

Bilgisayar Organizasyonu ve Mimarisi

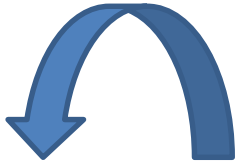
Hafta 3

Hafıza İşlemleri

- Arm Load-Store model: Sadece hafıza komutları erişebilir.
- Hafızadaki veri ile işlem yapabilmek için öncelikle kaydedicilere aktarım yapmak gerekiyor.

Hafıza Komutları

- Temel olarak iki komut vardır.



- **LDR R1,[R2]** :Hafızada R2 adresindeki değer R1 kaydedicisine aktarılır.



- **STR R1,[R2]** : R1 Kaydedicisindeki değer hafızadaki R2 adresindeki yere aktarılır.

```
ldr    r0, addr_var1
```

```
ldr    r1, addr_var2
```

```
ldr    r2, [r0]
```

```
str    r2, [r1]
```

Registers

R0	R1	R2
0x00000000	0x00000000	0x00000000

Memory

	...	
0x00010098		
0x00010094	0x04	<var2>
0x00010090	0x03	<var1>
	...	

Offset Modları

- **Immediate** value offset: Sayısal bir değer kullanılarak ulaşılır.
- **Register** offset: Kaydedicideki değer ile ulaşılır.
- **Scaled register** offset: Kaydedici kaydırılarak erişilir.

Adresleme Modları

- Offset Adresleme: ofsetlenen adresteki değer işleme alınır. Base register değişmez.
- Pre-Indexed : Base register değişir, ardından bu adresteki değer üzerinden işlem yapılır.
- Post-Indexed: Önce ofsetlenen adresteki değer işlenir ardından base register değeri değişir.

- `ldr r2, [r1, #4] ;`
- `ldr r2, [r1, #4]! ;` pre-indexed $r1=r1+4$
- `ldr r3, [r1], #4 ;` post-indexed $r1=r1+4$

Registers

R0	R1	R2	R3
0x00010098	0x0001009c	0x00000003	0x00000000

```
ldr r0, adr_var1
ldr r1, adr_var2
ldr r2, [r0]
str r2, [r1, r2]
str r2, [r1, r2]!
ldr r3, [r1], r2
```

Memory

	...	
0x0001009F		
0x0001009C	0x00000004	<var2>
0x00010098	0x00000003	<var1>
	...	

Scaled Register

- LDR R1, [R2, R3, <shifter>]
- STR R1, [R2, R3, <shifter>]
- LDR r2, [r1, r3, LSL#2]

LDR PC-RELATIVE ADDRESS

- Arm assembly 8 bit immediate value
- ISA sebebiyle
- LDR R3,=123456
- Pseudo-Instruction
- Literal pool referans eder. (sabit,string,offset literal pool içerisinde)

Data Allocation

Directive	Description	Memory Space
DCB	Define Constant Byte	Reserve 8-bit values
DCW	Define Constant Half-word	Reserve 16-bit values
DCD	Define Constant Word	Reserve 32-bit values
DCQ	Define Constant	Reserve 64-bit values
DCFS	Define single-precision floating-point numbers	Reserve 32-bit values
DCFD	Define double-precision floating-point numbers	Reserve 64-bit values
SPACE	Defined Zeroed Bytes	Reserve a number of zeroed bytes
FILL	Defined Initialized Bytes	Reserve and fill each byte with a value

Data Allocation

AREA	myData, DATA, READWRITE		
hello	DCB	"Hello World!",0	; Allocate a string that is null-terminated
dollar	DCB	2,10,0,200	; Allocate integers ranging from -128 to 255
scores	DCD	2,3.5,-0.8,4.0	; Allocate 4 words containing decimal values
miles	DCW	100,200,50,0	; Allocate integers between -32768 and 65535
Pi	DCFS	3.14	; Allocate a single-precision floating number
Pi	DCFD	3.14	; Allocate a double-precision floating number
p	SPACE	255	; Allocate 255 bytes of zeroed memory space
f	FILL	20,0xFF,1	; Allocate 20 bytes and set each byte to 0xFF
binary	DCB	2_01010101	; Allocate a byte in binary
octal	DCB	8_73	; Allocate a byte in octal
char	DCB	'A'	; Allocate a byte initialized to ASCII of 'A'

Örnek

- f,g,h sırasıyla R0,R1,R2
- Dizilerin başlangıç adresi R6,R7
- $f = g + h + B[4]$
- $f = g - A[B[4]]$

Soru

Address	Data
12	1
8	6
4	4
0	2

Yukarıdaki hafıza yerlerine bakarak, en büyük veriyi en büyük adrese, küçük verileri küçük adrese denk gelecek şekilde yerleştiren ARM assembly kodlarını yazınız. Dizinin başlangıç adresi R6 kaydedicisindedir.