|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение высшего образования**  **«Московский государственный технический университет имени Н.Э. Баумана**  **(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Информационная безопасность» (ИУ8)

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОЙ РАБОТЕ НА ТЕМУ:**

**«Реализация устройства кодирования данных с помощью кода Хаффмана»**

|  |  |
| --- | --- |
| Студент: группа ИУ8-62 л.д. 20У582 Лагов Сергей Павлович | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |
| Руководитель курсовой работы: Ковынёв Николай Витальевич | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

**Министерство образования и науки Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

**(МГТУ им. Н.Э. Баумана)**

УТВЕРЖДАЮ

Заведующий кафедрой ИУ-8

(Индекс)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.А.Басараб

(И.О.Фамилия)

«17» февраля 2023 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине Электроника и схемотехника

Студент группы ИУ8-62

Лагов С.П.

(Фамилия, имя, отчество)

Тема курсовой работы Реализация устройства кодирования данных с помощью кода Хаффмана.

Направленность КР (учебная, исследовательская, практическая, производственная, др.)

учебная

Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения КР: 25% к 4 нед., 50% к 7 нед., 75% к 10 нед., 100% к 14 нед.

***Техническое задание реализовать устройство кодирования данных с помощью кода Хаффмана. На вход подаются данные в формате ASCII. Длина входной последовательности данных не превышает 16 байт. Входной алфавит ограничен 10-ю символами. Индикация результата осуществляется с помощью светодиодов.***

***Оформление курсовой работы:***

Расчетно-пояснительная записка на 20 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

1.Схема электрическая функциональная 2.Схема электрическая принципиальная

Дата выдачи задания «17» февраля 2023 г.

**Руководитель курсовой работы**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.В. Ковынёв

(Подпись, дата) (И.О.Фамилия)

**Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** С.П. Лагов

(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

# АННОТАЦИЯ

В курсовой работе выполняется проектирование устройства, кодирующего данные с помощью кода Хаффмана.

Цель курсовой работы – рассчитать элементы микросхемы устройства кодирования данных кодом Хаффмана, получить опыт в разработке микросхем подобного типа.

Результатом работы является принципиальная схема микросхемы, содержащая номинальные значения всех пассивных и активных элементов, которые обеспечивают правильную работу схемы устройства в соответствии с заданными требованиями.

Работа выполнена на 20 листах и включает в себя схемы, рисунки, технические данные по используемым элементам.

# СОДЕРЖАНИЕ

[АННОТАЦИЯ 2](#_Toc134934067)

[СОДЕРЖАНИЕ 4](#_Toc134934068)

[ВВЕДЕНИЕ 5](#_Toc134934069)

[ОСНОВНАЯ ЧАСТЬ 6](#_Toc134934070)

[1 Описание структурной схемы ОЗУ 6](#_Toc134934071)

[2 Составление и расчет схемы устройства 8](#_Toc134934072)

[2.1 Составление функциональной схемы обработчика входа 9](#_Toc134934073)

[2.2 Составление функциональной схемы многоадресного регистра 11](#_Toc134934074)

[2.3 Составление функциональной схемы обработчика выхода 16](#_Toc134934075)

[2.4 Реализация интерфейса ввода и вывода UART 17](#_Toc134934076)

[2.4.1 Составление функциональной схемы преобразователя входа 18](#_Toc134934077)

[2.4.2 Составление функциональной схемы преобразователя выхода 22](#_Toc134934078)

[2.5 Объединение составленных элементов 23](#_Toc134934079)

[2.6 Расчет элементов микросхемы 26](#_Toc134934080)

[2.7 Выбор реальных элементов для схемы 27](#_Toc134934081)

[2.8 Расчет мощность устройства 27](#_Toc134934082)

[3 Моделирование схемы 28](#_Toc134934083)

[ЗАКЛЮЧЕНИЕ 31](#_Toc134934084)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 32](#_Toc134934085)

[ПРИЛОЖЕНИЕ А 33](#_Toc134934086)

# ВВЕДЕНИЕ

Кодирование информации — процесс преобразования сигнала из формы, удобной для непосредственного использования информации, в форму, удобную для передачи, хранения или автоматической переработки.

Алгоритм Хаффмана — жадный алгоритм оптимального префиксного кодирования алфавита с минимальной избыточностью. Был разработан в 1952 году аспирантом Массачусетского технологического института Дэвидом Хаффманом при написании им курсовой работы. В настоящее время используется во многих программах сжатия данных.

Классический алгоритм Хаффмана на входе получает таблицу частотностей символов в сообщении. Далее на основании этой таблицы строится дерево кодирования Хаффмана (Н-дерево).

1. Символы входного алфавита образуют список свободных узлов. Каждый лист имеет вес, который может быть равен либо вероятности, либо количеству вхождений символа в сжимаемое сообщение.
2. Выбираются два свободных узла дерева с наименьшими весами.
3. Создается их родитель с весом, равным их суммарному весу.
4. Родитель добавляется в список свободных узлов, а два его потомка удаляются из этого списка.
5. Одной дуге, выходящей из родителя, ставится в соответствие бит 1, другой — бит 0. Битовые значения ветвей, исходящих от корня, не зависят от весов потомков.
6. Шаги, начиная со второго, повторяются до тех пор, пока в списке свободных узлов не останется только один свободный узел. Он и будет считаться корнем дерева.

Результатом работы будет являться спроектированное устройство, аппаратно реализующее приведенный алгоритм и кодирующее последовательность 8-битных ASCII-символов размером не более 16 и мощностью алфавита не более 10 с последовательным выводом кода. Индикация кода будет производиться на двухцветных диодах.

# ОСНОВНАЯ ЧАСТЬ

### Описание структурной схемы устройства

Структурная схема предназначена для описания принципа работы устройства и его состава в общем виде. На схеме изображаются все основные функциональные части изделия, а также основные взаимосвязи между ними. Построение такой схемы даёт наглядное представление о совместном взаимодействии функциональных узлов.

На схеме, изображенной на рисунке 1, следующие функциональные части:

1. Блок ввода данных — часть микросхемы, производящая запись входной последователь.
2. Блок сортировки данных — часть микросхемы, преобразующая данные в отсортированный по частотности по возрастанию массив пар символ-частотность.
3. Блок построения кодового дерева — часть микросхемы, структурно связывающая данные между собой в кодовое дерево Хаффмана.
4. Блок построения кода — часть микросхемы, производящая сопоставление символ-код на основе построенного дерева.
5. Блок вывода данных — часть микросхемы, реализующая последовательный вывод итогового кода в соответствии с определенным правилом.
6. ОЗУ — память устройства.
7. Блок управления ОЗУ — часть микросхемы, отвечающая за выбор источника адреса, вводимых данных, а также команд управления ОЗУ среди функциональных частей 1 — 6 на основе флагов состояния работы частей.

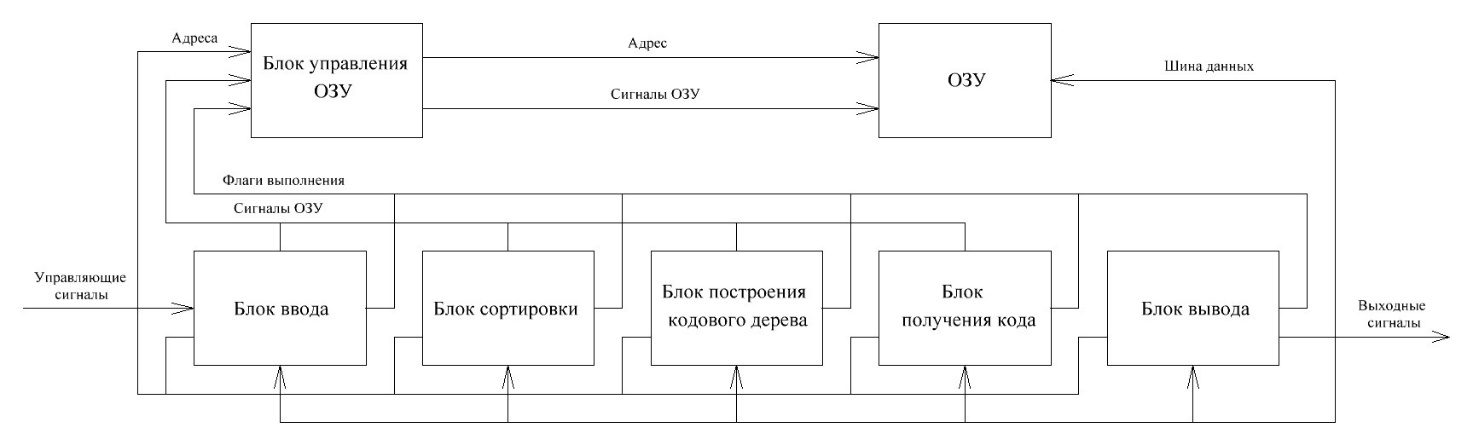


Рисунок 1 – Структурная схема кодирующего устройства

Устройство имеет управляющие сигналы: C, AE, R, S, D0…D7.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C | AE | R | S | Действие |
| 0 | X | X | X | — |
| 1 | 0 | 0 | 0 | — |
| 1 | 0 | 0 | 1 | Начать работу схемы |
| 1 | 0 | 1 | 0 | Сброс |
| 1 | 0 | 1 | 1 | Запрещенная комбинация |
| 1 | 1 | 0 | 0 | Добавить символ D0…D7 |
| 1 | 1 | 0 | 1 | Запрещенная комбинация |
| 1 | 1 | 1 | 0 | Запрещенная комбинация |
| 1 | 1 | 1 | 1 | Запрещенная комбинация |

Таблица 1 — Таблица соответствий входных сигналов устройства

Блок устройства начинает работу в тот момент, когда предыдущий блок выставил бит, сигнализирующий о конце работы. Все блоки взаимодействуют с ОЗУ устройства. Все блоки могут читать данные на выходах ОЗУ. Доступ к записи в память блоки имеют только в тот момент, когда они находятся в работе.

Блок управления (БУ) ОЗУ от каждого блока получает адрес ОЗУ, данные для записи, а также сигналы управления запоминающим устройством. БУ подаёт на ОЗУ сигналы, основываясь на текущих флагах состояния работы остальных блоков.

В работе участвуют 5 блоков: блок ввода (INPUT), блок сортировки (SORT), блок построения дерева (TREE), блок получения кода (CODE), блок вывода (OUTPUT). В таблице приведена таблица истинности, в соответствии с которой реализуется выбор подаваемых блоками в БУ сигналов для управления ОЗУ.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| INPUT | SORT | TREE | CODE | OUTPUT | Y3 | Y2 | Y1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

Других комбинаций флагов работы появиться не может, так как логические единицы выставляются последовательно на каждом из них в вышеуказанном порядке. INPUT всегда обозначен как 1, так как при сбросе на нем единственном не меняется бит состояния. Для того, чтобы давать возможность писать в ОЗУ 5-ю блокам, на БУ должен подаваться 3-битный адрес этого блока.

Заметим, что максимальная длина кода символа при размере последовательности до 16 и мощности алфавита до 10 символов не может быть больше 4. Таким образом можно дать оценку сверху числа светодиодов для индикации вывода — 16 × 4 = 64. Значит, индикация будет производиться на сетке светодиодов 8×8.

### Составление и расчет схемы устройства

Задачу получения принципиальной схемы устройства можно разделить на несколько частей:

1. Составление функциональной схемы блока ввода
2. Составление функциональной схемы блока сортировки
3. Составление функциональной схемы блока построения дерева
4. Составление функциональной схемы блока получения кода
5. Составление функциональной схемы блока вывода
6. Объединение составленных элементов
7. Расчет элементов микросхемы
8. Выбор реальных элементов для схемы

#### **Составление функциональной схемы обработчика входа**

Выбор регистра по заданному адресу будет осуществляться с помощью дешифратора. Номер каждого из 32 регистров можно представить в бинарном виде, максимальный номер – 31 (так как нумерация начинается с 0) – представляется пятью битами, поэтому требуется регистр 5 в 32. Его можно получить каскадированием дешифраторов 4 в 16.

Принципиальная схема обработчика входа изображена на рисунке 3.

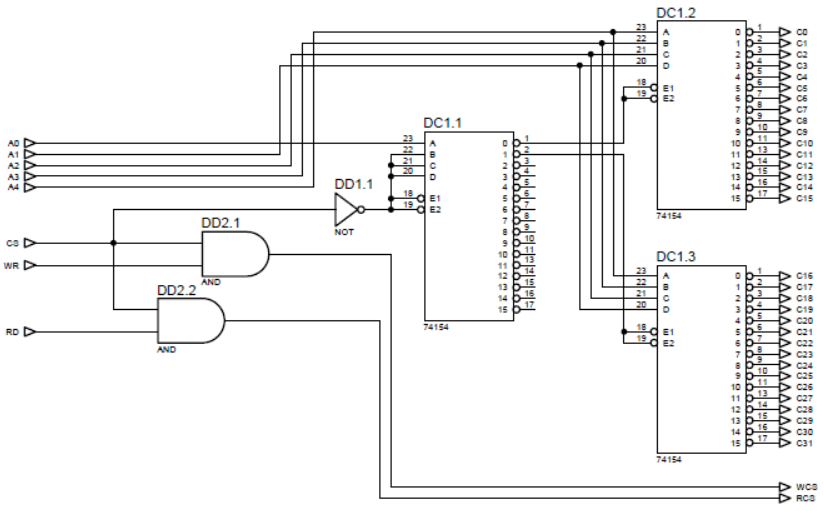


Рисунок 3 – Функциональная схема обработчика входа

На составленной схеме входы A0, …, A4 – адресные; входы CS, WR, RD – управляющие.

Управляющие сигналы WR и CS, RD и CS поступают на элементы И, выходные значения определяют, разрешена ли работа с микросхемой, разрешена ли запись или чтение. Далее конъюнкции управляющих сигналов будут обозначены WCS и RCS соответственно.

Адресные сигналы поступают на дешифратор, работа которого разрешается сигналом CS, его выходы далее будут обозначены C0, …, C31 – сигналы выбора регистра. Эти сигналы имеют инверсное значение.

Для упрощения итоговой схемы ОЗУ создан отдельный функциональный элемент, реализующий описанные выше функции, – INPUT\_HANDLER. Его графическое обозначение приведено на рисунке 4.

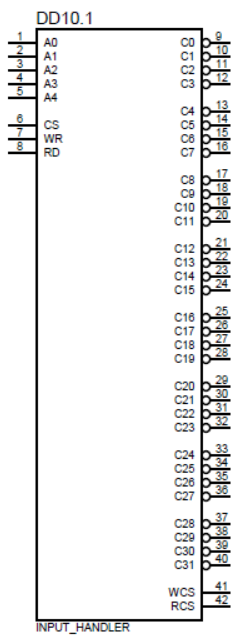


Рисунок 4 – Графическое обозначение функционального элемента INPUT\_HANDLER

#### **Составление функциональной схемы многоадресного регистра**

Единичную ячейку памяти реализует триггер. Всего существует четыре типа триггеров: RS, D, T, JK. Из них наиболее подходящий D – он имеет два входа: информационный и синхронизирующий, может использоваться для хранения одного бита информации.

Схема составленного из триггеров 8-ми битного регистра приведена на рисунке 5.

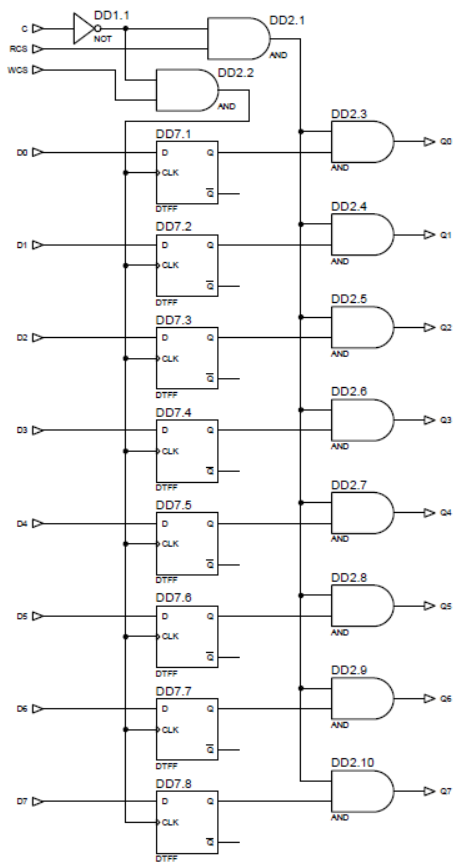


Рисунок 5 – Функциональная схема 8-ми битного регистра

На приведенной схеме входы RCS и WCS являются управляющими, C – вход сигнала выбора регистра, необходим для разрешения или запрета взаимодействия с содержимым регистра; сигналы D0, …, D7 – данные для записи значений на каждый из триггеров; сигналы Q0, …, Q7 – выходы регистра, сформированные значениями в триггерах.

Информация записывается в единичные ячейки, считывается с них, отправляется на выход только при условии соответствующих значений управляющих сигналов WCS, RCS и сигнала выбора регистра – проверка этих условий реализуется с помощью элементов И.

Для упрощения итоговой схемы регистр собирается в отдельных функциональный элемент. Его графическое обозначение приведено на рисунке 6.

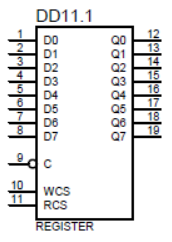


Рисунок 6 – Графическое обозначение функционального элемента REGISTER

Соединение 8 регистров по соответствующим входам, побитовая передача их выходов на элементы 8ИЛИ дает 8-адресный регистр (далее будет обозначен как 8-регистр). Схема устройства приведена на рисунке 7.

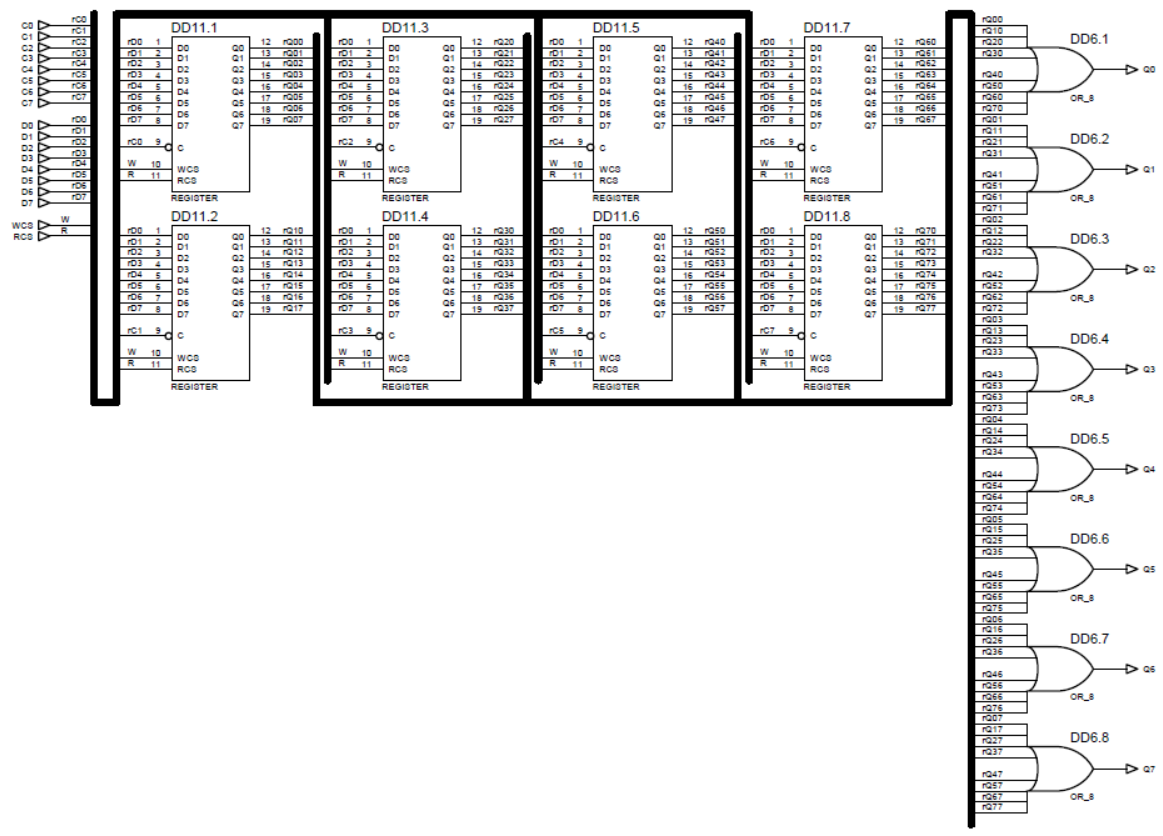


Рисунок 7 – Функциональная схема 8-регистра

Выходным значением 8-регистра будет побитовая дизъюнкция слов, записанных в выбранные регистры при условии, что чтение разрешено. Согласно устройству обработчика входа, генерирующего значения CN (где N – номер адреса – номер регистра), только один из этих сигналов может иметь значение 0, поэтому при подключении 8-регистров к обработчику входа, на выходе будет генерироваться слово, хранящееся по выбранному адресу (при условии, что чтение разрешено).

Графическое обозначение функционального элемента 8-регистр (8\_REGISTER) изображено на рисунке 8.

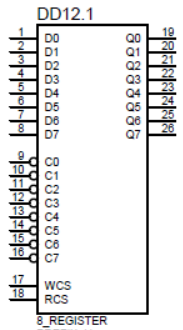


Рисунок 8 – Графическое обозначение функционального элемента 8\_REGISTER

Соединение четырех микросхем 8\_REGISTER по соответствующим входам образует хранилище памяти размера 256 бит (32 на 8) – многоадресный регистр. Схема такого подключения с использованием шин данных изображена на рисунке 9.

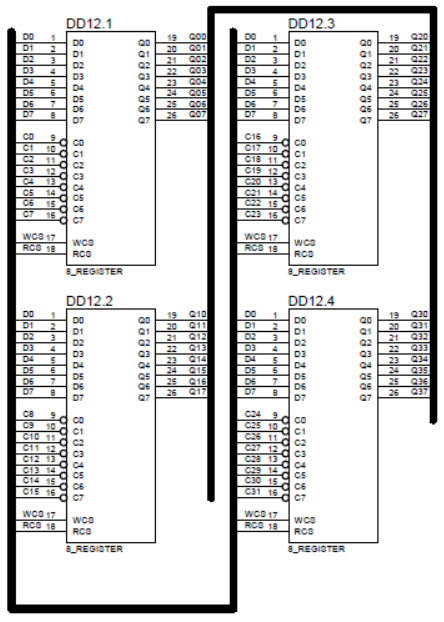


Рисунок 9 – Схема МАР, построенного на элементах 8\_REGISTER

#### **Составление функциональной схемы обработчика выхода**

Выбор определенных записанных в память значений в зависимости от управляющих сигналов осуществляются на 8-регистрах. Поэтому в МАР на выходе оказывается 32 значения: четыре восьмиразрядных слова с каждого из 8-регистров. Чтобы выходом являлось 8 бит – содержимое выбранной ячейки памяти, – нужно произвести побитовое ИЛИ для всех слов.

Таким образом обработчик выхода частично совмещен с функциональными элементами, реализующими МАР – такой подход к проектированию устройства позволяет оптимизировать и упростить схему.

#### **Реализация интерфейса ввода и вывода UART**

Задача осуществления ввода и вывода данных посредством интерфейса UART заключается в создании входа (RX) и выхода (TX), реализующих последовательную передачу данных в формате, изображенном на рисунке 10.



Рисунок 10 – Формат данных, передаваемых посредством интерфейса UART

Между посылками по каналам RX, TX передается значение логической единицы, поэтому стартовым битом является 0 – он не несет информацию, но принимающее устройство, получив стартовый бит, следующие L бит обработает как информационные – над которыми нужно произвести некоторые действия. L бит данных – передаваемая информация. Завершающий бит – 1 – значение, передаваемое между посылками.

В разрабатываемой микросхеме на вход будет поступать посылка размером 16 бит (5 адресных, 8 информационных, 3 управляющих); на выходе будет отправляться посылка с 8 битами – прочитанная из памяти информация.

Реализацию интерфейса ввода и вывода микросхемы можно разделить на две части:

1. Составление функциональной схемы преобразователя входа
2. Составление функциональной схемы преобразователя выхода

##### **Составление функциональной схемы преобразователя входа**

Принимающая линия RX должна быть подключена ко входу последовательно-параллельного регистра и к устройству, которое реагирует на смену сигнала – на начало передачи. После получения стартового бита нужно начать отсчет принятых сигналов, в течение отсчета в регистр будет проталкиваться получаемая информация. Когда будет получено 16 бит, запись данных в регистр прекратится. Информация, принятая последовательно, будет сниматься с преобразователя параллельно.

Поскольку итоговые сигналы на выходе появляются не моментально, необходимо реализовать механизм, задерживающий управляющий сигнал CS, так как в противном случае содержимое памяти может быть непроизвольно изменено. Также при получении комбинации управляющих сигналов, разрешающих чтение, на преобразователь выхода нужно отправлять сигнал о том, что нужно сформировать и отправить содержимое выбранной ячейки памяти. Обе задачи можно решить следующим образом: когда посылка принята и полученный сигнал CS равен логической единице, отправить сигнал длиной в один тактовый период на соответствующий вход обработчика входа; когда посылка принята и полученный сигнал RD равен логической единице, отправить сигнал длиной в один тактовый период на вход запуска преобразователя выхода. При такой реализации в случае, когда конъюнкция значений сигналов RD и CS равна 0, выходная посылка формироваться не будет.

Для отправки сигнала длительностью в один тактовый период используется делитель частоты, схема которого изображена на рисунке 11. Графическое обозначение функционального элемента INDICATOR, реализующего описанные выше функции, на рисунке 12.

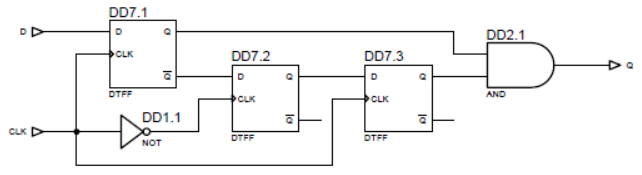


Рисунок 11 – Функциональная схема делителя частоты

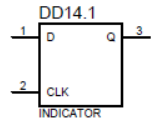


Рисунок 12 – Графическое обозначение функционального элемента INDICATOR

На вход CLK поступает тактовый сигнал, на вход D – сигнал, возбуждающий импульс, который поступит на выход Q.

Схема построенного устройства преобразования и графическое изображение функционального элемента RX\_CONVERTER приведены на рисунках 13, 14 соответственно.

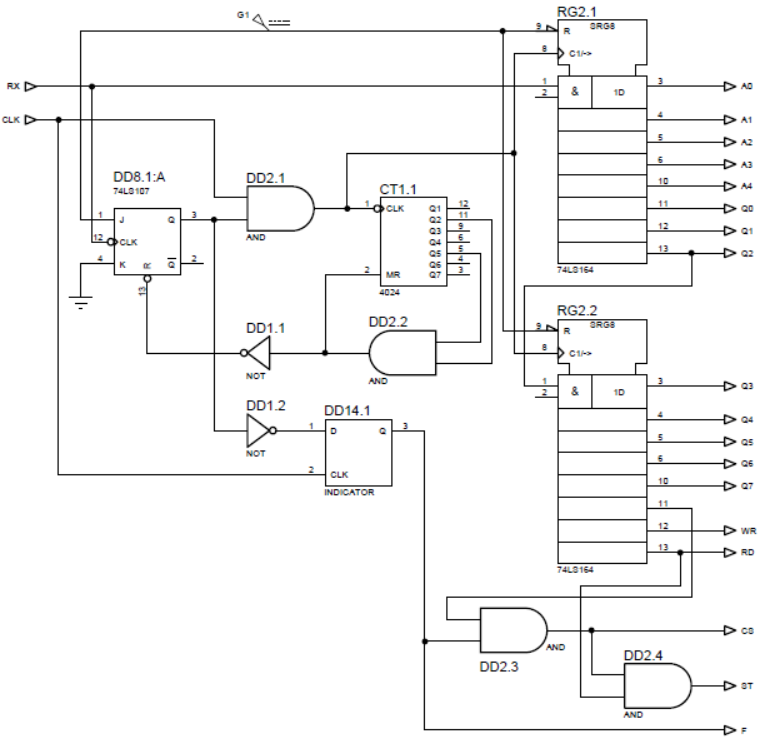


Рисунок 13 – Функциональная схема преобразователя входа

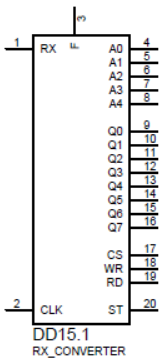


Рисунок 14 – Графическое обозначение функционального элемента RX\_CONVERTER

Устройство, реагирующее на входной бит – JK-триггер. В периоды времени, когда посылка не передается, он находится в режиме хранения информации, после приема стартового бита переходит в режим установки единицы. Значение с триггера поступает на И с тактовым сигналом CLK – полученный сигнал осуществляет проталкивание бита посылки в регистр и инкрементирует значение в счетчике. Когда значение счетчика доходит до 18, все информационные биты приняты, и триггер сбрасывается в состояние хранения информации, запись в регистр прекращается.

Частота тактирующего сигнала CLK определяет скорость передачи информации. Выходы подписаны в соответствии со своими значениями, кроме CS: если этот сигнал равен единице, то соответствующее выходное значение не будет установлено – вместо этого испустится единичный импульс; с выхода ST выходит аналогичный сигнал, если разрешено чтение из памяти; также аналогичный сигнал выходит с F после полного приема посылки вне зависимости от ее содержимого – этот сигнал нужен для обратной связи с пользователем или устройством, которое передает данные с помощью преобразователя.

##### **Составление функциональной схемы преобразователя выхода**

Для начала формирования серии выходных сигналов преобразователь выхода должен получать сигнал, по которому на выход будет отправляться логический 0, сигналы со всех параллельных входов, после чего устанавливаться единица.

Схема спроектированного устройства преобразователя представлена на рисунке 15.

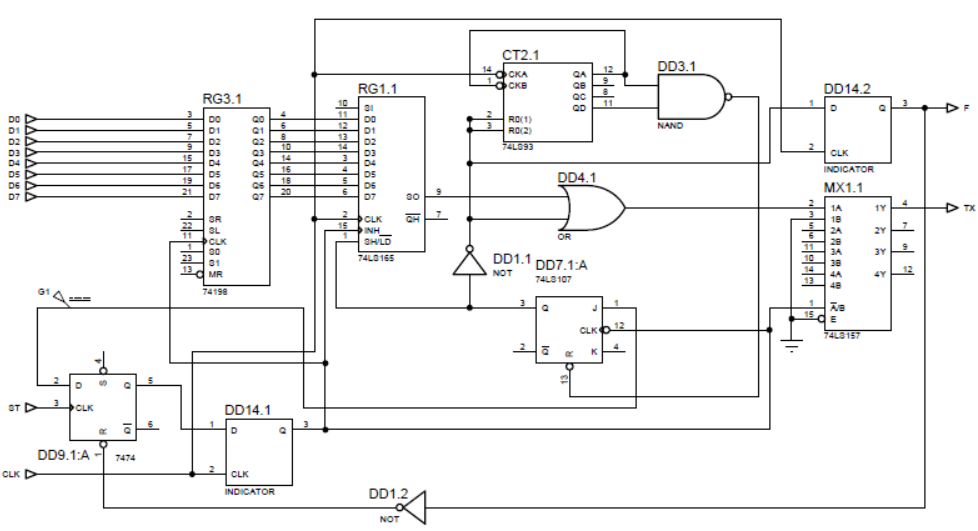


Рисунок 15 – Функциональная схема преобразователя выхода

Входные сигналы D0, …, D7 поступают на параллельный регистр для их запоминания в случае быстрой смены входных значений. Вход ST принимает запускающий сигнал, который переключит мультиплексор и выдаст на выход значение логического нуля (что означает начало передачи данных), а также запустит счетчик и чтение из параллельно-последовательного регистра. Счетчик при достижении значения 9 запретит чтение из регистра и передача сообщения закончится, на выходе установится значение логической единицы.

Сигнал ST поступает на сбрасываемый D-триггер, который запоминает значение логической единицы и передает ее на INDICATOR, после чего единичный импульс запускает процесс передачи посылки. После завершения D-триггер сбрасывается. Такая обработка входа нужна для того, чтобы в процессе отправки информации изменение сигнала ST не влияло на работу микросхемы.

Частота тактового сигнала CLK определяет скорость передачи данных. На выход F передается сигнал длиной в один тактовый период после окончания передачи посылки – он служит для обратной связи с пользователем или устройством, которое использует преобразователь.

Графическое обозначение функционального элемента TX\_CONVERTER, осуществляющего работу преобразователя параллельной передачи данных в последовательную, изображено на рисунке 16.

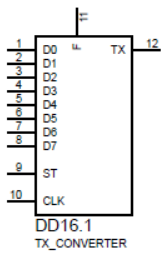


Рисунок 16 – Графическое обозначение функционального элемента TX\_CONVERTER

#### **Объединение составленных элементов**

Чтобы получить схему ОЗУ с параллельным вводом и выводом, нужно соединить построенные ранее функциональные части по соответствующим контактам.

На рисунке 17 приведена схема полученного с использованием функциональных элементов INPUT\_HANDLER, 8\_REGISTER запоминающего устройства.

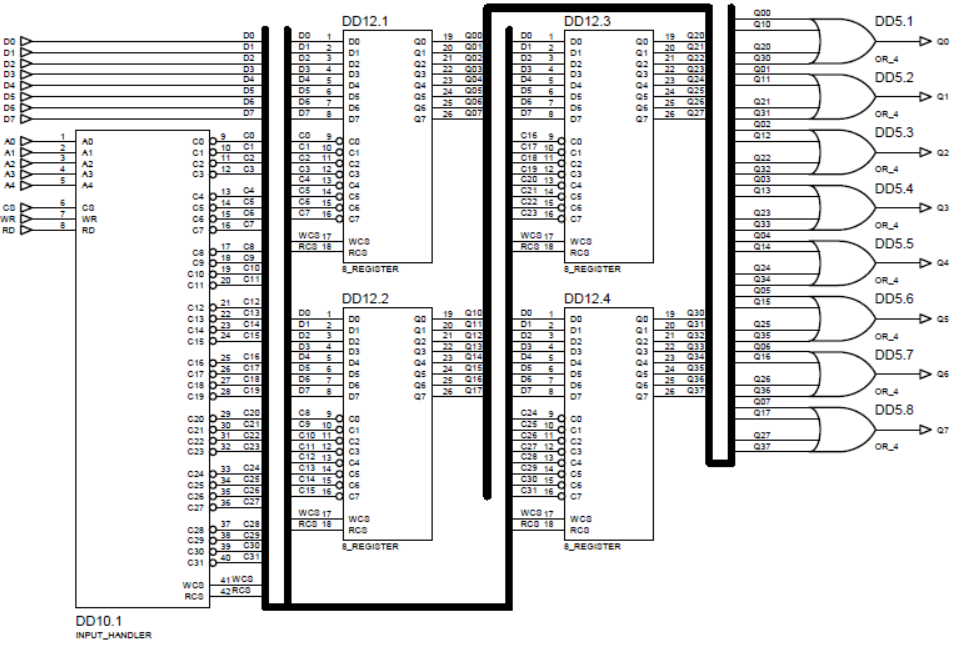


Рисунок 17 – Схема ОЗУ на функциональных элементах INPUT\_HANDLER, 8\_REGISTER

Входы микросхемы: адрес (5 бит), данные для записи (8 бит), управляющие сигналы (3 бита); выходные сигналы: прочитанные данные (8 бит).

Графическое обозначение функционального элемента RAM, выполняющее функции оперативного запоминающего устройства с параллельными входами и выходами, представлено на рисунке 18.

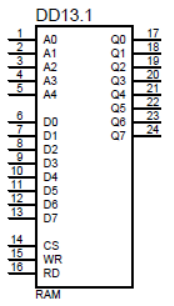


Рисунок 18 – Графическое обозначение функционального элемента RAM

Чтобы получить схему ОЗУ с последовательным вводом и последовательным выводом к параллельным входам и выходам нужно подключить преобразователь входа и выхода соответственно.

Схема итогового устройства, состоящего из компонентов RX\_CONVERTER и TX\_CONVERTER и RAM, приведена на рисунке 19.

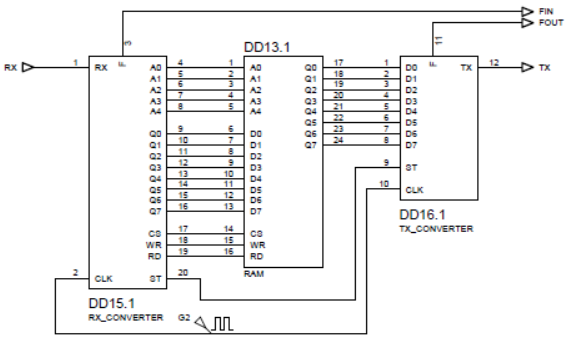


Рисунок 19 – Схема итогового устройства на функциональных элементах RAM, RX\_CONVERTER, TX\_CONVERTER

Передаваемые по каналу RX данные должны иметь определенную структуру (помимо установленной протоколом UART): A0/…/A5/D0/…/D7/CS/WR/RD – пять адресных сигналов, восемь информационных, три управляющих. При нарушении этой структуры результат работы устройства будет отличаться от ожидаемого.

Графическое обозначение функционального элемента RAM с входом RX и выходом TX изображено на рисунке 20.

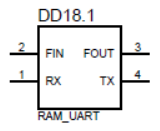


Рисунок 20 – Графическое обозначение функционального элемента RAM

#### **Расчет элементов микросхемы**

Для построения микросхемы были использованы логические элементы, простейшие запоминающие устройства, комбинационные микросхемы – компоненты, не требующие расчета; генераторы: постоянного напряжения и прямоугольных импульсов. Первый класс выполняет роль источника сигнала логической единицы, поэтому должны выдавать напряжение 5 В. Второй класс генерирует тактовые сигналы, обеспечивающие прием и передачу данных посредством интерфейса UART, частота такого генератора определяет скорость передачи информации.

По заданию скорость UART составляет 9600 бод, то есть 9600 бит за секунду. Схема построена так, что за один период тактового сигнала передается один бит, значит 9600 периодов должны занимать одну секунду – это соответствует частоте 9600 Гц. Частота генератора тактовых импульсов должна составлять 9600 Гц.

#### **Выбор реальных элементов для схемы**

Компонент 8ИЛИ не является реальным, вместо него выпускают 8ИЛИ-НЕ, поэтому все 8ИЛИ нужно заменить на последовательность 8ИЛИ-НЕ, НЕ.

Перечень использованных элементов, их реальные аналоги приведены в Приложении А. Количество требуемых компонентов указано с учетом того, что некоторые логические элементы выпускаются в одном корпусе по несколько штук.

#### **Расчет мощности устройства**

Для определения потребляемой мощности устройства необходимо вычислить сумму потребляемых мощностей для каждого из использованных элементов – эти значения можно получить из документов соответствующих реальных компонентов. В таблице 2 указаны мощности каждого элемента.

Таблица 2 – Мощности использованных компонентов

|  |  |
| --- | --- |
| Компонент | Потребляемая мощность, мВт |
| Дешифратор 4-16 | 45 |
| Семиразрядный счетчик | 500 |
| Четырехразрядный счетчик | 45 |
| Мультиплексор | 500 |
| Пар.-посл. регистр | 105 |
| Посл.-пар. регистр | 168 |
| Сдвиговый регистр | 100 |
| НЕ | 125 |
| И | 200 |
| И-НЕ | 40 |
| ИЛИ | 100 |
| 4ИЛИ | 120 |
| 8ИЛИ-НЕ | 51 |
| D-триггер | 136 |
| JK-триггер | 160 |
| D-триггер сбрасываемый | 120 |

– потребляемая мощность смоделированного устройства.

### Моделирование схемы

Для проверки корректности работы полученного устройства на вход RX передается последовательность сигналов: адресных, информационных, управляющих (с помощью преобразователя 16 параллельных входов в последовательные – TX\_CONVERTER\_16, схема и графическое обозначение которого показаны на рисунках 22 и 23 соответственно); на выходе подключен виртуальный терминал, принимающий посылки размером 8 бит – он отображает информацию из памяти в формате hex.

Будем подавать различные значения на вход с определенными комбинациями управляющих сигналов и сравнивать выходные значения с ожидаемыми. Схема, на которой происходит испытание, изображена на рисунке 21. Результаты наблюдений приведены в таблице 2.

Таблица 3 – Результаты проверки корректность работы микросхемы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | CS/WR/RD | Адрес | Вход (HEX, бинарный) | Выход (HEX) | Ожидаемый выход (HEX, бинарный) |
| 1 | 0/0/0 | 00110 | 32, 00110010 | Нет | Нет |
| 2 | 0/1/0 | 00110 | 32, 00110010 | Нет | Нет |
| 3 | 1/0/0 | 00110 | 32, 00110010 | Нет | Нет |
| 4 | 1/1/0 | 00101 | 35, 00110101 | Нет | Нет |
| 5 | 1/1/0 | 10001 | FE, 11111110 | Нет | Нет |
| 6 | 1/0/1 | 00110 | Любой | 00 | 00 |
| 7 | 1/0/1 | 00101 | Любой | 35 | 35 |
| 8 | 1/0/1 | 10001 | Любой | FE | FE |

Из проведенных экспериментов видно, что микросхема работает корректно, проектирование выполнено успешно.

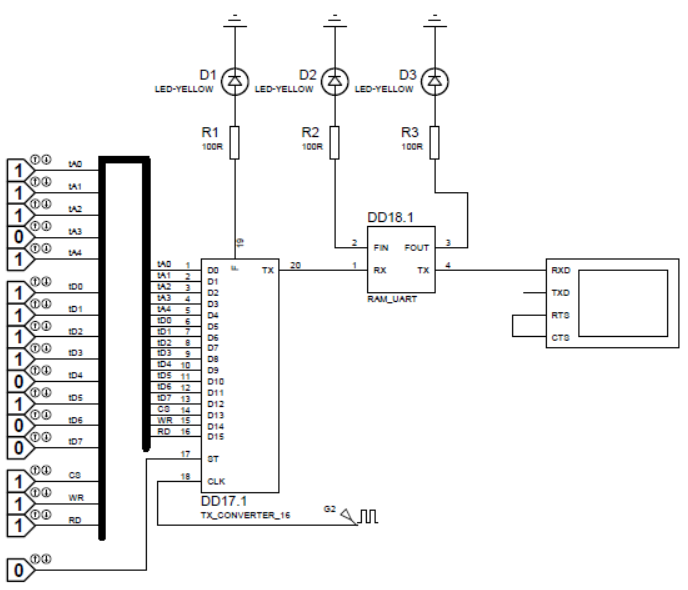


Рисунок 21 – Схема испытания спроектированного устройства

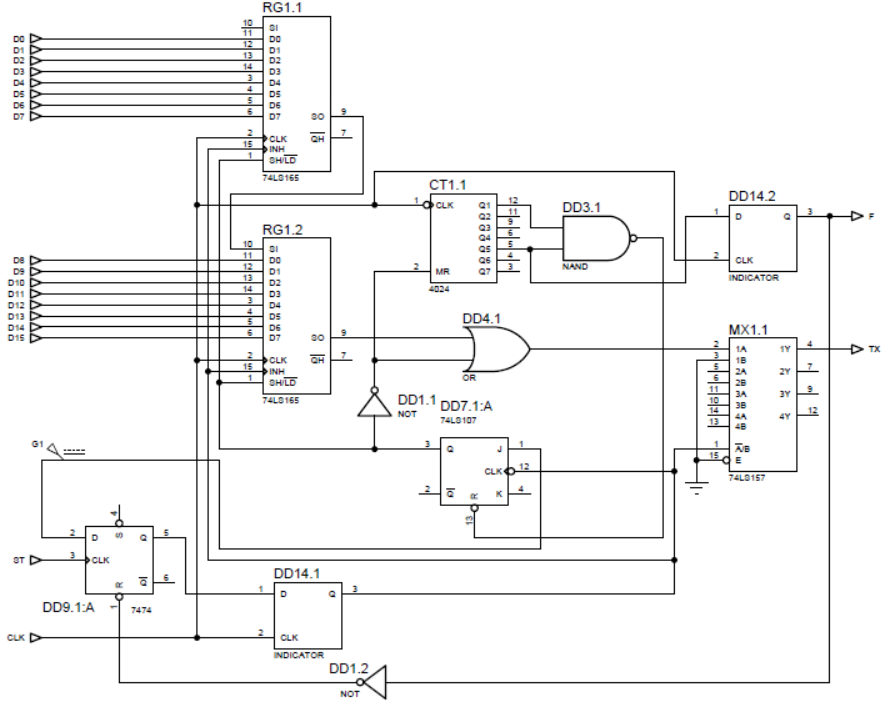


Рисунок 22 – Схема преобразователя 16 параллельных входов в последовательный выход

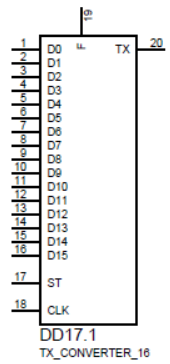


Рисунок 23 – Графическое обозначение функционального элемента TX\_CONVERTER\_16

# ЗАКЛЮЧЕНИЕ

В ходе работы было разработано устройство для кодирования последовательности ASCII-символов не длиннее 16, содержащее не более 10 уникальных символов, с помощью алгоритма Хаффмана. Полученный код индицируется на сетку светодиодов.

Потребляемая мощность схемы составляет .

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Энциклопедия о высоких технологиях / Статическая оперативная память [Электронный ресурс]. – Режим доступа: <http://all-ht.ru/inf/pc/mem_sram.html> (Дата обращения: 03.04.2023)
2. Сайт сетевого издания iXBT / Современная оперативная память [Электронный ресурс]. – Режим доступа: <https://www.ixbt.com/mainboard/ram-faq-2006.shtml> (Дата обращения: 03.04.2023)
3. Образовательный проект «Электроника для всех» / Передача данных через UART [Электронный ресурс]. – Режим доступа: <http://easyelectronics.ru/avr-uchebnyj-kurs-peredacha-dannyx-cherez-uart.html> (Дата обращения: 12.04.2023)
4. Национальный открытый университет ИНТУИТ / Лекция 8: Регистры [Электронный ресурс]. – Режим доступа: <https://intuit.ru/studies/courses/104/104/lecture/3043> (Дата обращения: 15.04.2023)
5. Национальный открытый университет ИНТУИТ / Лекция 7: Триггеры [Электронный ресурс]. – Режим доступа: <https://intuit.ru/studies/courses/104/104/lecture/3041> (Дата обращения: 15.04.2023)

# ПРИЛОЖЕНИЕ А