תכן לוגי
תרגיל רטוב #2

תאריך ההגשה נמצא באתר הקורס בלשונית "תרגילי בית".
המתרגל האחראי על התרגיל: משה ליכטנשטיין.
שאלותיכם במייל (smosesli@campus **ולא** cs@), (כולל עניינים מנהלתיים) יופנו רק אליו.
שאלות בעל-פה ייענו על ידי כל מתרגל.

הוראות הגשה:

- יש להגיש 3 קבצים, את שלושתם מגישים באתר הקורס.
- ההגשה בזוגות.
- שאלות ותשובות הנוגעות לתרגיל יפורסמו באתר הקורס תחת הלשונית "שאלות ותשובות".
- שמות הקבצים להגשה הינם :
  - forwarding_unit.v
  - forwarding_instance.v
  - forwarding_unit.pdf

תרגיל רטוב 2 – Forwarding unit

בתרגיל זה נממש Forwarding unit עבור Pipelined MIPS תוך שימוש בקוד Verilog. התרגיל אינו דורש כתיבת קוד רבה אלא הבנה של החומר. ודאו שאתם ניגשים לפתירת התרגיל לאחר שאתם סגורים על החומר הנלמד כדי לא לבזבז לעצמכם זמן יקר.

בקובץ הzip המצורף תמצאו מספר קבצי Verilog:

מודולים המתאימים ללוגיקה צירופית:

- adder
- ALU
- br_controller
- controller
- mux
- mux3
- sign_extension

מודולים סינכרוניים: (הערה: להסבר על Verilog Arrays קראו בקישור)
https://www.verilogpro.com/verilog-arrays-plain-simple/

- program_counter
- register_file (שימו לב לRF split)
- data_mem
- inst_mem

חמישה מודולים המתאימים לשלבי ה Pipe ומכילים בתוכם את המודולים הקודמים: (שמות המודולים מתאימים לשלבים st_{IF,ID,EX,MEM,WB})

- **IF** מכיל את הpc, זיכרון הפקודות, mux וadder. בהתאם המודול סינכרוני.
- **ID** מכיל את הבקר, sign_extension, בקר הקפיצות ו mux (אך לא את הRF). הבורר מחליט האם הארגומנט השני שיקבל הALU יהיה תוכן rt או ה immediate בהתאם למידע שיקבל מהבקר. שימו לב שבארכיטקטורה זו ההחלטה על קפיצה מתבצעת בשלב השני בהתאם לתוצאה מה br_controller.
- **EXE** מכיל את הALU ושלושה בוררים. שניים מתוכם מתאימים לשתי הארגומנטים של הALU ובורר נוסף עבור הערך שיעבור לכתיבה לזיכרון. ***תפקיד המודול forwarding_unit שתכתבו הוא לשלוט על שלושת הבוררים כלומר להחליט מה יהיו הארגומנטים שיקבל הALU ואיזה מידע ייכתב לזיכרון.***
- **MEM** מכיל את זיכרון הנתונים, מודול סינכרוני.
- **WB** מכיל בורר. תפקידו הוא להעביר לRF את תוצאת הקריאה מזיכרון הנתונים או את תוצאת חישוב הALU, ההחלטה תתקבל בהתאם לאותות ההוראה המגיעים מהבקר דרך שלבי הpipe.

ארבעה מודולים המתאימים לארבעת הרגיסטרים שבין שלבי ה Pipe. (שמות המודולים בהתאם reg_{IF_ID,ID_EX,EX_MEM,MEM_WB}) ארבעת הרגיסטרים מקבלים את שעון המערכת ותפקידם הוא להעביר את המידע הנצרך ואותות הוראה משלב לשלב.

המודול pipelined_MIPS מכיל את חמשת השלבים, ארבעת הרגיסטרים שביניהם, ואת ה RF, תפקידו לחבר בין כל הרכיבים שתוארו. גם המופע של המודול forwarding_unit שתכתבו יוכל בו.

בקובץ pipelined_MIPS_tb נמצא הtest-bench, הוא יוצר מופע של pipelined_MIPS והוא האחראי לייצר את אות השעון ואות ה reset עבור המחשב.

קבצים נוספים שנמצאים בתיקיה:

- קובץ defines - מכיל את הop-codes המתאימים ל8 הפקודות שנתמכות על ידי הארכיטקטורה { add, sub, addi, subi, beq, bne, lw, sw} ואת הקידוד המתאים ל2 הפעולות שנתמכות בALU (חיבור וחיסור).
- קובץ python ושני קבצי txt שיעזרו לכם לטעון קוד MIPS שתכתבו אל זיכרון הפקודות. כדי להריץ קוד על הpipelined_MIPS צריך כמובן לטעון את אותו קוד אל זיכרון הפקודות, הרצה של inst_gen.py תקרא פקודות MIPS שתכתבו ב inst.txt (פקודות מהצורה addi $3 $2 -4), תיצור מהפקודות האלו קידוד בינארי מתאים, ותכתוב אותם לזיכרון. קוד הpython ייצור מתוך 2 הקבצים inst.txt, inst_base.txt את המודול inst_mem.v – מודול הכתוב ב Verilog ומתאר זיכרון פקודות בו טעונות הפקודות שכתבתם.

מה צריך להגיש? 2 קבצי קוד וקובץ PDF אחד. שלושת הקבצים יוגשו כzip דרך האתר.

- forwarding_unit.v – בקובץ זה יש לממש את המודול המתאים ששמו forwarding_unit.
- forwarding_unit.pdf – בקובץ זה הסבירו בקצרה (לא יותר מעמוד אחד) על המימוש שלכם, ניתן להוסיף איור לפי בחירתכם. בפרט, הסבירו איזה כניסות בחרתם למודול ואת הלוגיקה שעומדת מאחורי בחירה זו.
- forwarding_instance.v - במקום להגיש את כל הקובץ pipelined_MIPS.v שמכיל מופע של המודול forwarding_unit שכתבתם, עליכם להגיש רק את שורות הקוד החדשות שהוספתם לקובץ pipelined_MIPS.v בקובץ זה.
הסבר נוסף: אם תסתכלו בקובץ pipelined_MIPS.v תוכלו לראות את השורות הבאות

```
//TODO: Implement the forwarding_unit module

// Write your code (forwarding_unit instantiation, not the module) here

assign {sel_alu1, sel_alu2, sel_store_val} = 0;

// end TODO
```

כל הקוד שעליכם לכתוב / לשנות בקובץ pipelined_MIPS צריך להופיע במקום השורה שבין ההערות (כלומר, במקום פקודת ה-assign). לאחר שסיימתם לכתוב את הקוד ולהריץ את הטסטים שלכם, העתיקו את הקוד שלכם לתוך הקובץ forwarding_instance.v.

נספח: התקנת פייתון

Windows:

1. התקינו את anaconda 3.7 מהקישור הבא: https://www.anaconda.com/download/
2. לאחר ההתקנה חפשו את anaconda prompt תחת anaconda3 בתפריט התחל.
3. בחלון שנפתח, עברו לתיקייה שבה מופיע קוד הפייתון (בעזרת cd)
4. רשמו את הפקודה הבאה: python inst_gen.py

Linux:

https://docs.python-guide.org/starting/install3/linux/

Mac:

https://docs.python-guide.org/starting/install3/osx/