Methodology

- Past design effort focused on lower levels
- Higher levels lack well-defined methodology and tools
- Paradigm shift to higher levels can increase productivity
- Need methodology and tools for system level
Outline

- Basic concepts in design methodology
- Example
- A design methodology
- A generic synthesis system
- Conceptualization environment
Items a design methodology must specify

- Syntax and semantics of input and output
- Algorithms for transforming input to output
- Components to be used in the design implementation
- Definition and ranges of constraints
- Mechanism for selection of architectural styles
- Control strategies (scenarios or scripts)
Example: Interactive TV processor
Example’s data flow behavior

Digital subsystem
- audio_in
  - StoreAudio
    - audio1[100k][8]
    - GenerateAudio
    - audio_out
- video_in
  - video[500k][8]
  - StoreGenerateVideo
  - OverlayCharacters
    - fonts[128][16][16]
- av_cmd
  - ProcessAVCmd
    - av_cmd[8]
    - StoreAVCmd
- main_cmds
  - ProcessMainCmds
- button
  - ProcessRemoteButtons
  - screen_chars[30][30][8]
Example’s implementation after system design
An example design methodology

Current practice

Functionality specification

Natural language

Functional specification

Manual

System design

Allocation

Partitioning

Reﬁnement

Component implementation

detailed bus protocol

Executable language

Processor

Funct. Spec.

ASIC

Funct. Spec.

ASIC

Funct. Spec.

Memory

Variables

Proposed methodology

C

code

RTL

struct.

RTL

struct.

Memory

mapped address

space
System-design tasks

<table>
<thead>
<tr>
<th>Functional objects</th>
<th>Allocation</th>
<th>Partitioning</th>
<th>Refinement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables</td>
<td>Memories</td>
<td>Variables to memories</td>
<td>Address assignment</td>
</tr>
<tr>
<td>Behaviors</td>
<td>Processors</td>
<td>Behaviors to processors</td>
<td>Interfacing</td>
</tr>
<tr>
<td>Channels</td>
<td>Buses</td>
<td>Channels to buses</td>
<td>Arbitration/protocols</td>
</tr>
</tbody>
</table>
One possible ordering of tasks

1. Functionality specification
   - Specification
   - Memory allocation
   - Variable-to-memory partitioning
   - Bus allocation

2. System design
   - Channel-to-bus partitioning
   - ASIC/processor allocation
   - Behavior-to-ASIC/processor partitioning
   - Interface synthesis
   - Arbiter synthesis

3. Component implementation
   - Implement software
   - Implement hardware
Generic synthesis system requirements

- Completeness
  - All levels of design, all implementation styles

- Extensibility
  - Allow addition of new algorithms and tools

- Controllability
  - User control of tools, design-quality feedback

- Interactivity
  - Partial design, design modification

- Upgradability
  - Evolve to describe-and-synthesize method
A generic synthesis system

Methodology
A generic system-synthesis tool

System behavioral specification

Compiler

Transformer

SR

Allocator

Estimators

Partitioner

Interface & arbitration synthesis

System−module behavioral specifications

To software synthesis

To chip synthesis
A generic chip-synthesis tool
A generic logic-synthesis tool

- State tables
- Boolean expressions
- Timing diagrams
- Memory specifications

- State minimization
- State encoding
- Logic minimization
- Technology mapping
- Timing graph compiler
- Interface synthesis
- Memory synthesis

Physical design
Conceptualization environment

- Tool is only effective if the designer can use it
  - Understandable display of data
  - Highlight design parts that need attention

- Must support many design avenues
A system-synthesis tool interface

- Allocation
- Partition
- Estimates
- Constraints

<table>
<thead>
<tr>
<th>Mappings</th>
<th>Module type</th>
<th>$</th>
<th>Execution time</th>
<th>Area</th>
<th>Pins</th>
<th>Instr</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASIC1</td>
<td>X100</td>
<td>105/100$</td>
<td>100/110</td>
<td>16000/20000</td>
<td>46/60</td>
<td></td>
</tr>
<tr>
<td>CaptureAudio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GenerateAudio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASIC2</td>
<td>X100</td>
<td>100/110</td>
<td>100/110</td>
<td>18000/20000</td>
<td>48/60</td>
<td></td>
</tr>
<tr>
<td>CaptureGenerateVideo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CaptureAVCmd</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory1</td>
<td>V1000</td>
<td>10</td>
<td>100/110</td>
<td>16000/20000</td>
<td>46/60</td>
<td></td>
</tr>
<tr>
<td>audio_array1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>audio_array2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory2</td>
<td>V1000</td>
<td>10</td>
<td>100/110</td>
<td>16000/20000</td>
<td>46/60</td>
<td></td>
</tr>
<tr>
<td>video_array</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Processor1</td>
<td>Y900</td>
<td>25</td>
<td></td>
<td>6000/5000*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProcessRemoteButtons</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ProcessMiscCmds</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cost: 5.43

[View options] [Partition/Allocate] [Refine]
An optional design view

<table>
<thead>
<tr>
<th>Quality metric</th>
<th>Estimate/Constraint</th>
<th>Violation?</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(System)$</td>
<td>105/100</td>
<td></td>
</tr>
<tr>
<td>Execution−time(CaptureAudio)</td>
<td>100/110</td>
<td></td>
</tr>
<tr>
<td>Execution−time(GenerateAudio)</td>
<td>100/110</td>
<td></td>
</tr>
<tr>
<td>Execution−time(CaptureGenerateVideo)</td>
<td>100/110</td>
<td></td>
</tr>
<tr>
<td>Execution−time(CaptureAVCmd)</td>
<td>100/110</td>
<td></td>
</tr>
<tr>
<td>Area(ASIC1)</td>
<td>16000/20000</td>
<td></td>
</tr>
<tr>
<td>Area(ASIC2)</td>
<td>18000/20000</td>
<td></td>
</tr>
<tr>
<td>Pins(ASIC1)</td>
<td>56/60</td>
<td></td>
</tr>
<tr>
<td>Pins(ASIC2)</td>
<td>58/60</td>
<td></td>
</tr>
<tr>
<td>Instr(Processor1)</td>
<td>6000/5000</td>
<td></td>
</tr>
</tbody>
</table>
Summary

- Three-step design methodology
  - Functionality specification
  - System design
  - Component implementation

- Major tasks in system design
  - Allocation
  - Partitioning
  - Refinement

- Generic synthesis tool

- Conceptualization environment
  - Crucial to practical use
Future directions

- Advanced estimation methods
- Formal verification
- Testability
- Frameworks and databases
- Regularity exploiting
- System-level transformations
- Feedback incorporation