



# ОСНОВЫ ПРОГРАММИРОВАНИЯ СИСТЕМ УПРАВЛЕНИЯ

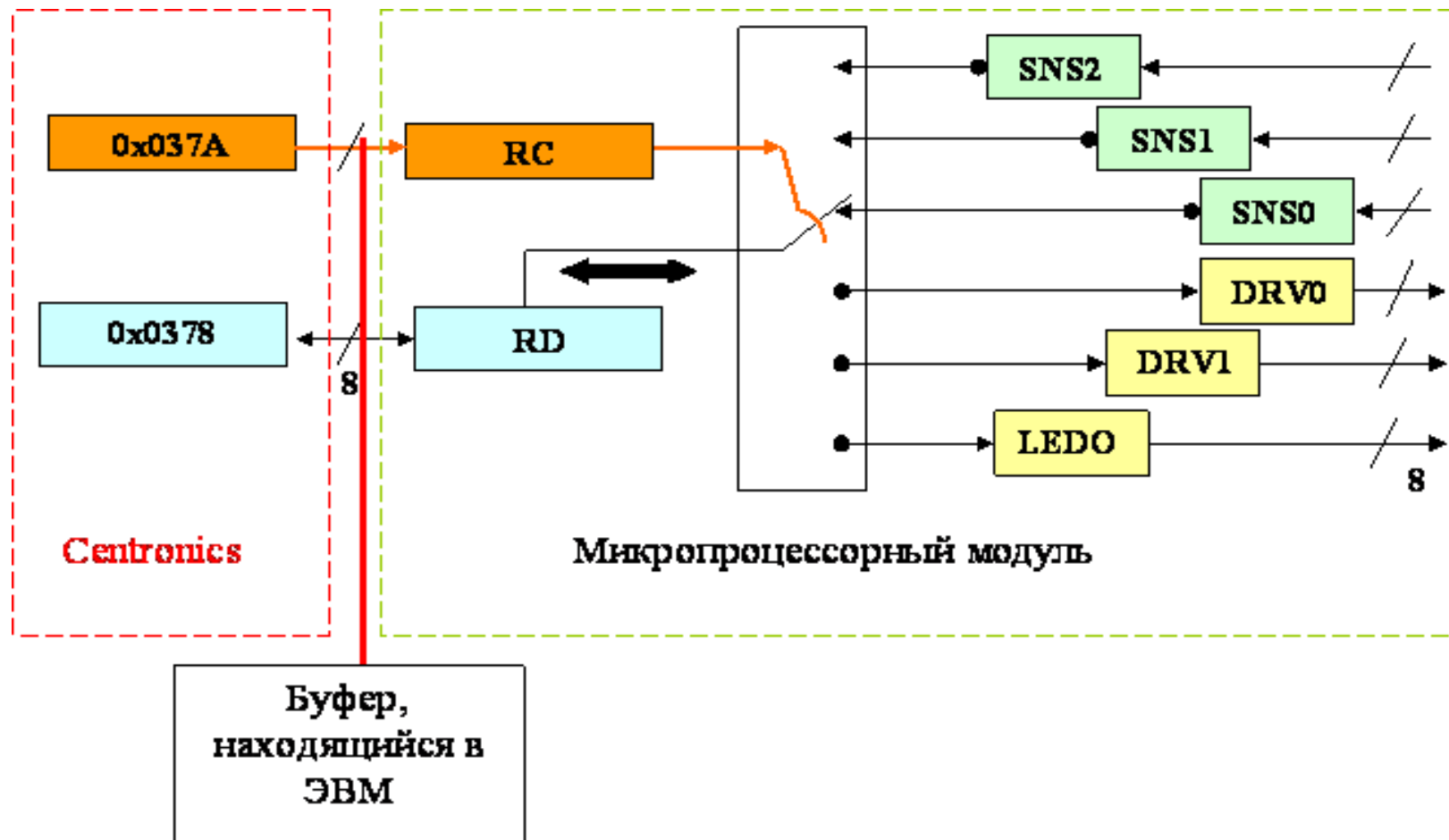


Лекция №6

# Что нужно знать при написании функции чтения данных из регистров микропроцессорного модуля БУ УРТК

1. Назначение разрядов регистра чтения данных по формату *Centronics*,
2. Алгоритм формирования байтов при чтении,
3. Временную диаграмму для чтения данных из регистров 0x378 и 0x37A,
4. Правила формирования данных по тактам и правила формирования байтов,
5. Соответствие между числами в разных системах счисления, и использование кодов в функции чтения,
6. Некоторые библиотечные функции `inport ()`, `outport ()`, `delay()`,

# Схема обмена данными между Centronics и микропроцессорным модулем БУ.



# Чтение и запись информации.

## *Схема диалога с блоком управления:*

- В регистр RC заносится константа – управляющее слово (команда), определяющая режим диалога.
- Через регистр RD производится чтение или запись данных. Если необходимо передать более одного байта, передача байт через регистр RD производится последовательно.
- После отправки последнего байта действие считается завершенным.

# Алгоритм действий.

Значение константы	Описание действия	Алгоритм действия
11h	Установка состояния двигателей	1. В регистр RC отправить константу 11h; 2. Через регистр RD последовательно передать два байта состояния двигателей: DRV1, DRV0.
12h	Установка состояния светодиодных индикаторов	1. В регистр RC отправить константу 12h; 2. Через регистр RD передать байт состояния светодиодных индикаторов LED0.
13h	Получение текущего состояния датчиков	1. В регистр RC отправить константу 13h; 2. Через регистр RD последовательно принять три байта состояния датчиков: SNS0, SNS1 и SNS2.
14h	Получение текущего состояния клавиатуры	1. В регистр RC отправить константу 14h; 2. Через регистр RD последовательно принять три байта состояния клавиатуры: KBD0 и KBD1.

# Назначение разрядов регистра управления Centronics

Номер разряда	Название	Инверсия	Назначение разряда
0	RD	Да	Строб чтения данных из СУ
1	A0	Да	Младший разряд шины адреса
2	WR	Нет	Строб записи данных в СУ
3	A1	Да	Старший разряд шины адреса
4			Не используется. Должен быть равен 0
5	B5	Нет	Дополнительный бит для режима чтения
6			Не используется. Должен быть равен 0
7			Не используется. Должен быть равен 0



# Функции чтения и записи.

***Запись данных:***

*outportb(Addr, Data)*

*Где:*

Addr – адрес порта,

Data – записываемые данные.

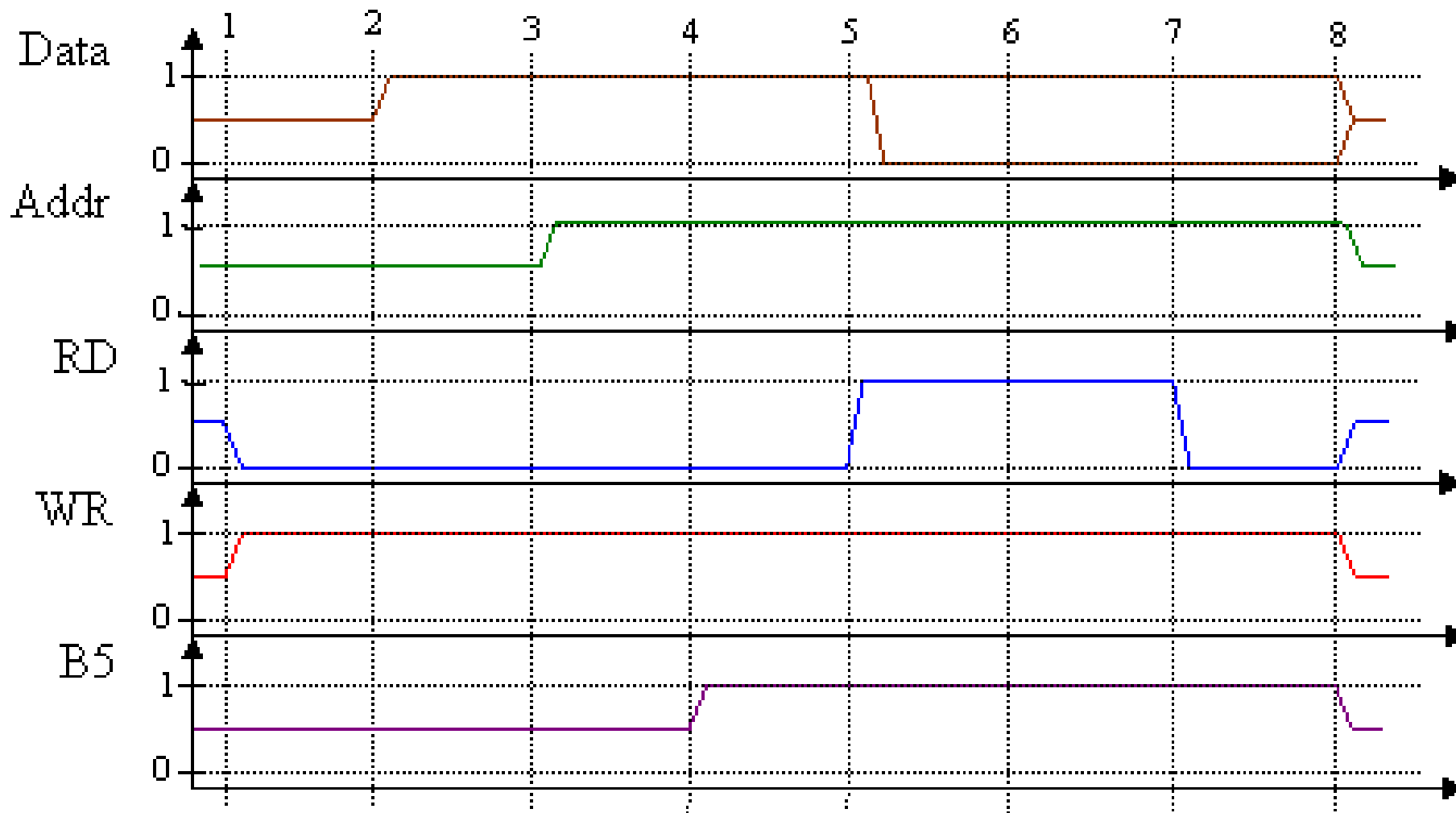
***Чтение данных:***

*inportb(Addr)*

*Где:*

Addr – адрес порта.


# Временная диаграмма чтения данных из СУ.







# Последовательность этапов чтения.

- Регистр RC устанавливается в начальное положение.
  - В регистр RD устанавливается управляющая константа.
  - В регистр RC устанавливается адрес.
  - В регистр RC устанавливается байт в котором бит B5 имеет значение "1".
  - Происходит чтение данных DATA (после задержки).
  - Регистр RC устанавливается в начальное положение.
- 

## Формирование данных.

- 00000100 ~ 0<sub>x</sub>04 или `outportb(0x37A, 0x04);`
- 11111111 ~ 0<sub>x</sub>FF или `outportb(0x378, 0xFF);`
- 00001110 ~ 0<sub>x</sub>0E или `outportb(0x37A, 0x0E);`
- 00101110 ~ 0<sub>x</sub>2E или `outportb(0x37A, 0x2E);`
- 00101111 ~ 0<sub>x</sub>2F или `outportb(0x37A, 0x2F);`
- Задержка 2-7 ms или `delay(5);`
- Чтение данных ~ `data = inportb(0x378);`

**! На 8-м такте нет необходимости в каких-либо действиях.**

При чтении из *RD* в нем может быть байт SNS<sub>0</sub>, SNS<sub>1</sub>, SNS<sub>2</sub>

## Пример программы.

```
outport(0x37A, 0x2F); //в регистр управления  
                        записывается двоичное слово  
  
delay(5);             //задержка на 5 мсек  
  
result=inportb(0x378); // чтение данных из регистра данных
```

# Пример функции чтения.

```
unsigned char rbfr()  
{  
    unsigned char data;  
    outportb(0x37A,0x04);  
    outportb(0x378,0xFF);  
    outportb(0x37A,0x0E);  
    outportb(0x37A,0x2E);  
    outportb(0x37A,0x2F);  
    delay(5);  
    data=inportb(0x378);  
    return data;  
}
```

# Управление исполнительными устройствами.

Номер разряда	Разряды DRV <sub>0</sub>	Разряды DRV <sub>1</sub>
0	M <sub>0</sub> –	M <sub>4</sub> –
1	M <sub>0</sub> +	M <sub>4</sub> +
2	M <sub>1</sub> –	M <sub>5</sub> –
3	M <sub>1</sub> +	M <sub>5</sub> +
4	M <sub>2</sub> –	D <sub>0</sub>
5	M <sub>2</sub> +	D <sub>1</sub>
6	M <sub>3</sub> –	D <sub>2</sub>
7	M <sub>3</sub> +	E <sub>w</sub>

M<sub>0</sub>-M<sub>5</sub> – степени ММ

D<sub>0</sub>-D<sub>2</sub> - нереверсивные двигатели (схват, фреза).

E<sub>w</sub> – обмотка возбуждения.

# Включение двигателей.

Мо в положительном направлении  
М1 в отрицательном направлении } DRV<sub>0</sub>/DRV<sub>1</sub> ?

DRV<sub>0</sub>

0	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---

DRV<sub>1</sub>

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Алгоритм запуска двигателей:


- В регистр RC отправляется константа 11h
- Передать байт DRV<sub>1</sub>
- Передать байт DRV<sub>0</sub>

# Регистры СУ УРТК для опроса датчиков.

Номер разряда	Разряды SNSo	Разряды SNS1	Разряды SNS2
0	Mo нач	M2 нач	M4 нач
1	Mo кон	M2 кон	M4 кон
2	Mo имп	M2 имп	M4 имп
3	M1 нач	M3 нач	M5 нач
4	M1 кон	M3 кон	M5 кон
5	M1 имп	M3 имп	M5 имп
6	Резерво	Резерв2	Резерв4
7	Резерв1	Резерв3	Резерв5



# Алгоритм чтения состояния датчиков.

- в регистр RC отправить константу 13h
  - через регистр RD последовательно считать три байта информации состояния датчиков: SNS<sub>0</sub>, SNS<sub>1</sub> и SNS<sub>2</sub>.
- 



# Пример(фрагмент программы чтения состояния датчиков )

...

```
unsigned char SNS0, SNS1,  
SNS2;
```

//переменные, которым будут  
//присвоены прочитанные байты

...

```
write (REG, 0x13);
```

//REG – переменная, определяющая в какой  
//из регистров (RC или RD) будет происходить  
//запись (в данном случае регистр RC)

```
SNS0 = ~read ();
```

```
SNS1 = ~read ();
```

```
SNS2 = ~read ();
```

//последовательно считываются три байта  
//информации состояния датчиков

...

# Управление светодиодами индикаторами.

Номер разряда	Разряды LEDo
0	L0
1	L1
2	L2
3	L3
4	L4
5	L5
6	L6
7	L7

"1" зажигает соответствующий светодиод,  
"0" выключает светодиод.