

# דו"ח מסכם לפרויקט גמר קורס

"מעבדת ארכיטקטורת מעבדים מתקדמת ומאיצי חומרה"

הפקולטה למדעי ההנדסה  
אחראי מעבדה: רבוא חנניה

מגייסים:

בנימין אומנסקי - ת"ז 322688946  
יעדו רון - ת"ז 322384330

## הסבר על המערכת

בפרויקט זה, התבקשנו למשם מעבד עם ליבת MIPS, התומך במיפוי זכרון O/I ובקבלת פסיקות חיצונית וניהול רוטינות פסיקה. Pipelined MIPS המשלב ליבת S Single Cycle ליבת ה- Pipeline ממומשת ע"י CPU במקום ליבת. ארכיטקטורת MIPS בעלת 5 שלבים התומכת בזיכרון וטיפול בעיות של Data dependencies תוך שימוש ביחידות - Forwarding Unit Hazard Detection.

את המערכת הכוללת שבניינו צרבנו ובדקנו בשכב ה- De2-115. בoard FPGA Altera המשלב רכיבי פריפריה שונים כגון כפتورים, לדימ, מסכי *segment* 7 ועוד.

# MCU

קובץ ה-MCU משמש כ"מעטפת" (Wrapper) אשר מכילה ארבעה מודולים עיקריים:

## Pipelined MIPS

## GPIO

## Interrupt Controller

## Basic Timer

## Fir Filter

בהתאם לאיר המצורף:

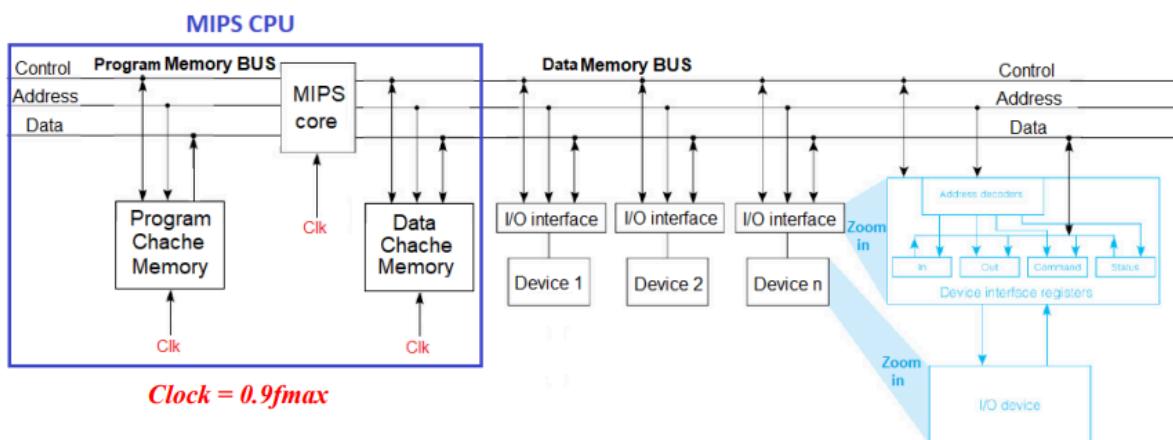
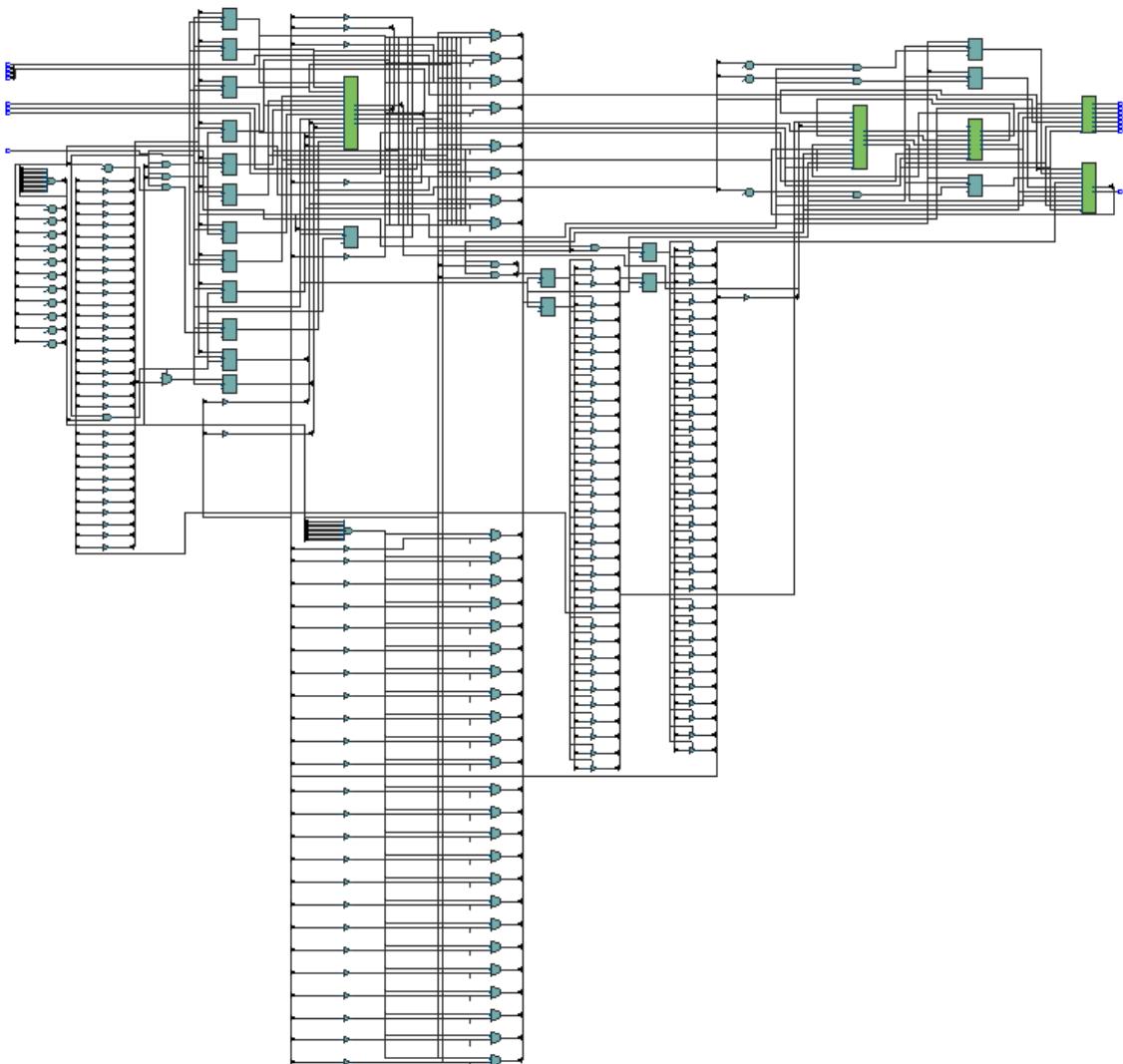


Figure 1 : MCU System architecture

קובץ זה מכיל בנוסף לשוטף הכלל 3 תת-באים: מידע כתובות ובקרה. הבאים המשוטף הינו אמצעי התקשרות שבין ליבת הבקר לרכיבים הפריפריאליים.

להלן תרשימים ה-RTL:

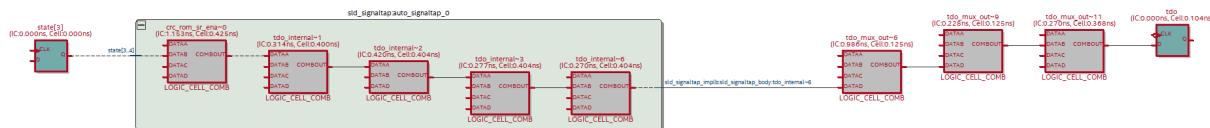


## תדר מקסימלי (Fmax)

Slow 1200mV 85C Model				
	Fmax	Restricted Fmax	Clock Name	Note
1	73.44 MHz	73.44 MHz	altera_reserved_tck	

קיבלונו תדר שעון מקסימלי 73.44 MHZ

## Critical Path



# Pipelined MIPS

קובץ ה-SIM משמש מעתפת למודולים:

IFETCH

IDECODE

EXECUTE

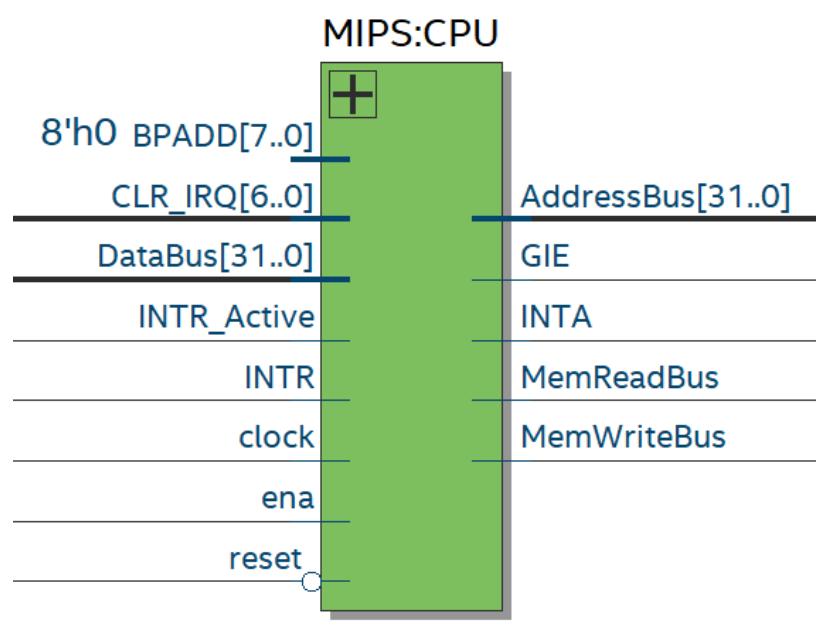
DMEMORY

CONTROL

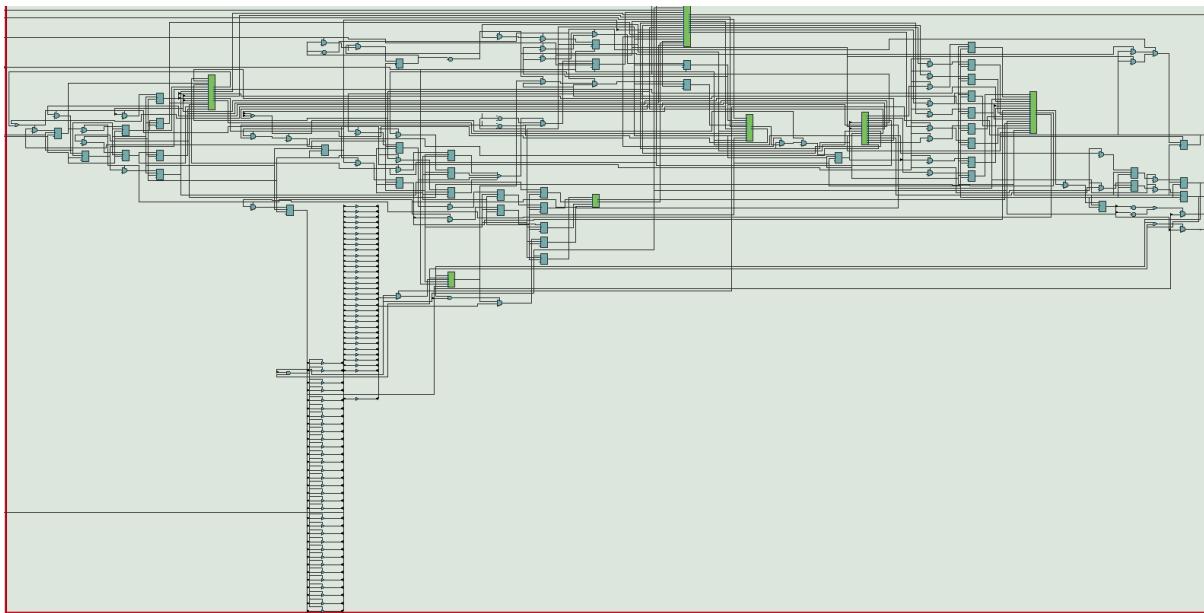
Forwarding

HazardDetection

הכניסות: Reset, Clock, Ena  
היציאות: סיגנלים רבים המשמשים לניתוח וריפוי קציה  
.Quartus או SignalTap ב-Modelsim ב-



## תרשים ה- RTL



## וריפיקציה פונקציונלית

- תוכנית MIPS להרצת חיבור מטריצות (תרגום מקוד C לאסמבלי).
- סימולציות ב- Modelsim (Wave Diagram, Memory List).
- סימולציות על ה-FPGA לוריפיקציה בזמן אמיתי.

## **GPIO**

מודול לתמיכת בקילט/פלט חומרתי באמצעות אמצעות O/I.

- **פלטים:** לדים, תצוגות Hex.
- **קלטים:** מוגדים SW0-SW7 (כולל SW8 לקביעת O/I).  
\_SW9 לאישור Breakpoint.
- כתובות במרחב הכתובות מ-0x800 ואילך.

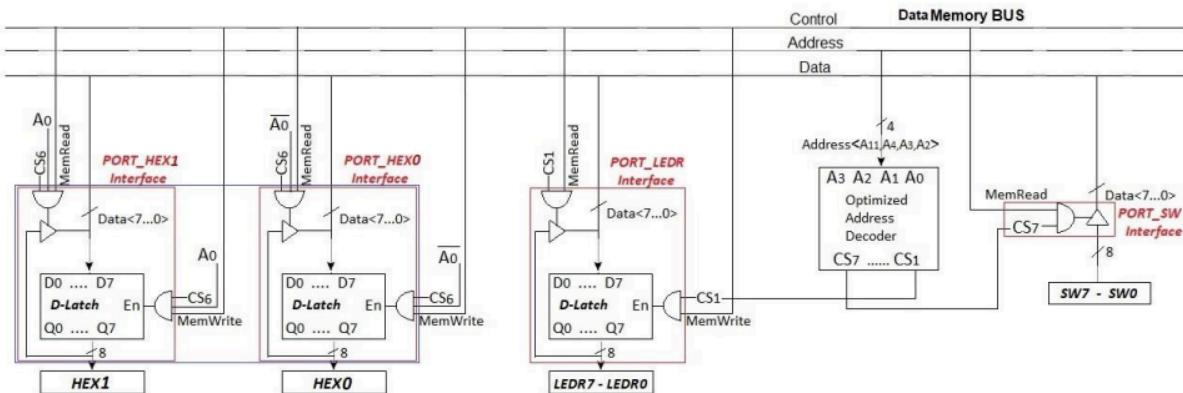
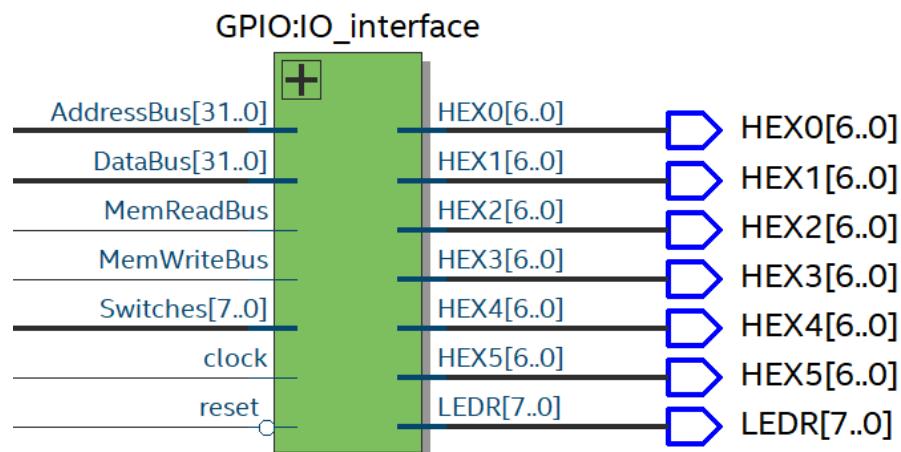


Figure 3: Primitive GPIO peripheral connection using Memory Mapped I/O approach

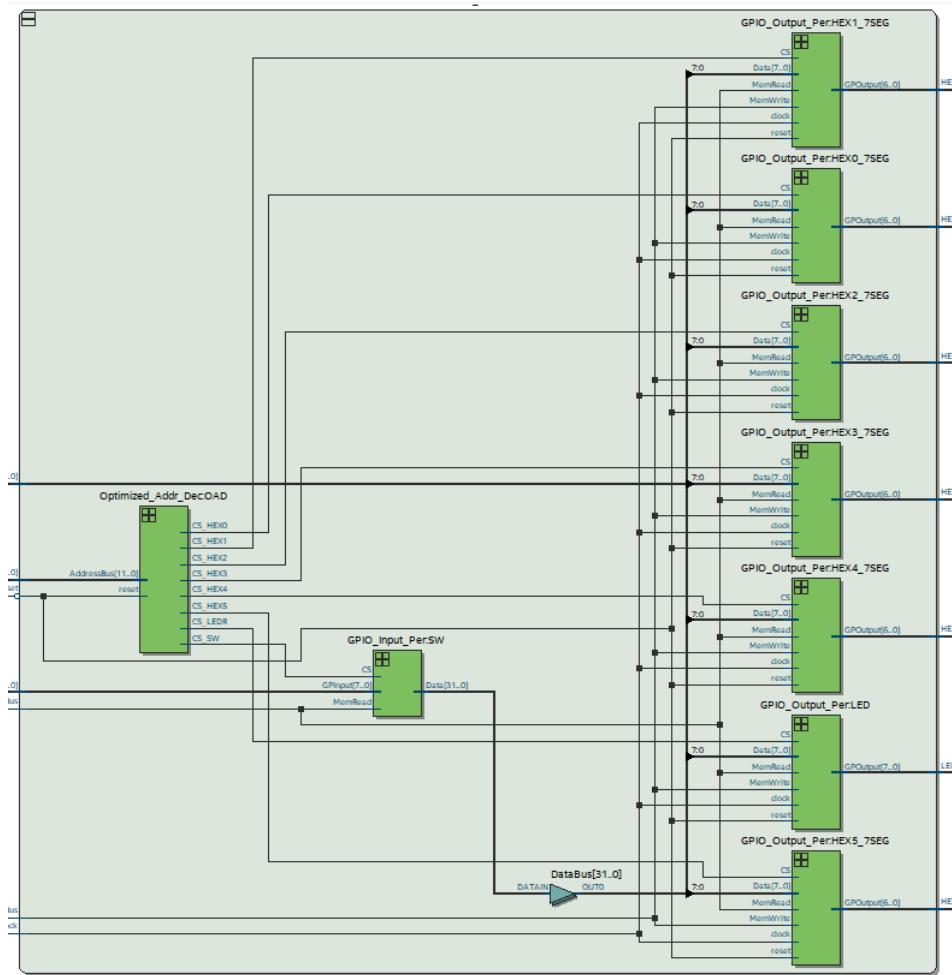
הרכיבים המוגדרים כפל"ט של המערכת הם ה לדים ותצוגות segment Hex-7 קלט למערכת (שינוי במסגרת פסיקות חיצונית) הינו מערך מתגים SW7 – SW0 שניתנים לקריאה, ואחריהם בין היתר (בפעולת Breakpoint למערכת, שאליה ירוז התכנית ותעזור (בקבלת סיגנל טריגר ST\_Trigger).  
במודול ה-MCU הגדרנו את מתג SW9 כאחראי על הפעלת קו הבקרה enable שמאפשר את ריצת התכנית.

תקשרות בין מודול ה-MIPS לפריפריות נעשית על גבי הבוא המשותף. בשלב ה-EX נעשית כתיבת הכתובת השנייה לכתובת הרצiosa של ה-GPIO, בשלב ה-MEM מתבצע שימוש בזכרון כתוצאה מהערך שעלה הבוא יהיה הערך הנכתב.

בפקודת כתיבה לדוגמה ללד 0-Hex או ל-LED, כמו `sw $t0, 40x0`, מודול ה-GPIO יהיה זה שיפענח את הכתובת וידע כי יש לבצע Write לערך המידע, בזכות בחירת Chip Select



תרשים ה-RTL:



**Test0** – כתיבה בלולאה לערכיהם עולים על כל הרכיבים.

**Test1** – בדיקת מצב מתגים, מונה עולה/ירד.

**Test2** – בדיקת מתגים להכפלה/חלוקת.

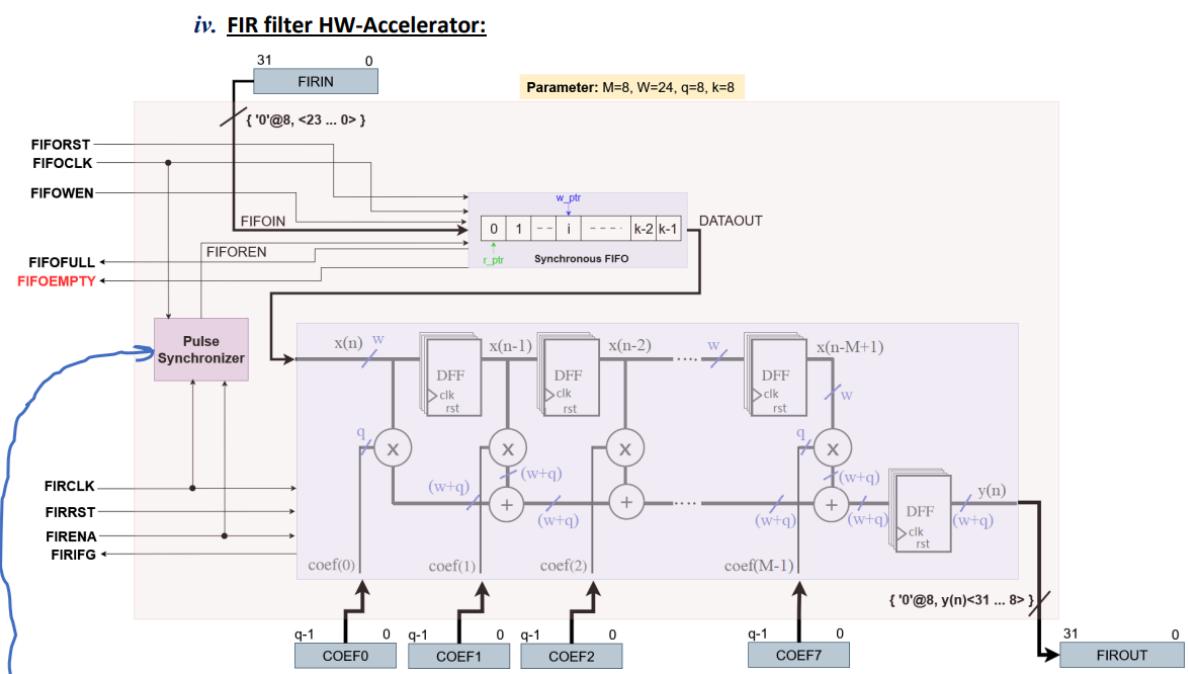
**Test3** – בדיקת מתגים לחיבור/חיסור/הכפלה.

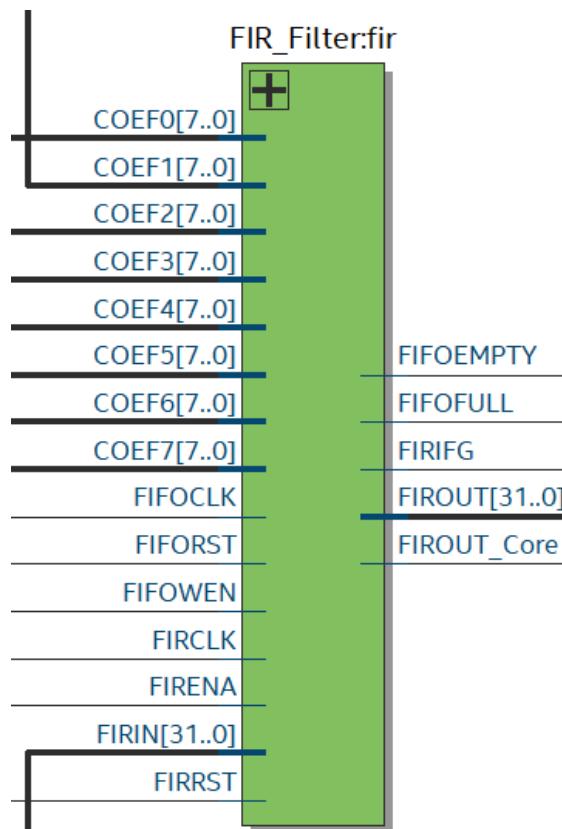
## FIR FILTER

הפילטר כולל 3 רכיבים:

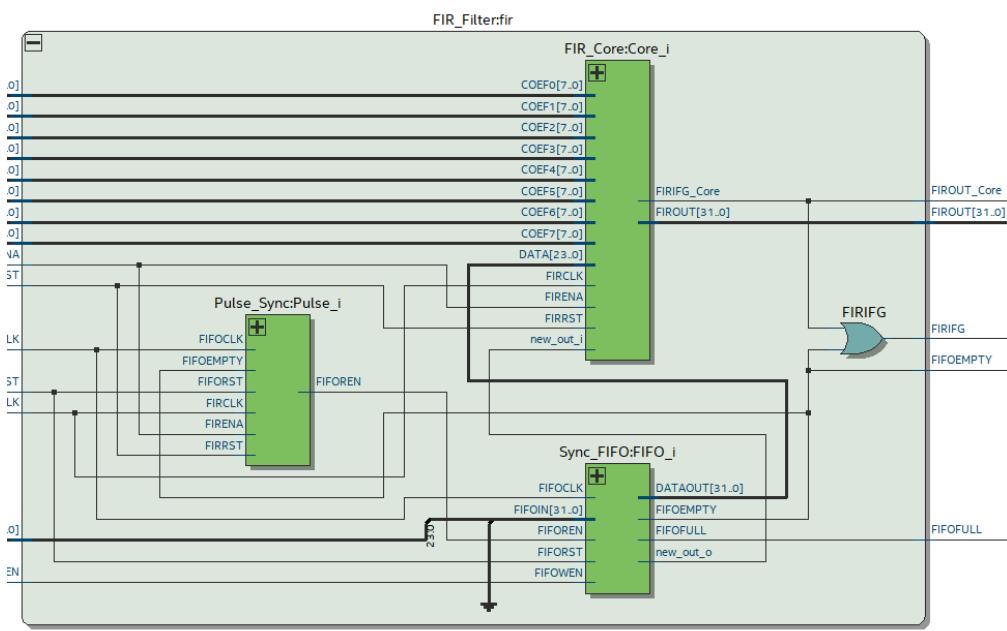
## Pulse Synchronizer Synchronous FIFO FIR Core

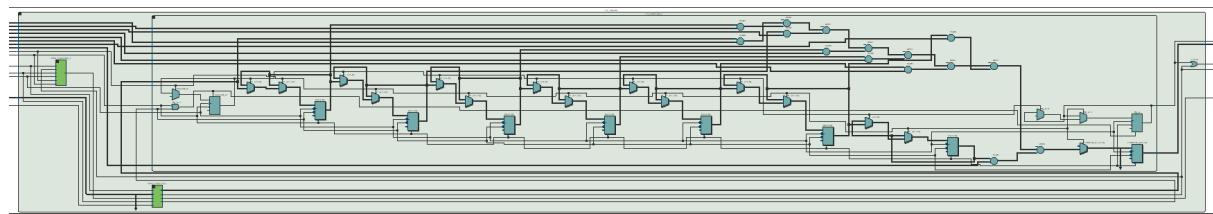
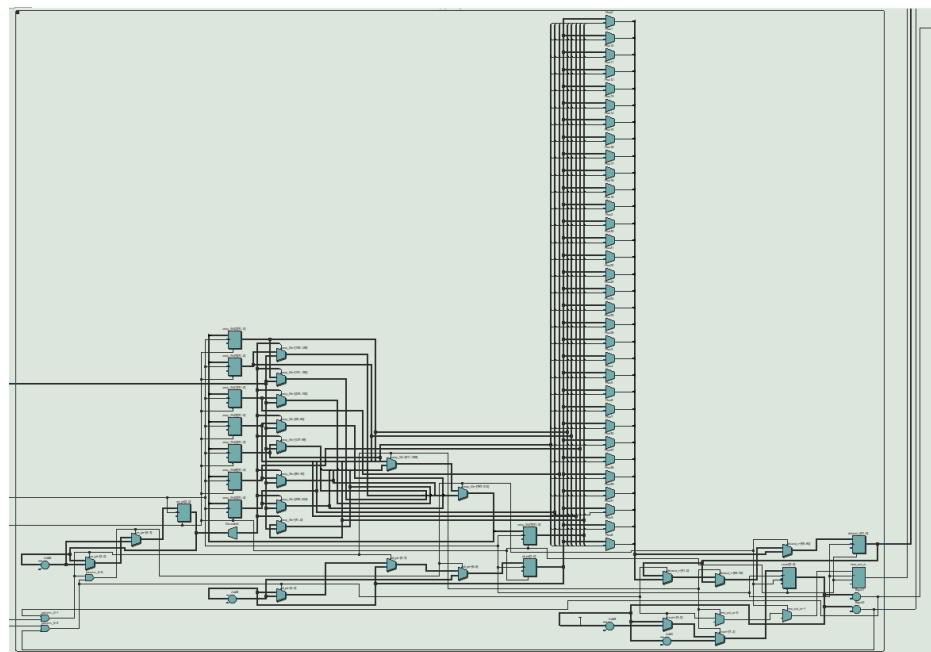
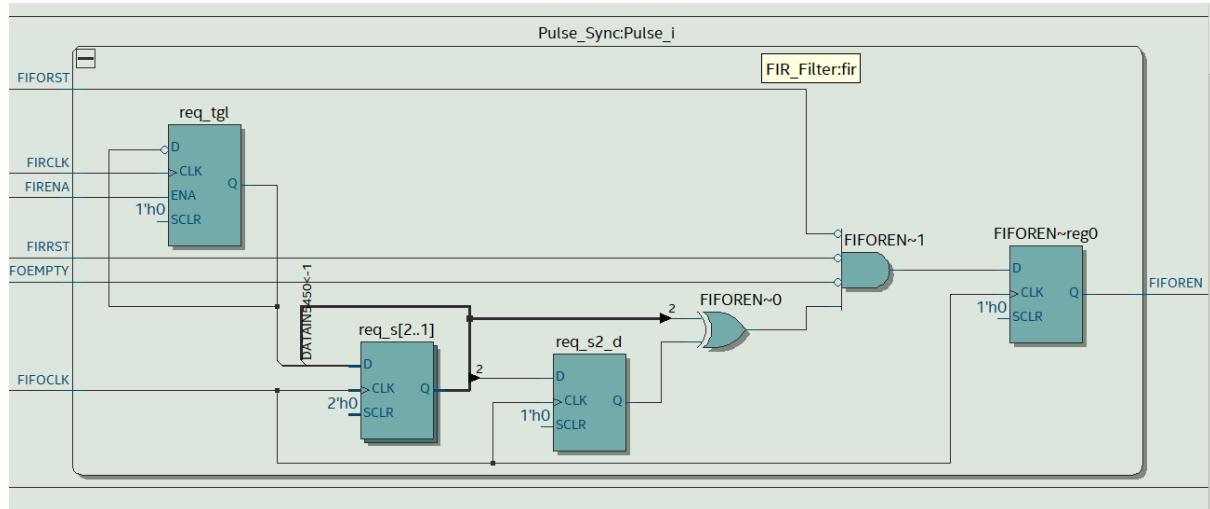
זהו מסנן ספרתי שմבצע **(Finite Impulse Response)** חישוב ליניארי על רצף דגימות קלט. הוא מבוסס על סכימה משוקללת של דגימות עבר, כאשר המשקלות קבועות על ידי מקדמים





:RTL תרשימים ה-





# Interrupt Controller

מודול לניהול פסיקות חיצונית.

מקורות הפסיקה:

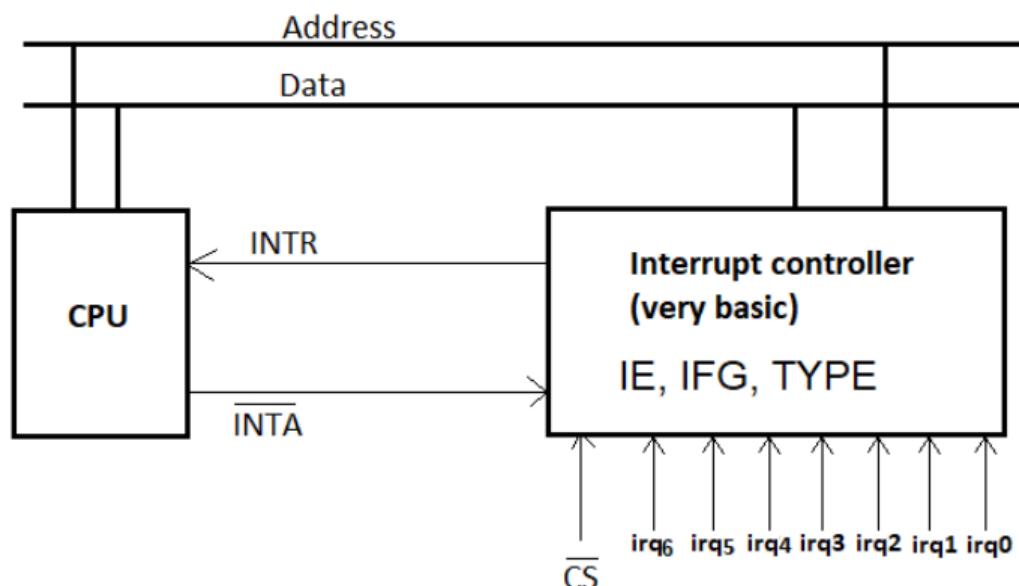
(KEY0) Reset

(KEY1–KEY3) Interrupts

Basic Timer

Fir Filter

## Interrupt controller:



מקורות הפסיכה:

- לחץ **KEY0** שאחראי על ביצוע Reset לתוכנית (קפיצה לכתובת תחילת התוכנית).
- לחצנים **KEY3 – KEY1** שאחראים על ייצור פסיקות והפעלת רוטינות פסיקה ייעודיות המוגדרות בראש בתוכנית.
- **Basic Timer** – מודול נוסף שיוצר פסיקות אשר אחראי על ייצור אוטות מחזוריים המשמשים כמקור פסיקה נוסף וכן כאות MW<sub>C</sub> חיצוני.

תקשרות בין מודול ה-MIPS לפריפריות נעשית על גבי הבאים המשותף.

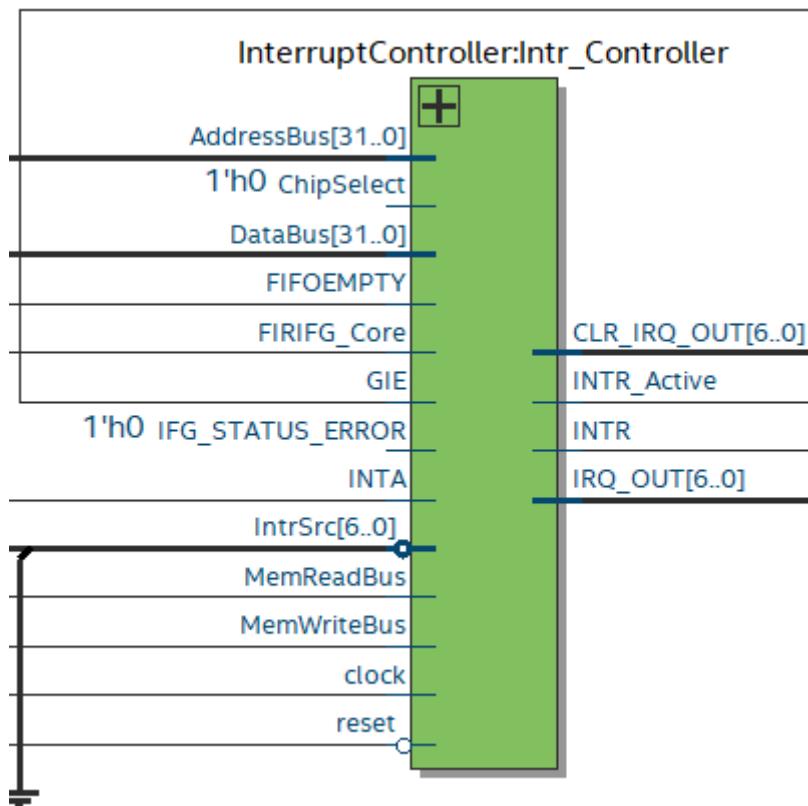
כתיבה לרגיסטר ה-CI נעשית באופן דומה לכתיבת רכיבי GPIO, ע"י ביצוע פעולות SW, AS לכתבות הזיכרון המתאימות.

קיבלה פסיקות אסינכרוניות מתבצעת באופן הבא:

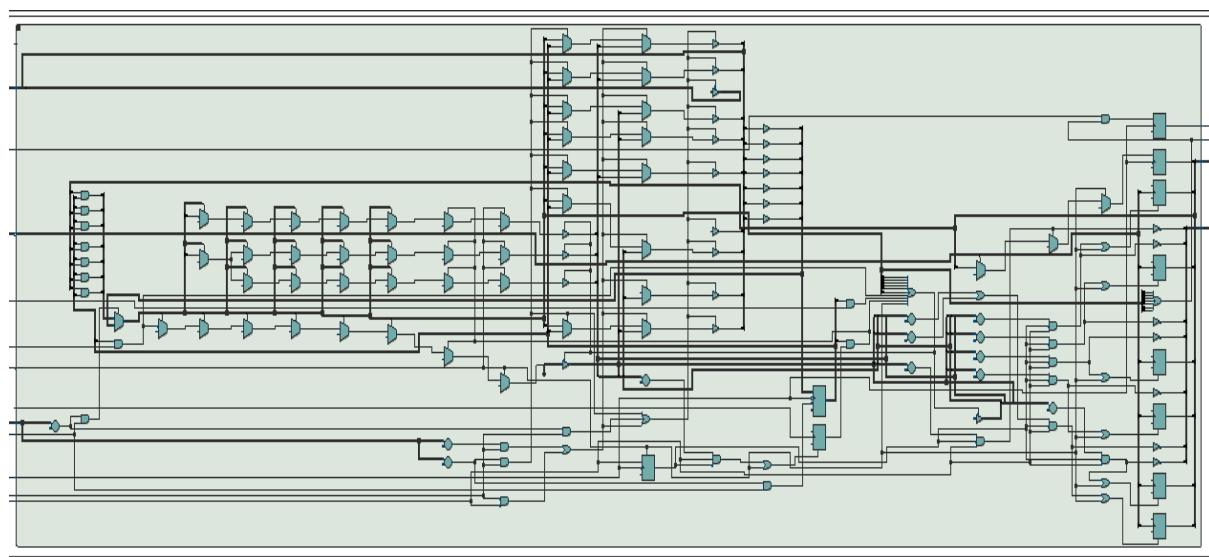
1. גורם פסיקה מבצע פעולה Trigger (לחיצה על כפתור/מטריה הטימר הגיעו לערך הנדרש וכו').
2. קו ה-INTR מעלה 1 ומתריע ל-CPU על בקשת פסיקה.

3. ה-CPU עוצר את הריצה הנוכחית ומתחילה טיפול בפסיקה:

- a. ביצוע **stall** של הפקודות עד שלב ה-EX – תכולת ה-EX תישמר לזכרון.
  - b. ביצוע **Flush** לשלבים הקודמים עד EX.
  - c. כתיבת ערך ה-GIE (Global Interrupt Enable) המתאים לפסיקה על מיקום ייעודי ל-LSB של רגיסטר 0k.
  - d. כתיבת ערך ה-type המתאים לפסיקה על מקום ייעודי ברגיסטר הפסיקות.
  - e. קפיצה לכתובת ה-ISR המתאימה (קפיצה לכתובת של רוטינת הפסיקה).
4. ה-CPU מבצע את הפקודות שברוטינת הפסיקה.
5. בהגעה לפקודת **reti** מתבצע JR לכתובת 1k, בנוסף ערך רגיסטר 1k חוזר ל-1 לטובת המשך קבלת פסיקות.



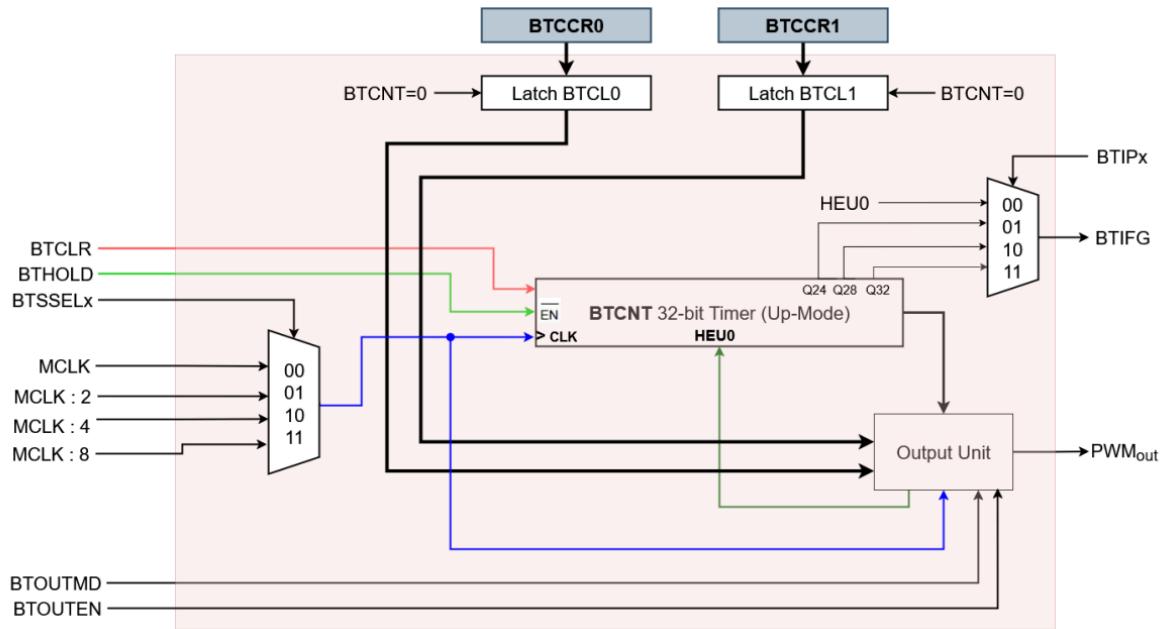
תרשים ה-RTL



## Basic Timer

מודול להפקת אוטות מחזוריים (PWM / מקור פסיקה).  
רегистרים:

- **BTCTL** – זהו רегистר השיליטה, המכיל את ביטי הבקרה השונים, כגון BTIP, BTSEL, BTHOLD ועוד.
- **BTCNT** – זהו רегистר המניה בגודל 32 ביט אשר ערכו גדול בכל עליית שעון (ניתן לבחירה) וכאשר BTIP הינו כבוי. ניתן לדגם באופן פרטני את הביטים השונים של הרегистר לקבלת פסיקות בתדר שונה, על פי שינוי ערך BTIP.
- **BTCCR0, BTCCR1** – אלו הם רегистרי ההשוואה, אשר אחראים על תדר השעון וערך ה-C.D של אות ה-PWM המצא, Out Signal.
- ערכם של רегистרים אלה נקבע ע"י התכונות כחלק מהקונפיגורציה של הטימר.  
בהתאם לערך המשווה, ערך המצא של ה-PWM מתהפרק, אשר הערך מגיע ל-T-BTCNT ערך CCR0-5 מתאפס.

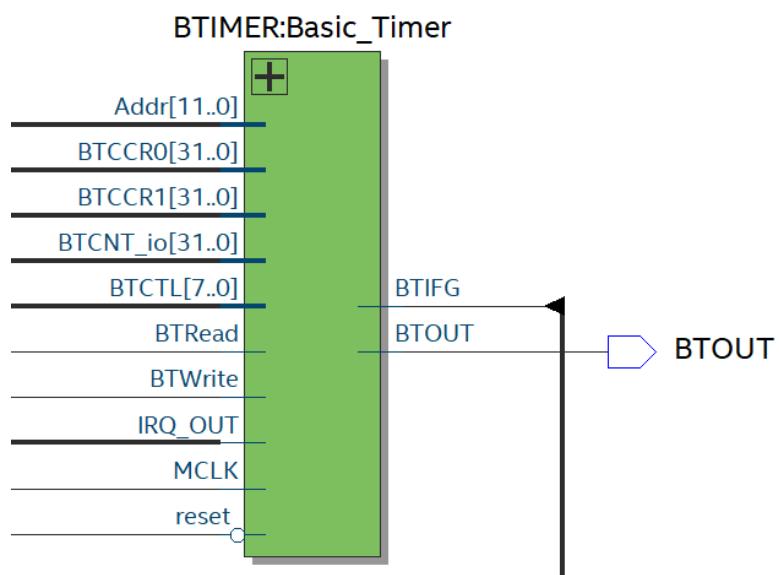


## בדיקות

• – חישוב פשוט עם כתיבה ל-LEDs.

• – מימוש טימר המעדן HEX לפי לחיצת מקשיים.

• – שינוי תדר Clock לפי מקשיים.



## תרשים ה-RTL:

