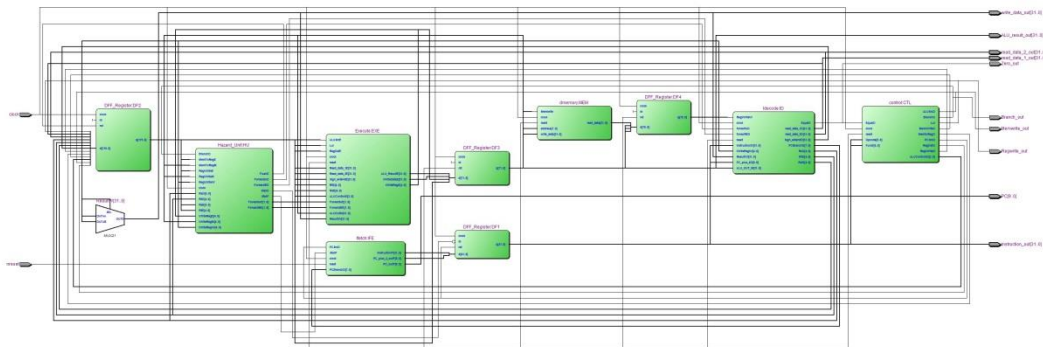


# Cpu – lab 3 – kobi eini 201553245

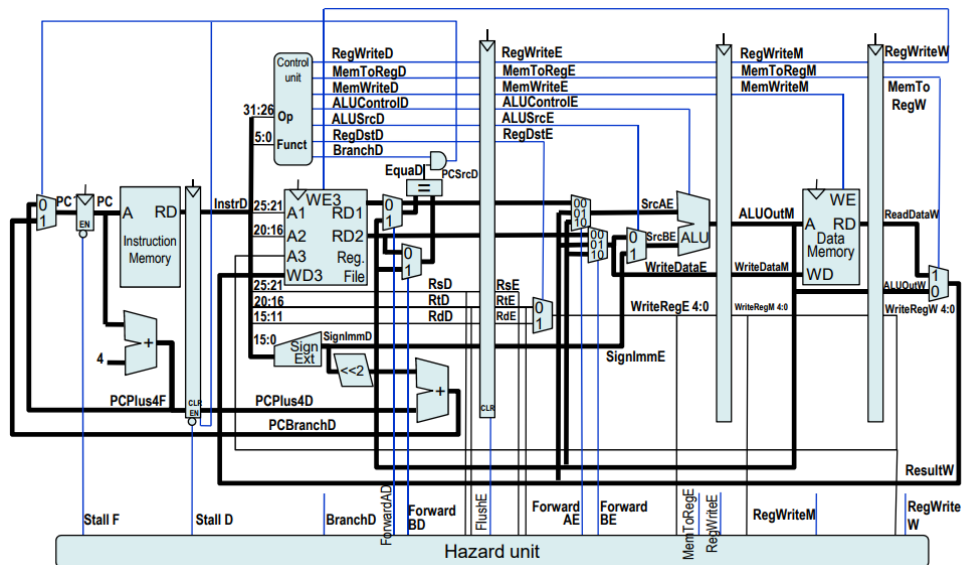
במעבדה זו התבקשנו לייצר מעבד MIPS בארכיטקטורת pipeline.

ראשית נראה את הארכיטקטורה :



איור 1 – שרטוט מעבד ה - MIPS

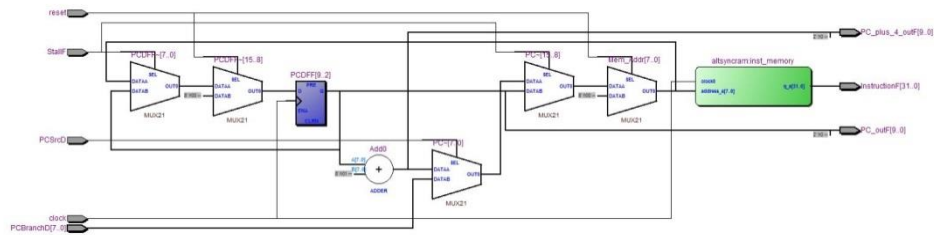
הארכיטקטורה נבנתה בהשראת השרטוט :



איור 2 – רעיון המימוש

המעבד שלנו בנוי מיחידת FETCH, יחידת DECODE, יחידת CONTROL, יחידת ECXECUTE, יחידת MEMORY, יחידת WRITEBACK.

## :FETCH

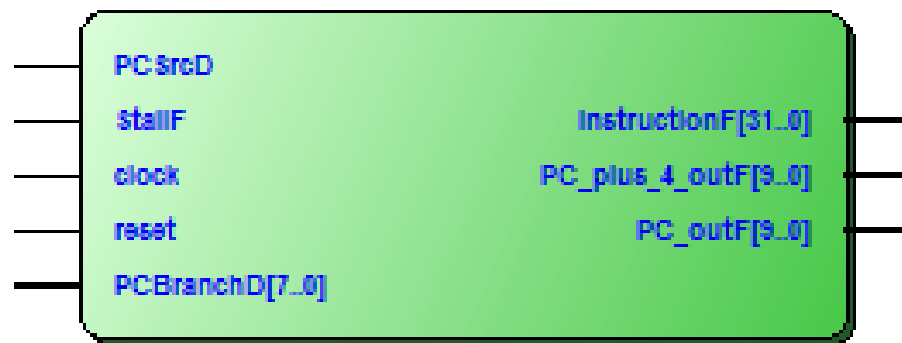


איור 3 – RTL FETCH

יחידה זו מבצעת משיכה של הפקודות מן ה - program.hex .

היחידה מבצעת קידום של ה - PC הנוכחי בארבע, ובוררת את משיכת הכתובת הבאה לפי קו PCsrcD, המתקבל מיחידת ה - DECODE, ובורר בקיום קפיצות. היחידה יכולה גם לעקב את משיכת הפקודה הבאה כטיפול ב - hazards.

### ifetch:IFE



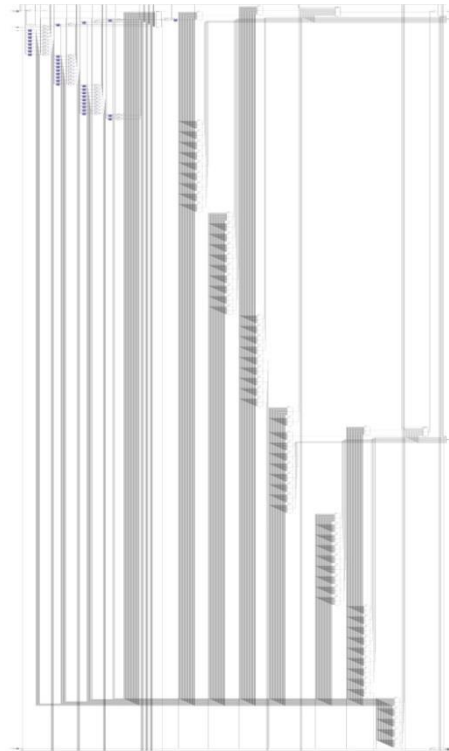
איור 4 - דיאגרמת בלוק עבור FETCH

שם	גודל	כיוון	פונקציה
PCsrcD	1	IN	ברירת כתובת הבאה
StallF	1	IN	עיקוב משיכת כתובת הבאה
PCBranchD	8	IN	כתובת עבור קפיצה
InstructionF	32	OUT	הכתובת המועבדת ליחידת ה - DECODE
PCplus4OUTF	10	OUT	לחישוב כתובת הקפיצה
PCOutF	10	OUT	מיקום ה-PC

איור 5 - טבלת פנים עבור FETCH

## :DECODE

יחידה זו מבצעת תרגום של הפקודה, מבצעת קריאה וכתיבה מ/לרגיסטרים, מחשבת כתובת מיועדת במקרה של לקיחת קפיצה והאם תנאי הקפיצה מתקיים, בנוסף, מסייעת בזיהוי האזרדים.



איור 6 – RTL DECODE



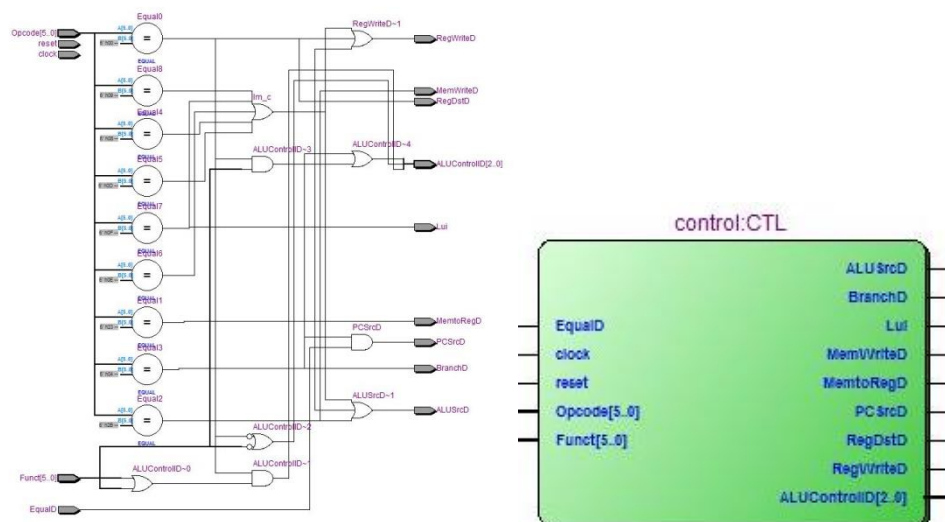
איור 7 - דיאגרמת בלוק עבור DECODE

שם	גודל	כיוון	פונקציה
RegWriteW	1	IN	לביצוע כתיבה לרגיסטר
forwarfBA	1	IN	קידום ארגומנט מיחידת ה - MEM
forwarfBA	1	IN	קידום ארגומנט מיחידת ה - MEM
Write RegW	5	IN	כתובת הרגיסטר לכתיבה
ResultW	32	IN	הערך לכתיבה
PCplus4D	10	IN	לחישוב כתובת הקפצה
ALUOUTM	32	IN	לקידום תוצאה אריתמטית לטיפול בהאזרד.
EqualD	1	OUT	האם התנאי עבור קפצה מתקיים.
Read data 1	32	OUT	תוחלת רגיסטר 1
Read data 2	32	OUT	תוחלת רגיסטר 2
Signex	32	OUT	עבור פעולת immediate
PCbranchD	10	OUT	כתובת הקפצה באם תלקח
RsD	5	OUT	רגיסטר מקור, עבור חישוב קיום האזרד
RtD	5	OUT	רגיסטר ארגומנט, עבור חישוב קיום האזרד
RdD	5	OUT	רגיסטר יעד, עבור חישוב קיום האזרד

איור 8 – טבלת פינים עבור DECODE

## :CONTROL

יחידה זו מבצעת חישובים לוגיים, לקביעת קווים עבור סנכרון המערכת וביצוע הפקודות.



איור 9 – RTL CONTROL ודיאגרמת בלוק

שם	גודל	כיוון	פונקציה
EqualD	1	IN	לחישוב PCsrcD, לביצוע קפיצה או לא
Opcode	6	IN	סוג הפקודה לביצוע
Funct	6	IN	עבור חשבו הקוד ליחידת ה- ALU
ALUsrc	3	OUT	לחישוב הפעולה לביצוע ביחידה ה- ALU
BranchD	1	OUT	לזיהוי האזרדים
Lui	1	OUT	עבור פקודת טעינה ל – 16 הגבויים
MEMWriteD	1	OUT	לביצוע כתיבה לזיכרון
MEMtoReg	1	OUT	לביצוע כתיבה מזיכרון לרגיסטר
PCsrcD	1	OUT	1 – ביצוע קפיצה, 0 – לא
RegDestD	1	OUT	ברירת רגיסטרים לבצוע פקודות ב - EXECUTE
RegWriteD	1	OUT	לסינכרון כתיבה לרגיסטר
ALUcontrol	3	OUT	הפעולה לביצוע ביחידה ה- ALU

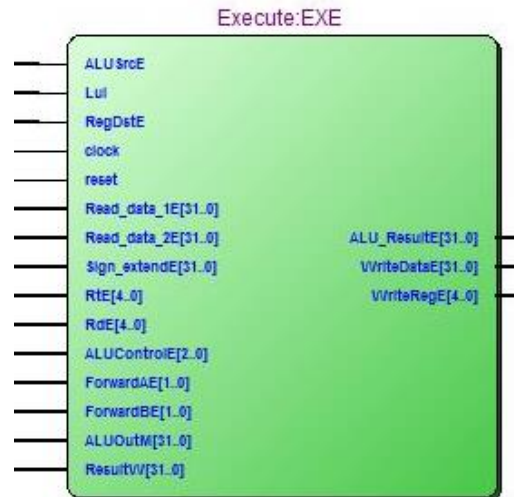
איור 10 – טבלת פנים עבור CONTROL

### EXECUTE:

יחידה לביצוע הפעולות המתמטיות והלוגיות. בארכיטקטורת ללא פייפליין, יחידה זו גם לפעמים מבצעת את חישוב הכתובות לקפיצה. בעיצוב שלנו, יצרנו יחידה נפרדת לחישוב הכתובות בשלב ה – ID. פעולה זו יצרה בעיית סינכרון עליה לא ניתן דגש, עב חוסר זמן, ולכן יש צורך הכנסת פקודת nope, טרם קפיצה. לבעיה זו ניתן פתרון מאוחר יותר. ביחידה זו הוספנו גם את הרכיבים לביצוע פעולות על יצוג IEEE.



איור 11 – RTL EXECUTE



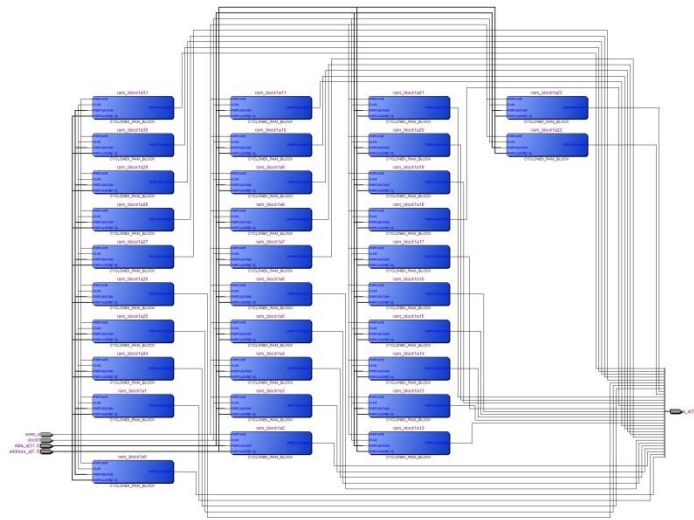
איור 12 – דיאגרמת בלוק עבור EXECUTE

שם	גודל	כיוון	פונקציה
ALUSrc	1	IN	ברירת מקור האופרנד עבור כניסה B
Lui	1	IN	עבור כתיבה לערך עליון של רגיסטר
RegDest	1	IN	ברירת רגיסטר לכתיבה
ReadData1	32	IN	אופרנד 1
ReadData2	32	IN	אופרנד 2
SignEx	32	IN	עבור פקדות immediate
RtE	5	IN	עבור זיהוי הזארדים
RdE	5	IN	עבור זיהוי הזארדים
ALUcontrol	3	IN	הפעולה לביצוע על ידי יחידה ה - ALU
ForwardA	2	IN	לקידום אופרנד עבור טיפול בהזארד
ForwardB	2	IN	לקידום אופרנד עבור טיפול בהזארד
ALUOUTM	32	IN	לקידום אופרנד עבור טיפול בהזארד
ResultW	32	IN	לקידום אופרנד עבור טיפול בהזארד
ALU_Result	32	OUT	מוצא ה - ALU
WriteDataE	32	OUT	לכתיבה לזיכרון
WriteRegE	5	OUT	לזיהוי הזארדים

איור 13 – טבלת פנים עבור EXECUTE

## :MEMORY

יחידה זו מבצעת את הקריאה והמשיכה אל ומהזיכרון.



איור 14 – RTL MEM



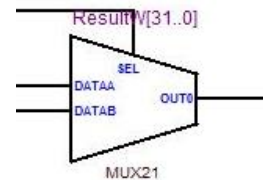
איור 15 – דיאגרמת בלוק עבור MEM

שם	גודל	כיוון	פונקציה
Memwrite	1	IN	לביצוע כתיבה לזיכרון
Address	8	IN	כתובת זיכרון לכתיבה
Write_data	32	IN	ערך לכתיבה לזיכרון
ReadData	32	OUT	הערך הנקרא מהזיכרון

איור 16 – טבלת פנים עבור MEM

## :WRITEBACK

יחידה לביצוע ברירת הערך המוחזר לרגיסטרים של המעבד. יחידה זו היא למעשה רק MUX.



איור 17 – RTL WRITEBACK

שם	גודל	כיוון	פונקציה
ReadDataW	32	IN	מידע מהזיכרון
ALUOUTW	32	IN	מידע מיחידת ה – ALU
MEMtoReg	1	IN	ברירה בין הקווים הנ"ל
ResultW	32	OUT	מוצא ה – MUX

איור 18 – טבלת פנים עבור WRITEBACK

## ניתוח מסלול מקסימאלי:



איור 19 – מסלול קריטי

כמו שניתן לראות, המסלול הקריטי נמצא בשלב ה – EXECUTE, במחבר ה – IEEE. כמו במעבדה הקודמת, פעולה זו דורשת הכי הרבה זמן, עקב שילוב כל המחברים, ולכן תקבע את זמן המחזור הכולל של המערכת לתפקוד תקין.

על מנת לייעל את זמן המחזור, ניתן לחלק את הרכיבים המשתתפים בפעולת חיבור ה-IEEE, בין שלבי ה – pipeline, או להוסיף שלבים ל – pipeline.



## להלן מספר היחידות הלוגיות שנדרשו לכל יחידה:

ty	Logic Cells	Dedicated Logic Registers	I/O Registers
MIPS	3007 (33)	1282 (0)	0 (0)
control:CTL	14 (14)	0 (0)	0 (0)
DFF_Register:DF1	41 (41)	40 (40)	0 (0)
DFF_Register:DF2	100 (100)	99 (99)	0 (0)
DFF_Register:DF3	235 (235)	72 (72)	0 (0)
DFF_Register:DF4	71 (71)	71 (71)	0 (0)
Execute:EXE	943 (322)	0 (0)	0 (0)
Hazard_Unit:HU	41 (41)	0 (0)	0 (0)
Idecode:ID	1528 (1528)	992 (992)	0 (0)
Ifetch:IFE	17 (17)	8 (8)	0 (0)
dmemory:MEM	0 (0)	0 (0)	0 (0)

Total logic elements	3,962
Total combinational functions	3,007
Dedicated logic registers	1,282

איור 20 - יחידות לוגיות

כמו שניתן לראות, סה"כ יחידות לוגיות למימוש המבנה הינו 3962.

	Fmax	Restricted Fmax	Clock Name	Note
1	18.38 MHz	18.38 MHz	clock	

איור 21 - תדר מקס'

כמו שניתן לראות התדר המקס' הינו 18.38 מגה הרץ.

הסבר על ה – tb:

הערכים המוכנסים בקובץ ה data הינם:

```

7 .data
8 i: .word 0x80000001
9 j: .word 0x10000002
10 g: .word 0x00000005
11 h: .word 0x00100004
12 f: .word 0x00000005
13 k: .word 0x80000005
14 l: .word 0x00000007
15 m: .word 0x00000002
16 # num in rising order : 0x80000005 0x80000001 0x00000002 0x00000005 0x00000005 0x00000007 0x00100004 0x10000002

```

איור 22 - קלט המיון

הערכים מיוצגים ב – IEEE וכוללים מספרים חיוביים ושליילים, בעלי מנטיסה ושבר דומים ושונים. זאת על מנת להריץ טסט כולל כניתן.

לאחר הרצת התוכנית, על מנת שנוכל לעקוב אחרי שינוי קובץ הדאטה, ביצעתי קריאה של קובץ הדאטה לרגיסטרים 9-17 :

```
+ /mips_tb/U_0/ID/register_array ... {80000005} {80000001} {00000002} {00000005} {00000005} {00000007} {00100004} {10000002} ...
```

איור 23 - פלט המיון

כמו שניתן לראות, הערכים מוינו נכון לפי גודלם.