SYSTEM-LEVEL SYNTHESIS:
From Specification to Transaction-Level Models

Daniel D. Gajski
Center for Embedded Computer Systems
University of California, Irvine

http://www.cecs.uci.edu/~ese/
System Design Trends

Past
- Platform
- HW Dev.
- Board
- SW Dev.
- Board + BSP
- App. Dev.
- Prototype

Present
- Platform
- Platform Modeling
- Virtual Platform
- SW Dev.
- Board + BSP
- App. Dev.
- Prototype

Future
- System Dev.
- C/C++
- Platform
- TLM Gen.
- SW Gen.
- HW Gen.
- TLM
- Board + BSP + App
- ASIC/FPGA Tools
- Prototype

Source: S. Abdi
Model Based Synthesis

Specification

System Synthesis

TLMs

SW/HW Synthesis

PCAM
Three Models Needed

Pin / Cycle Accurate Model
Transaction Level Model
Specification Model

7. Application
6. Presentation
5. Session
4. Transport
3. Network
2b. Link + Stream
2a. Media Access Ctrl
2a. Protocol
1. Physical

Spec

TLMs

Address lines
Data lines
Control lines
P/CAM

Source: G Schirner
Model-Based-Synthesis Advantages

- No basic change in design methodology required
  - Design flow follows present design process
- Productivity gain of more than 1000X demonstrated
  - Designers do not write models
- Simple design update: 1-day change
  - No rework for new design decisions
- High error-reduction: Automation + verification
  - Error-prone tasks are automated
- Simplified globally-distributed design
  - Fast exchange of design decisions and easy impact estimates
- Benefit through derivatives designs
  - No need for complete redesign
- Better market penetration through customization
- Shorter Time-to-Market through automation
Embedded Systems Environment

ESE Front – End
System Capture + Platform Development

Timed TLM

ESE Back – End
SW Development + HW Development

C + RTL

Application Tools: Compilers/Debuggers

Commercial Tools: FPGA, ASIC

Decision User Interface (DUI)
- Create
- Select
- Partition
- Map
- Compile
- Replace

Validation User Interface (VUI)
- Compiler
- Debugger
- Stimulate
- Verify
- TIMED
- CYCLE ACCURATE
- Compile
- Check
- Simulate
- Verify

Embedded System Environment Copyright ©2008, CECS
ESE Front End Design Flow

System Definition

Application
Platform

PE/RTOS Models
Timing Estimation
Timed Application

Bus/IF/Mem Models
TLM Generation

SystemC TTLM
SystemC Simulation

Metrics

Design Optimization
Application Model: set of communicating processes
Platform Architecture

Netlist of SW processors, HW, buses and interfaces
System Definition = Platform + Application
Output: SystemC Timed TLM

TLM Generation Technique
- Application code $\rightarrow$ sc_thread
- Processing element $\rightarrow$ sc_module
- OS Model $\rightarrow$ sc_module
- Bus $\rightarrow$ sc_channel
- Memory $\rightarrow$ Array inside sc_module
- Interface $\rightarrow$ FIFO channel+sc_process
Model Accuracy vs. Execution Time

Time and accuracy trade off among different models

- **Board**
  - Accuracy: 100%
  - Execution Time: 2 sec

- **Timed TLM**
  - Accuracy: ~92%
  - Execution Time: 3~4 hrs

- **Func. TLM**
  - Accuracy: ~80%
  - Execution Time: 3~4 hrs

- **ISM**
  - Accuracy: ~80%
  - Execution Time: 3~4 hrs

- **PCAM**
  - Accuracy: ~80%
  - Execution Time: 15~18 hrs

**Abbreviations**
- TLM: Transaction Level Model
- ISM: Instruction Set Model
- PCAM: Pin/Cycle Accurate Model
ESE Back End Prototyping Flow

SystemC TLM

SW Synthesis → C→RTL (Forte/NISC) → Interface Synthesis

OR

SW/RTOS Library

RTL IP Library

Binary → HW RTL → IF RTL

Pin/Cycle Accurate Model (PCAM) Generator

CA Sim. Tools → C/Verilog PCAM → FPGA Tools

Bus Library

Prototype
Cycle-Accurate Software Synthesis

CPU1

Program

EXE

RTOS

HAL

Compile

RTOS/Driver Synthesis

Bus1

HW IP

Interface

Bus2

CPU2

Program

EXE

RTOS

HAL

Compile

RTOS/Driver Synthesis

HAL Interface

Program
Cycle-Accurate Hardware Synthesis

CPU1

Mem

Interface

Bus1

Processes in C

Bus2

Cycle-accurate Synthesis

CPU2

HW IP (RTL)

P3
Cycle-Accurate Interface Synthesis

CPU1

Arbiter

IC

HW IP

Mem

Interface

CPU2

Interface Synthesis
PCAM is downloaded automatically for fast prototyping with FPGAs or simulated using validation tools.
MP3 Design with ESE

- MP3 Decoder (>12K LOC) on Xilinx Multimedia FPGA board
  - Microblaze soft-core with 0/1/2/4 HW components
Design Quality: ESE

- **Area**
  - ESE designs use fewer FPGA slices and more BRAMs than manual HW

- **Performance**
  - ESE designs execute at same speed as manual designs
Development Time: ESE vs. Manual

- **ESE drastically cuts RTL and Board development time**
  - Manual development includes months of RTL coding
  - Models can be developed at Spec level with ESE
  - TLM, RTL and Board models are generated automatically by ESE
Validation Time: ESE vs. Traditional

- ESE cuts validation time from hours to seconds
  - No need to verify RTL models for every design change
  - Designers can perform high speed validation with TLM and board
Technology Summary

• C based application input
  • Supports model based design and legacy reuse
• Automatic functional and timed TLM generation
  • Enables early design validation and reliable estimation
• Automatic SW synthesis
  • Provides modular, verifiable, platform specific SW code
• Automatic interface synthesis
  • Allows rapid implementation of heterogeneous networks
• FPGA and C/HDL export
  • Generates standard input for commercial prototyping and CA validation tools
Business Benefits

- Huge productivity gain
- Easy change and upgrade management
- Model-based design and manufacturing
  - Executable specification for suppliers
  - Easy verification of supplied parts
  - Easy integration, assembly, repair
- New platform exploration and definition possibilities
- In-house early prototyping possibilities
- Early evaluation and implementation of standards
- Huge competitive edge
Acknowledgement

I would like to thank ESE team at CECS that has contributed many ideas through numerous discussions and made ESE tool work:

Samar Abdi
Lochi Yu
Yonghyun Hwang
Hansu Cho
and
Gunar Schirner
Thank You