UNIVERSITY OF GOTHENBURG

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DIT051 Computer Architecture, 7.5 credits

Computer Architecture, 7,5 högskolepoäng

Second Cycle

Confirmation

This course syllabus was confirmed by The IT Faculty Board on 2006-11-17 and was last revised on 2018-02-09 by Department of Computer Science and Engineering to be valid from 2018-08-19, autumn semester of 2018.

Field of education: Science 100%
Department: Department of Computer Science and Engineering

Position in the educational system

The course is part of the Computer Science Master´s programme and an elective course at University of Gothenburg.

The course can be part of the following programmes: 1) Computer Science, Master's Programme (N2COS), 2) Applied Data Science Master's Programme (N2ADS), 3) Software Engineering and Management, Master´s Programme (N2SEM), 4) Game Design & Technology Master's Programme (N2GDT), 5) Software Engineering Master's Programme (N2SOM), 6) Computer Science, Bachelor´s Programme (N1COS) and 7) No translation available (NDATM)

Main field of studies	*Specialization*
Computer Science	A1N, Second cycle, has only first-cycle course/s as entry requirements

Entry requirements

The requirement for the course is to have successfully completed two years of studies within Computer Science or equivalent. The course DIT122 Computer system engineering (Datorsystemteknik) or equivalent is required.

Applicants must prove knowledge of English: English 6/English B or the equivalent level of an internationally recognized test, for example TOEFL, IELTS.

Learning outcomes

After completion of the course the student should be able to:

Knowledge and understanding

- master the terminology and key concepts in computer architecture in order to follow the research advances in this field.

Skills and abilities

- state and describe software performance for enhancement tecniques: especially advanced pipelining techniques that can execute multiple instructions in parallel,
- state and describe the principles behind modern memory hierarchies and storage systems.

Judgement and approach

- quantitatively assess the impact of an architectural technique on the performance of application software using state-of-the-art simulation tools.

Course content

The course covers architectural techniques essential for achieving a high performance for application software. It also covers simulation-based analysis methods for quantitative assessment of the impact a certain architectural technique has on performance. The content is divided into the following parts:

1. The first part covers trends that affect the evolution of computer technology including Moores law, metrics of performance (execution time versus throughput), benchmarking as well as fundamentals of computer performance such as Amdahls law and locality of reference.
2. The second part covers instruction-set architecture (ISA) design and covers different ISA models and their tradeoffs with respect to compiler design and architecture efficiency. It also illustrates how quantitative assessments can guide the definition of an ISA.
3. The third part covers various techniques for exploitation of instruction level parallelism (ILP) by defining key concepts for what ILP is and what limits it. The techniques covered fall into two categories: dynamic and static techniques. The most important dynamic techniques covered are Tomasulos algorithm, branch prediction, and speculation. The most important static techniques are loop unrolling, software pipelining, LCD test, trace scheduling, and predicated instructions.
4. The fourth part deals with memory hierarchies. This part covers techniques to attack the different sources of performance bottlenecks in the memory hierarchy

such as techniques to reduce the miss rate, the miss penalty, and the hit time. Example techniques covered are victim caches, lockup-free caches, prefetching, virtually addressed caches. Also memory technology is covered in this part.

5. The fifth part covers storage systems and one example technology that is discussed is RAID.
6. The sixth part provides an overview of multiprocessor systems: a technology that lays the foundation for all microprocessors to be designed in the next decade.

Sub-courses

1. **Written exam** *(Tentamen),* 6 higher education credits
   Grading scale: Pass with Distinction (VG), Pass (G) and Fail (U)
2. **Project** *(Projekt),* 1.5 higher education credits
   Grading scale: Pass (G) and Fail (U)

Form of teaching

The course is organized into lectures, exercises, two laboratory tasks, and a mini research project assignment. An important methodology to systematically design computers is to assess the impact of an architectural technique on performance. This will be trained by a number of illustrative exercises as well as in labs and in the mini "research project assignment".

Language of instruction: English

Assessment

The student is examined by written Project report (1.5 hecs) individually or in group and a Written exam (6.0 hecs) done individually in an examination hall.

A student who has failed the same examination twice has the right to request of the department a change of examiner. The request is to be in writing and submitted as soon as possible. The department is to grant such a request withour undue delay.

In cases where a course has been discontinued or major changes have been made a student should be guaranteed at least three examination occasions (including the ordinary examination occasion) during a time of at least one year from the last time the course was given.

Grades

The grading scale comprises: Pass with Distinction (VG), Pass (G) and Fail (U).
In order to be granted Pass with Distinction (VG) as complete course grade, the candidate must have have passed the project report and haveat least 70% of the total number of credit points at the written exam. In order to be granted Pass (G) as complete

course grade, the candidate must have passed the project report and have at least 40% of the total number of credit points at the written exam. If the number of credit points on the written exam is less than 40% of the total number of the credit points, the mark is U.

Course evaluation

The course is evaluated through meetings both during and after the course between teachers and student representatives. Further, an anonymous questionnaire can be used to ensure written information. The outcome of the evaluations serves to improve the course by indicating which parts could be added, improved, changed or removed.