

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Ильшат Ринатович Мухаметзянов

Должность: директор

Дата подписания: 14.07.2023 09:36:08

Уникальный идентификатор:

aba80b84033c9ef196388e9ea0434f90a83a40954ba270e84bce64f02d1d8d0

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический

университет им. А.Н. Туполева-КАИ»

(КНИТУ-КАИ)

Чистопольский филиал «Восток»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ЛАБОРАТОРНЫМ РАБОТАМ

по дисциплине

МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ

Индекс по учебному плану: **Б1.В.07**

Направление подготовки: **09.03.01 Информатика и вычислительная техника**

Квалификация: **Бакалавр**

Профиль подготовки: **Автоматизированные системы обработки информации и управления**

Типы задач профессиональной деятельности: **проектный, производственно-технологический**

Рекомендовано УМК ЧФ КНИТУ-КАИ

Чистополь

2023 г.

Введение

Данные методические указания предназначены для проведения лабораторных работ по дисциплине «Микропроцессорные системы»

Цикл лабораторных работ включает задания различного уровня.

Лабораторные работы №1, №2, №3, №4 ориентированы на изучение структуры команд микропроцессора Intel 80386.

Лабораторные работы №5, №6 ориентированы на изучение локального интерфейса микропроцессора Intel 80386.

Лабораторные работы №7, № 8 ориентированы на изучение локального интерфейса микропроцессора Intel 80486.

№ п/п	№ раздела	Наименование лабораторных работ	Трудоемкость (час.)
1	1	Команды пересылки данных микропроцессора Intel: память → регистр; регистр → память	4
2	1	Команды пересылки данных микропроцессора Intel: данные в память; данные в регистр	4
3	1	Команды пересылки данных микропроцессора Intel из аккумулятора в память и из памяти в аккумулятор	4
4	1	Команды пересылки данных микропроцессора Intel из памяти в сегментный регистр и из сегментного регистра в аккумулятор	4
5	1	Локальный интерфейс Intel 80386: простой цикл простой цикл с ожиданием	4
6	1	Локальный интерфейс Intel 80386: конвейерный цикл	4
7	1	Локальный интерфейс Intel 80486: простой цикл	4
8	1	Локальный интерфейс Intel 80486: последовательность циклов пакетный цикл	4

1 Лабораторная работа №1 «Команды пересылки данных микропроцессора Intel: память → регистр; регистр → память»

1.1 Цель работы

Целью настоящей лабораторной работы является изучение способов пересылки данных микропроцессора Intel 80386 из памяти в регистр и из регистра в память.

1.2 Общие положения

В лабораторной работе №1 задаются команды пересылки данных из памяти в регистр и из регистра в память в виде формулы на языке Ассемблера.

В соответствии с заданной формулой команды необходимо заполнить таблицу формата команды двоичными кодами полей.

1.3 Задание на лабораторную работу №1

1. Ознакомиться с форматом команды микропроцессора Intel 80386 (Приложение 1).
2. Заполнить таблицу формата команды двоичными кодами полей.

1.4 Ход работы

В процессе работы выполнить следующие действия:

- 1) включить компьютер;
- 2) загрузить программную модель выполнения команд микропроцессора Intel 80386];
- 3) выбрать команды пересылки данных из памяти в регистр и из регистра в память;

- 4) заполнить таблицу формата команд двоичными кодами полей;
- 5) представить результаты работы преподавателю в устной форме.

1.5 Выполнение команды пересылки данных из памяти в регистр

Результаты выполнения команды пересылки данных из памяти в регистр приведены на рис. 1.

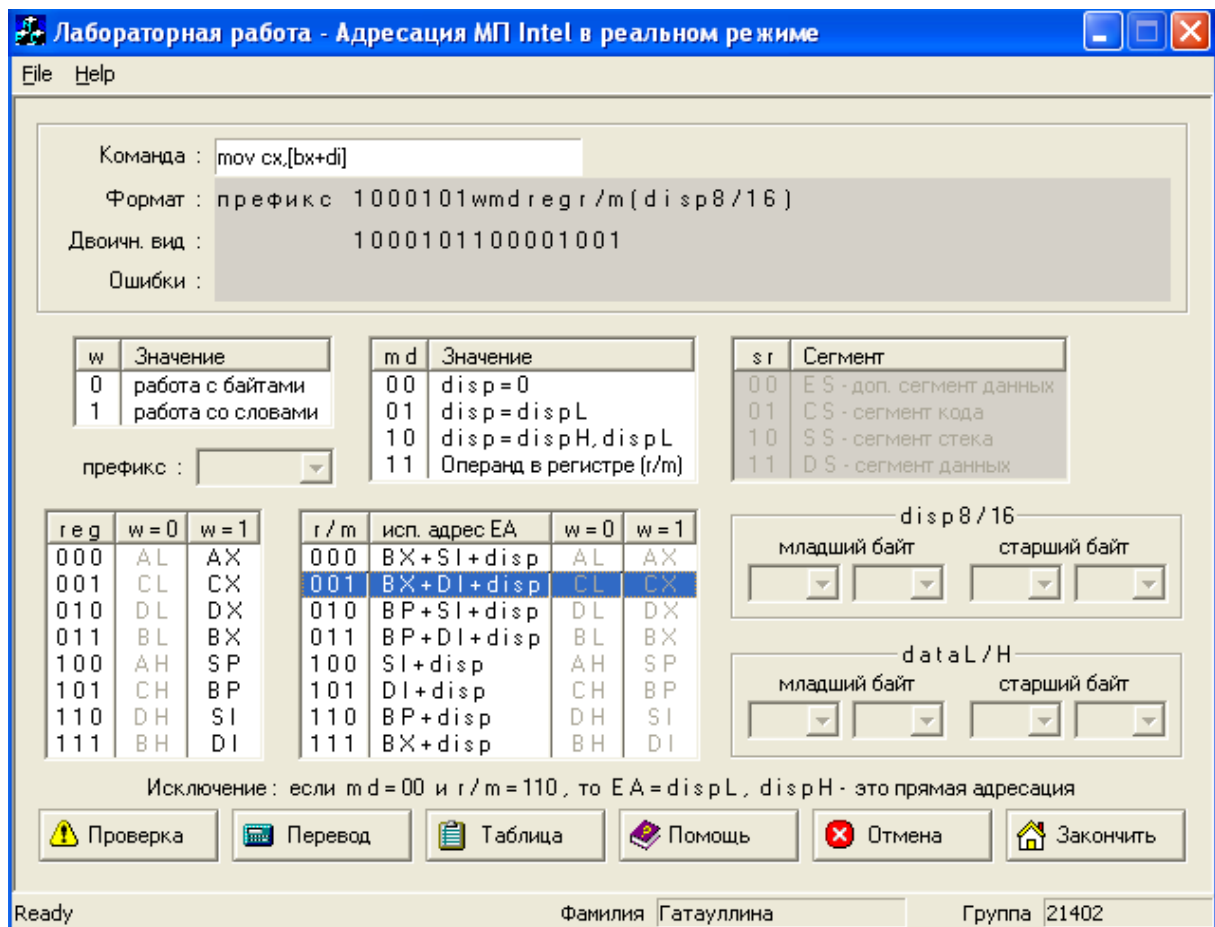


Рисунок 1 – Пример пересылки данных из памяти в регистр

Данная команда пересылки **mov cx,[bx+di]**, пересылает данные из второго операнда [bx+di], в регистр cx.

Поскольку указан регистр cx, то это означает, что работа будет со словами, соответственно бит w выбираем равным единице: **w=1**.

Так как бит **w=1** и мы работаем с регистром **cx**, тогда **reg** устанавливаем равным **001**, согласно Таблице 2: **reg=001**.

Поскольку второй операнд **[bx+di]**, указывает базово-индексную адресацию, то согласно Таблице 2 устанавливаем **md= 00, r/m=001**.

1.6 Выполнение команды пересылки данных из регистра в память

Результаты выполнения команды пересылки данных из регистра в память приведены на рис. 2.

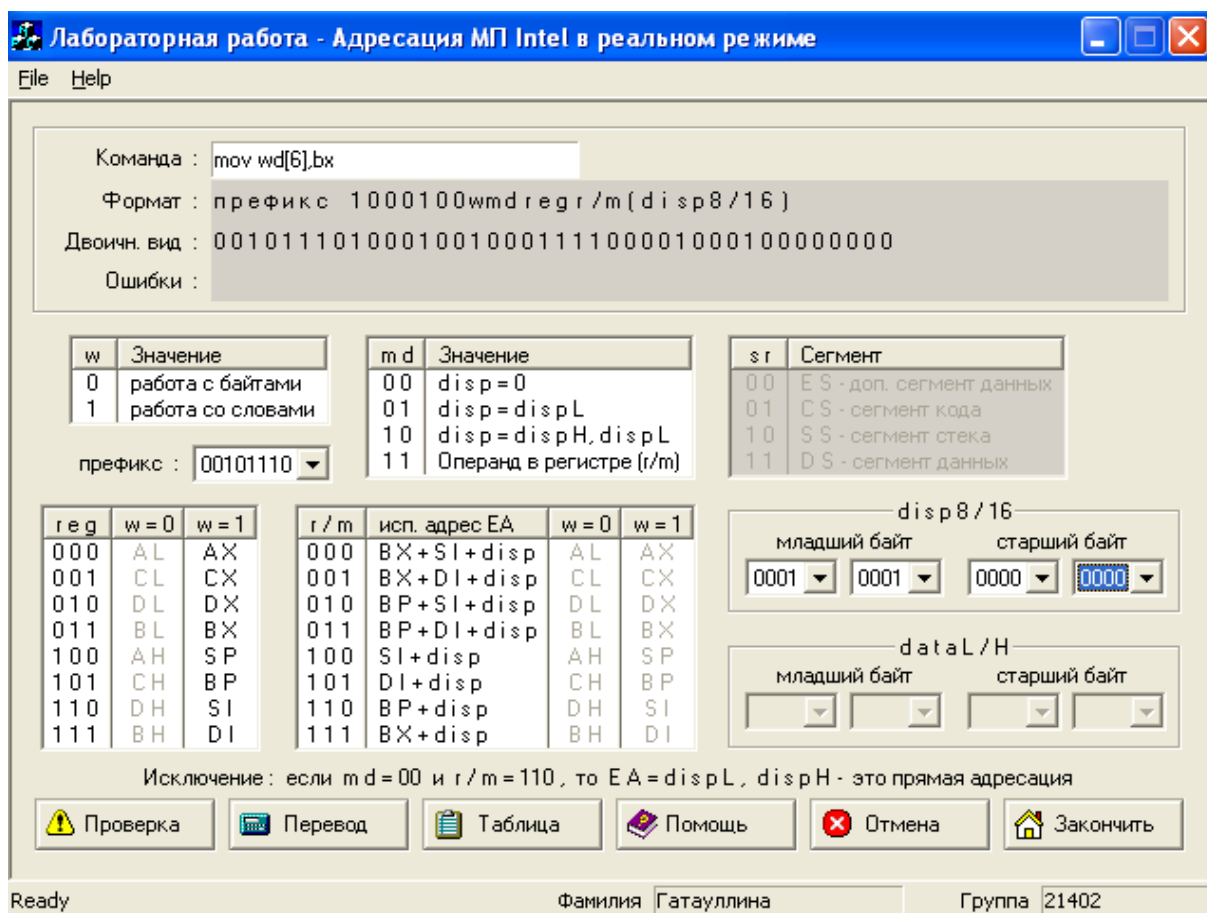


Рисунок 2 – Пример пересылки данных из регистра в память

Данная команда пересылки **mov wd[6], bx** пересылает данные из регистра **bx** в ячейку памяти, помеченную меткой **wd**.

В команде указано смещение **+6**. Поскольку указан регистр **bx**, то это означает, что работа будет со словами, соответственно бит **w** выбираем равным единице: **w=1**.

В таблице метка **wd** является данным и имеет внутреннее смещение **+11**, тогда общее смещение устанавливаем равное **6+11=+17 (0001 0001)**.

0001	0001
------	------

Поскольку метка со смещением означает прямую адресацию, то устанавливаем: **md=00, r/m=110**.

1.7 Представление результатов работы

Отчет по лабораторной работе должен содержать:

- 1) оглавление;
- 2) описание цели работы;
- 3) описание хода работы в соответствии с данными методическими указаниями;
- 4) выводы по результатам выполнения работы;
- 5) список литературы;

Выводы

В результате выполнения данной лабораторной работы приобретены навыки представления в машинном коде команд пересылки данных из памяти в регистр и из регистра в память микропроцессора Intel 80386.

Список литературы

1. Гуров В.В. Микропроцессорные системы: Учеб.пособие. –М.: ИНФРА-М, 2017. – 336 с. + Доп.материалы/ Электронный ресурс; Режим доступа <http://www.znaniium.com>]. –(Высшее образование: Бакалавриат).
2. Русанов, В.В. Микропроцессорные устройства и системы [Электронный ресурс] : учеб. пособие / В.В. Русанов, М.Ю. Шевелев. — Электрон. дан. — Москва : ТУСУР, 2012. — 184 с. — Режим доступа: <https://e.lanbook.com/book/10931>. — Загл. с экрана.
3. Борисов, В.В. Нечеткие модели и сети [Электронный ресурс] : учеб. пособие / В.В. Борисов, В.В. Круглов, А.С. Федулов. — Электрон. дан. — Москва : Горячая линия-Телеком, 2012. — 284 с. — Режим доступа: <https://e.lanbook.com/book/5126>. — Загл. с экрана.

2 Лабораторная работа №2 «Команды пересылки данных микропроцессора Intel в память и данных в регистр»

2.1 Цель работы

Целью настоящей лабораторной работы является изучение способов пересылки данных микропроцессора Intel 80386 в память и пересылки данных в регистр.

2.2 Общие положения

В лабораторной работе №1 задаются команды пересылки данных в память и пересылки данных в регистр в виде формулы на языке Ассемблера.

В соответствии с заданной формулой команды необходимо заполнить таблицу формата команды двоичными кодами полей.

2.3 Задание на лабораторную работу №2

1. Ознакомиться с форматом команды микропроцессора Intel 80386 (Приложение 1).
2. Заполнить таблицу формата команды двоичными кодами полей.

2.4 Ход работы

В процессе работы выполнить следующие действия:

- 3) включить компьютер;
- 4) загрузить программную модель выполнения команд микропроцессора Intel 80386];
- 3) выбрать команды пересылки данных в память и пересылки данных в регистр;

- 4) заполнить таблицу формата команд двоичными кодами полей;
- 5) представить результаты работы преподавателю в устной форме.

2.5 Выполнение команды пересылки данных в память

Результаты выполнения команды пересылки данных в память приведены на рис. 1.

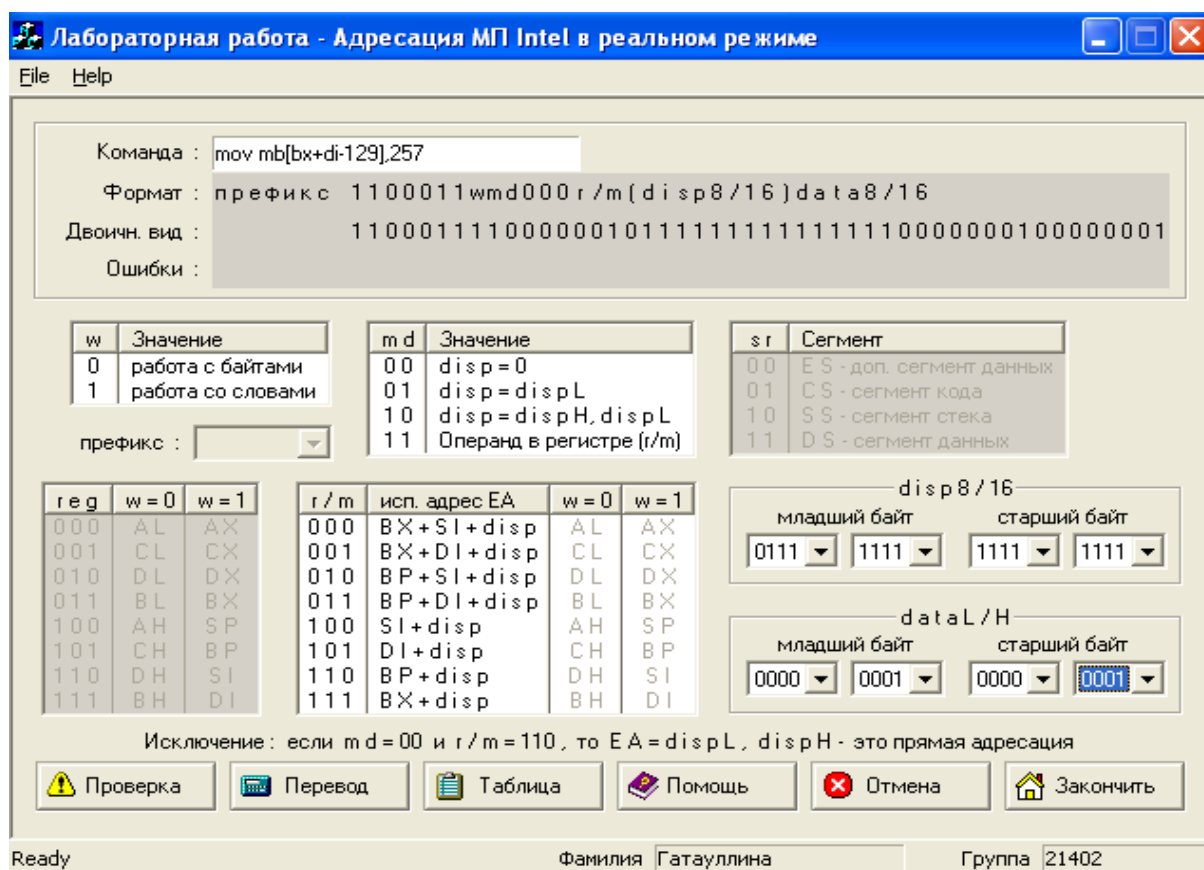


Рисунок 1 – Пример пересылки данных в память

Данная команда **mov mb[bx+di-129], 257** пересылает число 257 в ячейку памяти, помеченную меткой **mb**.

Поскольку указано символическое имя **mb**, то это означает, что работа будет со словами, соответственно бит **w** выбираем равным единице: **w=1**.

В таблице метка **mb** указывает на ячейку с данными и не имеет внутреннего смещения, тогда общее смещение устанавливаем равное **-129 (0111 1111 1111 1111)**.

0111	1111	1111	1111
------	------	------	------

Переводим число, которое нам надо переслать в ячейку памяти **mb**, то есть 257, в двоичный код и записываем его в поле DataL/H (**0000 0001 0000 0001**).

0000	0001	0000	0001
------	------	------	------

Поскольку второй операнд [**bx+di-129**], указывает базово-индексную адресацию, то согласно Таблице 2 устанавливаем **md= 10, r/m=001**.

2.6 Выполнение команды пересылки данных в регистр

Результаты выполнения команды пересылки данных в регистр приведены на рис.2.

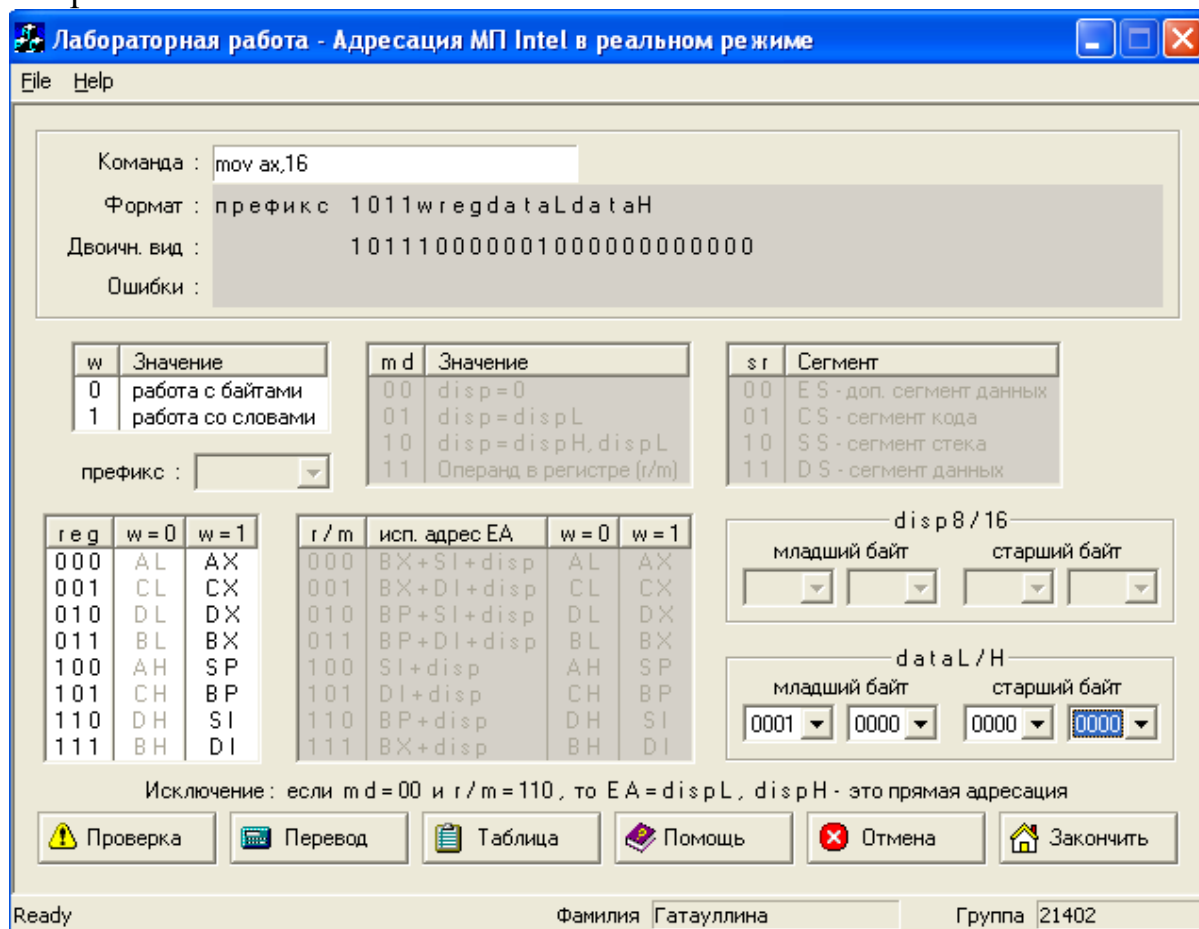


Рисунок 2 – Пример пересылки данных в регистр

Данная команда пересылки `mov ax, 16` пересылает данные в регистр.

Поскольку указан регистр `ax`, то это означает, что работа будет со словами, соответственно бит `w` выбираем равным единице: `w=1`.

Переводим число, которое нам надо переслать в регистр `ax`, то есть 16 в двоичный код и вставляем его в поле `DateL/H` (`0001 0000 0000 0000`).

0001	0000	0000	0000
------	------	------	------

Так как бит `w=1` и мы работаем с регистром `ax`, тогда `reg` устанавливаем равным 000, согласно Таблице 2: `reg=000`.

В данном примере используется непосредственный способ адресации, так как операнд является константой.

2.7 Представление результатов работы

Отчет по лабораторной работе должен содержать:

- 1) оглавление;
- 2) описание цели работы;
- 3) описание хода работы в соответствии с данными методическими указаниями;
- 4) выводы по результатам выполнения работы;
- 5) список литературы;

Выводы

В результате выполнения данной лабораторной работы приобретены навыки представления в машинном коде команд пересылки данных в память и команд пересылки данных в регистр микропроцессора Intel 80386.

Список литературы

1. Гуров В.В. Микропроцессорные системы: Учеб.пособие. –М.: ИНФРА-М, 2017. – 336 с. + Доп.материалы/ Электронный ресурс; Режим доступа <http://www.znanium.com>]. –(Высшее образование: Бакалавриат).
2. Русанов, В.В. Микропроцессорные устройства и системы [Электронный ресурс] : учеб. пособие / В.В. Русанов, М.Ю. Шевелев. — Электрон. дан. — Москва : ТУСУР, 2012. — 184 с. — Режим доступа: <https://e.lanbook.com/book/10931>. — Загл. с экрана.
3. Борисов, В.В. Нечеткие модели и сети [Электронный ресурс] : учеб. пособие / В.В. Борисов, В.В. Круглов, А.С. Федулов. — Электрон. дан. — Москва : Горячая линия-Телеком, 2012. — 284 с. — Режим доступа: <https://e.lanbook.com/book/5126>. — Загл. с экрана.

3 Лабораторная работа №3 «Команды пересылки данных микропроцессора Intel из аккумулятора в память и из памяти в аккумулятор»

3.1 Цель работы

Целью настоящей лабораторной работы является изучение способов пересылки данных микропроцессора Intel 80386 из аккумулятора в память и из памяти в аккумулятор.

3.2 Общие положения

В лабораторной работе №3 задаются команды пересылки данных из аккумулятора в память и из памяти в аккумулятор в виде формулы на языке Ассемблера.

В соответствии с заданной формулой команды необходимо заполнить таблицу формата команды двоичными кодами полей.

3.3 Задание на лабораторную работу №3

1. Ознакомиться с форматом команды микропроцессора Intel 80386 (Приложение 1).
2. Заполнить таблицу формата команды двоичными кодами полей.

3.4 Ход работы

В процессе работы выполнить следующие действия:

- 5) включить компьютер;
- 6) загрузить программную модель выполнения команд микропроцессора Intel 80386];

3) выбрать команды пересылки данных из аккумулятора в память и из памяти в аккумулятор;

4) заполнить таблицу формата команд двоичными кодами полей;

5) представить результаты работы преподавателю в устной форме.

3.5 Выполнение команды пересылки данных из памяти в аккумулятор

Результаты выполнения команды пересылки данных из памяти в аккумулятор приведены на рис.1.

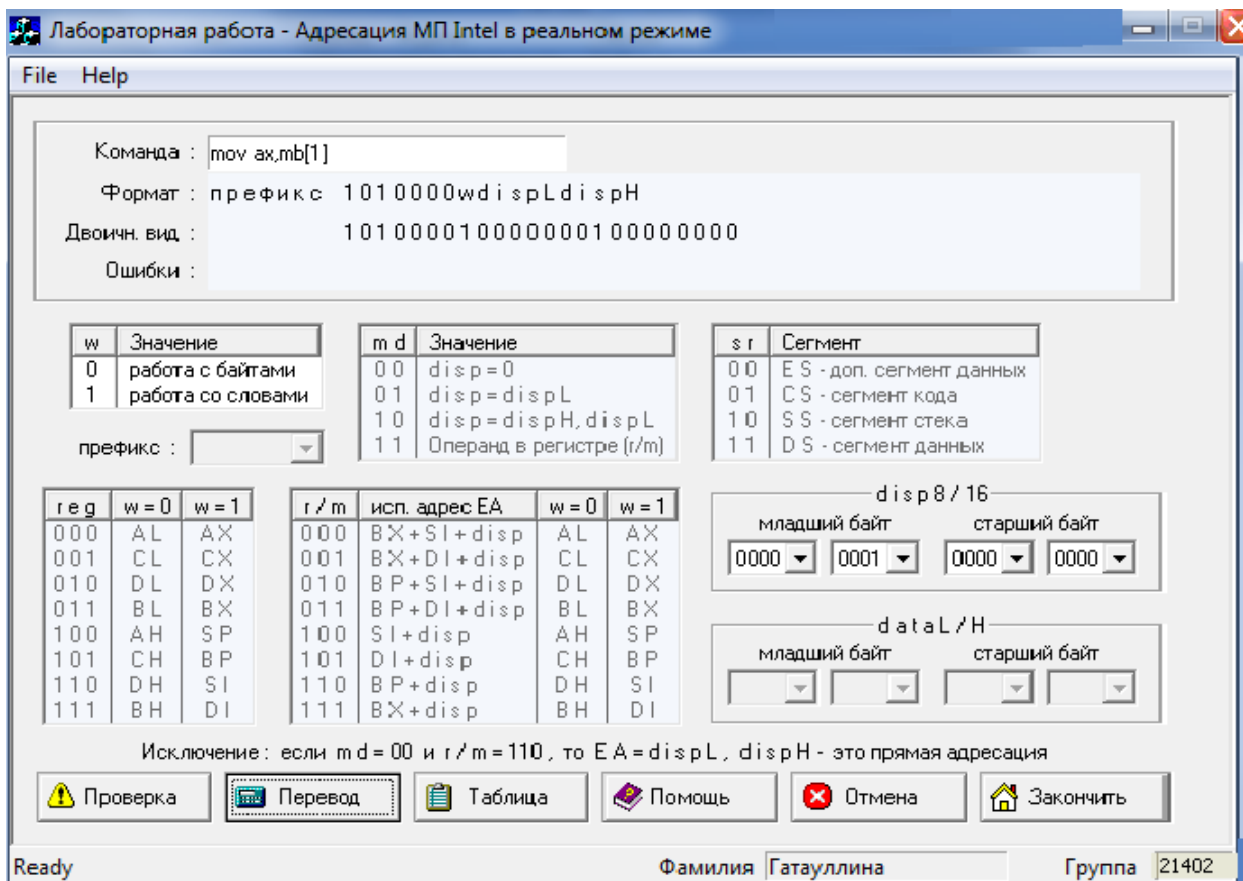


Рисунок 1 – Пример пересылки данных из памяти в аккумулятор

Данная команда **mov ax,mb[1]** пересылает данные из ячейки памяти, помеченной меткой **mb**, в аккумулятор **ax**.

Поскольку указано символическое имя **mb**, то это означает, что работа будет со словами, соответственно бит **w** выбираем равным единице: **w=1**.

В данном примере используется прямой способ адресации.

В таблице метка **mb** указывает на ячейку с данными и не имеет внутреннего смещения, тогда общее смещение устанавливаем равное

1 (0000 0000 0000 0001).

0000	0000	0000	0001
------	------	------	------

3.6 Выполнение команды пересылки данных из аккумулятора в память

Результаты выполнения команды пересылки данных из аккумулятора в память приведены на рис.2.

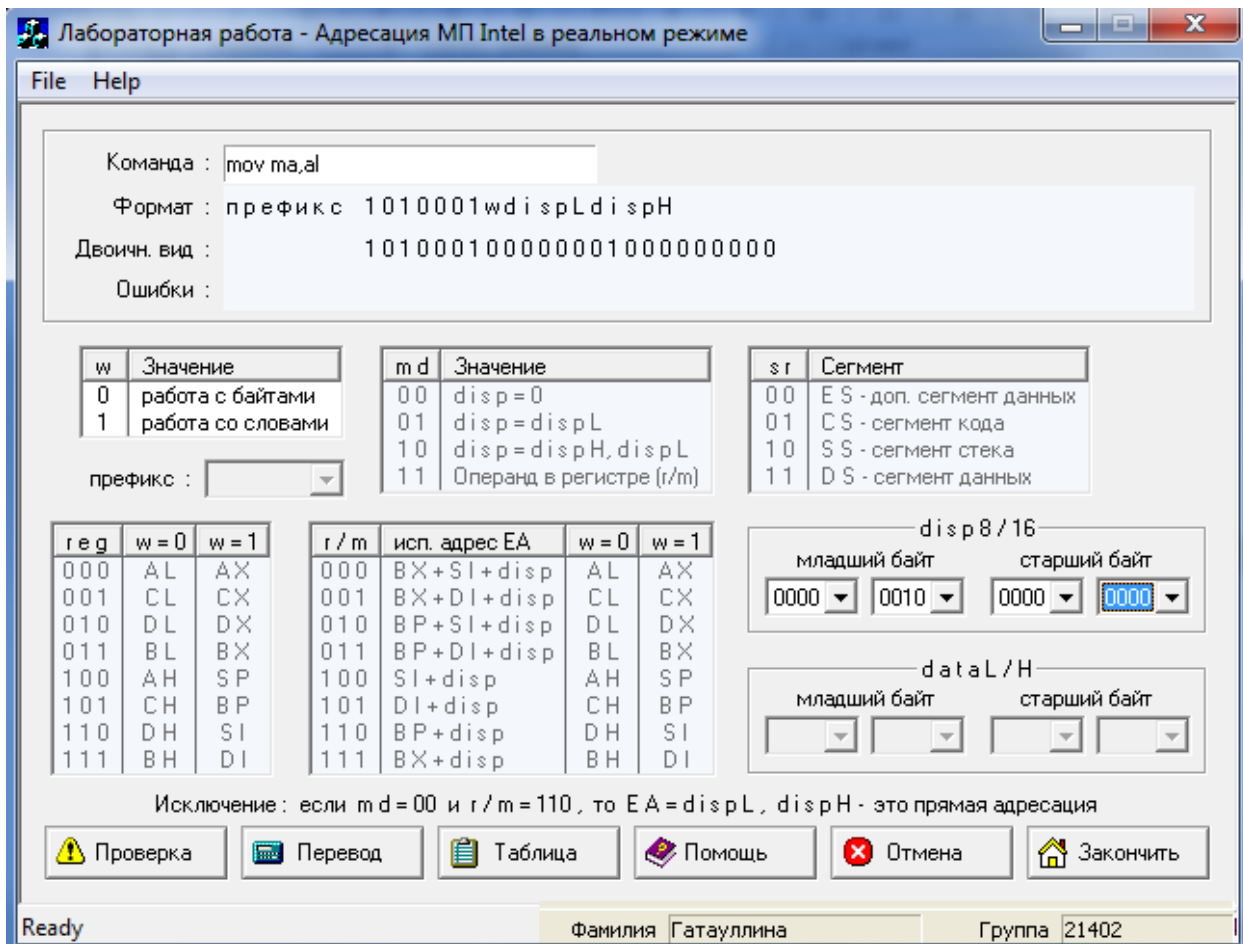


Рисунок 1 – Пример пересылки данных из аккумулятора в память

Данная команда пересылки **mov ma, al** пересылает данные из ячейки памяти, помеченную меткой **ma**, в аккумулятор **al**.

Поскольку указан регистр **al**, то это означает, что работа будет с байтами, соответственно бит **w** выбираем равным нулю: **w=0**.

В данном примере используется прямой способ адресации.

В таблице метка **ma** указывает на ячейку с данными и имеет внутреннее смещение равное **2 (0000 0000 0000 0010)**.

0000	0000	0000	0010
------	------	------	------

3.7 Представление результатов работы

Отчет по лабораторной работе должен содержать:

- 1) оглавление;
- 2) описание цели работы;
- 3) описание хода работы в соответствии с данными методическими указаниями;
- 4) выводы по результатам выполнения работы;
- 5) список литературы;

Выводы

В результате выполнения данной лабораторной работы приобретены навыки представления в машинном коде команд пересылки данных из памяти в аккумулятор и команд пересылки данных из аккумулятора в память микропроцессора Intel 80386.

Список литературы

1. Гуров В.В. Микропроцессорные системы: Учеб.пособие. –М.: ИНФРА-М, 2017. – 336 с. + Доп.материалы/ Электронный ресурс; Режим доступа <http://www.znanium.com>]. –(Высшее образование: Бакалавриат).
2. Русанов, В.В. Микропроцессорные устройства и системы [Электронный ресурс] : учеб. пособие / В.В. Русанов, М.Ю. Шевелев. — Электрон. дан. — Москва : ТУСУР, 2012. — 184 с. — Режим доступа: <https://e.lanbook.com/book/10931>. — Загл. с экрана.
3. Борисов, В.В. Нечеткие модели и сети [Электронный ресурс] : учеб. пособие / В.В. Борисов, В.В. Круглов, А.С. Федулов. — Электрон. дан. — Москва : Горячая линия-Телеком, 2012. — 284 с. — Режим доступа: <https://e.lanbook.com/book/5126>. — Загл. с экрана.

4 Лабораторная работа №4 «Команды пересылки данных микропроцессора Intel из памяти в сегментный регистр и из сегментного регистра в аккумулятор»

4.1 Цель работы

Целью настоящей лабораторной работы является изучение способов пересылки данных микропроцессора Intel 80386 из памяти в сегментный регистр и из сегментного регистра в аккумулятор.

4.2 Общие положения

В лабораторной работе №4 задаются команды пересылки данных из памяти в сегментный регистр и из сегментного регистра в аккумулятор в виде формулы на языке Ассемблера.

В соответствии с заданной формулой команды необходимо заполнить таблицу формата команды двоичными кодами полей.

4.3 Задание на лабораторную работу №4

1. Ознакомиться с форматом команды микропроцессора Intel 80386 (Приложение 1).
2. Заполнить таблицу формата команды двоичными кодами полей.

4.4 Ход работы

В процессе работы выполнить следующие действия:

- 1) включить компьютер;
- 2) загрузить программную модель выполнения команд микропроцессора Intel 80386];

- 3) выбрать команды пересылки данных из памяти в сегментный регистр и из сегментного регистра в аккумулятор;
- 4) заполнить таблицу формата команд двоичными кодами полей;
- 5) представить результаты работы преподавателю в устной форме.

4.5 Команда пересылки данных из памяти в сегментный регистр

Результаты выполнения команды пересылки данных из памяти в сегментный регистр приведены на рис.1.

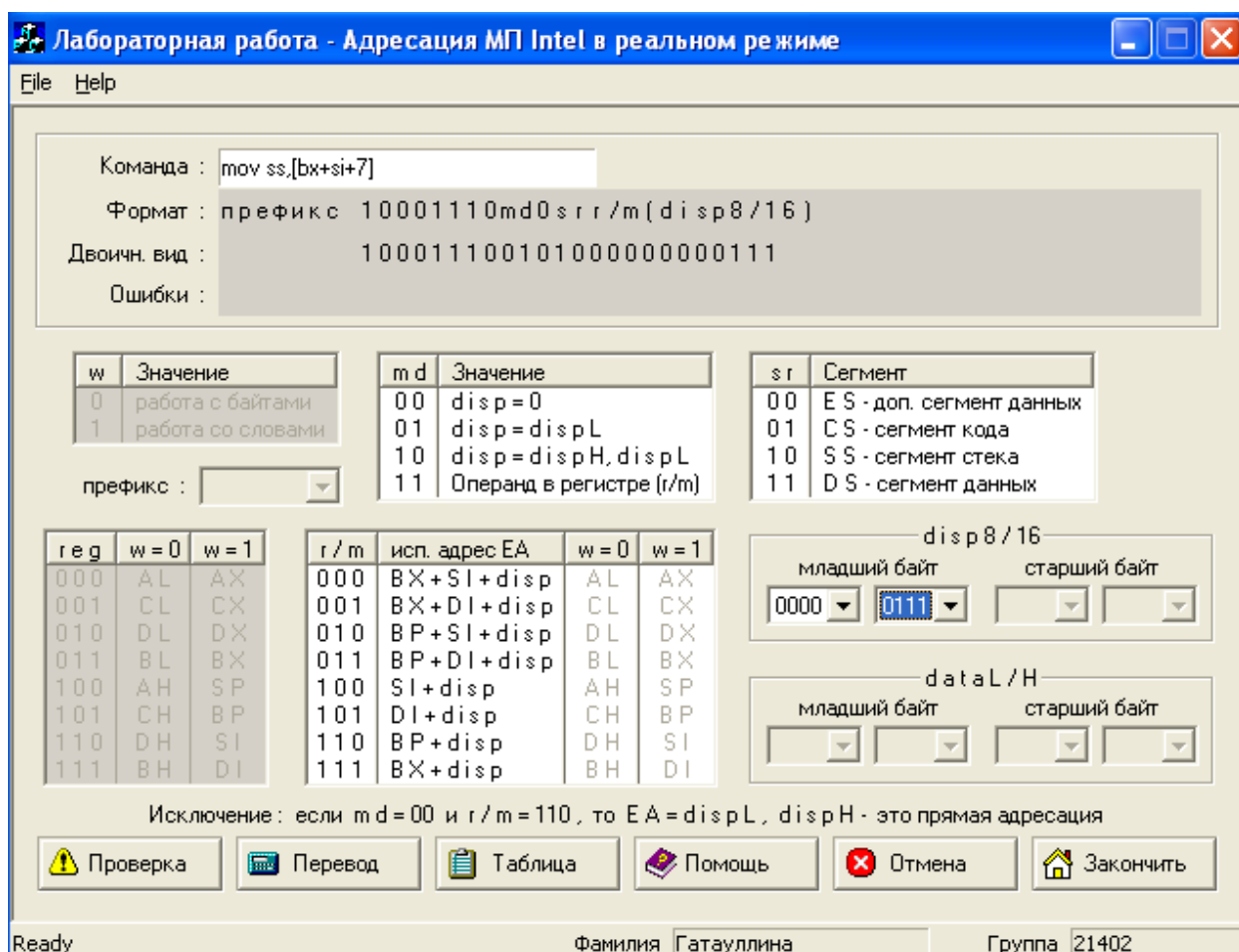


Рисунок 1 – Пример пересылки данных из памяти в сегментный регистр

Данная команда **mov ss, [bx+si+7]** пересылает данные из ячейки памяти с базово-индексной адресацией **[bx+si+7]**, то согласно Таблице 2 устанавливаем **md= 01 r/m=000. .**

Смещение устанавливаем равное 7 (0000 0111)

0000	0111
------	------

Так как используется сегментный регистр *ss* (сегмент стека), *sr* устанавливаем равным 10: **sr=10**.

4.6 Команда пересылки данных из сегментного регистра в аккумулятор

Результаты выполнения команды пересылки данных из сегментного регистра в аккумулятор приведены на рис.2.

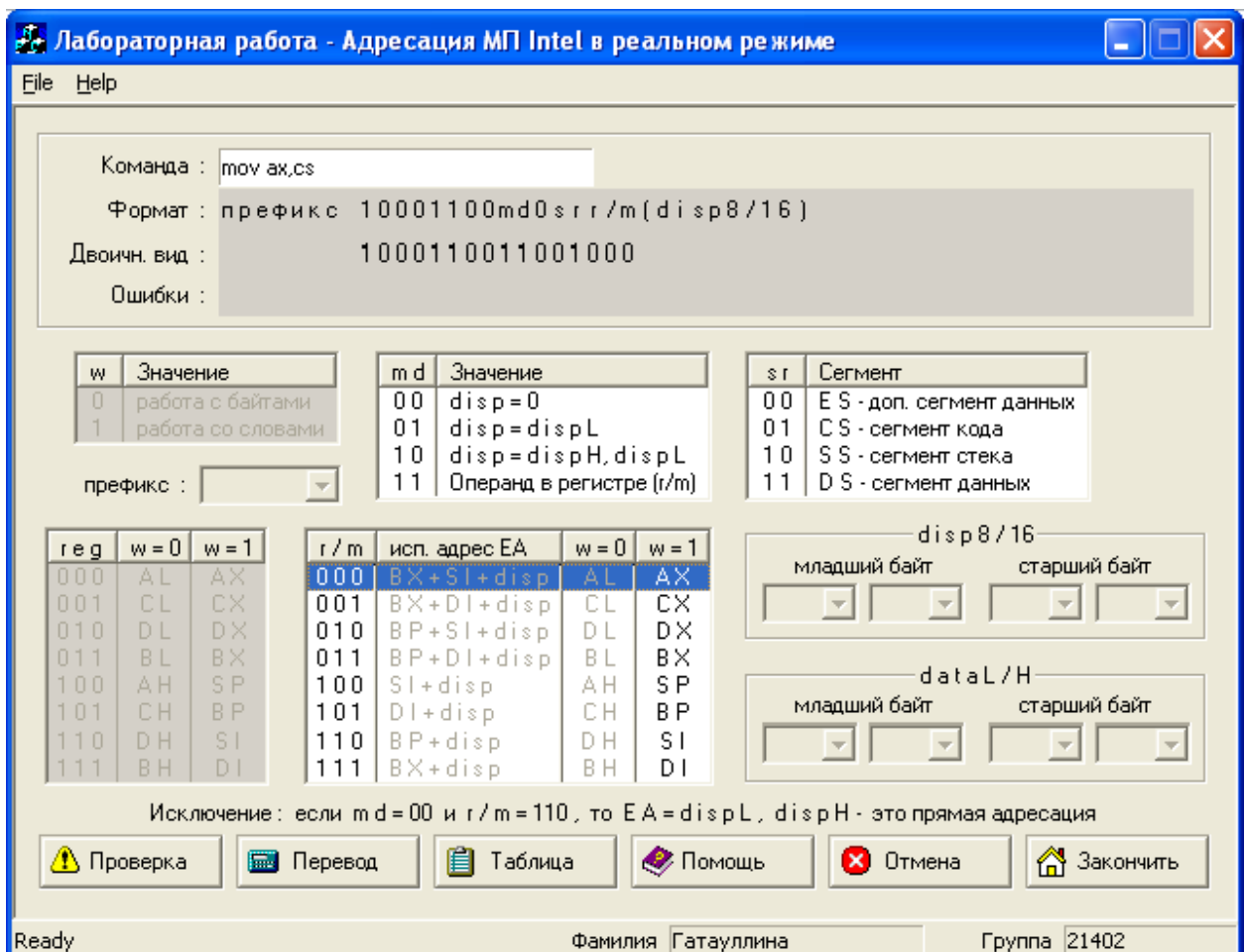


Рисунок 2 – Пример пересылки данных из сегментного регистра в аккумулятор

Данная команда пересылки **mov ax, cs** пересылает данные из регистра **ax** в сегментный регистр **cs**.

В данном примере используется регистровый прямой способ адресации.

Поскольку операнд находится в регистре, то **md** устанавливаем равным 11 и **r/m** соответственно 000: **md= 11 r/m=000**.

Так как используется сегментный регистр **cs** (сегмент кода), **sr** устанавливаем равным 01: **sr=01**.

4.7 Представление результатов работы

Отчет по лабораторной работе должен содержать:

- 1) оглавление;
- 2) описание цели работы;
- 3) описание хода работы в соответствии с данными методическими указаниями;
- 4) выводы по результатам выполнения работы;
- 5) список литературы;

Выводы

В результате выполнения данной лабораторной работы приобретены навыки представления в машинном коде команд пересылки данных из памяти в сегментный регистр и команд пересылки данных из сегментного регистра в аккумулятор микропроцессора Intel 80386

Список литературы

1. Гуров В.В. Микропроцессорные системы: Учеб.пособие. –М.: ИНФРА-М, 2017. – 336 с. + Доп.материалы/ Электронный ресурс; Режим доступа <http://www.znaniium.com>]. –(Высшее образование: Бакалавриат).
2. Русанов, В.В. Микропроцессорные устройства и системы [Электронный ресурс] : учеб. пособие / В.В. Русанов, М.Ю. Шевелев. — Электрон. дан. — Москва : ТУСУР, 2012. — 184 с. — Режим доступа: <https://e.lanbook.com/book/10931>. — Загл. с экрана.
3. Борисов, В.В. Нечеткие модели и сети [Электронный ресурс] : учеб. пособие / В.В. Борисов, В.В. Круглов, А.С. Федулов. — Электрон. дан. — Москва : Горячая линия-Телеком, 2012. — 284 с. — Режим доступа: <https://e.lanbook.com/book/5126>. — Загл. с экрана.

5 Лабораторная работа №5 «Локальный интерфейс Intel 80386: простой цикл и в режиме простого цикла с ожиданием»

5.1 Цель работы

Целью настоящей лабораторной работы является изучение функционирования локального интерфейса микропроцессора Intel 80386 в режиме простого цикла и в режиме простого цикла с ожиданием.

5.2 Общие положения

В лабораторной работе №5 моделируются сигналы локального интерфейса микропроцессора Intel 80386.

В соответствии с заданными в лабораторной работе условиями необходимо установить требуемые уровни управляющих сигналов в различных системных состояниях при выполнении простых циклов и в режиме простых циклов с ожиданием.

5.3 Задание на лабораторную работу №5

1. Ознакомиться с принципами функционирования локального интерфейса микропроцессора Intel 80386 в режиме простого цикла и в режиме простого цикла с ожиданием (Приложение 2).

2. Построить временные диаграммы функционирования локального интерфейса микропроцессора Intel 80386 в режиме простого цикла и в режиме простого цикла с ожиданием на основе графической модели.

5.4 Ход работы

В процессе работы выполнить следующие действия:

- 1) включить компьютер;
- 2) загрузить программную модель функционирования локального интерфейса микропроцессора Intel 80386;
- 3) построить временные диаграммы функционирования локального интерфейса микропроцессора Intel 80386 в режиме простого цикла и в режиме простого цикла с ожиданием на основе графической модели;
- 4) представить результаты работы преподавателю в устной форме.

5.5 Работа локального интерфейса микропроцессора Intel 80386 в режиме простого цикла

Результаты функционирования локального интерфейса микропроцессора Intel 80386 в режиме простого цикла приведены на рис.1.

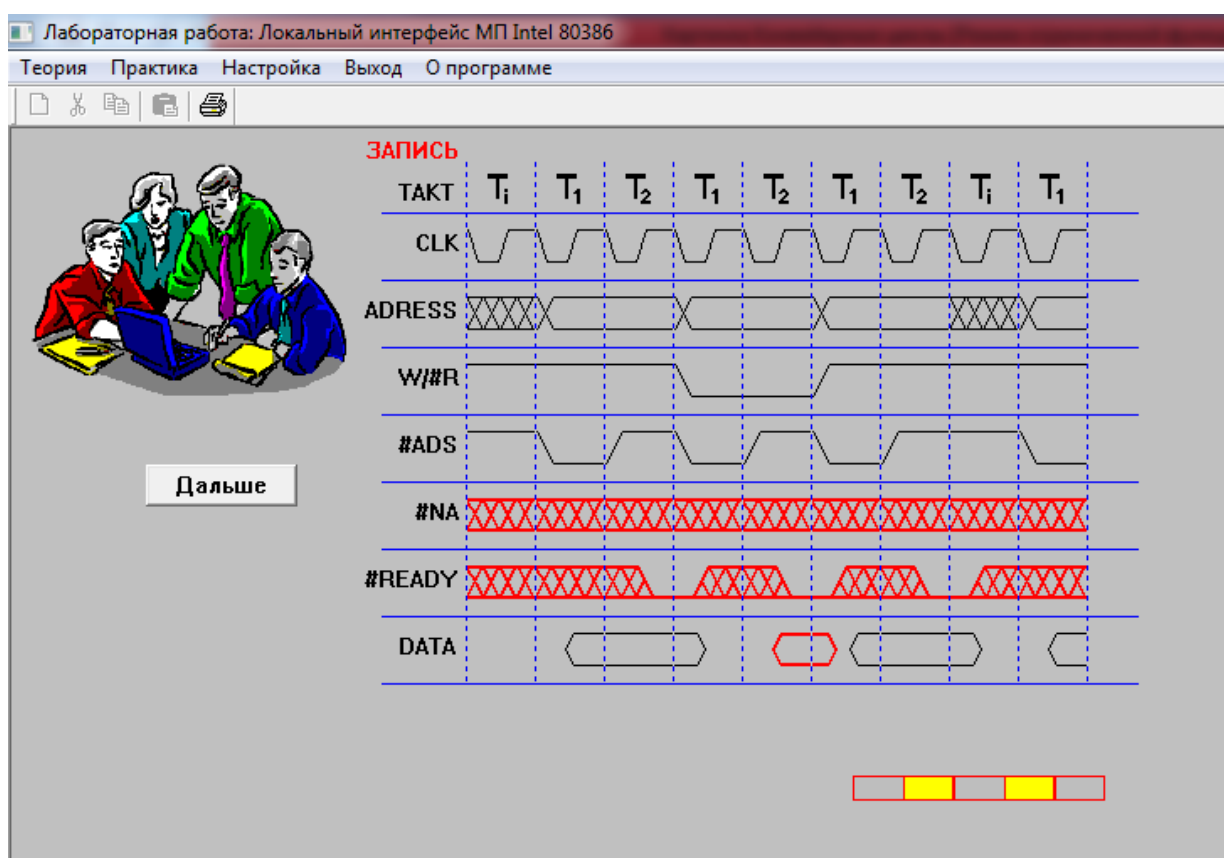


Рисунок 1 – Простой цикл

5.6 Работа локального интерфейса микропроцессора Intel 80386

в режиме простого цикла с ожиданием

Результаты функционирования локального интерфейса микропроцессора Intel 80386 в режиме простого цикла с ожиданием приведены на рис.2.

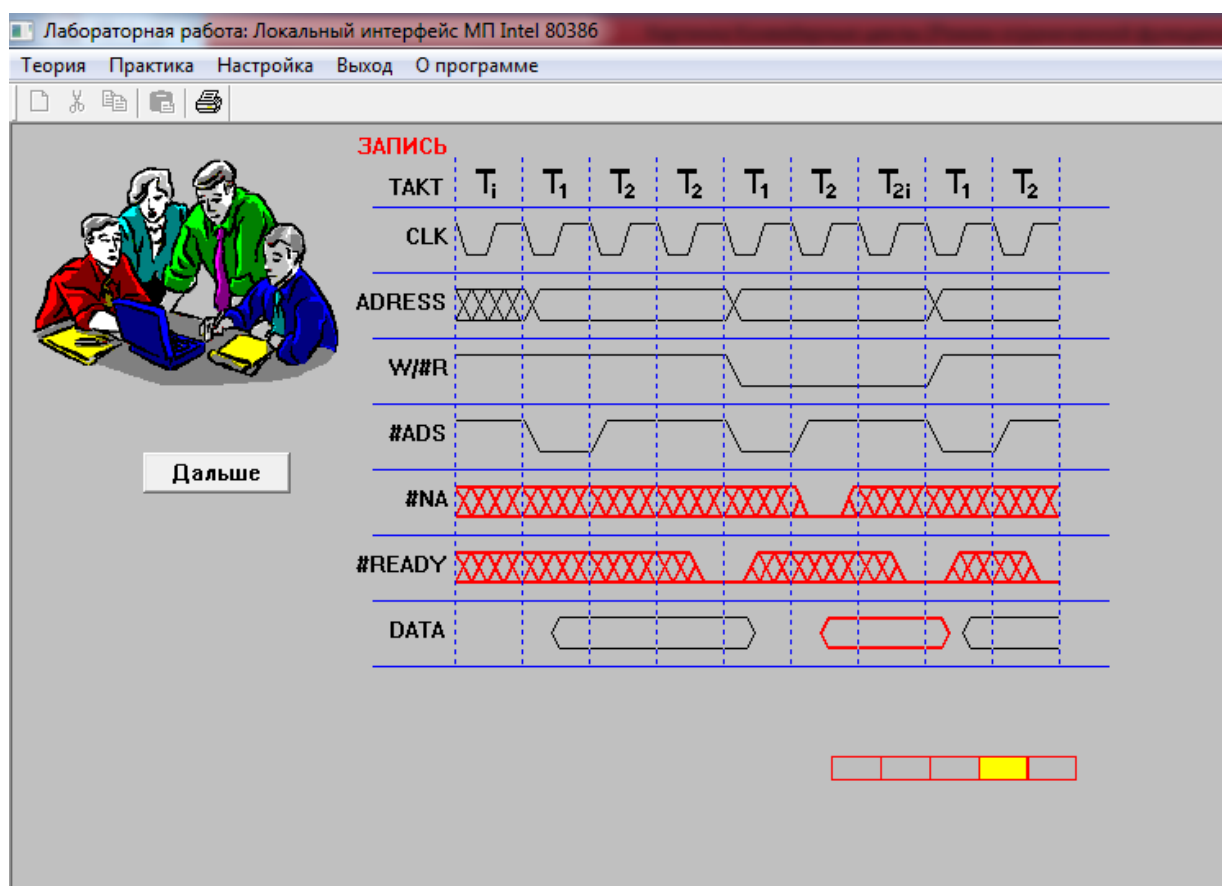


Рисунок 2 – Простой цикл с ожиданием

5.7 Представление результатов работы

Отчет по лабораторной работе должен содержать:

- 1) оглавление;
- 2) описание цели работы;
- 3) описание хода работы в соответствии с данными методическими указаниями;
- 4) выводы по результатам выполнения работы;
- 5) список литературы;

Выводы

В результате выполнения данной лабораторной работы изучены принципы функционирования локального интерфейса микропроцессора Intel 80386 в режиме простого цикла и в режиме простого цикла с ожиданием.

Список литературы

1. Гуров В.В. Микропроцессорные системы: Учеб.пособие. –М.: ИНФРА-М, 2017. – 336 с. + Доп.материалы/ Электронный ресурс; Режим доступа <http://www.znaniium.com>]. –(Высшее образование: Бакалавриат).
2. Русанов, В.В. Микропроцессорные устройства и системы [Электронный ресурс] : учеб. пособие / В.В. Русанов, М.Ю. Шевелев. — Электрон. дан. — Москва : ТУСУР, 2012. — 184 с. — Режим доступа: <https://e.lanbook.com/book/10931>. — Загл. с экрана.
3. Борисов, В.В. Нечеткие модели и сети [Электронный ресурс] : учеб. пособие / В.В. Борисов, В.В. Круглов, А.С. Федулов. — Электрон. дан. — Москва : Горячая линия-Телеком, 2012. — 284 с. — Режим доступа: <https://e.lanbook.com/book/5126>. — Загл. с экрана.

6 Лабораторная работа №6 «Локальный интерфейс Intel 80386: конвейерный цикл»

6.1 Цель работы

Целью настоящей лабораторной работы является изучение функционирования локального интерфейса микропроцессора Intel 80386 в режиме конвейерного цикла.

6.2 Общие положения

В лабораторной работе №5 моделируются сигналы локального интерфейса микропроцессора Intel 80386.

В соответствии с заданными в лабораторной работе условиями необходимо установить требуемые уровни управляющих сигналов в различных системных состояниях при выполнении конвейерных циклов.

6.3 Задание на лабораторную работу №6

1. Ознакомиться с принципами функционирования локального интерфейса микропроцессора Intel 80386 в режиме конвейерного цикла (Приложение 2).

2. Построить временные диаграммы функционирования локального интерфейса микропроцессора Intel 80386 в режиме конвейерного цикла на основе графической модели.

6.4 Ход работы

В процессе работы выполнить следующие действия:

- 1) включить компьютер;
- 2) загрузить программную модель функционирования локального интерфейса микропроцессора Intel 80386;
- 3) построить временные диаграммы функционирования локального интерфейса микропроцессора Intel 80386 в режиме конвейерного цикла на основе графической модели;
- 4) представить результаты работы преподавателю в устной форме.

6.5 Работа локального интерфейса микропроцессора Intel 80386 в режиме конвейерного цикла

Результаты функционирования локального интерфейса микропроцессора Intel 80386 в режиме конвейерного цикла приведены на рис.1.

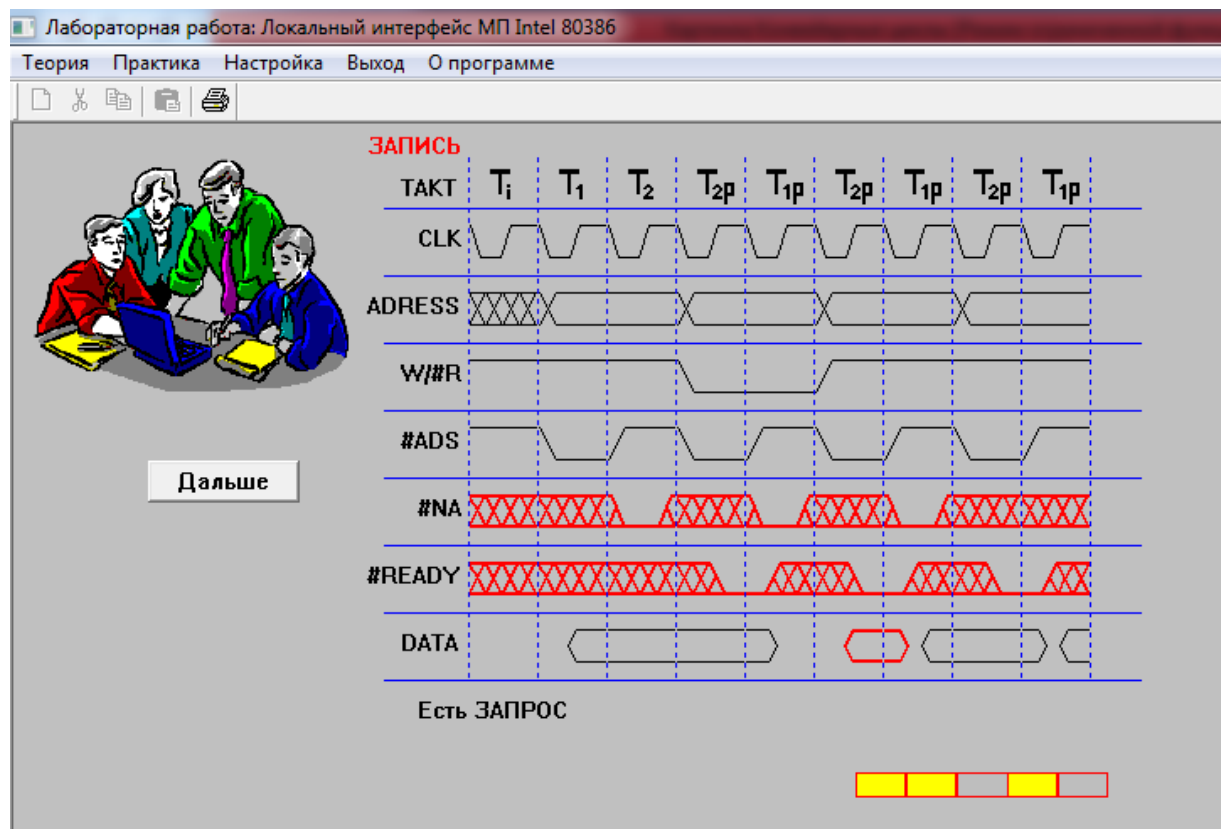


Рисунок 1 – Конвейерный цикл

6.6 Работа локального интерфейса микропроцессора Intel 80386 в режиме конвейерного цикла с ожиданиями

Результаты функционирования локального интерфейса микропроцессора Intel 80386 в режиме конвейерного цикла с ожиданиями приведены на рис.2.

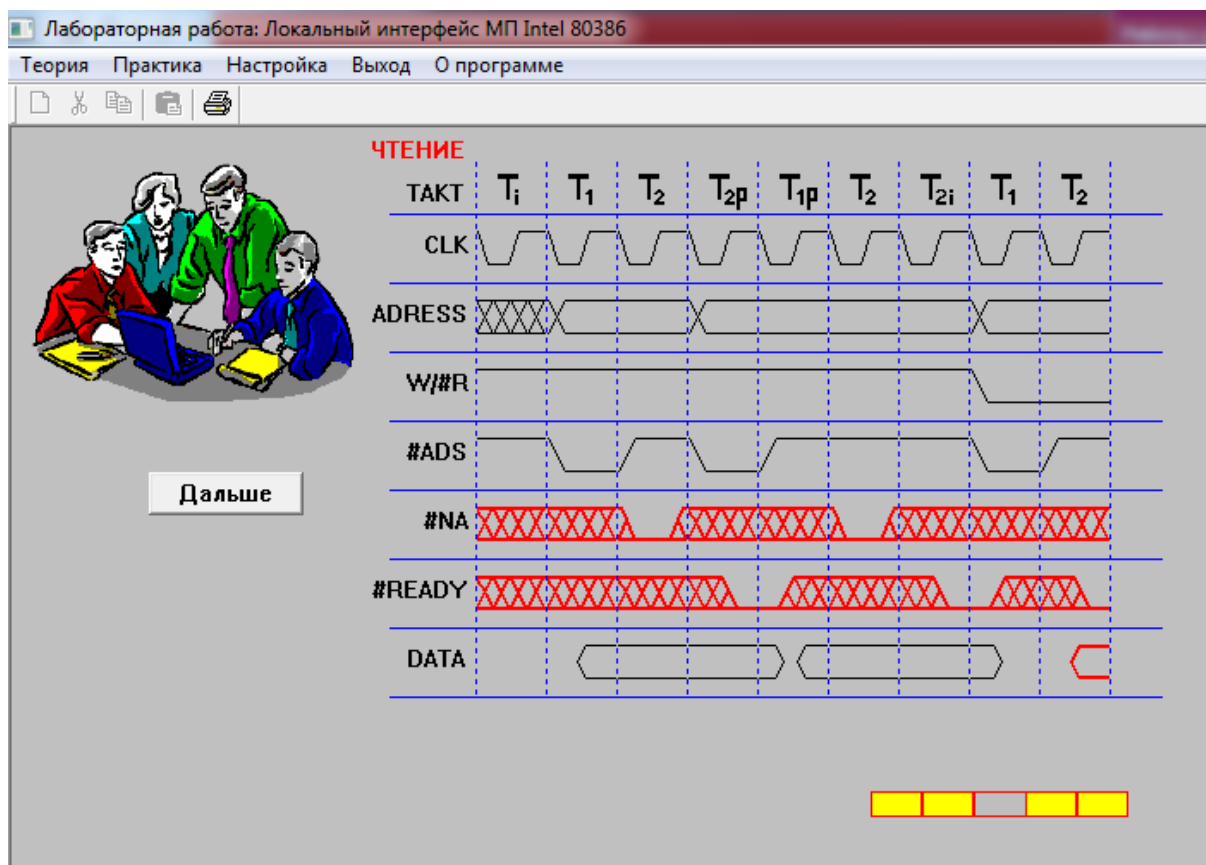


Рисунок 2 – Конвейерный цикл с ожиданиями

6.7 Представление результатов работы

Отчет по лабораторной работе должен содержать:

- 1) оглавление;
- 2) описание цели работы;
- 3) описание хода работы в соответствии с данными методическими указаниями;
- 4) выводы по результатам выполнения работы;
- 5) список литературы;

Выводы

В результате выполнения данной лабораторной работы изучены принципы функционирования локального интерфейса микропроцессора Intel 80386 в режиме конвейерного цикла и в режиме конвейерного цикла с ожиданием.

Список литературы

1. Гуров В.В. Микропроцессорные системы: Учеб.пособие. –М.: ИНФРА-М, 2017. – 336 с. + Доп.материалы/ Электронный ресурс; Режим доступа <http://www.znaniium.com>]. –(Высшее образование: Бакалавриат).
2. Русанов, В.В. Микропроцессорные устройства и системы [Электронный ресурс] : учеб. пособие / В.В. Русанов, М.Ю. Шевелев. — Электрон. дан. — Москва : ТУСУР, 2012. — 184 с. — Режим доступа: <https://e.lanbook.com/book/10931>. — Загл. с экрана.
3. Борисов, В.В. Нечеткие модели и сети [Электронный ресурс] : учеб. пособие / В.В. Борисов, В.В. Круглов, А.С. Федулов. — Электрон. дан. — Москва : Горячая линия-Телеком, 2012. — 284 с. — Режим доступа: <https://e.lanbook.com/book/5126>. — Загл. с экрана.

7 Лабораторная работа №7 «Локальный интерфейс Intel 80486: простой цикл»

7.1 Цель работы

Целью настоящей лабораторной работы является изучение функционирования локального интерфейса микропроцессора Intel 80486 в режиме простого цикла.

7.2 Общие положения

В лабораторной работе №7 моделируются сигналы локального интерфейса микропроцессора Intel 80486.

В соответствии с заданными в лабораторной работе условиями необходимо установить требуемые уровни управляющих сигналов в различных системных состояниях при выполнении простых циклов.

7.3 Задание на лабораторную работу №7

1. Ознакомиться с принципами функционирования локального интерфейса микропроцессора Intel 80486 в режиме простого цикла (Приложение 3).
2. Построить временные диаграммы функционирования локального интерфейса микропроцессора Intel 80486 в режиме простого цикла на основе графической модели.

7.4 Ход работы

В процессе работы выполнить следующие действия:

- 1) включить компьютер;
- 2) загрузить программную модель функционирования локального интерфейса микропроцессора Intel 80486;
- 3) построить временные диаграммы функционирования локального интерфейса микропроцессора Intel 80486 в режиме простого цикла на основе графической модели;
- 4) представить результаты работы преподавателю в устной форме.

7.5 Работа локального интерфейса микропроцессора Intel 80486 в режиме простого цикла

Результаты функционирования локального интерфейса микропроцессора Intel 80486 в режиме простого цикла приведены на рис.1.

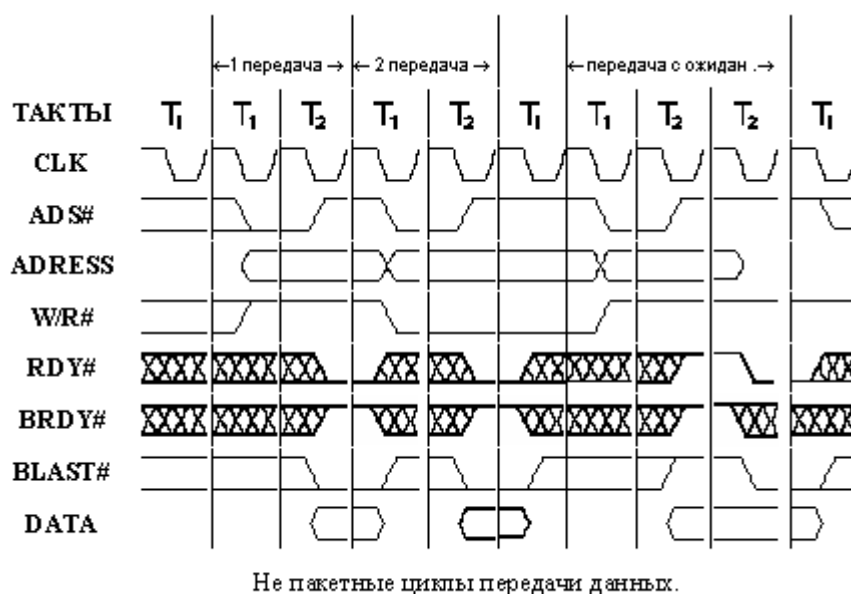


Рисунок 1 – Простые циклы

7.6 Представление результатов работы

Отчет по лабораторной работе должен содержать:

- 1) оглавление;
- 2) описание цели работы;
- 3) описание хода работы в соответствии с данными методическими указаниями;
- 4) выводы по результатам выполнения работы;
- 5) список литературы;

Выводы

В результате выполнения данной лабораторной работы изучены принципы функционирования локального интерфейса микропроцессора Intel 80486 в режиме простого цикла.

Список литературы

1. Гуров В.В. Микропроцессорные системы: Учеб.пособие. –М.: ИНФРА-М, 2017. – 336 с. + Доп.материалы/ Электронный ресурс; Режим доступа <http://www.znanium.com>]. –(Высшее образование: Бакалавриат).
2. Русанов, В.В. Микропроцессорные устройства и системы [Электронный ресурс] : учеб. пособие / В.В. Русанов, М.Ю. Шевелев. — Электрон. дан. — Москва : ТУСУР, 2012. — 184 с. — Режим доступа: <https://e.lanbook.com/book/10931>. — Загл. с экрана.
3. Борисов, В.В. Нечеткие модели и сети [Электронный ресурс] : учеб. пособие / В.В. Борисов, В.В. Круглов, А.С. Федулов. — Электрон. дан. — Москва : Горячая линия-Телеком, 2012. — 284 с. — Режим доступа: <https://e.lanbook.com/book/5126>. — Загл. с экрана.

8 Лабораторная работа №8 «Локальный интерфейс Intel 80486: последовательность циклов и пакетный цикл»

8.1 Цель работы

Целью настоящей лабораторной работы является изучение функционирования локального интерфейса микропроцессора Intel 80486 в режиме последовательности циклов и в режиме пакетного цикла.

8.2 Общие положения

В лабораторной работе №8 моделируются сигналы локального интерфейса микропроцессора Intel 80486.

В соответствии с заданными в лабораторной работе условиями необходимо установить требуемые уровни управляющих сигналов в различных системных состояниях при выполнении последовательности циклов и в режиме пакетного цикла.

8.3 Задание на лабораторную работу №8

1. Ознакомиться с принципами функционирования локального интерфейса микропроцессора Intel 80486 в режиме последовательности циклов и в режиме пакетного цикла (Приложение 3).

2. Построить временные диаграммы функционирования локального интерфейса микропроцессора Intel 80486 в режиме последовательности циклов и в режиме пакетного цикла на основе графической модели.

8.4 Ход работы

В процессе работы выполнить следующие действия:

- 1) включить компьютер;
- 2) загрузить программную модель функционирования локального интерфейса микропроцессора Intel 80486;
- 3) построить временные диаграммы функционирования локального интерфейса микропроцессора Intel 80486 в режиме последовательности циклов и в режиме пакетного цикла на основе графической модели;
- 4) представить результаты работы преподавателю в устной форме.

8.5 Работа локального интерфейса микропроцессора Intel 80486 в режиме последовательности циклов

Результаты функционирования локального интерфейса микропроцессора Intel 80486 в режиме последовательности циклов приведены на рис.1.

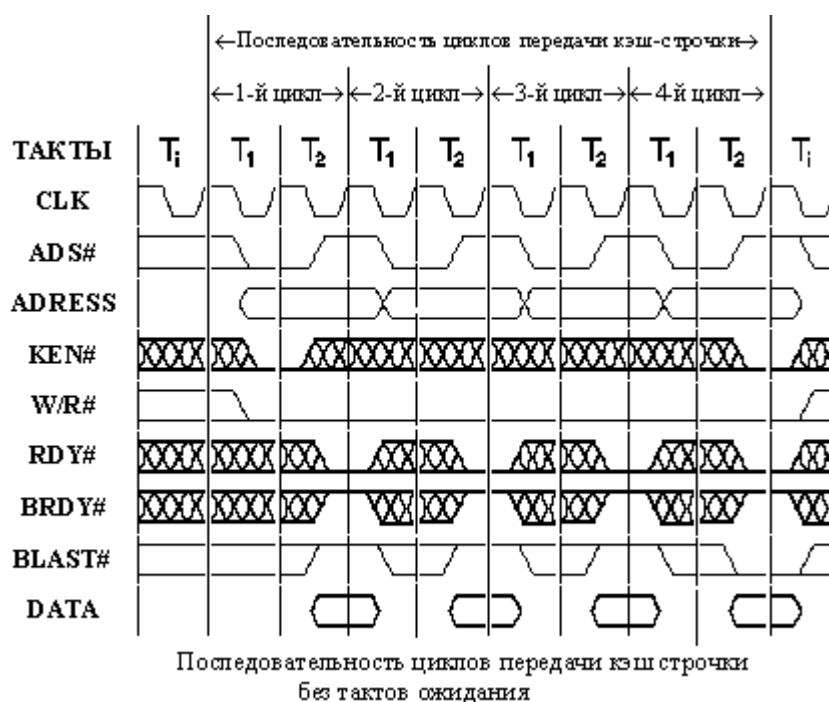
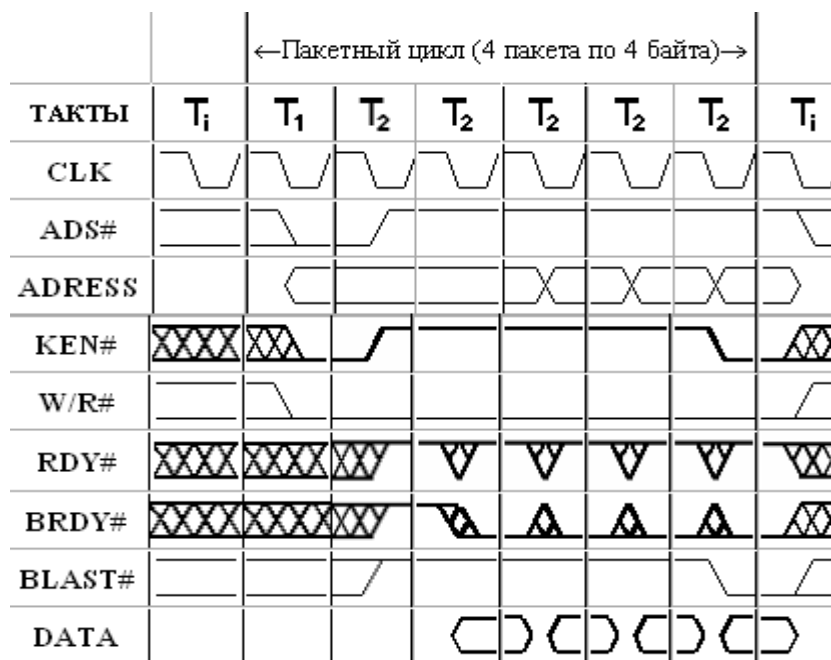


Рисунок 1 – Последовательность циклов

8.6 Работа локального интерфейса микропроцессора Intel 80486 в режиме пакетных циклов

Результаты функционирования локального интерфейса микропроцессора Intel 80486 в режиме пакетных циклов приведены на рис.2.



Пакетный цикл передачи кэш-строчки с одним тактом ожидания перед передачей данных.

Рисунок 2 – Пакетный цикл

8.7 Представление результатов работы

Отчет по лабораторной работе должен содержать:

- 1) оглавление;
- 2) описание цели работы;
- 3) описание хода работы в соответствии с данными методическими указаниями;
- 4) выводы по результатам выполнения работы;
- 5) список литературы;

Выводы

В результате выполнения данной лабораторной работы изучены принципы функционирования локального интерфейса микропроцессора Intel 80486 в режиме последовательности циклов и в режиме пакетных циклов.

Список литературы

1. Гуров В.В. Микропроцессорные системы: Учеб.пособие. –М.: ИНФРА-М, 2017. – 336 с. + Доп.материалы/ Электронный ресурс; Режим доступа <http://www.znaniium.com>]. –(Высшее образование: Бакалавриат).
2. Русанов, В.В. Микропроцессорные устройства и системы [Электронный ресурс] : учеб. пособие / В.В. Русанов, М.Ю. Шевелев. — Электрон. дан. — Москва : ТУСУР, 2012. — 184 с. — Режим доступа: <https://e.lanbook.com/book/10931>. — Загл. с экрана.
3. Борисов, В.В. Нечеткие модели и сети [Электронный ресурс] : учеб. пособие / В.В. Борисов, В.В. Круглов, А.С. Федулов. — Электрон. дан. — Москва : Горячая линия-Телеком, 2012. — 284 с. — Режим доступа: <https://e.lanbook.com/book/5126>. — Загл. с экрана.

Формат команды микропроцессора Intel 80386

Команда микропроцессора Intel 80386 может быть короткой один байт или длинной в несколько байтов. Обязательным является только байт кода операции. Код операции задаёт основные действия, которые выполняет данная команда, все остальные поля, так или иначе, связаны с адресацией операнда.

Префикс	Код операции	W	md	reg	r/m	dispH	dispL	ДатаH	ДатаL
1б	1...2 байта	постбайт			смещение		данные		

Префикс имеет размер в один байт и указывает на то, какой из сегментных регистров будет использоваться.

В зависимости от состояния битов **SR** поля префикса выбирается сегментный регистр:

SR=00 - ES дополнительный сегмент данных;

SR=01 - CS сегмент команд;

SR=10 - SS сегмент стека;

SR=11 - DS сегмент данных.

Состояние разрядов постбайта задает режимы адресации операндов.

Один из разрядов постбайта указывает на режим работы с байтами или словами, этот разряд имеет обозначение **W**, при

W=1 –работаем со словами

W=0 –работаем с байтами

При работе со словами используется обозначение регистров:

AX, BX, CX, DX, SP, BP, SI, DI

При работе с байтами используется байтовая часть регистров:

AH, AL, BH, BL, CH, CL, DH, DL

Следующие за постбайтом два байта смещения задают необходимое смещение внутри команды DispH DispL.

Использование варианта смещения задаётся группой битов **md** (Mode).

Таким образом, поле **md** задает режим адресации операндов. Это поле используется вместе со следующим полем **r/m** (register memory). Совместное использование этих двух полей позволяет определить способ адресации операнда.

Если **md=11** – операнд содержится в регистре.

Во всех остальных случаях операнды находятся в памяти, в этом случае значение кода **md** определяет величину смещения, которое задается в следующих байтах команды **DispH DispL**:

md=00 – $disp=0$ – смещение равно нулю;

md=01 – **dispL** - используется восьмибитовое смещение, т.е. используется один младший байт поля смещения;

md=10 – **dispHdispL** - используется шестнадцатиразрядное смещение.

Если величина численного значения смещения, заданного командой языка ассемблера, не превышает значения **127** для положительных значений или **128** – для отрицательных, то смещение задается байтом (**md=01**), иначе – словом (**md=10**). При нулевом значении смещения, байты смещения в машинном коде не используются (**md=00**).

Существует исключение: если задается прямая адресация:

md=00; **rm=110**; **disdLdispH** – является адресом операнда.

Если в формуле команды используется символическое имя (метка), то смещение (**disp**) всегда имеет размерность слова (2 байта) и в постбайте значение **md** следует задать равным **md=10**.

Сводная таблица адресации

r/m	md		
	00	01	10
000	(BX)+(SI)	(BX+SI)+disp8	(BX+SI)+ disp16
001	(BX)+(DI)	(BX+DI)+ disp8	(BX+DI)+ disp16
010	(BP)+(SI)	(BP+SI)+ disp8	(BP+SI)+ disp16

011	(BP)+(DI)	(BP+DI)+ disp8	(BP+DI)+disp16
100	(SI)	(SI)+ disp8	(SI)+disp16
101	(DI)	(DI)+ disp8	(DI)+ disp16
110	disp16	(BP)+ disp8	(BP)+ disp16
111	(BX)	(BX)+disp8	(BX)+ disp16

Поле `reg` занимает 3 разряда и следует в команде за полем **`md`**.

В зависимости от значения поля **`reg`** выбирается регистр, с которым будет выполняться операция. На выбор регистра соответствующим образом влияет бит **`W`**.

red	w=0	w=1
000	AL	Ax
001	CL	Cx
010	DL	Dx
011	BL	Bx
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

Поле для задания данных `DateH` и `DateL`

Эти поля служат для задания данных, т.е. операндов внутри команды.

Таким образом, операнд может быть размер слова или размер байта. Данные в команду помещаются при использовании непосредственной адресации. В случае байтовой операции (**`W=0`**) непосредственное значение байтового операнда располагается в байте **`DateL`**, если команда работает со словом, т.е. **`W=1`**, то используются оба поля **`DateH`** и **`DateL`**.

В исходном тексте программы на языке Ассемблер (формуле команды) данные и смещения записываются в десятичном виде, а в машинный код ко-

манды их нужно поставить в двоичном виде. Для перевода десятичных чисел в двоичные в программной модели имеется специальная таблица перевода, отрицательные числа записываются в особом виде.

Для задания размерности операндов в ассемблере Intel существует специальный оператор указания типа операнда **PTR**.

Byte **PTR** Alfa; - переменная Alfa имеет размер байта;

Word **PTR** Delta; - переменная Delta имеет размер слова.

В программной модели команд микропроцессора Intel 80386 используются некоторые команды, которые работают с ячейками памяти, помеченными метками. В этом случае необходимо обратиться к таблице значений меток, в которой показано, какое значение имеет данная метка.

Например: ma Byte Date 0002

Запись означает, что данная ячейка **ma** содержит операнд байтовой размерности, т.е. с этим операндом должна выполняться операция байтовой размерности ($W=0$). Слово **Date** означает, что операнд представляет собой данные и для его адресации необходимо выбрать сегмент данных **DS**. Кроме того, внутри этой ячейки находится смещение $+2_{16}$, т.е. при нахождении смещения (*disp*) для адресации операнда смещение в формуле команды алгебраически складывается с этим внутренним смещением.

1 Локальный интерфейс Intel 80386

Локальная шина соединяет процессор с сопроцессором оперативной памятью и КЭШ-памятью, а через конвертер (буфер) с другими устройствами микро ЭВМ.

Локальная шина работает в синхронном режиме. Основными компонентами локального интерфейса является **A**-адресная шина, **D** – шина данных, сигналы идентификации адресов и строб адреса, строб данных. Можно выделить следующие особенности функционирования локального интерфейса.

Локальный интерфейс микропроцессора Intel является многоуровневой шиной, линии этой шины непосредственно соединяются на плате с выводами микросхемы процессора. Основным назначением локального интерфейса является передача информационных слов между основными элементами микропроцессорной системы: процессор, сопроцессор, ОЗУ, КЭШ-память и др.

Основными компонентами интерфейса являются:

1. Адресная шина **A**;
2. Шина данных **D**;
3. Сигналы идентификации циклов;
4. Строб адреса;
5. Строб данных.

Микропроцессор Intel 80386 имеет разрядность 32 бита и может обращаться к сегментам памяти в диапазоне 4 Гбайта и портам ввода/вывода в диапазоне 64 Кбайта. При этом периферийные устройства в микропроцессорной системе могут быть отнесены, либо к пространству памяти, либо к пространству ввода/вывода. По линиям интерфейса могут передаваться операнды размерностью 40 разрядов или 32. Особенность этого процессора состоит в том, что по линиям интерфейса могут передаваться параллельно 4 байта. Внутри процессора имеются сигналы выбора байт. Эти сигналы такие: **BE0#**, **BE1#**, **BE2#**, **BE3#**. Кодовое сочетание этих сигналов используется для адресации секции шины данных, по которым передается байт данных. Значок # использу-

ется для обозначения того, что активным этот сигнал воспринимается, если имеет низкий уровень, т.е. единичное значение закодировано низким уровнем. Забегая вперед, скажем, что все единичные значения сигналов на шине кодируются низкими уровнями.

BE0# - “A” 0...7p

BE1# - “B” 8...15p

BE2# - “C” 16...23p

BE3# - “D” 24...31p

Одной из особенностей интерфейса Intel 80386 является то, что шина позволяет работать в режиме конвейеризации адреса. Смысл этого приема заключается в том, что пока процессор выполняет операцию чтения или записи по зафиксированному в регистре адреса памяти текущему адресу, процессор может выставить на шину адрес следующей ячейки памяти, с которой будет выполнена операция в следующем цикле. Такой прием существенно повышает быстродействие микропроцессорной системы. Другой особенностью работы системного интерфейса является применение сигналов идентификации микропроцессорного цикла. Циклом (транзакцией) – называется один сеанс связи устройств, в течение которого осуществляется передача данных через локальный интерфейс. Здесь используется синхронная передача, каждый цикл имеет несколько фаз, называемых тактами. Длительность такта задается синхросигналами, которые доступны всем устройствам, подключенным к локальному интерфейсу. Циклы могут иметь переменное число тактов в зависимости от сложившейся на шине ситуации, минимальное число тактов в цикле – 2.

В первом такте микропроцессор задает адрес устройства, с которым будет работать и тип цикла. Первый такт помечается стробом задатчика, в качестве которого используется сигнал стробирования адреса ADS#.

В последнем такте оперативная память или другой исполнитель выставляют строб данных. Этим стробом является сигнал окончания цикла READY, т.е. этим сигналом стробируются данные на шине.

При операциях чтения, когда осуществляется передача данных от исполнителя к задатчику, исполнитель этим сигналом указывает такт, в котором задатчик может снять данные с шины данных.

При выполнении операция записи, исполнитель этим сигналом указывает задатчику такт, в котором были сняты данные с шины данных, т.е. это своеобразная квитанция исполнителю и сообщение задатчику, что текущий цикл можно завершить и перейти на следующий цикл.

Минимальное число тактов в цикле (2 такта) получается в том случае, когда быстроедействие задатчика и исполнителя согласовано. Если это быстроедействие не согласовано, то в цикле возможны дополнительные такты, которые позволяют процессору дождаться выполнения команды медленными устройствами.

В общем случае, микропроцессор может выполнять следующие типы циклов:

1. Чтение из памяти;
2. Запись в память;
3. Чтение из устройства ввода/вывода;
4. Запись в устройство ввода/вывода;
5. Подтверждение к прерыванию;
6. Индикация остановок или выключение.

Тип операции в цикле запись или чтение определяется сигналом W/R: запись «H» - высокий уровень, чтение «L» - низкий уровень, сигнал D/C – определяет тип передаваемой информации: данные или команда. Следующий сигнал M/IO – обращение к памяти или обращение к устройству ввода/вывода. LOCK – блокировка шины в последовательности цикла

2 Модель функционирования локального интерфейса микропроцессора Intel 80386

Данная модель выполнена в виде графа, в котором вершинами представлены состояния микропроцессора, а дугами – переходы из одного состояния в другой (Рис.1). Вершиной графа обозначены символы тактов микропроцессора, а дуги обозначают условия перехода из одного состояния в другое.

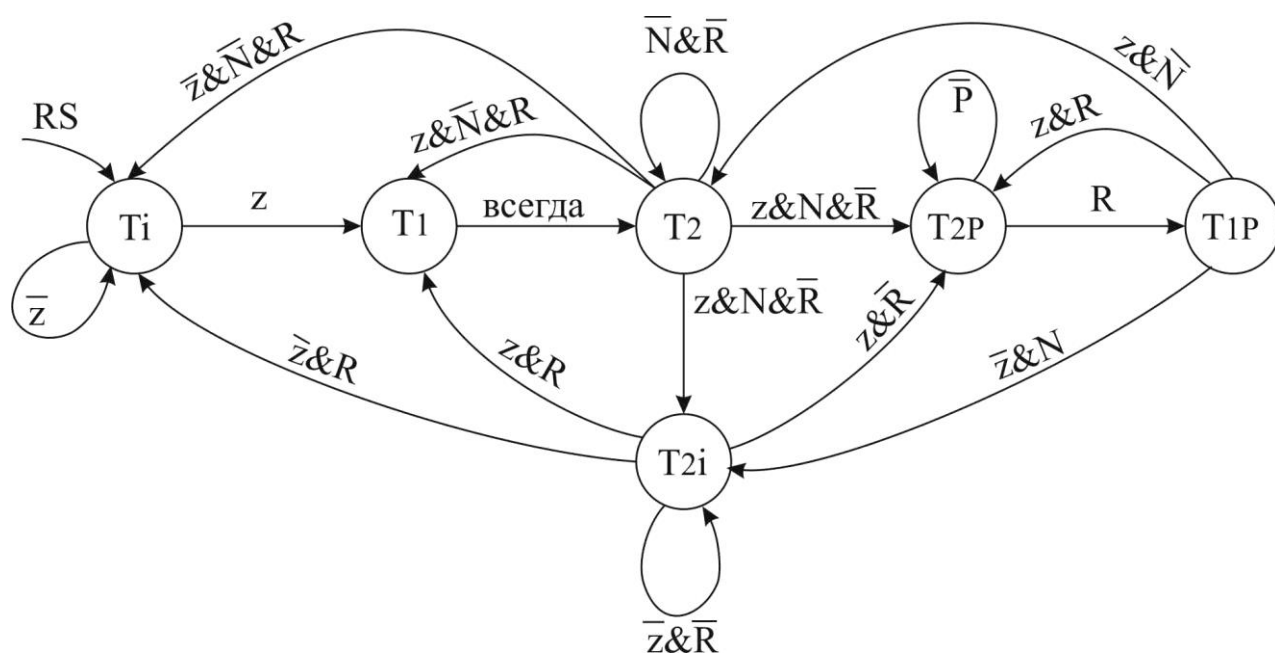


Рисунок 1 - Модель функционирования локального интерфейса микропроцессора Intel 80386

Пусть первоначально процессор находится в состоянии T_i – это состояние холостого хода, т.е. отсутствие циклов передачи информации. Это начальный такт после включения микропроцессора.

В этом состоянии микропроцессор будет находиться до тех пор, пока не появится внутренний сигнал Z – сигнал запроса на цикл интерфейса.

При появлении сигнала Z микропроцессор переходит в состояние T_1 , которое является начальным тактом простого (не конвертированного) цикла передачи информации.

В такте T1 процессор выставляет следующие сигналы: адрес устройства, с которым будет выполняться обмен данными, сигналы идентификации цикла, (он выставляет M/IO, D/C, W/R), сигнал стробирования адреса ADS, чтобы приемник смог воспринять адрес, выставленный на шине адреса.

При выполнении цикла записи процессор выставляет данные на шину данных.

При выполнении цикла чтения данные с шины снимаются.

На этом функции такта T1 заканчиваются, после него всегда следует такт T2.

В этом такте, который называется тактом продолжения или завершения цикла, процессор выполняет следующие действия:

1. Сохраняет значения адресов;
2. Сохраняет значения сигнала операции записи или чтения;
3. Переводит в неактивное состояние строб адреса ADS.

При выполнении цикла записи значение данных на шине сохраняется.

Поскольку такт T2 является завершающим в цикле, то процессору нужно понять завершен ли цикл и куда следует двигаться дальше.

Для этого процессор в течение такта T2 проверяет активность следующих сигналов:

- 1) сигнал внутреннего запроса Z;
- 2) сигнал запроса следующего адреса NA (в случае конвейерной обработки);
- 3) сигнал окончания текущего цикла Ready (на графе сигнал R).

Если сигнал окончания текущего цикла R не поступал, и нет запроса следующего адреса NA, то процессор продолжает оставаться в состоянии T2.

Если поступил сигнал внутреннего запроса Z и присутствует сигнал окончания цикла R, но нет сигнала запроса следующего адреса NA, то процессор переходит к циклу T1.

Если же сигнал окончания цикла R появляется, но нет сигнала внутреннего запроса Z и запроса следующего адреса NA, то микропроцессор переходит в состояние холостого такта T2i.

Если процессор находится в состоянии T2 и текущий цикл еще не закончился (отсутствует сигнал R), а появляется запрос нового адреса NA, то это означает, что начинается работа с конвейеризацией, т.е. начинается новый цикл передач до окончания текущего.

Процессор переходит в состояние T2P – это такт передачи нового адреса в цикле с конвейеризацией.

Процессор в этом такте выставляет:

1. Адрес данных;
2. Сигнал использования ADS;
3. Сигналы идентификации цикла.

В отличие такта T1, в такте T2P процессор не выставляет на шину новые данные до окончания текущего цикла.

Выход из состояния T2P возможен только при наличии сигнала окончания цикла R.

При появлении сигнала R процессор переходит в состояние T1P.

Такт T1P называется началом конвейерного цикла, процессор здесь повторяет все ранее выставленные сигналы, кроме stroba адреса ADS, который процессор переводит в неактивное состояние.

Если есть внутренний запрос Z и запрос следующего адреса NA, то это означает продолжение конвейерной обработки и процессор переходит в состояние T2P.

Если есть внутренний запрос Z, но нет запроса следующего адреса NA, то процессор переходит в состояние T2.

Если запрос следующего адреса NA есть, но нет внутреннего запроса Z, то процессор переходит на промежуточный такт T2i – это такт завершения цикла конвейеризации.

Процессор в этом случае повторяет все выставленные сигналы, а строб адреса ADS сбрасывает, если он находился в активном состоянии. В этом такте процессор проверяет наличие сигнала внутреннего запроса Z и сигнала R. Если их нет, то процессор продолжает оставаться в состоянии T2i.

Когда сигнал внутреннего запроса Z появляется, но нет сигнала R, то процессор переходит в состояние T2p.

Если сигнал внутреннего запроса Z появился и цикл закончен (т.е. появился сигнал R), то процессор переходит к началу следующего цикла, т.е. в состояние T1.

Если сигнал внутреннего запроса Z отсутствует, а цикл закончен (т.е. появился сигнал R), то необходимо дождаться сигнала внутреннего запроса Z и для этого процессор переходит в состояние Ti.

Если процессор, находящийся в состоянии T2 увидит наличие сигнала запроса следующего адреса NA для работы в конвейерном режиме, но при этом не будет сигнала внутреннего запроса Z и нет сигнала окончания цикла R, то процессор перейдет в состояние промежуточного такта T2i.

3 Диаграммы работы

После запуска или процедуры рестарта интерфейс устанавливается в состояние холостых тактов Ti. Это отсутствие циклов передачи. Цикл передачи начинается по внутреннему запросу процессора Z. Локальный интерфейс МП i80386 поддерживает два типа циклов связи:

- 1) простые циклы;
- 2) конвейерные циклы.

3.1 Простые циклы

Это циклы без конвейеризации. Простые циклы содержат минимально два такта (рис.1). При работе в не конвейерном режиме в первом состоянии T1 процессор выдаются на шины интерфейса адрес и сигналы определения типа цикла (W/R#), и одновременно устанавливается активный уровень stroба адреса (ADS#), означающий начало цикла.

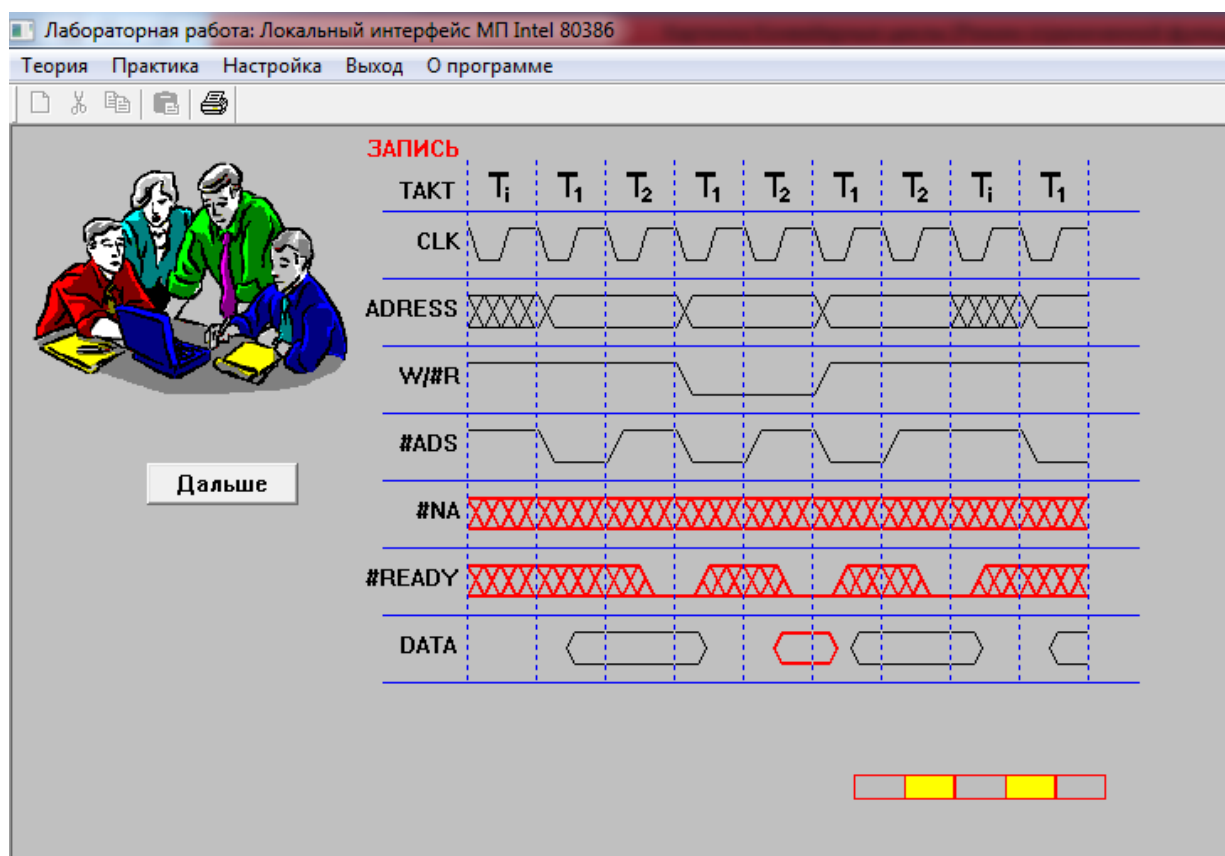


Рисунок 1 – Простой цикл

Во время выполнения цикла чтения МП переключает шину данных таким образом, чтобы принять данные от адресованного внешнего устройства в следующем такте.

В цикле записи МП устанавливает данные во второй фазе состояния T1 и удерживает их в первой фазе следующего такта. Следующим тактом после T1 всегда является такт продолжения и завершения цикла (T2). В такте T2 процессор сохраняет значение адреса и значение сигнала операции W/R#, переводит в неактивное (единичное) состояние сигнал ADS#, в цикле записи (W/R# = 1) сохраняет значение данных.

3.2 Конвейерные циклы

Первый цикл (первой передачи данных на рис.2) после такта T_i всегда выполняется без конвейеризации адреса. Для того, чтобы следующий цикл обращения к памяти был конвейеризованным, вместо простого много тактного (с ожиданиями), память (как исполнитель) должна в предпоследнем такте T_2 (за один такт до выдачи сигнала $READY\#$) установить активный уровень сигнала выдачи адреса $NA\#$ (рис.2). МП по активному сигналу выдачи адреса $NA\#$, инициализирует вместо такта T_2 такт T_{2p} , в котором на шине МП устанавливается адрес и управляющие сигналы для выполнения следующего цикла обращения к памяти (вторая передача на рис.2).

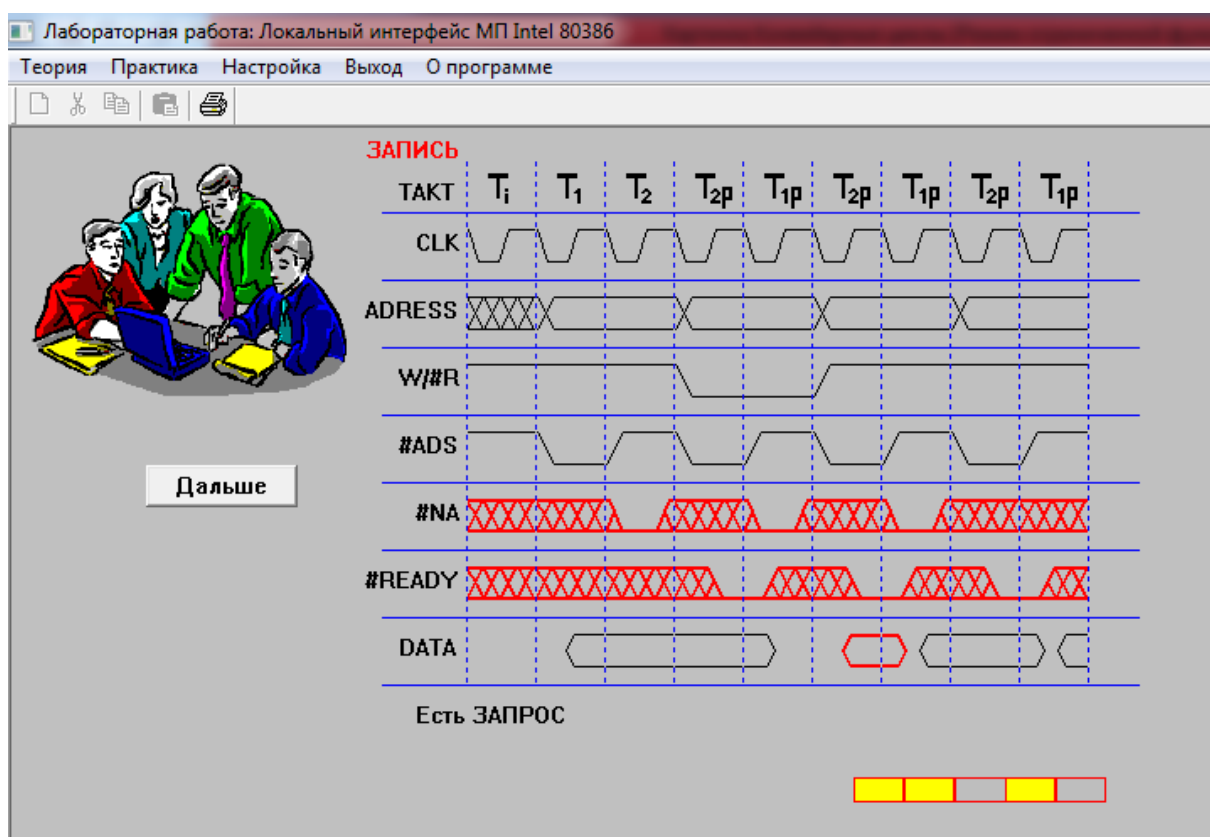


Рисунок 2 – Конвейерный цикл

Цикл из последовательности T_1 , T_2 , T_{2p} считается неконвейерным, но переходным, так как в такте T_{2p} заканчивается текущая передача данных и начинается следующая передача данных, а именно, от задатчика к исполнителю передается адрес для следующего цикла. Таким образом, еще до окончания операции чтения или записи на шине адреса выставляется новый адрес.

Это вполне допустимо, так как текущий адрес уже зафиксирован (в регистре адреса памяти) в такте T2, а следующий адрес, вероятнее всего, будет адресовать другой банк памяти (при расслоении адресов). В случае же повторной адресации одного и того же банка памяти, произойдет повторение такта T2p до освобождения адресуемого банка памяти. Но, в такте T2p, в циклах чтения память (или другой исполнитель) выставляет на шину данных запрошенные данные. Поэтому, в отличие от такта T1 процессор не может выставлять данные (при записи в следующем цикле). Данные выставляются после завершения такта T2p. Задержка по данным при трех тактных передачах допустима, так как они используются только после задержки в дешифрации адреса обращения.

Одновременно такт T2p является завершающим тактом текущей передачи. В нем процессор проверяет сигнал `READY#` окончания цикла текущей передачи (первой на рис.2).

1 Локальный интерфейс МП i80486

Программная модель МП i80486 позволяет сформировать последовательность тактов работы шины, выставить адреса, данные и необходимые стробы сопровождения на шину.

В МП i486, как и в МП i80386, шина адреса содержит:

- 34 линии для задания адреса операнда с точностью до двойного слова;
- 4 линии для задания адреса операнда с точностью до байта;

секций шины данных, по которым передаются байты операнда (сигналы выбора байтов: **BE3#**, **BE2#**, **BE1#** и **BE0#**).

Локальный интерфейс МП i486 также допускает передачу шестнадцати и тридцати двух разрядных операндов, не выровненных по границам слова или двойного слова. При этом возможна параллельная передача по шине данных от одного до четырех байт.

2 Особенности локального интерфейса МП i80486

Корпус микропроцессора i486 имеет существенно большее количество выводов по сравнению с микропроцессором i80386. Поэтому локальный интерфейс микропроцессора i486 получил дополнительные системные возможности.

2.1 Добавление новых сигналов

Основные новые сигналы используются для организации пакетной передачи данных:

- 1) **KEN#** (cache enable) – предназначен для указания возможности обмена с кэш-памятью;
- 2) **BS8#** (bus size 8 bits) – сигнал от устройства к процессору, означающий, что исполнитель – 8-разрядное устройство по шине данных;
- 3) **BREADY#** – сигнал окончания такта передачи данных в пакетных циклах;
- 4) **BLAST#** – сигнал ожидания последнего такта в пакетном цикле;

- 5) **PLOCK#** – дополнительный сигнал идентификации цикла (псевдоблокированный). Сигнализирует, что процессор производит комбинированную операцию: чтение-модификация-запись.

2.2 Кодировка циклов

В микропроцессоре i486 используются следующие сигналы идентификации циклов:

- 1) **W/R#** – операция в цикле (запись или чтение),
- 2) **D/C#** – тип передаваемой информации (данные или команды),
- 3) **M/IO#** – тип обращения (к памяти или к устройствам ввода/вывода),
- 4) **LOCK#** и **PLOCK#** – блокированный и псевдоблокированный типы циклов.

Таблица 1 – Типы циклов микропроцессора i486

Кодирование типов цикла в МП i486.				
M/IO#	D/C#	W/R#	LOCK#	Тип цикла шины
0	0	0	0	Подтверждение прерывания
0	0	1	0	<i>Останов</i>
0	1	0	1	Чтение данных из устройства в/в
0	1	1	1	Запись данных в устройство в/в
1	0	0	1	Чтение команды из памяти
1	0	1	1	<i>Не возникает</i>
1	1	0	0/1	Чтение данных из памяти
1	1	1	0/1	Запись данных в память

Отличием в кодировке циклов для МП i486 (по отношению к МП i80386) заключается в кодировании состояний "Останов" и неиспользуемого состояния (в таблице – *курсив*).

2.3 Авто конфигурация шины данных

В микропроцессоре i486 имеется возможность динамического изменения ширины шины данных. Эта возможность обеспечивается при взаимодействии не только с 32-разрядными и с 16-разрядными, но и с 8-разрядными внешними устройствами.

Для взаимодействия с 8-ми разрядными устройствами локальный интерфейс содержит, кроме линии для передачи сигнала **BS16#**, дополнительную линию для передачи сигнала **BS8#** от устройства к процессору.

Активные сигналы на линиях **BS8#** или **BS16#** означают, что исполнитель, соответственно, восьми или шестнадцатиразрядное устройство.

При обнаружении в тактах приема данных любого активного сигнала изменения разрядности шины данных (**BS8#** или **BS16#**), микропроцессор автоматически продолжает передачу данных до передачи последнего байта.

Но, в отличие от локального интерфейса МП i80386, в МП i486 повторная передача "не принятых" байт данных производится по тем же секциям данных (в соответствии с их адресами в двойном слове). При этом устройство (память), а не процессор, переключает линии интерфейса для приема данных.

Для этого контроллер устройства должен иметь:

- 1) все 32 входные линии шины данных,
- 2) коммутатор выбора секций входной шины данных,
- 3) устройство формирования сигналов автоконфигурации шины данных, и управления коммутатором и значение младших разрядов адреса (**A1, A0**).

2.4 Использование встроенной кэш-памяти

Микропроцессор i486 имеет в своем корпусе процессор операций с плавающей запятой и встроенный блок локальной кэш-памяти команд и данных на 8Кб.

Размер блока (строки) кэш-памяти – 16 байт. При этом в МП i486 сохранен 16-байтовый буфер кодовой строки. Команды на регистр команд выбира-

ются из этого буфера со сдвигом содержимого и освобождением его младших разрядов. Но, после освобождения четырех младших байт этого буфера формируется запрос не к оперативной памяти, а к кэш-памяти. Внутренний запрос **Z** формируется только при кэш-промахах, как при обращении к данным, так и при выборке строки кода.

В МП i486 используется кэш-память со сквозной записью. Этим определяется размер передачи данных в операциях записи – два двойных слова (4 байта). Но при обновлении строки кэш-памяти – выбирается вся строка в 16 байт (128 бит).

Кроме обращений к оперативной памяти, процессор может производить обмен данными через локальный интерфейс и конверторы шин расширения с другими устройствами. В общем случае возможны запросы процессора на передачи данных от одного до 16 байт.

Разрядность шины данных в МП i486 осталась 32-х разрядной. Для передачи 16 байт, даже при использовании памяти, обеспечивающей самый короткий двух тактового цикла передачи, потребуется последовательность из четырех обычных циклов передачи.

Конвейерная передача адреса здесь проблему не решает.

2.5 Организации пакетных циклов

Для пакетной передачи данных необходима поддержка режима пакетных циклов со стороны других устройств (памяти или ввода/вывода). Для универсальности работы с разными устройствами предусмотрены множество режимов передач:

- 1) простые (не пакетные) циклы;
- 2) последовательность циклов;
- 3) пакетные циклы.

2.6 Передачи данных с использованием простых циклов

При любых передачах первым тактом в цикле является такт выставления процессором на линиях интерфейса адреса, сигналов идентификации цикла. В первом же такте, если производится обращение к памяти (после кэш-промаха по чтению), процессор проверяет активность сигнала разрешения кэширования **KEN#**. При запрещенном кэшировании запрос на данные ограничивается удвоенным словом (4 байта). Такая передача реализуется простым (не пакетным циклом).

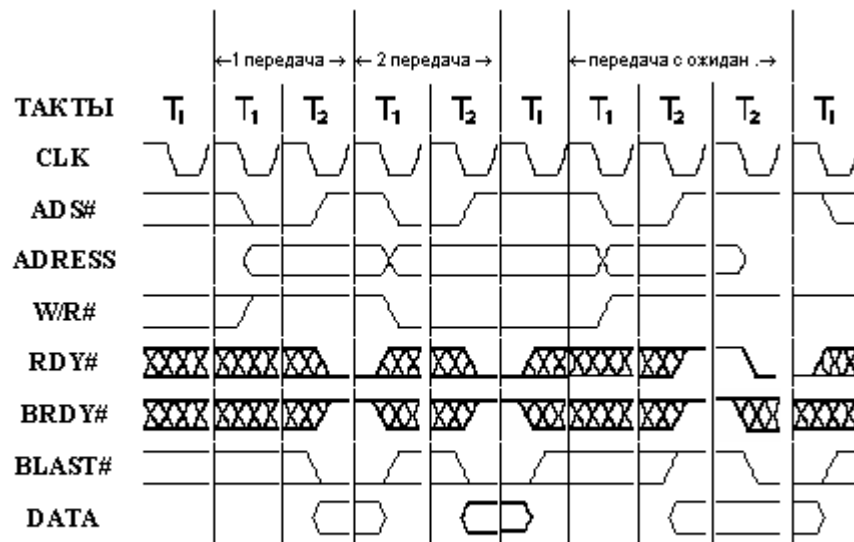
Во втором такте простого цикла в цикле записи процессор выставляет данные на шине данных, а в цикле чтения – ожидает приема данных от устройства. Цикл оканчивается по сигналу **READY#**. Этот же сигнал для процессора служит стробом готовности данных. Для сообщения устройству, что данный цикл является простым (не пакетным), процессор выставляет в такте **T2** сигнал **BLAST#**.

Схема передачи: такт передачи адреса, такт передачи данных.

При использовании менее быстродействующих устройств возможны циклы с дополнительными тактами **T2**. Серии дополнительных тактов оканчиваются сигналами **READY#** от устройства.

Схема передачи: такт передачи адреса, такты ожидания, такт передачи данные.

Диаграмма простых циклов без ожидания и с ожиданиями представлена на рис. 1. В диаграмме приведен случай использования 32-разрядных устройств.



Не пакетные циклы передачи данных.

Рисунок 1 – Простые циклы

2.7 Передача данных с использованием последовательности циклов

Любая передача начинается с такта передачи адреса, задания типа циклов и выставления сигнала использования адреса **ADS#**. Если у процессора, имеется запрос на чтение более четырех байт, то он в следующем такте **T2** оставляет сигнал **BLAST#** пассивным. Пассивный сигнал **BLAST#** для устройства (памяти) в первом такте **T2** означает намерение процессора начать передачу данных более 4-х байт. Если эта передача является заполнение строки кэш-памяти, то процессор в такте **T1** проверяет сигнал разрешения кэширования **KEN#**. Сигнал **KEN#** повторяется в последнем такте передачи строки кэш-памяти.

В зависимости от конструкции устройства зависит выбор сигналов готовности: **READY#** или **BREADY#**.

Если устройство отвечает сигналом **BREADY#**, то передача данных производится в пакетном цикле, если – сигналом **READY#**, то формируется последовательность циклов.

Диаграмма передачи данных с использование последовательности циклов представлена на рис. 2.

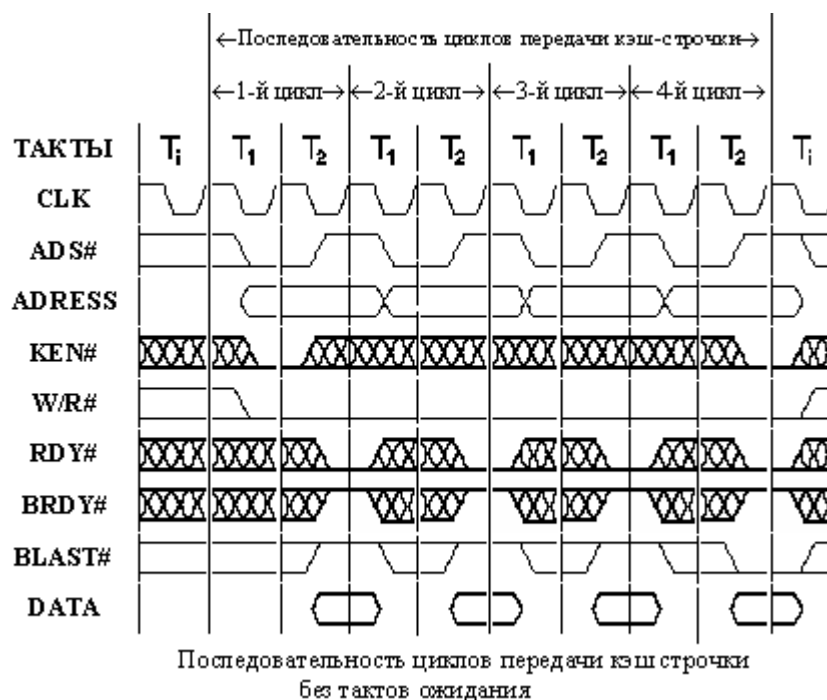


Рисунок 2 – Последовательность циклов

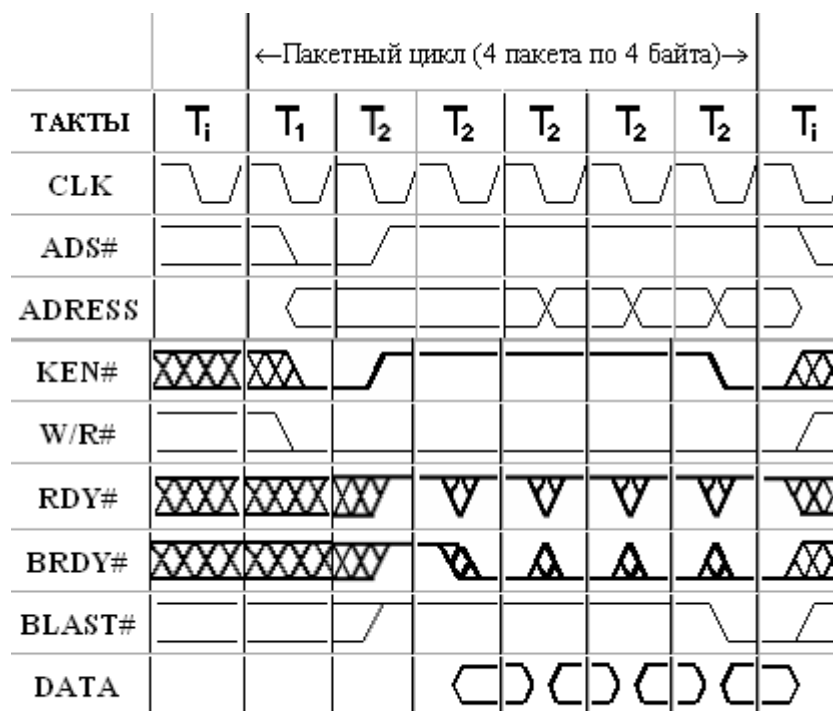
2.8 Передача данных с использованием пакетных циклов

Пакетный цикл является основным циклом передачи данных в локальном интерфейсе МП i486.

Пакетный цикл начинается, если в первом такте **T2** процессор не активировал **BLAST#** (указание о многотактной передаче), и устройство ответило сигналом готовности **BRDY#**.

В этом случае схема передачи выглядит так: передача адреса, передача до четырех байт данных.

Общее число байт данных, передаваемых в пакетном цикле по одному внутреннему запросу процессора, и их адреса должны не выходить за пределы одной строки кэш-памяти. На рис. 3 представлен один из типичных случаев передач пакетными циклами. Это наличие одного такта ожидания (такт **T2** после такта **T1**). Он обусловлен задержками дешифрации адреса и формировании данных (всей строки) в регистре данных или буфере строк в микросхемах памяти. Считывание следующих данных с регистра (в пределах кэш-строки) производится без задержек.



Пакетный цикл передачи кэш-строчки с одним тактом ожидания перед передачей данных.

Рисунок 3 – Пакетный цикл

В циклах локального интерфейса могут использоваться 5 типов тактов: **T_i, T₁, T₂, T_b и T_{1b}**.

В обычных циклах такты **T_b** и **T_{1b}** не используются. Их назначение – обслуживание специальных ситуаций. Это:

1. T_b – Второй и последующий такты прерванных циклов магистрали;
2. T_{1b} – Первый такт цикла перезапуска микропроцессора. На линиях адреса и состояния достоверные значения, сигнал **ADS#** активен.

В случае организации передачи данных можно ограничиться только тактами: **T_i, T₁ и T₂**

3 Модель функционирования локального интерфейса МП i486

На рис. 4 представлена упрощенная модель функционирования локального интерфейса МП i486 в виде графа состояний и переходов. Упрощение за-

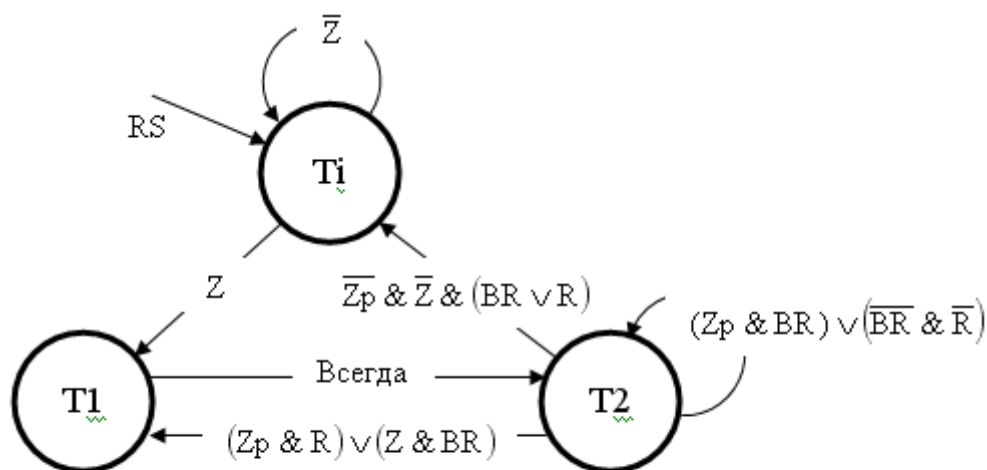
ключается в том, что в модели не рассматриваются такты Tb и N1b. То есть моделируется только взаимодействие задатчика с исполнителем без арбитража и смены задатчика.

В модели вершины графа соответствуют отдельным тактам возможных циклов, дуги – переходам. Вершины графа снабжены символами названия соответствующих тактов, переходы – условиями переходов.

Внутренний запрос процессора на передачу данных может потребовать один простой цикл, один пакетный цикл или одну последовательность циклов, отдельными циклами которой могут быть пакетными. В пакетном цикле и в одной последовательности циклов могут передаваться только данные одной строки кэш-памяти.

В связи с этим в модели внутренний запрос процессора на передачу данных различается на:

- 1) запрос передачи данных новой строки кэш-памяти – **Z**;
- 2) запрос на продолжение передачи данных той же строки кэш-памяти – **Zp**.



Граф состояний механизма пакетной передачи в локальном интерфейсе МП i486 (без учета внепроцессорных передач).

Рисунок 4 – Модель функционирования локального интерфейса МП i486

3.1 Описание тактов и условий переходов

3.1.1 Описание такта Ti

Шины интерфейса находится в пассивном состоянии. Сигналы на линиях адреса и сигналы статуса имеют неопределенные значения, три-стабильные выходы могут находиться в высокоимпедансном состоянии.

При включении или рестарте процессора на шинах интерфейса устанавливается такт **T_i**. Процессор находится в пассивном состоянии при отсутствии внутреннего запроса процессора на передачу данных.

При появлении внутреннего запроса на передачу данных производится переход на такт **T₁**.

Состояние подтверждения прерывания для процессора также является пассивным состоянием. Это тоже такт **T_i**, но в состояние захвата процессор активизирует идентифицирующий выходной сигнал **HLDA**.

3.1.2 Описание такта T₁

Первый такт цикла магистрали. На линиях адреса, идентификации цикла устанавливаются достоверные значения, сигнал **ADS#** активен.

После такта **T₁** всегда следует такт **T₂**.

3.1.3 Описание такта T₂

Второй и последующие такты цикла магистрали. Данные формируются в цикле записи или принимаются в цикле чтения. Проверяется активность сигналов **RDY#** и **BRDY#** и наличие внутренних запросов процессора **Z** и **Z_p**.

При одновременной пассивности сигналов **RDY#** и **BRDY#** или активности **BRDY#** и **Z_p** – такт **T₂** повторяется.

При активных сигналах **Z_p** и **RDY#** или сигналах **Z** и **BRDY#** – такт **T₂** сменяется тактом **T₁**.

При одновременной пассивности сигналов **Z_p** и **Z**, но активности одного из сигналов готовности **RDY#** или **BRDY#** – такт **T₂** сменяется тактом **T_i**.

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ»
(КНИТУ-КАИ)
Чистопольский филиал «Восток»

Кафедра компьютерных и телекоммуникационных систем

Отчет

по лабораторной работе № 1

по дисциплине «Микропроцессорные системы»

Команды пересылки данных микропроцессора Intel:

память → регистр;

регистр → память

Выполнил

ст. группы 21402 Петров И.И.

Проверил

к. т. н., доцент Белош В.В.

г. Чистополь

2023 г.

Отчет

по лабораторной работе № 2

по дисциплине «Микропроцессорные системы»

Команды пересылки данных микропроцессора Intel:

данные в память;

данные в регистр

Отчет

по лабораторной работе № 3

по дисциплине «Микропроцессорные системы»

Команды пересылки данных микропроцессора Intel:

из аккумулятора в память

из памяти в аккумулятор

Отчет

по лабораторной работе № 4

по дисциплине «Микропроцессорные системы»

Команды пересылки данных микропроцессора Intel:

из памяти в сегментный регистр

из сегментного регистра в аккумулятор

Отчет

по лабораторной работе № 5

по дисциплине «Микропроцессорные системы»

Локальный интерфейс Intel 80386:

простой цикл

простой цикл с ожиданием

Отчет

по лабораторной работе № 6

по дисциплине «Микропроцессорные системы»

Локальный интерфейс Intel 80386:

конвейерный цикл

Отчет

по лабораторной работе № 7

по дисциплине «Микропроцессорные системы»

Локальный интерфейс Intel 80486:

простой цикл

Отчет

по лабораторной работе № 8

по дисциплине «Микропроцессорные системы»

Локальный интерфейс Intel 80486:

последовательность циклов

пакетный цикл