Computer Architecture

Curated computer science study track (not a degree, no licensure claim). Original explanations and practice only, following the reviewed course handout.

The machine below the program

- 6.1-6.2 the contract and the machine that honors it; 6.3-6.4 make it fast; 6.5-6.6 connect it.
- Every timing claim here is computed from stated parameters and re-derived by the suite.

Instruction set architecture: the stable contract

1. **Formats R-type**: opcode, rd, rs, rt, shift, funct - 28 bits used. **I-type** trades the third register for a 16-bit immediate.
2. **Range** Immediate spans -32 768 to 32 767, so 40 000 cannot encode; word offsets are multiples of 4, reaching 64 KiB from the base.
3. **Zero** addi R0, R0, 0 is the no-op; add R5, R0, R0 zeroes a register; beq against R0 needs no constant.

Instruction encodings

two formats, one 32-bit width

R-type



I-type



field widths in bits below each box - R-type: 4+4+4+4+4+8 = 28, padded to 32; I-type: 4+4+4+4+16 = 32

example: add R3, R1, R2 = 0001 0011 0001 0010 0000 00000001

Two formats, one width: **R-type** carries three registers and a function code; **I-type** trades the third register for a 16-bit immediate.

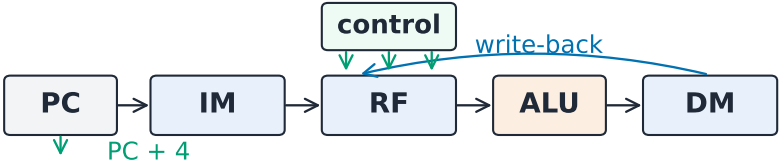
exercises.lol exercises.com.cn

Datapath and control: six bits per instruction

1. **Path** PC to IM to RF to **ALU** to DM to write-back: lw walks all of it - 200 + 100 + 200 + 250 + 150 = 900 ps.
2. **Signals** lw R4, 8(R1): RegWrite 1, MemRead 1, MemToReg 1, ALUSrc 1, Branch 0 - six bits tell the story.
3. **Store** sw flips the port: RegWrite 0, MemWrite 1, and MemToReg becomes a do-not-care.

Single-cycle datapath

one clock, one instruction



control bits: RegWrite · MemRead · MemWrite · MemToReg · ALUSrc · Branch

for lw R4, 8(R1): RegWrite 1 · MemRead 1 · MemToReg 1 · ALUSrc 1 · Branch 0

cycle time = the slowest path (lw: IM + RF + ALU + DM + wiring = 900 ps)

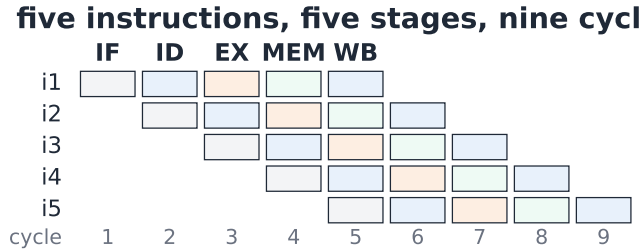
One clock, one instruction: fetch at PC, read registers, compute at the **ALU**, touch data memory if asked, write back, and choose the next PC.

exercises.ror exercises.com.cn

Pipelining: n + k - 1 cycles

1. **Formula** n instructions in k stages take n + k - 1 cycles: five instructions take 9, not 25; at n = 100, 104.
2. **Stages** Slowest stage rules: IF 250, ID 150, EX 200, MEM 250, WB 100 - every stage 250 ps against the 900 ps single cycle.
3. **CPI** 1000 instructions, 25 percent loads, half load-use: 1000 + 125 stalls + 4 fill = 1129 cycles, **CPI** 1.129.

Pipeline timeline



each instruction starts one cycle after its predecessor; one retires per cycle once the pipe is full

n instructions take n + k - 1 cycles: here 5 + 5 - 1 = 9, not 25

Five instructions through five stages: 9 cycles total, not 25 - the pipe fills once and drains once.

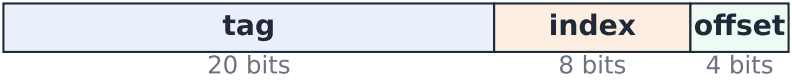
exercises.lol exercises.com.cn

Memory hierarchy: split the address, price the miss

1. **Split** 4 KiB, 16-byte lines: 256 lines, so 8 **index** bits, 4 offset, 20 **tag**; 0x4A8 lands at offset 8, line 74.
2. **AMAT** 1 + miss ratio times penalty: at 2 percent and 100 cycles, 3 cycles; halve the misses to reach 2.
3. **Layout** Row-major over 16-byte lines misses 1 in 4; stride-16 misses every access - layout is a performance feature.

Address decomposition and hierarchy

a 32-bit address split by the cache geometry



4 KiB cache, 16-byte lines: 256 lines - so 8 index bits, 4 offset bits, 20 tag bits

0x4A8: offset 8 (byte 8 of the line), index 74 (line 74 of 256)



a miss at one level is a request at the next: AMAT = hit + miss ratio x penalty

The 32-bit address splits by the cache geometry into **tag**, **index** and offset.

Input/output: three ways to move a block

1. **Poll** Flag every 2 microseconds, poll 50 cycles at 250 ps: 160 polls per event, 8000 cycles of attention per word.
2. **Interrupt** The handler costs 200 cycles per word; receiving 1024 words one interrupt each: 204 800 cycles of overhead.
3. **DMA** One 500-cycle setup plus one 200-cycle completion for 1024 words: 0.68 cycles per word - 292 times less than per-word interrupts.

Three I/O paths

three ways to move one block

polling	interrupts	DMA
core spins on status	device signals ready	engine moves the block
160 polls per event	200-cycle handler	one interrupt at the end
8000 cycles per word	200 cycles per word	0.68 cycles per word

CPU attention per word: 8000 vs 200 vs 0.68 cycles - the same bytes, three bills

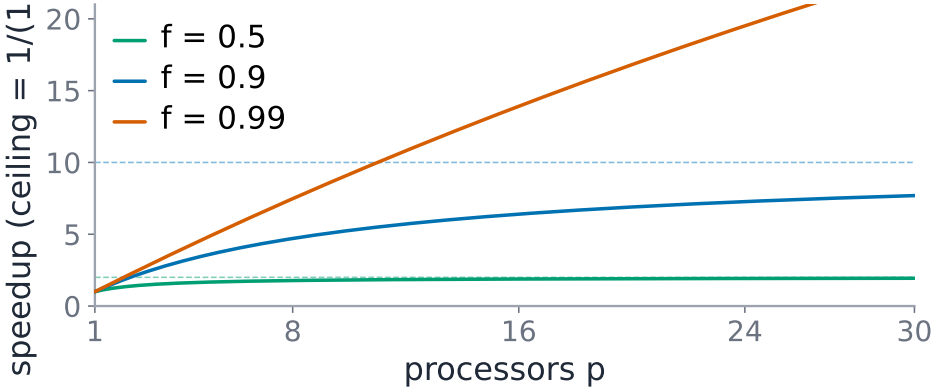
rule of thumb: poll what must be sampled, interrupt what arrives, DMA what streams

> Polling burns the core; interrupts charge per event; **DMA** charges once per block.

Parallelism and performance: read Amdahl

1. **Ceiling** f = 0.9 caps every machine at 10 times; p = 8 gives 4.71, p = 16 gives 6.40 - the second eight add 1.7.
2. **Price** 36 s serial + 64 s parallel: f = 0.64; on 4 processors 1.92 times and 52.0 s, on 8 it is 2.27 and 44.0 s.
3. **Exact** f = 0.99 reaches exactly 50 times at p = 99: diminishing returns are structural.

Amdahl speedup curves



The serial fraction caps every speedup curve.

Check yourself

- A 16-bit signed immediate tops out at…? (32 767.)
- Five instructions, five stages, total cycles…? (9.)
- 4 KiB cache, 16-byte lines: **index** bits…? (8.)
- f = 0.9: the speedup ceiling…? (10 times.)