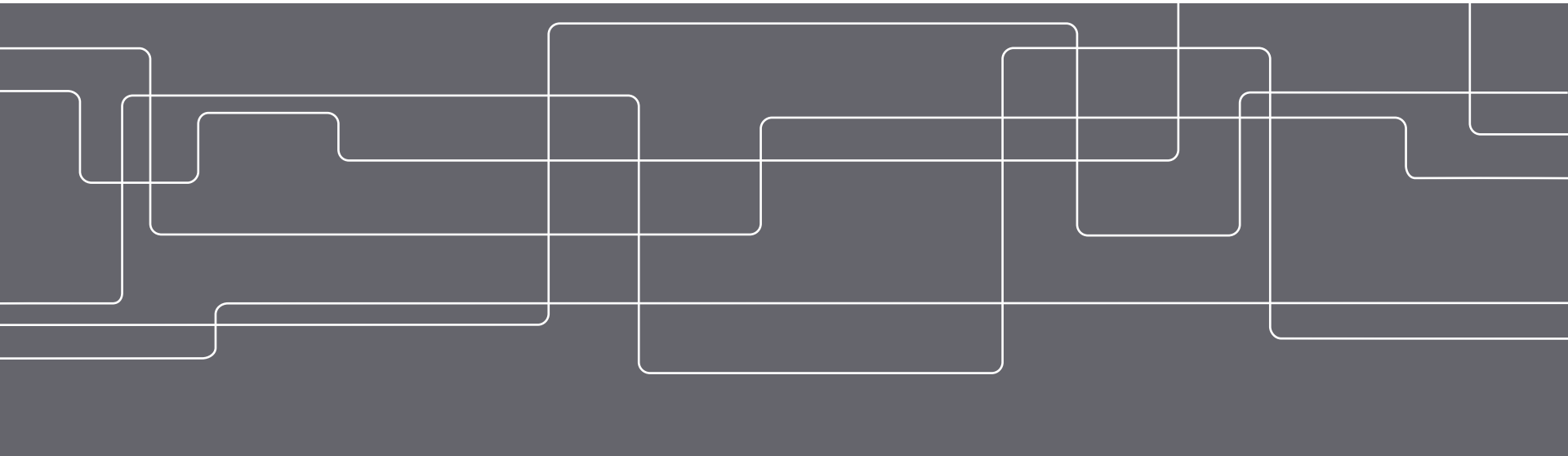
Computer Hardware Engineering (IS1200)
Computer Organization and Components (IS1500)

Spring 2021

Lecture 14: Course Summary and Project Awards

David Broman, Daniel Lundén, Artur Podobas

KTH Royal Institute of Technology



Course Structure

Module 1: C and Assembly Programming

LE1 → LE2 → LE3 → LE4 →

EX1 LAB1 S1 LAB2

Module 2: I/O Systems

LE5 → LE6 →

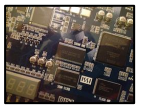
EX2 LAB3

Module 3: Logic Design **(IS1500 only)**

LE7 → LE8 →

PROJ START

EX3 LD-LAB



Module 4: Processor Design

LE9 → LE10 →

EX4 S2 LAB4

Module 5: Memory Hierarchy

LE11 →

EX5 S3

Module 6: Parallel Processors and Programs

LE12 → LE13 →

EX6 S4

Proj. Expo

LE14

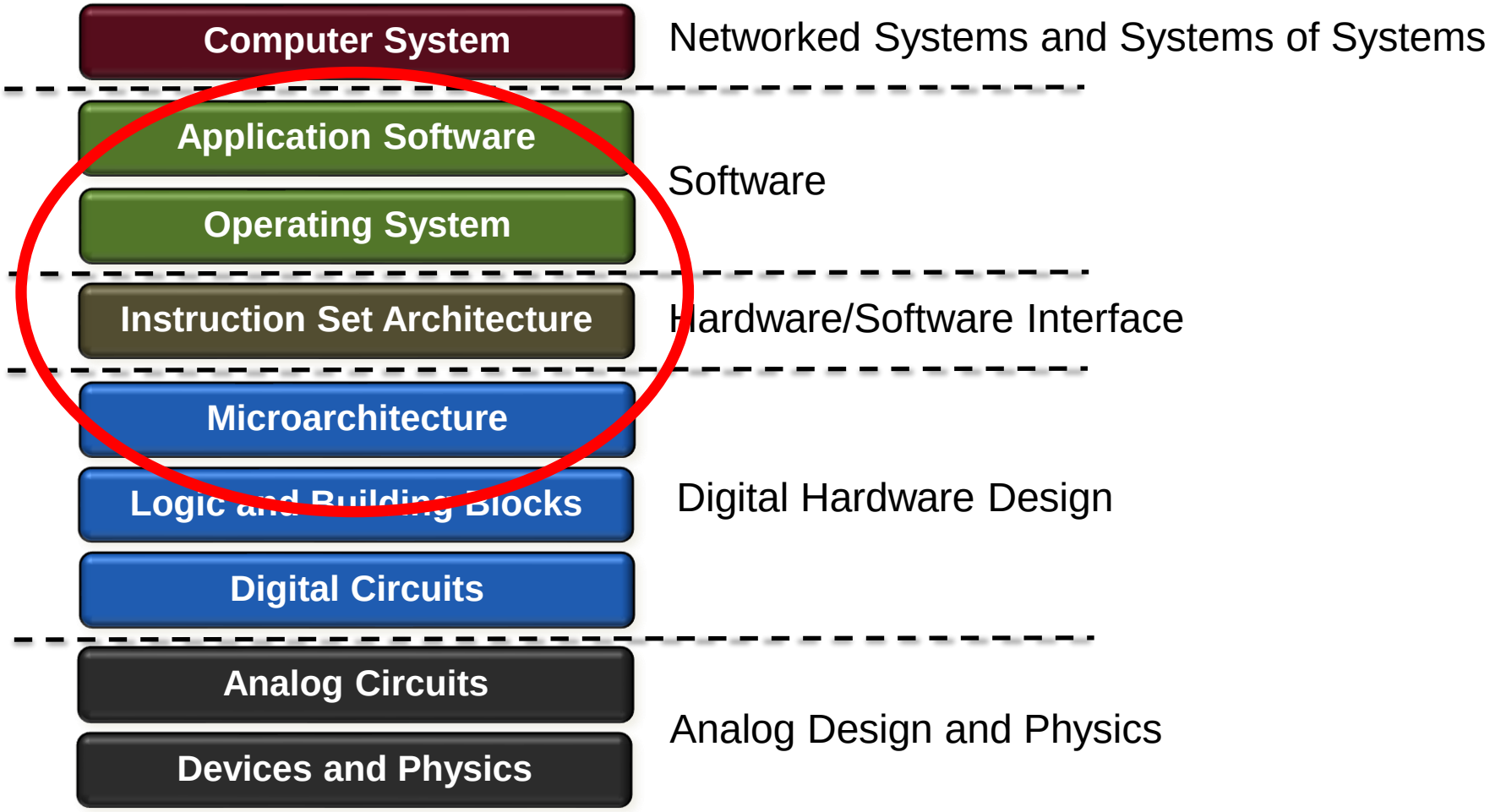
Artur Podobas
podobas@kth.se

Part I
Exam Structure and Grading

Part II
Study Advice

Part III and IV
Demo and Key Concepts

Abstractions in Computer Systems



Agenda

Part I

Exam Structure and Grading

Part II

Study Advice

Part III

Virtual Image Installation and Canvas Quiz Questions

Part IV

Key Concepts



Project and Awards!

Artur Podobas
podobas@kth.se

Part I

Exam Structure

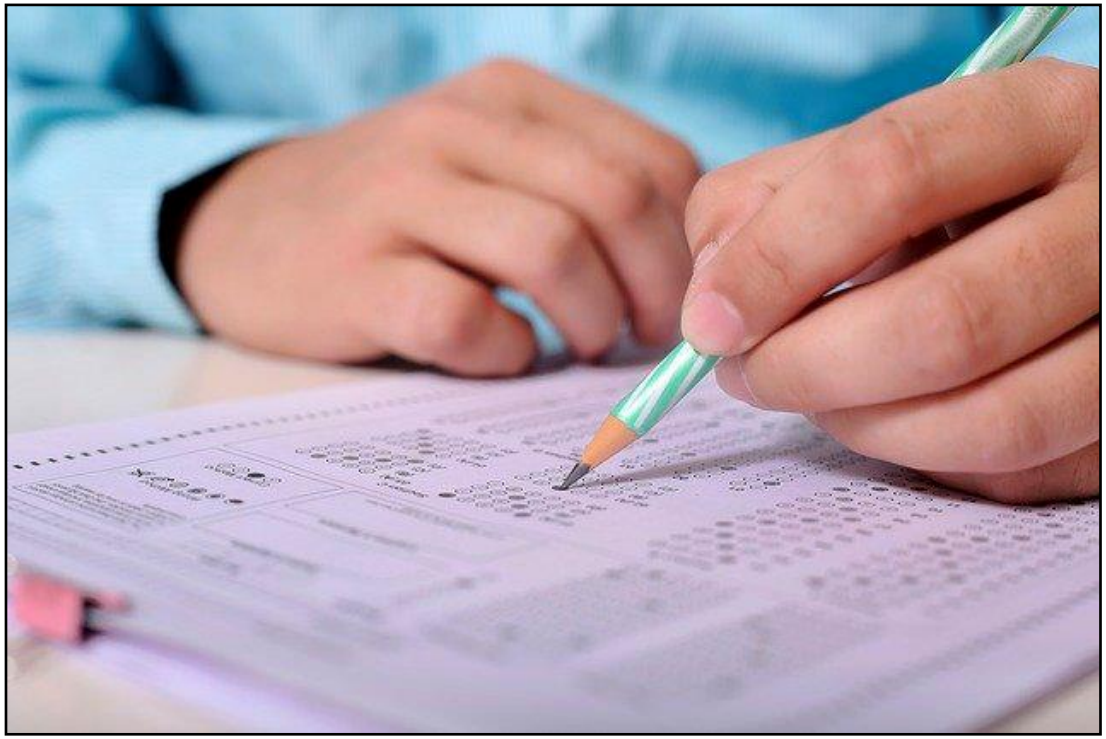
Examined Course Parts

1.5hp ANN1. LA-LAB, Logic Design
Grades P and F
(IS1500 only, not IS1200)

4.5hp LAB1. Labs 1,2,3,4 and Mini Project
Grades P and F

3hp TEN1. Exam (tenta)
Grades A, B, C, D, E, FX, F

Written Exam (Tenta)



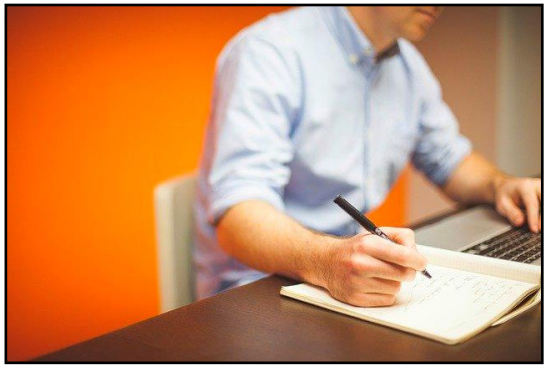
Due to the Covid situation, we will not have any ordinary written exams anymore

Artur Podobas
podobas@kth.se

Written Exam (Tenta)

Instead we do this **…almost**

Written Exam (Tenta)



Wrong view, you need to show the computer screen.

Correct view

Main points

- Computer based, using VirtualBox (VMWare is ok) with Ubuntu as the guest OS.
- Exam over Zoom
- Mobile camera, show computer. Be online on Zoom
- Same exam format (kinds of questions), but in Canvas. You can study old exams.
- Everyone gets different questions. Randomized within each module.
- You can solve solutions by hand (on paper) or on the computer (using gcc and MARS).
- Rationale: better examination (real programming), harder to cheat (individual exercises), faster to correct (parts automatic).
- You will get more exact information before the exam.

Written Exam - Grading Criteria

Written Exam (Tenta)

- March 18, 2021, (5h)
- Retake exams June 2021, January 2022
- Allowed aids: One sheet of handwritten A4 paper (both sides) with notes.

(Dry-run at March 16, 13:00-14:30)

The exam has two parts

- **Part I: Fundamentals**
  - Max 40 points.
  - 8 points for each of the 5 modules.
  - Short answers.
- **Part II: Advanced.** 3 questions:
  - 1. Discuss (Focus module 6)
  - 2. Construct (Focus modules 1, 2)
  - 3. Analyze (Focus modules 3, 4)

Criteria: Satisfactory (S), Good(G), Very Good (VG)

Grading of Exam

- To get a pass grade (A, B, C, D, or E), it is required to get at least 2 points on each module and in total 30 points on Part I (including bonus points).

Grading scale:

On part II:

- A: 3VG **or** 2VG & 1G.
- B: 1VG & 2G **or** 2VG & 1S.
- C: 3G **or** 2G & 1S **or** 1VG & 2S **or** 1VG & 1G & 1F **or** 2VG & 1F **or** 1VG & 1G & 1S
- D: 3 S **or** 1G & 1S & 1F **or** 2G & 1F **or** 1VG & 1S & 1F **or** 1VG & 2F **or** 1G & 2S
- E: No requirements on part II.
- FX: At least 30 points on Part I, and at most one module with less than 2 points. No req. on Part II.
- F: otherwise

To get A or B, an advanced project is also needed.

Part II

Study Advice

Some advices on how to study for the exam…

Lecture slides and videos

Go through all slides/videos very carefully. Make sure that you understand **every** key concept.

Read the books and check references

Read **selected parts** of the course book, according to the reading guidelines. Check both Patterson & Hennessy and Harris &Harris.

Exercises and old exams

Practice on the exercises and the last exams. Solve the exercises **yourself**. Look at the solution **after** you have made a solution.

A4 sheet and MIPS Sheet

Summarize. Make notes on formulas, structures, and key concepts. Make sure you understand every detail of the MIPS reference sheet.

Write questions and answers in Canvas

Ask questions and try to help your fellow students by making a Q&A in the discussion forum on Canvas!

Part III

Virtual Image Installation and Canvas Quiz Questions


Virtual Box Installation and Canvas Quiz Format
Demo

Exam is virtual

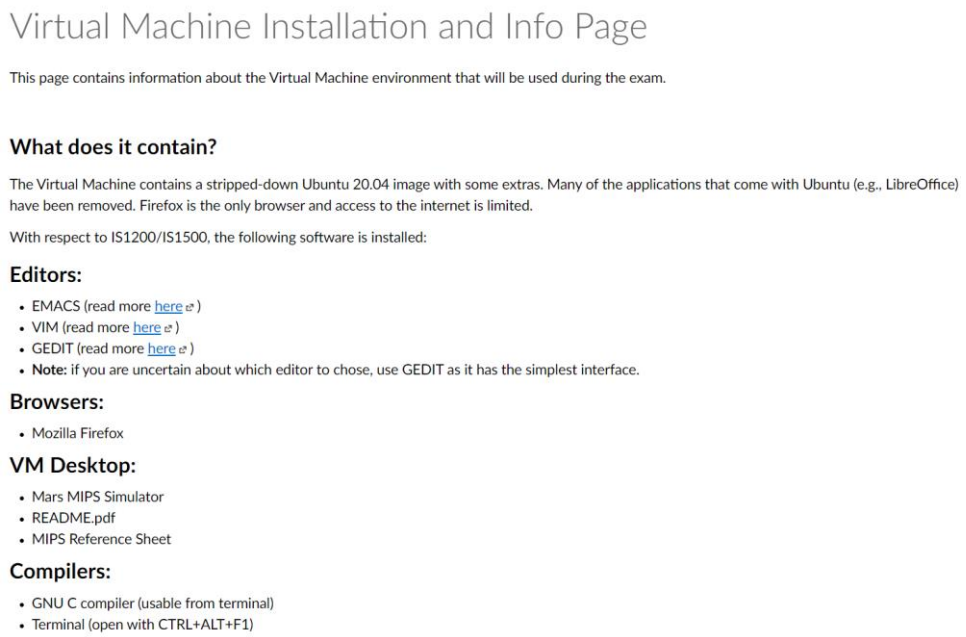
- Find virtual image at Canvas
- Installation information
- Frequency Asked Questions
- VMWare is OK

Demonstration

- Installation procedure

Start Installing and Familiarizing yourself with as soon as possible!

(Do not wait until the day before exam)



Virtual Machine Installation and Info Page

This page contains information about the Virtual Machine environment that will be used during the exam.

What does it contain?

The Virtual Machine contains a stripped-down Ubuntu 20.04 image with some extras. Many of the applications that come with Ubuntu (e.g., LibreOffice) have been removed. Firefox is the only browser and access to the internet is limited.

With respect to IS1200/IS1500, the following software is installed:

Editors:

- EMACS (read more here)
- VIM (read more here)
- GEDIT (read more here)
- **Note:** if you are uncertain about which editor to chose, use GEDIT as it has the simplest interface.

Browsers:

- Mozilla Firefox

VM Desktop:

- Mars MIPS Simulator
- README.pdf
- MIPS Reference Sheet

Compilers:

- GNU C compiler (usable from terminal)
- Terminal (open with CTRL+ALT+F1)

Part IV

Key Concepts


Module 1
C and Assembly Programming

Key Concepts

- Variables, if-statements, and expressions
- Loops (while and for)
- ISA
- Registers
- Instructions
- Translating from C to ASM and back
- Pointer and Arrays
- Two's complement and Sign extension
- Machine instruction encoding
- R, I, and J types
- Stack, Parameters and Arguments

Artur Podobas
podobas@kth.se

MIPS Reference Sheet

INSTSTRUCTION SET (SUBSET)

Name (format, op, funct)	Syntax	Operation
add (R,0,32)	`add rd,rs,rt`	reg(rd) := reg(rs) + reg(rt);
add immediate (I,8,na)	`addi rt,rs,imm`	reg(rt) := reg(rs) + signext(imm);
add immediate unsigned (I,9,na)	`addiu rt,rs,imm`	reg(rt) := reg(rs) + signext(imm);
add unsigned (R,0,33)	`addu rd,rs,rt`	reg(rd) := reg(rs) + reg(rt);
and (R,0,36)	`and rd,rs,rt`	reg(rd) := reg(rs) & reg(rt);
and immediate (I,12,na)	`andi rt,rs,imm`	reg(rt) := reg(rs) & zeroext(imm);
branch on equal (I,4,na)	`beq rs,rt,label`	if reg(rs) == reg(rt) then PC = BTA else NOP;
branch on not equal (I,5,na)	`bne rs,rt,label`	if reg(rs) != reg(rt) then PC = BTA else NOP;
jump and link register (R,0,9)	`jalr rs`	$ra := PC + 4; PC := reg(rs);
jump register (R,0,8)	`jr rs`	PC := reg(rs);
jump (J,2,na)	`j label`	PC := JTA;
jump and link (J,3,na)	`jal label`	$ra := PC + 4; PC := JTA;
load byte (I,32,na)	`lb rt,imm(rs)`	reg(rt) := signext(mem[reg(rs) + signext(imm)]$_{7:0}$);
load byte unsigned (I,36,na)	`lbu rt,imm(rs)`	reg(rt) := zeroext(mem[reg(rs) + signext(imm)]$_{7:0}$);
load upper immediate (I,14,na)	`lui rt,imm`	reg(rt) := concat(imm, 16 bits of 0);
load word (I,35,na)	`lw rt,imm(rs)`	reg(rt) := mem[reg(rs) + signext(imm)];
multiply, 32-bit result (R,28,2)	`mul rd,rs,rt`	reg(rd) := reg(rs) * reg(rt);
nor (R,0,39)	`nor rd,rs,rt`	reg(rd) := not(reg(rs) | reg(rt));
or (R,0,37)	`or rd,rs,rt`	reg(rd) := reg(rs) | reg(rt);
or immediate (I,13,na)	`ori rt,rs,imm`	reg(rt) := reg(rs) | zeroext(imm);
set less than (R,0,42)	`slt rd,rs,rt`	reg(rd) := if reg(rs) < reg(rt) then 1 else 0;
set less than unsigned (R,0,43)	`sltu rd,rs,rt`	reg(rd) := if reg(rs) < reg(rt) then 1 else 0;
set less than immediate (I,10,na)	`slti rt,rs,imm`	reg(rt) := if reg(rs) < signext(imm) then 1 else 0;
set less than immediate unsigned (I,11,na)	`sltiu rt,rs,imm`	reg(rt) := if reg(rs) < signext(imm) then 1 else 0;
shift left logical (R,0,0)	`sll rd,rt,shamt`	reg(rd) := reg(rt) << shamt;
shift left logical variable (R,0,4)	`sllv rd,rt,rs`	reg(rd) := reg(rt) << reg(rs$_{4:0}$);
shift right arithmetic (R,0,3)	`sra rd,rt,shamt`	reg(rd) := reg(rt) >>> shamt;
shift right logical (R,0,2)	`srl rd,rt,shamt`	reg(rd) := reg(rt) >> shamt;
shift right logical variable (R,0,6)	`srlv rd,rt,rs`	reg(rd) := reg(rt) >> reg(rs$_{4:0}$);
store byte (I,40,na)	`sb rt,imm(rs)`	mem[reg(rs) + signext(imm)]$_{7:0}$:= reg(rt)$_{7:0}$;
store word (I,43,na)	`sw rt,imm(rs)`	mem[reg(rs) + signext(imm)] := reg(rt);

REGISTERS

Name	Number	Description
$zero	0	constant value 0
$at	1	assembler temp
$v0	2	function return
$v1	3	function return
$a0	4	argument
$a1	5	argument
$a2	6	argument
$a3	7	argument
$t0	8	temporary value
$t1	9	temporary value
$t2	10	temporary value
$t3	11	temporary value
$t4	12	temporary value
$t5	13	temporary value
$t6	14	temporary value
$t7	15	temporary value
$s0	16	saved temporary
$s1	17	saved temporary
$s2	18	saved temporary
$s3	19	saved temporary
$s4	20	saved temporary
$s5	21	saved temporary
$s6	22	saved temporary
$s7	23	saved temporary
$t8	24	temporary value
$t9	25	temporary value
$k0	26	reserved for OS
$k1	27	reserved for OS
$gp	28	global pointer
$sp	29	stack pointer
$fp	30	frame pointer
$ra	31	return address

Artur Podobas
podobas@kth.se

Part I
Exam Structure and Grading

Part II
Study Advice

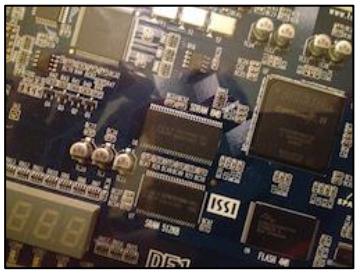
Part III and IV
Demo and Key Concepts

Module 2
I/O Systems

Key Concepts

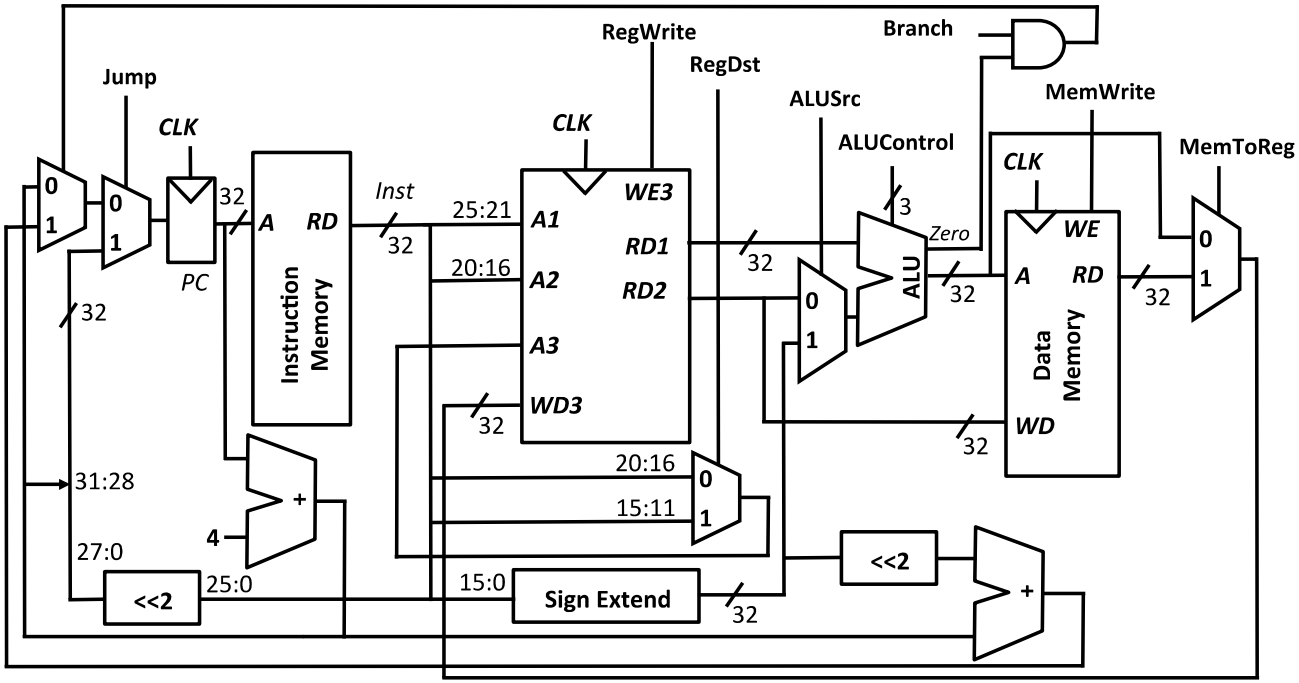
- Memory Mapped I/O
- Volatile keyword in C
- GPIO
- Timers
- Exceptions and Interrupts
- Buses (synchronous vs. asynchronous)
- DMA

Module 4
Processor Design



Key Concepts

- ALU
- Datapath
- Control Unit
- Single cycle processor (details of signals)
- Performance Analysis
- Critical Path
- CPI and IPC
- Pipelining (concept)
- Pipelined datapath
- Data Hazards
- Control Hazards
- Forwarding and Stalling


Jump
CLK
PC
32
A
RD
Instruction Memory
Inst
32
25:21
20:16
A1
A2
A3
WD3
WE3
RD1
RD2
CLK
RegWrite
RegDst
32
32
31:28
27:0
4
+
<<2
25:0
15:0
Sign Extend
32
20:16
15:11
0
1
ALUSrc
ALUControl
3
ALU
Zero
Branch
MemWrite
MemToReg
CLK
WE
A
RD
Data Memory
WD
32
32
32
<<2
+

Module 5
Memory Hierarchy

Key Concepts

- Temporal Locality
- Spatial Locality
- Miss and Hit Rate
- Capacity, Sets, Blocks, Associativity
- Direct Mapped Cache
- Instruction vs. Data Cache
- N-way Set Associative Cache
- Replacement Policy
- Write Policy
- Virtual Memory
- Page Table

Module 6
Parallel Processors and Programs

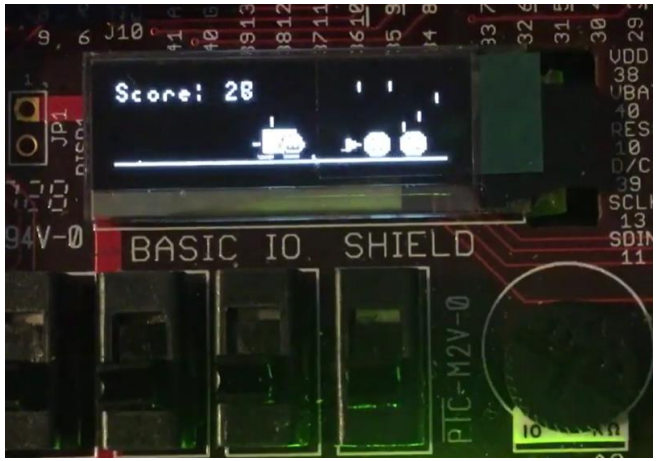
Key Concepts

- Moore's law
- Power Wall
- Parallelism vs. Concurrency
- Speedup
- Amdahl's law
- Data-level vs. task-level parallelism
- SISD, SIMD, MIMD
- Instruction-Level Parallelism (ILP)
- VLIW vs. Superscalar (basic idea)
- Hardware Multithreading
- Shared Memory Multiprocessor
- Cache Coherence and False Sharing
- Processors, Threads, and Multicore
- Semaphores
- Clusters and Message Passing

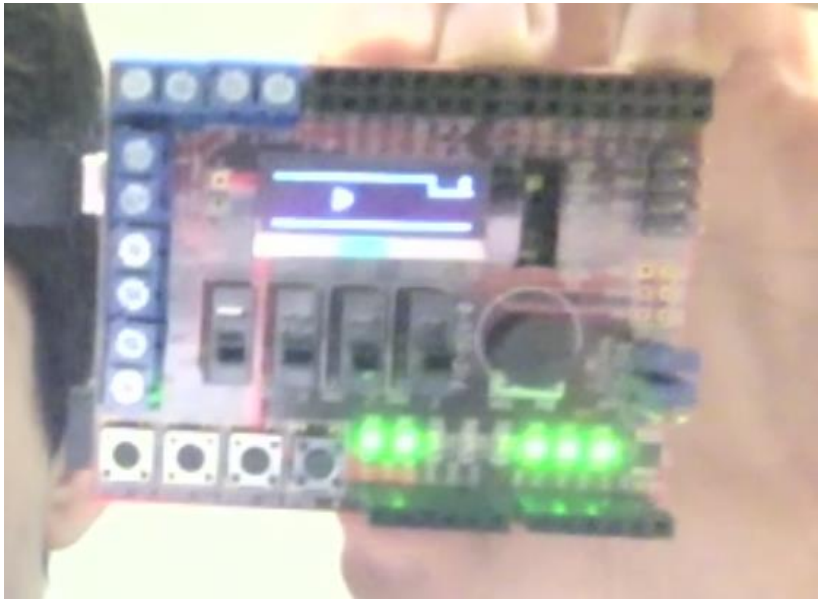
Project and Awards!

Category
“Greatest Legacy Game(s)”

Nyan Pong Ball!



Asteroid Shooter



Artur Podobas
podobas@kth.se

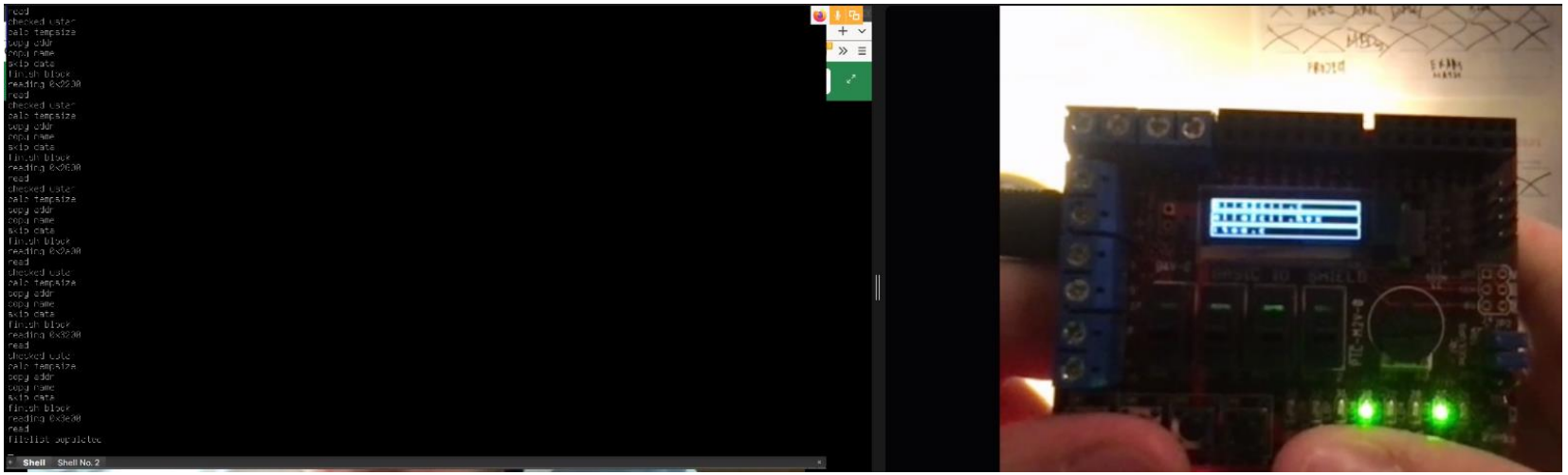
Part I
Exam Structure and Grading

Part II
Study Advice

Part III and IV
Demo and Key Concepts

Category
“Alternative Projects”

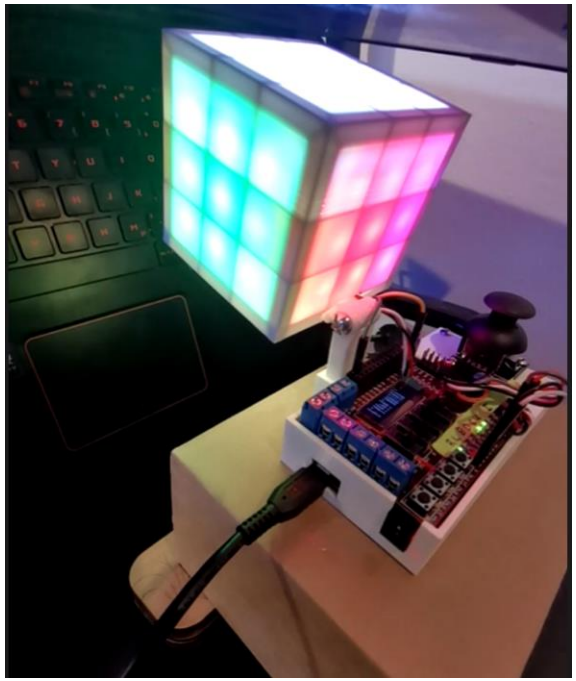
Bootloader on MIPS!



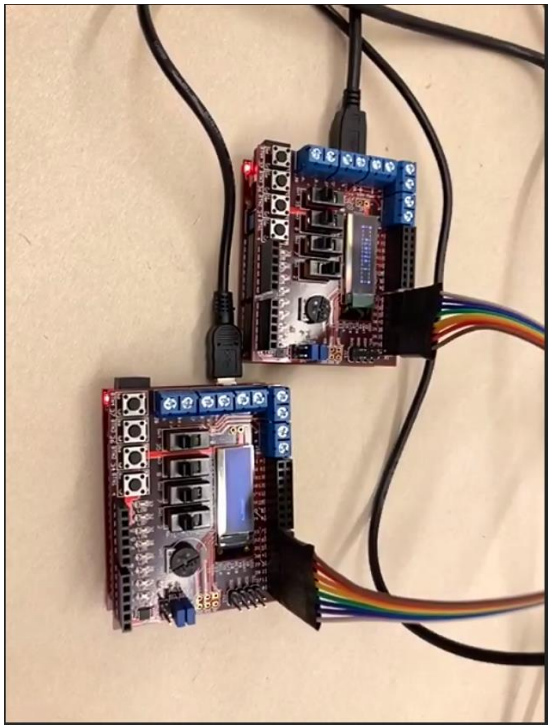
Artur Podobas
podobas@kth.se

Category
“Augmenting Reality”

Rubix Giant-Cube!



Multi-Snake



Artur Podobas
podobas@kth.se

Part I
Exam Structure and Grading

Part II
Study Advice

Part III and IV
Demo and Key Concepts

Summary

Some key take away points:

- **Computers** are fun!
- Please make sure to **study hard** for the exam.
- Deliver back the labkits as soon as possible.

Thank you and Good Luck!