

УДК 004.31 : 371.355

Модель цифрового процессора

Ю. А. Журавель, С. Н. Рева

Харьковский национальный университет имени В.Н. Каразина

Кафедра электроники и управляющих систем

Факультет компьютерных наук

Описана учебная модель цифрового процессора, реализованная на элементах малой степени интеграции. Проведен анализ требований к цифровой модели как к наглядному пособию. Рассмотрены принципы организации процессов выполнения простейших команд, а также реализующие их схемные решения. Разработана структура командного слова и система команд модели цифрового процессора. Приведены электрические схемы отдельных блоков и описан их принцип работы. Описана конструкция и программные возможности модели, а также результаты ее использования в учебном процессе.

Ключевые слова: *цифровой процессор, учебная модель, наглядное пособие, архитектура процессора, система команд.*

Описана учбова модель цифрового процесора, що реалізована на елементах малої міри інтеграції. Проведено аналіз вимог до цифрової моделі як до наочного посібника. Розглянуто принципи організації процесів виконання простих команд, а також схемні рішення, що їх реалізують. Розроблено структуру командного слова та систему команд моделі цифрового процесора. Приведені схеми окремих електронних блоків та описано їх принцип дії. Описана конструкція і програмні можливості моделі, а також результати її використання в учбовому процесі.

Ключові слова: *цифровий процесор, учбова модель, наочний посібник, архітектура процесора, система команд.*

The educational model of a digital processor which uses the elements of small integration degree is described. The analysis of requirements to the digital model as to the visual aid is conducted. Processes organization principles of the simplest commands implementation and realizing them schematics are considered. The command word structure and the instruction set of the digital processor model have been developed. The electrical circuits of the certain units are given and the principles of their operation are described. A construction and programmatic facilities of the model are described. The results of its usage in an educational process are provided.

Key words: *digital processor, educational model, visual aid, processor architecture, instruction set.*

1. Введение

Разработку современных электронных устройств и систем сегодня уже не возможно представить без применения микроконтроллеров и микропроцессоров. Микропроцессоры, как элементная база, все более широко используются при проектировании измерительных приборов, систем автоматического управления, бытовых приборов и даже игрушек для детей. Создание таких устройств требует от разработчиков высокого уровня квалификации как в области электроники, так и программирования. К сожалению, многие учебные заведения технического направления готовят специалистов достаточно узкого профиля. Выбрав для себя программирование будущей специальностью, студенты далеко не всегда уделяют должное внимание изучению основ электроники и схемотехники. Такие специалисты могут прекрасно себя реализовать, работая системными и web-

программистами, но наверняка столкнуться со сложностями при разработке управляющих программ для микроконтроллерных устройств и сложных аппаратных комплексов. Реализация таких проектов требует высокой квалификации в обеих областях знаний, понимания физики происходящих в системах процессов и принципов взаимодействия электронных и программных средств.

К сожалению, на сегодняшний день почти отсутствует учебное оборудование [1], позволяющее эффективно организовать процесс ознакомления студентов с основами схемотехники микропроцессорных устройств.

Для повышения уровня подготовки программистов в области электроники кафедрой электроники и управляющих систем Харьковского национального университета имени В. Н. Каразина при участии студентов старших курсов разработана учебная модель цифрового процессора. Она призвана заинтересовать студентов в изучении основ организации и работы микропроцессорных систем, помочь в освоении схемных решений и архитектуры микроконтроллеров, понять особенности программирования на языках низкого уровня.

2. Требования к модели цифрового микропроцессора

Приступая к проектированию модели процессора, авторами разработки были сформулированы следующие требования, предъявляемые к модели как к наглядному пособию.

1. В отличие от большинства современных микропроцессоров, имеющих достаточно сложную и громоздкую архитектуру, модель должна быть простой и доступной для восприятия, чтобы при работе с ней можно было легко понять принципы функционирования и отследить прохождение и обработку отдельных сигналов. Целесообразно разработать модель, которая по своей архитектуре будет приближенная к архитектуре фон Неймана [2].

2. Модель должна быть достаточно функциональной и использовать в качестве инструкций и параметров команд восьмиразрядные двоичные числа, прививая тем самым студентам навыки работы с байтовой организацией данных. Следовательно, разрабатываемая модель должна создаваться как восьмиразрядный цифровой процессор и использовать восьмиразрядные шины адреса и данных.

3. Модель должна выполнять основные типы операций, реализованных в большинстве промышленных микропроцессоров, такие как операции пересылки данных, логические и арифметические операции, а также команды безусловного и условного перехода.

4. Для понимания принципов организации и функционирования микропроцессорного ядра модель должна быть построена с использованием элементной базы малой степени интеграции. Основой модели должны стать базовые логические элементы, триггеры, регистры и счетчики.

5. Необходимо оснастить модель узлами ввода-вывода данных и обеспечить ее взаимодействие с внешними цифровыми устройствами.

6. Модель должна иметь высокую степень наглядности, должна быть оснащена встроенными индикаторами состояния основных регистров и шин.

7. Необходимо предусмотреть возможность сопряжения модели цифрового процессора с персональным компьютером для автоматизации процессов написания, отладки и записи учебных программ, а также для мониторинга текущего состояния модели и его отображения на экране дисплея.

3. Машинный цикл и организация процессов выполнения команд

Для того, чтобы иметь возможность представить выполнение команды как совокупность строго синхронизированных процессов, происходящих в ядре, разработчиками было принято решение построить модель с многотактовым машинным циклом, то есть, организовать процесс выполнения команды за несколько периодов тактовой частоты. По этому принципу организован машинный цикл в микроконтроллерах серии Intel 8051 [3]. При этом сигналы управления будут формироваться таким образом, чтобы в течение одного такта выполнялось не более одного действия какого-либо процесса, например: на первом такте формируется сигнал разрешения чтения программной памяти, на втором — формируется сигнал записи прочитанных данных, и данные заносятся в какой-либо регистр и т. д.

Чтобы обеспечить процесс последовательного выполнения программы необходимо на последнем этапе выполнения текущей команды путем модификации состояния счетчика команд сформировать новое значение адреса, по которому расположена инструкция следующей команды.

Наиболее доступной для понимания моделью ядра процессора является модель, реализующая выполнение команды как процесс перезаписи данных из некоторого источника (регистра, ячейки памяти, арифметического сумматора и т. п.) в приемник данных, роль которого выполняет один из регистров ядра.

Для реализации этого процесса машинный цикл разделен на несколько этапов. На первом этапе необходимо обеспечить перезапись инструкции команды в специальный регистр (регистр команд) к выходу которого подключена схема анализа выполняемой инструкции. В процессе дешифрации инструкции определяются адреса источника и приемника данных, после чего на следующих этапах машинного цикла могут быть сформированы сигналы разрешения выдачи данных из источника на внутреннюю шину процессора, а затем — записи данных в приемник. Процесс выполнения команды должен завершаться инкрементом счетчика команд, который является формирователем адресной шины.

Машинный цикл модели процессора состоит из восьми периодов тактовой частоты. Основные сигналы, обеспечивающие выполнение операций, показаны на рис.1.

Машинный цикл начинается с формирования низкого логического уровня сигналов RM и WC. Сигнал чтения данных из памяти программ RM (Read Memory) предназначен для разрешения выдачи данных из ячейки памяти, адрес которой в этот момент времени присутствует на шине адреса. С некоторой задержкой (в один период тактовой частоты), необходимой для завершения переходных процессов на шине данных, формируется положительный перепад сигнала WC (Write Command). Этот сигнал предназначен для записи инструкции команды в регистр команд. По окончании второго периода тактовой частоты сигнал RM переходит в высокий логический уровень, освобождая тем самым

шину данных. В этот же момент времени формируется специальный строб в виде низкого логического уровня сигнала ЕС (Enable Comand), который предназначен для разрешения проведения операции пересылки данных из источника в регистр-приемник.

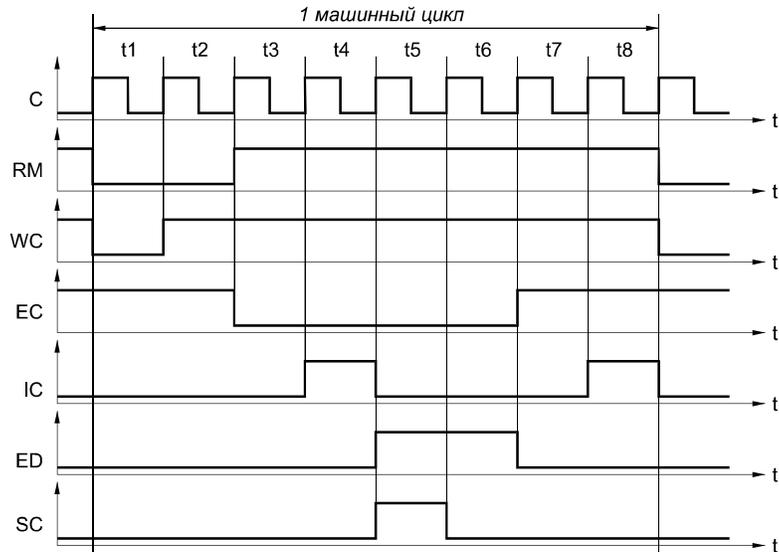


Рис.1. Сигналы машинного цикла

Если моделью процессора выполняется двухбайтная команда, параметр которой хранится в ячейке памяти, следующей за инструкцией текущей команды, то необходимо выполнить инкремент счетчика команд. При этом на шине адресе будет сформирован адрес ячейки, в которой хранится параметр. Для этой цели генератором сигналов машинного цикла формируется импульс управляющего сигнала IC (Increment Counter). Если выполняется однобайтная команда, то этот импульс будет замаскирован в течение периода времени, пока присутствует низкий логический уровень сигнала ЕС, и в этом случае инкремент счетчика команд не будет выполнен. Активный уровень сигнала IC формируется в течение четвертого периода тактовой частоты в рамках машинного цикла. По его окончанию начинается процесс пересылки данных из источника данных в регистр-приемник. Для этого на пятом периоде формируется высокий уровень двух сигналов: сигнала ED (Enable Data), который предназначен для разрешения выдачи данных на шину из источника данных; и сигнала SC (Strobe Command), предназначенного для стробирования дешифраторов сигналов записи в регистры-приемники. Непосредственно процесс перезаписи данных выполняется по спаду сигнала SC в конце пятого периода машинного цикла. По окончании шестого периода сигнал ED сбрасывается, что приводит к освобождению шины данных. Завершение процесса выполнения команды происходит на восьмом периоде тактовой частоты, в течение которого формируется второй импульс сигнала IC, приводящий к инкременту счетчика команд. Этот импульс не может быть замаскирован. Таким образом, после

завершения восьмого периода на шине адреса устанавливается адрес, соответствующий расположению первого байта следующей команды.

4. Генератор сигналов машинного цикла

Для получения последовательности описанных выше сигналов в модели используется схема, показанная на рис. 2.

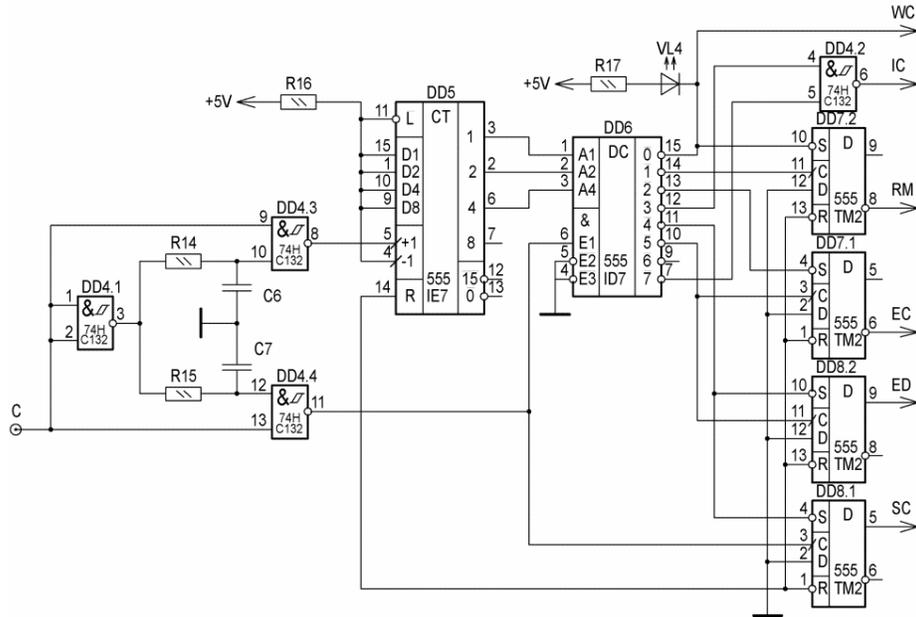


Рис. 2. Схема формирования управляющих сигналов

Схема реализована по принципу микропрограммного автомата [4]. Тактовая частота поступает на устройство выделения фронта, которое построено на логических элементах DD4.1, DD4.3. На выходе элемента DD4.3 во время прохождения фронта тактовых импульсов формируются короткие сигналы нулевого логического уровня, которые задерживают момент переключения двоичного синхронного счетчика DD5 на несколько десятков наносекунд относительно фронта импульсов тактовой частоты. В это время на выходе элемента DD4.3 также присутствует импульс нулевого логического уровня, длительность которого превышает длительность импульса на первом выходе формирователя. Он используется для запрета работы дешифратора DD6 на время изменения состояния счетчика DD5 с целью устранения неоднозначности состояния схемы. Первый выходной сигнал дешифратора используется в качестве управляющего сигнала WC. Синтез сигнала IC производится с помощью логического объединения двух выходных сигналов дешифратора на элементе DD4.4. Остальные управляющие сигналы формируются в результате переключения триггеров DD7.1, DD7.2, DD8.1, DD8.2 на соответствующих фазах работы генератора сигналов машинного цикла.

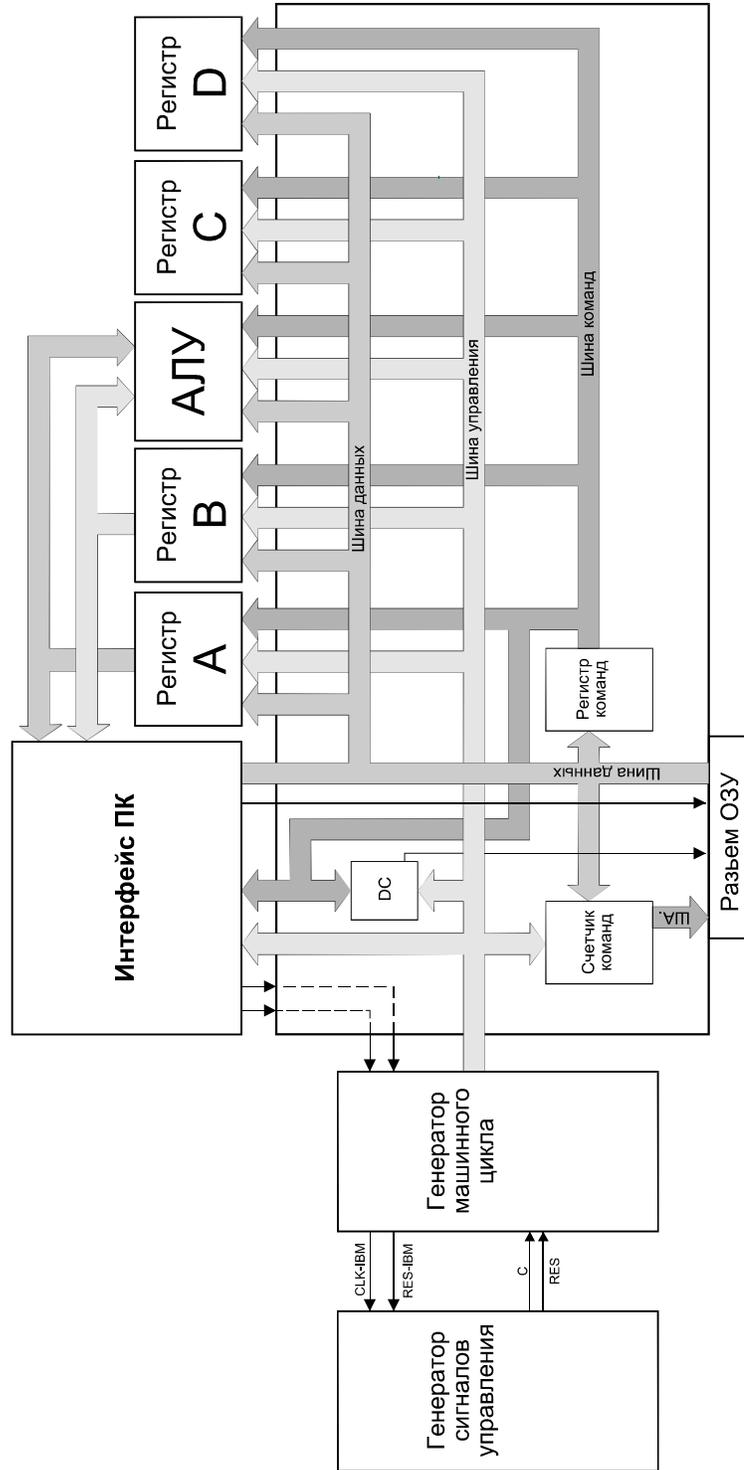


Рис. 3. Модель цифрового процессора. Схема структурная

5. Структура процессорного ядра

Общая структурная схема модели цифрового процессора показана на рис. 3. Основной структурный блок модели включает в себя восьмиразрядный двоичный счетчик с возможностью параллельной загрузки данных, который выполняет функции счетчика команд. Его выходные сигналы формируют внешнюю адресную шину устройства. Кроме того, в блоке расположен восьмиразрядный регистр временного хранения выполняемой инструкции (регистр команд) и системный дешифратор, предназначенный для анализа байта выполняемой инструкции.

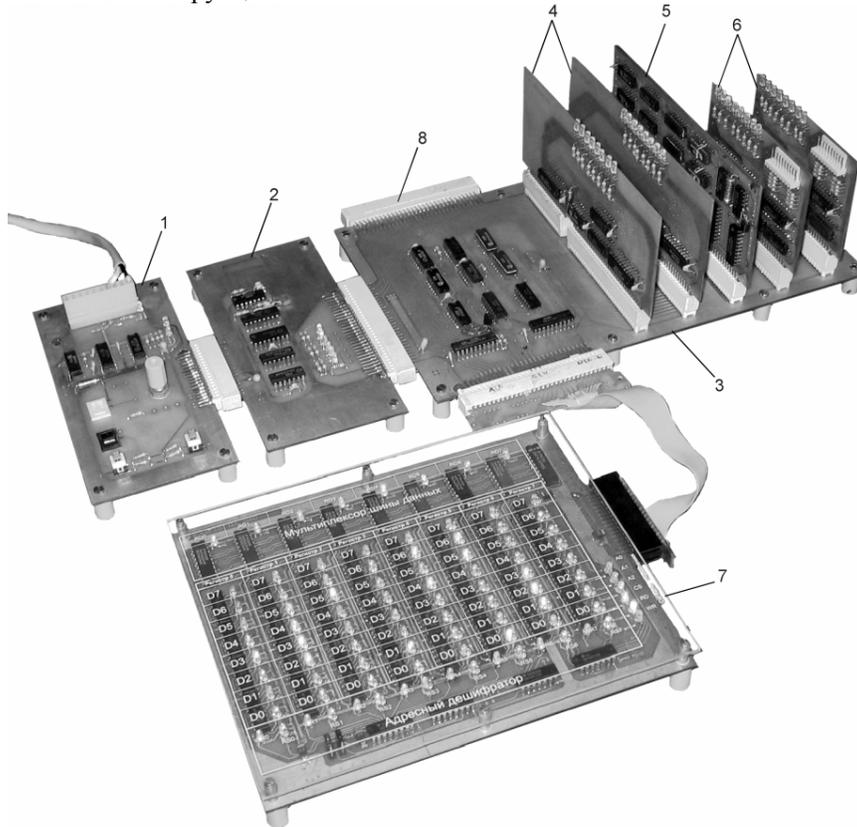


Рис. 4. Внешний вид модели цифрового процессора

1 - генератор сигналов управления; 2 - генератор сигналов машинного цикла; 3 - основной блок модели процессора; 4 - регистры A и B; 5 - АЛУ; 6 - регистры C и D; 7 - модуль памяти программ; 8 - разъем подключения интерфейсного модуля для сопряжения с персональным компьютером

Остальную часть блока составляют внутренние параллельные шины, предназначенные для передачи данных и управляющих сигналов во все регистры модели цифрового процессора. Среди них можно выделить восьмиразрядную двунаправленную шину данных, объединяющую между собой

все регистры ядра и поступающую на вход счетчика команд и внешние интерфейсы, однонаправленную шину управления, по которой выходные сигналы генератора машинного цикла поступают на все функциональные узлы модели, а также однонаправленную шину команд, передающую выполняемую инструкцию во все регистры ядра, на вход системного дешифратора и на интерфейс подключения персонального компьютера. В основном блоке модели процессора установлен ряд магистральных разъемов для подключения регистров А, В, С и D а также арифметико-логического устройства (АЛУ) к перечисленным выше шинам (смотри рис. 4). Блок также содержит разъем для предназначенный для подключения памяти программ, через который выводятся шины адреса и данных и сигналы чтения и записи.

Модель процессора предусматривает возможность подключения специального интерфейса, позволяющего контролировать и формировать все основные сигналы модели с помощью персонального компьютера.

Генератор тактовой частоты также представлен в виде отдельного блока, функциональные возможности которого будут описаны далее.

6. Структура командного слова

Инструкция каждой выполняемой команды представляет собой восьмиразрядное двоичное число, разряды которого определяют состояние сигналов на шине команд. Все биты, записанного в регистр команд байта инструкции приобретают свое функциональное назначение. Структура командного слова представлена в таблице 1.

Табл. 1. Назначение бит командного слова

Бит	Назначение
CM0	Адрес регистра
CM1	приемника данных
CM2	
CM3	
CM4	Адрес источника данных
CM5	
CM6	Бит-признак двухбайтной команды
CM7	Бит-признак командного перехода

В качестве регистра приема данных при выполнении команд пересылки данных может быть один из четырех регистров А, В, С или D, входящих в состав модели процессора. Они же могут служить и источниками данных. Однако в структуре командного слова зарезервировано пять разрядов для формирования адреса возможных источников данных. Это сделано для того, чтобы в качестве источника данных можно было определить одну из логических (арифметических) схем, входящих в состав арифметико-логического устройства. В этом случае любая из операций АЛУ может быть также выполнена как операция пересылки данных, только источником будет являться, например, схема логического суммирования или инверсии. Установленный в единицу бит CM5 командного слова является битом-признаком работы с АЛУ, а биты CM2 ..

СМ4 определяют адрес источника данных в составе АЛУ. В работу модели процессора заложено ограничение, которое при выполнении логических и арифметических операций определяет в качестве приемника регистр А.

Для выполнения двухбайтных операций в состав командного слова введен бит-признак СМ6, при установке которого в единицу снимается маскирование первого импульса сигнала ІС, что позволяет в середине машинного цикла выполнить инкремент адреса для чтения второго байта (параметра) команды. Снятие маскирования этого импульса выполняется также при установке в единицу бита СМ7. Но этот бит аппаратно разрешает запись в счетчик команд, точнее назначает счетчик команд в качестве приемника данных. Таким образом в макет заложена возможность программного изменения текущего адреса, являющаяся основой команд условного и безусловного перехода.

7. Базовая схема регистра

Организацию одного из регистров модели процессора рассмотрим на примере регистра В. Его принципиальная схема показана на рис. 5.

Регистр выполнен в виде отдельного модуля, который устанавливается в один из разъемов системной магистрали макета. В состав модуля входит восьмиразрядный регистр хранения данных DD1, магистральный формирователь DD4, светодиодный индикатор состояния регистра, построенный на светодиодах V1..V8, а также дешифратор собственного адреса (микросхемы DD2, DD3).

В качестве регистра хранения данных можно было бы использовать микросхему с управляемым третьим состоянием, например, K555IP22 (74LS373) или K555IP23 (74LS374). В этом случае не было бы необходимости в использовании магистрального формирователя. Но постоянный доступ к данным был бы невозможен, так как основную часть времени регистр находился бы в закрытом состоянии, и не было бы возможности индцировать его содержимое с помощью светодиодов. Поэтому функции регистра хранения данных выполняет микросхема K555IP35 (74LS273), выходы которой всегда находятся в активном состоянии и управляют работой светодиодного индикатора. Подключение модуля к шине данных в качестве источника информации производится с помощью микросхемы DD4 (K555АП6) при подаче на управляющий вход ОЕ нулевого логического уровня сигнала RD_В. Так как регистр В может использоваться в составе модели процессора для хранения одного из операндов арифметических и логических операций, то данные этого регистра непосредственно выведены на дополнительную шину АЛУ через разъем ХМ2.

Регистр В имеет системный адрес равный единице. Поэтому дешифраторы DD2 и DD3 включены таким образом, чтобы нулевые уровни сигналов WR_В и RD_В формировались при установке на управляющей шине первого адреса в разрядах СМ0, СМ1 или СМ2, СМ3 соответственно. Дешифратор DD3 стробируется сигналом ED, что обеспечивает подключение регистра на шину данных в качестве источника в соответствующей фазе машинного цикла, а дешифратор DD2 стробируется сигналом SC, что позволяет записывать данные в регистр по спаду этого сигнала.

Аналогичным образом построены регистры А, С и D, но они имеют свои особенности. Дешифратор регистра А разрешает запись данных в регистр не только при установке нулевого адреса (системный адрес регистра А), но и при установке в единицу бита CM5. В этом регистре также присутствует схема совпадения нулевого состояния всех разрядов для формирования сигналов разрешения операции условного перехода по нулю в регистре А.

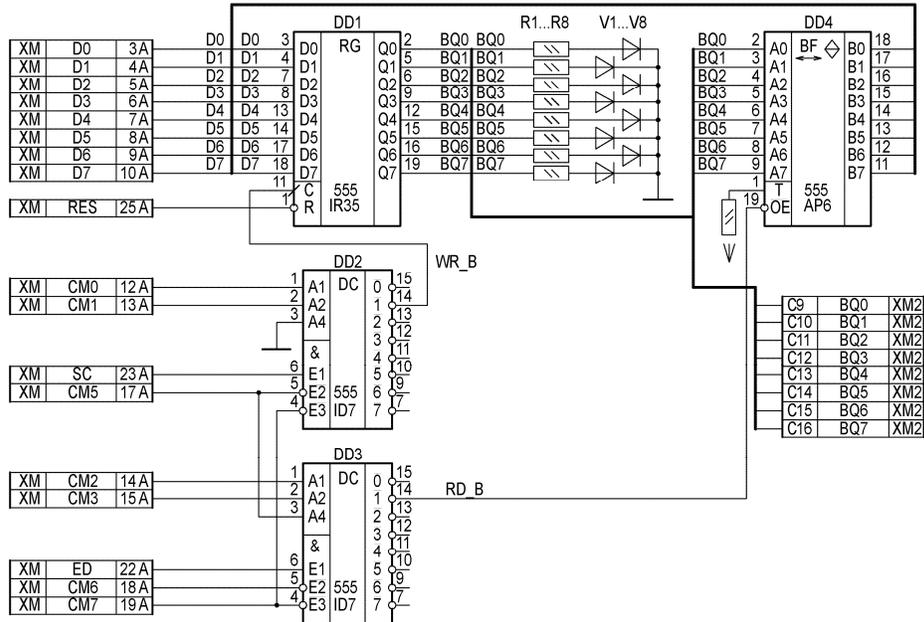


Рис. 5. Регистр В. Схема электрическая принципиальная

В регистрах С и D выходные сигналы с регистра хранения на шинный формирователь поступают через элементы с открытым коллектором, что позволяет принудительно изменять их логический уровень внешними сигналами, поступающими через дополнительный разъем. Таким образом регистры С и D приобретают возможность ввода и вывода внешних сигналов и могут выполнять функции портов.

8. Арифметико-логическое устройство

Как упоминалось ранее выполнение арифметических и логических операций организовано в макете как процесс перезаписи данных из некоторого функционального узла в регистр А. В качестве операндов при выполнении этих операций могут использоваться данные, записанные в регистры А и В. Отличительной особенностью регистров А и В является наличие дополнительного выхода данных непосредственно с регистра хранения на специальные дополнительные шины, по которым данные поступают непосредственно в АЛУ. Записанная в регистры информация присутствует на этих шинах все время независимо от стадии машинного цикла.

Фрагмент принципиальной схемы, поясняющий организацию АЛУ, показан на рис. 6. Его работу рассмотрим на примере выполнения функции циклического сдвига влево (Func. RL) и функции логического перемножения (Func. AND).

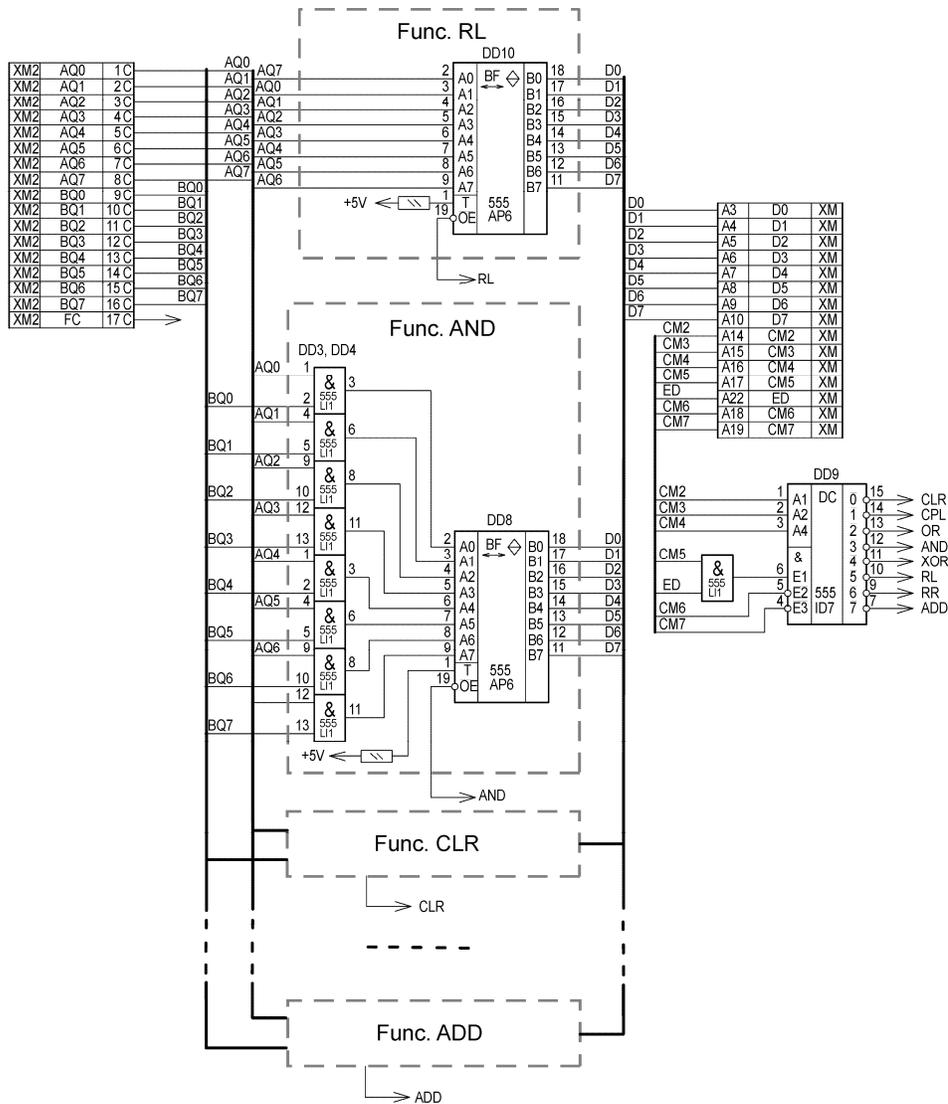


Рис. 6. Схема АЛУ

Данные из регистра А по дополнительной восьмиразрядной шине AQ7..AQ0 поступают на вход шинного формирователя DD10 со сдвигом, то есть разряд AQ0 подключен ко входу A1, AQ1 — ко входу A2 и т.д., а разряд AQ7 подключен на вход A0. Таким образом, если микросхема DD10 будет являться

источником данных в команде перемещения данных, то информация будет записана в регистр-приемник с циклическим смещением разрядов.

Вторым примером является реализация функции логического перемножения. Данные из регистров А и В по дополнительным шинам AQ7..AQ0 и BQ7..BQ0 поразрядно поступают на входы логических элементов «И» микросхем DD3, DD4. Полученные логические произведения отдельных бит в виде сигналов поданы на входы шинного формирователя DD8. Если при выполнении команды перемещения данных этот шинный формирователь будет выступать в роли источника, то логическое произведение двух байт будет записано в регистр-приемник (регистр А).

Кроме рассмотренных выше функций арифметико-логическое устройство макета может также выполнять функции циклического сдвига вправо, логического сложения, инверсии байта, обнуления байта, функцию «исключающее ИЛИ» и функцию арифметического сложения. Активизация этих функций выполняется при формировании логического нуля на одном из выходов микросхемы DD9, которая является дешифратором командного слова.

9. Генератор сигналов управления

Работа модели цифрового процессора синхронизируется сигналами тактовой частоты, поступающими на вход генератора сигналов машинного цикла. Для выработки этих сигналов в состав модели включен модуль генератора сигналов управления (рис. 7), который содержит в себе генератор тактовых импульсов и генератор сигнала сброса (Reset). Для обеспечения наглядности происходящих в модели процессов предусмотрено несколько режимов работы генератора тактовых импульсов. Импульсы могут вырабатываться однократно при каждом нажатии кнопки S1, генерироваться с помощью автогенератора, собранного на базе логического элемента DD2.1, или формироваться на основе внешнего управляющего сигнала CLK-IBM, поступающего с персонального компьютера. Выбор режима производится с помощью двух переключателей S2 и S3.

Если переключатель S2 замкнут, то низким логическим уровнем, поступающим на вход 1 элемента DD2.1, блокируется работа автогенератора. В этом режиме возможно формирование тактовых импульсов от кнопки S1, которая при нажатии поочередно формирует низкие логические уровни на R и S входах триггера, собранного на элементах DD1.1 и DD1.2. Триггер предназначен для подавления дребезга контактов кнопки с целью формирования одиночного импульса при однократном нажатии. В этом же режиме возможно формирование тактовых импульсов с помощью управляющего компьютера. Логическое объединение сигналов выполняется элементом DD3.1. При разомкнутом положении переключателя S2 начинает работать автогенератор. Его рабочая частота определяется параметрами цепочки R4,R5,C2 и может регулироваться с помощью переменного резистора R5 в достаточно широких пределах (от 10 до 100 кГц). Данный режим предназначен для отображения последовательности сигналов, протекающих в течение машинного цикла, на экране осциллографа. Если переключатель S3 замкнут, то параллельно конденсатору C2 подключается электролитический конденсатор C1 большой емкости, благодаря чему рабочий диапазон генератора смещается в область инфранизких частот (0,5..5 Гц). Такой режим позволяет визуально наблюдать

формирование и прохождение сигналов с помощью светодиодных индикаторов, установленных на основных шинах модели процессора.

Сигналы сброса могут формироваться либо с помощью триггера на элементах DD1.3 и DD1.4 при нажатии на кнопку S4, либо под воздействием внешнего сигнала RES-IBM, поступающего с управляющего компьютера.

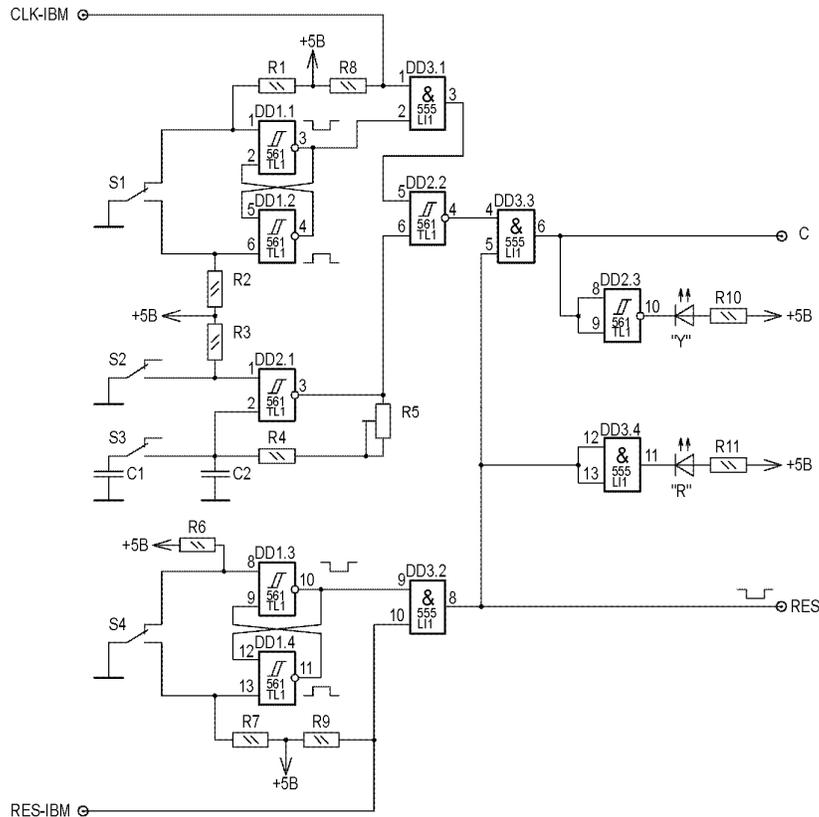


Рис.7. Схема генератора управляющих сигналов

10. Заключение

Описанная в статье модель цифрового процессора реализована на кафедре электроники и управляющих систем Харьковского национального университета имени В.Н. Каразина. На протяжении двух лет она используется в качестве наглядного пособия и лабораторного макета в процессе преподавания дисциплин «Архитектура компьютеров», «Микропроцессоры и их применение», «Системы автоматического контроля и управления». Для макета разработана и изготовлена интерфейсная плата, позволяющая подключить его к персональному компьютеру. При участии сотрудников кафедры и студентов старших курсов факультета компьютерных наук разработано программное обеспечение, которое позволяет дополнительно визуализировать состояние

основных регистров и шин модели на экране монитора, выполнять пошаговое управление моделью процессора, а также непосредственно работать с памятью программ, подключаемой к модели. Создан модуль программной памяти емкостью 256 байт, позволяющий создавать достаточно объемные демонстрационные и учебные программы, которые в полной мере раскрывают возможности модели.

Ее использование в учебном процессе формирует у студентов более глубокое понимание физических процессов, происходящих в микропроцессорных устройствах, тем самым подготавливает базу для осмысленного перехода к абстрактному мышлению на уровне языков программирования.

Удачное конструктивное и дизайнерское оформление наглядного пособия всегда привлекает внимание студентов и пробуждает интерес к изучению микропроцессорной техники, а следовательно способствует расширению их технической эрудиции и повышению уровня квалификации будущих специалистов.

ЛИТЕРАТУРА

1. Джулгаков В.Г., Руденко К.И. Универсальная учебная микроконтроллерная система УУМС-2. – Х.: Харьковский аэрокосмический университет имени Н.Е. Жуковского, кафедра систем управления летательными аппаратами.
2. Полунов Ю.Л. От абака до компьютера: судьбы людей и машин. Книга для чтения по истории вычислительной техники в двух томах. Том 1. – М.: Русская Редакция, 2004. – 480 с.
3. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристалльных микроконтроллерах. – М.: Энергоатомиздат, 1990. – 223с.
4. Савельев А.Я. Арифметические и логические основы цифровых автоматов программируемой логики. – М.: Высшая школа, 1980. – 255 с.