CPE 323 Introduction to Embedded Computer Systems: An Introduction

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Dr. Aleksandar Milenkovic

Outline

- Administration
- Technology Trends
- Embedded systems
  - What are they?
  - Where do we find them?
  - Structure and Organization
  - Software Architectures

CPE 323 Administration

- Syllabus
  - instructor, lab instructors, office hours
  - textbook & other references
  - grading policy
  - important dates
  - course outline
- Prerequisites
  - Digital systems: number representation, combinational (gates), and sequential logic (latches, flip-flops)
  - Computer architecture and organization
  - C/C++ Programming
- Embedded Systems Laboratory
  - Located in EB 106
  - Lab policies
  - Lab assistants

CPE 323 DOs (1 – 5)

- 1. Work hard, have fun ☺ ... because
  - It's better to have fun than not to have fun ...
  - Many students landed jobs thanks to skill gained in this course
  - Important course for your senior design
    - >80% of senior design projects use microcontrollers
- 2. Manage your time well: lectures, homeworks, labs
- 3. Attend classes
- 3. Make maximum use of class time
  - Skim through lecture notes in advance
  - Take notes
- 4. Ask questions when in doubt
  - *It's better to be a fool for 5 minutes of your life by asking a question than to be a fool for the rest of your life by not knowing the answer*
- 5. Use instructor's office hours to get extra help
  - I plan to have several non-mandatory recitation session on Friday (time to be determined)

CPE 323 DOs (6 – 10)

- 6. Make the best from your time in laboratory
  - Learning by doing
  - *I hear and I forget, I see and I remember,*
    I do and I understand
- 7. Make the best from your homeworks
  - Learning by doing; Problems similar to exams
  - *I hear and I forget, I see and I remember,*
    I do and I understand
  - Do not start your homework one hour before the due date
- 8. Collaboration=YES, Cheating=NO
  - Exams tells whether you very practicing the latter
- 9. Learn how to be independent
  - Read textbook, user manuals, reference manuals
  - Important skill to get right information on time
- 10. Make the best of this class
  - Learn how to survive tough course/instructor ☺
  - Get real-world problem solving skills
  - Become proficient in embedded systems (they are everywhere)

CPE 323 Laboratory DOs

- 1. Read the lab tutorial in advance
  - It has a demo example deconstructed
- 2. Understand assignment(s)
  - Note: Many assignments can be completed outside the lab
- 3. Develop a plan for solving problems
- 4. Give a try to design solution (pen & paper)
- 5. Write the code
- 6. Document your code
- 7. Test your design (try different inputs, corner cases)
- 8. Demonstrate your program executing
- 9. Be proactive: give brief description to demonstrate your solution & good understanding
- 10. Answer questions when asked
  - You will be asked to explain specifics of your solution
  - You will not get FULL credits (if any) if you turn in your code without demonstration

CPE323 DON'Ts

- 1. Do not talk during class
  - Respect others (treat others the way you would like to be treated)
- 2. Do not sleep in class
  - Not comfortable, too much noise
  - Classes are not mandatory
- 3. Do not cheat on homeworks
  - Comply with University policies
  - They are perfect tool for exam preparation
  - Give you an important feedback
- 4. Do not cheat in laboratory
  - Comply with University policies
  - Exams include questions related to labs
  - Red flags if discrepancies are uncovered
- 5. Do not be disrespectful to lab instructors
  - They are there to help you, but not to do your work

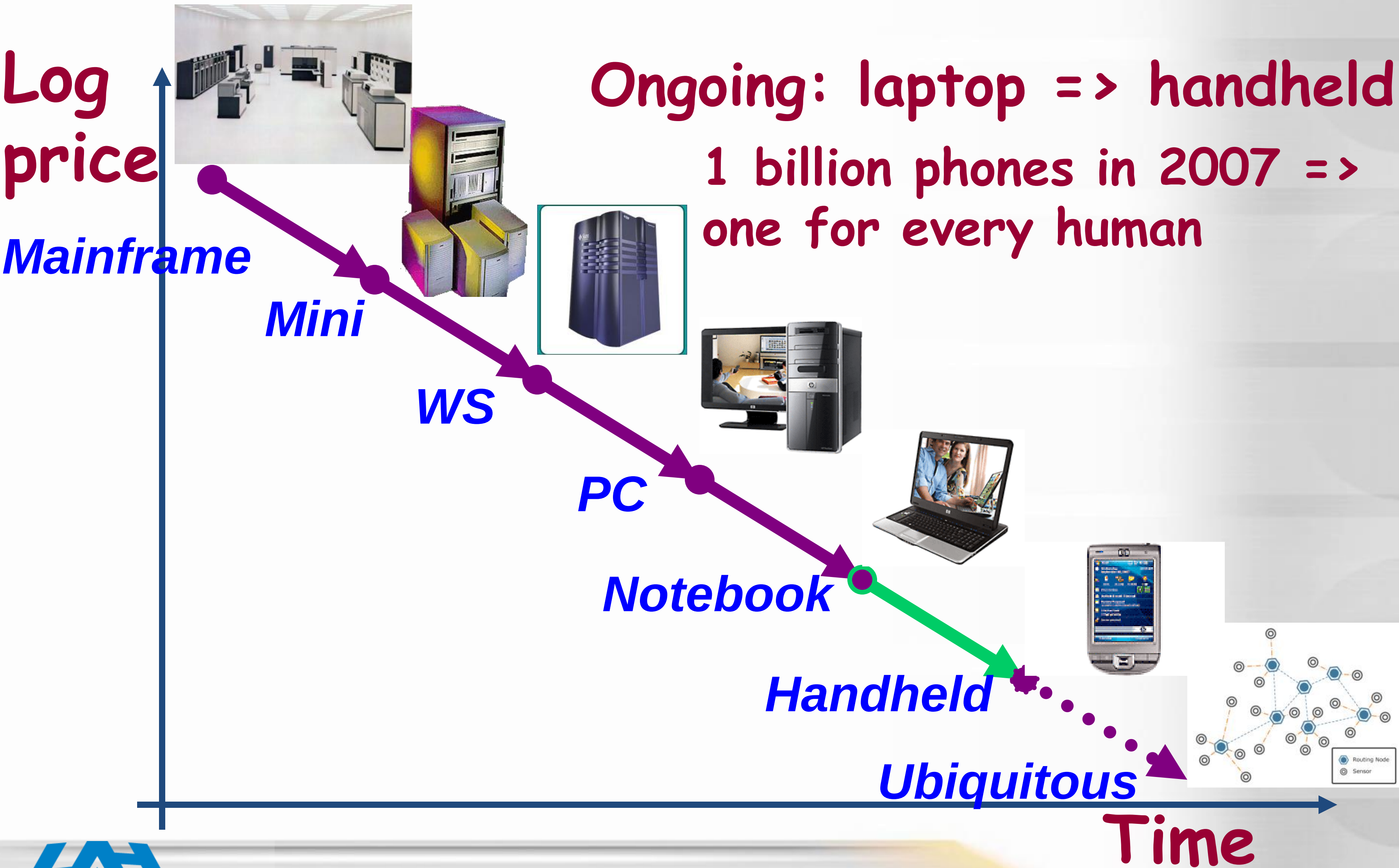
Outline

- Administration
- Technology Trends
- Embedded systems
  - What are they?
  - Where do we find them?
  - Structure and Organization
  - Software Architectures

History of Computing

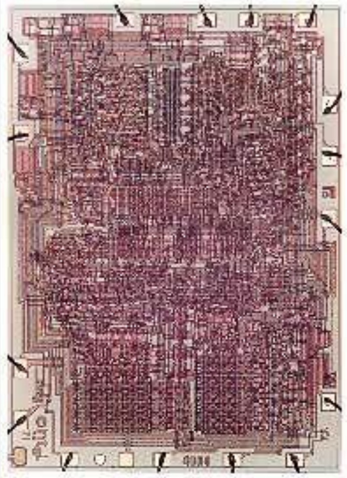
Ongoing: laptop => handheld

1 billion phones in 2007 => one for every human

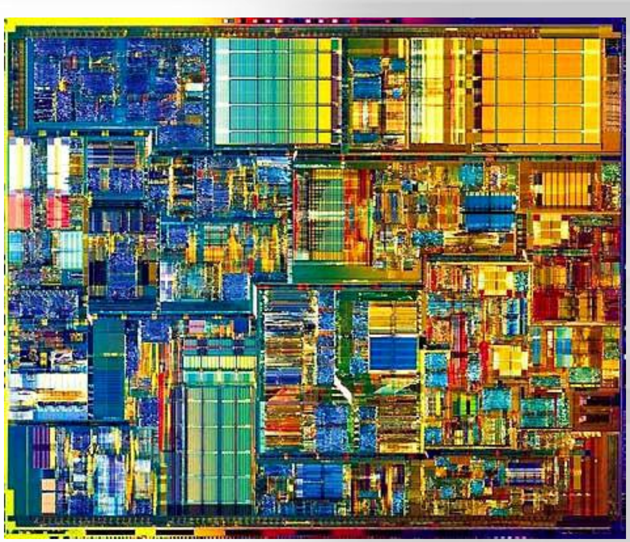


Intel: First 30+ Years

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



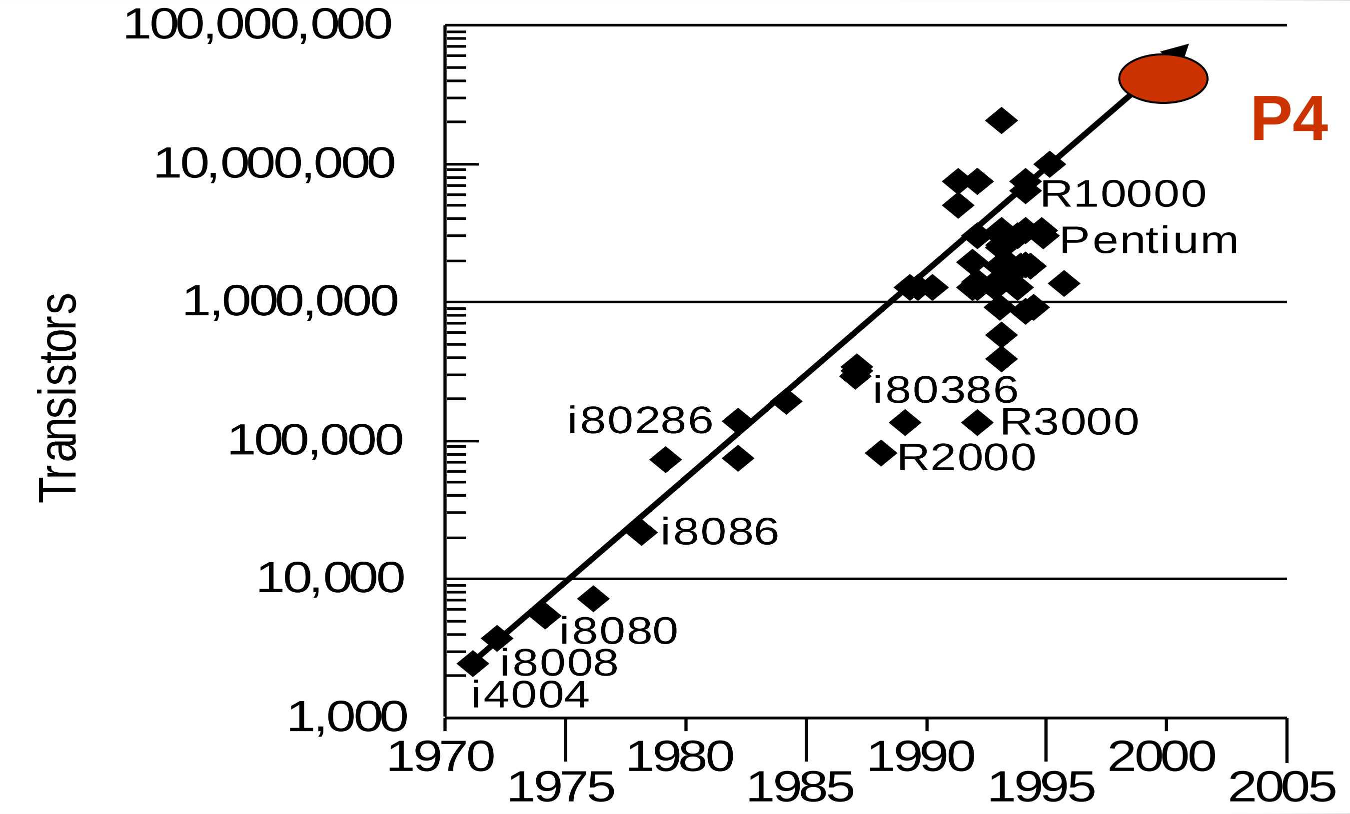
- Intel Pentium 4
  - August 27, 2001
  - 32-bit architecture, 1.4 GHz (now 3.08), 42M transistors (now 55+M), 0.18-micron technology (now 0.09)



Technology Directions: SIA Roadmap

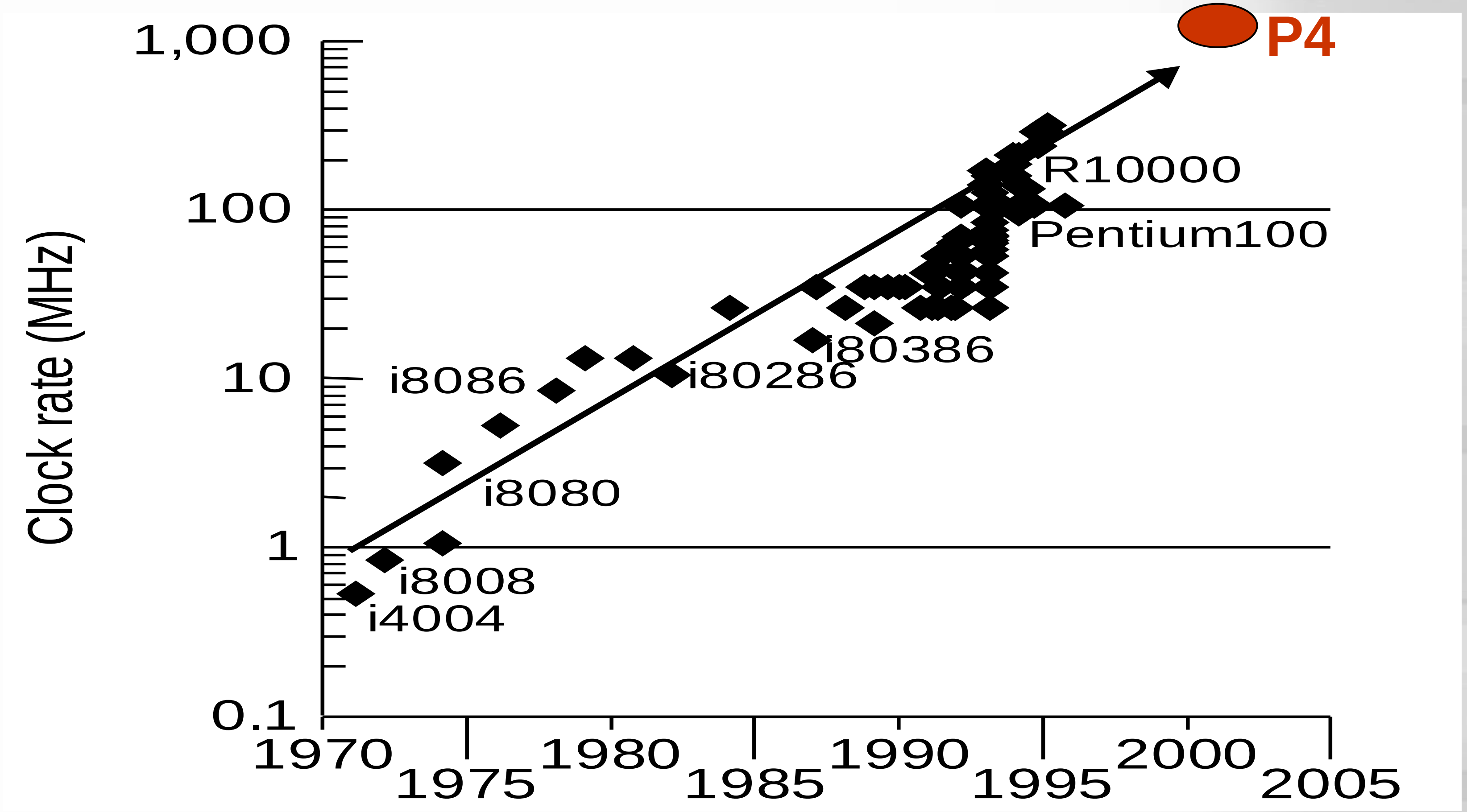
Year	1999	2002	2005	2008	2011	2014
Feature size (nm)	180	130	100	70	50	35
Logic trans/cm^2	6.2M	18M	39M	84M	180M	390M
Cost/trans (mc)	1.735	.580	.255	.110	.049	.022
#pads/chip	1867	2553	3492	4776	6532	8935
Clock (MHz)	1250	2100	3500	6000	10000	16900
Chip size (mm^2)	340	430	520	620	750	900
Wiring levels	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

Transistor Count Growth Rate



Moore’s Law

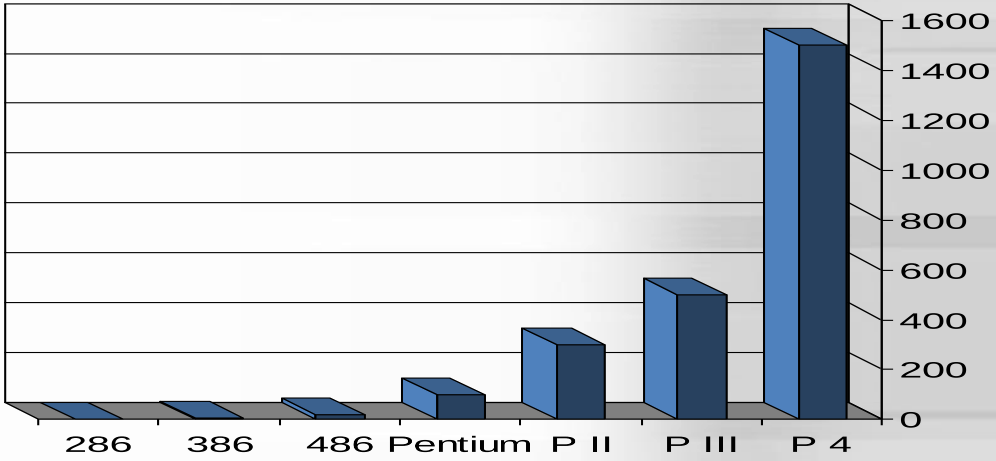
Clock Frequency Growth Rate



- 30% per year

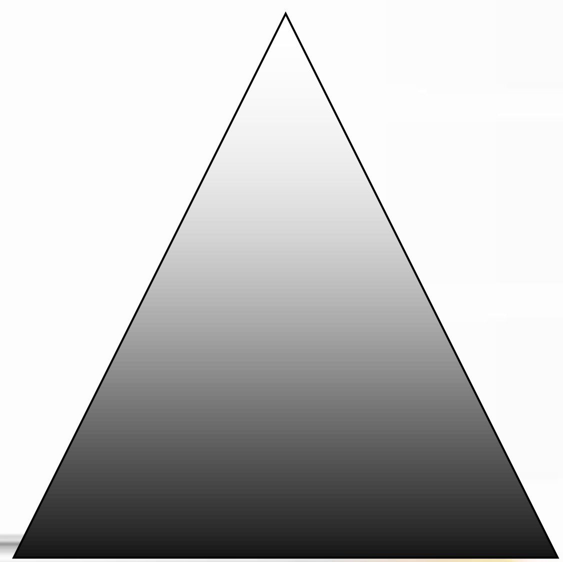
Performance Trends

Year	Proc.	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



Storage

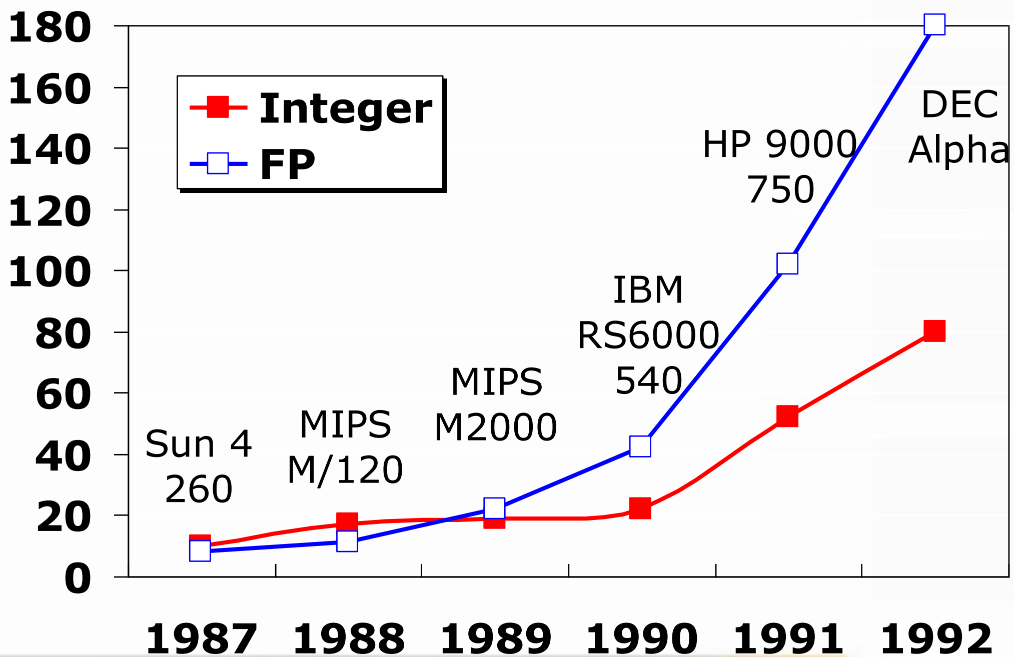
- Divergence between memory capacity and speed more pronounced
  - Capacity increased by 1000x from 1980-95, speed only 2x
  - Gigabit DRAM by c. 2000, but gap with processor speed much greater
- Larger memories are slower, while processors get faster
  - Need to transfer more data in parallel
  - Need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

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



Trends & Challenges

- 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

More on Challenges

- Scalability
  - billions of small devices
  - performance
- Availability
  - hardware changes
  - system upgrade
  - failures
  - code enhancements
- Fault tolerance

Outline

- Administration
- Technology Trends
- Embedded systems
  - What are they?
  - Where do we find them?
  - Structure and Organization
  - Software Architectures

What are Embedded Computer Systems

- An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions
- Main Characteristics
  - Usually embedded as a part of a complete device that serves a more general purpose (e.g., in car or in MP3 player)
  - Usually heavily optimized for the specific tasks, reducing cost of the product or reducing the size or increasing the reliability and performance
  - Often with real-time computing constraints that must be met, for reasons such as safety (e.g., anti-block systems) and usability (e.g., video consoles)
  - Range from low-end 4-bit microcontrollers to high-performance multiple processor cores on a single chip
  - Software written for embedded systems is often called firmware, and is usually stored in read-only memory or Flash memory chips rather than a disk drive

Early History of Embedded Systems

- Apollo Guidance Computer
  - One of the first publicly recognized embedded systems
  - Developed by Charles Stark Draper at the MIT Instrumentation Laboratory
- Autonetics D-17 (1961)
  - Guidance computer for the Minuteman missile
- Intel 4004 (1971), first microprocessor
  - Used in calculators
- Automobiles used microprocessor-based engine controllers (1970's)
  - Control fuel/air mixture, engine timing, etc.
  - Multiple modes of operation: warm-up, cruise, hill climbing, etc.
  - Provides lower emissions, better fuel efficiency

Modern Embedded Systems

- Modern Microcontrollers: (mid 1980s)
  - Microprocessors that include I/O devices and on-chip memory on a chip
- Digital Signal Processors (DSP):
  - Microprocessors optimized for digital signal processing
- Typical embedded processor word sizes: 8-bit, 16-bit, 32-bit

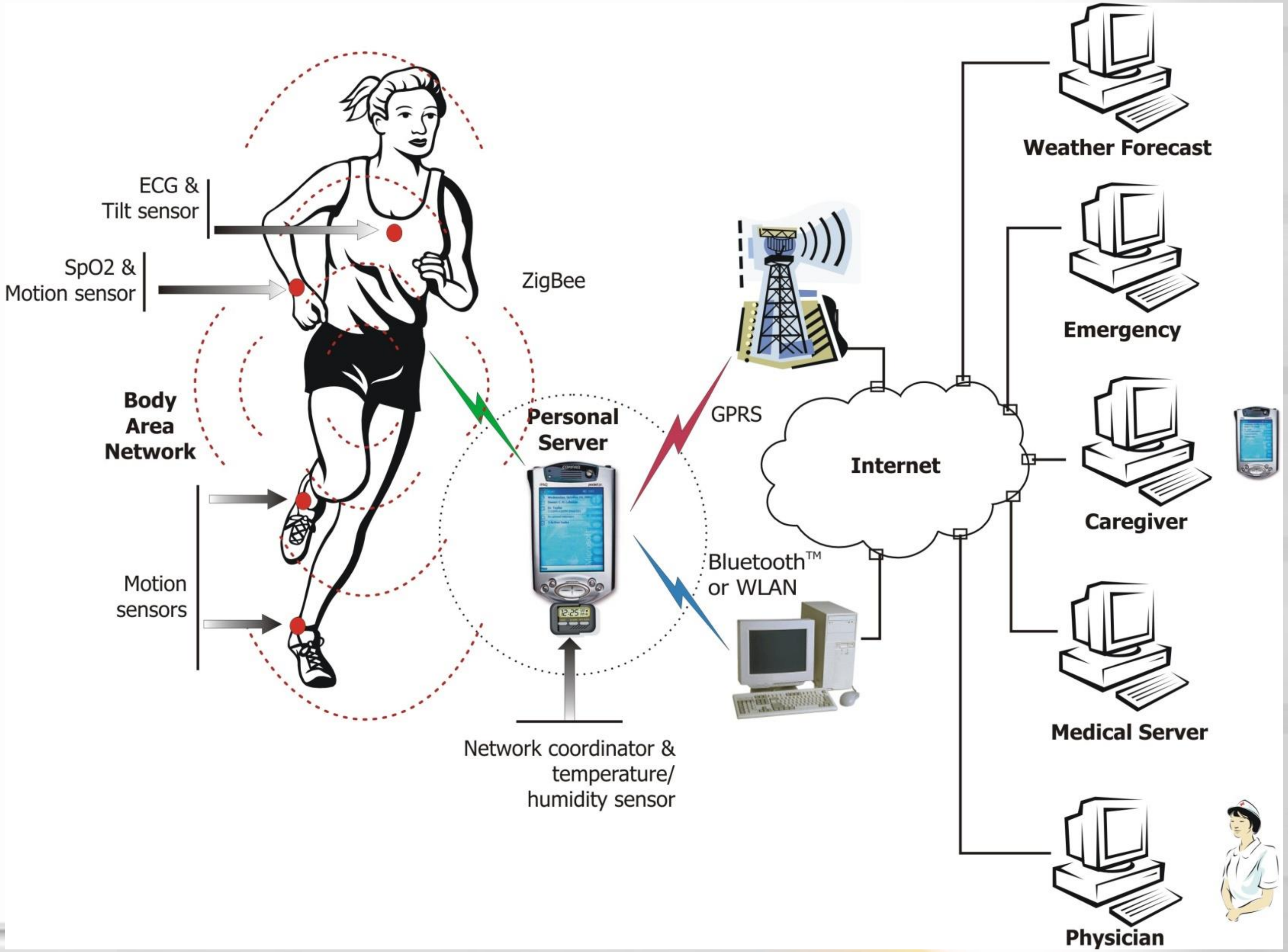
Embedded Systems Applications

- Telecommunication equipment: telephone switches, voice and data network bridges and routers
- Consumer electronics: MP3 players, DVD players, digital cameras, GPS receivers, game consoles, …
- Home appliances: microwave ovens, dishwashers, washers, …
- Transportation systems: aviation electronics (avionics), vehicle electronics (to increase efficiency and safety, reduce pollution, …)
- Medical electronics: health monitors, medical imaging (PET, SPECT, CT, MRI)

Future Applications

- Deeply embedded into the environment
  Wireless Sensor Networks
- Applications
  - Health Monitoring
  - Smart Transportation Systems
  - Smart Roads
  - Habitat Monitoring
  - Military
  - …
- Wireless Sensor Networks @ UAHuntsville
  - TinyHMS and SVEDECs

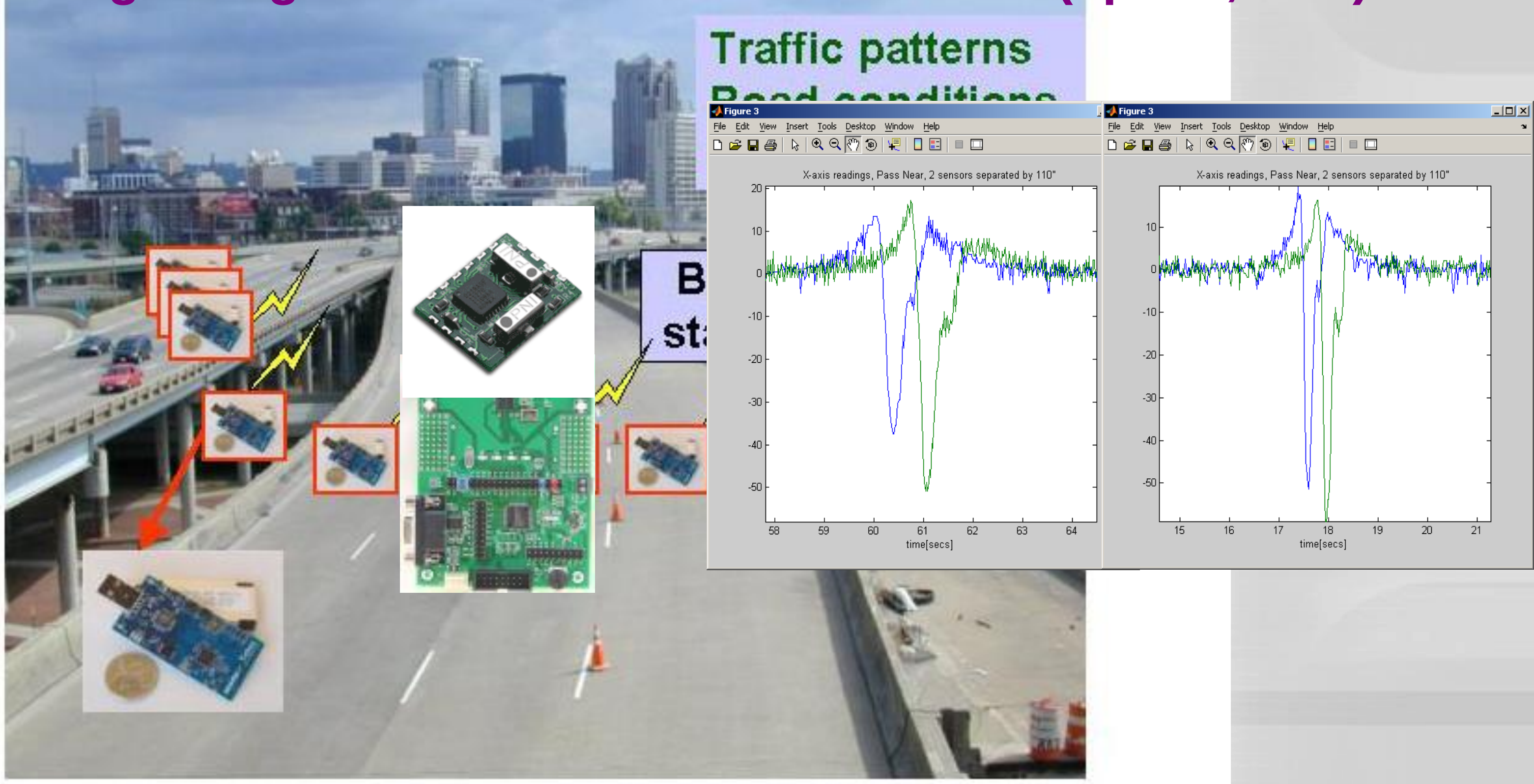
Ubiquitous Health Monitoring



SVEDECs

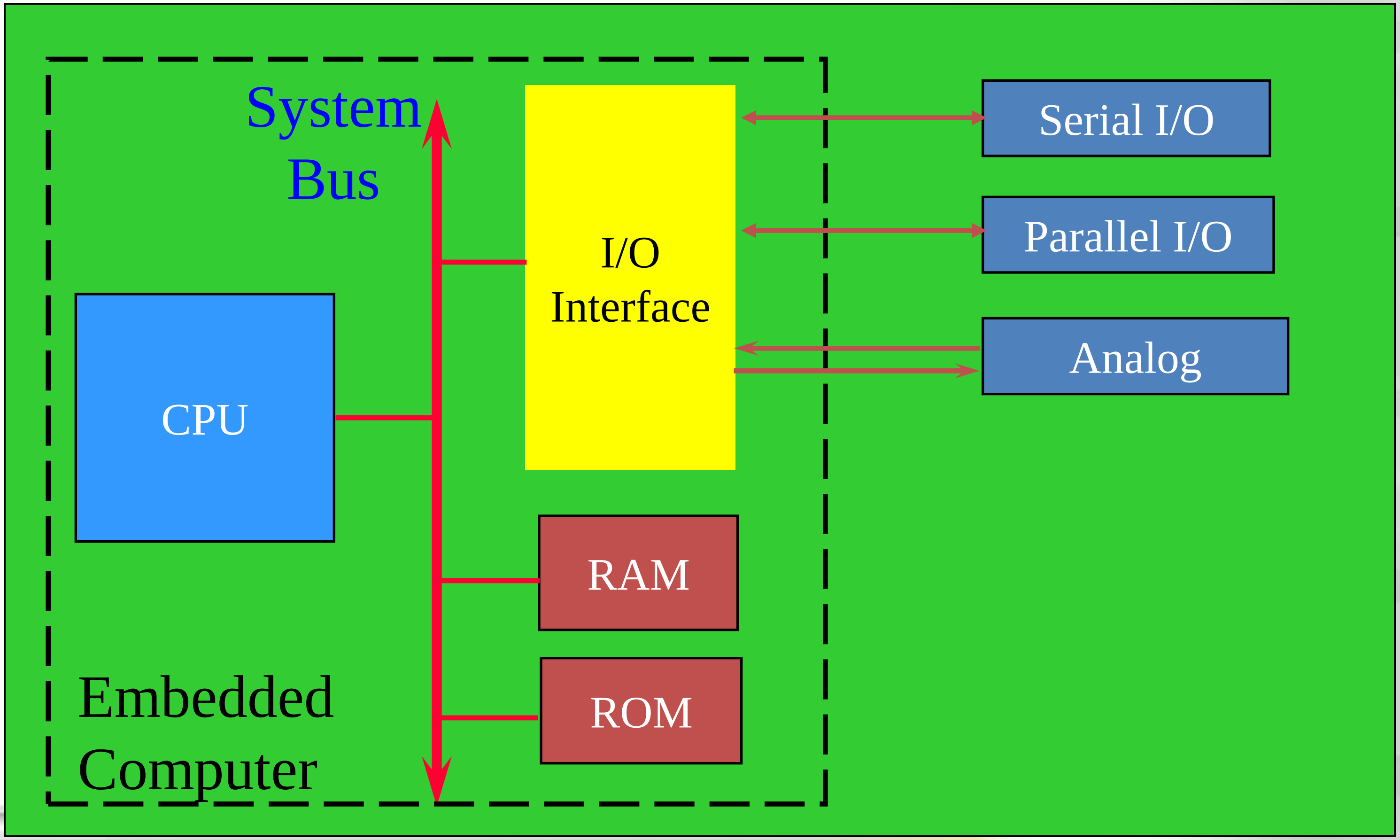
Traffic Monitoring Using TMotes

Vehicle Detection (speed, size)



Embedded Systems Organization

- 4 major components: CPU, Memory, System Bus, and I/O Peripherals



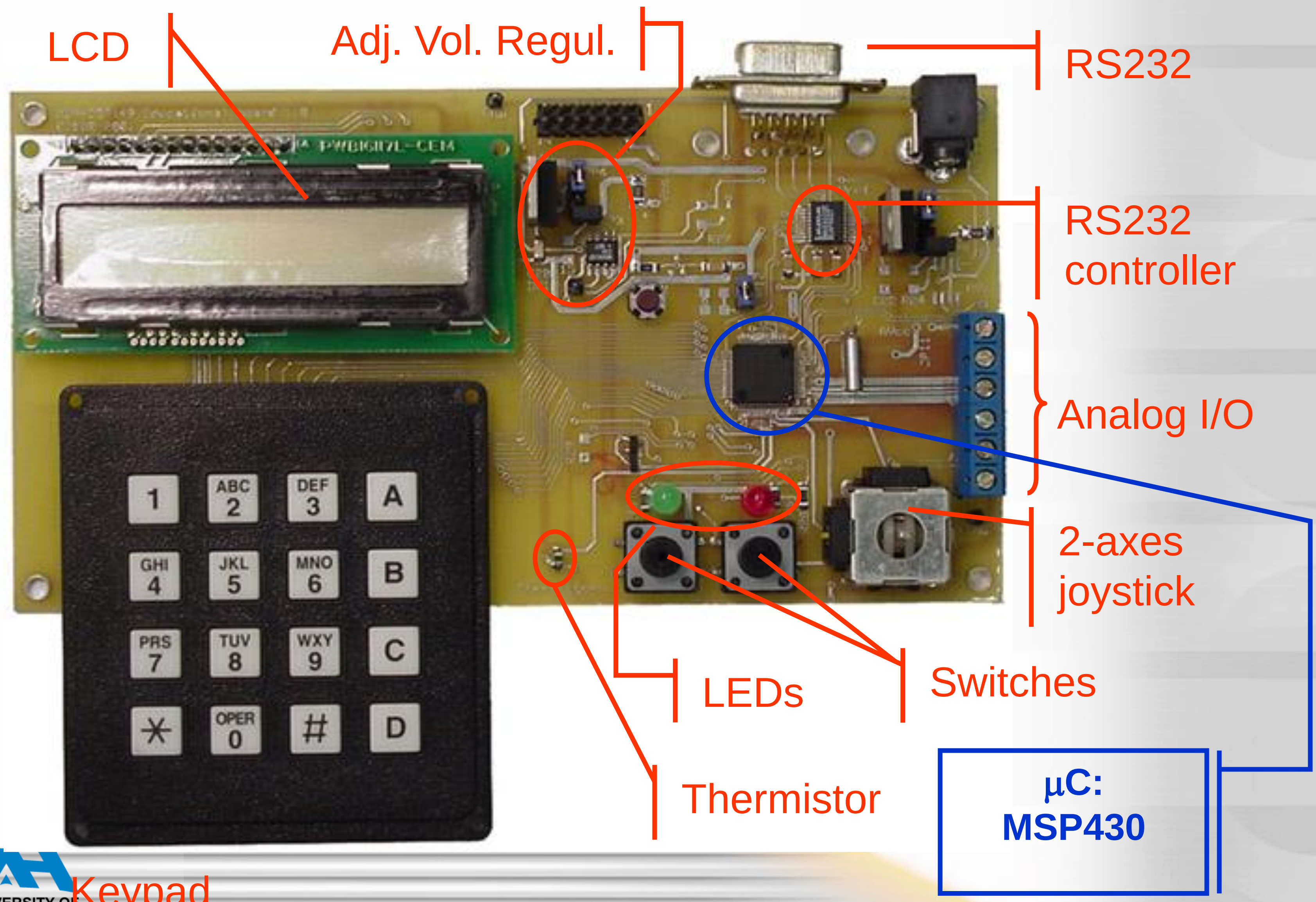
CPUs

- Unlike the personal and server computer markets the embedded processors are fairly diverse featuring
  - Von Neumann as well as Harvard architectures
  - RISC as well as non-RISC and VLIW;
  - Word lengths from 4-bit to 64-bits and beyond (mainly in DSP processors) although the most typical remain 8/16-bit.
  - A large number of different variants and shapes, many of which are also manufactured by several different companies
  - Common architectures are: 65816, 65C02, 68HC08, 68HC11, 68k, 8051, ARM, AVR, AVR32, Blackfin, C167, Coldfire, COP8, eZ8, eZ80, FR-V, H8, HT48, M16C, M32C, MIPS, MSP430, PIC, PowerPC, R8C, SHARC, ST6, SuperH, TLCS-47, TLCS-870, TLCS-900, Tricore, V850, x86, XE8000, Z80, etc.
- Typically embedded CPUs are integrated together with memories and I/O peripherals on a single chip
  to reduce the cost and size and increase reliability

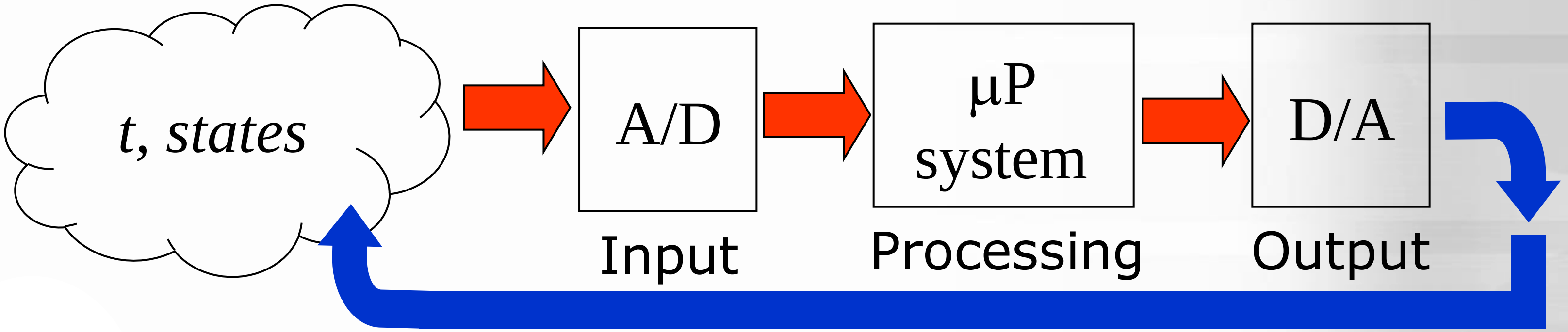
I/O Peripherals

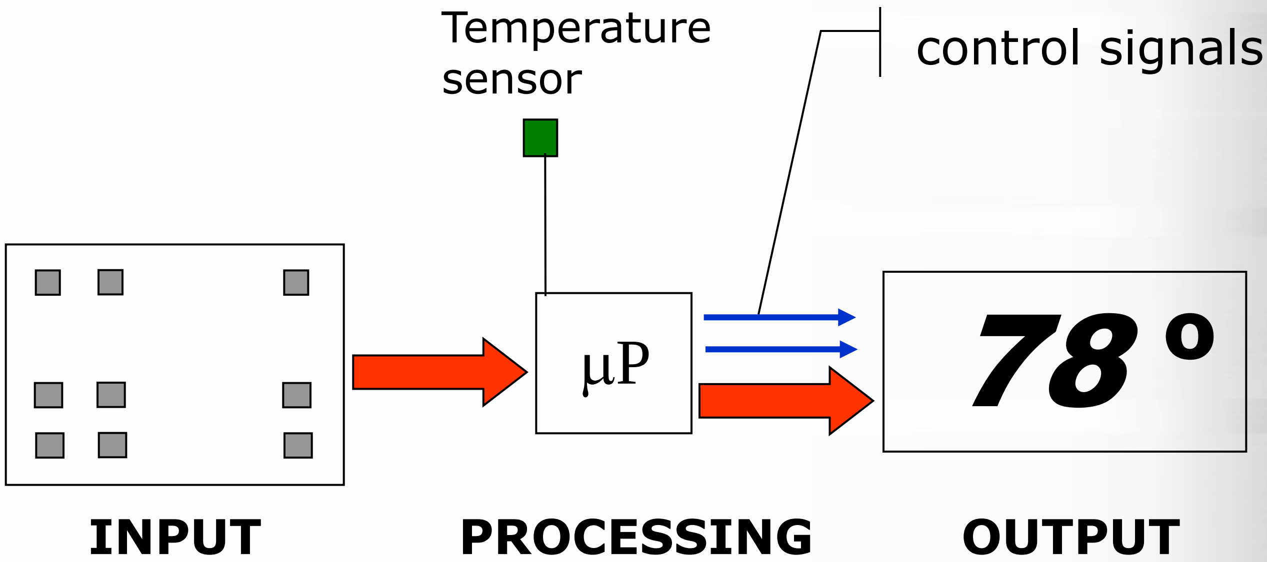
- Embedded Systems talk with the outside world via peripherals, such as:
  - Serial Communication Interfaces (SCI): RS-232, RS-422, RS-485 etc
  - Synchronous Serial Communication Interface: I2C, JTAG, SPI, SSC and ESSI
  - Universal Serial Bus (USB)
  - Networks: Ethernet, Controller Area Network, LonWorks, etc
  - Timers: PLL(s), Capture/Compare and Time Processing Units
  - Discrete IO: aka General Purpose Input/Output (GPIO)
  - Analog to Digital/Digital to Analog (ADC/DAC)

A Microcontroller-Based System: An Example



Data Flow


t, states
A/D
μP
system
D/A
Input
Processing
Output


Temperature
sensor
control signals
μP
78 o
INPUT
PROCESSING
OUTPUT

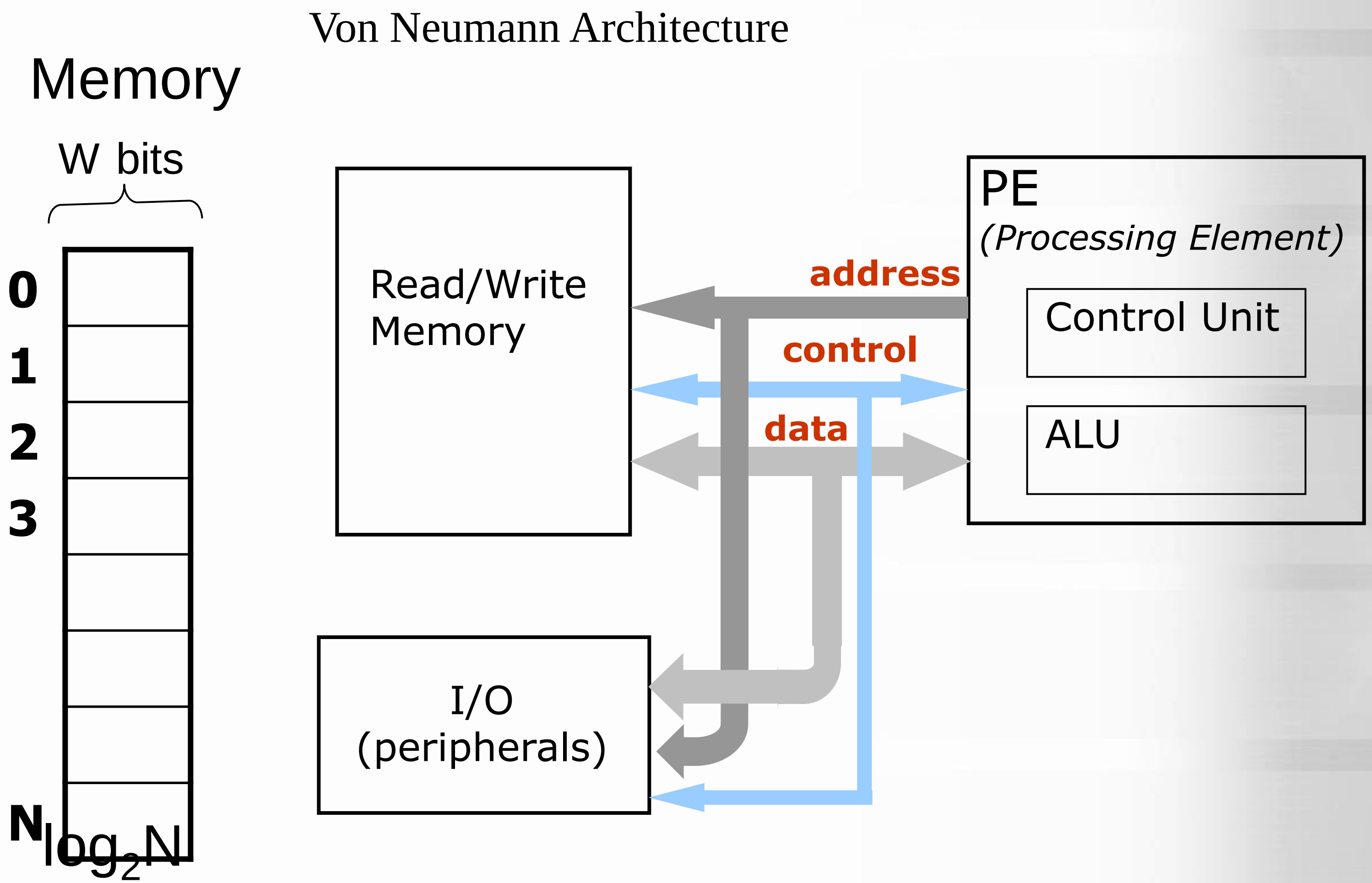
THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

Backup Slides

Von Neumann Architecture

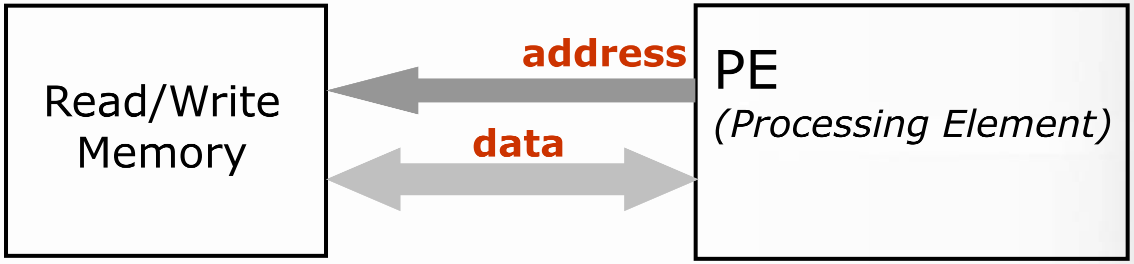
- Processing Elements
  - sequential execution
- Read/Write Memory
  - linear array of fixed size cells
  - Data and instruction store
- I/O unit
- Address/Data/Control bus

Von Neumann Architecture



Von Neumann vs. Harvard

Von Neumann Architecture



Harvard Architecture