Hardware description and verification

Getting hardware designs right [using ideas from computer science]

**BACKGROUND KNOWLEDGE**

Verification beginning to dominate development cost (cf. ITRS)

The idea that expressiveness and ease of proof have to be traded off against each other

The importance of automation

FV remains a complement to simulation (see talk about verification at IBM)

Mixing sim. and FV now hot (called semi-formal, cf. IBM Sixth Sense)

Other spe. Langs include SVA (System Verilog Assertions)

**TECHNICAL**

How to write small Lava descriptions and properties

Properties as observers notion of a safety property

Use of recursion in structural cct desc. (examples and answers in LT)

Advantages and disadvantages

Where it fits in a standard approach

SAT-based verification temporal induction

**BACKGROUND (languages)**

Two level language (generators), domain specific embedded language

Patterns for verification and not just for circuits would be useful

Need *both* to take account of low level physical details *and* raise level of abstraction at which design is done!

Wired attempts to address a small version of this problem

See also Matthew Naylor’s cool A Recipe for controlling Lego using Lava

**BACKGROUND (languages)**

Use of functional languages in hardware design and verification at Intel. IDV very interesting (see Seger’s slides)

But it is not clear if there are real openings in large companies for non-standard languages

However, look at Mitrionics AB!

In the mainstream, System Verilog looks to be winning

Note Bluespec SV
BACKGROUND (verification)

Need for hierarchy (Assume Guarantee reasoning, environment models)
Commercial formal tools based on a variety of methods (BDD-based MC, BMC, induction, CEGAR, ...)
Scalability of underlying algorithms still a major issue (and research problem)
Verification of embedded software combined with hardware is a current hot topic, but note that HARDWARE VERIFICATION IS NOT A SOLVED PROBLEM (see interesting talks on FMCAD 2006 website)

Please give feedback about the course to Joel or Thomas (or of course to me directly).

Masters thesis project topics
Using Lava or Wired to design and analyse X
  can have emphasis on Lava/Wired or on X
  X is typically at the level of adder or multiplier but we are open to suggestions
[PhD position also available]

Masters thesis project topics
Using a standard flow to design and analyse an X
  usually I have invented or modified X

Could probably be jointly supervised by Per Larsson-Edefors

Masters thesis project topics
Back to FPGA programming using Lava
  e.g. implementing sorters
Using Lava-like ideas but at a higher level of abstraction (cf. Naylor’s Lego mindstorms work)
Returning to the notion of design by refinement
Parallelising EDA algorithms
GPU programming!