

Миранян Сурен Артёмович

Санкт-Петербургский государственный политехнический университет
Факультет технической кибернетики

Программно аппаратная защита информации

Системы контроля доступа

Курс лекций

Санкт-Петербург 2011г.

Введение

В настоящем курсе рассматриваются аппаратные и программные средства современных систем контроля доступа, таких как считыватели магнитных карт, электронные таблетки (iButton), устройства бесконтактной радиочастотной идентификации, с примерами аппаратной и программной реализации этих устройств.

В курсе лабораторных работ, на основе изученных принципов построения устройств разрабатывается программное обеспечение для микропроцессора, реализующее рассматриваемые методы и алгоритмы.

Как правило рассматриваемые системы имеют в своем составе устройства сопряжения с компьютером, индикации, и клавиатуру. Мы рассмотрим работу и подключение распространенных алфавитно-цифровых и графических жидкокристаллических дисплеев к микропроцессору, а также предельно компактный способ подключения матричной клавиатуры к МП фирмы Intel.

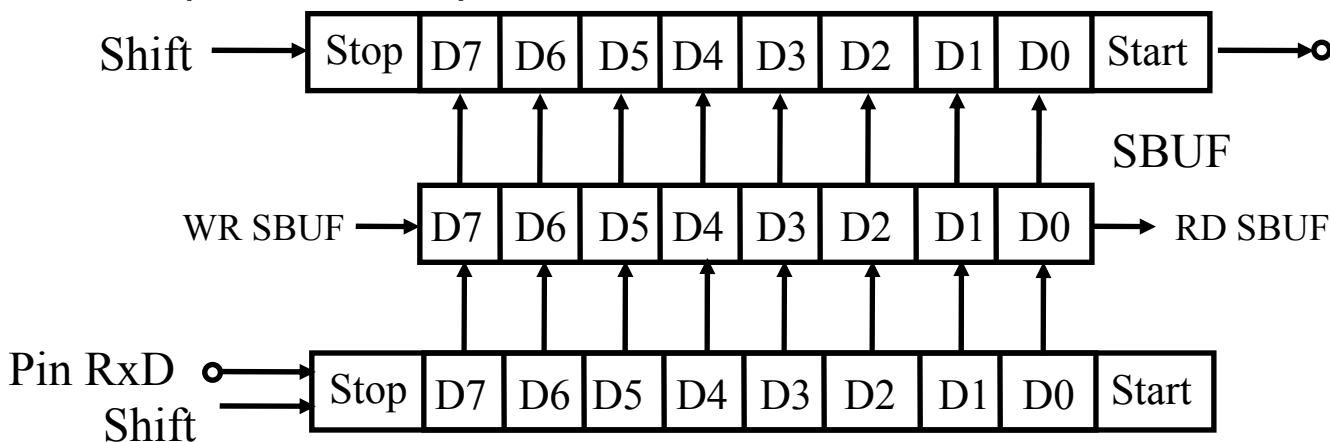
1. Последовательный порт UART

(Universal asynchronous Receiver / Transmitter)

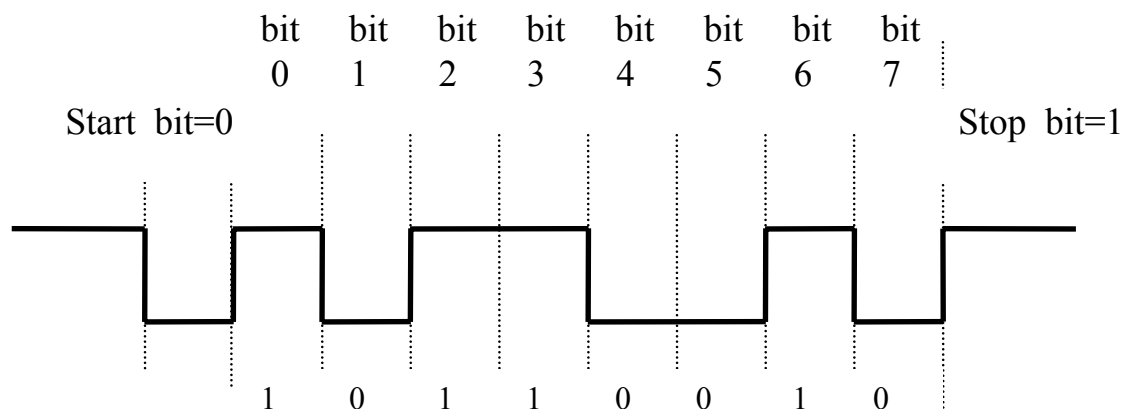
2. Локальная сеть на базе

RS485

- UART осуществляет прием и передачу информации в последовательном коде (младшим битом вперед) в полном дуплексном режиме, т. е. одновременно можно передавать и принимать данные.
- UART включает в себя сдвиговый регистр приёмника и передатчика а также буферный регистр SBUF , общий для приемника и передатчика.
- Запись байта в SBUF приводит к автоматической перезаписи его в сдвигающий регистр передатчика и инициирует передачу
- В режиме передачи UART преобразует параллельный код в последовательный и добавляет к нему стартовый и стоповый биты (Start , Stop) , которые служат для определения границ байта данных.



- При завершении приема данных в сдвиговый регистр приемника, его содержимое автоматически переписывается в SBUF и устанавливается флаг RI готовности приемника .
- Если к моменту окончания приема предыдущий байт не был считан из SBUF и флаг RI не будет сброшен , то следующий байт будет потерян.



- UART имеет 4 режима работы.
- Выбор режима осуществляется программированием разрядов регистра управления порта **SCON** (Serial Port Control Register).
- Передача данных инициируется записью байта в регистр **SBUF**.

Режим 0 – синхронный режим с постоянной скоростью обмена равной $1 / 12 \cdot F_{osc}$

$$BR_0 = F_{osc} / 12$$

- В режиме 0 ножка **RxD CPU** используется для ввода / вывода данных , ножка **TxD** – для вывода импульсов сдвига.

Режим 1 – асинхронный режим , передача 10 бит с программируемой скоростью обмена.

- 8 битов данных, стартовый бит = 0, стоп бит = 1.
- Скорость обмена задается таймером 1 или 2 .

$$BR_{1,3} = (2^{SMOD} / 32) * F_{ov}$$

где: **BR_{1,3}** – скорость передачи порта (бит/сек)

F_{ov} – частота переполнения таймера 1 или 2

SMOD-бит в регистре управления мощностью **PCON** – при значении 1 удваивает частоту UART, по умолчанию **SMOD** = 0.

- Таймер при этом может работать в любом режиме 0 , 1 или 2 . Прерывание при этом должно быть заблокировано. Однако наиболее удобно работать в режиме таймера с автоперезагрузкой

Для таймера 1: $BR_{1,3} = 2^{SMOD} * F_{osc} / 12 * 2 * 16 * (256 - TH1) ,$

$$TH1 = 256 - [2^{SMOD} * F_{osc} / (384 * BR_{1,3})]$$

SMOD-бит в регистре PCON – при значении 1 удваивает частоту UART, по умолчанию = 0.

Для таймера 2: $BR_{1,3} = F_{osc} / \{ 32 * [65536 - RCAP2H, RCAP2L] \}$
 $RCAP2H, RCAP2L = 65536 - F_{osc} / [32 * BR_{1,3}]$

Режим 2 – асинхронный режим , передача 11 бит с постоянной скоростью обмена.

- Передаются стартовый бит = 0, 8 битов данных , программируемый 9-й бит, стоп бит .
- Для повышения достоверности передачи путем контроля четности в 9-й бит может быть записан бит паритета из слова состояния программы PSW.0
- Скорость передачи равна $1 / 32$ или $1 / 64 F_{osc}$ в зависимости от значение бита SMOD =(1, 0)

$$BR_2 = F_{osc} * (2^{SMOD} / 64)$$

Режим 3 – асинхронный режим , передача 11 бит , скорость обмена , задается таймером 1 или 2 и рассчитывается как в режиме 1.

Структура Таймеров 1,2 при работе их в качестве генератора тактовых импульсов порта RS232 а также назначение битов регистра управления режимом порта SCON и таймеров T0,1-TMOD приведены ниже.

Для инициализации порта требуется установить их значения

Структура Таймера1

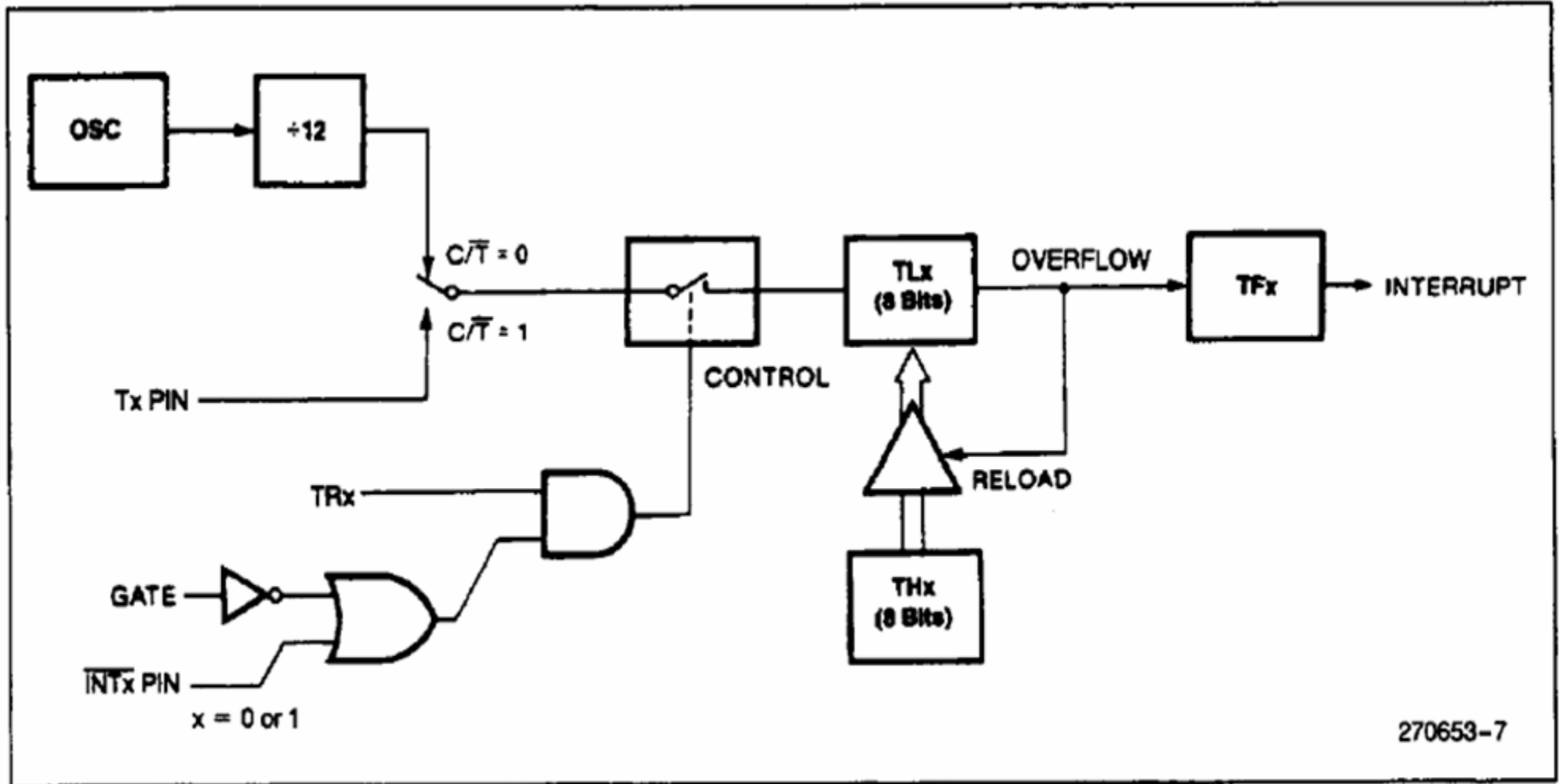
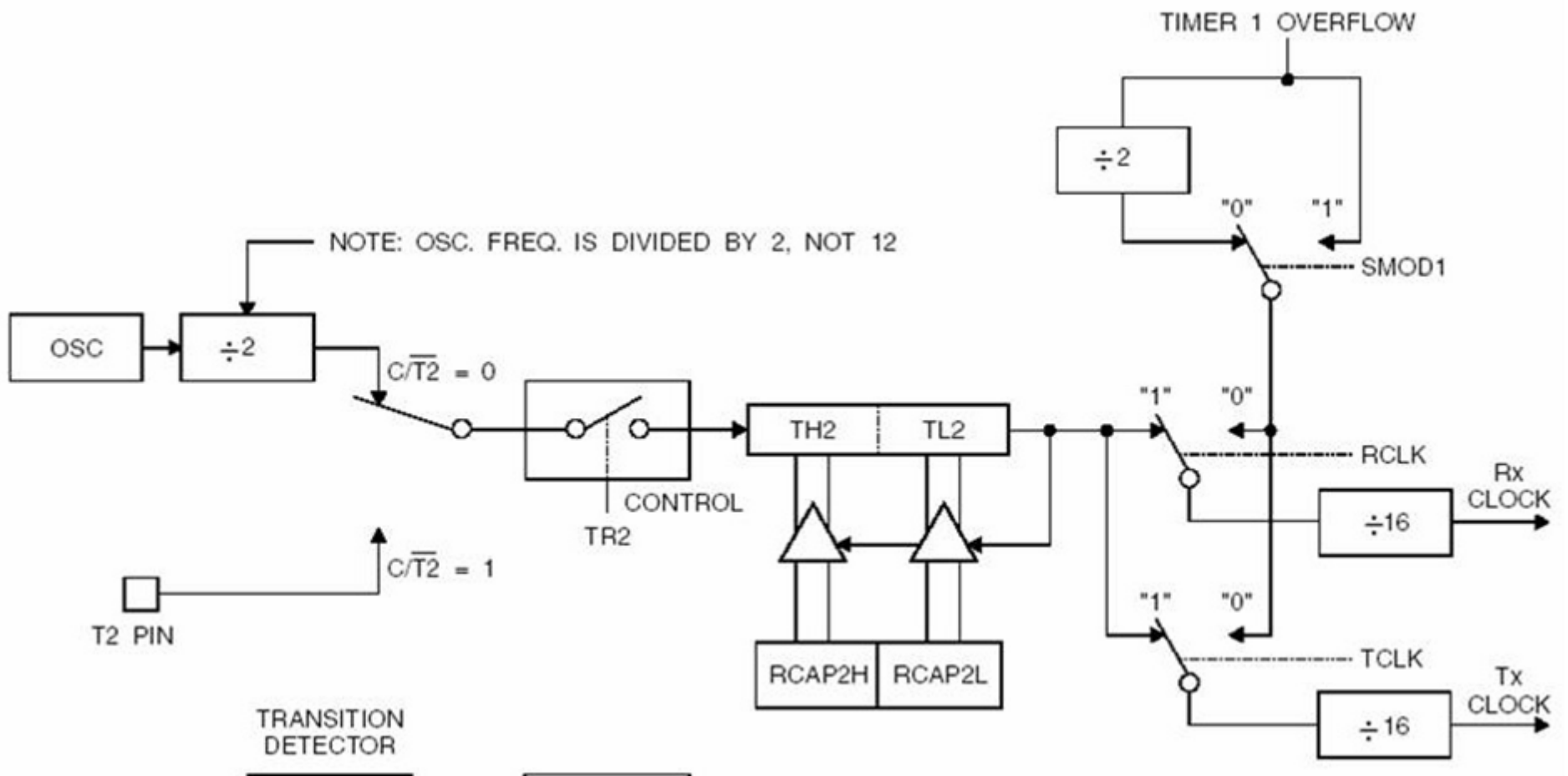


Figure 10. Timer/Counter 1 Mode 2: 8-Bit Auto-Reload

Структура Таймера2



Структура Таймера1

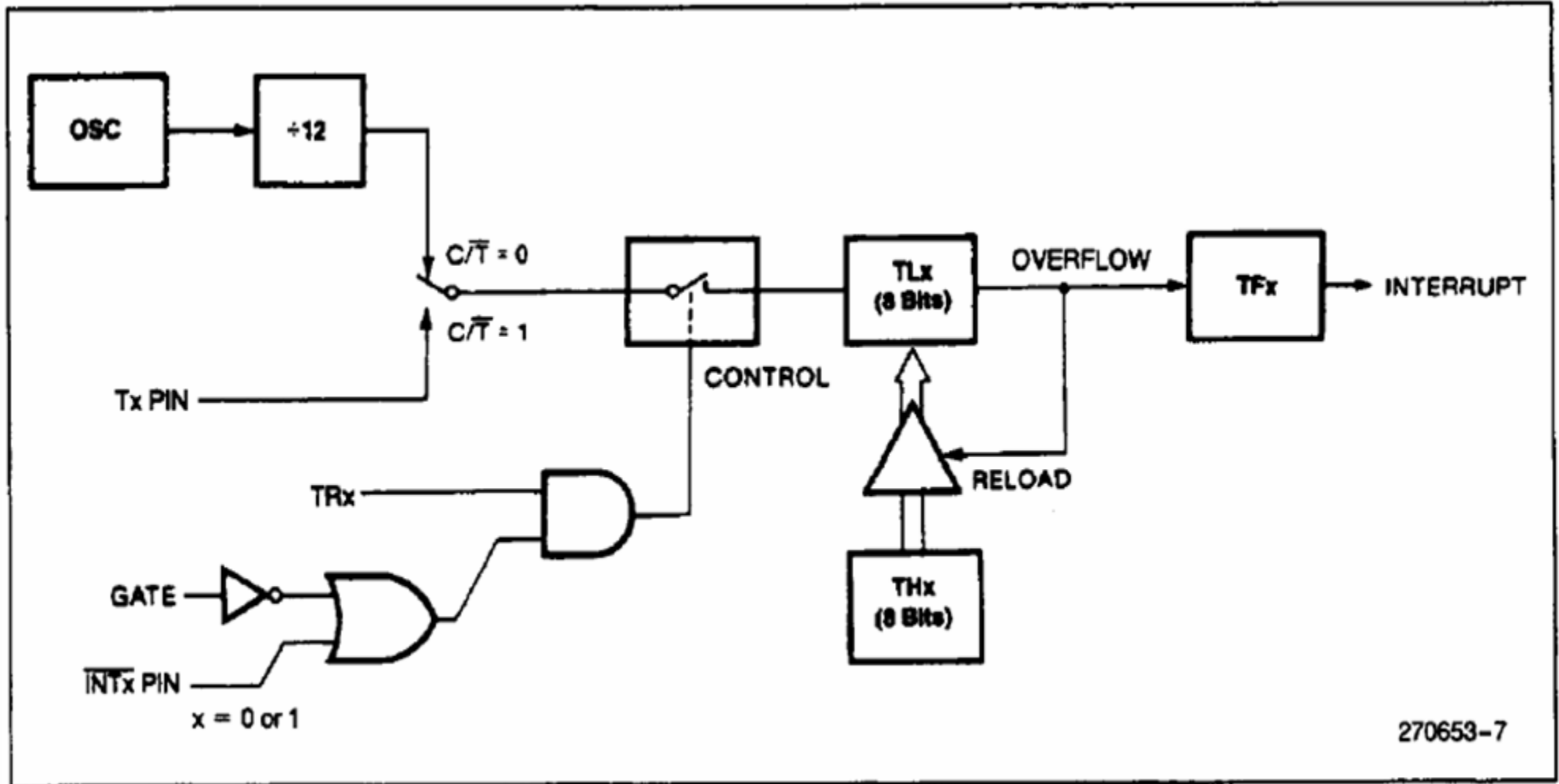
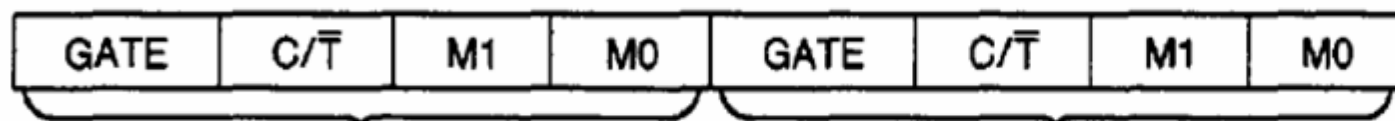


Figure 10. Timer/Counter 1 Mode 2: 8-Bit Auto-Reload

TMOD: TIMER/COUNTER MODE CONTROL REGISTER. NOT BIT ADDRESSABLE.



TIMER 1

TIMER 0

GATE When TR_x (in TCON) is set and GATE = 1, TIMER/COUNTER_x will run only while INT_x pin is high (hardware control). When GATE = 0, TIMER/COUNTER_x will run only while TR_x = 1 (software control).

C/ \bar{T} Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).

M1 Mode selector bit. (NOTE 1)

M0 Mode selector bit. (NOTE 1)

NOTE 1:

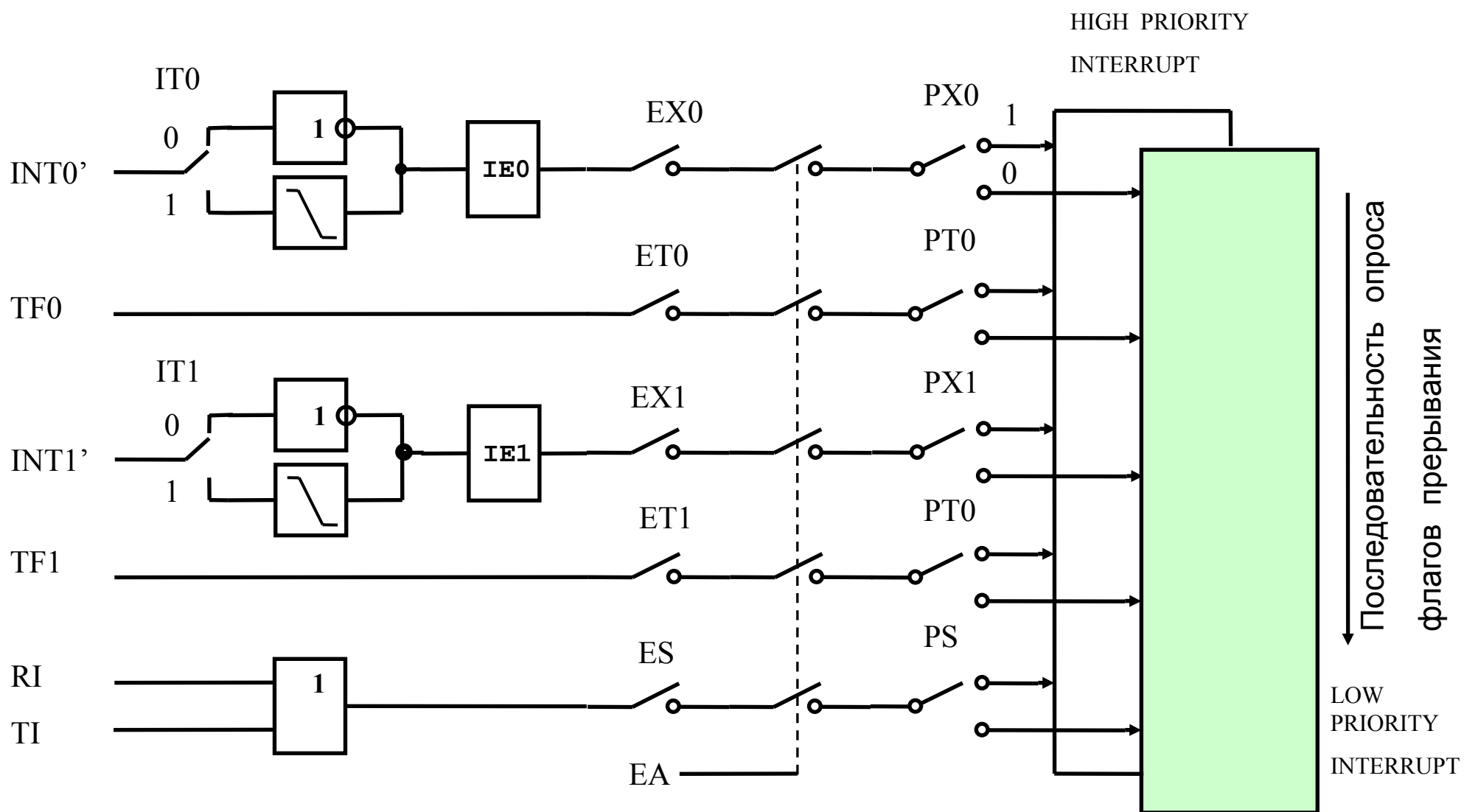
M1	M0	Operating Mode
0	0	0 13-bit Timer (MCS-48 compatible)
0	1	1 16-bit Timer/Counter
1	0	2 8-bit Auto-Reload Timer/Counter
1	1	3 (Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits, TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3 (Timer 1) Timer/Counter 1 stopped.

РЕГИСТР SCON УПРАВЛЕНИЯ ПОСЛЕДОВАТЕЛЬНОГО ПОРТА

Обоз.	Разряд	Наименование и назначение
RI	SCON.0	Флаг прерывания приемника. Устанавливается аппаратно при приеме байта в SBUF. Сбрасывается программно после обслуживания прерывания
TI	SCON.1	Флаг прерывания передатчика. Устанавливается аппаратно после завершения передачи байта. Сбрасывается программно после обслуживания прерывания
RB8	SCON.2	Приембита 8. В режиме 0 не используется. В режиме 1 при SM2=0 служит для занесения стоп-бита. В режимах 2 и 3 служит для приема 8-го бита данных
TB8	SCON.3	Передача бита 8. В режимах 2 и 3 устанавливается/сбрасывается программно. Предназначен для записи передаваемого 8-го бита данных
REN	SCON.4	Разряд разрешения приема. Устанавливается/сбрасывается программно для разрешения/запрета приема последовательных данных.
SM2	SCON.5	Разряд управления режимом приемопередатчика. Устанавливается программно для запрета приема сообщения, в котором девятый бит имеет значение 0.
SM1	SCON.6	Разряды управления режимом приемопередатчика: SM0 = 0, SM1=0 - синхронный режим; SM0 = 0, SM1=1 - синхронный режим, 8 бит, изменяемая скорость передачи;
SM0	SCON.7	SM0 = 1, SM1=0 - синхронный режим, 9 бит, фиксированная скорость передачи; SM0 = 1, SM1=1 - синхронный режим, 9 бит, изменяемая скорость передачи.

Регистр управления режимом **SCON**

- Содержит управляющие биты, задающие режим порта а также 9-й бит передаваемых и принимаемых данных TI и RI
- Во всех режимах передача инициируется загрузкой байта в SBUF.
- Прием разрешается если RI=0 и REN=1
- Флаг прерывания передатчика TI устанавливается аппаратно (автоматически) после окончания передачи байта . Программа перед выводом следующего байта , должна сбросить этот бит.
- Флаг прерывания приемника RI устанавливается аппаратно после приема байта . Сбрасывается программно для разрешения приема следующего байта.



- Уровень приоритета определяет очередность обслуживания когда сигналы запроса приходят одновременно.
- Программа обслуживания прерываний устройства с низким приоритетом может быть прервана прерыванием от другого устройством с более высоким приоритетом.

- Прерывания от последовательного порта инициируются флагами приемника или передатчика RI, TI.
- Эти флаги должны быть сброшены программно.
- Поскольку флаги прерываний приемника и передатчика RI, TI вызывают один вектор прерываний, то в программе обработки прерываний UART должны проверяться оба флага RI, TI.

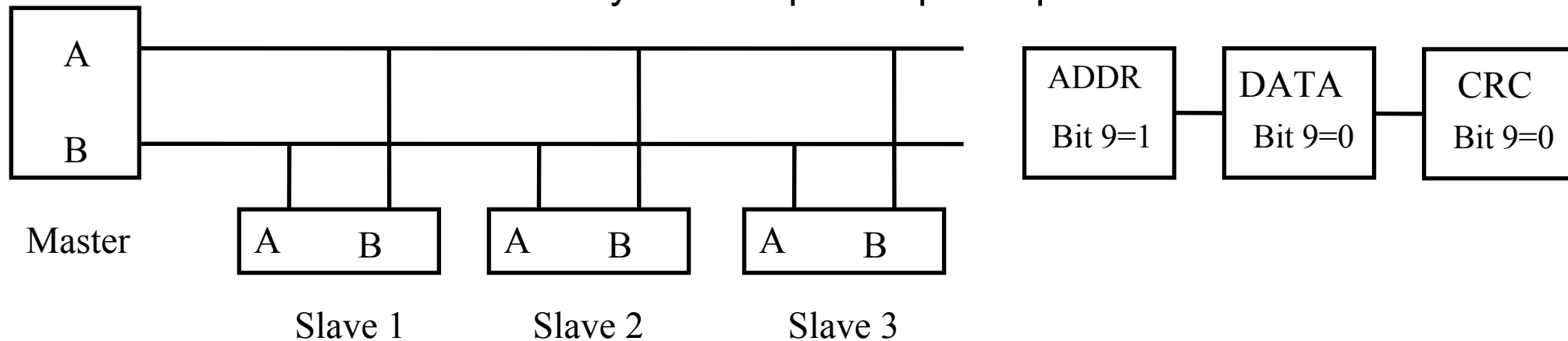
```
//-----
void rs_int ( void )  interrupt 4
{
unsigned char data_RS ;
if (TI)  TI=0;
if ( RI)  {  RI=0;
           data_RS=SBUF;           // Чтение с RS
           SBUF= data_RS;         //Эхо вывод на RS
           }
}
//-----
```

Пример программ работы с портом без прерываний

```
char getchar ()
{ while (!RI);
  RI = 0;
  return SBUF;
}
void putchar(char c)
{
  SBUF = c;
  while(!TI);
  TI=0;
}
//Вывод массива на RS
void putstr(unsigned char mass[])
{
  unsigned char i=0;
  while (mass[i]!=00)
    {putchar(mass[i]);
    i++;}
}
//-----
```

```
char data;
main()
{
  ini_RS();
  while(1)
    { putstr("ЭХО ВЫВОД С РС")
    data= getkey (); //RD SBUF
    putchar(data); //WR SBUF
    }
}
```

Работа UART в мультимикроконтроллерных системах



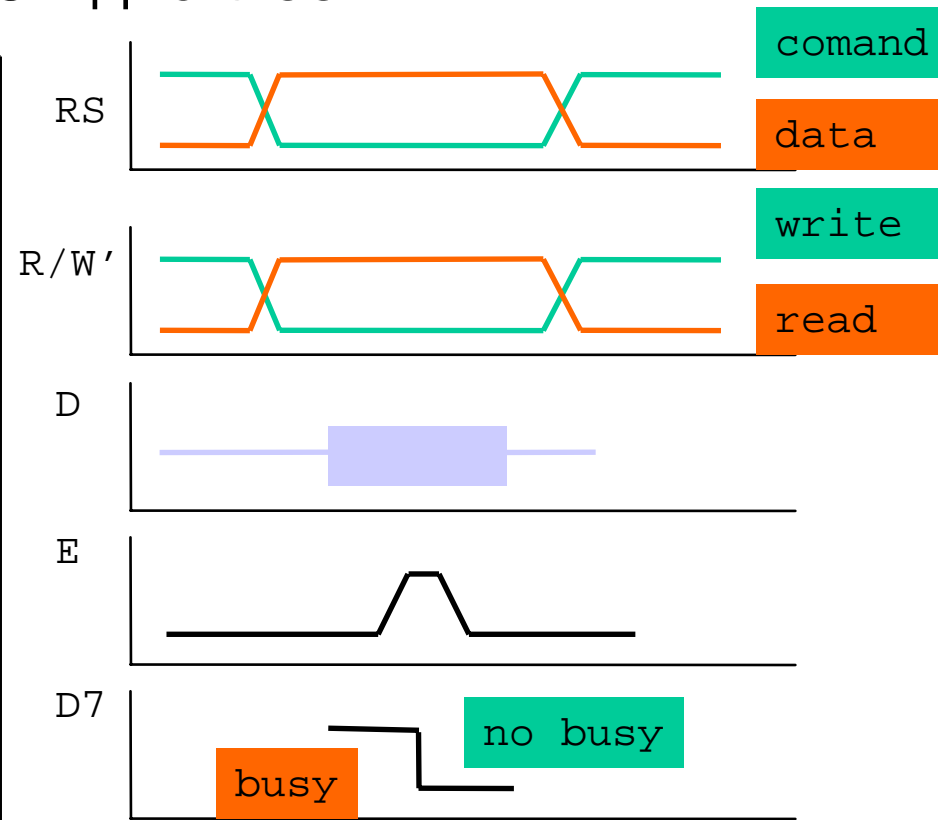
- В локальных сетях с магистральной архитектурой RS485 порты микроконтроллеров объединяются через общую магистраль.
- Для минимизации времени просушивания магистрали контроллерами, к которым нет обращения от мастера, используется бит SM2, запрещающий прием сообщения, в котором девятый бит имеет значение 0. Делается это следующим образом.
- Мастер передает в линию сообщение - сначала адрес, у которого 9-й бит =1, потом данные, у которых 9-й бит =0.
- При этом у всех приемников бит SM2 установлен, поэтому они принимают только код адреса.
- Контроллер, адрес которого совпал с адресом на линии сбрасывает бит SM2, разрешая прием данных (с 9-м битом равным нулю).
- После завершения приема данных восстанавливает значение бита SM2=1.
- Таким образом контроллеры, к которым нет обращения входят в прерывание и принимают только адрес на линии. Без использования этого метода каждый контроллер прослушивал бы все сообщения на линии.

ЖИДКОКРИСТАЛИЧЕСКИЕ АЛФАВИТНОЦИФРОВЫЕ ДИСПЛЕИ

Winstar, Bolemin ...

Управляющие сигналы LCD дисплеев

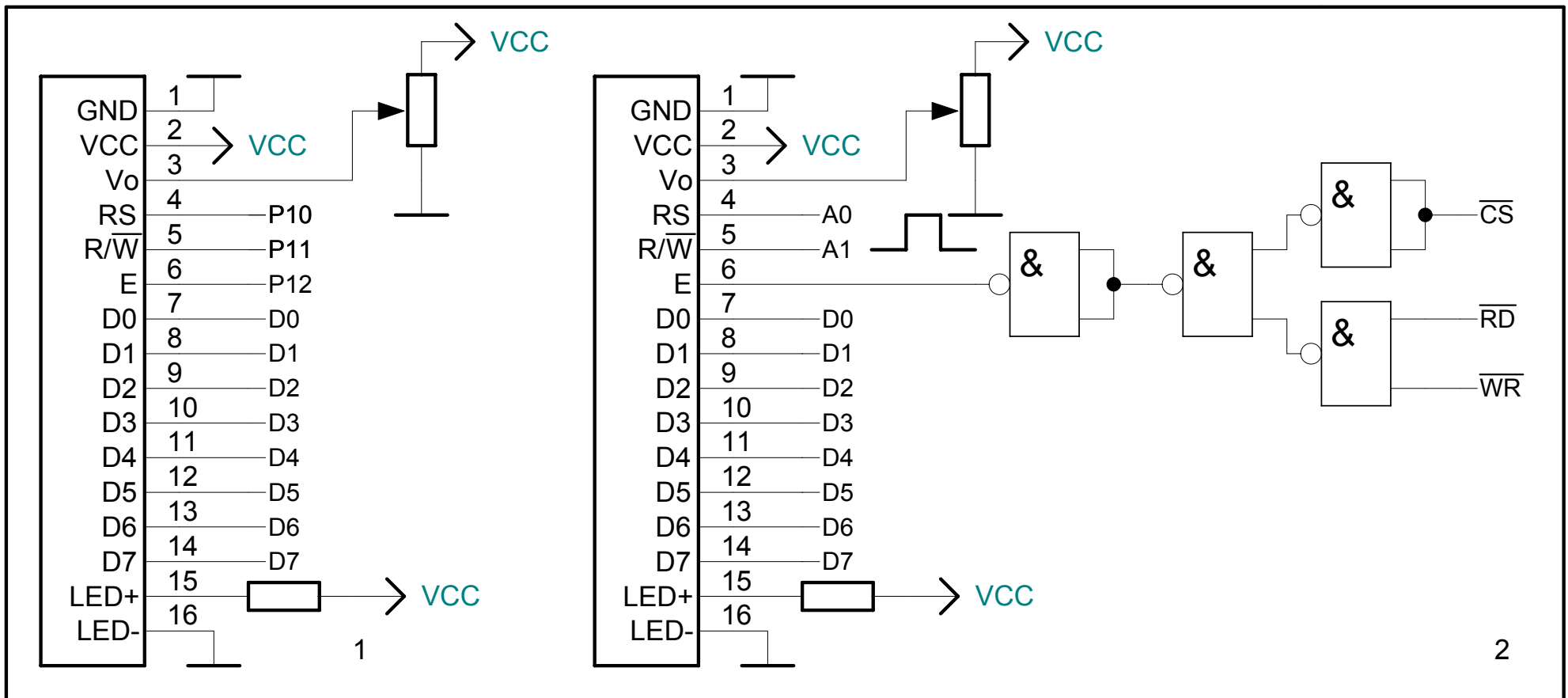
Pin	Сигнал	Назначение
1	GND	0 V
2	Vdd	Установка контрастности (0,3...0,8V}
3	VCC	+5 V
4	RS	Выбор регистра 1- регистр данных 0- регистр команд
5	R/WR'	1 / 0 -> Вывод / ввод данных в дисплей
6	E	Сигнал разрешения записи / чтения
7...14	D0...D7	Шина данных



ОПЕРАЦИЯ	RS	R/WR'
Запись команд	0	0
Чтение Busy, адреса	0	1
Запись данных	1	0
Чтение данных	0	1

- Запись команд- устанавливаются режимы работы дисплея, адрес курсора
- Чтение Busy-определение момента завершения внутренних операций

Подключение LCD к CPU



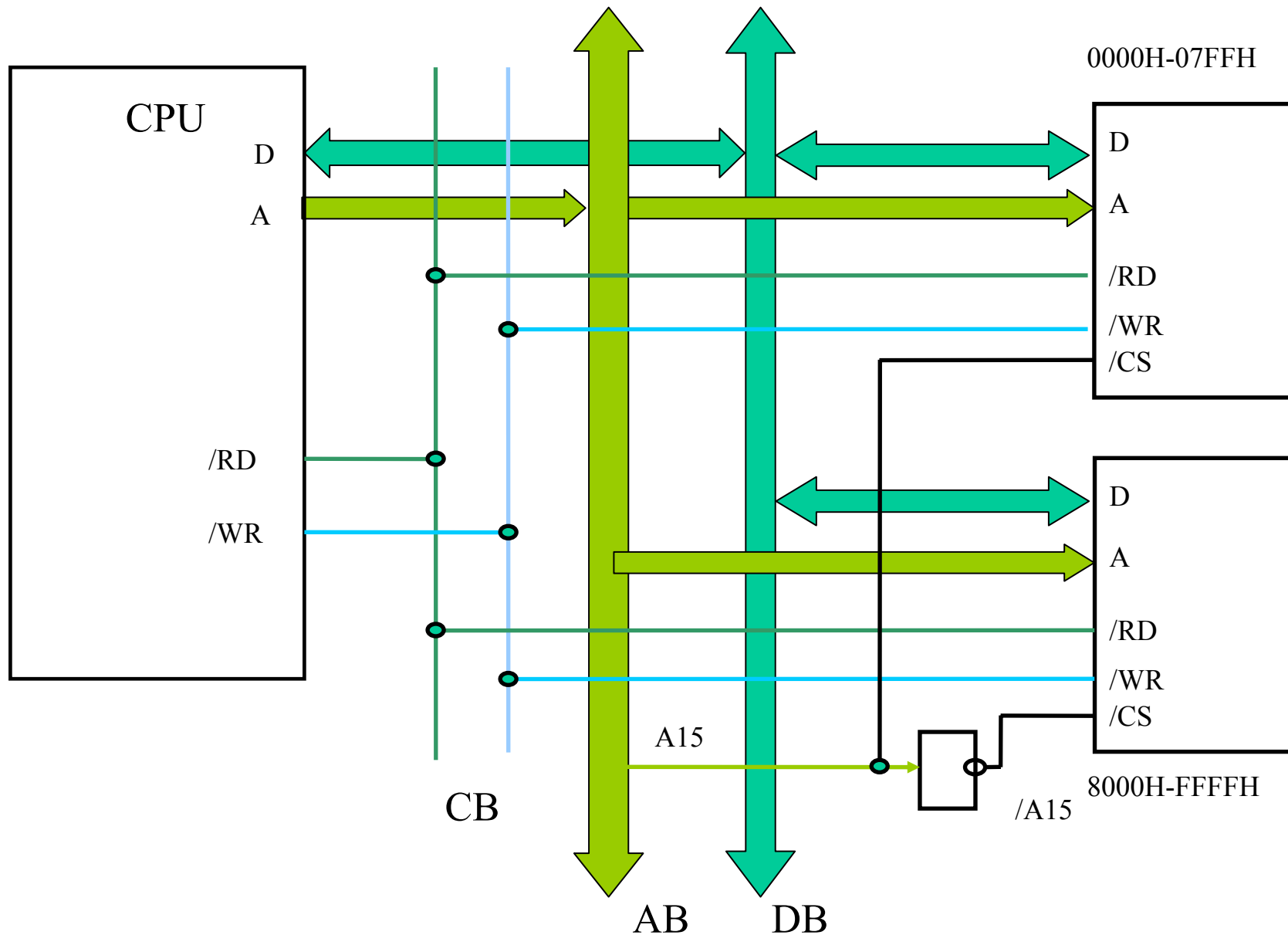
1

2

1. Побитное управление каждым из сигналов

2. Подключение к шине процессора – сигналы RS, R/~w, E формируются при выполнении одной команды – вывода/ввода во внешнюю память XBYTE[adr]=dat

Подключение внешних устройств к CPU



ИНИЦИАЛИЗАЦИЯ ДИСПЛЕЯ

RS	R/WR	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	1	x	x	x	x

Сброс дисплея, Пауза 30 mS 3 раза

RS	R/WR	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	DL	N	F	x	x

Устан 8/4битный обмен (DL), кол строк (N), точек на знак (F)

RS	R/WR	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	1

Стирание дисплея

RS	R/WR	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	1	I/D	S

Устан. реж. Ввода-Инкр/Декрем. I/DS
Замена/вставка S

RS	R/WR	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	1	D	C	B

Устан. реж. Вкл./выкл. дисплея,
курсора

DL-1/0- Обмен 8/4 бит

N-1/0 - 2/1 строка

F -1/0 - 10/7 dot

I/D -1/0 - Инкремент/Декремент поз курс

S -1/0 - Замена/вставка

D -1/0 - Вкл./выкл. дисплея

C -1/0 - Вкл./выкл. курсора

B -1/0 - Вкл./выкл. мигания курсора

Таблица управляющих кодов дисплея

RS	D7	D6	D5	D4	D3	D2	D1	D0	КОД	ФУНКЦИЯ, ОПИСАНИЕ
0	0	0	0	0	0	0	0	1	1	Очистка экрана и буфера. Установка курсора в позицию 0
0	0	0	0	0	0	0	1	x	2	Установка курсора в позицию 0. Содержимое DDRAM неизменно.
0	0	0	0	0	0	1	I/D	S	4	Установка режима ввода
0	0	0	0	0	0	1	0	0	4	Замена влево
0	0	0	0	0	0	1	0	1	5	Вставка влево
0	0	0	0	0	0	1	1	0	6	Замена вправо
0	0	0	0	0	0	1	1	1	7	Вставка вправо
0	0	0	0	0	1	D	C	B	8	Вкл/выключение дисплея
0	0	0	0	0	1	0	x	x	8	Дисплей выключен.(DDRAM не стерт)
0	0	0	0	0	1	1	0	0	12	Дисплей включен. Курсор включен
0	0	0	0	0	1	1	0	1	13	Дисплей вкл. Курс выключен. Мигающий символ
0	0	0	0	0	1	1	1	0	14	Дисплей включен. Курсор включен
0	0	0	0	0	1	1	1	1	15	Дисплей курсор включены. Мигающий символ
0	0	0	0	1	S/C	R/L	x	x	16	Позиционирование курсора/дисплея
0	0	0	0	1	0	0	x	x	16	Перемещение курсора влево
0	0	0	0	1	0	1	x	x	20	Перемещение курсора вправо
0	0	0	0	1	1	0	x	x	24	Перемещение экрана влево
0	0	0	0	1	1	1	x	x	28	Перемещение экрана вправо
0	0	0	1	DL	N	F	x	x	32	Установка режима
0	0	0	1	1	0	0	x	x	48	Режим_1 строка_x 16 символов_(7x5)точек
0	0	0	1	1	0	1	x	x	52	Режим_1 строка_x 16 символов_(10x5)точек
0	0	0	1	1	1	0	x	x	56	Режим_2 строки_x 16 символов_(7x5)точек
0	0	0	1	1	1	1	x	x	60	Режим_2 строки_x 16 символов_(10x5)точек
0	0	1	Адрес CGRAM 0..63							Установка адреса з накопителя
0	1	Адрес DDRAM 0..127							Установка адреса памяти данных	

ЖИДКОКРИСТАЛИЧЕСКИЕ
ГРАФИЧЕСКИЕ
ДИСПЛЕИ

WG12864, PG12864

с контроллером

S6B0108

фирм Winstar , Powertip

ГРАФИЧЕСКИЕ ДИСПЛЕИ

WG12864, PG12864 с контроллером S6B0108

Для построения графических дисплеев используется большая по сравнению с алфавитно-цифровыми дисплеями номенклатура контроллеров. Одним из наиболее распространенных контроллеров является контроллер S6B0108 фирмы Samsung Electronics. В графических ЖКИ с разрешением 128x64 используется 2 контроллера, управляющие 1, и 2-ой половиной ЖКИ экрана размером 64x64 точек

Сигналы шины управления S6B0108

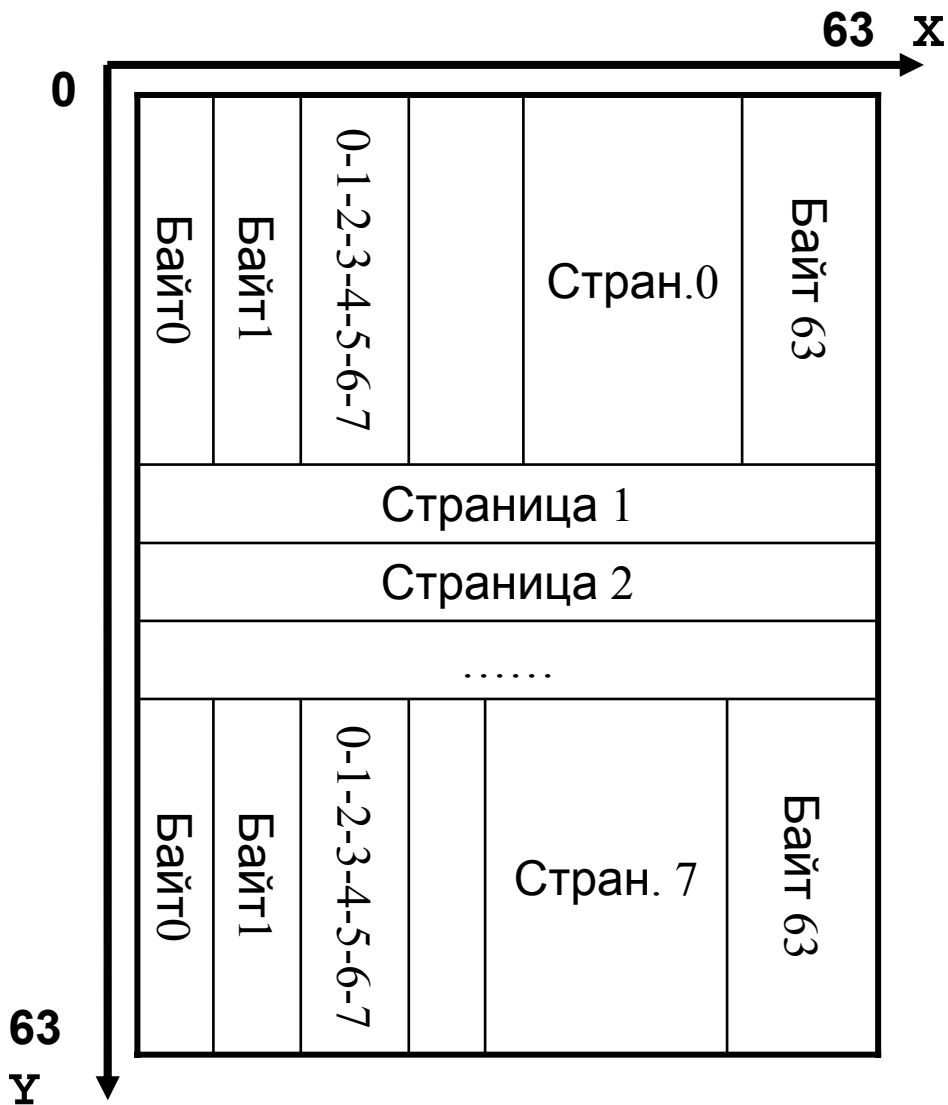
Сигнал	Функция
D0...D7	Двунаправленные линии данных. Используются для записи/чтения команд или данных
D/I#	Вход выбора регистра («1» — регистр данных, «0» — регистр команд)
R/W#	Вход направления передачи («1» — чтение из S6B0108, «0» — запись в S6B0108)
E	Вход строба разрешения
# CS1	Входной сигнал выбора кристалла 1 контроллера
# CS2	Входной сигнал выбора кристалла 2 контроллера
# Reset	Вход сброса S6B0108 в начальное состояние

С точки зрения программиста ЖКИ имеет 2 регистра - **регистр данных и команд**. При сигнале на входе **D/I# = 0** байт с шины данных **D0...D7** записывается в регистр данных, при **D/I# = 1** - в регистр команд. Направление передачи задается сигналом **R/W#**.

Функциональное назначение интерфейсных выводов S6B0108

D /#I	R/ #WR	Операция
0	0	Запись команды в регистр команд S6B0108
0	1	Чтение слова состояния из S6B0108
1	0	Запись байта в регистр данных с последующей передачей в ОЗУ и автоинкрементом адреса
1	1	Чтение байта из регистра данных ОЗУ последующим автоинкрементом адреса

Структура видеопамяти контроллера S6B0108



Экран ЖКИ размером 64x64 точек управляется встроенной видео RAM объемом $64*64=4096$ бит =512 байт. Эта память разбита на 8 страниц по 64 байта. Координаты выводимой на экран точки по X, Y задаются 6 битным числом в пределах 0..63.

1. Координата по оси X задается командой **“Установки адреса”** – **01xxxxxx**, где xxxxxx – 6-и битное значение по оси X.

2. Для задания координаты по оси Y по команде **“Установить страницу”**- **10111rrr** , где **rrr=Nстраницы = Y/8**.

Для этого в регистр команд записывается значение **0xB8 + (Y/8)** .

3. Третьим действием в **регистр данных** записывается байт в котором бит соответствующий подсвечиваемой точке=1, остальные биты =0. Номер подсвечиваемого бита **Nbit = Y% 8**.

Значение байта с таким битом **Dbyte=2^Nbit**

0

Y		Нстр= =Y/8	Nbit = =Y%8	Значение байта Дбайт=2^ Nбита
0		0	0	1
1		0	1	2
2		0	2	4
3		0	3	8
4		0	4	16
5		0	5	32
6		0	6	64
7		0	7	128
8		1	0	1
9	■	1	1	2
10	*	1	2	4
11		1	3	8
12		1	4	16
13		1	5	32
14		1	6	64
15		1	7	128
--	--	---	---	---
63		7	7	128

63

Y

$$Nbit = Y \% 8$$

$$Dbyte = 2^{Nbit}$$

Этот байт Dbyte удобно получить табличным способом

$Dbyte = tab_n_poz [y \% 8]$; где массив
 $tab_n_poz[8] = \{1, 2, 4, 8, 16, 32, 64, 128\}$;

После записи в регистр данных происходит автоматический автоинкремент адреса X.

Система команд контроллера S6B0108

Для инициализации дисплея требуется выполнить команду "Включить дисплей" и стереть его, записав в видео RAM нули.

КОМАНДА	Код	ФУНКЦИЯ
Отключить дисплей	00111110 3E	Отключает отображение содержимого памяти от ЖК матрицы.
Включить дисплей	00111111 3F	Включает отображение содержимого памяти к ЖКИ матрице.
Установить адрес	01xxxxxx	Задаёт адрес внутри страницы xx=0–63, к которому будет происходить очередное обращение
Установить страницу	1011pppp	Задаёт страницу памяти p=0...7, куда будет осуществляться запись или чтение
Задать начальную строку скроллинга	11ssssss	Задаёт номер строки памяти ss=0–63, которая будет отображаться в верхней строке секции ЖКИ матрицы (используется для скроллинга).

Вывод символов в графическом ЖКИ

Для вывода текстового сообщения на ЖКИ в памяти процессора записывается таблица символов в виде двумерного массива, в котором на строчке с номером ASCII кода символа располагаются код строчек повернутого на 90° символа

```
code unsigned char font1[256][5]= { // HEX DEC ASCII
```

```
    {0x00,0x00,0x00,0x00,0x00}, // 0    0
```

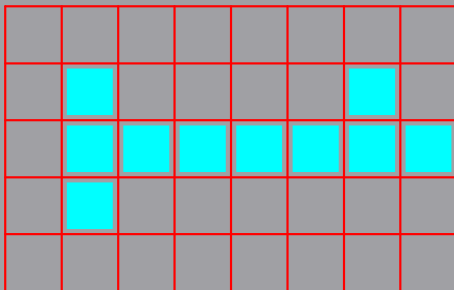
```
    .....
```

```
    {0x3E,0x51,0x49,0x45,0x3E}, // 30  48  '0'
```

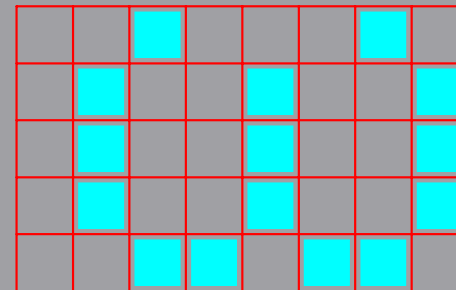
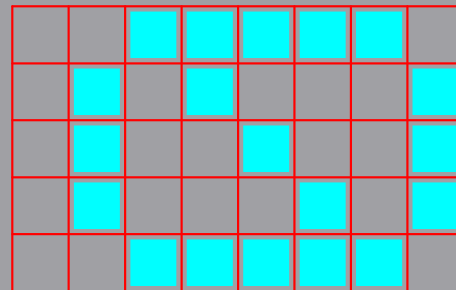
```
    {0x00,0x42,0x7F,0x40,0x00}, // 31  49  '1'
```

Пример шрифта символа “1”, “0” размером 5x7 с правым поворотом- 5 байт на СИМВОЛ.

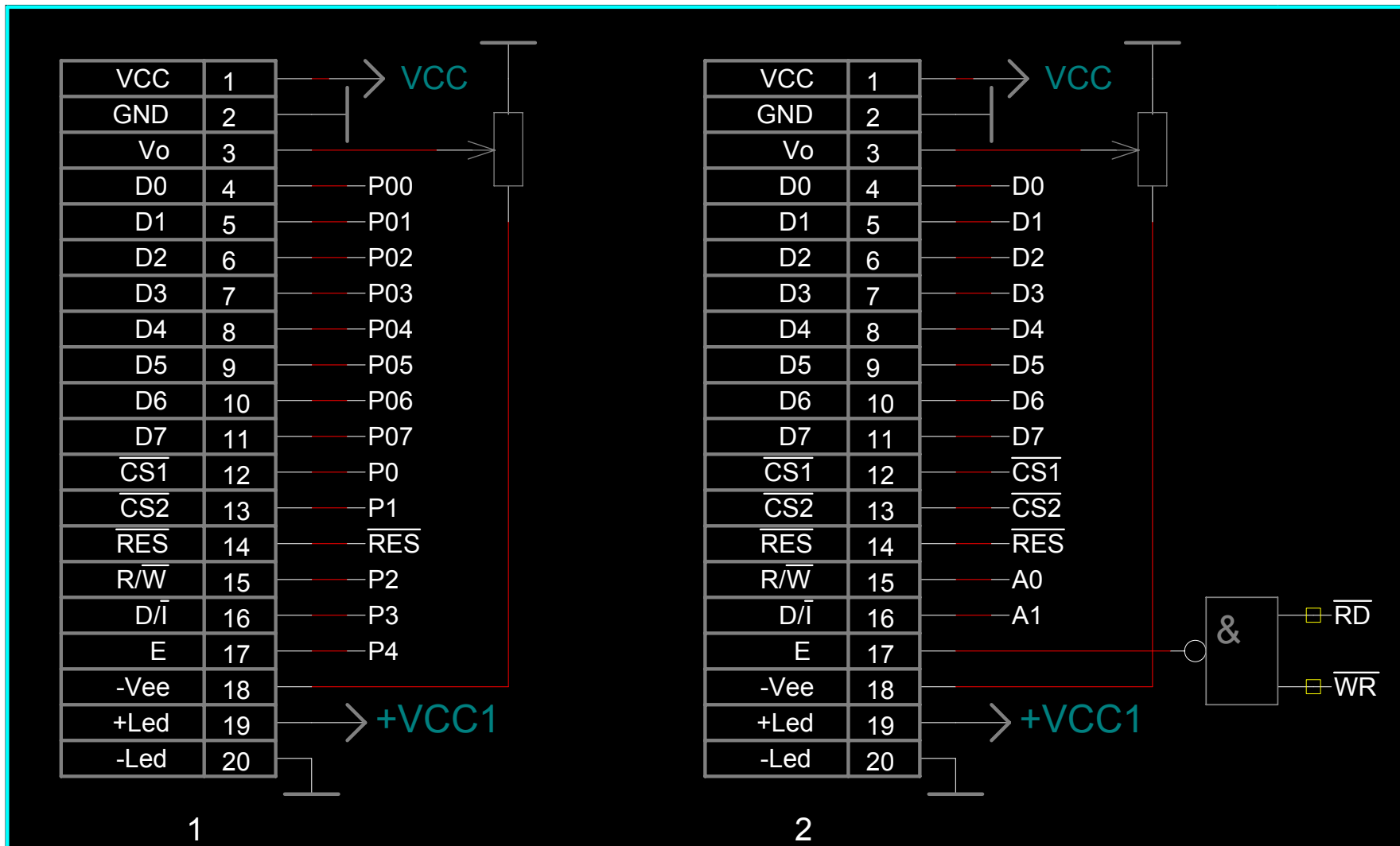
{ 0x00,0x42,0x7F,0x40,0x00 }



{ 0x3E,0x51,0x49,0x45,0x3E }



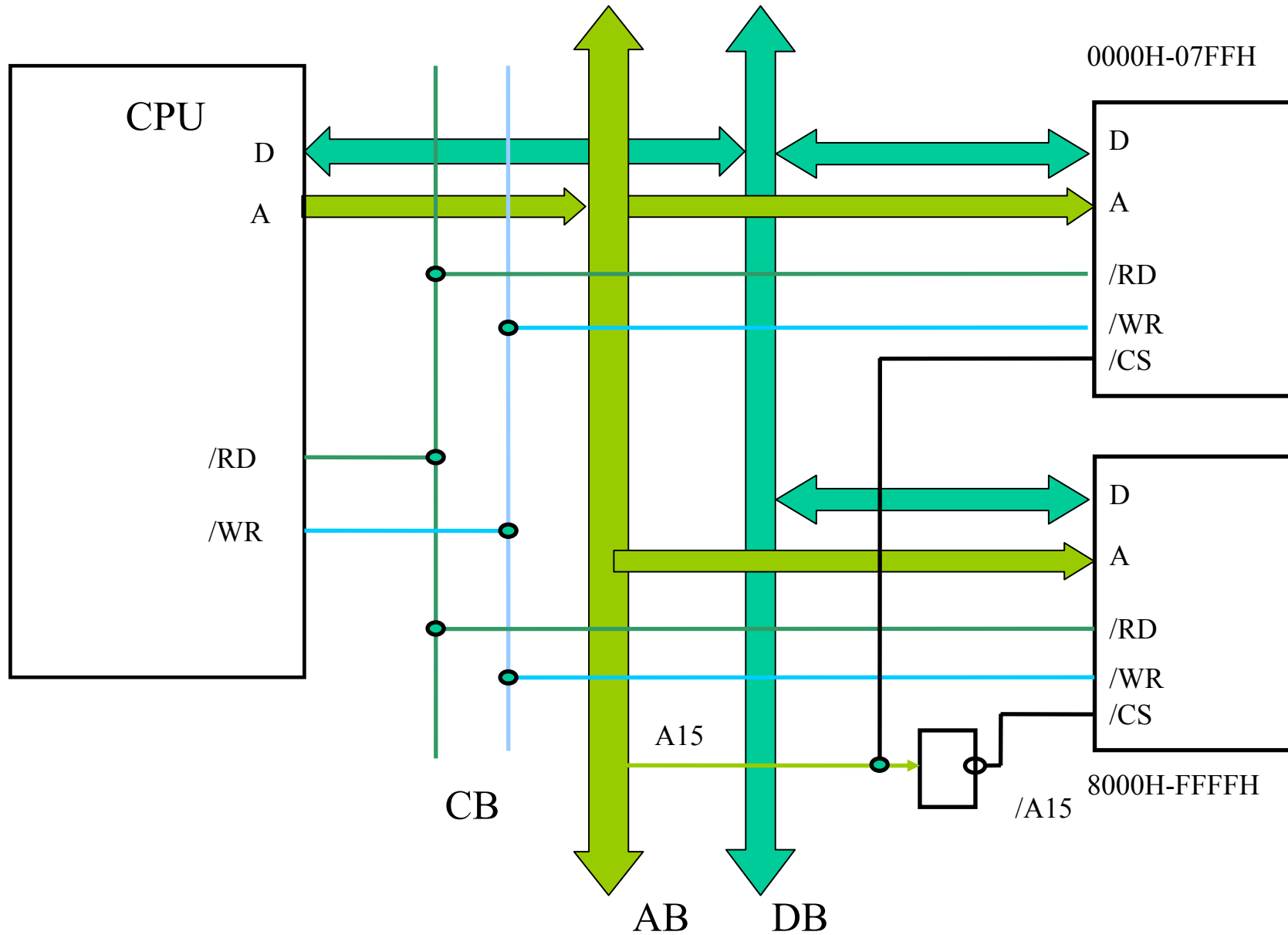
Подключение LCD к CPU



1. Побитное управление каждым из сигналов

2. Подключение к шине процессора – сигналы RS, R/~W, E формируются при выполнении одной команды – вывода/ввода во внешнюю память

Подключение внешних устройств к CPU



Временные диаграммы сигналов контроллера S6B0108

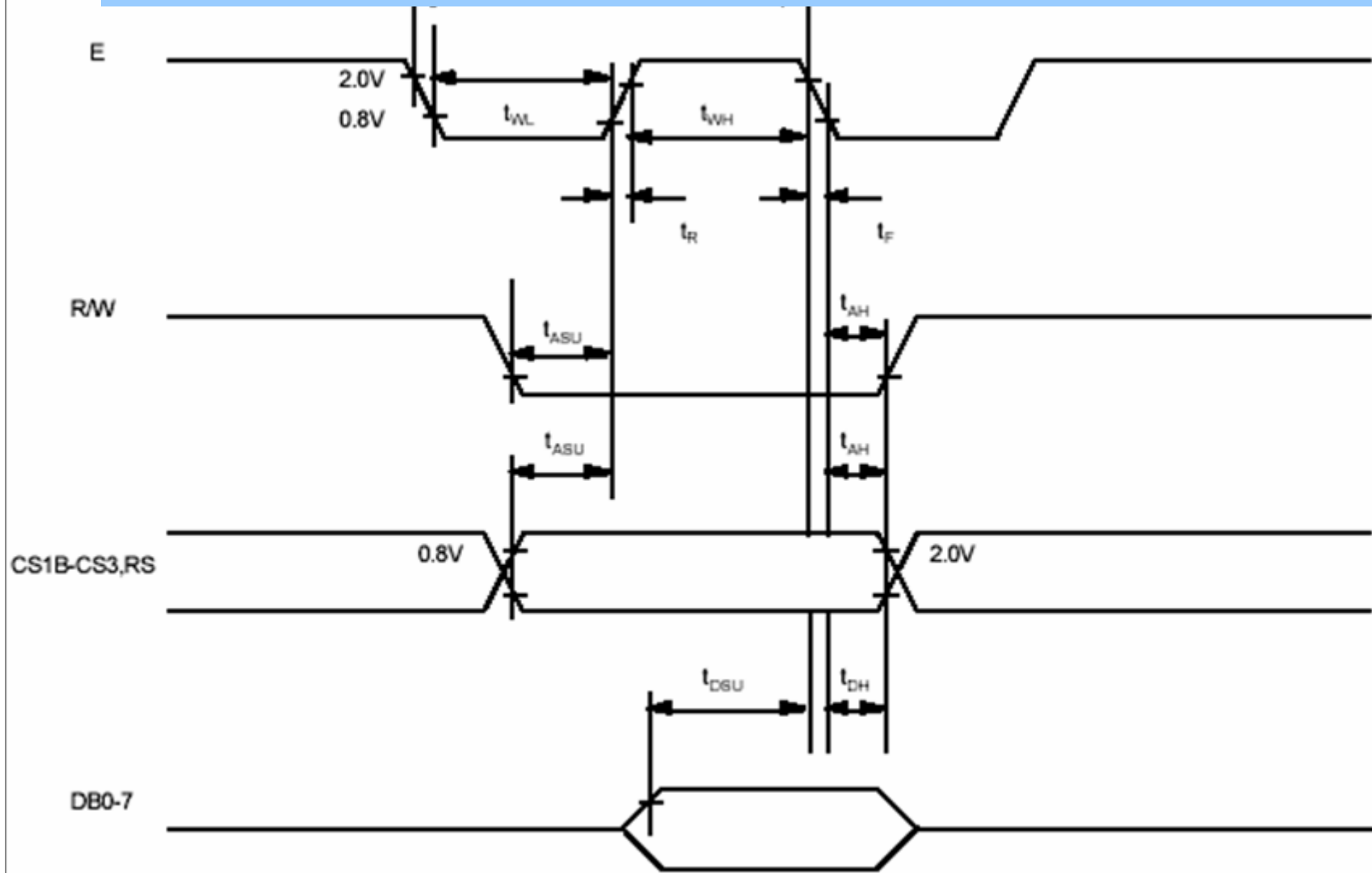
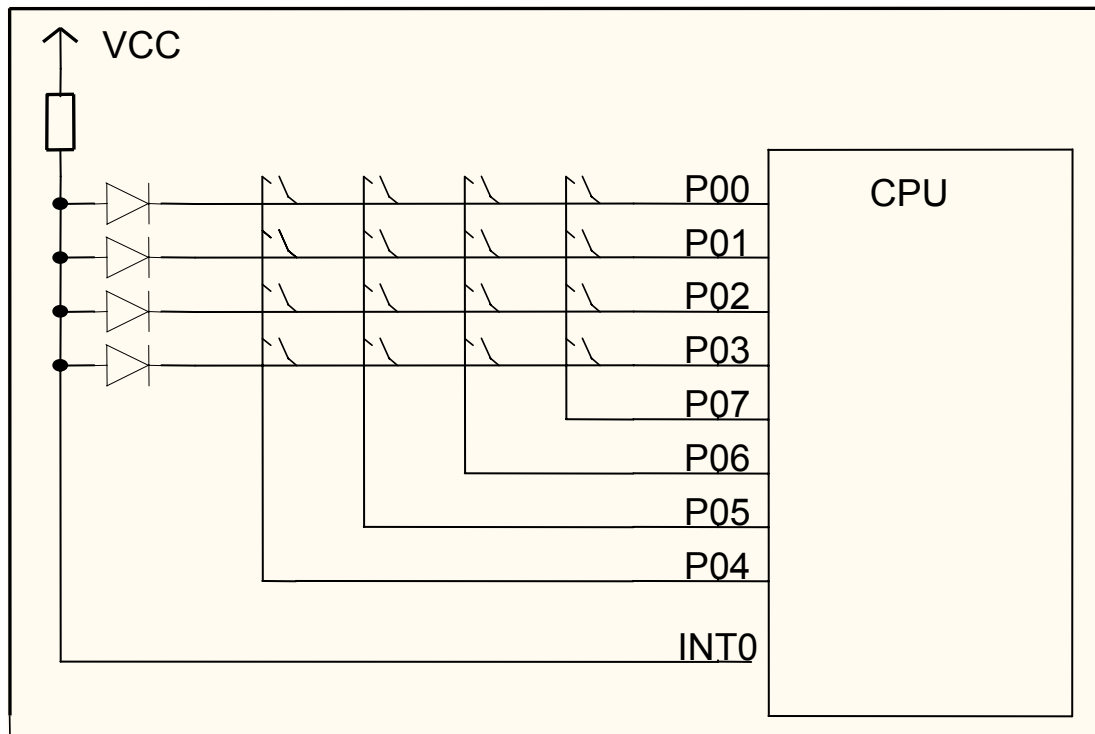


Fig 3. MPU write timing

КЛАВИАТУРА В МП СИСТЕМАХ

Интеловский алгоритм опроса клавиатуры



1	2	3	A
4	5	6	B
7	8	9	C
*	0	#	D

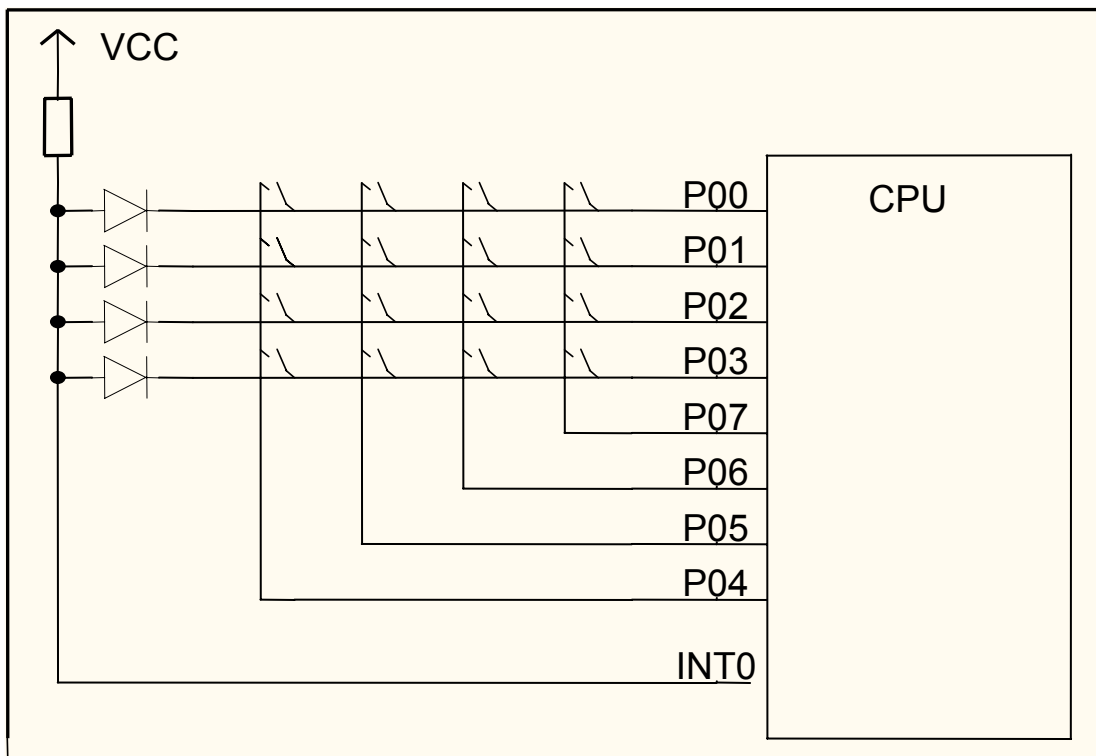
Рис. 1

Для уменьшения выводов клавиатуры кнопки клавиатуры соединяются в виде матрицы.

- Клавиши условно нумеруются—например слева направо, сверху—вниз.
- Порядковый номер клавиши $N_{key} = N_{col} + (N_{row} - 1) * 4$, где: N_{col} и N_{row} — это номер колонки и строки на которой располагается кнопка N_{key} .
- Преобразование номера кнопки N_{key} в соответствующий ASCII код Cod_key можно произвести по таблице кодировки,

`Code_key = Tabkey[Nkey];` например для телефонной клавиатуры Рис.1

`Tab_key = {'1', '2', '3', 'A', '4', '5', '6', 'B', '7', '8', '9', 'C', '*', '0', '#', 'D'}`



1. P0= 00001111; -0 на колонки
2. Вход в прерывание. Чтение активной строки
3. P0= 11110000; 0 на строки
4. Чтение активной колонки
5. Вычисление N клавиши

$$N_{key} = N_{col} + (N_{row} - 1) * 4;$$
 Кодирование клавиши
6. Code_key=Tabkey[Nkey];
7. Выход из прерывания

При инициализации на все колонки матрицы клавиатуры выводятся нули, на строки - единицы.

- При нажатии на любую клавишу на катод диода попадает «0» - и на INT, будет потенциал 0.7 V. Процессор входит в прерывание. На одном из разрядов P00..P03 появляется 0. Это активная строка.

- Далее на все колонки матрицы клавиатуры выводятся единицы, на строки - нули и читается активная колонка

- Для устранения дребезга контактов после вхождения в прерывание, оно запрещается на время 50..100 mS

Для некоторых типов процессоров переключение порта с режима ввода на вывод требует дополнительных команд (например для МП SIGNAL).

В таком случае можно предложить такой алгоритм работы:

- Порты строк установить в режим чтения, порты колонок – в режим вывода.
- Первоначально на все колонки подать нули.
- После нажатия на любую кнопку и вхождения в прерывание прочитать строки и определить номер активной строки.
- Далее поочередно убирать нули с колонок, заменяя их единицами до тех пор пока не пропадет ноль на активной строке, т.е. строки не перейдут в состояние единицы.
- Активной является та колонка, замена значения которой с единицы на ноль приведет к чтению единиц на всех строках.
- Далее, как и в предыдущем случае по номеру активной строки и колонки вычисляется номер клавиши и ее код.

Считыватели магнитных карт

Общие сведения

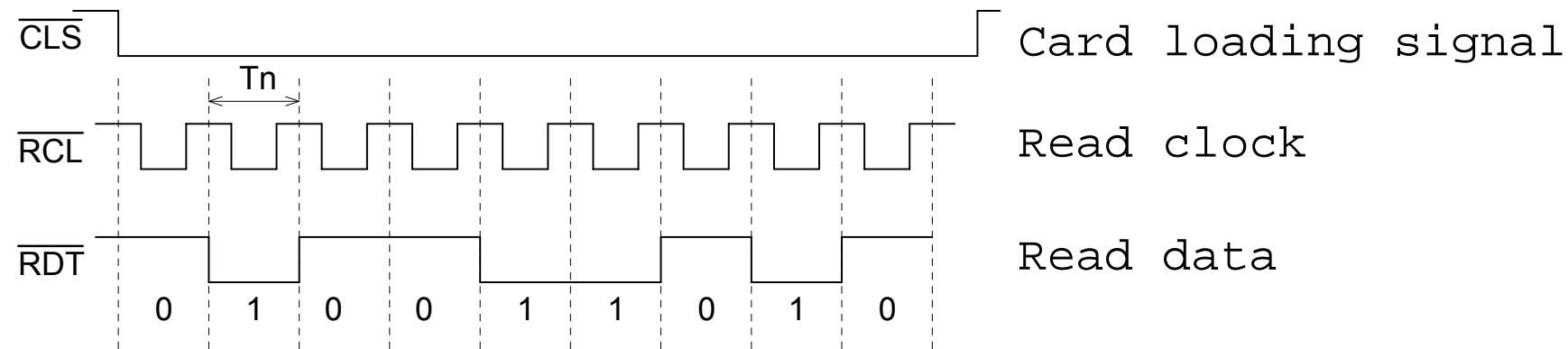
Магнитная карта-пластиковая карта с нанесенной магнитной полосой.

На полосе может быть 1..3 дорожек записи. Характеристики записи определяются стандартами ANSI, ISO7811.

1-я дорожка в банковских карт обычно хранит имя владельца, номер карты. 2-я дорожка- плотность записи 75 bpi, 39 символов по 5 бит. 3-я дорожка-используется редко, для перезаписи информации на карте.

ISO 7811		
1 дорожка	2 дорожка	3 дорожка
210 bpi	75 bpi	210 bpi
79 СИМВОЛОВ по 7 бит	40 СИМВОЛОВ по 5 бит	107 СИМВОЛОВ по 5 бит

Сигналы и временные диаграммы считывателя KDR1321



Плотность записи	75 BPI		210 BPI	
	Скорость протяжки	10 см/сек	120 см/сек	10 см/сек
Период бит.инт. T_n	3,38 mS	282 μ S	1.2 mS	100 μ S

Сигнал CLS-присутствие карты в считывателе

RDT-последовательные данные -инверсные

RCL-синхроимпульсы

Формат данных считывателя (2-я дорожка)

№симв		D0	D1	D2	D3	P
1	1 старт символ 0xB (3B=«;»)	1	1	0	1	1
2	37 СИМВОЛОВ ДАННЫХ P1...P39-бит четности строк					P2
.						P3
.						.
.						.
.						.
38						P38
39		Стоп символ 0xF (30+F='?')	1	1	1	1
40	LRC – бит паритета по столбцам	PC0	PC1	PC2	PC3	PP

Вначале полосы записывается последовательность нулей для калибровки считывателя-для измерения длительности битового интервала при данной скорости проводки карты.

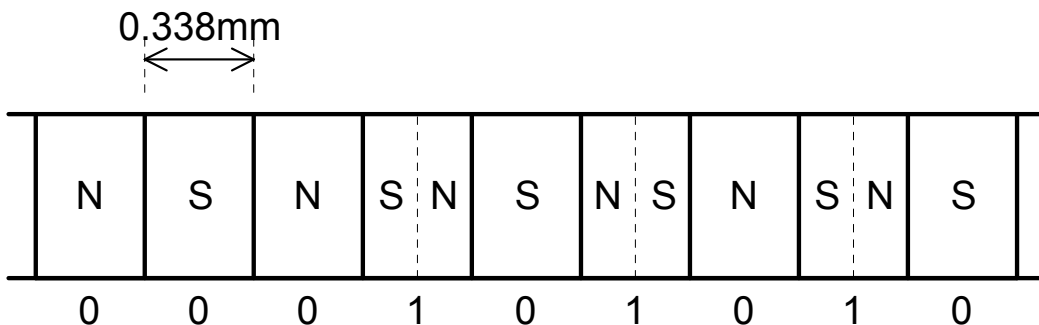
- Данные выводятся младшим битом вперед.
- Первое значение «1»-это первый бит данных **стартового символа** 1011=B hex.
- Каждый символ состоит из 4 битов данных и одного бита контроля четности символа (сумма по модулю 2 всех битов строки).
- Информационных символов может быть до 37.
- После информационных символов идет **Стоп символ** -1111=F hex
- Далее следуют 4 бита контроля нечетности разрядов D0...D3 всех символов -PC0...PC3 (колонок) и бит PP-паритета строки PC0...PC3

- Поскольку код с карты должен выводиться в ASCII коде, то стартовый символ 0x0B преобразуется в 0x3B –т.е. В код «;» а 0x0F преобразуется в 0x3F –т.е. В код «?» .

- Далее до конца магнитной полосы записываются нули.

ЗАПИСЬ НА КАРТУ осуществляется магнитной головкой с зазором 0,1-10 мкм. При плотности записи 75 bpi (бит на дюйм) ширина битового интервала=0,338мм, длительность битового интервала при скорости проводки 10 см/сек-3,38mS.

Направление тока через головку ,(а следовательно и полярность магнитного поля на ленте) в течении битового интервала меняется на противоположное 1 раз при записи «0» и 2 раза- при записи «1» .



Используются магнитные покрытия с малым и большим значением напряженности поля перемагничивания.

Последние трудно размагничиваются и используются для хранения информации, не требующей обновления

Основные достоинства устройств идентификации на магнитных картах следующие:

- Низкая стоимость карт
- Возможность перезаписи

Недостатки

- Невысокая защищенность от подделки
- Контактный способ считывания неудобный, например для контроля транспорта
- Невысокая пропускная способность
- Износ магнитных головок и ленты

**Сетевой интерфейс
1-Wire
MicroLAN**

Общие сведения

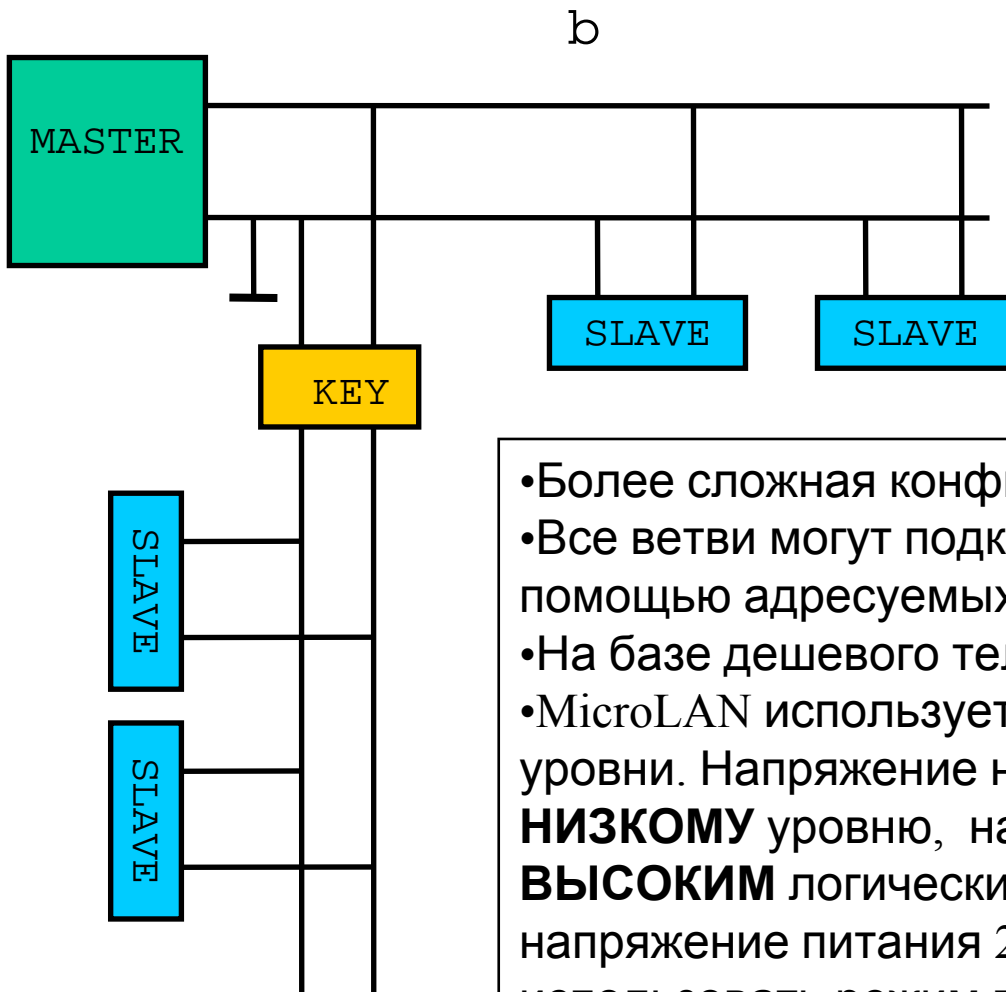
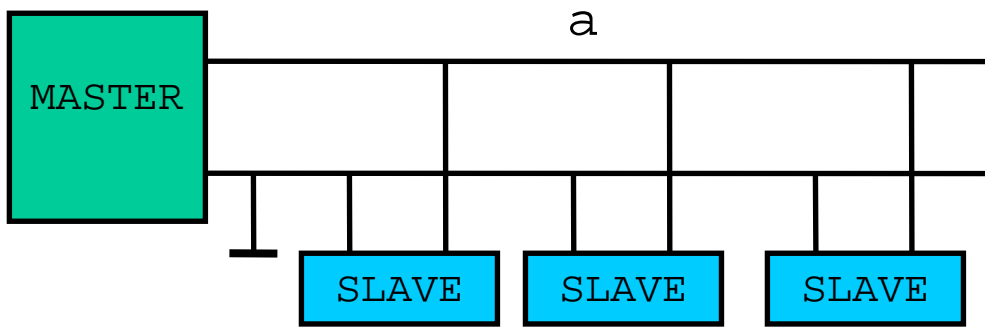
Однопроводной интерфейс 1-Wire , разработан в конце 90-х годов фирмой Dallas Semiconductor , регламентирован разработчиками для применения в трех основных областях:

- приборы в специальных корпусах MicroCAN для решения проблем идентификации, переноса информации (технология iButton),
- системы автоматизации (технология 1-Wire сетей).

Основная идея сетей MicroLAN – использование функционально законченных компонентов со встроенными аппаратными и программными компонентами , необходимыми для работы в сети.

Для построения сети систем автоматизации на базе MicroLAN требуется только разработать программу для мастера сети .

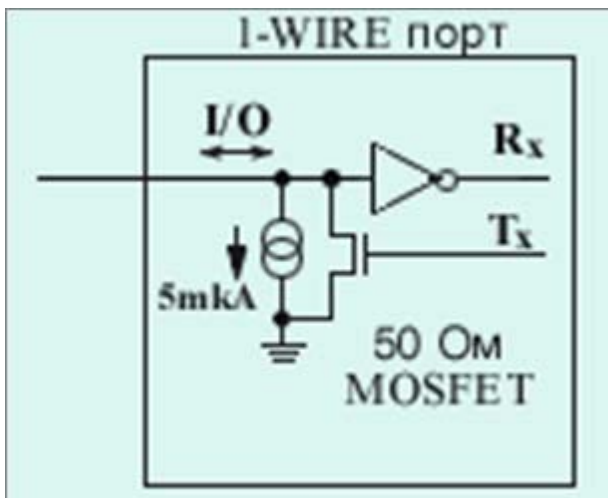
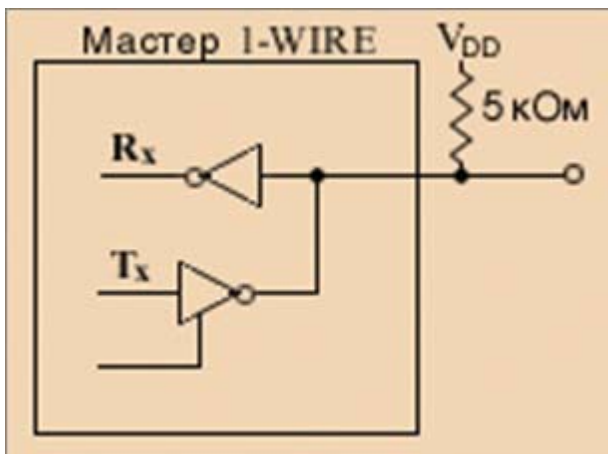
Системы автоматизации на базе 1-Wire-шины еще не получили должного признания из за ограниченного набора компонентов для организации применений в области автоматизации. Однако, номенклатура их постоянно расширяется.



- Сеть MicroLAN использует архитектуру с одним ведущим шины и многочисленными ведомыми. Расстояние –до 300 м.
- В небольших конфигурациях она представляет из себя шинную структуру, с подключением всех приборов на одну общую магистраль (a)

- Более сложная конфигурация- древовидная структура.
- Все ветви могут подключаться к сети или отключаться от нее с помощью адресуемых ключей. (b)
- На базе дешевого телефонного кабеля с витыми парами
- MicroLAN использует стандартные КМОП/ТТЛ логические уровни. Напряжение ниже 0.8 В соответствует логическому **НИЗКОМУ** уровню, напряжение выше 2.2 В является **ВЫСОКИМ** логическим уровнем. Сеть использует рабочее напряжение питания 2.8 ... 6.0 В. Приборы MicroLAN могут использовать режим питания от линии связи.

Подключение компонентов к MicroLAN



Структура формирователя шины MicroLAN- это двунаправленная шина, обеспечивающая ток на землю не менее 1mA.

Для создания сети с мастером на PC, фирма Dallas предлагает недорогой адаптер 1-Wire-линии, а также свободно распространяемую программу iButton Viewer.. Программа позволяет идентифицировать любое из ведомых однопроводных устройств на линии и проверить в полном объеме правильность его функционирования в составе конфигурируемой сети.

Адаптеры :

для параллельного порта -DS1410E,

для COM-порта -DS9097E , DS9097U,

для USB-порта DS9490R.

При создании мастера на базе МП сигналы 1-Wire протокола могут формировааться как самим процессором, так и конверторами:

- конвертор сигналов параллельного порта МП в 1-Wire- DS1481
- конвертор сигналов последовательного интерфейса - DS2480В

Кроме того Dallas создала процессор семейства MCS51- DS80C400, который содержит встроенный в кристалл автомат поддержки 1-Wire-протокола [1]

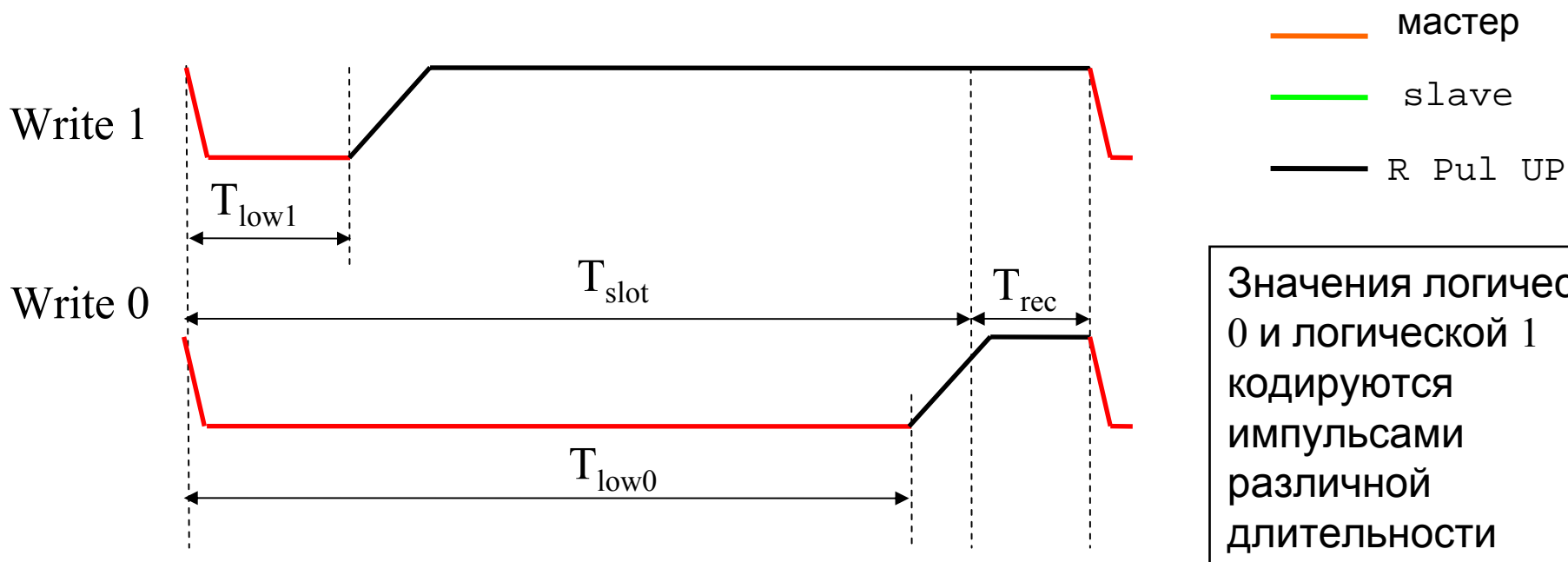


В сети MicroLAN не существует опасности конфликта сетевых адресов и недостаточности адресного пространства. При производстве **гарантируется уникальность сетевого адреса для каждого выпускаемого прибора.**

Последовательность выполнения команд.

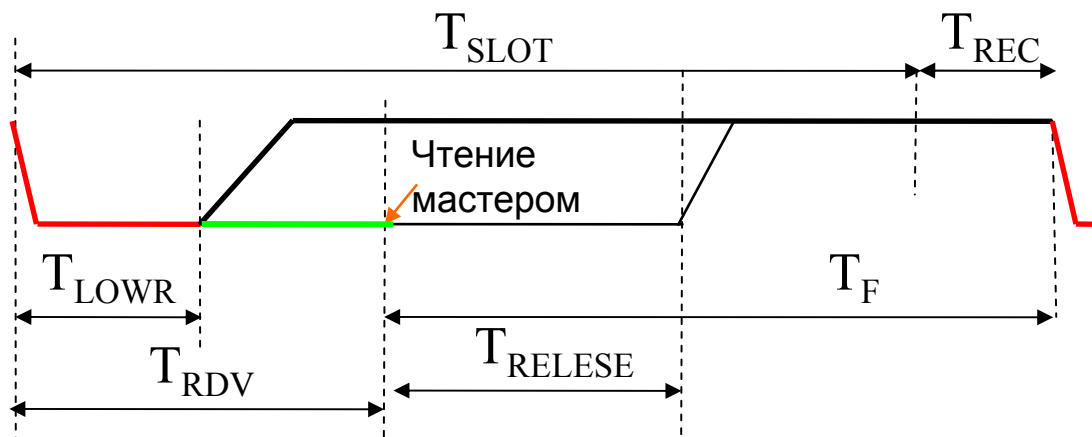
- Любой цикл обмена данными на шине microLAN начинается с передачи мастером шины *импульса сброса*. По этому импульсу прекращается всякий обмен на шине и происходит вывод всех приборов в сети из состояния ожидания.
- Подключение прибора к шине аналогично подаче *импульса сброса*
- Последовательность работы прибора microLAN после подключения к шине полностью идентична последовательности работы после подачи *импульса сброса*.
- После получения *импульса сброса* от мастера все приборы в сети формируют *импульс присутствия*.
- *Импульс присутствия* сообщает мастеру шины, что к шине подключен хотя бы один прибор.
- Далее необходимо провести идентификацию всех приборов в сети.
- После передачи *импульса присутствия* все приборы на шине переходят на сетевой уровень и ожидают команд, связанных с идентификацией.
- В качестве идентификатора каждого прибора используется его уникальный серийный номер, записанный в ПЗУ.

Временные интервалы записи мастером



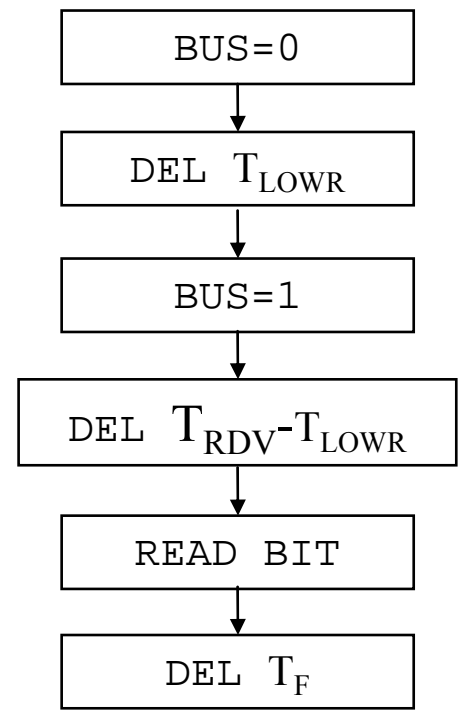
- Синхронизации работы всех приборов MicroLAN производится переключением 1 / 0 сигнала мастера.
- При передаче бита «0» или «1» линия удерживается в нулевом состоянии на разное время – T_{low0} , T_{low1}
- активная часть временного интервала 1-проводной шины (T_{SLOT}) составляет 60 мкс.
- После окончания активной части временного интервала требуется освобождение линии на время T_{rec} , чтобы напряжение на ней по крайней мере на 1 мкс превысило пороговый уровень 2.8 В для зарядки внутренних конденсаторов питания приборов на шине.
- Через интервал времени $T_{SLOT} + T_{rec}$ можно передавать следующий бит [2]

Временные интервалы чтения мастером



Чтение данных от Slave синхронизируется фронтом 1->0 мастера

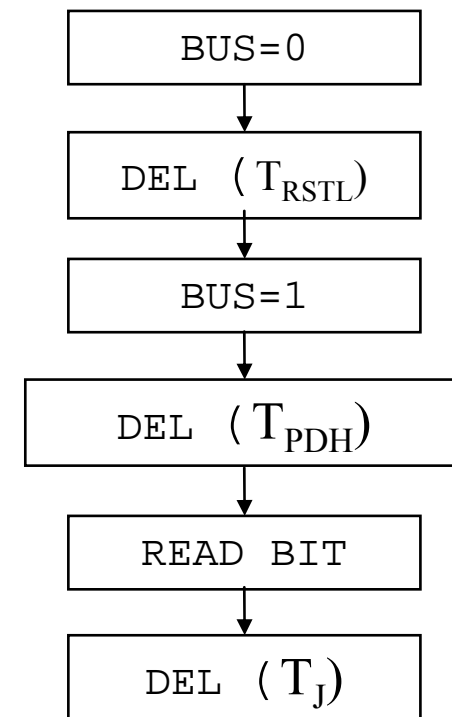
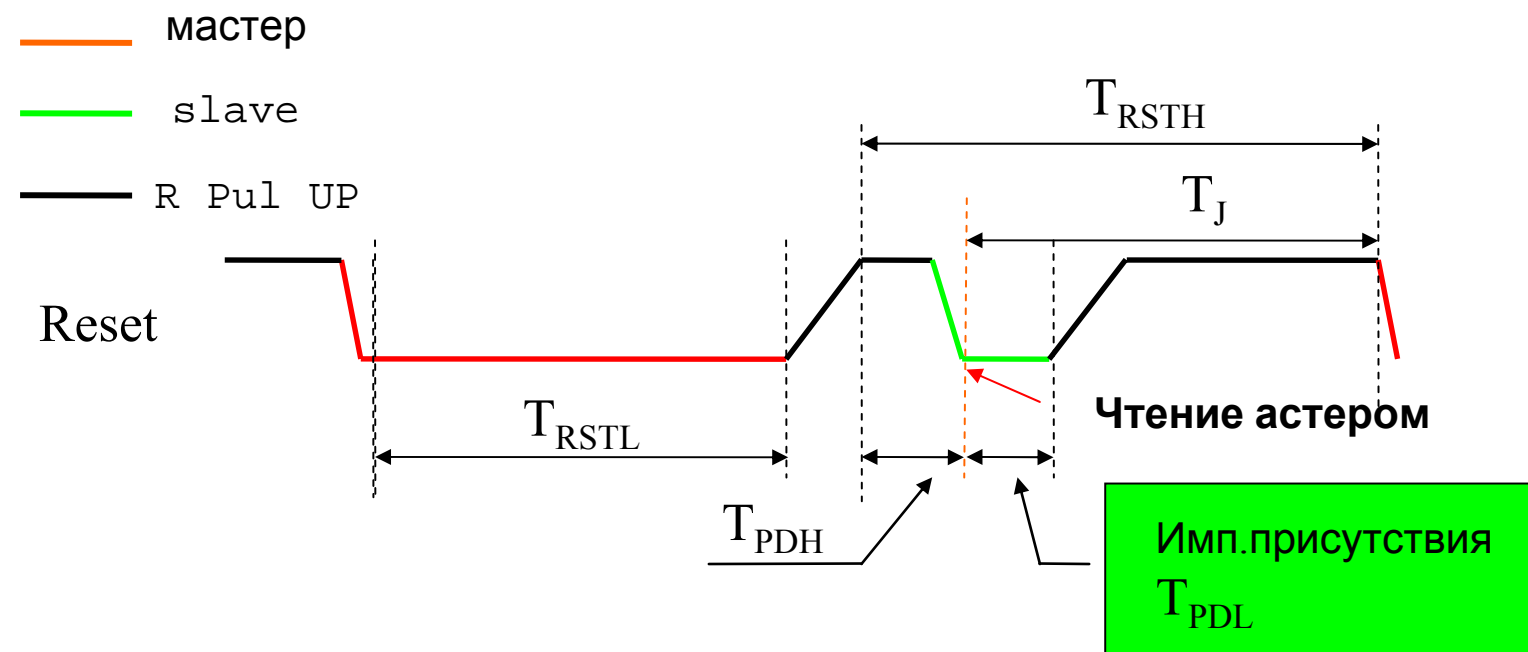
- мастер
- slave
- R Pul UP



По переходу 1 --> 0 от мастера, ведомый прибор посылает один бит адресуемой информации на линию.

- Если этот бит данных равен 1, прибор не предпринимает никаких действий на шине, оставляя шину в единице
- Если бит данных равен 0, прибор MicroLAN сохраняет НИЗКИЙ уровень на шине данных в течение интервала времени T_{RDV} , составляющего 15 мкс.
- Через время T_{RDV} мастер читает состояние шины
- В течении времени $T_{RELEASE}$ ведомый освобождает шину
- Через время T_F от момента чтения бита можно начинать чтение следующего бита

Сброс и обнаружение присутствия на линии



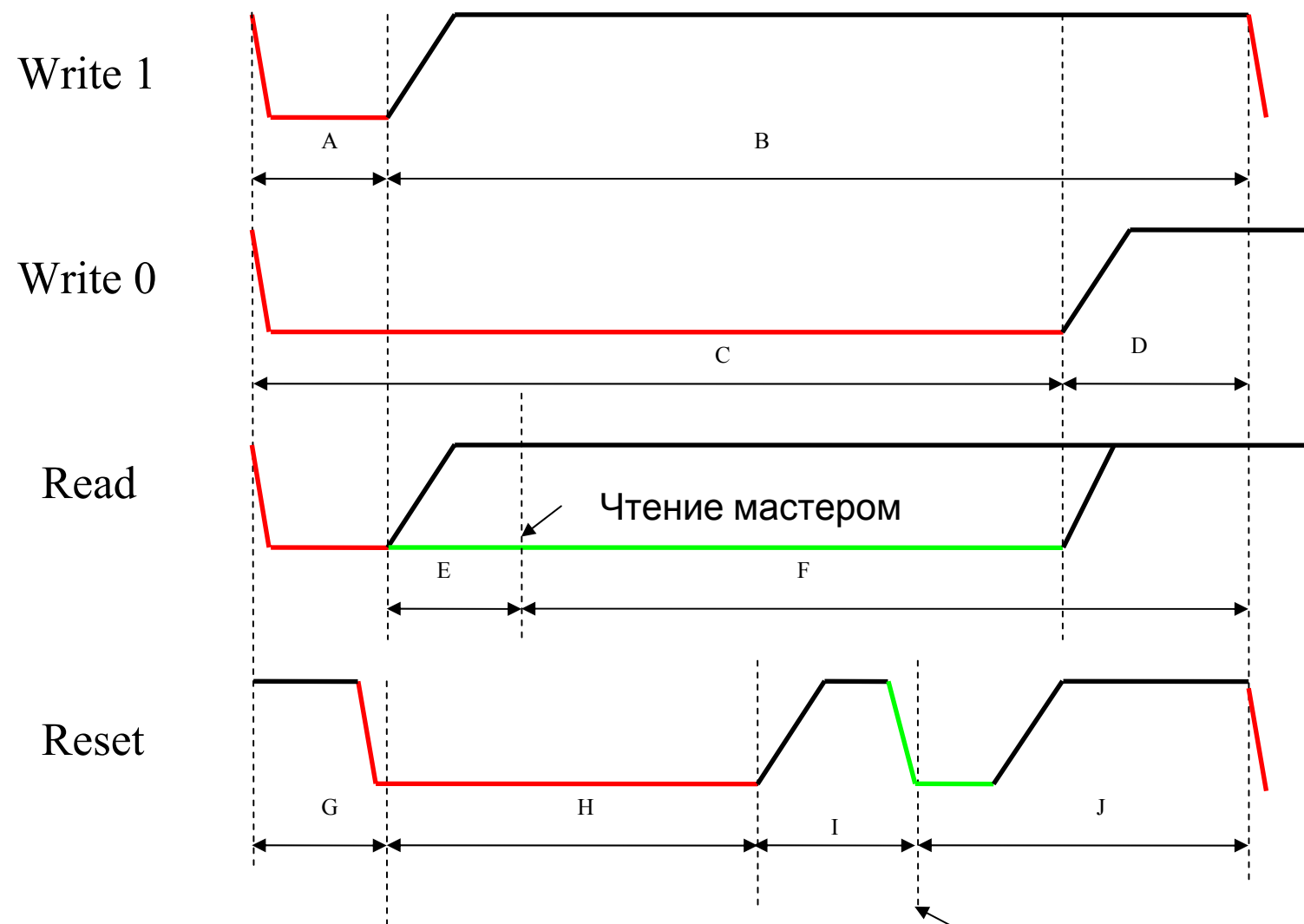
• **Импульс сброса** T_{RSTL} определен как одиночный импульс НИЗКОГО уровня минимальной продолжительностью в восемь временных интервалов (480 мкс) после которого следует ВЫСОКИЙ уровень импульса сброса T_{RSTH} длительностью также 480 мкс. Это состояние ВЫСОКОГО уровня необходимо для того, чтобы приборы на шине MicroLAN могли генерировать **Импульс присутствия**.

• Если ведущий шины посылает Импульс сброса, то прибор MicroLAN ожидает в течение времени T_{PDTH} , и затем генерирует Импульс присутствия «0» продолжительностью T_{PDL} .

Это позволяет ведущему легко определить, находится ли на шине хоть один прибор.

• Далее, через интервал T_J можно подавать следующую команду

Значения временных интервалов



Пар	T_{US}	
	STD	HI
A	6	1.5
B	64	7.5
C	60	7.5
D	10	2.5
E	9	0.75
F	55	7
G	0	2.5
H	480	70
I	70	8.5
J	410	40

Чтение мастером имп присутствия

Примеры программ основных команд на СИ [5]

- Отключение прибора от шины равносильно сбросу.
- Как только прибор снова подключается к шине и обнаруживает высокий уровень на шине данных, он генерирует **Импульс присутствия**. Эта особенность может использоваться для генерации прерывания при подключении на шину каждого нового устройства.

Обмен на шине начинается либо с генерации ведущим шины импульса сброса, либо установкой шины в единицу и циклическим чтением состояния линии мастером до тех пор пока не подключится прибора на шину и на шину будет подан нулевой импульс сброса

- **Импульс присутствия** указывает ведущему, что на шине MicroLAN присутствует прибор, идентификационный номер которого может быть прочитан ведущим. После этого ведущий шины передает ведомому команду.
- Далее, в зависимости от команды, ведущий либо читает данные, либо записывает.

Команды сетевого уровня

Этот уровень обеспечивает идентификацию приборов MicroLAN. Каждый прибор, предназначенный для работы в составе сети MicroLAN содержит страницу данных, необходимую для идентификации прибора, и называемую областью ПЗУ. В эту область при производстве микросхемы записывается с помощью лазерного луча уникальный для каждой микросхемы серийный номер. Занесение этого кода очень строго контролируется в процессе производства, и фирма Dallas гарантирует уникальность серийного номера для каждой микросхемы. Поэтому именно значение серийного номера используется для идентификации прибора в составе сети и для управления доступом к отдельным приборам. Кроме серийного номера в область ПЗУ заносится групповой код, отражающий функциональное назначение микросхемы и контрольная сумма всех данных в области ПЗУ.

Из-за использования области ПЗУ для идентификации и адресации отдельных приборов в сети, все команды, которые относятся к сетевому уровню также названы командами ПЗУ. За исключением DS1990A, все приборы MicroLAN поддерживают все команды Сетевого уровня. DS1990A поддерживает только команды Чтение ПЗУ и Поиск ПЗУ. Команды Пропуск ПЗУ и Совпадение ПЗУ в этом приборе не поддерживаются, так как в нем не имеется больше никакой памяти, требующей доступа. При подаче этих команд на DS1990A, не будет предпринято никаких дальнейших действий на 1-проводной шине.

Краткий обзор команд сетевого уровня приведен таблице

Команда	Код	Краткое назначение команды
Чтение ПЗУ	33h	Идентификация прибора-чтение номера Для прибора DS1990A -0Fh
Совпадение ПЗУ	55h	Адресация только одного прибора, среди нескольких подключенных на шину
Поиск ПЗУ	F0h	Получение серийных номеров всех приборов на шине MicroLAN
		Получение серийного номера одного прибора на шине и одновременная его адресация
Пропуск ПЗУ	CCh	Пропуск адресации, если на шине подключен только один прибор
		Одновременная связь со всеми приборами на шине, например старт преобразования температуры в код в термодатчике DS1820

Команда поиск ПЗУ OWSearch

- Далее посылается однокбайтная команда **поиска ПЗУ. [4]**
- После передачи этой команды мастер 2 раза производится чтение бита с линии, при этом все участвующие приборы одновременно посылают на шину прямое, затем инверсное значение первого бита своих адресов.
- Если содержимое этого бита у всех устройств равно 0, то будет принята последовательность **01**
- Алгоритм поиска начинается со сброса всех приборов, и обнаружения импульса присутствия
- Если содержимое этого бита у всех устройств равно 1, то будет принята последовательность **10**
- Если же содержимое бита будет разное (1,0 или 0,1), то будет принята последовательность **00**.
- Далее мастер передает на шину 1 (или 0)-чтобы оставить в работе только те приборы у которых значение этого бита равно 1 (или 0)..Все остальные приборы выключатся до следующего импульса **Сброса**.
- Оставшиеся активные приборы во время двух последующих циклов чтения передадут на линию прямое и инверсное значение следующего бита своих адресов.
- Таким образом за 64 цикла мастер адресует 1 прибор,отключив все остальные
- Для определения адресов остальных устройств в линии все повторяется, но при конфликте на линии мы посылаем в шину 0 ,отключаем устройства со значением совпадающего бита=1.
- Определив один раз адреса всех устройств-далее можно к ним обращаться по считанному адресу команд **Совпадение ПЗУ**.

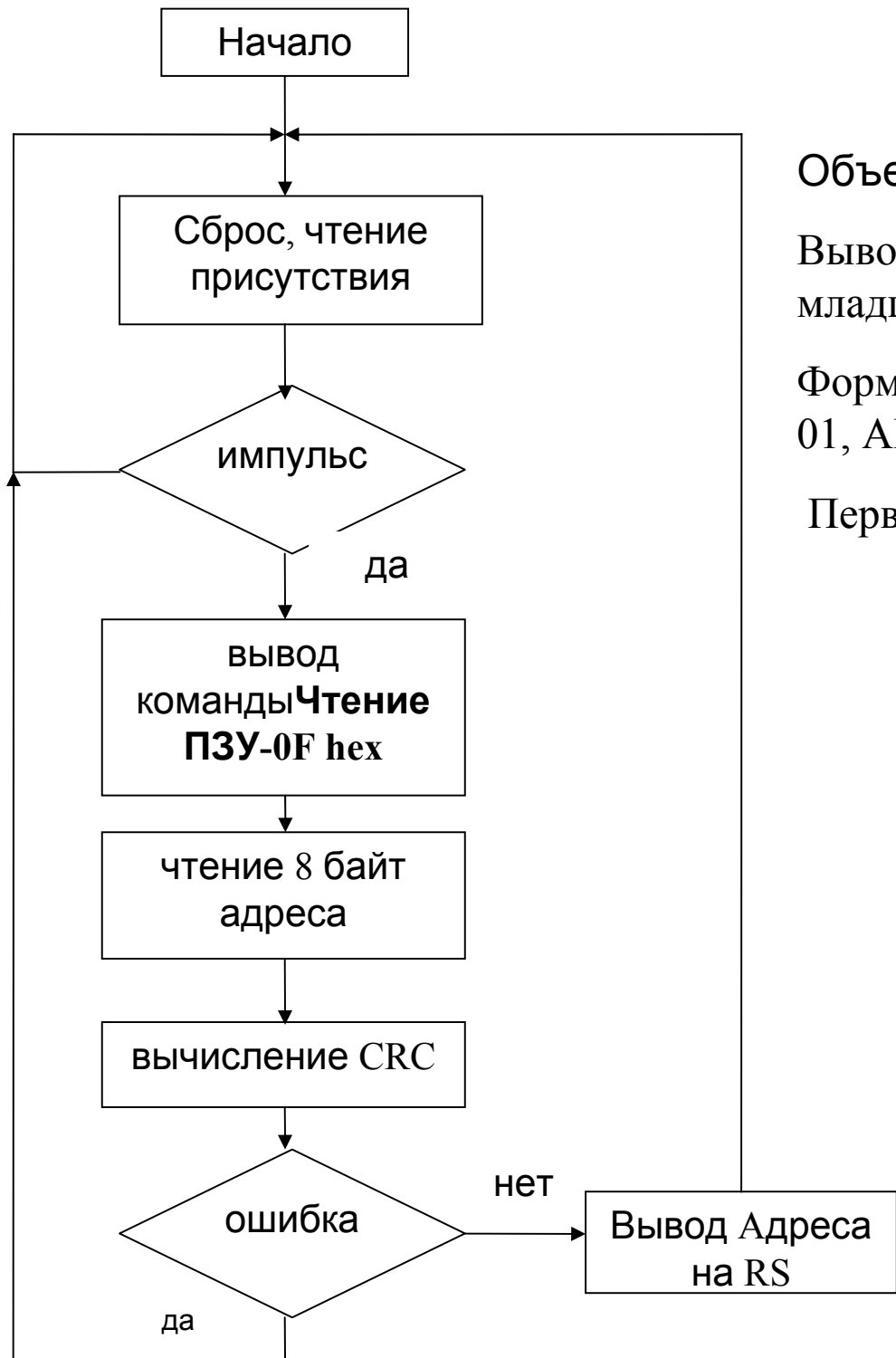
Команда поиск ПЗУ OWSearch

	0	1	1	0	A _i - биты адреса 1 устройства
	0	0	1	0	B _i - биты адреса 2 устройства
	0	0	1	0	A & B – состояние шины при прямых значениях битов адр.
	0	0	0	1	!A & !B – состояние шины при инверсных значениях битов адр.

Транспортный уровень

Этот уровень ответственен за передачу данных между всеми областями памяти устройств MicroLAN (кроме области ПЗУ) и ведущим шиной,

Алгоритм чтения DS1990



Объем памяти ПЗУ-64 бита- 8 байт

Вывод данных –8 байт начиная с младшего бита младшего байта.

Формат данных при чтении с таблетки- 01, AD0, AD1...AD5, CRC

Первый байт 01-номер серии прибора .

Алгоритм вычисления CRC

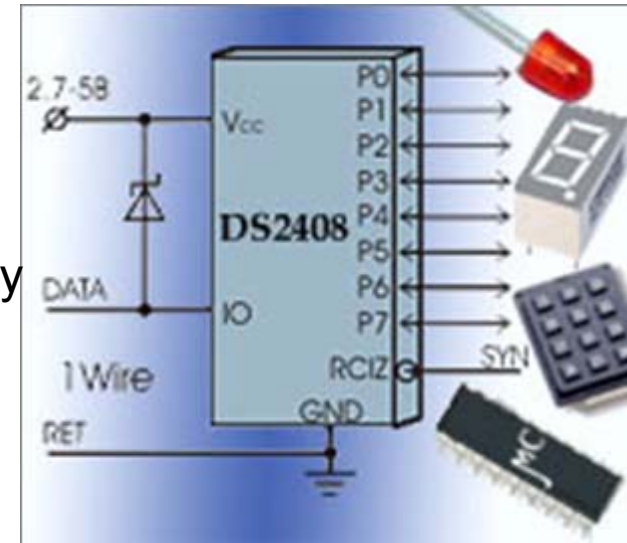
```
crc=0;
```

```
for(i=0; i< 7;i++)
```

```
    crc=TAB_CRC[crc^dat[i]];
```

Ведомые однопроводные компоненты

При всем многообразии однопроводных компонентов, наиболее универсальным из них является индивидуально двунаправленный восьмиразрядный свободно поразрядно программируемый по 1-Wire-шине порт ввода/вывода DS2408. Он позволяет реализовать любой интерфейс между внешним устройством произвольной модификации и однопроводной линией. Этот прибор имеет двунаправленный вывод внешней синхронизации, обеспечивающий аппаратное тактирование передаваемых или принимаемых данных.



Использование микросхемы DS2408 позволяет обеспечить управление посредством 1-Wire-шины: сосредоточенным двунаправленным вводом/выводом по 8 независимым каналам, светодиодными и жидкокристаллическими индикаторами, сканированием матричных клавиатур и дискретных датчиков самых различных типов, а так же позволяет реализовать действительно полномасштабный интерфейс с различными типами микроконтроллеров, как в последовательном, так и в параллельном коде.

Литература

1. Интерфейс 1-wire.htm www.RS232.ru
2. MicroLAN.htm Сетевой интерфейс MicroLAN. Особенности построения.pdf
3. Схема адаптера DS9097 (пассивная)
4. Новая концепция построения однпроводной сети dodeka_ml.pdf
5. Примеры программ на СИ основных команд Микролан.
AN126 _Software.htm

**Устройства
бесконтактной
радиочастотной
идентификации (RFID)**

Устройства бесконтактной радиочастотной идентификации (Radio Frequency Identification -RFID)

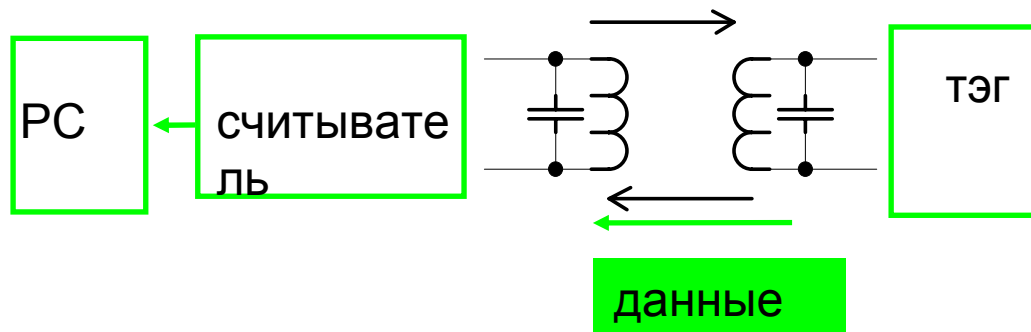
Основные области применения:

- Идентификация животных.
- Контроль доступа, электронные ключи.
- Фишки для казино.
- Спортивные состязания.

СОСТАВ СИСТЕМЫ RFID

Типичная система RFID состоит из:

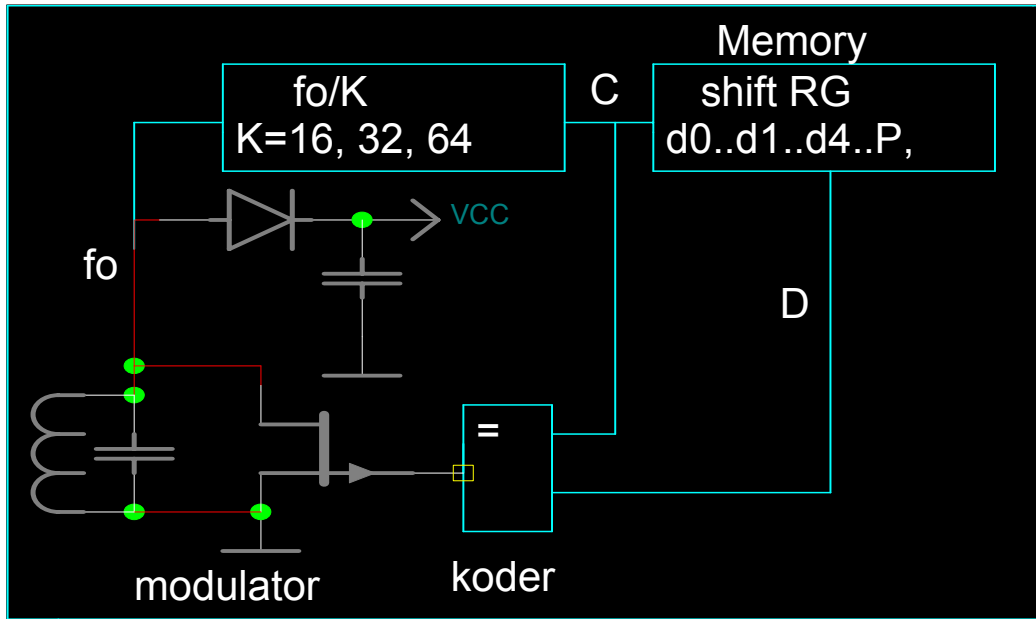
- радиочастотной метки или транспондера - Tag, Transponder в виде карты, брелка и т.д
- считывателя информации Reader
- компьютера для обработки информации .



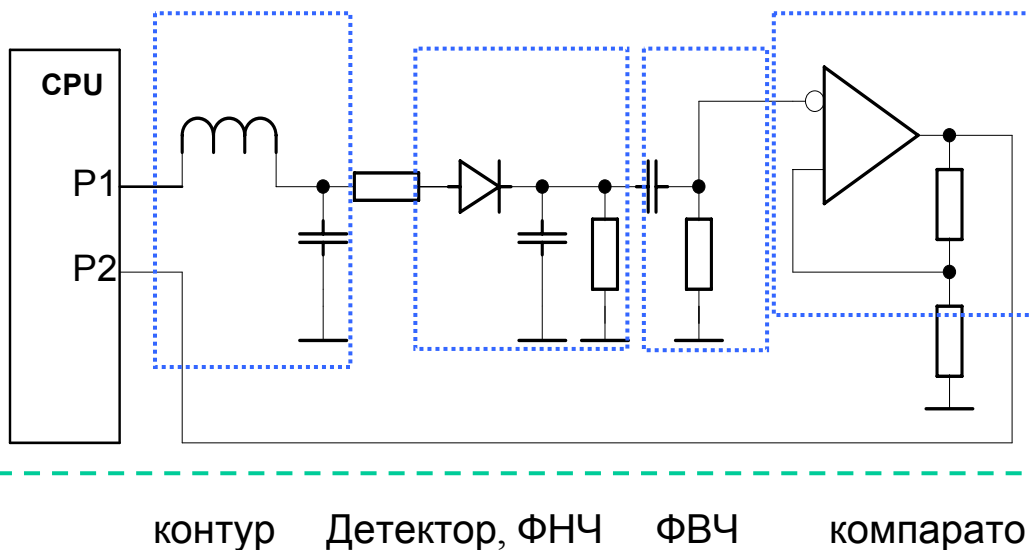
Транспондер пассивное устройство, состоит из микросхемы с фиксированным или перепрограммируемым кодом, и колебательного контура.

- Считыватель содержит генератор с колебательным контуром на ту же частоту
- При попадании транспондера в поле считывателя в его контуре индуцируется эдс, которая питает микросхему транспондера.
- Закодированная информация переключает полевой транзистор, подключенный параллельно контуру карточки, амплитудно модулируя ток в его контуре. Это изменение тока передается по индуктивной связи на контур считывателя, детектируется и декодируется.

Структура транспондера



Структура считывателя



• Далее компаратором формируются фронты сигнала и он подается на CPU для декодирования

Транспондер (Тэг) включает в себя:

- Выпрямитель с накопительной емкостью для питания схемы.
- Частота амплитудной модуляции-такт сдвига импульсов из регистра, задается делителем fo/K на 16, 32, 64 от несущей частоты fo .
- Для передачи данных и сигнала по одному каналу-в данном случае через индуктивную связь трансформатора, производится Манчестерское кодирование последовательных данных с выхода регистра сдвига и синхросигнала. Это сумма по модулю 2 сигналов C и D.
- Модулирующий элемент-это полевой транзистор, шунтирующий контур Тэга

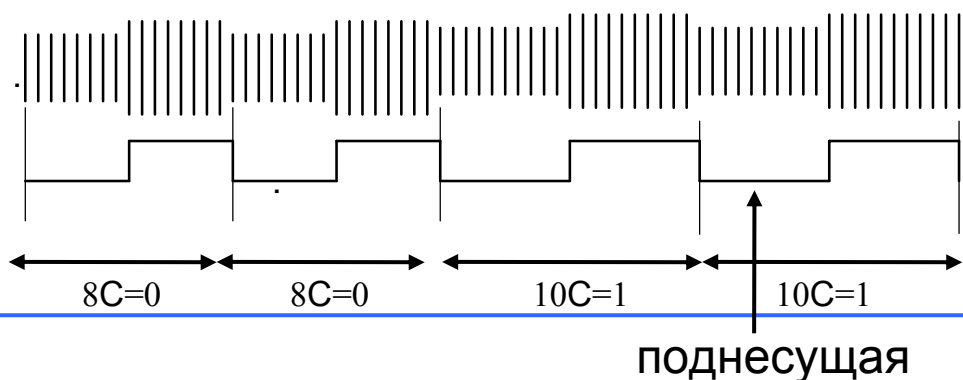
Считыватель содержит генератор с колебательным контуром на ту же частоту

• При шунтировании контура Тэгом, из за индуктивной связи, в контуре считывателя происходит амплитудная модуляция несущей частоты формируемой на выходе P1 CPU.

• Этот амплитудно модулированный сигнал детектируется, фильтром нижних частот удаляется несущая высокая частота, а фильтром верхних удаляется постоянная составляющая.

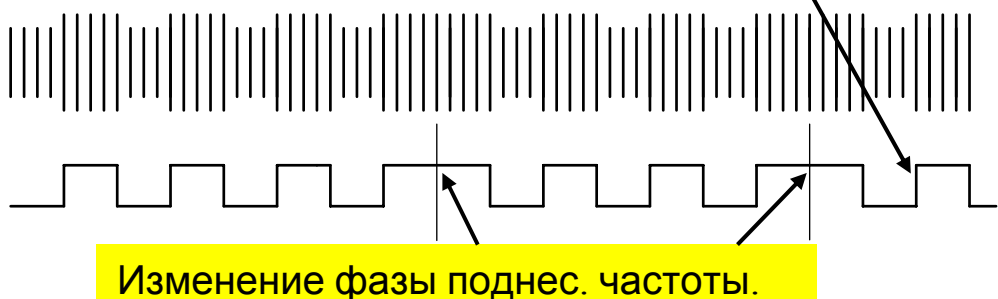
Модуляция сигнала Выбор способа модуляции сигнала влияет на помехоустойчивость системы. $U=A\sin(\Omega t)*\sin(\omega t)$ - амплитудная модуляция аналоговым сигналом частотой Ω . $U=B*\sin(\omega t)$ - амплитудная манипуляция дискретным сигналом $B=0,1$. Самый простой способ модуляции-амплитудная-наиболее подвержена влиянию помех. Поэтому применяется частотная или фазовая манипуляция поднесущей частоты. При этом поднесущая частота амплитудно манипулирует несущую, а поднесущий сигнал манипулируется передаваемым кодом по частоте или по фазе.

Частотная манипуляция



При частотной манипуляции поднесущей, каждому биту данных соответствует период колебания поднесущей. 8 периодов CLK соответствует нулю данных, 10 периодов-единице.

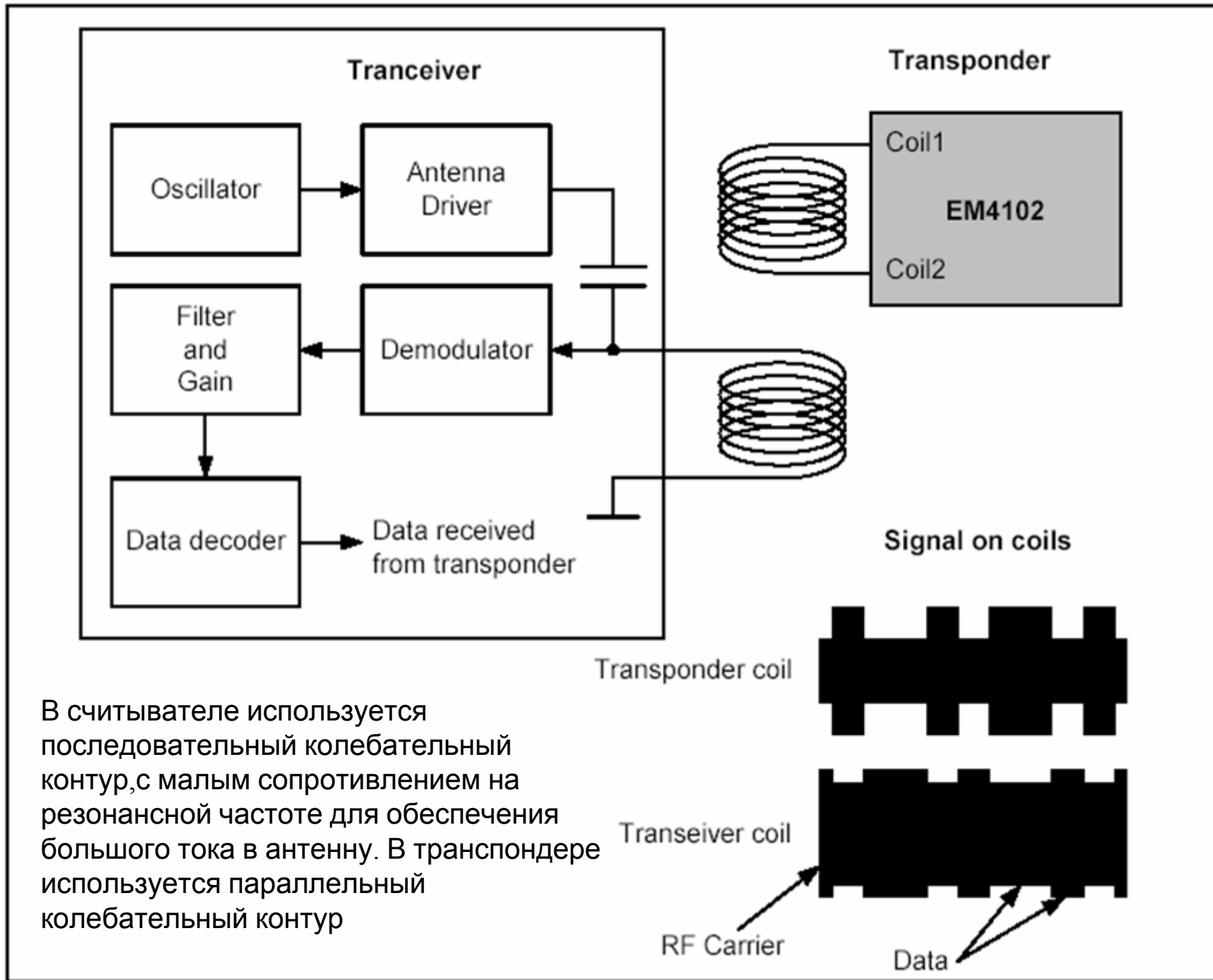
Фазовая манипуляция



При фазовой манипуляции меняется фаза поднесущей частоты на 180 градусов. Используется 2 основных типа фазовой манипуляции:
 1. Изменение фазы на 180° при передаче каждого значения «0»- (двухфазный код).
 2. Изменение фазы на 180° при переходе «0-1» или «1-0» (манчестерский код)

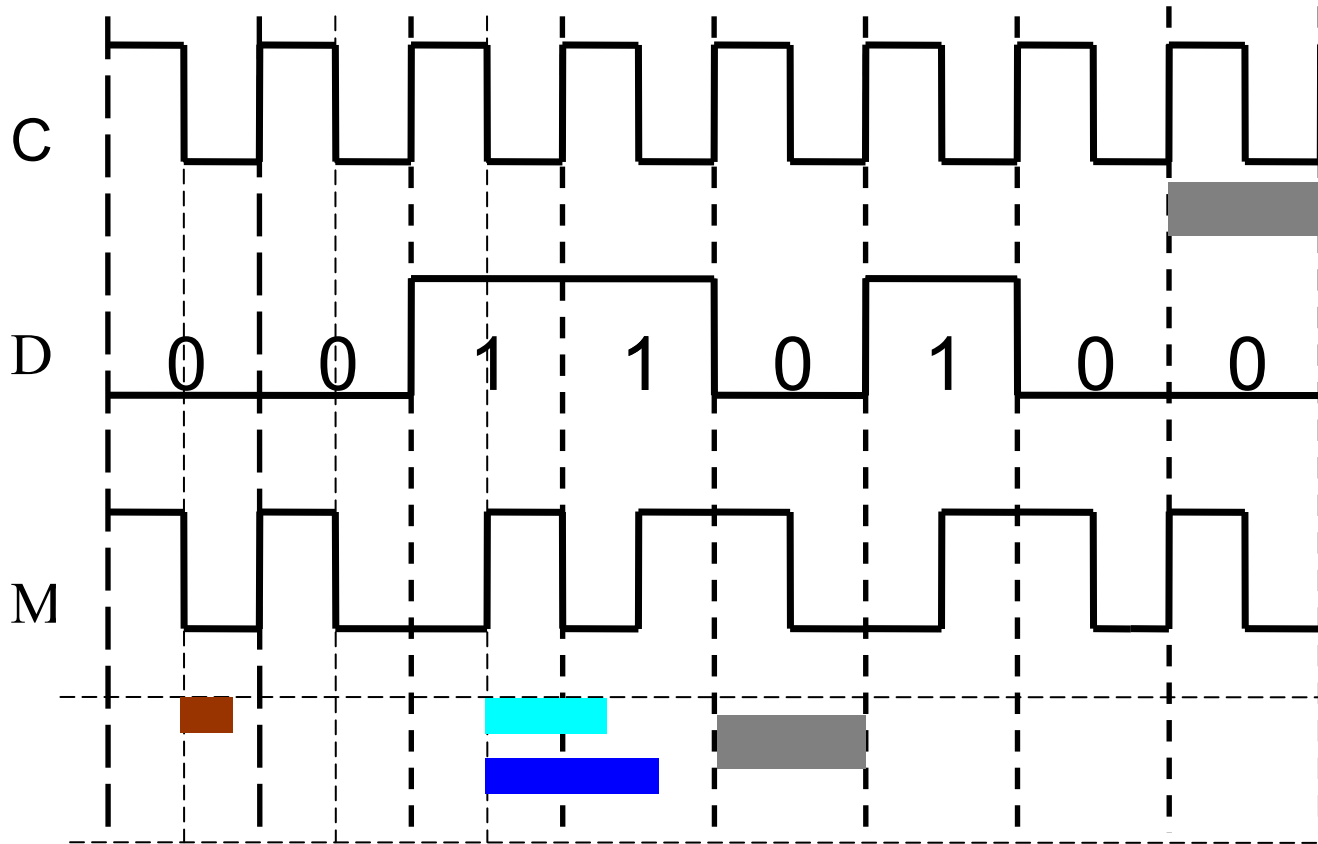
Оба кода- двухфазный , и манчестерский - самосинхронизированные коды-т.е. содержат в себе данные и синхросигнал

System Principle



В считывателе используется последовательный колебательный контур, с малым сопротивлением на резонансной частоте для обеспечения большого тока в антенну. В транспондере используется параллельный колебательный контур

Временные диаграммы. Манчестерский код



Для чтения манчестерского кода привязка производится к середине битового интервала

TS_{min} 150 μs

TL_{max} 600 μs

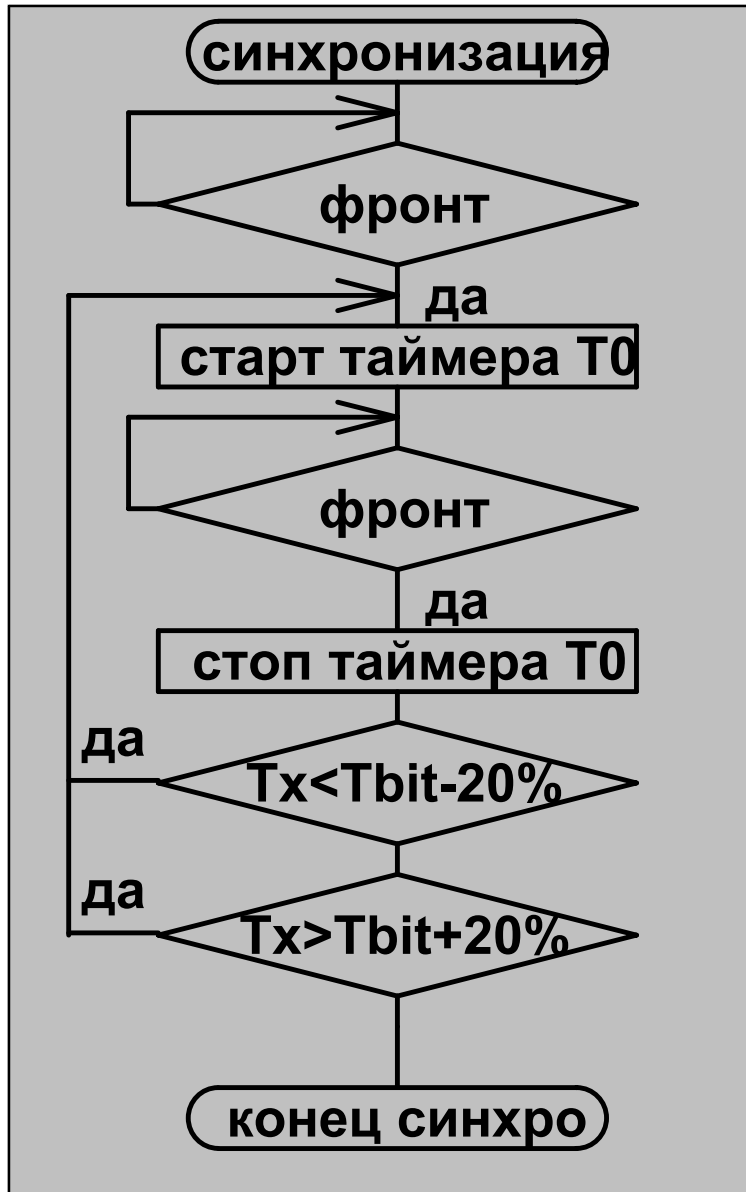
T_{next} 275 μs

Битовый интервал

$T_{бит} = 512 \mu s$ при $V=64F_{osc}$

- Манчестерский код (M) - это сумма по модулю 2 синхро сигнала (C) и данных (D). $M=C \oplus D$, т.е. Сигнал $M=1$ когда D и C разные и сигнал $M=0$ когда D и C одинаковы.
- Единица данных в Манчестерском коде представляется положительным перепадом в середине битового интервала, Ноль данных - отрицательным перепадом
- Для RFID с $F_{clk}=125$ кГц длительность битового интервала может быть 16,32,64 периодов синхро т.е $T_c=8 \mu s$; $T_{бит}=128, 256, 512 \mu s$. Скорость передачи - 7,8 3,9 1,95 кбод.
- Алгоритм декодирования манчестерского кода требует нахождения середины битового интервала для вхождения в синхронизацию, и далее определения перепада 0->1 или 1->0 через интервал времени, равный битовому интервалу

Чтение Манчестерского кода



Синхронизация

-середина битового интервала –это момент окончания длинного импульса.
-Для его нахождения при обнаружении перепада данных включается таймер, на следующем перепаде он останавливается. -
Если измеренная длительность лежит в пределах $T_{бит} \pm 20\%$ - то середина битового интервала найдена. Если нет-то возврат к началу.

Чтение последующих битов. Чтение последующих битов производится в серединах следующих битовых интервалов. Для этого запускаются два таймера, T_x -длительности следующего импульса, и таймер задержки на время $T_{next} \approx T_{бит} - 20\%$. После завершения времени T_{next} отлавливается фронт сигнала. По его обнаружению проверяется длительность текущего импульса. Если она не укладывается в пределы $T_{бит} \pm 20\%$. то это ошибка чтения и возвращаемся в начло. Иначе в буфер данных записывается значение бита в зависимости от направления перепада и переходим к чтению следующего бита.

После чтения очередного бита таймер сбрасывается. Таким образом синхронизация не нарушается

Формат данных на карточке

- Число бит на карточке – 64.
- Начало сообщения задается последовательностью девять единиц.- 111111111
- После этого следуют 2 HEX символа идентификатора изготовителя с битами паритета (сумма по модулю 2 информационных битов.)
- Далее-8 символов данных D7...D0 с паритетами P7...P0.
- И наконец 4 бита паритета колонок PC4..PC0

1	1	1	1	1	1	1	1	1
8 бит идентификатора				D93	D92	D91	D90	P9
32 бит ДАННЫХ				D83	D82	D81	D80	P8
P0...P9-бит паритета строк				D73	D72	D71	D70	P7
				D63	D62	D61	D60	P6
				D53	D52	D51	D50	P5
				D43	D42	D41	D40	P4
				D33	D32	D31	D30	P3
				D23	D22	D21	D20	P2
				D13	D12	D11	D10	P1
				D03	D02	D01	D00	P0
4 бита парит. кол				PC3	PC2	PC1	PC0	0

Преимущества Манчестерского кода

- В коде NRZ данных из памяти могут присутствовать частоты от 0 до F_{clk}
- В Манчестерском коде присутствуют частоты F_{clk} и $F_{clk} : 2$. Постоянная составляющая примерно равна нулю. Поэтому приемник может быть более узкополосным и следовательно-более помехозащищенным. Кроме того легко реализуется трансформаторная и емкостная связь.
- Преимущество Манчестерского кода по сравнению с асинхронным кодом (RS232) – простота синхронизации и допустимый уход часов приемника и передатчика-до $\pm 25\%$. Тогда как для RS232-4%

ЛИТЕРАТУРА.

1. Atmel. U2270B Read/Write Base Station
2. Atmel. U2270B Application Note
3. EM MICROELECTRONIC - MARIN - SA EM4102. Read Only Contactless Identification Device. www.emmicroelectronic.com
4. Семейство меток Hitag для радиочастотной идентификации от Philips Semiconductors
5. Волковицкий В.Д. Системы контроля и управления доступом. СПб 2003г.

Цветной графический дисплей
AM-320240
фирмы AMPIRE

Цветной графический дисплей AM-320240 фирмы AMPIRE

Основные характеристики

- - разрешение 320x240 пикселей
- - количество цветов 65К,
- - размер по диагонали 5,7”.
- - Может работать с шиной данных 8, 9, 16, 18 Бит
- - Есть модификация, адаптированная к шине MCS51 с 8 разрядной шиной данных и сигналами управления:
 - - RS – Выбор регистра команд / данных
 - - /CS– Выбор устройства
 - - /WR -Запись
 - - /RD -Чтение

Программная модель дисплея состоит из 55 8-битных регистров, а также оперативной памяти [1].

На рис. 1 представлена временная диаграмма, определяющая порядок записи данных в один или несколько регистров дисплея.

55-pin connector pinout:

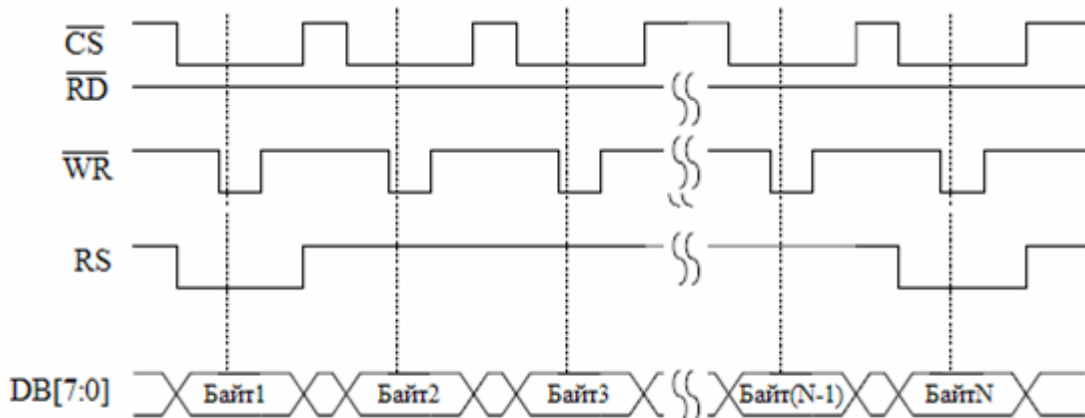
5	RES
6	RS
7	CS
8	WR
9	RD
10	DB0
11	DB1
12	DB2
13	DB3
14	DB4
15	DB5
16	DB6
17	DB7
18	DB8
19	DB9
20	DB10
21	DB11
22	DB12
23	DB13
24	DB14
25	DB15
26	DB16
27	DB17
28	65K/262K
30	SK/X1
31	DO/X2
32	DI/Y1
33	TPCS/Y2
34	INT
35	VDD
36	VDD
37	VDD
38	DGND
39	DGND
40	DGND
41	DGND
42	DGND
43	DGND
44	DGND

DIS_AM320240N6TMQW-TW4H

Цветной графический дисплей AM-320240 фирмы AMPIRE

На рис. 1 представлена временная диаграмма, определяющая порядок записи данных в один или несколько регистров дисплея.

Запись данных в регистры дисплея



При записи каждого байта линии /RS и /WR необходимо установить в «0», а линию /RD в «1».

Байт1 должен содержать адрес регистра, в который будет осуществлена запись данных, а Байт2 - записываемые данные.

При посылке байта, содержащего адрес регистра, на линии /RS необходимо выставить «0», а при посылке байта с данными – «1».

Автоматический инкремент адреса регистра позволяет записывать данные в несколько регистров без указания их адреса. Так, значение Байта3 будет скопировано в регистр, адрес которого на единицу больше адреса, указанного в Байте1 и т.д. Для того, чтобы закончить запись в регистры дисплея, необходимо послать байт со значением 0x80, выставив при линии RS «0».

Процедура записи данных в оперативную память дисплея выполняется аналогичным образом. Разница заключается лишь в том, что Байт1 должен содержать значение 0xC1, разрешающее запись в память.

Вывод данных на дисплей

Перед тем, как выводить данные на дисплей, необходимо задать прямоугольную область вывода. Для этой цели в программной модели дисплея предусмотрено 8 регистров REG[0]... REG[7]

MIN X address --> (REG[0] & REG[1])

MAX X address --> (REG[2] & REG[3])

MIN Y address --> (REG[4] & REG[5])

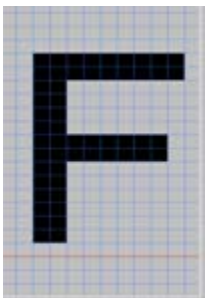
MAX Y address --> (REG[6] & REG[7])

После задания координат области вывода в экранную память выводится 16-и битовый код, задающий цвет точки в следующем формате R5 G6 B5



При последовательной записи в память осуществляется автоматический инкремент адреса пикселя в пределах заданной области вывода.

Одномерный массив знакогенераторов можно получить конвертируя фонты Windows конвертором «Font and Bitmap Converter» фирмы Microchip



Вывод символа на дисплей

Точечная картинка символа оцифровывается побайтно начиная с верхнего правого угла. Формат таблицы символов в Таб.1. Для вывода символа на экран задается размер знакоместа, далее побайтно читается BMP байты из знакогенератора, анализируются их биты

Если бит в равен 1-то в ОЗУ дисплея загружается RGB слово соответствующее цвету цвет символа, если 0-то слово, соответствующее фону.

Таб1

Блок	Назначение	Наименование	Кол. байт	Описание
Заголовок шрифта		Резерв	1	записан 0
		Идентификатор шрифта	1	номер шрифта задается пользователем
			2	ASCCI код первого символа
			2	ASCCI код последнего символа
		Резерв	1	записан 0
		Резерв	1	записан 0
		Высота символа	1	Высота символа в пикселях
Характеристики символов	Первый сим.	Сдвиг Hi byte	1	Сдвиг от начала BMP блока Hi byte
		Ширина симв.	1	Ширина в пикселях
		Сдвиг Low byte	2	Сдвиг от начала BMP блока Low byte
	Последний символ	Сдвиг Hi byte	1	Сдвиг от начала BMP блока Hi byte
		Ширина симв.	1	Ширина в пикселях
		Сдвиг Low byte	2	Сдвиг от начала BMP блока Low byte
BMP символ.	первый сим.		W	BMP массив первого символа
	второй символ		W	BMP массив второго символа
	
	послед. симв.		W	BMP массив последнего символа

Таб2

0x0020 ...										
	!	"	#	\$	%	&	'	()	*
+	,	-	.	/	0	1	2	3	4	5
6	7	8	9	:	;	<	=	>	?	@
A	B	C	D	E	F	G	H	I	J	K
L	M	N	O	P	Q	R	S	T	U	V
W	X	Y	Z	[\]	^	_	'	a
b	c	d	e	f	g	h	i	j	k	l
m	n	o	p	q	r	s	t	u	v	w
x	y	z	{		}	~				
... 0x007E										

Создание знакогенератора определенного набора символов

-Создается .txt файл.фильтра в котором задаются требуемые для вывода на дисплей строковые последовательности

1 3 2 : 1 3 2

6 5 7 : 6 5 7

-Из ASCII таблицы Таб.2 выбираются символы, содержащиеся в файле фильтра, т.е. подлежащие к дальнейшему выводу.

-BMP изображения этих символов, в порядке возрастания их ASCII номеров записываются в массив знакогенератора(Таб3).

Таб3

Симв.	'SP'	'1'	'2'	'3'	'5'	'6'	'7'
ASCII	0x20	0x31	0x32	0x33	0x35	0x36	0x37
New №	0x20	0x21	0x22	0x23	0x24	0x25	0x26

-Записанным символам присваиваются адреса, начиная с 0x21.-Таб.3.-строка 3

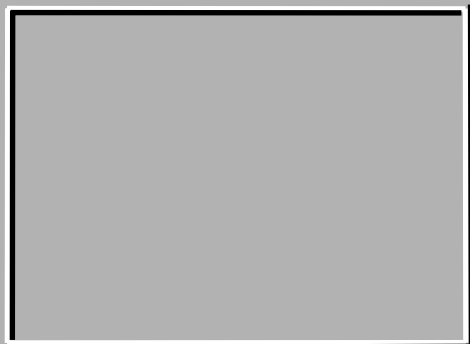
- Создается массивы 132 [] и 657 [], вида

char 132 [] = {0x0021, 0x0023, 0x0022, 0x0000}; (A)

char 657 [] = {0x0025, 0x0024, 0x0026, 0x0000}; (B)

с конвертированными согласно Таб.3.(строка 3) номерами для строки "123" и "657"

-Программа вывода массива на дисплей из сокращенного массива Таб3 поочередно выбирает символы с номерами соответственно (A), (B).



Элементы дизайна экрана приборов
с цветным графическим дисплеем-кнопки, рамки...



ЛАХТАМИЛОН



MULTIPULSE MODE 973 nm ●

POWER 5.0 W

DURATION 1000 ms

INTERVAL 1000 ms

Energy 0 J

Count 00s

visualisation
pulse energy 5,00 J

▶◀ pulse counter 0 sound [||||] ▶

READY BUTTON → **START** MENU



POWER OFF



ЛАХТАМИЛОН



MIX MODE 973 nm

POWER 5.0 W

DURATION 1000 ms

INTERVAL 1000 ms

count 0 J

ENERGY 50,0 J

revers count 50,0 J

visualisation

pulse energy 5,00 J

count 00s

EXPOSURE 20s

revers count 20s

▶◀ pulse counter 0

READY BUTTON ▶ **START** MENU



STOP

START

ON

POWER OFF

