

Министерство цифрового развития, связи и
массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего
образования «Сибирский государственный университет телекоммуникаций и
информатики» (СибГУТИ)

Кафедра вычислительных систем

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
по дисциплине «Архитектура ЭВМ»

Выполнил: студент
гр. ИС-241
«5» октября 2024 г.

/Арещенко Е.А. /

Проверил:
преподаватель
«__» октября
2024 г.

/Майданов Ю.С./

Оценка «_____»

Новосибирск 2024
ОГЛАВЛЕНИЕ

ПОСТАНОВКА ЗАДАЧИ.....	3
БЛОК-СХЕМЫ АЛГОРИТМОВ	6
РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ	8
ЗАКЛЮЧЕНИЕ	10
ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА	11

ПОСТАНОВКА ЗАДАЧИ

В рамках курсовой работы необходимо:

- Разработать транслятор с языка Simple Assembler. Итог работы транслятора – бинарный файл с образом оперативной памяти Simple Computer, который можно загрузить в модель и выполнить;
- Доработать модель Simple Computer – реализовать алгоритм работы блока «L1-кэш команд и данных» и модифицировать работу контроллера оперативной памяти и обработчика прерываний таким образом, чтобы учитывался простой процессора при прямом доступе к оперативной памяти;
- Разработать транслятор с языка Simple Basic. Итог работы транслятора – текстовый файл с программой на языке Simple Assembler.

Транслятор с языка Simple Assembler

Разработка программ для Simple Computer может осуществляться с использованием низкоуровневого языка Simple Assembler. Для того чтобы программа могла быть обработана Simple Computer необходимо реализовать транслятор, переводящий текст Simple Assembler в бинарный формат, которым может быть считан консолью управления. Пример программы на Simple Assembler:

00 READ 09 ; (Ввод A)
01 READ 10 ; (Ввод B)
02 LOAD 09 ; (Загрузка A в аккумулятор)
03 SUB 10 ; (Отнять B)
04 JNEG 07 ; (Переход на 07, если отрицательное)
05 WRITE 09 ; (Вывод A)
06 HALT 00 ; (Останов)
07 WRITE 10 ; (Вывод B)
08 HALT 00 ; (Останов)
09 = +0000 ; (Переменная A) 10 = +9999 ; (Переменная B) Программа транслируется по строкам, задающим значение одной ячейки памяти. Каждая строка состоит как минимум из трех полей: адрес ячейки памяти, команда (символьное обозначение), операнд. Четвертым полем может быть указан комментарий, который обязательно должен начинаться с символа точка с запятой. Название команд представлено в таблице 1. Дополнительно используется команда =, которая явно задает значение ячейки памяти в формате вывода его на экран консоли (+XXXX).

Команда запуска транслятора должна иметь вид: sat файл.sa файл.o, где файл.sa – имя файла, в котором содержится программа на Simple Assembler, файл.o – результат трансляции.

Транслятор с языка Simple Basic

Для упрощения программирования пользователю модели Simple Computer должен быть предоставлен транслятор с высокоуровневого языка Simple Basic. Файл, содержащий программу на Simple Basic, преобразуется в файл с кодом Simple Assembler. Затем Simple Assembler-файл транслируется в бинарный формат.

В языке Simple Basic используются следующие операторы: rem, input, output, goto, if, let, end. Пример программы на Simple Basic:

```
10 REM Это комментарий
20 INPUT A
30 INPUT B
40 LET C = A - B
50 IF C < 0 GOTO 20
60 OUTPUT C
70 END
```

Каждая строка программы состоит из номера строки, оператора Simple Basic и параметров. Номера строк должны следовать в возрастающем порядке. Все команды за исключением команды конца программы могут встречаться в программе многократно. Simple Basic должен оперировать с целыми выражениями, включающими операции +, -, *, и /. Приоритет операций аналогичен C. Для того чтобы изменить порядок вычисления, можно использовать скобки.

Транслятор должен распознавать только букв верхнего регистра, то есть все символы в программе на Simple Basic должны быть набраны в верхнем регистре (символ нижнего регистра приведет к ошибке). Имя переменной может состоять только из одной буквы. Simple Basic оперирует только с целыми значениями переменных, в нем отсутствует объявление переменных, а упоминание переменной автоматически вызывает её объявление и присваивает ей нулевое значение. Синтаксис языка не позволяет выполнять операций со строками.

БЛОК-СХЕМЫ АЛГОРИТМОВ

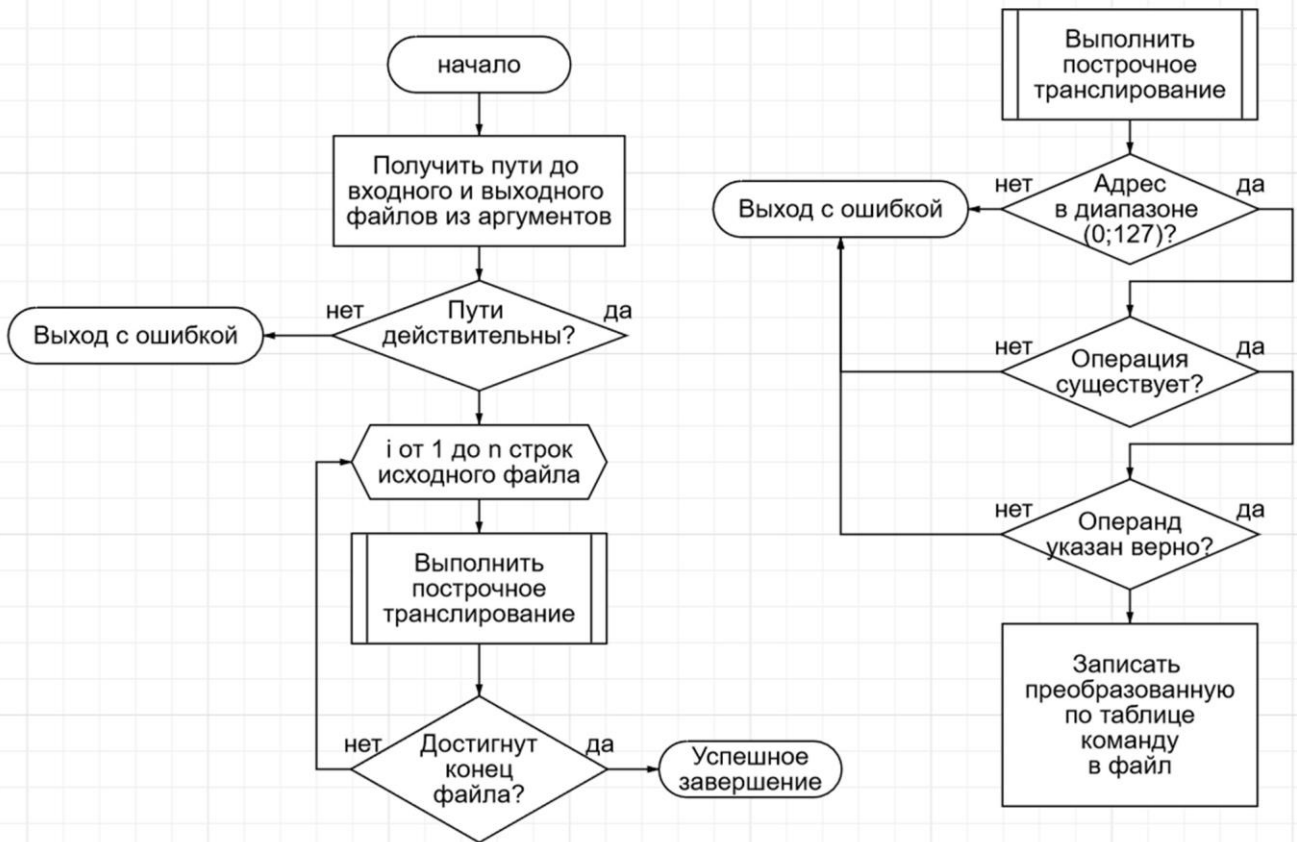


Рисунок 1. Блок-схема транслятора Simple Assembler

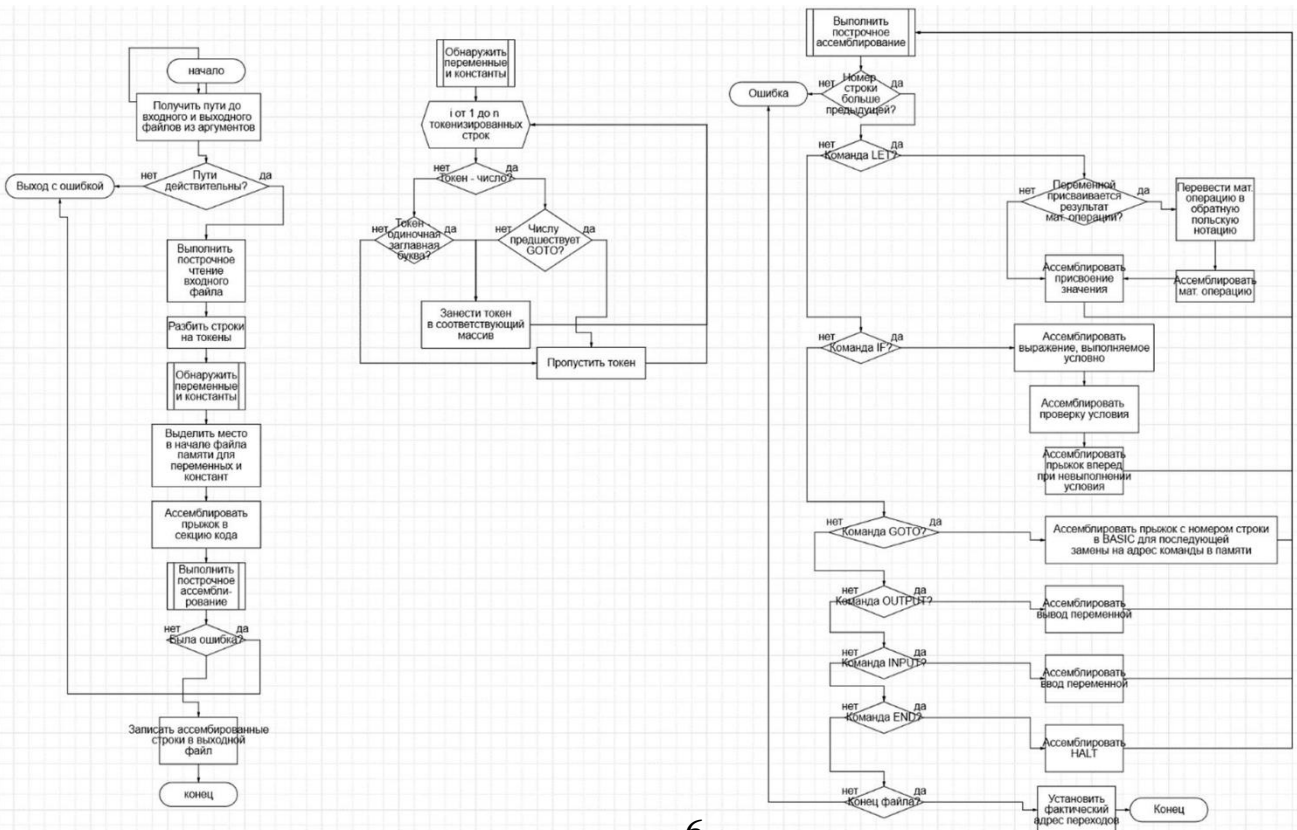


Рисунок 2. Блок-схема транслятора Simple Basic



Рисунок 3. Блок-схема модуля кэша L1

РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

Трансляция программы для вычисления факториала с Simple Basic в
программу на Simple Assembler:

```
simplebasic > ≡ file.sb
1  10 REM FACTORIAL
2  20 INPUT N
3  30 LET F = 1
4  40 LET C = 1
5  50 IF C = N GOTO 90
6  60 LET C = C + 1
7  70 LET F = F * C
8  80 GOTO 50
9  90 OUTPUT F
10 100 END
```

Рисунок 4. Программа для вычисления факториала на Simple Basic

```
simplebasic > ≡ file.sa
1  0 JUMP 5
2  1 = +0000 ; n
3  2 = +0001 ; f
4  3 = +0001 ; c
5  4 = +0001 ; 1
6  5 READ 1
7  6 LOAD 1 ; input N=3
8  7 SUB 3 ; subtract c=1 from n
9  8 JNEG 11
10 9 JNS 11
11 10 JUMP 18
12 11 LOAD 3 ; load c
13 12 ADD 4
14 13 STORE 3
15 14 LOAD 2
16 15 MUL 3
17 16 STORE 2
18 17 JUMP 6
19 18 WRITE 2
```

Рисунок 5.
Результат
трансляции
в ассемблер

Запустим mySimpleComputer с файлом file.o:

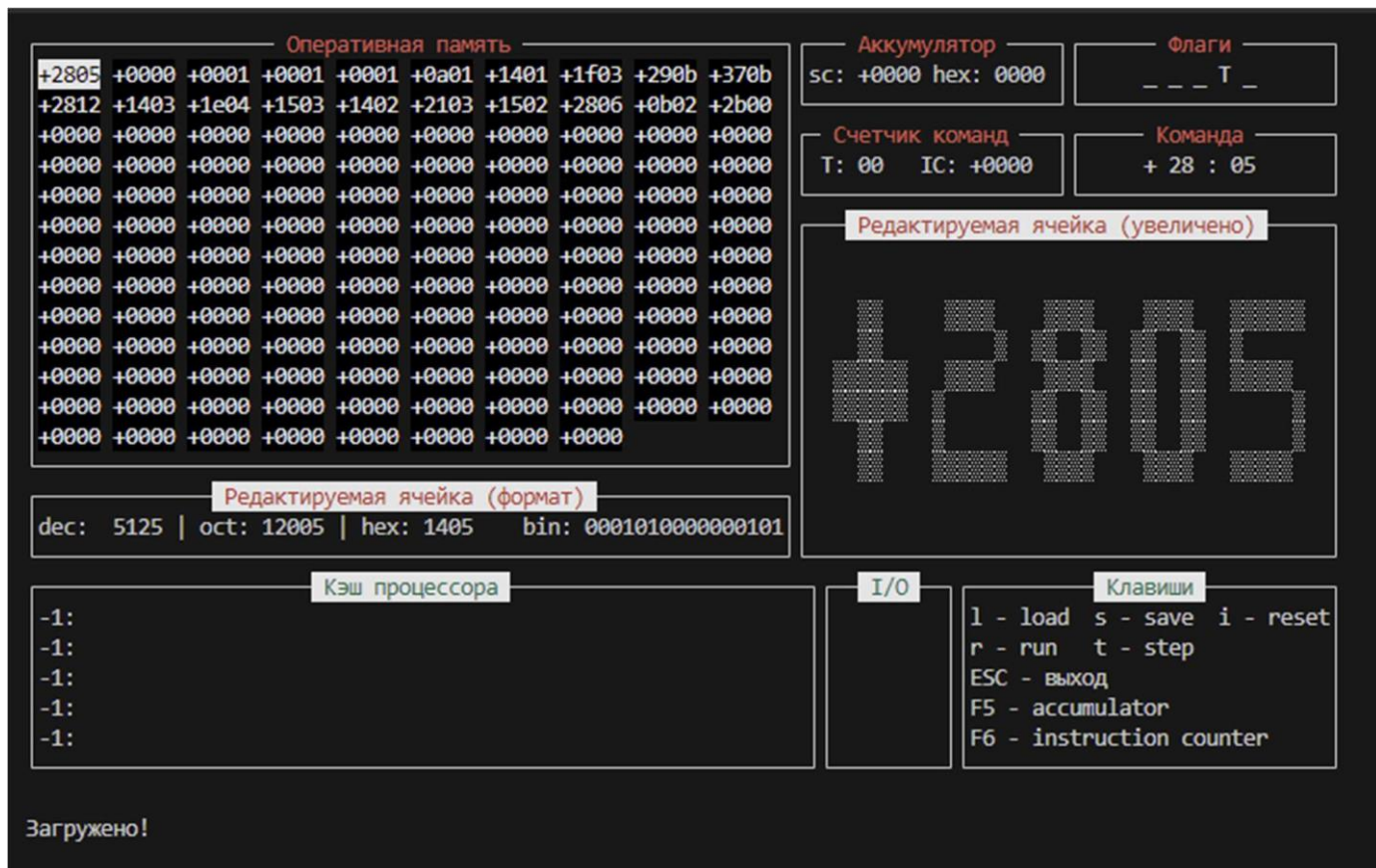


Рисунок 6. Результат трансляции из ассемблера в машинный код в окне консоли

Запустим программу и введём 4:

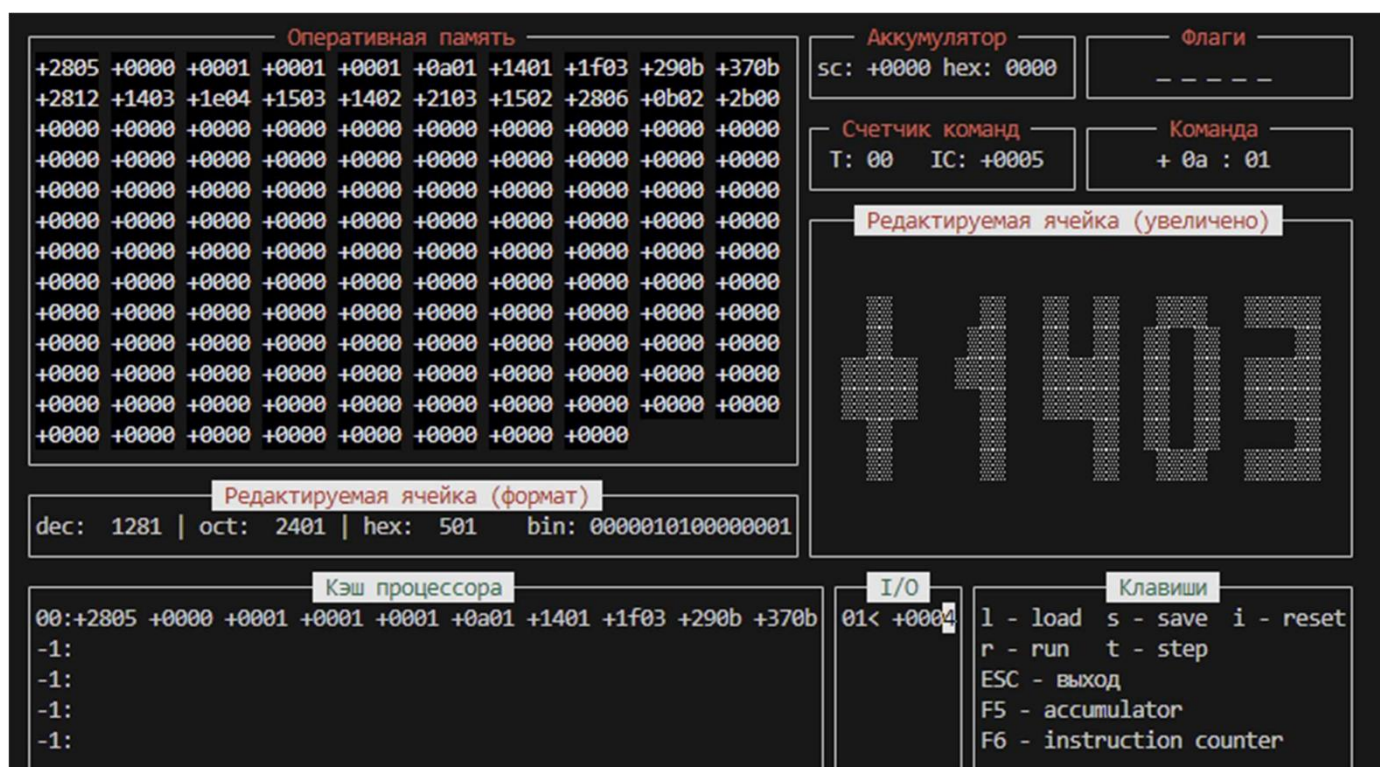


Рисунок 7. Ввод значения, для которого вычисляется факториал.

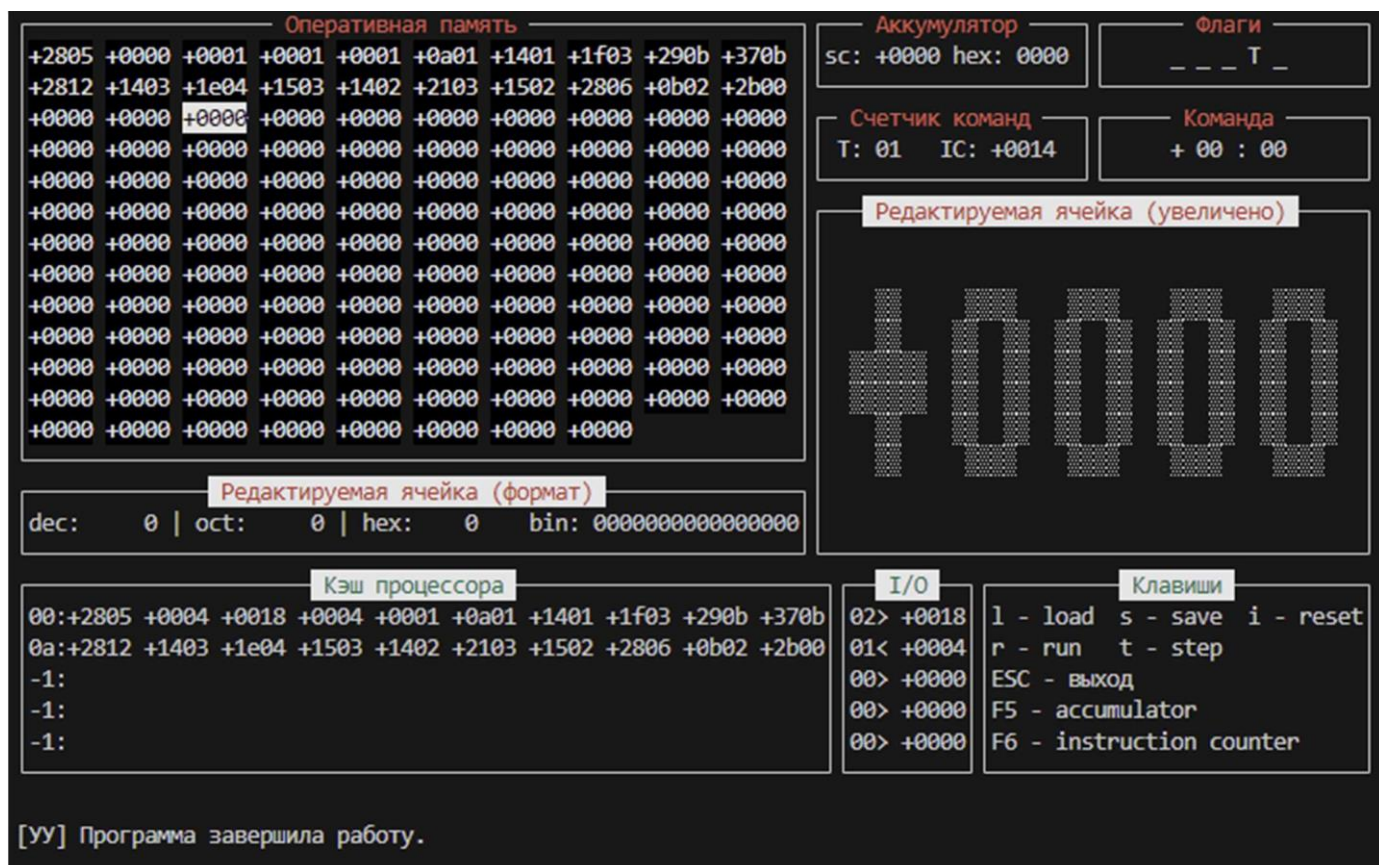


Рисунок 8. Результат работы программы

В поле ввода-вывода виден результат работы программы – факториал числа 4. ($0x18 = 4! = 24$)

ЗАКЛЮЧЕНИЕ

В рамках данной курсовой работы была доработана модель Simple Computer – реализован алгоритм «L1-кэш команд и данных» и модифицирована работа контроллера оперативной памяти и обработчика прерываний таким образом, чтобы учитывался простой процессора при прямом доступе к оперативной памяти. Разработаны трансляторы с Simple Basic и с Simple Assembler.

ИСПОЛЬЗОВАННАЯ ЛИТЕРАТУРА

1. Мамойленко С.Н., Молдованова О.В. ЭВМ и периферийные устройства:
Учебное пособие. – Новосибирск: СибГУТИ, 2012. – 106 с.