

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра параллельных вычислительных технологий

“УТВЕРЖДАЮ”
ДЕКАН ФПМИ
д.т.н. Тимофеев В. С.

“ ____ ” _____ г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Архитектура электронно-вычислительных машин и вычислительных систем

Образовательная программа: 01.03.02 Прикладная математика и информатика, профиль:
Компьютерное моделирование и информационные технологии

Курс: 1, семестр : 2

Факультет прикладной математики и информатики,

		Семестр
№	Вид деятельности	2
1	Всего зачетных единиц (кредитов)	3
2	Всего часов	108
3	Всего занятий в контактной форме, час.	45
4	Лекции, час.	36
5	Практические занятия, час.	0
6	Лабораторные занятия, час.	0
7	из них в активной и интерактивной форме, час.	7
8	Аттестация, час.	2
9	Консультации, час.	7
10	Самостоятельная работа, час.	63
11	Виды самостоятельной работы (курсовой проект, курсовая работа, РГЗ, подготовка к контрольной работе)	РГЗ
12	Вид аттестации	ДЗ

Рабочая программа составлена на основании федерального государственного образовательного стандарта высшего образования по направлению (специальности): 01.03.02 Прикладная математика и информатика

ФГОС введен в действие приказом №228 от 12.03.2015 г. , дата утверждения: 14.04.2015 г.

Место дисциплины в структуре учебного плана: Блок 1, вариативная

Рабочая программа разработана на основе компетентностной модели выпускника по направлению (специальности): 01.03.02 Прикладная математика и информатика

Рабочая программа обсуждена на заседании кафедры

ПВТ, протокол заседания кафедры №5 от 20.06.2017

Утверждена на совете факультета прикладной математики и информатики, протокол № 6 от 21.06.2017

Программу разработал:

доцент, к.т.н. Маркова В. П.

Заведующий кафедрой:

Ответственный за образовательную программу:

заведующий кафедрой Соловейчик Ю. Г.

1. Внешние требования

Таблица 1.1

Компетенция ФГОС: ОПК.3 способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям; в части следующих результатов обучения:
32. знать архитектуру современных компьютеров
Компетенция ФГОС: ОПК.4 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; в части следующих результатов обучения:
у7. уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе
Компетенция ФГОС: ПК.3 способность критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности; в части следующих результатов обучения:
31. знать основные тенденции развития компьютерных технологий

2. Требования НГТУ к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий
<i>Архитектура электронно-вычислительных машин и вычислительных систем</i>	
ОПК.3.32 знать архитектуру современных компьютеров	
1.О принципах организации ЭВМ, вычислительных систем и сетей.	Лекции; Самостоятельная работа
ПК.3.31 знать основные тенденции развития компьютерных технологий	
2.О способах параллельной обработки информации	Лекции; Самостоятельная работа
3.О тенденциях развития современных ЭВМ и ВС.	Лекции; Самостоятельная работа
ОПК.4.у7 уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе	
4.О современном состоянии развития современных компьютеров, вычислительных систем и сетей ЭВМ (список TOP500).	Лекции; Самостоятельная работа
ОПК.3.32 знать архитектуру современных компьютеров	
5.Базисные вычислительные модели. Основные компоненты традиционных ЭВМ.	Лекции; Самостоятельная работа
6.Организация памяти, способы управления памятью.	Лекции; Самостоятельная работа
7.Функционирование процессора. Технику конвейеризации. Причины остановки конвейера. Способы уменьшения остановок конвейера. SIMD-формат вычислений.	Лекции; Самостоятельная работа
8.Набор команд. Представление команды. Основные характеристики CISC и RISC-процессоров.	Лекции; Самостоятельная работа
9.Особенности организации связи в современных ЭВМ.	Самостоятельная работа
10.Принципы организации архитектур с параллелизмом на уровне команд: суперскалярные микропроцессоры и микропроцессоры с явным параллелизмом (EPIC).	Лекции; Самостоятельная работа

11. Принципы организации архитектур с параллелизмом на уровне потоков. Понятие треда. Варианты реализации многопоточности. Программное обеспечение (OpenMP, треды). Примеры реализации многоядерных процессоров. Компьютеры с разделяемой памятью (UMA, NUMA, cc-NUMA, СОМА) (мультипроцессоры). Компьютеры с распределенной памятью (мультикомпьютеры). Основные компоненты ВС. Примеры реализации современных мультипроцессоров и мультикомпьютеров и кластеров.	Лекции; Самостоятельная работа
12. Принципы организации архитектур с параллелизмом на уровне процессов. Компьютеры с разделяемой памятью (UMA, NUMA, cc-NUMA), компьютеры с распределенной памятью (мультикомпьютеры, кластеры). Компьютеры с Основными компонентами ВС. Сравнительный анализ коммуникационных сред. Программное обеспечение Примеры реализации современных мультипроцессоров и мультикомпьютеров и кластеров.	Лекции; Самостоятельная работа
13. По упрощенной схеме микропроцессора и вычислительной системы понять, как они работают, за счет каких архитектурных усовершенствований получены соответствующие характеристики производительности.	Лекции; Самостоятельная работа
14. Написать и отладить программу с учетом особенностей архитектуры ЭВМ (например, организации памяти, нескольких ядер и т.д.).	Лекции; Самостоятельная работа
15. Некоторые навыки практической работы на компьютерах разной архитектуры.	Лекции; Самостоятельная работа

3. Содержание и структура учебной дисциплины

Таблица 3.1

Темы лекций	Активные формы, час.	Часы	Ссылки на результаты обучения	Учебная деятельность
Семестр: 2				
Дидактическая единица: Принципы построения традиционных ЭВМ (принципы фон Неймана).				
1. Традиционная архитектура фон Неймана. Основные архитектурные принципы построения компьютера (ЭВМ). Компьютер фон Неймана. Узкие места компьютера фон Неймана и его усовершенствования..	0	2	1, 5	
Дидактическая единица: Организация памяти. Иерархия памяти.				
2. Иерархия памяти. Основной принцип построения иерархической памяти.	1	4	11, 13, 14, 2, 5, 6	
Дидактическая единица: Организация кэш-памяти				
3. Типичная схема иерархии памяти. Кэш-память. Алгоритмы отображения (прямое, ассоциативное, частично-ассоциативное). Сравнительный анализ алгоритмов отображения. Алгоритмы записи (сквозной, с обратным копированием и буферизированной сквозной). Сравнительный анализ алгоритмов записи. Алгоритмы замещения (FIFO, LRU). Проблемы поддержания когерентности.	1	4	11, 13, 14, 2, 5, 6	

Дидактическая единица: Организация виртуальная память, интерливинг.				
4. Понятие виртуальной памяти. Способы управления памятью: страничный, сегментный и странично-сегментный). Интерливинг.	1	3	11, 13, 14, 2, 5, 6	
Дидактическая единица: Процессор, набор команд				
5. Процессор, состав и функционирование. Набор команд.	1	4	1, 13, 14, 2, 3, 4, 8	
Дидактическая единица: Техника конвейеризации, причины остановки конвейера. Векторная обработка.				
6. Конвейерная обработка. Техника конвейеризации. Передача данных конвейере. Временные оценки сложности. Командный конвейер. Примеры командного конвейера. Количество ступеней. Причины приостановки конвейера и техника их преодоления. Арифметический конвейер. Представление данных. Основные арифметические операции. Примеры целочисленных конвейеров и конвейеров с плавающей точкой. Векторная обработка данных (SIMD-вычисления).	1	3	1, 13, 14, 2, 3, 4, 7	
Дидактическая единица: RISC-архитектура.				
7. Архитектура с сокращенным набором команд (RISC) Основные характеристики CISC-архитектуры. Формирование концепции RISC-архитектуры. Основные характеристики RISC-архитектур. Конвейер RISC-процессоров. Понятие регистрового окна. Оптимизирующий компилятор. Сравнительный анализ CISC и RISC архитектур.	1	3	1, 13, 14, 2, 3, 4, 8	
Дидактическая единица: Архитектуры с параллелизмом на уровне				
8. Классификация ILP-архитектур.	1	3	1, 10, 11, 12, 13, 15, 2, 3, 5	Основные семейства архитектур с параллелизмом на уровне команд: суперскаляр и VLIW/EPIC. Их преимущества, недостатки и способы их преодоления.
Дидактическая единица: Суперскалярные микропроцессоры				

9. Архитектура суперскалярных процессоров. Динамическое исполнение команд (предсказание переходов, переименование регистров, спекулятивное выполнение и выполнение вне порядка). Основные блоки суперскалярных процессоров. Причины, ограничивающие эффективность суперскалярных процессоров. Примеры процессоров (Opteron, Power PC 970).	0	3	1, 10, 11, 12, 13, 15, 5	
Дидактическая единица: Микропроцессоры с явным параллелизмом на уровне команд. Способы выявления скрытого параллелизма.				
10. Архитектура EPIC-процессоров (с длинным командным словом / явным параллелизмом на уровне команд). Особенности работы EPIC-процессора (спекуляция по коду и данным, предикатное исполнение команд, регистровый стек, программная конвейеризация циклов, ветвление). Микропроцессор Itanium 2. Сравнение способов выявления скрытого параллелизма в микропроцессорах обоих классов. Организация связи в ILP-архитектурах. Перспективы данного класса архитектур.	0	3	1, 10, 11, 12, 13, 15, 5	
Дидактическая единица: Организация архитектур с параллелизмом на уровне тредов.				
11. Мультиредовые архитектуры. Понятие треда. Варианты реализации многопоточности. Проблема поддержания когерентности данных. Программное обеспечение (OpenMP, треды). Примеры реализации современных многоядерных процессоров.	0	4	1, 11, 12, 13, 15, 5	

4. Самостоятельная работа обучающегося

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
Семестр: 2				
1	РГЗ	14, 15, 2, 8	20	2
: Эффективное программирование современных микропроцессоров : учебное пособие / [В. П. Маркова и др.] ; Новосиб. гос. техн. ун-т. - Новосибирск, 2014. - 147, [1] с. : ил., табл.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000190282				
2	Подготовка к занятиям	1, 10, 12, 13, 2, 3, 7, 8, 9	14	0

: Эффективное программирование современных микропроцессоров : учебное пособие / [В. П. Маркова и др.] ; Новосиб. гос. техн. ун-т. - Новосибирск, 2014. - 147, [1] с. : ил., табл.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000190282				
3	Дополнительная учебная деятельность	1, 10, 11, 12, 2, 3, 4, 5, 6, 7, 8, 9	0	0
: Эффективное программирование современных микропроцессоров : учебное пособие / [В. П. Маркова и др.] ; Новосиб. гос. техн. ун-т. - Новосибирск, 2014. - 147, [1] с. : ил., табл.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000190282				
4	Подготовка к аттестации	1, 10, 11, 12, 2, 5, 6, 7	29	5
: Эффективное программирование современных микропроцессоров : учебное пособие / [В. П. Маркова и др.] ; Новосиб. гос. техн. ун-т. - Новосибирск, 2014. - 147, [1] с. : ил., табл.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000190282				

5. Технология обучения

Для организации и контроля самостоятельной работы обучающихся, а также проведения консультаций применяются информационно-коммуникационные технологии (табл. 5.1).

Таблица 5.1

Деятельность	Информационно-коммуникационные технологии
Информирование	e-mail; Личный типовой сайт
Консультирование	e-mail; Чат
Контроль	e-mail
Размещение учебных материалов	e-mail; Личный типовой сайт; ЭБС

Таблица 5.2

Активные и интерактивные формы проведения занятий

№	Наименование активных форм	Коды формируемых компетенций
1	Дискуссия	ОПК.3;
Формируемые умения: з2. знать архитектуру современных компьютеров		
Краткое описание применения:		
Подробная информация об использовании технологии приводится в приложении №1 "Эффективное программирование современных микропроцессоров : учебное пособие / [В. П. Маркова и др.] ; Новосиб. гос. техн. ун-т. - Новосибирск, 2014. - 147, [1] с. : ил., табл.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000190282 "		

6. Правила аттестации обучающихся по учебной дисциплине

Для аттестации обучающихся по дисциплине используется балльно-рейтинговая система (БРС), позволяющая выставять оценки по традиционной шкале и 15-уровневой ECTS. Краткая информация о БРС приведена в табл. 6.1.

Таблица 6.1

Оцениваемые виды деятельности обучающихся	Мин. балл	Максимальный балл
Семестр: 2		
РГЗ №1: Измерение времени работы программы по выбранному алгоритму. Исследование влияния опций компилятора на время выполнения программы.	10	20

Контролирующие материалы приводятся в "Эффективное программирование современных микропроцессоров : учебное пособие / [В. П. Маркова и др.] ; Новосиб. гос. техн. ун-т. - Новосибирск, 2014. - 147, [1] с. : ил., табл. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000190282 "		
РГЗ №2: Исследование влияния параметров иерархии памяти на время выполнения программы.	15	30
Контролирующие материалы приводятся в "Эффективное программирование современных микропроцессоров : учебное пособие / [В. П. Маркова и др.] ; Новосиб. гос. техн. ун-т. - Новосибирск, 2014. - 147, [1] с. : ил., табл. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000190282 "		
РГЗ №3: Исследование векторных расширений на простейших алгоритмах	15	30
Контролирующие материалы приводятся в "Эффективное программирование современных микропроцессоров : учебное пособие / [В. П. Маркова и др.] ; Новосиб. гос. техн. ун-т. - Новосибирск, 2014. - 147, [1] с. : ил., табл. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000190282 "		
Зачет:	10	20

В таблице 6.2 представлено соответствие форм контроля заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.2

Коды компетенций ФГОС	Результаты обучения	Формы контроля		
		Защита РГЗ	Зачет	Прочее
ОПК.3	32. знать архитектуру современных компьютеров	+	+	+
ОПК.4	у7. уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе		+	
ПК.3	31. знать основные тенденции развития компьютерных технологий	+	+	

Фонд оценочных средств по дисциплине представлен в приложении № 1 к рабочей программе.

7. Литература

Основная литература

1. Орлов С. А. Организация ЭВМ и систем : [учебник для вузов] / С. А. Орлов, Б. Я. Цилькер. - СПб. [и др.], 2011. - 686 с. : ил., табл.
2. Новожилов О. П. Архитектура ЭВМ и систем : учебное пособие для бакалавров / О. П. Новожилов. - Москва, 2012. - 527 с. : ил., табл.
3. Жмакин А. П. Архитектура ЭВМ : [учебное пособие] / Анатолий Жмакин. - Санкт-Петербург, 2010. - 344 с. : ил., табл. + 1 CD-ROM. - На обл. в подзаг.: Функциональная организация ЭВМ, машинная арифметика и синтез устройств, программные модели АЛУ, архитектура микропроцессорных систем, программная модель учебной ЭВМ, лабораторный практикум и курсовое проектирование.
4. Матушкин Г. Г. Архитектура компьютера [Электронный ресурс] : учебное пособие / Г. Г. Матушкин ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000166425. - Загл. с экрана.

Дополнительная литература

1. Корнеев В. В. Современные микропроцессоры / Виктор Корнеев, Андрей Киселев. - СПб., 2003. - 440 с. : ил.
2. Степанов А. Н. Архитектура вычислительных систем и компьютерных сетей : [учебное пособие для вузов по специальности "Прикладная математика и информатика" (010200) и по направлению "Прикладная математика и информатика" (510200)] / А. Н. Степанов. - СПб. [и др.], 2007. - 508 с. : ил., табл. - Издательская программа 300 лучших учебников для высшей школы.
3. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы : учебное пособие для вузов / В. Г. Олифер, Н. А. Олифер. - СПб. [и др.], 2007. - 957 с. : ил.
4. Таненбаум Э. С. Архитектура компьютера : [перевод] / Э. Таненбаум. - СПб., 2006. - 698 с. : ил.

5. Корнеев В. В. Вычислительные системы / В. В. Корнеев. - М., 2004. - 510, [1] с. : ил.
6. Касперски К. Техника оптимизации программ. Эффективное использование памяти / Крис Касперски. - СПб., 2003. - 455, [1] с. : ил. + 1 CD-ROM.

Интернет-ресурсы

1. ЭБС НГТУ : <http://elibrary.nstu.ru/>
2. ЭБС «Издательство Лань» : <https://e.lanbook.com/>
3. ЭБС IPRbooks : <http://www.iprbookshop.ru/>
4. ЭБС "Znaniy.com" : <http://znaniy.com/>
5. :

8. Методическое и программное обеспечение

8.1 Методическое обеспечение

1. Эффективное программирование современных микропроцессоров : учебное пособие / [В. П. Маркова и др.] ; Новосиб. гос. техн. ун-т. - Новосибирск, 2014. - 147, [1] с. : ил., табл. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000190282

8.2 Специализированное программное обеспечение

1 GCC

9. Материально-техническое обеспечение

Презентационное оборудование

№	Наименование	Назначение
1	Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления)	для сопровождения лекций

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»

Кафедра параллельных вычислительных технологий

“УТВЕРЖДАЮ”
ДЕКАН ФПМИ
д.т.н., доцент В.С. Тимофеев
“ ” _____ г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УЧЕБНОЙ ДИСЦИПЛИНЫ

Архитектура электронно-вычислительных машин и вычислительных систем
Образовательная программа: 01.03.02 Прикладная математика и информатика, профиль:
Компьютерное моделирование и информационные технологии

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине Архитектура электронно-вычислительных машин и вычислительных систем приведена в Таблице.

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ОПК.3 способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	з2. знать архитектуру современных компьютеров	Архитектура EPIC-процессоров (с длинным командным словом / явным параллелизмом на уровне команд). Особенности работы EPIC-процессора (спекуляция по коду и данным, предикатное исполнение команд, регистровый стек, программная конвейеризация циклов, ветвление). Микропроцессор Itanium 2. Сравнение способов выявления скрытого параллелизма в микропроцессорах обоих классов. Организация связи в ILP-архитектурах. Перспективы данного класса архитектур. Архитектура суперскалярных процессоров. Динамическое исполнение команд (предсказание переходов, переименование регистров, спекулятивное выполнение и выполнение вне порядка). Основные блоки суперскалярных процессоров. Причины, ограничивающие эффективность суперскалярных процессоров. Примеры процессоров (Opteron, Power PC 970). Иерархия памяти. Основной принцип построения иерархической памяти. Мультипроцессорные архитектуры. Понятие треда. Варианты реализации многопоточности. Проблема поддержания когерентности данных. Программное обеспечение (OpenMP, треда). Примеры реализации современных многоядерных процессоров. Понятие виртуальной памяти. Способы управления памятью: страничный, сегментный и странично-сегментный). Интерливинг. Типичная схема иерархии	РГЗ, разделы 1-5 задания 1, разделы 1-3 задания 2, разделы 1-2 задания 3.	Зачет, вопросы 1-7, 11-13

		памяти. Кэш-память. Алгоритмы отображения (прямое, ассоциативное, частично-ассоциативное). Сравнительный анализ алгоритмов отображения. Алгоритмы записи (сквозной, с обратным копированием и буферизированный сквозной). Сравнительный анализ алгоритмов записи. Алгоритмы замещения (FIFO, LRU). Проблемы поддержания когерентности. Традиционная архитектура фон Неймана. Основные архитектурные принципы построения компьютера (ЭВМ). Компьютер фон Неймана. Узкие места компьютера фон Неймана и его усовершенствования..		
ОПК.4 способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	у7. уметь оценивать состояние и тенденции развития информационных технологий и информатики в современном обществе	Процессор, состав и функционирование. Набор команд.		Зачет, вопросы 8-10, 14-18
ПК.3/НИ способность критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности	з1. знать основные тенденции развития компьютерных технологий	Архитектура с сокращенным набором команд (RISC) Основные характеристики CISC-архитектуры. Формирование концепции RISC-архитектуры. Основные характеристики RISC-архитектур. Конвейер RISC-процессоров. Понятие регистрового окна. Оптимизирующий компилятор. Сравнительный анализ CISC и RISC архитектур. Конвейерная обработка. Техника конвейеризации. Передача данных конвейере. Временные оценки сложности. Командный конвейер. Примеры командного конвейера. Количество ступеней. Причины приостановки конвейера и техника их преодоления. Арифметический конвейер. Представление данных. Основные арифметические операции. Примеры	РГЗ разделы 1-2 задания 3.	Зачет, вопросы 8-10, 18

		целочисленных конвейеров и конвейеров с плавающей точкой. Векторная обработка данных (SIMD-вычисления). Процессор, состав и функционирование. Набор команд.		
--	--	---	--	--

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 2 семестре - в форме дифференцированного зачета, который направлен на оценку сформированности компетенций ОПК.3, ОПК.4, ПК.3/НИ. Из времени на самостоятельную работу на подготовку к промежуточной аттестации выделяется 29 часов.

Зачет проводится в устной форме, по билетам, составленным из вопросов, приведенных в паспорте зачета, позволяющих оценить показатели сформированности соответствующих компетенций

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 2 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (работа) (РГЗ(Р)). Требования к выполнению РГЗ(Р), состав и правила оценки сформулированы в паспорте РГЗ(Р).

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.3, ОПК.4, ПК.3/НИ, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Паспорт зачета

по дисциплине «Архитектура электронно-вычислительных машин и вычислительных систем», 2 семестр

1. Методика оценки

Зачет проводится в устной форме, по билетам. Билет формируется по следующему правилу: первый вопрос выбирается из диапазона вопросов 1-9, второй вопрос из диапазона вопросов 10-18 (список вопросов приведен ниже). В ходе экзамена преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

Форма билета для зачета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет ФПМИ

Билет № _____

к зачету по дисциплине «Архитектура электронно-вычислительных машин и
вычислительных систем»

Вопрос 1 Основные архитектурные принципы построения компьютера. Компьютер фон Неймана. Узкие места компьютера фон Неймана и его усовершенствования.

Вопрос 2. Многопоточные архитектуры. Понятие потока. Варианты реализации многопоточности.

Утверждаю: зав. кафедрой ПВТ _____ профессор, Малышкин В.Э.
(подпись)

(дата)

2. Критерии оценки

- Ответ на билет для зачета считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений, в ответах на вопросы допускает принципиальные ошибки, оценка составляет *от 0 до 9 баллов*.
- Ответ на билет для зачета засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений, в ответах на вопросы допускает не принципиальные ошибки, оценка составляет *от 10 до 15 баллов*.
- Ответ на билет для зачета билет засчитывается на **базовом** уровне, если студент при ответе на вопросы формулирует основные понятия, законы, дает характеристику

процессов, явлений, проводит анализ причин, условий, может представить качественные характеристики процессов, не допускает серьезных ошибок в ответах на вопросы, оценка составляет от 16 до 17 баллов.

- Ответ на билет для зачета билет засчитывается на **продвинутом** уровне, если студент при ответе на вопросы проводит сравнительный анализ подходов, проводит комплексный анализ, выявляет проблемы, предлагает механизмы решения, способен представить количественные характеристики определенных процессов, приводит конкретные примеры из практики, не допускает ошибок в ответах на вопросы, оценка составляет от 18 до 20 баллов.

3. Шкала оценки

Зачет считается сданным, если сумма баллов по всем заданиям билета оставляет не менее 10 баллов (из 20 возможных).

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Вопросы к зачету по дисциплине «Архитектура электронно-вычислительных машин и вычислительных систем»

Вопрос 1. Основные архитектурные принципы построения компьютера. Компьютер фон Неймана. Узкие места компьютера фон Неймана и его усовершенствования.

Вопрос 2. Иерархия памяти. Основной принцип построения иерархической памяти. Типичная схема иерархии памяти. Банки памяти. Интерливинг.

Вопрос 3. Кэш-память. Принцип построения кэш-памяти.

Вопрос 4. Алгоритмы отображения команд и данных. Сравнительный анализ алгоритмов.

Вопрос 5. Алгоритмы записи.

Вопрос 6. Алгоритмы замещения. Проблемы поддержания когерентности.

Вопрос 7. Схемы управления памятью. Понятие виртуальной памяти. Два способа управления виртуальной памятью. Сравнение способов управления.

Вопрос 8. Процессор, состав, функционирование. Набор команд. Конвейерная обработка. Причины приостановки конвейера и техника их преодоления.

Вопрос 9. Архитектура с сокращенным набором команд (RISC). Основные характеристики RISC-процессоров. Конвейер RISC-процессоров. Понятие регистрового окна. Оптимизирующий компилятор. Сравнение CISC и RISC архитектур.

Вопрос 10. Понятие последовательного и параллельного исполнения. Уровни параллелизма. Классификация Флинна. Уточненная классификация. Идея SIMD-расширений. Форматы SIMD-команд.

Вопрос 11. Классификация архитектур с параллелизмом на уровне команд. Архитектура суперскалярных микропроцессоров. Базисные принципы организации суперскалярных микропроцессоров.

Вопрос 12. Архитектура микропроцессоров с явным параллелизмом на уровне команд (EPIC). Базисные принципы организации. Сравнение способов выявления скрытого параллелизма у трех типов архитектур. Перспективы данного класса архитектур.

Вопрос 13. Многопоточные архитектуры. Понятие потока. Варианты реализации многопоточности.

Вопрос 14. Классификация архитектур с параллелизмом на уровне процессов (вычислительные системы). Архитектуры с разделяемой памятью (UMA, NUMA, cc-NUMA). Структура узла. Коммуникационные среды. Программное обеспечение. Поддержка когерентности кэш-памяти. Примеры современных мультимикропроцессоров.

Вопрос 15. Архитектуры с распределенной памятью (мультимикропроцессоры). Коммуникационные среды. Программное обеспечение. Примеры современных

мультимониторных.

Вопрос 16. Кластеры. Проектирование кластерных систем. Организация памяти. Структура узла. Сеть связи. Поддержка кэш-когерентности. Сравнительный анализ коммуникационных сред. Программное обеспечение. Примеры современных кластеров.

Вопрос 17. Анализ списка TOP500. Тенденции развития микропроцессоров и вычислительных систем. Оценки производительности.

Вопрос 18. Инструментальное программное обеспечение программиста. Оптимизирующий компилятор. Компоновщик. Отладчик и профилировщик. Назначение и основные функции.

Паспорт расчетно-графического задания (работы)

по дисциплине «Архитектура электронно-вычислительных машин и вычислительных систем», 2 семестр

1. Методика оценки

В рамках расчетно-графического задания (работы) по дисциплине студенты должны построить программную реализацию для задачи по заданному варианту.

При выполнении расчетно-графического задания (работы) студенты должны выбрать алгоритм решения задачи по заданному варианту, построить его программную реализацию, оптимизировать ее и собрать статистику времени исполнения.

Отчет должен содержать титульный лист, описание целей работы, листинги программ, таблицы и графики с собранной статистикой и выводы по проделанной работе.

На оценку работы влияет соответствие реализованного набора функций исходному описанию задачи, качество построенной реализации (в первую очередь надежность и эффективность), полнота собранной статистики времени выполнения, а также качество оформления полученных результатов в виде отчета по РГР.

2. Критерии оценки

- Работа считается **не выполненной**, если отсутствует работающая реализация задачи по выбранному варианту, оценка составляет от 0 до 39 баллов.
- Работа считается выполненной **на пороговом** уровне, если реализация для задачи по выбранному варианту написана не полностью и с недочетами, оценка составляет от 40 до 57 баллов.
- Работа считается выполненной **на базовом** уровне, если полностью написана, но не оптимизирована по времени исполнения реализация для задачи по выбранному варианту, собрана не вся необходимая статистика, оценка составляет от 58 до 69 баллов.
- Работа считается выполненной **на продвинутом** уровне, если полностью написана и оптимизирована по времени исполнения реализация для задачи по выбранному варианту, собрана вся необходимая статистика, оценка составляет от 70 до 80 баллов.

3. Шкала оценки

РГЗ считается сданным, если выставленные баллы составляют не менее 40 баллов (из 80 возможных).

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Примерный перечень тем РГЗ(Р)

Задание №1. Определение времени работы прикладных программ, исследование зависимости времени выполнения программ от уровня оптимизации компилятора.

Цели работы:

1. Научиться измерять интервалы времени в программах на языке C.
2. Исследовать зависимость времени работы прикладных программ от уровня оптимизации компилятора.

Форма отчета:

Результаты работы представляются в виде файла с отчётом о проделанной работе в произвольной форме. Отчёт должен содержать следующую информацию:

1. Исполнитель (ФИО, № группы)
2. Вариант задания
3. Таблица измерений, содержащая следующие поля:
 - Уровень оптимизации
 - Способ измерения времени
 - Величина параметра прикладного алгоритма
 - Время работы функции (с указанием единиц измерений)
4. Команды компиляции и запуска (с учётом регистра символов)
5. Исходный код программы

Задание:

1. Реализовать на языке C или C++ одну из функций на выбор (см. след. раздел). Проверить правильность работы программы.
2. Измерить время работы вычислительной функции полученной программы с использованием трёх различных способов измерения времени из приложения Б. При этом необходимо придерживаться методики измерения времени.
3. Параметр прикладного алгоритма необходимо выбрать таким образом, чтобы относительная погрешность измерения времени не превышала 1%.
4. Реализованную в п. 1 программу скомпилировать компилятором GNU C/C++ Compiler (далее GCC) с уровнями оптимизации -O0, -O1, -O2, -O3 (регистр ключей имеет значение), а также под архитектуру процессора, на котором будет осуществляться запуск программы. Измерить время работы программы при фиксированном параметре – одном и том же значении N для всех уровней.
5. Объяснить зависимость времени работы программы от уровня оптимизации.

Варианты функций:

- Вычисление числа Пи с помощью разложения в ряд по формуле:

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \dots,$$

параметр N – число итераций.

- Вычисление определенного интеграла сложной функции методом трапеций:

$$\int_a^b f(x) dx = \frac{b-a}{N} \sum_{k=0}^{N-1} \frac{f(x_k) + f(x_{k+1})}{2},$$

$f(x) = e^x \sin(x)$, $a = 0$, $b = \pi$.

Параметр: N - число интервалов.

- Вычисление квадратного корня с помощью алгоритма Ньютона:

Сходящаяся серия итераций: $a_{i+1} = (a_i + x / a_i) / 2$.

Начальное значение a_0 можно взять произвольным.

Параметр: N - число итераций.

- Сортировка методом пузырька.

Дан массив случайных чисел длины N. На первой итерации попарно упорядочиваются все соседние элементы; на второй - все, кроме последнего; на третьей - кроме последнего и предпоследнего и т.п.

Параметр: N - размер массива.

- Вычисление числа Пи метом Монте-Карло.

Генерируется N точек (x;y), равномерно распределенных на квадрате $[0;1] \times [0;1]$. Вычисляется M - число точек, попавших в четверть круга с радиусом 1 ($x^2 + y^2 \leq 1$). Число Пи можно приближенно вычислить по формуле:

$$\pi \approx \frac{4M}{N}.$$

Параметр: N – число точек.

Контрольные вопросы:

1. Почему при многократном измерении времени работы программы наиболее точным является минимальное время, а не среднее?
2. Почему не существует одного универсального способа измерения времени?
3. Назовите способы измерения времени работы программы. Перечислите их особенности.
4. Каким способом лучше измерять большие промежутки времени (порядка нескольких часов)?
5. Каким способом лучше измерять малые промежутки времени (порядка времени работы нескольких команд процессора)?
6. Всегда ли оптимизирующая компиляция позволяет уменьшить время работы программы?
7. Чем отличается общая оптимизация от оптимизации под архитектуру?

Задание №2. Исследовать влияние порядка обхода данных в кэш-памяти на время работы программы.

Цели работы:

1. Исследовать влияние кэш-памяти на время работы программ обработки данных.
2. Исследовать влияние порядка обработки данных в памяти на время работы программы.
3. Экспериментально определить некоторые параметры кэш-памяти (объем, размер строки и т.д.)

Задание:

1. Написать программу, многократно выполняющую чтение элементов массива заданного размера. Элементы массива представляют собой связный список, в котором значение очередного элемента представляет собой номер следующего. Способы обхода массива —

прямой, обратный и случайный.

2. Построить графики зависимости среднего времени обращения к элементу массива от размера обрабатываемого массива для трех видов обхода. На графиках должно быть видно влияние кэш-памяти (1 и 2 уровня).

3. По результатам измерений сделать вывод о скорости различных способов обхода массива, а также о размерах различных уровней кэш-памяти (насколько полученные результаты соответствуют действительности).

Форма отчета

Результаты работы представляются в виде файла с отчётом о проделанной работе в произвольной форме. Отчёт должен содержать следующую информацию:

1. Исполнитель (ФИО, № группы)
2. График зависимостей среднего времени доступа к одному элементу массива от общего размера массива (для 3-х вариантов обхода, на одном графике).
3. Команды компиляции и запуска программы
4. Предположение о размерах кэша 1-го и 2-го уровней (на основе графика)
5. Исходный код программы

Контрольные вопросы:

1. Объясните, почему случайный обход массива выполняется, как правило, дольше, чем последовательные.
2. Почему для малых размеров массива все три способа обхода массива работают с одинаковой скоростью?
3. Как будет вести себя график, если его неограниченно продолжить в сторону увеличения размера массива?

Задание №3 «Использование векторных расширений»

Цели работы:

1. Научиться использовать в программах на языке C векторные расширения архитектуры x86.
2. Исследовать уменьшение времени работы программы при векторизации вычислений.

Задание:

Обращение матрицы A размером $N \times N$ можно выполнить с помощью разложения в ряд:

$$A^{-1} = (I + R + R^2 + \dots)B,$$

где $R = I - BA$,

$$B = \frac{A^T}{\|A\|_1 \cdot \|A\|_\infty}, \quad \|A\|_1 = \max_j \sum_i |A_{ij}|, \quad \|A\|_\infty = \max_i \sum_j |A_{ij}|,$$

I - единичная матрица (на диагонали - единицы, остальные - нули), A_{ij} - элемент матрицы A с индексом (i,j) .

1) Написать три варианта программы вычисления обратной матрицы:

- без использования специальных расширений (обычный вариант),
- с использованием встроенных векторных функций расширения SSE.
- с использованием библиотеки BLAS.

Каждый вариант программы:

- оптимизировать по скорости, насколько это возможно,
- проверить на правильность на небольшом тесте.

Использовать тип данных float.

Размер N предполагать кратным четырем.

2) Сравнить время работы четырех вариантов программы для $N=1024$, число шагов 10:

- обычный без векторизации компилятором,
- обычный с векторизацией компилятором (нужно указать тип процессора),
- с ручной векторизацией,
- с использованием BLAS.

Определить:

- среднее время одной итерации цикла (умножение + сложение матриц),
- время вычислений вне цикла (общее время минус время в цикле).

Замер времени выполнить несколько раз, в качестве результата взять минимальное время. Для измерения времени использовать функцию times (измерения времени работы процесса).

По результатам измерений сделать вывод.

Контрольные вопросы:

1. Объясните основную идею векторизации вычислений.
2. Любую ли программу можно векторизовать?
3. Какие векторные регистры используются в расширении SSE?