

ALT SEVİYE PROGRAMLAMA

Hafta 4

Dr. Öğr. Üyesi Erkan USLU

DİZİ KOMUTLARI

Dizi (string) Komutları

- MOVSB
- MOVSW
- CMPSB
- CMPSW
- SCASB
- SCASW
- LODSB
- LODSW
- STOSB
- STOSW
- CBW
- CWD
- REP
- REPE/REPZ
- REPNE/REPNZ

Dizi (string) Komutları

- Tekrarlamalı işlemler için tasarlanmış komutlardır
- İşlem tekrar sayısı CX yazmacında tutulur
- Her işlem adımında CX değeri bir otomatik olarak 1 azaltılır
- Tekrarlı dizi işlemlerini belirtmek için önekler kullanılır
(REP,REPE/REPZ,REPNE/REPZ)

Dizi (string) Komutları

- İşlem yönü DF bayrağı ile belirlenir
- DF=0 ise her adımında dizinin bir sonraki adresi için işlem tekrarlanır
- DF=1 ise her adımında dizinin bir önceki adresi için işlem tekrarlanır

Dizi (string) Komutları

- MOVS: bir bellek bölgesinin bir başka bellek bölgesine kopyalanması
- CMPS: farklı iki bellek bölgesinin içeriklerinin karşılaştırılması
- SCAS: bellek bölgesinin içeriğinin AX/AL ile karşılaştırılması
- LODS: bellek bölgesindeki değerin AX/AL ye yüklenmesi
- STOS: bellek bölgesinin AX/AL değeri ile doldurulması
- CBW: işaret bitini koruyarak byte'ı word'e genişletme
- CWD: işaret bitini koruyarak word'u double word'e genişletme

Dizi (string) Komutları

- MOVSB: move string byte
- $ES:[DI] \leftarrow DS:[SI]$
- $DI \leftarrow DI+1$; (DF=1 olsaydı $DI \leftarrow DI-1$)
- $SI \leftarrow SI+1$; (DF=1 olsaydı $SI \leftarrow SI-1$)

- MOVSW: move string word
- $ES:[DI] \leftarrow DS:[SI]$
- $DI \leftarrow DI+2$; (DF=1 olsaydı $DI \leftarrow DI-2$)
- $SI \leftarrow SI+2$; (DF=1 olsaydı $SI \leftarrow SI-2$)

Dizi (string) Komutları

- CMPSB: compare string byte
- DS:[SI]-ES:[DI]
- DI \leftarrow DI+1; (DF=1 olsaydı DI \leftarrow DI-1)
- SI \leftarrow SI+1; (DF=1 olsaydı SI \leftarrow SI-1)

- CMPSW: compare string word
- DS:[SI]-ES:[DI]
- DI \leftarrow DI+2; (DF=1 olsaydı DI \leftarrow DI-2)
- SI \leftarrow SI+2; (DF=1 olsaydı SI \leftarrow SI-2)

Dizi (string) Komutları

- SCASB: scan string byte
- AL-ES:[DI]
- $DI \leftarrow DI+1$; (DF=1 olsaydı $DI \leftarrow DI-1$)

- SCASW: scan string word
- AX-ES:[DI]
- $DI \leftarrow DI+2$; (DF=1 olsaydı $DI \leftarrow DI-2$)

Dizi (string) Komutları

- LODSB: load string byte
- $AL \leftarrow DS:[SI]$
- $SI \leftarrow SI+1$; (DF=1 olsaydı $SI \leftarrow SI-1$)

- LODSW: load string word
- $AX \leftarrow DS:[SI]$
- $SI \leftarrow SI+2$; (DF=1 olsaydı $SI \leftarrow SI-2$)

Dizi (string) Komutları

- STOSB: store string byte
- $ES:[DI] \leftarrow AL$
- $DI \leftarrow DI+1$; (DF=1 olsaydı $DI \leftarrow DI-1$)

- STOSW: store string word
- $ES:[DI] \leftarrow AX$
- $DI \leftarrow DI+2$; (DF=1 olsaydı $DI \leftarrow DI-2$)

Dizi (string) Komutları

- CBW: convert byte to word
- $AX \leftarrow AL$

- CWD: convert word to double word
- $DX:AX \leftarrow AX$

Dizi (string) Komutları

- LEA DI, dizi
- CLD
- XOR AL, AL
- MOV CX, size
- REP STOSB

- dizi \leftarrow 0

Dizi (string) Komutları

- LEA SI, dizi1
- LEA DI, dizi2
- CLD
- MOV CX, size
- REP MOVSB

- dizi2 ← dizi1

Dizi (string) Komutları - Önekler

- REP: repeat
- CX=0 olana kadar tekrarla

- REPE/REPZ: repeat equal
- CX=0 veya ZF=0 olana kadar tekrarla

- REPNE/REPNZ: repeat not equal
- CX=0 veya ZF=1 olana kadar tekrarla

GİRİŞ ÇIKIŞ KOMUTLARI

Giriş Çıkış Komutları

- IN
- OUT

Giriş Çıkış Komutları

- IN: input from port adress
- IN acc, idata; 0-255 arası portlara erişim
- IN acc, DX; 0-65535 arası portlara erişim

Giriş Çıkış Komutları

- OUT: output to port address
- OUT idata, acc; 0-255 arası portlara erişim
- OUT DX, acc; 0-65535 arası portlara erişim

DURDURMA ve BEKLETME KOMUTLARI

Durdurma ve Bekletme Komutları

- HLT: halt, işlemci durur, kesme ile işlemci kaldığı komuttan devam eder
- NOP: no operation
- WAIT: wait until not busy, yardımcı işlemcinin sonucunu beklemek için kullanılır

BCD DÜZENLEME KOMUTLARI

BCD Düzenleme Komutları

- AAA
- AAD
- AAM
- AAS
- DAA
- DAS

BCD Düzenleme Komutları

- AAA: ASCII adjust after addition
- ASCII değer olarak tutulan iki sayının toplamını BCD olarak düzenler

- XOR AX, AX
- MOV AL, "6"
- ADD AL, "7"
- AAA ; AX ← 0103H , BCD olarak 13 sayısı

BCD Düzenleme Komutları

- AAD: ASCII adjust before division
- BCD olarak tutulan iki basamaklı sayının bölme öncesi düzenlenmesi için kullanılır

- MOV AH, 04H
- MOV AL, 05H; AX'te BCD 45 sayısı var
- AAD; AX \leftarrow 002DH

BCD Düzenleme Komutları

- AAM: ASCII adjust after multiplication
- ASCII değer olarak tutulan iki sayının toplamını BCD olarak düzenler

- MOV AL, 4
- MOV AH, 8
- MUL AH; AX'te 0020H var
- AAM ; AX \leftarrow 0302H , BCD olarak 32 sayısı

BCD Düzenleme Komutları

- AAS: ASCII adjust after subtraction
- ASCII değer olarak tutulan iki sayının farkını BCD olarak düzenler

- MOV AL, "5"
- SUB AL, "7"; AL'de FEH var
- AAS; AL \leftarrow 08H, CF=1; $-10+8=-2$

BCD Düzenleme Komutları

- DAA: decimal adjust AL after addition
- Sıkıştırılmış BCD formatlı sayıların toplanması sonucu AL'deki değeri sıkıştırılmış BCD olarak düzenler

- MOV AL, 75H
- ADD AL, 19H; AL'de 8EH var
- DAA; AL ← 94H

BCD Düzenleme Komutları

- DAS: decimal adjust AL after subtraction
- Sıkıştırılmış BCD formatlı sayıların çıkarılması sonucu AL'deki değeri sıkıştırılmış BCD olarak düzenler

- MOV AL, 42H
- SUB AL, 13H; AL'de 2FH var
- DAS; AL ← 29H