

# Последовательный канал информационного обмена по стандарту ARINC-429

(ГОСТ 18977-79)

## Лабораторная работа №405\_ND

Стандарт ARINC-429, разработанный фирмой ARINC, предназначен для межсистемного обмена информацией в коммерческих и транспортных самолётах (в России это ГОСТ-18977-79). Скорость передачи 12.5, 50 или 100 кбит/сек. Соединительные провода — экранированные витые пары. На одной шине (витой паре) может быть только один передатчик и не более 20 приемников. Передатчик всегда активен, он либо передаёт слова данных или выдаёт "пустой" уровень (0 В). Размер слова составляет 32 бита. Бит 32 — контроль четности (дополнение до нечетного числа единиц). Код — биполярный самосинхронизирующийся, с возвратом к нулю (RZ). Логической единице соответствует положительный импульс, а логическому нулю — отрицательный импульс. Длительность импульса равна половине интервала следования (длительности такта). Импульсы должны иметь пологие фронты примерно 0.1-0.3 от длительности импульса. Пример фрагмента временных диаграмм сигналов шины стандарта ARINC 429 приведен на рис.1.

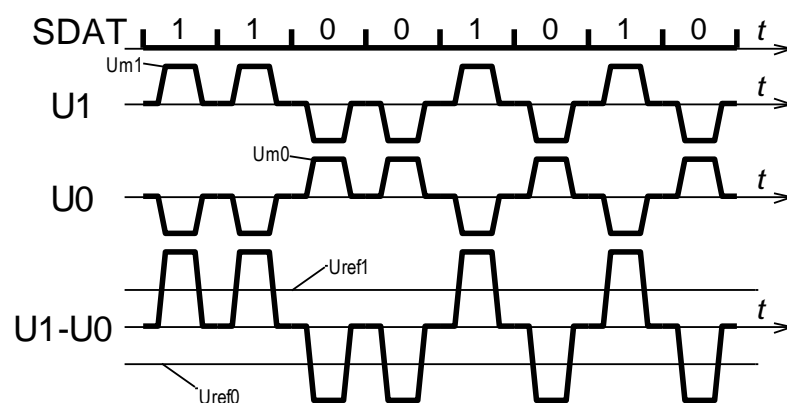


Рис.1. Пример диаграмм сигналов стандарта ARINC 429

Для формирования аналоговых сигналов дифференциальной линии U1, U0 можно использовать, например, микросхему HI-8570 (рис.2) фирмы HOLT, специализирующейся на выпуске микросхем для ARINC-429. Вход SLP предназначен для управления крутизной фронтов сигналов на линии передачи. При SLP=0 длительность фронтов импульсов равна 1.5 мкс, а при SLP=1 длительность фронтов импульсов равна 10 мкс.

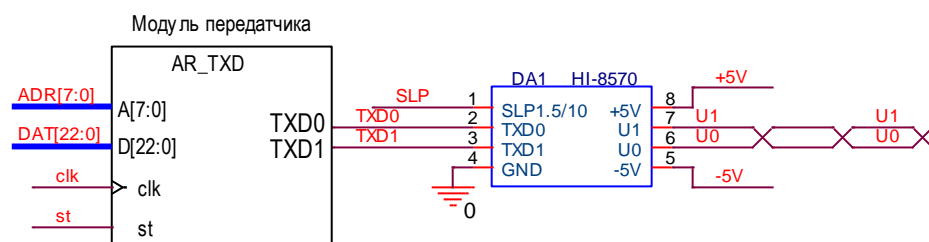


Рис.2 Микросхема выходного драйвера ARINC-429 фирмы HOLT

Для обратного преобразования аналоговых сигналов U0 и U1 в логические сигналы RXD0 и RXD1 можно использовать, например, микросхему HI-8591 той же фирмы HOLT.

Первые 8 бит являются адресом абонента, следующие 23 бита являются данными, а последний контрольный бит дополняет число единиц в слове до нечетного числа. Особенностью формата ARINC слова является то, что первые 8 бит адреса передаются старшими битами вперед, а 23 бита данных передаются младшими битами вперед. Инверсия нумерации бит считается недопустимой. Минимальная пауза между словами 4 бита.

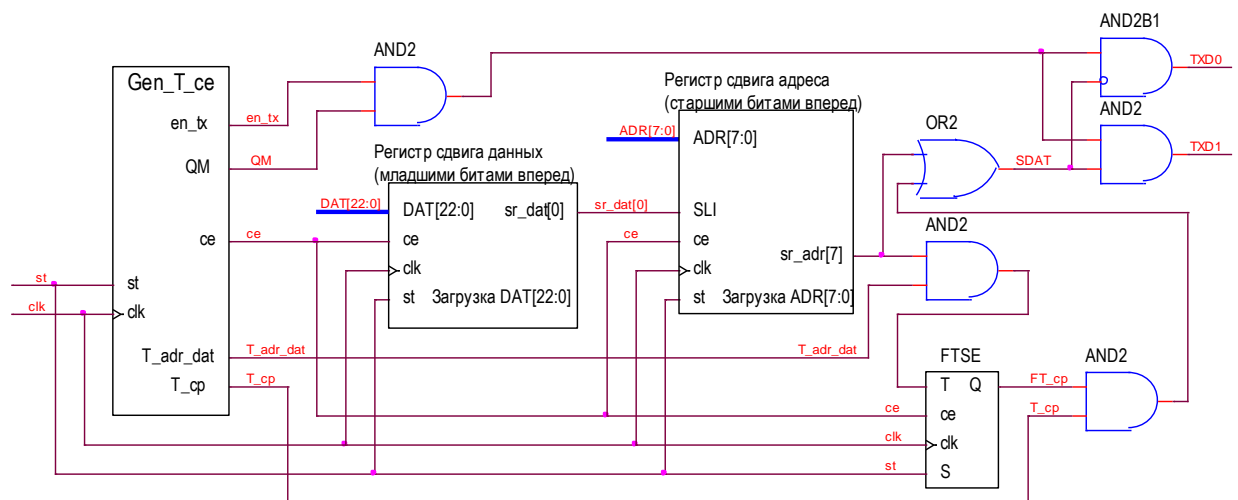


Рис.3 Структурная схема передатчика AR\_TXD

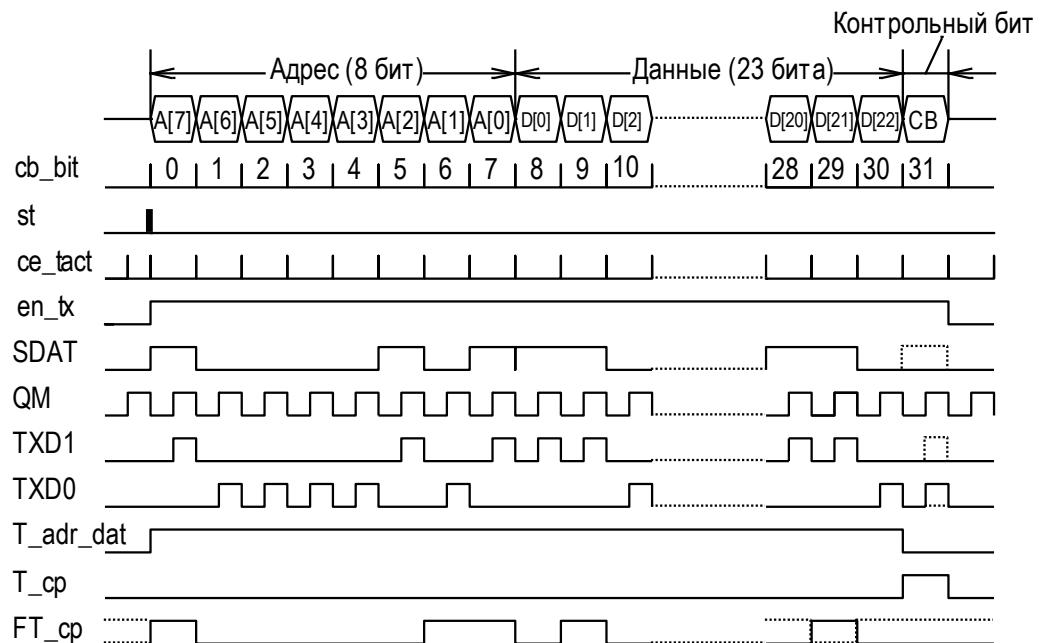


Рис.4 Структура ARINC слова и пример диаграмм сигналов передатчика AR\_TXD

Импульсы каналов TXD1 и TXD0 на интервале en\_tx ARINC слова формируются при помощи логических элементов AND2 из сигналов последовательных данных SDAT en\_tx и меандра QM с периодом T\_bit : TXD1= en\_tx & QM & SDAT, TXD0= en\_tx & QM & !SDAT.

Контрольный бит формируется при помощи T триггера, который импульсом старта st устанавливается в 1 (FT\_cp <= 1), а затем на интервале T\_adr\_dat переключается импульсами TXD1. На такте контрольного бита (T\_cp) последовательным данным SDAT присваивается значение FT\_cp. Если количество импульсов TXD1, т.е. количество переключений четное, то к последнему такту FT\_cp=1 и соответственно число единиц в SDAT нечетное. Если же количество импульсов TXD1 не четное, то к последнему такту FT\_cp=0, поэтому в этом случае число единиц в SDAT нечетное.

Ниже приведена схема модуля **AR\_TXD** ARINC передатчика, написанная на языке VERILOG.

```
module AR_TXD (
    input clk,                output wire ce_tact, // Скорость (Tce=1/Vel)
    input [1:0] Nvel,         output wire TXD1, //Импульсы канала 1
    input [7:0] ADR,         output wire TXD0, //Импульсы канала 0
    input [22:0] DAT,        output wire SLP, // Крутизна фронтов
    input st,                output reg en_tx =0, // Интервал передачи данных
                                output wire T_cp, // Такт контрольного бита
                                output reg FT_cp=0, // Триггер контроля четности
                                output wire SDAT, // Последовательные данные
                                output reg QM=0, // Модулятор
                                output reg[5:0]cb_bit=0, //Счетчик бит
                                output reg en_tx_word=0); /*Интервал передачи слова (32+4
                                                                такта)*/
parameter Fclk=50000000 ; //50 MHz
parameter V1Mb=1000000 ; // 1000 kb/s
parameter V100kb=100000 ; // 100 kb/s
parameter V50kb= 50000 ; // 50 kb/s
parameter V12_5kb=12500 ; // 12.5 kb/s
wire [10:0]AR_Nt = (Nvel [1:0]==3)? (Fclk/(2*V1Mb)) : //1000.000 kb/s
                  (Nvel [1:0]==2)? (Fclk/(2*V100kb)) : // 100.000 kb/s
                  (Nvel [1:0]==1)? (Fclk/(2*V50kb)) : // 50.000 kb/s
                  (Fclk/(2*V12_5kb)); // 12.500 kb/s

reg [10:0]cb_ce=0 ; // Счетчик полутакта
reg [ 7:0]sr_adr=0 ; // Регистр сдвига адреса
reg [22:0]sr_dat=0 ; // Регистр сдвига данных
wire ce = (cb_ce==AR_Nt); // Tce=1/(2*VEL)
assign ce_tact = ce & QM ; // Tce_tact=2* Tce =1/VEL
assign T_cp = (cb_bit==31) ; // Такт контрольного бита
wire T_adr_dat = en_tx & !T_cp ; // Интервал адреса и данных
```

```

wire ce_end_word = (cb_bit==35) & ce_tact ; //Импульс конца слова
assign SDAT = sr_adr[7] | (T_cp & FT_cp) ; //Последовательные данные
assign TXD1 = en_tx & QM & SDAT ; //Импульсы канала 1
assign TXD0 = en_tx & QM & !SDAT ; //Импульсы канала 0
assign SLP = (Nvel ==0) ; // Крутизна фронтов
wire start = st & !en_tx_word ; // Запрет запуска пока идет передача слова

always @ (posedge clk) begin
cb_ce <= (start | ce)? 1 : cb_ce+1 ;
QM <= start? 0 : (en_tx_word & ce)? !QM : QM ; //Переключение триггера модулятора
cb_bit <= start? 0 : (en_tx_word & ce_tact)? cb_bit+1 : cb_bit ; //Счет бит
en_tx_word <= start? 1 : ce_end_word? 0 : en_tx_word ; /*Формирование интервала передачи
с паузой*/
en_tx <= start? 1 : (T_cp & ce_tact)? 0 : en_tx ; //Формирование интервала передачи данных
FT_cp <= start? 1 : (sr_adr[7] & ce_tact & T_adr_dat)? !FT_cp : FT_cp ; /*Счет четности*/
sr_adr <= start? ADR : (ce_tact & en_tx)? sr_adr <<1 | sr_dat[0] : sr_adr ; /*Сдвиг адреса
старшими битами вперед*/
sr_dat <= start? DAT : (ce_tact & en_tx)? sr_dat >>1 : sr_dat ; /*Сдвиг данных младшими
битами вперед */
end
endmodule

```

Входами модуля являются:

clk – сигнал синхронизации с периодом Tclk=20 нс,

Nvel[1:0] – указатель скорости,

ADR[7:0] – 8 бит адреса,

DAT[22:0] – 23 бита данных,

st – положительный импульс запуска с длительностью Tclk.

Указатель скорости имеет 4 состояния, 4-е состояние Nvel[1:0]=3 обеспечивает не стандартную скорость 1000 кбит/сек, которую удобно использовать при моделировании и отладке схемы модуля.

Кроме необходимых сигналов TXD0 и TXD1 на выходные порты модуля выведены дополнительные сигналы, которые в данной работе предназначены для визуального контроля процесса работы внутренних элементов модуля при моделировании.

Сигнал SLP равен 1 только для самой малой скорости 12.5 кбит/сек.

Счетчик бит cb\_bit в модуле передатчика имеет не 5 а 6 разрядов для того, чтобы можно было сформировать en\_tx\_word, который имеет длительность не 32, а 36 тактов. Сигнал start = st & !en\_tx\_word предотвращает повторный запуск не только на интервале передачи данных, но и на интервале минимальной паузы в 4 длительности такта.

## 1. Задание к допуску (стоимость 2)

Получить от преподавателя номер набора параметров (Таблица 1), в который входят: скорость VEL (кбит/сек) адрес ADR[7:0] и данные DAT[22:0].

Таблица 1

№	Vel (kbit/sec)	ADR[7:0] (HEX,bin)	DAT[22:0] (HEX)
1	100	8'h81=8'b10000001	23'h123400
2	50	8'h82=8'b10000010	23'h567800
3	12.5	8'h83=8'b10000011	23'h789A00
4	100	8'h84=8'b10000100	23'h112200
5	50	8'h85=8'b10000101	23'h2A5500
6	12.5	8'h86=8'b10000110	23'h448800
7	100	8'h87=8'b10000111	23'h6E1100
8	50	8'h88=8'b10001000	23'h4C6600
9	12.5	8'h89=8'b10001001	23'h087700
10	100	8'h8A=8'b10001010	23'h1ABC00
11	50	8'h8B=8'b10001011	23'h32C300
12	12.5	8'h8C=8'b10001100	23'h26A500
13	100	8'h8D=8'b10001101	23'h702D00
14	50	8'h8E=8'b10001110	23'h70E100
15	12.5	8'h8F=8'b10001111	23'h633C00
16	100	8'h90=8'b10010000	23'h3C4D00
17	50	8'h91=8'b10010001	23'h702D00
18	12.5	8'h92=8'b10010010	23'h433C00
19	100	8'h93=8'b10010011	23'h433C00
20	50	8'h94=8'b10010100	23'h087700

Старший бит адреса у всех вариантов равен 1, а число, образованное младшими битами, совпадает с номером варианта. Единица старшего бита адреса позволяет отображать все слово на экране осциллографа в режиме ждущей развертки с запуском фронтом первого импульса TXD1 канала единиц.

**1.1** Начертить в тетради структурную схему модуля **AR\_TXD** (Рис.3).

**1.2** Переписать в тетрадь схему модуля **AR\_TXD**. Прочитать и записать все комментарии к регистрам и сигналам.

**1.3** Начертить в тетради эскиз временных диаграмм генератора **AR\_TXD** сигналов TXD1 и TXD0 для 8 бит адреса заданного варианта параметров.

**1.4** Предложить алгоритм приема ARINC слова. На основе предложенного алгоритма составить структурную схему модуля AR\_RXD приемника сигналов TXD1 и TXD0 модуля AR\_TXD. Желательно предусмотреть автоматический сброс счетчика бит в паузе между словами.

Для контроля четности в приемнике достаточно однобитным счетчиком FT\_cp (FT-триггером) начиная с нуля подсчитать число фронтов всех импульсов канала единиц. Если это число не четное, то на последнем такте  $T_{cp} = (cb\_bit\_rx == 31) \text{ FT\_cp} = 1$ . В качестве сигнала верности приема можно использовать  $ok\_rx = T_{cp} \& FT\_ct$ . В конце слова спадом  $T_{cp}$  или импульсом res надо FT\_cp “сбросить” в 0.

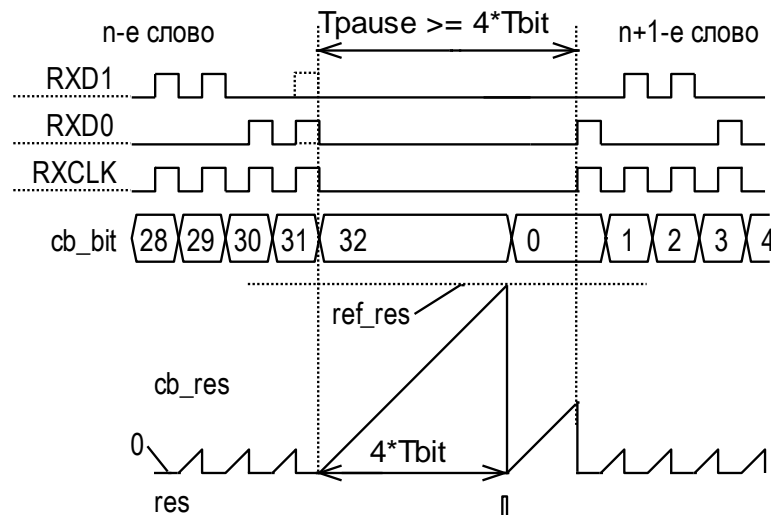


Рис.5 Временные диаграммы формирования сигнала сброса в паузе между словами

## 2. Задание к выполнению (стоимость 4)

Создать проект с именем Lab405AD, для ПЛИС, используемой в макете NEXYS-2.

**2.1** В окне источников (Sources) создать (New Source) модуль AR\_TXD генератора сигналов TXD1 и TXD0 стандарта ARINC-429 и далее на Verilog-е или в «схематике» составить схему этого модуля. Выполнить синтез (Synthesize - XST) созданного модуля. При необходимости исправить синтаксические ошибки.

Создать для этого модуля задание на моделирование (Verilog Test Fixture). Для входных сигналов установить заданные значения Nvel, adr и dat (см. таблицу 1).

Пример содержательной части задания на моделирование:

```
parameter Tclk=20; //Tclk=20ns
always begin clk=1'b0; #(Tclk/2) clk=1'b1; #(Tclk/2); end
initial begin
    st = 0; Nvel = 0; ADR = 0;          DAT = 0;
    #1005; st = 1; Nvel = 3; ADR = 8'b10000001; DAT = 23'h123400;
    #20;  st = 0; Nvel = 3; ADR = 8'b10000001; DAT = 23'h123400;
end
```

**2.1.1** Провести моделирование работы модуля AR\_TXD при заданных параметрах. Проверить правильность формирования контрольного бита. Зарисовать в тетради, полученные временные диаграммы сигналов.

**2.1.2** Проверить возможность повторного запуска до полного окончания передачи слова.

**2.2** На основе предложенного в пункте 1.4 задания к допуску алгоритма составить схему модуля AR\_RXD приемника сигналов TXD1 и TXD0 модуля AR\_TXD. Обязательными выводами окончательного варианта отлаженного модуля AR\_RXD должны быть входы: Inp1, Inp0 и выходы: регистр адреса (sr\_adr[7:0]), регистр данных (sr\_dat[23:0]) и сигнал (se\_wr) подтверждения правильности приема слова. При отладке могут потребоваться и другие выходы, поэтому в схеме Test\_Sch\_AR\_RXD моделирования работы приемника AR\_RXD более удобно использовать текстовое описание связей между модулями. «Текстовая» схема избавляет от необходимости редактировать или создавать новый символ при добавлении или удалении выводов. Например, при моделировании проектируемого приемника AR\_RXD необходимы:

- cb\_bit[4:0] – счетчик бит,

- T\_cp – интервал контрольного бита,
- en\_rx – интервал приема слова,
- FT\_cp – T-триггер контроля четности,
- ce\_wr – сигнал подтверждения правильного приема слова,
- cb\_res[15:0] – счетчик паузы,
- res – импульс сброса.

**2.3** Начертить в тетради временные диаграммы входных и выходных сигналов отлаженного модуля AR\_RXD.

#### 2.4 Пример схемы тестирования модуля приемника

```

module Test_Sch_AR_RAD(           //Выходы передатчика
    input clk,                     output wire ce_tx,      // Скорость (Tseo=1/Vel)
    input [1:0]Nvel,               output wire TXD1,   //Импульсы канала 1
    input [7:0]ADR,                output wire TXD0,   //Импульсы канала 0
    input [22:0]DAT,               output wire SLP,    // Крутизна фронтов
    input st,                      output wire en_tx_dat, // Интервал передачи данных
                                output wire T_cp_tx, // Такт конца кадра
                                output wire FT_tx,   // Триггер контроля четности
                                output wire SDAT,    // Последовательные данные
                                output wire QM,      // Модулятор
                                output wire [5:0]cb_bit_tx, //Счетчик бит передатчика
                                output wire en_tx_word, //Интервал передачи слова

    //Выходы приемника
    output wire[23:0] sr_dat_rx, //Регистр сдвига данных
    output wire[ 7:0] sr_adr_rx, //Регистр сдвига адреса
    output wire ce_wr //Разрешение записи принятого слова
    //output wire [4:0]cb_bit_rx // Счетчик бит приемника
    //output wire T_cp, // Интервал контрольного бита
    //output wire en_rx, //Интервал приема слова
    //output wire FT_cp_rx, //Т-триггер контроля четности
    //output wire ok_rx, //Правильный прием слова
    //output wire [15:0]cb_res, // Счетчик паузы
    //output wire res //Импульс сброса
);

// Модуль передатчика
AR_TXD DD1 (
    .clk(clk), .ce(ce_tx),           // Скорость (Tseo=1/Vel)
    .Nvel(Nvel), .TXD1(TXD1),        //Импульсы канала 1
    .ADR(ADR), .TXD0(TXD0),          //Импульсы канала 0
    .DAT(DAT), .SLP(SLP),            // Крутизна фронтов
    .st(st), .en_tx(en_tx_dat),      // Интервал передачи данных
    .T_cp(T_cp_tx),                  // Такт конца кадра
    .FT_cp(FT_tx),                   // Триггер контроля четности
    .SDAT(SDAT),                     // Последовательные данные
    .QM(QM),                         // Модулятор
    .cb_bit(cb_bit),                 //Счетчик бит
    .en_tx_word(en_tx_word)); //Интервал передачи слова

```

```

// Модуль приемника
AR_RXD DD2 ( .Inp0(TXD0),          .sr_dat(sr_dat_rx),
              .Inp1(TXD1),          .sr_adr(sr_adr_rx),
              .clk(clk),             .ce_wr(ce_wr)
              //cb_bit(cb_bit_rx),
              //T_cp(T_cp_rx),
              //en_rx(en_rx),
              //FT_cp(FT_cp_rx),
              // ok_rx(ok_rx),
              // cb_res(cb_res),
              //res(res)
              );

endmodule

```

### 3. Задание к сдаче работы (стоимость 4)

**3.1** Составить схему S\_Sch\_Lab405AD. Пример такой схемы приведен на рис.6. В состав схемы должны входить:

- модуль передатчика AR\_TXD,
- модуль приемника AR\_RXD,
- модуль семи сегментного индикатора DISPLAY для отображения передаваемых и принятых данных (приложение 4.1),
- модуль источника скорости адреса и данных ADR\_DAT\_BL (приложение 4.2),
- модуль регистра записи принятых адреса и данных BUF32bit (составить самостоятельно),
- модуль мультиплексора AR\_MUX передаваемых и принятых данных для DISPLAY (составить самостоятельно)

3.1.1 Составить схемы и символы этих модулей.

Модуль источника **ADR\_DAT\_BL** имеет вход BTN, логическая единица на котором инвертирует один бит данных, например, бит, номер которого совпадает с номером варианта. Это позволяет наглядно показывать реакцию контрольного бита на число единиц в слове.

Пример выделения модулем **AR\_MUX** 16 бит данных для модуля DISPLAY из 32 бит передаваемых и принятых ARINC слов передатчика и приемника:

- {DAT\_TX[7:0],ADR\_TX[7:0]},
- {1'b0,DAT\_TX[22:8]},
- {DAT\_RX[7:0],ADR\_RX[7:0]},
- {DAT\_RX[23:8]}.

Входные мультиплексоры M2\_1 при SW(3)=0 соединяют входы Inp0 и Inp1 приемника с выходами Tx0 и Tx1 передатчика без внешних проводных связей (JB1->JA1, JB7->JA7).

Кнопка BTN0 «сбрасывает» принятые данные.



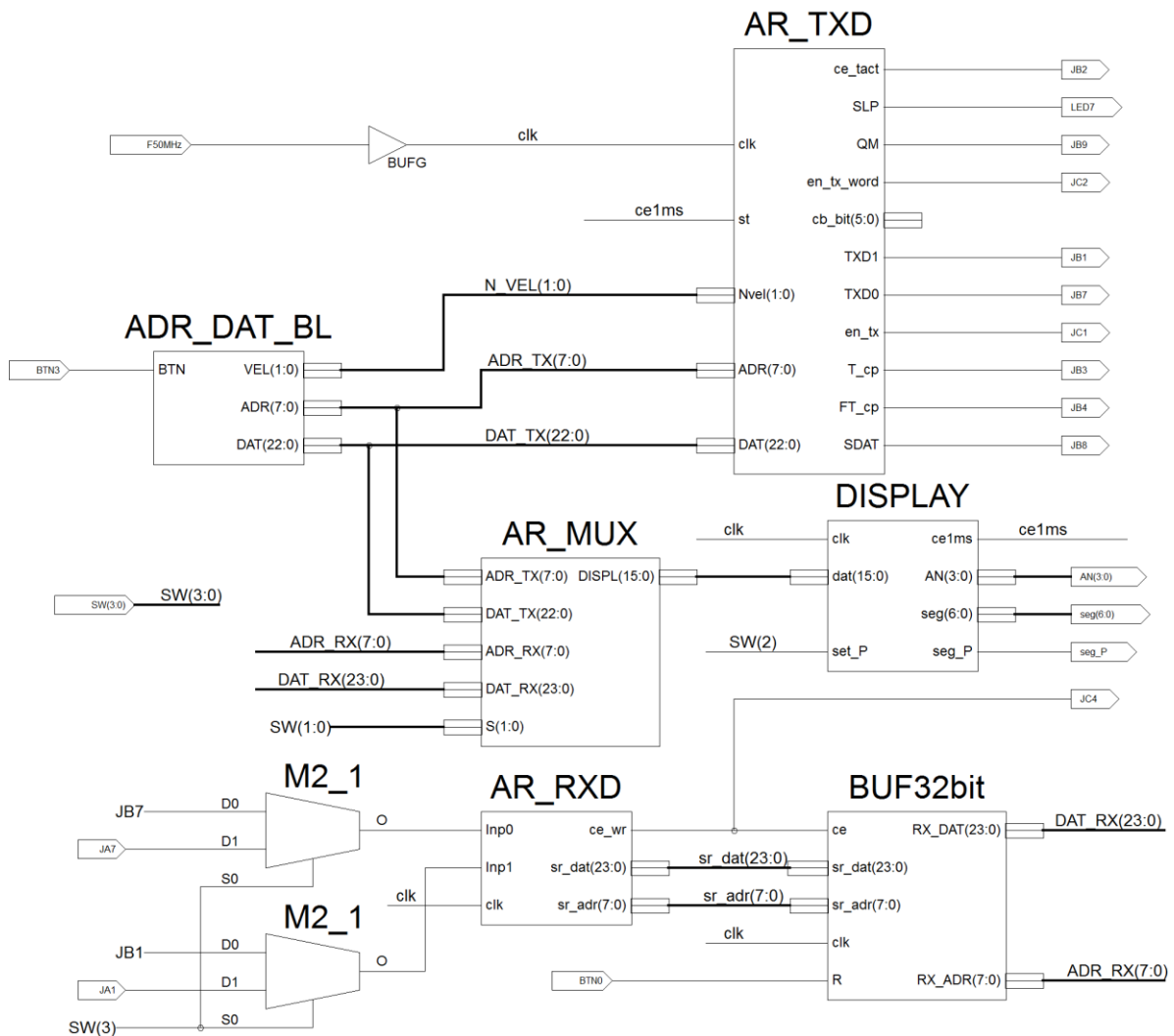


Рис.6 Пример схемы S\_Sch\_Lab405AD для сдачи работы

**3.2** Создать для схемы S\_Sch\_Lab405AD \*.ucf файл (приложение 4.3). Выполнить синтез и имплементацию схемы, загрузить в ПЛИС макета продемонстрировать работу схемы. Проверить реакцию приемника на кратковременный разрыв внутренней связи переключателем SW[3].

**3.3** Получить и сохранить осциллограммы сигналов SDAT, TXD1, TXD0, FT\_cp.

## 4. Приложения

### 4.1 Модуль семи сегментного индикатора

```
module DISPLAY( input clk,                output wire[3:0] AN, //Аноды
                input [15:0]dat,          output wire [6:0] seg, //Сегменты
                input set_P,              output wire seg_P, //Точка
                output reg ce1ms=0);

parameter Fclk=50000 ;    //50000 kHz
parameter F1kHz=1 ;      //1 kHz
wire [1:0]ptr_P = set_P? 2'b11 : 2'b00 ; //Точка справа или слева
reg [15:0] cb_1ms = 0 ;
```

```

wire ce = (cb_1ms==Fclk/F1kHz) ;
//-----Генератор сигнала ce1ms (период 1 мс, длительность Tclk=20 нс) -----
always @ (posedge clk) begin
    cb_1ms <= ce? 1 : cb_1ms+1 ;
    ce1ms <= ce ;
end
//----- Счетчик цифр -----
reg [1:0]cb_dig=0 ;
always @ (posedge clk) if (ce) begin
    cb_dig <= cb_dig+1 ;
end
//-----Переключатель «анодов»-----
assign AN = (cb_dig==0)? 4'b1110 : //включение цифры 0 (младшей)
            (cb_dig==1)? 4'b1101 : //включение цифры 1
            (cb_dig==2)? 4'b1011 : //включение цифры 2
            4'b0111 ; //включение цифры 3 (старшей)
//-----Переключатель тетрад (HEX цифр)-----
wire[3:0] dig =(cb_dig==0)? dat[3:0]:
               (cb_dig==1)? dat[7:4]:
               (cb_dig==2)? dat[11:8]: dat[15:12];
//-----Семисегментный дешифратор-----
//gfedcba
assign seg= (dig== 0)? 7'b1000000 ://0   a
            (dig== 1)? 7'b1111001 ://1 f|      |b
            (dig== 2)? 7'b0100100 ://2   g
            (dig== 3)? 7'b0110000 ://3 e|      |c
            (dig== 4)? 7'b0011001 ://4   d
            (dig== 5)? 7'b0010010 ://5
            (dig== 6)? 7'b0000010 ://6
            (dig== 7)? 7'b1111000 ://7
            (dig== 8)? 7'b0000000 ://8
            (dig== 9)? 7'b0010000 ://9
            (dig==10)? 7'b0001000 ://A
            (dig==11)? 7'b0000011 ://b
            (dig==12)? 7'b1000110 ://C
            (dig==13)? 7'b0100001 ://d
            (dig==14)? 7'b0000110 ://E
            7'b0001110 ;//F
//-----Указатель точки-----
assign seg_P = !(ptr_P == cb_dig) ;
endmodule

```

#### 4.2 Модуль адреса, данных и скорости

```

module ADR_DAT_BL(input BTN, output wire [1:0] VEL,

```

```
output wire [7:0] ADR,
output wire [22:0] DAT);
```

```
wire [4:0] my_nom = 5'b01101 ; //5'b01101=13 ;
assign ADR = {3'b100,my_nom} ;
parameter my_dat = 23'h702D00 ;
parameter my_VEL = 2'b00 ; // 12.5 kBod
wire my_bit_dat=(my_dat[13]^BTN) ;
assign DAT={my_dat[22:14],my_bit_dat,my_dat[12:0]} ;
assign VEL = my_VEL ;
endmodule
```

#### 4.3 Связь портов схемы с контактными площадками ПЛИС (файл \*.ucf)

```
NET "F50MHz" LOC = "B8" ; #clk
NET "AN<0>" LOC = "F17" ;
NET "AN<1>" LOC = "H17" ;
NET "AN<2>" LOC = "C18" ;
NET "AN<3>" LOC = "F15" ;
```

```
NET "BTN0" LOC = "B18" ; # RESET
#NET "BTN1" LOC = "D18" ;
#NET "BTN2" LOC = "E18" ;
NET "BTN3" LOC = "H13" ;
```

```
NET "seg<0>" LOC = "L18" ;
NET "seg<1>" LOC = "F18" ;
NET "seg<2>" LOC = "D17" ;
NET "seg<3>" LOC = "D16" ;
NET "seg<4>" LOC = "G14" ;
NET "seg<5>" LOC = "J17" ;
NET "seg<6>" LOC = "H14" ;
NET "seg_P" LOC = "C17" ; #DOT
```

```
NET "SW<0>" LOC = "G18" ;
NET "SW<1>" LOC = "H18" ;
NET "SW<2>" LOC = "K18" ;
NET "SW<3>" LOC = "K17" ;
#NET "SW<4>" LOC = "L14" ;
#NET "SW<5>" LOC = "L13" ;
#NET "SW<6>" LOC = "N17" ;
#NET "SW<7>" LOC = "R17" ;
#NET "LED0" LOC = "J14" ; #
#NET "LED1" LOC = "J15" ; #
#NET "LED2" LOC = "K15" ; #
#NET "LED3" LOC = "K14" ; #
#NET "LED4" LOC = "E17" ; #
#NET "LED5" LOC = "P15" ; #
#NET "LED6" LOC = "F4" ; #
NET "LED7" LOC = "R4" ; # SLP
```

```
#NET "TXD" LOC = "P9" ;
```

```

#NET "RXD" LOC = "U6" ;

NET "JA1" LOC = "L15" ; # Inp1
#NET "JA2" LOC = "K12" ;#
#NET "JA3" LOC = "L17" ;#
#NET "JA4" LOC = "M15" ;#
NET "JA7" LOC = "K13" ; # Inp0
#NET "JA8" LOC = "L16" ;#
#NET "JA9" LOC = "M14" ;#
#NET "JA10" LOC = "M16" ;#

NET "JB1" LOC = "M13" ; # TXD1
NET "JB2" LOC = "R18" ; # ce_tact
NET "JB3" LOC = "R15" ; # T_cp
NET "JB4" LOC = "T17" ; # FT_cp
NET "JB7" LOC = "P17" ; # TXD0
NET "JB8" LOC = "R16" ; # SDAT
NET "JB9" LOC = "T18" ; # QM
#NET "JB10" LOC = "U18" ;#

NET "JC1" LOC = "G15" ; #en_tx
NET "JC2" LOC = "J16" ; #en_tx_word
#NET "JC3" LOC = "G13" ;#
NET "JC4" LOC = "H16" ; # ce_wr
#NET "JC7" LOC = "H15" ;#
#NET "JC8" LOC = "F14" ;#
#NET "JC9" LOC = "G16" ;#
#NET "JC10" LOC = "J12" ;#

#NET "JD1" LOC = "J13" ;#
#NET "JD2" LOC = "M18" ;#
#NET "JD3" LOC = "N18" ;#
#NET "JD4" LOC = "P18" ;#
#NET "JD7" LOC = "K14" ;#
#NET "JD8" LOC = "K15" ;#
#NET "JD9" LOC = "J15" ;#
#NET "JD10" LOC = "J14" ;#

```

## 5. Контрольные вопросы

- 5.1 Чему равен период следования ARINC слов в схеме рис.6 для VEL=12.5 кбит.сек?
- 5.2 Можно ли составить схему ARINC приемника без сигнала синхронизации (clk Fclk=50MHz)?
- 5.3 Как зависит схема ARINC приемника от скорости передачи слова?
- 5.4 Можно ли в ARINC приемнике измерить скорость передачи бит?