Institute for the Wireless Internet of Things
at Northeastern University

EECE5155: Wireless Sensor Networks and the Internet of Things

Josep Miquel Jornet
Associate Professor, Department of Electrical and Computer Engineering
Director, Ultrabroadband Nanonetworking Laboratory
Member, Institute for the Wireless Internet of Things
Northeastern University
jmjornet@northeastern.edu
www.unlab.tech

Institute for the Wireless Internet of Things at Northeastern

Module T3: Local Data Processing

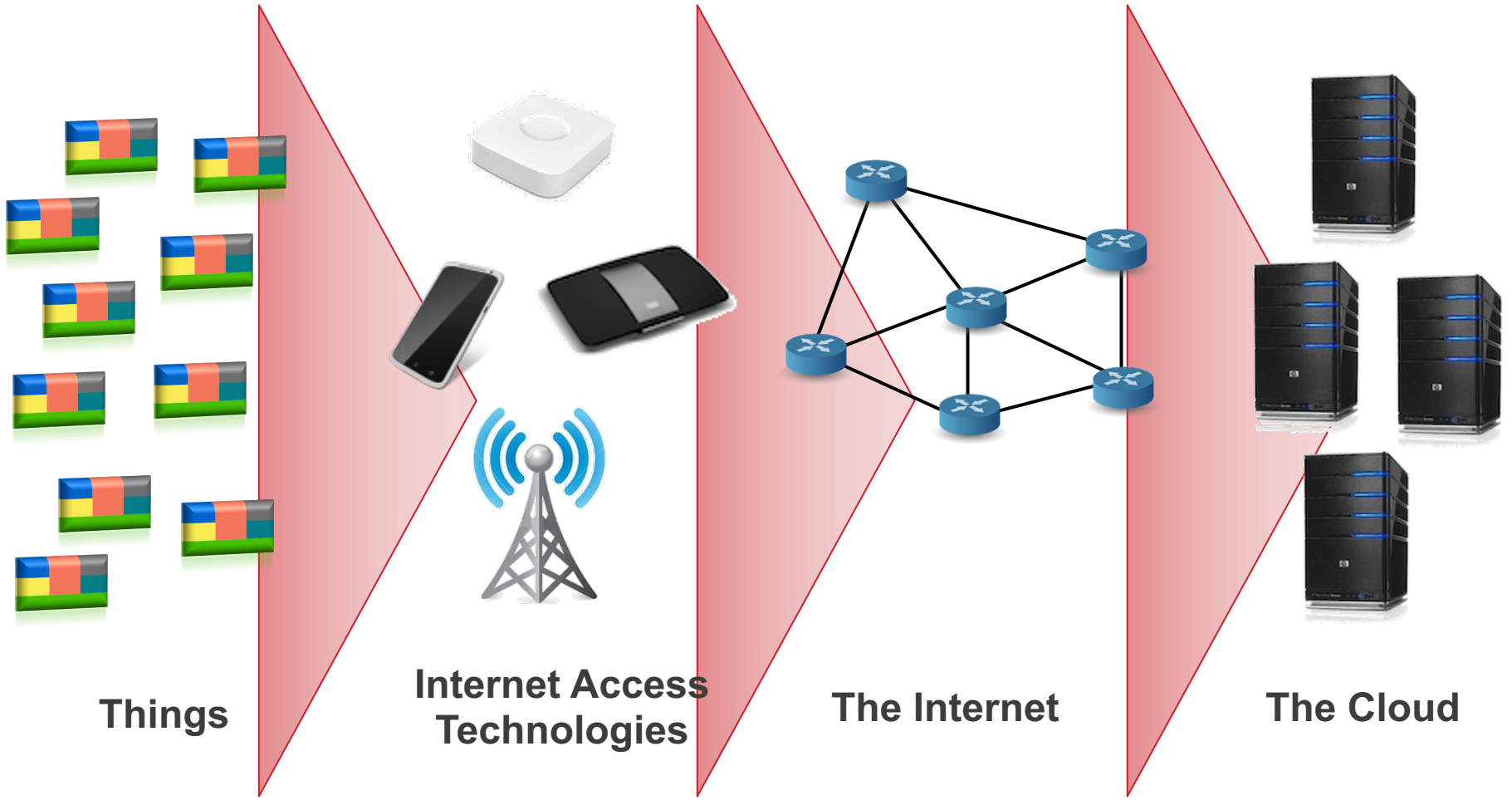
Outline

Institute for the Wireless Internet of Things at Northeastern

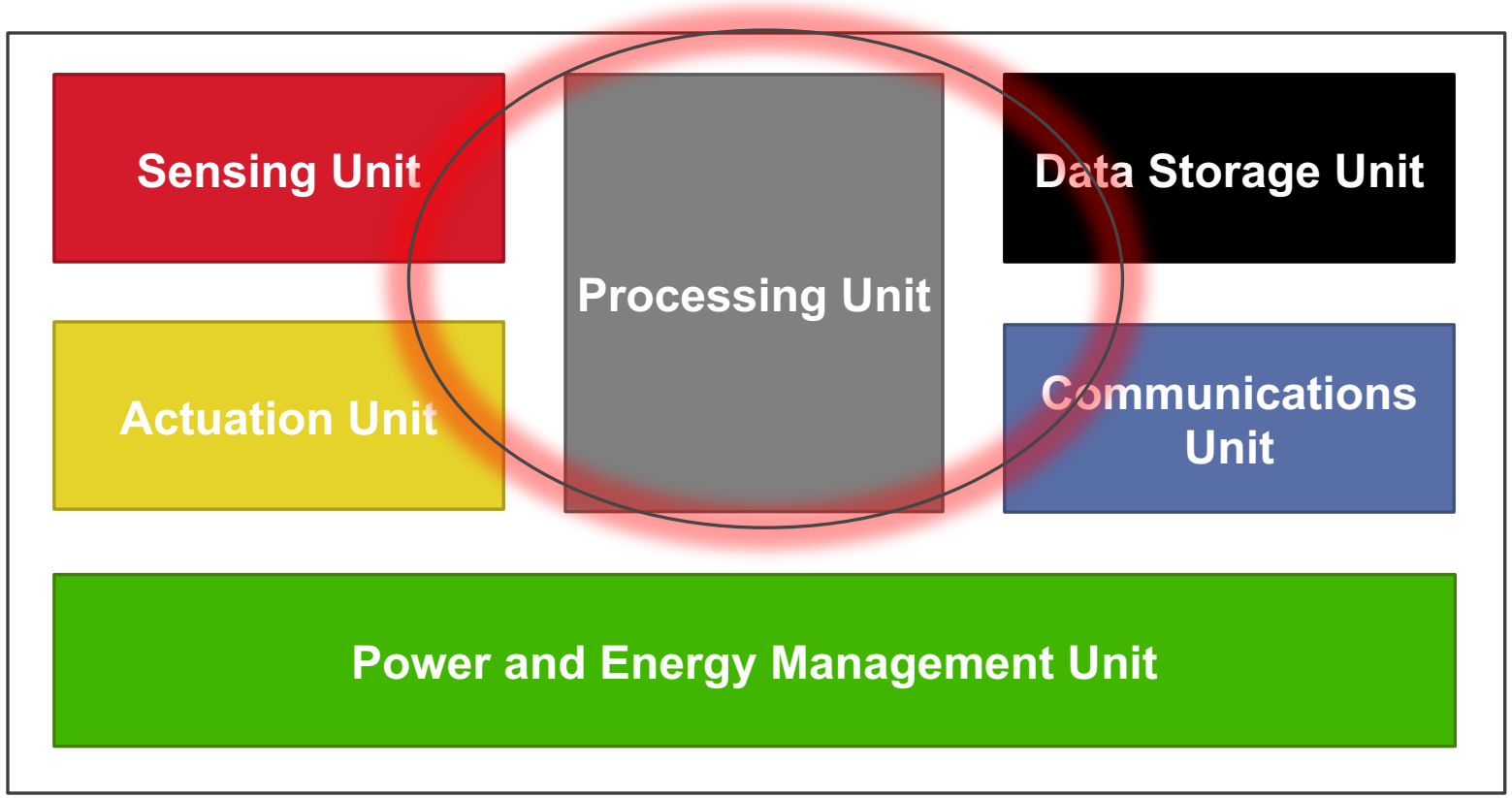
- **Introduction**
- **Computing Architectures**
- **Specifications and Trends**

Internet of Things: Abstraction

Institute for the Wireless Internet of Things at Northeastern



Things = Embedded Systems



Introduction

Institute for the Wireless Internet of Things at Northeastern

- **Why do we need data processing?**
  - For example, consider an application in which we have to "switch on and off the LEDs when there is a sound impact"
    - **Without data processing:**
      - Sound impact sensor directly connected to the LEDs
      - Result: …
    - **With data processing:**
      - Sound impact sensor connected to microcontroller, microcontroller connected to the LEDs
      - Result: …

Computing Architectures

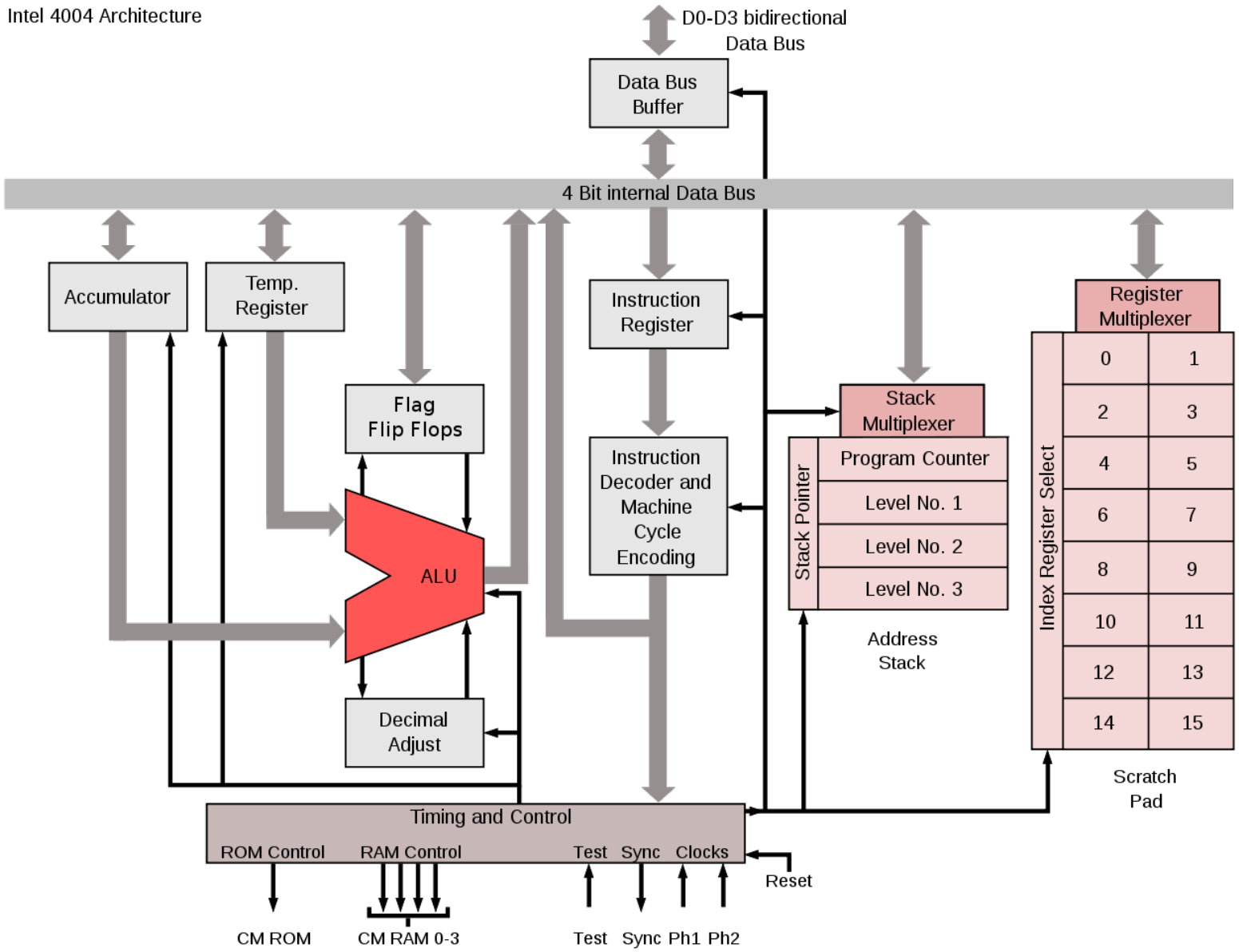
Institute for the Wireless Internet of Things at Northeastern

- **Definition:** A set of rules and methods that describe the functionality, organization and implementation of computer systems

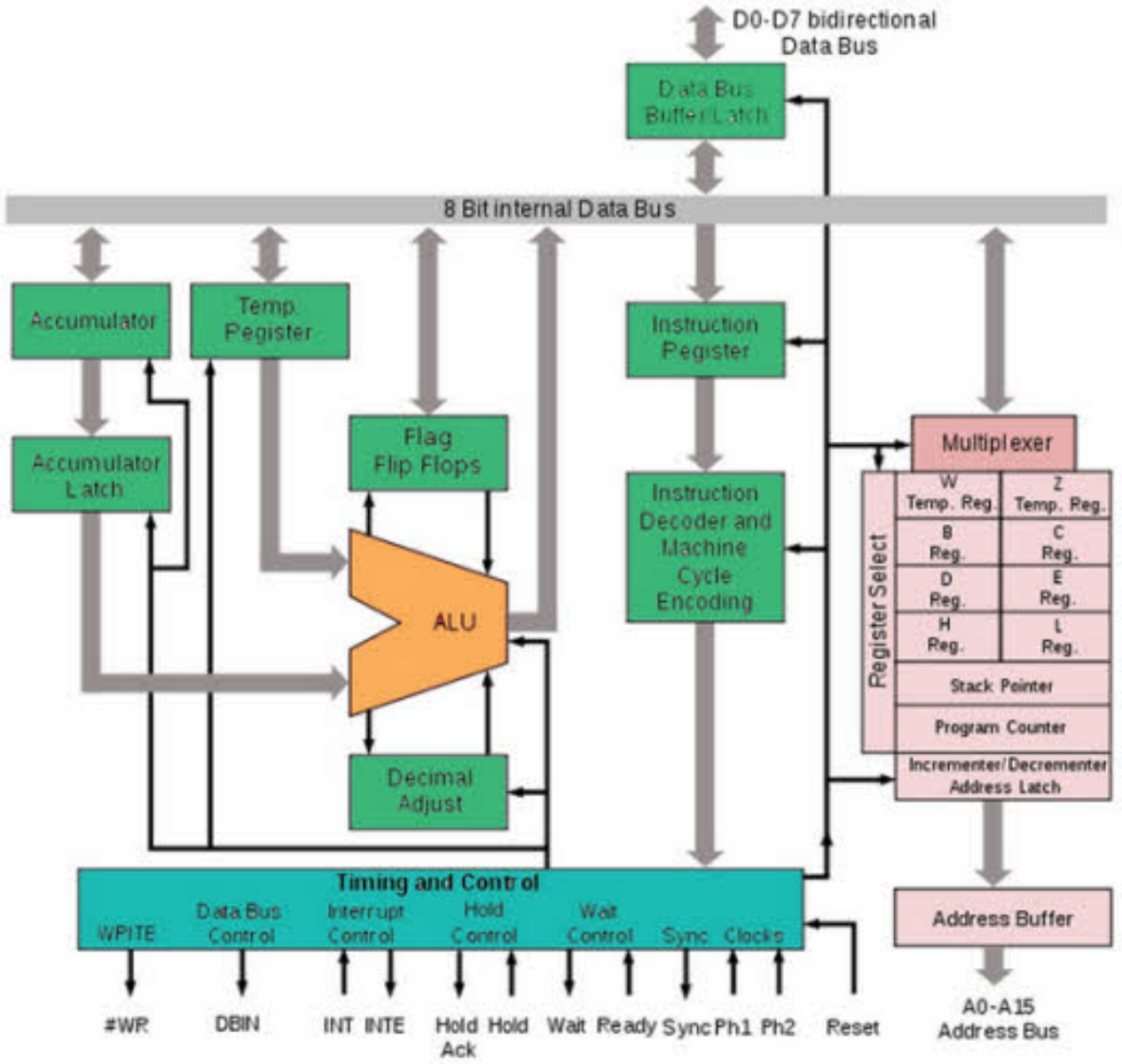
- **Components:**
  1. **Instruction Set Architecture (ISA):**
     - Defines the machine code that a processor reads and acts upon, as well as the word size, memory address modes, processor registers, data type, etc.
  2. **Microarchitecture:**
     - Describes how a particular processor will implement the ISA.
  3. **System Design:**
     - Includes all of the other hardware components within a computing system, e.g., data processing other than the CPU, virtualization, multiprocessing, and software features.

Intel 4004 (4-bit processor, 1971)

Institute for the Wireless Internet of Things at Northeastern

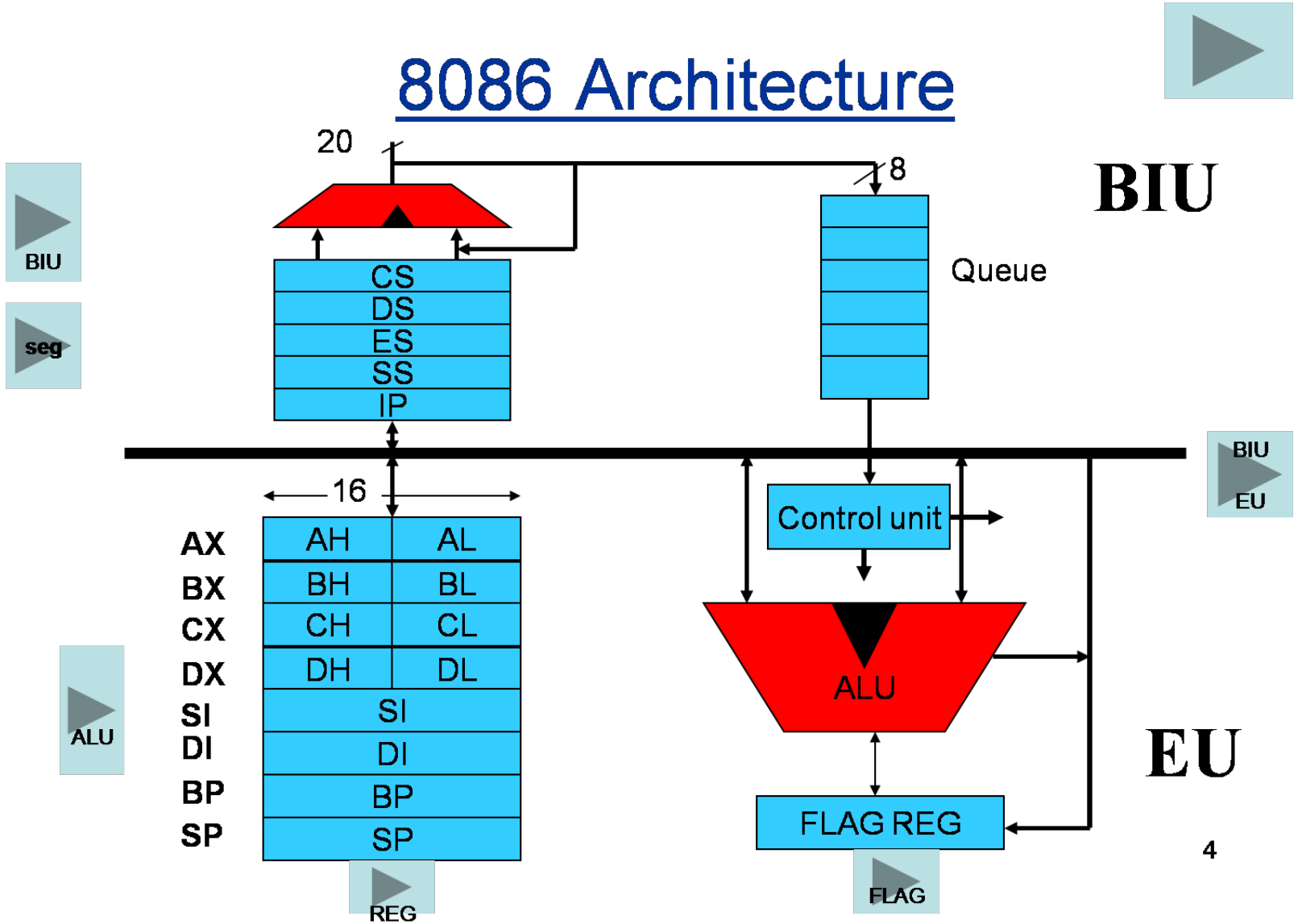


Intel 8008 (8-bit processor, 1972)

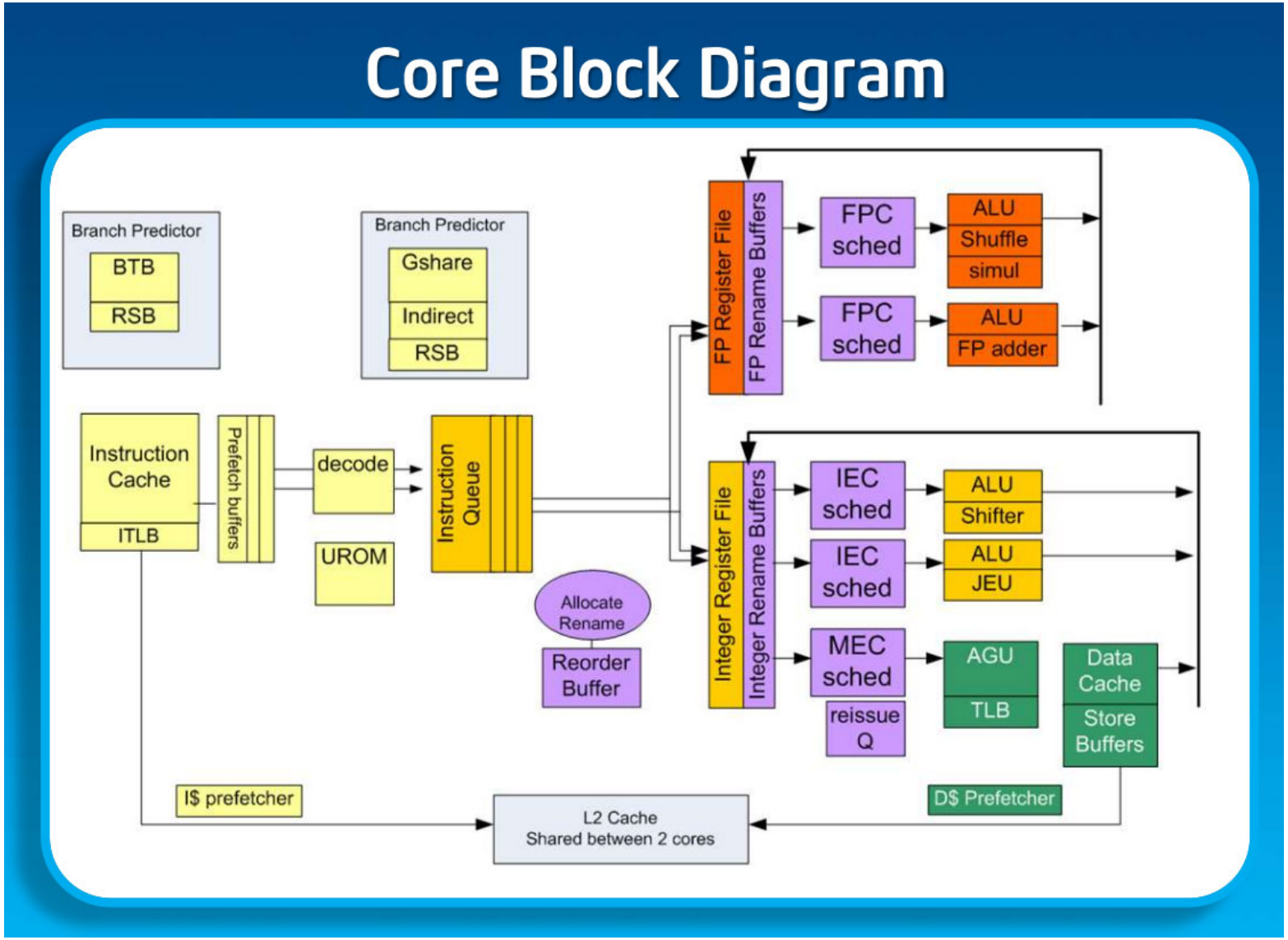


Intel 8086 (1978, 16-bit processor)

Institute for the Wireless Internet of Things at Northeastern



Intel Core (from 2006, 32/64-bit)



Types of Instructions Set Architectures

- **Complex Instruction Set Computer (CISC):**
  - Architecture in which single instructions can execute several low-level operations (such as a load from memory, an arithmetic operation, and a memory store) or are capable of multi-step operations or addressing modes within single instructions.
  - **Examples:** Intel x86 family, Intel 8051 family, Motorola 68000 family, …

- **Reduced Instruction Set Computer (RISC):**
  - Architecture that has a small set of simple and general instructions, rather than a large set of complex and specialized instructions.
    - Usually, where memory is only accessed through specific instructions, rather than as a part of most instructions.
  - **Examples:** ARM, Atmel AVR, MIPS, …

Examples

- **Your computer**
  - x86
- **PlayStation 4/5, Microsoft XBox One**
  - x86
- **Nintendo Switch**
  - ARM Cortex
- **iPhone, iPad**
  - ARM core designed by Apple (but still ARM)
- **Android devices**
  - ARM core (Qualcomm's Snapdragon, Mediatek Mali, Samsung Exynos)
- **Arduino**
  - Atmel
- **Raspberry Pi**
  - ARM

Specifications

- The exact form of a computer system depends on the constraints and goals

- Computer architectures usually exhibit some **trade offs** between
  - Performance in terms of speed, latency and throughput
  - Power requirements and energy consumption
  - Memory capacity
  - Size and weight
  - Cost

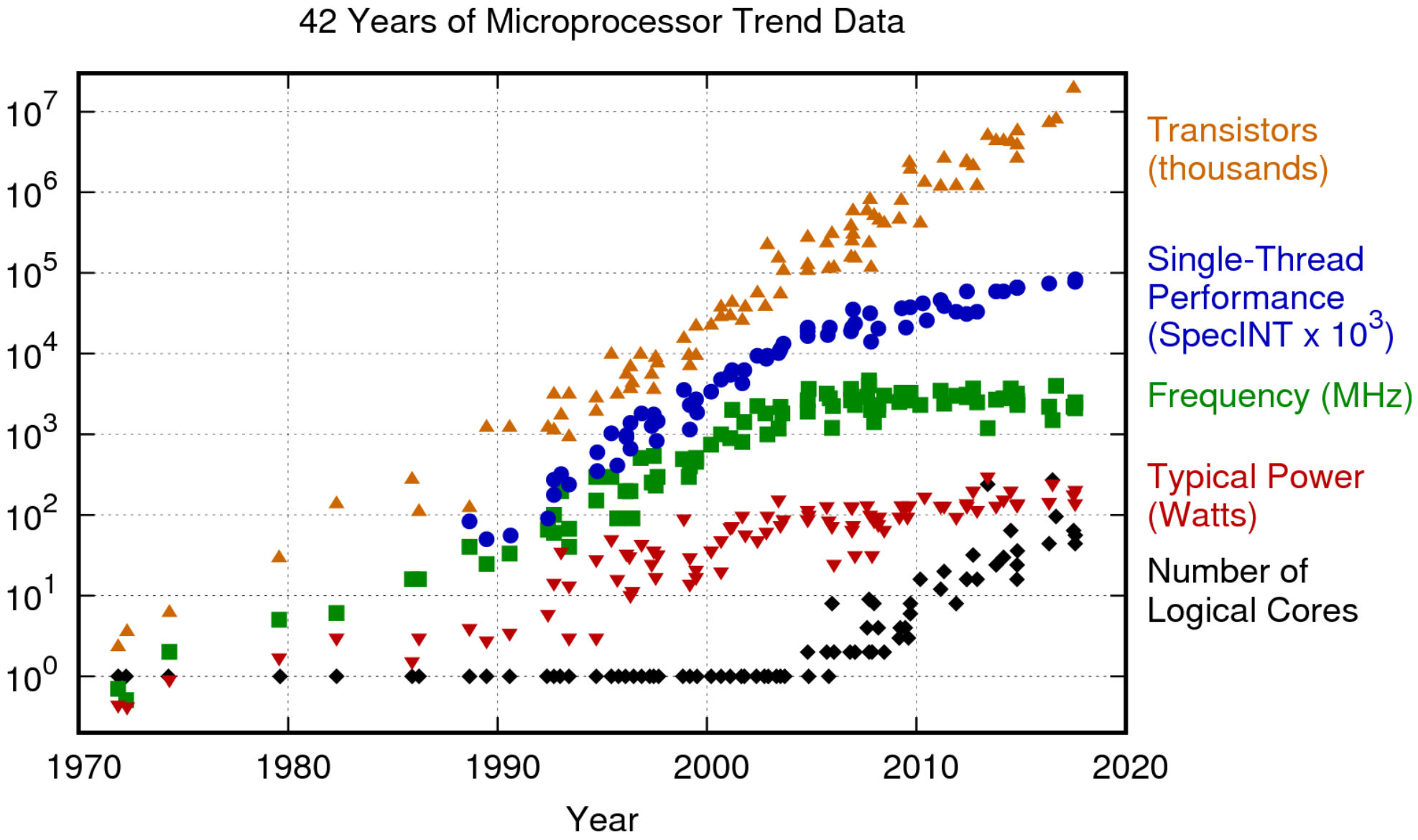
- There is no "absolute best" computing architecture, it all depends on the requirement of your specific application

Specifications

Institute for the Wireless Internet of Things at Northeastern

Feature	Personal mobile device	Desktop	Server	Cluster computer	Embedded
Price of system	$100–$1500	$300–$2500	$5000–$10,000,000	$100,000–$200,000,000	$10–$100,000
Price of micro-processor	$10–$100	$50–$500	$200–$2000	$50–$250	$0.01–$100
Critical system design issues	Cost, energy, media performance, responsiveness	Price- performance, energy, graphics performance	Throughput, availability, scalability, energy	Price-performance, throughput, energy proportionality	Price, energy, application-specific performance

Trends in Computing Architecture



Processing Speed

Institute for the Wireless Internet of Things at Northeastern

- Depends on:
  - The clock frequency (MHz, GHz, **THz**)
    - In part set by the underlying transistor technology (32 nm, 20 nm, 14 nm, **1 nm**)
  - Data unit size (8-bit, 16-bit, 32-bit, 64-bit)
  - Type of instructions, execution order

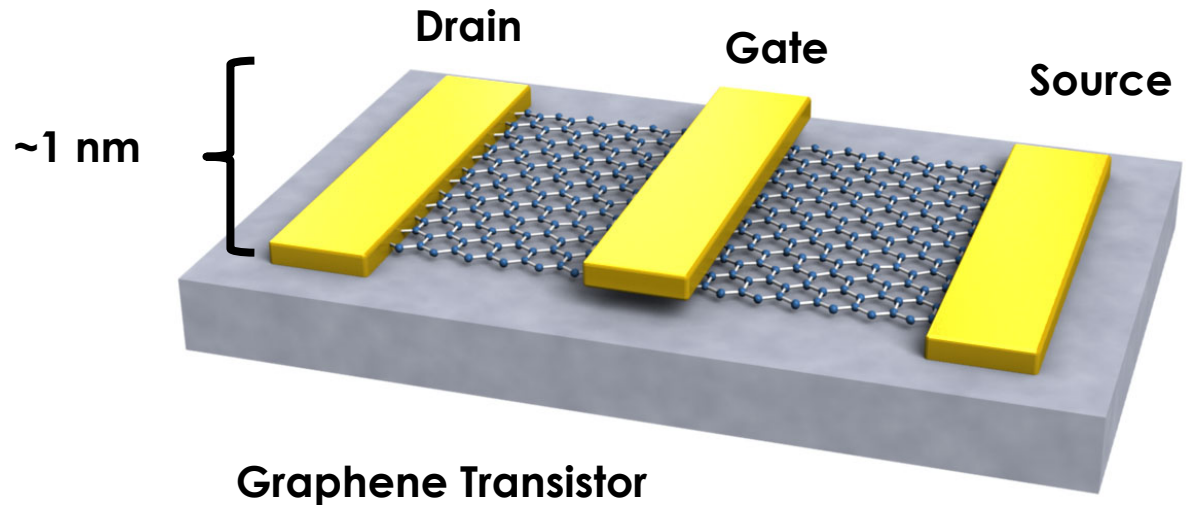
Energy Consumption, Power Requirements

Institute for the Wireless Internet of Things at Northeastern

- Depend on the:
  - Cost per instruction (Joules/instruction), set by
    - Word size
    - Underlying technology
  - Number of instructions to complete operation
    - Joules/instruction * Number of instructions = Joules → **Energy consumption**
  - Instructions per second, set by
    - Clock frequency
    - Instruction set and their implementation
    - Joules/instructions * Instructions/second = Joules/second = Watts → **Power requirement**

True-Nano Processors

- 14 nm, 7 nm transistor technology is commercially available already (e.g., Intel, Apple, Samsung, Qualcomm, …)!
  - This is silicon-based CMOS technology
  - Operation frequency: a few GHz
- **World's smallest transistor** (2008) is **just 1 atom x 10 atoms** (1 nm transistor)
  - This is based on graphene
  - Operating frequency close to 1 THz

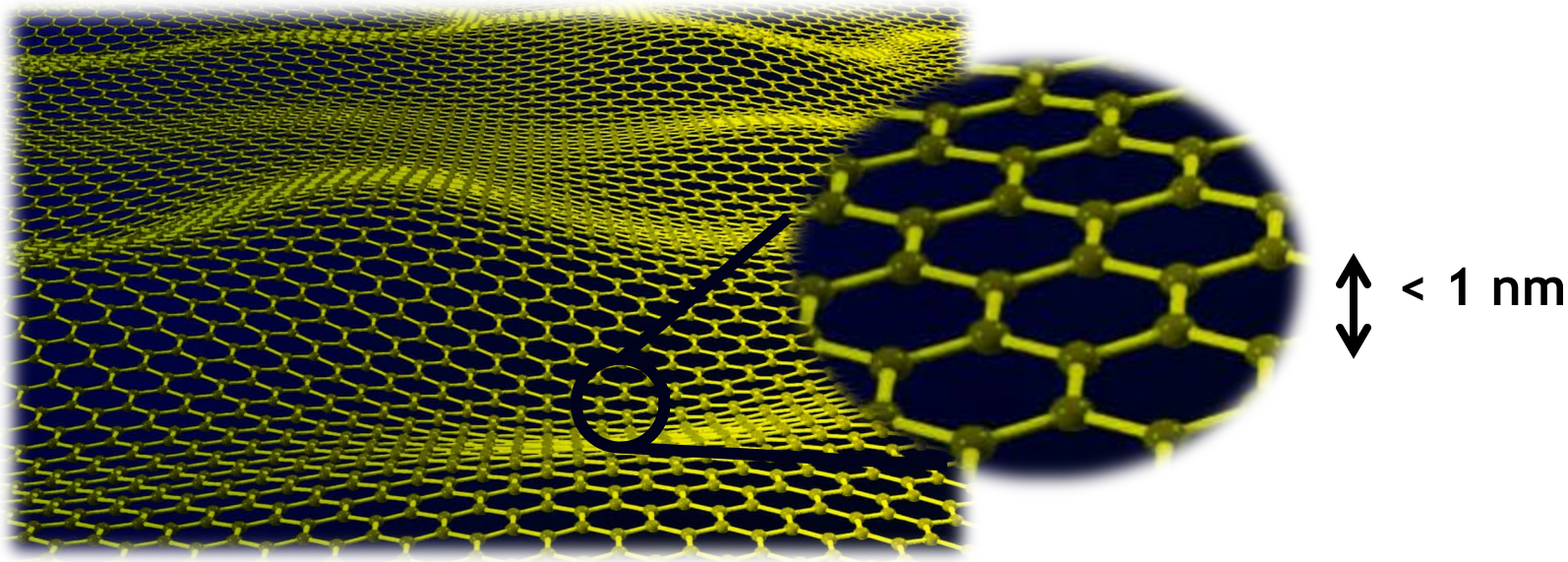


Graphene Transistor

Graphene

Institute for the Wireless Internet of Things at Northeastern

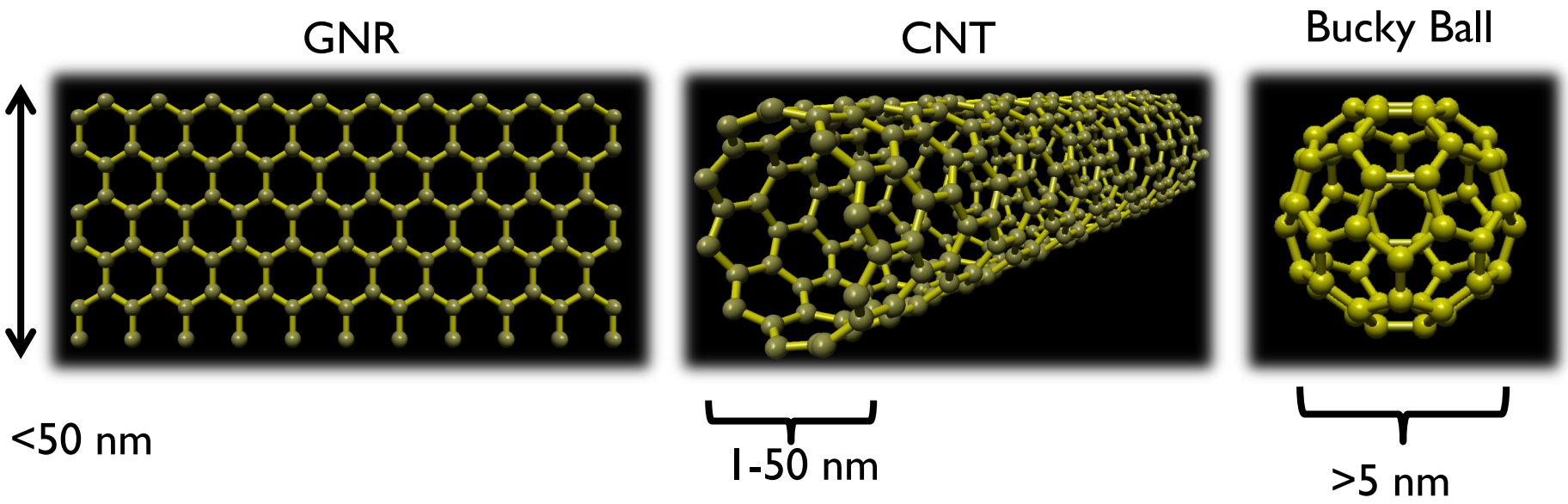
- **A one-atom-thick planar sheet of bonded carbon atoms in a honeycomb crystal lattice:**
  - Theoretically investigated since 1859
  - Experimentally obtained for the first time in 2004 by Andre Geim and Konstantin Novoselov
    - Nobel Prize in Physics (2010)



Graphene Derivatives

Institute for the Wireless Internet of Things at Northeastern

- **Graphene Nanoribbons (GNR):** A thin strip of graphene
- **Carbon Nanotubes (CNT):** Rolled graphene
- **Bucky Balls:** A graphene sphere



Graphene Properties

Institute for the Wireless Internet of Things at Northeastern

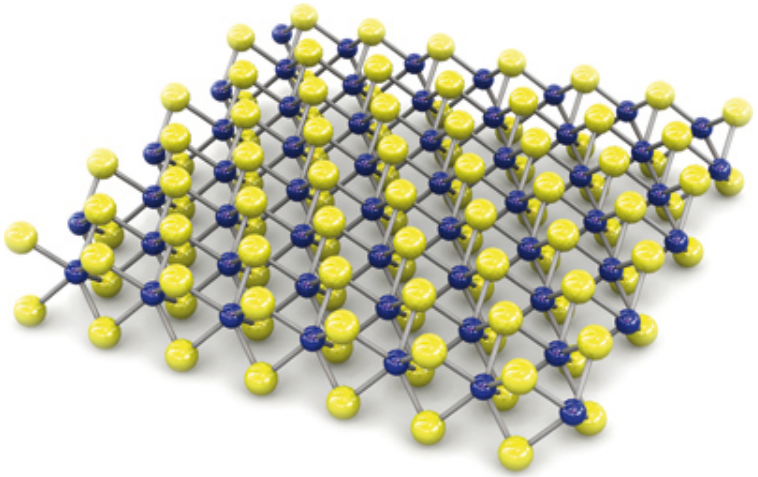
- **The wonder material of the 21st century?**
  - **Mechanical properties:** world's thinnest and lightest material, harder than diamond, highly bendable
  - **Electronic properties:** very high electron mobility
  - **Optical properties:** almost transparent material, non-linear optical behavior

- **New opportunities for device technology**
  - Nano-processors, nano-memories, nano-batteries, nanosensors, nano-actuators, …

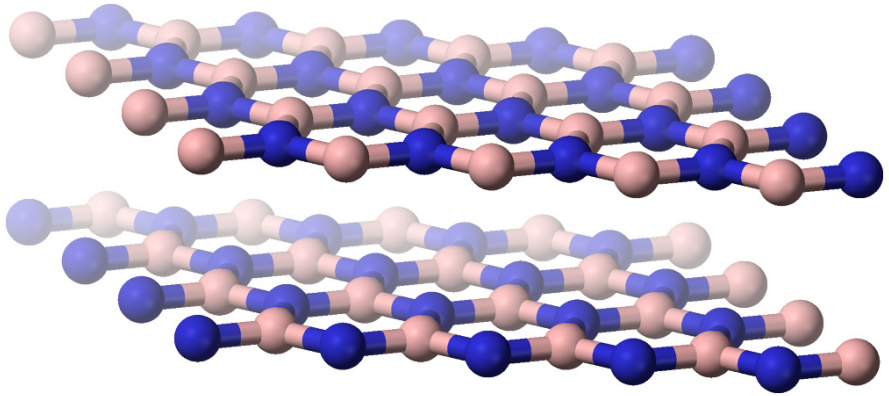
Other 2D Nanomaterials?

Institute for the Wireless Internet of Things at Northeastern

- Graphene is "the first of a kind" → But not the only one!



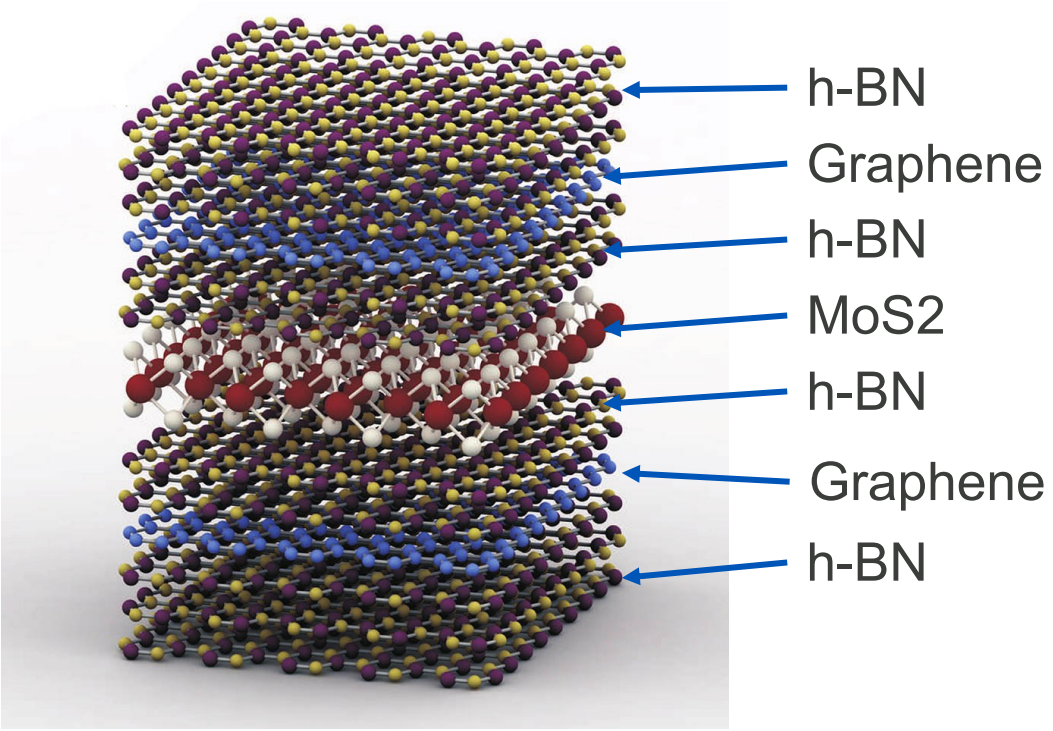
Molybdenum disulfide (MoS2):
Molybdenum + Sulfur
1-atom thick dielectric



Hexagonal Boron Nitride (h-BN):
Boron + Nitrogen
1-atom thick dielectric,
Combines very well with graphene

2D Nanomaterial Stacks

- New structures can be created by stacking different 2D nanomaterials
  - Unprecedented electrical & optical properties
    - **This is just the beginning!**



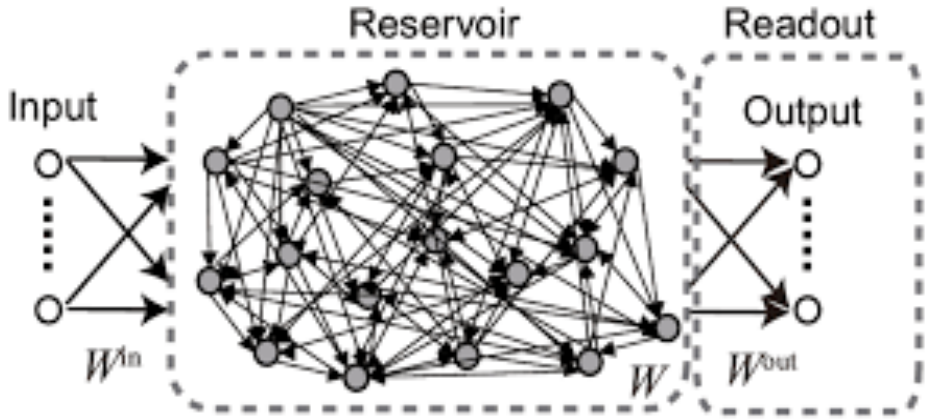
Non-conventional Computing

Institute for the Wireless Internet of Things at Northeastern

- **Brain-inspired or neuromorphic computing**
- **Quantum Computing**
- **Molecular / DNA computing**

Brain Inspired or Neuromorphic Computing

- A form of analog computing:
  - Instead of dealing with discrete "0"s and "1"s, the computations are performed with continuous signals
- **Examples:**
  - Neural networks
  - Reservoir computing
- **Implementation:**
  - In *software*, on a traditional digital computer
  - With dedicated *hardware*, utilizing devices with a non-linear response
    - In the case of brain-inspired computing, the preferred response is usually a sigmoid
      - Examples include memristors (a memory resistor) and crossbar junctions

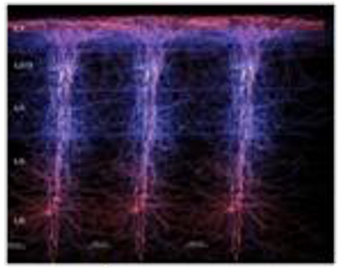


Brain Inspired or Neuromorphic Computing

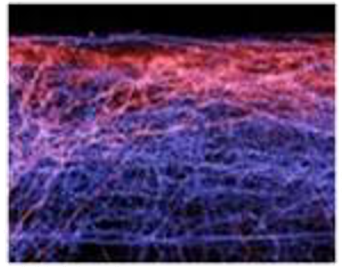
Institute for the Wireless Internet of Things at Northeastern

$\sim 10^{10}$ synapses/cm^2

$\sim 10^{6}$ neurons/cm^2



$\sim 10^{4}$ neurons/ cortical column

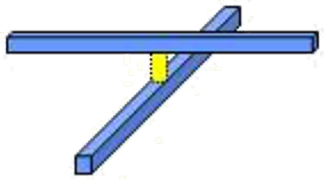


$\sim 5X10^{8}$ long range axons @ 1 Hz

Biological Brain

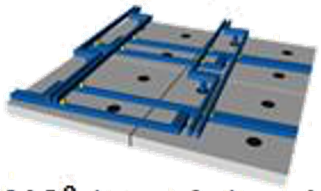
Synapse | Neurons | Microcircuit | Long-range interconnects | Brain

CROSSBAR JUNCTION



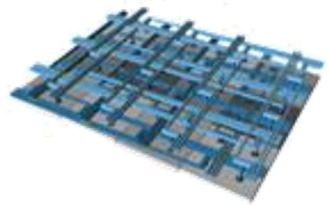
$\sim 10^{10}$ intersections/ cm^2 @ 100 nm pitch

CMOS SUBSTRATE



$5X10^{8}$ transistors/ cm^2 @ 500 transistors/ neuron

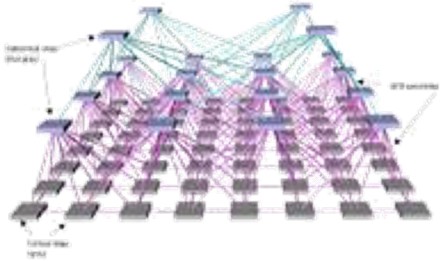
LAMINAR CIRCUIT



Layered cortical circuits with $\sim 10^{6}$ neurons/cm^2

HIGH SPEED BUS

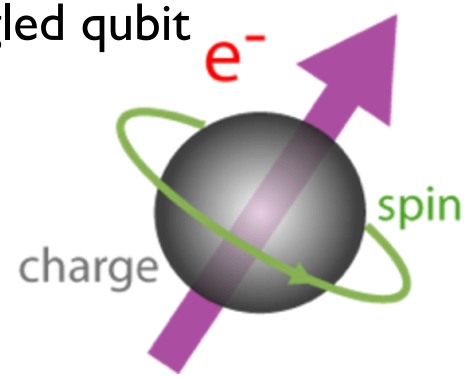
Multi-Gbit/sec digital comms



Electronic Brain

Quantum Computing

- Perform operations by leveraging the properties of quantum bits or *qubits*
- Qubits can be superposed
  - A qubit can be a "0", a "1" or in a superposition of both
  - Qubits can be entangled, or, in simpler terms, their properties can be fully "correlated"
    - If we know the properties of one qubit, we can deduce the properties its entangled qubit
- Examples of qubits:
  - Polarization of a photon, with vertical and horizontal polarization
  - Spin of an electron, with spin up and spin down



- The key idea in quantum computing is to leverage both quantum superposition and quantum entanglement
- Key application: quantum cryptography

Quantum Computer

- **State of the art:** 54-qubit computer
  - Electronic qubits on vacuum-sealed, cooled superconducting chambers

- Great for some operations
  - Currently, not that useful for practical problems

Molecular / DNA Computing

- Living cells can be thought of as embedded systems:
  - They interact collect information from their surroundings
  - They process the information from the surroundings
  - They can communicate with other cells

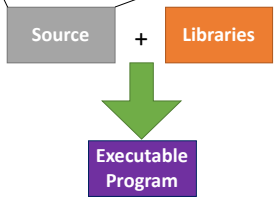
- The processing of information in cells is very similar from any other processing system:
  - No bits, no qubits
  - All-based on DNA, proteins and enzymes

Wirelessly (Re)Programming Genome (in Living Neurons)

Institute for the Wireless Internet of Things at Northeastern

J. M. Jornet, Y. Bae, C. Handelmann, B. Decker, A. Balcerak, A. Sangwana, P. Miao, A. Desai, L. Feng, E. K. Stachowiak and M. K. Stachowiak,
"Optogenomic Interfaces: Bridging Biological Networks with the Electronic Digital World," Proceedings of the IEEE, July 2019.

```
import socket
import sys

# Create a TCP/IP socket
sock = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)

# Connect the socket to the server
server_address = (host, port)

try:
    # Send data
    message = 'Hello World!'
    sock.sendall(message)

    # Look for the response
    amount_received = 0
    amount_expected = len(message)

...
```



General Purpose Computer:
A device that can be instructed to carry out different sequences of commands, i.e., programs, by means of computer programming.

Programming:
To design the algorithms to achieve a given functionality-applications and to code them in a given programming language. These are written in a human-friendly high-level language.

Operating System:
The program that manages the computer resources, especially the allocation of those resources among other programs.

Running Programs:
Programs being executed access different computer resources and generate different outputs according to specific inputs.

Compiling, Linking and Assembly:
Integrating the high-level instructions with libraries and other dependency files, and then convert the result to low-level instructions or machine-language that the processor can understand.

Execution:
The program runs according to the given inputs, utilizing the resources given by the operating system, and generates different outputs.

General Purpose Cell Computer:
A device that can be instructed to carry out different sequences of commands, i.e., programs or applications, by means of Genome programming.

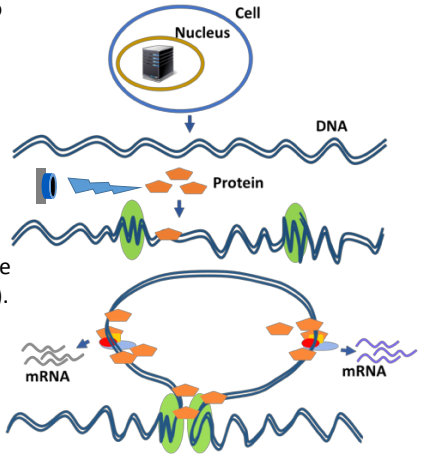
Programming:
DNA in the nucleus contains all the genes (code) needed for a cell to operate. Specific applications (i.e.. Cell Development) are generated by organizing individual genes into functional/coordinated program. These are written in a 3D high-level DNA interactome language by nuclear proteins.

Operating System:
The proteins present in the nucleus determine which applications are used and running in the background and how the cell will respond to a stimulus.

Running Programs:
Developmental gene programs being executed access different gene sets and generate different outputs (cell phenotype) according to specific inputs (environmental signals).

Compiling, Linking and Assembly:
Integrating the high-level environmental instructions with the DNA library and its interactome, and then convert the result to low-level instructions for the individual genes using the language of transcriptional processor.

Execution:
The genomic program runs according to the given environmental inputs (including optogenetic signals), utilizing the resources given by the operating system (DNA interactome), and generates biological outputs, i.e., generation of neurons → synaptic connections → desired communicating neuronal network.



But let's not lose focus…

Institute for the Wireless Internet of Things at Northeastern

- Things = Embedded Systems
  - Target metrics:
    - Compact → Ideally nonintrusive
    - Energy efficient → Ideally perpetual operation (energy harvesting?)
    - Speed → We don't need a supercomputer on board…
    - Other aspects: heat dissipation, reliability, hardware security

Microprocessors vs Microcontrollers

Institute for the Wireless Internet of Things at Northeastern

- **Microprocessor:** An integrated circuit (IC) that contains only a Central Processing Unit (CPU)
  - It does not contain memory (RAM, ROM), input/output (I/O) pins
    - Instead, it is connected to a shared bus, where those items can be connected
  - Advantage: general purpose, you can create your own system based on what you need

- **Microcontroller:** An IC that integrates a microprocessor, ROM and RAM memory, I/O pins, as well as other peripherals like serial ports, counters, clocks, etc.
  - Applications are limited by the peripherals included in the IC
  - Advantage: are more compact and cheaper

CPU or GPU?

Institute for the Wireless Internet of Things at Northeastern

- A **Graphics Processing Unit (GPU)** is (usually) designed to accelerate creation of images for a computer display, including
  - texture mapping, image rotation, translation, shading, …
  - motion compensation, calculation of inverse DCT, for accelerated video decoding

- A CPU consists of a few cores optimized for sequential serial processing…
  - ... while a GPU consists of thousands of smaller, more efficient cores designed for handling multiple tasks simultaneously

- **Conclusion:** GPUs are not really what we need for the majority of IoT applications
  - except if they involve multimedia and high performance…
  - but these come with higher cost and energy consumption

- **Note:** There are other "specialized" ICs that you can consider, e.g., Digital Signal Processors (DSPs), if you need heavy signal processing

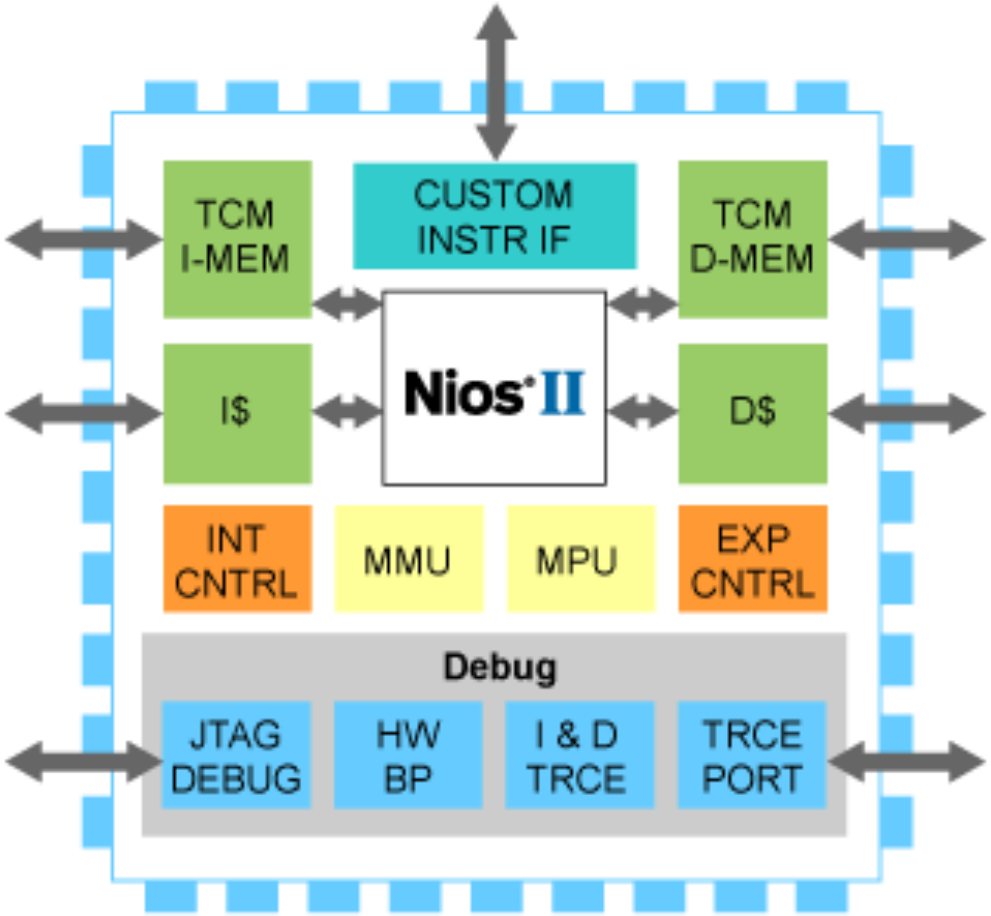
ASIC or FPGA?

- **Application Specific Integrated Circuit (ASIC):** a chip customized to perform a specific task rather than general-purpose applications:
  - E.g., every single IC you use in normal IoT applications, such as TI microcontrollers,

- **Field Programmable Gate Array (FPGA):** a device that can be reconfigured to create different digital circuits:
  - E.g., Xilinx Artix 7 in ZedBoard, Intel Cyclone V in Terasic DE1

ASIC vs FPGAs

- In an FPGA, you can create "any" IC:
  - You can create a CPU (e.g., Nios II Processor)
  - You can create a microcontroller (e.g., Intel 8051)
- Advantages:
  - Fully customizable
  - Potentially faster
- Disadvantages:
  - Cost and energy consumption
- Might be an "overkill" for many of the IoT applications

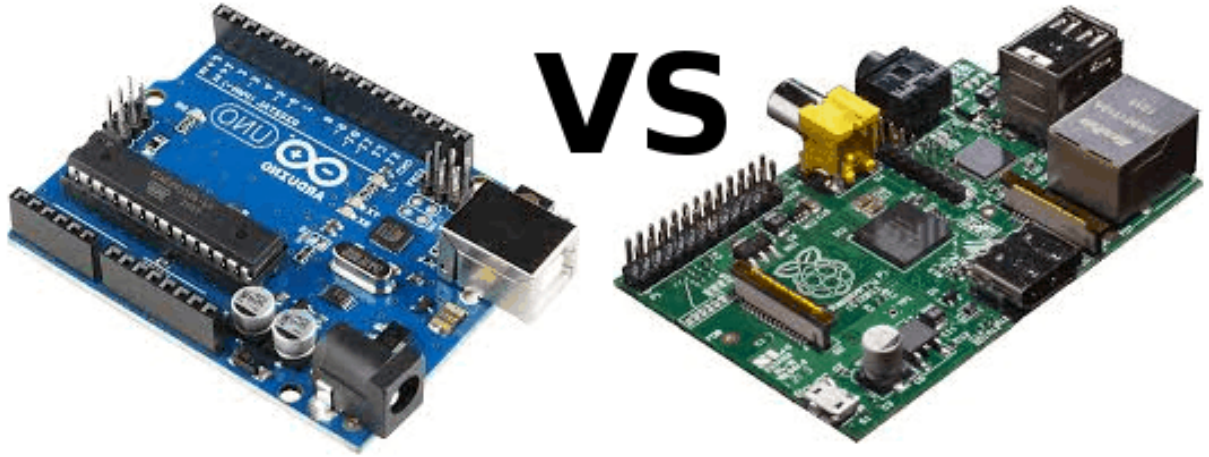


Arduinos, Pies

Institute for the Wireless Internet of Things at Northeastern

- **Arduino:** A microcontroller-based board
  - Easy to use, great educational tool
  - Provides very limited hardware control

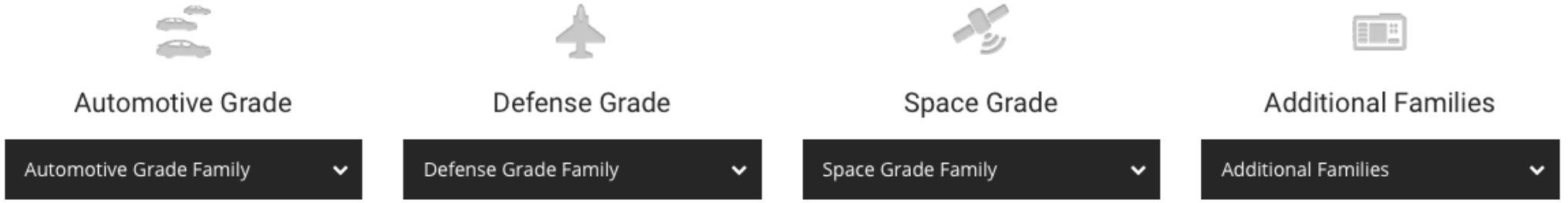
- **Raspberry Pi:** A compact computer on a board (single-board computer)
  - General purpose computer, running Linux..
  - Many other "flavors": Banana Pi, Orange Pi



Application-specific Factors

Institute for the Wireless Internet of Things at Northeastern

- Where will your device operate?
  - Medium
    - Inside the body
    - Inside an engine
    - Indoors and outdoors
    - In space
  - Temperature
    - Commercial grade (-40 to 85 C)
    - Military grade (-55 to 125 C)
  - …



Course Contents

Institute for the Wireless Internet of Things at Northeastern

- **Module T1:** Introduction to the Internet of Things ✓
- **Module T2:** Data Acquisition ✓
- **Module T3:** Local Data Processing ✓
- **Module T4:** Data Communication
- **Module T5:** Data Streaming
- **Module T6:** Data Storage & Cloud
- **Module T7:** Data Analytics