Embedded Systems

Learning Objectives

- **Define what an embedded system is in terms of its interface**
- **Enumerate and describe the components of an embedded system**
- **Describe the interactions of embedded systems with the physical world**

Embedded Systems – Natual Interface with Hidden complexicity

For Example: Take the camera that we're filming with right in front of me. This camera, you could've gotten a camera 40 years ago that you can use to take a picture with, snap a button, take a picture. A new camera, an Internet of Things camera, is network, it's got computers inside, computational technology. The same interface, you snap the button and it takes the picture, but it does a lot more. It has a lot more enhanced features. Maybe it does jitters control or something like that, so it’s embedded within the device, but it's well hidden. It's embedded behind a nice, what we'll call natural interface.

Embedded Devices – Physical World

The device should conform to what the user wants, and it's to have an interface that makes it easy for the user to deal with. So, in order to do that, the assistance, they have to be embedded within regular devices, and they have to interact with the physical world through the use of sensors that read information about the physical world, maybe somebody pressed a button, a light turned on, something like that. Also, a set of actuators that cause something to happen in the physical world that basically change the state of the physical world, so maybe motors or lights or something.

What Are Embedded Systems?

Computer-based systems that do not appear to be computers – complexity is hidden from the user

What Are Embedded Systems?

They interact with users via simple interface

- Digital camera, TV, cellphone

What Are Embedded Systems?

They interact with another device, invisible to user

- Disk drive, memory stick, anti-lock braking system

Efficiency Rules!

- Most embedded products are in cost-critical markets (e.g. consumer electronics)
- Other applications are in performance or power critical markets (e.g. military, medical)

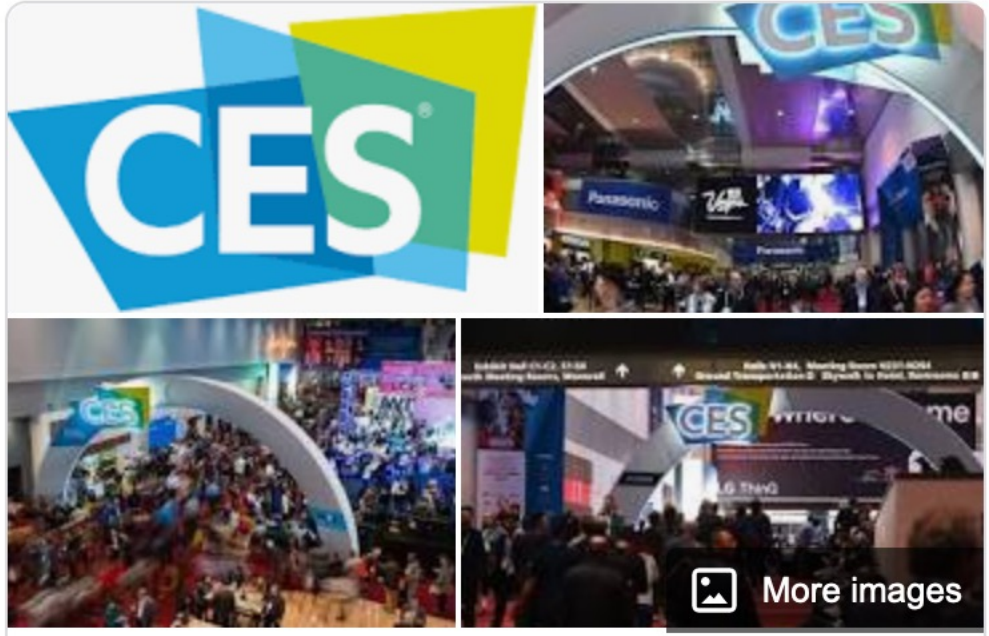
Tight Constraints

- Manufacturing cost
- Design cost
- Performance
- Power
- Time-to-market

Tight Constraints

Very different from traditional software engineering

- Moore’s law will save you eventually



CES

Convention

CES is an annual trade show organized by the Consumer Technology Association. Held in January at the Las Vegas Convention Center in Winchester, Nevada, United States, the event typically hosts presentations of new products and technologies in the consumer electronics industry. Wikipedia

Attendance: 182,000 (2019)

Application Specificity

Embedded systems tend to be application-specific

- Perform one task or set of related tasks
- Some devices blur the line (e.g. cell phones)

Application Specificity

Design is focused on one application -- unlike general-purpose systems (e.g. laptops)

Application Specificity

Higher design efficiency is possible

- Special-purpose vs. general purpose (e.g. video games)

Hardware/Software Codesign

Hardware and software are often designed together

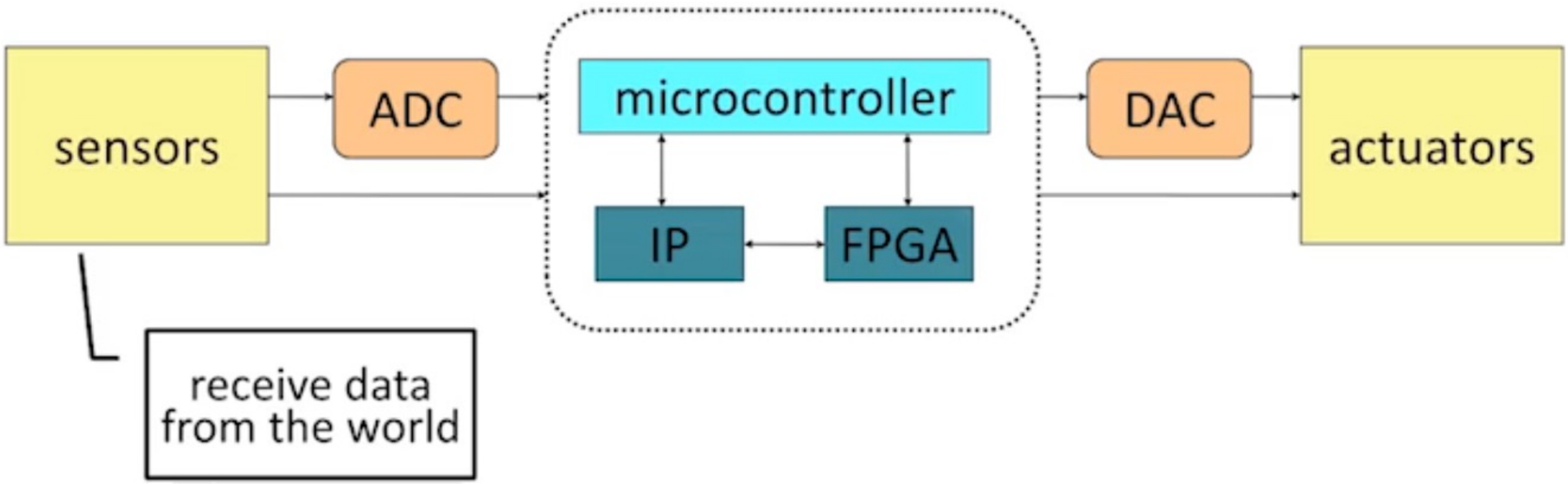
- General-purpose systems use hardware and software developed by different companies

Hardware/Software Codesign

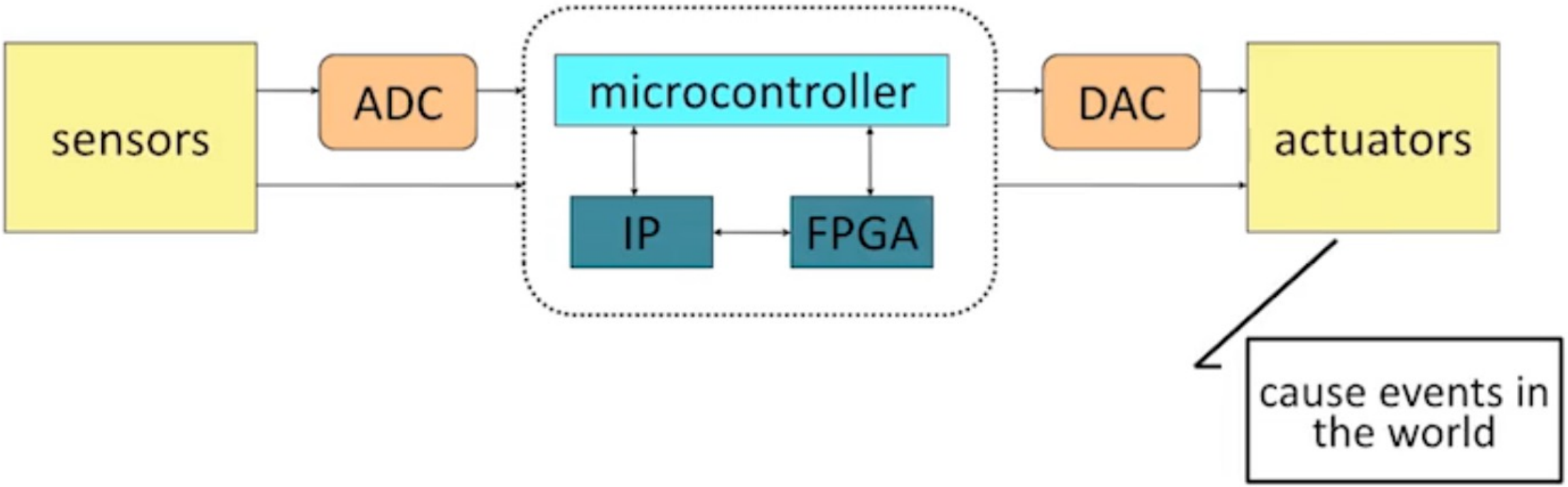
More work for the designers

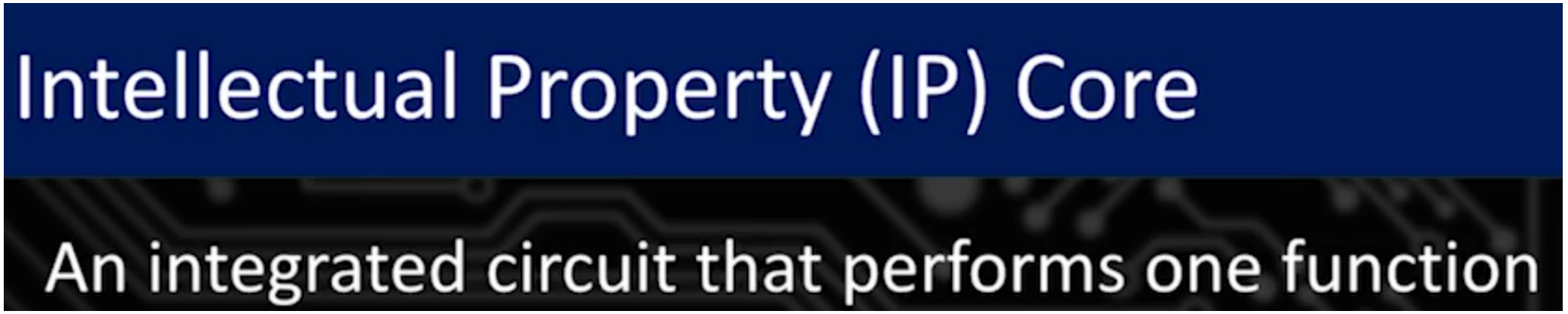
- Must understand both hardware and software

Generic Embedded System Structure



Generic Embedded System Structure




Intellectual Property (IP) Core
An integrated circuit that performs one function


Intellectual Property (IP) Core
Cheap in high volume

Intellectual Property (IP) Core

Very useful for common tasks

- Network controllers (Ethernet, CAN)
- Audio/video (audio codec, VGA controller)

Intellectual Property (IP) Core

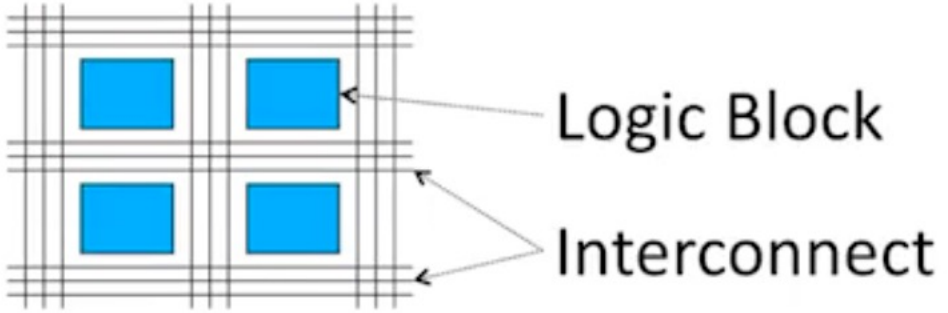
Must interact with the microcontroller

- Consider communication protocol

Field Programmable Gate Array (FPGA)

Hardware that can be reconfigured via RAM

- Faster than SW, slower than ASIC
- No fabrication needed

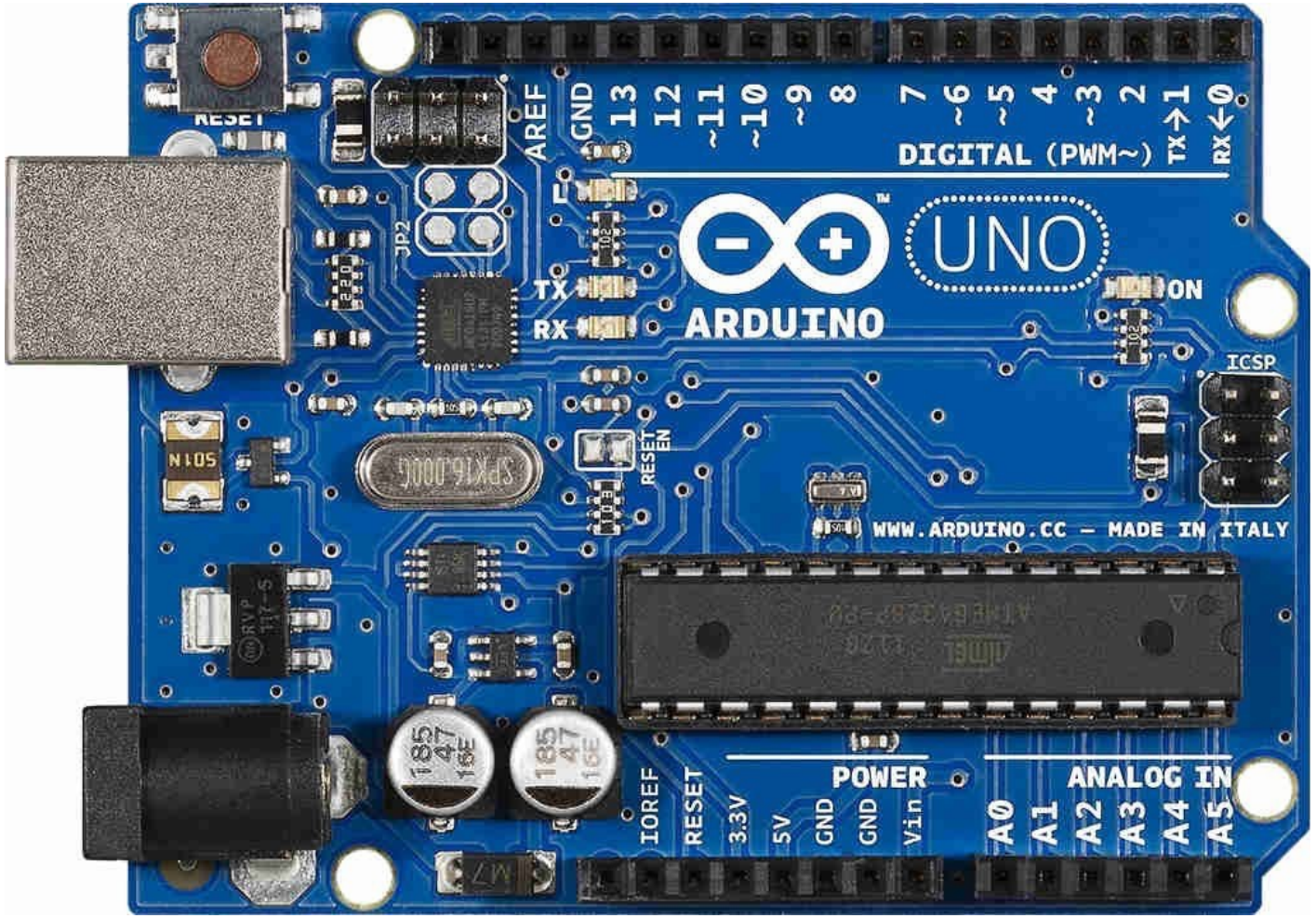


Hardware Discription Langauage

1. VERILOG HDL
2. VHDL

Application Spacific Integrared Circuits (ASIC)

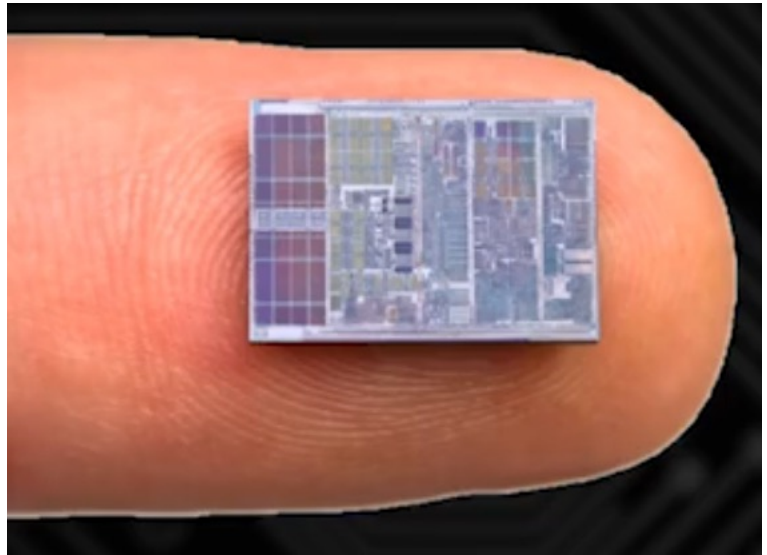
Microcontroller



Microcontrollers

Integrated circuit that executes a program

- Microcontroller vs. microprocessor
- Slower, 16MHz – 500MHz
- Less memory, fewer features



Microcontrollers

- Connected to other hardware components
- Sends commands and receives data

Microcontrollers

Needs to be programmed

- Write a program in a language, such as C
- Place the program in the mctrlr memory

Programming Microcontrollers

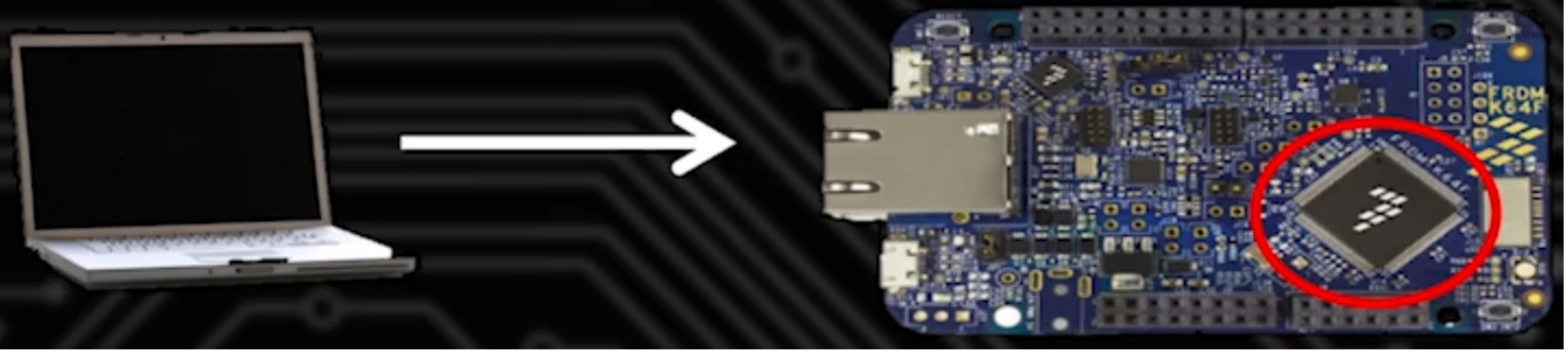
- Write code on a host machine – regular desktop or laptop
- Programming the microcontroller – transferring the program from host to microcontroller

Using a Programmer

Programming hardware can be used to place program in microcontroller memory

Using a Development Board

Includes hardware needed for programming



General Purpose Processors

- Used for any application
- Many features included

Digital Signal Processors

- Made to support DSP functions
- Vector instructions
- Cheaper but more limited

Sensors

Receive information from the environment

Provide simple information

- Thermistor: reports temperature
- Photoresistor: reports light intensity

- Light sensor
- Motion sensor
- Temperature sensor
- Magnetic fields sensor
- Gravity sensor
- Humidity sensor
- Moisture sensor
- Vibration sensor
- Pressure sensor
- Electrical fields sensor
- Sound sensor
- Position sensor

Complex Sensors

More complicated data

- CMOS camera: captures images
- Ethernet controller: enables network communication

Actuators

- Cause events to occur in the environment
- Simple actuators
  - Light-Emitting Diodes (LEDs) – small lights
  - LCD Display – simple display

Complex Actuators

- Servo motor: moves something
- Ethernet controller: enables network communication