CPE 323 Introduction to Embedded Computer Systems: An Introduction

Electrical and Computer Engineering
University of Alabama in Huntsville

Aleksandar Milenkovic

milenka@ece.uah.edu

http://www.ece.uah.edu/~milenka

Outline

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

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 ENG 106
  - Lab policies
  - Lab assistants

How to Succeed in CPE 323 / CPE 323 DOs (1 – 5)

- **1. Work hard, have fun ☺ ... because**
  - It’s better to have fun than not to have fun ... (have a can do, will do, done attitude)
  - Many students landed jobs thanks to skill gained in this course
  - Important course for your senior design
- **2. Manage your time well**
  - Lectures, labs, quizzes, …
- **3. Attend classes & make maximum use of class time**
  - Skim through lecture notes in advance
  - Have clean notes downloaded if you want to 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 may offer non-mandatory recitation sessions

How to Succeed in CPE 323 / DOs (6 – 10)

- **6. Make the best from your time in laboratory**
  - Learning by doing; Come prepared (read tutorials, watch extra videos)
  - **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 (perfect tool for exam preparation)
  - Start with your homework as soon as it is posted
  - **I hear and I forget, I see and I remember, I do and I understand**
- **8. Study in groups; Collaboration=YES, Cheating=NO**
  - Exams tell me whether you were 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**
  - Develop real-world problem solving skills
  - Become proficient in embedded systems (they are everywhere)
  - Your professional career is what matters

CPE 325 DOs

- **1. Read the lab tutorial in advance; test all demo examples**
- **2. Understand assignment(s): a well-defined problem is a half-solved one**
- **3. Develop a plan for solving the problem**
- **4. Give a try to design solution (pen & paper)**
- **5. Write the code**
- **6. Document your code**
- **7. Test your solution (try different inputs, corner cases)**
- **8. Demonstrate your program executing to TA**
- **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

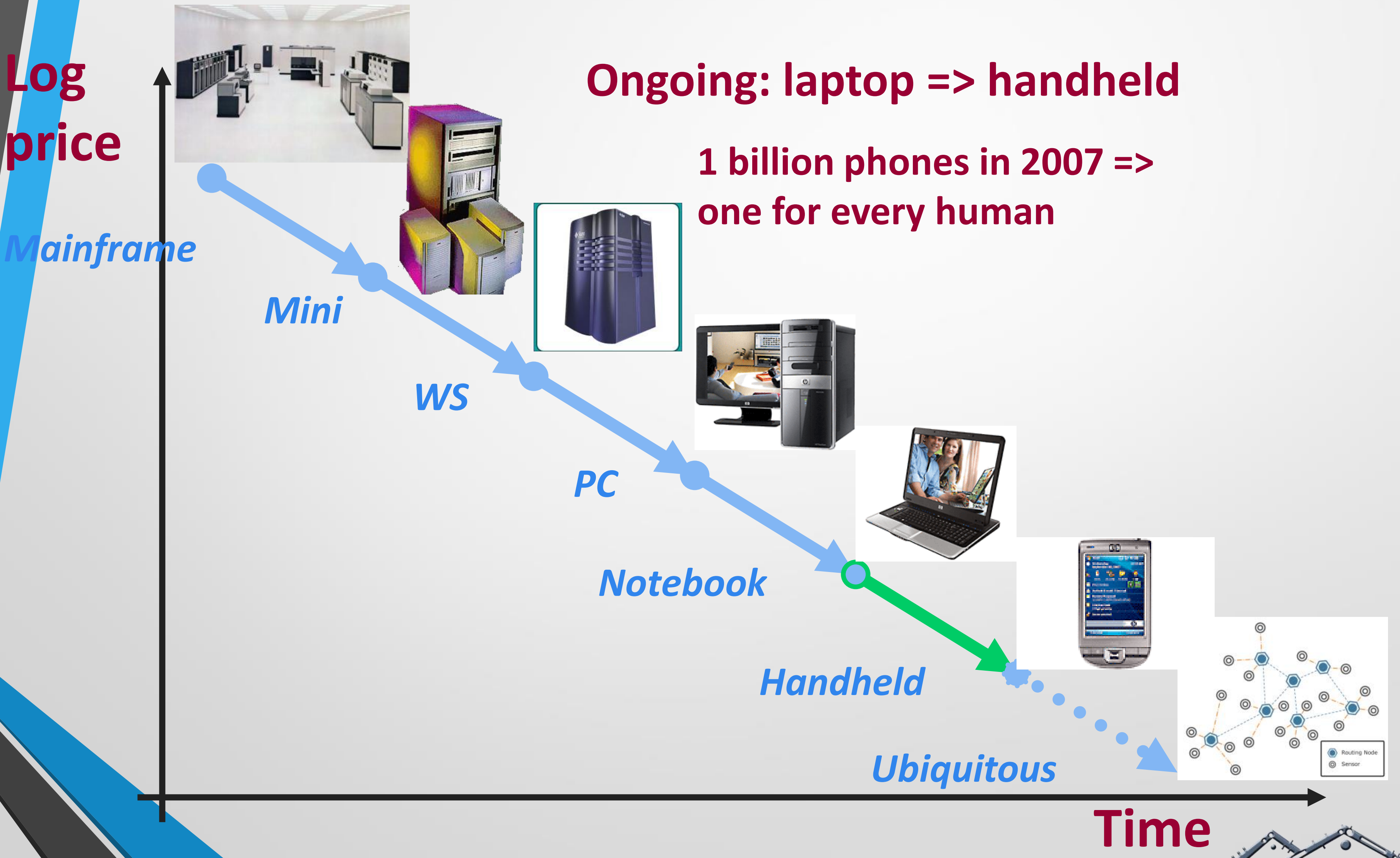
How to fail in CPE 323: an incomplete list

- 1. Do not bother attending the classes
- 2. Annoy your classmates (talk in classes, watch movies, sleep, leave in middle of class, …)
- 3. Do not bother studying (reading slides, readers, …)
- 4. Start your assignments one hour before the due date
- 5. Submit my solutions of homeworks from prior years as your work
- 6. Take code from your friends and submit as your lab solution
- 7. Do not bother to take practice quizzes
- 8. Setup hidden cheating communication channels during quizzes and exams
- 9. Do not bother adjusting your plan, ignore warning signs

History of Computing

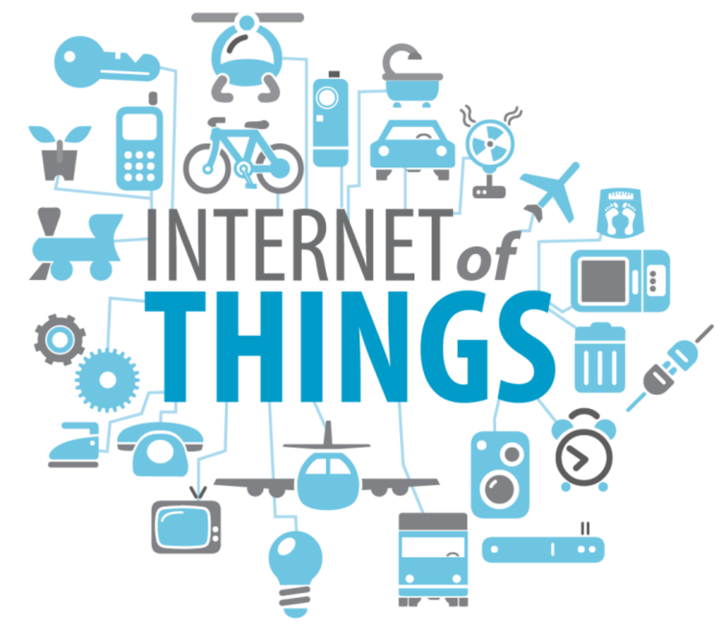
Ongoing: laptop => handheld

1 billion phones in 2007 => one for every human

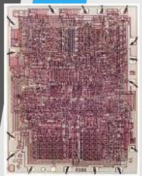


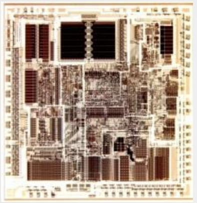
Internet of Things

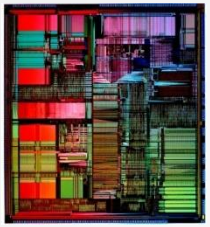
- The Internet of Things (IoT) is a scenario in which objects, animals or people are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. IoT has evolved from the convergence of wireless technologies, micro-electromechanical systems (MEMS) and the Internet.

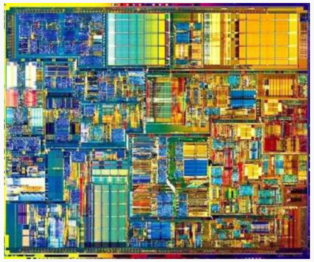


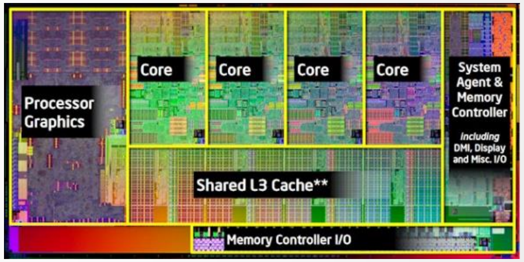
Computing (R)Evolution







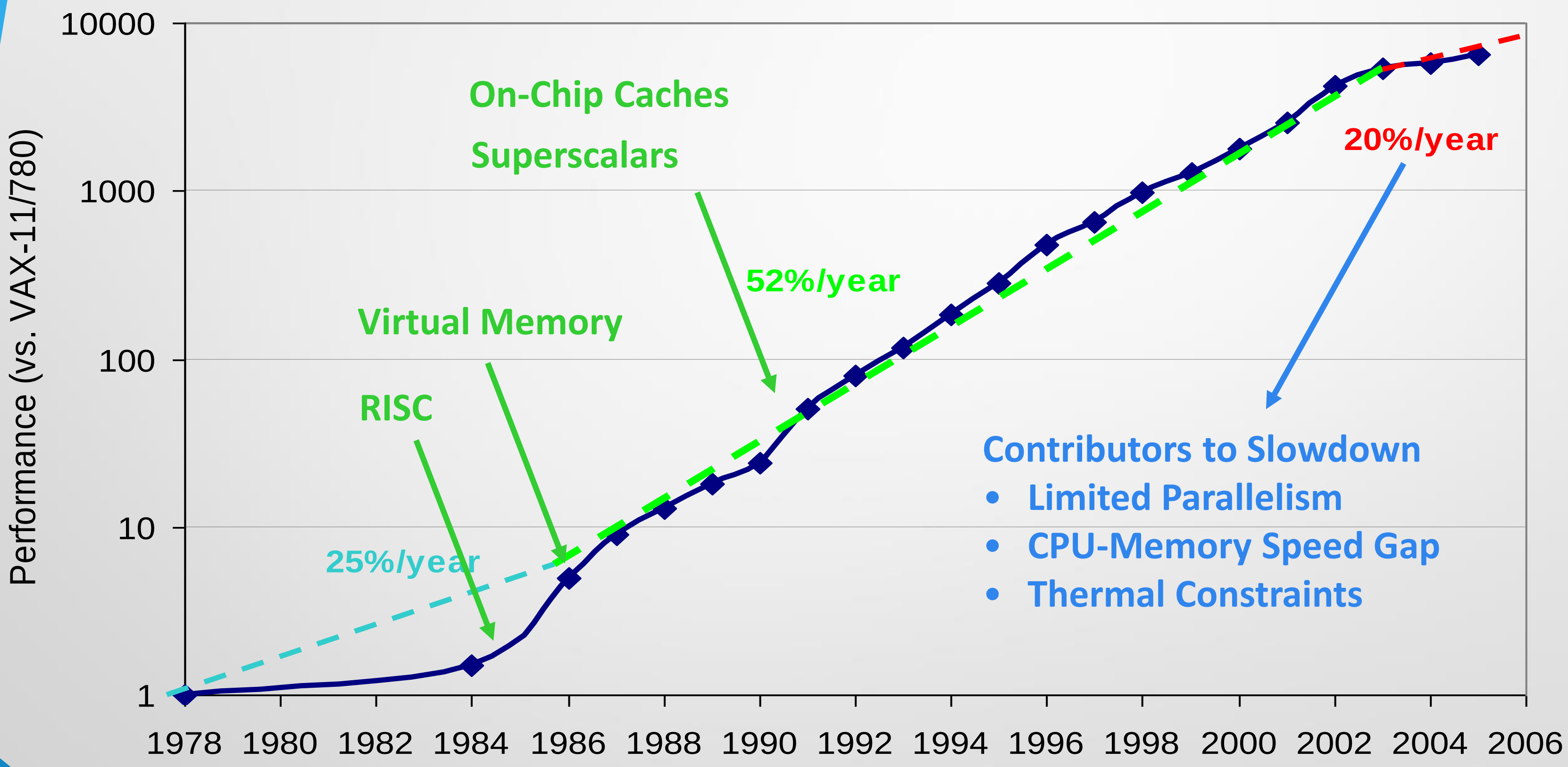




Intel 4004	**Intel 20286**	**Intel Pentium**	**Intel Pentium 4**	**Intel Sandy Bridge**
Y 1971	Y 1982	Y 1993	Y 2001	Y 2011
C 108KHz	C 6 MHz	C 60 MHz	C 1.4 GHz	C 3.8 GHz
T 2,300	T 134 K	T 3.1 M	T 55 M	T 1 B
Th 10μm	Th 1.5 μm	Th 0.8 μm	Th 0.18 μm	Th 0.032 μm (32 nm)
A 4-bit	A 16-bit	A 32-bit	A 32-bit	A 64-bit
AM 640B	AM 16 MB	AM 4 GB	AM 4 GB	AM 2^64 B

- Technology trends
  - 2x #transistors on a chip every 18-24 months
  - 4x memory capacity every 3-4 years
  - 2x disk capacity every year
- Applications: ever-increasing demands for faster and cheaper computing

Performance Trends



Contributors to Slowdown

- Limited Parallelism
- CPU-Memory Speed Gap
- Thermal Constraints

- VAX : 25%/year 1978 to 1986
- RISC + x86: 52%/year 1986 to 2002
- RISC + x86: 20%/year 2002 to present

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

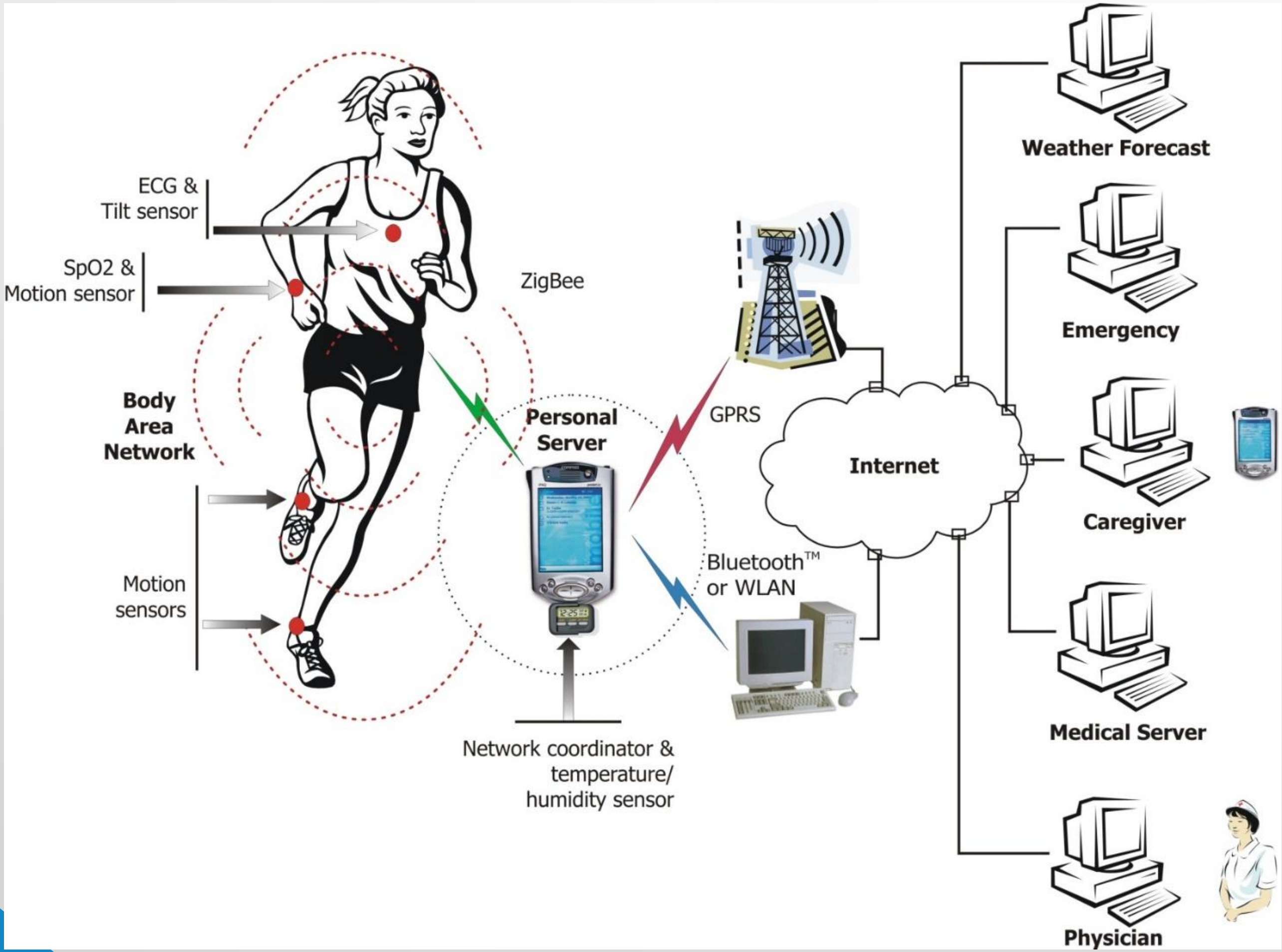
- Telecommunication equipment: telephone switches, voice and data network bridges and routers
- Consumer electronics: MP3 players, DVD players, digital cameras, GPS receivers, game consoles, …
- Home appliances: microwave ovens, dishwashers, washers, …
- Transportation systems: aviation electronics (avionics), 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

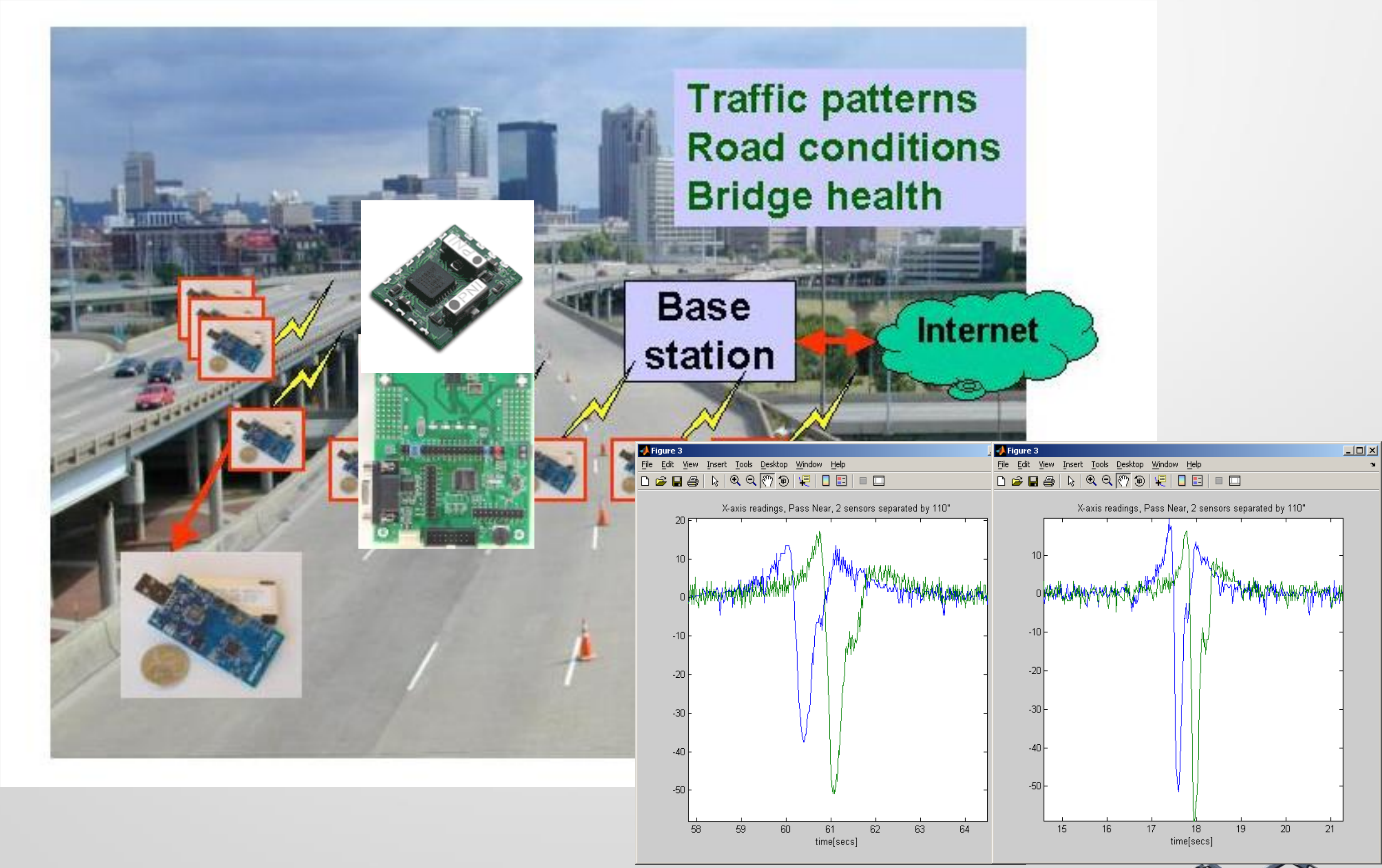
- mHealth portal: http://portal.mhealth.uah.edu



SVEDECs

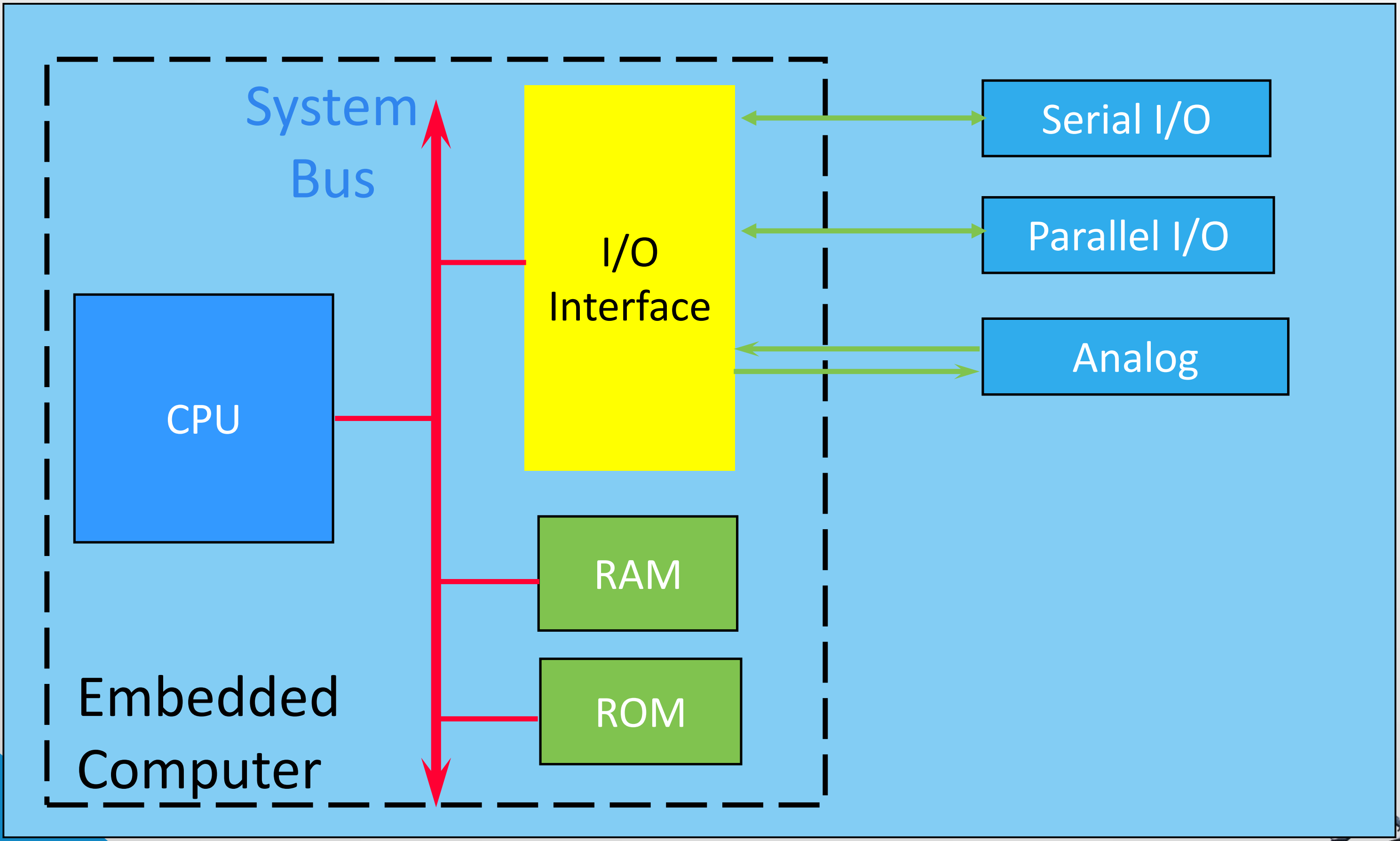
Traffic Monitoring Using TMotes

Vehicle Detection (speed, size)


Traffic patterns
Road conditions
Bridge health
Base station
Internet
Figure 3
File Edit View Insert Tools Desktop Window Help
X-axis readings, Pass Near, 2 sensors separated by 110"
20
10
0
-10
-20
-30
-40
-50
58
59
60
61
62
63
64
time[secs]
Figure 3
File Edit View Insert Tools Desktop Window Help
X-axis readings, Pass Near, 2 sensors separated by 110"
10
0
-10
-20
-30
-40
-50
15
16
17
18
19
20
21
time[secs]

Embedded Systems Organization

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



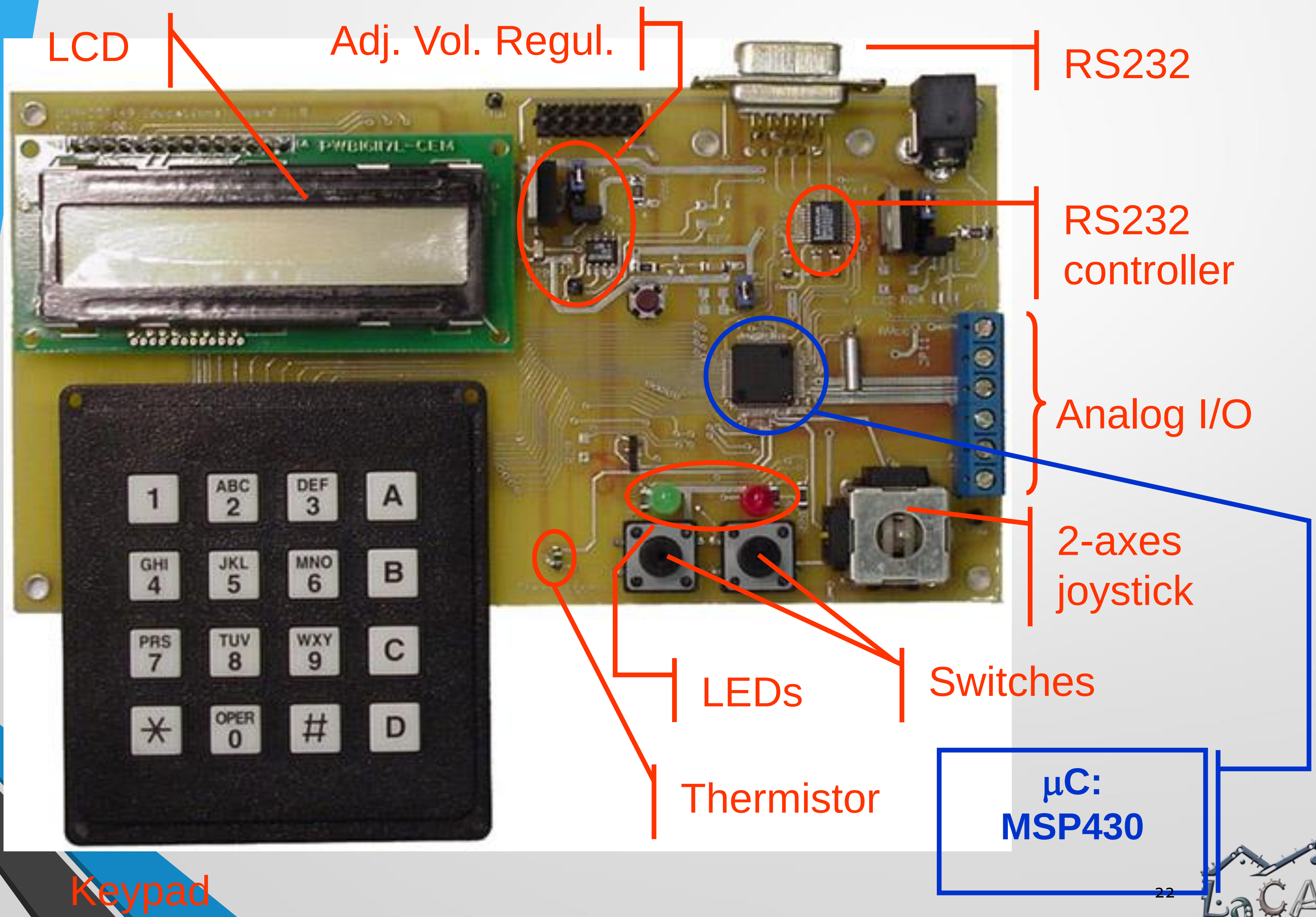
CPUs

- 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



LCD

Adj. Vol. Regul.

RS232

RS232 controller

Analog I/O

2-axes joystick

Switches

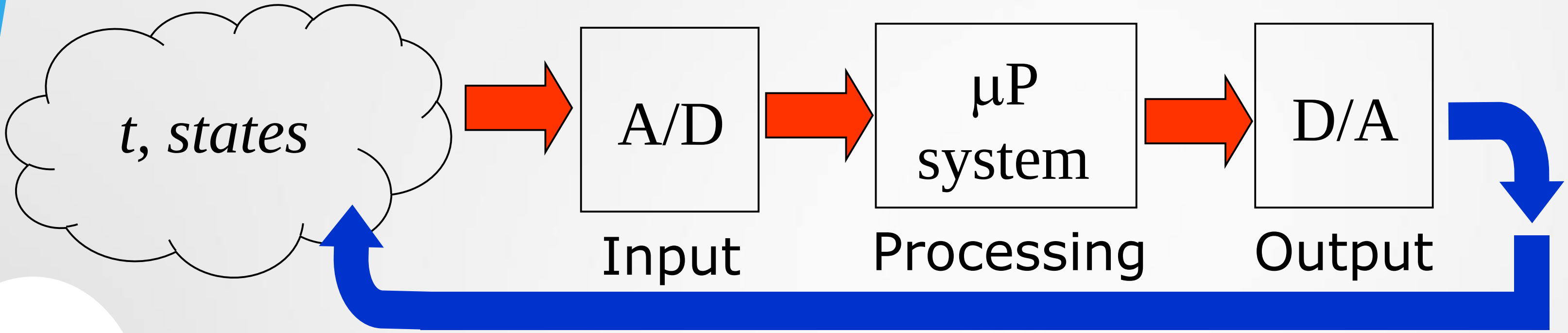
LEDs

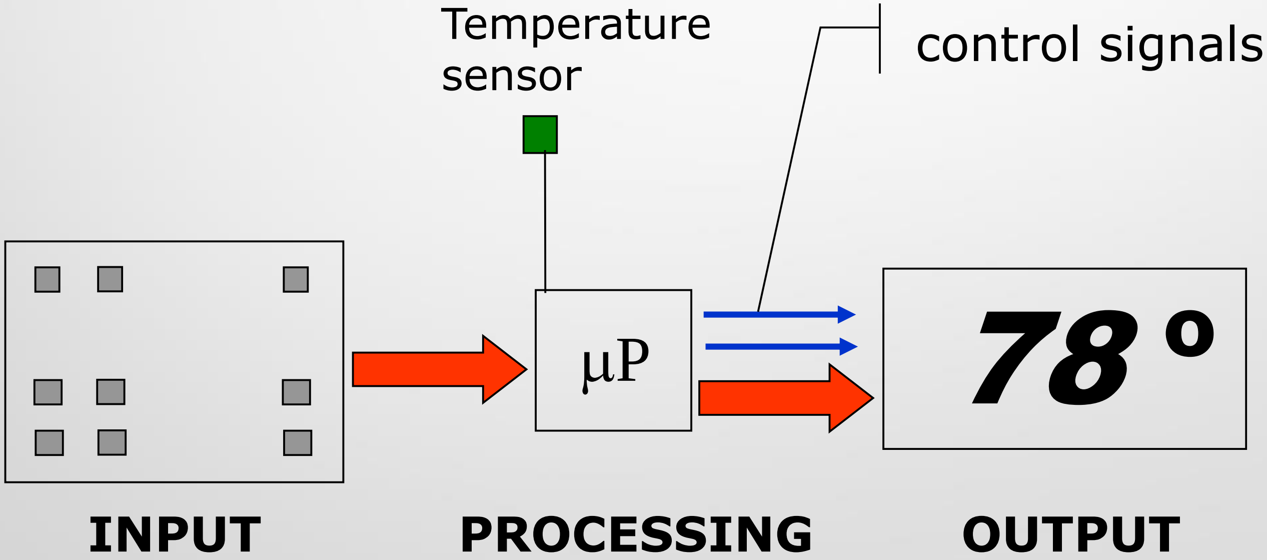
Thermistor

µC: MSP430

Keypad

Data Flow


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

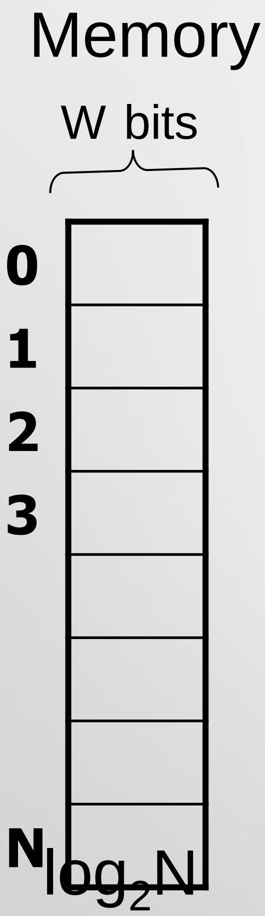

Temperature
sensor
control signals
µP
78 °
INPUT
PROCESSING
OUTPUT

Backup Slides

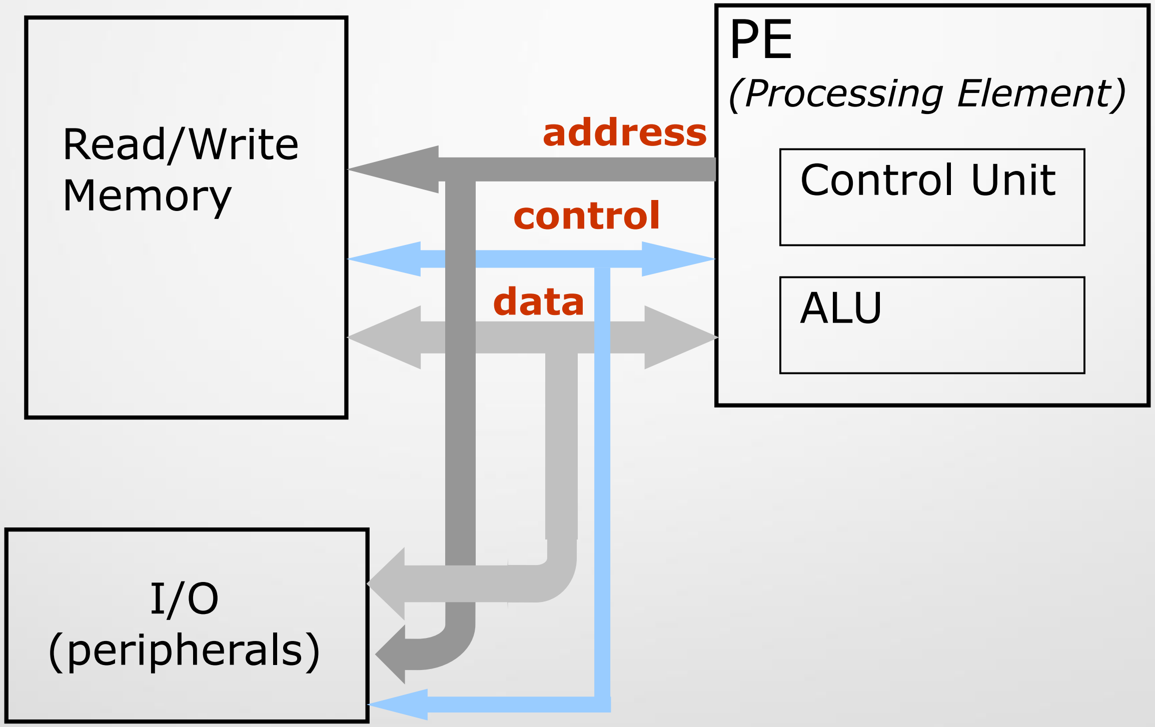
Von Neumann Architecture

- 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

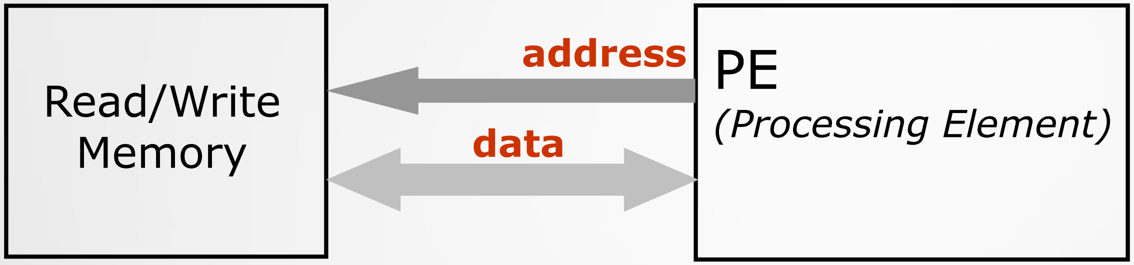


Von Neumann Architecture



Von Neumann vs. Harvard

Von Neumann Architecture



Read/Write Memory

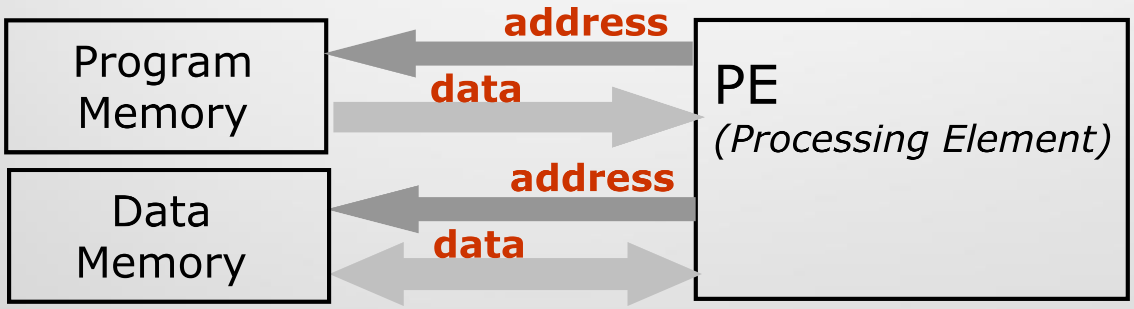
address

data

PE

(Processing Element)

Harvard Architecture



Program Memory

Data Memory

address

data

address

data

PE

(Processing Element)