

final project

mips based MCU architecture and design

2024 semester B

submitted by :

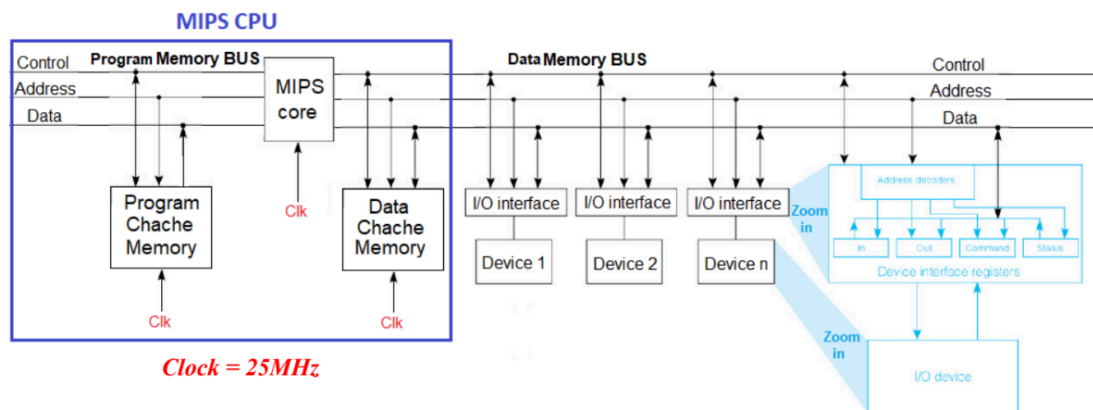
Etay kandil

Yarin Oziel

תיאור כללי של המערכת

בפרויקט זה תכננו ובנינו MCU המורכב ממעבד Single cycle MIPS מאיצי חומרה, מערכת קלט-פלט, יחידות זיכרון ובקר פסיקות. התקשורת בין רכיבי החומרה תתבצע על ידי 3 קווי BUS :

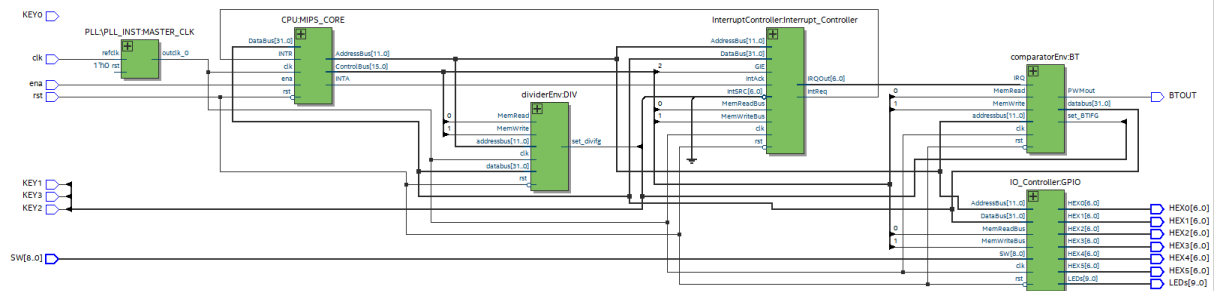
1. control bus
2. data bus
3. memory bus



כפי שניתן לראות מן התמונה הזיכרון מחולק ל 2 תתי יחידות : אחת עבור ה program memory והשניה עבור ה data memory .

כל אחת בעלת רוחב כתובת של 12 ביט עם 1024 ו4096 תאי זיכרון בהתאמה .

RTL כללי עבור כל המערכת



Logic Usage

	Resource	Usage
1	Estimate of Logic utilization (ALMs needed)	2160
2		
3	▼ Combinational ALUT usage for logic	2776
1	-- 7 input functions	64
2	-- 6 input functions	1213
3	-- 5 input functions	399
4	-- 4 input functions	344
5	-- <=3 input functions	756
4		
5	Dedicated logic registers	1780
6		
7	I/O pins	69
8	Total MLAB memory bits	0
9	Total block memory bits	163840
10		
11	Total DSP Blocks	2
12		
13	▼ Total PLLs	1
1	-- PLLs	1
14		
15	Maximum fan-out node	PLL:\...re[0]
16	Maximum fan-out	1598
17	Total fan-out	21469
18	Average fan-out	4.50

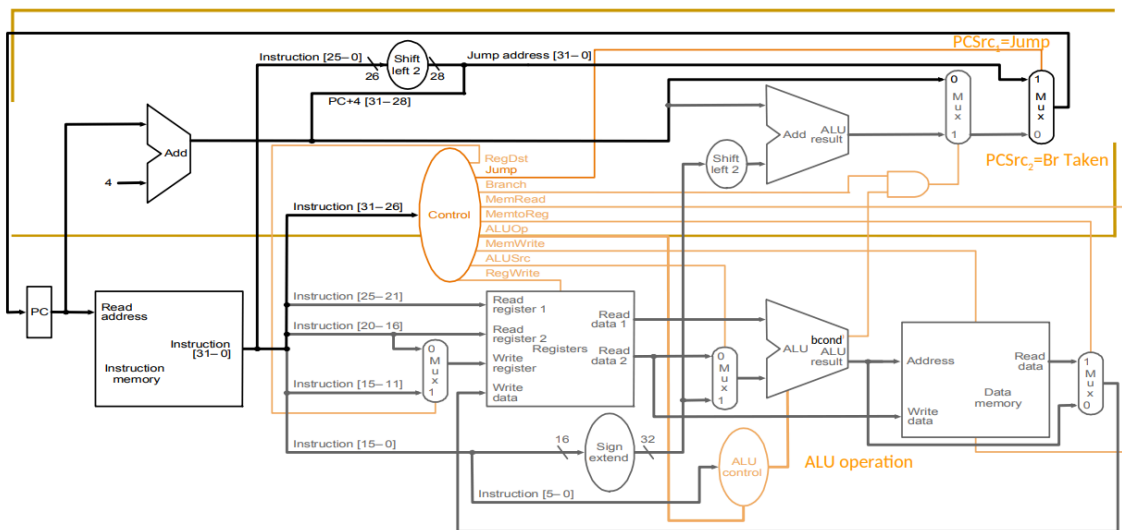
single cycle mips

ה (risc based) single cycle mips הוא יחידת העיבוד המרכזית בבקר.

המעבד מבוסס על ארכיטקטורת וון ניומן, בדומה למה שהוצג לנו בקורס התיאורטי- כלומר מכיל יחידת alu, datapath, יחידת בקרה, זיכרון פנימי (GPR)

ויכולת התקשרות עם הרכיבים הפריפריאליים השונים. המעבד מסוגל לתמוך בהוראות שונות כגון ביצוע פעולות אריתמטיות-לוגיות , טעינה/שמירה של ערכים אל הרגיסטרים מן הזיכרון ולהפך והסתעפות על עבר כתובות תוכנית שונות כתלות בקיום תנאים שונים. המעבדה מתחיל ומסיים הוראה במחזור שרון אחד .

מבנה המעבד :



מבנה ההוראות :

BASIC INSTRUCTION FORMATS

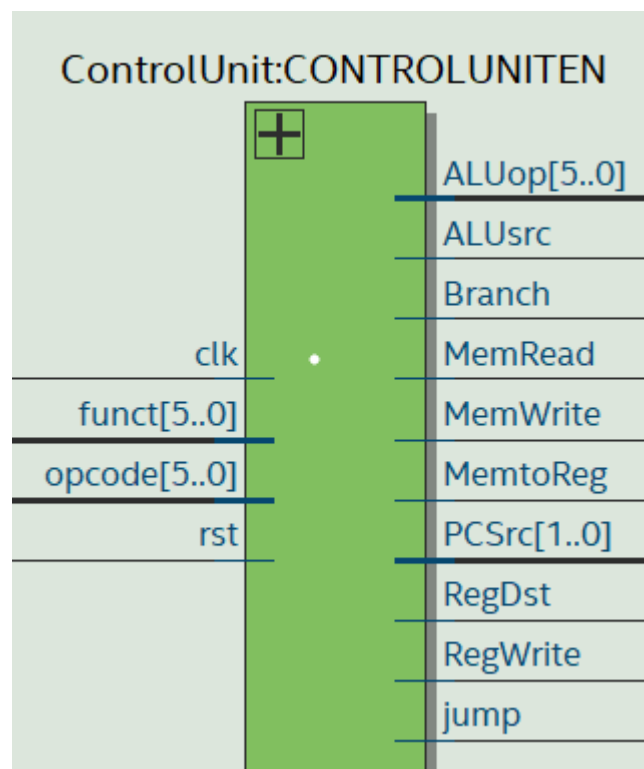
R	opcode	rs	rt	rd	shamt	funct
	31	26 25	21 20	16 15	11 10	6 5 0
I	opcode	rs	rt	immediate		
	31	26 25	21 20	16 15	0	
J	opcode	address				
	31	26 25	0			

כפי שניתן לראות ההוראה "מפורשת" בצורה שונה עבור כל אחד מן סוגי הפעולות כתלות ב 6 הביטים האחרונים של ההוראה - opcode ולכן המידע המתקבל מן הprogram memory מקבל משמעות שונה .

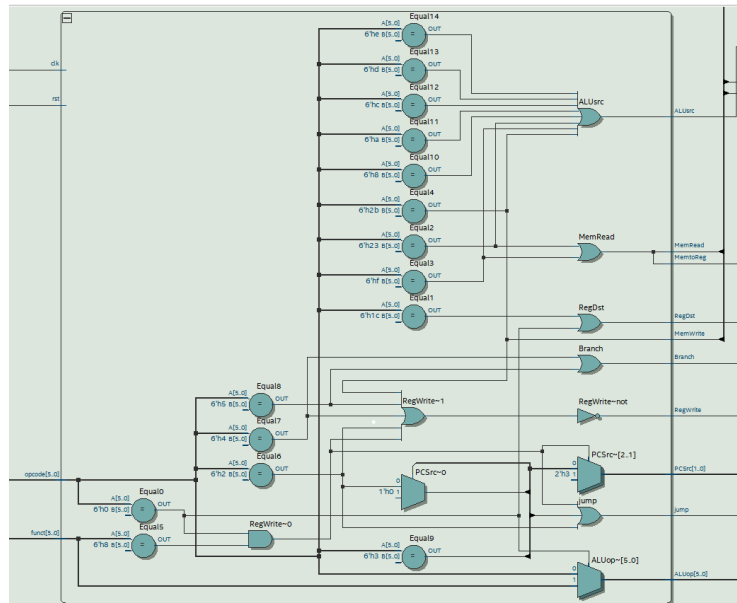
control unit

היחידה האחראית על תכלול פעולות תתי המודולים השונים במעבד היא יחידת הבקרה . היחידה מוזנת על ידי 5 הביטים הראשונים (במידה וששת הביטים האחרונים הם 0 - כלומר פעולת R TYPE) וששת הביטים האחרונים אחרת . תפקיד היחידה הוא להזין את ה datapath בסיגנלים המתאימים כתלות בסוג ההוראה כדי שתבוצע כראוי

מודל גרפי:



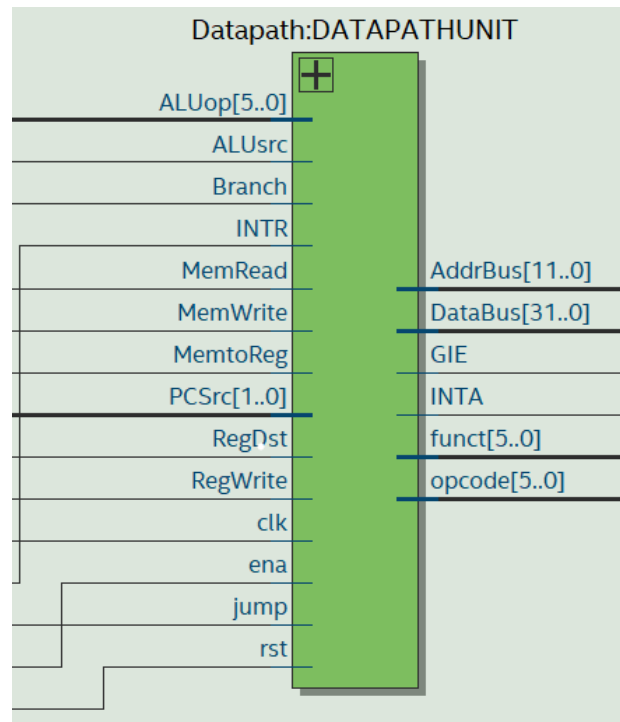
rtl view:



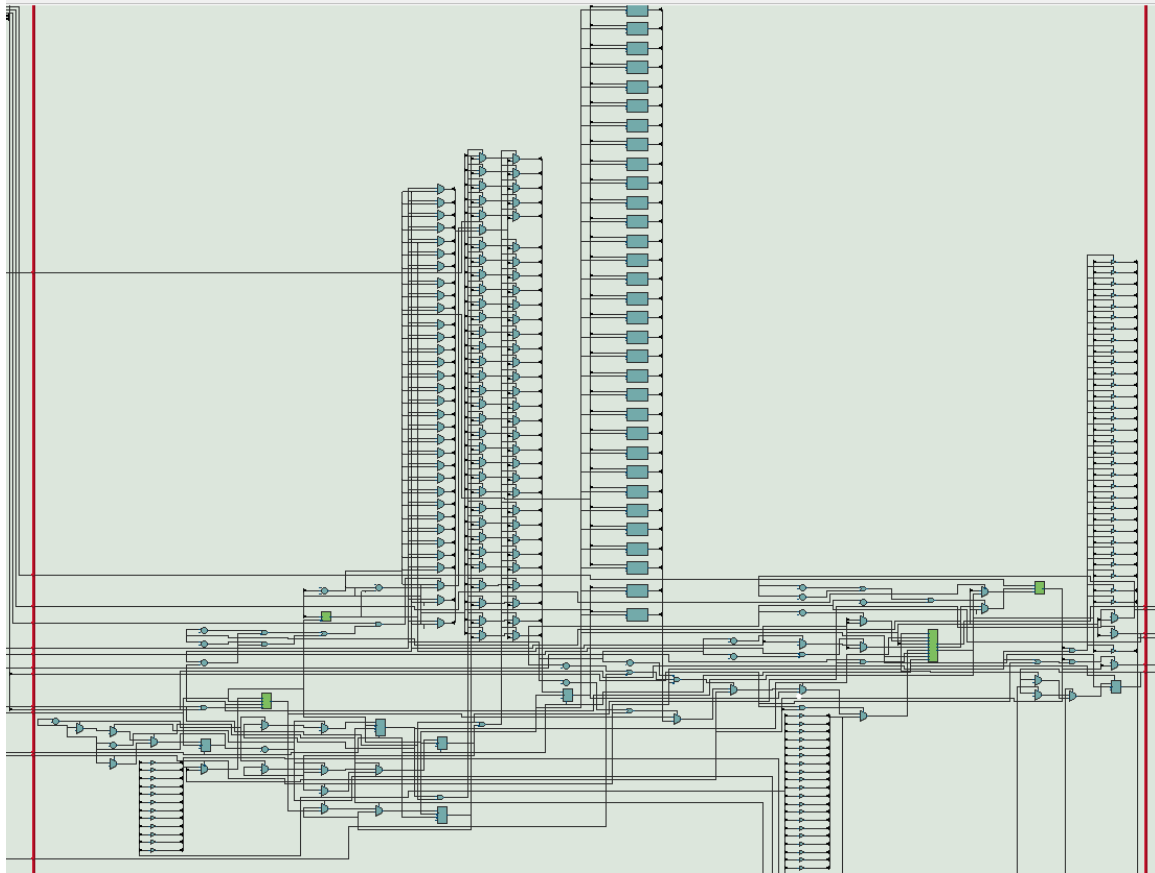
data path

היחידה האחראית על הקישור בין תתי המודולים השונים במעבד היא ה datapath . היחידה דואגת להעברת וקבלת סיגנלים שונים בין alu ה gpr ה control unit ה data memory ו program memory .

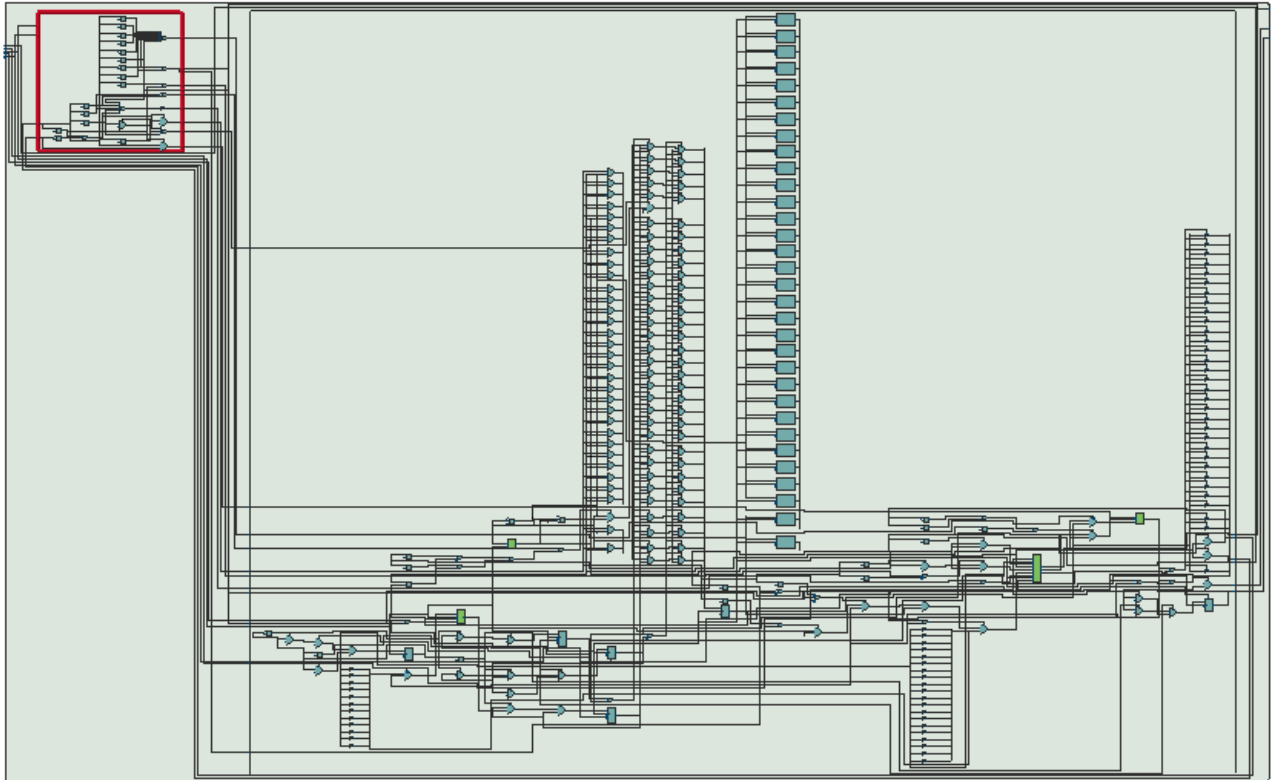
מודל גרפי:



rtl view :



CPU full RTL:



hardware accelerators

multi cycle divider

המחלק הוא מודול פריפריאלי הפועל במקביל למעבד ואינו תלוי בו.
המחלק פועל לפי אלגוריתם חילוק ארוך - משך הפעולה הוא 32 מחזורי שעון (כאורך ה datawidth).
קלט המחלק הוא 3 סיגנלים :

rst : מחזיר את המחלק למצב ההתחלתי
ena : מורה על תחילת פעולת האלגוריתם - עולה ל 1 כאשר ערך חדש נטען במחלק
מחלק : ערך המחלק , נשמר במיקום ייעודי בזיכרון
מחולק : ערך המחולק , נשמר במיקום ייעודי בזיכרון

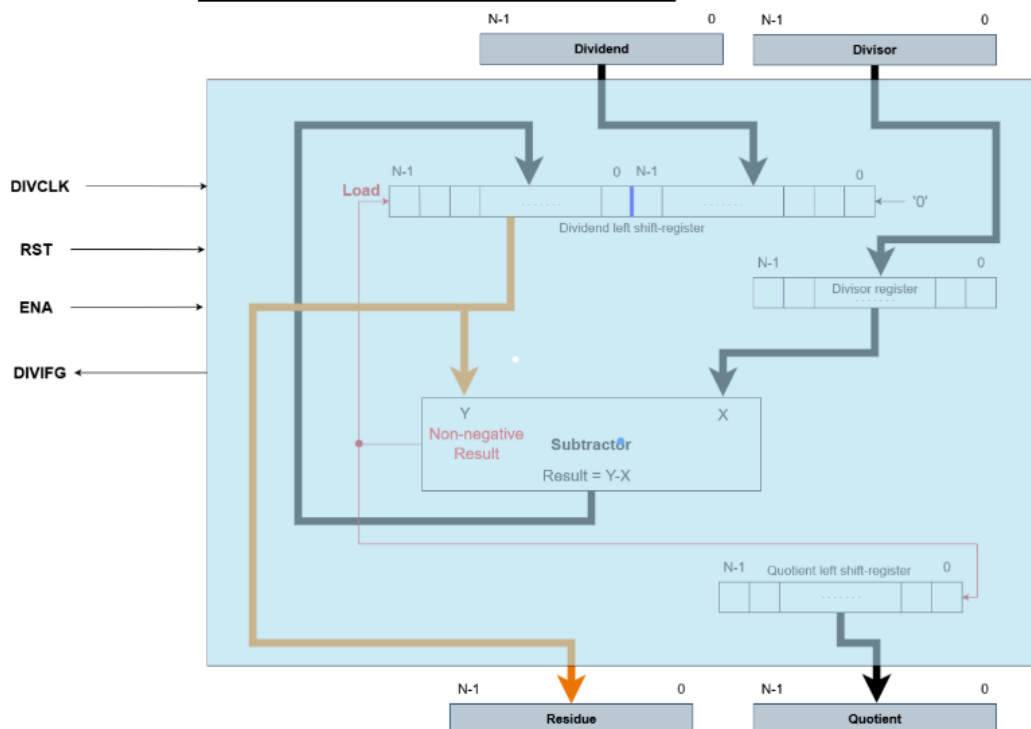
פלט המחלק הוא 3 סיגנלים , המתקבלים לאחר 32 מחזורי שעון מתחילת פעולת החילוק :

מנה : מנת פעולת החילוק , נשמרת במיקום ייעודי בזיכרון
שארית : שארית פעולה החילוק , נשמרת במיקום ייעודי בזיכרון
דגל בקשה לפסיקה : ביט בודד המוזן אל תוך בקר הפסיקות , תפקידו להודיע על סיום פעולת החילוק ושליחת בקשה לכניסה ל

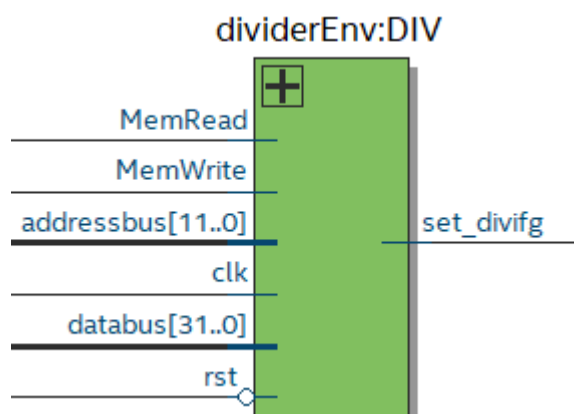
div_isr

מבנה המחלק :

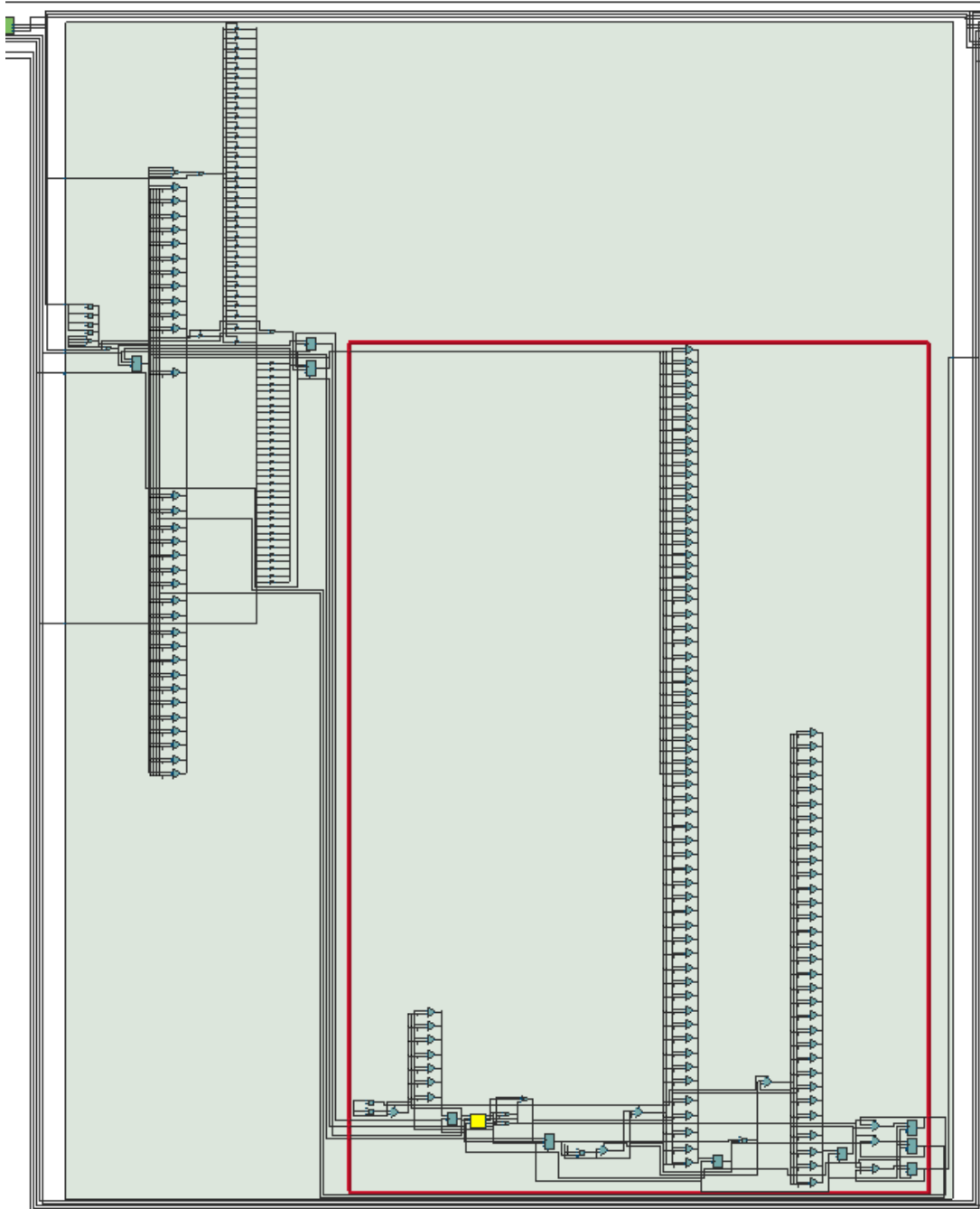
iv. Unsigned Binary Division Multicycle Accelerator:



מודל גרפי :



rtl view :



basic timer

בדומה למחלק זהו רכיב פריפריאלי הפועל במקביל למעבד ואינו תלוי בו.
לרכיב זה 2 יכולות בסיסיות :

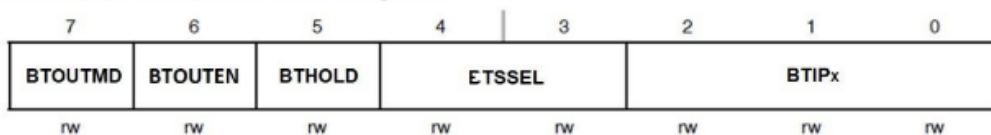
1. מונה המסוגל לפלוט דגל בקשה לפסיקה כתלות בערכו
2. פליטת אות pwm

קלט המונה הוא 4 סיגנלים :

rst : מאפס את ערך המונה

btctl : מכיל את אותות הבקרה של המונה

BTCTL, Basic Timer Control Register



- 3 הביטים הראשונים אחראים על בחירת התנאי להעלאת הבקשה לפסיקת bt_isr
- 2 הביטים הבאים אחראים לבחירת השעון המזין את הרכיב
- הביט השישי אחראי על אפשרור הגדלת ערך המונה ב1 בכל מחזור שעון
- הביט השביעי אחראי על אפשרור שינוי הערך במcpw
- הביט השמיני אחראי על מצב המcpw : טוגל מ0 ל1 או מ1 ל0 כשהתנאי לטוגל מתקיים.
- נשמר במיקום יעודי בזיכרון.

: btccrx

btccro הוא הערך המקסימלי של המונה, מתאפס לאחריו .
ערך זה מכתיב את מחזור השעון של האות.

btccr1 הוא ערך המונה בו מתבצע הטוגל .
ערך זה מכתיב את duty cycle של האות.
שניהם נשמרים במיקום ייעודי בזיכרון .

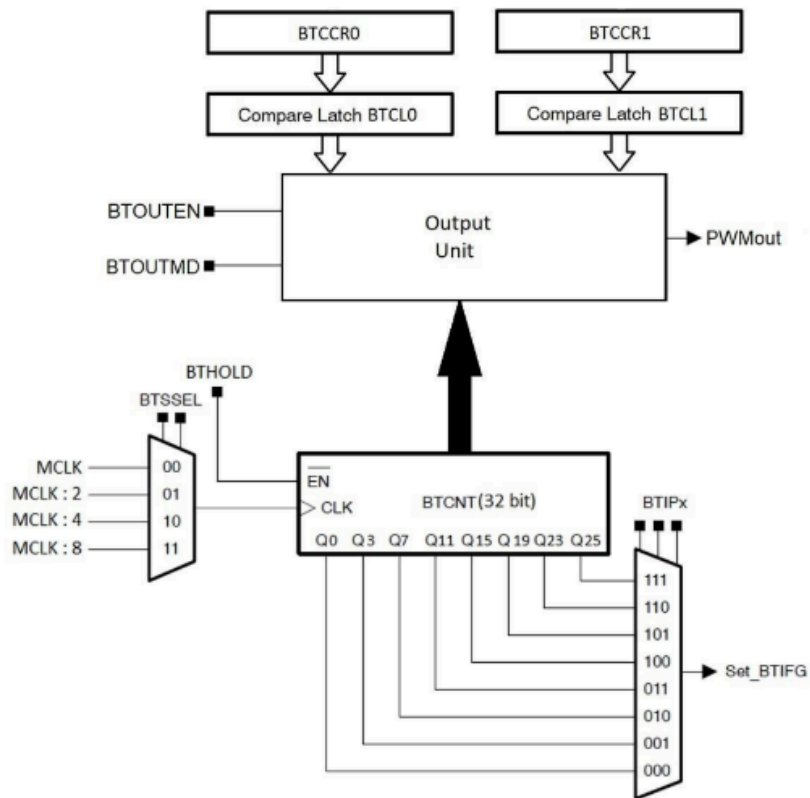
פלט המונה הוא 3 סיגנלים :

btcnt : ערך המונה

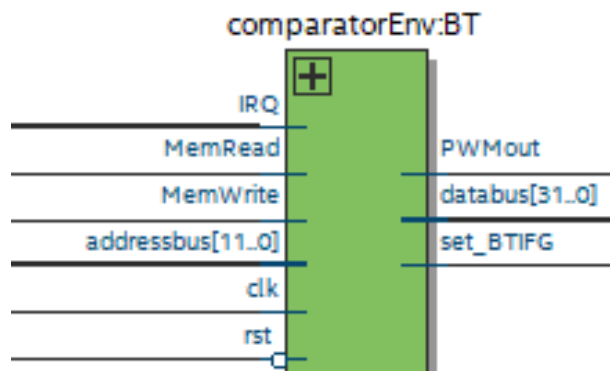
pwm : pwm אות

set_btifg : דגל בקשה לפסיקת טיימר, משתנה כתלות בערך המונה ואותות הבקרה

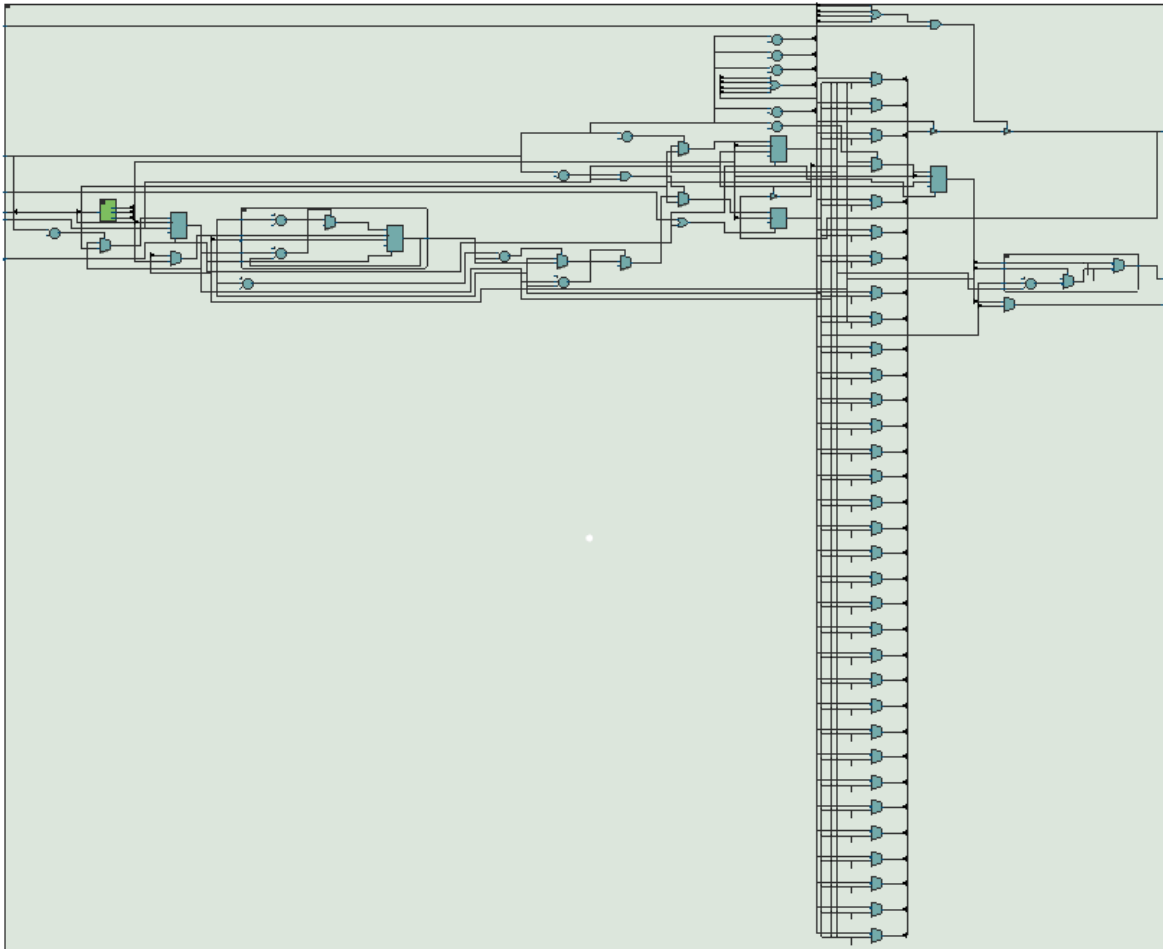
iii. Basic Timer with output compare capability:



מודל גרפי :



rtl view :



interrupt controller

כדי להוסיף אל ה-MCU תמיכה בפסיקות תכננו בקר פסיקות . בקר הפסיקות מוזן מ-5 מקורות שונים : שלושת הכפתורים (מערכת קלט-פלט) ו-2 מאיצי חומרה . כל אחד מן המקורות האלה מסוגל לייצא בקשה לפסיקה (כתלות בהגדרת המשימה שלו , עליה הסברנו בתיאור כל אחד מן המודלים) , תפקיד הבקר הוא לקבל את הבקשה , לבדוק האם היא מאופשרת , לתעדף בין כל בקשות הפסיקה המאופשרות ולבסוף לשלוח אל ה-CPU בקשת פסיקה בצירוף כתובת רוטינת הפסיקה בזיכרון . לאחר קבלת אות מן המעבד כי הבקשה התקבלה , בקר הפסיקות יאפס את הדרישה לפסיקה מן הרכיב שבקשתו התקבלה .

פרוטוקול הכניסה לפסיקה הוא : עצירת עדכון ה-CS , שמירת ערכו ברגיסטר 27 הורדת דגל אפשר הפסיקה הגלובלי ל 0 ולבסוף הכנסת כתובת רוטינת הפסיקה הרלוונטית אל ה-CS . בסיום הפסיקה ה-CS יתעדכן עם הערך שנשמר ברגיסטר 27 ודגל איפשר הפסיקה הגלובלי יעלה ל 1 .

לבקר הפסיקות 3 סיגנלי כניסה :

INTSRC : סיגנל המכיל את כל מקורות הפסיקה שהתקבלו במחזור השעון

IE : מכיל את דגלי האיפשר של כל הפסיקות , נשמר במיקום ייעודי בזיכרון

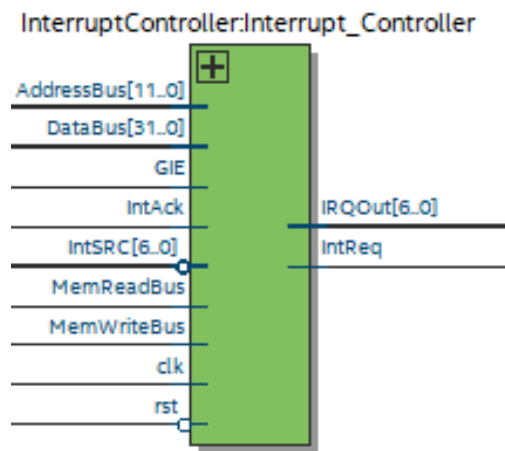
INTA : סיגנל המגיע מן המעבד שתפקידו להודיע על קבלת הדרישה לפסיקה

לבקר הפסיקות 2 סיגנלי מוצא :

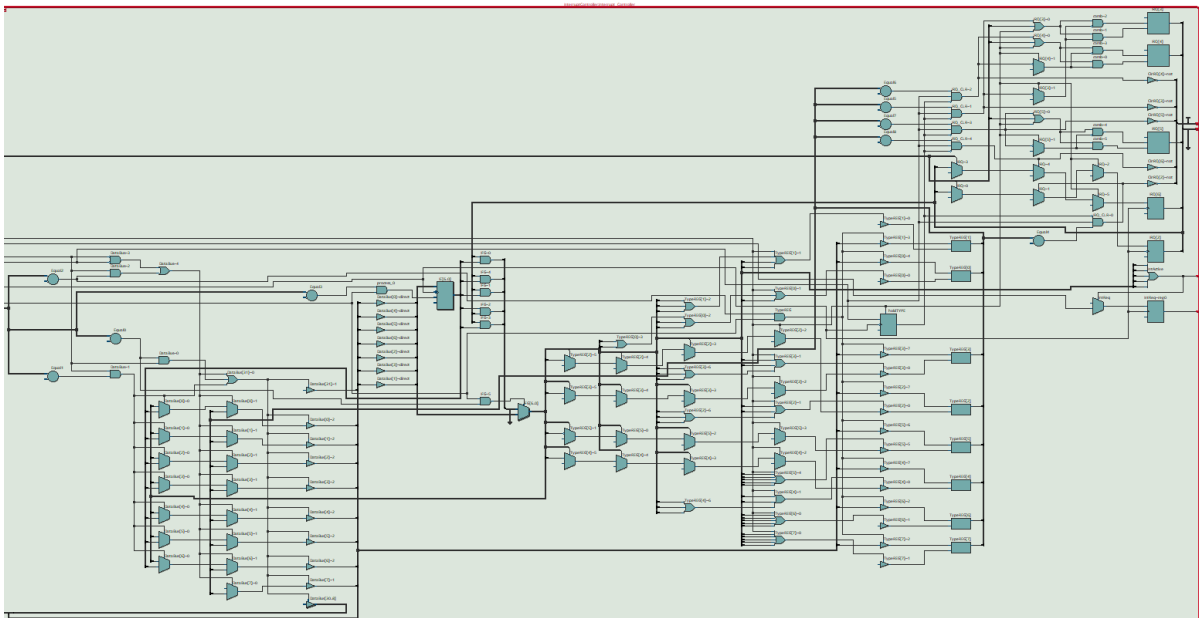
INTR : סיגנל המגיע על המעבד - תפקידו להודיע על דרישה לפסיקה

TYPE REG : מכיל את הכתובת בזיכרון של רוטינת הפסיקה בעלת העדיפות הגבוהה ביותר , נשמר במקום ייעודי בזיכרון

מודל גרפי :



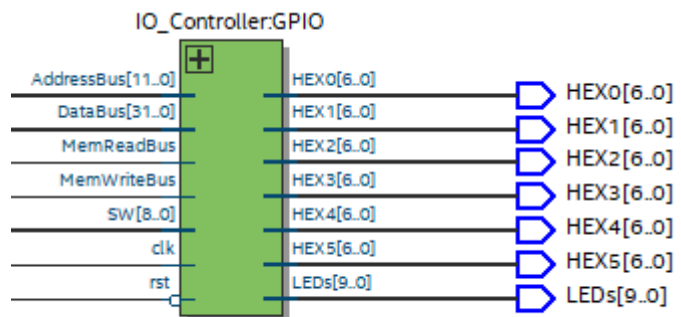
rtl view :



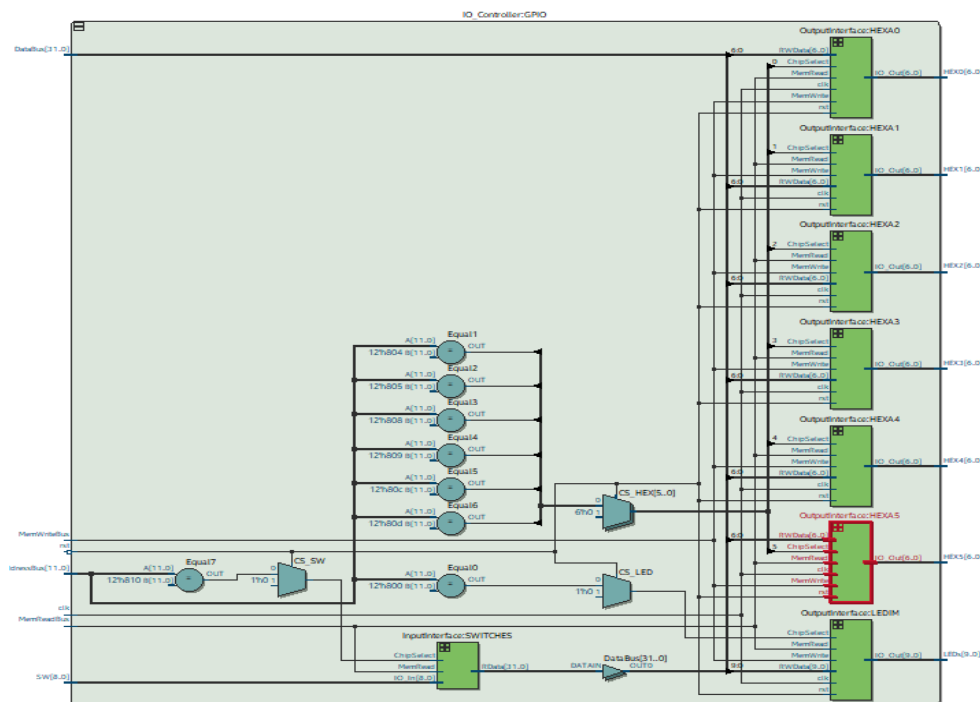
GPIO CONTROLLER

אל ה MCU הוספנו תמיכה במערכת קלט- פלט .
ערכים אלה נשמרים בזיכרון והכתיבה והקריאה מהם נעשית דרך databus .
קליטת ערכים נעשית על ידי הסוויצים והלחצנים , הפלט מוצג על גבי הלדים
וההקסות. ניתן לנצל את יכולות המערכת כדי לקלוט ערכים בהם נשתמש לחישובים
כלשהם הנעשים במעבד ולאחר מכן הצגתם, או קליטת בקשות לפסיקות .

מודל גרפי :



rtl view :



תדר מקסימלי :

Slow 1100mV 85C Model Fmax Summary

 <<Filter>>

	Fmax	Restricted Fmax	Clock Name
1	24.8 MHz	24.8 MHz	\PLL_INST:MASTER_CLK pll_inst al...].gpll~PLL_OUTPUT_COUN
2	89.12 MHz	89.12 MHz	altera_reserved_tck

Slow 1100mV 0C Model Fmax Summary

 <<Filter>>

	Fmax	Restricted Fmax	Clock Name	Note
1	25.44 MHz	25.44 MHz	\PLL_INST...ER divclk	
2	91.68 MHz	91.68 MHz	altera_reserved_tck	

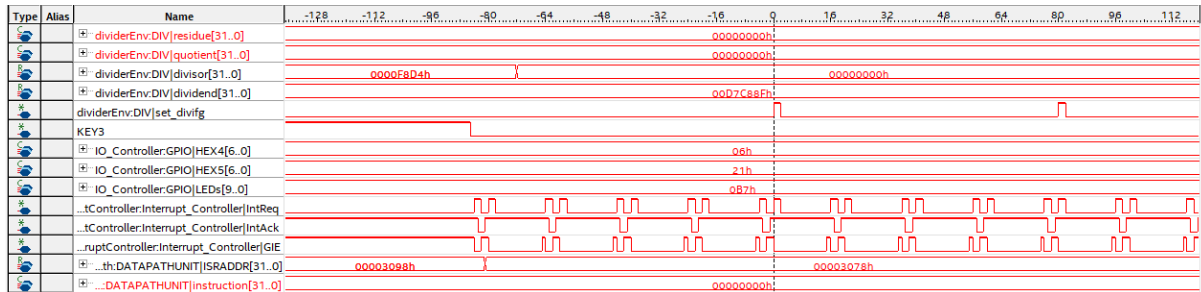
- כאן ניתן לראות כי המעבד קופץ לפונקציה אשר כותבת ערכים שונים לאמצעי פלט (לדים והקסות) כפי המוגדר בפונקציה.

[illegible]

- ניתן לראות בלחיצה תוכנית על KEY3 רגיסטר TYPE של יחידת הפסיקות טוענת את הכתובת של ISR המתאים. לאחר ה ISR אשר טוען למחלק ומחולק ערכים, יחידת החלוקה מתחילה לעבוד עד אשר מסיימת עבודה ומוציאה בקשה לפסיקה. כפי שניתן לראות בקר הפסיקות טוען את הכתובת ISR של המחלק אשר טועמת את המנה והשארית ללדים ולהקסות.

Signal-Tap analysis in Quartus

ביצענו בדיקת SIGNAL TAP על תוכנית פסיקה אשר מגיבה ללחיצת KEY3 ומפעילה פסיקה לאכלוס מחלק ומחולק כך שפסיקת מחלק תקרא:



- ניתן לראות בלחיצת KEY3 כי קיים BOUNCING אך לאחר מכן הפסיקה מתייצבת ועוברת בקריאה ל PC.