Computer Hardware Design (CSC-312)
Tribhuvan University
Institute of Science and Technology
Soch College of Information Techology

Course Title: Computer Hardware Design
Course no: CSC-312 ---------- **Full Marks:** 80+20
Credit hours: 3 ---------- **Pass Marks:** 32+8
Nature of course: Theory (3 Hrs.)
Course Synopsis: To introduce students to theoretical and practical concepts relevant to the structure and design of modern digital computers. The course covers computer architecture from gate-level logic through processor design to multiprocessor and network issues.
Goal: This course will make the student able to design the hardware components.

Course Contents:

Unit 1. Introduction, Computer Abstractions and Technology ---------- 2 Hrs.
Hierarchical approach to understanding & designing a complex system, Software, Hardware, Computer components, Processor: Control, Data path. Memory, Input & output, Components of retail price in the computer industry, Overview of computer hardware, IO, Computer processors; CISC, RISC, DSP, Hybrid, Measuring performance. Execution time, Operations per second, Throughput, Real-time computing and performance metrics

Unit 2. Digital Logic Design ---------- 6 Hrs.
Gates, truth tables, and logic equations, Combinational logic and basic components. PLAs and ROMs, Memory elements. Finite state machines

Unit 3. Data Representation, Manipulation and Addressing ---------- 6 Hrs.
Signed and unsigned numbers, Addition and subtraction. Design of ALUs.
Multiplication. Floating-point representation, Addressing: An application of unsigned integers: Byte-addressed memory, Byte ordering conventions, Big-endian, Little-endian, Pointers: Address vs. contents, Signed representations of integers

Unit 4. Basic Processor Design ----------12 Hrs.
Design of the datapath of an ALU that executes the add, sub, and, or instructions, Control signals for the ALU, State elements and clocking,
Block view of a single-clock-cycle processor datapath, Control of the single-clock-cycle implementation, Control of the multiple-clock-cycle implementation, Exceptions and interrupts, Karnaugh maps, Multiplexors, Adders, Decoders, Data paths.
Single-cycle, control. Multi-cycle control, Microprocessor design: Microprogramming, Hardwired programming. Parallel processors, SIMD computers--Single Instruction Stream, Multiple Data Streams, MIMD Computers--Multiple Instruction Streams, Multiple Data,

Streams; Programming MIMDs, MIMDs connected by a single bus, MIMDs connected by a network, Future directions for parallel processors, Programming for parallel processors in a higher-level language

Unit 5. Sequential Logic Circuits ----------5 Hrs.
Outputs and next state as vectors of Boolean functions of inputs and present state,
Latches: Set and reset latches, SR latch, CSR latch, JK latch, D latch, Master-slave D flip-flop, Lightning introduction to finite state machines

Unit 6. Pipelining ---------- 5 Hrs.
A pipelined data path, Pipelined control, Visualization of pipelined data flow, Pipeline diagrams, Gantt charts, Data hazards, Compiler elimination of data hazards, Hardware control for data hazards: Reducing data hazards: Forwarding, Branch hazards, Performance of pipelined systems, Programming for a pipelined processor in a higher-level language

Unit 7. Memory Hierarchies ---------- 3 Hrs.
Hardware implementations of 1-bit memory, DRAM, SRAM, ROM, Hardware implementations of multiple-bit memory, DRAM, SRAM, ROM, SRAM and DRAM chip and system architectures, System bus architectures (processor to/from memory), Hierarchical memory systems, The processor-memory speed gap, Interleaved memory, Caches, Direct-mapped caches, Fully associative caches Set-associative caches, Virtual memory, A common framework for memory hierarchies

Unit 8. Multiprocessors ---------- 2 Hrs.
Single-bus networks, Cache consistency, Networks and clusters.

Unit 9. Introduction to Assembly Language ---------- 4 Hrs.
Instructions, The fetch-execute cycle, Format of an assembly-language program, Comments, Directives, Data declarations in SPIM, Executable instructions, Survey of differences between SAL (Simple Abstract Language), human-coded MIPS assembly language, and true MIPS assembly language, Load-store architectures, Addressing modes, MIPS addressing modes and the corresponding formats in assembly language and object code, Implementation of I/O, Arrays, Usage of arithmetic and logical instructions in SAL, Branch instructions in SAL and SPIM, Stacks, Support for procedures in computer hardware, Alternatives to the MIPS approach

Textbooks:
David A. Patterson and John L. Hennessy. "Computer Organization and Design: The Hardware/Software Interface"

References:

M. M. Mano "Digital logic Design"

Prerequisite(s):
Fundamentals of design methodology and descriptive tools; performance and cost; overview of instruction set issues; processor implementation techniques; memory hierarchy; input/output; parallel computer systems, introduction to formal computer aided design tools and simulations.