

Лабораторная работа №406D

В данной работе осваиваются методы проектирования цифровых модулей модулятора передатчика и дешифратора сигналов стандарта MIL-1553 (ГОСТ 26765.52-87, ГОСТ Р. 52070-2003).

Формат сигнала, аналоговые методы его формирования и дешифрации изучались в лабораторных работах №406A_V1 и №406A_V2. В приложении 5.1 приведен вариант цифровой схемы шифратора сигналов MIL-1553, написанный на языке Verilog и предназначенный для реализации на программируемой логической интегральной схеме (ПЛИС) макета Nexys2. Выходами модулятора передатчика являются сигналы TXP и TXN. Запускается модулятор фронтом сигнала txen. Если высокий уровень txen сохраняется и после окончания передачи контрольного слова, то следующими передается слова данных. Сигналом к смене данных DAT на входе модулятора может служить сигнал конца кадра (t_end).

В соответствии со стандартом размер кадра - 20 бит. Первые 3 бита – (3 мкс) сигнал синхронизации (два импульса по 1.5 мкс разной полярности), затем 16 бит данных и последний бит в кадре – дополнение до нечетности. Слово данных (WD) отличается от контрольного слова (CW) только полярностью сигнала синхронизации. Контрольное слово (CW) начинается с положительного, а слово данных (WD) с отрицательного импульса синхронизации. На рис.1 показано слово данных.

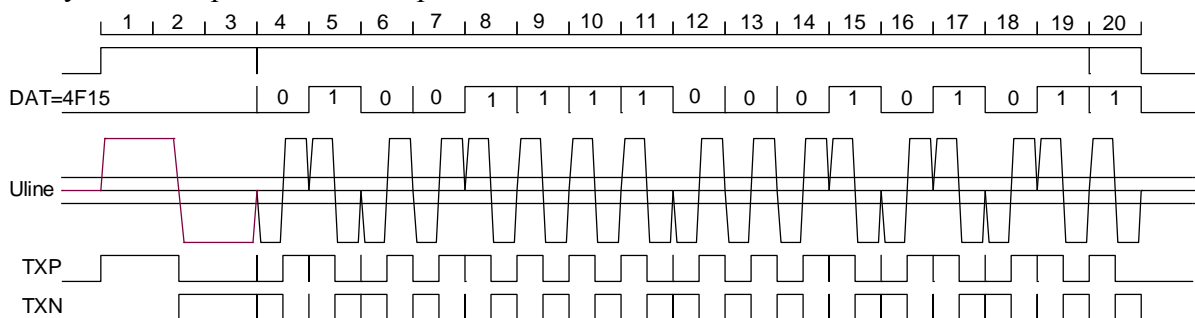


Рис.1 Временные диаграммы сигналов MIL-1553

Получить от преподавателя номер набора параметров (из таблицы 1), в который входят: контрольное слово CW и слово данных WD .

Таблица 1

№	CW (HEX)	WD (HEX)
1	1234	5678
2	5678	789A
3	9ABC	6523
4	DEF0	2233
5	FEDC	55AA
6	BA98	8811
7	7654	1188
8	3210	6699
9	1122	7711
10	3344	BCDE
11	5566	C3A5
12	7788	A587
13	6699	2D0F
14	AA55	E178
15	CC33	3C5A
16	00FF	4D20

17	FF00	55AA
18	F0F0	CC33
19	0FF0	4DD4
20	F00F	8181

1. Задание к допуску

1.1. Начертить или написать в тетради схему модуля MIL_TXD модулятора MIL-1553 (см. приложение 5.1).

1.2. Начертить в тетради эскиз временных диаграмм сигналов SY1, SY2, sdat, fd_cp, TXP и TXN модуля модулятора MIL_TXD для заданных CW и WD (см. таблицу 1).

2. Задание к выполнению

Создать проект с именем Lab406D, для ПЛИС, используемой в макете Nexys или Nexys2. Если на ПК установлен Modelsim и в окне Properties проекта в строке Simulator - выбрать Modelsim и в дальнейшем моделирование рекомендуется проводить в Simulate Post Translate Verilog Model. Если на ПК не установлен Modelsim, то в окне Properties проекта в строке Simulator – выбрать ISE Simulator.

2.1. В окне источников (Sources) создать (New Source) создать модуль MIL_TXD генератора сигналов TXP и TXN стандарта MIL-1553 и далее на Verilog-е или в «схематике» составить схему этого модуля (см. приложение 5.1).

Создать, если необходимо, схемотехнический символ модуля (Design Utilites/Create Schematic Symbol).

Создать для этого модуля задание на моделирование (Test Bench Waveform или Verilog Test Fixture). Параметры периодического сигнала синхронизации clk в Verilog Test Fixture можно задать через параметр период

parameter PERIOD = 20;

always begin clk1 = 1'b0; #(PERIOD/2) clk1 = 1'b1; #(PERIOD/2); end

или непосредственно, через длительности 0 и 1 (в наносекундах):

always begin clk1 = 1'b0; #10 clk1 = 1'b1; #10; end

В блоке “initial begin...end” вначале (#100) установить заданное значение контрольного слова и через и с задержкой ~10 мкс (#10000) слова данных. Суммарная длительность сигнала txen должна быть больше длительности контрольного слова. Например:

initial begin

clk1= 0; dat = 0; txen = 0; //

#100; dat = 16'h1234; txen = 1; // CW

#10000; dat = 16'h5678; txen = 1; // WD

#24000; dat = 16'h0000; txen = 0;

end

Установить заданные значения контрольного слова и слова данных.

2.2. Провести моделирование работы модуля для заданных слов (см. например, Verilog Test Fixture в 5.2 Приложений). При моделировании в ISE Simulator-е рекомендуется использовать режим Simulate Behavioral Model. При этом все регистры модулей должны быть инициализированы (например, приравнены к нулю). Подкорректировать, если необходимо, схему и временные диаграммы входных сигналов. Установить удобную для моделирования скорость. Получить содержательные временные диаграммы (например, такие как 5.3 приложений). Зарисовать полученные временные диаграммы.

3. Задание к сдаче работы

3.1. Составить схему модуля MIL_RXD приемника-дешифратора сигналов TXP и TXN модулятора. При составлении схемы дешифратора можно воспользоваться «идеями» схемам дешифратора MIL_RXD из лабораторной работы Lab406A.

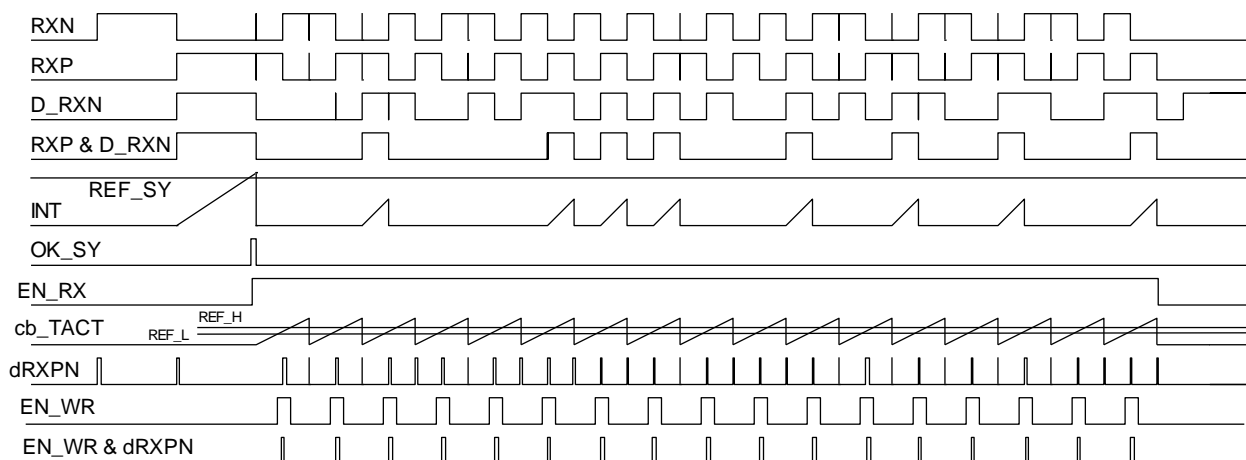


Рис.2 Временные диаграммы сигналов возможного варианта дешифратора (Lab104A_V1)

На этом рисунке:

DRXN – задержанный на 1.5 мкс сигнал RXN,

INT – интегратор (счетчик),

OK_SY – импульс превышения порога ($INT > REF_SY$),

EN_RX- разрешение приема,

cb_TACT – счетчик тактов с емкостью ($T_{tact}/T_{clk} = 1 \text{ мкс} / 20 \text{ нс} = 50$),

dRXPN – импульсы всех перепадов входных сигналов,

EN_WR - интервал разрешения реагирования на перепада EN_WR= ($REF_L < cb_TACT < REF_H$).

3.2. Провести предварительное моделирование работы фрагментов схем модуля MIL_RXD. Подкорректировать, если необходимо, схему и временные диаграммы входных сигналов.

3.3. Составить для моделирования схему SCH1_LAB406D, состоящую из последовательно соединенных модулей MIL_TXD и MIL_RXD (см. например 5.4 приложений).

3.4. Создать файл задания для моделирования (Verilog Test Fixture см. например 5.5 Приложений). Провести моделирование работы схемы. Подкорректировать, если необходимо, схемы модулей и временные диаграммы сигналов. Получить содержательные временные диаграммы (например, такие как 5.6 приложений). Начертить в тетради эскизы полученных временных диаграмм.

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

3.6 Для загрузки в макет Nexys2 составить схему SCH2_LAB406D (см. например 5.7 Приложений). В состав схемы, кроме MIL_TXD и MIL_RXD должен входить и модуль (Display) отображения передаваемых и принимаемых данных. Выводы модулей MIL_TXD и MIL_RXD можно вывести на разъёмы JA, JB, JC, JD макета, например, так, как приведено в приложении 5.8.

3.7 Для схемы SCH2_LAB406D создать Implementation Constraints File (см. 5.7 Приложений).

3.8 Создать файл конфигурации SCH2_LAB406D.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 sdat, TXP и TXN показаниям семи сегментного индикатора. Зарисовать осциллограммы сигналов sdat, TXP и TXN (см. например, 5.9 приложений: осциллограммы сигналов sdat и TXP модуля MIL_TXD).

3.9 Соединить выходы передатчика со входами приемника. Сопоставить показания индикатора передаваемых и принимаемых данных. Провести при помощи осциллографа наблюдение выведенных сигналов дешифратора. При необходимости отладить схему дешифратора. После достижения правильной работы дешифратора зарисовать осциллограммы его сигналов.

4. Контрольные вопросы

- 4.1 Осенить предельно допустимую относительную разность периодов эталонов времени передатчика и дешифратора.
- 4.2 Влияет ли небольшая не симметрия сигналов RXP и RXN на качество работы дешифратора?
- 4.3 Влияет ли наличие промежутков между соответствующими фронтами и спадами сигналов RXP и RXN на качество работы дешифратора?
- 4.4 Можно ли использовать для декодирования сигнала синхронизации кадра MIL-1553 линию задержки (регистр сдвига SR_SY[m-1:0], m=1500ns/20ns) и счетчик длительности?
- 4.5 Как можно определить, чему соответствуют принятые данные: контрольному слову (CW) или слову данных (WD)?

5. Приложения

5.1 Схема модуля модулятора передатчика MIL-1553

```
module MIL_TXD
    ( input clk,                output wire TXP,
      input[15:0]dat,          output wire TXN,
      input txen,              output wire SY1,
                               output wire SY2,
                               output reg en_tx=0,
                               output reg T_dat=0,
                               output wire T_end,
                               output wire sdat,
                               output reg fd_cp=0,
                               output reg [4:0]cb_bit )=0;

    parameter cp_tx = 1; //Дополнение до нечетности
    parameter TXvel = 1000000 ; // 1MHz
    parameter Fclk = 50000000 ; // 50 MHz
    reg [5:0]cb_cet =0;
    reg bf_SY1=0, bf_SY2=0, CW_WD=0, ttxen=0, q =0;
    wire start = ttxen & !en_tx ;
    assign SY1 = en_tx & bf_SY1 ; // Первый импульс синхронизации
    assign SY2 = en_tx & bf_SY2 ; // Второй импульс синхронизации
    reg [15:0]sr_dat=0 ;// Регистр сдвига данных
    wire st1 = (cb_bit==1) & en_tx ;
```

```

wire st2 = (cb_bit==2) & en_tx ;
wire st18 = (cb_bit==18) & en_tx; //Конец интервала данных
assign T_end = (cb_bit==19) & en_tx; //Конец кадра, бит контроля четности
assign TXP = en_tx & (( CW_WD & SY1)      |
                      (!CW_WD & SY2)      |
                      (T_dat & (sr_dat[15]^q)) |
                      (T_end & (fd_cp^q))) ;
assign TXN = en_tx & ((!CW_WD & SY1)      |
                      (CW_WD & SY2)      |
                      (T_dat & (sr_dat[15]^!q)) |
                      (T_end & (fd_cp^!q))) ;
assign sdat = sr_dat[15] & T_dat ; // Последовательные данные
assign cet = (cb_cet == (Fclk/(2*TXvel)));

always @ (posedge clk) begin
cb_cet <= cet? 1 : cb_cet+1 ;
q <= cet? !q : q ; // Триггер меандра модулятора последовательных данных
end
always @ (posedge clk) if (cet) begin
bf_SY1 <= (st1 & !q)? 0 : ((T_end & q) | (txen & !en_tx))? 1 : SY1 ;
bf_SY2 <= (st1 & !q)? 1 : (st2 & q)? 0 : SY2 ;
end

wire MIL_ce = cet & q ;
always @ (posedge clk) if (MIL_ce) begin
txen <= txen ;
en_tx <= txen? 1 : (!txen & T_end)? 0 : en_tx ;
CW_WD <= T_end? 0 : start? 1 : CW_WD ;
cb_bit <= (!en_tx | T_end)? 0 : en_tx? cb_bit+1 : cb_bit ;
T_dat <= st2? 1 : st18? 0 : T_dat ;
sr_dat <= st2? dat : T_dat? sr_dat<<1 : sr_dat ;
fd_cp <= st2? cp_tx : (T_dat & sr_dat[15])? !fd_cp : fd_cp ;
end
endmodule

```

5.2 Пример Verilog Test Fixture - текстового задания на моделирование модуля MIL_TXD

```

module test_MIL_TXD_v;
    // Inputs                // Outputs
    wire TXP;                wire TXN;
    reg clk1;                wire SY1;
    reg [15:0] dat;           wire SY2;
    reg txen;                wire en_tx;
                                wire T_dat;
                                wire T_end;
                                wire sdat;
                                wire fd_cp;

```

<pre>module SCH1_Lab406D // Входы</pre>	<pre> output wire TXP,</pre>
<pre> (input clk,</pre>	<pre> output wire TXN,</pre>
<pre> input[15:0]dat,</pre>	<pre> output wire tx_SY1,</pre>
<pre> input txen,</pre>	<pre> output wire tx_SY2,</pre>
<pre> input clk1,</pre>	<pre> output wire en_tx,</pre>
	<pre> output wire T_dat,</pre>
	<pre> output wire T_end,</pre>
	<pre> output wire tx_sdat,</pre>
	<pre> output wire [4:0]cb_tx_bit,</pre>

```

//
// Выходы приемника
output wire    dx_IN,
output wire    RX_clk,
output wire [4:0]cb_rx_bit,
output wire ce_rx_dat,
output wire    ok_SY,
output wire    en_rx,
output wire    ft_cp,
output wire    ok_rx,
output wire [15:0] rx_dat);

// Модуль передатчика
MIL_TXD DD1 (    .clk(clk),        .TXP(TXP),
                .dat(dat),        .TXN(TXN),
                .txen(txen),      .SY1(tx_SY1),
                                .SY2(tx_SY2),
                                .en_tx(en_tx),
                                .T_dat(T_dat),
                                .T_end(T_end),
                                .sdat(tx_sdat),
                                .cb_bit(cb_tx_bit)
                );

// Модуль приемника
MIL_RXD DD2 (    .clk(clk1),      .dx_IN(dx_IN),
                .RXP(TXP),        .stb_clk(RX_clk),
                .RXN(TXN),        .sr_dat(rx_dat),
                                .ok_SY(ok_SY),
                                .cb_bit(cb_rx_bit),
                                .en_rx(en_rx),
                                .ce_dat(ce_rx_dat),
                                .ft_cp(ft_cp),
                                .ok_rx(ok_rx));

endmodule

```

5.5 Пример текстового задания Verilog Test Fixture - на моделирование схемы SCH1__LAB406D

```

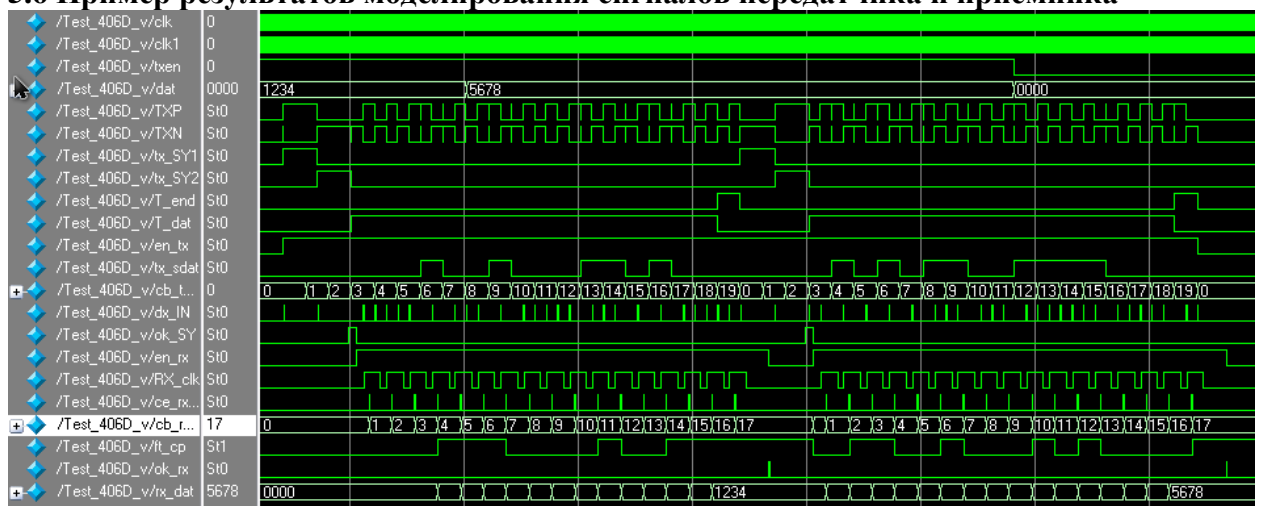
module test_LAB406D;
// Входы    схемы
    reg clk, clk1, txen;
    reg [15:0] dat;
// Выходы передатчика
    wire TXP, TXN, tx_SY1, tx_SY2, T_end, T_dat, en_tx, tx_sdat ;
    wire [4:0]cb_tx_bit;
// Выходы приемника
    wire en_rx, ce_rx_dat, ok_SY, ft_cp, ok_rx, RX_clk, dx_IN ;
    wire [4:0]cb_rx_bit;
    wire [15:0]rx_dat;
// Генератор сигнала синхронизации передатчика, PERIOD = 20 +/- 1;
// always begin clk1 = 1'b0; #10 clk1 = 1'b1; #9; end // PERIOD = 19
    always begin clk1 = 1'b0; #10 clk1 = 1'b1; #10; end // PERIOD = 20
// always begin clk1 = 1'b0; #10 clk1 = 1'b1; #11; end // PERIOD = 21

```

```
// Генератор сигнала синхронизации приемника
parameter PERIOD = 20;
always begin clk = 1'b0; #(PERIOD/2) clk = 1'b1; #(PERIOD/2); end
// Все выходы тестируемой схемы (UUT)
SCH1__LAB406D uut
    (//Входы схемы      Выходы передатчика
      .clk(clk),          .TXP(TXP),
      .txen(txen),        .TXN(TXN),
      .clk1(clk1),        .tx_SY1(tx_SY1),
      .dat(dat),          .tx_SY2(tx_SY2),
                        .en_tx(en_tx),
                        .T_dat(T_dat),
                        .T_end(T_end),
                        .tx_sdat(tx_sdat),
                        .cb_tx_bit(cb_tx_bit),
    //                  Выходы приемника
                        .dx_IN(dx_IN),
                        .ok_SY(ok_SY),
                        .en_rx(en_rx),
                        .RX_clk(RX_clk),
                        .ce_rx_dat(ce_rx_dat),
                        .ft_cp(ft_cp),
                        .cb_rx_bit(cb_rx_bit),
                        .rx_dat(rx_dat),
                        .ok_rx(ok_rx) );

// Временные параметры и данные входных сигналов
initial begin
    // Initialize Inputs
    dat = 0;          txen = 0; // Wait 100 ns for global reset to finish
    #100;             dat = 16'h1234; txen = 1; // CW
    #10000;           dat = 16'h5678; txen = 1; //WD
    // #100;          dat = 16'hEDCB; txen = 1;
    // #10000;        dat = 16'hA987; txen = 1;
    #24000;           dat = 16'h0000; txen = 0;
end
endmodule
```

5.6 Пример результатов моделирования сигналов передатчика и приемника



5.7 Выводы схемы SCH2_Lab406D для макета Nexys2

```

module SCH2_Lab406D      //Выходы Дисплея
    (input F50MHz,        output wire[7:0] seg, // Сегменты
     input [1:0]SWT,      output wire[3:0] an,  // Аноды
                                     output wire[7:0] ld, //Светодиоды

    //-----Выводы модулей MIL_TXD и MIL_RXD-----
        input JB1,                //MIL_RXP
        input JB2,                //MIL_RXN
                                     output wire JA1, //MIL_TXP
                                     output wire JA2, //MIL_TXN
                                     output wire JA3, //MIL_SY1
                                     output wire JA4, //MIL_SY2
                                     output wire JC1, // MIL_SDAT
                                     output wire JC2, // ok_SY
                                     output wire JC3, // ok_rx
                                     output wire JC4, //en_rx
                                     output wire JD1, //ce_rx_dat
                                     output wire JD2, //RL
                                     output wire JD3, //RH
                                     output wire JD4 ); //MIL_en_tx

assign JA1 = TXP                ; assign JA2 = TXN;          //Выходы передатчика
assign JA3 = SY1                ; assign JA4 = SY2 ;         //Выходы передатчика
assign JC1 = MIL_SDAT           ; assign JC2 = ok_SY ;
assign JC3 = ok_rx              ; assign JC4 = MIL_en_tx;    // Выходы дешифратора
assign JD1= ce_rx_dat           ; assign JD2 = RL ;          // Выходы дешифратора
assign JD3 = RH                 ; assign JD4 = en_rx;        // Выходы дешифратора

```

5.8 Выводы ПЛИС XC3500E-4fg320 макета Nexys2

```

NET "JA1" LOC = "L15" ;#Pin1
NET "JA2" LOC = "K12" ;#Pin2
NET "JA3" LOC = "L17" ;#Pin3
NET "JA4" LOC = "M15" ;#Pin4
NET "JB1" LOC = "M13" ;#Pin1
NET "JB2" LOC = "R18" ;#Pin2
NET "JC1" LOC = "G15" ;#Pin1
NET "JC2" LOC = "J16" ;#Pin2
NET "JC3" LOC = "G13" ;#Pin3
NET "JC4" LOC = "H16" ;#Pin4
NET "JD1" LOC = "J13" ;#Pin1
NET "JD2" LOC = "M18" ;#Pin2
NET "JD3" LOC = "N18" ;#Pin3
NET "JD4" LOC = "P18" ;#Pin4
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 "F50MHz" LOC = "B8" ;#F50MHz
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<7>" LOC = "C17" ;#CP
NET "SWT<0>" LOC = "G18" ;#SWT0
NET "SWT<1>" LOC = "H18" ;#SWT1

```

5.9 Пример осциллограмм сигналов sdat и TXP модуля MIL_TXP

