

# Arm assembler: conditional branching

# Arm assembler: conditional branching

## Decisions in C

*if (condition)....*

# Arm assembler: conditional branching

## Decisions in C

*if (condition)....*

*while (condition)....*

# Arm assembler: conditional branching

## Decisions in C

*if (condition)....*

*while (condition)....*

*switch (variable){  
case (equal some value):....*

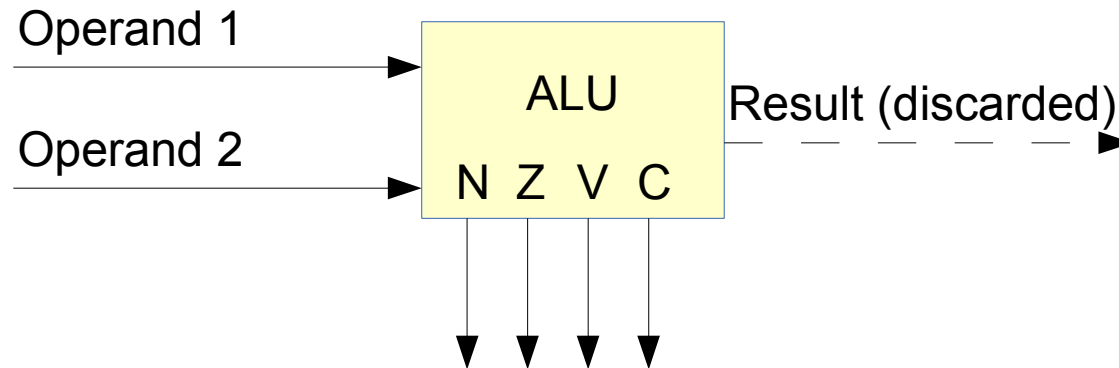
# Arm assembler: conditional branching

These decisions are made by ***comparing***

**Compare** involves subtraction

# Arm assembler: conditional branching

**Compare** involves subtraction



## Arithmetic flags

**N**egative

**Z**ero

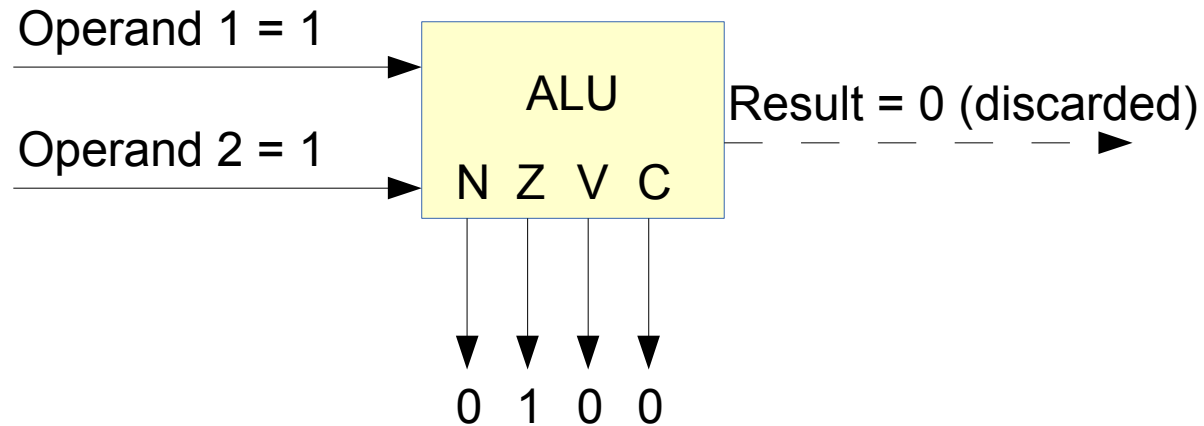
signed **oV**erflow

**C**arry (unsigned overflow)

## Arm assembler: conditional branching

```
A=1;  
:  
B=1;  
:  
if (A==B)  
{  
}
```

# Arm assembler: conditional branching

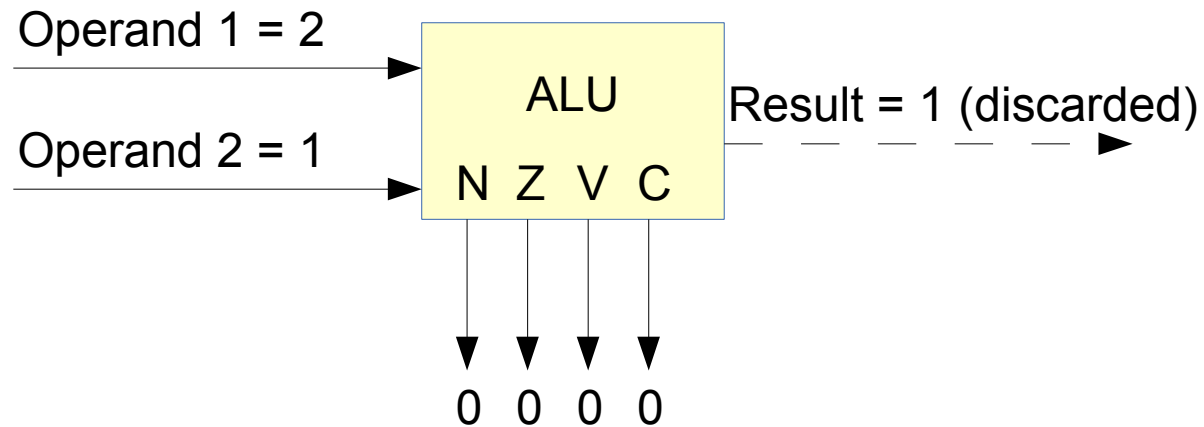




## Arm assembler: conditional branching

```
A=2;  
:  
B=1;  
:  
if (A==B)  
{  
}
```

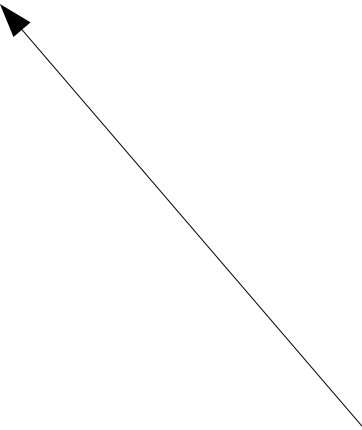
# Arm assembler: conditional branching



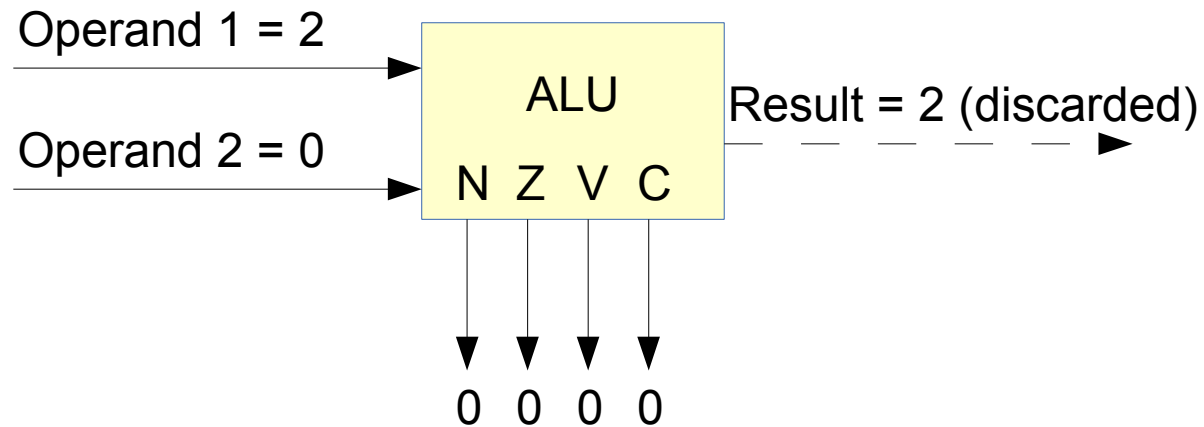
# Arm assembler: conditional branching

```
A=2;  
while(A)  
{  
}
```

Implied comparison  
with ZERO



# Arm assembler: conditional branching



# Arm assembler: conditional branching

Assembler implementation of decision making: Part 1

The compare instruction:

28.5.5.4.1 Syntax

CMN Rn, Rm

**CMP Rn, #imm**

**CMP Rn, Rm**

where:

Rn is the register holding the first operand.

Rm is the register to compare with.

imm is the immediate value to compare with.

28.5.5.4.2 Operation

These instructions compare the value in a register with either the value in another register or an immediate value. **They update the condition flags on the result, but do not write the result to a register.**

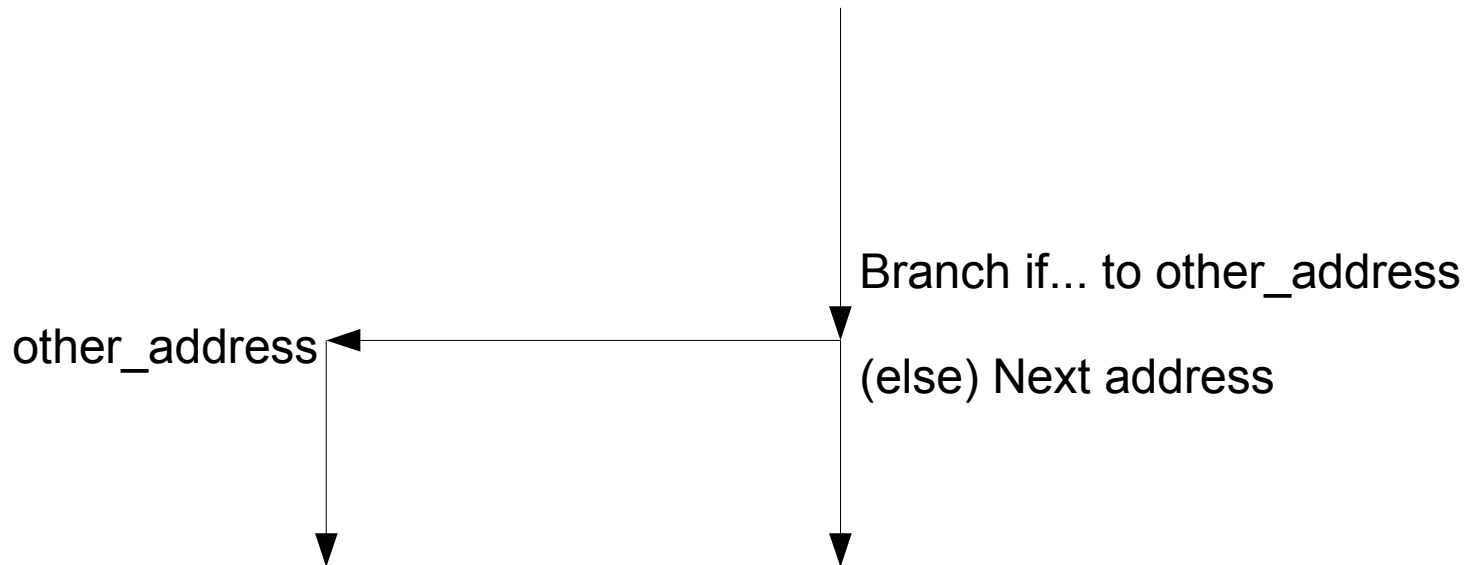
The CMP instruction subtracts either the value in the register specified by Rm, or the immediate imm from the value in Rn and updates the flags. This is the same as a SUBS instruction, except that the result is discarded.

The CMN instruction adds the value of Rm to the value in Rn and updates the flags. This is the same as an ADDS instruction, except that the result is discarded.

# Arm assembler: conditional branching

## Assembler implementation of decision making: Part 2 Conditional Branches:

28.5.6.1.1 Syntax  
**B**{cond} label



# Arm assembler: conditional branching

**Table 432. Condition code suffixes**

Suffix	Flags	Meaning
EQ	Z = 1	Equal, last flag setting result was zero
NE	Z = 0	Not equal, last flag setting result was non-zero
CS or HS	C = 1	Higher or same, unsigned
CC or LO	C = 0	Lower, unsigned
MI	N = 1	Negative
PL	N = 0	Positive or zero
VS	V = 1	Overflow
VC	V = 0	No overflow
HI	C = 1 and Z = 0	Higher, unsigned
LS	C = 0 or Z = 1	Lower or same, unsigned
GE	N = V	Greater than or equal, signed
LT	N != V	Less than, signed
GT	Z = 0 and N = V	Greater than, signed
LE	Z = 1 and N != V	Less than or equal, signed
AL	Can have any value	Always. This is the default when no suffix is specified.

## Arm assembler: conditional branching

- Examples (A,B unsigned)
  - `if (A==B) ...`
  - `if (A != B) ...`
  - `if (A > B) ...`
  - `if (A < B) ...`
  - `if (A >=B) ...`
  - `if (A <= B) ...`



# Arm assembler: conditional branching

– `if (A==B) ...`

## C Code

```
if (A == B)
{
    Equal:
}
else
{
    NotEqual:
}
Done:
```

## ARM Assembler

*Assume A is in R0 and B is in R1*

	CMP R0,R1
	BZ Equal
NotEqual	(do some stuff)
	B Done
Equal	(do some other stuff)
	B Done

*Finish remaining examples.*

## Arm assembler: conditional branching

- Examples (A,B signed)
  - if (A==B) ...
  - if (A != B) ...
  - if (A > B) ...
  - if (A < B) ...
  - if (A >=B) ...
  - if (A <= B) ...