דו"ח מכין, מעבדה מס' 2 – Stack, Routine, Macro

הוראות עבודה במעבדה:

השלט הבא נמצא בכל עמדה בכיתת המעבדה 204/33, סעיף B רלוונטי החל מניסוי מספר 3 ואילך.

A. סדר פעולות בסיום יום העבודה:

1) **ביצוע shut down** מלא למחשב.

2) **כיבוי** מכשירי המדידה.

B. במידה והתקבלה בחלון סביבת IAR אחת ההודעות:

"Failed to initialize"

"Communication error"

נתק למשך 5 שניות את החיבור בין שני כבלי ה- USB (מאחורי ערכת הפיתוח של MSP430).

חומר עזר:

- ספר מעבדה MSP430x4xx user guide עמודים: 45, 407-414 (ללא עמודים 411,412)
- Tutorial 2.1, Tutorial 2.2 (חומר כתוב + וידאו).

A. חלק תיאורטי:

1. הסבר מהי מחסנית, את הצורך בה ואופן שימושה.
2. הסבר מהי רוטינה את הצורך בה ואופן שימושה וכיצד היא משפיעה על המחסנית.
3. הסבר מהי פונקציית MACRO את הצורך בה ואופן שימושה, רשום טבלת יתרונות וחסרונות בין פונקציית MACRO לבין רוטינה.

B. חלק מעשי – כתיבת תוכנית באסמבלי:

את משימת דו"ח מכין נדרש לכתוב בקובץ מקור חדש בשם **pre2.s43**

כאשר $ID1$, $ID2$ הם שני מערכים בגודל 8 המכילים את מספרי ת"ז (8 ספרות נמוכות), של חברי הקבוצה.

גודל המערך יוגדר ע"י משתנה $IDsize$ מטיפוס int. איבר בכל אחד משני המערכים הוא באורך 16bit.

עליכם להגדיר בנוסף משתנה בשם num בגודל 16bit (גודל טיפוס int).

נדרש לכתוב פונקציה (**בשימוש מחסנית – ראה Tutorial2.1 page 9**) המממשת את הביטוי לפי הטבלה הבאה:

הגרסה לביצוע הינה לפי ספרת האחדות של מספר הזהות הנמוך $ID_i < ID_j$ מבין שני בני הזוג.

לדוגמה: עבור זוג סטודנטים עם מספרי ת"ז הבאים ID1=2**0447105**0 , ID2=3**15212875**

מספר הגרסה לביצוע הוא 0. בגרסה זו לדוגמה (ראה סעיף C) נדרש לממש את הביטוי הבא:

$$num = \sum_{i=0}^{7} (ID1[i] \cdot ID2[i])$$

המטרה היא לבצע תרגום של הפסיאודו קוד הבא לקוד אסמבלי של MSP430 (המורכב מחלק של הגדרת הפונקציה וחלק של הקריאה לפונקציה).

הערה: בגרסאות אחרות פעולת הכפל מתחלפת בפעולה אחרת.

אפשרות 1 – עם ערך החזרה:

הגדרת הפונקציה func

הגדרת משתנים

קריאה לפונקציה func

```
int func (int id1[],int id2[],int size){
      int sum;
      for(int i=0 ; i<size ; i++) sum += id1[i]*id2[i];
      return sum;
}

int main(){
      int ID1={4,3,8,2,7,2,9,3},ID2={2,3,4,2,5,2,1,9},IDsize=8,num=0;
      num = func (ID1,ID2,IDsize); // IDsize=8
}
```

דגשים:

- כניסות של הפונקציה func:

id1,id2 – **pointer** to array of kind **int** (access to the array itself)

size – get the value of variable kind **int**

- מוצא של הפונקציה func:

The function returns value of kind **int**

אפשרות 2 - ללא ערך החזרה:

הגדרת הפונקציה func

הגדרת משתנים

קריאה לפונקציה func

```
void func (int id1[],int id2[],int size,int* pnum){
      for(int i=0 ; i<size ; i++) *pnum += id1[i]*id2[i];
}

int main(){
      int ID1={4,3,8,2,7,2,9,3},ID2={2,3,4,2,5,2,1,9},IDsize=8,num=0;
      func (ID1,ID2,IDsize,&num);  // IDsize=8, &num is the num addess
}
```

דגשים:

- כניסות של הפונקציה func:

id1,id2 – **pointer** to array of kind **int** (access to the array itself)

size – get the value of variable kind **int**

- מוצא של הפונקציה func:

pnum - **pointer** to variable of kind **int** (access to the variable itself)

C. חלוקה לגרסאות:

Version	Function	Note
0	$num = \sum_{i=0}^{7}(ID1[i] \cdot ID2[i])$	
1	$num = \sum_{i=0}^{7}(ID1[i] + ID2[i])$	
2	$num = \sum_{i=0}^{7}(ID1[i]\ \boldsymbol{or}\ ID2[i])$	
3	$num = \sum_{i=0}^{7}(ID1[i]\ \boldsymbol{xor}\ ID2[i])$	
4	$num = \sum_{i=0}^{7}(ID1[i] - ID2[i])$	
5	$num = \sum_{i=0}^{7}(ID1[i]\ \boldsymbol{and}\ ID2[i])$	
6	$num = \sum_{i=0}^{7}\boldsymbol{max}(ID1[i], ID2[i])$	
7	$num = \sum_{i=0}^{7}\boldsymbol{min}(ID1[i], ID2[i])$	
8	$num = \sum_{i=0}^{7}\boldsymbol{min\_odd}(ID1[i], ID2[i])$	If there is no odd ID digit, the result is 0
9	$num = \sum_{i=0}^{7}\boldsymbol{max\_even}(ID1[i], ID2[i])$	If there is no even ID digit, the result is 0

כתיבת פונקציה בשימוש מחסנית חייבת לשמור על העקרונות הבאים:

I. מה- main קוראים לפונקציה ע"י טעינה תחילה של הארגומנטים של הפונקציה למחסנית ולאחר מכן ביצוע קריאה לפונקציה (ע"י פקודת call).

II. בגוף הפונקציה מבצעים תחילה שליפת ארגומנטים מתוך המחסנית ולאחר מכן ביצוע גוף הפונקציה בעזרת רגיסטרים בלבד. במידה והפונקציה מחזירה ערך, נדרש לטעון אותו למחסנית לפני היציאה מהפונקציה (פקודת ret).

III. בחזרה ל- main (להמשך ביצוע) נדרש שערך רגיסטר SP יהיה שווה לערך שלפני הקריאה לפונקציה.

הבהרות:

- לבדיקת התוכנית יש להריצה בסימולטור.
- רשום את גודל התוכנית (לפי כתובת ראשונה ואחרונה של התוכנית בשימוש Disassembly)
- רשום את זמן הריצה שלה (ראה משתנה CYCLECOUNTER בחלון הרגיסטרים, המונה את מספר מחזורי השעון כמתואר ב- Tutorial 1.2). ערך תדר ברירת המחדל של שעון MCLK הוא:

$$f_{MCLK} = 32 \cdot 32768 = 2^{20} = 1,048,576\ Hz \rightarrow T_{MCLK} = \frac{1}{2^{20}} \approx 0.954\ \mu sec$$

צורת הגשה דוח מכין:

- הגשת מטלת דוח מכין תיעשה ע"י העלאה למודל של תיקיית zip מהצורה **id1_id2.zip** (כאשר id1 < id2), **רק הסטודנט עם הת"ז id1 מעלה את הקבצים למודל.**
- התיקייה תכיל את שני הפרטים הבאים בלבד:
  - ✓ **קובץ pre_lab$_x$.pdf** – מכיל תשובות לחלק תיאורטי דו"ח מכין
  - ✓ תיקייה בשם **IAR** - מכילה את קובצי המקור בלבד (קבצים עם סיומת ***.s43**) של מטלה מעשית דוח מכין.

צורת הגשה דוח מסכם:

- הגשת מטלת דוח מכין תיעשה ע"י העלאה למודל של תיקיית zip מהצורה **id1_id2.zip** (כאשר id1 < id2), **רק הסטודנט עם הת"ז id1 מעלה את הקבצים למודל.**
- התיקייה תכיל את שני הפרטים הבאים בלבד:
  - ✓ **קובץ final_lab$_x$.pdf** – מכיל תיאור והסבר לדרך הפתרון של מטלת זמן אמת.
  - ✓ תיקייה בשם **IAR** - מכילה את קובצי המקור בלבד (קבצים עם סיומת ***.s43**) של מטלת זמן אמת.

בהצלחה.