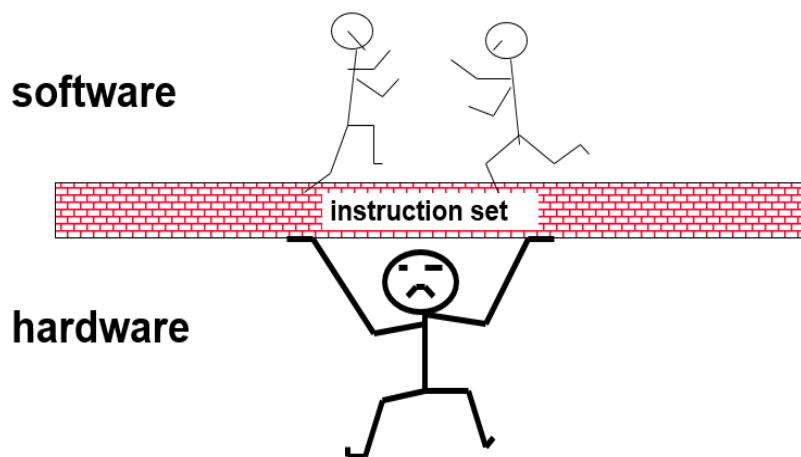
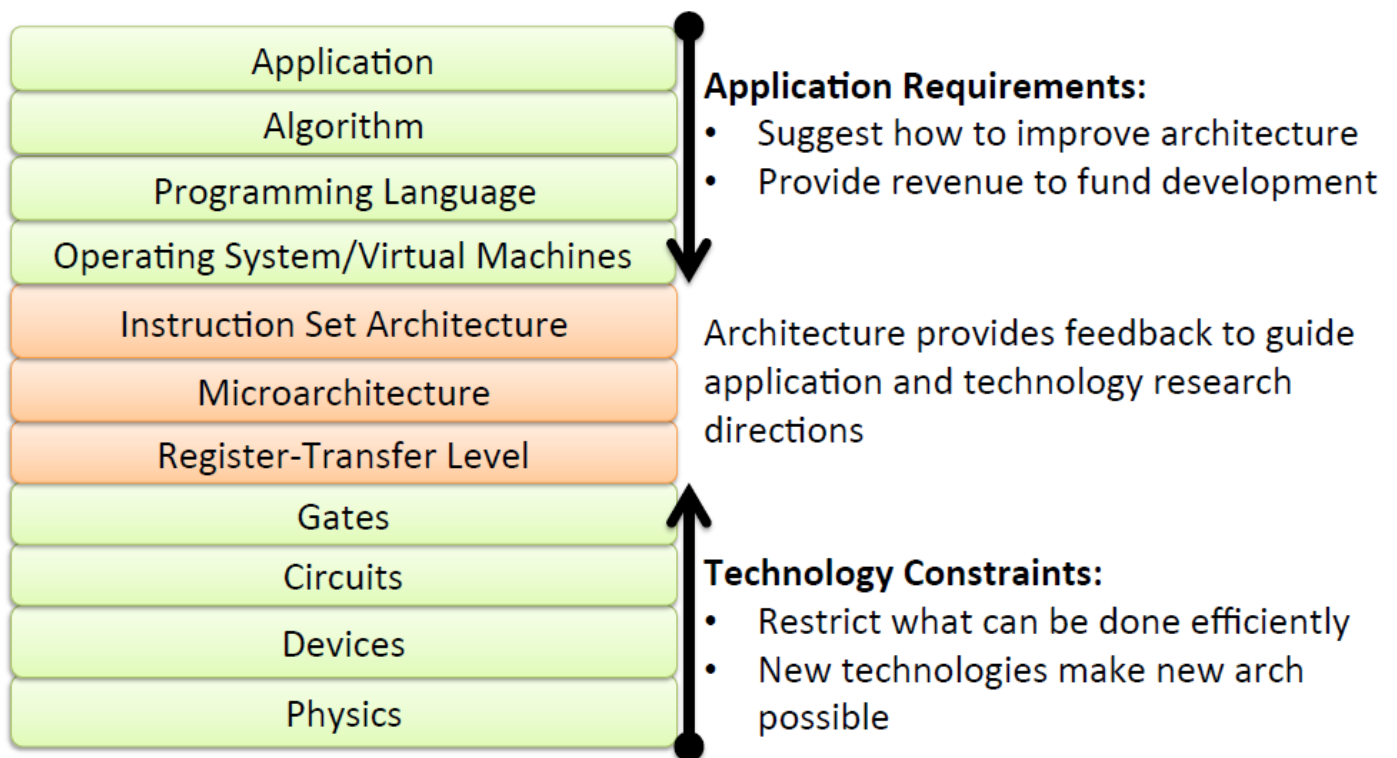
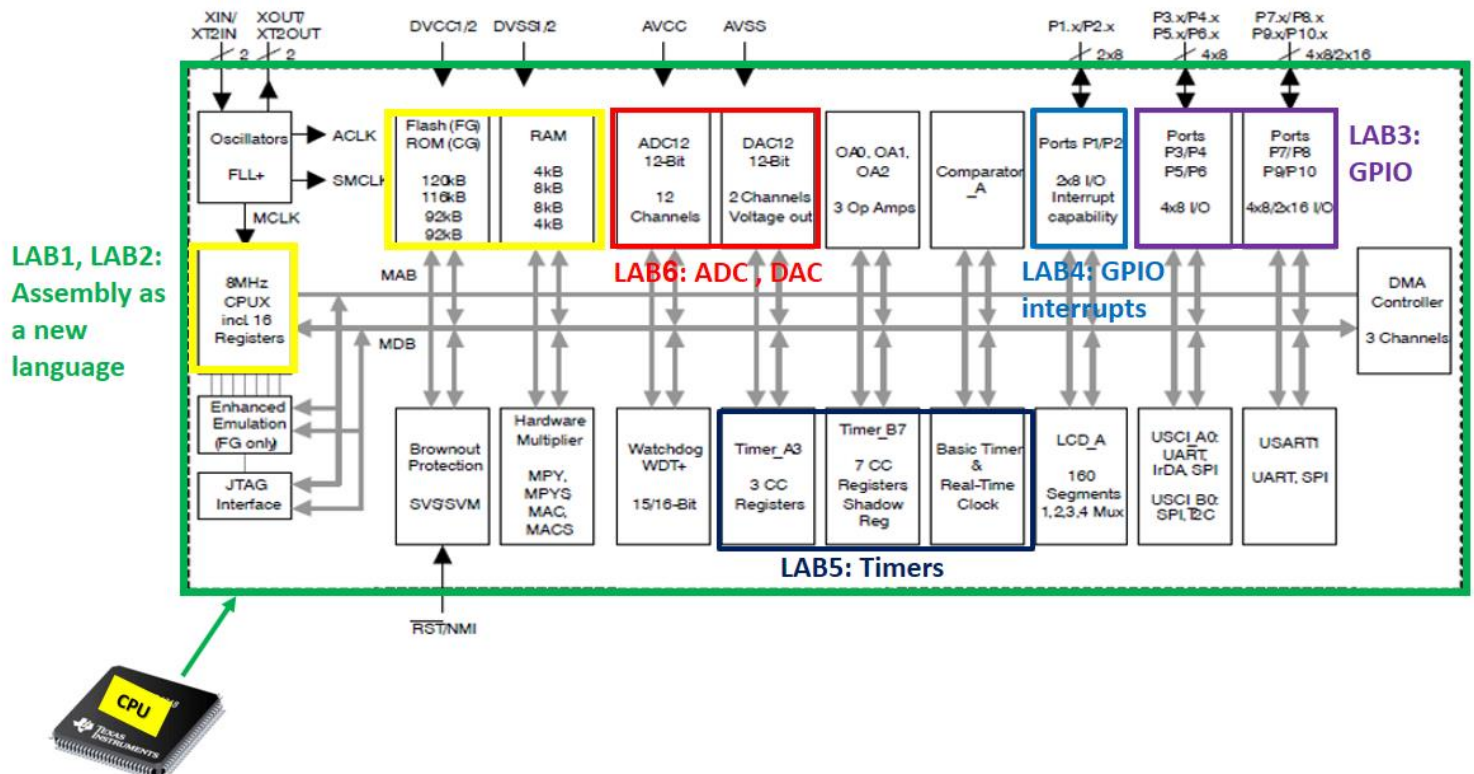
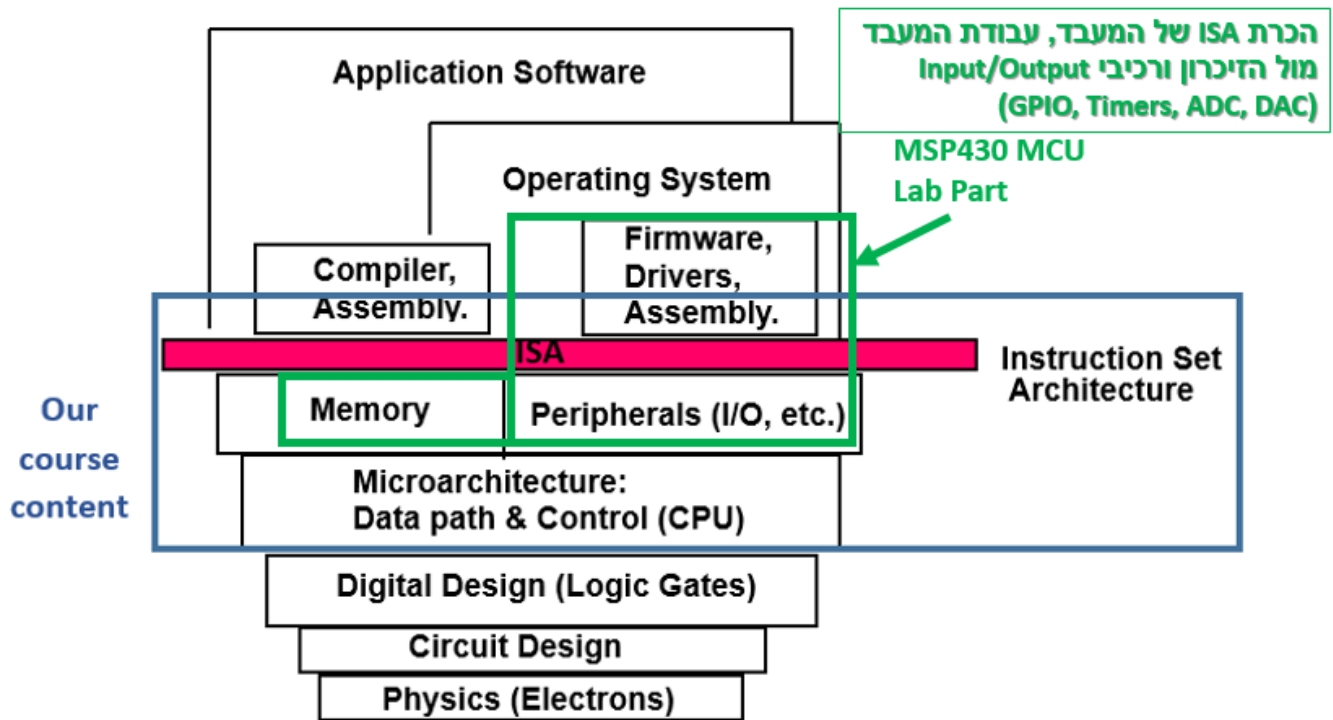


Review: Instruction Set Design

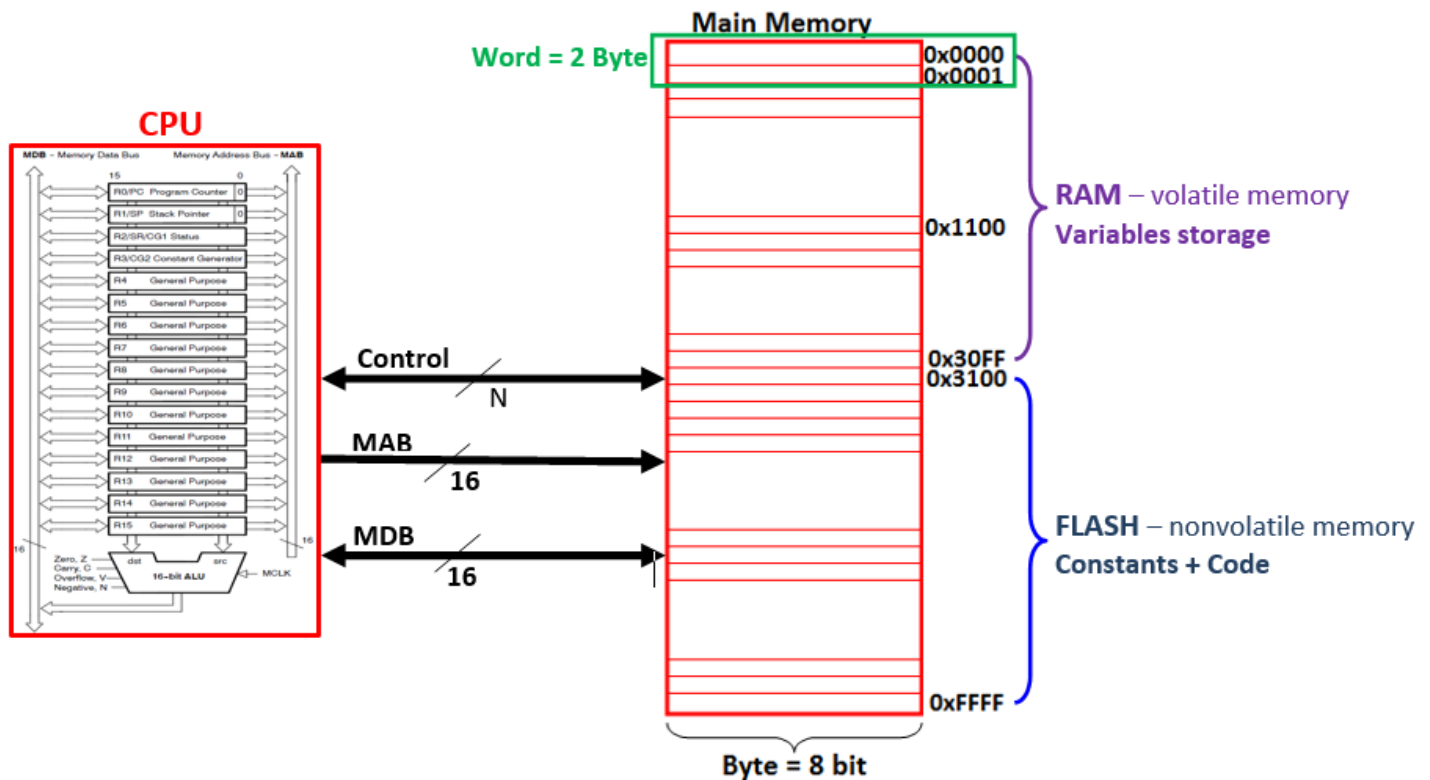


חלוקת מרחב פתרון בעיה מבוססת מחשב לשכבות אבסטרקציה

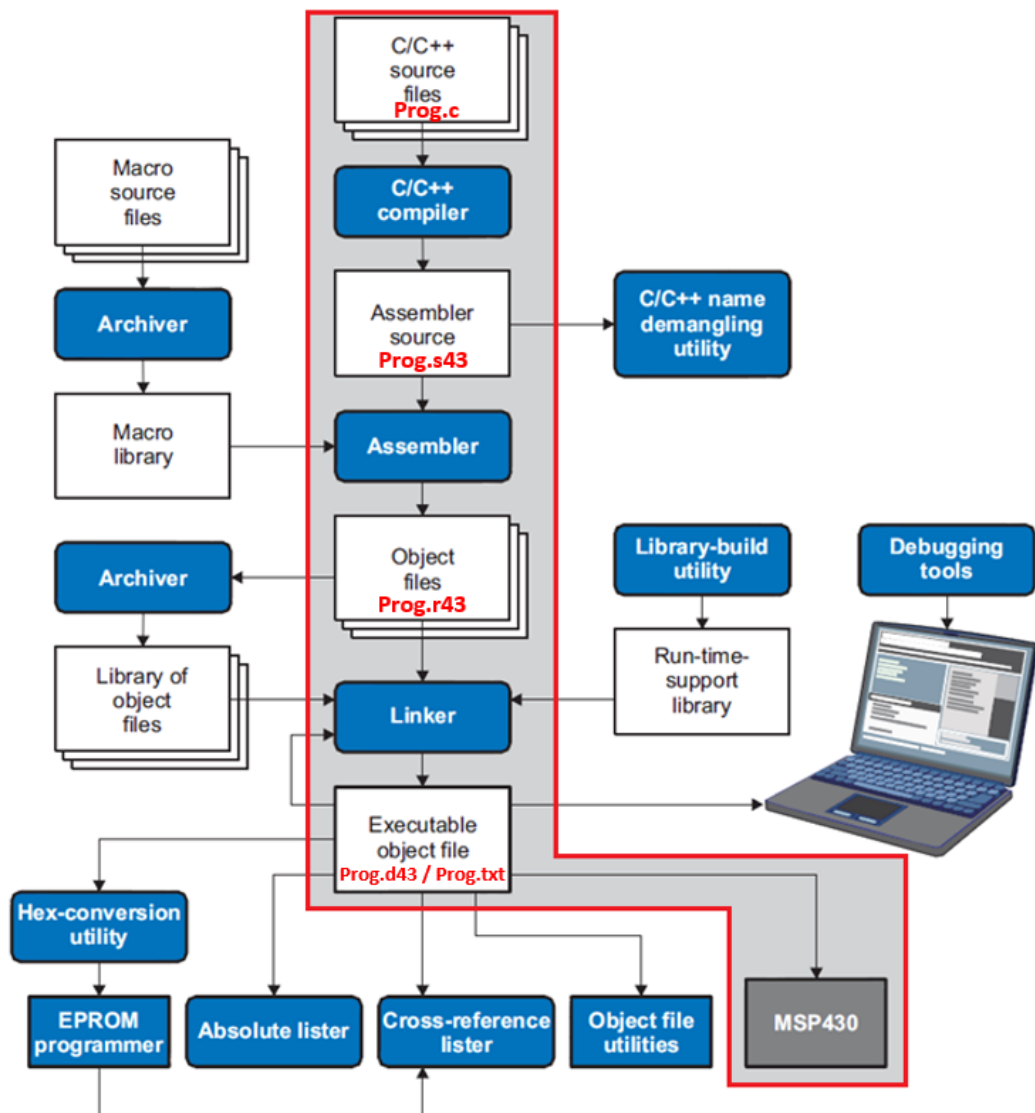




קשר אבסטרקטי בין מעבד MSP430 לזיכרון קוד והנתונים של התוכנית (ללא הגבלת הכלליות)



Cross Compilation Process:



שיטות מעון של ISA במעבד RISC MSP430 (ללא הגבלת הכלליות)

את מיקום הנתונים ע"י אופרנד המקור (**Operand_SRC**) ניתן לציין בשבעה אופנים ואת מיקום הנתונים ע"י אופרנד היעד (**Operand_DST**) ניתן לציין בארבעה אופנים. אופני ייצוג אלו נקראים "שיטות מעון".

ADDRESS MODE	S	D	SYNTAX	EXAMPLE	OPERATION
Register	●	●	MOV Rs,Rd	MOV R10,R11	$R10 \rightarrow R11$
Indexed	●	●	MOV X(Rn),Y(Rm)	MOV 2(R5),6(R6)	$M(2+R5) \rightarrow M(6+R6)$
Symbolic (PC relative)	●	●	MOV EDE,TONI		$M(EDE) \rightarrow M(TONI)$
Absolute	●	●	MOV & MEM, & TCDAT		$M(MEM) \rightarrow M(TCDAT)$
Indirect	●		MOV @Rn,Y(Rm)	MOV @R10,Tab(R6)	$M(R10) \rightarrow M(Tab+R6)$
Indirect autoincrement	●		MOV @Rn+,Rm	MOV @R10+,R11	$M(R10) \rightarrow R11$ $R10 + 2 \rightarrow R10$
Immediate	●		MOV #X,TONI	MOV #45,TONI	$\#45 \rightarrow M(TONI)$

NOTE: S = source D = destination

Note: MOV #LABEL,PC ; Branch to address LABEL
 MOV LABEL,PC ; Branch to address contained in LABEL
 MOV @R14,PC ; Branch indirect to address in R14

שיטות מעון של ISA במעבד RISC SRC (ללא הגבלת הכלליות)

פקודה	משמעות	תאור	שיטת מיעון
$la\ r_a, c_2$	$R[r_a] \leftarrow c_2$	הפקודה מכילה את הערך המבוקש, ולכן ניתן לאחסן ברגיסטרים בלבד. c_2 מכיל 16 סיביות.	<u>מידע</u>
$ld\ r_a, c_2$	$R[r_a] \leftarrow M[c_2]$	הפקודה מכילה את הכתובת בזיכרון בה מצוי האופרנד.	<u>ישיר</u>
$ld\ r_a, (c_2)$	$R[r_a] \leftarrow M[M[c_2]]$	הפקודה מכילה את הכתובת של הכתובת של האופרנד.	<u>עקיף</u>
$add\ r_a, r_b, r_c$	$R[r_a] \leftarrow R[r_b] + R[r_c]$	האופרנד הוא תוצאת חיבור של הרגיסטרים המצוינים.	<u>ישיר רגיסטר</u>
$ld\ r_a, (r_b)$	$R[r_a] \leftarrow M[R[r_b]]$	הכתובת נמצאת ברגיסטר המצוין הערה: אין אפשרות לבצע מיעון היסט עם r_0 , במקום זאת יתבצע מיעון ישיר.	<u>עקיף רגיסטר</u>
$ld\ r_a, 20(r_b)$	$R[r_a] \leftarrow M[R[r_b] + 20]$	הכתובת ברגיסטר + היסט נתון.	<u>היסט</u>
$ldr\ r_a, 20$	$R[r_a] \leftarrow M[PC + 20]$	הכתובת ב- PC (זהו גם רגיסטר) + היסט נתון.	<u>יחסי</u>

תרגילים במעבר מידע בין מעבד-זיכרון בשפת אסמבלי של MSP430 (כולל פתרון ב Python)

הערה חשובה: כפי שפורסם התרגילים הבאים מבוססים הכנה מקדימה של חומר ההכנה למעבדה הנמצא בקובץ הבא:

[Tutorial 1.1 - Assembly acquaintance, Part 1](#)

תרגיל 1:

הגדר שני מערכי ת"ז של זוג סטודנטים ID1, ID2 באורך 8 .

נדרש להגדיר בזיכרון ה- RAM את שני המערכים Parity1 , Parity2 באורך 8 בצורה הבאה:

Parity1 **DS16** 8

Parity2 **DS16** 8

בגוף התוכנית נדרש לבצע את הפעולות הבאות:

$$\forall i \in [0,7] \text{ Parity1}[i] = \begin{cases} 1, ID1[i] \text{ is odd} \\ 0, ID1[i] \text{ is even} \end{cases}$$

$$\forall i \in [0,7] \text{ Parity2}[i] = \begin{cases} 1, ID2[i] \text{ is odd} \\ 0, ID2[i] \text{ is even} \end{cases}$$

הסבר: כל אחד מהמערכים Parity1 , Parity2 מוגדר לכל איבר בנפרד להיות 1 אם האיבר באינדקס זהה במערך

תעודת הזהות הוא אי-זוגי אחרת 0.

פתרון תרגיל 1:

Id1	9	6	2	8	3	5	7	4
-----	---	---	---	---	---	---	---	---

Id2	8	6	3	2	3	2	0	6
-----	---	---	---	---	---	---	---	---

Parity1	1	0	0	0	1	1	1	0
---------	---	---	---	---	---	---	---	---

Parity2	0	0	1	0	1	0	0	0
---------	---	---	---	---	---	---	---	---

1. פתרון קוד בשפת Python (שפה עילית ברמת אבסטרקציה גבוהה) ראה קובץ בשם **EX1_array.py**

2. פתרון קוד בשפת C (שפה עילית ברמת אבסטרקציה נמוכה) ראה קובץ בשם **EX1_array.c**

3. פתרון קוד בשפת אסמבלי של מעבד MSP430 (שפה סף ללא רמת אבסטרקציה) ראה קובץ בשם **EX1_array.s43**

תרגיל 2:

הגדר שני מערכי ת"ז של זוג סטודנטים Arr1, Arr2 באורך 8.

נדרש להגדיר בזיכרון ה-RAM את שני המערכים mod4_ID1, mod4_ID2 באורך 8 בצורה הבאה:

mod4_ID1 **DS16** 8

mod4_ID2 **DS16** 8

בגוף התוכנית נדרש לבצע את הפעולות הבאות:

$$\forall i \in [0,7] \text{ mod4_ID1}[i] = \begin{cases} 0, & \text{Arr1}[i] \bmod 4 = 0 \\ -1, & \text{else} \end{cases}$$

$$\forall i \in [0,7] \text{ mod4_ID2}[i] = \begin{cases} 0, & \text{Arr2}[i] \bmod 4 = 0 \\ -1, & \text{else} \end{cases}$$

הערה: תוצאת mod4 של מספר בבסיס 2 הוא הערך של שתי סיביות LSB של המספר.

פתרון תרגיל 2:

Arr1	21	80	99	49	18	81	77	13
------	----	----	----	----	----	----	----	----

Arr2	31	13	88	65	25	10	52	40
------	----	----	----	----	----	----	----	----

mod4_ID1	-1	0	-1	-1	-1	-1	-1	-1
----------	----	---	----	----	----	----	----	----

mod4_ID2	-1	-1	0	-1	-1	-1	0	0
----------	----	----	---	----	----	----	---	---

1. פתרון קוד בשפת Python (שפה עילית ברמת אבסטרקציה גבוהה) ראה קובץ בשם **EX2_array.py**
2. פתרון קוד בשפת C (שפה עילית ברמת אבסטרקציה נמוכה) ראה קובץ בשם **EX2_array.c**
3. פתרון קוד בשפת אסמבלי של מעבד MSP430 (שפה סף ללא רמת אבסטרקציה) ראה קובץ בשם **EX2_array.s43**

תרגיל 3:

עליך לכתוב תוכנית לחישוב העקבה (TRACE) של מטריצה סימטרית.

העקבה של מטריצה סימטרית בגודל $n \times n$ מוגדרת ע"י סכום איברי האלכסון של המטריצה, כדלקמן:

עבור מטריצה סימטרית $A_{n \times n}$ העקבה נתונה ע"י הנוסחה: $\text{Trace}(A) = a_{11} + a_{22} + \dots + a_{nn}$

גודל כל איבר במטריצה הוא 16 סיביות. הערך המתקבל עבור עקבת המטריצה המחושבת, יש לאחסן

ברגיסטרים R12, R13. כאשר, החלק הנמוך LSB של העקבה יאוחסן ברגיסטר R12 והחלק הגבוה MSB

של העקבה יאוחסן ברגיסטר R13. כאשר **Matrix** הינה מטריצה ותוויית זז, היא כתובת האיבר הראשון של

המטריצה. המשתנה **Lines** מייצג את מספר השורות של המטריצה.

את תוצאת העקבה יש לאחסן במשתנה בשם Trace.

מטריצה מאוחסנת בזיכרון ע"י מערך של שורות (שורה אחר שורה החל משורה ראשונה).

פתרון תרגיל 3:

$$\text{Matrix}_{4 \times 4} \triangleq \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

$$a_{i,j} = \text{Matrix}[\text{Cols} \cdot 2i + 2j] \text{ when } 0 \leq i, j \leq \text{Lines} - 1$$

- Part of Diagonal index = MatrixPTR+2j+Lines*2i
- $0 \leq i \leq \text{Lines}-1$ and $0 \leq j \leq \text{Lines}-1$ for all Parts of Diagonal $i=j$ So, Part of Diagonal index = MatrixPTR+2i(Lines+1)
- Trace=sum of M[MatrixPTR+2i(Lines+1)] when $0 \leq i \leq \text{Lines}-1$

1. פתרון קוד בשפת Python (שפה עילית ברמת אבסטרקציה גבוהה) ראה קובץ בשם EX2_array.py

2. פתרון קוד בשפת C (שפה עילית ברמת אבסטרקציה נמוכה) ראה קובץ בשם EX2_array.c

3. פתרון קוד בשפת אסמבלי של מעבד MSP430 (שפה סף ללא רמת אבסטרקציה) ראה קובץ בשם EX2_array.s43