Последовательный канал информационного обмена по стандарту **MIL-STD-1553**

(ГОСТ 26765.52-87, ГОСТ Р. 52070-2003)

Лабораторная работа №406_DX

Стандарт MIL-STD-1553, изначально разрабатывался по заказу МО США для использования в военной бортовой авионике. Впервые опубликован в США как стандарт ВВС в 1973 году, применён на истребителе F-16 . Принят в качестве стандарта НАТО — STANAG 3838 AVS. Позднее спектр его применения существенно расширился, стандарт стал применяться и в гражданских системах.

Данные передаются по витой проводной паре последовательно словами по 16 бит. Длительность каждого слова 20 мкс и состоит из 20 тактов по 1 мкс. В первые три такта передаются 2 импульса синхронизации с длительностью 1.5 мкс каждый. Затем в течение 16 тактов передаются 16 бит данных (D[15:0] - старшими битами вперед) и на последнем 20-м такте передается бит контроля четности (дополнение до нечетности числа единиц в слове). Полярность импульсов синхронизации определяется назначением слова. Например, в командном слове (CW) и в ответном слове (RW) первый импульс синхронизации положительный, а в слове данных (DW) – отрицательный. В качестве кода передачи используется биполярный фазоманипулированный код (Манчестер II). Биты данных передаются не потенциально, а перепадом напряжения в центре такта. Перепад напряжения от “+” к “–“ (⎾⏌) соответствует единице, а перепад от “–“ к “+” (⏌⎾) соответствует нулю. Размах напряжения на линии может быть в интервале от 1.4 В до 20 В. Пример временных диаграмм слов приведен на рис. 1.

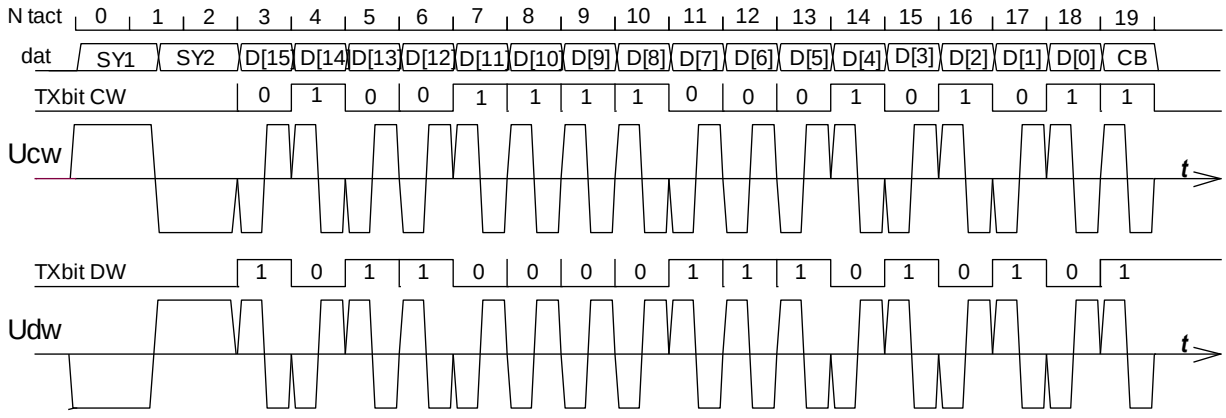

Рис.1 Пример временных диаграмм контрольного слова CW и слова данных DW стандарта MIL-STD-1553

Код Манчестер II является самосинхронизирующимся, т.е. он передает не только данные, но и эталон времени передатчика. В середине каждого такта данных обязательно имеются перепады напряжения, по которым можно принимать данные и синхронизировать эталон времени приёмника.

Слова данных передаются без промежутка (“впритык”) к командному слову или ответному слову так и между собой.

На первом этапе выполнения в данной работе в системе проектирования цифровых устройств на ПЛИС моделируется работа модуля MIL_TXD логических сигналов TXP и TXN контрольного слова и слова данных (см. рис.2). TXP соответствует положительным импульсам, а TXN – отрицательным импульсам канала связи.

На втором этапе выполнения на основе предложенных алгоритмов, временных диаграмм и структурных схем дается задание на проектирование и отладку модуля MIL_RXD приемника сигналов TXP и TXN.

На этапе сдачи работы для макета NEXYS-2 составляется схема, в состав которой входят отлаженные модули, и проверяется их работа.

1. Генератор сигналов MIL-1553

1.1 Модуль параметров CONST.v

```
`define Fclk  50000000      //50MHz - частота сигнала синхронизации NEXYS-2
`define Tclk   20           //20ns - период сигнала синхронизации NEXYS-2
`define Tsy    3000         //3000ns - длительность синхронизации
`define Tbit   1000         //1000ns – длительность бита (такта)
`define Ntact `Tbit/`Tclk   //1000/20=50  - число Tclk в Tbit
//`define Nzer     4        //2*Nzer - число Tclk в паузе между импульсами
//--Параметры для приемника, T_corr – интервал середины такта
`define set_M  `Ntact/2     //25 - центр такта
`define ref_H  `set_M+17    //42 - верхняя граница T_corr
`define ref_L  `set_M-5     //20 - нижняя граница T_corr
`define Nsy `Ntact+(`Ntact/2) //75 – число Tclk сигнала синхронизации
`define  ref_SY  `Nsy-22    //53 – порог обнаружения сигнала синхронизации
`define  ref_bit 25         //30 – порог считывания бита
```

1.2 Схема модуля имитатора сигналов MIL-1553

```
`include "CONST.v"
module MIL_TXD (    output wire T_SY,       //Интервал синхронизации
    input st,       output wire TXPN,       //Пара импульсов TXP, TXN в такте
    input clk,      output wire TXP,        //"Положительные"  импульсы
    input [15:0]CW, output wire TXN,        //"Отрицательные" импульсы
    input [15:0]DW, output reg en_tx=0,     //Интервал передачи
    input [2:0]Ndw, output reg [7:0] cb_tact=0,//Счетчик такта
    input [2:0]Nzer, output wire ce_tact,   //Границы тактов
                    output reg [4:0] cb_bit=0, //Счетчик бит
                    output wire ce_bit,     //Середины тактов
                    output wire T_dat,      //Интервал данных
                    output wire T_cp,       //Интервал контрольного бита
                    output reg FT_cp=0,     //Счетчик числа единиц
                    output wire TXbit,      //Биты данных с метками границ бит
                    output wire ce_end_W,//Конец слова
                    output reg QM=0,        //Модулятор
                    output wire QMD,        //Модулятор с данными
                    output reg [1:0]cb_DW=0,//Счетчик слов данных
                    output wire CW_DW); //Назначение слова

reg [15:0] sr_dat =0 ; //Регистр сдвига данных
```

```
assign CW_DW = (cb_DW==0) & en_tx ;   //Назначение слова
assign T_SY = (cb_bit==0) & en_tx ;          //Интервал синхронизации
wire [7:0]Ntact = T_SY? `Tsy/`Tclk : `Tbit/`Tclk ;//Число тактов 75 в T_SY или 50 в Tbit
assign ce_tact = (cb_tact==Ntact-1) ;          //Границы тактов
assign T_dat = ((cb_bit>=1) & (cb_bit<=16)) ;
assign T_cp = (cb_bit==17) & en_tx ;
assign ce_end_W = T_cp & ce_tact ;
wire ce_end_bl = ce_end_W & (cb_DW==Ndw) ;     //Конец передачи пакета слов
assign TXPN = ((cb_tact>Nzer) & (cb_tact<Ntact/2-Nzer)) |
                ((cb_tact>Ntact/2+Nzer) & (cb_tact<Ntact-Nzer)) ;
assign QMD = T_SY? (CW_DW ^ QM) :
                T_dat? (TXbit ^ QM) :
                T_cp? (FT_cp ^ QM) : 0 ;
assign TXP = TXPN &  QMD ;
assign TXN = TXPN & !QMD ;
assign ce_bit = (cb_tact==Ntact/2) ;
wire start = st & !en_tx ;   //Запрет старта пока не закончилась передача Ndw слов
assign TXbit = (sr_dat[15] & !T_SY) ;

always @ (posedge clk) begin
cb_DW <= st? 0 : ce_end_W? cb_DW+1 : cb_DW ;
cb_tact <= (start | ce_tact)? 0 : en_tx? cb_tact+1 : cb_tact ;
QM <= (start | ce_tact)? 0 : ce_bit? 1 : QM ;
cb_bit <= (start | ce_end_W)? 0 : ce_tact? cb_bit+1 : cb_bit ;
FT_cp <= (start | ce_end_W)? 1 : (T_dat & TXbit & ce_bit)? !FT_cp : FT_cp ;
en_tx <= start? 1 : ce_end_bl? 0 : en_tx ;
sr_dat <=   (T_SY & ce_tact & CW_DW)? CW :
            (T_SY & ce_tact & !CW_DW)? DW :
            (T_dat & ce_tact)? sr_dat<<1 : sr_dat ;end
endmodule
```

Модуль MIL_TXD рассчитан на передачу заданного контрольного слова и не более 7-и слов данных, первое из которых – заданное (см. таблицу 1). Вход Nzer[2:0] задает длительность паузы.

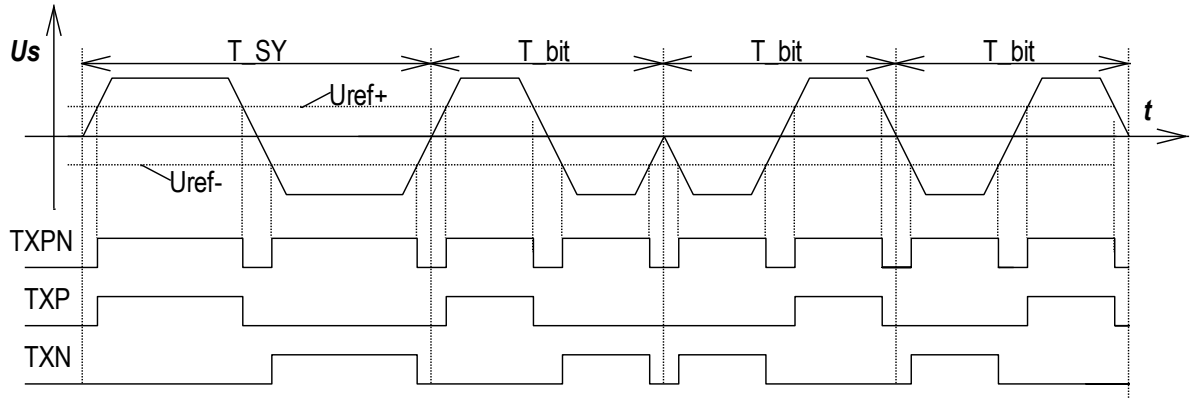

Рис.2 Пример временных диаграмм сигналов TXP, TXN и TXPN = (TXP | TXN) модуля **MIL_TXD**

Сигнал TXPN это пара импульсов на интервале синхронизации T_SY или на каждом бите данных. Первый (левый) импульс TXP имитирует сигнал выхода компаратора положительных импульсов канала, а второй (правый) TXN имитирует сигнал выхода

компаратора отрицательных импульсов канала. Импульсы разделены промежутками (паузами). Длительность пауз зависит от длительностей фронтов и соотношения порогов компараторов и амплитуд импульсов. Например, при максимальной длительности фронтов равной половине длительности бита и порогах компараторов равных половине амплитуды длительность паузы также равна половине длительности бита.

Импульсы TXPN разделяются на два канала TXP и TXN сигналом QMD, фаза которого для сигнала синхронизации определяется сигналом CW_DW назначения слова, а на интервале данных битом данных TXbit.

Счетчик слов данных cb_DW по сигналу старта st устанавливается в 0, а инкрементируется в конце каждого слова. Сигнал CW_DW = (cb_DW==0), определяющий формат импульсов синхронизации, только для первого слова равен 1, что обеспечивает передачу первым контрольного слова, а всех остальных слов в формате слова данных.

Триггер en_tx по сигналу старта st устанавливается в 1, а сбрасывается в 0 только в конце последнего слова данных.

Регистр сдвига данных sr_dat[15:0] при CW_DW=1 загружается данными контрольного слова, а при CW_DW=0 данными слова данных в конце сигнала синхронизации.

Триггер FT_cp является однобитным счетчиком числа единиц данных. В начале каждого слова он устанавливается в 1 и на интервале T_dat 16-и бит данных переключается столько раз, сколько единиц в слове, а на последнем такте T_cp передается его состояние, дополняющее количество единиц в полном слове до нечетного.

2. Приемник сигналов MIL-1553

2.1 Детектор сигналов синхронизации слов

Логические сигналы RXP и RXN, с которыми должен работать приемник это задержанные на один такт Tclk сигналы компараторов In_P и In_N. Для обнаружения сигнала синхронизации контрольного слова и слов данных предлагается сформировать сигналы D_RXP и D_RXN задержанные относительно RXP и RXN на Tsy=1.5 мкс. Импульсы логических произведений сигналов (RXP & D_RXN) и (RXN & D_RXP) только на интервале второго импульса синхронизации имеют длительность 1.5 мкс, остальные импульсы имеют длительность не более 1 мкс. Поэтому для обнаружения сигнала синхронизации достаточно на интервале T_SY=(D_RXP & RXN) | (D_RXN & RXP) при помощи счетчика cb_SY измерять, например, в единицах Tclk длительности этих импульсов, а в паузах между ними удерживать в 0. При отсутствии пауз максимальное значение числа на счетчиках на интервале второго импульса синхронизации равно 75 (1500ns/20ns=75), а на остальных интервалах не более 50 (1000ns/20ns=50). Если это число, например, превышает ref_SY=75-**12**=63, то сигнал синхронизации будет обнаружен: ok_SY = (cb_SY >= ref_SY). Здесь **12** это целая часть половины разности максимальных значений (75-50)/2.

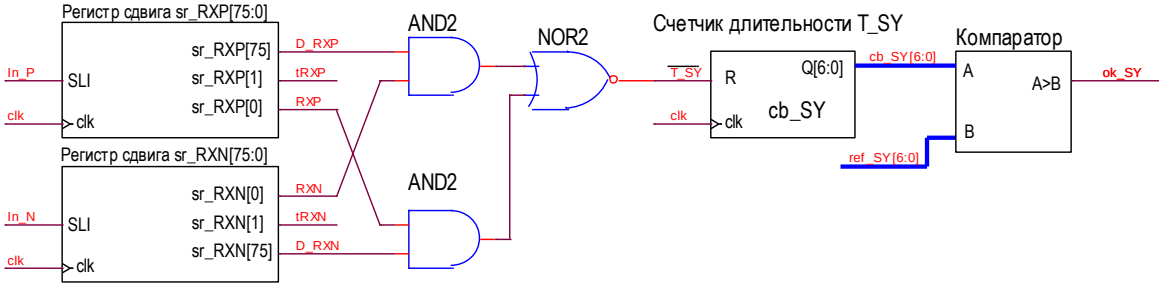

Рис. 3 Структурная схема обнаружителя сигналов синхронизации контрольного слова и слова данных

Паузы снижают максимальные значения cb_SY. Например, при длительности паузы 10*Tclk=200ns максимальные значения равны 65 и 40 поэтому при выбранном выше значении порога 63 сигнал синхронизации будет обнаружен.

При выборе порога сигнала синхронизации надо учитывать не только возможные паузы, но и не одинаковость эталонов времени передатчика и приемника. Например, если период сигнала синхронизации приемника на 5% (21ns) превышает период сигнала синхронизации передатчика (20ns), то при паузе 200ns максимальные значения cb_SY будут равны 62 и 38 поэтому при выбранном выше пороге 63 сигнал синхронизации не будет обнаружен. При отсутствии пауз и периоде сигнала синхронизации приемника, например, меньшей на 5% (19ns), чем период сигнала синхронизации передатчика (20ns) максимальные значения cb_SY будут равны 79 и 53. Таком образом, в описанных условиях нижняя граница порога синхронизации равна 53. Поэтому, если рассчитывать на максимальную относительную разность эталонов времени передатчика и приемника $\pm 5\%$ и максимальную паузы 200ns, то порог REF_SY можно установить в середине допустимого диапазона REF_SY=(62+53)/2=57.

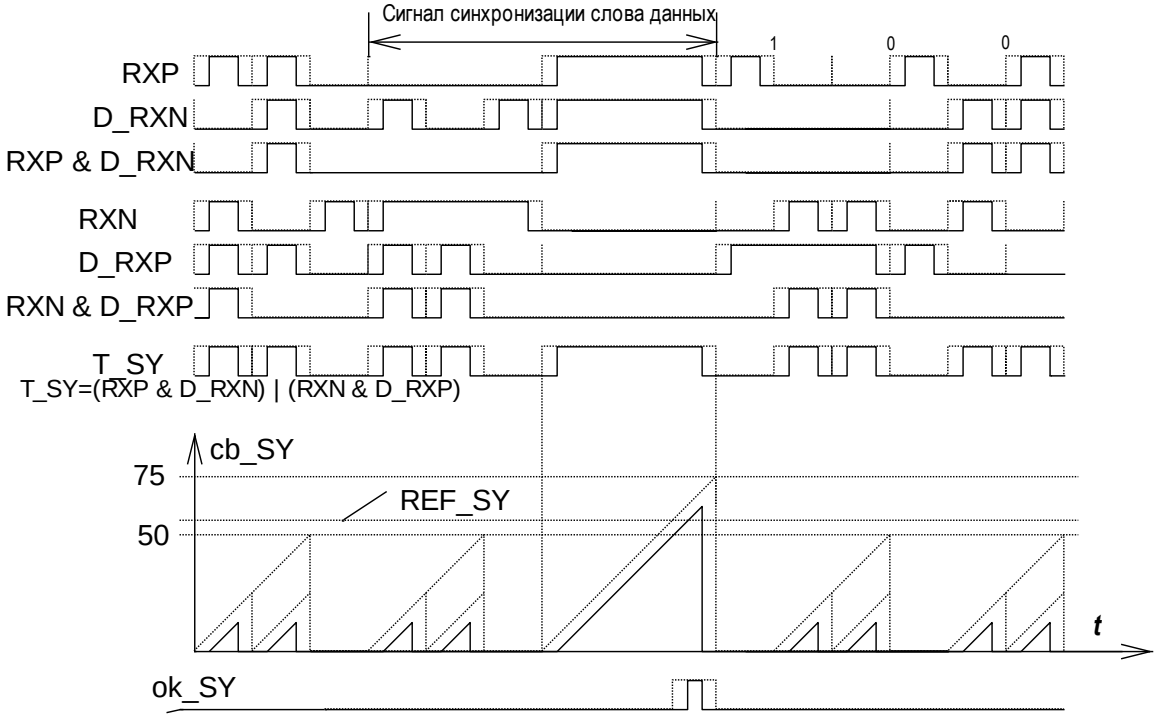

Рис. 4 Пример временных диаграмм детектора сигнала синхронизации (пунктирные линии соответствуют сигналам RXP и RXN без пауз)

2.2 Детектор сигналов данных слов

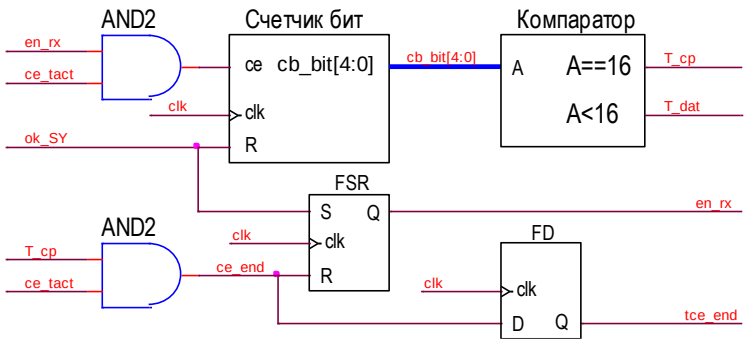

Рис. 5 Структурная схема формирователя сигналов интервалов времени: en_rx, T_dat, T_cp

В этой схеме интервал приема данных en_rx формируется SR триггером, который сигналом ok_SY устанавливается в 1, а сигналом конца слова ce_end=((cb_bit==16) & ce_tact) сбрасывается в 0. Счетчик бит cb_bit[4:0] сигналом ok_SY сбрасывается в 0 и на интервале en_rx инкрементируется с тактом ce_bit.

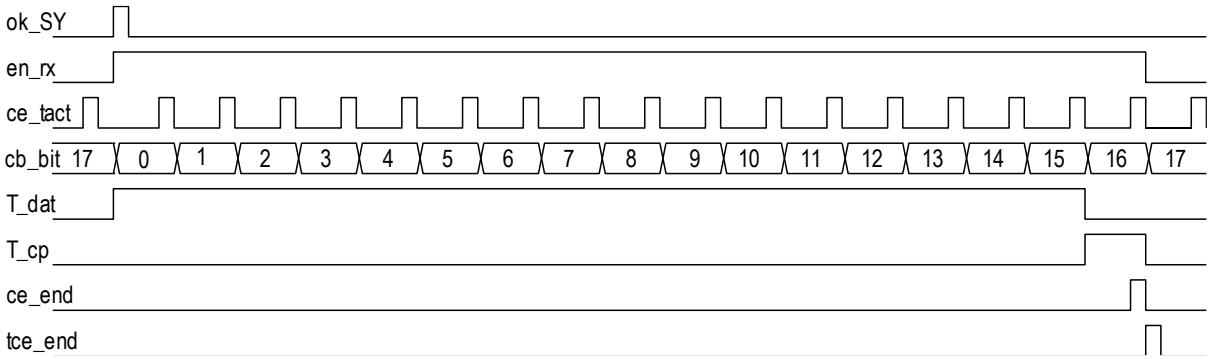

Рис.6 Временные диаграммы сигналов: en_rx, T_dat, T_cp, ce_end и tce_end

Сигнал tce_end необходим для формирования сигнала правильного приема слова ok_rx: ok_rx <= tce_end & FT_cp & (cb_bit==17), где FT_cp счетчик числа единиц в слове, включая контрольный бит.

Счетчик такта приемника cb_tact[6:0] сигналом сбрасывается в 0 как сигналом ok_SY, так и периодически сигналом границы такта (ce_tact=(cb_tact==Tbit/Tclk-1)).

Не одинаковость эталонов времени приемника и передатчика сказывается и на декодировании бит данных. Например, если Tclk приемника на ±5% (21ns/19ns) больше или меньше Tclk передатчика (20ns), то к последнему 17-му такту контрольного бита середина такта приемника сместится более чем на половину длительности такта передатчика, что практически исключит возможность правильного определения значений последних бит слова.

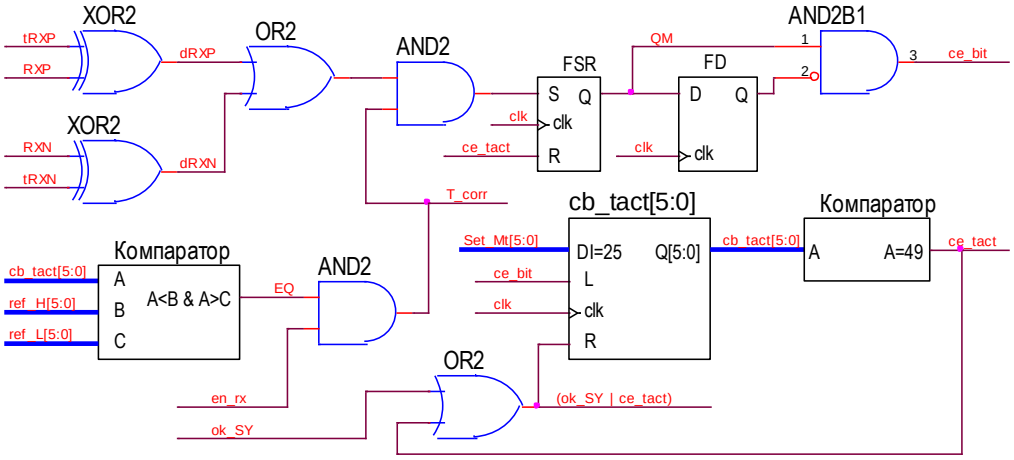

Рис. 7 Структурная схема коррекции счетчика cb_tact[5:0]

В середине такта передатчика обязательно есть один или два (при наличии пауз) перепада сигналов RXP или RXN. Если на каждом такте первым из этих перепадов устанавливать счетчик cb_tact в номинальное значение set_M=25, соответствующее середине такта, то ошибка смещения не будет накапливаться и на последнем такте T_cp будет достаточно малой даже при 5% относительной разности эталонов времени

передатчика и приемника. Для отделения перепадов около границ такта необходимо сформировать окно T_corr середины такта:

T_corr=en_rx & (cb_tact>=`ref_L) & (cb_tact<=`ref_H), где ref_L и ref_H левая и правая границы окна коррекции (см. модуль параметров CONST.v).

Сигнал QM формируется RS триггером, который устанавливается в 1 перепадом в окне T_corr и сбрасывается в 0 в конце такта. Сигнал ce_bit это фронт QM.

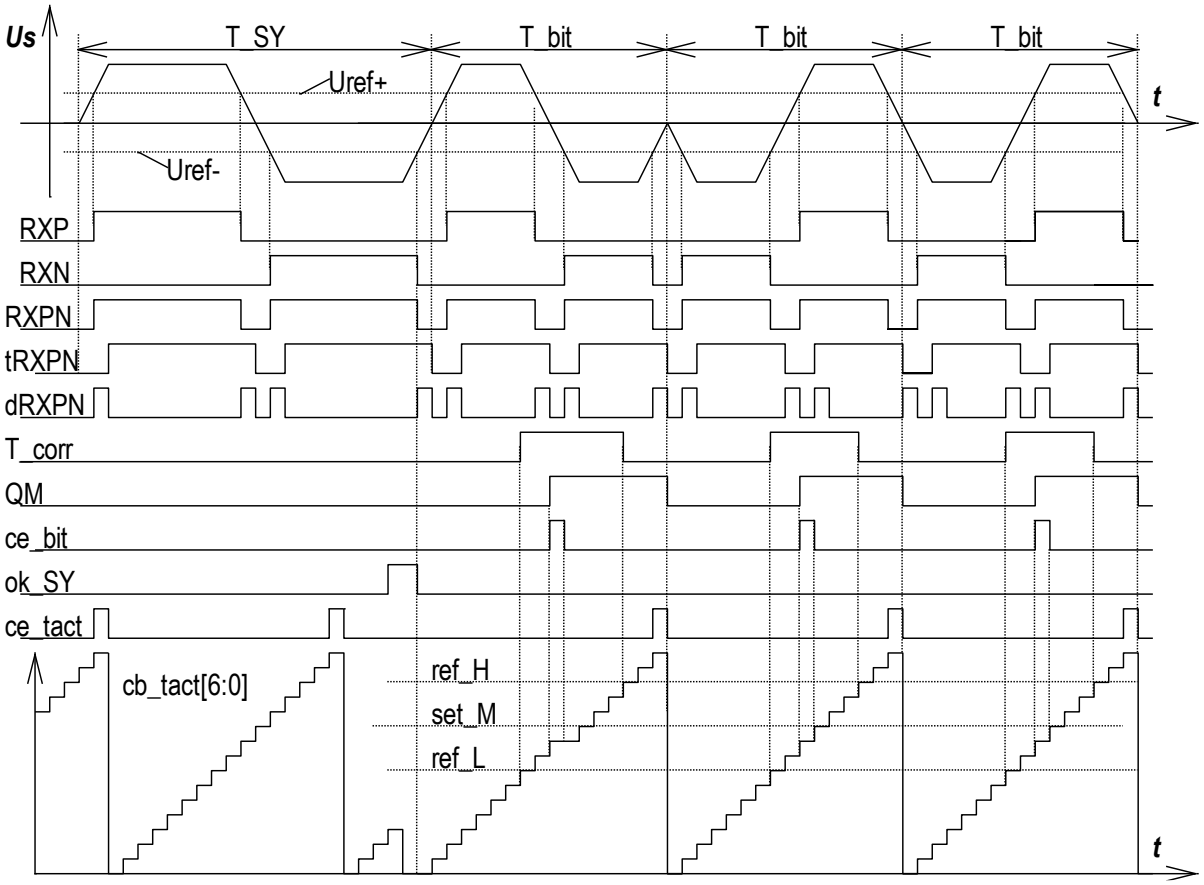

Рис. 8 Временные диаграммы коррекции счетчика cb_tact[5:0]

Сигнал QM имеет положительный перепад в середине такта, что соответствует сигналу RXP для bit=0 и отсутствии пауз, поэтому для определения значения бита достаточно сравнить RXP с QM, QMX=QM^RXP. Если RXP совпадает с QM (QMX=0), то bit=0, иначе (QMX=1) bit=1.

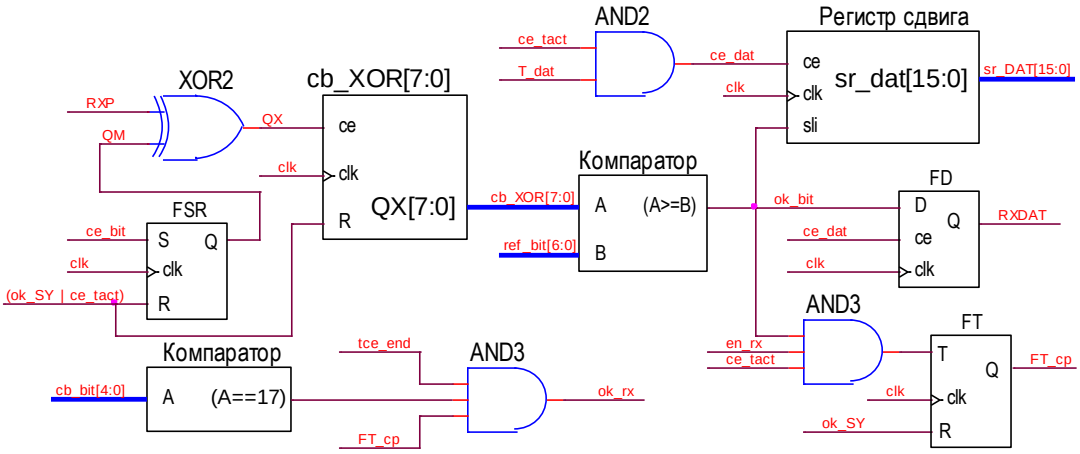

Рис. 9 Структурная схема декодера бит

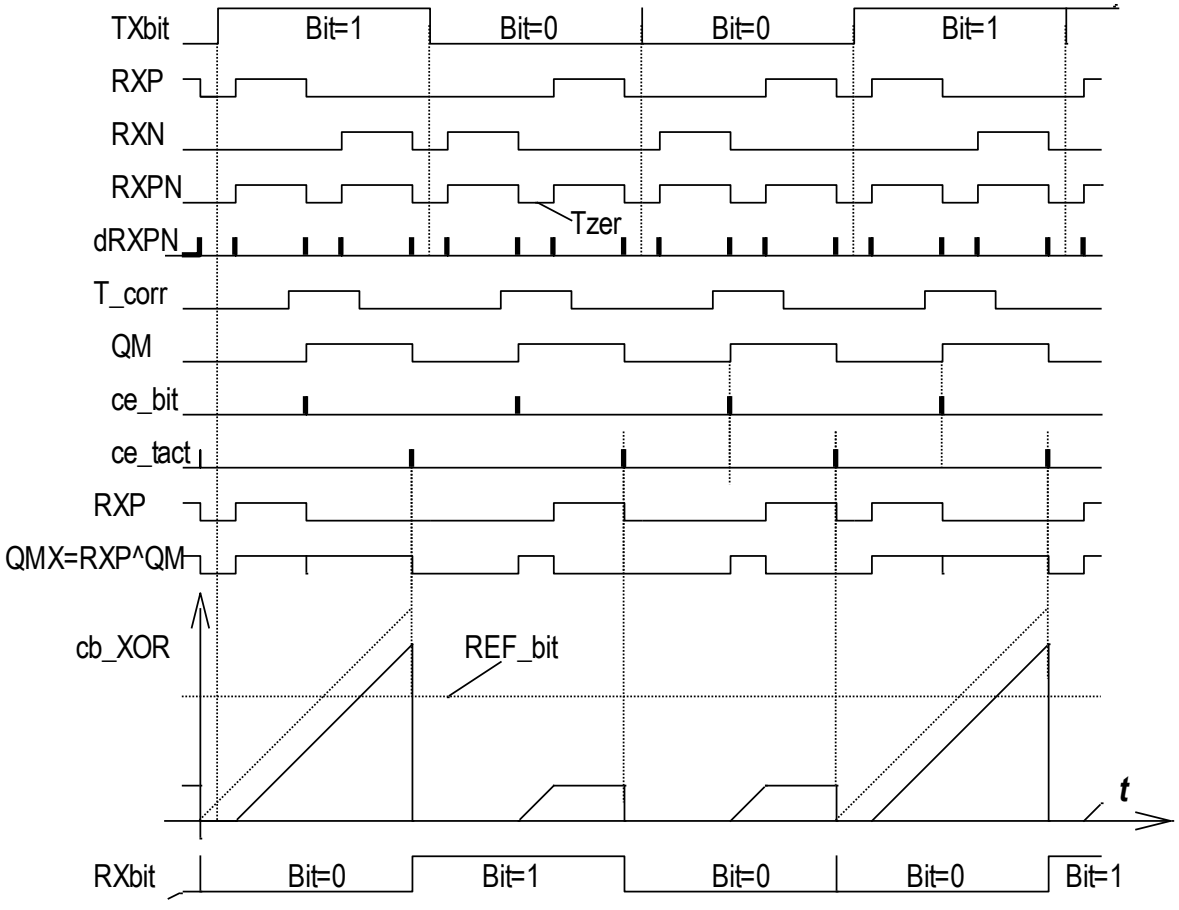

Рис.10 Пример временных диаграмм декодера бит

Длительность интервала не совпадения RXP с QM определяется счетчиком cb_XOR, который на границах тактов сбрасывается в 0, а на интервалах не совпадения (QMX=1) инкрементируется с тактом Tclk. При отсутствии пауз для bit=1 в конце такта cb_XOR=50, а для bit=0 cb_XOR=0. При наличии пауз для bit=1 в конце такта cb_XOR=50-2*Nzer, а для bit=0 cb_XOR=2*Nzer, где Nzer число тактов половины паузы между импульсами. Порог REF_bit должен быть меньше 50-2*Nzer и больше 2*Nzer. Например, если порог REF_bit это среднее значение, REF_bit = (50-2*Nzer +2*Nzer)/2=25, то maxNzer<=12.

Для контроля правильности принятого слова удобно использовать однобитный счетчик FT_cp (Т-триггер), которым подсчитывается количество единиц в слове включая контрольный бит. Если в начале каждого слова сигналом ok_SY триггер FT_cp устанавливать в 0, то при нечетном числе 1 в слове после окончания приема слова FT_cp переключившись нечетное число раз будет равен 1, что и будет означать отсутствие ошибки контроля четности в принятом слове.

Сигнал ok_rx разрешения считывания слова ok_rx = FT_cp & tce_end, где tce_end импульс после приема последнего бита.

RS-триггер CW_DW – указатель принятого слова должен устанавливаться в 1 сигналом ok_SY & RXN, а сбрасываться в ноль после приема контрольного слов сигналом ok_SY & RXP.

3. Задание к допуску (стоимость 2)

3.1 Получить от преподавателя номер набора параметров (из таблицы 1), в который входят: контрольное слово CW, слово данных DW и число тактов половины паузы Nzer.

Таблица 1

№	CW (HEX)	DW (HEX)	Nzer
1	1234	5678	1
2	5678	789A	2
3	9ABC	6523	3
4	DEF0	2233	4
5	FEDC	55AA	5
6	BA98	8811	6
7	7654	1188	5
8	3210	6699	4
9	1122	7711	3
10	3344	BCDE	2
11	5566	C3A5	1
12	7788	A587	2
13	6699	2D0F	3
14	AA55	E178	4
15	CC33	3C5A	5
16	00FF	4D20	6
17	FF00	55AA	5
18	F0F0	CC33	4
19	0FF0	4DD4	3
20	F00F	8181	2

3.2 Начертить в тетради временные диаграммы сигналов модуля **MIL_TXD**: TXbit, TXP, TXN, и FT_cp для Nzer=0 и заданных CW и DW (см. рис.1, рис.2 и таблицу 1).

3.3 Переписать в тетрадь схему модуля **MIL_TXD**.

4. Задание к выполнению (стоимость 5)

Создать проект с именем **Lab406_DX**, для ПЛИС XC3S500E-5FG320, используемой в макете NEXYS-2. В окне **Properties** проекта в строке **Simulator** - выбрать **ISim** в дальнейшем моделирование проводить в **Simulate Behavioral Model**.

4.1. В окне источников (Sources) создать (New Source) модуль **MIL_TXD** имитатора сигналов TXP и TXN стандарта MIL-1553STD.

4.2. Провести моделирование работы модуля MIL_TXD для заданных слов, параметра паузы Nzer=0, Nzer=7 и Nzer своего варианта. Проверить соответствие полученных временных диаграмм временным диаграммам рисунка пункта 3.2 задания к допуску. Измерить длительности импульсов и пауз.

Для этого создать задание на моделирование (Verilog Test Fixture) tf_MIL_TXD. В содержательной части создать генератор сигнала синхронизации clk.

always begin clk = 0; #10; clk = 1; #10; end

В блоке “initial begin…end” после #1000 установить заданные значения контрольного слова CW, слова данных DW своего варианта и числа слов данных Ndw=2. Например:

```
always begin clk=0; #10; clk=1; #10; end
        initial begin
            st=0; CW= 0;            DW=0;              Ndw=0;     Nzer=0;
#1000;  st=1; CW=16'h4E72;  DW=16'hB18D;    Ndw=2;
#20;    st=0;
#20000; Nzer=4; //Данные своего варианта
#20000; Nzer=7; //Максимальная пауза
        end
```

4.3 Используя описанные выше алгоритмы декодирования сигнала синхронизации и бит данных, а также структурные схемы и временные диаграммы составить схему модуля MIL_RXD приемника контрольного слова и слов данных.

4.4 Для отладки созданного модуля **MIL_RXD** необходимо составить модуль схемы **Sch_test_MIL_RXD**, в которую входят два модуля: **MIL_TXD** и **MIL_RXD**.

```
module Sch_test_MIL_RXD (
                //-----Вход и выходы MIL_TXD--------------------
        input st,               output wire T_SY_tx,      //Интервал синхронизации
        input clk_tx,           output wire TXP,          //"Положительные"   импульсы
        input [15:0]CW_tx,      output wire TXN,          //"Отрицательные" импульсы
        input [15:0]DW_tx,      output wire en_tx,        //Интервал передачи
        input [2:0]Ndw,         output wire [4:0]cb_bit_tx,//Счетчик бит
        input clk_rx,           output wire TXbit,        //Биты данных
        input [2:0]Nzer,        output wire [2:0]cb_DW_tx, //Счетчик слов данных
        input res,              output wire TXPN,         //Импульсы TXP or TXN
                                output wire CW_DW_tx,     //Назначение слова
                //-----Вход и выходы MIL_RXD--------------------
                        output wire T_SY_rx,   //Интервал синхронизации
                        output wire ok_SY,     //Есть сигнал синхронизации
```

```
                output wire[5:0]cb_tact,//Счетчик такта
                output wire ce_tact,      //Границы бит
                output wire en_rx,        //Интервал приема данных
                output wire[4:0]cb_bit_rx, //Счетчик бит
                output wire dRXPN,     //Все перепады импульсов In_P,In_N
                output wire ce_bit,       //Перепады RXPN в центре бита
                output wire T_corr,       //Интервал коррекции
                output wire RXbit,        //RXDAT с ok_SY и границами бит
                output wire[15:0]sr_dat,        //Регистр сдвига данных
                output wire ce_wr_DW,        //Сигнал правильного приема слова
                output wire [1:0]cb_DW_rx,  //Счетчик слов данных
                output wire CW_DW_rx,       //Указатель слова
                output wire [15:0] CW_rx);    //Принятое контрольное слово

        MIL_TXD DD1 (
            .st(st),              .T_SY(T_SY_tx),
            .clk(clk_tx),         .TXP(TXP),
            .CW(CW_tx),           .TXN(TXN),
            .DW(DW_tx),           .en_tx(en_tx),
            .Ndw(Ndw),            .cb_bit(cb_bit_tx),
            .Nzer(Nzer),          .TXbit(TXbit),
                                  .cb_DW(cb_DW_tx),
                                  .TXPN(TXPN),
                                  .CW_DW(CW_DW_tx));
        MIL_RXD DD2(
            .In_P(TXP),           .T_SY(T_SY_rx),
            .In_N(TXN),           .ok_SY(ok_SY),
            .clk(clk_rx),         .cb_tact(cb_tact),
            .res(res),            .en_rx(en_rx),
                                  .cb_bit(cb_bit_rx),
                                  .ce_tact(ce_tact),
                                  .dRXPN(dRXPN),
                                  .ce_bit(ce_bit),
                                  .T_corr(T_corr),
                                  .sr_dat(sr_dat),
                                  .ce_wr_DW(ce_wr_DW),
                                  .RXbit(RXbit),
                                  .cb_DW(cb_DW_rx),
                                  .CW_DW(CW_DW_rx),
                                  .CW(CW_rx) );
        endmodule

//Сигналы синхронизации clk_tx и clk_rx модулей передатчика и приемника должны быть разные.
// Сигнал синхронизации передатчика
parameter PTX = 20.0 ;          //Период tx_clk
always begin clk_tx = 1'b0; #(PTX/2); clk_tx = 1'b1; #(PTX/2);  end
// Варианты  сигнала синхронизации приемника
parameter PRX = 20.0 ;     // Период clk_rx = 20.0
//parameter PRX = 21.0 ;   // Период clk_rx = 21.0
```

```
//parameter PRX = 19.0 ;  // Период clk_rx = 19.0
always begin clk_rx = 1'b0; #(PRX/2); clk_rx = 1'b1; #(PRX/2);  end   initial begin
        st = 0;    CW_tx = 0;            DW_tx = 0;              Ndw = 0;       Nzer = 0;    res=0;
#1000;st = 1;  DW_tx = 16'h0000; CW_tx = 16'hB18D;    Ndw = 4;       Nzer = 7;     res=0;
#20;  st = 0;
#20000;         DW_tx = 16'h1111;
#20000;         DW_tx = 16'h2222;
#20000;         DW_tx = 16'h3333;
#20000;         DW_tx = 16'h4444;
   end
endmodule
```

4.5 При clk_rx=20ns и заданном Nzer отладить созданный модуль NIL_RXD. Сравнить полученные временные диаграммы моделирования с временными диаграммами рисунков рис.8 и рис.10. Провести моделирование при заданных значениях CW и DW для трех вариантов периода сигнала синхронизации приемника clk_rx.

4.6 Определить диапазон допустимой относительной разности периодов сигналов синхронизации модулей MIL_TXD и MIL_RXD.

В схеме рис.11 модуль DAC_2bit это формирователь сигналов {X1,X0]} для 2-х битного цифроаналогового преобразователя (ЦАП), который подключается к выводам портов JA1 и JA7 макета NEXYS-2 (см. рис 12). Схема и временные диаграммы ЦАП приведены на рис.13.

Данные контрольного слова CW и 4-х слов данных: DW0, DW1, DW2, DW3 формируются в модуле Gen_CW_4DW. Номер передаваемого слова DW_tx задается счетчиком cb_DW[1:0] модуля MIL_TXD. Номер отображаемого слова DW задается переключателями SW[7:6].

ПереключательSW[2] управляет мультиплексорами **M2_1.** При SW[2]=0 сигналы TXP и TXN подаются непосредственно с выходов модуля MIL_TXD, а при SW[2]=1 через перемычки J1 и J2 (см.рис.12) между портами JA3=>JA9 и JA4=>JA10.

Модуль **Bf_4DW** сигналу ce_wr_DW по адресу cb_DW записывает в 4 регистра принятые слова данных, а по адресу SW[7:6] выдает одно из них для индикации.

Мультиплексор **MUX16_4_1** предназначен для передачи на семи сегментный индикатор DISPLAY передаваемых слов CW, DW и принимаемых слов CW_rx, DW_rx. Адрес отображаемых данных задается переключателями SW[1:0].

Светодиод LED7 включается сигналом en_tx передатчика MIL_TXD, а светодиод LED0 сигналом en_rx приемника MIL_RXD.

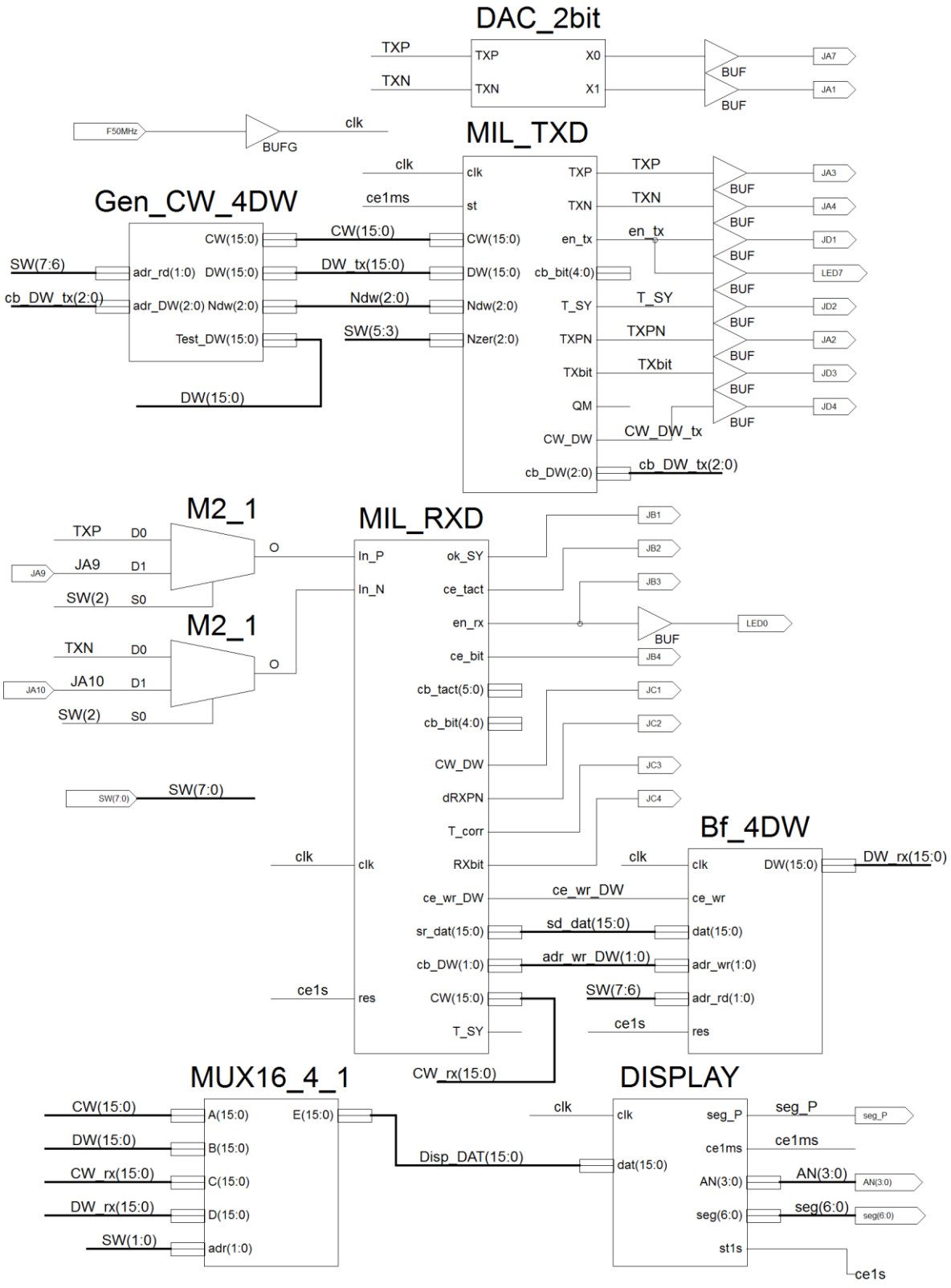

Рис.11 Пример схемы S_Sch_Lab406_DX для сдачи работы

5. Задание к сдаче работы (стоимость 3)

5.1 Создать модули и символы: Gen_CW_4DW, Bf_4DW, DISPLAY, MUX16_4_1 и DAC-2bit (см. Приложения 5.1-5.4).

5.2 Для загрузки в макет NEXYS-2 составить схему S_Sch_LAB406_DX (см., например, рис.11). В состав схемы, кроме MIL_TXD и MIL_RXD должен входить и модуль отображения передаваемых и принимаемых данных (Display), модуль импульса запуска и заданных вариантов слов (Gen_txen_DAT), буфер принятых данных (BUF_RX_DAT) и мультиплексор данных для индикатора (MUX16_4_1).

5.3 Для схемы Sch_LAB406 создать Implementation Constraints File (см. Приложение 5.5).

5.8 Создать файл конфигурации S_Sch_LAB406_DX.bit (Generate Programming File) или *.mcs (Generate PROM, ACE, or JTAG File), загрузить его в макет. Продемонстрировать при помощи осциллографа работу передатчика. Проверить соответствие осциллограмм сигналов TXdat, TXP и TXN показаниям CW и DW семи сегментного индикатора. Сохранить осциллограммы сигналов sdat, TXP и TXN.

5.9 Соединить выходы передатчика со входами приемника (можно проводные связи заменить внутренними т.е. SW[2]=0). Сопоставить показания индикатора передаваемых и принимаемых данных. Провести при помощи осциллографа наблюдение выведенных сигналов приемника. При необходимости отладить схему приемника. После достижения правильной работы приемника сохранить осциллограммы его сигналов.

6. Приложения

6.1 Модуль двухбитного цифроаналогового преобразователя

```
module DAC_2bit(
          input TXP,     output X1,
          input TXN,     output X0);
wire [1:0] DAC ;
assign X1=DAC[1], X0=DAC[0] ;
assign DAC = ({TXP,TXN}==2'b00)? 2'b01 :
                ({TXP,TXN}==2'b01)? 2'b00 :
                ({TXP,TXN}==2'b10)? 2'b10 : 2'b00 ;
endmodule
```

6.2 Модуль контрольного слова и слов данных

```
module Gen_CW_4DW(        output wire [15:0] CW,
          input [2:0] adr_DW,     output wire [15:0] DW,
                                  output wire [2:0] Ndw);
assign Ndw=4 ;
assign CW = 16'h1234 ;                      //My CW из таблицы 1
assign DW =   (adr_DW==1)? 16'h5678 :   //My DW из таблицы 1
              (adr_DW==2)? 16'hDB02 :
              (adr_DW==3)? 16'hDB03 :
              (adr_DW==4)? 16'hDB04 : 16'hFFFF ;
```

```
endmodule
```

6.3 Буфер принятых 4-х слов данных

```
module Bf_4DW(
        input [15:0] dat,          output wire [15:0] DW,
        input clk,
        input res,
        input ce_wr,
        input [1:0] adr_wr,
        input [1:0] adr_rd);

reg [15:0] DW0=0, DW1=0, DW2=0, DW3=0 ;
assign DW = (adr_rd==0)? DW0 :
                (adr_rd==1)? DW1 :
                (adr_rd==2)? DW2 : DW3 ;

always @ (posedge clk) begin
DW0 <= res? 0 : ((adr_wr==0) & ce_wr)? dat : DW0 ;
DW1 <= res? 0 : ((adr_wr==1) & ce_wr)? dat : DW1 ;
DW2 <= res? 0 : ((adr_wr==2) & ce_wr)? dat : DW2 ;
DW3 <= res? 0 : ((adr_wr==3) & ce_wr)? dat : DW3 ;
end
endmodule
```

6.4 Модуль семи сегментного индикатора данных

```
module DISPLAY(  input clk,          output wire[3:0] AN,    //Аноды
                 input [15:0]dat,    output wire[6:0] seg,   //Сегменты
                                     output wire seg_P,      //Точка
                                     output reg ce1ms=0,
                                     output reg st1s=0);
wire [1:0] ptr_P = 2'b00 ;  //Точка справа
parameter Fclk=50000 ;    //50000 kHz
parameter F1kHz=1 ;       //1 kHz
reg [15:0] cb_1ms = 0 ;
wire ce = (cb_1ms==Fclk/F1kHz) ;
reg [9:0]cb_1s =0 ;
wire ce1s = (cb_1s==1000) & ce ;
//-------Генератор сигнала ce1ms (период 1 мс, длительность Tclk=20 нс) ----------
always @ (posedge clk) begin
cb_1ms <= ce? 1 : cb_1ms+1 ;
ce1ms <= ce ;
cb_1s <= ce1s? 1 : ce? cb_1s+1 : cb_1s ;
st1s <= ce1s ;
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
```

5.5 Распределение сигналов по контактным площадкам ПЛИС (файл *.ucf)

```
NET "F50MHz"  LOC = "B8"  ; #F50MHz

NET "AN<0>"  LOC = "F17"  ; #AN0
NET "AN<1>"  LOC = "H17"  ; #AN1
NET "AN<2>"  LOC = "C18"  ; #AN2
NET "AN<3>"  LOC = "F15"  ; #AN3

NET "seg<0>"  LOC = "L18"  ; #CA
NET "seg<1>"  LOC = "F18"  ; #CB
NET "seg<2>"  LOC = "D17"  ; #CC
NET "seg<3>"  LOC = "D16"  ; #CD
NET "seg<4>"  LOC = "G14"  ; #CE
NET "seg<5>"  LOC = "J17"  ; #CF
NET "seg<6>"  LOC = "H14"  ; #CG
NET "seg_P"  LOC = "C17"  ; #CP

NET "SW<0>"  LOC = "G18"  ; #adr[0]
```

```
NET "SW<1>"  LOC = "H18"  ; #adr[1]
NET "SW<2>"  LOC = "K18"  ; #EXT/INT
NET "SW<3>"  LOC = "K17"  ; #Nzer[0]
NET "SW<4>"  LOC = "L14"  ; # Nzer[1]
NET "SW<5>"  LOC = "L13"  ; # Nzer[2]
NET "SW<6>"  LOC = "N17"  ; #Ndw[0]
NET "SW<7>"  LOC = "R17"  ; # Ndw[1]

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

NET "LED0"  LOC = "J14"  ; #en_rx
#NET "LED<1>"  LOC = "J15"  ; #LD1
#NET "LED<2>"  LOC = "K15"  ; #LD2
#NET "LED<3>"  LOC = "K14"  ; #LD3
#NET "LED<4>"  LOC = "E17"  ; #LD4
#NET "LED<5>"  LOC = "P15"  ; #LD5
#NET "LED<6>"  LOC = "F4"  ; #LD6
NET "LED7"  LOC = "R4"  ; #en_tx

#NET "TXD"  LOC = "P9"  ; #TXD P9
#NET "RXD"  LOC = "U6"  ; #TXD U6

NET "JA1"  LOC = "L15"  ; #DAC[1]
NET "JA2"  LOC = "K12"  ; #TXPN
NET "JA3"  LOC = "L17"  ; #TXP
NET "JA4"  LOC = "M15" ; #TXN
NET "JA7" LOC = "K13" ; #DAC[0]
#NET "JA8" LOC = "L16"  ; #
#NET "JA9"  LOC = "M14"  ;#
#NET "JA10" LOC = "M16"  ;#

NET "JB1"  LOC = "M13"  ; #ok_SY
NET "JB2"  LOC = "R18"  ; #ce_tact
NET "JB3"  LOC = "R15"  ; #en_rx
NET "JB4"  LOC = "T17"  ; #ce_bit
#NET "JB7"  LOC = "P17"  ; #
#NET "JB8" LOC = "R16" ; #
#NET "JB9"  LOC = "T18" ;#
#NET "JB10" LOC = "U18" ;#

NET "JC1"  LOC = "G15"  ; # CW_DW_rx
NET "JC2"  LOC = "J16"  ; # dRXPN
NET "JC3"  LOC = "G13"  ; #T_corr
NET "JC4"  LOC = "H16"  ; #RXbit
#NET "JC7" LOC = "H15"  ; #
#NET "JC8" LOC = "F14"  ; #
#NET "JC9"  LOC = "G16"  ; #
```

```
#NET "JC10" LOC = "J12"  ; #

NET "JD1"  LOC = "J13"  ; #en_tx
NET "JD2"  LOC = "M18"  ; #T_SY
NET "JD3"  LOC = "N18"  ; #TXbit
NET "JD4"  LOC = "P18"  ; #CW_DW_tx
#NET "JD7"   LOC = "K14"  ; #LD3
#NET "JD8"   LOC = "K15"  ; #LD2
#NET "JD9"  LOC = "J15"  ; #LD1
#NET "JD10" LOC = "J14"  ; #LD0
```

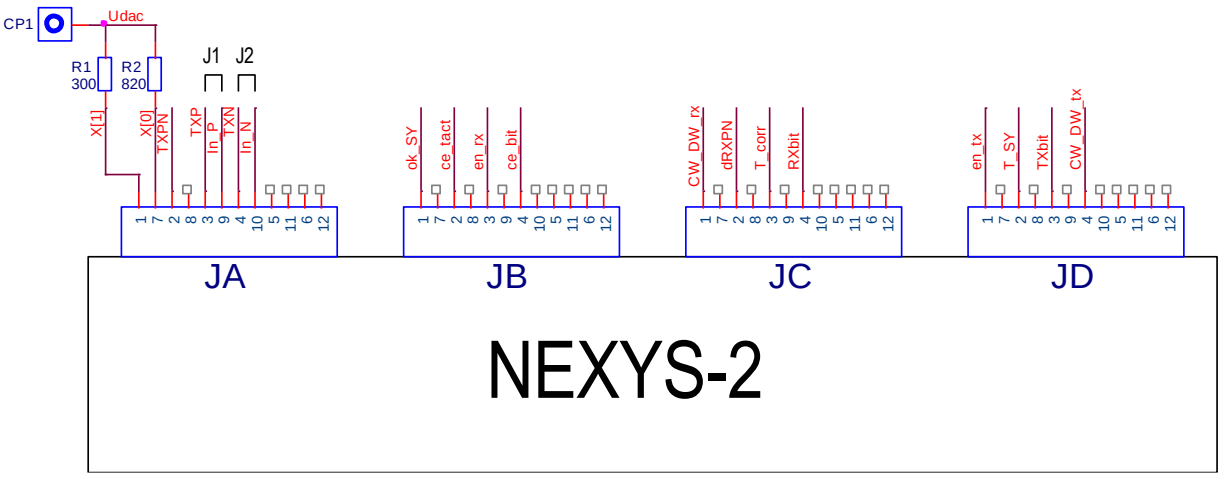

Рис.12 Распределение сигналов по выводам макета NEXYS-2

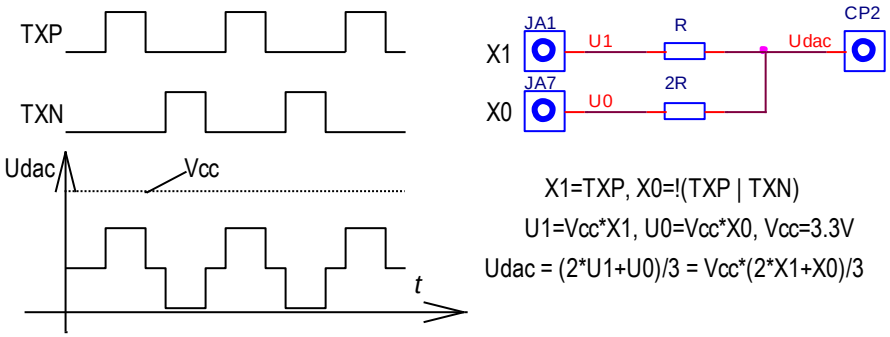

Рис.13 Временные диаграммы и схема 2-х битного цифроаналогового преобразователя

7. Контрольные вопросы

7.1 Оценить предельно допустимую относительную разность периодов эталонов времени передатчика и приемника.

7.2 Влияет ли небольшая не симметрия сигналов RXP и RXN на качество работы дешифратора?

7.3 Влияет ли наличие промежутков между соответствующими фронтами и спадами сигналов RXP и RXN на качество работы приемника?

7.4 Можно ли в принципе без линии задержки декодировать сигнал синхронизации кадра MIL-1553?

7.5 Как определить, чему соответствуют принятые данные: контрольному слову (CW) или слову данных (DW)?