

דו"ח מסכם לפרוייקט גמר קורס

"מעבדת ארכיטקטורת מעבדים מתקדמת ומאיצי חומרה"

הפקולטה למדעי ההנדסה
אחראי מעבדה: רבוא חנניה

מגישים:

בנימין אומנסקי - ת"ז 322688946

עידו רון - ת"ז 322384330

הסבר על המערכת

בפרוייקט זה, התבקשנו לממש מעבד עם ליבת MIPS, התומך במיפוי זכרון I/O ובקבלת פסיקות חיצוניות וניהול רוטינות פסיקה. בחרנו לממש את סעיף הבנוס המשלב ליבת Pipelined MIPS CPU במקום ליבת Single Cycle ליבת ה-Pipeline ממומשת ע"י ארכיטקטורת MIPS בעלת 5 שלבים התומכת בזיהוי וטיפול בבעיות של Data dependencies תוך שימוש ביחידות Forwarding Unit ו-Hazard Detection.

את המערכת הכוללת שבנינו צרבנו ובדקנו בשבב ה-De2-115 Altera FPGA Board המשלב רכיבי פריפריה שונים כגון כפתורים, לדים, מסכי 7 segment ועוד.

MCU

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

Pipelined MIPS

GPIO

Interrupt Controller

Basic Timer

Fir Filter

בהתאם לאיור המצורף:

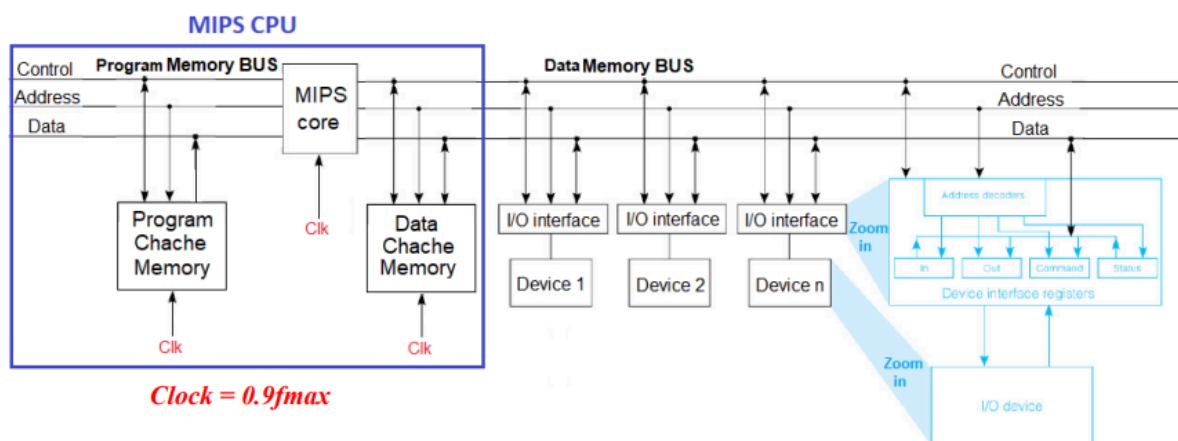
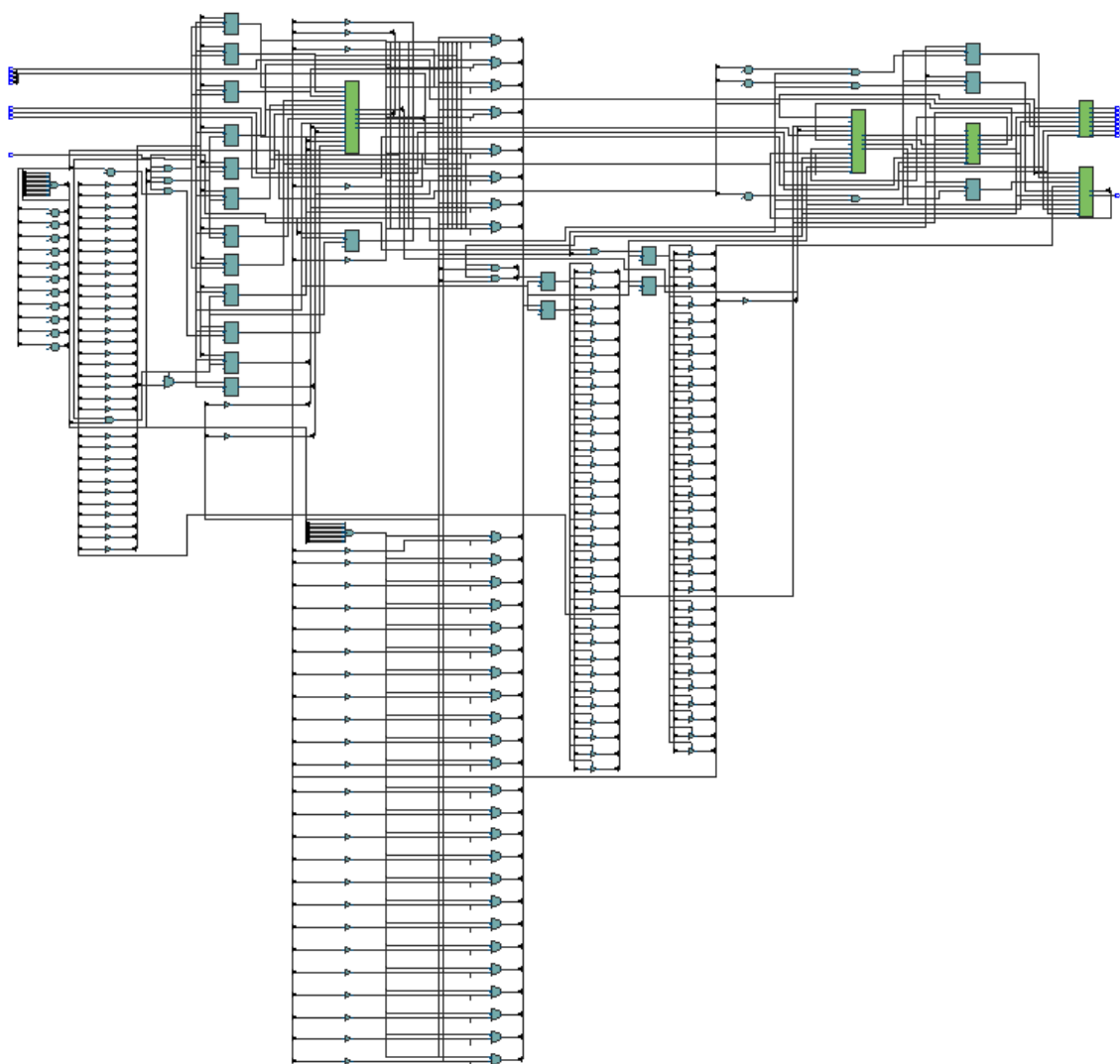


Figure 1 : MCU System architecture

קובץ זה מכיל בנוסף באס משותף הכולל 3 תתי באסים: מידע, כתובות ובקרה. הבאס המשותף הינו אמצעי התקשורת שבין ליבת הבקר לרכיבים הפריפריאליים.

להלן תרשים ה-RTL:

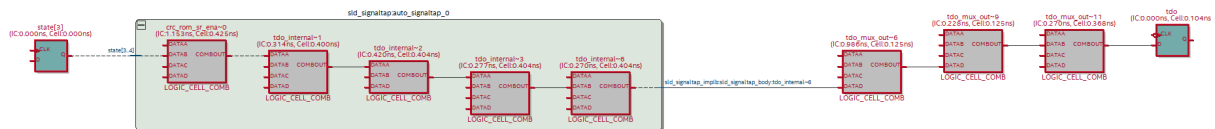


תדר מקסימלי (F_{max})

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

קיבלנו תדדר שעון מקסימלי MHz 73.44

Critical Path



Pipelined MIPS

קובץ ה-Pipelined MIPS משמש מעטפת למודולים:

IFETCH

IDECODE

EXECUTE

DMEMORY

CONTROL

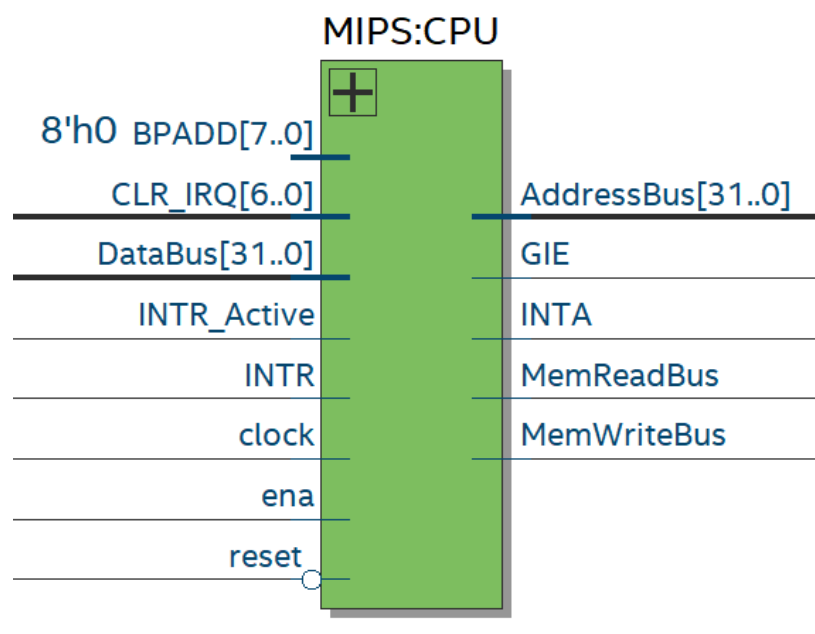
Forwarding

HazardDetection

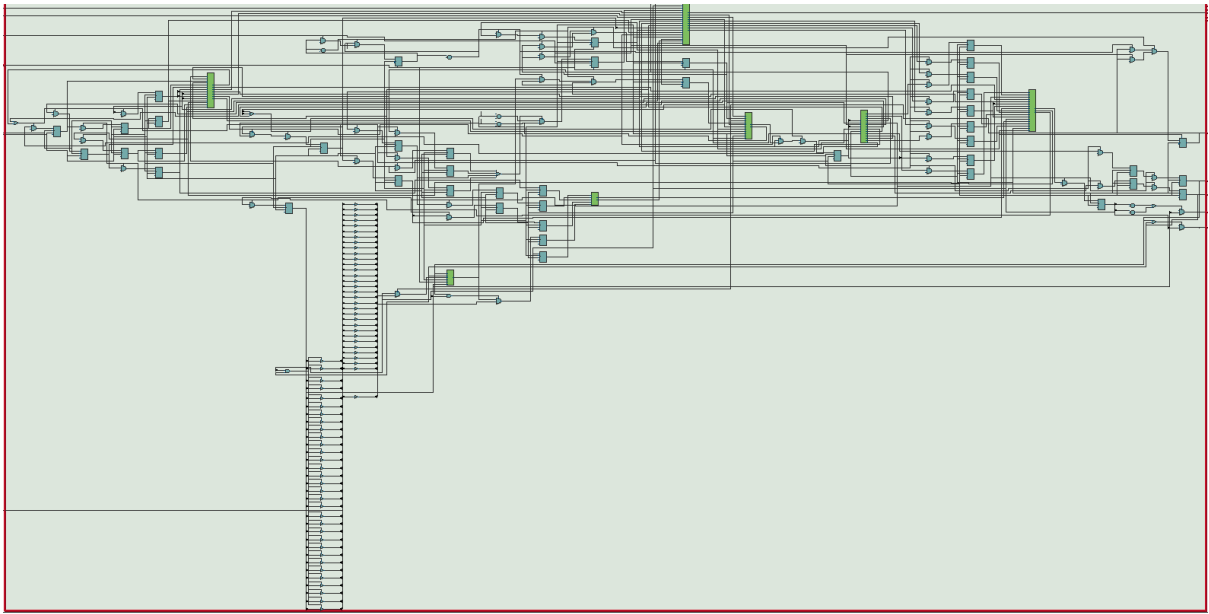
הכניסות: Reset, Clock, Ena.

היציאות: סיגנלים רבים המשמשים לניתוח וריפיקציה

ב-Modelsim או SignalTap ב-Quartus.



תרשים ה-RTL



וריקציה פונקציונלית

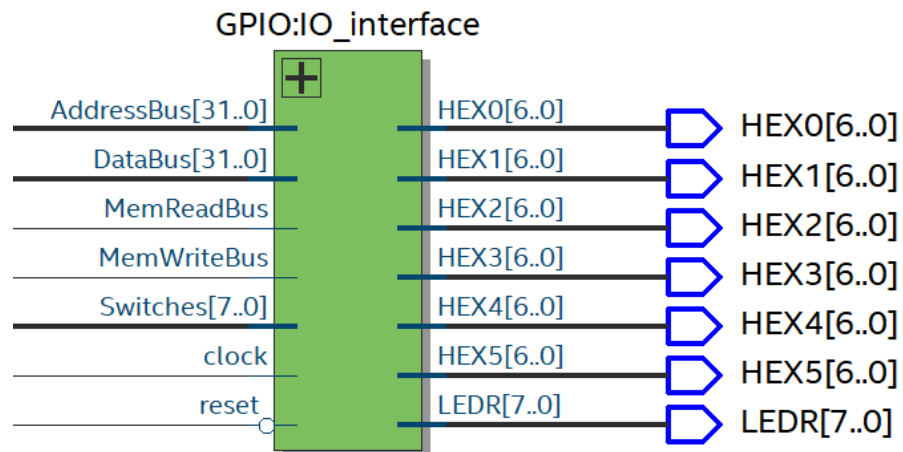
- תוכנית MIPS להרצת חיבור מטריצות (תרגום מקוד C לאסמבלי).
- סימולציות ב-Modelsim (Wave Diagram, Memory List).
- SignalTap על ה-FPGA לוריקציה בזמן אמת.

GPIO

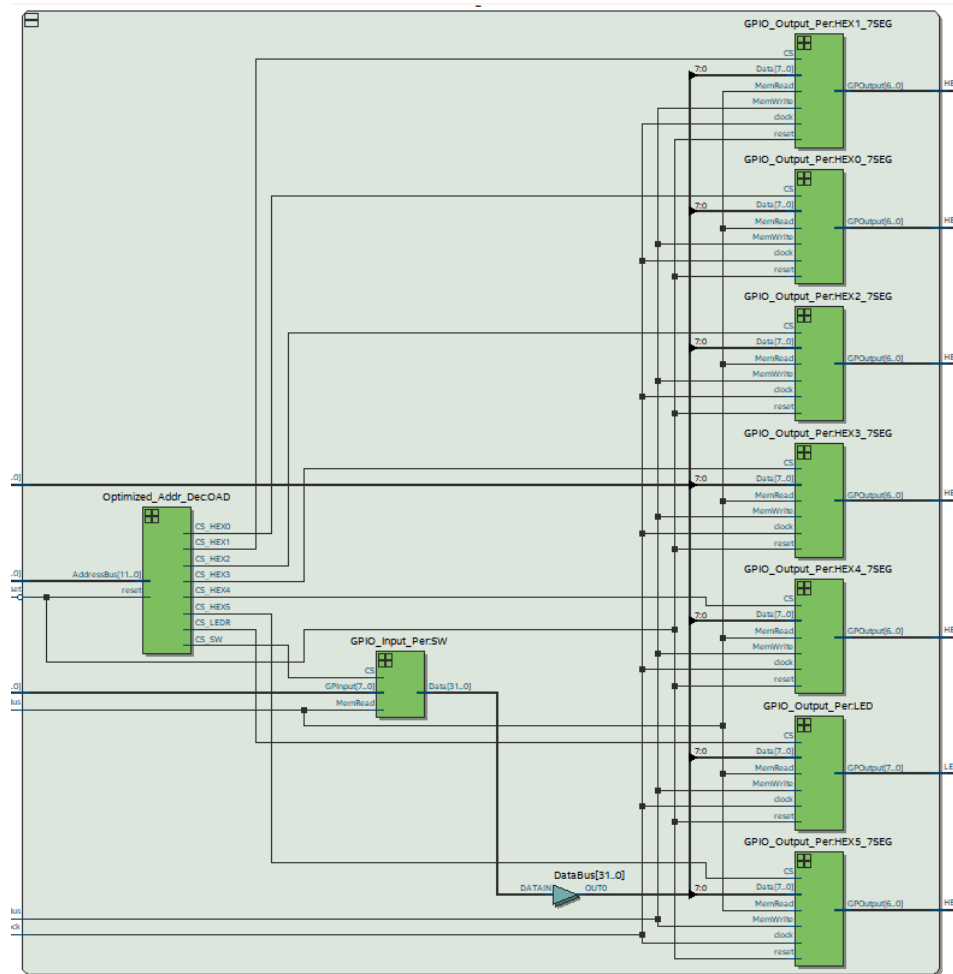
מודול לתמיכה בקלט/פלט חומרתי באמצעות Memory Mapped I/O.

- פלטים: לדים, תצוגות Hex.
- קלטים: מתגים SW0–SW7 (כולל SW8 לקביעת Breakpoint ו-SW9 לאישור Ena).
- כתובות במרחב הכתובות מ-0x800 ואילך.

בפקודת כתיבה לדוגמה ללד Hex-0 או ל-LED, כמו `sw $t0, 0x804`, מודול ה-GPIO יהיה זה שיפענח את הכתובת וידע כי יש לבצע Write לערך המידע, בזכות בחירת ה-Chip Select



תרשים ה-RTL:



Test0 – כתיבה בלולאה לערכים עולים על כל הרכיבים.

Test1 – בדיקת מצב מתגים, מונה עולה/יורד.

Test2 – בדיקת מתגים להכפלה/חלוקה.

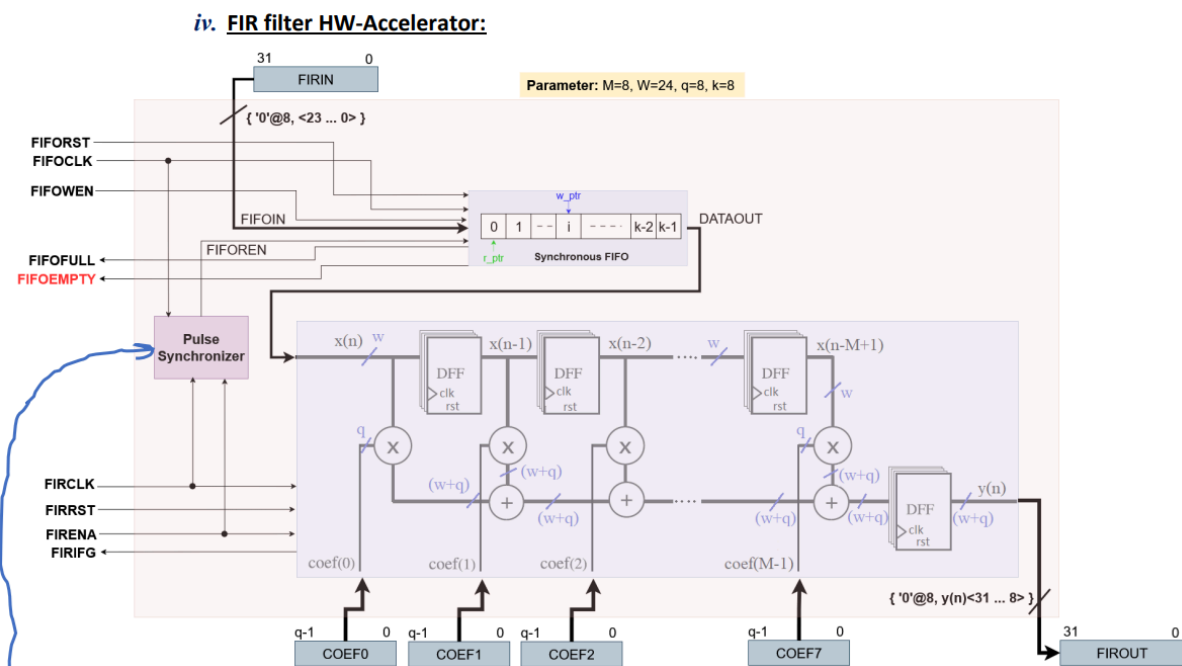
Test3 – בדיקת מתגים לחיבור/חיסור/הכפלה.

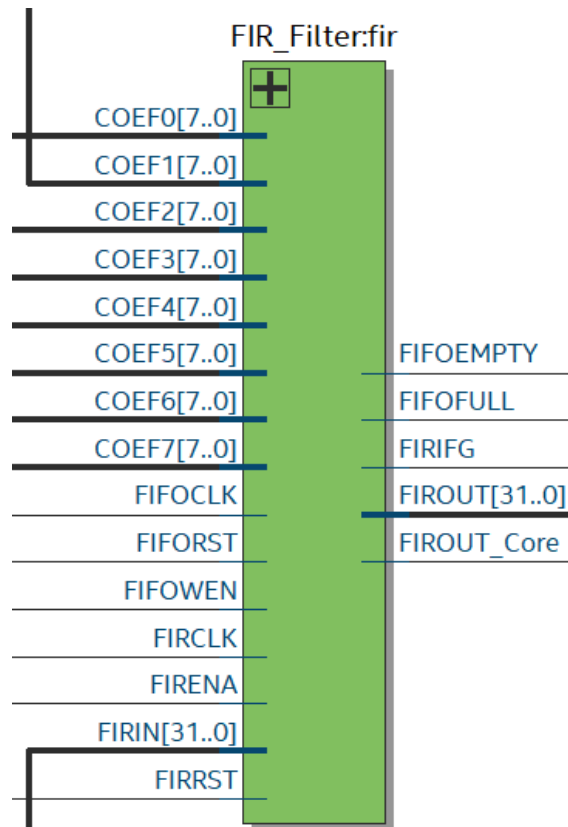
FIR FILTER

הפילטר יכלול 3 רכיבים:

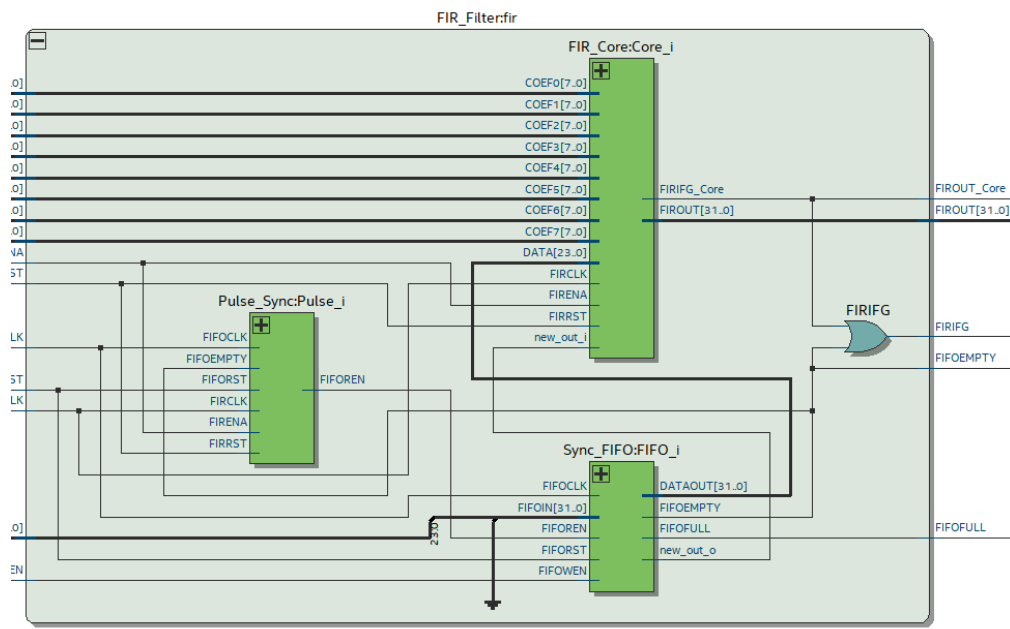
Pulse Synchronizer
Synchronous FIFO
FIR Core

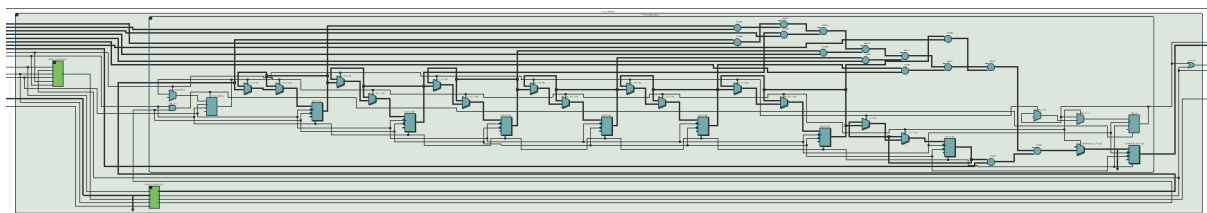
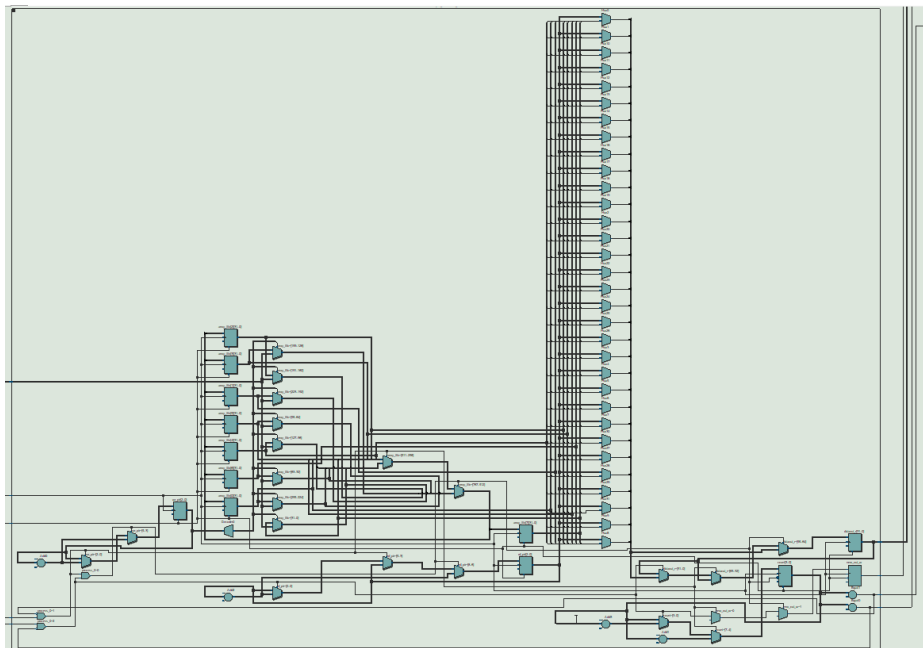
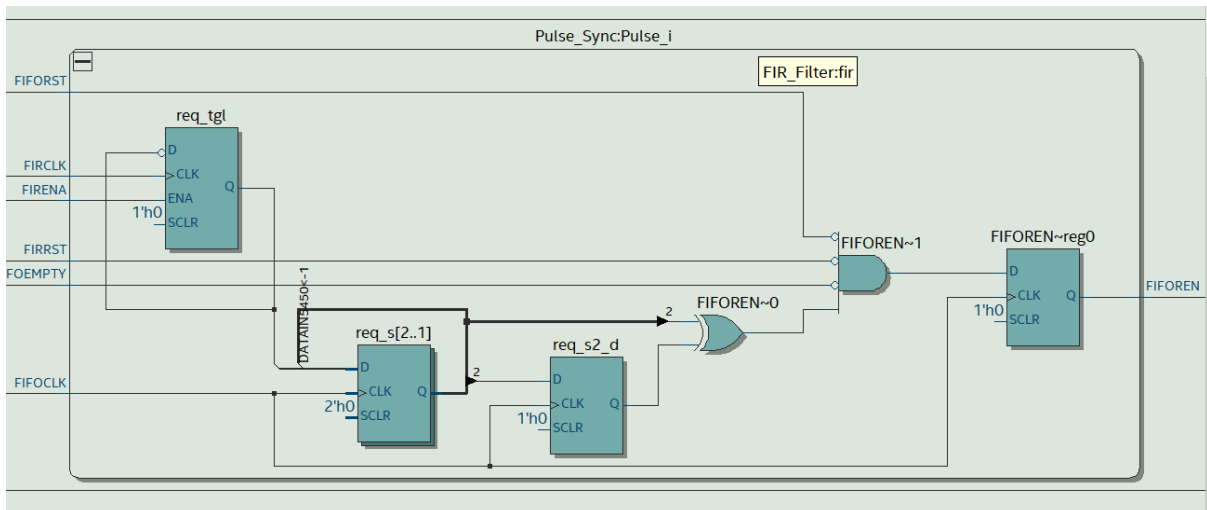
זהו מסנן ספרתי שמבצע **FIR (Finite Impulse Response)** חישוב ליניארי על רצף דגימות קלט. הוא מבוסס על סכימה משוקללת של דגימות עבר, כאשר המשקלות נקבעות על ידי מקדמים





תרשים ה-RTL:





Interrupt Controller

מודול לניהול פסיקות חיצוניות.
מקורות הפסיקה:

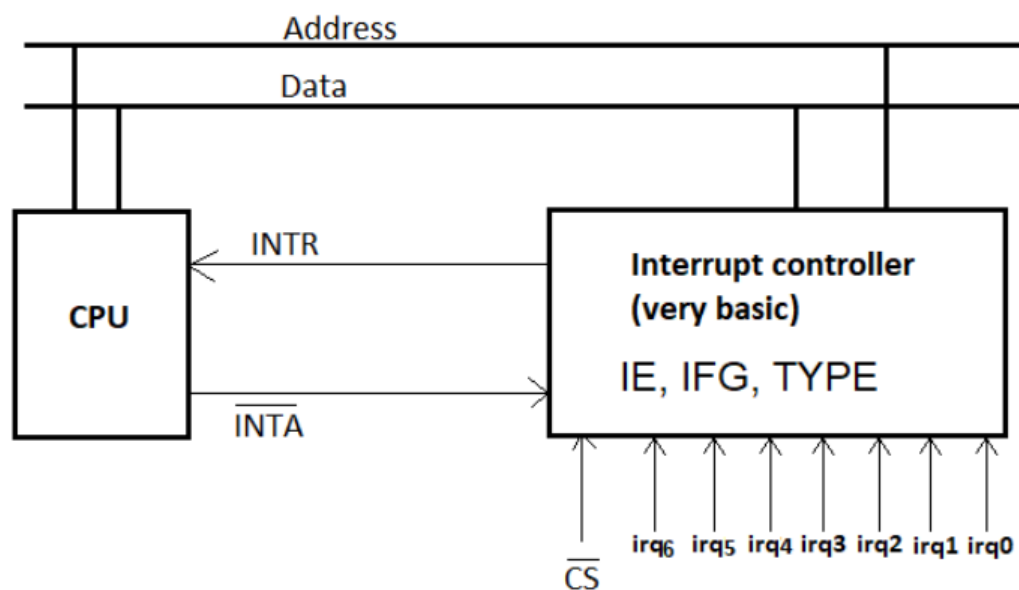
(KEY0) Reset

(KEY1–KEY3) Interrupts

Basic Timer

Fir Filter

Interrupt controller:



מקורות הפסיקה:

- לחצן **KEY0** שאחראי על ביצוע Reset לתוכנית (קפיצה לכתובת תחילת התכנית).
 - לחצנים **KEY1 – KEY3** שאחראים על יצירת פסיקות והפעלת רוטינות פסיקה ייעודיות המוגדרות בראש בתכנית.
 - **Basic Timer** – מודול נוסף שיוצר פסיקות אשר אחראי על יצירת אותות מחזוריים המשמשים כמקור פסיקה נוסף וכן כאות PWM חיצוני.
- תקשורת בין מודול ה-MIPS לפרופריט נעשית על גבי הבאס המשותף.
- כתיבה לרגיסטר ה-IC נעשית באופן דומה לכתיבה לרכיבי GPIO, ע"י ביצוע פעולות lw, sw לכתובות הזיכרון המתאימות.

קבלת פסיקות אסינכרוניות מתבצעת באופן הבא:

1. גורם פסיקה מבצע פעולת Trigger (לחיצה על כפתור/מטרייה הטיימר הגיעה לערך הנדרש וכו').
2. קו ה-INTR מעלה 1 ומתריע ל-CPU על בקשת פסיקה.

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

a. ביצוע **stall** של הפקודות עד שלב ה-EX – תכולת ה-EX תישמר לזיכרון.

b. ביצוע **Flush** לשלבים הקודמים עד EX.

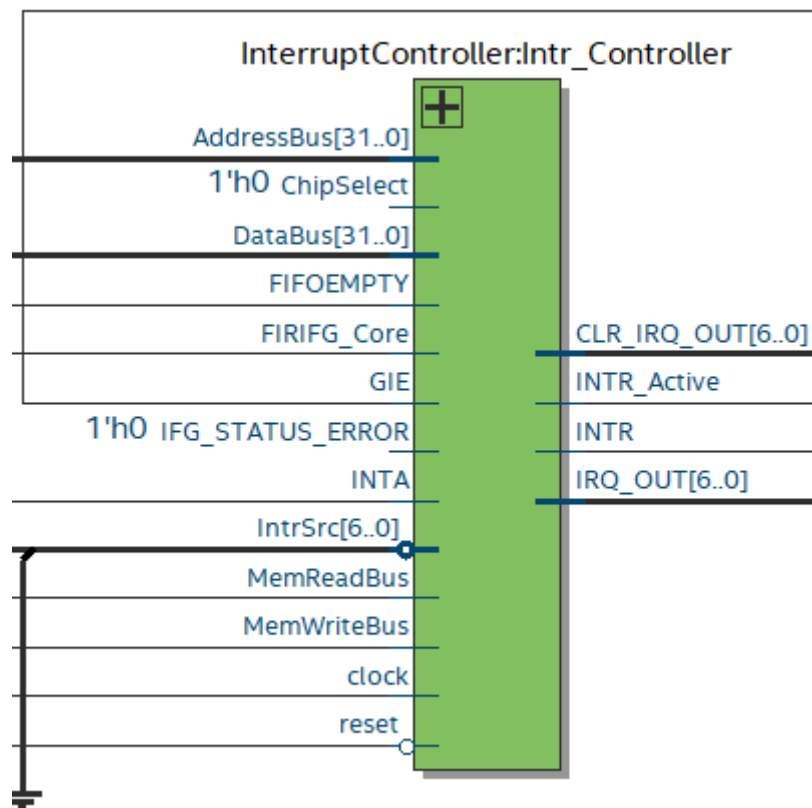
c. כתיבת ערך ה-Global Interrupt Enable (**GIE**) ל-LSB של רגיסטר k0.

d. כתיבת ערך ה-**Type** המתאים לפסיקה על מיקום ייעודי ברגיסטר הפסיקות.

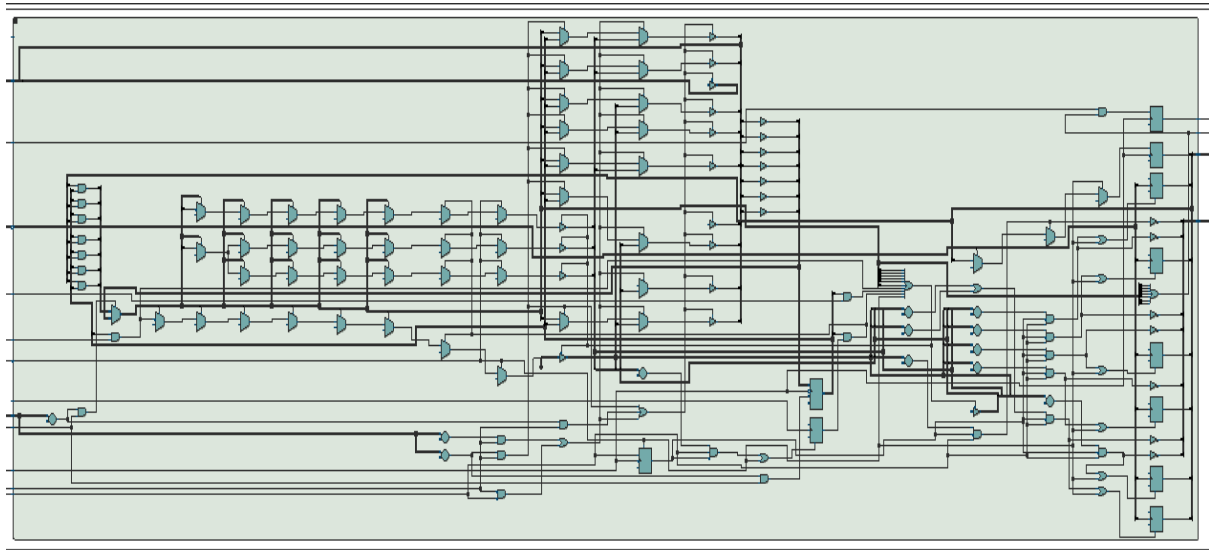
e. קפיצה לכתובת ה-ISR המתאימה (קפיצה לכתובת של רוטינת הפסיקה).

4. ה-CPU מבצע את הפקודות שברוטינת הפסיקה.

5. בהגעה לפקודת **reti** מתבצע JR לכתובת k1, בנוסף ערך רגיסטר k1 חוזר ל-1 לטובת המשך קבלת פסיקות.



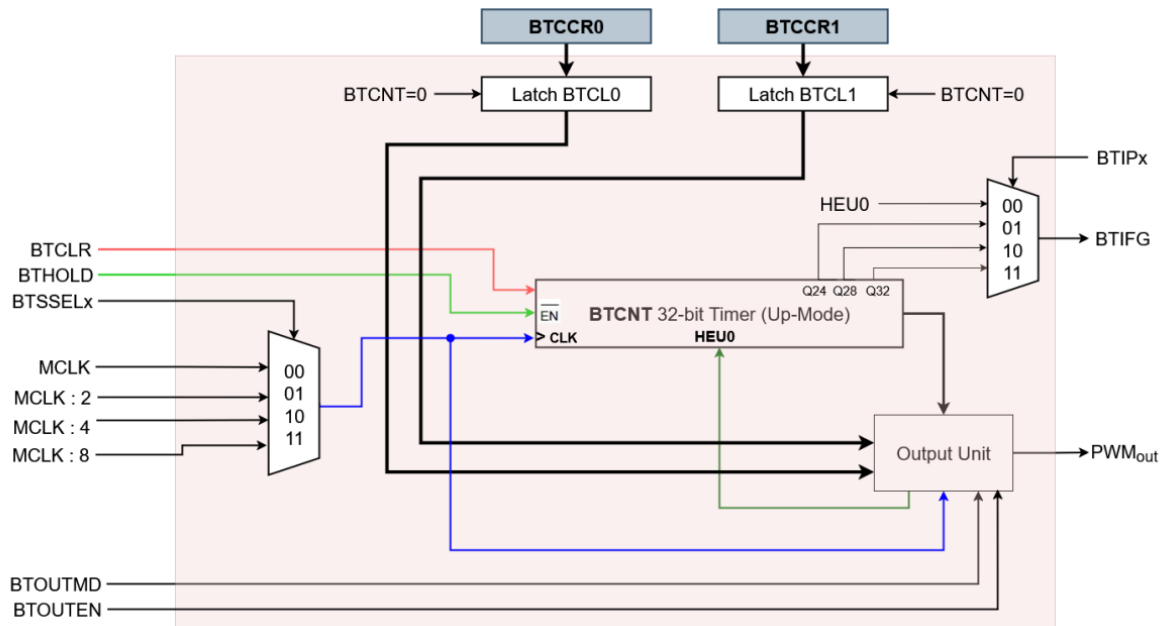
תרשים ה-RTL:



Basic Timer

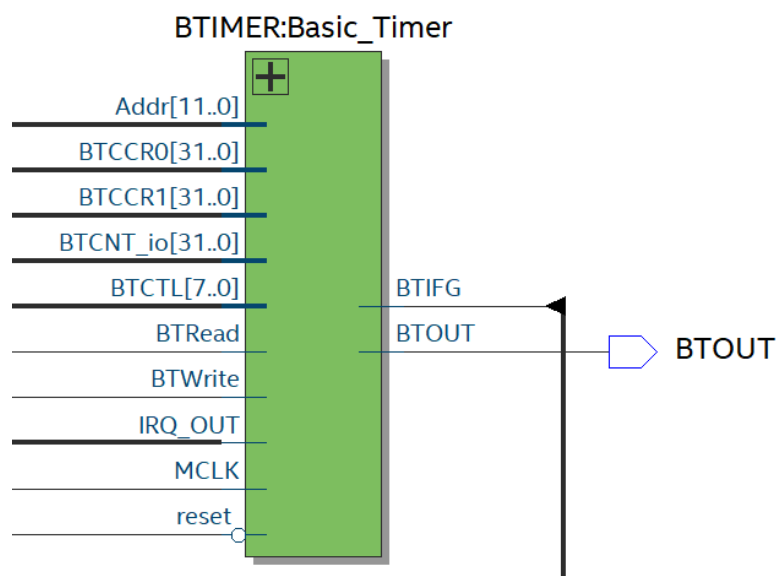
מודול להפקת אותות מחזוריים (PWM / מקור פסיקה).
רגיסטרים:

- **BTCTL** – זהו רגיסטר השליטה, המכיל את ביטי הבקרה השונים, כגון BTIP, BTSSEL, BTHOLD ועוד.
 - **BTCNT** – זהו רגיסטר המניה בגודל 32 ביט אשר ערכו גדל בכל עליית שעון (ניתן לבחירה) וכאשר BTHOLD כבוי. ניתן לדגום באופן פרטני את הביטים השונים של הרגיסטר לקבלת פסיקות בתדר שונה, על פי שינוי ערך ה-BTIP.
 - **BTCCR0, BTCCR1** – אלו הם רגיסטרי ההשוואה, אשר אחראיים על תדר השעון וערך ה-D.C של אות ה-PWM המוצא, Out Signal.
 - ערכם של רגיסטרים אלה נקבע ע"י התכנות כחלק מהקונפיגורציה של הטיימר.
- בהגעה של BTCNT לערך המשווה, ערך המוצא של ה-PWM מתהפך, ואשר הערך מגיע ל-BTCNT ערך CCR0-5 מתאפס.



בדיקות

- **Test1** – חישוב פשוט עם כתיבה ל-LEDs.
- **Test2** – מימוש טיימר המעדן HEX לפי לחיצת מקשים.
- **Test3** – שינוי תדר Clock לפי מקשים.



תרשים ה-RTL:

