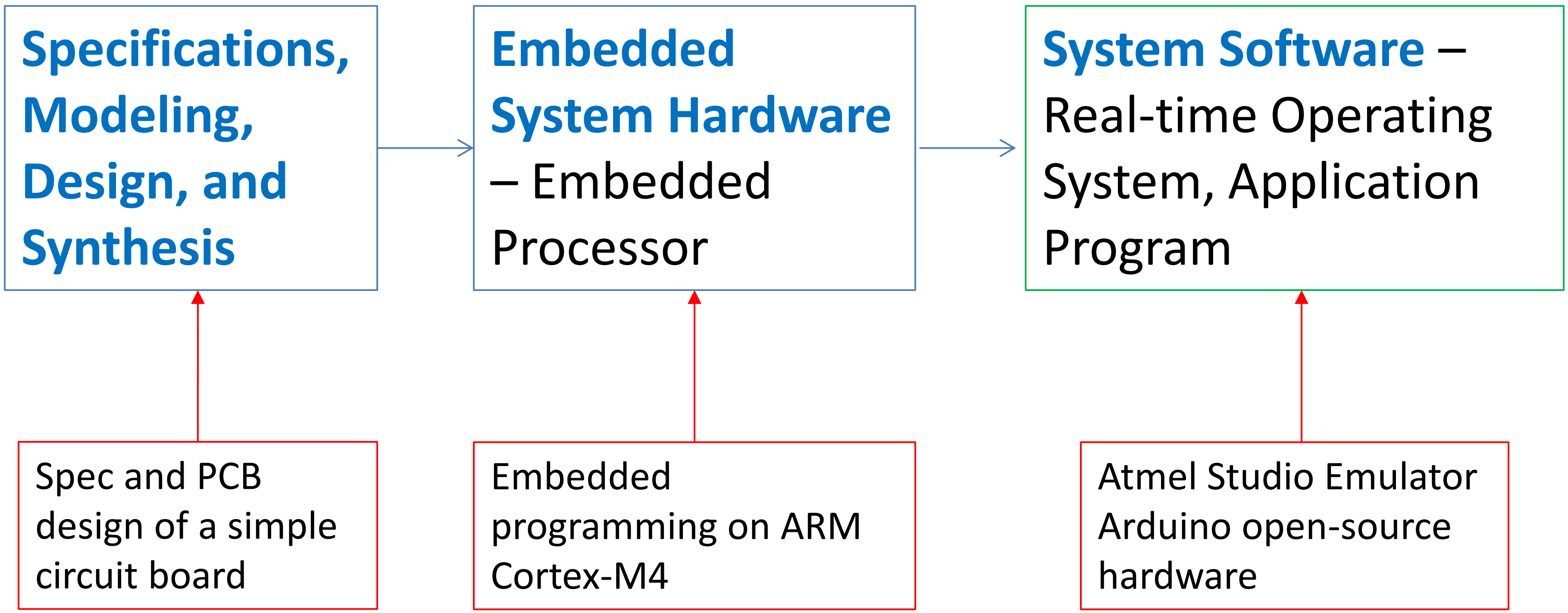
Embedded Systems Design

Chapter 1. Introduction

Dr. Liang Dong
Electrical and Computer Engineering
Baylor University

Embedded System Design - Overview



What is Embedded System

- An **embedded system** is a special computer system designed for specific control functions within a larger system, often with real-time computing constraints. e.g. mp3 player, traffic lights, industrial control unit, GPS, smart phone

What is Embedded System

- By contrast, a general-purpose computer, such as a personal computer (PC), is designed to be flexible and to meet a wide range of end-user needs.

- Embedded systems contain processing cores that are typically either microcontroller units (MCU) or digital signal processors (DSP).

Scope of Embedded Systems

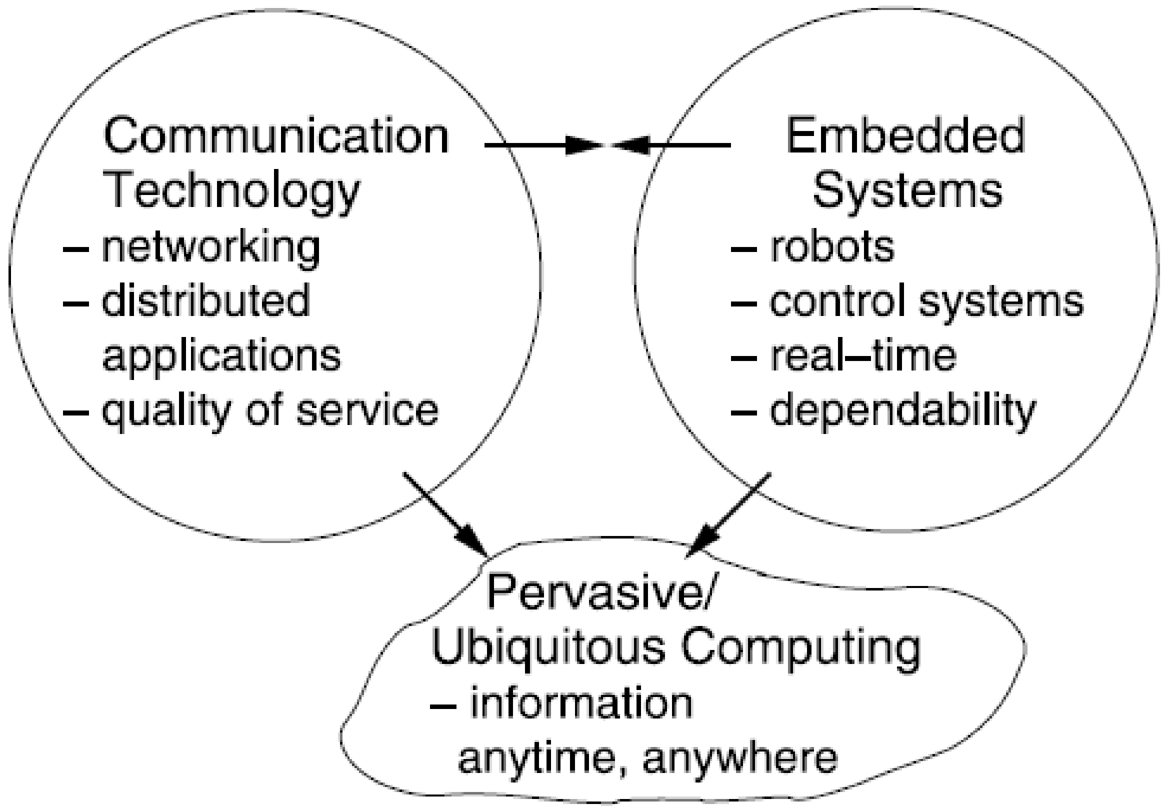
- Miniaturization (the disappearing computer)
- Ubiquitous computing
- Pervasive computing
- Ambient intelligence
- Cyber-physical system
- Internet of Everything (service & things)

- Future Information and Communication Technology (ICT)
  - Embedded systems
  - Communication technologies

Scope of Embedded Systems Design

- Specification techniques,
- hardware components,
- system software,
- application mapping,
- evaluation and validation,
- exemplary optimizations and test methods.

ICT



"Managing time and concurrency in computational systems."

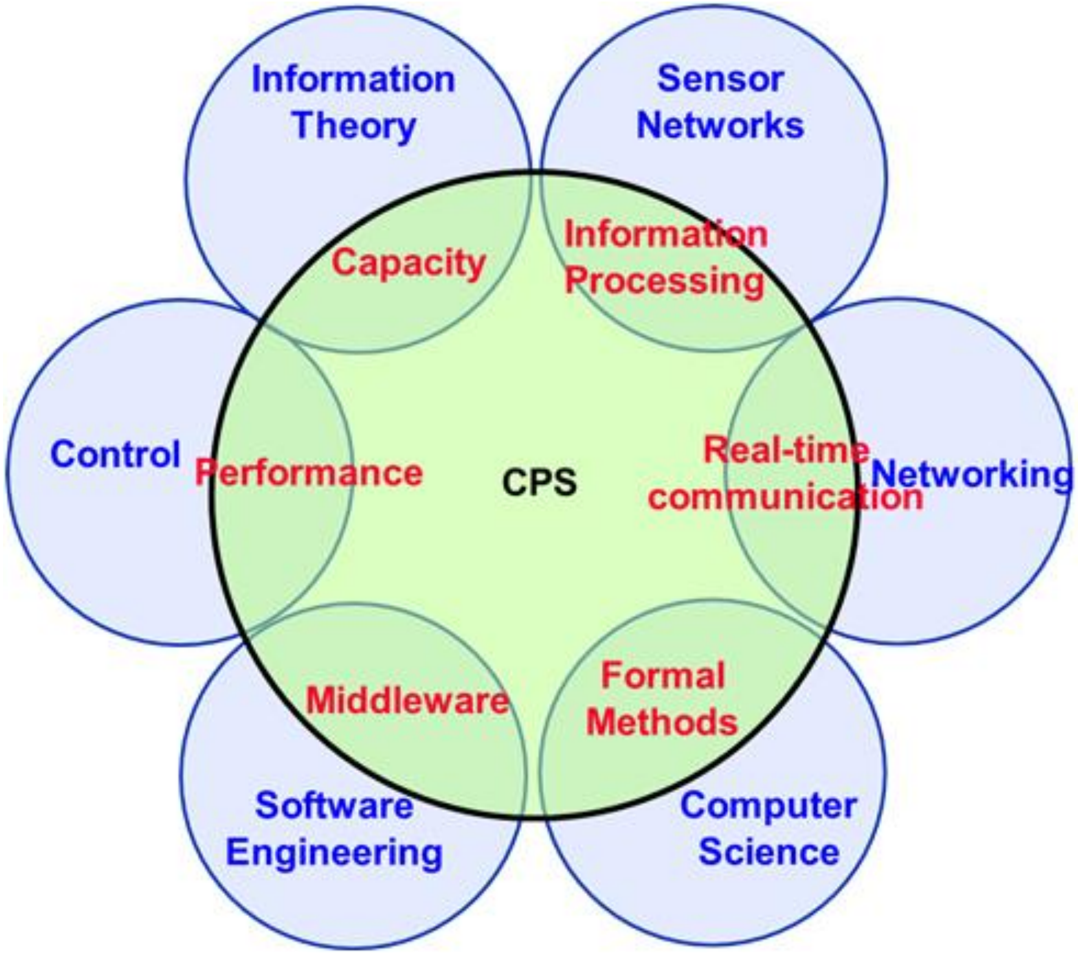
Interacting Methods of Networked System – Cyber-Physical System

Computation, Communication
Information System

Monitoring, Control
Physical System

Cyber-Physical System Networked (large-scale, distributed coordination), adaptable, autonomous, efficient, reliable, safe

Interacting Methods of Networked System – Cyber-Physical System

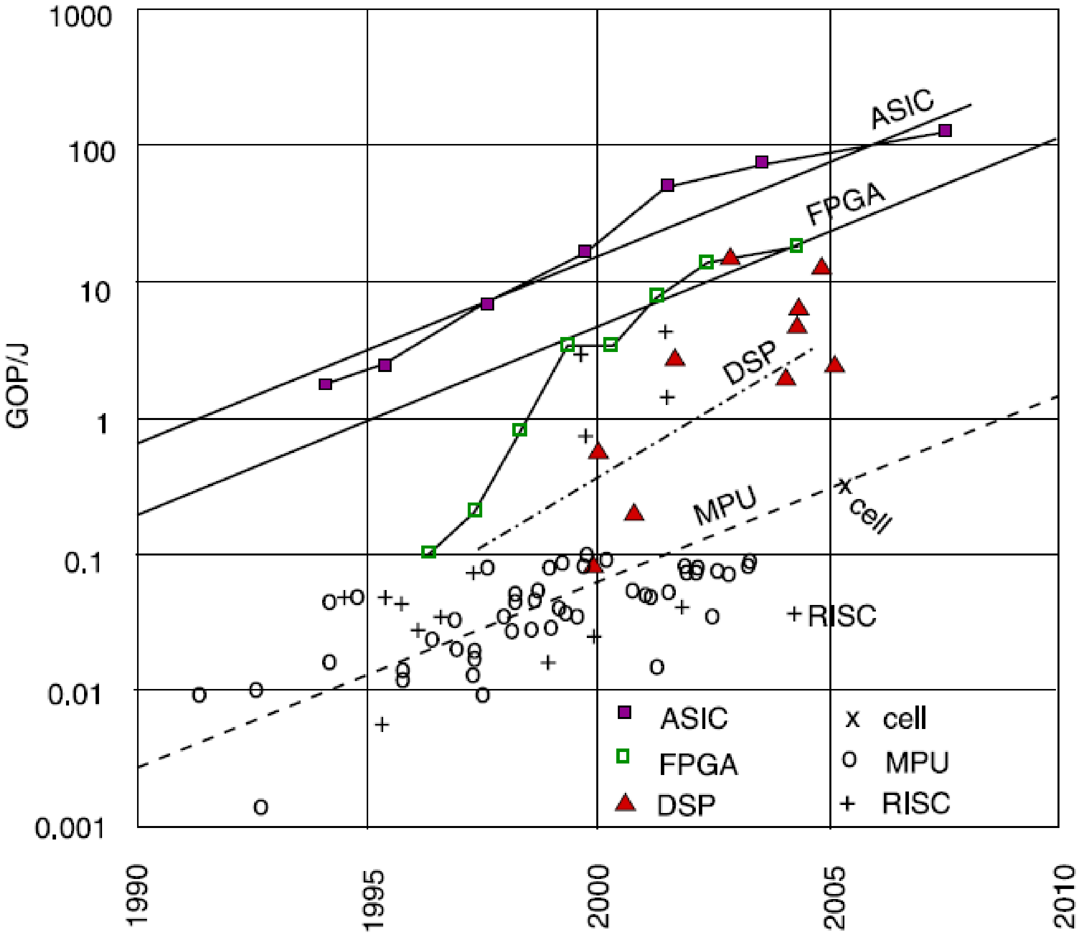


Characteristics

- Dependability
  - Reliability
  - Maintainability
  - Availability
  - Safety
  - Security

Characteristics

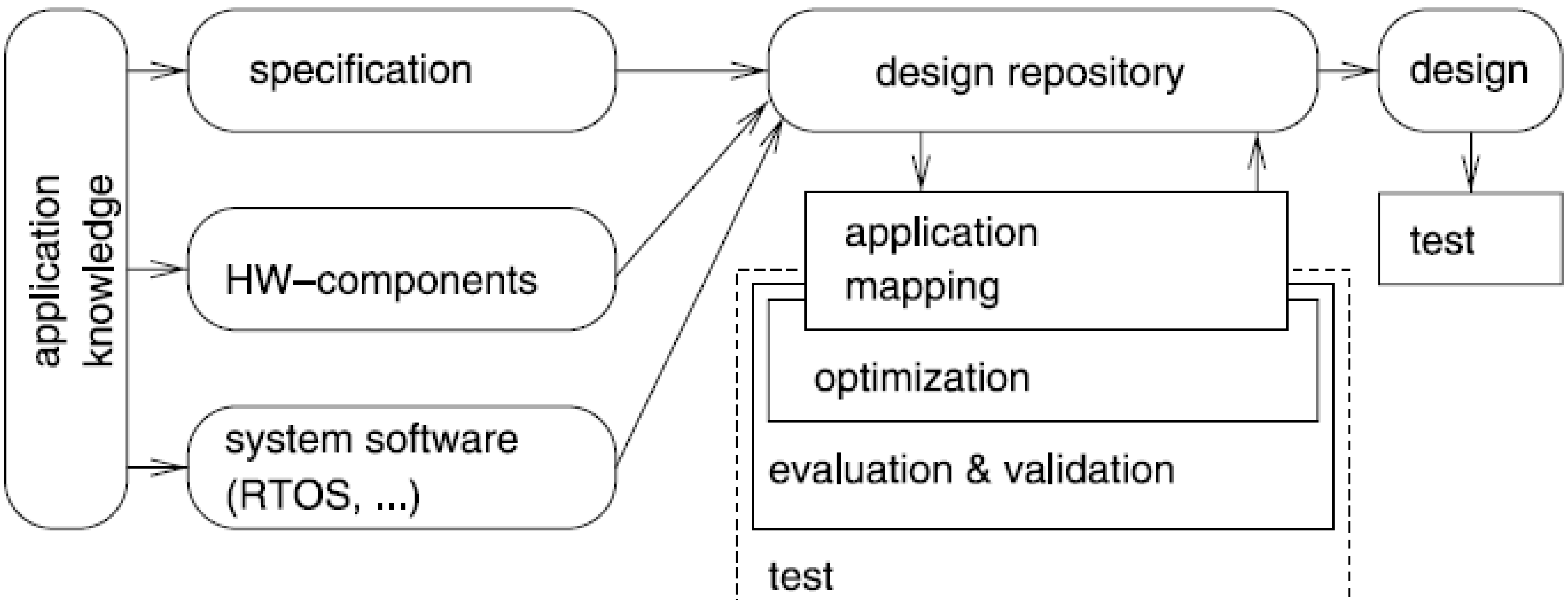
- Efficiency
  - Energy efficiency



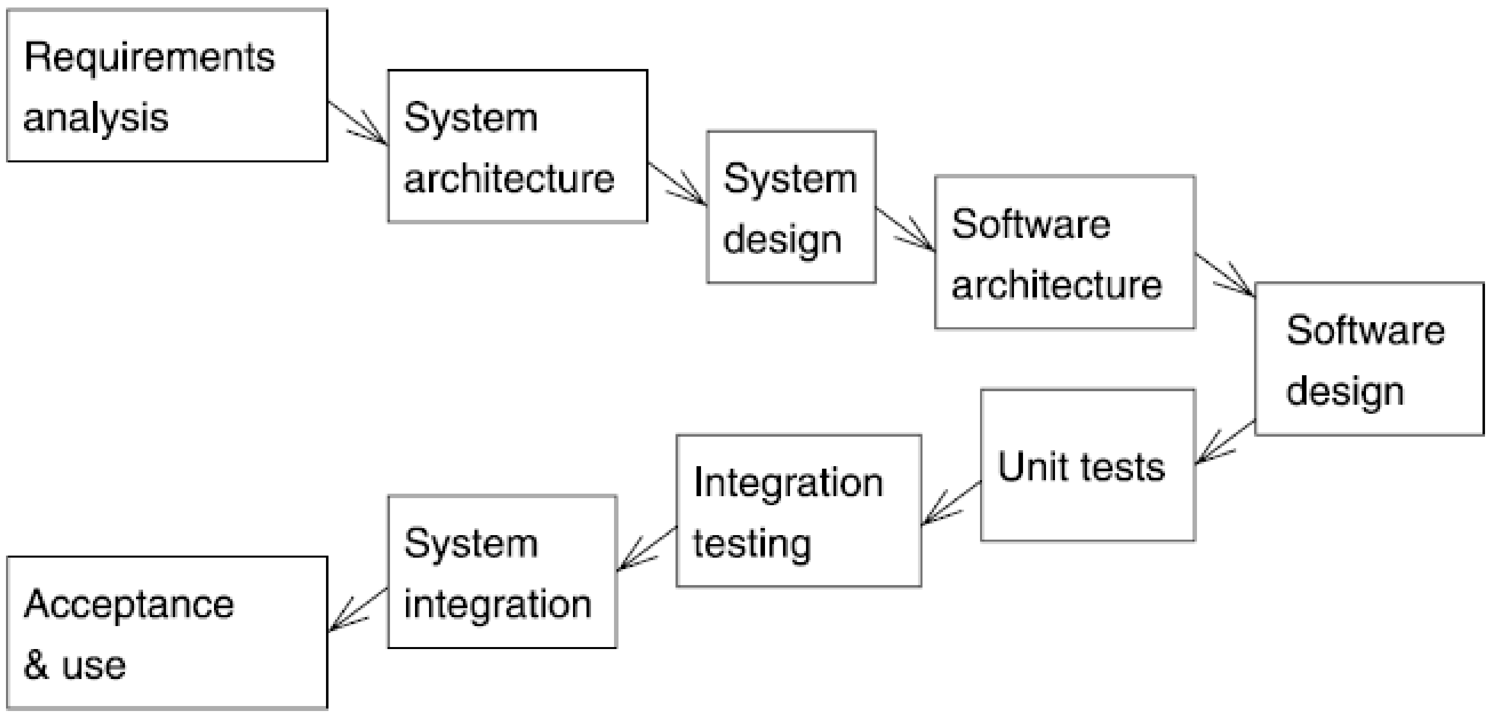
Characteristics

- Efficiency
  - Energy efficiency
  - Run-time efficiency
  - Code size
  - Weight
  - Cost

Design Flows



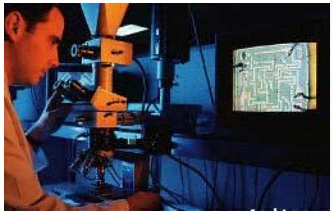
V-Model Design Flow

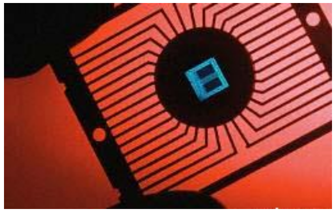


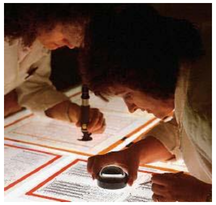
System Design Methodology

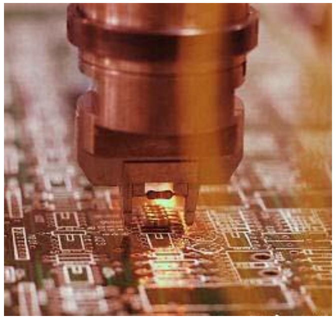
- Design methodologies have evolved together with
  - manufacturing technology
  - design complexity
  - design automation

- Improvements in technology have increased design complexity to the point that designers are no longer capable of making complex designs manually






System Design Methodology

- Design Automation – Computer-aided design (CAD) tools
  - more stringent design rules,
  - parameterize components and
  - minimize component libraries.

- New design abstraction levels and using the same design strategy
  - from the circuit level,
  - to the logic level,
  - to the processor level,
  - and finally to the system level.

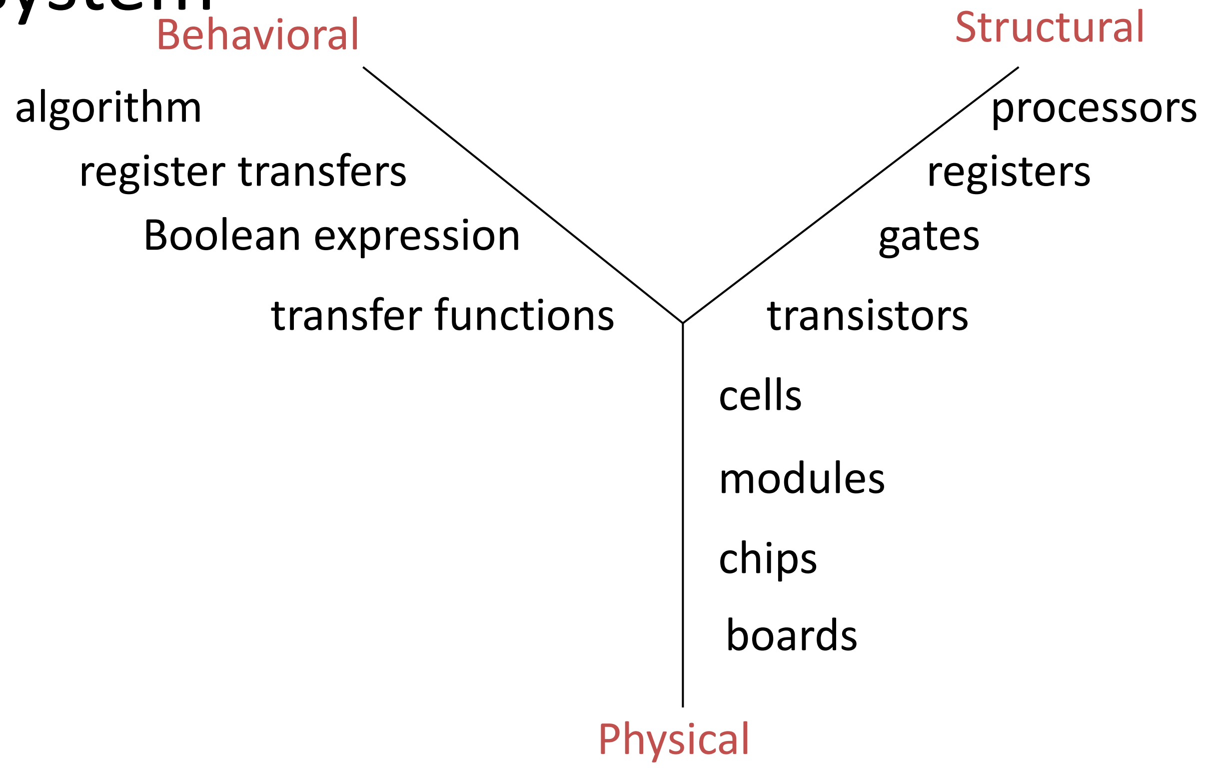
Bottom-Up Methodology

- “Building parts before assembling the whole product.”

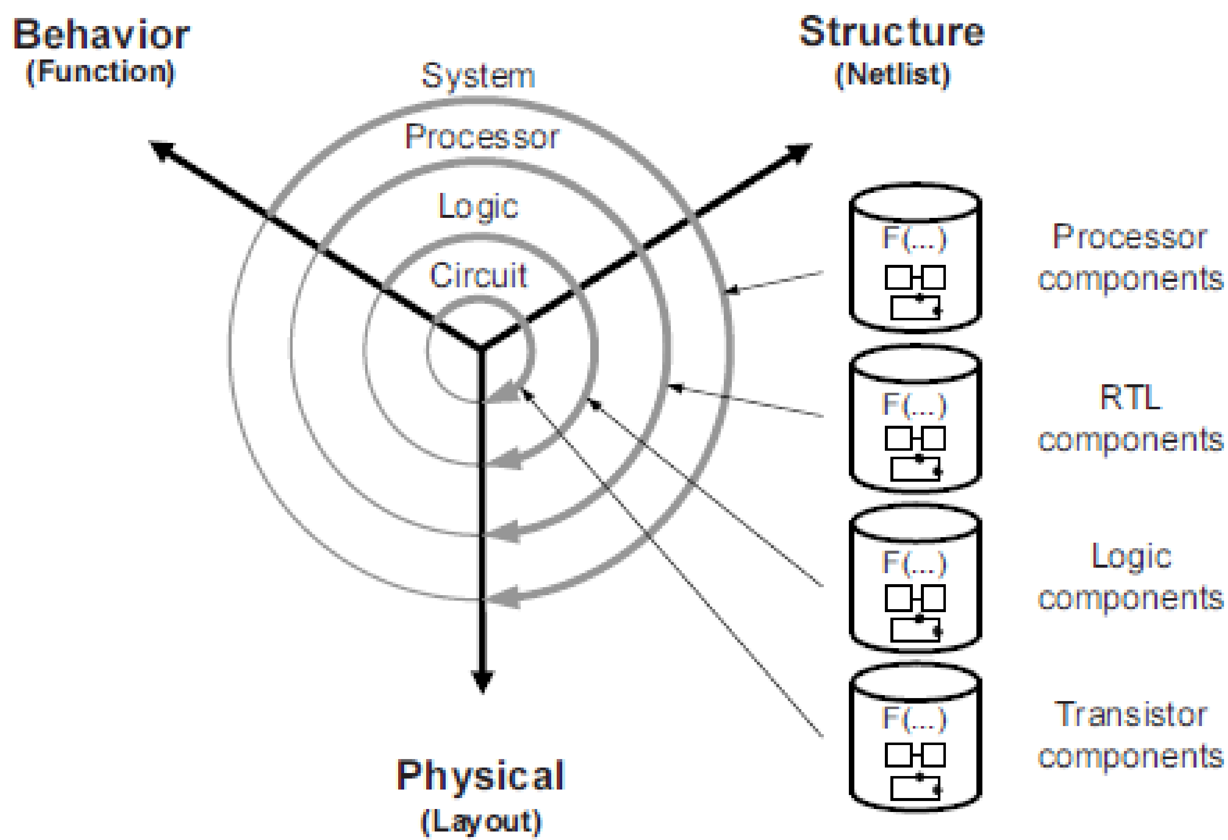
- In a typical bottom-up methodology, designers develop components and then store them in a library for use on the next-higher abstraction level.

Levels of Abstraction

- Y-chart for representations of the different views and levels of abstraction in a digital system



Bottom-Up Methodology



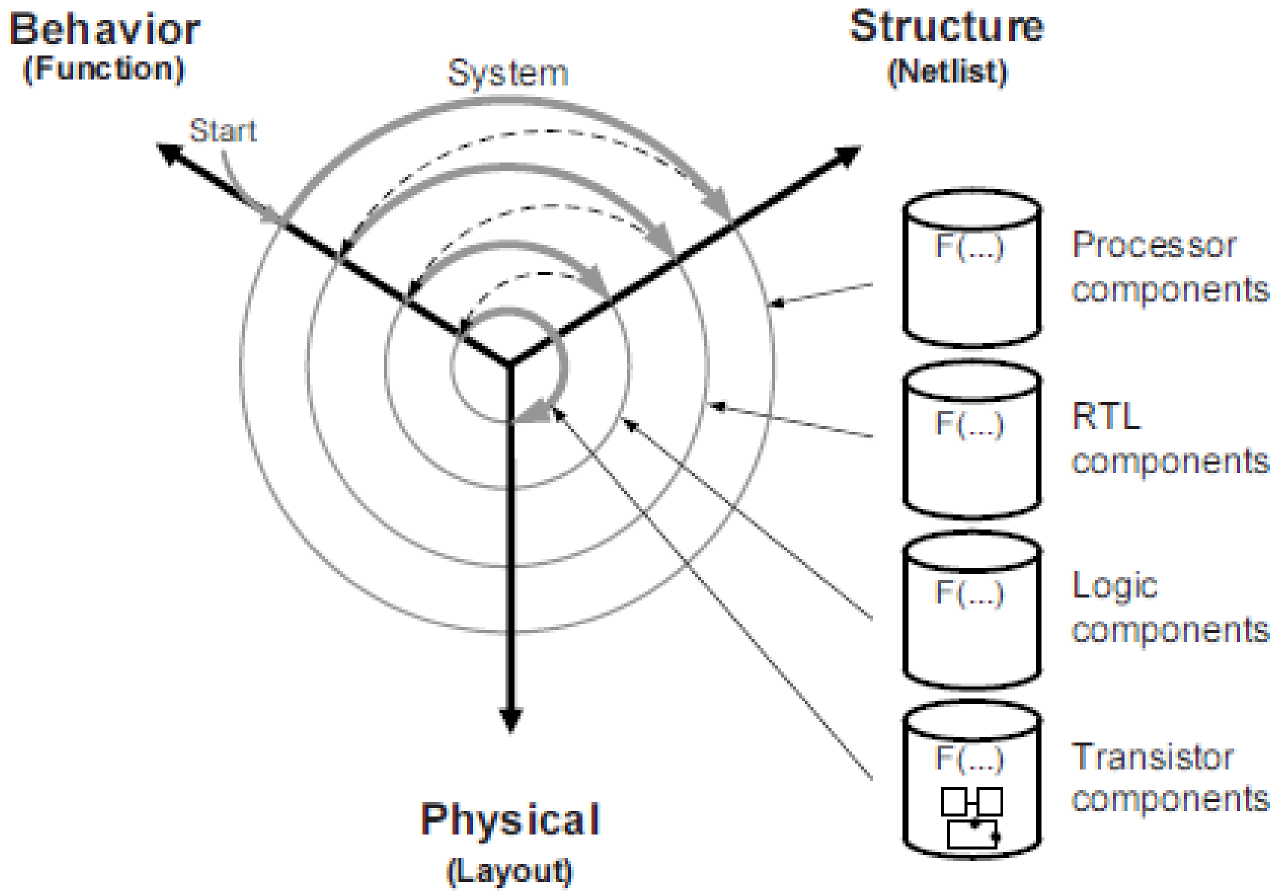
Bottom-Up Methodology

- The advantage of bottom-up methodology is that abstraction levels are clearly separated, each with its own library.
- The disadvantage of this approach is that the libraries must include all possible components with all possible parameters and that these must be optimized for the metrics required by any present and possible future applications.
- This is a very difficult and never-ending task since it is very difficult to anticipate on the lower abstraction level all the needs on the next higher abstraction level.

Top-Down Methodology

- Components that were generated on the System level are decomposed into RTL components with their parameters and required metrics defined.
- Each functional unit, such as the ALU, has all its functions specified, as well as its delay and power requirements. After those are determined, each of the RTL components is further decomposed into logic components or gates.
- Finally, each logic component is broken down into a transistor netlist, in which each transistor layout represents a basic cell.

Top-Down Methodology



Top-Down Methodology

- Such top-down methodologies were in use in design of early computers but today's designs are too complex for such a complete top-down methodology.

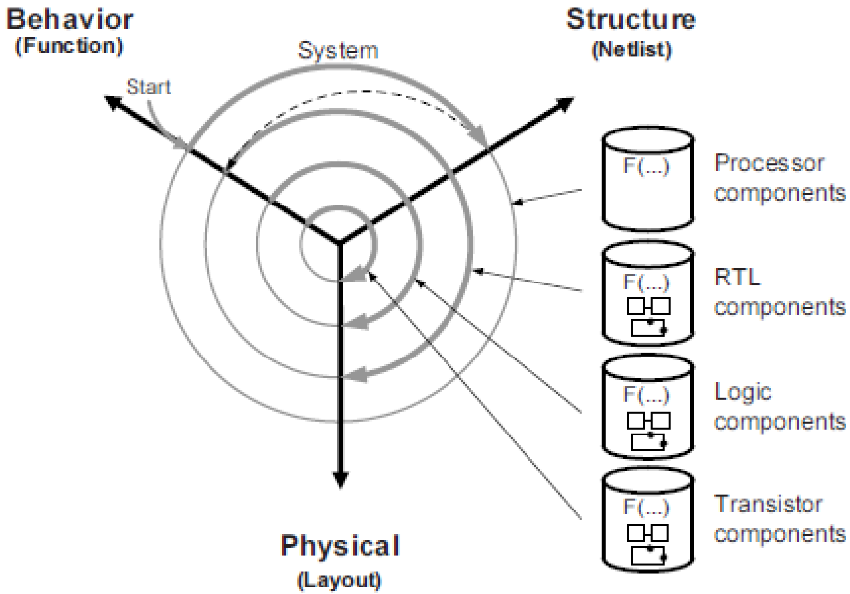
- The system and component metrics are not known until the last step and therefore it is very difficult to optimize the whole design.

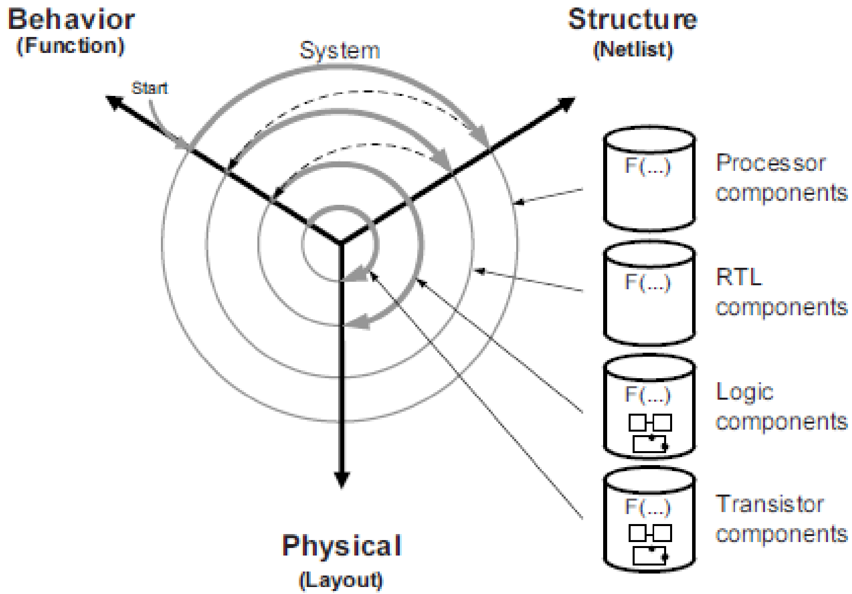
- The design decomposition or synthesis has to be repeated over and over again without designers really knowing whether optimization is going in the right direction.

Meet-In-The-Middle Methodology

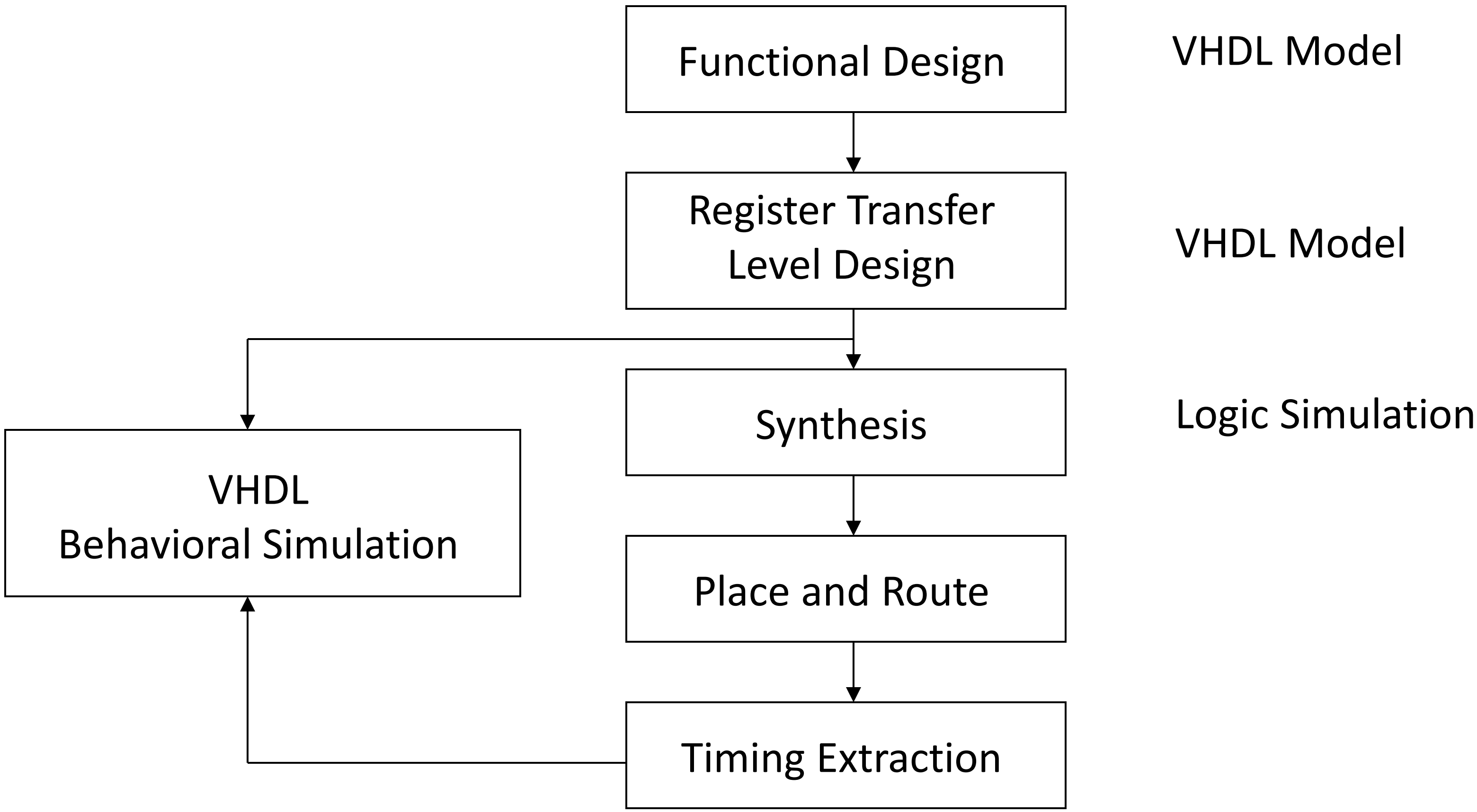
- A meet-in-the-middle methodology allows a designer to take advantage of the tools available for lower level abstractions while also reducing design layouts on higher abstraction levels.
- This is convenient because the design standards and CAD tools on the lower levels of abstractions are well understood and developed, but on the processor and system level they are not.

Meet-In-The-Middle Methodology





Field-Programmable-Gate-Array (FPGA) Methodology



ASIC Design Flow