Hardware & Logic

A-Level Computer Science ▪ Topic 15

A-Level Computer Science

What we will cover

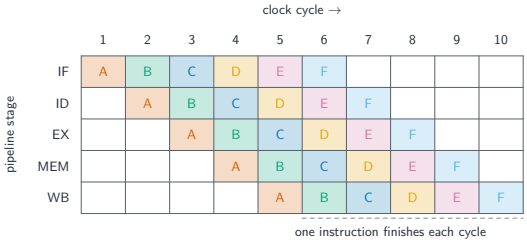
1. RISC vs CISC
2. Pipelining
3. Flynn's taxonomy
4. Boolean algebra & K-maps
5. Adders & flip-flops
6. Recap

RISC vs CISC processors

- **RISC** 精简指令集: few, simple, fixed-length instructions – one per cycle
- **CISC** 复杂指令集: many complex instructions, variable length

RISC keeps each instruction simple so it pipelines well and uses less power – which is why phones use it.

Pipelining



cpu

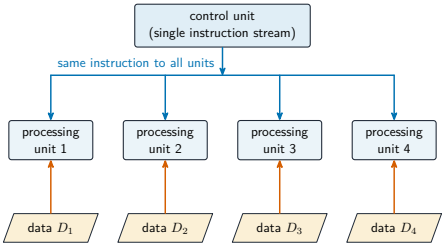
cooler

Flynn's taxonomy

Classifies parallelism by instruction and data streams:

- SISD – one of each (a simple CPU)
- SIMD – one instruction, many data (a GPU)
- MIMD – many of each (multi-core, clusters)

A **GPU** 图形处理器 is SIMD: the same operation on thousands of pixels at once.



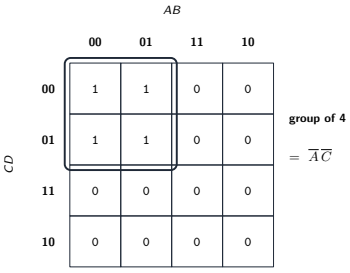
Boolean algebra and K-maps

Simplify logic before building it:

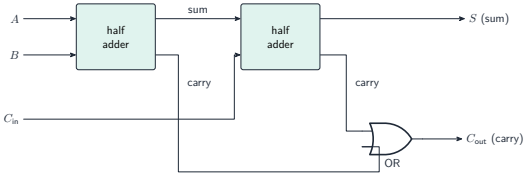
$$A + \overline{A}B = A + B, \quad A(A + B) = A$$

- **De Morgan** 德摩根: $\overline{A \cdot B} = \overline{A} + \overline{B}$
- a **Karnaugh map** 卡诺图 groups 1s in powers of two to read off the simplest expression

Fewer gates ⇒ cheaper, faster, cooler circuits.



Adders and flip-flops



gpu card

Topic 15 – key takeaways

1 RISC/CISC

simple vs complex instruction sets

2 Pipeline

overlap stages – higher throughput

3 Flynn

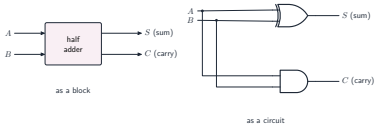
SISD / SIMD / MIMD

4 Logic

simplify with K-maps; adders + flip-flops

Check yourself

1. Which pipelines better, RISC or CISC?
2. Does pipelining speed up one instruction?
3. What Flynn class is a GPU?
4. Sum bit of a half adder: which gate?



Answers **1.** RISC **2.** no, throughput **3.** SIMD **4.** XOR