

Preface Lab 1

מגישים: יעקב קוזמינסקי 206511966, אור יעקובי 206827164

שאלות חלק תיאורטי – הרצת קוד לדוגמא בסביבת IAR:

1. הסבר את ההבדל בין משתנים לוקאלים למשתנים גלובליים.

ההבדלים בין משתנה לוקאלי למשתנה גלובלי מתבטאים בכמה אופנים:

מיקום הגדרת המשתנה - משתנה גלובלי מוגדר מחוץ לכל הפונקציות, בדרך כלל בתחילת התוכנית וערכו הדיפולטיבי הינו 0. בעוד שמשתנה לוקאלי מוגדר בתוך פונקציה וערכו הדיפולטיבי הוא ערך זבל.

"זמן החיים" - משתנה גלובלי יהיה קיים בזיכרון לכל אורך התוכנית וניתן יהיה לשנות את ערכו ולבצע עליו חישובים בכל חלקי התוכנית. בעוד שמשתנה לוקאלי מקבל הקצאת זיכרון (במחסנית) בשורת הגדרתו בתוך הפונקציה ונמחק בסיומה.

דוגמא למשתנה גלובלי: maxTrace והסקופ שלו הוא מהגדרתו ועד סוף התכנית.
דוגמא למשתנה לוקאלי: Max שהוא משתנה לוקאלי של הפונקציה DiagonalsMax והסקופ שלו הוא מהגדרתו ועד סיום הפונקציה.

2. מה כתובת המערך Mat 2 בזיכרון ומה טווח הכתובות אותו הוא מכסה. מהו סוג זיכרון זה

Address	Value
0x00001ee0	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001ef0	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001f00	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001f10	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001f20	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001f30	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001f40	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001f50	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001f60	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001f70	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001f80	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001f90	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001fa0	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001fb0	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001fc0	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001fd0	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001fe0	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00001ff0	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00002000	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00002010	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00002020	cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd cd
0x00002030	00 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00

sel 0x1f6c-0x2033

עבור הIAR:

כתובת המערך הינה 0x1f6c והטווח הינו עד 0x2033.

סך הכל 200 כתובות, כמספר הערכים של מטריצה כפול גודל של int בזיכרון שהוא 2 byte.

הערך של המערך בכל איבריו הוא cd cd

המערך נמצא בזיכרון stack מכיוון שהוא משתנה לוקאלי של הפונקציה הראשית main.

עבור הCCS:

Name	Type	Value	Location
Mat2	int[10][10]	[[0,0,0,0,...],[0,0,0,0,...],[0,0,0,...	0x04C6

המערך הוא מערך בגודל 10*10 של משתנים מסוג Int, לכן אמור לחסות כ100*גודל משתנה בזיכרון, ניתן לראות כי:

0x04C6 start, 0x058C end לכן מכסה בסה"כ 200 בתים (הכתובת סיום היא מקום המשתנה האחרון המכיל 2 בתים גם)

3. רשום את כתובת תחילת מיקום המחסנית בזיכרון הנקבע ע"י המהדר.

עבור IAR:

Name	Value
PC	0x0'2118
SP	0x0'20fc

בתחילת הריצה רגיסטר SP הוא המצביע על של המחסנית מאותחל לכתובת הראשונה של המחסנית מכיוון שעוד לא נכנס אליה שום ערך. להלן הכתובת:

0x020fc

עבור CCS:

Name	Value	Description
Core Registers		Core Registers
PC	0xC082	Core
SP	0x03FE	Core
SR	0x0003	Core

ניכנס לחלונית הרגיסטרים ונראה את כתובת ה SP בתחילת ההרצה: לכן כאן SP הוא 0x3FE

4. רשום את תוכן SP כאשר רגיסטר PC מצביע על הפקודה הראשונה של פונקציה ComputeTrace

עבור IAR:

Name	Value
PC	0x0'2360
SP	0x0'1f64

ניתן לראות כי הכתובת שעליה מצביע ה SP השתנתה כי בקריאה לפונקציה הכתובת האחרונה שבה היה ה PC. נכנסת למחסנית על מנת שבסיום התכנית נחזור לנקודה האחרונה בה התכנית הקודמת הייתה.

עבור CCS:

The screenshot shows the CCS IDE with the `ComputeTrace` function called at line 56 of `main.c`. The disassembly window shows the following instructions:

```

c2d8: 120A      PUSH    R10
c2da: 4C0A      MOV.W   R12, R10
57:      int Trace=0,i;
c2dc: 430B      CLR.W   R11
58:      for(i=0 ; i<M ; i++) Trace += Mat;
c2de: 430F      CLR.W   R15
c2e0: 903F 000A  CMP.W   #0x000a, R15
c2e4: 340B      JGE     ($C$L9)
c2e6: 4F0C      MOV.W   R15, R12
  
```

נראה כי כתובת ה PC מבצעה על השורה הראשונה בפונקציה לפי ההדגשה בחלקן הדיסאסמבלי. ולכן כתובת ה SP כעת היא 0x0268.

5. רשום את כתובת הפונקציה FillMatrix בזיכרון. מה גודל קוד הפונקציה FillMatrix בבתים ? מהו סוג זיכרון זה?

ב-IAR:

```
void FillMatrix(int Mat[M][M]){
```

FillMatrix:

```
002322 153B          pushm.w #4,R11
```

בחלון ה disassembly ניתן לראות את הפונקציה במלואה עם כל שורות הקוד שלה וכיצד הפקודות מאוחסנות בזיכרון.

כתובת הפונקציה היא הכתובת הראשונה והיא, 0x02322

גודל הקוד הוא מהכתובת האחרונה פחות הראשונה: byte 0x0235C-0x02322 = 59

סוג הזיכרון לאחסון הפונקציות הוא זיכרון flash.

ב-CCS:

נראה את קוד האסמבלי החלון הדיסאסמבלי:

נוכל לחסר בין הכתובת האחרונה לראשונה בפונקציה ונראה כי גודל הפונקציה הוא 76 בתים.

FillMatrix():

```
c288: 120A          PUSH
c28a: 1209          PUSH
c28c: 1208          PUSH
c28e: 4C08          MOV.W
*100  for(i=0 ; i<M ; i++){
c290: 430A          CLR.W
c292: 903A 000A     CMP.W
c296: 341E          JGE
101  for(j=0 ; j<M ; j++){
$c$L20:
c298: 430B          CLR.W
c29a: 903B 000A     CMP.W
c29e: 3416          JGE
102  Mat[i][j] = i*M+j;
$c$L21:
c2a0: 4A0C          MOV.W
c2a2: 403D 000A     MOV.W
c2a6: 12B0 C330     CALL
c2aa: 4C09          MOV.W
c2ac: 5B09          ADD.W
c2ae: 4A0C          MOV.W
c2b0: 403D 0014     MOV.W
c2b4: 12B0 C330     CALL
c2b8: 4B0F          MOV.W
c2ba: 5F0F          RLA.W
c2bc: 5F0C          ADD.W
c2be: 580C          ADD.W
c2c0: 498C 0000     MOV.W
101  for(j=0 ; j<M ; j++){
c2c4: 531B          INC.W
c2c6: 903B 000A     CMP.W
c2ca: 3BEA          JL
100  for(i=0 ; i<M ; i++){
$c$L22:
c2cc: 531A          INC.W
c2ce: 903A 000A     CMP.W
c2d2: 3BE2          JL
105  }
$c$L23:
c2d4: 4030 C35E     BR
```

6. מהו זמן ריצת הקוד של הפונקציה FillMatrix ביחידות מחזור של MCLK.

ב-IAR:

לפני ריצת הפונקציה:

CYCLECOUNTER	58
--------------	----

לאחר ריצת הפונקציה:

CYCLECOUNTER	3'407
--------------	-------

סך הכל 3349 מחזורי שעון.

7. מהו ה scope של משתנה Trace2mat בתוכנית, מהו מיקומו בזמן ה- scope

המשתנה Trace2mat הוא משתנה לוקאלי של פונקציית הmain ולכן ה scope שלו הוא כל פונקציית הmain. בזמן ה scope המשתנה נמצא במחסנית.

8. רשום את קוד האסמבלי המתורגם ע"י המהדר עבור שורת הקוד הבאה $maxTrace = mat1Trace > mat2Trace ? mat1Trace : mat2Trace;$

ב-IAR:

מתוך הdisassembly:

maxTrace = mat1Trace > mat2Trace ? mat1Trace : mat2Trace;			
002158	9A0C	cmp.w	R10, R12
00215A	3403	jge	0x2162
00215C	4A82 1100	mov.w	R10, &maxTrace
002160	3C02	jmp	0x2166
002162	4C82 1100	mov.w	R12, &maxTrace

ב-CCS:

33	maxTrace = mat1Trace > mat2Trace ? mat1Trace : mat2Trace;		
c0f8:	990C	CMP.W	R9, R12
c0fa:	3401	JGE	(\$C\$L7)
c0fc:	490C	MOV.W	R9, R12
\$C\$L7:			
c0fe:	4C82 0202	MOV.W	R12, &maxTrace
35	break;		

נוכל לראות שהפוקודת זהות כמעט לגמרי וההבדל הוא ברגיסטרים ובלייבלים בלבד.

