

LAB 4 REPORT

ADVANCED CPU ARCHITECTURE AND HARDWARE

FPGA based Digital Design

Tal Adoni – 319087300

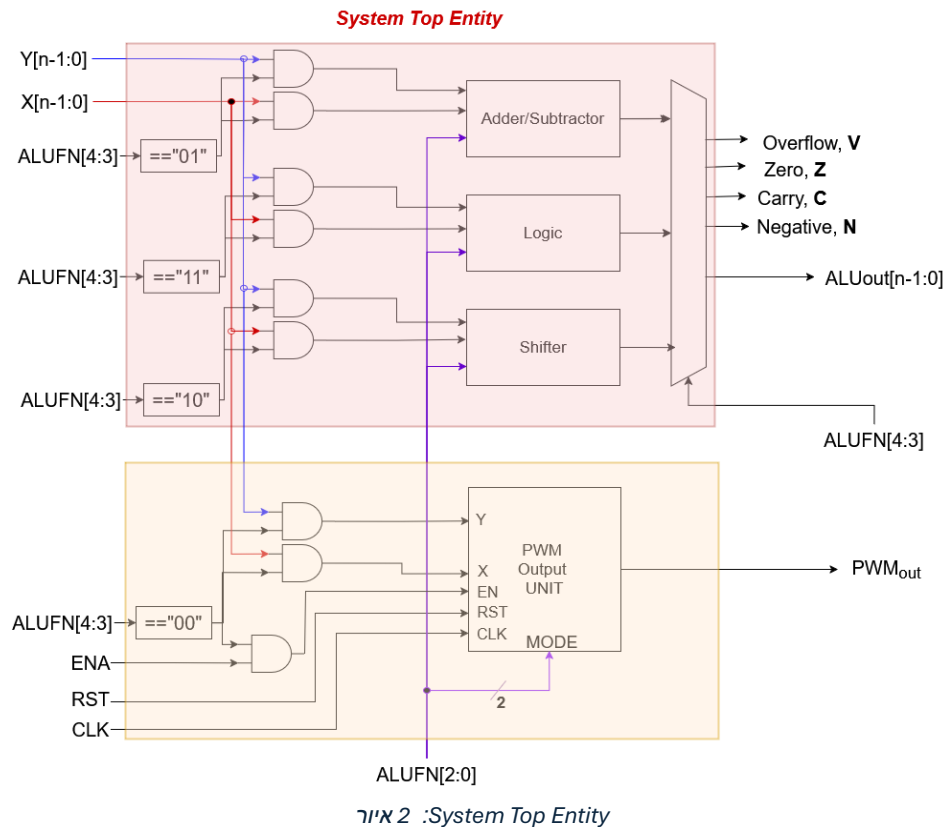
Omri Aviram – 312192669

במעבדה זו היה עלינו להיעזר במערכת שאותה בנינו במעבדה 1, יחד עם מספר פונקציות ומצבים חדשים, וצורב אותה לערכת מעבדה של *FPGA* שניתנה לנו – *DE10 – Standard*, בהתאם לפקודות המצורפות:

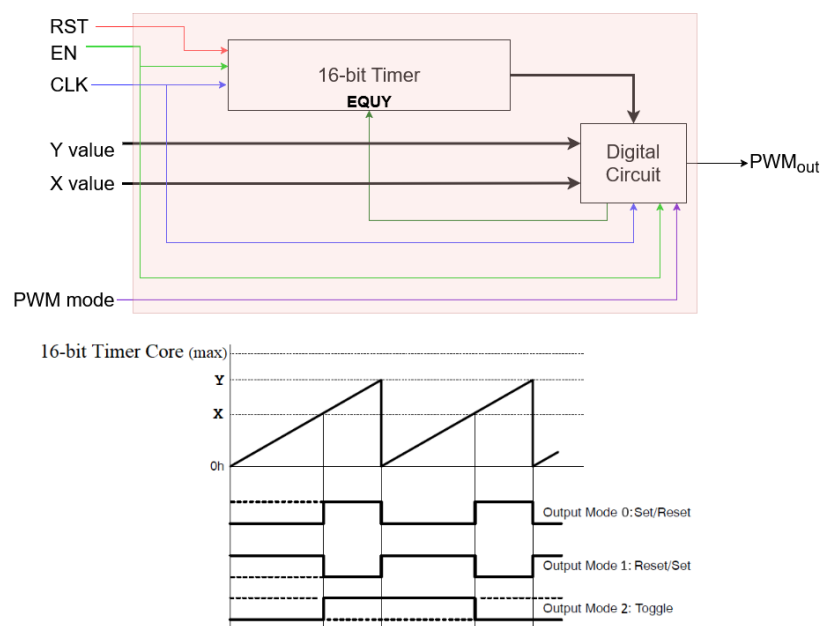
Function Type	Decimal value	ALUFN	Operation	Note
PWM Output (<i>Y and X are 16-bit width</i>)	0	00000	PWM MODE0	PWM Mode is Set/Reset
	1	00001	PWM MODE1	PWM Mode is Reset/Set
	2	00010	PWM MODE2	PWM Mode is Toggle
Arithmetic (<i>Y and X are 8-bit width</i>)	8	01000	Res=Y+X	
	9	01001	Res=Y-X	Used also for comparison operation
	10	01010	Res=neg(X)	
	11	01011	Res=Y+1	Increment of Y in one
	12	01100	Res=Y-1	Decrement of Y in one
	13	01101	Res=swap(Y)	Res=(YLSHW, YMSHW)
Shift (<i>Y and X are 8-bit width</i>)	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 (<i>Y and X are 8-bit width</i>)	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	11110	Res=Y xnor X	

איור 1:Instructions

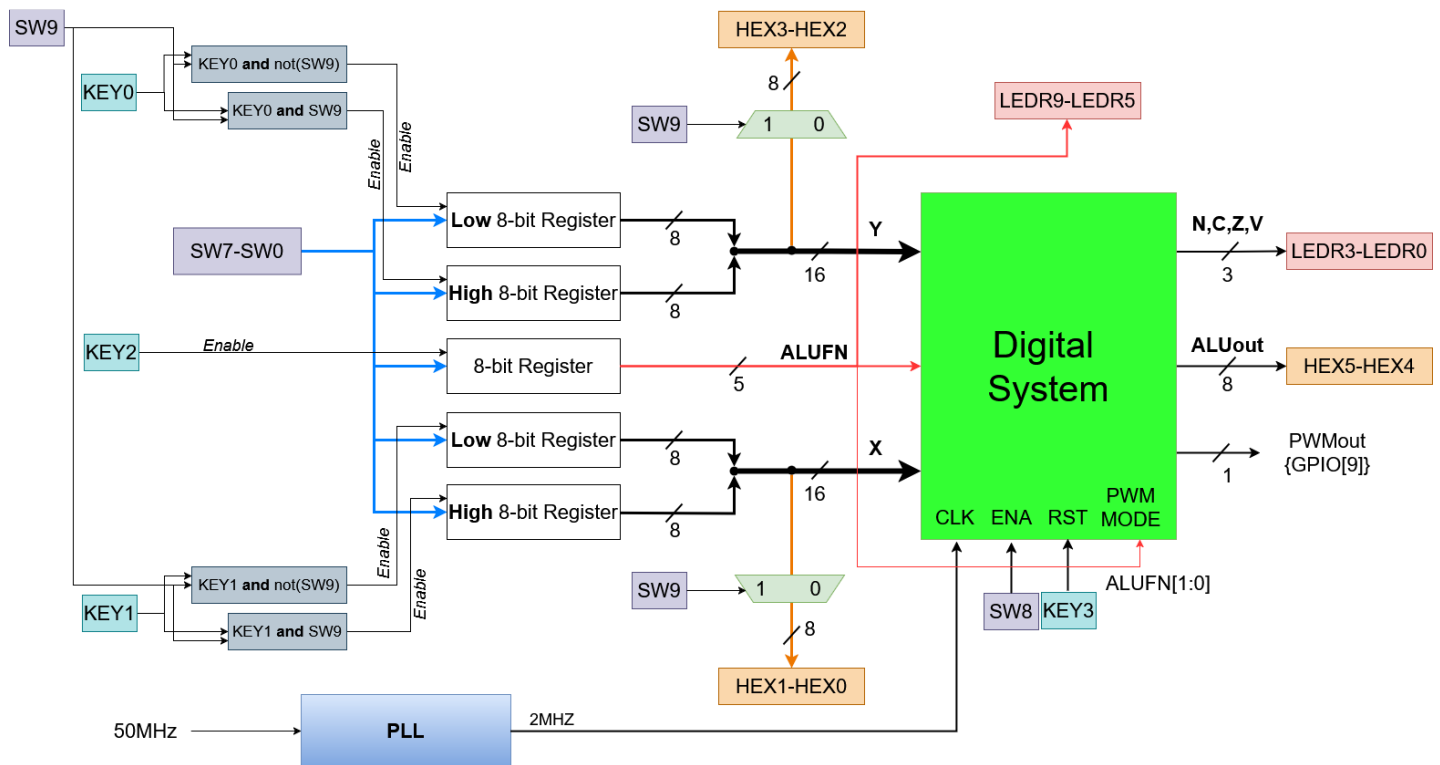
תחילה סידרנו את יחידת ה-*Combinational Digital Circuit* אשר מטפלת בפעולות על X ועל Y והוספנו את פעולת ה-*Swap* החדשה אל היחידה האריתמטית ולאחר מכן התחלנו לבנות את היחידה ה-*Synchronous Digital Circuit* שבה היה עלינו לבצע ספירה ולפי תרשים להוציא אות PWM מתאים. היחידות במערכת יחד עם קווים :



כפי שצינו, ברכיב הסינכרוני שלנו היה עלינו להוציא אות PWM בהתאם לזמן שעון פנימי, לכן מימשנו שעון אשר מקבל אות מתי עליו להתאפס ולבצע ספירה חזרה בנוסף לקווי הבקרה הנתונים כפי שמתואר באיור המצורף. בכדי למנוע ספירה מחדש מימשנו את הרכיב כך שבזמן אמת שלושת האותות המבוקשים נוצרים אך אנו נעזרים בבורר אשר בוחר לפי ה- $PWM\ mod$ איזה אות להוציא :



לאחר מכן התחלנו בהגדרת המעגל יחד עם הרכיבים הפיזיים של הערכה באופן דיגיטלי, זאת בכדי שנוכל לקשר אותם בעת עבודה על ה-FPGA, לפי הגדרות העבודה:



איור 4: Digital system with I/O interface

בשונה מרכיבים אחרים, עבור תצוגת ה-HEX משום שיש קידוד פנימי להדלקת הנורות הגדרנו Decoder שבהתאם למספר המתקבל בקו שנכנס אל המנורה או מבצעים קידוד מתאים לצורך הצגת המספר הנכון על גבי הצג.

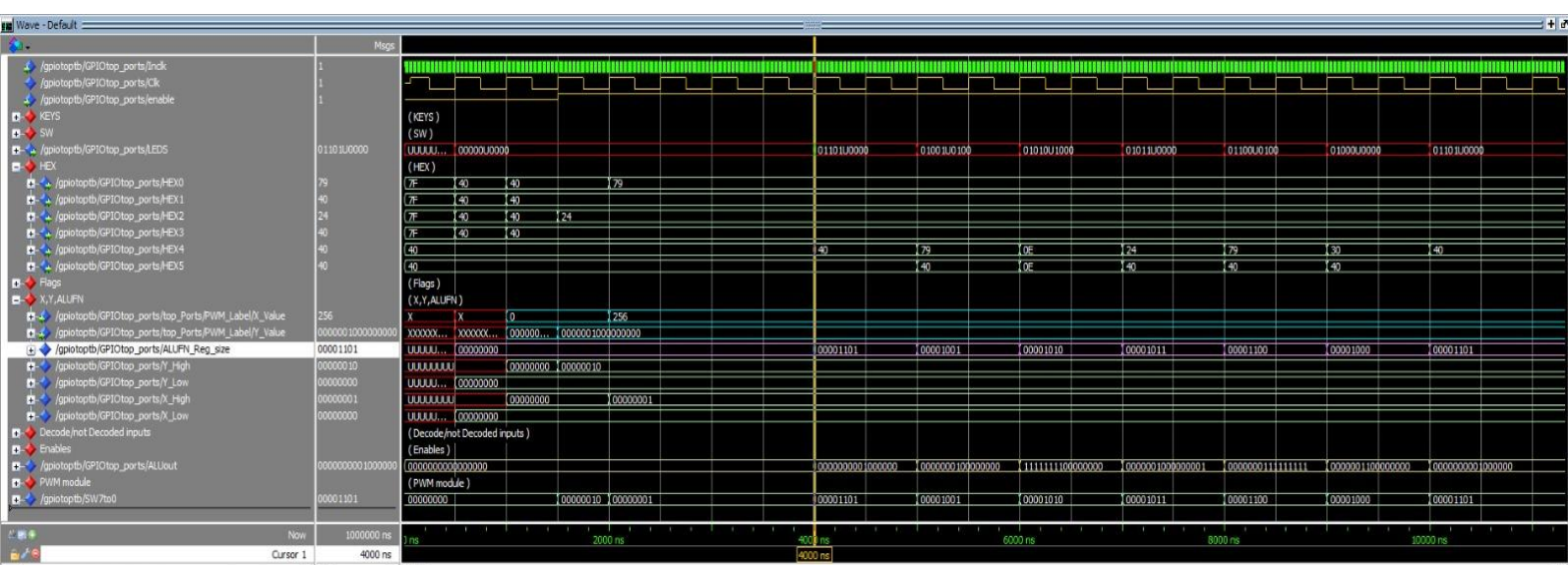
לבסוף משום שאנחנו עובדים על רכיב פיזי בנוסף לבדיקת הגלים בתכנת ה-ModelSIM היה עלינו לבדוק את עבודת הקוד בערכת ה-FPGA עליה אנו עובדים, זאת לאחר עדכון וחיבור הערכים המתאימים לפי ה-User Guide כפי שמצוין בהמשך העבודה.

תצוגת סימולציה ModelSim :

כעת נציג תרשים גלים של המודול יחד עם שמות הסיגנלים המתאימים לחומרה (KEYx, SWx...) ונראה כי הפעולות אכן פועלות כצפוי :

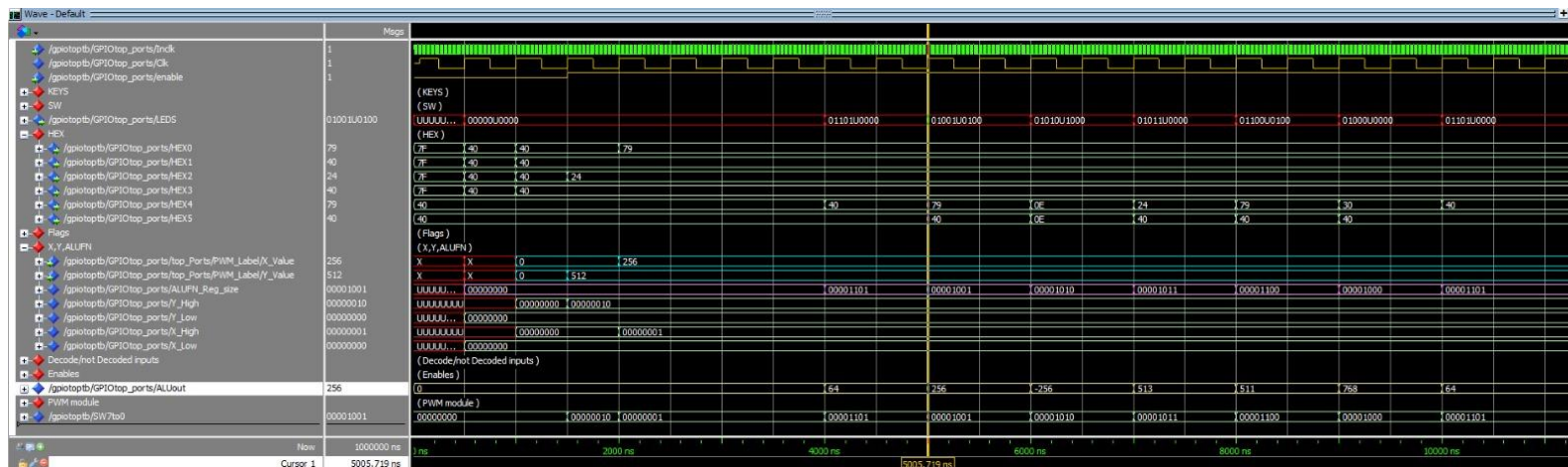
תחילה ביצענו פעולת swap(y) אשר מבצעת החלפה של כל הספרות הבינאריות של המספר y ומתקבל פלט הפוך מהכניסה.

בתחילת התהליך רואים כי טרם הוכנסו ערכים למערכת ולאחר הכנסה של הערכים אנו רואים כי בזמן 4000 ns היכן שה-cursor נמצא אנו משנים את ה-ALUFN לפונק' SWAP כפי שצינו וכפי שניתן לראות ALUout אכן מוציא ערכים הפוכים מהכניסה של y :



איור 5: פעולת SWAP ב-ModelSim

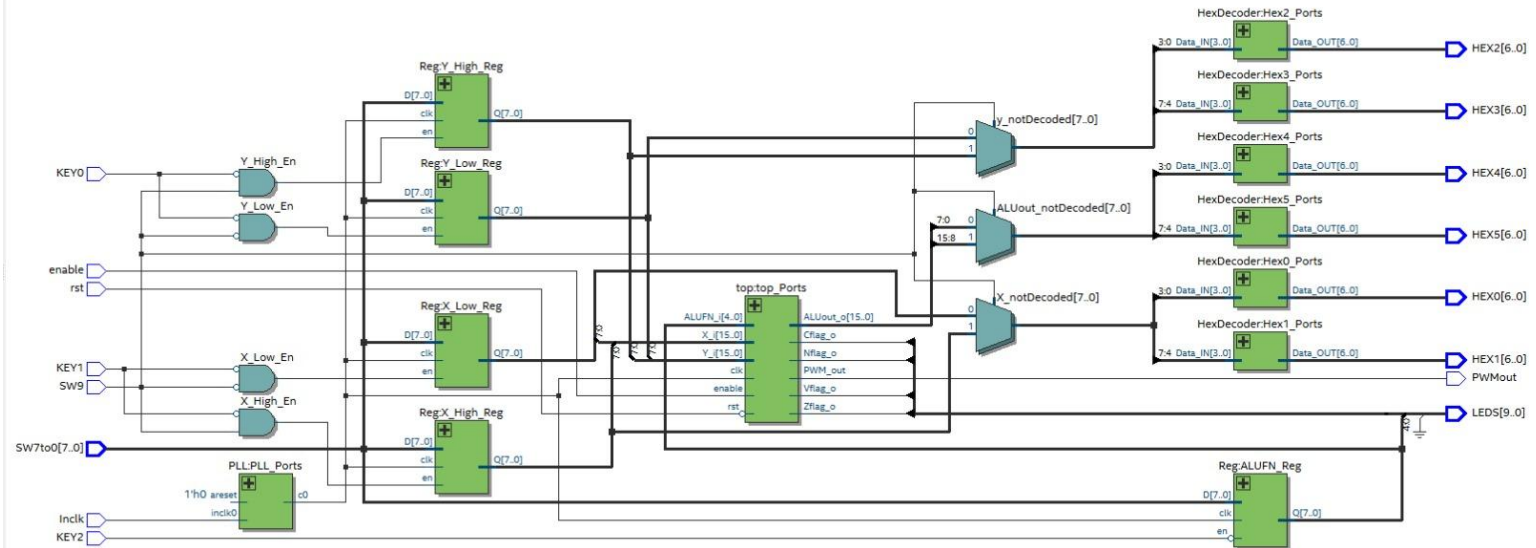
כעת בפעולה השנייה ביצענו פעולת חיסור Y-X אשר והפלט שלנו מתקבל במוצא ALUout. בתחילת התהליך רואים שוב כי טרם הוכנסו ערכים למערכת ולאחר הכנסה של הערכים אנו רואים כי בזמן 5000 ns היכן שה-cursor נמצא אנו משנים את ה-ALUFN לפונק' Sub כפי שצינו וכפי שניתן לראות ALUout אכן מוציא את ההפרש :



איור 6: פעולת חיסור ב-ModelSim

עבודה על תוכנת Quartus :

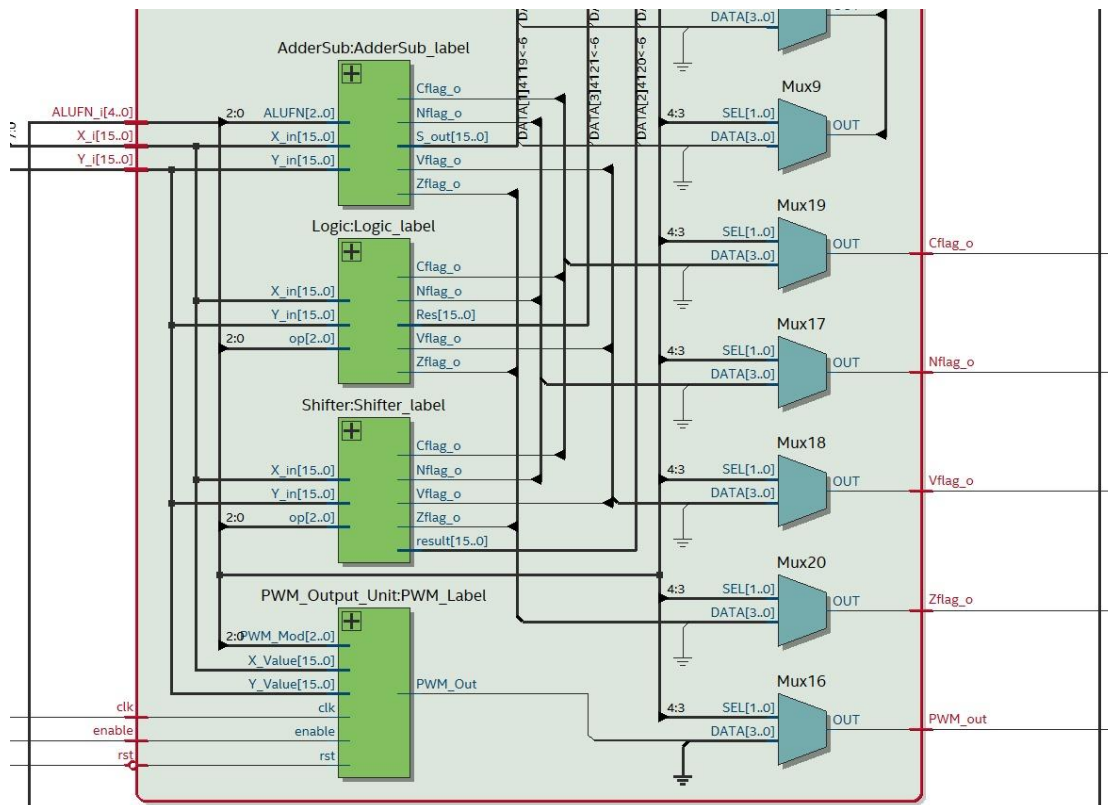
כחלק מהפונק' שיש לתוכנת ה-Quartus בעזרת RTL viewer אפשר לקבל דיאגרמת בלוקים של המערכת לפי המימוש שביצענו באופן הבא :



איור 7 :RTL View - GPIOtop

בתמונה אנחנו רואים את אופן החיבור של המערכת, שבאופן לא מפתיע דומה לאיור ומציגה את אותה הפונקציונליות כאשר כל אחד מהפינים בסופו של דבר מקושר לכניסה ומוצא מתאימים על גבי ה-I/O ב-FPGA.

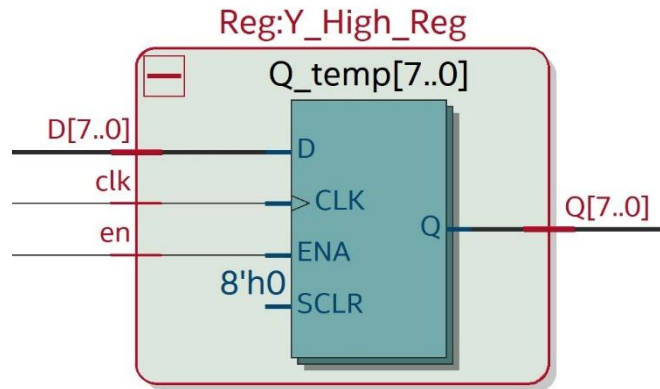
כאשר אנחנו לוחצים על + באחד המלבנים ניתן לראות את יחידה המתוארת, למשל בלחיצה על יחידת ה-Top במרכז המעגל אנו מקבלים את המבנה הפנימי של יחידת ה-top במעגל :



איור 8 :RTL View - Top

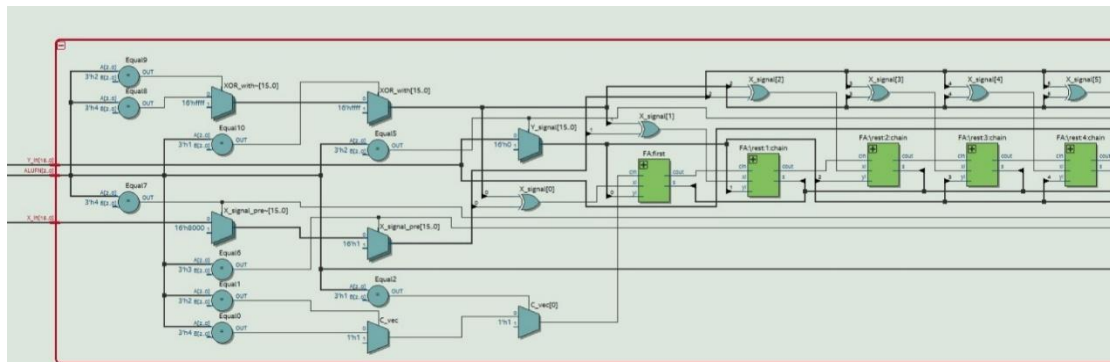
באותו אופן ביצענו את הפעולה לכל רכיבי המעגל.

Reg – נציין כי כלל הרגיסטרים שמופיעים ב-GPIOTop בנויים באופן הבא:



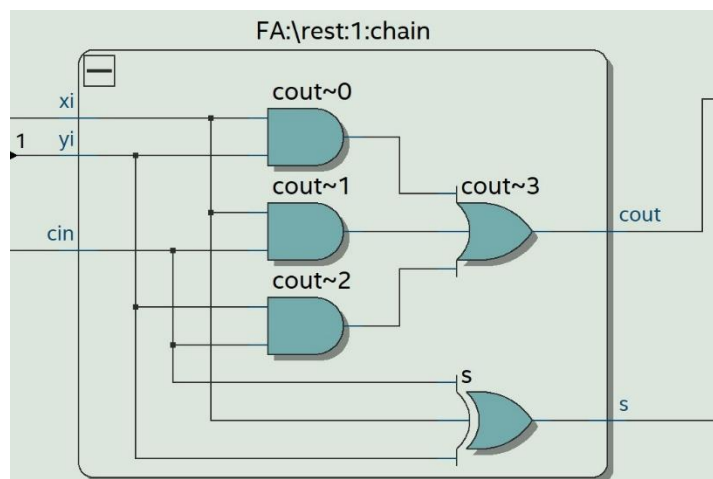
איור 9: Reg

Adder Subtractor – במודול זה אנחנו שרשרנו FA אחד לשני בהתאם לכניסות כפי שניתן לראות בתמונה וכך מתקיים בהמשך שלה מימין: כדי לשמור על תמונה מובנת חתכנו את התמונה



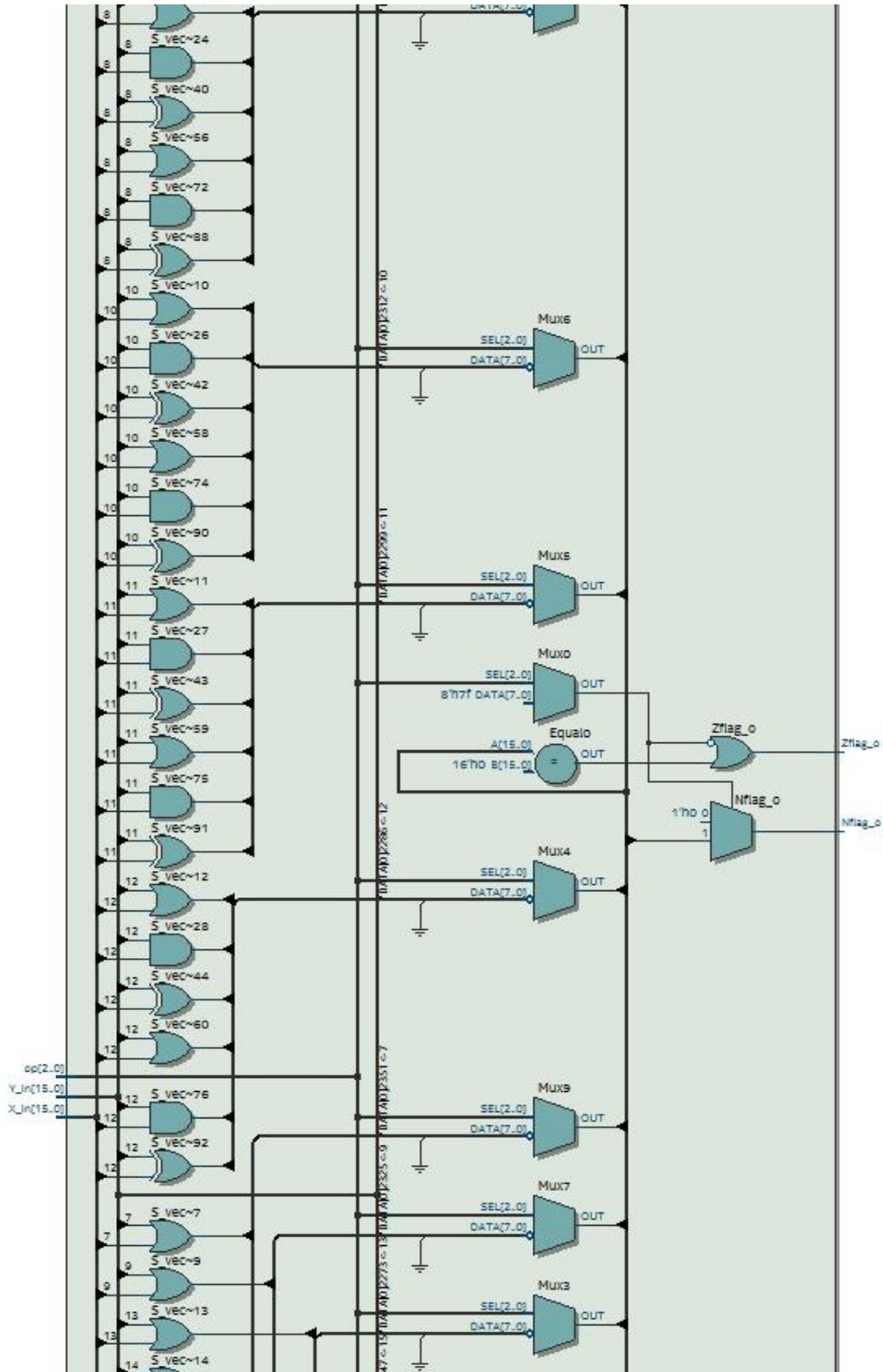
איור 10: Adder Subtractor

– FA

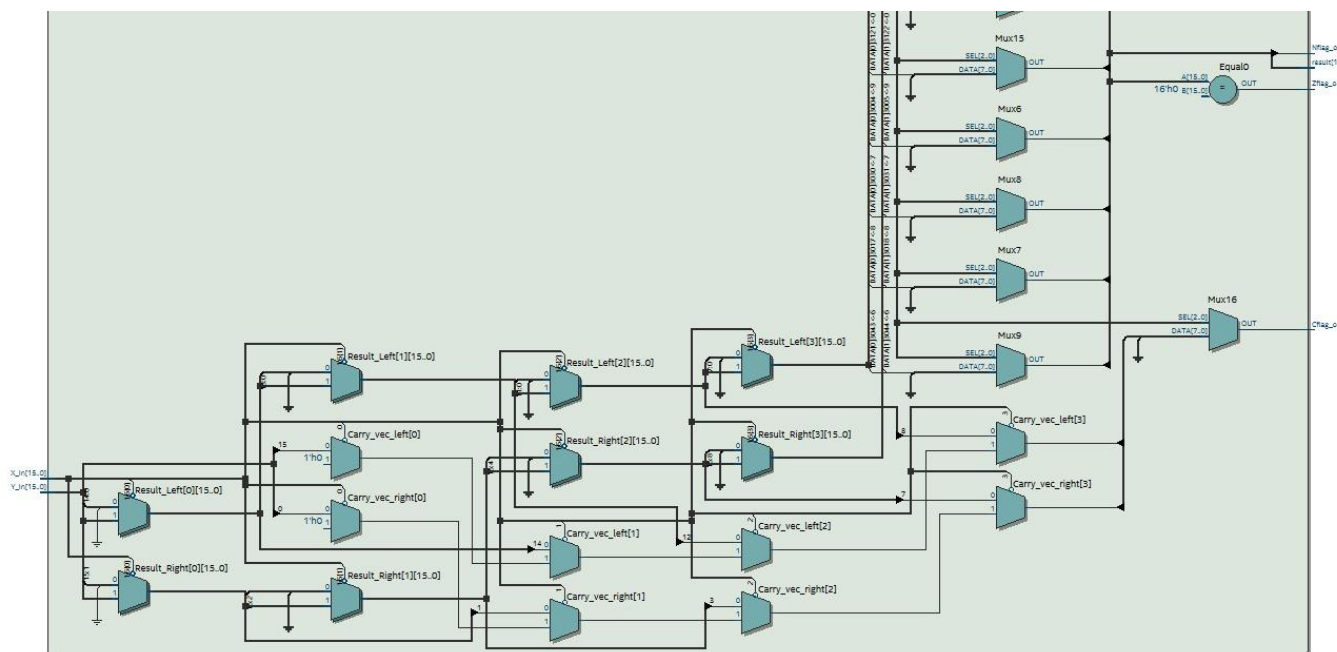


איור 11: FA

Logic – במודול זה יש לנו עוד שערים לוגיים מתחת ומעל התמונה.
כדי לשמור על תמונה מובנת חתכנו את התמונה.

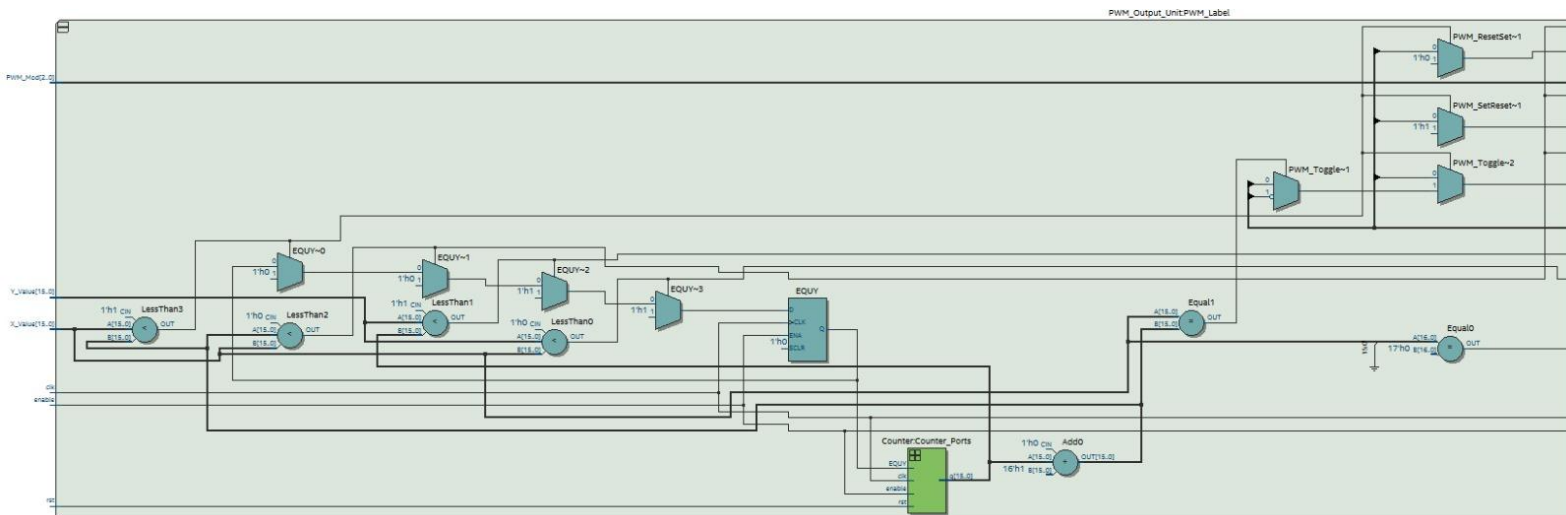


Shifter – במודול זה יש לנו עוד שערים לוגיים מתחת ומעל התמונה.
כדי לשמור על תמונה מובנת חתכנו את התמונה.



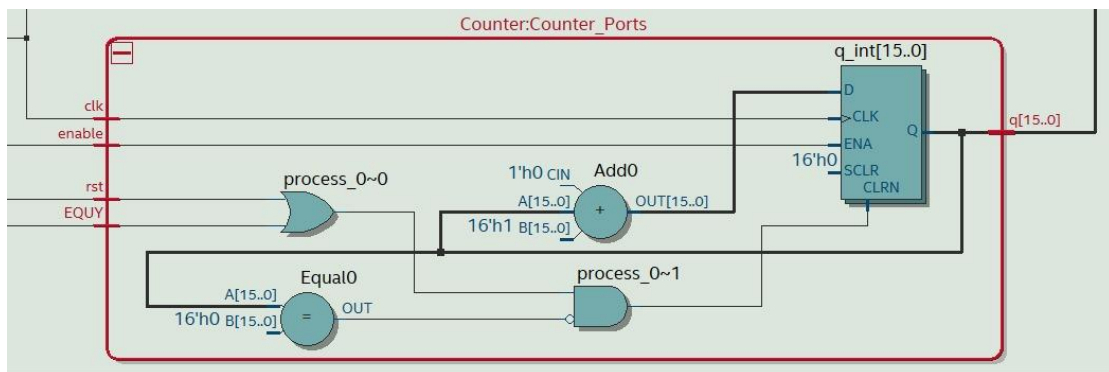
איור 13: Shifter

PWM Unit – בצד שמאל של התמונה ישנם שלושה בלוקים אשר מוציאים גל בהתאם למצב כפי שצינו קודם (Reset/Set, Set/Reset, Toggle) ו-MUX שבורר ביניהם.
כדי לשמור על תמונה מובנת חתכנו את התמונה.



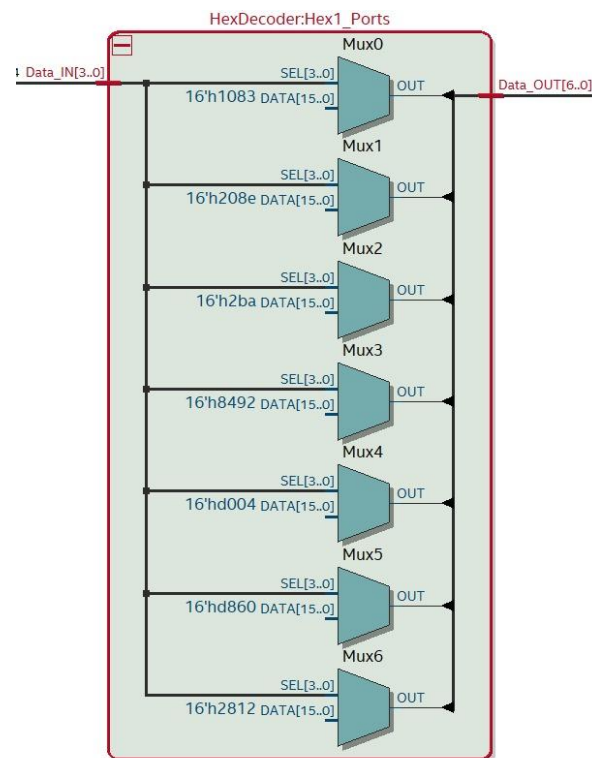
איור 14: PWM Unit

Counter – נמצא ביחידת ה-PWM Unit.



איור 15 :Counter

HexDecoder – נציין כי כלל ה-Decoders שמופעים ב-GPIOTop בנויים באופן הבא :



איור 14 : HexDeco

לאחר הצגת הרכיבים, נבצע כעת תצוגה של התדר המקסימלי (f_{max}) וזאת על ידי השמת יחידה א-סינכרונית בין שני רכיבים סינכרוניים, במקרה שלנו ניעזר ברגיסטרים :

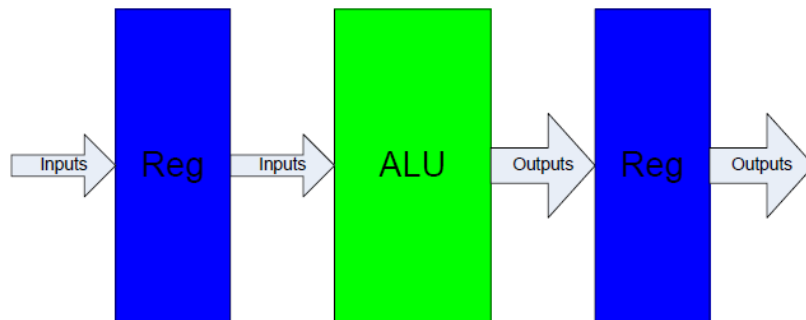
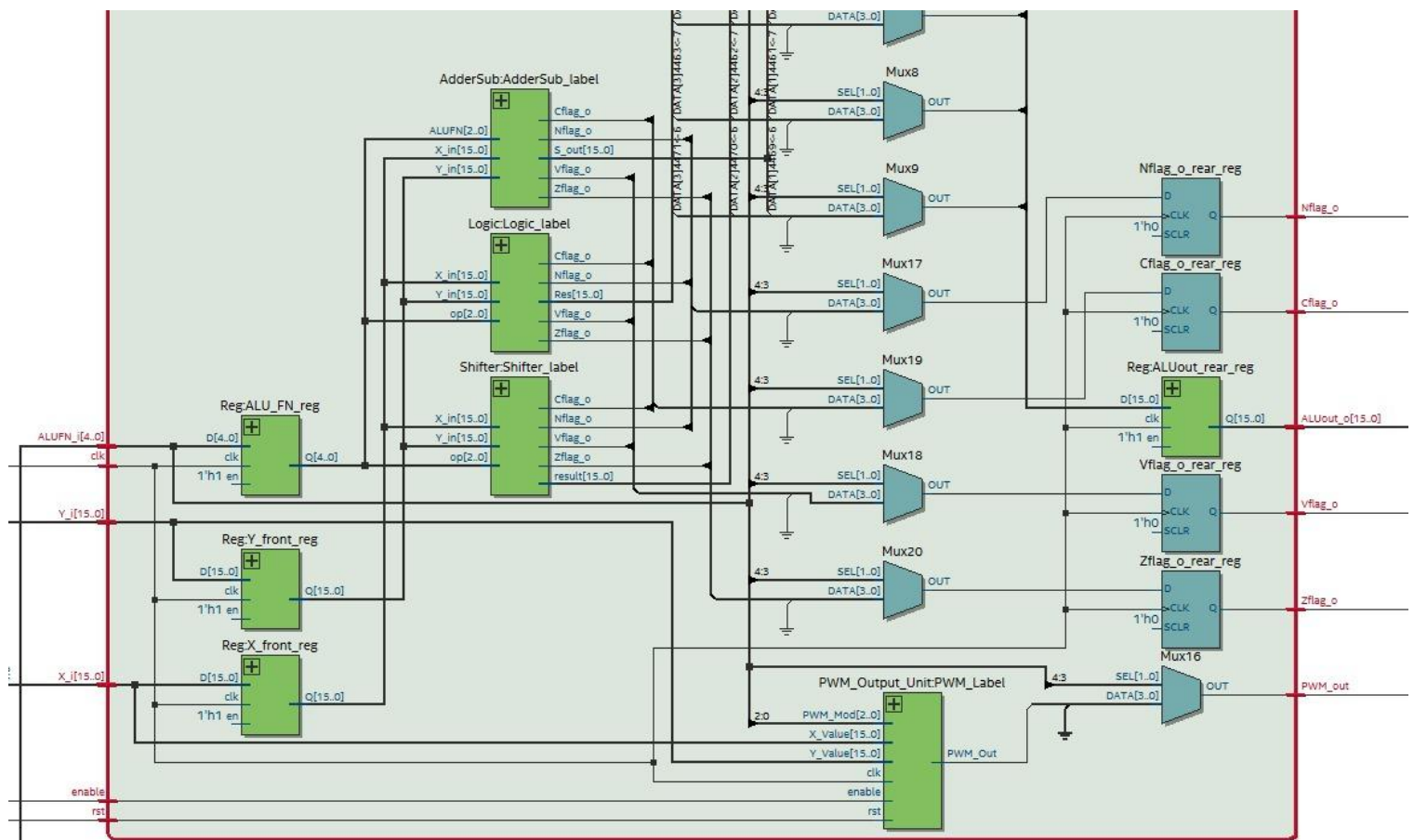


Figure 1 : Test Case in case of pure logic system as ALU

F_{max} : מציאת 16 איור

הרכיבים הא-סינכרוניים שלנו במעגל הם ה-Logic, AdderSubtractor, Shifter שכן כל שאר חלקי המעגל הינם עטופים ברגיסטרים ושעונים, לכן לצורך מדידת F_{max} שלהם הוספנו להם רגיסטרים בכניסה (Front Reg) וביציאה שלהם (Rear Reg) עבור הכניסות X, Y, ALUFN שלהם והמוצאים ALUout ולכל אחד מהדגלים.

מצורף סרטוט של החומרה יחד עם רגיסטרים :



איור 15: Logic, AdderSubtractor, Shifter

מציאת f_{max} :

על מנת למצוא לבצע אופטימיזציה של הקוד ולקבל תדר טוב יותר, לכן ננסה להימנע Latch בקוד ובנוסף נשים לכניסות ולמוצאים את הפינים הקרובים ביותר שהתוכנה מציעה ב - location fitter.

לאחר ביצוע השלבים נרמפל את המודאל ונמצא את התדר המקסימלי.

בתוכנה התקבלו שתי האופציות הבאות ל- f_{max} :

עבור 85C

Slow 1100mV 85C Model Fmax Summary			
<<Filter>>			
	Fmax	Restricted Fmax	Clock Name
1	111.23 MHz	111.23 MHz	PLL_Ports altpll_compone...LL_OUTPUT

איור 16: f_{max} 85C

עבור 0C:

Slow 1100mV 0C Model Fmax Summary			
<<Filter>>			
	Fmax	Restricted Fmax	Clock Name
1	112.35 MHz	112.35 MHz	PLL_Ports altpll_comp...OUTPUT_COUNTER divclk

איור 17: f_{max} 0C

LOGIC USAGE

בכל רכיב שאנחנו רוצים לבדוק לו אנחנו מקמפלים את הרכיב ולאחר מכן בודקים analysis מהו השימוש הלוגי שלו:

Analysis & Synthesis Resource Usage Summary		
<<Filter>>		
	Resource	Usage
1	Estimate of Logic utilization (ALMs needed)	56
2		
3	▼ Combinational ALUT usage for logic	94
1	-- 7 input functions	0
2	-- 6 input functions	17
3	-- 5 input functions	19
4	-- 4 input functions	28
5	-- <=3 input functions	30
4		
5	Dedicated logic registers	0
6		
7	I/O pins	55
8		
9	Total DSP Blocks	0
10		
11	Maximum fan-out node	ALUFN[1]~input
12	Maximum fan-out	32
13	Total fan-out	450
14	Average fan-out	2.21

AdderSub Logic Usage: ש 17 איור

Analysis & Synthesis Resource Usage Summary		
<<Filter>>		
	Resource	
1	Estimate of Logic utilization (ALMs needed)	19
2		
3	▼ Combinational ALUT usage for logic	36
1	-- 7 input functions	0
2	-- 6 input functions	2
3	-- 5 input functions	18
4	-- 4 input functions	9
5	-- <=3 input functions	7
4		
5	Dedicated logic registers	0
6		
7	I/O pins	55
8		
9	Total DSP Blocks	0
10		
11	Maximum fan-out node	op[2]~input
12	Maximum fan-out	26
13	Total fan-out	232
14	Average fan-out	1.59

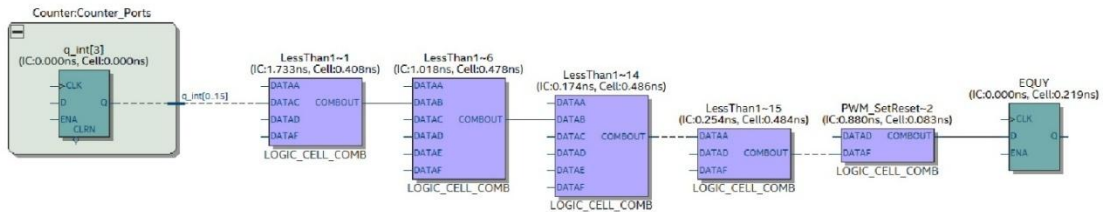
איור 18: Shifter Logic Usage

Analysis & Synthesis Resource Usage Summary		
<<Filter>>		
	Resource	Usage
1	Estimate of Logic utilization (ALMs needed)	19
2		
3	▼ Combinational ALUT usage for logic	36
1	-- 7 input functions	0
2	-- 6 input functions	2
3	-- 5 input functions	18
4	-- 4 input functions	9
5	-- <=3 input functions	7
4		
5	Dedicated logic registers	0
6		
7	I/O pins	55
8		
9	Total DSP Blocks	0
10		
11	Maximum fan-out node	op[2]~input
12	Maximum fan-out	26
13	Total fan-out	232
14	Average fan-out	1.59

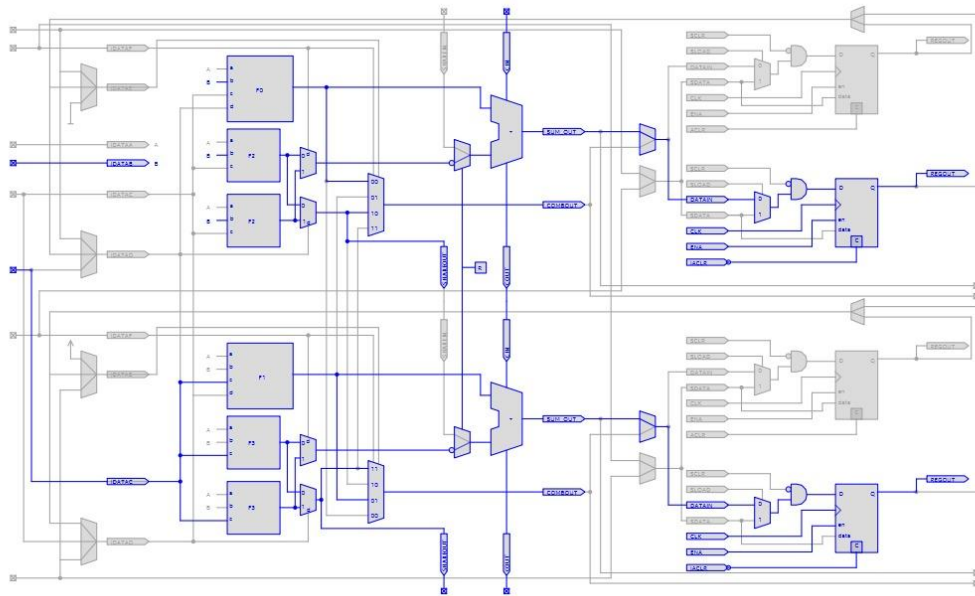
איור 19 :Logic - Logic Usage

:CRITICAL PATH

לצורך מציאת הנתבי הקריטי היה עלינו תחילה לבצע הרצה של Fitter ורק לאחר מכן התקבלה אנליזה של Timquest analyzer מתאימה שבה קיבלנו את הנתבי הקריטי על פי ה-FPGA שבשימוש :

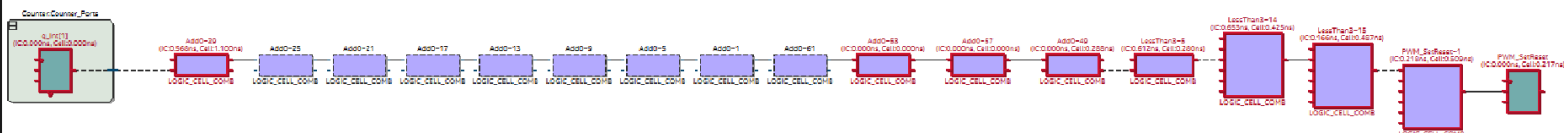


איור 20: Critical Path - technology map viewer



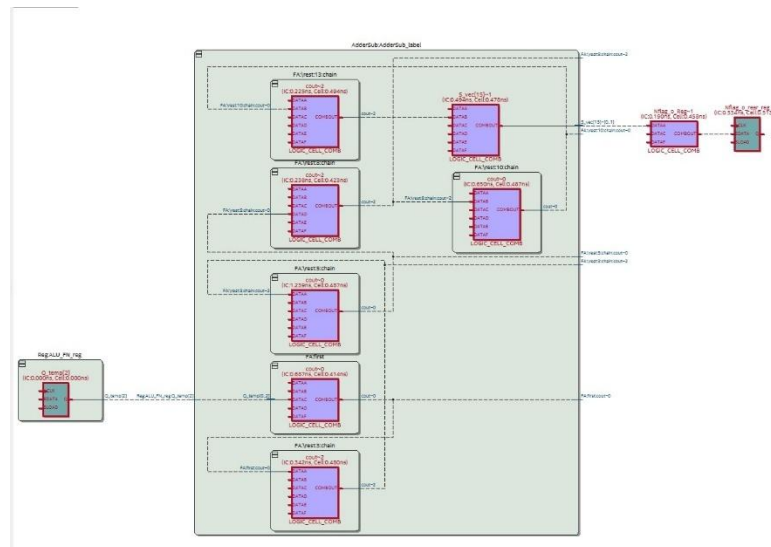
איור 21: Critical Path - resource property editor

:עבור PWM



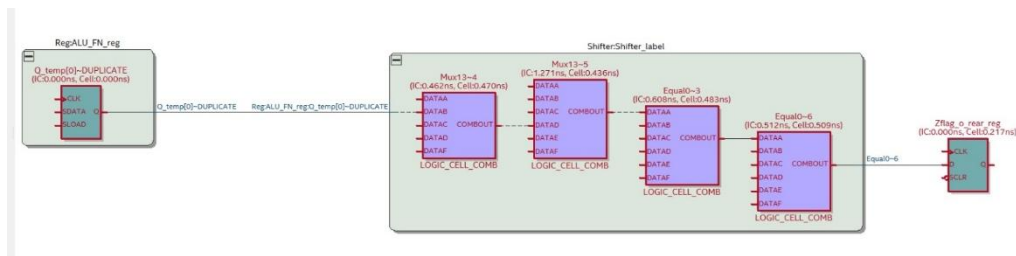
איור 22:Critical Path -PWM

עבור AdderSub :



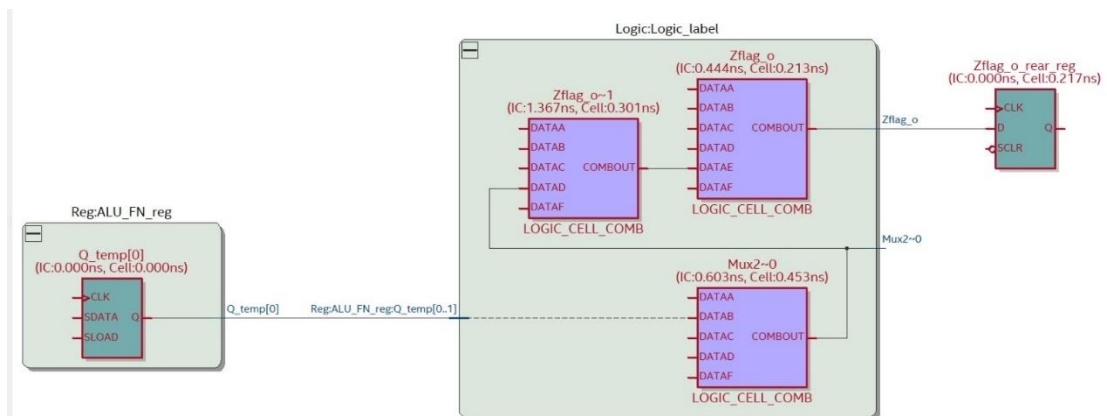
איור 23: Critical Path - AdderSub

עבור Shifter :



איור 24: Critical Path - Shifter

עבור Logic :

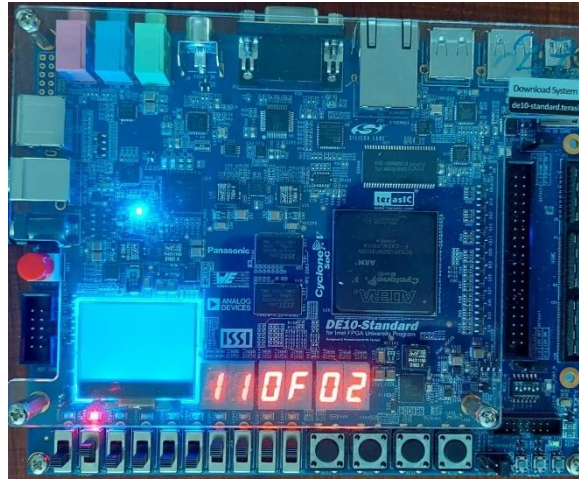


איור 25: Critical Path - Logic

: SIGNAL TAP

כעת נדרשנו להציג היתכנות ופעילות של החומרה ושל הקוד על גבי ה-FPGA ולצורך כך אנחנו משתמשים ב-Signal Tap של קווארטוס כפי שמתואר בתמונות הבאות:

1. בתמונה זו אנו רואים תוצאת חיבור של $F + 2 = 11$ ב-HEX.



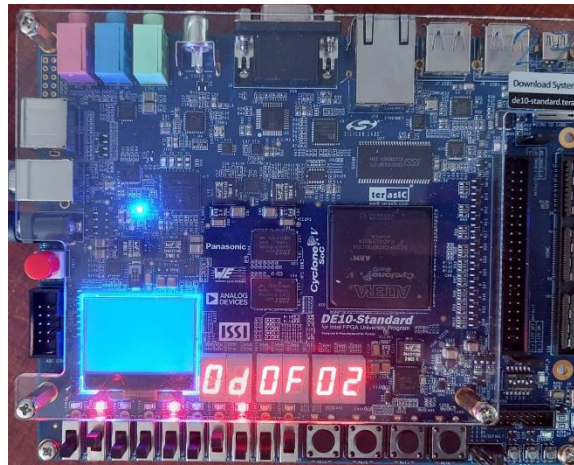
איור 26: תמונת FPGA בתוצאת חיבור



איור 27: ניתוח Signal Tap חיבור

בניתוח זה ניתן לראות כי שינינו את ערך Y ע"י לחיצה על KEY0 ובכך קיבלנו enable ל-Y מה שהוביל לשינוי הערכים על גביי HEX0,1 ובנוסף על גביי צגי התוצאה HEX4,5. כמו כן ניתן לראות כי ALUFN עדיין נשאר בקידוד של ביצוע פעולת חיבור 08h ומופיע בלדים.

2. כעת ביצענו חיבור של $D = F - 2$ ב-HEX.



איור 28: תמונת FPGA בתוצאת חיבור

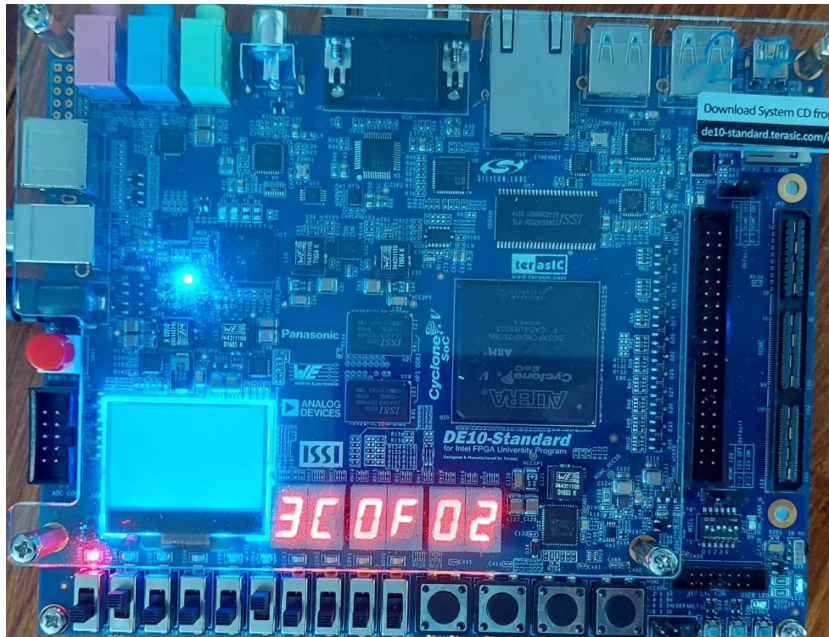
Instance	Status	Enabled	LEs: 1287	Memory: 274432	Small: 0/0	Medium: 34/553	Large: 0/0	Hardware: DE-SoC [USB-1]	Setup...
auto_singaltap_0	Not running	✓	1287 cells	274432 bits	0 blocks	34 blocks	0 blocks	Device: @2: 5C5EBA6[IES]/SCSEMA6/_ (0x02D0200D)	Scan Chain
								SOF Manager:	LAB/LAB 2025/LAB4/our LAB/30.5/output_files/PRELAB4.sof

Type	Alias	Name	-32	0	32	64	96	128	160	192	224	256	288	320	352	384	416	448
HEX	HEX0[6.0]																	
HEX	HEX1[6.0]																	
HEX	HEX2[6.0]																	
HEX	HEX3[6.0]																	
HEX	HEX4[6.0]																	
HEX	HEX5[6.0]																	
LED	LED5[9.0]																	
KEY	KEY0																	
KEY	KEY1																	
KEY	KEY2																	
X	X_High_En																	
X	X_Low_En																	
Y	Y_High_En																	
Y	Y_Low_En																	
Reg	RegALUFN_Reg[0[7.0]																	

איור 29: ניתוח Signal Tap חיבור

בניתוח זה ניתן לראות כי שינינו את ערך ALUFN ע"י לחיצה על KEY2 ובכך קיבלנו enable ל-ALUFN מה שהוביל לשינוי הערך על גבי צגי התוצאה HEX4,5. כמו כן ניתן לראות את הייצוג של ALUFN ע"י נורות הלדים 09h.

3. כעת ביצענו פעולת shl על המספר ב-Y פעמיים והוא המוצג בצג התוצאה



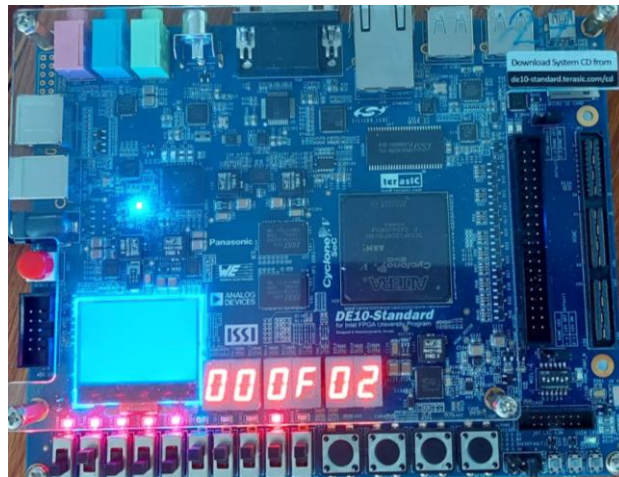
איור 30: תמונת FPGA בתוצאת shl

log: Trig @ 2025/05/30 14:44:06 (0:0:4.4 elapsed)			click to insert time bar															
Type	Alias	Name	-32	0	32	64	96	128	160	192	224	256	288	320	352	384	416	448
		HEX0[6.0]																
		HEX1[6.0]																
		HEX2[6.0]																
		HEX3[6.0]																
		HEX4[6.0]																
		HEX5[6.0]																
		LEDS[9.0]																
		KEY0																
		KEY1																
		KEY2																
		X_High_En																
		X_Low_En																
		Y_High_En																
		Y_Low_En																
		Reg:ALUFN_Reg[Q[7.0]																

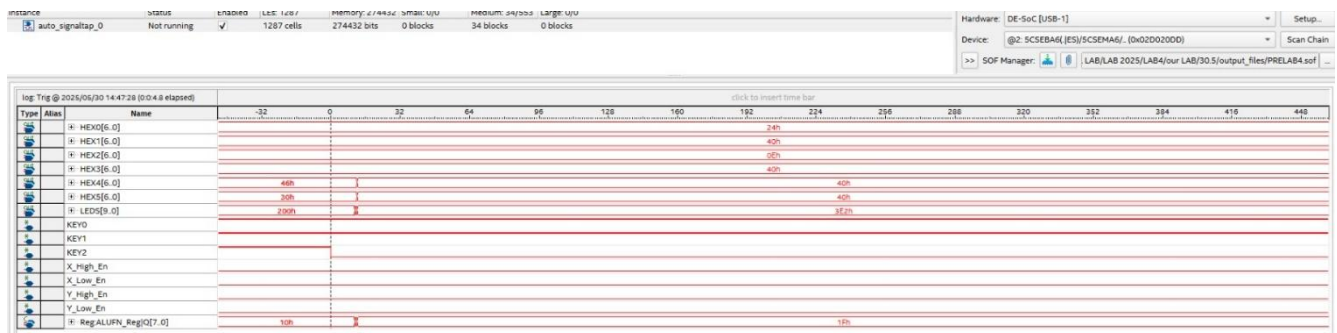
איור 31: ניתוח Signal Tap של shl

בניתוח זה ניתן לראות כי שינינו את ערך ALUFN ע"י לחיצה על KEY2 ובכך קיבלנו enable ל-ALUFN מה שהוביל לשינוי הערך על גבי צגי התוצאה HEX4,5. כמו כן ניתן לראות את הייצוג של ALUFN ע"י נורות הלדים 10h.

(4) לסיום הצגנו פעולה בעלת ALUFN לא חוקי ולכן נקבל בתוצאה 0 :



איור 32: תמונת FPGA פעולה לא חוקית



איור 33 : ניתוח Signal Tap פעולה לא חוקית

בניתוח זה ניתן לראות כי שינינו את ערך ALUFN עי"י לחיצה על KEY2 ובכך קיבלנו enable ל-ALUFN מה שהוביל לשינוי הערך ל-00 על גבי צגי התוצאה HEX4,5. כמו כן ניתן לראות את הייצוג של ALUFN עי"ג נורות הלדים 1Fh.