**Министерство общего и профессионального образования**

**Российской Федерации**

**УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**КАФЕДРА РЭИС**

# модуль оперативного запоминающего устройства

## Пояснительная записка

**К курсовой работе по дисциплине**

***«Цифровые устройства и микропроцессоры»***

 Руководитель: Добряк В.А.

 Студент:

###  Группа:

Екатеринбург

2000

ЗАДАНИЕ НА ПРОЕКТИРОВАНИЕ

Модуль ОЗУ и программа его тестирования.

***Основные технические требования:***

**–** Ёмкость – 128 Кб;

– Раздельные шины адреса и данных;

– Статическое оперативное запоминающее устройство;

– Микропроцессор – Z80.

**СОДЕРЖАНИЕ**

##### Введение

1. Принцип работы устройства
2. Описание работы программы функционального контроля памяти
3. Алгоритм теста
4. Программа тестирования устройства на ассемблере

###### Список использованной литературы

Приложение 1. Схема электрическая принципиальная модуля ОЗУ

Приложение 2. Технические характеристики микросхемы ОЗУ UM621024A

ВВЕДЕНИЕ

Развитие микроэлектроники в начале семидесятых годов привело к появлению микропроцессоров – новой разновидности БИС, особенность которых заключается в реализации полного перечня функций, выполняемых процессором современных компьютеров. В связи с широким спектром возможных применений эволюция архитектуры микропроцессоров пошла по нескольким различным направлениям, в результате чего появились следующие их классы:

• простые однокристальные 4- и 8-разрядные контроллеры невысокой производительности для применения в бытовых приборах и небольших подсистемах;

• быстродействующие секционные комплекты микропроцессорных БИС для создания машин произвольной разрядности, адаптируемых к алгоритмам обработки данных на микропрограммном уровне;

• мощные однокристальные 16- и 32-разрядные микропроцессоры, сравнимые по быстродействию с мини-ЭВМ;

• процессоры цифровой обработки сигналов, подключаемые к стандартным интерфейсам микроЭВМ для обработки арифметических алгоритмов, таких, как быстрое преобразование Фурье;

• аналоговые микропроцессоры – устройства, включающие АЦП, ЦАП, устройства цифровой обработки и представляющиеся пользователю как системы с аналоговым входом и аналоговым выходом.

Существенное отличие архитектуры, ориентированной на различные классы задач, порождает взаимную незаменяемость разных классов микропроцессоров. Развитие всех направлений дополняет друг друга, в результате чего создаётся элементная база, ориентированная на эффективное использование в составе систем и устройств различной производительности и потребления энергии.

Использование микропроцессоров позволило создать новый класс малопотребляющих компактных машин – микроЭВМ. Данная курсовой работа, в частности, и рассматривает одну из наиболее важных частей такой микроЭВМ – работу одного из однокристальных микропроцессоров – Z80 с оперативным запоминающим устройством ёмкостью 128 Кб.

1. ПРИНЦИП РАБОТЫ УСТРОЙСТВА

Выбранный микропроцессор Z80 как нельзя лучше подходит для создания микроЭВМ. Хотя в данное время производительность данного класса машин стремительно растёт, остаются ниши для использования и данного микропроцессора. Например, такие машины вполне себя оправдывают, если их использовать в технологических линиях контроля и сборки. Причём самых различных изделий. На такие рабочие места новую высокопроизводительную технику ставить нецелесообразно, а собрать на базе данного микропроцессора устройство, измеряющее параметры изделия и производящее регулировку его, достаточно просто и не требует значительных затрат.

Наиболее важное ограничение в использовании микропроцессора Z80 накладывает, то, что данный микропроцессор напрямую может использовать не более 216 ячеек памяти. Т.е. микропроцессор адресует всего 64 Кбайт памяти. Зачастую в тех областях, где применяется данный процессор более и не требуется, но иногда существует острая потребность в расширении адресного пространства. Это довольно просто удовлетворяется путём некоторой модернизации схемы работы микропроцессора с памятью.

В данной курсовой работе рассматривается один из способов расширения адресного пространства микропроцессора Z80 – метод окна.

Метод окна позволяет очень просто разрешать указанные проблемы. Идея метода состоит в отображении на «большом» отрезке лишь части «малого» отрезка. Эта часть соответствует свободным адресам микроЭВМ и называется окном.

РЕГИСТР

ДЕШИФРАТОР

И

микроЭВМ

G

К главной памяти

d

Адрес

Окно

MSYN

Рис.1. Схемная реализация метода окна

Система работает следующим образом:

1. Процессор загружает программно-доступный регистр старших разрядов адреса с шины данных, задавая положение проекции окна в адресном пространстве главной памяти.
2. Процессор обращается к некоторому адресу, лежащему внутри окна.
3. Дешифратор опознаёт принадлежность текущего адреса фиксированному окну и разрешает передачу в главную память сигнала сопровождения адреса. В главную память выдаётся полный адрес, определяющий одну из её ячеек.
4. Выбранная ячейка главной памяти выдаёт или принимает информацию в зависимости от сигнала на линии управления режимом работы и посылает сигнал по линии передачи ответа.

Принципиальная схема устройства находится в Приложении 1.

В данном случае используется двухстраничное деление памяти. Всё ОЗУ находится на одной микросхеме, и переключение страниц производится путём подачи на регистр по шине данных сигнала высокого уровня, который переключает диапазон адресов ОЗУ. Используется только один разряд шины данных. Использование всех разрядов позволяет получить 40322 страницы памяти по 64 Кбайт. Т.е. применение данной схемы позволяет адресовать около 2,64 Гбайт памяти.

Для обеспечения режима ожидания предусмотрена схема перевода микропроцессора в режим ожидания. Ожидание обозначает некоторый период времени, необходимый для выполнения операций чтения или записи в память. Когда время ответа (реакции) памяти больше, чем время цикла команды, цикл команды должен быть увеличен. Период ожидания должен быть достаточным для срабатывания устройства памяти. Можно отметить, что в данной схеме выход второго триггера соединяется с выводом 4 вентиля НЕ И. Т.к. время доступа памяти выбранного типа не превышает 100 нс., а период последовательности тактовых импульсов микропроцессора Z80 не менее 1 мкс., то задержка на один период последовательности тактовых импульсов будет достаточной. Сигнал запроса ожидания на вход триггера типа D поступает с выхода микропроцессора MREQ. Для подачи синхроимпульсов используется линия, по которой поступают синхроимпульсы на вывод 6 микропроцессора Z80.

**2. ОПИСАНИЕ РАБОТЫ ПРОГРАММЫ ФУНКЦИОНАЛЬНОГО КОНТРОЛЯ ПАМЯТИ**

Под функционированием какого-либо объекта понимается выполнение предписанного ему алгоритма функционирования при применении объекта по назначению. Методы функционального контроля основаны на сравнении с эталонными сигналами выходных реакций (сигналов) тестируемой схемы на заданные входные воздействия.

Одним из основных узлов системы функционального контроля является генератор тестов, предназначенный для формирования последовательности тестирующих и эталонных сигналов по заданному закону. Наборы входных сигналов, задаваемые в виде машинных слов (кодов), определяют порядок обращения к элементам памяти и последовательность выполняемых операций. Математические адреса элементов памяти могут не совпадать с их физическими координатами на кристалле.

Для тестирования памяти в нашем случае, применим линейный алгоритм “Марш”, с некоторой модернизацией, позволяющей более качественно провести функциональный контроль. Данный алгоритм отличается высоким быстродействием, позволяющим применять его для технологического контроля микросхем памяти. Модернизация алгоритма связана с тем, что сначала память тестируется несколькими разными тестовыми значениями, так называемыми, шаблонами. Эти шаблоны, подобраны специально таким образом, что позволяют значительно эффективней выявлять ошибки чтения/записи памяти. А затем проводится тестирование памяти перемещающимся разрядом. Также данный тест обеспечивает тестирование двух страниц памяти.

В каждый байт записывается 0,FF16,AA16(101010102) и 5516(010101012) и проверяется, могут ли они быть правильно считаны. В каждую позицию разряда каждого байта помещается 1 и проверяется, может ли быть эта единица правильно считана при всех остальных нулевых разрядах. Если все тесты работают правильно, то сбрасывается флаг переноса. При нахождении ошибки осуществляется немедленный выход из программы с установлением флага переноса и возвратом тестового значения и адреса, при котором произошла ошибка.

При проверке с одним значением (с 0,FF16,AA16 и 5516) сначала заполняется область памяти, а затем каждый байт сравнивается с заданным значением. Заполнение сначала всей области памяти должно обеспечить достаточный временной разрыв между записью и чтением, чтобы определить ошибки сохранения данных (которые могут возникнуть при неправильной разработке схем обновления). Затем, начиная с разряда 7, выполняется тест с перемещающимся разрядом; здесь записываются данные в память, и осуществляется попытка немедленно прочитать их обратно для сравнения. Далее все тесты повторяются для второй страницы памяти.

**3. Алгоритм теста**

Начало

I=0

I=N-1

Контроль

Конец

I=N-1

Счит [AI]

I=I+1

Изменение Т и повторение теста

Контроль

–

+

[AI]=E

Счит [AI]

Сдвиг Е

–

+

[AI]=T

I=0

I=I+1

Смена страницы памяти и повторение всех тестов

**4. ПРОГРАММА тестирования УСТРОЙСТВА НА АССЕМБЛЕРЕ**

Программа тестирования памяти по алгоритму «Марш»

START: LXI H,0 ;HL=базовый адрес

LXI D,65536 ;число байтов

CALL TEST ;вызов подпрограммы тестирования

MOV A,80h ;запись в аккумулятор значения

OUT OPORT ;включение второй страницы памяти

JMP START ;переход на начало тестирования

TEST:

;заполнение памяти числом 0 и проверка

MVI C,0 ;загрузка тестового значения

CALL LOOP ;вызов подпрограммы тестирования

RC ;выйти, если найдена ошибка

;заполнение памяти числом FFh и проверка

MVI C,0FFh ;загрузка тестового значения

CALL LOOP ;вызов подпрограммы тестирования

RC ;выйти, если найдена ошибка

;заполнение памяти числом AAh и проверка

MVI C,0ААh ;загрузка тестового значения

CALL LOOP ;вызов подпрограммы тестирования

RC ;выйти, если найдена ошибка

;заполнение памяти числом 55h и проверка

MVI C,55h ;загрузка тестового значения

CALL LOOP ;вызов подпрограммы тестирования

RC ;выйти, если найдена ошибка

;тест с перемещающимся разрядом

 CALL N3 ;вызов подпрограммы тестирования

 RC ;выйти, если найдена ошибка

 RET ;выйти в программу

LOOP: PUSH H ;сохранить базовый адрес

 PUSH D ;сохранить размер области

;заполнение памяти тестовым значением

N1: MOV M,C ;заполнить байт тестовым значением

 INX H ;следующий байт

 DCX D ;уменьшить и проверить 16-разрядный

 MOV A,D ;счётчик байтов

 ORA E ;сброс флага

 JNZ N1 ;продолжать, пока не заполнена вся память

 POP D ;восстановить размер области

 POP H ;восстановить базовый адрес

 PUSH H ;сохранить базовый адрес

 PUSH D ;сохранить размер области

;сравненне памяти и тестового значения

N2: MOV A,M ;взять значение из памяти

 CMP C ;сравнение с тестовым значением

 JNZ ERR ;нет, выйти, указав на ошибку

 INX H ;следующий байт

 DCX D ;уменьшить и проверить 16-разрядный

 MOV A,D ;счётчик байтов

 ORA E ;сброс флага

 JNZ N2 ;продолжать до сравнения всех байтов

;ошибка не найдена

 POP D ;DE=размер области в байтах

 POP H ;HL=базовый адрес

RET ;выйти (ORA E оставляет флаг переноса равным нулю)

 ;выход по ошибке, установить флаг переноса

 ;HL=адрес ошибки

 ;A=тестовое значение

N3: MVI A,10000000B ;разряд 7 равен единице, остальные – нулю

RBIT: MOV M,A ;записать шаблон теста в память

 CMP M ;прочитать его

 STC ;флаг переноса, на случай ошибки

 RNZ ;выйти при ошибке

 RRC ;циклический сдвиг шаблона

 CPI 10000000B ;сравнение

 JNZ RBIT ;продолжать, пока единица не вернётся в разряд 7

 MVI M,0 ;очистить уже проверенный байт

 INX H ;уменьшить и

 DCX D ;проверить 16-разрядный

 MOV A,D ;счётчик

 ORA E ;сброс флага

 JNZ N3 ;продолжать для всей проверяемой памяти

 RET ;возврат

ERR: MOV A,C ;А=тестовое значение

 POP D ;DE=размер области в байтах

 POP B ;BC=базовый адрес

 STC ;установка флага переноса, указывающего на ошибку

 RET ;возврат

 END ;окончание программы

**СПИСОК ИСПОЛЬЗОВАННОЙ литературы**

1. Полупроводниковые БИС запоминающих устройств Гордонов А.Ю., Бекин Н.В. Москва, “Радио и связь” 1990 384 с.
2. Микропроцессоры и микроЭВМ в системах автоматического управления С.Т.Хвощ, Н.Н. Варлинский, Е.А.Попов, Ленинград “Машиностроение”, 1987 640 с.
3. Коффрон Дж. Технические средства МП систем “Мир”, 1983 344 c.
4. Коффрон Дж., Лонг В., Расширение МП систем “Машиностроение”, 1987 354 с.
5. Б.В. Шевкопляс Микропроцессорные структуры, инженерные решения. Москва “Радио и связь”, 1990, 512 с.
6. Программирование на языке ассемблера для микропроцессоров 8080 и 8085

Л.А. Левенталь, У. Сэйвилл, Москва “Радио и связь”,1987 444 с.

1. Методические указания к лабораторной работе №2 по курсу “Цифровые устройства и микропроцессоры” В.А. Добряк, В.К. Рагозин. Екатеринбург: Изд-во УГТУ, 1999. 32 с.

GND

GND

DD1

Z80

30

31

32

33

34

35

36

37

38

39

40

1

2

3

4

5

14

15

12

8

7

9

10

13

27

19

20

22

21

28

18

24

16

17

26

25

23

6

A0

A1

A2

A3

A4

A5

A6

A7

A8

A9

A10

A11

A12

A13

A14

A15

D0

D1

D2

D3

D4

D5

D6

D7

M1

MREQ

IORQ

WR

RD

REFSH

HALT

WAIT

INT

NMI

RESET

BUSRQ

BUSAK

CLK

UM621024A

12

11

10

9

8

7

6

5

27

26

23

25

4

28

29

3

2

13

14

15

17

18

19

20

21

1

22

24

31

30

A0

A1

A2

A3

A4

A5

A6

A7

A8

A9

A10

A11

A12

A13

A14

A15

A16

DQ0

DQ1

DQ2

DQ3

DQ4

DQ5

DQ6

DQ7

VPP

CE

OE

WE

RP

74F794/SO

1

11

12

13

14

15

16

17

18

19

2

3

4

5

6

7

8

9

OE

CLK

Q7

Q6

Q5

Q4

Q3

Q2

Q1

Q0

D0

D1

D2

D3

D4

D5

D6

D7

&

K155ЛА2

1

2

8

3

4

5

6

11

12

КР580ВК28

6

8

10

12

15

17

19

21

2

3

4

22

5

7

9

11

13

16

18

20

24

25

26

27

28

14

D0

D1

D2

D3

D4

D5

D6

D7

HLDA

WR

DBIN

BUSEN

DB0

DB1

DB2

DB3

DB4

DB5

DB6

DB7

MEMR

I/O R

MEMW

I/O OW

VCC

GND

1

K133ЛЛ1

1

2

3

&

К133ЛЛ2

1

2

3

DD5

DD6

DD8

DD7

DD9

DD10

5В

D

C

C

D

Q

Q

5

4

6

Q

7474

7474

7400

DD2

DD3

DD4