STM32 Blue Pill / Cortex-M3 Instruction Set (Summary)

1️⃣ Data Processing Instructions

Instruction	Description
`MOV Rd, #imm`	Move immediate value to register
`MOV Rd, Rm`	Move value from register to register
`MVN Rd, Rm`	Move NOT of register
`ADD Rd, Rn, Rm/#imm`	Add
`SUB Rd, Rn, Rm/#imm`	Subtract
`RSB Rd, Rn, Rm`	Reverse subtract
`ADC Rd, Rn, Rm`	Add with carry
`SBC Rd, Rn, Rm`	Subtract with carry
`AND Rd, Rn, Rm/#imm`	Bitwise AND
`ORR Rd, Rn, Rm/#imm`	Bitwise OR
`EOR Rd, Rn, Rm/#imm`	Bitwise XOR
`BIC Rd, Rn, Rm`	Bit clear
`CMP Rn, Rm/#imm`	Compare (sets flags)
`CMN Rn, Rm`	Compare negative (adds and sets flags)

`TST Rn, Rm`	Test bits (AND and set flags)

2️⃣ Load / Store Instructions

Instruction	Description
`LDR Rd, [Rn, #offset]`	Load word from memory
`LDRB Rd, [Rn, #offset]`	Load byte
`LDRH Rd, [Rn, #offset]`	Load half-word
`LDRSB Rd, [Rn, #offset]`	Load signed byte
`LDRSH Rd, [Rn, #offset]`	Load signed half-word
`STR Rd, [Rn, #offset]`	Store word
`STRB Rd, [Rn, #offset]`	Store byte
`STRH Rd, [Rn, #offset]`	Store half-word

Example: `LDR R0, [R1,#4]` → load 32-bit word from address R1+4

3️⃣ Branch / Control Flow

Instruction	Description
`B label`	Unconditional branch
`BL label`	Branch with link (call function)

`BX Rm`	Branch to register (return from function)
`BLX Rm`	Branch with link to register
`BEQ label`	Branch if equal (Z=1)
`BNE label`	Branch if not equal (Z=0)
`BGT/BGE/BLT/BLE/...`	Conditional branches based on flags

4️⃣ Stack / Subroutine

Instruction	Description
`PUSH {Rn,...,LR}`	Push registers to stack
`POP {Rn,...,PC}`	Pop registers from stack (PC = return)
`MSR`	Move value to special register (PSR, CONTROL)
`MRS`	Move special register to general register

5️⃣ Bit Manipulation / Misc

Instruction	Description
`LSL Rd, Rm, #imm`	Logical shift left
`LSR Rd, Rm, #imm`	Logical shift right
`ASR Rd, Rm, #imm`	Arithmetic shift right
`ROR Rd, Rm, #imm`	Rotate right

`TST Rn, Rm`	Test bits
`CPSIE i`	Enable interrupts
`CPSID i`	Disable interrupts

6️⃣ Special Instructions for STM32 Peripherals

- Memory-mapped I/O → use **LDR/STR** for peripheral registers
- Example: Toggle LED on PC13

```
LDR R0,=GPIOC_BASE
LDR R1,[R0,#GPIOC_ODR]
EOR R1,R1,#(1<<13)
STR R1,[R0,#GPIOC_ODR]
```

- Cortex-M3 এ **all peripheral interaction = memory-mapped**, কোনো special I/O instruction নেই

🔹 Notes

1. STM32 Blue Pill = Cortex-M3 → **ARMv7-M Thumb / Thumb-2** only
2. All instructions are **16-bit or 32-bit Thumb**
3. Arithmetic, logical, load/store, branch, stack, bit-manipulation supported
4. Peripheral access via **LDR/STR** on memory-mapped addresses