SEMESTER S7

ADVANCED VLSI ARCHITECTURES FOR DSP

Course Code	**PEEVT 755**	**CIE Marks**	40
Teaching Hours/Week (L: T:P: R)	3-0-0-0	**ESE Marks**	60
Credits	5/3	**Exam Hours**	2Hrs.30 Min.
Prerequisites (if any)	VLSI		

Course Objectives:

1. Equip students with the knowledge of pipelining and parallel processing techniques
2. Enable students to comprehend and apply retiming and unfolding techniques for system optimization.
3. Develop the ability to implement and analyze fast convolution algorithms and advanced filtering methods.
4. Train students to design and optimize bit-level arithmetic architectures, focusing on efficiency and noise reduction.

SYLLABUS

Module No.	Syllabus Description	Contact Hours
1	Pipelining and Parallel Processing - Pipelining of FIR Digital Filters, Data-Broadcast Structures, Fine-Grain Pipelining. Parallel Processing, Pipelining, and Parallel Processing for Low Power. Retiming: Definition and Properties, Solving System of Inequalities, Retiming Techniques- Cutset Retiming and Pipelining, Retiming for Clock Period Minimization, Retiming for Register Minimization	**9**
2	Unfolding: Properties of Unfolding, Critical Path, Unfolding and Retiming, Application of Unfolding, Sample Period Reduction, Parallel Processing. Folding: Introduction to Folding Transformation, Register Minimization Techniques, Register Minimization in Folded Architectures, Folding in Multirate Systems.	**9**
3	Fast convolution – Cook Toom Algorithm, Winograd Algorithm, Iterated convolution, Cyclic convolution	**9**

	Algorithmic strength reduction in filters and transforms- Parallel FIR filters, DCT and IDCT, Pipelined and parallel recursive and adaptive filters- pipeline interleaving in Digital filters, Pipelining in IIR digital filters, Parallel processing for IIR filters, Low power IIR filter design using pipelining and parallel processing.	
4	Digital lattice filter structures- Schur algorithm, Digital basic lattice filters, Derivation of one multiplier Lattice filter, Derivation of scaled-normalized lattice filter, Round off noise calculation in Lattice filters. Bit level arithmetic architectures- parallel multipliers, interleaved floor plan and bit plane based digital filters, bit serial filter design and implementation, Canonic signed digital arithmetic.	**9**

Criteria for Evaluation (Evaluate and Analyse): 20 marks

Evaluation Methods:

1: Practical Experiments Using Design and Analysis Tools (10 marks)

Students will perform specific experiments using tools like MATLAB/Simulink, Cadence Virtuoso, Xilinx Vivado Design Suite, or Synopsys Design Compiler. Each experiment will focus on implementing and analyzing different DSP architectures and techniques.

2: Course Project (10 marks)

Comprehensive project involving design, implementation, and analysis of different DSP architectures models. Project phases: Proposal, Design, Implementation, Testing, Final Report, Presentation, and Viva Voce.

Sample Experiments:

Experiment 1: Design and simulate a pipelined FIR filter.

- ***Objective:*** Understand the impact of pipelining on the performance and power consumption of FIR filters.
- ***Tools Used:*** MATLAB/Simulink
- ***Steps Involved:***
    - Define the FIR filter specifications (filter order, cutoff frequency).

- Design the FIR filter using MATLAB.
- Implement the filter in Simulink and add pipelining stages.
- Simulate the filter and compare performance metrics (latency, power consumption) with and without pipelining.
- Analyze and document the results.

Experiment 2: Design an unfolded DSP system to reduce the critical path.

- ***Objective:*** Learn how unfolding can reduce the critical path and enhance system throughput.
- ***Tools Used:*** MATLAB/Simulink
- ***Steps Involved:***
  - Select a DSP algorithm (e.g., FIR filter).
  - Implement the algorithm in MATLAB.
  - Apply unfolding techniques to the algorithm.
  - Simulate the original and unfolded designs in Simulink.
  - Compare critical paths and document the performance improvements.

Experiment 3: Design and simulate the Winograd convolution algorithm.

- ***Objective:*** Demonstrate the computational benefits of the Winograd algorithm over conventional methods.
- ***Tools Used:*** Xilinx Vivado
- ***Steps Involved:***
  - Study the Winograd algorithm theory.
  - Implement the algorithm in HDL.
  - Simulate the design in Vivado.
  - Compare the computational efficiency with conventional convolution methods.
  - Document the findings.

Experiment 4: Implement a scaled-normalized lattice filter and analyze noise reduction techniques.

- ***Objective:*** Evaluate round-off noise reduction in digital lattice filters.
- ***Tools Used:*** Cadence Virtuoso
- ***Steps Involved:***
  - Design a lattice filter in HDL.
  - Implement the design in Cadence Virtuoso.
  - Apply scaled-normalized techniques for noise reduction.
  - Simulate the filter and analyze round-off noise.
  - Document the noise reduction results.

Sample Project Topics:

1. Design a low-power parallel processing architecture for a DSP application
2. Apply retiming to minimize the clock period of a digital circuit
3. Optimize the register usage in digital circuits through retiming.
4. Apply unfolding techniques to reduce the sample period of a digital filter.
5. Implement a folded architecture for a multirate system focusing on register minimization.
6. Apply folding transformations and understand their impact on hardware resource optimization
7. Analyze the efficiency of the Cook-Toom algorithm compared to standard convolution methods
8. Design and implement a parallel FIR filter using algorithmic strength reduction techniques.
9. Implement a pipelined recursive filter using algorithmic strength reduction.
10. Design and simulate a digital lattice filter with one multiplier.
11. Design and implement a parallel multiplier architecture and evaluate their performance and efficiency.
12. Implement a bit-serial filter design for understanding the trade-offs between hardware complexity and performance in bit-serial designs.

Criteria for Evaluation: Lab Experiments (10 marks)

Understanding of Concepts (3 marks)

- Demonstrates a thorough understanding of the theoretical concepts related to the experiments.
- Correctly explains the purpose and expected outcomes.

Implementation and Accuracy (3 marks)

- Correctly implements the DSP architectures using appropriate tools.
- Ensures the design functions as expected with minimal errors.

Analysis and Problem-Solving (2 marks)

- Effectively analyzes the model performance and identifies issues.
- Demonstrates problem-solving skills in addressing challenges encountered during experiments.

Documentation and Reporting (1 mark)

- Provides detailed documentation of the experimental setup, process, and outcomes.
- Includes visualizations, code snippets, and analysis of results.

Presentation and Communication (1 mark)

- Clearly presents the experiments and their results.
- Able to answer questions and explain design choices.

Course Project (10 marks)

Project Proposal and Planning (2 marks)

- Submits a well-defined project proposal outlining objectives, methodology, and expected outcomes.
- Demonstrates thorough planning and a clear timeline for the project.

Design and Implementation (3 marks)

- Implements the project design accurately using appropriate tools and techniques.
- The design is functional and meets the project objectives.

Innovation and Creativity (2 marks)

- Introduces innovative ideas or unique approaches in the design and implementation.
- Demonstrates creativity in solving problems or optimizing designs.

Analysis and Testing (2 marks)

- Effectively analyzes the project design to identify and address any issues.
- Conducts thorough testing to verify the functionality and performance of the model.

Final Report and Presentation (1 mark)

- Submits a comprehensive final report detailing the project, including objectives, design, methodology, analysis, and results.
- Clearly presents the project and its outcomes, and effectively communicates the key points.

End Semester Examination Marks (ESE):

In Part A, all questions need to be answered and in Part B, each student can choose any one full question out of two questions

Part A	Part B	Total
• 2 Questions from each module. • Total of 8 Questions, each carrying 3 marks **(8x3 =24marks)**	2 questions will be given from each module, out of which 1 question should be answered. Each question can have a maximum of 3 sub divisions. Each question carries 9 marks. **(4x9 = 36 marks)**	**60**

Course Outcomes (COs)

At the end of the course students should be able to:

Course Outcome		Bloom's Knowledge Level (KL)
CO1	Demonstrate an understanding of pipelining and parallel processing techniques in DSP systems, including the design and implementation of pipelined FIR digital filters and low power parallel processing architectures.	**K3**
CO2	Apply retiming techniques to optimize digital circuits for clock period and register minimization, solving system inequalities and implementing cutset retiming and pipelining.	**K5**
CO3	Utilize unfolding and folding transformations to reduce critical path and register usage in DSP architectures, and apply these techniques to multirate systems for sample period reduction.	**K4**
CO4	Implement advanced convolution algorithms such as Cook-Toom and Winograd, and apply algorithmic strength reduction techniques to enhance the performance of FIR and IIR filters	**K6**
CO5	Design and analyze digital lattice filter structures and bit-level arithmetic architectures, including parallel multipliers, bit-plane-based digital filters, and bit-serial filter designs, focusing on performance optimization and noise reduction.	**K6**

Note*: K1- Remember, K2- Understand, K3- Apply, K4- Analyse, K5- Evaluate, K6- Create*

CO-PO Mapping Table (Mapping od Course Outcomes to Program Outcomes)

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	**3**	**3**	**2**		**3**							
CO2	**3**	**3**	**2**		**3**							
CO3	**3**	**3**	**2**		**3**							
CO4	**3**	**3**	**2**		**3**							
CO5	**3**	**3**	**2**		**3**							

Note: *1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High), -: No Correlation*

Text Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	VLSI DSP Systems- Design and Implementation	Keshab K Parhi	John Wiley	1st Edn 1999.
2	Digital Signal processing for multimedia systems	Keshab K Parhi and Takao Nishitami	CRC press	2018
3	Digital Signal Processing with Field Programmable Gate Arrays	Uwe Meyer-Baese	Springer	4th Edn, 2014
4	VLSI Design Methodology Development	Thomas Dillinger	Prentice Hall	1st Edn 1998
5	Advanced Digital Signal Processing and Noise Reduction	Saeed V. Vaseghi	Wiley	4th Edn 2008

Reference Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition and Year**
1	Understanding digital signal processing	Richard G Lyons	Pearson Education India	3,1997
2	DSP Integrated Circuits	Lars Wanhammar	Academic Press	1st Edn 1999
3	Digital Filter Design	T. W. Parks and C. S. Burrus	Wiley-Interscience	1st Edn 1987
4	Architectures for Digital Signal Processing	Peter Pirsch	John Wiley & Sons	1st Edn 1998
5	CMOS VLSI Design: A Circuits and Systems Perspective	Neil H. E. Weste and David Harris	Pearson	4th Edn 2010

Video Links (NPTEL, SWAYAM…)	
Module No.	**Link ID**
1	https://onlinecourses.nptel.ac.in/noc20_ee44/preview
2	https://www.youtube.com/playlist?list=PLT1QAv48lhQKwFZ0TkqpJaUm2LeW9226z