

## 2.2. Лабораторная работа № 2.

### Подключение внешней памяти и ее тестирование

**Цель работы:** разработать схему подключения микроконтроллера с внешней памятью и протестировать память.

#### 2.2.1. Особенности подключения к МК внешней памяти и периферийных устройств

Микроконтроллер 8051 может работать с внешней памятью данных емкостью до 64 Кбайт, построенной на одной или нескольких микросхемах статической памяти.

В БД Multisim имеются микросхемы RAM с байтовой организацией объемом 2Kx8 и 8Kx8 бит (рис. 2.17).

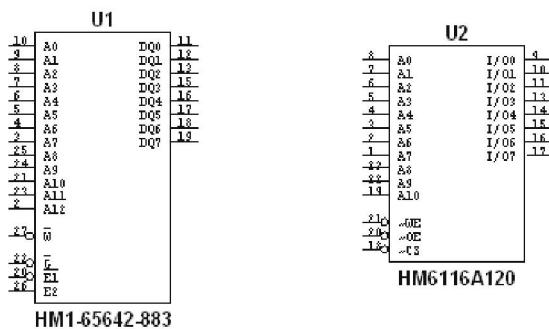


Рис. 2.17. Статическая RAM 8Kx8 и 2Kx8

Такие микросхемы имеют 8 двунаправленных выводов данных (D0–D7), 11 или 13 адресных входов (A0–A10 или A0–A12). Вход WE (W) определяет характер обращения: если на нем установлена 1, то осуществляется чтение из выбранной ячейки, при WE = 0 в ячейку будет записана информация. Вход CS (E1,E2) активизирует микросхему памяти: когда на входе CS установлено 1, она выключена, при CS = 0 допускается любое обращение к памяти. Нулевой сигнал на входе OE (G) разрешает работу выходной шины данных микросхемы.

В БД Multisim также представлены микросхемы ПЗУ (рис. 2.18): ROM (32Kx8), EPROM (8Kx8, 16Kx8). Такие микросхемы в рабочем режиме допускают только считывание информации. Выводы микросхем аналогичны микросхемам RAM, кроме вывода PGM, отвечаю-

щего за программирование. Подробно работа микросхем ОЗУ и ПЗУ рассматривается, например, в [8].

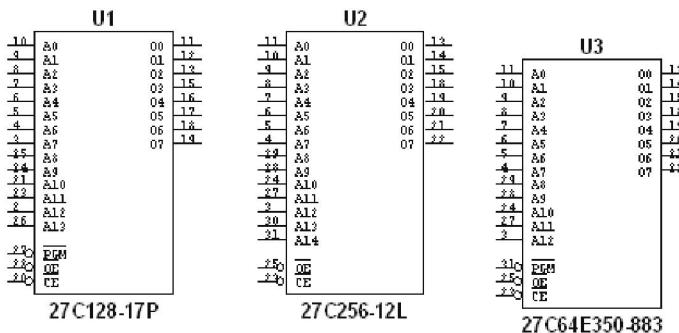


Рис. 2.18. ПЗУ EPROM 16Kx8, ROM 32Kx8 и EPROM 8Kx8

В микроконтроллерах MK51 существует 4 многофункциональных 8-битовых порта ввода/вывода P0, P1, P2, P3, предназначенных для обмена информацией с различными внешними устройствами и для выполнения специализированных функций, таких как подключение внешней памяти программ, данных, программирование внутреннего ПЗУ и др.

Каждый порт может адресоваться как побайтно, так и побитно, по конкретным физическим адресам. При подключении к МК внешней памяти через порт P0 выводится младший байт адреса, а также передается и принимается в микроконтроллер байт данных (в мультиплексированном режиме). В 1 и 2 тактах машинного цикла при обращении к внешней памяти на линиях P0 активизируется адресная информация A0–A7 при высоком уровне сигнала ALE, а затем на этих же линиях появляется сигнал D0–D7 (при низком уровне сигнала ALE). Для фиксации байта адреса в течение всего машинного цикла используются регистры-защелки, например, 74LS373N, информация в которых фиксируется по спаду сигнала на его входе ENG (рис. 2.25).

Через порт P2 выводится старший байт адреса (разряды A8–A15) внешней памяти программ и данных. Для каждого из битов порта P3 имеется ряд альтернативных функций. Сигналы стробов записи (WR#) и чтения (RD#) внешней памяти формируются на линиях P3.6 и P3.7 соответственно. Альтернативные функции всех портов реализуются только в том случае, если в соответствующий разряд фиксатора-защелки порта записана логическая «1», иначе на соответствующем выводе будет присутствовать «0».

Каждый вывод портов P0–P3 может использоваться как вход или выход независимо от других. Для настройки линии порта на ввод информации необходимо в соответствующий разряд порта записать «1», а для использования в качестве выхода – «0». При системном сбросе в регистрах защелках всех портов устанавливается значение FFh.

## 2.2.2. Порядок выполнения лабораторной работы

### 2.2.2.1. Создание схемного проекта

Открываем и сохраняем новый схемный проект Circuit2.

Размещаем на рабочем поле МК-51 (см. лабораторную работу № 1), микросхему памяти емкостью 2 Кбайта (Place – Component – MCU – RAM), регистр-защелку, например 4037ВР, которую можно найти в группе CMOS в семействе CMOS\_5V, землю и питание (Place – Component – Sources – POWER\_SOURCES). Собранная электронная схема представлена на рис. 2.25.

После выбора компонентов из БД и размещения их на схеме, необходимо соединить компоненты между собой. В программе Multisim действие мышью на схеме зависит от положения курсора. Внешний вид курсора меняется в зависимости от того, на какой объект он наведен (рис. 2.19.) Для того чтобы провести соединяющий провод между элементами, необходимо кликнуть ЛКМ по выводу одного элемента, затем, не отпуская кнопку, провести соединение до вывода другого элемента. После появления соединяющего проводника между элементами Multisim автоматически присвоит ему номер. Нумерация линий увеличивается последовательно, начиная с 1.

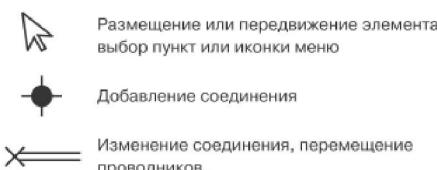


Рис. 2.19. Виды курсора

Заземляющие провода всегда имеют номер 0, это требование связано с работой скрытого эмулятора SPICE. Чтобы изменить нумерацию соединения или присвоить ему логическое имя, необходимо дважды кликнуть по соединительному проводнику (рис. 2.20).

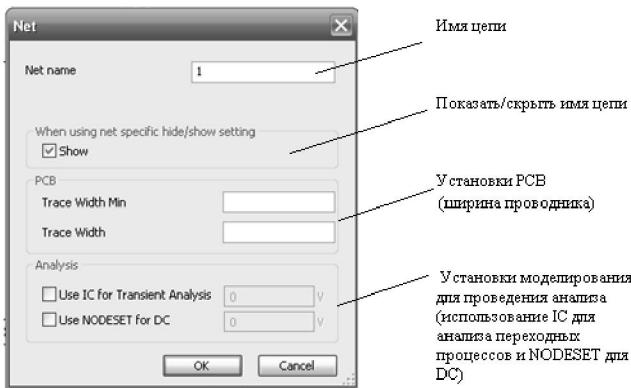


Рис. 2.20. Установки цепи

В Multisim имеется функция автоматического соединения выводов между собой и с проводниками. При добавлении компонента к существующей сети соединений необходимо, чтобы выводы компонента касались существующей сети, либо поместить компонент параллельно соединению.

Для размещения соединяющей шины в схемном проекте необходимо выбрать пункт меню Place – Bus или значок на панели компонентов или горячую клавишу Ctrl-U.

Рассмотрим подключение требуемых компонентов к шине на примере компонента U3 (см. рис. 2.25). При помощи ЛКМ размещаем шину напротив выводов, необходимых для подключения компонентов, например, напротив выводов D0–D7 элемента U3, затем фиксируем положение шины ПКМ. Далее в контекстном меню компонента U3, соединяемого с шиной, нажимаем Bus Vector Connect (рис. 2.21).

Слева окна соединений представлены необходимые данные о компоненте (обозначение, положение выводов, выводы), справа – ошине (обозначение, шинные линии). Выбираем в подменю Pins выводы компонента 1D–8D для подключения к шине, удерживая клавишу Shift и при помощи стрелок отправляем их в нижнее окошко. Справа окна соединений выбираем в поле Name шину, в нашем случае Bus1. Далее необходимо сформировать вектора (линии шины), Multisim предлагает для этого два способа: ручной и автоматический.

**Формирование векторов вручную.** Нажимаем кнопку Add buslines (см. рис. 2.21) для задания векторов, появляется соответствующее окно (рис. 2.22).

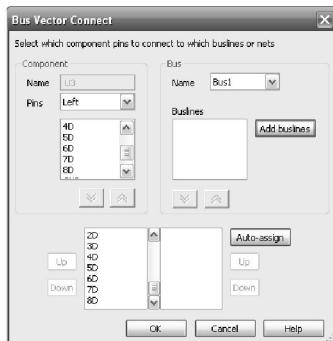


Рис. 2.21. Окно соединения компонента с шиной

Здесь указываются префикс (метка), например, *In*, начальное значение вектора, инкремент и количество векторов (в нашем случае их 8). После нажатия на кнопку OK (рис. 2.22) полученные линии шины (векторы) отразятся в предыдущем окне. Количество линий шины должно быть эквивалентно выбранным для подключения кшине выводам. Далее выделяем полученные линии шины и нажимаем кнопку OK (см. рис. 2.21).

**Автоматическое формирование векторов.** Вектора вводятся автоматически при нажатии кнопки Auto-assign (см. рис. 2.21).

В завершении нажимаем кнопку OK. В таком же порядке подключаем к шине выводы MK51.

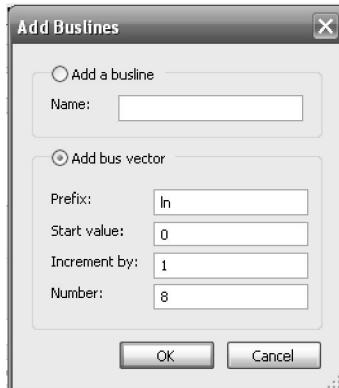


Рис. 2.22. Окно задания линий шин вручную

При подключении внешней памяти аналогичным образом рисуем шину напротив необходимых выводов элемента U2 и фиксируем ее. Эта шина автоматически будет названа Bus2. Для объединения шины Bus2 к шине Bus1 необходимо выбрать в контекстном меню Bus2 пункт Properties (рис. 2.23).



Рис. 2.23. Окно свойств шины

В открывшемся окне нажимаем кнопку Merge. В следующем открывшемся окне (рис. 2.24) слева указана первая шина, предназначенная для объединения Bus1, а справа вторая – Bus2. В окне объединения шин в поле Name второй шины выбираем Bus1, после чего в окошке Buslines появятся заданные ранее вектора Bus1. Ниже указана объединенная шина с её векторами.

Далее в окне объединения шин нажимаем кнопку Merge и в окне свойств шины кнопку OK. Как и ранее, выбираются выводы компонента и вектора шины, только в этом случае при нажатии кнопки OK Multisim предупреждает, что фрагмент шины, к которой необходимо произвести подключение, выбирается двойным кликом ЛКМ. Готовая схема представлена на рис. 2.25.

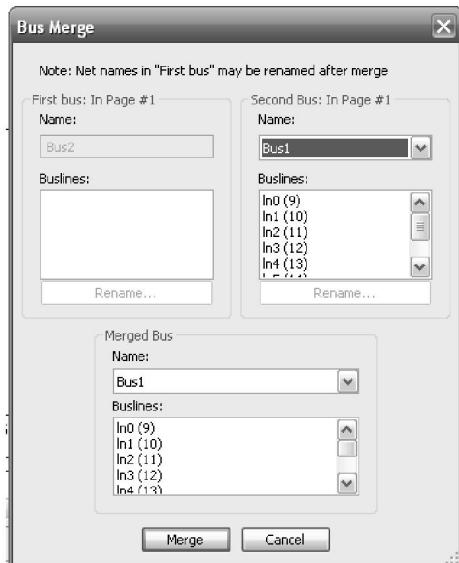


Рис. 2.24. Окно объединения шин

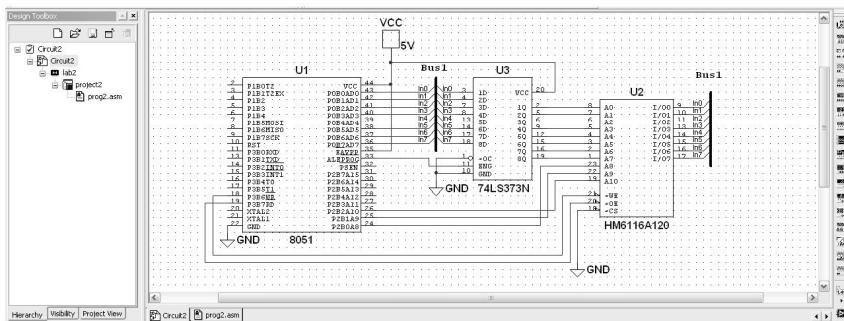


Рис. 2.25. Схема подключения внешней памяти к МК-51

В окне разработки (см. рис. 2.25) указана вся структура созданного проекта.

В заключении в свойствах МК на вкладке Value в поле Built-in External RAM необходимо указать объем подключенной внешней памяти (рис. 2.26).

## *Разработка программного файла*

Активируем закладку программного файла prog2.asm (prog2.c), щелкнув по ней ЛК мыши либо выбрав программный файл в окне разработки.

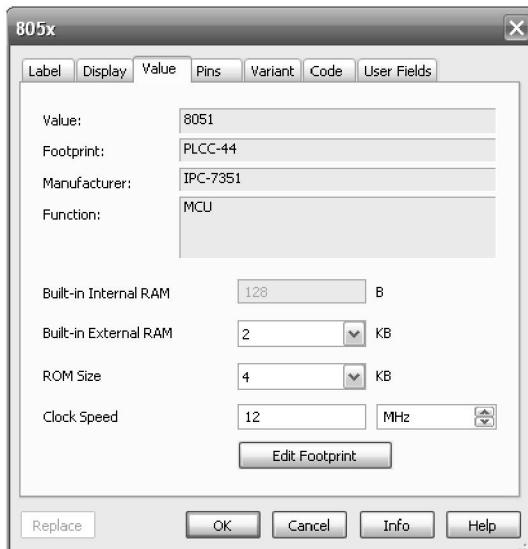


Рис. 2.26. Окно свойств МК, вкладка Value

Если имеется готовая программа, разработка которой выполнена, например, в среде PRO View [1], то вставляем полученный ассемблерный (или C) программный код и сохраняем его.

Покажем программу **тестирования внешней памяти** на следующем примере: требуется проверить 255 байт внешней памяти, начиная с ячейки 00h, используя тестовый набор 055h.

Ассемблерный файл программы:

```
$MOD51      ;подключение МК-51
org 00h      ;начинаем программу с адреса 00h
mov dptr,#00h ;загрузить в регистр-указатель число 00h
mov r2,#0ffh ;загрузить в регистр R2 число 0ffh (счетчик цикла)
mov r1,#055h ;загрузить в регистр R1 число 55h
test:
mov a,r1    ;загрузить аккумулятор ACC операндом из регистра R1
```

```

movx @dptr,a ;переслать в ячейку внешней памяти XRAM содержи-
               мое аккумулятора ACC
movx a,@dptr ;считать в аккумулятор содержимое текущей ячейки
               внешней памяти
xrl a,#055h ;операция XOR считанного и изначального операнда,
               если 0 в Акк, то ячейка работает нормально
jnz error    ;ошибка – выход из программы
inc dptr    ;инкремент содержимого регистра DPTR – переход
               к следующему адресу
djnz r2,test ;вычесть 1 из содержимого регистра R2 и перейти по
               метке, если в ячейке не 0
error:
END

```

Выбираем в меню MCU – MCU 8051 U1, пункт Build, который позволяет откомпилировать и отредактировать программный файл. В окне сообщений на экране отражается информация об ошибках и предупреждениях. Если имеются ошибки, в окне сообщений указаны номера строк, в которых они находятся. Для отображения нумерации строк необходимо выбрать меню MCU – Show lines numbers.

Моделирование работы схемы производится через меню Simulate – Run, либо горячую клавишу F5 или иконку на панели Моделирование – . При этом открывается окно отладчика. Для пошагового моделирования используется пункт Step into в меню MCU, горячая клавиша F11 или иконка на панели Моделирование .

Кнопка Pause или горячая клавиша F6 инициируют паузу моделирования, при нажатии которой возможно посмотреть, на каком этапе находится симуляция.

Кнопка Stop останавливает выполнение моделирования.

Просмотр памяти микроконтроллера активируется через меню MCU – MCU 8051 U1 – Memory view, также можно использовать пункт MCU Windows, где во всплывающем окне можно отметить галочкой необходимые для работы окна атрибуты. При необходимости выполнения программы до определенной инструкции в Multisim предусмотрены точки останова, установить которые возможно через меню MCU – Toggle breakpoint или через панель Моделирование – .

Кнопка Remove all breakpoints – удаляет все точки останова.

После выполнения программой инструкции djnz r2, test посмотрим результаты записи во внешнюю память (рис. 2.27). В ACC записано число 00h, в R1 – 55h, в R2 – FEh, в первой ячейке внешней памяти XRAM – 55h.

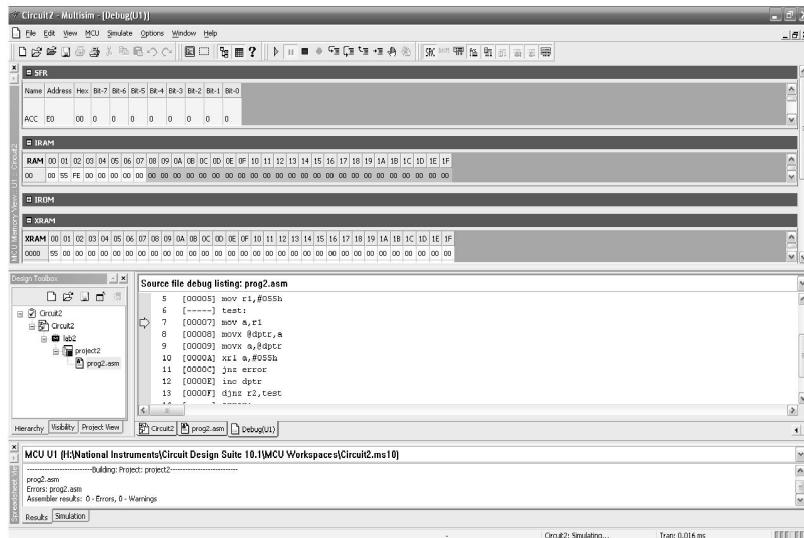


Рис. 2.27. Результаты моделирования после инструкции djnz r2, test

После завершения моделирования в каждой ячейке внешней памяти будет записано число 55h, регистр R2 будет иметь нулевое значение.

Работа с С-файлом аналогична. Ниже рассматривается вариант тестирования внешней памяти с адресами от 400h до 7FFh.

```
#include <8051.h>           //подключаем заголовочные файлы
void main() {                //главная функция, точка входа в программу
{
    int i;                  //объявляем переменные
    //переменная ptr – указатель на адрес ячейки памяти внешнего ОЗУ
    char xdata *ptr;
    //символьные переменные test, nabor (байтовые переменные)
    char test, nabor;
    nabor=0x55;              //тестовый набор
}
```

```

ptr= (char xdata *) 0x400; //начальный адрес внешней памяти
for(i=0; i<1024;i++)      //будут проверены 1024 ячейки памяти
{
//в каждую ячейку памяти посыпается значение тестового набора
*ptr=nabor;
//переменная test принимает значение содержимого ячейки памяти
test=*ptr;
//если test ≠ nabor, то конец цикла (ошибка), иначе проверяется следующая ячейка памяти;
if(test!=nabor)
{
break;
}
ptr++;
}
}

```

Из сравнения программных кодов видно, что тестирование больших массивов данных ( $> 255$  байт) легче организовать на С, чем на ассемблере, так как в ассемблерной программе потребуется организация двойных циклов.

В заключение лабораторной работы научимся пользоваться виртуальным осциллографом. Произведем подключение осциллографа к данной схеме. Выбираем на панели инструментов иконку Oscilloscope и устанавливаем на рабочей области. Символ осциллографа имеет два канала: Channel A и Channel B. Подсоединяем «+» канала A к выводу ALE/PROG микроконтроллера, «-» – к земле. На панели прибора необходимо указать деление временной шкалы, равное одному машинному циклу, то есть 1 мкс. Выбираем по оси у цену деления для канала A, примем ее равной 5 В. После запуска моделирования виден процесс генерации сигнала ALE, дважды за машинный цикл (рис. 2.28).

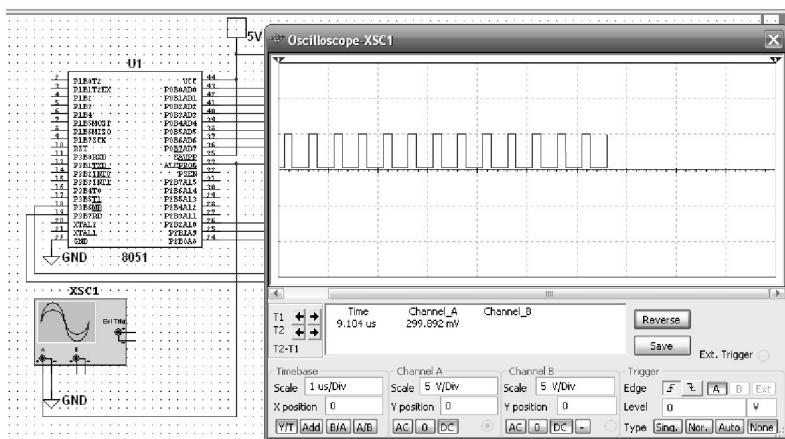


Рис. 2.28. Подключение осциллографа и его инструментальная панель во время моделирования

### 2.2.3. Задания для лабораторной работы

1. Подключить к МК внешнее ОЗУ емкостью N кбайт (табл. 2.9) и используя тестовый набор XX, произвести тестирование области памяти 1 кбайт, начиная с адреса ZZZ. Включить светоиндикатор, если записанный и считанный из ячейки памяти наборы не совпадают. Подключить осциллограф, как показано на рис. 2.28, и снять осциллограмму с вывода 0 ALE МК.

Таблица 2.9

Задание 1

Параметры	1	2	3	4	5	6
N, кбайт	8	2	8	8	8	2
XX	0AAh	55h	00h	0FFh	55h	0AAh
ZZZ	800h	100h	0C00h	1000h	1400h	400h

2. Подключить к МК внешнее ПЗУ емкостью N Кбайт (табл. 2.10). Подсчитать контрольную сумму области памяти емкостью Кбайт, начиная с адреса ZZZ и сравнить ее с константой, находящейся в резидентном ПЗУ. Если сравниваемые величины совпадают, тестирование ПЗУ прошло успешно, иначе включить светоиндикатор. Подключить осциллограф, как показано на рис. 2.28, и снять осциллограмму с вывода ALE МК.

Таблица 2.10

## Задание 2

Параметры	7	8	9	10	11	12
$N$ , Кбайт	8	16	32	8	16	32
$K$ , байт	100	300	500	150	350	650
ZZZ	800h	1000h	2000	0C00h	1000h	400h

**2.2.4. Содержание отчета**

1. Наименование и цель работы.
2. Описание микросхемы памяти с назначением выводов и осциллографа, использованного в эксперименте, с его краткими характеристиками.
3. Копия схемного файла во время моделирования с указанием позиционных обозначений элементов и шин.
4. Копия программного файла (на ассемблере или на С) с подробными комментариями.
5. Полученные результаты и выводы по работе.

**2.2.5. Вопросы для самоконтроля**

1. Какие функции выполняет регистр-защелка при подключении к МК внешней памяти?
2. Назовите альтернативную функцию, выполняемую портом Р2.
3. Объясните назначение сигнала PSEN (37 вывод МК-51).
4. Поясните, в чем отличия в работе процессора при обращении к внешней памяти данных по сравнению с резидентной?
5. Каково назначение вывода EA/Vpp МК?
6. Каким образом можно использовать внешнее ОЗУ для обращения к данным и к программе?
7. Как подсчитывается контрольная сумма заданной области памяти?
8. Объясните назначение выводов CE (CS), OE микросхем памяти.
9. Каковы конструктивно-технологические отличия однократно-программируемых микросхем ПЗУ (PROM) по сравнению с репрограммируемыми ПЗУ (EPROM – с ультрафиолетовым стиранием)?
10. Поясните особенности тестирования микросхем ОЗУ по сравнению с ПЗУ.