שפת אסמבלי למחשב האישי

מהדורה שנייה

הוצאת הוד-עמי

לספרי מחשבים 09-9564716

www.hod-ami.co.il

הוראות התקנת התקליטור נמצאות בנספח

מוקדש

לאשתי חנה ובתי מירי

היקרות והאהובות

ברצוני להודות למורי ועמיתי, אשר הואילו לעבור על כתב היד, העירו והאירו:
רבקה הלפמן, הרצל נח, שמואל כהן.

עורך ראשי: **יצחק עמיהוד**

עריכה לשונית ועיצוב: **קרן לנדאו, שרה עמיהוד**

עיצוב עטיפה: **שרון רז**

פרק 19 נכתב על ידי **אסף שלי**

שמות מסחריים

שמות המוצרים והשירותים המוזכרים בספר הינם שמות מסחריים רשומים של החברות שלהם. **הוצאת הוד-עמי** עשתה כמיטב יכולתה למסור מידע אודות השמות המסחריים המוזכרים בספר זה ולציין את שמות החברות, המוצרים והשירותים. שמות מסחריים רשומים (registered trademarks) המוזכרים בספר צוינו בהתאמה.

הודעה

ספר זה מיועד לתת מידע אודות מוצרים שונים. נעשו מאמצים רבים לגרום לכך שהספר יהיה שלם ואמין ככל שניתן, אך אין משתמעת מכך אחריות כלשהי.

המידע ניתן "כמות שהוא" ("as is"). הוצאת הוד-עמי אינה אחראית כלפי יחיד או ארגון עבור כל אובדן או נזק אשר ייגרם, אם ייגרם, מהמידע שבספר זה, או מהתקליטור המצורף.

www.hod-ami.co.il

לשם שטף הקריאה כתוב ספר זה בלשון זכר בלבד. ספר זה מיועד לגברים ונשים כאחד ואין בכוונתנו להפלות או לפגוע בציבור המשתמשים/ות.

- **טלפון: 09-9564716**
- **פקס: 09-9571582**
- **דואר אלקטרוני: Info@hod-ami.co.il**
- **אתר באינטרנט: http://www.hod-ami.co.il**

שפת אסמבלי למחשב האישי

מהדורה שנייה

אלי כהן

www.hod-ami.co.il

הוצאת הוד-עמי
לספרי מחשבים

The PC Assembly Language

2 Edition

By **Eli Cohen**

Editor: **I. Amihud**

(C)

כל הזכויות שמורות

הוצאת הוד-עמי
לספרי מחשבים בע"מ
ת.ד. 6108 הרצליה 46160
טלפון: 09-9564716 פקס: 09-9571582
info@hod-ami.co.il

אין להעתיק או לשדר בכל אמצעי שהוא ספר זה או קטעים ממנו בשום צורה ובשום אמצעי אלקטרוני או מכני, לרבות צילום והקלטה, אמצעי אחסון והפצת מידע, ללא אישור בכתב מאת ההוצאה, אלא לשם ציטוט קטעים קצרים בציון שם המקור.

www.hod-ami.co.il

הודפס בישראל 1999

קדם דפוס: **טלאור טכנולוגיות**

All Rights Reserved
HOD-AMI Ltd.
P.O.B. 6108, Herzliya
ISRAEL, 1999

ISBN 965-361-207-7 מסת"ב

הקדמה

שפת התכנות אסמבלי - Assembly Language - למחשבים אישיים נלמדת במסגרות לימודים רבות, וביניהן: בתי ספר מקצועיים ומכללות שבהם לומדים את מקצועות האלקטרוניקה, מכשור ובקרה (כיתות י"א עד י"ג); מגמות טכ"מ; הפקולטות למתמטיקה ומחשבים באוניברסיטאות; הפקולטות למהנדסי אלקטרוניקה ובקרה; מכללות ללימודי מחשבים ועוד. רבים משתמשים בשפת אסמבלי לכתיבת שגרות ותוכניות כחלק מתוכנות עיבוד גרפיות, מולטימדיה, תקשורת ועוד.

הספר צמח מתוך ניסיון עבודה בהוראת המקצוע לתלמידים ולסטודנטים, ומתוך ניסיון מעשי. הוא מכיל הסברים מפורטים של פקודות שפת אסמבלי, דוגמאות תרגילים עם פתרונות מפורטים, ולעיתים יותר מפתרון אחד לתרגיל, שיטות לניפוי ותיקון שגיאות ועוד. המחבר שם דגש על פישוט הצגת הנושאים ונותן דוגמאות ותרשימים להמחשת ההסבר. התוכניות כתובות לפי כללי תכנות מתקדמים ובכל מקרה שהיה צורך, הודגמו טעויות אופייניות ודרכים למניעתן.

לכל הלומדים והמורים מוקדש ספר זה.

מהדורה זו הנוכחית מתבססת על רב המכר "שפת אסמבלי למחשב האישי", שיצא לאור בהוצאת הוד-עמי.

הנחיות לקורא

www.hod-ami.co.il

- הספר נכתב לפי שיטת "קליפות", על פיה החשיפה לחומר הנלמד הינה הדרגתית. בתחילה ניתן הסבר קצר "כיצד לבצע" פעולה כלשהי, ורק מאוחר יותר ניתן פירוט "מדוע לבצע כך".
- במקרים אחרים מובא הסבר חלקי של הנושא, ורק לאחר שתלמד חומר נוסף, אנו חוזרים ומלמדים את הפרטים בהרחבה. שיטה זו נועדה להקל על הבנת החומר.
- קרא בעיון את כל דוגמאות התרגילים ופתרונותיהם. הדבר יסייע לך להבין את החומר התיאורטי ויעזור לך בעת פתרון תרגילים מתקדמים. שים לב שלבעיה כלשהי עשויים להיות מספר פתרונות. בדרך כלל, הוצג פתרון אחד ובמקרים אחרים שניים, או שלושה פתרונות והודגשו הדרכים השונות לפתרון.
- בסוף כל פרק ניתנו תרגילים רבים לצורך חזרה והתנסות בתכנות. מומלץ מאוד לפתור אותם, ואם מתקשים בתרגיל כלשהו, אפשר לפנות לאחת הדוגמאות או להסברים ולחזור לפתרון התרגיל.

  ייתכן שבמקרים מסוימים יהיו לך שאלות אודות החומר, או תחושה שטרם הבנת את הדברים בצורה מלאה. רשום שאלה זו בצד, ונסה לענות עליה **בעצמך** מאוחר יותר, כאשר יובהרו נושאים נוספים. זכור שחלק מהלימוד של שפה הינה באמצעות "חקירתה" וניסיון "לפענחה".
- הספר נכתב בלשון זכר לשם פישוט הכתיבה, אך יש להבין זאת כפנייה גם אליך התלמידה, או הסטודנטית.

אני מאחל לכם הלומדים והמורים שימוש מועיל בספר.

המחבר

תוכן עניינים מקוצר

הקדמה **15**

פרק 1: מושגי יסוד **17**

פרק 2: פקודות בסיסיות **27**

פרק 3: כתיבת תוכנית שלמה **53**

פרק 4: תרגול מעשי **57**

פרק 5: לולאות **75**

פרק 6: פסיקות **105**

פרק 7: כתובות זיכרון **131**

פרק 8: מושגים **141**

פרק 9: אוגר הדגלים **143**

פרק 10: עבודה בסיביות **159**

פרק 11: משתנים ומערכים **205**

פרק 12: המחסנית **227**

פרק 13: פרוצדורות **241**

פרק 14: קבועים (Constants) **257**

פרק 15: אסמבלי מותנה (Conditional Assembly) **261**

פרק 16: מאקרו (Macro) **267**

פרק 17: פקודות מחרוזת **277**

פרק 18: פקודות נוספות **289**

פרק 19: אסמבלי ושפת C **297**

נספח: התקליטור המצורף **307**

אינדקס **329**

www.hod-ami.co.il

תוכן העניינים

הקדמה 15

פרק 1: מושגי יסוד 17
המיקרומעבד 17
שפות תכנות 19
יחידות אחסנה 21
בסיסי ספירה 21
ייצוג מספרים 21
הזיכרון 22
האוגרים 23
אוגרים כלליים (General Purpose registers) 23
אוגרי המקטע (Segment registers) 24
אוגרים מצביעים (Pointer registers) 24
אוגר הדגלים (Flags register) 25

פרק 2: פקודות בסיסיות 27
כיצד כותבים פקודות בשפת אסמבלי 27
הפקודה MOV (Move) 28
העתקה ישירה לאוגר 28
העתקת נתון מאוגר לאוגר 29
"כלל בחל" 1: התאמה בגודל האופרנדים 30
העתקת נתון מאוגר לתא זיכרון 30
"כלל בחל" 2: פנייה לזיכרון 31
העתקת נתון מהזיכרון לאוגר 32
דוגמאות לתוכניות ופתרונות 32
תרגילים 35
הפקודה INC (Increment) 36
הפקודה DEC (Decrement) 36
הפקודה ADD (Addition) 36
הפקודה SUB (Subtract) 37
הפקודה NOP (No Operation) 37
הפקודה CMP (Compare) 37
הפקודה JMP (Jump) 38
תווית (label) 38
פקודת קפיצה מותנית (Conditional Jump) 39
תוכנית לדוגמה 39
הצבת ערכים הקסדצימליים באוגר 40
תרגילים ופתרונות 40

פרק 3: כתיבת תוכנית שלמה 53
מהי תוכנית שלמה? 53

פרק 4: תרגול מעשי 57
כיצד מריצים תוכנית אסמבלי? 57
תרגיל מעשי: שלבי תכנות, הרצה ובדיקה 58
כתיבת תוכנית דוגמה באמצעות עורך 58

www.hod-ami.co.il

תרגום התוכנית באמצעות אסמבלר MASM 59
קישור באמצעות תוכנת LINK 59
הרצת התוכנית ובדיקתה על ידי תוכנת DEBUG 59
תרגילים מעשיים לדוגמה 61
תרגיל מעשי א' 61
תרגיל מעשי ב' 62
תרגיל מעשי ג' 63
שלבי הרצת הבדיקה באמצעות DEBUG 64
תרגיל מעשי ד' 65
השלבים לכתיבת התוכנית והרצתה 65
שלבי ההרצה והבדיקה 66
טעויות אופייניות בתכנות: הסבר וסיכום 67
שגיאות בהרצת MASM 67
שגיאות בהרצת תוכנת DEBUG 69
תקציר פקודות DEBUG 70
תרגילים 72

פרק 5: לולאות 75
לולאה בתוכנית 75
הפקודה LOOP 78
דוגמאות לביצוע לולאות 78
תרגיל 1: תוכנית שרצה כראוי 78
תרגיל 2: איך בודקים תוכנית שגויה 80
תרגיל 3: הצבת נתונים בבלוק (קטע) זיכרון 82
תרגיל 4: העתקת בלוקים 83
תרגיל 5: ספירת תאי זיכרון המתאימים לקריטריון 84
תרגיל 6: בדיקת ערכי תאים 85
תרגיל 7: בדיקת מיון בלוק נתונים 86
תרגיל 8: סיכום ערכים בזיכרון 88
תרגיל 9: הזזת ערכים בזיכרון 89
תרגיל 10: מציאת הערך הגבוה ביותר 90
תרגיל 11: חיפוש כתובת על פי קריטריון 92
תרגיל 12: בדיקת סכום ערכים בבלוק נתונים 93
תרגיל 13: חיפוש מספר בזיכרון 95
תרגיל 14: חיפוש ערכים על פי קריטריון 96
תרגיל 15: בדיקת תנאים בזיכרון 97
תרגיל 16: בדיקת תנאים בזיכרון 98
תרגיל 17: השוואת בלוקים 99
תרגיל 18: השוואה והצבה 100
תרגיל 19: חיפוש רצף של נתונים בזיכרון 101
תרגיל 20: בדיקת תחום ערכים בזיכרון 102
תרגילים 103

פרק 6: פסיקות 105
מה זו פסיקה וכיצד מפעילים אותה? 105
פסיקה לסיום תוכנית ויציאה למערכת ההפעלה 106
פסיקה להצגת הודעה על המסך 108
פסיקה לקליטת מקש מהמקלדת 110
פסיקה להצגת תו בודד על המסך 113
פסיקה הבודקת אם הוקש מקש 119
קליטת מקשים מיוחדים 120
פסיקה לקביעת מיקום הסמן במסך 122

www.hod-ami.co.il

פסיקה לקליטת מחרוזת מהמקלדת..... 123
פסיקה לאתחול המחשב (Reset)..... 125
ניפוי שגיאות בתוכנית הכוללת פסיקות..... 126
דוגמה 1: הצגת הודעה..... 126
דוגמה 2: קליטת מקש והצגתו..... 128
סיכום עיקרי הפסיקות..... 129
תרגילים..... 129

פרק 7: כתובות זיכרון..... 131
שיטת הכתובות במחשב האישי..... 131
פנייה לכתובת פיסית באמצעות תוכנת ניפוי (כדוגמת DEBUG)..... 134
תרגילים בכתובות פיסיות (באמצעות DEBUG)..... 135
פנייה לכתובות פיסיות באמצעות תוכנית..... 136
הארות והערות..... 138
תרגילים בכתובות פיסיות..... 139
כתיבת נתונים בזיכרון..... 139
תרגילים..... 140

פרק 8: מושגים..... 141
פקודות והנחיות..... 141
סגמנטים - מקטעי זיכרון..... 141
תרגילים..... 142

פרק 9: אוגר הדגלים..... 143
הדגלים ותפקידיהם..... 143
הדגלים לסוגיהם..... 143
דגל האפס..... 144
תרגילים..... 145
דגל הסימן..... 145
שיטת המשלים ל-2..... 146
הסבת מספרים מבסיס הקסדצימלי לבסיס בינארי..... 147
תרגילים..... 149
הסבת מספר בינארי לייצוג הקסדצימלי..... 149
תרגילים..... 150
דגל הזכור (נשא)..... 150
דגל הגלישה..... 152
הצגת הדגלים על ידי DEBUG..... 153
תרגילים ופתרונות לדוגמה..... 154
תרגילים..... 157
סיכום פקודות הקפיצה המותנית הנוספות..... 158

פרק 10: עבודה בסיביות..... 159
הפקודה MOV עם המאפיינים Byte PTR ו- Word PTR..... 159
הפקודה AND..... 160
כפל לוגי של סיביות..... 160
מיסוך באמצעות הפקודה AND..... 162
תרגול הפקודות..... 165
תרגילים..... 165
הפקודה TEST..... 166
תוכניות דוגמה והרצתן ב-DEBUG..... 167
תרגילים..... 170
הפקודה OR..... 170

www.hod-ami.co.il

תרגילים לדוגמה..... 171
תרגילים..... 172
הפקודה NOT..... 172
הפקודה NEG..... 173
הפקודה XOR..... 173
דוגמאות לשימוש בפקודה..... 174
הפקודה SHR..... 175
חילוק על ידי פעולת SHR..... 176
דוגמאות לפקודת ההזזה..... 178
הרצת התוכנית ב-DEBUG..... 178
פתרון בדרך מהירה יותר..... 179
בדיקת התוכנית ב-DEBUG..... 180
הפקודה SHL..... 182
הפקודה ROR..... 183
הפקודה ROL..... 185
תרגילים ופתרונות לדוגמה..... 186
בידוד סיביות..... 186
חיפוש סיבית בתא זיכרון..... 187
החלפה בין ערכים של שתי סיביות..... 189
חיפוש נתון בזיכרון..... 190
הצגת ייצוג בינארי של מקשים..... 191
קליטת ספרות הקסדצימליות וחישוב הסכום..... 194
חישוב ממוצע והצגתו..... 196
מיון של קבוצת מספרים..... 197
קליטת משפט והצגת המילים..... 198
תרגילים בנושא "החברים של..."..... 199
דוגמאות תרגול ל"חברים של..."..... 201
תרגילים..... 202
תרגילים כלליים..... 202
תרגילים בנושא "החברים של AL"..... 203

פרק 11: משתנים ומערכים..... 205
מה הם המשתנים?..... 205
הגדרת המשתנים בתוכנית..... 206
דוגמאות להגדרת משתנים בתוך מקטע התוכנית..... 206
דוגמאות להגדרת משתנים במקטע נתונים..... 208
כללי שימוש במשתנים..... 210
המשמעות של הגדרת משתנים בתוכנית..... 211
תרגילים ופתרונות..... 212
מערכים..... 217
הגדרת המערך ואיבריו..... 217
תוכניות דוגמה למערכים..... 218
תרגילים..... 223

פרק 12: המחסנית..... 227
הגדרת המחסנית..... 228
הפקודה PUSH..... 228
הפקודה POP..... 229
תוכניות לדוגמה..... 229
איך פועלת המחסנית..... 233
הכנסה והוצאה של נתונים מהמחסנית..... 235
בדיקת המחסנית באמצעות תוכנית ניפוי..... 237

www.hod-ami.co.il

תרגילים 238

פרק 13: פרוצדורות 241
הגדרת הפרוצדורה 241
מנגנון ביצוע הפרוצדורה 244
העברת נתונים אל הפרוצדורה ומהפרוצדורה 247
העברת נתונים באמצעות אוגרים 247
העברת נתונים באמצעות משתנים 248
העברת נתונים באמצעות מחסנית 249
פרוצדורה קרובה ופרוצדורה רחוקה 251
מתי נצטרך להגדיר פרוצדורה רחוקה 251
כיצד מגדירים פרוצדורה רחוקה 251
מה ההבדל במנגנון הקריאה והחזרה מהפרוצדורה 252
תיעוד הפרוצדורה 252
תרגילים 255

פרק 14: קבועים (Constants) 257
הקבועים ותפקידם 257

פרק 15: אסמבלי מותנה (Conditional Assembly) 261
מתי משתמשים באסמבלי מותנה 262
הוספת קטעי קוד לניפוי שגיאות 262
התאמות ייחודיות למשתמש 263
הנחיות נוספות של אסמבלי מותנה 263

פרק 16: מאקרו (Macro) 267
מהו מאקרו? 267
מאקרו עם ארגומנט 269
שילוב מאקרו עם אסמבלי מותנה 270
מאקרו לשכפול קוד REPT 271
מאקרו לשכפול קוד על פי רשימה - IRP 273
מאקרו לשכפול קוד על פי רשימה - IRC 274
תרגילים 275

פרק 17: פקודות מחרוזת 277
הפקודה LODSB (LOaD String Byte) 277
הפקודה STOSB (STOre String Byte) 278
הפקודה LODSW (LOaD String Word) 279
הפקודה STOSW (STOre String Word) 279
הפקודה REP (REPeat) 280
הפקודה CMPSB (CoMPare String Byte) 282
הפקודה REPE (REPeat Equal) 282
הפקודה CMPSW (CoMPare String Word) 284
הפקודה SCASB (SCAn String Byte) 284
הפקודה REPNE (REPeatNot Equal) 285
הפקודה SCASW (SCAn String Word) 286
הפקודות std ו- cld 286
דוגמה מסכמת לשימוש בפקודות מחרוזת 287
תרגילים 288

www.hod-ami.co.il

פרק 18: פקודות נוספות 289
הפקודה MUL 289
הפקודה DIV 291
כפל וחילוק מספרים מסומנים - IDIV, IMUL 293
הפקודה CBW 293
הפקודה CWD 293
הפקודה DAA 294
הפקודה DAS 294
פקודות לשמירה ואיחזור ערך אוגר הדגלים 294
פקודות קלט-פלט: IN ו-OUT 295
תרגילים 296

פרק 19: אסמבלי ושפת C 297
שפת אסמבלי לעומת שפות עיליות 297
שילוב קוד אסמבלי בפקודות שפת C 298
שילוב אסמבלי עם משתני שפת C 298
קישור אסמבלי לשפת C 300
קישור 301
העברת פרמטרים משפת אסמבלי לשפת C ולהיפך 301
הדגמת הקישור 303
הערות חשובות 305

נספח: התקליטור המצורף 307
קטלוג CD - הקטלוג הצבעוני האינטראקטיבי של הוד-עמי 308
קטלוג HTML - קטלוג צבעוני 315
התקנת קטלוג HTML 317
Acrobat Reader - התקנה 318
ספרים לדוגמה 318
חושבים חלונות - הפעלה 318
מבחן אישי - התקנה 320
מה עוד בתקליטור? 321
FontsPekan 321
Terra - מימד חדש בתצוגה - ים המלח ממעוף הציפור 322
התקנת תוכנת גלישה לאינטרנט Microsoft Internet Explorer 5 324
NETEX 324
היכן נמצאים הקבצים הקשורים לספר זה? 326
העתקת קבצי המקור לדיסק 326
תיקיה ראשית SoftWare (רשימה חלקית) 327

אינדקס 329

www.hod-ami.co.il

9

אוגר הדגלים

הדגלים ותפקידיהם

אוגר הדגלים, שגודלו 16 סיביות, מכיל 9 סיביות המשמשות כ**דגלים** (flags). כל אחת מהסיביות יכולה להיות במצב 0 לוגי ("0") או במצב 1 לוגי ("1"). הדגלים מושפעים מפקודות אריתמטיות ולוגיות בלבד! חלק מהפקודות אינן משפיעות על הדגלים. יש לזכור: מצב הדגלים יישאר באוגר עד אשר אחת (או יותר) מהפקודות יגרמו לשינוי שלהם!

את הערכים של חלק מהדגלים אפשר **לקבוע** על ידי פקודות מיוחדות.

הדגלים לסוגיהם

1. **דגל אפס** (zero flag): דגל זה יהיה במצב "1" כאשר התוצאה של הפעולה האריתמטית או הלוגית **האחרונה** היא 0.
2. **דגל סימן** (sign): מצבו "1" כאשר תוצאת הפעולה האריתמטית הינה ערך שלילי.
3. **דגל גלישה** (overflow): מצבו "1" כאשר הערך הנובע מהפעולה גדול מדי מכדי להיכנס לתא הזיכרון, או לאוגר שהוקצו לה.
4. **דגל זכור/נשא** (carry): מצבו "1" כאשר יש זכור בפעולה האריתמטית. סיבית שערכה "1" יוצאת מחוץ לאוגר משמאל, או מימין.
5. **דגל זכור-עזר** (auxiliary carry): מצבו "1" כאשר יש העברה של "1" מה-nibble התחתון ל-nibble העליון (nibble = קבוצת 4 סיביות, חצי אוגר).
6. **דגל פסיקה** (interrupt): כאשר דגל זה במצב "0", לא ניתן לבצע פסיקות.
7. **דגל זוגיות** (parity): נמצא במצב "1", אם יש מספר זוגי של סיביות "1" בבית (byte) שבאופרנד היעד.
8. **דגל כיוון** (direction): דגל זה קובע את הכיוון עבור פקודות מחרוזת. אלו פקודות מיוחדות המטפלות במחרוזות בזיכרון, נדון בהן בהמשך.
9. **דגל צעד-יחיד** (trap): כאשר הדגל במצב "1" מתבצעת הרצת צעד-יחיד. מכנים זאת גם "מלכודת", מכיון שלוכדים כל פקודה בנפרד כאשר היא מתבצעת.

מיקום הדגלים קבוע בתוך האוגר. הסיבית הימנית ביותר הינה סיבית 0:

סיבית	הדגל
0	זכור (נשא)
2	זוגיות
4	זכור-העזר
6	אפס
7	סימן

www.hod-ami.co.il

8	צעד-יחיד (מלכודת)
9	פסיקה
10	כיוון
11	גלישה

דגל האפס

דגל זה נקרא באנגלית **Zero Flag**, או בקצרה: Z.F. כאשר מתבצע **חישוב** והתוצאה הינה 0, דגל זה עובר למצב "1". לפניך מספר דוגמאות לפעולות שלאחריהן דגל האפס יקבל את הערך "1":

1.לאחר ביצוע הפקודות:

```
MOV AL,4
SUB AL,4
```

2.לאחר הפקודות:

```
MOV CL,0FFH
ADD CL,1
```

הסבר: בתחילה מוצב הערך FFH באוגר CL. ערך זה הוא למעשה 1-. לאחר הוספת 1, התוצאה היא 0, ולכן דגל האפס הוא "1".

3.לאחר הפקודות:

```
MOV CX,1
DEC CX
```

דגל האפס **לא** יהיה "1" כאשר לא מתבצעת פעולה חשבונית. הפעולה הבאה, למשל, **לא** תגרום לכך שדגל האפס יהיה "1", מכיון שזו אינה פעולה אריתמטית!

```
MOV BX,0
```

מקרה נוסף שיגרום ל-"1" בדגל האפס הוא כאשר פעולת **השוואה** מניבה תוצאת **שוויון**. לאחר ביצוע הפקודות:

```
MOV DH,5
CMP DH,5
```

דגל האפס יעלה ל-"1", מכיון שקיים שוויון.

כאשר אנו כותבים את הפקודות:

```
DEC CX
JZ FINISH
```

למעשה, אנו נעזרים בדגל האפס: הפקודה JZ בודקת את מצב הדגל, ועל פי מצבו מחליטה אם לקפוץ לתווית FINISH.

כאשר נכתוב את הפקודות הבאות:

```
CMP DL,4
JE GOOD
```

נבחין שתוכנת DEBUG **שינתה** את הפקודה JE לפקודה JZ. שתי הפקודות הללו **זהות**! כלומר, ניתן לכתוב כל אחת מהן, כי המשמעות זהה.

כאשר מבצעים פעולת השוואה CMP, מתבצעת "הפחתה מדומה". בדוגמה שלפנינו מופחת 4 מערך אוגר DL, ללא שינוי הערך של DL; האוגר היחיד המושפע מביצוע הפקודה CMP הינו אוגר הדגלים. אם DL היה שווה ל-4, תוצאת "ההפחתה המדומה" הינה 0, ולכן דגל האפס יעלה ל-"1".

תרגילים

באילו מהמקרים הבאים יעלה דגל האפס ל-"1"?

א. MOV DL,0

ב. MOV AL,5
SUB AL,5

ג. MOV BL,0FFH
ADD BL,1

ד. MOV CX,36H
CMP CX,'6'

ה. MOV AH,-2
CMP AH,0FEH

דגל הסימן

דגל זה נקרא **Sign Flag**, או בקיצור: S.F. כאשר הערך המתקבל כתוצאה של פעולה חשבונית, או לוגית, הוא ערך שלילי, דגל זה עולה ל-"1". אם הערך חיובי, הדגל יהיה "0".

מספרים שליליים נרשמים במחשב בשיטת **המשלים ל-2**. על עקרונות הייצוג של מספרים שליליים נעמוד להלן.

עיקרי השיטה: הסיבית השמאלית ביותר (MSB - Most Significant Bit - הסיבית הבכירה) מציינת את הסימן: "0" מציין מספר חיובי, "1" מציין מספר שלילי.

www.hod-ami.co.il

דוגמאות

1. הערך 6 הינו חיובי.
2. הערך 78H הינה חיובי.
3. הערך 83H הינו **שלילי** (!).

   נסביר זאת: נהפוך את הערך 83H לבסיס 2, כדי לבדוק את הערך של הסיבית החשובה ביותר (משמאל). נקבל את המספר הבינארי 0011 1000.

   אנו רואים שהסיבית השמאלית ביותר הינה "1" ועל כן, זהו מספר שלילי!
4. הערך 99H הוא שלילי. נסביר זאת: הערך 99H בייצוג בינארי הינו 1001 1001. הסיבית השמאלית ביותר הינה "1" ועל כן, לפנינו מספר שלילי.

שיטת המשלים ל-2

לצורך הדיון נסביר בקצרה את שיטת המשלים ל-2 (2's complement) לייצוג מספרים שליליים. באמצעות שיטה זו ניתן להפוך ייצוג של מספר חיובי למספר שלילי, ולהיפך. כדי להסביר השיטה נשתמש בדוגמה: נרשום את המספר השלילי -26H.

1. נכתוב את המספר **החיובי** 26H במספר בינארי (בסיס 2).

   נקבל את הערך הבינארי: 0010 0110.
2. נהפוך, באמצעות שיטת המשלים ל-2, את המספר החיובי הזה למספר שלילי. כדי לעשות זאת, נפעל בסדר הבא:

   ❖ נעתיק החל מצד ימין את כל הסיביות, עד שתופיע לראשונה הסיבית "1", וגם אותה נעתיק.

בדוגמה שלנו, נעתיק מתוך המספר: 0110 0010

את הסיביות: 10. . . . . .

(הנקודות מציינות את הסיביות שלא העתקנו).

❖את שאר הסיביות נהפוך: סיבית "0" תועתק כ-"1" ולהיפך, סיבית "1" תועתק כ-"0":

נקבל: 1101 1010

3. נייצג את המספר הבינארי שהתקבל, על פי בסיס 16.

נקבל את הערך **DAH**, אשר מייצג במחשב את הערך השלילי **-26H**.

דוגמאות

1.א. נרשום במחשב את הערך 2-. פתרון:

❖הערך 2 בייצוג בינארי הינו: 0000 0010

❖נכתוב אותו כמספר שלילי בשיטת המשלים ל-2:

א. נעתיק עד להופעת "1" ראשון: 10. . . . . .

ב. נהפוך את שאר הסיביות: 1111 1110

❖נתרגם לבסיס 16, והערך שנקבל הוא FEH.

2. מהו הערך המוצב לאוגר AL בפקודה MOV AL,0FFH?

❖כאן אנו מתבקשים לבצע פעולה הפוכה: לרשום מהו הערך של המספר הרשום במחשב.

נכתוב את הייצוג הבינארי של המספר FFH: 1111 1111

ניתן לראות שזהו מספר שלילי, כי הסיבית השמאלית היא "1".

❖נשתמש בשיטת המשלים ל-2, כדי להפוך את הערך השלילי לערך חיובי. כאמור, אפשר להפוך מספר שלילי לחיובי באמצעות שיטה זו.

א. נעתיק עד להופעת-אורח של "1": 1 . . . . . .

ב. נהפוך את שאר הסיביות: 0000 0001

❖נכתוב את התוצאה בייצוג הקסדצימלי, ונקבל את המספר "1". כלומר, הערך 0FFH הינו 1-, מכיון שהתוצאה ההפוכה היתה 1.

בדוק את פתרונך לתרגילים שנית בנושא דגל האפס!

www.hod-ami.co.il

הסבת מספרים מבסיס הקסדצימלי לבסיס בינארי

ראינו שכדי למצוא ערכים שליליים או חיוביים, דרוש ידע בביצוע הסבות מבסיס הקסדצימלי לבסיס בינארי, ולהיפך. בנוסף, לצורך טיפול בפקודות רבות נוספות שנלמד בהמשך, יש לשלוט היטב בפעולות ההסבה האלו.

כדי להסב מספר המיוצג בבסיס הקסדצימלי למספר המיוצג על פי בסיס בינארי, יש להפוך **כל ספרה** הקסדצימלית ל-4 ספרות בינאריות.

לדוגמה, מספר הקסדצימלי 56 יוצג כך בבסיס 2:

❖את הספרה 6 נהפוך ל-4 ספרות בינאריות: 0110

❖את הספרה 5 נהפוך ל-4 ספרות בינאריות: 0101

נצרף את שתי קבוצות המספרים ונקבל את המספר הבינארי: 0110 0101
(הרווח שבין שתי הקבוצות במספר הוא לצורך בהירות הכתיבה בלבד!).

כדי להימנע מהצורך לזכור את הסבת המספרים בשני הכיוונים, תוכל להיעזר בטבלה שלהלן:

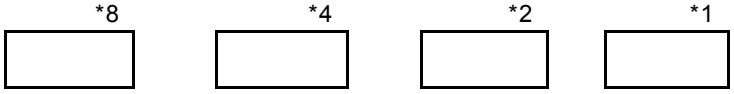
דצימלי בסיס 10	הקסדצימלי בסיס 16	בינארי בסיס 2
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

אינך צריך לזכור את הטבלה, אולם עליך לדעת לבצע בעצמך את התרגום של כל ספרה הקסדצימלית ל-4 ספרות בינאריות.

השיטה לביצוע פעולה זו: לכל ספרה **בינארית** יש משקל, או ערך מיקומי (כמו לספרות בבסיס עשרוני). במילים אחרות, הספרה הימנית ביותר תוכפל ב-1 (2^0), הספרה השנייה מימין תוכפל ב-2 (2^1), הספרה הבאה תוכפל ב-4 (2^2) והספרה השמאלית ביותר מבין 4 הספרות תוכפל ב-8 (2^3). **סכום** המכפלות של 4 הספרות שווה לערך הספרה ההקסדצימלית.

נבהיר זאת באמצעות דוגמה: כיצד נרשום את המספר הבינארי המייצג את הערך ההקסדצימלי 9?

תשובה: עלינו לחפש מספר בין 4 סיביות (ספרות בינאריות), אשר סכום המכפלות שלו יתן את הערך 9.



נראה שאם נציב 1 ב-"תא 8*" ו-1 ב-"תא 1*", נקבל:

$$1*8 + 1*1 = 9$$

באותו אופן נוכל להסב כל ספרה הקסדצימלית ל-4 ספרות בינאריות. למשל, כדי להסב את הספרה F בבסיס 16, נבצע:

- הספרה ההקסדצימלית F שווה בערכה ל-15. נמלא את הריבועים כך שייתנו סכום מכפלות השווה ל-15.
- נרשום 1 ב"תא 8*". הסכום כעת הינו 8, חסר עוד 7 כדי להגיע ל-15.
  נרשום 1 ב-"תא 4*". הסכום גדל כעת ב-4, ולכן נותר להוסיף 3, כדי להגיע ל-15.
  נרשום 1 ב-"תא 2*" והסכום גדל ב-2, ונותר 1.
  נרשום 1 ב-"תא 1*". כעת הסכום הגיע ל-15.

המספר שקיבלנו: 1111. זהו הייצוג הבינארי של F.

זכור, כל ספרה הקסדצימלית תהפוך בנפרד ל-4 ספרות בינאריות!

תרגילים

1. תרגם את המספרים הבאים הנתונים בבסיס 16, לבסיס בינארי:

א.	2	ו.	A7
ב.	D	ז.	E0
ג.	93	ח.	14C
ד.	25	ט.	D6
ה.	81	י.	125

2. אילו מהמספרים הבאים, שניתנים בייצוג הקסדצימלי (בסיס 16) הם שליליים?

א.	29	ד.	A2
ב.	78	ה.	99
ג.	90	ו.	0F

הסבת מספר בינארי לייצוג הקסדצימלי

כדי להפוך מספר בינארי למספר בייצוג הקסדצימלי, יש להפוך **כל ארבע** סיביות, החל מימין, לספרה הקסדצימלית **בודדת**.

דוגמאות

1. נתון המספר הבינארי 0100 0111 (הרווח הוא לשם בהירות בלבד):
   - 4 הספרות הימניות "0100" מייצגות את הספרה 4 בבסיס 16.
   - 4 הספרות הבאות "0111" מייצגות את הספרה 7.

   לכן, המספר הוא "74" בבסיס 16.

2. נתון המספר הבא בבסיס 2: 10101010110. מהו הייצוג שלו בבסיס 16?
   - רביעיית הספרות הראשונה מימין: 0110 (הספרה 6).
   - הרביעיה הבאה: 0101 (הספרה 5).
   - הרביעיה הבאה כוללת רק 3 ספרות: 101.

   כאשר חסרות ספרות לרביעיה, נשלים אותה ב"אפסים" **משמאל** ונקבל: 0101 (הספרה 5).

   המסקנה היא שהמספר הוא 556 בייצוג הקסדצימלי (בסיס 16).

3. מהו הייצוג בבסיס 16 של המספר הבינארי 11010111?
   - הרביעיה הראשונה מימין (0111), היא הספרה 7.
   - הרביעיה הבאה (1101), היא הספרה D.

   הייצוג ההקסדצימלי (בסיס 16) של המספר הוא: D7.

www.hod-ami.co.il

תרגילים

שים לב: כל המספרים מיוצגים בבסיס 16, אלא אם יצוין אחרת.

1. איזה מספר גדול יותר: 78 או 88?
2. איזה מספר נקבל (בבסיס 16), אם נחסר 2 מ-0?
3. כיצד יוצג המספר השלילי 45H-?
4. מה התוצאה של החישוב: FE+5?
5. מה קורה כאשר מוסיפים למספר 7F את המספר 2?
6. כיצד ייתכן שהמספר הבינארי 1100 הינו חיובי, למרות שהסיבית השמאלית ביותר שלו הינה "1"?
7. תן דוגמאות לפקודות לוגיות, הגורמות לדגל הסימן לעלות ל-"1".

דגל הזכור (נשא)

דגל הזכור (נשא) נקרא גם **Carry Flag**, או בקיצור: C.F. הדגל עולה ל-"1", כאשר בפעולה חשבונית או פעולה לוגית קיים **זכור** (carry), או **לווה** (borrow). כלומר, נוצר מצב שבו סיבית "1" "יוצאת" מחוץ למסגרת המקום המוקצב למספר. נראה זאת בדוגמאות שבהמשך.

דוגמאות

א. הפקודות הבאות יגרמו לדגל הזכור לעלות ל-"1":

```
MOV DL,0FFH
ADD DL,1
```

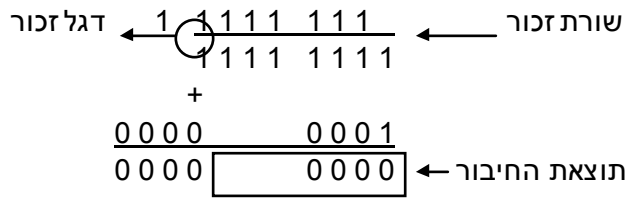
כדי להבין מדוע יש זכור, יש להפוך את המספרים לבסיס 2:

- המספר FF בייצוג בינארי: 1111 1111
- המספר 1 בייצוג בינארי: 0000 0001

נחבר שני מספרים אלה. מכיר שבפעולת חיבור מתקיימים הכללים הבאים:

חיבור הערכים	תוצאה	זכור
0 + 0	0	0
0 + 1	1	0
1 + 1	0	1
1 + 1 + 0	0	1
1 + 1 + 1	1	1

נחזור לחיבור שבדוגמה זו:



הסבר: מתחילים לחבר מימין, כמקובל. חיבור 1 עם 1 בעמודה הימנית, נותן תוצאה 0 (הנרשמת למטה) וזכור "1", הנרשם למעלה, בעמודה **הבאה**.

בעמודה הבאה מחברים 3 סיביות: את הזכור מהעמודה הקודמת (1), את הסיבית 1 והסיבית 0. התוצאה בעמודה זו: 0. הזכור המועבר לטור הבא "1".

בעמודה האחרונה משמאל, תוצאת החיבור הינה 0, אך לזכור "1" אין מקום! לפיכך,הזכור **נשמר** באוגר הדגלים, בסיבית השמורה ל**דגל הזכור**.

ב.יש פקודות המזיזות את תוכן האוגר ימינה או שמאלה (על פעולת ההזזה, shift, נלמד בהמשך). פקודות אלו משפיעות על דגל הזכור.

לדוגמה, אוגר CL מכיל את המספר הבינארי: 1001 1101

פקודת הזזה שמאלה תגרום להזזת כל סיבית שמאלה: 101 0011 1

כך נראה שהסיבית השמאלית ביותר (שערכה 1), **הוצאה** אל מחוץ לאוגר ו**הועתקה** לדגל הזכור. במקרה זה, דגל הזכור יהיה "1".

אם נבצע הזזה ימינה של המספר הזה: 1110 1000, נקבל: 0 0111 100.

במקרה זה יקבל דגל הזכור ערך "0", כי הסיבית שהוצאה היא 0.

דגל הגלישה

דגל הגלישה נקרא גם **Overflow Flag**, או בקיצור: O.F. כאשר אופרנד היעד, שהוא האוגר או התא המקבלים את התוצאה, אינו גדול דיו כדי להכיל את התוצאה, עולה דגל הגלישה ל-"1".

עניין זה פשוט למדי, ניתן מספר הסברים בעזרת מספר דוגמאות.

דוגמה 1

נבחן את הפקודות האלו:

```
MOV CL,7FH
ADD CL,5
```

הפקודות מציבות באוגר CL את הערך 84H, שהוא תוצאת החיבור 7F+5.

האם דגל הגלישה יעלה ל-"1"?

התשובה היא כן! למרות שאוגר CL יכול להכיל את הערך 84H התבצעה גלישה, מכיון שהוספנו למספר החיובי 7FH את הערך 5. במקום לקבל תוצאה של מספר חיובי גדול יותר, קיבלנו מספר שלילי! (הערך 84H מייצג מספר שלילי, כי הסיבית השמאלית ביותר שלו הינה 1). כלומר, אוגר CL אינו גדול דיו, כדי להכיל את התוצאה החיובית האמורה להתקבל, ולכן דגל הגלישה עולה ל-"1".

כאן המקום להסביר עניין חשוב:

- הפקודה MOV AL,84H - מציבה באוגר AL ערך שלילי.
- הפקודה MOV AX,84H - מציבה באוגר AX ערך חיובי.

הסבר: כאשר מוצב המספר 84H באוגר AL (מחצית אוגר), הוא מכיל את הסיביות: 1000 0100. הסיבית השמאלית ביותר היא 1, ולכן המספר הינו שלילי.

כאשר מוצב המספר 84H באוגר AX (אוגר שלם), נקבל: 0000 0000 1000 0100

ניתן לראות, שהסיבית השמאלית ביותר כעת היא 0, ולכן המספר מוצג כמספר חיובי!

כעת נכתוב כך את הפקודות שבדוגמה:

```
MOV AX,7FH
ADD AX,5
```

פקודות אלו יציבו את הערך **החיובי** 0084H באוגר AX, ועל כן דגל הגלישה במקרה זה יהיה "0".

דוגמה 2

נריץ את הפקודות הבאות:

```
MOV AH,0FFH
ADD AL,3
```

פקודות אלו יגרמו להצבת תוצאת החיבור FFH+3 באוגר AH, שהוא מחצית האוגר בלבד. איזה ערך יכיל אוגר AH? FFH שווה בערכו למספר 1- ולכן, כאשר מוסיפים לערך זה 3, מקבלים את הערך 2. והתשובה היא, שהאוגר AH יכיל את הערך 2.

האם תהיה גלישה? ובכן, הערך 2 יכול להיות מיוצג באוגר AH, כך שלא תהיה גלישה ודגל הגלישה יהיה "0".

דוגמה 3

צמד הפקודות הבא גורם להצבת תוצאת החיבור 81H+3 באוגר DL:

```
MOV DL,81H
SUB DL,3
```

הערך 81H הינו מספר שלילי, מכיון שהסיבית השמאלית ביותר של המספר הינה "1". כדי לדעת מהו ערך שלילי זה, נשתמש בשיטת המשלים ל-2.

המספר 81H בבסיס 2: 1000 0001

תרגום באמצעות משלים ל-2: 0111 1111

כלומר, הערך הינו 7FH-. אם נפחית 3 מערך זה נקבל את הערך השלילי 82H-.

לא ניתן להציג ערך כזה באמצעות אוגר DL, מכיון שאין מקום במחצית האוגר לייצוג המספר והסימן, ולכן תהיה גלישה והדגל יהיה "1".

www.hod-ami.co.il

הצגת הדגלים על ידי DEBUG

ניתן לראות את הדגלים על ידי ביצוע ההוראה **R** (Register) של התוכנה DEBUG: הדגלים מופיעים בשמותיהם הקצרים, כפי שנסביר להלן.

כדי לשנות את מצבו של דגל אחד או יותר נשתמש בהוראה **RF** (Register Flags). כל זוג אותיות לועזיות מציינות את מצב אחד הדגלים. בדרך כלל, האותיות מרמזות על שם הדגל, למשל:

סימון	משמעות הסימון	מצב הדגל
CY	Carry Yes	דגל הזכור במצב "1"
NC	No Carry	דגל הזכור במצב "0"
ZR	Zero	דגל האפס במצב "1"
NZ	No Zero	דגל האפס במצב "0"

לפניך טבלת סימוני הדגלים בתוכנת DEBUG:

	אפס	סימן	זכור	גלישה	כיוון	פסיקות	זכור-עזר	זוגיות
במצב "1"	ZR	NG	CY	OV	DN	EI	AC	PE
במצב "0"	NZ	PL	NC	NV	UP	DI	NA	PO

שים לב, שדגל המלכודת (דגל צעד-יחיד) אינו מוצג על ידי DEBUG.

ניתן לשנות מצב של דגל אחד או יותר. נניח שדגל האפס הינו במצב "1" וברצוננו לשנותו ל-"0". הפעולות שנבצע:

- נכתוב את ההוראה **RF**, שמשמעותה: הצג ואפשר שינוי של אוגר הדגלים. המצב הנוכחי של הדגלים (הסיביות) באוגר הדגלים יוצג ברשימה. כל דגל מתואר על ידי שתי אותיות, לפי הטבלה שלעיל.
- נכתוב את אותיות ההוראה NZ, כלומר את דגל האפס ב-"0", ונקיש Enter.
- נכתוב שוב **RF** ונראה שהדגל השתנה. נסיים בהקשת Enter.

בשיטה זו אפשר לשנות מספר דגלים בפעולה אחת. לדוגמה, כדי לאפס את דגלי הסימן, הפסיקות והכיוון, נכתוב:

- ההוראה: RF.
- סימון מצב הדגלים: DI, PL, UP ו-Enter.

אין חשיבות לסדר הכתיבה של הדגלים שעומדים לשנות!

תרגילים ופתרונות לדוגמה

נציג עתה מספר תרגילים ופתרונות הקשורים לנושא הדגלים. בכמה מהם יוצגו גם פקודות חדשות.

תרגיל 1

המטלה: כתיבת תוכנית לבדיקת תוכן תא הזיכרון שכתובתו 270H.

אם ערכו שלילי, יש להציב 1 באוגר AX.
אם ערכו חיובי או 0, יש להציב 0 באוגר AX.

פתרון א': בתחילה נקבע את אוגר AX כ-0. לאחר מכן, תבוצע השוואה בין ערך התא ל-0. אם הערך אינו קטן מ-0, התוכנית תסתיים. אחרת, יוצב 1 באוגר AX. בתוכנית זו נשתמש ב**פקודה JL** (Jump Less: קפוץ אם קטן) ונוסיף תנאי שלילה, Not: **JNL**.

התוכנית (001.asm)**:**

```
CSEG SEGMENT
    ASSUME CS:CSEG,DS:DSEG
MAIN:   MOV AX,CSEG
        MOV DS,AX
        MOV AX,0
        MOV BX,270H
        MOV DL,[BX]
        CMP DL,0
        JNL POS
        MOV AX,1
POS:    NOP
CSEG ENDS
END MAIN
```

קפוץ אם לא קטן

פתרון ב': למדנו שהמספר 7F (בבסיס 16, כמובן) הינו המספר החיובי הגדול ביותר. לכן, אם נשווה את ערך התא ל-80H ונמצא מספר נמוך יותר, זהו מספר חיובי. כעת נשתמש בפקודות אלו:

```
CMP DL,80H
JL POS
```

נמצא, שעבור הערך 71H למשל, לא תתבצע קפיצה, כי 71H אינו קטן מ-80H! לכן הפקודה **JL** אינה מתאימה. במקומה ניתן לרשום את הפקודה **JB** (Jump Below: קפוץ אם נמוך מ-). פקודה זו מתייחסת למיקום. כך למשל, המספר 74H נקטן מהמספר 99H; והאות "K" נמוכה בערכה מהאות "Z".

נבחן את התוכנית לפתרון זה (002.asm):

```
CSEG SEGMENT
   ASSUME CS:CSEG,DS:CSEG
FIRST:          MOV AX,CSEG
        MOV DS,AX
        MOV AX,0
        MOV BX,270H
        MOV DL,[BX]
        CMP DL,80H
        JB POSIT                    jump if below 80H
        MOV AX,1
POSIT:  NOP
CSEG ENDS
END FIRST
```

פתרון ג': נבדוק את דגל הסימן: אם הוא "1", לפנינו מספר שלילי. כדי להשפיע על מצב הדגל, עלינו לבצע פעולה חשבונית. אחת האפשרויות היא להוסיף 1 למספר, ולאחר מכן לחסר ממנו 1. כך יישאר המספר בערכו המקורי ודגל הסימן יכיל "1" עבור ערך שלילי או "0" עבור ערך חיובי.

ה**פקודה JS** (Jump Sign: קפוץ אם דגל הסימן הוא "1") שבוחנת אם התוצאה שלילית, תשמש אותנו למטרה זו. בתוכנית זו נשתמש בתנאי השלילה: **JNS** - אם המספר אינו (Not) שלילי. להלן התוכנית (003.asm):

```
CODE SEGMENT
  ASSUME CS:CODE,DS:CODE
BEGIN:  MOV AX,CODE
        MOV DS,AX
        MOV AX,0
        MOV BX,270H
        MOV DL,[BX]
        ADD DL,1                    מוסיפים 1 למספר
        SUB DL,1                    מפחיתים 1
        JNS POSIT                   אם המספר אינו שלילי
        MOV AX,1
POSIT:  NOP
CODE ENDS
END BEGIN
```

www.hod-ami.co.il

תרגיל 2

כתוב תוכנית המחברת את הערכים של האוגרים AL ו-BL ובודקת אם קיים זכור. אם יש זכור, הצב 1 באוגר CX; אחרת, אפס את האוגר CX.

בתוכנית זו נשתמש ב**פקודה JC** (Jump Carry: קפוץ אם קיים זכור). גם כאן, כמו בדוגמה הקודמת, נשתמש בתנאי השלילה Not. התוכנית בשלמותה (004.asm):

```
CODE SEGMENT
   ASSUME CS:CODE
COOL:   MOV CX,0
        ADD AL,BL
        JNC FIN                     קפוץ אם אין זכור
        MOV CX,1
 FIN:   NOP
CODE ENDS
END COOL
```

הערה לתוכניות הדוגמה:

בוודאי שמת לב, שבכל התוכניות עד כה הוספנו בפקודה ASSUME גם את אוגר DS. הפקודות הראשונות גם אתחלו את אוגר DS:

```
MOV CODE
MOV DS,AX
```

בדוגמה 2 לא עשינו זאת, מכיון שכאשר בתוכנית **אין פנייה לזיכרון**, **אין צורך** לאתחל את אוגר DS. אין זו טעות להוסיף את הפקודות המתייחסות לאוגר זה כפי שעשינו בתוכניות שונות עד כה, אולם ניתן היה להשמיט אותן ללא פגיעה בתוכנית.

תרגילים

1. כתוב תוכנית שתספור כמה מספרים שליליים נמצאים בתאי זיכרון שבכתובות 600H עד 610H. התוצאה תוצב בתא שכתובתו 900H.
2. ציין עבור כל אחת מהפקודות, או מקבוצת הפקודות שלפניך כיצד מושפע אחד מהדגלים:

א.
```
MOV AL,82H
```

ב.
```
CMP AL VOM
SUB DL,36H
```

ג.
```
ADD AL,' '
CMP DL,20H
```

ד.
```
MOV AL,6
ADD AL,31H
CMP AL,'7'
```

ה.
```
MOV AL,0F0H
ADD AL,20H
```

ו.
```
MOV CH,66H
ADD CH,66H
```

www.hod-ami.co.il

סיכום פקודות הקפיצה המותנית הנוספות

- JB (Jump if Below) - קפוץ אם נמוך (75H נמוך מ- 85H).
- JA (Jump if Above) - קפוץ אם מעל (88H גבוה מ-60H לעומת JG: 88H קטן מ-60H).
- JNB (Jump if not Below) - קפוץ אם לא נמוך.
- JNA (Jump if not Above) - קפוץ אם לא מעל.
- JC (Jump if Carry) - קפוץ אם יש זכור (נשא).
- JNC (Jump if not Carry) - קפוץ אם אין זכור (נשא).
- JS (Jump if Sign) - קפוץ אם יש סימן (כלומר, התוצאה שלילית).
- JNS (Jump if not Carry) - קפוץ אם אין סימן (כלומר, התוצאה חיובית).