# Микропрограммный автомат на постоянном запоминающем устройстве для кодирования манчестерского кода

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ УКРАИНЫ

"КИЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ"

КАФЕДРА КЭВА

Курсовая работа

по курсу: "Аналоговая и цифровая электроника"

"Микропрограммный автомат на ПЗУ для кодирования манчестерского кода"

Выполнил:

студент группы ДК-71

Феськов Д.А.

Проверил:

ст. пр. Собченко В.В.

Киев – 2010

СОДЕРЖАНИЕ

Введение

1. Теоретические сведения

1.1 Триггеры и Регистры

1.2 Мультиплексоры

1.3 ПЗУ

1.4 Микропрограммный автомат на ПЗУ

1.5 Код Манчестер-II

1.6 Протокол передачи данных

2. Структурная схема устройства

3. Симуляция схемы в САПР Altera Quartus II

Выводы

Приложение

ВВЕДЕНИЕ

К настоящему времени наиболее совершенные принципы и средства взаимодействия человека с окружающим миром (технологии взаимодействия) обеспечила цифровая техника. Ее наименее избыточный алфавит – двухуровневые символы, которыми оказалось возможным представлять (кодировать) любую информация – привел к создания чрезвычайно точных, надежных, малогабаритных и функционально-наращиваемых устройств. Использование в цифровой технике двухсимвольного алфавита привело к созданию новых, исключительно эффективных методов передачи, хранения и преобразования сигналов, к новым средствам обработки информации – информационным технологиям (под этим словосочетанием понимают технологию обработки информации с использованием современных средств цифровой техники и ее вершины – вычислительной техники). Так родились основанные на новых принципах современные информационные технологии: связи (цифровая связь и цифровое телевидение), обнаружения (цифровая радиолокация и цифровая навигация), вычислений и автоматического управления (электронно-вычислительная техника), техники измерений и т.д.

Цифровая техника стоит на трех "китах". Первый "кит" - теорема о дискретизации. В этой теореме теоретически обоснована возможность получения цифрового эквивалента (цифрового образа) аналогового сигнала, хранить предавать и обрабатывать который оказалось значительно проще и точнее, чем осуществлять аналогичные действия над аналоговым сигналом. Второй "кит" - алгебра логики (булева алгебра). Алгебра логики позволила поставить анализ и синтез цифровых схем на прочных математический фундамент. Третий "кит" - импульсная техника, из которой цифровая техника заимствовала многие принципы, элементы и устройства.

Цифровые устройства обладают рядом преимуществ перед аналоговыми: огромная степень интеграции, составляющая сотни миллионов транзисторов в одной микросхеме, чрезвычайно низкая погрешность, малая зависимость от параметров окружающей среды.

Области применения цифровой техники поистине безграничны. К сказанному ранее можно добавить, что в настоящее время до 90 % всех разрабатываемых устройств – цифровые.

В данной работе предлагается к рассмотрению способ передачи данных на большие расстояния

Код Манчестер-II широко используется при передаче сигналов на большие расстояния, в частности, в локальных сетях.

1.                ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Триггеры и регистры

Триггеры и регистры являются простейшими представителями цифровых микросхем, имеющих внутреннюю память. Если выходные сигналы логических элементов и комбинационных микросхем однозначно определяются их текущими входными сигналами, то выходные сигналы микросхем с внутренней памятью зависят также еще и от того, какие входные сигналы и в какой последовательности поступали на них в прошлом, то есть они помнят предысторию поведения схемы. Именно поэтому их применение позволяет строить гораздо более сложные и интеллектуальные цифровые устройства, чем в случае простейших микросхем без памяти. Микросхемы с внутренней памятью называются еще последовательными или последовательностными, в отличие от комбинационных микросхем.

Триггеры и регистры сохраняют свою память только до тех пор, пока на них подается напряжение питания. Иначе говоря, их память относится к типу оперативной (в отличие от постоянной памяти и перепрограммируемой постоянной памяти, которым отключение питания не мешает сохранять информацию). После выключения питания и его последующего включения триггеры и регистры переходят в случайное состояние, то есть их выходные сигналы могут устанавливаться как в уровень логической единицы, так и в уровень логического нуля. Это необходимо учитывать при проектировании схем.

Большим преимуществом триггеров и регистров перед другими типами микросхем с памятью является их максимально высокое быстродействие (то есть минимальные времена задержек срабатывания и максимально высокая допустимая рабочая частота). Именно поэтому триггеры и регистры иногда называют также сверхоперативной памятью. Однако недостаток триггеров и регистров в том, что объем их внутренней памяти очень мал, они могут хранить только отдельные сигналы, биты (триггеры) или отдельные коды, байты, слова (регистры).

Триггер можно рассматривать как одноразрядную, а регистр — как многоразрядную ячейку памяти, которая состоит из нескольких триггеров, соединенных параллельно (обычный, параллельный регистр) или последовательно (сдвиговый регистр или, что то же самое, регистр сдвига).

В основе любого триггера (англ. — "тrigger" или "flip-flop") лежит схема из двух логических элементов, которые охвачены положительными обратными связями (то есть сигналы с выходов подаются на входы). В результате подобного включения схема может находиться в одном из двух устойчивых состояний, причем находиться сколь угодно долго, пока на нее подано напряжение питания.

Пример такой схемы (так называемой триггерной ячейки) на двух двухвходовых элементах И-НЕ представлен на рисунке 1.1.1. У схемы есть два инверсных входа: –R — сброс (от английского Reset), и –S — установка (от английского Set), а также два выхода: прямой выход Q и инверсный выход –Q.

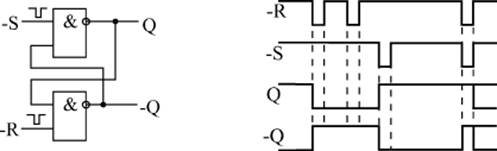


Рисунок 1.1.1 – Схема триггерной ячейки

Для правильной работы схемы отрицательные импульсы должны поступать на ее входы не одновременно. Приход импульса на вход -R переводит выход -Q в состояние единицы, а так как сигнал -S при этом единичный, выход Q становится нулевым. Этот же сигнал Q поступает по цепи обратной связи на вход нижнего элемента. Поэтому даже после окончания импульса на входе -R состояние схемы не изменяется (на Q остается нуль, на -Q остается единица). Точно так же при приходе импульса на вход -S выход Q в единицу, а выход -Q — в нуль. Оба эти устойчивых состояния триггерной ячейки могут сохраняться сколь угодно долго, пока не придет очередной входной импульс, — иными словами, схема обладает памятью.

Если оба входных импульса придут строго одновременно, то в момент действия этих импульсов на обоих выходах будут единичные сигналы, а после окончания входных импульсов выходы случайным образом попадут в одно из двух устойчивых состояний. Точно так же случайным образом будет выбрано одно из двух устойчивых состояний триггерной ячейки при включении питания. Временная диаграмма работы триггерной ячейки показана на рисунке 1.1.1.

В стандартные серии цифровых микросхем входит несколько типов микросхем триггеров, различающихся методами управления, а также входными и выходными сигналами. На схемах триггеры обозначаются буквой Т. В отечественных сериях микросхем триггеры в зависимости от типа имеют наименование ТР (RS-триггер), ТВ (JK-триггер) и ТМ (D-триггер).

Самый распространенный D-триггер занимает, можно сказать, промежуточное положение между RS-триггером и JK-триггером. Помимо общих для всех триггеров входов установки и сброса –S и –R, он имеет один информационный вход D (вход данных) и один тактовый вход C. Примером может служить показанная на рисунке 1.1.2 микросхема ТМ2, содержащая в одном корпусе два D-триггера с прямыми и инверсными выходами. Таблица истинности D-триггера ТМ2 приведена в таблице 1.1.1.

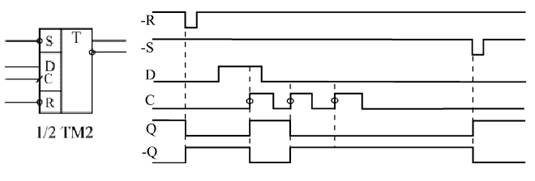


Рисунок 1.1.2 – УГО и временные диаграммы D-триггера

Таблица 1.1.1 – Таблица истинности D-триггера

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| -S | -R | C | D |  | - |
| 0 | 1 | Ф | Ф | 1 | 0 |
| 1 | 0 | Ф | Ф | 0 | 1 |
| 0 | 0 | Ф | Ф | Ф | Ф |
| 1 | 1 |  | 1 | 1 | 0 |
| 1 | 1 |  | 0 | 0 | 1 |
| 1 | 1 | 0 | Ф | Q | Q |
| 1 | 1 | 1 | Ф | Q | Q |
| 1 | 1 | 1→0 | Ф | Q | Q |

Тактируется триггер (то есть меняет свое состояние) по положительному фронту сигнала С (по его переходу из нуля в единицу) в зависимости от состояния входа данных D.

Регистры (англ. register) представляют собой, по сути, несколько D-триггеров (обычно от 4 до 16), соединенных между собой тем или иным способом. Поэтому принципиальной разницы между ними и отдельными D-триггерами не существует. Правда, триггеры, входящие в состав регистров, не имеют такого количества разнообразных управляющих входов, как одиночные триггеры.

На схемах регистры обозначаются буквами RG. В отечественных сериях микросхем регистрам соответствуют буквы ИР. Все регистры делятся на две большие группы: параллельные регистры; регистры сдвига (или сдвиговые регистры).

Существуют регистры и других типов, но они применяются гораздо реже, чем параллельные и сдвиговые, так как имеют узкоспециальное назначение.

В параллельных регистрах каждый из триггеров имеет свой независимый информационный вход (D) и свой независимый информационный выход. Тактовые входы (С) всех триггеров соединены между собой. В результате параллельный регистр представляет собой многоразрядный, многовходовый триггер.

В сдвиговых регистрах все триггеры соединены в последовательную цепочку (выход каждого предыдущего триггера соединен со входом D следующего триггера). Тактовые входы всех триггеров (С) объединены между собой. В результате такой триггер может рассматриваться как линия задержки, входной сигнал которой последовательно перезаписывается из триггера в триггер по фронту тактового сигнала С. Информационные входы и выходы триггеров могут быть выведены наружу, а могут и не выводиться - в зависимости от функции, выполняемой регистром.

Параллельные регистры, в свою очередь, делятся на две группы: регистры, срабатывающие по фронту управляющего сигнала С (или тактируемые регистры); регистры, срабатывающие по уровню управляющего сигнала С (или стробируемые регистры).

Чаще всего в цифровых схемах используются регистры, управляемые фронтом (то есть тактируемые), однако и стробируемые регистры имеют свой круг задач, в которых их ничто не может заменить.

Принцип действия регистров, срабатывающих по фронту тактового сигнала, ничем не отличается от принципа действия D-триггера. По положительному фронту тактового сигнала С каждый из выходов регистра устанавливается в тот уровень, который был в этот момент на соответствующем данному выходу входе D, и сохраняется таковым до прихода следующего положительного фронта сигнала С. То есть если триггер запоминает один сигнал (один двоичный разряд, один бит), то регистр запоминает сразу несколько (4, 6, 8, 16) сигналов (несколько разрядов, битов).

В стандартные серии входит несколько типов параллельных регистров, срабатывающих по фронту (Рисунок 1.1.3). Различаются они количеством разрядов, наличием или отсутствием инверсных выходов, наличием или отсутствием входа сброса (–R) или разрешения записи (–WE), а также типом выходных каскадов (2С или 3С) и, соответственно, наличием или отсутствием входа разрешения –EZ. Иногда на схемах тактовый вход С обозначается WR - сигнал записи в регистр.

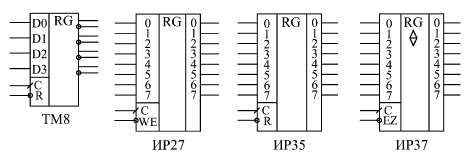


Рисунок 1.1.3 - Параллельные регистры стандартных серий, срабатывающие по фронту

Большинство регистров имеют восемь разрядов, то есть запоминают один байт информации.

Таблицы истинности регистров очень просты и не отличаются принципиально от таблицы истинности D-триггеров. Отличие от триггеров появляется только в случае наличия у регистра дополнительных управляющих входов разрешения записи –WE и разрешения выхода –EZ. В качестве примера в таблице 1.1.3 приведена таблица истинности регистра ИР27.

Таблица 1.1.2 – Таблица истинности параллельного регистра ИР27

|  |  |  |  |
| --- | --- | --- | --- |
| -WE | C | D |  |
| 0 |  | 0 | 0 |
| 0 |  | 1 | 1 |
| 0 | 0 | Ф | Q |
| 0 | 1 | Ф | Q |
| 0 |  | Ф | Q |

Одно из основных применений регистров состоит в хранении требуемого кода в течение нужного времени. Если для работы остальной части схемы необходимо иметь входной код, который можно легко изменять, то для этого как раз подходит регистр.

На рис. 8.3 показана типичная схема включения регистра для хранения кода и временная диаграмма его работы.



Рисунок 1.1.4 - Хранение кода в параллельном регистре

Код на входе регистра может изменяться произвольным образом, но в тот момент, когда этот код принимает необходимое значение, на вход С триггера подается синхросигнал (строб), который записывает код в регистр. Этот код будет храниться в регистре до прихода следующего строба. Причем важно и то, что все разряды выходного кода регистра будут переключаться одновременно даже в том случае, когда разряды входного кода переключаются не одновременно. Главное, чтобы к приходу положительного фронта строба (сигнала С) все разряды входного кода приняли нужное, устойчивое значение.

1.2 Мультиплексоры

Мультиплексоры (английское Multiplexer) предназначены для поочередной передачи на один выход одного из нескольких входных сигналов, то есть для их мультиплексирования. Количество мультиплексируемых входов называется количеством каналов мультиплексора, а количество выходов называется числом разрядов мультиплексора. Управление работой мультиплексора (выбор номера канала) осуществляется с помощью входного кода адреса.

Мультиплексоры бывают с выходом 2С и с выходом 3С. Выходы мультиплексоров бывают прямыми и инверсными. Выход 3С позволяет объединять выходы мультиплексоров с выходами других микросхем, а также получать двунаправленные и мультиплексированные линии. Некоторые микросхемы мультиплексоров имеют вход разрешения/запрета С (другое обозначение — S), который при запрете устанавливает прямой выход в нулевой уровень.

На рисунке 1.2.1 показаны для примера несколько микросхем мультиплексоров из состава стандартных серий. В отечественных сериях мультиплексоры имеют код типа микросхемы КП. На схемах микросхемы мультиплексоров обозначаются буквами MS.

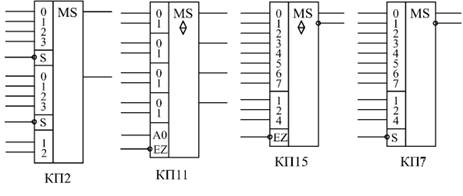


Рисунок 1.2.1 - Примеры микросхем мультиплексоров

В таблице 1.2.1 в качестве примера приведена таблица истинности одноразрядного 8-канального мультиплексора КП7.

Таблица 1.2.1 – Таблица истинности 8-канального мультиплексора КП7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 2 | 1 | -S | Q | -Q |
| Ф | Ф | Ф | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | D0 | -D0 |
| 0 | 0 | 1 | 0 | D1 | -D1 |
| 0 | 1 | 0 | 0 | D2 | -D2 |
| 0 | 1 | 1 | 0 | D3 | -D3 |
| 1 | 0 | 0 | 0 | D4 | -D4 |
| 1 | 0 | 1 | 0 | D5 | -D5 |
| 1 | 1 | 0 | 0 | D6 | -D6 |
| 1 | 1 | 1 | 0 | D7 | -D7 |

Задержки выходного сигнала мультиплексора по входам управляющего (адресного) кода примерно в два раза превышают задержки логических элементов, а по информационным входам — примерно в полтора раза. Точные величины задержек надо смотреть в справочниках.

1.3 ПЗУ

Память — это всегда очень сложная структура, включающая в себя множество элементов. Правда, внутренняя структура памяти — регулярная, большинство элементов одинаковые, связи между элементами сравнительно простые, поэтому функции, выполняемые микросхемами памяти, не слишком сложные.

Память предназначена для запоминания, хранения каких-то массивов информации, проще говоря, наборов, таблиц, групп цифровых кодов. Каждый код хранится в отдельном элементе памяти, называемом ячейкой памяти. Основная функция любой памяти как раз и состоит в выдаче этих кодов на выходы микросхемы по внешнему запросу. А основной параметр памяти — это ее объем, то есть количество кодов, которые могут в ней храниться, и разрядность этих кодов.

Принцип организации памяти записывается следующим образом: сначала пишется количество ячеек, а затем через знак умножения (косой крест) — разрядность кода, хранящегося в одной ячейке. Например, организация памяти 64Кх8 означает, что память имеет 64К (то есть 65536) ячеек и каждая ячейка — восьмиразрядная.

В зависимости от способа занесения (записи) информации и от способа ее хранения, микросхемы памяти разделяются на следующие основные типы:

-                   Постоянная память (ПЗУ — постоянное запоминающее устройство, ROM — Read Only Memory — память только для чтения), в которую информация заносится один раз на этапе изготовления микросхемы. Такая память называется еще масочным ПЗУ. Информация в памяти не пропадает при выключении ее питания, поэтому ее еще называют энергонезависимой памятью.

-                   Программируемая постоянная память (ППЗУ — программируемое ПЗУ, PROM — Programmable ROM), в которую информация может заноситься пользователем с помощью специальных методов (ограниченное число раз). Информация в ППЗУ тоже не пропадает при выключении ее питания, то есть она также энергонезависимая.

-                   Оперативная память (ОЗУ — оперативное запоминающее устройство, RAM — Random Access Memory — память с произвольным доступом), запись информации в которую наиболее проста и может производиться пользователем сколько угодно раз на протяжении всего срока службы микросхемы. Информация в памяти пропадает при выключении ее питания.

Существует множество промежуточных типов памяти, а также множество подтипов, но указанные — самые главные, принципиально отличающиеся друг от друга. Хотя, разница между ПЗУ и ППЗУ с точки зрения разработчика цифровых устройств, как правило, не так уж велика. Только в отдельных случаях, например, при использовании так называемой флэш-памяти (flash-memory), представляющей собой ППЗУ с многократным электрическим стиранием и перезаписью информации, эта разница действительно чрезвычайно важна. Можно считать, что флэш-память занимает промежуточное положение между ОЗУ и ПЗУ.

В общем случае любая микросхема памяти имеет следующие информационные выводы:

-                   Адресные выводы (входные), образующие шину адреса памяти. Код на адресных линиях представляет собой двоичный номер ячейки памяти, к которой происходит обращение в данный момент. Количество адресных разрядов определяет количество ячеек памяти: при количестве адресных разрядов n количество ячеек памяти равно 2n.

-                   Выводы данных (выходные), образующие шину данных памяти. Код на линиях данных представляет собой содержимое той ячейки памяти, к которой производится обращение в данный момент. Количество разрядов данных определяет количество разрядов всех ячеек памяти (обычно оно бывает равным 1, 4, 8, 16). Как правило, выходы данных имеют тип выходного каскада ОК или 3С.

-                   В случае оперативной памяти, помимо выходной шины данных, может быть еще и отдельная входная шина данных, на которую подается код, записываемый в выбранную ячейку памяти. Другой возможный вариант — совмещение входной и выходной шин данных, то есть двунаправленная шина, направление передачи информации по которой определяется управляющими сигналами. Двунаправленная шина применяется обычно при количестве разрядов шины данных 4 или более.

-                   Управляющие выводы (входные), которые определяют режим работы микросхемы. В большинстве случаев у памяти имеется вход выбора микросхемы CS (их может быть несколько, объединенных по функции И). У оперативной памяти также обязательно есть вход записи WR, активный уровень сигнала на котором переводит микросхему в режим записи.

ППЗУ делятся на репрограммируемые или перепрограммируемые ПЗУ (РПЗУ, EPROM — Erasable Programmable ROM), то есть допускающие стирание и перезапись информации, и однократно программируемые ПЗУ. В свою очередь, РПЗУ делятся на ПЗУ, информация в которых стирается электрическими сигналами (EEPROM — Electrically Erasable Programmable ROM), и на ПЗУ, информация в которых стирается ультрафиолетовым излучением через специальное прозрачное окошко в корпусе микросхемы (собственно EPROM — Erasable Programmable ROM). Запись информации в любые ППЗУ производится с помощью подачи определенных последовательностей электрических сигналов (как правило, повышенного напряжения) на выводы микросхемы.

Различаются микросхемы постоянной памяти своим объемом (от 32 байт до 8 Мбайт и более), организацией (обычно количество разрядов данных бывает 4, 8 или 16), способами управления (назначением управляющих сигналов), типами выходных каскадов (обычно ОК или 3С), быстродействием (обычно задержка составляет от единиц до сотен наносекунд). Но суть всех микросхем ПЗУ остается одной и той же: имеется шина адреса, на которую надо подавать код адреса нужной ячейки памяти, имеется шина данных, на которую выдается код, записанный в адресуемой ячейке, и имеются входы управления, которые разрешают или запрещают выдачу информации из адресуемой ячейки на шину данных.

На рис. 1.3.1 представлены для примера несколько простейших и типичных микросхем постоянной памяти.

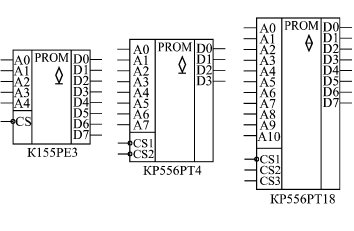


Рисунок 1.3.1 - Примеры микросхем ППЗУ отечественного производства

Микросхема К155РЕ3 (аналог — N8223N) представляет собой однократно программируемое ППЗУ с организацией . Исходное состояние (до программирования) — все биты всех ячеек нулевые. Для программирования (записи информации) используется специальный программатор, подающий на разряды данных импульсы высокого напряжения. Тип выходных каскадов — открытый коллектор, то есть обязательно надо включать на выходах резисторы, подсоединенные к шине питания. Имеется один управляющий вход –CS, при положительном уровне сигнала на котором на всех выходах устанавливаются единицы.



Основные временные характеристики микросхем ПЗУ — это две величины задержки. Задержка выборки адреса памяти — время от установки входного кода адреса до установки выходного кода данных. Задержка выборки микросхемы — время от установки активного разрешающего управляющего сигнала CS до установки выходного кода данных памяти. Задержка выборки микросхемы обычно в несколько раз меньше задержки выборки адреса.

Содержимое ПЗУ обычно изображается в виде специальной таблицы, называемой картой прошивки памяти. В таблице показывается содержимое всех ячеек памяти, причем в каждой строке записывается содержимое 16 (или 32) последовательно идущих (при нарастании кода адреса) ячеек. При этом, как правило, используется 16-ричное кодирование.

1.4 Микропрограммный автомат на ПЗУ

На основе микропрограммных автоматов можно строить устройства, которые работают по довольно сложным алгоритмам, выполняют различные функции, определяемые входными сигналами, выдают сложные последовательности выходных сигналов. При этом алгоритм работы микропрограммного автомата может быть легко изменен заменой прошивки ПЗУ.

В отличие от устройств на "жесткой" логике, принцип работы которых однозначно определяется используемыми элементами и способом их соединения, микропрограммные автоматы с помощью одной и той же схемы могут выполнять самые разные функции. То есть они гораздо более гибкие, чем схемы на "жесткой" логике. К тому же проектировать микропрограммные автоматы с точки зрения схемотехники довольно просто. Недостатком любого микропрограммного автомата по сравнению со схемами на "жесткой" логике является меньшее предельное быстродействие и необходимость составления карты прошивки ПЗУ с микропрограммами, часто довольно сложными.

Наиболее распространенная структура микропрограммного автомата (Рисунок 1.4.1) включает в себя всего лишь три элемента: ПЗУ, регистр, срабатывающий по фронту, и тактовый генератор.

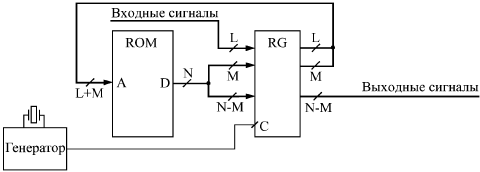


Рисунок 1.4.1 - Структура микропрограммного автомата

ПЗУ имеет () адресных разрядов и N разрядов данных. Регистр применяется с количеством разрядов (). Разряды данных ПЗУ записываются в регистр по положительному фронту тактового сигнала с генератора. Часть этих разрядов (М) используется для образования адреса ПЗУ, другая часть () служит для формирования выходных сигналов. Входные сигналы (L) поступают на входы регистра и используются совместно с частью выходных разрядов ПЗУ для получения адреса ПЗУ.



Схема работает следующим образом. В каждом такте ПЗУ выдает код данных, тем самым определяя не только состояние выходных сигналов схемы, но и адрес ПЗУ, который установится в следующем такте (после следующего положительного фронта тактового сигнала). На этот следующий адрес влияют также и входные сигналы. То есть в отличие от формирователя последовательности сигналов, рассмотренного в предыдущем разделе, в данном случае адреса могут перебираться не только последовательно (с помощью счетчика), но и в произвольном порядке, который определяется прошивкой ПЗУ, называемой микропрограммой.

Для обеспечения правильной работы необходимо обеспечить, чтобы за один период тактового сигнала должны успеть сработать регистр и ПЗУ. Иначе говоря, сумма задержки регистра и задержки выборки адреса ПЗУ не должна превышать периода тактового сигнала.

1.5 Код Манчестер-II

Код Манчестер-II широко используется при передаче сигналов на большие расстояния, в частности, в локальных сетях.

Суть манчестерского кодирования иллюстрируется рисунком 1.5.1. Входной сигнал представляет собой последовательность бит равной длительности. В каждом такте передается один бит информации. Манчестерский код заменяет единичный информационный бит на отрицательный переход в центре битового интервала, а нулевой информационный бит — на положительный переход в центре битового интервала. Таким образом, в центре каждого битового интервала сигнала в манчестерском коде обязательно имеется фронт (положительный или отрицательный), который может быть использован приемником этого сигнала для синхронизации приема каждого информационного бита. Поэтому манчестерский код называется самосинхронизирующимся кодом.

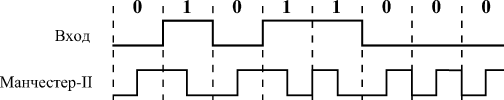


Рисунок 1.5.1 - Манчестерское кодирование

1.6 Протокол передачи данных

Набор правил, по которому выполняется передача информации между двумя устройствами, называется протоколом передачи данных. Грамотно разработанный протокол передачи данных может существенно упростить устройство. Для правильного декодирования манчестерского кода протокол должен содержать следующую информацию о передаваемых данных: частота (или период) манчестерского кода; количество передаваемых данных (битов) за один цикл передачи; направление передачи данных (от младших разрядов входного параллельного кода к старшим или наоборот); бит синхронизации (необходим для активации работы декодера, поскольку последний находится в состоянии ожидания к моменту получения байта информации).

Разработанный протокол передачи данных представлен в таблице 1.6.1.

Таблица 1.6.1 – Протокол передачи данных

|  |  |
| --- | --- |
| Параметр | Значение |
| Частота манчестерского кода | 1 Гц |
| Размер передаваемого пакета данных | 8 Бит |
| Направление передачи данных | От младших разрядов к старшим |
| Бит синхронизации | "0", младший разряд передаваемого кода |

2 СТРУКТУРНАЯ СХЕМА УСТРОЙСТВА

Структурная схема разрабатываемого устройства представлена на рисунке 2.1.

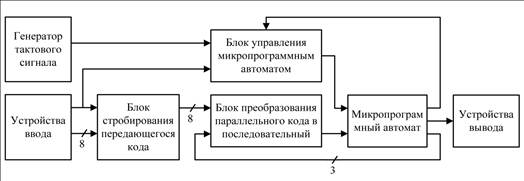


Рисунок 2.1 – Структура кодировщика манчестерского кода

Генератор тактового сигнала представляет собой устройство, генерирующее управляющий сигнал (CLK), который определяет момент выполнения элементом или узлом его функции. Пример реализации тактового генератора с использованием двух инверторов (в нестандартном включении) представлен на рисунке 2.2.

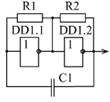


Рисунок 2.2 – Пример реализации тактового генератора на инверторах

Период тактового сигнала на выходе генератора определяется постоянной времени RC-цепочки, равной произведению сопротивления одного из резисторов () на емкость конденсатора . Так как необходимо, чтобы устройство работало на частоте 1 Гц, то емкость конденсатора , например, может быть равна 2мФ, а значения сопротивлений резисторов R1 и R2 по 500 Ом. В данном устройстве тактовый генератор в основном используется для синхронизации работы микропрограммного автомата, который собственно и выполняет кодирование входного кода.



Устройства ввода представляют собой набор DIP-переключателей ВДМ1-8, при помощи которых формируется передающийся восьмиразрядный код и тактильную кнопку ТС-0403, которая запускает цикл передачи устройством одного байта информации. После завершения передачи данных устройство переходит в состояние ожидания инициализации следующего цикла передачи данных (следующего нажатия тактильной кнопки).

Блок стробирования передающегося кода представляет собой параллельный регистр К555ИР27, срабатывающий по фронту. По нажатию тактильной кнопки код с DIP-переключателей защелкивается в регистр. Таким образом, есть возможность формировать следующий байт информации на входе устройства, еще до завершения передачи текущего байта информации.

Манчестерское кодирование (Подраздел 1.5) подразумевает представление входного (передающегося) кода в последовательном виде. Для выполнения этой функции предназначен блок преобразования параллельного кода в последовательный, который реализован в виде восьмиканального мультиплексора К155КП7. Управление работой мультиплексора (выбор номера канала) осуществляется с помощью входного кода адреса. Входной код адреса формирует микропрограммный автомат (последний выполняет перебор адресов мультиплексора от 0 до 7).

Микропрограммный автомат (вместе с мультиплексором, которым он управляет) представлен на рисунке 2.3. Основной задачей микропрограммного автомата является кодирование входного последовательного кода в манчестерский.

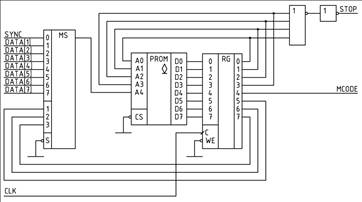


Рисунок 2.3 – Микропрограммный автомат для кодирования Манчестер-II

Как видно из рисунка 2.3 регистр, входящий в состав микропрограммного автомата стробирует данные по положительному фронту сигнала CLK. Таким образом, сигнал CLK управляет работой микропрограммного автомата (если CLK не изменяется во времени, то автомат находится в ждущем состоянии). Когда цикл передачи данных завершен, автомат взводит флаг STOP (активный уровень "0").

Для управления работой автомата (Рисунок 2.3) разработан блок управления микропрограммным автоматом (Рисунок 2.4), который управляет прохождением сигнала CLK с его входа на выход (при подаче на вход сигнала START сигнал CLK проходит на выход; при подаче STOP – на выходе "0").

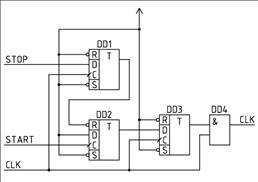


Рисунок 2.4 – Блок управления микропрограммным автоматом

Микропрограмма, выполняющая кодирование входного сигнала в манчестерский код и управление мультиплексором представлена в таблице 2.2. Микропрограмма, представленная в таблице 2.2 достаточно проста, потому подробно рассматриваться не будет.

Таблица 2.2 – Микропрограмма кодировщика манчестерского кода

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адрес ПЗУ | | | | | Данные ПЗУ | | | | | | | | Комментарий |
| 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Вх. | Адрес | | | | MS | | | Вых. | Сл. адрес | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Кодирование SYNC бита |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | Кодирование 1-ого ("0") разряда входного кода |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | Кодирование 2-ого ("0") разряда входного кода |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | Кодирование 3-его ("0") разряда входного кода |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | Кодирование 4-ого ("0") разряда входного кода |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | Кодирование 5-ого ("0") разряда входного кода |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | Кодирование 6-ого ("0") разряда входного кода |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | Кодирование 7-ого ("0") разряда входного кода |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | Кодирование SYNC бита (исключено протоколом) |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | Кодирование 1-ого ("1") разряда входного кода |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | Кодирование 2-ого ("1") разряда входного кода |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | Кодирование 3-его ("1") разряда входного кода |
| 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | Кодирование 4-ого ("1") разряда входного кода |
| 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | Кодирование 5-ого ("1") разряда входного кода |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | Кодирование 6-ого ("1") разряда входного кода |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | Кодирование 7-ого ("1") разряда входного кода |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Необходимо только отметить, что поскольку входной бит кодируется фронтом выходного сигнала, то для формирования последнего необходимо два такта работы микропрограммного автомата. Поэтому инкрементация адреса мультиплексора происходит каждый второй такт микропрограммного автомата. Когда младшие 4 разряда данных ПЗУ равны нулю флаг STOP взводится и работа автомата прекращается (блок управления микропрограммного автомата не пропускает тактовый сигнал). Поскольку один из восьми битов пакета передаваемых данных используется для синхронизации, то

В качестве устройства вывода используется светоизлучающий диод АЛ307Г-М.

Таким образом, выше были прокомментированы все основные узлы (по отдельности) спроектированного кодировщика манчестерского кода. Полная принципиальная схема кодировщика представлена в приложении.

3 СИМУЛЯЦИЯ СХЕМЫ В САПР ALTERA QUARTUS II

Схема , реализующая микропрограммный автомат на ПЗУ для кодирования манчестерского кода представлена на рисунке 3.1. Карта прошивки ПЗУ представлена на рисунке 3.2.

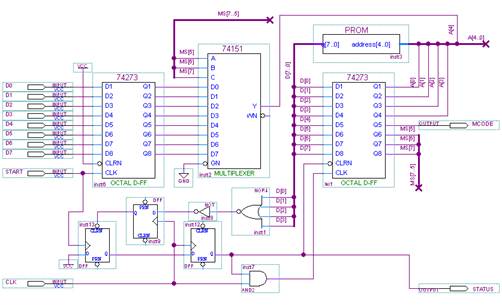


Рисунок 4.1 – Кодировщик манчестерского кода

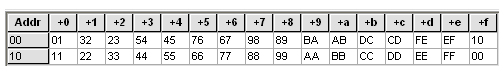


Рисунок 3.2 – Карта прошивки ПЗУ микропрограммного автомата

При помощи средств симуляции и временного анализа САПР Altera Quartus II получена временная диаграмма (Рисунок 3.3) для схемы кодировщика манчестерского кода, изображенного на рисунке 3.1. Временная диаграмма на рисунке 3.3 подтверждает правильность работы полученной схемы.

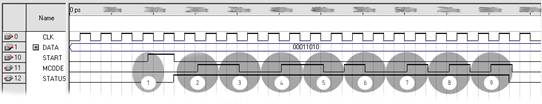


Рисунок 3.3 – Временная диаграмма кодировщика манчестерского кода

Пронумерованные маски на временной диаграмме синтезированного кодировщика (Рисунок 3.3) имеют следующие пояснения:

1.                Запуск кодировщика по переднему фронту сигнала START.

2.                Кодирование DATA[0], т.е. "0".

3.                Кодирование DATA[1], т.е. "1".

4.                Кодирование DATA[2], т.е. "0".

5.                Кодирование DATA[3], т.е. "1".

6.                Кодирование DATA[4], т.е. "1".

7.                Кодирование DATA[5], т.е. "0".

8.                Кодирование DATA[6], т.е. "0".

9.                Кодирование DATA[7], т.е. "0". Остановка кодировщика (переход кодировщика в состояние ожидания следующего сигнала START).

ВЫВОДЫ

Код Манчестер-II является биполярным двухуровневым самосинхронизирующимся кодом. Логическому нулю соответствует положительный фронт, а логической единицы отрицательный фронт закодированного сигнала. Бит обозначен переходом в центре тактового интервала, по которому и выделяется синхросигнал. Несомненное достоинство кода – отсутствие постоянной составляющей при передачах длинных последовательностей нулей или единиц.

Для чего во внешних запоминающих устройствах используется код Манчестер II? До недавнего времени был самым распространенным в локальных сетях (диаграмма г). Применяется в технологиях Ethernet и Token Ring. Для кодирования единиц и нулей используется перепад потенциала, то есть фронт импульса. При манчестерском кодировании каждый такт делится на две части. Информация кодируется перепадами потенциала, происходящими в середине каждого такта. Единица кодируется перепадом от высокого уровня сигнала к низкому, а ноль - обратным перепадом. Переходы производятся в середине временного интервала, отведенного каждому двоичному биту. Код Манчестер II легко получается из кода без возвращения к нулю, если последний подать на один вход схемы логической равнозначности, на второй вход которой подан синхросигнал в виде меандра с периодом, равным периоду кода БВН, и синфазный с ним. [+] обладает хорошими самосинхронизирующими свойствами. [+]Полоса пропускания манчестерского кода уже, чем у биполярного импульсного. [+]нет постоянной составляющей (половину времени сигнал положительный, половину – отрицательный). [+] требуется дополнительного источника питания для линии связи [+] в среднем ширина полосы манчестерского кода в полтора раза уже, чем у биполярного импульсного кода Код Манчестер-II, или манчестерский код, получил наибольшее распространение в локальных сетях. Он также относится к самосинхронизирующимся кодам, но в отличие от кода RZ имеет не три, а всего только два уровня, что способствует его лучшей помехозащищенности. Логическому нулю соответствует положительный переход в центре бита (то есть первая половина битового интервала - низкий уровень, вторая половина — высокий), а логической единице соответствует отрицательный переход в центре бита (или наоборот). Обязательное наличие перехода в центре бита позволяет приемнику кода Манчестер-П легко выделить из пришедшего сигнала синхросигнал, что дает возможность передавать информацию сколь угодно большими пакетами без потерь из-за рассинхронизации. Допустимое расхождение часов приемника и передатчика может достигать величины 25%.

ПРИЛОЖЕНИЕ

Таблица – ManchCoder Firmware

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ROM ADDRESS | ADDR BIN | | | | | DATA BIN | | | | | | | | ROM DATA |
| 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| HEX | In | Cell | | | | MS | | | Out | Next Cell | | | | HEX |
| 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 01 |
| 01 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 32 |
| 02 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 23 |
| 03 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 54 |
| 04 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 45 |
| 05 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 76 |
| 06 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 67 |
| 07 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 98 |
| 08 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 89 |
| 09 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | BA |
| 0A | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | AB |
| 0B | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | DC |
| 0C | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | CD |
| 0D | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | FE |
| 0E | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | EF |
| 0F | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 10 |
| 10 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 11 |
| 11 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 22 |
| 12 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 33 |
| 13 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 44 |
| 14 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 55 |
| 15 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 66 |
| 16 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 77 |
| 17 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 88 |
| 18 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 99 |
| 19 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | AA |
| 1A | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | BB |
| 1B | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | CC |
| 1C | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | DD |
| 1D | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | EE |
| 1E | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | FF |
| 1F | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 00 |