

МИНОБРНАУКИ РОССИИ
Ярославский государственный университет им. П.Г. Демидова

Кафедра теоретической информатики

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

« 23 » мая 2023 г.

Рабочая программа дисциплины
«Архитектура вычислительных систем»

Направление подготовки
02.03.02 Фундаментальная информатика и информационные технологии

Профиль
«Информатика и компьютерные науки»

Квалификация выпускника
Бакалавр

Форма обучения
очная

Программа рассмотрена
на заседании кафедры
от 12 апреля 2023 г.,
протокол № 10

Программа одобрена НМК
факультета ИВТ
протокол № 6 от
28 апреля 2023 г.

Ярославль

1. Цели освоения дисциплины

Целью освоения дисциплины «Архитектура вычислительных систем» является изучение технических и логических основ вычислительной техники; изучение структурной организации и принципов функционирования основных компонентов компьютеров; освоение принципов программирования на ассемблере.

Основной направленностью дисциплины является формирование системотехнического мировоззрения, развивающего способность ориентироваться и разбираться в многообразии технических средств и конфигураций современных компьютеров. Студенты должны быть готовы использовать полученные в этой области знания как при изучении смежных дисциплин, так и в профессиональной деятельности.

2. Место дисциплины в структуре ОП бакалавриата

Дисциплина «Архитектура вычислительных систем» относится к базовой части ОП бакалавриата. Она основывается на знаниях, полученных слушателями при изучении таких математических дисциплин, как «Дискретная математика», «Информатика».

3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП бакалавриата

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
Общепрофессиональные компетенции		
ОПК-5 Способен устанавливать и сопровождать программное обеспечение информационных систем и баз данных, в том числе отечественного происхождения, с учетом информационной безопасности	ОПК-5.1 Знает технические и логические основы вычислительной техники ОПК-5.2 Знает принципы устройства операционных систем, приёмов и методики их администрирования	Знать: – принципы логической и технической организации вычислительных машин; – основы функционирования компонентов вычислительных машин, принципы аппаратного и программного управления компонентами; – принципы программирования на ассемблере. Уметь: – оценивать техническую конфигурацию, состав и основные характеристики вычислительных машин; – проектировать общую схему и описывать

		<p>отдельные блоки дискретного преобразователя;</p> <p>– создавать проект на языке C/C++, содержащий ассемблерный код, в среде Microsoft Visual Studio.</p> <p>Владеть навыками:</p> <p>– оценки, выбора и конфигурирования технических средств в составе компьютерных систем;</p> <p>– синтеза сетей из функциональных элементов;</p> <p>– программирования на ассемблере.</p>
--	--	--

4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 3 зач.ед., 108 акад.час.

№ п/п	Темы (разделы) дисциплины, их содержание	Се м е ст р	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)					Формы текущего контроля успеваемости		Форма промежуточной аттестации (по семестрам)
			лек ц и и	пра кти чес кие	лаб ора тор ны е	кон сул ьта ции	ат те ст ац ио нн ые ис пы та ния	сам осто ятел ьна я раб ота		
			Контактная работа							
1	Техническая организация компьютера	2	10			1			Тест в LMS «Moodle ЯрГУ»	
2	Представление информации и избыточное кодирование	2	2	6		2			Контрольная работа №1. Коллоквиум	
3	Проектирование дискретных преобразователей	2	1	12		2			Контрольная работа №2. Коллоквиум	
4	Основы ассемблера	2	4	16		2		14	Индивидуальное	

								здание
		2				2	0,5	Экзамен
	Всего за 2 семестр		17	34		9	0.5	14
	Всего		17	34		9	0.5	14

Содержание разделов дисциплины:

Тема 1. Техническая организация компьютера.

1.1. Общие принципы функционирования компьютеров: Архитектура фон Неймана. Вычислительные системы. Структура, архитектура. Открытые и замкнутые системы. Функционирование ЭВМ. Процесс и поток. Представление различных видов информации в компьютере. Структуры данных.

1.2. Оперативная память: Классификация элементов памяти. Физические принципы построения. Матричная организация элементов памяти. Кэширование памяти. Архитектура кэш-памяти.

1.3. Центральный процессор: Исполнение программного кода. Переключение задач и виртуальные машины. Защищённый режим и виртуальная память. Архитектура и микроархитектура процессоров. Конвейеризация. Режимы работы процессоров. Архитектурные регистры и типы данных. Набор инструкций. События – прерывания и исключения. Эффективный адрес и преобразование адресов. Страничная трансляция адресов и виртуальная память. Мультипроцессорные и избыточные системы.

1.4. Системная шина: Информационная магистраль первого поколения – шина ISA. Информационная магистраль второго поколения – шина PCI. Информационная магистраль третьего поколения – шина PCI-Express.

1.5. Жесткий диск: Принципы магнитной записи и физическое устройство жёсткого диска. Системная организация HDD. Интерфейсы устройств хранения. RAID-массивы. Логическая структура дисков. Файловая система. SSD-накопитель.

1.6. Видеосистема.

Тема 2. Представление информации и избыточное кодирование.

Прямой, обратный, дополнительный и модифицированный дополнительный код. Код Хэмминга. CRC-кодирование.

Тема 3. Проектирование дискретных преобразователей.

Принципы разработки логических блоков вычислительного устройства с помощью сетей из функциональных элементов.

Тема 4. Основы ассемблера.

Создание проекта с ассемблерным кодом в среде Microsoft Visual Studio. Типы данных, переменные и операнды в ассемблере. Основные целочисленные команды ассемблера. Условный переход и циклы в ассемблере. Ассемблерные вставки, ассемблерные процедуры и их вызов.

5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине

В процессе обучения используются следующие образовательные технологии:

Вводная лекция – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

Академическая лекция (или лекция общего курса) – последовательное изложение

материала, осуществляемое преимущественно в виде монолога преподавателя.

Требования к академической лекции: современный научный уровень и насыщенная информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

Практическое занятие – занятие, посвященное освоению конкретных умений и навыков и закреплению полученных на лекции знаний.

Консультации – групповые занятия, являющиеся одной из форм контроля самостоятельной работы студентов. На консультациях по просьбе студентов рассматриваются наиболее сложные моменты в решении задач, которые возникают у них в процессе самостоятельной работы, обсуждаются результаты решения заданий, выполненных студентами самостоятельно.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости)

В процессе осуществления образовательного процесса используются:

- для формирования текстов материалов для промежуточной и текущей аттестации – программы Microsoft Office, издательская система LaTeX;
- для поддержки организации обучения – электронная образовательная среда университета «Электронный университет Moodle ЯрГУ»; в частности, в «Moodle ЯрГУ» выкладываются конспекты пройденных тем, презентации по отдельным разделам курса, индивидуальные задания и ссылки на видеолекции по теме 4; там же организуется тест по теме 1;
- для размещения видеолекций – видеохостинг YouTube;
- для организации процесса проверки и сдачи индивидуальных заданий – среда Microsoft Visual Studio;
- для поиска учебной литературы библиотеки ЯрГУ – Автоматизированная библиотечная информационная система "БУКИ-NEXT" (АБИС "Буки-Next").

7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

а) основная:

1. Рублев, В. С., Алгоритмы. Машины Тьюринга, проверка истинности булевых функций, эффективная реализация множеств на компьютере : (индивидуальная работа №10 по дисциплине "Основы дискретной математики") : метод. указания / В. С. Рублев ; Яросл. гос. ун-т, Ярославль, ЯрГУ, 2010, 43с
2. Рублев, В. С., Алгоритмы. Машины Тьюринга, проверка истинности булевых функций, эффективная реализация множеств на компьютере [Электронный ресурс] : (индивидуальная работа №10 по дисциплине "Основы дискретной математики") : метод. указания / В. С. Рублев ; Яросл.го, Ярославль, ЯрГУ, 2010, 43с <http://www.lib.uniyar.ac.ru/edocs/iuni/20190401.pdf>
3. Архитектура информационных систем : учебник для вузов / Б. Я. Советов и др., М., Академия, 2012, 284с
4. Пятибратов, А. П., Вычислительные системы, сети и телекоммуникации : учеб. пособие для вузов / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко; под ред. А. П. Пятибратова, М., КНОРУС, 2013, 372с

б) дополнительная:

1. Курчидис, В. А., Функционально-логические узлы ЭВМ : учеб. пособие для вузов / В. А. Курчидис, С. М. Магдалинский, Д. И. Асеев ; Яросл. гос. ун-т, Ярославль,

ЯрГУ, 1991.

в) ресурсы сети «Интернет»

1. Электронная библиотека учебных материалов
ЯрГУ

(http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php).

2. Елена Вставская. Язык ассемблера. URL: <https://prog-cpp.ru/category/asm-posts/>

8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

– учебные аудитории для проведения занятий лекционного типа и практических занятий;

– учебные аудитории для проведения групповых и индивидуальных консультаций,

– учебные аудитории для проведения текущего контроля и промежуточной аттестации;

– помещения для самостоятельной работы;

– помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, хранящиеся на электронных носителях и обеспечивающие тематические иллюстрации, соответствующие рабочим программам дисциплин.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий – списочному составу группы обучающихся.

Автор:

Доцент кафедры теоретической информатики, к.ф.-м.н.

_____ (подпись)

А. В. Смирнов

**Приложение №1 к рабочей программе дисциплины
«Архитектура вычислительных систем»**

**Фонд оценочных средств
для проведения текущей и промежуточной аттестации студентов
по дисциплине**

**1. Типовые контрольные задания или иные материалы,
необходимые для оценки знаний, умений, навыков и (или) опыта
деятельности, характеризующих этапы формирования компетенций**

**1.1 Контрольные задания и иные материалы,
используемые в процессе текущей аттестации**

Тест по теме 1

Тест формируется автоматически в LMS «Moodle ЯрГУ» из банка вопросов, охватывающего основные разделы темы 1. Задание теста содержит 8 вопросов (однозначный выбор, множественный выбор, числовой ответ), на прохождение даётся 20 минут.

Пример варианта теста:

1. Кванты времени используются при ... Выберите один ответ:
 1. многопроцессорной обработке
 2. любом виде мультипрограммирования
 3. невывесняющей многозадачности
 4. вытесняющей многозадачности
2. Почему статическая и динамическая память получили такие названия? Выберите один ответ:
 1. Динамическая память требует регенерации заряда, статическая хранит информацию, пока есть питание
 2. Динамическая память работает быстрее статической
 3. Данные в динамической памяти могут меняться, в статической они постоянны
 4. Динамическая память энергозависима, статическая — нет
3. Какие методы динамического исполнения помогают бороться с проблемой зависимости инструкций по данным?
Выберите один или несколько ответов:
 1. Продвижение данных.
 2. Спекулятивное исполнение.
 3. Переименование регистров.
 4. Предсказание ветвлений.
 5. Исполнение с изменением порядка инструкций.
4. Какое максимальное число сегментов можно определить в таблице дескрипторов сегментов?
5. В каких поколениях шин реализован многшинный подход? Выберите один или несколько ответов:
 1. В первом.
 2. Во втором.
 3. В третьем.
6. Выберите интерфейсы, при работе с которыми используются «интеллектуальные» контроллеры на устройстве.
Выберите один или несколько ответов:

1. SATA.
 2. SAS.
 3. ST-506.
 4. IDE.
 5. FDD.
 6. SCSI.
7. Какова разрядность адреса LBA при использовании метода разметки GPT?
8. Какой вариант мультидисплейной системы позволяет получить одинаковую картинку на обоих мониторах?
- Выберите один ответ:
1. DualHead DVDMax.
 2. DualHead Clone.
 3. DualHead Zoom.
 4. Extended Desktop.

Контрольная работа № 1 (тема 2)

1. Найти в дополнительном коде значение выражения $y = a + b - c$, если $a = 86; b = -25; c = 100$.

Каждое число кодируется с помощью 8 разрядов.

2. Закодировать с помощью кода Хэмминга последовательность
11101100110111.

Для этой же последовательности произвести декодирование.

3. Определить контрольную сумму (CRC) для сообщения 100000000000001 и полинома (4, 1, 0).

Способы решения задач из контрольной работы №1 рассмотрены в книге [1] из списка основной литературы.

Кроме того, результаты решения заданий обсуждаются на консультациях по просьбе студентов.

Контрольная работа № 2 (тема 3)

1. Для входного n -разрядного положительного числа x , записанного в прямом коде, выходное число z получается как результат уменьшения числа x на 1. Реализовать сеть из функциональных элементов, оценить сложность. Вычитание производить в прямом коде.

2. Для входного 9-разрядного двоичного кода x выходной разряд z равен 1, если число 1 в коде в два раза больше числа 0; в противном случае z равен 0. Реализовать сеть из функциональных элементов, оценить сложность для произвольного числа разрядов n .

Способы решения задач из контрольной работы №2 рассмотрены в книгах [2]–[3] из списка основной литературы.

Кроме того, результаты решения заданий обсуждаются на консультациях по просьбе студентов.

Индивидуальное задание по теме 4

Общие требования к решению:

В среде Microsoft Visual Studio требуется реализовать консольное приложение в соответствии с индивидуальным заданием. Проект должен содержать два модуля:

- код оболочки на языке C или C++ (файл .cpp);
- код основной процедуры на ассемблере (файл .asm).

Оболочка отвечает за ввод-вывод данных и проверку введённых значений. Данные вводятся через консоль и выводятся на неё же.

Выполнять основные действия в коде C/C++ строго запрещено.

Основная процедура должна быть полностью реализована на ассемблере. Вызов процедуры из кода C осуществляется через объявленную внешнюю процедуру с использованием конвенции языка C.

Параметры передаются через стек, возвращаемое значение, если оно требуется по заданию, передаётся через регистр EAX.

Ассемблерный код должен быть тщательно комментирован.

Варианты индивидуальных заданий:

Вариант 1

Входные данные:

- $1 < n \leq 30$;
- одномерный массив *arr* из *n* элементов;
- параметр *p*: $0 < p < n$.

Основная функция: выполнить циклический сдвиг элементов массива на *p* элементов вправо.

Выходные данные: преобразованный массив *arr*.

Вариант 2

Входные данные:

- $0 < n \leq 29$;
- одномерный массив *arr* из *n* элементов;
- параметр *p*: $0 \leq p \leq n$;
- число *b*.

Основная функция: вставить элемент *b* на *p*-ое место в массиве; вернуть новое количество элементов массива.

Выходные данные: преобразованный массив *arr*.

Вариант 3

Входные данные:

- $0 < n \leq 30$;
- одномерный массив *arr* из *n* элементов.

Основная функция: переместить минимальный элемент массива в начало (если их несколько, переместить последний).

Выходные данные: преобразованный массив *arr*.

Вариант 4

Входные данные:

- $0 < n \leq 30$;
- одномерный массив *arr* из *n* элементов.

Основная функция: если нечётных значений больше чем чётных, найти их сумму; в противном случае найти сумму чётных значений.

Выходные данные: значение суммы.

Вариант 5

Входные данные:

- $1 < n \leq 30$;
- одномерный массив *arr* из *n* элементов;
- параметр *p*: $0 < p < \lfloor n/2 \rfloor$.

Основная функция: выполнить циклический сдвиг элементов массива с нечётными индексами на *p* элементов влево.

Выходные данные: преобразованный массив *arr*.

Вариант 6

Входные данные:

- $0 < n \leq 30$;
- одномерный массив *arr* из *n* элементов.

Основная функция: сдвинуть все отрицательные значения в конец массива.

Выходные данные: преобразованный массив *arr*.

Вариант 7

Входные данные:

- $1 < n \leq 30$;
- одномерный массив *arr* из *n* элементов;
- параметр *p*: $1 < p \leq n$.

Основная функция: в каждом *p*-ом элементе разместить сумму предшествующего (*p* – 1) элемента, умноженную на *p*. Вернуть число со значением 0 или 1: 1, если в каком-то действии произошёл переход за 32 разряда (в регистр EDX), 0 в противном случае

Выходные данные: преобразованный массив *arr* и возвращаемое значение функции.

Вариант 8

Входные данные:

- $0 < n \leq 30$;
- одномерный массив *arr* из *n* элементов.

Основная функция: определить, каких значений – положительных, отрицательных или нулевых – в массиве больше. Учесть все варианты, когда условие выполнено сразу для нескольких видов значений. Закодировать ответ в виде числа и вернуть его.

Выходные данные: интерпретация полученного числа (текстовый ответ).

Вариант 9

Входные данные:

- $0 < n \leq 30$;
- одномерный массив *arr* из *n* элементов.

Основная функция: найти и вернуть разность между максимальным и минимальным элементами массива.

Выходные данные: разность.

Вариант 10

Входные данные:

- $0 < n \leq 30$;
- одномерный массив *arr* из *n* элементов.

Основная функция: найти и вернуть сумму неповторяющихся элементов массива.

Выходные данные: сумма.

Вариант 11

Входные данные:

- $0 < n \leq 30$;
- одномерный массив *arr* из *n* элементов;
- целочисленный параметр $p \geq 0$.

Основная функция: срезать элементы массива, модуль которых больше *p*, до +*p* или –*p* соответственно.

Выходные данные: преобразованный массив *arr*.

Вариант 12

Входные данные:

- $0 < n \leq 30$;
- одномерный массив *arr* из *n* элементов.

Основная функция: если длина массива чётна, переставить половины местами, иначе – обнулить каждый третий элемент.

Выходные данные: преобразованный массив *arr*.

Вариант 13

Входные данные:

- $0 < n \leq 30$;
- одномерный массив *arr* из *n* элементов.

Основная функция: выполнить сортировку каждой четвёрки элементов по возрастанию; неполная четвёрка (если она есть) тоже сортируется.

Выходные данные: преобразованный массив *arr*.

Вариант 14

Входные данные:

- $0 < n \leq 30$;
- одномерный массив *arr* из *n* элементов.

Основная функция: если минимальный элемент массива меньше нуля, сменить знак у всех элементов массива, в противном случае вычесть из всех элементов значение минимума.

Выходные данные: преобразованный массив *arr*.

Вариант 15

Входные данные:

- $0 < n \leq 30$;
- одномерный массив *arr* из *n* элементов;
- параметр *p*: $0 \leq p \leq 31$.

Основная функция: найти и вернуть количество элементов массива, в двоичной записи которых на *p*-ом месте стоит 1.

Выходные данные: количество.

Вариант 16

Входные данные:

- $0 < n \leq 30$;
- одномерный массив *arr* из *n* элементов;
- целочисленный параметр *p*.

Основная функция: если чисел, равных *p*, в массиве больше половины, то заменить все такие числа на нули, а нули – на *p*.

Выходные данные: преобразованный массив *arr*.

Вариант 17

Входные данные:

- $0 < n \leq 30$;
- одномерный массив *arr* из *n* элементов.

Основная функция: записать все нечётные значения в массиве в обратном порядке.

Выходные данные: преобразованный массив *arr*.

Способы решения задач по программированию на ассемблере рассмотрены в книге [4] из списка основной литературы.

Кроме того, результаты решения заданий обсуждаются на консультациях по просьбе студентов.

Коллоквиум

1. Закодировать с помощью кода Хэмминга последовательность

11101101010110111.

Для этой же последовательности произвести декодирование.

2. Для кодирования входного алфавита $\Sigma = \{a_1, \dots, a_n\}$ используются только числа, являющиеся степенями тройки. Реализовать сеть шифратора для $n = 4$, оценить сложность полученной сети. В случае произвольного n оценить сложность сети с точностью до упрощения СДНФ.

3. Для входного 8-разрядного числа x выходной разряд z равен 1, если удвоенная сумма четырехразрядных чисел, на которые можно разделить входной код, больше исходного кода x . В противном случае z равен 0. Реализовать сеть из функциональных элементов, оценить ее сложность для $n = 8$ и в случае произвольного n .

Способы решения задач коллоквиума рассмотрены в книгах [1]–[3] из списка основной литературы.

Кроме того, результаты решения заданий обсуждаются на консультациях по просьбе студентов.

1.2 Список вопросов для проведения промежуточной аттестации Список вопросов к экзамену:

1. Представление числовой информации.
2. Представление различных видов информации (кроме числовой). Структуры данных.
3. Архитектура фон Неймана.
4. Вычислительные системы. Открытые и замкнутые системы. Процесс и поток.
5. Классификация элементов памяти. Физические принципы построения.
6. Матричная организация элементов памяти.
7. Кэширование памяти.
8. Архитектура кэш-памяти.
9. Исполнение программного кода. Переключение задач и виртуальные машины. Защищённый режим и виртуальная память.
10. Архитектура и микроархитектура процессоров. Конвейеризация.
11. Режимы работы процессоров.
12. Архитектурные регистры и типы данных.
13. Набор инструкций. События – прерывания и исключения.
14. Эффективный адрес и преобразование адресов.
15. Страничная трансляция адресов и виртуальная память.
16. Мультипроцессорные и избыточные системы.
17. Информационная магистраль первого поколения – шина ISA.
18. Информационная магистраль второго поколения – шина PCI.
19. Информационная магистраль третьего поколения – шина PCI-Express.
20. Физическое устройство жёсткого диска. HDD и SSD.
21. Системная организация HDD. Интерфейсы устройств хранения.
22. RAID-массивы.
23. Логическая структура дисков. MBR и GPT.
24. Файловая система.
25. Видеосистема.

Темы практических заданий к экзамену:

1. Представление информации в вычислительных системах:
Представление числовой информации. Двоичная система счисления и вычисления в ней. Прямой, обратный, дополнительный и модифицированный дополнительный код.

2. Избыточное кодирование информации:

Код Хэмминга. CRC-кодирование.

3. Сети из функциональных элементов:

Синтез сетей из функциональных элементов. Оценка сложности и оптимизация.

Макет экзаменационного билета

Утверждаю:

Зав. кафедрой

д.ф.-м.н., профессор

_____ В.А. Соколов

«___» _____ 20__г.

02.03.02 1 курс

ФГБОУ ВО «Ярославский государственный университет им. П.Г. Демидова»

Фундаментальная информатика и информационные технологии

Кафедра теоретической информатики

Дисциплина «Архитектура вычислительных систем»

Билет № 1

1. Архитектура фон Неймана.

2. Решить задачу.

Для входного 9-разрядного числа x выходной разряд z равен 1, если трехразрядные числа, на которые можно разделить входной код, образуют невозрастающую последовательность. В противном случае z равен 0. Реализовать сеть из функциональных элементов, оценить ее сложность для $n = 9$ и в случае произвольного n .

Разработал:

Доцент кафедры теоретической информатики

к.ф.-м.н. _____ А.В. Смирнов.

Рассмотрены и одобрены на заседании кафедры

«__» _____ 20__г.

Протокол № _

2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания

2.1. Шкала оценивания сформированности компетенций и ее описание

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

Пороговый уровень – предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

Продвинутый уровень – предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

Высокий уровень – предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.

2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Код компетенции	Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
				Пороговый уровень	Продвинутый уровень	Высокий уровень
Общепрофессиональные компетенции						
ОПК-5	Экзамен	1	<p>Знать:</p> <ul style="list-style-type: none"> – принципы логической и технической организации вычислительных машин; – основы функционирования компонентов вычислительных машин, принципы аппаратного и программного управления компонентами; – принципы программирования на ассемблере. <p>Уметь:</p> <ul style="list-style-type: none"> – оценивать техническую конфигурацию, состав и основные характеристики вычислительных машин; – проектировать общую схему и описывать отдельные 	Знает	Знает и умеет	Знает, умеет и владеет навыками

			<p>блоки дискретного преобразователя ;</p> <p>– создавать проект на языке C/C++ , содержащий ассемблерный код, в среде Microsoft Visual Studio.</p> <p>Владеть навыками:</p> <p>– оценки, выбора и конфигурирования технических средств в составе компьютерных систем;</p> <p>– синтеза сетей из функциональных элементов;</p> <p>– программирования на ассемблере.</p>			
ОПК-5	Контрольная работа №1, Контрольная работа №2, Коллоквиум	2	<p>Уметь:</p> <p>– проектировать общую схему и описывать отдельные блоки дискретного преобразователя ;</p> <p>Владеть навыками:</p> <p>– синтеза сетей из функциональных элементов.</p>	Решает некоторые задачи	Решает большинство задач	Решает все задачи
	Индивидуальное задание	4	<p>Знать:</p> <p>– принципы программирования на ассемблере.</p> <p>Уметь:</p> <p>– создавать проект</p>	Знает	Знает и умеет	Знает, умеет и владеет навыками

			<p>на языке C/C++, содержащий ассемблерный код, в среде Microsoft Visual Studio.</p> <p>Владеть навыками:</p> <ul style="list-style-type: none">– программирован ия на ассемблере.			
--	--	--	---	--	--	--

3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объеме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;
- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;

- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Оценивание результатов обучения студентов по дисциплине «Архитектура вычислительных систем» осуществляется по регламенту текущего контроля и промежуточной аттестации.

Текущий контроль в семестре проводится с целью обеспечения своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы студентов.

Текущий контроль проводится в виде контрольных работ и коллоквиума. Критериями оценивания степени овладения умениями и навыками, полученными в результате освоения данной дисциплины, являются следующие:

Критерии оценки контрольной работы

«Отлично» (5 баллов) – ставится за работу, выполненную полностью без ошибок и недочетов. «Хорошо» (4 балла) – ставится за работу, выполненную полностью, но при наличии в ней не более одной ошибки и одного недочета, или не более трех недочетов. «Удовлетворительно» (3 балла) – ставится за работу, если обучающийся правильно выполнил не менее 2/3 всей работы. «Неудовлетворительно» (2 балла) – ставится за работу, если число ошибок и недочетов превысило норму для оценки «3» или правильно выполнено менее 2/3 всей работы.

Критерии оценки коллоквиума:

Оценка «5»

- глубокое и прочное усвоение программного материала;
- полные, последовательные, грамотные и логически излагаемые ответы при видеоизменении задания;
- свободно справляющиеся с поставленными задачами, знания материала;
- правильно обоснованные принятые решения;
- владение разносторонними навыками и приемами выполнения практических работ.

Оценка «4»

- на вопросы даны в целом верные ответы;
- грамотное изложение, без существенных неточностей в ответе на вопрос;
- правильное применение теоретических знаний;
- владение необходимыми навыками при выполнении практических задач.

Оценка «3»

- усвоение основных элементов материала;
- при ответе допускаются неточности и возможны недостаточно правильные формулировки;
- нарушение последовательности в изложении программного материала;
- студент в целом ориентируется в тематике учебного курса, но испытывает проблемы с раскрытием конкретных вопросов;
- затруднения в выполнении практических заданий.

Оценка «2»

- незнание даже основных элементов материала;
- при ответе возникают ошибки;
- затруднения при выполнении практических заданий.

3.2 Описание процедуры выставления оценки

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка. Для дисциплин, изучаемых в течение нескольких семестров, оценка может выставляться не только по окончании ее освоения, но и в промежуточных семестрах. Вид оценки («отлично», «хорошо», «удовлетворительно», «неудовлетворительно», «зачтено», «не зачтено») определяется рабочей программой дисциплины в соответствии с учебным планом.

Оценка «отлично» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована на высоком уровне.

Оценка «хорошо» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на продвинутом уровне.

Оценка «удовлетворительно» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «неудовлетворительно» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Шкала оценивания успеваемости текущего контроля и промежуточной аттестации

В зависимости от уровня сформированности компетенции по окончании освоения дисциплины студенту выставляется оценка по четырехбалльной шкале.

Шкала оценивания результатов теста

Каждый вопрос теста оценивается числом от 0 до 1. Максимальное суммарное значение равно 8. Пороговому уровню освоения темы соответствует значение 4, продвинутому – 6, высокому – 8.

Шкала оценивания результатов контрольных работ

Шкала оценивания решения задачи:

0 баллов – полное отсутствие решения; 0,5 балла – частичное выполнение критерия; 0,8 балла – полное выполнение критерия с незначительными ошибками, 1 балл – полное выполнение

критерия. Контрольная
работа №1:

Суммируем баллы по всей контрольной работе. Выставляем за контрольную полученное количество баллов + 2 с округлением по стандартным правилам.

Контрольная работа №2:

Оценка за контрольную выставляется по формуле (оценка_задачи_1 + 2*оценка_задачи_2 + 2) с округлением по стандартным правилам.

Шкала оценивания индивидуального задания

Индивидуальное задание оценивается по двухбалльной системе – «0» и «1». Оценка «1» выставляется в случае, если представленная на проверку программа отвечает всем общим требованиям и требованиям индивидуального задания. В противном случае выставляется оценка «0». Допускается многократная сдача этой работы до получения оценки 1.

Шкала оценивания результатов коллоквиума

Шкала оценивания решения задачи:

0 баллов – полное отсутствие решения; 0,5 балла – частичное выполнение критерия; 0,8 балла – полное выполнение критерия с незначительными ошибками, 1 балл – полное выполнение критерия.

Оценка за коллоквиум выставляется по формуле (оценка_задачи_1 + оценка_задачи_2 + 2*оценка_задачи_3 + 1) с округлением по стандартным правилам.

Шкала оценивания экзамена

«2» – неудовлетворительно:

Теоретический вопрос: студент не раскрыл теоретический вопрос, на заданные экзаменаторами вопросы не смог дать удовлетворительный ответ.

Практический вопрос: студент не понял смысла задачи, не смог выполнить задания. На заданные экзаменатором вопросы ответил неудовлетворительно, не продемонстрировал сформированность требующихся для выполнения заданий знаний и умений.

«3» – удовлетворительно:

Теоретический вопрос: студент смог с помощью дополнительных вопросов воспроизвести основные положения темы, но не сумел привести соответствующие примеры или аргументы, подтверждающие те или иные положения.

Практический вопрос: студент понял смысл задачи, но смог выполнить задание лишь после дополнительных вопросов, предложенных экзаменатором. При этом на поставленные экзаменатором вопросы не вполне ответил правильно и полно, но подтвердил ответами понимание вопросов и продемонстрировал отдельные требующиеся для выполнения заданий знания и умения.

«4» – хорошо:

Теоретический вопрос: студент (не допуская ошибок) правильно изложил теоретический вопрос, но недостаточно полно или допустил незначительные неточности, не искажающие суть понятий, теоретических положений, правовых и моральных норм. Примеры, приведенные учеником, воспроизводили материал учебников. На заданные экзаменатором уточняющие вопросы ответил правильно.

Практический вопрос: студент понял смысл задачи, предложенные задания выполнил правильно, но недостаточно полно. На заданные экзаменатором вопросы ответил правильно. Проявил необходимый уровень всех требующихся для выполнения заданий знаний и умений.

«5» – отлично:

Теоретический вопрос: студент полно и правильно изложил теоретический вопрос, привел собственные примеры, правильно раскрывающие те или иные положения, сделал обоснованный вывод;

Практический вопрос: студент полно и правильно выполнил предложенные задания, проявил высокий уровень всех требующихся для выполнения заданий знаний и умений.

Приложение №2 к рабочей программе дисциплины «Архитектура вычислительных систем»

Методические указания для студентов по освоению дисциплины

Основной формой изложения учебного материала по дисциплине «Архитектура вычислительных систем» являются лекции, причем в достаточно большом объеме. По ряду тем предусмотрены практические занятия, на которых происходит закрепление лекционного материала путем применения его к конкретным задачам.

Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, как в аудитории, так и самостоятельно в качестве домашних заданий. Примеры решения задач разбираются на лекциях и практических занятиях, при необходимости по наиболее трудным темам проводятся дополнительные консультации. Основная цель решения задач – помочь усвоить фундаментальные понятия и основы архитектуры вычислительных систем. Для решения всех задач необходимо знать и понимать лекционный материал. Поэтому в процессе изучения дисциплины рекомендуется регулярное повторение пройденного лекционного материала. Материал, законспектированный на лекциях, необходимо дома еще раз проработать и при необходимости дополнять информацией, полученной на консультациях, практических занятиях или из учебной литературы. Большое внимание должно быть уделено выполнению домашней работы. В качестве заданий для самостоятельной работы дома студентам предлагаются задачи, аналогичные разобранным на лекциях и практических занятиях или немного более сложные, которые являются результатом объединения нескольких базовых задач.

Для проверки и контроля усвоения теоретического материала, приобретенных практических навыков в течение обучения проводятся мероприятия текущей аттестации в виде теста, двух контрольных работ, индивидуального задания и коллоквиума. Также проводятся консультации (при необходимости) по разбору заданий для самостоятельной работы, которые вызвали затруднения. В конце семестра студенты сдают экзамен.

Необходимым условием допуска к экзамену является сдача индивидуальной работы на оценку «1».

Экзамен принимается по экзаменационным билетам, каждый из которых включает в себя теоретический вопрос и задачу по теме «Проектирование дискретных преобразователей». На самостоятельную подготовку к экзамену выделяется 3 дня, во время подготовки к экзамену предусмотрена групповая консультация. Освоить вопросы, излагаемые в процессе изучения дисциплины «Архитектура вычислительных систем» самостоятельно студенту крайне сложно. Это связано со сложностью изучаемого материала и большим объемом курса. Поэтому посещение всех аудиторных занятий является совершенно необходимым. Без упорных и регулярных занятий в течение семестра сдать экзамен по итогам изучения дисциплины студенту практически невозможно.

Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

В качестве учебно-методического обеспечения рекомендуется использовать литературу, указанную в разделе № 7 данной рабочей программы.

Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет (http://lib.uniyar.ac.ru/opac/bk_login.php) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ (http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/пароллю.

3. Электронная картотека «Книгообеспеченность» (http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека «Книгообеспеченность» доступна в сети университета и через Личный кабинет.

Примеры выполнения заданий контрольных работ

Примеры выполнения заданий контрольных работ рассматриваются в источниках из списка основной литературы (см. раздел №7 настоящей программы), а именно:

- контрольная работа №1 – источник [1];
- контрольная работа №2 – источники [2]–[3];
- индивидуальное задание – источник [4];
- коллоквиум – источники [1]–[3].

Наиболее сложные моменты в решении задач обсуждаются на консультациях по просьбе студентов.

Задания для самопроверки

Компетенция ОПК-5:

1. Какое утверждение не относится к защищённому режиму работы процессора? А. Процессор работает с виртуальными адресами.
В. Используется сегментация.
С. Используется страничная трансляция.
D. Используется только плоская модель памяти.
2. Как получить дополнительный код из прямого кода?
А. Побитово инвертировать число и прибавить единицу.
В. Побитово инвертировать число и вычесть единицу.
С. Прибавить единицу и выполнить циклический сдвиг на 1 разряд влево.
D. Инвертировать знаковый разряд.
3. Назовите булевы функции, образующие базис сети из функциональных элементов (для случая двоичных входов и выходов). Выберите все верные ответы.
А. Конъюнкция.
В. Дизъюнкция.
С. Импликация.
D. Инверсия.
4. Как оценивается сложность сети из функциональных элементов?
А. Это количество блоков, полученных с помощью декомпозиции.
В. Это количество входов и выходов.
С. Это количество функциональных элементов.

D. Это длина максимального пути от входа к выходу.

5. Выберите правильный порядок следования уровней архитектурной модели PCI-Express (от верхнего к нижнему).

A. Канальный уровень; уровень транзакций; физический уровень.

B. Физический уровень; уровень транзакций; канальный уровень.

C. Уровень транзакций; канальный уровень; физический уровень.

D. Уровень транзакций; физический уровень; канальный уровень.

Ключ:

1 – D; 2 – A; 3 – A, B, D; 4 – C; 5 – C.