

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ
(ТУСУР)

Кафедра промышленной электроники (ПрЭ)

А.И. Воронин

Цифровая и микропроцессорная техника

Лабораторный практикум для
студентов направления подготовки
11.03.04 "Электроника и наноэлектроника"

2018

Воронин А.И.

Цифровая и микропроцессорная техника:
лабораторный практикум для студентов направления
подготовки 11.03.04 "Электроника и микроэлектроника"
очной формы обучения – Томск: Томский
государственный университет систем управления и
радиоэлектроники, 2018.– 75 с.

СОДЕРЖАНИЕ

1. Лабораторная работа №1. Синтез цифровых устройств на логических элементах	4
2. Лабораторная работа №2. Проектирование комбинационных цифровых устройств.	21
3. Лабораторная работа №3. Проектирование синхронного счетчика	25
4. Лабораторная работа №4. Программная модель и система Команд МК51....	33
5. Лабораторная работа №5. . Управление устройствами ввода/вывода с помощью параллельных портов МК51	54
6. Лабораторная работа №6. Управление жидкокристаллическим индикатором.....	66
7. Список рекомендуемой литературы.....	75

1.Лабораторная работа №1

Синтез цифровых устройств на логических элементах

1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является приобретение навыков логического проектирования комбинационных цифровых устройств и практическая реализация таких устройств на логических элементах, экспериментальная проверка правильности их функционирования. Преподаватель назначает вариант задания (от 1 до 12), выполняемого каждой рабочей подгруппой. Схема устройства собирается на монтажном поле лабораторного макета соединительными проводниками.

2. ЛАБОРАТОРНАЯ УСТАНОВКА

Лабораторная установка включает лабораторный макет (Приложение В), приставку к персональному компьютеру PCSU1000 (Приложение Б), реализующая двухканальный виртуальный осциллограф на экране персонального компьютера, соединительные провода, комплект логических элементов ТТЛ (Приложение А).

3. БАЗОВЫЕ СХЕМЫ ЭЛЕМЕНТА ТТЛ

Простейший логический элемент (рис.1) строится на базе многоэмиттерного транзистора V_{Tm} , выполняющего функцию И для сигналов, подаваемых на его эмиттеры, и транзисторного ключа (V_{T1}), выполняющего функцию НЕ. Если на всех входах высокие уровни напряжения ($A=1$ и $B=1$), закрыты эмиттерные переходы V_{Tm} , открывается переход база-коллектор этого транзистора и ток I открывает и насыщает ключевой транзистор V_{T1} , формируя на выходе низкий уровень напряжения ($F=0$). Если хотя бы на одном из входов низкий уровень напряжения, ток I переключается в выходную цепь источника сигнала, закрывая V_{T1} . При этом $F=1$. Таким образом, схема реализует таблицу истинности логического элемента 2И-НЕ.

Для повышения быстродействия выходной ключ выполняют по схеме сложного инвертора (рис.2). При $A=B=1$ открыты транзисторы V_{T2}, V_{T1} (транзистор V_{T3} закрыт) и емкость нагрузки быстро

разряжается через сопротивление открытого ключа VT1. Для любой другой комбинации входных сигналов емкость нагрузки имеет возможность быстро зарядиться до высокого уровня напряжения через низкое выходное сопротивление эмиттерного повторителя VT3 (VT2 и VT1 закрыты).

Элементы ТТЛШ отличаются от ТТЛ тем, что коллекторные переходы транзисторов зашунтированы диодами Шоттки. Транзисторы элементов ТТЛШ не входят в насыщение, что существенно уменьшает задержку выключения транзисторов. К тому же они меньших размеров, что уменьшает емкости р-п-переходов и потребляемую мощность.

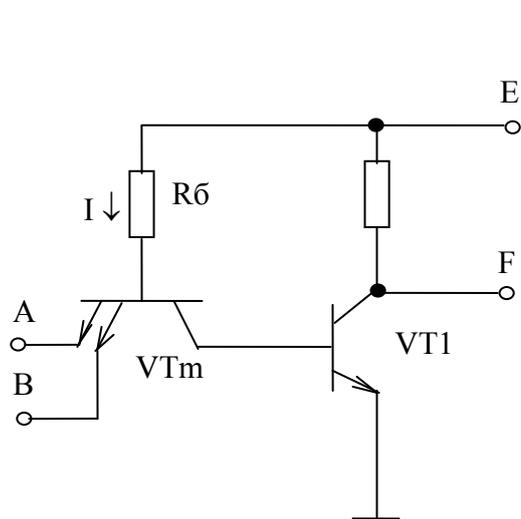


Рисунок 1

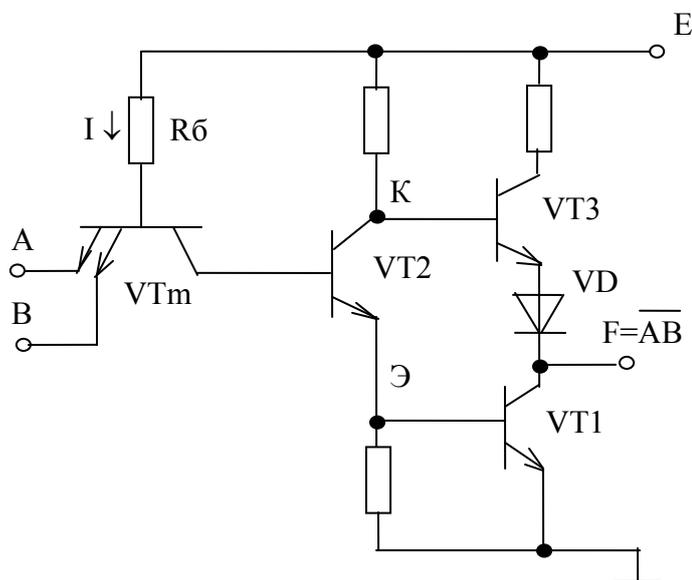


Рисунок 2 - Базовая схема элемента ТТЛ

В элементе с открытым коллекторным выходом (рис.3) VT3 и VD отсутствуют. Вместо них подключают элементы автоматики (обмотки реле) или индикации (например, светодиод). Такие элементы допускают объединение выходов. Пример применения логического элемента с открытым коллекторным выходом (микросхема К155ЛА8) показан на рис.4.

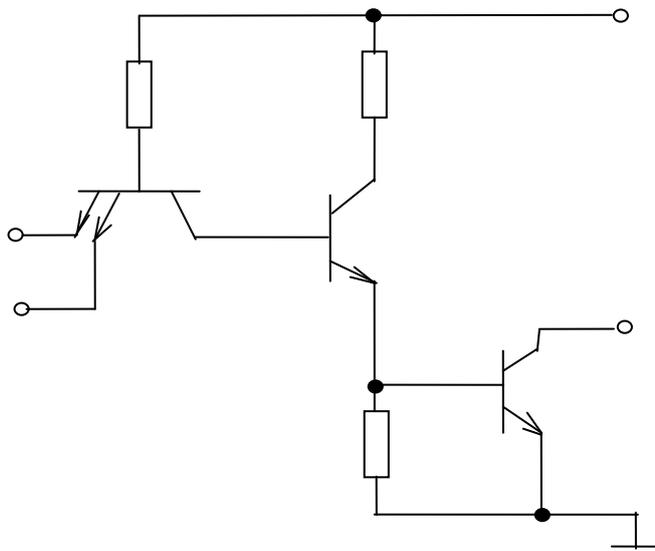


Рисунок 3

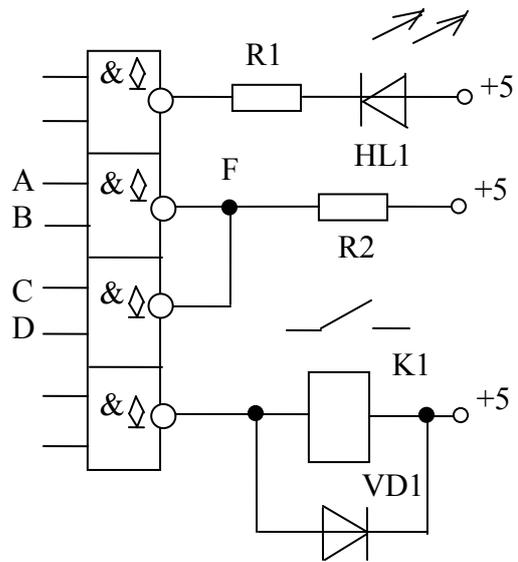


Рисунок 4

На выходе F реализуется логическая функция $F=AB+CD$, т.е. логический ноль наблюдается при совпадении логических единиц на входах A и B или C и D. Таким образом, объединение выходов позволяет путем монтажа выполнить логическую операцию ИЛИ (монтажное ИЛИ, монтажное И).

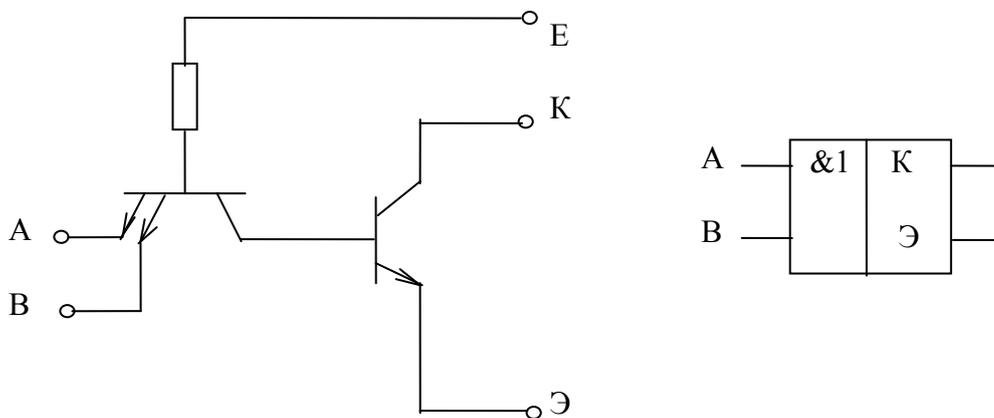


Рисунок 5 - Логический расширитель

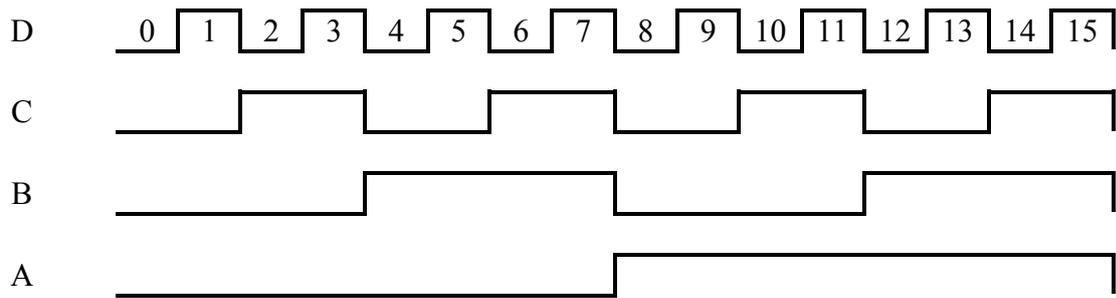
В схеме логического расширителя (рис.5) задействованы лишь элементы Rб, VTm и VT2 базовой схемы. Логические расширители (например, K155ЛД1) используются совместно с другими логическими элементами, имеющими соответствующие входы К (открытый коллектор) и Э (открытый эмиттер), например, K155ЛР3

или К155ЛР1.

4. ВАРИАНТЫ ЗАДАНИЙ

Лабораторная работа включает проектирование генератора импульсов сложной формы по заданной временной диаграмме выходного сигнала в течение периода (рис.6). Такой генератор, кроме генератора тактовых импульсов макета (ГТИ) и четырехразрядного двоичного счетчика (микросхема К555ИЕ18), включает комбинационную схему, которую необходимо синтезировать и экспериментально проверить.

Значения булевой функции F , реализуемой комбинационной схемой на 16 наборах входных сигналов A, B, C, D (выходы двоичного счетчика), соответствуют уровням выходного напряжения генератора на 16 тактах периода генерируемых импульсов. Таким образом, для каждого варианта задана таблица истинности логической функции четырех переменных F .



Варианты заданий

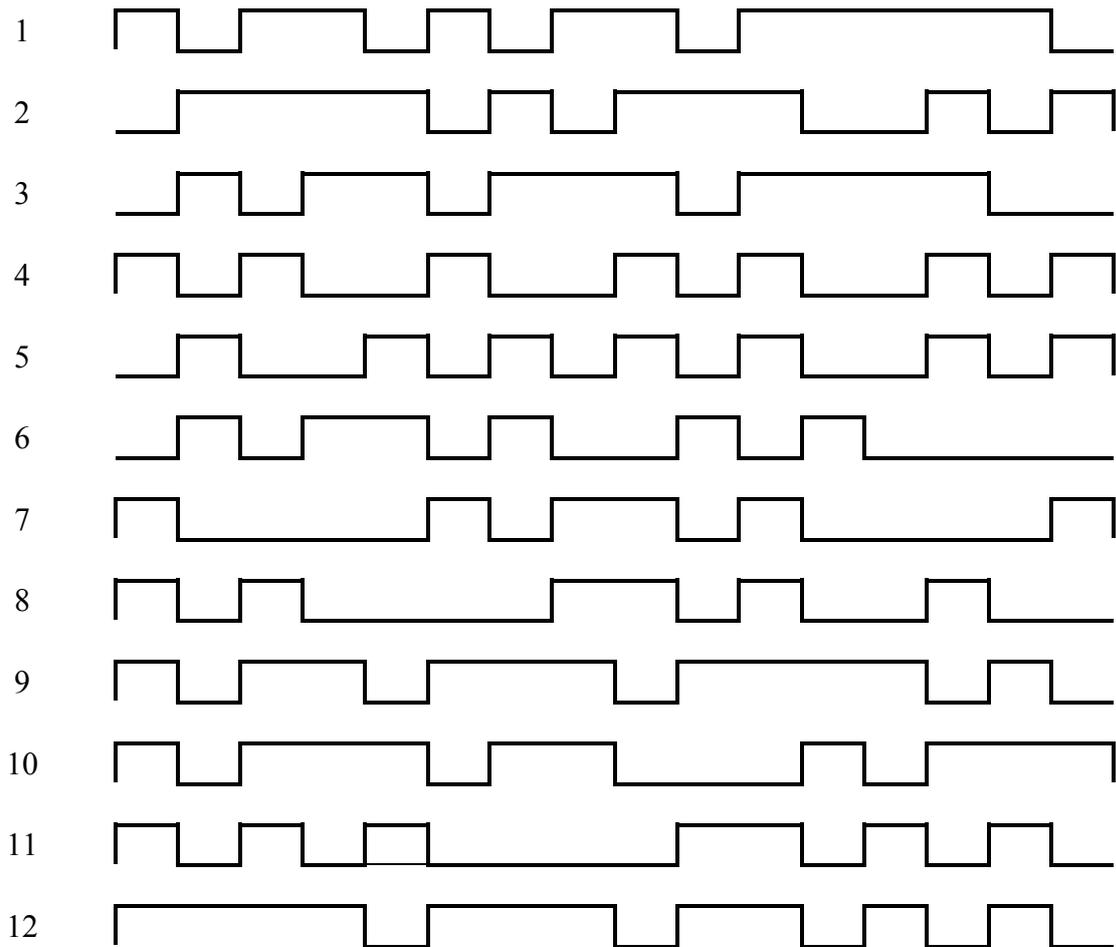


Рисунок 6 - Варианты заданий

При проектировании комбинационного цифрового устройства на логических элементах для минимизации булевой функции удобно воспользоваться картой Карно. Пример карты Карно приведен на рис.7, в каждую клетку карты Карно записывается значение булевой функции на соответствующем тактовом интервале работы комбинационного устройства, рис. 8.

		<u>A</u>				
		1	0	0	0	
		1	1	0	1	D
C		1	1	0	0	
		1	1	1	1	
		<u>B</u>				

Рисунок 7 - Карта Карно

0	4	12	8
1	5	13	9
3	7	15	11
2	6	14	10

Рисунок 8

Правила записи минимизированного выражения для логической функции по карте Карно:

- 1) выделяются блоки, заполненные единицами;
- 2) блок должен быть прямоугольным и содержать 1, 2, 4, 8 клеток;
- 3) блоки должны быть возможно большими, а их количество наименьшим;
- 4) левая и правая, а также верхняя и нижняя строки карты считаются соседними;
- 5) одна и та же клетка может входить в несколько блоков;
- 6) функция может доопределяться произвольно (на тех наборах, где стоят X), чтобы получить наиболее крупные блоки;
- 7) функция записывается в виде логических произведений (ЛП), описывающих выделенные блоки;
- 8) переменная не включается в ЛП, если блок областью ее прямых значений делится пополам;
- 9) переменная включается в ЛП с инверсией, если рассматриваемый блок лежит в области ее инверсных значений;
- 10) при группировке в блоки клеток, заполненных нулями, по тем же правилам получаем инверсное значение логической функции.

5. ПРОГРАММА РАБОТЫ

5.1 Включив макет, зафиксировать на осциллографе временные диаграммы сигналов на выходах двоичного счетчика D, C, B, A. Измерить их частоту и амплитуду. При снятии осциллограмм

использовать синхронизации режим по первому каналу (СН1) по отрицательному фронту.

5.2 Провести синтез комбинационного цифрового устройства для заданного варианта на логических элементах и после проверки его преподавателем произвести макетирование устройства монтажном поле лабораторного макета. Включив макет, пронаблюдать временную диаграмму выходного сигнала и сравнить с заданной.

ВНИМАНИЕ: монтаж проводить при выключенном лабораторном макете.

5.3 Подготовить отчет, включив в него результаты проектирования, наблюдаемые осциллограммы, результаты измерений и ответы на контрольные вопросы.

6. КОНТРОЛЬНЫЕ ВОПРОСЫ

6.1. Какие способы применяются для минимизации логических функций?

6.2. Запишите выражение для логической функции, связывающей выходной и входные сигналы мультиплексора.

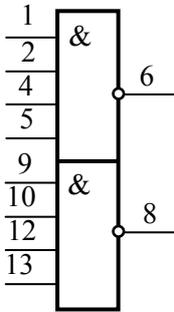
6.3. Запишите минимизированное выражение для булевой функции по карте Карно, представленной на рис.7, организовав блоки по единицам.

6.4. Чем отличаются комбинационные цифровые устройства от последовательностных? Приведите примеры микросхем цифровых устройств комбинационного типа.

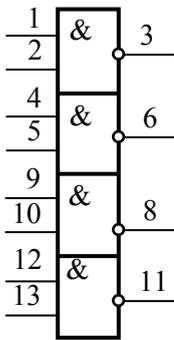
6.5. Сравните элементы ТТЛ, ТТЛШ и КМОП по быстродействию и экономичности.

Микросхемы ТТЛ к лабораторному макету

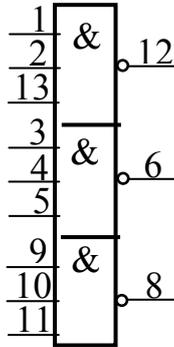
К555ЛА1



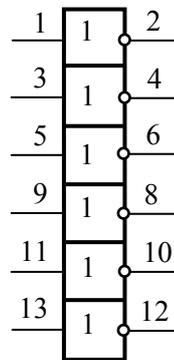
К555ЛА3



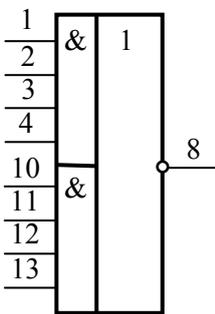
К155ЛА4



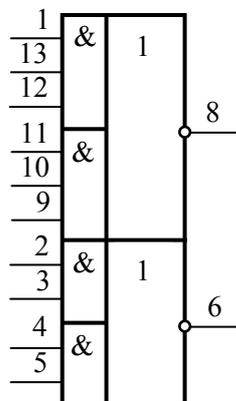
К555ЛН1



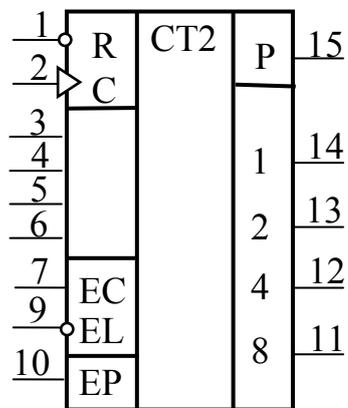
К555ЛР4



К555ЛР11



К555ИЕ18



Выходы питания логических элементов: 7 – "общий", 14 – "+5 В".

Выходы питания счетчика: 8 – "общий", 16 – "+5 В".

1. Информация по безопасности

Важная информация по безопасности.

- Максимальное напряжение между измерительным пробником и землей не должно превышать 30 В. Перед проведением измерений убедитесь, что цепь гальванически развязана с сетевым питанием. При необходимости используйте изолирующий трансформатор.
- Безопасными объектами являются:
- Приборы, работающие от батарей.
- Приборы, работающие через трансформаторы или адаптеры.

Небезопасными объектами являются:

- Приборы, подсоединенные непосредственно к сети.
- Приборы, содержащие компоненты, которые подсоединены непосредственно к сети. При проведении измерений на подобных приборах рекомендуется использовать изолирующий трансформатор.

Соблюдайте особую осторожность при проведении измерений непосредственно в сети; следует помнить, что "земля" обоих каналов взаимосвязана и соединена с землей компьютера!



Рис.1 Внешний вид приставки

2. Запуск программного обеспечения.

Открыть файл Pc-Lab2000.exe. В открывшейся заставке (рис.2), установить PCSU1000 в левом окне заставки, в правом – None. Кликнуть кнопку ОК. Откроется окно двухканального осциллографа PCSU1000 (рис.3)

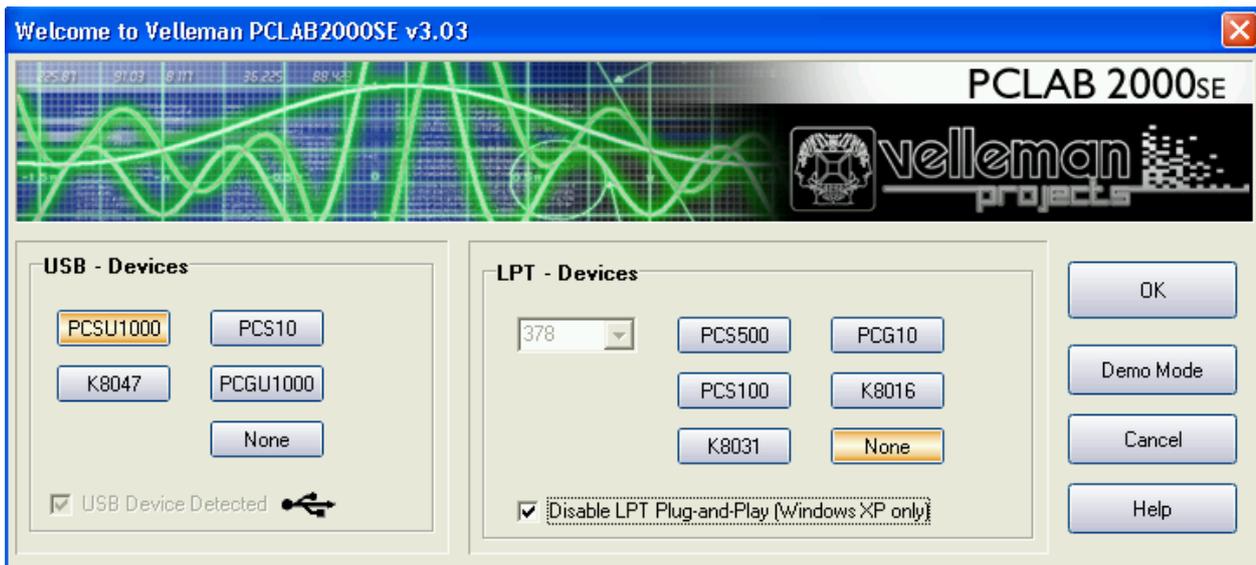


Рисунок 2.

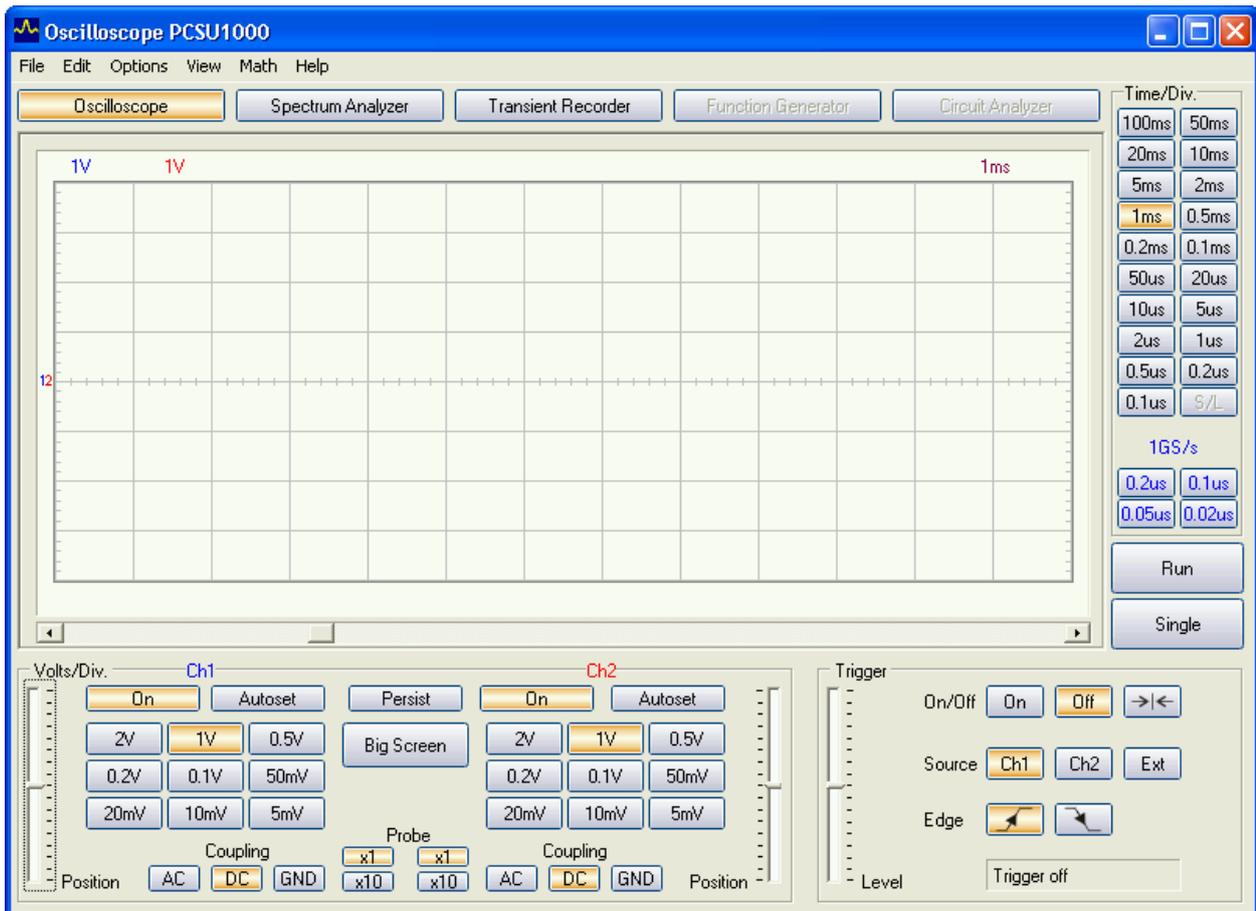


Рисунок 3.

3. Элементы управления. Режим осциллографа VOLTS/DIV

Выбранное значение показывает полный размах напряжения, необходимый для того, чтобы вызвать двойное отклонение в пределах одной единицы деления шкалы на экране.

Big Screen

Увеличение формы сигнала с отдельной полосой кнопок. Для возврата в обычный режим отображения информация выберите Normal Screen.

Coupling

AC: емкостная связь входного сигнала с входным усилителем/аттенюатором. Возможность измерения только переменных составляющих.

GND: нет входного сигнала, и входной усилитель/аттенюатор соединен с землей. Используйте это положение для определения нулевого уровня на дисплее.

DC: входной сигнал напрямую соединен с входным усилителем/аттенюатором. Возможность измерения как постоянного, так и переменного напряжения.

Probe x 1/x10

Используйте эти кнопки для адаптации показаний к установкам измерительных пробников x1 и x10, соответственно.

CH1 On, CH2 On

Изображение осциллограммы включается и выключается кнопками ON или OFF, Для того чтобы установить курсор в режим измерения значения напряжения на канале CH2 следует установить CH1 в положение OFF.

Autoset

Автоматическая установка таких параметров, как чувствительность по вертикали (Volts/div), чувствительность по горизонтали (Time/div) и уровень синхронизации для получения устойчивой осциллограммы в масштабе, удобном для просмотра. Синхронизация срабатывает, если амплитуда сигнала на экране более 0,5 деления.

Для правильной работы режима авто установки сигнал должен быть периодичным: амплитуда в диапазоне 5 мВ - 100 В, частота 50 Гц, рабочий цикл более 10%.

Persist

Режим наложения изображений. При нажатии данной кнопки осциллограф захватывает события сигнала и отображает их на экране. Включается режим наложения осциллограмм друг на друга при каждом запуске развертки. Этот режим используется для наблюдения за шумами в сигнале и т.п.

Используя данную функцию, пользователь может анализировать самые неблагоприятные варианты, такие как дрожание или шум. Также, данная функция может быть использована для определения локализации ошибок цифровых сигналов. Ошибки будут гарантированно фиксироваться даже при их единичном возникновении. Кроме того, функция Persist позволяет сравнивать известные и неизвестные схемы, Нажмите кнопку Single для захвата нескольких форм сигналов на одном дисплее.

TIME/DIV

Возможность изменения масштаба основного деления по оси времени путем установки различных временных значений. С помощью выбора различных значений TIME/DIV возможно масштабировать зафиксированный сигнал на экране.

TRIGGER On/Off

Можно выбрать режим синхронизации или автоколебаний.

TRIGGER Level

Выбирает уровень синхроимпульса, достаточный для запуска синхронизации. Установочная метка синхронизации показана горизонтальной линией с левой стороны экрана.

TRIGGER Channel

Выбирает источник синхроимпульса (CH1, CH2 или EXT).

TRIGGER Edge

Выбирает полярность режима синхронизации:

Стрелка вверх: Синхронизация происходит при возрастании величины уровня синхроимпульса.

Стрелка вниз: Синхронизация происходит при уменьшении величины уровня синхроимпульса.

>I<

Переустанавливает отправную точку начала синхронизации по оси X.

Установочная метка показана вертикальной линией внизу экрана.

RUN

Выбирает функцию постоянного обновления изображения (RUN).

Повторное нажатие кнопки фиксирует изображение.

SINGLE

Когда кнопка отжата и достигнут уровень начала синхронизации, обновление изображения происходит только один раз.

X-POSITION SCROLLBAR (под осциллограммой)

Устанавливает осциллограмму по горизонтали. Установочная метка показана вертикальной линией внизу экрана.

S/L (сглаживающая или линейная интерполяция)

Кнопка выбирает линейную (L) или сглаживающую (S) интерполяцию. В линейной интерполяции точки соединяются прямыми линиями. В сглаживающей интерполяции точки соединяются кривыми.

Сглаживающая интерполяция лучше показывает форму сигнала в области высших частот. Линейная интерполяция лучше для ступенчатых сигналов. Выбор S/L действует только при установках развертки по горизонтали (TIME/DIV) на 0,2 мкс и 0,1 мкс.

Примечание: сглаживающая интерполяция может привести к ошибочному отображению пиковых участков сигнала на частотах более 5 МГц.

Режим стробирования 1 GS/s

Данный режим 1GS/s действует только в диапазонах 0,2 мкс/дел, 0,1 мкс/дел, 0,05 мкс/дел и 0,02 мкс/дел.

CH1+CH2

CH1-CH2

XY Plot

INV.CH2

Данные кнопки появляются на дисплее только в математическом режиме измерений. Осциллограф позволяет переключаться между обычным и математическим режимом.

Совет: используйте колесико мышки для более точной настройки уровня синхронизации и положения по оси Y.

4. Возможные неисправности.

На дисплее осциллографа нет сигнала:

- отсутствует подключение к компьютеру (проверьте подключение кабеля к USB порту)
- если USB кабель подключен, закройте программу. Рассоедините устройства и подключите осциллограф повторно, запустите программу PC-Lab2000.
- Кнопка RUN находится в положении OFF.
- Рабочий канал находится в положении OFF.
- Параметр TIME/DIV выбран неправильно.
- Включен режим синхронизации, установите TRIGGER OFF.
- Вход осциллографа установлен на GND.
- Неправильно выбрано положение Y.
- Амплитуда входного сигнала слишком большая, отрегулируйте параметр VOLTS/DIV.

Если выше перечисленные способы не устранили неисправность, попробуйте подключить прибор к другому компьютеру или USB порту.

Примечания: перед рассоединением кабеля закройте программу.

Параметры могут выбираться не только для текущих сигналов, но и для ранее сохраненных осциллограмм.

Пользователь может повторно открыть сохраненную осциллограмму для проведения подобных измерений.

Примечание: при выводе на дисплей сохраненных форм сигналов не изменяйте

настройки осциллографа.

5. Добавление комментариев на дисплей

Для дополнительных разъяснений каждое измерение может сопровождаться комментарием. Данный текст может быть сохранен вместе с параметрами осциллограммы на диск.

Для ввода текста:

1. Щелкните правой кнопкой мыши на экране.
2. Откроется текстовое окно для записи комментария.
3. Нажмите кнопку "Add text on screen" (добавить текст на экран) или кнопку "Remove" (удалить текст) для удаления ранее записанного текста,
4. С помощью мышки разместите текст в требуемой стороне дисплея.
5. Нажмите кнопку "Close" (закрыть).

Если Вы хотите, чтобы текст сливался с фоном, выберите "Transparent text". Текст станет такого же цвета, как и вертикальные маркеры времени/частоты.

6. Меню программы

6.1, Меню File (Файл)

Примечание: при первом запуске программы по умолчанию будет сформирована папка DATA, куда будут сохраняться файлы с изображениями и данными.

Open Image (открыть изображение), открывает файл изображения и показывает его на экране.

Open DSO Data (открыть файл осциллограммы)

Открывает и показывает данные о форме сигнала, сохраненные в текстовом формате.

Save Image (сохранить изображение)

Сохраняет изображение в файл в формате Windows Bitmap (*.BMP).

Save DSO Data (сохранить осциллограмму)

Сохраняет осциллограмму сигнала в текстовом формате. Все собранные данные (4096 точек/канал) сохранены.

Save FFT Data (сохранить данные БПФ)

Сохраняет данные БПФ в текстовом формате. Только часть данных показанных на дисплее сохраняется (250 точек).

Save Settings (сохранение настроек)

Сохранение установок осциллографа, спектроанализатора или самописца в файл. Также, в файл сохраняются установки функционального генератора (частота, амплитуда, сдвиг и рабочий цикл).

Recall Settings (настройки сохраненных сигналов)

Загрузка файла с ранее сохраненными настройками.

Print (печать)

Распечатывает изображение в цвете. Вы можете редактировать подписи к изображению.

Print Setup (настройка печати)

Выбирает принтер и устанавливает опции принтера перед печатью.

Доступные опции зависят от выбранного принтера.

Exit (выход)

Завершает работу программы.

Calibrate и Exit (калибровка и выход)

Осуществляет калибровку осциллографа, сохраняет значения калибровки в файл Win DSO.INI и завершает работу программы. Эту опцию следует использовать, когда новый осциллограф проработал в течение 1 часа.

Эта опция выполняет следующие операции:

1. Прецизионная настройка координаты Y на шкалах Volts/Div и Time/Div.
2. Устанавливает осциллографические метки (на левой стороне экрана) таким образом, чтобы они соответствовали нулевому уровню.
3. Устанавливает метку уровня запуска таким образом, чтобы она совпадала с уровнем запуска.

6.2. Меню Edit (Редактирование)

Copy (копировать)

Копирует изображение в буфер Windows.

Paste (вставить)

Переносит изображение, находящееся в буфере, на экран.

Внешний вид лабораторного макета приведен на рис.1. Макет состоит из платы беспаячного монтажа и набора функциональных узлов. Общий провод функциональных узлов соединен с общим проводом внутреннего источника питания и монтажной платой.

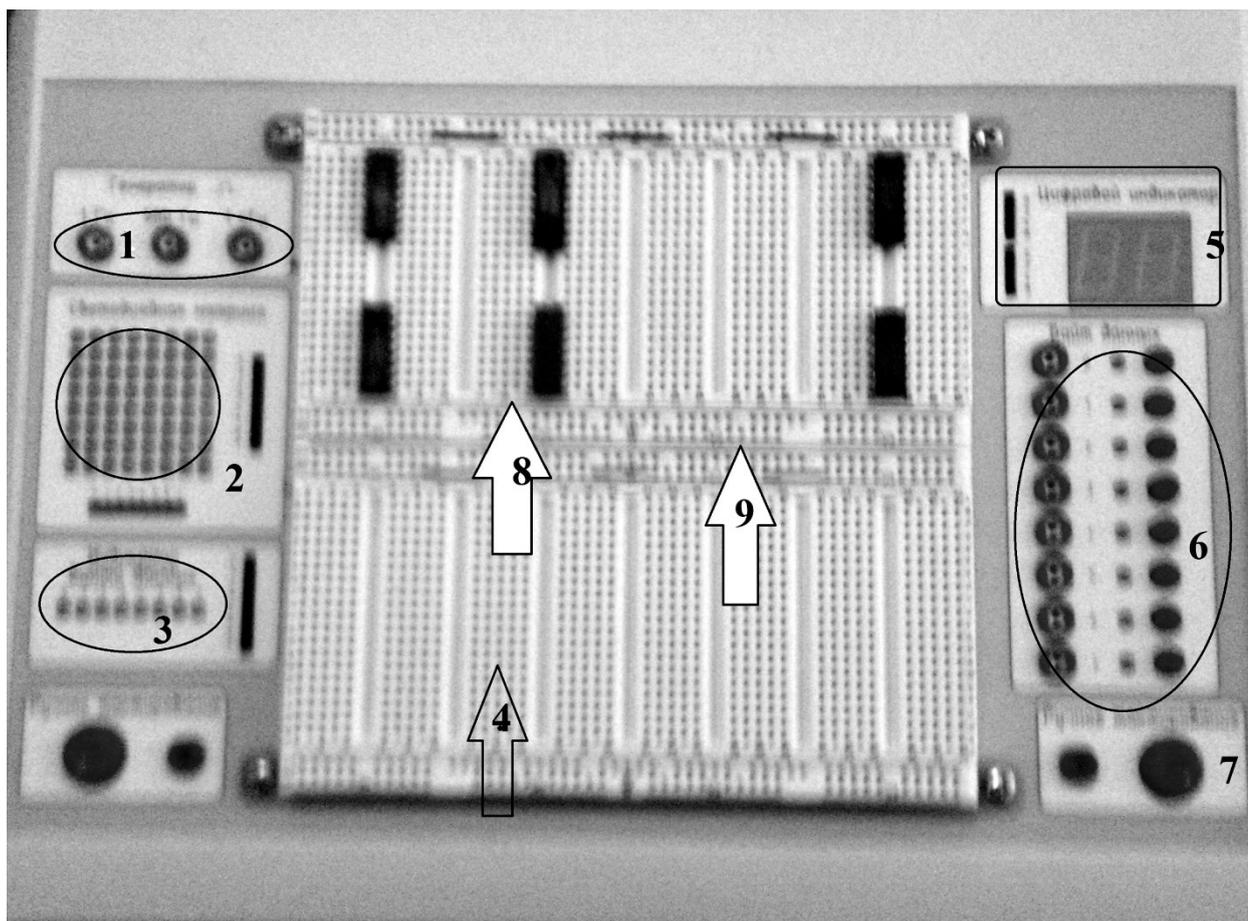


Рисунок 1.

На рис.1 обозначены:

1. Генератор тактовых импульсов (ГТИ), с фиксированными частотами – 1 Гц, 500 Гц, 1 кГц.
2. Матрица светодиодов.
3. Линейка светодиодов, для индикации байта.
4. Монтажная плата беспаячного монтажа. Стрелка указывает на отверстия одного потенциала.
5. Двухразрядный семисегментный индикатор. Индикатор управляется дешифратором, поэтому входы индикатора – входы дешифратора для подключения двоичного кода.
6. Выводы логических уровней, светодиод горит – логическая единица, нет – логический ноль.

7. Выход генератора одиночного импульса.

8,9. Выводы источника питания, синий – провод «общий», красный – «+5 В».

Включение макета осуществляется переключателем на задней панели макета.

Внимание! Монтаж, демонтаж или изменения в схеме на монтажной плате макета проводить при выключенном источнике питания.

2. Лабораторная работа №2

Проектирование комбинационных цифровых устройств

1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является приобретение навыков проектирования комбинационных цифровых устройств, как правило выпускаемых виде отдельных микросхем, практическая реализация таких устройств, экспериментальная проверка правильности их функционирования. Преподаватель назначает вариант задания (от 1 до 12), выполняемого каждой рабочей подгруппой. Схема устройства собирается на монтажном поле лабораторного макета соединительными проводниками.

2. ЛАБОРАТОРНАЯ УСТАНОВКА

Лабораторная установка включает лабораторный макет, приставку к персональному компьютеру PCSU1000, реализующая двухканальный виртуальный осциллограф на экране персонального компьютера, соединительные провода, комплект логических элементов ТТЛ.

3. ВАРИАНТЫ ЗАДАНИЙ

На рис.1 приведены условно-графические обозначения вариантов комбинационных цифровых устройств:

1. Вариант 1 – дешифратор.
2. Вариант 2 – шифратор.
3. Вариант 3 – мультиплексор.
4. Вариант 4 – демультиплексор.
5. Вариант 5 – компаратор для функции "равно".
6. Вариант 6 – компаратор для функции "больше".
7. Вариант 7 – компаратор для функции "меньше".
8. Вариант 8 – перемножитель.
9. Вариант 9 – вычитатель.
10. Вариант 10 – преобразователь кода.
11. Вариант 11 – двоичный сумматор.
12. Вариант 12 – алгебраический сумматор.

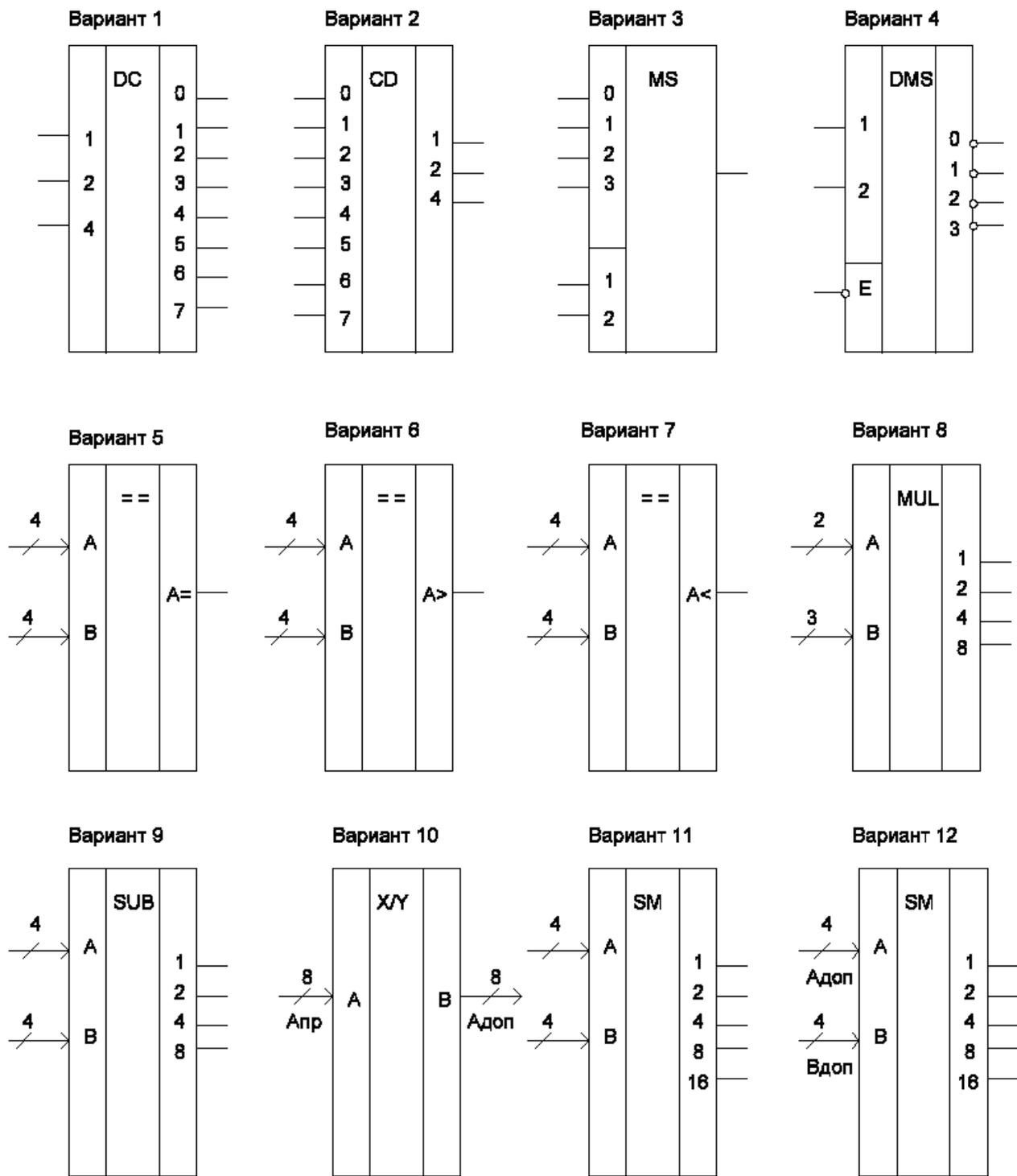


Рисунок 1.

4. ПРОГРАММА РАБОТЫ

4.1 По заданному варианту на имеющейся элементной базе (Приложение А) спроектировать комбинационное цифровое устройство.

4.2 Собрать устройство на макетном поле лабораторного макета. Разработать программу проведения эксперимента по его испытанию. Провести эксперимент и зафиксировать его результаты.

ВНИМАНИЕ: монтаж проводить при выключенном лабораторном макете.

4.3 Подготовить отчет, включив в него результаты проектирования и эксперимента, ответы на контрольные вопросы.

5. Контрольные вопросы

5.1 Дайте определение дешифратору, мультиплексору, сумматору, демультимплексору, цифровому компаратору.

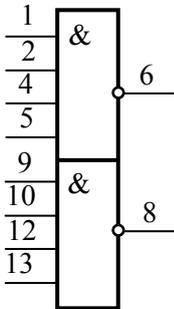
5.2 В чем вы видите достоинства представления чисел со знаком в дополнительном коде? Запишите результат сложения дополнительных кодов чисел плюс 80 и минус 33. Каждое число отображается байтом.

5.3 Какие коды может сравнивать цифровой компаратор? Запишите возможные варианты функций сравнения.

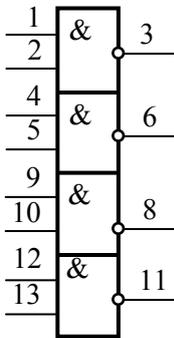
5.4 Как построить восьмиразрядный цифровой компаратор из двух микросхем К555СП1? Приведите схему.

Микросхемы к лабораторной работе

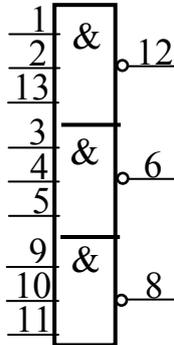
К555ЛА1



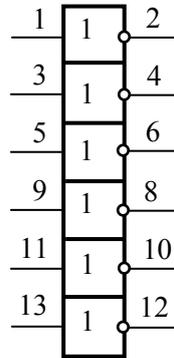
К555ЛА3



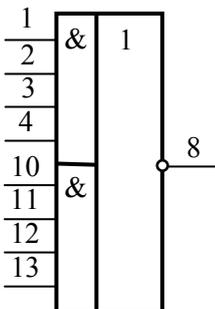
К155ЛА4



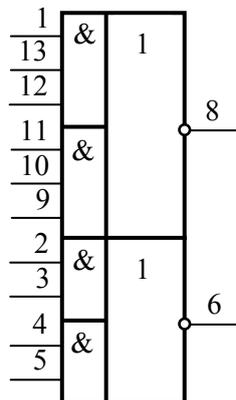
К555ЛН1



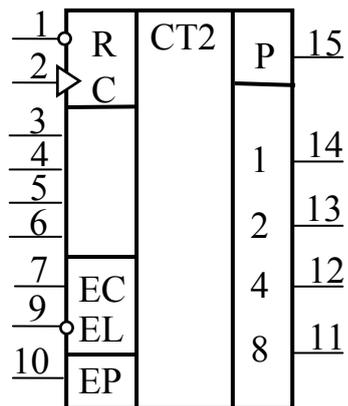
К555ЛР4

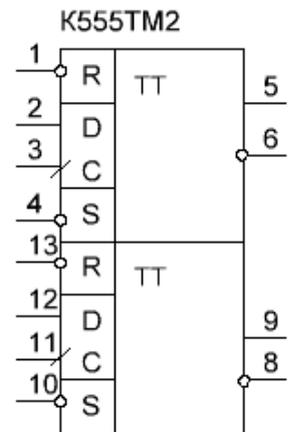
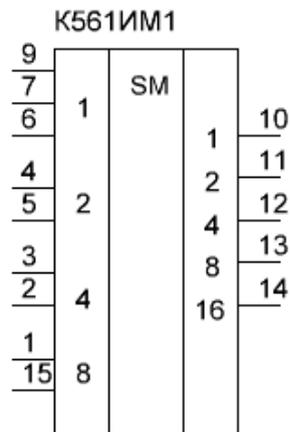
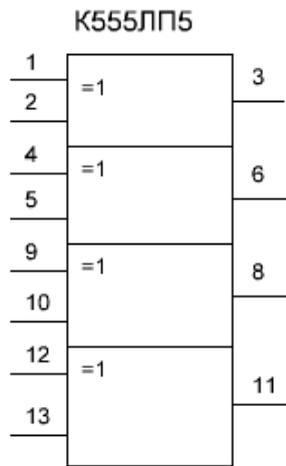


К555ЛР11



К555ИЕ18





Выходы питания микросхем ЛА1, ЛА3, ЛА4, ЛН1, ЛР4, ЛР11, ЛП5, ТМ2 :
7 – "общий", 14 – "+5 В".

Выходы питания микросхем ИЕ18, ИМ1: 8 – "общий", 16 – "+5 В".

3. Лабораторная работа №3

Проектирование синхронного счетчика

1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является приобретение навыков проектирования синхронного счетчика с заданной последовательностью чисел на выходе, практическая реализация счетчика, экспериментальная проверка правильности его функционирования. Преподаватель назначает вариант задания (от 1 до 28), выполняемого каждой рабочей подгруппой. Схема устройства собирается на монтажном поле лабораторного макета соединительными проводниками.

2. ЛАБОРАТОРНАЯ УСТАНОВКА

Лабораторная установка включает лабораторный макет, приставку к персональному компьютеру PCSU1000, реализующая двухканальный виртуальный осциллограф на экране персонального компьютера, соединительные провода, комплект микросхем (Приложение А).

3. ПРИМЕР ПРОЕКТИРОВАНИЯ СЧЕТЧИКА

Пусть стоит задача спроектировать вычитающий двоичный счетчик с коэффициентом пересчета $K_{сч} = 6$. Для фиксации шести состояний счетчика возьмем три тактируемых по положительному фронту D -триггера, объединенные входы синхронизации которых будем использовать как счетный вход счетчика (рис. 1). Будем характеризовать состояние счетчика N трехразрядным двоичным словом $Q_3Q_2Q_1$ (N должно циклически меняться от 5 до 0).

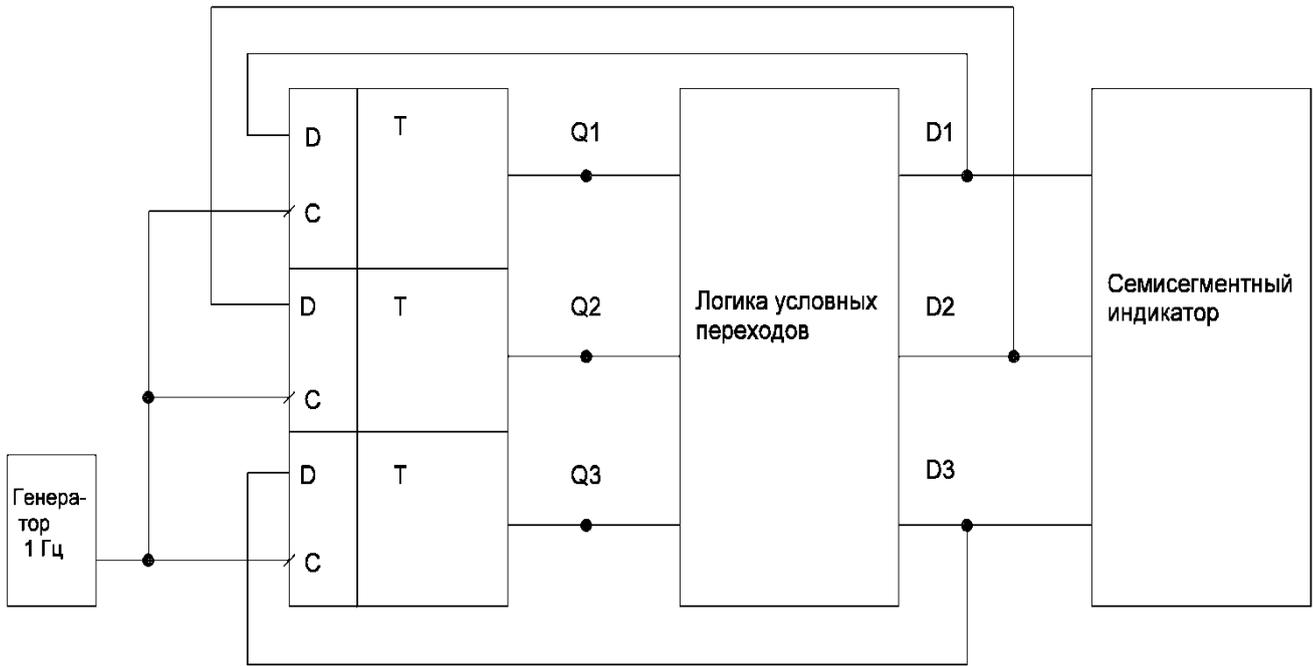


Рисунок 1 – Функциональная схема счетчика

Каждый импульс генератора переписывает на выходы триггеров Q_1 , Q_2 , Q_3 информацию с входов D_1 , D_2 , D_3 . Поэтому дальнейший синтез счетчика сводится к построению комбинационной схемы (логики переходов), формирующей из выходных сигналов Q_i уровни сигналов D_i на информационных входах триггеров, необходимые для перехода в следующее состояние. Для этого составим таблицу переходов (табл. 1) и запишем логические выражения для сигналов D_1 , D_2 , D_3 в ДНФ (дизъюнктивной нормальной форме):

$$D_1 = Q_3 \cdot \overline{Q_2} \cdot \overline{Q_1} + \overline{Q_3} \cdot Q_2 \cdot \overline{Q_1} + \overline{Q_3} \cdot \overline{Q_2} \cdot \overline{Q_1}; \quad (1)$$

$$D_2 = Q_3 \cdot \overline{Q_2} \cdot \overline{Q_1} + \overline{Q_3} \cdot Q_2 \cdot Q_1; \quad (2)$$

$$D_3 = Q_3 \cdot \overline{Q_2} \cdot Q_1 + \overline{Q_3} \cdot \overline{Q_2} \cdot \overline{Q_1}. \quad (3)$$

Для минимизации логических функций можно воспользоваться основными законами булевой алгебры или картами Карно (рис. 2), причем в клетках, соответствующих шестому и седьмому состоянию счетчика, логические функции можно доопределять по собственному усмотрению, так как в схеме проектируемого счетчика они не реализуются.

По картам Карно запишем минимизированные выражения для функций D_1 и D_2

$$D_1 = \overline{Q_1}; \quad (4)$$

$$D_2 = Q_3 \cdot \overline{Q_1} + Q_2 \cdot Q_1. \quad (5)$$

Выражение для D_3 получим путем преобразования соотношения (3), так как карта Карно не позволяет провести эффективную минимизацию этой функции

$$D_3 = \overline{Q_2} \cdot \overline{Q_1} \oplus Q_3. \quad (6)$$

С учетом соотношений (4), (5) и (6) построена схема эксперимента по изучению работы счетчика (рис. 3) в среде ASIMEC. Счетные импульсы снимаются с выхода тактового генератора G. Индикация состояний счетчика производится с помощью элемента DD9, выполняющего функцию преобразования четырехразрядного двоичного числа в его шестнадцатеричный эквивалент на семисегментном индикаторе. Удобно выбрать частоту генератора равной 1 Гц.

Таблица 1

Текущее состояние счетчика N				Последующее состояние счетчика N^+			
N	Q_3	Q_2	Q_1	N^+	D_3	D_2	D_1
5	1	0	1	4	1	0	0
4	1	0	0	3	0	1	1
3	0	1	1	2	0	1	0
2	0	1	0	1	0	0	1
1	0	0	1	0	0	0	0
0	0	0	0	5	1	0	1

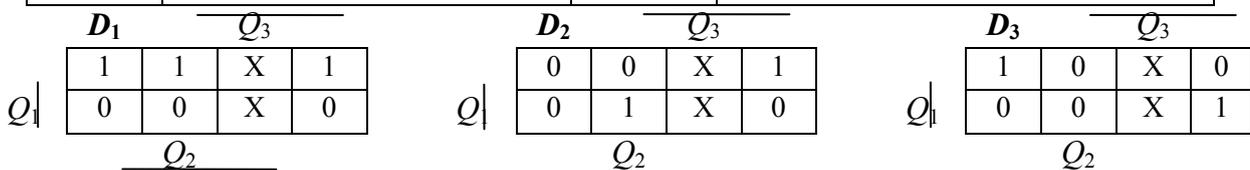


Рисунок 2 – Карты Карно для функций D_1, D_2, D_3

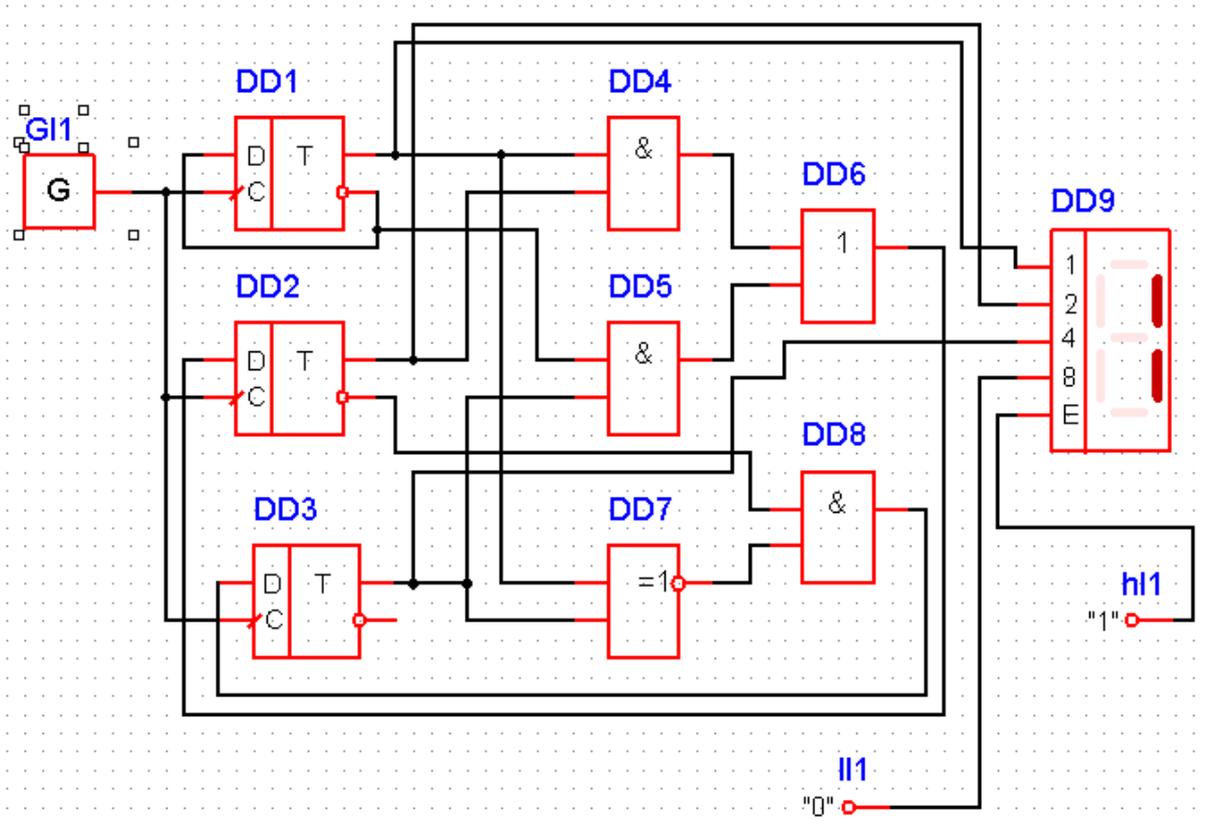


Рисунок 3 – Синхронный вычитающий счетчик с Ксч=6

3. ВАРИАНТЫ ЗАДАНИЙ

Таблица 2

Вариант	Состояния счетчика								
1	0	1	3	4	5	4	3	2	1
2	0	1	4	5	6	7	8	9	5
3	0	2	5	9	8	7	9	8	7
4	0	2	6	7	8	9	5	4	3
5	0	3	1	5	7	9	8	6	4
6	0	3	2	1	4	5	6	7	8
7	0	4	1	2	3	9	8	7	6
8	0	4	2	1	9	8	7	6	5
9	0	5	3	2	1	5	6	7	9
10	0	5	4	3	2	1	9	8	14
11	0	6	3	5	1	7	2	4	6

Вариант	Состояния счетчика								
12	0	6	4	9	2	5	3	7	8
13	0	7	5	8	3	4	6	8	9
14	0	7	6	7	4	3	7	8	9
15	0	8	1	5	5	2	7	9	3
16	0	8	2	3	6	1	7	9	3
17	0	9	3	8	7	6	5	4	2
18	0	9	4	7	8	6	5	3	2
19	0	1	5	6	9	7	8	3	1
20	0	2	6	5	4	8	7	3	2
21	0	6	7	4	5	9	5	2	1
22	0	1	8	3	4	3	5	1	2
23	0	2	1	3	5	7	9	8	7
24	0	1	2	4	6	8	6	7	9
25	0	2	3	4	6	7	9	2	9
26	0	3	4	7	9	2	3	5	8
27	0	4	5	7	9	8	6	4	2
28	0	4	3	5	8	6	7	3	5

4. ПРОГРАММА РАБОТЫ

4.1 По заданному варианту на имеющейся элементной базе спроектировать синхронный двоичный счетчик с коэффициентом счета 9.

4.2 Собрать счетчик на макетном поле лабораторного макета, подключить счетчик семисегментному индикатору. Провести эксперимент и зафиксировать его результаты.

ВНИМАНИЕ: монтаж проводить при выключенном лабораторном макете.

4.3 Подготовить отчет, включив в него результаты проектирования и эксперимента, ответы на контрольные вопросы.

5. КОНТРОЛЬНЫЕ ВОПРОСЫ

5.1 Дайте классификацию триггерных устройств.

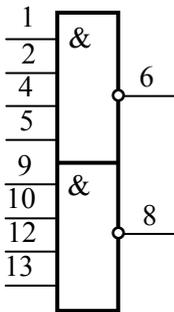
5.2 В чем отличие триггеров со статическим и динамическим тактовым входом?

5.3 Перечислите известные Вам области применения регистров памяти и сдвига.

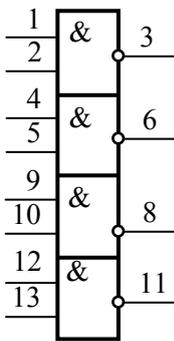
5.4 Приведите классификацию счетчиков.

Микросхемы к лабораторной работе

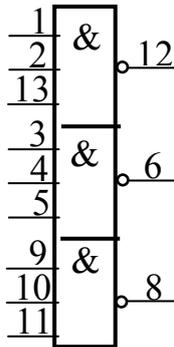
К555ЛА1



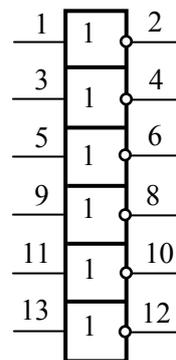
К555ЛА3



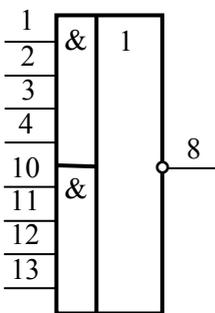
К155ЛА4



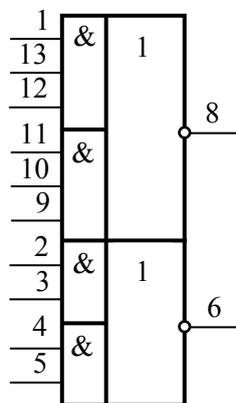
К555ЛН1



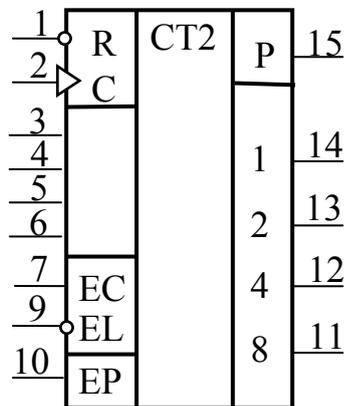
К555ЛР4

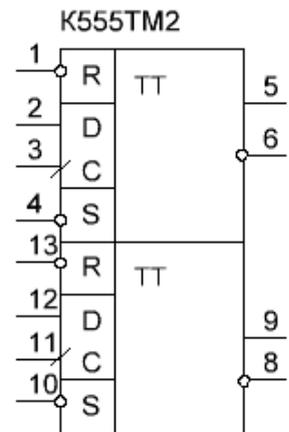
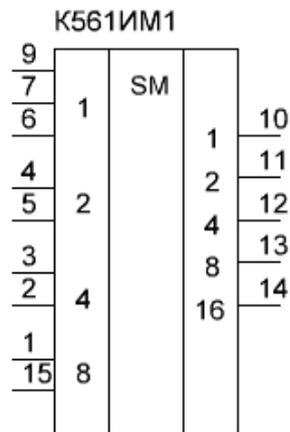
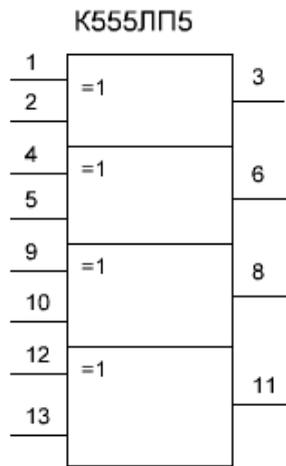


К555ЛР11



К555ИЕ18





Выходы питания микросхем ЛА1, ЛА3, ЛА4, ЛН1, ЛР4, ЛР11, ЛП5, ТМ2 :
7 – "общий", 14 – "+5 В".

Выходы питания микросхем ИЕ18, ИМ1: 8 – "общий", 16 – "+5 В".

4. Лабораторная работа №4

Программная модель и система команд МК51

1 Цель работы

Целью лабораторной работы является приобретение навыков программирования на языке Ассемблер и использования интегрированной среды разработки MSU 8051 IDE (<http://www.moravia-microsystems.com/mcu-8051-ide>, далее среда разработчика) при отладке прикладных программ для однокристальных микроконтроллеров семейства MCS-51.

2 Запись программы на языке ассемблера и ее трансляция

Язык ассемблера допускает представление всех элементов программы в символической (буквенно-цифровой) форме, отражающей их содержательный смысл. В качестве алфавита допустимых символов принят код ASCII (американский стандартный код для обмена информацией). Каждая строка ассемблера соответствует одной команде или псевдокоманде (директиве) и может содержать поля метки, мнемоники команды, операнда и комментария. При наличии в программе синтаксических ошибок ассемблер в процессе трансляции выдает сообщения об ошибках.

Метка ассоциируется с 16-битовым адресом той ячейки памяти, где будет размещен первый байт отмеченной команды. Использование меток освобождает программиста от необходимости оперировать абсолютными адресами памяти при записи команд передачи управления. Метка должна начинаться с буквы и заканчиваться двоеточием. Не допускается использовать в качестве меток мнемокоды команд, обозначения операндов и директив ассемблера. Символическое обозначение может появиться в поле метки только один раз.

Мнемокод команды может включать до четырех символов и вместе с обозначениями операндов образует группу ключевых слов ассемблера.

Поле операнда содержит числовые и символьные непосредственные данные, обозначения регистров и регистровых пар МК, адреса памяти. Возможно использование выражений, содержащих простейшие арифметические и логические операции (см. Таб 1). Операнд в виде строки символов, заключенной в апострофы, транслируется в последовательность кодов ASCII этих символов.

Поле комментария начинается с точки с запятой и полностью игнорируется ассемблером. В поле комментария фиксируется обычно

функция, которую выполняет группа команд в конкретной прикладной программе.

Кроме команд программа может содержать директивы ассемблера (или псевдокоманды). Директивы выполняются компилятором и дают программисту дополнительный сервис при компоновке прикладных программ. Список наиболее употребляемых директив приведен в Табл. 2. Полный список директив приведен в среде разработчика **Главное меню/Help/Handbook**.

Таблица 1

Допустимые выражения Ассемблера

Оператор	Выражение	Пример
Унарные операторы		
NOT	Отрицание (инверсия)	NOT 0A5H
LOW	Выделение младшего байта	LOW 0A5F5H
HIGH	Выделение старшего байта	HIGH 0A5F5H
Бинарные операторы		
+	Сложение	11+12
-	Вычитание	12-11
*	Умножение	11*12
/	Деление	15/3
AND	Логическое И	12 AND 11
OR	Логическое ИЛИ	12 OR 11
XOR	Исключающее ИЛИ	12 XOR 11
SHL	Логический сдвиг влево	12 SHL 2
SHR	Логический сдвиг Вправо	12 SHR 2

Таблица 2

Директивы Ассемблера

Директива	Наименование	Синтаксис	Пример
ORG	Начальный адрес массива	ORG <адрес>	ORG 2BH
END	Прекращение трансляции	END	END
DB	Определить байт	DB <данные>, <данные>	DB 23,'A', 23H, (22+15)
DW	Определить слово (два байта)	DW <данные>, <данные>	DW 2215, 0A22H

Окончание таблицы 2

Директива	Наименование	Синтаксис	Пример
EQU	Присвоение, эквиволентность	<символ> EQU выражение>	abc EQU R0 xyz EQU 4EH+12
SET	Определение переменной или регистра	<имя> SET <значение или регистр>	abc SET R1 xyz SET 2AH*2
BIT	Определение прямо адресуемого бита	<имя> BIT <значение>	abc BIT P4.2

В таблице 3 приведены правила записи констант в допустимых системах счисления. Строковые константы в памяти микроконтроллера хранятся в виде байта ASCII. Таблицу можно найти в среде разработчика **Главное меню/Utilites/ASCII Charts**. Например, при записи abc SET 'C' константе abc будет присвоено значение 43H.

Таблица 3

Система счисления	Допустимые символы	Пример записи
Двоичная	0,1	00110011b
Восьмеричная	0,1,2,3,4,5,6,7	321q
Десятичная	0,1,2,3,4,5,6,7,8,9	123
Шестнадцатеричная	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F	02Fh

3 Программа работы

1. Создать проект, имя проекта должно быть уникальным, набрано латинским шрифтом. Рекомендуется все программы по циклу лабораторных работ записывать в одном проекте. Определив номер варианта (от N=1 до N=12), создать программу TEST.ASM:

; Программа тестирования

ассемблера VAR EQU N

X EQU 512/2+(VAR))

Y SET LOW(X)

Z SET HIGH(X)

```
ABC SET    'A'
  ORG     0H
  MOV     R0, #Y
  MOV
  R1,    #Z
  CALL
  CHAR
  MOV
  R2,    A
  JMP     $
CHAR: MOV  A, (ABC+1)
  ADD    A, #VAR
  RET
  END
```

Написать комментарий к каждой строке программы. До выполнения программы зафиксировать значение указателя стека. Объяснить содержимое R0, R1, R2 после выполнения программы. С какого адреса располагается стек? Что находится в вершине стека? Как изменяется содержимое указателя стека при выполнении программы?

2. Проверить работу программы IND.asm. Программа считает количество нажатий кнопки в двоично-десятичном коде подключенной к P3.2 и выводит на двухразрядный семисегментный индикатор число нажатий кнопки.

```
org 0
  mov
  dptr, #0050h
  mov    a, #0h
m1:  jb   p3.2, $
     jnb  p3.2, $
     inc  a
     da   a
     call simvol
     jmp  m1
simvol: push acc
       mov   b, a
       anl  a, #0fh
       movc a, @a+dptr
       mov  p1, a
```

```

mov a, b
swap a
anl a, #0fh
movc a, @a+dptr
mov p2, a
pop acc
ret
org 50h
db 11000000b ; 0
db 11111001b ; 1
db 10100100b ; 2
db 10110000b ; 3
db 10011001b ; 4
db 10010010b ; 5
db 10000010b ; 6
db 11111000b ; 7
db 10000000b ; 8
db 10010000b ; 9
end

```

Семисегментные индикаторы подключить к портам P1 (единицы), P2 (десятки). На рис.1 показан пример подключения индикатора к P1. Зайдите в настройки индикатора, установите схему включения индикатора с общим анодом (Common electrode/Common anode), установите флажок "Поверх всех окон" (Window always on top).

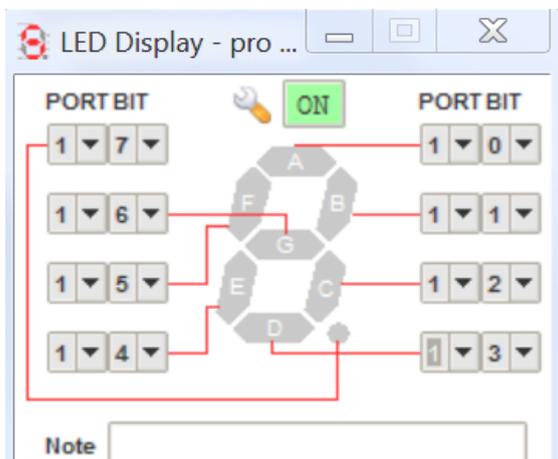


Рис.1

Написать комментарий к программе. Каким образом в программе определяется код выводимого символа на семисегментный индикатор? Заблокировать команду da a, овторить

эксперимент, объяснить результат.

3. Разработать и отладить на эмуляторе индивидуальное задание в соответствии со своим вариантом:

3.1 Установка 0 на линии P1.0 циклически изменяет состояние семисегментного индикатора от цифры 0 до цифры 5, фиксация на P1.0 уровня логической единицы останавливает счет.

3.2 Первое нажатие кнопки считывает двоично-десятичный код (числа 0...99) с параллельного порта, второе нажатие – выводит двоичный код на двухразрядный семисегментный индикатор.

3.3 Первое нажатие кнопки считывает двоичный код (числа 0...99) с параллельного порта, второе нажатие – выводит двоично-десятичный код на двухразрядный семисегментный индикатор.

3.4 На одnorазрядном семисегментном индикаторе проимитировать работу двоично-десятичного счетчика с коэффициентом пересчета 7.

3.5 На одnorазрядном семисегментном индикаторе проимитировать работу двоичного счетчика с коэффициентом пересчета 12.

3.6 Нажатие кнопки циклически выводит на семисегментный индикатор следующие символы: A, b, C, d, E, F.

3.7 Считать с параллельного порта байт данных (двоично-десятичный код), на одnorазрядный семисегментный индикатор вывести сумму тетрад, если результат больше числа 9, на индикатор вывести "0".

3.8 Считать с параллельного порта байт данных (двоично-десятичный код), на одnorазрядный семисегментный индикатор вывести разность тетрад, если результат отрицательный, на индикатор вывести символ "-".

3.9 Считать с параллельного порта байт данных (двоично-десятичный код), на одnorазрядный семисегментный индикатор вывести произведение тетрад, если результат больше числа 9, на индикатор вывести символ "П".

3.10 Считать с параллельного порта байт данных (двоично-десятичный код), на одноразрядный семисегментный индикатор вывести целую часть от деления тетрад, при делении на 0, на индикатор вывести символ "-".

3.11 Считать с параллельного порта байт данных (двоично-десятичный код), на одноразрядный семисегментный индикатор вывести остаток от деления тетрад, при отсутствии остатка, на индикатор вывести символ "0".

3.12 На одноразрядном семисегментном индикаторе проимитировать работу вычитающего двоично-десятичного счетчика с коэффициентом пересчета 7.

4 Контрольные вопросы

1. Чему равно содержимое регистров МК51 после системного сброса?
2. Чем отличаются друг от друга команды MOV R5,7 и MOV 5,#7?
3. Как выполняется команда MUL AB?
4. С помощью каких команд можно прочитать в регистр В информацию с датчиков, подключенных к линиям порта P1?

5 Содержание отчета

Отчет по лабораторной работе должен содержать: пункты Программы работы, исходные тексты программ, ответы на вопросы.

Система команд МК51

Список сокращений, символических имен и аббревиатур, примененных в таблице:

A	- регистр-аккумулятор
addr	- прямой 8-битный адрес передачи управления
addr11	прямой 11-битный адрес передачи управления
-addr16	прямой 16-битный адрес передачи управления
B	- регистр-расширитель аккумулятора
bit	- прямой 8-битный адрес бита
C	- признак переноса/заем
#data8	- 8- битный непосредственный операнд (константа)
#data16	- 16-битный непосредственный операнд (константа)
adr	- прямо адресуемый 8-битовый адрес ячейки внутренней памяти данных
sadr	- адрес приемника данных
ячейки	источника данных
dadr	- адрес ячейки
DPTR	- регистр-указатель данных
PC	- счетчик команд
Rn	- обобщенное имя регистра общего назначения (R0 - R7)
SP	- регистр/указатель стека
@Ri	- косвенно адресуемая ячейка памяти данных (0-256 байт)

@DPTR - косвенно адресуемая ячейка памяти данных (0-64 кБайта)

[i-j] - (i-j)-ые разряды операнда

<- - занести

<--> - поменять местами ами

Таблица 4

Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
Команды передачи данных				
MOV A,R0	E8	1	1	A ← R0
A,R1	E9	1	1	A ← R1
A,R2	EA	1	1	A ← R2
A,R3	EB	1	1	A ← R3
A,R4	EC	1	1	A ← R4
A,R5	ED	1	1	A ← R5
A,R6	EE	1	1	A ← R6
A,R7	EF	1	1	A ← R7
MOV A, adr	E5	2	1	A ← (adr)
MOV A,@R0	E6	1	1	A ← @R0
A,@R1	E7	1	1	A ← @R1
MOV A, #data8	74	2	1	A ← data8
MOV R0,A	F8	1	1	R0 ← A
R1,A	F9	1	1	R1 ← A
R2,A	FA	1	1	R2 ← A
R3,A	FB	1	1	R3 ← A
R4,A	FC	1	1	R4 ← A
R5,A	FD	1	1	R5 ← A
R6,A	FE	1	1	R6 ← A
R7,A	FF	1	1	R7 ← A
MOV R0, adr	A8	2	2	R0 ← (adr)
R1, adr	A9	2	2	R1 ← (adr)
R2, adr	AA	2	2	R2 ← (adr)
R3, adr	AB	2	2	R3 ← (adr)
R4, adr	AC	2	2	R4 ← (adr)
R5, adr	AD	2	2	R5 ← (adr)
R6, adr	AE	2	2	R6 ← (adr)
R7, adr	AF	2	2	R7 ← (adr)

Продолжение таблицы 4

Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
MOV R0,#data8	78	2	1	R0 <- data8
R1,#data8	79	2	1	R1 <- data8
R2,#data8	7A	2	1	R2 <- data8
R3,#data8	7B	2	1	R3 <- data8
R4,#data8	7C	2	1	R4 <- data8
R5,#data8	7D	2	1	R5 <- data8
R6,#data8	7E	2	1	R6 <- data8
R7,#data8	7F	2	1	R7 <- data8
MOV adr, A	F5	2	1	(adr) <- A
MOV adr ,R0	88	2	2	(adr) <- R0
adr ,R1	89	2	2	(adr) <- R1
adr ,R2	8A	2	2	(adr) <- R2
adr ,R3	8B	2	2	(adr) <- R3
adr ,R4	8C	2	2	(adr) <- R4
adr ,R5	8D	2	2	(adr) <- R5
adr ,R6	8E	2	2	(adr) <- R6
adr ,R7	8F	2	2	(adr) <- R7
MOV adr ,@R0	86	2	2	(adr) <- (R0)
adr ,@R1	87	2	2	(adr) <- (R1)
MOV adr ,#data8	84	2	2	(adr) <- #data8
MOV sadr, dadr	85	2	2	(sadr) <- (dadr)
MOV @R0,A	F6	1	1	(R0) <- A
@R1,A	F7	1	1	(R1) <- A
MOV @R0, adr	A6	2	1	(R0) <- (adr)
@R1, adr	A7	2	1	(R1) <- (adr)
MOV @R0, #data8	76	2	1	(R0) <- #data8
@R1, #data8	77	2	1	(R1) <- #data8
MOV DPTR,#data16	90	3	2	DPTR <- #data16

Продолжение таблицы 4

Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
Команды передачи байта данных из памяти программ				
MOVC	93	1	2	A ← (A+DPTR)
A,@A+PC	83	1	2	A ← (A+PC)
Команды передачи данных из внешней памяти данных				
MOVX A,@R0	E2	1	2	A ← (R0)
A,@R1	E3	1	2	A ← (R1)
MOVX A,@DPTR	E0	1	2	A ← (DPTR) расширенная ВПД
MOVX @R0,A	F2	1	2	(R0) ← A
@R1,A	F3	1	2	(R1) ← A
MOVX @DPTR,A	F0	1	2	(DPTR) ← A расширенная ВПД
Команды работы со стеком				
PUSH adr	C0	2	2	SP ← SP+1 (SP) ← (adr)
POP adr	B0	2	2	(adr) ← (SP) SP ← SP-1
Команды обмена данными				
XCH A,R0	C8	1	1	A ↔ R0
A,R1	C9	1	1	A ↔ R1
A,R2	CA	1	1	A ↔ R2
A,R3	CB	1	1	A ↔ R3
A,R4	CC	1	1	A ↔ R4
A,R5	CD	1	1	A ↔ R5
A,R6	CE	1	1	A ↔ R6
A,R7	CF	1	1	A ↔ R7
XCH A, adr	C5	2	1	A ↔ (adr)
XCH A,@R0	C6	1	1	A ↔ (R0)
A,@R1	C7	1	1	A ↔ (R1)
XCHD A,@R0	D6	1	1	A[0...3] ↔ (R0[0...3])
A,@R1	D7	1	1	A[0...3] ↔ (R1[0...3])

Продолжение таблицы 4

Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
Команды арифметических операций				
Сложение				
ADD A,R0	28	1	1	A← A+R0
A,R1	29	1	1	A← A+R1
A,R2	2A	1	1	A← A+R2
A,R3	2B	1	1	A← A+R3
A,R4	2C	1	1	A← A+R4
A,R5	2D	1	1	A← A+R5
A,R6	2E	1	1	A← A+R6
A,R7	2F	1	1	A← A+R7
ADD A, adr	25	2	1	A← A+(adr)
ADD A,@R0	26	1	1	A← A+(R0)
A,@R1	27	1	1	A← A+(R1)
ADD A,#data8	24	2	1	A← A+data8
ADDC A,R0	38	1	1	A← A+R0+C
A,R1	39	1	1	A← A+R1+C
A,R2	3A	1	1	A← A+R2+C
A,R3	3B	1	1	A← A+R3+C
A,R4	3C	1	1	A← A+R4+C
A,R5	3D	1	1	A← A+R5+C
A,R6	3E	1	1	A← A+R6+C
A,R7	3F	1	1	A← A+R7+C
ADDC A, adr	35	2	2	A← A+(adr)+C
ADDC A,@R0	36	1	1	A← A+(R0)+C
A,@R1	37	1	1	A← A+(R1)+C
ADDC A,#data8	34	2	2	A← A+data8+C
Десятичная коррекция				
DA A	E4	1	1	Если A[0-3]>9 или AC=1,
				то A ← A+06h, и далее

Продолжение таблицы 4

Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
				если A[4-7]>9 или C= I,
				то A <- A+60h
Вычитание				
SUBB A,R0	98	1	1	A<- A-R0-C
A,R1	99	1	1	A<- A-R1-C
A,R2	9A	1	1	A<- A-R2-C
A,R3	9B	1	1	A<- A-R3-C
A,R4	9C	1	1	A<- A-R4-C
A,R5	9D	1	1	A<- A-R5-C
A,R6	9E	1	1	A<- A-R6-C
A,R7	9F	1	1	A<- A-R7-C
SUBB A, adr	95	2	1	A<- A-(adr)-C
SUBB A,@R0	96	1	1	A<- A-(R0)-C
A,@R1	97	1	1	A<- A-(R1)-C
SUBB A, #data8	94	2	1	A<- A- #data8-C
Инкремент				
INC A	04	1	1	A <- A+1
INC R0	08	1	1	R0 <- R0+1
R1	09	1	1	R1 <- R1+1
R2	0A	1	1	R2 <- R2+1
R3	0B	1	1	R3 <- R3+1
R4	0C	1	1	R4 <- R4+1
R5	0D	1	1	R5 <- R5+1
R6	0E	1	1	R6 <- R6+1
R7	0F	1	1	R7 <- R7+1
INC adr	05	2	1	(adr) <- (adr)+1

Продолжение таблицы 4

Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
INC @RO	06	1	1	(RO) <- (RO)+1
@RI	07	1	1	(R1) <- (R1)+1
INC DPTR	A3	1	2	DPTR <- DPTR+1
Декремент				
DEC A	14	1	1	A <- A-1
DEC RO	18	1	1	RO <- RO-1
R1	19	1	1	R1 <- R1-1
R2	1A	1	1	R2 <- R2-1
R3	1B	1	1	R3 <- R3-1
R4	1C	1	1	R4 <- R4-1
R5	1D	1	1	R5 <- R5-1
R6	1E	1	1	R6 <- R6-1
DEC adr	15	2	1	(adr) <- (adr)-1
DEC @RO	16	2	1	(RO) <- (RO)-1
@R1	17	2	1	(R1) <- (R1)-1
Умножение				
MUL AB	A4	1	4	A <- мл. байт A*B B <- ст. байт A*B
Деление				
DIV AB	84	1	4	A <- A:B A - частное B - остаток
Команды логических операций				
Операция "И"				
ANL A,RO	58	1	1	A <- A & RO
A,R1	59	1	1	A <- A & R1

Продолжение таблицы 4

Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
A,R2	5A	1	1	A <-A & R2
A,R3	5B	1	1	A <-A & R3
A,R4	5C	1	1	A <-A & R4
A,R5	5D	1	1	A <-A & R5
A,R6	5E	1	1	A <-A & R6
A,R7	5F	1	1	A <-A & R7
ANL A, adr	55	2	1	A <-A & (adr)
ANL A,@R0	56	1	1	A <-A & (R0)
A,@R1	57	1	1	A <-A & (R1)
ANL A,#data8	54	2	1	A <-A & data8
ANL adr,A	52	2	1	(adr) <- (adr) & A
ANL adr ,#data8	53	3	2	(adr) <- (drc) & #data8
Операция "ИЛИ"				
ORL A,R0	48	1	1	A <-A or R0
A,R1	49	1	1	A <-A or R1
A,R2	4A	1	1	A <-A or R2
A,R3	4B	1	1	A <-A or R3
A,R4	4C	1	1	A <-A or R4
A,R5	4D	1	1	A <-A or R5
A,R6	4E	1	1	A <-A or R6
A,R7	4F	1	1	A <-A or R7
ORL A, adr	45	2	1	A <-A or (adr)
ORL A,@R0	46	1	1	A <-A or (R0)
A,@R1	47	1	1	A <-A or (R1)
ORL A ,#data8	44	2	1	A <-A or data8
ORL adr, A	42	2	1	(adr) <-(adr) or A
ORL adr, #data8	43	2	2	(adr) <-(drc) or #data8
Операция "Исключающее ИЛИ"				
XRL A,R0	68	1	1	A <- A xor R0

Продолжение таблицы 4

Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
A,R1	69	1	1	A ← A xor R1
A,R2	6A	1	1	A ← A xor R2
A,R3	6B	1	1	A ← A xor R3
A,R4	6C	1	1	A ← A xor R4
A,R5	6D	1	1	A ← A xor R5
A,R6	6E	1	1	A ← A xor R6
A,R7	6F	1	1	A ← A xor R7
XRL A, adr	65	2	1	A ← A xor (adr)
XRL A,@R0	66	1	1	A ← A xor (R0)
A,@R1	67	1	1	A ← A xor (R1)
XRL A,#data8	64	2	1	A ← A xor data8
XRL adr, A	62	2	1	(adr) ← (adr) xor A
XRL adr, #data8	63	3	2	(adr) ← (drc) xor #data8
Команды логических операций. Команды сдвигов				
RL A	23	1	1	A[n+1] ← A[n], A[0] ← A[7] и C ← A[7],
RLC A	33	1	1	A[n+1] ← A[n], A[0] ← C, C ← A[7]
RR A	03	1	1	A[n] ← A[n+1], A[7] ← A[0] и C ← A[0]
RRC A	13	1	1	A[n] ← A[n+1], A[7] ← C, C ← A[0]
CLR A	E4	1	1	A ← 0
CPL A	F4	1	1	A ← A
SWAP A	C4	1	1	A[0...3] ← A[4...7] A[4...7] ← A[0...3]
Команды операций с битами				
CLR C	C3	1	1	C ← 0
CLR bit	C2	2	1	bit ← 0

Продолжение таблицы 4

Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
SETB C	B3	1	1	C ← 1
SETB bit	B2	2	1	bit ← 1
CPL C	B3	1	1	C ← C
CPL bit	B2	2	1	bit ← bit
ANL C, bit	82	2	2	C ← C & bit
ANL C, bit	B0	2	2	C ← C & bit
ORL C, bit	72	2	2	C ← C or bit
ORL C, bit	A0	2	2	C ← C or bit
MOV C, bit	A2	2	1	C ← bit
MOV bit, C	97	2	1	bit ← C
Холостая команда				
NOP	00	1	1	PC ← PC+1
Команды передачи управления. Безусловный переход				
LJMP addr16	02	3	2	PC ← addr16
AJMP addr11	01	2	2	
	21	2	2	
двоичный код команды a10a9a8 00001 a7.....a0	41	2	2	
	61	2	2	
	81	2	2	
	A1	2	2	
	C1	2	2	

Продолжение таблицы 4

Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
	E1	2	2	
SJMP addr	80	2	2	PC ← PC+addr
JMP @A+DPTR	73	1	2	PC ← (A+DPTR)
Условный переход				
JZ addr	60	2	2	PC ← PC+2; если A=0, то PC ← PC+addr
JNZ addr	70	2	2	PC ← PC+2; если A≠0, то PC ← PC+addr
JC addr	40	2	2	PC ← PC+2; если C=1, то PC ← PC+addr
JNC addr	50	2	2	PC ← PC+2; если C=0, то PC ← PC+addr
JB bit, addr	20	3	2	PC ← PC+3; если bit=1, то PC ← PC+addr
JNB bit, addr	30	3	2	PC ← PC+3; если bit=0, то PC ← PC+addr
JBC bit, addr	10	3	2	PC ← PC+3; если bit=1, то bit ← 0 PC ← PC+addr
DJNZ R0,addr	D8	2	2	PC ← PC+2;
R1,addr	D9	2	2	Rn ← Rn-1,
R2,addr	DA	2	2	и если Rn=0,
R3,addr	DB	2	2	то PC ← PC+addr
R4,addr	DC	2	2	
R5,addr	DD	2	2	
R6,addr	DE	2	2	

Продолжение таблицы 4

Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
DJNZ adr, addr	D5	2	2	PC <- PC+2;
				(adr) <- (adr)-1, и если (adr)=0, то PC <- PC+addr
Команды передачи управления				
Условный переход				
CJNE A, #data8, addr	B4	3	2	PC <- PC+3; а если #data8 ≠ A, то PC <- PC+addr
CJNE A, adr, addr	B5	3	2	PC <- PC+3; а если (adr) ≠ A, то PC <- PC+addr
CJNE R0, #data8, addr	B8	3	2	PC <- PC+3;
RI, #data8, addr	B9	3	2	а если Rn ≠ A,
R2, #data8, addr	BA	3	2	то PC <- PC+addr
R3, #data8, addr	BB	3	2	
R4, #data8, addr	BC	3	2	
R5, #data8, addr	BD	3	2	
R6, #data8, addr	BE	3	2	
R7, #data8, addr	BF	3	2	
CJNE @R0, #data8, addr	B6	3	2	PC <- PC+3; а если @Ri ≠ data8,
@R1, #data8, addr	B7	3	2	то PC <- PC+addr
Команды вызова подпрограмм				
LCALL addr16	12	3	2	PC <- PC+3, (SP) <- PC[0...7], (SP+1) <- PC [8... 15], SP <- SP+2, PC <- addr16,
ACALL addr11	11	2	2	PC <- PC+2,

Окончание таблицы 4

Мнемоника	Код команд	Кол-во байт	Кол-во циклов	Описание команд
	31	2	2	(SP) <- PC[0...7],
двоичный код	51	2	2	(SP+1) <- PC[8. ..15],
команды	71	2	2	SP <- SP+2,
	91	2	2	PC[0. ..10] <- addr11
a10a9a8 10001	B1	2	2	
a7..... a0	DI	2	2	
	FI	2	2	
Команды возврата из подпрограмм				
RET	22	1	2	PC[8...15] <- (SP)
				PC[0..7] <- (SP-1)
				SP <- SP-2
RETI	32	1	2	PC[8...15] <- (SP)
				PC[0..7] <- (SP-1)
				SP <- SP-2

5. Лабораторная работа №5

Управление устройствами ввода/вывода с помощью параллельных портов МК51

1 Цель работы

Целью лабораторной работы является изучение особенности работы параллельных портов МК51 приобретение навыков работы с матричной клавиатурой 4*4 и светодиодной матрицей 8*8.

2 Устройства ввода вывода

Для реализации взаимодействия пользователя с микроконтроллером используют различные устройства ввода-вывода информации. В самом простом случае в роли устройства ввода может выступать кнопка, представляющая собой элементарный механизм, осуществляющий замыкание-размыкание контактов. Схема подключения кнопки к линии ввода параллельного порта ввода микроконтроллера показана на рисунке 1. Когда контакты кнопки S1 разомкнуты, то через резистор R1 на входе микроконтроллера фиксируется логический уровень "1", когда контакты замкнуты, то вход оказывается соединенным с общим проводом, что соответствует логическому уровню "0". Если параллельный порт микроконтроллера имеет встроенный генератор тока, то в схеме можно обойтись без резистора R1.

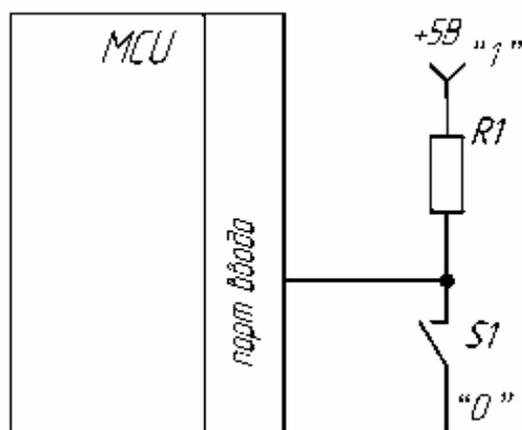


Рис. 1 Подключение кнопки к параллельному порту

Недостаток данного подключения заключается в том, что для каждой кнопки отдельный вывод параллельного порта. Для уменьшения количества линий ввода-вывода микроконтроллера используется клавиатура, представляющая собой двухмерную матрицу кнопок, организованных в ряды и столбцы.

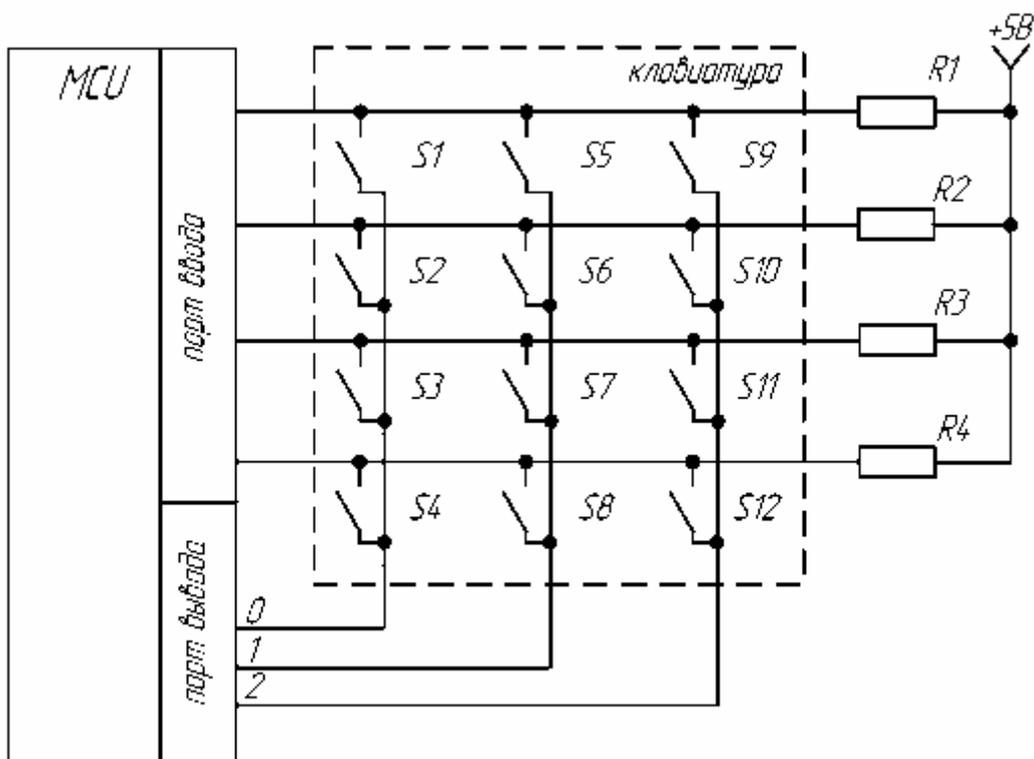


Рис.2 Матричная клавиатура 4*3

Подключение клавиатуры отличается от схемы подключения одиночной кнопки тем, что потенциал общего провода на опрашиваемые кнопки подается не непосредственно, а через порт вывода. В каждый момент времени сигнал низкого уровня (логический ноль) подается только на один столбец кнопок, на остальные должна подаваться логическая единица, что исключает неоднозначность определения номера нажатой кнопки. Двоичные сигналы, присутствующие при этом на строках клавиатуры, считываются через порт ввода микроконтроллера. На рис.3 показаны временные диаграммы порта вывода.

Опрос клавиатуры заключается в последовательном сканировании каждого столбца, для этого на соответствующую линию порта вывода подается логический ноль (эквивалент общего провода), на остальных столбцах должен быть высокий уровень, после чего с порта ввода, к которому подключены строки, считывается код. Если считаны все единицы, то ни одна из клавиш не нажата, в противном случае код содержит информацию о нажатых клавишах

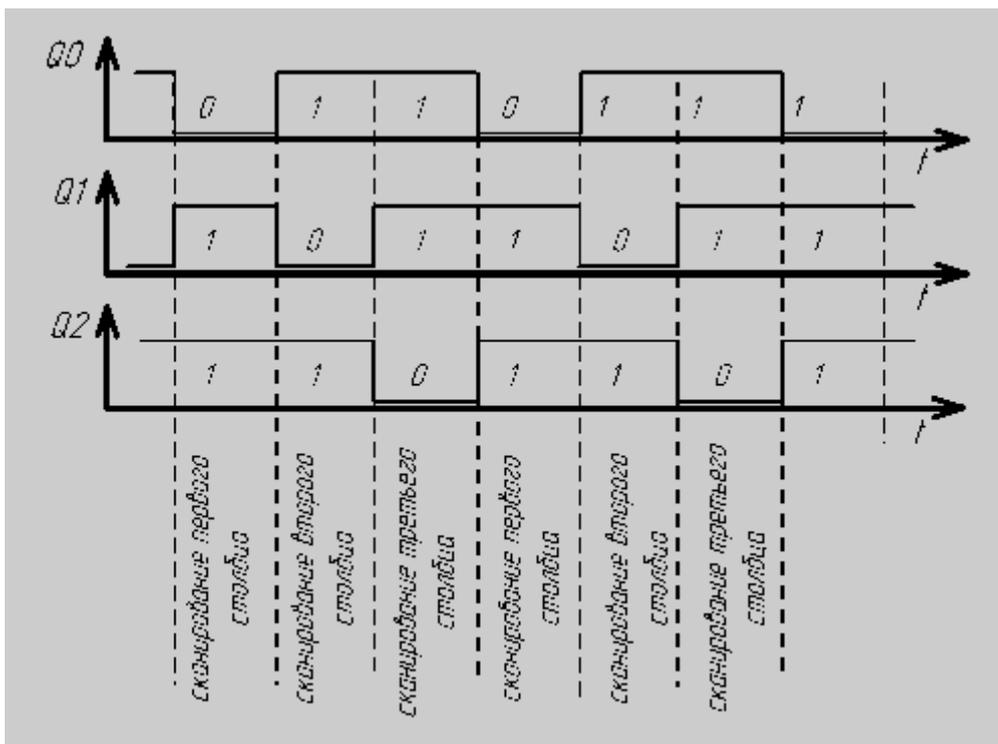


Рис.3 Временные диаграммы порта вывода

Для визуализации информации в микроконтроллерных устройствах зачастую используют матричные индикаторы. Матричный индикатор — устройство отображения информации, элементы отображения (светодиоды зачастую называют - пиксель) которого сгруппированы по строкам и столбцам. Матричный индикатор относится к знаковосинтезирующим цифро-буквенным индикаторам и предназначен для отображения информации в виде букв, цифр, математических и специальных знаков, знаков препинания, мнемонических символов. Матричным индикатором считается устройство, объединенное в законченном конструктиве - корпусе. В отличие от мониторов, дисплеев или экранов, матричным индикатором принято считать устройство с относительно небольшим количеством пикселей, или устройство, предназначенное для вывода одного или нескольких символов, хотя граница довольно размыта.

Как правило, матричный индикатор имеет два и более рядов и два и более столбцов однотипных элементов отображения (точек, пикселей) с индивидуальным управлением. Практическое применение имеют матричные индикаторы 5 x 7, 5 x 8, 8 x 8 и более пикселей. Форма пикселя обычно - круглая, но встречаются квадратные, а также структурированные пиксели (см. рис.4). Цвет пикселя определяется используемой технологией изготовления индикатора. Выпускаются монохромные (одноцветные) и многоцветные индикаторы

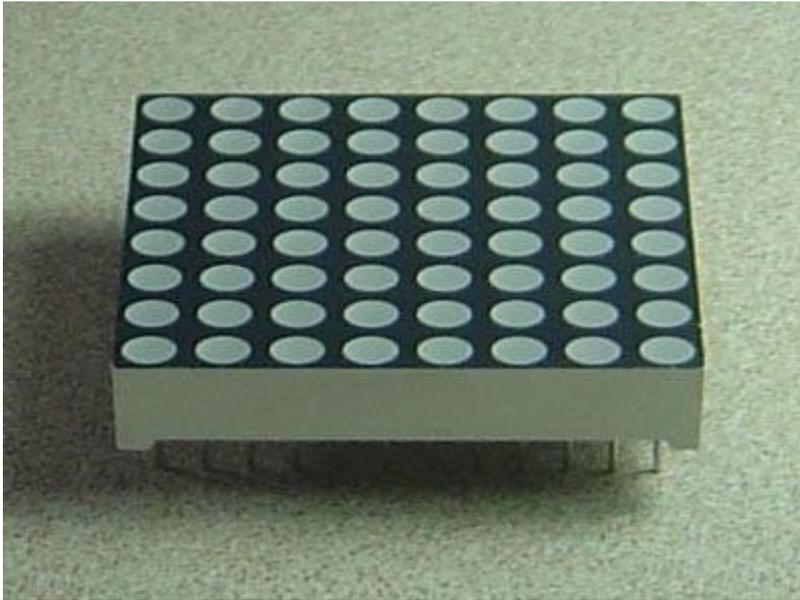


Рис. 4 Светодиодная матрица 8*8

Корпус светодиодного матричного индикатора изготавливается из пластика, внутри - печатная плата с установленными светодиодами, чаще SMD типа. Корпус полностью залит компаундом. Пиксели оформлены в виде окошек, заполненных полупрозрачным светорассеивающим материалом. Большинство матричных индикаторов изготавливается таким образом, чтобы можно было установить их рядом друг с другом, с получением непрерывного графического поля.

Матричный индикатор должен позволять управление каждым пикселем индикатора. Применяется два распространенных метода управления - статический и динамический (мультиплексный, растровый).

При статическом способе используется непосредственное управление каждым пикселем матрицы. Каждый элемент изображения имеет ячейку памяти и светодиодный драйвер. Такая схема включения используется в матрицах с большим размером пикселя, когда мощность и стоимость драйвера значительна. Кроме того, статическая схема включения применяется в устройствах с высокими требованиями к электромагнитному излучению.

Для упрощения схемы управления, а также, для снижения количества выводов индикатора, используется динамический способ управления. Динамический способ подразумевает поочередное включение различных групп элементов отображения со скоростью, превышающей время реакции человеческого глаза (частота переключений более 24 Гц.). Несмотря на то, что изображение на индикаторе при таком способе управления в каждый момент времени неполное, глаз человека интегрирует его и видит целостную картинку.

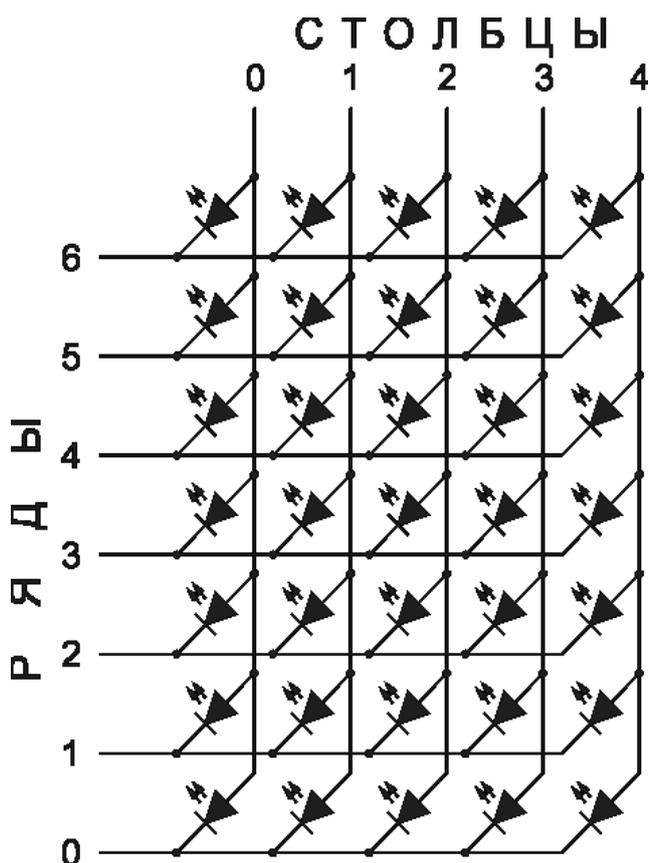


Рис. 5 Светодиодный матричный индикатор 5*7

Рассмотрим, для примера, светодиодный монохромный матричный индикатор 5 *7 пикселей (рис. 5). Для реализации динамического способа управления, все светодиоды в рядах (Row в среде разработчика) объединяются по катодам, а в столбцах (Column в среде разработчика) - по анодам. Видна экономия выводов индикатора - вместо 70 выводов получилось 12. На аноды последовательно подаются положительные периоды напряжения (фазы), на катоды подается отрицательный сигнал (код). Для формирования полного изображения, необходимо для каждого столбца (фазы) последовательно установить соответствующий скан-код. Так для индикатора на пять столбцов по семь пикселей в столбце, необходимо установить для каждой из пяти фаз код из семи линий (бит). Весь цикл обработки индикатора должен уложиться в 20 мс (50 Гц) или менее, для того, чтобы картинка на индикаторе была устойчивой и дрожание изображения было незаметным.

Матричные светодиодные индикаторы применяются в виде единичных символьных индикаторов, либо в виде непрерывных графических полей различной размерности. Матричные индикаторы с размерами 8 x 8 пикселей применяются для изготовления непрерывных

графических экранов или табло бегущая строка. Выбор размерности 8 x 8 пикселей обусловлен размером байта и номенклатурой выпускаемых драйверов светодиодов. На таких экранах возможно отображение графических картинок, многострочной символьной информации. А при использовании полноцветных индикаторов, создаются графические видео экраны - уличные телевизоры.

Нагрузочная способность портов ввода-вывода портов микроконтроллера – один ТТЛ вход (для порта P0 – два ТТЛ входа). Поэтому для обеспечения рабочего тока свечения светодиода (10 мА.) необходимы буферные элементы в столбцах и строках матрицы. В лабораторной работе буферные элементы не подключаются.

3 Программа работы

1. Проверить работу программы KEY.asm. Данная программа позволяет определять какой символ набран на матричной клавиатуре 4*4 и вывести на семисегментный индикатор соответствующей символ. Индикация статическая. Коды цифр для индикатора находятся в таблице TABLE. Программа определяет номер нажатой кнопки, которая является сдвигом относительно базового адреса таблицы TABLE при определении соответствующего кода индикатора.

Номер нажатой кнопки определяется умножением на 4 номера строки нажатой кнопки и суммированием с номером столбца нажатой кнопки. Подключить матричную клавиатуру (см. рис.4) и семисегментный индикатор (порт P3).

```

COLUMN EQU 50h           ;Адреса для хранения номера
строки ROW EQU 51h       ;и столбца нажатой кнопки
VAL EQU 52h              ;Адрес для хранения номера нажатой
DEL EQU 05h              ;Задание длительности задержки
TABL EQU 90h             ; Таблица символов
org 0000h
mov DPTR, #TABL
mov P2, #0FFh           ;начальная установка порта P2
mov P2, #0F7h           ;Запись в порт P2 числа
11110111b
mov R7, #0F7h           ;Запись в регистр R7 числа
11110111b
mov P3, #0ffh
L1: mov A, R7            ;Запись R7 в
A rl A                  ;Сдвиг A
влево mov R7, A         ;Запись A в
R7

```

```

orl      A, #00Fh    ;Запись 1 в мл. тетраду A, для
;настройки мл. тетрады P2 на ввод.
mov      P2, A
;Сканирование линии 0
jb       P2.0, L2
lcall
        DELA
Y jb     P2.0,
L2
mov      COLUMN, #00h ;Запись номера линии 0000000b
ljmp S1
;Сканирование линии
1 L2:
jb       P2.1, L3
lcall
        DELA
Y jb     P2.1,
L3
mov      COLUMN, #01h
ljmp S1
; Сканирование линии 2
L3:
jb       P2.2, L4
lcall
        DELA
Y jb     P2.2,
L4
mov      COLUMN, #02h
ljmp S1

; Сканирование линии 3
L4:
jb       P2.3, L1
lcall
        DELA
Y jb     P2.3,
L1
mov      COLUMN, #03h
ljmp S1

; Сканирование столбца 0
S1:
jb       P2.4, S2

```

```
mov     ROW,  
#00h ljmp CALC
```

```
; Сканирование столбца 1 S2:  
jb      P2.5, S3  
mov     ROW,  
#01h ljmp CALC
```

```
; Сканирование столбца 2  
S3:  
jb      P2.6, S4  
mov     ROW,  
#02h ljmp CALC
```

```
; Сканирование столбца 3  
S4:  
jb      P2.7, L1  
mov     ROW,  
#03h ljmp CALC
```

```
;Вычисление позиции в таблице и передача информации на индикатор  
CALC:
```

```
mov     A, COLUMN    ;Запись номера линии в A  
mov     B, #04h      ;Запись в B числа 4h  
mul     AB            ;Умножение A на B  
ADD A, ROW           ;Суммирование результата с номером  
;нажатой кнопки  
inc     a  
mov     VAL, A        ;Запись номера нажатой кнопки в  
VAL movc A, @A+DPTR  ;Запись в A кода нажатой кнопки  
mov     P3, A        ;запись в порт P3 для индикации  
кнопки ljmp L1
```

```
;Временная  
задержка DELAY:  
mov R4, #DEL  
D1: djnz R4, D1  
ret
```

```
;Таблица кодов символов ORG  
90h  
db 11000000b ; 0  
db 11111001b ; 1  
db 10100100b ; 2
```

```

db 10110000b ; 3
db 10001000b ; A
db 10011001b ; 4
db 10010010b ; 5
db 10000010b ; 6
db      10000011b ; b
db      11111000b ; 7
db      10000000b ; 8
db      10010000b ; 9
db      11000110b ; C
db      01111111b ; .
db      11000000b ; 0
db      10011100b ; #
db      10100001b ; d

```

end



Рис. 4 Подключение матричной клавиатуры к МК

Для устранения "дребезга контактов" в программе предусмотрена временная задержка. Поясните термин "дребезг контактов". При работе симулятора временную задержку можно отключить.

В какой ячейке памяти данных содержится номер кнопки клавиатуры ? Объясните содержимое этой ячейки при нажатии символа "А" на клавиатуре.

2. Проверить работу программы LED.asm. Программа в режиме динамической индикации выводит на светодиодную матрицу 8*8 символ X.

```

    COLUMN EQU 01      ;Начальный номер столбца матрицы
    ROW EQU 0          ;начальный номер ряда матрицы
    TABL EQU 50h       ; адрес таблицы скан-кода символа
org      0             ;начало основной программы
mov      a, #COLUMN
mov      b, #ROW
mov      dptr,
#TABL
mov      r0, #0        ;установка счетчика столбцов
m1: call  led          ; вызов подпрограммы определения скан-кода
      mov  p2, a       ;подключения одного столбца
      mov  p1, b       ;и всех рядов матрицы
      call loop        ;задержка на 2,5 мс (частота миганий 50
ГЦ) r1 a              ;следующий столбец
      inc  r0          ;икримент счетчика
столбцов orl p1, #0ffh ;гашение текущего столбца
      cjne r0, #8, m1  ;продолжать до конца столбцов матрицы
      mov  r0, #0      ;начинается
      mov  r1, #0      ; следующий
      jmp  m1          ; цикл
      led: push acc
      mov  a, r1
      movc a, @A+DPTR mov
           b, a
      inc  r1
      pop
           ac
      c ret
loop: MOV R2, #01h
      MOV R3, #01h
      DJNZ R3, $
      DJNZ R2, $-5
      ret
org      50h
db      01111110b
db      10111101b
db      11011011b
db      11100111b
db      11100111b
db      11011011b
db      10111101b
db      01111110b

```

end

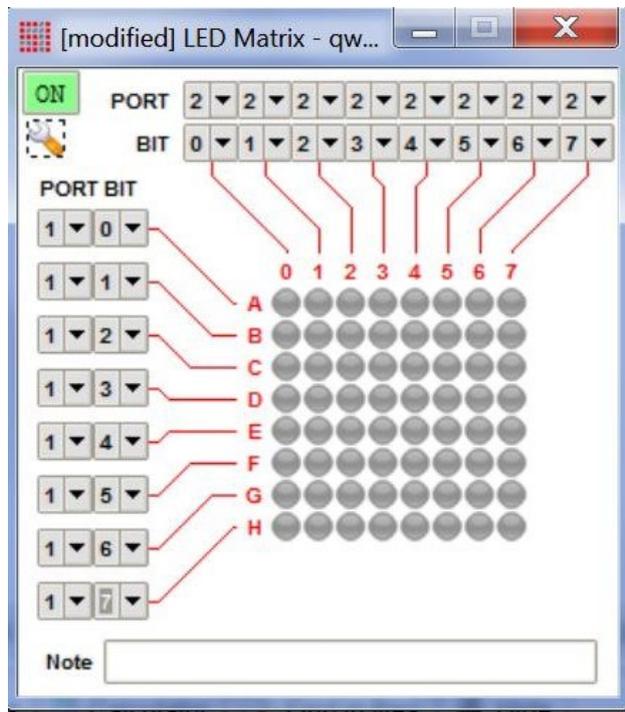


Рис.5 Подключение матричного индикатора

Подключить матричный индикатор к портам микроконтроллера (см. рис.5). Открыть в нем вкладку инструменты/послесвечение (Fade out interval) и установить не менее 500. Во вкладке Light up when установить Row 0 & Column 1.

Написать комментарий к подпрограммам led, loop. Рассчитать время задержки подпрограммы loop.

3. Разработать и отладить на эмуляторе индивидуальное задание в соответствии со своим вариантом (N):

При нажатии кнопки N на матричной клавиатуре 4*4 символ отображается на матричном индикаторе 8*8, в противном случае матричный индикатор потушен.

4 Контрольные вопросы

1. Чему равно содержимое портов МК51 после системного сброса?
2. Чему равен выходной ток логического нуля и единицы портов микроконтроллера (мА)?

3. Какими командами можно установить бит P1.1?
4. Приведите схему подключения светодиода к линии P2.0

5 Содержание отчета

Отчет по лабораторной работе должен содержать: пункты Программы работы, исходные тексты программ, ответы на вопросы.

6. Лабораторная работа №6

Управление жидкокристаллическим индикатором

1 Цель работы

Целью лабораторной работы является программирование жидкокристаллического индикатора (ЖКИ) на основе контроллера HD44780.

2 Контроллер HD44780

Контроллер HD44780 фирмы Hitachi фактически является промышленным стандартом и широко применяется при производстве алфавитно-цифровых ЖКИ-модулей. Аналоги этого контроллера или совместимые с ним по интерфейсу и командному языку микросхемы, выпускают множество фирм, среди которых: Epson, Toshiba, Sanyo, Samsung, Philips. Еще большее число фирм производят ЖКИ-модули на базе данных контроллеров. Эти модули можно встретить в самых разнообразных устройствах: измерительных приборах, медицинском оборудовании, промышленном и технологическом оборудовании, офисной технике - принтерах, телефонах, факсимильных и копировальных аппаратах.

Контроллер HD44780 потенциально может управлять 2-мя строками по 40 символов в каждой (для модулей с 4-мя строками по 40 символов используются два однотипных контроллера), при матрице символа 5 x 7 точек. Контроллера также поддерживает символы с матрицей 5 x 10 точек, но в последние годы ЖКИ-модули с такой матрицей практически не встречаются, поэтому можно считать, что фактически бывают только символы 5 x 7 точек.

Существует несколько различных стандартных форматов ЖКИ-модулей (символов x строк): 8 x 2, 16 x 1, 16 x 2, 16 x 4, 20 x 1, 20 x 2, 20 x 4, 24 x 2, 40 x 2, 40 x 4. Встречаются и менее распространенные форматы: 8 x 1, 12 x 2, 32 x 2 и др., - принципиальных ограничений на комбинации и количество отображаемых символов контроллер не накладывает - модуль может иметь любое количество символов от 1 до 80.

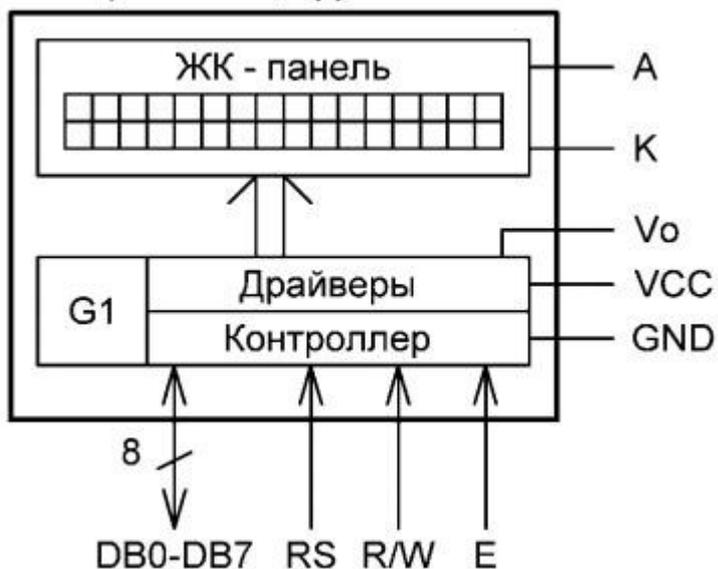


Рис.1 Упрощенная структура ЖКИ-модуля

Для соединения ЖКИ-модуля с управляющей системой (например микроконтроллером МК51) используется параллельная синхронная шина, насчитывающая 8 или 4 (выбирается программно) линий данных DB0...DB7, линию выбора операции R/W, линию выбора регистра RS и линию стробирования/синхронизации E. Кроме линий управляющей шины имеются две линии для подачи напряжения питания 5 В - GND и V_{CC}, и линия для подачи напряжения питания драйвера ЖКИ - V₀.

Обмен с ЖКИ-модулем выполняется чисто программными средствами, через порты ввода-вывода управляющего микроконтроллера. Следует помнить, что быстродействие микроконтроллера выше чем ЖКИ, поэтому между командами управления необходимы временные задержки. Длительность их указывается в техническом описании ЖКИ. Или другой вариант – перед посылкой команды проводить опрос специального бита занятости контроллера ЖКИ.

Управляющие выходы ЖКИ

RS – Register Select – выбор регистра. При помощи этого вывода сообщается ЖКИ какой тип данных будет записан/прочитан. **RS = 0**, работаем с регистром команд (Instruction Register) - используется для настройки ЖКИ.

RS = 1, работаем с регистром данных (Data Register) – используется для индикации символов на ЖКИ.

R/W – Read/Write – запись/чтение. При помощи этого вывода переключается дисплей в режим записи или чтения. **RW = 0**, запись данных/команд в ЖКИ. **RW = 1** чтение данных/переменных из ЖКИ.

E – Enable – разрешение. При помощи этого вывода, активизируется исполнение операции записи/чтения команд/данных. Или другими словами, на этот вывод подается “стробирующий сигнал”, без которого не может быть выполнена ни одна операция. Исполнение операций ЖКИ, начинаются при

ИЛИ команду, а младшие содержат собственно флаги.

Таблица 1. Флаги, управляющие работой контроллера HD44780

I/D:	режим смещения счетчика адреса АС, 0 - уменьшение, 1 - увеличение.
S:	флаг режима сдвига содержимого экрана. 0 - сдвиг экрана не производится, 1 - после записи в DDRAM очередного кода экран сдвигается в направлении, определяемым флагом I/D: 0 - вправо, 1 - влево. При сдвиге не производится изменение содержимого DDRAM. изменяются только внутренние указатели расположения видимого начала строки в DDRAM.
S/C:	флаг-команда, производящая вместе с флагом R/L операцию сдвига содержимого экрана (так же, как и в предыдущем случае, без изменений в DDRAM) или курсора. Определяет объект смещения: 0 - сдвигается курсор, 1 - сдвигается экран.
R/L:	флаг-команда, производящая вместе с флагом S/C операцию сдвига экрана или курсора. Уточняет направление сдвига: 0 - влево, 1 - вправо.
D/L:	флаг, определяющий ширину шины данных: 0 - 4 разряда, 1 - 8 разрядов.
N:	режим развертки изображения на ЖКИ: 0 - одна строка, 1 - две строки
F:	размер матрицы символов: 0 - 5 x 8 точек, 1 - 5 x 10 точек.
D:	наличие изображения: 0 - выключено, 1 - включено
C:	курсор в виде подчеркика: 0 - выключен, 1 - включен
B:	курсор в виде мерцающего знакоместа: 0 - выключен, 1 - включен.

Таблица 2. Значения управляющих флагов после подачи питания на контроллер

I/D = 1:	режим увеличения счетчика на 1
S = 0:	без сдвига изображения
D/L = 1:	8-ми разрядная шина данных
N = 0:	режим развертки одной строки
F = 0:	символы с матрицей 5 x 8 точек
D = 0:	отображение выключено
C = 0:	курсор в виде подчеркика выключен
B = 0:	курсор в виде мерцающего знакоместа выключен

Список управляющих комбинаций битов регистра IR и выполняемые ими команды приведены в таблице 3. Так как на момент включения ЖКИ ничего не отображает (флаг D = 0), то для того, чтобы вывести какой-либо текст необходимо, как минимум, включить отображение, установив флаг D = 1.

Перед работой с ЖКИ необходимо провести его инициализацию (предварительную настройку). Пример широко распространенной последовательности команд для инициализации ЖКИ: 38h, 0Ch, 06h

38h устанавливает режим отображения 2-х строк с матрицей 5 x 8 точек и работу с 8-ми разрядной шиной данных;

0Ch включает отображение на экране ЖКИ, без отображения курсоров;

06h устанавливает режим автоматического перемещения курсора слева-направо после вывода каждого символа.

Таблица 3. Управляющие комбинации битов регистра IR

D7	D6	D5	D4	D3	D2	D1	D0	Назначение
0	0	0	0	0	0	0	1	Очистка экрана, счетчик адреса AC = 0, адресация AC на DDRAM
0	0	0	0	0	0	1	-	AC = 0, адресация на DDRAM, сброшены сдвиги, начало строки адресуется в начале DDRAM
0	0	0	0	0	1	I/D	S	Выбирается направление сдвига курсора или экрана
0	0	0	0	1	D	C	B	Выбирается режим отображения
0	0	0	1	S/C	R/L	-	-	Команда сдвига курсора/экрана
0	0	1	DL	N	F	-	-	Определение параметров развертки и ширины шины данных
0	1	AG	AG	AG	AG	AG	AG	Присвоение счетчику AC адреса в области CGRAM
1	AD	AD	AD	AD	AD	AD	AD	Присвоение счетчику AC адреса в области DDRAM

Контроллер HD44780 поддерживает как операции записи так и операции чтения. Чтение регистра данных DR приводит к загрузке содержимого DDRAM или CGRAM, в зависимости от текущего режима, при этом курсор смещается на одну позицию, как и при записи. Чтение регистра IR возвращает 8 значащих разрядов, причем в 7-ми младших содержится текущее значение счетчика AC (7 разрядов, если адресуется DDRAM, и 6 - если CGRAM), а в старшем - флаг занятости BF. Этот флаг имеет значение 1 когда контроллер занят и 0 - когда

свободен. Необходимо учитывать, что большинство операций, выполняемых контроллером, занимают значительное время, около 40 мкс, а время выполнения некоторых доходит до единиц миллисекунд, поэтому цикл ожидания снятия флага BF должен обязательно присутствовать в программах драйвера ЖКИ и предшествовать совершению любой операции (естественно, кроме операции проверки флага BF).

Один важный момент! После совершения операции записи или чтения DDRAM и появления после нее признака готовности ($BF = 0$), прочитанное в этом же цикле (вместе с флагом BF) значение AC скорее всего не будет достоверным. Дело в том, что между появлением признака готовности и вычислением контроллером нового значения AC существует некоторый временной интервал, составляющий около 4 мкс при тактовой частоте контроллера 270 кГц. Поэтому, если необходимо получить истинное значение AC, нужно совершить повторную операцию прочтения IR спустя не менее чем 4 мкс (если контроллер работает на частоте 270 кГц время ожидания необходимо пропорционально увеличить).

Вывод на экран символа производится записью его кода в регистр DR. При этом символ размещается в DDRAM по текущему адресу, указываемому AC, а значение AC увеличивается или уменьшается на 1. Чтобы произвести переустановку курсора на нужную позицию, необходимо присвоить AC соответствующее значение (см. таблицу 3). Здесь есть одна тонкость. Когда производится последовательная запись символов и в результате заполняется вся строка, курсор автоматически переходит на вторую строку, но если необходимо принудительно установить курсор, скажем, на начало второй строки, то будет неверным присвоить AC казалось бы логичное значение 28h (40), правильным является значение 40h (64). Значения адресов DDRAM в диапазоне 28h...3fh (а равно и 68h...7Fh) являются неопределенными и результаты работы с ними могут быть непредсказуемыми.

На рис.3 показана последовательность сигналов управления и данных при чтении и записи ЖКИ.

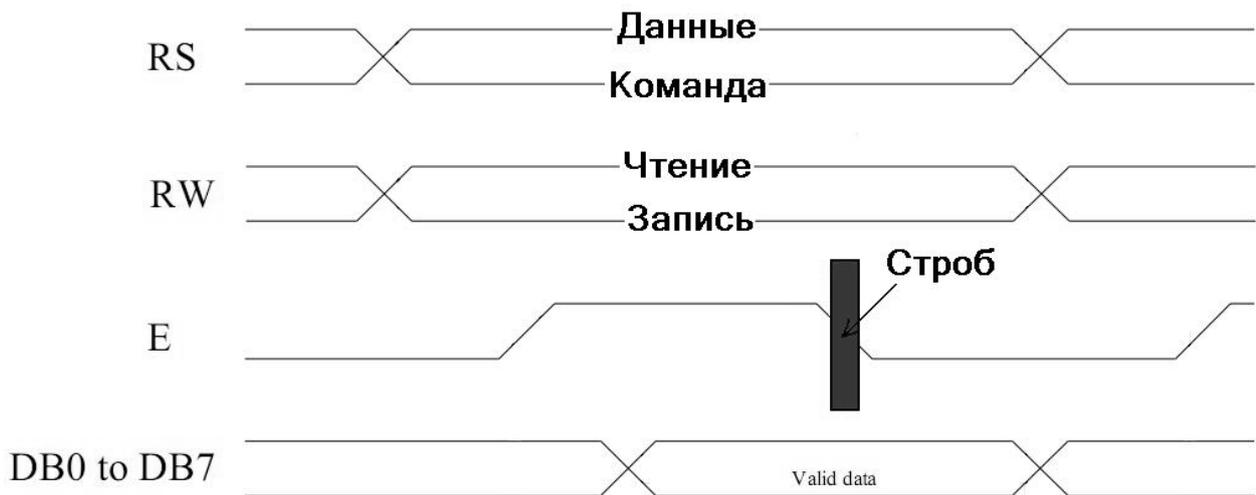


Рис.3 Временные диаграммы сигналов управления

Из приведенных диаграмм видна последовательность действий в управляющей программе для работы с ЖКИ. Например, для записи в ЖКИ данных (символа) необходимо:

1. Установить значение линии RS=1, R/W=0
2. Вывести значение байта на шину DB7...DB0
3. Установить значение линии E=1
4. Установить значение линии E=0

При выводе на ЖКИ следующего символа эту последовательность необходимо повторить. Для нормальной работы ЖКИ задержка после каждого шага должна быть не менее 250 нс.

3 Программа работы

3.1 Проверить работоспособность программы HD.asm. Для проверки программы подключить ЖКИ как на рис.4 в режиме 2x16. Для этого в главном меню выбрать Virtual HW/Led display (HD44780)/2x16 и установить значения портов.

```
org 0
    jmp start

RS bit P2.2    ;бит команда данные
RW bit P2.1    ;бит чтения/записи
E bit P2.0     ;бит строба
D equ P1       ;порт данные
;Макрос записи данных/команд в HD44780
hd macro hd_code
    setb E
    mov D, hd_code
    clr E
```

```

endm                                ;конец макроса

string:    db    'Welcome to TUSUR '

start: mov  D, #0
        clr  RW
        mov  DPTR, #string
        ;Инициализация дисплея
main: clr   RS
        hd   #00000001b    ; Очистить дисплей
        hd   #00000010b    ; Установить курсор в начало
        hd   #00000110b    ; Выбор направления курсора
        hd   #00001111b    ; Включение дисплея и мерцающего
                                ; курсора
        hd   #00011110b    ; Сдиг курсора вправо
        hd   #00111100b    ; Определение параметров развертки
        hd   #10000001b    ; Определение начального адреса в
                                ; DDRAM
        ;Конец инициализации

        ;Печать символов в цикле
        setb RS
        mov  R0, #0
print: mov  A, R0
        inc  R0
        movc A, @A+DPTR
        hd   A
        cjne A, #0, print
        sjmp main
        end

```

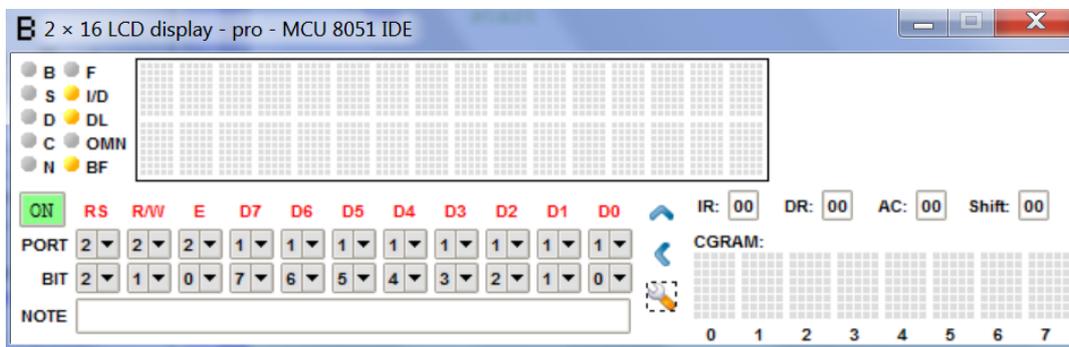


Рис. 4 Подключение ЖКИ к микроконтроллеру
Зафиксировать состояние DDRAM (иконка Инструмент ЖКИ/DDRAM) до

работы программы и во время работы программы. Что записано в ячейки DDRAM до выполнения программы?

В программе применяются элементы Макроассемблера. Макрос начинается с имени `hd` и заканчивается директивой `endm`. Что выполняет Макрос?

3. Изменить программу `HD.asm` и отладить на эмуляторе индивидуальное задание в соответствии со своим вариантом (N):

При работе программы на верхней строке отображается Ваше имя, на нижней - фамилия, курсор – знак подчеркивания (для четного N)

При работе программы на верхней строке отображается Ваша фамилия, на нижней - имя, режим без курсора (для нечетного N)

4 Контрольные вопросы

1. Запишите макрос для чтения бита занятости HD44780.
2. Какой объем памяти DDRAM HD44780?
3. Сколько символов пользователя может воспроизвести HD44780?
4. Где хранятся символы пользователя HD44780?

5 Содержание отчета

Отчет по лабораторной работе должен содержать: пункты Программы работы, исходные тексты программ, ответы на вопросы.

7. Список рекомендуемой литературы

1. Шарапов А.В. Микроэлектроника. Цифровая схемотехника: Учебное пособие / А.В. Шарапов. – Томск: Томский государственный университет систем управления и радиоэлектроники, 2007. – 162 с.: ил.,табл. – (Приоритетные национальные проекты. Образование). – ISBN 978-5-86889-400-8 (наличие в библиотеке ТУСУР - 90 экз.)

2. Основы микропроцессорной техники: Учебное пособие / Шарапов А. В. - 2008. 240 с. [Электронный ресурс] - Режим доступа: <https://edu.tusur.ru/publications/834>, дата обращения: 30.05.2017.

3. Сайт Цифровая и микропроцессорная техника-1 [Электронный ресурс]. - <https://sdo.tusur.ru/course/view.php?id=88> (доступ авторизованный).

4. Сайт Цифровая и микропроцессорная техника-2 [Электронный ресурс]. - <https://sdo.tusur.ru/course/view.php?id=427> (доступ авторизованный).