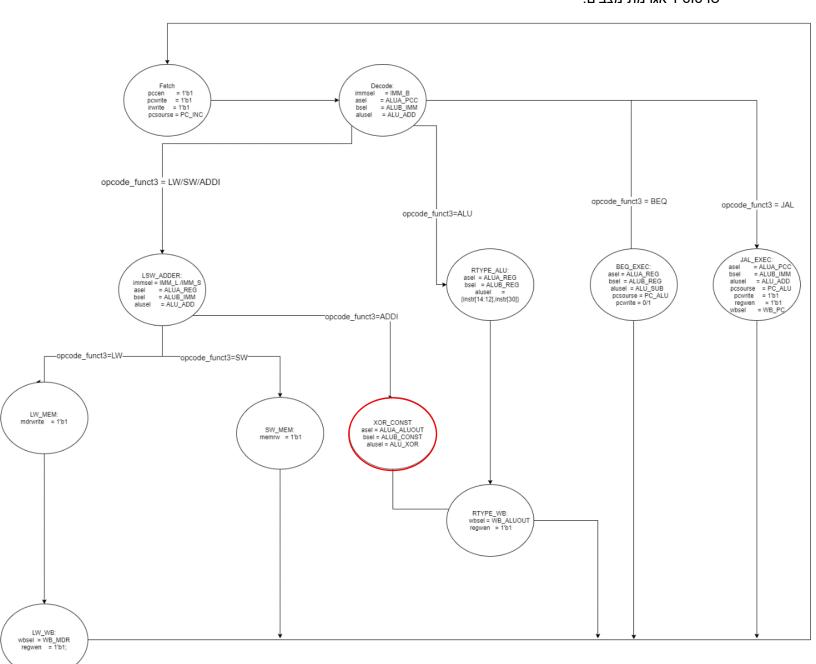
314722950	אופיר אלדר
209086578	בן הלפרין

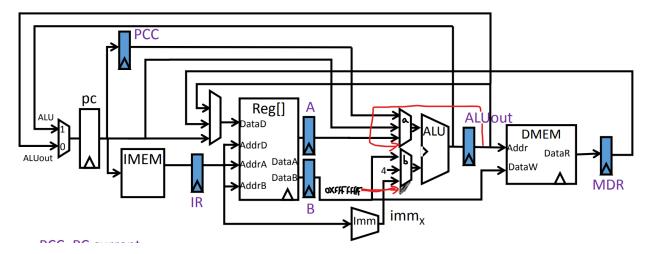
## סימולציה 3 חלק יבש

## 2.1: שרטוט דיאגרמת מצבים:



ספי שניתן לראות, הוספת התמיכה בפקודה החדשה דרשה שינוי מינימלי במכונת המצבים, כאשר המצב היחיד שנוסף הוא XOR\_CONST. מהלך ביצוע הפקודה הוא כזה: לאחר שלבי ה fetch וה ACOR\_CONST. מהלך ביצוע הפקודה הוא כזה: לאחר שלבי ה sw ו ווא בלבד. תפקידו הוא לחשב כתובת פקודה, נעבור למצב LSW\_ADDER. עד עכשיו מצב זה שימש לפקודות liw בלבד. תפקידו הוא לחשב כתובת יעד באמצעות חיבור בין כתובת הבסיס השמורה ברגיסטר rs1 לבין rs1 לבין cmmediate לעומת כתובת הבסיס. זוהי בדיוק הפעולה הראשונה שמבוצעת ע"י iaddi: חיבור בין תוכן rs1 לבין emmediate לכן פקודת immediate תשתמש במצב הזה. נשים לב כי נדרש שהמצב יתמוך בקריאת ה immediate בפורמט בו הוא כלול בפקודה w, ולכן נקבע XOR\_IMME = IMM\_L לאחר מכן נדרש לבצע פעולת ROR בין תוצאת החיבור לבין קבוע. פעולה זו לא התרחשה עד כה באף אחת מהפקודות הקיימות ולכן דרשה יצירת מצב חדש, שנקרא XOR\_CONST. מצב זה דורש מהLDO לבצע בהכרח פעולת ROR, בין הרגיסטר ALUOUT בו נמצאת כרגע פעולת החיבור מהמצב הקודם, לבין הקבוע Oxffffffff שהתווסף ככניסה שלאות בשרטוט המעבד. השלב האחרון שנדרש בביצוע הפקודה הוא כתיבת תוצאת הROR לרגיסטר rd לכן נעבור מתוך XOR\_CONST ל R-TYPE\_WB ששימושו המקורי היה כתיבת תוצאות פעולות rd לתוך rd. לכן נעבור מתוך XOR\_CONST ל RTYPE\_WB ומשם חזרה ל Fetch, מה שמציין את סיום ביצוע הפקודה.

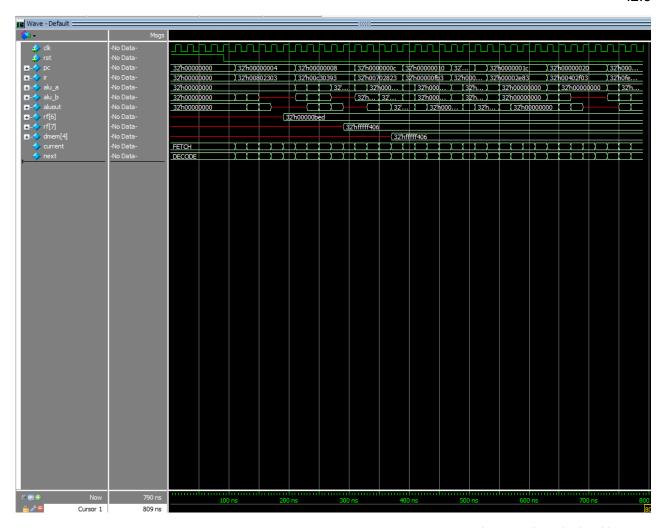
## :שרטוט מעבד



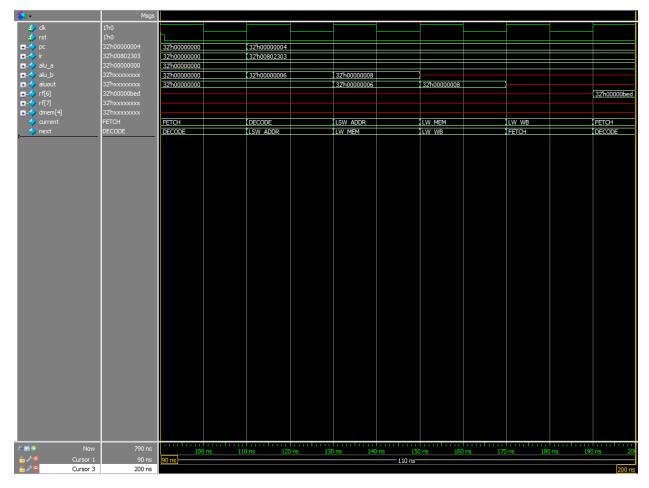
התוספות שלנו מסומנות באדום.

הוספנו לmuxa חיבור לתוצאת הalu על מנת לתמוך בשימוש בתוצאת החיבור בפעולת ה xor שבאה לאחר מכן. (ולכן לסלקטור של הmuxn הזה התווסף ביט נוסף).

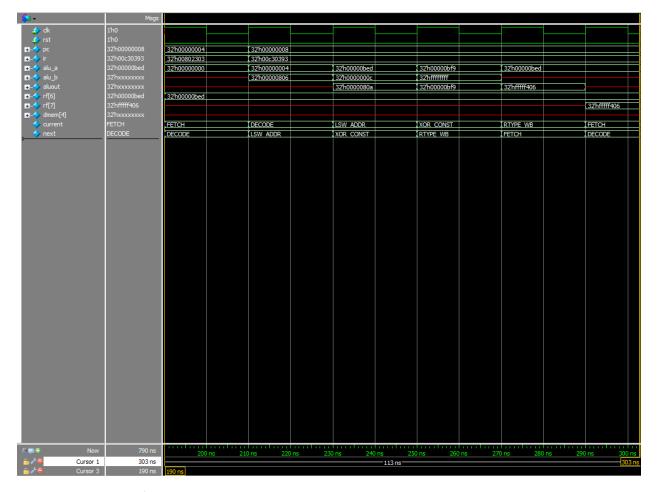
הוספנו לmuxb חיבור לקבוע 0xffffffff, על מנת ביצוע פעולת הxor איתו. (ולכן גם לסלקטור של הmux הזה התווסף ביט נוסף).



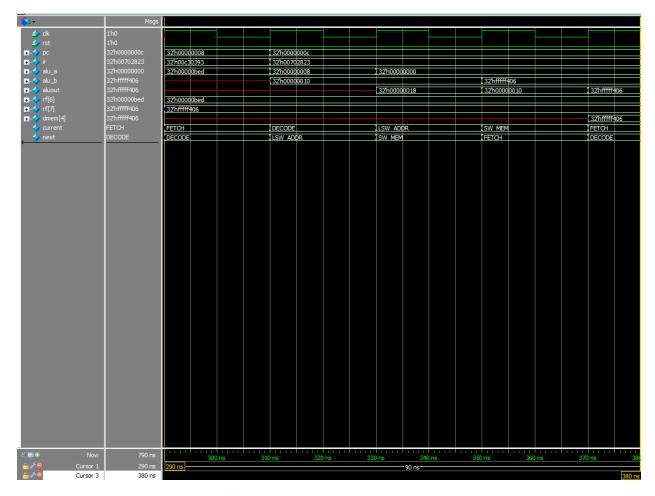
מבט כללי על כל מהלך הסימולציה.



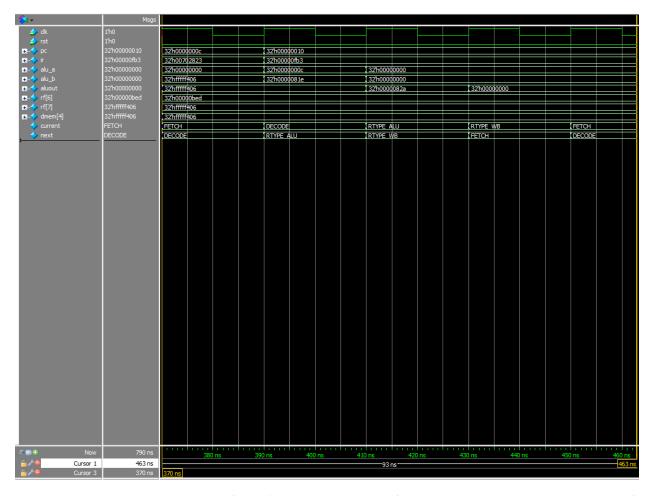
הפקודה הראשונה בtest היא פקודת lw שטוענת את התוכן של הכתובת 8 בזיכרון לרגיסטר tt שהוא הרגיסטר באינדקס 6. ניתן לראות כי מצבי המכונה במהלך ביצוע הפקודה תואמים לחמשת המצבים של LW בדיאגרמת מכונת המצבים. בנוסף, לאחר שלב LSW\_ADDER הכתובת ב aluout ממנה ייטען הזיכרון היא אכן 8, ולאחר שלב LW\_WB האחרון, ב[6] שהוא t1 מופיע הערך הדרוש.



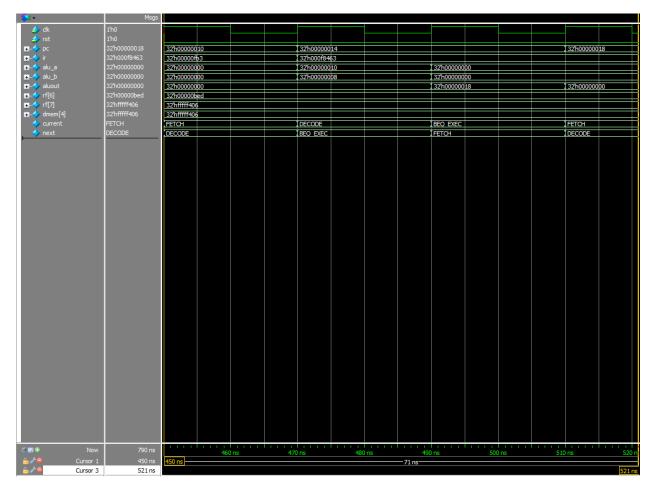
הפקודה השנייה היא פקודת addi שהוספנו. מצבי המכונה תואמים את חמשת המצבים של addi בדיאגרמת מכונת המצבים, כולל שימוש במצב XOR\_CONST החדש. הפקודה מוסיפה לערך בt1 את הקבוע 12: ניתן לראות מכונת המצבים, כולל שימוש במצב החיבור, הערך של alu\_b הוא הערך שנטען קודם לתוך t1, והערך של alu\_b הוא שבשלב LSW\_ADDR בו מתבצע החיבור, הערך של alu\_a הוא שלב ביצוע הxor. כעת alu\_a מתעדכן 12. הערך של aluout מתעדכן לתוצאת החיבור הנכונה בשלב הבא, שהוא שלב ביצוע הoxffffffff בשלב הכתיבה להיות תוצאת החיבור שהייתה בaluout, בעוד שalu\_b מתעדכן להיות הערך הקבוע oxffffffff. בשלב הכתיבה לרגיסטר כבר ניתן לראות את תוצאת הפעולה בaluout, ולאחר עליית שעון ניתן לראות שהיא נכתבה ל[7], כלומר לt2.



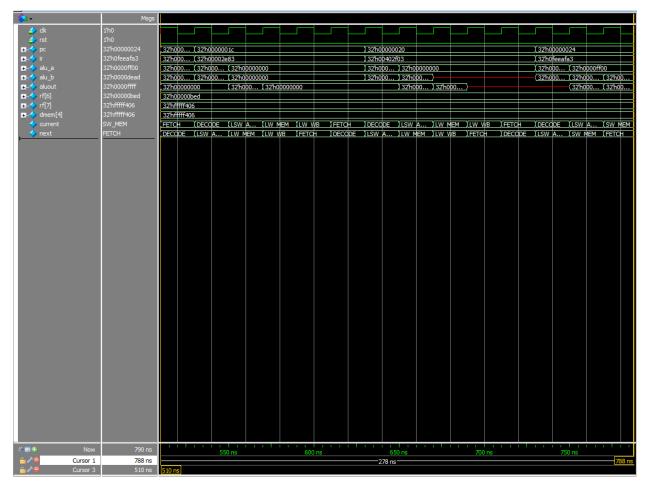
הפקודה הבאה היא פקודת sw ששומרת את הערך שנמצא בt2 בכתובת 16 בזיכרון. פקודת sw קצרה יותר מהקודמות ועוברת דרך 4 מצבים, כלומר אורכת 4 מחזורי שעון. לאחר שלב LSW\_ADDR שמחשב את הכתובת בזיכרון בה יישמר הערך ברגיסטר, ניתן לראות כי הערך השמור בaluout הוא 10 בבסיס הקסדצימלי, או 16 בבסיס דצימלי כנדרש. לאחר שלב SW\_MEM בו מתבצעת הכתיבה לזיכרון, הערך בזיכרון נהיה התוצאה שהייתה שמורה בt2.



לאחר מכן מתבצעת פקודת add שמכניסה לרגיסטר t6 את הערך 0. בשלב RTYPE\_ALU בו מתבצע החיבור ניתן מלאחר מכן aluout וגם aluout מקבלים את הערך 0, וזה גם ערך תוצאת החיבור בaluout לראות כי גם alu\_a



הפקודה הבאה מבצעת השוואה בין t6 ל0 – בפקודה הקודמת קבענו את ערכו של t6 ל0 ולכן תתקיים קפיצה pc הכתובת BEQ\_EXEC שמציין את הכתובת 18 שחושבה כבר בשלב הDECODE. לאחר סיום BEQ\_EXEC הכתובת בסת מתעדכנת להיות הכתובת שחושבה.



סוף הסימולציה מבצע 3 פעולות lw,lw,sw. מטרתן לטעון לt4 את הערך 0 ששמור בx0, לטעון לt5 את הערך ששמור בx0, לטעון לt5 את הערך ששמור בכתובת 4, ולשמור את הערך שנטען לt5 בכתובת FF בבסיס הקסדצימלי שהיא 255 בבסיס דצימלי. לאחר מכן התוכנית נכנסת ללולאה אינסופית כיוון שהיא ממשיכה לבדוק האם t6 מכיל את הערך 0 בלי לשנות אותו במהלך הלולאה, אבל הסימולציה מפסיקה באופן אוטומטי. הפקודה lw כוללת 5 מצבים בעוד ש sw כוללת 4 מצבים, כלומר בסך הכל שלב זה של התכנית כולל 14 מצבים כלומר אורך 14 מחזורי שעון, כפי שניתן לראות בסימולציה.