ארגון המחשב ושפת סף (203:1130) סמסטר ב' תשס"ה בחינה סופית - מועד א'

הוראות לנבחן:

- משך הבחינה שלש שעות.
- מותר להשתמש בכל חומר עזר, למעט מחשבים ומחשבונים מכל סוג.
 - יש להשיב על כל השאלות.
 - יש לרשום את התשובות בגוף השאלון במקומות המיועדים לכך.
 - נא לכתוב בכתב יד ברור ונקי. מומלץ להשתמש בעפרון ומחק.
 - בשאלון זה 14 דפים, כולל דף זה. ודא כיַ כל הדפים נמצאים.

בהצלחה!

ציון	ניקוד	
21	25	שאלה 1
19	25	שאלה 2
19	25	שאלה 3
17	25	שאלה 4
76	100	. סה״כ



שאלה מס׳ 1 (25 נקודות)

- א. לגבי סגמנטים בתכנית שרצה על מעבד X86 במצב real , אילו מהטענות הבאות נכונות? תתכן יותר מתשובה אחת נכונה. הקף בעגול את כל התשובות הנכונות.
 - כתובת התחלת סגמנט הקוד מיושרת (aligned) על כפולה של 16.
 - .ii. בתכנית שרצה על מעבד 8086 יש לכל היותר ארבעה סגמנטים.
 - . בתים 2^{20} בתים הגודל המקסימאלי של סגמנט הוא בונו.
 - יב. כתובות התחלת הסגמנטים ידועות בזמן הקישור של התכנית.
 - . ע. בכל תכנית יש סגמנט מחסנית אחד ויחיד.
 - .vi כל סגמנט חייב להיות חלק מקבוצה (group).
- ב. לגבי תוויות (labels) בקוד שכתוב בשפת אסמבלי של X86, אילו מהטענות הבאות נכונות? תתכן יותר מתשובה אחת נכונה. הקף בעגול את כל התשובות הנכונות.
 - ב. תווית חייבת להיות מוגדרת לפני השימוש הראשון בה.
 - ָלכל תווית יש ערך מספרי. 🚉
 - תווית אינה יכולה להיות מוגדרת יותר מפעם אחת בכל קובץ מקור של התכנית.
 - התוויות משמשות בתהליך האסמבלי בלבד. 🤕
 - . ע. באופרנד של פקודה (בשיטת המיעון) יכולה להופיע לכל היותר תווית אחת.
 - .vi בקוד אטמבלי אין אבחנה בין אותיות קטנות וגדולות בשמות התוויות.
- ג. עבור כל אחד משלשת קטעי הקוד שלהלן, רשום את תוכנו של האוגר dx בגמר ביצוע הקטע. רשום את התשובה בבסיט 10.

.data list db .code xor lea mov add mov mov c: dec pop cmp loopn mov	dx,dx bx,list sp,bx bx,13 cx,7 ax,@data ds,ax ss,ax bx ax ax,[bx]	6,7,8,9,10,11,12,13	b:	xor		xor dx,dx n = 4 rept n local a mov cx,n a: inc dx n = n - 1 loop a endm
dx =	2	,	dx		64	dx =

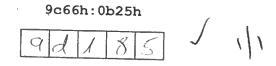
-2/7

<u>המשך שאלה מסי 1 בדף הבא</u>

שאלה מס' 1 (המשך)

ד. תרגם את הכתובת הלוגית (segment:offset) שלהלן לכתובת פיזית, כשהמעבד במצב real. רשום את התוצאה בבסיס 16.

90685 h



ה. בצע את פעולות החיבור והחיסור שלהלן בשיטת המשלים ל- 2 ברוחב של 16 ביטים. כל המספרים נתונים בבסיס 16. רשום גם את התוצאות בבסיס 16. ציין את ערכי הדגלים CF ו- OF בגמר כל פעולה, כפי שהיו נקבעים על ידי ביצוע במעבד X86.

abbb	abbb	5£63	5£63	
3562 E// O	3562 7657	b2c2 //225	b2c2	ſ
CF=	CF=_()	CF=_/	CF=_/	8/8
OF= <u></u>	OF=_/	OF= 0	OF= <u>/</u>	

ו. תרגם את המספרים שבטבלה מבסיס 10 לייצוג סטנדרטי בשיטת הנקודה הצפה בבסיס 2. רשום את החזקה <u>ללא bias</u> (ראה דוגמא).

Decimal	Sign	Exponent	Mantissa
9.0	0	+3	1.001
-51.625	1	-5	1.100///0/
0.21875	0	. 3	1.11

2/2

ז. להלן משתנים של תכנית אסמבלי, המכילים ערכים בשיטת הנקודה הצפה. תרגם את הערכים לבסיס 10 בייצוג <u>ללא חזקה</u> (ראה דוגמא). <u>תזכורת</u>: בשיטת הנקודה הצפה במחשב, שדה החזקה הוא מספר ללא סימן הכולל bias.

float1 dd 0c1180000h

-9.5 -<u>52.675</u>

float2 dd 0c2568000h

59375

double1 dq 3fe3000000000000h

שאלה מס' <u>2</u> (25 נקודות)

שפת המכונה של המעבד x86 כוללת פקודות להזזה וסיבוב של אופרנד באורך 1, 2 או 4 בתים. אין בשפת המכונה פקודות להזזה וסיבוב של רצף בתים באורך אחר.

שאלה זו עוסקת בהזזה וסיבוב של רצף (מערך) בתים בזיכרון, באורך $1 \le n$ כלשהו. המגבלה היחידה על האורך $1 \le n$ הבתים כולו חייב להיכלל בסגמנט אחד.

בכל הסעיפים בשאלה זו אנו מניחים כי האורך ת הוא חוקי, ואין צורך לבדוק זאת.

א. להלן השגרה בשפת shr_long1 שכותרתה (בשפת C) היא:

```
void shr_long1(char * byteSequence, unsigned int n);
```

הפרמטר byteSequence הוא מצביע (היסט) לרצף של <u>בתים</u> שנמצא בסגמנט ה- data. הפרמטר מ הוא אורך הרצף (1≤n). השגרה <u>מזיזה ימינה בביט אחד</u> את רצף הבתים, בדומה לפקודת המכונה shr. עם החזרה מהשגרה, הדגל cf מכיל את הביט שיצא מקצהו הימני של הרצף. אין חשיבות למצב שאר דגלי התנאי עם החזרה מהשגרה.

```
shr_long1 proc
              push bp
              mov bp, sp
              push bx
              push cx
              mov bx, [bp+4]
              mov cx, [bp+6]
              clc
   loop_shr: rcr byte ptr [bx],1
              inc bx
11
              loop loop_shr
12
              pop cx
13
              pop
                   bx
              pop
15
              endp
```

יב. מה תבצע השגרה אם נחליף את שורה 8 בצמד הפקודות הבא?

8.1 sar byte ptr [bx],1
8.2 rcl byte ptr [bx],1

(5.0 lb bra 5: CF 8 0:35

D50 85 ka SAD 1862 in avan po in

נשנה את הגדרת הפרמטר byteSequence, כך שמעתה יהיה זה היסט לרצף בתים. בתוך הסגמנט אליו מצביע האוגד es (במקום ds בהגדרה המקורית).

עליך להכניס שינויים בקוד השגרה shr_long1 כדי להתאימה להגדרה החדשה. רשום את מספרי השורות בשגרה שיש לשנות, ואת קוד האסמבלי החדש של שורות אלה. בצע שנויים הכרחיים בלבד.

קוד אסמבלי חדש

1100 Sx. 62: [6p=4.

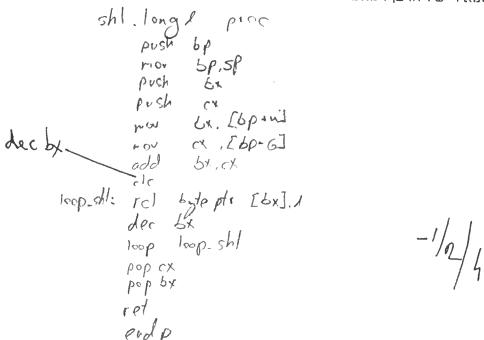
-|/|

6

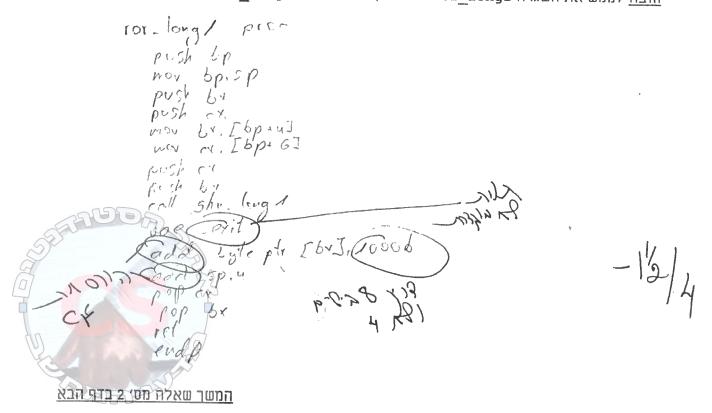
המשך שאלה מס' 2 בדף הבא

שאלה מס 2 (המשך)

כתוב בשפת אסמבלי שגרה בשם shl_long1, שמקבלת פרמטרים זהים לאלו של השגרה shr_long1 מסעיף א' (<u>ללא</u> השינוי בתת סעיף iii). השגרה shl_long1 <u>מזיזה שמאלה בביט אחד</u> את רצף הבתים שהועבר כפרמטר, בדומה לפקודת המכונה shl. עם החזרה מהשגרה, הדגל cf מכיל את הביט שיצא מקצהו השמאלי של הרצף. מותר לשגרה להרוס את שאר דגלי התנאי.



ג. כתוב בשפת אסמבלי שגרה בשם ror_long1, שמקבלת פרמטרים זהים לאלו של השגרה shr_long1 מסעיף א' (ללא השינוי בתת טעיף iii). השגרה shr_long1 מסובבת ימינה בביט אחד את רצף הבתים שהועבר כפרמטר, בדומה לפקודת המכונה ror. עם החזרה מהשגרה, הדגל cf מכיל את הביט שעבר מקצהו הימני של הרצף אל קצהו השמאלי. מותר לשגרה להרום את שאר דגלי התנאי. מובה לממש את השגרה ror_long1 בעזרת קריאה לשגרה לשגרה.



שאלה מס 2 (המשך)

ד. כתוב בשפת אסמבלי שגרה בשם shift_long שכותרתה בשפת C היא:

void shift_long(char * byteSequence, unsigned int n, int k);

שני הפרמטרים הראשונים של השגרה מאורה באונד זהים לפרמטרים של השגרה באוני הפרמטרים של השגרה גורה אוני הפרמטר א' (ללא השינוי בתת סעיף בא'). הפרמטר א הוא מספר בשיטת המשלים ל- 2. הנח כי $\mathbf{k} \neq \mathbf{0}$.

השגרה shift_long מבצעת כדלקמן: אם k>0, רצף הבתים שהועבר כפרמטר מוזז ימינה ב- k ביטים. אם k<0, רצף הבתים מוזז שמאלה ב- k- ביטים. עם החזרה מהשגרה, הדגל cf מכיל את הביט האחרון שיצא מקצה הרצף (מימין או משמאל, לפי המקרה). מותר לשגרה להרוס את שאר דגלי התנאי.

<u>רמז</u>: השתמש בקריאות לשגרות מסעיפים קודמים של שאלה זו.

public skitt-long shift-long piece

rush sp

rush sp

rush bp.sp

rush bx

push cx 100 6x. [6p.4] nov 14, [6p= 8] המשך שאלה מס' 2 בדף הבא

שאלה מס 2 (המשך)

להלן השגרה shr_long8, שמקבלת פרמטרים זהים לאלו של השגרה shr_long1 מסעיף א' (<u>ללא</u> השינוי בתת סעיף בבו). השגרה shr_long8 <u>מזיזה ימינה ב-8 ביטים</u> את רצף הבתים שמועבר כפרמטר. עם החזרה מהשגרה, הדגל cf מכיל את הביט <u>האחרון</u> שיצא מקצהו הימני של הרצף. אין חשיבות למצב שאר דגלי התנאי עם החזרה מהשגרה.

1	shr_long8	proc		
2	_	push	bp	
3		mov	bp,sp	
4		push	8	
5		push	word ptr	[bp+6]
6		push	word ptr	
7		call	shift_lor	
8		mov	sp,bp	-
9		pop	bp	
10		ret	-	
11		ando		

יי מה עושה שורה 28 מתצה היישנית למצפה הקומם ללא שיני ה א

בבי לאפשר קריאה לשגרה shr_long8 הן משפת C והן משפת אסמבלי באותו שם בדיוק, בדי לאפשר קריאה לשגרה shr_long8 הן משפת לפני שורה מס' 1. רשום שתי שורות אלה. יש להוסיף לקוד האסמבלי של השגרה שתי שורות, לפני שורה מס' 1. רשום שתי שורות אלה.

son is ox shr long 8:

בבב. השגרה long8 אינה יעילה, וניתן לכתוב שגרה יעילה יותר באופן משמעותי. נמק טענה זו. באוב באופן משמעותי. נמק טענה זו.

an possess a sursans events and all such as

Shr. long 8 proc

prof. 50 pro

http://cs.haifa.ac.il/students/

<u>שאלה מט' 3</u> (25 נקודות)

-3/3

להלן הגדרת המאקרו xtosr המקבל פרמטר אחד שהוא אוגר רב תכליתי בכל רוחב אפשרי. המאקרו מחליף בין תוכן האוגר המועבר כפרמטר לבין תוכן ראש המחסנית.

> xtosr macro reg push bp MOA bp,sp xchg reg, [bp+2] pop bp endm

האם המאקרו xtosx עובד נכון כאשר הפרמטר המועבר למאקרו הוא האוגר pp. הסבר.

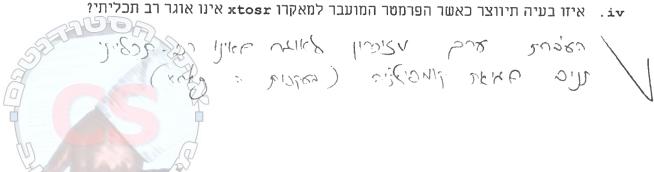
27 . OURS C 2001 25 St. - 901 CUSO . 19

ii. הגדר מאקרו בשם xtosbp ללא פרמטרים, אשר מחליף בין תוכן האוגר p גבין המילה. שבראש המחסנית. <u>אסור</u> למאקרו לשנות את הדגלים. אפשר להשתמש בקריאה למאקרו xtosr.

angalic Boxy 5p < 0x

. החבר מאקרו xtosr היווצר כאשר הפרמטר המועבר למאקרו sp? הוא האוגר גיווב. איזו בעיה עלולה להיווצר כאשר

Dilarno 610,60 logi culing circle ily dice (4)



המשך שאלה מסיצבדף הבא

שאלה מס' 3 (המשך)

ב. ז. הגדר מאקרו בשם isDenorm שכותרתו נתונה להלן. הפרמטר float הוא מספר ממשי בייצוג סטנדרטי בנקודה צפה ברוחב 32 ביטים. הפרמטר יכול להיזת אוגר רב תכליתי של המעבד X86, או ערך בזיכרון בכל שיטת מעון אפשרית.
 המאקרו בודק האם המספר float הוא בלתי מנורמל. אם המספר בלתי מנורמל, המאקרו מדליק את הדגל cf באוגר הדגלים, ואחרת המאקרו מכבה את הדגל cf.
 אח הדגל franca בפקודות של המעבד המתמטי. מותר למאקרו להרום את שאר דגלי התנאי.

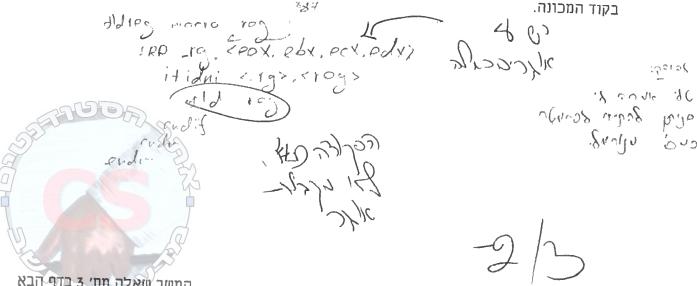
isDenorm macro float

local b

local plocal office of office of the open of th

ii. הגדר מאקרו בשם Denorm שכותרתו נתונה להלן. הפרמטר float זהה לפרמטר של המאקרו isDenorm, ואילו הפרמטר dest זהה לאופרנד של פקודת מכונה להסתעפות מותנית. המאקרו מבצע הסתעפות מותנית לכתובת dest אם ורק אם המספר float בלתי מנורמל. חובה להשתמש בקריאה למאקרו isDenorm. <u>אטור</u> למאקרו לשנות את הדגלים.

ג. הגדר מאקרו בשם fldreg, המקבל פרמטר אחד שהוא אוגר רב תכליתי ברוחב 32 ביטים במעבד X86. המאקרו טוען את תוכן האוגר שמועבר כפרמטר לתוך האוגר (0) של המעבד המתמטי. אם הפרמטר אינו אחד האוגרים הרב תכליתיים ברוחב 32 ביטים, המאקרו אינו פורש אף פקודה



שאלה מס׳ 3 (המשך)

ד. להלן הגדרת המאקרו sum. המאקרו מקבל לכל היותר ששה,פרמטרים, כאשר כל פרמטר הוא בגודל מילה ובכל שיטת מעון. המאקרו מחשב את סכום כל הפרמטרים ומציב את התוצאה באוגר ax.

```
1 sum macro n1,n2,n3,n4,n5,n6
 2
   k = 0
 3
    irp n,<&n1,&n2,&n3,&n4,&n5,&n6>
 4
      ifidni <&n>,<ax>
 5
        k = k + 1
 6
      endif
 7
    endm
    if k eq 0
 8
 9
      xor ax,ax
10
    elseif k ge 2
11
      push dx
12
      mov dx, k
13
      imul dx
14
      pop
           dx
15
    endif
16
    irp n,<&n1,&n2,&n3,&n4,&n5,&n6>
17
      ifnb <&n>
18
        ifdifi <&n>,<ax>
19
          add ax,n
20
        endif
21
      endif
22
    endm
23 endm
```

להלן מספר דוגמאות של קריאות למאקרו sum. עבור כל דוגמא, רשום את הקוד בשפת אסמבלי שמתקבל מפרישת הקריאה למאקרו. יש לרשום רק שורות שיוצרות <u>קוד מכונה</u> (אין לרשום את ההנחיות לאסמבלי מותנה, וכד').

sum bx,y,[bp+2],,,4	sum bx,ax,y	sum ax,bx,ax,ax	sum bx	sum ax
add cx, bx	add ar. by	pop dx	xop.or.ox	OI\$
add ar. [bp. ?]	V	add ar ox		
ASTOR		J	7	J

<u>שאלה מס' 3</u> (המשך)

מה תפקידו של משתנה האסמבלי \mathbf{k} בהגדרת המאקרו שבדף הקודם?

10.11 (1450 & 2011 Ox 0.1120 240 / 1,50y

iii. האם אפשר להחליף בין השורות 17 ו-18 בהגדרת המאקרו wwa? נמק.

4 big was: mic. oil 12 vo or 1494; 2004 20 big by

יב. תן דוגמא לקריאה למאקרן שעם אשר תגרום לשגיאה בזמן אסמבלי. הסבר מהי השגיאה. $\lambda = \frac{1}{2} + \frac$

יסה יקרה אם נקרא למאקרו sum ללא אף פרמטרי. v

100 04.0x 000

.vi בגמר ביצוע המאקרו sum, האם ערכו של הדגל zf מתאים ממיד לתוכנו של האוגר.

10x 5 -07 10 18)



שאלה מס׳ 4 (25 נקודות)

א. כידוע, שגרת השרות של פסיקה מס' 21h מספקת מספר תתי שירות שונים לביצוע קלט של תו בודד מן המקלדת. בפרט, תת שרות מס' 1 מבצע echo של התו שנקלט, ואילו תת שירות מס' 8 אינו מבצע

ברצוננו לשנות את שגרת הפסיקה 21h כך שתתי השירות 1 ו, 8 יתחלפו ביניהם. כלומר, לאחר השינוי, קריאה לתת שירות 1 תגרום לביצוע קלט ללא echo, ואילו קריאה לתת שירות 8 תגרום לבצוע קלט עם echo. אין שינוי בתתי שירות אחרים של פסיקה 21h.

כתוב להלן שגרת שירות חדשה לפסיקה 21h שתממש את השינוי המתואר לעיל. השגרה תתחיל בתווית newInt21. <u>הנח</u> כי וקטור הפסיקה של שגרת השרות הקודמת נשמר במשתנה oldInt21 בסגמנט הקוד של שגרת השירות החדשה.

oldInt21 dd?

newInt21:

profit

(Mp. onl,)

(Mp. onl,



שאלה מס' 4 (המשך)

ב. ב. כתוב שגרת שרות חדשה לפסיקת שרותי השעון (פסיקה מס Ich), אשר תבצע כדלקמן: כל 5 דקות בדיוק תושמע סדרה של 10 צפצופים, במרווחים של 2 שניות בין צפצוף לצפצוף. <u>שים לב</u>: משך הזמן בו מושמעת סדרת צפצופים <u>נכלל</u> ב- 5 הדקות שעוברות עד לסדרה הבאה.

<u>תזכורת</u>: צפצוף נוצר על ידי הדפסת קוד אסקי 7 למסך.

שגרת השרות החדשה תתחיל בתווית newInt1c. <u>הנח</u> כי וקטור הפסיקה של שגרת השרות הקודמת נשמר במשתנה oldInt1c בסגמנט הקוד של שגרת השרות החדשה. <u>הקפד</u> על כללי התכנות המקובלים בשגרת שרותי השעון.

```
.386
. code
oldInt1c dd?
public 900 0 300 3000 1000 30
count do s
    Pust ox
    push dx
                        2 beeper 010'le 50'le
    odd v sers, 55
    orb existic Source 5-276 Win 012,5
     THE MERCE BUE OLO
                             inproved,
13: vou 11,7
                              jup 11
                                       -4/9
     pop de
     pup ax
         statist 10
```

ii. האם, לדעתך, הצפצופים בכל סדרת צפצופים אכן יישמעו בקצב אחיד ומדוי<mark>ק? הס</mark>בר.

356 26 8.6. Clopodo de 85.0 x -5/1

המשך שאלה מס' 4 בדף הבא

שאלה מס׳ 4 (המשך)

ג. לצרכי דיבוג של תכנית המשתמש, ברצוננו לבנות מנגנון שיופעל עם כל ביצוע של פקודת המכונה add, ויבדוק האם פעולת החיבור גרמה לגלישה. במידה והייתה גלישה, תודפס מייד על המסך הודעת אזהרה כדלקמן: Warning: Overflow on add הערה: עבור כל מקרה של גלישה בפקודת add תודפס הודעה נפרדת.

ניתן לממש את המנגנון המתואר לעיל על ידי שגרת שרות לפסיקת single-step (פסיקה מס' 1). כדי להפעיל מנגנון זה על קטעי קוד נבחרים בתכנית המשתמש, ניתן, לדוגמא, להשתמש במאקרוים traceOff 1- traceOn

> עליך כתוב שגרת שרות לפסיקה מס׳ 1 שתממש את המנגנון המתואר לעיל. אין צורך לבצע שרשור לשגרת הפסיקה הקודמת.

הנת שה- opcode של פקודת add (כלומר הבית הראשון בקוד המכונה) הוא ערך בין 0 ל- 5 (כולל).

<u>רמז</u>: בעת שנוצרת פסיקת single-step נשמרת במחסנית כתובת הפקודה <u>הבאה,</u> ולא כתובת הפקודה שהסתיימה כרגע. על שגרת הפסיקה "לזכור" את הפקודה הבאה עד לאחר ביצועה.

