

1990/91

## Advanced Logic

For graduate students

Lectures

24 h

### *Goals:*

The purpose of the course is to develop an intuitive grasp of the main results of modern logic and to explain their relationship to computer science, particularly artificial intelligence and logic programming.

### *Prerequisites:*

TDDA 15 Logic I

### *Organization:*

12 two-hour lectures, twice a week over 6 weeks. Independent reading, approx. 4 hours each week.

### *Contents:*

- Löwenheim - Skolem Theorem
- Tarski semantics
- Skolem-Herbrand-Gödel Theorem
- Sequent calculus
- Gentzen's Theorem
- Church's Theorem
- Gödel's Incompleteness Theorem

### *Literature:*

E. Mendelson, Introduction to Mathematical Logic, Wadsworth 1987  
J Gallier, Logic for Computer Science, Wiley (paperback)

### *Teachers:*

Douglas Busch

### *Examiner:*

Douglas Busch

### *Schedule:*

November-December (period 2)

*Examination:*

A written report on a chosen topic.

*Credit:*

Up to 4 points.



1990/91

# Artificial Intelligence Paradigms

For graduate students

Lectures:

30 h

## *Goals:*

A long term goal in AI is the construction of autonomous agents capable of functioning successfully, with some “degree of intelligence”, in dynamic environments. There are a number of approaches currently being used to reach the long term goal, each with its own philosophical and cognitive presuppositions. An “AI paradigm” is the buzzword commonly used to characterize a particular approach. The choice of a paradigm determines to some extent what the important research issues are and the strategies used to find solutions to the problems associated with the issues. In the course, we will study the different AI paradigms and try to reach a consensus as to whether these are the only paradigms or whether there is such a thing as a distinct paradigm at all. We will consider how the AI discipline has developed historically and how it might develop in the future.

## *Prerequisites:*

Introduction to AI

Introduction to Logic

Knowledge Representation

AI-programming

## *Organization:*

A series of 2 hour seminars will be given, the first hour consisting of a video lecture of a prominent AI researcher and the second hour open to discussion. A series of articles dealing with the basic theme of each video will be used as a basis for the discussion along with the contents of the video. We will conclude the seminar with a panel debate concerning some of the issues discussed during the seminars.

## *Contents:*

(The same as the Goals).

## *Literature:*

A series of articles to be handed out.

Several books, copies of which will be accessible to the students during the course.

## *Teachers:*

Patrick Doherty, Dimiter Driankov

## *Examiner:*

Dimiter Driankov

## *Schedule:*

September - November. Two seminars a week.

*Examination:*

75% participation at the seminars with active input in the discussions

A report which integrates several of the articles with the contents from the tapes and discussions.

*Credit:*

4 points for 75% participation and a written report

1 point for just a written report

Those students who attended the “Advanced AI Course” last year are encouraged to attend this course for two reasons:

- A major part of the material, including the videos, is new;
- Contributions to the discussion from more experienced participants are appreciated

Credit for active participation will in this case be decided after negotiation with the student's main supervisor (1 point recommended).



1990/91

## Compiler Construction - Advanced Course

For graduate students	Part 1, Lectures	8 h
	Part 2, Lectures	30 h
	Project lab work	10-20 h

### *Goals:*

The goal of this course is to give the student a thorough insight in compiler construction. The course includes topics not included in undergraduate courses or advanced research topics.

### *Prerequisites:*

The undergraduate compiler courses Compiler Construction (TDDA37) or Compilers and Interpreters (TDDA28), or equivalent. Pteori I is also recommended but not required.

### *Contents:*

Part 1:

- **Attribute grammars, Denotational semantics**  
(Only for students who have not taken Pteori I (TDDA43) or equivalent)

Part 2:

- **Code generation:**  
Code generator generators, register allocation, register colouring, code generation by pattern matching
- **Data flow analysis:**  
Basic blocks, flow graphs, Intraprocedural data flow analysis, Interprocedural data flow analysis, reducible flow graphs, iterative data flow algorithms, monotone data flow analysis frameworks
- **Optimization:**  
Code-improving transformations, Alias analysis, Peephole optimization
- **Compilation methods for Very High Level Languages:**  
Data type transformations, Finite differencing
- **Incremental Compilation Techniques**
- **Compiler Generation Techniques:**
- **Project: Practical programming exercises on solving data flow analysis equations**

### *Literature:*

Aho, Sethi, Ullman: Compilers - Principles, Techniques and Tools, Addison-Wesley  
(M.S. Hecht: Flow Analysis of Computer Programs, North-Holland)  
Articles and reports.

### *Teachers:*

Course leader, lectures: Peter Fritzon  
Invited lecturer: Arthur Krepski

*Schedule:*

March - April

*Examiner:*

Peter Fritzson

*Examination:*

Part 1: written examination.

Part 2: exercises and written examination, or alternatively: exercises and a written summary of the course material.

*Credit:*

Part 1 gives 1 point for students who have not taken Pteori I or equivalent.

Part 2 gives 4 points for the theoretical part, and 1-2 points for the project part.





1990/91

# Compiling for Parallelism

For graduate students

Lectures

24 h

## *Goals:*

The goal of this course is to give the student insight in the area of compilation for parallel and vector computers.

## *Prerequisites:*

The undergraduate compiler courses Compiler Construction (TDDA37) or Compilers and Interpreters (TDDA28), or equivalent.

## *Contents:*

- **Parallel architectures and compilers**
- **Program restructuring for parallel execution**
- **Automatic packaging and scheduling of parallelism**
- **Static and dynamic loop scheduling**
- **Run-time overhead**
- **Static program partitioning**
- **Static task scheduling**
- **Compiling for massively parallel machines**
- **The paralation model**

## *Literature:*

Constantine Polychronopoulos: *Parallel Programming and Compilers*, Kluwer Academic Publishers

Gary Sabot: The Paralation Model (small excerpts)

## *Teachers:*

Lectures: Peter Fritzson, Krzysztof Kuchcinski, Johan Fagerström

## *Examiner:*

Peter Fritzson

## *Schedule:*

December - January. Intensive course in January.

## *Examination:*

Exercises and/or written examination.

## *Credit:*

3 points





1990/91

# Computational Geometry

For graduate students

Lectures

20 h

## *Goals:*

Computational geometry is concerned with computational aspects of geometric problems. It has applications in robotics, computer graphics, computer aided design etc. The course will present a collection of problems and methods central to computational geometry.

## *Prerequisites:*

A course on design and analysis of algorithms.

## *Organization:*

The course consists of ten lectures. But for attending the lectures, students are obliged to solve a number of problems distributed as homework.

## *Contents:*

The following topics will be covered: basic data structures, geometric searching, convex hulls, Voronoi diagrams and their applications, intersections, and numerical stability of geometric computations

## *Literature:*

Computational Geometry, An Introduction by F. P. Preparata and M.I. Shamos, Texts and Monographs in Computer Science, Springer Verlag, New York.

## *Teachers:*

Per-Olof Fjällström, Jyrki Katajainen.

## *Examiner:*

Per-Olof Fjällström

## *Schedule:*

November - December (two times per week)

## *Examination:*

To each of the topics of the course, there will be a set of obligatory exercises. Each student must submit written solutions to these exercises as well as be prepared to orally present his/hers solutions.

## *Credit:*

3 points.

# Construction of Correct Programs

For graduate students

Lectures

24 h

## *Goals:*

The students should learn how to construct smallish programs together with a formal correctness proof. This will give insight into methods of constructing larger-scale software without full formal proofs, but in a way that will drastically reduce the time spent on “debugging”.

## *Prerequisites:*

Elementary knowledge of predicate calculus.

## *Organization:*

Lectures, seminars and discussions, homework assignments (including individual study of the literature).

## *Contents:*

- Programs as predicate transformers
- Formal derivation of small programs
- Principles of effective programming

## *Literature:*

- Dijkstra: A Discipline of Programming
- Gries: The Science of Programming
- Several other books and articles

## *Teachers:*

Feliks Kluzniak

## *Examiner:*

Jan Maluszynski

## *Schedule:*

November - December 1990

## *Examination:*

Active participation in seminars  
An individual project

## *Credit:*

3 points







1990/91

## Document Description and Architecture - Theories and Standards

For graduate students

Lectures

24 h

### *Goals:*

To give an understanding of the issues in generating and describing “documents”: viz. composition and manipulation, presentation, retrieval, and the current approaches to solving them. A thorough review of the current standards and ongoing standardization work that directly or indirectly are concerned with document description and architecture will be the main base for the course.

### *Prerequisites:*

None.

### *Organization:*

12 two-hour-lectures discussing material previously assigned for study

### *Contents:*

- Introduction: editors, documents
- Document dimensions: content, logical structure, layout
- Document models, markup and parsing
- Description of content, logical structure, layout and style
- Document interchange

### *Literature:*

- SGML & ODA (DS/INF 48),

### **Relevant Parts of:**

- ISO-8879: Information processing - Text and office systems - Standard Generalized Markup Language (SGML)
- ISO-9069: Information processing - SGML support facilities - SGML Document Interchange Format (SDIF)
- ISO/IEC TR 9573: Information processing - SGML support facilities - Techniques for using SGML
- ISO-8613: Information processing - Text and office systems - Office Document Architecture (ODA) and Interchange Format. Pt. 1 -8.
- ISO-9735: Electronic Data Interchange for Administration, Commerce and Trade (EDIFACT) - Application Level Syntax Rules
- ISO/IEC DIS 10021: Information Processing Systems - Text Communication - Message Oriented Text Interchange System (MOTIS) (CCITT X.400: Message Handling Systems)
- Articles and reports handed out during the course

*Teachers:*

Roland Hjerppe, invited lecturers

*Examiner:*

Roland Hjerppe

*Schedule:*

January - April

*Examination:*

A written report on a chosen topic

*Credit:*

3 points



1990/91

## Empirical Research Methods

For graduate students

Lectures

24 h

### *Goals:*

To provide a basic knowledge of quantitative and qualitative empirical research methods.

### *Prerequisites:*

Basic statistics (only a short review will be given during the course).

### *Organization:*

The course will be given as lectures, group seminars, and obligatory homework.

### *Contents:*

- Introduction
- Instrument, data collection and analysis: a quantitative study
- Grounded theory: fundamentals of a qualitative study
- Methodological triangulation
- How to critically read a scientific paper

### *Literature:*

Copies of relevant material on quantitative methods will be distributed.  
Patton MQ. Qualitative evaluation methods. Sage Publ., London, 1980.

### *Teachers:*

Toomas Timpka

### *Examiner:*

Toomas Timpka

### *Schedule:*

April - June 1991.

### *Examination:*

To write a term paper. To implement an empirical study.

### *Credit:*

3 points





1990/91

## Epistemic Logics

For graduate students

Lectures

24 h

### *Goals:*

The purpose of this course is to survey the current applications of techniques from modal logic to knowledge representation within AI, including the use of auto-epistemic logic in representing NMR. This will also be compared with non-modal approaches using classical logic.

### *Prerequisites:*

TDDA 15 Logic I

TDFR 16 Modal Logic

(TDDA 16 AI kunskapsrepresentation - helpful but not required)

### *Organization:*

12 two-hour lectures, twice a week over 6 weeks

Independent reading, approx 4-hours each week.

### *Contents:*

A study of various articles.

### *Literature:*

J. Halpern (ed.) Theoretical Aspects of Reasoning about Knowledge, Morgan Kaufmann 1986  
(plus texts for Modal Logic).

### *Teachers:*

Douglas Busch

### *Examiner:*

Douglas Busch

### *Schedule:*

April - May (Period 4)

### *Examination:*

A written report on a chosen topic.

### *Credit:*

Up to 4 points.







1990/91

## Hardware Verification

For graduate students

Lectures

20h

### *Goals:*

The goal of the course is to give students a basic knowledge about different hardware description and verification methods and the CIRCAL calculus.

### *Prerequisites:*

Understanding of basic hardware design alternatively a documented interest in the specification and analysis of parallel systems or formalisms such as CCS.

### *Organization:*

The course will consist of lectures (possibly also some labs or demos on tools provided by Milne).

### *Contents:*

1. Introduction to formal description and verification methods.
2. CIRCAL (CIRcuit CALculus) and its relation to HOL (Higher Order Logic).
3. Relation to VHDL language
4. Relation to simulation methods

### *Literature:*

Textbook on CIRCAL by G Milne (draft).

### *Teacher:*

George Milne , University of Strathclyde.

### *Examiner:*

Krzysztof Kuchcinski

### *Schedule:*

Middle to end of August.

### *Examination:*

Obligatory homework (possibly related to labs).

### *Credit:*

3-4 points.





1990/91

## **Hypermedia: History, concepts and applications**

For graduate students

Lectures

24 h

### *Goals:*

To give a basic knowledge of hypermedia systems and their applications.

### *Organization:*

The course will consist of lectures, seminars given by the participants, and obligatory homework. It is a continuation of the course given 1989-90, but it can be followed independently as well.

### *Contents:*

- Introduction
- The history of hypermedia systems
- Concepts and definitions: text, graphics, sound and animation
- Hypermedia interfaces
- Object-oriented databases
- Interactive video
- Review of existing systems
- Applications in work-life and education

### *Literature:*

Course material will be distributed in due time before the first meeting

### *Teachers:*

Toomas Timpka. Guest lecturers will be invited.

### *Examiner:*

Toomas Timpka.

### *Schedule:*

March-June 1991.

### *Examination:*

To write a term paper and/or to implement a project.

### *Credit:*

3 points





1990/91

# Knowledge Based Planning

For graduate students	Lectures	36 h
	Practical lab work	4 h

## Goals:

The aim of the course is to study fundamentals of AI-planning. The course will cover the following issues:

- The ontology and basic problems underlying planning
- The variety of different approaches to plan representation and generation
- The theories and techniques used in practical planning systems
- Related issues and recent developments

## Prerequisites:

TDDA 15, Logic, undergraduate course

TDDA 16, AI and Knowledge Representation, undergraduate course

TDFR 13, Temporal Logics, postgraduate course

**Organization:** The course will consist of lectures, article presentations and laboratory work. The lectures will be given by the course organizers and selected articles will be presented by the course participants. The laboratory work involves implementing the backtracking procedure for STRIPS in Common Lisp.

## Contents:

### *Representing and Reasoning about Time and Action:*

- Situation Calculus and the Frame Problem (McCarthy, Hayes)
- STRIPS-representation (Genesereth and Nilsson ch. 11, Lifschitz)
- The prediction, the ramification and the qualification problems
- Temporal logics (Shoham 88, Sandewall 88-89)
- Possible Worlds Approach (Ginsberg and Smith 86)

### *Plan generation methods (formal and informal):*

- Linear and non-linear planning
- Hierarchical and non-hierarchical planning
- SIPE (Wilkins)
- Devisor (Vere)
- Constraint-Posting Approach (TWEAK)
- Sandewall's Decision Procedure (Sandewall 88)

### *Plan selection and scheduling (or plan coordination)*

- Reasoning about plans and goals

### *Reactive and real-time planning*

### *Plan revision*

### *Plan execution and monitoring*

### *Multi-agent planning*

### *Plan Recognition*

## Literature:

1. Genesereth, M. R., Nilsson, N. J., Logical Foundation of Artificial Intelligence, Morgan Kaufmann, Chapters 11 and 12.
2. Nilsson, N. J., Principles of Artificial Intelligence, Chapters 7 and 8.
3. Wilkins, D., Practical Planning.



4. Charniak, E., McDermott, D., Introduction to Artificial Intelligence, Addison Wesley, Chapter 9.
5. A selection of articles to be announced later

*Teachers:* Jalal Maleki, Dimiter Driankov

*Examiner:*

Dimiter Driankov

*Schedule:*

April - June 1991. Twice a week.

*Examination:*

- Solving assignments
- Completing the laboratory work (not applicable to those who have done this earlier)
- Presentation of an article
- A term report

*Credit:* 5 points



# Methodology of Research in Computer Science

For graduate students

Lectures

16 h

Seminars

10-15 h

## *Goals:*

The goal of this course is to give the student an understanding for methodological problems in the process of performing research in general and in computer science in particular. The course also includes practical issues in connection with selecting a research problem, planning a project, carrying out the research and reporting the results.

## *Prerequisites:*

None.

## *Organization:*

The course will have the character of an advanced seminar with presentations on selected topics, mainly by invited lecturers. Some lectures will be used for a review of more practical issues in planning and carrying out research. There will also be a number of more discussion-oriented seminars, where specific papers or issues are treated. An additional 3 points may be obtained for writing an essay on a topic suggested or approved by the course leader

## *Contents:*

- Introduction to the theory of science
- Scientific paradigms
- Project conception and planning
- Scientific writing and publication
- Seminars with discussion of papers and dissertations from a methodological point of view

## *Literature:*

To be decided later.

Articles and reports

## *Teachers:*

Sture Hägglund, Roland Hjerpe, Erik Sandewall, invited lecturers.

## *Examiner:*

Sture Hägglund

## *Schedule:*

September - December, 1992

## *Examination:*

Active participation in lectures and seminars. A written summary of course material and other completed assignments.

## *Credit:*

3 points (+3 points)





1990/91

## Modal Logic

For graduate students

Lectures

24 h

### *Goals:*

The purpose of the course is to study the development of modern modal logic and to explain the Kripke semantics, partly for its own sake and partly as a preparation for further study in temporal logic, conditional logic, epistemic logic and auto-epistemic logic.

### *Prerequisites:*

TDDA15 Logic I

(TDDA 16 AI Kunskapsrepresentation - helpful but not required)

### *Organization:*

12 two-hour lectures, twice a week over 6 weeks. Independent reading, approx. 4 hours each week.

### *Contents:*

History of modal logic, connections with conditional, relevant, intuitionistic and other “deviant” logics.

Applications in AI to knowledge representation and non-monotonic logic.

Comparative presentation of the main systems K, T, S4, S5.

Kripke semantics and completeness theorems.

### *Literature:*

R. Goldblatt, Logics of Time and Computation, CSLI

J. van Bentham, A Manual of Intensional Logic (2nd Edition), CSLI

(OBS! These will also be used in the courses Epistemic Logic and Temporal Logic).

### *Teachers:*

Douglas Busch

### *Examiner:*

Douglas Busch

*Schedule:*

September-October (period 1)

*Examination:*

A written report on a chosen topic.

*Credit:*

Up to 4 points





1990/91

# Object-Oriented Programming and Design Methodology

For graduate students

Lectures

20 h

## Goals:

The goals of this course is to give graduate students a firm understanding for the object-oriented paradigm of programming, state of the art object-oriented programming languages and design methodology

## *Prerequisites:*

Formal specification, Abstract Data types, Programming Languages, Compiler Construction, Program Development Methodology and Project.

## *Organization:*

Introductory lectures, seminars with student presentations and projects.

## *Contents:*

- Object-oriented programming paradigm
  - Encapsulation, Inheritance
- Language constructs for object-oriented programming
  - Class and instance model
- Prototype model
  - Relation model
  - Visibility; abstract data types
  - Memory management
- Design problems and common mistakes
  - The use of inheritance; levels of reusability
  - Error handling
- Design methodology
  - Data flow modeling of a design
  - Design by contract
- Design for exchange, extension and evolution
  - Exchange; specification and implementation
  - Extension; inheritance
  - Evolution; abstraction and generalization

## *Literature:*

Meyer, Object-oriented Software Construction, Prentice-Hall,1989.

Shrive, Wegner (ed), Research Directions in Object-oriented Programming, MIT Press, 1987

*Teachers:*

Mikael Patel, Göran Rydqvist

*Examiner:*

Krzysztof Kuchcinski

*Schedule:*

September- November.

*Examination*

1. Written report and presentation. The report should be written as a high quality research report of about 10 typed pages. Additionally, the author is expected to analyse and form a critique of an other participant's report.
2. Discussion protocols. Participants alternate at taking the record or discussions. This preliminary protocol should be distributed to all participants in typed format before the next seminar. Participants are priviled to add/remove as they see fit. A final version of the discussion protocols should be produced at the end of the seminars.

*Credit:*

2 points for written report and presentation

2 points for design project

Maximum of 4 points.



1990/91

# Qualitative Reasoning

For graduate students

Lectures

16 h

## *Goals:*

The seminar will provide an introduction to the field and an overview of the existing approaches and their historical roots. It will also attempt a critical reflection of the applicability and inherent limitations of the current techniques, including mathematical arguments.

## *Prerequisites:*

Discrete Mathematics

Logic

Artificial Intelligence

Knowledge Representation

## *Organization:*

General Goals of Qualitative Physics:

- Simulation, Explanation, Mental Models

Fundamental concepts in qualitative physics:

- Structure behavior, function

Three QP modeling paradigms:

- Component-based approach

- Process-based approach

- Constraint-based approach

Basic issues in QP modeling:

- Time

- Space

- Causality

Deeper issues in QP modeling:

- Gold Standards

- Ambiguity and Assumption Management

- Domain dependencies: simple physics, kinematics, electronics, medicine

- Task dependencies: simulation, diagnosis, design, tutoring

## *Contents:*

This course will provide a basic introduction to Qualitative Physics (QP) along with an appreciation for the good, the bad, and the ugly sides of qualitative modeling and simulation. We will scrutinize contemporary QP research with an eye towards the general modeling problems that make this field interesting, challenging, and controversial.

*Literature:*

Readings in Qualitative Reasoning About Physical Systems by Daniel Weld and Johan de Kleer;  
Morgan Kaufmann, 1990.

*Teachers:*

Keith Downing

*Examiner:*

Erik Sandewall

*Schedule:*

January - February

*Examination:*

Considerable reading and a few writing assignments.

*Credit:*

2 points



1990/91

## Reason Maintenance Systems

For graduate students

Lectures

16 h

### *Goals:*

Systematic knowledge of the concepts, foundations, and algorithms of truth maintenance systems, as well as an overview of their applications.

### *Prerequisites:*

Logic

AI Knowledge Representation

AI Programming

(Nonmonotonic Logic recommended but not necessary)

### *Organization:*

Intensive course with 8 lectures, given during 2 weeks. Some simple reason maintenance systems will be made available during the course for experimentation.

### *Contents:*

- Introduction
- Dependency Networks and Admissible States
- Dependency Network Management
- Consistency Maintenance
- TMS-Based Problem Solvers
- Other Approaches
- Applications
- Advance Topics (logical foundations, unified theories, control issues)

### *Literature:*

Michael Reinfrank: Fundamentals of Truth Maintenance, Lecture Notes Siemens Report INF2 ARM 5-88, revised and extended version, August 1989

Kenneth Forbus and Johan de Kleer: IJCAI-89 Tutorial on Truth Maintenance Systems. AAAI, 1989.

### *Teachers:*

Michael Reinfrank

*Examiner:*  
Michael Reinfrank

*Schedule:*  
February

*Examination:*  
An optional short term paper after the course gives up to 3 points

*Credit:*  
Up to 3 points





1990/91

## Types and Proofs

For graduate students

Lectures

24 h

### *Goals:*

The course gives an introduction to type theory and attempts to link it to computer science. In particular, the idea of extracting programs from proofs should be discussed in some detail.

### *Prerequisites:*

Some basic knowledge of logic.

### *Organization:*

Lectures and seminars.

### *Contents:*

- Semantics through proofs
- Relating proofs and functional programs
- Sequent Calculus and Prolog
- Linear Logic
- Basic concepts of Martin L of's theory

### *Literature:*

J.Y. Girard et al: Proofs and Types, Cambridge University Press, 1989. B. Nordstr om, K. Petersson, J. Smith: An Introduction to Martin-L of's Type Theory.

### *Teachers:*

Jan Maluszynski, guest speakers.

### *Examiner:*

Jan Maluszynski

### *Schedule:*

March through May, 2 h per week.

### *Examination:*

To each of the topics of the course, there will be a set of obligatory exercises. Each student must submit written solutions to these exercises as well as be prepared to orally present his/hers solutions

### *Credit:*

3 points.





1990/91

# Hardware Description Methods and the VHDL Language

For graduate students

Lectures

20h

## *Goals:*

The goal of the course is to give students a basic knowledge about different hardware description languages used by CAD-tools. Special emphasis will be put on the VHDL language (Very high speed integrated circuits Hardware Description Language) because it is an emerging standard for hardware description languages.

## *Prerequisites:*

Basic knowledge in programming languages (e.g. Pascal or Ada language).  
Understanding of basic computer organization and logic design.

## *Organization:*

The course will consist of lectures and seminars (possibly also some labs with a VHDL simulator).

## *Contents:*

Introduction, motivation and basic definitions.  
Review of selected hardware description languages  
Basic features of VHDL  
Basic modeling techniques  
Chip-level modelling  
System modeling  
Other modeling issues  
Examples of CAD tools built around VHDL

## *Literature:*

James R. Armstrong "Chip-Level Modelling with VHDL", Prentice-Hall, Inc., 1989.  
"VHDL Tutorial for IEEE Standard 1076 VHDL" by CAD Language Systems, Inc.

## *Teachers:*

Krzysztof Kuchcinski, Zebo Peng

## *Examiner:*

Krzysztof Kuchcinski

## *Schedule:*

January - March

## *Examination:*

Three obligatory exercises will be distributed. Every solution submitted by students should include a VHDL description of a chosen hardware and its simulation results.

## *Credit:*

Three points will be given based on the homework. An additional point can be given for students interested in writing and presenting a report on a selected subject. The presentation will be made during the course.



