CPE/EE 421 Microcomputers

Instructor: Dr Aleksandar Milenkovic
Lecture Notes
S01

*Material used is in part developed by
Dr. D. Raskovic and Dr. E. Jovanov

Class#1

CPE/EE 421 Microcomputers

- Syllabus
  - textbook & other references
  - grading policy
  - important dates
  - course outline
- Prerequisites
  - memory organization
  - decoding
  - combinatorial and sequential logic
  - important for system architecture
- Microcomputer Lab (EB 106)
  - Introduction sessions
  - Lab instructor

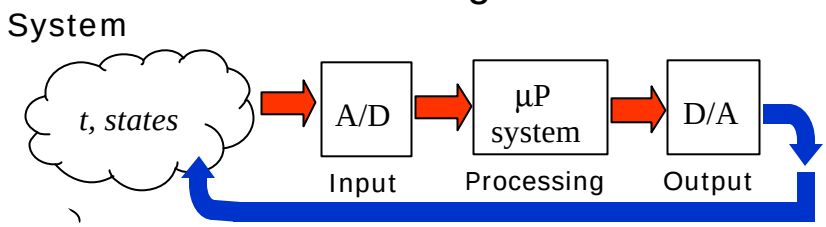
CPE/EE 421 Microcomputers

- LAB Session
  - on-line LAB manual
  - Access cards
  - Accounts
- Lab Assistant: Rami Al'namneh
- Lab sessions
  (Select 4 in class, 5th for Rami's Office Hours)
  - Option #1: Monday 6:00 – 7:00 PM
  - Option #2: Monday 7:00 – 8:00 PM
  - Option #3: Tuesday 7:00 – 8:00 PM
  - Option #4: Tuesday 8:00 – 9:00 PM
  - Option #5: Wednesday 6:00 – 7:00 PM
  - Option #6: Wednesday 7:00 – 8:00PM
- Sign-up sheet - if needed

Microcomputer

- Stand alone system
  based on a microprocessor
- An embedded system –
  dedicated to a specific application
  - control system/monitoring system
  - optimization for a single function
    (system resources, extension, ...)
  - block diagram
    - inputs → processing → outputs
- Number of microprocessors
  in our environment?

Block Diagram



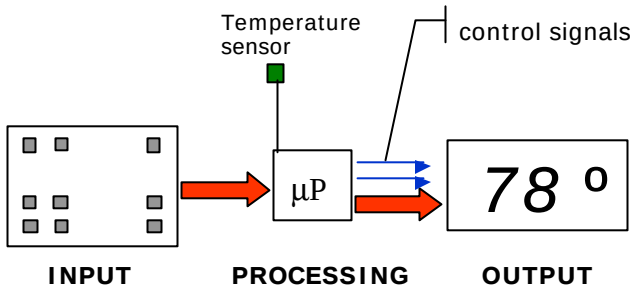
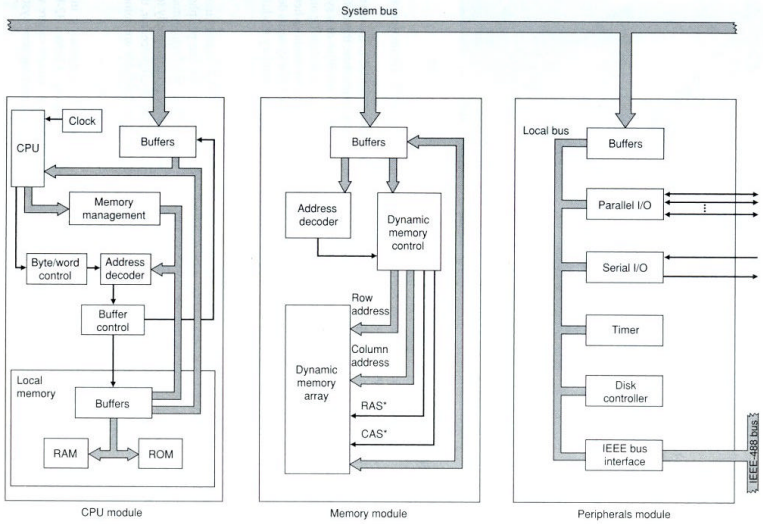


Figure 1.1 Block diagram of a possible modular microcomputer



Microprocessor System Architecture

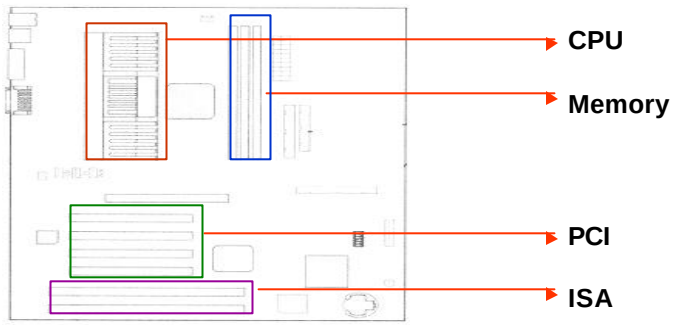
- System Architecture
  - Single Board Computers (SBC)
  - block diagram (modules, cards)
- System bus
  - Multibus, VME, ISA, PCI, ...
  - Multiple masters
- CPU
  - Clock and CPU Control Circuits
    - 1 MHz - 1 GHz
    - power-up and reset circuits
  - Address Decoder
  - Address and Data Bus Buffers
  - Bus Arbitration Control
  - Memory management

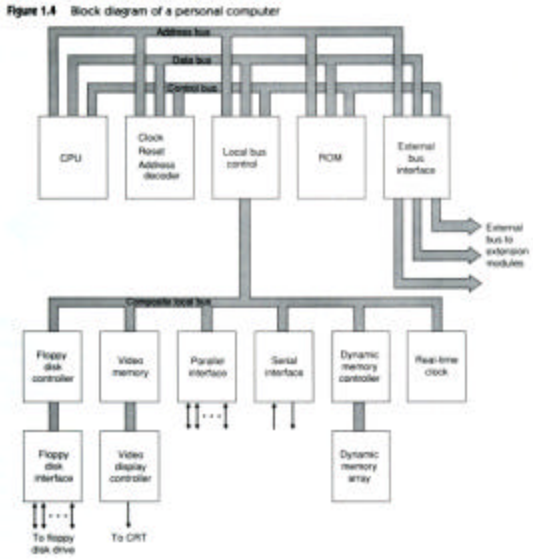
Microprocessor System Architecture #2

- Memory Module
  - Virtual memory / Physical memory
  - ROM, RAM, video memory
  - static/dynamic
- Peripheral Module
  - serial interface
    - RS232 (CRT, mouse), USB, Firewire (IEEE 1394)
  - parallel interface
    - printer interface
  - timer
    - time/frequency measurement

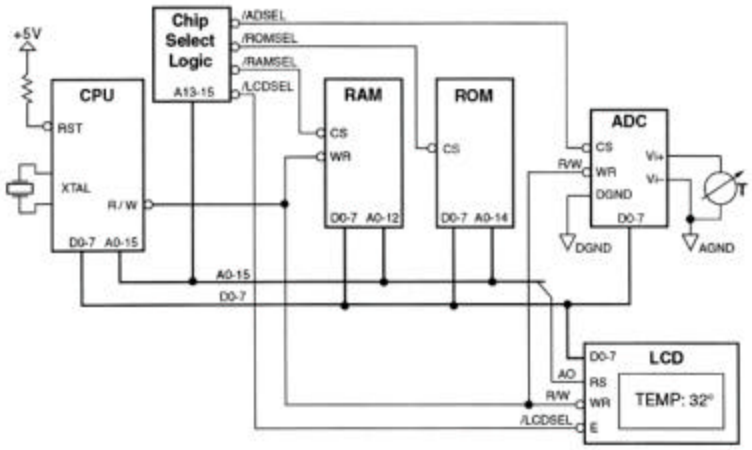
Microprocessor System Architecture #3

- PC architecture



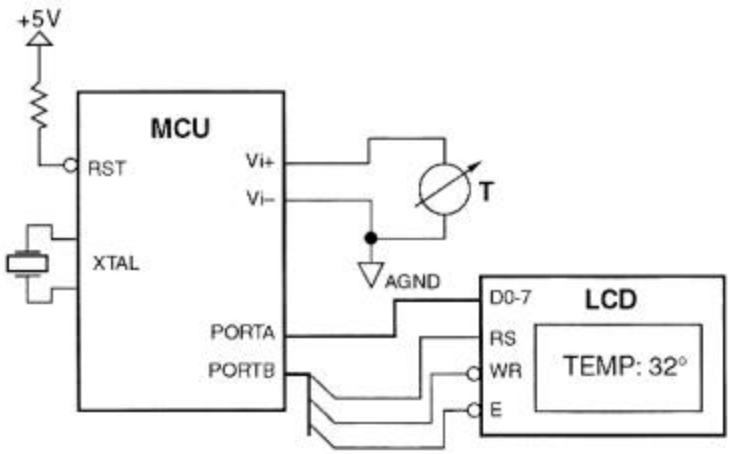


Block Diagram of a personal computer



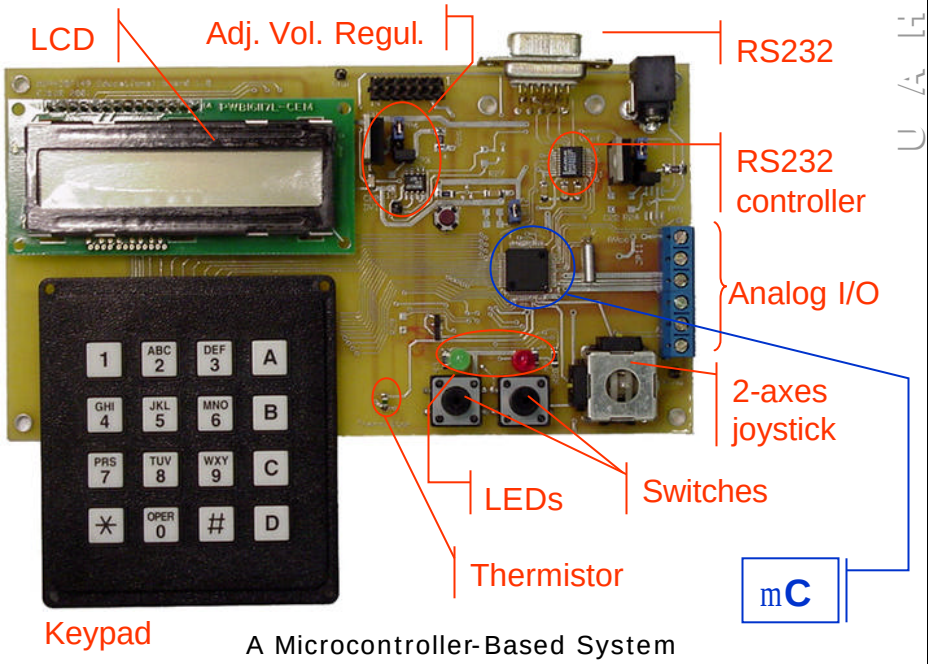
Todd D. Morton
Embedded Microcontrollers

A Typical Small Embedded System – Digital Thermometer



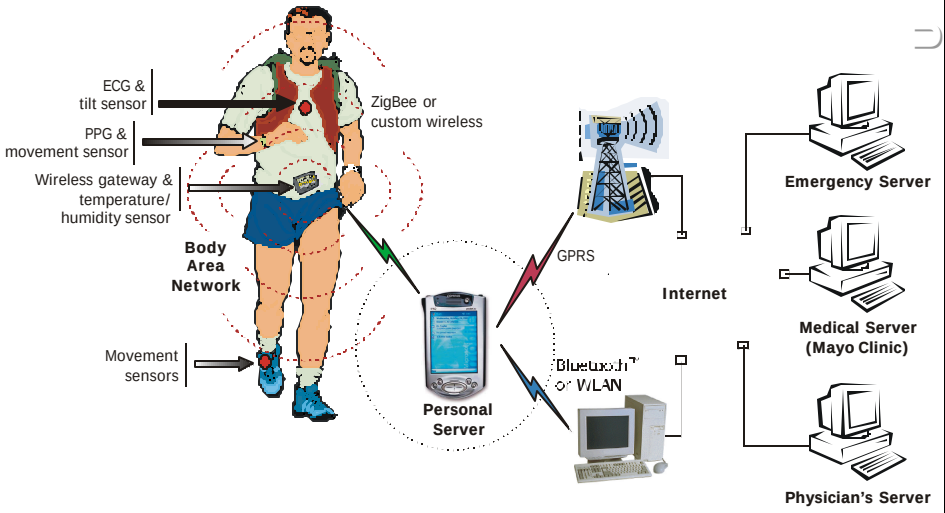
Todd D. Morton
Embedded Microcontrollers

A Microcontroller-Based System

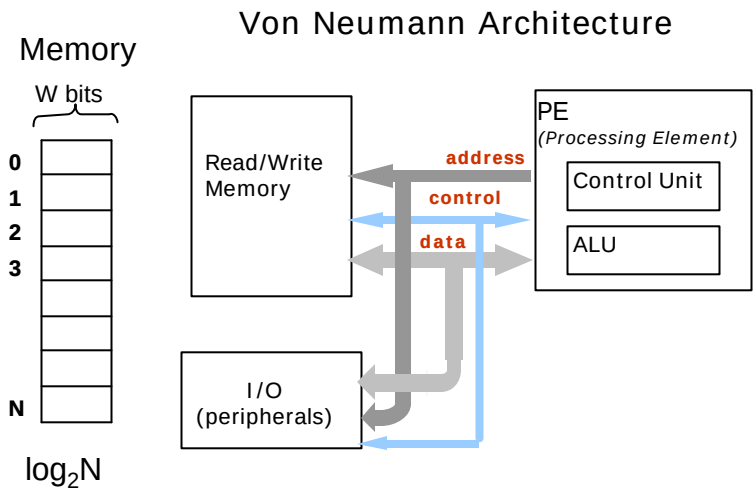


A Microcontroller-Based System

Wireless Body Area Network



Conventional Computer Architecture

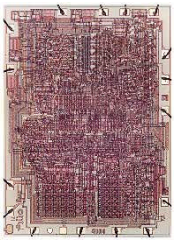


Microprocessors - History

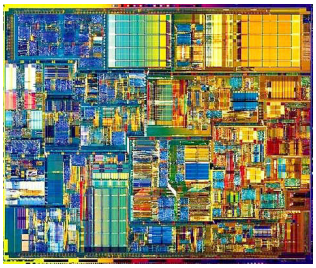
- Implementations
  - size (room, cabinet, desktop, handheld, ...)
  - speed (doubling 18-20 months)
  - power consumption
- Von Neumann Architecture
  - Processing Elements
    - sequential execution
  - Read/Write Memory
    - linear array of fixed size cells
    - Data and instruction store
  - I/O unit
  - Address/Data/Control bus

Intel: First 30+ Years

- Intel 4004
  - November 15, 1971
  - 4-bit ALU, 108 KHz, 2,300 transistors, 10-micron technology



- Intel Pentium 4
  - August 27, 2001
  - 32-bit architecture, 1.4 GHz (now 3.08), 42M transistors (now 55+M), 0.18-micron technology (now 0.09)

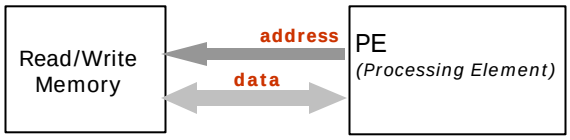


Technology Directions: SIA Roadmap

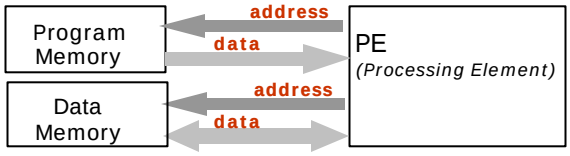
Year	1999	2002	2005	2008	2011	2014
Feature size (nm)	180	130	100	70	50	35
Logic trans/cm^2	6.2M	18M	39M	84M	180M	390M
Cost/trans (mc)	1.735	.580	.255	.110	.049	.022
#pads/chip	1867	2553	3492	4776	6532	8935
Clock (MHz)	1250	2100	3500	6000	10000	16900
Chip size (mm^2)	340	430	520	620	750	900
Wiring levels	6-7	7	7-8	8-9	9	10
Power supply (V)	1.8	1.5	1.2	0.9	0.6	0.5
High-perf pow (W)	90	130	160	170	175	183

History #2

Von Neumann Architecture



Harvard Architecture



History #3

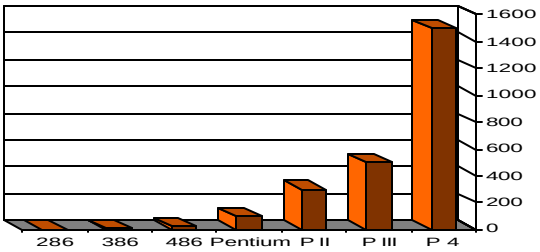
- Processor/memory discrepancy
  - Memory hierarchy
  - On-chip/off-chip memory
- Microprocessor execution
  - Fetch > Decode > Execute
- System on a chip - Microcontroller
  - Cost, smaller PCB, reliability, power.
  - Applications
- Evolution
  - Microprocessor
  - Microprocessor-on-a-chip
  - System-on-a-chip
  - Distributed-system-on-a-chip

History #4

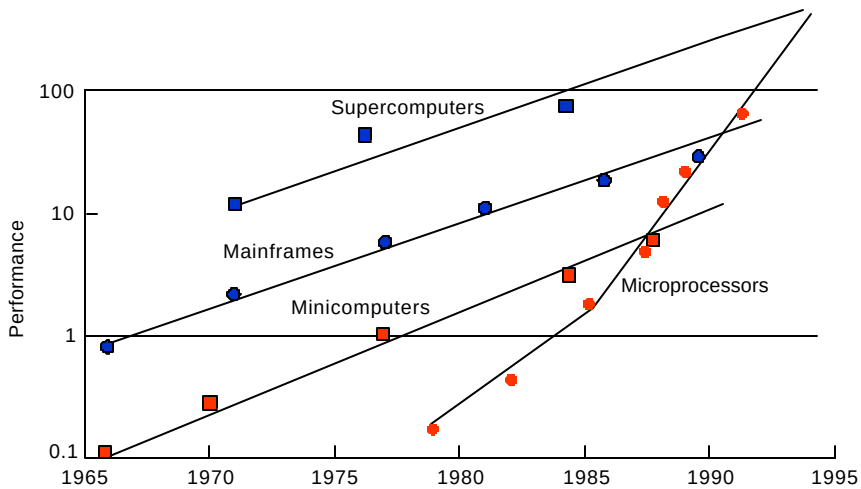
- Challenges
  - scalability
    - billions of small devices
    - performance
  - availability
    - hardware changes
    - system upgrade
    - failures
    - code enhancements
  - fault tolerance

History #5

Year	Processor	MIPS
1969	4004	0.06
1970's	808x	0.64
1982	286	1
1985	386	5
1989	486	20
1993	Pentium	100
1996	P II	250
1999	P III	500
2000	P 4	1500

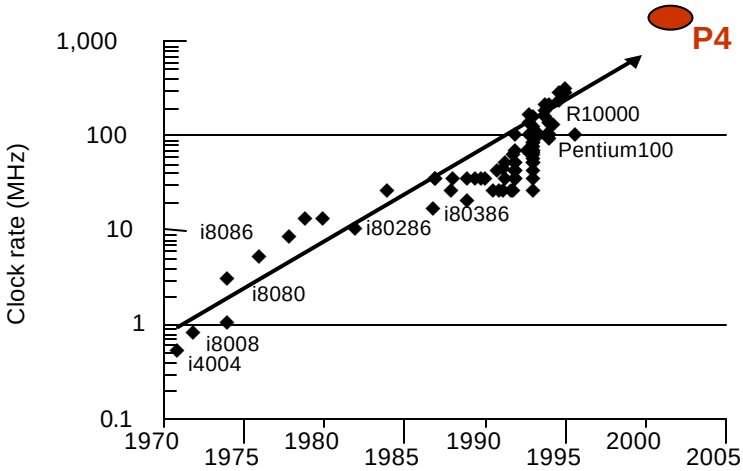


Technology Trends



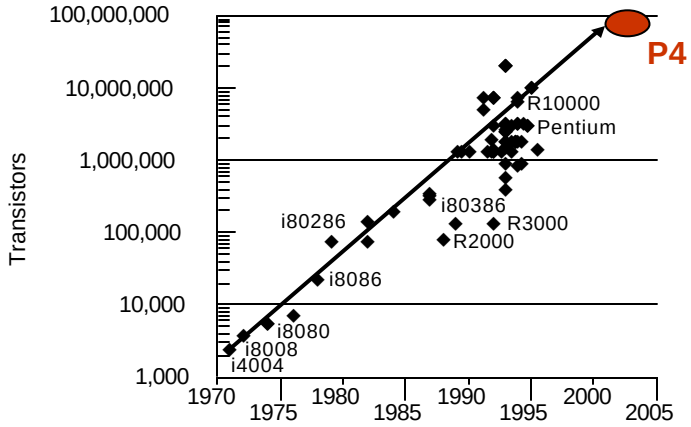
The natural building block for multiprocessors is now also about the fastest!

Clock Frequency Growth Rate



- 30% per year

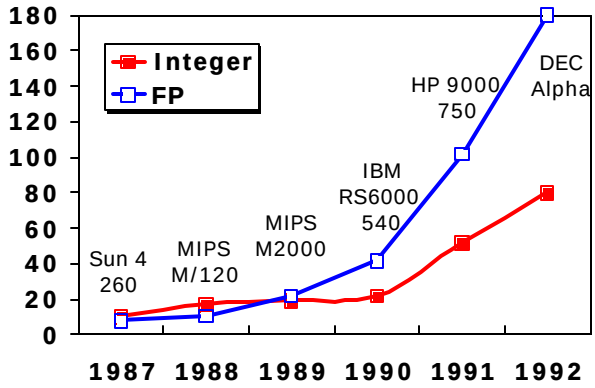
Transistor Count Growth Rate



- 100 million transistors on chip by early 2000's A.D.
- Transistor count grows much faster than clock rate
  - 40% per year, order of magnitude more contribution in 2 decades

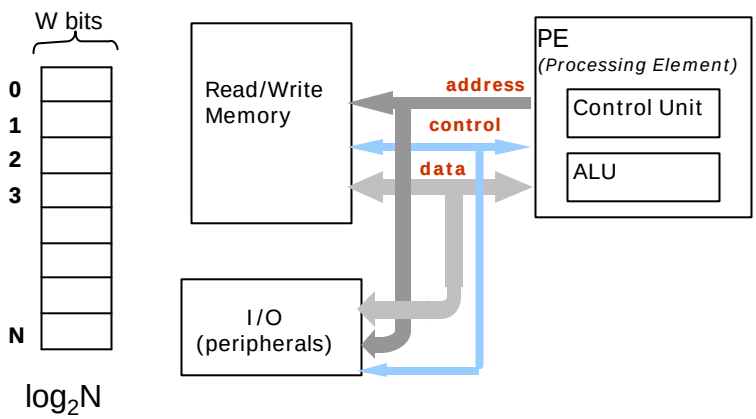
General Technology Trends

- Microprocessor performance increases 50%-100% per year
- Transistor count doubles every 3 years
- DRAM size quadruples every 3 years
- Huge investment per generation is carried by huge commodity market



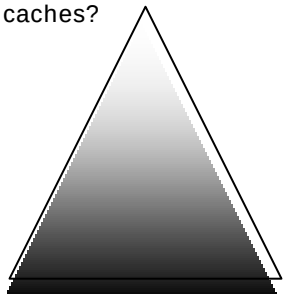
Conventional Computer Architecture

Von Neumann Architecture



Storage

- Divergence between memory capacity and speed more pronounced
  - Capacity increased by 1000x from 1980-95, speed only 2x
  - Gigabit DRAM by c. 2000, but gap with processor speed much greater
- Larger memories are slower, while processors get faster
  - Need to transfer more data in parallel
  - Need deeper cache hierarchies
  - How to organize caches?



	Speed	Size
Registers	ns	~KB
Cache	10ns	~MB
Main memory	100ns	~100MB
Hard disk	10ms	~10GB
Archive	>100ms	~TB

Instruction Sets

- Software costs growing faster than hardware costs (1970s)
  - Machine language v.s. HLL
  - Support for high-level languages
  - Gap between high level languages and computer hardware - semantic gap
- CISC - **C**omplex **I**nstruction **S**et **A**rchitecture
  - Variety of instructions and addressing modes
  - DEC VAX
- HLLCA - **H**igh **L**evel **L**anguage **C**omputer **A**rchitecture

RISC Architectures

- Resolve problems using simpler architecture
  - "The case for the reduced instruction set computers" Patterson & Ditzel [1980]
- Stanford MIPS (Hennessy, 1981)
- Commercial processors: MIPS R2000 (1986), IBM RS6000, SPARC, PowerPC, etc.
- Good design methodology
- Efficient pipelining and compiler-assisted scheduling of pipeline
- Make the Common Case Fast
  - favor the frequent case

RISC Methodology

Program execution time:

$$T = \sum_i n_i \cdot CPI_i \cdot T_{cycle}$$

For all instructions
in the instruction set

processor cycle time [s]

cycles per instruction

instruction count

80x86 Instruction Mix for SPECint92 Programs [PatHen96]

Instruction	Frequency (%)
load	22
cond. branch	20
compare	16
store	12
add	8
and	6
sub	5
mov reg-reg	4
or	1
xor, not, etc.	1
uncond. branch	1
call	1
return, jmp indirect	1
shift	1

42%

70%

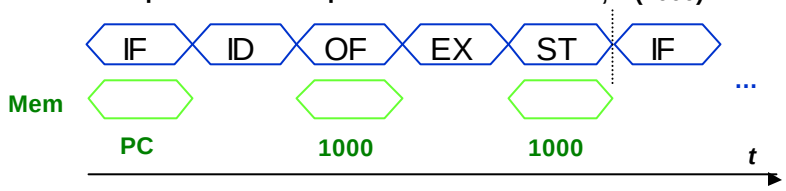
Instruction Execution

A=B+C

result = op1 *OPERATION* op2

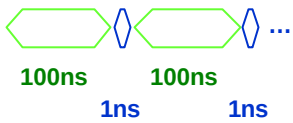
A+=C

ADD R1, M(1000)



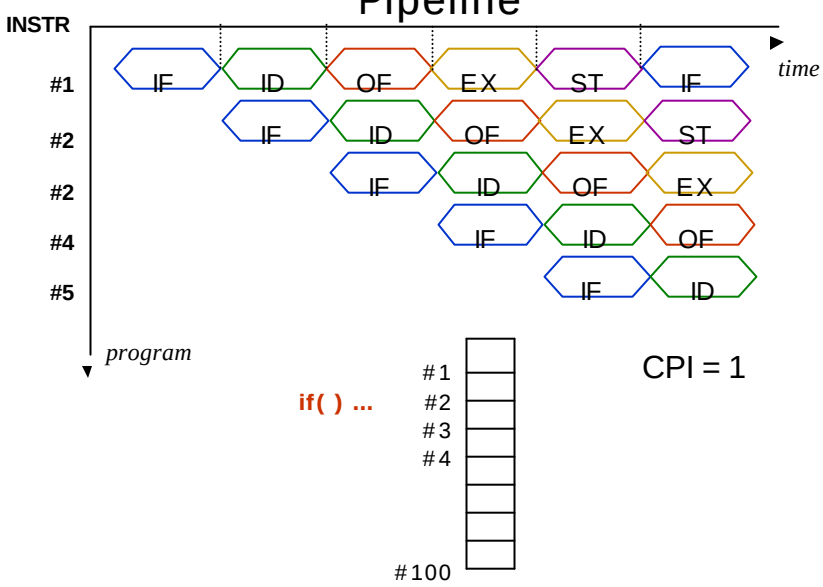
When f = 5MHz ('70s) T_{cycle} = 200ns
T_{mem} = 200ns

When f = 1000MHz ('90s) T_{cycle} = 1ns
T_{mem} = 50ns



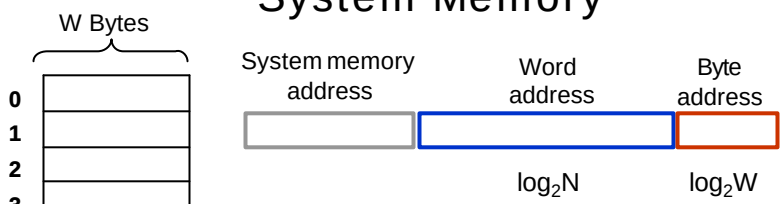
3 × 100ns = 300ns
2 × 1ns = 2ns
302ns

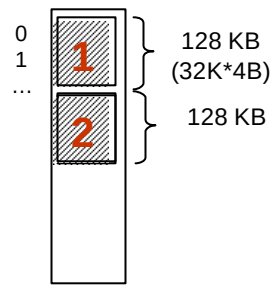
Pipeline



CPI = 1

System Memory





Example: 2 banks, 32 KW each, 4-byte words