

Bilgisayar Organizasyonu ve Mimarisi - Uygulama

Hafta 2

Veri Tipleri

Komutlar `-b` veya `-sb` eklentisi alır

Byte



Komutlar `-h` veya `-sh` eklentisi alır

Half Word

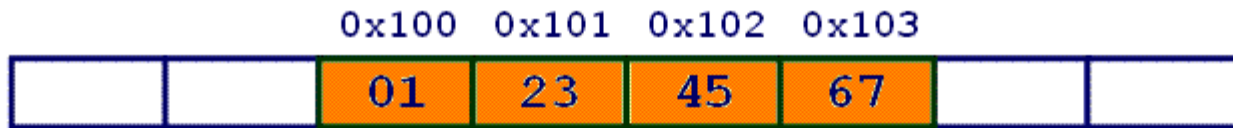


Word



Endiannes

- Veriler saklanırken en önemli bitin nerede tutulacağını söyler.
- Değer: 0x01234567



Big Endian



Little Endian

Endiannes

- Intel x86 mimarisindeki işlemciler Little-Endian
- Arm Mimarisinde ise v3 ten önce Little-Endian
- Daha Sonra bi-endian
- Kontrol CPSR içerisindeki E biti ile yapılır.
- JVM Big Endian
- Adobe Photoshop,JPEG Big-Endian
- GIF,BMP Little-Endian

Endiannes

Big-Endian çalışan JVM'de little kullanmak için

```
private static final int BUFFER_SIZE = 1024;

ByteBuffer readBuffer = ByteBuffer.allocate(BUFFER_SIZE).order(ByteOrder.BIG_ENDIAN);
int bytesRead = socketChannel.read(readBuffer);
readBuffer.flip();
```

ARM Kaydedicileri

- Kullanıcıların erişebildiği kaydediciler 16 Adet R0-R15
- R0-R12 genel amaçlı kullanım için.
- R7 – System Call kullanımında numarasını tutmak için kullanılabilir.

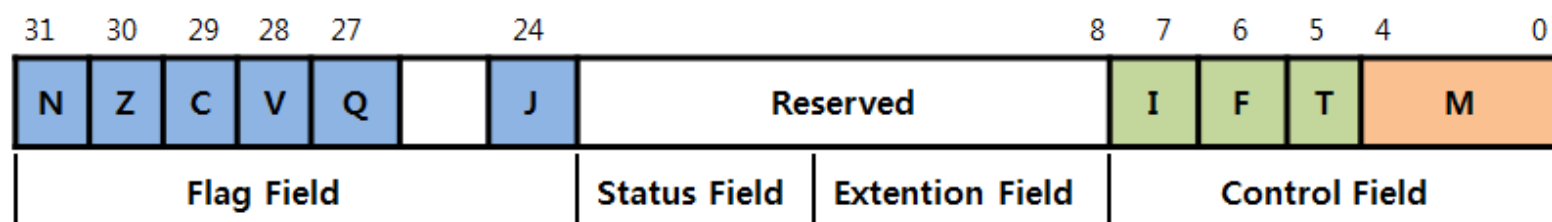
ARM Kaydedicileri

- **R13:SP** (Stack Pointer): Stack bölgesinin tepesinin adresi tutulur. İşlem yapılırken stack bölgesi kullanılacaksa ilerletilerek verilere erişim sağlanır.
- **R14: LR** (Link Register): Alt Program çağrıldığında LR çağrılan yerden bir sonraki komutun adresini tutar.

ARM Kaydedicileri

- **R15: PC** (Program Counter): Komutlar çağrıldıkça değeri artar. Bir sonraki komutun adresini tutar.
 - (<http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0473j/dom1359731138020.html>)
- **CPRS:** (Current Program Status Register):
- Çalıştırılan komutlar sonrası gerçekleşen durumların tutulduğu kaydedicidir.

CPRS



Flag Field	
N	Negative result from ALU
Z	Zero result from ALU
C	ALU operation caused Carry
V	ALU operation oVerflowed
Q	ALU operation saturated
J	Java Byte Code Execution

Control bits	
I	1: disables IRQ
F	1: disables FIQ
T	1: Thumb, 0: ARM

Mode bits M[4:0]	
0b10000	User
0b11111	System
0b10001	FIQ
0b10010	IRQ
0b10011	SVC(Supervisor)
0b10111	Abort
0b11011	Undefined

ARM Komutlarına Giriş

- Arm komutlarını aşağıdaki şablon şeklinde gösterebiliriz.
- **MNEMONIC{S}{condition} {Rd}, Operand1, Operand2**
- MNEMONIC - Komutun kısa adı
- {S} -Eğer kullanılırsa bayraklar etkilenir.
- {condition} - Şart gerçekleşirse işlem yapılır.
- {Rd} - Sonucun tutulacağı kaydedici
- Operand1 - Kaydedici veya immediate değer olabilir.
- Operand2 -İmmediate değer veya kaydedici veya shift edilmiş olabilir.

ARM Komutlarına Giriş

- **#123** - Immediate değer.
- **Rx** - Kaydedici x (R1, R2, R3 ...)
- **Rx, ASR n** - Sağa n bit kaydırılmış kaydedici ($1 = n = 32$)
- **Rx, LSL n** - Sola n bit kaydırılmış kaydedici ($0 = n = 31$)
- **Rx, LSR n** - Sağa n bit kaydırılmış kaydedici ($1 = n = 32$)
- **Rx, ROR n** - Sağa n bit rotate ($1 = n = 31$)

ARM Komutlarına Giriş

- **ADD R0, R1, R2** - R1 ve R2 toplanıp sonuç R0.
- **ADD R0, R1, #2** - R1'e 2 eklenir sonuç R0.
- **MOVLE R0, #5** -R0'a 5 atılır. (LE – Küçük veya eşit olduğu sürece)
- **MOV R0, R1, LSL #1** - R1 bir defa sola kaydırılır ardından içindeki değer R0'a atılır.

Mov Komutu

- MOV R0,0x12
- MOV R1,#1955
- MVN R2,#4
- MOVW R3,0xABCD
- MOVT R3,0xEF12
- MOV R1,R3

Aritmetik İşlemler

- ADD R0,R1,R1 ;R0=R1+R1
- ADD R2,R1,#2 ;R2=R1+2
- ADD R2,R0 ;R2=R2+R0

- ADC R1,R2,R3 ; R1=R2+R3+C

- SUB R1,R2,R3 ; R1=R2-R3
- SUBC R1,R2,R3 ; R1=R2-R3+C-1
- RSB R1,R2,R3 ; R1=R3-R2

MUL

- Immediate kullanılamaz
- MUL R4,R2,R1 ;R4=R2*R1
- UMULL R6,R8,R0,R1 ;{R6,R8}=R0*R1

DIV

- Integer Bölme
- `UDIV R3,R2,R0`
- `SDIV R6,R5,R4`