Introduction to Computer Organization

Welcome

Amey Karkare
`karkare@cse.iitk.ac.in`

Department of CSE, IIT Kanpur

Basics

- Required Background
    - Programming (Native Compilation techniques)
        - Pick up C/C++ if not known already.
    - Knowledge of digital gates, flip-flops, latches, counters etc.
    - Binary number system and operations
- What will be covered?
    - Programming using Assembly Language
    - Circuit descriptions in Bluespec Verilog (BSV)
    - Computer Organization

Course Structure

- Lectures:
  - Wednesday, Thursday, Friday; 11:00 AM - 11:50 AM, L-5
- Labs:
  - Groups of 2 students
  - New language for design descriptions: BlueSpec Verilog (BSV)
  - HW programming using FPGAs (Optional, if time and resources permit)
  - Assembly language programming

Course Policies

One midsem exam	25 %
One endsem exam	40 %
Labs	25 + 10 %

Tentative. May change as the course progresses.

Course will be heavy

Text

- No text books
- Follow the lecture notes very closely
- Reference Books
  - BlueSpec Verilog:
    - Rishiyur S. Nikhil and Kathy Czeck: BSV by Example [`www.bluespec.com/forum/download.php?id=140`]
  - Assembly Language: Online notes
    - `http://asm.sourceforge.net/`
    - `http://chortle.ccsu.edu/AssemblyTutorial/index.html`
  - Computer Organization:
    - Patterson, Hannessy: Computer Organization and Design
    - Hamacher et.al.: Computer Organization
    - Tanenbaum: Structured Computer Organization
    - Parhami: Computer Architecture
    - Stallings: Computer Organization and Architecture

Logistics

- I will form the groups of 2 students for assignments, and inform the class
- Give feed back through out the semester
  - Feel free to discuss anything with the instructor
  - including criticism of instructor, TAs
- Participate in the discussion
- No question is silly enough
  - Only Stupid Question = NO Question

Logistics

- Keep checking the course web page for announcements.
- Do not be late for the class
- Keep your cell phones switched off
- I travel extensively. Be ready for substitute classes on the week-ends/holidays and/or in the evenings.
  - Extra class on **Saturday, January 4th, 11 AM.**

DO NOT CHEAT

I follow zero tolerance policy.

- Straight **F** for copying in assignment
  - I do not distinguish between the *source* and the *destination* nodes of the *copy* edge.
- Names are also reported to authorities to be put in personal files.

Any Questions/Comments/Suggestions?

Review of Digital Circuits

- Basic Gates
  - AND, OR, NAND, NOR, XOR, XNOR, NOT, BUF
- Memory Elements
  - Latch, Flip Flops
  - Registers
    - Level Sensitive, Edge Triggered
    - Input output behaviour (Parallel or Serial): SISO (Delay Line), SIPO, PISO, PIPO
  - Basic Circuits
    - Multiplexers, Counters, ALU, Adders, Multipliers

Combinatorial Gates

```
always @(a or b)
   y = a & b;
```

OR

```
assign y = a & b;
```

Memory Elements

- Latches

```
always @( LE or D )
    if (LE == 1) Q = D;
```

- Flip Flops

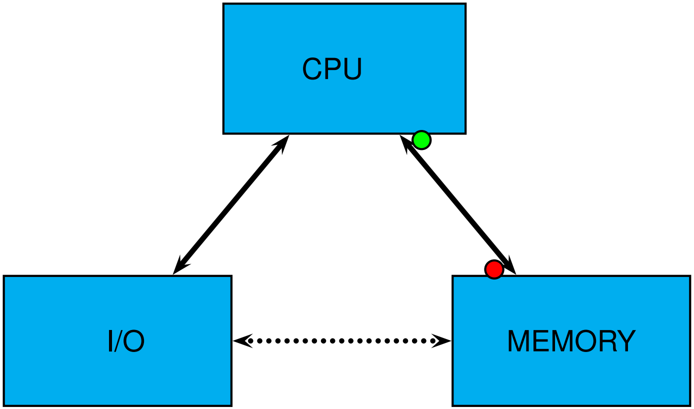
```
always ( posedge CLK )
    Q = D;
```

Latches are level sensitive while flip flops are edge triggered.

A Computer

- Computer is a “machine”
- As long as power is supplied, processor keeps executing instructions
  - Stored program model
  - Sequential order of execution
- Memory: Program and data storage
- Disk: File storage (passive data storage)

The machine is electronic



Fetch-Execute-Store Model

- Processor reads an instruction from memory
- Instruction:
  - A sequence of bits, understood and operated upon by processor
- Processor interprets these bits and operates on data
  - Stored in internal registers or in memory
- Result of the execution is stored in memory/registers.
- Processor continues fetching the next instruction.
- Program:
  - Collection of instructions that are fetched and executed one after another

Fetch-Execute-Store Model

- Processor takes program only as a sequence of bits
  - Machine Language
- We understand programs in various forms
  - Assembly language: One to one correspondence with the machine language
  - High level languages:
    - Translators are needed to convert to machine language (compilers)
  - Programs for Virtual machine:
    - Languages such as Java, compile code to a virtual machine which is then *interpreted* by a tool.