KU THE UNIVERSITY OF KANSAS

EECS 388: Embedded Systems, Spring 2023

Lecture 1

Instructor: Tamzidul Hoque, Assistant Professor, Dept. of EECS, University of Kansas (hoque@ku.edu), office: Eaton 2038

About me

- Assistant Professor
  - University of Kansas
  - Co-director: ASTHA Labs

ASTHA

Assurance of Security and Trust for Human and HArdware (ASTHA) Lab

Faculty

Dr. Tamzidul Hoque
Assistant Professor
Department of EECS
Email: hoque@ku.edu

Dr. Sumaiya Shomaji
Assistant Professor
Department of EECS
Email: shomaji@ku.edu

Graduate Students

S M Mojahidul Ahsan
GTA, Dept. of EECS
smmojahidul.ahsan@ku.edu

Mahmudul Hasan
GRA, Dept. of EECS
m.hasan@ku.edu

Tanvir Hossain
GRA, Dept. of EECS
tanvir@ku.edu

Md. Mashfiq Rizvee
GRA, Dept. of EECS
mashfiq.rizvee@ku.edu

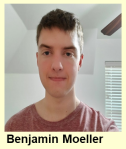
Fairuz Shadmani Shishir
GRA, Dept. of EECS
shishir@ku.edu

Asadullah Khan
Dept. of EECS
asadullahkhan@ku.edu

Shailesh Pandey
Dept. of EECS
shaileshpandey@ku.edu

Undergraduate Students

RokunuzJahan Rudro
Dept. of EECS
asadullahkhan@ku.edu



Benjamin Moeller
Dept. of EECS
benlynmoeller@ku.edu

Landen Doty
Dept. of EECS
landoty@ku.edu

Jamie King
Dept. of EECS
Jamie.king@ku.edu

Carolina Castellanos
Dept. of EECS
cmedina@ku.edu

2

Education and Work

- PhD: University of Florida, ECE, 2020
- MS: University of Toledo, EE, 2015
- BSc: Ahsanullah University of Sci. Tech., Bangladesh. 2012

- Work Experience:
  - Cisco Systems: Research Intern summer 2016 and 2017
  - University of Kansas, Asst. Professor, 2020 to present

What do we learn in EECS 388?

Theory

- Hardware/software architecture of embedded systems

Experiment

- Program a microprocessor to sense physical world and actuate upon

Overview of the topic

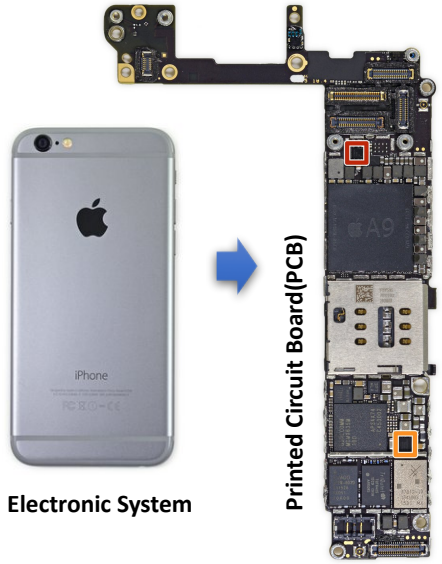
Embedded systems

Computers designed for specific purpose

6

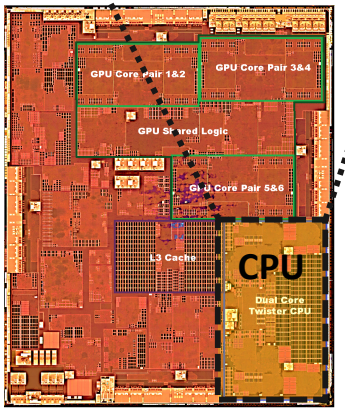
Breakdown of a Modern Embedded System

- Example: Apple iPhone



Electronic System

Integrated Circuit (IC)



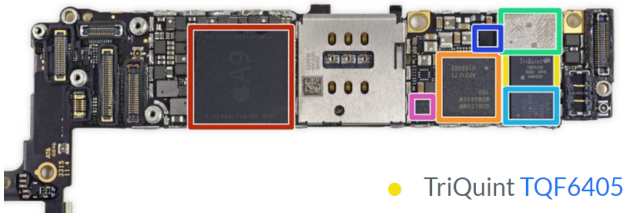
Physical Layout of the IC

- Arithmetic logic Unit (ALU)
- Control Unit
- Memory

https://www.ifixit.com/Teardown/iPhone+6s+Teardown/48170

7

Breakdown of Modern Electronic System



- Apple A9 APL1022 SoC + SK Hynix LPDDR4 RAM as denoted by the markings H9HKNNNBTUMUMR-NLH (we presume it is 2 GB LPDDR4 RAM, the same as in the iPhone 6s)
- Qualcomm MDM9635M LTE Cat. 6 Modem (vs. the MDM9625M found in the iPhone 6)
- TriQuint TQF6405 Power Amplifier Module
- Skyworks SKY77812 Power Amplifier Module
- Avago AFEM-8030 Power Amplifier Module
- Qualcomm QFE1100 Envelope Tracking IC
- Likely a InvenSense 6-axis gyroscope and accelerometer combo

https://www.ifixit.com/Teardown/iPhone+6s+Plus+Teardown/48171

Takeaway: Modern embedded systems contain a large number chips or integrated circuits (ICs)

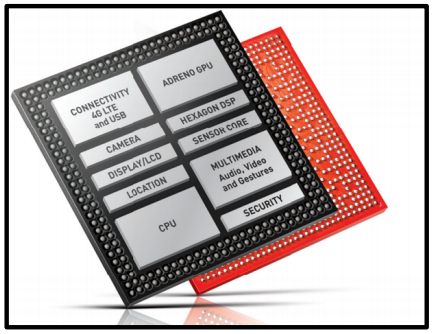
Integrated Circuits: Types

Types: Digital, Analog, and Mixed Signals

❑**Digital IC:**
- Microprocessor
- Application Specific IC (ASIC)
- Field Programmable Gate Array (FPGA)
- Memory (SRAM, DRAM, Flash)

❑**Analog IC:**
- Sensors
- Power management circuits
- Operational amplifiers

Analog-mixed-signal (AMS):
Contains both digital and analog in a single chip/IC



Example: AMS IC

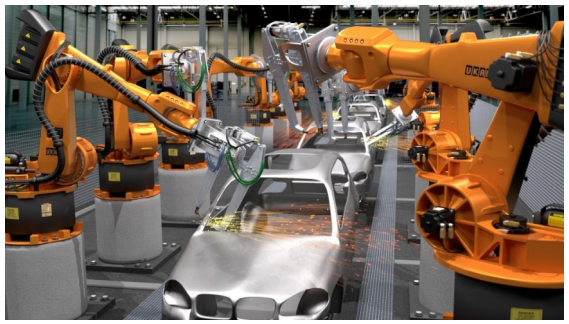
9

Internet of Things (IoT)

Internet connected embedded systems

Cyber-Physical Systems (CPS)

- Cyber system (Computer) + Physical system (Plant)
- Embedded systems, but integration of physical system is emphasized



11

Real-time systems

- The correctness of the system depends on not only on the logical result of the computation but also on the time at which the results are produced
- **A correct value at a wrong time is a fault.**

- CPS are often real-time systems
  - Because physical process depends on time

What makes embedded systems different?

Embedded vs. general computing systems: Number of applications

Limited

Many

Embedded vs. general computing systems: Programmability

Not end-user programmable

End-user programmable

Embedded vs. general computing systems: Run-time constraints

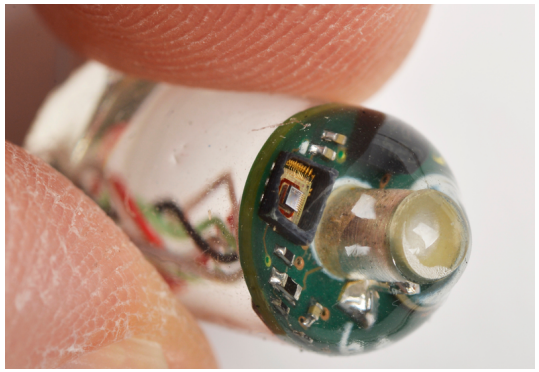
On-time

Faster is
always better!

Embedded vs. general computing systems: Resource constraints

Power, memory,
& compute limited:

Ex: smart watch, implantable healthcare device



A Tiny Pill Monitors Vital Signs From Deep Inside The Body, NPR

Greater resource!

Ex: servers

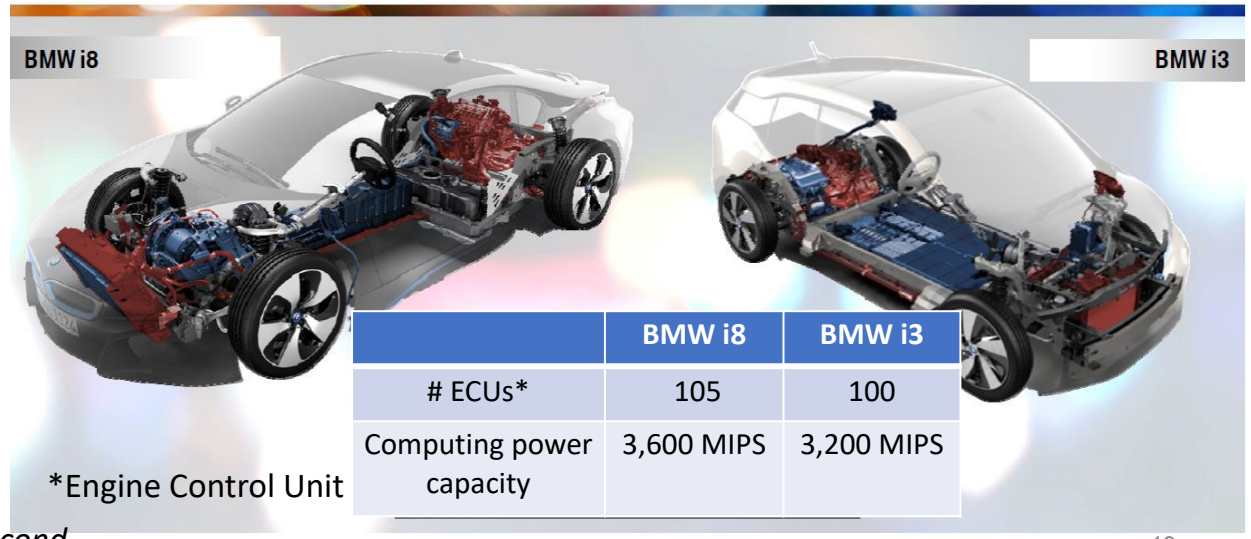
Trends in modern embedded systems

Trend: Cheaper, more powerful, and more connected computing

18

Trends in modern embedded systems

- The ECU computing cacpicity in BMW i8 is greater than i3.
- These computers were developed in 2014 or earlier.
- Raspbarry pi 2 released in 2015 had even greater speed.
  - Compute power of a Raspberry Pie 2 is 4,744 MIPS at 1.0GHz!



	BMW i8	BMW i3
# ECUs*	105	100
Computing power capacity	3,600 MIPS	3,200 MIPS

*Engine Control Unit

MIPS: Millions of instructions per second

Simon Fürst, BMW, EMCC2015 Munich, adopted from OSPERT2015 keynote

19

Trends in modern embedded systems

Human-like intelligence needs for autonomous cars

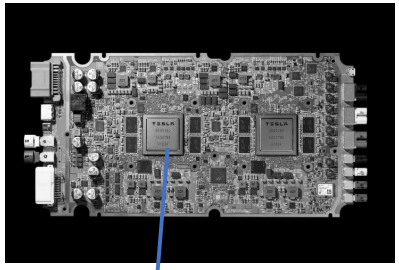
- Sophisticated sensors and algorithms
- High-performance embedded computers

Semi autonomous car

Trends in modern embedded systems

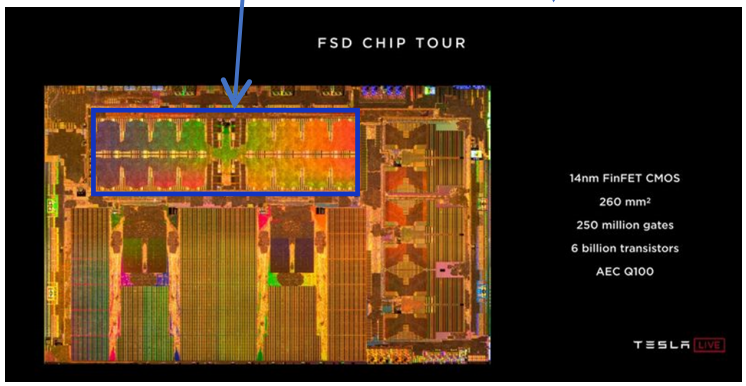
Super-computer on a car: Tesla FSD chip introduced in 2019

- Full Self-Driving Chip (FSD Chip, previously Autopilot Hardware 3.0)
- aimed at autonomous levels 4 and 5 (fully self driving).



GPU: 1Ghz, 600 GFLOPs

- What's inside:
  - Incorporates 3 quad-core Cortex-A72 clusters for a total of 12 CPUs operating at 2.2 GHz
  - GPU operating 1 GHz
  - 2 neural processing units operating at 2 GHz
  - various other hardware accelerators



Learn more: https://www.youtube.com/watch?time_continue=4988&v=Ucp0TTmvqOE

FSD: Full Self-Driving Chip

GFLOPs: Gigaflop: one billion floating-point operations/sec.

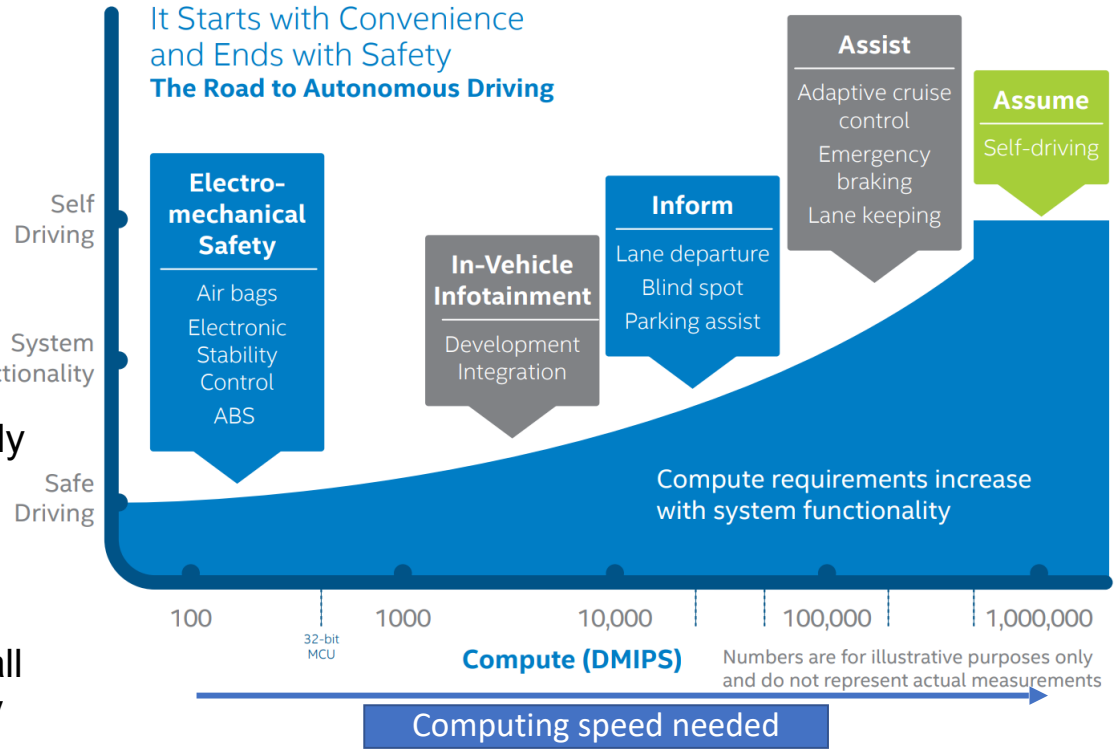
Requirements for CPS/IoT/Embedded Systems

1. Performance
2. Efficiency
3. Safety
4. Security and Privacy

Performance

Intel describes the trajectory of change in the automobile as a three-step process: “Inform, Assist, Assume.”

- **Inform**: Drivers wanted a way to seamlessly meld their connected lifestyle with their car.
- **Assist**: new capabilities that actively “assist” the driver.
- **Assume**: able to communicate, collaborate, and ultimately fulfill the human driving functions in almost all driving situations (e.g., parking, city driving, freeway driving).



23

Intel, “Technology and Computing Requirements for Self-Driving Cars”

Performance

Processing real-time data from many sensors requires powerful computers

- Approximately 1 GB of data will need to be processed each second in the car's real-time operating system.

Source: http://on-demand.gputechconf.com/gtc/2015/presentation/S5870-Daniel-Lipinski.pdf

Efficiency

Size, weight, power and cost constraints

CMU's "Boss" Self-driving car, circa 2007
10 dual-processor blade servers on the trunk

Audi A8

Audi's zFAS platform. 2016-2018
A single-board computer with multiple CPUs, GPU, FPGA

Safety

Therac 25

- Computer controlled radiation therapy machine
- Six people died/injured due to massive overdoses (1985-1987)
- Caused by synchronization mistakes

Arian 5

- 7 billion dollar rocket was destroyed after 40 secs (6/4/1996)
- *"caused by the complete loss of guidance and altitude information "* → Caused by 64bit floating to 16bit integer conversion

https://www.bugsnag.com/blog/bug-day-race-condition-therac-25

Security and Privacy

Confidentiality issues: Information could be stolen from the system.

Attack can be done in many ways:

1. **Side channel leakage:** observing the power consumption of the chip to guess the password.
2. **Hardware or software Trojan:** hidden functionality in software or hardware that leaks information (e.g., secret data) to outsider
3. **Physical attack:** Decapsulation and deprocessing of the chip to see what's inside (e.g., secret data)

Security and Privacy

Integrity issues: An attacker can initiate functional failure within the system

Attack can be done in many ways:

1. **Fault injection:** Causing a voltage glitch to corrupt the output of the chip
2. **Hardware or software Trojan:** hidden functionality in software or hardware that corrupts the output when the attacker activates the Trojan
3. **Physical attack and tampering:** mod chip attack on gaming console to bypass restrictions

Summary

Embedded systems, cyber-physical systems, real-time systems, IoT

- Limited function computers integrated into physical world
- Requirements: performance, efficiency, safety, security, and privacy

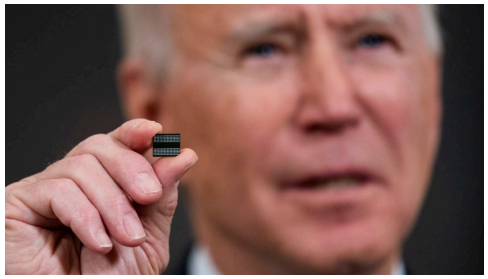
In this course we learn concepts and skills for developing embedded systems

Future Job Prospect in Semiconductor Industry

New Chip-factory in Ohio

- “Intel will invest more than $20 billion … for advanced chipmaking in the Midwest.”
- Expected to create 3,000 Intel jobs
- “Could grow to as much as $100 billion over the next decade, making it one of the largest semiconductor manufacturing sites in the world”

- Source: Link



The CHIPS and Science Act

- US is dependent on a globalized supply chain for chip design
- President Joe Biden recently signed a law to aid the U.S. chip industry
- $52.7 billion for research and development and domestic manufacturing of semiconductors

30

Instructor and GTAs

Tamzidul Hoque
Instructor, hoque@ku.edu

Arin Dutta, GTA
arindutta60@ku.edu

Soma Pal, GTA
soma_pg2021@ku.edu

Sm Ishraq-Ul Islam, GTA
ishraq@ku.edu

31

Course Structure

- Lecture (in-person)
  - Focus on key concepts
- Homework, In class quiz, exams
- Lab (in-person)
  - Hands-on embedded systems programming experiences

Tentative Grading

- Homework: 10%
- In class quiz: 5%
- Midterm: 20%
- Final: 25%
- Lab: 40%
  - *(Weekly labs 25% + final project: 15%)*

- **Grading Scale**

A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F
93<=	[89-93)	[86-89)	[83-86)	[79-83)	[76-79)	[73-76)	[69-73)	[66-69)	[63-66)	[59-63)	<59

Lecture Recording

- I will try to record it, but there is no guarantee for all lectures to be available/recorded correctly.
  - Don’t count on recording and miss the classes
  - Use it for reviewing before the exam
- Absence in the class may lead to missing in class quizzes

Please do not cheat,

and lets enjoy the course together

Syllabus will be uploaded today with detailed course policies