GUJARAT TECHNOLOGICAL UNIVERSITY

ELECTRONICS & COMMUNICATION (COMMUNICATION SYSTEMS ENGG) (04)

ASIC DESIGN

SUBJECT CODE: 2710432

SEMESTER: I

Type of course: Major Elective - I

Prerequisite: Logic gates and flip-flops, Combinational and sequential Logic circuits

Rationale: This course is first step for PG students in the field of ASIC prototyping which is also called FPGA (Field programmable gate array) / SoC (System on Chip) prototyping. With large resource of configurable logic blocks FPGA is designed to be configured by designer using hardware description language for digital system design. This Course offers a variety of topics of immediate relevance to digital VLSI System Design

Teaching and Examination Scheme:

Teaching Scheme			Credits	Examination Marks						Total Marks
L	T	P	C	Theory Marks		Practical Marks				
				ESE (E)	PA (M)	PA (V)		PA (I)		
						ESE	OEP	PA	RP	
3	0	2	4	70	30	20	10	20	0	150

Content:

Sr. No.	Topics	Teaching Hrs.	Module Weightage
1	**Introduction** ASIC Design flow, Design Methodologies, Hardware modeling issues, Overview of FPGA and CPLD technology	2	5
2	**Hardware Description Language** Elements of VHDL, Entity, architecture, configuration declaration. Identifiers, data types and operators, Assignment statement, Objects in VHDL – signals, variables, constants, files, Attributes of objects.	4	5
3	**Behavioral modeling** Process statement, Signal and variable assignment, Wait statement, if statement, Case statement, Loops, exit, and next statement, Assertion and report statement, Multiple process, Postponed process. Subprograms – procedures and functions, Subprogram overloading and operator overloading. RTL description	6	15
4	**Dataflow modeling** Concurrent signal assignment, sequential signal assignment, delta delay, multiple drivers, conditional signal assignment using when … else, selected signal assignment using with select, block statement, concurrent assertion signal	6	15
5	**Structural modeling** Component declaration, generics and component instantiation, Example of making hierarchical circuit. Generate statement, aliases, mixed modeling style	4	15
6	**Finite State Machine (FSM)**	6	15

	Stat diagrams and state tables, Moore and Mealy finite state machine, encoding style, FSM issues, Timing issues, pipelining, resource sharing, metastability, synchronization, MTBF Analysis, setup/hold time of various types of flip-flops, synchronization between multiple clock domains, reset recovery		
7	**Configuration and packages** Configuration specification and declaration, conversion functions, direct instantiation, incremental binding. Package declaration, package body, design file	2	5
8	**Test Benches and library** File I/O operations, Test bench architecture, test bench examples, library declaration and user defined library	2	5
9	**Programmable Logic Design** Basics of Programmable logic devices - PROM, PAL, PLA, etc, CPLD architecture and its building blocks, FPGA architectures and its building blocks, Carry chains in FPGA, Dedicated multipliers and memory in FPGA, RTL synthesis test methodology, Design synthesis, Technology mapping for FPGAs: SRAM, Fuse, Antifuse, EPROM programming techniques. Design implementation using CPLD and FPGA, Floor planning, Placement and routing	8	20

Reference Books:

1. Douglas L. Perry, VHDL programming by examples, 4th Ed., Tata McGraw Hill.
2. J. Bhasker, VHDL Primer, Pearson Education Asia, Low Price Edition
3. Charles H Roth, Jr., Principles of Digital Systems Design using VHDL, Cengage Learning
4. Kevin Skahill, VHDL for programmable logic, Pearson Education Asia, Low Price Edition
5. Michael John Sebastian Smith, Application Specific Integrated Circuits, Pearson Education Asia.
6. Wakerly, J. F., Digital Design: Principles and Practices 4th Edition, Pearson
7. Xilinx and Altera Application Notes on the architecture of FPGAs and CPLDs

Course Outcome:

After learning the course the students should be able to design digital systems using hardware description languages required for VLSI Design such as VHDL. The students shall learn how to program and test programs on FPGAs or CPLD. The students shall get exposure to the Various Digital Design Issues, ASIC Technology and Design techniques.

List of Experiments: (with Open Ended Problems)

1. Write entity with 2 inputs a, b and 7 outputs x1 to x7 for logic gates NOT, AND, OR, NAND, NOR, XOR, XNOR respectively.
2. Write code for 4x1 MUX with structural modeling and using same, implement 8x1 MUX. Extend above programs for 16x1 MUX.
3. Write code for 2x4 Decoder with structural modeling and using same, implement 3x8 decoder. Extend program for 4x16 decoder.

4. Write code for D-latch with structural modeling and implement 4-bit register using same. Extend above program for 16-bit register.
5. Write VHDL code to realize a 4 bit, 4 X 1 multiplexer.
   - Write the architecture using equations
   - Write the architecture using dataflow model using (with ... select) construct
   - Write the architecture using dataflow model using (when ... else) construct
   - Write the behavioral code using (case ... when ...) construct
   - Write the behavioral code using (if ... then ...) construct.

6. Implement the following in VHDL code.

    - An 8-bit ripple adder (use equations)
    - An 8-bit carry look ahead adder (use equations)
    - Implement an 8 bit adder using the operator .+. in .ieee.std_logic_unsigned package.
    - 8-bit ripple adder using structural modeling with generate statement.
    - 8-bit ripple adder and subtractor using mode control input.

Synthesis code for any target FPGA device and compare the area (resource utilization) and performance (delay) in each case.

7. Write VHDL code to implement various flip flops.
8. Write VHDL code to implement 16-bit universal shift register, which supports shift-left, shift right, and parallel-in modes.
9. Write VHDL code for the clock divider to get 1 Hz clock from 1 MHz clock input.
10. Write VHDL code to implement up-down counter.
11. Write VHDL code to implement Parallel to serial converter and Serial to Parallel Converter.
12. Write VHDL code for sequence detector using FSM

Major Equipments: Kits with Xilinx / Altera FPGA

List of Open Source Software/learning website:

Xilinx ISE