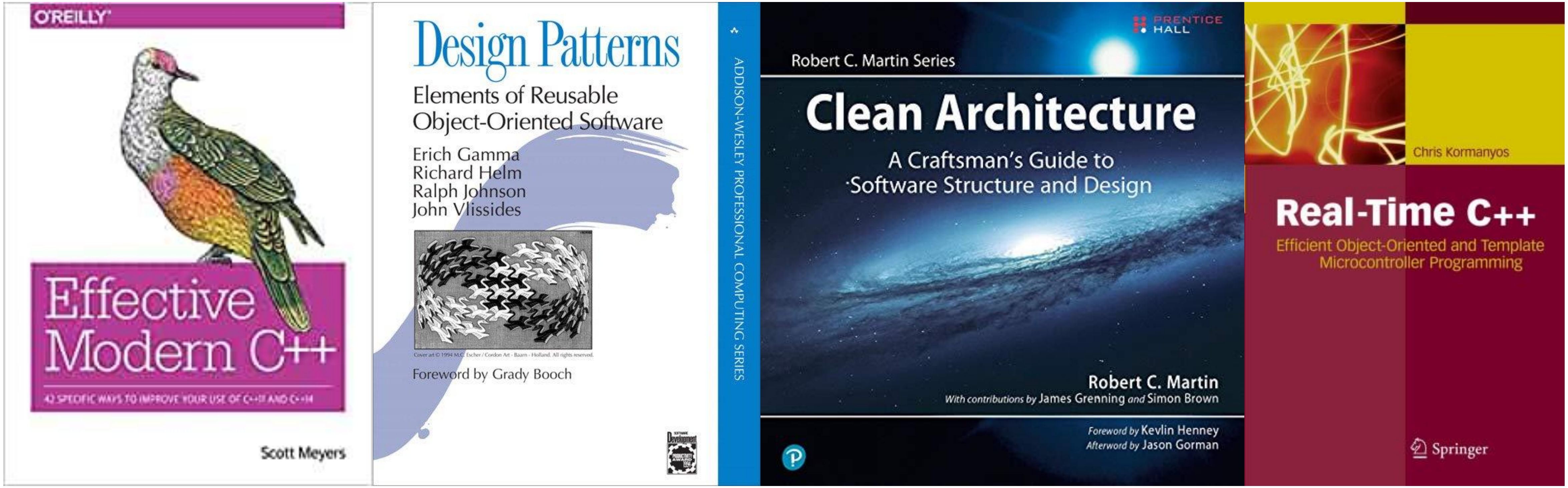
Introduction to Embedded Systems

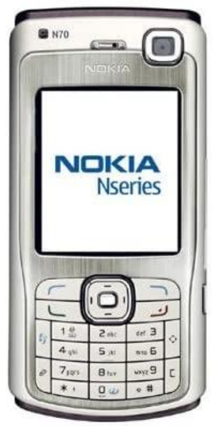
Mohamed Saied

Books


O'REILLY
Effective Modern C++
Scott Meyers
Design Patterns
Elements of Reusable Object-Oriented Software
Erich Gamma
Richard Helm
Ralph Johnson
John Vlissides
Foreword by Grady Booch
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES
PRENTICE HALL
Robert C. Martin Series
Clean Architecture
A Craftsman's Guide to Software Structure and Design
Robert C. Martin
With contributions by James Grenning and Simon Brown
Foreword by Kevlin Henney
Afterword by Jason Gorman
Chris Kormanyos
Real-Time C++
Efficient Object-Oriented and Template Microcontroller Programming
Springer

What's an Embedded System?

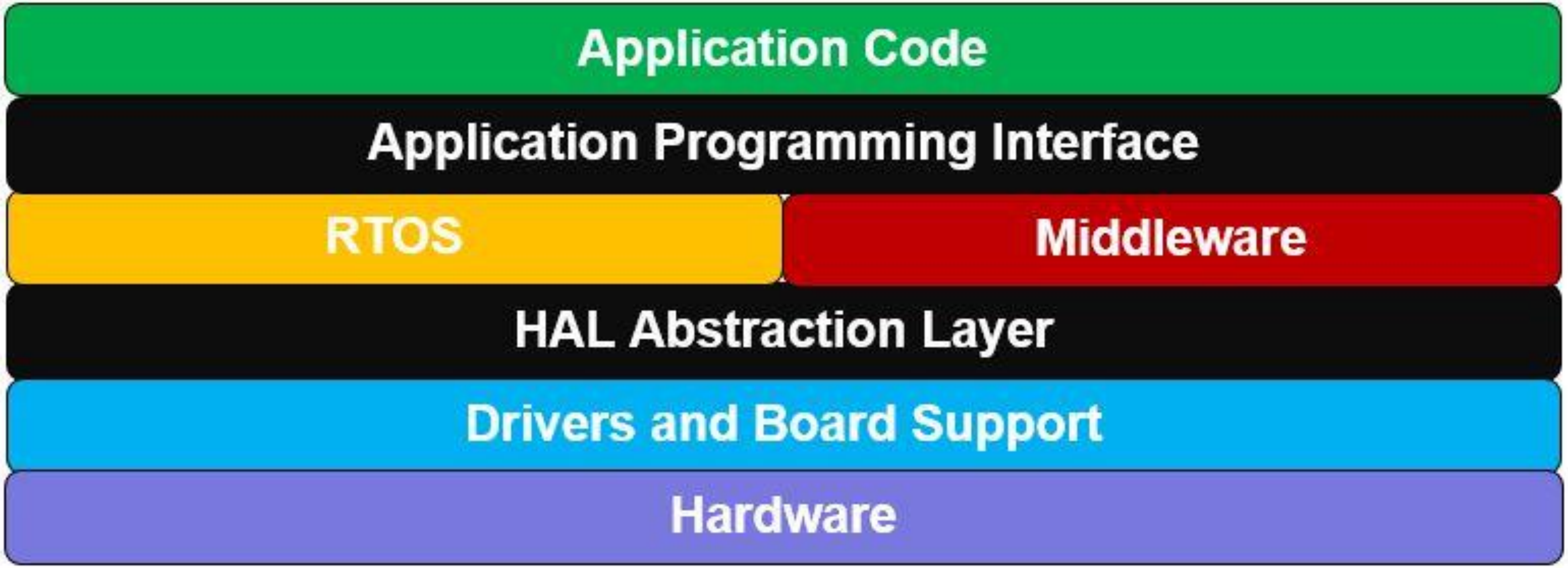
- Embedded systems =
  - information processing systems embedded into a larger product
- Two types of computing
  - Desktop – produced millions/year
  - Embedded – billions/year
- Non-Embedded Systems
  - PCs, servers, and notebooks
- The future of computing!
  - Automobiles, entertainment, communication, aviation, handheld devices, military and medical equipments.



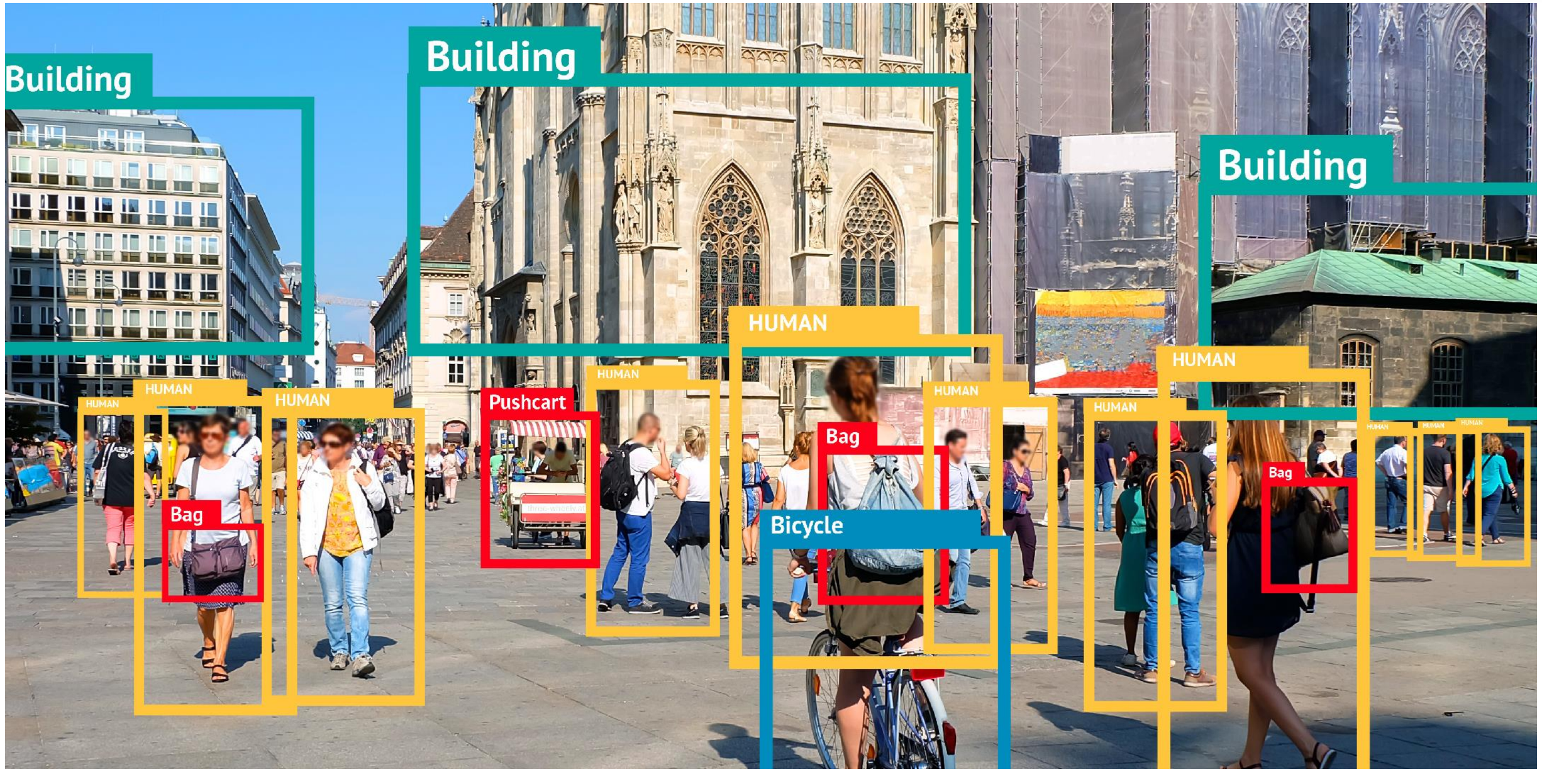


Languages of Embedded Systems

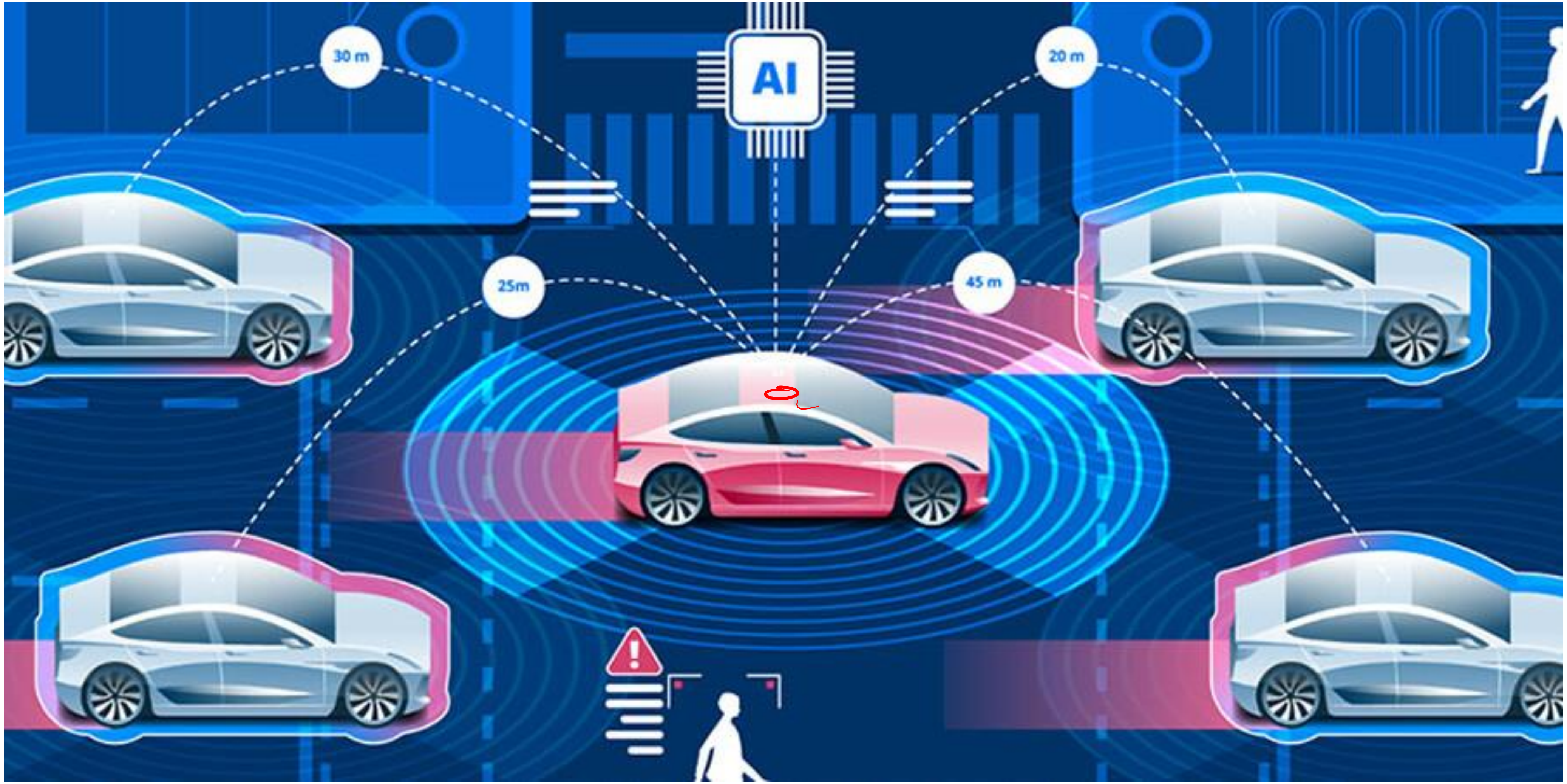
Where to find C++



Where to find C++



Connected Cars

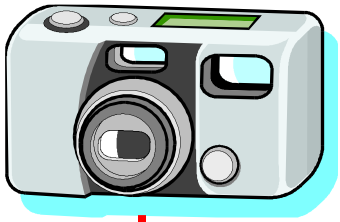


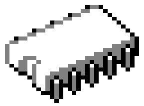
Embedded Systems

- Devices other than desktop PCs, servers, and notebooks
  - Electricity running through
  - Perform something intelligent

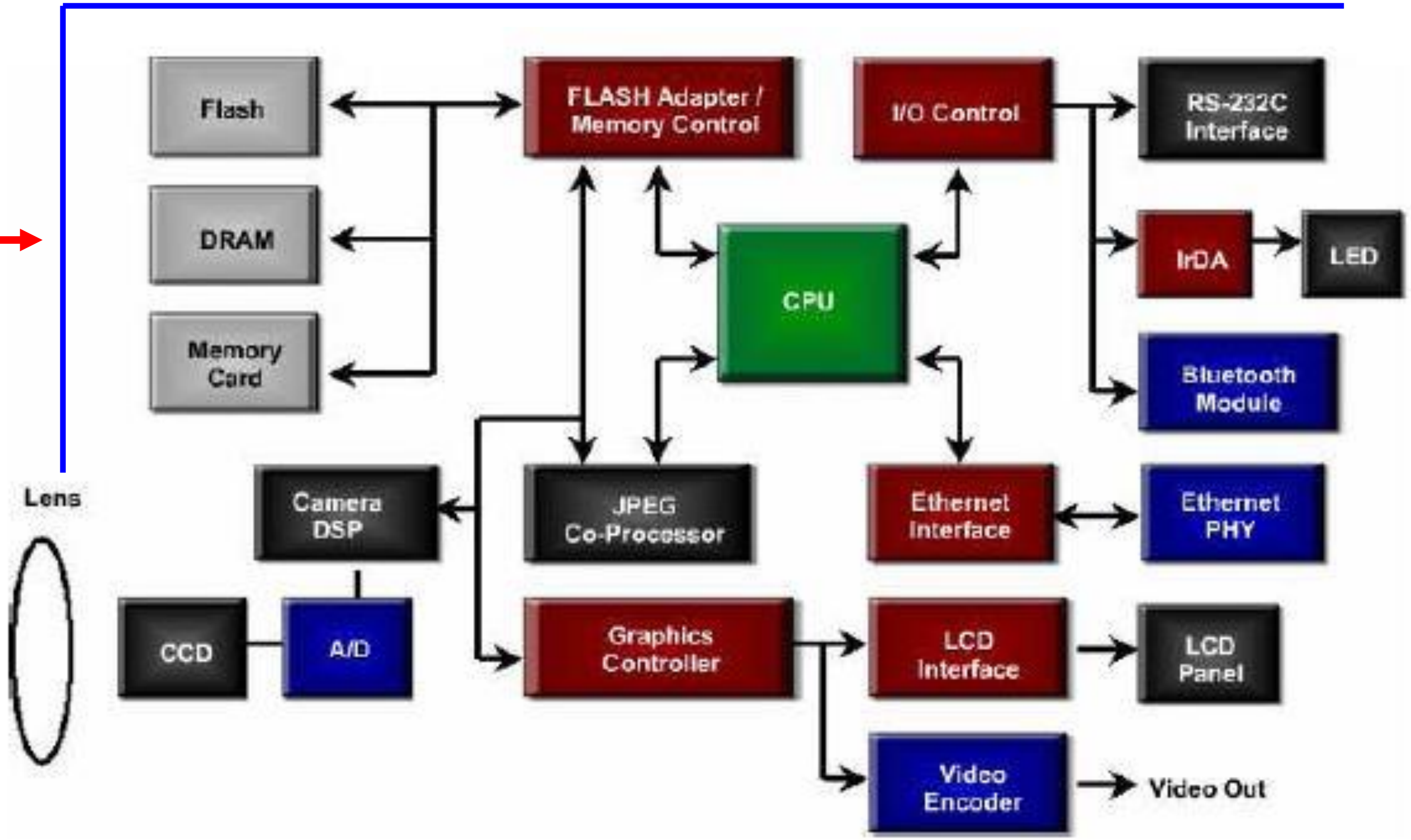
- Hardware/software which form a component of a larger system, but are concealed from user
- Computers camouflaged as non-computers
- The future of computing!

An Example Embedded System

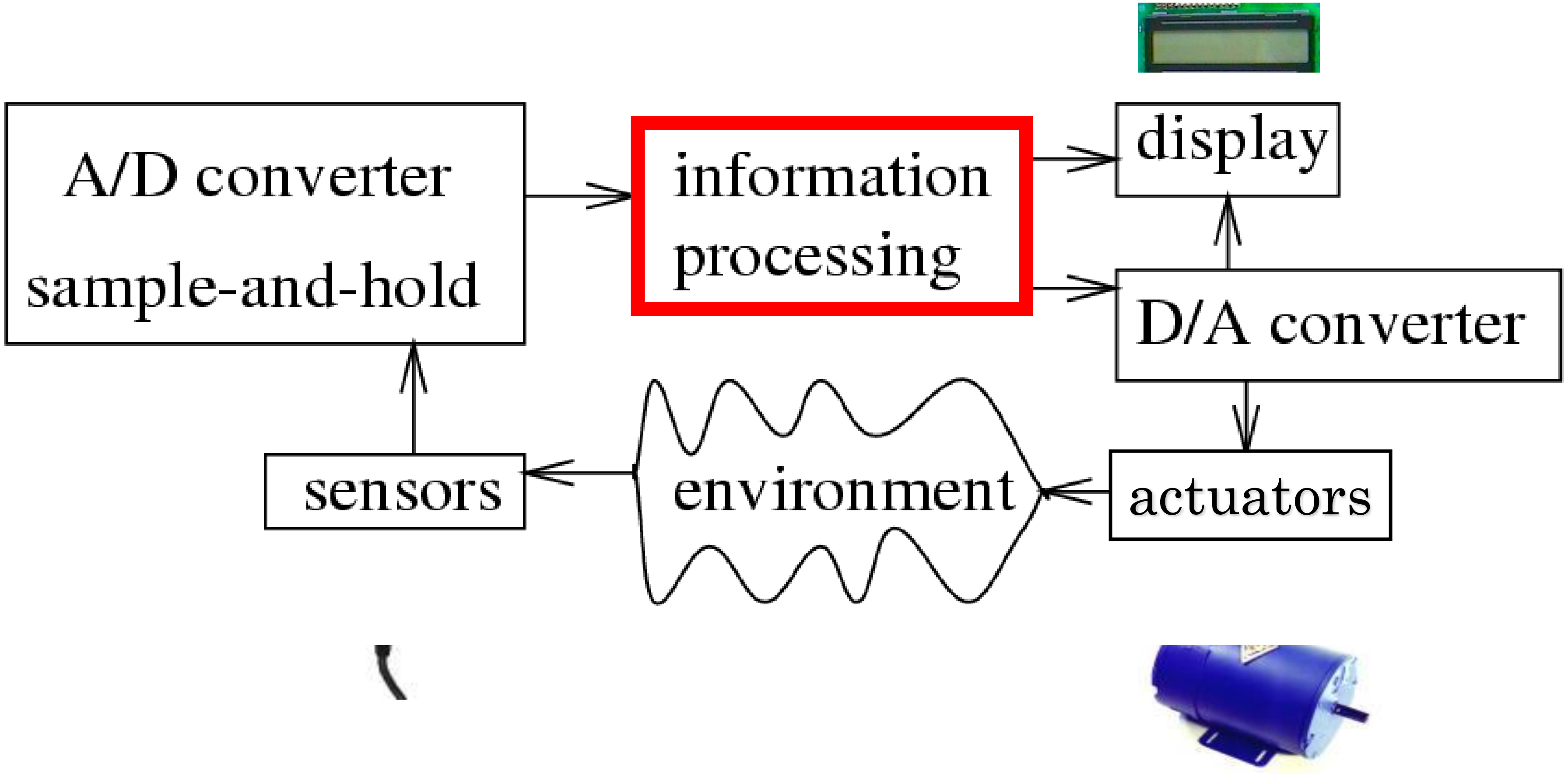




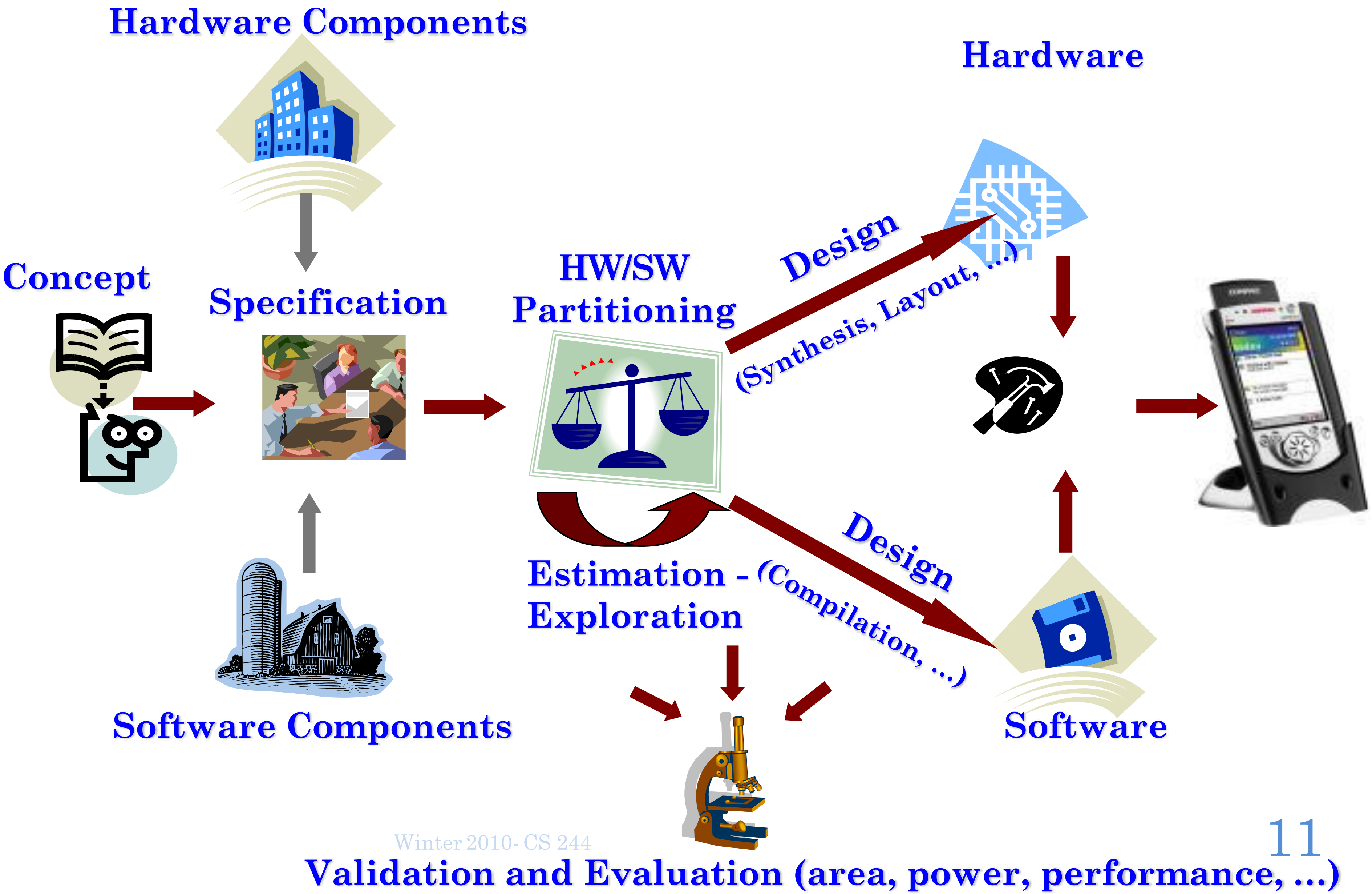
Digital Camera Block Diagram



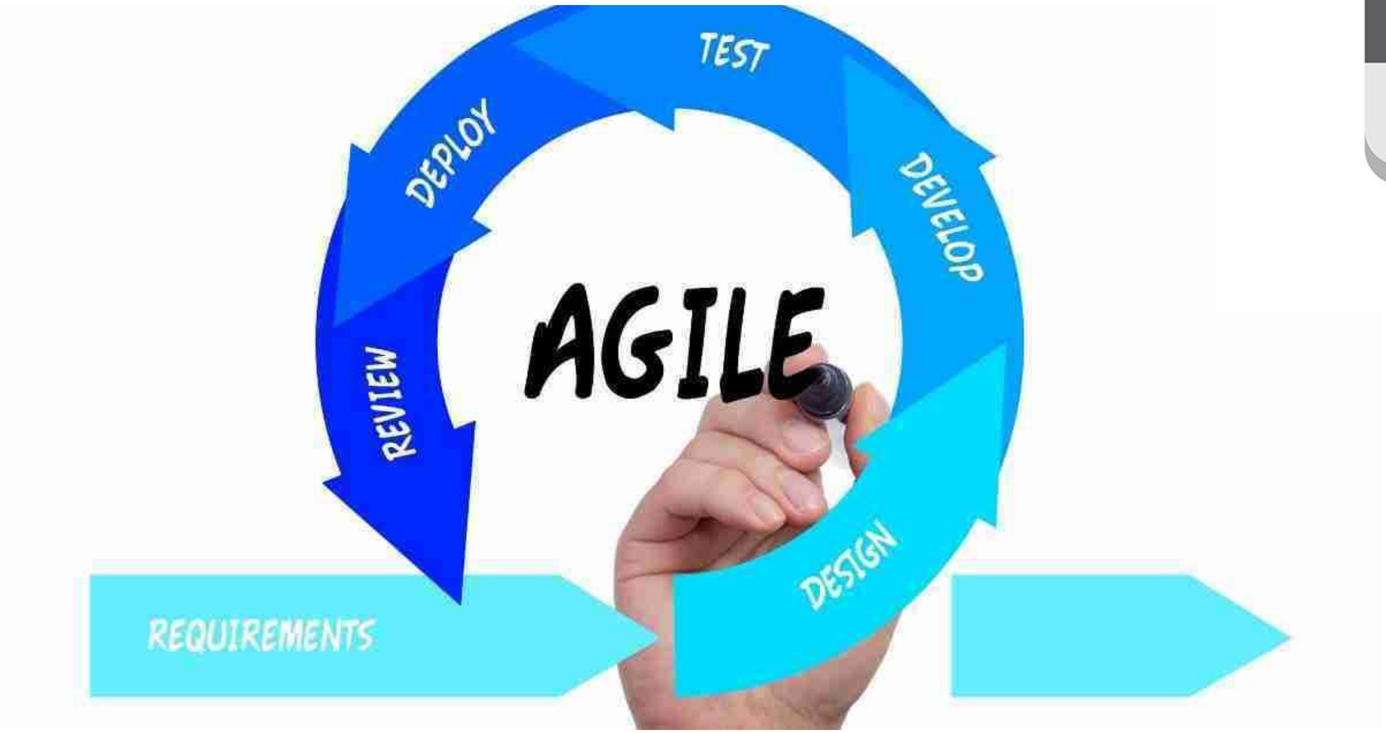
ES: Simplified Block Diagram



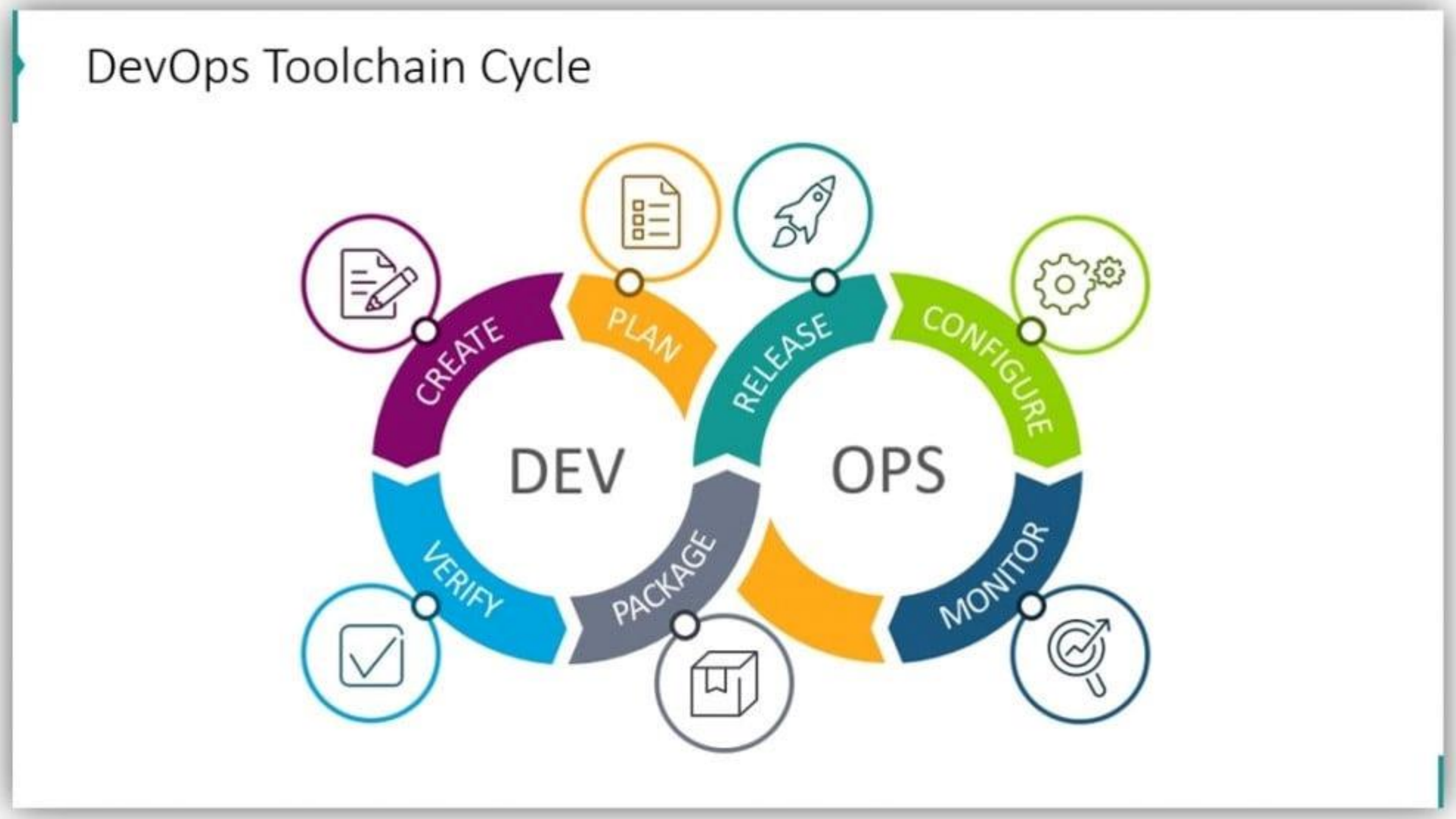
Course Outline



Modern Software Engineering Practices



Continuous Integration CI



Components of Embedded Systems

Analog Components

- Sensors, Actuators, Controllers, …

Digital Components

- Processor, Coprocessors
- Memories
- Controllers, Buses
- Application Specific Integrated Circuits (ASIC)

Converters – A2D, D2A, …

Software

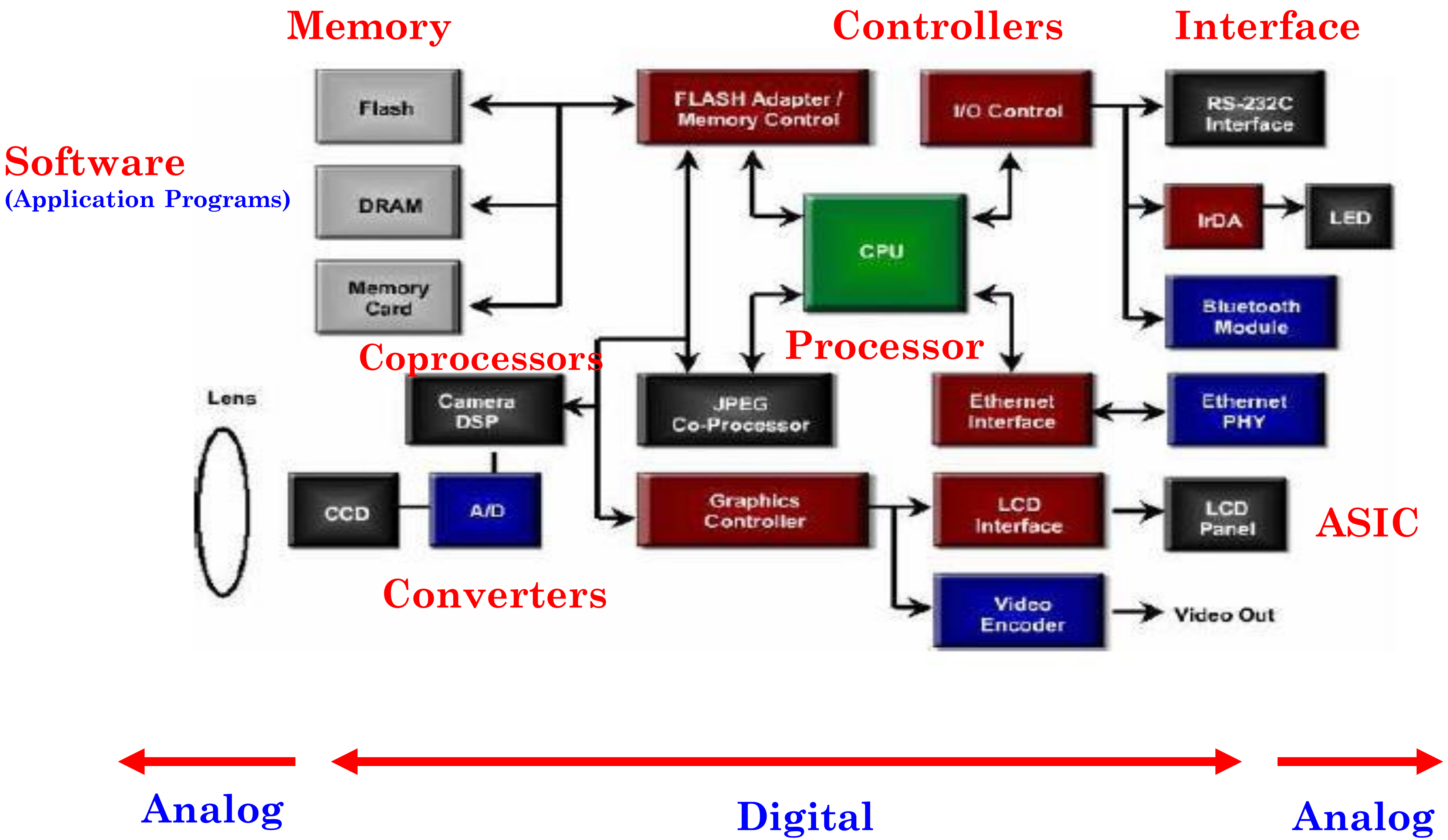
- Application Programs
- Exception Handlers

Hardware

Software

Hardware Components

Hardware Components of Embedded Systems- an example



Processors

What is a processor?

- Artifact that computes (runs algorithms)
- Controller and data-path

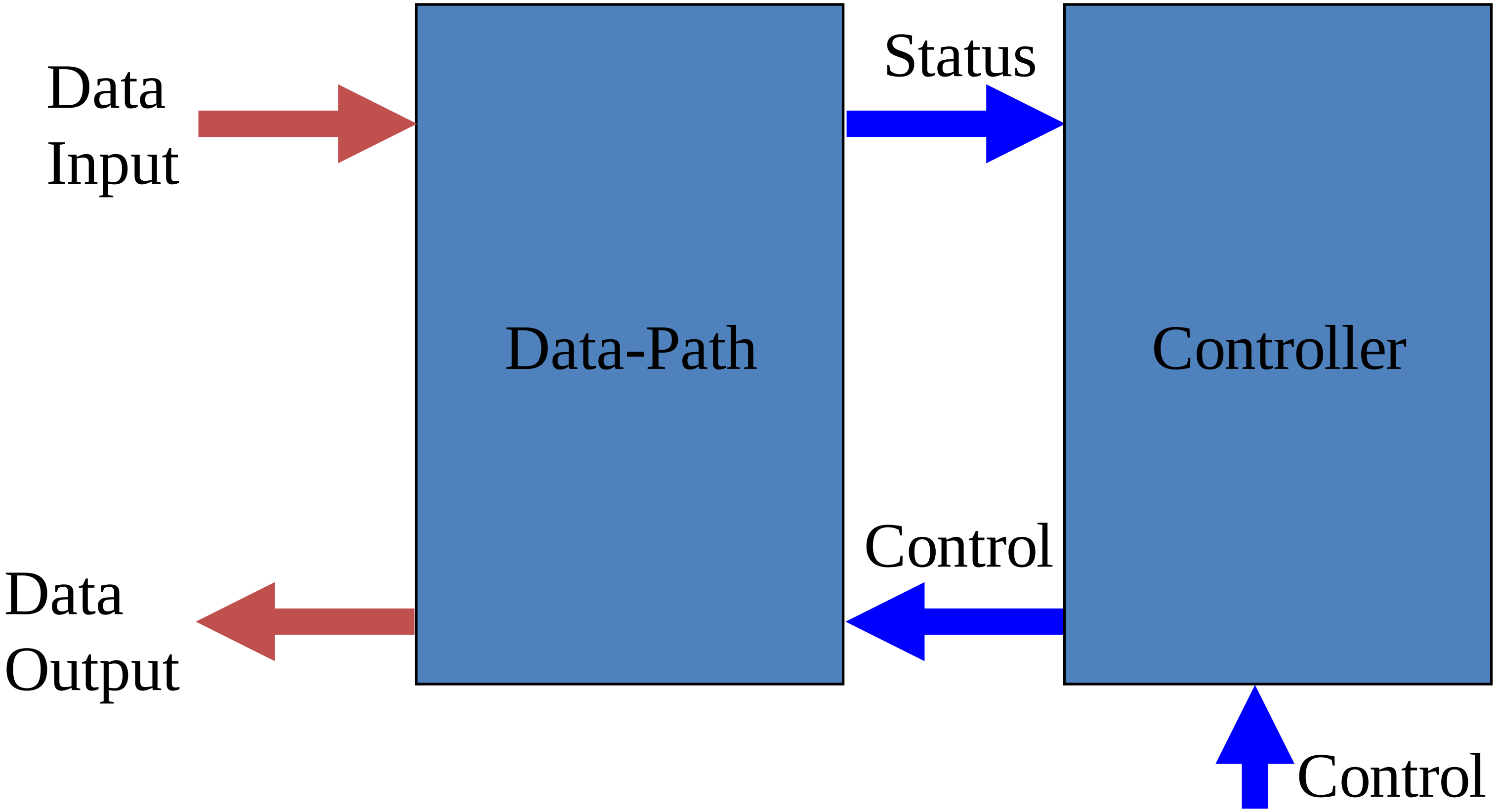
General-purpose (GP) processors:

- Variety of computation tasks
- Functional flexibility and low cost at high volumes (maybe)
- Slow and power hungry

Single-purpose (SP) processors (or ASIC)

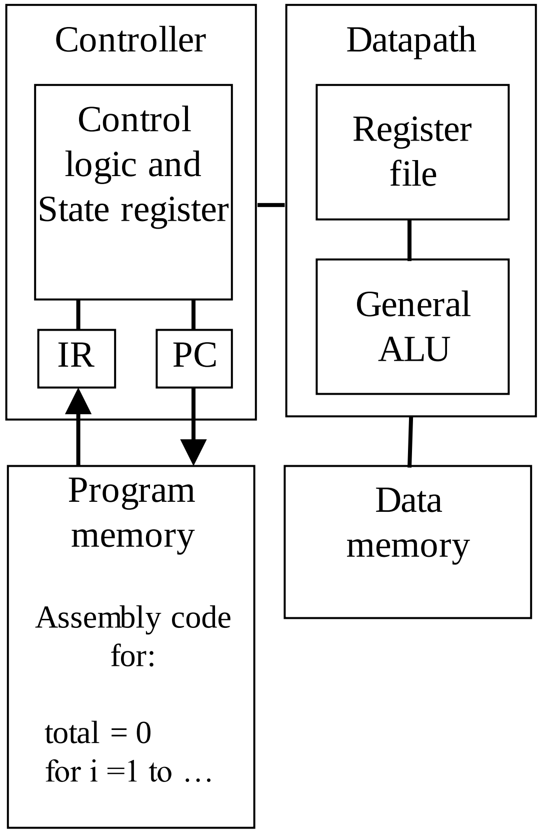
- One particular computation task
- Fast and power efficient
- Functional inflexibility and high cost at low volumes (maybe)

GP/SP Processor Architecture



General-purpose processors

- Programmable device used in a variety of applications
  - Also known as “microprocessor”
- Features
  - Program memory
  - General datapath with large register file and general ALU
- User benefits
  - Low time-to-market and NRE costs
  - High flexibility
- Examples
  - Pentium, Athlon, PowerPC



Application-specific IS processors (ASIPs)

Programmable processor optimized for a particular class of applications having common characteristics

- Compromise between general-purpose and ASIC (custom hardware)

Features

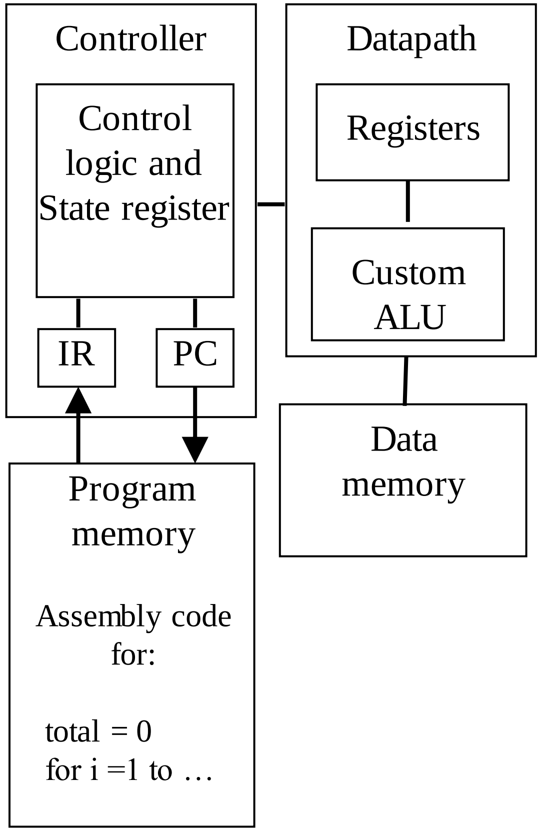
- Program memory
- Optimized datapath
- Special functional units

Benefits

- Some flexibility, good performance, size and power

Examples

- DSPs, Video Signal Processors, Network Processors,..



Application-Specific ICs (ASICs)

Digital circuit designed to execute exactly one program

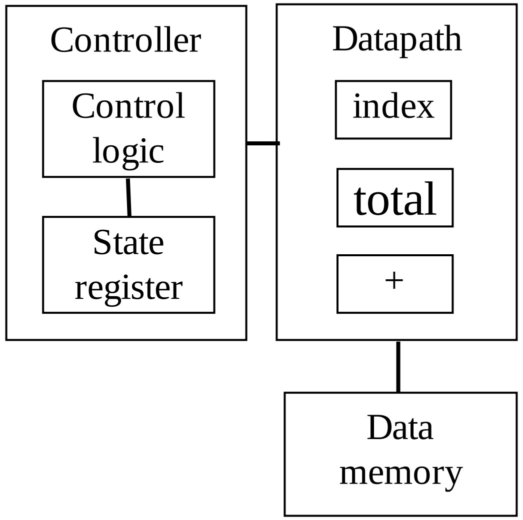
- coprocessor, hardware accelerator

Features

- Contains only the components needed to execute a single program
- No program memory

Benefits

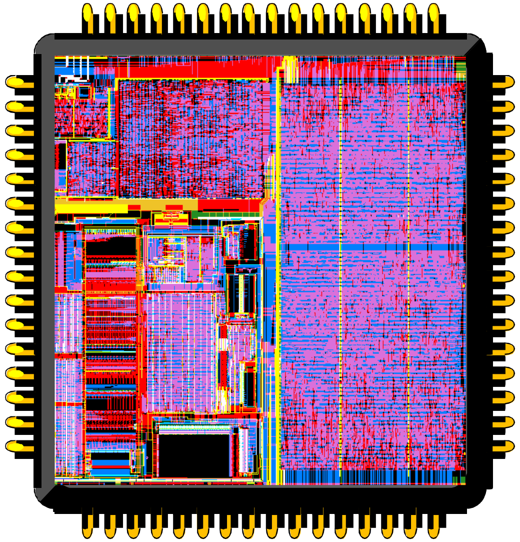
- Fast
- Low power
- Small size



Application Specific Circuits (ASIC)

Custom-designed circuits necessary if ultimate speed or energy efficiency is the goal and large numbers can be sold.

Approach suffers from long design times and high costs.



GP vs. SP Processors

GP:

- Programmable controller
  - Control logic is stored in memory
  - Fetch/decode overhead
- Highly general data-path
  - Typical bit-width (8, 16, 32, 64)
  - Complete set of arithmetic/logic units
  - Large set of registers
- High NRE/sale-volume

ASIC:

- Hardwired controller
  - No need for program memory and cache
  - No fetch/decode overhead
- Highly tuned data-path
  - Custom bit-width
  - Custom arithmetic/logic units
  - Custom set of registers
- Low NRE/sale-volume

Storage

What is a memory?	Artifact that stores bits Storage fabric and access logic
Write-ability	Manner and speed a memory can be written
Storage-permanence	ability of memory to hold stored bits after they are written
Many different types of memories	Flash, SRAM, DRAM, etc.
Common to compose memories	

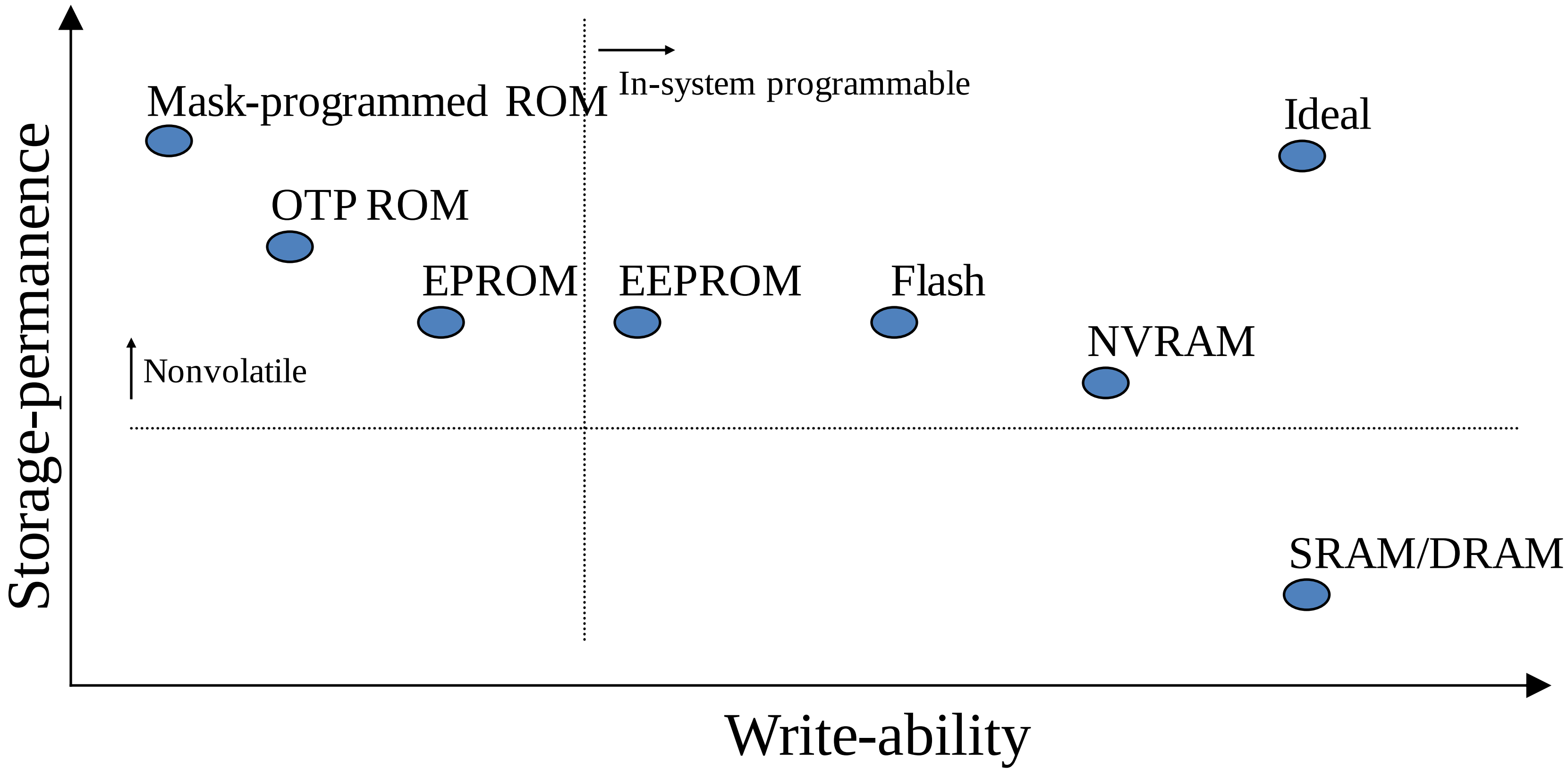
Write-ability

- Ranges of write ability
  - High end
    - Processor writes to memory simply and quickly
    - E.g., RAM
  - Middle range
    - Processor writes to memory, but slower
    - E.g., FLASH, EEPROM
  - Lower range
    - Special equipment, “programmer”, must be used to write to memory
    - E.g., EPROM, OTP ROM
  - Low end
    - Bits stored only during fabrication
    - E.g., Mask-programmed ROM

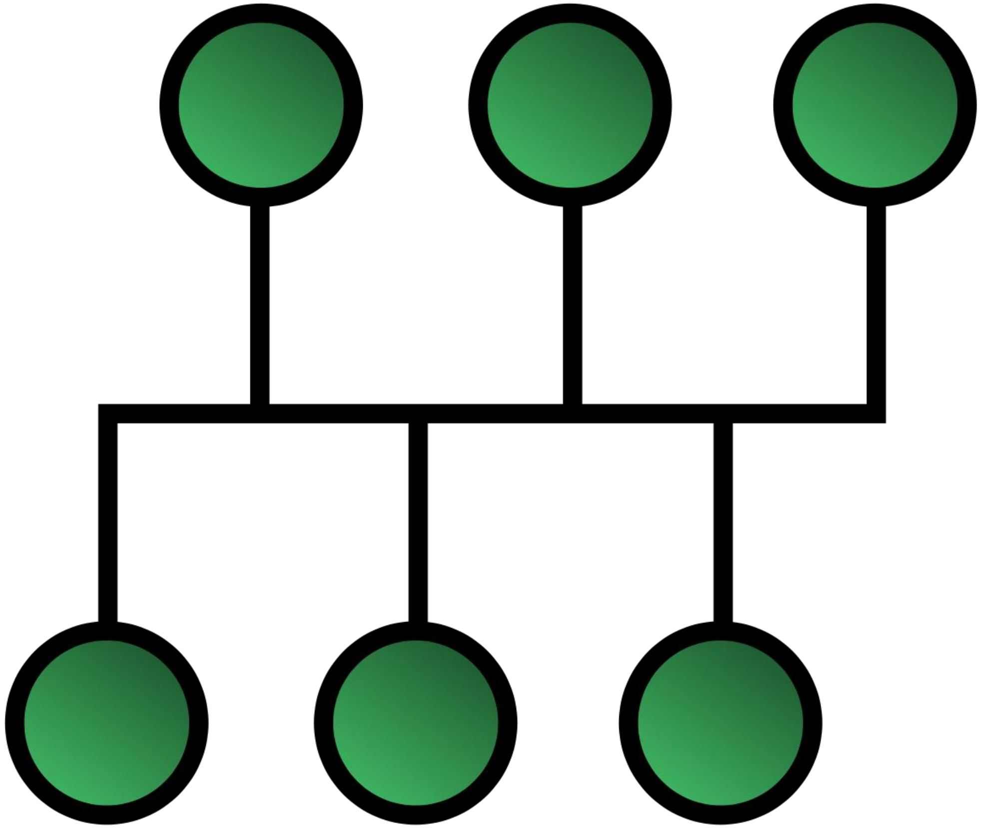
Storage-permanence

- Range of storage permanence
  - High end
    - Essentially never loses bits
    - E.g., mask-programmed ROM
  - Middle range
    - Holds bits days/months/years after memory's power source turned off
    - E.g., NVRAM
  - Lower range
    - Holds bits as long as power supplied to memory
    - E.g., SRAM
  - Low end
    - Begins to lose bits almost immediately after written
    - E.g., DRAM

Memory Types



Communication



- What is a bus?
  - An artifact that transfers bits
  - Wires, air, or fiber and interface logic
- Associated with a bus, we have:
  - Connectivity scheme
    - Serial Communication
    - Parallel Communication
    - Wireless Communication
  - Protocol
    - Ports
    - Timing Diagrams
    - Read and write cycles
  - Arbitration scheme, error detection/correction, DMA, etc.

Serial Communication

A single wire used for data transfer

One or more additional wires used for control (but, some protocols may not use additional control wires)

Higher throughput for long distance communication

- Often across processing node

Lower cost in terms of wires (cable)

E.g., USB, Ethernet, RS232, I^2C, etc.

Parallel Communication

- Multiple buses used for data transfer
- One or more additional wires used for control
- Higher throughput for short distance communication
  - Data misalignment problem
  - Often used within a processing node
- Higher cost in terms of wires (cable)
- E.g., ISA, AMBA, PCI, etc.

Wireless Communication

Infrared (IR)

Electronic wave frequencies just below visible light spectrum

Diode emits infrared light to generate signal

Infrared transistor detects signal, conducts when exposed to infrared light

Cheap to build

Need line of sight, limited range

Radio frequency (RF)

Electromagnetic wave frequencies in radio spectrum

Analog circuitry and antenna needed on both sides of transmission

Line of sight not needed, transmitter power determines range

Peripherals

Perform specific computation task

Custom single-purpose processors

Designed by us for a unique task

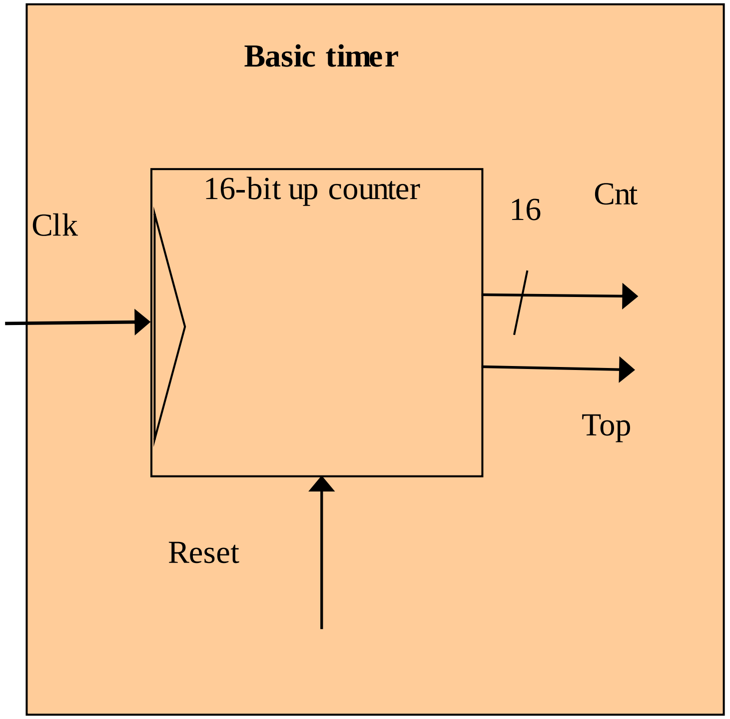
Standard single-purpose processors

"Off-the-shelf"

pre-designed for a common task

Timers

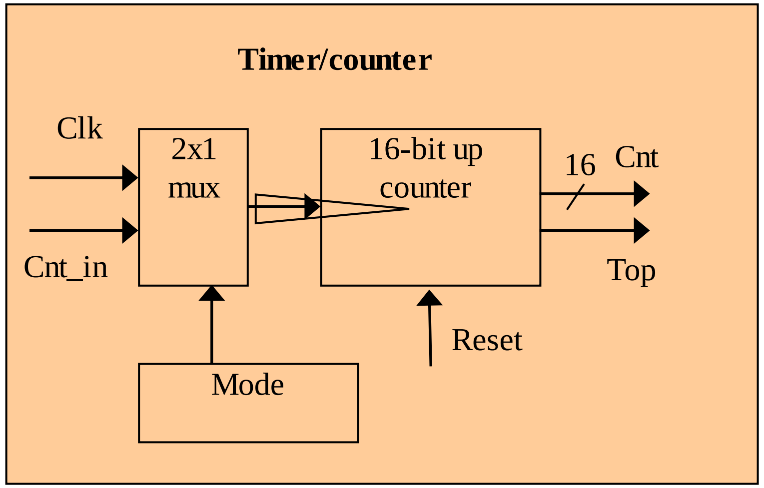
- Timers: measure time intervals
  - To generate timed output events
  - To measure input events
  - Top: max count reached
- Range and resolution



Counters

- Counter: like a timer, but counts pulses on a general input signal rather than clock

  - e.g., count cars passing over a sensor
  - Can often configure device as either a timer or counter

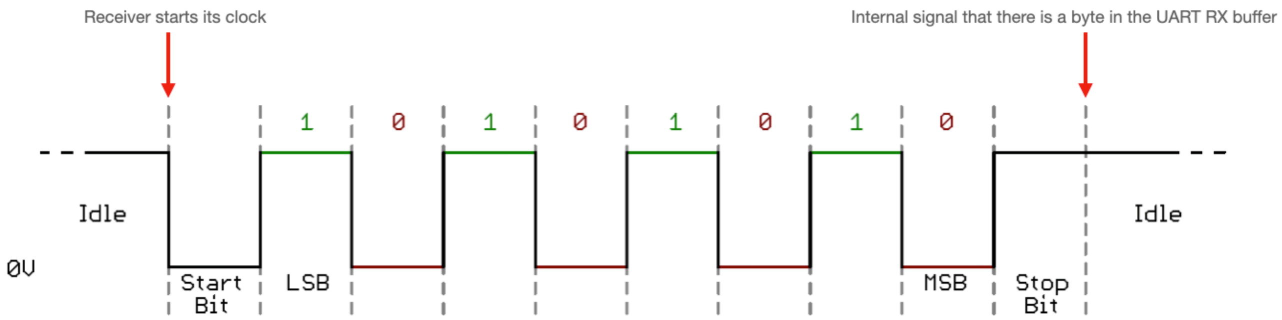


Watchdog Timer

Must reset timer every X time unit, else timer generates a signal

Common use: detect failure, self-reset

UART



UART: Universal Asynchronous Receiver Transmitter

- Takes parallel data and transmits serially
- Receives serial data and converts to parallel

Parity: extra bit for simple error checking

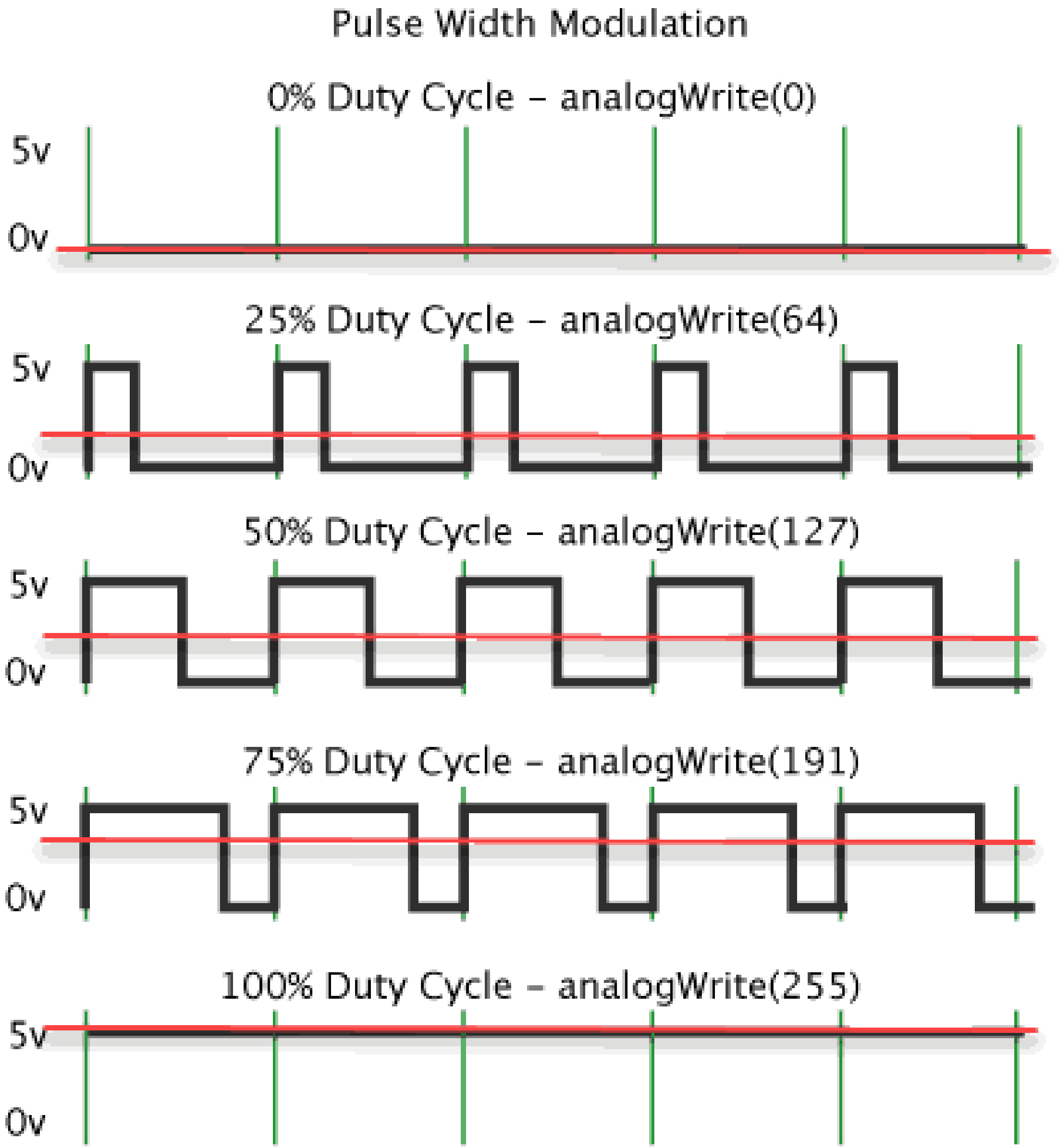
Start bit, stop bit

Baud rate

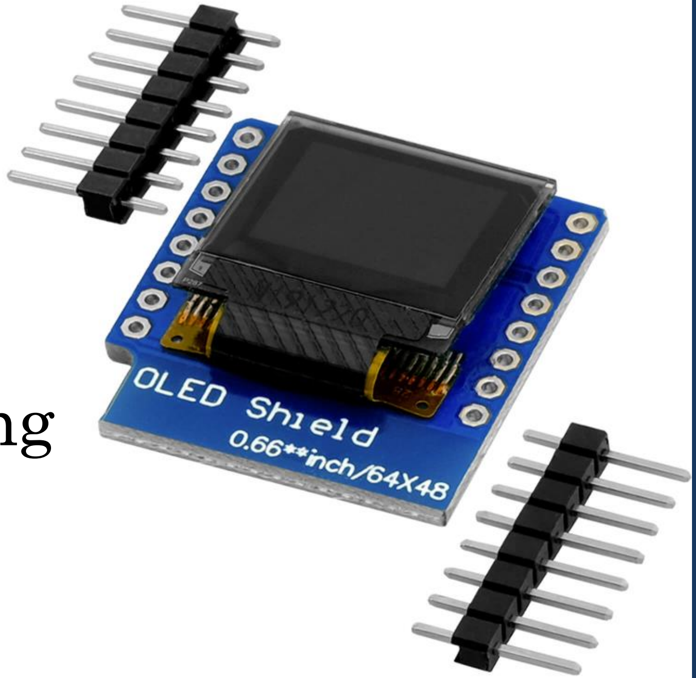
- Signal changes per second
- Bit rate, sometimes different

Pulse Width Modulator (PWM)

- Generates pulses with specific high/low times
- Duty cycle: % time high
  - Square wave: 50% duty cycle
- Common use: control average voltage to electric device
  - Simpler than DC-DC converter or digital-analog converter
  - DC motor speed, dimmer lights


Pulse Width Modulation
0% Duty Cycle - analogWrite(0)
5v
0v
25% Duty Cycle - analogWrite(64)
5v
0v
50% Duty Cycle - analogWrite(127)
5v
0v
75% Duty Cycle - analogWrite(191)
5v
0v
100% Duty Cycle - analogWrite(255)
5v
0v

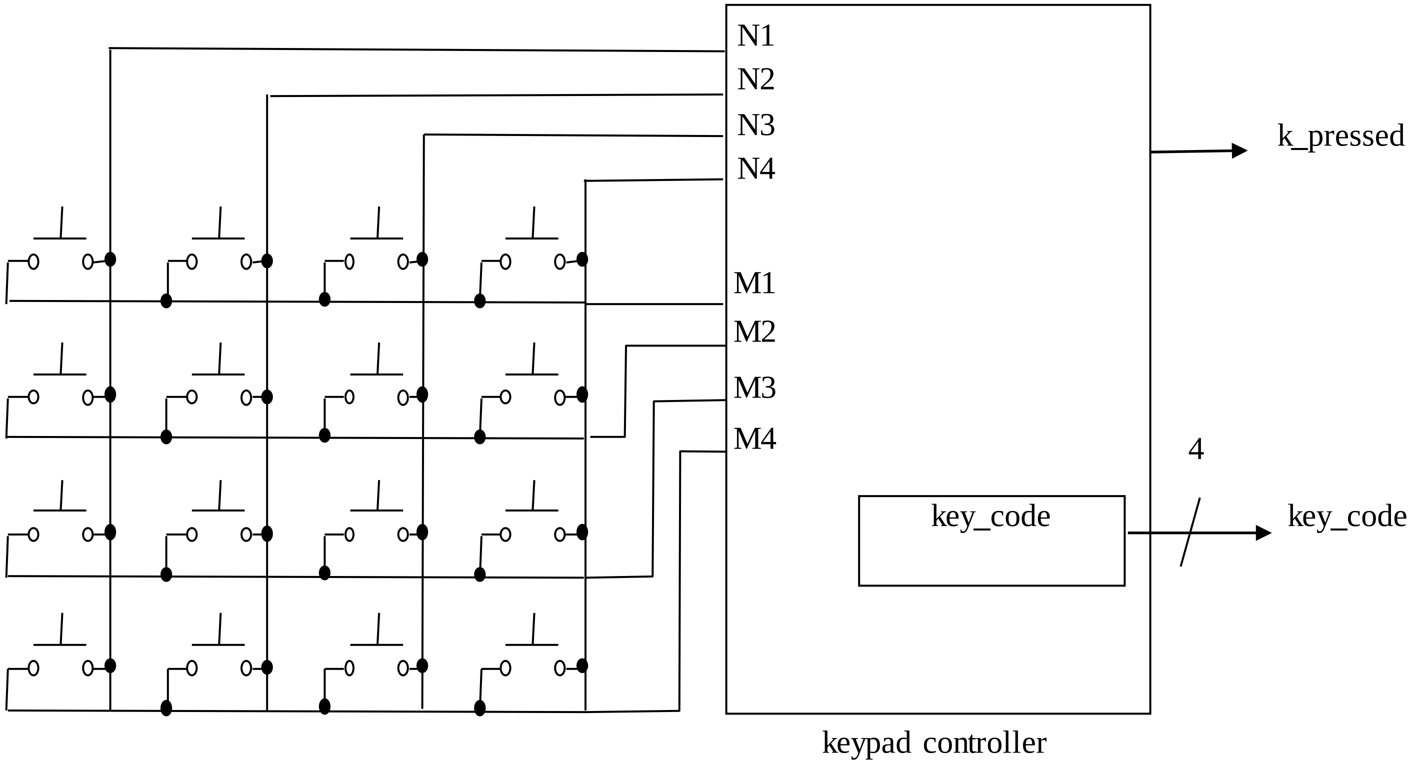
LCD and OLED



- Liquid Crystal Display
- N rows by M columns
- Controller build into the LCD module
- Simple microprocessor interface using ports
- Software controlled

- Organic Light emitting Diodes
- N rows by M columns
- Communicates over serial
- Software controlled

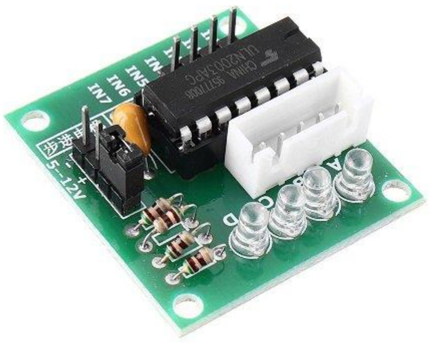
Keypad

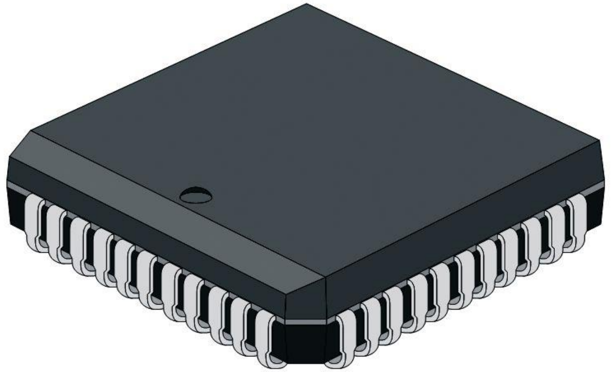

N1
N2
N3
N4
M1
M2
M3
M4
k_pressed
4
key_code
key_code
keypad controller

N=4, M=4

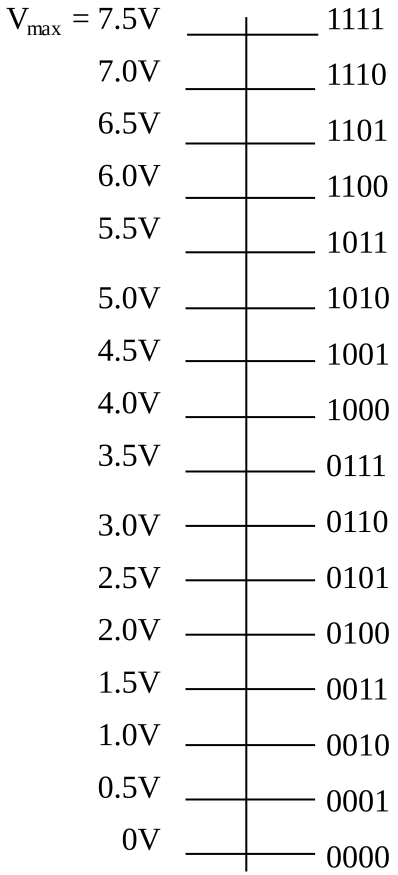
Stepper Motor Controller

- Stepper motor: rotates fixed number of degrees when given a “step” signal
  - In contrast, DC motor just rotates when power applied, coasts to stop
- Rotation achieved by applying specific voltage sequence to coils
- Controller greatly simplifies this

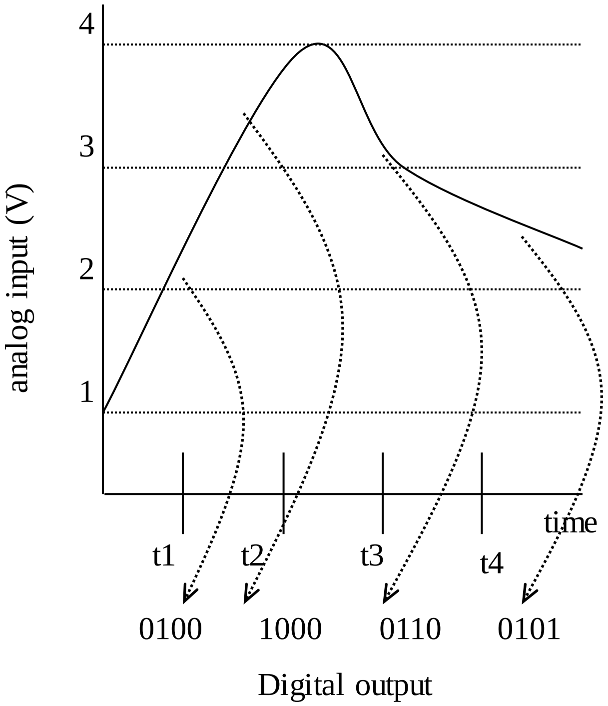




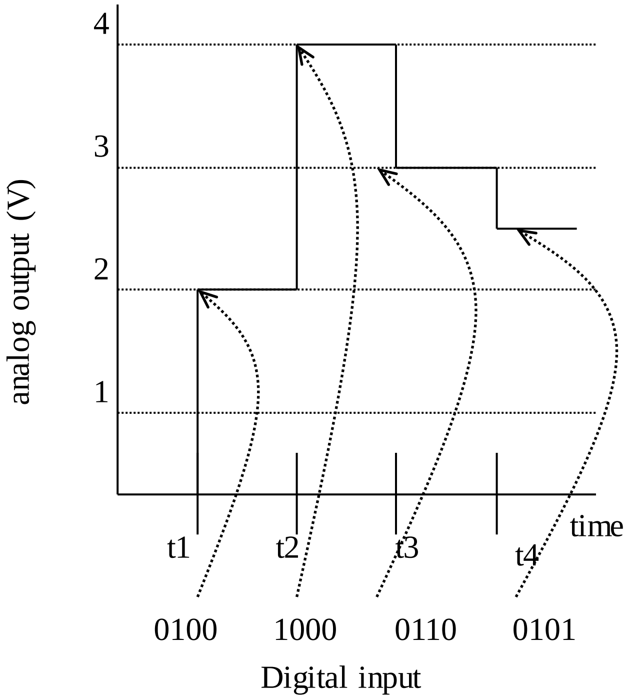
Analog-to-Digital Converter



proportionality



Analog to Digital (A/D)



Digital to Analog (D/A)

Summary

- Hardware: Information Processing
  - Processors
  - Memories
  - Communication
  - Peripherals