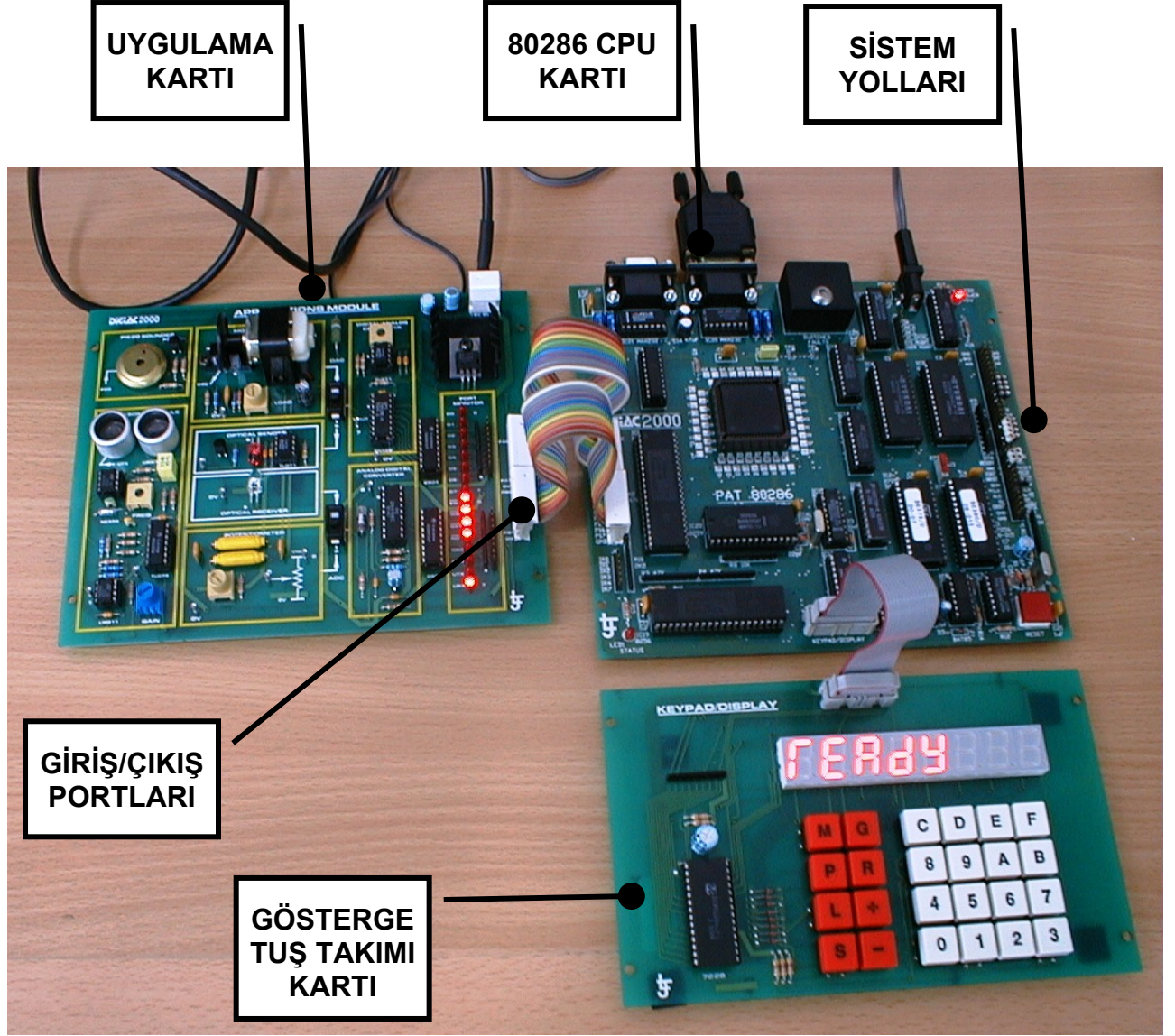


6. Mikroşlemcili Geliştirme Sistemleri



Bir mikroşlemcili geliştirme sistemi donanımının görünümü ve parçaları

DEBUG PROGRAMI

DEBUG program MSDOS işletim disketi içinde hazır bulunan bir 8086 line assembler / disassembler / debugger programıdır.

d:debug [filename.ext[parametreler...]]

yukarıdaki biçiminde bir MSDOS komut satırı yazılarak çalıştırılır.

DEBUG KOMUTLARI:

A adr.

Macro assemblerda kullanılan assembly dilini belleğe kodlar.

ör: A 100

CS:0100H'den itibaren assembly yazım işlemine geçilir.

Boş satır (sadece **Enter**) veya **Ctrl-C** ile çıkılır.

C alan adr.

Bellek bölgelerini karşılaştırarak farklılıklar gösterir.

ör: C 100 102 109

D adr.

D alan

Belleğin içeriğini hexadecimal ve ASCII olarak görüntüler.

ör: D 3000:0

DS=3000:0'den itibaren göster.

ör: D 0 100

DS:0'dan DS:100'e kadar göster.

E adr.

E adr. list.

Bellekteki bilgileri görerek değiştirmeyi sağlar.

ör: E 100

DS:100'den itibaren bilgileri tek tek göster ve dışarıdan girilen değişiklikleri yap.

ör: E 0 1 2 3 'ABCDE'

DS:0'dan itibaren 1, 2, 3, 'A', 'B', 'C', 'D', 'E' bilgilerini yerleştir.

F alan list.

Bellek bölgesini verilen bir sabit dizi ile doldur.

ör: F 0:10 20 41 'B'

0:10 adresinden 20 adresine kadar bellek bölgesini 41H, 42H, 41H, 42H, şeklinde doldur.

G

G=adr:adr adr

Bellekteki program koşturma.

ör: G

O anki CS:IP adresinde bulunan programı koştur.

ör: G=1B0C:0100 120

1B0C:0100 adresinde bulunan program 100 adresinden 120 adresine kadar koştur.

Son adres programın kırılma noktasıdır (breakpoint).

H değer değer

Verilen hexadecimal iki değer toplamını ve farkını hesaplar.

ör: H 22 20

42 2 sonucunu verir. (22+20=42 ve 22-20=2)

I portadr

Verilen I/O port adresindeki bilgiyi görüntüler.

L

L adr

L adr sürüc baş.sekt. sekt.sayısı

Disk veya disketten program yükleme komutu.

sürücü: A=0, B=1, ...

sekt.=sektör

Double side, 40 track, 9 sektör/track disket için sektör özellikleri:

0 Boot sektör.

1- 2 1.FAT bölgesi.

3- 4 2.FAT bölgesi.

5- 11 Root directory bölgesi.

12-719 Kullanıcı dosyalarının saklandığı data bölgesi.

ör: L 120

N komutu ile verilen program bulunulan sürücünden okuyarak CS:120 adresine yerleştir.

ör: L 100 1 2 3

B sürücüsünde 2. sektörden itibaren 3 sektör 100 adresine oku.

M alan adr

Bir bellek bölgesini başka bir yere transferi.

ör: M 100 110 150

100'den 110'a kadar olan bölgenin 150 adresinden başlayan bölgeye transfer edilmesi.

N filename

Disk veya diskette işleme girecek dosyanın adının verilmesi.

ör: N STAR.COM

Dosyanın ad STAR.COM olarak belirlenir.

O portadr byte

Verilen çıkış I/O port adresinden verilen bilgiyi çıkar.

ör: O 3BC 2E

3BC portundan 2E bilgisini dışarı çıkar.

P=adr değer

Başlangıç adresinden itibaren verilen değer kadar satır koştur.

ör: P 100 4

100 adresinden itibaren 4 satır koştur.

Q

DEBUG programından DOS'a çıkış.

R

R F

R reg.adı

Herhangi bir registerin , registerlerin içeriğinin değiştirilmesini veya görüntülenmesini sağlar.

ör: R

Bütün reg. ve IP'nin bulunduğu yerdeki komut satır görüntülenir.

ör: R F

Flag reg.'daki bayrakların içerikleri görüntülenir ve değişiklik yapmaya olanak sağlar.

ör: R reg.adı

Ad verilen reg.'ın içeriği görüntülenir ve değişiklik yapmaya olanak sağlar.

S alan list.

Verilen alan içinde liste ile verilen dizinin bulunduğu yerleri arar.

ör: S 1B0C:100 110 'AB'

1B0C:100'den 110'a kadar olan alanda 'AB' dizisinin bulunup bulunmadığını eğer varsa adreslerini verir.

T=adr değer

Program belirtilen adresten itibaren verilen değer kadar adım adım çalıştırır.

ör: T=100 4

Program 100 adresinden itibaren 4 adım çalıştırır.

U adr**U alan**

Bellekteki kodlar assembly diline çevirir.

ör: U 100

100 adresinden itibaren yerleşmiş programı görüntüler.

ör: U 1B0C:100 120

1B0C:100 adresiyle 120 adresi arasında bulunan programı görüntüler.

W**W adr****W adr sürüc baş.sekt. sekt.sayısı**

Bellekteki program N komutu ile verilen adda diske yazmayı sağlar. Bu komut çalıştırılmadan önce, uzunluk bilgisi BX:CX reg. çiftine girilmelidir.

ör: W

CS:100 adresinden itibaren BX:CX uzunluğundaki, N komutuyla verilen addaki program diske yazılır.

XA sayı

Verilen say kadar EMS sayfasın kullanıma açar.

XD handle

XA komutuyla açılmış sayfalar kullanıma kapatır.

XM lpage ppage handle

EMS işlemi için EMS sayfaların düzenler. (Fiziksel, mantıksal map işlemi)

XS

EMS durumunu gösterir.

MOV / PUSH / POP / PUSHF / POPF

-R

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0C8A ES=0C8A SS=0C8A CS=0C8A IP=0100 NV UP EI PL NZ NA PO NC
0C8A:0100 B83412 **MOV AX,1234**

-T

AX=1234 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0C8A ES=0C8A SS=0C8A CS=0C8A IP=0103 NV UP EI PL NZ NA PO NC
0C8A:0103 50 **PUSH AX**

-D FFE0

0C8A:FFE0 00 00 00 00 34 12 00 00-03 01 8A 0C **E2 06** 00 00 ..4.....
0C8A:FFF0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

-T

AX=1234 BX=0000 CX=0000 DX=0000 SP=FFEC BP=0000 SI=0000 DI=0000
DS=0C8A ES=0C8A SS=0C8A CS=0C8A IP=0104 NV UP EI PL NZ NA PO NC
0C8A:0104 5B **POP BX**

-D FFE0

0C8A:FFE0 00 00 34 12 00 00 04 01-8A 0C E2 06 **34 12** 00 00 ..4.....4...
0C8A:FFF0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

-T

AX=1234 BX=1234 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0C8A ES=0C8A SS=0C8A CS=0C8A IP=0105 NV UP EI PL NZ NA PO NC
0C8A:0105 9C **PUSHF**

-D FFE0

0C8A:FFE0 00 00 34 12 34 12 00 00-05 01 8A 0C **E2 06** 00 00 ..4.4.....
0C8A:FFF0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

-T

AX=1234 BX=1234 CX=0000 DX=0000 SP=FFEC BP=0000 SI=0000 DI=0000
DS=0C8A ES=0C8A SS=0C8A CS=0C8A IP=0106 **NV UP EI PL NZ NA PO NC**
0C8A:0106 9D **POPF**

-D FFE0

0C8A:FFE0 00 00 34 12 00 00 06 01-8A 0C E2 06 **02 32** 00 00 ..4.....2..
0C8A:FFF0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

-RF

NV UP DI NG NZ NA PE NC -OV DN DI NG ZR AC PE CY

-R

AX=FFFF BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0CAC ES=0CAC SS=0CAC CS=0CAC IP=0105 **OV DN DI NG ZR AC PE CY**
0CAC:0105 8207DA **ADD BYTE PTR [BX],DA** DS:0000=CD

-

LAHF / SAHF

-R

AX=1005 BX=1000 CX=0000 DX=0000 SP=FFF4 BP=0000 SI=0005 DI=0000
DS=2F5B ES=0CC3 SS=0CC3 CS=0CC3 IP=010B NV UP DI PL NZ NA PO NC
0CC3:010B 9F LAHF

-T

AX=0205 BX=1000 CX=0000 DX=0000 SP=FFF4 BP=0000 SI=0005 DI=0000
DS=2F5B ES=0CC3 SS=0CC3 CS=0CC3 IP=010C NV UP DI PL NZ NA PO NC
0CC3:010C 9E SAHF

-RF

NV UP DI PL NZ NA PO NC -

-RAX

AX 0205

:FFFF

-T

AX=FFFF BX=1000 CX=0000 DX=0000 SP=FFF4 BP=0000 SI=0005 DI=0000
DS=2F5B ES=0CC3 SS=0CC3 CS=0CC3 IP=010D NV UP DI NG ZR AC PE CY

LDS / LEA

-D 0CC3:1000

0CC3:1000 5D 20 5B 2F 57 5D 20 5B-2F 41 5B 5B 3A 5D 94 7A] [/W] [/A[:].z
0CC3:1010 6E 69 74 65 6C 69 6B 6C-65 72 5D 5D 0D 0A 20 20 nitelikler]..

-R

AX=FFFF BX=1000 CX=0000 DX=0000 SP=FFF4 BP=0000 SI=0005 DI=0000
DS=0CC3 ES=0CC3 SS=0CC3 CS=0CC3 IP=0107 NV UP DI NG ZR AC PE CY
0CC3:0107 C500 LDS AX,[BX+SI] DS:1005=205D

-T

AX=205D BX=1000 CX=0000 DX=0000 SP=FFF4 BP=0000 SI=0005 DI=0000
DS=2F5B ES=0CC3 SS=0CC3 CS=0CC3 IP=0109 NV UP DI NG ZR AC PE CY
0CC3:0109 8D00 LEA AX,[BX+SI] DS:1005=0000

-T

AX=1005 BX=1000 CX=0000 DX=0000 SP=FFF4 BP=0000 SI=0005 DI=0000
DS=2F5B ES=0CC3 SS=0CC3 CS=0CC3 IP=010B NV UP DI NG ZR AC PE CY
0CC3:010B 9F LAHF

-

; Turbo Assembler Copyright (c) 1988, 1991 By Borland International, Inc.

; HELLO.ASM - Display the message "Hello World"

; From the Turbo Assembler Users Guide - Getting started

```
.MODEL small
.STACK 100h
.DATA
HelloMessage DB 'Hello, world',13,10,'$'
.CODE
mov ax,@data
mov ds,ax          ;set DS to point to the data segment
mov ah,9           ;DOS print string function
mov dx,OFFSET HelloMessage ;point to "Hello, world"
int 21h           ;display "Hello, world"
mov ah,4ch        ;DOS terminate program function
int 21h           ;terminate the program
END
```

Turbo Assembler Version 3.1 13/12/04 10:19:36 Page 1
HELLO1.asm

```
1 ; Turbo Assembler Copyright (c) 1988, 1991 By Borland
International, Inc.
2
3 ; HELLO.ASM - Display the message "Hello World"
4
5 ; From the Turbo Assembler Users Guide - Getting started
6
7 0000 .MODEL small
8 0000 .STACK 100h
9 0000 .DATA
10 0000 48 65 6C 6C 6F 2C 20+ HelloMessage DB 'Hello, world',13,10,'$'
11 77 6F 72 6C 64 0D 0A+
12 24
13 000F .CODE
14 0000 B8 0000s mov ax,@data
15 0003 8E D8 mov ds,ax ;set DS to point to the data segment
16 0005 B4 09 mov ah,9 ;DOS print string function
17 0007 BA 0000r mov dx,OFFSET HelloMessage ;point to "Hello, world"
18 000A CD 21 int 21h ;display "Hello, world"
19 000C B4 4C mov ah,4ch ;DOS terminate program function
20 000E CD 21 int 21h ;terminate the program
21 END
```

Turbo Assembler Version 3.1 13/12/04 10:19:36 Page 2
Symbol Table

Symbol Name	Type	Value
??DATE	Text	"13/12/04"
??FILENAME	Text	"HELLO1 "
??TIME	Text	"10:19:36"

```

??VERSION                Number 030A
@32BIT                    Text 0
@CODE                     Text _TEXT
@CODESIZE                 Text 0
@CPU                      Text 0101H
@CURSEG                   Text _TEXT
@DATA                     Text DGROUP
@DATASIZE                 Text 0
@FILENAME                 Text HELLO1
@INTERFACE                Text 00H
@MODEL                    Text 2
@STACK                    Text DGROUP
@WORDSIZE                 Text 2
HELLOMESSAGE              Byte DGROUP:0000

```

Groups & Segments

Bit Size Align Combine Class

Group	Bit	Size	Align	Combine	Class
DGROUP					
STACK	16	0100	Para		Stack STACK
_DATA	16	000F	Word		Public DATA
_TEXT	16	0010	Word		Public CODE

```

.MODEL small
.STACK 100h
.DATA
S0 DW 1
A0 DW 1234h,9,0,10Ch
A1 DW -1,-2,-3,-4
M0 DW 4 DUP (1)
.CODE
mov ax,@data
mov ds,ax
mov es,ax
; ORG 5FAH
MOV AL,0FEH
MOV BL,0FCH
IMUL BL
MOV AX,0080H
MOV DS,AX
MOV ES,AX
MOV SI,0010
MOV AX,0000
MOV DX,0000
MOV CX,0003
L0613H: ADD AX,[SI]
ADC DX,00
INC SI
INC SI
LOOP L0613H
DIV SI
MOV [M0+2],AX
MOV [M0+4],DX
END
RET
END

```