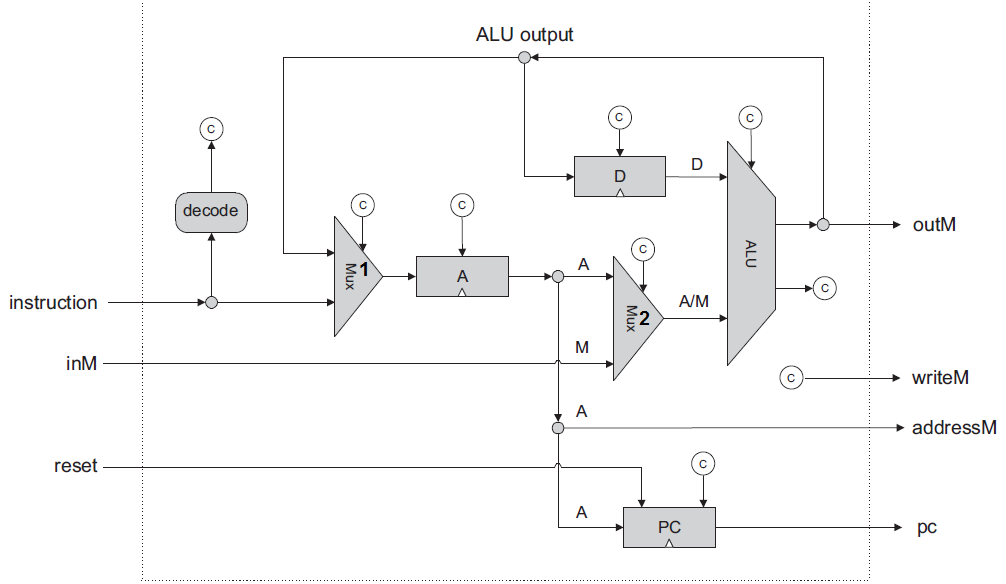
מבוא למערכות מחשוב – עבודה מספר 2.2

כללי – בעבודה זאת תרכיבו מהרכיבים אותם מימשתם בעבודה הקודמת מכונת פון נוימן מתפקדת. העבודה קשה ומאתגרת, ורצוי להתחיל בה בהקדם האפשרי.

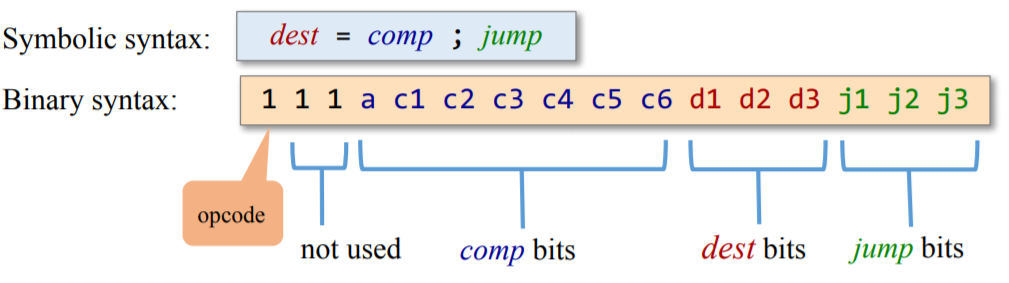
מימוש מכונת פון נוימן – אתם תממשו מכונת פון נוימן כפי שהוצגה בכיתה. המכונה משתמשת ברכיבים דומים לאלו שמימשתם בתרגיל הראשון, אך למטרת שיפור יעילות הקוד, אנו מספקים לכם מימוש אלטרנטיבי מהיר יותר בחבילה SimpleComponents. כפי שתוכלו לראות, הפרויקט Machine בו תשתמשו בסעיף זה כבר מכיל reference לפרויקט SimpleComponents, וניתן להשתמש בכל המחלקות מפרויקט זה. למעשה, מרבית רכיבי המכונה כבר ממומשים עבורכם, ומחוברים כיאות:

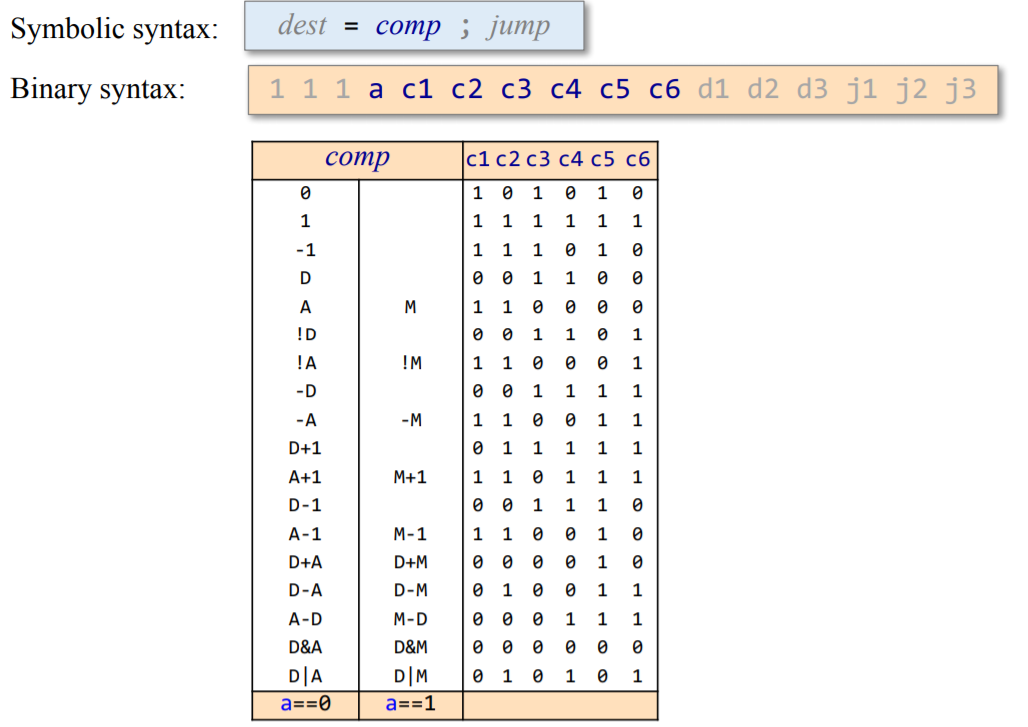
1. Machine16 - המחלקה המאחדת את כל הרכיבים למכונת פון נוימן. למכונה יש שני זכרונות, אחד עבור הקוד ואחד עבור המידע, מקלדת ומסך. החיבורים בין רכיבי המכונה כבר ממומשים.
2. Memory – ממומש בפרויקט SimpleComponents. לזכרון נוספה פונקציונליות נוספת המאפשרת לכתוב ישירות לזכרון (למשל machine.Data[0] = 100), מאחר וחלק מהתוכניות משתמשות כקלט בתוכנם של התאים הראשונים R0,…,R15.
3. Keyboard and Screen – מחלקות אלו כבר מומשו עבורכם, ואין צורך לשנות בהן כלום
4. CPU16 – זאת המחלקה העיקרית בה תדרשו לממש קוד בסעיף זה. במחלקה כבר ממומש תרשים 5.9 מהספר (ראו למטה) ובוצעו כל החיבורים המתוארים בתרשים בין הרכיבים השונים, כפי שניתן לראות בבנאי של המחלקה. למחלקה ישנה פונקציה בשם PrintState שיכולה לסייע בשלב ה-debugging. אתם רשאים להוסיף לפונקציה זאת הדפסות כרצונכם.

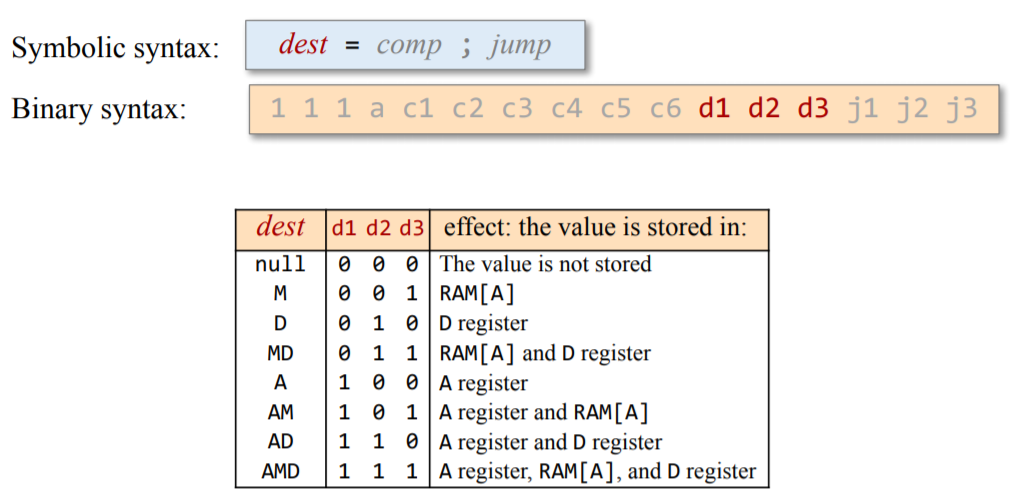


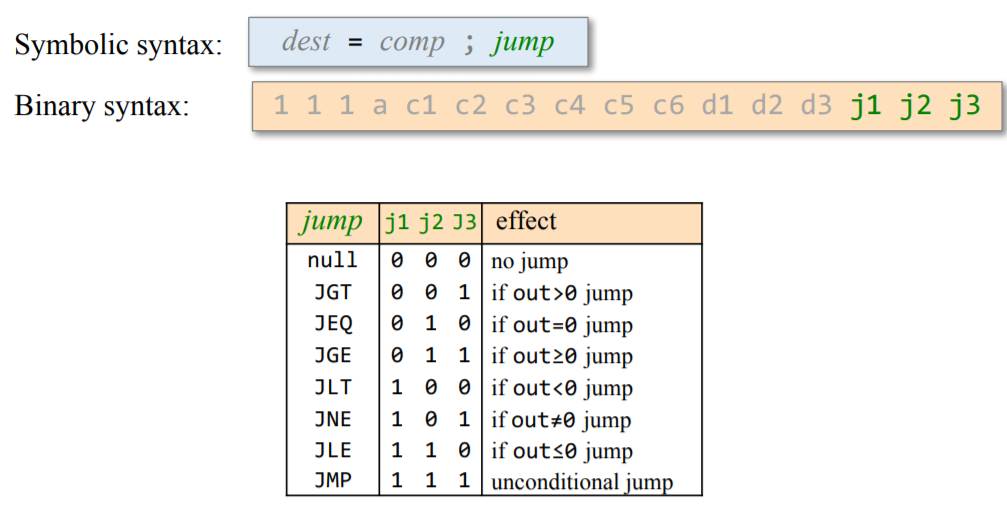
* 1. משימתכם היא לממש את מעגלי השליטה (control unit) של ה-CPU. לשם כך עליכם להוסיף קוד רק לשיטה ConnectControls במחלקה CPU16, ולהוסיף רכיבים כשדות למחלקה (מעל השיטה).
  2. בתוך השיטה מופיעות הערות המציעות את סדר העבודה על מימוש מעגלי השליטה. שימו לב, כאשר תסיימו את שלב 7, תוכלו להריץ כל תוכנית שאינה משתמשת בקפיצות בקוד. מומלץ לבדוק שלב זה עם תוכניות כגון Add ורק אחר כך לממש קפיצות.
  3. על מנת לבדוק את הקוד, עליכם להשתמש בתוכנות אסמבלי. לשם כך מצורפת לעבודה זאת תיקיה המכילה דוגמאות אסמבלי, והחבילה tools.zip המכילה מימוש של אסמבלר בו תוכלו להשתמש בכדי לתרגם את הדוגמאות לשפת מכונה. עליכם לתרגם תוכנית בעזרת האסמבלר ואז להריצה כפי שמודגם ב-Program.cs. מומלץ להתחיל ב-Add, אח"כ Max, ולבסוף ב-ScreenExample (התוכנה היחידה הדורשת שימוש במסך).

הגדרות תפקידי הביטים בשפת המכונה (<https://b1391bd6-da3d-477d-8c01-38cdf774495a.filesusr.com/ugd/56440f_12f488fe481344328506857e6a799f79.pdf>):









עבודה 2 כוללת מספר פרויקטים. כאשר solution כולל מספר פרויקטים, חלק מהפרויקטים יכולים להכיל מחלקות בהם ישתמשו בפרויקטים אחרים (כגון SimpleComponents ) ויתכנו גם מספר פרויקטים שניתן להריץ בעבודה זאת. כדי לקבוע איזה פרויקט מריצים כרגע, יש ללחוץ עם הכפתור הימני על הגדרת הפרויקט בsolution explorer (מצד ימין בד"כ) ולבחור set as start up project.

בכדי לאפשר לפרויקט אחד לגשת למחלקות שנמצאות בפרויקט אחר יש להוסיף reference. לחיצה ימנית על ה-references ב-solution explorer פותחת מסך המאפשר להוסיף התיחסות לפרויקט אחר.

לעבודה מצורפות תוכנות עזר בתיקיה tools.zip, הממשות, בין השאר, אסמבלר ומדמה (סימולטור) של התנהגות ה-CPU, שיאפשרו לכם לתרגם ולהריץ כל קוד אסמבלי שתרצו על מנת לבדוק האם המכונה והאסמבלר אותם בניתם מתפקדים כראוי.

אתיקה – את התרגיל יש לעשות לבד (לא בזוגות). מותר להתיעץ עם סטודנטים אחרים, אך לא להעתיק קטעי קוד מסטודנטים או מהאינטרנט. כל שליחה או קבלה של קטע קוד ממקור כלשהו – סטודנט אחר או מהאינטרנט, במייל, בדיסק נייד, או דרך אתרי שיתוף קבצים אסורה בהחלט! אל תעתיקו!

שאלות ובעיות יש לפרסם בפורום. במידה ולא תקבלו תשובה, יש לפנות לגיא במייל [shanigu@bgu.ac.il](mailto:shanigu@bgu.ac.il)

בהצלחה!

.