

**Московский авиационный институт
(национальный исследовательский
университет)**

Институт №3.

«Системы управления, информатика и электроэнергетика»

Кафедра №304

«Автоматизированные системы обработки информации и
управления»

Отчет по Лабораторной работа № 4.

по учебной дисциплине

«Архитектуры вычислительных систем»

на тему

«Моделирование группно-ассоциативной буферной памяти»

Группа М30-307Б

Выполнили:

Гордеев Н.

Принял:

Ходоровский А. З.

Грабовский М. Н.

Цель работы

Изучение функционирования группо-ассоциативного буфера со сквозной записью.

Задание к лабораторной работе

Все возможные варианты разбиты на три группы в соответствии с чем задаются исходные данные:

Выполнение работы

Работы содержит две части: буферная память с признаковым обменом и со сквозной записью. Первая часть представлена в виде примера и содержит код готовой программы. Ознакомившись с ее работой, можно приступить ко второй части, где **необходимо изменить исходный код и показать работу буферной памяти со сквозной записью, при этом программа запускается при выбранном пункте «Сквозная запись», пункт «Признаковый обмен» служит в качестве примера в обучающих целях.**

Для вариантов 6-10 задаются следующие данные:

Номер обращения к памяти	Исполнительный адрес			Код операции	
	адрес	N сегмента	N блока		
1	10	1	2	чтение	1
2	30	3	6	запись	0
3	6	0	6	чтение	1

Содержимое ОП и БП различно для каждого варианта.

Вариант 8

Содержимое буферной памяти :

Сектор 0				Сектор 1		
Блок	Сег.	Изм.	Данные	Сег.	Изм.	Данные
0	1	0	20	2	0	28
1	1	0	21	2	0	29
2	1	0	22	2	0	30
3	1	0	23	2	0	31
4	1	0	24	2	0	32
5	1	0	25	2	0	33
6	1	0	26	2	0	34
7	1	0	27	2	0	35

Содержимое оперативной памяти:

Блок #	0	1	2	3	4	5	6	7
Сегмент #0	12	13	14	15	16	17	18	19
Сегмент #1	20	21	22	23	24	25	26	27
Сегмент #2	28	29	30	31	32	33	34	35
Сегмент #3	36	37	38	39	40	41	42	43

! N обращ. ! Числа для ! N вытесн. !
! ! записи ! сегмента !

! 1 ! 12 ! 1 !
! 2 ! 13 ! 2 !
! 3 ! 14 ! 3 !

Программа на языке MCL, моделирующая функционирование селекторного канала

```
writeln " *****",
writeln " * Модель группо-ассоциативной КЭШ памяти со сквозной записью *",
writeln " *****",
{ работа моделируется на уровне блоков }
{***** ОПИСАНИЕ ТИПОВ*****}
{ типы используемые п/п }
var i,j,k,l, { вспомогательные переменные }
    choice, { номер варианта }
    op_code, { код операции }
    data, { данные }
    replace, { номер замещаемого сегмента }
    s_no; { номер сектора, содержащего необходимый сегмент }
mem op[32][8]; { оперативная память 32 слова по 8 бит }
mem bp[16][10]; { буферная память 16 слов по 11 бит }
mem ispadr[3][5]; { массив для хранения исполнительных адресов }
mem inputs[3][8]; { массив для хранения чисел, которые должны быть записаны в БП }
mem coper[3][1]; { массив для хранения кодов операций 1-чтение, 0-запись }
mem rep_seg[3][2]; { массив для хранения номеров сегментов подлежащих вытеснению }
struct iadr[5] { структура для разбиения исполнительного адреса на старшую и младшую часть }
    blok[3], { номер блока }
    seg [2]; { номер сегмента }
struct bpadr[4] { структура для выделения старшей и младшей части в адресе БП }
    blok[3], { номер блока }
    sect[1]; { номер сектора }
struct buf_reg[10] { структура для выделения данных, бита изменения и бита присутствия }
    data[8], { данные }
    address[2]; { номер сегмента }
struct subiadr[5] { переменная используется в подпрограммах }
    blok[3],
    seg [2];
struct subbpadr[4] { переменная используется в подпрограммах }
    blok[3],
    sect[1];
struct subbuf_reg[10]{ переменная используется в подпрограммах }
    data[8],
    address[2];

{***** начало программы *****}
gosub variant; { задание начальных значений всех переменных }
for l:=0 to 2; { цикл для работы с 3-мя исполнительными адресами }
{задаем случайное значение номера сектора не равное 0 или 1}
s_no:=2;
{получаем исполнительный адрес}
iadr:=ispadr[l];
{получаем код операции}
op_code:=coper[l];
{загружаем данные для записи}
data:=inputs[l];
{получаем номер вытесняемого сегмента}
replace:=rep_seg[l];
writeln "Исполнительный адрес - ", $d2 iadr;
pause;
bpadr.blok:=iadr.blok; { выделение блоков с нужным номером, номер блока в БП = номеру блока в
исполнительном адресе}
{ Поиск наличия блока в БП }
```

```

gosub poisk_bloka;
if s_no=0 then goto get_blok; { Если блок нужного сегмента присутствует в БП, то продолжаем работу с
ним}
if s_no=1 then goto get_blok;
{ Если блока нет, то переходим к вытеснению }
write "Блок ", $d2 iadr.blok;
writeln " ,сегмента " , $d2 iadr.seg, " отсутствует в БП";
pause;
writeln "Вытеснение блока, принадлежащего сегменту" , $d2 replace;
pause;
gosub vitesnenie;
get_blok:
writeln "Блок " , $d2 iadr.blok, " сегмента " , $d2 iadr.seg, " присутствует в БП";
pause;
{ выполнение операции по коду }
if op_code=1 then gosub read_data; { если 1 то чтение}
if op_code=0 then gosub write_data; { если 0 то запись}
goto end_job;
end_job:
gosub print_bp; { печать содержимого БП }
gosub print_op; { печать содержимого ОП }
pause;
next;
end;
{***** конец программы *****)
{***** подпрограммы *****)

{ п/п для задания варианта, использует п/п first,second,third }
variant:
{ задание исходных данных по вариантам}
read " Введите номер варианта: ",choice;
if choice<=5 then gosub first;
if ((choice>5) and (choice<=10)) then gosub second;
if choice>10 then gosub third;
return; { variant }
{ п/п первый вариант }
first:
{ присваивание начальных значений ячейкам ОП }
for i:=0 to 31;
    op[i]:=(choice+2)*2+i-2;
next;
{ присваивание начальных значений ячейкам БП }
for i:=0 to 7;
    buf_reg.data:=(choice+2)*2+i-2;
    buf_reg.ch_bit:=0;
    buf_reg.address:=0;
    bp[i]:=buf_reg;
next;
for i:=8 to 15;
    buf_reg.data:=(choice+2)*2+i-2;
    buf_reg.ch_bit:=0;
    buf_reg.address:=1;
    bp[i]:=buf_reg;
next;
{ назначение исполнительных адресов }
ispadr[0]:=4;
ispadr[1]:=28;
ispadr[2]:=22;
{ задание чисел для записи }

```

```

for i:=0 to 2;
  inputs[i]:=i+8;
next;
{ задание КОП }
coper[0]:=0;
coper[1]:=1;
coper[2]:=0;
{ задание номеров замещаемых сегментов }
rep_seg[0]:=0;
rep_seg[1]:=0;
rep_seg[2]:=1;
gosub print_bp; { печать содержимого БП }
pause;
gosub print_op; { печать содержимого ОП }
gosub print; { печать записываемых чисел и замещаемых сегментов }
pause;
return; { first }
{ п/п второй вариант }
second:
{ присваивание начальных значений ячейкам ОП }
for i:=0 to 31;
  op[i]:=(choice+2)*2+i-8;
next;
{ присваивание начальных значений ячейкам БП }
for i:=0 to 7;
  buf_reg.data:=(choice+2)*2+i;
  buf_reg.address:=1;
  bp[i]:=buf_reg;
next;
for i:=8 to 15;
  buf_reg.data:=(choice+2)*2+i;
  buf_reg.address:=2;
  bp[i]:=buf_reg;
next;
{ назначение исполнительных адресов }
ispadr[0]:=10;
ispadr[1]:=30;
ispadr[2]:=6;
{ задание чисел для записи }
for i:=0 to 2;
  inputs[i]:=i+12;
next;
{ задание КОП }
coper[0]:=1;
coper[1]:=0;
coper[2]:=1;
{ задание номеров замещаемых сегментов }
rep_seg[0]:=1;
rep_seg[1]:=2;
rep_seg[2]:=3;
gosub print_bp; { печать содержимого БП }
pause;
gosub print_op; { печать содержимого ОП }
gosub print; { печать записываемых чисел и замещаемых сегментов }
pause;
return; { second }
{ п/п третий вариант }
third:
{ присваивание начальных значений ячейкам ОП }

```

```

for i:=0 to 31;
  op[i]:=(choice+2)/2+i+5;
next;
{ присваивание начальных значений ячейкам БП }
for i:=0 to 7;
  buf_reg.data:=(choice+2)/2+i+21;
  buf_reg.ch_bit:=0;
  buf_reg.address:=2;
  bp[i]:=buf_reg;
next;
for i:=8 to 15;
  buf_reg.data:=(choice+2)/2+i+21;
  buf_reg.ch_bit:=0;
  buf_reg.address:=3;
  bp[i]:=buf_reg;
next;
{ назначение исполнительных адресов }
ispadr[0]:=8;
ispadr[1]:=26;
ispadr[2]:=10;
{ задание чисел для записи }
for i:=0 to 2;
  inputs[i]:=i+23;
next;
{ задание КОП }
coper[0]:=0;
coper[1]:=0;
coper[2]:=1;
{ задание номеров замещаемых сегментов }
rep_seg[0]:=2;
rep_seg[1]:=2;
rep_seg[2]:=3;
gosub print_bp; { печать содержимого БП }
pause;
gosub print_op; { печать содержимого ОП }
gosub print; { печать записываемых чисел и замещаемых сегментов }
pause;
return; { third }
{ п/п печати содержимого БП }
print_bp:
  writeln "Содержимое буферной памяти :";
  writeln "    Сектор 0      Сектор 1";
  writeln "Блок  Сег.  Данные  Сег.  Данные";
  for i:=0 to 7;
    write " ", $d2 i;
    subbpadr.sect:=0;
    subbpadr.blok:=i;
    subbuf_reg:=bp[subbpadr];
    gosub print_br;
    subbpadr.sect:=1;
    subbuf_reg:=bp[subbpadr];
    gosub print_br;
    writeln;
  next;
return; { print_bp }
{ п/п печати буферного регистра }
print_br:
  write "    ", $b1 subbuf_reg.address;
  write "    ", $d4 subbuf_reg.data;

```

```

return; { print_br }
{ п/п печати содержимого ОП}
print_op:
  writeln "Содержимое оперативной памяти:";
  writeln " Блок #   0 1 2 3 4 5 6 7";
  for i:=0 to 3;
    write " Сегмент #", $d1 i, " ";
    subiadr.seg:=i;
    for j:=0 to 7;
      subiadr.blok:=j;
      write $d4 op[subiadr];
    next;
  writeln;
next;
return; { print_op }
{ п/п печати записываемых чисел и замещаемых сегментов }
print:
  writeln "-----";
  writeln "! № обращ. ! Числа для ! № вытесн. !";
  writeln "!       ! записи   ! сегмента !";
  writeln "-----";
  writeln "!   1   !", $d7 inputs[0], "   !", $d6 rep_seg[0], "   !";
  writeln "!   2   !", $d7 inputs[1], "   !", $d6 rep_seg[1], "   !";
  writeln "!   3   !", $d7 inputs[2], "   !", $d6 rep_seg[2], "   !";
  writeln "-----";
return; { print }
poisk_bloka:
  bpadr.sect:=0;
  buf_reg:=bp[bpadr];
  if buf_reg.address=iadr.seg then s_no:=0;
  bpadr.sect:=1;
  buf_reg:=bp[bpadr];
  if buf_reg.address=iadr.seg then s_no:=1;
return; { poisk_bloka }
vitesnenie:
  { Поиск вытесняемого блока}
  bpadr.sect:=0;
  buf_reg:=bp[bpadr];
  if buf_reg.address=replace then s_no:=0;
  bpadr.sect:=1;
  buf_reg:=bp[bpadr];
  if buf_reg.address=replace then s_no:=1;
  { Вытеснение сегмента }
  bpadr.sect:=s_no;           { нахождение сектора}
  buf_reg:=bp[bpadr];         { считывание в буферный регистр}
  buf_reg.address:=iadr.seg; { занесение номера сегмента в адресную часть}
  buf_reg.data:=op[iadr]; { запись адресуемого блока из ОП}
  bp[bpadr]:=buf_reg;
  pause;
  k:=iadr;
  iadr.seg:=buf_reg.address;
  op[iadr]:=buf_reg.data;
  iadr:=k;
  pause;
return; { vitesnenie }
write_data:
  bpadr.sect:=s_no; { нахождение адреса }
  buf_reg:=bp[bpadr]; { чтение блока из БП в буферный регистр }
  buf_reg.data:=data;

```

```

bp[bpadr]:=buf_reg; { запись из регистра в буферную память }
{*****}
op[iadr]:=data;
{*****}
writeln "Записаны данные ", $d4 data;
pause;
return; {write data}
read_data:
bpadr.sect:=s_no; { нахождение адреса }
buf_reg:=bp[bpadr]; { чтение блока из БП в буферный регистр }
data:=buf_reg.data;
writeln "Прочитаны данные ", $d4 data;
pause;
return; {read_data}}

```


Результаты выполнения программы в виде распечатки содержимого поля для вывода текстовой информации и скриншота в момент завершения выполнения программы.

* Модель групп-ассоциативной КЭШ памяти со сквозной записью *

Введите номер варианта:

8

Содержимое буферной памяти :

		Сектор 0		Сектор 1	
Блок	Сег.	Данные	Сег.	Данные	
0	1	20	0	28	
1	1	21	0	29	
2	1	22	0	30	
3	1	23	0	31	
4	1	24	0	32	
5	1	25	0	33	
6	1	26	0	34	
7	1	27	0	35	

Содержимое оперативной памяти:

Блок #	0	1	2	3	4	5	6	7
Сегмент #0	12	13	14	15	16	17	18	19
Сегмент #1	20	21	22	23	24	25	26	27
Сегмент #2	28	29	30	31	32	33	34	35
Сегмент #3	36	37	38	39	40	41	42	43

! № обращ. ! Числа для ! № вытесн. !

! ! записи ! сегмента !

!	1	!	12	!	1	!
!	2	!	13	!	2	!
!	3	!	14	!	3	!

Исполнительный адрес - 10

Блок 2 сегмента 1 присутствует в БП

Прочитаны данные 22

Содержимое буферной памяти :

		Сектор 0		Сектор 1	
Блок	Сег.	Данные	Сег.	Данные	
0	1	20	0	28	
1	1	21	0	29	
2	1	22	0	30	
3	1	23	0	31	
4	1	24	0	32	
5	1	25	0	33	
6	1	26	0	34	
7	1	27	0	35	

Содержимое оперативной памяти:

Блок #	0	1	2	3	4	5	6	7
Сегмент #0	12	13	14	15	16	17	18	19
Сегмент #1	20	21	22	23	24	25	26	27
Сегмент #2	28	29	30	31	32	33	34	35
Сегмент #3	36	37	38	39	40	41	42	43

Исполнительный адрес - 30

Блок 6 ,сегмента 3 отсутствует в БП

Вытеснение блока, принадлежащего сегменту 2

Блок 6 сегмента 3 присутствует в БП

Записаны данные 13

Содержимое буферной памяти :

Сектор 0 Сектор 1

Блок Сег. Данные Сег. Данные

0	1	20	0	28
1	1	21	0	29
2	1	22	0	30
3	1	23	0	31
4	1	24	0	32
5	1	25	0	33
6	1	26	1	13
7	1	27	0	35

Содержимое оперативной памяти:

Блок # 0 1 2 3 4 5 6 7

Сегмент #0 12 13 14 15 16 17 18 19

Сегмент #1 20 21 22 23 24 25 26 27

Сегмент #2 28 29 30 31 32 33 34 35

Сегмент #3 36 37 38 39 40 41 13 43

Исполнительный адрес - 6

Блок 6 ,сегмента 0 отсутствует в БП

Вытеснение блока, принадлежащего сегменту 3

Блок 6 сегмента 0 присутствует в БП

Прочитаны данные 18

Содержимое буферной памяти :

Сектор 0 Сектор 1

Блок Сег. Данные Сег. Данные

0	1	20	0	28
1	1	21	0	29
2	1	22	0	30
3	1	23	0	31
4	1	24	0	32
5	1	25	0	33
6	1	26	0	18
7	1	27	0	35

Содержимое оперативной памяти:

Блок # 0 1 2 3 4 5 6 7

Сегмент #0 12 13 14 15 16 17 18 19

Сегмент #1 20 21 22 23 24 25 26 27

Сегмент #2 28 29 30 31 32 33 34 35

Сегмент #3 36 37 38 39 40 41 13 43

Содержимое таблицы переменных

00 Переменная: ZF типа регистр; разрядность 1 бит; значение = 1

01 Переменная: NF типа регистр; разрядность 1 бит; значение = 0

02 Переменная: CF типа регистр; разрядность 1 бит; значение = 0

03 Переменная: I типа простая переменная; Значение = 4

04 Переменная: J типа простая переменная; Значение = 8

05 Переменная: K типа простая переменная; Значение = 6

06 Переменная: L типа простая переменная; Значение = 3

07 Переменная: CHOICE типа простая переменная; Значение = 8

08 Переменная: OP_CODE типа простая переменная; Значение = 1

09 Переменная: DATA типа простая переменная; Значение = 18

10 Переменная: REPLACE типа простая переменная; Значение = 3

11 Переменная: S_NO типа простая переменная; Значение = 1

12 Переменная: OP типа память; из 32 ячеек по 8 бит

13 Переменная: BP типа память; из 16 ячеек по 10 бит

14 Переменная: ISPADR типа память; из 3 ячеек по 5 бит

15 Переменная: INPUTS типа память; из 3 ячеек по 8 бит

16 Переменная: COPER типа память; из 3 ячеек по 1 бит

17 Переменная: REP_SEG типа память; из 3 ячеек по 2 бит

- 18 Переменная: IADR типа структура; разрядность 5 бит; значение = 6
Поле: BLOK; разрядность 3 бит;
Поле: SEG; разрядность 2 бит;
- 19 Переменная: BPADR типа структура; разрядность 4 бит; значение = 14
Поле: BLOK; разрядность 3 бит;
Поле: SECT; разрядность 1 бит;
- 20 Переменная: BUF_REG типа структура; разрядность 10 бит; значение = 18
Поле: DATA; разрядность 8 бит;
Поле: ADDRESS; разрядность 2 бит;
- 21 Переменная: SUBIADR типа структура; разрядность 5 бит; значение = 31
Поле: BLOK; разрядность 3 бит;
Поле: SEG; разрядность 2 бит;
- 22 Переменная: SUBBPADR типа структура; разрядность 4 бит; значение = 15
Поле: BLOK; разрядность 3 бит;
Поле: SECT; разрядность 1 бит;
- 23 Переменная: SUBBUF_REG типа структура; разрядность 10 бит; значение = 547
Поле: DATA; разрядность 8 бит;
Поле: ADDRESS; разрядность 2 бит;

ОР Содержимое оперативной памяти:

Блок	0	1	2	3	4	5	6	7
Сегм...	12	13	14	15	16	17	18	19
Сегм...	20	21	22	23	24	25	26	27
Сегм...	28	29	30	31	32	33	34	35
Сегм...	36	37	38	39	40	41	13	43

БР Содержимое буферной памяти:

№ блока	Данные	Сегмент	Данные	Сегмент
0	20	1	28	2
1	21	1	29	2
2	22	1	30	2
3	23	1	31	2
4	24	1	32	2
5	25	1	33	2
6	26	1	18	0
7	27	1	35	2

[Исполнительный адрес](#)

№ сегмента

№ блока

Тип операции

Запись данных

№ вытесняемого сегмента

Считываемые данные