Preparation Report LAB4

דביר חן סעד -214078792

ירון חדד- 316411057

תוכן עניינים

מטרת המעבדה.....3

בדיקת ביצועים.....3

סימולציה.....4-6

מציאת תדר מקסימלי.....7

תיאור המערכת.....8-10

הצגת הרכיבים RTL ומסלול קריטי.....10-16

שימוש בסיגנלים 17

מטרת המעבדה

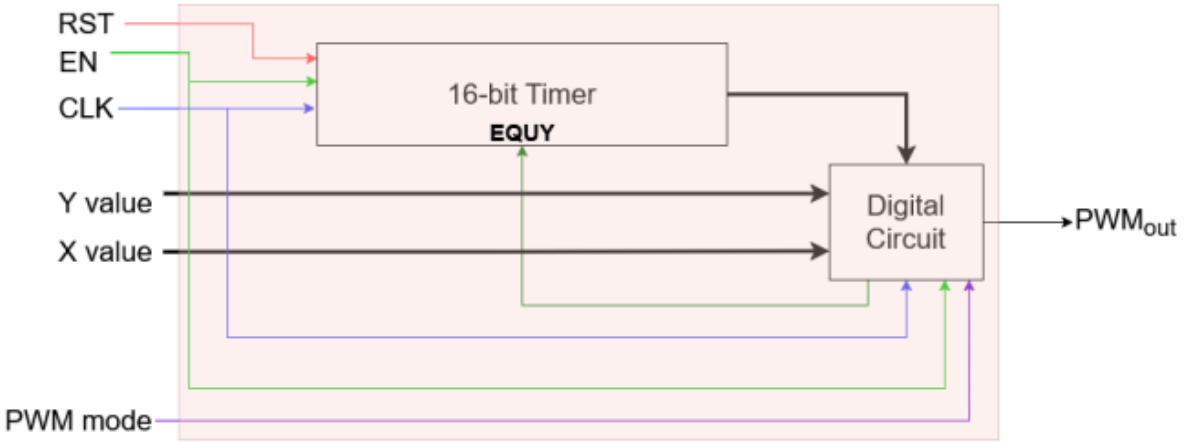
במעבדה זו נלמד כיצד לבצע סינתזה עבור מודלים שאנו נפתח בעזרת תוכנת Quartus ובפרט נשתמש במודל אשר פיתחנו במעבדה 1 ונוסיף לו עוד מודל סינכרוני אשר מוגדר בהמשך.

את הסינתזה נבצע על cyclone v FPGA.

בנוסף למודל של מעבדה 1 התבקשנו להוסיף עוד מודל סינכורני אשר מייצר לנו גלי PWM לפי קוד הALU המבוקש.

את הרכיב ממישנוב בעזרת מונה(counter) ויחידה לוגית אשר אחראית למימוש הגלים.

להלן המודל אותו ממשנו בתמונה מס1

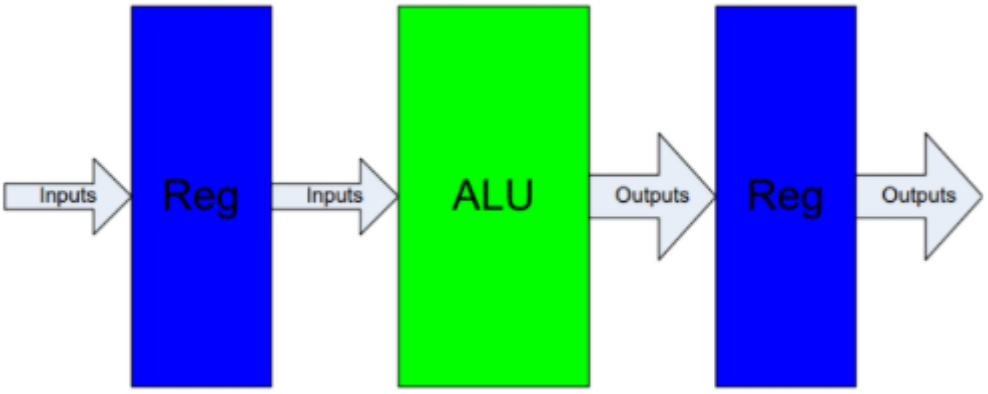


תמונה 1 מעטפת למודל הPWM

בדיקת ביצועים

בכדי לבדוק את הביצועים של המודל אשר פיתחנו במעבדה 1 אך בכדי לבצע סינתה נצטרך רכיב סינכרוני אך הרכיב של מעבדה 1 לא סינכרוני ולכן נוסיף באופן מלאכותי מערכת סינכרונית של רגיסטר כניסה ורגיסטר מוצא אשר הם יפעלו לפי אותו שעון.

כפי שמוצג בתמונה מס 2



תמונה 2 מעטפת למודל הALU

כאשר אל הרגיסטרים מחובר השעון של הכרטיס -

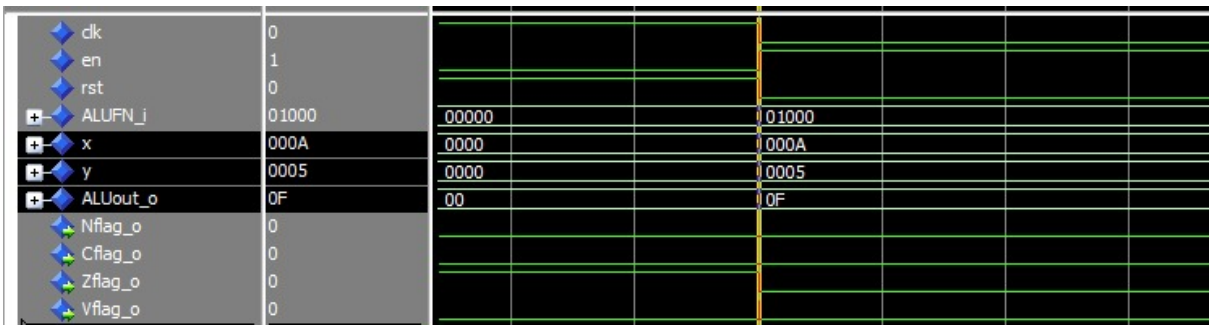
signal name	FPGA Pin No	Description	I/O Standard
CLOCK_50	PIN_AF14	50MHz clk input	3.3V

סימולציית Model-sim

בסימולציה זו ניצור TB אשר יבדוק את המודל באופן פרטני עבור כל מודל כאשר הכל עטוף במעטפת של הtop.

נסתכל על מספר של בדיקות ונראה שאכן מתבצע כמצופה

פעולת add

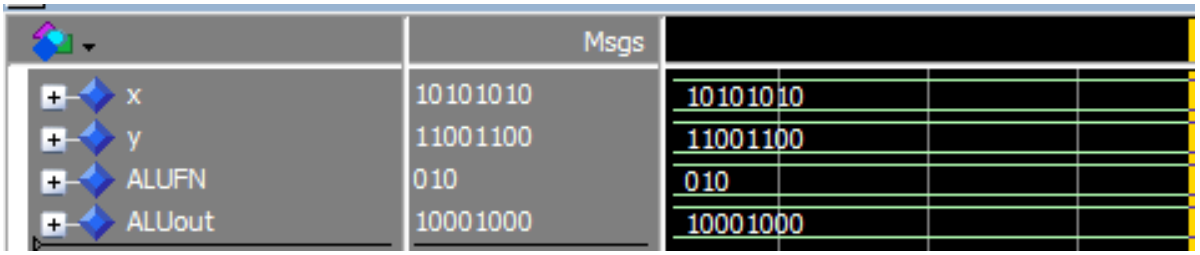


תמונה 3 תוצאת פעולת החיבור

כפי שאנו רואים ערכי הרגיסטר X וערך הרגיסטר Y נסכמים ואכן אנו מקבלים את התוצאה הרצויה A+5=F

כלומר 15=10+5

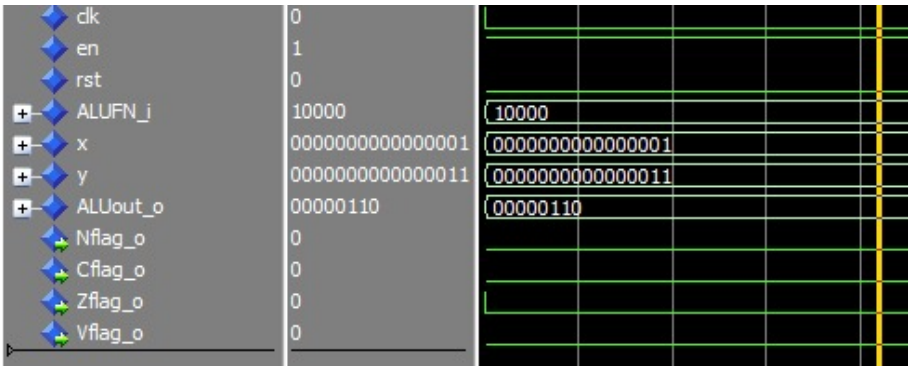
פעולת Logic



תמונה 4 פעולת logic

אנו רואים כי קוד הפעולה 010 הינו פעולת הand ואנו רואים כי רק ביט הMSB והביט ה4 דולקים אצל שני הכניסות ואכן תוצאת הפעולה היא שיש 1 רק עבור 2 הביטים הנ"ל

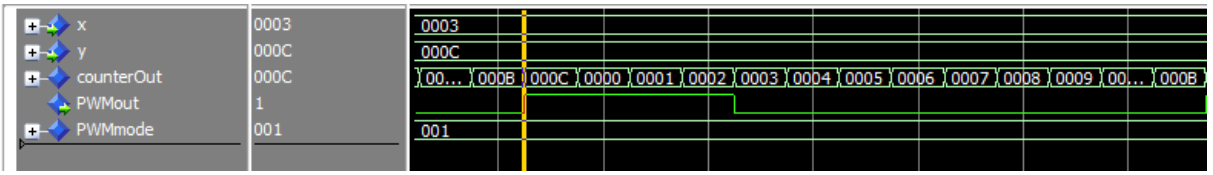
פעולת SHL



תמונה 5 תוצאת פעולת הזזה

נשים לב כי אותות הכניסה הינם 1 ו3 כלומר
ערך המוזז הינו 3
ואנו נצטרך לבצע הזזה אחת כלומר 011-> 110 כלומר נצפה לקבל את הערך 6
ואכן המוצא הינו 6.

פעולת PWM



תמונה 6 תוצאת פעולת אות pwm

כפי שניתן לראות קוד הפעולה הינו 00001 כלומר כפי שהוגדר הגל אמור לעלות בערך y ולרדת כאשר אנו מגיעים לערך x
במצב שלנו x=3,y=c ואכן אנו רואים כי בזמן הגעת השעון לערך c האות עולה ל1 ונשאר על הערך 1 עד הגעת השעון לערך 3.

מציאת תדר מקסימלי

בכדי למצוא את התדר המקסימלי של המערכת שלנו כמו שהזכרנו למעלה הוספנו 2 רגיסטרים בכניסה ובמוצא של מודל הalu.
כעת נבצע קימפול וסינתזה של הרכיב כפי שמבוקש .
לאחר הקימפול נוכל להכנס לTool ולאחר מכן לבחור את האופציה שלtimer ושם יש לנו את העמודה שמראה מהו התדר המקסימלי:

Slow 1100mV 85C Model Fmax Summary

<<Filter>>

	Fmax	Restricted Fmax	Clock Name	Note
1	288.18 MHz	288.18 MHz	clk	

תמונה 7 מציאת התדר המקסימלי של המודל

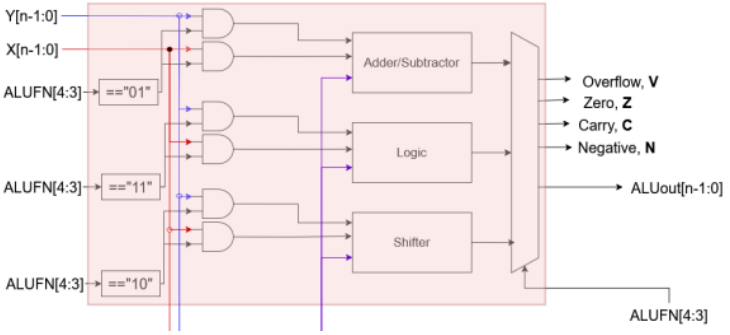
לאחר שמצאנו את התדר הנ"ל נצטרך להציב אותו במערכת כדי ליצור חסם על התדר בכדי לא לחרוג מהמערכת וגם בכדי לא לבזבז הספק לחינם.

תיאור המערכת

המערכת שלנו מפוצלת ל2 חלקים נפרדים כאשר כל מודול מורכב מתתי רכיבים
-רכיב הalu – רכיב אסינכרוני אשר פועל ללא תלות בשעון
-רכיב הSDC– רכיב סינכרוני אשר פועל בזמן של עליית שעון
כעת נפרק מה הלוגיקה של כל אחד מהם ונראה את הסינתזה שלהם בעזרת תוכנת Quartus.

רכיב הALU

נרצה לממש מערכת אשר מבצעת מספר של פעולות כאשר לכל פעולה יש קוד פעולה ספציפי אשר מגדיר מתי נשתמש בה .



תמונה 8 מודל הALU

המודל מקבל 3 כניסות

- X- כניסה בעלת 16 ביטים
- Y -כניסה בעלת 16 ביטים
- ALUFN -קו בקרה בעל 5 סיביות שמגדיר את הפעולה שצריך לבצע

<u>מוצאי המערכת הם:</u>

- דגלים Cflag,Vflag,Zflag,Nflag- שמופעלים בהתאם לתוצאה של הפעולה שהתבצעה .
- ALUOUT מוצא בעל 16 סיביות שהוא התוצאה של הפעולה על x ועל y.

Arithmetic (*Y and X are 8-bit width*)	8	**01**000	Res=Y+X	
	9	**01**001	Res=Y-X	Used also for comparison operation
	10	**01**010	Res=neg(X)	
	11	**01**011	Res=Y+1	Increment of Y in one
	12	**01**100	Res=Y-1	Decrement of Y in one
	13	**01**101	Res=swap(Y)	Res=(Y_{LSHW}, Y_{MSHW})
Shift (*Y and X are 8-bit width*)	16	**10**000	Res=SHL Y,X(k-1 to 0)	Shift Left Y of q≜X(k-1 ..0) times Res=Y(n-1-q…0)#(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-1 ..0) times Res=(q@0)#Y(n-1...q) ***When*** $k = log_2 n$
Boolean (*Y and X are 8-bit width*)	24	**11**000	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**110	Res=Y xnor X	

תמונה 9 הפעולות אשר ממושמות במודל שלALU

רכיב הPWM

רכיב בעל 6 כניסות

- X- כניסה בעלת 16 ביטים
- Y -כניסה בעלת 16 ביטים
- ALUFN -קו בקרה בעל 5 סיביות שמגדיר את הפעולה שצריך לבצע
- קוי בקרה En,Rst שאחראיים על תקפוד המערכת
- כניסת שעון clk

<u>מוצאי המערכת הם:</u>

- אות PWM שעליתו וירידתו הם כתוצאה מX,Y וקוד הפעולה

Function Type	Decimal value	ALUFN	Operation	Note
PWM Output (*Y and X are 16-bit width*)	0	**00**000	PWM MODE0	PWM Mode is Set/Reset
	1	**00**001	PWM MODE1	PWM Mode is Reset/Set
	2	**00**010	PWM MODE2	PWM Mode is Toggle

תמונה 10 הפעולות אשר ממושמות במודל שלPWM

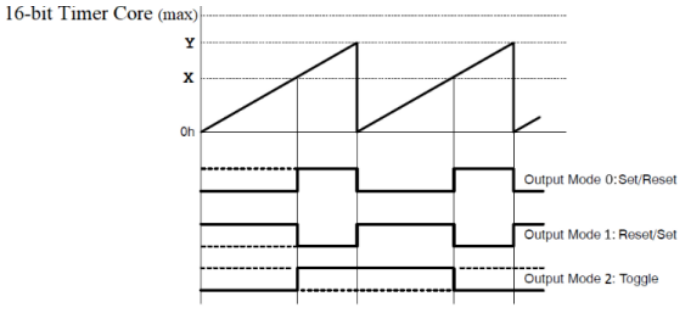
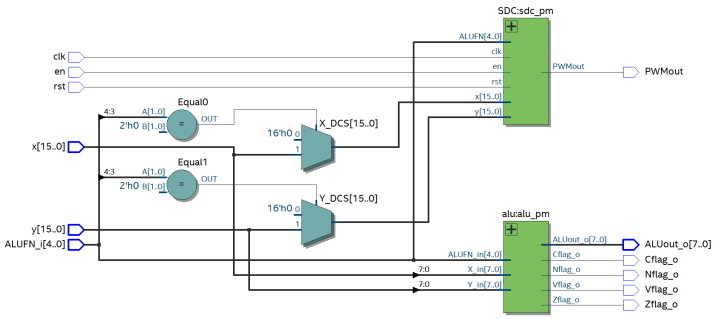


Figure 5: PWM output unit architecture

תמונה 11 סוגי הגלים אשר יוצאים מתוך המודולו כתלות בcounter

שרטוט הRTL



תמונה 11 RTL של המערכת הכוללת

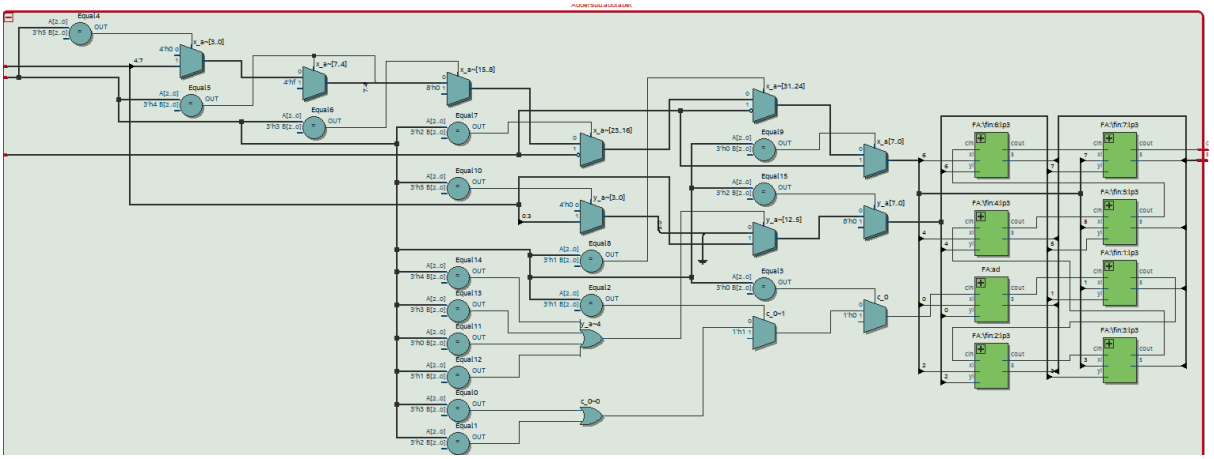
אפשר לראות שהמערכת מורכבת מ2 רכיבים אחד סינכרוני ואחד אסינכרוני כאשר יש mux שבורר בין 0 לערכי הכניסות וזה בכדי לשמור על בזבוז ההספק של המערכת.

בזמן שהרכיב לא משומש נכנס הערכים 0000 בכל כניסה.

רכיב AdderSub

רכיב זה מומש כבר במעבדה 1 ולכן רק נציג את התוספות שתורמת לנו התוכנה של Quartusה

שרטוט הRTL



תמונה 12 RTL של הaddersub

שימוש בלוגיקה

כאשר פיתחנו את הקוד הלוגי אנו משתמשים באלמנטים הניתנים לקינפוג בהתאם לקוד הכתוב, בעזרת הקוארטוס בזמן תהליך הסינתזה אפשר להראות את הלוגיה של הרכיב:

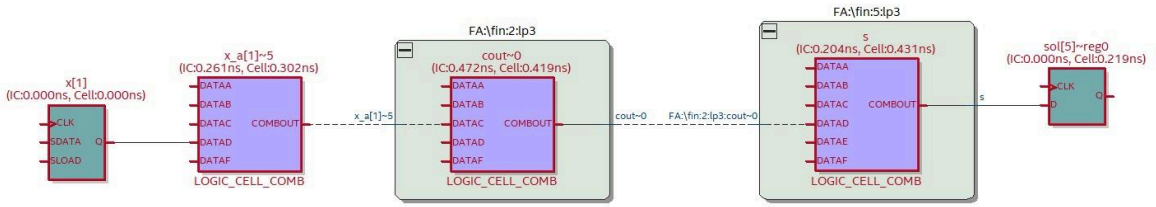
Flow Status	Successful - Mon May 26 16:53:56 2025
Quartus Prime Version	21.1.0 Build 842 10/21/2021 SJ Lite Edition
Revision Name	lab4
Top-level Entity Name	AdderSub
Family	Cyclone V
Device	5CSXFC6D6F31C6
Timing Models	Final
Logic utilization (in ALMs)	20 / 41,910 (< 1 %)
Total registers	24
Total pins	29 / 499 (6 %)
Total virtual pins	0
Total block memory bits	0 / 5,662,720 (0 %)
Total DSP Blocks	0 / 112 (0 %)
Total HSSI RX PCSs	0 / 9 (0 %)
Total HSSI PMA RX Deserializers	0 / 9 (0 %)
Total HSSI TX PCSs	0 / 9 (0 %)
Total HSSI PMA TX Serializers	0 / 9 (0 %)
Total PLLs	0 / 15 (0 %)
Total DLLs	0 / 4 (0 %)

תמונה 13 לוגיה של הaddersub

נתיב קריטי

הנתיב הקריטי חשוב ביותר מכיוון שהוא נותן לנו מידע על הזרימה של המידע וגם נותן לנו מגבלות עליה

מכיוון שלרכיבים יש השהייה עד להתייצבות הכניסות נצטרך לקחת זאת בחשבון להלן הנתיב הקריטי :

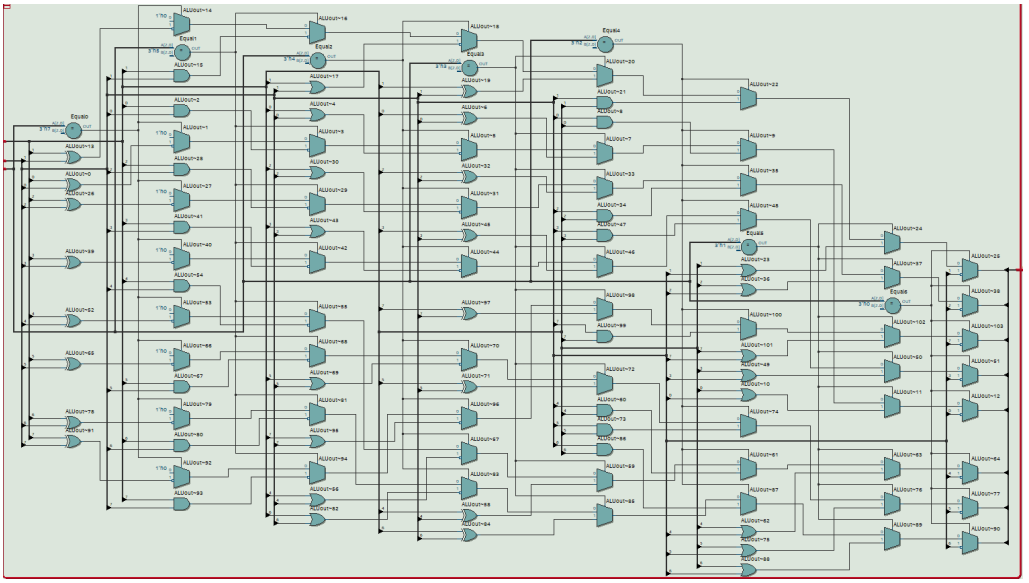


תמונה 14 נתיב קריטי של הaddersub

רכיב Logic

רכיב זה מומש כבר במעבדה 1 ולכן רק נציג את התוספות שתורמת לנו התוכנה של Quartusה

שרטוט הRTL



תמונה 15 RTL של הlogic

שימוש בלוגיקה

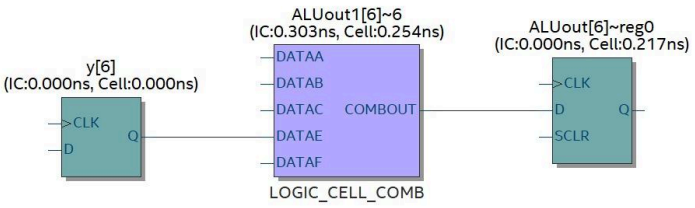
כאשר פיתחנו את הקוד הלוגי אנו משתמשים באלמנטים הניתנים לקינפוג בהתאם לקוד הכתוב, בעזרת הקוארטוס בזמן תהליך הסינתזה
אפשר להראות את הלוגיה של הרכיב:

Analysis & Synthesis Status	Successful - Mon May 26 14:46:06 2025
Quartus Prime Version	21.1.0 Build 842 10/21/2021 SJ Lite Edition
Revision Name	lab4
Top-level Entity Name	Logic
Family	Cyclone V
Logic utilization (in ALMs)	N/A
Total registers	24
Total pins	28
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

תמונה 16 לוגיה של logic

נתיב קריטי

הנתיב הקריטי חשוב ביותר מכיוון שהוא נותן לנו מידע על הזרימה של המידע וגם נותן לנו מגבלות עליה
מכיוון שלרכיבים יש השהייה עד להתייצבות הכניסות נצטרך לקחת זאת בחשבון להלן הנתיב הקריטי :

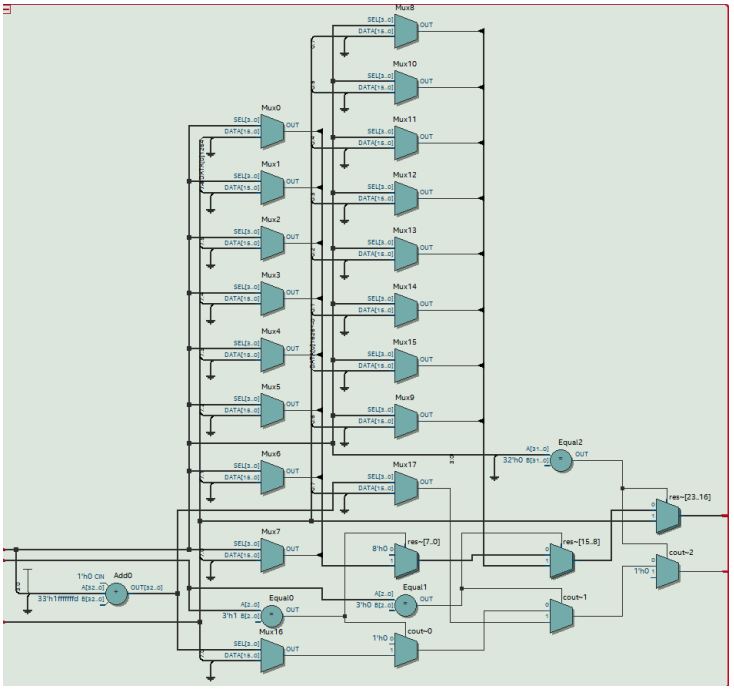


תמונה 17 נתיב קריטי של הlogic

רכיב Shifter

רכיב זה מומש כבר במעבדה 1 ולכן רק נציג את התוספות שתורמת לנו התוכנה של Quartus

שרטוט הRTL



תמונה 18 RTL של הshifter

שימוש בלוגיקה

כאשר פיתחנו את הקוד הלוגי אנו משתמשים באלמנטים הניתנים לקינפוג בהתאם לקוד הכתוב, בעזרת הקוארטוס בזמן תהליך הסינתזה אפשר להראות את הלוגיה של הרכיב:

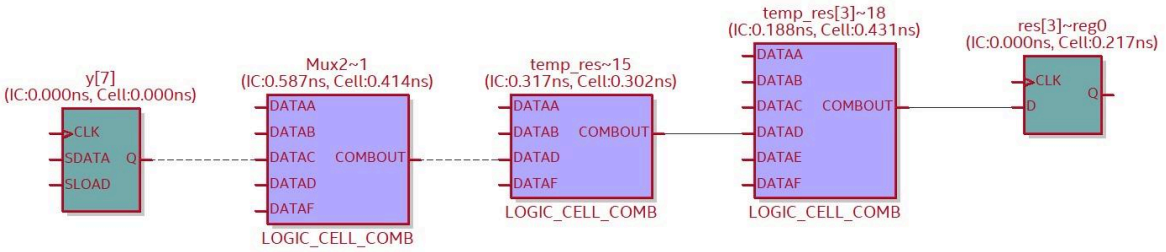
Analysis & Synthesis Status	Successful - Mon May 26 16:31:42 2025
Quartus Prime Version	21.1.0 Build 842 10/21/2021 SJ Lite Edition
Revision Name	lab4
Top-level Entity Name	Shifter
Family	Cyclone V
Logic utilization (in ALMs)	N/A
Total registers	20
Total pins	29
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

תמונה 19 לוגיה של shifter

נתיב קריטי

הנתיב הקריטי חשוב ביותר מכיוון שהוא נותן לנו מידע על הזרימה של המידע וגם נותן לנו מגבלות עליה

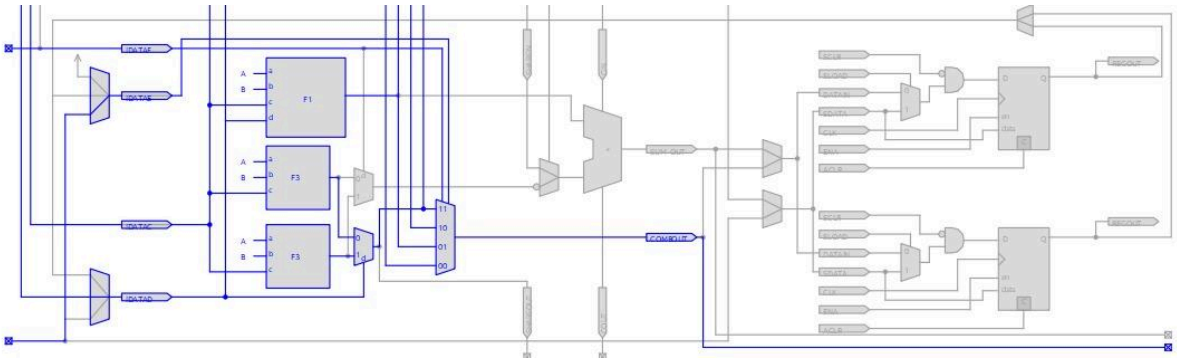
מכיוון שלרכיבים יש השהייה עד להתייצבות הכניסות נצטרך לקחת זאת בחשבון להלן הנתיב הקריטי :



תמונה 20 נתיב קריטי של הshifter

מסלול קריטי כלל ALU

נראה מהו המסלול הקריטי של כל יחידת הALU



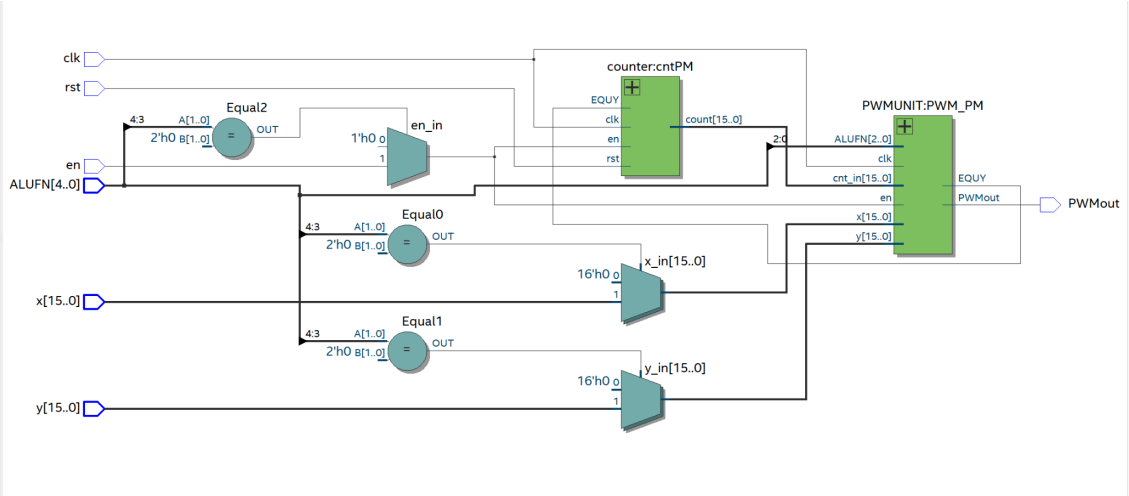
תמונה 21 נתיב קריטי של הalu

רכיב הSDC

רכיב זה הוא רכיב סינכרוני אשר תלוי בשעון ואות כניסה en
רכיב זה מורכב מ2 תתי רכיבים :
-רכיב מנייה
-רכיב בקרה (control)
רכיב המנייה בכל עליית שעון עולה עד לערך הy ובהגיעו לערך הוא מתאפס ומתחיל את המנייה מחדש.

רכיב הבקרה מחליט על מצב הפעולה והוצאת המוצא למערכת ובמקביל נותן מידע לרכיב המנייה על הגעת המנייה לערך y.

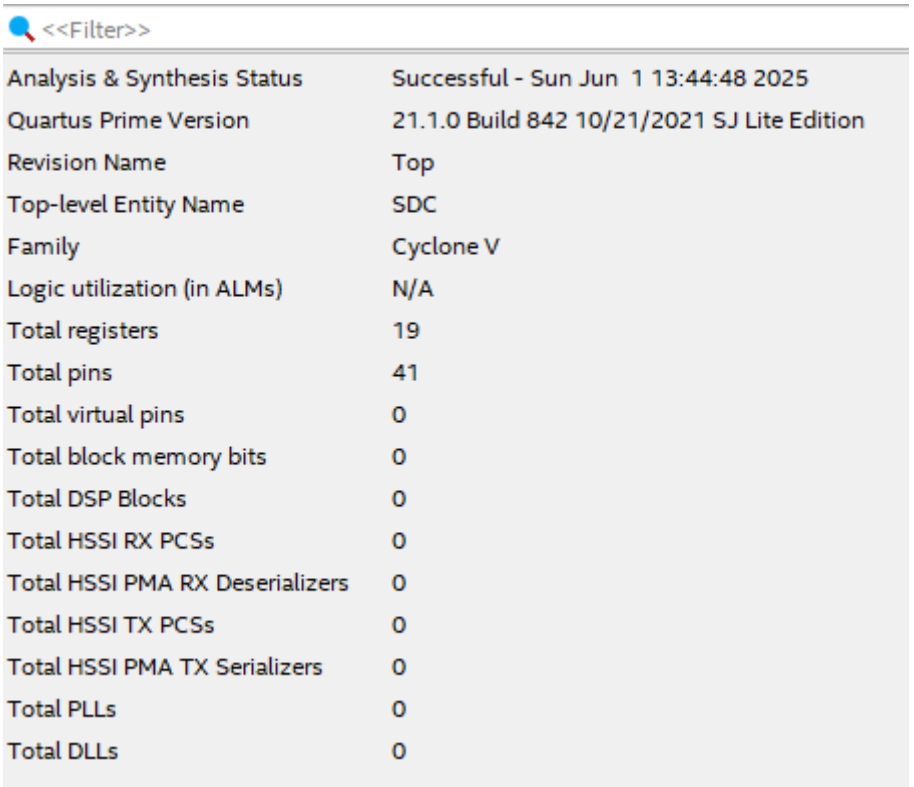
שרטוט RTL



תמונה 22 RTL של רכיב סינכרוני

שימוש בלוגיקה

כאשר פיתחנו את הקוד הלוגי אנו משתמשים באלמנטים הניתנים לקינפוג בהתאם לקוד הכתוב, בעזרת הקוארטוס בזמן תהליך הסינתזה אפשר להראות את הלוגיה של הרכיב:



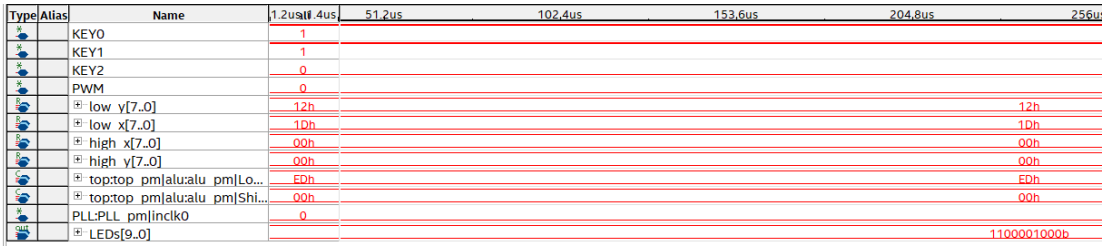
<<Filter>>	
Analysis & Synthesis Status	Successful - Sun Jun 1 13:44:48 2025
Quartus Prime Version	21.1.0 Build 842 10/21/2021 SJ Lite Edition
Revision Name	Top
Top-level Entity Name	SDC
Family	Cyclone V
Logic utilization (in ALMs)	N/A
Total registers	19
Total pins	41
Total virtual pins	0
Total block memory bits	0
Total DSP Blocks	0
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0
Total DLLs	0

תמונה 23 לוגיקה של רכיב סינכרוני

Signal Tap

נרצה לבצע וריפיקציה של החומרה על ידי שימוש בתוכנת הsignal-tap שמובנת בתוך תוכנה הquartus.
בזמן אמת נוכל לתפוס את השינוי בפעולה על ידי בקשה של falling edge על גבי key2 שהוא יהיה הטריגר עצירה שלנו

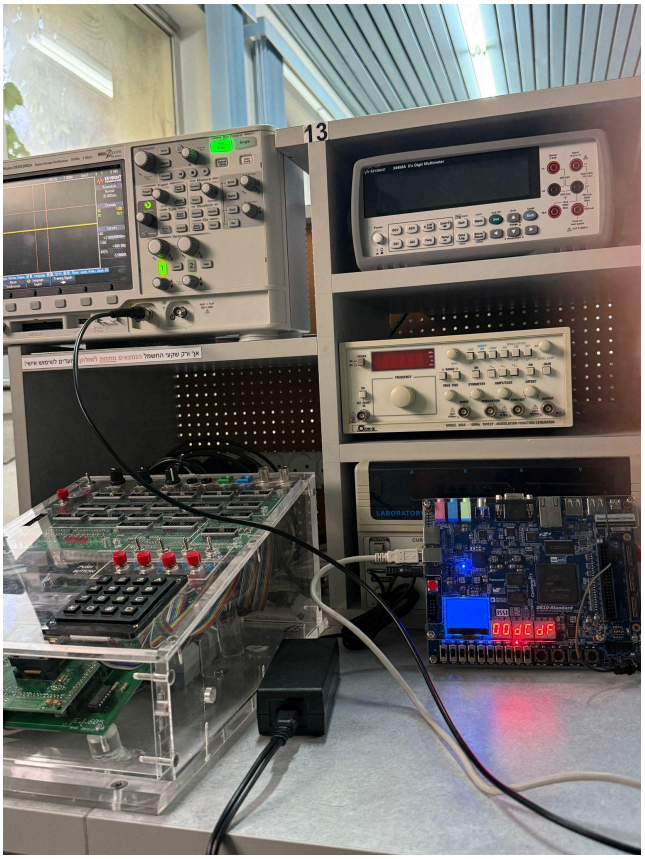
הכנסנו את הסיגנלים של הleds בכדי לעקוב אחרי הopc של המערכת
הוספנו גם את רגיסטרים x,y ואת רגיסטר המוצא כדי לעקוב אחר התוצאה ולראות שכן התוצאה נכונה.



Type	Alias	Name	1.2us	51.2us	102.4us	153.6us	204.8us	256us
		KEY0	1					
		KEY1	1					
		KEY2	0					
		PWM	0					
		low_y[7..0]	12h					12h
		low_x[7..0]	1Dh					1Dh
		high_x[7..0]	00h					00h
		high_y[7..0]	00h					00h
		top:top_pm|alu:alu_pm|Lo...	EDh					EDh
		top:top_pm|alu:alu_pm|Shi...	00h					00h
		PLL:PLL_pm|inclk0	0					
		LEDs[9..0]						1100001000b

תמונה 24 פעולת or על x,y

מעבדה מהנה סהכ



תמונה 25 הליכה לבדוק את פעולה pwm