ARM Assembly Cheat Sheet for Visual2

Move & Load

MOV: Move constant or register to another (e.g., MOV R0, #5)

MVN: Move NOT of a register (e.g., MVN R1, R2)

LDR: Load data from memory or constant (e.g., LDR R0, =0x100)

STR: Store data to memory (e.g., STR R0, [R1])

ADR: Load address of a label (e.g., ADR R0, label)

Arithmetic & Logic

ADD: Add two registers (e.g., ADD R0, R1, R2)

SUB: Subtract (e.g., SUB R0, R1, #1)

RSB: Reverse subtract (e.g., RSB R0, R1, R2)

AND: Bitwise AND (e.g., AND R0, R1, R2)

ORR: Bitwise OR (e.g., ORR R0, R1, R2)

EOR: Bitwise XOR (e.g., EOR R0, R1, R2)

Shifts

LSL: Logical Shift Left (e.g., LSL R0, R1, #1)

LSR: Logical Shift Right (e.g., LSR R0, R1, #1)

ASR: Arithmetic Shift Right (e.g., ASR R0, R1, #1)

Comparison & Branching

CMP: Compare and set flags (e.g., CMP R1, #0)

B: Branch unconditionally (e.g., B label)

BEQ: Branch if equal (e.g., BEQ label)

BNE: Branch if not equal (e.g., BNE label)

BLE: Branch if less or equal (e.g., BLE label)

BGE: Branch if greater or equal (e.g., BGE label)

Memory Operations (Indexing Modes)

LDR R0, [R1]: load from address in R1

STR R0, [R1]: store to address in R1

STR R0, [R1], #4: store then post-increment pointer

LDR R0, [R1, #4]!: pre-increment pointer then load

Notes & Gotchas

Visual2 doesn't support: PUSH, POP, BL, BX LR, syscalls.

For subroutines: use B label for 'call', explicit return labels.

Example Snippet

```
MOV R0, #10
MOV R1, #0x100
STR R0, [R1], #4
CMP R0, #0
BEQ final
B loop

loop
    SUB R0, R0, #1
    STR R0, [R1], #4
    CMP R0, #0
    BNE loop
final
    MOV R7, #0
```