



Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новгородский государственный университет имени Ярослава Мудрого»
МНОГОПРОФИЛЬНЫЙ КОЛЛЕДЖ
ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ
Учебно-методическая документация

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ПРАКТИЧЕСКИМ ЗАНЯТИЯМ

МДК 04.02 АРХИТЕКТУРА ЭВМ И ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

Специальность:

09.02.01 «Компьютерные системы и комплексы»

Квалификация выпускника: техник

Разработчик: Савинова Марина Александровна, преподаватель ПТК НовГУ.

Методические рекомендации приняты на заседании предметной цикловой комиссии дисциплин профессионального цикла Политехнического колледжа протокол № 1 от 05.09.2014

Председатель предметной цикловой комиссии  / Л. Н. Цымбалюк

Пояснительная записка	Ошибка! Закладка не определена.
Тематический план	6
Содержание практических занятий	11
Практическая работа №1	11
Практическая работа №2.....	14
Практическая работа №3.....	17
Практическая работа №4.....	20
Практическая работа №5.....	24
Практическая работа №6.....	28
Практическая работа №7.....	33
Практическая работа №8.....	38
Практическая работа №9.....	45
Практическая работа №10.....	47
Практическая работа №11.....	51
Практическая работа №12.....	53
Практическая работа №13.....	59
Практическая работа №14.....	63
Практическая работа №15.....	68
Практическая работа №17.....	78
Практическая работа №18.....	89
Практическая работа №19.....	101
Практическая работа №20.....	104
Практическая работа №21.....	109
Практическая работа №22.....	114
Практическая работа №23.....	120
Практическая работа №24.....	125
Практическая работа №25.....	131
Информационное обеспечение обучения.....	134
Лист регистрации изменений	135

Пояснительная записка

Методические рекомендации по практическим занятиям, являющиеся частью учебно-методического комплекса по дисциплине «Архитектура ЭВМ и вычислительных систем» составлены в соответствии с:

1 Федеральным государственным образовательным стандартом по специальности 09.02.01 «Компьютерные системы и комплексы»;

2 Рабочей программой учебной дисциплины;

3 Положением о планировании, организации и проведении лабораторных работ и практических занятий студентов, осваивающих основные профессиональные образовательные программы среднего профессионального образования в колледжах НовГУ.

Методические рекомендации по выполнению практических работ содержат задания для изучения тем разделов «Базовые понятия и основные принципы построения архитектур вычислительных систем», «Архитектура и структура вычислительных машин и систем», . В них рассматриваются приемы работы с текстовыми редакторами, выполнение расчетных задач в электронных таблицах, работа с графической информацией, обработкой аудио и видеоинформации.

Методические рекомендации включают 30 практических занятий, предусмотренных рабочей программой учебной дисциплины в объеме 60 часов.

В результате выполнения практических заданий обучающийся должен:

уметь:

- получать информацию о параметрах компьютерной системы;
- подключать дополнительное оборудование и настраивать связь между элементами компьютерной системы;
- производить установку и настройку программного обеспечения компьютерных систем.

знать:

- базовые понятия и основные принципы построения архитектур вычислительных систем;
- типы вычислительных систем и их архитектурные особенности;
- организацию и принцип работы основных логических блоков компьютерных систем;
- процессы обработки информации на всех уровнях компьютерных архитектур;
- основные компоненты программного обеспечения компьютерных систем;
- основные принципы управления ресурсами и организации доступа к этим ресурсам;

В практических работах предлагаемые задания носят репродуктивный, частично-поисковый характер и поисковый характер.

Основные теоретические положения даются студентам на лекционных занятиях. Перед проведением практического занятия проводится инструктаж, выдается индивидуальное задание для самостоятельного выполнения. Во время проведения практической работы решение индивидуального задания проверяется, корректируется, проводится анализ решения, подводятся итоги.

Степень овладения студентами запланированных умений оценивается во время защиты практической работы. Критерием оценки практических работ является качество выполненных заданий, правильность ответов во время защиты.

Оценка «5» ставится в случае, когда результатом работы является полностью выполненная работа, даны ответы на поставленные контрольные вопросы.

Оценка «4» ставится, когда итогом работы является правильно выполненная практическая работа, но могут быть небольшие неточности при выполнении.

Оценка «3» - не выполнено до конца одно из заданий, даны ответы не на все контрольные вопросы.

Оценка «2» ставится в случае неподготовленности студента к защите практической работы.

Возможным вариантом защиты практических работ, может быть защита электронного отчета, структура которого содержит титульный лист, цели выполнения практической работы, краткое описание работы и выводы.

2.2. Тематический план и содержание учебной дисциплины «Архитектура ЭВМ и вычислительных систем»

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся		Объем часов	Уровень усвоения	
1	2		3	4	
Раздел 1. Базовые понятия и основные принципы построения архитектур вычислительных систем			56		
Тема 1.1. Базовые понятия	Содержание учебного материала		3	1,2	
	1.	Введение в предмет. Понятия ЭВМ и ВС. Понятие архитектуры ВС			
	2.	Эволюция вычислительной техники. Теория эволюции компьютеров. Закон Мура. Принципы фон Неймана. Поколения ЭВМ. Системы eniac, edvac.			
	3.	Информация, кодирование, обработка в ЭВМ. Системы счисления. Правила десятичной арифметики. Дополнительный код числа. Числа с фиксированной и плавающей точкой	2		
	Практические занятия				
	№ 1. Изучение принципов представления информации и принципов выполнения операций в двоичной ССЧ с использованием дополнительного кода и обратного кода.				
	№ 2. Изучение принципов представления информации и принципов выполнения операций в восьмеричной и шестнадцатеричной ССЧ		2		
	Контрольные работы: Входной контроль – тестирование на владение материалом. Двоичное кодирование информации. Операции в двоичной системе.		2		
	Самостоятельная работа обучающихся		4		
	№ 1. Базовые понятия и основные принципы построения архитектур вычислительных систем				
	№ 2. Сложение и вычитание чисел с использованием прямого, дополнительного и обратного кода по индивидуальному заданию				
	№ 3. Арифметические операции в восьмеричной и шестнадцатеричной ССЧ по индивидуальному заданию				
Тема 1.2. Организация и принцип работы основных логических блоков компьютерных систем;	Содержание учебного материала		3		1,2,3
	1.	Логические основы ЭВМ, элементы и узлы. Алгоритмы и программы			
	2.	Изучение работы сумматоров различных типов.			
	3.	Изучение шифраторов и принципов их работы. Изучение дешифраторов и принципов их работы	2		
	Практические занятия				
	№ 3. Комбинации логических элементов. Булева алгебра. Уровни логических сигналов. Поиск не-				

	исправностей	2		
	№ 4. Логические элементы "И-НЕ" (NAND), "ИЛИ-НЕ" (NOR). Эквивалентные схемы.			
	№ 5. Элемент "ИСКЛЮЧАЮЩЕЕ ИЛИ" (XOR). Цифровой компаратор.			
	№ 6. Элемент "ИСКЛЮЧАЮЩЕЕ ИЛИ" (XOR).Поиск неисправностей.			2
	№ 7. Изучение работы RS-, JK- триггеров и принципов их работы. Асинхронные счетчики.			4
	№ 8. D-триггеры. Цепи синхронизации.			4
	№ 9. D-триггеры. Поиск неисправностей.			4
	№ 10.. Таймер. Двоичный счетчик с последовательным переносом			4
	№ 11. Цифро-аналоговый преобразователь. Аналогово-цифровой преобразователь.			4
	№ 12. Дешифраторы и мультиплексоры. Демультимплексор.			2
	№ 13. Арифметико-логическое устройство (АЛУ)			2
		2		
	Самостоятельная работа обучающихся		5	
№ 4. ПРОЕКТ «Таблица истинности и логическая схема по индивидуальному заданию»				
№ 5. Подготовка докладов по разделу: Базовые понятия и основные принципы построения архитектур вычислительных систем				
№ 6. Спроектировать следующие устройства: 1. «Одноразрядный сумматор на 2 входа. Таблица истинности Описать принцип работы». 2. «Одноразрядный сумматор на 3 входа. Таблица истинности. Описать принцип работы» 3. «Сумматор последовательного действия. Таблица истинности Описать принцип работы»				
№ 7. Проект «Построить шифратор на элементах ИЛИ-НЕ. Таблица истинности. Описать принцип работы.				
Тема 1.3. Основные принципы построения архитектур вычислительных систем	Содержание учебного материала		2	1
	1.	Архитектура системы команд. CISC и RISC архитектуры процессоров		
	Самостоятельная работа обучающихся		1	2
№ 8. Графическое изображение структуры текста: Развитие архитектур современных МП				
Раздел 2. Архитектура и структура вычислительных машин и систем		49		
Тема 2.1. Принципы технической реализации модели коллектива вычислителей	Содержание учебного материала		3	1
	1.	Принципы построения вычислительных систем (ВС). Архитектурные свойства ВС. Способы повышения производительности ЭВМ при обработке информации. Системы параллельного программирования.		
	2.	Способы классификации ВС. Многомашинные и многопроцессорные ВС. Уровни и средства комплексирования.		
	3.	Классификация вычислительных систем. Параллельные алгоритмы		
	Самостоятельная работа обучающихся		4	
№ 9. Подготовка презентаций по темам: «Способы повышения производительности ЭВМ при обработке информации» «Языки параллельного программирования». «Многомашинные и многопроцессорные ВС»				

Тема 2.2. Процессы обработки информации на всех уровнях компьютерных архитектур	№ 10. Подготовка доклада «Параллельные алгоритмы»			
	Содержание учебного материала			
	1.	Алгоритмы маршрутизации. Методы передачи данных. Передача данных между двумя процессорами и широковещательная передача Латентность и пропускная способность сети.	10	1,2,3
	2.	Организация памяти вычислительных систем		
	3.	Основные принципы управления ресурсами и организации доступа к этим ресурсам.		
	4.	Составление программ системного программирования на языке Ассемблер		
	5.	Основные компоненты программного обеспечения компьютерных систем		
	Практические занятия			
	№ 14. Изучение регистров и принципов их работы		2	
	№ 15. Изучение принципов работы микрокомпьютера на примере 8051: (Управляющие подпрограммы; Команды передачи данных; Порты ввода-вывода; Арифметические и логические операции; Операции передачи управления)		2	
	№ 16. Изучение принципов работы микрокомпьютера на примере 8051: (Внешние прерывания; Таймеры и счетчики событий; Поиск неисправностей – тестирование; Использование клавиатуры и дисплея; Последовательная передача)		2	
	№ 17. Изучение микроконтроллера Intel 8051 и принципов его работы: (Сигналы микрокомпьютера; Программируемый таймер; Периферийный интерфейс; Сигналы аналоговых устройств)		2	
	№ 18. Изучение микроконтроллера Intel 8051 и принципов его работы: Интерфейс связи; Объединение периферийных устройств		2	
	№ 19. Изучение процессов обработки информации на всех уровнях компьютерной архитектуры RISC на примере 32-разрядного - процессора 80960SA (Среда Отладчика; Загрузка и отладка; Команды перемещения данных);		2	
	№ 20. Изучение процессов обработки информации на всех уровнях компьютерной архитектуры RISC на примере 32-разрядного - процессора 80960SA Порт 8255 и команды ввода-вывода (I/O);		2	
	№ 21. Изучение процессов обработки информации на всех уровнях компьютерной архитектуры RISC на примере 32-разрядного - процессора 80960SA (Арифметические команды; Логические команды и команды ветвления; Основы подпрограмм; Основы прерываний)		2	
	№ 22. Изучение процессов обработки информации на всех уровнях компьютерной архитектуры RISC на примере 32-разрядного - процессора 80960SA (Арифметические команды; Последовательная передача данных);		2	
	№ 23. Изучение процессов обработки информации на всех уровнях компьютерной архитектуры RISC на примере 32-разрядного - процессора 80960SA, Программирование UART;		2	
	№ 24. Изучение процессов обработки информации на всех уровнях компьютерной архитектуры RISC на примере 32-разрядного - процессора 80960SA Цифро-аналоговое и аналого-цифровое преобразование (ЦАП и АЦП))		2	
	№ 25. Изучение конфигурации компьютера, аппаратного мониторинга с помощью утилит AIDA 32 и стандартных утилит.		2	

	Самостоятельная работа обучающихся		8	
	№ 11. Решение задач по теме: «Команды ассемблера»			
	№ 12. Установка программной утилиты AIDA 32 . Получение информации о параметрах домашней компьютерной системы с помощью утилит AIDA 32 и стандартных утилит.			
	№ 13. Подготовка реферата «Настройка компьютерного обеспечения компьютерных систем»			
Раздел 3. Типы вычислительных систем и их архитектурные особенности			21	
Тема 3.1. Векторные и векторно-конвейерные вычислительные системы	Содержание учебного материала		3	1,2,3
	1.	Понятие вектора и размещение данных в памяти. Понятие векторного процессора. Структура векторного процессора. Обработка длинных векторов и матриц.		
	2.	Векторно-конвейерные вычислительные системы STAR-100, RAY Y-MP C90		
	Самостоятельная работа обучающихся		2	
	№ 14. Подготовка презентации «PVP-архитектура»			
Тема 3.2. Матричные вычислительные системы	Содержание учебного материала		4	1,2,3
	1.	Обобщенная модель матричной ВС. Интерфейсная ВМ. Контроллер массива процессоров. Массив процессоров. Структура процессорного элемента. Подключение и отключение процессорных элементов. Сети взаимосвязей процессорных элементов.		
	2.	Ассоциативная память. Ассоциативные ВС. Систематические структуры		
	Самостоятельная работа обучающихся		3	
	№15. Подготовка презентаций «BC ILLIAC IV», «Ассоциативные ВС», «Систематические структуры»			
	№ 16. Подготовка докладов по темам: «Векторно-конвейерные вычислительные системы» «Ассоциативная память» «Систематические структуры»			
	№ 17. Подготовка рефератов: «Основные компоненты программного обеспечения компьютерных систем»; «Основные принципы управления ресурсами ВС» «Организации доступа к ресурсам ВС.»			
Тема 3.3. Кластерные и MPP-системы	Содержание учебного материала		4	1,2
	1.	Топологии кластеров. Кластер Beowulf. Кластер AC3 Velocity Cluster. Кластер NCSA NT Supercluster. Кластер Thunder		
	2.	Системы с массовым параллелизмом (MPP-системы). CRAY T3D		
	Контрольная работа: Итоговая контрольная работа		2	
	Самостоятельная работа обучающихся		3	
	№ 18. Подготовка презентации «MPP-системы», «Кластерные ВС»			

	Всего	126	
--	--------------	------------	--

Для характеристики уровня освоения учебного материала используются следующие обозначения:

1. – **ознакомительный** (узнавание ранее изученных объектов, свойств);
2. – **репродуктивный** (выполнение деятельности по образцу, инструкции или под руководством)
3. – **продуктивный** (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

Содержание практических занятий

Раздел 1. Базовые понятия и основные принципы построения архитектур вычислительных систем

Тема 1.1. Базовые понятия

Практическая работа №1.

Изучение принципов представления информации и принципов выполнения операций в двоичной ССЧ с использованием дополнительного кода и обратного кода.

Объём учебного времени – 2ч

Методические рекомендации

1. Цель работы

- изучить принципы представления информации в двоичной системе счисления.
- изучить принципы выполнения операций сложения в двоичной системе счисления.

2. Перечень необходимых средств обучения:

2.1 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.2 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007

3. Практические указания:

Двоичная система счисления имеет набор цифр {0, 1}, $p=2$. В общем виде, используя формулу (1), двоичное число можно представить выражением:

$$A_{(2)} = a_n \cdot 2^n + \dots + a_1 \cdot 2^1 + a_0 \cdot 2^0 \quad (3)$$

Например, число 101101(2) можно записать так:

$$101101(2) = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

Двоичная система счисления имеет особую значимость в информатике: внутреннее представление любой информации в компьютере является двоичным, т.е. описывается набором символов только из двух знаков 0 и 1.

Шестнадцатеричная система счисления имеет набор цифр {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}, $p = 16$. Для изображения чисел в шестнадцатеричной системе счисления требуются 16 цифр. Для обозначения первых десяти цифр используются цифры десятичной системы счисления, шесть остальных — первых шесть прописных букв латинского алфавита. По формуле (1) шестнадцатеричное число может быть представлено так:

$$A_{(16)} = a_n \cdot 16^n + \dots + a_1 \cdot 16^1 + a_0 \cdot 16^0 \quad (4)$$

Пример

1. Число E7F8140 по формуле (4) запишется так:

$$E \cdot 16^6 + 7 \cdot 16^5 + F \cdot 16^4 + 8 \cdot 16^3 + 1 \cdot 16^2 + 4 \cdot 16^1 + 0 \cdot 16^0$$

Представление информации, хранящейся в памяти компьютера, в ее истинном двоичном виде весьма громоздко из-за большого количества цифр. Поэтому при записи такой информации на бумаге или выводе ее на экран принято использовать восьмеричную или шестнадцатеричную системы счисления. В современных компьютерах чаще используется шестнадцатеричная система счисления.

Полезно помнить некоторые степени двойки и шестнадцати.

k	1	2	3	4	5	6	7	8	9	10	11	12
2 ^k	2	4	8	16	32	64	128	256	512	1024	2048	4096
16 ^k	16	256	4096	65536	1048576							

В вычислительной технике наиболее часто выполняется операция сложения. Пусть заданы два целых положительных числа в позиционной системе счисления с основанием p . Запишем эти числа в виде:

$$A = a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p + a_0 \quad 5)$$

$$B = b_n p^n + b_{n-1} p^{n-1} + \dots + b_1 p + b_0 \quad 6)$$

Сумма этих чисел равна числу, которое может быть записано в аналогичном виде:

$$S = s_n p^n + s_{n-1} p^{n-1} + \dots + s_1 p + s_0 \quad 7)$$

Вычисления выполняются по следующим правилам:

- операция сложения выполняется поразрядно, начиная с младших разрядов в слагаемых;
- в каждом одноименном разряде слагаемых суммируются соответствующие цифры и перенос из предыдущего разряда суммы;
- если сумма цифр одноименных разрядов слагаемых и переноса меньше основания системы счисления, то перенос в следующий разряд равен нулю, если равна или больше — то равен единице.

Рассмотрим правила выполнения арифметических операций над однозначными числами. Представим их в виде таблиц.

Правила сложения	Правила вычитания	Правила умножения
$0 + 0 = 0$	$0 - 0 = 0$	$0 * 0 = 0$
$0 + 1 = 1$	$0 - 1 = -1$	$1 * 0 = 0$
$1 + 0 = 1$	$1 - 0 = 1$	$0 * 1 = 0$
$1 + 1 = 10$	$1 - 1 = 0$	$1 * 1 = 1$

Пример

Сложить два числа: $1010(2) + 10101(2) = 11111(2)$

4 Контрольные вопросы.

1. Опишите правила записи чисел в десятичной системе счисления: какие символы образуют алфавит десятичной системы счисления?
2. Что является основанием десятичной системы счисления?
3. Как изменяется вес символа в записи числа в зависимости от занимаемой позиции?

4. Какие числа можно использовать в качестве основания системы счисления?
5. Какие системы счисления применяются в компьютере для представления информации?
6. Охарактеризуйте двоичную систему счисления: алфавит, основание системы счисления, запись числа.
7. Почему двоичная система счисления используется в информатике?
8. По каким правилам выполняется сложение двух положительных целых чисел?
9. Каковы правила выполнения арифметических операций в двоичной системе счисления?
10. Для чего используется перевод чисел из одной системы счисления в другую?
11. Сформулируйте правила перевода чисел из системы счисления с основанием p в десятичную систему счисления и обратного перевода: из десятичной системы счисления в систему счисления с основанием S . Приведите примеры

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Письменные ответы на контрольные вопросы

6. Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб. пособие для сред. проф. образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб. пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

1. <http://novtex.ru/IT> - журнал "Информационные технологии"
2. <http://infojournal.ru> - журнал «Информатика и образование»
3. <http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №2.

Изучение принципов представления информации и принципов выполнения операций в восьмеричной и шестнадцатеричной ССЧ

Объём учебного времени – 2ч

Методические рекомендации

1. Цель работы

- изучить принципы представления информации в восьмеричной и шестнадцатеричной системе счисления.

2. Перечень необходимых средств обучения:

2.3 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.4 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007

3. Практические указания:

В восьмеричной системе счисления числа записываются с помощью восьми цифр: 0, 1, 2, 3, 4, 5, 6, 7. Сама восьмерка (как и основание любой системы счисления) записывается двумя цифрами: 10 (почему?). Для записи чисел в шестнадцатеричной системе счисления необходимо располагать уже шестнадцатью различными символами, используемыми как цифры. В качестве первых десяти шестнадцатеричных цифр используются цифры десятичной системы счисления. Для обозначения остальных шести цифр (в десятичной системе они соответствуют числам 10, 11, 12, 13, 14, 15) используют буквы латинского алфавита — А, В, С, D, E, F. Особая привязанность программистов к восьмеричной и шестнадцатеричной системам счисления объясняется тем, что переход к записи числа в любой из этих систем от его двоичной записи очень прост, так как 8 и 16 являются степенями числа 2.

Полезно привести таблицу:

Основные системы счисления			
10	2	8	16
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8

9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

При переводе в восьмеричную систему счисления двоичное число из трех и менее цифр записывается одной восьмеричной цифрой.

При переводе в шестнадцатеричную систему счисления двоичное число из четырех и менее цифр записывается одной шестнадцатеричной цифрой.

Например:

1. Перевести числа из двоичной системы счисления в восьмеричную и шестнадцатеричную: 11100011; 101110,1; 111111,01001.
2. Перевести числа из восьмеричной системы счисления в двоичную: 567; 204,72. Перевести числа из шестнадцатеричной системы счисления в двоичную: 4D2A; 31,05; 40A,7.

4 Контрольные вопросы.

1. Какой основной недостаток двоичных чисел?
2. Каким образом производится переход из восьмеричной системы счисления в двоичную?
3. Каким образом производится переход из десятичной системы счисления в восьмеричную?
4. Каким образом производится переход из шестнадцатеричной системы счисления в двоичную?
5. Каким образом производится переход из шестнадцатеричной системы счисления в восьмеричную?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб. пособие для сред. проф. образования. - М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.

- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Тема 1.2. Организация и принцип работы основных логических блоков компьютерных систем

Практическая работа №3.

Комбинации логических элементов. Булева алгебра. Уровни логических сигналов. Поиск неисправностей

Объём учебного времени – 2ч

Методические рекомендации

1. Цель работы

- изучить основные логические функции и принципы работы логических элементов.

2. Перечень необходимых средств обучения:

2.5 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.6 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007

3. Практические указания:

История логики насчитывает около двух с половиной тысячелетий. Она берет начало от формальной логики Аристотеля. Дальнейшее развитие логики связано с именами Готфрида Вильгельма Лейбница и Джорджа Буля, которые разработали математический аппарат алгебры логики. Именно поэтому алгебру высказываний и называют булевой алгеброй. Математическая логика в XIX—XX вв. рассматривалась как базис для логического обоснования в различных областях математики, но в последние десятилетия математическая логика находит применение во многих областях, в частности, в кибернетике, теории ЭВМ, теории алгоритмов.

Основным понятием математической логики является понятие высказывания. Высказывание — предложение, про которое всегда можно сказать, истинно оно или ложно. Высказывания бывают простые и сложные. Сложное высказывание состоит из простых, соединенных знаками логических операций. Простые высказывания обычно обозначают большими латинскими буквами: A, B, C, — и т.д.

Рассмотрим следующие логические операции:

не(отрицание),

или(конъюнкция),

и(дизъюнкция),

→(следование).

Логические операции определяются через таблицы истинности.

Пусть имеются два высказывания: A — юноша в школе; B — юноша на уроке. Составим таблицы, которые показывают смысл операций и, или, не.

Операция “и”		
A	B	A и B
И	И	И
И	Л	Л
Л	И	Л
Л	Л	Л

Операция “или”		
A	B	A и B
И	И	И
И	Л	И
Л	И	И
Л	Л	Л

Операция “не”	
A	не A
И	Л
Л	И

Пусть имеются два высказывания: A — у человека высокая температура, B — человек болен. Составим таблицу, которая показывает смысл операции следования.

A	B	Из A следует B
И	И	И
И	Л	Л
Л	И	И
Л	Л	И

Решение задач.
Дана таблица:

Формула	Высказывание	Тигр	Волк	Бурундук	Заяц
A	Зверь полосатый				
B	Зверь хищный				
не A					
не B					
A и B					
A или B					

Ответ:

Формула	Высказывание	Тигр	Волк	Бурундук	Заяц
A	Зверь полосатый	И	Л	И	Л
B	Зверь хищный	И	И	Л	Л
не A	Зверь не полосатый	Л	И	Л	И
не B	Зверь не хищный	Л	Л	И	И
A и B	Зверь полосатый и хищный	И	Л	Л	Л
A или B	Зверь полосатый или хищный	И	И	И	И

4 Контрольные вопросы.

1. Операция конъюнкции. Таблица истинности.
2. Операция дизъюнкции. Таблица истинности.
3. Операция отрицания. Таблица истинности.

5. Структура отчёта по практической работе

- Титульный лист.

- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №4.

Логические элементы "И-НЕ" (NAND), "ИЛИ-НЕ" (NOR). Эквивалентные схемы
Объём учебного времени – 2ч

Методические рекомендации

1.Цель работы

- изучить принципы представления информации в ЭВМ.
- изучить принципы построения логических элементов.

2.Перечень необходимых средств обучения:

2.7 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.8 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007

3. Практические указания:

Посмотрим на микросхему. (Годом рождения полупроводниковой микросхемы, принято считать, 1959 год. Авторами изобретения, радикально изменившего образ жизни человечества, стали американские инженеры Джек Килби (Jack Kilby), работавший в то время в компании Texas Instruments, и будущий основатель корпорации Intel Роберт Нойс (Robert Noyce).

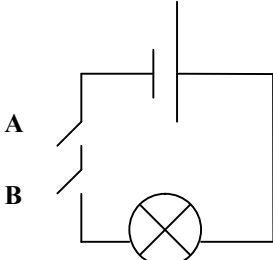
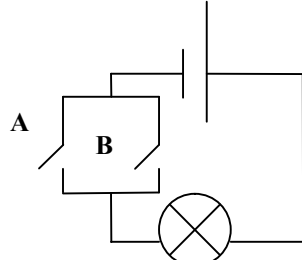
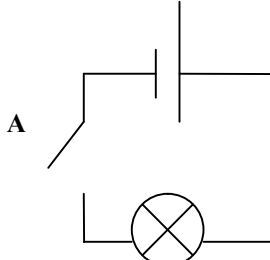
Дорогу к созданию микросхемы проложило изобретение полупроводникового транзистора в 1947 году работавшими в Bell Laboratories американскими учеными Джоном Бардином, Уильямом Шокли и Уолтером Браттейном.

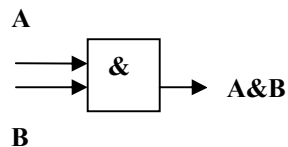
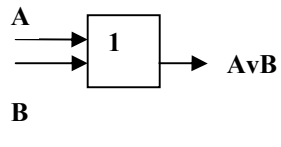
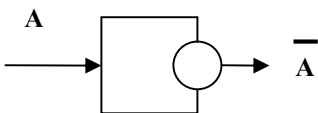
До этого большинство электронных устройств строилось на базе громоздких электронных ламп, потреблявших огромное количество электроэнергии. Транзистор позволял усиливать электрические сигналы и стал удобной, дешевой и эффективной заменой вакуумным приборам.

На первый взгляд ничего того, что нас удивило бы, мы не видим. Но если рассматривать её при сильном увеличении она поразит нас своей стройной архитектурой. Хотя компьютерные микросхемы выглядят абсолютно плоскими керамическими пластинками, (кремний является самым распространенным на земле элементом после кислорода и используется в производстве микросхем, поскольку он является естественным полупроводником.) на них может находиться более 20 уровней сложных электрических цепей и транзисторов.

Что бы понять, как она работает, вспомним, что компьютер работает на электричестве, т.е. любая информация представлена в виде электрических импульсов. Поговорим о них. С точки зрения логики электрический ток либо течёт, либо не течёт; электрический импульс есть или его нет; электрическое напряжение есть или его нет. В связи с этим поговорим о различных вариантах управления включением и выключением обыкновенной лампочки. Для этого рассмотрим электрические контактные схемы, реализующие логические операции.

Таблица 1.

Конъюнкция			Дизъюнкция			Инверсия	
							
A	B	Результат	A	B	Результат	A	Результат
1	1	1	1	1	1	1	0
1	0	0	1	0	1	0	1
0	1	0	0	1	1		
0	0	0	0	0	0		

Конъюнктор		Дизъюнктор		Инвертор	
					

1 схема (составляем в основной таблице таблицу истинности).

- 1) Оба контакта в положении «включено». Тогда ток через лампочку идёт и она горит.
- 2) Первый контакт в положении «включено», второй – в положении «выключено». Ток не идёт, лампочка не горит.
- 3) Оба контакта в положении «выключено». Тока нет. Лампочка не горит.

Вывод: первая схема реализует логическую операцию «И».

2 схема (составляем в основной таблице таблицу истинности).

- 1) Оба контакта в положении «включено». Ток через лампочку идёт и она горит.
- 2) Первый контакт в положении «включено», второй – а положении «выключено». Ток идёт, лампочка горит.
- 3) Обратная ситуация. Первый контакт в положении «выключено», второй – в положении «включено». Ток идёт, лампочка горит.
- 4) Оба контакта в положении «выключено». Тока нет. Лампочка не горит.

Вывод: вторая схема реализует логическую операцию «ИЛИ».

3 схема (составляем в основной таблице таблицу истинности).

В этом устройстве в качестве переключателя используется автоматический ключ. Когда тока в нём нет, пластинка замыкает контакты и лампочка горит. Если на ключ подать напряжение, то вследствие явления электромагнитной индукции пластинка прижимается и цепь размыкается. Лампочка не горит.

Вывод: третья схема реализует логическую операцию «НЕ».

Недостатками контактных схем являлись их низкая надёжность и быстродействие, большие размеры и потребление энергии. Поэтому попытка использовать такие схемы в ЭВМ не оправдала себя. Появление вакуумных и полупроводниковых приборов позволило создавать логические элементы с быстродействием от 1 миллиона переключений в секунду. Именно такие электронные схемы нашли своё применение в качестве элементной базы ЭВМ. Вся теория, изложенная для контактных схем, была перенесена на электронные схемы. Элементы, реализующие базовые логические операции, называли базовыми логическими элементами или **вентильями** и характеризуются они не состоянием контактов, а наличием сигналов на

входе и выходе элемента. Их названия и условные обозначения являются стандартными и используются при составлении и описании логических схем компьютера.

Почему необходимо уметь строить логические схемы?

Дело в том, что из вентилях составляют более сложные схемы, которые позволяют выполнять арифметические операции и хранить информацию. Причем схему, выполняющую определенные функции, можно построить из различных по сочетанию и количеству вентилях. Поэтому значение формального представления логической схемы чрезвычайно велико. Оно необходимо для того, чтобы разработчик имел возможность выбрать наиболее подходящий ему вариант построения схемы из вентилях. Процесс разработки общей логической схемы устройства (в том числе и компьютера в целом), становится иерархическим, причем на каждом следующем уровне в качестве "кирпичиков" используются логические схемы, созданные на предыдущем этапе.

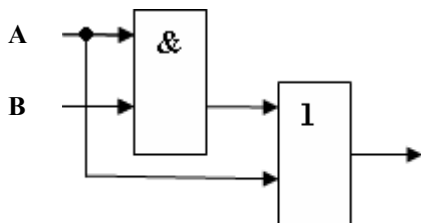
Алгоритм построения логических схем.

- 1) Определить число логических переменных.
- 2) Определить количество базовых логических операций и их порядок.
- 3) Изобразить для каждой логической операции соответствующий ей вентиль.
- 4) Соединить вентили в порядке выполнения логических операций.

Пример 1.

Составить логическую схему для логического выражения: $F = A \vee B \& A$.

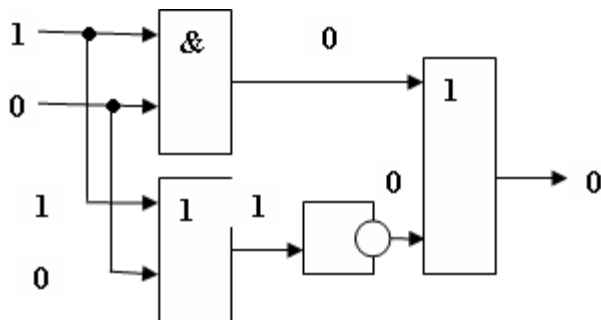
- 1) Две переменные – А и В.
- 2) Две логические операции: 1-&, 2- $\vee$.
- 3) Строим схему:



Пример 2.

Постройте логическую схему, соответствующую логическому выражению $F = A \& B \vee \neg(B \vee A)$. Вычислить значения выражения для $A=1, B=0$.

- 1) Переменных две: А и В; 1 4 3 2
- 2) Логических операций три: & и две $\vee$; $A \& B \vee \neg(B \vee A)$.
- 3) Схему строим слева направо в соответствии с порядком логических операций:



4) Вычислим значение выражения: $F = 1 \& 0 \vee \neg(0 \vee 1) = 0$

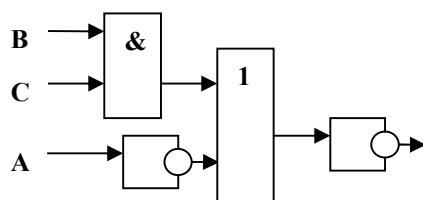
4 Контрольные вопросы.

1. Построить логические схемы по формулам:

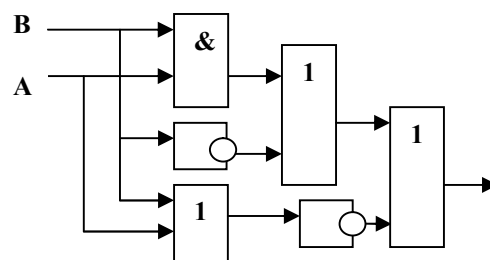
- $F = A \vee B \& C$, если $A=1, B=0, C=1$;
- $F = (A \vee B) \& (C \vee B)$, если $A=0, B=1, C=0$;
- $F = (A \& B \& C)$, если $A=0, B=0, C=1$.

2. Составить логические выражения по схемам:

а)



б)



5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Письменные ответы на контрольные вопросы

6. Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб. пособие для сред. проф. образования. - М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. - 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. - М.: Форум: ИНФРА-М, 2012. - 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб. пособие для сред. проф. образования. - Ростов-н/Д: Феникс, 2007. - 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. - СПб.: Питер, 2007. - 848 с.

Интернет-ресурсы:

1. <http://novtex.ru/IT> - журнал "Информационные технологии"
2. <http://infojournal.ru> - журнал «Информатика и образование»
3. <http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №5.

Элемент "исключающее или" (XOR). Цифровой компаратор.

Объём учебного времени – 2ч

Методические рекомендации

1.Цель работы

- изучить принципы представления информации в ЭВМ.
- изучить принципы построения логических элементов.

2.Перечень необходимых средств обучения:

2.9 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.10 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007

3. Практические указания:

Элементы Искключающее ИЛИ (по-английски — Exclusive-OR) также можно было бы отнести к простейшим элементам, но функция, выполняемая ими, несколько сложнее, чем в случае элемента И или элемента ИЛИ. Все входы элементов Искключающее ИЛИ равноправны, однако ни один из входов не может заблокировать другие входы, установив выходной сигнал в уровень единицы или нуля.

Обозначения элементов

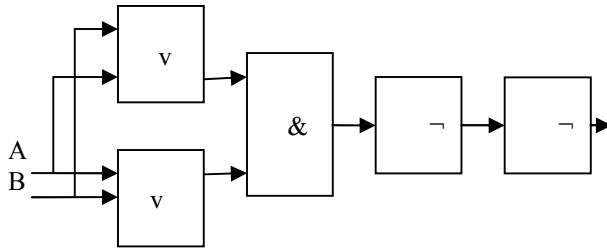


Исключающее ИЛИ: зарубежные (слева) и отечественные (справа)

Под функцией «Исключающее ИЛИ» понимается следующее: единица на выходе появляется тогда, когда только на одном входе присутствует единица. Если единиц на входах две или больше, или если на всех входах нули, то на выходе будет нуль. Таблица истинности двухвходового элемента Исключающее ИЛИ приведена в табл. 4.1. Обозначения, принятые в отечественных и зарубежных схемах, показаны на рис. 4.1. Надпись на отечественном обозначении элемента Исключающее ИЛИ "=1" как раз и обозначает, что выделяется ситуация, когда на входах одна и только одна единица.

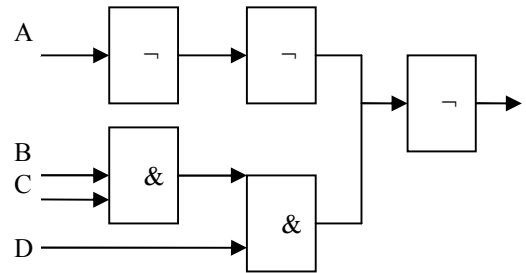
Вариант 1

1. Составьте логическую схему к выражению: $F = (A \vee B) \& \neg (A \& B)$.
2. Составьте логическое выражение по схеме:



Вариант 1

1. Составьте логическую схему к выражению: $F = \neg (A \vee B) \& (\neg A \& \neg B)$.
2. Составьте логическое выражение по схеме:



Если логическое выражение содержит большое число операций, то составлять для него таблицу истинности достаточно сложно, так как приходится перебирать большое количество вариантов. В таких случаях формулы удобно привести к нормальной форме.

Формула имеет нормальную форму, если в ней отсутствуют знаки эквивалентности, импликации, двойного отрицания, при этом знаки отрицания находятся только при логических переменных.

Для приведения формулы к нормальной форме используют законы логики и правила логических преобразований.

Пример 1

Упростите логическое выражение $F = \neg((A \vee B) \rightarrow \neg(B \vee C))$.

Это логическое выражение необходимо привести к нормальной форме, т.к. в нем присутствует импликация и отрицание логической операции.

1. Избавимся от импликации и отрицания. Воспользуемся (9). Получится:

$$(A \vee B) \rightarrow (B \vee C) = (A \vee B) \& (B \vee C).$$

2. Применим закон двойного отрицания (4). Получим:

$$(A \vee B) \& (B \vee C) = (A \vee B) \& (B \vee C).$$

3. Применим правило дистрибутивности (16). Получим:

$$(A \vee B) \& (B \vee C) = (A \vee B) \& B \vee (A \vee B) \& C$$

4. Применим закон коммутативности (18) и дистрибутивности (16). Получим:

$$(A \vee B) \& B \vee (A \vee B) \& C = A \& B \vee B \& A \& C \vee B \& C.$$

5. Применим (7) и получим:

$$A \& B \vee B \& A \& C \vee B \& C = A \& B \vee A \& C \vee B \& C$$

6. Применим (16), т.е. вынесем за скобки B. Получим:

$$A \& B \vee A \& C \vee B \& C = B \& (A \vee 1) \vee A \& C \vee B \& C.$$

7. Применим (6). Получим:

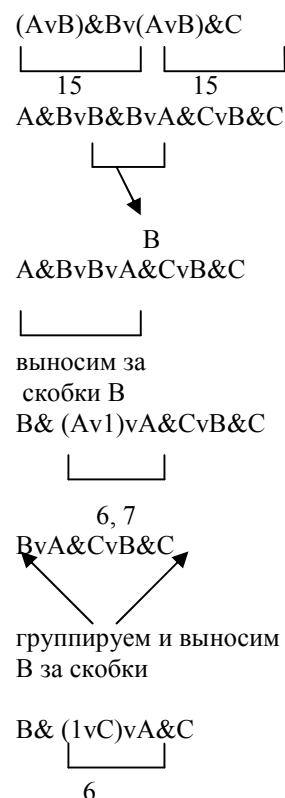
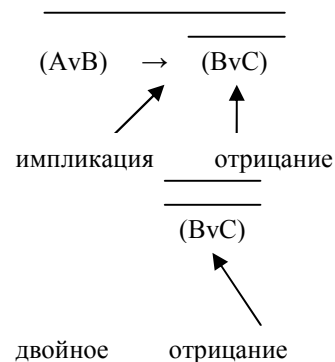
$$B \& (A \vee 1) \vee A \& C \vee B \& C = B \vee A \& C \vee B \& C.$$

8. Переставим местами слагаемые, сгруппируем и вынесем B за скобки. Получим:

$$B \vee A \& C \vee B \& C = B \& (1 \vee C) \vee A \& C.$$

9. Применим (6) и получим ответ:

$$B \& (1 \vee C) \vee A \& C = B \vee A \& C.$$



Ответ: $F = \neg((A \vee B) \rightarrow \neg(B \vee C)) = B \vee A \& C.$

4 Контрольные вопросы.

1) $F = \neg(X \& Y \vee \neg(X \& Y)).$

2) $F = \bar{X} \& \neg(\bar{Y} \vee X).$

3) $F = (X \vee Z) \& (X \vee \bar{Z}) \& (\bar{Y} \vee Z).$

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.

- Описание основных этапов работы
- Результаты выполнения работы;
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №6.

Элемент "исключающее или" (XOR). Поиск неисправностей.

Объём учебного времени – 4 ч

Методические рекомендации

1.Цель работы

- изучить принципы представления информации в ЭВМ.
- изучить принципы построения логических элементов.
- Ознакомление с возможностями программы анализа электронных и логических схем Electronic WorkBench.
- Изучение интерфейса пользователя. Ознакомление с терминами и понятиями. Построение простейших электронных и логических схем

2.Перечень необходимых средств обучения:

2.11 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.12 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

3.1 Структура окна и система меню

Окно содержит строку команд меню, строку основных типовых электронных устройств, поле для составления исследуемой схемы и полосы управления прокруткой. Первые команды меню типовые и пояснений не требуют.

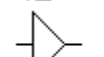
Таблица 6.1 Основные команды меню

Меню Circuit	
Rotate	вращение выделенного компонента (Ctrl + R)
Flip Horizontal	перевернуть по горизонтали
Flip Vertical	перевернуть по вертикали
Components Properties	свойства компонента
Create SubCircuit	(Ctrl + B) - объединяет выделенные элементы схемы в под-схему и помещает ее(полученную подсхему) под выбранным именованием в окно Custom
Меню Analysis	
Activate	запуск моделирования (Ctrl + Q)
Pause	прерывание моделирования (F9)
Stop	остановка моделирования (Ctrl + T)

3.2 Технология создания схем

Для открытия нужной библиотеки компонентов нужно подвести курсор мыши к соответствующей иконке и нажать один раз её левую кнопку. После размещения компонентов производится соединение их выводов проводниками. Для выполнения подключения курсор мыши подводится к выводу компонента и нажимается левая кнопка и протягивается к выводу другого компонента, после чего кнопка мыши отпускается и соединение готово.

3.3 Основные компоненты

- ✓ группа **SOURCES**
 -  заземление (метка) **Ground**
 -  источник фиксированного напряжения **Battery**
- ✓ группа **BASIC**
 -  точка соединения проводников, используется также для введения на схему надписей длиной не более 14 символов
 -  переключатель **Switch**, управляемый нажатием задаваемой клавишей клавиатуры (в квадратных скобках), по умолчанию - клавиша пробела
- ✓ группа **INDICATORS**
 -  **светоиндикатор** (свет свечения может быть настроен красным, зелёным и синим)
- ✓ группа **LOGIC GATES**
 -  логический элемент "И"
 -  логический элемент "ИЛИ"
 -  логический элемент "НЕ"
 -  логический элемент "ИЛИ-НЕ"
 -  – логический элемент "И-НЕ"
 -  – логический элемент исключающее "ИЛИ"
 -  – буфер

3.4 Задание к работе

3.4.1 Собрать схему логического элемента "ИЛИ" для четырёх входных сигналов.

- В группе **Logic Gates** требуется выбрать логический элемент **ИЛИ**.
- К выходу логического элемента присоединяем из группы **INDICATORS** красный светодиод.
- Для получения логического сигнала (0 или 1) воспользуемся источником напряжения **Battery** и переключателем **Switch**
- Затем набираем 4 источника и 4 переключателя, при этом присваиваем каждому переключателю клавишу переключения (**q, w, e, r**).
- Затем соединяем входы логической схемы "ИЛИ" с каждым из переключателей.
- Проверьте работу схемы. Проверка состоит в подаче различных кодовых комбинаций на вход логической схемы. На выходе логической схемы "ИЛИ" появляется логическая 1 (горит светодиод) при подаче логических 1 (потенциал 5 вольт) хотя бы от одного из четырех входов логической схемы "ИЛИ"

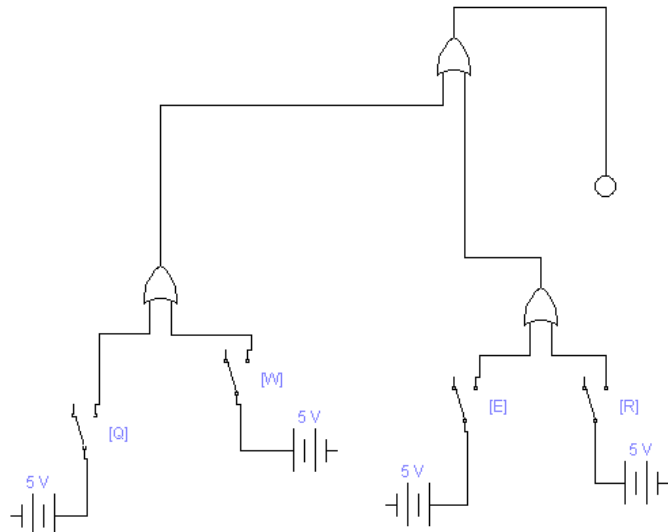


Рисунок 6. 1 Схема 4-ИЛИ

- Добавьте заземление к каждому элементу питания.
- Создайте подсхему, которую назовите **OR** (для этого выделите схему и выберите в меню **Circuit - Create SubCircuit**).

3.4.2 Собрать схему логического элемента "И" для четырёх входных сигналов.

- При выборе элемента «И». Двумя щелчками мыши на изображении логического элемента переходим к настройкам параметров логического элемента "И". Выбираем количество входов = 4.
- Присвойте название логическому элементу **AND_4**.
- Присвойте каждому переключателю клавишу переключения (1, 2, 3, 4)
- Добавьте индикаторы А, В, С и D.
- Постройте схему и проверьте её работу:

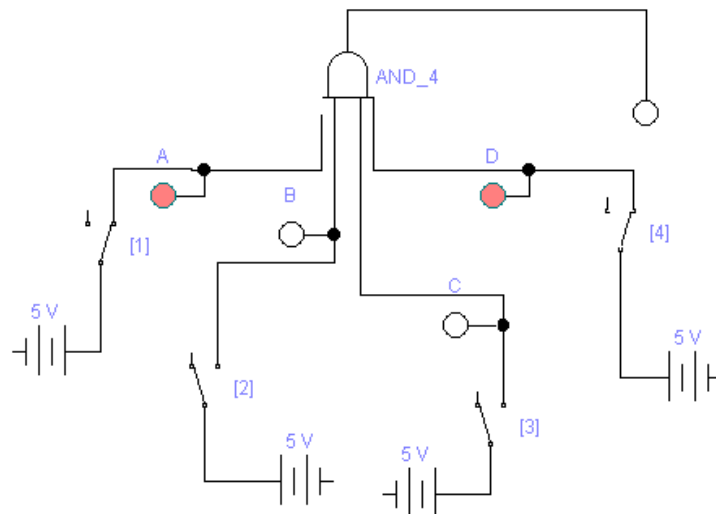


Рисунок 6.2 Схема 4-И

- На основании созданной схемы создайте подсхему, которую назовите **And**.
- Собрать схему логического элемента "ИЛИ-НЕ" для четырёх входных сигналов. Проверьте работу построенной схемы. Создайте подсхему **Or_No**.

– Собрать схему логического элемента "И-НЕ" для четырёх входных сигналов. Проверьте работу построенной схемы. Создайте подсхему And_No. Сохраните документ.

– Собрать схему логического элемента "XOR" – исключающее ИЛИ ($x \oplus y$) для четырёх входных сигналов. Проверьте работу построенной схемы. Создайте подсхему XOR. Сохраните документ.

– Создайте новый документ. Собрать схемы, предложенные ниже. С помощью ключей (управляются клавишами) подать на вход схемы различные комбинации переменных A, B и C. Значения функции F занести в таблицу 9:

Таблица 6.2 Результаты эксперимента

A	B	C	F1	F2	F3	F4
0	0	0				
1	0	0				
0	1	0				
0	0	1				
1	1	0				
1	0	1				
0	1	1				
1	1	1				

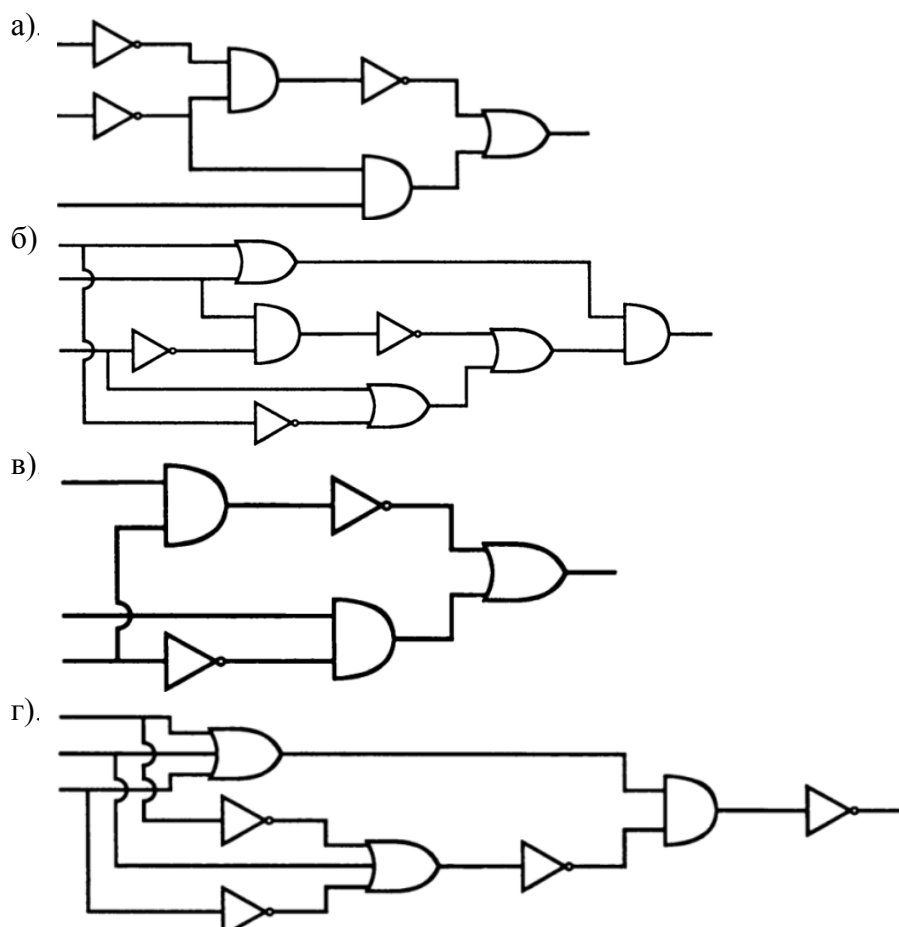


Рисунок 6.3. Схемы для построения

4 Контрольные вопросы.

- Какие логические элементы имеются в библиотеке EWB?
- Перечень основных элементов из библиотеки EWB, необходимый для моделирования логических схем и цифровых устройств ЭВМ.
- Как производится соединение более двух входов или выходов между собой?
- Какой командой можно скопировать изображение схемы в отчёт по лабораторной работе, подготавливаемой в текстовом редакторе WORD.
- С помощью каких элементов можно смоделировать подачу логической 1 и логического 0?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Собранные схемы элементов: 4-ИЛИ, 4-И, 4-ИЛИ-НЕ, 4-И-НЕ, 4-XOR
- Заполненная таблица 6.2 как результат работы собранных схем 6.3.6 а) б) в) г).
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №7.

Изучение работы RS-, JK- триггеров и принципов их работы. Асинхронные счетчики **Объем учебного времени – 4ч**

Методические рекомендации

1.Цель работы

- изучить принципы построения логических элементов.
- Ознакомление с возможностями программы анализа электронных и логических схем Electronic WorkBench.
- Изучение интерфейса пользователя. Ознакомление с терминами и понятиями. Построение простейших электронных и логических схем

2.Перечень необходимых средств обучения:

3.5 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

3.6 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Триггеры - это простейшие представители цифровых устройств последовательностного типа т.е. устройств или автоматов, обладающих памятью. Последовательностные устройства характеризуются определенным числом внутренних состояний. В каждый конкретный момент времени оно может находиться только в одном из возможных состояний. Переход устройства из одного состояния в другое осуществляется под действием внешних управляющих сигналов. Однако, значение выходного сигнала нельзя определить только по состоянию входных, поскольку оно зависит не только от входной информации, но и от предыдущего состояния устройства.

Триггером называется устройство способное находиться в одном из двух устойчивых состояний и скачкообразно переходить из одного в другое под действием внешних управляющих сигналов. Данные состояния триггера определяются как состояние 0 и состояние 1. Триггер может находиться в любом из состояний неограниченный промежуток времени, до поступления внешнего воздействия или отключения питания.

Простейший триггер представляет собой одноразрядную ячейку памяти. В общем случае он снабжается определенной входной комбинационной схемой. Триггер снабжается двумя выходами: прямым Q и инверсным. Состояние сигналов на данных выходах может быть только противофазным. Говоря о состоянии триггера подразумевают значение выходного сигнала на выходе Q. При наличии уровня лог. 1 на прямом выходе Q говорят, что: «триггер находится в состоянии 1», либо: «триггер установлен», либо «триггер взведен». При наличии же на данном выходе лог. 0 оперируют понятиями: «триггер находится в состоянии 0», либо: «триггер сброшен».

Существующие типы триггеров могут быть классифицированы по различным признакам. Наиболее часто триггеры классифицируют по типу используемых информационных входов. Различают следующие типы основных информационных входов триггера:

- R – раздельный вход сброса триггера ($Q=0$);
- S – раздельный вход установки триггера ($Q=1$);
- K – вход сброса универсального триггера ($Q=0$);
- J – вход установки универсального триггера ($Q=1$);
- T – счетный вход триггера;
- D – информационный вход переключения триггера в состояние, соответствующее логическому уровню на этом входе;
- C – управляющий или синхронизирующий вход.

Кроме этих основных входов некоторые триггеры могут снабжаться входом V. Вход V блокирует работу триггера и он сколь угодно долго может сохранять ранее записанную в него информацию.

С точки зрения типа используемых входов различают RS-, D-, T-, JK-, VD-, VT-триггеры.

По виду реакции на входные сигналы триггеры подразделяют на асинхронные и синхронные

Асинхронный триггер изменяет свое состояние непосредственно в момент изменения сигнала на его информационных входах. Синхронный триггер изменяет свое состояние лишь в строго определенные (тактовые) моменты времени, соответствующие действию активного сигнала на его синхронизирующем входе C.

УГО асинхронного RS – триггера представлено на рисунке 7.1.

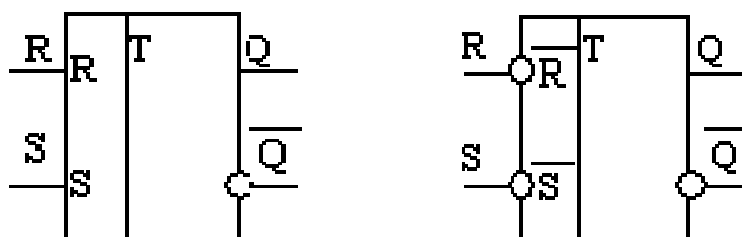


Рисунок 7.1 УГО асинхронный RS – триггер с прямыми и инверсными входами

По виду активного логического сигнала, действующего на синхронизирующем входе триггеры подразделяют на статические – управляемые уровнем, и динамические – управляемые перепадом входного сигнала. При этом информационные входы могут быть прямыми и инверсными. Для переключения триггера на его прямой вход необходимо подать сигнал X, а на инверсный сигнал.

Так, определения "синхронный RS-триггер с прямыми статическими входами" и "синхронный RS-триггер со статическими входами" означают, что рассматриваемый триггер имеет два информационных входа: вход установки S, вход сброса R и синхронизирующий вход C. Переключение триггера происходит в моменты времени, определяемые появлением активного логического сигнала на входе синхронизации ($C = 1$), причем для переключения на входы R или S необходимо подать высокий логический уровень, т. е. сигнал лог.1 ($R = 1$ или $S = 1$). Эти входы соответственно обозначают: C, R, S. УГО данного триггера приведено на рисунке 7.2.а. Если триггер определен как "синхронный RS-триггер с инверсными статическими входами", то в этом случае, он также имеет два информационных входа: S, R и синхронизирующий вход C. Однако переключение триггера будет происходить при наличии на входе C сигнала лог.0. И активными уровнями для входов R и S будут сигналы лог. 0. УГО данного триггера приведено на рисунке 7.2.

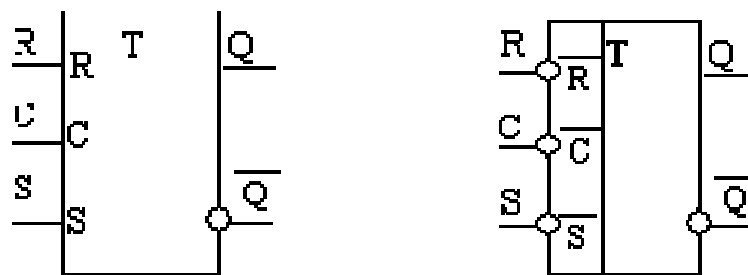


Рисунок 7.2. УГО синхронного RS – триггера с прямыми и инверсными входами.

На УГО RS триггеров допускается менять местами входы R и S. Для триггеров других типов это не допустимо. Простейшие асинхронные RS триггеры строятся на двух элементах 2ИЛИ-НЕ либо 2И-НЕ. Поскольку активными сигналами для элементов ИЛИ, ИЛИ-НЕ являются лог. 1, то триггер, построенный на таких элементах, будет чувствителен к сигналам высокого уровня. Он будет переключаться при подаче на его входы лог.1. Для элементов и, И-НЕ активными уровнями являются уровни лог.0. Поэтому триггер, созданный из данных элементов, будет переходить в новое состояние при поступлении на его входы сигналов лог.0. Функциональные схемы данных триггеров приведены на рисунке 7.3.а и 7.3.б – соответственно. Из рисунка видно, что в триггерах меняется положение входов R и S.

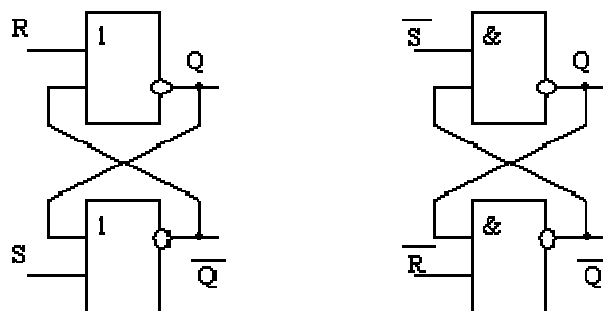


Рисунок 7.3. Функциональные схемы асинхронных RS триггеров: а - на элементах ИЛИ-НЕ; б – на элементах И-НЕ.

Описание работы триггера может задано в виде таблицы истинности или в виде функции алгебры логики (ФАЛ). Отличительной особенностью описания триггеров, как представителей класса последовательных устройств, является использование в качестве дополнительной входной переменной значения сигнала Q_n , т.е. предыдущего значения выходного сигнала триггера. Обобщенная таблица истинности асинхронных RS триггеров, построенных на логике ИЛИ-НЕ и И-НЕ представлена в таблице 7.1.

Таблица 7.1. Таблица истинности RS триггеров.

Значения исходных параметров			Состояние выхода	
			Логика ИЛИ-НЕ	Логика И-НЕ
R	S	Q_n	Q_{n+1}	Q_{n+1}
0	0	0	0	запрет
0	0	1	1	
0	1	0	1	1
0	1	1	1	
1	0	0	0	0
1	0	1	0	
1	1	0	запрет	Q_n
1	1	1		

Триггер типа RS не допускает одновременно наличие на входах R и S активных сигналов. В этом случае не выполняется условие его функционирования, поскольку на выходах Q и логические уровни перестают быть взаимно инверсными и принимают одинаковые значения. Данные комбинации считаются запрещенными. Логическое выражение, определяющее функционирование асинхронного RS – триггера с прямыми входами, имеет вид:

$$Q_{n+1} = S \vee \bar{R}Q_n \quad (7.1)$$

а для триггера с инверсными входами:

$$Q_{n+1} = \bar{S}VRQ_n \quad (7.2)$$

Синхронный RS – триггер снабжен дополнительным входом синхронизации C. ФАЛ синхронных RS – триггеров с прямыми и инверсными входами легко получить из выражений (7.1) и (7.2):

$$Q_{n+1} = \bar{C}Q_n \vee C(S \vee \bar{R}Q_n) \quad (7.3)$$

$$Q_{n+1} = \bar{C}Q_n \vee C(\bar{S} \vee RQ_n) \quad (7.4)$$

В качестве примера на рисунке 7.4 приведена функциональная схема синхронного RS триггера на элементах ИЛИ-НЕ.

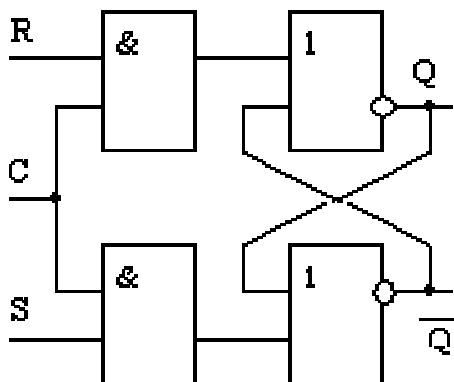


Рисунок 7.4 Функциональная схема синхронного RS триггера на элементах ИЛИ-НЕ

4.Контрольные вопросы.

- В чем отличие комбинационных и последовательностных логических устройств?
- Какие триггеры называются асинхронными, а какие синхронными?

5.Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Функциональные схемы и УГО RS, D, T, JK-триггеров
- Таблицы истинности и ФАЛ указанных триггеров
- Результаты выполнения работы;
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

– Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.

- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №8.
D-триггеры. Цепи синхронизации.
Объём учебного времени – 4 ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы устройств триггера
- Знакомство с базовыми устройствами триггер из библиотеки EWB

2.Перечень необходимых средств обучения:

2.1 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.2 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Более сложным преобразователем информации являются **схемы с памятью**. Наличие памяти в схеме позволяет запоминать промежуточные состояния обработки и учитывать их значения в дальнейших преобразованиях. Выходные сигналы $Y = (y_1, y_2, \dots, y_m)$ в схемах данного типа формируются не только по совокупности входных сигналов $X = (x_1, x_2, \dots, x_n)$, но и по совокупности состояний схем памяти $Q = (q_1, q_2, \dots, q_n)$. При этом различают текущий дискретный момент времени t и последующий $(t + 1)$ момент времени (рис. 33).

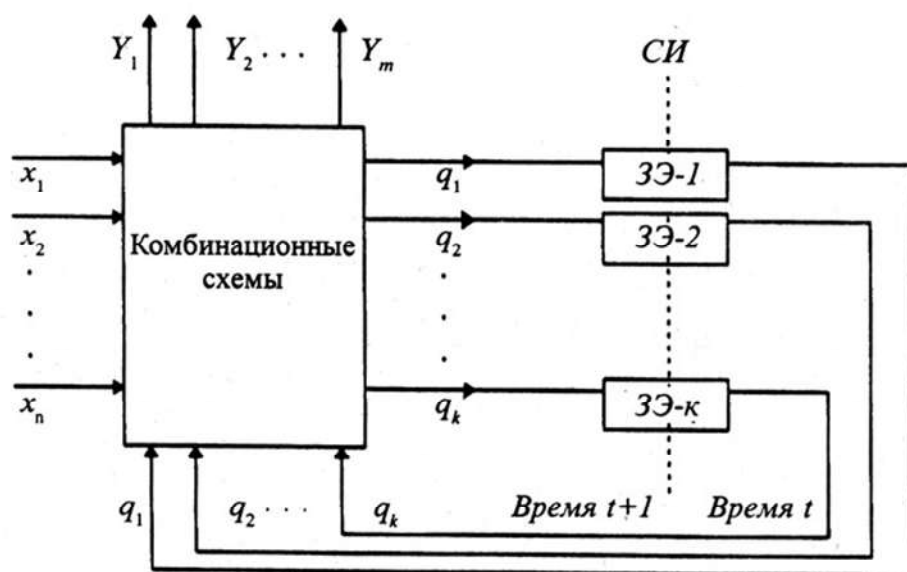


Рисунок 8.1 Обобщенная структура схемы с памятью

Передача значения Q между моментами времени t и $(t+1)$ осуществляется обычно с применением двухступенчатой памяти и синхронизирующих импульсов (СИ).

В качестве простейшего запоминающего элемента в современных ЭВМ используют триггеры. Самая простейшая схема триггера (RS-триггер) может быть синтезирована по общим правилам.

В данном случае для этого необходимо построить автомат памяти - триггер, имеющий вход **R** (Reset - сброс), для установки элемента в «нулевое состояние» и вход **S** (Set - установка) - для установки элемента в «единичное» состояние. При отсутствии сигналов **R=S=0** элемент должен сохранять свое состояние до тех пор, пока не будут получены новые сигналы на входе **R** или **S**.

Условия работы триггера могут быть показаны в виде таблицы переходов, представляющей собой модификацию таблицы истинности. Наиболее простой вид она имеет для автомата, описываемого системой уравнений автомата Мура.

Триггеры – устройства, имеющие два устойчивых состояния. Под действием управляющих сигналов они переходят из одного состояния в другое и после снятия сигналов хранят это состояние до тех пор, пока не отключено напряжение питания. Таким образом, триггер является ячейкой памяти для одного двоичного разряда, т. е. бита информации. Справа приведена схема асинхронного одноканального RS – триггера на логических элементах И-НЕ.

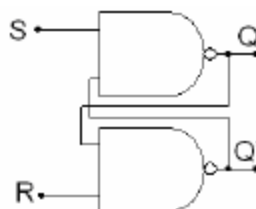


Рисунок 8.2. Схема асинхронного одноканального RS – триггера

По характеру реакции на входные сигналы триггеры делятся на два типа: асинхронные и синхронные. Асинхронный триггер характеризуется тем, что входные сигналы действуют на состояние триггера непосредственно с момента их подачи на входы, в синхронных триггерах - только при подаче синхронизирующего сигнала на управляющий вход С.

Таблица 8.1 Условия работы триггера

Входы			Выход	
R	S	q_t	Состояние q_{t+1}	Режим
0	0	0	0	Хранение
0	0	1	1	Хранение
1	0	0	0	Установка 0
1	0	1	0	Установка 0
0	1	0	1	Установка 1
0	1	1	1	Установка 1
1	1	0	?	Запрещенное состояние
1	1	1	?	Запрещенное состояние

Содержание таблицы расшифровывается следующим образом. Элемент памяти может сохранять значение $q_t=0$ или $q_t=1$ в зависимости от установки ранее установленного состоя-

ния. При отсутствии входных сигналов на входах R и S ($R=0$ и $S=0$) значения q_{t+1} в первой строке таблицы в точности повторяют значения q_t . При поступлении сигнала $R=1$ (сигнала установки «нуля») элемент независимо от своего состояния принимает значение, равное нулю, $q_{t+1} = 0$. Если же на вход S поступает сигнал установки «единицы» ($S=1$), то $q_{t+1}=1$ независимо от предыдущего состояния q_t . Одновременное поступление сигналов на входы R и S является запрещенной ситуацией, так как она может привести к непредсказуемому состоянию. В схемах формирования сигналов R и S должны быть предусмотрены блокировки, исключающие их совпадения, $S=R=1$.

RS-триггер нашел широкое распространение в схемах ЭВМ. Одиночные триггеры этого типа часто используются в различных блоках управления. В асинхронных RS-триггерах имеется один существенный недостаток, обусловленный самой логикой их построения, т.е. в них сигналы R и S должны быть разнесены во времени. Дополнение этого триггера комбинационными схемами синхронизации на входе и выходе позволяет получить триггеры с более сложной логикой работы: синхронные RS-, T-, JK-, D-триггеры и целый ряд комбинированных RST-, JKRS-, DRS-триггеров

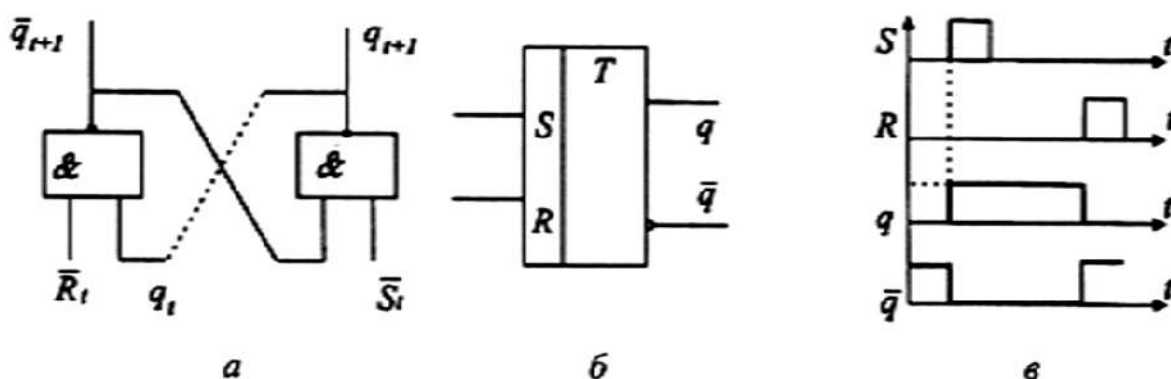


Рисунок 8.3. Схема асинхронного RS-триггера: а — схема; б — обозначение на принципиальных электрических схемах; в — временная диаграмма

Прописные буквы в названиях триггеров обозначают:

- R (Reset — сброс) — вход установки триггера в нулевое состояние, $Q = 0$
- S (Set — установка) — вход установки триггера в единичное состояние, $Q = 1$
- T (Toggle — релаксатор) — счетный вход триггера
- J (Jerk — внезапное включение) — вход установки JK-триггера в единичное состояние, $Q = 1$
- K (Kill — внезапное выключение) — $Q = 0$
- D (Delay — задержка) — вход установки триггера в единичное или нулевое состояние на время, равное одному такту
- C (Clock — часы) — вход синхронизирующих тактовых импульсов

T-триггер или, иначе говоря, триггер со счетным входом при значении $X_t = 0$ триггер сохраняет свое ранее установленное состояние - режим хранения состояния, при $X_t = 1$ триггер переходит в противоположное состояние. Таблица переходов (табл. 19) и диаграмма работы (рис. 36 в) отражают динамику работы этого элемента.

Таблица 8.2 Таблица переходов T-триггера

Входные сигналы	Состояние q_t		Режим
X_t	0	1	
0	0	1	Хранение Инверсия
1	1	0	

Используя таблицу переходов, можно получить логическую функцию, реализуемую Т-триггером:

$$q_{t+1} = \overline{q_t} \cdot x_t \vee q_t \cdot \overline{x_t} = q_t \oplus x_t$$

Нетрудно видеть, как реализовать полученную зависимость – для этого необходимо в RS-триггере выход q_t соединить с R входом RS-триггера, а выход $\neg q_t$ соединить с S входом. На рис. 6.1, а показано, как двухтактный RS-триггер преобразуется в Т-триггер

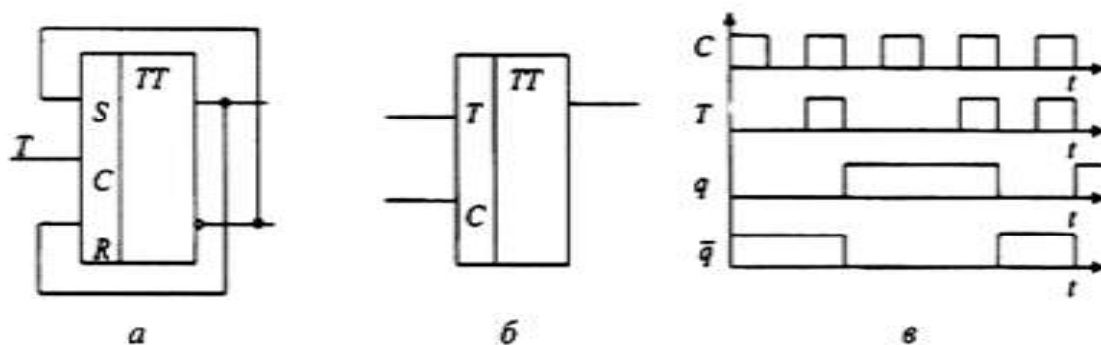


Рисунок 8.4 Схема триггера со счетным входом: а — функциональная схема; б — условное обозначение; в — временная диаграмма

D-триггер обычно строится на основе двухтактного RS- или JK-триггера. Он предназначен для хранения состояния (1 или 0) на один период тактовых импульсов (с задержкой на 1 такт). Его переходы отражены в табл.20

Таблица 8.3 Таблица переходов D-триггера

Входные сигналы	Состояние q_t		Режим
D	0	1	
0	0	0	Установка 0
1	1	1	Установка 1

Используя таблицу переходов, можно получить логическую функцию, реализуемую D-триггером:

$$q_{t+1} = q_t x_t \vee \overline{q_t} \overline{x_t}$$

На рис. 8.5, а и б представлены варианты построения D-триггера

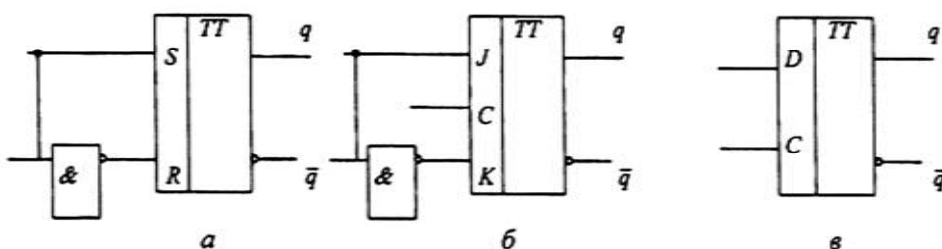


Рисунок 8.5 D-триггер: а — функциональная схема на основе RS-триггера; б — функциональная схема на основе JK-триггера; в — условное обозначение

Все перечисленные элементы памяти позволяют хранить одну единицу информации бит.

Триггеры используются, в основном, в вычислительной технике для организации компонентов вычислительных систем: регистров, счётчиков, процессоров, ОЗУ.

В библиотеке EWB триггеры представлены тремя типами: RS, JK и D:

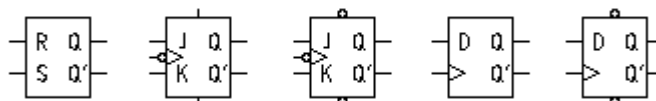


Рисунок 8.6 Типы триггеров: а) RS триггер, б) JK триггер, в) D триггер

Назначение выводов триггеров следующее:

Для всех триггеров выходы Q – **прямой**, Q' – **инверсный** (обратный).

Для **RS – триггера** R – установка триггера в 0, при сигнале 1 на этом входе Q=0, Q'=1; S – установка в 1, при сигнале 1 на этом входе Q=1, Q'=0; комбинация R=1, S=1 не изменяет состояние выходов и относится к запрещенным.

Для **JK триггера** J, K – информационные входы, > - тактовый вход; вывод сверху – асинхронная предустановка триггера в единичное состояние (Q=1) вне зависимости от состояния сигналов на входах (функционально аналогичен входу S RS триггера); вывод внизу – асинхронная предустановка триггера в нулевое состояние (так называемая очистка триггера, после которой Q'=1); наличие кружочков на изображениях выводов обозначает, что активными являются сигналы низкого уровня, а для тактового входа – что переключение триггера производится не по переднему фронту тактового импульса, а по его срезу.

Для **D – триггера** вход D – информационный, состояние этого входа после подачи тактового импульса запоминается триггером, т. е. при D=1 имеем Q=1, при D=0 – Q=0.

3.1Задание к работе

3.1.1 Постройте схему функционирования RS-триггера:

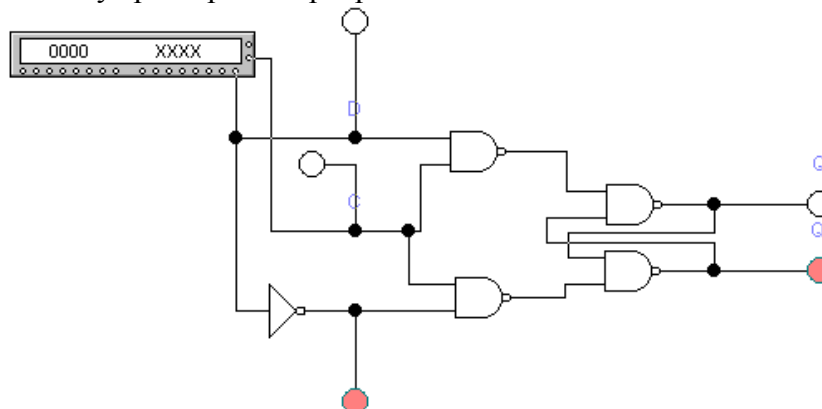
- Какую роль играет сигнал синхронизации C? При каком значении сигнала C происходит установка триггера в 0 и 1?
- В тетради составьте таблицу истинности RS-триггера:

Таблица 8.4 Таблица истинности RS-триггера

Вход 1 (S)	Вход 2 (R)	Выход 1 (Q)	Действие триггера
1	0		
0	1		
0	0		
1	1	неустойчивое состояние	запрещено

- Если на входы подать «0», то как изменится значение на выходах?
- Каковы должны быть возможные комбинации входных сигналов для имитации работы триггера в различных режимах.
- Назовите вход установки в 0, в 1?

- Постройте схему функционирования RS-триггера с применением нужного типа триггера из библиотеки EWB. Проверьте работу триггера по построенной таблице.
- Дополните схему триггера инвертором:



– Рисунок 8.7. Схема функционирования RS-триггера

- Сделайте вывод, от значения каких сигналов зависит состояние выхода триггера?
- Проверьте состояние выхода при $D=1$, $D=0$.
- Для получения какого сигнала используется выход синхросигнала генератора слова?

Если в D – триггере D – вход соединить с инверсным выходом Q' , то получится T – триггер с одним тактовым C – входом

Постройте аналогичную схему с применением нужного типа триггера из библиотеки EWB. Проверьте работу триггера

4 Контрольные вопросы.

- Что такое триггер? Какого типа они бывают?
- Что такое синхронный (асинхронный) триггер?
- Проведите исследование типов триггеров, не описанных в задании

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Схема функционирования D-триггера
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

1. <http://novtex.ru/IT> - журнал "Информационные технологии"
2. <http://infojournal.ru> - журнал «Информатика и образование»
3. <http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №9.
D-триггеры. Поиск неисправностей
Объём учебного времени – 2ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы устройств триггера.
- Знакомство с базовыми устройствами триггер из библиотеки EWB.

2.Перечень необходимых средств обучения:

2.1 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.2 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Универсальный JK триггер также относится к классу двухступенчатых. Он может быть только синхронным. По логике своей работы JK-триггер наиболее близок к RS-триггеру, но в отличие от последнего не имеет запрещенных входных комбинаций. Вход J является входом установки триггера в состояние лог. 1, а вход K – в состояние лог. 0. Наличие активных уровней сигналов на данных входах не нарушает логику работу триггера. В этом случае он функционирует как Т-триггер. Функциональная схема JK-триггера приведена на рисунке 1.13, а УГО – на рисунке 1.14. Универсальный JK-триггер может быть использован в качестве базового для реализации на нем всех типов триггеров.

ФАЛ JK-триггера имеет вид:

$$Q_{n+1} = CQ_n \vee C(Q_n J \vee Q_n K) \quad (9.1)$$

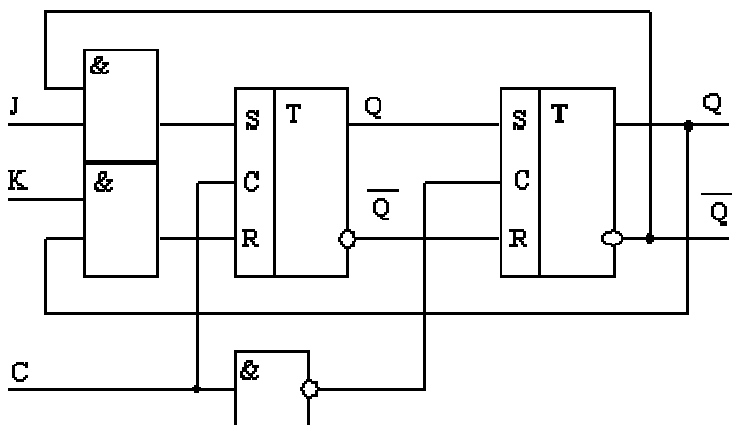


Рисунок 9.1. Функциональная схема JK–триггера

3.1 Постройте JK триггер (самостоятельно).

- Назовите вход триггера, который устанавливает его состояние на 0.
- Назовите вход триггера, который устанавливает его состояние на 1.
- Назовите тактовый вход триггера.
- При подключении используйте также два информационных входа. Сделайте вывод: при каких входных значениях сигналов происходит изменение сигналов на выходе?

4 Контрольные вопросы.

- Чем вызвана необходимость введения двухступенчатости триггеров?
- Какие типы триггеров должны быть двухступенчатыми?
- Какой тип триггеров называется «универсальным» и почему?
- Какой тип триггеров имеет запрещенные комбинации входных сигналов и почему?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Схема функционирования JK-триггера
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб. пособие для сред. проф. образования. - М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб. пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

1. <http://novtex.ru/IT> - журнал "Информационные технологии"
2. <http://infojournal.ru> - журнал «Информатика и образование»
3. <http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №10.

Таймер. Двоичный счетчик с последовательным переносом.

Объём учебного времени – 4ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.

2.Перечень необходимых средств обучения:

1. Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2. Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Счётчик числа импульсов — устройство, на выходах которого получается двоичный (двоично-десятичный) код, определяемый числом поступивших импульсов. Счётчики могут строиться на Т-триггерах. Основной параметр счётчика — модуль счёта — максимальное число единичных сигналов, которое может быть сосчитано счётчиком. Счётчики обозначают через СТ (от англ. counter).

Счётчики классифицируют:

- по числу устойчивых состояний триггеров
- на двоичных триггерах
- на троичных триггерах[1]
- на n-ичных триггерах
- по модулю счёта:
- двоично-десятичные (декада);
- двоичные;
- с произвольным постоянным модулем счёта;
- с переменным модулем счёта;
- по направлению счёта:
- суммирующие;
- вычитающие;
- реверсивные;
- по способу формирования внутренних связей:
- с последовательным переносом;
- с параллельным переносом;
- с комбинированным переносом;
- кольцевые;

- по способу переключения триггера:
- синхронные;
- асинхронные;

Схему двоичного счетчика можно получить с помощью формального синтеза, однако более наглядным путем представляется эвристический. Таблица истинности двоичного счетчика — последовательность двоичных чисел от нуля до $M-1$. Наблюдение за разрядами чисел, составляющих таблицу, приводит к пониманию структурной схемы двоичного счетчика. Состояния младшего разряда при его просмотре по соответствующему столбцу таблицы показывают чередование нулей и единиц вида 01010101..., что естественно, т. к. младший разряд принимает входной сигнал и переключается от каждого входного воздействия. В следующем разряде наблюдается последовательность пар нулей и единиц вида 00110011... . В третьем разряде образуется последовательность из четверок нулей и единиц 00001111... и т.д. Из этого наблюдения видно, что следующий по старшинству разряд переключается с частотой, в два раза меньшей, чем данный. Известно, что счетный триггер делит частоту входных импульсов на два. Сопоставив этот факт с указанной выше закономерностью, видим, что счетчик может быть, построен в виде цепочки последовательно включенных счетных триггеров. Заметим, кстати, что согласно ГОСТу входы элементов изображаются слева, а выходы справа. Соблюдение этого правила ведет и тому, что в числе, содержащемся в счетчике, младшие разряды расположены левее старших.

Двоичные счетчики с параллельным переносом

Выше рассмотрены схемы двоичных последовательных счетчиков, то есть таких счетчиков, в которых при изменении состояния определенного триггера возбуждается последующий триггер, причем триггеры меняют свои состояния не одновременно, а последовательно. Если в данной ситуации должны изменить свои состояния n триггеров, то для завершения этого процесса потребуется n интервалов времени, соответствующих времени изменения состояния каждого из триггеров. Такой последовательный характер работы является причиной двух недостатков последовательного счетчика: меньшая скорость счета по сравнению с параллельными счетчиками и возможность появления ложных сигналов на выходе схемы. В параллельных счетчиках синхронизирующие сигналы поступают на все триггеры одновременно. Последовательный характер переходов триггеров счетчика является источником мощных сигналов на его выходах. Например, в счетчике, ведущем счет в четырехразрядном двоичном коде с “весами” 8-4-2-1, при переходе от числа $710 = 01112$ к числу $810 = 10002$ на выходе появится следующая последовательность сигналов: 0111 → 0110 → 0100 → 0000 → 1000. Это означает, что при переходе из состояния 7 в состояние 8 на входах счетчика на короткое время появятся состояния 6; 4; 0. Эти дополнительные состояния могут вызвать ложную работу других устройств. С целью уменьшения времени протекания переходных процессов можно реализовать счетчик в варианте с подачей входных импульсов одновременно на все триггеры. В этом случае получим счетчик с параллельным переносом. Здесь на информационные входы триггеров подаются сигналы, являющиеся логической функцией состояния счетчика и определяющие конкретные триггеры, которые изменяют свое состояние при данном входном импульсе. Принцип стробирования сводится к следующему: триггер меняет свое состояние при пропуске очередного импульса синхронизации, если все предыдущие триггеры находились в состоянии логической единицы. Параллельные счетчики имеют более высокое быстродействие по сравнению с последовательными, поскольку синхронизирующие импульсы поступают на все триггеры одновременно. Максимальным быстродействием обладают синхронные счетчики с параллельным переносом, структуру которых найдем эвристически, рассмотрев процессы прибавления единицы к двоичным числам и вычитания ее из них.

Счетчики с последовательно-параллельным переносом

В связи с ограничениями на построение счетчиков с параллельным переносом большой разрядности широкое распространение получили счетчики с групповой структурой, или

счетчики с последовательнопараллельным переносом. Разряды таких счетчиков разбиваются на группы, внутри которых организуется принцип параллельного переноса. Сами же группы соединяются последовательно с использованием конъюнкторов, формирующих перенос в следующую группу при единичном состоянии всех триггеров предыдущих. При единичном состоянии всех триггеров группы приход очередного входного сигнала создаст перенос из этой группы. Эта ситуация подготавливает межгрупповой конъюнктор к прямому пропуску входного сигнала на следующую группу. В наихудшем для быстродействия случае, когда перенос проходит через все группы и поступает на вход последней, $t_{УСТ} = t \cdot (l - 1) + t_{ГР}$ где l – число групп, $t_{ГР}$ – время установления кода в группе. В развитых сериях ИС обычно имеется по 5...10 вариантов двоичных счетчиков, выполненных в виде 4-х разрядных групп (секций). Каскадирование секций может выполняться путем их последовательного включения по цепям переноса, организации параллельно-последовательных переносов или для более сложных счетчиков с двумя дополнительными управляющими входами разрешения счёта и разрешения переноса путем организации параллельных переносов и в группах и между ними. Особенностью двоичных счетчиков синхронного типа является наличие ситуаций с одновременным переключением всех его разрядов (например, для суммирующего счетчика при переходе от кодовой комбинации 11...1 к комбинации 00...0 при переполнении счетчика и выработке сигнала переноса). Одновременное переключение многих триггеров создает значительный токовый импульс в цепях питания ЦУ и может привести к сбою в их работе. Поэтому в руководящих материалах по использованию некоторых БИС/СБИС программируемой логики, в частности, имеется ограничение на разрядность двоичных счетчиков заданной величиной k (например, 16). При необходимости применения счетчика большей разрядности рекомендуется переходить к коду Грея, для которого переходы от одной кодовой комбинации к другой сопровождаются переключением всего одного, разряда. Правда, для получения результата счёта в двоичном коде придется использовать дополнительно преобразователь кода, но это является платой, за избавление от токовых импульсов большой интенсивности в цепях питания.

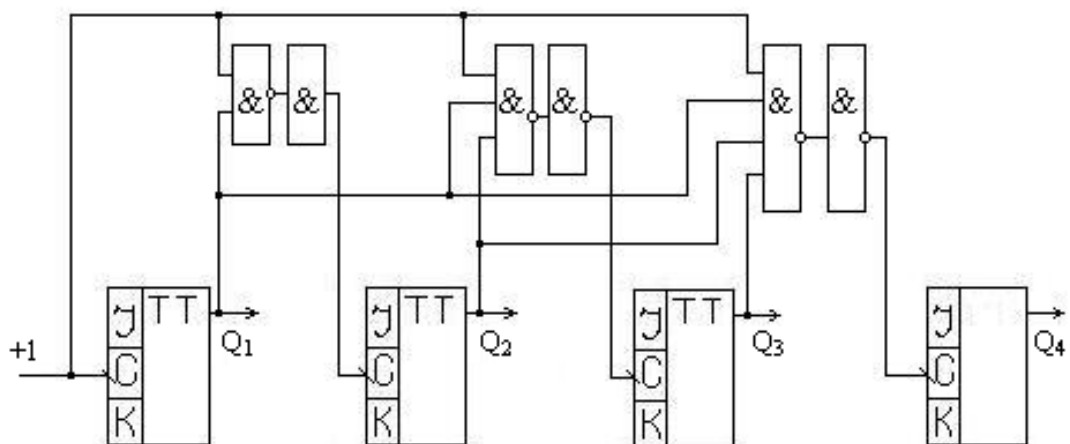


Рисунок 10.1 Схема асинхронного счетчика с последовательным переносом

3.2 Постройте схему реверсивный счетчика (самостоятельно).

4 Контрольные вопросы.

- Какие типы двоичных счетчиков Вы знаете?
- Чем определяется быстродействие счетчика?
- Какие способы повышения быстродействия счетчиков Вы знаете?
- Что такое граф перехода?
- Что такое функция перехода?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Схема асинхронного счетчика с последовательным переносом
- Схема реверсивного счетчика
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №11.

Цифро-аналоговый преобразователь. Аналогово-цифровой преобразователь. Объём учебного времени – 2ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.

2.Перечень необходимых средств обучения:

3. Технические средства обучения:

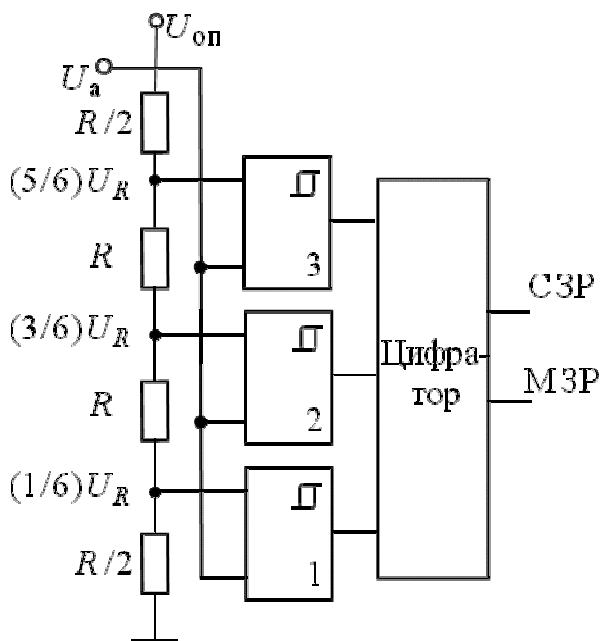
Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

4. Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Аналого-цифровой преобразователь (АЦП) преобразует аналоговое входное напряжение в выходной двоичный цифровой код, соответствующий квантованному входному сигналу.



3.1 Постройте схему параллельного аналого-цифровой преобразователя (самостоятельно).

4 Контрольные вопросы.

- Какие типы Аналого-цифровой преобразователей Вы знаете?
- Опишите назначения каждого из них.

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Схема АЦП двойного интегрирования
- Письменные ответы на контрольные вопросы

6. Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб. пособие для сред. проф. образования. - М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб. пособие для сред. проф. образования. - Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

1. <http://novtex.ru/IT> - журнал "Информационные технологии"
2. <http://infojournal.ru> - журнал «Информатика и образование»
3. <http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №12.
Дешифраторы и мультиплексоры. Демультимплексор.
Объём учебного времени – 2ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.

2.Перечень необходимых средств обучения:

5. Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

6. Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Анализатор предназначен для отображения на экране монитора 16-разрядных кодовых последовательностей одновременно в 16 точках схемы, а также в виде двоичных чисел на входных клеммах-индикаторах.

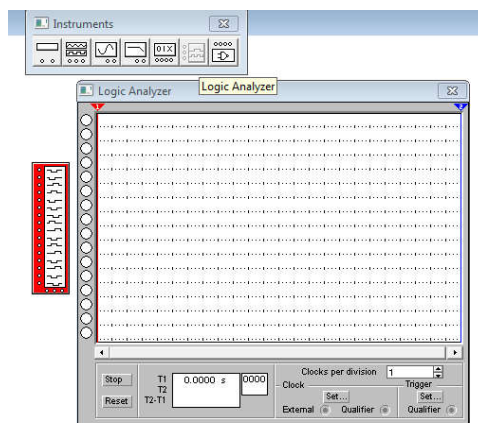


Рисунок 12.1. Logic Analyzer

Длительность развертки задается в окне TIME BASE. В блоке Clock имеются клеммы как для обычного (Extend), так и избирательного (Qualifier) источника запускающих сигналов, параметры которых могут установлены с помощью меню, вызываемого кнопкой Set. Запуск генератора можно осуществлять по переднему (Positive) или заднему (Negative) фронту запускающего сигнала с использованием внешнего (External) или внутреннего (Internal) источника. В окне Clock qualifier можно установить значение логического сигнала (0, 1 или X), при котором производится запуск анализатора.

Дополнительные условия запуска анализатора могут быть выбраны с помощью меню, которое вызывается кнопкой Set в блоке Trigger. С помощью этого окна в каналах А, В и С можно задать нужные двоичные 16-разрядные комбинации сигналов и затем в строке Trigger combinations установить дополнительные условия отбора:

- A OR B – запуск анализатора от канала А или В;
- A THEN B – запуск анализатора от канала А, если сигнал в канале В равен 1;
- (A OR B) THEN C – запуск анализатора от канала А или В, если сигнал в канале С равен 1.

В окне канала Trigger qualifier можно задать логические сигналы 1, 0 или X, при наличии которых производится запуск анализатора.

3.1 Дешифраторы и шифраторы

Дешифраторы и шифраторы по существу принадлежат к числу преобразователей кодов. С принятием шифрации связано представление о сжатии данных, с понятием дешифрации – обратное преобразование. Комбинационная схема, преобразующая поступающий на входы код в сигнал только на одном из ее выходов, называется **дешифратором**. В условных обозначениях дешифраторов и шифраторов используются буквы **DC** и **CD** (от слов decoder и coder соответственно).

В общем случае дешифратор с “n” входами может иметь до 2^n выходов, которые обычно нумеруются десятичными цифрами, эквивалентными значениям двоичных кодов. Например, в трехвходовом дешифраторе при подаче на вход двоичного кода 011 единичный сигнал появится на выходе 3, а на остальных выходах сигнал будет равен нулю.

Дешифраторы (ДШ) — это комбинационные схемы с n входами и $m = 2^n$ выходами. Единичный сигнал, формирующийся на одном из m выходов, однозначно соответствует комбинации входных сигналов.

Дешифратор, устройство для расшифровки (декодирования) сообщения и перевода содержащейся в нём информации на язык (в код) воспринимающей системы. В общем случае Д. имеет n входов и m выходов. Поступающая на входы Д. информация преобразуется — дешифрируется, — и на соответствующем выходе (группе выходов) выделяется сигнал, указывающий признак (или содержание) входной информации.

Функционирование дешифратора описывается системой логических уравнений составленных на основе таблицы истинности.

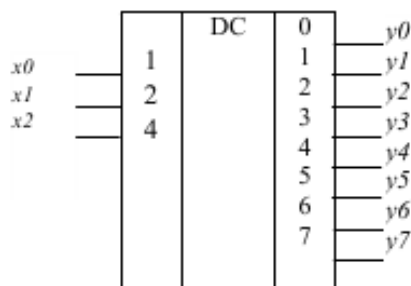


Рисунок 12.2 Условные обозначения дешифратора

Шифратор (ШР) решает задачу, обратную схемам ДШ, т. е. по номеру входного сигнала формирует однозначную комбинацию выходных сигналов. Номер входного сигнала определяется присутствием логической единицы на соответствующем входе (только одном). Соответствие комбинации выходных сигналов номеру входного можно задать соответствующей таблицей.

Шифратор — логическое устройство, выполняющее преобразование позиционного кода в n-разрядный двоичный код. Таким образом, шифратор - это комбинационное устройство, реализующее обратную дешифратору функцию. Шифраторы как отдельный класс

функциональных устройств представлены в наиболее богатой ТТЛ-серии всего двумя ИМС — 74147 и 74148.

Для 4 входных сигналов она может выглядеть следующим образом - таблица 12.1.

Таблица 12.1 Таблица соответствия выходных сигналов номеру входного сигнала шифратора

Входы				Выходы	
x ₁	x ₂	x ₃	x ₄	y ₁	y ₂
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	0	1
0	0	0	1	1	0

Состояния входов табл. 12.1 содержат только по одному единичному элементу. Другие произвольные комбинации входов недопустимы.

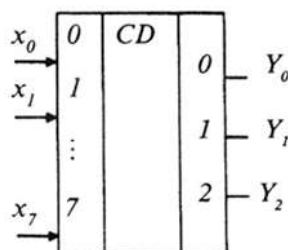


Рисунок 12.3. Обозначение шифратора на принципиальных электрических схемах

3.2 Задание к работе

3.2.1 **Исследование дешифратора** (при n информационных входах до 2ⁿ выходов) с числом входов =3 по таблице истинности:

Таблица 12.2. Таблица истинности дешифратора

№	x ₂	x ₁	x ₀	y ₀	y ₁	y ₂	y ₃	y ₄	y ₅	y ₆	y ₇
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

– Заметьте, что указанный в первом столбце порядковый номер строки является фактически десятичным эквивалентом подаваемого на входы двоичного числа. Приведите пример.

- Синтезируйте дешифратор как комбинационную схему, имеющую 8 (=23) выходов и 3 входа.
- Добавьте на проект **Word Generator** и **Logic Analyzer**. Введите в **Word Generator** слова для проверки правильности работы собранной схемы (от 0 до кол-ва выходов). Соединяем **Logic Analyzer** с выходами собранной схемой.

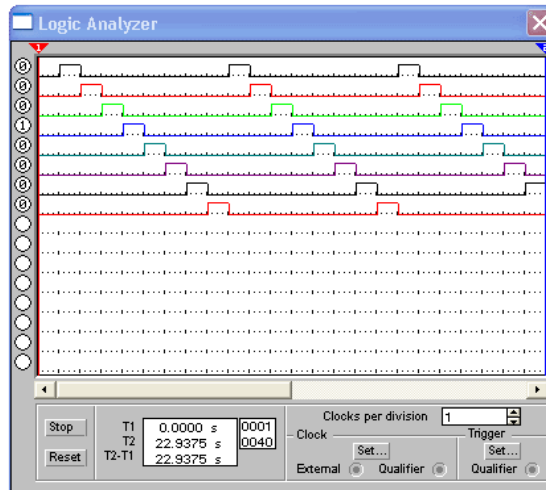


Рисунок 12.4. Диаграмма работы дешифратора

3.3 Анализ схемы включения дешифратора 74154 (отечественный аналог К155ИДЗ).

ИМС 74154 имеет четыре адресных входа A, B, C, D, два входа разрешения G1, G2 и шестнадцать выходов 0...15 (выходы не прямые, а инверсные, т.е. в исходном состоянии на выходах сигнал логической единицы).

В режиме дешифратора с генератора слова на входы G1, G2 подается 0, а на адресные входы — код в диапазоне 0000...1111.

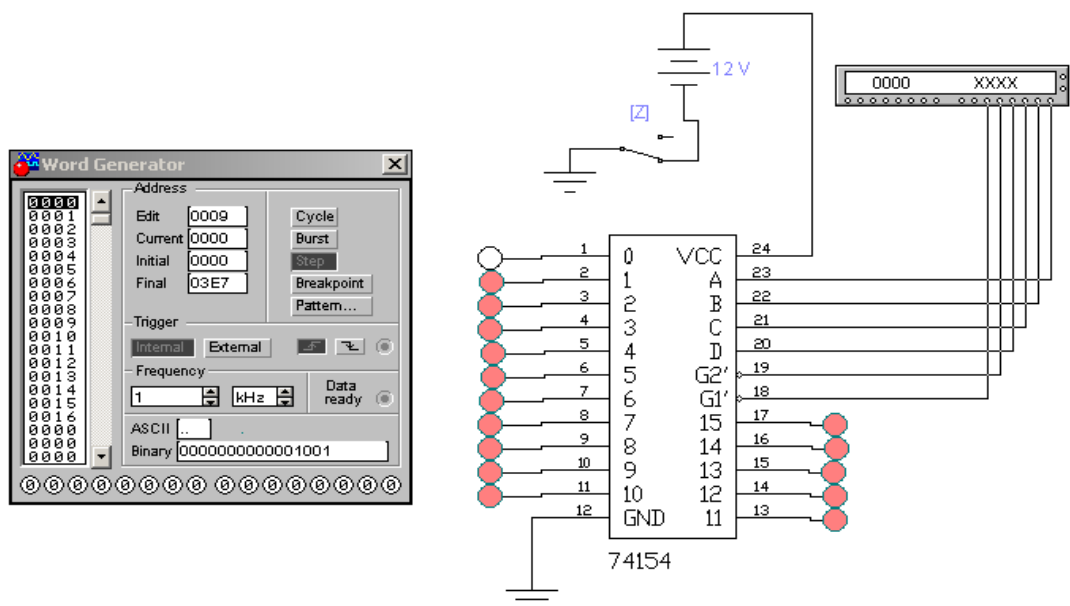


Рисунок 12.5. Анализ работы дешифратора 74154

3.2.2 Исследование работы шифратора приоритетного:

Создайте предложенную ниже схему шифратора:

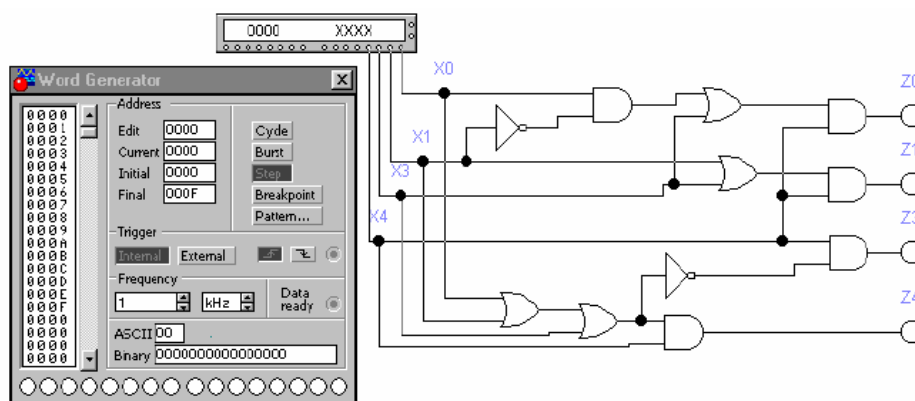


Рисунок 12.6. Схема дешифратора

Построить таблицу истинности для схемы:

Таблица 12.3. Таблица истинности схемы работы шифратора приоритетного

X ₀	X ₁	X ₂	X ₃	Z ₀	Z ₁	Z ₂	Z ₃
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

4 Контрольные вопросы.(к пункту 3.2.2)

- Какие функции реализует данное устройство?
- Назначение входов X₁, X₂, X₃, X₄?
- Назначение выходов Z₁, Z₂, Z₃, Z₄?
- Какие кодовые слова формируются на выходах Z₀ и Z₁, их физический смысл?
- Какой сигнал формируется на выходе Z₂, Z₃, их физический смысл?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.

- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Таблица истинности схемы работы шифратора приоритетного
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №13.
Арифметико-логическое устройство (АЛУ)
Объем учебного времени – 2ч

Методические рекомендации

1. Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.
- Ознакомление с возможностями моделирования работы арифметико-логического устройства (АЛУ).
- Исследование работы АЛУ на примере выполнения арифметических и логических операций.

2. Перечень необходимых средств обучения:

2.1 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.2 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Арифметико-логическое устройство предназначено для выполнения арифметических и логических операций над многоразрядными операндами в зависимости от кодов, подаваемых на управляющие входы.

В вычислительных устройствах АЛУ является базовым узлом и работает в сочетании с ОЗУ, регистрами сдвига, регистрами общего назначения и др. Микросхемы АЛУ, принадлежащие к разным видам логик, функционально во многом совпадают. Аналогом микросхемы K155ИПЗ в программе EWB имеется микросхема 74181. Она представляет собой четырехразрядное АЛУ.

АЛУ работает в режиме выполнения логических операций при значении управляющего сигнала $M=1$ и в режиме выполнения арифметических операций при значении управляющего сигнала $M=0$.

Эта микросхема обеспечивает 32 режима работы АЛУ в зависимости от управляющих сигналов на входах M , S_0 - S_3 . Возможные режимы задаются путем подачи сигналов на входы управления S_0 , S_1 , S_2 , S_3 . Если сигнал на входе M равен 0, то выполняются 16 арифметических операций (16 комбинаций сигналов S_0 ... S_3) с учетом переноса по входу CN (если $CN=0$) или без учета переноса (если $CN=1$). При сигнале на входе M равном 1, выполняются 16 логических операций, задаваемых S_0 - S_3 .

Четырехразрядные операнды A и B задаются на входах A_0 - A_3 и B_0 - B_3 соответственно. Результат арифметической или логической операции появляется на выходах F_0 - F_3 .

В микросхеме предусмотрены выходы переноса CN+4, ускоренного переноса P, ускоренного группового переноса G, равенства операндов A=B. Питание микросхемы осуществляется подключением источника питания 5 В ко входу VCC и заземлением входа GND.

В приведенной таблице содержатся выполняемые логические и арифметические операции в зависимости от кодовой комбинации на управляющих входах S0, S1, S2, S3.

Таблица 13.1 Логические и арифметические операции

Выбор функции				Логические функции M=1	Арифметические функции, M=0	
S3	S2	S1	S0		С переносом CN=0	Без переноса CN=1
0	0	0	0	Not A	A + 1	A
0	0	0	1	not (A or B)	A + B + 1	A or B
0	0	1	0	(Not A) and B	A + (not B) + 1	A or (not B)
0	0	1	1	Логический 0	0	-1
0	1	0	0	Not (A and B)	A + Not (A and B) + 1	A + (A and (not B))
0	1	0	1	Not B	(A or B) + (A and (not B)) + 1	(A+B) + (A and (not B))
0	1	1	0	A xor B	A - B	A-B-1
0	1	1	1	A and (not B)	Not (A and B)	(A and (not B)) - 1
1	0	0	0	(Not A) or B	A + B + 1	A+(A and B)
1	0	0	1	Not (A xor B)	A + B + 1	A + B
1	0	1	0	B	(A or (not B)) + (A and B) + 1	(A or (not B)) + A and B
1	0	1	1	A and B	A and B	A and B - 1
1	1	0	0	Логическая 1	A + A + 1	A + A
1	1	0	1	A or (not B)	(A or B) + A + 1	(A or B) + A
1	1	1	0	A or B	(A or (not B)) + A + 1	(A or (not B)) + A
1	1	1	1	A	A	A - 1

Используемые элементы EWB.

На панели Digital ICs выбирается схема из множества 741хх. Под номером 74181 - Четырёх-разрядное АЛУ.

3.1 Задание к работе

3.1.1 Построение АЛУ

- Подготовить микросхему АЛУ на рабочем столе.

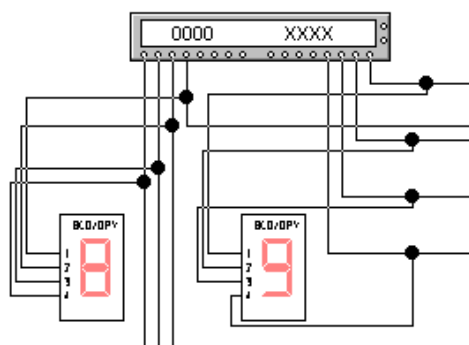


Рисунок 13.1 Фрагмент АЛУ

- Подготовить 4 переключателя для задания режимов управления.
- Сигналы S подаются на входы S0.....S3 АЛУ.
- Ещё один переключатель используется для задания режима M.

- И в том и другом случае логическая 1 подаётся от 5V источника постоянного тока, логический 0 от заземления.
- Шестой переключатель осуществляет роль входного переноса и подключается таким же образом к входу CN.
- Значения четырёхразрядных операндов А и В задаются с помощью генератора слова и в шестнадцатеричном коде отображаются на алфавитно-цифровых индикаторах и подсоединяются к входам A0....A3 и B0.....B3.
- На выходах F0....F3 формируется результат операции АЛУ. Для отображения результата к выходам F0....F3 присоединяется алфавитно-цифровой индикатор.
- К выходу VCC присоединить источник 5V.
- К входу GND подсоединить заземление.
- При коде 1111 на выходах F и при равенстве операндов выход A=B переводится в единичное состояние. Поскольку этот выход представляет собой каскад с открытым коллектором, то на него подаётся питание +5 вольт через резистор 1 кОм. Выход A=B совместно с выходом переноса CN+4 и выходом P подтверждения переноса используются для формирования признаков A>B и A<B с помощью дополнительных логических элементов ИЛИ-НЕ и НЕ.
- Изменяя состояния сигналов на управляющих входах по приведённой таблице, можно промоделировать большинство функций АЛУ, используемых в микропроцессорах.

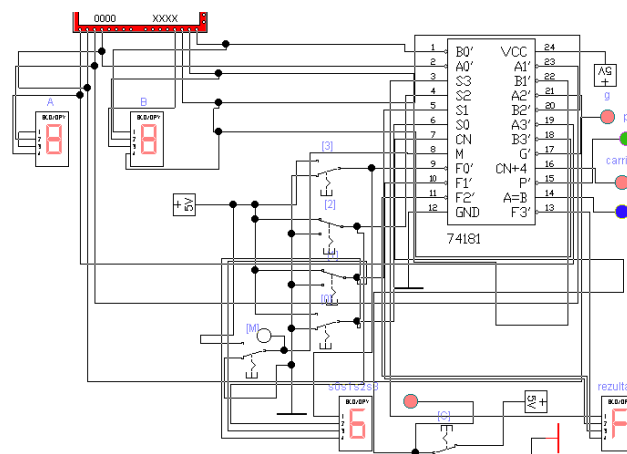


Рисунок 13.2 АЛУ

3.2 Проведите моделирование в 5 режимах.

4 Контрольные вопросы.

- Назначение входа переноса в АЛУ.
- Чем отличаются логические операции от арифметических операций?
- Как можно выполнить операцию инкремента?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Схема работы АЛУ

- Моделирование работы логических и арифметических операций.
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №14.
Изучение регистров и принципов их работы
Объём учебного времени – 4ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.
- Изучение принципа работы схем триггерных регистров
- Приобретение практических навыков в выполнении микроопераций на регистрах в статическом режиме.

2.Перечень необходимых средств обучения:

2.1 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.2 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Регистры предназначены для хранения и преобразования многоразрядных двоичных чисел. Для запоминания отдельных разрядов числа могут применяться триггеры различных типов. Одиночный триггер можно считать одnorазрядным регистром.

Занесение информации в регистр называется операцией записи. Операция выдачи информации из регистра – считывание. Перед записью информации в регистр, его необходимо обнулить.

Классификация регистров:

- 1 по способу ввода/вывода информации:
 - параллельные (регистры хранения) – информация вводится и выводится одновременно по всем разрядам;
 - последовательные (регистры сдвига) – информация бит за битом «проталкивается» через регистр и выводится также последовательно;
 - комбинированные – параллельный ввод и последовательный вывод (и наоборот).
- 2 по способу представления информации:
 - однофазные – информация представляется в прямом или обратном (инверсном) виде;
 - парафазные – информация представляется и в прямом, и в обратном виде.

Параллельные регистры осуществляют прием и выдачу информации в параллельном коде, а это значит, что для передачи каждого разряда используется отдельная линия.

Для записи информации в регистр на его входных выводах (D0-D3) нужно установить логические уровни, после чего на вход синхронизации (C) подать разрешающий импульс — логическую единицу. После этого на выходах Q0-Q3 появится записанное слово. Регистры

запоминают входные сигналы только в момент времени, определяемый сигналом синхронизации.

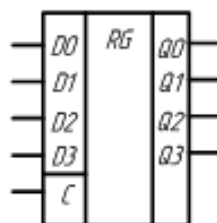


Рисунок 14.1 – Условно-графическое обозначение параллельного регистра

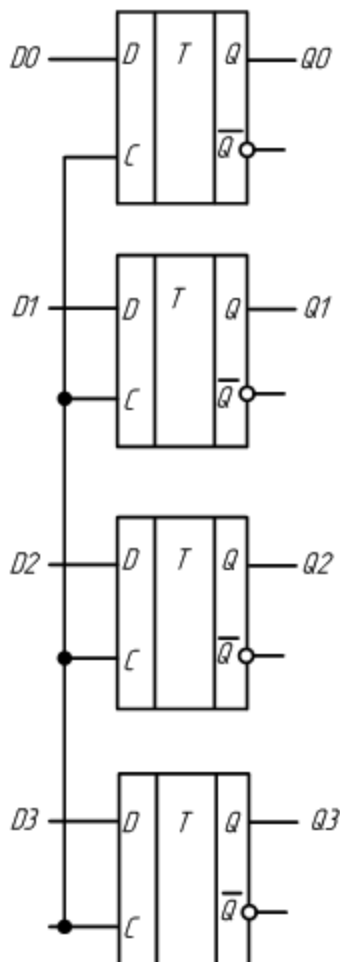


Рисунок 14.2 – Схема параллельного регистра

Кроме параллельного соединения триггеров для построения регистров используются последовательное соединение этих элементов.

Последовательный регистр (регистр сдвига) обычно служит для преобразования последовательного кода в параллельный и наоборот. Применение последовательного кода связано с необходимостью передачи большого количества двоичной информации по ограниченному количеству соединительных линий. При параллельной передаче разрядов требуется большое количество соединительных проводников. Если двоичные разряды последовательно бит за битом передавать по одному проводнику, то можно значительно сократить размеры соединительных линий на плате (и размеры корпусов микросхем).

Принципиальная схема последовательного регистра, собранного на основе D-триггеров и позволяющего осуществить преобразование последовательного кода в параллельный, приведена на рисунке 14.3.

Рассмотрим работу этого регистра.

Можно предположить, что в начале все триггеры регистра находятся в состоянии логического нуля, т.е. $Q0=0$, $Q1=0$, $Q2=0$, $Q3=0$. Если на входе D-триггера Т1 имеет место логический 0, то поступление синхроимпульсов на входы «С» триггеров не меняет их состояния.

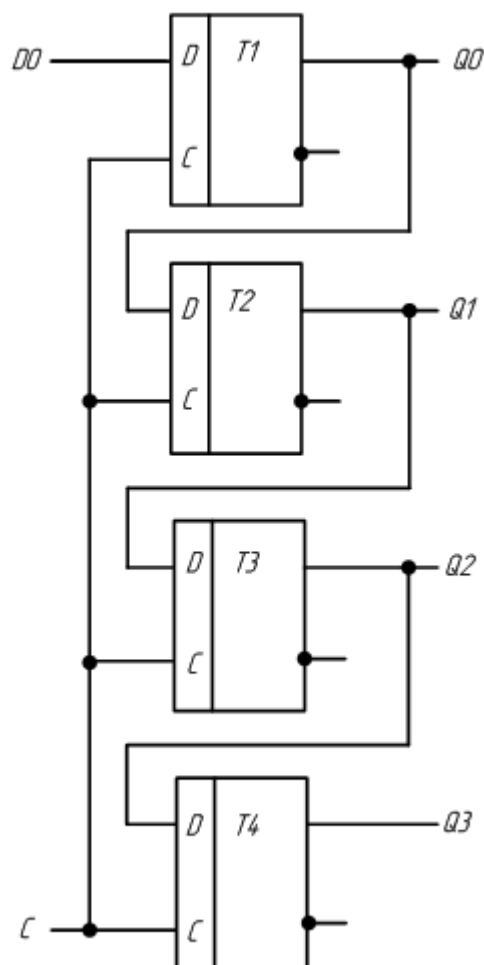


Рисунок 2.3 – Схема последовательного регистра

3.2 Задание к работе

3.2.1 Исследовать параллельный регистр

Записать целые десятичные числа от 0 до 15 в двоичной системе счисления в регистр и считать их. Заполнить таблицу 14.1.

Таблица 14.1 – Коды, записанные в параллельный регистр

Записываемое десятичное число	Считанное из регистра двоичное число
0	
1	
...	
15	

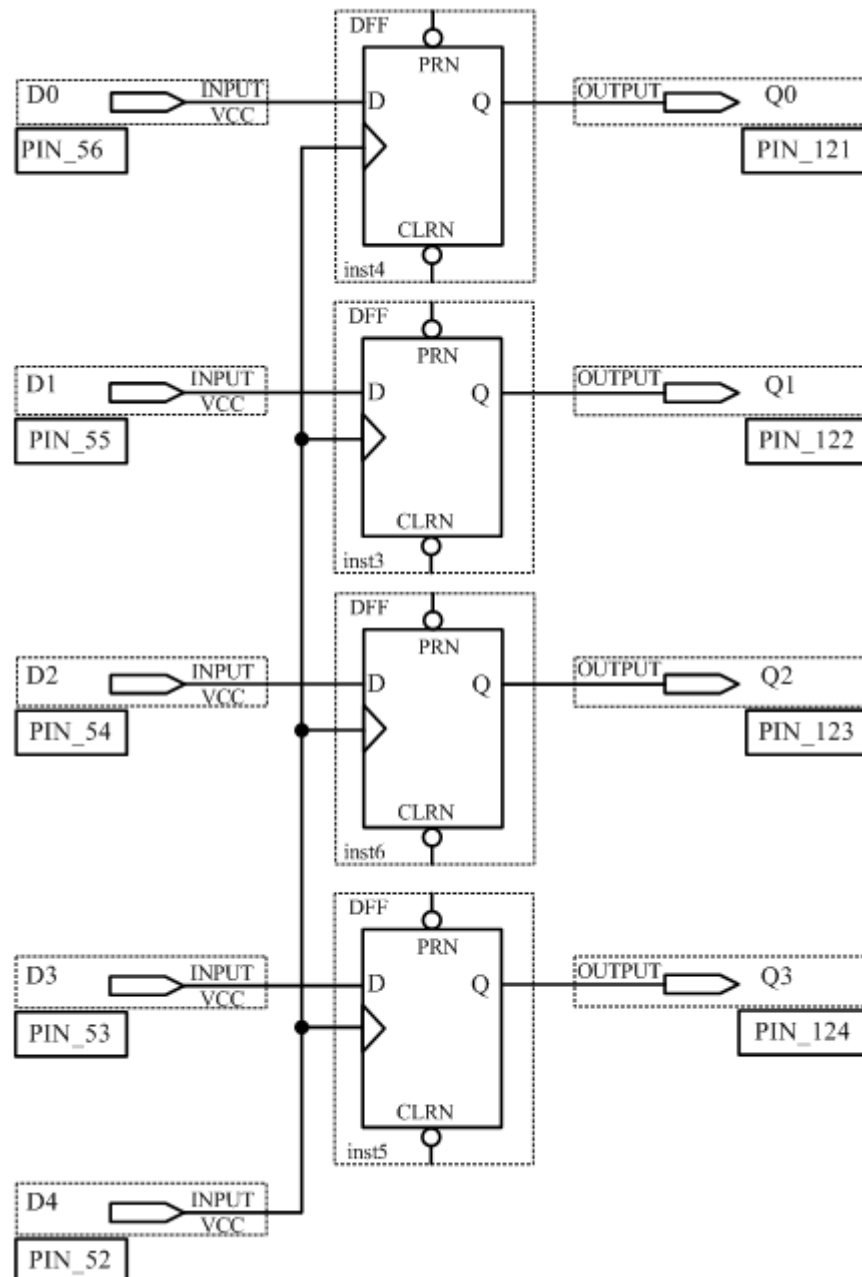


Рисунок 14.4 – Схема 4-х битного параллельного регистра

4 Контрольные вопросы.

- Назначение регистров.
- По каким признакам классифицируются регистры?
- Чем определяется разрядность регистров?
- Назначение параллельного регистра.
- Объяснить принцип работы последовательного регистра. Что такое функция перехода?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.

- Описание основных этапов работы
- Результаты выполнения работы;
- Схема 4-х битного параллельного регистра
- Таблица кодов
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №15.

Изучение принципов работы микрокомпьютера на примере 8051: (Управляющие под-программы; Команды передачи данных; Порты ввода-вывода; Арифметические и логические операции; Операции передачи управления)

Объём учебного времени – 2ч

Методические рекомендации

1.Цель работы

- Изучение общих принципов структурной организации ЭВМ
- Изучение центрального процессора
- Изучение алгоритма выполнения процессором команд.

2.Перечень необходимых средств обучения:

2.3 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.4 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Общая структурная схема микрокомпьютера представлена на рисунке 15.1.

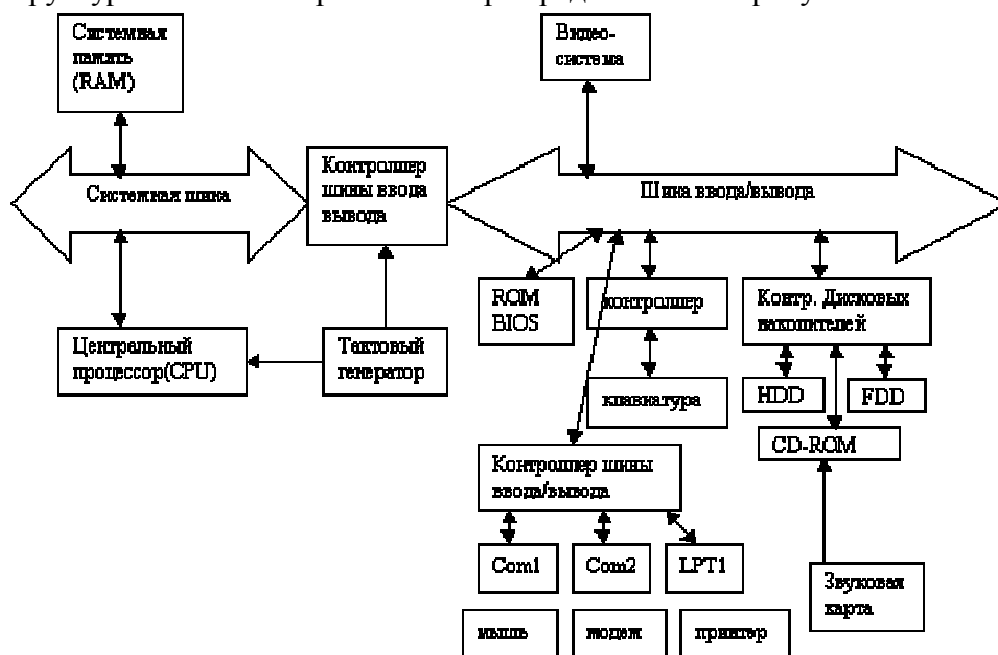


Рисунок 15.1 – Общая структурная схема микрокомпьютера

Микропроцессор. Он является главным компонентом микро-компьютера. Характеристики микропроцессора – длина разрядной сетки (или разрядность слова), набор выполняемых команд, быстродействие и другие – в основном определяют характеристики всего микрокомпьютера. Микро-процессор, называемый иногда *центральным процессором* (ЦП) микрокомпьютера, выполняет следующие функции: управление и координация работы всех других компонентов микрокомпьютера; выборка команд и обрабатываемых данных из основной памяти; декодирование команд; выполнение с помощью арифметическо-логического устройства (АЛУ) арифметических, логических и других операций, закодированных в командах; передача данных между микропроцессором и основной памятью, а также между микропроцессором и устройствами ввода-вывода; обработка сигналов от устройств ввода-вывода, в том числе обработка сигналов прерывания с этих устройств. Микропроцессор – весьма сложное цифровое электронное устройство. Приступая к написанию программ, программист должен уяснить в основном лишь те его элементы, которые отражены в программах или влияют на временные характеристики выполнения программы. К этим элементам относятся: форматы и системы команд микропроцессора; длительность выполнения разных команд; имена (или номера) программно-доступных регистров (так называются регистры, которые могут использоваться в составляемых программах); длина разрядной сетки (разрядность); правила адресации внешних устройств и особенности выполнения операций ввода-вывода; размер адресного пространства; схема обработки прерываний.

Перечисленные элементы образуют основу архитектуры микропроцессора и в совокупности представляют собой его модель с точки зрения программиста. Для разных типов МП существует своя модель.

На рисунке 15.2 представлены важнейшие компоненты микропроцессора, а также его связь с основной памятью при помощи трех магистралей данных, адресов и управления. В состав МП входят устройство управления (УУ), арифметическо-логическое устройство (АЛУ) и набор регистров.

Устройство управления предназначено для управления работой всех компонентов микрокомпьютера и обеспечения должного взаимодействия различных компонентов друг с другом. Управление осуществляется с помощью импульсных сигналов, посылаемых УУ на соответствующие входы управляемых компонентов. Кроме того, УУ может получать ответные сигналы с управляемых компонентов.

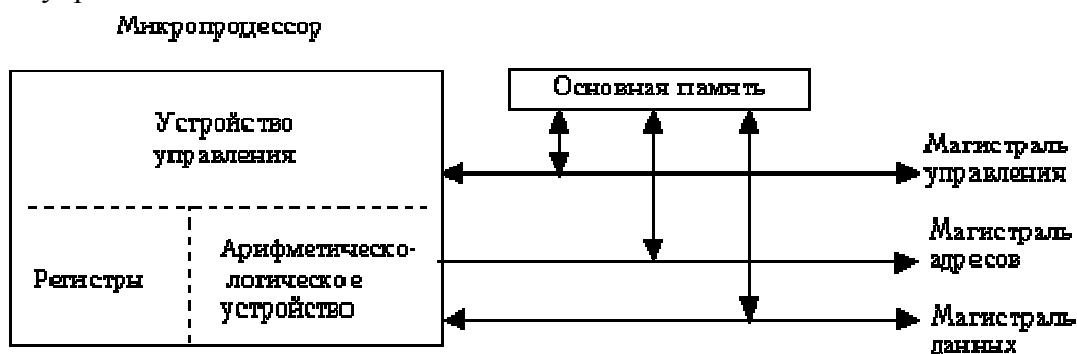


Рисунок 15.2 – Микропроцессор и его связи с основной памятью

Физически УУ представляет собой цифровую электронную схему, на вход которой поступают коды подлежащих выполнению операций, а входом являются серии управляющих сигналов. Восприняв код той или иной операции, УУ формирует цепочку управляющих сигналов и подает их в нужные точки микрокомпьютера.

Арифметическо-логическое устройство предназначено для исполнения арифметических и логических операций. Основу АЛУ составляет *операционный блок* – цифровое электронное устройство, которое может настраиваться на различные операции и непосредствен-

но осуществлять их. Настройка операционного блока на конкретную операцию и последовательность шагов ее выполнения обеспечиваются с помощью управляющих сигналов от УУ.

Регистры являются важными элементами микропроцессора. *Регистр* – это электронное цифровое устройство для временного запоминания информации в форме двоичного числа или кода. Запоминающим элементом в регистре является триггер, который может находиться в одном из двух состояний. Одно из этих состояний соответствует запоминанию двоичного нуля, а другое – запоминанию двоичной единицы. В общем случае регистр содержит несколько связанных друг с другом триггеров – по одному триггеру на каждый разряд запоминаемого двоичного числа. Число триггеров в регистре называется *разрядностью* регистра. Например, регистр из восьми триггеров – это 8-разрядный или 8-битовый регистр (так как каждый разряд регистра обеспечивает хранение одного бита информации).

Многие регистры специализированы по своей функции. Так, существуют регистр-аккумулятор или просто аккумулятор, программный счетчик, регистр команд, регистр адреса памяти и т.д. Аккумулятор входит в АЛУ и предназначен для хранения одного из операндов перед выполнением операции в АЛУ или для кратковременного запоминания результата операции. Операнд – это данное, используемое в текущей операции. Например, в операции суммирования операндами являются оба слагаемых.

Программный счетчик (счетчик команд, регистр адреса команды) служит для формирования и запоминания адреса очередной выполняемой команды. После выполнения каждой команды программный счетчик содержит адрес следующей команды, по которому эта команда хранится в памяти микрокомпьютера.

Регистр команд используется для хранения кода текущей выполняемой команды. Входящий в состав команды код операции используется, как уже говорилось, для формирования в УУ определенной серии управляющих сигналов, зависящей от конкретного кода операции. Оставшаяся часть кода команды может содержать информацию об адресах операндов, участвующих в выполнении данной команды.

Регистр адреса памяти служит для запоминания адреса кода команды, операнда или результата операции во время извлечения (чтения) команды или операнда из памяти или записи результата операций в память. Регистр адреса памяти может входить не в состав МП, а в состав элементов памяти микрокомпьютера.

Изменить роль специализированных регистров или даже узнать их содержимое программным путем нельзя, т.е. эти регистры, как говорят, программно-недоступны. Но в состав МП входят и регистры, которые программист может использовать в своей программе. Такие регистры микропроцессора называются *программно-доступными*. Состав и назначение их различны в разных типах микропроцессоров. Однако среди них почти всегда имеются регистр слова состояния процессора (РССП) и несколько регистров общего назначения (РОН).

Регистр слова состояния процессора хранит слово состояния процессора (ССП), отражающее информацию о состоянии МП и выполняемой им программы в каждый данный момент времени.

Регистры общего назначения обычно не имеют конкретного функционального назначения. Программист может в своей программе задействовать их так, как он считает нужным. Чтобы отличить РОНЫ друг от друга им присвоены *уникальные имена (или номера)*, которые и записываются в программе.

Основная память. Микропроцессор может обрабатывать только те данные, которые находятся в основной памяти. Основная память обычно состоит из двух частей – ОЗУ и ПЗУ (смотри рисунок 15.1).

Оперативное запоминающее устройство обеспечивает чтение находящихся в нем данных и запись в него новых данных. В микрокомпьютерах ОЗУ обычно реализуется как *энергозависимая память*, т.е. такая память, содержимое которой разрушается (“стирается”) при выключении микрокомпьютера.

Постоянное запоминающее устройство обеспечивает только чтение данных, которые однажды были записаны в ПЗУ. Таким образом, содержимое ПЗУ не может быть изменено

микропроцессором, оно постоянно (отсюда и название этого вида памяти). Это устройство создается как энергонезависимая память: ее содержимое не “стирается” при выключении питания микрокомпьютера. Запись нужных данных в ПЗУ осуществляется на специальных устройствах, вне микрокомпьютера. В ПЗУ помещают обычно некоторые особо важные или не подлежащие изменению программы и разнообразные константы.

Интерфейсы для устройств ввода (ИВВ) и вывода (ИВЫВ). Они представляют собой блоки сопряжения с устройствами соответственно ввода и вывода. Не останавливаясь пока на описании устройств ввода-вывода, отметим лишь, что термин “интерфейс” означает приблизительно “сопряжение двух устройств друг с другом с помощью аппаратных и программных средств”. Необходимость интерфейсов, или, как их еще называют, *интерфейсных блоков (контроллеров)*, вызывается тем, что УВВ и УВЫВ (например, дисплей, печатающее устройство, НГМЛ и т.д.) нельзя непосредственно подключать к микрокомпьютеру. Одной из причин этого является то, что количество и характер сигналов, передаваемых и принимаемых по системной магистрали, с которой связаны все компоненты микрокомпьютера, как правило, отличаются от количества и типа сигналов, формируемых или воспринимаемых устройством ввода-вывода. Соответствующий интерфейсный блок обеспечивает должное согласование сигналов системной магистрали и устройства ввода-вывода.

Системная магистраль (СМ). Обычно она содержит магистрали адресов (МА), данных (МД) и управления (МУ). Каждая из них состоит из набора проводников, по которым МП передает или принимает определенные электрические сигналы. Магистраль адреса предназначена для цифрового адреса ячейки или внешнего устройства, магистраль данных – для передачи и приема данных (например, передача информации из МП в ОЗУ происходит по МД). Магистраль управления используется для передачи сигналов управления, которые сопровождает любую передачу адреса или данных.

Источник питания (ИП). Обычно ИП формирует выпрямленные стабилизированные напряжения +12, –5 и +5 В. Этот набор напряжений может изменяться для различных классов микрокомпьютеров.

Структура микрокомпьютера. На рисунке 15.3 дана структурная схема, отражающая важнейшие компоненты МП, ОП и связи компонентов друг с другом с помощью магистралей данных, адреса и управления.

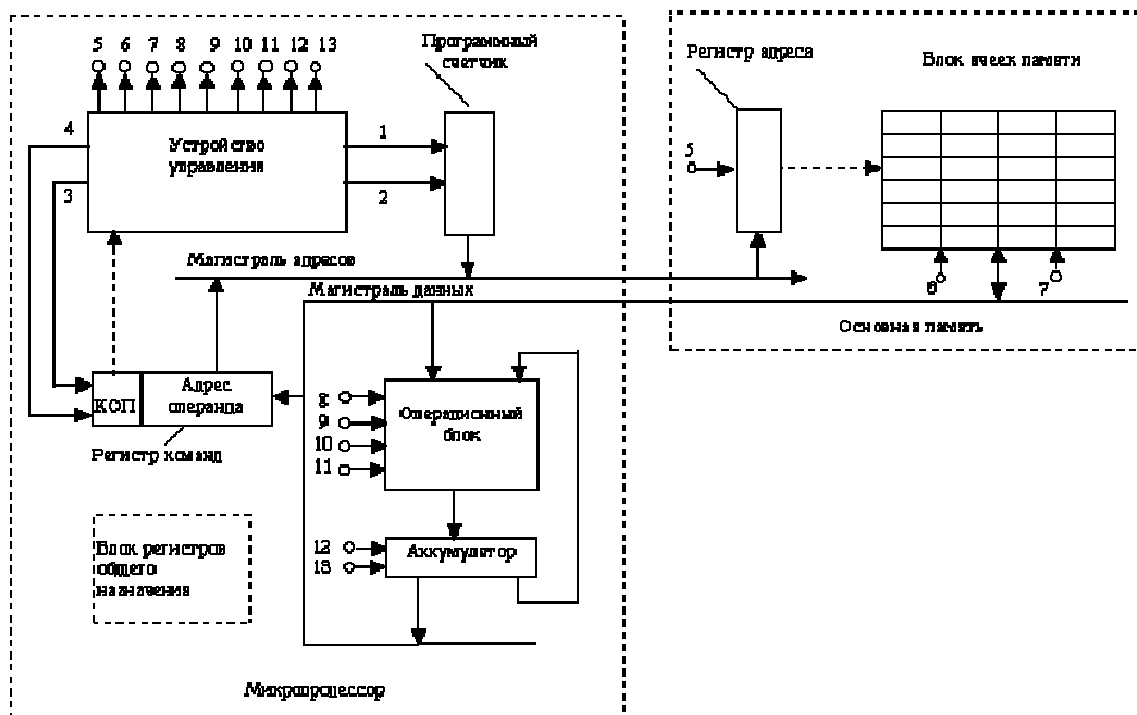


Рисунок 15.3 – Развернутая структурная схема микрокомпьютера (без устройств ввода-вывода)

Основная память представлена в совокупности блока ячеек памяти и регистра адреса. Процесс чтения (выборки) информации из ячейки или записи (занесения) информации в ячейку называется *доступом* к памяти. Для доступа к памяти предварительно по магистрали адреса в регистр адреса должен быть помещен адрес той ячейки, к которой производится доступ.

В состав МП входит генератор тактовой частоты (на рисунке 1.3 не показан). Он предназначен для синхронизации (т.е. согласования во времени) работы компонентов микрокомпьютера. Генератор формирует периодическую последовательность импульсов с частотой от нескольких сотен килогерц до нескольких мегагерц в зависимости от типа микропроцессора. Напомним, что *герц* – это единица измерения частоты колебаний, равная одному колебанию в секунду.

Регистры

Регистры процессора можно разделить на следующие группы:

- регистры общего назначения;
- сегментные регистры;
- регистр указателя команд и флаговый регистр;
- регистры дескрипторов;
- управляющие регистры;
- регистры системных адресов;
- отладочные регистры;
- тестовые регистры.

Регистры общего назначения являются 32 разрядными. При этом для сохранения совместимости с младшими моделями процессор 80386 поддерживает структуру регистровых файлов процессоров 8086 и 80286, действительны все наименования, применявшиеся для обозначения 16- и 8-разрядных регистров. Например, для процессора 80386 все следующие операторы являются допустимыми:

```
MOV EAX,0
MOV AX,0
MOV AH,0
MOV AL,0
```

31	16 15	8 7	0	
	AH	A	X	AL
	BH	B	X	BL
	CH	C	X	CL
	DH	D	X	DL
	SI			EAX
	DI			EBX
	BP			ECX
	SP			EDX
				ESI
				EDI
				EBP
				ESP

Флаговый регистр имеет 32-разрядную структуру, представленную на рисунке.

Резерв	VM	RF	x	NT	IOPL	OF	DF	IF	TF	SF	ZF	x	AX	x	PF	x	CF
--------	----	----	---	----	------	----	----	----	----	----	----	---	----	---	----	---	----

Флаговый регистр EFlags.

Флаги регистра EFlags используются следующим образом:

- OF (Overflow) - флаг переполнения. Указывает, что в результате выполнения арифметической команды возникло переполнение старшего разряда результата;
- DF (Direction) - флаг направления. Определяет, в каком направлении будут пересылаться данные строковыми командами - от младших адресов к старшим или наоборот;
- IF (Interrupt) - флаг разрешения прерываний. Указывает на возможность внешних прерываний;
- TF (Trap) - флаг пошагового выполнения. Устанавливает выполнение команд процессора в пошаговом режиме (для отладки программ);
- SF (Sign) - знаковый флаг. Содержит знак результата после выполнения арифметических операций (0 = '+', 1 = '-');
- ZF (Zero) - флаг нуля. Содержит 1, если результат арифметической операции или операции сравнения ненулевой, в противном случае содержит 0;
- AF (Auxiliary) - дополнительный флаг переноса. Содержит значение переноса из 3-го бита восьмибитовых данных;
- PF (Parity) - флаг контроля четности. Содержит 1, если сумма единиц в восьми младших разрядах регистра является четным числом, или 0, если она нечетна;
- CF (Carry) - флаг переноса. Используется арифметическими командами и командами сдвигов;
- VM (Virtual Mode) - при VM=1 указывает на включение виртуального 8086 в защищенном режиме процессора 80386;
- RF (Resumption Flag) - используется совместно с отладочными регистрами контрольных точек или пошагового режима. При RF=1 ошибки, возникшие во время отладки при исполнении команды, игнорируются до выполнения следующей команды;
- NT (Nested Task) - признак вложенности задач; NT=1 указывает, что текущая задача является вложенной по отношению к другой задаче;
- IOPL (Input / Output Privilege Level) - два бита, содержимое которых указывает на уровень текущего приоритета (от 0 до 3).

Дескрипторные регистры не могут использоваться программно. Каждый дескриптор связан с одним из шести селекторов и содержит 32 - байтовый базовый адрес сегмента, 32-байтовый размер сегмента и его атрибуты.

В IBM-совместимых ПК применяются процессоры (CPU — Central Processor Unit), совместимые с семейством 80x86 фирмы. В оригинальной IBM PC использовался процессор 8088 с 16-разрядными регистрами. Все старшие модели процессоров, в том числе 32-разрядные (386, 486, Pentium, Pentium Pro) и с 64-разрядным расширением MMX, включают в себя подмножество системы команд и архитектуры нижестоящих моделей, обеспечивая совместимость с ранее написанным ПО.

Процессоры класса Pentium

Процессоры Pentium фирмы Intel представляют пятое поколение процессоров семейства 80x86. По базовой регистровой архитектуре и системе команд они совместимы с 32-битными процессорами, но имеют 64-битную шину данных, благодаря чему их иногда ошибочно называют 64-разрядными. По сравнению с предыдущими поколениями процессоры Pentium имеют следующие качественные отличия:

- Суперскалярная архитектура: процессор имеет два параллельно работающих конвейера обработки (U-конвейер с полным набором и V-конвейер с несколько ограниченным набором инструкций), благодаря чему он способен одновременно выполнять две инструкции. Однако преимущества этой архитектуры полностью реализуются только при специальном режиме компиляции ПО;
- Применение технологии динамического предсказания ветвлений совместно с выделенным внутренним кэшем команд объемом 8 Кбайт обеспечивает максимальную загрузку конвейеров.

- Применение технологии динамического предсказания ветвлений совместно с выделенным внутренним кэшем команд объемом 8 Кбайт обеспечивает максимальную загрузку конвейеров.

- Внутренний (Level 1) кэш данных объемом 8 Кбайт в отличие от 486 работает с отложенной (до освобождения внешней шины) записью и настраивается на режим сквозной или обратной записи, поддерживая протокол MESI.

- Внешняя шина данных ради повышения производительности имеет разрядность 64 бит, что требует соответствующей организации памяти.

- Встроенный сопроцессор за счет архитектурных улучшений (конвейеризации) в 2-10 раз превосходит FPU-486 по производительности.

- Введено несколько новых инструкций, в том числе распознавание семейства и модели CPU.

- Применено выявление ошибок внутренних устройств (внутренний контроль паритета) и внешнего интерфейса шины, контролируется паритет шины адреса.

- Введена возможность построения функционально избыточной двухпроцессорной системы.

- Реализован интерфейс построения двухпроцессорных систем с симметричной архитектурой (начиная со второго поколения Pentium).

- Введены средства управления энергопотреблением.

- Применена конвейерная адресация шинных циклов.

- Сокращено время (количество тактов) выполнения инструкций.

- Введена трассировка инструкций и мониторинг производительности.

- Расширены возможности виртуального режима — введена виртуализация флага прерываний.

- Введена возможность оперирования страницами размером 4 Мб (вместо 4 Кб) в режиме страничной переадресации (Paging).

Все Pentium-процессоры имеют средства SMM, возможности которых расширились по мере появления новых моделей.

Средства тестирования включают возможность выполнения встроенного теста BIST (Built-in Self Test), обеспечивающего выявление ошибок микрокодов, программируемых логических матриц, тестирование командной кэш-памяти, кэш-памяти хранения данных, буфера быстрой переадресации и ROM. Все процессоры имеют стандартный тестовый порт IEEE 1149.1, позволяющий тестировать процессор с помощью интерфейса JTAG.

Процессоры Pentium MMX (P55C) — новое поколение процессоров, основанное на MMX-технологии, которая ориентирована на мультимедийное, 2D - и SD-графическое и коммуникационное применение. В логическую архитектуру Pentium введены восемь 64-битных регистров, 4 новых типа данных и 57 дополнительных мнемоник инструкций для одновременной обработки нескольких единиц данных SIMD (Single Instruction Multiple Data). Одновременно обрабатываемое 64-битное слово может содержать как одну единицу обработки, так и 8 однобайтных, 4 двухбайтных или 2 четырехбайтных операнда. В остальных командах обеспечивается совместимость с Pentium. На самом деле, регистры MMX физически расположены в стеке регистров FPU, так что новых регистров этот процессор не предоставляет, и чередование использования программой инструкций FPU и MMX приводит к снижению эффективности работы, связанному с необходимостью пересылок данных из стека в память и обратно. Эффективность MMX вызывает некоторые сомнения, поскольку те функции, для которых они целесообразны, с успехом выполняются акселераторами графических карт, которые стали уже обыденными.

Интерфейс шины процессоров Pentium

По интерфейсу и составу сигналов шина процессора Pentium похожа на шину 486, но имеет значительные отличия. Новые особенности направлены на поддержку политики обратной записи кэша, повышение производительности и обеспечение дополнительных функциональных возможностей.

Шина данных - 64-битная, для повышения производительности обмена с памятью.

Расширения архитектуры

Процессоры Pentium (и более старшие) имеют ряд расширений относительно базовой архитектуры 32-разрядных процессоров и ее развития в процессорах четвертого поколения, появляющихся по мере совершенствования моделей. Для возможности получения сведений о них в систему команд включена инструкция CPUID, позволяющая программно в любой момент времени (а не только сразу после сигнала RESET) получить сведения о классе, модели и архитектурных особенностях конкретного процессора.

В дополнение к базовой архитектуре 32-разрядных процессоров, Pentium имеет набор регистров, специфических для модели — MSR (Model Specific Registers). В их число входит группа тестовых регистров (TR1...TR12), средства мониторинга производительности, регистры-фиксаторы адреса и данных цикла, вызвавшего срабатывание контроля машинной ошибки. Название этой группы регистров указывает на возможную их несовместимость для разных классов (Pentium и Pentium Pro) и даже моделей процессоров. Программа, их использующая, должна опираться на сведения о процессоре, полученные с помощью инструкции CPUID.

Средства мониторинга производительности включают таймер реального времени и счетчики событий. Таймер TSC (Time Stamp Counter) представляет собой 64-битный счетчик, инкрементируемый с каждым тактом ядра процессора. Для чтения его содержимого предназначена инструкция RDTSC.

Счетчики событий CTRO, CTR1 разрядностью по 40 бит программируются на подсчет событий различных классов, связанных с шинными операциями, исполнением инструкций, событиями во внутренних узлах, связанных с работой конвейеров, кэша, контролем точек останова и т. п. Шести битные поля типов событий позволяют каждому из счетчиков независимо назначить подсчет событий из обширного списка. Состояние счетчиков может быть предустановлено и считано программно.

Тестовые регистры позволяют управлять большинством функциональных узлов процессора, обеспечивая возможность весьма подробного тестирования их работоспособности. С помощью бит регистра TR12 можно запретить новые архитектурные свойства (предсказание и трассировку ветвлений, параллельное выполнение инструкций), а также работу первичного кэша:

- Бит 0 — NBP (No Branch Prediction) — запрещает заполнение буфера BPB. При этом прежние вхождения продолжают действовать, для полного отключения предсказания необходимо загрузить регистр CR3 (это вызовет сброс таблицы ветвлений).
- Бит 1 — TR — разрешает формирование специального цикла сообщения о ветвлении.
- Бит 2 — SE (Single Pipe Execution) — запрещает работу второго конвейера (отменяет парное исполнение инструкций).
- Бит 3 — 3 (Cache Inhibit) - запрещает заполнение строк первичного кэша.
- Бит 9 — ITR (10 Trap Restart) — разрешает поддержку рестарта инструкций ввода/вывода при прерываниях SMI.

Двухпроцессорные системы

Процессоры Pentium, начиная со второго поколения, имеют специальные интерфейсные средства для построения двухпроцессорных систем. Интерфейс позволяет на одной локальной системной шине устанавливать два процессора, при этом почти все их одноименные выходы просто непосредственно объединяются. Целью объединения является либо использование симметричной мультипроцессорной обработки SMP (Symmetric Multi-Processing), либо построение функционально избыточных систем FRC (Functional Redundancy Checking).

В системе с SMP каждый процессор выполняет свою задачу, порученную ему операционной системой. Поддержку SMP имеют такие ОС, как Novell Net Ware, Windows NT, OS/2.

Процессоры Pentium Pro и Pentium II

От процессора Pentium Pro принято отсчитывать начало шестого поколения С точки зрения принципа организации вычислений, главное отличие поколения заключается в применении динамического исполнения, при котором внутри процессора инструкции могут выполняться не в том порядке (out of order), который предполагает программный код.

Процессор Pentium Pro (P6) относительно классического процессора Pentium имеет следующие усовершенствования:

- Применено динамическое исполнение — комбинация методов предсказания множественных ветвлений, анализа прохождения данных и виртуального выполнения. При этом команды, не зависящие от результатов предыдущих операций, могут выполняться в измененном порядке, но последовательность выгрузки результатов в память и порты будет соответствовать исходному программному коду.

- Применена архитектура двойной независимой шины, повышающая суммарную пропускную способность. Одна шина — системная — служит для общения ядра с основной памятью и интерфейсными устройствами, другая — внутренняя — предназначена исключительно для обмена со вторичным кэшем.

- В корпусе процессора интегрирован синхронный L2-кэш объемом 256-512 Кбайт, подключенный к внутренней шине.

- В систему команд введены инструкции условной пересылки данных, позволяющие сократить количество условных переходов.

Для повышения достоверности шина поддерживает ЕСС-контроль.

4 Контрольные вопросы.

- В чем особенности структурной организации IBM совместимых компьютеров?
- В чем отличие внутреннего представления положительных и отрицательных чисел?
- Поясните алгоритм выполнения команд процессором.
- Когда ЦП может начать программу обслуживания прерывания?
- Какой из регистров входит в состав АЛУ?
- Чем определяется разрядность регистра?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

– Глушаков С.В. Персональный компьютер: учеб. пособие для сред. проф. образования. - М.; Владимир: АСТ; ВКТ, 2008. - 475с.

– Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.

– Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.

– Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб. пособие для сред. проф. образования. - Ростов-н/Д: Феникс, 2007.- 448с.

– Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

1. <http://novtex.ru/IT> - журнал "Информационные технологии"
2. <http://infojournal.ru> - журнал «Информатика и образование»
3. <http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №16.

Изучение принципов работы микрокомпьютера на примере 8051: (Внешние прерывания; Таймеры и счетчики событий; Поиск неисправностей – тестирование; Использование клавиатуры и дисплея; Последовательная передача)

Объем учебного времени – 2ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.
- Изучение общих принципов структурной организации ЭВМ
- Изучение центрального процессора
- Изучение алгоритма выполнения процессором команд.

2.Перечень необходимых средств обучения:

2.5 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.6 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Большинство задач управления, которые реализуются с помощью МК, требуют исполнения их в реальном времени. Под этим понимается способность системы получить информацию о состоянии управляемого объекта, выполнить необходимые расчетные процедуры и выдать управляющие воздействия в течение интервала времени, достаточного для желаемого изменения состояния объекта.

Возлагать функции формирования управления в реальном масштабе времени только на центральный процессор неэффективно, так как это занимает ресурсы, необходимые для расчетных процедур. Поэтому в большинстве современных МК используется аппаратная поддержка работы в реальном времени с использованием таймера (таймеров).

Модуль таймера 8-разрядного МК представляет собой 8-ми или 16-разрядный счетчик со схемой управления. Схемотехникой МК обычно предусматривается возможность использования таймера в режиме счетчика внешних событий, поэтому его часто называют таймером/счетчиком. Структура типичного 16-разрядного таймера/счетчика в составе МК приведена на рис. 16.1

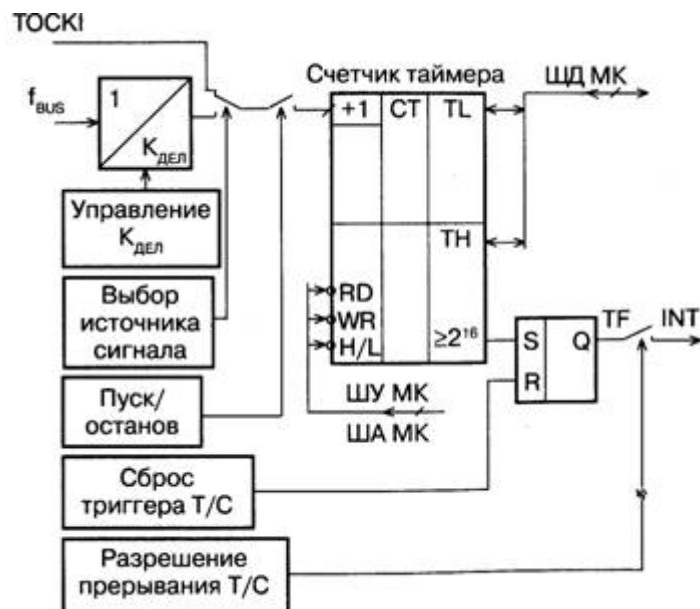


Рис. 16.1. Структура модуля таймера/счетчика

Модули таймеров служат для приема информации о времени наступления тех или иных событий от внешних датчиков событий, а также для формирования управляющих воздействий во времени.

В памяти МК 16-разрядный счетчик отображается двумя регистрами: ТН – старший байт счетчика, ТЛ – младший байт. Регистры доступны для чтения и для записи. Направление счета – только прямое, то есть при поступлении входных импульсов содержимое счетчика инкрементируется. В зависимости от настройки счетчик может использовать один из источников входных сигналов:

- импульсную последовательность с выхода управляемого делителя частоты f_{BUS} ;
- сигналы внешних событий, поступающие на вход ТОСКИ контроллера.

В первом случае говорят, что счетчик работает в режиме таймера, во втором – в режиме счетчика событий. При переполнении счетчика устанавливается в «единицу» триггер переполнения ТФ, который генерирует запрос на прерывание, если прерывания от таймера разрешены. Пуск и останов таймера могут осуществляться только под управлением программы. Программным способом можно также установить старший и младший биты счетчика в произвольное состояние или прочесть текущий код счетчика.

Рассмотренный «классический» модуль таймера/счетчика широко применяется в различных моделях относительно простых МК. Он может использоваться для измерения временных интервалов и формирования последовательности импульсов. Основными недостатками «классического» таймера/счетчика являются:

- потери времени на выполнение команд пуска и останова таймера, приводящие к появлению ошибки при измерении временных интервалов и ограничивающие минимальную длительность измеряемых интервалов времени единицами мс;
- сложности при формировании временных интервалов (меток времени), отличных от периода полного коэффициента счета, равного 216;
- невозможность одновременного обслуживания (измерения или формирования импульсного сигнала) сразу нескольких каналов.

Первые их двух перечисленных недостатков были устранены в усовершенствованном модуле таймера/счетчика, используемом в МК семейства MCS-51 (Intel). Дополнительная логика счетного входа позволяет тактовым импульсам поступать на вход счетчика, если уровень сигнала на одной из линий ввода равен «1». Такое решение повышает точность измерения временных интервалов, так как пуск и останов таймера производится аппаратно. Также в усовершенствованном таймере реализован режим перезагрузки счет-

чика произвольным кодом в момент переполнения. Это позволяет формировать временные последовательности с периодом, отличным от периода полного коэффициента счета.

Однако эти усовершенствования не устраняют главного недостатка модуля «классического» таймера – одноканального режима работы. Совершенствование подсистемы реального времени МК ведется по следующим направлениям:

- увеличение числа модулей таймеров/счетчиков. Этот путь характерен для фирм, выпускающих МК со структурой MCS-51, а также для МК компаний Mitsubishi и Hitachi;
- модификация структуры модуля таймера/счетчика, при которой увеличение числа каналов достигается не за счет увеличения числа счетчиков, а за счет введения дополнительных аппаратных средств входного захвата (input capture - IC) и выходного сравнения (output подход), используется, в частности, в МК компании Motorola.

Принцип действия канала входного захвата таймера/счетчика показан на рис. 16.2

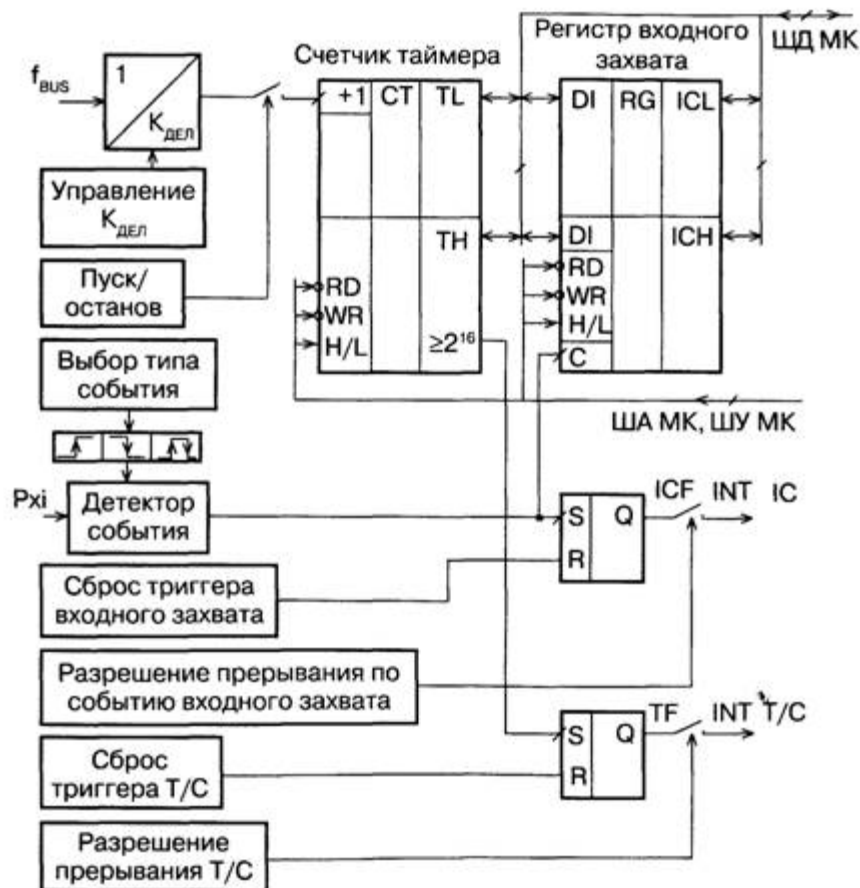


Рис. 16.2. Структурная схема канала входного захвата таймера

4 Контрольные вопросы.

- Какие типы двоичных счетчиков Вы знаете?
- Чем определяется быстродействие счетчика?
- Какие способы повышения быстродействия счетчиков Вы знаете?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Результаты тестирования работоспособности ПК средствами операционной системы.
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №17.

Изучение микроконтроллера Intel 8051 и принципов его работы:

(Сигналы микрокомпьютера; Программируемый таймер; Периферийный интерфейс; Сигналы аналоговых устройств)

Объём учебного времени – 2ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.
- Изучение общих принципов структурной организации ЭВМ
- изучение архитектуры микроконтроллеров

2.Перечень необходимых средств обучения:

2.7 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.8 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

В микропроцессорной технике выделился самостоятельный класс интегральных схем – микроконтроллеры, которые предназначены для встраивания в приборы различного назначения. От класса однокристальных микропроцессоров их отличает наличие встроенной памяти, развитые средства взаимодействия с внешними устройствами.

Микроконтроллер выполнен на основе высокоуровневой n-МОП технологии. Через четыре программируемых параллельных порта ввода/вывода и один последовательный порт микроконтроллер взаимодействует с внешними устройствами. Основу структурной схемы (рис. 1) образует внутренняя двунаправленная 8-битная шина, которая связывает между собой основные узлы и устройства микроконтроллера: резидентную память программ (RPM), резидентную память данных (RDM), арифметико-логическое устройство (ALU), блок регистров специальных функций, устройство управления (CU) и порты ввода/вывода (P0-P3).

3.1. Арифметико-логическое устройство

8-битное арифметико-логическое устройство (ALU) может выполнять арифметические операции сложения, вычитания, умножения и деления; логические операции И, ИЛИ, исключающее ИЛИ, а также операции циклического сдвига, сброса, инвертирования и т.п. К входам подключены программно-недоступные регистры T1 и T2, предназначенные для временного хранения операндов, схема десятичной коррекции (DCU) и схема формирования признаков результата операции (PSW).

Простейшая операция сложения используется в ALU для инкрементирования содержимого регистров, продвижения регистра-указателя данных (RAR) и автоматического вы-

Таким образом, ALU может оперировать четырьмя типами информационных объектов: булевыми (1 бит), цифровыми (4 бита), байтными (8 бит) и адресными (16 бит). В ALU выполняется 51 различная операция пересылки или преобразования этих данных. Так как используется 11 режимов адресации (7 для данных и 4 для адресов), то путем комбинирования операции и режима адресации базовое число команд 111 расширяется до 255 из 256 возможных при однобайтном коде операции.

3.2. Резидентная память программ и данных

Резидентные (размещённые на кристалле) память программ (RPM) и память данных (RDM) физически и логически разделены, имеют различные механизмы адресации, работают под управлением различных сигналов и выполняют разные функции.

Память программ RPM имеет емкость 4 Кбайта и предназначена для хранения команд, констант, управляющих слов инициализации, таблиц перекодировки входных и выходных переменных и т.п. Память имеет 16-битную шину адреса, через которую обеспечивается доступ из программного счётчика PC или из регистра указателя данных (DPTR). DPTR выполняет функции базового регистра при косвенных переходах по программе или используется в операциях с таблицами.

Память данных RDM предназначена для хранения переменных в процесс выполнения прикладной программы, адресуется одним байтом и имеет емкость 128 байт. Кроме того, к её адресному пространству примыкают адреса регистров специальных функций, которые перечислены в табл. 1.

Память программ, так же как и память данных, может быть расширена до 64 Кбайт путем подключения внешних микросхем.

Блок регистров специальных функций

Символ	Наименование	Адрес
* A	Аккумулятор	0E0H
* B	Регистр-расширитель аккумулятора	0F0H
* PSW	Слово состояния программы	0D0H
SP	Регистр-указатель стека	81H
DPTR	Регистр-указатель данных (DPH)	83H
	(DPL)	82H
* P0	Порт 0	80H
* P1	Порт 1	90H
* P2	Порт 2	0A0H
* P3	Порт 3	0B0H
* IP	Регистр приоритетов прерываний	0B8H
* IE	Регистр маски прерываний	0A8H
TMOD	Регистр режима таймера/счётчика	89H
* TCON	Регистр управления/статуса таймера	88H
TH0	Таймер 0 (старший байт)	8CH
TL0	Таймер 0 (младший байт)	8AH
TH1	Таймер 1 (старший байт)	8DH
TL1	Таймер 1 (младший байт)	8BH
* SCON	Регистр управления приёмопередатчиком	98H
SBUF	Буфер приёмопередатчика	99H
PCON	Регистр управления мощностью	87H

3.3. Аккумулятор, регистры общего назначения и флаги

Аккумулятор (A) является источником операнда и местом фиксации результата при выполнении арифметических, логических операций и ряда операций передачи данных. Кроме того, только с использованием аккумулятора могут быть выполнены операции сдвигов, проверка на нуль, формирование флага паритета и т.п. В распоряжении пользователя имеются 8 регистров общего назначения R0–R7 одного из четырёх возможных банков.

При выполнении многих команд в ALU формируется ряд признаков операции (флагов), которые фиксируются в регистре PSW. В табл. 2 приводится перечень флагов PSW, даются их символические имена и описываются условия их формирования.

Таблица 17.2 Формат слова состояния программы PSW

Символ	Разряд	Имя и назначение
C	PSV.7	Флаг переноса. Устанавливается и сбрасывается аппаратно или программно при выполнении арифметических и логических операций
AC	PSV.6	Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратно при выполнении команд сложения и вычитания и сигнализирует о переносе или займе в бите 3
F0	PSV.5	Флаг 0. Может быть установлен, сброшен или проверен программой как флаг, специфицируемый пользователем
RS1	PSV.4	Выбор банка регистров. Устанавливается и сбрасывается
RS0	PSV.3	программно для выбора рабочего банка регистров (табл. 3)
OV	PSV.2	Флаг переполнения. Устанавливается и сбрасывается аппаратно при выполнении арифметических операций
-	PSV.1	Не используется
P	PSV.0	Флаг паритета. Устанавливается и сбрасывается аппаратно в каждом цикле и фиксирует нечётное/чётное число единичных битов в аккумуляторе, т.е. выполняет контроль по четности

Наиболее “активным” флагом PSW является флаг переноса, который принимает участие и модифицируется в процессе выполнения множества операций, включая сложение, вычитание и сдвиги. Кроме того, флаг переноса (C) выполняет функции “булева аккумулятора” в командах, манипулирующих с битами. Флаг переполнения (OV) фиксирует арифметическое переполнение при операциях над целыми числами со знаком и делает возможным использование арифметики в дополнительных кодах. ALU не управляет флагами селекции банка регистров (RS0, RS1), их значение полностью определяется прикладной программой и используется для выбора одного из четырёх регистровых банков.

В микропроцессорах, архитектура которых опирается на аккумулятор, большинство команд работают с ним, используя неявную адресацию. В Intel 8051 дело обстоит иначе. Хотя процессор имеет в своей основе аккумулятор, он может выполнять множество команд и без его участия. Например, данные могут быть переданы из любой ячейки RDM в любой регистр, любой регистр может быть загружен непосредственным операндом и т.д. Многие логические операции могут быть выполнены без участия аккумулятора. Кроме того, переменные могут быть инкрементированы, декрементированы и проверены без использования аккумулятора. Флаги и управляющие биты могут быть проверены и изменены аналогично.

3.4. Регистры-указатели

8-битный указатель стека (SP) может адресовать любую область RDM. Его содержимое инкрементируется прежде, чем данные будут запомнены в стеке в ходе выполнения команд PUSH и CALL. Содержимое SP декрементируется после выполнения команд POP и RET. Подобный способ адресации элементов стека называют прединкрементным/постдекрементным. В процессе инициализации микроконтроллера после сигнала RST в SP автоматически загружается код 07H. Это значит, что если прикладная программа не перепределяет стек, то первый элемент данных в стеке будет располагаться в ячейке RDM с адресом 08H.

Двухбайтный регистр-указатель данных DPTR обычно используется для фиксации 16-битного адреса в операциях с обращением к внешней памяти. Командами микроконтроллера регистр-указатель данных может быть использован или как 16-битный регистр, или как два независимых 8-битных регистра (DPH и DPL).

3.5. Регистры специальных функций

Регистры с символическими именами IP, IE, TMOD, TCON, SCON и PCON используются для фиксации и программного изменения управляющих бит и бит состояния схемы прерывания, таймера/счётчика, приёмопередатчика последовательного порта и для управления энергопотреблением. Их организация будет описана ниже при рассмотрении особенностей работы микроконтроллера в различных режимах.

3.6. Устройство управления и синхронизации

Кварцевый резонатор, подключаемый к внешним выводам микроконтроллера, управляет работой внутреннего генератора, который в свою очередь формирует сигналы синхронизации. Устройство управления (CU) на основе сигналов синхронизации формирует машинный цикл фиксированной длительности, равной 12 периодам резонатора. Большинство команд микроконтроллера выполняется за один машинный цикл. Некоторые команды, оперирующие с 2-байтными словами или связанные с обращением к внешней памяти, выполняются за два машинных цикла. Только команды деления и умножения требуют четырех машинных циклов. На основе этих особенностей работы устройства управления производится расчёт времени исполнения прикладных программ.

На схеме микроконтроллера к устройству управления примыкает регистр команд (IR). В его функцию входит хранение кода выполняемой команды.

Входные и выходные сигналы устройства управления и синхронизации:

- * PSEN – разрешение программной памяти,
- * ALE – выходной сигнал разрешения фиксации адреса,
- * PROG – сигнал программирования,
- * EA – блокировка работы с внутренней памятью,
- * VPP – напряжение программирования,
- * RST – сигнал общего сброса,
- * VPD – вывод резервного питания памяти от внешнего источника,
- * XTAL – входы подключения кварцевого резонатора.

3.7. Параллельные порты ввода/вывода информации

Все четыре порта (P0-P3) предназначены для ввода или вывода информации побайтно. Каждый порт содержит управляемые регистр-защёлку, входной буфер и выходной драйвер.

Выходные драйверы портов 0 и 2, а также входной буфер порта 0 используются при обращении к внешней памяти. При этом через порт 0 в режиме временного мультиплексирования сначала выводится младший байт адреса, а затем выдается или принимается байт данных. Через порт 2 выводится старший байт адреса в тех случаях, когда разрядность адреса равна 16 бит.

Все выводы порта 3 могут быть использованы для реализации альтернативных функций, перечисленных в табл. 4. Эти функции могут быть задействованы путем записи 1 в соответствующие биты регистра-защёлки (P3.0-P3.7) порта 3.

Альтернативные функции порта P3 Таблица 4

Символ	Разряд	Имя и назначение
RD	R3.7	Чтение. Активный сигнал низкого уровня формируется аппаратно при обращении к внешней памяти данных
WR	R3.6	Запись. Активный сигнал низкого уровня формируется аппаратно при обращении к внешней памяти данных
T1	R3.5	Вход таймера/счётчика 1 или тест-вход
T0	R3.4	Вход таймера/счётчика 0 или тест-вход
INT1	R3.3	Вход запроса прерывания 1. Воспринимается сигнал низкого уровня или срез
INT0	R3.2	Вход запроса прерывания 0. Воспринимается сигнал низкого уровня или срез
TXD	R3.1	Выход передатчика последовательного порта в режиме UART.

		Выход синхронизации в режиме регистра сдвига
RXD	R3.0	Вход приёмника последовательного порта в режиме UART. Ввод/вывод данных в режиме регистра сдвига

Порт 0 является двунаправленным, а порты 1-3- квазидвунаправленными. Каждая линия портов может быть использована независимо для ввода или вывода. По сигналу RST в регистры-защёлки всех портов автоматически записываются единицы, настраивающие их тем самым на режим ввода..

Все порты могут быть использованы для организации ввода/вывода информации по двунаправленным линиям передачи. Однако порты 0 и 2 не могут быть использованы для этой цели в случае, если система имеет внешнюю память, связь с которой организуется через общую разделяемую шину адреса/данных, работающую в режиме временного мультиплексирования.

Обращение к портам ввода/вывода возможно с использованием команд, оперирующих с байтом, отдельным битом, произвольной комбинацией битов. При этом в тех случаях, когда порт является одновременно операндом и местом назначения результата, устройство управления автоматически реализует специальный режим, который называется “чтение-модификация-запись”.

Этот режим обращения предполагает ввод сигналов не с внешних выводов порта, а из его регистра защёлки, что позволяет исключить неправильное считывание ранее выведенной информации. Этот механизм обращения к портам реализован в командах:

ANL – логическое И, например, ANL P1,A;
 ORL – логическое ИЛИ, например, ORL P2,
 XRL – исключающее ИЛИ, например, XRL
 JBC – переход, если в адресуемом бите единица, например, JBC P1.1, LABEL;
 CPL – инверсия бита, например, CPL P3.3;
 INC – инкремент порта, например, INC P2;
 DEC – декремент порта, например, DEC P2;
 DJNZ – декремент порта и переход, если например, DJNZ r, LABEL;
 MOV PX.Y,C – передача бита переноса в бит
 SET PX.Y – установка бита Y порта X;
 CLR PX.Y – сброс бита Y порта X.

4 Контрольные вопросы.

- Перечислите характерные черты архитектуры однокристальных микроконтроллеров.
- Укажите программно-доступные узлы K1816BE51 и назначение регистров специальных функций.
- Дайте определение понятию “булев” процессор.
- Назовите и охарактеризуйте четыре типа информационных объектов, с которыми может оперировать арифметико-логическое устройство микроконтроллера.
- Какова ёмкость адресуемой памяти программ и данных в K1816BE51?
- Какой регистр выполняет функции базового регистра при косвенных переходах в программе?
- Какие операции могут быть выполнены только с использованием аккумулятора?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №18.

**Изучение микроконтроллера Intel 8051 и принципов его работы: Интерфейс связи;
Объединение периферийных устройств
Объём учебного времени – 2ч**

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.
- Изучение общих принципов структурной организации ЭВМ
- Изучение архитектуры микроконтроллеров

2.Перечень необходимых средств обучения:

2.9 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.10 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа ProView
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Работа UART в мультиконтроллерных системах

В системах децентрализованного управления, которые используются для управления и регулирования в топологически распределенных объектах, возникает задача обмена информацией между множеством микроконтроллеров, объединенных в локальную вычислительно-управляющую сеть. Как правило, локальные сети на основе Intel 8051 имеют магистральную архитектуру с разделяемым моноканалом (коаксиальный кабель, витая пара, оптическое волокно), по которому осуществляется обмен информацией между контроллерами.

Бит SM2 в SCON позволяет простыми средствами реализовать межконтроллерный обмен. Механизм обмена построен на том, что в режимах 2 и 3 программируемый девятый бит данных при приёме фиксируется в бите RB8. UART может быть запрограммирован таким образом, что при получении стоп-бита прерывание от приёмника будет возможно только при условии RB8=1. Ведущий контроллер всем ведомым передаёт широковещательное сообщение с байтоидентификатором абонента, которое отличается от байтов данных только тем, что в его девятом бите содержится 1. Ведомые по этому признаку вызывают подпрограммы сравнения байта-идентификатора с кодом собственного сетевого адреса.

Адресуемый контроллер сбрасывает свой SM2 и готовится к приёму блока данных. Остальные ведомые микроконтроллеры оставляют неизменными свои SM2=1 и передают управление основной программе. При SM2=1 информационные байты в сети прерывания не вызывают.

В режиме 1 автономного микроконтроллера SM2 используется для контроля истинности стоп-бита. В режиме 0 SM2 не используется и должен быть сброшен.

3.1. Скорость приёма/передачи

Скорость зависит от режима работы UART. В режиме 0 частота зависит только от резонатора: $f_0 = f_{\text{рез}}/12$. За один машинный цикл передается один бит. В режимах 1-3 скорость зависит от значения управляющего бита SMOD в регистре специальных функций PCON (табл. 18.1).

В режиме 2 частота передачи $f_2 = (2^{\text{SMOD}}/64)f_{\text{рез}}$.

Таблица 18.1 Регистр управления мощностью PCO

Символ	Разряд	Наименование и функция
SMOD	PCON.7	Удвоенная скорость передачи. Если бит установлен в 1, то скорость передачи вдвое больше, чем при SMOD=0
-	PCON.6-4	Не используются
GF1	PCON.3	Флаги, специфицируемые пользователем (флаги общего назначения)
GF0	PCON.2	
PD	PCON.1	Бит пониженной мощности. При установке в 1 микроконтроллер переходит в режим пониженного энергопотребления
IDL	PCON.0	Бит холостого хода. Если бит установлен в 1, то микроконтроллер переходит в режим холостого хода

В табл. 18.2 приводится описание способов настройки T/C1 для получения типовых частот передачи данных через UART.

Таблица 18.2 Настройка таймера 1 для управления частотой работы UART

Частота приёма/ передачи (BAUD RATE)		Частота резонатора, МГц	SMOD			
				C/T	Режим (MODE)	Перезагружаемое число
Режим 0, макс.:	1 МГц	12	X	X	X	X
Режим 2, макс.:	375 кГц	12	1	X	X	X
Режимы 1,3:	62.5 кГц	12	1	0	2	0FFH
	19.2 кГц	11,059	1	0	2	0FDH
	9.6 кГц	11,059	0	0	2	0FDH
	4.8 кГц	11,059	0	0	2	0FAH
	2.4 кГц	11,059	0	0	2	0F4H
	1.2 кГц	11,059	0	0	2	0E8H
	137.5 кГц	11,059	0	0	2	1DH
	110 Гц	11,0596	0	0	2	72H
	110 Гц	12	0	0	1	0FEVBH

3.2. Система прерываний

Внешние прерывания INT0 и INT1 (рис. 18.1) могут быть вызваны уровнем или переходом сигнала из 1 в 0 на входах микроконтроллера в зависимости от значений управляющих битов IT0 и IT1 в регистре TCON. От внешних прерываний устанавливаются флаги IE0 и IE1 в регистре TCON, которые инициируют вызов соответствующей подпрограммы обслуживания прерывания. Сброс этих флагов выполняется аппаратно только в том случае, если прерывание было вызвано по переходу (срезу) сигнала. Если же прерывание вызвано уровнем входного сигнала, то сбросом флага IE управляет соответствующая подпрограмма обслуживания прерывания путем воздействия на источник прерывания с целью снятия им запроса.

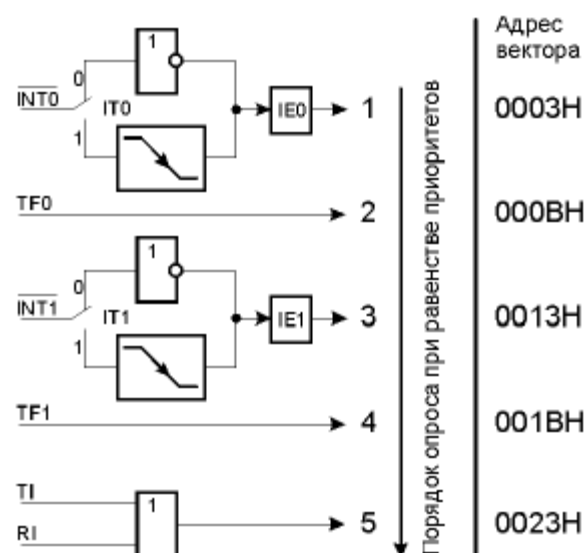


Рис. 18.1. Схема прерываний

Флаги запросов прерывания от таймеров TF0 и TF1 сбрасываются автоматически при передаче управления подпрограмме обслуживания. Флаги запросов прерывания RI и TI устанавливаются UART аппаратно, но сбрасываться должны программой. Прерывания могут быть вызваны или отменены программой, так как все перечисленные флаги программно доступны. В блоке регистров специальных функций есть два регистра, предназначенных для управления режимом прерываний и уровнями приоритета. Форматы этих регистров, имеющих символические имена IE и IP описаны в табл. 18.3 и 18.4соответственно.

Таблица 18.3 Регистр масок прерывания IE

Символ	Разряд	Имя и назначение
EA	IE.7	Снятие блокировки прерываний. Сбрасывается программно для запрета всех прерываний независимо от состояний IE4-IE0
-	IE.6,5	Не используются
ES	IE.4	Бит разрешения прерывания от UART. Установка/сброс программой для разрешения/запрета прерываний от флагов TI, RI
ET1	IE.3	Бит разрешения прерывания от таймера 1. Установка/сброс программой для разрешения/запрета прерываний от таймера 1
EX1	IE.2	Бит разрешения внешнего прерывания 1. Установка/сброс программой для разрешения/запрета прерываний
ET0	IE.1	Разрешение прерывания от таймера 0. Работает аналогично IE.3
EX0	IE.0	Разрешения внешнего прерывания 0. Работает аналогично IE.2

Таблица 18.4Регистр приоритетов прерывания IP

Символ	Разряд	Имя и назначение
-	IP.7-5	Не используются
PS	IP.4	Бит приоритета UART. Установка/сброс программой для назначения прерыванию от UART высшего/низшего приоритета
PT1	IP.3	Бит приоритета таймера 1. Установка/сброс программой для назначения прерыванию от таймера 1 высшего/низшего приоритета
PX1	IP.2	Бит приоритета внешнего прерывания 1. Установка/сброс программой для назначения прерыванию INT1 высшего/низшегоприоритета
PT0	IP.1	Бит приоритета таймера 0. Работает аналогично IP.3
PX0	IP.0	Приоритет внешнего прерывания 0. Работает аналогично IP.2

Возможность программной установки/сброса любого управляющего бита в этих двух регистрах делает систему прерываний исключительно гибкой.

Флаги прерываний опрашиваются в каждом машинном цикле. Ранжирование прерываний по приоритету выполняется в течение следующего машинного цикла.

Система прерываний сформирует аппаратно вызов LCALL соответствующей подпрограммы обслуживания, если она не заблокирована одним из условий:

- 1) в данный момент обслуживается запрос прерывания равного или более высокого уровня приоритета;
- 2) текущий машинный цикл – не последний в цикле выполняемой команды;
- 3) выполняется команда RETI или любая команда, связанная с обращением к регистрам IE или IP.

3.4. ВВЕДЕНИЕ В PROVIEW

ProView фирмы Franklin Software Inc. – интегрированная среда разработки программного обеспечения для микроконтроллеров семейства Intel 8051 и его клонов. Эта среда позволяет создавать, редактировать, компилировать, транслировать, компоновать, осуществлять загрузки и отладки программ. ProView содержит:

- стандартный интерфейс Windows,
- организатор проекта,
- полнофункциональный редактор исходных текстов,
- транслятор с языка C,
- ассемблер,
- отладчик,
- встроенную справочную систему.

На первом этапе разработки программы осуществляется запись её исходного текста на языке C или ассемблере.

Затем производится компиляция или трансляция его в коды команд микроконтроллера, используя транслятор или ассемблер.

Трансляторы и ассемблеры преобразуют исходный текст программы в объектные файлы, содержащие объектный код.

После компоновки объектных модулей выполняются этапы отладки программы, устранения ошибок, оптимизации и тестирования программы.

ProView объединяет все этапы разработки прикладной программы в единый рекурсивный процесс, позволяющий быстрый возврат к любому предыдущему этапу. ProView содержит следующие компоненты.

3.4.1. Оптимизирующий кросс-компилятор C51

C51 генерирует код для всего семейства микроконтроллеров Intel 8051. Транслятор сочетает гибкость программирования на языке C с эффективностью кода и быстродействием ассемблера.

Программирование на языке высокого уровня C имеет следующие преимущества над программированием на ассемблере:

- распределение регистров и способы адресации управляются полностью транслятором;
- лучшая читаемость программы, используются ключевые слова и функции, которые более понятны;
- время разработки программ и их отладки значительно короче в сравнении с программированием на ассемблере;
- библиотечные файлы содержат много стандартных подпрограмм, которые можно использовать в прикладных программах;
- модульные методы программирования позволяют отлаженные программы использовать в новых разработках;
- не требуется глубокого знания системы команд микроконтроллера.

3.4.2. Макроассемблер A51

Ассемблер A51 совместим с ASM-51 Intel для всего семейства микроконтроллеров Intel 8051. Ассемблер A51 транслирует символическую мнемонику в перемещаемый объектный код, имеющий высокое быстродействие и малый размер. Макросредства ускоряют разработку и экономят время, поскольку общие последовательности могут быть разработаны только один раз. Ассемблер поддерживает символический доступ ко всем элементам микроконтроллера и перестраивает конфигурацию для каждой разновидности Intel 8051.

A51 транслирует исходный файл ассемблера в перемещаемый объектный модуль. При отладке или при включенной опции “Include debugging information” этот объектный файл будет содержать полную символическую информацию для отладчика/имитатора или внутрисхемного эмулятора.

3.4.3. Компоновщик L51

Компоновщик объединяет один или несколько объектных модулей в одну исполняемую программу. Компоновщик размещает внешние и общие ссылки, назначает абсолютные адреса перемещаемым сегментам программ. Он может обрабатывать объектные модули, созданные транслятором C51, ассемблером A51, транслятором PL/M-51 Intel и ассемблером ASM-51 Intel.

Компоновщик автоматически выбирает соответствующие библиотеки поддержки и связывает только требуемые модули из библиотек. Установки по умолчанию для L51 выбраны так, чтобы они подходили для большинства прикладных программ, но можно определить и заказные установки.

3.4.4. Отладчик/симулятор WinSim51

Отладчик исходных текстов используется с транслятором C51, ассемблером A51, транслятором PL/M-51 Intel и ассемблером ASM-51 Intel. Отладчик/симулятор позволяет моделировать большинство особенностей Intel 8051 без наличия аппаратных средств. Можно использовать его для проверки и отладки прикладной программы прежде, чем будут изготовлены аппаратные средства. При этом моделируется широкое разнообразие периферийных устройств, включая последовательный порт, внешний ввод - вывод и таймеры.

3.5. БЫСТРЫЙ СТАРТ

“Быстрый старт” – это обычный приём разработчиков современных программных средств. Цель состоит в том, чтобы, не углубляясь пока в подробности, дать новичку или достаточно опытному пользователю первое представление о программном средстве, дать возможность быстро получить конкретный результат. Полное представление, знания и умения появятся позже в процессе работы и изучения справочных материалов.

В качестве примера возьмём простейшую программу, с которой начинают изучение языков программирования многие поколения студентов. “Hello World” - программа из папки \Fsi\Examples\Hello\, которая выводит в последовательный порт (UART) микроконтроллера строку символов “Hello World” (“Привет Мир”). Весь исходный текст программы содержится в файле hello.c:

```

/*****
/* YOUR FIRST 8051 PROGRAM */
*****/

#include <reg51.h> /* special function register declarations */
                  /* for the intended 8051 derivative */
#include <stdio.h> /* prototype declarations for I/O functions*/

/*****
/* main program */
*****/
void main (void) { /* execution starts here after stack init */
    SCON = 0x50;    /* SCON: mode 1, 8-bit UART, enable rcvr */
    TMOD |= 0x20;   /* TMOD: timer 1, mode 2, 8-bit reload */
    TH1 = 0xf3;     /* TH1: reload value for 2400 baud */
    TR1 = 1;        /* TR1: timer 1 run */
    TI = 1;         /* TI: set TI to send first char of UART*/

    printf ("Hello World\n"); /* the 'printf' function call */

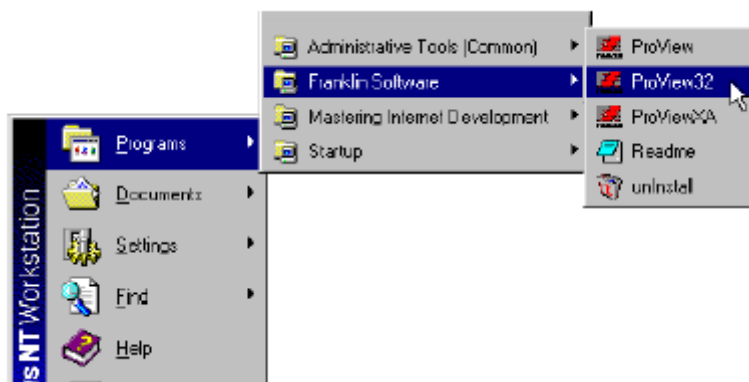
    while (1) {      /* An embedded program does not stop and */
        ; /* ... */ /* never returns. We've used an endless */
    }               /* loop. You may wish to put in your own */
}                  /* code were we've printed the dots (...) */

```

Прежде чем начать разработку проекта, скопируйте папку \Fsi\Examples\Hello\ в свою личную папку. В этой папке находится всего лишь один файл hello.c.

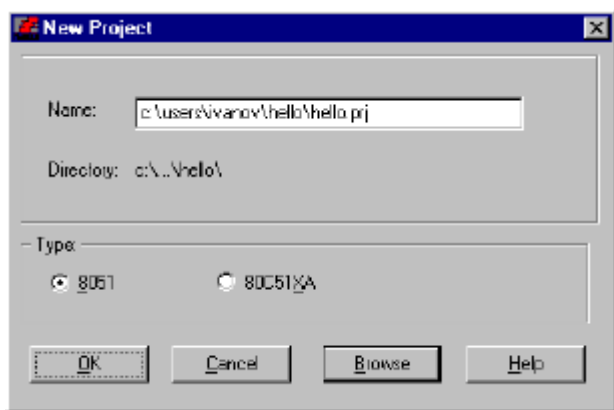
3.5.1. Запуск ProView и создание файла проекта

ProView запускается из стартового меню Windows подобно остальным приложениям. Если необходимо запустить программу из командной строки, её синтаксис имеет вид: PV32 [projectfile], где projectfile - имя файла проекта с расширением [.PRJ].



Любая новая работа в ProView, как и во всех современных компиляторах, начинается с создания нового файла проекта. Файл проекта содержит имена всех исходных файлов, связанных с проектом, а также установки компиляции, трансляции и связывания файлов, чтобы генерировать выполняемую программу.

Для того чтобы создать новый файл проекта, выберите New из меню Project. Откроется диалоговое окно New Project. Используйте кнопку Browse, чтобы войти в свою папку. Найдите папку \Hello и нажмите кнопку [OK]. Затем выберите “8051” как тип проекта.

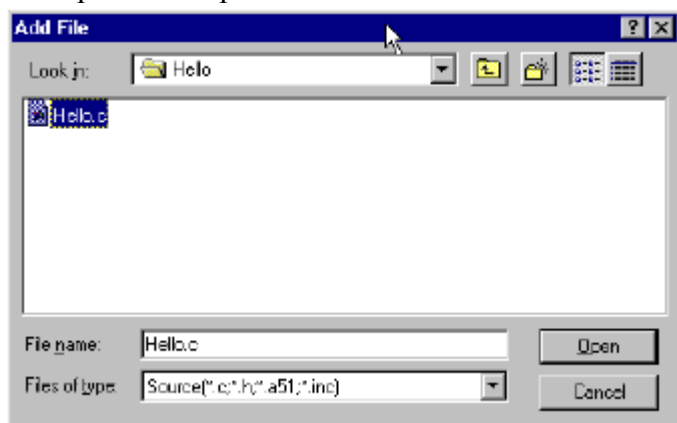


Когда менеджер проекта открывает файл проекта, окно проекта показывает включенные исходные файлы. В данном случае пока нет никаких исходных файлов. Имеется только один исходный файл, который необходимо подключить- hello.c.

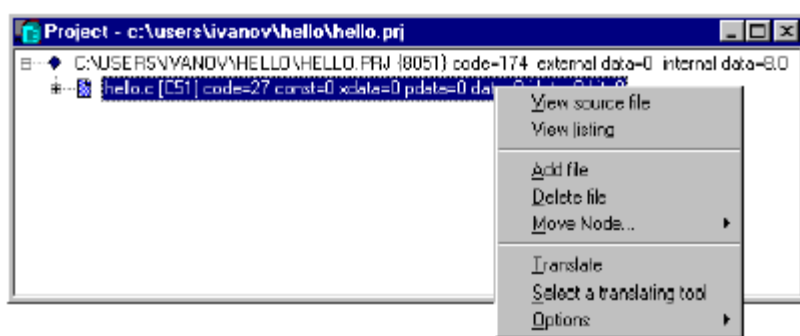
3.5.2. Добавка файла с исходным текстом и его редактирование

Теперь можно добавить hello.c к проекту. Выберите Add file из меню Project. Откроется диалоговое окно Add File . Выберите hello.c из списка.

Наш проект имеет только один исходный файл. В дальнейшем Ваши проекты, возможно, будут состоять из множества исходных файлов. Диалог Add File позволит Вам выбрать и добавить несколько файлов сразу. Для этого используют комбинацию клавиши [CTRL] и указателя мыши. Когда Вы нажмёте [Open], исходные файлы будут добавлены к проекту в выбранном порядке.

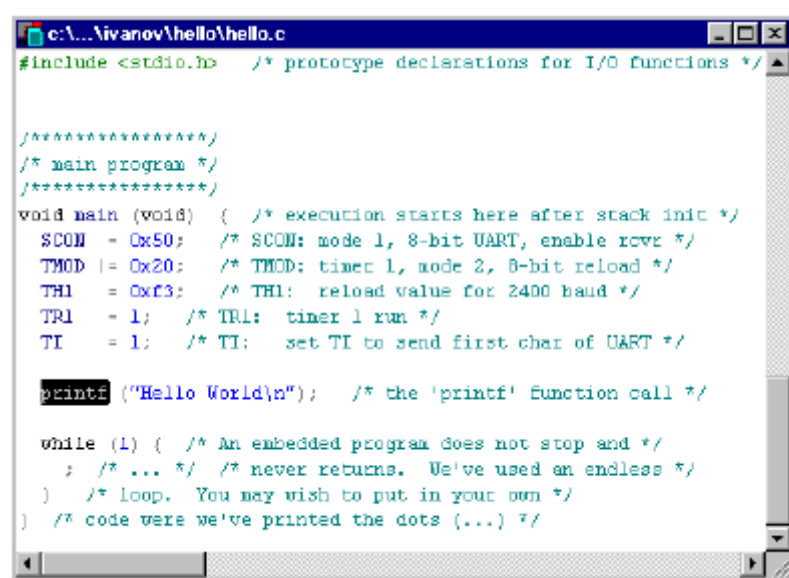


Теперь можно редактировать текст из файла hello.c. Выберите hello.c из окна Project . Нажмите его правой кнопкой мыши и выберите View source file, или просто дважды щёлкните мышью для того, чтобы просматривать файл в окне редактирования.



ProView загружает и показывает содержание hello.c в окне, где можно редактировать файл. Окно редактирования - полнофункциональный редактор исходного текста, предлагающий такие возможности, как высвечивание синтаксических элементов и контекстный по-

иск. Если выбрать “printf” и нажать клавишу [F1], ProView откроет систему справки и перейдёт к разделу справки о “printf”.



```
c:\...Nivanov\hello\hello.c
#include <stdio.h> /* prototype declarations for I/O functions */

/*****
/* main program */
*****/

void main (void) { /* execution starts here after stack init */
    SCON = 0x50; /* SCON: mode 1, 8-bit UART, enable rcv */
    TMOD = 0x20; /* TMOD: timer 1, mode 2, 8-bit reload */
    TH1 = 0xf3; /* TH1: reload value for 2400 baud */
    TRL = 1; /* TRL: timer 1 run */
    TI = 1; /* TI: set TI to send first char of UART */

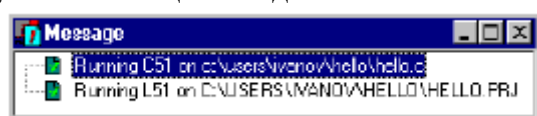
    printf ("Hello World\n"); /* the 'printf' function call */

    while (1) { /* An embedded program does not stop and */
        ; /* ... */ /* never returns. We've used an endless */
    } /* loop. You may wish to put in your own */
} /* code were we've printed the dots (...) */
```

3.5.3. Компиляция и компоновка

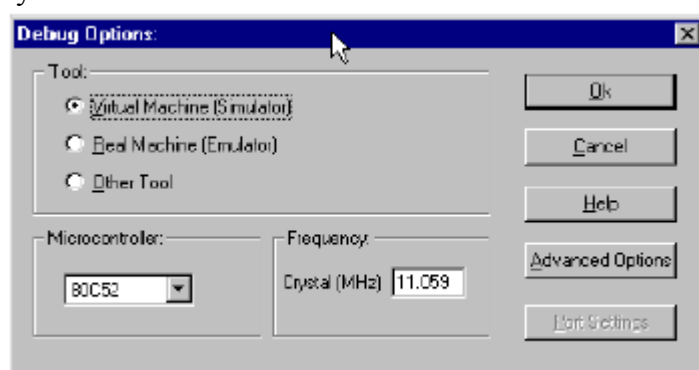
Этот процесс компилирует, связывает hello.c с библиотеками и создает абсолютный объектный модуль, который мы сможем проверить в отладчике WinSim. Выберите Make из меню Project. ProView отображает окно,

показывая текущее состояние процесса. Когда процесс компиляции закончится, в окне Message отображается сообщение завершения. Если были обнаружены какие-нибудь ошибки, о них сообщается здесь же.

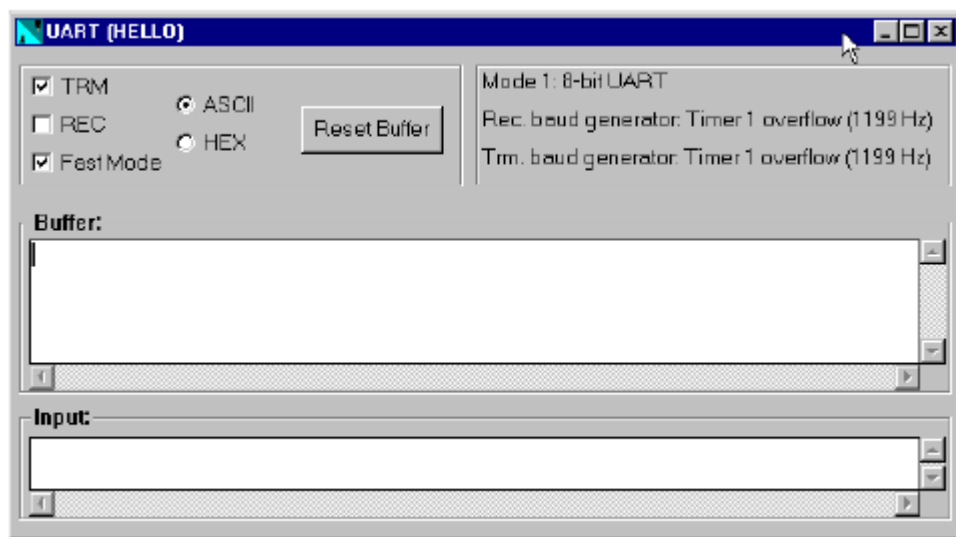


3.5.4. Тестирование и отладка

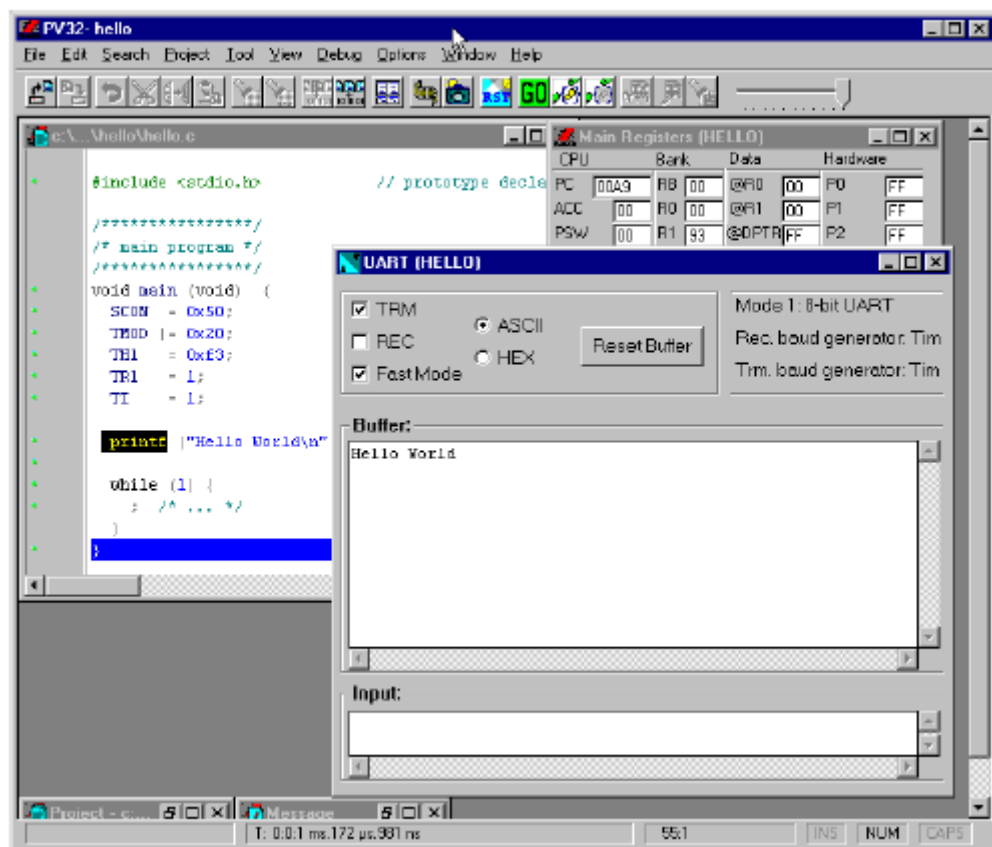
Выполним отладку программы. Если проект новый, откроется диалоговое окно Debug Options где Вы можете изменять установки отладчика. В дальнейшем можно установить опции отладчика, выбрав Debug из меню Options. Наш проект использует значения по умолчанию.



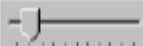
Выберите Hardware (аппаратные средства) из меню View. Выберите UART, откроется окно последовательного порта В дальнейшем при работе программы здесь можно будет увидеть всё, что выводит микроконтроллер в последовательный порт.



Выберите Run из меню Debug или нажмите кнопку **Run**. Так, как выглядит экран отладчика WinSim при выполнении программы и в окне UART выведен текст "Hello World".



При выводе символов в порт начинается выполнение бесконечного цикла. Вы можете остановить выполнение программы, выбрав Stop из меню Debug.

С помощью регулятора  при нажатой кнопке  на панели инструментов можно менять скорость работы отладчика. Строка состояния показывает текущее реальное время.

3.5.5. Пошаговый режим и выход из отладчика

Вы можете использовать отладчик, чтобы перемещаться по программе. Выберите Reset из меню Debug (эта команда сбросит моделируемый процессор) и выберите Step Into и Step Over из меню Debug.

Команды Step позволяют “шагать” по каждой строке исходного текста. Текущая команда высвечивается на каждом шаге. Step Into позволяет войти в вызываемую функцию, Step Over – перешагнуть через неё, не входя во внутрь.

Проделайте эти операции.

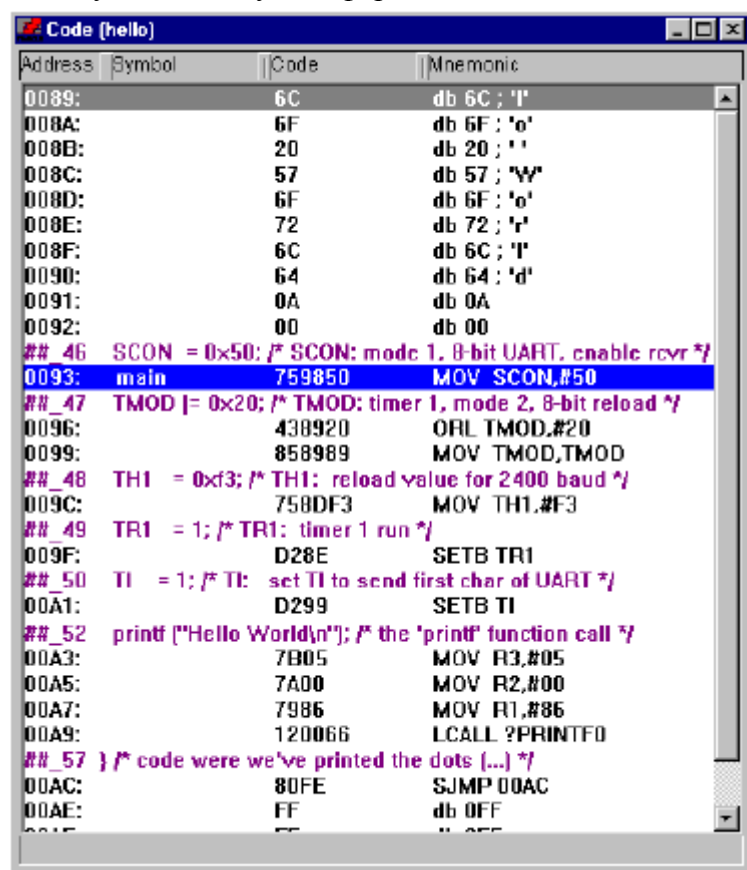
Для завершения работы с отладчиком в любой момент времени Вы можете выбрать Terminate из меню Debug и возвратиться в режим редактирования.

3.5.6. Следующий шаг

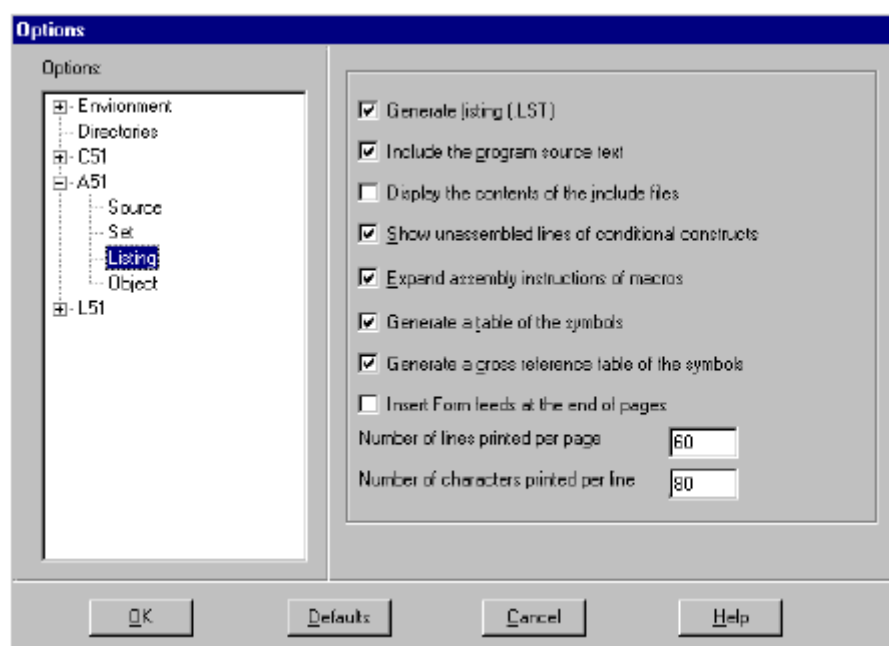
Обратите внимание, что в режиме отладки на экране видны ещё два окна. Первое – окно кода где в пошаговом режиме параллельно с исходным текстом на языке C идёт трассировка текста на ассемблере.

Прокрутите окно кода и изучите ассемблерный аналог исходного текста. С символов “##” начинаются строки, с помощью которых легко сопоставить ассемблерный текст и текст на языке C. Обратите внимание на то, сколько кода пришлось бы написать, если проектировать программу на ассемблере.

Ассемблерный аналог текста сохраняется в файле hello.lst, если в опциях проекта (Project из меню Options) отмечено Generate Listing. Здесь же можно указать, какую информацию включать в листинг.



Address	Symbol	Code	Mnemonic
0089:		6C	db 6C ; 'I'
008A:		6F	db 6F ; 'o'
008B:		20	db 20 ; ''
008C:		57	db 57 ; 'W'
008D:		6F	db 6F ; 'o'
008E:		72	db 72 ; 'r'
008F:		6C	db 6C ; 'I'
0090:		64	db 64 ; 'd'
0091:		0A	db 0A
0092:		00	db 00
##_46	SCON = 0x50; /* SCON: mode 1, 8-bit UART, enable rcvr */		
0093:	main	759850	MOV_SCON,#50
##_47	TMOD = 0x20; /* TMOD: timer 1, mode 2, 8-bit reload */		
0096:		438920	ORL_TM0D,#20
0099:		858989	MOV_TM0D,TM0D
##_48	TH1 = 0xf3; /* TH1: reload value for 2400 baud */		
009C:		758DF3	MOV_TH1,#F3
##_49	TR1 = 1; /* TR1: timer 1 run */		
009F:		D28E	SETB_TR1
##_50	TI = 1; /* TI: set TI to send first char of UART */		
00A1:		D299	SETB_TI
##_52	printf("Hello World\n"); /* the 'printf' function call */		
00A3:		7B05	MOV_R3,#05
00A5:		7A00	MOV_R2,#00
00A7:		7986	MOV_R1,#86
00A9:		120066	LCALL_?PRINTF0
##_57	/* code were we've printed the dots (...) */		
00AC:		80FE	SJMP_00AC
00AE:		FF	db 0FF
00AF:		FF	db 0FF



Изучите смысл других опций проекта в разделах Environment, C51, A51, L51. Откройте файл листинга с помощью View listing из меню View.

Изучите и постарайтесь понять содержание разделов файла листинга.

```

c:\...Nivanov\hello\hello.lst
ASSEMBLY LISTING OF GENERATED OBJECT CODE

; FUNCTION main (BEGIN)
0000 759850      MOV     SCON,#050H      ; SOURCE LINE # 46
0003 438920      ORL     TH0D,#020H      ; SOURCE LINE # 47
0006 858989      MOV     TH0D,TH0D
0009 758DF3      MOV     TH1,#0F3H      ; SOURCE LINE # 48
000C D28E        SETB   TR1            ; SOURCE LINE # 49
000E D299        SETB   TI            ; SOURCE LINE # 50
0010 7B05        MOV     R3,#005H      ; SOURCE LINE # 52
0012 7A00 R      MOV     R2,#000H
0014 7900 R      MOV     R1,#000H
0016 120000 R    LCALL  ?printf
0019           ?WHILE1:
0019 80FE        SJMP    ?WHILE1      ; SOURCE LINE # 55
; FUNCTION main (END)
  
```

Main Registers (HELLO)							
CPU	Bank	Data		Hardware			
PC	00AC	R8	00	@R0	00	P0	FF
ACC	00	R0	00	@R1	00	P1	FF
PSW	00	R1	93	@DPTR	FF	P2	FF
SP	08	R2	00	X@R0	FF	P3	FF
DPTR	0000	R3	05	X@R1	FF	TCON	00
B	00	R4	00	SPX	XX	THL0	0000
C	0	R5	00	XAREA	XX	THL1	F3F4
EA	0	R6	0A	Task	XX	THL2	0000
IE	00	R7	01	TaskP	XX	PCON	00

Второе окно, которое присутствует на экране во время отладки, – MainRegisters . В этом окне постоянно отображается текущее состояние всех программно-доступных регистров микроконтроллера. Более того, содержимое регистров можно менять во время отладки.

С помощью пункта Data dump из меню View можно посмотреть содержимое памяти различного типа в режиме отладки. Попробуйте это сделать.

4 Контрольные вопросы.

- Нарисуйте схему прерываний в K1816BE51. Перечислите и охарактеризуйте типы прерываний.
- Для чего нужен регистр масок прерывания? Как изменить приоритеты прерываний?
- Как переводится микроконтроллер в режим пониженного энергопотребления?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Схема прерываний в K1816BE51
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №19.

Изучение процессов обработки информации на всех уровнях компьютерной архитектуры RISC на примере 32-разрядного - процессора 80960SA (Среда Отладчика; Загрузка и отладка; Команды перемещения данных)

Объём учебного времени – 2ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.
- Изучение архитектуры процессоров

2.Перечень необходимых средств обучения:

2.11 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.12 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Существует два типа архитектуры процессоров — CISC и RISC. CISC (Complete Instruction Set Computing) — тип архитектуры процессора с полным набором команд. Традиционные центральные микропроцессоры основаны на CISC-архитектуре. При этом подходе выполнение любой сколь угодно сложной команды из системы команд процессора реализуется аппаратно внутри самого процессора. В систему команд CISC-процессора может входить, например, вычисление квадратного корня, что требует многих десятков тактов. Добавление каждой новой команды ведет к увеличению общего числа транзисторов в процессоре. CISC-архитектура позволяет создавать универсальные процессоры, но их производительность ограничена, в частности, сложностью микросхемы процессора. Микропроцессоры с CISC-архитектурой (например, Intel Pentium II, III, и IV или Athlon и Duron компании AMD) обычно устанавливаются в настольных и переносных компьютерах общего назначения. Напротив, в процессорах с RISC-архитектурой используется ограниченный набор быстрых команд. Каждая команда RISC-процессора должна выполняться за один машинный такт, так что вряд ли вы найдете в системе команд даже умножение. В таких микропроцессорах содержится меньшее количество транзисторов, что снижает их стоимость и энергопотребление. При этом, как правило, повышается их производительность. Нотам, где CISC-процессор выполняет одну команду, для RISC-процессора следует писать небольшую программу. Микропроцессоры с RISC-архитектурой устанавливаются в специализированных устройствах, например, в лазерных принтерах. Разработчики пытаются создать процессоры, которые сочетали бы в себе универсальность CISC-архитектуры с производительностью RISC-архитектуры. Вместе с тем, ряд RISC-процессоров (например, DEC Alpha или MIPS Orion 4600) используется в качестве центральных в высокопроизводительных рабочих станциях.

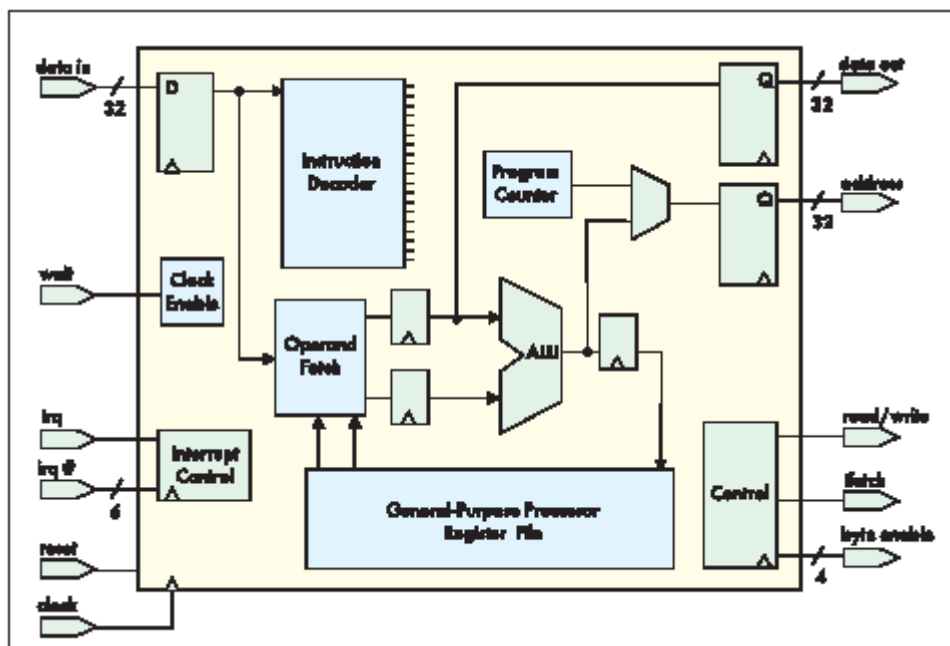


Рисунок 19.1 схема 32-разрядного процессора

3.1 Создание схемы 32-разрядного процессора в Electronic Work Bench 5.12

4 Контрольные вопросы.

- Какие типы двоичных счетчиков Вы знаете?
- Чем определяется быстродействие счетчика?
- Какие способы повышения быстродействия счетчиков Вы знаете?
- Что такое граф перехода?
- Что такое функция перехода?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Создание схемы 32-разрядного процессора
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

1. <http://novtex.ru/IT> - журнал "Информационные технологии"
2. <http://infojournal.ru> - журнал «Информатика и образование»
3. <http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №20.

Изучение процессов обработки информации на всех уровнях компьютерной архитектуры RISC на примере 32-разрядного - процессора 80960SA Порт 8255 и команды ввода-вывода (I/O)

Объем учебного времени – 2ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.
- Изучение архитектуры процессоров
- Ознакомление с возможностями моделирования работы арифметико-логического устройства (АЛУ).
- Исследование работы АЛУ на примере выполнения арифметических и логических операций

2.Перечень необходимых средств обучения:

2.13 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.14 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Арифметико-логическое устройство предназначено для выполнения арифметических и логических операций над многоразрядными операндами в зависимости от кодов, подаваемых на управляющие входы.

В вычислительных устройствах АЛУ является базовым узлом и работает в сочетании с ОЗУ, регистрами сдвига, регистрами общего назначения и др. Микросхемы АЛУ, принадлежащие к разным видам логик, функционально во многом совпадают. Аналогом микросхемы K155ИПЗ в программе EWB имеется микросхема 74181. Она представляет собой четырехразрядное АЛУ.

АЛУ работает в режиме выполнения логических операций при значении управляющего сигнала $M=1$ и в режиме выполнения арифметических операций при значении управляющего сигнала $M=0$.

Эта микросхема обеспечивает 32 режима работы АЛУ в зависимости от управляющих сигналов на входах M , S_0 - S_3 . Возможные режимы задаются путем подачи сигналов на входы управления S_0 , S_1 , S_2 , S_3 . Если сигнал на входе M равен 0, то выполняются 16 арифметических операций (16 комбинаций сигналов $S_0...S_3$) с учетом переноса по входу CN (если $CN=0$) или без учета переноса (если $CN=1$). При сигнале на входе M равном 1, выполняются 16 логических операций, задаваемых S_0 - S_3 .

Четырёхразрядные операнды А и В задаются на входах А0-А3 и В0-В3 соответственно. Результат арифметической или логической операции появляется на выходах F0-F3.

В микросхеме предусмотрены выходы переноса CN+4, ускоренного переноса Р, ускоренного группового переноса G, равенства операндов А=В. Питание микросхемы осуществляется подключением источника питания 5 В ко входу VCC и заземлением входа GND.

В приведенной таблице содержатся выполняемые логические и арифметические операции в зависимости от кодовой комбинации на управляющих входах S0, S1, S2, S3.

Таблица 1 Логические и арифметические операции

Выбор функции				Логические функции M=1	Арифметические функции, M=0	
S3	S2	S1	S0		С переносом CN=0	Без переноса CN=1
0	0	0	0	Not A	A + 1	A
0	0	0	1	not (A or B)	A + B + 1	A or B
0	0	1	0	(Not A) and B	A + (not B) + 1	A or (not B)
0	0	1	1	Логический 0	0	-1
0	1	0	0	Not (A and B)	A + Not (A and B) + 1	A + (A and (not B))
0	1	0	1	Not B	(A or B) + (A and (not B)) + 1	(A+B) + (A and (not B))
0	1	1	0	A xor B	A – B	A–B–1
0	1	1	1	A and (not B)	Not (A and B)	(A and (not B)) – 1
1	0	0	0	(Not A) or B	A + B + 1	A+(A and B)
1	0	0	1	Not (A xor B)	A + B + 1	A + B
1	0	1	0	B	(A or (not B)) + (A and B) + 1	(A or (not B)) + A and B
1	0	1	1	A and B	A and B	A and B – 1
1	1	0	0	Логическая 1	A + A + 1	A + A
1	1	0	1	A or (not B)	(A or B) + A + 1	(A or B) + A
1	1	1	0	A or B	(A or (not B)) + A + 1	(A or (not B)) + A
1	1	1	1	A	A	A – 1

Используемые элементы EWB.

На панели Digital ICs выбирается схема из множества 741хх. Под номером 74181 - Четырёх-разрядное АЛУ.

1. Задание к работе

1.1 Построение АЛУ

1.1.1 Подготовить микросхему АЛУ на рабочем столе.

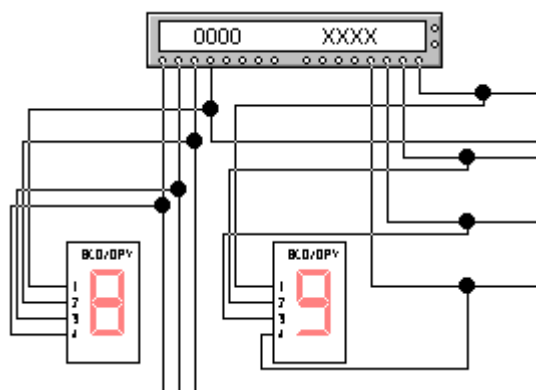


Рисунок 20.1 Фрагмент АЛУ

– Подготовить 4 переключателя для задания режимов управления.

- Сигналы S подаются на входы $S0.....S3$ АЛУ.
- Ещё один переключатель используется для задания режима M .
- И в том и другом случае логическая 1 подаётся от 5V источника постоянного тока, логический 0 от заземления.
- Шестой переключатель осуществляет роль входного переноса и подключается таким же образом к входу CN .
- Значения четырёхразрядных операндов A и B задаются с помощью генератора слова и в шестнадцатеричном коде отображаются на алфавитно-цифровых индикаторах и подсоединяются к входам $A0.....A3$ и $B0.....B3$.
- На выходах $F0....F3$ формируется результат операции АЛУ. Для отображения результата к выходам $F0....F3$ присоединяется алфавитно-цифровой индикатор.
- К выходу VCC присоединить источник 5V.
- К входу GND подсоединить заземление.
- При коде 1111 на выходах F и при равенстве операндов выход $A=B$ переводится в единичное состояние. Поскольку этот выход представляет собой каскад с открытым коллектором, то на него подаётся питание +5 вольт через резистор 1 кОм. Выход $A=B$ совместно с выходом переноса $CN+4$ и выходом P подтверждения переноса используются для формирования признаков $A>B$ и $A<B$ с помощью дополнительных логических элементов ИЛИ-НЕ и НЕ.
- Изменяя состояния сигналов на управляющих входах по приведённой таблице, можно промоделировать большинство функций АЛУ, используемых в микропроцессорах.

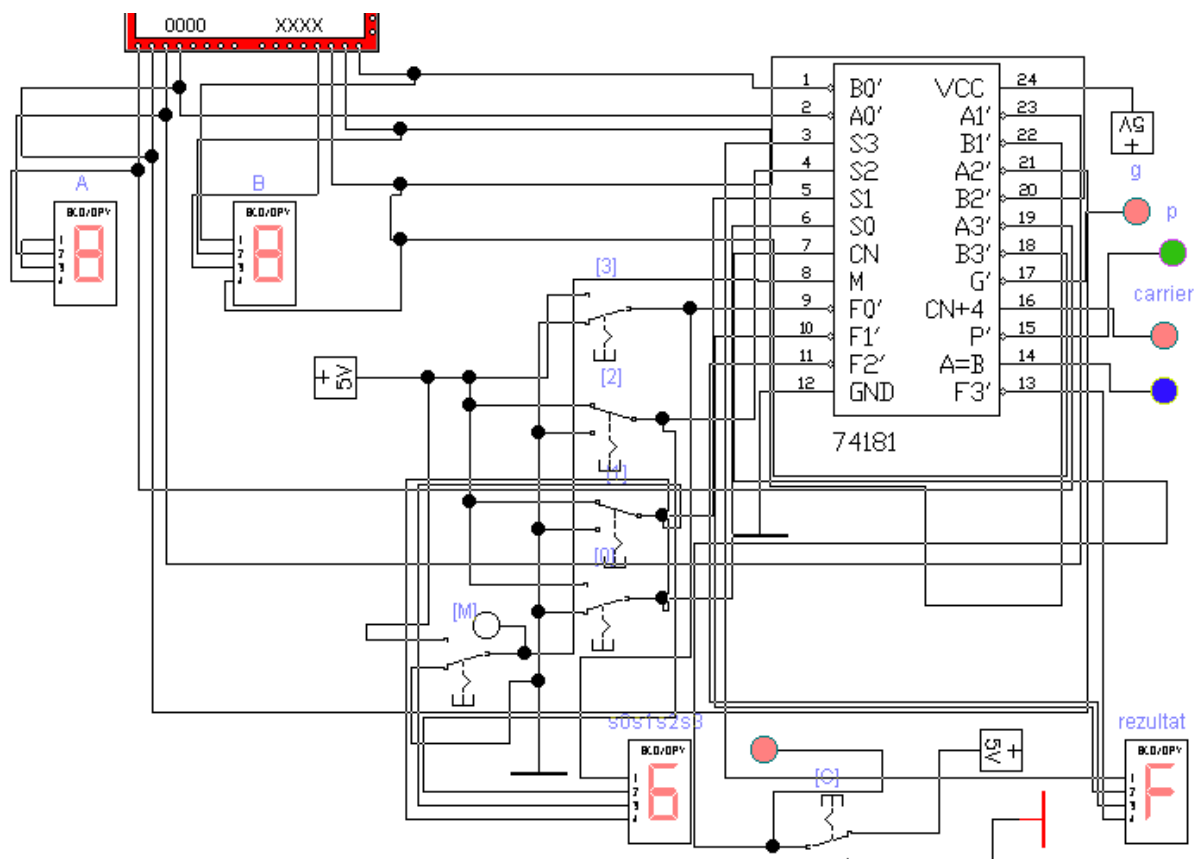


Рисунок 20.1 АЛУ

3.1 Создание схемы АЛУ в Electronic Work Bench 5.12

4 Контрольные вопросы.

- Назначение входа переноса в АЛУ.
- Чем отличаются логические операции от арифметических операций?
- Как можно выполнить операцию инкремента?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Схема работы АЛУ
- Моделирование работы логических и арифметических операций.
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №21.

Изучение процессов обработки информации на всех уровнях компьютерной архитектуры RISC на примере 32-разрядного - процессора 80960SA (Арифметические команды; Логические команды и команды ветвления; Основы подпрограмм; Основы прерываний)

Объём учебного времени – 2ч

Методические рекомендации

1.Цель работы

- Изучение архитектуры процессоров
- Познакомиться с интерфейсом учебной ЭВМ,
- Узнать методы ввода и отладки программы,
- Действия основных команд и способы адресации.
- Приобрести опыт программирования учебной ЭВМ в машинных кодах.

2.Перечень необходимых средств обучения:

2.15 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.16 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Программа на языке ЭВМ представляет собой последовательность команд. Код каждой команды определяет выполняемую операцию, тип адресации и адрес. Выполнение программы, записанной в памяти ЭВМ, осуществляется последовательно по командам в порядке возрастания адресов команд или в порядке, определяемом командами передачи управления.

Для того чтобы получить результат выполнения программы, пользователь должен:

- ✓ ввести программу в память ЭВМ;
- ✓ определить, если это необходимо, содержимое ячеек ОЗУ и РОН, содержащих исходные данные, а также регистров IR и BR;
- ✓ установить в РС стартовый адрес программы;
- ✓ перевести модель в режим **Работа**.

Ввод программы может осуществляться как в машинных кодах непосредственно в память модели, так и в мнемокодах в окно **программы** с последующим ассемблированием.

Команды в память учебной ЭВМ вводятся в виде шестизначных десятичных чисел, изменяющиеся в диапазоне "-99 999...+99 999", содержащие знак и 5 десятичных цифр. Старший разряд слова данных используется для кодирования знака: плюс (изображается как 0, минус (-) — как 1). Если результат арифметической операции выходит за пределы указан-

ного диапазона, то говорят, что произойдет переполнение разрядной сетки. АЛУ этом случае вырабатывает сигнал переполнения $OV = 1$. Деление на ноль вызывает переполнение. Результатом операции деления является целая часть частного.

3.1 Типы адресации

- ✓ **Прямая**, например, **add 23** – указание в команде непосредственно исполнительного адреса
- ✓ **Непосредственная**, например, **add #33** – которая заключается в указании в команде самого значения операнда, а не его адреса
- ✓ **Косвенная**, например, **add @33** – при которой в команде указывается адрес регистра или ячейки памяти, в которых хранится адрес операнда или его составляющие

3.3 Система команд

При рассмотрении системы команд ЭВМ обычно анализируют три аспекта: форматы, способы адресации и систему операций.

В форматах команд выделяется три поля: два старших разряда (0, 1) определяют код операции **COP**, разряд 2 может определять тип адресации, разряды [3:5] могут определять прямой или косвенный адрес памяти, номер регистра (в команде **MOV** номера двух регистров), адрес перехода или короткий непосредственный операнд. В двухсловных командах непосредственный операнд занимает поле [6:11].

Таблица 21.1 Система команд учебной ЭВМ

КОП	Мнемокод	Название	КОП	Мнемокод	Название
00	NOP	Пустая операция	23	ADD	Сложение
01	IN	Ввод $Acc \leftarrow IR$	24	SUB	Вычитание
02	OUT	Вывод $OR \leftarrow Acc$	25	MUL	Умножение
03	IRET	Возврат из прерывания	17	JNRZ	Цикл
10	JMP	Безусловный переход	30	MOV	Пересылка
11	JZ	Переход, если 0 ($Acc=0$)	36	DIV	Деление
12	JNZ	Переход, если не 0 ($Acc \neq 0$)	19	CALL	Вызов подпрограммы
14	JNS	Переход, если положительно	09	HLT	Стоп
13	JS	Переход, если отрицательно	41	RDI	Чтение
15	JO	Переход, если переполнение (если $Acc > 99999$)	43	ADI	Сложение
18	INT	Программное прерывание	44	SBI	Вычитание
21	RD	Чтение	45	MULI	Умножение
22	WR	Запись	46	DIVI	Деление

I — непосредственный операнд;

3.4 Программно-доступные регистры и флаги:

Acc — аккумулятор;

PC — счетчик адреса команды, содержащий адрес текущей команды;

SP — указатель стека, содержащий адрес верхушки стека;

RB — регистр базового адреса, содержащий базовый адрес;

RA — регистр адреса, содержащий исполнительный адрес при кос ной адресации;

IR — входной регистр;

OR — выходной регистр;

I — флаг разрешения прерываний.

3.5 Системные регистры и флаги:

DR — регистр данных АЛУ, содержащий второй операнд;

MDR — регистр данных ОЗУ;

MAR — регистр адреса ОЗУ;

RDR — регистр данных блока РОН;

RAR — регистр адреса блока РОН;

CR — регистр команд, содержащий поля:

✓ **COP** — код операции;

✓ **TA** — тип адресации;

✓ **ADR** — адрес или непосредственный операнд;

Z — флаг нулевого значения Acc;

S — флаг отрицательного значения Acc;

OV — флаг переполнения.

3.5.1 Задание к работе

Дана последовательность мнемочкодов, которую необходимо преобразовать в машинные коды, занести в ОЗУ ЭВМ, выполнить в режиме Шаг и зафиксировать изменение состояний программно-доступных объектов ЭВМ.

Запишите следующую программу в окно **Текст программы**

Таблица 21.2 Пример программы

Коды	Программа	Действие
21 1 020	rd #20	
22 0 030	wr 30	
23 0 005	add #5	
22 2 030	wr @30	
12 0 002	JNZ 002	

В меню **Вид** выберите **Программа**, нажмите кнопку **Компилировать**

Выполняя команду **Шаг** фиксируем изменения программно-доступных объектов (**Acc**, ячейки **ОЗУ 020 и 030**) в таблице 21.3 **Содержание регистров**:

Таблица 21.3 Содержание регистров для задачи 1

PC	Acc	M(30)	M(20)
000	000000	000000	000000
001	000020		
002		000020	
003	000025		
004			000025
002			
003	000030		
004			000030

В тетради заполните столбец **Действие** для каждой используемой команды программы. Нарисуйте алгоритм работы программы.

3.5.2 Запишите в ОЗУ программу, состоящую из 5 команд (из своего варианта). Команды разместите в ячейках памяти.

При необходимости установить начальное значение в устройство ввода IR.

Определить те программно-доступные объекты ЭВМ, которые будут изменяться при выполнении этих команд.

Выполнить в режиме Шаг введенную последовательность команд, фиксируя изменения значений объектов, определенных в п. 3.1.3, в таблице 21.3. Содержание регистров, по примеру:

Таблица 21.4 Содержание регистров для задачи 2

PC	Acc		
000			
001			
002			
003			

Если в программе образуется цикл, необходимо просмотреть не более двух повторений каждой команды, входящей в тело цикла.

Таблица 21.5 Задание для выполнения

Вариант №	IR	Команда 1	Команда 2	Команда 3	Команда 4	Команда 5
1	0000007	IN	MUL #2	WR 10	WR@10	JNS 001
2	x	RD #17	SUB #9	WR 16	WR @16	JNS 001
3	100029	IN	ADD #16	WR 8	WR @8	JS 001
4	x	RD #2	MUL #6	WR 11	WR @11	JNZ 00
5	000016	IN	WR 8	DIV #4	WR @8	JMP 002
6	x	RD #4	WR 11	RD @11	ADD #330	JS 000
7	000000	IN	WR 9	RD @9	SUB #1	JS 001
8	x	RD 4	SUB #8	WR 8	WR @8	JNZ 001
9	100005	IN	ADD #12	WR 10	WR @10	JS 004
10	x	RD 4	ADD #15	WR 13	WR @13	JMP 001

Напишите машинные коды команд, соответствующие варианту задания.

4 Контрольные вопросы.

- Что такое система команд ЭВМ?
- Способ представления данных в модели.
- Способы адресации. Рассмотреть на примере.
- Какие классы команд были использованы в задании?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Граф-схема алгоритма решения задачи.
- Распределение памяти (размещение в ОЗУ переменных, программы и констант).
- Программа с описанием действий.
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №22

Изучение процессов обработки информации на всех уровнях компьютерной архитектуры RISC на примере 32-разрядного - процессора 80960SA (Арифметические команды; Последовательная передача данных; Объём учебного времени – 2ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.
- Изучение архитектуры процессоров
- изучение способов способами взаимодействия процессора и внешних устройств в составе ЭВМ,
- научиться подключать внешние устройства в составе учебной ЭВМ
- составлять программы для вывода информации с клавиатуры на дисплей учебной ЭВМ

2.Перечень необходимых средств обучения:

3.6 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

3.7 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Связь процессора и ВУ может осуществляться в синхронном и асинхронном режиме.

Синхронный режим используется для ВУ, всегда готовых к обмену. В нашей модели такими устройствами являются **дисплей** и **тоногенератор** – процессор может обращаться к этим ВУ, не анализируя их состояние (правда дисплей блокирует приём данных после ввода 128 символов, формируя флаг ошибки).

Асинхронный обмен предполагает анализ процессором состояния ВУ, которое определяет готовность ВУ выдать или принять данные или факт осуществления какого-нибудь события, контролируемого системой. К таким устройством в модели является **клавиатура** и **блок таймеров**. Анализ состояния ВУ может осуществляться процессором двумя способами:

- в программно-управляемом режиме (предполагается обращение процессора к регистру состояния ВУ с последующим анализом значения соответствующего разряда слова состояния). Такое обращение следует предусмотреть в программе с некоторой периодичностью, независимо от фактического наступления контролируемого события (например, нажатия клавиши мыши)

- в режиме прерывания (при возникновении контролируемого события ВУ формирует процессору запрос на прерывание программы, по которому процессор и осуществляет связь в ВУ)

В состав **контроллера клавиатуры** входят три программно-доступных регистра:

- **DR** (адрес 0) — регистр данных;
- **CR** (адрес 1) — регистр управления, определяет режимы работы контроллера и содержит следующие флаги (устанавливаются и сбрасываются программно):
 - **E** — флаг разрешения приема кодов в буфер, при $E = 0$ контроллер игнорирует нажатие на клавиатуре, прием кодов в буфер не производится. На считывание кодов из буфера флаг **E** влияния не оказывает;
 - **I** — флаг разрешения прерывания, разрешает формирование запроса на прерывание от клавиатуры в момент установки флага готовности **Rdy**
 - **S** — флаг режима посимвольного ввода. При $S = 0$ флаг готовности **Rdy** формируется только после нажатия кнопки *Завершить ввод* в окне обозревателя клавиатуры, при $S = 1$ — после каждого нажатия клавиши;
- **SR** (адрес 2) — регистр состояния, содержит два флага, устанавливаемые контроллером «аппаратно»:
 - **Err** — флаг ошибки устанавливается при вводе в буфер 50-го символа, сбрасывается программно;
 - **Rdy** — флаг готовности, устанавливается в зависимости от значения флага **S**, сбрасывается «аппаратно» после выполнения команды ввода из регистра **DR** — считывание символа из буфера.

Дисплей представляет собой модель внешнего устройства, реализующую функции символьного дисплея. Дисплей может отображать символы, задаваемые ASCII-кодами, поступающими на его регистр данных.

Дисплей включает:

- видеопамять объемом 128 слов (ОЗУ дисплея);
- символьный экран размером 8 строк по 16 символов в строке;
- четыре программно-доступных регистра:
 - **DR** (адрес 0) — регистр данных;
 - **CR** (адрес 1) — регистр управления содержит следующие флаги (устанавливаются и сбрасываются программно):
 - **E** — флаг разрешения работы дисплея; при $E = 0$ запись в регистры **AR** и **DR** блокируется;
 - **A** — флаг автоинкремента адреса; при $A = 1$ содержимое **AR** автоматически увеличивается на 1 после любого обращения к регистру **DR** — по записи или чтению.
 - **SR** (адрес 2) — регистр состояния содержит единственный флаг *Err* — ошибки устанавливается аппаратно при попытке записать в регистр адреса число, большее 127, сбрасывается программно;
 - **AR** (адрес 3) — регистр адреса.

3.1 Задание к работе

3.1.1 Контроллер клавиатуры

Подключите внешнее устройство (меню Внешние устройства – Менеджер ВУ – Контроллер клавиатуры)

Для установки флага **E** в регистре **CR** требуется выполнить команды:

RD #10

OUT 1

Для установки **S** в «1» (режим посимвольного ввода) требуется выполнить следующие команды:

RD #103**OUT 1**

Для заданного текста найдите ASCII-код каждого символа, согласно варианта задания из таблицы 13 Варианты текста:

Таблица 22.1 Варианты текста

№ варианта	
1	В лесу родилась ёлочка, в лесу она росла.
2	Зимой и летом стройная, зелёная была.
3	Метель ей пела песенку: «Спи, ёлочка, бай-бай!»
4	Мороз снежком укутывал: «Смотри, не замерзай»
5	Трусишка-зайка серенький, под ёлочкой трусил.
6	Порою волк, сердитый волк, рысцою пробегал.
7	Теперь она нарядная на праздник к нам пришла.
8	И много, много радости детишкам принесла.
9	Наша Таня громко плачет, уронила в речку мячик.
10	Тише, Танечка, не плач – не утонет в речке мяч.

3.2 Контроллер дисплея

Подключите внешнее устройство (меню Внешние устройства – Менеджер ВУ – Контролер дисплея)

Пример программы, выводящей слово «Май» приведён в таблице 40 Пример программы, выводящей слово «Май»

Таблица 22.2 Пример программы, выводящей слово «Май»

Метка	Команда	Примечание
	RD #11	; включаем дисплей и устанавливаем
	OUT 11	; флаг автоинкремента;
	RD #0	; задаём начальный адрес
	OUT 13	; выводимого слова;
	RD #204	; ввод кода буквы «М»
	OUT 10	; вывод на дисплей
	RD #224	; ввод кода буквы «а»
	OUT 10	; вывод на дисплей
	RD #233	; ввод кода буквы «й»
	OUT 10	; вывод на дисплей
	HLT	

Выведите на экран один текстовый фрагмент вашего варианта.

Выведите данный текст – каждое слово на новой строке, строка должна начинаться со своего номера. Пример программы, выставляющей в начало каждой строки её номер приведён в таблице 22.3:

Таблица 22.3 Пример программы, выставляющей в начало каждой строки её номер

Метка	Команда	Примечание
	RD #10	; включаем
	OUT 11	; дисплей;
	RD #0	; задаём начальный
	WR R1	; адрес вывода;
	RD #49	; вводим код
	WR R2	; цифры «1»;
	RD #8	; вводим число
	WR R3	; повторений цикла;
M1:	RD R1	; читаем текущий адрес
	OUT 13	; и передаём в регистр адреса дисплея;
	ADD #16	; увеличиваем адрес на 16 – на начало следующей строки;
	WR R1	; сохраняем изменённый адрес;
	RD R2	; читаем код цифры – номер строки;
	OUT 10	; передаём код цифры на дисплей;
	RD @R2+	; увеличиваем содержимое R2 (код цифры) на единицу;
	JRNZ R3,M1	; декремент R3 и переход на начало цикла, если R3 $\neq$ 0;
	HLT	

Напишите программу для вывода на экран всех символов русского алфавита.

Выведите на экран один из трех текстовых сообщений, в зависимости от нажатой клавиши (1 – один текст, 2 – другой текст, 3 – третий текст). Текст выбрать по своему усмотрению.

Проанализируйте работу программы, дайте описание каждой команде в программе, приведённой в таблице 22.4:

Таблица 22.4 Примеры программ для работы с дисплеем

Вариант №	
1, 3, 5, 7, 9	RD #10 OUT 1 OUT 11 RD #103 OUT 1 M1:IN 2 JZ M1 IN 0 OUT 10 JMP M1

2, 4, 6, 8, 10	RD #10 OUT 11 RD #0 WR R1 RD #4 WR R2 M1:RD R1 OUT 13 ADD #16 WR R1 RD #42 OUT 10 JRNZ R2,M1 HLT
----------------	---

4 Контрольные вопросы.

- При каких условиях устанавливается и сбрасывается флаг готовности клавиатуры RD?
- По какой команде происходит вывод на дисплей символа?

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Граф-схема алгоритма решения задачи
- Распределение памяти (размещение в ОЗУ переменных, программы и констант)
- Программа с описанием действий
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №23.

Изучение процессов обработки информации на всех уровнях компьютерной архитектуры RISC на примере 32-разрядного - процессора 80960SA, Программирование UART
Объём учебного времени – 2ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.
- Изучение архитектуры процессоров
- Приобрести опыт программирования учебной ЭВМ в машинных кодах

2.Перечень необходимых средств обучения:

2.1Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.2 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Типы адресации

Прямая, например, **add 23** – указание в команде непосредственно исполнительного адреса

Непосредственная, например, **add #33** – которая заключается в указании в команде самого значения операнда, а не его адреса

Косвенная, например, **add @33** – при которой в команде указывается адрес регистра или ячейки памяти, в которых хранится адрес операнда или его составляющие

3.1 Оператор условного перехода

Оператор ветвления (условная инструкция, условный оператор) — оператор, конструкция языка программирования, обеспечивающая выполнение определённой команды (набора команд) только при условии истинности некоторого логического выражения, либо выполнение одной из нескольких команд (наборов команд) в зависимости от значения некоторого выражения.

Оператор ветвления применяется в случаях, когда выполнение или невыполнение некоторого набора команд должно зависеть от выполнения или невыполнения некоторого условия. Ветвление — одна из трёх (наряду с последовательным исполнением команд и циклом) базовых конструкций структурного программирования.

Условный оператор реализует выполнение определённых команд при условии, что некоторое логическое выражение (условие) принимает значение «истина» true.

Условный оператор в стандартной форме состоит из условия (логического выражения) и двух последовательностей операторов.

Алгоритмы, в которых в зависимости от выполнения некоторого логического условия происходит разветвление вычислений по одному из нескольких возможных направлений, называют **разветвляющимися**. Подобные алгоритмы предусматривают выбор одного из альтернативных путей продолжения вычислений. Каждое возможное направление вычислений называется *ветвью*. Логическое условие называют *простым*, если разветвляющийся процесс имеет две ветви, и *сложным*, - если процесс разветвляется на три и более ветви.

Любое сложное логическое условие может быть представлено в виде простых. Рассмотрим пример разветвляющегося алгоритма с простым логическим условием.

Пример. Даны два числа a и b .

Найти
$$x = \begin{cases} a/b, & \text{если } a > b \\ b/a, & \text{если } a \leq b \end{cases}.$$

Очевидно, что для определения ветви, по которому необходимо производить процесс вычисления значения x , достаточно проверить выполнение одного из условий, например $a > b$. Если условие $a > b$ не выполняется, то очевидно и без дополнительной проверки, что будет выполнено условие $a < b$. Следовательно, вариант схемы алгоритма будет выглядеть следующим образом:

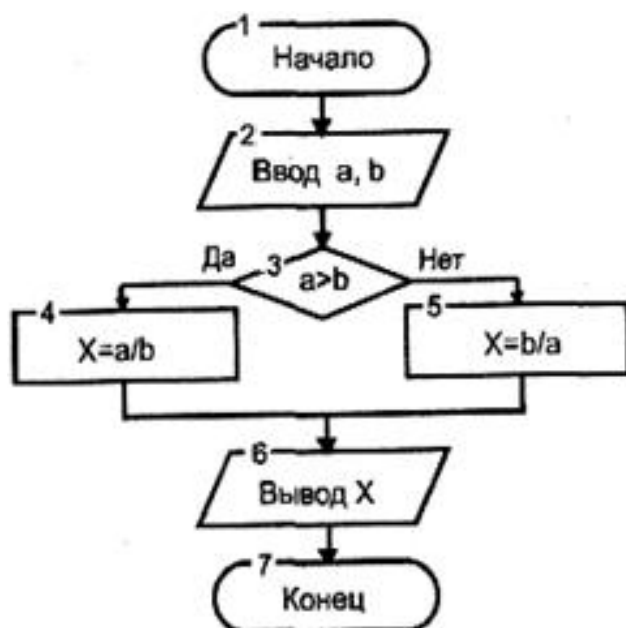


Рисунок 23.1 Схема простого разветвляющегося алгоритма

3.3 Задание к работе

Программирование разветвляющегося процесса. Разработать программу вычисления и вывода значения функции для вводимого из ИР значения аргумента x :

$$y = \begin{cases} f_i(x), & \text{при } x \geq a \\ f_j(x), & \text{при } x < a \end{cases}$$

Варианты заданий приведены ниже в таблице:

Таблица 23.4 Пример выполнения задания по блок-схеме

Мнемокод	Примечание
00) in	Ввод x
01) wr 30	Размещение x в ОЗУ
02) sub #16	Сравнение с границей – (x-16)
03) js m1	Переход по отрицательной разности
04) rd 30	Вычисления по первой формуле
05) sub #11	
06) wr 31	
07) mul 31	
08) sub #125	
09) jmp m1	
010) m1: rd 30	
011) mul 30	Переход на вывод результата
012) wr 31	Вычисления по второй формуле
013) rd 30	
014) mul #72	
015) add 31	
016) adi 106400	
017) divi 100168	
018) m2: out	
019) hlt	
	Вывод результата
	Стоп

4 Контрольные вопросы.

- Какие действия выполняют команды передачи управления?
- Как организовать безусловный переход в программе?
- Как поведёт себя программа из примера к третьему заданию, если в ней будет отсутствовать команда wr 31 по адресу 014

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Последовательность состояний регистров ЭВМ при выполнении программы в режиме Шаг для одного значения аргумента
- Письменные ответы на контрольные вопросы

6 Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №24.

Изучение процессов обработки информации на всех уровнях компьютерной архитектуры RISC на примере 32-разрядного -процессора 80960SA Цифро-аналоговое и аналого-цифровое преобразование (ЦАП и АЦП))

Объём учебного времени – 2ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.
- Изучение архитектуры процессоров
- Изучить теоретические сведения об аналого-цифровых и цифро-аналоговых преобразователях.
- Исследовать параметры аналого-цифрового и цифро-аналогового преобразователей
-

2.Перечень необходимых средств обучения:

2.1 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.2 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Устройства, в целом выполняющие преобразования аналоговых сигналов в цифровые и обратно, называются соответственно аналого-цифровыми и цифро-аналоговыми преобразователями (АЦП и ЦАП).

Преобразование ИКМ сигнала в аналоговый предусматривает последовательное выполнение основных операций:

- декодирование (преобразование ИКМ сигнала в АИМ сигнал);
- восстановление аналогового сигнала (выделение из спектра АИМ сигнала исходного сигнала).

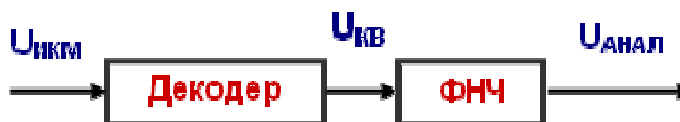


Рис. 24.1. Схема преобразования ИКМ сигнала в аналоговый сигнал

Временные диаграммы преобразований сигналов, происходящих в схеме рис. 24.1, показаны на рис. 24.2.

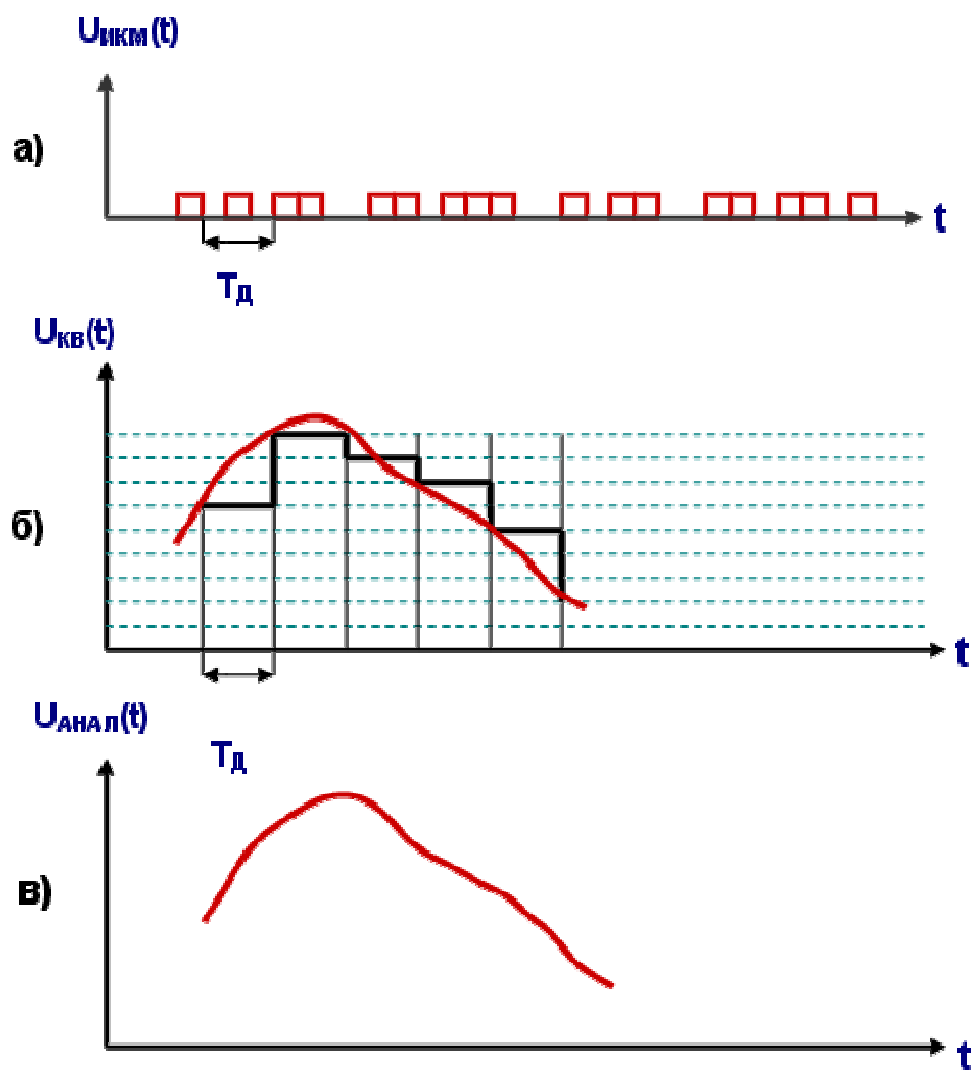


Рис. 24.2 – Диаграммы преобразования цифрового сигнала в аналоговый:
а – ИКМ сигнал; б – квантованный сигнал; в – аналоговый сигнал.

Из рисунка следует, что в схеме сначала восстанавливается непрерывность сигнала во времени, а затем из спектра ступенчатого сигнала выделяется спектр исходного аналогового сигнала.

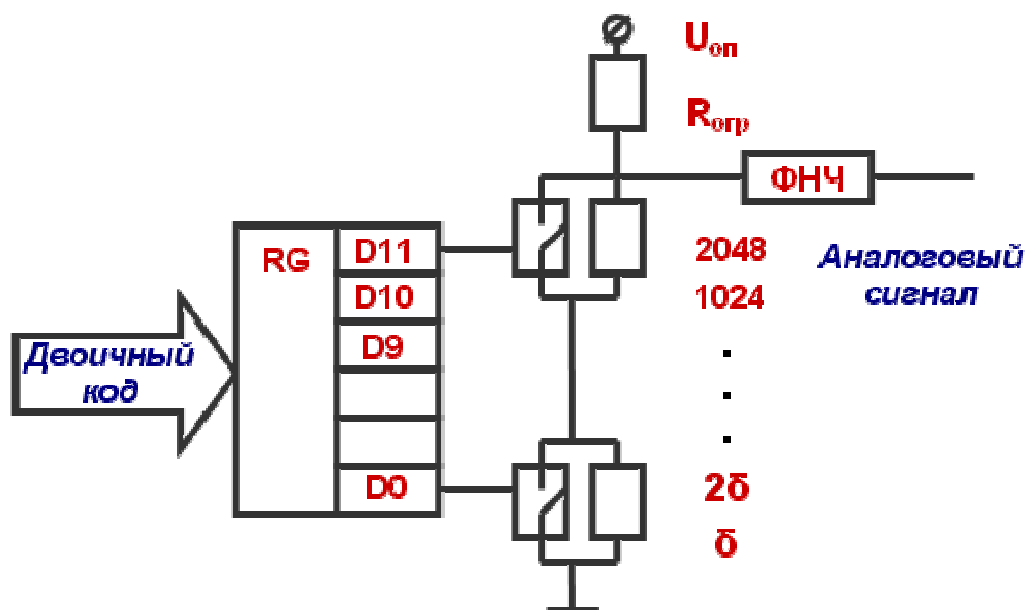


Рис.24.3. Структурная схема ЦАП

Рассмотрим принцип работы ЦАП (рис.24.3). Цифровой сигнал в двоичном коде (последовательном или параллельном) подается на буферный регистр RG. К выходам регистра RG подключены управляющие входы электронных ключей. К каждому из ключей подключены резисторы с сопротивлениями, соответствующими числу шагов квантования каждого из разрядов кодового слова цифрового сигнала. В зависимости от кодовой комбинации, т.е. включения и выключения соответствующих ключей, на входе ФНЧ будет присутствовать соответствующее напряжение. Смена кодовых комбинаций приведет к образованию на входе ФНЧ ступенчатого сигнала (рис. 2, б). ФНЧ выделяет исходный аналоговый сигнал.

Возможно построение АЦП на основе ЦАП. Схема такого АЦП показана на рис. 4. Аналоговый сигнал поступает на вход устройства выборки и хранения (УВХ), где подвергается дискретизации, т.е. преобразуется в сигнал АИМ. Этот сигнал поступает на один из входов схемы сравнения (СС), которая представляет собой компаратор, сравнивающий значения аналоговых сигналов на своих входах. Если значение сигнала на первом входе СС больше, чем на втором, то на выходе СС будет присутствовать сигнал логической 1, в противном случае – логического 0. Ко второму входу СС подключен аналоговый выход ЦАП.

Цифровые входы ЦАП подключены к порту вывода управляющего устройства (УУ), например микропроцессора. К порту ввода УУ подключен цифровой выход СС. Процесс квантования по уровню протекает следующим образом. Отсчет сигнала с выхода УВХ постоянно присутствует на нижнем по схеме входе СС. Устройство управления выполняет алгоритм приближения к данному значению, например, методом «золотого сечения». Сначала определяется значение старшего разряда кодового слова и далее до самого младшего. После определения самого младшего разряда схема готова к обработке следующего отсчета. Скорость работы схемы, т.е. частота дискретизации, зависит от скорости работы УУ и скорости преобразования ЦАП.

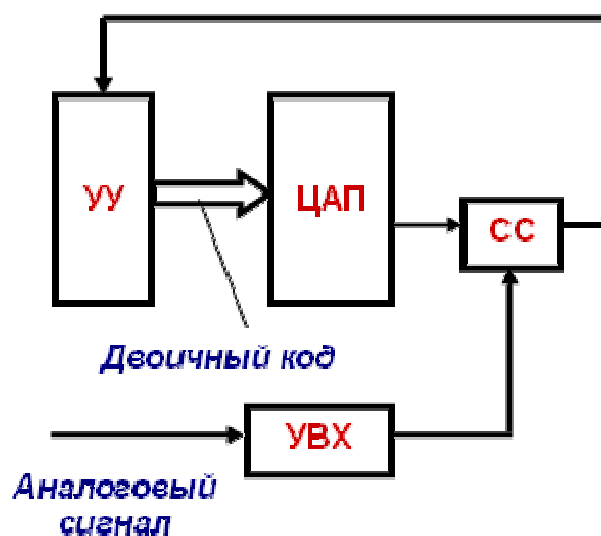


Рис.24.4. Структурная схема АЦП

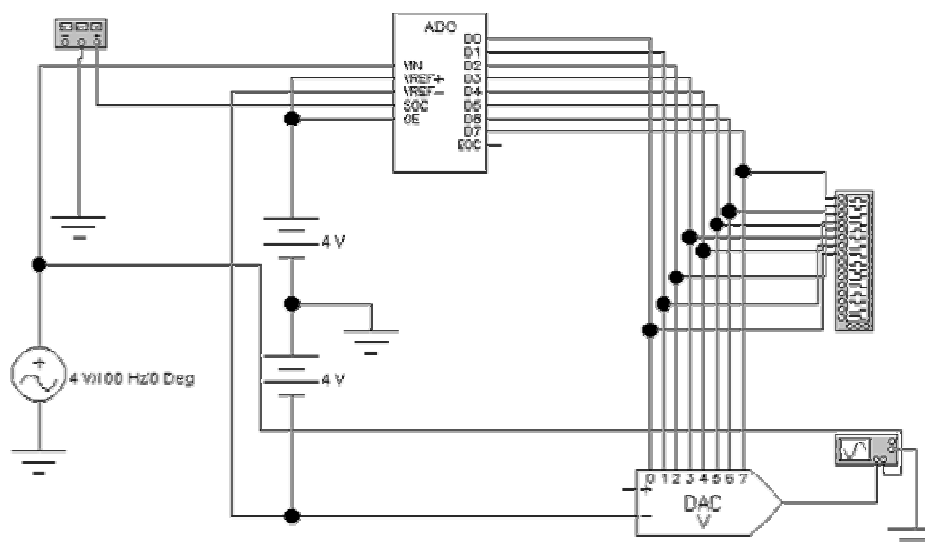


Рис.24.5. Структурная схема ЦАП.

3. 2. Исследование параметров цифро-аналоговых преобразователей.

1. Используя программу Elektronik Worbench синтезировать схему цифро-аналогового преобразователя, представленного на рис. 24.5..

2. Установить параметры исходного гармонического колебания, подаваемого на вход VIN АЦП: частота – 100 Гц, амплитуда – 4В; синхронизирующего сигнала, поступающего на вход SOC АЦП: частота 1-5 кГц, амплитуда 5В.

3. Для проведения исследований необходимо использовать АЦП и ЦАП из встроенной библиотеки Elektronik Worbench.

4. Визуально оцените точность воспроизведения аналогового сигнала из цифрового, поступающего от АЦП. Определите среднее значение [В] ошибки восстановления σ (рис.24.6).

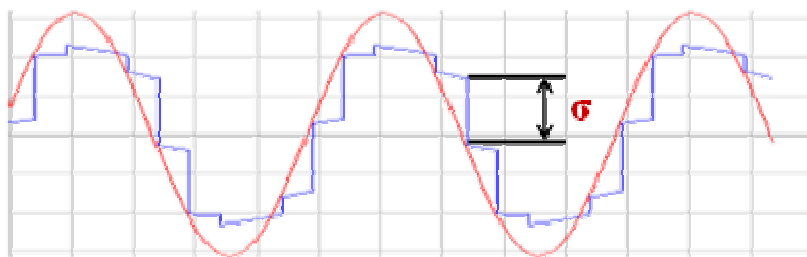


Рис 24.6. Внешний вид осциллограммы

6. Включите логический анализатор и проследите за цифровой информацией, поступающей на его входы (рис.24. 7) с аналого-цифрового преобразователя. Оцените длительность элемента в кодовых словах, представленных на логическом анализаторе и уясните взаимосвязь между частотой дискретизации (периодом дискретизации) и длительностью элемента кодовых слов.

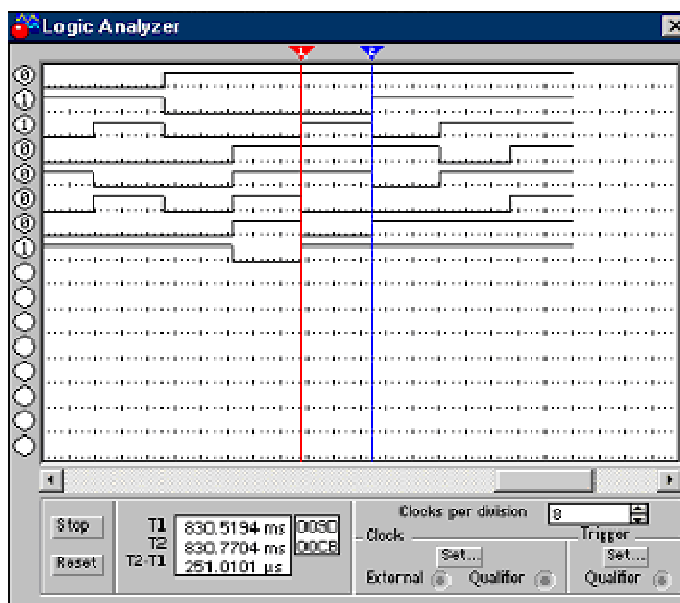


Рис.7. Внешний вид информации (кодовых слов), представленной в логическом анализаторе

7. Проведите исследование зависимости величины ошибки восстановления аналогового сигнала σ от значения частоты синхронизирующего колебания f_{SOC} , поступающего на вход SOC АЦП. Изменение $f_{SOC}=1 \dots 5$ кГц. Результаты занесите в таблицу 24.1.

Таблица 24.1. №

опыта	Частота дискретизации f , кГц	Ошибка восстановления σ , В
1	1	
2	2	
3	3	
4	4	
5	5	

8. Постройте график зависимости $\sigma = f(f_{SOC})$.

4 Контрольные вопросы.

- В каком частотном диапазоне лежат звуковые сигналы?
- Методы и типы АЦП.
- Статические параметры АЦП.
- Понятие дискретности, квантование, разрешающая способность.

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Объяснить принцип действия ЦАП, представленного на рис.3.
- Письменные ответы на контрольные вопросы

6. Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Практическая работа №25.

Изучение процессов обработки информации на всех уровнях компьютерной архитектуры RISC на примере 32-разрядного - процессора 80960SA (Среда Отладчика; Загрузка и отладка; Команды перемещения данных)

Объём учебного времени – 2ч

Методические рекомендации

1.Цель работы

- Изучение назначения и принцип работы логических узлов.
- Знакомство с базовыми устройствами из библиотеки EWB.
- Изучение архитектуры процессоров

2.Перечень необходимых средств обучения:

2.1 Технические средства обучения:

Компьютер Core i3 3.0, 4 Gb оперативной памяти, винчестер 250 Gb, DVD

2.2 Программное обеспечение:

- ОС Windows XP (7)
- Microsoft Office 2007
- Open Office 2007
- Программа «Учебная ЭВМ»
- Программа для моделирования Electronic Work Bench 5.12 и выше

3. Практические указания:

Обычно пользователь при приобретении нового компьютера хорошо знает лишь основные его внутренние характеристики, чаще всего это частота и тип процессора, объем системной памяти, видеопамяти и название графического чипа, емкость жесткого диска. Причем в последних трех случаях очень часто неизвестными остаются производители этих устройств, впрочем, как и ряд более тонких технических параметров. Если же компьютер был подарен, куплен по совету сторонних лиц либо предоставлен на работе, то пользователю ведомо о его конфигурации обычно еще меньше. Как тут со всем сразу разберешься, тем более что все и так вроде бы отлично функционирует? Иногда даже специалисты затрудняются дать однозначный ответ о тех или иных характеристиках, особенно, если это относится к столь распространенному виду комплектующих - «NONAME», а системные блоки надежно опломбированы, дабы долго оставались, соблюдены нерушимые гарантийные обязательства, а заодно и секретность их истинного содержания.

Но рано или поздно пользователю приходится разбираться с внутренностями своего электронного друга, и дело здесь не только в одной любознательности. Когда же наступает такой момент? Наиболее распространенным является случай, когда перестала загружаться операционная система и ее приходится переустанавливать с нуля. Естественно, никакая гарантия на ОС не распространяется. Но вот, наконец, пользователь переустановил ее своими силами - это только половина дела в конфигурировании системы. Далее идет черед за драйверами всех тех устройств, которые находятся внутри системного блока: материнской платы, видеокарты, звуковой платы, возможно, внутреннего модема или сетевой карты, иногда чего-то еще. Стандартные средства Windows не всегда могут полностью разрешить эту проблему. Тут уже нужно обладать определенными знаниями, чтобы правильно выбрать и установить

драйверы: сначала для чипсета материнской платы, потом и для всех остальных устройств. Если такая информация не была известна изначально, то ее придется добыть самостоятельно. К тому же делать это лучше заранее, а не дожидаться, пока гром грянет, и потом хвататься за все в экстренном порядке.

Если ваша система живет вполне благополучно, то даже в этом случае каждому любопытному пользователю будет интересно узнать, из чего же действительно состоит его компьютер. Выяснить это на самом деле не так уж и сложно, вам помогут специальные программы, предназначенные для определения конфигурации ПК, - и его не придется разбирать по частям.

Большинство подобных утилит предназначены для работы из-под Windows. Часто подобное ПО довольно функционально, предоставляя пользователю, возможность узнать конфигурацию не только аппаратной части, но и выяснить особенности операционной системы. К дополнительным возможностям относятся всевозможные тесты на производительность и работоспособность тех или иных компонентов. Хотя в последнем случае лучше пользоваться иными средствами - целевыми утилитами, специально разработанными для конкретных испытаний. Некоторые программы могут работать из-под OOS. Последнее может понадобиться при исследовании очень старых компьютеров, на которых Windows не установлена, что на сегодняшний день уже большая редкость. Гораздо чаще возникает другая ситуация - неработоспособность Windows. Тогда-то и выручает старенький DOS и диагностические программы под него, запущенные с загрузочной дискеты или диска.

3.1 Задание

- В подразделе **Summary** раздела **Computer** рассмотрите перечень основных компонентов, из которых состоит компьютер.
- В подразделе **Sensor** можно найдете перечень доступных датчиков средств аппаратного мониторинга, которыми оснащена материнская плата (но производители могут не задействовать их все), с выводом в окне их текущих показаний. Узнайте температуру процессора, чипсета, напряжения материнской платы и частоту вращения вентиляторов.

*Для более подробного рассмотрения каждого из перечисленных разделов нужно обратиться к соответствующим пунктам. Так, в подразделах **Motherboards** и **Chipset** дается довольно подробная техническая информация о соответствующих компонентах. Здесь можно найти даже геометрические размеры установленной в компьютере материнской платы.*

- В подразделе **Display** просмотрите характеристики видеокарты и монитора, видеоакселераторе.
- Запишите тип и характеристики видеочипа, на котором построена видеокарта, объем видеопамати, ее тип, рабочие частоты графического процессора и памяти.
- В подразделе **Fonts** просмотрите список установленных в системе шрифтов.
- В разделе **Operating System** AIDA32 рассмотрите основные характеристики Windows.
- В разделе **Processes** узнайте обо всех запущенных на текущий момент программах.
- В разделе **Software** просмотрите список всех проинсталлированных в системе приложений, узнайте о тех из них, что находятся в автозагрузке, проверьте зарегистрированные типы файлов.

4 Контрольные вопросы.

- Какая из утилит, по вашему мнению, наиболее эффективна для вывода информации о настройках компьютера?
- Какая из утилит, по вашему мнению, наиболее эффективна для вывода информации о настройках операционной системы?
- Дайте сравнительный анализ известных вам утилит по количественным показателям

5. Структура отчёта по практической работе

- Титульный лист.
- Содержание.
- Цель работы.
- Описание основных этапов работы
- Результаты выполнения работы;
- Письменные ответы на контрольные вопросы

6. Список рекомендуемой литературы:

- Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
- Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
- Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
- Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
- Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Интернет-ресурсы:

- 1.<http://novtex.ru/IT> - журнал "Информационные технологии"
- 2.<http://infojournal.ru> - журнал «Информатика и образование»
- 3.<http://www.compress.ru> - журнал «Компьютер пресс»

Информационное обеспечение обучения

Основная литература:

1. Глушаков С.В. Персональный компьютер: учеб.пособие для сред.проф.образования. -М.; Владимир: АСТ; ВКТ, 2008. - 475с.
2. Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. - М.: издательский дом «Академия», 2007. – 272с.
3. Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.: Форум: ИНФРА-М, 2012. – 512 с.
4. Микрюков В.Ю. Информация, информатика, информационные системы, компьютер, сети: учеб.пособие для сред. проф. образования.- Ростов-н/Д: Феникс, 2007.- 448с.
5. Таненбаум, Э. Архитектура компьютера/ Э. Таненбаум. – СПб.: Питер, 2007. – 848 с.

Дополнительная литература:

1. Гук М. Аппаратные интерфейсы ПК: Энциклопедия. – СПб.: Питер, 2010. - 528с.
2. Гук М. Аппаратные средства локальных сетей: Энциклопедия. – СПб.: Питер, 2010. - 634с.
3. Леонтьев В.П. Новейшая энциклопедия персонального компьютера.- ОЛМА-ПРЕСС Образование, 2006. — 734с.
4. Яшин В.Н. Информатика: аппаратные средства персонального компьютера: Учеб. пособие / – М.: ИНФРА-М, 2008. — 254с.

Интернет-ресурсы:

1. <http://novtex.ru/IT> - журнал "Информационные технологии"
2. <http://infojournal.ru> - журнал «Информатика и образование»
3. <http://www.compress.ru> - журнал «Компьютер пресс»

Лист регистрации изменений

Номер изме- нения	Номер листа				Всего лис- тов в доку- менте	ФИО и подпись ответст- венного за внесение изме- нения	Дата внесе- ния измене- ния	Дата введения изменения
	измененного	замененного	нового	изъятого				