

Lab 3: Introduction to ARM assembly language

Goals

1. Understanding of the fundamental concepts of Assembly Language
2. Able to write Simple Branching Assembly program using ARM v7 ISA
3. Develop familiarity with a Computer System Simulator: Web based Cputator

Theory:

Introduction:

Branching instructions in ARM assembly language play a crucial role in controlling the flow of execution based on conditions. These instructions use the current state of the flags in the **Current Program Status Register (CPSR)** (such as the Zero, Negative, and Carry flags) to decide whether to branch (jump) to another part of the program. Understanding them allows for creating efficient and dynamic control structures like conditional statements and loops.

Let's explore the key branching instructions: **BEQ**, **BNE**, **BGT**, and more.

1. BEQ (Branch if Equal)

Description:

- The BEQ instruction branches to a specific label if the **Zero flag (Z)** is set, which typically happens when two values are equal after a comparison.
- **Common Use:** It is used when you want to branch to a section of code if two values are equal.

Assembly program example:

```
.global _start  
_start:
```

```
    CMP R0, R1    @Compare R0 and R1  
    BEQ equal     @Branch to 'equal' if R0 equals R1
```

equal:

```
    MOV R3,#5     @it moves 5 to R3 if it's Equal  
    .end
```

2. BNE (Branch if Not Equal)

Description:

- The BNE instruction branches to a specific label if the **Zero flag (Z)** is **clear**, meaning that the two compared values are not equal.
- **Common Use:** It is used when you want to take action when two values are different.

Assembly program example:

```
.global _start
_start:
    MOV R0,#2
    MOV R1, #4
    CMP R0, R1    @Compare R0 and R1
    BNE n_equal   @Branch to 'equal' if R0 equals R1

n_equal:

    MOV R3,#5     @it moves 5 to R3 if it's Equal
.end
```

3. BGT (Branch if Greater Than)

Description:

- The BGT instruction branches to a specified label if the first operand is **greater than** the second after a comparison. This happens when both the **Negative (N)** and **Zero (Z)** flags are clear.
- **Common Use:** It is used to branch when one value is greater than the other.

Assembly program example:

```
CMP R0, R1    @Compare R0 and R1
BGT greater   @Branch to 'greater' if R0 is greater than R1

greater:
    @Code for greater condition
```

4. Other Conditional Branch Instructions:

In addition to BEQ, BNE, and BGT, ARM supports a wide range of conditional branch instructions that allow for different comparisons:

- **BLT (Branch if Less Than):** Branches if the first value is less than the second.
- **BGE (Branch if Greater or Equal):** Branches if the first value is greater than or equal to the second.
- **BLE (Branch if Less or Equal):** Branches if the first value is less than or equal to the second.
- **BMI (Branch if Minus):** Branches if the result is negative.

Mimicking If-else statement in ALP:

The equivalent of an if-else statement in assembly language is typically implemented using **conditional branch instructions**.

```
if (R0 > R1) {  
  R2 = R0;  
}  
else {  
  R2 = R1;  
}
```

Equivalent ARM :

```
MOV R0, #10    ;@Assume R0 = 10 (first number)  
MOV R1, #5     @ Assume R1 = 5 (second number)
```

```
CMP R0, R1     @Compare R0 and R1  
BGT greater    @Branch if R0 > R1 (greater)  
MOV R2, R1     @Else: move R1 to R2 (R2 = R1)  
B done         @Jump to done
```

```
greater:  
  MOV R2, R0    @If R0 > R1, move R0 to R2 (R2 = R0)
```

```
done:  
  .end
```

CSE331L/EEE453L: Microprocessor Interfacing and Embedded System
LabNorth South University, Dhaka

Lab Report

Lab 3: Introduction to ARM assembly language

Student Names: _____

Student IDs: _____

Section: _____

Answer the following questions:

1. Write about major categories of ARM instructions.
2. What are major Branch instructions in ARM? Explain all.
3. What does this ADDS mean?
4. How CMP and BCond (i.e. BEQ, BNE) work together?

