

COMPUTER HISTORY AND PERFORMANCE

ARKOM | IF221104 | #2



Yoga Ari Tofan, S.Kom., M.Kom.

Program Studi Informatika, Fakultas Ilmu Komputer
UPN “Veteran” Jawa Timur

Computer Architecture

LEARNING OBJECTIVES

By the end of this session, students should be able to:

1. Explain major milestones in computer technology.
2. Identify technology trends that drive computer performance.
3. Explain strategies for balanced and efficient system performance.
4. Describe the evolution of Pentium and PowerPC processors.

WHY STUDY COMPUTER HISTORY?

Computer history is not only a timeline, it explains why architectures change.

Limitation

Each generation faced limits in size, speed, reliability, cost, or power.

Innovation

New devices and architectural techniques were introduced to overcome those limits.

Trade-off

Higher performance usually creates new bottlenecks elsewhere in the system.

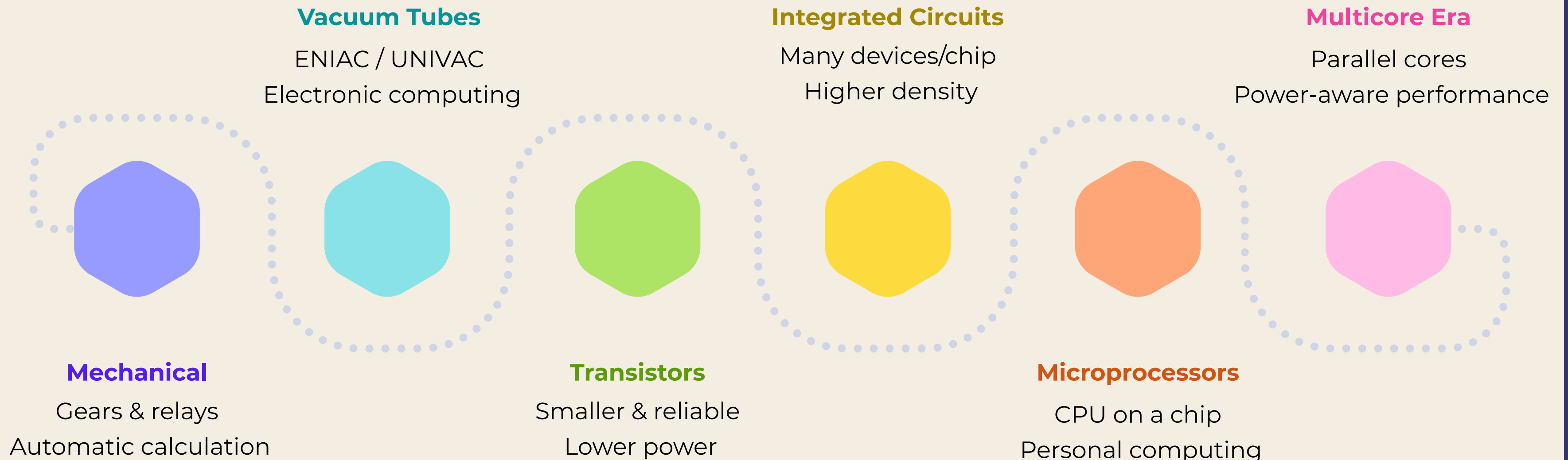


Key

Technology evolves → Processors become faster → Bottlenecks shift → Architecture adapts

EVOLUTION OF COMPUTER TECHNOLOGY

Focus on the dominant technology and its architectural impact.



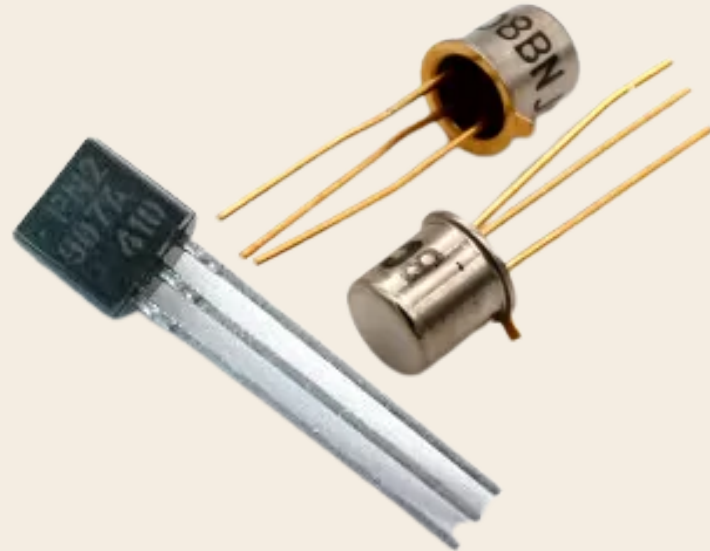
FROM ENIAC TO MODERN SYSTEMS

Era	Dominant Technology	What Improved	New Challenge
1940s-50s	Vacuum tubes	Electronic speed	Size, heat, reliability
1950s-60s	Transistors	Reliability, size, power	Growing complexity
1960s-70s	Integrated circuits	Density and cost	Memory / I/O balance
1970s-2000s	Microprocessors / VLSI	Clock speed, integration	Power and memory wall
2000s-today	Multicore / accelerators	Parallel throughput	Parallelism & efficiency

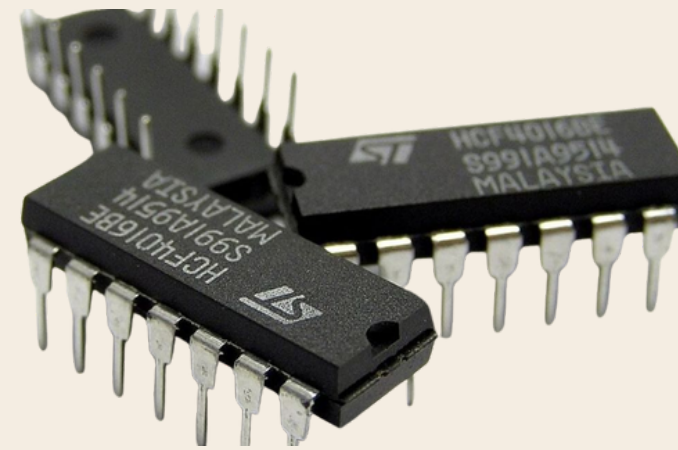
FROM ENIAC TO MODERN SYSTEMS



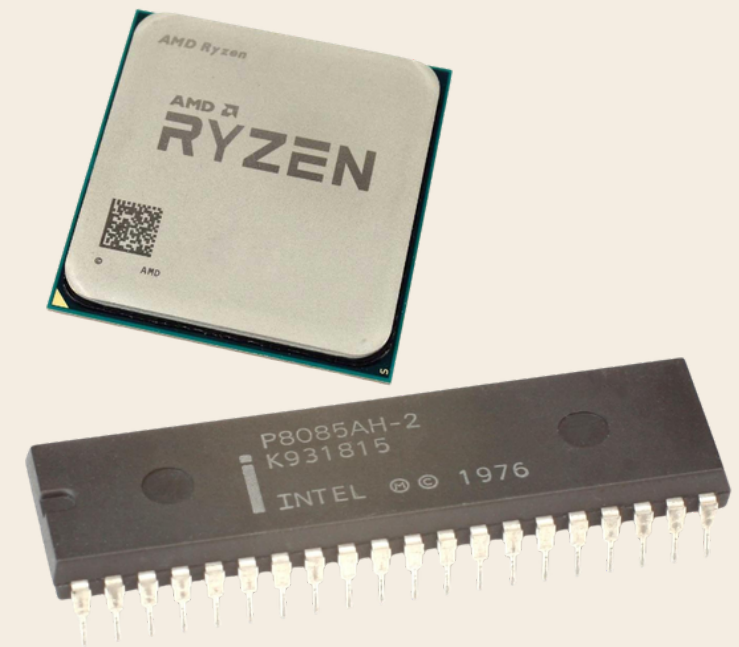
Vacuum Tubes



Transistors



Integrated Circuits



**Microprocessors
Multicore Era**

THE MICROPROCESSOR REVOLUTION

Integrating the CPU onto a single chip changed both computer design and accessibility.

Integration

More transistors on one chip
reduced size and cost.

Performance

Shorter on-chip paths enabled
faster operation.

Accessibility

Affordable processors accelerated personal
computing.



Key

Faster processors created a new question:
Can memory, I/O, and software keep up?

TECHNOLOGY SCALING AND MOORE'S LAW

Historically, transistor density increased rapidly, enabling more complex processors.

More transistors

- Larger caches
- Deeper pipelines
- More execution units
- Multiple cores

Important

Moore's Law describes a historical scaling trend—not a guarantee that application performance doubles automatically.



Key

Architecture determines how effectively additional transistor resources become useful performance.

PERFORMANCE IS MORE THAN CLOCK SPEED

A higher GHz value does not automatically mean a faster computer.

Clock Rate

How many clock cycles occur per second.

CPI

Average clock cycles needed per instruction.

Instruction Count

How much work the program actually executes.



Key

Real performance depends on the workload, instruction set, microarchitecture, memory behavior, and I/O—not clock rate alone.

UNDERSTANDING COMPUTER PERFORMANCE

Latency / Response Time

Time required to complete one task.

Lower is better.

Utilization

How intensively a resource is being used.

Context dependent.

Throughput

Amount of work completed per unit time.

Higher is better.

Scalability

Ability to handle more work by adding resources.

System-level metric.

A SIMPLE CPU PERFORMANCE MODEL

Use the equation to reason about where execution time comes from.

$$\text{CPU Time} = \text{Instruction Count} \times \text{CPI} \times \text{Clock Cycle Time}$$

Instruction Count

Depends on program,
algorithm, compiler, and ISA.

CPI

Depends strongly on microarchitecture
and memory behavior.

Clock Cycle Time

Inverse of clock rate; shorter
cycles can increase speed.

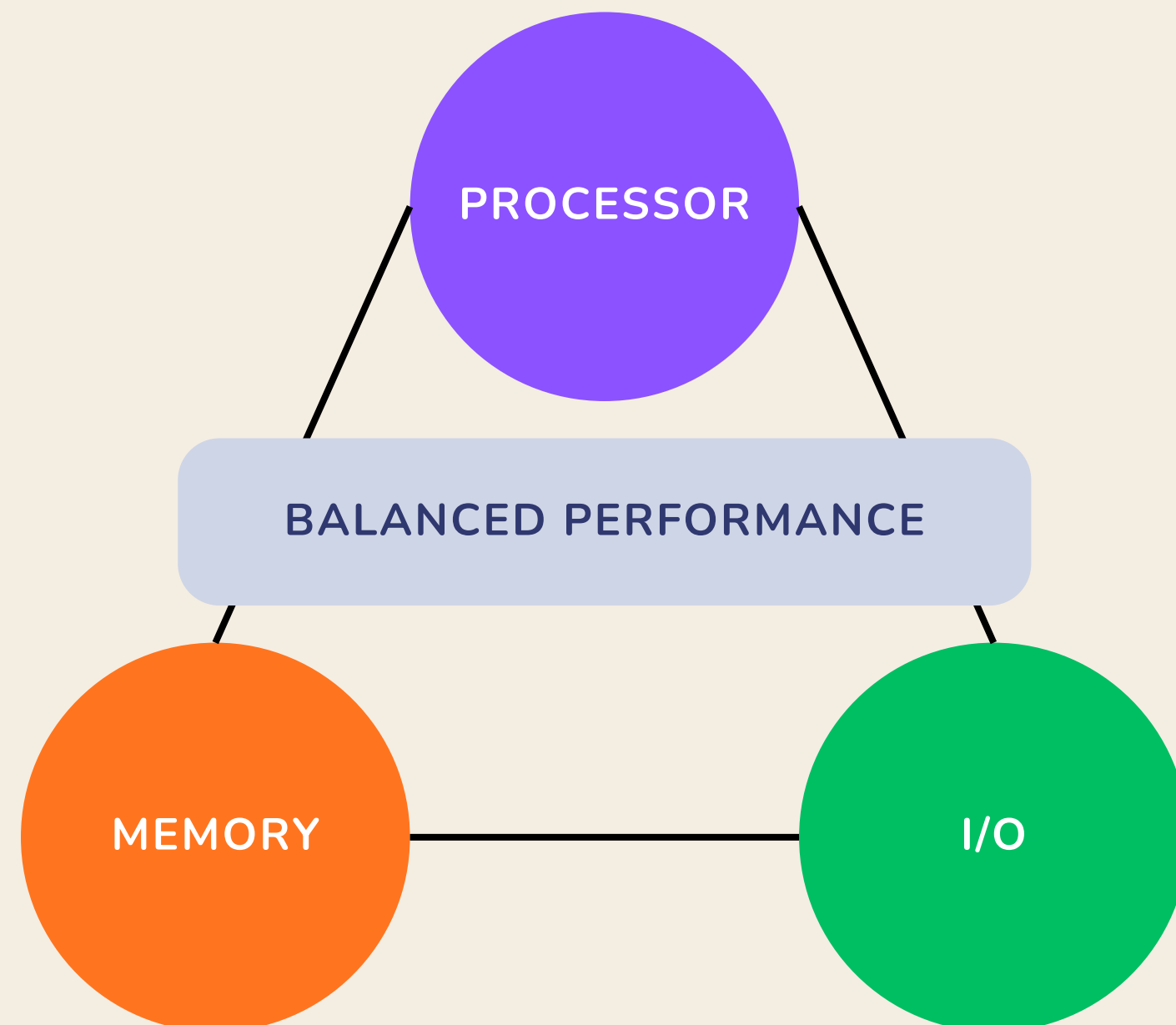


Key

Improving only one factor may not improve total
system performance proportionally.

THE PERFORMANCE BALANCE PROBLEM

A system is only as effective as the interaction among processor, memory, and I/O.



Key

A very fast CPU may spend much of its time waiting if memory or I/O cannot supply data fast enough.

STRATEGIES FOR IMPROVING PERFORMANCE

A system is only as effective as the interaction among processor, memory, and I/O.

Cache & Memory Hierarchy

Keep frequently used data closer to the CPU.

Pipelining

Overlap stages of instruction execution.

Multiple Cores

Increase throughput through parallel execution.

Wider / Faster Interconnects

Move data among components more efficiently.

Branch Prediction

Reduce stalls caused by control-flow decisions.

Specialized Accelerators

Use hardware optimized for particular workloads.

PENTIUM EVOLUTION

Selected milestones illustrate how x86 processors pursued higher performance.

- **Pentium**
1993 - Superscalar x86
- **Pentium Pro**
1995 - Out-of-order execution
- **Pentium II / III**
1997–99 - Improved pipeline & SIMD
- **Pentium 4**
2000 - Very deep pipeline / high clock
- **Later x86**
2000s+ - Multicore & efficiency



Lesson

Increasing clock frequency alone eventually ran into power and thermal limits.

POWERPC EVOLUTION

PowerPC represents a different design lineage, strongly associated with RISC principles.

- **PowerPC 601**
1993 - Early PowerPC implementation; bridge from earlier IBM designs.
- **PowerPC 603 / 604**
1990s - Variants optimized for different performance and power targets.
- **PowerPC G3 / G4**
1990–2000 - Used prominently in Apple systems; later designs added vector capability.
- **POWER / Power ISA**
Modern Lineage - Continued development in high-performance and enterprise systems.



Info

RISC vs. CISC will be revisited later in the course; here, focus on the historical design trajectory.

PENTIUM AND POWERPC: TWO DESIGN TRAJECTORIES

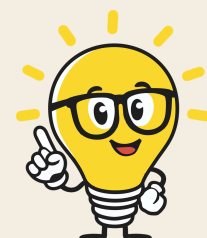
PowerPC represents a different design lineage, strongly associated with RISC principles.

PENTIUM / x86 LINEAGE

- Maintained compatibility with the x86 instruction-set family.
- Used increasingly sophisticated internal execution techniques.
- Performance evolved through superscalar, out-of-order, SIMD, and multicore designs.

POWERPC / RISC LINEAGE

- Designed around a RISC-oriented instruction-set philosophy.
- Emphasized regular instruction formats and efficient implementation.
- Evolved across desktop, embedded, and enterprise contexts.



Info

Different ISA philosophies can still employ similar microarchitectural techniques to improve performance.

FROM CLOCK-SPEED GROWTH TO MULTICORE

Power and thermal constraints changed the direction of processor performance.

Earlier Trend

Higher clock frequency + deeper pipelines → faster single-thread execution

The Limit

More frequency → more power & heat → diminishing returns

Modern Response

Multiple cores + parallelism + specialization → better efficiency



Key

Performance engineering shifted from “make one core faster” toward “use available silicon and power more efficiently.”

CLASS DISCUSSION

WHERE IS THE BOTTLENECK?

A laptop has a fast CPU, but applications still feel slow when opening large files.

Question:

Which component would you investigate first and what evidence would you collect?

CPU

Memory

Storage / I/O

Think in terms of latency, throughput, utilization, and the workload. A fast processor cannot compensate for every memory or I/O bottleneck.



CONCLUSION

- Computer architecture evolves in response to technology limits and workload needs.
- Performance is not determined by clock speed alone.
- Processor, memory, and I/O must be balanced.
- Pentium and PowerPC illustrate different historical design trajectories.
- Modern performance increasingly relies on parallelism and efficiency.





THANK YOU.

E: yoga.if@upnjatim.ac.id

P: 0813-3485-7761

L: Twin Tower B Lt.5 R.501