

תמונה של מעבד [1]

## 1. תקציר הפרויקט:

שם הפרויקט: מימוש של מעבד 16-ביט מסוג HACK

שמות המגשים : אוריאל קובלר – 312496045 מנדל ולס - 312250087

מנחים : פרופ' אבי כספי ד"ר שמעון מזרחי

### 1.1 מטרת הפרויקט:

כיום כמעט בכל רכיב אלקטרוני קיים יחידת עיבוד (CPU) פנימית קטנה שיש לה תפקידים רבים. במהלך הלימודים נגענו באבני היסוד של הארכיטקטורה של מעבד MIPS פשוט, ולכן בחרנו להעמיק ולהרחיב את הידע בבניית מעבד משלבי התיכנון ועד מימוש על חומרה.

## 2. תיאור הפרויקט בקצרה:

### 2.1 היסטוריה של המעבד:

המחשבים הראשונים שהומצאו היו עצומים בנפח ומאוד יקרים. בשנות ה-60 של המאה עשרים החלו להופיע מעגלים מוכללים שאפשרו מזעור של טרנזיסטורים רבים על גבי חתיכה אחת ובכך בעצם הקטינו את גודל המחשב. אך עדיין ה-CPU היה מורכב ממספר רב של רכיבים. נקודות המפנה הייתה בשנות ה-70 כאשר הצליחו לדחוס CPU שלם על גבי מעגל מוכלל, כלומר שבב יחיד, וזהו מה שאנו קוראים כיום מעבד.

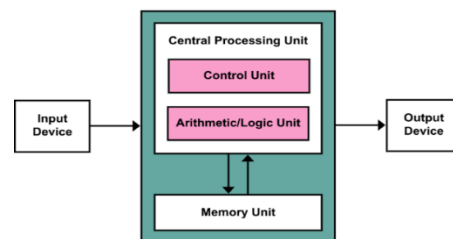
### 2.2 אז מה זה מעבד?

מעבד, יחידת העיבוד המרכזית (CPU - Central Processing Unit), הוא החלק המרכזי במחשב, שמבצע את הפקודות המאוחסנות בזיכרון המחשב. פקודות אלה הן פקודות אלמנטריות ביותר, כשכל פקודה בנויה מרצף קצר של ביטים, שמשמעותו מובנת על-ידי המעבד. רצף זה קרוי שפת מכונה. כל דגם של מעבד מתאפיין בסט הפקודות שלו.

2.3 הבסיס לכל המחשבים הקיימים כיום הם שני ארכיטקטורות בסיסיות של Von Neumann ושל Harvard ננסה להבחין בכמה הבדלים ביניהם ובסוף נבחן כיצד מעבד ה-HACK בנוי.



ארכיטקטורת Harvard [3]



ארכיטקטורת Von Neumann [2]

- הארכיטקטורה של Von Neumann יותר פשוטה.
- Von Neumann לוקח הוראה או מהתוכנית או מהמשתמש ואילו Harvard יכול שניהם בו זמנית.

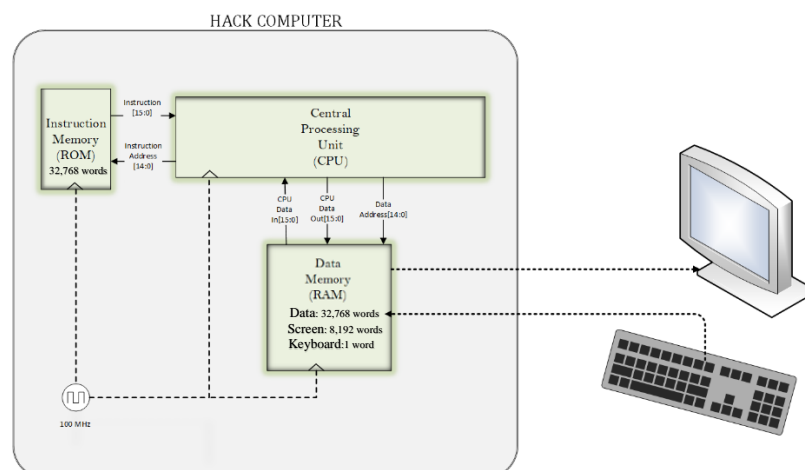
- ב-Harvard הזיכרון לא מנוצל עד הסוף כי הוא (זיכרון הנתונים וזיכרון התוכנית) מוקצה שני מקומות נפרדים ואילו Von Neumann מכיוון שזה באותו מקום אז הניצול יותר טוב.
- מעבד ה-HACK** כפי שנראה יש לו את היתרון של היחידה אחת שהוא יחסית פשוט, ומאידך יש בו את היתרון של Harvard שניתן לתקשר בו במקביל.
- 2.4 כיום כמעט בכל מכשיר קיים מעבד כלשהו שמבצע את הפעולות הנדרשות למכשיר, זהו ליבו של כל מכשיר, החל מהמחשבים טאבלטים טלפונים חכמים, וכן מכונת כביסה מזגן וכו'.

### 2.5 אז למה דווקא בחרנו ב-HACK?

תכנון המחשב HACK הינו תיאורטי כמתואר ב-[4], והוא מיועד לבניה וירטואלית מעשית לצורך לימוד והבנה. היתרון בו הוא פשטות המימוש תוך שמירה על עקרונות בניית המחשבים כפי שעיצב זאת Von Neumann.

המטרה שלנו היא להצליח לבנות את המחשב תוך כדי מימוש על FPGA והשתמשות בו, וכמובן להעמיק ולהבין את כל תהליכי הבניה מראשית ועד תום.

### 3. דיאגרמת בלוקים:



### 3.1 הסבר על המבנה של מחשב ה-HACK :

ארכיטקטורת המחשב בנויה על נתונים ברוחב של 16-bit.

#### 3.1.1 הבלוקים הפנימיים של המחשב

- יחידת עיבוד CPU, זיכרון נתונים וזיכרון הפקודות

#### 3.1.2 יחידות חיצוניות שהמחשב תומך בהם

- I/O מסך ומקלדת

### 3.2 הסבר על תפקיד הבלוקים במעבד:

- 3.2.1 ה-CPU : הינו היחידה המרכזית של המחשב, שבה מתבצעת כל העיבוד של המידע ומכיל בתוכו ALU (Arithmetic Logic Unit) שאחראי לבצע פעולות אריתמטיות

בסיסיות לדוגמה חיבור וחיסור וגם פעולות לוגיות בסיסיות כגון AND ו-OR, ובנוסף ה-CPU מכיל יחידת בקרה שיוודעת לנתב כל הוראה שנמצאת בזיכרון הפקודות ולוודא שהפקודה מתבצעת כראוי וגם PC (Program Counter) לחשב את הכתובת של ההוראה הבאה, ואח"כ לאחסן את התוצאה בזיכרון הנתונים, ובנוסף מכיל רגיסטרים שעליהם מתבצע העיבוד הנדרש.

3.2.2 זיכרון נתונים: הינו מסוג RAM (Random access memory) כאשר הגישה לכל נתון היא על פי כתובת ואין יחס למידע שמאוחסן בכתובת זו או אחרת, זיכרון הנתונים מאוחסן בכתובות  $0x0000 - 0x3FFF$ , כך שיש סה"כ  $16K$  words של נתונים שניתנים לקריאה וכתובה, ובכתובות  $0x4000 - 0x6000$  מאוחסן המידע ששייך ליחידת I/O (ראה הסבר בהמשך 3.2.4).

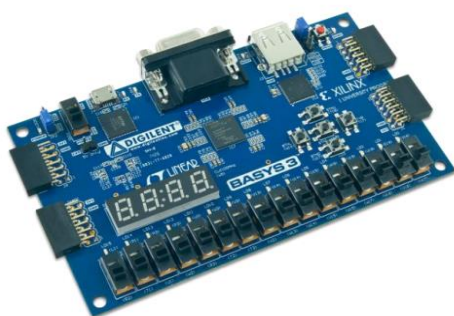
3.2.3 זיכרון הפקודות: הינו מסוג ROM (Read Only Memory), ומאפשרת רק קריאה של הרגיסטרים בלבד, רוחב וקטור הכתובות של הזיכרון גם כאן הינו 15 כך שישנם  $2^{15}$  Addresses, בהינתן כתובת על ידי ה-CPU הפקודה שנמצאת באותה כתובת נכנסת לרגיסטר Instruction Register שמכיל תמיד את הפקודה הנוכחית.

3.2.4 I/O מסך ומקלדת: המעבד תומך במסך שחור לבן של  $256 \times 512$  מערך דו מימדי לינארי של סה"כ  $2^{13}$  כתובות המורכבות מ-16bit (דהיינו  $256 \times 512 = 2^{17}$ ) כל אחת כך שכל ביט במערך הוא "דלוק" (כלומר '1') אזי התצוגה תהיה שחורה ואם "מכובה" (כלומר '0') אז לבנה, המעבד תומך גם במקלדת התווים נשמרים בקוד ASCII במיקום האחרון של זיכרון הנתונים.

3.3 שפת המכונה של המעבד מכילה שני סטים של פקודות A-instruction ו-C-instruction של אפסים או אחדות bit-16 שבאמצעותן ניתן לשלוט במעבד בפעולות האריתמטיות והלוגיות על הרגיסטרים המיועדים לכך, וכן פעולות קפיצה למקום אחר בזיכרון ופעולות של טעינה ואחסון של נתונים בזיכרון.

3.3.1 A-instruction מאופיין ע"י שה-MSB שווה 0, הפקודה מכניסה לרגיסטר A את הערך של המספר הבינארי של שאר הביטים, ובשפת אסמבלי זה נראה כך "@aaa" כאשר המספר "aaa" יכנס לרגיסטר A.

3.3.2 C-instruction מאופיין ע"י שה-MSB שווה 1, ובאמצעותו ניתן לבצע פעולות אריתמטיות ולוגיות ולהשתמש ב-ALU באמצעות ה-6 ביטים פקודה ניתן לברור בין הפעולות שה-ALU מבצע, וגם פקודות הסתעפות מותנת של בדיקה האם הערך ברגיסטר גדול קטן או שווה לאפס ולקפוץ לכתובת רצויה בזיכרון הפקודות, וגם לאחסן נתונים בזיכרון ה-RAM.



4. הערכה הנדסית לאופן מימוש המערכת על ה-Basys3:

4.1 הנתונים המרכזיים של הלוח[5]:

4.1.1 Artix-7 של חברת Xilinx

4.1.2 1,800 Kbits of fast block RAM

USB host connector	4.1.3
VGA connector	4.1.4
100 MHz oscillator	4.1.5
Pushbuttons	4.1.6
Shared UART/ JTAG USB port	4.1.7

ועוד מספר חלקים שמפורטים בדפי הנתונים שיתכן שנראה צורך נוסף להשתמש בהם.

4.2 מימוש החומרה יתבצע בשפת SystemVerilog על רכיב ה - Artix-7 , ובנוסף נעזר ברכיבים הפריפריאליים של הלוח, את המסך נחבר דרך ה - VGA ואת המקלדת דרך ה - USB. מדפי הנתונים עולה כי על הFPGA ישנו זיכרון RAM בגודל של 1800 Kbits ובו נשתמש לצורך מימוש זיכרון הנתונים. ואת זיכרון ההוראות נממש על הFPGA עצמו. לצורך טעינת זיכרון הפקודות אל הלוח נשתמש ב - UART במקום לסנתז מחדש את המעבד.

#### 4.3 סביבת הפיתוח:

לצורך הסימולציה נשתמש בתוכנת ModelSim	4.3.1
את הסינטזה וה - Place & Route והצריבה בתוכנת Vivado	4.3.2

#### 4.4 רשימת ציוד הנדרש:

לוח Basys3	4.4.1
מסך בעל חיבור VGA	4.4.2
מקלדת	4.4.3

5. הצעה למערך בדיקה (שלב הוורפיקציה וולידציה): כדי לבדוק שהמעבד פועל כנדרש נכתוב לכל בלוק פנימי Testbench שבדק שהמימוש נעשה כראוי, ואחרי שהמעבד נבדק נצרום את הקוד על רכיב ה - FPGA, ונטען תוכנית בשפת המכונה של המעבד ונדגים שימוש של המעבד ע"י יחידות הקלט / פלט.

## 6. תרשים גאנט:

התוכנית היא לשבת כ-12 שעות במוצאע בשבוע על הפרויקט כדי לעמוד ביעדים הבאים:

מס' שבועות	התחלה	מוקצה ל	תיאור אבן דרך
11	1/8/2022	ביחד	<b>דוח הכנה</b>
		ביחד	לימוד על מבנה המחשב
		מנדל	הבנת ארכיטקטורה של המעבד
		אוריאל	הבנת תקשורת I/O עם המעבד
		ביחד	מציאת FPGA וערכת פיתוח מתאימה לפרויקט
		ביחד	כתיבת דוח הכנה
12	19/10/22	ביחד	<b>מימוש של המעבד</b>
		מנדל, אוריאל	לימוד וריענון ורילוג וכלים לסימולציה
		מנדל	CPU
		אוריאל	RAM
		מנדל	ROM
		אוריאל	תשתית ל-UART ו-I/O
10	15/2/2023	ביחד	<b>וורפיקציה</b>
		מנדל, אוריאל	לכתוב תוכנית בדיקה
		מנדל, אוריאל	לחשוב ולנתח מקרי קצה ונקודת תורפה
		מנדל, אוריאל	כתיבת טסטבנגן לכל בלוק
7	22/3/2023	ביחד	<b>סינתזה + R&amp;P</b>
		אוריאל	אינטגרציה של הזיכרון RAM
		מנדל	בדיקת בעיות זמנים
		אוריאל	כתיבת קובץ XDC (לחיבוריות לפינים של הבורד)
		ביחד	התבוננות בתוצאות והשוואה לארכיטקטורה של HACK
8	10/5/2023	ביחד	<b>ולידציה + כתיבת תוכנה להריץ על המעבד</b>
		מנדל, אוריאל	לימוד על האסמבלר וסט הפקודות בשפת אסמבלי
		מנדל	ללמוד להשתמש באסמבלר קיים ואיך להריץ תוכניות
		אוריאל	כתיבת תוכנת אסמבלי להריץ על המחשב
48	1/7/2023	סה"כ	

## 7. ביבלוגרפיה:

- [1] [How to Choose a CPU - Newegg Insider](#)
- [2] [Von Neumann architecture - Wikipedia](#)
- [3] [Harvard architecture - Wikipedia](#)
- [4] [The Elements of Computing Systems: Building a Modern Computer from First Principles \(javier.io\)](#)
- [5] [basys3:basys3\\_rm.pdf \(digilent.com\)](#)