

גירסה 1.00 - 7.12.2002

VAX11 - סיכום נקודות

מסמך זה הורד מהאתר <http://underwar.livedns.co.il>.
אין להפיץ מסמך זה במדיה כלשהי, ללא אישור מפורש מאת המחבר.
מחבר המסמך איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע
המופיע במסמך, וכן לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת,
המחבר עשה את מירב המאמצים כדי לספק את המידע המדויק והמלא ביותר.

כל הזכויות שמורות לניר אדר

Nir Adar

Email: underwar@hotmail.com

Home Page: <http://underwar.livedns.co.il>

אנא שלחו תיקונים והערות אל המחבר.

אוגרים

ל-VAX11 יש אוגרים מיוחדים למספרים צפים, וכן 16 אוגרי 32 bit למספרים שלמים, המסומנים r0...r15.
 הרגיסטרים r0..r11 מיועדים לשימוש כללי, אך עם זאת ל-r0..r3 תפקידים מיוחדים בפונקציות ספריה רבות.
 r12..r15 הם רגיסטרי בקרה יעודיים, להם כינויים נוספים מיוחדים.

רגיסטר	שם מיוחד	משמעות
r12	ap	argument pointer
r13	fp	frame pointer
r14	sp	stack pointer
r15	pc	program counter

האוגרים משמשים לצבירת ערכי ביניים וכמצביעים/הסטים לזיכרון.

בנוסף קיימים דגלי PSW לתמיכה בהסתעפויות מותנות.

התקני קלט/פלט

Vax-11 תומך במגוון התקני קלט פלט בעלי גישה אקראית וסדרתית.
 sim תומך אך ורק בהתקן קלט סדרתי יחיד ובהתקן פלט סדרתי יחיד.
 - התקן הקלט יכול להיות מקשים (KBD) או שם קובץ קלט כלשהו.
 - התקן הפלט יכול להיות מסך פשוט (CRT) או שם קובץ פלט כלשהו.

אסמבלי - כללי

הערות בשפת אסמבלי מתחילות ב-# ונמשכות עד סוף אותה שורה. הערה יכולה להתחיל בתחילת שורה או באמצע.

התוכנית מחולקת לשני חלקים: הוראות (text) ונתונים (data). הוראות ה-text מקדימות את הוראות ה-data. אנו מזהים את חלק ההוראות על ידי ההוראה ".text" ואת הנתונים על ידי ".data".
 ניתן לקבוע שמות סמליים לכתובות בזיכרון, בין אם הן הוראות ובין אם הן נתונים, על ידי הגדרת סמל, ואחריה נקודותיים. סמלים מציינים כתובות בזיכרון.

סמלים והוראות כגון ".text" הם הוראות למהדר, שאינן מתורגמות לשפת מכונה.
 כל תוכנית / שיגרה חייבת לפתוח במילת מסיכה לאגירת אוגרים. מילה הינה 16 ביט, ולכן כל אוגר מיוצג על ידי סיבית מסוימת במילה. האוגרים המסומנים נשמרים/משוחזרים אוטומטית בתוך/מתוך המחסנית.

כל שורת אסמבלי עשויה לפתוח בתווית ובעקבותיה הוראה או פסאדו הוראה.
 פסאדו הוראות פותחות תמיד ב-.

פסאדו הוראות

.word <value>

הצבת המילה <value> למקום בו נמצאת הפסאדו הוראה.

.long <value>

הצבת המילה הכפולה <value> למקום בו נמצאת הפסאדו הוראה.

פונקציות ספרייה

פונקציות ספרייה סטנדרטיות של אסמבלי פותחות ב-. כגון "getchar".

getchar

דוגמא לקריאה:

calls \$0, .getchar

פעולה: תו נקרא והתוצאה נשמרת ב-r0. לפונקציה מועברים 0 פרמטרים. זיהוי סוף קובץ/קלט נעשה על ידי קבלת תו שלילי.

יחידות המידע ב-VAX-11

בית בודד הוא בגודל 8 ביטים. יחידות המידע השונות הינן:

Information Unit	Size (In Bytes)
Byte	1
Word	2
Long	4
Quad	8

הבית הראשון הינו ה-LSB. מבחינת ה-ALU, כל הטיפוסים הם בעלי סימן.

מבנה הוראת אסמבלי

א. מבנה כללי:

- לפני הוראת אסמבלי עשויה להיות תוית עם '!'.
- להוראת אסמבלי יש opcode (פעולה) ומספר operands (נפעלים/אופרנדים).
- הבית הראשון בהוראה הוא ה-opcode והוא קובע את מספר האופרנדים.
- כל אופרנד יכול להשתמש באחת מתוך 13 שיטות המיעון הקיימות ב-VAX11.

ב. דוגמת MOV:

MOV

Purpose:

move a scalar quantity

Format:

opcode src.rx, dst.wx

Operation:

dst <- src

Condition Codes:

N <- dst LSS 0

Z <- EQL 0

V <- 0

C <- 0

Exceptions:

None (integer); Reserved operand (floating point)

Opcodes:

90 MOVB Move Byte

B0 MOVW Move Word

D0 MOVL Move Long

7D MOVQ Move Quad

50 MOVF Move Floating

70 MOVD Move Double

Description:

The destination operand is replaced by the source operand. The source operand is unaffected.

Notes:

1. On a floating reserved operand fault, the destination operand is unaffected and the condition codes are unpredictable.
2. Unlike the PDP-11, but like the other VAX-11 instructions, MOVB and MOVW do not modify the high order bytes of a register destination. Refer to the MOVZxL and CVTxL instructions to update the full register contents.

נשים לב לתיאור הפורמט.

מקור	src
יעד	dst
קריאה	r
הסוג מצויין ב-opcode	x
עדכון (modify)	m

עקרונות הקריאה לשגרות

א. קריאה לשיגרה נעשית על ידי דחיפת נתונים למחסנית, ולאחר מכאן שימוש בפקודה calls כדי לקרוא לשגרה.

- דחיפת הפרמטרים חייבת להעשות לפני הקריאה לשיגרה.
- הפרמטרים נדחפים למחסנית בסדר הפוך.
- הקריאה ב-calls כוללת כאופרנדים את כתובת השיגרה ואת מספר הפרמטרים המועברים אל השיגרה.
- מספר הפרמטרים נדחף על ידי calls אל המחסנית.
- calls גם מעתיק את sp ל-ap באופן כזה שיצביע אל מספר הפרמטרים.
- נשים לב כי המחסנית ב-VAX11 מתפשטת בכיוון שלילי.

ב. הפרמטרים חייבים להיות כולם מסוג long.

- אופי הפרמטרים חייב להיות מוסכם בין השיגרה לבין המשתמש בה.
- אפשרי שמספר הפרמטרים המועבר בפועל ישתנה בין קריאה לקריאה. למשל: הפונקציות printf, scanf.

שיטות המיעון של VAX11

שיטת המיעון	סימון בשפה	מימוש במכונה	
1	register	אוגר r	אוגר+50
2	register deferred	(אוגר r)	אוגר+60
3	auto increment	(אוגר r)+	אוגר+80
4	auto decrement	(אוגר r)-	אוגר+70
5	auto increment deferred	(אוגר r)+*	אוגר+90
6a	displacement	הסט(r)הסט	אוגר+A0 הסט B
6b			אוגר+C0 הסט W
6c			אוגר+E0 הסט L
7a	displacement deferred	(אוגר r)הסט*	אוגר+B0 הסט B
7b			אוגר+D0 הסט W
7c			אוגר+F0 הסט L
8a	index	[אינדקס r](בסיס r)	אינדקס+40 על פי אופי כתובת הבסיס
8b			[אינדקס r]+ (בסיס r)
8c			[אינדקס r]-(בסיס r)
8d			[אינדקס r]+*(בסיס r)
8e			[אינדקס r](בסיס r)הסט
8f			[אינדקס r](בסיס r)הסט*
9	literal	ערך \$(0..63)	ערך+00
10	immediate	ערך \$	ערך (מספר בתים לפי הגודל) + (PC) שיטה 3
11	absolute	כתובת \$*	כתובת L + (PC)* שיטה 5
12	relative	כתובת	הסט L (PC) שיטה 6
13	relative deferred	כתובת *	הסט L (PC) שיטה 7

הסתעפויות מותנות

קפיצות בעלות סימן:

Operation	Mnemonic	Branch Condition
Branch on Equal	BEQL	Z=1
Branch on Not Equal	BNEQ	Z=0
Branch on Less Than	BLSS	N=1
Branch on Less Than or Equal	BLEQ	(N or Z) = 1
Branch on Greater Than	BGTR	(N or Z) = 0
Branch on Greater Than or Equal	BGEQ	N=0

קפיצות ללא סימן:

Operation	Mnemonic	Branch Condition
Branch on Equal Unsigned	BEQLU	Z=1
Branch on Not Equal Unsigned	BNEQU	Z=0
Branch on Less Than Unsigned	BLSSU	C=1
Branch on Less Than or Equal Unsigned	BLEQU	(C or Z) = 1
Branch on Greater Than Unsigned	BGTRU	(C or Z) = 0
Branch on Greater Than or Equal Unsigned	BGEQU	C=0

אופני קריאה לשגרות**א. קריאה לשגרה על ידי jsb.**

תחביר:

שם_שגרה JSB

1. הרגיסטר PC נדחף אל המחסנית.
2. החזרה מהשיגרה מתבצעת על ידי הפקודה .rsb.

ב. קריאה לשגרה על ידי calls.

תחביר:

שם_שגרה, מספר פרמטרים CALLS

0. דחיפת הפרמטרים על ידי הקורא.
1. דחיפת מספר הפרמטרים (N) - המספר המקסימלי המותר - 255.
2. שתי סיביות lsb של SPA <= sp .temp.
3. איפוסם גורר מילוי של 0-3 בתים.
4. דחיפת רגיסטרים r0-r11 על פי המסכה.
5. דחיפת הרגיסטרים ap, fp, pc (כתובת חזרה).
6. איפוס דגלי PSW.
7. דחיפת PSW + שונות.
8. דחיפת מילת אפסים לציון סוף המסגרת.
9. עדכון fp <= sp .temp <= ap.
10. כתובת השגרה + 2 <= PC.

ג. קריאה לשגרה על ידי callg.

תחביר:

שם_שגרה, קטע גלובלי CALLG

- משתמש בקטע זיכרון גלובלי להעברת פרמטרים. מבנה קטע הזיכרון הוא כמו במחסנית של CALLS.
- callg אינו מתאים כל כך לשגרה רקורסיבית כי הוא מחייב מקטע פרמטרים לכל מופע של הרקורסיה בעזרת זיכרון דינמי.
- מבנה המחסנית אחרי CALLG הוא כלהלן:
 0. הכנת רשימת פרמטרים בזכרון גלובלי.
 1. מספר הפרמטרים N בראש המחסנית.
 2. שתי סיביות lsb של SPA <= sp .temp.
 3. איפוסם גורר מילוי של 0-3 בתים.
 4. דחיפת רגיסטרים r0-r11 על פי המסכה.
 5. דחיפת הרגיסטרים ap, fp, pc (כתובת חזרה).
 6. איפוס דגלי PSW.
 7. דחיפת PSW + שונות.
 8. דחיפת מילת אפסים לציון סוף המסגרת.
 9. עדכון fp <= sp .temp <= ap. כתובת הקטע הגלובלי <= ap.
 10. כתובת השגרה + 2 <= PC.

ד. חזרה על ידי ret

1. $sp \leq fp + 4$
2. $temp \leq PSW + \text{שונות}$
3. שחזור הרגיסטרים pc, fp, ap
4. שחזור רגיסטרים r0-r11 על פי המסכה.
5. שחזור הרגיסטר sp לפי SPA מתוך temp.
6. שחזור PSW מתוך temp (מצב לפני איפוס).
7. דלג על המילוי לפני SPA.
8. אם $S=1$ אזי קרא N ודלג על הפרמטרים. (צעד זה מחייב שכל פרמטר יתפוס בדיוק מילה).

ה. משתנים מקומיים

- בכדי להקצות N בתים למשתנים מקוריים נקטין את SP ב-N.
אם לא נעשה זאת, השגרה לא תוכל לקרוא לשגרות אחרות.
לאחר שהקטנו את SP, ניתן לפנות את המשתנים בעזרת היסטים שליליים ביחס ל- fp : $-4(fp)$ -
למילה הראשונה, $-8(fp)$ עבור המילה השניה וכו.