

**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 № 3.

по учебной дисциплине

«Архитектуры вычислительных систем»

на тему

«Моделирование секторной буферной памяти»

Группа М30-307Б

Выполнили:

Гордеев Н.

Принял:

Ходоровский А. З.

Грабовский М. Н.

Цель работы

Изучение функционирования блока секторного 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	19	2	3	чтение	1

Содержимое ОП и БП различно для каждого варианта.

Вариант 8

Содержимое буферной памяти :

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

Содержимое РНПС:

!	сектор	!	сегмент	!
!	0	!	1	!
!	1	!	2	!

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	!	0	!
!	2	!	13	!	1	!
!	3	!	14	!	0	!

Программа на языке MCL, моделирующая функционирование селекторного канала

```
writeln "*****";
writeln " * Модель секторной КЭШ памяти с признаковым обменом *";
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 }
pris,            { для п/п проверки присутствия блока в БП }
changed;         { для п/п проверки изменения блока в БП }
mem op[32][8];   { оперативная память 32 слова по 8 бит }
mem bp[16][9];   { буферная память 16 слов по 9 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 seg_present[2][2]; { регистры номеров присутствующих сегментов}
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[9] { структура для выделения данных, бита изменения и бита
присутствия*****}
data[8],          { данные }
pres_bit[1];      { бит присутствия }
struct subiadr[5] { переменная используется в подпрограммах }
blok[3],
seg [2];
struct subbpadr[4] { permennaa ispol'zuetsa v podprogrammah }
blok[3],
sect[1];
struct subbuf_reg[9] { permennaa ispol'zuetsa v podprogrammah
*****}
data[8],
pres_bit[1];

{***** 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;
{определяем номер сегмента на основе старших разрядов и номер сектора БП, содержащего сегмент}
if seg_present[0]=iadr.seg then s_no:=0;
{если сегмент в секторе 0, то переходим к проверке наличия блока}
if s_no=0 then goto get_blok;
{если сегмент в секторе 1, то переходим к проверке наличия блока}
if seg_present[1]=iadr.seg then s_no:=1;
if s_no=1 then goto get_blok;
{если сегмента нет в секторах 0 и 1, то переходим к вытеснению сегмента}
writeln "Сегмент ", $d2 iadr.seg, " отсутствует в буферной памяти"; pause;
writeln "Вытеснение сегмента ", $d2 replace; pause;
gosub vitesnenie;
get_blok:
{проверка есть ли блок в БП}
gosub pris;
{pris=0-блок отсутствует, pris=1-присутствует}
{если блок присутствует, то пропускаем подкачку блока}
if pris=1 then goto est_blok;
{если блок отсутствует, то подгружаем его из ОП}
writeln "Блок ", $d2 iadr.blok, " сегмента ", $d2 iadr.seg, " отсутствует в БП"; pause;
writeln "Чтение блока из ОП...";
gosub read_OP;
goto do_it;
est_blok:
writeln "Блок ", $d2 iadr.blok, " сегмента ", $d2 iadr.seg, " присутствует в БП";
goto do_it;
do_it: { выполнение операции по коду}
if op_code=1 then gosub read_data; { если 1 то чтение}
{ если 0 то запись}
if op_code=0 then gosub write_data;
gosub print_bp; { печать содержимого БП }
gosub print_op; { печать содержимого ОП }
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 15;

```

```

buf_reg.data:=(choice+2)*2+i-2;
buf_reg.ch_bit:=0;
buf_reg.pres_bit:=1;
bp[i]:=buf_reg;
next;
for i:=2 to 5;
  buf_reg.data:=choice+i-2;
  buf_reg.ch_bit:=0;
  buf_reg.pres_bit:=0;
  bp[i]:=buf_reg;
next;
{ назначение исполнительных адресов }
ispadr[0]:=4;
ispadr[1]:=29;
ispadr[2]:=15;
{ 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 присутствующих сегментов }
seg_present[0]:=0;
seg_present[1]:=1;
{ задание номеров замещаемых сегментов }
rep_seg[0]:=0;
rep_seg[1]:=0;
rep_seg[2]:=0;
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 15;
  buf_reg.data:=(choice+2)*2+i;
  buf_reg.pres_bit:=1;
  bp[i]:=buf_reg;
next;
for i:=0 to 3;
  buf_reg.data:=choice-4;
  buf_reg.pres_bit:=0;
  bp[i]:=buf_reg;
next;
{ назначение исполнительных адресов }
ispadr[0]:=10;
ispadr[1]:=30;
ispadr[2]:=19;
{ 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 присутствующих сегментов }
seg_present[0]:=1;
seg_present[1]:=2;
{ задание номеров замещаемых сегментов }
rep_seg[0]:=0;
rep_seg[1]:=1;
rep_seg[2]:=0;
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 15;
  buf_reg.data:=(choice+2)/2+i+21;
  buf_reg.ch_bit:=0;
  buf_reg.pres_bit:=1;
  bp[i]:=buf_reg;
next;
for i:=4 to 7;
  buf_reg.data:=choice-7;
  buf_reg.ch_bit:=0;
  buf_reg.pres_bit:=0;
  bp[i]:=buf_reg;
next;
{ назначение исполнительных адресов }
ispadr[0]:=21;
ispadr[1]:=8;
ispadr[2]:=24;
{ задание чисел для записи }
for i:=0 to 2;
  inputs[i]:=i+23;
next;
{ задание КОП }
coper[0]:=1;
coper[1]:=1;
coper[2]:=0;
{ задание номеров присутствующих сегментов }
seg_present[0]:=2;
seg_present[1]:=3;
{ задание номеров замещаемых сегментов }
rep_seg[0]:=0;
rep_seg[1]:=2;
rep_seg[2]:=0;
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 Изм. Прис. Данные Изм. Прис. 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;
writeln "Содержимое РНПС:";
writeln "-----";
writeln "! сектор ! сегмент !";
writeln "-----";
writeln "! 0 !", $d7 seg_present[0], " !";
writeln "! 1 !", $d7 seg_present[1], " !";
writeln "-----";
return; { print_bp }
{ п/п печати буферного регистра }
print_br:
write " ", $b1 subbuf_reg.pres_bit;
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 }
```

```
{ п/п определения присутствия блока в БП}
pris:
bpadr.blok:=iadr.blok;
bpadr.sect:=s_no;
buf_reg:=bp[bpadr]; { чтение блока из БП в буферный регистр }
if buf_reg.pres_bit=1 then pris=1; { если блок присутствует в БП }
if buf_reg.pres_bit=0 then pris=0; { если блок отсутствует в БП }
return; {pris}
```

```
{ п/п записи данных в БП}
write_data:
buf_reg.data:=data;
buf_reg.pres_bit:=1;
pause;
bp[bpadr]:=buf_reg;
{*****}
op[iadr]:=data;
{*****}
writeln "Записаны данные ", $d4 data;
pause;
return; {write_data}
```

```
{ п/п для чтения данных из БП}
read_data:
data:=buf_reg.data;
writeln "Прочитаны данные ", $d4 data;
pause;
return; {read_data}
```

```
{п/п чтения из ОП в БП}
read_OP:
buf_reg.data:=op[iadr];
buf_reg.pres_bit:=1;
bp[bpadr]:=buf_reg;
writeln "Блок прочитан из ОП";
pause;
return; {read_OP}
```

```
{п/п для вытеснения сегмента из сектора БП}
vitesnenie:
if seg_present[0]=replace then s_no:=0; { поиск номера сектора, в котором находится вытесняемый сегмент}
if seg_present[1]=replace then s_no:=1;
for j:=0 to 7; { вытеснение сегмента}
bpadr.sect:=s_no; { определение адреса в БП}
bpadr.blok:=j;
buf_reg:=bp[bpadr];
gosub zero_bits; {обнуляем биты присутствия и изменения}
next;
seg_present[s_no]:=iadr.seg; {заносим в РНПС номер нового сегмента}
writeln "Сегмент успешно вытеснен";
pause;
return; {vitesnenie}
```



```
{п/п записи из БП в ОП}  
write_OP:  
k:=iadr;  
iadr.blok:=bpadr.blok;  
iadr.seg:=seg_present[s_no];  
op[iadr]:=buf_reg.data;  
writeln "Сохранение блока ", $d2 iadr.blok, " сегмента ", $d2  
iadr.seg, " в ОП ";  
iadr:=k; pause;  
return; {write_OP}
```

```
{п/п для обнуления битов присутствия и изменения}  
zero_bits:  
buf_reg.pres_bit:=0;  
bp[bpadr]:=buf_reg;  
return; {zero_bits}
```

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

* Модель секторной КЭШ памяти с признаковым обменом*

Введите номер варианта:

8

Содержимое буферной памяти :

Сектор 0 Сектор 1

Блок Изм. Прис. Данные Изм. Прис. Данные

0 0 4 1 28

1 0 4 1 29

2 0 4 1 30

3 0 4 1 31

4 1 24 1 32

5 1 25 1 33

6 1 26 1 34

7 1 27 1 35

Содержимое РНПС:

! сектор ! сегмент !

! 0 ! 1 !

! 1 ! 2 !

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 ! 0

!

! 2 ! 13 ! 1

!

! 3 ! 14 ! 0

!

Исполнительный адрес - 10

Блок 2 сегмента 1 отсутствует в БП

Чтение блока из ОП...

Блок прочитан из ОП

Прочитаны данные 22

Содержимое буферной памяти :

Сектор 0 Сектор 1

Блок Изм. Прис. Данные Изм. Прис. Данные

0 0 4 1 28

1 0 4 1 29

2 1 22 1 30

3 0 4 1 31

4 1 24 1 32

5 1 25 1 33

6 1 26 1 34

7 1 27 1 35

Содержимое РНПС:

! сектор ! сегмент !

! 0 ! 1 !

! 1 ! 2 !

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

Сегмент 3 отсутствует в буферной памяти

Вытеснение сегмента 1

Сегмент успешно вытеснен

Блок 6 сегмента 3 отсутствует в БП

Чтение блока из ОП...

Блок прочитан из ОП

Записаны дынные 13

Содержимое буферной памяти :

Сектор 0 Сектор 1

Блок Изм. Прис. Данные Изм. Прис. Данные

0 0 4 1 28

1 0 4 1 29

2 0 22 1 30

3 0 4 1 31

4 0 24 1 32

5 0 25 1 33

6 1 13 1 34

7 0 27 1 35

Содержимое РНПС:

! сектор ! сегмент !

! 0 ! 3 !

! 1 ! 2 !

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 - 19

Блок 3 сегмента 2 присутствует в БП

Прочитаны данные 31

Содержимое буферной памяти :

Сектор 0 Сектор 1

Блок Изм. Прис. Данные Изм. Прис. Данные

0 0 4 1 28

1 0 4 1 29

2 0 22 1 30

3 0 4 1 31

4 0 24 1 32

5 0 25 1 33

6 1 13 1 34

7 0 27 1 35

Содержимое РНПС:

! сектор ! сегмент !

! 0 ! 3 !

! 1 ! 2 !

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

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 = 0

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 = 31

10 Переменная: REPLACE типа простая переменная; Значение = 0

11 Переменная: S_NO типа простая переменная; Значение = 1

12 Переменная: PRIS типа простая переменная; Значение = 1

13 Переменная: CHANGED типа простая переменная; Значение = 0

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

15 Переменная: BP типа память; из 16 ячеек по 9 бит

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

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

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

19 Переменная: SEG_PRESENT типа память; из 2 ячеек по 2 бит

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

21 Переменная: IADR типа структура; разрядность 5 бит; значение = 19

Поле: BLOK; разрядность 3 бит;

Поле: SEG; разрядность 2 бит;

22 Переменная: BPADR типа структура; разрядность 4 бит; значение = 11

Поле: BLOK; разрядность 3 бит;

Поле: SECT; разрядность 1 бит;

23 Переменная: BUF_REG типа структура; разрядность 9 бит; значение = 287

Поле: DATA; разрядность 8 бит;

Поле: PRES_BIT; разрядность 1 бит;

24 Переменная: SUBIADR типа структура; разрядность 5 бит; значение = 31

Поле: BLOK; разрядность 3 бит;

Поле: SEG; разрядность 2 бит;

25 Переменная: SUBBPADR типа структура; разрядность 4 бит; значение = 15

Поле: BLOK; разрядность 3 бит;

Поле: SECT; разрядность 1 бит;

26 Переменная: SUBBUF_REG типа структура; разрядность 9 бит; значение = 291

Поле: DATA; разрядность 8 бит;

Поле: PRES_BIT; разрядность 1 бит;

ОР	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

ВР Содержимое бчферной памяти:

Сектор № 0 - Сегмент № 3

Сектор № 1 - Сегмент № 2

	№ блока	Данные	Бит присутствия	Бит изменения	Данные	Бит присутствия	Бит изменения
▶	0	4	0		28	1	
	1	4	0		29	1	
	2	22	0		30	1	
	3	4	0		31	1	
	4	24	0		32	1	
	5	25	0		33	1	
	6	13	1		34	1	
	7	27	0		35	1	

Исполнительный адрес

№ Сегмента

10

№ Блока

011

Тип операции

Чтение

Запись данных

№ вытесняемого сегмента

Считываемые данные

31