

# **Organisasi dan Arsitektur Komputer**

## **Pertemuan 1: Pengenalan Sistem Komputer**

# Architecture & Organization

- Architecture is those attributes visible to the programmer
  - Instruction set, number of bits used for data representation, I/O mechanisms, addressing techniques.
  - e.g. Is there a multiply instruction?
- Organization is how features are implemented
  - Control signals, interfaces, memory technology.
  - e.g. Is there a hardware multiply unit or is it done by repeated addition?

## Architecture & Organization 2

- All Intel x86 family share the same basic architecture
- The IBM System/370 family share the same basic architecture
- This gives code compatibility ... at least backwards
- Organization differs within members of the same family, e.g. floating point numerical co-processors with names like 8087, 80287 and 80387. With very few exceptions, the 80486 and subsequent x86 processors then integrated this x87 functionality on chip.

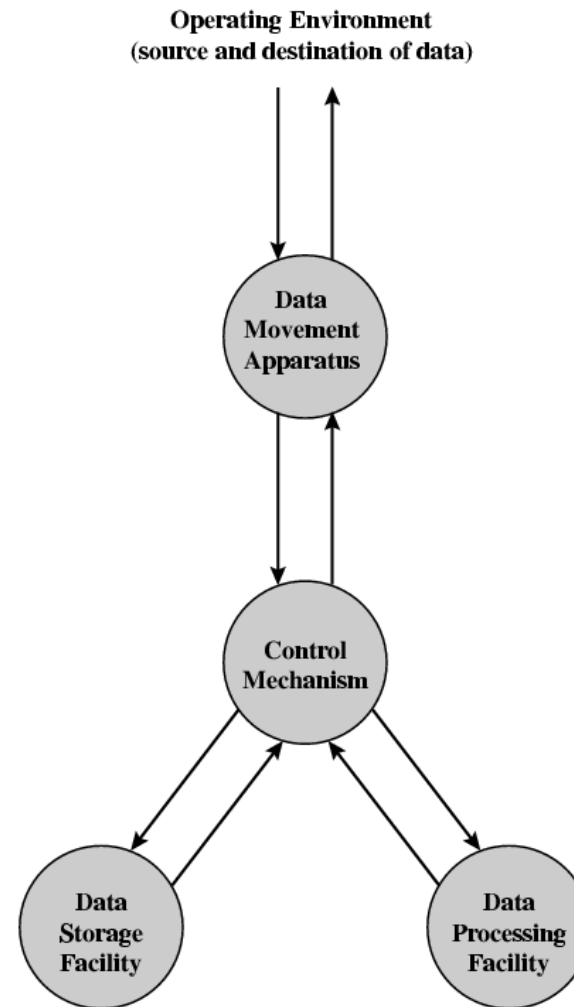
# **Structure & Function**

- Structure is the way in which components relate to each other
- Function is the operation of individual components as part of the structure

# Function

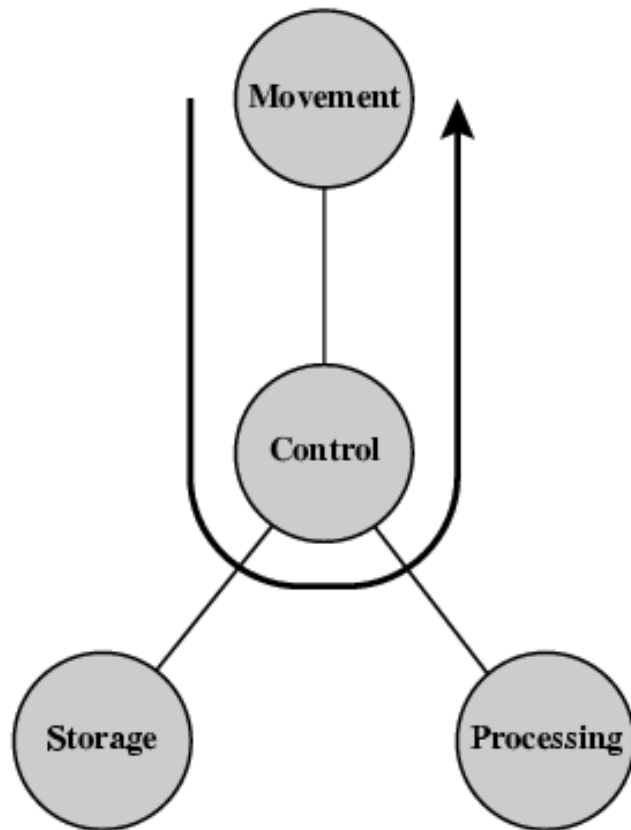
## General computer functions:

- Data processing
- Data storage
- Data movement
- Control



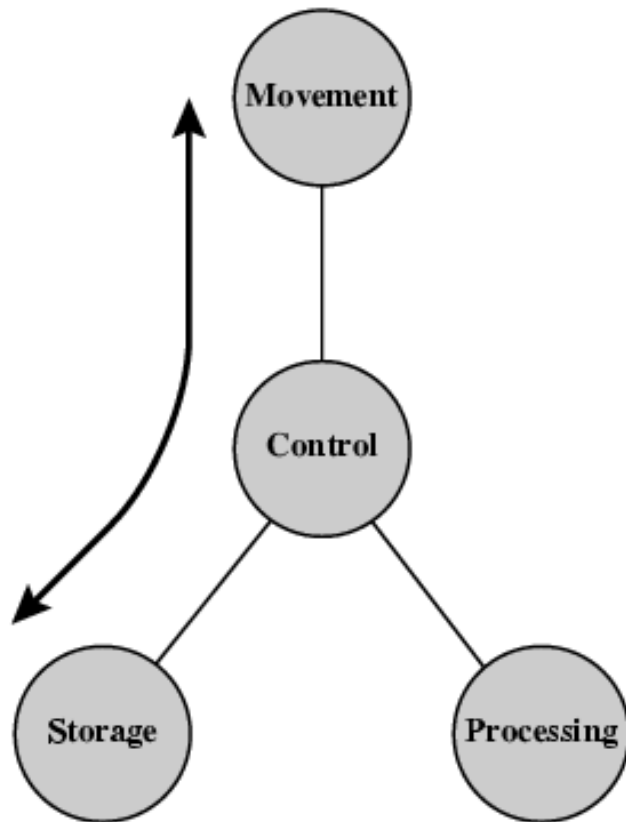
# Operations (a) Data movement

I/O (peripherals directly attached)  
Communications/Networking  
(communication lines)



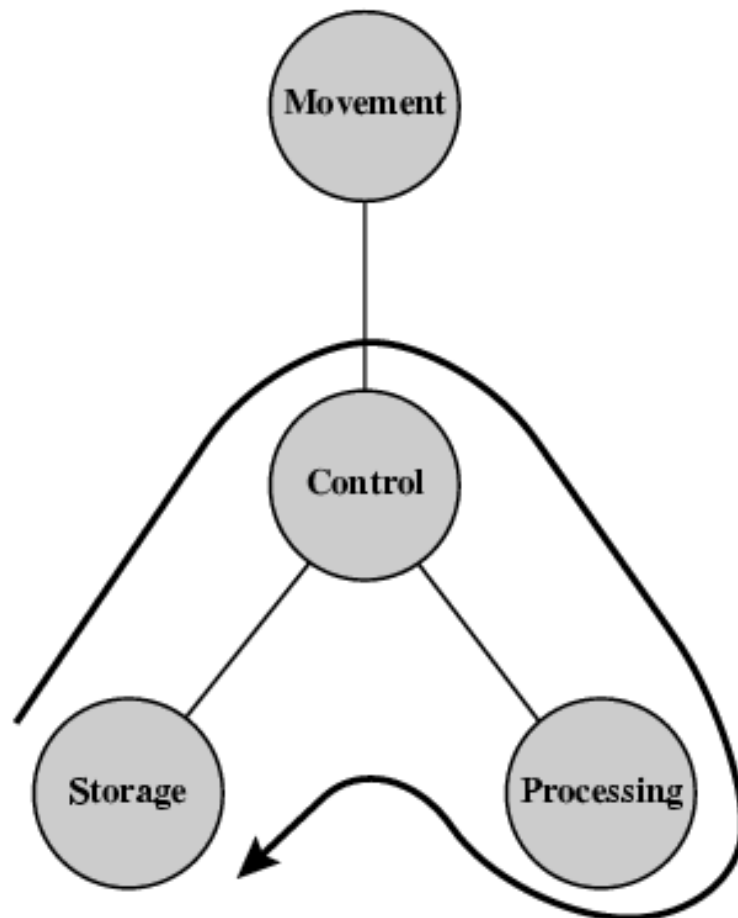
Camera attached to a PC,  
sending the frames to a  
window on the screen of the  
same PC.

## Operations (b) Storage



Playing an mp3 file stored in memory to earphones attached to the same PC.

## Operation (c) Processing from/to storage

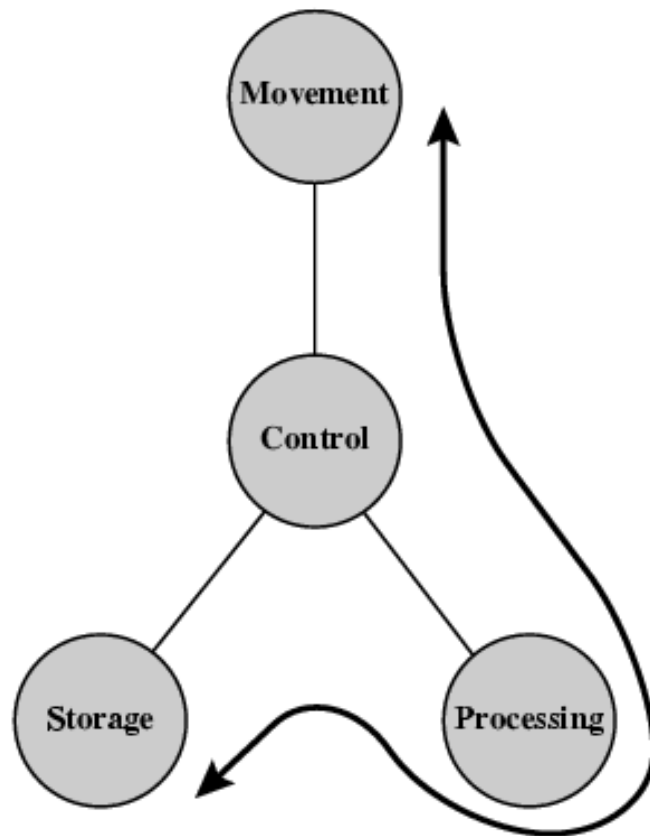


Any number-crunching application that takes data from memory and stores the result back in memory.



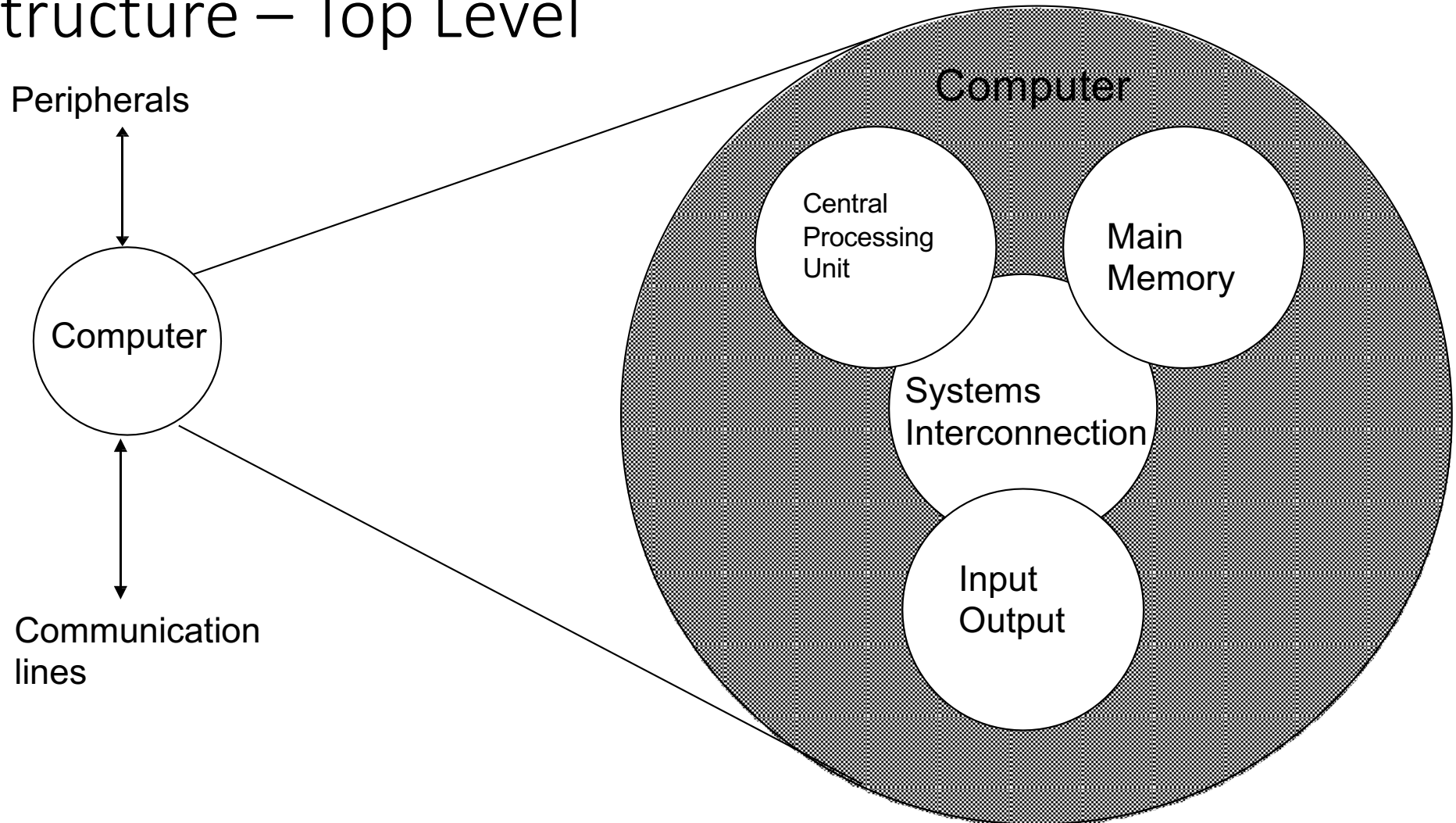
Operation (d)

Processing from storage to I/O

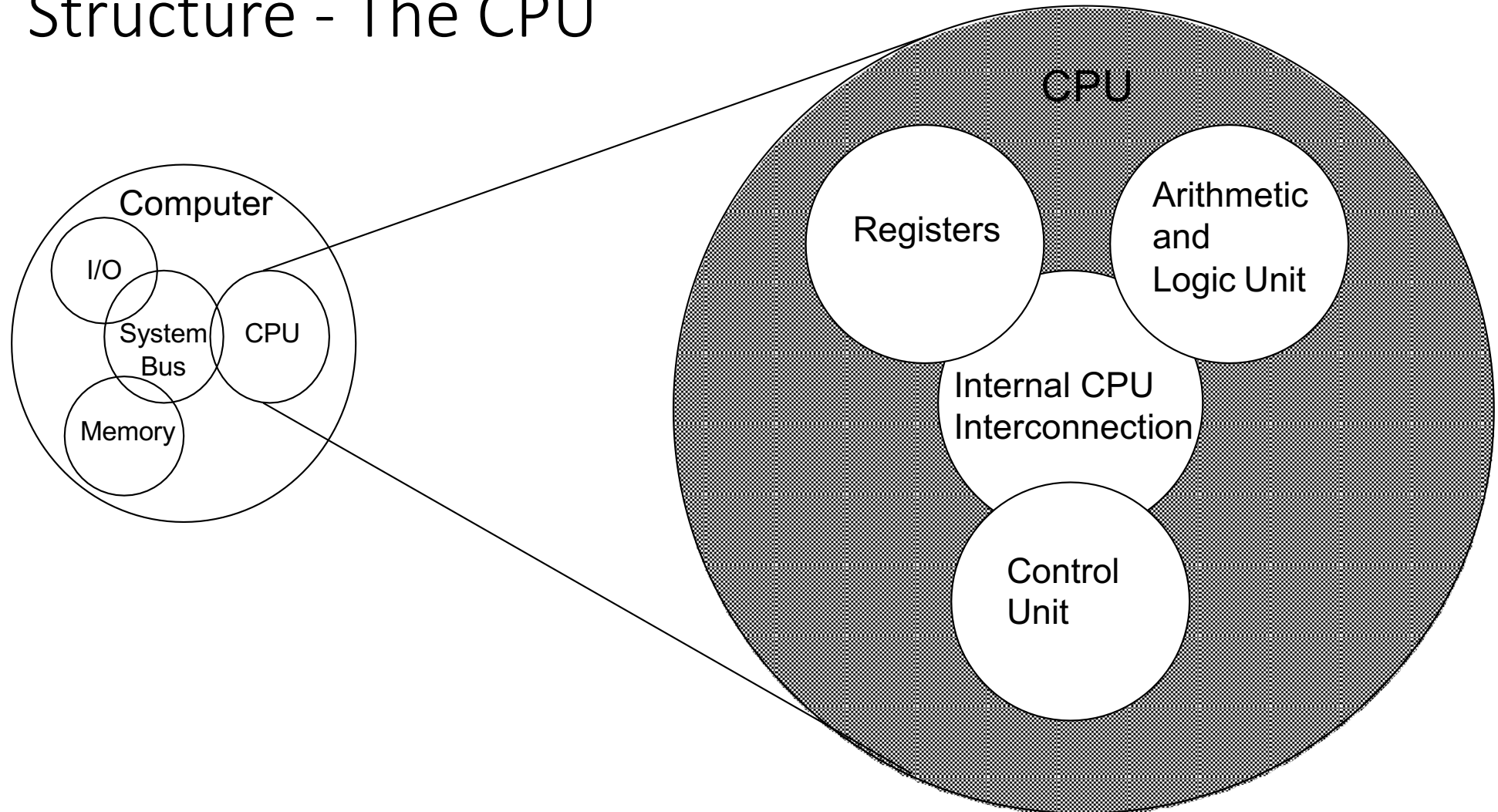


Receiving packets over a network interface, verifying their CRC, then storing them in memory.

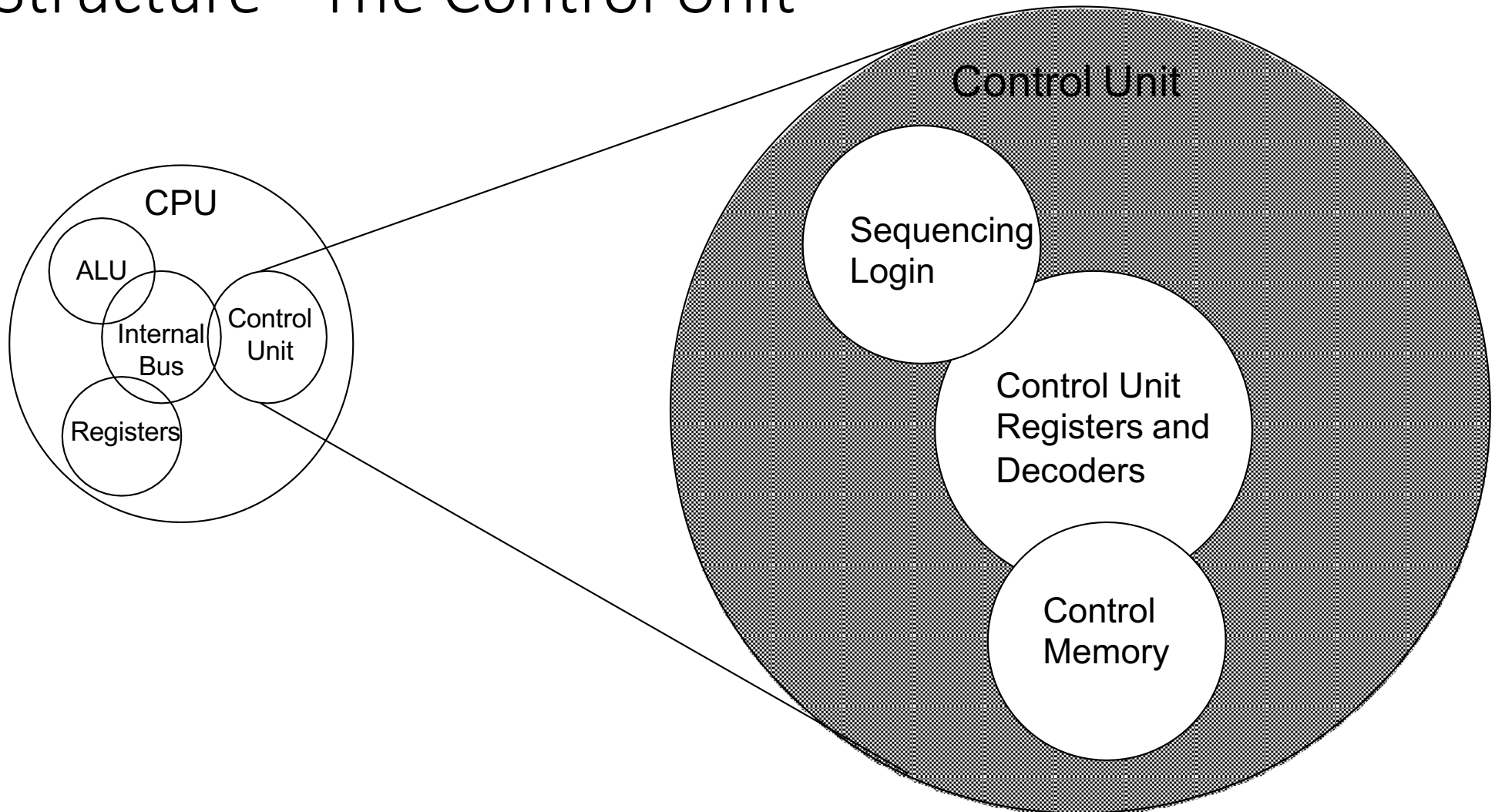
# Structure – Top Level



# Structure - The CPU



# Structure - The Control Unit



# ENIAC - background

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

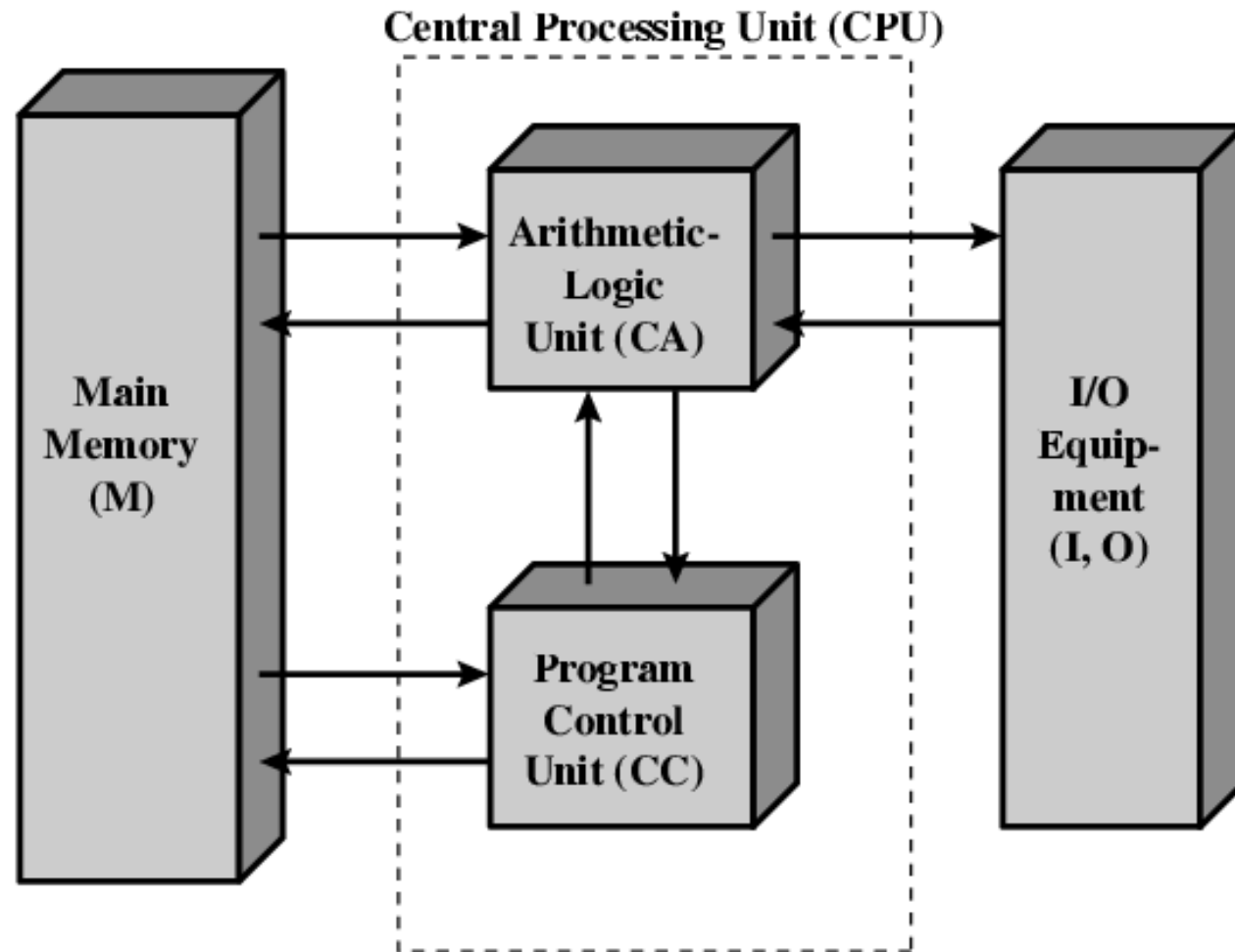
# ENIAC - details

- 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/Turing

- Stored Program concept
- 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

# Structure of von Neumann machine

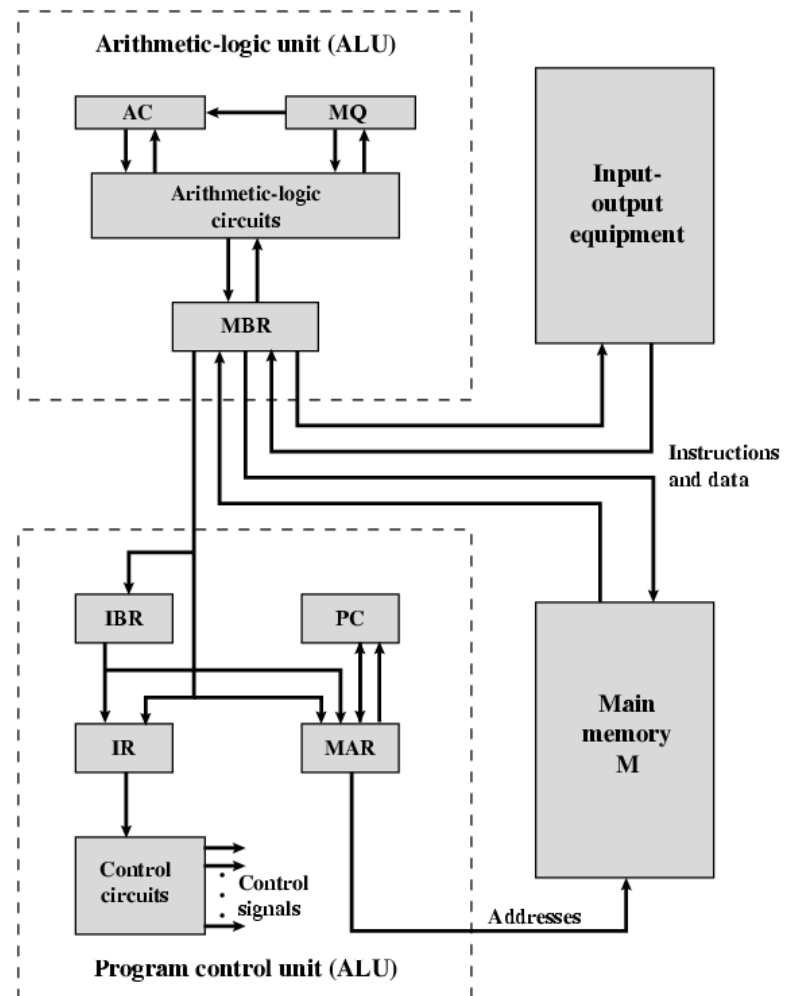




# IAS - details

- 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
  - Multiplier Quotient

# Structure of IAS – detail



# Commercial Computers

- 1947 - Eckert-Mauchly Computer Corporation
- UNIVAC I (Universal Automatic Computer)
- US Bureau of Census 1950 calculations
- Became part of Sperry-Rand Corporation
- Late 1950s - UNIVAC II
  - Faster
  - More memory

# IBM

- Punched-card processing equipment
- 1953 - the 701
  - IBM's first stored program computer
  - Scientific calculations
- 1955 - the 702
  - Business applications
- Lead to 700/7000 series

# Transistors

- Replaced vacuum tubes
- Smaller
- Cheaper
- Less heat dissipation
- Solid State device
- Made from Silicon (Sand)
- Invented 1947 at Bell Labs
- William Shockley et al.

# Transistor Based Computers

- Second generation machines
- NCR & RCA produced small transistor machines
- IBM 7000
- DEC - 1957
  - Produced PDP-1

# Microelectronics

- Literally - “small electronics”
- A computer is made up of gates, memory cells and interconnections
- These can be manufactured on a semiconductor
- e.g. silicon wafer

# Generations of Computer

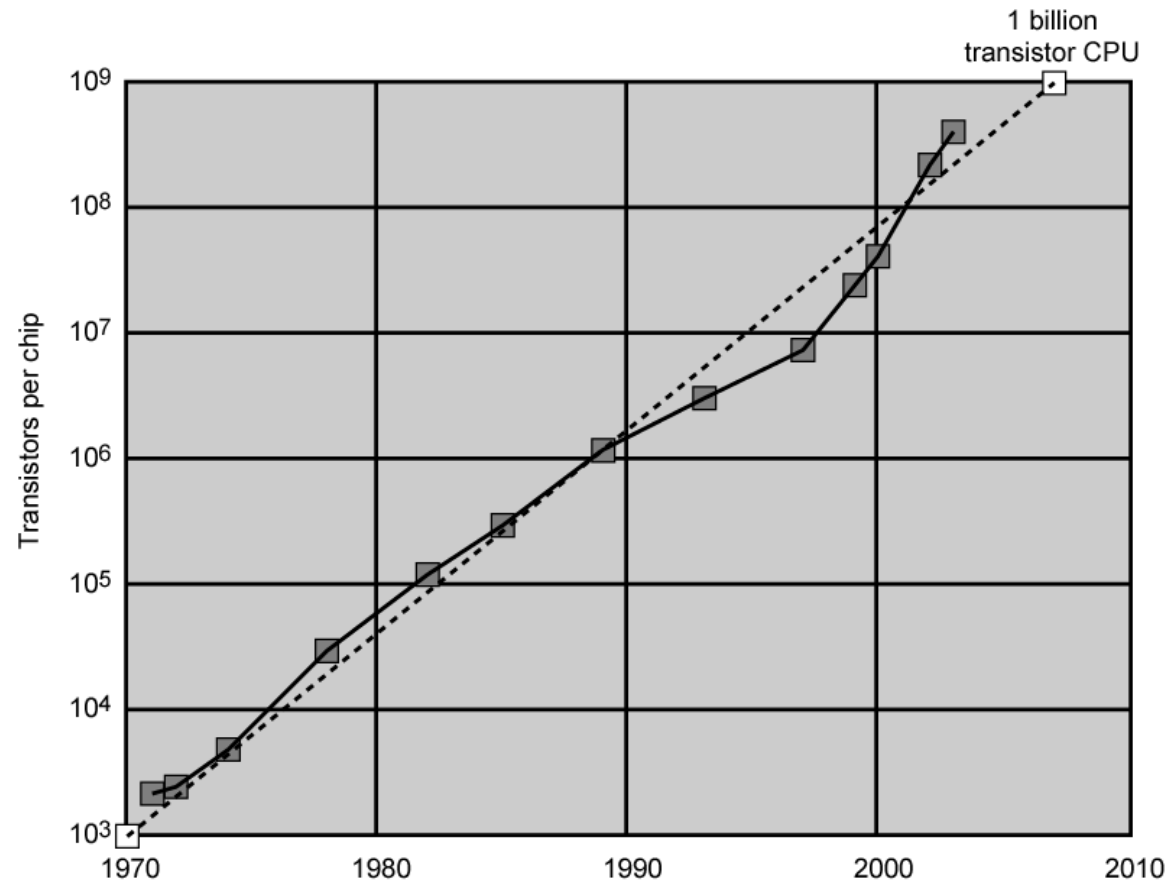
- Vacuum tube - 1946-1957
- Transistor - 1958-1964
- Small scale integration - 1965 on
  - Up to 100 devices on a chip
- Medium scale integration - to 1971
  - 100-3,000 devices on a chip
- Large scale integration - 1971-1977
  - 3,000 - 100,000 devices on a chip
- Very large scale integration - 1978 -1991
  - 100,000 - 100,000,000 devices on a chip
- Ultra large scale integration – 1991 -
  - Over 100,000,000 devices on a chip



# Moore's Law

- Increased density of components on chip
- Gordon Moore – co-founder 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



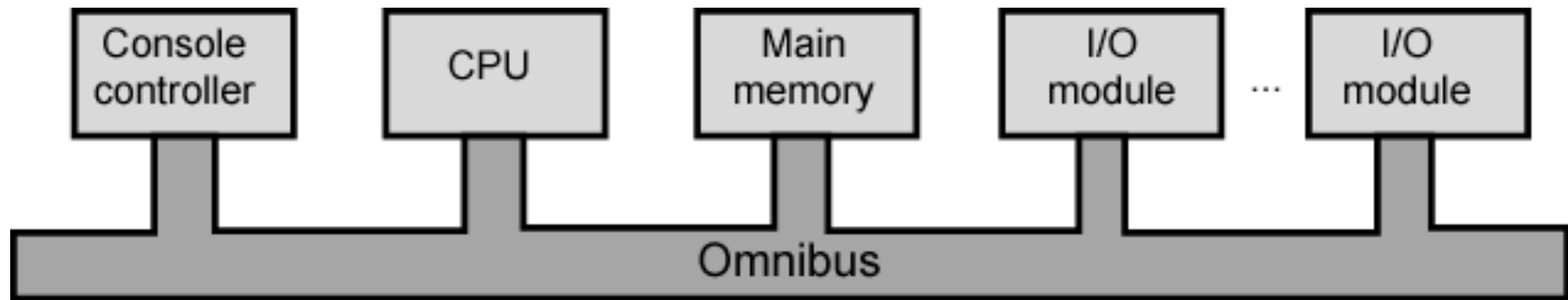
# IBM 360 series

- 1964
- Replaced (& not compatible with) 7000 series
- 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
- Multiplexed switch structure

# DEC PDP-8

- 1964
- First minicomputer (after miniskirt!)
- Did not need air conditioned room
- Small enough to sit on a lab bench
- \$16,000
  - \$100k+ for IBM 360
- Embedded applications & OEM
- BUS STRUCTURE

# DEC - PDP-8 Bus Structure



# Semiconductor Memory

- 1970
- 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
  - Both designed for specific applications
- 1974 - 8080
  - Intel's first general purpose microprocessor

# Speeding it up

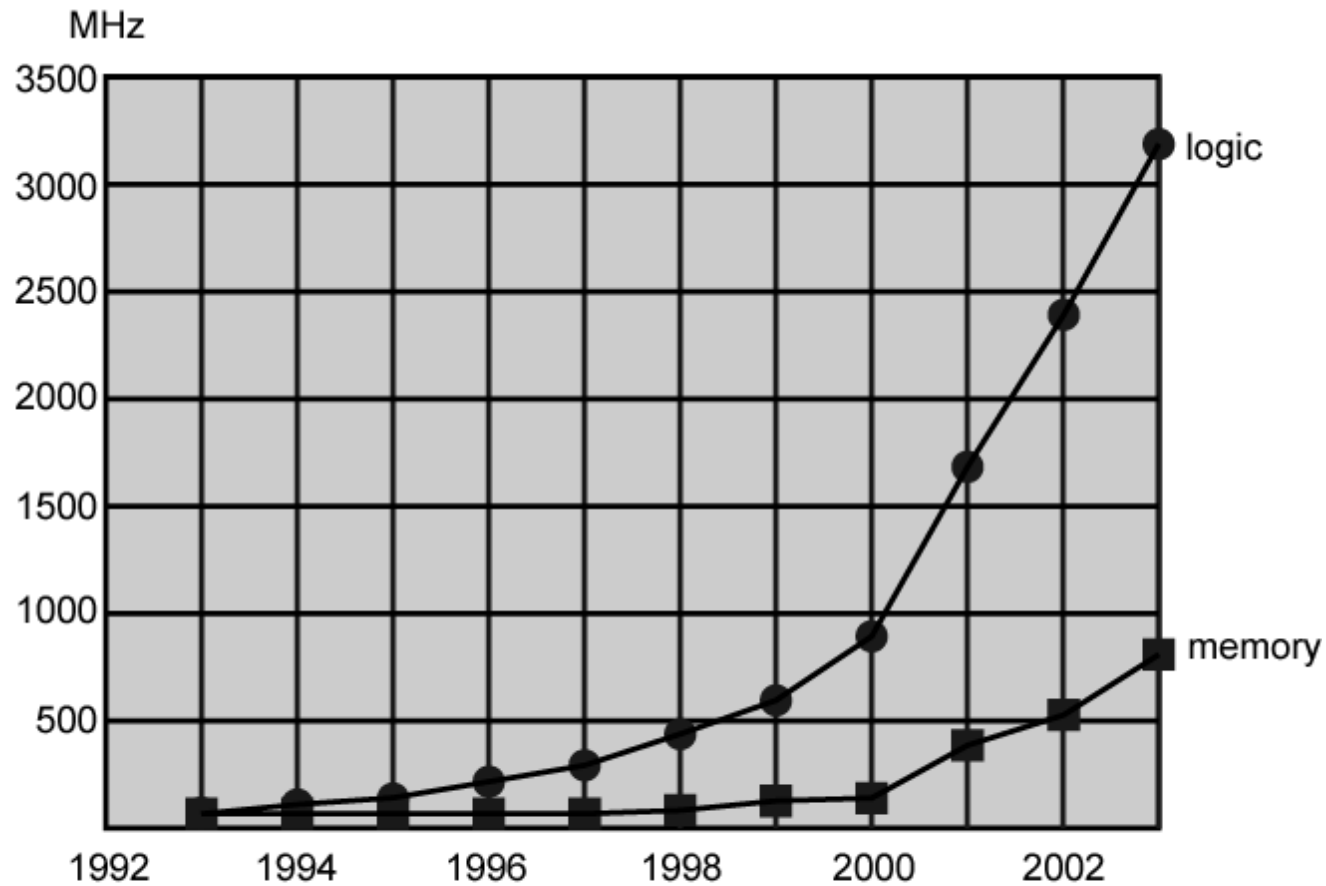
- Pipelining
- On board cache
- On board L1 & L2 cache
- Branch prediction
- Data flow analysis
- Speculative execution



# Performance Balance

- Processor speed increased
- Memory capacity increased
- Memory speed lags behind processor speed

# Logic and Memory Performance Gap



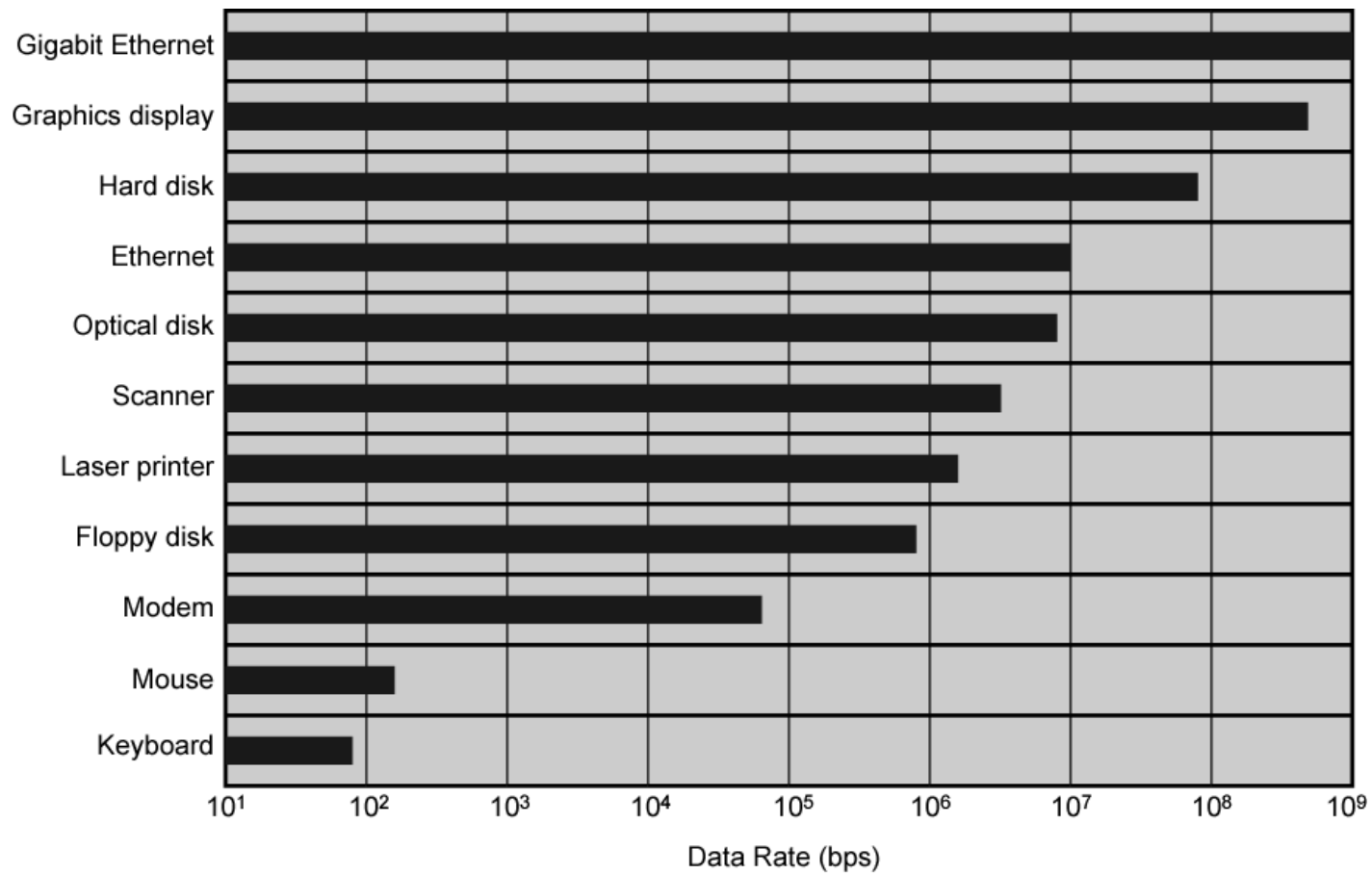
# Solutions

- 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
  - Hierarchy of buses

# I/O Devices

- Peripherals with intensive I/O demands
- Large data throughput demands
- Processors can handle this
- Problem moving data
- Solutions:
  - Caching
  - Buffering
  - Higher-speed interconnection buses
  - More elaborate bus structures
  - Multiple-processor configurations

# Typical I/O Device Data Rates



# Key is Balance

- Processor components
- Main memory
- I/O devices
- Interconnection structures

## Improvements in Chip Organization and Architecture

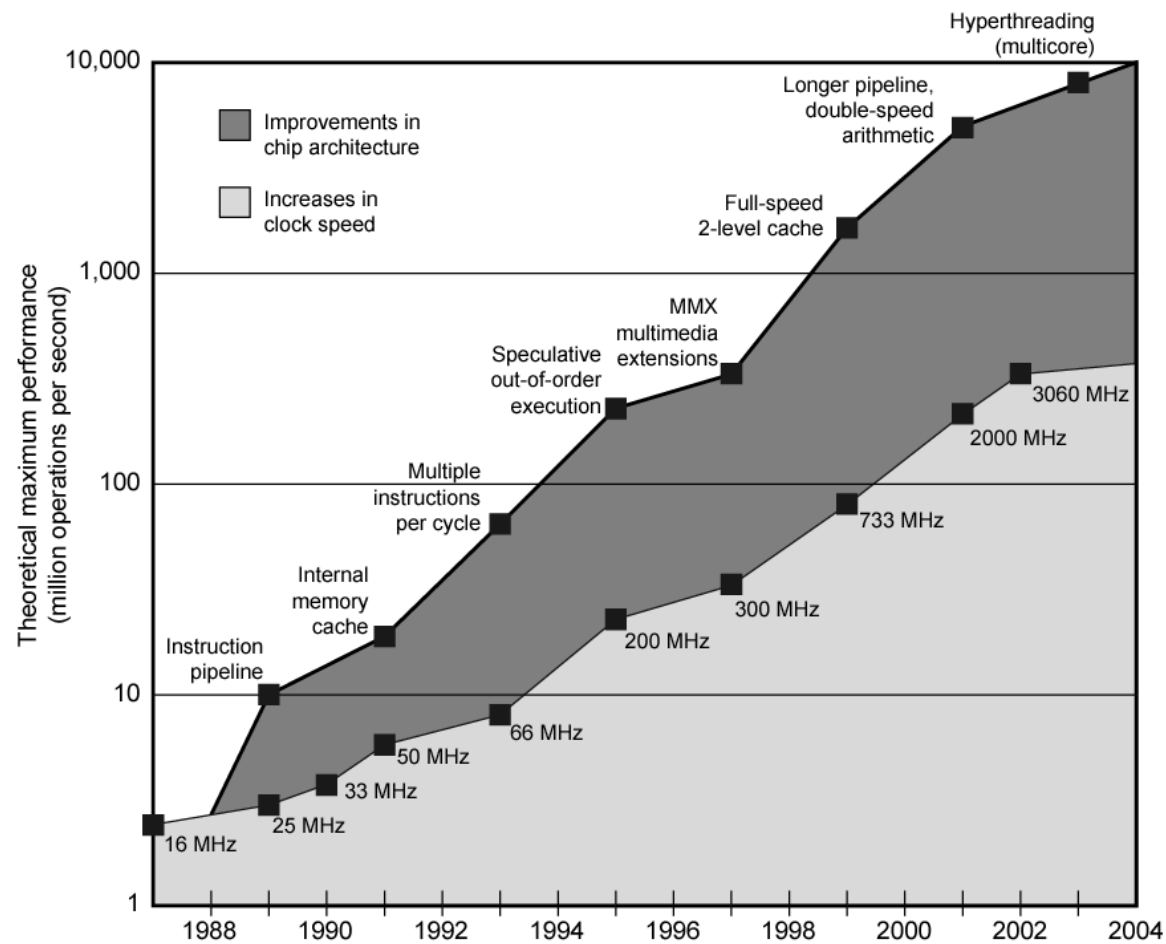
- Increase hardware speed of processor
  - Fundamentally due to shrinking logic gate size
    - More gates, packed more tightly, increasing clock rate
    - Propagation time for signals reduced
- Increase size and speed of caches
  - Dedicating part of processor chip
    - Cache access times drop significantly
- Change processor organization and architecture
  - Increase effective speed of execution
  - Parallelism

# Problems with Clock Speed and Logic Density

- Power
  - Power density increases with density of logic and clock speed
  - Dissipating heat
- RC delay
  - Speed at which electrons flow limited by resistance and capacitance of metal wires connecting them
  - Delay increases as RC product increases
  - Wire interconnects thinner, increasing resistance
  - Wires closer together, increasing capacitance
- Memory latency
  - Memory speeds lag processor speeds
- Solution:
  - More emphasis on organizational and architectural approaches



# Intel Microprocessor Performance



# Increased Cache Capacity

- Typically two or three levels of cache between processor and main memory
- Chip density increased
  - More cache memory on chip
    - Faster cache access
- Pentium chip devoted about 10% of chip area to cache
- Pentium 4 devotes about 50%

# More Complex Execution Logic

- Enable parallel execution of instructions
- Pipeline works like assembly line
  - Different stages of execution of different instructions at same time along pipeline
- Superscalar allows multiple pipelines within single processor
  - Instructions that do not depend on one another can be executed in parallel

# Diminishing Returns

- Internal organization of processors complex
  - Can get a great deal of parallelism
  - Further significant increases likely to be relatively modest
- Benefits from cache are reaching limit
- Increasing clock rate runs into power dissipation problem
  - Some fundamental physical limits are being reached

# New Approach – Multiple Cores

- Multiple processors on single chip
  - Large shared cache
- Within a processor, increase in performance proportional to square root of increase in complexity
- If software can use multiple processors, doubling number of processors almost doubles performance
- So, use two simpler processors on the chip rather than one more complex processor
- With two processors, larger caches are justified
  - Power consumption of memory logic less than processing logic

# x86 Evolution (1)

- 8080
  - first general purpose microprocessor
  - 8 bit data path
  - Used in first personal computer – Altair
- 8086 – 5MHz – 29,000 transistors
  - 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
- 80486
  - sophisticated powerful cache and instruction pipelining
  - built in maths co-processor

# x86 Evolution (2)

- Pentium
  - Superscalar
  - Multiple instructions executed in parallel
- Pentium Pro
  - Increased superscalar organization
  - Aggressive register renaming
  - branch prediction
  - data flow analysis
  - speculative execution
- Pentium II
  - MMX technology
  - graphics, video & audio processing
- Pentium III
  - Additional floating point instructions for 3D graphics

# x86 Evolution (3)

- Pentium 4
  - Note Arabic rather than Roman numerals
  - Further floating point and multimedia enhancements
- Core
  - First x86 with dual core
- Core 2
  - 64 bit architecture
- Core 2 Quad – 3GHz – 820 million transistors
  - Four processors on chip



# x86 Evolution (4)

- x86 architecture dominant outside embedded systems
- Organization and technology changed dramatically
- Instruction set architecture evolved with backwards compatibility
- ~1 instruction per month added
- 500 instructions available
- See Intel web pages for detailed information on processors

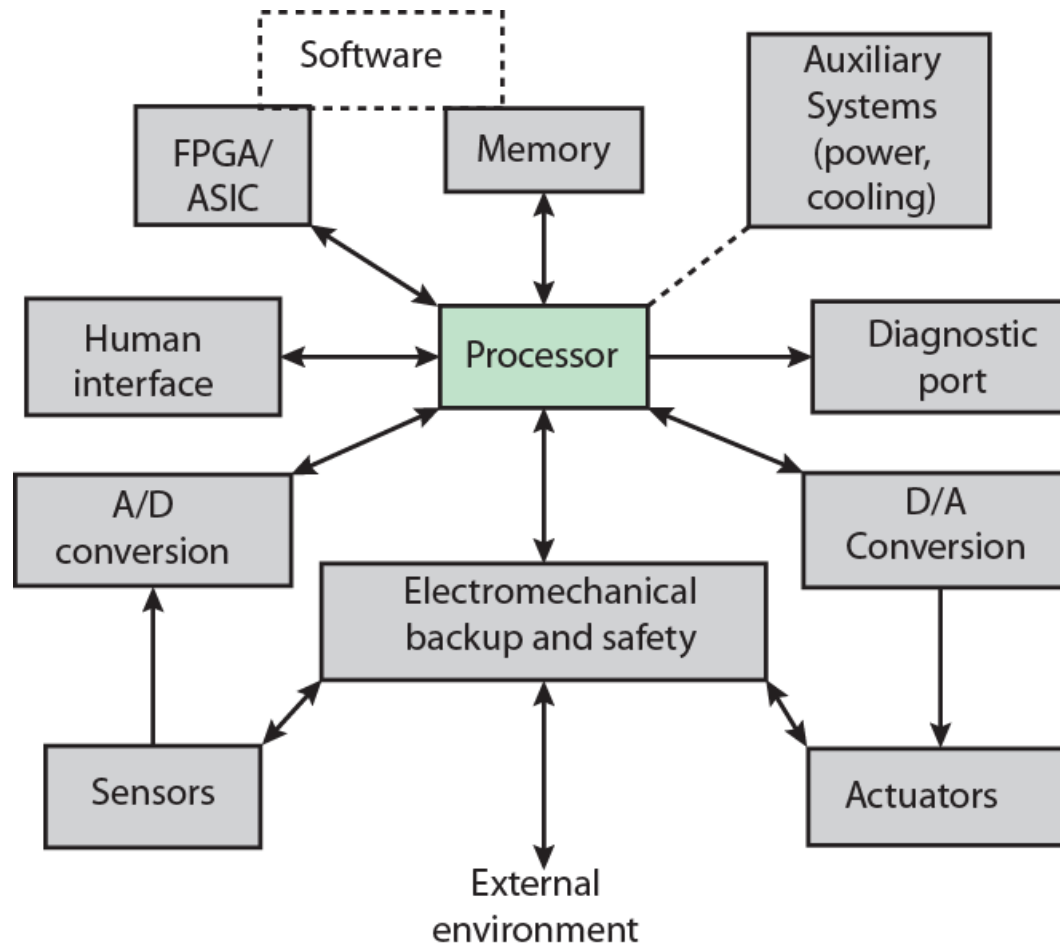
# Embedded Systems ARM

- ARM evolved from RISC design
- Used mainly in embedded systems
  - Used within product
  - Not general purpose computer
  - Dedicated function
  - E.g. Anti-lock brakes in car

# Embedded Systems Requirements

- Different sizes
  - Different constraints, optimization, reuse
- Different requirements
  - Safety, reliability, real-time, flexibility, legislation
  - Lifespan
  - Environmental conditions
  - Static v dynamic loads
  - Slow to fast speeds
  - Computation v I/O intensive
  - Discrete event v continuous dynamics

# Possible Organization of an Embedded System



# ARM Evolution

- Designed by ARM Inc., Cambridge, England
- Licensed to manufacturers
- High speed, small die, low power consumption
- PDAs, hand held games, phones
  - E.g. iPod, iPhone
- Acorn produced ARM1 & ARM2 in 1985 and ARM3 in 1989
- Acorn, VLSI and Apple Computer founded ARM Ltd.

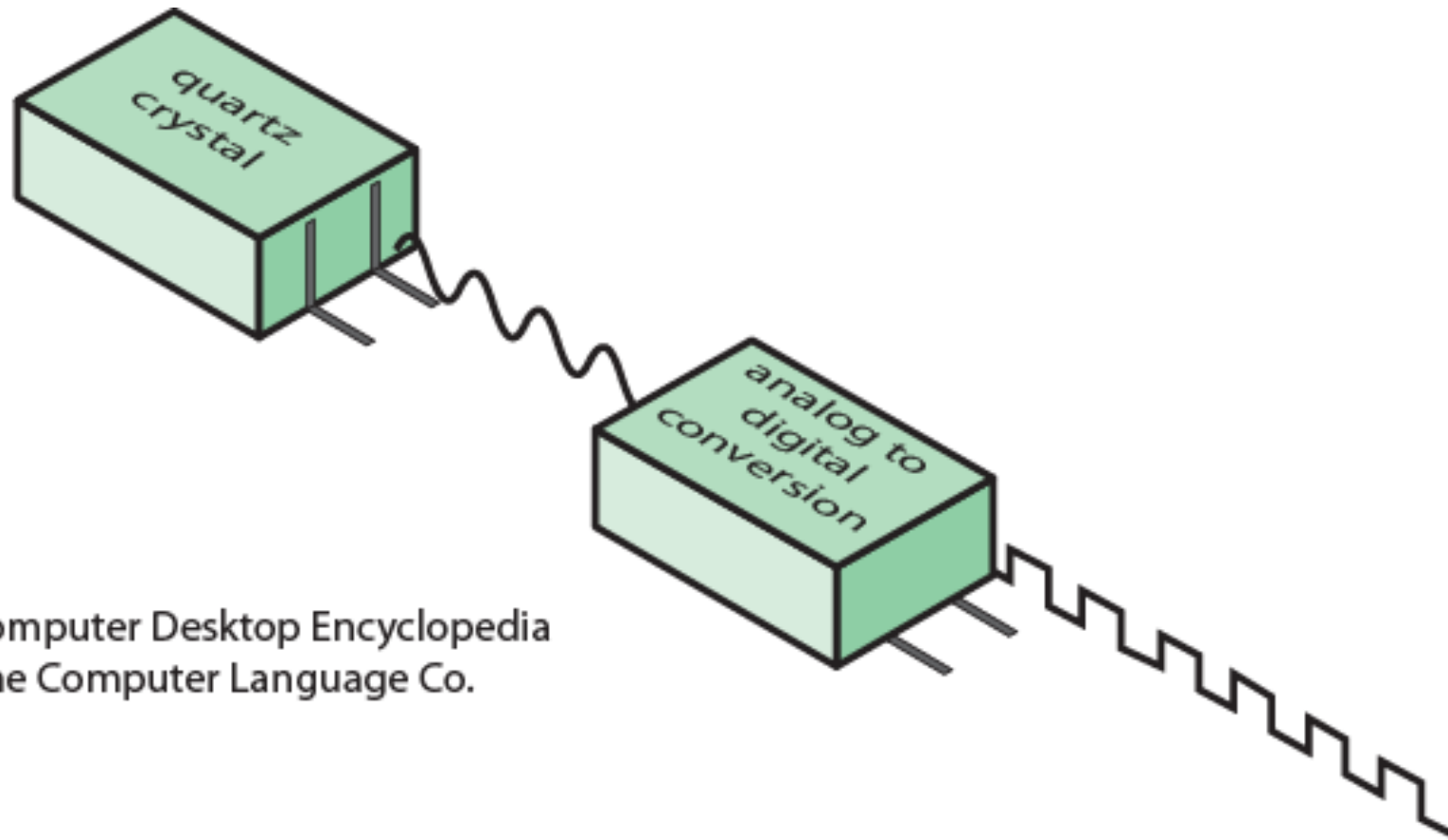
# ARM Systems Categories

- Embedded real time
- Application platform
  - Linux, Palm OS, Symbian OS, Windows mobile
- Secure applications

# Performance Assessment Clock Speed

- Key parameters
  - Performance, cost, size, security, reliability, power consumption
- System clock speed
  - In Hz or multiples of
  - Clock rate, clock cycle, clock tick, cycle time
- Signals in CPU take time to settle down to 1 or 0
- Signals may change at different speeds
- Operations need to be synchronised
- Instruction execution in discrete steps
  - Fetch, decode, load and store, arithmetic or logical
  - Usually require multiple clock cycles per instruction
- Pipelining gives simultaneous execution of instructions
- So, clock speed is not the whole story

# System Clock



From Computer Desktop Encyclopedia  
1998, The Computer Language Co.



# Instruction Execution Rate

- Millions of instructions per second (MIPS)
- Millions of floating point instructions per second (MFLOPS)
- Heavily dependent on instruction set, compiler design, processor implementation, cache & memory hierarchy

# Benchmarks

- Programs designed to test performance
- Written in high level language
  - Portable
- Represents style of task
  - Systems, numerical, commercial
- Easily measured
- Widely distributed

# Amdahl's Law

- Gene Amdahl [AMDA67]
- Potential speed up of program using multiple processors
- Concluded that:
  - Code needs to be parallelizable
  - Speed up is bound, giving diminishing returns for more processors
- Task dependent
  - Servers gain by maintaining multiple connections on multiple processors
  - Databases can be split into parallel tasks

# Amdahl's Law Formula

- For program running on single processor
  - Fraction  $f$  of code infinitely parallelizable with no scheduling overhead
  - Fraction  $(1-f)$  of code inherently serial
  - $T$  is total execution time for program on single processor
  - $N$  is number of processors that fully exploit parallel portions of code

$$\text{Speedup} = \frac{\text{time to execute program on a single processor}}{\text{time to execute program on } N \text{ parallel processors}} = \frac{T(1-f) + Tf}{T(1-f) + \frac{Tf}{N}} = \frac{1}{(1-f) + \frac{f}{N}}$$

- Conclusions
  - $f$  small, parallel processors has little effect
  - $N \rightarrow \infty$ , speedup bound by  $1/(1-f)$ 
    - Diminishing returns for using more processors

# Internet Resources

- <http://www.intel.com/>
  - Search for the Intel Museum
- <http://www.ibm.com>
- <http://www.dec.com>
- Charles Babbage Institute
- PowerPC
- Intel Developer Home

# References

- AMDA67 Amdahl, G. “Validity of the Single-Processor Approach to Achieving Large-Scale Computing Capability”, *Proceedings of the AFIPS Conference, 1967.*