

Контрольные вопросы:

1. Особенности организации программ обмена данными с использованием прерываний. Сохранение и восстановление значений регистров.

Для реализации данного типа обмена необходимо заменить программный цикл ожидания готовности ВУ при асинхронной передаче (TSF, BR) аппаратной проверкой наличия внешнего прерывания, т. е. сигнала готовности по линии "Запрос прерывания" (см. рис. 3.1). В этом случае ЭВМ может выполнять любую программу (будем называть ее основной), а когда с ВУ поступит запрос прерывания (запрос на передачу или прием данных), приостановить (прервать) выполнение этой программы на время осуществления обмена данными. Все действия по приостановке программы и переходу к подпрограмме обмена реализуются в базовой ЭВМ с помощью контроллера прерываний, входящего в состав устройства управления.

Действия, выполняемые при этом процессором, как правило, те же, что и при обращении к подпрограмме. Только при обращении к подпрограмме они инициируются **командой**, а при обработке прерывания - **управляющим сигналом от ВУ**, который называют "Запрос на прерывание" или "Требование прерывания".

Эта важная особенность обмена с прерыванием программы позволяет организовать обмен данными с ВУ в произвольные моменты времени, не зависящие от программы, выполняемой в микроЭВМ. Таким образом, появляется возможность обмена данными с ВУ в реальном масштабе времени, определяемом внешней по отношению к микроЭВМ средой. Обмен с прерыванием программы существенным образом экономит время процессора, затрачиваемое на обмен. Это происходит за счет того, что исчезает необходимость в организации программных циклов ожидания готовности ВУ, на выполнение которых тратится значительное время, особенно при обмене с медленными ВУ.

Прерывание программы по требованию ВУ не должно оказывать на прерванную программу никакого влияния кроме увеличения времени ее выполнения за счет приостановки на время выполнения подпрограммы обработки прерывания. Поскольку для выполнения подпрограммы обработки прерывания используются различные регистры процессора (счетчик команд, регистр состояния и т.д.), то информацию, содержащуюся в них в момент прерывания, необходимо сохранить для последующего возврата в прерванную программу.

Обычно задача сохранения содержимого счетчика команд и регистра состояния процессора возлагается на **аппаратные средства** обработки прерывания. Сохранение содержимого других регистров процессора, используемых в подпрограмме обработки прерывания, производится непосредственно в подпрограмме. Предпочтительными с точки зрения повышения производительности микроЭВМ являются уменьшение числа команд, обеспечивающих сохранение информации о прерванной программе, и реализация этих функций аппаратными средствами. В нашем случае достаточно сохранить содержимое аккумулятора и регистра переноса С.

Формирование сигналов прерываний - запросов ВУ на обслуживание происходит в контроллерах соответствующих ВУ. В простейших случаях в качестве сигнала прерывания может использоваться сигнал "Готовность ВУ", поступающий из контроллера ВУ в КОНТРОЛЛЕР ПРЕРЫВАНИЯ Устройства управления УУ микроЭВМ. Для передачи в КОНТРОЛЛЕР ПРЕРЫВАНИЙ состояния флажков всех ВУ используется линия "Запрос прерывания", а для передачи в РЕГИСТР СОСТОЯНИЙ ВУ (Ф) состояния флажка опрашиваемого ВУ – линия "Состояние флага". Данные обычно передаются по одной шине данных, которая в базовой ЭВМ (для упрощения описания) представлена двумя шинами: шиной ввода и шиной вывода.

2. Команды работы разрешения/запрещения прерывания БЭВМ. Название, назначение и тип команды. Количество и название машинных циклов, потактовое выполнение команды.

Стр. 81-87 Архитектуры..

Тип команд – безадресные.

EI –FA00 Разрешение прерывания

После цикла выборки команды перешли на цикл Исполнение (МК):

8A	1000 0000	4800	EI	Разрешение прерывания
8B	8201 0008	8301		GOTO НАЧ(01)

GOTO НАЧ(01) – это переход на самую первую команду интерпретатора, т.е. на выборку следующей команды. **01: СК -> БР**

На 1-м такте (4800) устанавливается 1 в 4-й разряд РС (разрешение прерывания).

Количество и название машинных циклов – 2 цикла: выборка команды и исполнение (2 такта 8A и 8B). Цикла выборки адреса команды и цикла прерывания нет. То же самое и для команды DI (см. дальше)

DI – FB00 Запрещения прерывания

После цикла выборки команды перешли на цикл Исполнение:

8C	0800 0000	4400	DI	Запрещение прерывания
8D	8201 0008	8301		GOTO НАЧ(01)

01: СК -> БР

На 1-м такте (4400) устанавливается 0 в 4-й разряд РС (запрет прерывания).

88	0000 0001	4008	HLT	Останов машины
89	8201 0008	8301		GOTO НАЧ(01)

Регистр состояния	
Разряд	Содержимое
0	Перенос (C)
1	Нуль (Z)
2	Знак (N)
3	0 - используется для организации безусловных переходов в МПУ
4	Разрешение прерывания
5	Прерывание
6	Состояния ВУ (Ф)
7	Состояние тумблера РАБОТА/ОСТАНОВ (1 - РАБОТА)
8	Программа
9	Выборка команды
10	Выборка адреса
11	Исполнение
12	Ввод-вывод

3. Вектора прерываний. Преимущество использования векторов прерываний.

Вектор прерывания

Прочитай в Википедии.

Вектор прерывания — закреплённый за устройством номер, который идентифицирует соответствующий обработчик прерываний. Векторы прерываний объединяются в [таблицу векторов прерываний](#), содержащую адреса обработчиков прерываний. Местоположение таблицы зависит от типа и режима работы процессора.

Таблица векторов прерываний занимает первый килобайт оперативной памяти — адреса от 0000:0000 до 0000:03FF. Таблица состоит из 256 элементов — FAR-адресов обработчиков прерываний. Эти элементы называются **векторами прерываний**. В первом слове элемента таблицы записано смещение, а во втором — адрес сегмента обработчика прерывания. Векторами являются просто полные адреса памяти программы (в сегментированной форме), которая должна быть активизирована в случае возникновения прерывания.

Прерыванию с номером 0 соответствует адрес 0000:0000, прерыванию с номером 1 — 0000:0004 и т.д. Адрес такой состоит из пары 2-байтовых слов, поэтому каждый из векторов занимает четыре байта.

Вектор прерываний – начальный адрес подпрограммы обслуживания конкретного прерывания. Разрядность вектора совпадает с разрядностью ША.

Вектором прерывания называют электрический сигнал, посылаемый на шины процессора. По вектору прерывания можно определить всю необходимую информацию для перехода к прерывающей программе, в том числе ее начальный адрес. Каждому уровню прерывания соответствует свой вектор прерывания.

Векторы прерывания обычно находятся в специально выделенных фиксированных ячейках памяти.

Необходимость проверки готовности всех внешних устройств существенно увеличивает время обслуживания тех ВУ, которые опрашиваются последними. Это является основным недостатком рассматриваемого способа организации прерываний. Поэтому обслуживание прерываний с опросом готовности ВУ используется только в тех случаях, когда отсутствуют жесткие требования на время обработки сигналов прерывания внешних устройств.

Организация системы прерываний в микроЭВМ с использованием векторов прерываний позволяет устранить указанный недостаток. При такой организации системы прерываний ВУ, запросившее обслуживания, само идентифицирует себя с помощью вектора прерывания - адреса ячейки основной памяти микроЭВМ, в которой хранится либо первая команда подпрограммы обслуживания прерывания данного ВУ, либо адрес начала такой подпрограммы. Таким образом, процессор, получив вектор прерывания, сразу переключается на выполнение требуемой подпрограммы обработки прерывания. В микроЭВМ с векторной системой прерывания каждое ВУ должно иметь собственную подпрограмму обработки прерывания.

По вектору прерывания можно определить всю необходимую информацию для перехода к прерывающей программе, в том числе ее начальный адрес. Каждому уровню прерывания соответствует свой вектор прерывания. Векторы прерывания обычно находятся в специально выделенных фиксированных ячейках памяти.

Таблица векторов прерываний

Вектор 1	Начальный адрес подпрограммы обработки вектора 1
...	...
Вектор n	Начальный адрес подпрограммы обработки вектора n

Процедура организации перехода к прерывающей программе выделяет из всех выставленных запросов тот, который имеет наибольший приоритет, выполняет передачу текущего состояния прерываемой программы из регистров процессора в стек, загружает в регистры процессора вектор прерывания и передает управление прерывающей программе.

4. Потактовое выполнение цикла прерывания после команды BR (0).

BR (0) – команда безусловного перехода по адресу, записанному в ячейке 0000.

Последней микрокомандой была: 49 8390, что означает GO TO ПРЕ(8F)

Цикл прерывания				
8F	8288 0080	8788	ПРЕ	IF BIT(7,PC) = 0 THEN HLT(88)
90	8201 0020	8501		IF BIT(5,PC) = 0 THEN НАЧ(01)
91	0000 0200	0020		0 ==> БР
92	0004 0000	4001		БР ==> РА
93	0000 0008	0300		СК ==> БР
94	0008 0000	4002		БР ==> РД
95	0100 0400	0012		РД ==> ОП(РА), 1 ==> БР
96	0020 0000	4004		БР ==> СК
97	0800 0000	4400		Запрещение прерывания
98	8201 0008	8301		GOTO НАЧ(01)

88	0000 0001	4008	HLT	Останов машины
89	8201 0008	8301		GOTO НАЧ(01)

Сразу же проверяется 7 бит PC, если он в состоянии ОСТАНОВ, то следует переход на микрокоманду 88.

5. Когда выполняется цикл обработки прерывания? После каких команд он не выполняется? Почему?

Прерывание программы наступает при наличии следующих условий: контроллер прерываний находится в состоянии разрешения прерывания, появился запрос на прерывание (флаг какого-либо ВУ установился в единичное состояние) и окончилось выполнение любой текущей команды, кроме команд D1 ("Запрещение прерывания") и E1. И ещё прерывания не будет после выполнения команды HLT. Если в момент запроса на прерывание исполнялась команда E1, то состояние прерывания наступит после выполнения следующей за E1 командой.

6. Обрабатываются ли прерывания в пошаговом режиме (режиме останов) работы программы? Почему?

Нет. Я думаю, что в пошаговом режиме в регистре состояния 7 разряд равен нулю, т.е. состояние – ОСТАНОВ (равен 1 в состоянии РАБОТА). А после выполнения команды HLT прерывание не выполняется. Работа в пошаговом режиме – останов после каждого нажатия кнопки ПРОДОЛЖЕНИЕ.

7. Что происходит при одновременном поступлении сигнала готовности нескольких внешних устройств? В какой последовательности они будут обработаны?

В какой последовательности задаёт программа.

8. Почему не будет обработано прерывание сразу после выполнения команды EI? Для чего сделано так, что после цикла исполнения команды EI цикл прерывания не вызывается и прерывания не обрабатываются?

Программа обработки прерывания завершается командами E1 и BR (0). EI должна обязательно располагаться непосредственно перед командой выхода из подпрограммы - BR (0). Это делается на случай очередного требования прерывания во время обработки рассматриваемого. Если между командами E1 и BR (0) поместить хотя бы одну команду, то после ее выполнения может произойти переход к обработке прерывания, содержимое ячейки 0 заменится новым адресом возврата, и таким образом путь возвращения к прерванной (основной) программе будет разрушен.

9. За что отвечают биты 4, 5, 6 и 7 регистра состояния? Когда изменяется их значение?

4 – команда EI устанавливает этот разряд в 1, а команда DI – в 0. Он сбрасывается и после вхождения в подпрограмму обработки прерывания. См. на эмуляторе в “Работа с МПУ”.

В то же состояние переходит этот контроллер и после нажатия кнопки ПУСК.

(В эмуляторе при наведении на разряды РС высвечивается их название.)

5 – Запрос прерывания. (Кстати, в пошаговом режиме он так и висит, хотя в режиме РАБОТА сразу же гасится после начала обработки прерывания.)

6- Состояние флага.

7 – РАБОТА/ОСТАНОВ

Регистр состояния	
Разряд	Содержимое
0	Перенос (C)
1	Ноль (Z)
2	Знак (N)
3	0 - используется для организации безусловных переходов в МПУ
4	Разрешение прерывания
5	Прерывание
6	Состояния ВУ (Ф)
7	Состояние тумблера РАБОТА/ОСТАНОВ (1 - РАБОТА)
8	Программа
9	Выборка команды
10	Выборка адреса
11	Исполнение
12	Ввод-вывод

С помощью команды DI контроллер прерываний устанавливается в состояние запрещения прерывания, и ЭВМ не реагирует на запросы обмена от любых внешних устройств. В то же состояние переходит этот контроллер и после нажатия кнопки ПУСК, поэтому для обеспечения обмена по прерыванию программы в последней обязательно должна присутствовать команда EI ("Разрешение прерывания").