CprE 588  
Embedded Computer Systems

Prof. Joseph Zambreno  
Department of Electrical and Computer Engineering  
Iowa State University

Lecture #8 – Architectural Specialization
Outline

• Motivation
• Related Work
• Design Flow
• Basic Concepts
• Multi-Metrics
• Experimental Results
• Conclusion

Motivation

SoC Complexity ↑

Time-to-market ↓

Higher levels of abstraction

Modeling level n

Estimation

Exploration

Verification n+1

Refinement
System Level Estimation

- Fast
- Accurate ➔ Fidelity
- Different abstraction levels
- Wide range of metrics
- Wide variety of target implementation
Related Work

- Static analysis-based approaches
  - **Examples**
    - WCET (Y. Li), scheduling analysis (G. Buttazzo)
    - Memory size estimation (Y. Zhao)
  - **Limitations**
    - Time-consuming, manual interference

- Dynamic simulation-based approaches
  - **Examples**
    - Profiling tools (GNU profiler)
    - Instruction-set simulators
    - Multi-processor, multi-level co-simulation (P. Gerin)
    - Trace-based simulation (K. Lahir, P. Lieverse)
  - **Limitations**
    - A simulation is required for each design alternative
    - Target/host machine-dependent characteristics
    - Operation-related data
Design Flow

Refinement | Estimation | GUI

Profiling

Spec model

Simulation

Profiling

Spec characteristics
Design Flow (cont.)

Refinement
- Spec model
  - Simulation
    - Profiling
      - Spec characteristics
        - Design decision
          - Implt Characteristics

Profiling
- Refining
  - Back annotation

Retargeting
- Retargeting
Design Flow (cont.)

Refinement:
- Spec model
- Back annotation
- Refined model
- Refining
- Instrumenting
- Refined model
- Refinement

Estimation:
- Profiling
- Simulation/estimation
- Impl characteristics
- Impl Estimates

GUI:
- Spec characteristics
- Design decision
Explore and Trim

Exploration Space

Profiling

Implt independent simulation

One-time retargeting

Implt dependent simulation/estimation

Profiling stage

Retargeting stage

Simulation-estimation stage

Design Time
Profiling

- Instrumentation-based profiling
  - $B_b$: The execution counts of basic block $b$
  - $C_{b,i,d}$: No. of computed characteristics for item type $i$ and data type $d$ in the block $b$
  - Data type $i$: float, int, ..
  - Item type $d$: metric dependant

- $R_{i,d} = \sum_b C_{b,i,d} B_b$
- $R = \sum_i \sum_d R_{i,d}$

```c
int b, c;
if (a = 0) {
    b++;
}
else {
    b++; c++;
}
```

- $R_{++, int} = \sum_i [ B_i * C_{i,++, int} ]$
  - $= 1 \times 1 + 3 \times 2$
  - $= 7$

$$B_1 = 1$$
$$C_{1,++, int} = 1$$
$$B_3 = 3$$
$$C_{3,++, int} = 2$$
Retargeting

- Impl. characteristics
  - \( R_{i,d} \): Spec. characteristics
  - \( W_{i,d} \): weights of components which the entity mapped to
    - Manual
    - Simulation

- \( E = \sum_i \sum_d (R_{i,d} \times W_{i,d}) \)

- Time complexity: \( O(n) \)
Challenges

- Separating dynamic and static analysis
Challenges (cont.)

- Separating dynamic and static analysis
- Supporting complex models
  - Hierarchy
  - Recursion
Challenges (cont.)

- Separating dynamic and static analysis
- Supporting complex models
  - Hierarchy
  - Recursion
- Multi-dimensional analysis

Traditional approach
Challenges (cont.)

• Separating dynamic and static analysis
• Supporting complex models
  • Hierarchy
  • Recursion
• Multi-dimensional analysis
  • Multi-entities
    • Behavior, channel, port, variable
Challenges (cont.)

- Separating dynamic and static analysis
- Supporting complex models
  - Hierarchy
  - Recursion
- Multi-dimensional analysis
  - Multi-entities
    - Behavior, channel, port, variable
  - Multi-metrics
    - Operation, traffic, storage
    - Static, dynamic
Challenges (cont.)

- Separating dynamic and static analysis
- Supporting complex models
  - Hierarchy
  - Recursion
- Multi-dimensional analysis
  - Multi-entities
    - Behavior, channel, port, variable
  - Multi-metrics
    - Operation, traffic, storage
    - Static, dynamic
  - Multi-levels
    - Application, transaction, bus-functional
Operation Metrics

- Entities: behavior
- Item types
  - 84 basic types: ‘+’.., ‘=’.., ‘if’..
  - Special operation types: global function
- Specification characteristics

<table>
<thead>
<tr>
<th>Static</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Def.</td>
<td>No. of operations in spec</td>
</tr>
<tr>
<td>Rep.</td>
<td>No. of executed Oper. during simulation</td>
</tr>
<tr>
<td>Code complexity</td>
<td>Computational complexity</td>
</tr>
</tbody>
</table>

- Mapping: behavior → PE
- Implementation characteristics

<table>
<thead>
<tr>
<th>Static</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rep.</td>
<td>Code Size:</td>
</tr>
<tr>
<td></td>
<td>Program memory size(SW)</td>
</tr>
<tr>
<td></td>
<td>Custom hardware controller(HW)</td>
</tr>
<tr>
<td></td>
<td>No. of executed clock cycles:</td>
</tr>
<tr>
<td></td>
<td>Execution time; power consumption</td>
</tr>
</tbody>
</table>
Traffic Metrics

- Entities: port, variable, channel, behavior
- Item types: in, out
- Specification characteristics

<table>
<thead>
<tr>
<th></th>
<th>Static</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Def.</td>
<td>No. of connected ports</td>
<td>No. of P/V/C accessed during simulation</td>
</tr>
</tbody>
</table>

- Mapping:
  - Port $\rightarrow$ PE
  - Variable/channel $\rightarrow$ bus
- Implementation characteristics

<table>
<thead>
<tr>
<th>Bus</th>
<th>Communication delay</th>
<th>Static</th>
<th>Sequential behaviors</th>
</tr>
</thead>
<tbody>
<tr>
<td>PE</td>
<td>Data preparation time</td>
<td>Dyn.</td>
<td>Parallel behaviors</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application level</th>
<th>Traffic between behaviors.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bus functional level</td>
<td>Fan-in/fan-out, bus capacity, traffic over pin</td>
</tr>
</tbody>
</table>
Deriving Traffic Metrics

- Profiling for the hierarchically instantiated behaviors and recursively called functions

- Different abstraction levels

  - Beh.port → Var
  - Beh.port → Chan
  - Chan in Chan
  - Beh.port → Pin
  - Chan.port ←→ Chan
Storage Metrics

- Entity: variable, behavior
- Item types: local, global
- Specification characteristics

<table>
<thead>
<tr>
<th>Static</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Def.</td>
<td>No. of static variables</td>
</tr>
<tr>
<td>Rep.</td>
<td>Static memory requirement</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Mapping:
  - Variable ➔ local/global memory
- Implementation characteristics

<table>
<thead>
<tr>
<th>Static</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rep.</td>
<td>Static memory size</td>
</tr>
</tbody>
</table>
Exp. Result: Vocoder Profiling

Floating-point not required
Dedicated hardware multipliers

 HW acceleration

Table 1: Computational complexity of top-level Vocoder behaviors

<table>
<thead>
<tr>
<th>Behavior</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>LP_Analysis</td>
<td>377.0 MOp</td>
</tr>
<tr>
<td>Open_Loop</td>
<td>337.1 MOp</td>
</tr>
<tr>
<td>Closed_Loop</td>
<td>478.7 Mop</td>
</tr>
<tr>
<td>Codebook</td>
<td>646.4 Mop</td>
</tr>
<tr>
<td>Update</td>
<td>43.6 Mop</td>
</tr>
</tbody>
</table>

Table 2: Codebook operation mix

<table>
<thead>
<tr>
<th>Operation</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x, int)</td>
<td>46.2%</td>
</tr>
<tr>
<td>(+, int)</td>
<td>33.5%</td>
</tr>
<tr>
<td>(-, int)</td>
<td>9.1%</td>
</tr>
<tr>
<td>(/ ,int)</td>
<td>7.1%</td>
</tr>
<tr>
<td>(others,int)</td>
<td>4.1%</td>
</tr>
</tbody>
</table>
8 behaviors $\rightarrow$ 3 PEs

Total $3^8 = 6561$ design alternatives

Evaluation time: 3:15 hour
- 1 simulation (2.23s)
- 1 profiling (8.41s)
- 6561 retargeting (0.8s)
- 6561 mapping (0.97s)
Exp. Result: JPEG Encoder

- Map 4 behaviors $\rightarrow$ 2 PEs: $2^4$ design alternatives
- Accuracy: 12.5%, fidelity = 100%
Conclusion

- Dynamic profiling + static retargeting
  - Profiling: helps completely comprehend the specification
  - Retargeting: ultra-fast (linear time), enables initial, exhaustive exploration of design space.

- Multi-dimensional analysis
  - Multi-entities
    - Behavior, channel, variable, port
  - Multi-abstraction levels
    - Application level, transaction level, bus-functional level
  - Multi-metrics
    - Operation, traffic, storage
    - Static, dynamic