SARVAJANIK UNIVERSITY
Sarvajanik College of Engineering and Technology
Bachelor of Technology

B. Tech. Semester V

Subject Name: **Computer Design with PLDs and FPGAs** **Subject Code: BTEC15502**

Type of course: OE

Prerequisite: Basic of Computer hardware , Digital logic and programming

Rationale: This course provides detail of a computer system's functional components, their characteristics, performance and interactions including system bus, different types of memory and input/output organization and CPU. It also covers architectural issues such as instruction set program and data types. It gives the foundation for RTL design in VLSI Systems along with practical design skills. At the end of the course students will learn how to design the basic building blocks of computer architecture including ALU, control unit and memory using hardware description language. By learning this course student will be able to apply and test hardware logic for chip implementation of Microprocessor which is brain of any computing system.

Teaching and Examination Scheme:

Teaching Scheme				Theory Marks			Practical Marks		Total
L	T	P	C	TEE	CA1	CA2	TEP	CA3	150
3	0	2	4	60	25	15	30	20	

CA1: Continuous Assessment (assignments/projects/open book tests/closed book tests **CA2:** Sincerity in attending classes/class tests/ timely submissions of assignments/self-learning attitude/solving advanced problems **TEE:** Term End Examination **TEP:** Term End Practical Exam (Performance and viva on practical skills learned in course) **CA3:** Regular submission of Lab work/Quality of work submitted/Active participation in lab sessions/viva on practical skills learned in course

Content:

Sr. no.	Topics	Teaching Hrs.	Module % Weightage
1.	**Data Representation and Micro-operations:** Basic computer data types, Complements, Fixed point representation, Floating point representation. Register Transfer language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro- Operations, Logic Micro-Operations, Shift Micro-Operations, Arithmetic logical shift unit.	4	10
2.	**Finite State Machines (FSM):** Mathematical representation, State transition, State table, MOORE model, MEALY model, Basic concepts in state machine analysis.	4	10
3.	**Computer Organization and Micro programmed Control:** Instruction codes, Computer registers, Instruction cycle, Memory-Reference Instructions, Input-output and interrupt, Complete computer description, Design of Basic computer,	8	15

	Design of Accumulator Unit. Programming Arithmetic and logic operations, Subroutines, I-O Programming, Control Memory, Address sequencing, Micro program example, Design of Control Unit.		
4.	**Central Processing Unit:** CPU Architecture, Register Organization, Stack Organization, Instruction formats, Addressing Modes, Data transfer and manipulation, Program control, Reduced Instruction Set Computer (RISC) & Complex Instruction Set Computer (CISC), Instruction interpretation and Sequencing, RTL interpretation of instructions, Case study - instruction sets of MIPS processor.	5	15
5.	**Pipeline And Vector Processing:** Flynn's taxonomy, Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction, Pipeline, RISC Pipeline, Vector Processing, Array Processors.	5	10
6.	**Computer Arithmetic:** Introduction, Addition and subtraction, Multiplication Algorithms: Array multiplier, (Booth Multiplication Algorithm, Division Algorithms, Decimal Arithmetic Unit.	4	10
7.	**Input-Output Organization:** Input-Output Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, DMA, Input-Output Processor (IOP), Serial communication.	4	10
8.	**Memory and Programmable Logic Devices:** Types of memory, Introduction to Programmable Logic Devices, Read-Only Memory, Programmable Logic Arrays (PLA), Programmable Array Logic (PAL), Complex Programmable Logic Devices (CPLDs), Field Programmable Gate Array (FPGA)	6	10
9.	**Multiprocessors:** Characteristics of Multiprocessors, Interconnection Structures, Inter-processor Arbitration, Inter-processor Communication and Synchronization, Cache Coherence, Shared Memory Multiprocessors. Specialized Architectures - Multi-core systems, GPU.	5	10

Suggested Specification table with Marks (Theory/Practical):

% Distribution of Marks					
R Level	**U Level**	**A Level**	**N Level**	**E Level**	**C Level**
15	20	15	20	15	15

Legends: R: Remembrance, **U:** Understanding; **A:** Application, **N:** Analyze, **E:** Evaluate **C:** Create and above Levels **(Revised Bloom's Taxonomy)**

Reference Text Books:

Sr. No.	Title of book /article	Author(s)	Publisher and details like ISBN	Year of publication	Publication Edition
1.	Digital Design and Computer Architecture	David Money Harris ,Sarah L. Harris	Morgan Kaufmann	2007	Latest
2.	Computer Architecture -- A Quantitative Approach	John L. Hennessy, David A. Patterson	Morgan Kaufmann	2012	5th
3.	Computer System Architecture	Morris Mano	Pearson	2019	Latest

Course Outcome:

Sr. No.	CO Statement After learning this subject students will be able to,	Marks % weightage
CO-1	Represent fixed and floating point numbers and use arithmetic as well as logical operations.	20
CO-2	Explain computer organization and micro programmed control for various operations.	20
CO-3	Discuss operations with pipelining and pipeline hazards.	10
CO-4	Describe computer arithmetic algorithms and data transfer between I/O memory and peripherals.	20
CO-5	Implement hardware logic using programmable logic devices.	15
CO-6	Explain multiprocessor architecture, interconnection and inter process communication.	15

Mapping with POs:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO-1	3	2	-	2	-	2	-	-	2	2	2	2
CO-2	3	3	3	3	2	3	-	-	2	2	2	3
CO-3	3	3	3	3	2	2	3	-	2	2	2	3
CO-4	3	2	2	2	2	2	-	-	2	2	2	3
CO-5	3	3	3	3	3	3	3	-	3	3	3	3
CO-6	2	3	2	3	-	2	-	-	3	2	3	3

List of practical:

1. Write VERILOG code for 8-bit adder using structural, behavioral and dataflow modeling. Show output with at least four combinations of input in a single waveform.
2. Implement 32-bit and 64-bit adder subtractor with mode control using the behavioral method. Measure and compare synthesis delay and resource utilization.
3. Implement 8×1 multiplexer and 3×8 decoder. Show output such that all combinations of inputs are visible.
4. Study of programmable devices like FPGA and CPLD. Study Basys3 FPGA board with Artix®-7 FPGA from Xilinx and understand the process of downloading bit files on FPGA as well as execution of code using external inputs.
5. Write VERILOG code for 8-bit register and 4–bit up / down counter.
6. Write Verilog code to implement and test simple ALU having the following specifications.
   - Inputs – A, B (8 bit)
   - Opcode – Operation Selection (3 bit - 8 operations) – Add, subtract, Complement, AND, OR, XOR, Comparison, Equality Comparison
   - Output – R (8 bit)
   - CF – carry out the flag, ZF – Zero flag Measure synthesis delay and resource utilization
7. Write VERILOG code for 64×8-bit ROM with predefined data and 32×8-bit RAM. Demonstrate reading of data with random/sequential address.
8. Write Verilog code to implement control logic using finite state machine.
9. Write Verilog code to implement array multiplier or multiplier using Booth's Algorithm.
10. Write Verilog code to implement arithmetic or logical instructions.
11. Implement opcode and fetch algorithm using pipeline processing.
12. Implement 16 bit single cycle MIPS processor in verilog HDL

Major Equipment:

- FPGA and CPLD boards

List of Open Source/learning websites:

- www.xilinx.com

List of Open Source software:

- https://archive.nptel.ac.in/courses/117/108/117108040/ - Digital System Design with PLDs and FPGAs
- http://www.intel.com/pressroom/kits/quickreffam.htm
- web.stanford.edu/class/ee282/