Introduction of Real-Time Embedded System Design

Department of Computer Science & Engineering
University of South Carolina
Spring, 2002

Outline

- Introduction
  - Real-time embedded systems
  - System-level design methodology
- Real time scheduling
- Power aware design
- Summary

What is the real-time embedded system?

- Embedded System
  - Processor based
    - General processors
    - Micro controllers
    - DSP
  - A subsystem
- Real-Time
  - Not only deliver correct results but when these results are delivered
    - Late results = wrong results

Examples In Your Daily Life

- …wake up …

- …have breakfast …

- …set home safety system …

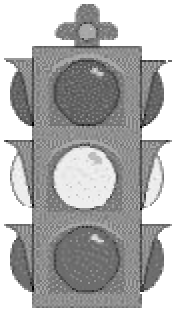
Examples In Your Daily Life (cont')

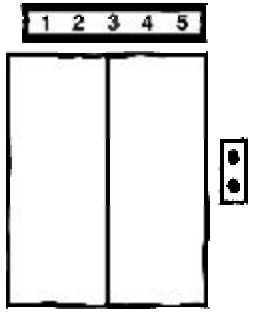
- …get into your car …

  - A late model car can have as many as 65+ processors for engine control, A/C control, cruise control, ABS, audio, etc
  - More than 30% of the cost of a car is now in electronics
  - 90% of all innovations will be based on electronic systems

Examples In Your Daily Life (cont’)

- …on your way to your office…





Examples In Your Daily Life (cont')

- …in your office …





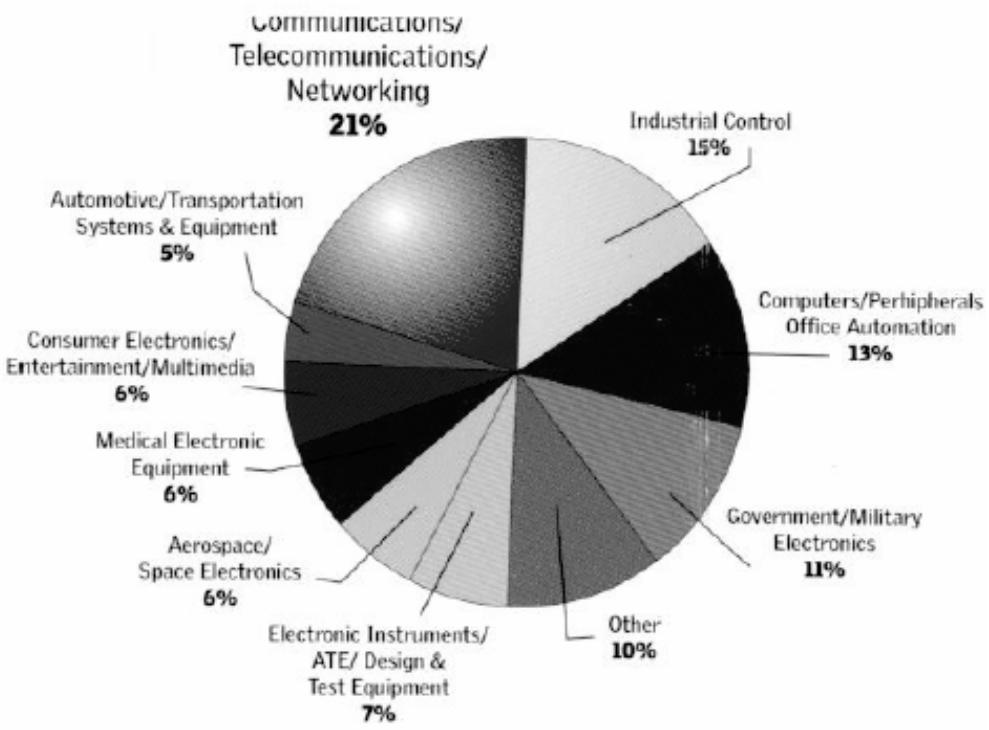


Examples In Your Daily Life (cont')

- …back home …

Several hundred processors can be involved in the course of one day for one person !

Embedded Everywhere



- A hundred billion dollar business
- More than 95% processor shipped today are used in embedded systems
- Market needs for embedded developers increased by 10 folds from 1997-2004

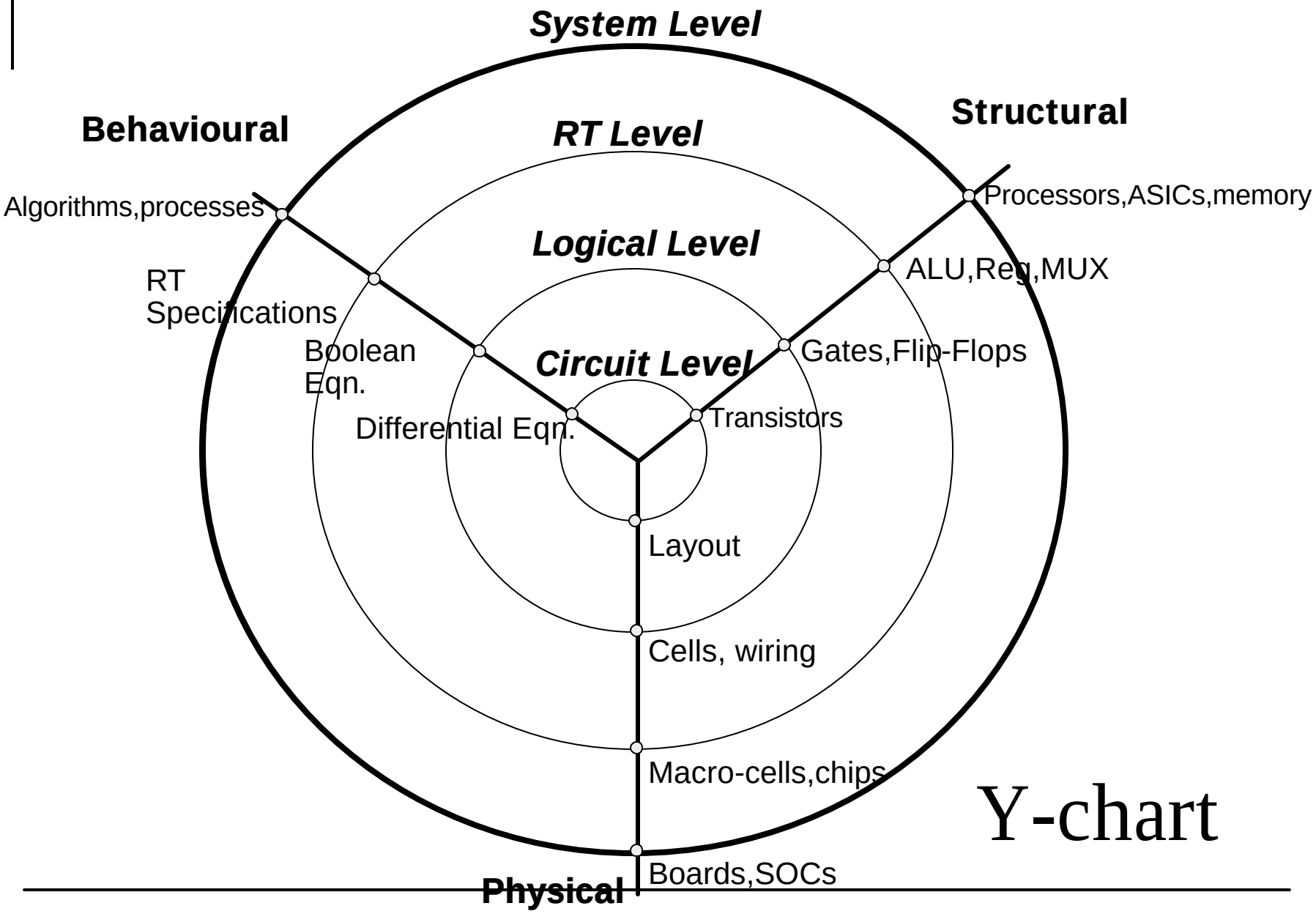
Design Challenges

- Complexity
  - Moore's Law
  - Design constraints
- Diversity
  - Application-specific
  - Design requirements
- Time-to-market

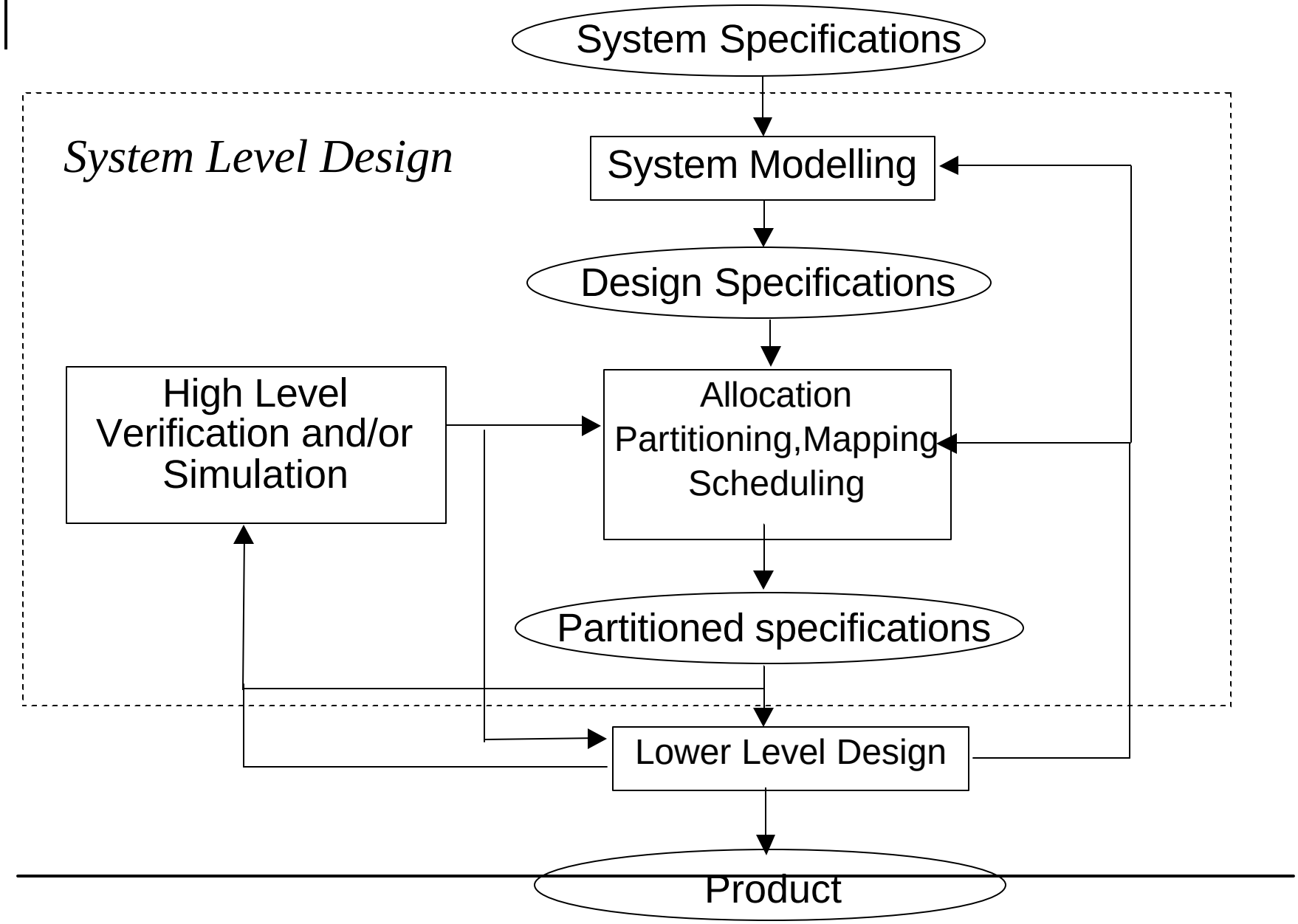
Design Requirements (Constraints)

- For general computing system
  - Performance, performance, performance
- For embedded systems
  - Timing
  - Power consumption
  - Cost sensitivity
  - Size & Weight
  - Safety & Reliability
  - Time-to-market
  - Others: component acquisition, upgrades, compatibility, etc.

How ?


System Level
Behavioural
Structural
RT Level
Algorithms,processes
Processors,ASICs,memory
Logical Level
RT
Specifications
ALU,Reg,MUX
Boolean
Eqn.
Circuit Level
Gates,Flip-Flops
Differential Eqn.
Transistors
Layout
Cells, wiring
Macro-cells,chips
Boards,SOCs
Physical

Y-chart


System Specifications
System Level Design
System Modelling
Design Specifications
High Level Verification and/or Simulation
Allocation Partitioning,Mapping Scheduling
Partitioned specifications
Lower Level Design
Product

Real-Time Scheduling

- The problem
  - Given
    - A set of tasks with timing specifications
    - Limited resources
  - Decide when and where the task to be executed such that
    - Required timing constraints are met
    - System performance is optimized

Why Real-Time Scheduling

- Timing is the essential
- Optimizing system resource usage
- Meeting other design objectives
  - Fault tolerance, Quality-of-Service, etc

Goals

- Feasibility Checking
  - Can a real time task sets be feasible under some given scheduling policy?
- Synthesis
  - For a given applications, what would be a proper scheduling policy?

Scheduling Policies

- Static/Dynamic
- Priority driven/Non-priority driven
- Preemptive/Non-preemptive
- Uniprocessor/Multiprocessor

Real-time Scheduling Research

- Two most famous policies
  - Rate Monotonic Scheduling (RMS)
    - higher the rate, higher the priority
  - Earliest Deadline First (EDF)
    - Closer the deadline, higher the priority
- Most of priority-based scheduling research adopts/extends these two policies
  - Different architecture
  - Different constraints (synchronization, fault tolerance, QoS, low power, etc)

Power Aware Design

- Why
  - Extending battery life
  - System reliability
  - Economic reasons
- How

Power / Energy Consumption

$$E = \int P(t)dt$$

$$P = aC_L V^2 f + I_{leak} V$$

Techniques

- Ranging from circuit level to system level
- The principle ?
  - Reduce t, a, C_L, V, f, I_{leak}
- Examples
  - Number representation
  - Encoding
  - Pipelining
  - Compiler techniques
  - Clock gating
  - Power aware real-time scheduling

Summary

- Real-time embedded systems
- Design challenges and general approach
- To address time and energy issues in real-time embedded systems
  - Real-time scheduling
  - Power aware computing
- Want to know more?
  - CSCE790 (spring) and CSCE818 (fall)