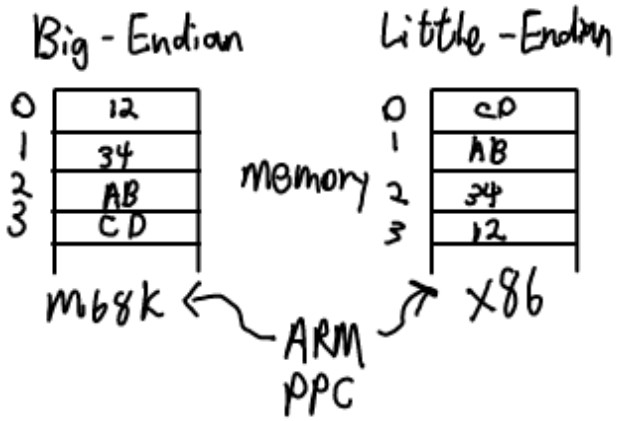
Computers

6) Memory

- a processor can access a finite amount of physical memory, determined by the # of address pins.
- memory is measured in binary units but reported with S.I. prefixes (e.g. 1kb = 1024 bytes)
- Hard disks are measured in decimal units (e.g. 1 kb = 1000 bytes)
- IEC Introduced binary units to eliminate confusion (e.g. 1 kiB = 1024 bytes)
- endianness
  - big-endian: MSB (most sig. byte) is at low address. LSB at high address.
  - little-endian: MSB at high address LSB at low address.

e.g. $\underbrace{12}_{MSB}34AB\underbrace{CD}_{LSB}$



ARM

1) Background

- Acorn / Advanced RISC Machines
- license designs to other companies to manufacture.
- target low power / low cost
- www.davespace.co.uk

2) Design Principles

- RISC but with some CISC characteristics.

RISC:
- fixed instruction size
- load/store architecture

CISC:
- autoincrement/decrement addressing modes
- move multiple values from regs. to mem. in one instruction
- condition codes

3) Memory

- data sizes:
  - word = 32 bits
  - half-word = 16 bits
  - byte = 8 bits
- word addresses are "word aligned" (multiple 4)
- little or big-endian (lab: little-Endian)
- loads of half-words or bytes are zero extended or sign extended to 32 bits

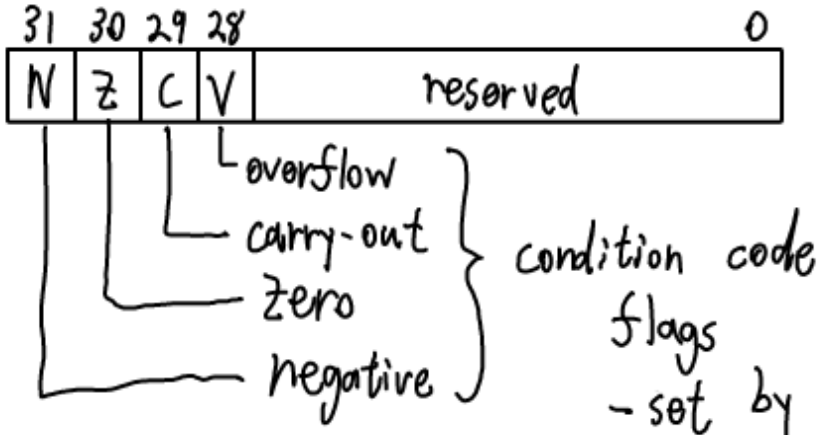
unsigned (8) 1000 → 00001000
signed (-8) 1000 → 11111000

unsigned (4) 0100 → 00000100
signed (4) 0100 → 00000100

4) Registers

- all regs are 32-bits.
- 13 general purpose registers: R0 – R12
  - R13 is the stack pointer (SP)
  - R14 is the link register (LR)
  - R15 is the program counter (PC)

- program status register (PSR)

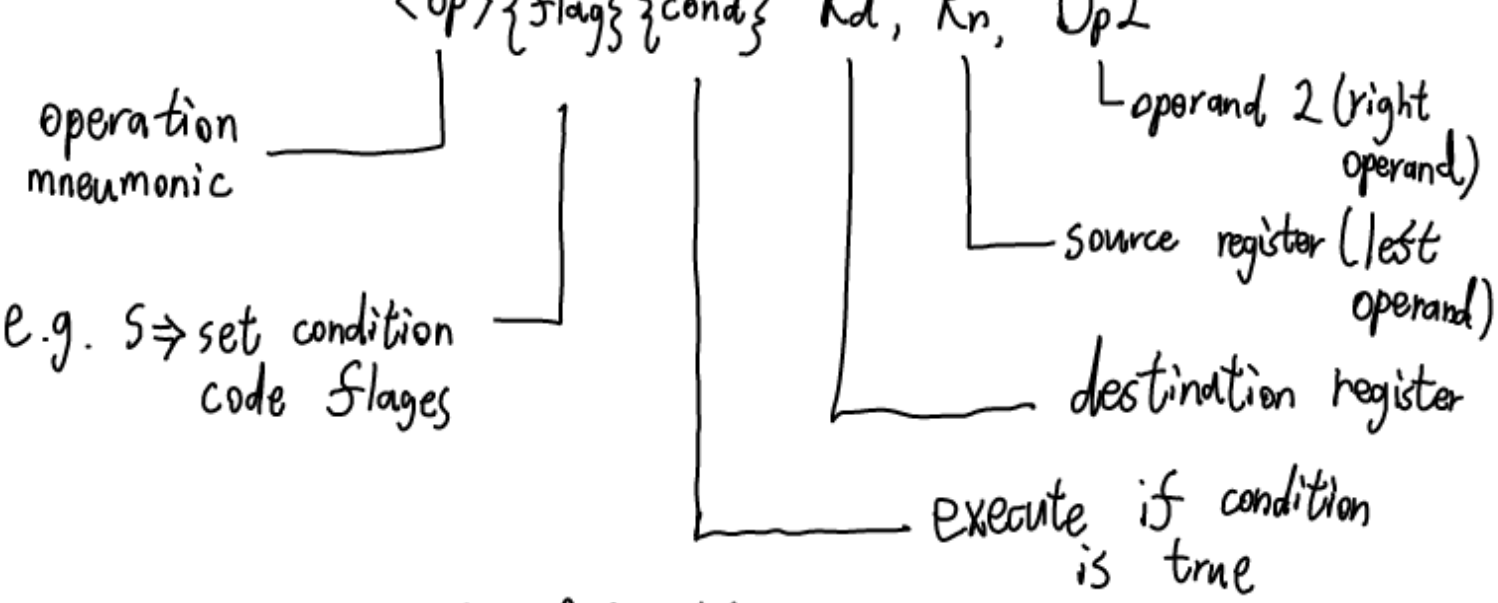


5) Instruction Set

- 3 variants:
  - ARM : 32-bit
  - Thumb : 16-bit (compact, limited instrs. & operands)
  - Thumb-2 :- mix of 16 and 32-bit
    - cortex-M3 in lab.

5.1) Data Processing Instructions

- most have this format:

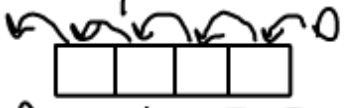


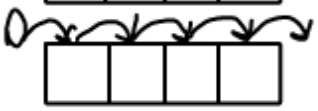
e.g. ADD EQ R2, R0, #1

if Z==1, R2 ← [R0]+1

zero flag

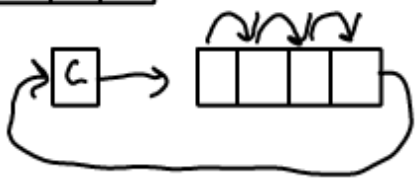
- operand 2
  - an 8-bit constant (optionally rotated)
  - register value (Rm) optionally shifted

LSL : logical shift left  (multiply)

LSR : logical shift right  (unsigned divide by 2)

ASR : Arithmetic shift right  (signed divide by 2)

ROR : rotate right 

RRX : rotate right extended 

e.g. ADD r2, r0, r1, LSL #2

Rd Rn Rm

r2 ← [r0] + [r1] << 2

= r2 ← [r0] + 4 * [r1]

- arithmetic

  ADD, ADC (add w/ carry), SUB, SBC, RSB (reverse subtract)

- logical (bit wise)

  AND, ORR, EOR, BIC, ORN

  ORN: or not

  BIC: and not (bit-wise clear)

(no NOT) - EOR Rd, Rn, #0xFFFFFFFF