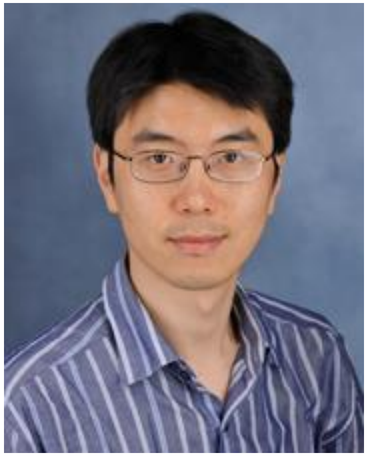
EECS 388: Embedded Systems

1. Introduction

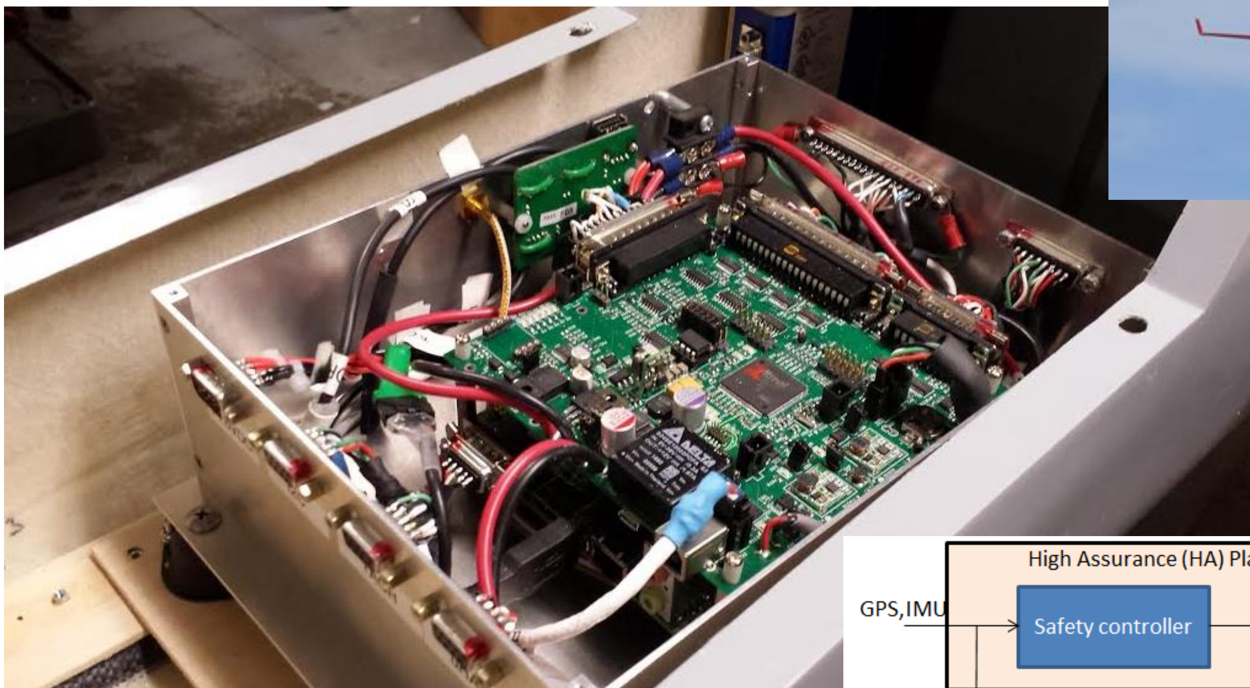
Heechul Yun

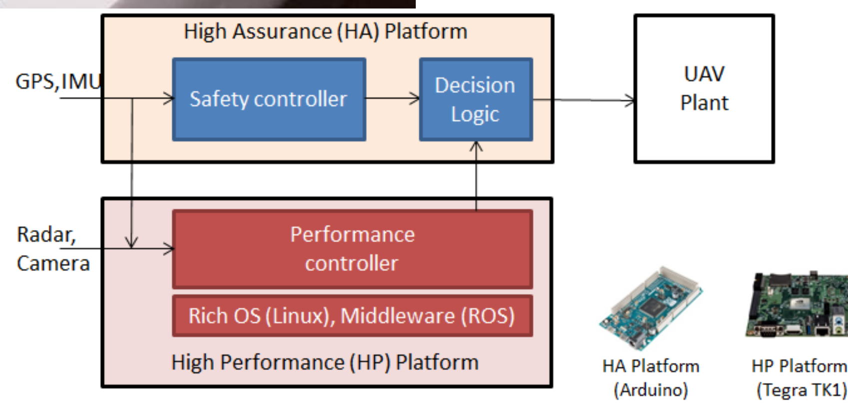
Heechul Yun



- Associate Prof., Dept. of EECS
- Offices: 3040 Eaton, 236 Nichols
- Email: heechul.yun@ku.edu
- KU EECS faculty since 2013
- Education: UIUC (PhD), KAIST (MS, BS)
- Embedded software engineer at Samsung
- Research Areas
  - Embedded/real-time systems, OS, architecture
- More Information
  - http://ittc.ku.edu/~heechul

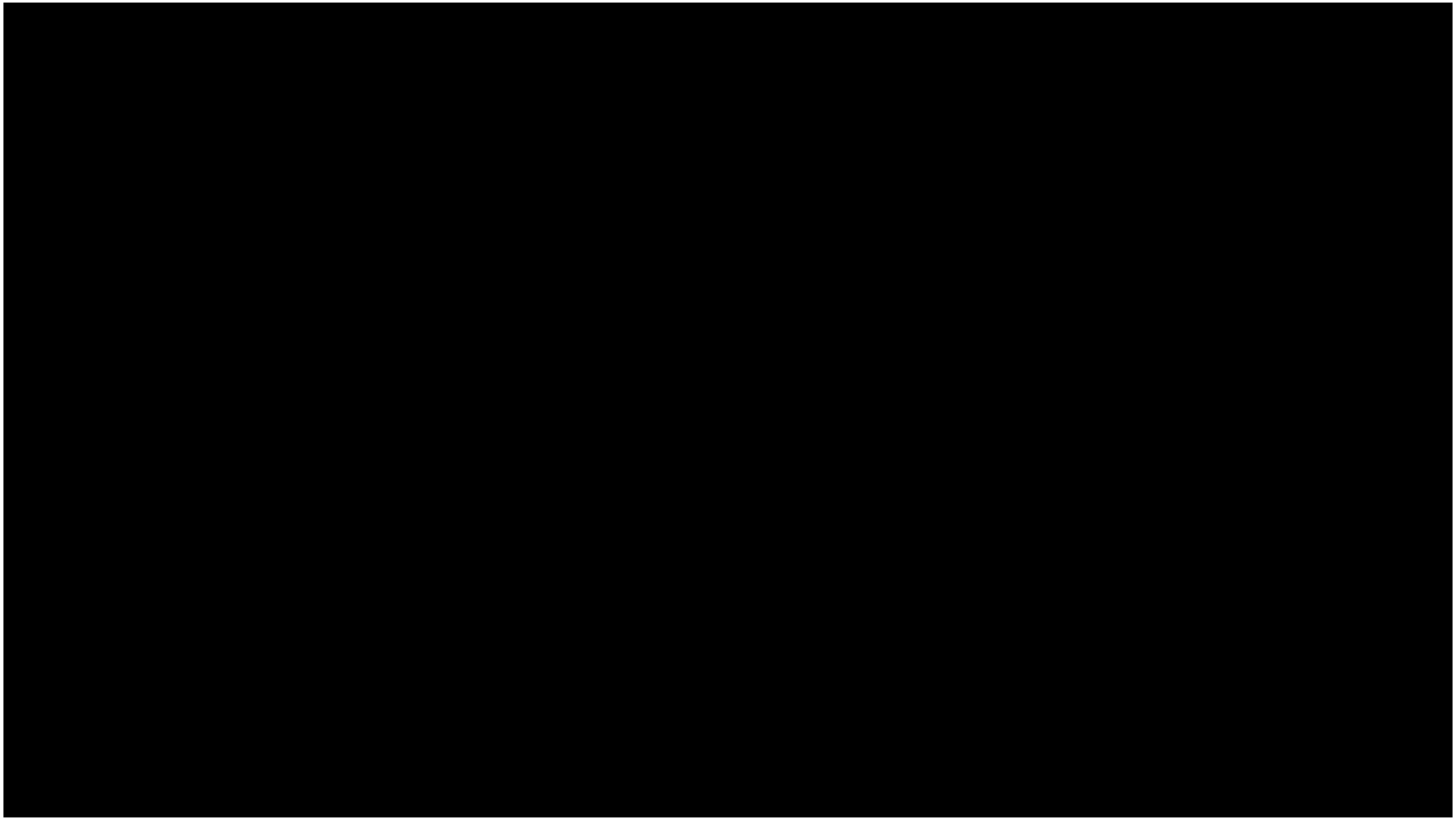
KU Fixed-wing UAV





(*) Prasanth Vivekanandan, Gonzalo Garcia, Heechul Yun, Shawn Keshmiri. A Simplex Architecture for Intelligent and Safe Unmanned Aerial Vehicles. In *RTCSA*, IEEE, 2016.

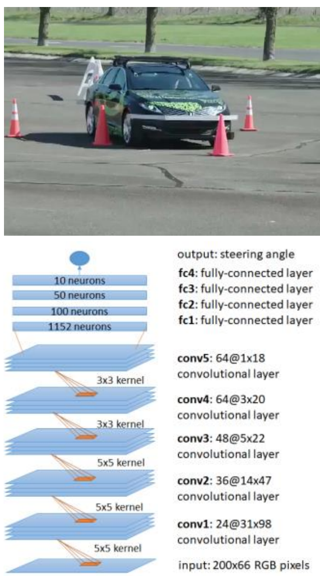
KU Fixed-wing UAV

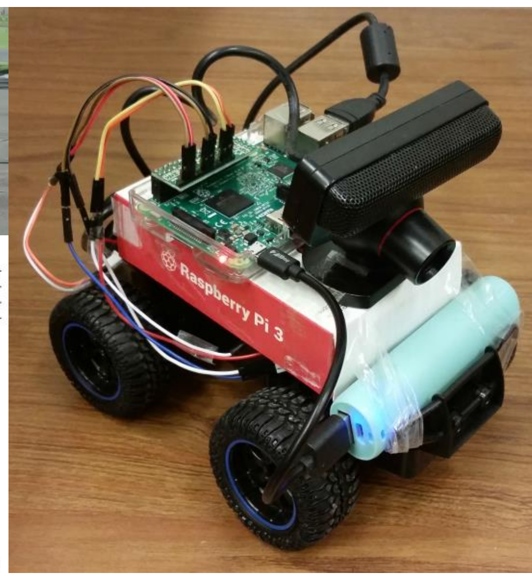


(*) Prasanth Vivekanandan, Gonzalo Garcia, Heechul Yun, Shawn Keshmiri. A Simplex Architecture for Intelligent and Safe Unmanned Aerial Vehicles. In *RTCSA*, IEEE, 2016.

DeepPicar

- End-to-end deep learning: *pixels* to *steering*
- Using **identical** DNN with NVIDIA's DAVE-2

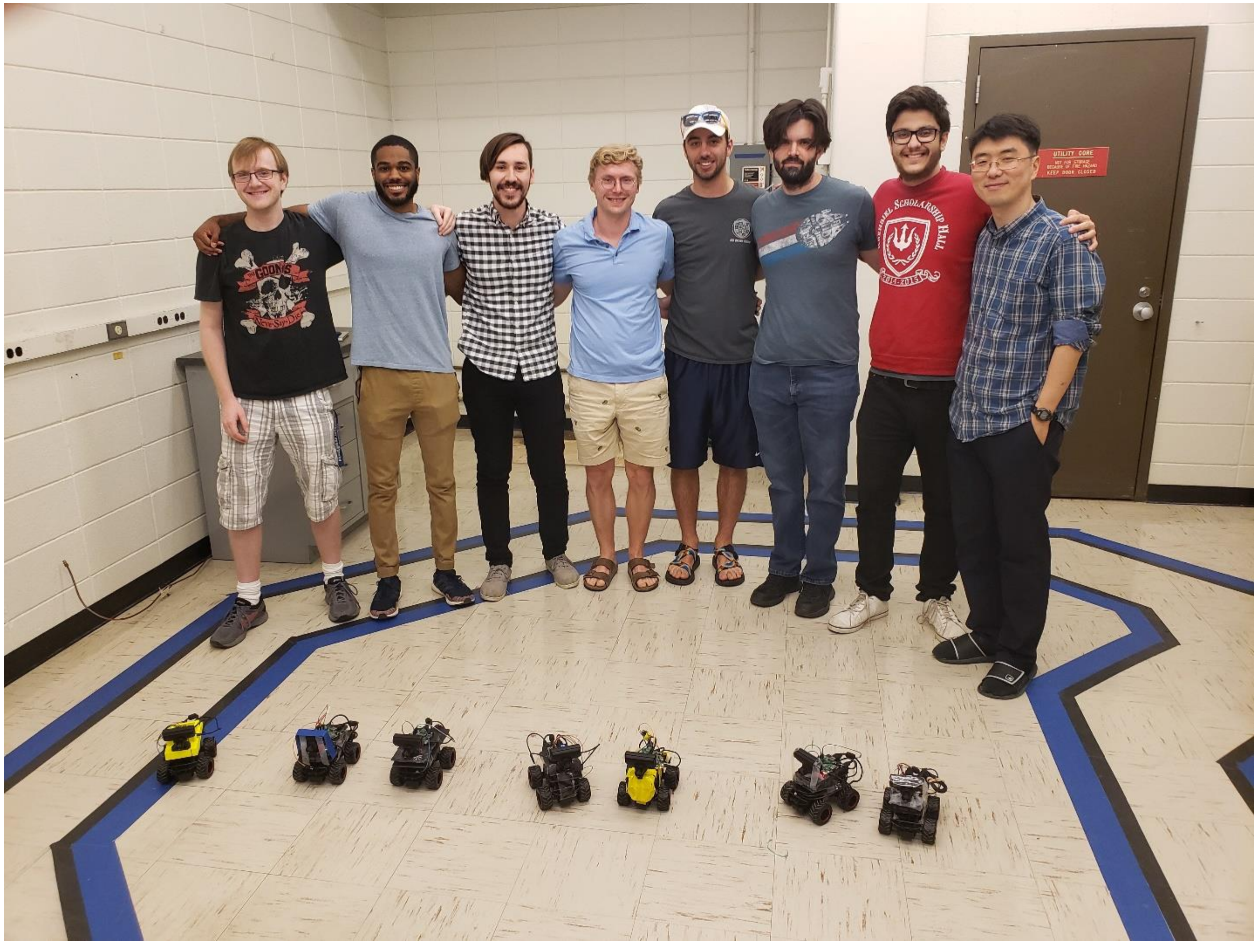




More self-driving videos: https://photos.app.goo.gl/q40QFieD5iI9yXU42

Michael G. Bechtel, Elise McEllhiney, Minje Kim, Heechul Yun. "DeepPicar: A Low-cost Deep Neural Network-based Autonomous Car." In *RTCSA*, 2018.

EECS 753



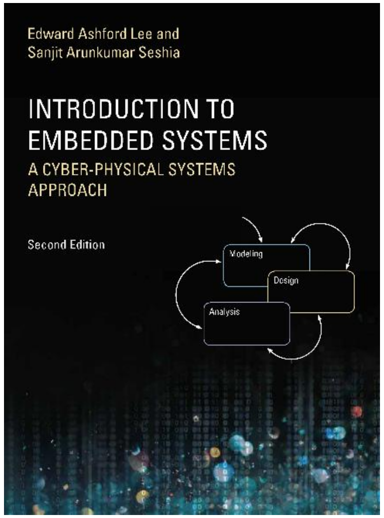
DeepPicar Competition

EECS 753 Embedded Real-Time Systems Final Project

May 6, 2019

About This Class

- Textbook
  - Introduction to Embedded Systems: A cyber-physical systems approach
  - http://LeeSeshia.org/
- Objectives
  - Learn key concepts and practical skills to develop cyber-physical/embedded systems
- Course website
  - http://ittc.ku.edu/~heechul/courses/eecs388



Course Structure

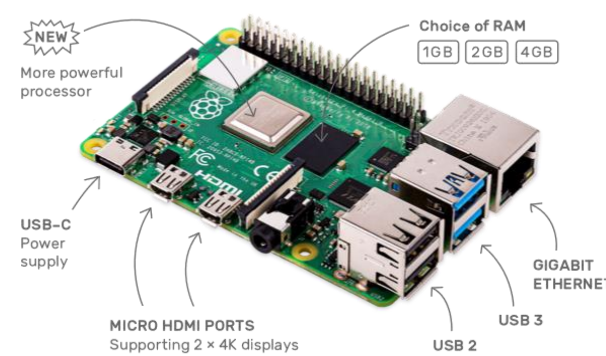
- Lecture
  - Embedded systems design and implementation
  - Focus on key concepts
- Quiz
  - Weekly online quizzes to check your understanding
- Lab
  - Hands-on embedded systems programming experiences.
- Project
  - Self-driving car prototype

Lab

- HiFive1 (rev b) board
  - RISC-V micro-controller
  - Limited resources/performance
  - "Bare-metal" programming in C
    - Directly access hardware w/o OS

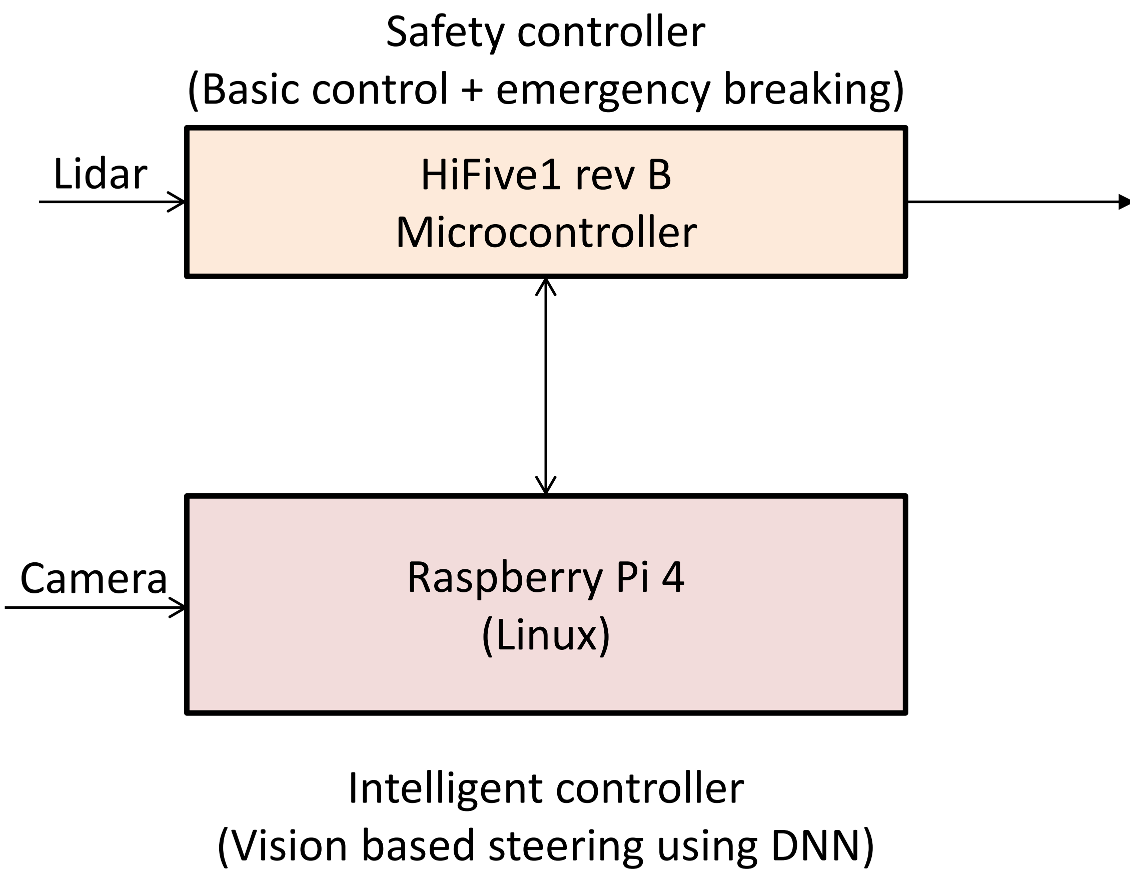
HiFive 1 rev B

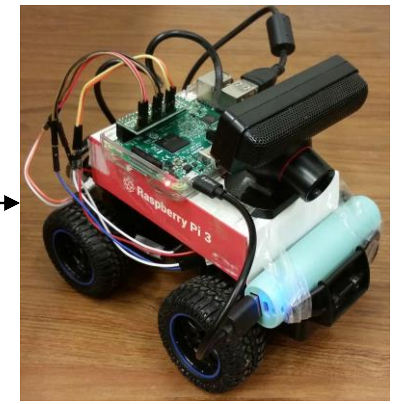
- Raspberry Pi 4
  - Powerful quad-core ARM CPU
  - Run fully featured OS (Linux)
  - Standard PC-like programming environment



Raspberry Pi 4

Project





Self-Driving Car

Grading

- Attendance: 5%
- Exam: 50% (Mid:20%, Final:30%)
- Quiz: 5%
- Lab: 30%
- Project: 10%

Grading

- 90+ : A
- 80-89: B
- 70-79: C
- 50-69: D
- 0-49: F

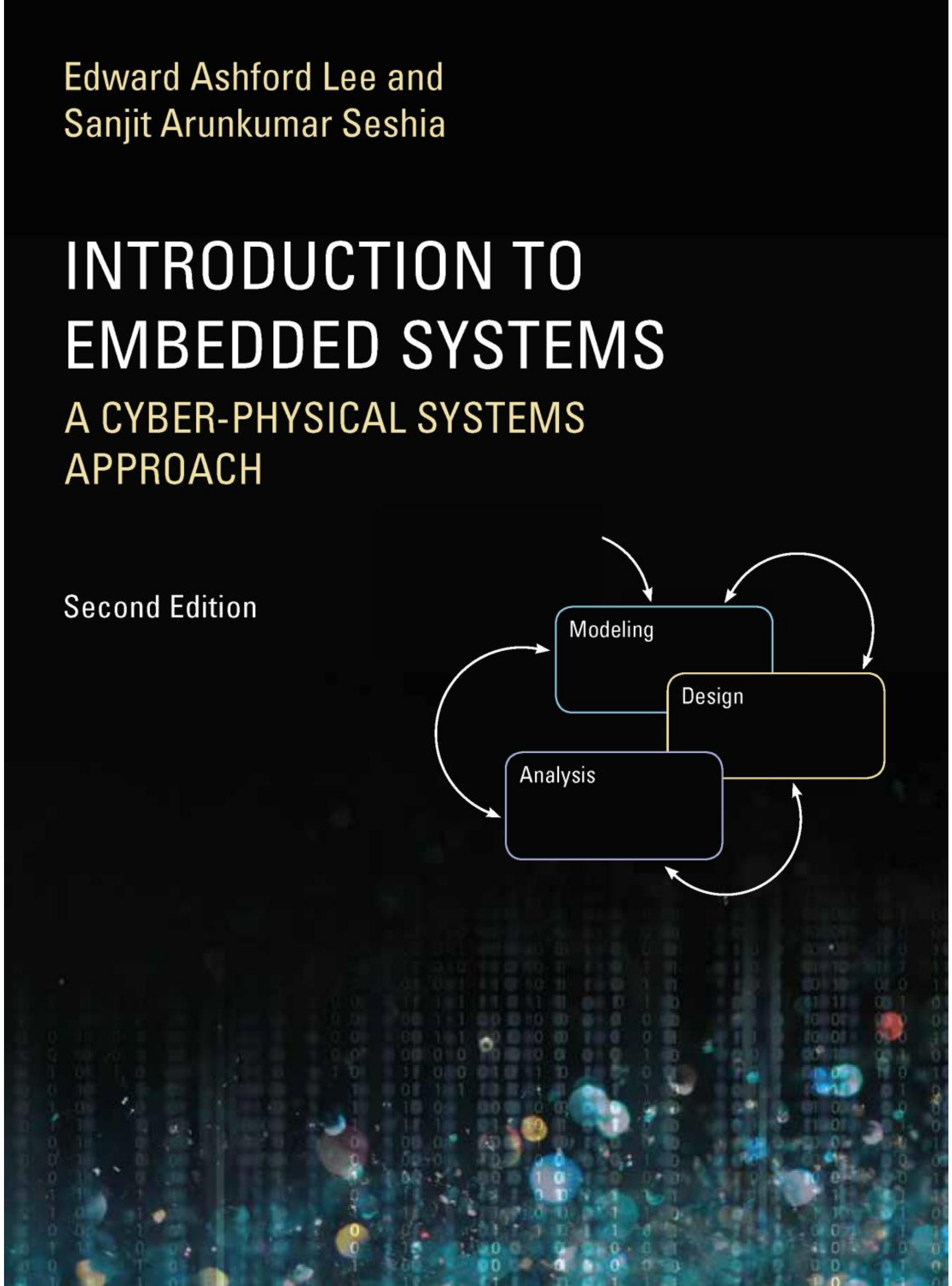
KU THE UNIVERSITY OF KANSAS

Policy

- Late submissions
  - 20% off each additional 24 hours delay (~24h = 80%, ~48h = 60%, ~72h=40%, ~96h=20%, >96h = 0%)
- Cheating
  - You can discuss about code and help find bugs of your peers. However, copying another's code (e.g., from github) or writing code for someone else is cheating and, if identified, **the involved students will be notified to the department chair**
- Public code repository
  - Do not post your lab solutions on publicly accessible web sites (e.g., GitHub).
  - Do not download other students' solutions.

Schedule

- http://www.ittc.ku.edu/~heechul/courses/eecs388/schedule.html


Edward Ashford Lee and
Sanjit Arunkumar Seshia
INTRODUCTION TO
EMBEDDED SYSTEMS
A CYBER-PHYSICAL SYSTEMS
APPROACH
Second Edition
Modeling
Design
Analysis

Embedded Systems

- Computing systems designed for **specific purpose**.
- Embedded systems are everywhere

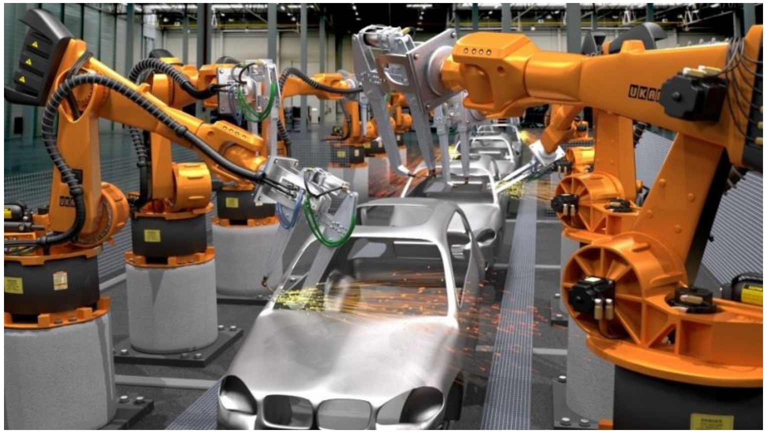


Internet of Things (IoT)

- IoT ~= **Internet** connected embedded systems

Cyber-Physical Systems (CPS)

- Cyber system (Computer) + Physical system (Plant)
- Still embedded systems, but **integration of physical systems** is emphasized.



THE UNIVERSITY OF KANSAS

Real-Time Systems

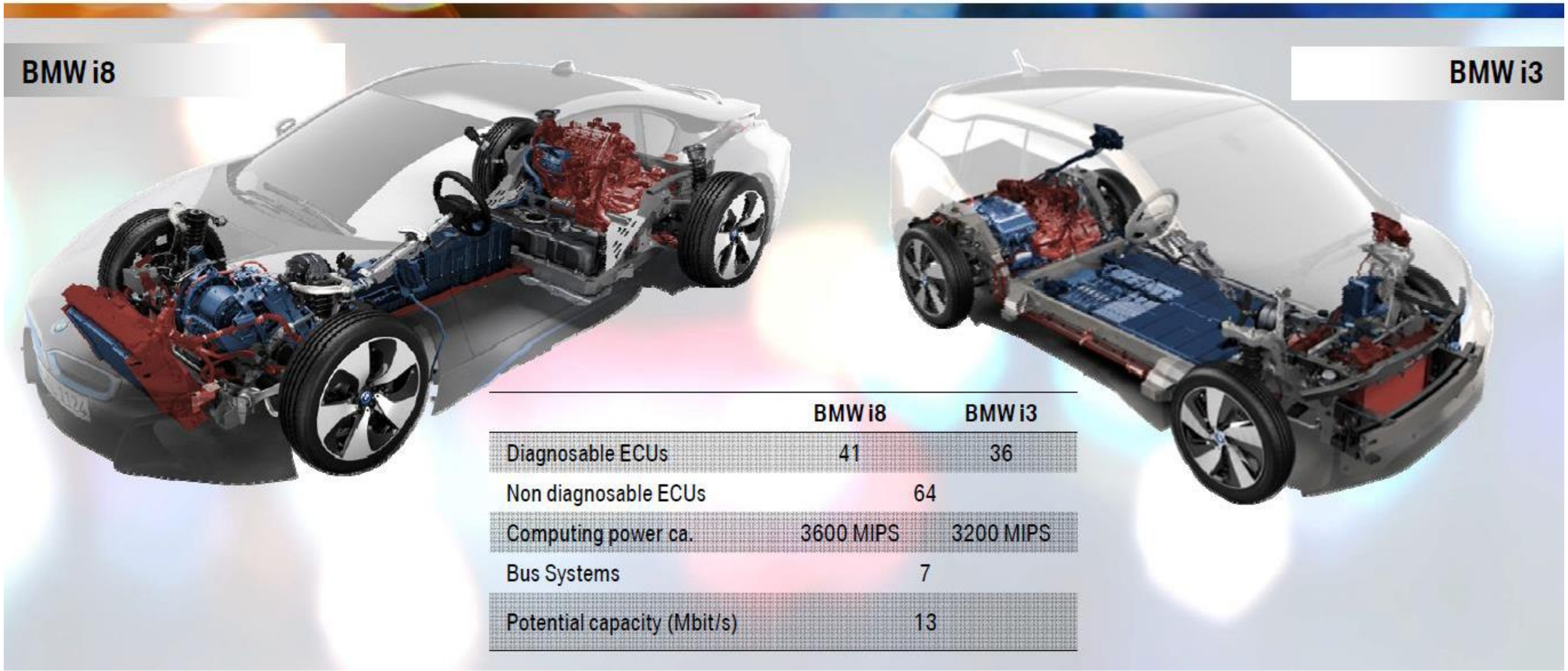
- The correctness of the system depends on not only on the logical result of the computation but also on the time at which the results are produced
- **A correct value at a wrong time is a fault.**
- CPS are often real-time systems
  - Because physical process depends on time

Trends

- More **powerful** and **cheaper** computing
- More **connected**

Today's Car

- A cyber-physical system: computers control the car
- Quiz. How many embedded processors are in a car?
  - A: ~100s



	BMW i8	BMW i3
Diagnosable ECUs	41	36
Non diagnosable ECUs	64	
Computing power ca.	3600 MIPS	3200 MIPS
Bus Systems	7	
Potential capacity (Mbit/s)	13	

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

Autonomous Car

- Human-like intelligence needs
  - Sophisticated sensors and algorithms
  - high-performance embedded computers

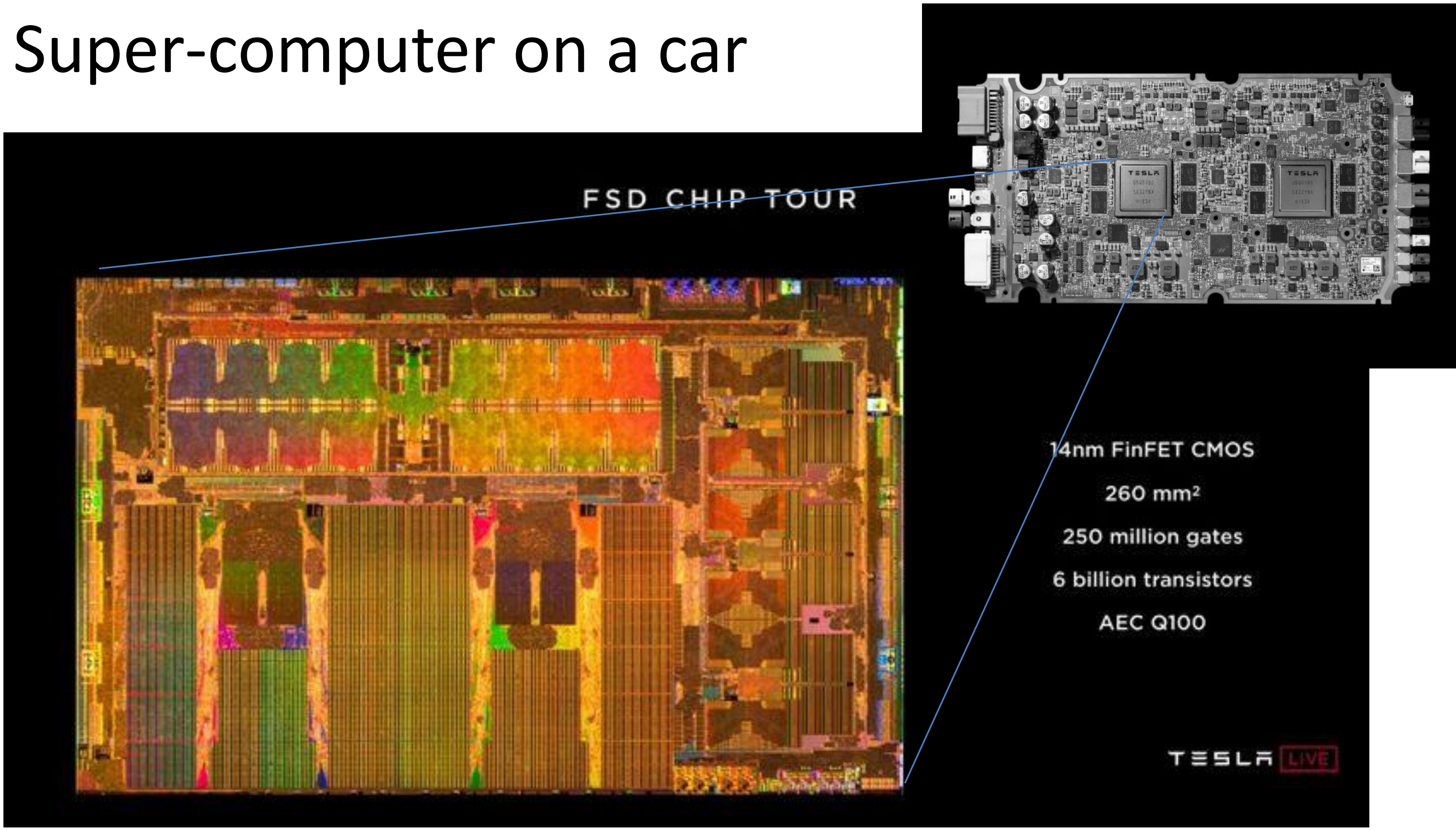
Semi autonomous car

https://www.latimes.com/business/autos/la-fi-waymo-self-driving-california-20181030-story.html

Fully autonomous car

Tesla FSD Chip

- Super-computer on a car



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

Today’s Airplane

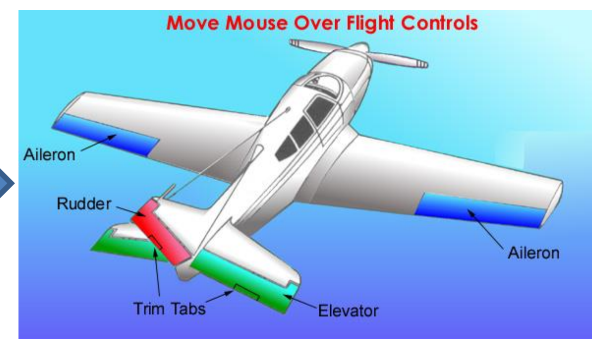
- Avionics: electronic systems on an airplane
  - Aviation + electronics
  - Multiple subsystems: communications, navigation, display, flight control, management, etc.
- Modern avionics
  - Increasingly computerized

Fly-by-wire

- Modern aircrafts rely on computers to fly
- Pilots do not directly move flight control surfaces (ailerons, elevator, rudder)
- Instead, Electronic Flight Control System (FCS) does.

Yoke

FCS



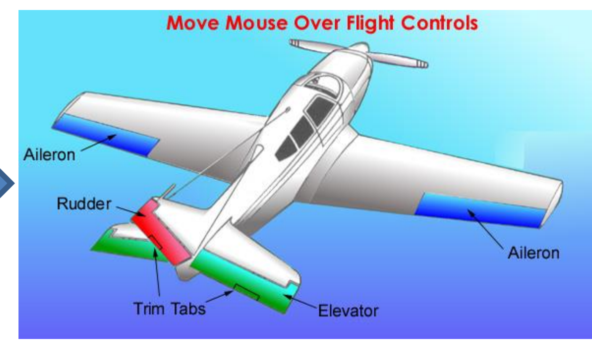
Control surfaces

Autopilot

- Specify desired track: heading, course, waypoints, altitude, airspeed, etc.

Yoke

→ FCS →



Control surfaces

Modern Cyber-Physical Systems

- Cyber Physical Systems (CPS)
  - Cyber (Computer) + Physical (Plant)
- Real-time
  - Control physical process in real-time
- Safety-critical
  - Can harm people/things
- Intelligent
  - Can function autonomously

CPS Requirements

- Performance and efficiency
  - Meet deadlines in processing large amounts of real-time data from various sensors (e.g., autonomous cars)
  - Many constraints: size, weight, and power (SWaP); cost
- Safety
  - Interact with the environment, human, in real-time
  - Can hurt humans, destroy things, blow up (e.g., Nuclear plants)
  - Need *both* logical and temporal (time) correctness
- Security
  - Communicate over the internet (cloud servers etc.)
  - Remote software update (fix bugs, …)
  - Run untrusted 3rd party software (e.g., Apple CarPlay)

Efficiency

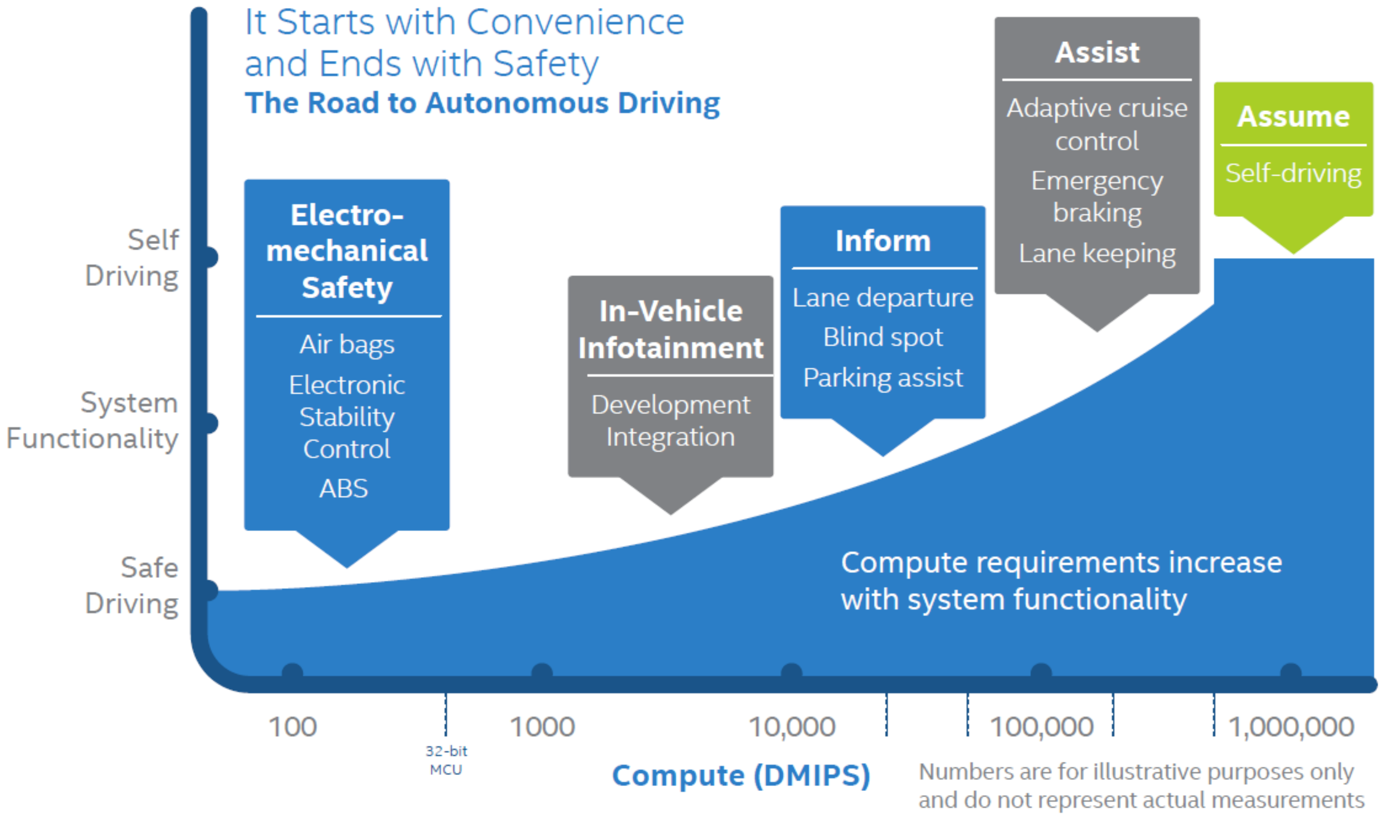
- Many cyber-physical systems (CPS) need:
  - *More* performance for higher autonomy
  - *Less* cost, size, weight, and power

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

Compute Performance Demand



[Intel, "Technology and Computing Requirements for Self-Driving Cars"]

Real-Time Data

- Big data needs powerful computers

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

Size, Weight, and Power (SWaP) Constraints

- Maximum performance with minimal resources
  - Cannot afford too many or too power hungry ECUs

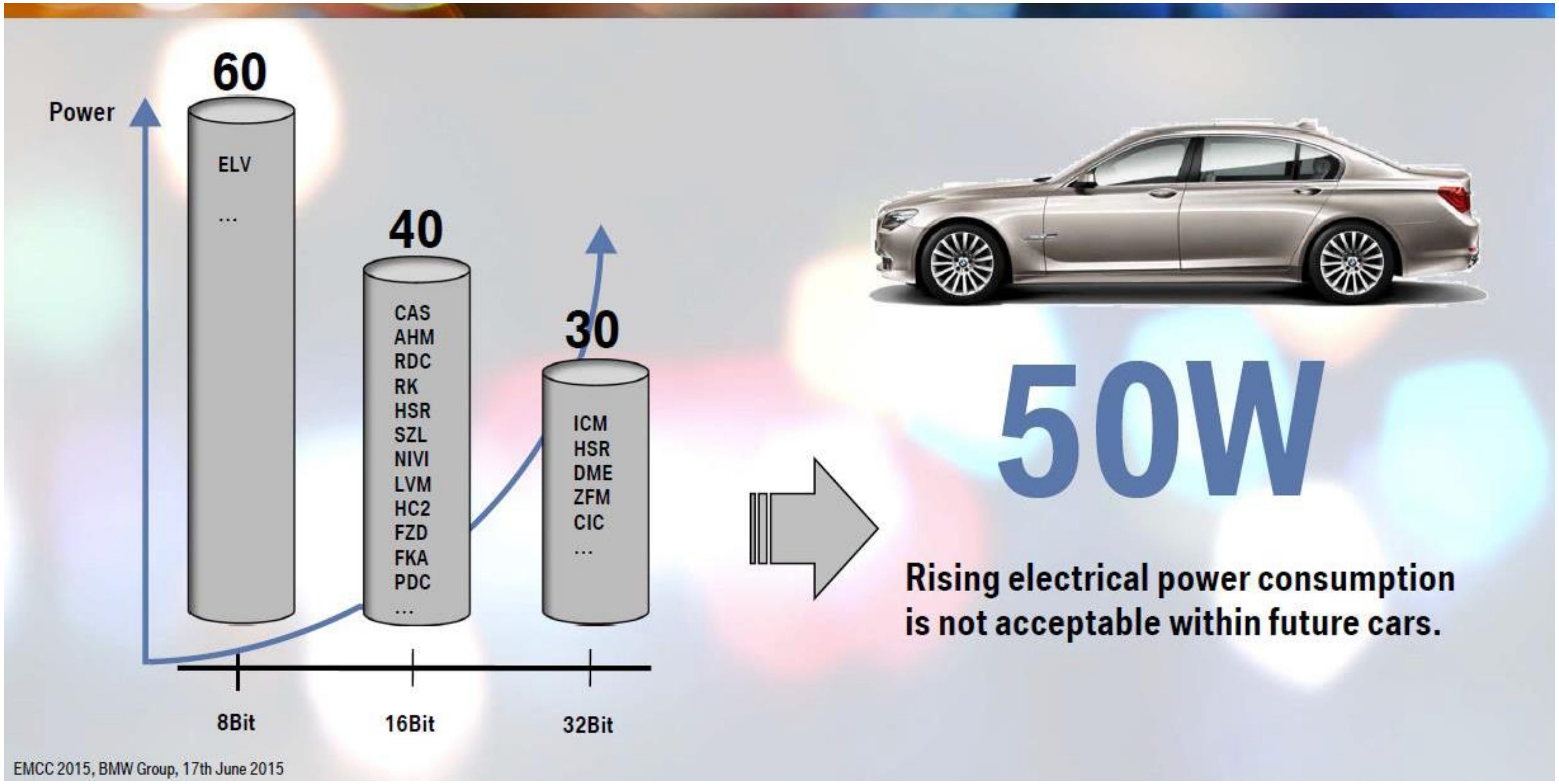
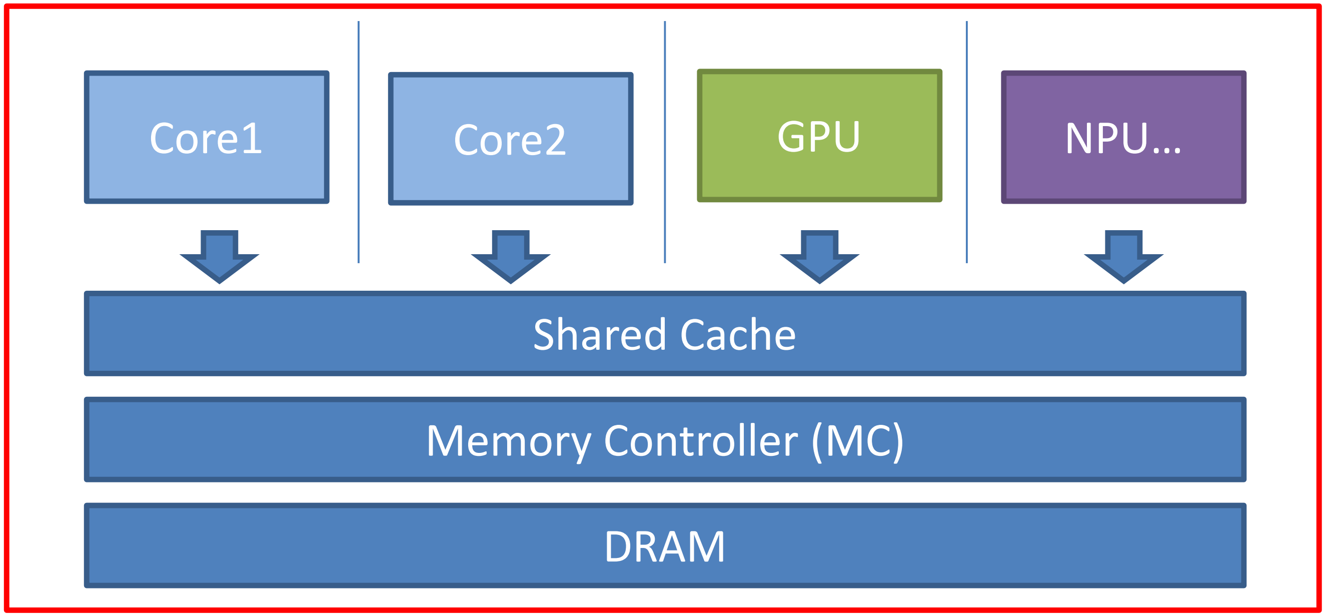
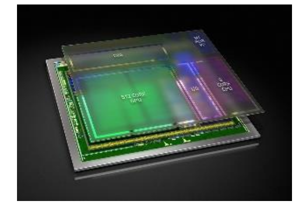


Figure source: OSPERT 2015 Keynote by Leibinger

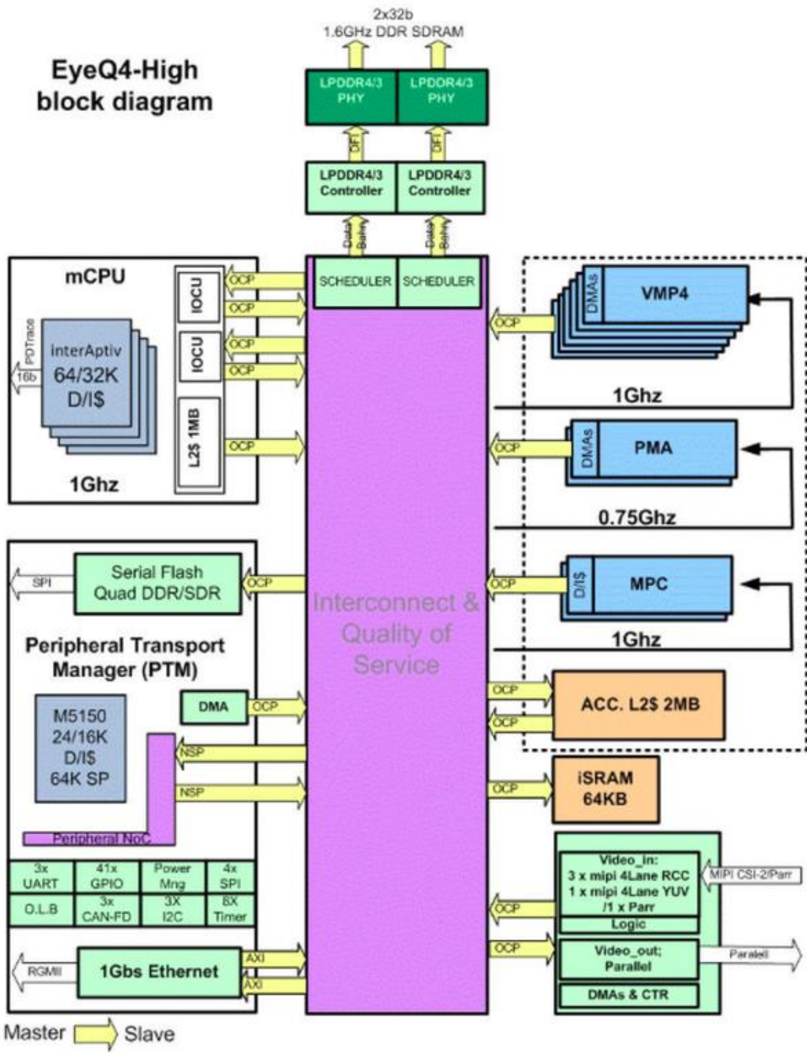
Modern System-on-a-Chip (SoC)





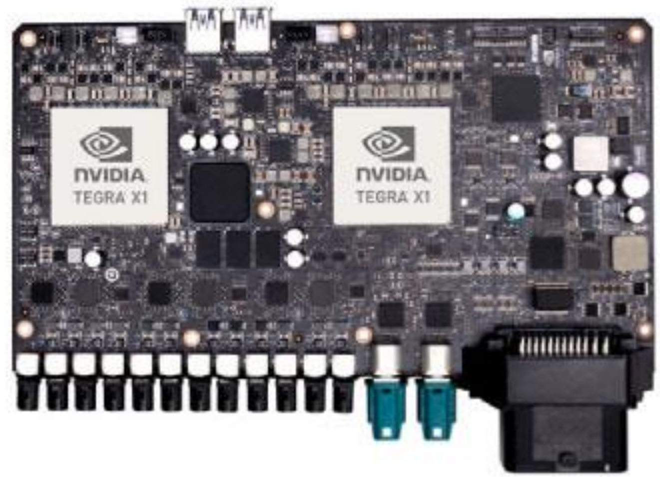
- Integrate multiple cores, GPU, accelerators
- Good performance, size, weight, power
- Introduce new challenges in real-time, security

Mobileye EveQ4

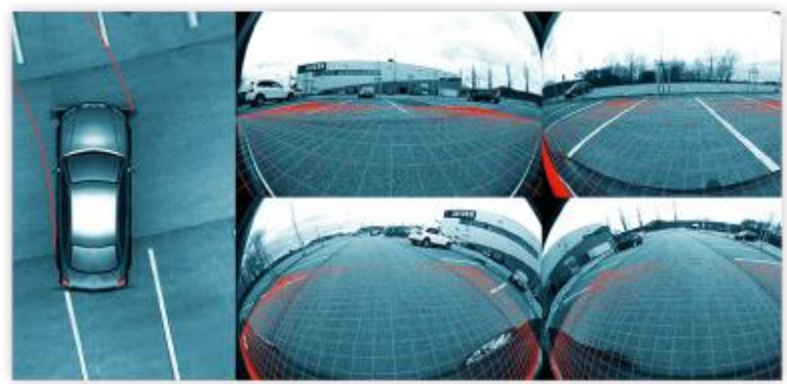


- Real-time vision processor w/ DNN
- 2.5 teraflops @ 3W
- 8 cameras @ 36 fps
- Tesla uses EveQ3
- 14 cores
  - – 4 MIPS cores
  - – 10 vector cores

Nvidia's Drive PX2 Platform



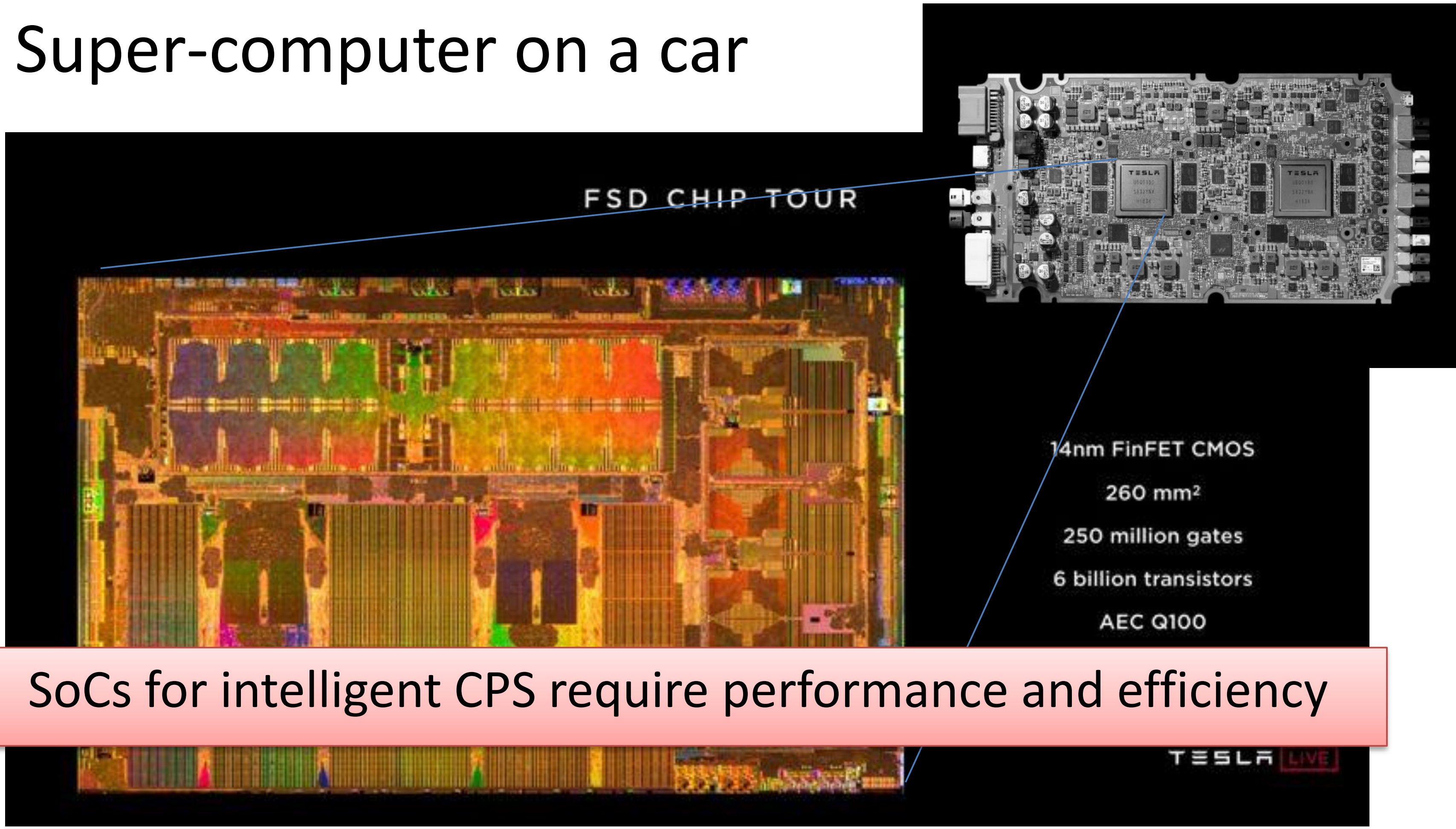
- 12 CPU + 2 GPU
  - 8 Tegraflops @250W
- Real-time processing of
  - Up to 12 cameras, radar, ..
  - Deep Neural Network (DNN) for detection, classification



http://www.nvidia.com/object/drive-px.html

Tesla FSD Chip

- Super-computer on a car



SoCs for intelligent CPS require performance and efficiency

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

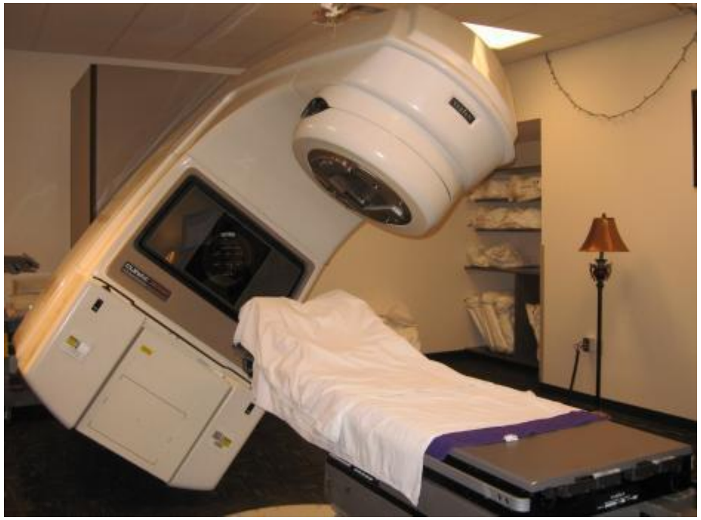
Safety

- Many CPS are safety-critical systems
  - Can harm people or things

CPS Challenge Problem: Prevent This

From Dr. Edward A. Lee, UCB

Safety Failures



Therac 25

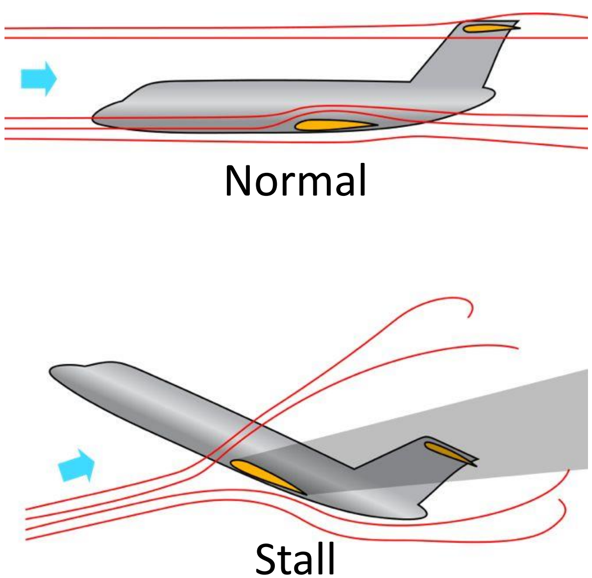
- Computer controlled medical X-ray treatments
- Six people died/injured due to massive overdoses (1985-1987)
- Caused by synchronization mistakes

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

Air France 447 (2009)

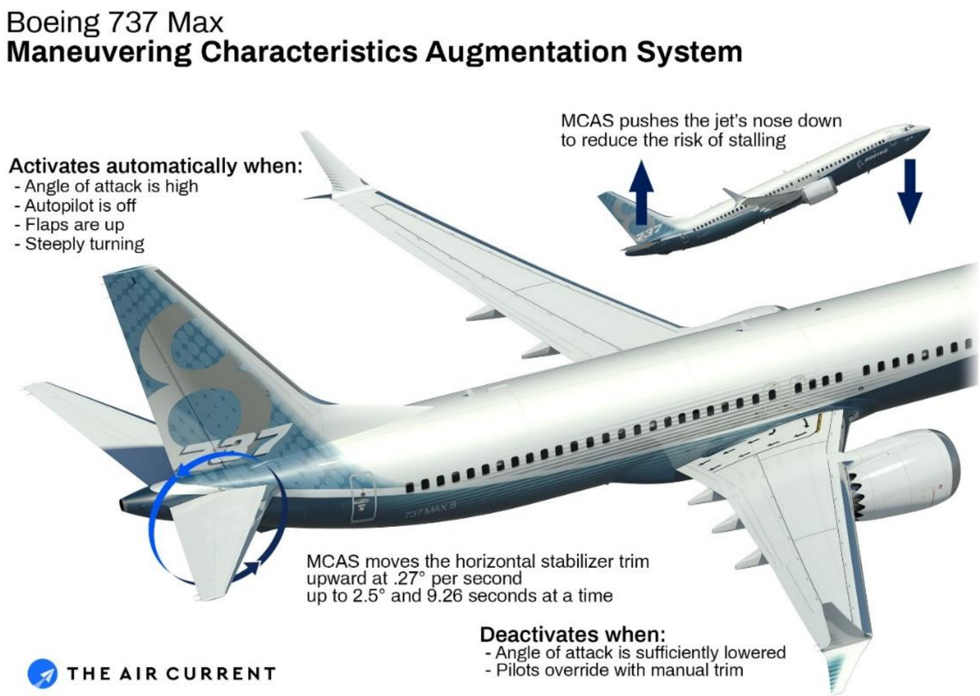
- Airbus A330 crashed into the Atlantic Ocean in 2009
- Caused in part by computer's misguidance
  - Pitot tube (speed sensor) failure → Flight Director (FD) malfunction (shows "head up") → pilots follow the faulty FD → enter **stall**



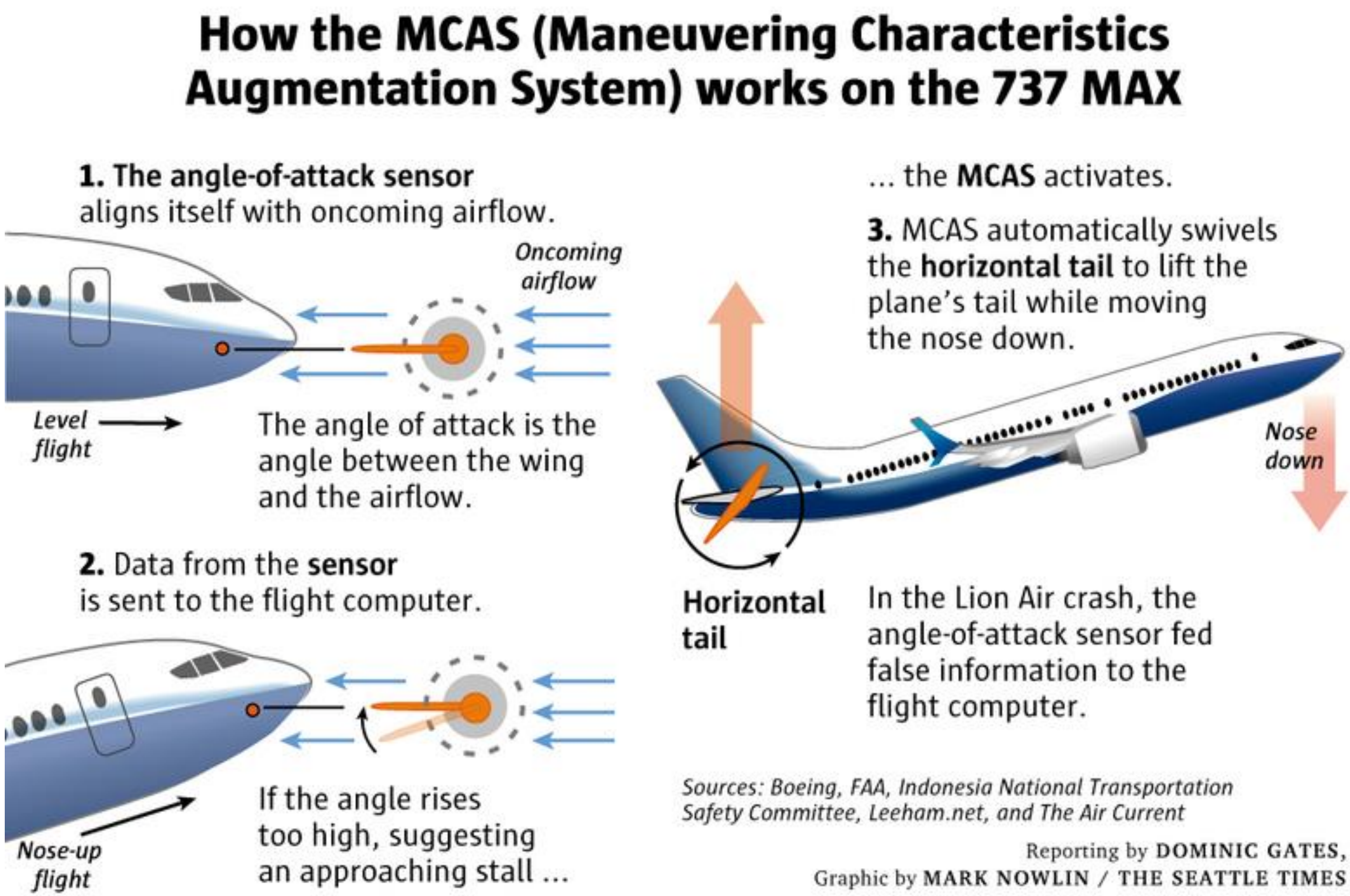
http://www.slate.com/blogs/the_eye/2015/06/25/air_france_flight_447_and_the_safety_paradox_of_airline_automation_on_99.html
http://www.spiegel.de/international/world/experts-say-focus-on-manual-flying-skills-needed-after-air-france-crash-a-843421.html

Lion Air Flight 610 (2018)

- Boeing 737 crashed into the Java See in 2018
- Caused by stall prevention system (MCAS)
  - sensor error (plane is “stall”) → nose down (to the ocean)



Ethiopian Air 302 (2019)



https://www.seattletimes.com/business/boeing-aerospace/failed-certification-faa-missed-safety-issues-in-the-737-max-system-implicated-in-the-lion-air-crash

Design Issues of 737 MAX's MCAS

- Operated on a single AoA sensor
  - A single source of failure
  - Despite of having two redundant sensors
- Repeated activation
  - No limit on how much the system can push the plane downward
  - MCAS > pilot's manual control
- Planned solutions
  - Use both sensors
  - Limited activation to limit the potential harm
  - MCAS < pilot's manual control

https://www.boeing.com/commercial/737max/737-max-software-updates.page

KU THE UNIVERSITY OF KANSAS

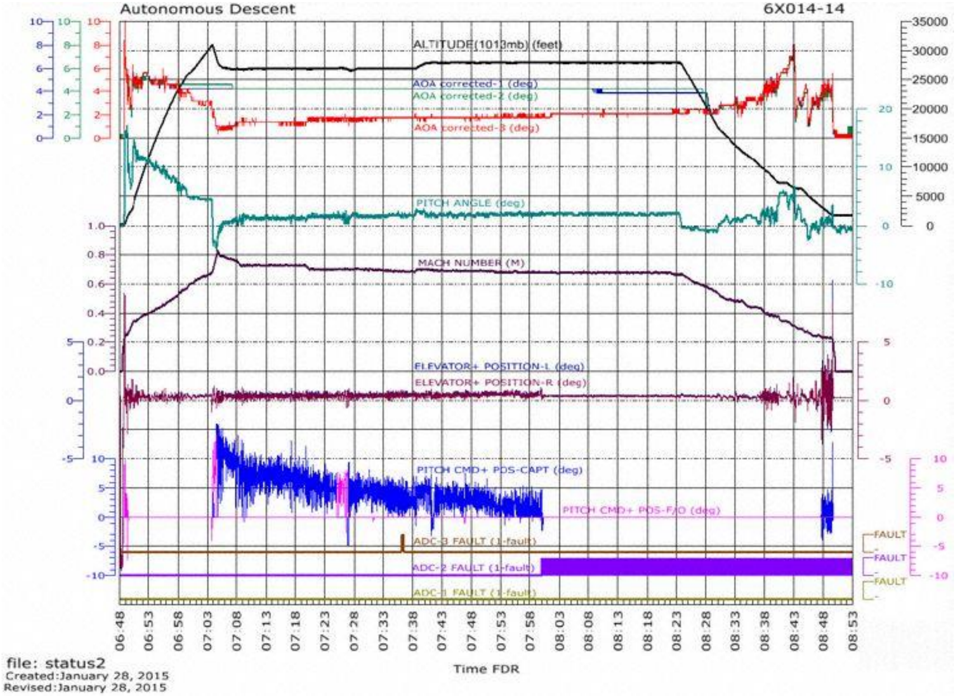
Lufthansa A321 (2014)

- Similar prior incidents that didn't kill people.
- Faulty AoA sensor readings (ice) trigger an automated stall prevention system, resulting 4,000 ft loss of altitude
- "When Alpha Prot is activated due to blocked AOA probes, the flight control laws order a continuous nose down pitch rate that, in a worst case scenario, cannot be stopped with backward sidestick inputs, even in the full backward position."

https://avherald.com/h?article=47d74074

Lufthansa A321 (2014)

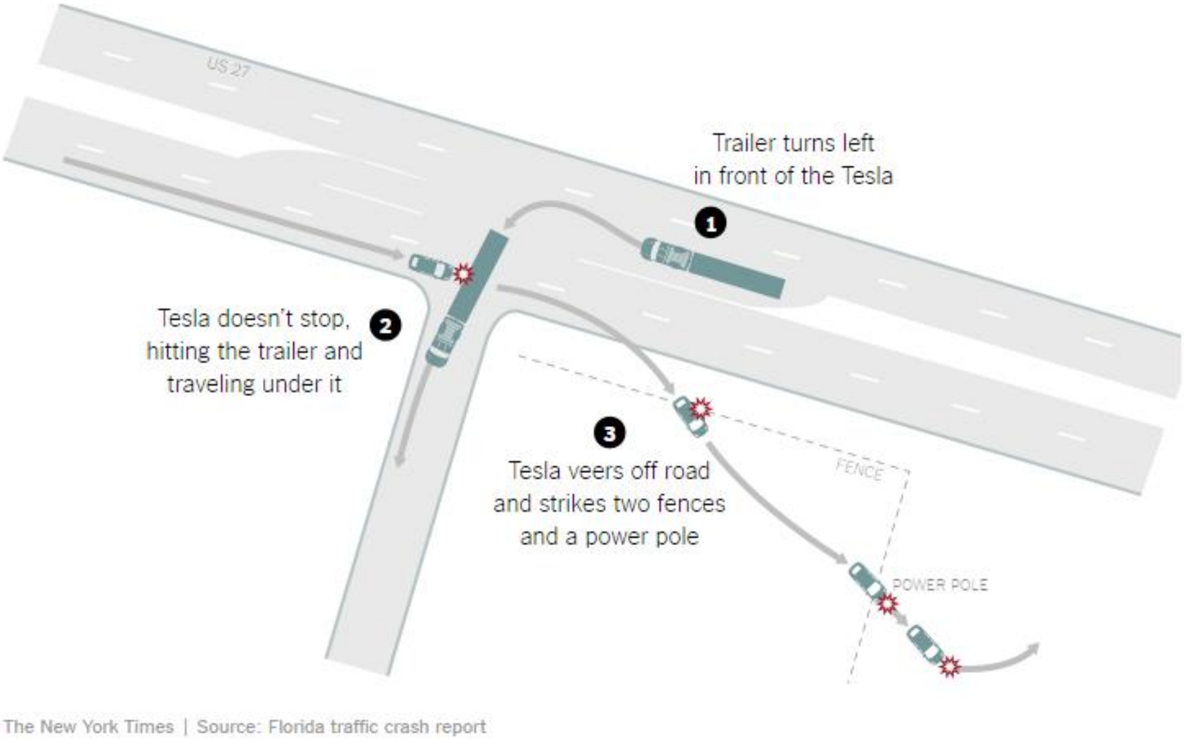
- Three redundant AoA sensors, but two freeze up simultaneously.
- The correct sensor's outputs were discarded.



KU THE UNIVERSITY OF KANSAS

Tesla Autopilot (2016)

- Tesla autopilot failed to recognize a trailer resulting in a death of the driver



http://www.nytimes.com/interactive/2016/07/01/business/inside-tesla-accident.html

NHTSA Report

- Both the radar and camera sub-systems are designed for **front-to-rear collision** prediction mitigation or avoidance.
- The system requires **agreement from both sensor systems** to initiate automatic braking.
- The camera system uses Mobileye's EyeQ3 processing chip which uses a large dataset of the rear images of vehicles to make its target classification decisions.
- Complex or unusual vehicle shapes may **delay or prevent** the system from **classifying certain vehicles as targets/threats**

https://static.nhtsa.gov/odi/inv/2016/INCLA-PE16007-7876.PDF

NHTSA Report

- Object classification algorithms in the Tesla and peer vehicles with AEB technologies are **designed to avoid false positive brake activations**.
- The Florida crash involved a target image (side of a tractor trailer) that would **not be a "true" target in the EyeQ3 vision system** dataset and
- The tractor trailer was **not moving in the same longitudinal direction** as the Tesla, which is the vehicle kinematic scenario the radar system is designed to detect

https://static.nhtsa.gov/odi/inv/2016/INCLA-PE16007-7876.PDF

KU THE UNIVERSITY OF KANSAS

Tesla Autopilot (2019)

- Similar condition

KU THE UNIVERSITY OF KANSAS

https://www.ntsb.gov/investigations/AccidentReports/Pages/HWY19FH008-preliminary-report.aspx

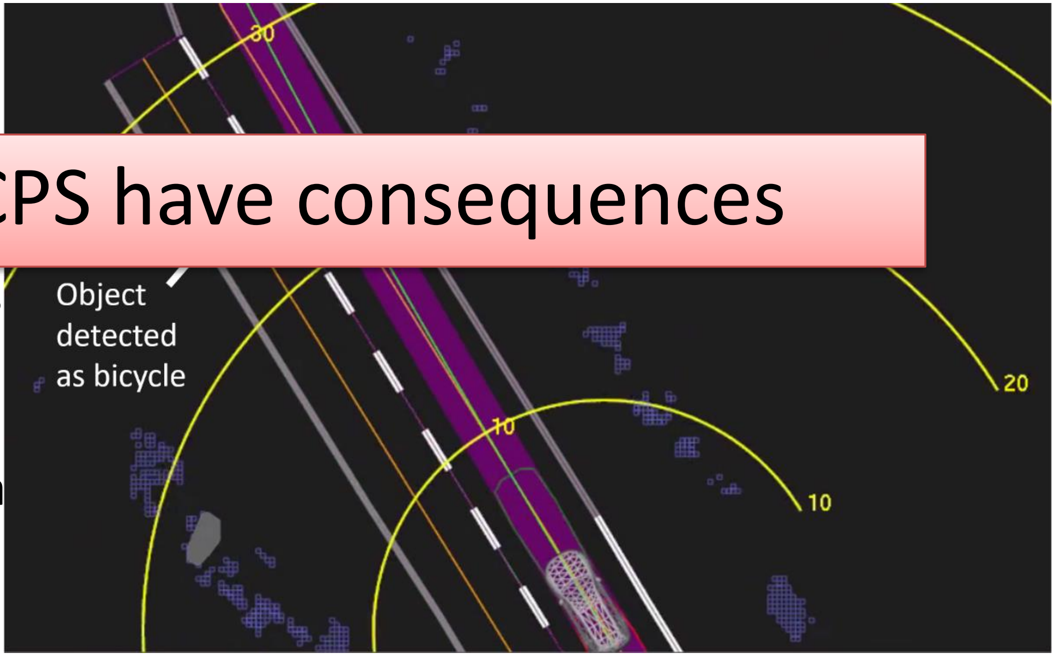
Uber Self-Driving Car (2018)

- Kill a pedestrian crossing a road in Arizona

https://www.nytimes.com/2018/03/19/technology/uber-driverless-fatality.html

NTSB Report

- The system first registered radar and LIDAR observations of the pedestrian about 6 seconds before impact
- Software classified the pedestrian as an unknown object, as a vehicle, and then as a bicycle with varying expectations of future travel path.
- At 1.3 seconds before impact, the system determined that an en mane
- Emergency braking maneuvers are not enabled while the vehicle is under computer control, to reduce the potentia for erratic vehicle behavior



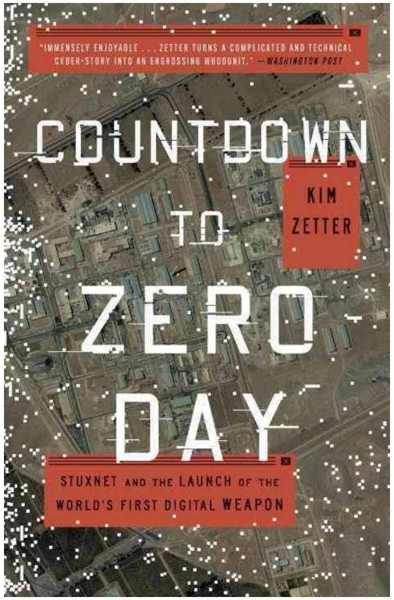
Failures in CPS have consequences

https://www.ntsb.gov/investigations/AccidentReports/Reports/HWY18MH010-prelim.pdf

KU THE UNIVERSITY OF KANSAS

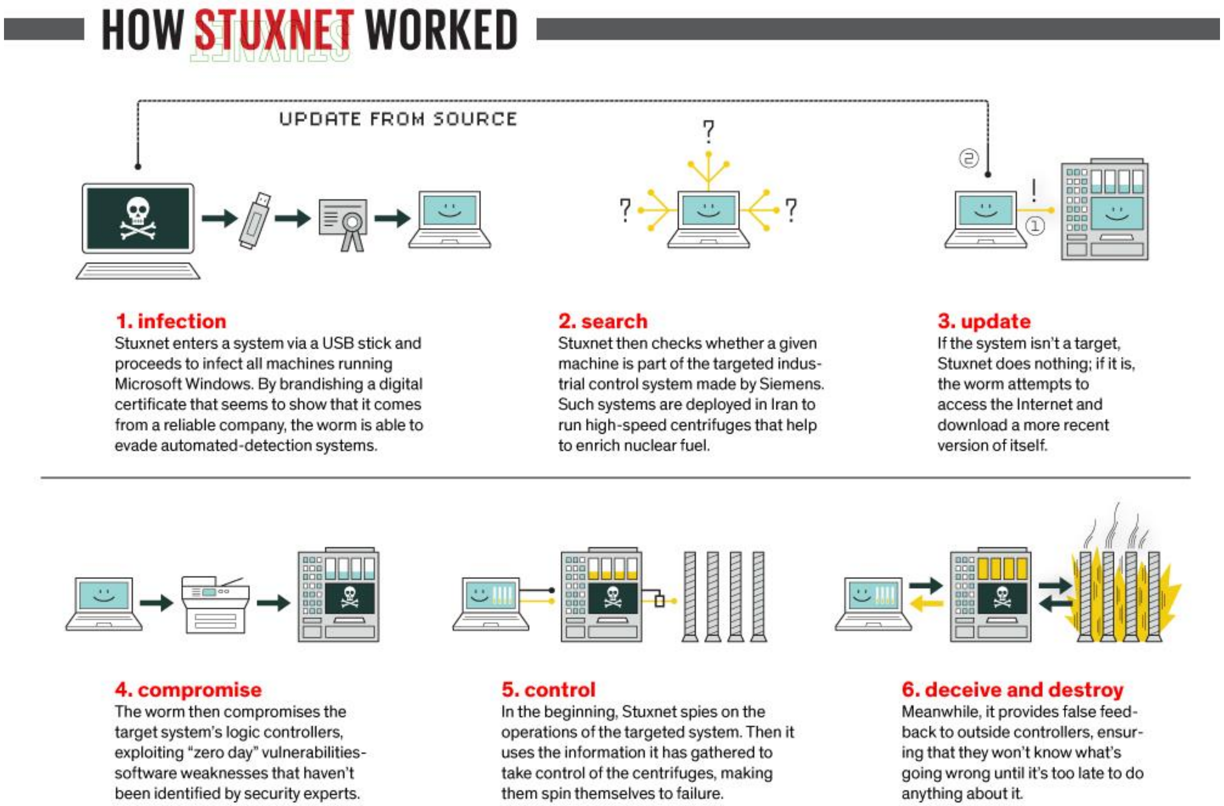
Security

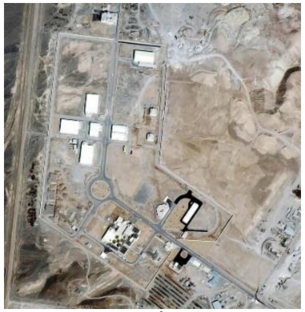
- CPS must be secure
  - Should prevent malicious access/use of the system
- But many CPS are open to various attacks
  - Networked CPS are especially vulnerable
- Examples
  - Stuxnet: Iranian nuclear power plant hacking
  - Vermont power grid hack by Russia
  - Remote hack into cars (Jeep)
  - Police drone hacking
  - Sensor hacking: GPS spoofing. IMU spoofing



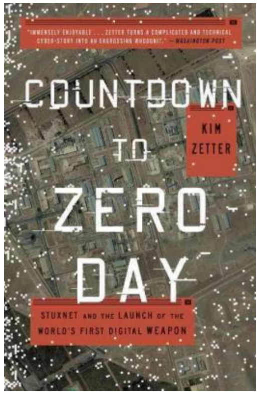
Stuxnet (2010)

- The first known cyber weapon
  - Modify centrifuges' rotation speed to their destruction





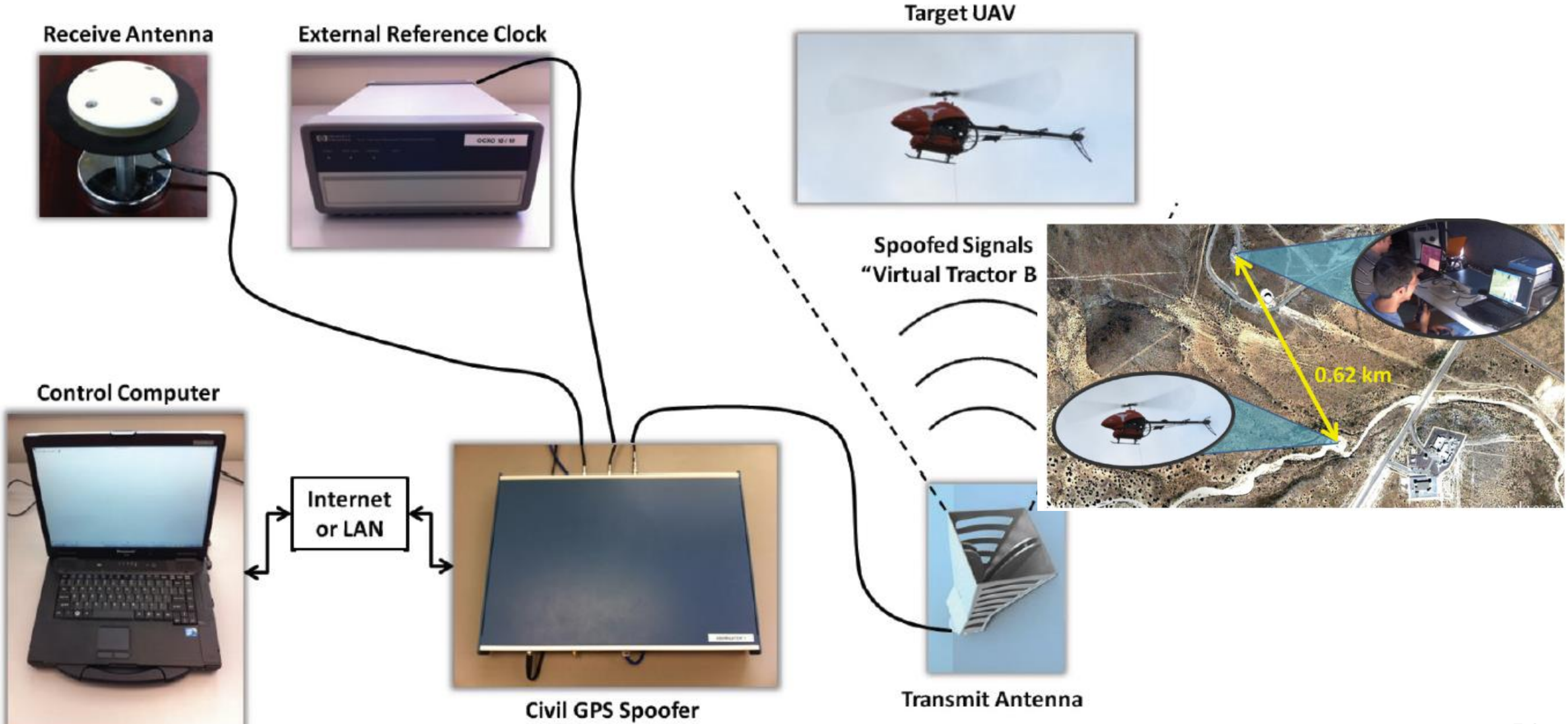
Targeted Iranian nuclear facility



https://spectrum.ieee.org/telecom/security/the-real-story-of-stuxnet

Drone GPS Spoofing (2012)

- Fool GPS sensors
  - Attacker can control the trajectory of the UAV



https://radionavlab.ae.utexas.edu/images/stories/files/papers/drone_hack_shepard.pdf

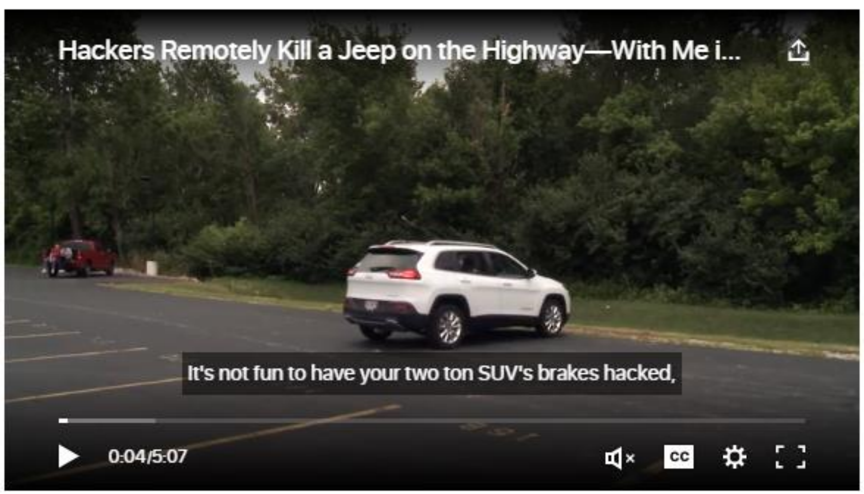
Remote Attack on Jeep (2015)

- Able to remotely (via cellular network) control steering, brake, and other critical functions via the car's infotainment system

ANDY GREENBERG SECURITY 07.21.15 06:00 AM

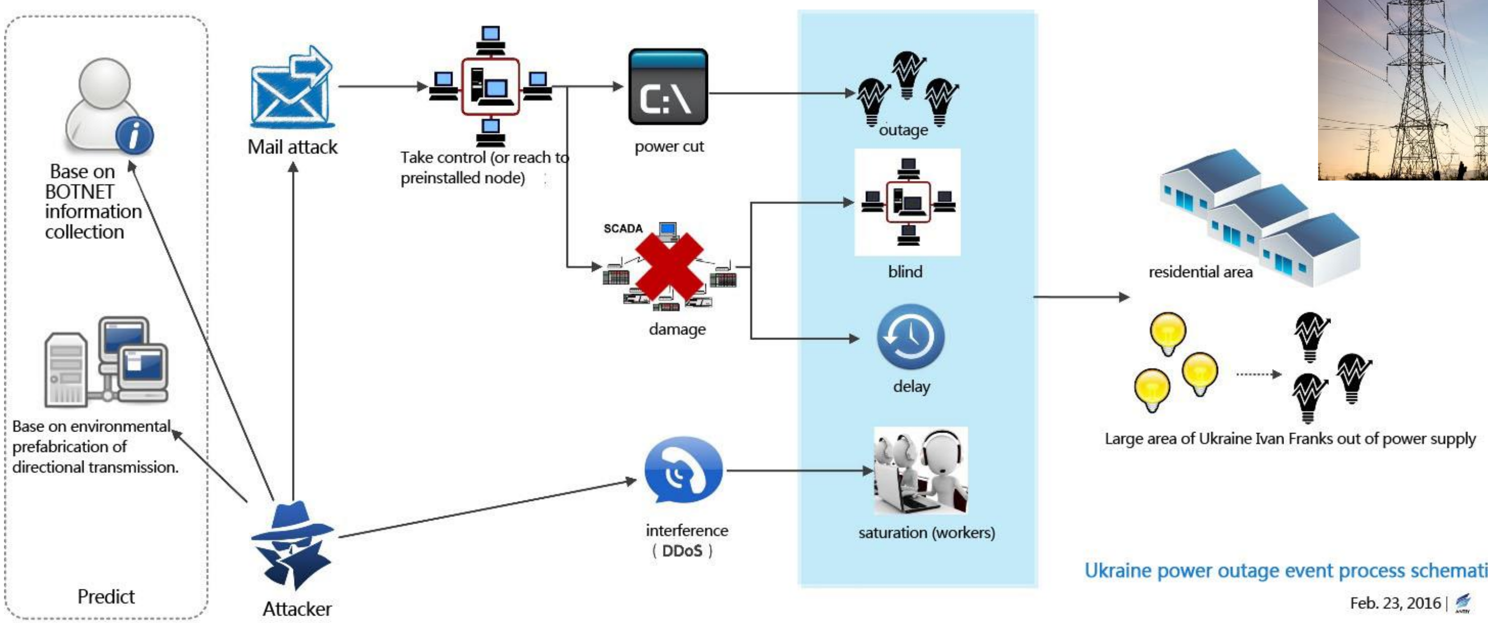
HACKERS REMOTELY KILL A JEEP ON THE HIGHWAY—WITH ME IN IT

https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/



Ukraine Power Grid Attack (2016)

- Attack on SCADA control network of a power grid in Ukraine, causing blackout on 80K users.



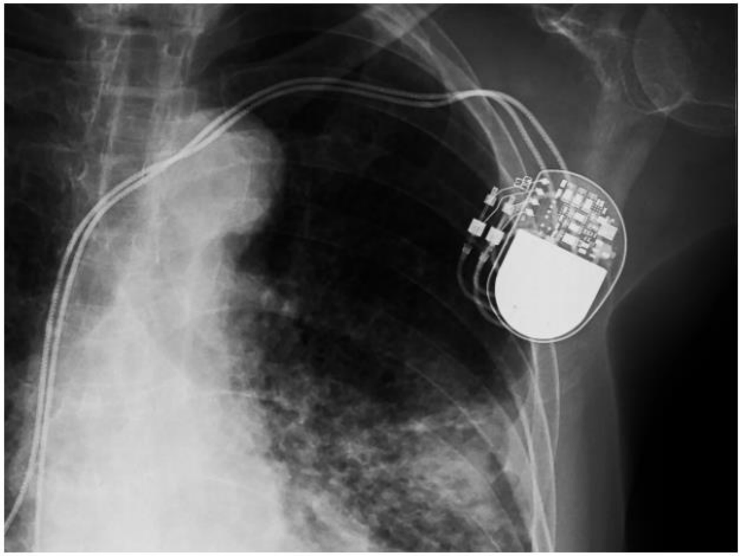
https://www.antiy.net/p/comprehensive-analysis-report-on-ukraine-power-system-attacks/

KU THE UNIVERSITY OF KANSAS

Pacemaker Hack (2017,2018)

A New Pacemaker Hack Puts Malware Directly on the Device

Researchers at the Black Hat security conference will demonstrate a new pacemaker-hacking technique that can add or withhold shocks at will.



CHOO CHIN/GETTY IMAGES

https://www.wired.com/story/pacemaker-hack-malware-black-hat/

Hacking risk leads to recall of 500,000 pacemakers due to patient death fears

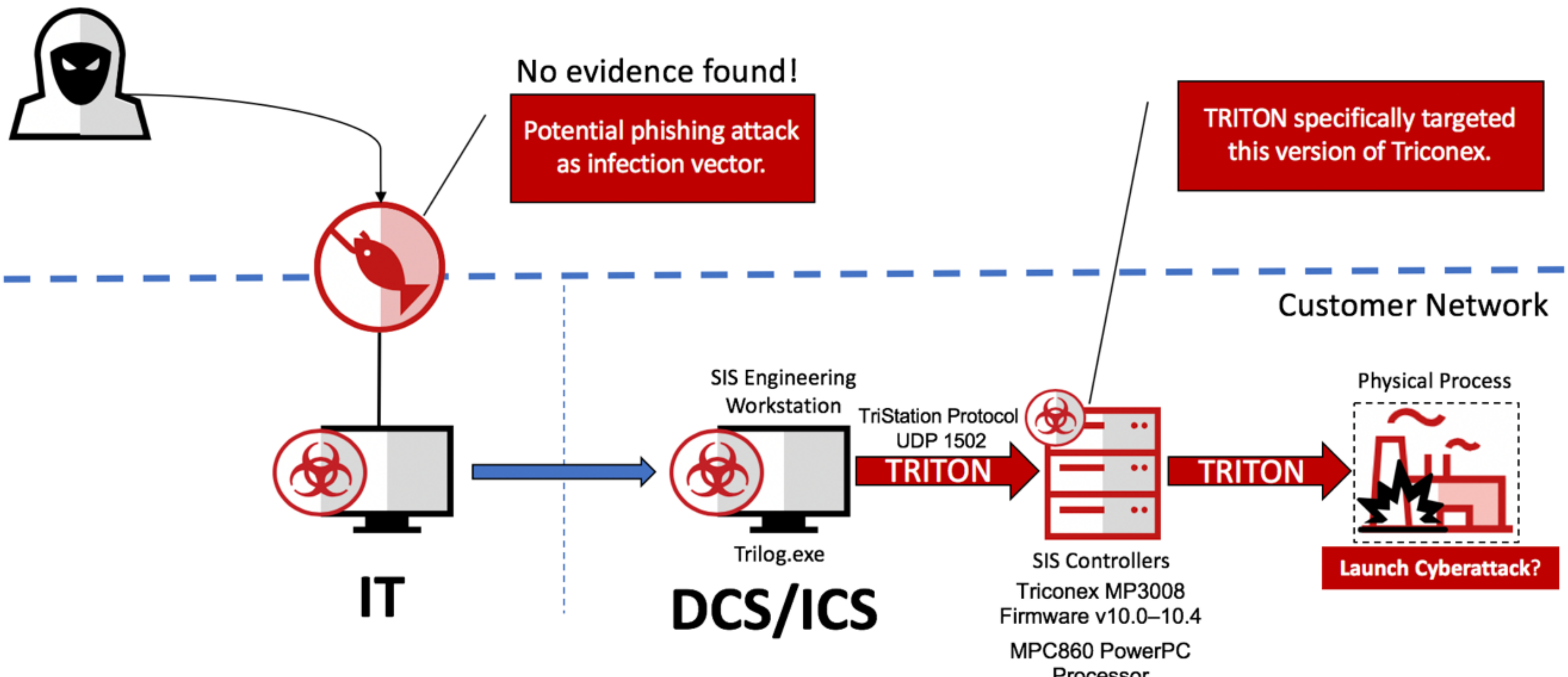
FDA overseeing crucial firmware update in US to patch security holes and prevent hijacking of pacemakers implanted in half a million people

▲ Abbott / St Jude Medical's Accent MRI pacemaker, one of the affected devices that had to be recalled. Photograph: Abbott / St Jude Medical

https://www.theguardian.com/technology/2017/aug/31/hacking-risk-recall-pacemakers-patient-death-fears-fda-firmware-update

Triton (2018)

- Attack safety systems of industrial control systems
  - Target an oil plant in Saudi Arabia



https://securingtomorrow.mcafee.com/other-blogs/mcafee-labs/triton-malware-spearheads-latest-generation-of-attacks-on-industrial-systems/

IoT WiFi Attacks (2019)

ESP8266 AND ESP32 WIFI HACKED!

by: Elliot Williams

37 Comments

September 5, 2019

[Matheus Garbelini] just came out with three (3!) different WiFi attacks on the popular ESP32/8266 family of chips. He notified Espressif first (thanks!) and they've patched around most of the vulnerabilities already, but if you're running software on any of these chips that's in a critical environment, you'd better push up new firmware pretty quick.

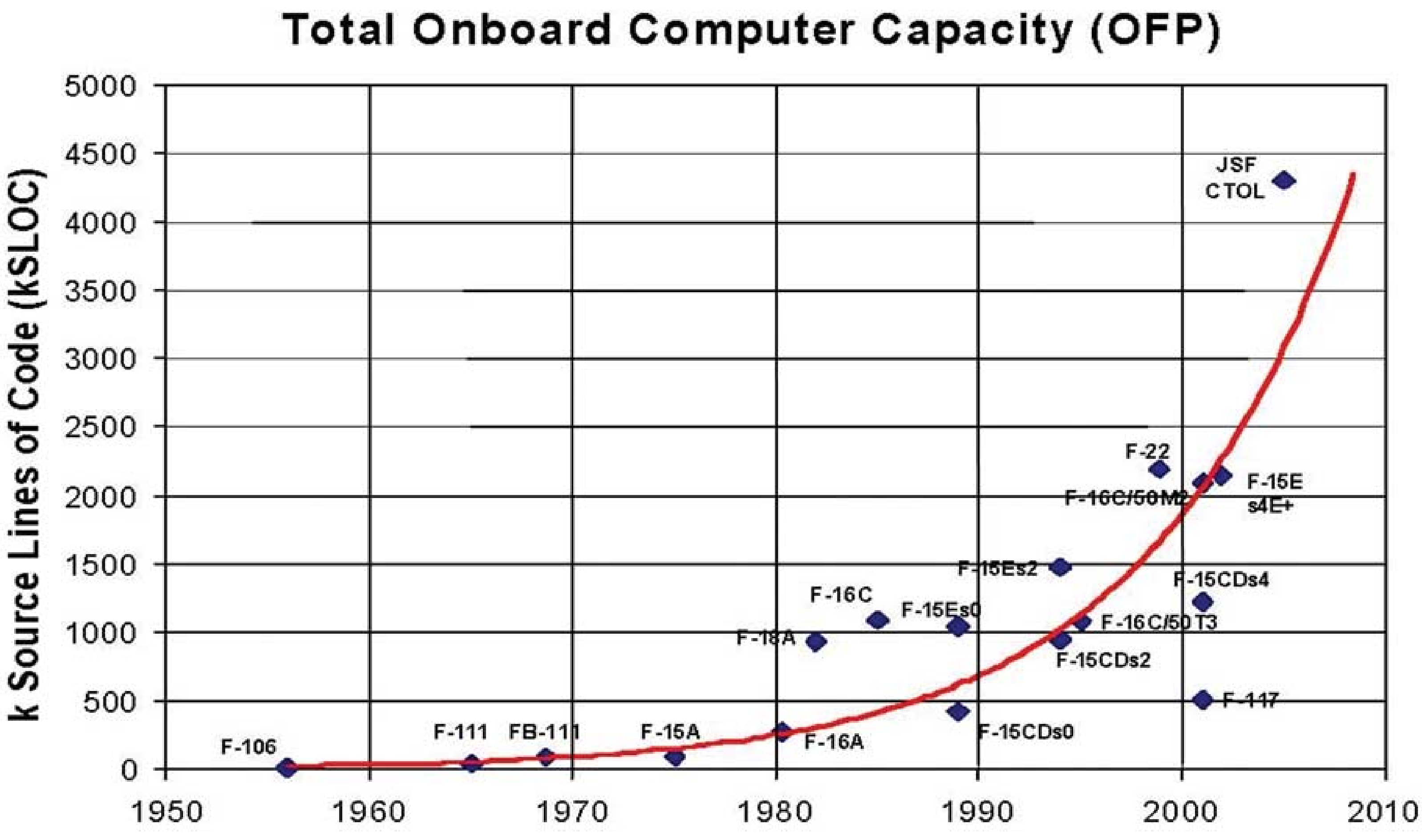
https://hackaday.com/2019/09/05/esp8266-and-esp32-wifi-hacked/

KU THE UNIVERSITY OF KANSAS

Challenges

- Complexity
- Reliability
- Security
- Time Predictability

Complexity



Total Onboard Computer Capacity (OFP)
k Source Lines of Code (kSLOC)
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1950
1960
1970
1980
1990
2000
2010
JSF
CTOL
F-22
F-15E
s4E+
F-16C/50M2
F-15Es2
F-15CDs4
F-16C
F-15Es0
F-18A
F-16C/50T3
F-15CDs2
F-117
F-106
F-111
FB-111
F-15A
F-16A
F-15CDs0

Complexity

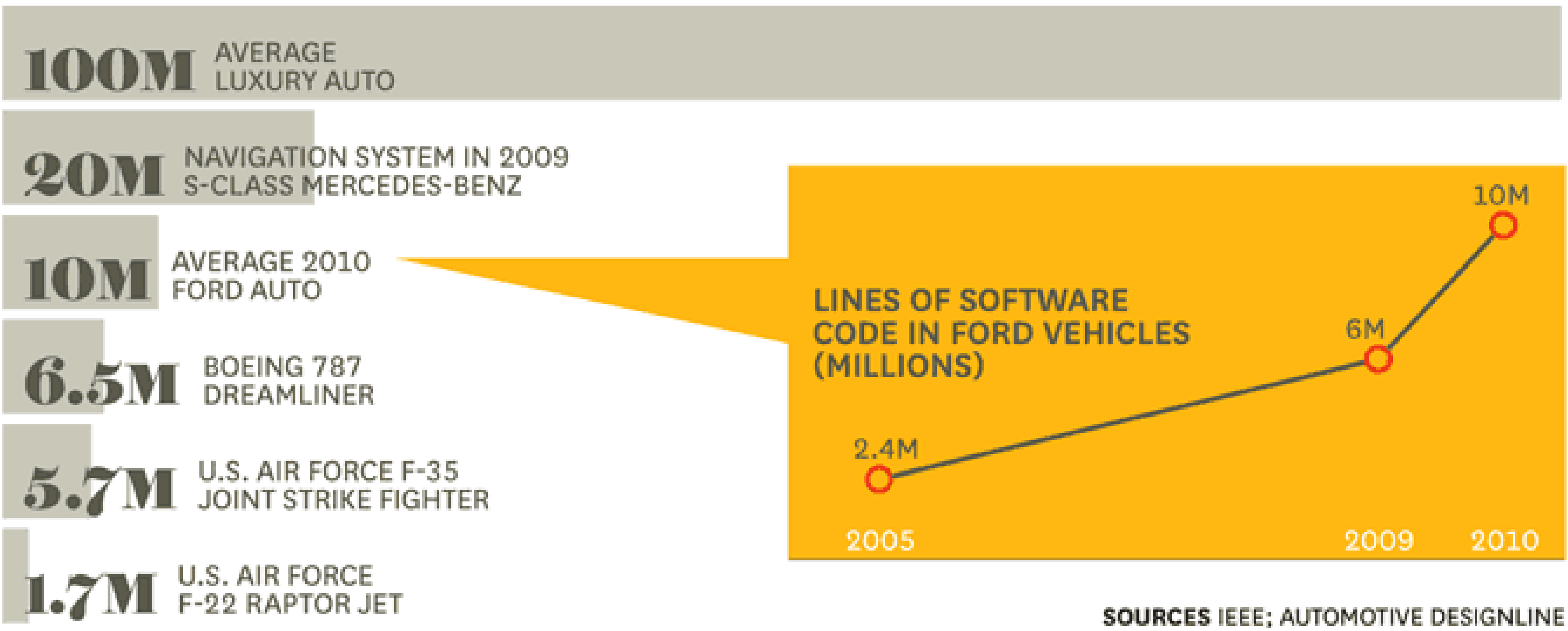


Image source: https://hbr.org/resources/images/article_assets/hbr/1006/F1006A_B_lg.gif

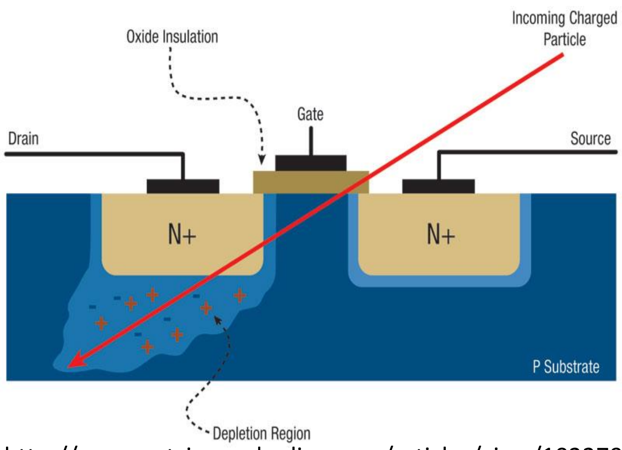
Example: F-22

- In 2007, 12 F-22s were going from Hawaii to Japan.
- After crossing the IDL, all 12 experienced multiple crashes.
  - No navigation
  - No fuel subsystems
  - Limited communications
  - Rebooting didn't help
- F-22 has 1.7 million lines of c

Complex software is hard to write and verify

Reliability

- Transient hardware faults
  - Single event upset (SEU)
    - Due to alpha particle, cosmic radiation
  - Manifested as software failures
    - Crashes
    - Silent data corruption (wrong output)
  - Bigger problem in advanced CPU
    - Increased density, frequency → higher chance for transient faults



http://www.cotsjournalonline.com/articles/view/102279

Example: SRAM Soft Error Rate (SER)

- SRAM SER vs. technology scaling
  - Per-bit SER decreases
  - Per-chip SER increases (due to higher density)

Design rule	SER (A.U)		MCUratio	MCU maximum size	Maximum bit multiplicity
nm	per device	per Mbit	%	bit	bit
130	1	1	5.8	182	10
90	1.9	0.94	13.5	2790	15
65	3.1	0.77	18.2	110860	19
45	4.3	0.53	26.4	118665	42
32	5.8	0.36	37	1933244	53
22	6.7	0.21	42.6	1075296	174

Ibe et al., “Scaling Effects on Neutron-Induced Soft Error in SRAMs Down to 22nm Process” (Hitachi)

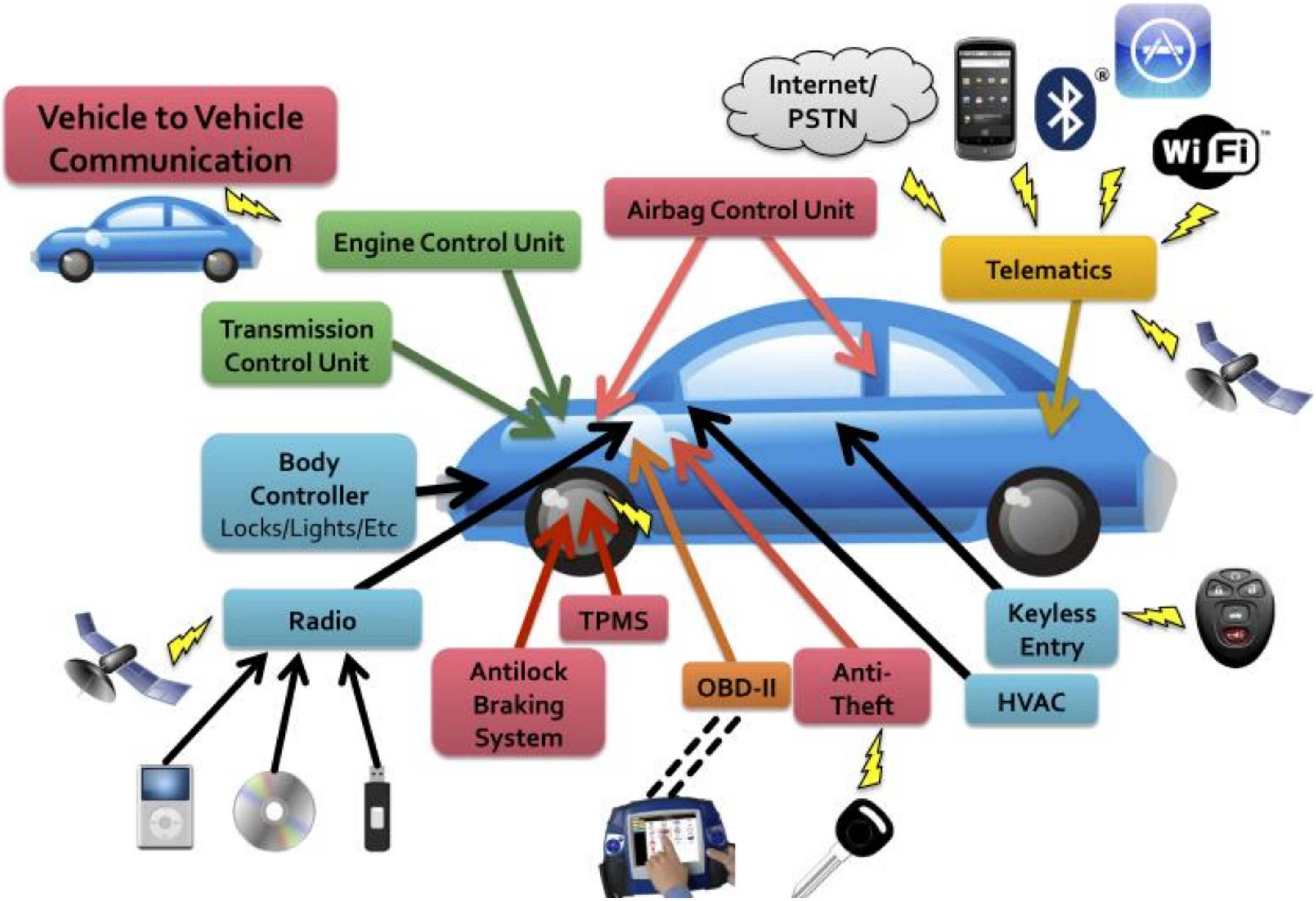
Complex hardware may be less reliable

KU THE UNIVERSITY OF KANSAS

Security

- General principles
  - Confidentiality: no data disclosure to unauthorized parties
  - Integrity: data cannot be modified by unauthorized parties
  - Availability: data/system must be available when needed
  - Safety: do no physical harm (critical in CPS)
- Defender's disadvantage
  - An attackers needs to find one vulnerability; while the defender needs to prevent ALL vulnerabilities.
- Challenges
  - Many access vectors
  - Many attack techniques

Access Vectors



Comprehensive Experimental Analyses of Automotive Attack Surfaces, USENIX Security, 2011

Goto Fail Bug

JPMorgan Chase and Co. [US] | chase.com

iOS 7.0.6
Data Security
Available for: iPhone 4 and later, iPod touch (5th generation), iPad 2 and later

Impact: An attacker with a privileged network position may capture or modify data in sessions protected by SSL/TLS

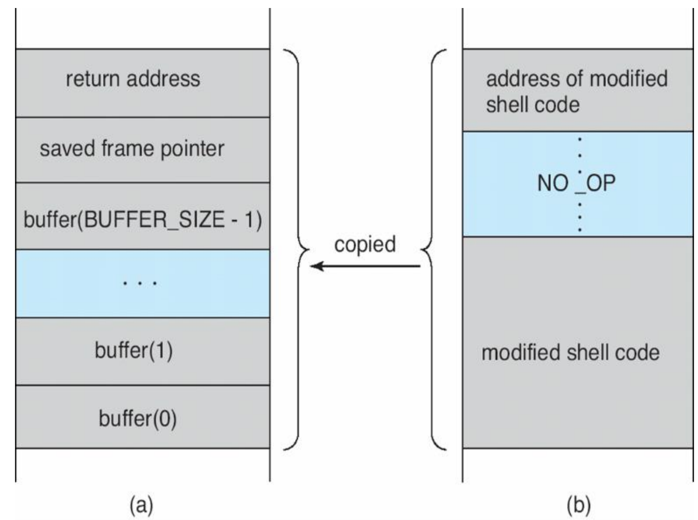
Description: Secure Transport *failed to validate the authenticity of the connection*. This issue was addressed by restoring missing validation steps.

```
    err = 0
    . . .
    hashOut.data = hashes + SSL_MD5_DIGEST_LEN;
    hashOut.length = SSL_SHA1_DIGEST_LEN;
    if ((err = SSLFreeBuffer(&hashCtx)) != 0)
     goto fail;
    if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
     goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
     goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
     goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
     goto fail;
     goto fail;      <-- MISTAKE! THIS LINE SHOULD NOT BE HERE
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
     goto fail;

    err = sslRawVerify(...); // This code must be executed
    . . .
fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    Return err;
```

Buffer Overflow

- What is wrong with this code?

```
#define BUFFER_SIZE 256
int process_args(char *arg1)
{
    char buffer[BUFFER SIZE];
    strcpy(buffer,arg1);
    ...
}

int main(int argc, char *argv[])
{
    process_args(argv[1]);
    ...
}
```



Before

After executing *strcpy(buffer, **arg1**)*

Linux Kernel Buffer Overflow Bugs

6	CVE-2010-2521	119	DoS Exec Code Overflow	2010-09-07	2012-03-19	10.0	None	Remote	Low	Not required	Complete	Complete Complete

Multiple buffer overflows in fs/nfsd/nfs4xdr.c in the XDR implementation in the NFS server in the Linux kernel before 2.6.34-rc6 allow remote attackers to cause a denial of service (panic) or possibly execute arbitrary code via a crafted NFSv4 compound WRITE request, related to the read_buf and nfsd4_decode_compound functions.

9	CVE-2009-0065	119	Overflow	2009-01-07	2012-03-19	10.0	Admin	Remote	Low	Not required	Complete	Complete Complete

Buffer overflow in net/sctp/sm_statefuns.c in the Stream Control Transmission Protocol (sctp) implementation in the Linux kernel before 2.6.28-git8 allows remote attackers to have an unknown impact via an FWD-TSN (aka FORWARD-TSN) chunk with a large stream ID.

10	CVE-2008-5134	119	Overflow	2008-11-18	2012-03-19	10.0	None	Remote	Low	Not required	Complete	Complete Complete

Buffer overflow in the lbs_process_bss function in drivers/net/wireless/libertas/scan.c in the libertas subsystem in the Linux kernel before 2.6.27.5 allows remote attackers to have an unknown impact via an "invalid beacon/probe response."

11	CVE-2008-3915	119	Overflow	2008-09-10	2012-03-19	9.3	None	Remote	Medium	Not required	Complete	Complete Complete

Buffer overflow in nfsd in the Linux kernel before 2.6.26.4, when NFSv4 is enabled, allows remote attackers to have an unknown impact via vectors related to decoding an NFSv4 acl.

12	CVE-2008-3496	119	Overflow	2008-08-06	2012-03-19	10.0	None	Remote	Low	Not required	Complete	Complete Complete

Buffer overflow in format descriptor parsing in the uvc_parse_format function in drivers/media/video/uvc/uvc_driver.c in uvcvideo in the video4linux (V4L) implementation in the Linux kernel before 2.6.26.1 has unknown impact and attack vectors.

13 CVE-200 … ete Complete

Complex software has lots of security bugs

KU THE UNIVERSITY OF KANSAS

Software Attacks on Hardware

Meltdown

Meltdown breaks the most fundamental isolation between user applications and the operating system. This attack allows a program to access the memory, and thus also the secrets, of other programs and the operating system.

Spectre

Spectre breaks the isolation between different applications. It allows an attacker to trick error-free programs, which follow best practices, into leaking their secrets. In fact, the safety checks of said best practices actually increase the attack surface and may make applications more susceptible to Spectre

https://meltdownattack.com/

Micro-Architectural Side-Channels

- Many micro-architectural components contain hidden state which leaks secret
  - often via observable *timing* variations
- Known to exist in cache, DRAM bank, OoO speculation, branch predictor, etc.
- **Logically correct, proven software is also vulnerable**

Spectre Attack

```
if (x < array1_size)
    y = array2[array1[x] * 256];
```

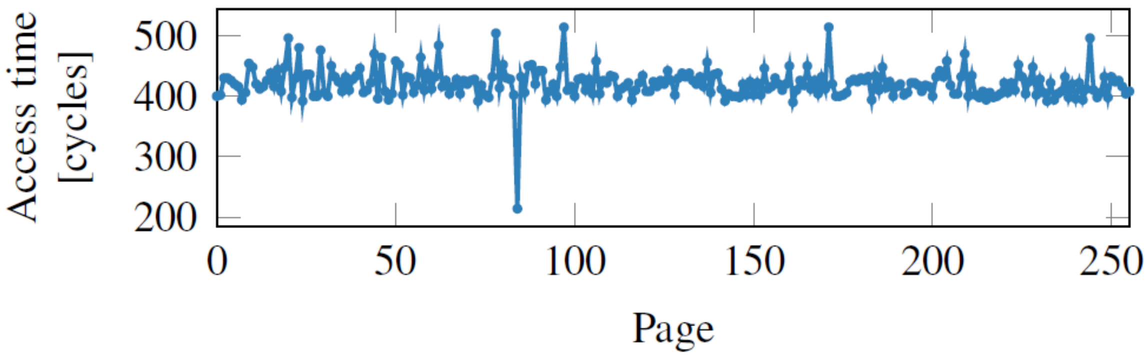
Listing 1: Conditional Branch Example

- Wrong branch is speculatively taken.
- *x* is maliciously chosen by the attacker.
- The attacker probes *arrary2* to recover secret: *array1[x]*

KU THE UNIVERSITY OF KANSAS

(Cache) Timing Channel Attack

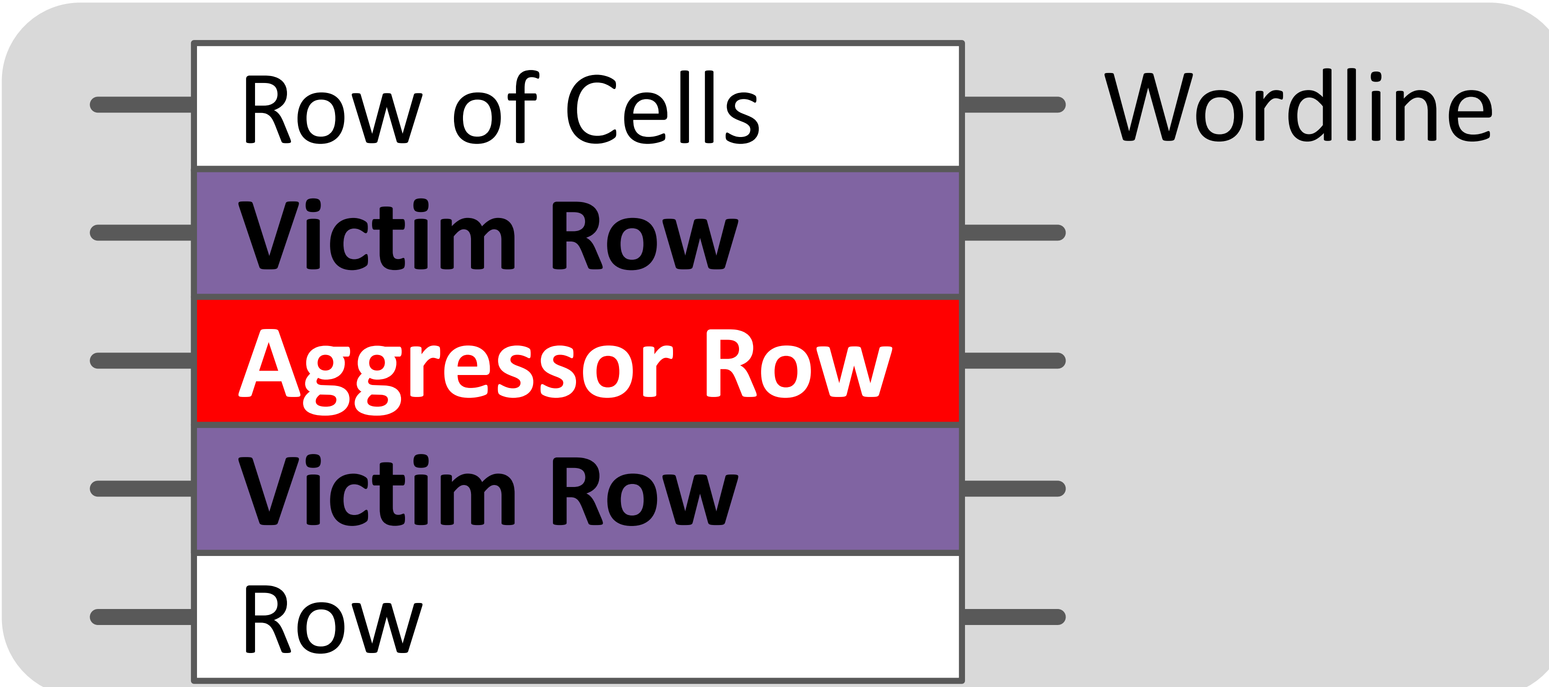
- By measuring access timing differences of a memory location, an attacker can determine whether the memory is cached or not.



- This can be used to **leak secret information**
- Methods: Flush + Reload, Prime + Probe, etc.

Image source: M. Lipp et al., “Meltdown,” arXiv Prepr., 2018.

RowHammer Attacks



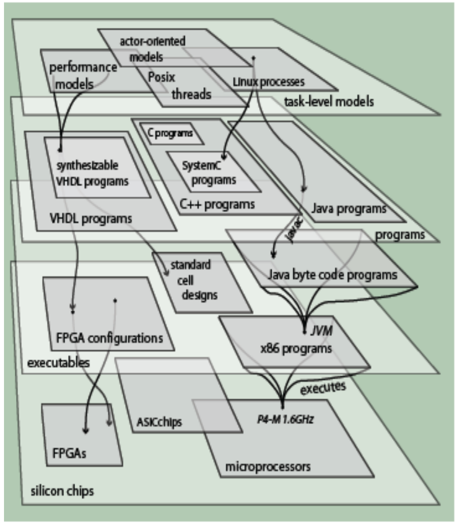
- **R**epeatedly opening and closing a DRAM row can induces **bit flips** in adjacent rows storing sensitive

Complex hardware may not be secure

Credit: This slide is from Dr. Yoongu Kim's presentation slides of the following paper:
"Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," In *ISCA*, 2014

Time Predictability

- At low-level, hardware is deterministic timing
- At higher-levels, not so much→ ignore timing
  - Pipeline, caches, Out-of-order execution, speculation, ISA
  - Process, thread, lock, interrupt
- Focus on average case, not worst-case. No guarantees
  - Fine in cyber world
  - Real-world doesn't work that way

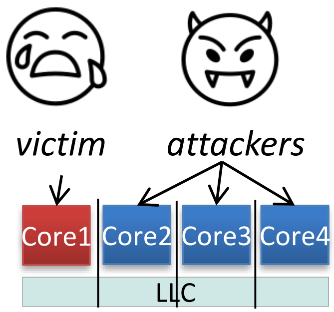


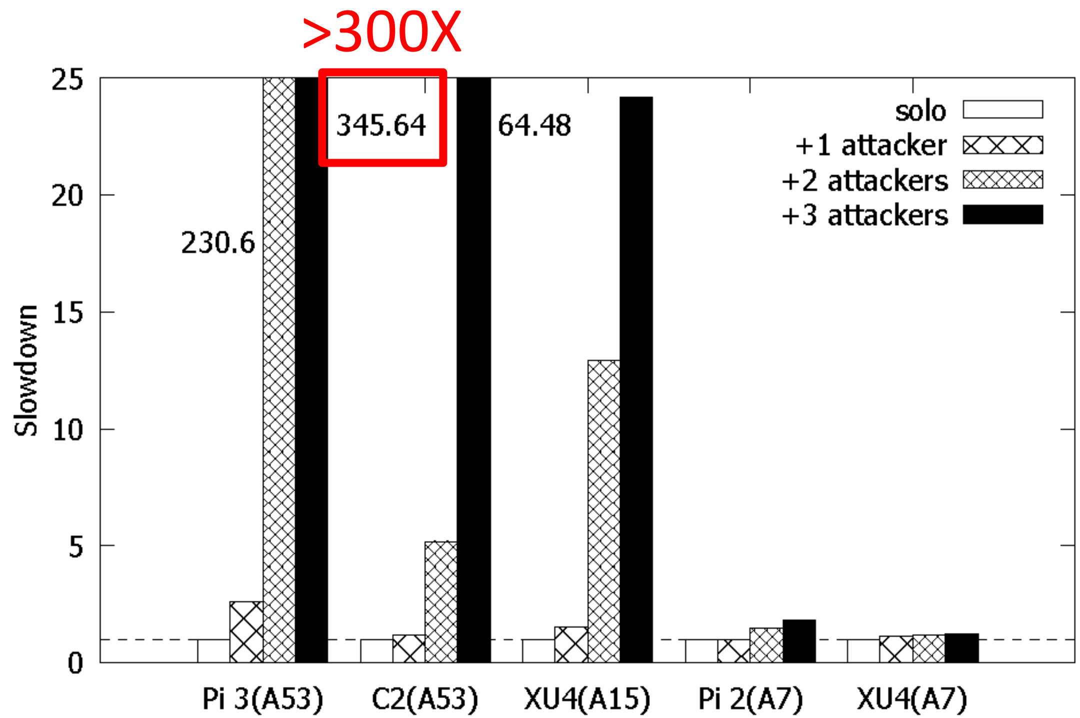
From Dr. Edward A. Lee, UCB

Timing Predictability

- Q. Can you tell exactly how long a piece of code will take to execute on a computer?
  - Used to be (relatively) easy to do so.
    - Measure timing. Use the timing for analysis.
  - Very difficult to answer in today's computers
    - Pipeline, cache, out-of-order and speculative execution, multicore, shared cache/dram→ very **high variance**.

Cache Denial-of-Service Attacks

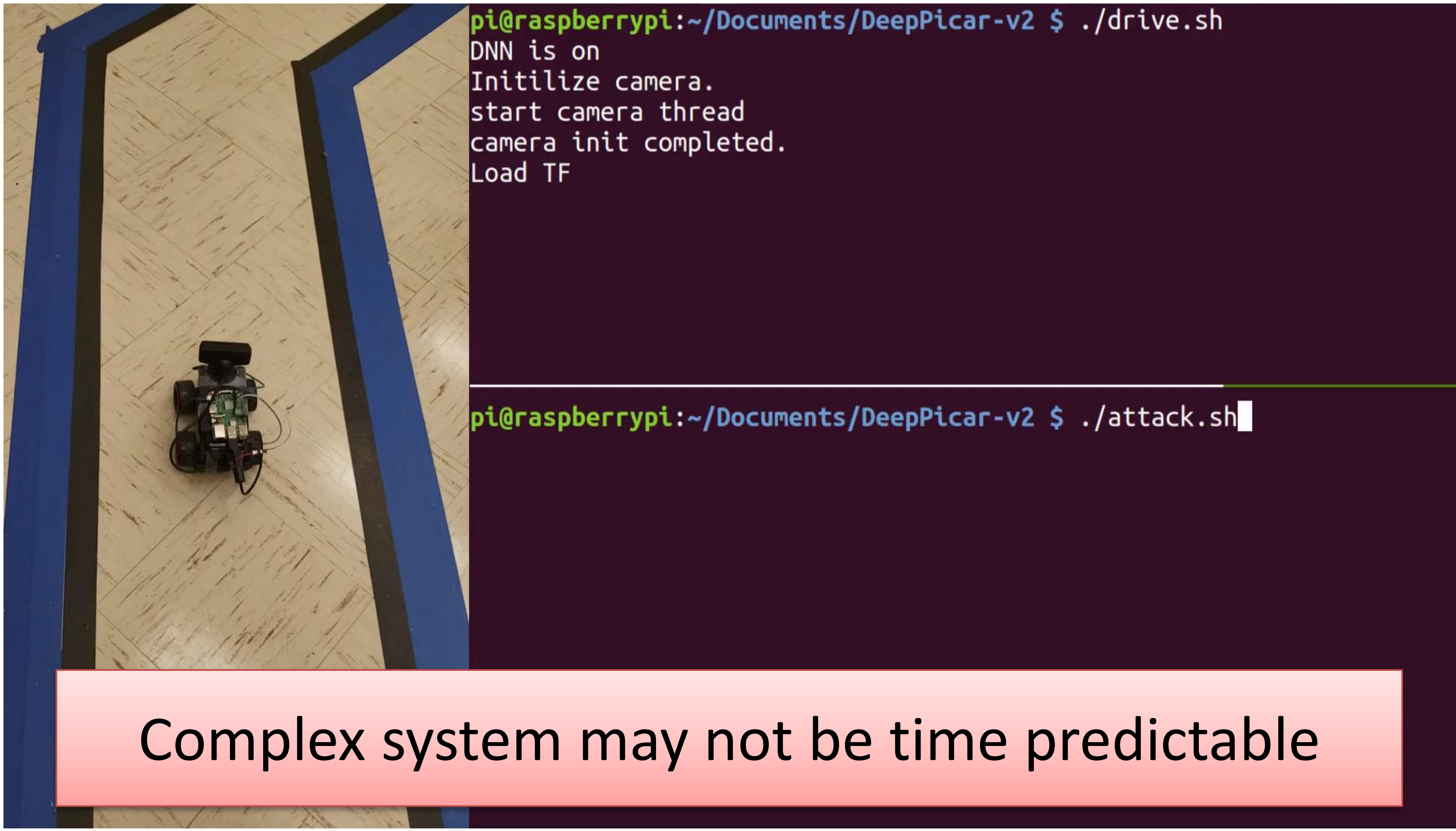




- Observed worst-case: >300X (times) slowdown
  - On popular in-order multicore processors
  - Due to contention in cache write-back buffer

M. G. Bechtel and H. Yun. “Denial-of-Service Attacks on Shared Cache in Multicore: Analysis and Prevention.” In *RTAS*, 2019

Safety and Real-Time



Complex system may not be time predictable

https://youtu.be/Jm6KSDqlqiU

Related Areas

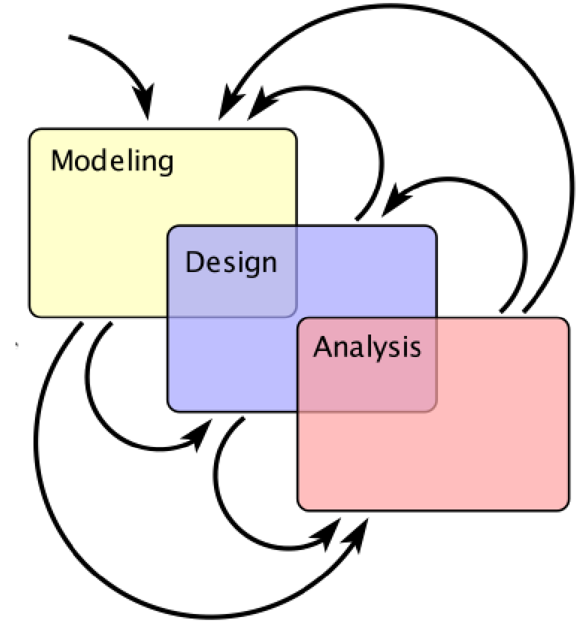
- CPS/embedded systems development requires inter disciplinary approach
  - EECS (on cyber systems)
    - Computer architecture
    - Real-time systems
    - Formal method
    - Software engineering
  - Aerospace, and other engineering (on physical)
    - Physical systems (plant/actuator) modeling/control

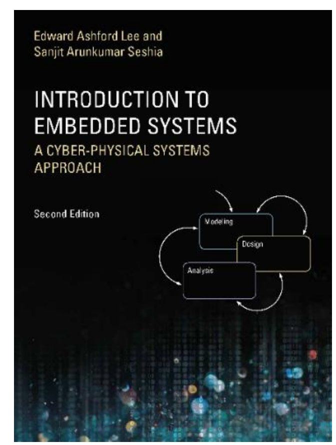
THE UNIVERSITY OF KANSAS

Topics

- Focus on **design**

- CPU & memory
- I/O interface
- Sensors & actuators
- Interrupt & multitasking
- Real-time scheduling
- Advanced topics



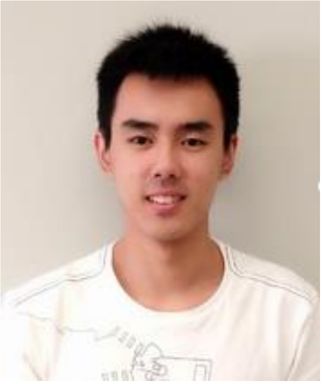


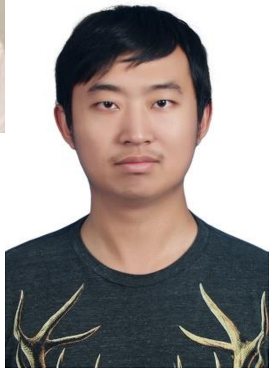
Summary

- Embedded systems
  - Purpose built systems
  - Everywhere as more "things" are computerized
  - Related terminologies: Cyber-Physical Systems (CPS), real-time systems, Internet-of-things (IoT)
  - Efficiency, safety, security are essential but difficult
- This course
  - Learn concepts and skills to develop embedded systems.

Crew

- **Teaching Assistants**
  - Yiju Yang
    - Email: y150y133@ku.edu
    - Office hours: TBD
    - Office: 3002 EATON
  - Xiaohan Zhang
    - Email: speed1224@ku.edu
    - Office hours: TBD
    - Office: 3002 EATON
  - Arin Dutta
    - Email: arindutta40@ku.edu
    - Office hours: TBD
    - Office: 3002 EATON





Appendix