CSE 30: Computer Organization and Systems Programming

Lecture 12: ARM Assembly Data Transfer Instructions

Diba Mirza
University of California, San Diego
(Recap) Arithmetic Instructions
Assignment Instructions

In C:

a = b;

In ARM:

a = b;

a = 10;
Data transfer (memory) Instructions

• Separate instructions to transfer data between registers and memory:
  – Memory to register (load)
  – Register to memory (store)

• Load/store basic syntax
Base Register Addressing Mode

- The memory location to be accessed is held in a base register
  - STR r0, [r1] @Store contents of r0 to location pointed to by contents of r1.
  - LDR r2, [r1] @Load r2 with contents of memory location pointed to by contents of r1.
Data Transfer: Memory to Register

LDR r2, [r1]

The contents of which of the following changes when the above instruction is executed?

A. r1
B. r1 and memory
C. r2
D. r2 and memory
E. r1, r2 and memory
Data Transfer: Memory to Register

```
STR r2, [r1]
```

The contents of which of the following changes when the above instruction is executed?

A. R1  
B. memory  
C. R2  
D. r1, r2 and memory

![Memory and Register Diagram](image)
Variations on load/store
Data Transfer Instructions

In C:

```c
void foo (int *p){
    *p=10;
}
```

Which of the following instructions is needed to translate the above code to ARM?

A. LDR and MOV
B. STR and MOV
C. Only LDR
D. Only STR
Data Transfer Instructions

In C:

```c
void foo (int *p){
    int a=*p;
}
```

In ARM:
Immediate Offset Addressing Mode

• To specify a memory address to copy from, specify:
  – A register which contains a pointer to memory
  – A numerical offset (in bytes)
• The desired memory address is the sum of these two values.
• Example: \([r0, #8]\)
  – specifies the memory address pointed to by the value in \(r0\), plus 8 bytes

\[
\text{LDR } r1, [r0, #8] \\
\text{STR } r1, [r0, #8]
\]
Compile by hand

• $g = h + A[8]$
Data Transfer: Memory to Register

LDR r2,[r1, #-4]

What are the contents of r2 after the above instruction is executed? (Assume Little Endian byte ordering)

A. 0x200
B. 0xaa
C. 0xaabbccddd
D. 0xddcbbbaa
void foo (int *p, int size) {
    *p = 0;
    *(p+1) = 0;
    *(p+size-1) = 0;
}

In ARM:
Data Transfer Instructions

```c
void swap (int *x, *y) {
    int tmp = *x;
    *x = *y;
    *y = tmp;
}
```

In ARM: