“Basics of microcontroller”

Lecture 4.

Microprocessor operation, machine cycles, instruction cycles, memory write and read cycles

Lecturer: Onon Otgonbaatar, MD.

ASCII үйлдлүүд

ASCII → 2-тын форматад, 2-т → ASCII форматад шилжүүлэх командууд байдаг.
SAM → 53.41.4DH
1234 → 31.32.33.34H

1. AAA (ASCII adjust for addition) – ASCII тэмдэгтүүдийн нийлбэрийг засахад хэрэглэх команд

   Өмнөх нэмэх үйлдлийн үр дүнд үүссэн AL регистрийн утган дээр үйлдлээ хийнэ. Ингэхдээ AL регистрийн ахлах 4 битийг 0 болгох ба хэрэв бага 4 бит нь 9-өөс их (A → F гэсэн утгууд байх юм бол) эсвэл AF=1 бол AH регистрийн утгыг нэгээр нэмэгдүүлээд, AL регистрийн утга дээр 6-г нэмнэ. Мөн CF=AF=1 болно.

   Жишээ нь:
   (AX)=38, (BX)=34 гэсэн утгууд байна. Эдгээр нь 8 ба 4-ийн тоонуудын ASCII кодууд юм.

```
ADD   AL, BL      ; (AL)=6CH
AAA               ; (AH)=0102 болгон нийлбэрийг өөрчилнө.
OR    AX, 3030h   ; (AH)=3132 болгон нийлбэрийг засна.
```

   38h+34h=6Ch нь ямар ч тоо биш, харин “l” үсгийн ASCII код болно. Үүнийг тоогоор нь бодвол 8+4=12 болох бөгөөд 12 гэсэн тооний ASCII кодууд нь 31, 32 юм. Иймд хувиргаж нийлбэрийг засав.

---

2 ширхэг 3 байт ASCII тоонуудыг нэмж, 4 байт нийлбэр гаргах жишээ программ

- Орон шилжилтийг тооцохын тулд ADC командыг ашиглана.
- CFC нь CF-ийг 0 болгоно.
- AH-ийг давталт бүр дээр цэвэрлэнэ.
- Флагийг хадгалахын тулд PUSHF үйлдлийг OR үйлдлийн өмнө хийгээд, POPF үйлдлийг дараа нь хийж болно.

```
.data
        ASC1        DB      ‘578’
        ASC2        DB      ‘644’
        ASC3        DB      ‘0000’
.code
        CLC
        LEA         SI,     ASC1+2
        LEA         DI,     ASC2+2
        LEA         BX,     ASC3+3
        MOV         CX,     3
NEXT:
        MOV         AH, 0
        MOV         AL, [SI]
        ADC         AL, [DI]
        AAA
        MOV         [BX], AL
        DEC         SI
        DEC         DI
        DEC         BX
```

```
LOOP    NEXT
MOV     [BX], AH
RET
```

2. AAS (ASCII adjust for subtraction)- ASCII тэмдэгтүүдийн ялгаварыг засахад хэрэглэх команд

   Энэ нь AAA-тай төстэй ажиллагаатай. Өмнөх хасах үйлдлийн үр дүнд үүссэн AL регистрийн утган дээр үйлдлээ хийнэ. Ингэхдээ AL регистрийн ахлах 4 битийг 0 болгох ба хэрэв бага 4 бит нь 9-өөс их (A → F гэсэн утгууд байх юм бол) эсвэл AF=1 бол AH регистрийн утгыг нэгээр хорогдуулаад, AL регистрийн утганаас 6-г хасна. Мөн CF=AF=1 болгоно.

   Жишээ 1:
   ASC1=38H, ASC2=34H гэсэн утгууд байна. Эдгээр нь 8 ба 4-ийн тоонуудын ASCII кодууд юм.

```
MOV  AL, ASC1        ; (AX)=0038h гэсэн утгатай болно.
SUB  AL, ASC2        ; (AX)=0004H, AF bit=0 болно.
AAS                  ; (AX)=0004H, AF bit=0 болно.
```

Энэ жишээнд AAS үйлдлээр засвар хийх шаардлагагүй болох нь харагдана.

   Жишээ 2:

```
MOV     AL, ASC2     ; (AX)=0034h гэсэн утгатай болно.
SUB     AL, ASC1     ; (AX)=00FCH, AF bit=1 болно.
AAS                  ; (AX)=FF06H, AF bit=1 болно.
```

   AL-ийн бага 4 бит нь 0CH учир 0Ch-6=6-г AL регитсрт хийгээд, (AH)-1=00-1=FFh ба AF=CF=1 болно. FF06H=-4D юм.

3. AAM (ASCII adjust for multiplication) - ASCII тэмдэгтүүдийн үржвэрийг засахад хэрэглэгдэх команд

   ASCII тэмдэгтүүдийг нэг нэг байтаар нь л үржих боломжтой. Энэ үйлдэл нь AL регистрийн утгыг 10 (0AH)-т хуваан, (AH)-д коэффициентийг нь, (AL)–д үлдэгдлийг нь хийнэ. Жишээ нь:
   (AL)=35h, (CL)=39h гэсэн утгууд байг.

```
AND     CL, 0FH      ; (CL)=09H болно.
AND     AL, 0FH      ; (AL)=05H болно.
MUL     CL           ; (AL)=(AL)*(CL) үйлдлээр (CL)=002DH (45D)
AAM                  ; 2DH/0AH=> (AX)=0405H утгатай болно.
OR      AX, 3030H    ; (AX)=3435 утгатай болно.
```

4. AAD (ASCII adjust for division)- ASCII тэмдэгтүүдийн хуваасан үр дүнг засахад хэрэглэх команд

   AX регистрийн агуулж буй 2 байт утгыг 0-9 хоорондох утганд ASCII-гаар хуваах боломжтой. Үүний тулд AH регистрийн утгыг 10-аар үржүүлээд, үржвэрийг нь AL регистрийн агуулж буй утган дээр нэмнэ. AH регистрийн утгыг 0 болгоно.
   Жишээ нь:
   (AX)=3238H , (CL)=37H утгатай байхад доорх үйлдлүүдийг хийе.

```
AND     CL, 0FH         ; (CL)=07 болно.
AND     AX, 0F0FH       ; AX=0208 болно.
AAD                     ; 28D гэсэн утгыг AX=1Ch болгоно.
DIV     CL              ; 1Ch / 07h=0004=(AX)
```

Дээрх жишээнд AAD нь AH регистрийн утгыг 10-аар үржээд, 20(14H) утган дээр AL регистрийн утга болох 8-ийг нэмэхэд AH регистрт 01CH буюу 10-таар 28 гэсэн утга орно.

BCD ҮЙЛДЛҮҮД

BCD формат нь 10-тын тоон утгуудыг хэлнэ. Өөрөөр хэлбэл, бид тоонуудын ASCII формат 30h, 31h, …, 39h гэх мэтийг өмнө үзсэн. Харин эдний сүүлийн оронгуудыг авбал 0, 1, …, 9 болно. Эдгээрийг BCD формат гэнэ. Жишээ нь: 30393234 гэсэн ASCII тоонуудыг BCD хэлбэрт оруулбал, 0924 гэсэн утга болно.
BCD тооны урт нь ASCII тооны хагас нь байх нь тодорхой боллоо. Гэхдээ 0924 гэсэн тоо нь 10-тын тооллын системд учир 039Ch гэсэн 16-тын тоотой ижил болно.
Нэмэх, хасах үйлдлүүдийг BCD (binary coded decimal) өгөгдлүүд дээр хийдэг доорх командууд байна.

- DAA(Decimal adjustment for addition) – Нэмэх үйлдэл хийхэд 10-тын тоо болгон засах үйлдэл

BCD 2 тоог нэмэхэд AL регистрт гарах үр дүнг засна. Бага 4 битийн утга 9-өөс их эсвэл AF төлөвийн бит 1 бол энэ үйлдэл нь AL регистрийн утга дээр 6-г нэмэн, AF төлөвийн битийг 1 болгоно. Хэрэв AL регистрийн утга 9F-ээс их, эсвэл CF төлөвийн бит 1 бол энэ үйлдэл нь AL регистрийн утга дээр 60-ийг нэмээд, CF төлөвийн битийг 1 болгоно.

- DAS(Decimal adjustment for subtraction) - Хасах үйлдэл хийхэд 10-тын тоо болгон засах үйлдэл

BCD 2 тоог хасахад AL регистрт гарах үр дүнг засна. Бага 4 битийн утга 9-өөс их эсвэл AF төлөвийн бит 1 бол энэ үйлдэл нь AL регистрийн утгаас 6-г хасан, AF төлөвийн битийг 1 болгоно. Хэрэв AL регистрийн утга 9F-ээс их, эсвэл CF төлөвийн бит 1 бол энэ үйлдэл нь AL регистрийн утгаас 60-ийг хасаад, CF төлөвийн битийг 1 болгоно.

Тухайн тохиолдолд нэг л байт дээр үйлдлийг хийнэ.

Доорх жишээ программд 2 ширхэг ASCII тоонуудыг BCD формат уруу хөрвүүлээд тэднийг хооронд нь нэмнэ.
CONVERSION процедур нь тоог ASCII форматаас BCD формат уруу хөрвүүлнэ. 2 ширхэг ASCII байтаар 1 байт BCD өгөгдлийг гаргана.
ADDITION процедур нь BCD өгөгдлүүдийг хооронд нь нэмнэ. Эцсийн үр дүн: 127263 болохыг DEBUG программаар DUMP хийн, өгөгдлийн санах ойгоос харж болно.

```
TITLE BCDADD CONVERT ASCII TO BCD AND ADD
.model small
.stack 100H
.data
        ASC1        DB      '057836'
        ASC2        DB      '069427'
        BCD1        DB      '000'
        BCD2        DB      '000'
        BCD3        DB      4 DUP(0)
```

```
.code
        MAIN PROC NEAR
          mov    ax, @data
          mov    ds, ax
          mov    es, ax

                LEA         SI, ASC1+4          ; ASC1-ийн хаяг
                LEA         DI, BCD1+2          ; BCD1-ийн хаяг
                CALL        CONVERTION          ;хөрвүүлэх процедурыг дуудна.
                LEA         SI, ASC2+4          ; ASC2-ийн хаяг
                LEA         DI, BCD2+2          ; BCD2-ийн хаяг
                CALL        CONVERTION          ; хөрвүүлэх процедурыг дуудна.
                CALL        ADDITION            ; нэмэх процедурыг дуудна.
                RET
        MAIN ENDP

        ;-------ASCII –аас BCD форматад хөрвүүлэх
        CONVERTION      PROC
                MOV         CL, 04      ; шилжүүлэх фактор
                MOV         DX, 03      ; хөрвүүлэх үгийн тоо
        B20:
                MOV         AX, [SI]    ; 2 ASCII тэмдэгтийг авна.
                XCHG        AH, AL
                SHL         AL, CL      ; ASCII тэмдэгтийн
                SHL         AX, CL      ; 3-уудыг нь арилгана.
                MOV         [DI], AH    ; BCD оронгуудыг хадгална.
                DEC         SI
                DEC         SI
                DEC         DI
                DEC         DX
                JNZ         B20
                RET
        CONVERTION      ENDP
        ;-------ADD BCD NUMBERS
        ADDITION    PROC
                XOR         AH, AH      ; AH-ийг 0 болгоно.
                LEA         SI, BCD1+2  ; BCD тоонуудын
                LEA         DI, BCD2+2  ;хаягуудыг авна.
                LEA         BX, BCD3+3
                MOV         CX, 03      ; 3 байттай ажиллана.
                CLC
        C20:
                MOV         AL, [SI]    ; BCD1-ийг авна.
                ADC         AL, [DI]    ; BCD2-ийн нэмнэ.
                DAA                     ; 10-тад тохируулна.
                MOV         [BX], AL    ; BCD3-т хадгална.
                DEC         SI
                DEC         DI
                DEC         BX
                LOOP            C20     ; 3 удаа давтана.
                RET
        ADDITION    ENDP
```

END MAIN

ASCII форматаас 2-тын (BINARY) формат уруу эсвэл BINARY форматаас ASCII форматад хөрвүүлэх нь

ASCII болон BCD форматууд нь багахан хэмжээний өгөгдлүүдэд хийгдэхэд тохиромжтой. Харин 2-тын формат нь арифметик үйлдлүүдэд илүү тохиромжтой. ASCII форматаас 2-т уруу хөрвүүлэх нь BCD-ээс 2-т уруу хөрвүүлэхээс харьцангуй хялбар.
Хөрвүүлэлт хийхдээ эхлээд ASCII байтын баруун талын оронгуудыг аван, зүүн талын 3-уудыг хасна. Дараа нь үлдсэн оронгуудаа харгалзуулан 1, 10, 100, 1000 (1, A, 64, 3E8) гэх мэтээр үржүүлнэ. Жишээ нь:
1234 (BCD формат) → 04D2H (2-тын буюу 16-тын формат)
1234 гэсэн тооны ASCII формат нь 31H, 32H, 33H, 34H болно. Эдгээрээс сүүлийн оронг авч, 3-уудыг хасвал 01, 02, 03, 04 болно.
Үүнийг харгалзах оронгуудын коэффициентүүдээр үржвэл:

04*1	=4	4h
03*10	=30	1Eh
02*100	=200	C8h
01*1000	=1000	3E8h
Үр дүн:	1234	04E8h

Жишээ программд хийснээр ASCIIBIN процедур нь ASCII тоог 1234 болгон 16-т уруу хөрвүүлнэ. ASCLEN –д 4 гэсэн утгаар ASCII тооны уртыг өгнө. Программын эхэнд ASCII талбарын хаягийг SI регистрт хийн ASCVAL-1 утгыг олгон. BX регистрт уртыг хийнэ. B20 хаягт байгаа MOV AL, [SI+BX] гэсэн үйлдэл нь ASCVAL+3 буюу хамгийн баруун талын байтыг заана. Давталт бүрт BX регистрийн утга 1-ээр хорогдох ба дараагийн зүүн талын байтыг заана. Энэ тохиолдолд CX-ийг хэрэглэж болохгүй, учир нь LOOP командыг хэрэглэж чадахгүй болно. Давталт бүрт MULT10-ыг 1, 10, 10 гэх мэтээр үржүүлнэ. Үржигч нь SI буюу DI регистрүүдэд хадгалагдана.

04D2 гэсэн 16-тын тоог ASCII форматад хөрвүүлэхдээ доорх үйлдлийг хийнэ.16-тын тоогоо 10-т хуваах ба үр дүнг 10-аас бага тоо гартал үргэлжлүүлэн хуваана. Үлдэгдэлүүд нь 0-9 хоорондох тоо байх ба эдгээрээр ASCII тоог үүсгэж болно. Жишээ нь:

Үйлдэл	Коэффициент	Үлдэгдэл
04D2H / A(10)	=7BH	4
7BH / A	=0CH	3
0CH / A	=1	2

Коэффициент нь 1 гарч 10-аас бага тоо гарсан тул үйлдэл дууссан.
Үр дүн 1234 болно. Жишээн программд BINASCII процедур уг хөрвүүлэлтийг хийнэ.

Программ

```
TITLE EXCONV conversion of ASCII and BINARY formats
.model small
.stack
        ORG 100H
.data
        ASCVAL      DB     '1234'   ; өгөгдлүүд
        BINVAL      DW     0
        ASCLEN      DW     4
        MULT10      DW     1
```

```
.code
        MAIN PROC NEAR
          MOV    AX, @DATA
          MOV    DS, AX
          MOV    ES, AX
        CALL ASCIIBIN     ; ---ASCII-г BINARY-д хөрвүүлэх дэд процедурыг дуудна.
        CALL BINASCII     ; ---BINARY-г ASCII-д хөрвүүлэх дэд процедурыг дуудна.
             RET
        MAIN ENDP
        ASCIIBIN     PROC
             MOV          CX, 10              ; үржүүлэх фактор
             LEA          SI, ASCVAL-1        ; ASCVAL-ийн хаяг
        --   MOV          BX, ASCLEN          ; ASCVAL-ийн урт
        B20:
             MOV          AL, [SI+BX]         ; ASCII тэмдэгтийг сонгоно.
             AND          AX, 000FH           ; Бага 4 битийг сонгоно.
             MUL          MULT10              ; 10-аар үржинэ.
             ADD          BINVAL, AX          ; 16-тын утга дээр нэмнэ.
             MOV          AX, MULT10          ; дараагийн 10-ын факторыг
             MUL          CX                  ;тооцоолно.
             MOV          MULT10, AX
             DEC          BX
             JNZ          B20                 ; сүүлийн тэмдэгт биш бол үсэрнэ.
             RET
        ASCIIBIN     ENDP
        ;----CONVERT BINARY TO ASCII
        BINASCII     PROC
             MOV          CX, 0010            ; хуваах фактор
             LEA          SI, ASCVAL+3        ; ASCVAL-ийн хаяг
             MOV          AX, BINVAL          ; 2-тын тоо
        C20:
             CMP          AX, 0010            ; утга 10-аас бага уу?
             JB           C30                 ; тийм бол гарна.
             XOR          DX, DX              ; DX-ийг 0 болгоно.
             DIV          CX                  ; DX:AX-ийн утгыг CX-д хуваана.
             OR           DL, 30H
             MOV          [SI], DL            ; ASCII тэмдэгтийг хадгална.
             DEC          SI
             JMP    C20
        C30:
             OR           AL, 30H             ; Сүүлийн коэффициентийг ASCII
             MOV          [SI], AL            ;тэмдэгт болгон хадгална.
             RET
        BINASCII     ENDP
          END    MAIN
```

МАТРИЦТАЙ АЖИЛЛАХ ҮЙЛДЛҮҮД (table processing)

Матрицан өгөгдөлтэй ажиллах олон хэрэглээ гардаг. Жишээ нь: нэрс, үзүүлэлт, үнэ, тоо ширхэг гэх мэт. Өмнө үзсэн командууд дээр нэмээд XLAT гэсэн командыг үзэхэд матрицан өгөгдөлтэй ажиллах боломжтой болно.

Матриц тодорхойлох жишээ нь:

```
STACK     DW    64     DUP(?)                          ; 64 word урттай стек
MONTAB    DB    'JAN', 'FEB', 'MAR', …, 'DEC'          ; сарууд
NUMTAB    DB    205, 208, 209, 212, 215, 219, ….       ; өрөөний дугаар

STOKTBL   DB    12, 'Computers'
          DB    14, 'Papers'
          DB    17, 'Diskettes'                        ; холимог мэдээлэл зарлаж болно.
```

Матрицад шууд хандах:

Доорх саруудын нэртэй матрицаас 3 дугаар элементэд шууд хандах зарчимыг авч үзье. 3 гэсэн тоог оруулахад үүнийг March гэж хөрвүүлье. Бүх саруудын нэрийг ижил урттай оруулахын тулд хамгийн урт нэртэй сар September нь 9 ширхэг тэмдэгттэй байна. Иймд бүх бичлэгүүд 9 тэмдэгттэй.

```
MONTAB    DB    'January..'        ; MONTAB+0 хаяг дээр
          DB    'February.'        ; MONTAB+9 хаяг дээр
          DB    'March….'          ; MONTAB+18 хаяг дээр
          …
```

Программд доорх үйлдлүүдийг хийнэ.

1. Оруулсан ASCII тэмдэгт 33-ийг 3 гэж хөрвүүлнэ.
2. Сараас нэгийг хасна. 03-1=02
3. Сарыг 9-өөр үржүүлнэ. 02*9=18
4. Энэ үржвэрийг MONTAB-ийн хаяг дээр нэмнэ. Ингэж шаардлагатай хаяг гарч ирнэ.

Программ

```
TITLE DIRECT Хүснэгтэд шууд хандах арга
.model small
.stack
        ORG 100H
.data
        THREE       DB     3
      MONIN   DB     '09'
        ALFMON      DB     '???', '$'
        MONTAB      DB     'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN'
                    DB     'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC'
.code
  MAIN:
      mov    ax, @data
      mov    ds, ax
      mov    es, ax
        CALL CONVERTION        ; 16-т уруу харвүүлэх дэд проц.
        CALL LOCATE            ; Сарыг байрлуулах дэд проц.
        CALL   DISP            ; Сарыг дэлгэцэнд хэвлэх дэд проц
```

```
      JMP    NNN
       ;----CONVERT ASCII TO BINARY
       CONVERTION      PROC
              MOV      AH, MONIN       ; сарыг тавина.
              MOV      AL, MONIN+1
              XOR      AX, 3030H       ; ASCII 3-уудыг арилгана.
              CMP      AH, 00          ; 01-09 мөн үү?
              JZ       C20             ; тийм бол дэд процедураас гарна.
              SUB      AH, AH          ; үгүй бол AH-ийг цэвэрлэнэ.
              ADD      AL, 10          ; тоог засна.
       C20:   RET
       CONVERTION      ENDP
       ;----LOCATE MONTH IN TABLE
       LOCATE      PROC
              LEA      SI, MONTAB
              DEC      AL              ; хүснэгтийг засна.
              MUL      THREE           ; AL-ийг 3-аар үржинэ.
              ADD      SI, AX
              MOV      CX, 03          ; 3-тэмдэгтийг шилжүүлнэ.
              CLD
              LEA      DI, ALFMON
              REP      MOVSB           ; 3 тэмдэгтийг шилжүүлнэ.
              RET
       LOCATE      ENDP
       ;----DISPLAY ALPHA MONTH
    DISP   PROC
              LEA      DX, ALFMON
              MOV      AH, 09          ; дэлгэцэнд хүссэн
              INT      21H             ;сарыг хэвлэнэ.
              RET
    DISP   ENDP
    NNN:
         MOV    AX, 4C00H
         INT    21H
 END    MAIN
```

- CMP командыг ашиглан хүснэгтээс хайлт хийх программ

Энэ программд барааны дугаар болон нэрийг агуулсан 6 ширхэг хос өгөгдлийг оруулсан. Барааны нэрс нь янз бүрийн урттай тэмдэгтүүд байгаа тул тэднийг ижил урттай болгох үүднээс богино нэрс дээр хоосон тэмдэгтүүдийг оруулж өгсөн. Хайлт хийх хэсэг нь STOCKNIN –д өгсөн утгыг хүснэгтэд байгаа дугааруудтай харьцуулах замаар ажиллана. Харьцуулалт зөрсөн бол дараагийн дугаартай харьцуулахын тулд хаягийг нэмнэ. Хэрэв ижил бол A30 хаяг хаяг дахь хэсэгт хийгдэх DESCRN –д утгыг хадгална. Энэ давталт 6 удаа хийгдэх ба хэрэв хайж буй дугаартай өгөгдөл олдоогүй бол программ алдааний хэсгийг дуудан, ажиллуулна.

```
TITLE TABSRCH1 TABLE SEARCH
.MODEL SMALL
.STACK
        ORG 100H
```

```
.DATA
        STOCKNIN   DW     '23'
        STOCKTAB   DB     '05', 'EXCAVATORS'
                          DB      '08', 'LIFTERS              '
                          DB      '09', 'PRESSES      '
                          DB      '12', 'VALVES         '
                          DB      '23', 'PROCESSORS'
                          DB      '27', 'PUMPS          '
        DESCRN             DB      10 DUP(?)
.CODE
        MAIN PROC NEAR
                MOV         AX, @DATA
                MOV          DS, AX
                MOV          ES, AX

                MOV         AX, STOCKNIN            ; дугаарыг авна.
                XCHG        AL, AH
                MOV         CX, 06                  ; оролтын өгөгдлүүдийн тоо
                LEA         SI, STOCKTAB            ; хүснэгтийн эхлэх утга
        A20:
                CMP         AX, [SI]                ; дугааруудыг харьцуулна.
                JE          A30                     ; ижил бол A30-уруу үсэрнэ.
                ADD         SI, 12                  ; үгүй бол хаягийг нэмэгдүүлнэ.
                LOOP        A20
                CALL        ERROR                   ; хүснэгтэд байхгүй.
           RET
        A30:
                MOV         CX, 05                  ; барааны нэрийн урт
                LEA         DI, DESCRN              ; барааны нэрийн хаяг
                INC         SI
                INC         SI                      ; барааны нэрийг хүснэгтээс
                REP         MOVSW                   ; хуулж бичнэ.
                RET
        MAIN ENDP
        ;----
        ERROR       PROC
        ;       <Алдааны мэдээг хэвлэх хэсэг>
                RET
        ERROR       ENDP
          END    MAIN
```

AX регистрт STOCKNIN гэсэн хайж байгаа дугаарыг авахад STOCKNIN нь хэдийгээр 3233 гэж тодорхойлогдсон боловч MOV командаар AX регистрт 3332 гэж орно. Иймд XCHG командаар 3233 болгон засна.

- CMPSB тэмдэгттэй ажиллах командыг ашигласан программ

```
TITLE TABSRCH2 TABLE SEARCH
.MODEL SMALL
.STACK
        ORG 100H
.DATA
```

```
        STOCKNIN   DW     '123'
        STOCKTAB   DB     '035', 'EXCAVATORS'; хүснэгтийн эх
                          DB     '038', 'LIFTERS              '
                          DB     '049', 'PRESSES      '
                          DB     '102', 'VALVES         '
                          DB     '123', 'PROCESSORS'
                          DB     '127', 'PUMPS          '
                          DB     ‘999’, 10 DUP(‘ ’) ; хүснэгтийн сүүлч
        DESCRN             DB     10 DUP(?)
.CODE
        MAIN PROC NEAR
                MOV    AX, @DATA
                MOV    DS, AX
                MOV    ES, AX

                CLD
                LEA            SI, STOCKTAB        ; хүснэгтийн эхлэх утга
        A20:
                MOV            CX, 03              ; харьцуулах байтын тоо
                LEA            DI, STOCKNIN        ; хайх утгын эхлэл
                REPE  CMPSB
                JE             A30                 ; ижил бол A30-уруу үсэрнэ.
                JA             A40                 ; above-хүснэгтэд байхгүй бол A40
                ADD            SI, CX              ; CX-ээр хаягийг нэмэгдүүлнэ.
                ADD            SI, 10              ; дараагийн хүснэгтийн утга
                JMP A20
        A30:
                MOV            CX, 05              ; 5 word урт
                LEA            DI, DESCRN          ; барааны нэрийн хаяг
                REP            MOVSW               ; хүснэгтээс хуулж бичнэ.
                RET
        A40:
                CALL ERROR
                RET
        MAIN ENDP
        ;---------------------------------
        ERROR          PROC
        ;       <Алдааны мэдээг хэвлэх хэсэг>
                RET
        ERROR          ENDP
          END    MAIN
```

STOCKTAB-ийн сүүлчийн утга нь 999 гэсэн дугаартай бөгөөд энэ нь хайлтыг дуусгах зорилготой. Хайлт нь дараах байдлаар хийгдэнэ.

Барааны дугаар	STOCKNIN харьцуулалт	
035	123	бага, дараагийн утгыг шалга
038	123	бага, дараагийн утгыг шалга
049	123	бага, дараагийн утгыг шалга
102	123	бага, дараагийн утгыг шалга
123	123	тэнцүү, өгөгдөл олдсон.

Үүний өмнөх жишээн дээр SI, DI регистрүүдийг ашиглан байт байтаар нь нэмэгдүүлэн харьцуулж шалгаж байсан. Харин энэ программд CX регистрт 3 гэсэн утга олгоод, SI регистрт 3, DI регистрт 0 утга олгосон. Хамгийн эхний байтын харьцуулалтаар (035:123) SI=04, DI=01 утгатай болох ба эхний 2 байт тэнцүү биш учир дараагийн өгөгдлийг харьцуулах хэрэгтэй болно. Үүний тулд SI=16, DI=0 гэсэн утгуудтай байх ёстой. Дараагийн өгөгдлийн хаягийг авахдаа доорх үйлдлийг хийнэ. CX регистр нь харьцуулагдаагүй үлдсэн байтуудын тоог агуулж үлдэнэ.

CMPSB үйлдлийн дараах SI утга: 04
Нэмэх нь CX регистрийн утга: 02
Нэмэх нь барааны нэрийн урт: 10

Дараагийн өгөгдлийн шилжлэг хаяг: 16

Дээрх байдлаар дараагийн элементийн эхлэх хаягийг тодорхойлно.

Textbook

1. Peter Abel, “IBM PC Assembler language and programming”, USA, 1987
2. Jim Mischel, "Macro magic with Turbo Assember ”, USA, 1993
3. William C.Runnion, “Structured programming in Assembly Language for the IBM PC”, Boston, 1988
4. Thomas A.Wadlow, “Memory resident programming on the IBM PC”, USA, 1987
5. Robert S.Lai, “Writing MS-DOS device drivers”, USA, 1987
6. E. Majigsuren, “IBM assembly language”, MGL, 2003
7. Muhammad Ali Mazidi, Janice Gillispie Mazidi, Rolin D. McKinlay, “The 8051 microcontroller and Embedded Systems Using Assembly and C”,USA, 2007