

Учреждение образования  
«Белорусский государственный университет информатики и  
радиоэлектроники»

Факультет вечернего, заочного и дистанционного обучения  
Кафедра ЭВМ

Контрольная работа № 2  
по дисциплине «Микропроцессорные средства и системы»  
студента 5 курса 500502 учебной группы  
Авсеева С.П.

Минск 2010

# **Содержание**

<b>1</b>	<b>Техническое задание</b>	<b>2</b>
<b>2</b>	<b>Краткое описание системы</b>	<b>3</b>
<b>3</b>	<b>Краткое описание элементов схемы электрической принципиальной</b>	<b>4</b>
<b>4</b>	<b>Программа для процессоров группы 1</b>	<b>6</b>
<b>5</b>	<b>Программа для процессора общей группы</b>	<b>7</b>

# **1. Техническое задание**

1. Сосредоточенная система, 6 процессоров Z80 в 2 группах
2. ПЗУ 11 кб
3. ОЗУ 13 кб
4. Внешняя память 32 МБ
5. Интерфейс ввода/вывода LPT

## 2. Краткое описание системы

На рис. 2.1 приведена структурная схема сосредоточенной системы. Обмен между группами и общими ресурсами производится через шину 2. Шина 1 является рабочей внутри группы 1, шина 3 является рабочей для группы 2. Через шины передаётся 16-бит адреса и 8-бит данных.

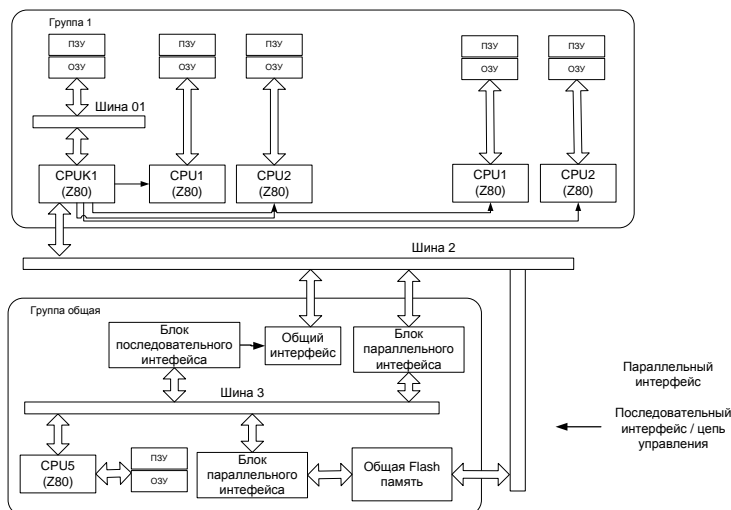


Рис. 2.1: Структурная схема сосредоточенной системы

### **3. Краткое описание элементов схемы электрической принципиальной**

Назначение элементов схемы для группы 1:

- DD2, DD15, DD17, DD20, DD21 — процессоры группы.
- DD10, DD12, DD13 — микросхемы локальной ПЗУ для CPU коммутирующего.
- DD33, DD35, DD36 — микросхемы локальной ПЗУ для CPU1.
- DD42, DD44, DD45 — микросхемы локальной ПЗУ для CPU2.
- DD52, DD54, DD55 — микросхемы локальной ПЗУ для CPU3.
- DD58, DD60, DD61 — микросхемы локальной ПЗУ для CPU4.
- DD8, DD11, DD14 — микросхемы локальной ОЗУ для CPU коммутирующего.
- DD32, DD34, DD37 — микросхемы локальной ОЗУ для CPU1.
- DD41, DD43, DD46 — микросхемы локальной ОЗУ для CPU2.
- DD51, DD53, DD56 — микросхемы локальной ОЗУ для CPU3.
- DD57, DD59, DD62 — микросхемы локальной ОЗУ для CPU4.
- DD24 — контроллер параллельного интерфейса для шин.
- ZQ1, ZQ2 — кварцевые генераторы частоты для процессоров групп.

Назначение элементов схемы для общей группы:

- DD2 — CPU устройства управления.
- DD14, DD16, DD17 — микросхемы локальной ПЗУ.
- DD13, DD15, DD18 — микросхемы локальной ОЗУ.
- DD20, DD21 — микросхемы общей памяти (flash).
- DD22 — контроллер общего интерфейса.
- ZQ1 — кварцевый генератор частоты для процессора.

Распределение адресного пространства локальной памяти для групп:

- 1000H-1FFFH — локальное ПЗУ

- 2000H-4FFFFH — локальное ОЗУ
- 8000H-83FFFH — локальное ОЗУ
- 8400H-8FFFFH — локальное ПЗУ

Порты ввода/вывода:

- 18h — порт ввода/вывода по шине,
- 28h — порт ввода/вывода по общей памяти,
- 38h — порт ввода/вывода по общему интерфейсу.

## 4. Программа для процессоров группы 1

Программа выполняет приём байта из общей памяти через параллельный интерфейс в локальную память. При каждом обращении к процессору производится инкремент адреса локальной памяти

```
BB55:      .equ 18h      ;номер порта для работы с микросхемой KP580BB55
BB55_2:    .equ 28h      ;номер порта для записи в группу
PORTA:     .equ 0        ;порт А микросхемы KP580BB55
PORTB:     .equ 1        ;порт В микросхемы KP580BB55
PORTC:     .equ 2        ;порт С микросхемы KP580BB55
BB55_CTL:  .equ 3        ;регистр управления микросхемы KP580BB55

entry:
    .org 0000h
    jp entry          ;переход на начало программы
    .org 0100h
entry:
    ld  A, 9Bh
    out (BB55 + BB55_CTL) ;конфигурирование BB55 на ввод
    ld  A, 0h
    out (BB55_2 + BB55_CTL) ;конфигурирование BB55 на вывод
copy1:
    ld  bc, 2000h      ;начальный адрес локальной памяти
copybyte:
    in  l, (BB55+PORTA)
    in  h, (BB55+PORTB)
    in  a, (BB55+PORTC)
    out (BB55_2+PORTA), l ;передача байтов в защёлки для второй группы
    out (BB55_2+PORTB), h
    out (BB55_2+PORTC), a
    ld  d, a          ;сохраняем принятое значение
    ld  a, b
    cp  H              ;если принятый адрес не совпадает с указателем
    jp  nz, getnext   ;переключаемся на другой процессор
    ld  a, c
    cp  l
    jp  nz, getnext
    ld  a, d          ;приём байта данных, если адреса совпали
    ld  (bc), a      ;записываем байт в локальную память
    inc bc
getnext:
    halt              ;ожидаем разрешающего сигнала от второго процессора группы
    ld  a, c
    cp  4Fh           ;достигли конца локальной памяти?
    jr  nz copybyte   ;если, продолжаем
    jp  copy1
    .end
```

## 5. Программа для процессора общей группы

Программа выполняет приём байта из общей памяти через параллельный интерфейс в локальную память. При каждом обращении к процессору производится инкремент адреса локальной памяти

```
BB55:      .equ 18h    ;номер порта для работы с микросхемой KP580BB55
BB55_2:    .equ 28h    ;номер порта для записи во вторую группу
PORTA:     .equ 0      ;порт А микросхемы KP580BB55
PORTB:     .equ 1      ;порт В микросхемы KP580BB55
PORTC:     .equ 2      ;порт С микросхемы KP580BB55
BB55_CTL:  .equ 3      ;регистр управления микросхемы KP580BB55
BB55_0:    .equ 38h    ;номер порта для работы с микросхемой общего
интерфейса

entry:
    .org 0000h
    jp     entry      ;переход на начало программы
    .org 0100h
entry:
    ld     a, 9Bh
    out   (BB55 + BB55_CTL) ;конфигурирование BB55 на ввод
    ld     a, 0h
    out   (BB55_2 + BB55_CTL) ;конфигурирование BB55 на вывод
    out   (BB55_0 + BB55_CTL) ;конфигурирование контроллера общего
                                ;интерфейса на вывод

copy1:
    ld     bc, 2000h    ;начальный адрес локальной памяти
copybyte:
    in     l, (BB55+PORTA)
    in     h, (BB55+PORTB)
    in     a, (BB55+PORTC)
    out   (BB55_2+PORTA), l ;передача байтов в защёлки для второй группы
    out   (BB55_0+PORTA), l ;передача байтов в защёлки для общего
                                ;интерфейса
    out   (BB55_2+PORTB), h
    out   (BB55_0+PORTB), h
    out   (BB55_2+PORTC), a
    out   (BB55_0+PORTC), a
    ld     a, b
    cp     h            ;если принятый адрес не совпадает с указателем
    jp     nz, getnext ;переключаемся на другой процессор
    ld     a, c
    cp     l
    jp     nz, getnext
    ld     a, d        ;приём байта данных, если адреса совпали
    ld     (bc), a    ;записываем байт в локальную память
    inc   bc
```



```
getnext:
    halt                ;ждем разрешающего сигнала от второго
                       ;процессора группы
    ld    a, c
    cp    4Fh           ;достигли конца локальной памяти?
    jr   nz copybyte   ;если, продолжаем
    jp    copy1
    .end
```