

Flags, Conditionals, Branches



Flags

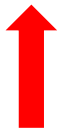
- Flags are bits which can be set depending on the output of an instruction.
- Flags are held in a special register called the Current Program Status Register (CPSR).
- There are four flags:
 - Negative (N): result had a 1 in the most significant bit.
 - Zero (Z): result was 0.
 - Overflow (V): the result overflowed.
 - Carry (C): operation carried out.



Negative Flag

- ARM calls this the 'S' flag, for “sign”.
- If the most significant bit is set, then $N = 1$.
- Otherwise, $N = 0$.
- All other bits are irrelevant.
- Example:

10000000 00000000 00000000 00000000



MSB



Zero Flag

- If all bits in the register are zero, $Z = 1$.
- Otherwise, $Z = 0$.
- Example:

00000000 00000000 00000000 00000000



Overflow Flag

- If the MSB's of both operands are equal, and the output value's MSB is different, $V = 1$.
- Meaning: if both operand's MSB equals 1, and the output's MSB is 0 (or vice versa), the overflow flag is set.
- This is important in signed arithmetic as it indicates:
 - Two negative numbers produced a positive number; or,
 - Two positive numbers produced a negative number.



Overflow with 8-bit Registers

```
int8_t a = 65; // 01000001b
```

```
int8_t b = 64; // 01000000b
```

```
int8_t result = a + b;
```

$$\begin{array}{r} 01000001_b \\ +01000000_b \\ \hline 10000001_b \end{array}$$

Expected result: 129_d

Actual result: -127_d



Overflow Flag (cont)

- Negative integers may also overflow (we will discuss this when we get to processor arithmetic).
- What happens to the overflow flag if we're operating on unsigned integers?
 - V will still be set.
 - The processor doesn't know those bits represent an unsigned integer.



Carry Flag

- The carry flag indicates the result of an operation was too large (or too small) to fit in the number of available bits.
- Meaning: the result “carried out” from (or “borrowed in” to) the most significant bit.



Carry out with 8-bit Registers

```
uint8_t a = 129; // 10000001b  
uint8_t b = 127; // 01111111b  
uint8_t result = a + b;
```

$$\begin{array}{r} 10000001_{\text{b}} \\ +01111111_{\text{b}} \\ \hline 100000000_{\text{b}} \end{array}$$

Expected result: 256_{d}

Actual result: 0_{d}



Setting the Flags

- We can set the flags by putting the 'S' operand at the end of any command. Examples:
 - ADDS
 - SUBS
 - MOVS
 - ...Et cetera.
- What happens if we add 'S' to something other than an arithmetic command?
 - The flags are still set.
 - The processor doesn't know the output isn't an integer.



CMP and TST

- Two commands, compare (CMP) and test (TST), will automatically set the flags without the 'S' operand.
- Compare is a subtract operation that doesn't save the output.
- Test is a logical AND that doesn't save the output.



Conditionals

- Flags are checked when you use a conditional suffix.
- A conditional suffix can be added to any command.
- Example conditionals:
 - **EQ** (equal)
 - **NE** (not equal)
 - **LT** (less than, signed), **GE** (greater or equal, signed)
 - **CC** (less than, unsigned), **CS** (greater or equal, unsigned)
 - **MI** (negative)
 - **PL** (positive)
 - Many others....



Branches

- A branch statement moves the code execution to a label.
- These are the “goto” statements you’ve always been told not to use.
- An unconditional branch will always jump to a label.
- A branch used with a conditional suffix will only jump if the condition is met.



Branch Commands

- B
 - Unconditional branch
 - Always move to label
- BL
 - Branch-and-link
 - Moves to label, saves return address in link register
- Bsuffix
 - Conditional branch
 - Will branch if suffix condition is met

