

יש קשר בין מה שניראה בקוד המכונה והפעולה, אולם הקשר נראה די קלוש (או מקרי בהחלט). כפי הנראה משהו ברמת הביטים משתנה ולכן התוצאה בהקסה ממש לא עקבית.

ההוראה מורכבת מעד 6 בתים. הבית הראשון משמאל הוא תמיד הפקודה (operation). אם הפקודה מתיחסת לנתונים, נוסף גם בית של כתובת, אם צריך גם לציין מיקום בתוך הכתובת (displacement) או אופרנדים נוספים הם באים בהמשך הבתים מימין.

בהוראה של העברת נתונים (כמו mov), בבית השני משמאל, ביט הימני (ביט 0) מסמל אם אנו מעבירים בית אחד (ערך 0) או מילה שלמה (ערך 1) נקרא ביט W. הביט השני מימין (ביט 1) נקרא ביט D ומסמל כיוון (direction) = מיהו המקור ומיהו יעד של הנתונים. (להוראות שאינן מעבירות נתונים אין את הביטים האלה).

לדוגמה פעולת mov שמעבירה נתונים בין רגיסטרים יכול להיות קוד כמו:

1000 10DW

זה גם מסביר את השוני בייצוג ההקסה בין הפקודות :

mov AX, DX ו mov DX, AX

הבית הבא של חלק מהפקודות קשור לתצורת הכתובת והוא בפורמט (כל אות מיצגת ביט):

mm rrr sss

על מנת לתארו, צירפתי טבלה למטה. mm מציינים את צורת התירגום של הביטים sss ואת מספר הבתים שיעקבו עם ההזזה הדרושה (displacement). rrr מציינים את האוגר של המעבד על פי הטבלה המצורפת (2) העמודות הימניות מציינות W=0 חצי אוגר כי רק בית אחד, W=1 מילה שלמה אוגר מלא).

אם ביט הכיוון הוא אפס, rrr הוא רגיסטר המקור ו sss הוא היעד (r/m), אם הביט הוא אחד < ההפך.

אם mm=11, הביטים sss, rrr מתייחסים לרגיסטרים. אם mm=00 ההתייחסות לזיכרון היא למה שמצוין בעמודה השנייה של הטבלה (mem) ללא בתי הזזה, מלבד כאשר sss=110 שבמקרה זה 16 הביטים של אופסט מצויינים על ידי 2 הבתים הבאים. כברירת מחדל אופסט זה ביחס לרגיסטר של סגמנט DS. כאשר mm=01 ישנו אופסט של 8 ביטים וכאשר mm=10 האופסט הוא של 16 ביט. עבור sss=110 הכתובת היא: BP+DISP (לא בטבלה).

הקודים של הפקודה גם תלויים בגירסה של האסמבלר (מיקרוסופט, בורלנד או אחר), למשל הפקודה:

mov AH,AL

יכולה להיות מתורגמת ל: 88 E0 או 8A C4 או בגירסה שהשתמשנו ל 8A E0

r/m	mem	w=0	w=1
000	BX+SI	AL	AX
001	BX+DI	CL	CX
010	BP+SI	DL	DX
011	BP+DI	BL	BX
100	SI	AH	SP
101	DI	CH	BP
110	ABS	DH	SI
111	BX	BH	DI