

Контрольные вопросы, ОВТ лаб 5

Самое важное! (ну, или почти)

Само устройство понимает, что пора забрать из своего РД данные и обработать их только по обнулению своего флага!

То есть код:

...

X: WORD DEAD

BEGIN: ADD X

OUT 3

CLA

OUT 3

CLF 3 -- вывод на экран произойдет только в этот момент

...

Выведет на экран ВУ-3 только 00, а не AD и 00

Системная шина в БЭВМ

Шина адреса памяти(РА - Пам) + шина данных памяти(РД-Пам) + шина ввода-вывода(РД ВУ - Аккум) = Системная шина

Синхронный и асинхронный режимы передачи данных

Синхронная инициация: обмен начинается в заранее известный промежуток времени (например, каждую минуту).

Асинхронная инициация: программе неизвестно время начала обмена данными и она вынуждена периодически проверять возможность обмена (готовность внешнего устройства).

Программно-управляемый и управляемый прерываниями ввод-вывод, прямой доступ к памяти. Преимущества и недостатки.

Программно-управляемый. Программа напрямую управляет процессом ввода-вывода: проверяет состояние устройства, посылает команды ввода-вывода, при выполнении которых процессор должен ожидать завершения операции.

По прерываниям. Чтобы исключить периодическую проверку готовности, устройства могут сами инициировать обмен по специальному аппаратному

Способы и формат представления символьных и строковых данных в БЭВМ. Кодировки ASCII, KOI-8, Windows-1251, ISO-8859-5, UTF-8, UTF-16.

.ASCII изначально разработана как таблица символов с 7-битными кодами.

В KOI-8 добавлен восьмой бит для кодирования кириллицы, при этом символы русского алфавита размещены так, что их позиции соответствуют фонетическим аналогам в английском алфавите (если убрать последний бит, «Русский Текст» превращается в «rUSSKIJ tEKST»). Широко использовалась как основная русская кодировка в Unix-совместимых ОС до распространения Юникода.

Windows-1251 - 8-битная кодировка для русских версий Windows. В отличие от KOI-8, символы расположены в алфавитном порядке.

ISO 8859-5 - 8-битная кодировка кириллицы. Основным недостатком - отсутствие некоторых символов, такие как тире (—), кавычки-ёлочки («»), градус (°), поэтому в России почти не использовалась (вместо нее до Юникода применялись предыдущие).

С 00 по 20 и в конце там СПЕЦСИМВОЛЫ

по Нижняя часть таблицы (7 бит) в восьмибитных кодировках соответствует ASCII.

UTF-8 —распространённый стандарт кодирования текста, позволяющий более компактно хранить и передавать символы Юникода, используя переменное количество байт (от 1 до 4), и обеспечивающий полную обратную совместимость с 7-битной

Принцип кодирования

Правила записи кода одного символа в UTF-8

1. Если размер символа в кодировке UTF-8 = 1 байт

Код имеет вид (0aaa aaaa), где «0» — просто ноль, остальные биты «a» — это код символа в кодировке ASCII;

2. Если размер символа в кодировке в UTF-8 > 1 байт (то есть от 2 до 6):

2.1 Первый байт содержит количество байт символа, закодированное в **единичной** системе счисления;

2 — 11

3 — 111

4 — 1111
5 — 1111 1
6 — 1111 11

2.2 «0» — бит терминатор, означающий завершение кода размера

2.3 далее идут значащие байты кода, которые имеют вид (10xx xxxx), где «10» — биты признака продолжения, а «x» — значащие биты.

В общем случае варианты представления **одного символа** в кодировке UTF-8 выглядят так:

(1 байт) 0aaa aaaa
(2 байта) 110x xxxx 10xx xxxx
(3 байта) 1110 xxxx 10xx xxxx 10xx xxxx
(4 байта) 1111 0xxx 10xx xxxx 10xx xxxx 10xx xxxx
(5 байт) 1111 10xx 10xx xxxx 10xx xxxx 10xx xxxx 10xx xxxx
(6 байт) 1111 110x 10xx xxxx 10xx xxxx 10xx xxxx 10xx xxxx 10xx xxxx

с рубежка)

- В UTF-16 символы кодируются двухбайтовыми словами с использованием всех возможных диапазонов значений (от 0000_{16} до $FFFF_{16}$). При этом можно кодировать символы Unicode. Сдвигаем символ в 8 в диапазонах $0000_{16}..D7FF_{16}$ и $E000_{16}..10FFFF_{16}$. Исключенный отсюда диапазон $D800_{16}..DFFF_{16}$ используется как раз для кодирования так называемых суррогатных пар — символов, которые кодируются двумя 16-битными словами. Символы Unicode до $FFFF_{16}$ включительно (исключая диапазон для суррогатов) записываются как есть 16-битным словом. Символы же в диапазоне $10000_{16}..10FFFF_{16}$ (больше 16 бит) уже кодируются парой 16-битных слов. Для этого их код арифметически сдвигается до нуля (из него вычитается минимальное число 10000_{16}). В результате получится значение от нуля до FFFF, которое занимает до 20 бит. Старшие 10 бит этого значения идут в лидирующее (первое) слово, а младшие 10 бит — в последующее (второе). При этом в обоих словах старшие 6 бит используются для обозначения суррогата. Биты с 11 по 15 имеют значения 11011_2 , а 10-й бит содержит 0 у лидирующего слова и 1 — у последующего.

Порядок байтов в памяти от младшего к старшему (little-endian) и от старшего к младшему (big-endian).

Big-endian - прямой порядок, является стандартным в TCP/IP (сетевой порядок байт) и некоторых файловых форматах (PNG). Используется в процессорах SPARC.

Little-endian - обратный порядок записи, используется в архитектуре x86.

Endian-independent и не Endian-independent в контексте UTF (*требуется ответа*)

В БЭВМ порядок байтов определяется программой, т.к. команды не работают с отдельными байтами

Система команд ввода-вывода БЭВМ. Команды IN, OUT, CLF, TSF - название, назначение и тип команды. Количество и название машинных циклов, потактовое выполнение команды, с перечислением всех шин, участвующих в обмене.

Флаг - однобитовый регистр готовности ВУ, устанавливаемый в единичное состояние, когда ВУ готово к обмену информацией

CLF В (E0xx, xx - две последние 16-ричные цифры адреса ВУ) - сброс в нуль флага готовности ВУ.

TSF В (E1xx) - проверка готовности к обмену ВУ. Если флажок этого ВУ сброшен (ВУ не готово к обмену), то выполняется команда, расположенная вслед за TSF В. В противном случае следующая команда пропускается и выполняется команда, расположенная через одну за TSF В.

IN В (E2xx) - пересылка содержимого регистра данных контроллера ВУ в восемь младших разрядов аккумулятора.

OUT В (E3xx) - пересылка содержимого восьми младших разрядов аккумулятора в регистр данных контроллера ВУ.

Уметь сократить до двух - IN и OUT. (Просто CLF и TSF выполняются автоматически после и до приказа соответственно)

"Придумайте мне интересную новую команду ввода-вывода"

-Можно реализовать работу со старшими битами аккумулятора, но для этого нужно добавить дополнительный регистр состояния у контроллеров ВУ для понимания, по какой шине передавать, и провести эту самую ещё одну шину в старшие разряды аккумулятора.

Какие режимы передачи данных и управления вводом-выводом реализуемы в БЭВМ? Почему не возможно реализовать другие?

Программно-управляемый и по прерываниям. Отсутствует прямой доступ к памяти, так как БЭВМ не обладает контроллером ПДП.

Может ли ВУ определить в каком режиме с ним работают?

Нет. Контроллер ВУ знает только - пришел приказ IN, я отдаю содержимое РД, OUT - принимаю данные в РД. Для него абсолютно не важно исходя из чего процессор формирует эти приказы

В каких случаях ВУ может что-то менять в регистре данных контроллера ВУ?

Вроде, при условии, что данное ВУ - устройства ввода и флаг готовности установлен в 0.

Циклы:

--- Происходит выборка команды, частичное декодирование

--- Исполнение

-- Младшие 8 бит регистра данных отправляются по шине адреса на дешифраторы адресов и приказов внешних устройств

-- Следующие четыре бита отправляются по шине Приказа на ввод-вывод на дешифраторы адресов и приказов внешних устройств (первые 4 бита нет смысла передавать, там просто E)

-- Дешифраторы адреса и приказов сравнивают пришедший адрес со своим, нужный триггерится {тут должна быть таблица соответствий} и принимает приказ, который дешифрируется

-- Приказ исполняется

- Если приказ 0 (CLF):

Выполняется сброс (установка в ноль) флага готовности устройства

- Если 1 (TSF):

Содержимое флага готовности отправляется в индикатор состояния ВУ

Отправляется содержимое флага готовности, обновляет 6 бит регистра состояния ВУ.

по линии "Состояние флага"

счетчик команд с этим значением

Если оно равно 1 - то следующая команда пропускается, путем инкрементации СК, иначе -

выполнится

- Если 2 **(IN)**:

Содержимое РД ВУ пересылается в младшие биты аккумулятора по шине ввода

- Если 3 **(OUT)**:

Содержимое восьми младших бит аккумулятора помещается в регистр данных данного ВУ (по шине вывода)

Для перевода в UTF-8, UTF-16

<https://r12a.github.io/apps/conversion/>