



МИНОБРНАУКИ РОССИИ

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

Факультет электроники и вычислительной техники

УТВЕРЖДЕНО

Факультет электроники и вычислительной
техники

Декан Авдеюк О.А.
г.

Низкоуровневая оптимизация приложений и программирование ускорителей

рабочая программа дисциплины (модуля, практики)

Закреплена за кафедрой	Электронно-вычислительные машины и системы		
Учебный план	Направление 09.04.01 Информатика и вычислительная техника Программа "Встраиваемые вычислительные системы в робототехнике и интернете вещей"		
Профиль			
Квалификация	Магистр		
Срок обучения	2 года		
Форма обучения	очная	Общая трудоемкость	3 ЗЕТ
Виды контроля в семестрах:	экзамены 3		

Семестр(Курс.Номер семестра на курсе)	3(2.1)		Итого	
	УП	ПП	УП	ПП
Практические	8	8	8	8
Лабораторные	16	16	16	16
Итого ауд.	24	24	24	24
Контактная работа	24.35	24.35	24.35	24.35
Сам. работа	48	48	48	48
Часы на контроль	35.65	35.65	35.65	35.65
Практическая подготовка	0	0	0	0
Итого трудоемкость в часах	108	108	0	0

ЛИСТ ОДОБРЕНИЯ, СОГЛАСОВАНИЯ И АКТУАЛИЗАЦИИ РАБОЧЕЙ ПРОГРАММЫ

Разработчик(и) программы:

зав. каф. Андреев А.Е. кпн

доцент Егунов В.А. ктн

Рецензент(ы):

(при наличии)

Рабочая программа дисциплины (модуля, практики)

Низкоуровневая оптимизация приложений и программирование ускорителей

разработана в соответствии с ФГОС ВО:

Федеральный государственный образовательный стандарт высшего образования - магистратура по направлению подготовки 09.04.01 Информатика и вычислительная техника (приказ Минобрнауки России от 19.09.2017 г. № 918)

составлена на основании учебного плана:

Направление 09.04.01 Информатика и вычислительная техника

Программа "Встраиваемые вычислительные системы в робототехнике и интернете вещей"

Профиль:

утвержденного учёным советом вуза от 05.06.2019 протокол № 12.

Рабочая программа одобрена на заседании кафедры

Электронно-вычислительные машины и системы

номер протокола 2019 г.

Зав. кафедрой Андреев Андрей Евгеньевич

СОГЛАСОВАНО:

Факультет электроники и вычислительной техники

Председатель НМС факультета: Авдеюк О.А.

Протокол заседания НМС от

г. №

Рабочая программа дисциплины (модуля, практики) актуализирована 31.08.2023

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ, ПРАКТИКИ). ВИД, ТИП ПРАКТИКИ, СПОСОБ И ФОРМА (ФОРМЫ) ЕЕ ПРОВЕДЕНИЯ.
Целью дисциплины является знакомство с программированием ускорителей и оптимизацией приложений для центральных процессоров.
Задачи дисциплины :
1. Знакомство с программированием графических ускорителей с помощью технологии OpenCL
2. Знакомство с приемами низкоуровневой оптимизации приложений для CPU.
3. Знакомство с высокопроизводительными вычислительными библиотеками.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ, ПРАКТИКИ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ОП:	Б1.В.ДВ.01
2.1	Требования к предварительной подготовке обучающегося:
2.1.1	Алгоритмы и структуры данных
2.1.2	Программное обеспечение инфокоммуникационных систем
2.1.3	Современные операционные системы
2.1.4	Параллельные и распределенные вычисления
2.2	Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
2.2.1	Выполнение и защита выпускной квалификационной работы
2.2.2	Производственная практика: Преддипломная практика
3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ, ПРАКТИКИ)	
ОПК-2: Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач;	
<i>ОПК-2.1: Знать: современные интеллектуальные технологии для решения профессиональных задач.</i>	
Результаты обучения: Знает существующие методы и алгоритмы оптимизации приложений, интеллектуальные технологии для решения профессиональных задач	
<i>ОПК-2.2: Уметь: обосновывать выбор современных интеллектуальных технологий и программной среды при разработке оригинальных программных средств для решения профессиональных задач.</i>	
Результаты обучения: Умеет создавать и производить оптимизацию параллельных программ для решения профессиональных задач	
<i>ОПК-2.3: Иметь навыки: разработки оригинальных программных средств, в том числе с использованием современных интеллектуальных технологий, для решения профессиональных задач.</i>	
Результаты обучения: Владеет навыками низкоуровневой оптимизации приложений и программирования ускорителей для решения профессиональных задач	
ОПК-5: Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем;	
<i>ОПК-5.1: Знать: современное программное и аппаратное обеспечение информационных и автоматизированных систем.</i>	
Результаты обучения: Знает общие принципы создания и оптимизации параллельных алгоритмов и программ, основные технологии параллельного программирования, программирования ускорителей	
<i>ОПК-5.2: Уметь: модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем для решения профессиональных задач.</i>	
Результаты обучения: Умеет создавать и оптимизировать программы для многопроцессорных систем с общей памятью, программировать ускорители	
<i>ОПК-5.3: Владеть: навыками разработки программного и аппаратного обеспечения информационных и автоматизированных систем для решения профессиональных задач.</i>	
Результаты обучения: Владеет навыками разработки и оптимизации параллельных алгоритмов и программ для параллельных вычислительных систем с общей памятью, программирования ускорителей	
ОПК-6: Способен разрабатывать компоненты программно-аппаратных комплексов обработки информации и автоматизированного проектирования;	
<i>ОПК-6.1: Знать: аппаратные средства и платформы инфраструктуры информационных технологий, виды, назначение, архитектуру, методы разработки и администрирования программно-аппаратных комплексов объекта профессиональной деятельности.</i>	
Результаты обучения: Знает архитектурные принципы построения параллельных вычислительных систем, методы низкоуровневой оптимизации приложений для вычислительных систем с различной архитектурой	
<i>ОПК-6.2: Уметь: анализировать техническое задание, разрабатывать и оптимизировать программный код для решения задач обработки информации и автоматизированного проектирования.</i>	
Результаты обучения: Умеет разрабатывать и осуществлять низкоуровневую оптимизацию приложений в соответствии с техническим заданием	

ОПК-6.3: Владеть: навыками составления технической документации по использованию и настройке компонентов программно-аппаратного комплекса.

Результаты обучения: Владеет навыками низкоуровневой оптимизации приложений, программировании ускорителей, составления технической документации

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ, ПРАКТИКИ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Форма контроля
1	Раздел 1. Обучение			
1.1	ЗНАКОМСТВО С ТЕХНОЛОГИЕЙ OPENCL ПОНЯТИЕ О НЕОДНОРОДНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ И ПРОГРАММАХ ДЛЯ НИХ /Тема/	3	0	
1.1.1	Понятие о неоднородных вычислительных системах и программах для них. Общая характеристика технологии OpenCL. Создание простых приложений на OpenCL. Структура программы для хоста. Код kernel. /Пр/	3	2	Э, К, Ко
1.1.2	Знакомство с OpenCL. Создание простых приложений с запуском на GPU /Лаб/	3	4	К, Ко
1.1.3	Подготовка к выполнению заданий по контрольной работе, подготовка к текущему контролю успеваемости /Ср/	3	12	К, Ко
1.2	РАБОТА С РАЗНЫМИ ТИПАМИ ПАМЯТИ И ОПТИМИЗАЦИЯ ПРИЛОЖЕНИЙ НА OPENCL МОДЕЛЬ ПАМЯТИ OPENCL /Тема/	3	0	
1.2.1	Модель памяти OpenCL. Использование разных типов памяти. Векторизация вычислений. Использование других способов оптимизации вычислений. /Пр/	3	2	Э, К, Ко
1.2.2	Работа с разными типами памяти и оптимизация приложения OpenCL на GPU /Лаб/	3	4	К, Ко
1.2.3	Подготовка к выполнению заданий по контрольной работе, подготовка к текущему контролю успеваемости /Ср/	3	12	К, Ко
1.3	ОПТИМИЗАЦИЯ ПРИЛОЖЕНИЙ ДЛЯ СИСТЕМ С ОБЩЕЙ ПАМЯТЬЮ ДЕКОМПОЗИЦИИ ЗАДАЧ /Тема/	3	0	
1.3.1	Оптимизация приложений для систем с общей памятью. Использование кэш – памяти, способы повышения производительности. Кэш – оптимизация на примере задач линейной алгебры. Векторизация как способ низкоуровневой оптимизации приложений. Способы использования векторизации /Пр/	3	2	Э, К, Ко
1.3.2	Использование кэш – памяти для оптимизации программ, векторизация вычислений /Лаб/	3	4	К, Ко
1.3.3	Подготовка к выполнению заданий по контрольной работе, подготовка к текущему контролю успеваемости /Ср/	3	12	К, Ко
1.4	ИЗУЧЕНИЕ БИБЛИОТЕКИ INTEL MKL. СТРУКТУРА INTEL MKL /Тема/	3	0	
1.4.1	Изучение библиотеки Intel MKL.BLAS, Parallel BLAS, LAPACK, ScaLapack, PARDISO, Iterative sparse solvers, FFTs. Хранение векторов, матриц, треугольных, разреженных матриц Основные функции BLAS, LAPACK. /Пр/	3	2	Э, К, Ко
1.4.2	Изучение библиотеки Intel MKL /Лаб/	3	4	К, Ко
1.4.3	Подготовка к выполнению заданий по контрольной работе, подготовка к текущему контролю успеваемости /Ср/	3	12	К, Ко
2	Раздел 2. Промежуточная аттестация			
2.1	Экзамен /Тема/	3	0	
2.1.1	Подготовка к экзамену /Экзамен/	3	35.65	Э
2.1.2	Контактная работа с ППС /КоПа/	3	0.35	Э

Примечание. Формы контроля: Эк – экзамен, К- контрольная работа, Ко- контрольный опрос, Сз- семестровое задание, З-зачет, ОП- отчет по практике.

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Оценочные средства планируемых результатов обучения представлены в виде фондов оценочных средств (ФОС), разработанных в соответствии с локальным нормативным актом университета. ФОС может быть представлен в Приложении к рабочей программе.

Темы письменных работ (контрольная работа)

На контрольную работу студенту выдается индивидуальное задание (по вариантам), заключающееся в разработке по варианту нескольких программ с использованием технологий параллельного программирования.

Работа выполняется в письменной форме в течение 10 недель с момента выдачи задания. Контрольный срок сдачи – последний месяц семестра.

Примерное содержание контрольной работы

1. Титульный лист.
2. Формулировка варианта задания.
3. Основная часть, включающая:
 - 1) решение первой задачи,
 - 2) решение второй задачи,
 - 3) решение третьей задачи,
 - 4). Список использованных источников (включая источники Интернет).

При описании решенных задач необходимо привести описание решения с использованием диаграмм UML, экранных форм разработанных программ. Также необходимо привести коды программ.

Правила оформления курсовой работы

- курсовая работа оформляется в редакторе MS Word / OpenOffice (*.doc, *.docx, *.odt);
- листы формата A4, ориентация книжная;
- поля: левое – 2 см, остальные – по 1 см;
- шрифт – Times New Roman;
- размер шрифта 14 pt;
- междустрочный интервал – 1,5;
- абзацный отступ – 1,25 см;
- нумерация страниц сквозная, номер на первой странице не ставится;
- в конце работы необходим список использованной литературы согласно ГОСТ Р 7.0.5 – 2008;
- объем работы зависит от степени раскрытия основных пунктов курсовой работы.

Примеры заданий к контрольной работе.

Варианты первой задачи.

Создать процессы, которые выполняются в последовательности, определяемой графом запуска в соответствии с вариантом задания.

Варианты второй задачи.

Необходимо разработать программу, выполняющую простые матричные вычисления по вариантам, векторизовать вычисления

Варианты третьей задачи

Необходимо разработать программу для параллельной обработки больших матриц. В качестве алгоритмов обработки матриц можно использовать алгоритмы Фокса и Кэннона умножения матриц, другие алгоритмы по заданию преподавателя.

Вопросы промежуточной аттестации

1. Поясните этапы работы программы на хосте (подготовка и запуск ядра).
2. Поясните параметры заданной преподавателем функции OpenCL.
3. Что такое work-item ? Рабочая группа ? Их назначение ?
4. Поясните модель памяти OpenCL.
5. Назначение OpenCL, отличие от CUDA. Преимущества и недостатки OpenCL
6. Что такое ускорение и эффективность параллельной программы?
7. Что такое суперлинейное ускорение?
8. Для чего нужна кеш-память процессора?
9. Поясните понятие «когерентность кешей».
10. Каким образом обеспечивается когерентность кеш-памяти?
11. Поясните понятие процесса, потока, основные механизмы синхронизации вычислительных потоков.
12. Что такое информационная зависимость? Какие виды информационных зависимостей Вы знаете?
13. Расскажите механизм работы протоколов обеспечения когерентности кешей на примере протокола MSI.
14. Назовите основные составляющие OpenMP.
15. Поясните модель памяти OpenMP
16. Назовите основные распараллеливающие конструкции OpenMP.
17. Как в OpenMP. осуществляется распараллеливание циклов?
18. Как в OpenMP. Осуществляется балансировка параллельных циклов?
19. Поясните механизм использования параллельных секций OpenMP.
20. Что такое атомарная операция, критическая секция в OpenMP?

21.	Барьерная синхронизация. Виды барьеров.
22.	Что такое MPI?
23.	Что такое группа процессов, коммуниторатор?
24.	Каким образом осуществляется запуск программ MPI?
25.	Типы данных MPI.
26.	Приведите пример простейшей программы MPI.
27.	Что такое операции «точка-точка» в MPI?
28.	Какие типы операций типа «точка-точка» Вы можете назвать в MPI?
29.	Блокирующие операции обмена типа «точка-точка»
30.	Неблокирующие операции обмена типа «точка-точка»
31.	Коллективные операции.
32.	Коллективная операция рассылки данных.
33.	Коллективные операция распределения данных.
34.	Коллективные операция сбора данных.
35.	Что такое векторизация вычислений?
36.	Приведите краткую характеристику расширений команд процессора MMX, SSE, AVX.
37.	Какую роль играет выравнивание данных при выполнении векторных инструкций? Приведите примеры.
38.	Интринсики. Назначение. Примеры использования.
39.	Поддержка векторизации компиляторами. Примеры.
40.	Что такое «неоднородная параллельная вычислительная система»? Приведите примеры.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ, ПРАКТИКИ)

6.1. Рекомендуемая литература

	Авторы, составители	Заглавие	Издательство, год.	Электронный адрес
Л.1	Вальковский В. А.	Распараллеливание алгоритмов и программ: Структурный подход	М.: Радио и связь, 1989	
Л.2	Миренков Н. Н.	Параллельное программирование для многомодульных вычислительных систем	М.: Радио и связь, 1989	
Л.3	Андреев А. Е., Егунов В. А.	Параллельное программирование: учеб.-метод. пособие	Волгоград: ВолгГТУ, 2019	
Л.4	Андреев А. Е., Егунов В. А.	Параллельное программирование: учеб.-метод. пособие	Волгоград: ВолгГТУ, 2019	

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Романенко А.А., Лаевский Ю.М. Оптимизация приложений с использованием библиотеки Intel MKL
Э2	Тоуманен Б. Программирование GPU при помощи Python и CUDA

6.3 Перечень программного обеспечения

6.3.1.1	Microsoft Visual Studio 2017 Enterprise; Лабораторные работы
6.3.1.2	Microsoft MPI, GCC, Intel Par-allel Studio XE, Intel MKL; Лабораторные работы

6.4 Перечень информационных справочных систем и электронных библиотечных систем (ЭБС)

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ, ПРАКТИКИ) /ОБОРУДОВАНИЕ

7.1	Лаборатория сетевых технологий / Мультимедийный класс
7.2	1) ПЭВМ Intel DualCore 2ГГц / 2Гб RAM / LCD 19" - 8 шт.; 2) экран EliteScreens; 3) проектор Acer 1200;
7.3	Учебная лаборатория
7.4	1) Ноутбуки HP Elitebook 8460p – 4 шт., 2) Ноутбуки HP EliteBook 8570p - 4 шт. 3) Ноутбук Lenovo ThinkPad T420 – 4 шт. 4) экран EliteScreens; 5) проектор Acer 1203; 6) доступ в Интернет и к наукометрическим базам данных
7.5	Аудитория для самостоятельной работы обучающихся./Учебная мебель, компьютерная техника с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета (читальный зал информационно-библиотечного центра)

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ, ПРАКТИКИ)

Андреев А. Е., Егунов В. А., Жариков Д. Н., Малолетков В. А. Реализация вычислительно-интенсивных алгоритмов на гибридных системах с реконфигурируемыми сопроцессорами: монография. Волгоград: ВолгГТУ, 2013

Андреев А. Е., Егунов В. А., Шаповалов О. В. Технологии программирования многопроцессорных систем: учеб. пособие Волгоград: ВолгГТУ, 2015

Андреев А. Е., Егунов В. А. Параллельное программирование: учеб.-метод. пособие Волгоград: ВолгГТУ, 2019