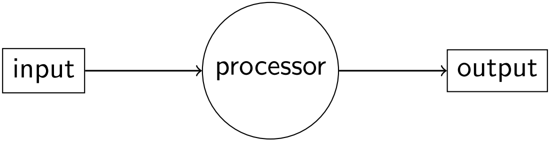
Introduction to the Use of Computers

Christophe Rhodes
c.rhodes@gold.ac.uk

Autumn 2012, Fridays: 10:00–12:00: WTA & 15:00–17:00: WHB 300

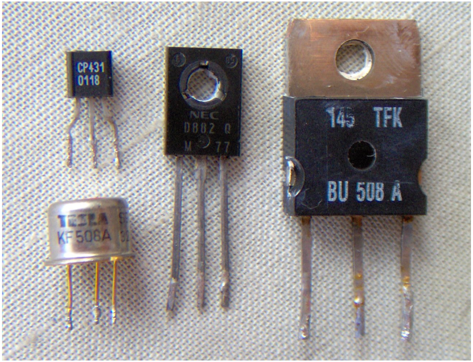
Processor

What is a computer?



Processor

Transistors



Wikimedia Commons (user FDominec)
CC-BY-SA 3.0

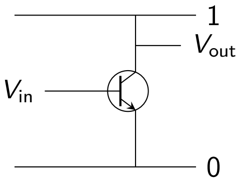
Uses:

- amplifier;
- switch.

Processor

Transistors

Transistor as switch:

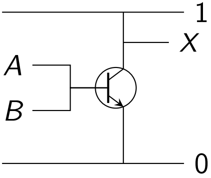


- if V_{in} is 1, the transistor's resistance is low;
  - so V_{out} is (close to) 0.
- if V_{in} is 0, the transistor's resistance is high;
  - so V_{out} is (close to) 1.

Processor

Transistor logic

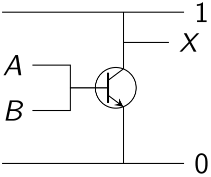
Transistors as logic components:



Processor

Transistor logic

Transistors as logic components:

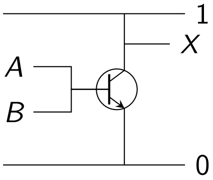


- if *either* A or B is 1, the transistor resistance is low
  - so X is (close to) 0;
- if *both* A and B are 0, the transistor resistance is high
  - so X is (close to) 1.

Processor

Transistor logic

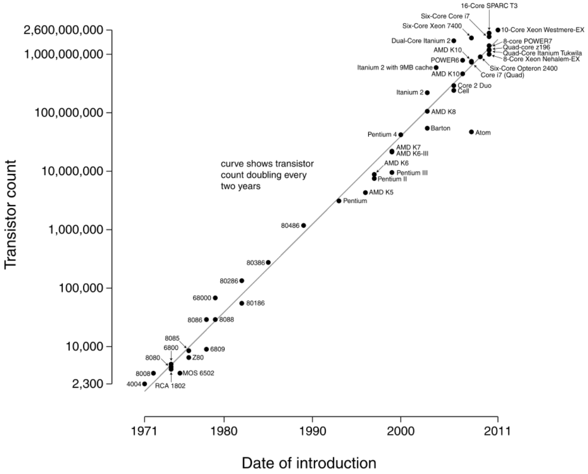
Transistors as logic components:



- if *either* A or B is 1, the transistor resistance is low
  - so X is (close to) 0;
- if *both* A and B are 0, the transistor resistance is high
  - so X is (close to) 1.
- NOR gate

Processor

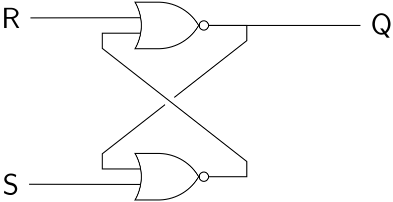
Moore’s Law



Wikimedia Commons (user Wgsimon)
CC-BY-SA 3.0

Processor

Transistor storage



Processor

ALU logic

Logical operations:

- identification of 0 with false and 1 with true;
- perform *bitwise* logic:
  - bit 0 of output is result of operation on bits 0 of inputs;
  - bit 1 of output is result of operation on bits 1 of inputs;
  - ...
  - bit 32 of output is result of operation on bits 32 of inputs.

Processor

ALU logic

Logical operations:

- identification of 0 with false and 1 with true;
- perform *bitwise* logic:
  - bit 0 of output is result of operation on bits 0 of inputs;
  - bit 1 of output is result of operation on bits 1 of inputs;
  - ...
  - bit 32 of output is result of operation on bits 32 of inputs.
- list of operations supported by processor varies:
  - NOT
  - AND, OR, XOR
  - ANDC2, ORC2
  - ANDC1, ORC1
  - ...

Processor

ALU arithmetic

Arithmetic operations:

- standard operations:
  - NEG
  - ADD, SUB
  - MUL, IMUL
  - DIV, IDIV

Processor

ALU arithmetic

Arithmetic operations:

- standard operations:
  - NEG
  - ADD, SUB
  - MUL, IMUL
  - DIV, IDIV
- shifting and rotating:
  - SHL, SHR
  - ROL, ROR

Processor

CPU instructions

What are the inputs and outputs?

- direct access: registers
  - small storage units;
  - directly addressable by CPU;
- (sometimes) direct access: memory
- (usually) transparent: CPU *cache*

Processor

CPU instructions

What are the inputs and outputs?

- direct access: registers
  - small storage units;
  - directly addressable by CPU;
- (sometimes) direct access: memory
- (usually) transparent: CPU *cache*

Memory operations:

- move values from RAM to registers
- move values from registers to RAM

Processor

FPU arithmetic

Integer formats:

- integers in the range $[0,2^{32})$
- variants on this theme – $[-2^{31},2^{31})$, $[0,2^{64})$

Floating point format:

- reduce maximum number of significant figures;
- increase numeric range:
  - single-precision floats: $[-2^{128},2^{128}]$
  - double-precision floats: $[-2^{1024},2^{1024}]$
- (sign,mantissa,exponent):
  - sign $\times$ mantissa $\times 2^{\text{exponent}}$

Processor

FPU arithmetic

Floating point format:

- represents numbers of the form
  - $\pm\frac{[0,2^{24})}{2^{24}\times 2^{[-128,128]}}$
  - (24-bit integers divided by powers of 2)

Processor

FPU arithmetic

Floating point format:

- represents numbers of the form
  - $\pm\frac{[0,2^{24})}{2^{24}\times 2^{[-128,128]}}$
  - (24-bit integers divided by powers of 2)

Consequences:

- many fractions can be represented:
  - $\frac{1}{2}$, $\frac{3}{8}$, $\frac{17}{256}$

Processor

FPU arithmetic

Floating point format:

- represents numbers of the form
  - $\pm\frac{[0,2^{24})}{2^{24}\times 2^{[-128,128]}}$
  - (24-bit integers divided by powers of 2)

Consequences:

- many fractions can be represented:
  - $\frac{1}{2}$, $\frac{3}{8}$, $\frac{17}{256}$
- many integers can be represented:
  - 1, 17, 2^{24}, $2^{24}+2$

Processor

FPU arithmetic

Floating point format:

- represents numbers of the form
  - $\pm \frac{[0,2^{24})}{2^{24} \times 2^{[-128,128]}}$
  - (24-bit integers divided by powers of 2)

Consequences:

- many fractions can be represented:
  - $\frac{1}{2}, \frac{3}{8}, \frac{17}{256}$
- many integers can be represented:
  - $1, 17, 2^{24}, 2^{24} + 2$
- some numbers can't be represented:
  - $2^{24} + 1$
  - $\frac{1}{3}$
  - $\frac{1}{10}, \frac{1}{100}$

Processor

FPU arithmetic

FPU arithmetic:

- perform *floating point* computations:
  - addition, subtraction
  - multiplication, division
  - square root, logarithms, trigonometric functions

Processor

FPU arithmetic

FPU arithmetic:

- perform *floating point* computations:
  - addition, subtraction
  - multiplication, division
  - square root, logarithms, trigonometric functions
- when answer cannot be exactly represented, rounding happens:
  - 0.1×0.1

Processor

FPU arithmetic

FPU arithmetic:

- perform *floating point* computations:
  - addition, subtraction
  - multiplication, division
  - square root, logarithms, trigonometric functions
- when answer cannot be exactly represented, rounding happens:
  - 0.1×0.1
- answers may not be what you expect:
  - 0.01×10

Processor

CPU

Machine code:

- binary encoding of instructions;
- binary encoding of data.

Processor

CPU

Machine code:

- binary encoding of instructions;
- binary encoding of data.

Central Processing Unit:

1. fetches next instruction;
2. executes instruction
   - possibly interacting with data;
   - possibly altering cpu state;
3. returns to step 1.

(Fetch-Execute cycle)

Processor

Input and Output

Input devices:

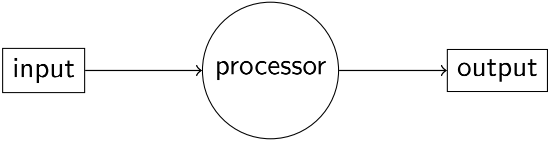
- keyboard;
- mouse;
- network card, camera, microphone, ...
- usb ports, serial ports, firewire, ...
- storage.

Output:

- screen;
- printer;
- network card;
- usb ports, serial ports, firewire, ...
- keyboard, headphones;
- storage.

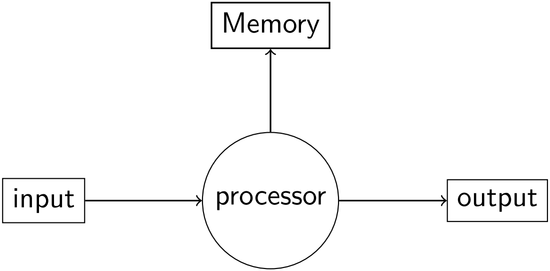
Operating System

Input and Output



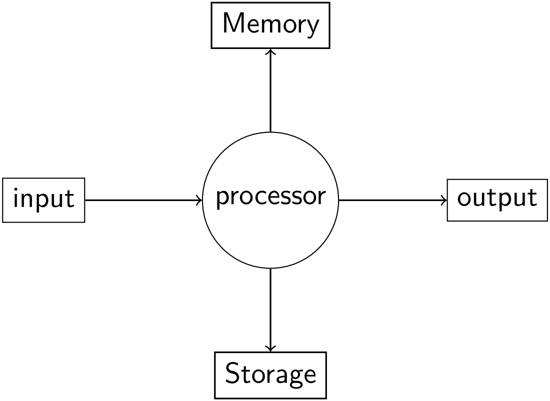
Operating System

Input and Output



Operating System

Input and Output



Operating System

Computer Buses

System Bus:

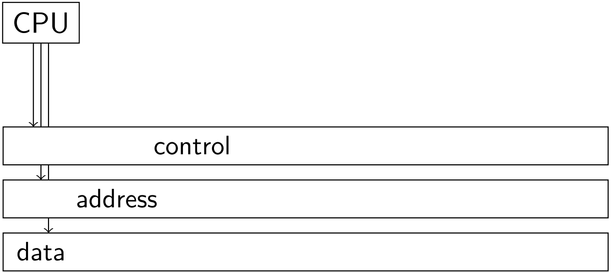
- simplified model;
- common in 1970s and 1980s.

Operating System

Computer Buses

System Bus:

- simplified model;
- common in 1970s and 1980s.

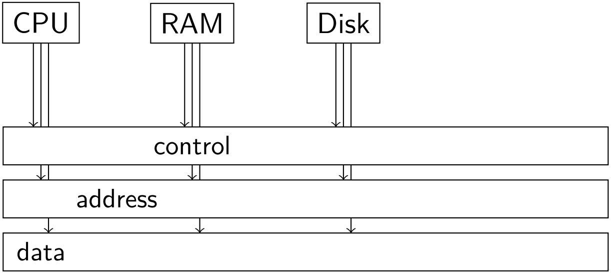


Operating System

Computer Buses

System Bus:

- simplified model;
- common in 1970s and 1980s.

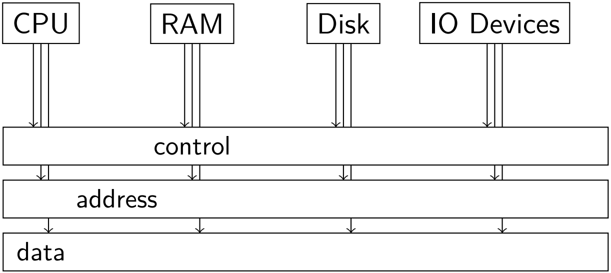


Operating System

Computer Buses

System Bus:

- simplified model;
- common in 1970s and 1980s.



Operating System

I/O Modules

Strategies for I/O:

- programmed I/O:
    - CPU tells device to perform task;
    - CPU pauses until task is complete.

Operating System

I/O Modules

Strategies for I/O:

- programmed I/O:
  - CPU tells device to perform task;
  - CPU pauses until task is complete.
  - (problem: high CPU *latency* while waiting)
- interrupt-driven I/O:
  - CPU tells device to perform task;
  - Device accesses memory directly;
  - CPU may perform other work;
  - Device *interrupts* CPU when task is complete
  - Direct memory access provided by DMA controller.

Operating System

I/O Modules

Strategies for I/O:

- programmed I/O:
    - CPU tells device to perform task;
    - CPU pauses until task is complete.
    - (problem: high CPU *latency* while waiting)
- interrupt-driven I/O:
    - CPU tells device to perform task;
    - Device accesses memory directly;
    - CPU may perform other work;
    - Device *interrupts* CPU when task is complete
    - Direct memory access provided by DMA controller.
    - (problem: potential for Bus contention)

Operating System

I/O Resource Management

Resource Management:

- I/O Devices accept coded input messages;
- inputs will only make sense if they are delivered whole;
- overlaying or interleaving requests will not work.

Operating System

I/O Resource Management

Resource Management:

- I/O Devices accept coded input messages;
- inputs will only make sense if they are delivered whole;
- overlaying or interleaving requests will not work.

Operating System:

- since Leo III (1961), multiple tasks on one computer;
- kinds of multitasking:
  - cooperative multitasking (Windows 3, Mac OS 9);
  - preemptive multitasking (Windows 95, Mac OS X).
- potential for multiple tasks to make requests of same device.

Operating System

I/O Resource Management

Resource Management:

- I/O Devices accept coded input messages;
- inputs will only make sense if they are delivered whole;
- overlaying or interleaving requests will not work.

Operating System:

- since Leo III (1961), multiple tasks on one computer;
- kinds of multitasking:
  - cooperative multitasking (Windows 3, Mac OS 9);
  - preemptive multitasking (Windows 95, Mac OS X).
- potential for multiple tasks to make requests of same device.
- OS acts as resource manager for multiple tasks.