

Archlab - Lab1

Gal Kashi – 204572861

Chen Eilon – 201617032

שאלה 1

1. LD R5 0 R1 10
2. LD R5 0 R1 -512
3. SUB R4 R3 R4 0
4. JLT 0 R3 R1 33
5. Assuming the constant is 0xaaaabbbb and the destination is R2:
ADD R2 0 R1 0xbbbb
LHI R2 0 0 0xaaaa
6. בשביל לאפשר קריאות קנוניות לפונקציות, נשמור מחסנית של כתובות חזרה בזיכרון. כאשר R6 יהיה הפוינטר לראש המחסנית. בתחילת כל subroutine נבצע פעולת ST כדי לשמור את כתובת החזרה מתוך R7 בכתובת המאוחסנת ב-R6, ואז נקדם את R6 ב-1. בסוף כל subroutine נחסר 1 מ-R6 ואז נבצע LD כדי לטעון את כתובת החזרה מהכתובת המאוחסנת ב-R6. לבסוף נבצע קפיצה לכתובת שנטענה כדי לחזור לקורא.

שאלה 2

1. התוכנית מחשבת סכום של 8 מספרים ששמורים בזיכרון מכתובת 15 עד כותבת 22.
2. קלטי הפונקציה מאוחסנים בכתובות 15 עד 22.
3. פלט (הסכום) הפונקציה נשמר בכתובת 22.
4. מצ"ב
5. פסאודו קוד:

0: R2 = 15
1: R3 = 0
2: R4 = 8
3: R5 = 0
4: PC = 10 if R3 == R4
5: R6 = MEM[R2]
6: R5 = R5 + R6
7: R2 = R2 + 1
8: R3 = R3 + 1
9: PC = 4
10: MEM[22] = R5
11: HALT

שאלה 3

1. Mult פסאודו-קוד (עפ"י הפקודות המוגדרות בשאלה, קוד אסמבלי מצורף בסוף המסמך):

```
0: R2 = MEM[1000]    /* Loading 2 numbers as mult input*/
1: R3 = MEM[1001]
2: R6 = 0 + 1
3: PC = 24 if (R2 == 0) /* If one of the numbers is zero, put zero in output cell & halt*/
4: PC = 24 if (R3 == 0)
5: R5 = XOR(R2,R3)    /* R5 will hold the multiplication sign*/
6: PC = 8 if(R3 > 0)   /* If one of the multiplicands is below zero, change its sign*/
7: R3 = 0-R3
8: PC = 10 if (R2 > 0 )
9: R2 = 0-R2
10: PC = 20 if (R2 > R3) /* Making sure R2 holds the smaller absolute multiplicand*/
11: PC = 26 if (R2 == 1) /* Looping till R2 is one (R2 is smaller for efficiency)*/
12: R4 = R2 AND 1      /* Let R4 hold the lsb of R2, which is the parity sign*/
13: PC = 17 if R4 == 0 /* If R2 is odd: */
14: R3 = R3 + R2       /* R3 will hold result*/
15: R2 = R2 - 1
16: PC = 11            /*Jump to loop condition*/
17: R2 >> 1            /*If R2 is even: */
18: R3 << 1
19: PC = 11            /*Jump to loop condition*/
20: R4 = R2             /* Swapping R2 and R3*/
21: R2 = R3
22: R3 = R4
23: PC = 11            /*Jump to loop condition (next thing to do)*/
24: MEM[1002] = 0      /*handling zero multiplicand*/
25: HALT
26: PC =28 if (R5 >= 0) /*Updating result sign*/
27: R3 = 0-R3
28: MEM[1002] = R3     /*Updating output*/
29: HALT
```

2. הקבצים מצורפים.

שאלה 4

1. Fibo פסאודו-קוד (עפ"י הפקודות המוגדרות בשאלה, קוד אסמבלי מצורף בסוף המסמך):

```
0: R2 = 1          /* Initializing first fibo sequence values*/
1: R3 = 1
2: MEM[1000] = R2   /* Storing first values in memory*/
3: MEM[1001] = R3
4: R4 = 1002        /* Updating first output memory address*/
5: R5 = 0
6: R6 = 98          /* Number of fibo numbers to calculate*/
7: PC = 15 if R5 == R6 /* Looping till we calculated all needed fibo numbers*/
8: R7 = R2 + R3     /* Adding 2 previous values in the series */
9: R2 = R3          /* Moving R2, R3 one place forward in the series*/
10: R3 = R7
11: MEM[R4] = R7    /* Update calculated value in memory*/
12: R4 = R4 + 1     /* Moving memory pointer forward, for next number*/
13: R5 = R5 + 1     /* Counting calculated fibo numbers*/
14: PC = 7
15: HALT
```

2. הקבצים מצורפים.

שאלה 5

הקבצים מצורפים.

נספחים

קוד אסמבלי שאלה 3:

```
LD, 2, 0, 1, 1000
LD, 3, 0, 1, 1001
ADD, 6, 0, 1, 1
JEQ, 0, 2, 0, 24
JEQ, 0, 3, 0, 24
XOR, 5, 2, 3, 0
JLT, 0, 0, 3, 8
SUB, 3, 0, 3, 0
JLT, 0, 0, 2, 10
SUB, 2, 0, 2, 0
JLT, 0, 3, 2, 20
JEQ, 0, 2, 6, 26
AND, 4, 2, 1, 1
JEQ, 0, 4, 0, 17
ADD, 3, 3, 3, 0
SUB, 2, 2, 1, 1
JEQ, 0, 0, 0, 11
RSF, 2, 2, 1, 1
LSF, 3, 3, 1, 1
JEQ, 0, 0, 0, 11
ADD, 4, 2, 0, 0
ADD, 2, 3, 0, 0
ADD, 3, 4, 0, 0
JEQ, 0, 0, 0, 11
ST, 0, 0, 1, 1002
HLT, 0, 0, 0, 0
JLE, 0, 0, 5, 28
SUB, 3, 0, 3, 0
ST, 0, 3, 1, 1002
HLT, 0, 0, 0, 0
```

קוד אסמבלי שאלה 4:

```
ADD, 2, 1, 0, 1
ADD, 3, 1, 0, 1
ST, 0, 2, 1, 1000
ST, 0, 3, 1, 1001
ADD, 4, 1, 0, 1002
ADD, 5, 1, 0, 0
ADD, 6, 1, 0, 98
JEQ, 0, 5, 6, 15
ADD, 7, 2, 3, 0
ADD, 2, 3, 0, 0
ADD, 3, 7, 0, 0
ST, 0, 7, 4, 0
ADD, 4, 4, 1, 1
ADD, 5, 5, 1, 1
JEQ, 0, 0, 0, 7
HLT, 0, 0, 0, 0
```