מאיה שמעון

REGISTERS – אוגרים

אוגר : אוגר הנו תא זיכרון הנמצא ב– CPU . יעילותו היא בכך שהגישה אליו מהירה ביותר.

האוגרים מחולקים ל- 4 קבוצות:

א	Flags register	אוגר הדגלים.
ב	General purpose register	אוגרים כלליים: מידע, אינדקס פוינטרים...
ג	Instruction pointer register	מצביע על הפקודה הבאה לביצוע.
ד	Segment register	אוגרי הסגמנט.

טבלת האוגרים במעבדים 8086 ו- 386

386 מילה כפולה - 32 bit	8086 מילה - 16 bit	 בית - 8 bit
EAX EBX ECX EDX	AX BX CX DX	AH,AL BH,BL CH,CL DH,DL
ESI EDI EBP ESP EIP	SI DI BP SP IP	
EFLAGS	FLAGS	
מילה - 16 ביט CS DS SS ES FS GS	CS DS SS ES	

Flags Register – אוגר הדגלים

אוגר זה מיוחד במינו. סיבית באוגר זה היא דגל המקבל ערך 0 (כבוי) או 1 (דלוק).
חלק מהדגלים מאפשרים מעקב ובקרה, הם מקבלים את ערכם בהתאם לפעולה האריתמטית או לוגית האחרונה שהתבצעה. למשל, דגל ה- ZF ידלוק אם תוצאת הפעולה האחרונה היתה אפס (הפקודה: MOV לא משנה את דגלים אלו).
שאר הדגלים, תפקידם לאפשר שליטה על ה– CPU. למשל, מצב ה- DF משפיע על כיוון של פעולות מסוימות הקשורות למערכים. השם: FLAGS אינו מופיע בשום פקודת אסמבלר.

פרוט תכונות של דגלים עיקריים:

OF	Overflow	דולק כשתוצאת חיבור שני מספרים חיוביים (שליליים) יוצאת שלילית (חיובית) בשיטת המשלים ל- 2.
DF	Direction	קיימות פעולות על מערכים שכיוונן נקבע על פי מצב הדגל.
IF	Interrupt	כאשר הדגל דולק - מתאפשרות פסיקות חומרה, אחרת, מנגנון פסיקות החומרה מושבת.
TF	Trap	כאשר הדגל דולק - מתאפשרת פסיקה מס 1 אחרת היא אינה מתאפשרת.
SF	Sign	דולק כאשר תוצאת הפעולה האחרונה שלילית.
AF	Auxiliary carry	דגל נשא שעוזר לתרגם ערכים ל- ascii ו- (binary code decimal) bcd.
PF	Parity	דולק כשבתוצאת הפעולה האחרונה יש מס' זוגי של "1".
ZF	Zero	דולק כאשר תוצאת הפעולה האחרונה היא 0.
CF	Carry	דולק כאשר הפעולה האחרונה יצרה נשא (carry) או לווה (borrow). יש לו שימושים נוספים.

אוגרים כלליים - general purpose registers

אוגרי המידע

כל אחד מאוגרי המידע מסוגל להכיל כל נתון שאפשר לבטא באמצעות 16 ביט.
AX, BX, CX ,DX מתנהגים באופן זהה לגבי רוב הפקודות אך לכל אחד מהם יש תפקיד המייחד אותו, להלן כמה מהתפקידים המייחדים כל אחד מהאוגרים הנ"ל:

האוגר AX (Accumulator)

תומך בפעולות קלט ופלט ומשמש לפעולות כפל וחילוק.
ישנן פעולות אריתמטיות כגון: SUB או ADD המהירות יותר ב- AX מאשר באוגרים אחרים ולכן הוא מכונה: Accmulator (צובר).

האוגר BX (Base)

משמש כאוגר בסיס ליצירת היסט. ההיסט הוא בדרך כלל יחסית לסגמנט שכתובת ההתחלה שלו נמצאת ב- DS.

האוגר CX (Counter)

אוגר זה משמש בד"כ כמונה בלולאות (למשל בפקודת LOOP).

האוגר DX (Data)

תומך בפעולות קלט ופלט ומשמש לפעולות כפל וחילוק.

הערות:

1. יש אפשרות לעבוד עם חצי אוגר מידע (את שאר האוגרים לא ניתן לפצל).

את **AX** ניתן לחלק לשני חלקים: **AH** - (8 הסיביות המשמעותיות) ו- **AL** (8 הסיביות הפחות משמעותיות).
את **BX** ניתן לחלק לשני חלקים: **BH** - (8 הסיביות המשמעותיות) ו- **BL** (8 הסיביות הפחות משמעותיות).
את **CX** ניתן לחלק לשני חלקים: **CH** - (8 הסיביות המשמעותיות) ו- **CL** (8 הסיביות הפחות משמעותיות).
את **DX** ניתן לחלק לשני חלקים: **DH** - (8 הסיביות המשמעותיות) ו- **DL** (8 הסיביות הפחות משמעותיות).

2. במעבד 386 ניתן להתייחס בנפרד לכל אחד מהחלקים הבאים:

EAX (32 bit) ⊇ AX (16 bit) ⊇ AL (8 bit), AH (8bit)
EBX (32 bit) ⊇ BX (16 bit) ⊇ BL (8 bit), BH (8bit)
ECX (32 bit) ⊇ CX (16 bit) ⊇ CL (8 bit), CH (8bit)
EDX (32 bit) ⊇ DX (16 bit) ⊇ DL (8 bit), DH (8bit)

אוגרי אינדקס ומחוונים

האוגרים BP ו- SP (stack pointer & base pointer)

משמשים בעיקר לקריאה וכתיבה של ערכים משטח בזיכרון הקרוי מחסנית (stack) .
האוגר SP תמיד יצביע על ראש המחסנית והאוגר BP משמש כאוגר בסיס ליצירת היסט יחסית לסגמנט שכתובת ההתחלה שלו נמצאת ב- SS.

האוגרים SI ו- DI (destination index & source index)

אוגרי האינדקס משמשים בעיקר ליצירת היסט.
יעילותם גדולה בעיקר בטיפול במחרוזות, לשם כך קיימות פקודות מיוחדות הפועלות ישירות עליהן ומתוכננות כך ש- SI יפעל באזור הנתונים על סגמנט שכתובתו ב- DS ו- DI על סגמנט שכתובתו ב- ES.

המצביע על הפקודה הבאה לביצוע

IP - Instruction Pointer register

האוגר IP מכיל את הכתובת של הפקודה הבאה לביצוע. בכל ביצוע פקודה חדשה ה- IP גדל במספר הבתים שמכילה הפקודה הנ"ל או שמשתנה ערכו לערך חדש לגמרי כתוצאה מהפעלת פקודות הסתעפות כגון JMP או CALL.
השם IP אינו מופיע בשום פקודת אסמבלר.

מאיה שמעון

אוגרי סגמנט – Segment Registers

סגמנט: בלוק זיכרון בגודל של 64K.

אוגרי הסגמנט: הם אוגרים שתפקידם להכיל כתובת של תחילת סגמנט.

אוגר Code Segment - CS

אוגר זה מכיל את הכתובת של תחילת סגמנט הקוד המכיל את הפקודות לביצוע.

הזוג CS:IP מצביע על הפקודה הבאה לביצוע.

ההנחיה **.CODE** קובעת את התחלת ה– code segment ודואגת להציב כתובת התחלה זו ב- CS.

אוגר Data Segment - DS

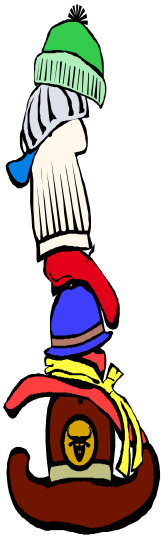
אוגר זה מכיל את הכתובת של תחילת סגמנט המידע המכיל נתונים כגון: משתנים סטטיים, משתנים גלובליים ומחרוזות קבועות.

ההנחיה **.DATA** קובעת את התחלת ה- data segment (סגמנט המידע) אך אינה דואגת להציב כתובת התחלה זו ב- DS ולכן על המתכנת לדאוג לאתחול ה- DS.

ה- @DATA הוא קבוע של מערכת ההפעלה המתעדכן (בזמן הקצאת המקום בזיכרון) בכתובת של התחלת סגמנט הנתונים.

מכיוון שאי אפשר להציב קבוע ישירות ל- DS נהוג לאתחל את DS בעזרת אוגר כללי:

```
MOV AX, @DATA
MOV DS, AX
```



אוגר Extra Segment - ES

אוגר המכיל כתובת של תחילת סגמנט כלשהו המיועד לשימושים שונים של התכנית.

בדרך כלל יעיל בטיפול במחרוזות וגם משמש כסגמנט יעד להעתקת בלוקים.

אוגר Stack Segment - SS

אוגר המכיל כתובת של תחילת סגמנט המחסנית.

כל ההוראות שעובדות עם אוגר SP (call, pop, push, ret) נעזרות בסגמנט זה.

אוגרי הסגמנט – מוצר הכרחי למיפוי הזיכרון במעבד 8086

מעבד 8086 מסוגל למפות 1MB של זיכרון. בכדי למפות 1MB של זיכרון, דרושים 20 ביט ($1_{MB} = 2^{20}$).

אבל כידוע 8086 משתמש באוגרים של 16 ביט המשמשים כמצביעים !

נשאלת השאלה:

כיצד בכל זאת מבטא ה - 8086 כתובת בעזרת אוגרים בני 16 ביט בלבד?

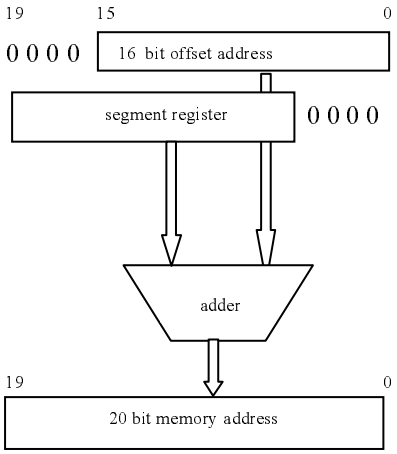
תשובה:

8086 משתמש בנוסחה של 2 חלקי זיכרון:

כל 16 ביט של offset משולב עם 16 ביט של אוגר סגמנט. והנוסחה לקבלת כתובת זיכרון אבסולוטית היא: **segment*16+offset**.

כלומר הזוג segment:offset של 8086 ממפה 1MB זיכרון.

FIGURE 1. How memory addresses are formed by an X86 CPU in real mode.



Example:

offset address = 1234h

seg reg = 5678h

```
                 01234h
                + 56780h
20 bit address    579B4h
```