William Stallings
Computer Organization and Architecture

Chapter 2
Computer Evolution and Performance

ENIAC - background

- Electronic Numerical Integrator And Computer.
- Eckert and Mauchly.
- University of Pennsylvania.
- Started 1943.
- Finished 1946.
  - Too late for war effort.
- Used until 1955.

ENIAC - details

- Decimal (not binary).
- 20 accumulators of 10 digits.
- Programmed manually by switches.
- 18,000 vacuum tubes.
- 30 tons.
- 15,000 square feet.
- 140 kW power consumption.
- 5,000 additions per second.

von Neumann

- Its Stored Program concept proposed 1946.
- Main memory storing programs and data.
- ALU operating on binary data.
- Control unit interpreting instructions from memory and executing.
- Input and output equipment operated by control unit.
- Princeton Institute for Advanced Studies
  - —IAS
- Completed 1952.

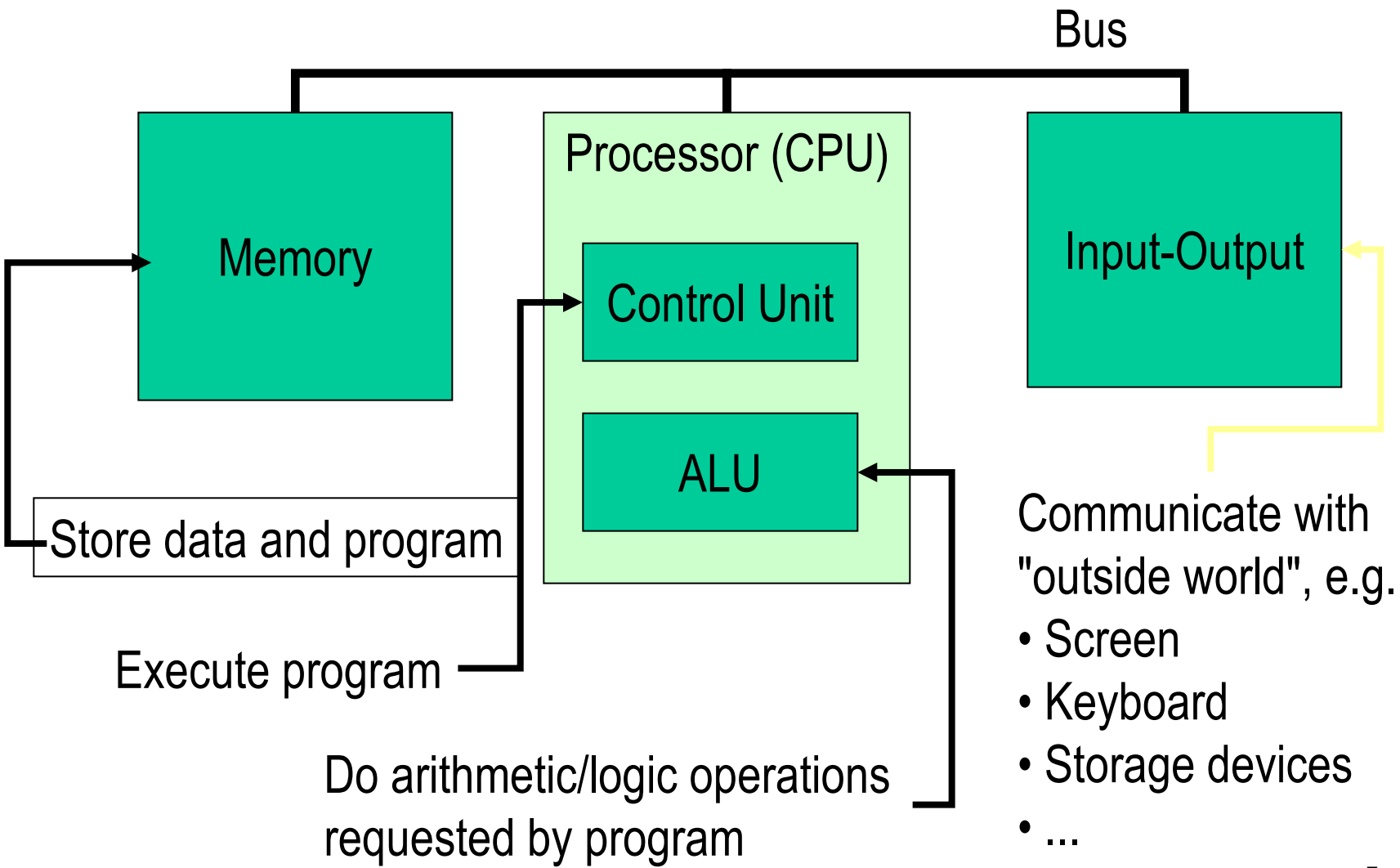
Designing Computers

- All computers more or less based on the same basic design, the Von Neumann Architecture!

The Von Neumann Architecture

- Model for designing and building computers, based on the following three characteristics:
  1) The computer consists of four main sub-systems:
     - Memory
     - ALU (Arithmetic/Logic Unit)
     - Control Unit
     - Input/Output System (I/O)
  2) Program is stored in memory during execution.
  3) Program instructions are executed sequentially.

The Von Neumann Architecture



Von Neuman Machine

- A main memory, which stored both of data and instruction.
- An arithmetic and logic unit (ALU) capable of operating on binary data.
- A control unit, interprets the instructions in memory and causes them to be executed.
- Input and Output equipment operated by the control unit

IAS - details

- Institute for advanced Study.
- 1000 x 40 bit words.
  - Binary number.
  - 2 x 20 bit instructions
- Set of registers (storage in CPU).
  - Memory Buffer Register.
  - Memory Address Register.
  - Instruction Register.
  - Instruction Buffer Register.
  - Program Counter.
  - Accumulator.

IBM

- Punched-card processing equipment.
- 1953.
  - —IBM’s first stored program computer.
  - —Scientific calculations.
- 1955.
  - —Business applications.

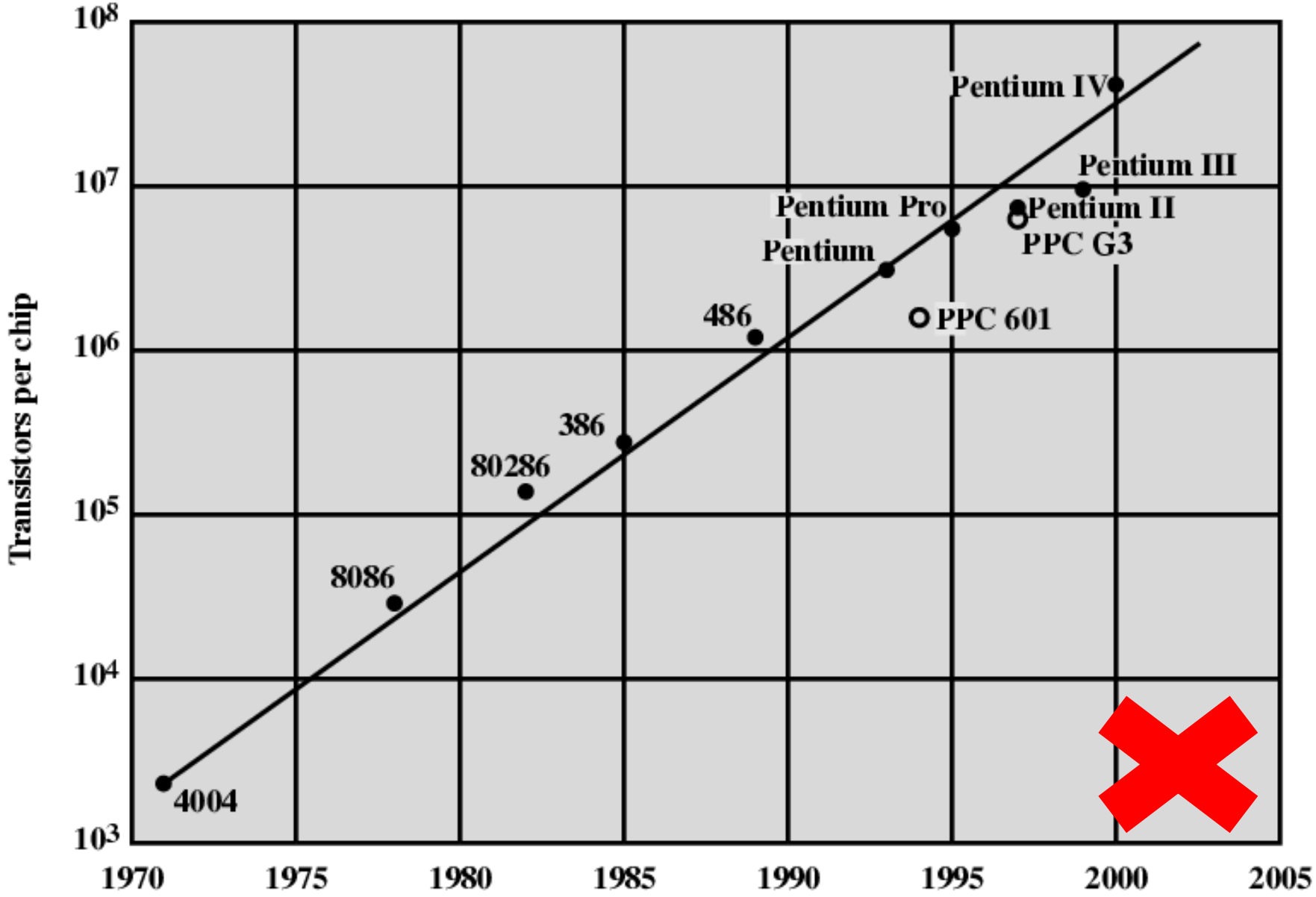
Generations of Computer

- Vacuum tube - 1946-1957
- Transistor - 1958-1964
- Small scale integration - 1965 on
  - —Up to 100 devices on a chip
- Medium scale integration - to 1971
  - —100-3,000 devices on a chip
- Large scale integration - 1971-1977
  - —3,000 - 100,000 devices on a chip
- Very large scale integration - 1978
  - —100,000 - 100,000,000 devices on a chip
- Ultra large scale integration
  - —Over 100,000,000 devices on a chip

Moore's Law

- Increased density of components on chip
- Gordon Moore - cofounder of Intel
- Number of transistors on a chip will double every year
- Since 1970's development has slowed a little
  - Number of transistors doubles every 18 months
- Cost of a chip has remained almost unchanged
- Higher packing density means shorter electrical paths, giving higher performance
- Smaller size gives increased flexibility
- Reduced power and cooling requirements
- Fewer interconnections increases reliability

Growth in CPU Transistor Count



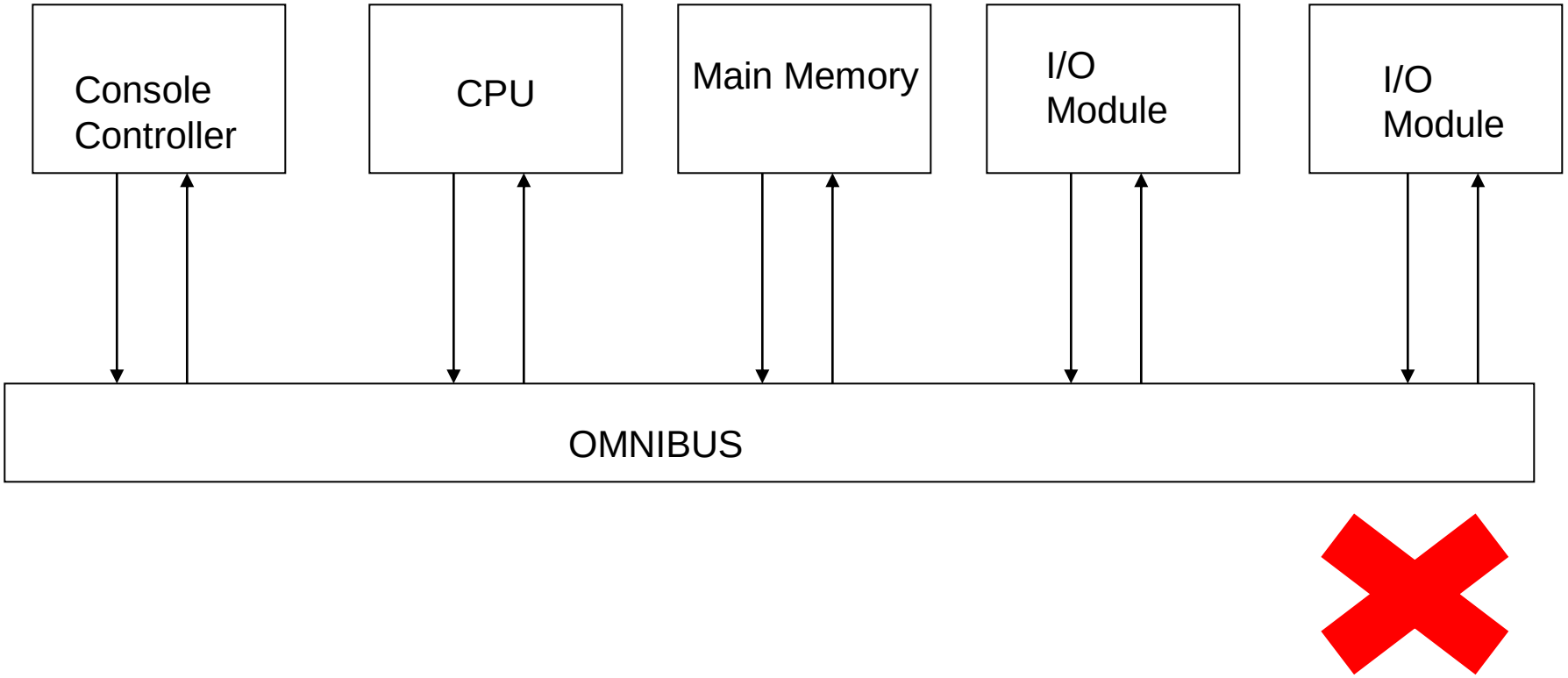
IBM 360 series

- 1964
- Replaced (& not compatible with) 7000 series
- Characteristics of First planned "family" of computers
  - Similar or identical instruction sets
  - Similar or identical O/S
  - Increasing speed
  - Increasing number of I/O ports (i.e. more terminals)
  - Increased memory size
  - Increased cost

DEC PDP-8

- 1964
- First minicomputer (after miniskirt!)
- Did not need air conditioned room
- Small enough to sit on a lab bench

DEC - PDP-8 Bus Structure



Semiconductor Memory

- 1970
- Was produced by Fairchild
- Size of a single core
  - i.e. 1 bit of magnetic core storage
- Holds 256 bits
- Non-destructive read
- Much faster than core
- Capacity approximately doubles each year

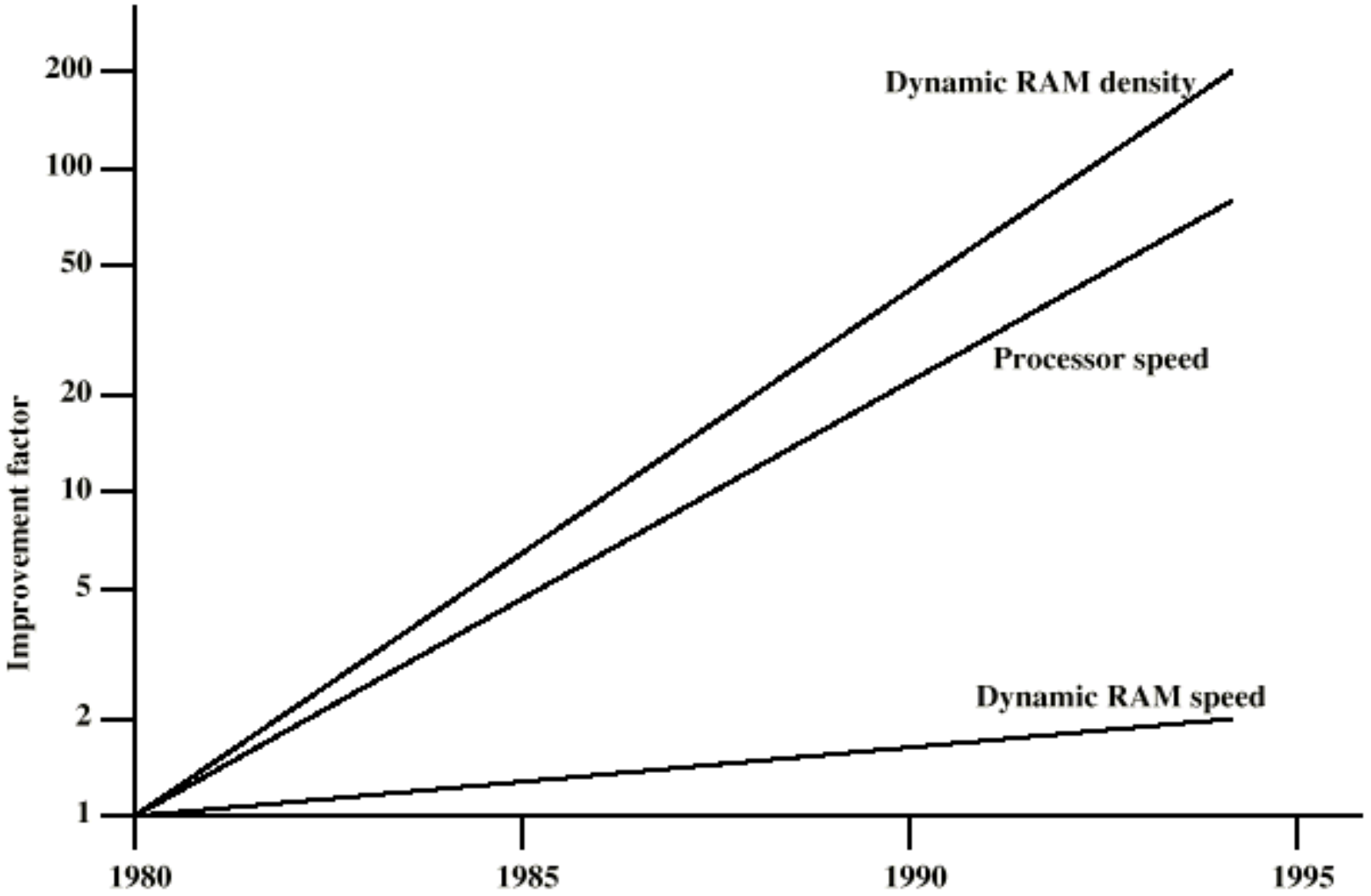
Intel

- 1971 - 4004
  - —First microprocessor
  - —All CPU components on a single chip
  - —4 bit
- Followed in 1972 by 8008
  - —8 bit = 1 byte
  - —Both designed for specific applications
- 1974 - 8080
  - —Intel’s first general purpose microprocessor

Speeding it up

- Pipelining
- On board cache
- Branch prediction
- Data flow analysis

DRAM and Processor Characteristics



Solutions

- Increase number of bits retrieved at one time
  - —Make DRAM “wider” rather than “deeper”
- Change DRAM interface
  - —Cache
- Reduce frequency of memory access
  - —More complex cache and cache on chip
- Increase interconnection bandwidth
  - —High speed buses

Pentium Evolution (1)

- 8080
  - first general purpose microprocessor
  - 8 bit data path
  - Used in first personal computer – Altair
- 8086
  - much more powerful
  - 16 bit
  - instruction cache, prefetch few instructions
  - 8088 (8 bit external bus) used in first IBM PC
- 80286
  - 16 Mbyte memory addressable
  - up from 1Mb
- 80386
  - 32 bit
  - Support for multitasking

الاستعمال
الحجم
الاسم
الإصدار

Pentium Evolution (2)

- 80486
  - —sophisticated powerful cache and instruction pipelining
  - —built in maths co-processor
- Pentium
  - —Superscalar
  - —Multiple instructions executed in parallel
- Pentium Pro
  - —Increased superscalar organization
  - —Aggressive register renaming
  - —branch prediction
  - —data flow analysis
  - —speculative execution

Pentium Evolution (3)

- Pentium II
  - —MMX technology
  - —graphics, video & audio processing
- Pentium III
  - —Additional floating point instructions for 3D graphics
- Pentium 4
  - —Note Arabic rather than Roman numerals
  - —Further floating point and multimedia enhancements
- Itanium
  - —64 bit
  - —see chapter 15
- See Intel web pages for detailed information on processors

Memory Subsystem

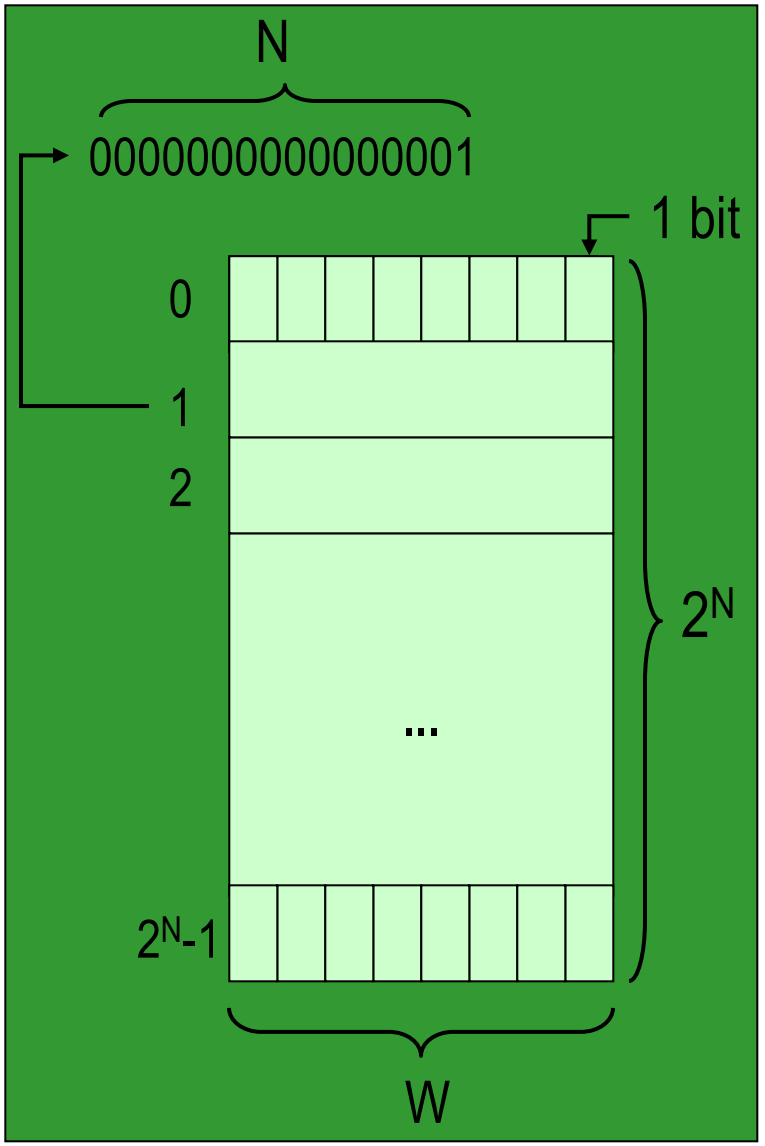
- Memory, also called RAM (Random Access Memory),
  - Consists of many memory cells (storage units) of a fixed size.
    Each cell has an address associated with it: 0, 1, …
  - All accesses to memory are to a specified address.
    A cell of RAM: is the minimum unit of access (fetch/store a complete cell).
  - The time it takes to fetch/store a cell is the same for all cells.

- When the computer is running, both
  - Program
  - Data (variables)

  are stored in the memory.

RAM

- Need to distinguish between
  - the address of a memory cell and the content of a memory cell
- Memory width (W):
  - How many bits is each memory cell, typically one byte (=8 bits)
- Address width (N):
  - How many bits used to represent each address, determines the maximum memory size = address space
  - If address width is N-bits, then address space is 2^N (0,1,...,2^N-1)



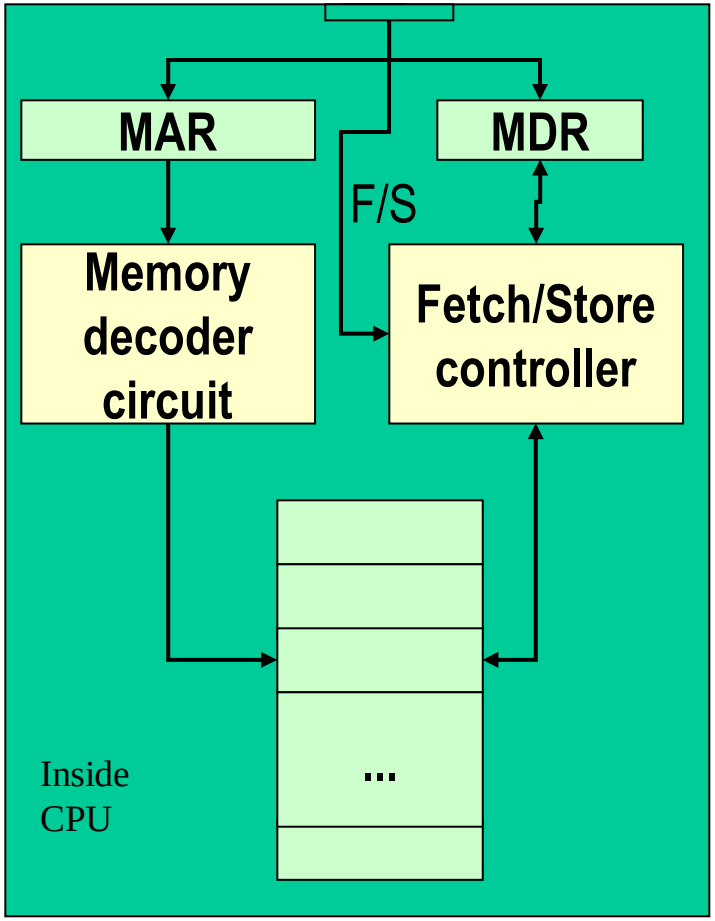
Memory Size / Speed

- Typical memory in a personal computer (PC):
  - 64MB - 256MB
- Memory sizes:
  - Kilobyte (KB)= 2^{10} = 1,024 bytes ~ 1 thousand
  - Megabyte(MB) = 2^{20} = 1,048,576 bytes ~ 1 million
  - Gigabyte(GB) = 2^{30} = 1,073,741,824 bytes ~ 1 billion
- Memory Access Time (read from/ write to memory)
  - 50-75 nanoseconds (1 nsec. = 0.000000001 sec.)
- RAM is
  - volatile (can only store when power is on)
  - relatively expensive

Operations on Memory

- Fetch (address):
  - Fetch a copy of the content of memory cell with the specified address.
  - Non-destructive, copies value in memory cell.
- Store (address, value):
  - Store the specified value into the memory cell specified by address.
  - Destructive, overwrites the previous value of the memory cell.
- The memory system is interfaced via:
  - Memory Address Register (MAR)
  - Memory Data Register (MDR)
  - Fetch/Store signal

Structure of the Memory Subsystem



- Fetch(address)
  - Load address into MAR.
  - Decode the address in MAR.
  - Copy the content of memory cell with specified address into MDR.
- Store(address, value)
  - Load the address into MAR.
  - Load the value into MDR.
  - Decode the address in MAR
  - Copy the content of MDR into memory cell with the specified address. (MAR الذي اخده من)

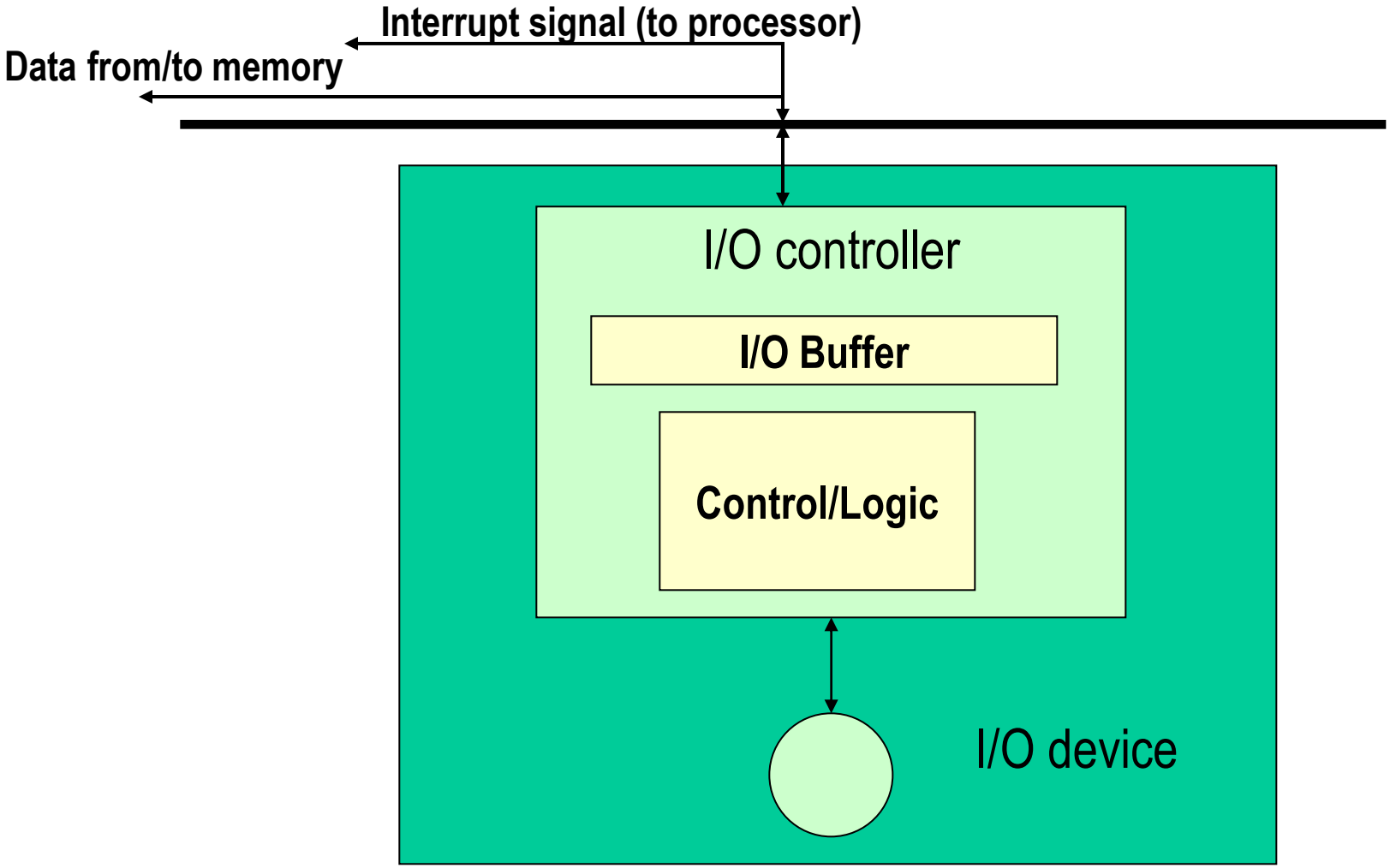
Input/Output Subsystem

- Handles devices that allow the computer system to:
  - Communicate and interact with the outside world
    - Screen, keyboard, printer, ...
  - Store information (mass-storage)
    - Hard-drives, floppies, CD, tapes, …
- Mass-Storage Device Access Methods:
  - Direct Access Storage Devices (DASDs)
    - Hard-drives, floppy-disks, CD-ROMs, ...
  - Sequential Access Storage Devices (SASDs)
    - Tapes (for example, used as backup devices)

I/O Controllers

- Speed of I/O devices is slow compared to RAM
  - RAM ~ 50 nsec.
  - Hard-Drive ~ 10msec. = (10,000,000 nsec)
- Solution:
  - I/O Controller, a special purpose processor:
    - Has a small memory buffer, and a control logic to control I/O device (e.g. move disk arm).
    - Sends an interrupt signal to CPU when done read/write.
  - Data transferred between RAM and memory buffer.
  - Processor free to do something else while I/O controller reads/writes data from/to device into I/O buffer.

Structure of the I/O Subsystem

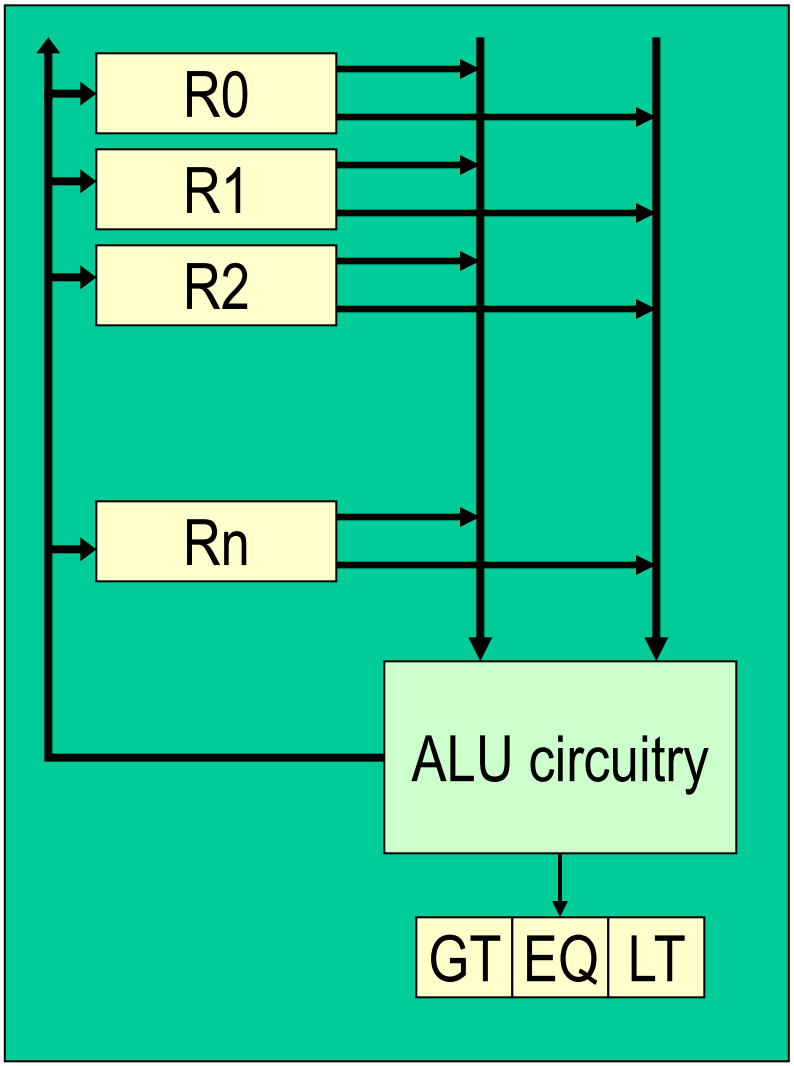


The ALU Subsystem

- The ALU (Arithmetic/Logic Unit) performs
  - mathematical operations (+, -, x, /, …)
  - logic operations (=, <, >, and, or, not, ...)
- In today's computers integrated into the CPU
- Consists of:
  - Circuits to do the arithmetic/logic operations.
  - Registers (fast storage units) to store intermediate computational results.
  - Bus that connects the two.

Structure of the ALU

- Registers:
  - Very fast local memory cells, that store operands of operations and intermediate results.
  - CCR (condition code register), a special purpose register that stores the result of <, = , > operations
- ALU circuitry:
  - Contains an array of circuits to do mathematical/logic operations.
- Bus:
  - Data path interconnecting the registers to the ALU circuitry.



The Control Unit

- Program is stored in memory
  - as machine language instructions, in binary
- The task of the control unit is to execute programs by repeatedly:
  - Fetch from memory the next instruction to be executed.
  - Decode it, that is, determine what is to be done.
  - Execute it by issuing the appropriate signals to the ALU, memory, and I/O subsystems.
  - Continues until the HALT instruction

Instruction Set Design

- Two different approaches:
  - Reduced Instruction Set Computers (RISC)
    - Instruction set as **small** and **simple** as possible.
    - Minimizes amount of circuitry --> faster computers
  - Complex Instruction Set Computers (CISC)
    - More instructions, very complex
    - Each instruction can do more work, but require more circuitry.

Typical Machine Instructions

- Notation:
  - We use X, Y, Z to denote RAM cells
  - Assume only one register R (for simplicity)
  - Use English-like descriptions (should be binary)
- Data Transfer Instructions
  - LOAD X — Load content of memory location X to R
  - STORE X — Load content of R to memory location X
  - MOVE X, Y — Copy content of memory location X to loc. Y

Example

- Pseudo-code: Set A to B + C
- Assuming variable:
  - A stored in memory cell 100, B stored in memory cell 150, C stored in memory cell 151
- Machine language (really in binary)

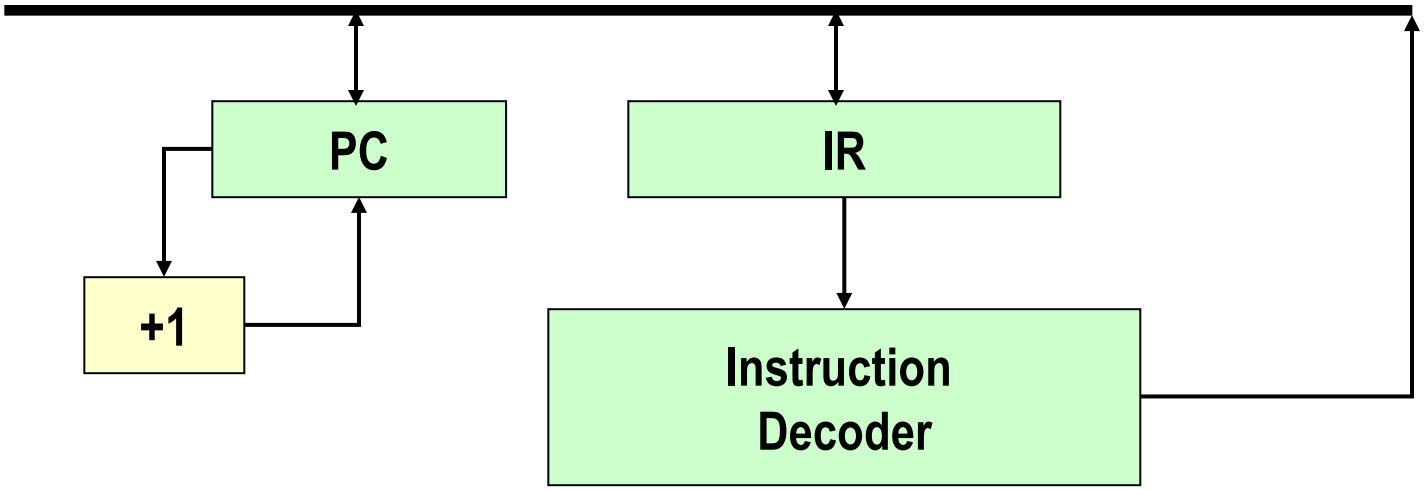
sisc
  - LOAD 150
  - ADD 151
  - STORE 100

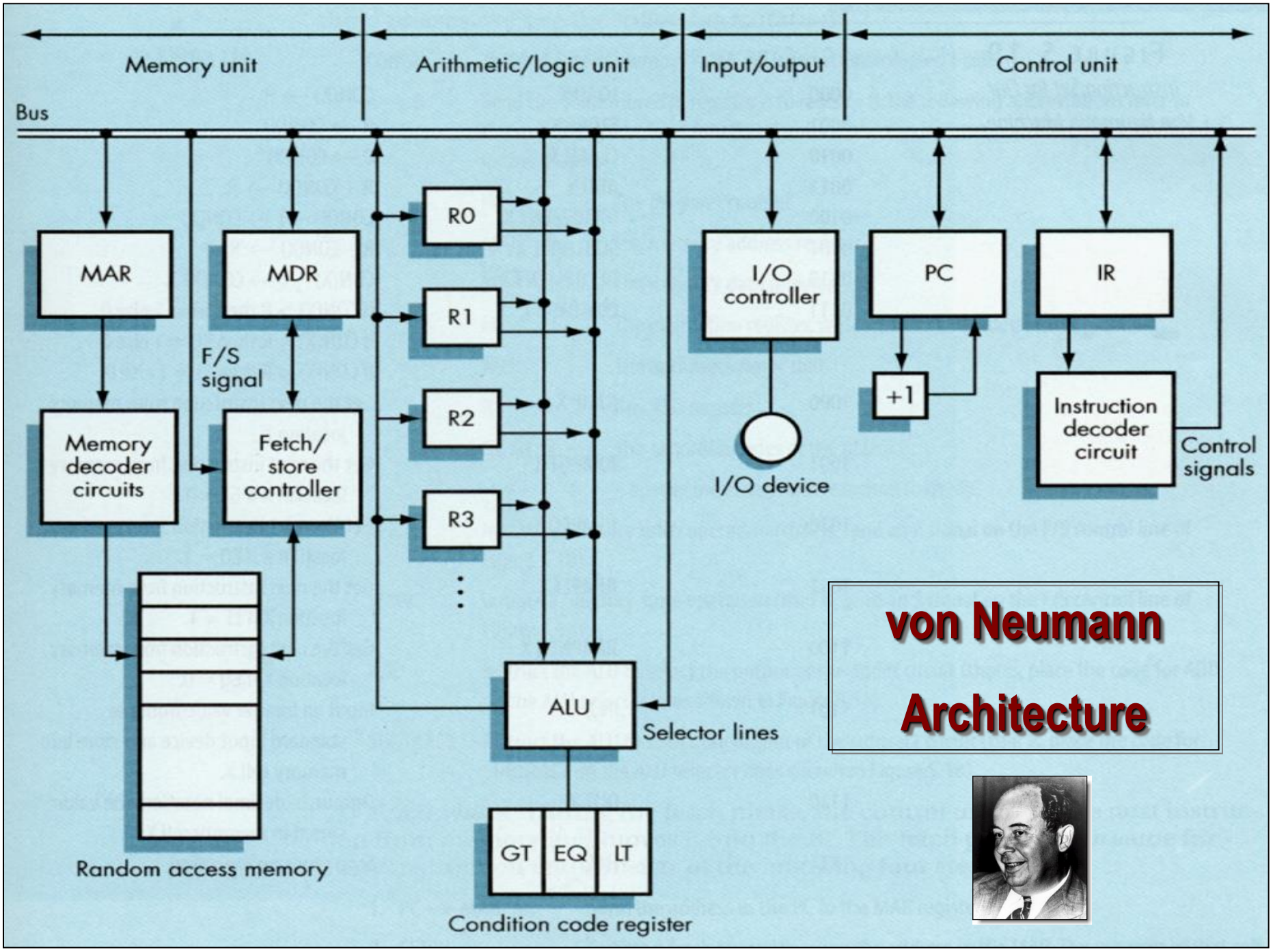
  - or

risc
  - (ADD 150, 151, 100)

Structure of the Control Unit

- PC (Program Counter):
  - stores the address of next instruction to fetch
- IR (Instruction Register):
  - stores the instruction fetched from memory
- Instruction Decoder:
  - Decodes instruction and activates necessary circuitry




Memory unit
Arithmetic/logic unit
Input/output
Control unit
Bus
MAR
MDR
F/S signal
R0
R1
R2
R3
I/O controller
PC
IR
+1
Instruction decoder circuit
Control signals
I/O device
Memory decoder circuits
Fetch/ store controller
ALU
Selector lines
GT
EQ
LT
Condition code register
Random access memory
von Neumann Architecture

How does this all work together?

- Program Execution:
  - PC is set to the address“ PC counter” where the first program instruction is stored in memory.
  - Repeat until HALT instruction or fatal error

    Fetch instruction

    Decode instruction

    Execute instruction

    End of loop