Module-1:Introduction
Slide Set-1

Introduction to Embedded System Design

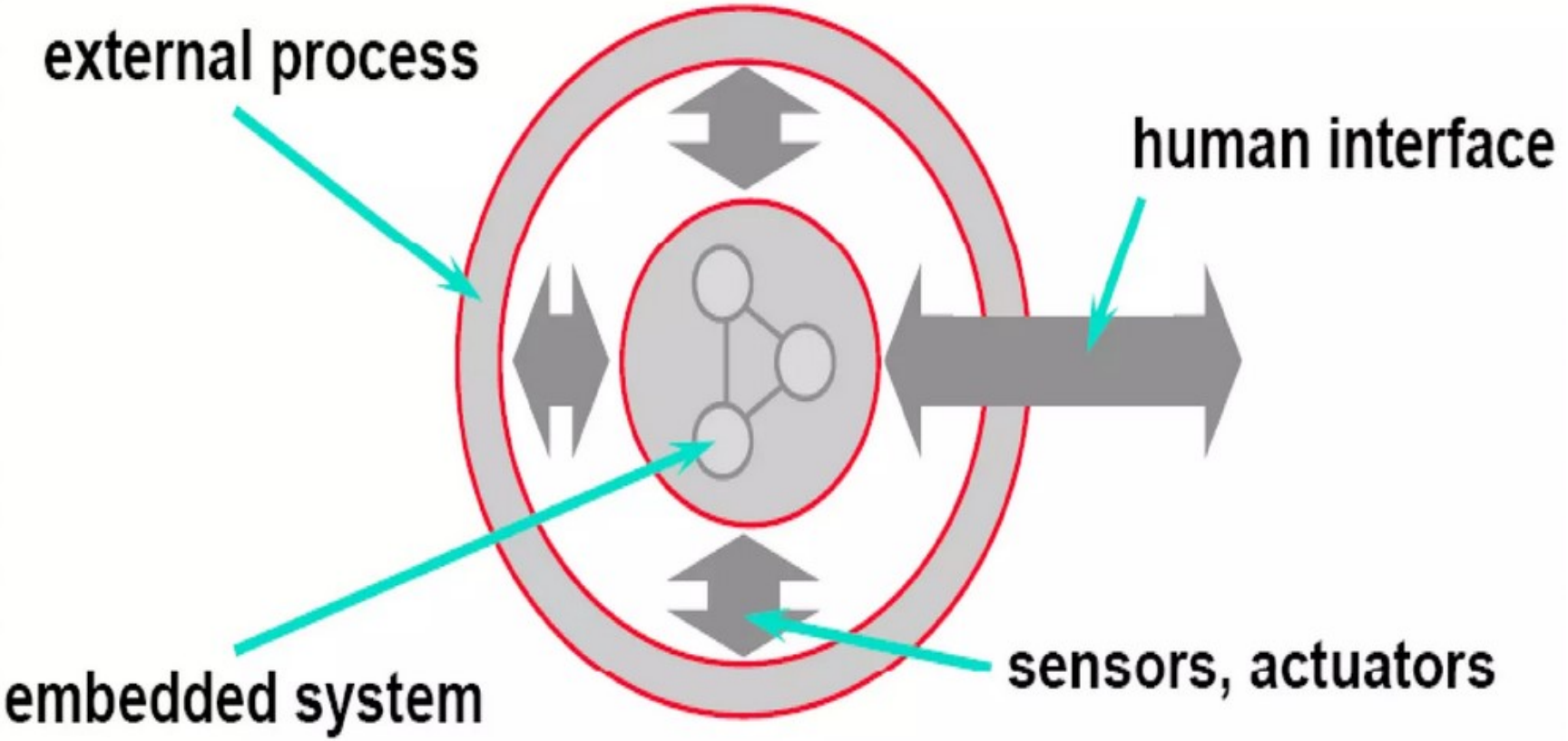
- *General Introduction to Embedded Systems*
- Hardware Platforms and Components
  - System Specialization
  - Application Specific Instruction Sets
    - Micro Controller

What is Embedded System?..

Embedded systems (ES) = **information processing systems embedded into a larger product**

An **embedded system** is a computer system designed to do one or a few dedicated and/or specific functions often with real-time computing constraints. It is *embedded* as part of a complete device often including hardware and mechanical parts.

What is Embedded System?..



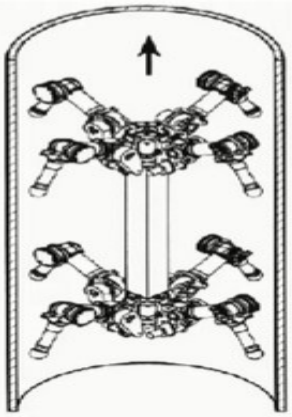
Application areas (1)

- Automotive electronics
- Avionics
- Trains
- Telecommunication

Application areas (2)

- Robotics

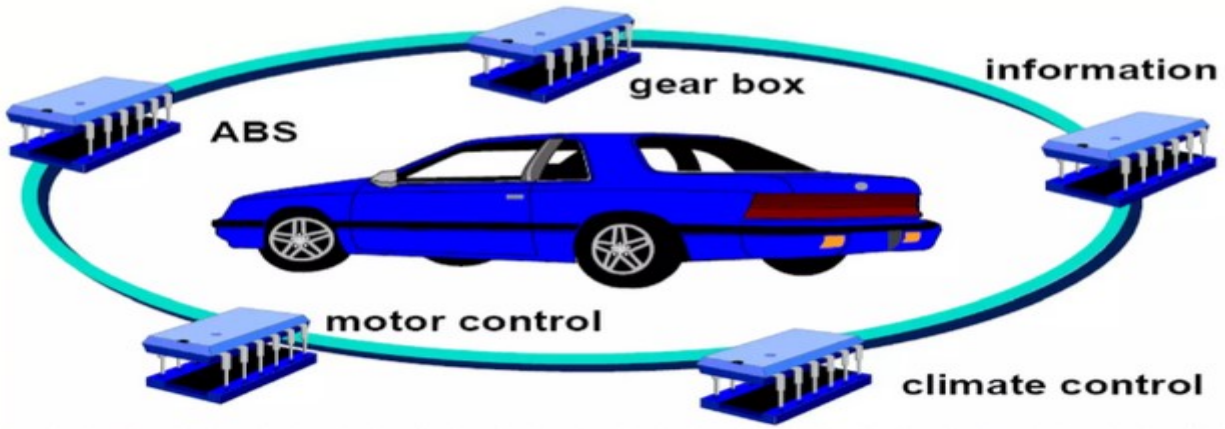
"Pipe-climber"



Robot "Johnnie" (Courtesy and ©: H.Ulbrich, F. Pfeiffer, TU München)

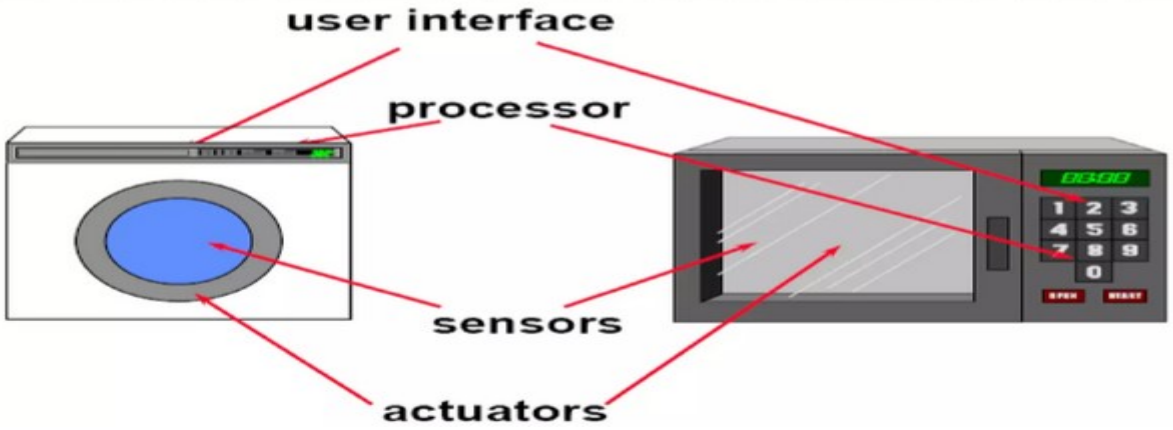
Examples of Embedded Systems

- Car as an integrated control, communication and information system.



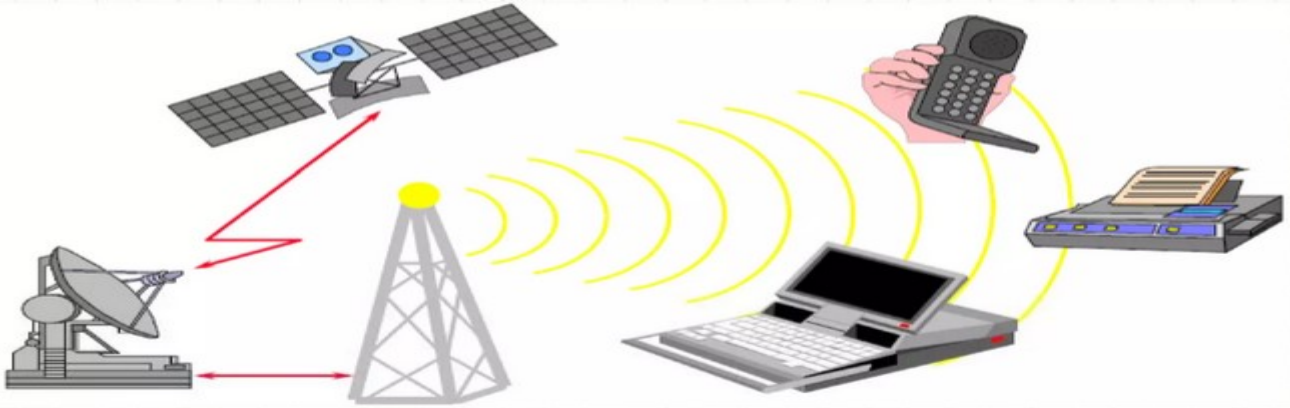
Examples of Embedded Systems

Consumer electronics, for example MP3 Audio, digital camera, home electronics, … .



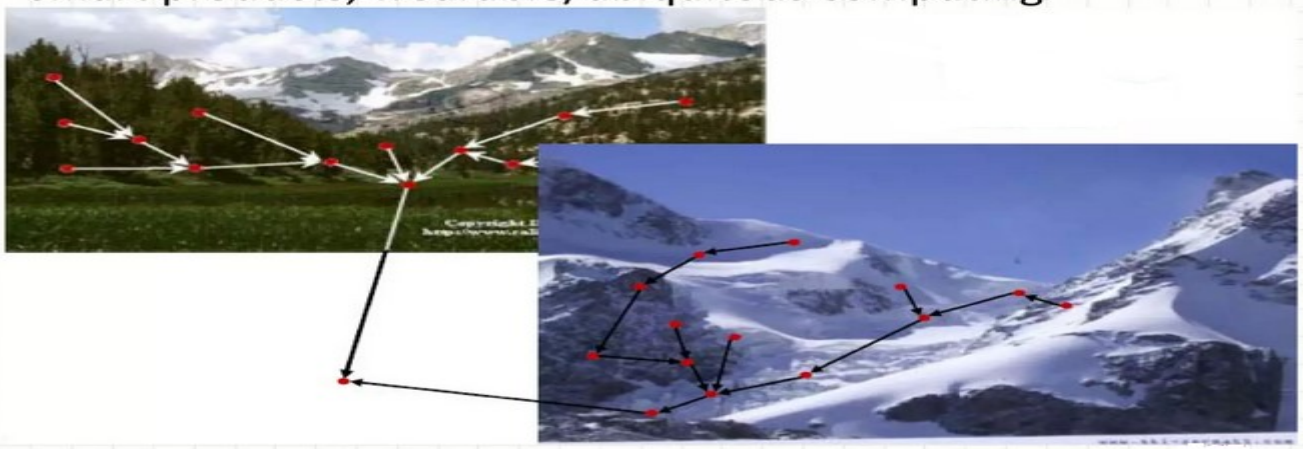
Examples of Embedded Systems

Information systems, for example wireless communication (mobile phone, Wireless LAN, ...), end-user equipment, router, ...



Communicating Embedded Systems

- sensor networks (civil engineering, buildings, environmental monitoring, traffic, emergency situations)
- smart products, wearable/ubiquitous computing



Characteristics of Embedded Systems (1)

- Must be ***dependable:***
  - ***Reliability: R(t)*** = *probability of system working correctly* provided that is was working at *t=0*
  - ***Maintainability:*** *M(d) = probability of system working* correctly *d time units after error occurred.*
  - ***Availability:*** *probability of system working at time t*
  - ***Safety:*** *no harm to be caused*
  - ***Security:*** *confidential and authentic communication*

Even perfectly designed systems can fail if the assumptions about the workload and possible errors turn out to be wrong. Making the system dependable must not be an after-thought, it must be considered from the very beginning.

Characteristics of Embedded Systems (2)

- Must be ***efficient:***
  - ***Energy*** *efficient* 
  - ***Code-size*** *efficient (especially for systems on a chip)* 
  - ***Run-time*** *efficient* 
  - ***Weight*** *efficient* 
  - ***Cost*** *efficient* 

- ***Dedicated*** *towards a certain* ***application:*** *Knowledge about* behavior at design time can be used to minimize resources and to maximize robustness.
- ***Dedicated*** *user* ***interface*** *(no mouse, keyboard and screen).* 

Characteristics of Embedded Systems (3)

- Many ES must meet ***real-time constraints:***
  - A real-time system must ***react to stimuli from the controlled*** object (or the operator) within the time interval dictated by the environment.
  - For real-time systems, right answers arriving too late (or even too early) are wrong.
- ***"A real-time constraint is called hard, if not meeting thatconstraint could result in a catastrophe"[Kopetz, 1997].***
- All other time-constraints are called soft.
- A guaranteed system response has to be explained without statistical arguments.

Characteristics of Embedded Systems (4)

- Frequently ***connected to physical environment*** *through* sensors and actuators,
- ***Hybrid systems*** *(analog + digital parts).*
- Typically, ES are ***reactive systems:***
- ***"A reactive system is one which is in continual interaction with is environment and executes at a pace determined by that environment" [Bergé, 1995]***
- Behavior depends on input and current state.
  - automata model often appropriate,

Comparison

Embedded Systems

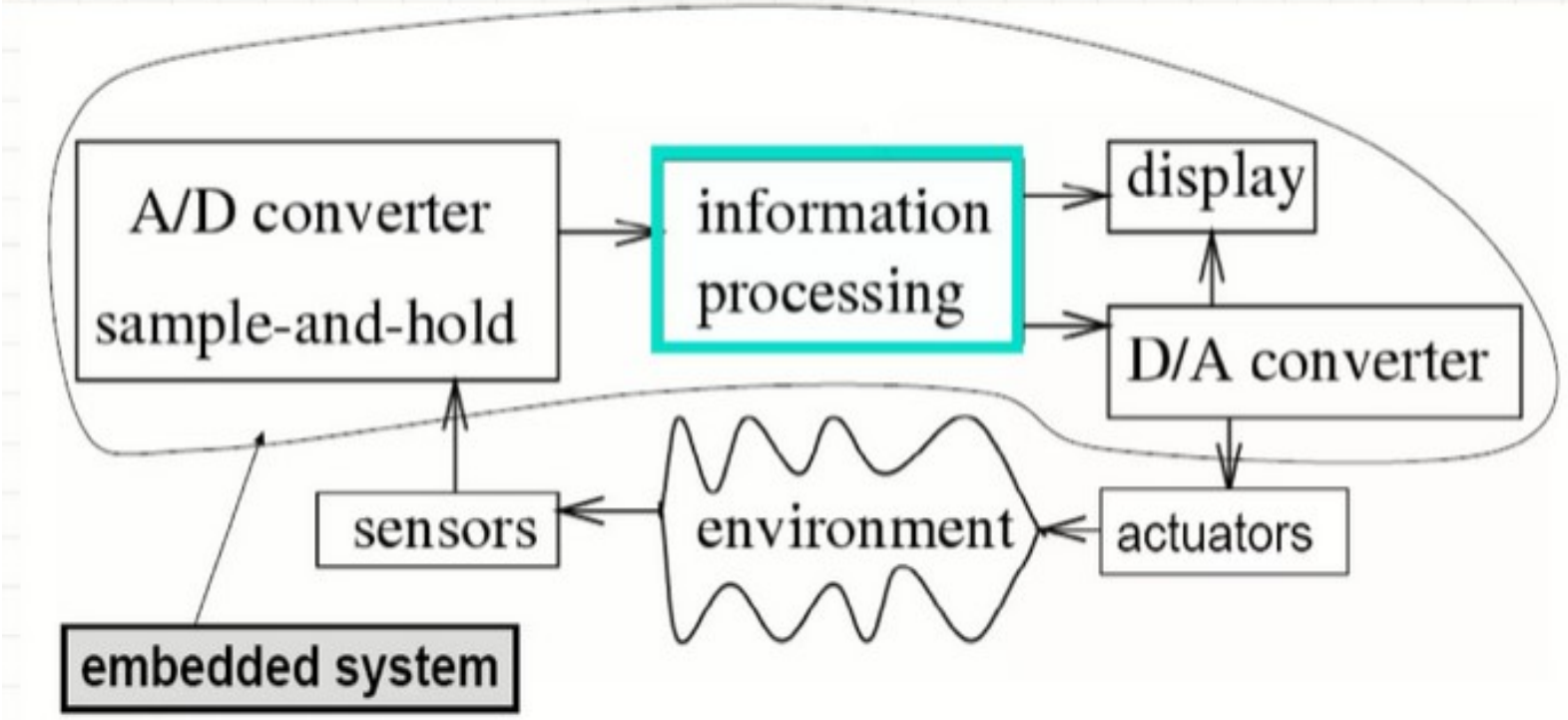
- Few applications that are known at design-time.
- Not programmable by end user.
- Fixed run-time equirements (additional computing power not useful).
- Criteria:
  - cost
  - power consumption
  - predictability

General Purpose Computing

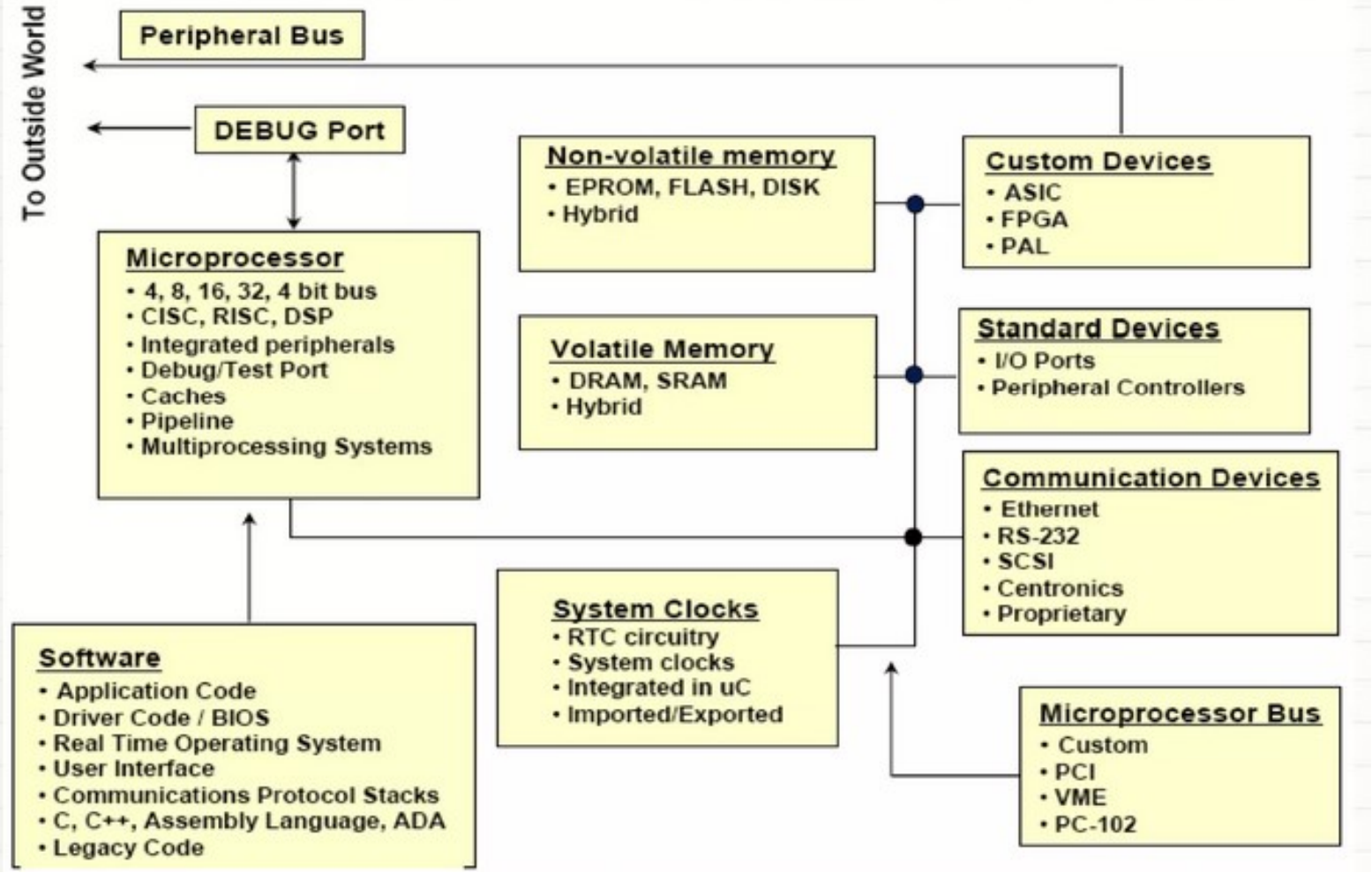
- Broad class of applications.
- Programmable by end user.
- Faster is better.
- Criteria:
  - cost
  - average speed

Embedded System Hardware

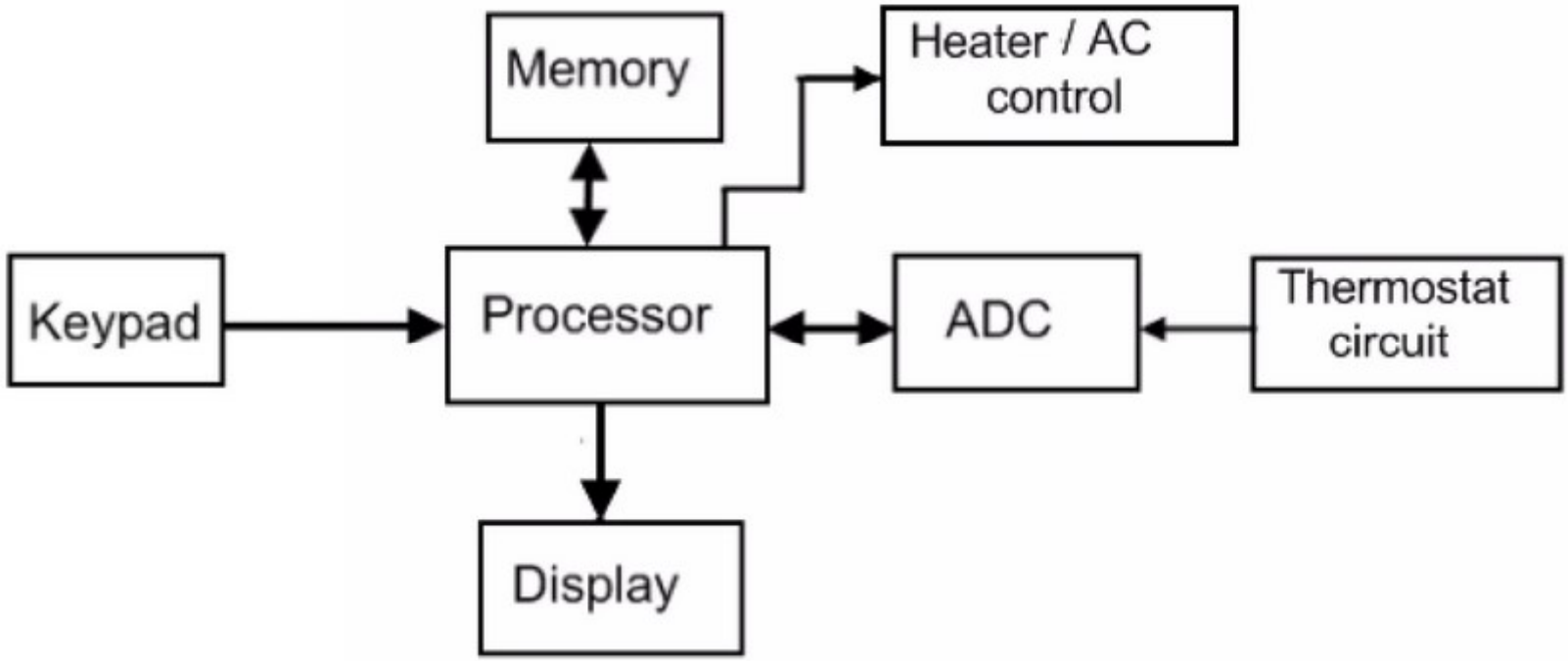
- Embedded system hardware is frequently used in a loop (***"hardware in a loop"):***



Typical Architecture

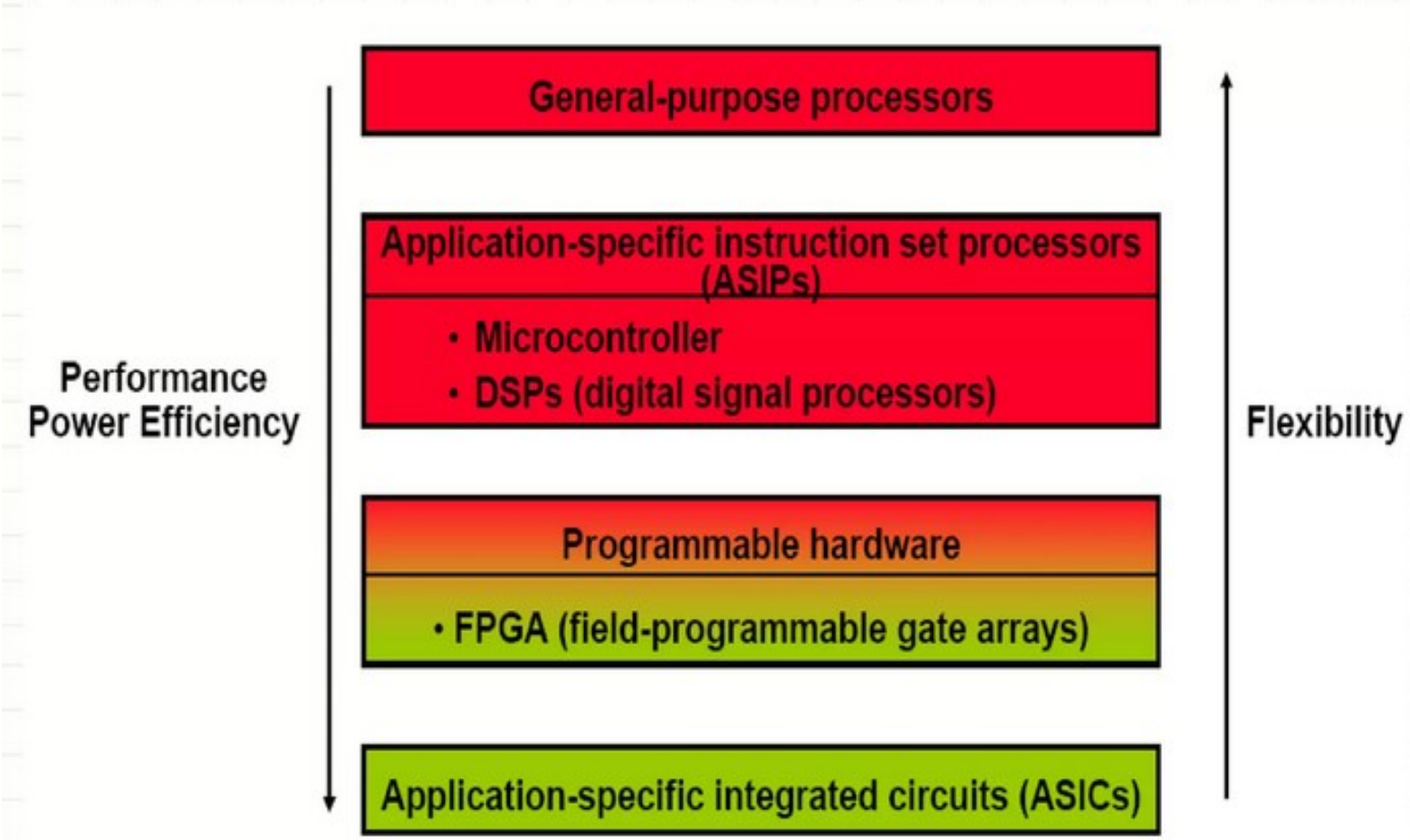


Example: Digital Thermostat



A digital thermostat as an example of embedded system

Implementation Alternatives



General-purpose Processors

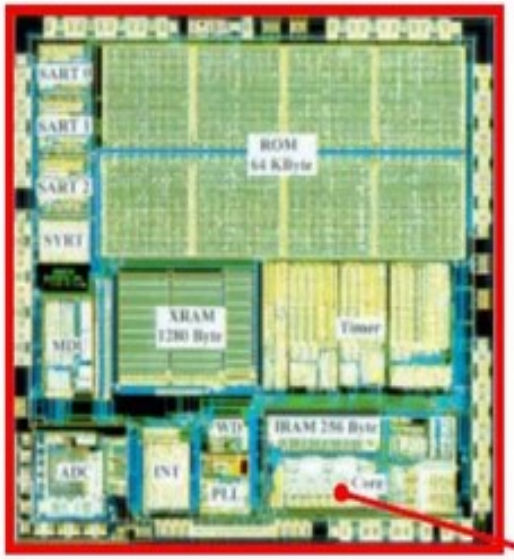
- ***High performance***
  - Highly optimized circuits and technology
  - Use of parallelism
    - superscalar: dynamic scheduling of instructions
    - super-pipelining: instruction pipelining, branch prediction, speculation
  - complex memory hierarchy
- ***Not suited for real-time applications***
  - Execution times are highly unpredictable because of intensive resource sharing and dynamic decisions
- ***Properties***
  - Good average performance for large application mix
  - High power consumption

System Specialization

- The main difference between general purpose highest volume microprocessors and embedded systems is ***specialization.***
- ***Specialization should respect flexibility***
  - application domain specific systems shall cover a class of applications
  - some flexibility is required to account for late changes, debugging
- ***System analysis required***
  - identification of application properties which can be used for specialization
  - quantification of individual specialization effects

Microcontroller

- control-dominant applications
  - supports process scheduling and synchronization
  - preemption (interrupt),context switch
  - short latency times
- low power consumption
- peripheral units often integrated
- suited for real-time applications

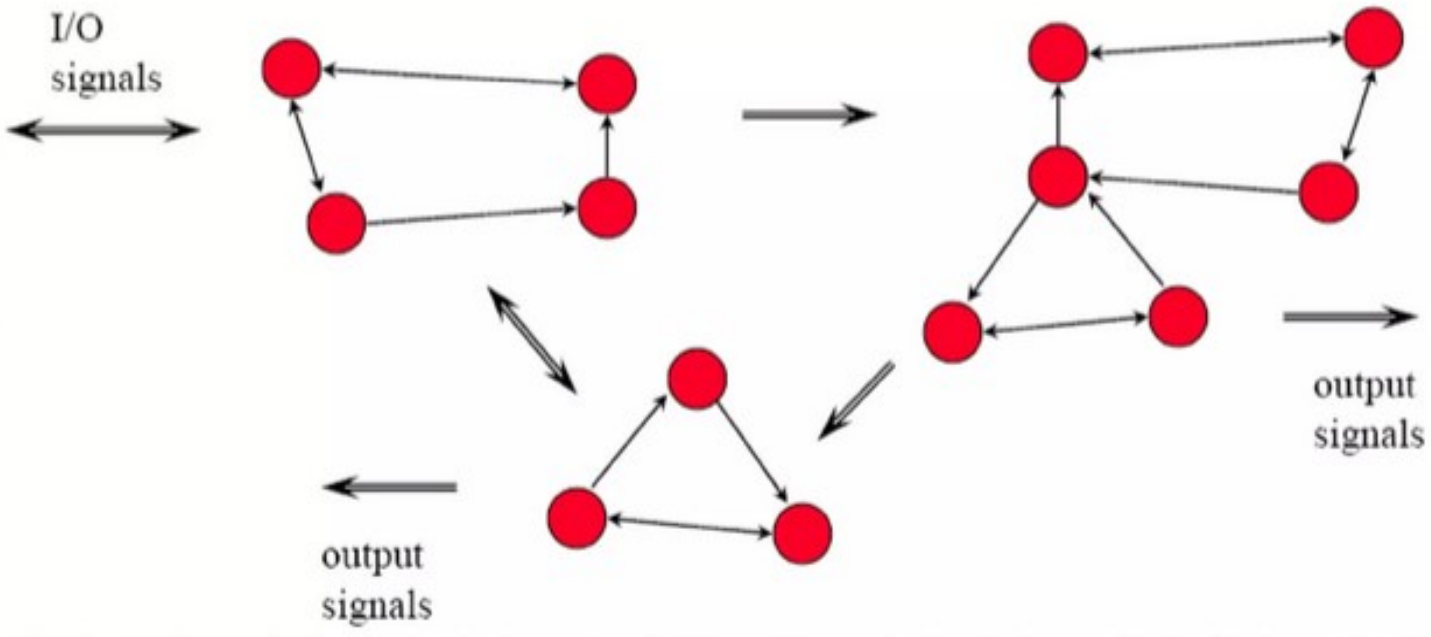


8051 core

SIECO51 (Siemens)

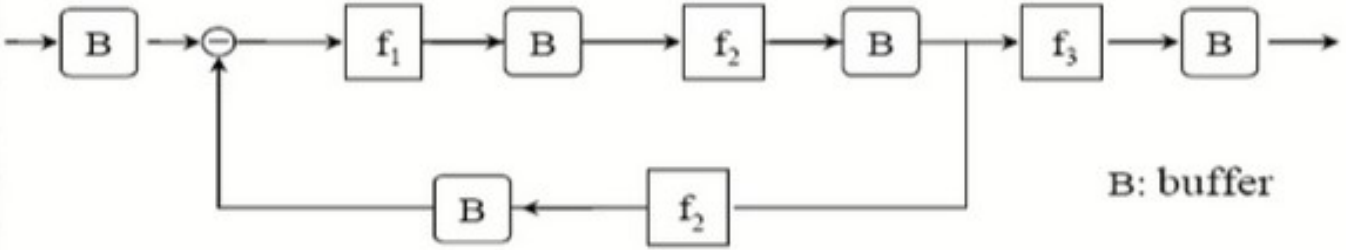
Control Dominated Systems

- Reactive systems with ***event driven behavior***
- Underlying semantics of system description ("input model of computation") typically (coupled) Finite State Machines



Data Dominated Systems

- ***Streaming oriented systems with mostly periodic*** behavior
- Underlying semantics of input description e.g. ***flow graphs ("input model of computation")***



- ***Application examples: signal processing, control*** engineering

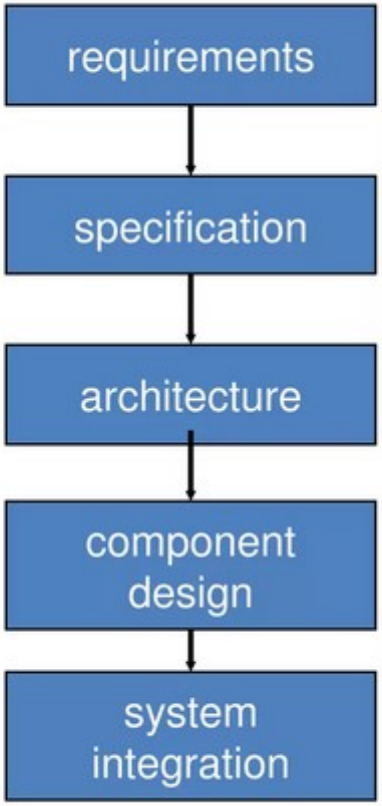
Design Metrics/Goals

- Reliability
- Power Consumption
- Cost :NRE cost & Manufacturing cost
- Weight/size
- Time: time to prototype & time to market
- Flexibility
- Performance
- Debuggability
- Safty & maintaince
- Maximum usage of resources

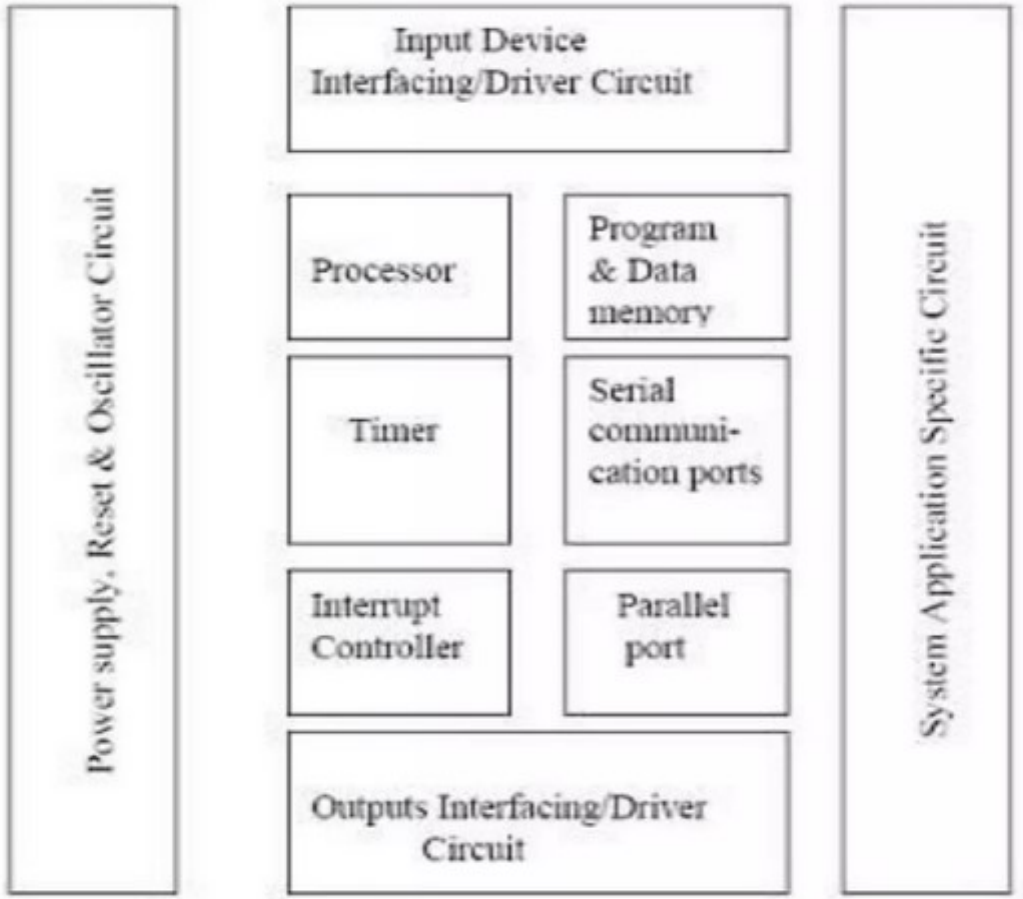
Challenges

- How much hardware do we need?
  - How big is the CPU? Memory?
- How do we meet our deadlines?
  - Faster hardware or cleverer software?
- How do we minimize power?
  - Turn off unnecessary logic? Reduce memory accesses?
- Does it really work?
  - Is the specification correct?
  - Does the implementation meet the spec?
  - Reliability in safety-critical systems
- How do we work on the system?
  - Complex testing
    - How do we test for real-time characteristics?
    - How do we test on real data?
  - Limited observability and controllability
  - Restricted development environments
    - What is our development platform?

Level Abstraction



Architecture



Components of Embedded system

- H/W
  - **Processor**
  - **Power source and clock**
  - **Reset circuit**
  - **Memory Unit**
  - **Interrupt Handler**
  - **Linking Embedded System H/W**
  - **I/O communication Unit**

- S/W
  - **ROM image/Applications/w**
  - **Programming Languages**
  - **Device Drivers**
  - **Program Models**

- RTOS/EOS

- S/W Tools
  - **Development tools**
  - **Simulator**
  - **Project Manager**
  - **IDE**

Types of Cores

Requirments

-Able to solve complex Algorithm
- Meet Deadlines
- No.of Bits to be operated
- Bus Width
- Clock Frequency
- Performance(MIPS/MFLOPS)

- GPP: General Purpose Processor
  - -Microprocessors
  - -Embedded Processors
- ASIP: Application Specific Instruction Processor
  - -Micro controller
  - -Embedded Micro controller
  - -DSP and media Processor
  - -Network Processor
- SPP: Single Purpose Processor
  - -Coprocessors **eg:Math-coprocessor**
  - -Accelerators **eg:java acce**
  - -Controllers **eg:DMA**

Types of Cores

- **ASIC/VLSI chip: Application specific Integrated circuit**

  -GPP/ASIP integrated with analog circuits on VLSI chip using EDA tools.

- **ASSP:Application Specific System Processor**

  -set top box

  -Mpeg

  -HD tv

  eg:vedio Processors

- **Multicore Processors/Multiprocessor using GPP**

  eg:Embedded firewall cum Router

Power Sources,Clock and Reset Circuit

- **Power supply**
  - -Own supply
  - -Supply from System
  - -Charge Pumps
- **Clock**
  - -External Clock supply
  - -Oscillator
  - -RTC
- **Reset**
  - -Power On
  - -External/Internal Reset
  - -WDT
  - -BOR

Memory Unit,Interrupt Handlers

Memory

- ROM/EPROM/FLASH (internal/External)
- RAM
- Caches

Interrupt Handler

- External port interrupt
- I/O,Timer, RTC, interrupts
- S/W Interrupt/Exceptions

Linking ES H/W

- Multiplexers

I/O communication Unit

- I/O,O/P devices
  - -Sensors,
  - -actuators
  - -converters
  - -keypads
  - -displays
- Buses
  - -Parallel Buses
  - -serial buses

RTOS/OS

It performs functions

- Multitasking
- Scheduling
- Management
- Resource protection
- Interprocess Communication

eg: Ucos-II, Vxworks, windows CE, RT linux,QNX

Embedded Memories

- Internal RAM at µc *–SRAM used as reg, temp data, stack*
- RAM AT SoC or External RAM
- Internal/external caches at µp *-hold copy of system memory pages*
- External RAM chips *-DRAM used to hold extra data*
- Flash EPROM/EEPROM *-result stored in NV memory*
- ROM/PROM/MROM/OTP *-Application S/W, OS*
- Memory addresses at system ports *-RAM buffers*
- Memory Stick *-large storage such as audio, video*

I/O Devices

- DAC using PWM
- ADC
- LCD,LED and Touch Screen
- Keypad/keyboard/T9 keypad
- Pulse dialer
- Modem
- Transceiver
- Interrupt handler *-mechanism to handle various interrupts and also to deal with pending services*

Difference between RISC and CISC

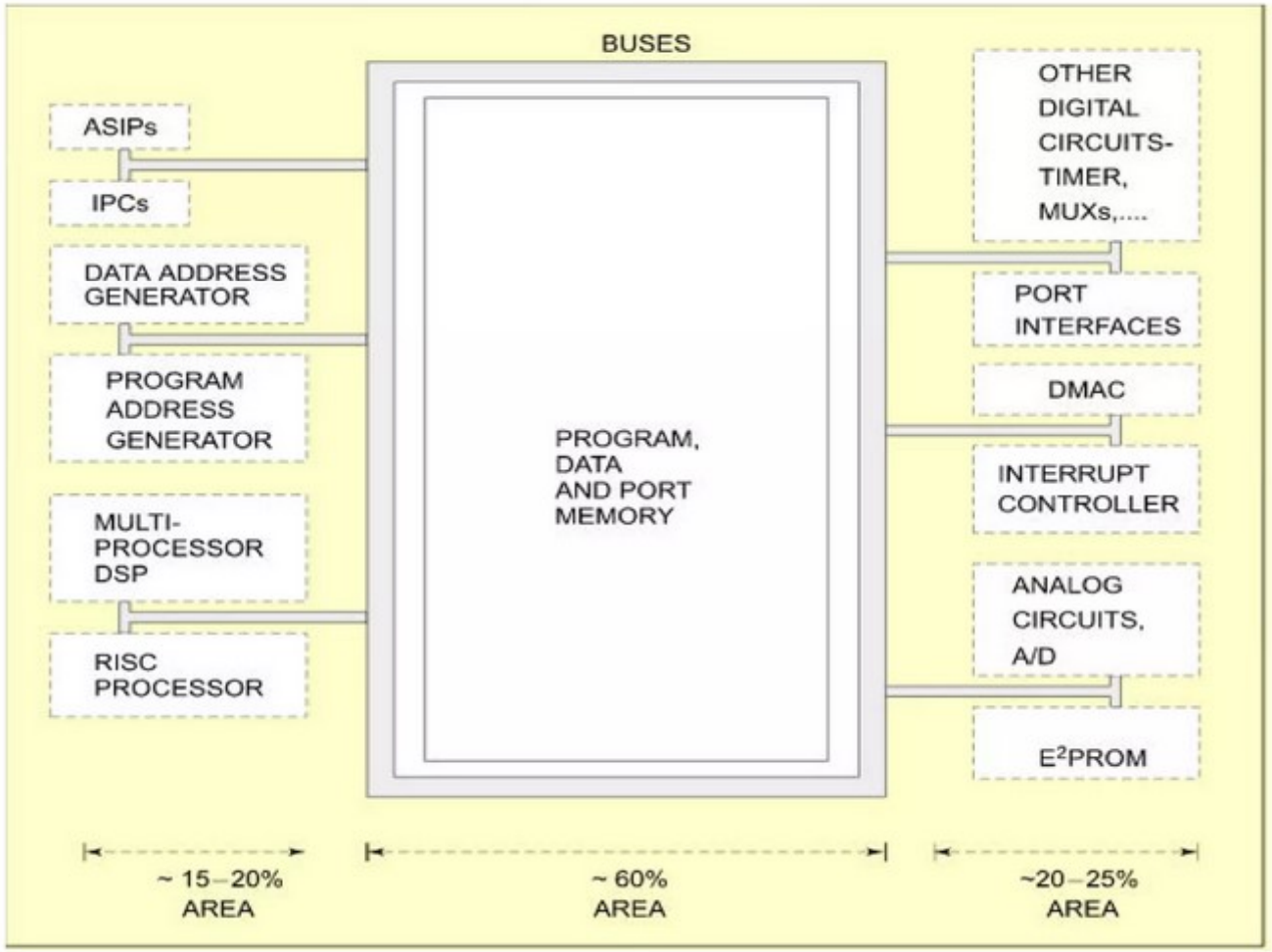
RISC	CISC
Reduced instruction set	Complex Instruction set
Maximum instructions are single cycle(fixed size),thus supports pipelining	Variable size Instructions,so generally do not have pipelining
Orthogonal instruction set	Non-Orthogonal
Operations are performed on registers, so large no of Registers. For memory only Load and Store	Operations are performed on both registers and memory. Limited number of GPRs

Difference between RISC and CISC

RISC	CISC
Hardwired Control unit	Microcode control Unit
Small in size with resp to die area and No.of pins	Comparatively large in size since more complex instruction needs to be implemented.
Harvard architecture	Harvard or Von-Neuman
Eg:PIC18,ARM	8051,8086

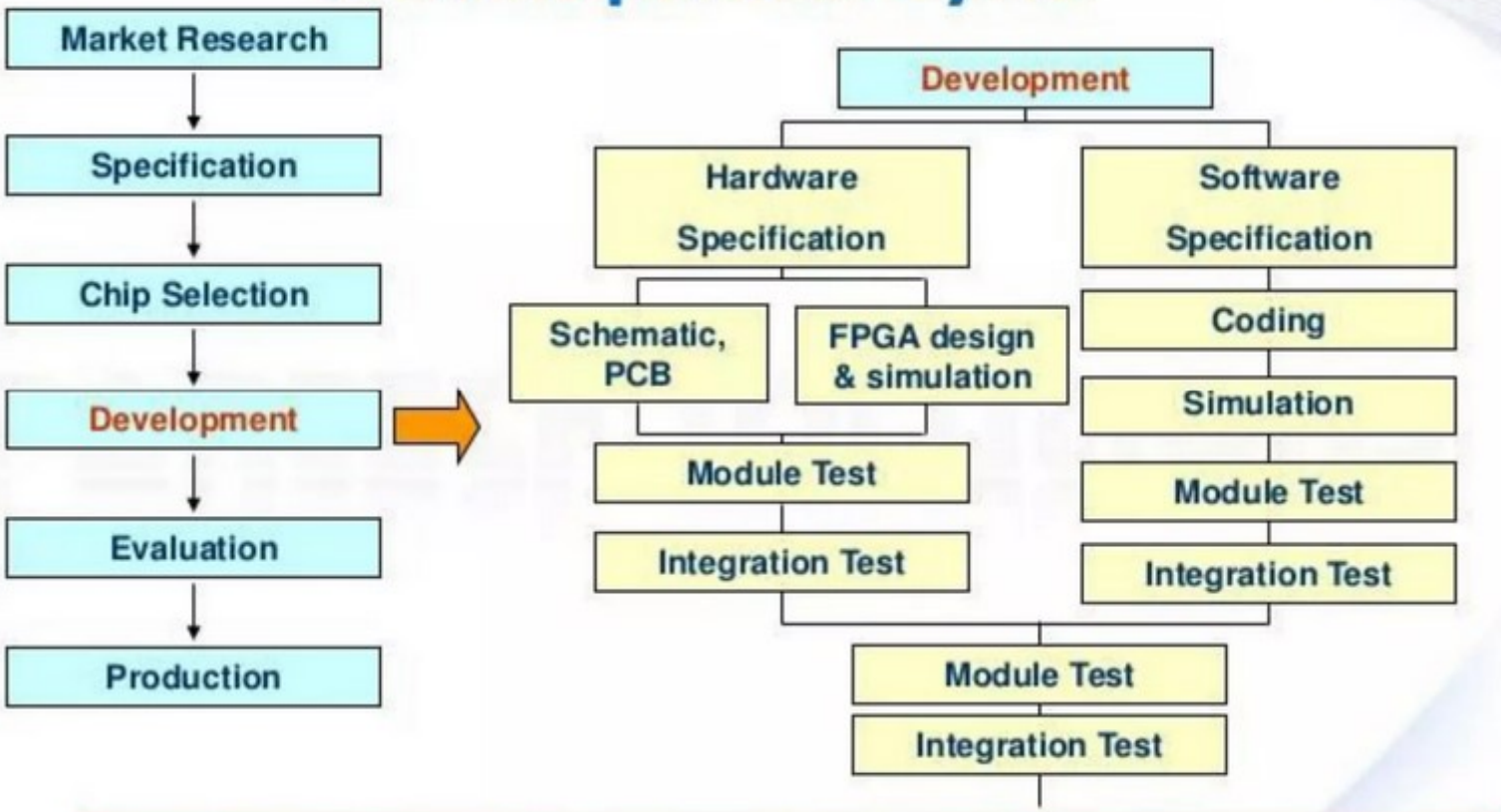
SoC

- System designed on a single chip
- Processor with all analog, digital and S/W build on a single VLSI chip



Development Process

Development Cycle



1. Statistic shows that Testing takes up major time in a Development cycle
2. Development Tools is an important factor to shorten the development time frame.

Levels of Abstraction from Top to Bottom

- Requirements
- Specifications
- Architecture
- Components
- System Integration

Requirement

Complete clarity of:

- Required Purpose
- Inputs
- Outputs
- Functioning
- Design metrics
- Validation requirements for finally developed system specifications
- Consistency in the requirements

Specifications

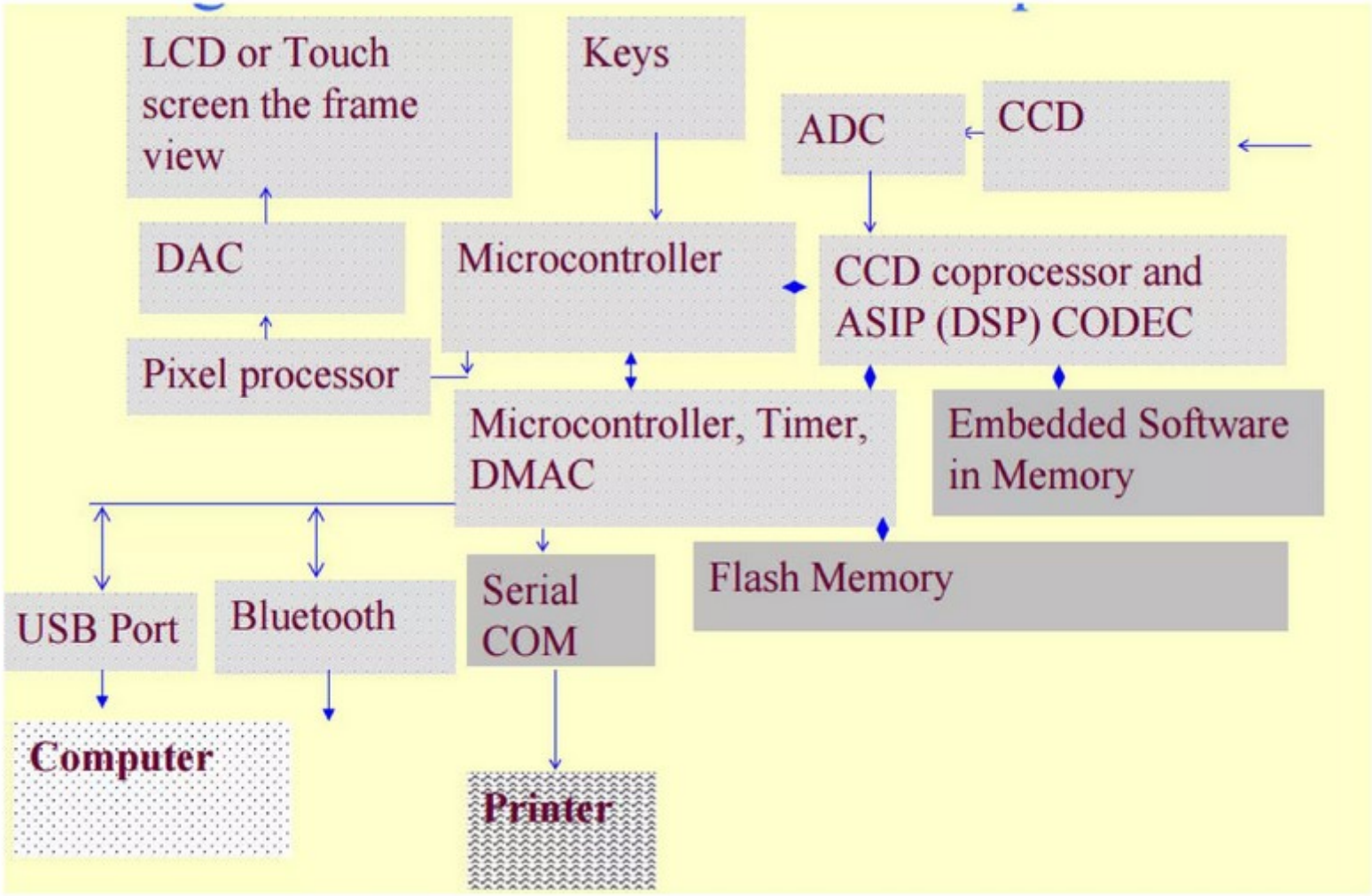
- Clear specification of customer expectations from the product
- Needs specification for
  - H/W, eg: Peripherals, Divices,Processors and memory specifications
  - Data types and processing specifications
- Expected system behavior specifications
- Constraints of design
- Expected lifecycle specifications of the product
- Process specifications analyzed by making list of I/Ps on event list, O/Ps on events, process activated on each event

H/W Components

- Processors, ASIP, Single Processors
- All Types of Memory as per requirement
- Internal and External peripherals and devices
- Ports and Buses in the system
- Power sources and battery

Design Examples

Digital Camera H/W Components



Digital Camera

- 4 M pixel/6 M pixel still images, clear visual display (ClearVid) CMOS sensor, 7 cm wide LCD photo display screen, enhanced imaging processor, double anti blur solution and high-speed processing engine, 10X optical and 20X digital zooms
- Record high definition video-clips. It therefore has speaker microphone(s) for high quality recorded sound.
- Audio/video Out Port for connecting to a TV/DVD player.

Digital Camera Arrangements

- Keys on the camera.
- Shutter, lens and charge coupled device (CCD) array sensors
- Good resolution photo quality LCD display unit
- Displays text such as image-title, shooting data and time and serial number. It displays messages. It displays the GUI menu when user interacts with the camera.
- Self-timer lamp for flash.

Internal Units

- Internal memory flash to store OS and embedded software and limited number of image files
- Flash memory stick of 2 GB or more for large storage.
- Universal Serial Bus (USB), Bluetooth and serial COM port for connecting it to computer, mobile and printer.

Digital Camera H/W

- Microcontroller or ASIP (Application Specific Instruction Set Processor)
- Multiple processors (CCDSP, DSP, Pixel Processor and others)
- RAM for storing temporary variables and stack
- ROM for application codes and RTOS codes for scheduling the tasks

Digital Camera S/W

- CCD signal processing for off-set correction
- JPEG coding
- JPEG decoding
- Pixel processing before display
- Memory and file systems
- Light, flash and display device drivers
- LCD, USB and Bluetooth Port device- drivers for port operations for display, printer and computer communication control