

# Formal Methods in Hardware (Circuit) Design

Budi Rahardjo  
PPAU Mikroelektronika ITB  
rahard@paume.itb.ac.id

## Background

- Increase in circuits and systems size and complexity.
- Designers are forced to leave low-level design methodologies and move towards a higher level of abstraction.
- Final design could be constructed from a high level description using synthesis tools or hand crafted by experienced designers.

## Background (cont.)

- By spending time at the higher level, one can concentrate on the design itself, rather than the detailed implementation.
- Similar trend in software design. (e.g. CASE tools)
- Hardware design resembles traditional computer programming (eg. the use of HDLs).

## Formal methods: intro

*Non-exhaustive testing can be used to show the presence of bugs, but never to show their absence.  
(E. W. Dijkstra)*

- Simulations have been the only technique to check the correctness of a design.
- For complex designs, this ad-hoc method is impractical (impossible?)

## FM: intro (cont.)

- Still, hardware and software designs must be tested/verified. Esp. if used in critical applications:
  - medical (heart pacemaker)
  - fly-by-wire control systems
  - missile guidance

## FM: intro (cont.)

- Important in commercial setting:
  - catch errors as early as possible
  - reduce design time
  - reduce cost

## Results of hard/software bugs

- Ariane5 rocket
- Rocket carrier for Mariner I
- Delay in Patriot upgrade (to intercept Scud missiles)
- Phone disconnected (LA, SF, wash. DC, Virginia, Baltimore, Greensboro, Juli 1991)
- Pentium bugs

## Formal methods: what is it?

- Formal methods use mathematics or mathematical analysis techniques in the development of a design.
- Exhaustive and partial simulations are replaced or augmented with mathematical proofs or a systematic state space search.

## FM in hardware design

- FM applications in hardware design is more successful compared to software:
  - Size of problems in hardware verification is usually smaller than to that of software.
  - Hardware problems are within capabilities of existing tools (e.g. theorem proving tools).
  - Hardware usually implements simple algorithms, leaving complex algorithms to their software counterparts.

## FM in hardware design (cont.)

- Hardware designers are familiar and more used to modular designs with a small restricted (and well tested) set of libraries as their building blocks.
- Individual components in the library can be verified separately, reducing the complexity of the whole design.

## Design process

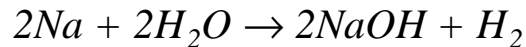
- Insert Figure 3.1 here.

## Formal Specification

- A specification is a description of what a hardware circuit must do without constraining how it is achieved implemented.
  - informal: in a natural language (eg. English), ambiguous interpretation of imprecise / poorly worded descriptions.
  - formal: HDL, mathematical notations, machine checkable syntax.

## Formal Specification

- In Chemistry:



- Math: **Fermat's Last Theorem**

*There do not exist four positive integers, the last being greater than two, such that the sum of the first two, each raised to the power of the fourth, equals to the third raised to that same power.*

There do not exist integers such that  $x^n + y^n = z^n$ , where  $x, y, z > 1$  and  $n > 2$

## Formal Specification: Hardware

- Specification languages: VHDL, Verilog, Circal, ELLA, etc.
- HDLs are **not** considered formal systems, i.e. one cannot reason about the design in that HDL directly.
- *VHDL has no formal mathematical semantics as part of its definition, hence, programs written in it have not been amenable to formal analysis.*

J.P. Van Tassel, "Femto-VHDL: the semantics of a subset of VHDL and its embedding into HOL," Phd thesis, Gonville and Caius College, University of Cambridge, July 1993.

## Formal Verification

- Formal verification: a formal (analytical) demonstration that an implementation satisfies (correctly meets) its specification.
- *The difference between formal verification and simulation is similar to the difference between deriving laws in physics from first principles and performing experiments.*

M. C. McFarland, "Formal Verification of sequential hardware: A tutorial," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 12(5), pp. 633-654, May 1993.

## Relationship between Implementation and Specification

- Forms of proof methods used in establishing formal relationship:  
(A. Gupta, "Formal Hardware Verification Methods: A Survey," in Formal Methods in System Design, vol 1, Kluwer Press, 1992.)
  - **theorem proving.** relationship is regarded as a theorem in logic, to be proved within the context of a proof calculus, where the implementation provides axioms and assumptions that the proof can be drawn upon.

## Relationship... (cont.)

- **model checking.** specification is in the form of logic formula, the truth of which is determined with respect to a semantic model provided by an implementation.
- **equivalence checking.** equivalence of functions, FSM, etc.
- **language containment.** the language representing an implementation is shown to be contained in the language representing a specification.

## Implementation

- State space search
- Theorem proving environment

more to come...

- Ordered Binary Decision Diagram (OBDD)

## Reading materials

- B. Rahardjo, "Formal Verification of Asynchronous Systems," PhD thesis, University of Manitoba, Canada, 1996.
- NASA Langley Research  
<ftp://air16.larc.nasa.gov/pub/fm/larc/RCP-papers>