

## תרגיל בית רטוב מספר 3

מגישים:

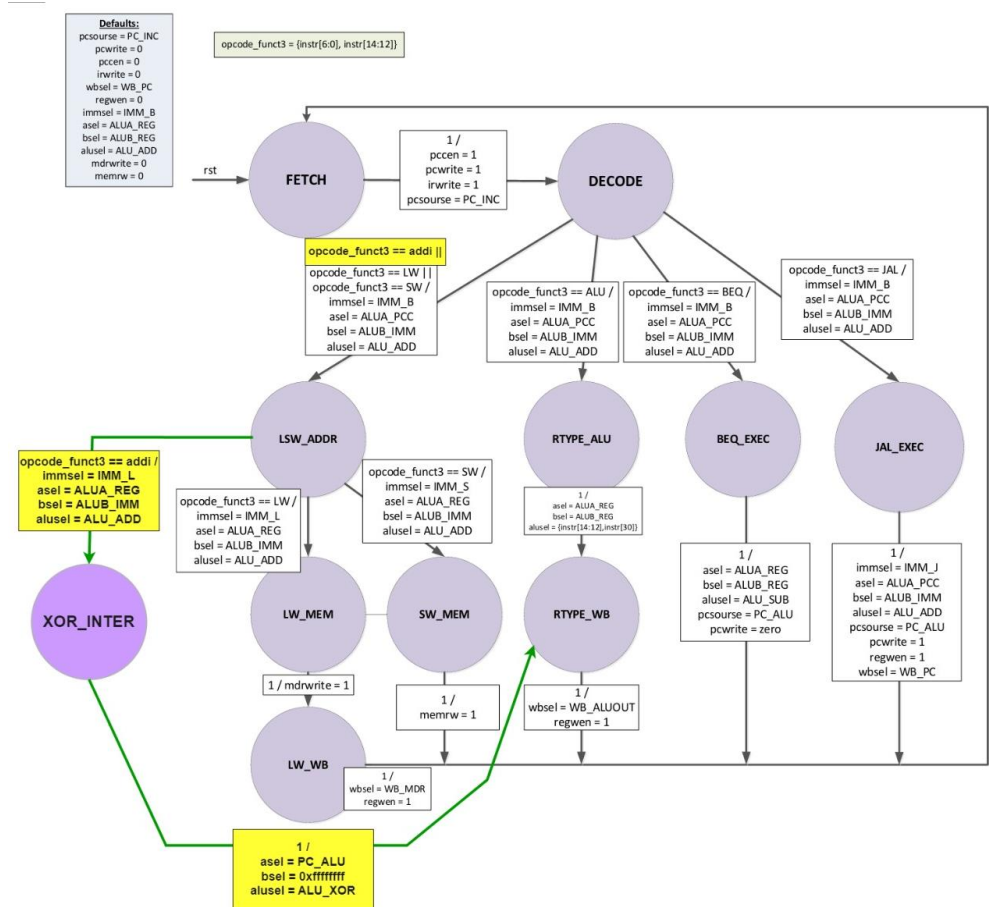
322654617	ינון יהב
327812483	נועם בן שמעון

22/08/2024

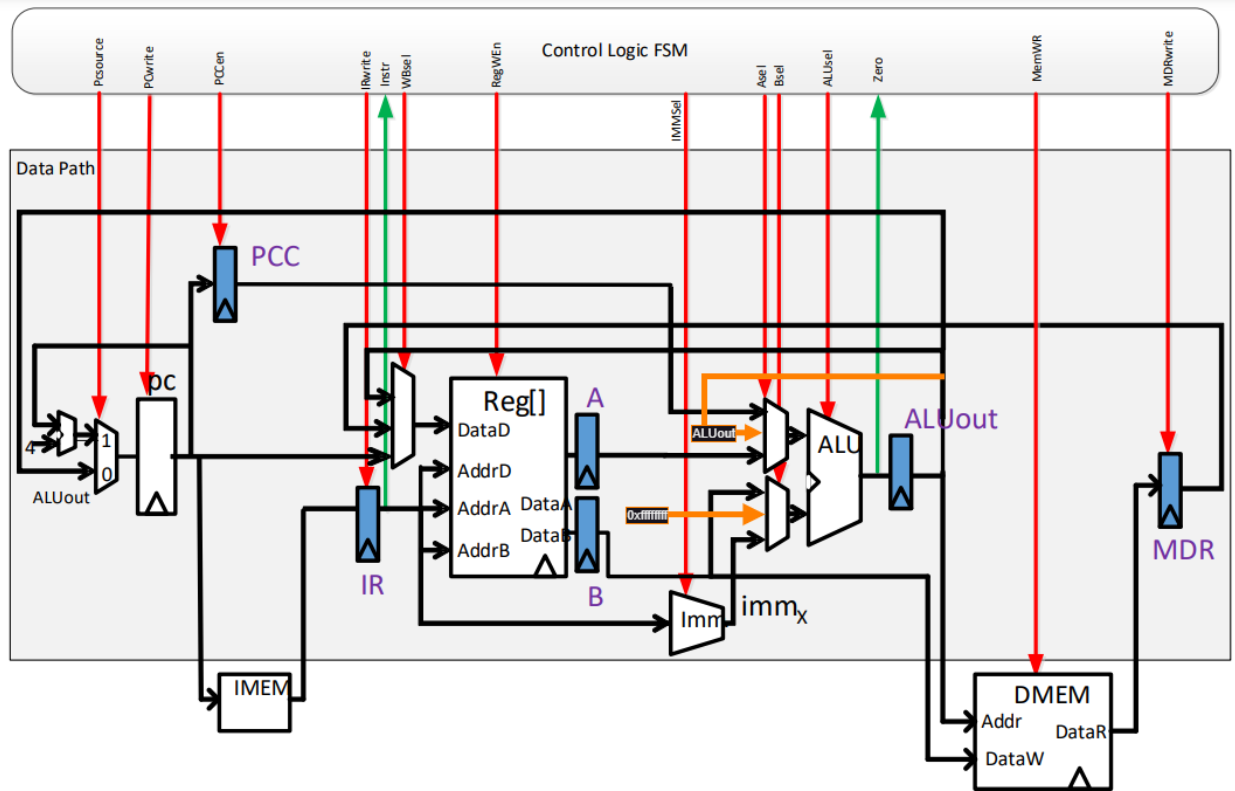
תאריך:

322654617	ינון יהב
327812483	נועם בן שמעון

## עדכון מכונת המצבים של ה-Controller לפקודה החדשה:



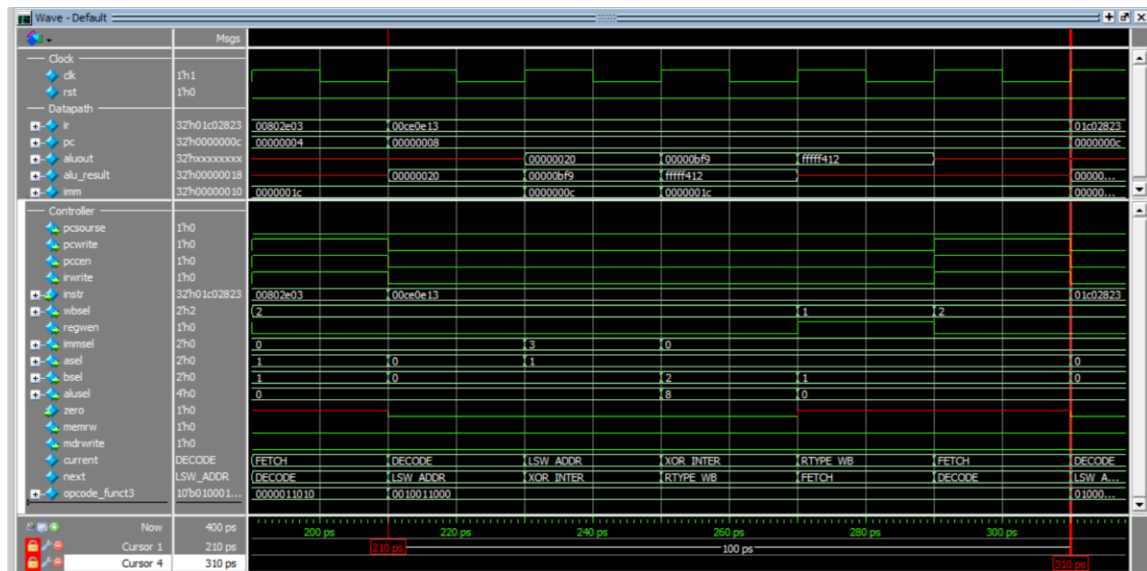
## עדכון ה-Datapath לפקודה החדשה:



אנו עשינו שני שינויים בתרשים כדי לממש את הפקודה החדשה.

רצינו שנוכל לבצע שני חישובים (שבהכרח יקרו בנפרד) כאשר השני ( $0xffffffff$ ) תלוי בראשון ( $+imm$ ). על מנת לאפשר זאת הוספנו כניסה שלישית ל- $mux$  שבחר לנו את הקלט  $A$  של ה- $ALU$ , ולכניסה זו חיברנו את הערך של  $ALUout$ , כך שאחרי החיבור הראשון הערך יועבר חזרה, ייבחר על ידי ה- $mux$ , ותבוצע עליו הפעולה השנייה. בנוסף, על מנת שנוכל לבצע את הפעולה השנייה, הוספנו גם ב- $mux$  שבחר לנו את הקלט  $B$  של ה- $ALU$  כניסה שלישית, וחיברנו אליה את הערך הקבוע  $0xffffffff$  שאיתו נרצה לעשות  $xor$  בחישוב השני.

## תרשים גלים:



אנו יכולים להראות את ה-*opcode\_func3* משתנה ב- $210 [ps]$  להיות הערך התואם לפקודה החדשה, לכן נסתכל על התרשים רק החל מנקודה זו.

ניתן לראות כי אנו נמצאים ב-*DECODE* עם שינוי ה-*opcode\_func3*, כרצוי. אז משום ש-*opcode\_func3* שווה ל-*addi* אנו שמים בתור המצב הבא את *LSW\_ADDR*, כרצוי.

ב-*LSW\_ADDR* אנו שמים את המצב הבא להיות *XOR\_INTER*. אנו מבצעים גם את חישוב הסכום של מה שבכתובת 8 בזיכרון ושל המספר הנתון 12 ( $(c)_{16} =$ ). אם נסתכל בזיכרון נראה שבכתובת 8 יש לנו את הערך  $(bed)_{16}$  ולכן הסכום שלהם הוא  $(bf9)_{16}$ . ניתן לראות כי התוצאה של ה-*ALU* מתאימה לערך זה.

לאחר מכן ב-*XOR\_INTER* אנו שמים את המצב הבא להיות *RTYPE\_WB*. אנו מבצעים בשלב זה *xor* עם הערך  $0xffffffff$ . התוצאה צריכה להיות:  $(fffff406)_{16}$ . ניתן לראות כי התוצאה שמתקבלת זהה.

ולסיום אנו ב-*RTYPE\_WB* שומרים את התוצאה ברגיסטר.