

דו"ח מכין מעבדה 2 :

חלק תאורטי:

הסבר מהי מחסנית, את הצורך בה ואופן שימושה:

מחסנית הינה מבנה נתונים הפועלת על עיקרון LIFO – האיבר האחרון אשר הוכנס אל המחסנית הוא האיבר הראשון שייצא בשליפה. על מנת לעבוד מול המחסנית, ישנו צורך במימוש מינימלי של פקודות אתחול כמו דחיפה ושליפה של המחסנית ובמצביע על הראש המחסנית (TOS). המחסנית נמצאת באזור הזיכרון של ה-RAM. היא מאכסנת בתוכנה ערכים שונים, כגון: רגיסטרים וכתובות וניתן להשתמש בה לצורך שימושים תכנותיים שונים. השימוש במחסנית הינו יעיל משום שניתן לחסוך באמצעותו סיבוכיות זמן וזיכרון (כמו לחסוך במספר של רגיסטרים ואף לקצר את זמן ריצת התוכנית וכדומה..) אופן שימושה של המחסנית:

במעבד ישנו רגיסטר ייחודי בשם SP שמטרתו לשמש מצביע(מכיל כתובת) לראש המחסנית וכפי שצינו הוא נקרא בשם TOS. כל תנועה במחסנית תשנה את ערכו של ה-SP, פקודות כמו PUSH, POP, RET, CALL and such. כאשר ה-SP הוא מצביע על ערך אחר במחסנית.

הסבר מהי רוטינה, את הצורך בה ואופן שימושה וכיצד היא משפיעה על המחסנית:

רוטינה(Routine) הינה קטע קוד אשר מיועד להרצה חוזרת, אשר כוללת תווית המצביעה לראש אזור הזיכרון בו מאוחסן קטע הקוד. הגישה לרוטינה הינה בזמן ריצת התוכנית. כאשר אנו יודעים שקוד שכתבנו צריך להיכתב מספר פעמים, או דרוש משתנים מוגדרים מראש על מנת לרוץ אנו נעדיף לכתוב את מקבץ הפקודות אשר יהווה את גוף הרוטינה ונבצע פקודת CALL, שלאחריה הרוטינה תרוץ לכל אורכה. בסוף הרוטינה נבצע פקודת RET אשר תעביר את ה-PC לשורה של הכתובת שלאחר פקודת ה-CALL. לרוטינה ישנן יתרונות רבים כמו מניעת סרבול, שמירה על סדר מובנות התכנית ואף הקלה על איתור תקלות עתידיות במידה ויקרו. הרוטינה משפיעה על המחסנית בכך שהיא עובדת על ערכים שונים שהכנסנו למחסנית ע"י טעינה ושליפה שלהם.

הסבר מהי פונקציית MACRO את הצורך בה ואופן שימושה:

רשום טבלת יתרונות וחסרונות בין פונקציית MACRO לבין רוטינה:

ברגיסטר SR, הסבר את צורת השימוש בדגלים V,C,N,Z ורשום פקודה לדוגמה המשפיעה על דגל בנפרד:

פונקציית MACRO בדומה לרוטינה, הינה פונקציה אשר משמשת את המתכנת כאשר יש צורך בכתיבת קוד שצפוי להיכתב מספר חוזר של פעמים ועוזר לתוכנה להיות מובנת יותר. למרות זאת, פונקציית המאקרו מתנהגת באופן שונה מהרוטינה.

פונקציית המאקרו אינה משפיעה על המחסנית ושומרת את ערכו של אוגר ה-PC בשונה מהרוטינה אשר עושה זאת ואף קופצת לאזור שונה בזיכרון. בנוסף לכך, פונקציית MACRO מחליפה שורת קוד מוגדרת מראש בקטע קוד שלם, כאשר תהליך זה מבוצע במהלך בניית התכנית.

אופן השימוש של פונקציית המאקרו: נרשום MACRO ולאחר מכן מקבץ פקודות אשר בסופה נרשום ENDM וכל פעם שנזדקק לפונקציה זו, נקרא לה.

יתרונות - מאקרו	חסרונות - מאקרו	יתרונות - רוטינה	חסרונות - רוטינה
מונע סרבול בתוכנית עם קטעי קוד חוזרים על עצמם.	מנפח את זיכרון הבקר	מונע סרבול בתוכנית עם קטעי קוד חוזרים על עצמם וקל לדיבוג.	מימוש פחות אינטואיטיבי- מצריך לזכור את סדר המשתנים המוכנס למחסנית. מחזורי שעון נוספים
שימוש אינטואיטיבי מזכיר מעט פונ' בשפות עילית(HLL)	מקשה על ניפוי שגיאות בתכנית בזמן ריצה.	קטע קוד מופיע פעם אחת יחידה בזיכרון	
חוסך בזמן ריצות השעון	יותר מורכב לדיבוג	קטע קוד משותף לכל הקריאות לרוטינה ומקל על ניפוי שגיאות	

הבהרות:

גודל התכנית - $0x3100 - 0x3140 = 0x40$

זמן ריצה - $126.882 \mu sec = 133 * 0.954 * 10^{-6}$ CycleCounter * T_mclk

מגישים:

יאיר טיירי- 207973017

עומר גראוברט 322480971