Universiteit Leiden

EssCS - Topic 2

Introduction to Computer Architecture

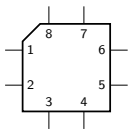
Lecture 5, 01.10.2024

Nuša Zidarič

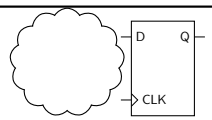
Group Practical Assignment (Mini-Project) - see Brightspace

Levels of Abstraction - part III.

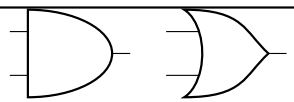
(bottom-up approach)



Processor level — Topic2



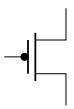
Register-Transfer Level — Topic1-part2



Gate level

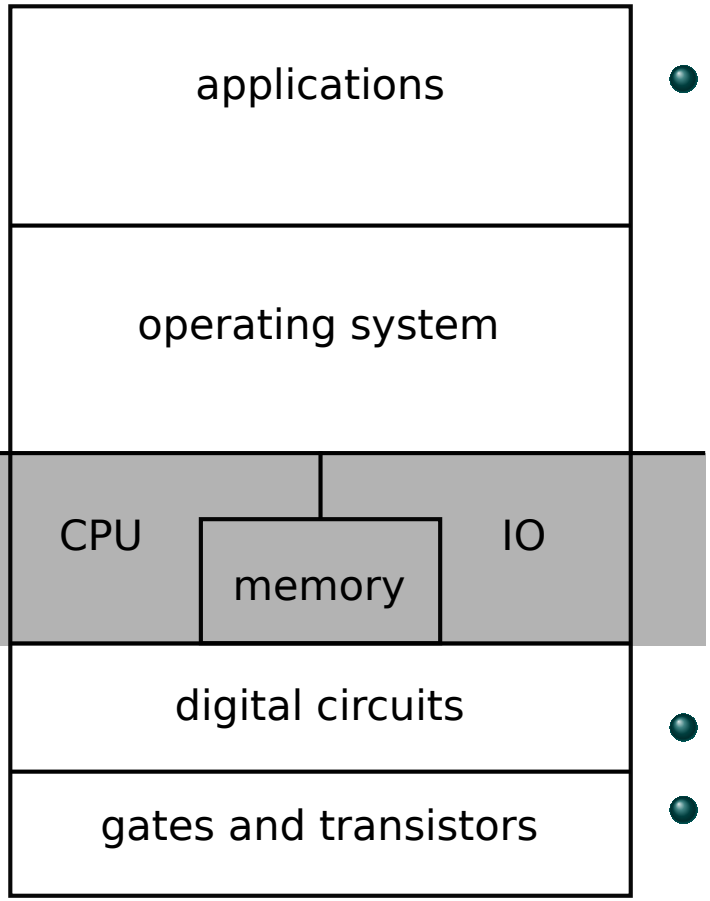
Topic1-part1

Switch level



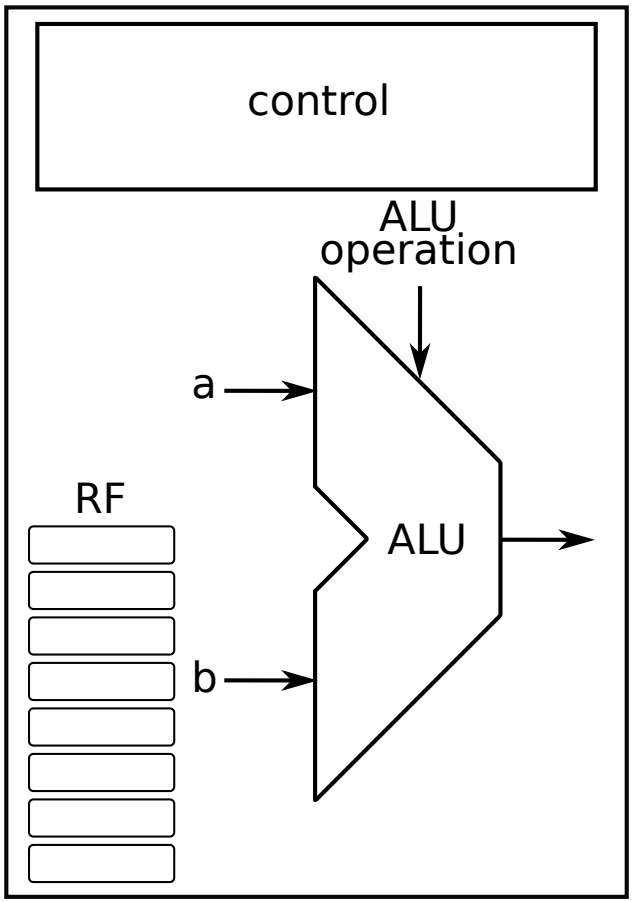
Transistor level

Introduction to Computer Architecture



- CPU = Central Processing Unit
  - we will talk about simple General Purpose CPUs
  - however: the domain plays an important role: hardware components have to be selected and interconnected in a way that meets *functional*, *performance* and *cost* requirements
  - domains: IoT, mobile devices, desktops, servers, supercomputers, ...
- memory
- we will only briefly mention IO (Input/Output)

  (example: hard drive)

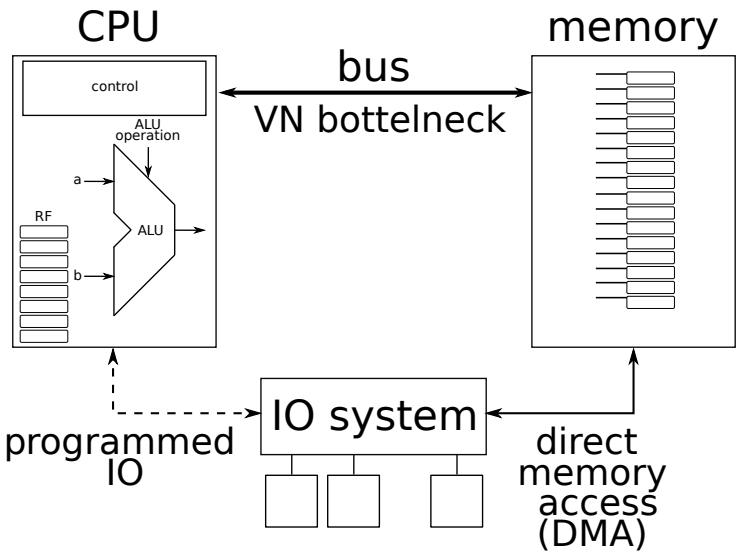


- For now we will assume that the CPU consists of the following 3 components:
  - ALU = Arithmetic-Logic Unit
    ALU performs basic operations, for example addition, subtraction, logic operations, ...
  - RF = Register File
    A collection of registers that provide the inputs to the ALU and hold (intermediate) results
  - Control unit that sets the control signals to select the desired ALU operation, appropriate register, ...
- CPU operation viewed in 3 steps:

  fetch, execute, write-back

- But how does control unit know what to do ? it decodes the *instruction* (insn.)

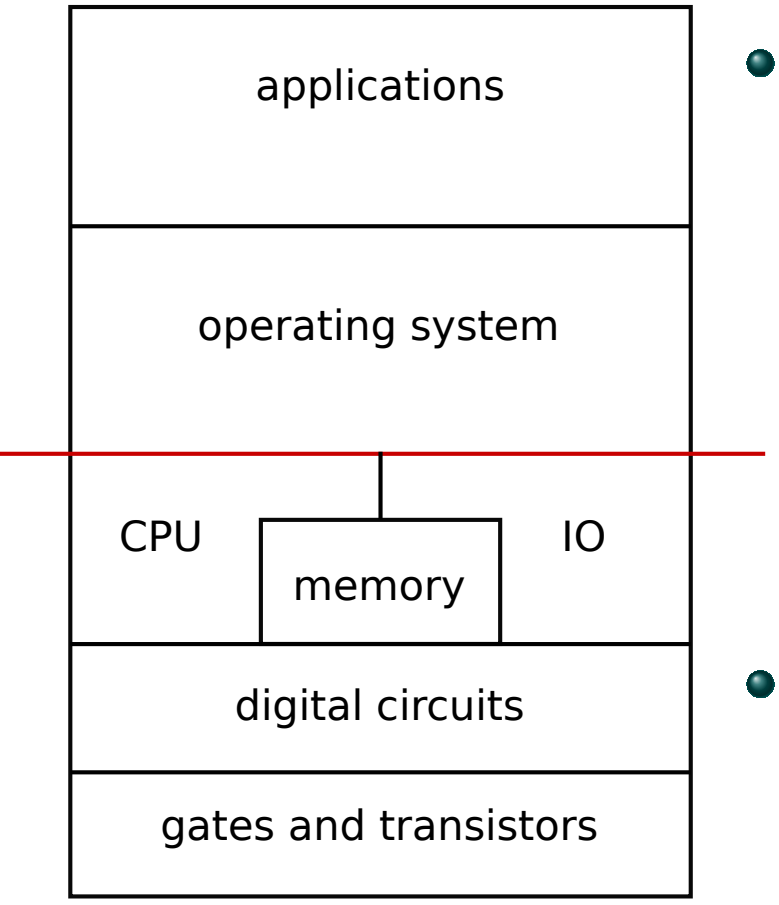
- von Neumann model



- Operation of VN model is completely defined with a sequence of instructions (sequential model)
- CPU reads insns from the main memory and executes them
- memory: array of memory words (address, data)
- bus: address, control (read/write), data
- which insn is next?
  CPU has a program counter (PC) register
  PC holds the address of the next insn

- But what are the instructions?

- Instruction Set Architecture



- ISA is a SW/HW interface (abstract model)

  a functional contract (programmers view)

  defines the behavior of the CPU by
  - defining operations
  - internal storage locations
    (eg. programmer visible registers)

- mircoarchitecture: physical implementation

  different design decisions and trade-offs

  same ISA can be implemented by different chips

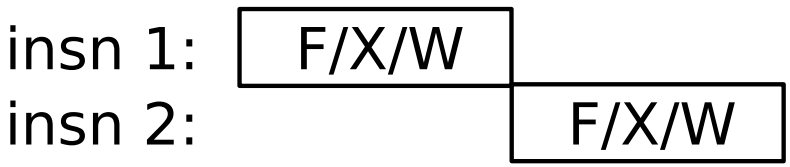
machine code	assembly	meaning	CPU action
000000 01000 01001 01010 00000 100000	add $t2, $t0, $t1	add c, a, b	$c = a + b$

other examples: lw (load word), sw(store word), j (jump), bne (branch if not equal)

CPU datapath

- recall CPU operation viewed in 3 steps: fetch (F), execute (X), write-back (W)

single cycle datapath



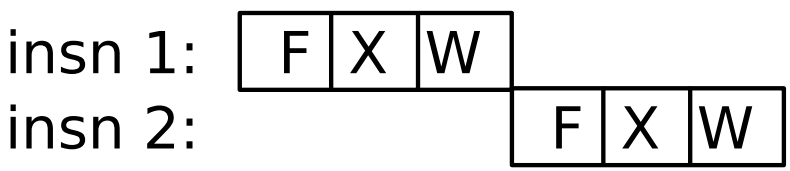
- true atomic F/X/W loop

  CPI $= 1$ (cycles-per-instruction)

  $Tput = 1$ (instructions-per-cycle)

  t_{clk} =large

multicycle datapath

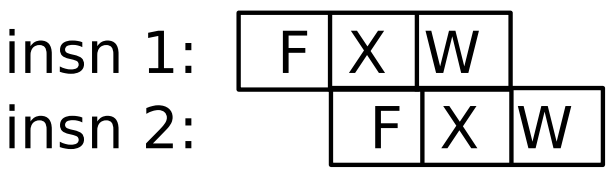


- one insn over multiple cycles

  CPI > 1 (and different for different insns)

  $Tput$ = small and t_{clk} =small

pipelined datapath



- overlap as much as possible

  CPI $= 1$ and $Tput = 1$ and t_{clk} =small

  NOTE: actual CPI and Tput are worse due to stalls