Automatic Generation of VHDL Code for Self-Timed Circuits from Simulink® Specification

Maurizio Tranchero
maurizio.tranchero@polito.it

Advisor: Leonardo M. Reyneri
leonardo.reyneri@polito.it

DELEN - Ph.D. Students Research Seminars
2007-07-17, Tuesday
Summary

- Embedded Systems and their design
- High-level design tool for ESs
- Asynchronous circuits and their advantages
- Issues in asynchronous circuits
- Results
- References
Embedded Systems (ESs)

- Are systems in which electronic is used for increase their performances
- Mixed-sigals (analog and digital)
- Tightly interacting to other parts
- More and more complex
Challenges in ESs Design

- Complexity
- Time-to-market constraints
- Portable device (cameras, PDA, ...)
  - Need for low power
- Unwanted interaction with other parts
  - Need for low electromagnetic emissions

High-Level Design Tools

Asynchronous Circuits
Available High-Level Design Tools

- Text-based
  - System C (mainly for ASIC)
  - Precision C (mainly for ASIC)

- Simulink-based
  - DSP-Build (only for Altera® FPGA)
  - System Generator (only for Xilinx® FPGA)
  - CodeSimulink (multi-platform, HW-SW)

- Methodologies Dealing with Asynchronous
  - Handshake Solutions (mainly for ASIC)
  - De-synchronization (no commercial tool available)

- None of them combine high-level design and asynchronous circuits
CodeSimulink Environment

- Based on Simulink®
  - Dataflow specifications
- Uses Simulink® as high-level simulation tool
- Simulates behavior of actual implementation (i.e., analog, digital, software)
- Generates
  - VHDL for digital systems
  - C for software
  - SPICE netlist for analog
Dataflow Specification

- Hw-Sw design methodology for parallel computation
- Simulink uses dataflow specification for its models
- In order to achieve a correct implementation of such models also HW blocks have to follow this specification
- To each datum is associated a validity property in order to avoid data duplication or loss
- Data validity is natural in asynchronous circuits
  - handshaking protocol

Only when both A and B are valid the block can compute Y
Asynchronous Systems

- Asynchronous
  - clock-less
- Clock in digital systems is necessary to
  - Synchronize every module
  - Give a timing sequence
- In asynchronous circuits these functions are performed by local synchronization called **handshake**
Advantages of Clock-less Logic

- Synchronization is performed locally and in an independent way from the other modules
  - Reduced spikes on current consumption
  - Lower EME
- Only parts with valid data are working
  - Power consumption reduction
Block Structure

- To ensure dataflow behavior, each block is composed by three parts:
  - Data-path
  - Protocol
  - Registers
Matching Delays

- Combinational logic has its own characteristic propagation delay
- In order to make the protocol signals match this delay, a special path is inserted
Partitioning Issues

- In asynchronous circuits how to make partition (i.e., division into independent blocks) is not trivial.
- In our case dataflow specification drive to an automatic partitioning.
  - Each elementary block is an independent entity.
Synthesis Issues

- Commercial compiler do not deal with asynchronous circuits
  - Delay chains optimization
  - Protocol Controller optimization
- Problems resolved using special synthesis directives
Placement Issues

- Interconnects delays are important
  - can affect circuit’s behavior!
- Delays after placement have to be used to synthesize correct delay chains
- This leads to an iterative process of synthesis + P&R + TA
Simulation

- Multi-Level Simulation
  - Simulink
  - Pre-Synthesis
  - Post-Synthesis
Interfaces

- Interaction with synchronous modules
- Communication between synchronous and asynchronous leads to Metastability
- Metastability cannot be avoided
  - We can reduce the probability of its occurrence
  - Multiple FF stage
### Preliminary Results

<table>
<thead>
<tr>
<th>Simple Datapath</th>
<th>Synchronous</th>
<th>Asynchronous</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Elements</td>
<td>202</td>
<td>317</td>
<td>+56%</td>
</tr>
<tr>
<td>Registers</td>
<td>99</td>
<td>89</td>
<td>-10%</td>
</tr>
<tr>
<td>Pins</td>
<td>42</td>
<td>37</td>
<td>-12%</td>
</tr>
<tr>
<td>Synthesis time</td>
<td>34’’</td>
<td>47’’</td>
<td>+38%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8th-order IIR filter</th>
<th>Synchronous</th>
<th>Asynchronous</th>
<th>Overhead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic Elements</td>
<td>849</td>
<td>1034</td>
<td>+22%</td>
</tr>
<tr>
<td>Registers</td>
<td>427</td>
<td>621</td>
<td>+45%</td>
</tr>
<tr>
<td>Pins</td>
<td>40</td>
<td>38</td>
<td>-5%</td>
</tr>
<tr>
<td>Synthesis time</td>
<td>1’ 21’’</td>
<td>3’ 13’’</td>
<td>+138%</td>
</tr>
</tbody>
</table>
Future Works

- Synthesis optimization (to reduce overhead)
- Interfaces to synchronous blocks
- Automatic delay chain calculation
- Automatic placement
- Full comparisons between synchronous and asynchronous circuits wrt
  - Area
  - Power
  - EME
References

- Asynchronous Systems Design

- High-Level Design Tools
  - SystemC Community, www.systemc.org/

- Simulink-based high-level design tools:

- Dataflow
The End

- Thank you for attention!

- Any Questions?