Содержание

Структура лабораторной установки

В состав лабораторной установки входит центральный процессор, оперативная память и набор устройств ввода-вывода, подключаемых к общей шине микропроцессорной системы (рисунок 1).

В состав центрального процессора входят стандартные устройства, блоки и узлы, функции которых определяются заданной системой команд и режимами адресации для чисел с фиксированной запятой, система счета времени, аппаратура прерываний, а также устройство управления памятью (УУП), в состав которого входят два блока:

- блок преобразования ЛА в ФА;
- блок защиты памяти.

С выходов УУП формируются 18-ти разрядный ФА, сигнал сегментной ошибки по вектору 250 при обращении к ОП и сигнал ошибки переполнения сумматора при вычисления ФА с вектором 160.

Сигналы ошибок по векторам 250 и 160 поступают на входы аппаратуры прерываний и параллельно на блок управления ОП (для блокирования обращения к памяти при операциях чтения или записи в зависимости от установленных атрибутов защиты).

Для имитации обмена с УВВ по системной магистрали через контроллеры ввода–вывода подключается ряд ПУ, создавая минимальную конфигурацию микроЭВМ:

- средства диалога пользователя с ЭВМ (дисплей и клавиатура);
- жесткий магнитный диск;
- гибкий магнитный диск;
- принтер.

В систему команд процессора не входят команды ввода-вывода IN и OUT, а обращение к ПУ осуществляется по обычным машинным командам через общее адресное пространство логических адресов ввода-вывода с номерами со 160000 по 177776 в восьмиричной системе счисления.

Тогда для вывода информации из ячейки с логическим адресом $10122_{8сс}$ в порт вывода ПУ с адресом $177170_{8сс}$ необходимо выполнить команду:

MOV @#10122, @#177170, где @# мнемоника абсолютной (прямой) адресации.

На рисунке 2 показано распределение адресного пространства логических и физических адресов процессора.

Значения x, y, z, X, Y, Z начальных адресов сегментов определяются длиной разработанных системных и пользовательских программ и базовыми адресами сегментов, определяемых в вариантах заданий на выполнение лабораторного практикума.

Рисунок 2 - Распределение адресного пространства программ, данных и устройств ввода-вывода

Из рисунка 2 видно, что логические адреса процессора не совпадают с физическими, а их распределение выполняет ОС в зависимости от принадлежности программы: к системным или пользовательским и назначения сегмента (кода, данных, стека). Границы базовых адресов сегментов зависят от размера ранее загруженных сегментов по границе кратной 64 байт.

Адресное пространство ввода-вывода всегда занимает последний банк емкостью 8 Кбайт в адресном пространстве виртуальных адресов процессора.

В адресном пространстве можно выделить два стека для системного и пользовательского режимов, а также области памяти для хранения сегментов данных и кода для каждого из режимов работы.

Функции, выполняемые устройством управления памятью и системой внутренних векторов прерываний процессора

УУП процессора выполняет следующие функции:

♦ расширение емкости адресуемой памяти с 64 Кбайт до 256 Кбайт путем преобразования логического адреса в физический адрес;

♦ защиту пользовательских и системных программ друг от друга за счет реализации двух режимов работы: системного и пользовательского, то есть использования различных областей адресного пространства для режима пользователя и ОС (системный режим);

♦ раздельное хранение дескрипторов сегментов (базовых адресов и атрибутов защиты), как для системных, так и для пользовательских сегментов программ и данных;

♦ контроль за доступом к ОЗУ, то есть выработка вектора ошибки с номером 250 с различными кодами ошибок в зависимости от нарушения атрибутов:

◊ защита сегментов данных с возможностью установки следующих прав доступа: защита как по записи так и по чтению и защита только по записи (без указания типа сегмента или только для сегментов данных). Комбинация с защитой только по чтению является бессмысленной, поэтому не используется так как, во-первых, чтение операнда из ОП не нарушает достоверности данных, во-вторых, большинство арифметико-логических команд заканчивается записью результата, но предварительно операнд должен быть выбран из памяти;

◊ защита кодовых сегментов (устанавливается при настройке): запись запрещена по умолчанию, выполнение команды разрешено всегда, возможна установка запрета чтения из сегмента;

◊ защита сегментов кода и данных по выходу за границы разрешенного размера сегмента;

♦ фиксацию кода ошибки и логического адреса, на котором обнаружена ошибка, для дальнейшего анализа и классификации их в ППОП;

♦ контроль переполнения при формировании физического адреса, то есть его выхода за допустимое адресное пространство процессора (выработка вектора с номером 160).

Аппаратура, вырабатывающая внутренние вектора прерываний процессора, формирует следующие сигналы прерывания:

♦ обращение по нечетному адресу для команд работы со словами (вектор 4);

- несуществующая инструкция (команда) процессора (вектор 10);
- прерывание по биту трассировки (вектор 14);
- попытка обращения к неподключенному ВУ (вектор 120).

Схему УУП можно также разделить на две части, которые функционируют параллельно:

- устройство преобразования ЛА в ФА (схема формирования ФА);
- схему формирования сигнала сегментной ошибки по размеру и правам доступа (атрибутам защиты) (вектор 250) и вектора 160 - переполнения сумматора.

Устройство преобразования ЛА в ФА

Преобразование 16-разрядного ЛА (рисунок 3) в 18-разрядный ФА выполняется только при включенном УУП (ДП). Содержимое ЛА рассматривается УУП в виде трех полей:

- биты [15-13] определяют номер сегмента (регистра PADR), в котором хранятся базовый адрес сегмента, его размер в блоках и атрибуты прав доступа;
- в битах [12-6] размещается адрес блока в сегменте относительно базового адреса;
- битах [5-0] указан адрес байта внутри блока.

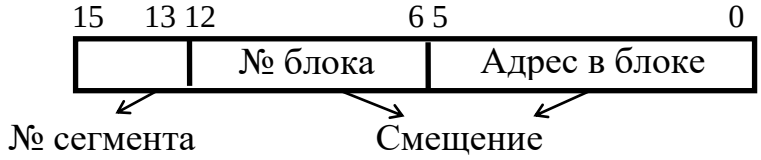


Рисунок 3 - Форматы логического адреса процессоров

Так как используется 16-разрядная шина интерфейса, то загрузка 32-разрядного дескриптора сегмента PADR состоит из двух шагов: загрузки регистра базового адреса PAR и регистра прав доступа PDR.

Схема формирования ФА приведена на рисунке 4. УУП может работать в двух режимах: с выключенным УУП (преобразователем ЛА в ФА и системой защиты памяти) и с включенным. При выключенном УУП ЛА совпадает с ФА и поступает в RgФА в обход сумматора. При этом, если по команде осуществляется обращение к сегменту ввода-вывода (ЛА=160000_8 - 177776_8), то аппаратные

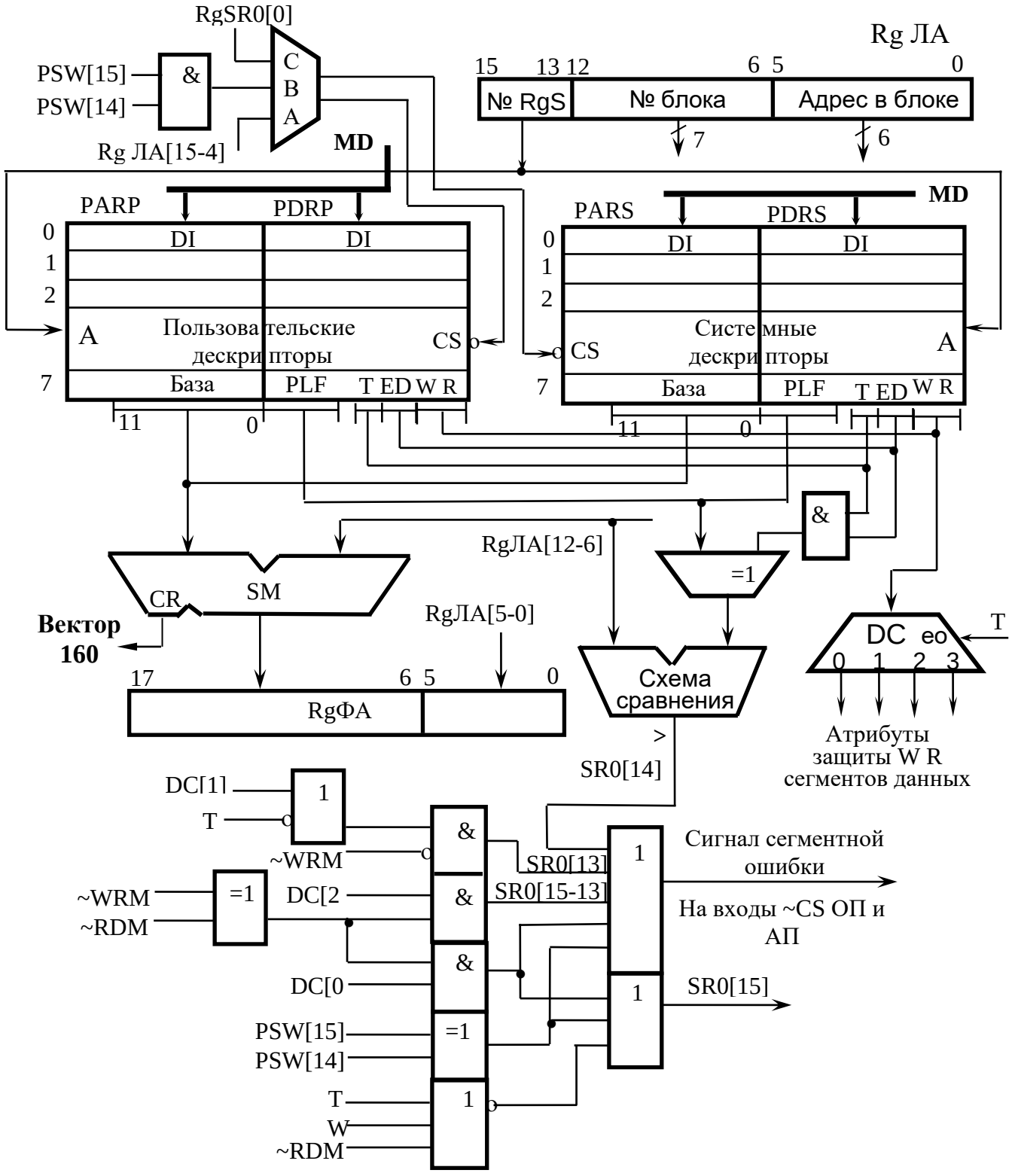


Рисунок 4 - Схема преобразования ЛА в ФА и формирования сигнала сегментной ошибки процессора

средства УУП формируют два (при 18-разрядном ФА) старших бита ФА равными единице, так как сегмент ввода-вывода содержит фиктивные адреса в адресном пространстве ЭВМ и всегда занимает последний сегмент (ввода-вывода) в адресном пространстве микроЭВМ (базовый адрес данного сегмента фиксирован и равен 7600_8). При выключенном УУП процессор адресует только 64 Кбайт памяти, а остальное адресное пространство может использоваться в качестве электронного диска. При включенном УУП базовый адрес сегмента размещается в дескрипторе PADR в одном из восьми регистров адреса PAR, а в одноименный регистр прав доступа PDR загружается информация, описывающая полномочия сегмента для организации защиты памяти и его размер.

В [1] показано, что при использовании двух режимов работы процессора (системного и пользовательского) существует два метода хранения дескрипторов сегментов:

- в общей памяти дескрипторов, а режим использования сегмента дополнительно указывается в дескрипторе (архитектура процессора Z8001);
- в раздельных памятях дескрипторов для каждого режима работы, что не требует дополнительной идентификации сегмента в его дескрипторе (архитектура процессора фирмы DEC).

При первом подходе емкость памяти дескрипторов должна быть достаточно большой, так как одновременно в память может быть загружено до нескольких десятков сегментов принадлежащих ОС и задачам пользователя. Второй подход с точки зрения выполнения лабораторной работы является более наглядным и понятным студентам и от него легко перейти к технической реализации первого метода.

Таким образом, УУП включает две группы регистров для системных и пользовательских сегментов данных и кода, что с одной стороны обеспечивает их защиту при неверном обращении к ним со стороны пользовательских программ или ОС, а с другой стороны вызвана малым размером сегментов (до 8 Кбайт) и небольшим числом дескрипторов (максимум 8).

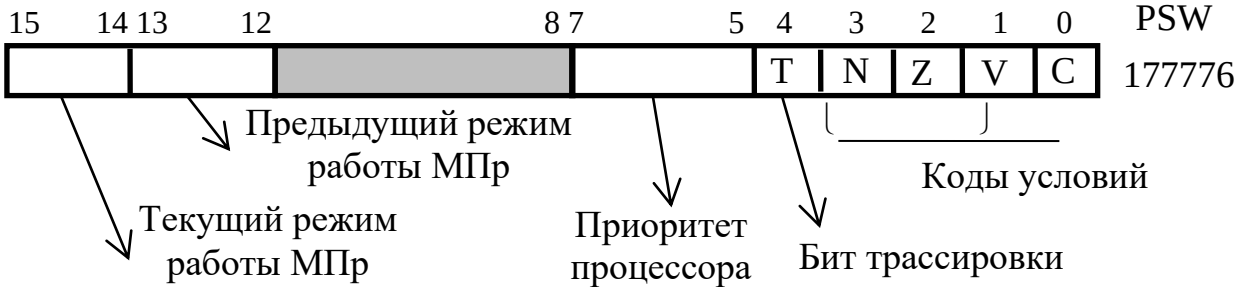
Обращение к пользовательским и системным регистрам-дескрипторам не выполняется одновременно поэтому, используя выходы с Z – состоянием, их можно объединить по монтажному ИЛИ. Для выбора группы дескрипторов используется сигнал разрешения выбора кристалла, который зависит от режима работы процессора (системный или пользовательский) и адресуемого простран-

ства регистров-дескрипторов (сигналы формируются с помощью селектора выбора адреса (СВА)) в общем адресном пространстве устройств ввода – вывода из двенадцати старших разрядов ЛА.

В таблице приведены логические адреса регистров-дескрипторов сегментов УУП.

Пользовательский режим			Системный режим		
№ регистра	PAR	PDR	№ регистра	PAR	PDR
0	177640	177600	0	172340	172300
1	177642	177602	1	172342	172302
2	177644	177604	2	172344	172304
3	177646	177606	3	172346	172306
4	177650	177610	4	172350	172310
5	177652	177612	5	172352	172312
6	177654	177614	6	172354	172314
7	177656	177616	7	172356	172316
SR0	177572				
SR2	177576				
PSW	177776				

Режим работы процессора задается в его слове состояния:



По значению битов PSW[15-14], в которых указывается режим работы процессора - системный - 00 или пользовательский - 11, определяется, какая из групп регистров (системная или пользовательская) в данный момент используется (рисунок 4). По номеру сегмента (ЛА[15-13]) выбирается соответствующий ему дескриптор, из которого 12-разрядный базовый адрес сегмента суммируется с номером блока в сегменте (ЛА[12-6]) и к полученной сумме операцией конкатенации присоединяются младшие 6 разрядов ЛА[5-0]. В результате получается 18-разрядный ФА.

Схема формирования сигнала сегментной ошибки

Рассмотрим организацию системы защиты памяти для исследуемого процессора. На рисунке 5 показан формат дескриптора сегмента. Регистр PDR содержит информацию для организации системы защиты памяти.

Поле ACF управляет доступом в память, т.е. устанавливает права доступа или атрибуты защиты (WP RP). Поле T указывает на тип сегмента: 1 - сегмент данных, 0 - сегмент кода. Для сегмента данных (бит типа сегмента T=1): 00 - сегмент не доступен ни для записи, ни для чтения; 01 - сегмент доступен только для чтения; 10 - не используется, так как является бессмысленным (запрет чтения с разрешением записи в сегмент) и 11 - сегмент доступен для чтения и для записи. Любой из этих атрибутов можно использовать только для сегментов данных, т.е. сегментов непосредственно данных и сегментов стека. Для сегментов кода (бит T=0) используется только два атрибута защиты, указываемые в бите WP поля ACF (бит D[2]=WP): 0 - разрешено только выполнение, 1 - допускается считывание из сегмента. Значение поля ACF загружается при инициализации системы и может изменяться ОС в процессе загрузки дескрипторов сегментов (свопинга сегментов). Нарушение прав доступа хотя бы по одному из установленных атрибутов защиты вызывает выработку вектора прерывания с номером 250.

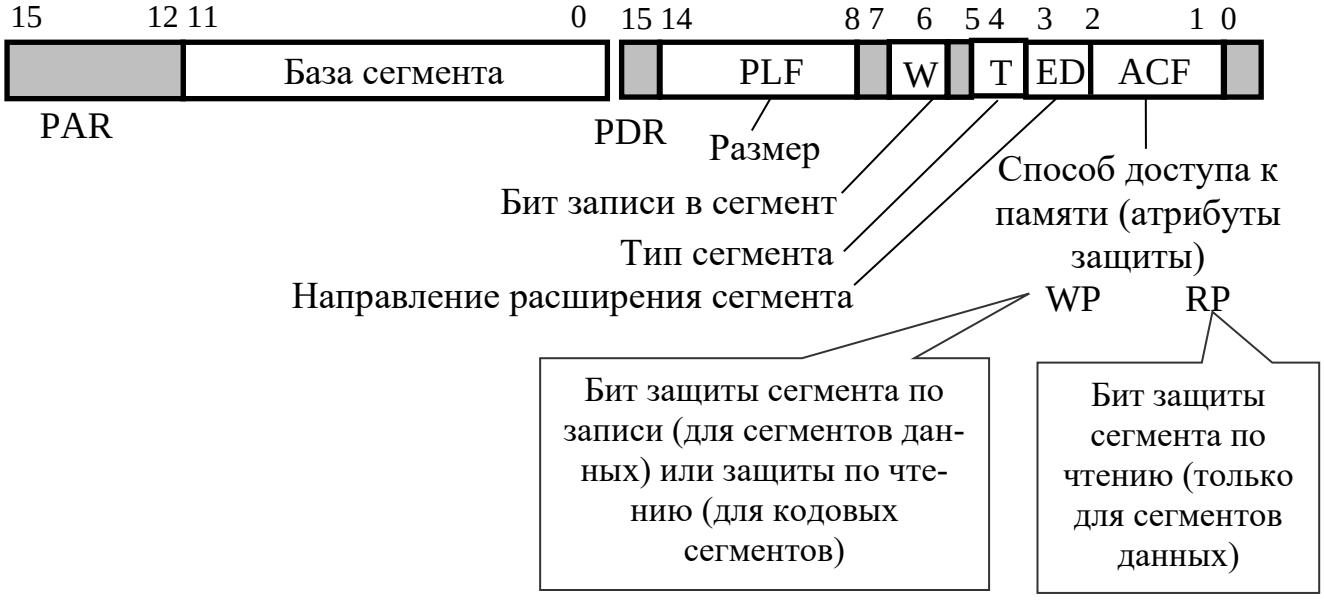


Рисунок 5 - Формат дескриптора сегмента PADR

Поле PLF указывает разрешенную длину сегмента в блоках. Максималь-

ная разрешенная длина сегмента 2^7=128 блоков.

Поле ED показывает в каком направлении расширяется сегмент данных. При ED=0 расширение сегмента происходит в сторону увеличения, а при ED=1 - в сторону уменьшения адресов, при этом содержимое поля PLF записывается в дополнительном коде (для ED=0 в прямом коде), т.е. бит ED выполняет функцию знакового разряда поля размера PLF сегмента данных и служит указателем сегмента стека. Для стекового сегмента (ED=1, T=1) PLF=0 соответствует максимальному размеру сегмента (128 блоков или 8 Кбайт), а для сегмента данных (ED=0) минимальному размеру в один блок (64 байта). Комбинация ED=1, T=0 для сегмента кода является запрещенной и никак не обрабатывается (значение бита ED не учитывается).

Поле W выполняет функцию контроля записи в сегмент и устанавливается в 1 только аппаратно, если в сегмент была произведена хотя бы одна запись. В дальнейшем значение данного бита используется при свопинге сегментов, находящихся на диске в режиме виртуальной адресации.

На рисунке 4 также приведена функциональная схема системы защиты памяти процессора. Параллельно с преобразованием ЛА в ФА выполняется сравнение размера сегмента из поля PLF дескриптора с номером текущего блока | ЛА[12-6] | > | PLF[14-8] | в зависимости от типа сегмента и значения бита ED, а также атрибутов защиты сегментов данных, декодируемых на дешифраторе для простоты понимания и сегментов кода, с сигналами ~RDM и ~WRM и сигналами состояния процессора, определяющими тип цикла шины и состояние процессора в текущий момент времени (на схеме сигналы состояния не показаны). Ниже приведены таблицы истинности формирования вектора прерывания с номером 250 для сегментов данных и кода по атрибутам защиты.

Таблица 1- Варианты атрибутов защиты и кодов ошибок по вектору 250 для сегментов данных и кода

PSW [15-14]	T	Атрибуты WP RP	Цикл шины ~WRM ~RDM	Код ошибки в SR0	Комментарии
0 1	X	X X	X X	SR0[15]=1	Несуществующий режим работы процессора
1 0	X	X X	X X	SR0[15]=1	
00 V 11	1	0 0	0 1	SR0[15]=1	Защита сегмента данных по чтению и по записи
00 V 11	1	0 0	1 0	SR0[15]=1	
00 V 11	1	0 1	0 1	SR0[13]=1	Защита сегмента данных по записи
00 V 11	1	1 0	0 1	SR0[15-13]=111	Несуществующий атрибут защиты сегмента данных
00 V 11	1	1 0	1 0	SR0[15-13]=111	
00 V 11	0	0 X	1 0	SR0[15]=1	Защита сегмента кода по чтению
00 V 11	0	X X	0 1	SR0[13]=1	Защита сегмента кода по записи (по умолчанию)

При нарушении размера сегмента или хотя бы одного атрибута защиты вырабатывается сигнал сегментной ошибки, доступ к памяти блокируется через вход ~CS выборки кристалла ОП, формируется вектор прерывания с номером 250 для обращения к таблице векторов прерываний IDT системного режима и переход на ППОП.

Таблица 2 - Варианты ошибок (вектор 250) по нарушению границ сегмента (размеру)

PSW [15-14]	T ED	Цикл шины ~WRM ~RDM	Код ошибки в SR0	Комментарии
00 V 11	1 0	X X	SR0[14]=1	PLF < Rg ЛА[12-6]
00 V 11	1 1	X X	SR0[14]=1	~PLF < Rg ЛА[12-6]
00 V 11	0 0	X X	SR0[14]=1	PLF < Rg ЛА[12-6]
00 V 11	0 1	X X		Запрещенная комбинация

В состав УУП также входят два регистра ошибок SR0 и SR2 (на схеме рисунка 4 не показаны), которые используются ОС для анализа кода ошибки и причины прерывания по защите памяти (в отличие от процессора Intel используется один вектор прерывания для всех причин нарушения доступа к памяти).

Регистр SR0 имеет ЛА 177572 на общей шине и указывает тип ошибки, состояние УУП и другую информацию на момент обнаружения ошибки, используемую в дальнейшем при выполнении ППОП для локализации места неисправности (рисунок 6).

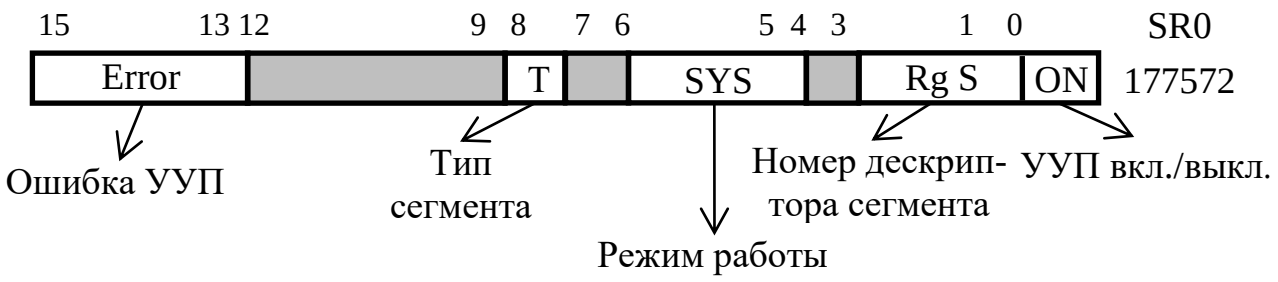


Рисунок 6 - Формат регистра ошибок SR0

Биты [15-13] SR0 указывают причину прерывания только для вектора 250, которые устанавливаются аппаратно (таблицы 3 и 4).

При обнаружении любой из перечисленных ошибок вырабатывается прерывание по вектору 250. В случае обнаружения нескольких одновременных ошибок ППОП обрабатывает их в порядке их приоритета: 15, 14, 13 биты SR0. В регистре SR0 также фиксируется режим работы процессора (биты [6-5]), тип сегмента (бит [8]) и номер сегмента (дескриптора) (биты [3-1]) на момент обнаружения ошибки. В бите [0] SR0 программно устанавливается бит включения/выключения УУП, определяющий емкость используемой ОП (64 Кбайт - реальный режим или 256 Кбайт - виртуальный режим работы процессора).

Таблица 3 – Установка бит кода ошибки в регистре SR0 для сегментов данных

Биты регистра ошибок	**Причина**
SR0[15]	Обращение к неактивному сегменту (ACF=00 - запрет по чтению и записи) или установлен недопустимый режим работы (PSW[15-14]=01 или 10) процессора
SR0[14]	Нарушение длины (размера) сегмента
SR0[13]	Попытка выполнить запись в сегмент, доступный только для чтения (ACF=01)
SR0[15-13]	В поле ACF дескриптора установлен недопустимый атрибут защиты (ACF=10 - защита по чтению с разрешением записи)

Таблица 4 – Установка бит кода ошибки в регистре SR0 для кодовых сегментов

Биты регистра ошибок	**Причина**
SR0[15]	Попытка чтения из сегмента с правом доступа только на выполнение или несуществующий режим работы процессора
SR0[14]	Нарушение длины (размера) сегмента
SR0[13]	Попытка выполнить запись в сегмент кода (запрет по умолчанию)

В регистре SR2 фиксируется ЛА, при обработке которого обнаружена ошибка, поэтому запись в регистры SR0 и SR2 производится только по сигналу сегментной ошибки вектора 250.

Процессор также контролирует значение базовых адресов сегментов и работу преобразователя ЛА в ФА путем формирования вектора прерывания ошибки 160 при возникновении единицы переноса из сумматора. Выработка такого

вектора возможна только при искажении или неверной загрузке базового адреса сегмента, превышающего значение 7600 в дескрипторе при обращении к УВВ (рисунок 4).

Защита ОС от обращений из пользовательского режима к системным устройствам ввода-вывода реализована следующим образом. Все адресное пространство логических адресов ввода-вывода (160000-177776) разделено на две равные части, из которых ЛА 160000-167776 можно использовать для подключения дополнительных периферийных и других устройств пользователя, а ЛА 170000-177776 принадлежат системным устройствам, к которым может обращаться только ОС, а пользователь только по команде системного программного прерывания EMT (только к разрешенным подпрограммам). Тогда при начальной инициализации ОС в регистры дескрипторы номера сегмента 7 и системного, и пользовательского режимов, принадлежащих сегменту ввода-вывода, загружает базовые адреса 7600. В поле размера пользовательского дескриптора загружается максимальный разрешенный размер сегмента равный 64 блока (0111111), а системного дескриптора - 128 блоков (1111111). Тогда попытка пользователя из своей программы напрямую обратиться к устройству (порту) с адресом, начиная со 170000, вызовет выработку вектора прерывания 250, а в SR0 будет установлен бит [14] - нарушение зашиты памяти по допустимому (разрешенному) размеру.

В таблице 5 представлен список векторов прерываний, реализованных в лабораторной установке. Вектор 4 вырабатывается при попытке доступа к памяти по нечетному адресу для словных команд, так как в данном процессоре слова должны быть выровнены по границе слова или при переполнении стека. Вектор 10 вырабатывается при попытке выполнения несуществующего кода команды, появляющегося вследствие случайных сбоев в процессоре, либо при выборке команды из памяти. При установке в PSW бита трассировки (пошаговый режим выполнения программы) вырабатывается вектор 14 без перехода на ППОП, а при попытке обращения к адресному пространству ввода-вывода с неиспользуемыми адресами портов ввода-вывода (неподключенному к системе устройству) вектор с номером 120.

Таблица 5 – Прерывания, реализованные в лабораторной установке

Вектор	Причина прерывания
4	Нечетный адрес для словной команды, переполнение стека
10	Нелегальные инструкции процессора
14	Внутреннее прерывание по биту трассировки Т слова состояния
120	Обращение к неподключенному внешнему устройству
160	Физический адрес внешнего устройства больше 777776_8
250	Ошибка обращения к сегменту по правам доступа

Выполнение лабораторной работы

Цель работы

Целью лабораторного практикума является:

- изучение архитектуры процессора и УУП (системы команд, режимов адресации, форматов слова состояния процессора, регистров ошибок, схем формирования ошибок и т.д.);
- изучение методов и средств преобразования логического адреса в физический, реализованных в УУП на примере одного из процессоров;
- изучение методов и средств защиты памяти, реализованных в УУП;
- получение практических навыков работы с УУП процессора на уровне ОС.

Выполнение лабораторной работы складывается из домашней подготовки, экспериментальных исследований на лабораторной установке и оформления отчета.

Домашняя подготовка

Домашняя подготовка включает повторение лекционного материала по теме: "Изучение принципов организации памяти: преобразование ЛА в ФА и средства защиты памяти" по литературным источникам [1-5] и методическим указаниям, знакомство с составом программного обеспечения лабораторного практикума и разработку следующих текстов программ:

- программ инициализации устройства управления памятью;
- подпрограмм обработки прерываний по векторам ошибок;
- текстов подпрограмм пользователя в соответствии с вариантом задания.

Экспериментальные исследования

Экспериментальные исследования включают в себя: ввод, отладку и вы-

полнение разработанных программ. После загрузки и запуска программы для лабораторных исследований на экран дисплея появляется окно для ввода сведений о студенте: Ф.И.О., номера группы и варианта задания (по номеру машины - первая подгруппа варианты с 1 по 10, вторая - с 11 по 20) для прохождения теста контроля знаний, а по завершении тестирования, выдачи результатов и записи их в журнал выводится экранная форма, изображенная на рисунке 7. Для повторной загрузки программы блок контроля можно обойти путем одновременного нажатия клавиш <Alt><F4> после ввода Ф.И.О. или установки флага "Пропустить входной тест-контроль". Экранная форма включает систему меню, входное окно и редактор для ввода команд в мнемонике языка Ассемблера (см. приложение 1), выходное окно, отображающее состояние регистров процессора и УУП.

Структурная схема лабораторной установки, функциональная схема УУП, система команд, виды адресаций, форматы регистров УУП, задание на выполне-

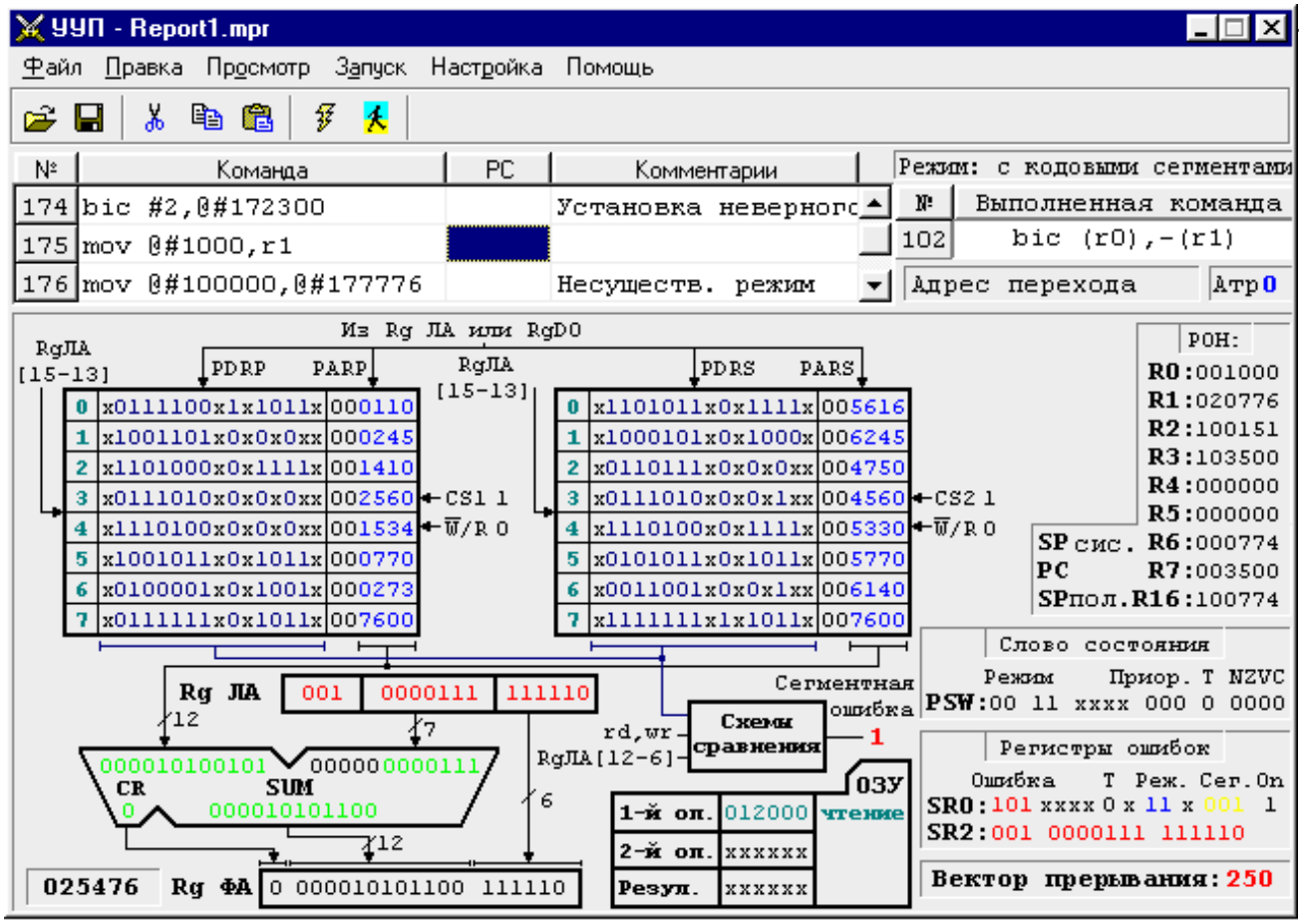


Рисунок 7 - Формат входного и выходного окна

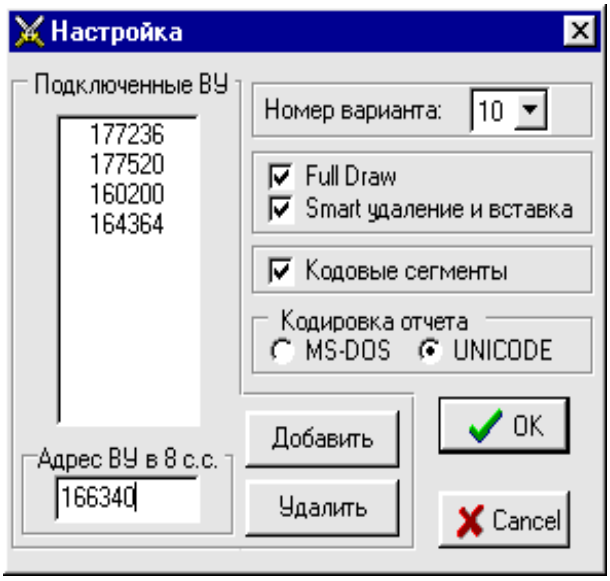
Отладку программ рекомендуется осуществлять в пошаговом режиме, а демонстрацию результатов - в автоматическом, с установкой точек останова в тех местах, где требуется изменять входную информацию.

Особенности системы команд и режимов адресации для исследуемого процессора (в качестве прототипа используется процессор фирмы DEC) приведены в приложении 1 и существенно отличаются от микропроцессоров семейства x86.

Методика исследований состоит из следующих этапов:

♦ открыть окно «Настройка», выбрать вариант задания, установить флажок «Кодовые сегменты», параметры кодировки отчета и другие настройки необходимые для проведения исследований. При установленном флажке "Кодовые сегменты" процессор работает с двумя типами сегментов: данных и кода. Формат регистров–дескрипторов в данном случае различается полями T и ACF (типом сегмента и атрибутами защиты). При отсутствии установки флажка процессор работает с сегментами без указания их типа (указатель типа сегмента T в дескрипторе игнорируется), а тип сегмента определяется атрибутами защиты;

♦ при необходимости подключить дополнительные внешние устройства к системе в окне настроек, т.е. в поле “Адрес ВУ в 8 с.с.” вводится адрес нового подключаемого внешнего устройства. При нажатии на кнопку "Добавить" происходит добавление в список подключенных дополнительного внешнего устройства, а его адрес отображается в списке "Подключенные ВУ". Для удаления одного из ВУ из списка его адрес необходимо выделить и нажать кнопку "Удалить";



♦ выполнить ввод текстов подпрограмм инициализации УУП, пользовательских и ППОП в соответствии с вариантом заданий;

♦ установить точки останова на необходимых командах;

♦ в меню <Просмотр> открыть окно <Таблица IDT> и ввести начальные адреса векторов прерываний, используемых пользователем. Все цифры в командах и таблице IDT должны вводится в восьме-

ричной системе счисления;

♦ выполнить инициализацию УУП путем загрузки регистров-дескрипторов и регистров базовых адресов в соответствии с вариантом задания;

♦ выполнить отладку разработанных программ и анализ работы УУП по преобразованию ЛА в ФА и средств защиты в соответствии с вариантом задания в пошаговом режиме;

♦ выполнить запуск программы в автоматическом режиме для получения окна результатов.

При обнаружении ошибок в программах и после их устранения необходимо привести лабораторную установку в исходное состояние путем нажатия клавиши <F9> или выбора подпункта "Сброс" в пункте меню "Запуск". При этом регистры процессора и УУП лабораторной установки принимают начальные состояния. После сброса выполнение программы начинается с команды с номером 0.

При получении вектора 160 (физический адрес внешнего устройства более 777776_8) каждый раз при обращении к сегменту ввода-вывода процессор будет выполнять некорректное обращение за пределы допустимого адресного пространства. Для выхода из этой ситуации необходимо в подпрограмме обработки прерываний вектора 160 или после ее выполнения нажать клавишу <F4> или в пункте меню "Запуск" выбрать подпункт "Мягкий сброс" после чего в процессоре автоматически восстанавливается некорректное значение регистра базового адреса в системном регистре дескриптора сегмента ввода-вывода (регистр PADR7) (для остальных регистров за восстановление базового адреса сегмента отвечает программист).

При получении вектора 250 для несуществующего режима работы необходимо принудительно восстановить значение режима работы в слове состояния PSW путем нажатия клавиши <F4> или в пункте меню "Запуск" выбрать подпункт "Мягкий сброс" сразу после завершения выполнения подпрограммы обработки прерываний вектора 250 (после выполнения команды RTI). Программно изменить содержимое регистра PSW невозможно, так как по команде RTI из стека восстанавливается старое значение PSW с указанием несуществующего режима работы и каждый раз будет возникать прерывание по несуществующему режиму работы.

Результатом работы является одноименное окно (рисунок 8), которое автоматически формируется при выработке векторов прерываний во время выпол-

нения разработанной программы.

Пример оформления текстов программ при домашней подготовке и рекомендуемая последовательность действий приведены в приложении 2.

Требования к отчету

Отчет по работе формируется автоматически из меню «Файл» в пункте «Генерация отчета» в соответствии с выполненными настройками сохраняется содержимое входного окна и окна результатов. Отчет по лабораторной работе должен содержать: титульный лист, задание, содержимое регистров-дескрипторов, таблицы IDT, листинг входного окна и окна результатов исследований с комментариями (с указанием причины прерывания), а также выводы.

Окно результатов

№ п/п	№ ком	№ вектора	Содержимое SR0					Логический адрес	Атри-буты	Команда	Размер	
			Ошибка	T	Реж	Сегм	On				PDR	ЛА
03	073	250	100	1	11	110	1	143500	00	add @#143500,@#350	041	035
04	074	120	xxx	x	xx	xxx	1	163500		add @#163500,@#350		
05	066	250	100	1	11	001	1	023500	00	add @#23500,@#350	115	035
06	067	250	010	1	11	010	1	043500	11	add @#43500,@#350	027	035
07	071	250	110	1	11	100	1	103500	00	add @#103500,@#350	013	035
08	074	120	xxx	x	xx	xxx	1	163500		add @#163500,@#350		
09	102	250	100	1	11	001	1	020776	00	bic (r0),-(r1)	115	007
10	105	250	001	1	11	011	1	060776	01	bic (r0),-(r1)	072	007
11	110	250	100	1	11	100	1	101000	00	bic (r0),-(r1)	013	010
12	113	120	xxx	x	xx	xxx	1	160776		bic (r0),-(r1)		

Рисунок 8 - Пример фрагмента листинга окна результатов

Задания на проведение лабораторного практикума

Задание 1:

1. В программе инициализации загрузить значения регистров базовых адресов и регистров прав доступа для системного режима в соответствии с вариантом задания, приведенным в таблице П4.1 приложения 4 или из файла заданий.

2. Включить устройство управления памятью.

3. Выполнить загрузку регистров базовых адресов и прав доступа для пользовательского режима в соответствии с вариантом задания, приведенным в таблице П4.2.

4. Установить пользовательский режим работы процессора.

5. Составить четыре подпрограммы, выполняющие следующие действия:

5.1. Нахождение суммы N-ых элементов сегментов данных с накоплением суммы в M-ой ячейке сегмента данных не имеющего установленных атрибутов защиты (разрешено чтение и запись). Режим работы процессора определяется выполняемой командой обращения к подпрограмме накопления суммы, заданной в п. 6. По первому и второму адресам абсолютная адресация (значения элементов N и M приведены в таблице П4.3);

5.2. Выполнение заданной в таблице П4.4 двухадресной команды с операндами из ячеек N и M (таблица 3), находящихся в сегментах со смежными номерами. Первый операнд команды (ячейка N) должен быть из сегмента с четным номером, а второй операнд (ячейка M) из сегмента с нечетным номером (то есть всего 4 команды с сочетанием операндов N-M из сегментов 0 – 1, 2 – 3, 4 – 5, 6 – 7). Виды адресаций к операндам приведены в таблице П4.4;

5.3. Выполнение одноадресных команд с заданной адресацией (таблица П4.5) для элементов с адресом N сегментов с 0 по 7 (в восьмеричной системе счисления).

6. Обратиться к разработанным в подпунктах 5.1, 5.2 и 5.3 подпрограммам по командам, приведенным в таблице: JSR, EMT, TRAP и JSR.

Вариант		1,6,11,16	2,7,12,17	3,8,13,18	4,9,14,19	5,10,15,20
Вызываемая подпрограмма	П.5.1	JSR	JSR	EMT	EMT	TRAP
	П.5.2	EMT	EMT	JSR	JSR	EMT
	П.5.3.а	TRAP	JSR	TRAP	JSR	JSR
	П.5.3.в	JSR	TRAP	JSR	TRAP	JSR

7. Выполнить изменение содержимого указанных в таблице П4.6 реги-

стров базовых адресов и прав доступа для пользовательского режима через вектор прерывания #К и повторить выполнение подпункта 5.3.а) или в) для одноадресной команды с записью (т.е. кроме команды TST).

Задание 2:

8. Составить и выполнить тестовые программы для выработки всех типов векторов прерываний для сегментов данных, реализованных в ЛУ (таблица 7), включая вектор 250 для заданных вариантов заполнения таблицы дескрипторов и базовых адресов для сочетаний причин прерываний, приведенных в таблице 8.

9. Составить и выполнить тестовые программы для выработки вектора 250 при обращении к кодовым сегментам для сочетаний причин прерываний, приведенных в таблице 9.

Примечания:

1 В подпрограммах обработки прерываний для векторов 4, 10, 120 и 160 в регистр R0 последовательно загрузить номер вектора прерывания и содержимое программного счетчика, сохраненного в стеке, на котором выработан сигнал прерывания. Для вектора 250 в ППОП в регистр R0 последовательно загрузить номер вектора, содержимое программного счетчика PC, содержимое регистра SR0, содержимое регистра SR2.

2 Регистры R1-R5 рекомендуется использовать при адресациях, использующих адрес, хранимый в РОН. Регистр R0 не рекомендуется использовать для этих целей, т.к. в ППОП его содержимое необходимо предварительно сохранять в стеке, а перед командой возврата - восстанавливать.

Приложение 1

Режимы адресации и система команд процессора

Режимы адресации, реализованные в процессоре

Мнемоника	Режим адресации	Пояснения	Пример
INC R4	Регистровая прямая	Операнд находится в РОН, указанном в команде	R4=100 Op=100
INC (R4)	Регистровая косвенная	Адрес операнда находится в РОН, указанном в команде	R4=100 Adr=100 Op=(100)
INC (R4)+	Прямая автоинкрементная (постинкрементная адресация)	Адрес операнда находится в РОН, указанном в команде. После выполнения команды содержимое регистра увеличивается на два при операциях над словами и на единицу при операциях над байтами	R4=100 Adr=100 Op=(100) R4'=102
INC -(R4)	Прямая автодекрементная (преддекрементная адресация)	До выполнения команды содержимое регистра уменьшается на два при операциях над словами и на единицу при операциях над байтами. Полученный результат является адресом операнда	R4=102 Adr=100 Op=(100) R4'=100
INC @(R4)+	Косвенная автоинкрементная (постинкрементная косвенная адресация)	В регистре, указанном в команде, находится адрес адреса операнда. После выполнения операции содержимое регистра всегда увеличивается на два, независимо от того, работает команда со словом или байтом	R4=100 Adr1=100 Adr=(100) Op=(Adr)= ((100)) R4'=102
INC @-(R4)	Косвенная автодекрементная (преддекрементная косвенная адресация)	В регистре, указанном в команде, до операции находится адрес адреса операнда, увеличенный на два. До выполнения операции содержимое регистра всегда уменьшается на два, независимо от того, работает команда со словом или байтом. Полученный результат является адресом адреса операнда	R4=102 Adr1=100 Adr=(100) Op=(Adr)= ((100)) R4'=100
INC 4(R4)	Прямая индексная	В регистре, указанном в команде, находится базовый адрес. До выполнения операции содержимое регистра суммируется с индексом, записанным перед РОН в команде. Полученная сумма является адресом операнда	R4=100 Adr=100+4=104 Op=(104)

Продолжение таблицы

INC @4(R4)	Косвенная индексная	Регистр, указанный в команде, содержит базовый адрес. Сумма индекса, записанного перед РОН в команде, и базового адреса определяет адрес адреса операнда	R4=100 Adr1=100+4=104 Adr=(104) Op=(Adr)=((104))
MOV #1234,R4	Непосредственная	Содержимое слова команды после знака # используется как операнд	Op=1234
INC @#400	Абсолютная	Содержимое слова команды после знака @# используется как адрес операнда	Adr=400 Op=(400)

где (100) - содержимое ячейки с адресом 100;

R4 - содержимое РОН с номером 4 до выполнения команды;

R4' - содержимое РОН с номером 4 после выполнения команды;

Adr - адрес операнда в ОП;

Adr1 - промежуточная ячейка ОП: адрес адреса операнда (адрес ячейки ОП, в которой хранится адрес операнда);

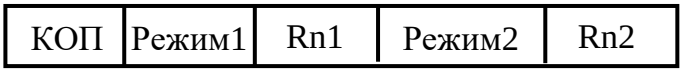
OP - значение операнда.

Процессор выполняет команды четырех базовых форматов: одноадресные, двухадресные, передачи управления и безадресные (команды управления).

Одноадресный формат:

Rn=DST=источник и приемник операнда

Двухадресный формат:



Rn1=SRC=источник операнда, Rn2=DST=приемник операнда

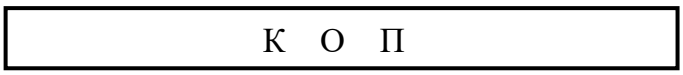
Передачи управления:

Для команд передачи управления используется мнемоническая запись в формате JNE #67, где знак # означает смещение в команде относительно программного счетчика PC, 67 - номер команды, на которую выполняется переход. В процессе трансляции программы номер команды преобразуется в смещение.

В командах программного прерывания по вектору указывается номер вектора прерывания в таблице IDT в формате: EMT #120, где 120 - номер вектора прерывния в системной таблице векторов прерываний IDT.

В командах JSR, JMP и SOB в адресной части команды указывается абсолютный адрес перехода: JMP @#124.

Безадресные:



Система команд процессора

Мнемо-ника	Название	Описание
	Двухадресные команды процессора	
MOV (B)	Пересылка	(DST):=(SRC)
CMP (B)	Сравнение	(SRC) - (DST)
ADD	Сложение	(DST):=(SRC) + (DST)
SUB	Вычитание	(DST):= (DST) - (SRC)
BIT (B)	Проверка бит	(SRC) Λ (DST)
BIC (B)	Очистка бит	(DST):=(SRC) Λ (DST)
BIS (B)	Установка бит	(DST):=(SRC) V (DST)
XOR	Исключающее ИЛИ	(DST):=(SRC) ⊕ (DST)
	Одноадресные команды процессора	
CLR (B)	Очистка	(DST) := 0
COM (B)	Инверсия	(DST):= ~(SRC)
INC (B)	Инкремент	(DST):=(SRC) + 1
DEC (B)	Декремент	(DST):=(SRC) - 1
NEG (B)	Смена знака (отрицание)	(DST):= - (SRC)
TST (B)	Проверка	(DST):=(SRC)
ASR (B)	Арифметический сдвиг вправо	(DST):=(SRC)/2
ASL (B)	Арифметический сдвиг влево	(DST):=(SRC) x 2
ROR (B)	Циклический сдвиг вправо	C:=(SRC[0]); (DST):=(C.SRC[7-1])
ROL (B)	Циклический сдвиг влево	C:=(SRC[7]); (DST):=(SRC[6-0].C)
SWAB	Перестановка байтов	(DST):=(SRC[7-0].[15-8])
ADC	Прибавление бита переноса	(DST):=(SRC) + C
SBC	Вычитание бита переноса	(DST):=(SRC) - C
SXT	Распространение знакового бита	(DST):=0, если N=0; (DST):= -1, если N=0
MFPS	Перенос младшего байта из слова состояния	(DST[7-0]):=PSW[7-0]
MTPS	Перенос младшего байта в слово состояния	PSW[7-0]:= (STC[7-0])
	Инструкции передачи управления	
BR	Ветвление (безусловное) BR #124	PC:=PC + 2 x Смещение
	С учетом знака	
BNE	Ветвление, если не равно нулю BNE #74	PC:=PC + 2 x Смещение, если Z=0

Продолжение таблицы

BEQ	Ветвление, если равно нулю	если Z=1
BPL	Ветвление, если плюс	если N=0
BMI	Ветвление, если минус	если N=1
BVC	Ветвление, если бит переполнения сброшен	если V=0
BVS	Ветвление, если бит переполнения установлен	если V=1
BCC	Ветвление, если бит переноса сброшен	если C=0
BCS	Ветвление, если бит переноса установлен	если C=1
BGE	Ветвление, если больше или равно нулю	если N ∀ V=0
BLT	Ветвление, если меньше нуля	если N ∀ V=1
BGT	Ветвление, если больше нуля	если [Z V (N ∀ V)]=0
BLE	Ветвление, если меньше или равно нулю	если [Z V (N ∀ V)]=1
	Без учета знака	
BHI	Ветвление, если выше	если [Z V (N ∀ V)]=1, Z=C=0
BLOS	Ветвление, если ниже или то же самое	если [Z V (N ∀ V)]=1, C=0
BHIS	Ветвление, если выше или то же самое	если [Z V (N ∀ V)]=1, C=0
BLO	Ветвление, если ниже	если [Z V (N ∀ V)]=1, C=1
JMP	Безусловный переход по абсолютному адресу	PC:=@#Adr
JSR	Переход к подпрограмме по абсолютному адресу	PC:=@#Adr, -(SP):=PC+2
RTS	Возврат из подпрограммы	PC:=(SP)+
SOB	Вычесть единицу и перейти (если не нуль) SOB Rn,@#Adr	Rn:=Rn-1, PC:=@#Adr, если Rn≠0
EMT	Командное прерывание для системных программ EMT #10	-(SP)s:=PSW; -(SP)s:=PC; PC:=(IDT)s
TRAP	Командное прерывание для пользовательских программ TRAP #34	-(SP)p:=PSW; -(SP)p:=PC; PC:=(IDT)p
RTI	Возврат из системного прерывания	PC:=(SP)s+; PSW:=(SP)s+
RTT	Возврат из пользовательского прерывания	PC:=(SP)p+;PSW:=(SP)p+
	Безадресные команды	
CLC	Очистить бит переноса	C:=0
CLV	Очистить бит переполнения	V:=0
CLZ	Очистить бит нулевого результата	Z:=0
CLN	Очистить бит отрицательного результата	N:=0
CCC	Очистить все биты условий	C=V=Z=N=0
SEC	Установить бит переноса	C:=1

Продолжение таблицы

SEV	Установить бит переполнения	V:=1
SEZ	Установить бит нулевого результата	Z:=1
SEN	Установить бит отрицательного результата	N:=1
SCC	Установить все биты условий	C=V=Z=N=1
NOP	Нет операции	

Приложение 2

Пример оформления листингов программ при домашней подготовке и последовательность выполнения экспериментальной части

Системный режим					Пользовательский				
№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	С	2016	53*		0	С	3016	34*	
1	Д	4562	101	З, Ч	1	К	3545	56	В
2	К	2760	14	В, Ч	2	С	6732	27*	
3	С	6230	27*		3	Д	4060	62	З
4	Д	5740	67		4	Д	5430	11	З, Ч
5	Д	1270	112	З	5	К	3070	23	В, Ч
6	К	4250	11	В	6	Д	5640	111	
7	Д	7600	177		7	Д	7600	77	

Исходные данные для исследований из таблиц 4.1 и 4.2 заданий:

Примечания:

В столбце тип сегмента:

С - сегмент стека;

Д - сегмент данных;

К - сегмент кода.

В столбце размер * означает расширение сегментов в сторону уменьшения адресов (сегмент стека С).

В столбце атрибутов защиты:

- З - атрибут защиты по записи;
- З, Ч - атрибуты защиты по записи и чтению;
- В - чтение из кодового сегмента запрещено (только выполнение);
- В, Ч - чтение из кодового сегмента разрешено.

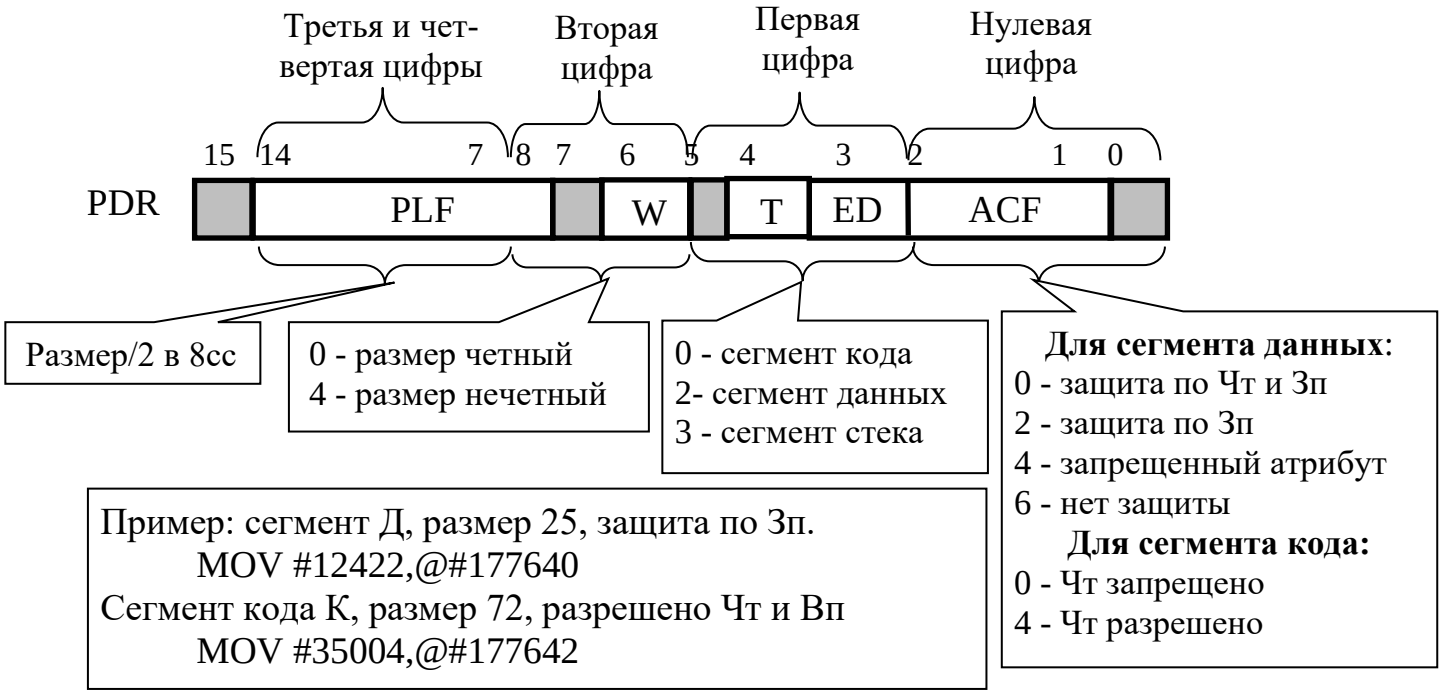
Пункт 1. Выполнить инициализацию регистров базовых адресов и прав доступа для системного режима работы.

```
MOV #2016,@#172340  ;  Загрузка регистров базовых адресов
MOV #4562,@#172342  ;  для системного режима
. . . . . .
MOV #7600,@#172356  ;

MOV #25436,@#172300;  Загрузка регистров прав доступа
MOV #40420,@#172302;  для системного режима
. . . . . .
MOV #77426,@#172316;
```

Рекомендации по кодированию полей регистра прав доступа:

Так как константа атрибутов защиты вводится в восьмиричной системе счисления, то рекомендуется следующая методика формирования цифр:



Пункт 2. Включить УУП:

```
MOV #1,@#177572        ; SR0:=1
```

Пункт 3. Выполнить инициализацию регистров базовых адресов и прав доступа для пользовательского режима работы.

```
MOV #3016,@#177640  ;  Загрузка регистров базовых адресов
MOV #3545,@#177642  ;  для пользовательского режима
. . . . . .
MOV #7600,@#177656  ;

MOV #16036,@#177600;  Загрузка регистров прав доступа
MOV #27000,@#177602;  для пользовательского режима
. . . . . .
MOV #37426,@#177616;
```

Пункт 4. Установить пользовательский режим работы процессора.

```
MOV #14000,@#177776        ; PSW:= 14XXXq, X - значение бит не
                           ; изменяется.
```

Пункт 5. Составить четыре подпрограммы (ПП) в соответствии с вариантом задания (значений адресов внутри сегментов N и M и видов адресаций).

Таблица 4.3 - Адреса операндов в сегменте

Вариант	N	M
21	1100	450

Таблица 4.4 – Вариант задания мнемоники и адресаций для двухадресной команды

Вариант	Команда	Адресация для операнда 1	Адресация для операнда 2
21	SUB	Индексная прямая	Автодекрементная прямая

Таблица 4.5 – Вариант задания мнемоник и адресаций для одноадресных команд

Вариант	Команда	Адресация	Команда	Адресация
21	TSTB	Абсолютная	CLR	Автодекрементная прямая

Примеры фрагментов подпрограмм.

В качестве сегмента для накопления суммы (п. 5.1) можно выбрать сегменты 0, 2 или 6, так все они являются сегментами данных для пользовательского режима работы без атрибутов защиты (обращение к сегменту по заданию по команде JSR из пользовательского режима). Выберем регистр PADR6.

ПП1: Нахождение суммы N-ых элементов сегментов:	ПП2: Вычитание элементов между соседними сегментами N и M:

CLR @#140450 ;
 	
ADD @#1100,@# 140450
ADD @#21100,@# 140450
. . . . . .
ADD @#161100,@# 140450
RTS
	MOV #1000,R1 ; Подготовка адресов
	MOV #20452,R2 ; операндов N0, M1
	SUB 100(R1),-(R2) ; Вычитание N0-M1
	. . . . .
	MOV #141000,R1 ; Подготовка адресов
	MOV #160452,R2 ; операндов N6, M7
	SUB 100(R1),-(R2) ; Вычитание N6-M7
	RTI ;

ПП3: Первая одноадресная команда:	ПП4: Вторая одноадресная команда:

TSTB @#1100 ;
TSTB @#21100 ;
TSTB @#41100 ;
. . . . . .
TSTB @#161100 ;
JMP ;
	MOV #1102,R1 ; Подготовка адреса
	CLR -(R1) ; операнда N0
	. . . . . .
	MOV #161102,R1 ; Подготовка адреса
	CLR -(R5) ; операнда N7
	RTT ;

Команда возврата из ПП зависит от варианта задания.

Из основной программы обратиться к разработанным ПП по командам:

JSR @#300

EMT #260

JSR @#320

TRAP #70

Команды JSR и TRAP не изменяют текущий режим работы процессора.

Пример одного из текстов ППОП:

```
MOV #4, R0      ; номер вектора в R0
MOV (R6),R0     ; адрес следующей команды, на которой выработано
                ; прерывание: PC в R0
RTI             ; возврат из прерывания
```

Пункт 7. Подпрограмму перезагрузки регистров дескрипторов предлагается выполнить самостоятельно и вновь вызвать заданную ПП по команде TRAP #70 (CLR).

Начальные адреса всех подпрограмм определяются разработчиком. После распределения адресного пространства для всех подпрограмм пп.1-7 необходимо подготовить таблицу прошивки IDT, т.е. для каждого вектора прерываний, используемого в программе, задать их начальные адреса, например:

Системный режим		Пользовательский режим	
Вектор	Адрес	Вектор	Адрес
4	120	70	200
10	130	72	220
120	140	120	250
160	150		
250	160		
260	170		

На этом домашняя подготовка заканчивается.

При лабораторных исследованиях необходимо ввести все тексты разработанных ПП инициализации, исследований пп. 5.1-5.3 и всех ППОП, заданных в таблице IDT. Открыть окно таблицы IDT в режиме "Просмотр" и ввести подготовленные начальные адреса ППОП в соответствии с номером вектора и режима работы процессора.

Установить начальные параметры лабораторной установки в окне настроек (п. "Экспериментальные исследования") и выполнить отладку разработанных программ в пошаговом режиме. В окне результатов будут фиксироваться и отображаться результаты выполнения программ только при выработке какого-либо вектора прерывания, что соответствует ведению журнала системных ошибок.

Рекомендации по самоконтролю за правильностью выполнения программ

Программы инициализации, все ППОП по векторам 4, 10, 120, 160, 250 и ПП перезагрузки регистров дескрипторов должны выполняться без выдачи сообщений об ошибках и векторов прерываний, так как они относятся к ОС и не содержат ошибок по доступу к сегментам.

При выполнении пп. 5, 6, 8 и 9 в поле вектор прерывания (рисунок 7) может появляться какой-либо номер вектора прерывания. На выполнении данной команды необходимо акцентировать внимание, выявить причину прерывания и ознакомиться с принципами работы системы защиты памяти, т.е. просмотреть содержимое ячейки ОЗУ и убедиться, что ее содержимое, например, при защите по записи в сегмент, не изменилось, либо блокировано выполнение командного цикла процессора, связанное с доступом к ОП по чтению и записи либо по другой причине прерывания. В исходном состоянии в ячейках ОП записан их адрес. Либо достаточно проанализировать информацию в поле <ОЗУ> в выходном окне ЛУ. Цифровое значение первого, второго операндов и результата означает выполнение доступа к ОП, а значение <xxxxxx> - блокирование обращения к памяти либо по чтению, либо по записи, либо по другой причине (вектору прерывания).

В пп. 8 и 9 предлагается составить две подпрограммы для сегментов данных и сегмента кода с выработкой всех возможных причин и кодов ошибок без изменения содержимого регистров дескрипторов, т.е. путем подбора типа команды, номера сегмента и логического адреса ячейки ОП, кроме выработки векторов 160 и 250 для несуществующих атрибута защиты сегмента данных и режима работы процессора. Для выработки этих векторов предварительно необходимо изменить содержимое базового адреса одного из дескрипторов (вектор 160), атрибут защиты сегмента данных (вектор 250 для несуществующего атрибута защиты) или режим работы процессора в PSW (вектор 250 для несуществующего режима работы процессора).

Приложение 3

Инструкция пользователя по управлению лабораторной установкой

Для копирования программы-модели на другую ПЭВМ в одном каталоге должны находится файлы, список которых приведен в таблице.

Таблица – Назначение файлов программы-модели

Имя файла	Назначение
DCM.exe	Исполняемый файл для запуска
DCM_que.dat, Test_dcm.dll, Testbase.tmp, DCM_tasks.zad, Hlp_dcm.dll	Файлы, необходимые для реализации блока входного контроля и вариантов заданий на лабораторную работу
DCM.cnt, DCM.hlp	Справочный файл и его содержание

Описание меню лабораторной установки

Пункт меню	Клавиши	Пояснения
Файл		
Создать		Создать новую программу
Открыть	Ctrl+O	Загрузить файл с программой
Сохранить	Ctrl+S	Сохранить программу
Сохранить как…		Сохранить программу под другим именем
Калькулятор	Alt+C	Вызвать встроенный калькулятор Windows
Генерация отчета		Создание отчета и сохранение его в текстовый файл
Выход	Alt+X	Выход из программы
Правка		
Вырезать	Ctrl+X	Копировать выделенный блок в буфер обмена с удалением
Копировать	Ctrl+Ins	Копировать выделенный блок в буфер обмена
Вставить	Shift+Ins	Вставить блок из буфера обмена
Вставить строку	Ctrl+N	Вставить пустую строку
Удалить строку	Ctrl+Y	Удалить строку
Размер окна	Ctrl+E	Распахнуть входное окно программ на весь экран
Просмотр		
Стека	Alt+S	Просмотр содержимого стека
Таблицы IDT	Alt+I	Просмотр и заполнение содержимого таблицы IDT
ОЗУ	Alt+O	Просмотр содержимого ОЗУ данных
Результатов	Alt+R	Просмотр окна результатов
Запуск		
Один шаг	F8	Пошаговое выполнение программы
Автоматически	Ctrl+F9	Запустить программу на автоматическое выполнение
Точка останова	Ctrl+F8	Установить точку останова на строку под курсором
Убрать точки останова		Удалить все точки останова из программы
Сброс	F9	Сбросить содержимое выходного окна и процессора
Отладка	F7	Выполнение команды без изменения счётчика команд
"Мягкий" сброс	F4	Сброс некорректных данных векторов 160 и 250 для несуществующего режима работы
Помощь		
Помощь	F1	Вызов справочной системы
Задание	Ctrl+F1	Просмотр выбранного варианта задания
Результаты теста	F5	Вывод на экран окна журнала тестирования

Запуск программы

Для запуска программы в ОС Windows необходимо дважды щелкнуть мышью на иконке DCM на рабочем столе Windows. Если иконка отсутствует, то нажать кнопку “Пуск”, выбрать в меню команду “Выполнить...” и в появившемся окне набрать при помощи клавиатуры полный путь и имя программы (DCM.exe) или найти ее с помощью кнопки “Обзор” после чего нажать кнопку “Ok”.

После загрузки и запуска программы для лабораторных исследований на экран дисплея появляется окно для ввода сведений о студенте: Ф.И.О., номера группы и варианта задания (по номеру машины - первая подгруппа номера с 1 по 10, вторая - с 11 по 20) для прохождения теста контроля знаний, а по завершении тестирования с выдачей результатов и записи в журнал выводится входная форма. При повторной загрузке программы блок контроля можно обойти путем одновременного нажатия клавиш <Alt><F4> после ввода Ф.И.О. или установки флага "Пропустить входной тест-контроль". Во время теста необходимо отмечать мышью правильные ответы. Для перемещения между ответами можно использовать клавиши <Tab>, а для отметки ответов – клавишу <Space> (<Пробел>). После ответа на 5 вопросов на экране монитора появляется форма с результатами тестирования. В случае успешной проверки знаний, после нажатия кнопки "Ок", на экране появится входная форма для исследования устройства управления памятью (рисунок 7). В верхней строке отображаются пункты меню: "Файл", "Правка", "Просмотр", "Запуск", "Настройка" и "Помощь".

Если разработанная студентом программа была записана на диск, то можно вызвать меню "Файл" клавишей <F10> или мышью и выбрать режим "Открыть Ctrl+O" или нажать комбинацию клавиш <Ctrl+O>. На экране монитора появится стандартное окно для загрузки программ. Выбрать требуемый файл с расширением *.mpr и нажать <Enter>. Загруженная программа отобразится на экране во входном окне программ.

По команде "Калькулятор Alt+C" вызывается встроенный калькулятор среды Windows, в котором можно выполнить необходимые расчеты, а также перевод из одной системы счисления в другую. Причем, если в системе уже запущен калькулятор, то произойдет переключение на него, а не запуск новой копии.

Вызов системы помощи

Для ознакомления с системой помощи необходимо выбрать в пункте меню "Помощь" подпункт "Помощь F1", после чего на экране монитора появится стандартное окно с системой помощи.

В этом же пункте меню при выборе подпункта "Задание Ctrl + F1" на экран монитора появится окно с индивидуальным заданием на выполнение лабораторной работы, в зависимости от установленного варианта.

С результатами пройденных тестов можно ознакомиться вызовом подпункта "Результаты теста F5".

Настройка лабораторной установки

Для указания параметров лабораторной установки служит пункт меню "Настройки" (см. выше). В поле "Номер варианта" вводится вариант (с 1 по 10 – первая подгруппа, с 11 по 20 – вторая подгруппа). При установленном флажке "Кодовые сегменты" лабораторная установка работает с двумя типами сегментов: данных и кода.

Формат регистров–дескрипторов в данном случае различается полями T и ACF (типом сегмента и атрибутами защиты). При отсутствии установки флажка процессор работает только с сегментами данных (указатель типа сегмента T в дескрипторе игнорируется).

Степень прорисовки выходных данных определяется флажком "Full draw". Если снять этот флажок, то при запуске программы в автоматическом режиме поле программ и содержимое регистров перерисовываться не будет, что позволяет увеличить быстродействие выполнения пользовательской программы.

Формат вставки строк определяет флажок "Smart удаление и вставка". Если его установить, то при редактировании, удаление и вставка строк будут действовать до первой пустой строки, иначе - на весь текст программы.

Кодировка отчета задается двумя кнопками в соответствующей панели. При установке кнопки "MS – DOS" отчет будет записан в кодировке DOS, а при установке кнопки "UNICODE" в кодировке UNICODE для Windows.

В лабораторной установке предусмотрена возможность подключения дополнительных внешних устройств. В поле “Адрес ВУ в 8 с.с.” вводится адрес нового подключаемого внешнего устройства. При нажатии на кнопку "Добавить" происходит добавление в список подключенных дополнительного внешнего устройства, а его адрес отображается в списке "Подключенные ВУ". Для удаления одного из ВУ из списка его адрес необходимо выделить и нажать кнопку "Удалить".

После настройки лабораторной установки необходимо нажать клавишу <Enter> или кнопку <ОК>.

Работа в режиме ввода и редактирования программ

Ввод программ осуществляется нажатием алфавитно-цифровых клавиш. В поле команд вводится инструкция на языке Ассемблера. Все числа в инструкциях должны вводится в восьмеричной системе счисления. В поле комментариев информацию можно вводить в символьном виде. Перемещение между полями осуществляется клавишами курсора и клавишей <Tab>. Для быстрого перехода на требуемую ячейку необходимо, удерживая клавишу <Ctrl>, набрать номер строки в восьмеричной системе счисления.

При вводе и редактировании предусмотрены следующие режимы:

- удаление строки со сжатием текста программы по клавишам <Ctrl><Y>;
- вставка строки с раздвижкой и перенумерацией строк внутри сплошного текста программы по клавишам <Ctrl><N>;
- копирование выделенного фрагмента программ в буфер обмена по клавишам <Ctrl><C>;
- вставка в любое место фрагмента программы из буфера обмена с наложением поверх существующей информации по клавишам <Ctrl><V>;
- очистка выделенного фрагмента по нажатию клавиш <Ctrl><X>;
- распахивание и свертка окна редактора программ по клавишам <Ctrl><E>.

Такие комбинации режимов редактирования позволяют выполнить любой вид перемещения или копирования текста за один или два этапа перемещения его фрагментов.

Для выделения фрагмента текста необходимо установить указатель на первую ячейку требуемого блока и удерживая клавишу <Shift> выделить с помощью клавиш

управления курсором блок необходимого размера. Выделение блока можно также осуществить с помощью мыши. Для этого надо щелкнуть по первой ячейке блока и, не отпуская левую клавишу мыши, растянуть выделение на весь необходимый блок. Для копирования выделенного фрагмента в буфер обмена, нажать комбинацию клавиш <Ctrl><C> или правую клавишу мыши и из контекстного меню выбрать команду "Копировать Ctrl + C". Затем переместить курсор на первую ячейку, куда необходимо вставить скопированный блок и нажать комбинацию клавиш <Ctrl><V> или правую клавишу мыши и из контекстного меню выбрать команду "Вставить Ctrl + V". В результате, скопированная информация заменит предыдущую.

Редактирование программ можно проводить через систему меню, для чего в верхней ее строке меню в пункте "Правка" выбрать требуемый подпункт. Кроме того, на панели инструментов ниже меню, расположены соответствующие пиктограммы для тех же целей.

При возникновении сложностей работы с управлением лабораторной установкой можно вызвать систему помощи нажатием клавиши <F1> или путем выбора в меню пункта "Помощь" подпункта "Помощь F1".

Отладка и выполнение программ

После ввода программы выполняется ее отладка. Для выполнения и отладки программ предусмотрены следующие режимы:

♦ выполнение текущей команды без автоматического перехода к следующей команде (клавиша <F7>);

♦ выполнение текущей команды с автоматическим переходом к следующей команде с номером большим на единицу или адрес которой указан в командах перехода, прерываний и так далее (клавиша <F8>);

♦ автоматическое выполнение последовательности команд до точки останова, устанавливаемой пользователем в программе, пустой строки в поле команд или до возникновения ситуации, вызывающей предупреждение пользователя (комбинация клавиш <Ctrl><F9>);

♦ установка точек останова в программе. Выполняется в контрольных точках для удобства отладки программ и демонстрации промежуточных результатов исследований или изменения входной информации в полях команды при многократном выполнении участков программ. Установка точек останова выполняется подведением указателя к строке, на которой необходимо остановить вычислительный процесс и нажатием клавиш <Ctrl><F8>, либо двойным щелчком мыши на номере этой строки, либо через меню <Запуск>. Точка останова выделяется красным цветом на экране в поле номера команды. Ее снятие осуществляется аналогично установке. Для снятия всех точек останова в программе необходимо выбрать в пункте меню "Запуск" соответствующий подпункт.

Все перечисленные действия можно выполнить через меню системы, для чего необходимо в верхней строке выбрать пункт "Запуск" через клавишу <F10> и далее соответствующий подпункт меню. Кроме того, для этого можно воспользоваться нажатием левой клавиши мыши на соответствующие пиктограммы на панели инструментов.

При обнаружении ошибок в программах и после их устранения необходимо привести лабораторную установку в исходное состояние путем нажатия клавиши <F9>

или выбора подпункта "Сброс" в пункте меню "Запуск". При этом регистры процессора и УУП лабораторной установки принимают начальные состояния. После сброса выполнение программы начинается с команды с номером 0.

При формировании вектора 160 (физический адрес внешнего устройства более 777776_8) каждый раз при обращении к сегменту ввода-вывода процессор будет выполнять некорректное обращение за пределы допустимого адресного пространства. Для выхода из этой ситуации необходимо в подпрограмме обработки прерываний вектора 160 или после ее выполнения нажать клавишу <F4> или в пункте меню "Запуск" выбрать подпункт "Мягкий сброс" после чего в процессоре автоматически восстанавливается некорректное значение регистра базового адреса в системном регистре дескриптора сегмента ввода-вывода (регистр PADR7). В остальных регистрах-дескрипторах старое значение базового адреса при необходимости может выполнить программист.

При формировании вектора 250 для несуществующего режима работы необходимо принудительно восстановить значение режима работы в слове состояния PSW путем нажатия клавиши <F4> или в пункте меню "Запуск" выбрать подпункт "Мягкий сброс" **сразу после завершения выполнения подпрограммы обработки прерываний вектора 250 (после выполнения команды RTI).** Программно изменить содержимое регистра PSW невозможно, так как по команде RTI из стека восстанавливается старое значение PSW с указанием несуществующего режима работы и каждый раз будет возникать прерывание по несуществующему режиму работы.

Разработанная программа может быть записана на диск. Для этого необходимо нажать комбинацию клавиш <Ctrl><S> или в меню "Файл" выбрать подпункт "Сохранить Ctrl + S". На экране монитора появится окно записи файла, если он до этого еще не был сохранен или произойдет перезапись файла с текущим именем. В поле "Имя файла" вводится имя сохраняемого файла, в случае первой записи его на диск. Если файл с таким именем уже существует, то будет выведен запрос на подтверждение перезаписи файла.

Работа с окнами

В процессе работы с лабораторной установкой иногда необходимо наблюдать на экране содержимое таблицы IDT, ОЗУ, стека и результатов исследований. Для этого служат дополнительные окна, которые вызываются из пункта меню "Просмотр" выбором соответствующего подпункта. Окно IDT предназначено не только для просмотра, но и для ввода и редактирования данных (начальных номеров команд ППОП). Перемещение в окнах просмотра производится с помощью клавиш курсора, <Page Up> и <Page Down>. Окна можно закрыть нажатием клавиш <Esc>, <Alt><F4>, мышью или с помощью системного меню.

Формирование отчета

Текст разработанной программы вместе с дескрипторами, таблицей IDT и окном результатов могут быть записаны в текстовый файл, а в дальнейшем дополнены комментариями и распечатаны для отчета. Для этого в меню "Файл" необходимо выбрать подпункт "Генерация отчета Alt+P" или просто нажать комбинацию клавиш <Alt><P>, и на экране откроется форма запроса имени файла для записи на диск в формате MS – DOS или UNICODE, в зависимости от настроек лабораторной установки (см. рисунок 8). После ввода имени файла или выбора его из списка текстовых файлов и нажатия

клавиши <Enter> программа вместе с результатами исследований будет записана на диск.

Назначение клавиш

В таблице приводится список функциональных клавиш, используемых для управления лабораторной установкой и их назначение.

Таблица – Назначение функциональных клавиш в программе

Клавиши	Назначение клавиш
<F1>	Вызов системы помощи
<Ctrl-F1>	Вызов задания на лабораторную работу
<F10>	Перевод курсора в верхнюю строку меню
<Tab>	Переход между полями в редакторе и окне настроек
<Пробел>	Установка элементов управления
<Esc>	Отмена в системе меню или закрытие активного окна
<Alt><C>	Вызов калькулятора
<Alt><F4>	Закрытие окна
<Alt><X>	Выход из программы
<Вниз>	На строку вниз
<Вверх>	На строку вверх
<Вправо>	На позицию вправо
<Влево>	На позицию влево
<Page Up>	На экран вверх
<Page Down>	На экран вниз
<End>	В конец строки
<Ctrl><Page Up>	В первую строку экрана
<Ctrl><Page Down>	В последнюю строку экрана
<Ctrl><Home>	В начало текста
<Ctrl><End>	В конец текста
<Ctrl>+номер	Быстрый переход на нужную строку

Работа с мышью

Используя мышь при работе с программой можно реализовать все функции, описанные ранее.

Для более удобной работы, наиболее часто используемые пункты меню продублированы кнопками быстрого запуска на панели инструментов с соответствующими пиктограммами. Вместо выбора подпунктов меню можно левой клавишей мыши нажимать на данные кнопки.

В правой стороне окон находятся полосы скроллинга. При нажатии левой кнопки без перемещения мыши осуществляется последовательная прокрутка информации в окне, а при движении мыши – перемещение указателя скроллинга.

Приложение 4

Варианты заданий на лабораторный практикум

Таблица П4.1 – Варианты заданий базовых адресов и атрибутов защиты системных сегментов

Вариант 1					Вариант 2					Вариант 3					Вариант 4				
№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	С	5016	33*		0	С	6016	24*		0	К	6716	37	В	0	С	4016	14*	
1	К	4432	101	В	1	К	6545	36	В, Ч	1	С	6432	57*		1	К	5245	47	В, Ч
2	Д	5560	24	З, Ч	2	С	6732	57*		2	Д	5560	62	З, Ч	2	Д	6432	57	З
3	С	5430	37*		3	Д	6060	72	З	3	Д	6330	11	З	3	Д	0430	72	З
4	Д	6040	57	З	4	Д	5430	11	З, Ч	4	С	6140	33*		4	С	6330	11*	
5	Д	6670	112	З	5	Д	5070	33	З	5	К	4770	101	В, Ч	5	К	4770	43	В
6	К	4250	11	В, Ч	6	К	4640	101	В	6	Д	6230	24	З	6	Д	6140	111	З, Ч
7	Д	7600	177		7	Д	7600	177		7	Д	7600	177		7	Д	7600	177	
Вариант 5					**Вариант 6**					**Вариант 7**					**Вариант 8**				
№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	С	5616	33*		0	С	6716	34*		0	С	6216	33*		0	С	6516	64*	
1	Д	5432	101	З, Ч	1	К	5760	57	В, Ч	1	К	5432	71	В	1	Д	5245	27	З
2	Д	4560	54	З	2	Д	5432	67	З	2	К	5560	144	В, Ч	2	Д	6432	47	З, Ч
3	С	4367	27*		3	Д	5560	112	З	3	С	6330	56*		3	К	4560	76	В, Ч
4	К	5140	57	В	4	Д	6330	14	З, Ч	4	Д	6140	77	З	4	Д	5100	12	З
5	Д	4770	62	З	5	С	4750	36*		5	Д	5600	72	З	5	С	5770	36*	
6	К	6010	13	В, Ч	6	К	5140	51	В	6	Д	5010	15	З, Ч	6	К	6140	51	В
7	Д	7600	177		7	Д	7600	177		7	Д	7600	177		7	Д	7600	177	

Примечания:

В столбце размер * означает расширение сегментов в сторону уменьшения адресов.

В столбце атрибутов защиты введены следующие обозначения:

- З - атрибут защиты по записи;
- З, Ч - атрибут защиты по записи и чтению.
- В - означает, что чтение из кодового сегмента запрещено.
- В, Ч означает, что чтение из кодового сегмента разрешено.

Продолжение таблицы П4.1

Вариант 9					Вариант 10					Вариант 11					Вариант 12				
№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	С	4716	63*		0	С	5616	54*		0	С	4250	73*		0	С	4640	64*	
1	Д	4432	111	З	1	Д	6245	105	З, Ч	1	Д	5016	101	З, Ч	1	К	6016	36	В, Ч
2	С	6560	34*		2	Д	4750	67	З	2	К	4432	24	В	2	С	6545	57*	
3	К	6330	57	В	3	Д	4560	72	З	3	С	5560	37*		3	Д	6732	72	З, Ч
4	Д	5140	77	З	4	С	5330	13*		4	Д	5430	57	З	4	Д	6060	11	З
5	Д	4770	112	З, Ч	5	К	5770	53	В, Ч	5	Д	6040	112	З	5	Д	5430	33	З
6	К	5560	11	В, Ч	6	К	6140	31	В	6	К	6670	11	В, Ч	6	К	5070	101	В
7	Д	7600	177		7	Д	7600	177		7	Д	7600	177		7	Д	7600	177	
Вариант 13					**Вариант 14**					**Вариант 15**					**Вариант 16**				
№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	Д	6230	37	З, Ч	0	С	4770	74*		0	С	6010	63*		0	С	5140	54*	
1	С	6716	57*		1	Д	4016	47	З	1	Д	5616	101	З	1	Д	6716	57	З
2	К	6432	62	В	2	К	5245	57	В	2	К	5432	54	В, Ч	2	Д	5760	67	З
3	Д	5560	11	З	3	К	6432	72	В, Ч	3	С	4560	57*		3	Д	5432	112	З, Ч
4	С	6330	33*		4	С	0430	11*		4	Д	4367	27	З	4	К	5560	14	В
5	Д	6140	101	З	5	Д	6330	43	З, Ч	5	К	5140	62	В	5	С	6330	36*	
6	К	4770	24	В, Ч	6	Д	6140	111	З	6	Д	4770	13	З, Ч	6	К	4750	51	В, Ч
7	Д	7600	177		7	Д	7600	177		7	Д	7600	177		7	Д	7600	177	
Вариант 17					**Вариант 18**					**Вариант 19**					**Вариант 20**				
№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	С	5010	53*		0	С	6140	64*		0	С	5560	73*		0	С	6140	64*	
1	Д	6216	71	З	1	Д	6516	27	З, Ч	1	К	4716	111	В	1	Д	5616	105	З
2	К	5432	144	В	2	К	5245	47	В	2	С	4432	34*		2	Д	6245	67	З
3	С	5560	16*		3	Д	6432	76	З	3	К	6560	57	В, Ч	3	К	4750	72	В, Ч
4	Д	6330	77	З	4	К	4560	12	В, Ч	4	Д	6330	77	З, Ч	4	С	4560	13*	
5	Д	6140	72	З, Ч	5	С	5100	36*		5	Д	5140	112	З	5	К	5330	53	В
6	К	5600	15	В, Ч	6	Д	5770	51	З	6	Д	4770	11	З	6	Д	5770	31	З, Ч
7	Д	7600	177		7	Д	7600	177		7	Д	7600	177		7	Д	7600	177	

Таблица П4.2 – Варианты заданий базовых адресов и атрибутов защиты пользовательских сегментов

Вариант 1

№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	К	1045	23	В
1	Д	2432	111	З
2	Д	1560	34	
3	С	3430	47*	
4	С	4000	67*	
5	К	1670	72	В, Ч
6	Д	3600	12	З, Ч
7	Д	7600	77	

Вариант 2

№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	К	0540	44	В, Ч
1	К	1545	56	В
2	Д	2732	67	
3	Д	3060	52	З
4	С	4000	11*	
5	Д	2070	23	З, Ч
6	С	3640	71*	
7	Д	7600	77	

Вариант 3

№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	С	2245	26*	
1	Д	2432	57	
2	Д	1560	42	З, Ч
3	Д	1330	14	З
4	С	4000	53*	
5	К	3770	121	В
6	К	3123	44	В, Ч
7	Д	7600	77	

Вариант 4

№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	Д	1210	14	
1	К	0245	55	В
2	Д	0432	67	З, Ч
3	К	1140	72	В, Ч
4	С	4000	11*	
5	Д	2770	73	З
6	С	3140	55*	
7	Д	7600	77	

Вариант 5

№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	Д	3245	63	
1	К	3432	121	В
2	Д	1560	44	З, Ч
3	С	1020	65*	
4	С	4000	47*	
5	Д	1770	32	З
6	К	2010	13	В, Ч
7	Д	7600	77	

Вариант 6

№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	С	1010	24*	
1	Д	0540	37	З
2	Д	1432	67	
3	Д	2560	112	З, Ч
4	С	4000	12*	
5	К	0750	34	В, Ч
6	К	3140	61	В
7	Д	7600	77	

Вариант 7

№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	С	2245	53*	
1	Д	3432	171	З
2	К	1560	34	В, Ч
3	Д	1330	67	З, Ч
4	С	4000	77*	
5	К	2523	42	В
6	Д	3010	15	
7	Д	7600	77	

Вариант 8

№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	К	1010	44	В
1	С	0245	77*	
2	Д	0432	15	
3	К	0560	66	В, Ч
4	С	4000	12*	
5	Д	1770	76	З
6	Д	2140	51	З, Ч
7	Д	7600	77	

Вариант 9

№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	К	1245	63	В, Ч
1	Д	0432	111	З
2	К	1560	44	В
3	С	0030	77*	
4	Д	4000	27	
5	Д	2770	52	З, Ч
6	С	3563	11*	
7	Д	7600	77	

Вариант 10

№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	К	0110	74	В
1	Д	0245	115	З, Ч
2	С	1410	27*	
3	Д	2560	72	З
4	С	4000	43*	
5	К	0770	113	В, Ч
6	Д	1140	11	
7	Д	7600	77	

Вариант 11

№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	К	3430	23	В, Ч
1	Д	1045	111	З
2	К	2432	34	В
3	С	1560	47*	
4	С	4000	67*	
5	Д	1670	72	З, Ч
6	Д	3600	12	
7	Д	7600	77	

Вариант 12

№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	К	3060	44	В
1	Д	0540	56	З, Ч
2	Д	1545	67	З
3	К	2732	52	В, Ч
4	С	4000	11*	
5	Д	2070	73	
6	С	3640	16*	
7	Д	7600	77	

Продолжение таблицы П4.2

Вариант 13					Вариант 14					Вариант 15					Вариант 16				
№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	С	1330	26*		0	Д	1140	14		0	Д	1020	63		0	С	2560	24*	
1	Д	2245	57	З, Ч	1	Д	1210	55	З	1	Д	3245	121	З, Ч	1	К	1010	37	В, Ч
2	К	2432	42	В, Ч	2	Д	0245	111	З, Ч	2	С	3432	44*		2	Д	0540	67	З, Ч
3	К	1560	14	В	3	К	0432	72	В, Ч	3	Д	1560	65	З	3	Д	1432	112	З
4	С	4000	53*		4	С	4000	11*		4	С	4000	47*		4	С	4000	12*	
5	Д	3770	121	З	5	К	2770	73	В	5	К	1770	12	В, Ч	5	К	0750	34	В
6	Д	3123	44		6	С	3140	65*		6	К	2010	43	В	6	Д	3140	61	
7	Д	7600	77		7	Д	7600	77		7	Д	7600	77		7	Д	7600	77	
Вариант 17					**Вариант 18**					**Вариант 19**					**Вариант 20**				
№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
0	С	1330	53*		0	К	0560	44	В	0	К	0030	63	В, Ч	0	Д	2560	74	
1	К	2245	171	В, Ч	1	Д	1010	27		1	Д	1245	111	З	1	К	0110	115	В
2	Д	3432	34	З, Ч	2	Д	0245	157	З, Ч	2	Д	0432	44		2	С	0245	27*	
3	К	1560	67	В	3	К	0432	66	В, Ч	3	С	1560	37*		3	К	1410	72	В, Ч
4	С	4000	77*		4	С	4000	12*		4	Д	4000	77	З, Ч	4	С	4000	13*	
5	Д	2523	42	З	5	Д	1770	76	З	5	К	2770	52	В	5	Д	0770	113	З
6	Д	3010	15		6	С	2140	51*		6	С	3563	11*		6	Д	1140	41	З, Ч
7	Д	7600	77		7	Д	7600	77		7	Д	7600	77		7	Д	7600	77	

Таблица П4.3 – Варианты заданий номеров используемых ячеек памяти

Вариант	N	M	Вариант	N	M
1	2000	500	11	3000	700
2	2050	600	12	3050	650
3	2100	700	13	3100	400
4	2200	750	14	3200	650
5	2400	550	15	3400	450
6	2450	650	16	4450	350
7	2500	400	17	4500	600
8	2700	450	18	4700	750
9	3000	300	19	5000	400
10	3500	350	20	5500	550

Таблица П4.4 – Варианты заданий мнемоник и адресаций для двухадресной команды

Вариант	Команда	Адресация первого операнда	Адресация второго операнда
1	MOVB	Индексная	Автодекрементная
2	BIC	Индексная	Регистровая косвенная
3	BISB	Регистровая косвенная	Абсолютная
4	SUB	Автоинкрементная	Индексная
5	MOV	Автодекрементная	Абсолютная
6	BICB	Абсолютная	Автоинкрементная
7	BIS	Автоинкрементная	Регистровая косвенная
8	XOR	Автодекрементная	Автоинкрементная
9	MOVB	Абсолютная	Индексная
10	BIC	Регистровая косвенная	Автодекрементная
11	BISB	Абсолютная	Регистровая косвенная
12	SUB	Автоинкрементная	Автодекрементная
13	MOV	Индексная	Автоинкрементная
14	BIC	Индексная	Абсолютная
15	BIS	Регистровая косвенная	Автоинкрементная
16	XOR	Автодекрементная	Индексная
17	MOVB	Абсолютная	Автодекрементная
18	BICB	Регистровая косвенная	Индексная
19	SUB	Автодекрементная	Регистровая косвенная
20	BIS	Автоинкрементная	Абсолютная

Таблица П4.5 – Варианты заданий мнемоник и адресации для одноадресных команд

Вариант	Команда	Адресация	Команда	Адресация
1	TSTB	Абсолютная	CLR	Автодекрементная
2	TST	Регистровая косвенная	COMB	Индексная
3	TSTB	Автоинкрементная	INC	Абсолютная
4	TST	Автодекрементная	DECB	Автоинкрементная
5	TSTB	Индексная	NEG	Регистровая косвенная
6	TST	Абсолютная	ASRB	Автодекрементная
7	TSTB	Регистровая косвенная	ASL	Индексная
8	TST	Автоинкрементная	RORB	Абсолютная
9	TSTB	Автодекрементная	ROL	Автоинкрементная
10	TST	Индексная	SWAB	Регистровая косвенная
11	TST	Индексная	ROLB	Регистровая косвенная
12	TSTB	Автодекрементная	SWAB	Абсолютная
13	TST	Автоинкрементная	NEGB	Автодекрементная
14	TSTB	Регистровая косвенная	ASR	Индексная
15	TST	Абсолютная	ASLB	Автоинкрементная

Продолжение таблицы П4.5

16	TSTB	Индексная	ROR	Регистровая косвенная
17	TST	Автодекрементная	INCB	Абсолютная
18	TSTB	Автоинкрементная	DEC	Автодекрементная
19	TST	Регистровая косвенная	CLRB	Индексная
20	TSTB	Абсолютная	COM	Автоинкрементная

Таблица П4.6 – Варианты заданий для изменения пользовательских дескрипторов для свопинга сегментов

№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты	№ сегмента	Тип сегмента	Базовый адрес	Размер	Атрибуты защиты
Вариант 1					**Вариант 2**					**Вариант 3**					**Вариант 4**				
0	Д	5604	10		6	Д	5073	27	З, Ч	1	Д	6140	37	З	4	К	4750	47	В
4	К	2210	66	В	5	К	1012	67	В	4	К	6700	42	В	0	Д	6310	51	
Вариант 5					**Вариант 6**					**Вариант 7**					**Вариант 8**				
4	К	6002	54	В, Ч	3	Д	5420	72	З	1	Д	5000	25	З, Ч	1	К	6140	54	В
6	К	4620	71	В	0	Д	5510	14	З, Ч	4	Д	5540	14	З	2	Д	5245	24	З
Вариант 9					**Вариант 10**					**Вариант 11**					**Вариант 12**				
1	Д	6560	7		5	К	6110	11	В, Ч	3	Д	4510	42	З, Ч	0	К	4640	37	В, Ч
3	Д	4430	77	З	2	К	5770	106	В	5	Д	4420	14		5	К	6016	35	В
Вариант 13					**Вариант 14**					**Вариант 15**					**Вариант 16**				
4	К	6610	54	В	1	К	4010	52	В	1	Д	4770	17	З, Ч	5	К	4750	42	В, Ч
3	Д	6430	72		4	Д	4512	61	З	4	К	5130	42	В	4	К	5560	37	В
Вариант 17					**Вариант 18**					**Вариант 19**					**Вариант 20**				
4	Д	6140	10	З	2	К	5245	37	В	3	К	4770	70	В	3	К	4560	11	В, Ч
5	К	6220	71	В	4	Д	5310	10	З, Ч	5	Д	5060	66	З	4	Д	4654	14	З

Таблица П4.7 – Варианты заданий на выработку векторов прерываний

Вектор	Причина прерывания
004	Нечетный адрес
010	Нелегальные или резервные инструкции процессора
014	Внутренне прерывание по биту трассировки Т слова состояния PSW
120	Обращение к неподключенному внешнему устройству
160	Физический адрес внешнего устройства больше 777776_8
250	Ошибка диспетчера памяти

Таблица П4.8 – Варианты заданий на выработку кодов ошибок для сегментов данных

Код ошибки в SR0	Причина прерывания
100	Защита по записи и чтению
010	Нарушение границ сегмента
001	Защита по записи
110	Защита по записи и чтению и нарушение границ сегмента
011	Защита по записи и нарушение границ сегмента
100	Несуществующий режим работы
111	Несуществующий атрибут защиты

Таблица П4.9 – Варианты заданий на выработку кодов ошибок для кодовых сегментов

Код ошибки в SR0	**Причина прерывания**
100	Защита по чтению из сегмента кода
010	Нарушение границ сегмента кода
001	Защита по записи в сегмент кода
110	Защита по чтению из сегмента кода и нарушение его границ
011	Защита по записи в сегмент кода и нарушение его границ
101	Защита по чтению из сегмента кода и по записи в него
111	Защита по чтению из сегмента кода, записи и нарушение его границ

Литература

1. Бакшаев А.М. Организация памяти ЭВМ. Учебное пособие. - Киров, ВятГТУ, 2000.- 140 с.

2. Микропроцессоры и микропроцессорные комплекты интегральных микросхем: Справочник: В 2 т./ Н.Н.Аверьянов, А.И.Березенко и др.; Под ред. В.А.Шахнова.- М.: Радио и связь, 1988.-Т.2.- 386с.

3. Г.Г. Кичев,Л.П. Некрасов. Архитектура и аппаратные средства миниЭВМ СМ-1600: Учеб. пособие для специалистов по эксплуатации аппаратных средств в области ВТ и АСУ. - М.: Машиностроение, 1988. - 440 с.

4. В.Лин. PDP-11 и VAX-11. Архитектура ЭВМ и программирование на языке Ассемблера: Пер. с англ. - М.: Радио и связь,1989. - 320 с.

5. МикроЭВМ: В 8 кн.: Практ. пособие/ Под ред. Л.Н. Преснухина. Кн.1. Семейство ЭВМ "Электроника-60". - М.: Высш. шк., 1988.- 172 с.