Design Verification
COMS30026

(COMS30066 with coursework)

WEEKLY LIVE SESSION – W5

Kerstin Eder

Trustworthy Systems Lab

University of BRISTOL

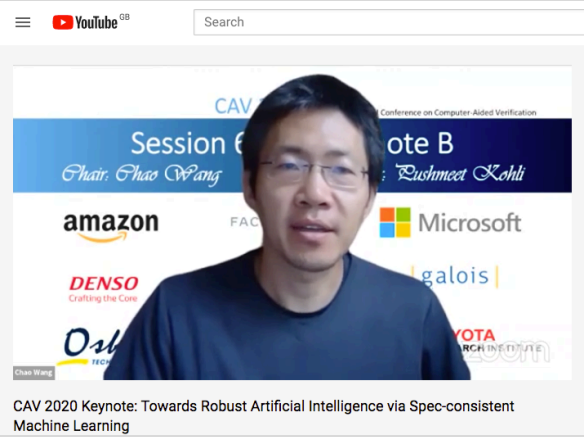
Department of COMPUTER SCIENCE

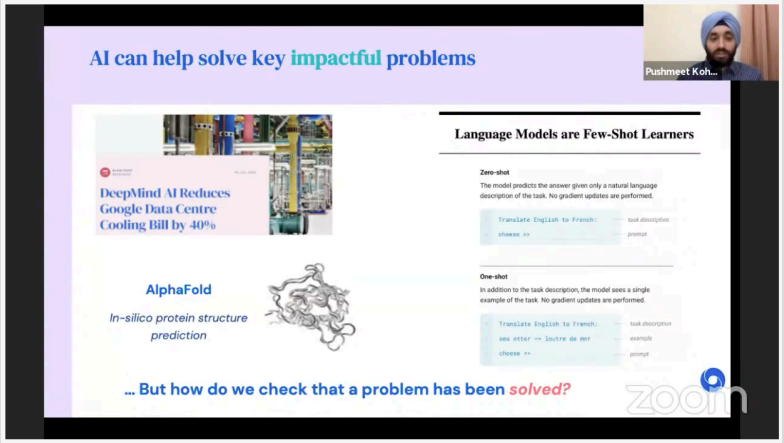
Opportunities – Week 4

- **CAV 2020 Keynote:**

  Towards Robust Artificial Intelligence via Spec-consistent Machine Learning

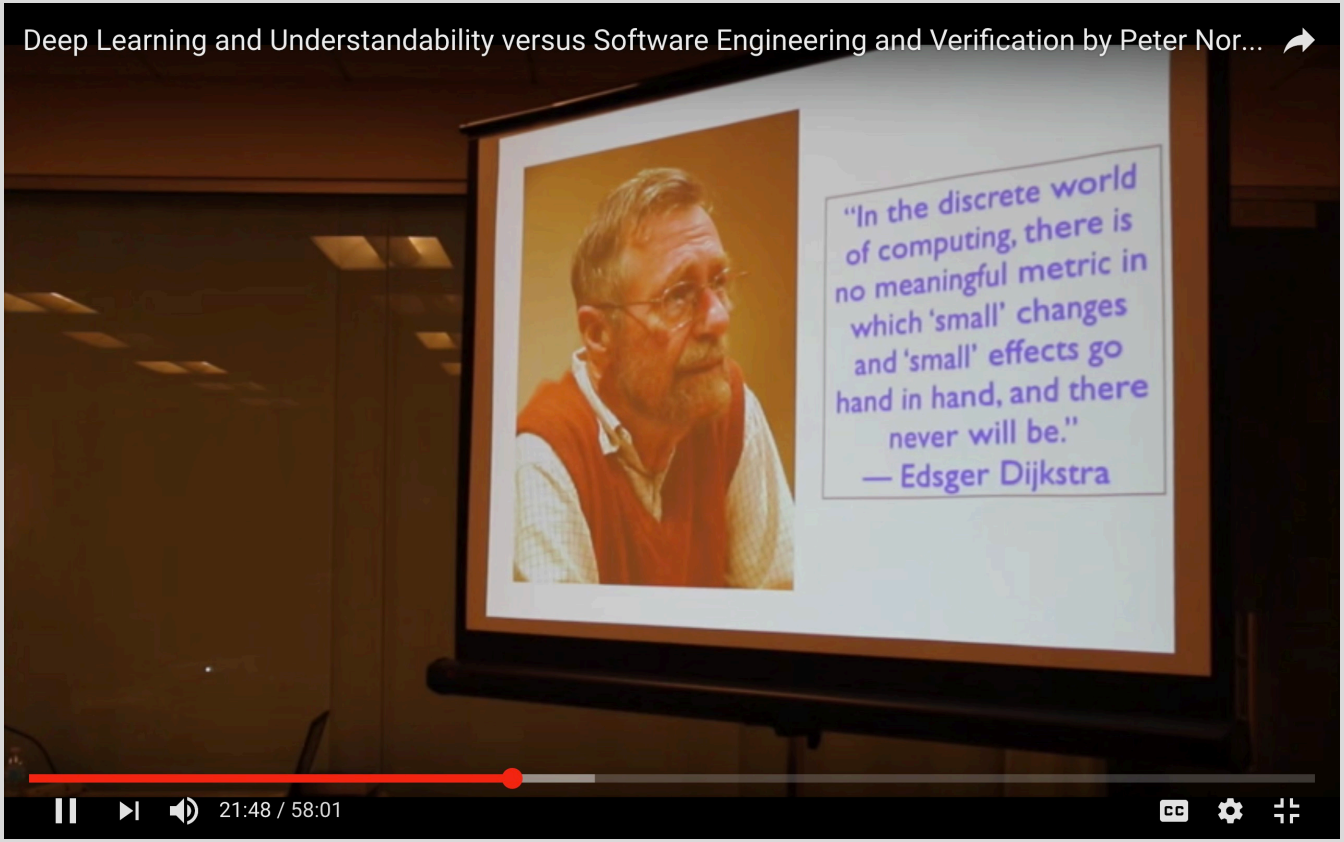
  by Pushmeet Kohli from DeepMind





https://www.youtube.com/watch?v=25RhyyOg5Pg&feature=youtu.be

Opportunities – Week 4

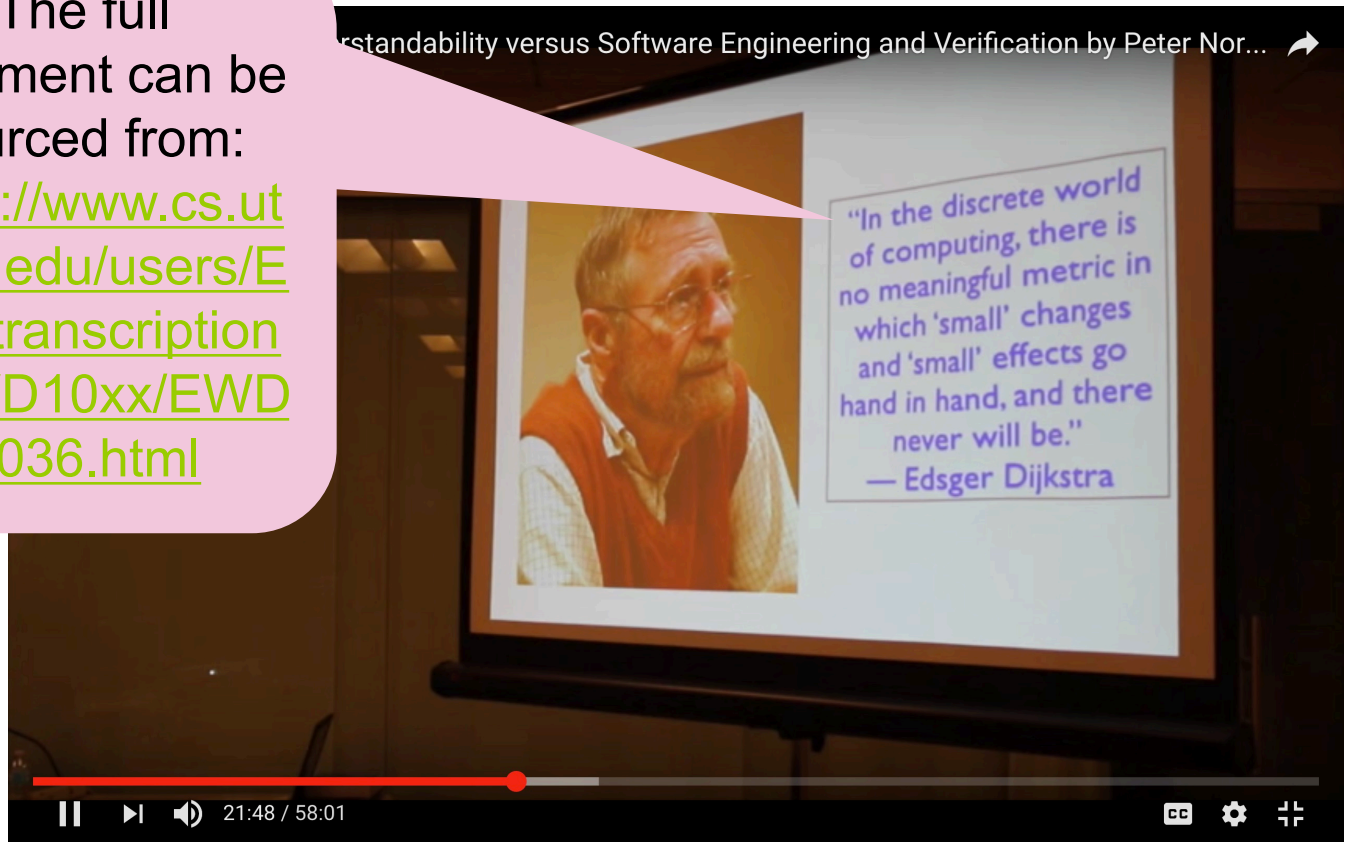


Peter Norvig on *Deep Learning and Understandability versus Software Engineering and Verification*

https://youtu.be/3Ulh8yUiuuY

Opportunities – Week 4

The full document can be sourced from: https://www.cs.utexas.edu/users/EWD/transcriptions/EWD10xx/EWD1036.html



Peter Norvig on *Deep Learning and Understandability versus Software Engineering and Verification*

https://youtu.be/3Ulh8yUiuuY

More fundamental questions

- Does testing work?
  - Is it valid to interpolate between test results?

Mechanical system ⟷ **Digital system**

The traditional testing assumption

- A mechanical balance behaves continuously within limits.
- Allows interpolation between test results.

- With a digital balance, the displayed weight depends on software.
- There is no underlying law to guarantee continuous behaviour.
- In general, it is **wrong** to interpolate between test results.

The traditional testing assumption

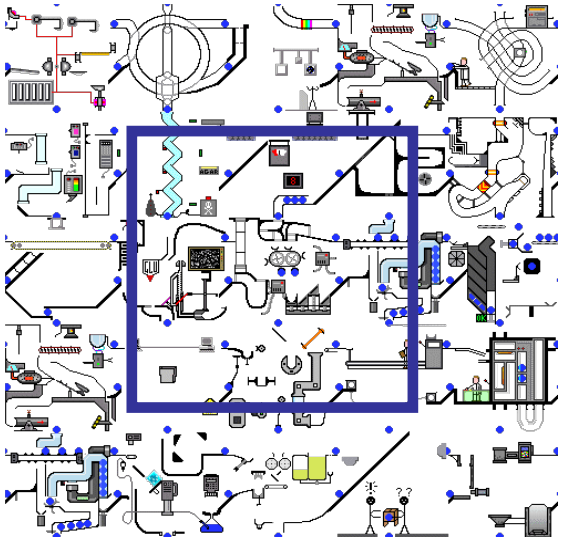
Most testing of digital systems is based on a false analogy!

- With a digital balance, the displayed weight depends on software.
- There is no underlying law to guarantee continuous behaviour.
- In general, it is **wrong** to interpolate between test results.

More fundamental questions

Mechanical system ⟷ **Digital system**

- Does testing work?
  - Is it valid to interpolate between test results?
- Does testing scale?
  - It is impossible to test all interleavings, and impossible to test any meaningful fraction of system states.
- Is testing an appropriate method to gain confidence in the correctness of complex systems?
  - What does testing tell you?
    - Useful for validation
  - Are formal methods the answer?
    - Formal verification
    - Correctness by design
      - Can we ensure functional correctness?
      - Can we ensure safety?
      - Can we ensure no deadlocks?



Four necessary and sufficient conditions for deadlock

System Deadlocks

E. G. COFFMAN, JR.

Pennsylvania State University, University Park, Pennsylvania

M. J. ELPHICK

University of Newcastle upon Tyne, Newcastle upon Tyne, England

A. SHOSHANI

System Development Corporation, Santa Monica, California

E. G. Coffman, M. Elphick, and A. Shoshani. 1971. System Deadlocks. ACM Comput. Surv. 3, 2 (June 1971), 67–78.

DOI: https://doi.org/10.1145/356586.356588

A problem of increasing importance in the design of large multiprogramming systems is the, so-called, deadlock or deadly-embrace problem. In this article we survey the work that has been done on the treatment of deadlocks from both the theoretical and practical points of view.

Key words and phrases: deadlocks, deadly embraces, system deadlocks, multiprogramming, interlock problems

CR categories: 4.10, 4.32

1 INTRODUCTION

One of the objectives of recent developments in operating systems—incorporating multiprogramming, multiprocessing, etc.—has been to improve the utilization of system resources (and hence reduce the cost to users) by distributing them among many concurrently executing *tasks*. In any operating system of this type the problem of deadlock must be considered. Requests by separate tasks for resources may possibly be granted in such a sequence that a group of

But if we assume that T_2 is in the middle of some file on disk, then, because of hardware or software limitations, a preemption of the disk from T_2 can be effectively tantamount to a sacrifice in the processing already accomplished by T_2. Under these circumstances, if T_1 and T_2 have no (temporary) alternative to waiting, they will be deadlocked and never able to proceed.

As shown by this example, deadlocks, or "deadly embraces" as E. W. Dijkstra has called them, can arise even though no single task requires more than the total resources

This offers you an opportunity to learn material beyond what has been taught in this unit.

When does deadlock occur?

Coffman, Elphick and Shoshani identified 4 *necessary and sufficient* conditions for deadlock [*System Deadlocks.* ACM Computing Surveys 3, 2 (June), p. 67-78, 1971.]

1. Agents claim exclusive control of the resources they require.

$\Rightarrow$ **"Mutual exclusion"** condition

2. Agents hold resources already allocated to them while waiting for additional resources.

$\Rightarrow$ **"Wait for"** condition

3. Resources cannot be forcibly removed from the agent holding them until the resources are used to completion.

$\Rightarrow$ **"No preemption"** condition

4. A circular chain of agents exists, s.t. each agent holds one or more resources that are being requested by the next task in the chain.

$\Rightarrow$ **"Circular wait"** condition

Topics W4

- ✓ Coverage Part I: Introduction and Code Coverage
- ✓ Coverage Part II: Functional Coverage
- ✓ Coverage Part III: Coverage Analysis

- **Lab W4:**

  Practical 1 (Weeks 2-5)

  – calc1 DUV provided as **black box design**
    - Specification review
    - Development of a **Verification Plan**
    - Debug calculator design using Modelsim/Questa
    - Example testbench on Blackboard

“Coverage is a measure of effort, not achievement.”

*** *Discuss* ***

NASA/TM-2001-210876

A Practical Tutorial on Modified Condition/ Decision Coverage

Kelly J. Hayhurst
Langley Research Center, Hampton, Virginia

Dan S. Veerhusen
Rockwell Collins, Inc., Cedar Rapids, Iowa

John J. Chilenski
The Boeing Company, Seattle, Washington

Leanna K. Rierson
Federal Aviation Administration, Washington, D.C.

https://ntrs.nasa.gov/citations/20010057789

Table 1. Types of Structural Coverage

Coverage Criteria	Statement Coverage	Decision Coverage	Condition Coverage	Condition/ Decision Coverage	MC/DC	Multiple Condition Coverage
Every point of entry and exit in the program has been invoked at least once		•	•	•	•	•
Every statement in the program has been invoked at least once	•					
Every decision in the program has taken all possible outcomes at least once		•		•	•	•
Every condition in a decision in the program has taken all possible outcomes at least once			•	•	•	•
Every condition in a decision has been shown to independently affect that decision's outcome					•	•[8]
Every combination of condition outcomes within a decision has been invoked at least once						•

[8] Multiple condition coverage does not explicitly require showing the independent effect of each condition. This will be done, in most cases, by showing that every combination of decision inputs has been invoked. Note, however, that logical expressions exist wherein every condition cannot have an independent effect.

2.1 The Verification Process

5. Wiener, Ruth: *Digital Woes, Why We Should Not Depend on Software*. Addison-Wesley Publishing Company, 1993.

"No product of human intellect comes out right the first time. We rewrite sentences, rip out knitting stitches, replant gardens, remodel houses, and repair bridges. Why should software be any different?" (ref. 5)

2.1 The Verification Process

5. Wiener, Ruth: *Digital Woes, Why We Should Not Depend on Software*. Addison-Wesley Publishing Company, 1993.

"No product of human intellect comes out right the first time. We rewrite sentences, rip out knitting stitches, replant gardens, remodel houses, and repair bridges. Why should software be any different?" (ref. 5)

According to DO-178B[3], the purpose of the verification process is to detect and report errors that have been introduced in the development processes. The verification process does not produce software; its responsibility is to ensure that the produced software implements intended function completely and correctly, while avoiding unintended function. Because each development process may introduce errors, verification is an integral process (see Figure 1), which is coupled with every development process. Including verification activities with each development activity is intended to help "build in" quality at each step, because "testing or analyzing in" quality at the end of the life cycle is impractical.

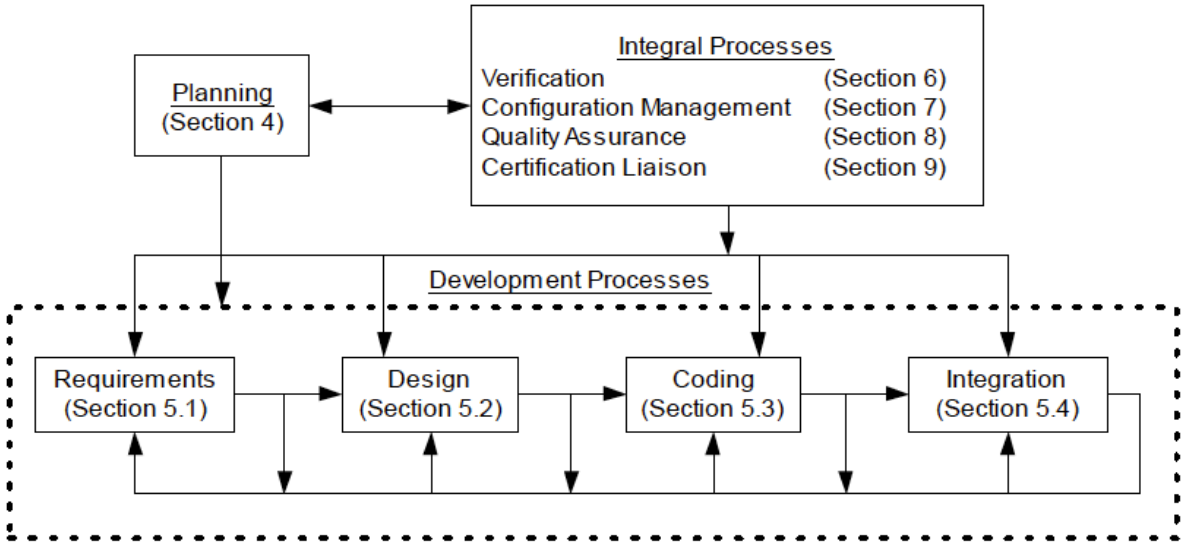


Figure 1. DO-178B software life cycle processes.[4]

Verification examines the relationship between the software product and the requirements. At an abstract level, the software verification process exists to constantly ask the question: are we building the system right?

RTCA, Inc.
1140 Connecticut Avenue, N. W., Suite 1020
Washington, D.C. 20036

SOFTWARE CONSIDERATIONS IN AIRBORNE SYSTEMS AND EQUIPMENT CERTIFICATION

Copyright © RTCA, Inc. 1992

RTCA/DO-178B

December 1, 1992

Prepared by:

RTCA SC- 167 / EUROCAE WG- 12

Software Level Definitions

Software level is based upon the contribution of software to potential failure conditions as determined by the system safety assessment process. The software level implies that the level of effort required to show compliance with certification requirements varies with the failure condition category. The software level definitions are:

a. Level A: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a catastrophic failure condition for the aircraft.

b. Level B: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a hazardous/severe-major failure condition for the aircraft.

c. Level C: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a major failure condition for the aircraft.

d. Level D: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a minor failure condition for the aircraft.

e. Level E: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function with no effect on aircraft operational capability or pilot workload. Once software has been confirmed as level E by the certification authority, no further guidelines of this document apply.

Requirements-based Testing

Table A-7
Verification of Verification Process Results

	Objective		Applicability by SW Level				Output		Control Category by SW level			
	Description	Ref.	A	B	C	D	Description	Ref.	A	B	C	D
1	Test procedures are correct.	6.3.6b	●	○	○		Software Verification Cases and Procedures	11.13	②	②	②	
2	Test results are correct and discrepancies explained.	6.3.6c	●	○	○		Software Verification Results	11.14	②	②	②	
3	Test coverage of high-level requirements is achieved.	6.4.4.1	●	○	○	○	Software Verification Results	11.14	②	②	②	②
4	Test coverage of low-level requirements is achieved.	6.4.4.1	●	○	○		Software Verification Results	11.14	②	②	②	
5	Test coverage of software structure (modified condition/decision) is achieved.	6.4.4.2	●				Software Verification Results	11.14	②			
6	Test coverage of software structure (decision coverage) is achieved.	6.4.4.2a 6.4.4.2b	●	●			Software Verification Results	11.14	②	②		
7	Test coverage of software structure (statement coverage) is achieved.	6.4.4.2a 6.4.4.2b	●	●	○		Software Verification Results	11.14	②	②	②	
8	Test coverage of software structure (data coupling and control coupling) is achieved.	6.4.4.2c	●	●	○		Software Verification Results	11.14	②	②	②	

LEGEND:		
	●	The objective should be satisfied with independence.
	○	The objective should be satisfied.
	Blank	Satisfaction of objective is at applicant's discretion.
	①	Data satisfies the objectives of Control Category 1 (CC1).
	②	Data satisfies the objectives of Control Category 2 (CC2).

Topics W5

- ✓ High-level Verification: sn and e – Part 1
- High-level Verification: sn and e – Part 2

Topics W5

- ✓ High-level Verification: sn and e – Part 1
- High-level Verification: sn and e – Part 2
- **Optional (but now very urgent)**
  - Sign up for Cadence e language training course
  - Follow the link on the Blackboard announcement

Questions | Responses 31

COMS30026 Sign up sheet

For COMS30026 we have been offered access to a professional training course on the e language provided by Cadence. The e language will be used for your second COMS30026 practical.

In order to be registered on this course, Cadence requires your name, surname and UOB email address. If you wish to take advantage of this offer, please fill in this form by 3pm Friday, 6th November 2020.

In compliance with GDPR we draw your attention to Cadence's privacy policy which can be viewed at https://www.cadence.com/content/cadence-www/global/en_US/home/privacy-policy.html.

1. Please confirm your full name:

Enter your answer

2. Confirmation of UoB email address

Enter your answer

+ Add new

Topics W5

- ✓ High-level Verification: sn and e – Part 1
- High-level Verification: sn and e – Part 2
- **Optional (but now very urgent)**
  - Sign up for Cadence e language training course
  - Follow the link on the Blackboard announcement
- **Lab W5:**

  Practical 1 (Weeks 2-5)
  - calc1 DUV provided as **black box design**
    - Specification review
    - Development of a **Verification Plan**
    - Debug calculator design using Modelsim/Questa
    - Example testbench on Blackboard

DESIGN VERIFICATION Practical 1: Debugging a Calculator Design (Weeks 2-5)

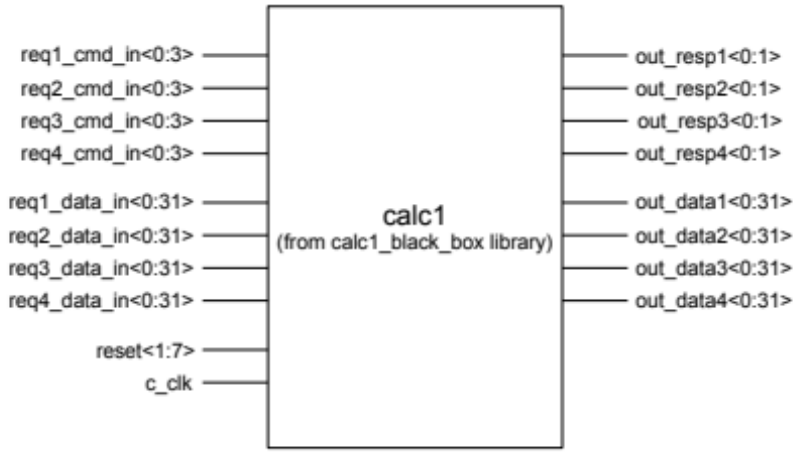
This practical requires you to find bugs in a calculator design. The calculator specification is given below. The design HDL code is not available for this practical. You are expected to use a black box verification approach.

For simulation with ModelSim, the calculator design has been provided in the pre-compiled library `calc1_black_box`. The library is available from BlackBoard as a ZIP file. BlackBoard also provides you with instructions on how to set up your workspace with this library. The top-level module of the calculator design is called `calc1`. Please use the example testbench file provided on BlackBoard to check that you have set everything up correctly.

Calculator Specification

Input/Output Specification

The calculator has four commands: add, subtract, shift left and shift right. It can handle (but not process) four requests in parallel. All four requestors use separate input signals. All requestors have equal priority. Figure 1 shows the input output ports of the calculator.



Next

- Recordings of lectures for **Week 5:**
  - ✓ High-level Verification: sn and e – Part 1
  - ✓ High-level Verification: sn and e – Part 2
  - uobdv.github.io/Design-Verification/ shows a **weekly schedule of topics** to watch BEFORE the next online session, ideally
  - Recordings are available from Blackboard unit page
- No sessions during Week 6 due to reading week!
- Tasks for you this week:
  - Submit issues found during your **Specification review** to Blackboard Discussion Forum
  - Develop a **Verification Plan** for the calc1 black box DUV
  - **Attend the lab session** with Xuan on Friday to get help with using ModelSim/Questa if required so you can do **Practical 1**

Apple Guest Lecture 2021

Pete Vigil on *Design Verification for GPUs - Practice and Challenges*

Wednesday, 15 December 2021, 10am GMT

This talk covers the broad range of verification tools, techniques and processes, introduces aspects of GPU architecture, and explores the unique challenges of verification for GPU designs.

Speaker Bio: Pete Vigil is the site lead and design verification manager for the GPU group at Apple's St. Albans site. He has worked in verification for over 30 years, from start-ups to large companies, and on many types of designs including CPUs, GPUs, and other IP. Pete holds a BSCSE from Santa Clara University, has co-founded two verification start-ups. and is co-author on a number of design patents.

Design vs Verification

Design vs Verification

IC INTELLECTUAL CAPITAL RESOURCES
global technology recruitment partner since 1999

2021 end of year
UK SALARY REVIEW
SOFTWARE | SEMICONDUCTOR | ELECTRONICS | SALES & MARKETING | SUPPLY CHAIN

Salaries

Experience		Graduate	3 years	5 years	10 years	12+ years
Digtial IC Design	*Perm (p.a)*	£34,000	£42,000	£52,000	£65,000	£75,000+
	Cont (p.h)	-	*£42*	*£48*	*£50*	*£52+*
Digital IC Verification		£34,000	£42,000	£55,000	£65,000	£80,000+
		-	*£42*	*£48*	*£52*	*£55+*
Physical Design		£34,000	£42,000	£52,000	£65,000	£75,000+
		-	*£40*	*£46*	*£50*	*£52+*
FPGA Design		£31,000	£40,000	£47,000	£60,000	£70,000+
		-	*£40*	*£48*	*£50*	*£52+*
Analog/Mixed Signal IC Design		£34,000	£42,000	£52,000	£65,000	£75,000+
		-	*£42*	*£48*	*£52*	*£55+*
RF IC Design		£37,000	£45,000	£57,000	£70,000	£85,000+
		-	*£42*	*£48*	*£52*	*£55+*
Analog / RF Layout		£30,000	£38,000	£41,000	£52,000	£60,000+
		-	*£40*	*£45*	*£50*	*£50+*
IC Test		£32,000	£38,000	£40,000	£45,000	£60,000+
		-	*£45*	*£50*	*£55*	*£60+*
IC Process		£32,000	£38,000	£40,000	£45,000	£60,000
		-	-	-	-	-

Opportunities – Week 5

DVclub

3310 W. Braker Lane
Building 2, Suite 300-301
Austin, TX 78758
978.824.1400
info@dvclub.org

HOME REGISTER SPONSORS PRESENTERS ABOUT CONTACT

Welcome to DVClub!

Our charter is to provide local education and networking events for the semiconductor verification and design community in North America. Founded by Eric Hennenhoefer in 2005, and now managed by Paradigm Works Engineers - Michael Hoyt, Jim Crocker, and Kelly Larson (Tesla). DVClub regularly holds events across North America including - San Jose, Austin, Boston, Portland, Toronto, RTP, and Fort Collins. If you are new to DVClub, please join DVClub or register for a specific event and follow us on Twitter for updates on upcoming events in your area!

Register For An Event | Join DVClub

DVClub Conference Schedule 2021 (Online)

May 5	Silicon Valley/LA/Portland	September 15	Austin/Ft Collins/MN
June 9	Austin/Ft. Collins/MN	October 20	North America
July 21	Toronto/Boston/RTP	November 17	Silicon Valley/LA/Portland
August 11	Silicon Valley/LA/Portland	December 8	Austin/Ft Collins/MN

Upcoming Events

Diamond Sponsors

arm

https://dvclub.org/

Opportunities – Week 5

DVClub

The Design and Verification Club (Europe and India)

The principal goal of each DVCLUB conference is the sharing of technical expertise amongst professionals while helping build the verification community through regular educational and networking events. DVClub membership is free and is open to all non-service provider semiconductor professionals.

DVClub currently has chapters in Austin, Bangalore, Boston, Bristol, Cambridge, Dallas, Delhi, Eindhoven, Grenoble, RTP, Shanghai, Silicon Valley, Sophia Antipolis and Toronto. T&VS organise events in Europe and India.

DVClub Europe

Forthcoming Events

DVClub Europe: Diary Dates

Using ML and AI in EDA Tools	Nov 23, 2021

Previous Events

Formal Verification Adoption Made Easy	Sep 07, 2021
Using Python in Verification	Apr 20, 2021
Accelerating your SoC Verification	Feb 23, 2021

DVClub India

Forthcoming and Previous Events

Safety Driven Verification	Sep, 2018
Unique Challenges in Verifying Automotive SoCs	Nov, 2017
Verification IPs – Build vs Buy	Apr, 2017
IP Verification Challenges	May, 2016
See all previous Events	

DVClub China

See previous Events

https://www.testandverification.com/conferences/dvclub

Opportunities – Week 5

eventbrite Search for events Browse Events Organise Help Create an event Sign In

NOV
23

DVClub Europe: Using ML and AI in EDA Tools

by Tessolve Semiconductor Ltd
34 followers Follow

Free

Register

Do not miss the opportunity to join us for interesting talks on industry specific topics

About this event

Using ML and AI in EDA Tools

- Four verification experts will spend 20 minutes each outlining tools and methodologies aimed at Using ML and AI in EDA Tools

Agenda (GMT)

Date and time

Tue, 23 November 2021
12:00 – 13:30 GMT
Add to calendar

Location

Almondsbury Interchange Hotel
Gloucester Road
Almondsbury
Bristol
BS32 4AA
View Map

https://www.testandverification.com/conferences/dvclub/europe/using-ml-and-ai-in-eda-tools

Opportunities – Week 5

eventbrite Search for events Browse Events Organise Help Create an event

NMI Meet and Greet Event

24th November

09:00 - 15:00

Holiday Inn, Newport

NOV
24

NMI Meet & Greet Event

by Angela Tansey

175 followers Follow

Free

Register

Bring R&D, Universities, Manufacturing and Suppliers together to discuss how emerging R&D capabilities bring new opportunities.

About this event

The purpose of the NMI Meet & Greet events is to bring R&D, Universities, Manufacturing and Suppliers together to discuss how emerging R&D capabilities bring new opportunities for collaboration in the future with the wider Semiconductor supply chain.

NMI existing members have the equipment, knowledge and expertise that supports both mass production and some of the smaller specialised FABS that are often located within the universities facilities.

Date and time

Wed, 24 November 2021
09:00 - 15:00 GMT
Add to calendar

Location

Holiday Inn Newport, an IHG Hotel
The Coldra
Newport
NP18 2YG
View Map

https://www.eventbrite.co.uk/e/nmi-meet-greet-event-tickets-192182562217

Questions

https://www.bristol.ac.uk/tsl