

CONDITIONAL EXECUTION IN ARM

Conditional execution is a powerful feature in ARM architecture that allows instructions to be executed based on the state of the condition flags. This can help optimize code by reducing the need for branch instructions.

Condition Codes

ARM uses four condition flags to control conditional execution:

- **N (Negative):** Set when the result of an operation is negative.
- **Z (Zero):** Set when the result of an operation is zero.
- **C (Carry):** Set when an operation results in a carry out.
- **V (Overflow):** Set when an operation causes an overflow.

Conditional Suffixes

Instructions can be made conditional by appending a two-letter suffix to the mnemonic. Here are some of the common condition suffixes:

- **EQ: Equal** ($Z == 1$)
- **NE: Not equal** ($Z == 0$)
- **CS/HS: Carry set/Unsigned higher or same** ($C == 1$)
- **CC/LO: Carry clear/Unsigned lower** ($C == 0$)
- **MI: Negative** ($N == 1$)
- **PL: Positive or zero** ($N == 0$)

- **VS: Overflow** ($V == 1$)
- **VC: No overflow** ($V == 0$)
- **HI: Unsigned higher** ($C == 1$ and $Z == 0$)
- **LS: Unsigned lower or same** ($C == 0$ or $Z == 1$)
- **GE: Signed greater than or equal** ($N == V$)
- **LT: Signed less than** ($N != V$)
- **GT: Signed greater than** ($Z == 0$ and $N == V$)
- **LE: Signed less than or equal** ($Z == 1$ or $N != V$)
- **AL: Always** (default, always executed)

Example of Conditional Execution

Here's an example that demonstrates conditional execution:

MOV R0, #5 ; Load 5 into R0

MOV R1, #10 ; Load 10 into R1

CMP R0, R1 ; Compare R0 and R1 (sets condition flags)

MOVEQ R2, #1 ; If $R0 == R1$, load 1 into R2 (equal)

MOVNE R2, #0 ; If $R0 != R1$, load 0 into R2 (not equal)

In this example:

- **CMP R0, R1** sets the condition flags based on the comparison of **R0** and **R1**.
- **MOVEQ R2, #1** executes if **R0** is equal to **R1** (i.e., the Zero flag is set).
- **MOVNE R2, #0** executes if **R0** is not equal to **R1** (i.e., the Zero flag is clear).

Using Conditional Execution for Branching

ARM assembly also supports conditional branches, which allow you to change the flow of control based on the condition flags.

CMP R0, R1 ; Compare R0 and R1

BEQ equal_label ; Branch to equal_label if R0 == R1

BNE not_equal_label ; Branch to not_equal_label if R0 != R1

equal_label:

; Code to execute if R0 == R1

B end_label ; Branch to end_label

not_equal_label:

; Code to execute if R0 != R1

end_label:

; Continue with the rest of the program

In this example:

- **BEQ** branches to **equal_label** if **R0** equals **R1**.
- **BNE** branches to **not_equal_label** if **R0** does not equal **R1**.

Using conditional execution efficiently can help minimize the number of branch instructions, which can improve performance and reduce code size.