2.5.1 General Structure of the i386

The i386 also follows the general structure of other microprocessors, as shown in Figure 2.3. Besides the control unit it has several registers. Additionally, several registers for memory management are available which are, however, important only in protected mode. Figure 2.5 shows all the implemented registers.

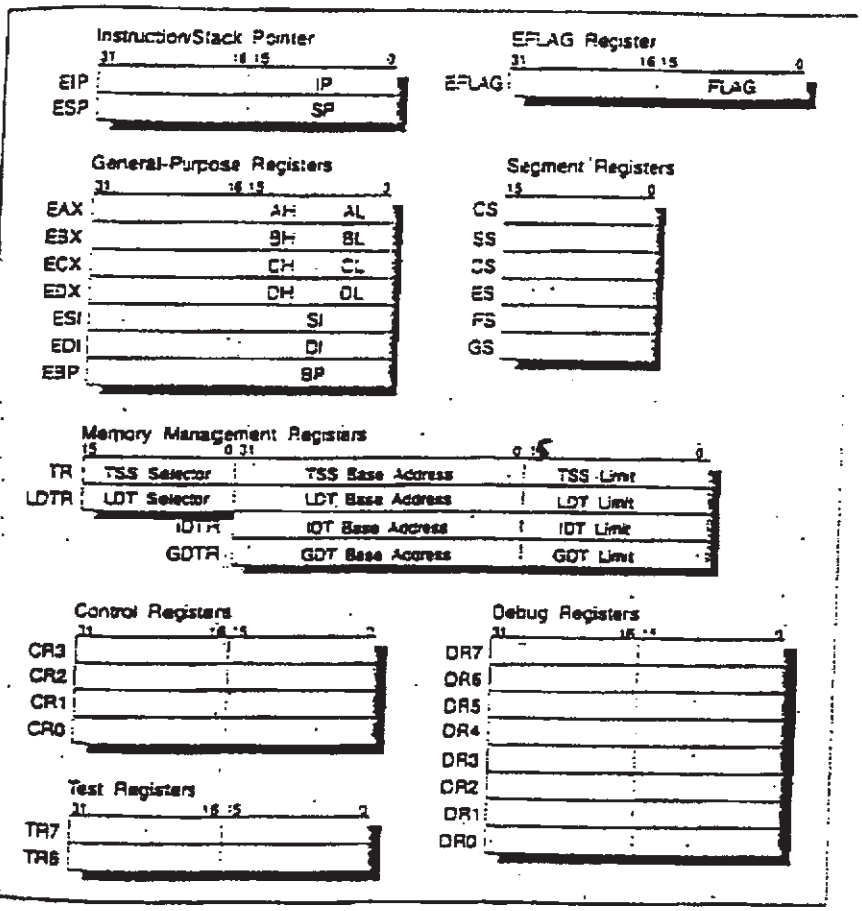


Figure 2.5: i386 processor registers. The general-purpose registers of the i386 are 32 bits wide, but can also be accessed as 16-bit or 8-bit registers. Additionally, there are six segment registers and an instruction pointer, which addresses the instruction to be executed next, as well as a flag register storing the current processor status flags and various control and test registers.

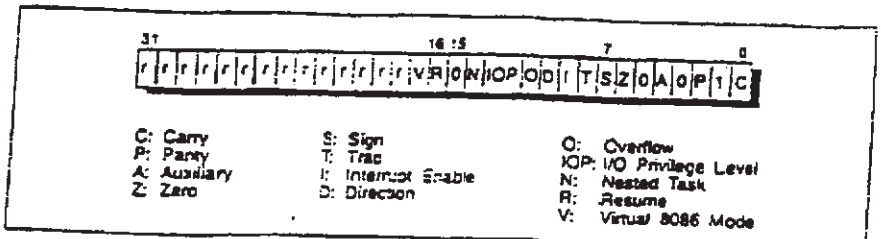
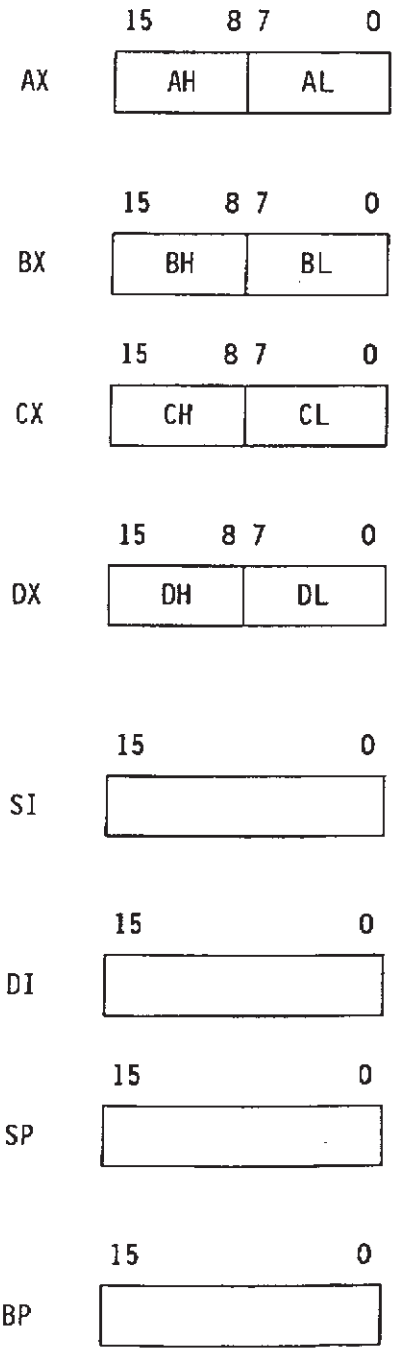
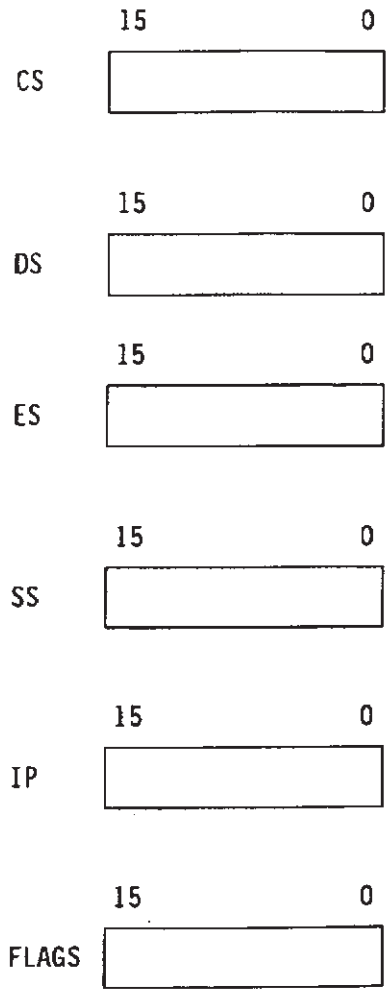
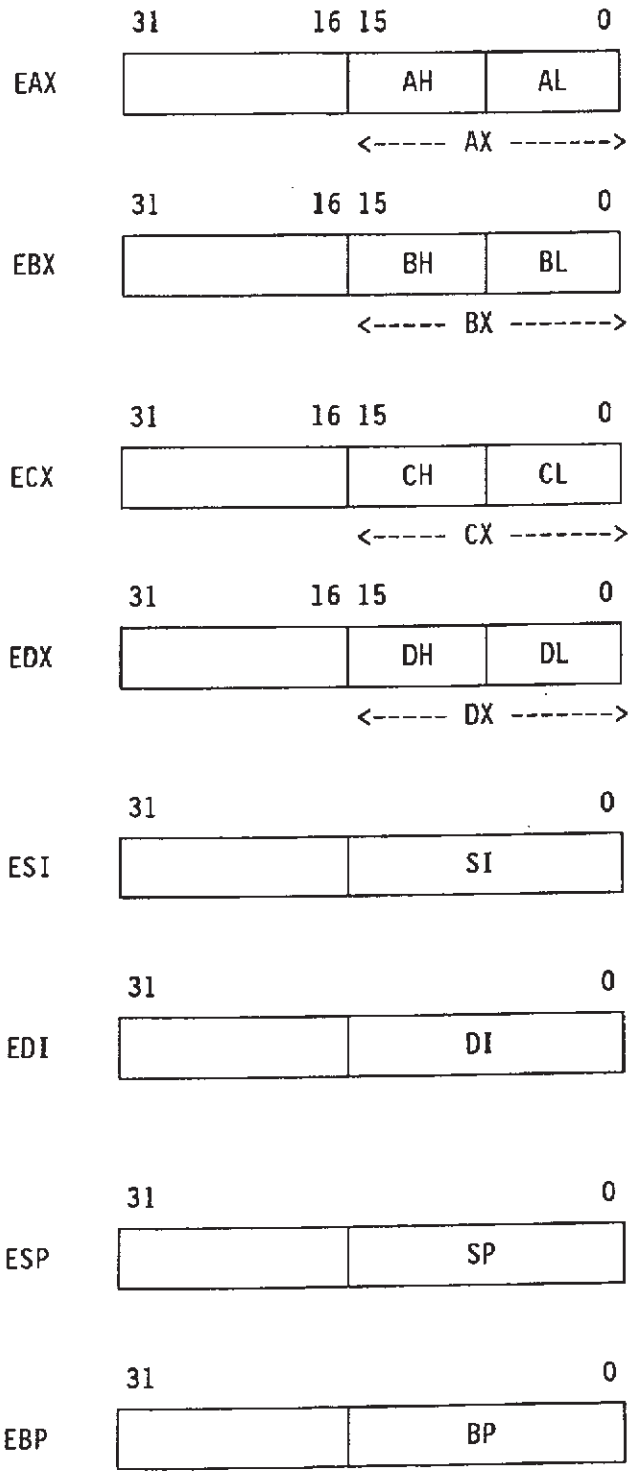


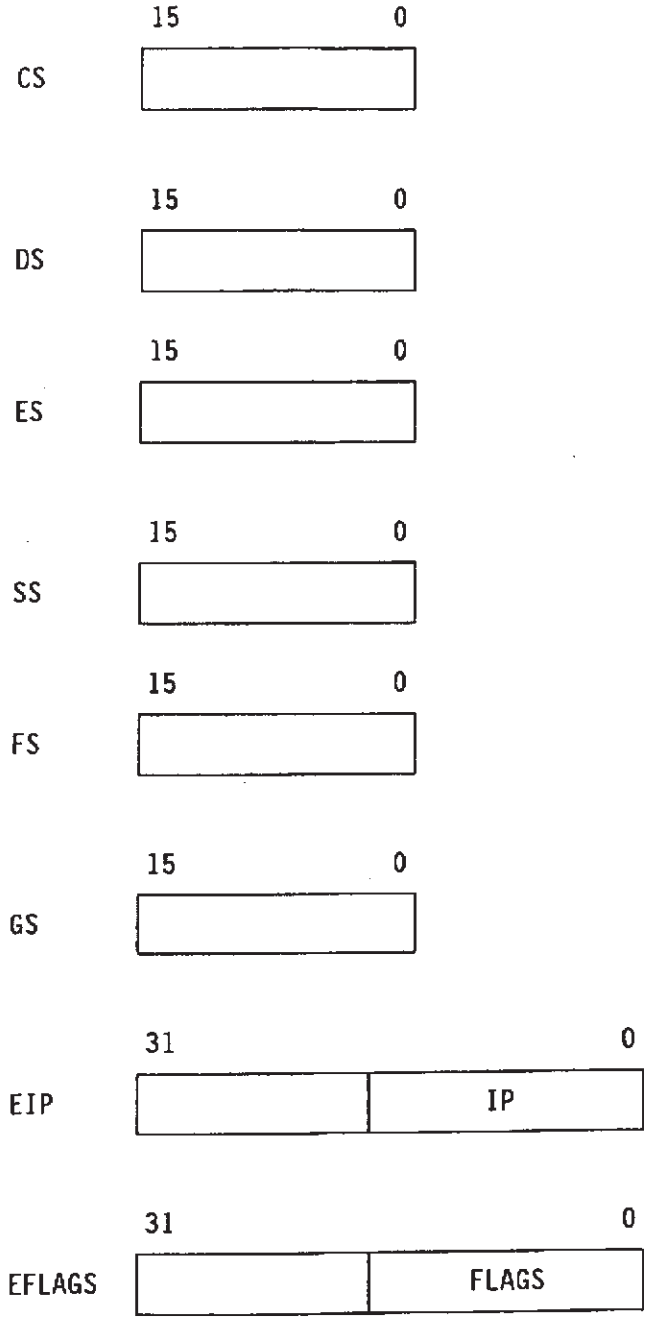
Figure 2.6: The EFlags of the i386. The i386 microprocessor comprises several flags which indicate the result of the previous operation or the current processor status. With the new operation modes of the i386, the number of flags has also increased compared to the 8086.


15 8 7 0
AX
AH
AL
15 8 7 0
BX
BH
BL
15 8 7 0
CX
CH
CL
15 8 7 0
DX
DH
DL
15 0
SI
15 0
DI
15 0
SP
15 0
BP


15
0
CS
15
0
DS
15
0
ES
15
0
SS
15
0
IP
15
0
FLAGS

אוגרי ה-386 ואילך לא כולל אוגרי מערכת (רמת אפליקציה בלבד)




15
0
CS
15
0
DS
15
0
ES
15
0
SS
15
0
FS
15
0
GS
31
0
EIP
IP
31
0
EFLAGS
FLAGS

תקציר מספר 2

אוגרי ה-CPU.

כפי שצוין קודם אוגר הוא תא זכרון הנמצא בתוך ה-CPU - תא זכרון שאליו הגישה היא המהירה ביותר. נוסף לכך, כל פעולה שאינה פעולת זכרון (שמירת ערך ושליפה של ערך) כגון חיבור, חיסור, כפל וכו' חייב להעשות באוגרים.

להזכירכם אנחנו מתיחסים רק לאותם אוגרים שנגישים לתוכנה.

אוגרים משמשים כ-

אוגרים כלליים:

אוגרים כלליים משמשים כ-

- צוברים - אכסון מידע, גם כמידע "טהור" (נתונים) וגם לצרכים נומריים.
- התיחסויות לכתובות בזכרון:
  - כפוינטרים,
  - כאינדקסים של מערכים.

אוגרים מיוחדים:

- מצביעים לפקודה הבאה לביצוע (כפי שהתיחסתי בתקציר הקודם).
- התחלה של שטחי זכרון (עוד נראה זאת).
- שמירת מצב (סטטוס) של ה-CPU.

- במעבדים מתקדמים (286 ואילך באינטל x86) גם אוגרים מיוחדים למנגנוני הגנה, ניהול זכרון, Multitasking ומימוש Debugger-ים במערכות מוגנות. אנחנו לא נתיחס לנושאים הללו בקורס הזה.

אוגרי ה-CPU במעבדי האינטל x86

כאן, כמו ברוב הקורס נעבוד לפי הכלל הבא: נתאר את המצב כפי שהיה ב-8086 (שם המצב היה די פשוט) ונראה כיצד הדברים השתנו כמעבר ל-386 ואילך.

ה-8086 היה מעבד 16 ביט (2 בתים). זה היה גודל ה-bus-ים וגודל האוגרים.

לאוגרים של ה-8086 יש תכונה שדי מבדילה אותה ממעבדים אחרים:
לכל אוגר ב-CPU יש תכונות אופיניות לו: אין שם שני אוגרים שהם זהים ברמת התפקוד. השימוש ברוב האוגרים (אם לא כולם) די מוכתב מראש מהתכונות שלו.

ב-8086 היו 14 אוגרי בגודל 16 ביט:

AX, BX, CX, DX, SI, DI, BP, SP, CS, DS, SS, ES, IP ו-FLAGS.

אוגרים כלליים General Purpose or Data Registers

האוגרים AX, BX, CX, DX, SI, DI, BP, SP הם האוגרים הכלליים של המעבד.

לאוגרים AX, DX הם מעין אוגרים מרכזיים לפעולות נומריות. כפל וחילוק שלמים מתבצעים רק בהם (בצורה מסוית מאד שעוד נראה). ב-8086, חלק מהפעולות האריתמטיות (כמו ADD, SUB) מהירות יותר ב-AX מאשר באוגרים האחרים. מסיבה זאת הוא נקרא לפעמים ה-Accumulator (הצובר). ה-DX נקרא לפעמים ה-Data Register.

לאוגרים SP, BP, SI, ו-DI יש תפקידים מיוחדים בכל הקשור <u>להיתיחסות לזכרון</u>.

השימוש ב-SP ו-BP הינו כמעט תמיד מוגבל למימוש המחסנית. נתיחס לזה בהמשך.

רק לאוגרים BX, SI, DI, BP היכולת לשמש כפוינטרים יחסיים ברוב הפקודות ההתיחסות לזכרון. יש הבדלים בין הזוג BX, BP (הנקראים Base Registers) לבין הזוג SI, DI (הנקראים Index Registers). נראה זאת בקרוב.

לאוגר CX יש תפקיד מיוחד של <u>מניה</u> בחלק מפקודות המכונה. מסיבה זו הוא נקרא ה-Count Register.

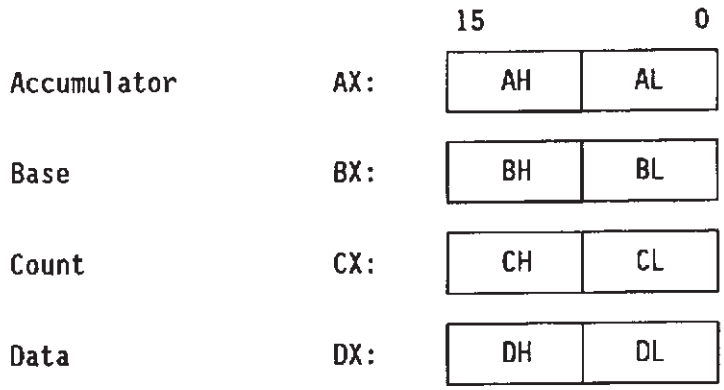
על מנת לאפשר פעולות על בית יחיד, האוגרים AX, BX, CX, DX מתחלקים ל-2. למשל AX = (AH, AL), כאשר AH זה החלק המשמעותי יותר ו-AL החלק המשמעותי פחות. השימוש בחצאי אוגרים הללו הם כאילו AH ו-AL הם אוגרים בפני עצמם

למשל:

```
MOV  AH,23
MOV  AL,55
ADD  AH,AL
```

יחד עם זאת אין לשכוח ש-AL אינו אוגר נוסף ל-AX הכנסת שינוי ב-AL או AH משנה גם את AX.

לפיכך



סה"כ כל אחד מהאוגרים הללו 16 ביטים - ניתן לראות כל אחד כזוג אוגרים 8 ביט.

<u>אוגרים מיוחדים</u>:

האוגרים המיוחדים במעבד 8086 הם אוגרי הסגמנטים, האוגר IP ואוגר הדגלים FLAGS.

אוגרי הסגמנטים משמשים להגדרת שטחי זכרון.
האוגר IP משמש לקביעת הפקודה הבאה לביצוע.
אוגר הדגלים משמש להגדרת סטטוס ה-CPU.

<u>אוגרי הסגמנטים Segment Registers</u>

אוגרי הסגמנטים הם אוגרים שמשמשים רק למטרה אחת: לקבוע נקודת התחלה פיזית לשטחי זכרון. איך מפורש תוכן אוגר הסגמנט לנקודת זכרון נראה בהמשך.

ב-8086 <u>כל</u> התיחסות לזכרון הינה <u>יחסית</u> לנקודת זכרון המוצבעת ע"י אחד <u>מאוגרי סגמנט</u>. הסיבה למה הדבר נעשה בצורה הזו יוסבר מעט בהמשך. תוכן אוגר הסגמנט עובר <u>המרה</u> לכתובת <u>פיזית</u> - <u>אבסולוטית</u> בזכרון. לערך הזה מתוסף ערך

נוסף, שהוא כתובת יחסית לכתובת המוגדרת לאוגר הסגמנט.

אוגרי הסגמנט של ה-8086 הם ה-CS, DS, SS, ES.

התפקיד של כל שטח זכרון המוצבע ע"י כל אוגר סגמנט יש תפקיד שונה:

CS - Code Segment - סגמנט פקודות המכונה של התוכנית.
DS - Data Segment - סגמנט מידע.
SS - Stack Segment - סגמנט מחסנית.
ES - Extra Segment - בדרך כלל לסגמנט מידע נוסף.

רוב פקודות המכונה המיחסים לאוגרים כללים (כמו ADD, SUB) לא פועלים על אוגרי הסגמנטים. לאוגרי הסגמנטים פועלים רק פקודות של קריאה מ-, כתיבה אל האוגר מאחד האוגרים הכלליים 16, או מהזכרון. לדוגמא,

```
ADD DS,3  <====  אין דבר כזה
MOV DS,AX <====  זה יש
MOV AX,DS <====  זה יש
MOV DS,Var1 <====  זה יש
```

האוגר CS מוגבל יותר מהאוגרים DS, SS, ES מסיבות שנראה מיד.

הפקודה הבאה לביצוע במעבד 8086 נקבע ע"י הזוג אוגר סגמנט CS, ואוגר IP. נהוג לציין זאת CS:IP. הזוג הזה CS:IP הוא זה שמתעדכן בזמן ביצוע כל פקודה.

למעשה IP הוא שם לא רשמי של האוגר, שכן הוא לא מופיע בשום פקודת אסמבלי: הפקודות הבאות פשוט לא קיימות:

```
MOV AX,IP  <====  אין דבר כזה
MOV IP,AX  <====  אין דבר כזה
ADD IP,3   <====  ובודאי שלא
```

ערך MOV IP, משמעותו שינוי הפקודה הבאה לביצוע ולמען האמת, אין זה הגיוני לאפשר זאת ע"י פקודה נוסף MOV או ADD.

את IP אפשר לשנות רק ע"י פקודות בקרה (JMP, CALL, INT,..).

מסיבות דומות אי אפשר לעשות

```
MOV CS,AX  <====  אין דבר כזה
        לעומת זאת
MOV AX,CS  <====  זה יש
```

למה משתמשים בסגמנטציה?

הסכמה של אוגרי הסגמנטים + היסט הופך כל התיחסות לזכרון להתיחסות יחסית לנקודות התחלה אבסולוטיות (התחלה של שטחי זכרון). זהו למעשה פתרון של בעיה שיש לנו בבואינו לממש תוכנית: יש לנו צורך להתיחס באופן עקבי לנקודות מסוימות בזכרון (משתנים, פרוצדורות וערד גורמי תכנות שעוד נראה) אבל מצד שני, אנחנו לא יכולים לדעת בזמן הקומפלציה או אסמבלי לאיזה כתובות תשען התוכנית. מערכת ההפעלה היא שטוענת את התוכנית לזכרון. הכתובת בזכרון אליו תועתק התוכנית ע"י מערכת ההפעלה תקבע לפי שטחי הזכרון הפנויים באותו רגע. הדבר תלוי במימוש של מערכת ההפעלה (למשל, כמה זכרון היא תופסת ושיטות העבודה שלה), והסטטוס הספציפי של מערכת ההפעלה במחשב שבו מורצת התוכנית (למשל אם מותקנים תוכנות לניהול העכבר, CD-ROM, מדפסת, צורב וכו'). הדבר עשוי להשתנות אפילו בהרצות שונות של אותה תוכנית על אותו מחשב תחת אותו מערכת הפעלה אם חל שינוי בסטטוס של המערכת, כמו למשל טעינת תוכנית אחרת בזכרון והשארתה שם (דבר שקורה אפילו ב-DOS, התוכנית doskey למשל). למעשה, הבעיה הזו היא אחת ההשלחות מהעובדה שהמחשבים היום תמיד עובדים תחת מערכת הפעלה.

הפתרון של הבעיה הזו הוא די פשוט: אנחנו אולי לא יכולים לקבוע בזמן קומפלציה היכן בזכרון ימומש (למשל) משתנה בית אחד בשם X, אבל אנחנו כן יכולים לקבוע ש-X ימומש (נניח) במרחק 400 בתים מנקודת ההתחלה של שטח המידע של התוכנית. נקודת ההתחלה של השטח הזה יקבע (נניח) לפי הערך של האוגר DS, ערך שיקבע ע"י מערכת ההפעלה. לפיכך התיחסות עקבית למשתנה X יהיה נקודת ההתיחסות של האוגר DS + 400.

אוגר הדגלים FLAGS

אוגר הדגלים שומר את הסטטוס של ה-CPU. ל-8086 ישנו מספר לא גדול של מצבים שהוא יכול להיות בהם בכל רגע (ב-386 המצב מורכב הרבה יותר), הנקבעים ע"י החומרה או התוכנה של המחשב, משיקולים שונים. כמו כן ישנה אינפורמציה הקשורה לפעולות אריתמטיות שיש צורך לשמור בצורה מסוימת. כל זה נעשה ע"י

אוגר 16 ביט בעל שם לא רשמי בשם FLAGS שכמו IP לא מופיע בשום פקודת אסמבלר. 9 מהביטים של האוגר הזה (האחרים אינם כשימוש ב-8086) קובעים כל אחד פרמטר סטטוס אחד לפי הערך שלהם הוא 0 או 1. הביטים הללו נקראים דגלים.

אוגר הדגלים ב-8086 Flags Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

להלן רשימת הדגלים וקצת תיאור שלהם. הסבר אמיתי של תפקידם יעשה מאוחר יותר בקורס.

שם הדגל	משמעות לערך 1	סוג הדגל
CF	carry	אריתמטי
PF	זוגיות	אריתמטי
AF	carry עשרוני	אריתמטי
ZF	תוצאה 0	אריתמטי
SF	תוצאה שלילית	אריתמטי
TF	אפשור פסיקה לצורך מימוש Debugger -ים	תפקוד CPU
IF	אפשור פסיקות	תפקוד CPU
DF	כיוון פעולות מחרוזת	תפקוד CPU
OF	גלישה אריתמטית	אריתמטי

האוגרים מאז ה-386

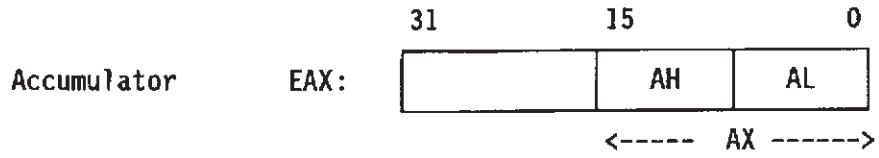
במעבר מ-8086 ל-286 נוספו לרשימת האוגרים של ה-8086 אוגרים מיוחדים להגנה וניהול זכרון, ודגלים חדשים. האוגרים והדגלים של ה-8086 נשארו ב-286, משיקולים של תאימות אחורה. לדעתי אין טעם להתיחס לכך היום, שכן הארכיטקטורה של ה-286 אינו רלוונטי יותר.

במעבר מה-286 ל-386 גובשה רשימת אוגרים (ארכיטקטורה) 32 ביט שלא השתנתה במעברים ל-486, פנטיום, ופנטיום II.

ב-386 אוגרי ה-8086 שאינם אוגרי סגמנט (כלומר AX, BX, CX, DX, SI, DI,

BP, SP, IP ואוגר הדגלים (FLAGS) הורחבו ל-32 ביט ונקראים
EAX, BEX, ECX, EDX, ESI, EDI, EBP, ESP, EIP ו-EFLAGS.
החפקידים של האוגרים הללו נשארו במידה רבה כמו קודם, למרות שיש הבדלים.

שמות אוגרי ה-8086 (AX, AH, AL, BX, BH, BL, ...) ממשיכים להתקיים ומתיחסים לחלקים הנמוכים של האוגרים הללו. לדוגמא, עבור EAX,



אוגרי הסגמנט CS, DS, SS, ES נשארו ללא שינוי. נוספו להם שני אוגרי סגמנט חדשים: FS ו-GS. אגב, האוגרים FS ו-GS הינם האוגרים היחידים באינטל x86 שאפשר לומר עליהם שהם זהים.

נוסף לכך נוספו אוגרים למנגנוני הגנה, ניהול זכרון, Multitasking ומימוש Debugger-ים במערכות מוגנות שרובם (לא כולם) היו ב-286. אנחנו נתיחס לנושאים המתקדמים הללו רק לקראת סוף הסמסטר.