

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Кабардино-Балкарский государственный
университет им. Х.М. Бербекова»
(КБГУ)

Институт электроники, робототехники и искусственного интеллекта

УТВЕРЖДАЮ

Руководитель ОПОП

 Р.Ш. Тешев


« 12 » февраля 2026 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПО ДИСЦИПЛИНЕ**

Б1.В.ДВ.07.01 «ЦИФРОВЫЕ УСТРОЙСТВА И МИКРОПРОЦЕССОРЫ»

Специальность

11.05.01 Радиоэлектронные системы и комплексы

Специализация

Радиоэлектронные системы передачи информации

Квалификация (степень) выпускника

Инженер

Форма обучения

Очная

Нальчик 2025

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Таблица 1

Код и формулировка компетенции	Индикаторы достижения	Планируемые результаты обучения по дисциплине (ЗУН)
<p>ПК-4. Способен к проведению диагностики, оценки качества и надежности в процессе эксплуатации радиоэлектронных систем и комплексов.</p>	<p>ПК-4.1. Способен учитывать специфику и особенности различного назначения радиоэлектронных систем и комплексов при оценке эффективности работы функциональных узлов и частей радиоэлектронной аппаратуры.</p> <p>ПК-4.2 Способен контролировать проведение диагностики и определять категории оценки качества на надежность, долговечность и безотказность работы радиоэлектронных систем и их составных частей.</p>	<p>Знать: специфику и особенности различного назначения радиоэлектронных систем и комплексов при оценке эффективности работы функциональных узлов и частей радиоэлектронной аппаратуры.</p> <p>Уметь: контролировать проведение диагностики радиоэлектронных систем и их составных частей.</p> <p>Владеть: методами оценки качества на надежность, долговечность и безотказность работы радиоэлектронных систем и их составных частей.</p>
<p>ПК-5. Способен осуществлять эксплуатацию и техническое обслуживание радиоэлектронных систем и комплексов.</p>	<p>ПК-5.1 Способен осуществлять эксплуатацию радиоэлектронных систем и комплексов.</p> <p>ПК-5.2 Способен осуществлять техническое обслуживание радиоэлектронных систем и комплексов.</p>	<p>Знать: аппаратуру обслуживаемых радиоэлектронных систем и комплексов и её функционирование.</p> <p>Уметь: аппаратуру обслуживаемых радиоэлектронных систем и комплексов и её функционирование.</p> <p>Владеть: навыками эксплуатации и технического обслуживания..</p>

2 Шкала оценивания планируемых результатов обучения

2.1 Текущий контроль

Оценка результатов текущей успеваемости в рамках контрольных точек осуществляется посредством 70-балльной системы, при этом за

добросовестное посещение занятий обучающийся может набрать до 10 баллов, за качественное прохождение оценочных мероприятий - до 60 баллов.

Таблица 2

Карта распределения рейтинговых баллов в рамках текущего контроля

№	Оценочное средство	Форма проведения	Порядок проведения	Максимальное количество баллов	Критерии оценивания
1	Лабораторная работа №1 «Знакомство с программой Logisim».	письменная	Работа включает в себя два задания, выполняется студентами попарно.	4	4- все задания выполнены верно, выводы по работе обоснованы; 3 - все задания выполнены верно, выводы по работе некорректны; 2 – задания выполнены частично или одно из заданий выполнено не верно, выводы содержат ошибки. 0 – задания не выполнены или все задания выполнены неверно
2	Лабораторная работа №2 «Комбинационные устройства. Минимизация».	письменная	Работа включает в себя два задания, выполняется студентами попарно.	4	4- все задания выполнены верно, выводы по работе обоснованы; 3 - все задания выполнены верно, выводы по работе некорректны; 2 – задания выполнены частично или одно из заданий выполнено не верно, выводы содержат ошибки. 0 – задания не выполнены или все задания выполнены неверно
3	Лабораторная работа №3 «Элементарные устройства памяти».	письменная	Работа включает в себя два задания, выполняется студентами попарно.	4	4- все задания выполнены верно, выводы по работе обоснованы; 3 - все задания выполнены верно, выводы по работе некорректны; 2 – задания выполнены частично или одно из заданий выполнено не верно, выводы

					содержат ошибки. 0 – задания не выполнены или все задания выполнены неверно
4	Лабораторная работа №4 «Декодер, мультиплексор и демультимплексор».	письменная	Работа включает в себя два задания, выполняется студентами попарно.	4	4- все задания выполнены верно, выводы по работе обоснованы; 3 - все задания выполнены верно, выводы по работе некорректны; 2 – задания выполнены частично или одно из заданий выполнено неверно, выводы содержат ошибки. 0 – задания не выполнены или все задания выполнены неверно
5	Лабораторная работа №4 «Арифметика. АЛУ».	письменная	Работа включает в себя два задания, выполняется студентами попарно.	4	4- все задания выполнены верно, выводы по работе обоснованы; 3 - все задания выполнены верно, выводы по работе некорректны; 2 – задания выполнены частично или одно из заданий выполнено неверно, выводы содержат ошибки. 0 – задания не выполнены или все задания выполнены неверно
6	Лабораторная работа №4 «Тактовые импульсы. Шина».	письменная	Работа включает в себя два задания, выполняется студентами попарно.	4	4- все задания выполнены верно, выводы по работе обоснованы; 3 - все задания выполнены верно, выводы по работе некорректны; 2 – задания выполнены частично или одно из заданий выполнено неверно, выводы содержат ошибки. 0 – задания не выполнены или все задания выполнены неверно
7	Лабораторная работа №4 «Сложные устройства памяти».	письменная	Работа включает в себя два задания, выполняется студентами	3	3- все задания выполнены верно, выводы по работе обоснованы;

	ОЗУ и ПЗУ».		попарно.		2 - все задания выполнены верно, выводы по работе некорректны; 1 – задания выполнены частично или одно из заданий выполнено не верно, выводы содержат ошибки. 0 – задания не выполнены или все задания выполнены неверно
8	Лабораторная работа №4 «Устройства ввода и вывода».	письменная	Работа включает в себя два задания, выполняется студентами попарно.	6	6- все задания выполнены верно, выводы по работе обоснованы; 4 - все задания выполнены верно, выводы по работе некорректны; 2 – задания выполнены частично или одно из заданий выполнено не верно, выводы содержат ошибки. 0 – задания не выполнены или все задания выполнены неверно
9	Тесты по 1 контрольной точке	с применением ДТ	Студент проходит компьютерное тестирование в ЭИОС.	7	Количество баллов пропорционально количеству правильных ответов
10	Тесты по 2 контрольной точке	с применением ДТ	Студент проходит компьютерное тестирование в ЭИОС.	7	Количество баллов пропорционально количеству правильных ответов
11	Коллоквиум по 1 контрольной точке	письменная	Студенты отвечают письменно на вопросы коллоквиума	8	10– ответы полные, точные, демонстрируют глубокое понимание темы, аргументация логична; 8 – ответы в основном правильные, но содержат незначительные ошибки; 6- ответы недостаточно полные; 4 – ответы частичные, содержат ошибки

					или требуют наводящих вопросов; 2-ответы не на все вопросы, частичные. 0 – ответы отсутствуют или полностью неверные.
12	Коллоквиум по 2 контрольной точке	письменная	Студенты отвечают письменно на вопросы коллоквиума	8	10– ответы полные, точные, демонстрируют глубокое понимание темы, аргументация логична; 8 – ответы в основном правильные, но содержат незначительные ошибки; 6- ответы недостаточно полные; 4 – ответы частичные, содержат ошибки или требуют наводящих вопросов; 2-ответы не на все вопросы, частичные. 0 – ответы отсутствуют или полностью неверные.
	Итого:			60	

Карта распределения баллов в рамках промежуточной аттестации

№	Оценочное средство	Форма проведения	Порядок проведения	Максимальное количество баллов	Критерии оценивания
1	Проведение зачета	Устный опрос	Билет содержит теоретических вопроса. На теоретические вопросы студент должен ответить устно.	2 Теоретические вопросы – 30 баллов.	Критерии оценивания теоретических вопросов: 25 до 30 баллов: Глубокий уровень владения материалом, точное знание ключевых концепций, способность анализировать и интерпретировать факты, грамотно строить высказывания, привести примеры, свободно

					<p>оперировать терминологией.</p> <p>От 19 до 24 баллов: Базовое владение предметом, умение последовательно раскрыть основную мысль вопроса, грамотное применение терминов, наличие существенных элементов анализа и обобщений, но недостаточное развертывание или отдельные неточности.</p> <p>От 13 до 18 баллов: Частичное освоение материала, попытка объяснить основной смысл вопроса, использование некоторых базовых терминов, но отсутствие глубокого понимания сложных моментов, логические недостатки изложения, отсутствие выводов.</p> <p>От 7 до 12 баллов: Ошибочные представления, слабо выраженное владение основными понятиями, значительные затруднения в интерпретации вопросов, существенные фактологические ошибки, отсутствие обоснованных выводов и примеров.</p> <p>От 0 до 6 баллов: Полное непонимание темы, неспособность сформулировать адекватный ответ, грубые ошибки, несоответствие требованиям задания.</p>
--	--	--	--	--	--

3. Оценочные материалы для текущего и промежуточного контроля успеваемости

3.1. Оценочные материалы для текущего контроля

Лабораторная работа № 1 «Знакомство с программой Logisim»

Logisim — это образовательный инструмент для разработки и моделирования цифровых логических схем. Её написал и продолжает активную разработку преподаватель Hendrix College, США, Carl Burch. Программа написана на Java и требует для работы установленную Java Runtime Environment. На момент выхода данного пособия последняя стабильная версия Logisim — 2.7.1, однако рекомендуется использовать тестовую версию 2.7.2.251, так как в ней исправлены некоторые существенные ошибки. Если же доступна более новая стабильная версия, то стоит использовать её. Официальная страница программы на русском языке: <http://cburch.com/logisim/ru>

Logisim является свободным программным обеспечением и распространяется на условиях лицензии GNU GPL. Это означает, что вы можете свободно использовать и распространять её в любых целях, читать и редактировать её исходный код, и распространять модифицированные версии программы на тех же условиях.

Интерфейс Logisim полностью переведён на русский язык, программа распространяется с подробнейшей справочной системой на русском языке. Данная глава пособия содержит только самые основные сведения о программе, необходимые для быстрого знакомства с ней. Почти все нюансы работы с программой описаны в справочной системе — от получения базовых навыков работы до подробного описания всех компонентов встроенной библиотеки.

Интерфейс программы состоит из строки меню, панели инструментов, панели проводника, таблицы атрибутов и холста. Панель проводника содержит список схем проекта и все инструменты загруженных библиотек. Любые инструменты можно выносить на панель инструментов. Состав панели инструментов настраивается в окне параметров проекта, которое вызывается из пункта «Параметры» меню

«Проект». На холсте располагаются все компоненты редактируемой в данный момент схемы. Холст покрыт сеткой, по которой выравниваются все компоненты и провода схемы. Таблица атрибутов содержит атрибуты выделенного в данный момент инструмента или компонента на холсте.

Лучший способ быстро освоить базовые навыки работы с Logisim — пройти все шаги «Пособия начинающего» из меню «Справка».

Logisim позволяет не только редактировать цифровые схемы, но и моделировать их поведение. Если флаг «Моделирование включено» из меню «Моделировать» установлен, то Logisim просчитывает процессы, происходящие в схеме прямо во время её редактирования: меняются значения на входах и выходах устройств, обновляется внутреннее состояние устройств памяти, устройства вывода отображают соответствующую информацию, а провода меняют цвет в зависимости от проходящих по ним значений. Для схем, на которых присутствуют

тактовые генераторы, можно либо осуществлять моделирование потактово (пункт «Один такт» из меню «Моделировать»), либо установить максимальную тактовую частоту (пункт «Тактовая частота») и запустить продолжительное моделирование схемы (пункт «Такты включены»). Вернуть схему в исходное состояние можно с помощью пункта «Сбросить моделирование».

1.1. Подсхемы

Одна из важнейших возможностей Logisim — создание подсхем. Идея подсхем аналогична идее подпрограмм в программировании: они служат для разбиения большой работы на более мелкие части, для повторного использования уже спроектированных частей схем и для упрощения процесса отладки.

Добавить схему можно с помощью пункта «Добавить схему...» из меню «Проект». Одна из схем в проекте обязательно является главной — она открывается первой при загрузке проекта. После запуска моделирования будет выполняться моделирование схемы, просматриваемой в данный момент. Чтобы включить схему в качестве подсхемы в другую схему, нужно выделить её и расположить на холсте, как и любой другой элемент. При этом образуется иерархия вложенных подсхем, верхним уровнем которой является просматриваемая схема. Чтобы редактировать схему, нужно дважды щёлкнуть на её названии в панели обозревателя. Располагать несколько копий одной схемы — совершенно нормальная практика. При этом редактирование этой схемы будет приводить к изменениям во всех копиях, но моделирование поведения каждой копии происходит автономно.

Передача сигналов между какой-либо схемой и расположенной в ней подсхемой происходит через контакты. Контакты бывают входные и выходные. По умолчанию на панели инструментов присутствуют оба вида контактов. Также компонент «Контакт» есть во встроенной библиотеке «Проводка». В цифровой технике при соединении элементов проводами всегда определено, какой элемент передаёт сигнал, а какой принимает. Входной контакт подсхемы будет принимать сигнал из внешней схемы и передавать его в подсхему. Выходной контакт подсхемы будет передавать сигнал из подсхемы во внешнюю схему.

Контакты подсхемы при расположении её копии на холсте другой схемы будут располагаться с той стороны «корпуса» подсхемы, в какую сторону направлен контакт (атрибут «Направление») внутри подсхемы. Очень важно давать чёткие осмысленные имена контактам (атрибут

«Метка»), тогда подключить к расположенной на холсте копии подсхемы провода будет очень легко.

Для верхней схемы в иерархии всегда показывается реальное текущее состояние во время моделирования. Однако чтобы наблюдать текущее состояние конкретной копии подсхемы, расположенной в иерархии ниже, нужно щёлкнуть правой кнопкой мыши на этой копии на холсте и выбрать в контекстном меню пункт «Рассмотреть...».

Logisim имеет небольшой встроенный редактор векторной графики для изменения внешнего вида подсхем при добавлении их в другие схемы. Он также позволяет изменять расположение контактов подсхемы. Этот редактор вызывается через пункт «Редактировать внешний вид схемы» из меню «Проект» или нажатием соответствующей кнопки на панели инструментов.

1.2. Провода

Располагать на холсте провода можно инструментами «Правка» или «Проводка». Провод и контакт элемента считаются соединёнными, если они располагаются строго в одной и той же точке сетки. То же относится и к соединению провода с другим проводом. При перемещении по холсту уже подключенных элементов, Logisim будет сохранять соединения, если это возможно. Этого можно избежать, если при перетаскивании нажать клавишу Shift.

Ещё одна важная функция Logisim — возможность собирать провода в пучки. Пучки обозначаются на схемах чёрным цветом. Очень многие компоненты библиотеки имеют атрибут «Разрядность данных». Этот атрибут может принимать значение от 1 до 32, и если он больше 1, то подключение провода к соответствующему контакту компонента автоматически создаст пучок проводов. Естественно, нужно соединять контакты с одинаковой разрядностью данных. Разрядность данных можно назначать также входным и выходным контактам схем, что значительно уменьшает количество проводов на схеме, а значит, повышает её удобочитаемость.

Когда возникает потребность разделить пучок проводов на отдельные провода или более мелкие пучки (или наоборот — собрать их в один пучок), нужно применить компонент «Разветвитель» из встроенной библиотеки «Проводка». Этот компонент также позволяет назначать в каком порядке провода входят в пучок — это настраивается в таблице атрибутов разветвителя.

Провода в Logisim могут иметь один из семи цветов, каждый из которых несёт определённую информацию о проводе.

- Серый — разрядность провода неизвестна (он не подключен ни к каким компонентам)
- Синий — провод несёт однобитный сигнал, но его состояние не

определено.

- Тёмно-зелёный — провод несёт однобитный сигнал, в данный момент этот сигнал — «0».
- Светло-зелёный — провод несёт однобитный сигнал, в данный момент этот сигнал — «1».
- Чёрный — провод несёт многобитный сигнал (его разрядность больше единицы). При этом любое количество отдельных однобитных проводов в пучке могут иметь неопределённое состояние.
- Красный — провод содержит значение «ошибка». Это происходит, когда провод замыкает сразу несколько выходов компонентов, и на выходах разное значение сигнала. На практике это означает короткое замыкание.
- Оранжевый — провод подключён к выходам компонентов, имеющим разную разрядность.

Выделение цветом проводов, несущих «0» или «1» позволяет наблюдать изменение значений на выходах компонентов и эффективно отлаживать схему.

1.3 Комбинационный анализ

Ещё одна замечательная функция Logisim — возможность проводить анализ и синтез комбинационных логических устройств.

Если схема не содержит последовательностных логических устройств, контактов разрядностью больше 1 и количество входных и выходных контактов не больше 12, то для схемы можно провести комбинационный анализ. Это можно сделать через пункт «Анализировать схему» из меню «Проект». В открывшемся окне «Комбинационный анализ» присутствуют вкладки «Входы», «Выходы», «Таблица», «Выражение» и «Минимизация». Вкладка «Таблица» содержит полную таблицу истинности, полученную для схемы; вкладка «Выражение» содержит список выражений каждой из булевых функций для выходных контактов. Вкладка «Минимизация» содержит минимизированные таблицы истинности и выражения для булевых функций. И выражения и таблицы истинности можно свободно редактировать, копировать и вставлять.

Открыв то же окно «Комбинационный анализ» из меню «Окно», можно ввести во вкладках «Входы» и «Выходы» имена входных и выходных контактов для будущей схемы; ввести таблицу истинности или выражения булевых функций, нажать кнопку «Построить схему», и указать название новой схемы. Logisim синтезирует схему, используя логические элементы.

1.4. Библиотека компонентов

Огромным преимуществом программы Logisim является обширная библиотека компонентов. Далее даётся краткое описание каждого компонента библиотеки. Если компонент является реализацией типичного для цифровой техники устройства, то никакого описания не приводится. Для получения подробной информации по каждому компоненту следует обратиться ко встроенной справке (пункт «Справка по библиотеке...» из меню «Справка»).

Библиотека «Проводка»

- Разветвитель — позволяет разделять пучки на отдельные провода и объединять отдельные провода в пучки.
- Контакт — передаёт значения между подсхемой и содержащей её внешней схемой.
- Датчик — отображает значение в точке схемы, к которой подключен. Позволяет выбирать основание системы счисления для отображения.
- Тоннель — позволяет соединять удалённые точки схемы без протягивания провода.
- Согласующий резистор — меняет в определённую сторону значение на проводе, к которому подключен, если значение на нём не определено.
- Тактовый генератор — формирует прямоугольные импульсы.
- Константа — источник постоянного значения
- Питание — источник логической единицы
- Земля — источник логического нуля
- Транзистор — полевой транзистор, работающий в режиме электронного ключа
- Передаточный вентиль — комбинация двух комплементарных полевых транзисторов
- Расширитель битов — преобразует значение в значение с другой разрядностью. При увеличении разрядности позволяет выбирать, чем заполнять старшие биты.

Библиотека «Элементы»

- Элемент НЕ
- Буфер — повторитель сигнала («элемент «ДА»)
- Элемент И

- Элемент ИЛИ
- Элемент И-НЕ
- Элемент ИЛИ-НЕ
- Элемент Исключающее ИЛИ
- Элемент Исключающее ИЛИ-НЕ
- Нечётность — на его выходе единица, если единица на начётном количестве входов
 - Чётность — на его выходе единица, если единица на чётном количестве входов
 - Управляемый буфер — позволяет создавать на его выходе неопределённое значение
 - Управляемый инвертор — то же, что и управляемый буфер, но выполняющий инвертирование выхода.

Библиотека «Плексоры»

- Мультиплексор
- Демультиплексор
- Декодер
- Шифратор приоритетов — выдаёт номер старшего входа, на котором «1»
- Селектор битов — разделяет многоразрядный вход на несколько групп битов, и пускает на выход выбранную управляющим входом группу

Библиотека «Арифметика»

- Сумматор
- Вычитатель
- Множитель
- Делитель
- Отрицатель — выдаёт значение в дополнительном коде, противоположное по знаку входному значению
- Компаратор — сравнивает два числа
- Сдвигатель — реализация левого и правого сдвига
- Сумматор битов — выдаёт количество единиц, поступающих на его многоразрядные входы
 - Искатель битов — выдаёт номер старшего или младшего нуля или единицы в многоразрядном входном значении.

Библиотека «Память»

- D-триггер

- Т-триггер
- JK-триггер
- RS-триггер
- Регистр
- Счётчик
- Сдвиговый регистр
- Генератор случайных чисел
- ОЗУ
- ПЗУ

Библиотека «Ввод/вывод»

- Шестнадцатеричный индикатор — 7-сегментный индикатор, выводящий шестнадцатеричное значение, поступающее на его 4-разрядный вход.
 - Светодиодная матрица
 - Терминал — отображает поступающие на его вход ASCII коды в виде строк текста

Библиотека «Базовые»

- Инструмент Нажатие — инструмент для взаимодействия с компонентами схемы: нажатия кнопок, движения джойстика, установки значений контактов, просмотра состояния пучков проводов, и т.д.
- Инструмент Правка — основной инструмент для редактирования схем: перемещения компонентов и добавления проводов. Этот инструмент совмещает в себе функции четырёх инструментов, описанных ниже (можно считать их устаревшими).
 - Инструмент Выбор — инструмент для выделения элементов схемы.
 - Инструмент Проводка — инструмент для добавления проводов.
 - Инструмент Текст — инструмент для добавления текстовых полей и редактирования меток компонентов.
 - Инструмент Меню — инструмент для вызова контекстного меню компонентов.
 - Метка — инструмент для добавления текстовых меток в любую точку холста.

1.5. Различные возможности

В этом параграфе приводится список различных дополнительных возможностей Logisim с краткими описаниями. Более подробную информацию о каждой возможности можно получить в соответствующем разделе справки.

Любой файл проекта Logisim можно загрузить в другой проект в качестве библиотеки. В таком случае все его схемы будут доступны как компоненты, но изменять их внутреннюю структуру будет нельзя. Это осуществляется через пункт «Загрузить библиотеку: Библиотека Logisim...» из меню «Проект».

Существует возможность писать на языке Java собственные библиотеки компонентов, обладающие всем набором функций, доступных в Logisim. Руководство по написанию таких библиотек — в разделе «Библиотеки JAR» руководства пользователя Logisim.

Моделировать поведение схем можно без запуска графического интерфейса пользователя Logisim. Существует множество параметров для запуска и автоматизированного моделирования схем из командной строки. Подробнее — в разделе «Проверка из командной строки» руководства пользователя Logisim.

Растровые изображения схем можно экспортировать в файлы — как индивидуально, так и группами. Это осуществляется через пункт «Экспортировать изображение...» из меню «Файл».

Любой инструмент можно привязать к определённой комбинации клавиш Shift, Control и Alt и кнопок мыши. Это настраивается на вкладке «Мышь» окна параметров проекта (пункт «Параметры...» из меню «Проект»).

Любой файл проекта Logisim можно использовать как шаблон для создания новых проектов. Это удобно, если у вас есть файл проекта с настроенной панелью инструментов, поведением мыши, и т.д. Это осуществляется на вкладке «Шаблон» окна настроек приложения (пункт «Настройки...» из меню «Файл»).

Графические обозначения логических элементов можно выбрать на вкладке «Международные» окна настроек приложения.

Отображение фактической тактовой частоты моделирования можно включить на вкладке «Окно» окна настроек приложения.

Полную статистику по типам и количеству компонентов, содержащихся в схеме и входящих в её состав подсхемах можно получить через пункт «Получить статистику схемы» из меню «Проект».

Существует возможность просматривать в виде дерева иерархию подсхем и перемещаться по ней. Для перехода в такой режим просмотра нужно выбрать пункт «Показать дерево моделирования» из меню «Проект», или нажать соответствующую кнопку на панели инструментов.

Можно прослеживать распространение сигналов по схеме; это может быть удобно при отладке схем. Для этого нужно в меню «Моделировать» снять флажок «Моделирование включено», затем выбрать пункт «Сбросить моделирование», и далее последовательно выбирать пункт «Шаг моделирования», наблюдая распространение сигнала.

Изменение во времени значений на контактах и некоторых компонентах схемы можно автоматически записывать в виде таблицы в файл. Подробнее — в разделе «Запись в журнал» руководства пользователя Logisim.

Лабораторная работа 2. Комбинационные устройства. Минимизация

Для выполнения этой лабораторной работы необходимо внимательно прочитать и осмыслить главу 2 и параграфы 1.3 и 1.4 данного пособия.

Задание 1. Спроектировать в Logisim комбинационное устройство, имеющее несколько входов и две группы по несколько выходов. Количество входов и выходов, а также поведение устройства для каждого варианта приведены в таблице 3.1. Вариант определяется по последней цифре номера студента в списке группы.

Под «значением на входах» и «значением на выходах» понимается некоторое целое число, двоичное представление которого формируется отдельными входными или выходными битами. Последний бит в такой записи — младший (бит 0). Состояния на выходах каждой из двух групп зависят от состояний одних и тех же входов. Таким образом, для варианта 1, например, нужно реализовать $1 + 3 = 4$ булевых функции от 5 переменных. Нумерация битов в многобитных значениях всегда начинается с нуля (с младшего бита). Однако нумерация обычных объектов (дней недели, букв, и т.п.) начинается с единицы. Если на входы поступает бессмысленное значение (например, 14 в качестве номера месяца), то значение на выходах может быть произвольным («don't care»).

Каждая из реализаций устройства должна быть составлена в Logisim на логических элементах в виде схемы с осмысленными названиями входов и выходов. Для отдельных битов многобитных значений названия должны оканчиваться на номер бита (например day2, day1, day0 (младший бит в конце) для варианта 4). Для составления таблиц истинности, их минимизации методом Куайна — Мак-Класки и генерации схемы используется модуль Logisim «Комбинационный анализ».

Для защиты работы нужно представить:

1. Таблицу истинности устройства, реализацию ДНФ, минимизированную методом Куайна—Мак-Класки.
2. Реализацию, предложенную Logic Friday, а в случае её чрезмерной

громоздкости — собственную (полученную в результате применения тождеств булевой алгебры), содержащую значительно меньше элементов, чем реализация из пункта 2.

Задание 2. Повторить все пункты задания 1 с учётом следующей информации: ни при каких обстоятельствах на входы устройства не могут прийти одновременно все нули или все единицы. Это обстоятельство позволяет существенно упростить схему устройства (дополнительно минимизировать булевы функции), используя значения «don't care».

Требования к выполнению работы: все задания выполняются в одном файле проекта Logisim; каждое самостоятельное устройство должно быть оформлено в виде отдельной схемы с осмысленным названием всех входов и выходов и самой схемы. Для демонстрации работоспособности всех реализаций нужно составить в Logisim схему, содержащую в качестве подсхем все реализации устройства (их будет четыре), и одновременно подающую на их входы одно и то же тестовое значение, при этом значения на выходах различных реализаций должны совпадать. Для этого потребуется объединить с помощью компонента Разветвитель отдельные биты значений на входах и выходах подсхем в пучки проводов и показать многобитные значения на них, подключив несколько компонентов «Датчик», установив значение «Беззнаковое десятичное» для их атрибута «Основание». Так как старшие биты на входах и выходах подсхем расположены сверху, для всех разветвителей нужно из контекстного меню выбрать пункт «Расставить по убыванию».

Дополнительные вопросы и задания:

1. Насколько дополнительное обстоятельство из задания 2 упростило схемы с точки зрения количества элементов? А с точки зрения задержек (см. параграф 1.5)?

2. Позволила ли реализация, предложенная Logic Friday (или ваша собственная), сократить общую задержку устройства?

3. По какой формуле можно рассчитать общее количество ячеек в таблице истинности, зная количество входов и выходов комбинационного устройства?

4. Сравните своё устройство с устройствами из других вариантов: какие устройства сложнее в реализации — с большим количеством входов или большим количеством выходов?

5. Просмотрев задания всех вариантов, попробуйте предположить, какие из этих устройств действительно могут использоваться в цифровой технике, а какие приведены в задании лишь «для интереса».

6. Попробуйте реализовать своё устройство только на элементах И-НЕ. Стало ли оно от этого проще? Зачем это может понадобиться на практике?

Таблица 1. Варианты заданий для лабораторной работы 1

Вариант	Кол-во входов	Кол-во выходов	Поведение устройства
1	5	1	Значение на входах — номер буквы в латинском алфавите. На выходе «1», если эта буква — согласная. В остальных случаях — «0».
		3	Значение на выходах — номер самого старшего входа, несущего «1».
2	4	3	Значение на выходах — количество входов, несущих «0».
		2	Значение на выходах — остаток от деления значения на входах на 3, увеличенный на 1.
3	4	2	Значение на входах — номер месяца. На выходах — количество слогов в названии этого месяца.
		3	Значение на выходах — модуль разности количества нулей и единиц на входах.
4	5	3	Значение на входах — число в декабре 2012 года. На выходах — номер дня недели для этого числа.
		1	На выходе «0», когда на входе больше нулей, чем единиц. В остальных случаях — «1».
5	3	8	Значение на выходах — факториал значения на входах, делённый на 21 и округлённый до целых.
		2	Значение на выходах — номер самого младшего входа, несущего «0».
6	5	3	Значение на выходах — количество входов, несущих
		1	На выходе «1», когда на входах нет ни одной пары нулей, идущих подряд. В остальных случаях — «0».
		3	Значение на входах — номер дня недели. На выходах

7	3	7	Когда на входах значение «7» — на всех выходах единицы. В любом другом случае единица только на одном выходе. Номер этого выхода задаётся значением на входах.
8	5	2	Значение на выходах — количество входов, несущих «0», не соседствующих с другими входами,
		2	Значение на выходах — остаток от деления значения на входах на 3.
9	5	1	На выходе «0», когда на входе число, большее 21. В остальных случаях — «1».
		3	Значение на выходах — количество цифр в записи римскими цифрами значения на входах.
10	4	4	Значение на выходах — остаток от деления значения на входах на 11.
		1	На выходе «1», если значение на входах в двоичном представлении симметрично. В остальных случаях —

Лабораторная работа 3. Элементарные устройства памяти

Для временного хранения информации в цифровых схемах применяют различные устройства памяти. Из простейших устройств памяти мы рассмотрим три класса: триггеры, регистры и счётчики. Все эти устройства есть во встроенной библиотеке Logisim «Память». Они могут быть и асинхронными (см. параграф 1.6 пособия), но в рамках данного курса мы будем рассматривать только синхронные. У каждого синхронного устройства есть синхронизирующий (тактовый) вход. Любое изменение внутреннего состояния устройства происходит только в тот момент, когда уровень сигнала на тактовом входе меняется (в таком случае говорят, что тактовый вход срабатывает).

Триггер $\bar{\text{D}}$ — простейшее устройство памяти (а значит, последовательностное устройство), хранящее один бит информации. Иными словами, триггер может иметь только два разных внутренних состояния — «0» или «1». В англоязычной литературе триггер называют «flip-flop». Существует четыре вида триггеров: D (data), T (toggle), JK (jump-kill) и RS (reset-set). Эти названия даны по названиям входов триггеров. Кроме этих входов каждый триггер имеет два выхода — Q (прямой) и Q' (инверсный); значение на прямом выходе всегда совпадает со внутренним состоянием триггера, а значение на инверсном — противоположное. Каждый из четырёх типов триггеров имеет разное поведение. Таблицы истинности для них приведены в таблице 3.2. Q' означает значение, противоположное значению, хранимому в триггере в данный момент.

Таблица 3.2. Триггеры

D-триггер		T-триггер		JK-триггер			RS-триггер		
D	Q	T	Q	J	K	Q	S	R	Q
0	0	0	Q	0	0	Q	0	0	Q
1	1	1	Q'	0	1	0	0	1	0
				1	0	1	1	0	1
				1	1	Q'	1	1	?

Можно дать словесное описание поведения триггеров:

- **D-триггер:** когда тактовый вход срабатывает, значение, хранящееся в триггере, мгновенно становится значением входа D (данные).

- **T-триггер:** когда тактовый вход срабатывает, значение, хранящееся в триггере, меняется или остаётся прежним в зависимости от того, какое значение на входе T (переключение):

«1» или «0».

- **JK-триггер:** когда тактовый вход срабатывает, значение, хранящееся в триггере, меняется, если на входах J и K единица; остаётся прежним, если на них 0; если значения на них различны, то значение становится единицей, если на входе J (прыжок) — «1»; или нулём, если на входе K (забой) — «1».

- **RS-триггер:** когда тактовый вход срабатывает, значение, хранящееся в триггере, остаётся неизменным, если на входах R и S — «0»; становится «0», если на входе R (сброс) — «1», и становится «1», если на входе S (установка) — «1». Поведение не определено, если на обоих входах «1». (В Logisim значение триггера остается неизменным.)

Физически триггеры реализуются на логических элементах (то есть в конечном итоге на транзисторах в составе интегральных схем), включенных, как правило, не совсем обычным для них способом — их выходы так или иначе соединяются с их входами. Как говорилось выше, триггеры могут быть синхронными и асинхронными. Тактовый вход триггеров и других устройств памяти в Logisim обозначается треугольником; в случае, когда вход нужно пометить буквой или строкой, используют «C» или «Clock». Синхронные триггеры, как правило, содержат большее количество логических элементов. Иногда синхронными ошибочно называют также особую разновидность триггеров, имеющих разрешающий вход. Изменение внутреннего состояния такого триггера происходит, когда на разрешающем входе «1». Этот вход иногда называют синхронизирующим, но на самом деле он таковым не является. На схемах ниже разрешающий вход отмечен буквой «E» (от англ. enable).

Асинхронные триггеры (с разрешающим входом или без него) иногда называют «прозрачными» (чаще в англоязычной литературе — “transparent”), а синхронные — «непрозрачными» (англ. “non-transparent” или “opaque”). Это связано с тем, что если на разрешающий вход (если таковой имеется) «прозрачного» триггера подать единицу, то помимо записи в память триггера, входной сигнал будет непосредственно подаваться на выход триггера (то есть можно сказать, что триггер будет работать в качестве повторителя). Если при этом сигнал на входе зависит от сигнала на выходе (то есть выход и вход триггера связаны через внешнюю схему, не содержащую синхронных устройств), то схема начнёт возбуждаться. Иными словами, если в схеме есть своего рода «замкнутый круг», то чтобы

предотвратить возбуждение схемы, нужно «разорвать» этот круг хотя бы в одном месте синхронным устройством (например, синхронным триггером). Об этом можно подробнее прочитать в параграфе 1.6.

На схемах представлены различные реализации триггеров на логических элементах (рис. 3.1, рис. 3.2, рис. 3.3, рис. 3.4). На рисунке 3.4 показан D-триггер, использующий специальную «надстройку» над RS-триггером, делающую его синхронным. Однако существует общая техника построения синхронных триггеров из триггеров с разрешающим входом — двухступенчатые триггеры (англ. master-slave flip-flop). В таком триггере соединены два триггера, на разрешающие входы которых подаются разрешающие входы которых подаются входом противоположные значения.

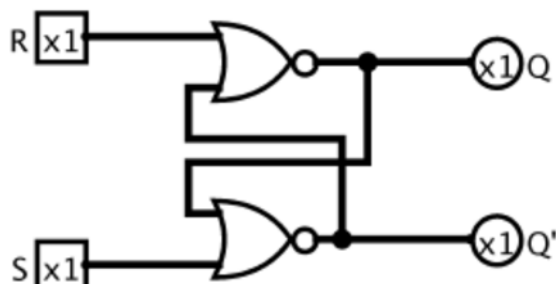


Рис. 3.1. Асинхронный RS-триггер

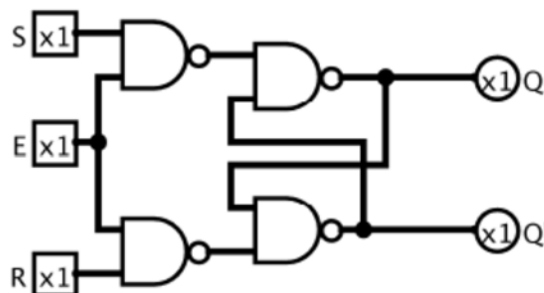


Рис. 3.2. RS-триггер с разрешающим входом

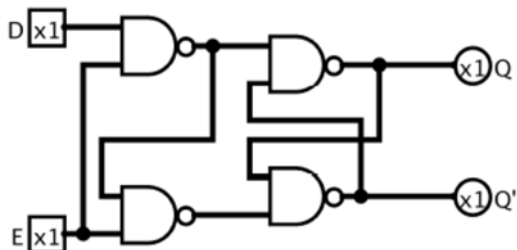


Рис. 3.3. D-триггер с разрешающим входом

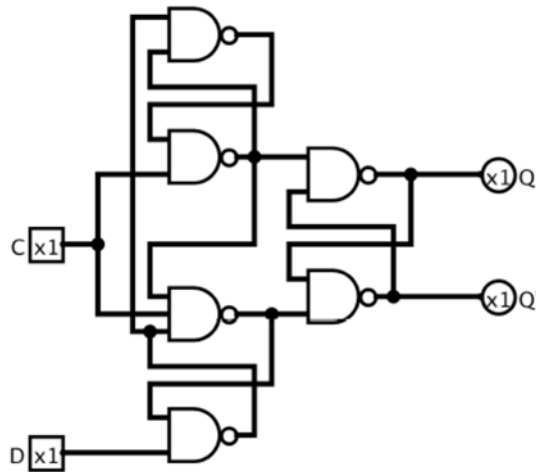


Рис. 3.4. Синхронный D-триггер

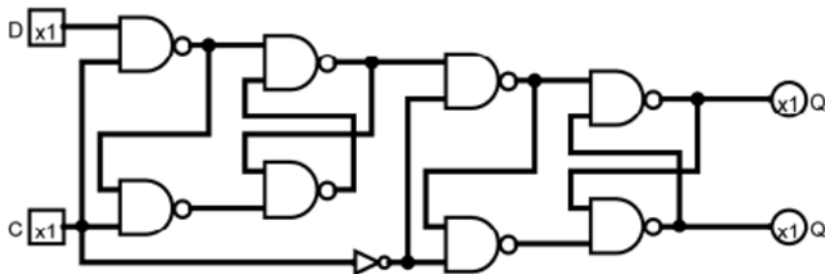


Рис. 3.5. Двухступенчатый синхронный D-триггер

Двухступенчатый синхронный D-триггер показан на рисунке 3.5. При поступлении на его тактовый вход переднего фронта, обновится только состояние на выходе первой ступени, а обновление состояния всего триггера произойдёт при заднем фронте.

Например, RS-триггер, построенный с использованием двух элементов ИЛИ-НЕ, будет возбуждаться без этой случайности, так как оба элемента будут обрабатывать свои входы «нога в ногу». До подачи первого значения на входы, состояние некоторых триггеров будет неопределённым, и Logisim будет показывать значение ошибки на выходах. Это особенности работы Logisim, на практике таких проблем не возникает.

Регистр – последовательностное логическое устройство, используемое для хранения n -разрядных (многобитных) двоичных слов (чисел) и выполнения преобразований над ними. Не считая счётчиков, в рамках данного курса мы будем использовать регистры только для хранения информации (компонент Logisim «Регистр»). Такой регистр имеет многобитный вход для загрузки данных, тактовый вход и многобитный выход, на который всегда поступает значение, сохранённое в регистре. В первом приближении можно представить себе синхронный регистр для

хранения n -разрядных слов как группу из n синхронных D-триггеров с соединёнными тактовыми входами.

Счётчик — последовательностное логическое устройство, на выходы которого поступает двоичный код (многобитное значение), определяемый числом поступивших на его тактовый вход импульсов. Компонент Logisim «Счётчик» по сути является регистром, который меняет хранимое значение на единицу при поступлении очередного тактового импульса. Кроме того, у него имеются два дополнительных входа («загрузка» и «счёт»), которые позволяют выбирать его поведение - увеличивать значение, уменьшать его, или работать как обычный регистр.

В библиотеке Logisim «Память» есть компонент «Генератор случайных чисел». При срабатывании тактового входа он подаёт на выход очередное значение заданной разрядности из псевдослучайной последовательности.

Чтобы сделать в Logisim триггер, регистр или счётчик асинхронным, нужно установить для его атрибута «Срабатывание» значение «Высокий уровень». В таком случае тактовый вход будет работать как разрешающий, и устройство станет «прозрачным».

Более подробную информацию о работе устройств памяти вообще и в Logisim в частности можно найти на страницах справки по библиотеке Logisim (библиотека «Память»).

Задание 1. Реализовать в Logisim асинхронный RS-триггер на логических элементах. Убедиться в том, что его поведение соответствует описанному в таблице и совпадает с поведением RS-триггера из встроенной библиотеки Logisim. Значения на входных контактах можно изменять инструментом «Нажатие».

Задание 2. Повторить задание 1 для RS-триггера с разрешающим входом.

Задание 3. Повторить задание 1 для D-триггера с разрешающим входом.

Задание 4. Повторить задание 1 для синхронного D-триггера.

Задание 5. Спроектировать асинхронный 8-разрядный регистр с разрешающим входом на основе D-триггера из задания 3.

Задание 6. Спроектировать синхронный 8-разрядный регистр на основе D-триггера из задания 4.

Задание 7. Спроектировать 4-разрядный счётчик на основе D-триггера из задания 4 и комбинационного устройства собственной разработки, имеющего 4 входа и 4 выхода. Это комбинационное устройство выдаёт на выходах 4-разрядное значение на единицу большее, чем 4-разрядное значение на его входах, а при подаче максимального значения выдаёт ноль.

Требования к выполнению работы: все задания выполняются в

одном файле проекта Logisim; каждое самостоятельное устройство должно быть оформлено в виде отдельной схемы с осмысленным названием входов и выходов, а также самой схемы.

Дополнительные вопросы и задания:

1. Может ли существовать асинхронный D-триггер без разрешающего входа? Почему?
2. Чем D-триггер с разрешающим входом существенно отличается от синхронного D-триггера? Какую последовательность сигналов нужно подать на их входы, чтобы наглядно продемонстрировать это отличие?
3. Какое минимальное изменение нужно внести в JK-триггер, чтобы получить из него T-триггер?

Лабораторная работа 4. Декодер, мультиплексор и демультиплексор

Декодер — комбинационное логическое устройство, имеющее несколько выходов и многобитный управляющий вход. Если разрядность управляющего входа равна m , то количество выходов равно 2^m . В каждый момент времени «1» будет поступать на один из выходов декодера, номер этого выхода будет равен многобитному значению, поступающему на управляющий вход. Иными словами происходит декодирование двоичного значения на входе в порядковый номер выхода. Декодер часто используется, когда нужно обратиться к какому-либо устройству по его коду (номеру); в дальнейшем мы не раз с этим встретимся. Чтобы построить декодер из логических элементов, нужно составить его таблицу истинности. Для примера рассмотрим построение декодера 2-к-4 (таблица 3.3). При условии, что разряды управляющего входа расставлены по убыванию, а номера выходов — по возрастанию, таблица истинности для декодера любой разрядности представляет

Таблица 3.3. Таблица истинности декодера 2-к-4

c1	c0	o0	o1	o2	o3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

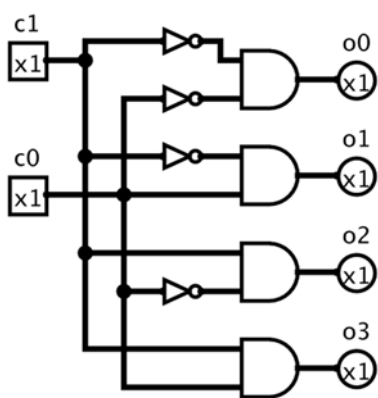


Рис. 3.6. Декодер 2-к-4

собой единичную матрицу — по главной диагонали стоят единицы, а все остальные элементы равны нулю. Схема декодера представлена на рисунке 3.6.

Мультиплексор — комбинационное

логическое устройство, имеющее несколько информационных входов, многобитный управляющий вход и один выход. В каждый момент времени на выход подаётся сигнал с одного из информационных входов. Номер этого входа задаётся двоичным кодом, поступающим на управляющий вход. Если разрядность управляющего входа равна m , то количество информационных входов равно 2^m . Реализация мультиплексора 4-к-1 представлена на рисунке 3.7. В его схему входит декодер 2-к-4.

На рисунке 3.8 показан ещё один вариант реализации мультиплексора — с использованием управляемых буферов.

Подробнее про управляемый буфер можно прочитать в статье «Управляемый буфер/инвертор» справки по библиотеке Logisim (библиотека «Элементы»). Вкратце — когда на управляющий вход управляемого буфера подаётся «1», он ведёт себя как буфер (то есть просто пропускает значение со своего входа на выход), а когда на управляющий вход подаётся «0», на выходе устанавливается высокоимпедансное состояние

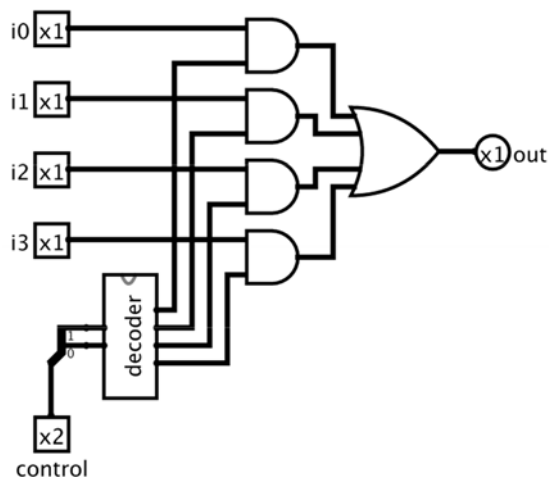


Рис. 3.7. Мультиплексор 4-к-1

(неопределённость). Такой вариант реализации мультиплексора гораздо экономичнее, так как содержит меньшее количество элементов, однако при работе такой схемы должно быть гарантировано, что в каждый момент времени логическая единица будет поступать на управляющий вход строго одного буфера. В противном случае может произойти физическое повреждение устройства.

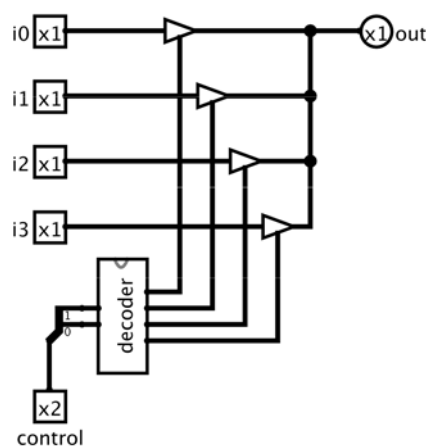


Рис. 3.8. Мультиплексор 4-к-1.

Вариант с управляемыми буферами

Logisim позволяет задавать разрядность значения, проходящего через мультиплексор (то есть разрядность информационных входов и выхода).

Можно представлять себе мультиплексор как аналог железнодорожной стрелки, управляемой управляющим входом. Мультиплексоры обозначают «MUX» (от англ. multiplexer), а также «MS» (от англ. multiplexer selector).

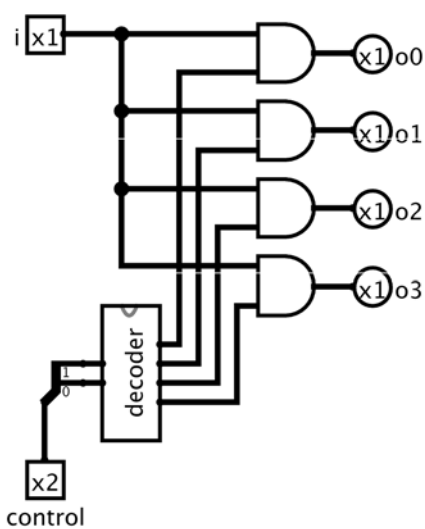
Мультиплексоры используются в цифровых схемах весьма часто. Одно из самых типичных применений — установка мультиплексора перед входом данных регистра. В этом случае на информационные входы мультиплексора подаются значения, которые могут быть записаны в регистр в разных ситуациях, а на управляющий вход — код (номер) значения, которое будет записано в текущем такте. Другое типичное применение — для преобразования нескольких параллельных сигналов в последовательность сигналов, проходящих по одному проводу. При таком использовании на управляющий вход циклически подаются

значения от 0 до $2^m - 1$, а на другом конце провода для обратного преобразования включается декодер или демультиплексор.

Демультиплексор — комбинационное логическое устройство, имеющее несколько выходов, многобитный управляющий вход и один информационный вход. В каждый момент времени на один из выходов подаётся сигнал с информационного входа. Номер выхода задаётся двоичным кодом, поступающим на управляющий вход. Если разрядность управляющего входа равна m , то количество выходов равно 2^m . Реализация демультиплексора 1-к-4 представлена на рисунке 3.9. В его схему также входит декодер 2-к-4.

Logisim позволяет задавать разрядность значения, проходящего через мультиплексор. Демультиплексоры обозначают «DEMUX» или «DMX». Как уже говорилось выше, демультиплексор в паре с мультиплексором используется для передачи нескольких параллельных сигналов по одному проводу посредством их преобразования в последовательные сигналы. Ещё одно типичное применение демультиплексора — для составления сложных устройств памяти (например, регистрового файла) из более простых: на вход демультиплексора подаётся сигнал для записи, его выходы подключаются к входам данных регистров, а на управляющий вход подаётся номер регистра, в который должна производиться запись.

Информационные входы и выходы мультиплексора и демультиплексора в Logisim могут быть многобитными. Это означает, что такое устройство может одновременно перенаправлять группу битов. Допустим, что разрядность управляющего входа равна m , а разрядность информационных входов и выходов равна k . Тогда физически такой «многобитный» мультиплексор или демультиплексор — это группа из k



одноразрядных мультиплексоров или демультиплексоров, и все их управляющие входы соединены в один, который и будет управляющим входом многобитного устройства. В случае мультиплексора каждый i -й одноразрядный мультиплексор будет иметь 2^m входов (как и всё многобитное устройство) и принимать на свой j -й вход i -й бит с j -го входа многоразрядного мультиплексора, а сигнал со своего выхода выдавать на i -й бит выхода многоразрядного мультиплексора. В случае демультиплексора каждый i -й

Рис. 3.9. Демультиплексор 1-к-4

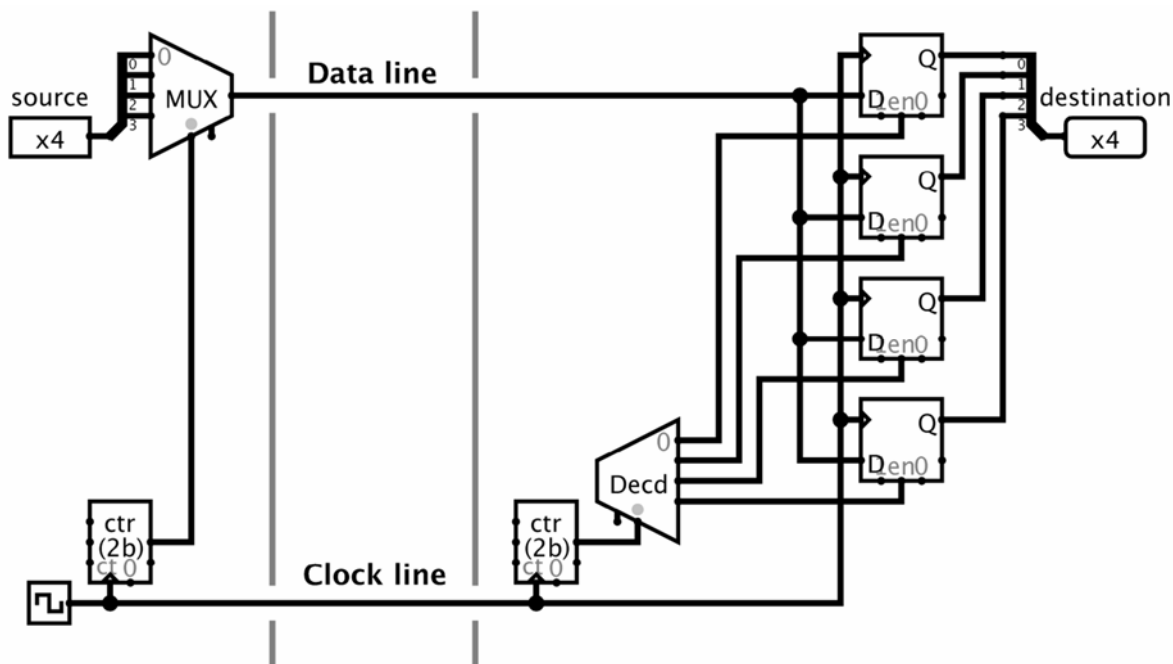


Рис. 3.10. Передача 4-битного значения посредством мультиплексирования

одноразрядный демультиплексор будет иметь 2^m выходов (как и всё многобитное устройство), принимать сигнал с i -го бита многоразрядного демультиплексора, и выдавать сигнал со своего j -го выхода на i -й бит j -го выхода многоразрядного демультиплексора.

Примитивная схема, демонстрирующая принцип передачи многоразрядного (4-разрядного в данном примере) значения по одному проводу приведена на рисунке 3.10. Такой приём называется мультиплексированием. Четырёхразрядное значение для передачи подаётся на вход «source». Счётчики в устройстве-передатчике и устройстве-приёмнике считают количество тактовых импульсов чтобы сформировать 2-битный управляющий сигнал для мультиплексора и декодера. Мультиплексор преобразует многоразрядное значение со входа в последовательные сигналы равной продолжительности, поступающие в каждый момент времени на входы всех D-триггеров. D-триггеры накапливают 4-битное значение, а декодер управляет тем, какой из четырёх триггеров должен сохранить новое значение с выхода демультиплексора. Принятое с линии и демультиплексированное значение снимается с выходов триггеров и поступает на выход

«destination». После прохождения четырёх полных тактов на вход может быть подано новое значение, и ещё через четыре такта оно будет на выходах триггеров, и т.д. Понять принцип работы этой схемы будет гораздо проще после выполнения лабораторной работы 5.

Более подробную информацию о мультиплексоре, демультиплексоре и

декодере в Logisim можно найти на страницах справки по библиотеке Logisim (библиотека «Плексоры»).

Задание 1. Спроектировать одноразрядные мультиплексор и демультиплексор. Для мультиплексора реализовать оба варианта — с управляемыми буферами и без них. Разумеется, предварительно нужно спроектировать декодеры с соответствующей разрядностью управляющего входа. Разрядности управляющего входа мультиплексора и демультиплексора указаны в таблице 3.4. Вариант определяется по последней цифре номера студента в списке группы.

Для декодеров с разрядностью управляющего входа 4 и 5 количество выходов равно соответственно 16 и 32. Модуль «Комбинационный анализ» Logisim не позволяет создавать схемы с количеством выходов большим 12, поэтому декодер придётся делать двухступенчатым. Для «первой ступени» нужно построить с помощью модуля «Комбинационный анализ» декодер 3-к-8 с дополнительным однобитным разрешающим входом. Если на такой вход поступает «0», то и на всех выходах должен быть «0». «Вторая ступень» будет содержать 2 или 4 таких декодера, разветвитель и дополнительное комбинационное устройство. Разветвитель разделяет значение, поступающее на управляющий вход декодера 4-к-16 или 5-к-32 таким образом, что младшие 3 бита посылаются на управляющие входы декодеров 3-к-8, а оставшиеся старшие биты (1 или 2) поступают на входы дополнительного комбинационного устройства, выходы которого управляют разрешающими входами декодеров 3-к-8. Это комбинационное устройство само будет декодером (1-к-2 или 2-к-4).

Задание 2. Используя одноразрядные мультиплексор и демультиплексор из задания 1, спроектировать многоразрядные мультиплексор и демультиплексор. Разрядности данных указаны в таблице 3.4.

Требования к выполнению работы: все задания выполняются в одном файле проекта Logisim; каждое самостоятельное устройство должно быть оформлено в виде отдельной схемы с осмысленным названием всех входов и выходов и самой схемы. Входы мультиплексоров и выходы демультиплексоров должны быть расположены в порядке возрастания их номеров. Для защиты каждого задания нужно продемонстрировать работоспособность каждого отдельного устройства на какой-либо тестовой схеме, в которой нужно сравнить поведение спроектированных устройств с поведением соответствующих устройств из библиотеки Logisim (они должны совпадать).

Дополнительные вопросы и задания:

1. Составьте формулу для вычисления суммарного количества логических элементов в составе декодера, мультиплексора и демультимплексора в зависимости от разрядности управляющего входа и разрядности данных.

2. Оцените общую задержку распространения сигнала для декодера, мультиплексора и демультимплексора.

3. Между двумя городами проложена линия для передачи мультиплексированного сигнала, скорость передачи данных по ней — 1024 Килобит в секунду. Кабель на линии заменили многожильным, и теперь она способна передавать параллельно 32 бита, но максимальная частота передачи данных для нового кабеля в 2 раза ниже. Какой стала скорость передачи данных после замены кабеля?

Таблица 3.4. Варианты заданий для лабораторной работы 2

Вариант	Разрядность управляющего входа		Разрядность данных	
	Мульти-плексора	демульти-плексора	мульти-плексора	демульти-плексора
1	4	5	6	7
2	5	3	8	8
3	3	5	5	6
4	5	3	7	5
5	5	4	8	7
6	4	5	5	6
7	5	4	6	5
8	5	3	8	7
9	3	5	7	8
10	5	3	6	5

Лабораторная работа 5. Арифметика. АЛУ

Группу битов можно рассматривать как отдельные разряды многобитного значения, которое, в свою очередь, можно рассматривать как двоичное представление целого числа. Мы уже не раз сталкивались с такой интерпретацией многоразрядных сигналов. В Logisim группу однобитных проводов можно объединить в пучок; в таком случае можно говорить, что пучок передаёт многобитное значение или целое число. При записи двоичного представления целого числа, нули и единицы записывают начиная со старшего разряда (бита), а при нумерации битов

«нулевым» считают младший (последний в записи) бит.

Однако таким очевидным способом можно представлять только неотрицательные целые числа. Существует несколько способов представления отрицательных целых чисел с помощью многобитных двоичных значений: прямой код, дополнительный код, дополнение до 1, и некоторые другие. Самым удобным, а потому самым популярным способом представления является дополнительный код, который также называют дополнением до 2 (англ. two's complement). Рассмотрим его подробно на примере восьмибитных значений.

При представлении неотрицательных чисел, восемью битами можно представить $2^8 = 256$ различных чисел: от 0 до 255 включительно. При использовании дополнительного кода, теми же восемью битами можно представить числа от -128 до 127, при этом ноль и положительные числа имеют в старшем разряде «0», а отрицательные — «1». Алгоритм нахождения противоположного по знаку значения для положительного числа в дополнительном коде показан с примерами в таблице 3.5.

Таблица 3.5. Алгоритм нахождения отрицания в дополнительном коде

Действие	Пример 1	Пример 2
1. Начиная справа, найти первую «1»	0101001	0101100
2. Инвертировать все биты слева от	1010111 (-	1010100 (-

Крайне важно понимать, что дополнительный код — это не способ «превратить» положительное число в отрицательное, а лишь способ

интерпретации двоичного значения. Одну и ту же последовательность бит определённой длины можно интерпретировать как положительное число, если рассматривать её как беззнаковое представление, или как отрицательное число, если считать её числом, записанным в дополнительном коде. Чтобы чётче уяснить этот момент, рассмотрим обе интерпретации для нескольких восьмибитных чисел (таблица 3.6).

Главное преимущество дополнительного кода — многие арифметические операции над числами, представленными в дополнительном коде, осуществляются цифровыми устройствами так же, как и над беззнаковыми числами. Попробуем сложить двоичные числа 00000001 и 10000001, считая что это беззнаковое представление, то есть складываться будут десятичные числа 1 и 129. Двоичный результат сложения — 10000010, и если интерпретировать это значение как беззнаковое, то оно означает 130, что соответствует действительности. С другой стороны, если бы мы интерпретировали слагаемые как числа в дополнительном коде, то складывали бы 1 и -127. Результат 10000010 в дополнительном коде означает -126, что снова соответствует действительности ($1 + (-127) = -126$).

В Logisim есть встроенная библиотека «Арифметика», содержащая компоненты для выполнения арифметических операций над многобитными двоичными

значениями. Вы можете задавать разрядность этих значений. Почти все компоненты из библиотеки «Арифметика» дают правильные результаты, если интерпретировать и операнды, и результат или как беззнаковые числа, или как числа в дополнительном коде. Исключение составляет компонент «Делитель» — он осуществляет беззнаковое деление.

Компонент «Компаратор», сравнивающий два значения, имеет атрибут «Формат числа», который позволяет задать, как следует интерпретировать входные значения. Компонент

Таблица 3.6. Примеры представления чисел в дополнительном коде

Двоичное значение	Представление в дополнительном коде	Беззнаковое представление
00000000	0	0
00000001	1	1
...
01111110	126	126
01111111	127	127
10000000	-128	128
10000001	-127	129
10000010	-126	130
...
11111110	-2	254
11111111	-1	255

«Отрицатель» позволяет находить противоположное по знаку число (отрицание) в дополнительном коде. Более подробно о каждом компоненте можно прочитать в справке по библиотеке Logisim

«Арифметика»).

Обратите внимание, что при преобразовании значения с меньшей разрядностью (например, восьмибитного) в значение с большей разрядностью (например, шестнадцатибитное), представленных в дополнительном коде, старшие биты нового значения (восемь бит в нашем случае) нужно заполнить нулями или единицами в зависимости от того, является число положительным или отрицательным. Для этой цели можно использовать компонент Logisim «Расширитель битов» (библиотека «Проводка»), со значением «Знак» для атрибута «Тип расширения».

В библиотеке Logisim «Проводка» есть компонент «Датчик», который отображает многобитные значения с возможностью выбирать представление значения: двоичное, восьмеричное, беззнаковое десятичное, знаковое десятичное (дополнительный код), и шестнадцатеричное. Это крайне удобный компонент при отладке схем, связанных с арифметикой.

Все компоненты, выполняющие арифметические операции (сумматор, вычитатель, множитель, делитель, и т.д.) — комбинационные логические устройства, а значит, могут быть реализованы на логических элементах. Рассмотрим принципы их построения.

Начнём с сумматора. Вспомним, как осуществляется сложение «столбиком»:

$$\begin{array}{r} 1234 \\ + 5678 \\ \hline \hline 6912 \end{array}$$

Точки стоят над разрядами, которые принимают единицу от предыдущего (младшего) разряда, так как сумма цифр предыдущего разряда больше 9. Правила сложения одинаковы для любой позиционной системы счисления, поэтому сложение «столбиком» в двоичной системе будет выглядеть точно также:

$$\begin{array}{r} \dots \\ 11011100 \text{ (220)} \\ + 01001110 \text{ (78)} \\ \hline \hline 100101010 \text{ (298)} \end{array}$$

В отличие от десятичной системы, в двоичной системе при сложении

цифр одного разряда могут получиться только числа 0, 1, 10 (2), 11 (3). В соответствующий разряд результата попадает младшая цифра этого числа, а старшая цифра передаётся для сложения со следующим разрядом. Эта старшая цифра называется битом переноса (англ. carry bit). Обратите внимание, что в результате сложения в данном примере получается двоичное значение длиной больше 8 битов. Однако выход сумматора с восьмибитными входами тоже восьмибитный, поэтому дополнительный старший бит не попадёт в результат сложения, и результат, вообще говоря, будет неверным. Но бит переноса из самого старшего разряда восьмибитного значения будет единицей, и этот бит, как правило, подаётся на специальный выход сумматора, и при необходимости может быть учтён при разработке схемы.

Получается, что элементарное устройство для выполнения сложения над одним разрядом должно принимать на вход две цифры из слагаемых (A и B) и бит переноса из предыдущего разряда (Cin — carry in), а на выходе давать цифру для соответствующего разряда результата (S — sum) и бит переноса в следующий разряд (Cout — carry out). Это устройство называется полным одноразрядным сумматором (англ. 1-bit full adder). Составив таблицу истинности для этих двух булевых функций от трёх переменных, получим реализацию полного сумматора (рис. 3.11).

Следующий шаг — построить на основе нескольких одноразрядных сумматоров многоразрядный сумматор, способный складывать многобитные значения. Самое очевидное решение — подавать бит переноса с выхода Cout каждого полного одноразрядного сумматора на вход Cin сумматора следующего разряда.

Получившееся в результате устройство — многоразрядный сумматор с последовательным переносом (англ. ripple carry adder), его блок-схема показана на рисунке 3.12. Такой многоразрядный сумматор прост в реализации, но он очень медленный: каждый одноразрядный сумматор должен дожидаться бита переноса из предыдущего разряда, то есть сумматоры соединены последовательно, а значит, задержки элементов одноразрядных сумматоров суммируются. В результате общая задержка многоразрядного сумматора получается огромной.

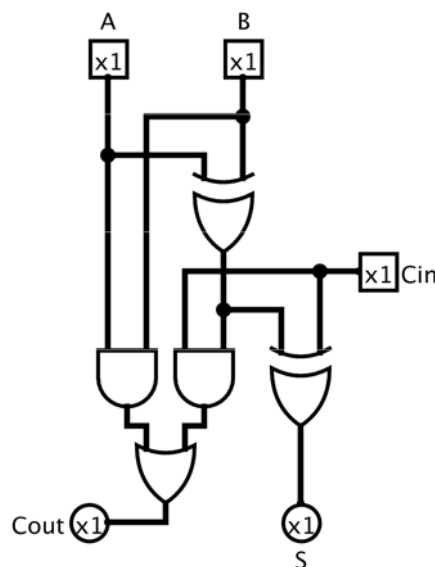


Рис. 3.11. Полный одноразрядный сумматор

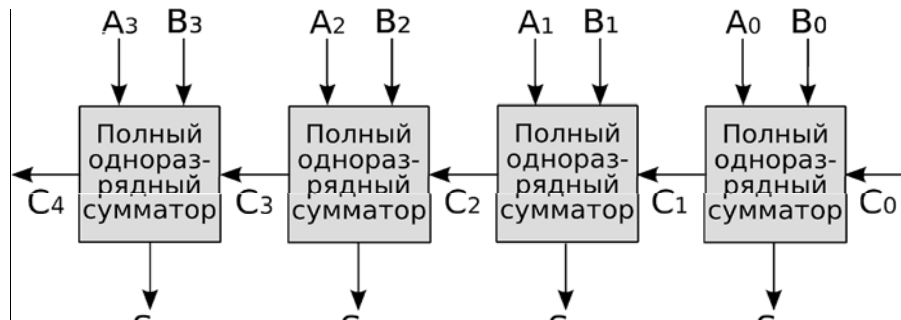


Рис. 3.12. Четырёхразрядный сумматор с последовательным переносом

Существует многоразрядный сумматор с параллельным переносом (англ. carry look-ahead adder), называемый также схемой ускоренного переноса, обладающий значительно меньшей общей задержкой. Полные одноразрядные сумматоры в составе схемы ускоренного переноса имеют небольшое отличие от одноразрядных сумматоров, описанных выше — вместо того, чтобы подавать на выход один бит переноса, они выдают два бита — G (generate) и P (propagate). Бит G — единица, если данный разряд генерирует бит переноса, и может быть вычислен как $G_i = A_i \cdot B_i$. Здесь и далее латинские буквы с индексами означают однобитные значения, индексы — номера разрядов, а знаки умножения и сложения обозначают логическое умножение и сложение (т. е. операции И и ИЛИ). Бит P — единица, если данный разряд распространяет (пропускает в старший разряд) бит переноса, и может быть вычислен как $P_i = A_i \oplus B_i$ (знак \oplus означает исключающее ИЛИ), что с учётом некоторых особенностей схемы ускоренного переноса может быть заменено выражением $P_i = A_i + B_i$.

Значение бита переноса для разряда может быть вычислено из значений битов G , P и бита переноса предыдущего разряда следующим образом: $C_{i+1} = G_i + P_i \cdot C_i$. Зная это, попробуем спроектировать четырёхразрядную схему быстрого переноса. Запишем выражения для битов переноса каждого разряда:

$$C_1 = G_0 + P_0 \cdot C_0$$

$$C_2 = G_1 + P_1 \cdot C_1$$

$$C_3 = G_2 + P_2 \cdot C_2$$

$$C_4 = G_3 + P_3 \cdot C_3$$

Подставляя C_1 в C_2 , затем C_2 в C_3 , а затем C_3 в C_4 , получим расширенные выражения:

$$C_1 = G_0 + P_0 \cdot C_0$$

$$C_2 = G_1 + G_0 \cdot P_1 + C_0 \cdot P_0 \cdot P_1$$

$$C_3 = G_2 + G_1 \cdot P_2 + G_0 \cdot P_1 \cdot P_2 + C_0 \cdot P_0 \cdot P_1 \cdot P_2$$

$$C_4 = G_3 + G_2 \cdot P_3 + G_1 \cdot P_2 \cdot P_3 + G_0 \cdot P_1 \cdot P_2 \cdot P_3 + C_0 \cdot P_0 \cdot P_1 \cdot P_2 \cdot P_3$$

Обратите внимание, что теперь ни один бит переноса не зависит от предыдущего (кроме C_0 , который подаётся в схему извне), а значит, все они вычисляются параллельно, что очень существенно сокращает общую задержку схемы. Общая блок-схема четырёхразрядного сумматора с параллельным переносом приведена на рисунке 3.13.

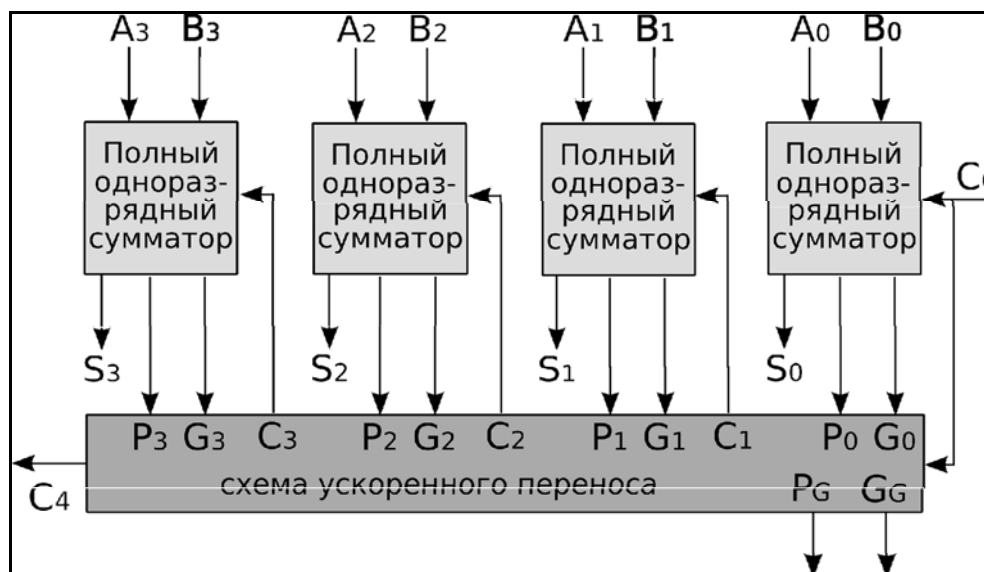


Рис. 3.13. Четырёхразрядный сумматор с параллельным переносом

Используя тот же метод, можно объединить четыре четырёхразрядных сумматора в шестнадцатиразрядный (рис. 3.14). Биты G_G и P_G , которые схемы ускоренного переноса будут посылать в вышестоящую схему, вычисляются по формулам:

$$P_G = P_0 \cdot P_1 \cdot P_2 \cdot P_3$$

$$G_G = G_3 + G_2 \cdot P_3 + G_1 \cdot P_2 \cdot P_3 + G_0 \cdot P_1 \cdot P_2 \cdot P_3$$

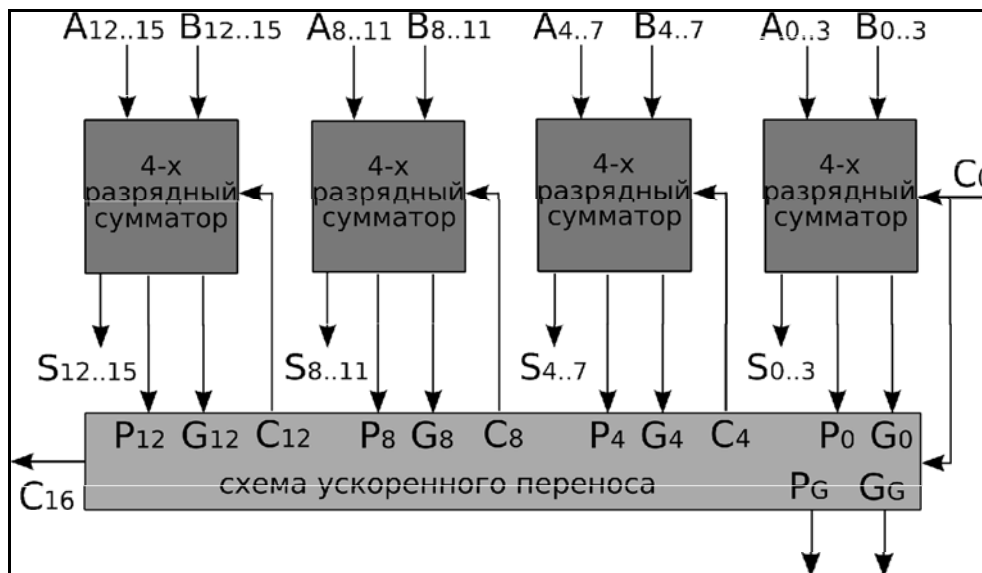


Рис. 3.14. Шестнадцатиразрядный сумматор с параллельным переносом

Несмотря на то, что на этих блок-схемах схемы ускоренного переноса принимают на входах биты с разными индексами, их внутреннее устройство полностью идентично и реализует шесть приведённых выше формул.

С помощью Logisim можно проследивать распространение сигнала по схеме; это полезно сделать, чтобы сравнить общую задержку сумматоров с последовательным и параллельным переносом. Для этого нужно в меню «Моделировать» снять флажок «Моделирование включено», затем выбрать пункт «Сбросить моделирование», и далее последовательно выбирать пункт «Шаг моделирования», наблюдая распространение сигнала.

Пользуясь приведённым здесь математическим аппаратом, можно спроектировать многоразрядный сумматор с параллельным переносом любой разрядности непосредственно (без иерархической структуры), однако сложность устройства с увеличением разрядности будет расти очень быстро.

Существует два подхода к проектированию устройства для вычитания (вычитателя, англ. subtractor). Первый состоит в том, чтобы спроектировать аналог полного одноразрядного сумматора для вычитания и использовать его в многоразрядных вычитателях. Такое устройство (полный одноразрядный вычитатель) будет передавать старшим разрядам не бит переноса, а бит займа (англ. borrow bit). Схема полного одноразрядного вычитателя показана на рисунке 3.15. Его входы: X — уменьшаемое, Y — вычитаемое, Bin (borrow in) — вход займа; выходы: D (difference) — разность, $Bout$ (borrow out) — выход займа. Трёхвходовый элемент исключающее ИЛИ выдаёт единицу, когда

на нечётном количестве входов единица. Построение многоразрядного вычитателя с последовательным займом ничем не отличается от построения аналогичного сумматора. Такой вычитатель будет работать и с беззнаковой интерпретацией чисел и с дополнительным кодом.

Второй подход — воспользоваться тем фактом, что вычитание — это сложение с числом, противоположным по знаку вычитаемому ($X - Y = X + (-Y)$). В вычитателе, построенном по такому принципу, сначала для вычитаемого будет значение (в дополнительном коде; этот алгоритм описан выше), а затем будет находиться противоположное по знаку производиться сложение с помощью любого многоразрядного сумматора.

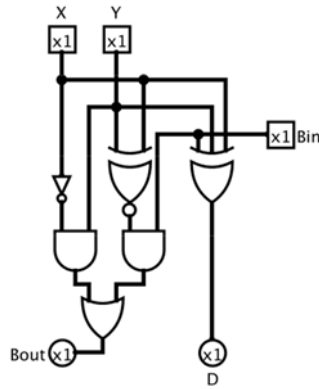


Рис. 3.15. Полный одноразрядный вычитатель

При построении устройства для умножения чисел (множителя) в первую очередь нужно помнить, что разрядность результата в два раза больше, чем разрядность сомножителей. Обычная практика — выдавать старшую и младшую половины результата на разные многобитные выходы, так что каждый из них имеет разрядность исходных данных (сомножителей). Построение множителя — задача весьма сложная; мы рассмотрим только самый простой вариант реализации, он использовался в самых первых компьютерах. Для этого вспомним умножение «столбиком»:

$$\begin{array}{r}
 123 \\
 \times 456 \\
 \hline
 738 \quad (123 \cdot 6) \\
 615 \quad (123 \cdot 5, \text{сдвинутое на одну позицию влево}) \\
 + 492 \quad (123 \cdot 4, \text{сдвинутое на две позиции влево}) \\
 \hline
 56088
 \end{array}$$

Умножение в двоичной системе будет производиться тем же способом:

$$\begin{array}{r}
 1110 \quad (14 \text{ в двоичном представлении}) \\
 \times 1011 \quad (11 \text{ в двоичном представлении}) \\
 \hline
 1110 \quad (1110 \cdot 1) \\
 1110 \quad (1110 \cdot 1, \text{сдвинутое на одну позицию влево}) \\
 0000 \quad (1110 \cdot 0, \text{сдвинутое на две позиции влево}) \\
 + 1110 \quad (1110 \cdot 1, \text{сдвинутое на три позиции влево}) \\
 \hline
 10011010 \quad (154 \text{ в двоичном представлении})
 \end{array}$$

Иными словами нужно только умножать (а умножение сводится либо к копированию первого множителя, либо к заполнению всех разрядов нулями),

сдвигать полученные произведения и последовательно складывать их. Это можно делать потактово (тогда множитель перестанет быть комбинационным устройством), или реализовав все операции в виде одного комбинационного устройства (оно будет иметь очень большую общую задержку, так как сумматоры будут включены последовательно). Поскольку левые сдвиги осуществляются на заранее известные количества позиций, то их реализация даже не будет требовать логических элементов — лишь копирование значений со входов множителя на входы сумматоров со смещением. Поскольку у нас уже есть готовые сумматоры, реализация множителя по описанному алгоритму не составит труда. Обратите внимание, что разрядность сумматоров, как и разрядность результата, в два раза больше разрядности исходных сомножителей.

Множитель, реализующий такой принцип, будет правильно работать только для беззнаковых чисел; если интерпретировать сомножители и результат как числа в дополнительном коде, результат будет неверным при хотя бы одном отрицательном сомножителе.

Реализация устройства для целочисленного деления (с остатком) является весьма сложной задачей; мы не будем рассматривать её в этом курсе. Напомним только, что как и в случае с умножением, результатом деления двух чисел разрядности n будут два числа разрядности n — частное и остаток; как правило, они выводятся на разные выходы.

Устройства для арифметических операций, рассмотренные выше, довольно редко используются в схемах непосредственно. Как правило, они включаются в состав арифметико-логического устройства (АЛУ). АЛУ — блок процессора, который служит для выполнения арифметических и логических преобразований над данными, представляемыми в виде многоразрядных значений (машинных слов), называемых операндами. В большинстве случаев АЛУ — комбинационное устройство. Примитивное АЛУ имеет два входа для операндов,

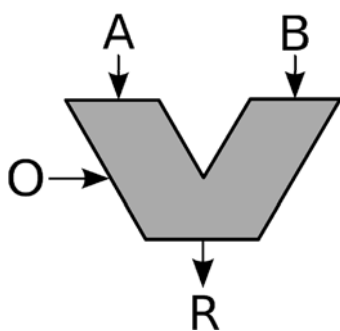


Рис. 3.16. Условное графическое обозначение АЛУ

один или два (если реализованы умножение и/или деление с остатком) выходов для результата, а также управляющий вход, на который подаётся код операции, которую необходимо выполнить над операндами. Опционально АЛУ может принимать и выдавать бит переноса, выдавать сигнал переполнения и деления на ноль, и т.д. Типичное условно-графическое обозначение АЛУ приведено на рисунке 3.16. Здесь A и B — операнды, R — результат, O — код операции.

На рисунке 3.17 изображена схема примитивнейшего восьмибитного АЛУ. Оно может выполнять четыре операции: побитовое ИЛИ, побитовое И, сложение и логический сдвиг.

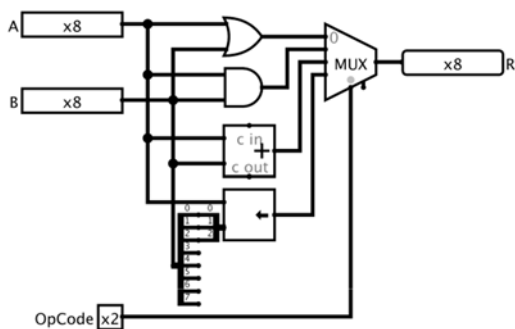


Рис. 3.17. Примитивное восьмибитное АЛУ

Двухбитный код операции поступает на управляющий вход мультиплексора, и на выход поступает результат выбранной операции. Левый сдвиг требует в качестве второго операнда трёхбитное значение, поэтому из операнда В предварительно выделены три младших бита с помощью компонентов «Разветвитель».

В этой лабораторной работе мы рассматривали арифметические операции только над целыми числами. Естественно, существуют цифровые устройства для выполнения операций над действительными числами (числами с плавающей точкой). Способы представления таких чисел в двоичном коде и организация устройств для работы с ними — отдельная и весьма сложная тема, она выходит за рамки этого курса. Вы можете рассмотреть её самостоятельно и даже попробовать реализовать такие устройства в Logisim.

Задание 1. Спроектировать в Logisim полный одноразрядный сумматор.

Задание 2. Спроектировать 8-разрядный сумматор с последовательным переносом, используя одноразрядный сумматор из задания 1.

Задание 3. Спроектировать 4-разрядный сумматор с параллельным переносом, используя видоизменённый полный одноразрядный сумматор.

Задание 4. Спроектировать 8-разрядный сумматор с параллельным переносом, используя два 4-разрядных сумматора из задания 3.

Задание 5. Спроектировать 16-разрядный сумматор с параллельным переносом, используя четыре 4-разрядных сумматора из задания 3.

Задание 6. Спроектировать полный одноразрядный вычитатель.

Задание 7. Спроектировать 8-разрядный вычитатель с последовательным займом, используя одноразрядный вычитатель из задания 6.

Задание 8. Спроектировать 8-разрядное устройство для нахождения числа, противоположного по знаку данному (отрицатель) в дополнительном коде. Сделать это можно, составив таблицу истинности вручную, однако такой способ весьма трудоёмок. Гораздо проще получить таблицу истинности, проанализировав с помощью модуля «Комбинационный анализ» схему, содержащую компонент «Отрицатель» из встроенной библиотеки Logisim.

Задание 9. Спроектировать 8-разрядный вычитатель, используя сумматор из задания 4 и отрицатель из задания 8.

Задание 10. Спроектировать множитель для беззнаковых чисел с 8-разрядными входами и двумя 8-разрядными выходами (для старшей и

младшей половин результата) в виде единого комбинационного устройства. Сумматоры в составе множителя — 16-разрядные с параллельным переносом из задания 5.

Задание 11. Спроектировать АЛУ для выполнения операций над двумя 8-разрядными операндами. АЛУ должно поддерживать выполнение восьми операций: сложение, вычитание, нахождение отрицания для первого операнда, умножение, поразрядное И, поразрядное ИЛИ, поразрядное исключающее ИЛИ, поразрядное НЕ над первым операндом. Все арифметические операции кроме умножения должны правильно работать как для беззнаковых чисел, так и для чисел в дополнительном коде. Вход кода операции — 3-разрядный. Соответствие кодов и операций — на ваше усмотрение.

Младшая половина результата умножения должна подаваться на общий выход для результата, а старшая половина — на специальный выход. Для выполнения поразрядных логических операций нужно предварительно спроектировать устройства, состоящие только из логических элементов с одноразрядными входами. Для выполнения арифметических операций должны использоваться только устройства, спроектированные в предыдущих заданиях лабораторной работы; какие из реализаций выбирать — на ваше усмотрение.

Мультиплексор в составе АЛУ — с восемью 8-разрядными информационными входами, реализованный на логических элементах с одноразрядными входами в виде отдельной подсхемы. Его реализация рассматривалась в лабораторной работе 3.

Убедитесь, что кроме контактов, проводов, разветвителей, констант и датчиков, все подсхемы АЛУ на всех уровнях иерархии содержат только логические элементы с одноразрядными входами. Для этого воспользуйтесь возможностью «Получить статистику схемы» из меню «Проект» Logisim. Какое общее количество логических элементов использовано в вашей реализации АЛУ?

Требования к выполнению работы: все задания выполняются в одном файле проекта Logisim; каждое самостоятельное устройство (начиная от одноразрядного сумматора и кончая АЛУ) должно быть оформлено в виде отдельной схемы с осмысленным названием всех входов и выходов и самой схемы. Для защиты каждого задания нужно продемонстрировать работоспособность каждого отдельного устройства на нескольких примерах входных значений, расположив устройство на отдельной тестовой схеме, общей для всей работы. Эти тестовые входные значения должны задаваться компонентами «Константа», к каждому из которых (а также к выходным контактам) должен быть

подключен компонент «Датчик», настроенный на отображение знакового десятичного представления значений.

Дополнительные вопросы и задания:

1. Возможно ли уменьшить количество логических элементов в составе вашего АЛУ без потери функциональности? Попробуйте это сделать. Для этого можно попробовать выбрать другую реализацию отдельного устройства, совместить сумматор и вычитатель, минимизировать некоторые булевы функции.
2. Определите, у кого в группе наименьшее количество логических элементов в составе АЛУ. Почему?
3. Попробуйте оценить максимальную задержку вашего АЛУ при выполнении операции. Это можно сделать, вручную посчитав длину самой большой цепочки последовательно соединённых логических элементов. Попробуйте проследить распространение сигнала по подсхемам, воспользовавшись возможностью Logisim, доступной через пункт «Шаг моделирования» из меню «Моделировать».
4. Определите, у кого в группе минимальная общая задержка при выполнении операции АЛУ. Какие из реализаций арифметических устройств им были выбраны?
5. Являются ли «победители соревнований» из пунктов 2 и 4 одним и тем же человеком? Почему? Попробуйте ответить на этот вопрос, даже если «соревнования» не проводились.
6. Попробуйте найти в литературе и реализовать устройство для умножения с меньшей общей задержкой.
7. Попробуйте предложить реализацию устройства для целочисленного деления (это сложное задание).

Спроектированное в процессе выполнения работы АЛУ будет содержать несколько сотен логических элементов, возможно даже несколько тысяч; кроме того, оно будет иметь довольно существенную максимальную общую задержку, так как умножитель содержит весьма большое количество элементов, соединённых последовательно. Причина этих недостатков в том, что мы не задумывались о дополнительной минимизации и оптимизации схем. Можно было провести минимизацию после проектирования, кроме того, можно было задуматься о том, как существенно упростить АЛУ без потери функциональности, проведя тщательный анализ всех схем в отдельности, их соединения, и всего устройство в целом ещё на начальной стадии проектирования, то есть до воплощения в виде логических элементов.

В конце 1960-х годов в составе уже упоминавшейся серии микросхем 7400 в виде отдельной микросхемы было выпущено 4-битное АЛУ 74181 (отечественный аналог — 155ИПЗ). Оно позволяет выполнять все семь логических операций, перечисленных в таблице 1.1, сложение и вычитание операндов, а также некоторые другие комбинации логических и арифметических операций. Умножение и деление в этом АЛУ не реализованы из-за их большой сложности. Объединив несколько таких АЛУ, выполняющих операции над 4-битными операндами, можно получить АЛУ большей разрядности. Биты переноса и займа можно передавать от одного 4-битного АЛУ к другому как непосредственно (получая сумматор и вычитатель с последовательным переносом), так и через дополнительную схему ускоренного переноса (74181 также генерирует биты P и G). Схема АЛУ 74181 приведена на рисунке 3.18.

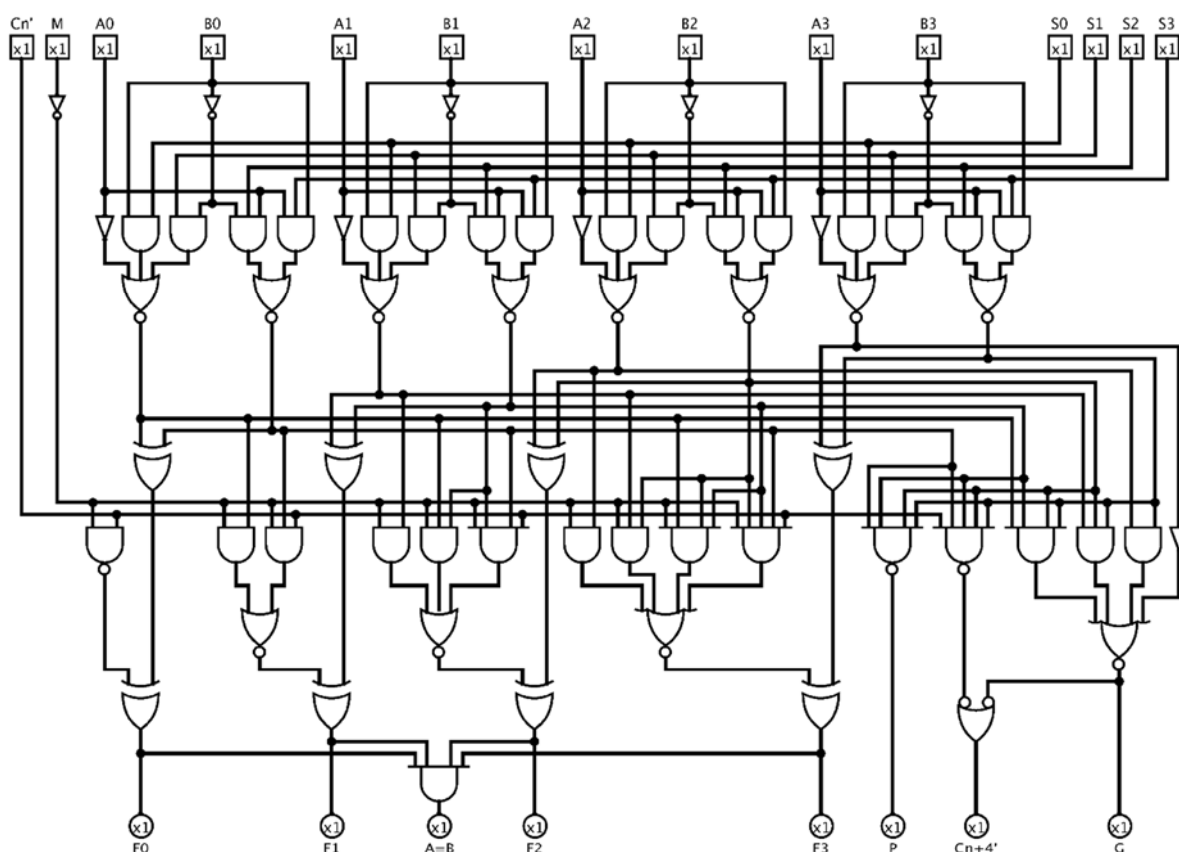


Рис. 3.18. АЛУ 74181

Схема этого АЛУ — великолепный пример того, насколько существенно можно уменьшить сложность внутреннего устройства при тщательнейшем анализе и квалифицированном проектировании: оно содержит всего около семидесяти логических элементов, а объединив два таких устройства, можно получить АЛУ эквивалентное по

функциональности спроектированному нами (и даже немного превосходящее его, за исключением операции умножения).

На этой схеме входы $A_0...A_3$ и $B_0...B_3$ — операнды, $S_0...S_3$ — код операции, C_n' — бит переноса (или займа), поступающий из предыдущего разряда, M — признак того, что код операции задаёт логические операции (когда на этом входе 0, выполняются арифметические). Выходы $F_0...F_3$ — результат выполнения операции, $A=B$ — признак равенства операндов при выполнении вычитания, C_{n+4}'

— выход бита переноса (или займа) в следующий разряд, P и G — биты

propagate и generate для схемы ускоренного переноса. Вход C_n' и выход C_{n+4}' — инвертированные, то есть наличие переноса (или займа) обозначается логическим нулём.

Соответствие кодов и операций (логических и арифметических) приведено в таблице 3.7.

Словами «plus» и «minus» обозначены соответствующие арифметические операции чтобы избежать путаницы с логическим сложением (то есть с операцией ИЛИ). Символ "<<" означает логический левый сдвиг.

При $M = 0$ (то есть при выполнении арифметических операций) для всех операций plus и minus учитывается и генерируется бит переноса (или займа), а также биты P и G ; при $M = 1$ выполняются исключительно поразрядные операции, то есть биты переноса не учитываются и не генерируются.

Таблица 3.7. Операции АЛУ 74181

S_3	S_2	S_1	S_0	Логические ($M = 1$)	Арифметические ($M = 0, C_n' = 1$)
0	0	0	0	\underline{A}	A
0	0	0	1	$A + B$	$A + \underline{B}$
0	0	1	0	$A \oplus B$	$A + B$
0	0	1	1	$\underline{0}$	minus 1 —
0	1	0	0	$A \oplus B$	A plus $A \oplus B$ —
0	1	0	1	B	(A + B) plus $A \oplus B$
0	1	1	0	$A \square \underline{B}$	A minus B minus 1
0	1	1	1	$A \oplus B$	$A \oplus B$ minus 1
1	0	0	0	$\underline{A + B}$	A plus $A \oplus B$
1	0	0	1	$A \square B$	A plus \underline{B}
1	0	1	0	B	(A + B) plus $A \oplus B$
1	0	1	1	$A \oplus B$	$A \oplus B$ minus 1
1	1	0	0	1 —	A plus (A << 1)
1	1	0	1	$A + B$	(A + B) plus A
1	1	1	0	$A + B$	(A + B) plus A
1	1	1	1	A	A minus 1

Лабораторная работа 5. Тактовые импульсы. Шина

В предыдущих лабораторных работах мы не пользовались сложными устройствами библиотеки компонентов Logisim, мы фактически проектировали их сами из логических элементов. Это было оправданно, так как мы изучали назначение различных цифровых компонентов, принципы их работы и внутреннее устройство. По сути, все предыдущие работы — лишь обязательная подготовка к собственно проектированию цифровых устройств, имеющих практическое применение. Начиная с этой лабораторной работы, мы непосредственно приступаем к такому проектированию, поэтому теперь, для экономии сил и времени, любой компонент, нужный для реализации задачи, мы будем брать из библиотеки компонентов Logisim — ведь теперь мы знаем, как он работает, и как при необходимости реализовать его на логических элементах.

Вопросы, связанные с понятием тактовых импульсов весьма подробно рассмотрены в параграфе 1.6 данного пособия; его прочтение необходимо для выполнения этой лабораторной работы.

Чтобы понять, что такое шина, рассмотрим небольшой практический пример. Допустим, у нас есть четыре устройства, которые могут выдавать восьмибитные значения («источники данных»), а также четыре устройства, которые могут принимать восьмибитные значения («приёмники данных»). Устройства могут быть любого рода — комбинационные, последовательностные, регистры, ОЗУ, устройства ввода и вывода, и т.д. Наша задача — иметь возможность по выбору передавать восьмибитные значения от любого устройства-источника к любому устройству-приёмнику. Совокупность устройств, необходимых для реализации такой возможности называется шиной (англ. bus, bidirectional universal switch — двунаправленный универсальный коммутатор). Кроме того, часто шиной называют непосредственно многобитный пучок проводов, несущий сигналы от источников к приёмникам. Судя по переводу английского названия шины, передача данных должна быть двунаправленной, то есть одно и то же устройство может быть как источником, так и приёмником данных. На первый взгляд, наш пример подразумевает только однонаправленную передачу, но это не так: устройство в нашем примере может выступать в качестве источника и приёмника одновременно, если мы будем считать источником данных его выходы, а приёмником данных — его же входы. В цифровой технике одни и те же контакты могут быть входами и выходами одновременно (мы это увидим на примере порта данных ОЗУ в следующей лабораторной работе), однако, разбирая работу шины, мы будем жёстко разграничивать входные и выходные контакты — с одной стороны, это позволит сделать объяснение более ясным, а с другой

стороны мы ограничены тем, что создаваемые пользователем контакты в Logisim не могут быть входами и выходами одновременно.

Рассмотрим способ реализации примитивной шины с использованием мультиплексора (рис. 3.19). Как правило, этот мультиплексор строится с использованием управляемых буферов — такая реализация оказывается гораздо более компактной и экономичной, так как мультиплексор, управляющий многобитными значениями, представляет собой группу из «однобитных» мультиплексоров, количество которых равно разрядности этого значения, и использование схемы с управляемыми буферами очень значительно снижает суммарное количество элементов.

Двухбитный вход «sourceCode» на схеме задаёт номер (код) устройства-источника, значение с выхода которого подаётся в данный момент на шину. Если максимальное количество устройств-источников равно n , а разрядность входа «sourceCode» равна m , то должно выполняться соотношение $2^m \geq n$. Очевидно, что в каждый момент времени происходит передача значения только от одного устройства.

На представленной схеме в любой момент времени все устройства-приёмники получают данные с шины. Если устройства-приёмники имеют тактовые входы (т. е. они синхронные), а это типично при использовании шины, то можно управлять тем, какие из устройств будут получать синхронизирующие импульсы. На схемах ниже представлены две реализации такой возможности: для случая, когда в каждый такт только один из приёмников может получать тактовый импульс (рис. 3.20), и для случая, когда тактовый импульс может быть передан любой комбинации устройств-приёмников (рис. 3.21).

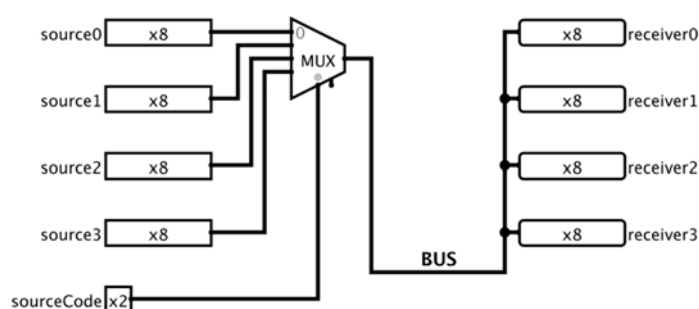


Рис. 3.19. Шина с

использованием мультиплексора. В первом варианте вход «receiverCode» задаёт номер (код) устройства, которому посылается тактовый импульс; его разрядность определяется по тому же соотношению, что и для входа «sourceCode». Во втором варианте вход «receiversFlags» задаёт комбинацию устройств для получения тактового импульса: единица в каждом отдельном бите этого входа означает, что устройство с соответствующим номером получит тактовый импульс.

Задание 1. Спроектировать схему для решения следующей задачи. Необходимо иметь возможность передавать последовательность 8-разрядных значений по одному проводу (плюс провод для передачи синхронизирующих импульсов), то есть нужно производить мультиплексирование сигнала.

Схема процесса мультиплексирования 4-разрядных значений показана в лабораторной работе 3 (рис. 3.10). Для решения задачи нужно разделить эту схему на две подсхемы — передатчик мультиплексированного сигнала и

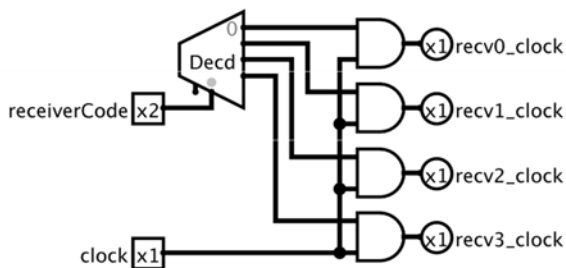


Рис. 3.20. Устройство управления тактовыми импульсами, вариант 1

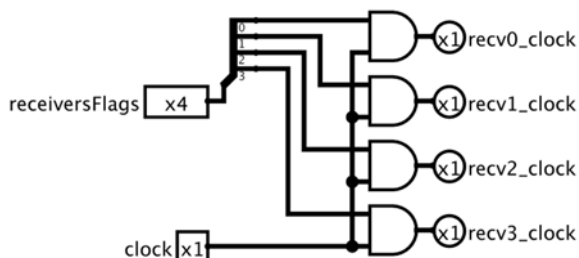


Рис. 3.21. Устройство управления тактовыми импульсами, вариант 2

его приёмник, и изменить разрядность входного и выходного значений на 8. Кроме того, и приёмник и передатчик должны передавать во внешнюю схему импульс, сигнализирующий о том, что процесс передачи очередного 8-разрядного значения завершён, на вход передатчика можно подавать новое значение, а с выхода приёмника — считывать новое принятое значение. Этот импульс можно снимать с выхода «перенос» счётчиков в составе передатчика и приёмника. Новые значения пусть генерируются компонентом «Генератор случайных чисел» из библиотеки «Память», а с выхода приёмника — записываются в 8-битный регистр. На внешней схеме кроме передатчика и приёмника должны присутствовать генератор случайных чисел, тактовый генератор, регистр, в который записываются принятые значения, и главное — провод, по которому передаётся мультиплексированный сигнал (и провод, несущий синхронизирующие импульсы).

Задание 2. Спроектировать устройство, поведение которого описано в таблице 3.8. Все устройства должны выполнять свою задачу потактово. Вариант определяется по последней цифре номера студента в списке группы. Все регистры имеют разрядность 32 бита.

Может быть полезным рассмотреть общий принцип решения каждой из этих задач всей группой, а выполнять реализацию — индивидуально.

Таблица 3.8. Варианты задания 2 для лабораторной работы 5

Вариант	Поведение устройства
1	Вычисляет факториал числа, подаваемого на вход устройства, потактово умножая содержимое одного регистра на содержимое второго, и увеличивая значение второго на 1.
2	Выполняет деление с остатком значения одного регистра на значение второго, потактово вычитая делитель из делимого и сравнивая разность с нулём.
3	Вычисляет сумму первых n членов арифметической прогрессии с разностью d , потактово увеличивая значение одного регистра на значение второго, и увеличивая значение второго на d . n и d задаются входами устройства.

4	Вычисляет сумму первых n членов геометрической прогрессии со знаменателем q , потактово увеличивая значение одного регистра на значение второго, и увеличивая значение второго в q раз. n и q задаются входами устройства.
5	Имеется 16 констант с заранее введёнными 32-битными значениями. Устройство находит номер константы с наибольшим значением, реализуя классический алгоритм.
6	Играет само с собой в «камень, ножницы, бумага»: каждый такт случайным образом генерируются два значения в диапазоне $[0; 2]$, сравниваются по правилам игры, и на выход выдаётся код «победителя» (или код ничьи).
7	Находит количество единиц в двоичном представлении значения регистра, потактово перебирая биты значения.
8	Проверяет качество работы компонента «Генератор случайных чисел»: на протяжении большого числа тактов считает количество выданных им 32-битных значений больших и меньших среднего, и выводит разность между этими количествами.
9	Циклически заполняет пять регистров случайными значениями. Останавливается, когда все пять значений оказываются расположенными в порядке возрастания.
10	Девять D-триггеров расположены квадратом 3 на 3. По нечётным тактам устройство заполняет их случайными значениями, а по чётным — «поворачивает» значения восьми регистров вокруг центрального на 45 градусов по часовой стрелке.

Задание 3. Спроектировать в виде отдельной схемы реализацию шины; в неё также должен быть включен первый вариант устройства управления тактовыми импульсами. Разрядность данных шины и количество устройств-источников и устройств-приёмников приведены в таблице 3.9. Для демонстрации работоспособности шины нужно на внешней схеме подключить к ней регистры с заранее занесёнными в них произвольными значениями, и каждый такт подавать на входы `sourceCode` и `receiverCode` случайные номера устройств. Попробуйте использовать некоторые регистры и как устройства-источники, и как

устройства-приёмники одновременно. Проследите за перемещением значений по регистрам. Таблица 3.9. Варианты задания 3 для лабораторной работы 5

Вариант	Разрядность данных	Количество устройств-источников	Количество устройств-приёмников
1	16	4	16
2	8	8	8
3	32	16	4
4	24	8	8
5	8	16	4
6	24	4	16
7	16	16	8
8	32	8	4
9	16	4	8
10	8	16	8

Требования к выполнению работы: все задания выполняются в одном файле проекта Logisim; каждое самостоятельное устройство должно быть оформлено в виде отдельной схемы с осмысленным названием всех входов и выходов и самой схемы. Для защиты каждого задания нужно продемонстрировать работоспособность каждого отдельного устройства на тестовой схеме.

Лабораторная работа 7.

Сложные устройства памяти: ОЗУ и ПЗУ

Для хранения больших объёмов данных в сложных цифровых устройствах используют ОЗУ (оперативное запоминающее устройство, англ. RAM, random access memory) и ПЗУ (постоянное запоминающее устройство, англ. ROM, read-only memory). Главное отличие между ними — ОЗУ позволяет динамически (оперативно) изменять своё состояние, а при отключении питающего напряжения сохранённая в нём информация теряется, а ПЗУ сохраняет информацию и после отключения питания (т.н. «энергонезависимая память»), но эта информация статична (записывается один раз и не может быть оперативно перезаписана).

Физически ячейка памяти для хранения одного бита в составе ОЗУ может быть триггером на логических элементах (память статического типа, англ. SRAM), но чаще — элементарным устройством из одного транзистора и одного конденсатора (память динамического типа, англ. DRAM). Память динамического типа компактнее, дешевле, но медленнее из-за того, что время заряда или разряда конденсатора больше, чем задержки логических элементов в составе триггера.

Существует большое количество разных физических реализаций ПЗУ: масочные (ROM), программируемые (PROM), перепрограммируемые (EPROM), электрически стираемые перепрограммируемые (EEPROM), и некоторые другие. Флеш-память, в частности, является разновидностью EEPROM.

Логическая структура обоих видов ЗУ представляет собой линейную последовательность отдельных ячеек памяти фиксированной разрядности. Как правило, разрядность ячейки (этот параметр ЗУ называется «разрядность данных») является степенью двойки (8 битов, 16 битов, 32 бита, и т.д.), однако могут быть исключения. Длина последовательности ячеек определяется разрядностью адреса запоминающего устройства: если разрядность адреса равна n , то количество ячеек памяти в устройстве равно 2^n .

Простейшее ОЗУ имеет многоразрядный вход адреса (A), значение на котором определяет номер ячейки, к которой мы обращаемся в данный момент; многоразрядный вход данных (D), на который подаётся значение для записи в выбранную ячейку; многоразрядный выход данных (D), на который выдаётся значение, сохранённое в текущей ячейке; и в случае

синхронного ОЗУ — тактовый вход (в Logisim обозначается треугольником), при срабатывании которого в выбранную ячейку записывается новое значение. Простейшее ПЗУ имеет только вход адреса и выход данных — поскольку запись в ПЗУ невозможна, вход данных и тактовый вход не нужны; значение на выходе данных обновляется сразу после поступления нового значения на вход адреса.

В Logisim также предусмотрен особый интерфейс данных, в случае использования которого ОЗУ имеет единый порт для записи и чтения данных (то есть он является одновременно входом и выходом). Для работы с таким интерфейсом данных нужно использовать компонент управляемый буфер. Подробнее про этот интерфейс можно прочитать в статье «ОЗУ» справки по библиотеке Logisim (библиотека «Память»), а про управляемый буфер — в статье «Управляемый буфер/инвертор» (библиотека «Элементы»). В случае использования единого порта для чтения и записи, в Logisim можно выбрать между синхронным и асинхронным ОЗУ. Значительная часть промышленно выпускаемых модулей памяти — асинхронные, с единым портом.

Редактировать содержащиеся в ОЗУ или ПЗУ значения в Logisim можно щёлкнув на соответствующем компоненте правой кнопкой мыши и выбрав пункт контекстного меню «Редактировать содержимое...». После этого откроется встроенный в Logisim шестнадцатеричный редактор. Содержимое ПЗУ сохраняется в файле проекта Logisim (*.circ), и является атрибутом компонента ПЗУ (иными словами, однажды отредактировав содержимое ПЗУ, можно не беспокоиться за его сохранность). С ОЗУ ситуация иная — информация в ОЗУ динамически меняется, а при сбросе моделирования или перезагрузке проекта всё содержимое ОЗУ заполняется нулями. Однако в Logisim есть возможность сохранять и загружать содержимое ОЗУ и ПЗУ, используя отдельные файлы — файлы образов памяти. Это делается через пункты контекстного меню «Сохранить образ...» и «Загрузить образ...». Файлы образов Logisim — это обычные текстовые файлы, в которых данные записаны в шестнадцатеричном виде. Более подробно про загрузку и сохранение файлов образов и их формат можно прочитать в статье

«Всплывающие меню и файлы» раздела «Компоненты памяти» руководства пользователя Logisim.

Кроме прочих, существует немного необычное использование ПЗУ. К нему можно прийти, если ответить на вопрос «является ли ПЗУ последовательным логическим устройством?» С одной стороны, ПЗУ — это устройство памяти, а значит оно, скорее всего, последовательное. Однако из определения последовательного

устройства ясно, что его внутреннее состояние изменяется со временем. Для ПЗУ это не так — сохранённая в нём информация жёстко «защита» и не может быть изменена, а значит, значения на выходах зависят только от значений на входах, то есть ПЗУ — комбинационное устройство. Отсюда вытекает вывод, что оно реализует набор булевых функций от значений на своих входах. Входами являются отдельные биты многоразрядного входа адреса, а выходами — отдельные биты многоразрядного выхода данных. Количество однобитных входов этого комбинационного устройства равно разрядности адреса ПЗУ, а количество выходов (т. е. булевых функций) равно разрядности данных ПЗУ. Зная это, можно использовать ПЗУ со введёнными в него данными вместо сложной схемы на логических элементах. На практике это почти всегда плохое решение — ПЗУ дороже, медленнее и потребляет больше энергии, чем комбинационное устройство, собранное на логических элементах. Но всё же в некоторых ситуациях это может быть оправданно. Например, в Logisim скорость моделирования сильнее зависит от количества элемента, чем от их сложности. Поэтому в ситуациях, когда скорость моделирования критична (например, при моделировании целого процессора), может быть разумно заменять комбинационные схемы из логических элементов на ПЗУ с введённой в него таблицей истинности.

Задание 1. Спроектировать устройство, заполняющее ОЗУ случайными значениями (с выхода компонента «Генератор случайных чисел» из библиотеки «Память»). Интерфейс данных ОЗУ (в этом задании и во всех последующих) — «Один асинхронный порт чтения/записи». Разрядность адреса ОЗУ — 10 битов, разрядность данных — 8 битов. После заполнения сохранить образ памяти в файл.

Задание 2. Спроектировать устройство, потактово считывающее содержимое одного ОЗУ, вычисляющее противоположное по знаку значение (в дополнительном коде) для каждого считанного значения, и записывающее результат во второе ОЗУ. Параметры ОЗУ — из задания 1. Перед началом считывания загрузить в ОЗУ-источник образ памяти, сохранённый в процессе выполнения задания 1; после копирования сохранить содержимое второго ОЗУ в отдельный файл.

Задание 3. Спроектировать устройство, потактово считывающее содержимое двух ОЗУ и складывающее считанные значения. Параметры ОЗУ — из задания 1. Перед считыванием загрузить в первое ОЗУ образ памяти, сохранённый в процессе выполнения задания 1, а во второе — сохранённый в процессе выполнения задания 2. Показать наглядно, что результат сложения всегда равен нулю.

Задание 4. Спроектировать устройство, осуществляющее копирование содержимого одного ОЗУ в другое по одному проводу (плюс провод для

передачи синхронизирующих импульсов), то есть нужно производить мультиплексирование сигнала. Параметры ОЗУ — из задания 1. Для реализации устройства нужно использовать схемы, созданные в результате выполнения задания 1 лабораторной работы 5. Перед копированием загрузить в первое ОЗУ образ памяти, сохранённый в процессе выполнения задания 1.

Задание 5. Спроектировать комбинационное устройство, описание которого приведено в задании 1 лабораторной работы 1, на основе ПЗУ (введя в ПЗУ таблицу истинности). Продемонстрировать работоспособность устройства. Один из способов выполнения этого задания — спроектировать небольшую схему, которая последовательно подаёт на входы устройства из лабораторной работы 1 и на адресный вход ОЗУ многобитные значения соответствующей разрядности, а многобитные значения с выходов этого устройства подаёт на вход данных ОЗУ для записи. После перебора всех вариантов входных значений содержимое ОЗУ можно сохранить в файл-образ, а затем загрузить этот образ в ПЗУ.

Требования к выполнению работы: все задания выполняются в одном файле проекта Logisim; каждое самостоятельное устройство

должно быть оформлено в виде отдельной схемы с осмысленным названием всех входов и выходов и самой схемы. Для защиты каждого задания нужно продемонстрировать работоспособность каждого отдельного устройства на тестовой схеме.

Лабораторная работа 8. Устройства ввода и вывода

Для ввода и вывода данных (то есть для взаимодействия схемы с пользователем) в Logisim существует несколько компонентов, они представлены в библиотеке «Ввод/вывод». Довольно полную информацию о них можно получить в справке Logisim по этой библиотеке; для выполнения этой лабораторной работы рекомендуется прочитать этот раздел справки Logisim.

Самое примитивное устройство ввода в Logisim — компонент «Кнопка». Он имеет единственный однобитный выход, на котором «0» в нормальном состоянии, и «1», если кнопка нажата. Для нажатия на кнопку нужно воспользоваться инструментом «Нажатие» (как и всякий раз, когда нужно провзаимодействовать с компонентом в схеме).

Простейшее устройство вывода — компонент «Светодиод». Когда на его единственный однобитный вход поступает «1», светодиод «загорается», при подаче на вход «0» светодиод потушен. Цвет включенного и выключенного светодиода можно настраивать.

Классическое устройство отображения информации — компонент «7-сегментный индикатор», хорошо знакомый по бытовой цифровой технике. Символы (как правило, цифры) формируются из семи сегментов и десятичной точки, каждый из которых может быть подсвечен посредством подачи «1» на соответствующий однобитный вход. Соответствие между входами и сегментами, а также принятые буквенные обозначения сегментов показаны на рисунке 1.

Такой индикатор способен формировать все шестнадцатеричные цифры, поэтому было бы удобно иметь комбинационное устройство, принимающее на входе четырёхбитное значение (шестнадцатеричную цифру), и имеющее семь выходов на сегменты индикатора. Его несложно спроектировать, начав с составления таблицы истинности (семь булевых функций от четырёх переменных). Далее можно реализовать это устройство на логических элементах, или введя таблицу истинности в ПЗУ.

Однако в библиотеке Logisim уже есть компонент «Шестнадцатиразрядный индикатор»,

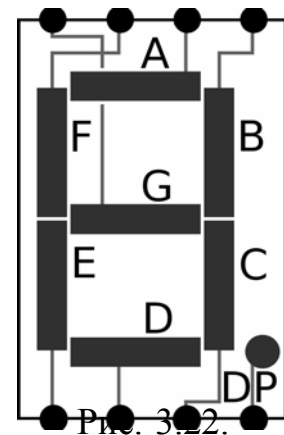


Рис. 1
Соответствие между входами и сегментами

- «Выбор Строки/Столбцы»: два входа на западном крае компонента. Верхний многобитный вход имеет столько битов, сколько столбцов в матрице; нижний многобитный вход имеет столько битов, сколько строк в матрице.

В первых двух форматах входа к матрице непосредственно подводится столько бит, сколько точек в матрице. Такой вариант крайне трудно осуществим на практике, и мы не будем использовать его в работе; как правило, промышленные матрицы имеют формат входа, соответствующий варианту «Выбор Строки/Столбцы» в Logisim. При таком варианте для формирования изображения используется развёртка (англ. scanning): в течение одного такта только один бит многобитного входа, определяющего подсвечиваемые строки, имеет значение «1». В этот момент на другой многобитный вход, определяющий подсвечиваемые столбцы, подаётся значение, в котором каждый бит «1» соответствует точке выбранной строки, которая должна быть подсвечена. После поочерёдного прохождения всех строк, снова подсвечивается первая, и т.д. Точки (пиксели) в составе реальных матриц не гаснут мгновенно, и за счёт этой инерции при прохождении развёртки не возникает мерцания. Чтобы сымитировать это, у компонента

«Светодиодная матрица» в Logisim есть атрибут «Продолжительность

свечения», который определяет, сколько тактов точка остаётся подсвеченной после подачи на неё единицы. Это довольно грубое приближение, однако оно позволяет успешно формировать развёртку. Чтобы полностью избежать мерцания, значение атрибута

«Продолжительность свечения» должно быть равно удвоенному количеству строк в матрице. На рисунке 3 изображена примитивная схема, использующая четырёхбитный счётчик и декодер, формирующую развёртку, и поочерёдно

подающая на вход матрицы значения из ПЗУ. Разрядность адреса ПЗУ (4 бита) позволяет адресовать 16 значений (по числу строк в матрице), а разрядность данных (8 бит) равна количеству столбцов в матрице. Значения в ПЗУ введены таким образом, чтобы при их последовательном выводе в строки матрицы

формировалось нужное

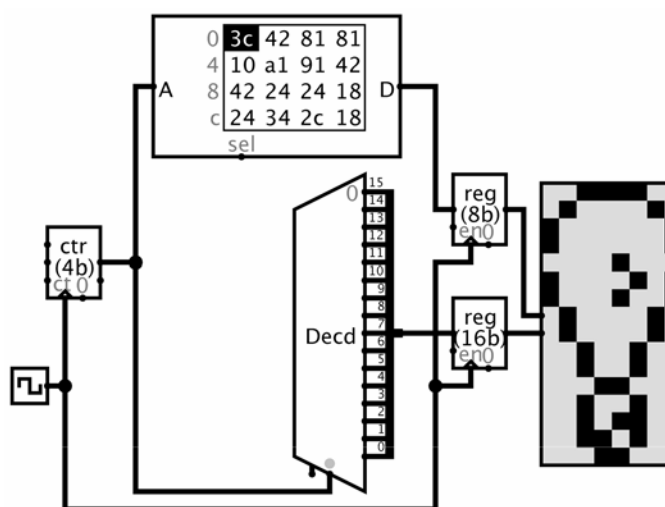


Рис. 3. Схема для формирования изображения на матрице

изображение. Перед обоими входами матрицы включены регистры соответствующей разрядности. Они позволяют подавать новый номер строки и значение, определяющее новое содержимое этой строки, строго одновременно (синхронно) — с поступлением на регистры нового тактового импульса. Если эти значения будут приходиться с ПЗУ и декодера не одновременно, то изображение будет формироваться неправильно. Обратите внимание, что в этой схеме разветвитель не только объединяет 16 однобитных значений в пучок, но и меняет их порядок на противоположный, чтобы развёртка пробегала строки сверху- вниз, а не наоборот.

В Logisim есть два компонента для работы с текстовыми символами - «Клавиатура» и «Терминал». Компонент «Клавиатура» позволяет схеме считывать символы с клавиатуры, если только они представимы в 7-разрядном ASCII коде (в него входят буквы латинского алфавита, цифры и специальные символы; кириллица в него не входит). После нажатия на компонент с помощью инструмента «Нажатие», пользователь может печатать символы, которые накапливаются в буфере. ASCII значение крайнего левого символа буфера постоянно поступает на крайний правый выход. Когда срабатывает тактовый вход, крайний левый символ исчезает из буфера и новый левый символ поступает на крайний правый выход. Поддерживаемые символы — все печатаемые ASCII символы, а также пробел, символ перехода на новую строку (Enter), Backspace и Control-L (очистка экрана). В дополнение, клавиши

«стрелка влево» и «стрелка вправо» перемещают курсор внутри буфера, а клавиша Delete удаляет символ справа от курсора (если он есть).

Компонент «Терминал» принимает последовательность ASCII кодов и при срабатывании тактового входа отображает каждый печатаемый символ, поступающий на его вход. Если текущая строка становится полной, то курсор перемещается на следующую строку, возможно прокручивая все текущие строки вверх, если курсор уже находился в нижнем ряду. Кроме того, компонент соответствующим образом реагирует на дополнительные коды, описанные выше.

На рисунке 4 приведена примитивная схема, которая каждый такт выводит на терминал очередной символ, введённый пользователем с клавиатуры.

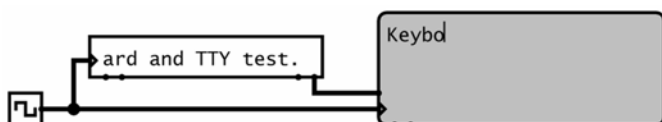


Рис. 4. Схема, выводящая введённые символы

Задание 1. Спроектировать в виде отдельной схемы комбинационное устройство, принимающее на входе 4-битное значение, и имеющее семь однобитных выходов. При подключении к выходам 7-сегментного индикатора, он должен отображать шестнадцатеричное представление значения на входе. Допустимо реализовать устройство на логических элементах или на основе ПЗУ. Отредактируйте в Logisim внешний вид подсхемы (в частности, расположение выходных контактов) таким образом, чтобы 7-сегментный индикатор мог подключаться к выходам подсхемы вообще без проводов, но при этом индикатор и подсхема не должны перекрываться.

Задание 2. Спроектировать комбинационное устройство, принимающее на входе беззнаковое 16-разрядное значение, и выводящее его десятичное представление на пять 7-сегментных индикаторов, каждый из которых подключен с помощью устройства из задания 1.

Задание 3. Спроектировать устройство для реализации следующей ситуации. Имеется два 16-разрядных регистра. Пользователь имеет в распоряжении четыре кнопки. Нажимая на первую или вторую, пользователь уменьшает или увеличивает значение первого регистра на значение второго регистра. Нажимая на третью или четвёртую кнопки, пользователь уменьшает или увеличивает значение второго регистра в 10 раз. Начальное значение второго регистра — 1, и оно не может опуститься ниже этого значения. Кнопки должны иметь осмысленные метки. Содержимое регистров отображается с помощью устройств из задания 2. Для арифметических действий допустимо использовать любые компоненты из библиотеки Logisim. Обратите внимание, что это устройство не требует наличия тактового генератора — тактовые импульсы для регистров генерируются нажатием кнопок.

Задание 4. Спроектировать устройство, выводящее на светодиодную матрицу размером 32x32 какое-либо осмысленное изображение, предварительно записанное в ПЗУ.

Задание 5. Спроектировать устройство, выводящее на светодиодную матрицу размером 32x32 «белый шум» — случайные значения с выхода компонента «Генератор случайных чисел» из библиотеки «Память».

Задание 6. Спроектировать устройство, реализующее совместный доступ устройств ввода и вывода к ОЗУ. Разрядность адреса ОЗУ — 5 битов, разрядность данных — 7 битов. По нечётным тактам в ячейку ОЗУ со случайным адресом (сгенерированным генератором случайных чисел) записывается ASCII код символа из буфера компонента

«Клавиатура», причём если буфер пуст (на выходе компонента — значение «0»), то запись не происходит. По чётным тактам происходит вывод содержимого ОЗУ на светодиодную матрицу размером 7x32 с помощью развёртки, то есть каждый чётный такт — следующая строка точек.

Задание 7. Спроектировать устройство, принимающее вводимые пользователем с клавиатуры символы, определённым образом обрабатывающее их, и выводящее на терминал. Алгоритм обработки указан в таблице 3.10. Вариант определяется по последней цифре номера студента в списке группы. Подразумевается, что символы, для которых таблицей не предусмотрена обработка, выводятся без изменений.

По сути, задание сводится к проектированию комбинационного устройства, которое нужно разместить между компонентами

«Клавиатура» и «Терминал». Поскольку количество символов в 7-битном наборе ASCII довольно велико (128), заполнять таблицу истинности для каждого символа слишком трудоёмко. Вместо этого для определения принадлежности введённого символа к определённой группе (прописные и строчные буквы, цифры, специальные символы) следует использовать компонент «Компаратор» из библиотеки «Арифметика».

Для получения таблицы соответствия 7-битных ASCII кодов и символов набора можно спроектировать схему, посылающую последовательные 7-битные значения на вход компонента «Терминал». Для обратной задачи (выяснения кода определённого символа) можно спроектировать другую схему — подключить выход компонента

«Клавиатура» к какому-либо устройству отображения (например, к компоненту «Датчик»), вводить интересующие символы и считывать их коды.

Таблица 3.10. Варианты заданий для лабораторной работы 7

Вари	Алгоритм обработки символа
1	Заменять все прописные буквы строчными
2	Заменять все строчные буквы прописными
3	Заменять каждую цифру большей на единицу
4	Не выводить никаких символов кроме букв и цифр
5	Заменять каждую прописную букву следующей в алфавите
6	Заменять каждую строчную букву следующей в алфавите
7	Заменять прописные гласные строчными
8	Заменять строчные гласные прописными
9	Заменять все согласные на тире
1	Выводить вместо символа младшую половину его шестнадцатеричного представления

Требования к выполнению работы: все задания выполняются в одном файле проекта Logisim; каждое самостоятельное устройство должно быть оформлено в виде отдельной схемы с осмысленным названием всех входов и выходов и самой схемы. Для защиты каждого задания нужно продемонстрировать работоспособность каждого отдельного устройства на тестовой схеме.

Вопросы для тестирования

1. _____ представляют собой законченное микроэлектронное устройство, способное выполнять функции аппаратуры
 - сверхбольшие интегральные схемы
 - малые интегральные схемы
 - транзисторы
 - лампы

2. Амплитудно-частотная характеристика (АЧХ) — это
 - зависимость модуля коэффициента усиления от частоты
 - зависимость модуля коэффициента усиления от напряжения
 - зависимость модуля коэффициента усиления от силы тока
 - зависимость модуля коэффициента усиления от мощности

3. Аналоговые интегральные схемы предназначены для
 - обработки сигналов, заданных в виде непрерывной функции
 - обработки сигналов, заданных в виде линейной функции
 - обработки сигналов, заданных в виде константы

4. Буквой Н в системе условных обозначений типов зарубежных интегральных схем определяют
 - МОП микросхемы
 - гибридные микросхемы
 - новые микросхемы
 - ДОП микросхемы

5. Важными параметрами усилителя являются
 - его входное и выходное напряжение
 - его входное и выходное сопротивления
 - его входное сопротивление
 - его выходное сопротивление

6. Второй элемент системы условных обозначений типов зарубежных интегральных схем представляет собой
 - букву, обозначающую гибридную микросхему
 - букву, выбираемую фирмой-изготовителем и не имеющую специального назначения;
 - букву, обозначающую страну изготовитель
 - букву, обозначающую страну импортер

7. Входной ток логической единицы интегральных схем - это
- входной ток, обеспечивающий формирование логической единицы
 - входной ток, обеспечивающий формирование логического нуля
 - входной ток, обеспечивающий формирование синусоидального сигнала
 - входной ток, обеспечивающий формирование логического нуля или единицы
8. Выходной каскад дифференциального усилителя выполнен по схеме
- двухтактного эмиттерного повторителя
 - двухтактного коллекторного повторителя
 - двухтактного повторителя
 - двухтактного усилителя
9. Генератор – это
- микросэлектронное устройство, предназначенное для создания электрических колебаний заданной формы и частоты
 - микросэлектронное устройство, предназначенное для преобразования поступающих сигналов
 - микросэлектронное устройство, предназначенное для создания сдвига по фазе
 - микросэлектронное устройство, предназначенное для усиления поступающих сигналов
10. Демультимплексор – операционный элемент, который
- осуществляет адресное подключение одного входного сигнала к одному из множества выходов
 - осуществляет адресное подключение многих входов сигнала к многим выходам
 - осуществляет адресное подключение одного входного сигнала к одному выходу
 - осуществляет адресное подключение многих выходов сигнала к одному входу
11. Дешифратор – операционный элемент, который
- преобразует n -разрядный входной код в сигнал только на одном из своих m выходов
 - преобразует m -разрядный входной код в сигнал только на одном выходе
 - осуществляет адресное подключение одного входного сигнала к одному из множества выходов
12. Дифференциальным входным сигналом называют
- разность напряжений, формируемых на входах
 - разность напряжений, формируемых на выходах
 - разность токов, формируемых на входах

- разность мощностей, формируемых на входах
13. Заказная интегральная схема представляет собой
- класс интегральных схем со стандартными элементами или узлами по заранее заданной функциональной схеме
 - конструктивно законченное изделие электронной техники, содержащее совокупность электрически связанных в функциональную схему электрорадиоэлементов, изготовленных в едином технологическом цикле
 - схемы без запоминания переменных
14. Инверсией называется
- операция НЕ и обозначается штрихом "-" переменной
 - операция ИЛИ и обозначается штрихом "+" переменной
 - операция И и обозначается штрихом "&" переменной
 - операция ИСКЛЮЧАЮЩЕЕ ИЛИ
15. Интегральная схема представляет собой
- конструктивно законченное изделие электронной техники, содержащее совокупность электрически связанных в функциональную схему электрорадиоэлементов, изготовленных в едином технологическом цикле
 - класс интегральных схем со стандартными элементами или узлами по заранее заданной функциональной схеме
 - схемы без запоминания переменных
 - схемы с временным запоминанием
16. Комбинационные логические схемы представляют собой
- схемы без запоминания переменных
 - конструктивно законченное изделие электронной техники, содержащее совокупность электрически связанных в функциональную схему электрорадиоэлементов, изготовленных в едином технологическом цикле
 - класс интегральных схем со стандартными элементами или узлами по заранее заданной функциональной схеме
 - схемы с запоминанием переменных
17. Компаратор – операционный элемент, который
- производит умножение двух чисел
 - производит СЛОЖЕНИЕ двух чисел
 - производит вычитание двух чисел
 - производит сравнение двух чисел
18. Логарифмический умножитель представляет собой
- устройство умножения двух или более аналоговых членов путем использования сложения синусов этих сигналов
 - устройство сложения двух или более аналоговых членов путем использования сложения логарифмов этих сигналов
 - устройство умножения двух или более аналоговых членов путем использования сложения логарифмов этих сигналов
 - устройство вычитания двух или более аналоговых членов путем

19. Логическими элементами ИС называются

- электронные схемы, выполняющие две логические операции
- электронные схемы, выполняющие три логические операции
- электронные схемы, выполняющие простейшие символьные операции
- электронные схемы, выполняющие простейшие логические операции

20. Максимальное обратное напряжение на переходах интегральных схем - это

- наибольшее значение падения тока на переходах интегральной схемы при протекании обратного тока
- наибольшее значение падения напряжения на переходах интегральной схемы при протекании прямого тока
- наибольшее значение падения напряжения на переходах интегральной схемы при протекании обратного тока
- наибольшее значение увеличения напряжения на переходах интегральной схемы при протекании прямого тока

21. **Малые интегральные схемы (МИС – 2.1) содержат**

- до 200 элементов и компонентов на кристалле
- до 300 элементов и компонентов на кристалле
- до 1000 элементов и компонентов на кристалле
- до 100 элементов и компонентов на кристалле

22. Микропроцессорные комплекты - это

- серии ИС программно и технологически несовместимые
- серии ИС программно и технологически равные
- серии ИС программно и технологически совместимые
- серии ИС программно не совместимые

23. Микропроцессоры – это

- класс ИС для транзисторных устройств
- класс ИС для аналоговых устройств
- класс ИС для вычислительных устройств
- класс ИС для ПЗУ

24. Модулятор – это

- микроэлектронное устройство, управляющее заданным параметром колебательного процесса в соответствии с сигналами передаваемого сообщения
- микроэлектронное устройство, управляющее заданным параметром колебательного процесса в соответствии с напряжением
- микроэлектронное устройство, управляющее заданным параметром силы тока в соответствии с сигналами передаваемого сообщения
- микроэлектронное устройство, управляющее заданным параметром колебательного процесса в соответствии с сопротивлением

25. Мультиплексор – операционный элемент, который
- микроэлектронное устройство, управляющее заданным параметром колебательного процесса в соответствии с сигналами передаваемого сообщения
 - осуществляет переключение трех входных сигналов на один выход
 - осуществляет адресное переключение заданного числа входных сигналов на один выход
 - осуществляет переключение трех входных сигналов на три выхода
26. Нагрузочная способность характеризует
- максимальное число логических элементов, которые можно одновременно подключить к выходу
 - минимальное число логических элементов, которые можно одновременно подключить к выходу
 - максимальное число логических элементов, которые можно одновременно подключить к входу
 - большое число логических элементов, которые можно одновременно подключить к выходу
27. Нанесение окисла в вакууме происходит
- путем вакуумного испарения или катодного расплавления
 - путем вакуумного испарения или анодного расплавления
 - путем вакуумного испарения или эмитторного расплавления
 - путем испарения или замерзания
28. Напряжение логического нуля интегральных схем - это
- значение низкого уровня тока для "положительной" логики и значение высокого уровня напряжения для "отрицательной" логики
 - значение высокого уровня напряжения для "положительной" логики и значение высокого уровня напряжения для "отрицательной" логики
 - значение низкого уровня напряжения для "положительной" логики и значение высокого уровня напряжения для "отрицательной" логики
 - это 1
29. Напряжение отпускания интегральных схем - это
- среднее значение напряжения постоянного тока на входе, при котором происходит переход ИС из одного устойчивого состояния в другое
 - наименьшее значение напряжения постоянного тока на входе, при котором происходит переход ИС из одного устойчивого состояния в другое
 - наибольшее значение напряжения постоянного тока на входе, при котором происходит переход ИС из одного устойчивого состояния в другое
30. Напряжение смещения интегральных схем - это
- значение напряжения постоянного тока на входе ИС, при котором

выходное напряжение равно нулю

- значение тока постоянного тока на входе ИС, при котором выходное напряжение равно нулю
- значение напряжения постоянного тока на входе ИС, при котором выходное напряжение равно единице значение напряжения постоянного тока на входе
- ИС, при котором выходное напряжение равно единице или нулю.

31. Напряжение срабатывания интегральных схем - это

- наибольшее значение напряжения постоянного тока на входе, при котором происходит переход ИС из одного устойчивого состояния в другое
- среднее значение напряжения постоянного тока на входе, при котором происходит переход ИС из одного устойчивого состояния в другое
- наименьшее значение напряжения постоянного тока на входе, при котором происходит переход ИС из одного устойчивого состояния в другое
- значение напряжения постоянного тока на входе равное 1В.

32. Неинвертирующий усилитель представляет собой

- микросэлектронное устройство, позволяющее усилить сигнал без изменения полярности
- микросэлектронное устройство, позволяющее гасить сигнал без изменения полярности
- микросэлектронное устройство, позволяющее усилить сигнал с изменением полярности
- микросэлектронное устройство, позволяющее усиливать сигнал на 10%.

33. Нижняя граничная частота полосы пропускания интегральных схем - это

- наименьшее значение частоты, на которой коэффициент усиления интегральной схемы уменьшается на 3 дБ при заданной частоте
- наибольшее значение частоты, на которой коэффициент усиления интегральной схемы уменьшается на 3 дБ при заданной частоте
- среднее значение частоты, на которой коэффициент усиления интегральной схемы уменьшается на 3 дБ при заданной частоте
- частота равная 0

34. Операционные усилители конструктивно выполняются в виде

- интегральных схем большой степени интеграции
- интегральных схем малой степени интеграции
- интегральных схем средней степени интеграции
- транзисторов

35. Операционный усилитель имеет в основе

- дифференциальный усилитель
- предварительный усилитель
- интегральный усилитель

- два конденсатора

36. Операционный усилитель обязательно имеет

- внешние цепи
- внутренние цепи
- обратные цепи
- два транзистора

37. Операционный усилитель – это

- микросэлектронное устройство, предназначенное для усиления как постоянного тока, так и электрических колебаний
- микросэлектронное устройство, предназначенное для усиления как переменного тока, так и электрических колебаний
- микросэлектронное устройство, предназначенное для усиления как постоянного тока, так и электрических колебаний
- микросэлектронное устройство, предназначенное для выпрямления постоянного тока

38. От короткого замыкания операционный усилитель защищают

- резисторы-ограничители
- транзисторы-ограничители
- диоды-ограничители
- предохранители

39. Первый элемент системы условных обозначений типов зарубежных интегральных схем представляет собой

- букву, отражающую принцип преобразования сигнала
- букву, отражающую фирму производитель
- букву, отражающую температурный режим
- букву, отражающую стоимость

40. Первый элемент системы условных обозначений типов отечественных интегральных схем представляет собой

- букву, отражающую принцип преобразования сигнала
- цифру, обозначающую группу интегральной микросхемы по конструктивно- технологическому исполнению
- цифру, обозначающую группу интегральной микросхемы по конструктивно- технологическому исполнению
- цифру, обозначающую год выпуска

41. Переходная характеристика усилителя — это

- зависимость выходного сигнала (тока, напряжения) от времени при скачкообразном входном воздействии
- зависимость выходного сигнала (тока, напряжения) от напряжения при скачкообразном входном воздействии
- зависимость входного сигнала (тока, напряжения) от времени при

скачкообразном входном воздействии

- зависимость входного сигнала от температуры

42. Полосовые фильтры - это

- микроэлектронное устройство, пропускающее колебания в заданном интервале частот
- микроэлектронное устройство, не пропускающее колебания в заданном интервале частот
- микроэлектронное устройство, пропускающее колебания в заданном интервале напряжения
- микроэлектронное устройство, пропускающее четные колебания

43. Полузаказная интегральная схема представляет собой

- класс интегральных схем, имеющих постоянную и переменную части
- класс интегральных схем, имеющих переменную часть
- класс интегральных схем, имеющих переменную часть
- класс интегральных схем, имеющих ровно 10000 транзисторов

44. Пороговое напряжение логического нуля интегральных схем - это

- наибольшее значение низкого уровня напряжения для "положительной" логики на входе ИС, при котором происходит переход ИС из одного устойчивого состояния в другое
- наибольшее значение высокого уровня напряжения для "положительной" логики на входе ИС, при котором происходит переход ИС из одного устойчивого состояния в другое
- наибольшее значение низкого уровня напряжения для "отрицательной" логики на входе ИС, при котором происходит переход ИС из одного устойчивого состояния в другое
- напряжение равно 1В.

45. Пороговое напряжение логической единицы интегральных схем - это

- наименьшее значение высокого уровня напряжения для "положительной" логики на входе ИС, при котором происходит переход ИС из одного устойчивого состояния в другое
- наибольшее значение высокого уровня напряжения для "положительной" логики на входе ИС, при котором происходит переход ИС из одного устойчивого состояния в другое
- наименьшее значение низкого уровня напряжения для "положительной" логики на входе ИС, при котором происходит переход ИС из одного устойчивого состояния в другое
- напряжение равно 1,5 В.

46. Преобразователь – это

- микроэлектронное устройство, преобразующее параметры сигналов: частоты, фазы, длительности, мощности и т.д.
- микроэлектронное устройство, пропускающее колебания в заданном интервале напряжения

- класс интегральных схем, имеющих переменную часть
- класс интегральных схем, имеющих полевые транзисторы

47. Программируемые логические матрицы представляют собой

- готовые изделия, конструктивно содержащие две вентиляльные матрицы
- готовые изделия, конструктивно содержащие одну вентиляльную матрицу
- готовые изделия, конструктивно содержащие три вентиляльные матрицы
- матрицы поддерживающие языки программирования высокого уровня

48. Регистр – последовательный элемент, который

- предназначен для передачи и (или) преобразования многоразрядных двоичных чисел
- предназначен для хранения многоразрядных двоичных чисел
- предназначен для хранения и (или) преобразования многоразрядных двоичных чисел
- предназначен для хранения только десятичных чисел

49. Результатом процесса формирования топологии интегральных схем являются

- формирование физических и геометрических параметров активных и пассивных элементов заданной схемы заданного функционального назначения
- формирование геометрических параметров активных и пассивных элементов заданной схемы заданного функционального назначения
- формирование физических и геометрических параметров активных элементов заданной схемы заданного функционального назначения
- формирование физических параметров активных элементов заданной схемы заданного функционального назначения

50. Синфазным входным сигналом называют

- разность полусуммы напряжений, формируемых на входах
- разность произведения напряжений, формируемых на входах
- разность полусуммы токов, формируемых на входах
- Сигнал с напряжением равным 1В

51. Счетчик – последовательный элемент, который

- предназначен для счета импульсов, поступающих на вход
- предназначен для счета импульсов, поступающих на выход
- предназначен для суммы импульсов, поступающих на вход
- предназначен для счета только нечетных импульсов, поступающих на выход

52. Ток утечки на входе интегральных схем - это

- значение тока во выходной цепи интегральной схемы при закрытом состоянии входа и заданных режимах на остальных выводах
- значение тока во входной цепи интегральной схемы при закрытом состоянии входа и заданных режимах на остальных выводах

- значение тока во входной цепи интегральной схемы при закрытом состоянии выхода и заданных режимах на остальных выводах
- значение тока во входной цепи интегральной схемы при открытом состоянии входа и заданных режимах на остальных выводах

53. Транзисторы приоткрывают в режиме покоя для

- увеличения нелинейных искажений
- уменьшения нелинейных искажений
- уменьшения линейных искажений
- умножения линейных искажений

54. Третий элемент системы условных обозначений типов зарубежных интегральных схем представляет собой

- букву, фирму производитель
- букву, обозначающую диапазон рабочих температур или другую важную характеристику
- букву, обозначающую год выпуска
- букву, обозначающую страну производитель

55. Триггер – последовательный элемент, который

- имеет два устойчивых выходных состояния
- имеет три устойчивых выходных состояния
- имеет четыре устойчивых выходных состояния
- имеет пять устойчивых выходных состояния

56. Усилитель – это

- микросэлектронное устройство, предназначенное для усиления сигналов в заданном диапазоне частот
- микросэлектронное устройство, предназначенное для уменьшение сигналов в заданном диапазоне частот
- микросэлектронное устройство, предназначенное для суммирования сигналов в заданном диапазоне частот
- то же самое что и транзистор

57. Фазочастотная характеристика (ФЧХ) — это

- зависимость угла сдвига фаз между входным и выходным напряжениями от частоты
- зависимость угла сдвига фаз между выходным и входным напряжениями от напряжения
- зависимость угла сдвига фаз между входным и выходным напряжениями от тока
- сила тока равная 1А

58. Фильтр – это

- микросэлектронное устройство, предназначенное для разделения

- электрических колебаний различных частот
- микронэлектронное устройство, предназначенное для суммирования электрических колебаний различных частот
- микронэлектронное устройство, предназначенное для разделения электрических колебаний различных частот
- устройство для усиления электрических сигналов

59. Фильтры нижних частот - это

- микронэлектронное устройство, пропускающее колебания не выше заданной граничной частоты
- микронэлектронное устройство, пропускающее колебания не ниже заданной граничной частоты
- микронэлектронное устройство, не пропускающее колебания не выше заданной граничной частоты
- микронэлектронное устройство, предназначенное для суммирования электрических колебаний различных частот

60. Чувствительность интегральных схем - это

- наименьшее значение входного напряжения, при котором электрические параметры интегральной схемы соответствуют заданным значениям
- наибольшее значение входного напряжения, при котором электрические параметры интегральной схемы соответствуют заданным значениям
- наименьшее значение выходного напряжения, при котором электрические параметры интегральной схемы соответствуют заданным значениям
- то же самое что и количество входов

61. Полупроводники по проводимости находятся . . .

- между диэлектриком и проводником
- наполовину выше проводников
- наполовину выше диэлектриков
- наполовину ниже диэлектриков
- наполовину ниже проводников

62. К недостаткам полупроводниковых приборов относится ...

- работа не с основными носителями
- необходимость низкого напряжения
- необходимость вакуума
- зависимость электропроводности кристалла
- ограниченный температурный режим

63. К полупроводникам р-типа относятся ...

- полупроводник с избытком концентрации дырок
- кристалл обладающий избытком концентрации электронов
- рекомбинированный переход
- кристаллическая решетка с избытком электронов
- полупроводник с зоной проводимостью

64. К полупроводникам p —типа относятся ...

- кристалл обладающий избытком концентрации электронов
- рекомбинированный переход
- полупроводник с зоной проводимостью
- полупроводник с избытком концентрации дырок
- кристаллическая решетка с избытком электронов

65. Для включения полупроводникового p - n перехода в прямом направлении необходимо ...

- полярность внешнего источника питания изменяют на противоположную
- положительный полюс источника соединяют с выводом от p -области, а отрицательный - с выводом от n -области
- положительный полюс питания соединяют с выводом от n -области, а отрицательный - с p —областью
- изменить структуру кристаллической решетки полупроводника
- изменить полярность внутреннего источника питания

66. Состояние, когда p - n переходу называется нейтральным.

- приложено обратное напряжение -
- не приложено никакое внешнее напряжение
- приложено прямое напряжение, равное потенциальному барьеру
- приложено обратное напряжение, равное потенциальному барьеру
- приложено прямое напряжение, обратное потенциальному барьеру

67. Для включения полупроводникового p - n перехода в обратном направлении необходимо ...

- положительный полюс питания соединить с выводом от n -области, а
- отрицательный с p -областью
- полярность внешнего источника питания изменить на противоположную
- положительный полюс источника соединить с выводом от p -области, а отрицательный - с выводом от n -области
- изменить структуру кристаллической решетки полупроводника
- изменить полярность внутреннего источника питания

68. Основное свойство полупроводникового диода ...

- пропускать ток в обратном направлении
- преобразовать постоянный ток в пульсирующий
- не пропускать постоянный ток
- преобразовать постоянный ток в переменный
- пропускать ток в прямом направлении

69. Основной недостаток полупроводникового диода заключается в

- в наличии малого обратного тока
- в наличии резкого увеличения прямого тока
- в зависимости выпрямленного тока от напряжения
- отсутствие в характеристике плавно изменяющейся кривизны
- в наличии большого прямого тока

70. Чему равен результат сложения чисел 1102 и 128?

- 1) 610
- 2) 1010
- 3) 100002
- 4) 178

71. Ячейка памяти компьютера состоит из однородных элементов, называемых:

- кодами
- разрядами
- цифрами
- коэффициентами

72. Обработывающее и управляющее устройство, выполненное с использованием технологии БИС (часто на одном кристалле) и обладающее способностью выполнять под программным управлением обработку информации, включая ввод и вывод информации, арифметические и логические операции и принятие решений называется ...

- микропроцессором
- процессором
- микропроцессорной *системой*

73. Действия по выбору из памяти и выполнению одной команды называются ...

- машинный цикл
- машинный такт
- командный цикл

74. Какой режим обмена предполагает отключение процессора?

- процессор никогда не отключается
- программный обмен
- обмен по прямому доступу к памяти
- обмен по прерываниям

75. Какой тип обмена используется в системной магистрали ISA?

- синхронный
- асинхронный
- синхронный с возможностью асинхронного обмена
- мультиплексированный

76. Структура какой шины влияет на разнообразие режимов обмена?

- шины данных
- шины управления
- шины питания
- шины адреса

77. Какая архитектура обеспечивает более высокое быстродействие?

- принстонская
- гарвардская
- фон-неймановская
- *быстродействие не зависит от архитектуры*

78. Какая структура шин адреса и данных обеспечивает большее быстродействие?

- мультиплексированная
- немultipлексированная
- двунаправленная
- *быстродействие от структуры не зависит*

79. Какой режим обмена обеспечивает наибольшую скорость передачи информации?

- обмен по прямому доступу к памяти
- программный обмен
- обмен по прерываниям
- все режимы одинаковы по быстродействию

80. Процессор, функционирующий с сокращенным набором команд:

- CISC
- RISC
- MISC
- VLIW

81. К основным параметрам МП не относится:

- тактовая частота
- внутренняя разрядность данных
- пропускная способность
- адресуемая память

82. Количество бит, которые МП может обрабатывать одновременно -а) внешняя разрядность данных

- б) тактовая частота
- в) внутренняя разрядность данных
- г) степень интеграции микросхемы

83. От разрядности микропроцессора зависит:

- а) количество используемых внешних устройств
- б) максимальный объем внутренней памяти и производительность компьютера
- в) возможность подключения к сети
- г) возможность сжатия данных

84. Конвейеризация – это...

- а) процесс реализации процессорных команд по нескольким линиям
- б) технология обработки команд
- в) многопоточная параллельная обработка команд
- г) технология обработки данных несколькими процессорами одновременно

85. Корпуса процессоров бывают:

- а) корпус с односторонним контактом и безкорпусный
- б) матрица со штырьковыми выводами и пленочный
- в) корпус с открытым кристаллом и матричный

86. Процессор имеет 14 регистров общего назначения. Сколько разрядов в поле команды необходимые для адресации к ним:

- а) 7
- б) 4
- в) 3
- г) 8

87. Представить десятичное число 45 в двоичном коде.

- 101101
- 110010
- 100011
- 111010

88. Какой режим микропроцессорных систем используется для передачи больших массивов информации между памятью и внешним устройством:

- ожидания
- прерывания

- прямого доступа к памяти
- прямой передачи данных

89. Режим работы микропроцессорных систем не требует обращения к внешним устройствам:

- внешний
- прерывания
- прямого доступа к памяти
- прямой передачи данных

90. Каково назначение контроллера прямого доступа к памяти:

- ускорить обмен между памятью и внешним устройством
- срочное обслуживание внешнего устройства
- выработка временных задержек
- организация обмена в последовательном коде

91. Каково назначение программного таймера:

- ускорить обмен между памятью и внешним устройством
- срочное обслуживание внешнего устройства
- выработка временных задержек
- организация обмена в последовательном коде

Вопросы, выносимые на коллоквиум

Первый коллоквиум

1. Основы алгебры логики и теории переключательных функций
2. Минимизация нескольких функций
3. Основы теории асинхронных потенциальных и синхронных автоматов
4. Асинхронные потенциальные триггеры
5. Синхронные триггеры
6. Стандартные интегральные схемы (ИС) ТТЛ и КМОП серий
7. Триггеры Шмитта. Интегральные схемы мультивибраторов и их применения.
8. Драйверы и приемопередатчики с открытым коллекторным выходом и тремя состояниями выхода.
9. Дешифраторы и демультиплексоры.
10. Мультиплексоры.
11. Комбинационные сумматоры.
12. Шифраторы.
13. Компараторы.
14. Сдвигающие регистры.
15. Счетчики.
16. Примеры применения интегральных схем при проектировании цифровых устройств.
17. Примеры применения интегральных схем для проектирования ввода- вывода.
18. Цифро-аналоговые преобразователи. Аналого-цифровые преобразователи.

Второй коллоквиум

1. Трехшинная архитектура микроЭВМ.
2. Архитектура однокристалльных МП.
3. Архитектура однокристалльных МП. 10.Однокристалльные МП: Форматы команд. 11.Однокристалльные МП: Адресация. 12.Однокристалльные МП: Арифметический сопроцессор.
4. Программное обеспечение микроконтроллеров.
5. Принципиальные схемы микроконтроллеров.
6. Запоминающие устройства.
7. Регистры памяти. Ввод-вывод.
8. Интерфейсные БИС: Параллельный интерфейс.
9. Интерфейсные БИС: Прерывания.
10. Интерфейсные БИС: Доступ к памяти. Связные интерфейсы.
11. Последовательные интерфейсы.
12. Мультипроцессорные системы.

Вопросы на зачет

1. Основы алгебры логики и теории переключательных функций
2. Минимизация нескольких функций
3. Основы теории асинхронных потенциальных и синхронных автоматов
4. Асинхронные потенциальные триггеры
5. Синхронные триггеры

6. Стандартные интегральные схемы (ИС) ТТЛ и КМОП серий
7. Триггеры Шмитта.
8. Интегральные схемы мультивибраторов и их применения.
9. Драйверы и приемопередатчики с открытым коллекторным выходом и тремя состояниями выхода
10. Дешифраторы и демультиплексоры
11. Мультиплексоры
12. Комбинационные сумматоры
13. Шифраторы
14. Компараторы
15. Сдвигающие регистры
16. Счетчики
17. Применения интегральных схем при проектировании цифровых устройств
18. Примеры применения интегральных схем для проектирования вводавывода
19. Цифро-аналоговые преобразователи.
20. Аналого-цифровые преобразователи
21. Трехшинная архитектура микроЭВМ
22. Архитектура однокристалльных микропроцессоров
23. Однокристалльные микропроцессоры: Форматы команд
24. Однокристалльные микропроцессоры: Адресация
25. Однокристалльные микропроцессоры: Арифметический сопроцессор
26. Программное обеспечение микроконтроллеров
27. Принципиальные схемы микроконтроллеров
28. Запоминающие устройства
29. Регистры памяти
30. Интерфейсные БИС: Параллельный интерфейс
31. Интерфейсные БИС: Прерывания
32. Интерфейсные БИС: Доступ к памяти. Связные интерфейсы
33. Последовательные интерфейсы
34. Мультипроцессорные системы