ARM-7 Fetch → Decode → Execute

Computer Arch. LEB

ARM → Advanced RISC Machine

① large array of register
② small no. of addressing modes
③ uniform instruction

CISC (complex)

Data processing Method

1. ARM developed by Acorn Computers.
2. RISC → Reduced Instruction Set Computer
   - Von Neumann (share same bus)
3. ARM micro processor
   - Von Neumann (share same bus)
   - Harvard Arch (Data & Instruction uses separate bus)

4. ARM Registers (32 bits)
   - 13 general purpose → [hold data or address]
   - 4 special purpose → SP, LR, PC, CPSR → perform particular func.

Total = 37
20 (hidden) ↳ banked register

1. SP → Stack Pointer (R13)
   ↳ stores top of stack
2. LR → Link Register (R14) → stores the return address
3. PC → Program Counter (R15) → contains address to next inst.
4. CPSR → current Program Status Register → to monitor & control
   SPSR → saved " ——— . → to store the current value of CPSR

for CPSR

5. Condition flags
   - N (-ve value)
   - Z (zero result)
   - C (carry)
   - V (overflow)
   - Q (overflow & saturation)

SWI &11 software interrupt
↳ to enter supervisor mode

ADDS / SUBS → to set the flag

6. MVN - Move Negative (Negation)
7. LSL → Logical shift Left (mul $2^n$) — << n = no. of times
8. LSR → Logical shift Right (Divide $2^n$) — >>
* STR → Store Reg. in Memory. $R_0 \rightarrow R_{14}$
9. AREA → Make a new Block of Data
10. ENTRY → Declare entry point
11. DCD → Allocate one or more words (32bit)
    - (8) DCB
    - (16) DCW
12. LDR → Load Data into Register

| Earlier | ARM |
|---|---|
| 16 bit Add. Bus | 32 bit |
| 8 bit Data Bus | 32 bit |

13. Processor modes
    - 6 privileged
    - 1 non privileged (user Mode)

0X01 → 1

Moments
PAGE :
DATE: / /

same
```
MUL  R0, R1, #8        ; R0 = R2 * 8
MOV  R0, R2, LSL #3    ; R0 = R2 * 8
```

Division
```
MOV  R1, R3, LSR #7    ; R1 = R3/128
```

# for Indirect Addressing, we need to do memory mapping

Go to Debug > Map Memory > Put the values in Map Range (all caps) separated by comma > checkmark all the boxes >> Map Range > close > click on memory (in bottom) > type address

0X0000...

6AA

| | | |
|---|---|---|
| CMP | Compare. | |
| BEQ | Branch on equal | |
| BNE | Branch Not equal. | |
| BLS | Branch on less then | |
| BLE | Branch on less then or equal | |
| BGT | Branch greater then. | |
| BGE | Branch on greater than or equal. | |
| | TR | Temporary Reg |
| | DR | Data Reg |
| | AR | Address Reg |
| even odd | IR | Instruction Reg |
| Q-11 | AC | Accumulator |
| | PC | Program counter |
| | INPR | Input Reg |
| | OUTR | Output Reg. |