Computer System Design CS401L

Jaynaryan T Tudu

Indian Institute of Technology Tirupati, India

2^{nd} August, 2023

Computer System Design

- It is a system!
- A system consists of more than one components with well defined functionality.
- Computer System is a set of components with well defined functionality.

Example of Systems

IIT Tirupati = { Department, Offices, People, Resources etc }
Human anatomy = { Organs, Tissues, Cells, Macro-molecules etc }
Universe = { Solar systems, Stars, Planets, Atoms, Quark etc }

A System could be a set of Systems!

Components of Computer Systems

Table : From physics to human being!

↓Human being
↓High-level Language/Apps ↑
↓Operating System ↑
↓Compiler ↑
↓Virtual Machine ↑
↓Assembler / Machine Language ↑
↓Architecture ↑
↓CPU | Memory System ↑
↓Boolean/Digital Logic ↑
↓CMOS Technology ↑
Device Physics ↑

Components of Computer Systems

The learning outcome:

- The main idea is to integrate all these system components and to make it work
- Some of the design and integration principles will be learnt
- Its an outcome based course where you have to design your own system and make it work.
- The course is targeted designing of the components: HACK to learn the principles, ARM, MIPS, and RISC-V to apply those principles.
- At the end of the course you will be trained to design a full-fledge computer systems by your own.

Lecture Plan

- Total 46 contact hrs
- We have alloted 40 hrs of lecture and 6 hrs for project discussion

Lecture 1: Introduction and organisational meeting (1:00 hr)

Lecture 2: Boolean Logic, Basic Gates and Hardware Descriptor Language (HDL - Hack HDL and Verilog HDL) (1:00 hr)

Lecture 3: Combinational Logic, Arithmetic Circuit Design and Simulation using a simulator (2:00 hrs)

Lecture 4: Sequential Design Concepts: Flip-flops, Register and Timing (1:00 hr)

Lecture 5: Memory System Design (Address, Data and Control) (2:00 hrs)

Lecture 6: HACK Machine Language (2:00 hrs)

Lecture Plan

Lecture 7	:	Specifying the Architecture Components: Single Cycle Processor (1:00 hr)
Lecture 8	:	Specifying the Architecture Components: Multi-cycle Processor (2:00 hrs)
Lecture 9	:	Implementation of processor and integration with memory system (2:00 hrs)
Lecture 10	:	Interfacing of input output system (Keyboard and Display screen)(1:00 hr)
Lecture 11	:	Assembler design: HACK Assembly Language Programming (1:00 hr)
Lecture 12	:	Assembler design: HACK Assembly to Binary Translation (2:00 hrs)

Lecture Plan

Lecture 13 : Virtual Machine (VM) stack model (1:00 hr)
Lecture 14 : Virtual Machine specification (1:00 hr)
Lecture 15 : Implementation of VM on HACK platform (2:00hrs)
Lecture 16 : VM Program Control: the Concept, specification and implementation (1:00 hr)
Lecture 17 : High-level Programming Language: Review of Concept (2:00hrs)
Lecture 18 : Jack: Specification of a programming language for HACK ISA(2:00hrs)
Lecture 19 : Jack standard library and applications (1:00 hr)

Lecture Plan

Lecture 20 :	Compiler concept review: syntax, semantic and code generation (2:00 hrs)
Lecture 21 :	Jack compiler specification for Syntax analysis(2:00 hrs)
Lecture 22 :	Implementation of syntax analyser(1:00 hr)
Lecture 23 :	Code generation concepts and specification (1:00 hr)
Lecture 24 :	Code generation Implementation(1:00 hr)
Lecture 25 :	Operating system concept review(1:00 hr)
Lecture 26 :	Specifying the operating system (2:00 hrs)
Lecture 27 :	Implementation of the operating system (2:00 hrs)
Lecture 28 :	System Integration (2:00 hrs)

System Design Project

- Majority of the learning in this course comes from the Project.
- this course is largely a "learning-by-doing"
- to learn How do you execute a system design project?
- to learn How to integrate a multiple components from different team members?

Project Execution Plan

Objective: to design a complete working computer system from the scratch.
- Everything has to be new and its your own!
- To earn super grade you need to show research skill set... kuch naya karna hai. kotta kawale!

System components	# of members
Multi-cycle and Pipeline Processor Design	3
Machine Language and Assembler Design	2
Virtual Machine Design	2
High-level Programming Language	2
Compiler Design	2
Operating System (Drivers and Management)	3
System Integration in FPGA board	2

Total Components = 7

Team size = 16

Project Execution Plan

Instruction set architecture (ISA) is the begining!

VM and Assembler $\leftarrow$ ISA $\rightarrow$ Processor design
Language and Compiler $\leftarrow$ VM
(OS is just an another program written in high-level Language)

Deciding on ISA (a suggestion)

Team 1: ARM instruction set architecture
Team 2: OpenMIPS instruction set architecture
Team 3: RISC-V instruction set architecture
Team 4: IBM Power instruction set architecture

ISA and Processor Design

- Once ISA is decided, the micro-architecture need to designed accordingly
- Make use of high-level hardware descriptor language (HDL) for describing the processor
- Example HDL: Verilog, VHDL, System Verilog, Chisel, Blue-spec
- For your project you may chose to use Verilog or VHDL!
- Tools for synthesis: YOSIS or ModelSIM. (other industry standard tools: DesignCompiler from Synopsys, RTL Compiler from Cadence). There are some cloud-based tool, you can explor them.
- In the course we will be using Nand2Tetris simulator

Memory System and I/O interfacing

In continuation from ISA and processor, the memory systems and I/O interfacing has to be designed.

- Memory systems need to be designed (detail will be discussed in lecture).
- Keyboard and Screen need to be connected to Processor.
- Memory mapped or I/O mapped can be used.

VM and Assembler

To design VM and Assembler, we will use the existing programming language such as C/C++[1].

- The VM specification has to be decided. Input: three-address code, Output: assembly codes for the specified ISA.
- Some of the example VM: Java VM, LLVM etc.

[1] Strictly speaking we are not designing them manually writing a machine level code from scratch.

Language, Compiler, OS and Applications

- The programming language and compiler go together!
- You have to have new programming language with minimum supported features.

- You will be using the existing high-level language C/C++ to design your programming language - designing compiler using C/C++.
- Once Programming language is done, you can use the programming language to write Operating system and other applications (this may be tough). Alternatively, you may use C/C++ to code OS and then compile it to the native ISA.

System Integration

Putting things together

- By now all the components must have been designed and tested for correctness
- The next important part is to integrate these.
- The final system will be built on FPGA board. You should be able to boot the OS, make keyboard and display work.

The end outcome will be a Computer System a user can edit a program using keyboard, compile it, execute it, and display results in screen.

Team Organisation

System components	# of members
Multi-cycle/Pipeline Processor Design	3
Machine Language and Assembler Design	2
Virtual Machine Design	2
High-level Programming Language	2
Compiler Design	2
Operating System (Drivers and Management)	3
System Integration in FPGA board	2

- Team formation: A team of 14 - 15 students has to be formed
- Task segregation: Task to be assigned to team members
- Every team should have a Team Leader and a Spoke person
- Team leader to ensure that the projects moves smoothly
- Spoke person to intimate the progress to Faculty/Co-ordinator

Team Organisation

Important

Each team member must be involved with encouraging team spirit!

Evaluation

Term Quiz:	20%
Class Test:	10%
Project:	40%
Final test:	30%

Evaluation

- Regular quiz: This will be an objective MCQ and Y/N type of question which will be conducted online.
- Midterm test: Two midterm tests.
- Project: The project will be having continuous progress checkpoints and two phase of evaluation. In the first phase processor and assembler design will be evaluated and in the final phase entire system will be evaluated.
- Final test: The test will cover the entire syllabus. The detail will be communicated as the course progress.

References

- Noam Nisan and Shimon Schocken, The Elements of Computing Systems: Building a Modern Computer from First Principle, The MIT Press, Cambridge, 2008.
  Link to course material: nand2teris.org.
- David Patterson and John L. Henessy, Computer Organisation and Design: The Hardware Software Interface, Morgan Kaufmann An Imprint of Elsevier, 2014.
- Ronald Bryant and David R O'Hallaron, Computer Systems: A Programmer Perspective, Pearson India Education, 2016.
- Lecture Notes of our class
- Lecture Notes of: http://course.ece.cmu.edu/ ece600/ Foundation of Computer Systems by J P Shen, CMU
- NPTEL Lecture of Prof V Kamakoti, IIT Madras
- Design and Engineering of Computer Systems, Prof Mythili Vutukur, IIT Bombay

Resources and Contacts

Instructor: Jaynarayan T Tudu [jtt@]
Teaching Asst: Shivam and Shivam's Friend

The course news, teaching materials etc will be updated in the course page hosted in google class room

For quick updates you may also brows at:

http://jayresearch.github.io.

Question and Discussion

Thank You!