



ARM Status Flags Cheat Sheet (Cortex-M3 / Thumb)

Flag পূর্ণ নাম কখন 1 হয় কখন 0 হয় উদাহরণ

| **N** | **Negative Flag** | রেজাল্টের MSB (Most Significant Bit) = 1 → মানে ফলাফল Negative | রেজাল্টের MSB = 0 → মানে Positive | `MOVS R0, #-5` → R0 = 0xFFFFFFF5 → N=1 |

| **Z** | **Zero Flag** | অপারেশনের ফলাফল = 0 | ফলাফল ≠ 0 | `SUBS R1, R1, #1` → যখন R1=1 থেকে 0 হয়, Z=1 |

| **C** | **Carry Flag** | Unsigned addition এ Carry হলে, বা subtraction এ Borrow না হলে | Unsigned addition এ Carry না হলে, বা subtraction এ Borrow হলে | `ADDS R0, R1, #1` → যদি Carry হয়, C=1 |

| **V** | **Overflow Flag** | Signed addition/subtraction এ overflow হলে (মানে ফলাফল sign change হয়ে গেছে) | Overflow না হলে | `ADDS R0, R1, #1` → যদি দুইটা positive যোগে negative হয়, V=1 |

◆ এখন দেখি — কোন ইনস্ট্রাকশন কোন ফ্ল্যাগে প্রভাব ফেলে

Instruction	N	Z	C	V	মন্তব্য
<code>ADDS</code>	✓	✓	✓	✓	সব ফ্ল্যাগ সেট হয়
<code>SUBS</code>	✓	✓	✓	✓	সব ফ্ল্যাগ সেট হয়
<code>RSBS</code>	✓	✓	✓	✓	Reverse subtract (0 - Rn)
<code>ANDS, ORRS, EORS</code>	✓	✓	✗	✗	লজিকাল অপারেশন — Carry/Overflow হয় না
<code>MOVS, MVNS</code>	✓	✓	✗	✗	কেবল N এবং Z
<code>CMP</code>	✓	✓	✓	✓	Compare = Subtract কিন্তু ফলাফল সংরক্ষণ হয় না
<code>CMN</code>	✓	✓	✓	✓	Compare Negative = Add কিন্তু ফলাফল সংরক্ষণ হয় না

TST	✓	✓	✗	✗	AND test only (no carry/overflow)
LSLs, LSRs	✓	✓	✓	✗	Shift এর সময় carry last bit এ যায়
RORS	✓	✓	✓	✗	Rotate এর সময় carry last bit এ যায়

♦ উদাহরণগুলো সহ ব্যাখ্যা 🧩

1 Zero Flag (Z)

```
MOVs R0, #5
SUBs R0, R0, #5 ; R0 = 0 → Z=1
```

- ✓ Z = 1 কারণ ফলাফল 0
 - ✗ N=0 কারণ রেজাল্ট নেগেটিভ না
-

2 Negative Flag (N)

```
MOVs R0, #-1 ; R0 = 0xFFFFFFFF
```

- ✓ N=1 কারণ MSB=1
 - ✓ Z=0 কারণ রেজাল্ট 0 না
-

3 Carry Flag (C)

```
MOVs R0, #0xFF
ADDs R0, R0, #1 ; 0xFF + 1 = 0x100 → Carry = 1
```

- ✓ C=1 (Unsigned overflow)
 - ✗ Z=0 (রেজাল্ট 0 না)
-

4 Overflow Flag (V)

```
MOVs R0, #127 ; 0x7F (signed +127)
ADDs R0, R0, #1 ; ফলাফল = 0x80 (signed -128)
```

- ✓ V=1 (signed overflow)
✗ C=0 (Unsigned carry হয়নি)
-

◆ লজিক্যাল অপারেশনে **Carry/Overflow** হয় না

MOV_S R0, #0xAA
AND_S R0, R0, #0x55 ; ফলাফল 0x00 → Z=1, C/V অপরিবর্তিত

◆ শিফট অপারেশনে **Carry** = শেষ শিফট হওয়া বিট

MOV_S R0, #0x01
LSR_S R0, R0, #1 ; ফলাফল = 0x00, Carry = 1 (শেষে শিফট হওয়া বিট)

◆ **Compare (CMP)** শুধুমাত্র ফ্ল্যাগ আপডেট করে

MOV_S R1, #5
CMP R1, #5 ; R1 == 5 → Z=1
CMP R1, #10 ; R1 < 10 → C=1 (no borrow)
CMP R1, #2 ; R1 > 2 → C=0 (borrow)

⚙ Summary Table (Quick Memory)

Flag	Meaning	Set (1) When	Example
N	Negative	MSB=1	-1 = 0xFFFFFFFF
Z	Zero	Result=0	5-5=0
C	Carry	Unsigned overflow/borrow none	0xFF+1
V	Overflow	Signed overflow	127+1



How to Check Which Flag is Set (ARM Cortex-M)

ARM এ **CPSR (Current Program Status Register)** নামে একটা রেজিস্টার আছে —
এখানেই সব **NZCV flags** সেভ থাকে 📌

Bit	Name	কাজ
31	N	Negative
30	Z	Zero
29	C	Carry
28	V	Overflow

❖ ১) Flags পড়া (Check করা)

তুমি নিচের মতো করে **CPSR** (বা **Cortex-M** এ **APSR**) পড়তে পারো:

```
MRS R0, APSR      ; APSR → R0 তে কপি
```

এখন R0 তে NZCV bits থাকবে:

```
R0[31] = N  
R0[30] = Z  
R0[29] = C  
R0[28] = V
```

❖ ২) নির্দিষ্ট ফ্ল্যাগ চেক করা (Bitwise Test)

```
MRS R0, APSR      ; সব ফ্ল্যাগ পড়া  
TST R0, #(1 << 30) ; Z ফ্ল্যাগ চেক (Zero flag)  
BNE flag_is_set    ; যদি Z ফ্ল্যাগ 1 হয়, তাহলে জাম্প করো
```

- ◆ এখানে **TST** মানে → AND করে শুধু ফ্ল্যাগ bit চেক করা
- ◆ $(1 \ll 30) = Z \text{ flag position}$

◆ ৩ প্রত্যেকটা **Flag** চেক উদাহরণ 🔍

Flag	Bit	Check Instruction	মানে
N	31	<code>TST R0, #(1 << 31)</code>	Negative হলে
Z	30	<code>TST R0, #(1 << 30)</code>	Result = 0 হলে
C	29	<code>TST R0, #(1 << 29)</code>	Carry হলে
V	28	<code>TST R0, #(1 << 28)</code>	Overflow হলে

◆ ৪ **Conditional Branch** দিয়ে ফ্ল্যাগ চেক (সবচেয়ে সহজ উপায়)

ARM এ প্রতিটি ব্রাঞ্চ/ইনস্ট্রাকশনে **condition suffix** আছে (eq, ne, mi, pl, cs, cc, vs, vc)
এগুলো সরাসরি ফ্ল্যাগের উপর ভিত্তি করে কাজ করে 👉

Condition	চেক করে	মানে	উদাহরণ
EQ	Z=1	Equal (Result = 0)	<code>BEQ label</code>
NE	Z=0	Not Equal	<code>BNE label</code>
MI	N=1	Negative	<code>BMI label</code>
PL	N=0	Positive	<code>BPL label</code>

CS / HS	C=1	Carry set / Higher or Same	BCS label
CC / LO	C=0	Carry clear / Lower	BCC label
VS	V=1	Overflow set	BVS label
VC	V=0	Overflow clear	BVC label

◆ Practical উদাহরণ

➤ Example: Compare দিয়ে Flag Check

```

MOVSI R1, #5
CMP R1, #5
BEQ equal_case      ; যদি Z=1 → equal
BNE not_equal_case  ; যদি Z=0 → not equal

```

➤ Example: Subtraction → Negative Check

```

MOVSI R0, #3
SUBS R0, R0, #5      ; R0 = -2 → N=1
BMI negative_case    ; যদি Negative → N=1
BPL positive_case    ; যদি Positive → N=0

```

➤ Example: Overflow Check

```

MOVSI R0, #127
ADDS R0, R0, #1      ; signed overflow → V=1
BVS overflow_case    ; যদি Overflow হয়
BVC ok_case          ; যদি না হয়

```

◆ সব একসাথে (Diagnostic Print Example)

```
MRS R0, APSR
```

```
TST R0, #(1 << 31)
```

```
BNE NegSet
```

```
TST R0, #(1 << 30)
```

```
BNE ZeroSet
```

```
TST R0, #(1 << 29)
```

```
BNE CarrySet
```

```
TST R0, #(1 << 28)
```

```
BNE OverflowSet
```

Summary Shortcut

Flag	Bit	Conditional Mnemonic	Example
N	31	MI / PL	BMI / BPL
Z	30	EQ / NE	BEQ / BNE
C	29	CS / CC	BCS / BCC
V	28	VS / VC	BVS / BVC