



Lab3 preparation report

Digital System Design

Multi-Cycle CPU Design

הפקולטה למדעי ההנדסה –
מעבדת ארכיטקטורת מעבדים מתקדמת ומאיצי חומרה

עבור: רבוא חנניה

מגישים:

אורן שור – 316365352

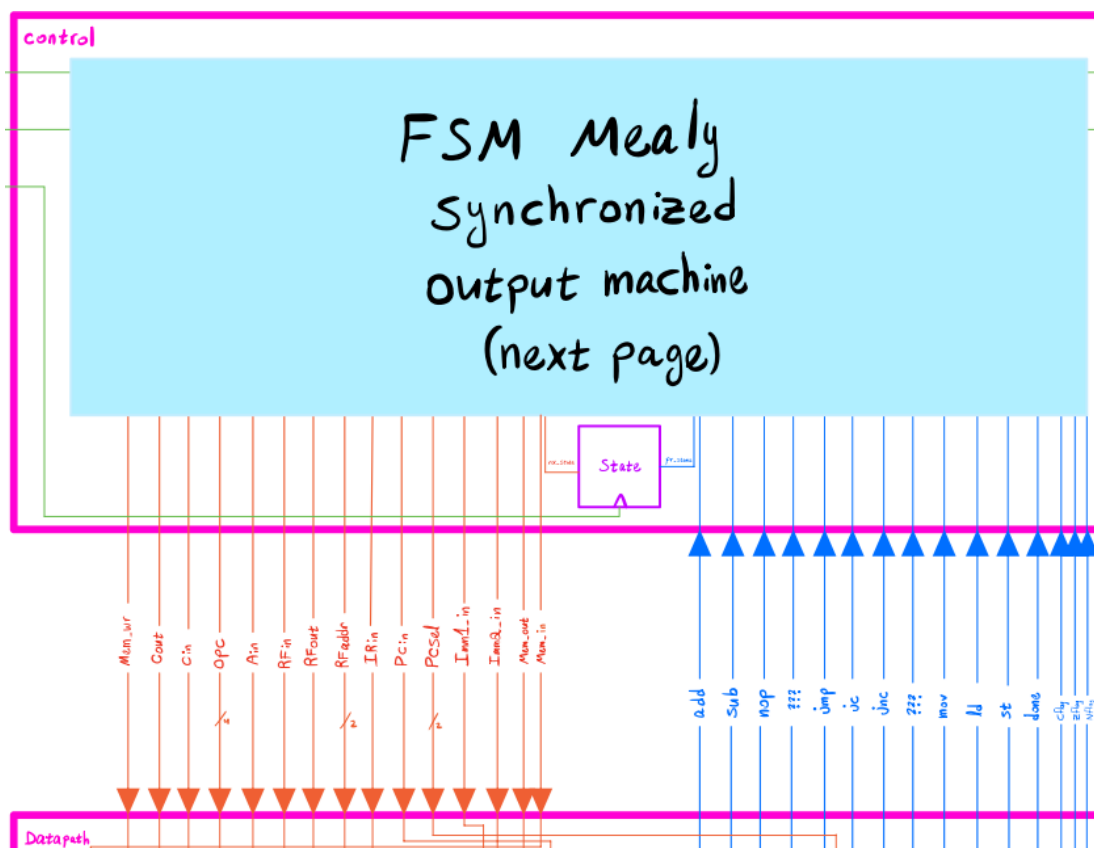
טל שוורצברג – 316581537



הסבר על המערכת

1. Control Unit

מודול זה מממש מכונת מצבים מסוג Mealy, כאשר כל מצב נוכחי (וקוי הבקרה האדומים הנשלחים ל-Datapath) נקבעים לפי פונקציה של המצב הקודם וקוי הבקרה הכחולים



אל מודול זה נכנסים בנוסף קוי בקרה מה-Testbench (ירוק), לצורכי סימולציה, וסיגנל שעון חיצוני.

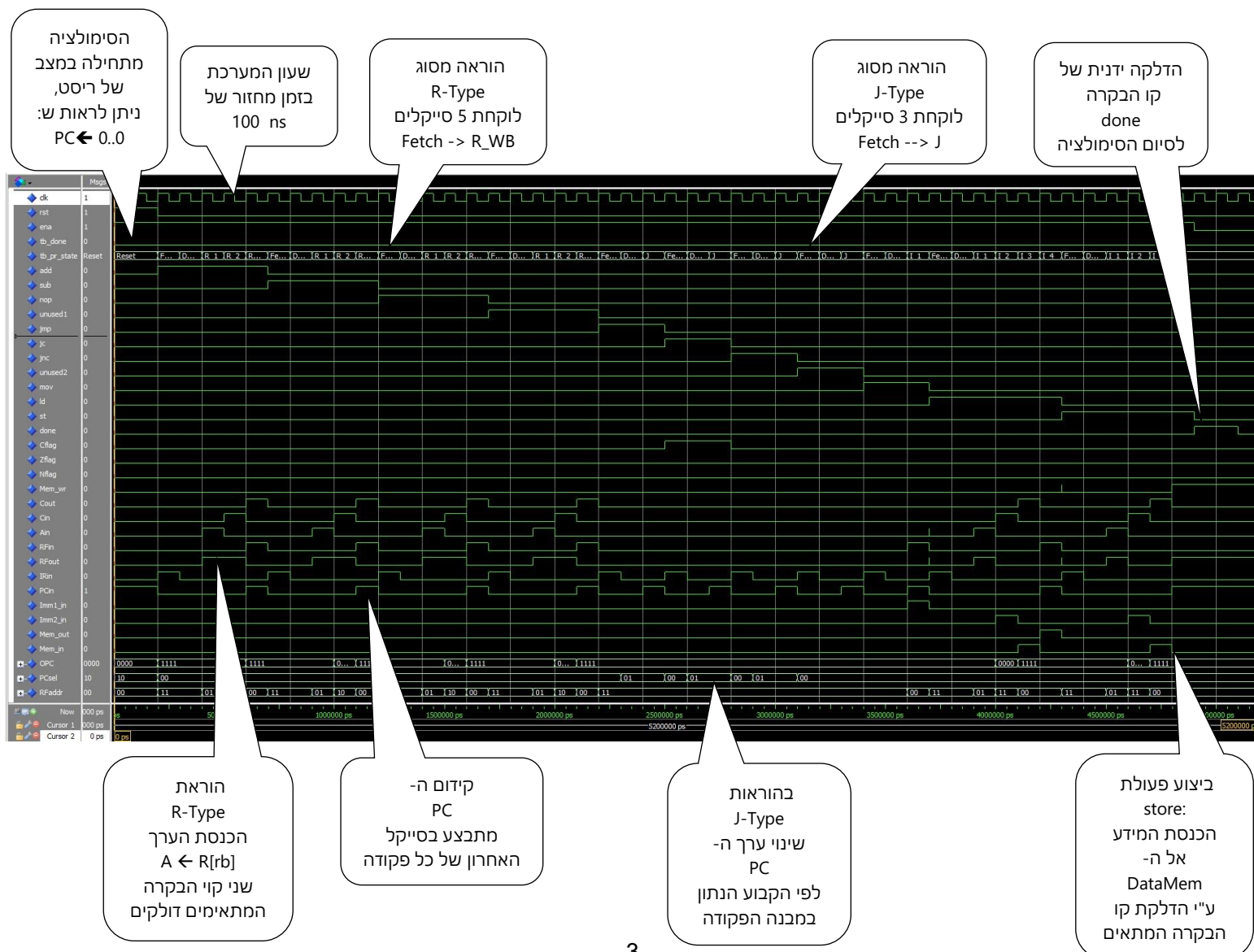


Control Unit - סימולציה:

בסימולציה הנ"ל הוגדרו מספר תהליכים –

- Gen_clk – תהליך המייצר שעון מערכת.
- StartTb – תהליך המדמה רצף פקודות לביצוע, זו אחר זו ולפי הסדר (משמאל לימין):
add, sub, nop, unused1, jmp, jc, jnc, unused2, mov, ld, st, done
בתהליך זה קוי הבקרה (הכחולים) מופעלים ידנית ומדמים את אופן פעולת ה-Datapath. כל קוי בקרה פועל לפרק זמן התואם את מספר מחזורי השעון של הפעולה המתאימה לו.

להלן תוצאות הסימולציה, המתוארות על פני תרשים ה-wave :

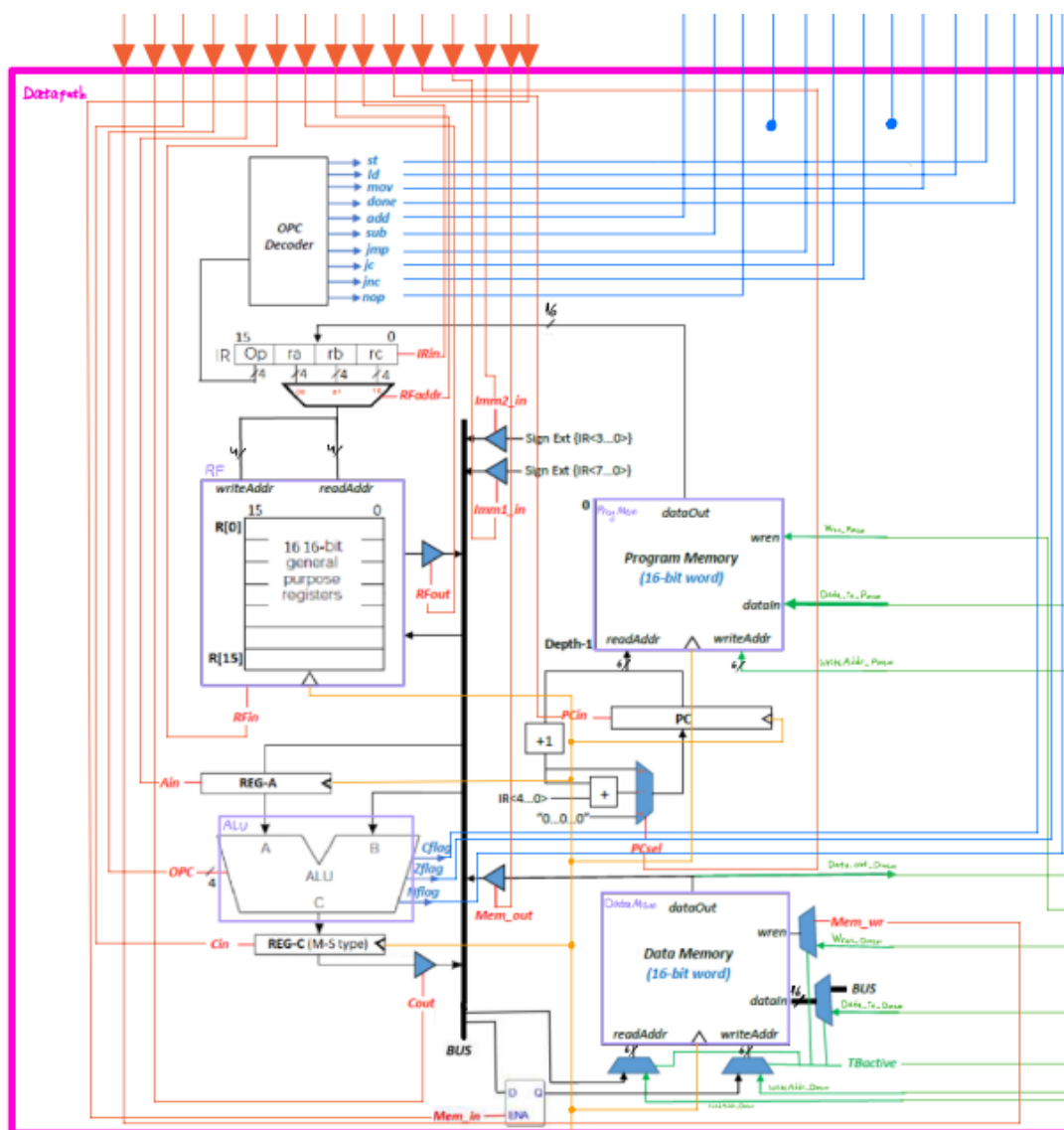


- Datapath .2

מזדול זה אחראי על מימוש המערכת, וכולל את האינטרקציה של כלל המודולים ב-CPU, כגון ALU, רכיבי זיכרון, באס משותף ועוד (מתוארים בהרחבה בהמשך).

אל מודול זה נכנסים קוי בקרה (באדום) המגיעים מה-Control Unit שתיארנו קודם. במודול זה ישנם רכיבים אסינכרוניים (מעגלים צירופיים, Mux), ורכיבים סינכרוניים המתוזמנים לעבודה בעליית שעון (רגיסטרים, זכרונות).
בכל שלב נתון, ובהינתן סטטוס קוי בקרה, מתבצעת פעולה אחת ביחידת ה-datapath, כחלק מסדרת פעולות בארכיטקטורת ה-Multi-Cycle.

ביחידה זו קיים מפענח (Decoder) אשר מפיק אותות בקרה (כחולים) ע"פ שדה ה- Opcode של הפקודה הנוכחית לביצוע. אותם קוי הבקרה יוצאים ממודול זה ונכנסים למודול ה- control שתיארנו קודם.



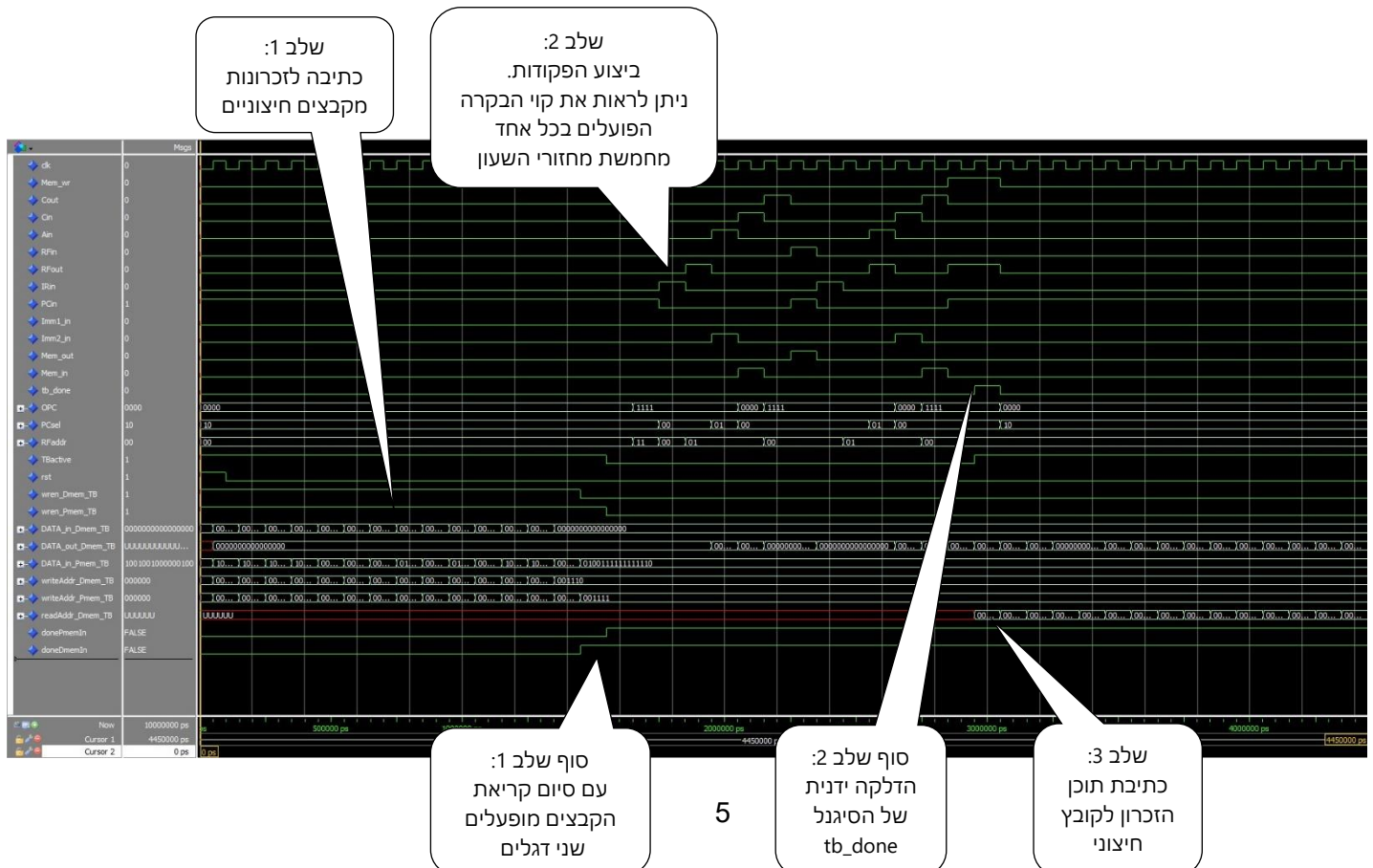


Datapath - סימולציה:

בסימולציה הנ"ל הוגדרו מספר תהליכים –

- Gen_clk – תהליך המייצר שעון מערכת.
- Gen_rst – תהליך המייצר אות אתחול למערכת.
- Gen_TB – תהליך המשנה את קו הבקרה Tbactive להתאם לשלב בתכנית (קריאה מקבצים, עבודה, כתיבה לקובץ)
- LoadDataMem – תהליך קריאת מידע מקובץ חיצוני (DTCMinit) וכתיבתו בזכרון הפנימי DataMem.
- LoadProgramMem – תהליך קריאת מידע מקובץ חיצוני (ITCMinit) וכתיבתו בזכרון הפנימי ProgMem.
- StartTb – תהליך המדמה שתי פקודות בלבד לביצוע: Load → Store
זו אחר זו (התהליך טוען את הקבוע '4' לרגיסטר R2 ולאחר מכן מאחסן אותו בזכרון בתא מספר 3 (יופיע בשורה רביעית בקובץ DTCMcontent)
בתהליך זה קוי הבקרה (האדומים) מופעלים ידנית ומדמים את אופן פעולת הControl Unit.
- WriteToDataMem – תהליך קריאת מידע מה-DataMem וכתיבתו בקובץ חיצוני (DTCMcontent)
בסימולציה זו אנו טוענים את זכרונות ה-CPU במידע ובפקודות לביצוע, מבצעים את הפעולות הנדרשות, ולבסוף כותבים את תוכן הזכרון לקובץ חיצוני.

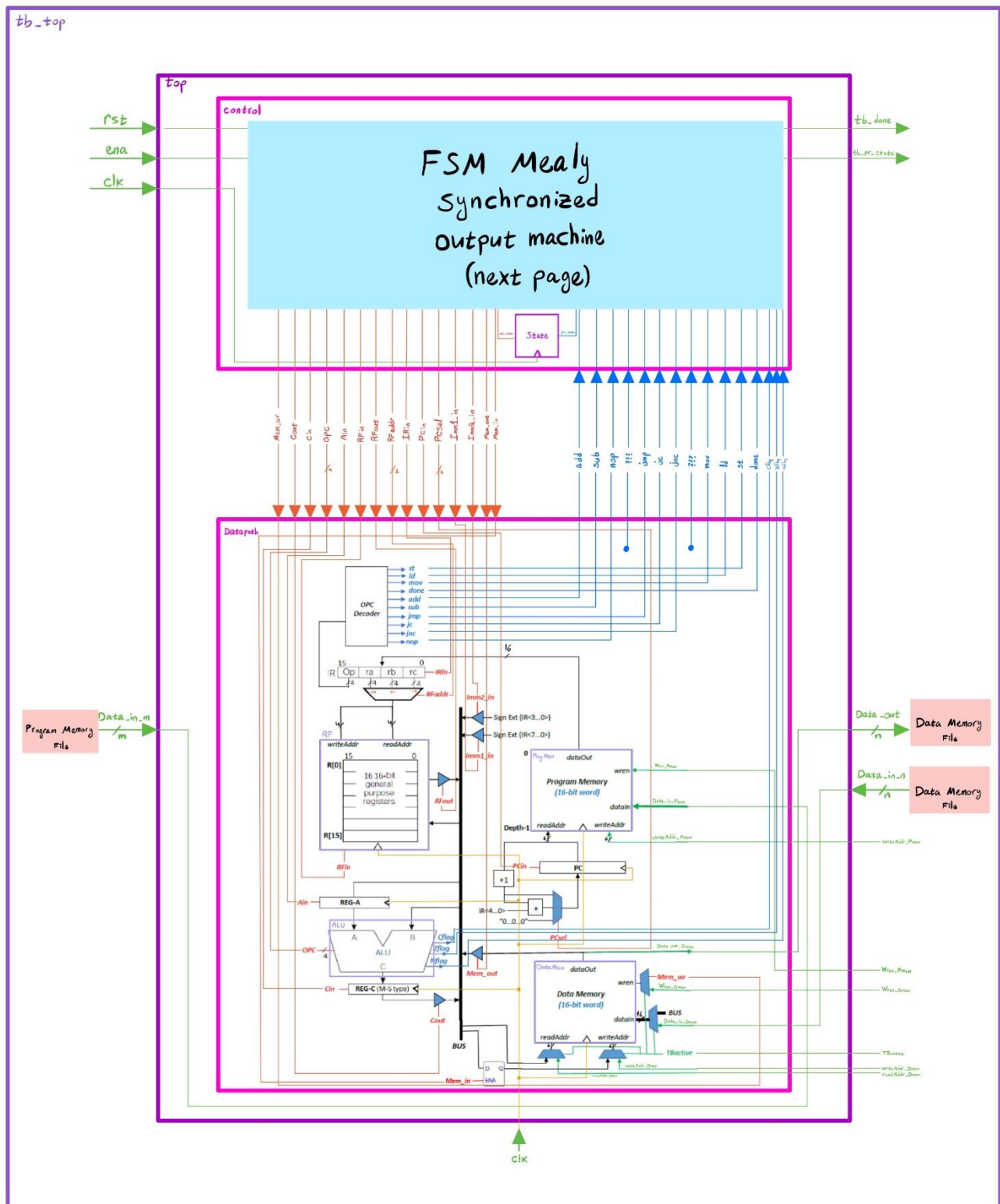
להלן תוצאות הסימולציה, המתוארות על פני תרשים ה-wave:



- Top .3

מודול זה מכיל בתוכו את שני המודולים שתיארנו לעיל, ואחראי על הקישור ושיתוף הפעולה ביניהם.

להלן איור המפרט את המודול המלא:





Top - סימולציה:

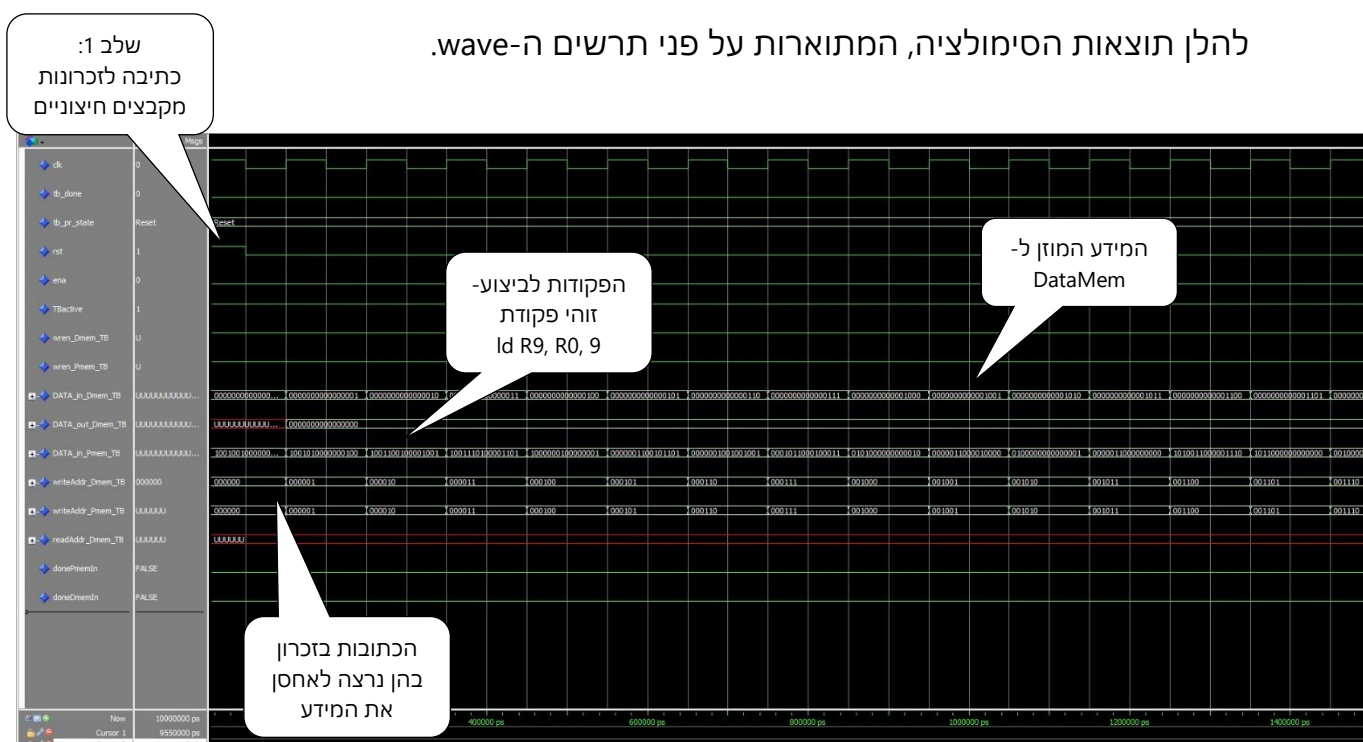
בסימולציה הנ"ל הוגדרו מספר תהליכים –

- Gen_clk – תהליך המייצר שעון מערכת.
- Gen_rst – תהליך המייצר אות אתחול למערכת.
- Gen_TB – תהליך המשנה את קו הבקרה Tbactive בהתאם לשלב בתכנית (קריאה מקבצים, עבודה, כתיבה לקובץ)
- LoadDataMem – תהליך קריאת מידע מקובץ חיצוני (DTCMinit) וכתיבתו בזכרון הפנימי DataMem.
- LoadProgramMem – תהליך קריאת מידע מקובץ חיצוני (ITCMinit) וכתיבתו בזכרון הפנימי ProgMem.
- WriteToDataMem – תהליך קריאת מידע מה-DataMem וכתיבתו בקובץ חיצוני (DTCMcontent)

- בסיום שני תהליכי הקריאה מקבצים חיצוניים, יופעל אות ה-En ובכך תתחיל פעולת ה-CPU עבור רצף הפקודות שהוגדרו. עם פקודת ה-tb_done ובסיום פעולת ה-CPU יחל שלב הכתיבה לקובץ החיצוני.

- ה-CPU מבצע את התכנית שפורסמה בקובץ המטלה, אשר משווה בין ערכם של שני רגיסטרים ובהתאם לכך משנה את תוכנו של תא מספר 14 (יופיע בשורה האחרונה בקובץ ה-DTCMcontent).

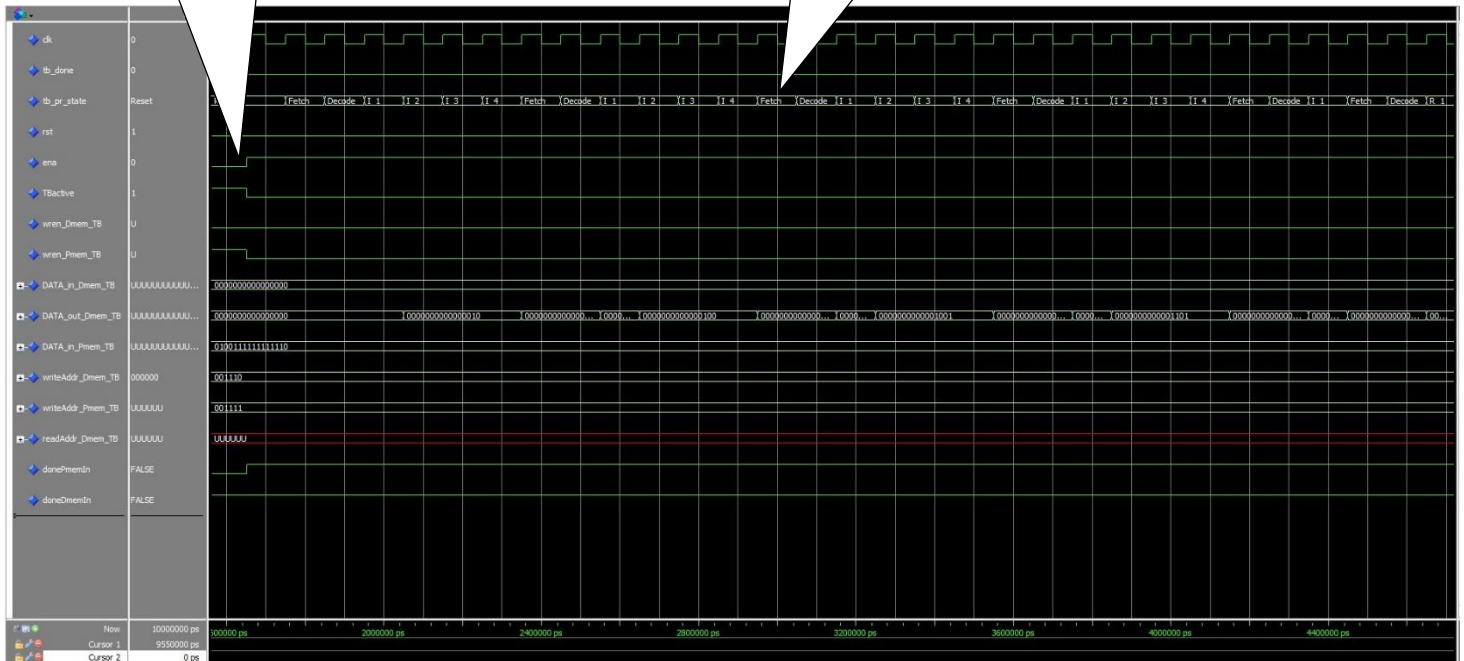
להלן תוצאות הסימולציה, המתוארות על פני תרשים ה-wave.



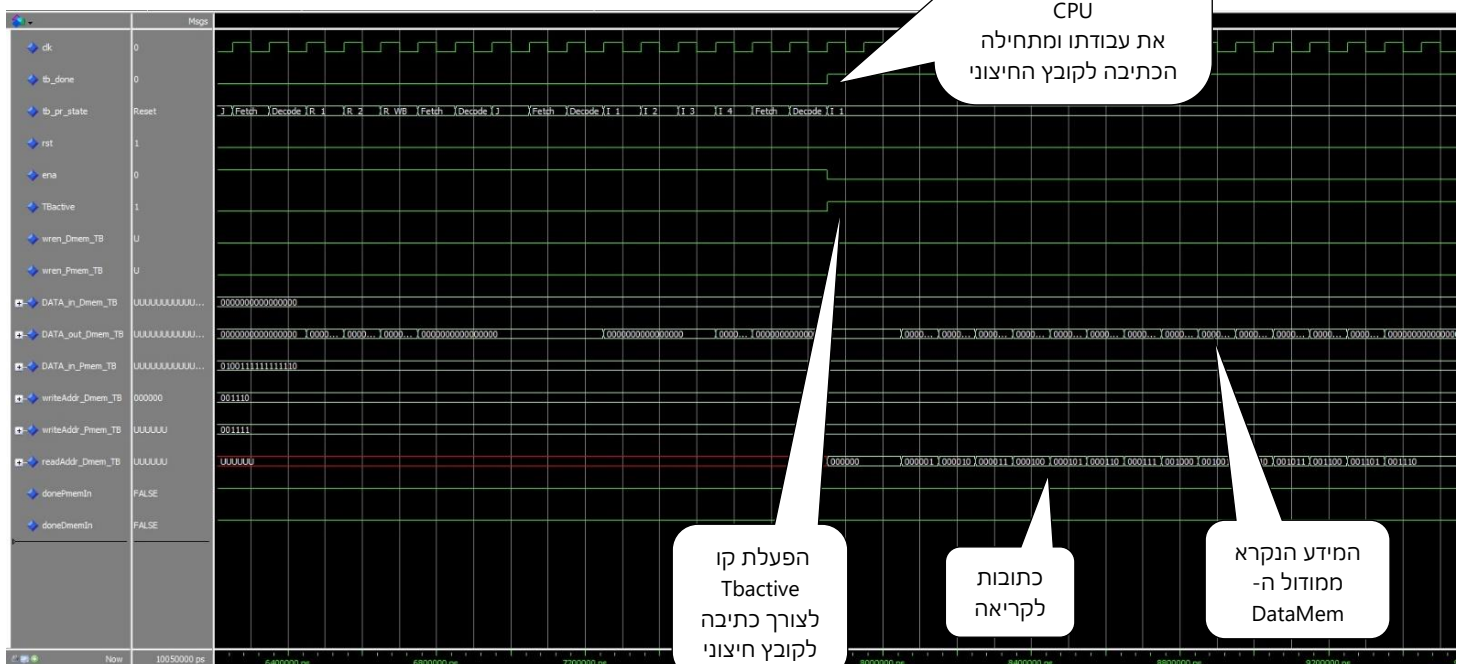


שלב 2 -
בעת הפעלת קו ה-
en
מתחילות להתבצע הפקודות

ביצוע 5 הסייקלים של
פקודת
load
שתוארה בעמוד הקודם



שלב 3-
עם עליית קו
tb_done
מסיים ה-
CPU
את עבודתו ומתחילה
הכתיבה לקובץ החיצוני



הפעלת קו
Tbactive
לצורך כתיבה
לקובץ חיצוני

כתובות
לקריאה

המידע הנקרא
ממודול ה-
DataMem



4. רכיבים נוספים:

כפי שציינו קודם, בתוך מודול ה-Datapath מופיעים מספר מודולים נוספים האחראיים על ביצוע הפעולות הנדרשות ב-CPU:

:ALU/FA

מודול זה מסייע לביצוע פעולות אריטמטיות ולוגיות בין שתי כניסותיו: A,B, למוצאו C. המודול ממומש באמצעות מחבר/מחסר אדווה (Carry Ripple Adder/sub) המורכב משרשור רכיבי FA.

:DataMem/ProgMem

מודולים אלו משמשים כרכיבי זיכרון RAM, ע"י מתן כתובת באורך 6 ביטים (מרחב כתובות של 2^6), כאשר בכל כתובת מאוחסן מידע באורך 16 ביטים. מודול אחד משמש לאחסון מידע הקשור לתכנית, בעוד השני מאחסן את סט הפקודות לביצוע.

:RF

בדומה לזכרונות ה-RAM, מודול זה מממש קובץ רגיסטרים בעל 16 רגיסטרים המכילים כל אחד 16 ביטים של מידע.

:BidirPin

מודול זה מדמה BUS דו כיווני, אשר מתחבר לכלל המודולים השונים המתוארים לעיל. על מנת למנוע מצב של Over driven, ועל מנת לשמור על הקוד סינטזבילי, קיים צורך במודל זה הכולל בתוכו רכיבי Latch, קוי אפשור, קוי כניסה ויציאה.

