מעבדת ארכיטקטורת מעבדים מתקדמת ומאיצי חומרה

פרויקט גמר

מיכאל קדיק 316645415
עמית ניסני 205664345

תיאור הפרויקט

בפרויקט זה התבקשנו לייצר CPU MIPS based MCU המבוסס על Single Cycle MIPS, המשתמש בMemory Mapped I/O ובעל תמיכה בפסיקות. לשם כך, יצרנו יחידות כמו GPIO, INTCTL ו-Timer בכדי לאפשר את כל הנדרש.

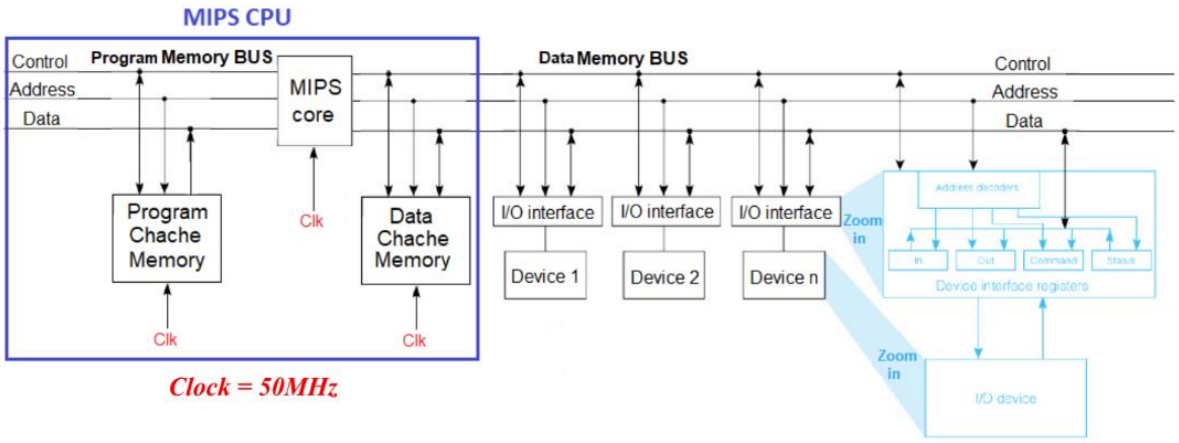

Figure 1. System architecture

את ה-GPIO מימשנו עפ"י הנדרש בפרויקט כפי שצוין באיור הבא.

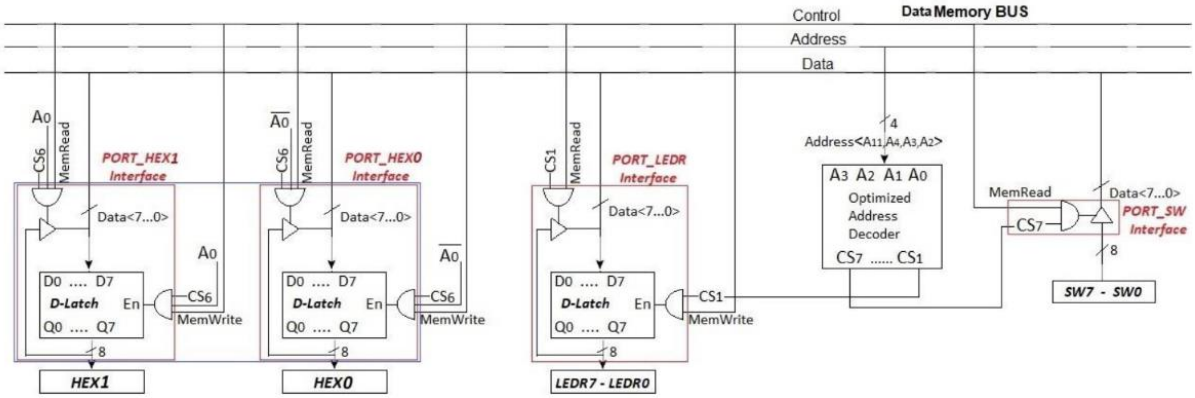

Figure 2. GPIO peripheral connection using Memory Mapped I/O

את ה-Timer מימשנו כפי שנדרש בהגדרת הפרויקט כך שהוא תומך בפסיקות ובהוצאת אות PWM ע"י שימוש בoutput compare mode כפי שניתן לראות באיור הבא.

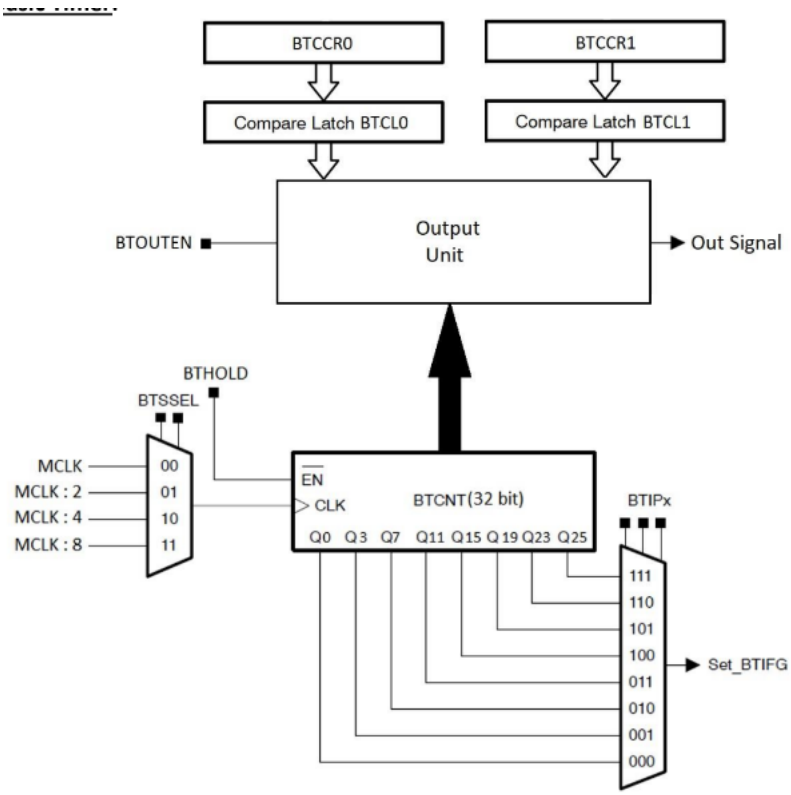

Figure 3. Basic Timer design

את התמיכה בפסיקות מימשנו ע"י רכיב INTCTL עפ"י הדרישות בהגדרת הפרויקט כפי שניתן לראות באיור הבא.

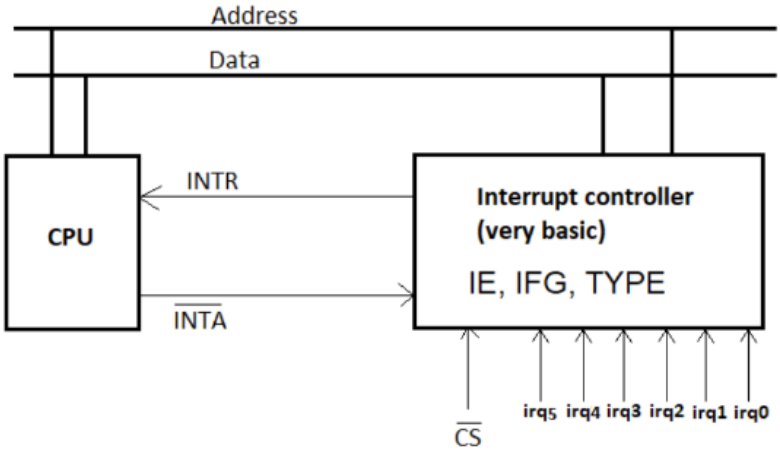

Figure 4. Interrupt controller design

בכדי ליצור את כל המצוין למעלה, השתמשנו ברכיבי חומרה עליהם נפרט באיורים הבאים.

Top

רכיב זה הינו ה- top level entity שלנו ומטרתו לסנכרן בין המעבד, רכיב ה-IO וה-BUS ולשמש כמעטפת חיצונית גם כן.

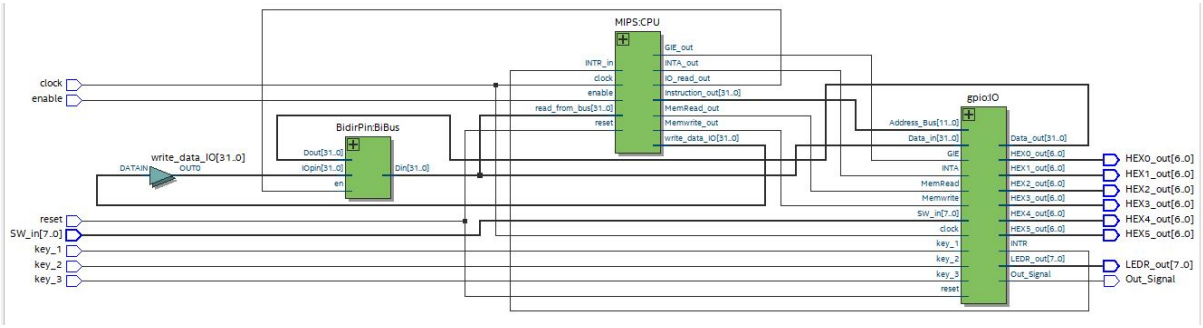

Figure 5. RTL Viewer of MIPS

Timing Models	Final
Logic utilization (in ALMs)	1,624 / 41,910 (4 %)
Total registers	1657
Total pins	65 / 499 (13 %)
Total virtual pins	0
Total block memory bits	90,112 / 5,662,720 (2 %)
Total DSP Blocks	2 / 112 (2 %)
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 %)

Figure 6. Logic Usage

MIPS

רכיב זה הינו ה- CPU ותפקידו לסנכרן בין כל רכיבי החומרה השונים המפורטים בהמשך הדו"ח (IFETCH,IDECODE,CONTROL,EXECUTE,DMEMORY).

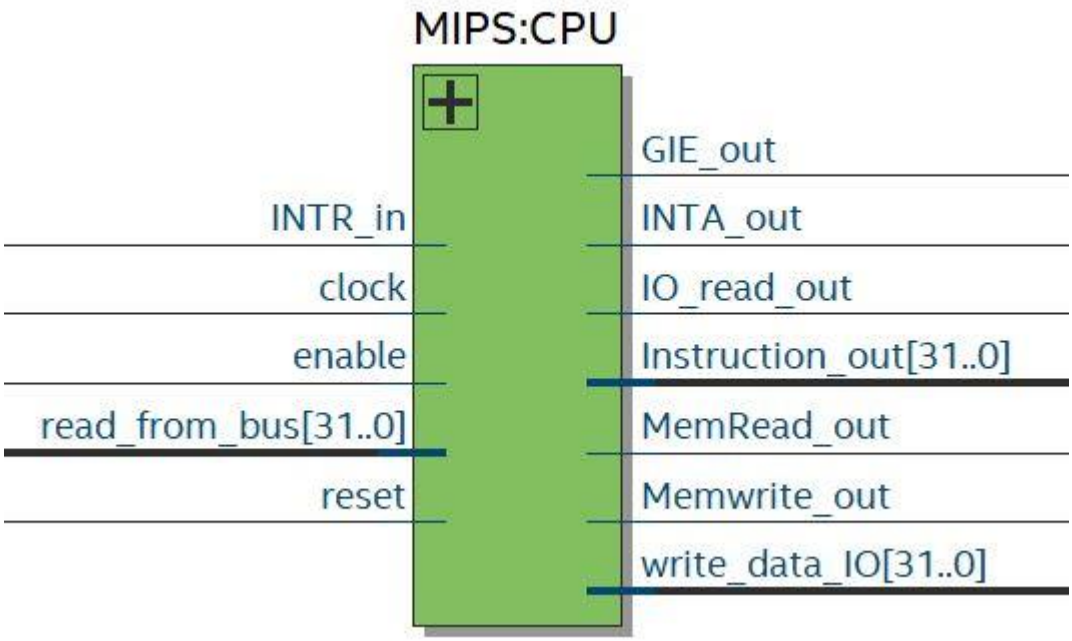

Figure 7. Graphical description of MIPS

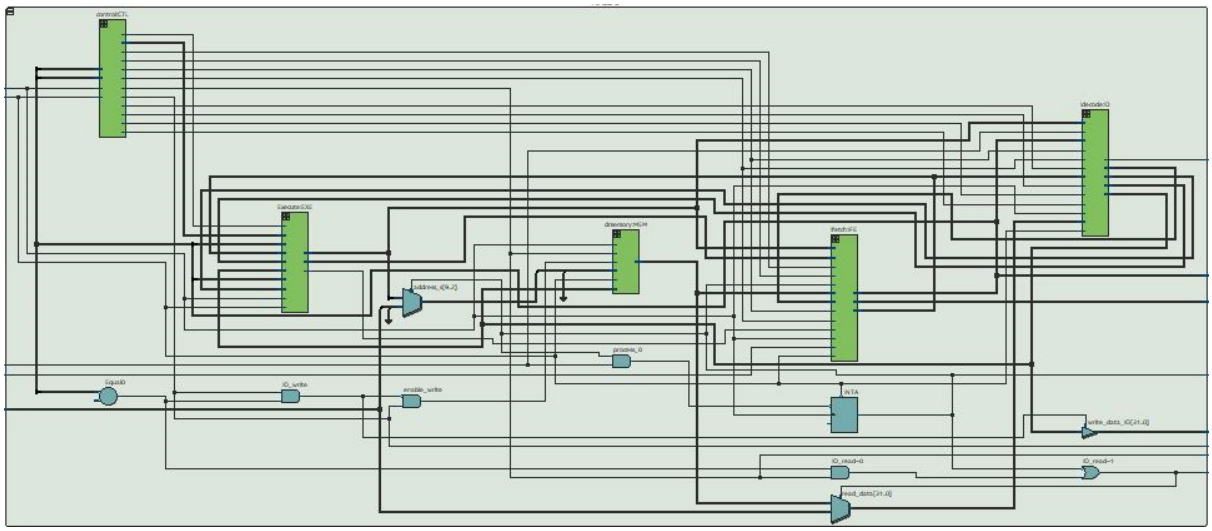

Figure 8. RTL Viewer of MIPS

Name	Direction	Size	Functionality
clock	IN	1	שעון
reset	IN	1	אתחול
enable	IN	1	אפשור PC
INTR_in	IN	1	בקרה
read_from_bus	IN	32	קריאת מידע מה-BUS
IO_read_out	OUT	1	בקרה
write_data_IO	OUT	32	כתיבת מידע אל ה-BUS
Instruction_out	OUT	32	פקודה שמתבצעת כרגע
MemRead_out	OUT	1	בקרה
Memwrite_out	OUT	1	בקרה
INTA_out	OUT	1	בקרה
GIE_out	OUT	1	בקרה

Table 1. Port Table of MIPS

IFETCH

תפקידו של רכיב זה הוא להביא את הפקודה שיש לבצע ולקדם את ה-pc לפקודה הבאה (שאינה בהכרח הפקודה העוקבת, כמו למשל אם יש jump).

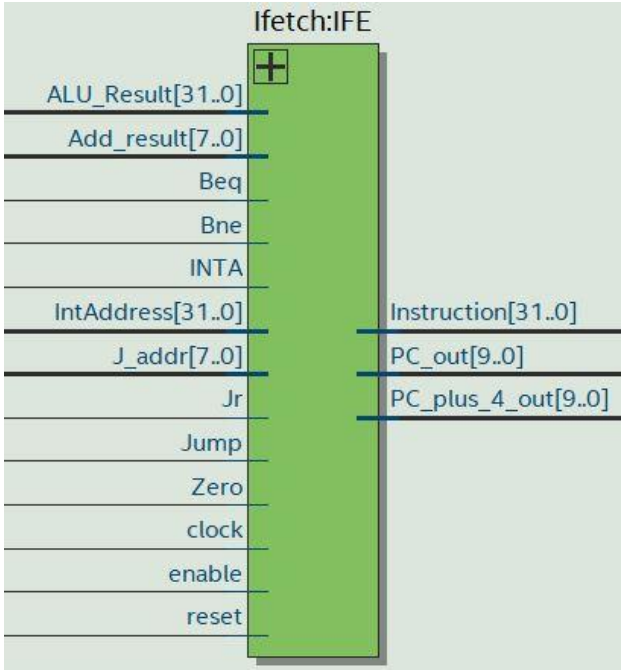

Figure 9. Graphical description of IFETCH

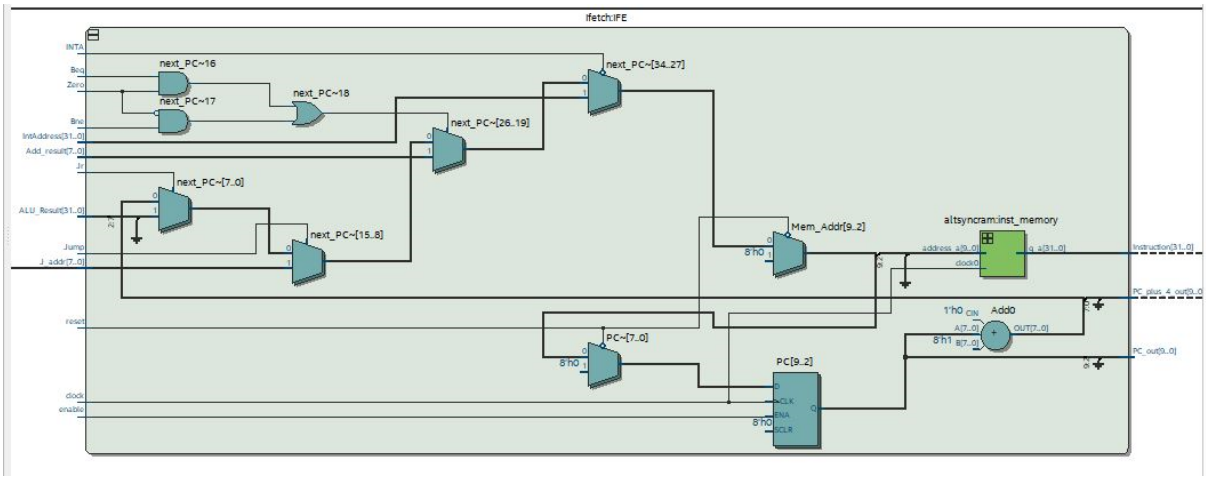

Figure 10. RTL Viewer of IFETCH

Name	Direction	Size	Functionality
clock	IN	1	שעון
reset	IN	1	אתחול
enable	IN	1	אפשור PC
Add_result	IN	8	כתובת הסתעפות
ALU_Result	IN	32	משמש לפקודת קפיצה
IntAddress	IN	32	כתובת למיקוף בוקטור הפסיקה
J_addr	IN	8	כותבת קפיצה
INTA	IN	1	בקרה
Jump	IN	1	בקרה
Zero	IN	1	בקרה
Beq	IN	1	בקרה
Bne	IN	1	בקרה
Jr	IN	1	בקרה
Instruction	OUT	32	פקודה לביצוע
PC_plus_4_out	OUT	10	כתובת של פקודה הבאה
PC_out	OUT	10	כתובת של פקודה נוכחית

Table 2. Port Table of IFETCH

IDECODE

תפקידו של רכיב זה הוא לפענח את הפקודה שיש לבצע, בנוסף ברכיב זה נמצא הregister file ולכן מתבצעת בו גם כתיבה לרגיסטרים.

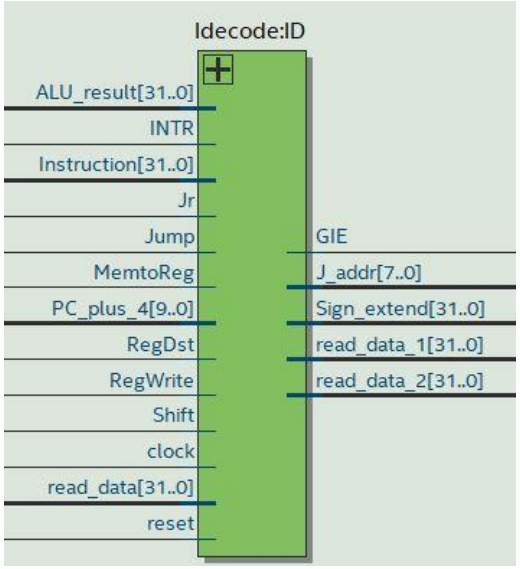

Figure 11. Graphical description of IDECODE

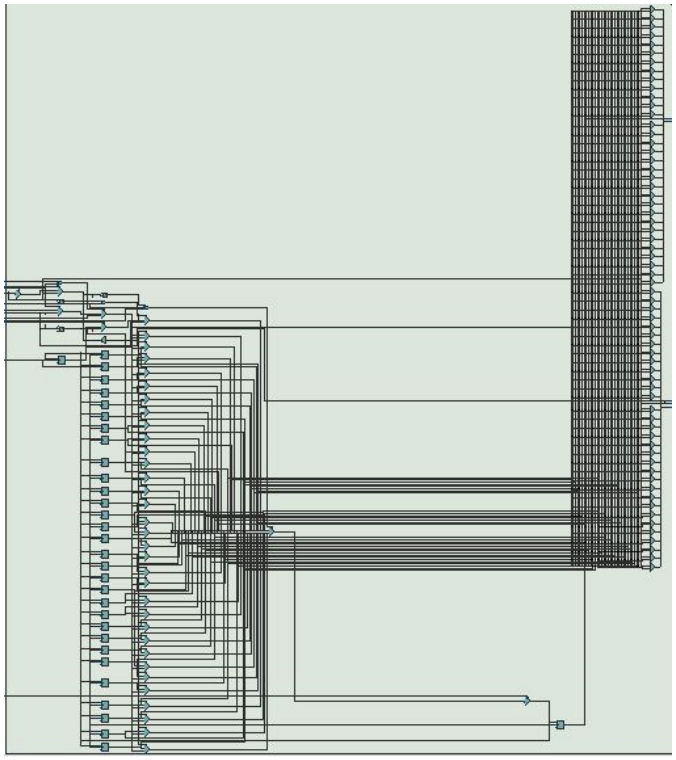

Figure 12. RTL Viewer of IDECODE

Name	Direction	Size	Functionality
clock	IN	1	שעון
reset	IN	1	אתחול
Instruction	IN	32	הפקודה הנוכחית לפענוח
read_data	IN	32	המידע בכתיבה לרגיסטרים
ALU_result	IN	32	תוצאת הפעולה ב- ALU
RegWrite	IN	1	בקרה
MemtoReg	IN	1	בקרה
RegDst	IN	1	בקרה
Jump	IN	1	בקרה
Jr	IN	1	בקרה
Shift	IN	1	בקרה
INTR	IN	1	בקרה
PC_plus_4	IN	10	כתובת של הפקודה הבאה לביצוע
read_data_1_out	OUT	32	המידע מרגיסטר 1
read_data_2_out	OUT	32	המידע מרגיסטר 2
Sign_extend	OUT	32	סיגנל מורחב בהתאם לסימן
J_addr	OUT	8	כתובת קפיצה
GIE	OUT	1	בקרה

Table 3. Port Table of IDECODE

CONTROL

תפקידו של רכיב זה הוא לייצר את קווי הבקרה בהתאם לפקודה הנוכחית.

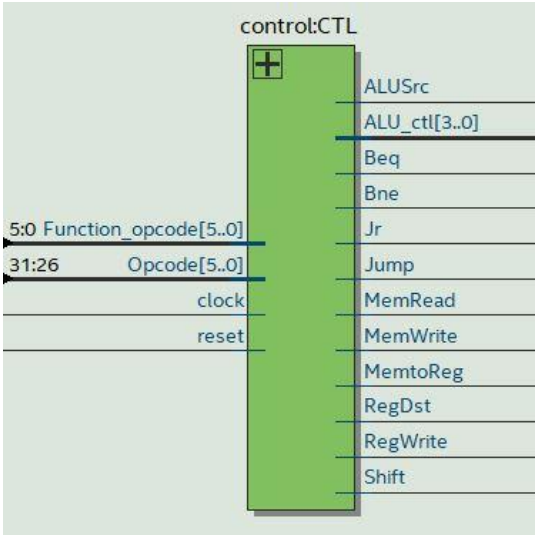

Figure 13. Graphical description of CONTROL

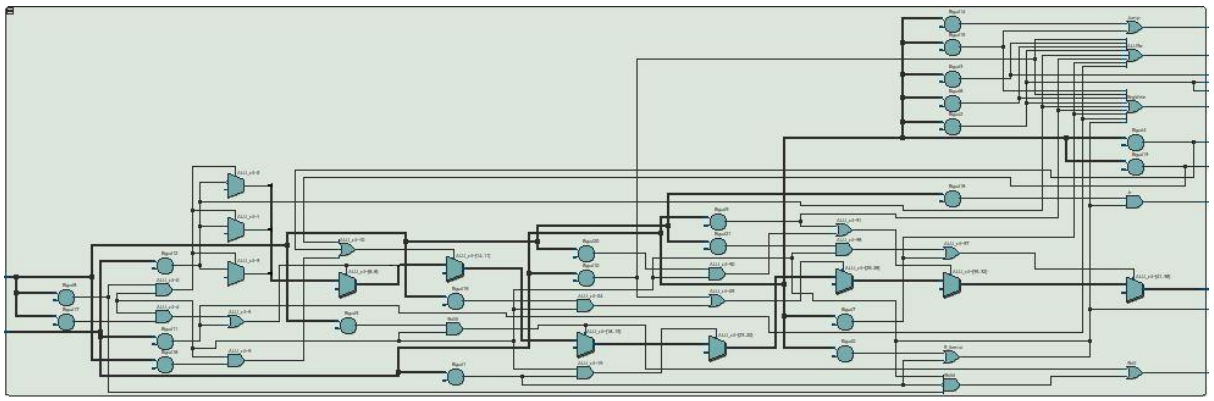

Figure 14. RTL Viewer of CONTROL

Name	Direction	Size	Functionality
clock	IN	1	שעון
reset	IN	1	אתחול
Opcode	IN	6	ביטים 26-31 בפקודה
Function_opcode	IN	6	ביטים 0-5 בפקודה
RegDst	OUT	1	קו בקרה לבחירת כתובת יעד הכתיבה לרגיסטר
ALUSrc	OUT	1	קו בקרה הבורר איזה מידע יכנס לכניסה B ב-ALU
MemtoReg	OUT	1	קו בקרה הבורר בין המידע שיכתב לתוך הרגיסטרים (מהזכרון או מה-ALU)
RegWrite	OUT	1	קו בקרה הבורר את כתובת הרגיסטר לכתיבה
MemRead	OUT	1	קו בקרה המבחין האם יש לקרוא מהזכרון
MemWrite	OUT	1	קו בקרה המבחין האם יש לכתוב לזכרון
Shift	OUT	1	קו בקרה המבחין האם יש פעולת shift
Jump	OUT	1	קו בקרה המבחין האם יש פעולת קפיצה
Jr	OUT	1	קו בקרה המבחין האם יש פעולת קפיצה לתוך ערך השמור ברגיסטר
Beq	OUT	1	קו בקרה המבחין האם יש פעולת הסתעפות
Bne	OUT	1	קו בקרה המבחין האם יש פעולת הסתעפות
ALU_ctl	OUT	4	קו בקרה הבורר איזה פעולה יש לעשות ב-ALU

Table 4. Port Table of CONTROL

EXECUTE

תפקידו של רכיב זה הוא ביצוע הפעולות הנדרשות בכדי לבצע את הפקודה.

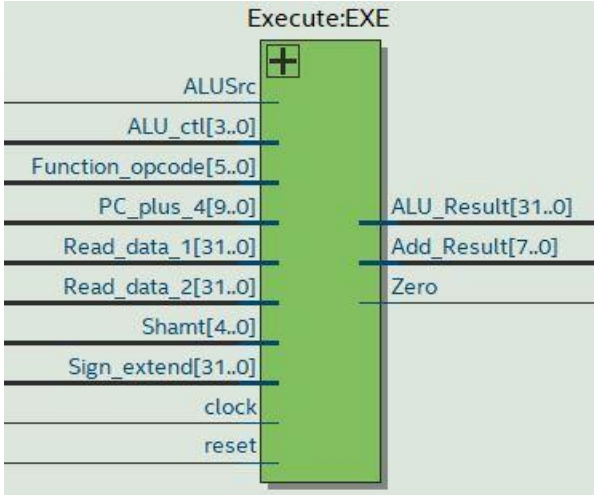

Figure 15. Graphical description of EXECUTE

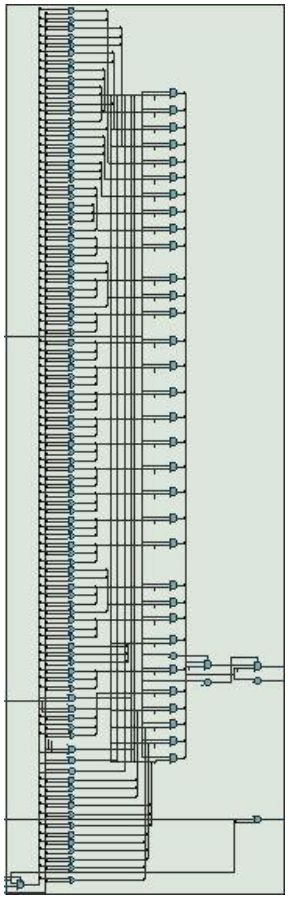

Figure 16. RTL Viewer of EXECUTE

Name	Direction	Size	Functionality
clock	IN	1	שעון
reset	IN	1	אתחול
Read_data_1	IN	32	מידע מרגיסטר 1
Read_data_2	IN	32	מידע מרגיסטר 2
Sign_extend	IN	32	מידע משדה הimmidiate מורחב
Function_opcode	IN	6	ביטים 0-5 של הפקודה
ALUSrc	IN	1	בקרה
PC_plus_4	IN	10	כתובת של הפקודה הבאה לביצוע
Shamt	IN	5	ביטים 6-10 בפקודה
ALU_ctl	IN	4	בורר איזה פקודה יש לבצע ב- ALU
ALU_Result	OUT	32	תוצאת ה-ALU
Add_Result	OUT	8	תוצאת חישוב כתובת ההסתעפות
Zero	OUT	1	בקרה

Table 5. Port Table of EXECUTE

DMEMORY

תפקידו של רכיב זה הוא לבצע קריאה מהזכרון או כתיבה לזכרון.

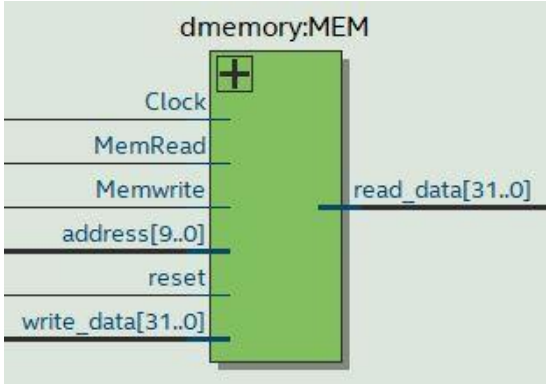

Figure 17. Graphical description of DMEMORY

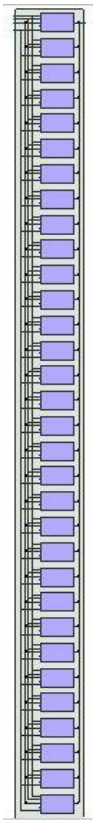

Figure 18. RTL Viewer of DMEMORY

Name	Direction	Size	Functionality
clock	IN	1	שעון
reset	IN	1	אתחול
address	IN	10	כתובת כתיבה לזכרון
write_data	IN	32	מידע שנכתב לזכרון
MemRead	IN	1	בקרה
Memwrite	IN	1	בקרה
read_data	OUT	32	מידע שיוצא מהזכרון

Table 6. Port Table of DMEMORY

GPIO

תפקידו של רכיב זה הוא לסנכרן בין כל רכיבי ה-IO השונים והעברת המידע מהם אל ה-BUS.

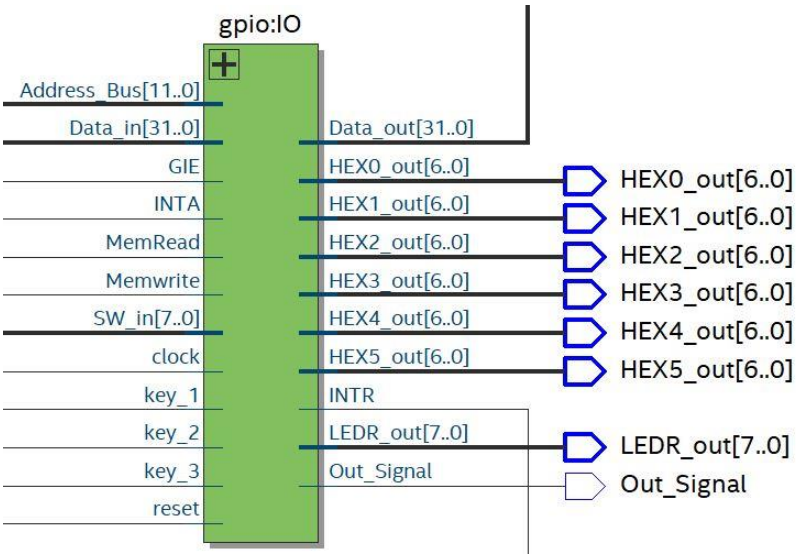

Figure 19. Graphical description of GPIO

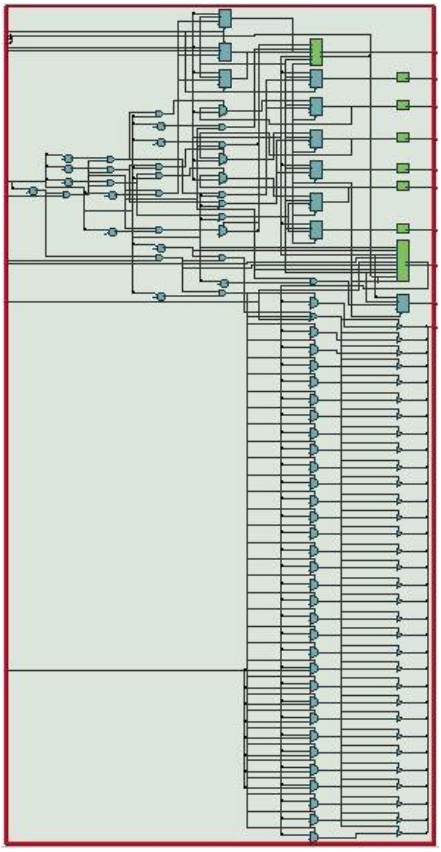

Figure 20. RTL Viewer of GPIO

Name	Direction	Size	Functionality
Data_in	IN	32	מידע שנכנס לרכיב
Address_Bus	IN	12	כתובת
MemRead	IN	1	בקרה
Memwrite	IN	1	בקרה
GIE	IN	1	אות בקרה המאפשר פסיקות גלובליות
INTA	IN	1	אות בקרה המגיע מהCPU ומעיד על אישור פסיקה
reset	IN	1	אתחול
clock	IN	1	שעון
key_1	IN	1	לחצן 1
key_2	IN	1	לחצן 2
key_3	IN	1	לחצן 3
SW_in	IN	8	מתגים
Data_out	OUT	32	מידע שיוצא מהרכיב
HEX0_out	OUT	7	מידע לאחר קידוד שנכתב על ה-HEX
HEX1_out	OUT	7	מידע לאחר קידוד שנכתב על ה-HEX
HEX2_out	OUT	7	מידע לאחר קידוד שנכתב על ה-HEX
HEX3_out	OUT	7	מידע לאחר קידוד שנכתב על ה-HEX
HEX4_out	OUT	7	מידע לאחר קידוד שנכתב על ה-HEX
HEX5_out	OUT	7	מידע לאחר קידוד שנכתב על ה-HEX
LEDR_out	OUT	8	מידע שמופיע על הלדים
INTR	OUT	1	קו בקרה המודיע ל CPU שיש בקשה לפסיקה
Out_Signal	OUT	1	אות PWM

Table 7. Port Table of GPIO

TIMER

תפקידו של רכיב זה הוא לשמש בתור Basic Timer המאפשר יצירת פסיקות בהתאם לקינפוגים שונים, כמו כן ניתן להוציא בעזרתו אות PWM.

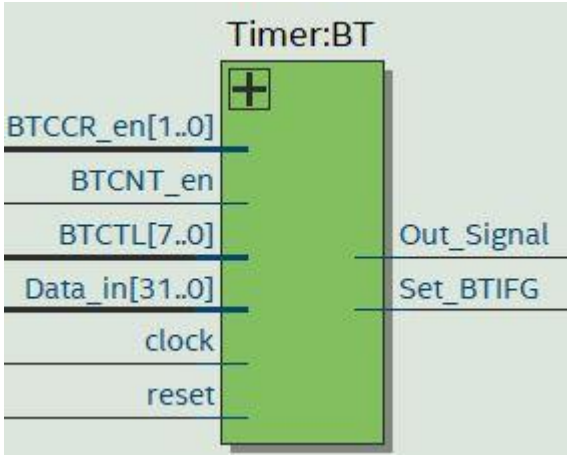

Figure 21. Graphical description of TIMER

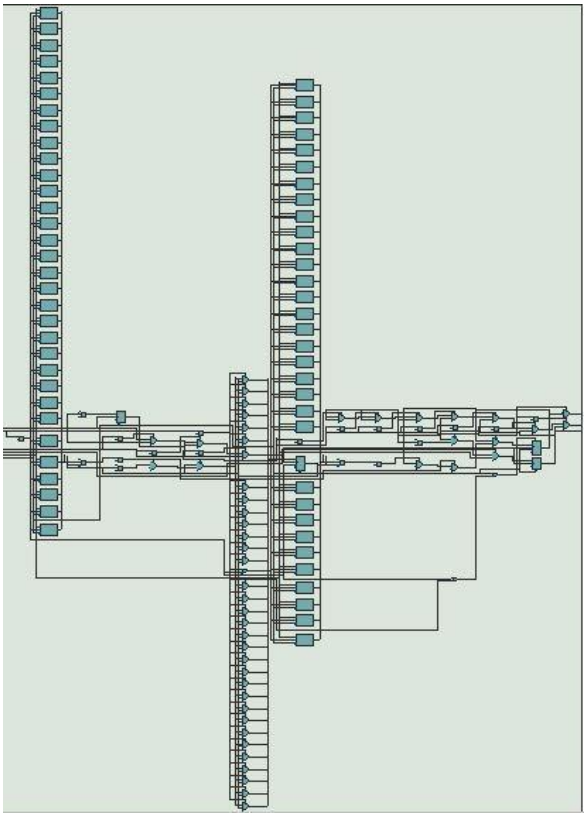

Figure 22. RTL Viewer of TIMER

Name	Direction	Size	Functionality
Data_in	IN	32	מידע הנכנס לרכיב
BTCTL	IN	8	רגיסטר שבעזרתו מקונפג השעון
BTCCR_en	IN	2	אפשור כתיבה לרגיסטרים BTCL0, BTCL1
BTCNT_en	IN	1	אפשור כתיבה לרגיסטר BTCNT
reset	IN	1	אתחול
clock	IN	1	שעון
BTCNT_out	OUT	32	ערך הספירה של הטיימר
Set_BTIFG	OUT	1	פסיקה של השעון
Out_Signal	OUT	1	אות PWM שנוצר ע"י הטיימר

Table 8. Port Table of TIMER

INTCTL

תפקידו של רכיב זה הוא לנהל ולשלוט על כל הפסיקות.

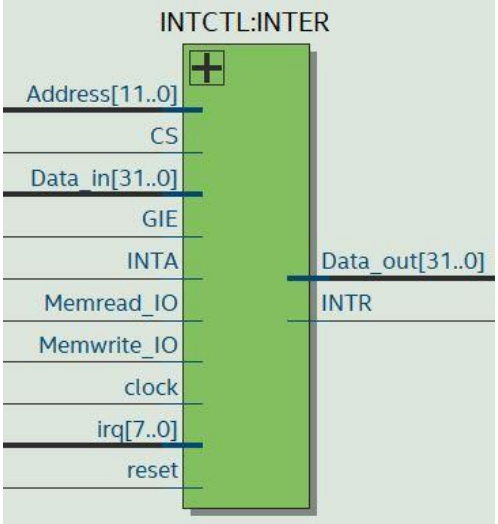

Figure 23. Graphical description of INTCTL

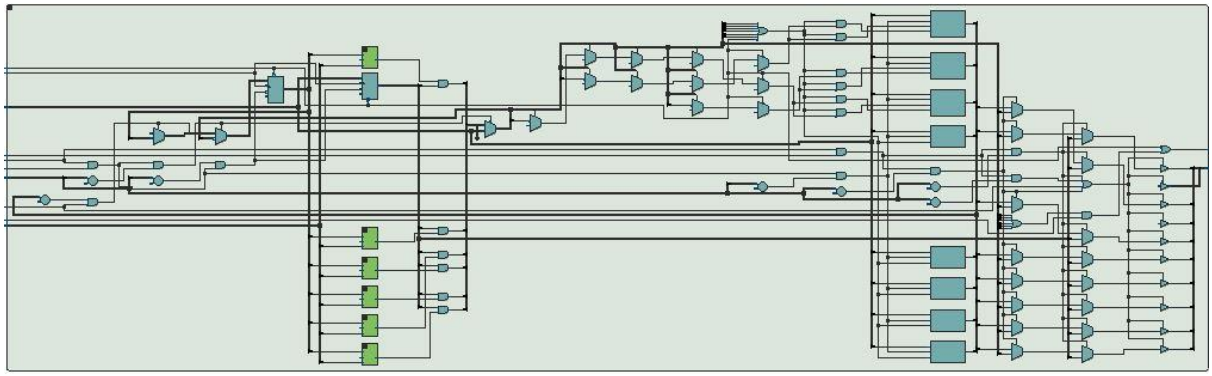

Figure 24. RTL Viewer of INTCTL

Name	Direction	Size	Functionality
Data_in	IN	32	מידע הנכנס לרכיב
Address	IN	12	כתובת
irq	IN	8	מודיע על בקשות פסיקה
INTA	IN	1	בקרה
reset	IN	1	אתחול
clock	IN	1	שעון
cs	IN	1	מעיד על פניה לרכיב
Memwrite_IO	IN	1	בקרה
Memread_IO	IN	1	בקרה
GIE	IN	1	בקרה
Data_out	OUT	32	מידע היוצא מהרכיב
INTR	OUT	1	בקרה

Table 9. Port Table of INTCTL

D_LATCH

רכיב עזר שמשומש בבקר הפסיקות.

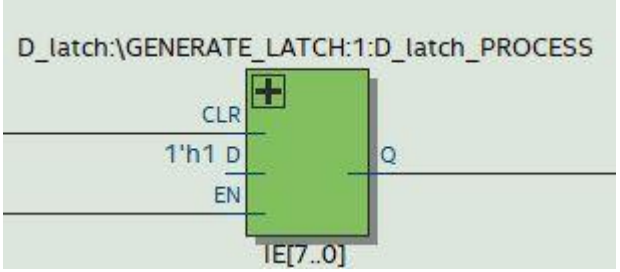

Figure 25. Graphical description of D_LATCH

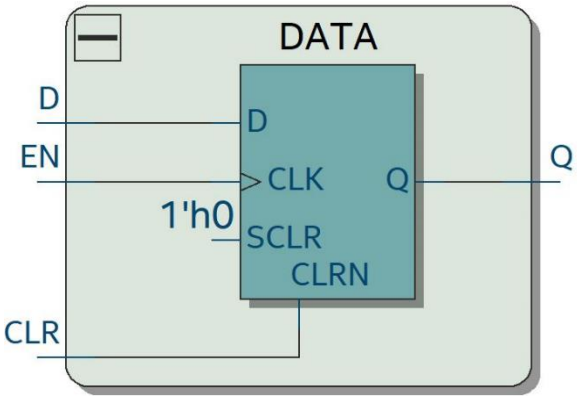

Figure 26. RTL Viewer of D_LATCH

Name	Direction	Size	Functionality
D	IN	1	1 קבוע
EN	IN	1	אפשור כתיבה
CLR	IN	1	איפוס
Q	OUT	1	מוצא

Table 10. Port Table of D_LATCH

HEX

רכיב עזר המשמש לקידוד המידע שמוצג ב-HEX.

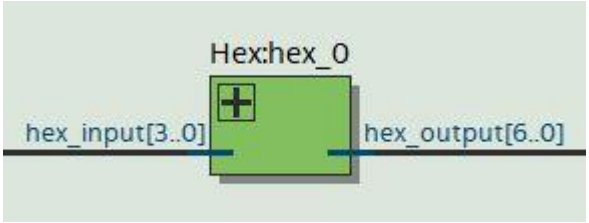

Figure 27. Graphical description of HEX

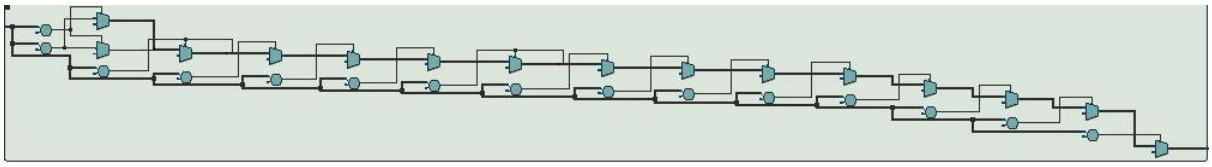

Figure 28. RTL Viewer of HEX

Name	Direction	Size	Functionality
Hex_input	IN	4	ערך שרוצים להציג
Hex_output	OUT	7	המרה להצגה

Table 11. Port Table of HEX

BidirPin

רכיב שתפקידו לשמש כ-Data bus ולהעביר מידע בין ה-CPU ורכיבי ה-IO.

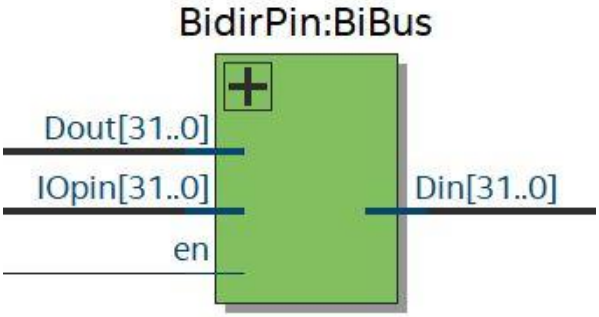

Figure 29. Graphical description of BidirPin

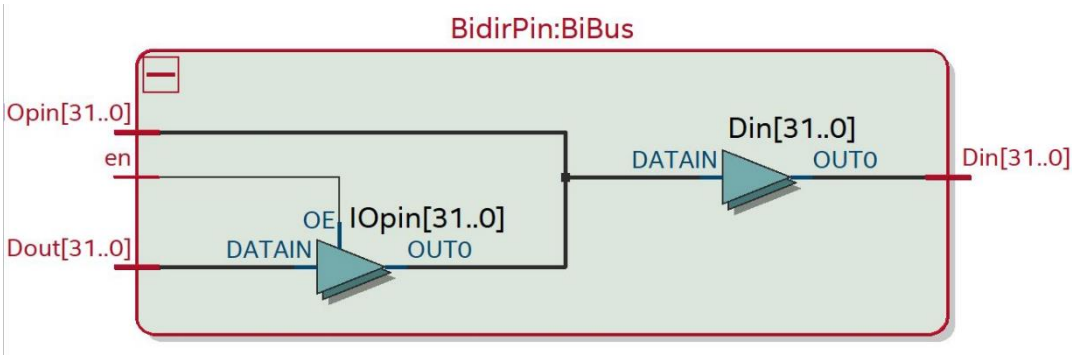

Figure 30. RTL Viewer of BidirPin

Name	Direction	Size	Functionality
Dout	IN	32	מידע היוצא מהרכיב אל ה-BUS
en	IN	1	אפשור כתיבה ל-BUS
Din	OUT	32	מידע הנכנס מה-BUS אל הרכיב
IOpin	INOUT	32	ה-BUS

Table 12. Port Table of BidirPin

כמו כן, בדקנו את ה-fmax ואת המסלול הקריטי עבור המימוש שלנו בפרויקט. ניתן לראות אותם באיורים הבאים.

	Fmax	Restricted Fmax	Clock Name	Note
1	37.93 MHz	37.93 MHz	clock	

Figure 31. Fmax

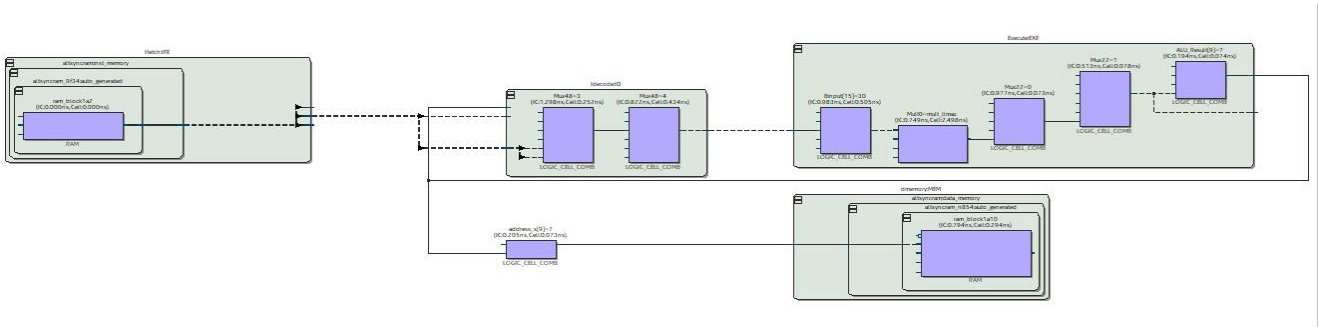

Figure 32. Critical Path

ניתן לראות כי המסלול הקריטי עובר בכל הרכיבים, כצפוי ממעבד Single Cycle. כמו כן, ניתן לראות כי על אף שהוספנו רכיבי IO, המסלול הקריטי הוא עדיין נמצא ב-CPU מכיוון שבמסלול זה מתבצעת גם קריאה מהזיכרון וגם כתיבה חזרה אל ה-RF.

בכדי לבדוק את נכונות המימוש שלנו ושאכן המעבד מבצע את כל הפקודות כנדרש וכל רכיבי החומרה, כולל רכיבי ה-IO מבצעים את תפקידם הרצנו את הקודים לבדיקה שפורסמו. חלק מהתוצאות מצורפות לדו"ח באיורים הבאים.

תחילה הרצנו את test3.asm ובדקנו שכל הערכים נכנסים כמו שצריך למקומות המיועדים להם. את התוצאות ניתן לראות באיור הבא.

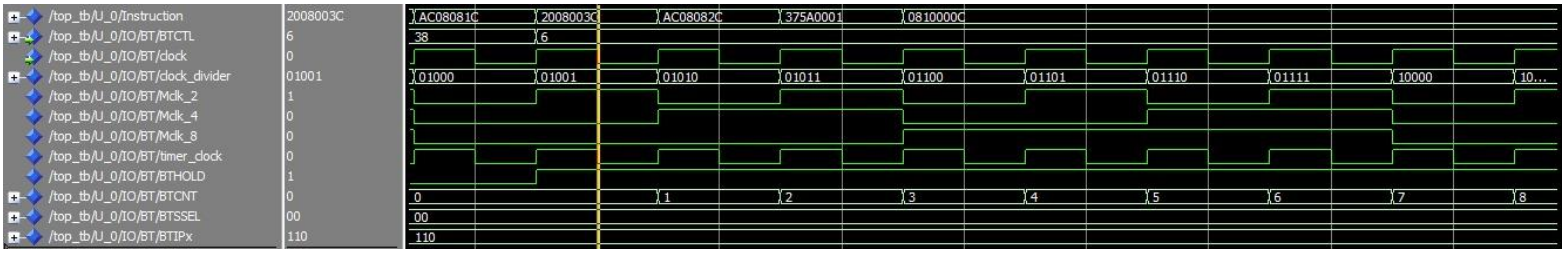

Figure 33. Modelsim simulation results

בתוכנית זו בדקנו שאכן כל הערכים שצריכים להכנס את הרגיסטרים וקווי הברקה של הטיימר אכן נכתבים כמו שצריך. ניתן לראות שאכן ה-BTCTL מקבל את הערך 0x06 כנדרש. בהתאם לכך ביטי הקנפוג משתנים בהתאם, BTSSEL = 00, BTIP = 110. כמו כן, ניתן לראות שהשעון שבאמצעותו הטיימר סופר, timer_clock הוא בדיוק כמו השעון הרגיל בהתאם לBTSSEL. עם זאת, ניתן לראות באמצעות Mclk_i שנוצרים כל השעונים הנדרשים בהתאם לחלוקה, ולכן עבור BTSSEL אחר היה נבחר שעון נכון גם כן. לבסוף, ניתן לראות שלאחר הקנפוג ואפשור הטיימר השעון אכל מתחיל לספור עפ"י הערך של BTCNT.

בנוסף, בדקנו את נכונות המימוש שלנו באמצעות הSignal Tap ב- Quartus. מכיוון שלא מצאנו קודים לבדיקה שבודקים את היווצרות אות ה-PWM מהטיימר, כתבנו קוד קצר לבדיקה של דבר זה. ניתן לראות את התוצאות באיור הבא.

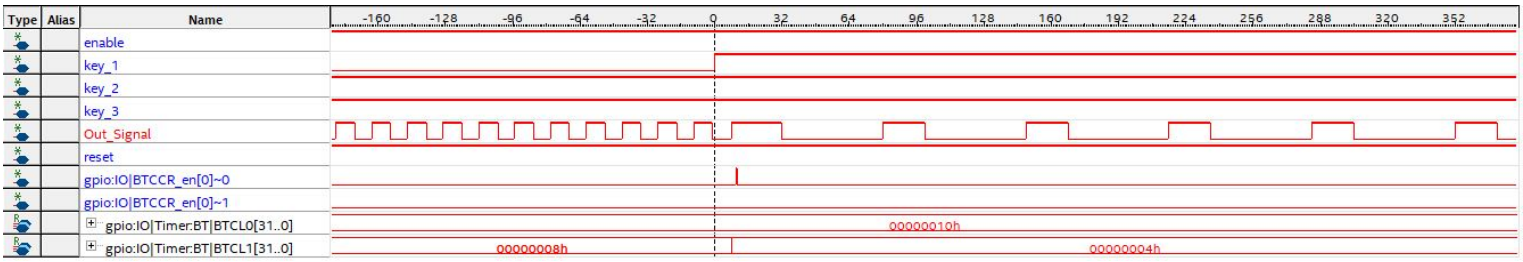

Figure 34. Signal Tap Verification

ניתן לראות שאכן נוצר אות PWM בDuty cycle מסוים בהתאם לערכים שנכנסו ל-BTCL0 ו-BTCL1. כמו כן, לאחר לחיצה על כפתור מסוים, ניתן לראות שהערך של BTCL1 קטן, ולכן הזמן בין BTCL0 ל – BTCL1 גדל, כלומר ערך האות הוא 0, וה- Duty cycle משתנה, בהתאם למה שהיינו מצפים לראות.