EECS 222C: System-on-Chip Software Synthesis Lecture 2

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 2: Overview

- The Concept of a Model
- Abstraction Levels
- Top-down Design Methodology

- Invited Guest Lecture
  - System Level Design of JPEG Encoder

- Assignment 1

The Concept of a Model

- What is a Model?
  - Definition: A Model is an Abstraction of Reality.
- Examples of Models
  - Toy car
    - Abstract model of a real car
      - Smaller scale
      - Simpler, many details left out (no motor, no lights, …)
      - Less expensive, less dangerous
      - Less, but sufficiently functional
  - Architectural blueprint of a house
    - 2-dimensional model of a real building (3-dimensional)
      - Smaller, but to scale (floor plan, room sizes, window placement..)
      - Simpler, many details left out (no bricks, just paper)
      - Some features over-emphasized (e.g. layout of pipes, cables)
      - Less expensive, easy to estimate and modify

The Concept of a Model

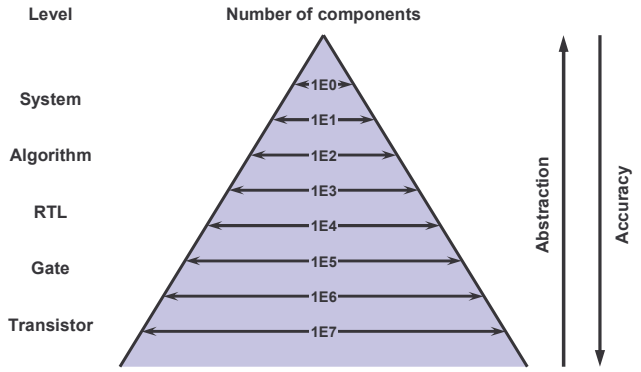
- What is a Model?
  - Definition: A Model is an Abstraction of Reality.
- What is Abstraction?
  - Part of model building
  - Simplification or omission of details
    - Some aspects of reality are simplified or omitted
  - Approach to reduce and factor out details
    - so that one can focus on a few features at a time
- What is Modeling?
  - Model building
  - To make or construct a model

Embedded System Models

- Modeling an Embedded System
  - Decide what feature/property/characteristic
    - is needed (and to what degree)
    - is not needed (can be abstracted away)
- Typical Features in System-on-Chip Models
  - Functionality: important, most often needed (to a varying degree of accuracy)
  - Executability: important, often needed
  - Structure: increasingly needed in later design phases
  - Communication: needed to a varying degree of accuracy
  - Timing: needed to a varying degree of accuracy
  - Power consumption: sometimes needed, sometimes not
  - Temperature: usually not needed

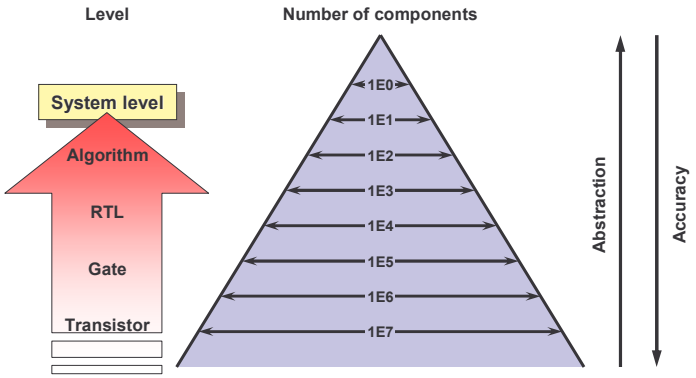
Abstraction Levels

- Embedded system design faces tremendous increase of design complexity



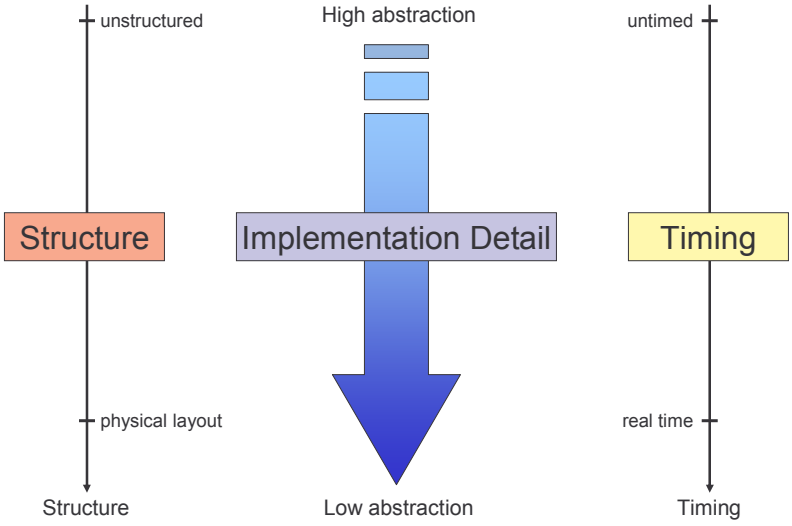
Abstraction Levels

- Embedded system design faces tremendous increase of design complexity
- Move to higher levels of abstraction!



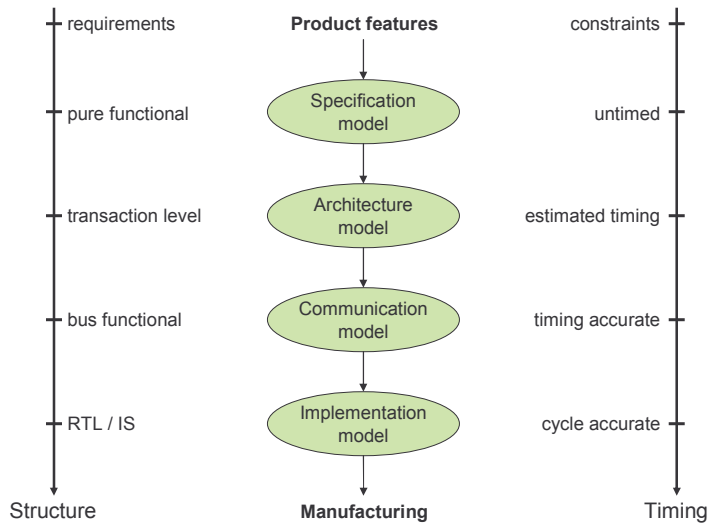
EECS222C: SoC Software Synthesis, Lecture 2 (c) 2008 R. Doemer 7

Abstraction Levels

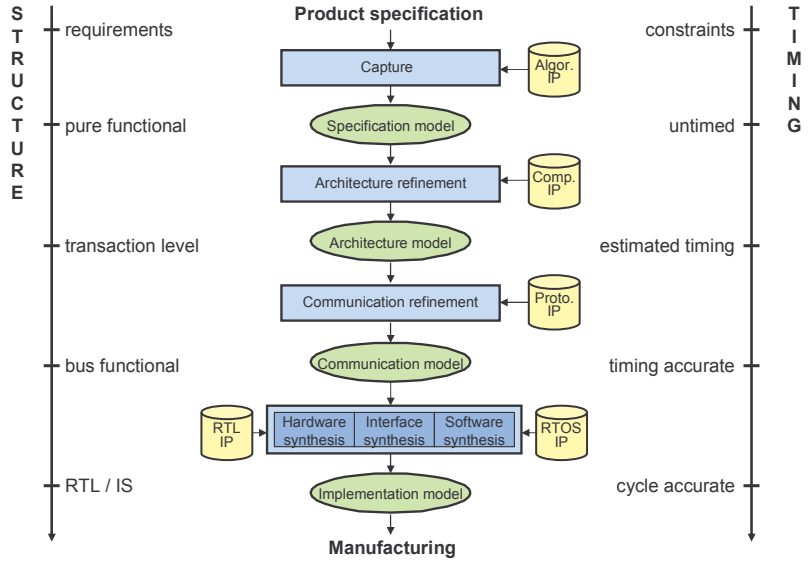


EECS222C: SoC Software Synthesis, Lecture 2 (c) 2008 R. Doemer 8

Top-down Design Methodology



Top-down Design Methodology



Invited Guest Lecture

- ***"System Level Design of JPEG Encoder"***
- Speaker:
  - **Dr. Samar Abdi**
    Center of Embedded Computer Systems
    UC Irvine
- Abstract:
  - The talk presents a top-down system-level design flow with multiple levels of abstraction and discusses recent results of a case-study on a JPEG Encoder application mapped to a Multi-Processor System-on-Chip (MPSoC) platform.
    A demonstration that maps the application onto a FPGA-based prototyping board will conclude the presentation.

Assignment 1

- Login on Server via SSH
  - **`epsilon.eecs.uci.edu`**
  - Account infos have been emailed
- Install JPEG Encoder example
  - **`mkdir eecs222c`**
  - **`cd eecs222c`**
  - **`gtar xvzf /home/doemer/EECS222C_F08/jpegencoder.tar.gz`**
  - **`cd jpegencoder`**
  - **`Make`**
- Become familiar with the application and its structure
  - Browse and read the source files
  - Combine all code into one single ANSI-C file
    - Keep the functional hierarchy, we need it!
  - Draw a block diagram of the functions and their communication