

פסאודו קוד:

סיכום כללי:

הגדרה:

פקודות אלו אנו Pseudo Instruction נועדו לאגד מספר פקודות מכונה על מנת לפשט ולהקל על תהליך הכתיבה של קוד. נשתמש בהן באופן שוטף במהלך כתיבה של קוד מבלי להתחשב בעובדה שהן לא מיושמות בצורה ישירה במעבד אלא כמספר פקודות.

הבדלים עקרוניים בין פקודות פסאודו ופונקציות באסמבלי:

(1) טבע הפקודות ומטרתן:

פקודות פסאודו אינן פונקציות במהותן אלא פירוש מילולי שנועד לפשט את הקוד. בפועל פקודת פסאודו מתבצעת ע"י סדרה של פקודות שה-CPU יודע לבצע. פונקציות לעומת זאת מורכבות מסט של פקודות אשר ה-CPU יודע לבצע ומוגדר עבורן.

(2) ייצוג:

פקודות פסאודו מיוצגות ע"י שורת קוד בודדת עם פורמט כתיבה ייעודי להן בעוד שפונקציות מוגדרות ע"י תוויות המצביעות על כתובות שונות בזיכרון וכן פעולות שונות שעל המעבד לבצע.

(3) שימושיות:

פקודות פסאודו נועדו בעיקר כדי לאפשר למתכנתים לכתוב קוד בשפה עילית ככל הניתן מתוך מטרה לפשט אותו ואת המורכבות שלו. פונקציות נועדו להפוך את הקוד למודולרי ומטרתן היא לאפשר לעקוב אחר התוכנה והשלבים שלה, לחלק את מבנה התוכנה לרמות ועוד.

פקודת move:

```
move $t0, $s3 # $t0 gets the content of $s3: $t0 = $s3.
```

פקודה בסיסית ושכיחה ביותר שכל מטרתה היא 'להזיז' תוכן מאוגר אחד לאחר. בפועל הפקודה ממומשת ע"י פקודת add עם אוגר האפס כך:

```
add $t0, $s3, $zero
```

פקודת li (Load Immediate):

```
li $t0, 0x50    # $t0 gets the value 0x50
```

פקודה המאפשרת לטעון ערך קבוע מיידית לאוגר מבוקש.
בפעול הפקודה מבוצעת ע"י פקודת addi עם אוגר האפס כך:

```
addi $t0, $zero, 0x50
```

פקודת la (Load Address):

```
la $t0, LABEL    # $t0 gets the address of LABEL
```

נניח ו-LABEL נמצאת בכתובת 0x12345678, אז הפקודה la מתבצע באופן הבא:

```
lui $t0, 0x1234
```

```
ori $t0, $t0, 0x5678
```

פקודת blt (Branch on Less Than):

```
blt $t0, $t1, LABEL    # if $t1 < $t0 then for to:  
                        # PC = PC + 4 + 4*LABEL
```

הפקודה משלבת שתי פקודות מוכרות והן slt ו-bne באופן הבא:

```
slt $at, $t0, $t1
```

```
bne $at, $zero, LABEL
```