

**Siberian Federal University**

# **05.13.05 - Elements and devices of computer technology and control systems**

## **Course Guide**

This course contributes to the requirements for the Degree of Candidate of Science in Computer Science.

Krasnoyarsk, 2020.

## Table of Contents

1. Course Description	Ошибка! Закладка не определена.
1.1 Course overview	2
1.2 Special features	3
1.3 Course aims and objectives	Ошибка! Закладка не определена.
1.4 Learning outcomes	Ошибка! Закладка не определена.
2. Course Lecturer, Contact Information	Ошибка! Закладка не определена.
3. Prerequisites	Ошибка! Закладка не определена.
4. Course Outline	5
4.1 Course requirements	Ошибка! Закладка не определена.
4.1.1 Web-page of the course	Ошибка! Закладка не определена.
4.1.2 Required reading	7
4.1.3 Course materials	Ошибка! Закладка не определена.
4.1.4 Required feedbacks	Ошибка! Закладка не определена.
4.2 Course Structure	8
4.3 Time schedule of the course and course outline	8
5. Assessment	Ошибка! Закладка не определена.
6. Attendance Policy	Ошибка! Закладка не определена.
7. Required Course Participation	12
8. Facilities, Equipment and Software	Ошибка! Закладка не определена.
Annex 1 Example of Self-Study Assignment	13
Annex 2 Example of Pre-Course Test Questions	Ошибка! Закладка не определена.
Annex 3 Outlines of Lab works	Ошибка! Закладка не определена.
Annex 4 Example of Final Oral Exam Questions	21

# 1. Course Description

**This course contributes to the requirements for the Degree of Candidate of Science in Computer Science.**

<b>Title of the Academic Program</b>	Postgraduate Programs in English “Elements and devices of computer technology and control systems”
<b>Type of the course</b>	core /mandatory
<b>Course period</b>	2 semesters First semester: from October, the 1st to February, the 1st (18 weeks) Second semester: from February, the 1st to June, the 1st (18 weeks)
<b>Study credits</b>	6 ECTS credits
<b>Duration</b>	216 hours
<b>Language of instruction</b>	English
<b>Academic requirements</b>	<ul style="list-style-type: none"><li>– Master's Degree in Computer Science or equivalent (transcript of records),</li><li>– good command of English (certificate or other official document)</li></ul> <b>Prerequisites:</b> <ul style="list-style-type: none"><li>– <b>Advanced knowledge</b> of math, digital electronics, programming skills.</li></ul>

## **1.1 Course overview**

«Elements and devices of computer technology and control systems» is a core course.

The course will advance graduate student skills in science research in field of modern and intelligent control systems. Elements and devices of computer technology and control systems is a course aimed at studying the principles of improving and creating fundamentally new elements and devices of computer technology and control systems. The course includes the following sections: "Digital electronic, elements and devices", "Sensors", "Microprocessor technology and software for microprocessor systems", "Programming logical devices".

In the process of training, graduate students prepare for solving scientific and technical problems to create and improve the theoretical and technical base of computer technology and control systems. The main practical result of training is the willingness of graduate students to carry out scientific research to create elements and nodes of advanced computing systems with high quality and performance indicators that accelerate scientific and technological progress.

Applying computer control systems requires skills in special design techniques and professional CAD. This course familiarizes graduate students with computer control system architecture features and shapes a complete set of skills in computer devices and control system.

The course will use evaluation boards (with sensors and expressions) and measurement equipment, such as Digital oscilloscopes, multimeters etc.

## **1.2 Special features**

The course provides an opportunity to graduate students to work personally using electronic and test equipment. The student will be able to go all the way from the emergence of a science idea to implementation by electronic devices.

## **1.3 Course aims and objectives**

### **Course Aims**

- The aim of the course is to teach graduate students to translate scientific ideas into practice, develop, test and debug electronic systems using computer-aided design and measuring equipment.

### **Course Objectives**

- to familiarize graduate students with principles functioning and architectures modern computer electronic systems;
- to acquaint graduate students with the internal and external interfaces of computer - based devices;

- to teach graduate students to use computer aided tools and techniques for designing applications for computer electronic systems.

### ***1.4 Learning outcomes***

**By the end of the course, graduate students will know:**

- design features and applications of elements of digital electronics;
- knowledge about internal and external computer interfaces;
- technology of low and high-level programming of microprocessor systems;
- general principles and approaches to the development of applications based on VLSI.

**By the end of the course, students will be able to:**

- choose the most suitable class of devices for solving a scientific problem;
- develop an interface diagram for external devices and implement systems based on VLSI;
- develop software for embedded systems designed to solve scientific and practical problems;
- perform modeling, optimization and debugging of projects using CAD tools.

**By the end of the course, students will possess:**

- the necessary skills for the design of computing devices and systems based on electronic components using computer tools and methods.

## 2. Course Lecturer, Contact Information



**Oleg V. Nepomnuashchiy,**

Ph.D. in Engineering, Professor, Head of Computer Science Dept, School of Space and Information Technologies, Siberian Federal University.

(room ULK 3-12B) 26, Kirenskogo st, Krasnoyarsk, Russia

e-mail: ONepomnuashy@sfu-kras.ru

Google Scholar page:

[https://scholar.google.ru/citations?user=JxdeoasAAAAJ&](https://scholar.google.ru/citations?user=JxdeoasAAAAJ&hl=ru)

[hl=ru](https://scholar.google.ru/citations?user=JxdeoasAAAAJ&hl=ru)

Additional information is available at:

<https://structure.sfu-kras.ru/node/2153>

Tel: +7 391 291 2931

## 3. Prerequisites

A background in basic electronic and programming will help in faster and better understanding of every topic. Nevertheless, each part of the course includes a short introduction of methods that are required for its study. Therefore, a graduate student without the denoted experience must be encouraged to make some additional efforts in education.

## 4. Course Outline

Week	Lectures	Seminars/ Assignments	Hours Lec/Lab/HA
<b>Semester 1</b>			
<b>1-10</b>	Digital electronic, elements and devices. Transistor Switches, logic inverters. Boolean Algebra and truth Tables, NOT, AND, OR gates. Digital electronic circuits.	Asynchronous combinational circuits. Clocking and synchronization.	10/8/54
<b>11-18</b>	Microprocessor technology and software for microprocessor systems. 8 <sup>st</sup> -bits RISC microcontroller. System-on-a-chip. Digital Interfaces.	SoC – high – level application design. Using SoC in conjunction with sensors, LCD and power resistance.	8/10/54
<b>Semester 2</b>			
<b>1-9</b>	Sensors. Circuitry of means for interfacing with analog sensors. Digital to analog converters (DAC). Micro-electro-mechanical sensors (MEMS). Intelligent sensors.	Digital remote control. MEMS to serial interface.	4/2/2
<b>10-18</b>	Programming logical devices. VLSI design by CAD. Using Verilog for FPGA project design. FPGA-based Systems-on-a-Chip.	Combinational circuits in Verilog. Typical digital modules for FPGA designs: counters, memory, frequency divider etc. Soft-Processor Nios-II in FPGA designs. Individual project.	14/7/7
	36	27	180
36	Final Exam		36

## 4.1 Course requirements

### 4.1.1 Web-page of the course

Course materials and required reading materials are available on the course webpage [Elements and devices of computer technology and control systems](https://e.sfu-kras.ru/course/view.php?id=27723). The webpage is available through the SibFU E-learning portal [www.e.sfu-kras.ru](http://www.e.sfu-kras.ru). You must be logged in to access this course. <https://e.sfu-kras.ru/course/view.php?id=27723>

### 4.1.2 Required reading

The main book for this course is The **Course Book**. It provides students with all the information they need to master methods and tools for research in science field.

1. John Morris. Digital Electronics. CRC Press; September 2013. p. 132. ISBN: 9781136076862.
2. Charles Platt. Make Electronics – Learning by Discovery by (2nd Edition) Helpful Corporation, 2015. p. 349. ISBN-13: 978-1680450262, ISBN-10: 9781680450262.
3. Paul Scherz. Practical Electronics for Inventors, Fourth Edition. McGraw-Hill Education. 2016. p. 1072. ISBN-13: 978-1259587542, ISBN-10: 1259587541.
4. Steven F. Barrett, Daniel J. Pack. Microchip AVR® Microcontroller Primer: Programming and Interfacing, Third Edition. Synthesis Lectures on Digital Circuits and Systems. Morgan & Claypool Publishers, 2019. p.373. ISBN 1681736233, 9781681736235.
5. Jacob Fraden. Handbook of Modern Sensors. Physics, Designs, and Applications. Springer International Publishing. 2016., p.758. ISBN 978-3-319-19302-1. DOI 10.1007/978-3-319-19303-8
6. John G. Webster. Measurement, Instrumentation, and Sensors Handbook: Spatial, Mechanical, Thermal, and Radiation Measurement. CRC Press; 2 edition (January 29, 2014) p.1640. ISBN-13: 978-1439848883.
7. Cem Unsalan, Bora Tar. Digital System Design with FPGA. Implementation Using Verilog and VHDL. Mc-Graw Hill Education. 2017. p. 609. ISBN 978-1-25-983791-3.
8. M. Rafiquzzaman, Steven A. McNinch. Digital Logic. With an Introduction to Verilog and FPGA-Based Design. John Wiley and Sons Inc. 2020. p. 439. ISBN 13: 978-1-119-62154-6.



### 4.1.3 Course materials

The main book that will guide a student through the course is *Elements and devices of computer technology and control systems* book. It contains all of topics of this course according to the schedule. It will provide you with useful links at the end of each chapter that will help graduate students to improve their understanding of the topics.

### 4.1.4 Required feedbacks

Graduate students are free to contact the lecturer by email. The name of department and a number of a group should be written in the subject or in the beginning of the letter for convenience. More information on how to contact the lecturer can be found in «Lecturer information» section of this Guide.

Student's Home or Lab Assignment reports must be attached as a separate pdf file. Student's name and group number should be written on the first page of the file. It is recommended to insert Assembler or C code with short comments for key elements of the code. Students send this report in electronic form only before the deadline.

If necessary, the lecturer will schedule a video-conference, upon request.

## 4.2 Course Structure

Learning Activities	Hours
Lectures	36
Practice sessions / Seminars,	27
Self-study Assignments	117
Final Exam (including preparation)	36
<b>Total study hours</b>	<b>216</b>

### 4.3 Time schedule of the course and course outline

№	Theme	Week	Learning Activities	Hours	Home Assignment and Reading
<b>Semester 1</b>					
1	Digital electronic	1-10	<b>Lecture 1</b> «Basic of digital electronic» <b>Part 1,2</b>	4	Course Book Chapter 1. Digital electronic. Answer the test questions on the topic in the e-course
			<b>Lecture 2</b> «Digital elements and devices» <b>Part 1,2,3</b>	6	Course Book Chapter 2. Digital elements. Answer the test questions on the topic in the e-course
			<b>Lab 1</b> «Asynchronous combinational circuits in control systems»	8	Design and debug application according to tasks. Choose a number of tasks in Course Book annex 1-1.
			<b>Home assignment 1</b>	54	Read the getting started with OrCAD. Read the books: John Morris. Digital Electronics. Charles Platt. Make Electronics – Learning by Discovery. Finish the Lab 1 and issue a report about.
2	Microprocessor technology	11-18	<b>Lecture 3</b> «Microprocessors, microcontrollers and SoC» <b>Part 1,2,3</b>	6	Course Book: Chapter 3 Microprocessor`s systems. Answer the test questions on the topic in the e-course
			<b>Lecture 4</b> «Digital Interfaces»	2	Course Book: Chapter 4 Digital Interfaces. Answer the test questions on the topic in the e-course

№	Theme	Week	Learning Activities	Hours	Home Assignment and Reading
			Lab 2 «SoC – high – level application design»	10	Design and debug application according to tasks. Choose a number of tasks in Course Book annex 1-2.
			Home assignment 2	54	Read the book: Steven F. Barrett, Daniel J. Pack. Microchip AVR® Microcontroller Primer: Programming and Interfacing. Finish the Lab 2 and issue a report about.
Semester 2					
3	Sensors	1-9	Lecture 5 The theory of Digital to analog converters	2	Course Book: Chapter 3 Sensors 3.1 Analog vs Digital. Answer the test questions on the topic in the e-course.
			Lecture 6 «Smart sensors»	2	Course Book: Chapter 3 Sensors 3.2 Smart Sensors. Answer the test questions on the topic in the e-course
			Lab 3 «Digital remote control»	2	Design and debug application according to tasks. Choose a number of tasks in Course Book annex 1-3.
			Home assignment 3	2	Read the books: Jacob Fraden. Handbook of Modern Sensors. Physics, Designs, and Applications. John G. Webster. Measurement,

№	Theme	Week	Learning Activities	Hours	Home Assignment and Reading
					Instrumentation, and Sensors Handbook. Finish the Lab 3 and issue a report about.
4	Programming logical devices.	10-18	<b>Lecture 7</b> «VLSI – based system design» <b>Part 1-7</b>	14	Course Book: Chapter 5 VLSI. Answer the test questions on the topic in the e-course
			<b>Lab 4</b> «Typical digital modules for FPGA designs»	3	Design and debug application according to tasks. Choose a number of tasks in Course Book annex 1-4, 1-5.
			<b>Lab 5</b> «Processor Nios-II in FPGA applications»	4	
			<b>Home assignment 4</b>	7	Read the books: Cem Unsalan, Bora Tar. Digital System Design with FPGA. Implementation Using Verilog and VHDL. M. Rafiquzzaman, Steven A. McNinch. Digital Logic. With an Introduction to Verilog and FPGA-Based Design. Finish the Lab 4,5 and issue a report about.
5	Final exam			36	Prepare to final exam. Preparation for answering exam questions (available at e-courses and course book). Preparation for solving control problems using the course book, main books and the e-course.

## 5. Assessment

Assessment strategy	Points, max	Evaluation criteria
Tests	10	Test questions for lectures in the e-course
Lab works	40	Lab report
Individual Project	40	Electrical schematics, code, report on the project, presenting the project
Final exam	10	2 questions and a practical task that require preparatory reading and knowledge of the concepts explained

Grade policy for final assessment is:

A (excellent work) 91–100 points

B (above average work) 81–90 points

C (average work) 71–80 points

D (below average work) 50–70 points

F (failed work) < 50 points

The final exam is oral and written test. Students should be able to:

- Answer two short theoretical questions;
- Develop a general algorithm for embedded software according to the assignment;
- Write a fragment of the program to initialize the built-in nodes of the microcontroller.

## 6. Attendance Policy

Graduate students are expected to attend classes regularly. In case of missing an in-lab activity a student should perform additional work submitted to the instructor within a week after a class was missed.

Every topic involves an assignment. A written report on the assignment should be submitted within two weeks from the moment students received a list of problems. The final mark will rely on the same grading policy as for the final exam.

## 7. Required Course Participation

There are no special requirements for the course participation. The preferred type of report submission is the electronic one. Students can use the web-version of the course (link) for a better progress. All problems for solution could be found there together with text from the course book.

## 8. Facilities, Equipment and Software

### **Software:**

Atmel Studio 7, Free, no license required;

The Intel® Quartus® Prime Lite Edition. Free, no license required.

Proteus Virtual System Modelling (VSM);

Microsoft Office®.

### **Laboratory equipment:**

Atmel STK 500, STK501, STK600 Evolution board;

Arduino Uno Evolution board;

DE2-115 – Cyclone IV (EP4CE115F29C7N) Evolution board;

DE-1-SoC- Cyclone IV (5CSEMA5F31C6N) Evolution board;

Sensors, actuators and connectors – bag.

### **Control, testing and measuring equipment:**

Digital oscilloscopes PV6501, GW Instek GDS-8205, Tektronix TPS 2024;

Multimetr ABM-4307;

Signal generator GW Instek SFG-2010.

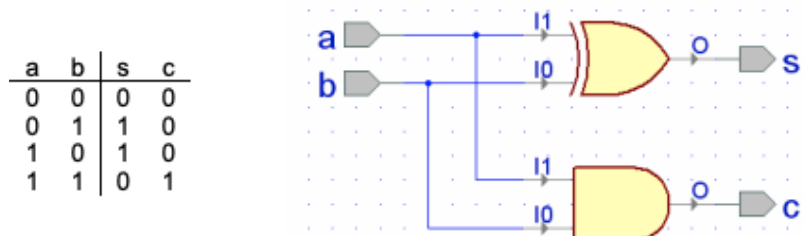
## Annex 1 Example of Self-Study Assignment

**The task:** Develop the FULL ADDER Using Verilog HDL.

### 1. Solution:

#### 1.1 Half Adder

The truth table for a half adder is shown in Fig. 1. In this table bit a is added to bit b to produce the sum bit s and the carry bit c. Note that if we add 1 to 1 we get 2, which in binary is 10 or 0 with a carry bit. The BDE logic diagram, halfadd.bde, for a half adder is also shown in Fig. 1. Note that the sum s is just the exclusive-or of a and b and the carry c is just a & b. The Verilog program corresponding to the circuit in Fig. 1 is shown in Listing 1. A simulation of halfadd.bde is shown in Fig. 2.



**Figure 1. Truth table and logic diagram halfadd.bde for a half-adder**

#### Listing 1. halfadd.v

```
// Example 1a: halfadd
module halfadd (
    input wire a,
    input wire b,
    output wire c,
    output wire s
);
    assign s = b ^ a;
    assign c = b & a;
endmodule
```

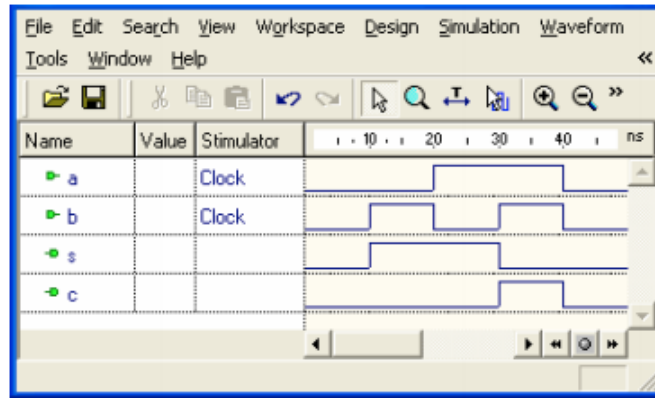


Figure 2. Simulation of the half-adder in Fig. 1

## 12.2 Full Adder

When adding binary numbers we need to consider the carry from one bit to the next. Thus, at any bit position we will be adding three bits:  $a_i$ ,  $b_i$  and the carry-in  $c_i$  from the addition of the two bits to the right of the current bit position. The sum of these three bits will produce a sum bit,  $s_i$ , and a carry-out,  $c_{i+1}$ , which will be the carry-in to the next bit position to the left. This is called a full adder and its truth table is shown in Fig. 3. The results of the first seven rows in this truth table can be inferred from the truth table for the half adder given in Fig. 1. In all of these rows only two 1's are ever added together. The last row in Fig. 3 adds three 1's. The result is 3, which in binary is 11, or 1 plus a carry. From the truth table in Fig. 3 we can write a sum of products expression for  $s_i$  as

$$s_i = \begin{aligned} &\sim c_i \& \sim a_i \& b_i \\ &| \sim c_i \& a_i \& \sim b_i \\ &| c_i \& \sim a_i \& \sim b_i \\ &| c_i \& a_i \& b_i \end{aligned} \quad (1)$$

$c_i$	$a_i$	$b_i$	$s_i$	$c_{i+1}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Figure 3 Truth table for a full adder

We can use the distributive law to factor out  $\sim c_i$  from the first two product terms and  $c_i$  from the last two product terms in Eq. (12.1) to obtain

$$s_i = \begin{aligned} &\sim c_i \& (\sim a_i \& b_i \mid a_i \& \sim b_i) \\ &| c_i \& (\sim a_i \& \sim b_i \mid a_i \& b_i) \end{aligned} \quad (2)$$

which can be written in terms of XOR and XNOR operations as



$$s_i = \sim c_i \& (a_i \wedge b_i) \mid c_i \& \sim(a_i \wedge b_i) \quad (3)$$

which further reduces to

$$s_i = c_i \wedge (a_i \wedge b_i) \quad (4)$$

Fig. shows the K-map for  $c_{i+1}$  from the truth table in Fig. 3. The map shown in Fig. 4a leads to the reduced form for  $c_{i+1}$  given by

$$c_{i+1} = a_i \& b_i \mid c_i \& b_i \mid c_i \& a_i \quad (5)$$

While this is the reduced form, a more convenient form can be written from Fig. 4b as follows:

$$\begin{aligned} c_{i+1} &= a_i \& b_i \mid c_i \& \sim a_i \& b_i \mid c_i \& a_i \& \sim b_i \\ &= a_i \& b_i \mid c_i \& (\sim a_i \& b_i \mid a_i \& \sim b_i) \\ &= a_i \& b_i \mid c_i \& (a_i \wedge b_i) \end{aligned} \quad (6)$$

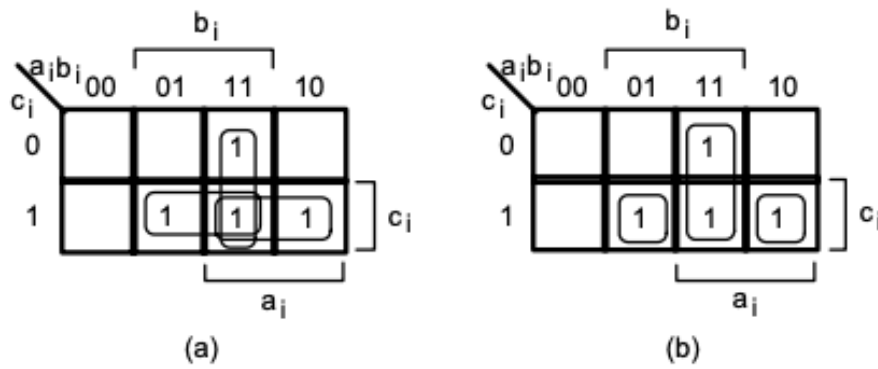


Figure 4. K-maps for  $c_{i+1}$  for full adder in Fig. 6.2

From Eqs. (4) and (6) we can draw the logic diagram for a full adder as shown in Fig. 5. Comparing this diagram to that for a half adder in Fig. 1 it is clear that a full adder can be made from two half adders plus an OR gate as shown in Fig. 6.

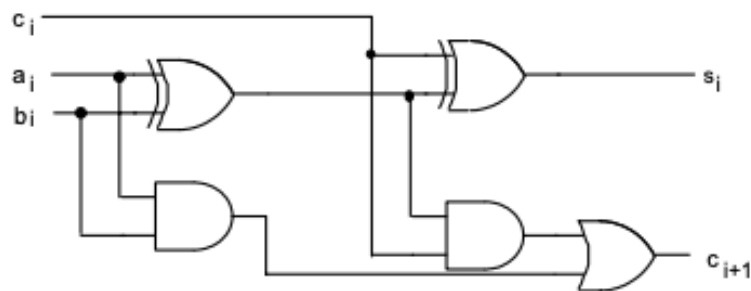


Figure 5. Logic diagram for a full adder

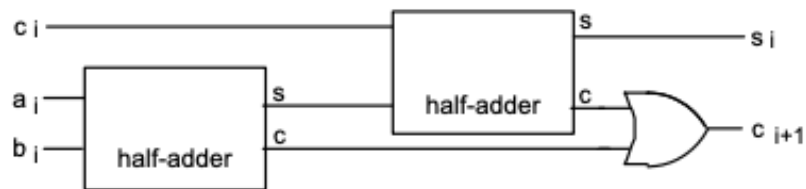


Figure 6. A full adder can be made from two half adders plus an OR gate

From Fig. 6 we can create a BDE design, fulladd.bde, as shown in Fig. 7. The Verilog program resulting from compiling this design is equivalent to that shown in Listing 2. A simulation of this full adder is shown in Fig. 8. Note that the outputs agree with the truth table in Fig. 3.

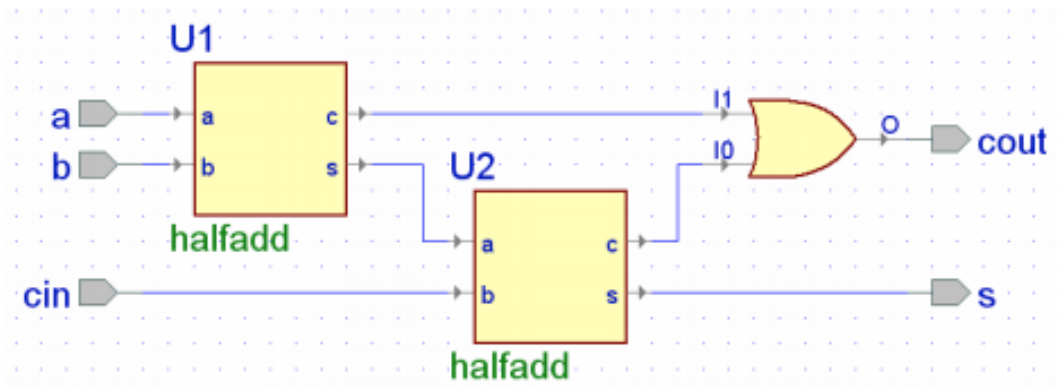


Figure 7. Block diagram fulladd.bde for a full adder

Listing 2. fulladd.v

```
// Example 1b: fulladd
module fulladd (
input wire a,
input wire b,
input wire cin,
output wire cout,
output wire s
);
wire c1;
wire c2;
wire s1;

assign cout = c2 | c1;

halfadd U1
(
.a(a),
.b(b),
.c(c1),
```

```

        .s(s1)
    );
halfadd U2 (
    .a(s1),
    .b(cin),
    .c(c2),
    .s(s)
);
endmodule

```

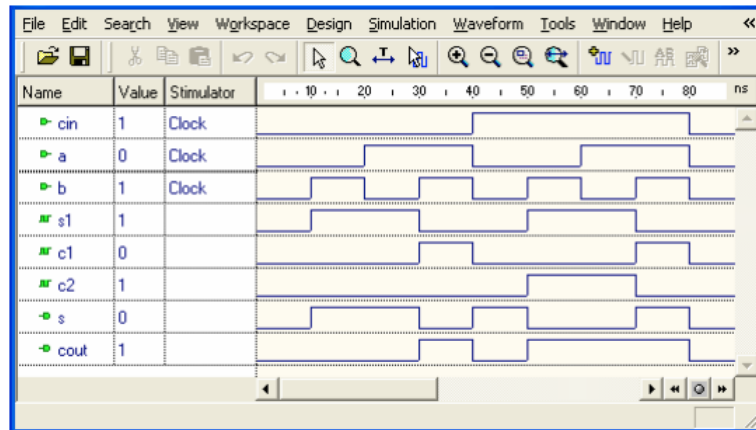


Figure 8. Simulation of the full adder in Fig. 7 and Listing 2

## **Annex 2 Example of Pre-Course Test Questions**

1. Synchronous logic gates. Special and multifunctional elements. Logic expanders. Digital switches and keys.
2. Sensors. Purpose, main types of sensors and physical principles of operation.
3. Analog-to-digital and digital-to-analog converters. Construction principles. Main characteristics and parameters.
4. Serial interfaces: RS232C, I2C, USB,
5. logical elements, decoders, encoders, code converters, adders, triggers,
6. Integrated microcircuits for storage devices (ROM, RAM, EPROM). Comparative evaluation of characteristics of RAM, SRAM, EPROM
7. Microprocessor-based information processing facilities in control systems.
8. Combination nodes of the computer: decoders, encryptors, priority encryptors,
9. Synchronization systems. Synchronous and asynchronous data exchange.
10. Radiation resistance of elements and devices. Types of influencing radiation: corpuscular, quantum, wave. Reversible and residual effects.
11. Electrically programmable ROM, ROM with burnout jumpers. Elements of reprogrammable ROM.
12. Programmable integrated circuits. Classification. FPGA basic concepts and definitions. Architecture of modern FPGAs.
13. Microcontrollers and systems on a chip. Classification. The functional composition of a typical 8-bit microcontroller.

## Annex 3 Outlines of Lab works

**(List one. The title)**

"SIBERIAN FEDERAL UNIVERSITY"

Institute of Space and Information Technologies

Department of Computer Science

Master's Degree Programs "Digital intelligent control systems"

Group No **(Group identifier)**

REPORT ON LABORATORY WORK No. **(Number of lab)**

Theme: **(Theme of task).**

Tutor: **(Tutor's / Lecture's Name and Surname).**

Student: **(Student's Name and Surname).**

Krasnoyarsk, 2020

**(List two, etc. The progress)**

Main aim: **(Describe the aim of lab).**

The task: **(Describe the task of lab).**

Solution: **(short description (no more than 2-3 pages) of the problem solving process).**

Annex A Diagram(s)

**(device connection diagrams, oscillograms and graphs).**

Annex B Code(s)

**(source code Included comment).**

## **Annex 4 Example of Final Oral Exam Questions**

1. Describe the functional composition of the simplest microprocessor system. What microprocessors system consists of? Describe the main ways of allocating address space in microprocessor systems.
2. Timers – counters of microcontrollers - review. PWM timer mode. Give an example of choose the frequency and duty cycle pulses in PWM mode.