

אסמבלר 8086

רשימת רגיסטרים

רגיסטר הצובר, בגודל מילה מחולק לשני רגיסטרים נוספים AH ו AL	AX - (Accumulator Reg)
רגיסטר בגודל מילה, משמש <u>כמצביע בסיס</u>	BP - (Base Pointer Reg)
רגיסטר הבסיס, בגודל מילה מחולק לשני רגיסטרים נוספים BH ו BL	BX - (Base Reg)
רגיסטר בגודל מילה, המצביע על תחילת סגמנט התכנית	CS - (Code Segment Reg)
רגיסטר המונה, בגודל מילה מחולק לשני רגיסטרים נוספים CH ו CL	CX - (Counter Reg)
רגיסטר בגודל מילה, משמש <u>כמצביע יעד</u>	DI - (Destination Reg)
רגיסטר בגודל מילה, המצביע על תחילת סגמנט הנתונים	DS - (Data Segment Reg)
רגיסטר הנתונים, בגודל מילה מחולק לשני רגיסטרים נוספים DH ו DL	DX - (Data Reg)
רגיסטר בגודל בית, ובו מחזיק ה-CPU את צורת המיעון הנוכחית	EA - (Effective Address Reg)
רגיסטר רזרבי בגודל מילה, מצביע על סגמנט הנתונים בנוסף לרגיסטר ה-DS	ES - (Extra Segment Reg)
רגיסטר בגודל מילה, מצביע על הפקודה הבאה לביצוע ומשמש כהיסט ל CS	IP - (Instruction Pointer Reg)
רגיסטר בגודל בית, ובו נשמרת הפקודה לביצוע לאחר קריאתה מהזיכרון	IR - (Instruction Reg)
רגיסטר בגודל מילה, משמש <u>כמצביע מקור</u>	SI - (Source Reg)
רגיסטר בגודל מילה, משמש <u>כמצביע המחסנית</u>	SP - (Stack Pointer Reg)
רגיסטר בגודל מילה, המצביע על תחילת סגמנט המחסנית	SS - (Stack Segment Reg)

רגיסטר הדגלים

דגל עזר הנשא (4), בודק האם קיים נשא או לווה בין הניבל הנמוך של AL לניבל הגבוה של AL (אם כן, AF=1)	AF -
דגל הנשא (0), בודק האם קיים נשא או לווה מסיבית ה-MSB של התוצאה (אם כן, CF=1)	CF -
דגל הכיוון (10), קובע כיוון התקדמות של הרגיסטרים המצביעים (SI, DI) בריצה על מערכים ומחרוזות (קידום, DF=1)	DF -
דגל אפשר פסיקות (9), קובע אם הפסיקות הניתנות למיסוך, יאופשרו, או לא (אם כן, IF=1)	IF -
דגל הגלישה (11), בודק אם תוצאה של פעולה מתמטית חורגת מטווח הקיבולת של משתנה היעד (במספרים מסומנים)	OF -
דגל הזוגיות (2), בודק את כמות הסיביות המכילות 1 בבית הנמוך של התוצאה (אם זוגי, PF=1)	PF -
דגל הסימן (7), בודק האם סיבית ה-MSB מכילה 1 או לא. (אם כן, SF=0)	SF -
דגל המלכודת (8), משמש לצורכי ניפוי שגיאות ריצה	TF -
דגל האפס (6), בודק לאחר פעולה מתמטית או לוגית, האם התוצאה שווה ל-0 (אם כן ZF=1)	ZF -

הוראות ופקודות

*אופ' = אופרנד

ADC	< Destination >, < Source >	הוסף את תוכן אופ' המקור בתוספת דגל הנשא לתוך אופ' היעד	- ADC
ADD	< Destination >, < Source >	הוסף את תוכן אופ' המקור אל אופ' היעד	- ADD
AND	< Destination >, < Source >	בצע AND סיבית סיבית בין אופ' המקור לאופ' היעד	- AND
CBW		הרחב את AL על פני כל AX (הוסף אפס או אחד ל AH בהתאם לסיבית ה MSB של AL)	- CBW
CLC	CF = 0	אפס את דגל הנשא	- CLC
CLD	DF = 0	אפס את דגל הכיוון (גורם לקידום של SI ו- DI)	- CLD
CMC	CF = ! CF	הפוך את מצב דגל הנשא	- CMC
CMP	< Destination >, < Source >	השווה בין אופ' המקור לבין אופ' היעד	- CMP
CTI	IF = 0	אפס את דגל הפסיקות	- CTI
CWD		הרחב את AX על פני כל DX : AX (הוסף אפס או אחד ל- DX בהתאם לסיבית ה MSB של AX)	- CWD
DEC	< Source >	הפחת 1 מאופ' היעד	- DEC
DIV		חלוקה של אופ' המקור (לא מסומן) בצובר. התוצאה תשמר ב- AL והשארית ב- AH (חלוקה בין בתיים), התוצאה תשמר ב- AX והשארית ב- DX (חלוקה בין מילים).	- DIV
IDIV	< Source >	חלוקה של אופ' המקור (מסומן) בצובר. התוצאה תשמר ב- AL והשארית ב- AH (חלוקה בין בתיים), התוצאה תשמר ב- AX והשארית ב- DX (חלוקה בין מילים).	- IDIV
IMUL		כפל של אופ' המקור (מסומן) בצובר. התוצאה תשמר ב- AX (כפל בין בתיים), התוצאה תשמר ב- AX : DX (כפל בין מילים).	- IMUL
IN	< Source >	קלוט אופ' מתוך Port	- IN
INC	< Source >	הוסף 1 לאופ' היעד	- INC
JMP		קפוץ לתווית המטרה	- JMP
LAHF		טען ל AH את חלקו הנמוך של רגיסטר הדגלים	- LAHF
LEA	< Destination >, < Source >	טען את הכתובת האפקטיבית של אופ' המקור אל אופ' היעד	- LEA
LOOP	< Label >	בצע לולאה, כל עוד CX שונה מ- 0, הפחת 1 מ- CX וקפוץ לתווית המטרה	- LOOP
MOV	< Destination >, < Source >	העבר תוכן אופ' המקור אל אופ' היעד	- MOV
MUL	< Source >	כפל של אופ' המקור (לא מסומן) בצובר. התוצאה תשמר ב- AX (כפל בין בתיים), התוצאה תשמר ב- AX : DX (כפל בין מילים).	- MUL
NEG	< Source >	בצע השלמה ל- 2 על אופ' היעד	- NEG
NOT	< Source >	הפוך את כל הסיביות של אופ' היעד	- NOT

OR	< Destination >, < Source >	בצע OR סיבית סיבית בין אופ' המקור לאופ' היעד	- OR
		פלוט אופ' אל תוך Port	- OUT
		טען את AH לחלקו הנמוך של רגיסטר הדגלים	- SAHF
SBB	< Destination >, < Source >	הפחת את תוכן אופ' המקור ואת דגל הנשא מאופ' היעד	- SBB
	CF = 1	קבע את דגל הנשא ל- 1	- STC
	DF = 1	קבע את דגל הכיוון ל- 1 (גורם להפחתה של SI ו DI)	- STD
	IF = 1	קבע את דגל הפסיקות ל- 1	- STI
	TF = 1	קבע את דגל המלכודת ל- 1	- STT
SUB	< Destination >, < Source >	הפחת את תוכן אופ' המקור מאופ' היעד	- SUB
XOR	< Destination >, < Source >	בצע XOR סיבית סיבית בין אופ' המקור לאופ' היעד	- XOR
XCHG	< Destination >, < Source >	החלף את התוכן של אופ' המקור בתוכן אופ' היעד, ולהפך.	- XCHG

הוראות קפיצה מותנית

(מספרים לא מסומנים)	קפוץ אם מעל	(Jump if Above)	- JA .1
(מספרים לא מסומנים)	קפוץ אם מעל או שווה	(Jump if Above or Equal)	- JAE .2
(מספרים לא מסומנים)	קפוץ אם מתחת	(Jump if Below)	- JB .3
(מספרים לא מסומנים)	קפוץ אם מתחת או שווה	(Jump if Below or Equal)	- JBE .4
CF = 1	קפוץ אם נשא	(Jump if Carry)	- JC .5
CX = 0	קפוץ אם רגיסטר המונה מאופס	(Jump if CX Zero)	- JCXZ .6
ZF = 1	קפוץ אם שווה	(Jump if Equal)	- JE .7
(מספרים מסומנים)	קפוץ אם גדול	(Jump if Greater)	- JG .8
(מספרים מסומנים)	קפוץ אם גדול או שווה	(Jump if Greater or Equal)	- JGE .9
(מספרים מסומנים)	קפוץ אם קטן	(Jump if Less)	- JL .10
(מספרים מסומנים)	קפוץ אם קטן או שווה	(Jump if Less or Equal)	- JLE .11
(מספרים לא מסומנים)	קפוץ אם לא מעל	(Jump if Not Above)	- JNA .12
(מספרים לא מסומנים)	קפוץ אם לא מעל או לא שווה	(Jump if Not Above Nor Equal)	- JNAE .13
(מספרים לא מסומנים)	קפוץ אם לא מתחת	(Jump if Not Below)	- JNB .14
(מספרים לא מסומנים)	קפוץ אם לא מתחת או לא שווה	(Jump if Not Below Nor Equal)	- JNBE .15
CF = 1	קפוץ אם אין נשא	(Jump if Not Carry)	- JNC .16
ZF = 0	קפוץ אם לא שווה	(Jump if Not Equal)	- JNE .17
(מספרים מסומנים)	קפוץ אם לא גדול	(Jump if Not Greater)	- JNG .18
(מספרים מסומנים)	קפוץ אם לא גדול או שווה	(Jump if Not Greater Nor Equal)	- JNGE .19
(מספרים מסומנים)	קפוץ אם לא קטן	(Jump if Not Less)	- JNL .20
(מספרים מסומנים)	קפוץ אם לא קטן או שווה	(Jump if Not Less Nor Equal)	- JNLE .21
(מספרים מסומנים) OF = 0	קפוץ אם אין גלישה	(Jump if Not Overflow)	- JNO .22
PF = 0	קפוץ אם לא זוגי	(Jump if Not Parity)	- JNP .23
(מספרים מסומנים) SF = 0	קפוץ אם סימן חיובי	(Jump if Not Sign)	- JNS .24
ZF = 0	קפוץ אם שונה מאפס	(Jump if Not Zero)	- JNZ .25
(מספרים מסומנים) OF = 1	קפוץ אם גלישה	(Jump if Overflow)	- JO .26
PF = 1	קפוץ אם זוגי	(Jump if Parity)	- JP .27
PF = 1	קפוץ אם זוגי	(Jump if Parity Even)	- JPE .28
PF = 0	קפוץ אם פרדי	(Jump if Parity Odd)	- JPO .29
(מספרים מסומנים) SF = 1	קפוץ אם סימן שלילי	(Jump if Sign)	- JS .30
ZF = 1	קפוץ אם אפס	(Jump if Zero)	- JZ .31

סוגי משתנים

8 Bits	משתנה בגודל בית אחד	(Define Byte -)	DB
16 Bits	משתנה בגודל שני בתים	(Define Word -)	DW
32 Bits	משתנה בגודל ארבעה בתים	(Define Double Word -)	DD
64 Bits	משתנה בגודל שמונה בתים	(Define Quattro Word -)	DQ
80 Bits	משתנה בגודל עשרה בתים	(Define Ten Bytes -)	DT

מושגים נוספים

MOV < Destination >, OFFSET < Source >	אופרטור המספק את ההיסט של אופרנד מסוים	OFFSET
MOV < Destination >, SEG < Source >	אופרטור המספק את הסגמנט של אופרנד מסוים	SEG
פקודה המשמשת להגדרת קבועים, ממוקמת לפני סגמנט הנתונים, לא ניתן לשנות ערך הקבועים במהלך התכנית.		EQU
פקודה המשמשת להגדרת קבועים, ממוקמת לפני סגמנט הנתונים, ניתן לשנות ערך הקבועים במהלך התכנית.		" = "
א. הגדרה של סוג המשתנה		PTR
ב. שינוי סוג משתנה קיים ע"י הרחבתו למשתנה בגודל המבוקש		

BYTE PTR < Operand >
WORD PTR < Operand >
DWORD PTR < Operand >

*Operand = **SI / DI / BP/ BX / VAR**

INT פסיקת תוכנה המתקשרת עם מערכת ההפעלה **DOS**, באמצעות קוד בקשה המועבר לרגיסטר **AH**

קוד בקשה:

MOV AH,1	קריאת תו בודד מהמקלדת, קוד ה- ASCII מוחזר ב- AL	•
MOV AH,2	הדפסת תו בודד, קוד ה- ASCII שלו נמצא ב- DL	•
MOV AH,8	קריאת תו בודד מהמקלדת, ללא הצגתו על המסך, קוד ה- ASCII מוחזר ב- AL	•
MOV AH,9	הדפסת מחרוזת תווים שכתובתה נמצאת ב- DX , (יש לציין סוף מחרוזת ב- '\$')	•
MOV AH,10	קריאת מחרוזת תווים, שכתובתה נמצאת ב- DX	•
MOV AH,4CH	חזרה למערכת ההפעלה	•

לאחר הפקודות הנ"ל תבוא הפסיקה:

INT 21H