SoC Design for the New Millennium

Daniel D. Gajski
Center for Embedded Computer Systems
University of California, Irvine
www.cecs.uci.edu/~gajski
Outline

- System gap
- Design flow
- Model algebra
- System environment
- Vision
- Conclusion
Past, Present and Future Design Flow

- **Capture & Simulate**
  - Specs
  - Algorithms
  - Design Logic Physical
  - Manufacturing
  - 1960’s

- **Describe & Synthesize**
  - Specs
  - Algorithms
  - Design Logic Physical
  - Manufacturing
  - 1980’s

- **Specify, Explore & Refine**
  - Executable Spec Algorithms
  - Architecture Network
  - SW/HW Design Logic Physical
  - Manufacturing
  - 2000’s

- System Gap
  - SW?

- Functions:
  - Functionality
  - Algorithms
  - Connectivity
  - Protocols
  - Performance
  - Timing
Missing Semantics: Simulation Dominated Design Flow

- Simulatable but not synthesizable/verifiable

Finite State Machine

Controller

Table Lookup

Memory

(VHDL, Verilog, C, SystemC, ...)

```
case X is
  when X1 =>
  ...
  when X2 =>
```

<table>
<thead>
<tr>
<th>3.415</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.715</td>
</tr>
</tbody>
</table>
Principles of Design Methodology

- **Well defined specification**
  - Complete
  - Just another model
- **Well defined system models**
  - Several possible models
  - Well defined semantics
  - Formal representation
- **Design synthesis**
  - Design decisions => transformations
  - Automatic model generation
- **Model verification**
  - Formally defined transformations
  - Provable equivalence
Modeling as Algebra

- **Algebra** = < {objects}, {operations} >
  Set of all possible expressions [ex: a * (b + c)]
- **Model Algebra** = < {objects}, {compositions} >
  Set of all possible models [ex: ]
- **Model Transformation** \( t \) is a change in objects or composition
- **Model refinement** is an ordered set of transformations
- **Methodology** is a sequence of models and corresponding refinements
Model Definition

- **Model Algebra** = < {objects}, {composition rules} >
- **Objects**
  - Behaviors (representing tasks / computations / functions)
  - Channels (representing communication between behaviors)
- **Composition rules**
  - Sequential, parallel, pipelined, FSM
  - Behavior composition creates hierarchy
  - Behavior composition also creates execution order
- **Relations**
  - Relations define a specific model \([a^*((b+c)*(b-c)))]\)
  - Relations define the model structure (composition and connectivity)
  - Relations \(\sim\) \{objects\} x \{compositions\}
Model Transformations
(Rearrange and Replace)

- **Rearrange object composition**
  - To distribute computation over components
- **Replace objects**
  - Import library components
- **Add / Remove synchronization**
  - To correctly transform a sequential composition to parallel and vice-versa
- **Decompose abstract data structures**
  - To implement data transaction over a bus
- **Other transformations …**

\[
a*(b+c) = a*b + a*c
\]

Distributivity of multiplication over addition

analogous to……

Distribution of behaviors (tasks) over components
Model Refinement

- **Definition**
  - Ordered set of transformations $< t_m, \ldots, t_2, t_1 >$ is a refinement
    - $model\ B = t_m( \ldots ( t_2( t_1( model\ A ) ) ) \ldots )$

- **Derives a more detailed model from an abstract one**
  - Specific sequence for each model refinement
  - Not all sequences are relevant

- **Automatic model generation**
  - Each transformation can be automated
  - Each refined model can be automatically generated

- **Equivalence verification**
  - Each transformation maintains functional equivalence
  - Refinement is thus correct by construction
Refinement based Methodology

- Refinement based system level methodology
  - Methodology := < {models}, {refinements} >
- Each transformation is uniquely defined
- Each transformation can be automatic
- No need for designer modeling
- No need for modeling languages
- Designers only make design decisions
System Synthesis through Refinement

- **Set of models**
- **Each design decisions => one model transformation**
  - Select components / connections/ IPs
  - Map behaviors / channels
  - Schedule behaviors/channels
  - Add new objects (IF, IC, Arbiters, …)
  - Synchronize to preserve equivalence

- **Design is a sequence of decisions**
- **Refinement is a sequence of model transformations**
- **Synthesis = design + refinement**
- **Challenge: define the design and refinement sequences**
System Verification through Refinement

- **Set of models**
- **Transformations preserve equivalence**
  - Same partial order of tasks
  - Same input/output data for each task
  - Same partial order of data transactions
  - Equivalent replacements
- **All refined models will be “equivalent” to input model**
  - Still need to verify
    - First model
    - Correctness of replacements
Y Chart

Behavior (Function)

Structure (Design)

System

RTL/CA

Logic

Transistor

Physical (Layout)
RTL Functional Model

- **Finite State Machine with Data (FSMD)**
  - Combined model for control and computation
    - FSMD = FSM + DFG
  - Implementation: controller plus datapath

![Diagram showing FSMD model (SFSMD model)]
RTL Structural Model

- Multi-pipelined data path
- Hardwired or programmable controller
- RISC or NISC style

Diagram showing control inputs, bus, data path, and various components like ALU, register, cache, and latch.
RTL Synthesis

Variable binding
Rescheduling
Allocation

Operation Binding
Bus Binding
FSM Synthesis

FSMD model

RTL/NISC structural model

Copyright ©2003 Daniel D. Gajski
System Functional Model

- **Program State Machine**
  - States described by procedures in a programming language
- **Example:** SpecC! (SystemC subset?!)
System Structural Model

- Arbitrary components (Proc., IP, Memory)
- Arbitrary connectivity (buses, protocols)
System Synthesis

- Behavior Binding
- Variable Binding
- Allocation
- Profiling
- Channel Binding
- IF Synthesis
- Refinement

System architecture:
- Memory
- Processor
- IP Comp.
- Interface
- Bus
- Custom HW
- Memory

PSM model:
- Proc
- Proc
- Proc
- Proc

Copyright ©2003 Daniel D. Gajski
System Synthesis (continued)

RTL/IS Implementation + results

HCFSMD model

RTL

MoC

RTL/IS Implementation + results

Control

Pipeline

IF FSM

Datapath

IF FSM

RAM

Memory

Control

IF FSM

State

State

State

State

State

State

IF FSM

Control

Datapath

IF FSM

Control

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

State

S
System Level Models

- Models based on time granularity of computation and communication
- A system level design methodology is a path from model A to F

A: “Specification Model”

Objects
- Computation
- Behaviors
- Communication
- Variables

Composition
- Hierarchy
- Order
- Sequential
- Parallel
- Piped
- States
- Transitions
- TI, TOC
- Synchronization
- Notify/Wait
B: “Component-Assembly Model”

Objects
- Computation
  - Proc
  - IPs
  - Memories
- Communication
  - Variable channels

Composition
- Hierarchy
- Order
  - Sequential
  - Parallel
  - Piped
  - States
- Transitions
  - TI, TOC
- Synchronization
  - Notify/Wait
C: “Bus-Arbitration Model”

Objects

- Computation
  - Proc
  - IPs (Arbiters)
  - Memories
- Communication
  - Abstract bus channels

Composition

- Hierarchy
- Order
  - Sequential
  - Parallel
  - Piped
  - States
- Transitions
  - TI, TOC
- Synchronization
  - Notify/Wait
D: “Bus-Functional Model”

Objects
- Computation
  - Proc
  - IPs (Arbiters)
  - Memories
- Communication
  - Protocol bus channels

Composition
- Hierarchy
- Order
  - Sequential
  - Parallel
  - Piped
  - States
- Transitions
  - TI, TOC
- Synchronization
  - Notify/Wait

Communication
- Cycle-timed
- Approximate-timed
- Un-timed

Computation
- Cycle-timed
- Approximate-timed
- Un-timed
E: “Cycle-Accurate Computation Model”

Objects
- Computation
  - Proc
  - IPs ( Arbiters )
  - Memories
  - Wrappers
- Communication
  - Abstract bus channels

Composition
- Hierarchy
- Order
  - Sequential
  - Parallel
  - Piped
  - States
- Transitions
  - TI, TOC
- Synchronization
  - Notify/Wait
F: “Implementation Model”

Objects
- Computation
  - Proc
- IPs (Arbiters)
- Memories
- Communication
  - Buses (wires)

Composition
- Hierarchy
- Order
  - Sequential
  - Parallel
  - Piped
  - States
- Transitions
  - TI, TOC
- Synchronization
  - Notify/Wait
SoC Environment

Refinement
User Interface (RUI)

Algorithm selection
System structure
Spec. optimization
Allocation
Beh. partitioning
Scheduling / RTOS
Protocol selection
Channel partitioning
Arbitration
Cycle scheduling
Protocol scheduling
SW/HW synthesis

Profiling
Comp. / IP attributes
Arch. synthesis
Comm. synthesis
Impl. synthesis
RTL/RTOS attributes

Profiling weights
Compl. / IP models
Protocol models
RTL/RTOS models

Profiling data
Design decisions
Estimation results
Estimation results
Estimation results

Specification model
Architecture model
Communication model
CA model

Lang. Translators

Validation
User Interface (VUI)

Capture
Simulate
Verify
Test
Simulate
Verify
Test
Simulate
Verify

Compile

Estimation

Impl. synthesis

CA refinement

Copyright ©2003 Daniel D. Gajski
Vocoder Results

• Experiment on GSM Vocoder design (10K lines of code)

Simulation Speed

Code Size

Refinement Effort

<table>
<thead>
<tr>
<th>Refinement Type</th>
<th>Modified lines</th>
<th>Manual</th>
<th>Automated User / Refine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spec→Arch</td>
<td>3,275</td>
<td>3~4 mons.</td>
<td>15 mins / &lt;1 min</td>
</tr>
<tr>
<td>Arch→Comm</td>
<td>914</td>
<td>1~2 mons.</td>
<td>5 mins/ &lt;0.5 min</td>
</tr>
<tr>
<td>Comm→Impl</td>
<td>6,146</td>
<td>5~6 mons</td>
<td>30 mins / &lt;2 mins</td>
</tr>
<tr>
<td>Total</td>
<td>10,355</td>
<td>9~12 mons</td>
<td>50 mins / &lt;4 mins</td>
</tr>
</tbody>
</table>

Conclusion

• Productivity gain >2,000X for industrial strength designs
• Compare 9-12 months (manual refinement) vs. 50+4 minutes (user decisions + automatic refinement)
• Enables extensive design exploration (60/day)
JPEG Results

- **Simulation Speed & Code size**

  ![Simulation Speed & Code Size Graph](image)

- **Refinement Effort**

<table>
<thead>
<tr>
<th>Refinement Effort</th>
<th>Modified lines</th>
<th>Manual</th>
<th>Automated User / Refine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spec→Arch</td>
<td>751</td>
<td>1–2 mons.</td>
<td>5 mins / &lt;0.5 min</td>
</tr>
<tr>
<td>Arch→Comm</td>
<td>492</td>
<td>~1 mons.</td>
<td>3 mins / &lt;0.5 min</td>
</tr>
<tr>
<td>Comm→Impl</td>
<td>1,278</td>
<td>3–4 mons</td>
<td>20 mins / &lt;1 mins</td>
</tr>
<tr>
<td>Total</td>
<td>2,521</td>
<td>5–7 mons</td>
<td>28 mins &lt;2 mins</td>
</tr>
</tbody>
</table>

- **Compare 5-7months (manual refinement) vs. 28 minutes (user decisions) + 2 minutes (automatic refinement)**
Conclusion

- New design flow paradigm based on SER methodology
- Semantics before syntax (language does not matter)
- Model algebra before semantics
- System verification, synthesis, modeling simplified
- SW = HW = computation is the inevitable truth
- SoC environment gives 1000X productivity gain
- Challenge: Define models, refinements, methodology