1 Introduction

This report summarizes the focus groups held at the 8th International Symposium on System Synthesis, Cannes, Cote D’Azur, France from September 13 to 15, 1995.

Inspired by last years success of the focus groups at the High-level synthesis Symposium in Niagara-on-the-lake, another session of focus groups was held at this years symposium. Last year the focus was on the past, present and future of High-level synthesis. It was clear that for many the future of High-level synthesis lies in the system-level synthesis. The name change of the symposium reflects this spirit. At the beginning of this new era of design automation, a more precise definition of what this actually means and what major problems lie ahead of us needs to be made. The assembly of many of the world-experts in high-level synthesis made this an excellent opportunity to focus on this subject.

The main goal at the symposium was to encourage interaction between all participants. However, various interesting results came out and I thought it to be worthwhile to bundle them in a small report. I hope this report will serve as the minutes for those who were present at the workshop, and will contain some valuable information for those who were not.

2 Organization

The theme of the focus groups was to define the field of system-level synthesis/design. This report will first shortly describe the organization of the focus groups and the questions distributed among them.

All attendees were grouped in nine groups of around ten people. Groups were selected at random from the list of attendees. All groups were giving the same set of questions. However, some groups were given the task to defend a particular statement, while other were asked to come up with counter arguments. This to inject some controversial points for the final discussion.

The focus groups met in three sessions. The first was a brainstorming session aimed at accumulating as many answers as possible to each question. In the second session the group members selected the best answers to each of the questions. In the final general session all results were reported back to the entire group of workshop attendees and the results were accumulated and discussed.

The next section lists all questions and gives a summary of the results. For each reader to draw her/his own conclusions, the remaining sections contain the answers prepared by each of the nine focus groups.

3 Questions

Last year we focused on high-level synthesis itself. The major achievements and the progress of the field in the last decade were discussed along with some future topics [1]. As we are moving more in the system level synthesis arena, the design environment becomes more and more important. Automating synthesis at the system level requires a major interaction with designers and is very heavingly influenced by design styles.

This year’s questions focus more on the external forces which work on system design CAD area and from there on try to define an outlook for system level synthesis.

1. How do designers design at the system level?
   (a) How is the design specified?
   (b) Is automation support really required at this level? (YES, NO)
   (c) How would this increase the designer’s efficiency?

2. Who are the main customers for system-level synthesis?

3. How domain specific is system-level synthesis? (VERY, NOT AT ALL). What are the domains?

4. Is system-level synthesis:
   (a) raising the level of abstraction or
   (b) system integration at the current abstraction level and why?

5. Is system level synthesis too large an area for our community to address?
   - What can be done to stimulate co-operation between research groups?
   - How to create funding in the different continents?
   - Who will pay the bill for research in system-level synthesis?
have existing architectures
- limits design exploration

- users are domain specific
- technology & application

- systems are heterogeneous
- spec is very general

- $\sum$ of blocks
- common abstraction level

<table>
<thead>
<tr>
<th>Domain Specific</th>
<th>Not at All</th>
<th>Very</th>
</tr>
</thead>
<tbody>
<tr>
<td>systems are heterogeneous</td>
<td></td>
<td></td>
</tr>
<tr>
<td>spec is very general</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sum$ of blocks</td>
<td></td>
<td></td>
</tr>
<tr>
<td>common abstraction level</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- what is the role of the CAD industry?

6. what do you think will be the five hottest system level topics for the next ISSS?

Answers to questions 1b, 3 and 4 were forced upon the groups. The odd numbered groups were asked to argue that design automation support on the system level is needed (1b), it is not domain specific (3) and raises the level of abstraction (4). The even numbered groups were asked to take the opposite position.

4 Summary of Results

Tables 1 to 3 summarize the results of the answers to the questions 1b, 3 and 4.

Most groups were convinced that some level of automation of the design process at the system level is required. However, they mainly focused on exploration, verification, validation and simulation tasks (see table 1) and expressed less initial need for System Level Synthesis (SLS). The synthesis tasks at the system level are (still) very diverse and no good more general formalisms are available.

However, for SLS to be feasible, it better not be too domain specific (see table 2). Dollars available in each of the domains may not justify design automation in each of these individual domains. We better search for a common abstraction level, which might very well be possible since systems are really sums of individual building blocks and general languages for specification are emerging. However, the diversity in system characteristics, system design practices and implementation technologies make the critics believe that domain specificity is essential to obtain acceptable quality results.

Interesting were the answers to the question regarding the level of abstraction suitable for SLS (see table 3). Arguments for raising the level of abstraction are the same as have been over years when progressing to logic and high-level synthesis. However, arguments to concentrate on system level synthesis at the current level were all formulated in a negative way. If research in this area can come up with unified formalisms, which will naturally lead to the formulation of optimization problems in SLS, the industry might get ready and the market will grow to a large enough size. No group came up with arguments why rising the level of abstraction would in principle be infeasible.

The general impression was that a lot of work needs to be done in CAD for system level design. Especially, a lot of ground-breaking work in the system verification and simulation is needed to create a fertile ground for system level synthesis. Several people argued that these problems better be solved first before we start thinking about the synthesis aspects. But this might lead to another simulation based specification (remember VHDL), and it is not certain that this is the thing we are waiting for. Synthesis should be considered right from the start. Or as Dan Gajski put it: “These are the signs of an emerging area. Most of you may be too young but I have been through this several times. The fact that we are not certain which steps to take towards system level synthesis make this a primary research topic”. Several of the research topics predicted to make a presence at the 1996 ISSS are listed in table 4.

5 Focus Groups Answers

In the following section all answers to all questions as prepared by each of the focus groups and presented at the final plenary session are given.

5.1 Focus Group 1

1. Yes, need of automation
   - executable specifications
   - modeling
• simulation
• flexibility

2. Customers
• designers
• IC company
• start up

3. System Level Synthesis is not domain specific
• Hard to defend, group 2, 4, 6, & 8 may explain why?
• 2 arguments: - common abstraction level - too many tools = too many $

4. Raise abstraction level
Need abstraction at a higher level → to be defined

5. Cooperations
• Reenforce links between Universities and Industry
• Industry should 1) invest 2) spend manpower
• More support from System Houses and Start ups
• Sort of join support as a non profit organization (SRC)

6. ISSS96 topics
• Low Power
• System Level Abstraction Level
• Reconfigurable Hardware in System Design
• Real-time debugging / simulation
• Design reuse
• Executable Specifications

5.2 Focus Group 2
1. Automated Support for System Design is not Required
• Problem is poorly specified
  – few, if any, formalisms
  – multi-objective cost functions
  – many, many degrees of freedom
  – many non-technical considerations
• Relies too heavily on designer
  – designer expertise needed
  – designer is best able to solve problems
• Software analogy
  – software compiler - behavioral synthesis
  – software system design tools do not exist
  – they’ve been at it way longer than we have

3. System-Level Design is Very Domain-Specific
• Systems differ widely in
  – criteria
  – approaches
  – characteristics
• Examples:
  – control vs. data dominated
  – synchronous vs. asynchronous
  – storage vs. arithmetic dominated
  – different data types, organizations
  – high, medium, or low throughput

4. System-Level Integration Needed - Not Synthesis
• Industry not ready for system synthesis
  – logic synthesis: accepted
  – behavioral synthesis: wary
  – system-level synthesis: ?
• Raising the abstraction level requires:
• unified formalism's
• unified languages
• System-Level Design is not an optimization problems
  – design re-use is very important
  – designs rarely start from scratch
• First things first!
  – why work on synthesis when we can’t do the basic things yet?
• Not enough market for synthesis tools
  – too domain specific - fragments market
  – no one entity willing to foot the bill

Group 2 conclusions:

<table>
<thead>
<tr>
<th>Low abstraction level</th>
<th>High Abstraction Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Few degrees of freedom Problems</td>
<td>Many degrees of freedom Problems</td>
</tr>
<tr>
<td>- well-understood</td>
<td>- not well-understood</td>
</tr>
<tr>
<td>- common across domains</td>
<td>- domain-specific</td>
</tr>
<tr>
<td>Many optimization problems</td>
<td>Few optimization problems</td>
</tr>
<tr>
<td>Little value-added</td>
<td>High value-added</td>
</tr>
<tr>
<td>Far from application</td>
<td>Close to application</td>
</tr>
<tr>
<td>Fertile ground for synthesis tools</td>
<td>Synthesis research is doomed</td>
</tr>
</tbody>
</table>

Thus: System-level simulation and verification is needed, not synthesis.

5.3 Focus Group 3
1. How do designers design a system level?
   (a) Design TODAY
   • AD HOC Design based on experience
   • use tools at hand because no formal design spec. meth is available
• Simulation Tools Cossap, Queueing Sys, Design Tools,...
• Easy to use, but probably used by few people

FUTURE
• VHDL will be bottom level of specification
• VHDL will be the basis of synthesis automation
• Systems will be defined in an OO-way using an OO language.

(b) Do we need CAD? is it more efficient?
Automation is needed because
• Design time is crucial
• It helps designing complex systems
• It helps verify the system

(c) Automation at System Level
• consists of exploration tools
• Will bring system design to everyone. Now, systems are designed by people with lots of experience

2. Who are the customers for system level design tools:
• System Prototypers
• Industry
• Multimedia Appl. Developers
• Universities
• SW Engineers (?)

3. How domain specific is system level synthesis?
It is NOT domain specific
• System = \sum Blocks, System Arch. are about the same
• Behavioral SPEC is very general e.g. you can use C for both DFOriented and CFOriented SPEC
• We don't know whether it is or it isn't. We need to do more research!

4. Is system level synthesis raising the level of abstraction?
It is:
• Current systems are designed ad hoc. We can only avoid this by
  – having higher abstraction
  – having this by means of a common level of specification
• Design Time vs. Design Quality When time is more important, we can afford the abstraction

5. Is System Level Synthesis too large of an area for our community?
We could also use a hand from:

• Software & S.E. people if we use a canonical form of HW/SW representation
• Domain-Specific people which bring application experience

6. The 5 hottest system level topics
• Multimedia applications
• Code generation S.E. techniques Network arch. on the protocol level
• Knowledge representation Partitioning + transformation tech.

5.4 Focus Group 4
1. Main players in system level design:
   System houses
   • have existing architectures & predefined implementations
   • Design knowledge is organization - specific
   • Most work is incremental updating of existing designs

2. Main customers?
• Product (System-) Definition People
• Skunkworks
• Marketing and customers

3. Domain - Specificity: VERY!
• Technology domain
  – DSP (Datapath)
  – Control dominated
  – Memory dominated
  – General Purpose
• Application domain
  – Consumer/Multimedia
  – Telcom
  – Portable
  – Medical
  – Defense/Aerospace
• Reasons for domain specificity:
  – Heterogeneous types of constraints (Power, Cost, Speed, Reliability,...)
  – Different Techniques needed
  – Users are domain-specific

4. Rise of Level of Abstraction? NO!
Most work is done making different tools work together = Integration

5. Cooperation
• Defense industry structured to promote system engineering
• Cooperation between system houses, research and EDA industry.
6. Top Four for ISSS96
   - Specification Languages
   - Combining multiple levels of abstraction
   - Design environments: Frameworks
   - Integration of tools into companies

5.5 Focus Group 5
1. Abstract specification (C models)
   RTL currently used in commercial tools, above it manual design
   By raising the level of abstraction we:
   - Better interact with other designers
   - Create customers
     - Existing ones: Network Multimedia, currently using ASIC CAD tools
     - New ones: Automotive Toy manufacture

2. Does system specification exist today?
   Requirement:
   - Domain independent
   - supports co-simulation
   - supports co-debugging
   - supports synthesis
   Methodology could be:
   - Domain independent
   - Automated
   - At a higher level of abstraction

3. System Level Synthesis is domain dependent while HLS was technology dependent.

4. We need estimation/exploration tool
   - Top down (raising) versus bottom up (integration)
   - Exploration space

5. Who will pay?
   - Complex relation between government, academy, system houses and CAD vendors. Everybody is waiting for the other. Government needs to bootstrap using $.
   - Need some initial success stories proof of concepts
   - Where are the volunteers?
   - Who will pay for Gajskis’ students?

6. ISSS96
   - Specification
   - Co-simulation
   - Estimation
   - Partitioning
   - Rapid prototyping

5.6 Focus Group 6
   - Position statement
     - System Level Synthesis (design) is the partitioning and integration of components at several different levels of abstraction.
     - Automatic synthesis at this level is NOT possible
     - Synthesis at higher levels of abstraction is very domain specific.
   - What does a system look like?
     - Many of today's systems are 80% software
     - Large blocks which are reused.
     - Different implementation styles & toolsets
     - Some analogue
   - System design is analysis, refinement & validation
     - analyzing requirement, often stochastically
     - removing ambiguity/ challenging assumptions
     - making major system decisions - usually apples to oranges
     - designing & architecture to implement
     - verifying design meets intent.
     - ie: System Design is removing ambiguity & partitioning the problem & specification of components & integrating & validating implementations.
   - Tools/support required
     - 'C' compiler
     - Estimators
     - Verification/validation
     - interface analysis (& maybe synthesis)
     - Domain specific synthesis for components

1. Use domain specific synthesis as next level of abstraction for sub components.
   - DSP (& Data dominated)
   - Controllers
   - Protocols
   - Interfaces
   - Memory/cache
   - Datapath

2. Enlarge involvement
   - Stimulate Co-operation?
     - More design examples (e.g., authors make examples public)
     - Industry Academia Co-Operation:
       * visits in both directions (co-ops, university/industry sabbaticals)
5.7 Focus Group 7

1. How do designers design?

(a) Design specification:
- incrementally, re-using previous results
- Automation
  - SHOULD
    - Verification
    - Integration
    - Domain specific
    - Test bench generation
  - SHOULD NOT
    - HW/SW Partitioning
- Efficiency improvement from
  - less chance of bugs at more natural description level
  - simulate faster and sooner
  - changing is more easy
  - automate test bench generation

2. Customers: Will there be more customers when there is system-level software?

3. Domain specific
- NOT specific: VHDL, C, design phases, lower-level synthesis

4. System level synthesis raises the abstraction level
- more natural specification
- faster simulation

5. Too large an area?
- no, just take small steps
- How do we stimulate cooperation/funding
  - exchanging people/tools/examples
  - industry funds students
  - maybe through research institute

6. Hottest topics
- DSP compilers
- More software partitioning/code generation
- Co-simulation (different semantics/domains)
- More domain specific
- Verification

5.8 Focus Group 8

1. Some arguments against automation support being required on system level
- Technical:
  - "Loose" ends - lack of integration
  - domain specific, domain mismatch
  - ROI is:
    - For designer: reduction of design time, documentation
    - For EDA vendor: # of customers
  - Limited design exploration:
    - tools limit design space
    - effect of parameter modifications on subsequent synthesis steps and results
    - embedding of own design expertise?
- Non-Technical:
  - proprietary
  - previous investments bias decisions
  - education (designers first have to exploit existing levels)
  - risk (new methodologies)

2. Who will buy these tools?
- Big system houses ...can afford but cannot live with suboptimality
- SMES ...cannot afford
- System house's customers ...???

3. System-Level Design: very domain specific!

<table>
<thead>
<tr>
<th>Domains:</th>
<th>Domains:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control vs. Data</td>
<td>Sync. vs. Async.</td>
</tr>
<tr>
<td>Pipelined vs. Nonp.</td>
<td>Analog vs. Digital</td>
</tr>
<tr>
<td>Queuing</td>
<td>MCM vs. Single Chip</td>
</tr>
<tr>
<td>Protocol</td>
<td>Microsystems</td>
</tr>
<tr>
<td>Switching/Networking</td>
<td>Computing</td>
</tr>
<tr>
<td>Compression</td>
<td>Wireless/PDA PCS</td>
</tr>
<tr>
<td>Video, Image</td>
<td>Optoelectronics</td>
</tr>
<tr>
<td>Audio/Speech</td>
<td></td>
</tr>
<tr>
<td>Memory intensive</td>
<td></td>
</tr>
</tbody>
</table>
4. Raising abstraction level versus integration
   - Necessary abstraction levels exist, main problem is integration
   - "Turing machine" single abstraction level
   - System design requires by nature heterogeneous specification
5. Who pays? See panel discussion on Wednesday
6. Hot Topics
   - Real time kernels & OS
   - Power
   - Emulation/Simulation
   - Compilers
   - Design Reuse
   - Testbench Synthesis

5.9 Focus Group 9
1. (a) How do designers design at the system level?
   - Any way he knows
   - Experience
   - Mixed bag of tools
(b) How is the design specified?
   - Incompletely
   - Moving target
(c) Is automation really necessary at this level? YES!
   - Design Validation
   - Evaluation - efficient exploration
   - Enforce Methodology
   - Remove Mundane, Repetitive Tasks
2. Who are the main customers for system-level synthesis?
   - Everybody, because SLS is not domain specific & abstraction level is high.
3. How domain specific is system-level synthesis?
   - It is not!
   - System-Level is by definition heterogeneous
   - Today, domain specific tools exist - System Level will bring them together.
   - System-Level Design can be compared to a SW toolbox that will bring solutions together
4. System Level Synthesis is: Raising the level of abstraction:
   - only this allows to cope with detail
   - Level of reasoning must be raised
5. Is SLS too large for our community?
6. Hottest Topics
   - Language Commonality - HW/SW
   - Code Gen & Optimization
   - Estimation
   - Codesign Methods - Retargeting
   - Tool Cooperation/Interoperability
6 Conclusions
   This report describes the focus groups held at the 8th ISSS. System level synthesis is perceived to be at its initial stages. Lots of work needs to be put into formalizing the description means at this level. This will enable the use of analysis and design verification tools and naturally break-the-way for system level synthesis tools. There are very many questions to be answered. In many cases it is not even defined what the questions really are. This is exactly what makes this an exciting area of research! and will provide many new insights to be presented at future symposiums.
7 Acknowledgments
   I would like to thank all participants to the Eighth International Symposium on System Synthesis for their very enthusiastic participation in these focus groups. In particular, all focus group leaders are acknowledged for their efforts to make this successful. Special thanks to Cheryl Krull to summarize all data from the overhead sheets and Francky Catthoor to help define the questions.
References