|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S Y L L A B U S** | | | | |
| **General information** | **Title and code of**  **subject, number of credits** | CMS 303 Digital Logic- 8 credits | | |
| **Department** | Department of Computer Engineering | | |
| **Program** | Bachelor | | |
| **Academic semester** | 2017 fall | | |
| **Lecturer** | Master of Science (Electronics Engineering)  Sabuhi Ganiyev | | |
| **E-mail:** | [s.ganiyev@gmail.com](mailto:s.ganiyev@gmail.com) | | |
| **Phone number:** | +994 77 520 73 50 | | |
| **Lecture room/Schedule** | 11 Mehseti Street, AZ1096 Baku, Azerbaijan (Neftchilar campus), room | | |
|  | **Consultations** | Monday 14:00 – 15:00 | | |
| **Course**  **language** | English | | | |
| **Type of the subject** | Major | | | |
| **Textbooks and additional materials** | ***Textbooks*:**   1. Fundamentals of digital logic with Verilog design, 3rd Edition, (2014) by S.D. Brown, Tata McGraw-Hill Education. (required)   Optional Reference Texts:   1. Fundamentals of digital logic with Verilog design, 2nd Edition, (2007) by S.D. Brown, Tata McGraw-Hill Education.   Additional Resource Texts:   1. Digital logic design using verilog: coding and RTL synthesis (2016) by Taraate Vaibbhav, Springer. 2. Digital Logic Design Principles, (2007) by N. Balabanian, B. Carlson, Wiley India Pvt. Limited. 3. Digital Principles and Logic Design, (2007) by A. Saha, N. Manna, Infınıty Scıence Press LLC.   **Auxiliary Web sources:**  <https://www.youtube.com/watch?v=M0mx8S05v60&list=PLBlnK6fEyqRjMH3mWf6kwqiTbT798eAOm>  <https://www.youtube.com/watch?v=j4fDYX5VZF0&list=PLEbnTDJUr_Ica5kK6UypsWpf95Ut2sK3o>  <https://www.youtube.com/watch?v=La9Nk6iwHvU&list=PLdNU3YZoxOwsx4YL3Grfixt1wNOZjOpyS>  <https://www.youtube.com/watch?v=QSEl_O0Gtoo&list=PLoM0uG7tqR3qVss3zhBRniXU7mhHy2bwj> | | | |
| **Teaching methods** | **Lecture** | | | 15 |
| **Group discussions at seminars** | | | 15 |
| **Assessment** | **Components** | | **Date/ Deadline** | **Percent (%)** |
| **Quizzes** | | During the semester | 20 |
| **Active participation** | | At each lesson | 5 |
| **Attendance** | |  | 5 |
| **Midterm exam** | |  | 30 |
| **Final exam** | |  | 40 |
| **Final** | |  | **100** |
| **Course description** | This subject teaches main principles of digiltal logic. It introduces the essential steps in the design process and discussed how CAD (Computer-Aided Design) tools can be used. Moreover, it explaines the concepts of logic circuit synthesis and optimization, and describes how Verilog can be used to specify the desired functionality and how CAD tools provide a mechanism for developing the required circuits. This course also presents combinational circuits that are used as building blocks which includes the encoder, decoder, and multiplexer circuits. Using above mentioned circuits the application of different Verilog constructs will be illustrated. It gives the students an opportunity to discover more advanced features of Verilog. Lastly, storage elements and synchronous sequential circuits will be analyzed and described in detail. | | | |
| **Course objectives** | The main objective of this course as follow:   1. To be able to perform the conversion among different number systems; 2. To familiarize with logic gates - AND, OR, NOT, NAND, NOR, XOR. 3. To understand basic properties of Boolean algebra, able to simplify simple Boolean functions. 4. To be able to design simple combinational logics using baisc gates. 5. To be able to optimize simple logic using Karnaugh maps. 6. To familiarize with SR lutch, D Flip-Flop and their usage and able to analyze sequential logic circuits. 7. To understand finite state machines (FSM) concepte. 8. To familiarize with basic combinational and sequential components circuits. 9. To be able to understand and use one high-level hardware description languages (Veriliog) to design combinational or sequential circuits. | | | |
| **Learning outcomes** | What students should know by the end of the course:  Concept of digital and binary systems, to design and analyze logic circuits, basic software tools for the design and implementation of digital circuits and systems. | | | |
| **Rules (Educational policy and behavior)** | Lesson organizationGeneral information on the subject will be provided for the students during lectures.Student’s knowledge on the previous topics will be evaluated and new topic will be explained by mins of visual aids during seminars. Student’s knowledge level will be tested oraly and in written forms before midterm and final exams. Submission of the individual works by the end of course is obligatory. Attendance  Participation of students at all classis is important. Students should inform dean’s office about missing lessons for particular reasons (illness, family issues and etc.). Students, missing more than 25% of lessons, are not allowed to take the exam.  Lates  Those students who are late for lessons for more than 15 minutes are not allowed to participate at the lesson. Despite this, the student is allowed to take part in the second part of the lesson.  Quizzes  Those students who have informed the teacher and the dean’s office about missing the quiz in advance for particular reasons, are allowed to take the quiz next week.  Exams  All the issues related to the participation and admission to the exam are regulated by the faculty dean.  Topics of midterm and final exams are provided for the students before the exams. The questions of midterm exam are not repeated in the final exam.  Violation of the rules of the exams  Disrupting the quiz and taking copy during midterm and final exams is forbidden. Quiz papers of the student who do not follow these rules are canceled and the students are expelled from the Quiz by getting 0 (zero).  The rule for completing the course  In accordance with the University rules the overall success rate to complete the course should be 60% or above. The students who failed the exam would be to take this subject next semester or next year.  Rules of conduct for Students  Disruption of the lesson and not following ethical norms during the lesson, as well as conduction of the discussions by the students without permission and using mobile phones is forbidden. | | | |

This program reflects the comprehensive information about the subject and information about any changes will be providedin advance.

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Dates**  **(planned)** | **Subject topics** | **Textbook/**  **Assignments** |
| ***1*** | ***19.09.17*** | ***Lecture №1.*** *Introduction:* Digital hardware. The design process. Structure of a computer. Logic circuit design in this book. Digital representation of information. Binary numbers. Conversion between decimal and binary systems. ASCII character code. Digital and analog information. | [1] p. 1-16 |
|  | ***Seminar №1:*** *Questions and Exercises* | [1] p. 18 |
| ***2*** | ***26.10.17*** | ***Lecture №2.*** *Introduction to logic circuits:* Variables and functions. Inversion. Truth tables. Logic gates and networks. Analysis of a logic network. Boolean algebra. The venn diagram. Notation and terminology. Precedence of operations. Synthesis using and, or, and not gates. Sum-of-products and product-of-sums forms. NAND and NOR logic networks. Design examples. Three-way light control. Multiplexer circuit. Number display | [1] p. 21-63 |
|  | ***Seminar №2:*** *Questions and Exercises* | [1] p. 111-119 |
| ***3*** | ***03.10.17*** | ***Lecture №3***. *Introduction to logic circuits:* Introduction to CAD tools. Design entry. Logic synthesis. Functional simulation. Physical design. Timing simulation. Circuit implementation. Complete design flow. Introduction to Verilog. Structural speciﬁcation of logic circuits. Behavioral speciﬁcation of logic circuits. Hierarchical verilog code. How *not* to write verilog code. | [1] p. 63-78 |
|  | ***Seminar №3:*** *Questions and Exercises* | [1] p. 111-119 |
| ***4*** | ***10.10.17*** | ***Lecture №4.*** *Introduction to logic circuits:* Minimization and karnaugh maps. Strategy for minimization. Terminology. Minimization procedure. Minimization of product-of-sums forms. Incompletely speciﬁed functions. Multiple-output circuits. | [1] p. 78-101 |
|  | ***Seminar №4:*** *Questions and Exercises* | [1] p. 111-119 |
| ***5*** | ***17.10.17*** | **Lecture №5**. *Implementation Technology*: Transistor. NMOS Logic Gates. CMOS Logic Gates. Speed of Logic Gate Circuits. Negative Logic System. Standard Chips. 7400-Series Standard Chips. Programmable Logic Devices. Programmable Logic Array (PLA). Programmable Array Logic (PAL). Programming of PLAs and PALs. Complex Programmable Logic Devices (CPLDs). Field-Programmable Gate Arrays. Custom Chips, Standard Cells, and Gate Arrays. Practical Aspects. Fabrication and Behavior. On-Resistance. Voltage Levels in Logic Gates. Noise Margin. Dynamic Operation of Logic Gates. Power Dissipation in Logic Gates.Passing 1s and 0s Through Transistor Switches.Transmission Gates. | [1] p. 733-786 |
|  | ***Seminar №5:*** *Questions and Exercises,* ***Quiz 1(Lec1-Lec4)*** | [1] p. 814-822 |
| ***6*** | ***24.10.17*** | ***Lecture №6.*** *Number Representation and Arithmetic Circuits:* Positional number representation. Unsigned integers. Octal and hexadecimal representations. Addition of unsigned numbers. Decomposed full-adder. Ripple-carry adder. Design example. Signed numbers. Negative numbers. Addition and subtraction. Adder and subtractor unit. Radix-complement schemes. Arithmetic overﬂow. Performance issues. Fast adders. Carry-lookahead adder. | [1] p. 122-151 |
|  | ***Seminar №6:*** *Questions and Exercises* | [1] p. 184-187 |
| ***7*** | ***31.10.17*** | ***Lecture №7.*** *Number Representation and Arithmetic Circuits:* Design of arithmetic circuits using CAD tools. Design of arithmetic circuits using schematic capture. Design of arithmetic circuits using Verilog. Using vectored signals. Using a generic speciﬁcation. Nets and variables in verilog. Arithmetic assignment statements. Module hierarchy in verilog code. Representation of numbers in verilog code. Multiplication. Array multiplier for unsigned numbers. Multiplication of signed numbers. Other number representations. Fixed-point numbers. Floating-point numbers. Binary-coded-decimal representation. | [1] p. 151-178 |
|  | ***Seminar №7:*** *Questions and Exercises* | [1] p. 184-187 |
| ***8*** | ***07.11.17*** | ***Lecture №8****.* *Combinational-circuit building blocks:* Multiplexers. Synthesis of logic functions using multiplexers. Multiplexer synthesis using Shannon’s expansion. Decoders. Demultiplexers. Encoders. Binary encoders. Priority encoders. Code converters. Arithmetic comparison circuits. For combinational circuits. Conditional operator. If-else statement. The case statement. The for loop. Verilog operators. The generate construct. Tasks and functions. | [1] p. 189-232 |
|  | ***Seminar №8:*** *Questions and Exercises,* ***Quiz 2(Lec5-Lec7)*** | [1] p. 243-245 |
| ***9*** | ***14.11.17*** | ***Mid term exam*** |  |
| ***Seminar:*** *Questions and Exercises* |  |
| ***10*** | ***21.11.17*** | ***Lecture №9.*** *Flip-flops, registers, and counters:* Basic latch. Gated SR latch. Gated SR latch with NAND gates. Gated D latch. Effects of propagation delays. Edge-triggered D flip-flops. Master-slave D flip-flop. Other types of edge-triggered D flip-flops. D flip-flops with clear and preset. Flip-flop timing parameters. T flip-flop. Jk flip-flop. Summary of terminology. | [1] p. 247-267 |
|  | ***Seminar №9:*** *Questions and Exercises* | [1] p.321-328 |
| ***11*** | ***28.11.17*** | ***Lecture №10.*** *Flip-flops, registers, and counters:* Registers. Shift register. Parallel-access shift register. Counters. Asynchronous counters. Synchronous counters. Counters with parallel load. Reset synchronization. Other types of counters. BCD counter. Ring counter. Johnson counter. Remarks on counter design. | [1] p. 267-283 |
|  | ***Seminar №10:*** *Questions and Exercises* | [1] p.321-328 |
| ***12*** | ***05.12.17*** | ***Lecture №11.*** *Flip-flops, registers, and counters:* Using storage elements with CAD tools. Including storage elements in schematics. Using Verilog constructs for storage elements. Blocking and non-blocking assignments. Non-blocking assignments for combinational circuits. Flip-flops with clear capability. Using Verilog constructs for registers and counters. Flip-flops and registers with enable inputs. Shift registers with enable inputs. Design example. Reaction timer. Register transfer level (RTL) code. Timing analysis of flip-ﬂop circuits. Timing analysis with clock skew. | [1] p. 283-314 |
|  | ***Seminar №11:*** *Questions and Exercises,* ***Quiz 3(Lec8-Lec11)*** | [1] p.321-328 |
| ***13*** | ***12.12.17*** | ***Lecture №12.*** *Synchronous sequential circuits:*Basic design steps. State diagram. State table. State assignment. Choice of flip-flops and derivation of next-state and output expressions. Timing diagram. Summary of design steps. State-assignment problem. One-hot encoding. Mealy state model. Design of finite state machines using CAD tools. Verilog code for moore-type FSMS. Synthesis of Verilog code. Simulating and testing the circuit. Alternative styles of Verilog code. Summary of design steps when using cad tools. Specifying the state assignment in Verilog code. Speciﬁcation of mealy FSMS using verilog. | [1] p. 331-363 |
|  | ***Seminar №12:*** *Questions and Exercises,* | [1] p. 416-419 |
| ***14*** | ***19.12.17*** | ***Lecture №13.*** *Synchronous sequential circuits:*Serial adder example. Mealy-type FSM for serial adder. Moore-type FSM for serial adder. Verilog code for the serial adder. Minimization. Partitioning minimization procedure. Incompletely speciﬁed FSMS. Design of a counter using the sequential circuit approach. State diagram and state table for a modulo-8 counter. State assignment. Implementation using D-type flip-flops. Implementation using JK-type flip-flops. Example—a different counter. | [1] p. 363-393 |
|  | ***Seminar №13:*** *Questions and Exercises,* | [1] p. 416-419 |
| ***15*** | ***26.12.17*** | ***Lecture №14.*** *Synchronous sequential circuits:*FSMS ss an arbiter circuit. Analysis of synchronous sequential circuits. Algorithmic state machine (ASM) charts. Formal model for sequential circuits. | [1] p. 393-405 |
|  | ***Seminar №14:*** *Questions and Exercises,* ***Quiz 4(Lec11-Lec13)*** | [1] p. 416-419 |
|  |  | ***Final Exam*** |  |