

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 33

УТВЕРЖДАЮ
Руководитель образовательной программы

доц., к.э.н., доц.
(должность, уч. степень, звание)

Т.Н. Елина
(инициалы, фамилия)
(подпись)

«20» февраля 2026 г

Лист согласования рабочей программы дисциплины

Программу составил (а)

доц., к.э.н., доц.
(должность, уч. степень, звание)

 20.02.26
(подпись, дата)

Т.Н. Елина
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 33

«20» февраля 2026 г, протокол № 7

Заведующий кафедрой № 33

д.т.н., проф.
(уч. степень, звание)

 20.02.26
(подпись, дата)

С.В. Бездатева
(инициалы, фамилия)

Заместитель директора института №3 по методической работе

доц., к.т.н.
(должность, уч. степень, звание)

 20.02.26
(подпись, дата)

Н.В. Решетникова
(инициалы, фамилия)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Архитектура ЭВМ»
(Наименование дисциплины)

Код направления подготовки/ специальности	10.03.01
Наименование направления подготовки/ специальности	Информационная безопасность
Наименование направленности/ специализации	Безопасность компьютерных систем
Форма обучения	очная
Год приема	2026

Аннотация

Дисциплина «Архитектура ЭВМ» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 10.03.01 «Информационная безопасность» направленности/специализации «Безопасность компьютерных систем». Дисциплина реализуется кафедрой «№33».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ОПК-2 «Способен применять информационно- коммуникационные технологии, программные средства системного и прикладного назначения, в том числе отечественного производства, для решения задач профессиональной деятельности»

ОПК-7 «Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности»

Содержание дисциплины охватывает круг вопросов, связанных с логико-математическими основами, аппаратной реализацией и принципами функционирования вычислительных машин.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа обучающегося.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена (4 семестр).

Общая трудоемкость освоения дисциплины составляет 3 зачетных единицы, 108 часов.

Язык обучения по дисциплине «русский»

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

Изучение принципов построения и функционирования аппаратного обеспечения вычислительных машин. Она формирует у будущих специалистов фундаментальные знания о структуре процессоров, систем памяти и устройств ввода-вывода, позволяя понимать процесс обработки данных на самом низком аппаратном уровне.

1.2. Дисциплина входит в состав обязательной части образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Общепрофессиональные компетенции	ОПК-2 Способен применять информационно-коммуникационные технологии, программные средства системного и прикладного назначения, в том числе отечественного производства, для решения задач профессиональной деятельности	ОПК-2.3.1 знает классификацию современных компьютерных систем, типовые структуры и принципы организации компьютерных сетей; назначение, функции и обобщённую структуру операционных систем; назначение и основные компоненты систем баз данных
Общепрофессиональные компетенции	ОПК-7 Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности	ОПК-7.3.1 знает основные принципы построения компьютера, формы и способы представления данных в персональном компьютере ОПК-7.У.1 умеет работать с интегрированной средой разработки программного обеспечения ОПК-7.В.1 владеет навыками разработки, документирования, тестирования и отладки программ

2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

- «Информатика»,
- «Математика»,

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и используются при изучении других дисциплин:

- «Производственная практика»,

– «Базы данных»

3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам
		№4
1	2	3
Общая трудоемкость дисциплины, ЗЕ/ (час)	3/ 108	3/ 108
Из них часов практической подготовки		
Аудиторные занятия, всего час.	34	34
в том числе:		
лекции (Л), (час)	17	17
практические/семинарские занятия (ПЗ), (час)		
лабораторные работы (ЛР), (час)	17	17
курсовой проект (работа) (КП, КР), (час)		
экзамен, (час)	36	36
Самостоятельная работа, всего (час)	38	38
Вид промежуточной аттестации: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.)	Экз.,	Экз.,

4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП/КР (час)	СР (час)
Семестр 4					
Раздел 1. Основы теории информации и логические основы Тема 1.1. Системы счисления (двоичная, шестнадцатеричная) Тема 1.2. представление чисел в памяти Тема 1.3. основы булевой алгебры и логические элементы (И, ИЛИ, НЕ, триггеры).	2				
Раздел 2. Архитектура фон Неймана Тема 2.1. Базовые принципы построения ЭВМ: процессор, память, шины данных, адресации и управления. Тема 2.2. Цикл команд (выборка, декодирование, исполнение).	2				

Раздел 3. Организация процессора Тема 3.1. Арифметико-логическое устройство (АЛУ) Тема 3.2. Устройство управления Тема 3.3. Регистровая память Тема 3.4. Конвейеризация команд и суперскалярные архитектуры.	2				
Раздел 4. Система памяти Тема 4.1. Иерархия запоминающих устройств: кэш-память, оперативная память (ОЗУ), внешние накопители (HDD, SSD). Тема 4.2. Принципы виртуальной памяти и страничной организации.	3				
Раздел 5. Система ввода-вывода Тема 5.1. Интерфейсы и шины (PCIe, USB), контроллеры, порты Тема 5.2. Принципы прямого доступа к памяти (ПДП) и прерывания.	4				
Раздел 6. Архитектура современных систем Тема 6.1 Многоядерные процессоры, многопроцессорные системы, RISC и CISC архитектуры Тема 6.2. Основы графических процессоров (GPU) и сопроцессоров.	4				
Итого в семестре:	17		17		38
Итого	17	0	17	0	38

Практическая подготовка заключается в непосредственном выполнении обучающимися определенных трудовых функций, связанных с будущей профессиональной деятельностью.

4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
1	Раздел 1. Основы теории информации и логические основы Содержание раздела включает в себя математический и физический фундамент, на котором строится вся цифровая техника. 1. Системы счисления Этот блок изучает способы кодирования чисел и правила перевода данных между человеком и машиной. Понятие позиционных систем: Основание системы, алфавит цифр, разрядный вес. Двоичная система (Base-2): Базовый алфавит (0 и 1). Почему компьютеры используют именно её (простота технической реализации двух устойчивых состояний). Шестнадцатеричная система (Base-16): Алфавит от 0 до 9 и от А до F. Использование для компактной записи двоичных кодов (1 шестнадцатеричный символ = 4 бита). Перевод чисел: Алгоритмы перевода (деление/умножение на основание) между 2-й, 10-й и 16-й системами счисления. Двоичная арифметика: Правила сложения, вычитания, умножения и

	деления «столбиком» в двоичном коде. 2. Представление чисел в памяти компьютера Изучаются конкретные форматы хранения числовых данных на физических носителях и в регистрах. Целые числа: Беззнаковое представление (только положительные числа). Знаковое представление: Прямой, обратный и дополнительный коды. Использование дополнительного кода для замены операции вычитания сложением. Переполнение разрядной сетки: Границы числовых типов (например, диапазон байта от -128 до 127). Вещественные числа (с плавающей запятой): Математическая модель: $A = M \cdot 2^p$ (где M – мантисса, p – порядок). Стандарт IEEE 754: Выделение битов под знак, смещенный порядок (экспоненту) и нормализованную мантиссу (форматы Single и Double precision). Ошибки округления и потеря точности при работе с вещественными числами. 3. Основы булевой алгебры Математический аппарат для проектирования и анализа цифровых схем. Логические переменные и функции: Принятие значений «Истина» (1) или «Ложь» (0). Базовые логические операции: Конъюнкция (И, логическое умножение) Дизъюнкция (ИЛИ, логическое сложение) Инверсия (НЕ, отрицание) Дополнительные операции: Исключающее ИЛИ (XOR), штрих Шеффера (И-НЕ), стрелка Пирса (ИЛИ-НЕ). Таблицы истинности: Построение таблиц для вычисления результатов сложных логических выражений. Законы алгебры логики: Законы де Моргана, дистрибутивности, поглощения и склеивания для упрощения логических функций. 4. Логические элементы и триггеры Переход от математических формул к реальным электронным схемам на транзисторах. Вентили (Логические элементы): Физическая реализация операций И, ИЛИ, НЕ. Условные графические обозначения (УГО) на схемах. Комбинационные схемы: Схемы, результат которых зависит только от текущих входных сигналов (дешифраторы, мультиплексоры, сумматоры). Последовательностные схемы: Схемы с памятью, результат которых зависит от текущих входов и предыдущего состояния. Триггеры (Flip-Flops): Элементарные ячейки памяти, способные хранить 1 бит информации. RS-триггер: Базовый триггер на элементах ИЛИ-НЕ / И-НЕ (установка Set и сброс Reset). Синхронные триггеры: Управляемые тактовым сигналом (CLK) для избежания «гонок» сигналов. D-триггер и JK-триггер: Модификации для надежного хранения данных и организации счетчиков.
2	Раздел 2. Архитектура фон Неймана Содержание раздела «Архитектура фон Неймана» раскрывает классическую концепцию построения вычислительных систем, которая лежит в основе большинства современных компьютеров. 1. Базовые принципы фон Неймана Изучаются четыре фундаментальных постулата, определяющих логику работы машины:

	Принцип однородности памяти: Программы (команды) и данные хранятся в одной и той же памяти. Компьютер не отличает команду от числа, пока на неё не укажет регистр команд. Принцип адресности: Память состоит из пронумерованных ячеек. Процессор имеет доступ к любой ячейке в произвольный момент времени по её адресу. Принцип программного управления: Программа состоит из набора команд, которые выполняются процессором автоматически друг за другом в определенной последовательности. Принцип условного перехода: Возможность изменять естественный порядок выполнения команд в зависимости от результатов промежуточных вычислений. 2. Ключевые компоненты ЭВМ Разбор структуры компьютера и зон ответственности каждого блока: Процессор (ЦП / CPU): Главный управляющий и вычислительный узел. Включает устройство управления (УУ) и арифметико-логическое устройство (АЛУ). Память (ОЗУ / RAM): Единый массив ячеек для временного хранения кодов инструкций и операндов. Устройства ввода-вывода: Элементы для связи ЭВМ с внешним миром и пользователем. Бутиловое горлышко фон Неймана (Von Neumann Bottleneck): Ограничение пропускной способности, связанное с тем, что процессор и память разделяют общую шину, из-за чего данные нельзя считывать одновременно с инструкциями. 3. Системные шины (Магистраль) Изучение физических и логических каналов связи между компонентами: Шина данных: Двухнаправленная шина для передачи непосредственных значений (чисел, символов, кодов команд). Ее разрядность часто определяет разрядность процессора. Шина адреса: Однонаправленная шина. Процессор передает по ней адрес ячейки памяти или порта ввода-вывода, к которому хочет обратиться. Разрядность определяет максимальный объем адресуемой памяти. Шина управления: Передает сигналы синхронизации и команды управления (например, «чтение», «запись», «готовность устройства»). 4. Принципы адресации и управления Механизмы, с помощью которых процессор находит нужные данные в памяти: Регистры управления: Изучение роли специальных ячеек внутри процессора, таких как счетчик команд (PC/IP), регистр команд (IR), указатель стека (SP). Режимы адресации: Способы указания адреса операнда в коде команды: Прямая (абсолютная): В команде записан сам адрес памяти. Непосредственная: В команде записано само значение (константа). Косвенная (регистровая/относительная): Адрес вычисляется динамически на основе базовых и индексных регистров. 5. Цикл команд (Машинный цикл) Пошаговый циклический алгоритм работы процессора при выполнении программы: Фаза выборки (Fetch): Извлечение кода следующей команды из памяти по адресу, хранящемуся в счетчике команд, и перенос её в регистр команд. Автоматическое увеличение счетчика команд. Фаза декодирования (Decode): Расшифровка Устройством Управления (УУ) кода операции: определение того, какие действия нужно выполнить и где находятся операнды. Выборка операндов: Извлечение необходимых данных (из памяти или регистров) для вычислений.
--	--

	Фаза исполнения (Execute): Выполнение операции в АЛУ, пересылка данных или организация перехода. Запись результата (Write-back): Сохранение полученного значения обратно в регистры или оперативную память. Прекращение цикла при поступлении команды останова (HALT).
3	Раздел 3. Организация процессора Содержание раздела «Организация процессора» раскрывает внутреннее устройство центрального процессора (ЦП) как главного вычислительного и управляющего органа ЭВМ, а также технологии повышения его производительности. 1. Арифметико-логическое устройство (АЛУ) Изучение исполнительного блока, который выполняет непосредственные преобразования данных: Классификация операций: Арифметические: Сложение, вычитание, умножение, деление (для целых чисел и чисел с плавающей запятой). Логические: И, ИЛИ, НЕ, Исключающее ИЛИ (XOR). Сдвиговые: Логические, арифметические и циклические сдвиги битов влево/вправо. Структура АЛУ: Сумматоры, комбинационные схемы, схемы формирования признаков (флагов). Регистр флагов (состояния): Фиксация признаков результата операции (нулевой результат ZF, переполнение OF, перенос CF, знак/отрицательное число SF). 2. Устройство управления (УУ) Изучение блока, координирующего синхронную работу всех узлов процессора и ЭВМ: Функции УУ: Выборка команд, их декодирование, генерация последовательности управляющих сигналов для АЛУ, памяти и внешних устройств. Способы реализации УУ: С жесткой (схемной) логикой: Высокое быстродействие, но сложная перестройка схемы (характерно для RISC). С микропрограммным управлением: Каждая машинная команда разбивается на микрокоманды, хранящиеся в ПЗУ микропрограмм. Легче модифицируется (характерно для CISC). Синхронизация: Тактовый генератор, понятие такта частоты и машинного цикла. 3. Регистровая память (внутренняя память ЦП) Разбор сверхбыстрой памяти, расположенной прямо на кристалле процессора для минимизации задержек: Регистры общего назначения (РОН): Используются программистом и компилятором для временного хранения операндов и промежуточных результатов вычислений. Специальные (управляющие) регистры: Счетчик команд (PC / IP): Адрес следующей команды. Регистр команд (IR): Хранит код текущей выполняемой команды. Указатель стека (SP): Адрес вершины стека в памяти для обработки вызовов функций и прерываний. 4. Конвейеризация команд (Pipelining) Технология параллельной обработки, увеличивающая пропускную способность процессора: Принцип конвейера: Разбиение процесса выполнения команды на независимые стадии (Выборка $\rightarrow$ Декодирование $\rightarrow$ Чтение операндов $\rightarrow$ Исполнение $\rightarrow$ Запись результата). Одновременная обработка нескольких команд на разных стадиях.

	Конвейерные конфликты (Hazards): Проблемы, замедляющие работу конвейера: Структурные: Два блока конвейера одновременно требуют один и тот же аппаратный ресурс (например, память). По данным (зависимости): Текущая команда требует результат предыдущей команды, которая еще не успела завершить исполнение. По управлению: Возникают при условных переходах, когда неизвестно, какую команду выполнять следующей. Способы решения: предсказание переходов (Branch Prediction). 5. Суперскалярные архитектуры Развитие идеи конвейера для выполнения нескольких команд за один такт: Множественные исполнительные конвейеры: Наличие на кристалле нескольких параллельных АЛУ и других блоков. Внеочередное исполнение (Out-of-Order Execution): Процессор анализирует поток команд и выполняет их не в порядке записи, а по мере готовности данных и освобождения АЛУ. Переименование регистров (Register Renaming): Аппаратное устранение конфликтов по именам регистров с помощью расширенного пула скрытых физических регистров. Спекулятивное исполнение (Speculative Execution): Выполнение команд по предсказанному направлению ветвления еще до того, как условие перехода окончательно вычислено.
4	Раздел 4. Система памяти Содержание раздела «Система памяти» охватывает структуру, принципы организации и алгоритмы управления всеми видами запоминающих устройств, используемых в ЭВМ для обеспечения баланса между скоростью работы и стоимостью хранения данных. 1. Иерархия запоминающих устройств Изучение концепции многоуровневой памяти, решающей компромисс: чем память быстрее, тем она дороже и меньше по объему. Пирамида памяти: Взаимосвязь уровней (Регистры Кэш ОЗУ Энергонезависимая память). Принцип локальности данных: Фундамент эффективности иерархии: Временная локальность: Если данные использовались сейчас, велика вероятность, что они понадобятся снова в ближайшее время. Пространственная локальность: Если процессор обратился к ячейке памяти, велика вероятность, что скоро он обратится к соседним ячейкам. 2. Кэш-память (Cache) Изучение сверхбыстрой статической памяти (SRAM), сглаживающей разницу в скорости между ЦП и ОЗУ. Архитектура кэша: Уровни кэш-памяти ($L1, L2, L3$), их разделение на кэш инструкций и кэш данных. Способы отображения памяти на кэш: Прямое отображение (Direct Mapped): Каждому адресу ОЗУ строго соответствует одна строка кэша. Полностью ассоциативный кэш: Любой блок ОЗУ может быть записан в любую строку кэша. Множественно-ассоциативный кэш: Оптимальный компромисс, где адрес ОЗУ привязан к определенному набору (сету) строк. Алгоритмы замещения строк (при промахе кэша): LRU (наиболее давно не использовавшийся), FIFO (первым пришел–первым ушел), Random (случайный выбор). Политика записи данных: Сквозная запись (Write-Through) и обратная запись (Write-Back). 3. Оперативная память (ОЗУ / RAM) Изучение динамической памяти (DRAM) как основного рабочего пространства для активных процессов.

	Принцип работы DRAM: Хранение бита информации в виде заряда на микроконденсаторе, необходимость постоянной регенерации (освежения) данных. Физическая организация: Матричная структура (строки и столбцы), сигналы выбора строки (RAS) и столбца (CAS), тайминги (задержки) памяти. Эволюция стандартов: Развитие синхронной памяти DDR (DDR4, DDR5), концепция передачи данных по обоим фронтам тактового сигнала. Многоканальные режимы: Организация работы нескольких модулей памяти параллельно для увеличения общей пропускной способности шины. 4. Внешние накопители (Энергонезависимая память) Изучение устройств долгосрочного хранения программ и пользовательских файлов. Жесткие диски (HDD): Физическое устройство (магнитные пластины, головки чтения-записи), сектора, дорожки, цилиндры. Проблема механических задержек (время поиска дорожки, скорость вращения шпинделя). Твердотельные накопители (SSD): Устройство на базе флэш-памяти NAND. Архитектура ячеек (SLC, MLC, TLC, QLC), ограниченный ресурс перезаписи (TBW). Интерфейсы и протоколы: Различия между классическим интерфейсом SATA (протокол AHCI) и высокоскоростными шинами PCIe (протокол NVMe). 5. Виртуальная память и страничная организация Технология автоматического расширения адресного пространства за счет ресурсов внешнего накопителя и изоляции процессов друг от друга. Физические и логические (виртуальные) адреса: Разделение адресного пространства программы и реальных микросхем ОЗУ. Диспетчер памяти (MMU): Аппаратный блок процессора, отвечающий за динамическую трансляцию виртуальных адресов в физические. Страничная организация (Paging): Разбиение памяти на виртуальные страницы и физические фреймы (обычно по 4 Кб). Таблица страниц (Page Table): Структура данных, хранящая карту соответствия между страницами и фреймами. Буфер ассоциативной трансляции (TLB): Специализированный аппаратный кэш внутри процессора для ускорения трансляции адресов страниц. Страничный сбой (Page Fault): Ситуация, когда нужной страницы нет в ОЗУ, и операционная система вынуждена подгружать её из файла подкачки (Swap) на HDD/SSD.
5	Раздел 5. Система ввода-вывода Содержание раздела «Система ввода-вывода» описывает механизмы, протоколы и аппаратные средства, с помощью которых центральный процессор взаимодействует с периферийными устройствами и внешней средой. 1. Интерфейсы и системные шины Изучение физических каналов передачи данных и правил сопряжения устройств. Эволюция шин: Переход от классических параллельных шин (PCI) к высокоскоростным последовательным интерфейсам «точка-точка». Шина PCIe (PCI Express): Архитектура каналов (lanes): конфигурации x1, x4, x8, x16. Поколения PCIe (от Gen 1 до современных версий), их пропускная способность и кодирование сигнала. Концепция слоистой структуры протокола (физический, канальный, транзакционный уровни). Шина USB (Universal Serial Bus): Топология сети (древовидная структура с хабами) и концепция ведущий/ведомый (Host/Device). Различия версий (USB 2.0, USB 3.x, USB4) и физических стандартов (Type-A, Type-C). Режимы передачи данных (управляющие, изохронные,

	прерывания, массивы данных). 2. Контроллеры, порты и адресация ввода-вывода Аппаратные посредники между процессором и периферией. Контроллеры (адаптеры) устройств: Назначение электронных схем, преобразующих высокоуровневые команды процессора в низкоуровневые сигналы управления конкретным железом. Регистры контроллера (Порты ввода-вывода): Внутренние ячейки контроллера, доступные процессору: регистры данных, регистры состояния (Status) и регистры управления (Control). Методы адресации ввода-вывода: Изолированный ввод-вывод (Port-mapped I/O): Использование отдельного адресного пространства для портов и специальных команд процессора (например, IN и OUT в архитектуре x86). Ввод-вывод, отображаемый на память (Memory-mapped I/O / MMIO): Выделение части общего адресного пространства оперативной памяти под регистры внешних устройств. Процессор работает с ними с помощью обычных команд чтения/записи (MOV). 3. Программный ввод-вывод (Опрос / Polling) Базовый и самый простой способ обмена данными. Принцип работы: Процессор запускает операцию ввода-вывода и в цикле непрерывно опрашивает регистр состояния контроллера, ожидая окончания его работы (флага готовности). Недостатки метода: Колоссальные непроизводительные затраты процессорного времени («пустые циклы» ожидания), полная блокировка одного потока вычислений во время медленных операций ввода-вывода. 4. Система прерываний (Interrupt-driven I/O) Механизм асинхронного уведомления процессора о внешних событиях. Принцип работы: Процессор отправляет команду устройству и продолжает выполнять основную программу. Когда устройство готово, оно посылает аппаратный сигнал на специальную линию. Цикл обработки прерывания: Передача сигнала $\rightarrow$ Завершение текущей команды $\rightarrow$ Сохранение контекста (регистров, счетчика команд) в стек $\rightarrow$ Определение источника $\rightarrow$ Вызов обработчика прерывания (ISR) $\rightarrow$ Восстановление контекста. Контроллер прерываний (APIC / PIC): Аппаратный блок, управляющий приоритетами сигналов от разных устройств и распределяющий их по ядрам процессора. Векторные прерывания: Таблица векторов прерываний в памяти, хранящая адреса начала программ-обработчиков для каждого типа события. 5. Прямой доступ к памяти (ПДП / DMA) Технология освобождения процессора от рутинной пересылки больших объемов данных. Контроллер ПДП (DMA Controller): Специализированный сопроцессор, способный брать на себя управление системными шинами (шиной адреса и данных). Алгоритм работы: Процессор задает DMA-контроллеру начальный адрес в ОЗУ, адрес порта устройства и количество байт для передачи, после чего переключается на другие задачи. DMA-контроллер самостоятельно пересылает данные напрямую между устройством и памятью, минуя регистры ЦП. Режимы работы: Блочная передача (захват шины до конца операции), поцикловое воровство (передача данных в моменты, когда процессор не использует шину) и пакетный режим. Окончание работы: По завершении всей транзакции данных DMA-контроллер генерирует прерывание для ЦП, сообщая об успехе.
6	Раздел 6. Архитектура современных систем Содержание финального раздела «Архитектура современных систем» описывает устройство актуальных вычислительных платформ, отход от классических

	однойдерных схем и принципы организации параллельных вычислений на аппаратном уровне. 1. Архитектурные парадигмы: RISC против CISC Изучение двух фундаментальных подходов к построению систем команд (ISA): CISC (Complex Instruction Set Computer): Философия «сложных команд» (пример: архитектура x86). Инструкции имеют переменную длину, выполняются за разное число тактов и могут напрямую работать с оперативной памятью. Плюсы и минусы: Компактный код программ, но чрезвычайно сложное аппаратное декодирование команд внутри процессора. RISC (Reduced Instruction Set Computer): Философия «сокращенного набора команд» (примеры: ARM, RISC-V, Apple Silicon). Инструкции имеют фиксированную длину (обычно 32 бита), выполняются за один такт. Работа с памятью идет только через специальные команды Load и Store, все остальные вычисления—внутри регистров. Плюсы и минусы: Простая аппаратная логика, идеальная предсказуемость для конвейера, высокая энергоэффективность, но код программ становится объемнее. Современный гибридный подход: Современные процессоры x86 (Intel, AMD) внутри имеют RISC-ядро. На этапе декодирования сложные CISC-инструкции аппаратно разбиваются на простые внутренние микрооперации (мопы), похожие на RISC. 2. Многоядерные процессоры (Multicore) Изучение интеграции нескольких вычислительных ядер на одном полупроводниковом кристалле: Архитектура ядра: Каждое ядро—это фактически самостоятельный процессор со своим ALU, УУ, конвейером и кэшем первого уровня (L1). Разделение кэш-памяти: Индивидуальные кэши L1 и L2 для каждого ядра и общий (разделяемый) кэш L3 для коммуникации между ядрами. Проблема когерентности кэшей: Ситуация, когда два разных ядра работают с одними и теми же данными в ОЗУ, копируют их в свои локальные кэши и изменяют. Изучение аппаратных протоколов синхронизации кэшей (например, протокол MESI), предотвращающих рассинхронизацию данных. Асимметричные многоядерные архитектуры (ARM big.LITTLE / Intel Hybrid): Сочетание мощных производительных ядер (P-cores) для тяжелых задач и энергоэффективных ядер (E-cores) для фоновых процессов ради экономии заряда. 3. Многопроцессорные системы Масштабирование вычислительной мощности путем объединения отдельных физических процессоров: Классификация Флинна: Базовая таксономия архитектур по потокам команд и данных (SISD, SIMD, MISD, MIMD). Архитектура SMP (Symmetric Multiprocessing): Симметричные многопроцессорные системы, где все процессоры имеют одинаковый и равноправный доступ к общей оперативной памяти по единой системной шине. Архитектура NUMA (Non-Uniform Memory Access): Архитектура с неоднородным доступом к памяти. Память физически распределена между процессорами. Каждый процессор быстро работает со своей «локальной» памятью и медленнее—с «чужой» памятью других процессоров через специализированные межпроцессорные интерфейсы (Intel UPI, AMD Infinity Fabric). 4. Основы графических процессоров (GPU) и сопроцессоров Переход от универсальных вычислений к специализированным высокопараллельным структурам: Архитектура GPU против CPU: CPU оптимизирован под последовательное выполнение задач,
--	--

	имеет сложные блоки предсказания ветвлений и огромный кэш для минимизации задержек одиночного потока. GPU оптимизирован под массивный параллелизм (архитектура SIMD или SIMT). Он состоит из тысяч очень простых вычислительных ядер, способных одновременно выполнять одну и ту же математическую операцию над огромными массивами данных (текстуры, матрицы, векторы). Сферы применения GPU общего назначения (GPGPU): Использование видеокарт не только для графики, но и для научных расчетов, криптографии, трехмерного моделирования и обучения нейросетей (технологии CUDA, OpenCL). Специализированные сопроцессоры: Изучение роли тензорных ядер (TPU/Tensor Cores) для матричного умножения в искусственном интеллекте и нейропроцессоров (NPU) в мобильных SoC
--	---

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Учебным планом не предусмотрено					
Всего					

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 4				
1	Проектирование и исследование последовательностных схем: триггеры и счетчики	2		1
2	Анализ выполнения машинного цикла: исследование фаз выборки, декодирования и исполнения команд	2		2
3	Исследование работы регистровой памяти и устройства управления процессора	2		3
4	Моделирование конвейерной обработки команд и анализ конвейерных конфликтов	2		3
5	Принципы функционирования виртуальной памяти: страничная организация и трансляция адресов	2		4
6	Исследование механизмов обработки аппаратных прерываний в ЭВМ	2		5
7	Исследование проблемы когерентности кэш-памяти в многоядерных и	2		6

	многопроцессорных системах			
8	Разработка простейших параллельных алгоритмов на GPU с использованием технологии CUDA (или OpenCL)	3		6
Всего		17		

4.5. Выполнение курсового проекта/ курсовой работы

Учебным планом не предусмотрено

4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 4, час
1	2	3
Изучение теоретического материала дисциплины (ТО)	14	14
Курсовое проектирование (КП, КР)		
Расчетно-графические задания (РГЗ)		
Выполнение реферата (Р)		
Подготовка к текущему контролю успеваемости (ТКУ)	12	12
Домашнее задание (ДЗ)		
Контрольные работы заочников (КРЗ)		
Подготовка к промежуточной аттестации (ПА)	12	12
Всего:	38	38

5. Перечень учебно-методического обеспечения

для самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. разделов 6-11.

6. Перечень печатных и электронных учебных изданий

Перечень печатных и электронных учебных изданий приведен в таблице 8.

Таблица 8– Перечень печатных и электронных учебных изданий

Шифр/ URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке (кроме электронных экземпляров)
004 М 27	Марковская, Наталья Владимировна (канд. техн. наук, доц.). Основные компоненты цифровых вычислительных машин : учебное пособие / Н. В. Марковская ; С.-Петербург. гос. ун-т аэрокосм. приборостроения. - Санкт-Петербург : Изд-во ГУАП, 2021. - 50 с. : рис., табл. - Библиогр.: с. 49 (4 назв.). - ISBN 978-5-8088-1620-6 : Б. ц. - Текст : непосредственный.	5

004 Н 63	Николаев, Дмитрий Андреевич (канд. техн. наук). Основы архитектуры и организации ЭВМ : учебное пособие / Д. А. Николаев ; С.-Петербург. гос. ун-т аэрокосм. приборостроения. - Санкт-Петербург : Изд-во ГУАП, 2024. - 44 с. - Библиогр.: с. 42 (18 назв.). - ISBN 978-5-8088-1961-0 : 98.27 р. - Текст : непосредственный.	5
004 М 27	Марковская, Наталья Владимировна (канд. техн. наук, доц.). Архитектура ЭВМ : учебно-методическое пособие / Н. В. Марковская, Т. В. Семенов ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - Санкт-Петербург : Изд-во ГУАП, 2025. - 64 с. : рис. - Библиогр.: с. 62 (6 назв.). - 35.56 р. - Текст : непосредственный.	5
519.6/.8 С 79	Степанов, Владимир Евгеньевич. Основы теории информации : учебно-методическое пособие / В. Е. Степанов, Ю. В. Трифонова ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - Санкт-Петербург : Изд-во ГУАП, 2023. - 76 с. : рис., табл. - Библиогр.: с. 73 -74 (14 назв.). - Б. ц. - Текст : непосредственный.	5
004 Б 39	Беззатеев, Сергей Валентинович (д-р техн. наук, доц.). Теория информации для криптографических методов защиты информации : учебное пособие / С. В. Беззатеев ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - Санкт-Петербург : Изд-во ГУАП, 2024. - 32 с. - Библиогр.: с. 30 (6 назв.). - ISBN 978-5-8088-1989-4 : 1.00 р. - Текст : непосредственный.	5
621.391 М 59	Миклуш, Виктория Александровна. Основы теории информации : учебно-методическое пособие / В. А. Миклуш, В. А. Ушаков ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - Санкт-Петербург : Изд-во ГУАП, 2024. - 44 с. - Библиогр.: с. 42 (5 назв.). - 1.00 р. - Текст : непосредственный.	5

7. Перечень электронных образовательных ресурсов
информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
	Не предусмотрено

8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10– Перечень программного обеспечения

№ п/п	Наименование
	Не предусмотрено

8.2. Перечень информационно-справочных систем,используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Мультимедийная лекционная аудитория	
2	Компьютерный класс	

10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Экзамен	Список вопросов к экзамену; Экзаменационные билеты*; Задачи; Тесты.

Примечание: *экзаменационные билеты формируются на основе вопросов и задач таблицы 15.

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила

использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 –Критерии оценки уровня сформированности компетенций

Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
«отлично» «зачтено»	Обучающийся: – глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно связывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи; – делает выводы и обобщения; – свободно владеет системой специализированных понятий. – правильно выполнил от 90% до 100% тестовых заданий**.
«хорошо» «зачтено»	Обучающийся: – твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; – не допускает существенных неточностей; – увязывает усвоенные знания с практической деятельностью направления; – аргументирует научные положения; – делает выводы и обобщения; – владеет системой специализированных понятий. – правильно выполнил от 70% до 89% тестовых заданий**.
«удовлетворительно» «зачтено»	– обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; – допускает несущественные ошибки и неточности; – испытывает затруднения в практическом применении знаний направления; – слабо аргументирует научные положения; – затрудняется в формулировании выводов и обобщений; – частично владеет системой специализированных понятий. – правильно выполнил от 51% до 69% тестовых заданий**.
«неудовлетворительно» «не зачтено»	– обучающийся не усвоил значительной части программного материала; – допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; – испытывает трудности в практическом применении знаний; – не может аргументировать научные положения; – не формулирует выводов и обобщений. – правильно выполнил менее 51% тестовых заданий**.

10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
1	Понятие позиционной системы счисления. Алгоритмы перевода чисел между 2-й, 10-й и 16-й системами счисления. Особенности выполнения арифметических операций в двоичной системе счисления. Форматы представления целых чисел в памяти ЭВМ.	ОПК-2.3.1

	Прямой, обратный и дополнительный коды, их назначение. Представление вещественных чисел с плавающей запятой в памяти ЭВМ. Стандарт IEEE 754. Основные логические операции булевой алгебры (И, ИЛИ, НЕ, Исключающее ИЛИ). Таблицы истинности. Законы алгебры логики и методы минимизации логических функций. Базовые логические элементы (вентили) и их роль в построении комбинационных схем. Последовательностные схемы: классификация, устройство и принцип работы RS-, D- и JK-триггеров. Классические принципы фон-неймановской архитектуры (однородность памяти, адресность, программное управление). Структурная схема ЭВМ фон Неймана. Назначение основных функциональных блоков. «Бутылочное горлышко» архитектуры фон Неймана, его суть и способы преодоления в современных системах. Системная магистраль (автобусная архитектура): назначение и разрядность шин адреса, данных и управления. Основные режимы адресации операндов в ЭВМ (прямая, непосредственная, косвенная, относительная). Этапы классического машинного цикла выполнения команды процессором (Fetch, Decode, Execute, Write-back).	
2	Арифметико-логическое устройство (АЛУ): классификация выполняемых операций, внутренняя структура. Регистр флагов (состояния) процессора и его роль в организации условных переходов. Устройство управления (УУ): функции, сравнение жесткой (схемной) и микропрограммной логики. Регистровая память процессора. Назначение регистров PC, IR, SP и регистров общего назначения (РОН). Конвейерная обработка команд: принцип работы, стадии конвейера и оценка эффективности. Конфликты в процессорах с конвейерной архитектурой (структурные, по данным, по управлению) и методы их разрешения. Суперскалярные процессоры: внеочередное (Out-of-Order) и спекулятивное выполнение команд, переименование регистров. Иерархическая структура системы памяти ЭВМ. Принцип	ОПК-7.3.1

	локальности данных (временной и пространственной). Кэш-память: назначение, уровни (L1, L2, L3) и принципы взаимодействия с оперативной памятью. Способы отображения оперативной памяти на кэш-память (прямое, ассоциативное, множественно-ассоциативное). Алгоритмы замещения строк в кэш-памяти (LRU, FIFO) и политики записи данных (Write-Through, Write-Back). Оперативная память (DRAM): физический принцип работы, матричная структура, назначение сигналов RAS и CAS. Внешняя энергонезависимая память. Сравнительный анализ архитектуры и принципов работы HDD и SSD. Виртуальная память: понятие, цели использования. Роль диспетчера памяти (MMU) и буфера TLB. Страничная организация виртуальной памяти. Структура таблицы страниц и обработка события Page Fault.	
3	Назначение системы ввода-вывода. Понятие интерфейса, контроллера и порта периферийного устройства. Сравнение способов адресации устройств ввода-вывода: изолированный ввод-вывод (MMIO) против отображаемого на память. Программный ввод-вывод методом опроса готовности (Polling): алгоритм, достоинства и недостатки. Организация ввода-вывода на основе прерываний. Понятие вектора прерывания и процедура обработки (ISR). Аппаратная реализация прерываний. Назначение и функции контроллера прерываний (APIC). Прямой доступ к памяти (ПДП/DMA): назначение, принцип работы и алгоритм взаимодействия ЦП и DMA-контроллера.	ОПК-7.У.1
4	Сравнительный анализ архитектурных концепций RISC и CISC: принципы построения систем команд, достоинства и недостатки. Особенности реализации современных процессоров x86: гибридная внутренняя архитектура. Многоядерные процессоры: организация межъядерного взаимодействия и иерархии кэш-памяти. Проблема когерентности кэшей в многоядерных системах. Суть аппаратного протокола MESI. Классификация вычислительных систем по Флинну (SISD, SIMD, MISD, MIMD). Сравнительный анализ многопроцессорных архитектур SMP и NUMA. Особенности доступа к памяти. Архитектура графических процессоров (GPU) общего	ОПК-7.В.1

	назначения: принципы SIMD/SIMT вычислений, отличия от CPU. Специализированные сопроцессоры (TPU, NPU) и их роль в современных вычислительных системах.	
--	--	--

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.

Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета	Код индикатора
	Учебным планом не предусмотрено	

Перечень тем для выполнения курсового проекта/ курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для выполнения курсового проекта / курсовой работы

№ п/п	Примерный перечень тем для выполнения курсового проекта/ курсовой работы
	Учебным планом не предусмотрено

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов	Код индикатора
1	Какое устройство выполняет арифметические и логические операции в процессоре? Арифметико-логическое устройство (АЛУ) Кэш-память Контроллер прерываний + Шина данных Как называется временная память, используемая для ускорения доступа к часто запрашиваемым данным? Оперативная память (ОЗУ) + Кэш-память Постоянная память (ПЗУ) Флеш-память Какой компонент материнской платы отвечает за взаимодействие между процессором и оперативной памятью? Южный мост + Северный мост BIOS Контроллер жесткого диска Какой тип памяти сохраняет данные только при наличии питания? ПЗУ (ROM) + ОЗУ (RAM) Жесткий диск Флеш-память	ОПК-2.3.1

	Как называется разъем для установки процессора на материнской плате? PCI-слот + Сокет DIMM-разъем SATA-порт Какой компонент управляет передачей данных между процессором и периферийными устройствами? Северный мост + Южный мост АЛУ Кэш-память Какой интерфейс используется для подключения видеокарты к материнской плате? SATA USB + PCI Express (PCIe) Ethernet Какой компонент отвечает за управление электропитанием компьютера? Северный мост + Блок питания (и контроллер питания на материнской плате) BIOS Шина данных Как называется технология, увеличивающая быстродействие за счет предварительной загрузки данных? Дефрагментация + Кэширование Виртуализация Архивирование	
2	Какой интерфейс используется для подключения жестких дисков и SSD? PCIe USB + SATA/NVMe Ethernet Как называется схема, управляющая порядком выполнения команд в процессоре? АЛУ + Устройство управления (УУ) Кэш-контроллер Шина адреса Какой параметр определяет количество операций, выполняемых процессором за секунду? Объем кэш-памяти + Тактовая частота	ОПК-7.3.1

	Разрядность Количество ядер	
3	Как называется устройство, преобразующее цифровой сигнал в аналоговый для вывода звука? Сетевой адаптер + ЦАП (Цифро-аналоговый преобразователь) Графический процессор Контроллер USB Какой компонент отвечает за хранение BIOS? ОЗУ + ПЗУ (или флеш-память) Жесткий диск Кэш-память Как называется набор микросхем, обеспечивающий работу материнской платы? Биос + Чипсет Контроллер памяти Шина данных Какой тип памяти используется для долговременного хранения данных? ОЗУ (RAM) Кэш-память + Жесткий диск (HDD/SSD) Регистры процессора Как называется режим работы процессора, при котором он выполняет несколько инструкций за один такт? Кэширование + Конвейеризация Виртуализация Многопоточность	ОПК-7.У.1
4	Как называется память, используемая для хранения загрузочных программ? ОЗУ + ПЗУ (BIOS/UEFI) Кэш-память Жесткий диск Как называется технология, позволяющая процессору выполнять несколько потоков команд одновременно? Виртуализация + Многопоточность (Hyper-Threading) Кэширование Тактовая частота Какой тип шины используется для передачи данных между процессором и оперативной памятью? USB	ОПК-7.В.1

	SATA + Системная шина (FSB или DMI) PCIe	
--	--	--

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
	Не предусмотрено

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

11. Методические указания для обучающихся по освоению дисциплины

11.1. Методические указания для обучающихся по освоению лекционного материала.

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

Структура предоставления лекционного материала:

Структурирование лекционного материала по «Архитектуре ЭВМ» строится по принципу «от простого к сложному» и «снизу вверх» (Bottom-Up): от физических вентилях и математических основ к логике процессора, архитектуре систем и параллельным вычислениям.

Ниже представлена эффективная структура подачи материала для каждой лекции, обеспечивающая максимальную усвояемость:

Единый модульный шаблон лекции

Каждое лекционное занятие (стандартные 2 академических часа / 90 минут) делится на 4 обязательных блока:

- Вводная часть (5–10 минут):
 - Актуализация: связь с предыдущей лекцией.
 - Проблема: постановка инженерной или архитектурной задачи (например: «Процессор работает в 100 раз быстрее ОЗУ. Как решить эту проблему?» перед лекцией о кэш-памяти).
- Теоретический базис (40–45 минут):
 - Разбор математических моделей, алгоритмов или физических принципов.
 - Построение схем, архитектурных диаграмм и временных графов.
- Практическая демонстрация (25–30 минут):
 - Разбор конкретного примера (трассировка команд, расчет объема памяти, минимизация схемы).
 - Привязка к коммерческим технологиям (Intel, AMD, ARM, Nvidia).
- Заключение и фидбек (5–10 минут):
 - Резюме лекции в трех ключевых тезисах.
 - Выдача экспресс-вопроса для самопроверки.

11.2. Методические указания для обучающихся по выполнению лабораторных работ

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;
- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;
- получение новой информации по изучаемой дисциплине;
- приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

Задание и требования к проведению лабораторных работ

Для эффективного закрепления теоретического материала каждая лабораторная работа должна иметь четкую структуру задания, измеримый результат и строгий регламент выполнения.

Ниже приведены общие требования к проведению, структура отчета и конкретные примеры заданий для каждого из 6 разделов дисциплины.

I. Общие требования к проведению и оцениванию

1. **Самостоятельность выполнения:** Каждый студент получает индивидуальный вариант задания (отличающийся логической функцией, исходными числами, архитектурными параметрами или ассемблерным кодом).
2. **Программная среда (Инструментарий):**
 - Раздел 1–3: Симулятор логических схем **Logisim-Evolution** или **Proteus**.
 - Раздел 2–3, 5–6: Эмуляторы и компиляторы **emu8086**, **SASM (NASM)** или специализированные учебные симуляторы (например, Учебный компьютер «Малютка», *MARS/Venus MIPS Simulators*).
 - Раздел 4, 6: Симуляторы кэша и памяти (например, *Dinero IV*, *Cache Simulator*) или языки C/C++ / Python для алгоритмического моделирования.

3. **Регламент защиты:** Защита состоит из демонстрации работающей схемы/программы, ответов на контрольные вопросы и предоставления текстового отчета.

Структура и форма отчета о лабораторной работе

Отчет оформляется в электронном виде (PDF) и должен содержать:

1. **Титульный лист** (вуз, кафедра, дисциплина, тема, ФИО студента и преподавателя, вариант).
2. **Цель работы** и краткое теоретическое введение (1–2 страницы).
3. **Постановка задачи** (индивидуальное задание и исходные данные).
4. **Основная часть:**
 - Математические расчеты (если применимо).
 - Графические схемы (скриншоты из Logisim) или листинг программы с подробными комментариями.
5. **Результаты тестирования:** Таблицы истинности, скриншоты пошагового выполнения (трассировки), логи работы программы.
6. **Вывод:** Анализ полученных результатов, оценка эффективности решения.

Требования к оформлению отчета о лабораторной работе

Требования к оформлению отчета о лабораторной работе по дисциплине «Архитектура ЭВМ» разрабатываются на основе действующих стандартов (ГОСТ 7.32 и ГОСТ 2.105) с учетом специфики ИТ-направления. Отчет является официальным документом, подтверждающим выполнение студентом учебного плана.

Ниже приведены стандарты форматирования, детальная структура разделов и правила оформления графического материала.

Общие стандарты форматирования текста

- Редактор: Microsoft Word, LibreOffice Writer или LaTeX.
- Формат файла для сдачи: Строго PDF (для сохранения разметки и формул).
- Параметры страницы: Формат А4, книжная ориентация.
- Поля: Левое—30 мм (под подшивку), правое—10 (или 15) мм, верхнее и нижнее—20 мм.
- Шрифт основного текста: Times New Roman (допускается Arial для технических схем).
- Размер шрифта (кегель): 14 пт (для таблиц и листингов кода допускается 10–12 пт).
- Межстрочный интервал: 1.5 (для кода и таблиц—1.0).
- Выравнивание: По ширине, с абзачным отступом (красная строка)—1.25 см.
- Нумерация страниц: Сквозная, арабскими цифрами внизу страницы (по центру или справа). На титульном листе номер не ставится, но включается в общую нумерацию.

II. Подробная структура отчета

Каждая – лабораторная работа должна содержать следующие структурные элементы (каждый крупный раздел начинается с новой страницы):

1. **Титульный лист:**
 - Наименование министерства и вуза, название кафедры.
 - Наименование дисциплины.
 - Тема и номер лабораторной работы.
 - Данные студента (ФИО, группа, номер варианта).
 - Данные проверяющего преподавателя (ФИО, ученая степень/должность, место для оценки и подписи).
 - Город и текущий год.
2. **Цель работы:** Краткое и емкое описание того, какие навыки и знания приобретаются (2–3 строки).
3. **Постановка задачи:** Формулировка индивидуального задания в строгом соответствии с выданным вариантом.
4. **Теоретическая часть (кратко):** Базовые понятия, формулы или алгоритмы, необходимые для выполнения работы (не более 1–2 страниц, без копирования учебника).
5. **Ход выполнения работы (Основная часть):**
 - *Для схемотехнических работ:* Этапы проектирования схемы, таблицы истинности, карты Карно, минимизация функций.
 - *Для программных работ:* Описание алгоритма, блок-схема (при необходимости), обоснование выбора структур данных или регистров.

6. **Результаты работы:**
- Скриншоты разработанных схем, интерфейсов программ или окон эмулятора.
 - Таблицы с результатами тестирования (трассировки, замеры времени, кэш-промахи).
7. **Вывод:** Итоговый аналитический текст. Должен соотноситься с целью работы (например: «В ходе работы была спроектирована схема АЛУ. Тестирование показало полное соответствие теоретической таблицы истинности и физических сигналов на флагах ZF и CF»).

III. Специфические правила для «Архитектуры ЭВМ»

Поскольку дисциплина инженерная и программная, к оформлению графики и кода предъявляются особые требования:

1. Оформление листингов программ (код на Ассемблере, C++, Python)

- Код вставляется в текст отчета моноширинным шрифтом (**Courier New** или **Consolas**), размер 10–12 пт, интервал 1.0.
- **Запрещено** вставлять код скриншотами. Только текст.
- Если код занимает более 2 страниц, он выносится в конец отчета в «Приложение А».
- Код должен быть снабжен комментариями: в Ассемблере комментируется назначение ключевых строк и использование регистров.

Пример оформления кода:

assembly

MOV AX, [BX] ; Загрузка в регистр AX данных из памяти по адресу в BX

ADD AX, CX ; Сложение содержимого AX с регистром CX, результат в AX

Используйте код с осторожностью.

2. Оформление рисунков и скриншотов схем (Logisim, Proteus)

- Все рисунки центрируются. Скриншоты должны быть четкими, с читаемыми подписями элементов, векторов и сигналов.
- Каждый рисунок должен иметь сквозной номер и подпись снизу.
- *Пример подписи:* Рисунок 1 – Схема 4-разрядного синхронного D-триггера в среде Logisim.
- В тексте отчета обязательно должна быть ссылка на рисунок (например: «...как показано на рисунке 1»).

3. Оформление таблиц (Трассировки, таблицы истинности)

- Все таблицы нумеруются и должны иметь заголовок сверху.
- *Пример заголовка:* Таблица 2 – Пошаговая трассировка состояния регистров процессора.
- Шапка таблицы и строки данных должны быть выровнены. Не допускается разрыв таблицы на следующую страницу, если там остается только одна строка (в случае переноса обязательно дублируется шапка таблицы с пометкой «Продолжение таблицы 2»).

4. Оформление математических формул

- Формулы (например, расчет мантиссы по IEEE 754 или среднего времени доступа к памяти) должны быть набраны в стандартном редакторе формул (MS Equation / MathType) или в LaTeX.
- Формулы выравниваются по центру, а их номера проставляются в крайнем правом положении в круглых скобках
- Ниже формулы обязательно дается расшифровка всех символов (где $\backslash(H)$ – вероятность попадания в кэш...).

11.3. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

В процессе выполнения самостоятельной работы у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет ему развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине;

– методические указания по выполнению контрольных работ (для обучающихся по заочной форме обучения).

11.4. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

Методические указания по прохождению текущего контроля успеваемости регламентируют процесс оценки знаний студентов в течение семестра. Текущий контроль по дисциплине «Архитектура ЭВМ» включает защиту лабораторных работ, прохождение компьютерного тестирования и выполнение контрольных опросов (коллоквиумов) по теоретическим модулям.

Ниже представлена структура методических указаний, критерии оценки и рекомендации по подготовке для студентов.

I. Структура и формы текущего контроля

В течение семестра успеваемость студента оценивается по трем основным направлениям:

1. **Выполнение и защита лабораторных работ (практический компонент):**
 - Проверка умения применять теорию на практике (схемотехника, низкоуровневое программирование, симуляция).
 - Каждая работа должна быть выполнена и защищена в установленные календарным планом сроки.
2. **Тематическое тестирование (теоретический компонент):**
 - Проводится на компьютерной базе (в СДО вроде Moodle) после завершения каждого крупного модуля (всего 3–4 теста за семестр).
 - Содержит вопросы с одиночным/множественным выбором, задания на установление соответствия и ввод короткого числового ответа.
3. **Контрольные опросы / Экспресс-коллоквиумы:**
 - Письменные летучки (10–15 минут) в начале лекций или семинаров для проверки усвоения материала предыдущего занятия.

II. Балльно-рейтинговая система (БРС) оценки

Если на курсе применяется БРС, итоговая оценка за текущий контроль формирует 60% от общей оценки за дисциплину (остальные 40% приходятся на промежуточную аттестацию–экзамен).

Примерное распределение баллов (из 100 максимальных за семестр):

- Выполнение и защита лабораторных работ (всего 16–18 работ)–**40 баллов**.
- Прохождение модульных тестов–**15 баллов**.
- Активность на занятиях и экспресс-опросы–**5 баллов**.
- *Итого за текущий контроль:* максимум **60 баллов**. Студент допускается к экзамену, если набрал не менее **35 баллов**.

III. Критерии оценивания элементов контроля

1. Защита лабораторных работ

- **Оценка «Отлично» (высокий уровень):** Работа выполнена полностью самостоятельно, программа/схема функционирует без ошибок при любых входных данных преподавателя. Отчет оформлен строго по ГОСТу. Студент демонстрирует глубокие знания, легко отвечает на дополнительные теоретические и архитектурные вопросы, свободно ориентируется в листинге кода или логических связях схемы.
- **Оценка «Хорошо» (средний уровень):** Работа выполнена в полном объеме, схема/программа работает. В отчете есть незначительные опечатки или мелкие дефекты оформления. При защите студент отвечает на большинство вопросов, но допускает неточности в деталях (например, путает назначение неосновных флагов АЛУ или конкретные тайминги памяти).
- **Оценка «Удовлетворительно» (минимальный уровень):** Работа выполнена, но функционирует с ограничениями (например, программа падает при некорректном вводе, схема имеет логические гонки). Отчет содержит явные нарушения стандартов оформления. Студент с трудом объясняет логику своего кода или структуры схемы, на вопросы отвечает только с наводящими подсказками преподавателя.

- **Оценка «Неудовлетворительно»:** Работа не выполнена, либо студент не может объяснить алгоритм и код представленной программы (подозрение на плагиат). Защита не засчитывается.
- 2. **Модульное тестирование**
 - «Отлично»—от 90% до 100% правильных ответов.
 - «Хорошо»—от 75% до 89% правильных ответов.
 - «Удовлетворительно»—от 60% до 74% правильных ответов.
 - «Неудовлетворительно»—менее 60% правильных ответов (требуется пересдача теста).

IV. Рекомендации по подготовке к текущему контролю

1. Как успешно выполнять и защищать лабораторные работы:

- **Изучайте теорию до практики.** Не начинайте собирать схему в Logisim или писать код на Ассемблере наугад. Сначала разберите синтаксис команд или логические уравнения.
- **Комментируйте свои решения.** При написании низкоуровневого кода оставляйте комментарии к каждой строке. На защите преподаватель может спросить: «Что делает инструкция по адресу 0x0040?» или «Зачем здесь этот триггер?». Комментарии помогут вам быстро вспомнить логику.
- **Проверяйте граничные условия.** Убедитесь, что ваше решение работает с крайними значениями (например, при вводе нуля, максимального знакового числа, или в ситуациях, вызывающих переполнение разрядной сетки OF=1).

2. Как готовиться к теоретическим тестам и опросам:

- **Ведите подробный конспект лекций.** Особое внимание уделяйте аббревиатурам (MMU, TLB, DMA, APIC, MESI) и их расшифровкам—это частая основа для тестовых вопросов.
- **Разбирайте архитектурные схемы.** Тесты часто содержат графические задания: определить тип триггера по схеме, указать пропущенный блок в цикле фон Неймана или определить направление стрелок на шинах.
- **Тренируйте расчетные задачи.** Автоматизируйте навыки перевода чисел, составления карт Карно и вычисления эффективного времени доступа к памяти. Ошибки по невнимательности в арифметике—главная причина потери баллов.

3. Соблюдение академических сроков (Дедлайнов):

- За несвоевременную сдачу лабораторной работы или пропуск теста без уважительной причины (подтвержденной документально) взимается штраф (например, снижение максимальной оценки за работу на 20–50%).
- Студенты, имеющие задолженности по лабораторным работам на момент окончания семестра, не допускаются к сдаче экзамена до ликвидации академических долгов в установленные деканатом дни пересдач.

11.5. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

– экзамен – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен, как правило, проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Промежуточная аттестация по дисциплине «Архитектура ЭВМ» является завершающим этапом обучения в семестре. Она призвана проверить уровень сформированности компетенций, знание теоретических основ построения вычислительных систем и навыки анализа работы аппаратных компонентов.

I. Порядок допуска к экзамену и процедура проведения

1. Условия допуска:

- К экзамену допускаются студенты, полностью выполнившие учебный план текущего контроля.
- Студент должен защитить **все лабораторные работы**, предусмотренные программой, и сдать отчеты по ним в формате PDF.

- При использовании балльно-рейтинговой системы (БРС) необходимо набрать минимальный пороговый балл текущей успеваемости (как правило, не менее 35 баллов из 60).
- 2. **Формат проведения:** Экзамен проводится в устно-письменной или компьютерной форме по утвержденным экзаменационным билетам. На подготовку ответа студенту выделяется **45–60 минут**.
- 3. **Структура экзаменационного билета:** Каждый билет включает три элемента:
 - **Вопрос 1 (Теоретический):** Логико-математические основы, принципы фон Неймана или организация систем памяти/ввода-вывода.
 - **Вопрос 2 (Архитектурно-аналитический):** Устройство современных процессоров, параллельные системы, конвейеризация, когерентность кэш-кэшей.
 - **Вопрос 3 (Практическая задача):** Расчетное или схемотехническое задание (перевод чисел в IEEE 754, минимизация функций, трассировка ассемблерного кода или расчет задержек конвейера).

II. Критерии оценивания ответа на экзамене

Итоговая оценка выставляется на основе полноты, логичности и технической грамотности ответов на вопросы билета и дополнительные вопросы экзаменатора.

- **Оценка «Отлично»:** Студент демонстрирует исчерпывающие знания учебного материала. Свободно владеет терминологией (MMU, DMA, NUMA, MESI и др.), безошибочно чертит архитектурные схемы и временные диаграммы. Практическая задача решена верно и аргументированно. На дополнительные и перекрестные вопросы даны четкие, логичные ответы.
- **Оценка «Хорошо»:** Студент твердо знает материал, грамотно излагает суть архитектурных процессов, однако допускает несущественные неточности при формулировке законов или черчении второстепенных связей на схемах. Практическая задача решена верно, но возможны мелкие арифметические опечатки. Наводящие вопросы экзаменатора позволяют студенту самостоятельно исправить замеченные ошибки.
- **Оценка «Удовлетворительно»:** Студент демонстрирует знание только базовых положений дисциплины (например, принципы фон Неймана, основы двоичной арифметики), но испытывает серьезные трудности при объяснении сложных механизмов (конвейерные конфликты, виртуальная память, когерентность кэш-кэшей). Схемы строятся с ошибками. Практическая задача решена частично или с грубыми нарушениями алгоритма.
- **Оценка «Неудовлетворительно»:** Студент обнаруживает пробелы в знаниях фундаментального материала (не может объяснить цикл команд, путает функции ОЗУ и процессора). Схемы отсутствуют или фундаментально неверны. Практическая задача не решена. На дополнительные вопросы студент ответить не может.

III. Рекомендации по подготовке к экзамену

1. Систематизация теоретического материала

- **Фокус на архитектурных схемах:** Архитектура ЭВМ – визуальная дисциплина. Не заучивайте текст, учитесь «читать» и воспроизводить схемы вручную (структуру процессора, схему трансляции MMU, взаимодействие контроллера ПДП с системной шиной). Если вы можете нарисовать схему и стрелками показать движение данных, вы ответите на любой теоретический вопрос.
- **Понимание причинно-следственных связей:** Большинство архитектурных решений создавались для преодоления конкретных физических ограничений. Готовя вопрос, всегда отвечайте себе: *Какую проблему решает эта технология?* (Например: кэш сглаживает разницу в скорости ЦП и ОЗУ; конвейер решает проблему простоя исполнительных блоков; дополнительный код устраняет необходимость в аппаратном вычитателе).

2. Подготовка к практической части

Обязательно доведите до автоматизма алгоритмы решения типовых задач, встречавшихся в лабораторных работах:

- **Раздел 1:** Перевод чисел в формат IEEE 754 (не забывайте про смещение экспоненты +127 для одинарной точности) и получение двоичного дополнительного кода для отрицательных чисел.
- **Раздел 2–3:** Расчет количества тактов, необходимых для выполнения последовательности команд на конвейере с учетом структурных конфликтов или конфликтов по данным.
- **Раздел 4:** Расчет эффективного времени доступа к памяти при известных вероятностях промаха (H) в кэш-памяти разных уровней.

3. Стратегия поведения на самом экзамене

- **Рациональное распределение времени:** Начните подготовку с практической задачи—ее решение требует сосредоточенности и расчетов. Затем набросайте тезисы и графические схемы для теоретических вопросов.
- **Используйте точные термины:** Избегайте бытового языка. Говорите «регистр-счетчик команд (РС)», а не «штука, которая показывает, где мы сейчас»; «переполнение разрядной сетки», а не «число не поместилось».
- **Пишите разборчиво:** Коды команд на Ассемблере, таблицы истинности и индексы в формулах должны быть написаны четко. Любая нечитаемая цифра (например, путаница между 0 и 1 в двоичном коде) трактуется экзаменатором как ошибка.

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой