

МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»

УТВЕРЖДАЮ:

Директор ИТТСУ



П.Ф. Бестемьянов

26 мая 2020 г.

Кафедра «Автоматика, телемеханика и связь на железнодорожном транспорте»

Автор Ваньшин Александр Евгеньевич, к.т.н.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Основы микропроцессорной техники и прикладное программирование



Специальность: 23.05.05 – Системы обеспечения движения поездов

Специализация: Телекоммуникационные системы и сети железнодорожного транспорта

Квалификация выпускника: Инженер путей сообщения

Форма обучения: очная

Год начала подготовки 2020

<p style="text-align: center;">Одобрено на заседании Учебно-методической комиссии института Протокол № 10 26 мая 2020 г. Председатель учебно-методической комиссии</p>  <p style="text-align: right;">С.В. Володин</p>	<p style="text-align: center;">Одобрено на заседании кафедры</p> <p>Протокол № 8 21 мая 2020 г. Заведующий кафедрой</p>  <p style="text-align: right;">А.А. Антонов</p>
---	--

Рабочая программа учебной дисциплины (модуля) в виде электронного документа выгружена из единой корпоративной информационной системы управления университетом и соответствует оригиналу

Простая электронная подпись, выданная РУТ (МИИТ)
ID подписи: 21905
Подписал: Заведующий кафедрой Антонов Антон
Анатольевич
Дата: 21.05.2020

Москва 2020 г.

1. ЦЕЛИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целью преподавания дисциплины является формирование у студентов базовых знаний и умений в области функционирования, построения и применения микропроцессорной техники для создания цифровых программных систем управления объектами энергоснабжения электрических железных дорог

2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ОП ВО

Учебная дисциплина "Основы микропроцессорной техники и прикладное программирование" относится к блоку 1 "Дисциплины (модули)" и входит в его вариативную часть.

2.1. Наименования предшествующих дисциплин

Для изучения данной дисциплины необходимы следующие знания, умения и навыки, формируемые предшествующими дисциплинами:

2.1.1. Информатика:

Знания: основы теории информации, программирования, работы в операционной системе Windows и офисных пакетах, дискретного программирования

Умения: представлять числа в различных позиционных системах счисления и различных форматах данных, составлять алгоритмы вычислительных задач

Навыки: навыками программирования на языках высокого уровня

2.2. Наименование последующих дисциплин

Результаты освоения дисциплины используются при изучении последующих учебных дисциплин:

2.2.1. Микропроцессорные информационно-управляющие системы

Знания: Принципы построения микропроцессорных систем, шинную архитектуру, классификацию микропроцессоров и микроконтроллеров, организацию памяти микропроцессорных систем

Умения: Представлять информацию в виде, используемом в микропроцессорных системах, составлять структурные схемы микропроцессорных систем с представлением информационных потоков, составлять алгоритмы вычислительных процессов

Навыки: Навыками составления программ на языке ассемблера, вести отладку программ в интегрированных средах, чтения схем микропроцессорных систем, распределения адресного пространства между кодами программ, данных, регистров периферийных устройств

2.2.2. Теоретические основы автоматики и телемеханики

Знания: Принципы построения схем мультиплексирования, дешифрации, кодо-импульсной передачи данных

Умения: Преобразовывать последовательную информацию в параллельную и обратно с формированием бита чётности

Навыки: Элементами теории автоматов для построения схем автоматики и телемеханики

2.2.3. Теория автоматического управления

Знания: дискретные формы представления информации и сигналов и методы их обработки

Умения: вычислять погрешности обработки дискретной информации на временной сетке

Навыки: навыками численного интегрирования

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫЕ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

В результате освоения дисциплины студент должен:

№ п/п	Код и название компетенции	Ожидаемые результаты
1	ПКС-7 Способен выполнять работы на производственном участке железнодорожной электросвязи по эксплуатации, техническому обслуживанию, ремонту и модернизации устройств и элементов телекоммуникационных систем и сетей. Способен осуществлять анализ и контроль качества и безопасности технологических процессов эксплуатации, технического обслуживания, ремонта и модернизации устройств и элементов ТСС. Способен использовать нормативно-технические документы и технические средства для диагностики технического состояния телекоммуникационных систем и сетей железнодорожного транспорта; выполнять технологические операции, связанные с безопасностью и управлением движением поездов,.	ПКС-7.6 Знает и демонстрирует готовность применять в профессиональной деятельности современные технологии проектирования и монтажа электрических и оптических линий связи, методы построения аналоговых и цифровых систем передачи сигналов; методы расчета параметров передачи линий связи и параметров взаимных влияний между ними, передаточных характеристик электрических и волоконно-оптических линий связи.

4. ОБЪЕМ ДИСЦИПЛИНЫ (МОДУЛЯ) В ЗАЧЕТНЫХ ЕДИНИЦАХ И АКАДЕМИЧЕСКИХ ЧАСАХ

4.1. Общая трудоемкость дисциплины составляет:

4 зачетные единицы (144 ак. ч.).

4.2. Распределение объема учебной дисциплины на контактную работу с преподавателем и самостоятельную работу обучающихся

Вид учебной работы	Количество часов	
	Всего по учебному плану	Семестр 5
Контактная работа	68	68,15
Аудиторные занятия (всего):	68	68
В том числе:		
лекции (Л)	34	34
лабораторные работы (ЛР)(лабораторный практикум) (ЛП)	34	34
Самостоятельная работа (всего)	31	31
Экзамен (при наличии)	45	45
ОБЩАЯ трудоемкость дисциплины, часы:	144	144
ОБЩАЯ трудоемкость дисциплины, зач.ед.:	4.0	4.0
Текущий контроль успеваемости (количество и вид текущего контроля)	ПК2, ТК	ПК2, ТК
Виды промежуточной аттестации (экзамен, зачет)	Экзамен	Экзамен

4.3. Содержание дисциплины (модуля), структурированное по темам (разделам)

№ п/п	Семестр	Тема (раздел) учебной дисциплины	Виды учебной деятельности в часах/ в том числе интерактивной форме						Формы текущего контроля успеваемости и промежуточной аттестации
			Л	ЛР	ПЗ/ТП	КСР	СР	Всего	
1	2	3	4	5	6	7	8	9	10
1	5	Раздел 1 Общая схема микроконтроллеров, их возможности.	12	2			4	18	
2	5	Тема 1.1 Содержание, цель и задачи дисциплины, её роль в становлении инженера по специальности «Автоматика, телемеханика и связь», литература, основные понятия и определения, терминология.	4	2				6	ТК, Контрольные вопросы 1-4
3	5	Тема 1.2 Архитектуры процессоров: Гарвардская и Принстонская	2					2	
4	5	Тема 1.3 RISC / CISC процессоры	2					2	
5	5	Тема 1.4 DSP-процессоры	2					2	
6	5	Тема 1.5 Отличие в архитектурах процессоров Гарвардской и Принстонской; RISC / CISC процессоры.	2					2	
7	5	Раздел 2 Реализация микроконтроллеров	10	2			4	16	
8	5	Тема 2.1 Примеры реализации реальных микроконтроллеров	2					2	Контрольные вопросы 5-11
9	5	Тема 2.2 Особенности подключения питания, потребляемая мощность в различных режимах работы и при	2					2	

№ п/п	Семестр	Тема (раздел) учебной дисциплины	Виды учебной деятельности в часах/ в том числе интерактивной форме						Формы текущего контроля успеваемости и промежуточной аттестации
			Л	ЛР	ПЗ/ПП	КСР	СР	Всего	
1	2	3	4	5	6	7	8	9	10
		различной технологии выполнения однокристальных микроконтроллеров, виды тактирования.							
10	5	Тема 2.3 Регистры, АЛУ в микроконтроллерах и микропроцессорах	2					2	
11	5	Тема 2.4 Цифровые, аналоговые порты ввода/вывода в микроконтроллерах	2					2	
12	5	Тема 2.5 Микроконтроллеры PICMicro, AVR, BasicStamp	2					2	
13	5	Раздел 3 Основы языка Ассемблера. Ресурсы программиста микроконтроллеров.	2	1			8	11	
14	5	Тема 3.1 Основные ресурсы программиста - регистры общего назначения. Виды команд. Форматы данных. Арифметические и логические команды. Биты, байты, слова, двойные слова.	2					2	Контрольные вопросы 12-15
15	5	Раздел 4 Стадии создания программного обеспечения.	2	4			5	11	
16	5	Тема 4.1 Языки описания алгоритма. Кодирование алгоритма. Тестирование и отладка программы. Команды сравнения и условного перехода. Проектирование ветвящихся	2					2	ПК2, Контрольные вопросы 16-21

№ п/п	Семестр	Тема (раздел) учебной дисциплины	Виды учебной деятельности в часах/ в том числе интерактивной форме						Формы текущего контроля успеваемости и промежуточной аттестации
			Л	ЛР	ПЗ/ТП	КСР	СР	Всего	
1	2	3	4	5	6	7	8	9	10
		алгоритмов.							
17	5	Раздел 5 Организация памяти в микропроцессорных системах. Циклические алгоритмы.	4				6	10	
18	5	Тема 5.1 Организация памяти. Виды адресации. Описание и обработка массивов. Использование компилятора Ассемблера. Типовая структура ассемблерной программы.	2					2	, Контрольные вопросы 22-26
19	5	Тема 5.3 Виды, разработка и кодирование циклических алгоритмов.	2					2	
20	5	Раздел 6 Ввод/вывод данных.	2	2				4	
21	5	Тема 6.1 Взаимодействие с внешними устройствами: параллельный ввод/вывод данных, преобразование логических уровней, последовательный ввод/вывод данных. Протоколы обмена: microwire, I2C, CAN.	2					2	, Контрольные вопросы 27-35
22	5	Раздел 7 Аналоговый ввод/вывод.	2	3				5	
23	5	Тема 7.1 Применение встроенных возможностей АЦП, построение ЦАП (последовательных и параллельных).	2					2	, Контрольные вопросы 36-38
24	5	Раздел 8 Подключение объектов контроля		6				6	

№ п/п	Семестр	Тема (раздел) учебной дисциплины	Виды учебной деятельности в часах/ в том числе интерактивной форме						Формы текущего контроля успеваемости и промежуточной аттестации
			Л	ЛР	ПЗ/ТП	КСР	СР	Всего	
1	2	3	4	5	6	7	8	9	10
		и управления.							
25	5	Зачет		14			4	63	Экзамен
26		Тема 8.1 Подключение светодиодов на цифровые выходы (одиночных, семисегментных индикаторов), подключение кнопочных выключателей (методы подавление звона контактов), подключение сдвиговых регистров.							
27		Тема 8.2 Ввод с матричной клавиатуры, управление ЖКИ, управление электромагнитным реле, преобразование уровней RS-232.							
28		Всего:	34	34			31	144	

4.4. Лабораторные работы / практические занятия

Практические занятия учебным планом не предусмотрены.

Лабораторные работы предусмотрены в объеме 34 ак. ч.

№ п/п	№ семестра	Тема (раздел) учебной дисциплины	Наименование занятий	Всего часов/ из них часов в интерактивной форме
1	2	3	4	5
1	5	РАЗДЕЛ 1 Общая схема микроконтроллеров, их возможности.	Содержание, цель и задачи дисциплины, её роль в становлении инженера по специальности «Автоматика, телемеханика и связь», литература, основные понятия и определения, терминология.	2
2	5	РАЗДЕЛ 2 Реализация микроконтроллеров	Изучение микроконтроллеров – подключение питания, тактирование системы и написание программы «мигание диодом»	2
3	5	РАЗДЕЛ 3 Основы языка Ассемблера. Ресурсы программиста микроконтроллеров.	Подключение и управление семисегментными индикаторами.	1
4	5	РАЗДЕЛ 4 Стадии создания программного обеспечения.	Программирование ветвящихся алгоритмов. Опрос состояния кнопки и управление светодиодом.	4
5	5	РАЗДЕЛ 6 Ввод/вывод данных.	Подключение объектов контроля и управления по стандартным протоколам передачи данных.	2
6	5	РАЗДЕЛ 7 Аналоговый ввод/вывод.	Программирование циклических алгоритмов. Работа с аналоговыми входами. Программирование – изменение яркости свечения светодиода, по средствам ШИМ, через потенциометр, подключенный к АЦП микроконтроллера.	3
7	5	РАЗДЕЛ 8 Подключение объектов контроля и управления.	Обработка символьной информации, вывод текстовой информации на ЖКИ.	6
8	5		Зачет	14
ВСЕГО:				34/0

4.5. Примерная тематика курсовых проектов (работ)

Курсовые работы (проекты) не предусмотрены.

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В соответствии с требованиями ФГОС по направлению подготовки реализация компетентностного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий (компьютерных симуляций, демонстрация компьютерных моделей и реальных работающих устройств) в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.

Лекционная деятельность, составляющая для уровня подготовки специалистов менее 50% аудиторной работы, сопровождается демонстрацией презентационных материалов. Через проектор выводятся как слайды презентаций, представляющих суть изучаемых тем, а также демонстрируются запущенные программные среды с "живой" работой в них.

В рамках курса предусмотрены встречи с представителями ОАО "Радиус-Автоматика", НИИЭФА "Энерго", а также предприятий компании Siemens, с которой у университета заключен договор сотрудничества в сфере подготовки специалистов.

На завершающем этапе изучения дисциплины проводится традиционная научно-техническая конференция студентов данного потока, по итогам которой лучшие выступления публикуются в сборнике студенческих работ «Неделя науки» и «Безопасность движения поездов».

Удельный вес занятий, проводимых в интерактивных формах, определяется главной целью программы, особенностью контингента обучающихся, содержанием дисциплины и составляет не менее 20% аудиторной работы.

Рабочая программа данной дисциплины выставляется на сайте университета для возможности организации самостоятельной работы, в т.ч. в форме удаленного доступа (дистанционная технология).

6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

№ п/п	№ семестра	Тема (раздел) учебной дисциплины	Вид самостоятельной работы студента. Перечень учебно-методического обеспечения для самостоятельной работы	Всего часов
1	2	3	4	5
1	5	РАЗДЕЛ 1 Общая схема микроконтроллеров, их возможности.	Структура МК. Подбор литературы и конспектирование.	4
2	5	РАЗДЕЛ 2 Реализация микроконтроллеров	Типы корпусов МК: PHT, SMT, BGA, TAP, QFB, SOB, технология изготовления кристаллов. Подбор литературы и конспектирование.	4
3	5	РАЗДЕЛ 3 Основы языка Ассемблера. Ресурсы программиста микроконтроллеров.	Изучение отличий в реализации команд ассемблера, для различных типов микроконтроллеров. Подбор литературы и конспектирование, подготовка примеров и их решений.	8
4	5	РАЗДЕЛ 4 Стадии создания программного обеспечения.	Изучение различных способов программирования МК. Изучение доступных способов отладки и трассировки. Подбор литературы и конспектирование.	5
5	5	РАЗДЕЛ 5 Организация памяти в микропроцессорных системах. Циклические алгоритмы.	Способы взаимодействия с внутренней и внешней памятью, типы памяти. Подбор литературы и конспектирование.	6
6	5		Зачет	4
ВСЕГО:				31

7. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1. Основная литература

№ п/п	Наименование	Автор (ы)	Год и место издания Место доступа	Используется при изучении разделов, номера страниц
1	Intel 64 and IA-32 Architectures Software Developers Manual	Интел	Интел, 2015	Все разделы
2	Архитектура ЭВМ и систем	Новожилов О.П.	М.: Юрайт, 2012	Все разделы

7.2. Дополнительная литература

№ п/п	Наименование	Автор (ы)	Год и место издания Место доступа	Используется при изучении разделов, номера страниц
3	Персональные IBM PC и XT. Программирование на языке ассемблера	Перевод с английского И.В.Емелин	«Радио и связь», 1989	Все разделы
4	Введение в микропроцессорную технику	Ч.Гилмор	«Мир», 1984	Все разделы
5	Процессоры семейства INTEL P6, Pentium II, Pentium III, Celeron и др. Архитектура, программирование, интерфейс	И.И.Шагурин, Е.М.Бердышев	«Горячая линия – Телеком», 2000	Все разделы
6	Программирование арифметических операций в микропроцессорах: Учебное пособие для технических ВУЗов	В.К.Злобин, В.Л.Григорьев	М.:Высш. шк., 1991	Все разделы
7	Микропроцессоры: Курс и упражнения	Р. Токхайм	М.:Энергоатомиздат, 1988	Все разделы
8	Алгоритмы и структуры данных	Н. Вирт	М.: Высшая школа, 1989	Все разделы
9	Искусство программирования на Ассемблере	Н.Г.Голубь	2002	Все разделы
10	Assembler. Учебник для вузов, 2-е издание	В.И. Юров	2003	Все разделы

8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ "ИНТЕРНЕТ", НЕОБХОДИМЫЕ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1. www.intel.ru
2. www.autex.ru
3. www.avr.ru
4. <http://www.dessy.ru/>
5. <http://www.freescale.com/>

9. ПЕРЕЧЕНЬ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ,

ИСПОЛЬЗУЕМЫХ ПРИ ОСУЩЕСТВЛЕНИИ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Для лекционного курса необходимо проекционное мультимедийное оборудование с широкоформатным экраном. Установленное программное обеспечение MS PowerPoint 2010, 2013 и выше, MatLab или Electronic WorkBench, или LabView, Embarcadero RAD Studio XE2 или выше, электронная указка.

Для выполнения лабораторного курса используются:

Компьютеры дисплейного класса кафедры «Электроэнергетика транспорта»

Intel Pentium E2160-1.80/2Gb/HDD 80Gb/Video on board+PCI/DVD-RW/LAN/300Wt – 28 шт.

Лицензионное программное обеспечение

1. Microsoft Windows XP Professional (предустановлена);
2. Microsoft Windows Server 2000 R2 (программа MSDN);
3. Microsoft Office 2013 (Корпоративная лицензия МГУПС (МИИТ));
4. Embarcadero RAD Studio XE2 (Покупка за счёт средств ИТТСУ);
5. Компас3D (Trial);
6. Microsoft Visio 2013 (программа MSDN);
7. Microsoft Access 2013 (программа MSDN);
8. DeviceLock 2010 (Покупка за счёт средств кафедры);
9. Программы, поставленные совместно с лабораторным оборудованием);

Для самостоятельной работы студентам, наряду с рекомендуемой и дополнительной литературой, предлагается использовать данные и информацию следующего характера (в том числе посредством поиска в сети Интернет):

- 1) справочно-информационного (словари, справочники, энциклопедии, библиографические сборники и т.д.);
- 2) официального (сборники нормативно-правовых документов, законодательных актов и кодексов);
- 3) первоисточники (исторические документы и тексты, литература на иностранных языках);
- 4) научного и научно-популярного (монографии, статьи, диссертации, научно-реферативные журналы, сборники научных трудов, ежегодники и т.д.);
- 5) периодические издания (профессиональные газеты и журналы); и т.д.

В качестве электронных поисковых систем и баз данных публикаций рекомендуется пользоваться следующими электронными ресурсами:

- Российская Государственная Библиотека <http://www.rsl.ru>
- Научная электронная библиотека eLIBRARY.RU <http://elibrary.ru>
- Государственная публичная научно-техническая библиотека России <http://www.gpntb.ru>
- Всероссийская государственная библиотека иностранной литературы <http://www.libfl.ru>
- Институт научной информации по общественным наукам Российской академии наук (ИНИОН РАН) <http://www.inion.ru>

Для подготовки статей, докладов, эссе, рефератов и т.п. необходимо наличие MS Word 2010, 2013 и выше.

10. ОПИСАНИЕ МАТЕРИАЛЬНО ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Дисплейный класс с ПК на базе процессоров семейства Intel P6, операционной системой Windows 7 и выше, системой программирования Delphi XE20 и выше, системами MatLab/Simulink и CoDeSys.

Комплект лабораторных стендов - микроконтроллеров семейства AVR.

Плакаты со структурными схемами внутренней организации микропроцессоров и микроконтроллеров.

11. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

После создания языков программирования высокого уровня, т.е. таких языков программирования, которые несколько приближены к языку и мышлению человека (таких, как Fortran, Basic, Pascal, C++ и их современные объектно-ориентированные продолжатели Visual Basic, Delphi, C# и т.д.) постоянно задается вопрос, а зачем надо изучать язык Ассемблера, относящийся к машинным, низкоуровневым языкам? С тех же времен на этот вопрос дается традиционный ответ, который не только не утратил своей актуальности, но и может быть расширен.

На сегодняшний день промышленностью разных стран выпускается большое количество микропроцессоров и микроконтроллеров, которые представляют собой интегральные микросхемы. Микроконтроллер – это самостоятельный класс устройств, который содержит в себе, в едином корпусе, микропроцессорное ядро и набор периферийного оборудования, такого как таймер/счетчики, каналы ввода/вывода, широтно-импульсные модуляторы, аналого-цифровые и цифро-аналоговые преобразователи, сторожевой таймер, менеджер питания, тепловой сенсор и т.д., поэтому далее при разговоре про ассемблер будем употреблять слово микропроцессор, подразумевая под ним и микроконтроллер с его микропроцессорным ядром. И микропроцессор, и микроконтроллер представляют собой конечные автоматы с программным принципом управления, а значит, последовательность двоичных кодов позволяет управлять внутренним состоянием микропроцессора и микроконтроллера и всеми внутренними компонентами, заставляя их, в конечном итоге, приходиться к состоянию, необходимому в данной задаче. Последовательность двоичных кодов и есть программа для микропроцессора или программа на машинном языке. Язык ассемблера – это переведенный на псевдочеловеческий язык машинный код. Т.е. ассемблер теснейшим образом связан с внутренним устройством, архитектурой и представлением информации в микропроцессоре. Значит, если Вы знаете язык ассемблера, то Вы однозначно знаете микропроцессор, а значит, Вы можете напрямую с ним «общаться» заставляя его эффективнее решать нужные Вам задачи. По аналогии можно сказать, что если Вы знаете какой-либо иностранный язык, то Вы можете эффективнее жить в государстве, где данный иностранный язык государственный, и не пользоваться переводчиками (людьми, разговорниками, электронными переводчиками), которые не всегда точно могут передать Вашу мысль собеседнику. Хотелось бы отметить, что языков Ассемблера множество, так как множество архитектур построения микропроцессоров. Фактически, для каждого семейства микропроцессоров есть свой ассемблер.

Зная о том, что язык Ассемблера – это машинный язык, а значит, с помощью его можно эффективнее управлять микропроцессором, профессиональные программисты используют его в самых критических по времени местах программ, даже если программируют на языках высокого уровня. Например, если посмотреть модуль System в Delphi, который выполняет все базовые функции и автоматически подключается ко всем проектам, то в нем можно увидеть множество ассемблерных вставок. Это понятно, так как, если этот модуль основной (т.е. в нем содержатся основные процедуры и функции, которые чаще всего выполняются), то нужно, чтобы они быстро и эффективно обрабатывались микропроцессором, а значит, их и реализуют на ассемблере. Все основные математические функции написаны на ассемблере. Вот, например, как выглядит реализация функции вычисления экспоненты:

```
function Exp(const X: Extended): Extended;
```

```
asm
```

```
FLD X
```

```
FLDL2E
FMUL
FLD ST(0)
FRNDINT
FSUB ST(1), ST
FXCH ST(1)
F2XM1
FLD1
FADD
FSCALE
FSTP ST(1)
end;
```

Также, все, что касается работы с памятью, тоже реализуется с помощью ассемблерных вставок, так как работа с памятью происходит постоянно, то здесь необходима самая высокая скорость и эффективность. Желающие могут посмотреть реализацию функции Move в модуле System, а также все, что касается реализации менеджера памяти.

Некоторые могут заметить, что компилятор и так занимается тем, что переводит с языка высокого уровня на машинный, а значит, и так понятно, что в конечном итоге все будет приведено к машинному коду. Это так, однако, ассемблерные вставки все равно используют. Почему? Дело в том, что язык высокого уровня строится на некотором базовом наборе функциональных возможностей или кирпичиков, с помощью которых реализуются все дальнейшие программные конструкции любой сложности. Поскольку набор таких кирпичиков, хоть и увеличивается с развитием языка, но до сих пор ограничен и обладает некоторым универсализмом, позволяющим использовать их в любых ситуациях и допустимых сочетаниях. Это приводит к тому, что при компиляции для каждого кирпичика формируется универсальный ассемблерный блок, который будет пригоден для любых ситуаций, но обладает некоторой избыточностью, как плата за универсальность, а значит, менее эффективен. В современных языках высокого уровня избыточность выходного машинного кода небольшая и ее стараются постоянно уменьшить, но избавиться совсем от нее невозможно.

Мы уже отмечали, что знание ассемблера означает знание микропроцессора, а каждый микропроцессор обладает своими особенностями. Быстродействие каждого микропроцессора зависит не только от тактовой частоты и объема кеш-памяти. На сегодняшний день разработано большое количество инженерных ухищрений, которые позволяют повысить быстродействие микропроцессора. К таким ухищрениям относятся суперскалярная конвейерная архитектура, блок предсказания ветвления, несколько исполнительных блоков (целочисленные вычисления, вычисления с плавающей точкой, выполнение инструкций мультимедийного расширения, потоковых команд и т.д.), внутренняя переадресация регистров, изменение очередности выполнения микрокоманд и т.д. Ситуация усложняется выходом на рынок многоядерных микропроцессоров, где архитектура еще сложнее. Все это приводит к тому, что реализация одних и тех же алгоритмов разными наборами машинных кодов на одном микропроцессоре, так же как и реализация одних и тех же алгоритмов одними и теми же наборами команд на разных микропроцессорах одного класса при прочих равных условиях, будут выполняться с разными скоростями! Это происходит из-за того, например, что внутренняя архитектура одного микропроцессора на одном ядре позволяет выполнять параллельно несколько команд в определенном сочетании, а у другого микропроцессора не позволяет. При этом, языки высокого уровня разрабатываются универсальными для разных типов процессоров, а значит, редко используют внутренние особенности архитектуры микропроцессоров, а значит, могут создавать не очень эффективный код для любых микропроцессоров или для одного более эффективный, а для другого менее эффективный. Если же программировать

на ассемблере под конкретный микропроцессор, зная все особенности его внутреннего построения, то машинный код можно будет всегда сделать эффективнее.

Если речь идет о встраиваемых микропроцессорных системах, мобильных устройствах, промышленных компьютерах, интеллектуальных микропроцессорных датчиках, измерителях и регистраторах и т.д., то вопрос об эффективности многократно возрастает, так как все подобные системы для уменьшения стоимости, энергопотребления, габаритных показателей производятся с ограниченными объемами памяти и производительностью, но всегда функционально загруженными «под завязку», да еще и есть желание расширить его функциональный состав. Программирование таких систем, в которых необходимо выполнять математические вычисления, контролировать работу всех периферийных устройств, принимать и передавать информацию и т.д., не обходится и не может обойтись без использования ассемблера.

Некоторые могут заметить, что они не собираются быть профессиональными программистами, тем более программировать мобильные системы или промышленные микропроцессорные контроллеры, а значит, знания об ассемблере им не нужны. Да, с этим можно отчасти согласиться. Но Вас никто на профессиональных программистов не готовит. Вы обучаетесь в университете, а значит, получаете академическое инженерное образование с широким кругозором в лучших традициях советской высшей школы. На сегодняшний день человечество вступило в век информации, а подавляющее большинство информационных процессов зиждется на цифровых технологиях с применением микропроцессорной техники, и теперь уже сложно представить рабочее место человека без персонального компьютера, а самого человека без мобильного телефона. Значит, современный инженер должен понимать основы функционирования современных микропроцессорных систем, знать их лучше, чем обыватель, ибо просто работать на компьютере может уже и первоклассник. А, поскольку, функционирование микропроцессорной техники основано на программном принципе управления, поскольку есть множество принципов построения архитектур микропроцессоров и микропроцессорных систем, поскольку есть множество видов представления информации в микропроцессорных системах (а знания о представлении информации позволяет более глубоко понимать работу всех вычислительных и не только программ, таких как MS Excel, СУБД, программы обработки изображений, векторной графики, инженерных программ, например, MatLab, MathCad, все языки программирования и т.д.), то знание низкоуровневого языка программирования ассемблера, изучение которого теснейшим образом связано с обозначенными вопросами, несомненно, является обязательным!