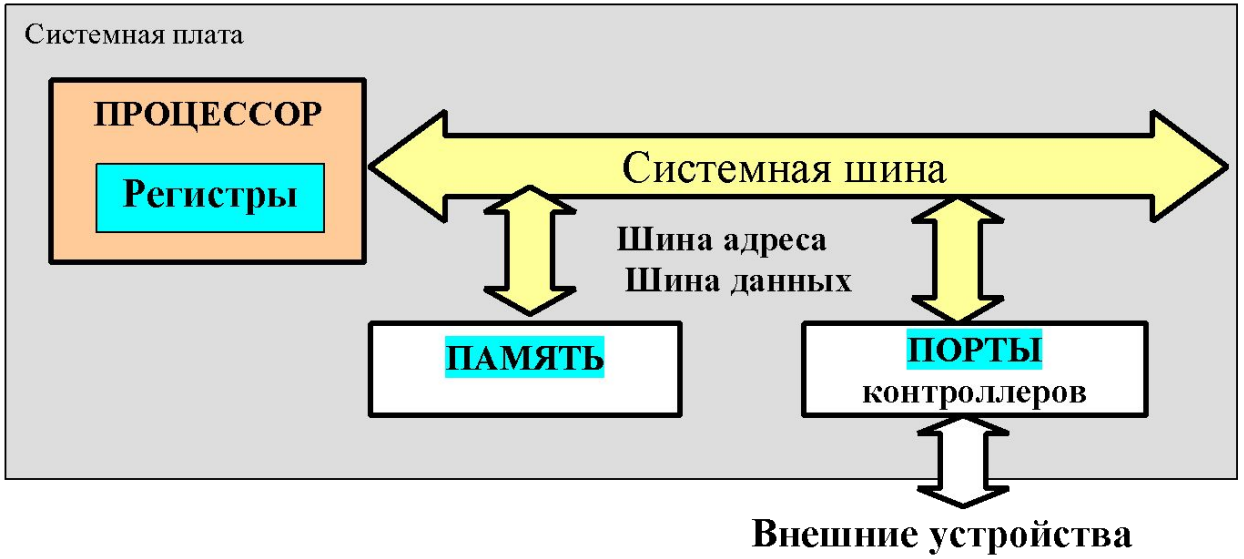
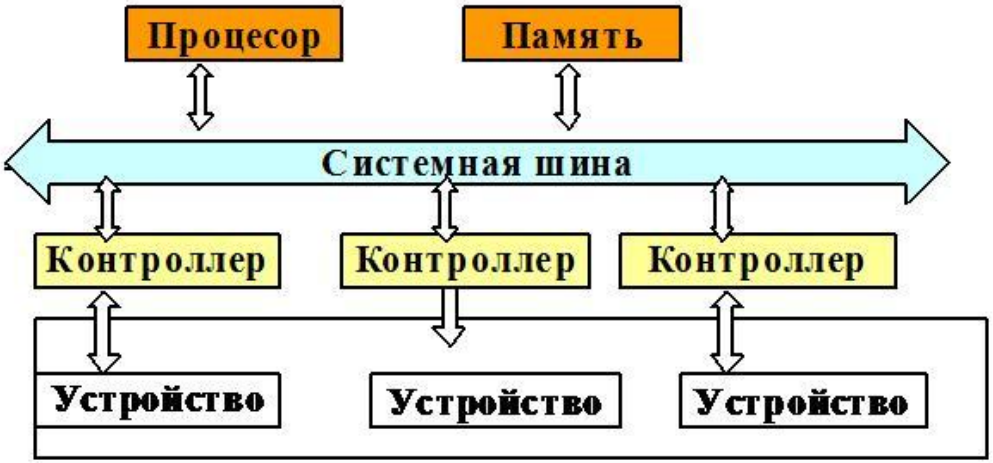
Лекция 18. Команды обращения к портам контроллеров

Память и порты контроллеров – единственные «внешние» по отношению к процессору объекты, к которым он может обратиться при выполнении своих команд



Контроллеры внешних устройств

- С любым внешним устройством процессор связан через специальный «контроллер»
- Контроллеры – это интеллектуальные устройства с собственным микропроцессором и памятью, выполняющие непосредственное управление определенным внешним устройством (клавиатура, экран, устройства хранения информации, принтеры и т.д.



«Порты» контроллеров

Внутри контроллеров есть 8-разрядные регистры. Их и называют «порты». Обычно, в контроллере 3-4 порта.

- Порты всех контроллеров имеют сквозную нумерацию в системе : 0, 1, 2, 3 и т.д.
- Каждый порт имеют определенное назначение, обычно: порт/порты данных, порт управления и порт состояния устройства

Через «порты данных» процессор может записывать/считывать байты данных из контроллера.

«Порт управления» используется для записи кодов, управляющих работой контроллера устройства.

Из «порта состояния» можно считать код текущего состояния контроллера.

- Процессор может обращаться к портам контроллеров по командам IN и OUT (в группе команд пересылки).

Команды обращения к портам

Две команды (и несколько их модификаций), заставляющие процессор обращаться к портам контроллеров:

- IN - прочитать из порта с заданным номером байт/слово/двойное слово в регистр AL/AX/EAX процессора
- OUT - записать в порт с заданным номером байт/слово/двойное слово из регистра AL/AX/ EAX

Номер порта можно задавать прямо или косвенно

а) прямо заданный номер порта (port). Диапазон номеров: 0-255

```
IN      AX/AL/EAX,  port
OUT  port,  AX/AL/EAX
```

б) косвенно заданный номер порта (задается в регистре DX: от 0 -65535)

```
IN      AX/AL/EAX,  DX
OUT   DX,  AX/AL/EAX
```

Примеры:

- прочитать байт из порта 60h

```
in  al, 60h
```

- прочитать байт из порта 3D0h

```
mov  dx, 3D0h
in al, dx
```

- записать байты в порты 80h и 81h

```
out 80h,  ax    ; в порт  80h ← al, в порт 81h ← ah
```

- записать байты в порты 80, 81,82 и 83h

```
out 80h,  eax  ; старший байт будет записан в порт со
                 старшим номером
```

Безоперандные строковые команды IN и OUT для пересылки между портами и памятью

- Пересылка из порта в память (байтов, слов, двойных слов):
  INSB , INSW , INSD
- Пересылка в порт из памяти:
  OUTSB, OUTSW, OUTSD

Операндами по умолчанию являются:
- номер порта - в регистре DX.
- адрес памяти - DS:SI в командах OUTS, ES:DI - в командах INS

Механизм исполнения - как в любых строковых командах:
- адрес памяти автоматически меняется в сторону увеличения или уменьшения в зависимости от флага DF
- в строковой команде можно использовать префикс повторения – REP. Тогда процессор повторяет исполнение команды, используя CX в качестве вычитающего счетчика циклов

  Пример: rep insw

Пример 1. Переслать байт из памяти (адрес ds:m1) в порт номер 3F1h

```
        .   .   .   .
met1    db      ?
        .   .   .   .
          lea  si,  met1
          mov   dx, 3f1h
          outsb
```

Пример 2. Переслать в порты контроллеров, начиная с номера 3F1h, десять слов из памяти, начиная с адреса ds: m1.

```
        .   .   .   .
m1  dw  10 dup(?)
        .   .   .   .
        lea  si,  m1
        mov  dx, 3f1h
        mov  cx, 10
        cld              ;   флаг DF←0
        rep outsw        ; цикл пересылки с повторением 10 раз
```

Пример 3. Программное обращение к устройству «CMOS-память»

- Энергонезависимая CMOS-память – это микросхема, включающая в себя часы реального времени и информацию о конфигурации компьютера (Setup).

CMOS-память питается от батарейки. Объем ее памяти - 64 или 128 байтов. Ее байты имеют номера – от 00 до 3Fh/или 80h. Размещение информации в байтах памяти строго определено и является справочной информацией.

- Программное обращение к CMOS-памяти делается через порты:

- порт 70 h – в него надо записать номер байта CMOS-памяти, из которого планируется чтение
- порт 71 h – отсюда можно будет прочитать значение байта, номер которого задан в порте 70 h

Задача: проверить состояние питающей батарейки CMOS-памяти. Старший бит в байте номер 0Dh CMOS-памяти показывает ее состояние: 1 – батарейка исправна, 0 - нет

```
;  запишем в порт 70h номер интересующего нас байта CMOS-памяти
            mov al, 0dh
            out  70h, al
;  чтение из порта 71h значения этого байта
            in    al, 71h
;  выделим и проверим старший бит прочитанного байта
            and al, 80h
            jz  short  batteryOff
; батарейка исправна
          . . . . . .
batteryOff :
; неисправна
          . . . . . .
```