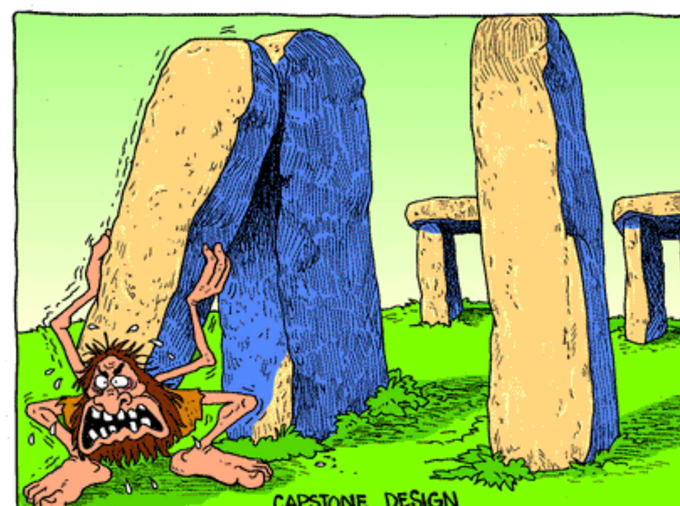
18-549: Embedded Systems Design

Lecture 1: Introduction & Logistics

Anthony Rowe
Electrical and Computer Engineering Department
Carnegie Mellon University

Overview of Lecture

- Logistics
- Grading criteria
- Policies
- Introduction to the course
- Course support: TA, office hours, piazza
- Schedule of lectures
- Schedule of project milestones
- Expectations on both sides



Welcome to 18-549

- Capstone course in embedded systems
- Builds on the fundamentals that you have acquired elsewhere
  - *18-348: Embedded System Engineering*
    - 8/16-bit microcontroller course
  - *18-349: Embedded Real-Time Systems*
    - 16/32/64-bit microprocessor course with real-time operating system
- Primary focus on a single **sizable project all semester long**
- Good news
  - Yes, the project will use real hardware!
  - Yes, you get to propose the project that you want to do
  - Teams will end up doing different projects, in all likelihood

About Me

- **What I enjoy in research**
  - ***Research:*** Networked Embedded Systems (with a focus on sensing and wireless communication)
  - ***Passion Projects***: Technology for Sustainability

- **What I enjoy in teaching**
  - 18-213: Introduction to Computer Systems
  - 18-549: This course
  - 18-648: Real-Time Embedded Systems
  - 18-848c: Networked Cyber-Physical Systems

Administrivia

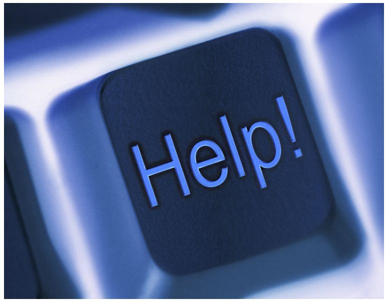
- Piazza for *all* discussions and announcements
  - www.piazza.com
- TAs
  - Adwait Dongare, Max Buevich, Ian Hartwig, Zach Rousselle, Xinwu Yang
  - Students with experience in embedded systems
- Lectures
  - Monday & Wednesdays, 12.00pm-1:20pm, MM 103
- Textbooks – None!
  - Unique course, with much of the content known only to a handful of experienced people, and not documented anywhere
  - You will leave this course understanding (through first-hand experience) the skills, critical thinking and trade-offs in designing embedded systems
- All necessary readings and handouts will be given in class and/or available online on http://www.ece.cmu.edu/~ece549

Office Hours

- Office hours for TAs
  - 2 hours per week for each TA
  - A TA will be available for you in the lab four (4) days out of the work-week
- Office hours for instructor
  - Monday & Wednesday, 1 hour after the lecture in my office (CIC 2217)
  - Design review meetings with ***me*** once every 2 weeks as part of TA meetings
    - Go over project status, difficulties, next steps, challenges, resources needed

Getting Help

- I expect an "average" CMU student will put in 12 hours/week
- If you start to see yourself exceed the number of hours greatly, then,
  - You might need additional background knowledge
  - You might be approaching the project the wrong way
  - Your project might not be well-scoped or well-defined
  - Your test-bed might not be appropriate



- Specific things I want to know about as soon as you face them
  - Being overwhelmed by the course material or the pace of the course
  - Hardware or software issues in project implementation
  - Problems with TA/instructor support or staffing
- I want to know **immediately** – please come and talk to me!
  - I want to see this course improve and flourish for many semesters to come
  - Let me know what we're doing right and what we're doing wrong

General Course Flow

- Lab 1: Learn PCB design, bring up and cross compiling
- Propose Project
- Present Proposal
- Mondays: Demo Work from last week in lab (10min)
- Wednesdays: Meet with TA for design review (20min)
- Weekly Individual Reports
- Final Demo (best project award)
- Final Presentation
- Final Report + Website

Classroom Protocol

- "Showing up is 80% of life." [Woody Allen]
- Please arrive on time; lecture begins promptly
  - We also promise to end on time
  - Handouts posted on course website before the lecture
- Questions are encouraged
  - If you don't understand, ask, because probably other students are struggling too
    - If you don't understand during class, you're unlikely to figure it out later
- There is no way to cover everything
  - If there is an interesting aspect (e.g., security, RFIDs) that we do not cover in class, feel free to incorporate it in your projects
  - I am happy to encourage projects that push the envelope, as long as you build an embedded system that is interesting and challenging

Overview of the Course

- Hands-on emphasis needed to acquire embedded-systems skills

- *Semester-long* team project to design and build a working embedded system

Course Organization

- 1 lab exercises (design and bring up of a simple PCB)
- NO mid-term exam
- NO final exam
- Project – team-based and focus of the course
  - Most of your time will probably be spent on the project and in the lab
  - Well-paced minor milestones every two weeks – *we will start right away*!
  - Phases of the project – concept, requirements, procurement, architecture, design, debugging, testing, measurement, prototyping
  - Interim demo and a final demo
- Lectures geared towards project and its milestones

Lab Protocol

- 1303 Hamerschlag Hall
- 24-hour key-card lab access
- We will give out access in the coming weeks
- Depending on the hardware that you select for your project
  - We will get a sufficient number of workstations set up with the right software

Quick Project Notes

- Work in groups of 3-4 (preferably 4)
- Open Design Project (you choose)
  - But we have to approve…
- Proposal will be a contract on what you will deliver
- ***No Arduino IDE allowed!***

- Entire ***next*** lecture on project!

Purchasing

- Each group has $700 budget
- You may use up to $200 out of pocket
- Google Purchasing Sheet Every Tuesday and Thursday by noon
- Vendors (usual suspects)
  - McMaster-Carr
  - Element 14 (aka Newark)
  - Jameco
  - Digikey
  - Sparkfun
  - Pololu
  - Mouser
  - MakerShed
  - Amazon

Other Resources

IDeATe@Hunt Collaborative Making Facility
(list to be expanded)

In the Spirit of a *Capstone* Course

- **What we WILL NOT DO in this course**
  - ☒ Cover background/prerequisite material that I expect you to have come in with
    - Either 18-348 or 18-349 material – you should have taken one of these courses before
    - Familiarity with one embedded processor in depth through one of these courses
    - Experience with embedded programming and debugging (both assembly and high-level)
  - ☒ Cover lecture material that is not relevant to the project or to the design of embedded systems in the real world

- **What we WILL DO in this course**
  - ☑ Walk through (and survive) the life-cycle -- concept to prototype -- of designing a sizable, useful embedded system
  - ☑ Encourage hands-on exploration of a variety of embedded systems issues that you have learnt mostly through lectures in other courses
    - Example: If you have learnt about real-time in an embedded systems course but have not yet had a chance to implement a time-critical system
  - ☑ Understanding first-hand (often the best way to remember) what it takes to build an embedded system when multiple constraints/needs must be satisfied

Policies on Collaboration

☑ What is okay

- Collaboration is encouraged/needed for projects within your team
  - Intended to teach you team spirit and the benefits of team work
  - Projects/demos/reports should be substantially the result of your team's efforts
- Sharing your findings on the mailing list, especially over something tricky
  - Postings from other students can be useful in getting over a tricky lab issue
  - Answering your fellow-students' questions, if you know the answer
  - As an instructor, I actually notice/appreciate individuals who help others

☒ What is not okay

- Using/borrowing code snippets (modified or otherwise) from other teams
- Letting someone in your team carry the lion's share of the load in the project
  - We will find you out during demos – in fact, demo questions will target your individual contributions to pinpoint your share of the "load" and your team's originality
- Cheating in the design reviews that you are supposed to complete individually

Some General Guidelines

- **I expect you to ….**
  - Attend classes [surprise, surprise]
  - Learn how to use embedded systems and try things out for yourself
  - Know how to find and use technical documentation
    - I will provide some of the more obvious starting points for searches
  - Do the required reading related to the lectures
  - Manage your time and stay on top of the milestones
    - In general, there will be one project milestone every 2 weeks; so, stay on track!
  - Submit your design reviews on time

- **Learning material in this course requires participation**
  - This is not a sit-back-and-listen kind of course; class participation is going to help you learn more and to get what you came here for!
  - Questions are welcome and appreciated
  - If you see something that is a problem, you need to tell us right away
    - Project issues, hardware issues, lab issues, classroom issues, ………

Grading Criteria

- Breakdown of grading
  - Labs, 18% of grade
  - Proposal and Presentations, 10% of grade
  - Final demo & Presentation, 20% of grade
  - Intermediate Demos, 25% of grade
  - Online documentation & report, 15% of grade
  - Peer review from team members, 12% of grade

- Grading Scale
  - 90% - 100% A
  - 80% - 89% B
  - 70% - 79% C
  - 60% - 69% D
  - 0 - 59% R

TASK	POINTS	DUE DATE	DESCRIPTION
0 Teaming info spreadsheet	0	1/13	Background info to facilitate team-forming
1 Platform Lab 1-3	15	2/4	PCB Design
2 Platform Lab 1-3	15	2/18	Assembly and Bring up
3 Platform Lab 1-3	15	3/2	Firmware
4 Project Proposal Document	15	2/11	Concept, specifications, and milestones
5 Project Proposal Presentation	10	2/16-2/18	In-class presentation of Project
17.1 Website check 1	6	2/20	First website check
5 First system demo	10	3/16	Second system/subsystems demonstration
6 Peer evaluation #1	10	3/5	Anonymous peer evaluation by teammates
7 Second system demo	10	3/23	Third system/subsystems demonstration
8 Third system demo	10	3/30	Fourth system/subsystems demonstration
9 Fourth system demo	10	4/6	Fifth system/subsystems demonstration
17.2 Website check 2	6	4/3	Second website check
10 Fifth system demo	10	4/13	Sixth system/subsystems demonstration
11 Sixth system demo	10	4/20	Seventh system/subsystems demonstration
12.1 Final system demo	20	4/22	Final full system demonstration
12.2 Final system demo encore	20	4/29	Final full system demonstration encore
13 Public demonstration	10	5/6	Public project description and demonstration
14 Final report	20	5/8	Comprehensive final report
15 Peer evaluation #2	10	5/9	Anonymous peer evaluation by teammates
16 Website check 3	8	5/11	Third website check
TOTAL POINTS	**250**	→	**260 minus (lowest score dropped among Tasks 3-6, 9-13)**

Schedule (1-3)

Wk.	Date	Day	Topics	Assignment/Activity	Due Date
January					
1	14	W	Lecture 1: Course Introduction	Task 0: Teaming info document	Tuesday, 1/21
2	19	M	Martin Luther King Day-No class		
	22	W	Lecture 2: Project Overview	Team formation complete Task 1: Lab 1 goes out	2/2
3	26	M	Lecture 3: PCB design		
	28	W	Lecture 4: Sensors and Actuators		
February					
4	2	M	Lecture 5: Communications	Lab 1 Due	
	4	W	Lecture 6: Low-Power Embedded		
5	9	M	Lecture 7: Embedded OS	Task 2: Lab 2 goes out	2/18
	11	W	Team Meetings	Task 4: Design concept proposal Due [document]	2/11
6	16	M	Proposal Presentation #1		
	18	W	Proposal Presentation #2	Lab 2 Due Task 3: Lab 3 goes out	3/2
	20	F	Lab: Lab 2 Demo, Website Check 1	Task 17.1: Website Check 1	2/20
7	23	M	Lab: demo Lab 2		
	25	W	Team Meetings	Task 6: System demo #1	3/16

Schedule (2-3)

March					
8	2	M	Lab: demo Lab 3	Lab 3 Due	
	4	W	Lecture 8: Ethics		
	5	Th	Peer evaluation #1	Task 7: Peer evaluation #1	3/5
9	9	M	Spring Break; No Classes		
	11	W	Spring Break; No Classes		
10	16	M	Lab: project demo #1		
	18	W	Team Meeting	Task 8: System demo #2	3/23
11	23	M	Lab: project demo #2		
	25	W	Team Meeting	Task 9: System demo #3	3/30
12	30	M	Lab: project demo #3		
April					
12	1	W	Team Meeting	Task 10: System demo #4	4/6
	3	F	Website Check 2	Task 17.2: Website Check 2	4/3
13	6	M	Lab: project demo #4		
	8	W	Team Meeting	Task 11: System demo #5	4/13

Schedule (3-3)

April					
14	13	M	Lab: project demo #5		
	15	W	Team Meeting	Task 12: System demo #6	4/20
15	20	M	Lab: project demo #6		
	22	W	Lab: Final system demo	Task 13: Final system demo	4/22
16	27	M	Course feedback session		
	29	W	Lab: Final system demo encore	Task 13: Final sys demo encore	4/29
May					
17	6	W	Public presentation (Final Exams May 4-12)	Task 14: Public presentation	5/6
	8	F	Final Report	Task 15: Final report	5/8
	9	S	Peer evaluation #2	Task 16: Peer evaluation #2	5/9
18	11	M	Website Check 3	Task 17.3: Website Check 3	5/11

What *are* Embedded Systems?

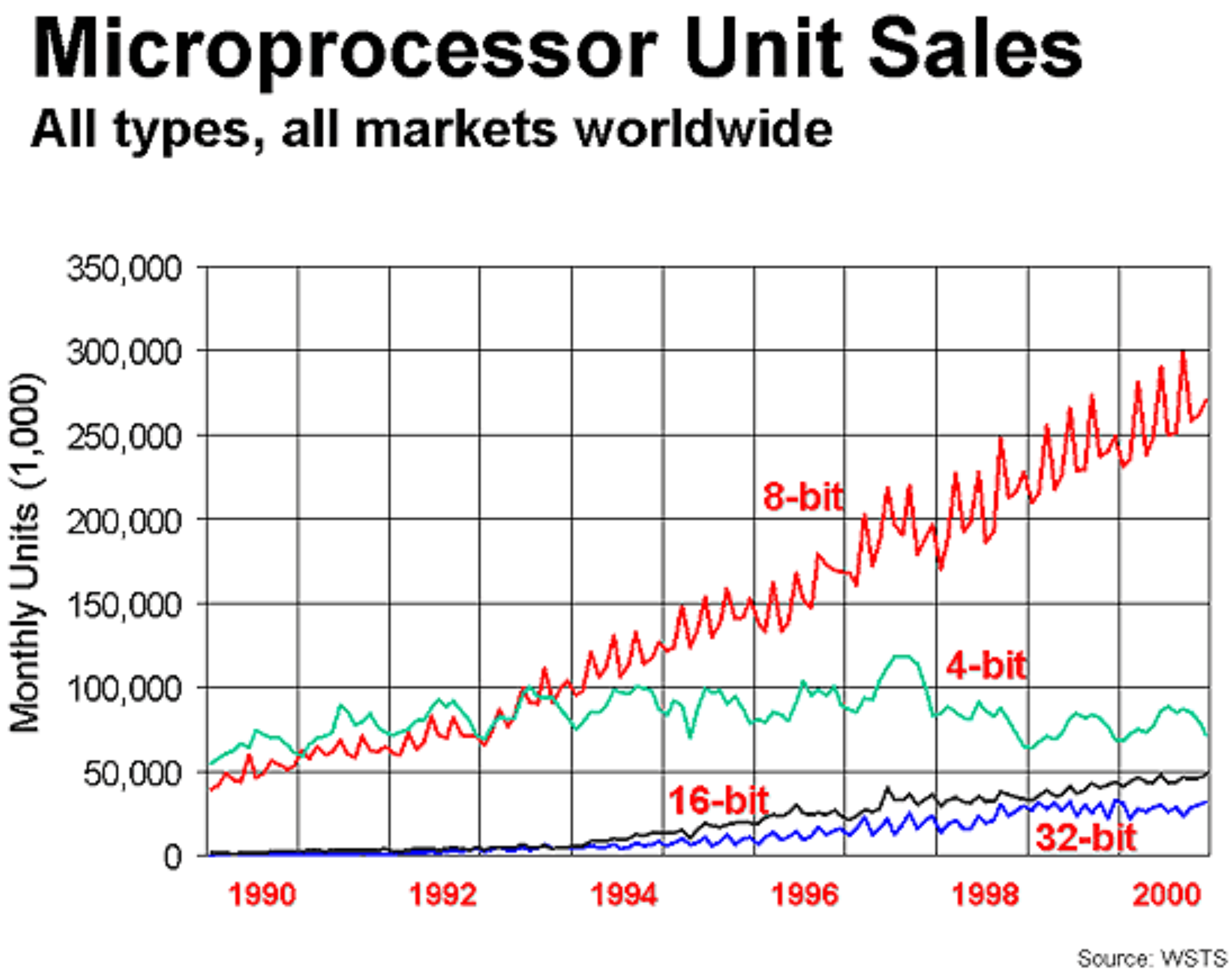
“Computer systems with dedicated functions within a larger mechanical or electrical system.”

“A computer system without a keyboard and screen.” (No longer true)

“A computer system you put in something or on something…”

Why Focus on Embedded Systems?

Over 90% of all processors are used in the embedded context



Embedded Computers *Rule* the Marketplace

- **~80 Million PCs vs. ~3 Billion Embedded CPUs annually**
  - Embedded market growing; PC market *mostly* saturated

- **Domain Experts Needed…**
  - **General Computing**
    - Set-top boxes, video game consoles, ATM, …
  - **Control Systems**
    - Airplane, Heating and Cooling System
  - **Signal Processing**
    - Radar, Sonar, Video Compression, Human-Brain interface
  - **Communication**
    - Internet, Wireless Communication, VoIP…

Embedded in Your Daily Life

- **Average American household has about 40 microprocessors**
  - 50, if you count your PC and its baggage
  - Bathroom scale with a digital readout
  - Iron that turns itself off automatically
  - Electronic toothbrush (with ~3000 lines of code)
  - Cooking range
  - Laundry machine
  - Toaster
  - Microwave
  - Furby (the toy that has more processing power than the original lunar lander)

- **Statistics from Dataquest**
  - Number of embedded processors sold in 1998 was 4.8 billion
    - Only about 120 million of these (~2.5%) were intended for PCs
  - In five years, the average number of chips in the home could grow to 280 and the number sold to 9.4 billion

Source: Honey, I programmed the blanket, Katie Hefner, *New York Times*

Typical Embedded System Challenges (1-2)

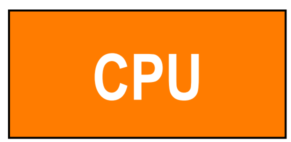
- **Small Size, Low Weight**
  - Hand-held electronics
  - Transportation applications -- weight costs money
- **Low Power**
  - Battery power for 8+ hours (laptops often last only 2 hours)
  - Limited cooling may limit power even if AC power available

Typical Embedded System Challenges (2-2)

- **Harsh environment**
  - Heat, vibration, shock
  - Power fluctuations, RF interference, lightning
  - Water, corrosion, physical abuse
- **Safety-critical operation**
  - Must function correctly
  - Must not function incorrectly
- **Extreme cost sensitivity**
  - $.05 adds up over 1,000,000 units

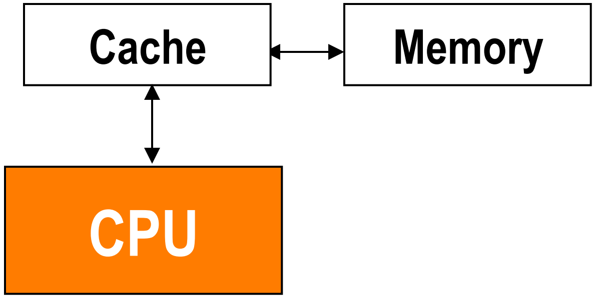
CPU: An All- Too -Common View of Computing

- Measured by: Performance



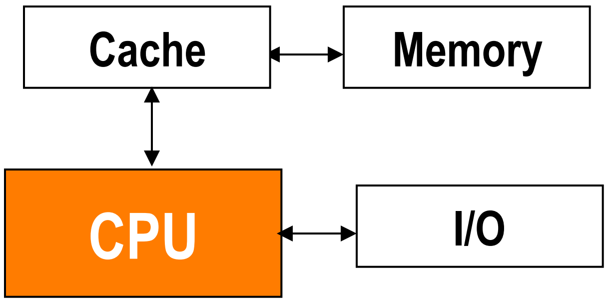
An Advanced Computer Engineer's View

- **Measured by: Performance**
  - Compilers matter too...



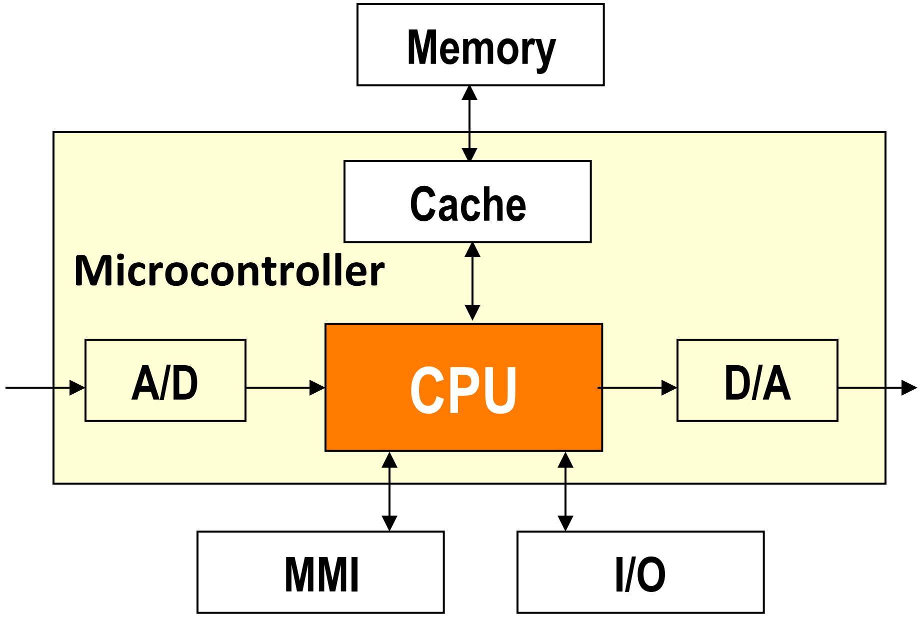
An Enlightened Computer Engineer's View

- **Measured by: Performance, Cost**
  - Compilers & OS matter



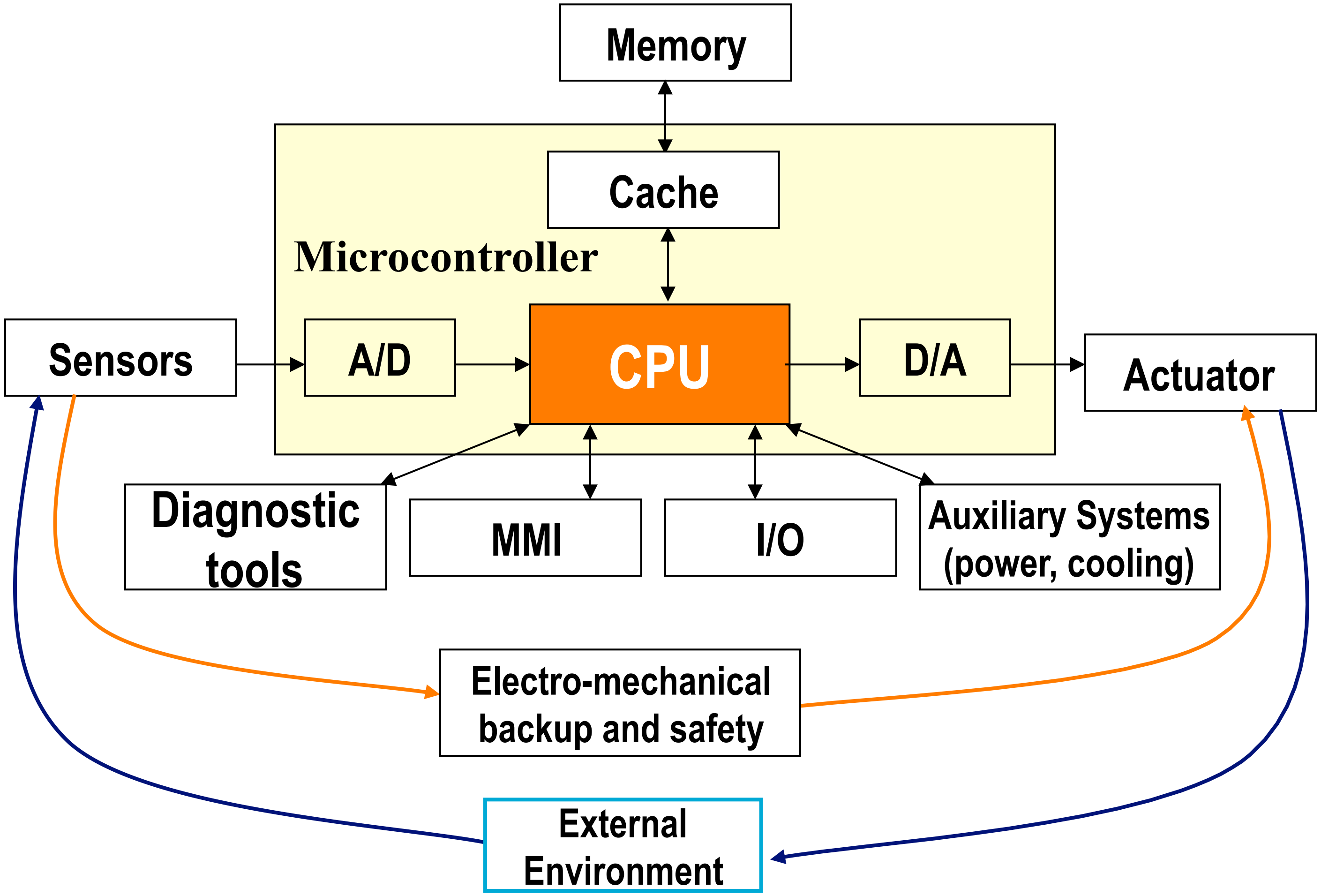
An Embedded Computer Designer's View

- Measured by: Cost, I/O connections, Memory Size, Performance



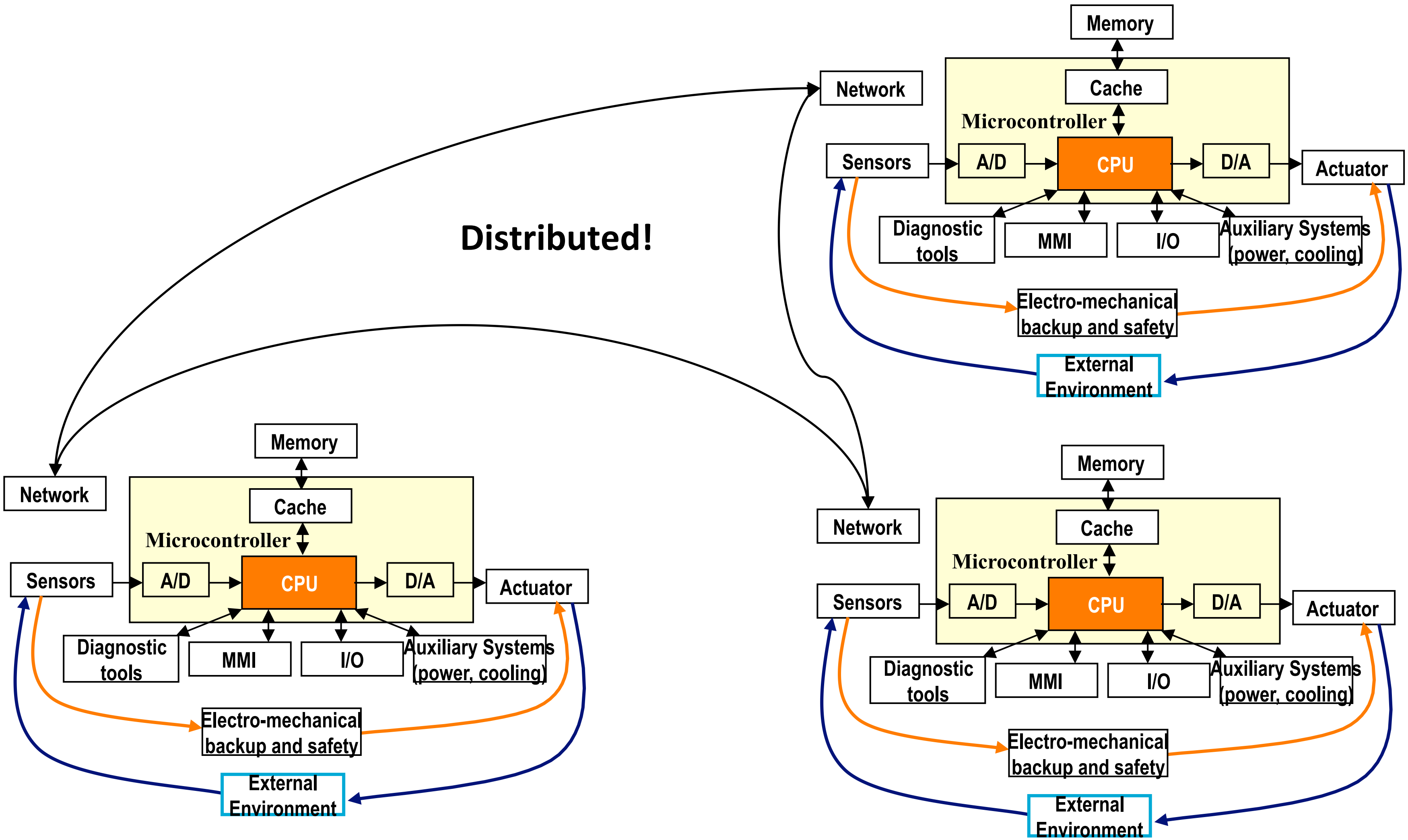
An Embedded Control System Designer's View

- Measured by: Cost, Time-to-market, Cost, Functionality, Cost & Cost.



Modern Embedded Systems View

- Measured by: Does it actually work? (and all of the other stuff)



What *are* Real-Time Systems?

System whose correctness depends on their ***temporal*** aspects as well as their ***functional*** aspects.

NOT **a really** ***FAST*** **system.**

Embedded Systems Careers

meraki

nest

TESLA MOTORS

CISCO

MICROCHIP

SPACEX

Boston Dynamics

BOSCH

GM

ATMEL

iRobot

Ford

ARM

BOEING

Green Hills SOFTWARE

intel

SAMSUNG

Microsoft

Form Groups NOW…

- Fill out and look at Google Doc
- Do some shopping
- Stay after class…

Questions?