

# лаба 6 - обмен данными с ВУ по прерываниям

Конспект составлен на основе информации из методички и статьи:

<https://vk.com/@-192451261-preryvaniya-kak-smysl-zhizni-ili-manual-k-6-laboratornoi-po>

## 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 аппаратной проверкой наличия внешнего прерывания, т. е. сигнала готовности по линии "Запрос прерывания". 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



**Прерывание** — сигнал от программного или аппаратного обеспечения, сообщающий процессору о наступлении какого-либо события, требующего немедленного внимания. В контексте БЭВМ это сигнал о готовности обмена данными с некоторым ВУ.

Также стоит отметить преимущества над асинхронным режимом передачи данных (его можно двумя способами: через spin-loop или одну проверку с последующим окном ввода-вывода в циклической полезной программе):

- процессор не простаивает в ожидании готовности ВУ

- организованная работа сразу со всеми нужными нам ВУ
- если все ВУ не готовы к обмену, то процессор занят полезной работой

Ярким отличием прерываний является обязательное **сохранение состояния процессора в момент прерывания**, чтобы когда прерывание было обработано, мы смогли вернуться в основную программу и смогли продолжить ее выполнение без разнообразных коллизий.

Действия, выполняемые при этом процессором, как правило, те же, что и при обращении к подпрограмме. Только при обращении к подпрограмме они иницируются

**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 БЭВМ. 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.

**Команды работы разрешения/запрещения прерываний БЭВМ, команда программного прерывания, команда возврата из прерывания. Название, назначение и тип команды. Количество и название машинных циклов, потактовое выполнение команды.**

Это все безадресные команды

**EI (0x1100)**

1. Разрешение прерываний

2. Устанавливает 5 бит регистра состояния (PS) в 1
3. Будут выполняться цикл выборки команды (Instruction Fetch), цикл исполнения (Execution) и цикл прерывания (Interrupt)


### DI (0x1000)

1. Запрет прерываний
2. Устанавливает 5 бит регистра состояния (PS) в 0
3. Будут выполняться цикл выборки команды (Instruction Fetch), цикл исполнения (Execution) и цикл прерывания (Interrupt)

### IRET (0x0B00)

1. Возврат из прерывания
2. Возвращает состояние процессора до прерывания
3. Будут выполняться цикл выборки команды (Instruction Fetch), цикл исполнения (Execution) и цикл прерывания (Interrupt)

## Вектора прерываний. Преимущества использования векторов прерываний.



## Вектор прерывания

- Совокупность адреса программы обработки прерывания и регистра состояния (PS)
- Необходимо инициализировать перед началом обработки прерывания
  - Хотя бы установить на подпрограмму, которая ничего не делает
  - Ответственность ОС и БИОС
- В БЭВМ-NG ячейки с 0x0 по 0xF
  - Всего 8 векторов, по два слова на вектор
  - На одном векторе может быть несколько прерываний

Адр .	Сод .
0x0	Адр 0
0x1	PS 0
0x2	Адр 1
0x3	PS 1
0x4	Адр 2
0x5	PS 2
0x6	Адр 3
0x7	PS 3
0x8	Адр 4
0x9	PS 4
0xA	Адр 5
0xB	PS 5
0xC	Адр 6
0xD	PS 6
0xE	Адр 7
0xF	PS 7

**Вектор прерывания** — это совокупность адреса обработчика прерывания и регистра состояния, с которым этот обработчик стартует (данное высказывание справедливо в контексте БЭВМ, в других ЭВМ количество нужных ячеек для вектора прерывания варьируется). На 1 векторе может быть несколько прерываний.

В БЭВМ доступно 8 векторов прерываний, и расположены они в ячейках памяти 0x0 — 0xF включительно. На один вектор прерывания может приходиться несколько прерываний.

В контексте лабораторной работы у нас 2 вектора прерывания, к которым будут привязаны обработчики с уникальным поведением. Остальные вектора прерываний привязываем к дефолтному обработчику. Данное действие нужно, потому что в случае привязывания какого-либо ВУ к непроинициализированному вектору прерывания все сломается при прерывании от данного ВУ.

В самой простейшей реализации данного дефолтного обработчика можно использовать IRET, но это недопустимо, если данным обработчиком обрабатывается какое-либо ВУ. Если оставить просто IRET, то программа заиклится в этом обработчике, поэтому всегда в дефолтном обработчике нужно сбрасывать готовность тех ВУ, которые обрабатываются данным обработчиком.

Таблица В.16

Ввод данных с использованием прерываний. Ассемблерный листинг.

адрес обработчика ← **Листинг примера: Готовность ВУ1: 2\*А-РДВУ1, Готовность ВУ3: РДВУ3-яч.3Ф**

```

ORG 0x0 ; Инициализация векторов прерывания
V0: WORD $DEFAULT, 0x180 ; Вектор прерывания #0
V1: WORD $INT1, 0x180 ; Вектор прерывания #1
V2: WORD $DEFAULT, 0x180 ; Вектор прерывания #2
V3: WORD $INT3, 0x180 ; Вектор прерывания #3
...
V7: WORD $DEFAULT, 0x180 ; Вектор прерывания #7
DEFAULT: IRET ; Просто возврат
START: DI
      CLA
      OUT 1 ; MR KBY-0 на вектор 0
      OUT 5 ; MR KBY-2 на вектор 0
      LD #9 ; разрешить прерывания и вектор №1
      OUT 3 ; (1000|0001=1001) в MR KBY-1
      LD #0xB ; разрешить прерывания и вектор №3
      OUT 7 ; (1000|0011=1011) в MR KBY-3

```

регистр состояния(PS), с которым этот обработчик начинает работу

1 1000 0000 0000 0000 0000 0000 0000

бит P (программа) бит W (работа)

При организации системы прерываний в БЭВМ с использованием векторов прерываний ВУ, запросившее обслуживания, само идентифицирует себя с

помощью вектора прерывания. Таким образом, процессор, получив вектор прерывания, сразу переключается на выполнение требуемой подпрограммы обработки прерывания.

## Регистр управления ВУ (MR DEV).

Регистр управления или Management Register — регистр, хранящий в себе информации о разрешении/запрете прерывания от данного ВУ и вектора прерывания, к которому привязано ВУ.

В 3х младших битах содержится номер вектора прерывания, в 3 бите будет 1, если прерывания от данного ВУ разрешены и 0 — если запрещены.



Для КВУ-1, КВУ-2, КВУ-3 MR и SR находятся по одному адресу, но это два **РАЗЛИЧНЫХ регистра** и при записи на данный адрес пишем именно в MR, а при чтении читаем именно из SR. В КВУ-4 SR и MR находятся уже на разных адресах.

## Сигналы шины БЭВМ, назначение, временные диаграммы сигналов Input и Output.

Контроллеры ВУ связаны с процессором при помощи системной шины БЭВМ, в сегменты (или, физически, разъемы для установки контроллеров) которой подсоединяются различные шины со стороны процессора и контроллера:

- шина данных (Data0..7), по которым происходит передача данных в процессор или из процессора;
- шина адреса (Addr0..7), по которой передается адрес внешнего устройства от процессора к КБУ и номер вектора запроса на прерывание (Int#) от КБУ к процессору, подтверждаемый сигналом выдачи вектора (IntV);
- сигнал запроса прерывания (IntRq), по которому выставляется требование прерывания от внешнего устройства;
- сигнал ввода (Input) - для передачи приказа на ввод (IN #reg);
- сигнал вывода (Output) - для передачи приказа на вывод (OUT #reg);
- начальный сигнал предоставления прерывания (IntSC), который управляется программно микрокодом в цикле прерывания по микрокоманде INTS.
- входящий цепочный сигнал предоставления прерывания (IntSCi# - Interrupt Supply Chain input), по которому контроллер, в соответствии с порядком подключения к шине, проверяет возможность предоставления ему прерывания в соответствии с очередью подключения, и генерирует прерывание, если они разрешены глобально командой EI и в регистре MR контроллера;
- исходящий цепочный сигнал предоставления прерывания (IntSCo# - Interrupt Supply Chain output), который передает контроллер по шине далее следующему КБУ, если нет необходимости вызвать прерывание вычислительного процесса;
- сигнал готовности (Rdy), подтверждающий завершение операции ввода-

вывода внутри цикла обмена между AC и DR соответствующего КБУ. В случае операции ввода Rdy подтверждает данные, передаваемые по шине данных, и в обоих случаях операции ввода-вывода сигнализирует о том, что цикл обмена с регистром данных DR контроллера завершен;

- сигнал синхронизации (Syn) от тактового генератора БЭВМ, который задает временной слот единичного обмена по шине БЭВМ.

### **Про ввод-вывод, управляемый по прерыванию**

Когда будет нажата кнопка готовности, то в случае, если в контроллере прерывания разрешены поступает сигнал "Запрос прерывания" по линии IntRq. После чего этот сигнал поступит на схему „И" центрального процессора с 5 битом регистра состояния (EI - разрешение прерываний) проверяющий разрешены ли прерывания в ЭВМ и записывающий результат схемы в 6 бит регистра состояния.

После цикла исполнения очередной инструкции основной программы в ЭВМ будет запущен комплекс программ, проверяющий, было ли требование прерывания, и если было, то формирует управляющий сигнал

"Предоставление прерывания" IntSC, который последовательно проходит через все контроллеры (сигналы IntSCo - выходной и IntSCi - входной) и проверяет было ли в этом контроллере прерывание.

Когда сигнал доходит до нужного контроллера, то он выставляет на шину адреса свой номер вектора прерывания. После чего процессор по номеру прерывания вызывает программу обработки прерывания.

Если прерывания разрешены, то после выполнения всех команд кроме HLT, а также если режим выполнения БЭВМ установлен в значение "РАБОТА", выполняются следующие действия:

**Шаг 1.** По завершению цикла исполнения текущей команды происходит переход на цикл прерывания (Interrupt). Если в этот момент 6 бит регистра состояния IRQ (прерывание) не равен 1, то происходит переход к следующей команде. При наличии требования прерывания БЭВМ формирует сигнал "Предоставление прерывания" (IntSC) через выполнение микрокоманды INTS.

**Шаг 2.** Сохраняется счетчик команд и регистр состояния БЭВМ в стеке.

**Шаг 3.** Младшие 8 разрядов (номер вектора прерывания) записываются в BR и вычисляется адрес с переходом на подпрограмму обработки прерывания, как номер вектора \* 2, после чего тот записывается в DR, а после в IP.

**Шаг 4.** Далее к младшим 8 разрядам BR прибавляется 1, чтобы выбрать адрес следующей ячейки вектора прерывания (новый регистр состояния PS), ограничивая результат 8-ю разрядами. После чего по этому адресу содержимое в памяти записывается в DR, а после в PS.

**Шаг 5.** Контроллер прерываний вновь переводится в состояние разрешение прерывания командой EI и осуществляется возврат к выполнению прерванной программы, т.е. к команде, адрес которой хранится в стеке, также восстанавливается сохраненный регистр состояния из стека.

## 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 (в старшем бите MR находится 1), появился запрос на прерывание и окончилось выполнение любой текущей команды. Цикл обработки прерывания не выполняется после HLT, происходит остановка программы

## Обрабатываются ли прерывания в пошаговом режиме (режиме "ОСТАНОВ") работы программы? Почему?

В режиме ОСТНОВ не обрабатываются, так как цикл Interrupt требует для отработки единицу в 7 бите регистра состояния (PS)

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

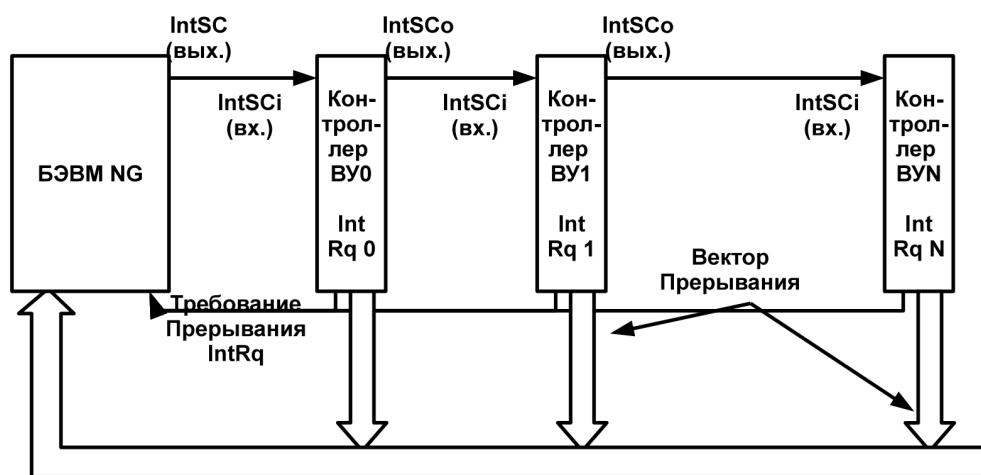


Рисунок В.15 Распространение сигналов предоставления прерывания

Сигнал предоставления прерывания проходит через все КВУ, пока не дойдет до того, чье ВУ вызвало прерывания.

В БЭВМ приоритет получения КВУ сигнала предоставления реализован аппаратно, как можно увидеть по картинке. Отсюда вытекает очень интересный факт, что **приоритет обработки прерываний от разных ВУ реализован аппаратно.**



Рассмотрим ситуацию: запрос на прерывание формирует сначала ВУ-3, а потом ВУ-2. В итоге в PS в 6 бите будет 1. Неизвестно от какого ВУ пришло прерывание. На момент предоставления прерывания сигнал IntSC зайдет сначала в ВУ-2 и обработается именно это ВУ, несмотря на то, что ВУ-3 запросил прерывание раньше ВУ-2. А потом уже обработается ВУ-3.

**За что отвечают биты 5, 6, 7 и 8 регистра состояния? Когда изменяется их значение?**

Бит	Мнем .	Содержимое
0	<b>C</b>	Перенос
1	<b>V</b>	Переполнение
2	<b>Z</b>	Ноль
3	<b>N</b>	Знак
4	<b>O</b>	0 – используется для организации безусловных переходов в МПУ
5	<b>EI</b>	Разрешение прерываний
6	<b>IRQ</b>	Требование прерывания (логическое “И” шины запроса на прерывание и бита 5 РС – “разрешение прерываний”)
7	<b>W</b>	Состояние тумблеров РАБОТА/ОСТАНОВ (1 – РАБОТА)
8	<b>P</b>	Программа

### **Бит 5 (EI)**

EI — бит разрешения прерываний. Изменяется с помощью команд DI & EI (запрет и разрешение прерываний). 1 когда прерывания разрешены и 0 когда запрещены.

### **Бит 6 (IRQ)**

