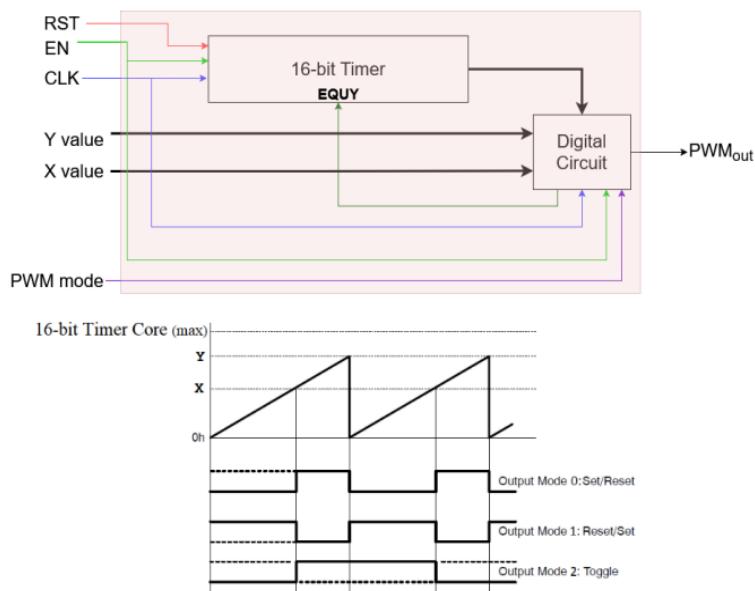


Figure 5: PWM output unit architecture

TOP

רכיב ה- TOP כולל בתוכו את ה- DIGITAL SYSTEM וגם את ניהול הכניסות והיציאות. המטרה היא ליצור קשר בין ה- INPUT של המשתמש מה- FPGA זהה נעשה ע"י הכנסת ערכיהם ל- ALUFN , X , Y , Z . בעזרתו מיתוג ה- SWITCHES וקבעת האם הכניסה הינה 8 הביטים התחטוניים אוعلילוניים בעזרת SW9. המוצאים יתממשקו בסופו של דבר אל מסכי ה- HEX ונורות ה- LED.

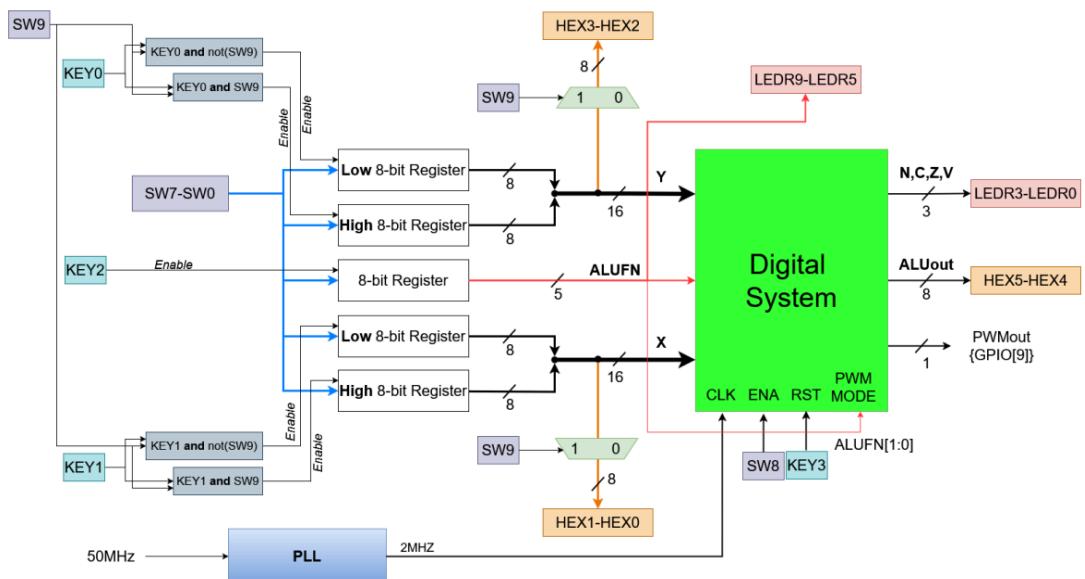


Figure 6: Digital system with I/O interface

בנוסף השתמש במאיץ PLL לצורך הגברת מהירות השעון שיכל לספק לנו הגביש המובנה ב- FPGA.

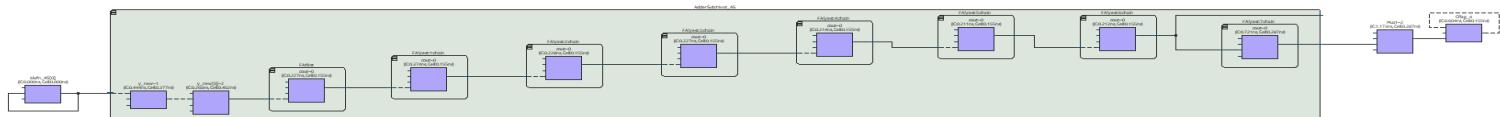
נתיב קרייטי:

הנתיב הקרייטי מגדר את הזמן הארוך ביותר שדרוש לאות לעبور בין שני איזורים במחזור שעון אחד וכך קובע את תדריות השעון המקסימלית. אם הנתיב הקרייטי ארוך מדי, יתכנו הפרות של תנאי ה-Hold/Setup ברישומים, והתפקוד ישתבש.

כדי ליעיל נהוג לקצר אותו באמצעות הוספת רישומים ביןיהם (pipelining) או שימוש ברכיבים חומרתיים ייעודיים. חשוב לשמור על נתיב קרייטי קצר כדי ליצור מערכת מהירה, אמינה ועילה מבחינה צריכת משאבים וחשמל.

אצלנו בעבודה כדי למצוא את הנתיב הקרייטי סמן 2 רגיסטרים שייעטו את המערכת, וכן למצוא את הנתיב הארוך ביותר ביניהם.

להלן הנתיב הקרייטי בSTA :



מציאת תדר מקסימלי:

נדרשו למצוא תדר מקסימלי של המערכת , لكن שמו 2 רגיסטרים שיעטפו את ה`ALU` מהכניסה והיציאה.

לאחר מכון קימפלנו וה`QUARTUS` הבאנו את התדרים הבאים בمعالות: 0 ו- 85 צלזיו.

- התדר המקסימלי בכל טבלה הוא השני .

Slow 1200mV 85C Model Fmax Summary			
	Fmax	Restricted Fmax	Clock Name
1	67.79 MHz	67.79 MHz	DigitalSystem:portmap_top ALUFN_i_s[3]
2	92.85 MHz	92.85 MHz	hivutPLL altpll_component auto_generated pll1 clk[0]

Slow 1200mV OC Model Fmax Summary			
	Fmax	Restricted Fmax	Clock Name
1	74.67 MHz	74.67 MHz	DigitalSystem:portmap_top ALUFN_i_s[3]
2	102.29 MHz	102.29 MHz	hivutPLL altpll_component auto_generated pll1 clk[0]

signal tap

ביצענו וריפיקציה של ה- FPGA על ידי Quartus. השימוש של signal tap. המטרה של הפעולה היא לבדוק את הסיגנלים של המערכת בזמן אמת.

נדרשנו לבדוק את פעולה אחת אריתמטית ואחת של shift.

לכל אחת מהפעולות נבדוק את :

Type	Alias	Node	Name
C		+ DigitalSystem:portmap top X [15..0]	
C		+ DigitalSystem:portmap top Y [15..0]	
R		+ ALUFN[4..0]	
C		+ DigitalSystem:portmap top ALUout o[7..0]	

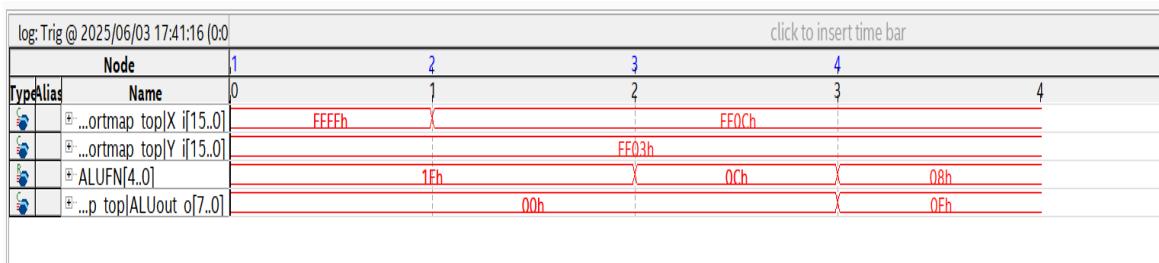
כלומר על המערכת נכנס את X ו Y באמצעות לחיצה על switches הנכונים.

נשים לב שבפעולות האלו מה שרלוונטי אילו רק 8 הביטים ה sbs של X ו Y.

את X נטען על ידי לחיצה על ה switches ובסוף ש 9 switch מכובה , לאחר מכן נלחץ על 0.key.

את X נטען על ידי לחיצה על ה switches ובסוף ש 9 switch מכובה , לאחר מכן נלחץ על 1.key.

פעולה ADD :



* ההערה: הערך ההתחלתי של X לפני הטעינה של הערך היה הערך FFFF.

טענו ל X את הערך FF0C ולי Z את FF03 . * הערה : מה שרלוונטי זה 8 ביטים LSB קלומר 2 הספרות הימניות.

לאחר מכן כדי להשתמש בADD לחצנו על switches לקלבל את הערך 8 ולחצנו על 3.key.

ואכן ניתן לראות שבתוצאה ביצוג 8 ביטים קיבלנו F0 בhex כדריש.

פעולה : shift left

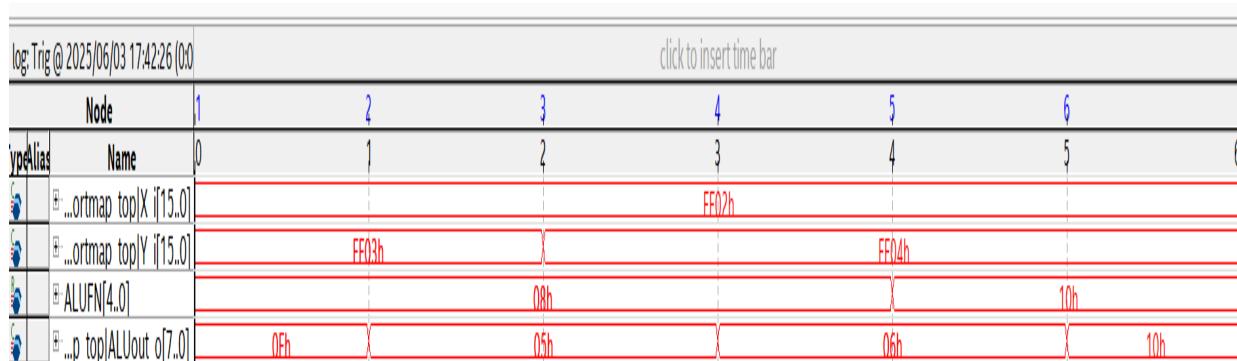
ביצענו את הפעולה shift left שהערך שלו הוא 16 .

מטרת הפעולה היא להזיז את Z ב-X פעים שמאלה.

טענו ל X את הערך FF02 ולי Z את FF04 . * הערה : מה שרלוונטי זה 8 ביטים ה- LSB קלומר 2 הספרות הימניות.

לאחר מכן כדי להשתמש בshift left לחצנו על switches לקלבל את הערך 16 ולחצנו על 3.key.

ואכן ניתן לראות שבתוצאה ביצוג 8 ביטים קיבלנו 05 בhex כדריש.

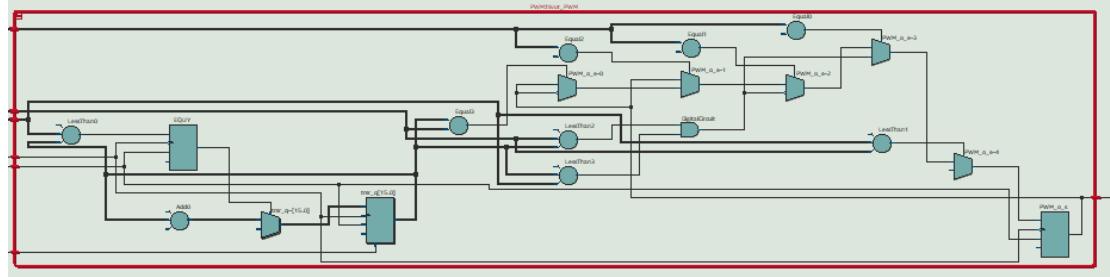
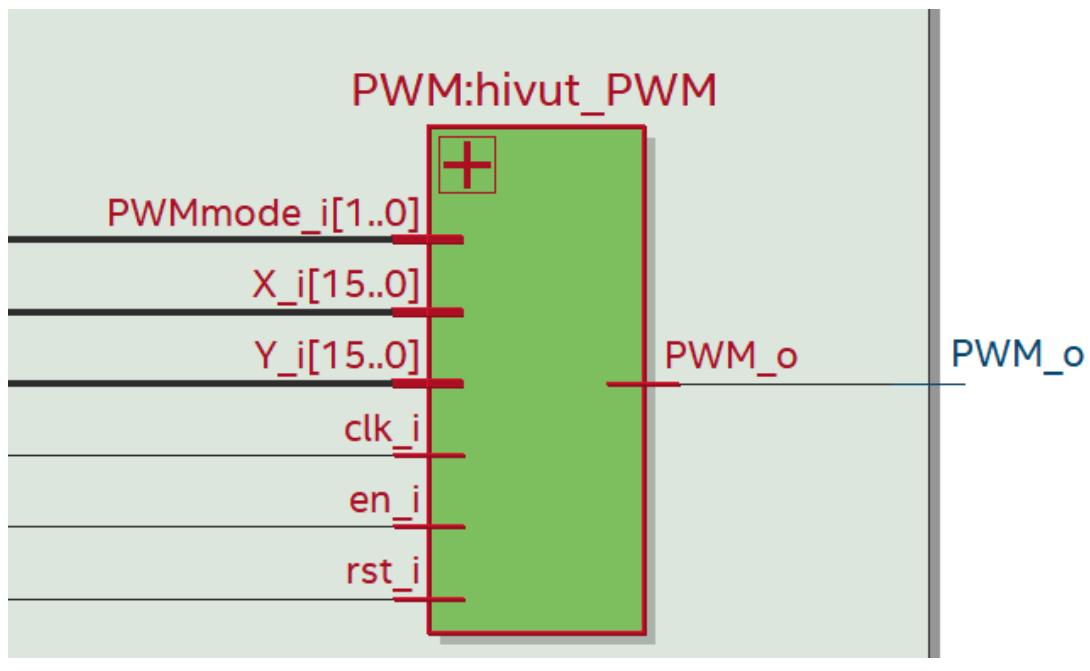


שרטוטי RTL

ב- Quartus צפינו בIMPLEMENTATION הchoicer של הקוד שכתבנו בVHDL על ידי .RTL VIEWER

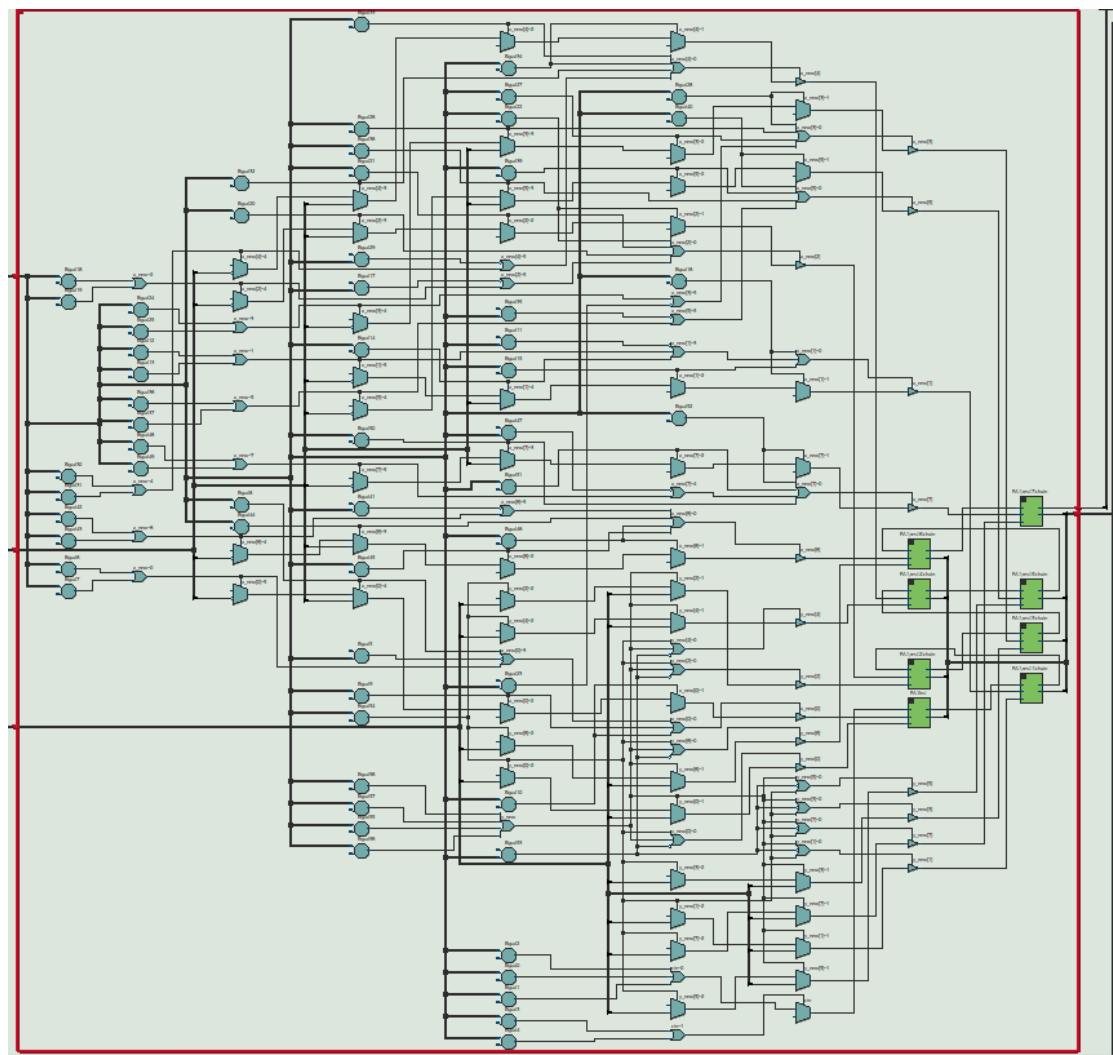
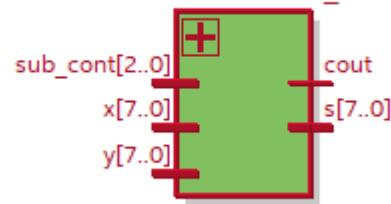
לכל מודול נראה את entity שלו ושרטוט ה- RTL :

PWM:

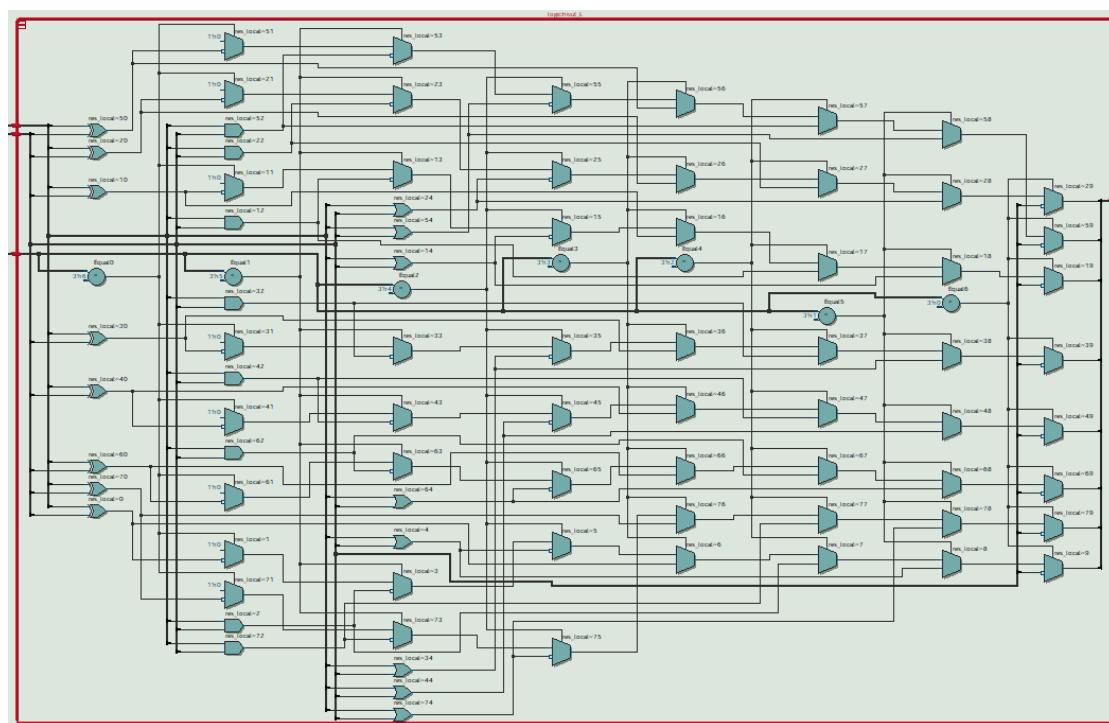
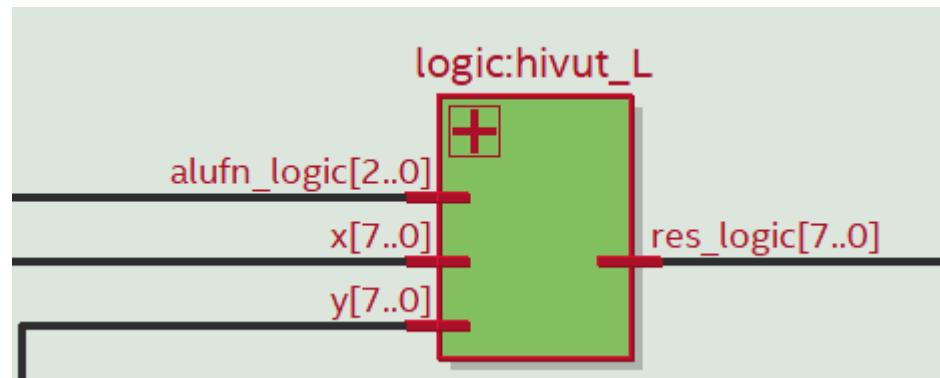


ADDERSUB

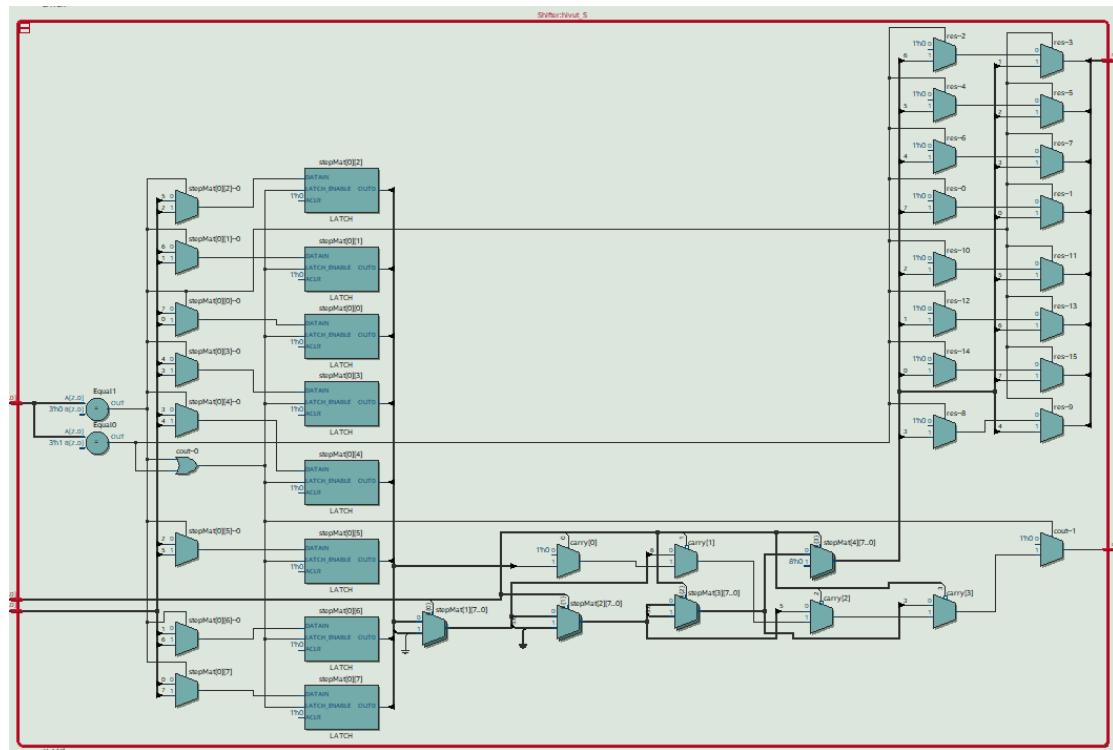
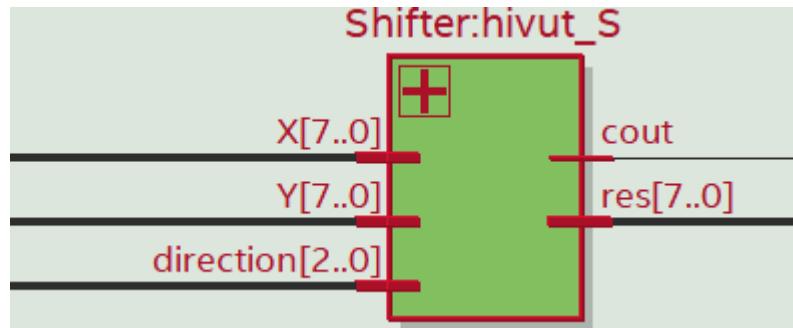
AdderSub:hivut_AS



LOGIC



SHIFTER



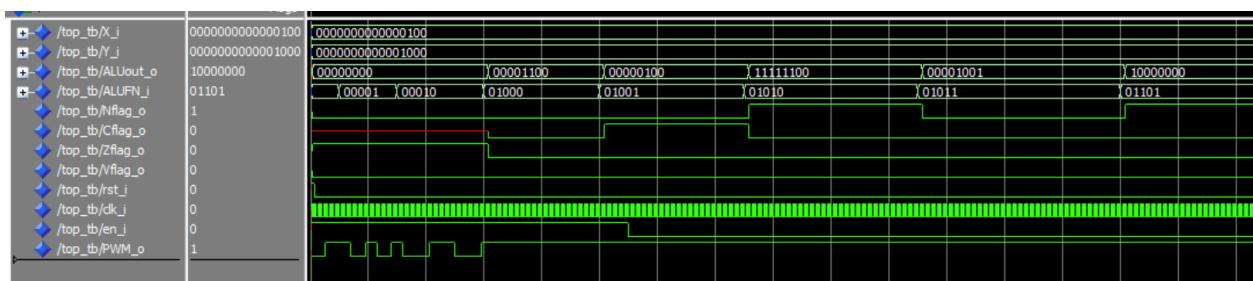
סימולציה modelsim:

צרכנו קבצי testbench , כדי לבדוק תחילה את המערכת שיצרנו לפני המעבר לחומרה.

עשינו testbench לכל תת מודול כדי לבדוק אותו.

טוטל כללי של ה- SYSTEM TOP ENTITY :

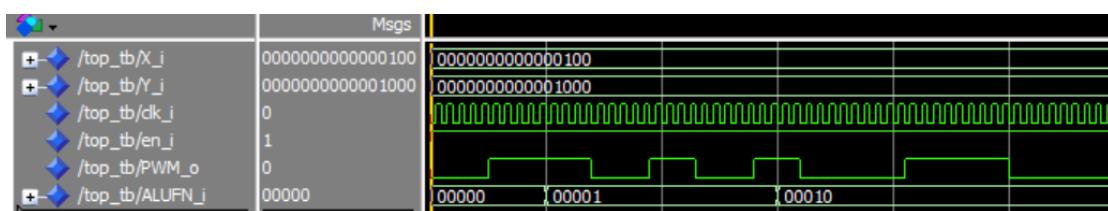
ניתן לראות במקביל תוצאות של גלים של ה- זום ובנוסף תוצאות של ALU עם דגלים שיוצאים בהתאם. כולם שילוב של 2 המערכות .sdc | cdc



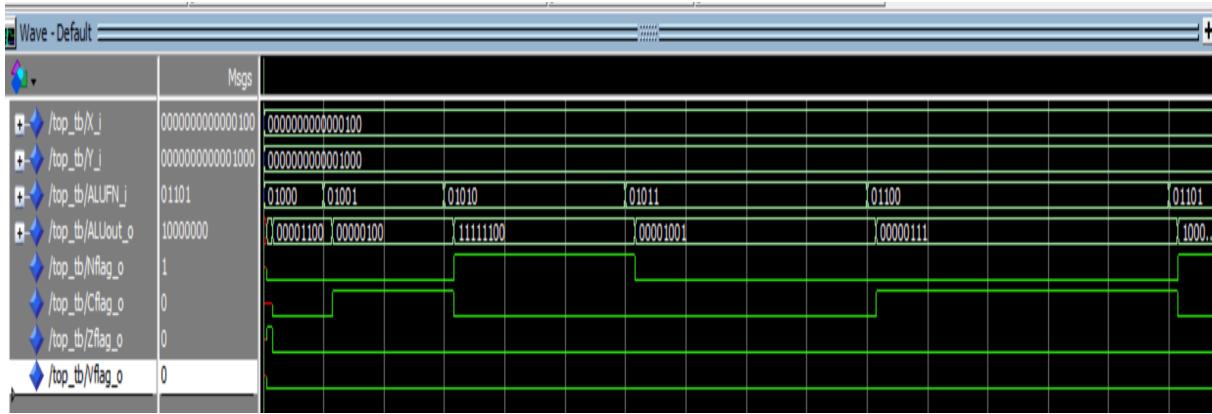
סימולציה של pwm:

ניתן לראות את הגל המתאים בהתאם לALUFN , לפי הסדר:

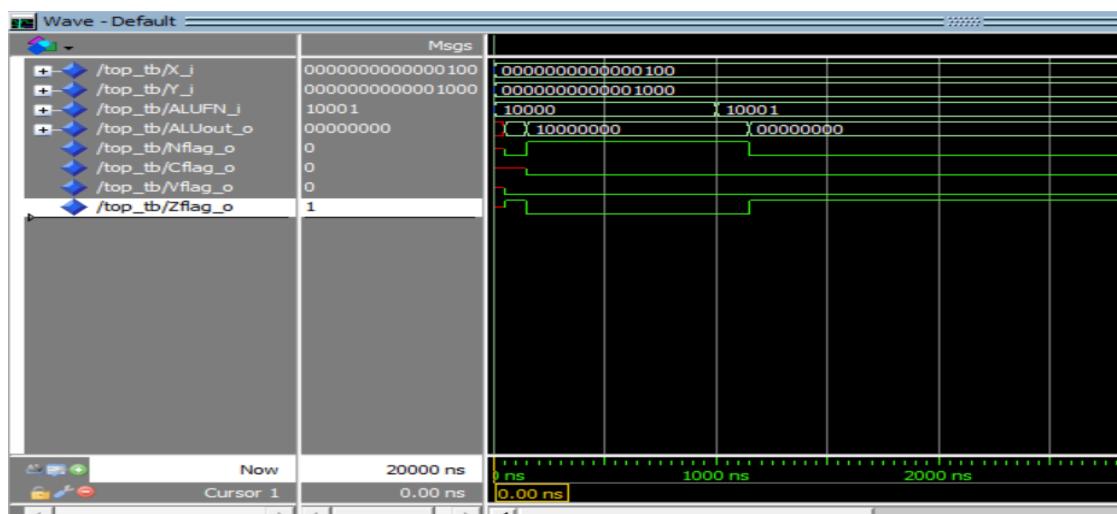
mode 0, mode 1, toggle (mode 2)



סימולציה של addersub:
שמננו קקלט את המספרים 4 ו 8 , ועברנו על כל הפעולות האריתמטיות.



סימולציה של shifter:
שמננו קקלט את המספרים 4 ו 8 , ועברנו על כל הפעולות ההזזה.



סימולציה logic

שמננו קקלט את המספרים 4 ו 8 , ועברנו על כל הפעולות הלוגיות

