Design Verification
COMS30026

(COMS30025 with coursework)

WEEKLY LIVE SESSION

Kerstin Eder

Trustworthy Systems Lab

University of BRISTOL

Department of COMPUTER SCIENCE

Week 2

Topics

- Introduction to DV
- Verification Hierarchy
- Fundamentals of Simulation-based Verification
  - Driving & Checking

Task(s) – Week 1

- **Review "*The limits of correctness*".**

  SIGCAS Comput. Soc. 14,15, 1,2,3,4 (Jan 1 1985), 18–26. DOI: https://doi.org/10.1145/379486.379512

  - *Identify the main lines of argument*
  - *Why does the author question the notion of "correctness"?*
  - *What are the two or three key take-away messages for you?*

THE LIMITS OF CORRECTNESS†

Brian Cantwell Smith*

1. Introduction

On October 5, 1960, the American Ballistic Missile Early-Warning System station at Thule, Greenland, indicated a large contingent of Soviet missiles headed towards the United States.[1] Fortunately, common sense prevailed at the informal threat-assessment conference that was immediately convened: international tensions weren't particularly high at the time. The system had only recently been installed. Kruschev was in New York, and all in all a massive Soviet attack seemed very unlikely. As a result no devastating counter-attack was launched. What was the problem? The moon had risen, and was reflecting radar signals back to earth. Needless to say, this lunar reflection hadn't been predicted by the system's designers.

Over the last ten years, the Defense Department has spent many millions of dollars on a new computer technology called "program verification" - a branch of computer science whose business, in its own terms, is to "prove programs correct". Pro-

What, we do well to ask, does this new technology mean? How good are we at it? For example, if the 1960 warning system had been proven correct (which it was not), could we have avoided the problem with the moon? If it were possible to prove that the programs being written to control automatic launch-on-warning systems were correct, would that mean there could not be a catastrophic accident? In systems now being proposed computers will make launching decisions in a matter of seconds, with no time for any human intervention (let alone for musings about Kruschev's being in New York). Do the techniques of program verification hold enough promise so that, if these new systems could all be proven correct, we could all sleep more easily at night? These are the questions I want to look at today. And my answer, to give away the punch-line , is no. For fundamental reasons - reasons that anyone can understand - there are inherent limitations to what can be proven about computers and computer programs. Although program verification is an important new technology, useful, like so many

Task(s) – Week 2

- **Review the foretellix blog** https://blog.foretellix.com/

HOME ABOUT WHY THIS BLOG TERMINOLOGY TOPICS ALL POSTS

GPT-3 and verification

July 20, 2020

Summary: This post talks about GPT-3, a new Machine Learning (ML) system currently making waves in the ML community. It explains why GPT-3 is a big deal, and then considers the verification implications of such systems. One way to look at GPT-3 (and the even-bigger GPT-4, GPT-5 etc. which are sure to follow) is as ...
More

Search ...

Archives

Select Month

Opportunities – Week 1

- RV2020 Keynotes
  - 20TH INTERNATIONAL CONFERENCE ON RUNTIME VERIFICATION
    - Los Angeles, October 6-9, 2020

RV Keynote by Lane Desborough on *The Physical Side of Cyber-Physical Systems*

1. It is possible and valuable to cross domains
2. The future is already here, it just hasn't been evenly distributed yet. – William Gibson
3. Bad things can happen during mode transitions, when the state of the system is changing
4. Never forget that the physical side of cyber-physical systems involves energy
5. Manage variation using hierarchical, temporal decoupling
6. In cyber-physical systems, most any parameter can be a critical parameter … so manage them all carefully … if you don't manage change, change will manage you
7. "Open" Process Automation Systems are nearly impossible to comprehend
8. Interoperability is not a panacea
9. So simple there are obviously no errors, or so complex there are no obvious errors
10. Hardware, software, wetware, and the cyber-physical systems comprised thereof are different and should be treated as such
11. Systems: the bigger they are, the less frequent – but harder - they fall
12. Take on only as much complexity as you can manage – lives may be at stake
13. When our commercial reach exceeds our technical grasp, we must look for new approaches
14. Humans and computers are good at different things; improper task allocation creates problems
15. Automation changes the nature of use error: from acts of commission to acts of omission
16. The methods, tools, and processes to compose small, medium, and large systems are different
17. Emergent properties will emerge
18. The challenges with complexity have been known for a long time
19. That which you do not have does not cause problems
20. Complexity is easy to add hard to remove
21. Complexity adds cost, risk, and delay (and technical debt, and late cycle surprises)
22. All models are wrong, but some are useful – George Box
23. We build models to efficiently characterize what the system will do
24. Modeling is a means not an end; different kinds of modeling serve different purposes
25. One must be prepared to use as many characterizations methods as necessary

NUDGE

Opportunities – Week 2

- *CDN Live 2020, 13-14 Oct 2020*
- 2020 Wilson Research Group Verification Survey Results
  - Presentation by Harry Foster tomorrow
  - Tue 13th Oct 2020 at 16:00 local time

CDN Live 2020 13-14 Oct 2020

- You can register using the promo-code "partneremea20" at https://events.cadence.com/CadenceLIVEEurope
- On the agenda you can see different sessions. Note that times are in CEST.
- Keynote at **10:00 on Wednesday, 14th October, by Alberto SanGiovanni-Vincentelli.**

2020 Wilson Research Group Verification Survey Results

EVENT

Register For This Web Seminar

Online - Oct 13, 2020
8:00 AM - 9:00 AM US/Pacific
Convert to Local Time

Register

Go URL https://go.mentor.com/5en5g

Overview

The 2019 global semiconductor market was valued at $385.4 billion after experiencing a 15% decline due to a 32% drop in the memory IC market, which is expected to recover in 2021. The IC/ASIC portion of the semiconductor market is valued at about $186.6 billion. The IC/ASIC semiconductor market is expected to reach a value of $233.4 billion by 2024, growing at a compounded annual growth rate (CAGR) of 4.6% during this forecast period. In contrast, the FPGA portion of the semiconductor market is valued at about $5 billion, and is expected to reach a value of $7.5 billion by 2030, growing at a compounded annual growth rate (CAGR) of 4.4% during this forecast period. While COVID-19 has had a negative impact in 2020 on a number of market segments (e.g., consumer and automotive), other market segments (e.g., data center computing, networking, storage, and communication) are experiencing positive growth required to support today's growing work-from-home (WFH)

ABOUT THE PRESENTER

Harry Foster
Harry Foster is Chief Scientist Verification for the Design Verification Technology Division of Mentor, A Siemens Business; and is the Co-Founder and Executive Editor for the Verification Academy. In addition, Harry is serving as the 2021 Design Automation Conference General Chair. He holds multiple patents in verification and has co-authored six books on verification. Harry was the 2006 recipient of the Accellera Technical Excellence Award for his contributions to developing industry standards, and was the original creator of the Accellera Open Verification Library (OVL) standard.

Week 1

Welcome to Design Verification

- Lecturer and Unit Director
  - Kerstin EDER
  - Department of Computer Science
- Lecture notes, exercises and additional information are available at uobdv.github.io/Design-Verification/
- Recordings of the lectures will be made available on a weekly basis via Blackboard
- Comments and feedback are always welcome

Design Verification Unit Details

- Lecture times (weeks 1-7 and weeks 11-12)
  - Please check your timetable, at the moment:
  - Monday (1h) 11:00 online synchronous session (Lecture)
  - Fridays (1h) 10:00 Peer learning sessions

- Practical Work (weeks 1-7 and weeks 11-12)
  - Tuesday (1h) 12:00 online lab-based seminar with remote desktop bookings
  - Teaching Assistant: Xuan Zheng

Timetable

University of BRISTOL Timetable of activities - Teaching Block 1 (weeks 1 - 12)

	08:00	08:30	09:00	09:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30	14:00	14:30	15:00	15:30	16:00	16:30	17:00	17:30	18:00	18:30
Monday							COMS30026 Wks:1 - 7, 11, 12 Design Verification KIE Lecture															
Tuesday									COMS30026 MVB 1.15 PC Wks:1 - 7, 11, 12 Design Verification - Room Booking - Remote Desktop COMS30026 Wks:1 - 7, 11, 12 Design Verification KIE Seminar													
Wednesday																						
Thursday																						
Friday					COMS30026 Wks:1 - 7, 11, 12 Design Verification - Peer Learning Teaching Other																	

Design Verification Assessment

- Assessment
  - COMS30026 will be assessed in the January exam period (exam only)
  - COMS30025 will be assessed by coursework (only)

Literature and Study Resources

- **Writing Testbenches: Functional Verification of HDL Models** by Janick Bergeron. Second Edition, Kluwer, 2003.
- **Comprehensive Functional Verification** by Bruce Wile, John Goss and Wolfgang Roesner. Elsevier, 2005.
- verificationacademy.com
- In addition:
  - Lecture slides and on-line tutorials on unit web page
  - On-line documentation of ModelSim/Questa Simulator and SpecMan Elite
  - Watch the unit web page for further supplementary literature.

[**Credits:** Parts of the lecture notes contain material from the book “Comprehensive Functional Verification” by Bruce Wile etal, the book ”Writing Testbenches: Functional Verification of HDL Models” by Janick Bergeron, the book ”The Verilog Hardware Description Language” by Donald Thomas and from lecture slides developed at IBM (by Avi Ziv and Jaron Wolfstal), the University of Pittsburgh, Penn State University, North Carolina State University and Ohio State University. The HDL for the assignments has been developed at IBM.]

What is this unit about?

Aim: To familiarise you with the state of the art in Design Verification, and to give you the **technical background** plus some of the **practical skills** expected from a professional Design Verification Engineer.

- Pre-/Co-requisites: programming experience

On successful completion of this unit, you will be able to:

- understand the complexities and limits of verification;
- carry out functional verification and determine its effectiveness;
- set appropriate verification goals, select suitable verification methods and assess the associated risks;
- compile a verification plan that fits into the flow of a design project.

Unit Outline

Lecture Topics

- Introduction: What is Verification? What is a Testbench?
- Verification Flow and Tools including basic Verilog HDL coding
- Verification cycle, methodology and plan including coverage
- Simulation-based Verification: Stimulus Generation and Checking
- Assertion-based Verification (ABV)
- Advanced Testbench Design Methodology with SpecMan Elite
- (Functional Formal Verification and Property Checking)

Practical work

- Exercise 1: Evita Verilog interactive tutorial (do @ home asap)
- Exercise 2: Introduction to the ModelSim/Questa Simulator
- Practical 1, weeks 2-5: Verification of calculator design with ModelSim
- Exercise 3: How to collect Code Coverage with ModelSim/Questa
- Exercise 4: Introduction to SpecMan Elite
- Practical 2, weeks 6-7 & 11-12: Advanced testbench design with SpecMan Elite and formal verification with JasperGold

What is Design Verification?

What is Design Verification?

"Design Verification is the process used to gain confidence in the correctness of a design w.r.t. the requirements and specification."

Types of verification:

- Functional verification
- Timing verification
- ...
- What about performance?

Verification vs Validation

- Verification:
  - Confirms that a system has a given input / output behaviour, sometimes called the **transfer function** of a system.
- Validation:
  - Confirms that the system's transfer function results in the intended system behaviour when the system is employed in its target environment, e.g. as a component of an embedded system.
- Validation is sometimes used when verification is meant.

Annex A
(informative)

The Role of Testing in Verification and Validation

Figure 11 — Hierarchy of Verification and Validation activities

Figure 11 defines the complete nature of verification and validation (V&V) activities. V&V can be done on system, hardware, and software products. These activities and planning are defined and refined in IEEE 1012 and ISO/IEC 12207. Much of V&V is accomplished by testing. The ISO/IEC 29119 standard addresses the Dynamic and Static software testing (directly or via reference), thus covering parts of this verification and validation model. ISO/IEC 29119 is not intended to address all the elements of the V&V model, but it is important for a tester to understand where they fit within this model.

DRAFT INTERNATIONAL STANDARD ISO/IEC DIS 29119-1

ISO/IEC JTC 1 — Secretariat: **ANSI**

Voting begins on **2012-10-09** — Voting terminates on **2013-01-09**

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОММИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Software and systems engineering — Software testing —
Part 1:
Concepts and definitions

Ingénierie du logiciel et des systèmes — Essais du logiciel —
Partie 1: Concepts et définitions

ICS 35.080

To expedite distribution, this document is circulated as received from the committee secretariat. ISO Central Secretariat work of editing and text composition will be undertaken at publication stage.

Pour accélérer la distribution, le présent document est distribué tel qu'il est parvenu du secrétariat du comité. Le travail de rédaction et de composition de texte sera effectué au Secrétariat central de l'ISO au stade de publication.

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENT AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

© International Organization for Standardization, 2012
International Electrotechnical Commission, 2012

Next

- Recordings of lectures on
  - Introduction to design verification and
  - Verification Hierarchy

  will be uploaded to BB within the coming two days. Recordings will then be uploaded on a weekly basis.

- **Tasks for you:**
  - **Attend the lab session** with Xuan tomorrow to set up remote access to the EDA tools
  - **Review "*The limits of correctness".*** *SIGCAS Comput. Soc. 14,15, 1,2,3,4 (Jan 1 1985), 18–26.* DOI: https://doi.org/10.1145/379486.379512
    - *Identify the main lines of argument*
    - *Why does the author question the notion of "correctness"?*
    - *What are the two or three key take-away messages for you?*

Paper review

Brian Cantwell Smith. 1985. The limits of correctness.
SIGCAS Comput. Soc. 14,15, 1,2,3,4 (Jan 1 1985), 18–26.
DOI: https://doi.org/10.1145/379486.379512

THE LIMITS OF CORRECTNESS†

Brian Cantwell Smith*

1. Introduction

On October 5, 1960, the American Ballistic Missile Early-Warning System station at Thule, Greenland, indicated a large contingent of Soviet missiles headed towards the United States.[1] Fortunately, common sense prevailed at the informal threat-assessment conference that was immediately convened: international tensions weren't particularly high at the time. The system had only recently been installed. Kruschev was in New York, and all in all a massive Soviet attack seemed very unlikely. As a result no devastating counter-attack was launched. What was the problem? The moon had risen, and was reflecting radar signals back to earth. Needless to say, this lunar reflection hadn't been predicted by the system's designers.

Over the last ten years, the Defense Department has spent many millions of dollars on a new computer technology called "program verification" - a branch of computer science whose business, in its own terms, is to "prove programs correct". Pro-

What, we do well to ask, does this new technology mean? How good are we at it? For example, if the 1960 warning system had been proven correct (which it was not), could we have avoided the problem with the moon? If it were possible to prove that the programs being written to control automatic launch-on-warning systems were correct, would that mean there could not be a catastrophic accident? In systems now being proposed computers will make launching decisions in a matter of seconds, with no time for any human intervention (let alone for musings about Kruschev's being in New York). Do the techniques of program verification hold enough promise so that, if these new systems could all be proven correct, we could all sleep more easily at night? These are the questions I want to look at today. And my answer, to give away the punch-line , is no. For fundamental reasons - reasons that anyone can understand - there are inherent limitations to what can be proven about computers and computer programs. Although program verification is an important new technology, useful, like so many