

Lab Report

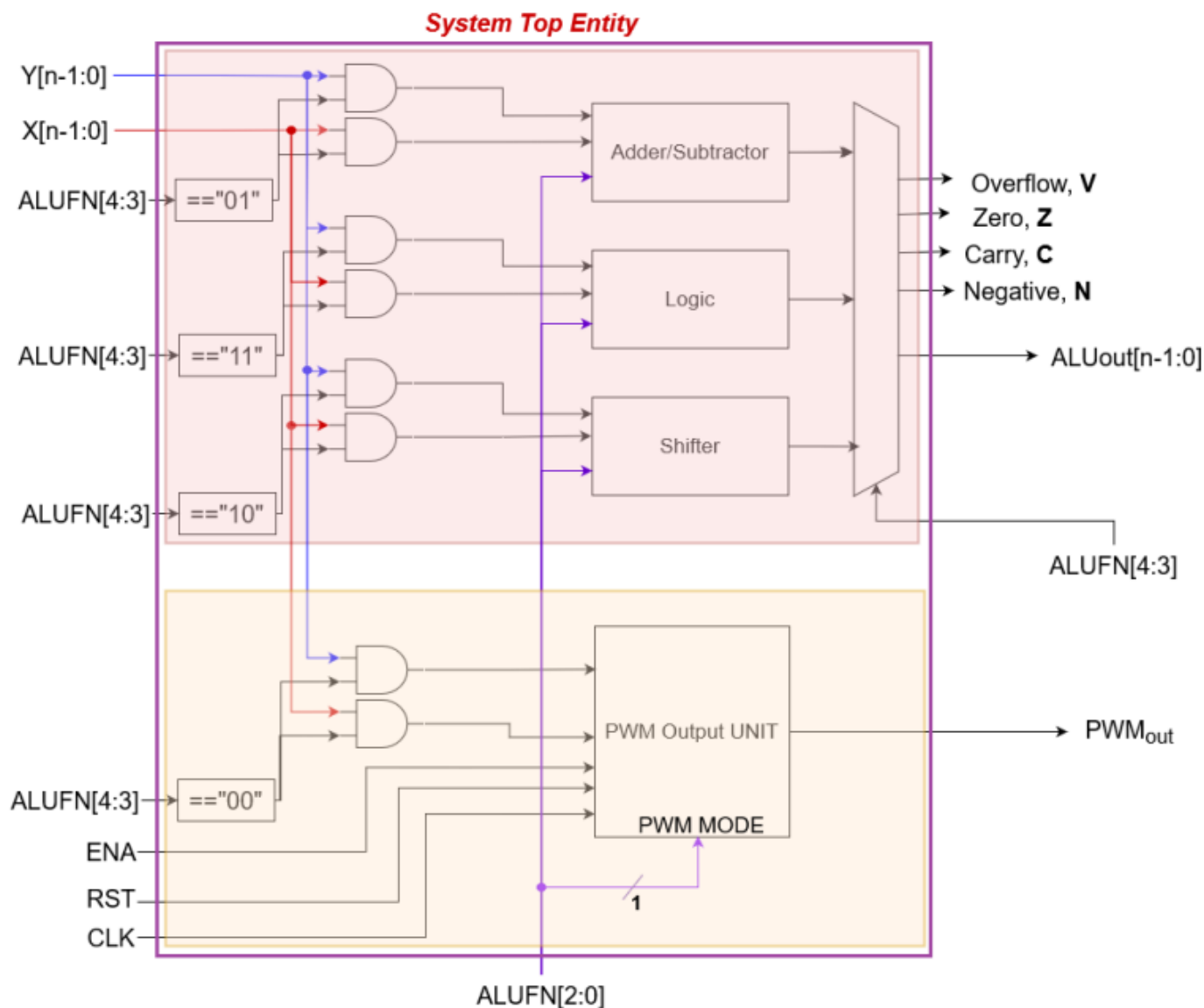
LAB4 - FPGA BASED DIGITAL DESIGN

Hagai Joseph- 207838178
Bar Kupferschmied - 318912193

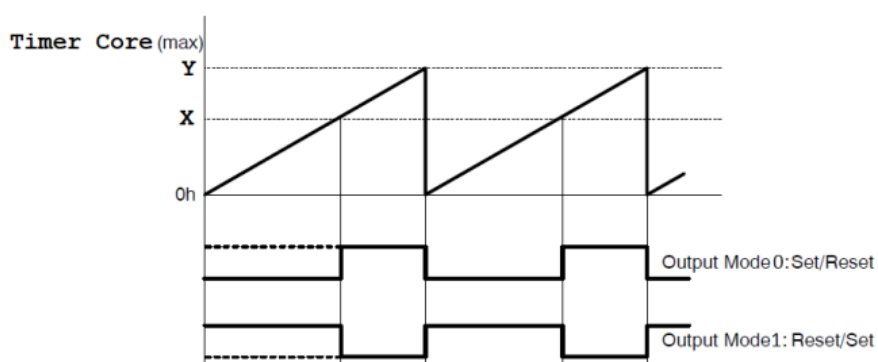
במעבדה זו למדנו להשתמש בתוכנת Quartus ובפרט לבצע סינתזה עבור המודל שפיתחנו במעבדה 1 בתוספת יחידה חדשה בשם PWM.

את הסינתזה ביצענו על גבי FPGA V Cyclone של כרטיס standard 10DE.

מבנה המערכת:



איור 2: system top entity



איור 2: PWM output modes

החלק העליון של המערכת (בצבע ורוד) מימשנו בקוד VHDL במעבדה 1. את החלק התחתון (צבע כתום בהיר), יחידת PWM, הוספנו למודל במעבדה זו. יחידה זו סינכרונית (בשונה מהיחידה העליונה) ומקבלת 2 מספרים X,Y אות איפשור (en) ואות איפוס (rst) ומצב עבודה PWM_MODE, היחידה מוציאה אות PWM_{out}, בהתאם למצב העבודה לפי החוקיות הבאה:

עבור מצב עבודה PWM_MODE=0:

$$PWM_{out} = \begin{cases} 0, & \text{when } 0 \leq counter < X \\ 1, & \text{when } X \leq counter < Y \end{cases}$$

עבור מצב עבודה PWM_MODE=1:

$$PWM_{out} = \begin{cases} 1, & \text{when } 0 \leq counter < X \\ 0, & \text{when } X \leq counter < Y \end{cases}$$

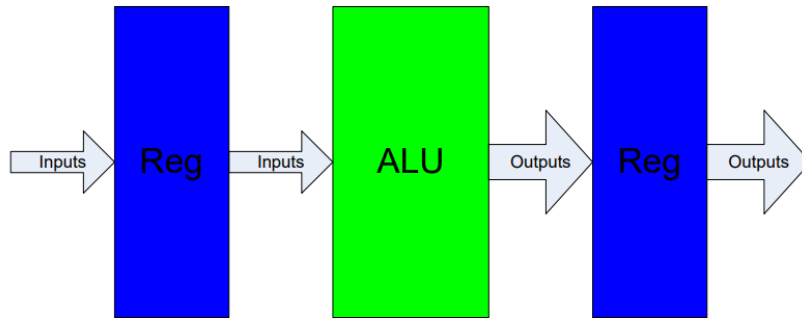
כאשר הסופר (counter) מתאפס כאשר הוא מגיע לY.

Function Kind	Decimal value	ALUFN	Operation	Note
PWM Output	0	00000	PWM MODE0	PWM Mode is Set/Reset
	1	00001	PWM MODE1	PWM Mode is Reset/Set
Arithmetic	8	01000	Res=Y+X	
	9	01001	Res=Y-X	Used also for compare operation
	10	01010	Res=neg(X)	
Shift	16	10000	Res=SHL Y,X(k-1 to 0)	Shift Left Y of $q \triangle X(k-1 \dots 0)$ 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 $q \triangle X(k-1 \dots 0)$ times Res=(q@0)#Y(n-1...q) When $k = \log_2 n$
Boolean	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	11111	Res=Y xnor X	

איור 3 : The Digital System ISA

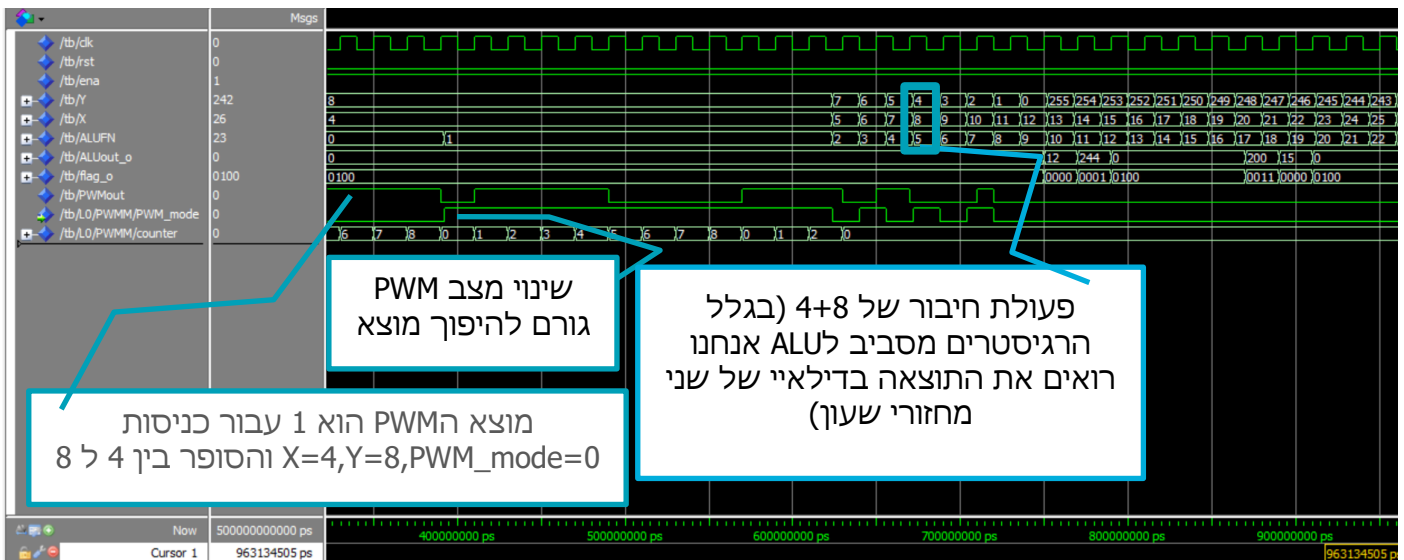
שלב 1: בדיקת ביצועים

בשלב הראשון ביצענו בדיקת ביצועים ראשונית של המערכת בשימוש בתוכנת Quartus. כיוון שיחידת הALU היא א-סינכרונית, נצטרך לחבר למערכת רגיסטרים סינכרוניים לכניסה ולמוצא על מנת לבצע אנליזה בזמן:



איור 4 : בדיקה במקרה של הALU שהיא מערכת לוגית טהורה

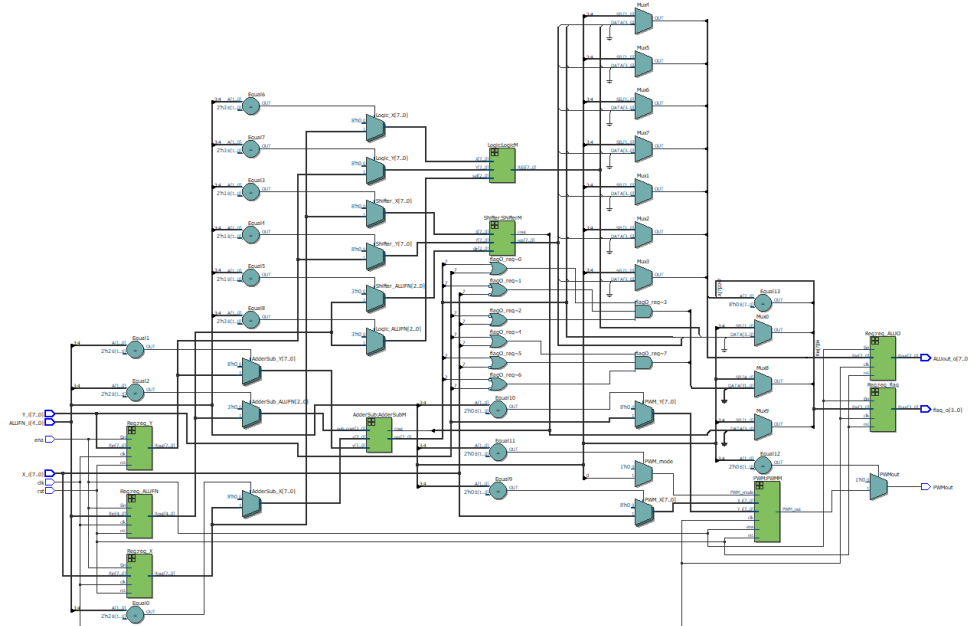
ניתוח בתוכנת MODELSIM:



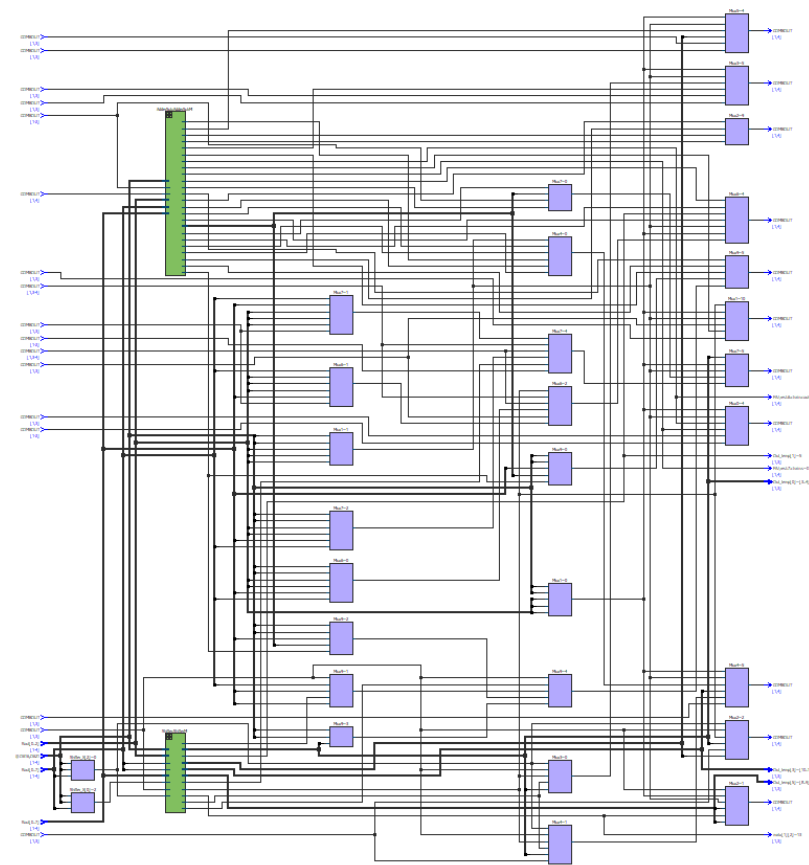
איור 5 : ניתוח top entity בתוכנת ModelSim

ניתן לראות כי עבור מצב 0 ביחידת הPWM, עבור כניסות X=4, Y=8 כאשר הסופר בין 8 ל 4 המוצא הוא 1. כאשר נשנה את מצב העבודה למצב 1 נקבל היפוך במוצא ל0. בנוסף ניתן לראות כי הסופר מתאפס ב8, ואחרי שניו המצב ל1 נקבל את האות שאנחנו מצפים לו במוצא. בנוסף ניתן לראות שהסופר מתאפס בשניו הכניסות כפי שציפינו.

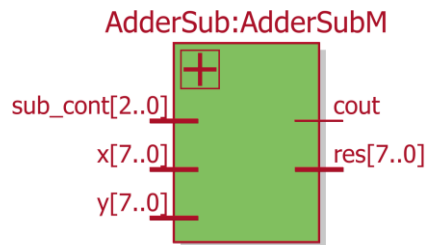
בנוסף ניתן לראות כי יחידת הALU מקיימת את הפעולות כפי שציפינו – עבור כניסת ALUFN של "01000" נקבל פעולה חיבור בין X=4, Y=8 ונקבל במוצא 12. נשים לב כי התוצאה מגיעה בדילאיי של 2 מחזורי שעון, שכן טפנו את היחידה ברגיסטרים בכניסה וביציאה לקראת ניתוח בתוכנת Quartus.



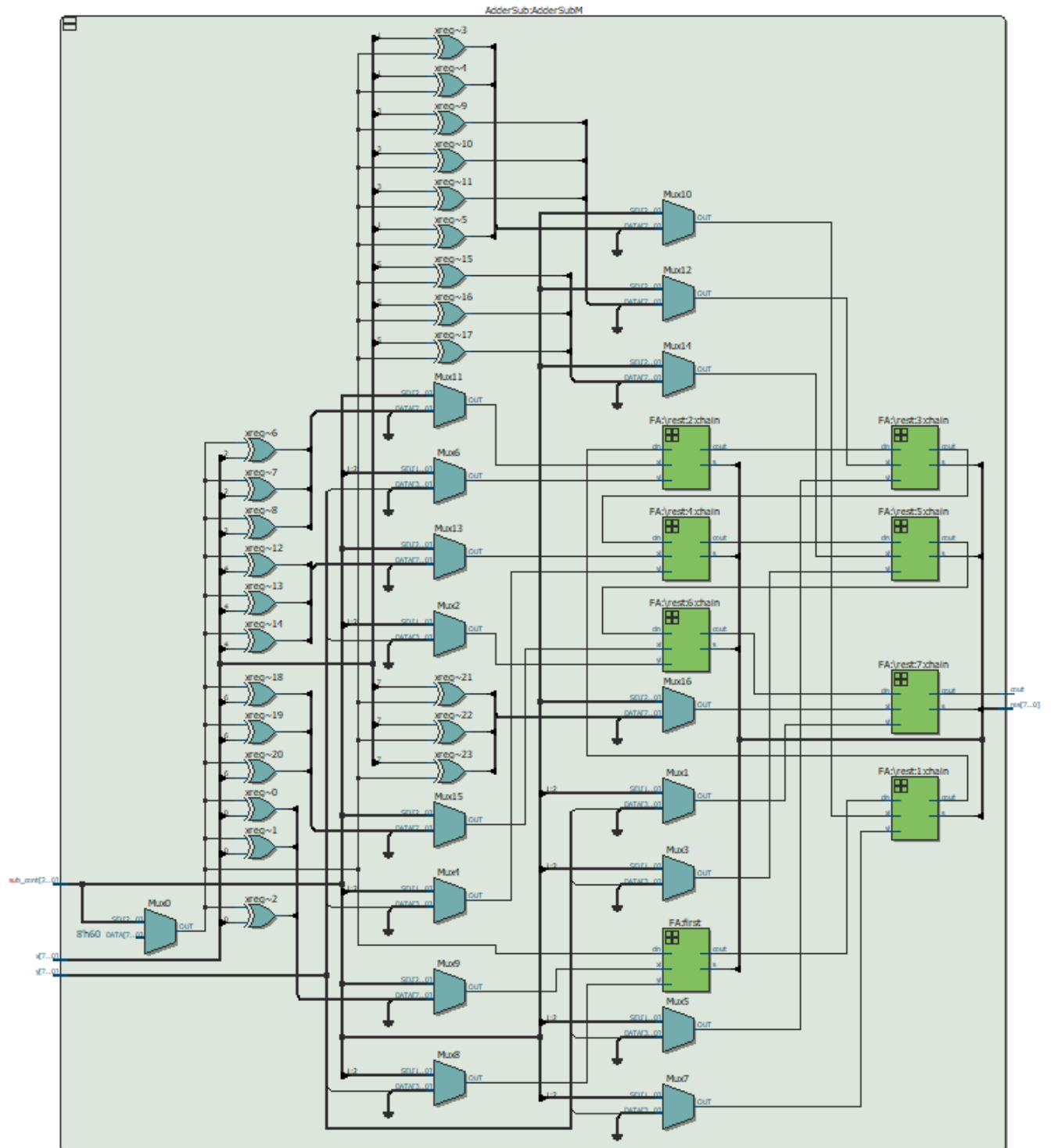
איור 7: *RTL view* של *top entity*



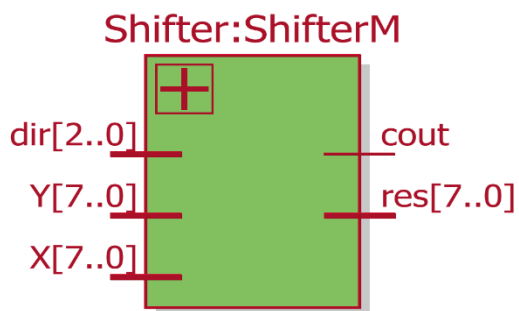
איור 7 : מפה סינתטית (אחרי התאמה) של *top entity*



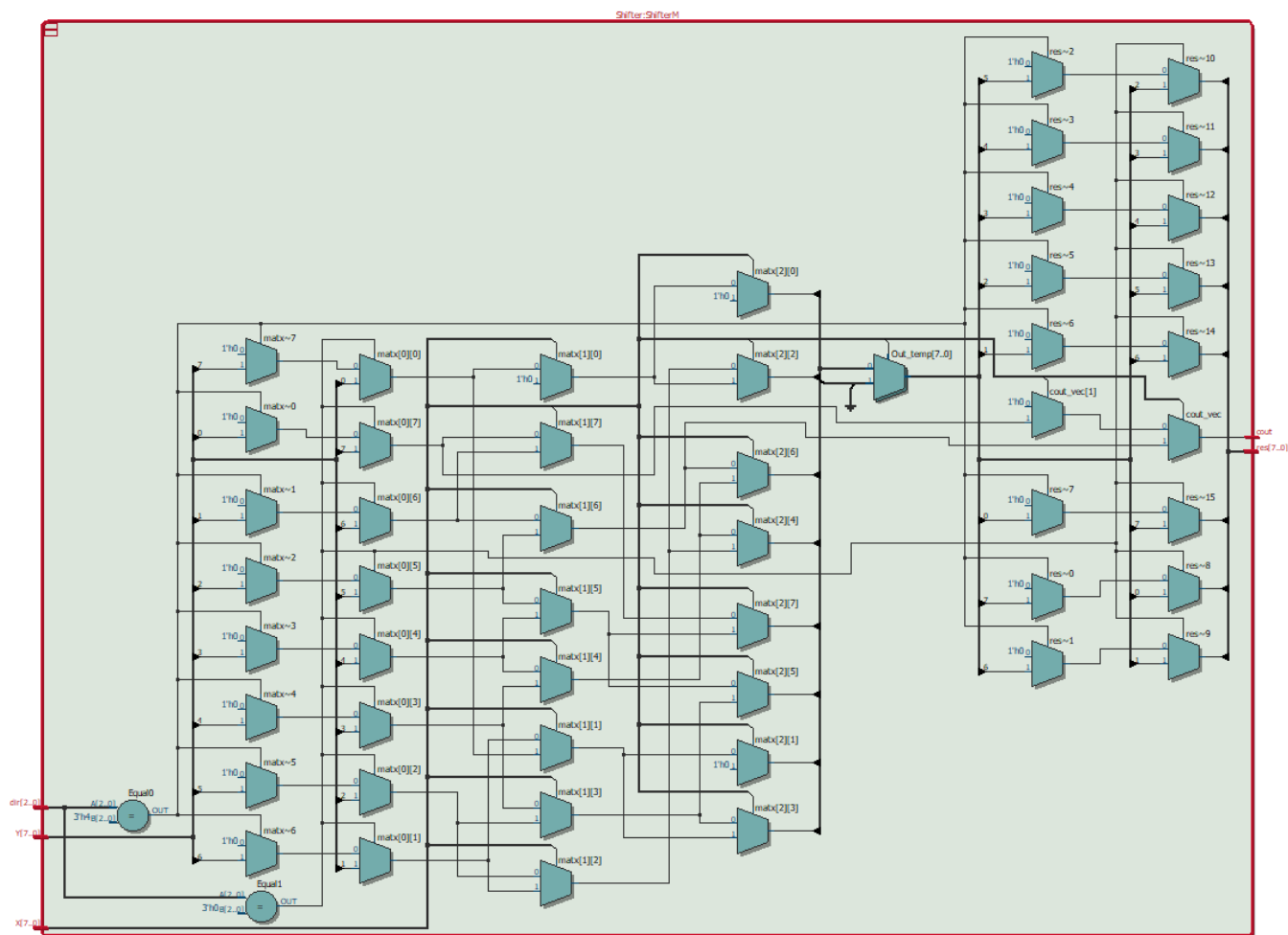
איור 8 : RTL של entity של
AddSuber



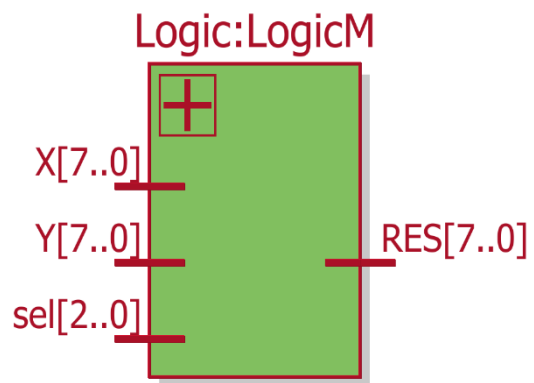
איור 9 : RTL view של מודל הAddSuber



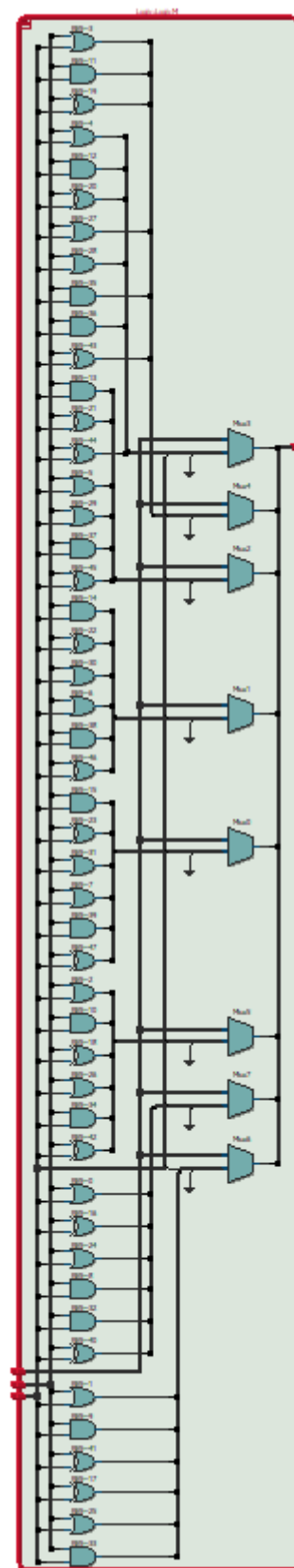
איור 10 : RTL של entity של ה-Shifter



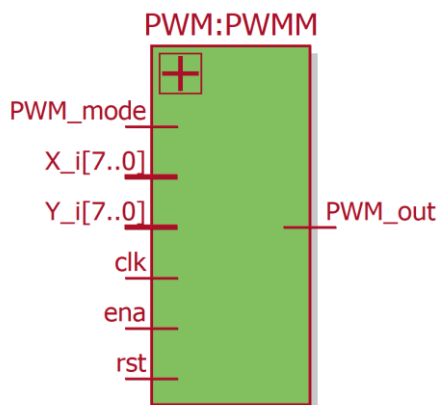
איור 11 : RTL view של מודל ה-Shifter



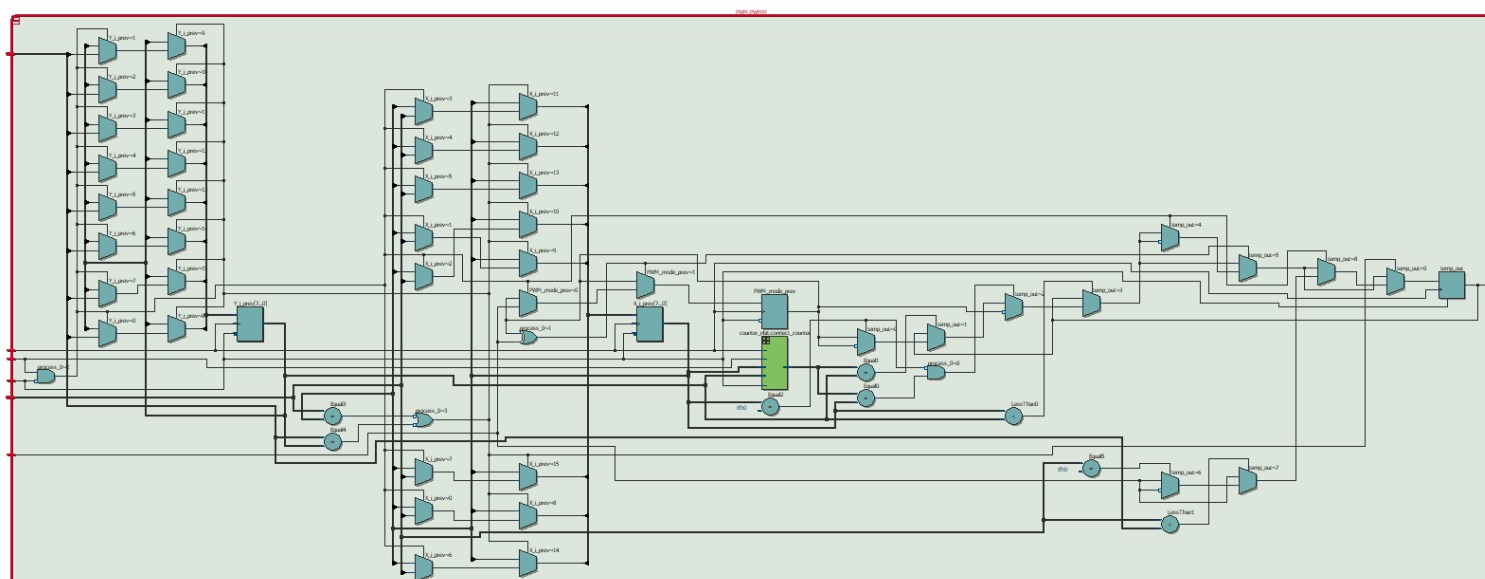
איור 12 : RTL של entity של LogicM



איור 13 : RTL view של מודל LogicM



איור 14 : RTL של entity של PWM



איור 15 : RTL view של מודל ה-PWM

מציאת f_{MAX} :

על מנת לבצע אופטימיזציה של הקוד ולקבל תדר טוב יותר, לכן ננסה להימנע מ Latches בקוד ובנוסף נשים לכניסות ולמוצאים את הפינים הקרובים ביותר שהתוכנה מציעה ב-location fitter. לאחר ביצוע השלבים נקמפל את המודל בתוכנה ונוציא את התדר המקסימלי:

Slow 1100mV 85C Model Fmax Summary				
	Fmax	Restricted Fmax	Clock Name	Note
1	121.45 MHz	121.45 MHz	clk	

איור 16b: תדר מקסימלי עבור 85C

Slow 1100mV 0C Model Fmax Summary				
	Fmax	Restricted Fmax	Clock Name	Note
1	120.74 MHz	120.74 MHz	clk	

איור 16a: תדר מקסימלי עבור 0C

Analysis & Synthesis Resource Usage Summary		
	Resource	Usage
1	Estimate of Logic utilization (ALMs needed)	158
2		
3	✓ Combinational ALUT usage for logic	217
1	-- 7 input functions	2
2	-- 6 input functions	93
3	-- 5 input functions	50
4	-- 4 input functions	24
5	-- <=3 input functions	48
4		
5	Dedicated logic registers	73
6		
7	I/O pins	37
8	Total DSP Blocks	0
9	Maximum fan-out node	ALUFN_i[3]~input
10	Maximum fan-out	91
11	Total fan-out	1355
12	Average fan-out	3.72

איור 17: Logic usage של המערכת

עבור מודל הAdderSuber:

Analysis & Synthesis Resource Usage Summary		
	Resource	Usage
1	Estimate of Logic utilization (ALMs needed)	20
2		
3	▼ Combinational ALUT usage for logic	28
1	-- 7 input functions	0
2	-- 6 input functions	5
3	-- 5 input functions	9
4	-- 4 input functions	7
5	-- <=3 input functions	7
4		
5	Dedicated logic registers	28
6		
7	I/O pins	31
8	Total DSP Blocks	0
9	Maximum fan-out node	clk~input
10	Maximum fan-out	28
11	Total fan-out	276
12	Average fan-out	2.34

איור 18: Logic usage של מודל הAdderSuber

עבור מודל הShifter:

Analysis & Synthesis Resource Usage Summary		
	Resource	Usage
1	Estimate of Logic utilization (ALMs needed)	26
2		
3	▼ Combinational ALUT usage for logic	32
1	-- 7 input functions	0
2	-- 6 input functions	17
3	-- 5 input functions	3
4	-- 4 input functions	3
5	-- <=3 input functions	9
4		
5	Dedicated logic registers	23
6		
7	I/O pins	31
8	Total DSP Blocks	0
9	Maximum fan-out node	clk~input
10	Maximum fan-out	23
11	Total fan-out	310
12	Average fan-out	2.65

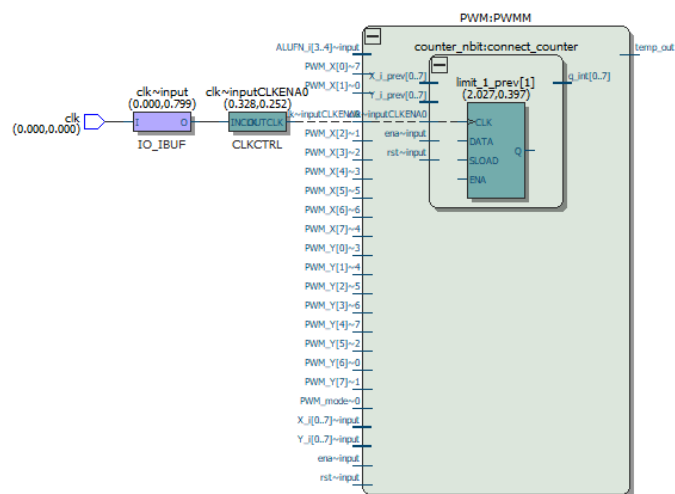
איור 19: Logic usage של מודל הShifter

Analysis & Synthesis Resource Usage Summary		
	Resource	Usage
1	Estimate of Logic utilization (ALMs needed)	14
2		
3	✓ Combinational ALUT usage for logic	9
1	-- 7 input functions	0
2	-- 6 input functions	0
3	-- 5 input functions	8
4	-- 4 input functions	0
5	-- <=3 input functions	1
4		
5	Dedicated logic registers	27
6		
7	I/O pins	30
8	Total DSP Blocks	0
9	Maximum fan-out node	clk~input
10	Maximum fan-out	27
11	Total fan-out	197
12	Average fan-out	2.05

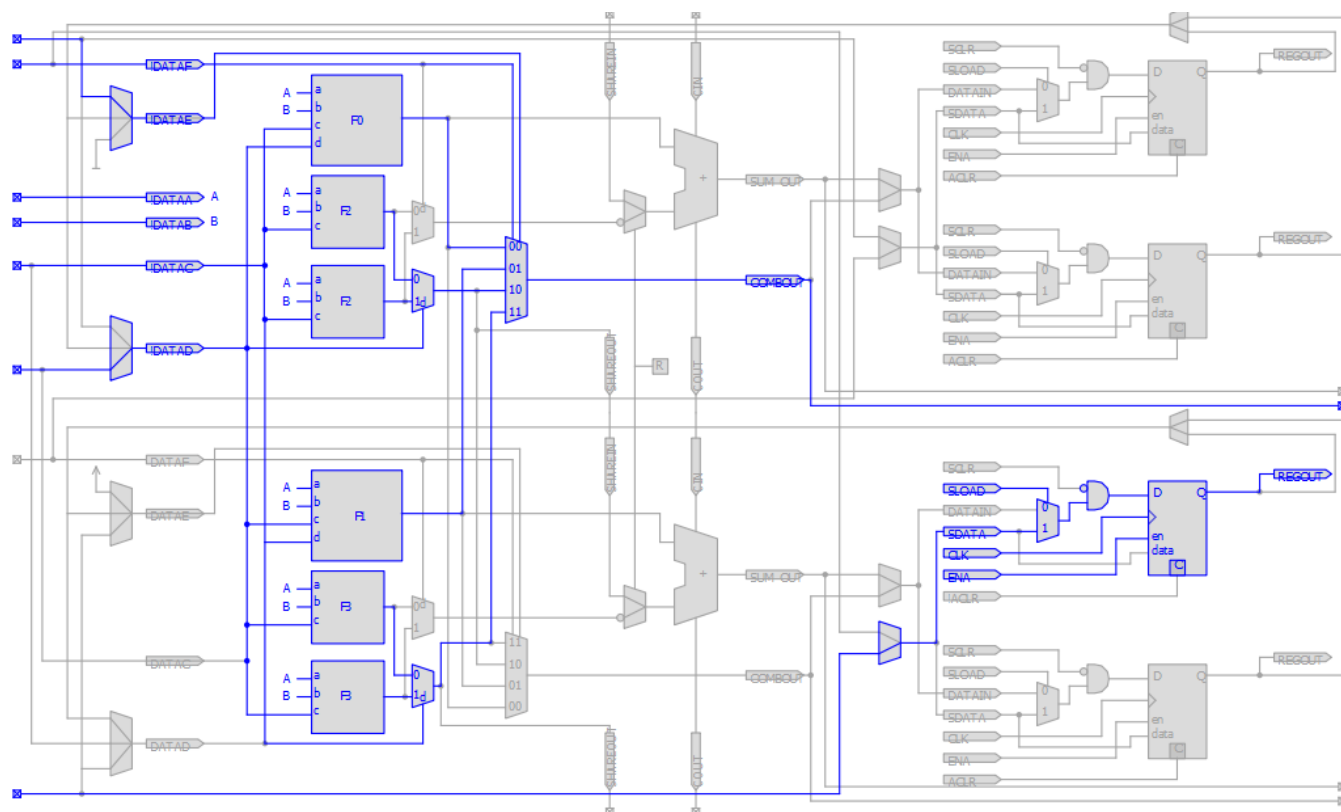
איור 20: Logic usage של מודול הLogic

Analysis & Synthesis Resource Usage Summary		
	Resource	Usage
1	Estimate of Logic utilization (ALMs needed)	44
2		
3	✓ Combinational ALUT usage for logic	60
1	-- 7 input functions	1
2	-- 6 input functions	23
3	-- 5 input functions	8
4	-- 4 input functions	14
5	-- <=3 input functions	14
4		
5	Dedicated logic registers	42
6		
7	I/O pins	21
8	Total DSP Blocks	0
9	Maximum fan-out node	clk~input
10	Maximum fan-out	42
11	Total fan-out	427
12	Average fan-out	2.97

איור 21: Logic usage של מודול הPWM

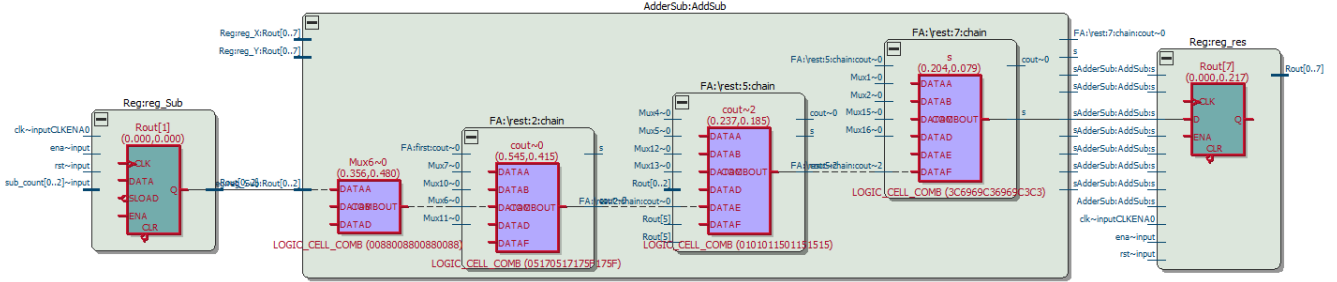


איור 21: הנתיב הקריטי של המערכת כולה



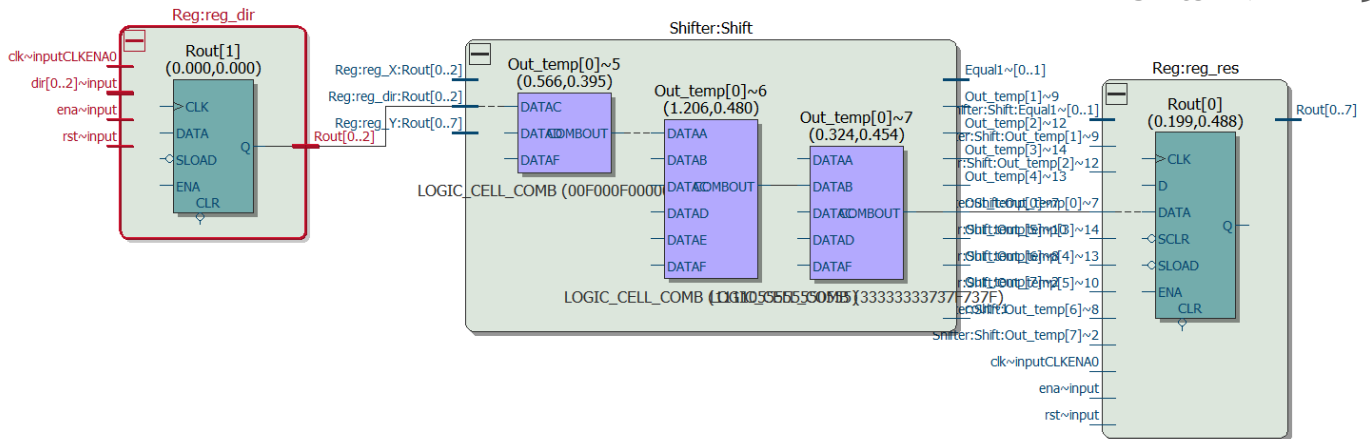
איור 22: הנתיב הקריטי של המערכת כולה בתצוגת מפה ותצורת Property Editor

עבור מודל הAddSub:



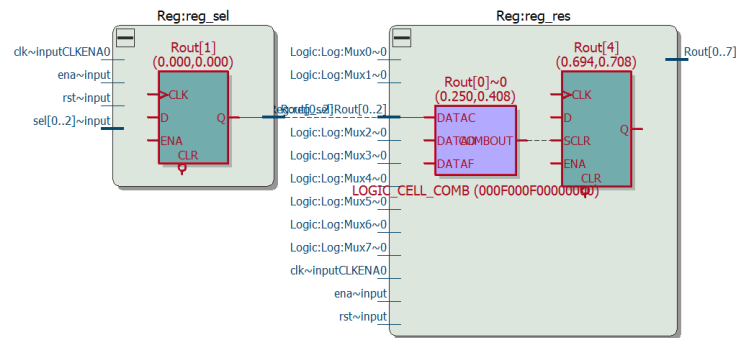
איור 23: הנתב הקריטי של מודל הAddSub

עבור מודל הShifter:



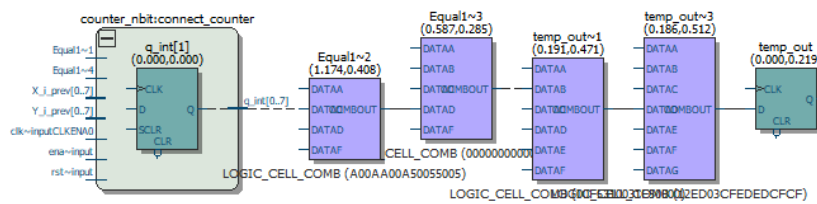
איור 24: הנתב הקריטי של מודל הShifter

עבור מודל הLogic:



איור 25: הנתב הקריטי של מודל הLogic

עבור מודל הPWM:



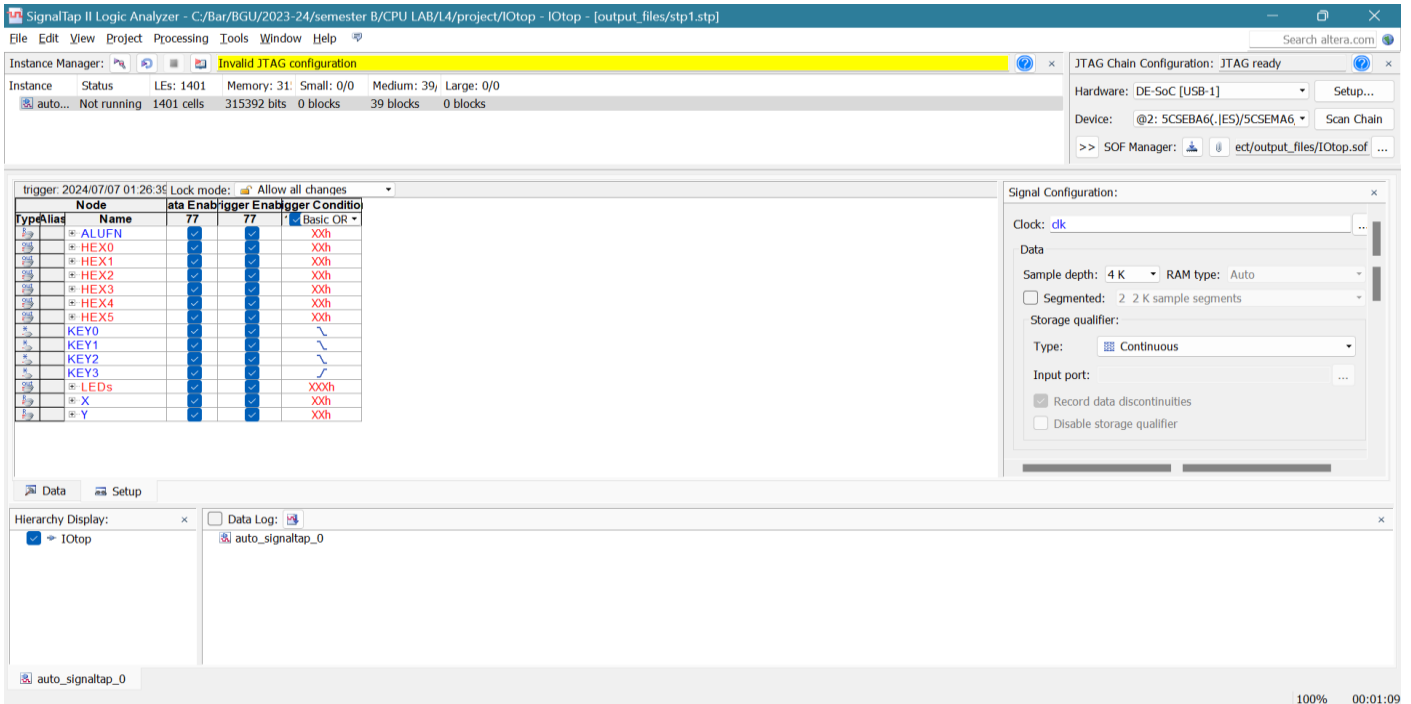
איור 26: הנתב הקריטי של מודל הPWM

:Single Tap

על מנת לבצע וורייקציה של החומרה נשתמש בפונקציית ה-tap signal של ה-Quartus. נתפוס בזמן אמת את מצב הסיגנלים של הרכיב, ובהתאם לסיגנל שאותו נרצה לתפוס, ברגע שהסיגנל ישתנה למה שאנחנו רוצים נקבל את תוצאות הסיגנלים שנדפיס למסך. נכניס את הסיגנלים הרצויים, הסיגנלים שאנחנו לוכדים הם ה-Keys שהם במצב Down Pull לכן נתפוס אותם בירידת מתח, ונדפיס את הכניסות והמוצאים של המערכת. בנוסף נשים את תנאי הלכידה כ Basic OR, כלומר שרק מספיק שאחד משתנה ולא כל הסיגנלים על מנת שתתרחש הלכידה.

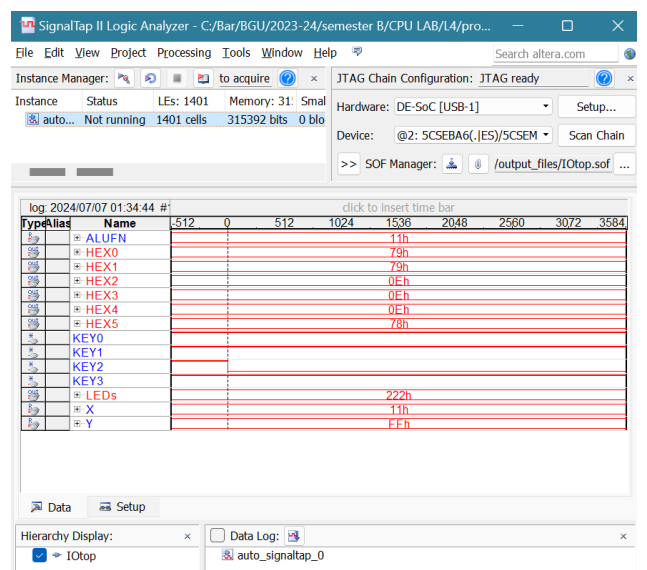
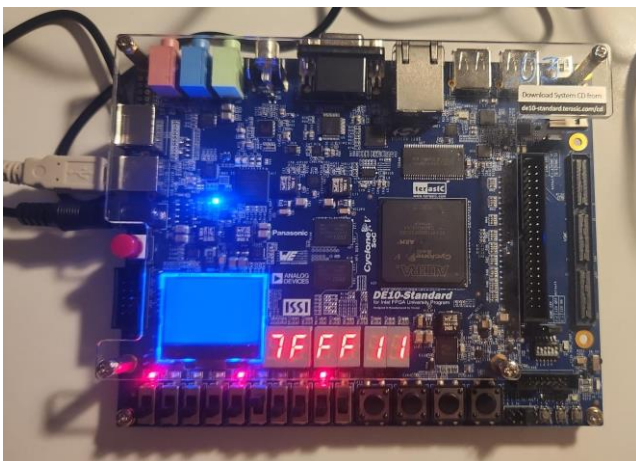
פעולה אריתמטית – חיסור:

נגדיר את הSTP:



איור 27: הגדרת STP עבור פעולת חיסור

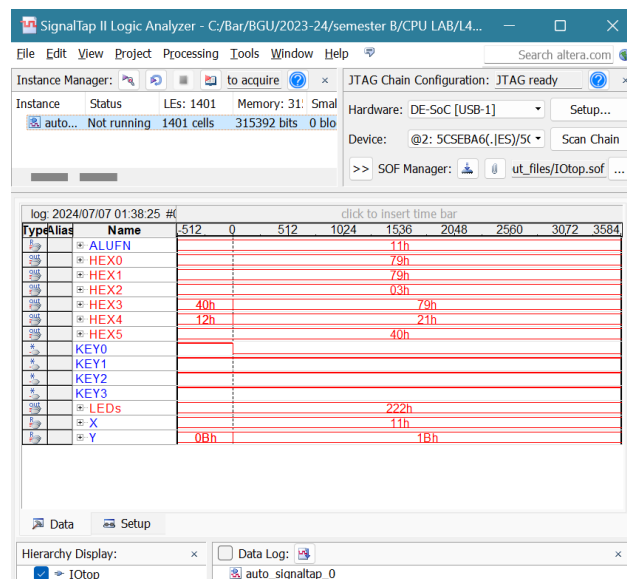
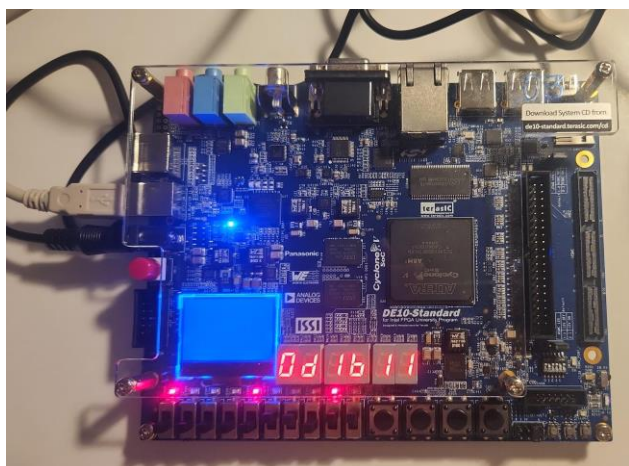
נאתחל את X ל17:



איור 28: איחול X ל17

ניתן לראות כי בלחיצה על KEY2 (ירד לו) קיבלנו את הליכדה וכי X התעדכן לערך שהזנו. בנוסף ניתן לראות כי Y לא התעדכן וכך גם הALUFN.

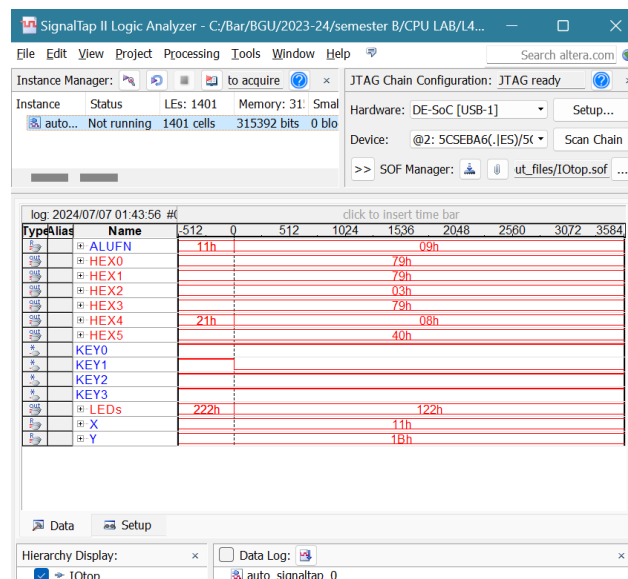
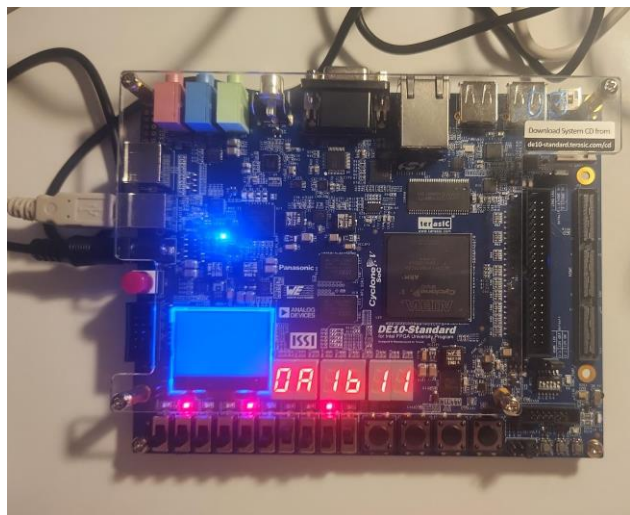
נאתחל את Y ל27:



איור 29: איתחול Y ל27

גם כאן ניתן לראות כי בלחיצה על כפתור KEY0 עדכנו את ערך Y לערך הרצוי וכי X וALUFN לא התעדכנו.

נאתחל את ALUFN להיות פעולת חיסור (01001) ונקבל גם את התוצאה:



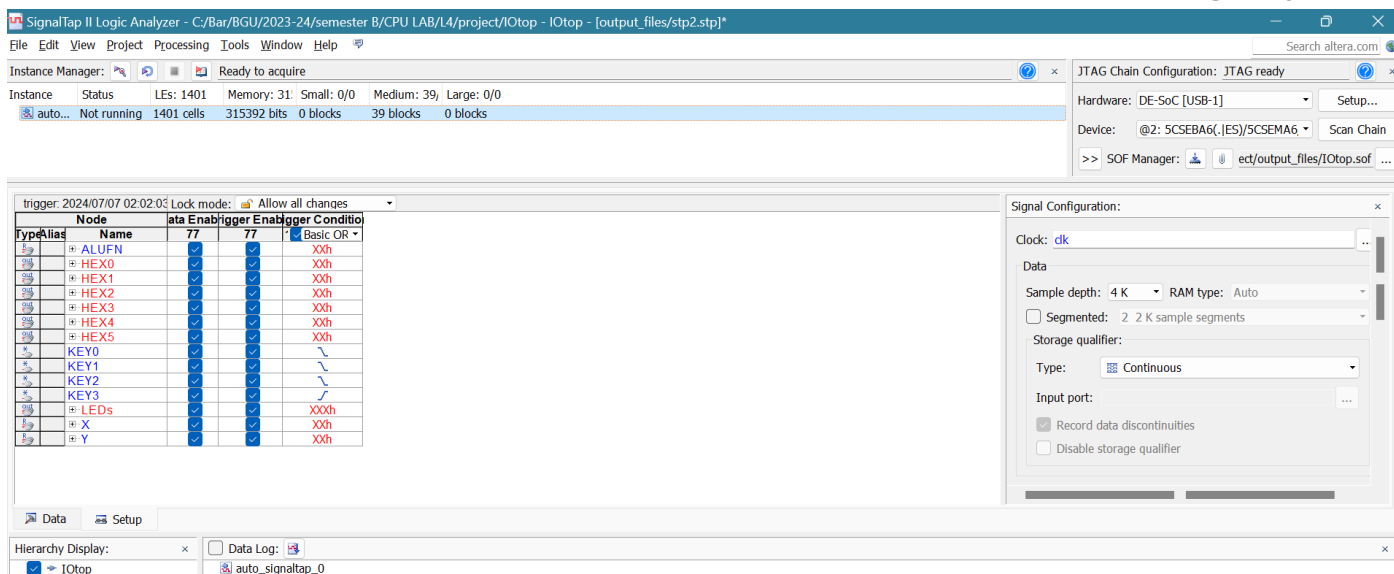
איור 30: איתחול ALUFN וקבלת תוצאת החיסור

ניתן לראות כי בלחיצה על כפתור KEY1 עדכנו את ALUFN את 9 דולקים (כפי שהיינו רוצים) וקיבלנו את התוצאה $0x0A = 10 = 27 - 17$ כפי שציפינו. בנוסף ניתן לראות כי הדגלים (0 עד 3 LEDr) דולקים כפי היינו מצפים (קיבלנו מספר חיובי ממש, לא היה over flow וקיבלנו carry שכן בפעולה זו נקבל carry).

פעולת הזהה – הזהה שמאלה:

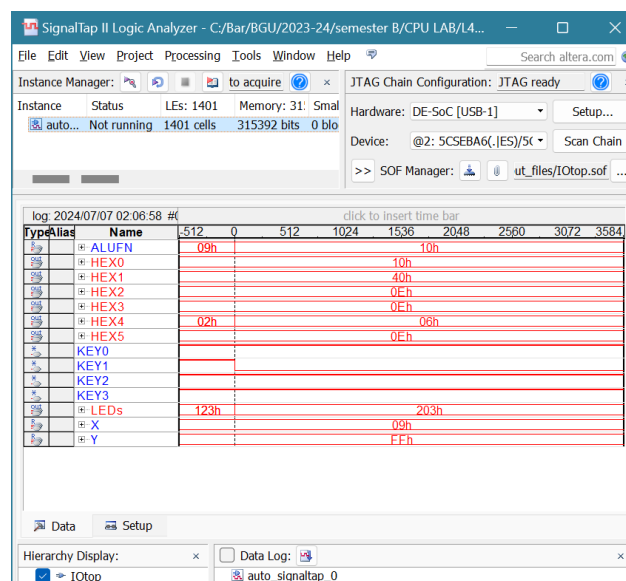
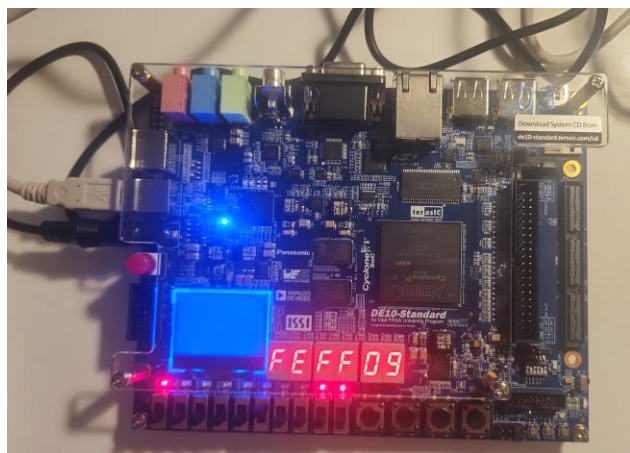
כעת נבצע פעולת הזהה של Y שמאלה X(2 dwonto 0) פעמים:

נגדיר את STP:



איור 31: הגדרת STP עבור פעולת הזהה שמאלה

נאתחל את ALUFN הפעם קודם לפעולת הזהה שמאלה (10000):

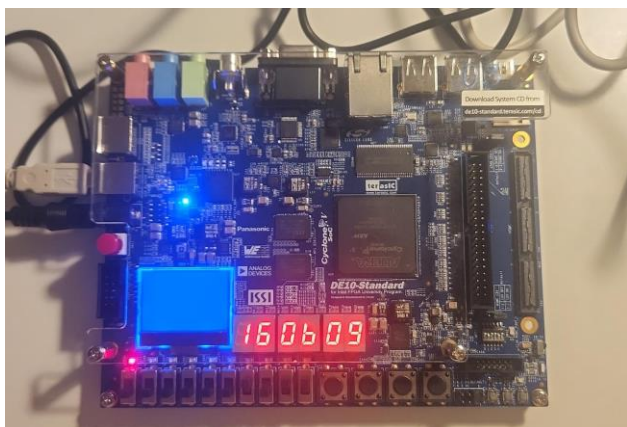


איור 32: איתחול פעולה ALUFN להזהה שמאלה

ניתן לראות שבלחיצת כפתור KEY2 ALUFN מתעדכן לערך שרצינו. בנוסף ניתן לראות כי הדגלים עובדים כפי שהיינו רוצים – המספר המופיע בצג התוצאה 0xFE שלילי והcarry הוא 1 (כל הזהה של כל מספר אי זוגי 255 פעמים תיתן לנו carry 1).

נאתחל את Y להיות 11:

גם כאן ניתן לראות כי בלחיצה על כפתור KEY0 עדכנו את ערך Y לערך הרצוי וכי X וALUFN לא התעדכנו. (בנוסף ניתן לראות כי הדגלים כבויים כפי שהיינו מצפים מהמצב הזה).



SignalTap II Logic Analyzer - C:/Bar/BGU/2023-24/semester B/CPU LAB/L4...

File Edit View Project Processing Tools Window Help

Instance Manager: to acquire x JTAG Chain Configuration: JTAG ready

Instance Status LEs: 1401 Memory: 31 Small
 auto... Not running 1401 cells 315392 bits 0 blo

Hardware: DE-SoC [USB-1] Setup...
 Device: @2: 5CSEBA6(.IES)/5C Scan Chain
 >> SOF Manager: ut_files/IOTop.sof ...

log: 2024/07/07 02:14:06 #

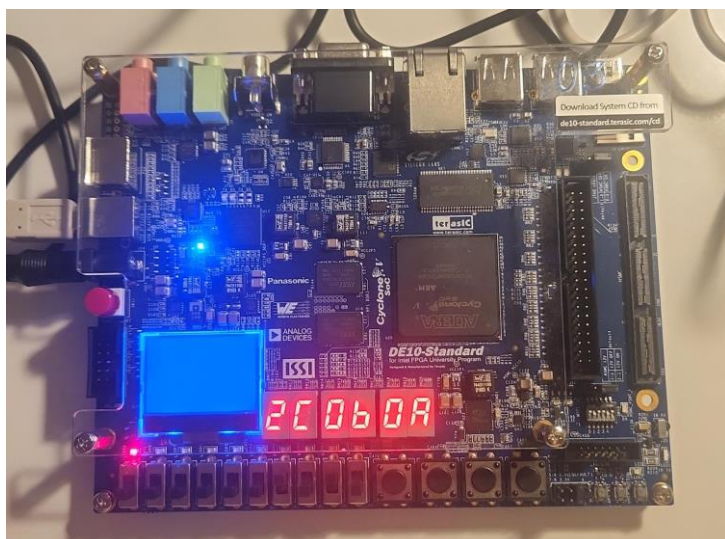
Type/alias	Name	512	0	512	1024	1536	2048	2560	3072	3584
ALUFN						10h				
HEX0						10h				
HEX1						40h				
HEX2				0Eh			03h			
HEX3				0Eh			40h			
HEX4				06h			02h			
HEX5				0Eh			79h			
KEY0										
KEY1										
KEY2										
KEY3										
LEDs			203h				200h			
X							09h			
Y			FFh				0Bh			

Data Setup

Hierarchy Display: x Data Log: x
 IOTop auto_signalsnap_0

איור 33: איתחול Y ל11

נאתחל את X להיות 10 ונקבל את התוצאה של הזזה שמאלה של 11 פעמיים:



SignalTap II Logic Analyzer - C:/Bar/BGU/2023-24/semester B/CPU LAB/L4...

File Edit View Project Processing Tools Window Help

Instance Manager: to acquire x JTAG Chain Configuration: JTAG ready

Instance Status LEs: 1401 Memory: 31 Small
 auto... Not running 1401 cells 315392 bits 0 blo

Hardware: DE-SoC [USB-1] Setup...
 Device: @2: 5CSEBA6(.IES)/5C Scan Chain
 >> SOF Manager: ut_files/IOTop.sof ...

log: 2024/07/07 02:19:15 #

Type/alias	Name	512	0	512	1024	1536	2048	2560	3072	3584
ALUFN						10h				
HEX0			10h				08h			
HEX1							40h			
HEX2							03h			
HEX3							40h			
HEX4			02h				46h			
HEX5			79h				24h			
KEY0										
KEY1										
KEY2										
KEY3										
LEDs							200h			
X			09h				0Ah			
Y							0Bh			

Data Setup

Hierarchy Display: x Data Log: x
 IOTop auto_signalsnap_0

איור 34: איתחול X ל10 וקבלת התוצאה הסופית

ניתן לראות שבלחיצה על KEY2 עדכנו את X להיות 10 וקיבלנו את התוצאה שציפינו לה – $11 \times 2 = 44$ = הזזה שמאלה של 11 פעמיים.