

Лабораторная работа №402N

1. Генераторы импульсов на ПЛИС

1.1 Ждущий генератор импульса с длительностью $MT \cdot T_{ce}$ ($T_{ce} = NT_{clk} \cdot T_{clk}$)

// V1.1 Схема модуля ждущего генератора импульса с длительностью $MT \cdot T_{ce}$ на Verilog-e

```

`define m 11
`define n 16
module WGMT (    input st,                output reg [`m-1:0] q=0,
                 input clk,                output wire end_PW,
                 input [`n-1:0] NTclk,    output reg ceo=0,
                 input [`m-1:0] MT,       output reg PW=0);

reg [`n-1:0] cb_ce=0 ;
assign end_PW = (q==1); //Выход компаратора
wire ce = (cb_ce==NTclk);
always @ (posedge clk) begin
  cb_ce <= ((st & !PW) | ce)? 1 : cb_ce + 1 ;
  q <= (st & !PW)? MT : (ce & PW)? q-1 : q ;
  PW <= st? 1 : (ce & end_PW)? 0 : PW ;
  ceo <= ce ;
end
endmodule
posedge_clk

```

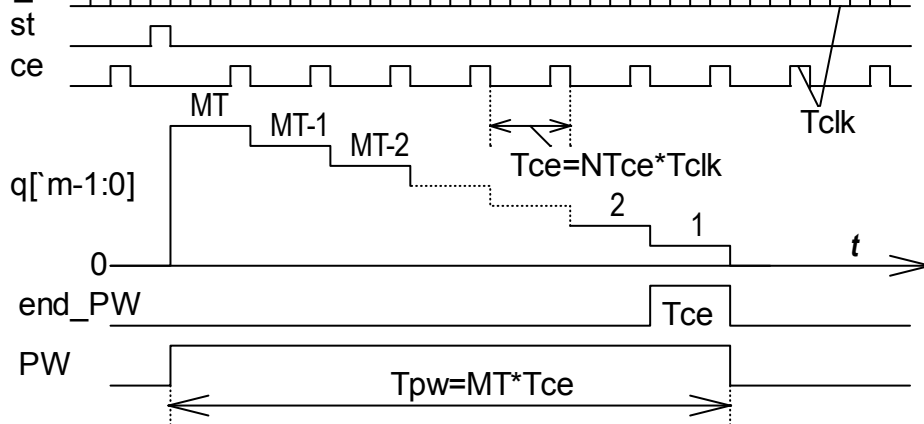


Рис.1 Временные диаграммы ждущего генератора импульса

Импульс запуска st должен иметь длительность $T_{clk}=20$ нс. Выходной сигнал ce с периодом $NT_{clk} \cdot T_{clk}$ предназначен для использования в других модулях.

1.2 Генераторы периодической последовательности импульсов (синтезаторы периода) с длительностью T_{ce} и периодом $N \cdot T_{ce}$ (T_{ce} - период сигнала ce).

// V.1.2.1 Схема модуля синтезатора периода с суммирующим счетчиком, $T_{tc} = N \cdot T_{ce}$

```

`define k 11 //Число разрядов
module AGNT (    input [`k-1:0] N,        output reg [`k-1:0] q=0,
                 input ce,                output wire TC,
                 input clk,                output wire ceo);

assign TC = (q==N) ; //Выход компаратора
assign ceo = ce & TC ;
always @ (posedge clk) if (ce) begin
  q <= TC? 1 : q+1 ; // Загрузка 1 и инкремент счетчика
end
endmodule

```

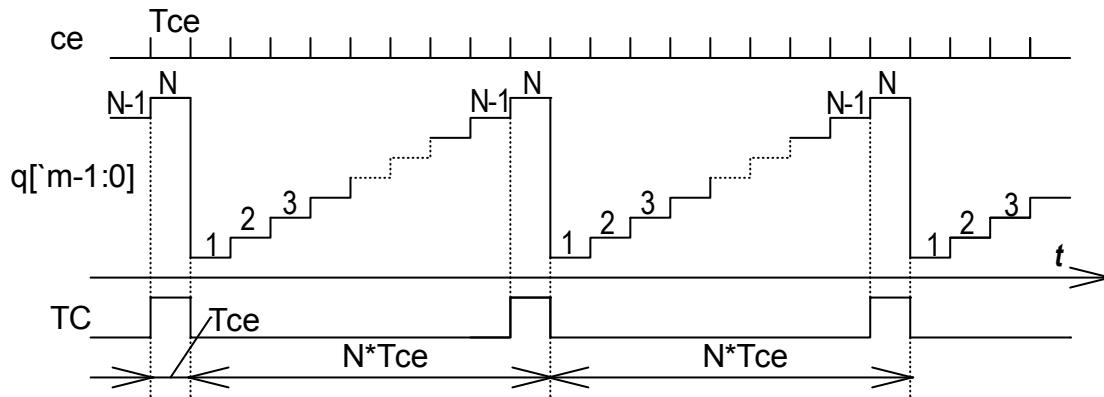


Рис.2 Временные диаграммы синтезатора периода с суммирующим счетчиком

// V.1.2.2 Схема модуля синтезатора периода с вычитающим счетчиком, $T_{tc}=N \cdot T_{ce}$

```

`define k 11 //Число разрядов
module AGNTD( input [`k-1:0] N, output wire TC,
              input ce, output reg [`k-1:0] q=1,
              input clk);
assign TC = (q==1); //Выход компаратора
always @ (posedge clk) if (ce) begin
q <= TC? N : q-1; //Загрузка N и декремент счетчика
end
endmodule

```

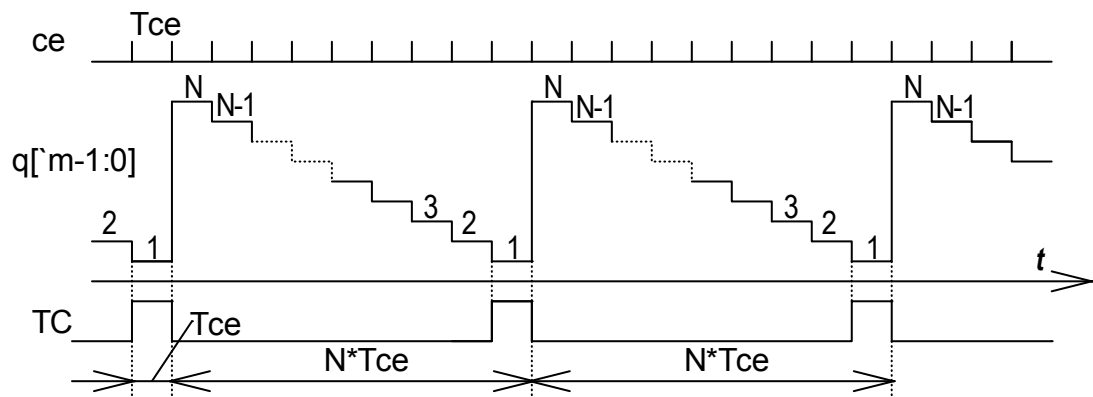


Рис.3 Временные диаграммы синтезатора периода с вычитающим счетчиком

1.3 Временные диаграммы синтезатора периода с вычитающим счетчиком

// V.1.3 Схема модуля генератора периодической последовательности импульсов с периодом $N \cdot T_{ce}$ и длительностью $MT \cdot T_{ce}$

```

`define s 11 //Число разрядов
module AGNMT ( input [`s-1:0] N, output reg PW = 0,
               input [`s-1:0] MT, output wire start_PW,
               input ce, output wire end_PW,
               input clk, output reg [`s-1:0] q=0);
assign start_PW = (q==N);
assign end_PW = (q==MT);
always @ (posedge clk) if (ce) begin
PW <= end_PW? 0: start_PW? 1 : PW;
q <= start_PW? 1 : q+1;
end
endmodule

```

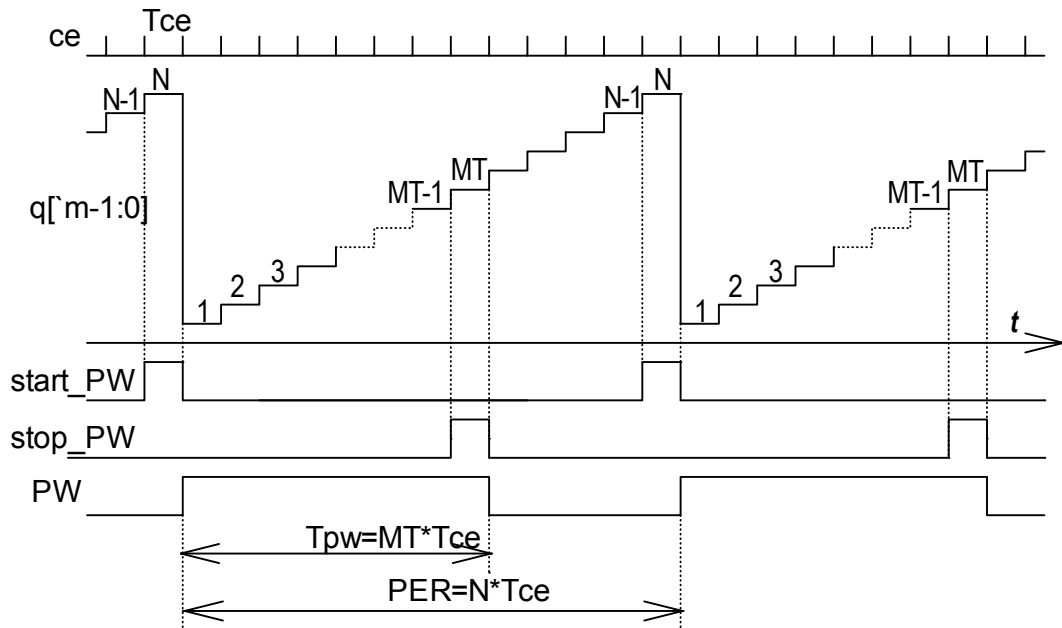


Рис.4 Временные диаграммы модуля генератора периодической последовательности импульсов с периодом $N \cdot T_{ce}$ и длительностью $MT \cdot T_{ce}$

1.4 Генератор "пилы" от Y до X

// V.1.4 Схема модуля генератора "пилы" от Y до X

```

`define m 4 //Число разрядов
module AGSAW_YX(    input ce,
                    input [m-1:0] Y,    output wire tc_Y,
                    input [m-1:0] X,    output wire tc_X,
                    input clk,          output reg [m-1:0] q=0);

assign tc_X = (q==X) ; //сигнал при q==X
assign tc_Y = (q==Y) ; //сигнал при q==Y
reg UP = 1;           //Триггер направления счета
always @(posedge clk) if (ce) begin
q <= UP? q+1 : q-1 ;    //max q = X, min q = Y
UP <= (q==Y+1)? 1 : (q==X-1)? 0 : UP ;
end
endmodule

```

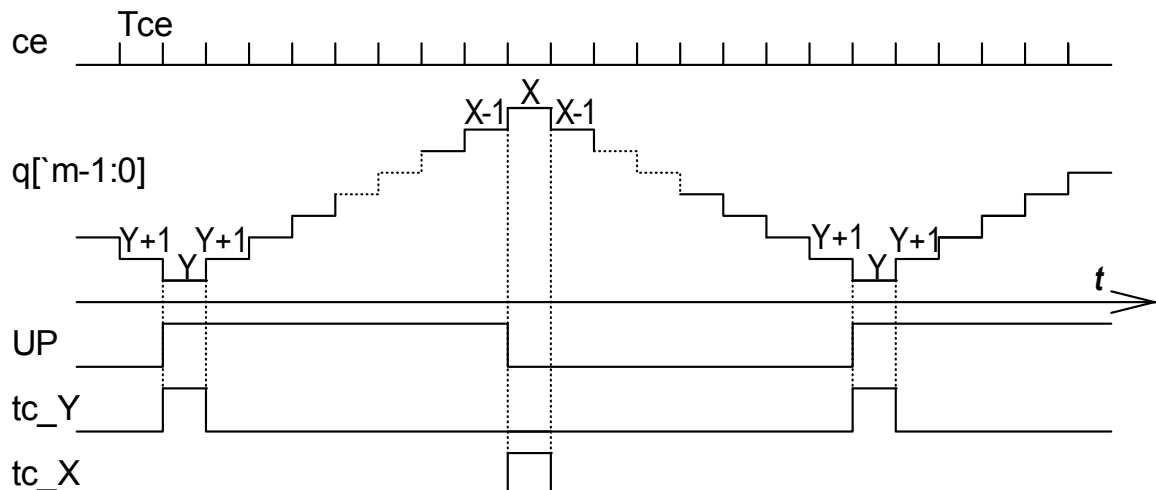


Рис.5 Временные диаграммы генератора "пилы" от Y до X

На макете Nexys2, на котором выполняется работа имеется упрощенный порт VGA, к выводам 1,2,3 которого подключены выходы резисторных цифро-аналоговых преобразователей (ЦАП). Эти ЦАП имеют по 3 разряда для красного и зеленого цветов и всего 2 разряда для синего. Для получения 4-х разрядного ЦАП выводы 1 и 2 разъема VGA (на некоторых экземплярах макетов) дополнительно соединены резистором с сопротивлением 2 кОм. Поэтому вывод 1 VGA разъема является выходом 4-х разрядного ЦАП, в котором RED0 – старший разряд, а GRN0 – младший разряд. Выводы ПЛИС P8 и P6 должны быть свободными. Напряжение на выходе этого ЦАП $U_{dac} = VCC \cdot X / 15$, где $VCC = 3.3V$, а $X = \sum_{i=0}^3 x_i \cdot 2^i$ ($x_0 = GRN0, x_1 = RED0, x_2 = RED1, x_3 = RED2$).

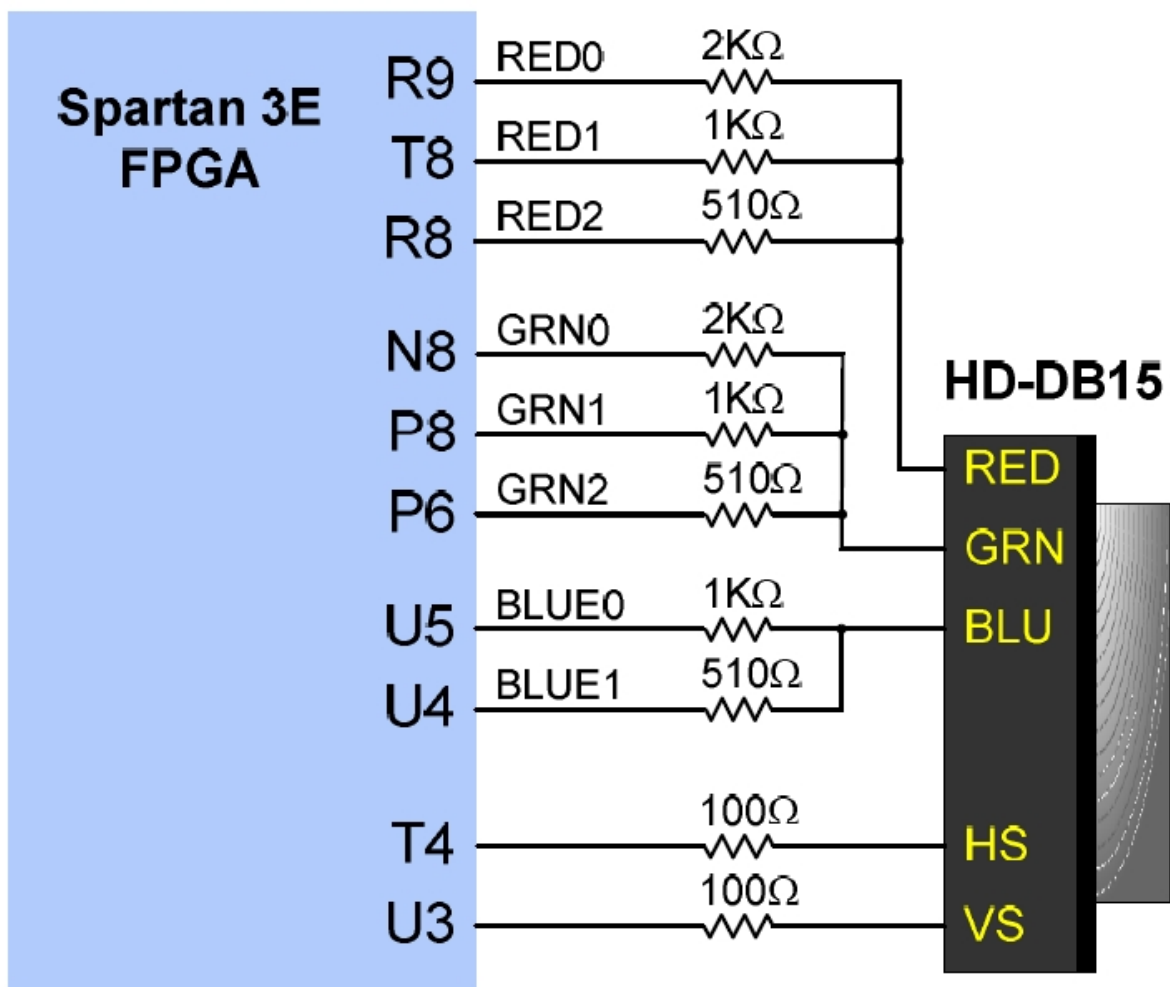
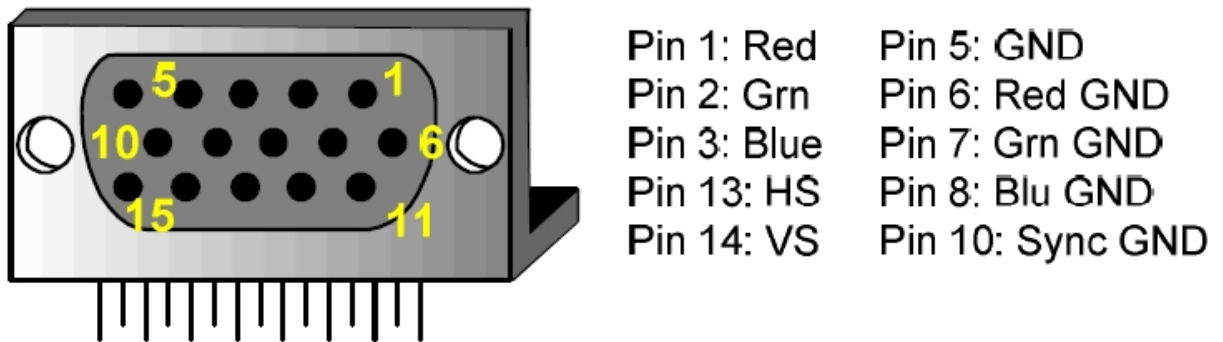


Рис.6 Порт VGA макета Nexys2 (ЦАП для генератора «пилы»)

1.5 Аккумулятор – накапливающий сумматор

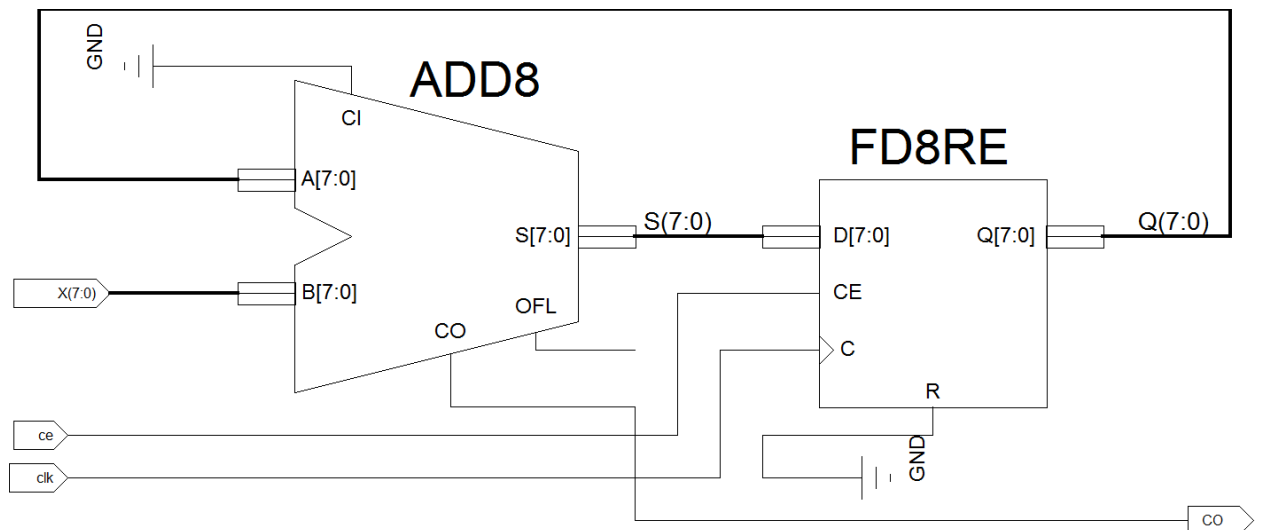


Рис.7 Схема 8-ми разрядного аккумулятора из библиотечных модулей: сумматора ADD8 и D-триггера FD8RE

В сумматоре ADD8 сумма $S[7:0] = A[7:0] + B[7:0] + CI$ по модулю $M = 2^m$. Сигнал переноса $CO = 1$, если $S[7:0] \geq 2^{m-1}$ ($m=8$). Сигнал арифметического переполнения OFL (чисел в дополнительном коде) в этом аккумуляторе не используется.

//V.1.5.1 Схема модуля аккумулятора с емкостью 2^m

```

`define m 4
module ACC2mE(  input [`m-1:0] X,      output reg [`m-1:0] ACC = 0,
                input ce,              output wire CO,
                input clk);
assign CO = (X+ACC >= 1<<m); //Сигнал переноса
always @ (posedge clk) if (ce) begin
ACC <= ACC + X; // Аккумулятор с емкостью  $2^m$ 
end
endmodule

```

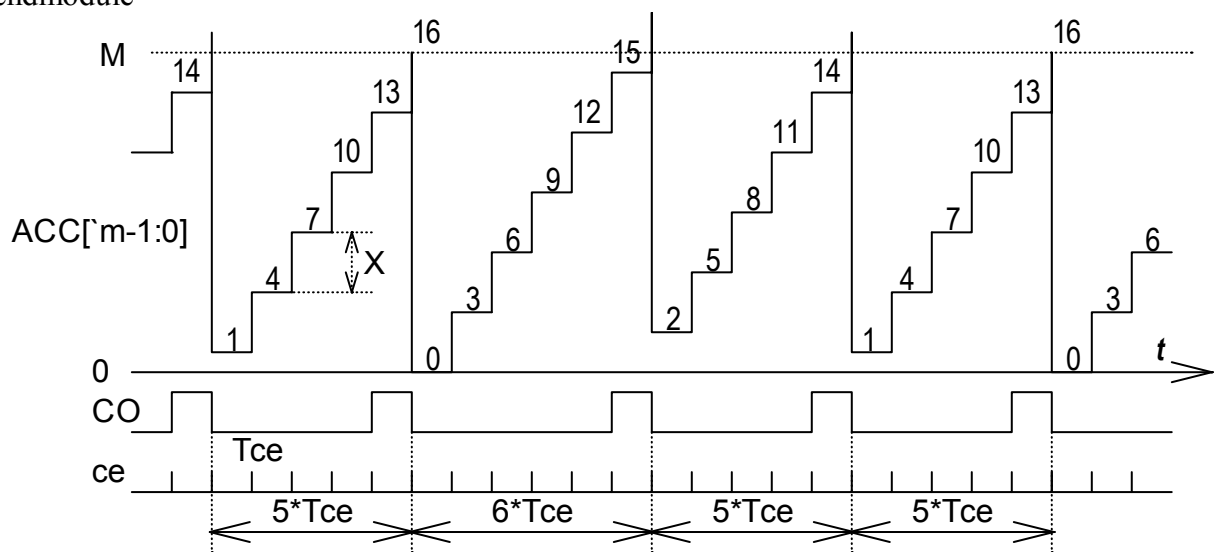


Рис.8 Пример временных диаграмм аккумулятора при $M=2^4=16$ и $X=3$

Из рис.8 можно увидеть, что период сигнала CO равен $M \cdot T_{ce}$.

Средняя частота F_{co} сигналов переноса СО аккумулятора $F_{co} = X \frac{F_{ce}}{2^m}$ пропорциональна числу X , частоте F_{ce} и обратно пропорциональна емкости аккумулятора $M=2^m$.

Если число X не кратно емкости аккумулятора M , то интервалы между соседними импульсами переноса СО не одинаковые, но принимают только 2 значения:

$$T_{ce} \cdot \left\lceil \frac{M}{X} \right\rceil \text{ или } T_{ce} \cdot \left(\left\lceil \frac{M}{X} \right\rceil + 1 \right), \text{ где } \left\lceil \frac{M}{X} \right\rceil - \text{целая часть отношения } \frac{M}{X}.$$

Относительная частота больших интервалов равна дробной части $\left\{ \frac{M}{X} \right\}$ отношения $\frac{M}{X}$.

V.1.5.2 Схема аккумулятора с емкостью $M < 2^m$ и генератора меандра.

```
`define a 11 //Число разрядов
module ACCM (    input [7:0] X, output reg [a-1:0] ACC = 0,
                input ce,      output wire CO,
                input clk,      output reg Mx=0);
parameter M=500 ;
assign CO = (X+ACC >= M); //Сигнал переноса СО=1 при X+ACC >= M
always @ (posedge clk) if (ce) begin
ACC <= CO? ACC + X - M : ACC + X; // Аккумулятор с емкостью M
Mx <= (CO & ce)? !Mx : Mx ;//Меандр
end
endmodule
```

$$\text{Частота сигнала Mx меандра } F_{Mx} = \frac{F_{co}}{2} = X \frac{F_{ce}}{2 \cdot M} = X \frac{F_{clk}}{2 \cdot M \cdot NTclk}.$$

Если заданы: дискрет изменения частоты меандра $\Delta F_{Mx} = F_{Mx}(X=1)$, F_{clk} и $NTclk$, то емкость аккумулятора $M = \frac{F_{clk}}{2 \cdot \Delta F_{Mx} \cdot NTclk}$. Например, $F_{clk} = 500000000$ Гц, $\Delta F_{Mx} = 1$ Гц,

$$NTclk = 50000: M = \frac{500000000}{2 \cdot 1 \cdot 50000} = 500.$$

2. Задание к допуску

2.1.1 Начертить в тетради временные диаграммы модуля ждущего генератора **WGMT** (рис.1).

2.1.2 Переписать в тетрадь схему (V.1.1) модуля **WGMT**. Для заданного варианта параметра $NTclk$ (см. таб.1).

2.2.1 Начертить в тетради временные диаграммы модуля **AGNT** или **AGNT(D)** синтезатора периода (рис.2 или рис.3).

2.2.2 Переписать в тетрадь схему выбранного модуля синтезатора периода (V.1.2.1 или V.1.2.2).

2.3.1 Начертить в тетради временные диаграммы модуля **AGNMT** генератора периодической последовательности импульсов с периодом $N \cdot T_{ce}$ и длительностью $MT \cdot T_{ce}$ ($N > MT$, рис.4).

2.3.2 Переписать в тетрадь схему модуля **AGNMT** (V.1.3).

2.4.1 Начертить в тетради временные диаграммы генератора “пилы” (рис.5).

2.4.2 Переписать в тетрадь схему модуля (V.1.4) **AGSAW** генератора “пилы”.

2.5.1 Начертить в тетради пример временных диаграмм аккумулятора (рис.8).

2.5.2 Переписать в тетрадь схему модуля (V.1.5.2) аккумулятора с емкостью M ($M < 2^m$).

Таблица 1

| № | NTclk (Период $T_{ce}=NT_{clk} \cdot T_{clk}$) | N (Период $T_{per}=N \cdot T_{ce}$) | MT (Длительность $T_{pw}=MT \cdot T_{ce}$) | Y | X | ΔF_{Mx} [Hz] | Вариант измерителя |
|----|--|---|--|---|----|-------------------------|-----------------------|
| 1 | 25 | 50 | 10 | 1 | 7 | 2000 | FMx (ACCM) |
| 2 | 50 | 100 | 12 | 2 | 8 | 1000 | PW/Tce (WGMT) |
| 3 | 500 | 40 | 5 | 3 | 9 | 100 | PER/Tce (AGNT(D)) |
| 4 | 5000 | 60 | 15 | 4 | 10 | 10 | PW/Tce (AGNMT) |
| 5 | 50000 | 75 | 14 | 3 | 11 | 1 | PAUS/Tce (AGNMT) |
| 6 | 5000 | 45 | 12 | 2 | 12 | 10 | FMx (ACCM) |
| 7 | 500 | 180 | 13 | 1 | 13 | 100 | PER/Tce (AGNMT) |
| 8 | 50 | 150 | 12 | 2 | 12 | 1000 | Fco (ACCM) |
| 9 | 25 | 120 | 11 | 3 | 11 | 2000 | FMx (ACCM) |
| 10 | 50 | 125 | 15 | 4 | 10 | 1000 | PW/Tce (WGMT) |
| 11 | 500 | 140 | 10 | 5 | 9 | 100 | PER/Tce (AGNT(D)) |
| 12 | 5000 | 150 | 13 | 4 | 8 | 10 | PW/Tce (AGNMT) |
| 13 | 50000 | 100 | 14 | 3 | 7 | 1 | PAUS/Tce (AGNMT) |
| 14 | 5000 | 200 | 10 | 2 | 8 | 10 | FMx (ACCM) |
| 15 | 500 | 250 | 11 | 1 | 6 | 100 | PER/Tce (AGNMT) |
| 16 | 50 | 140 | 14 | 0 | 7 | 1000 | Fco (ACCM) |
| 17 | 25 | 150 | 15 | 1 | 6 | 2000 | PAUS/Tce (AGNMT) |
| 18 | 50 | 120 | 12 | 2 | 9 | 1000 | PER/Tce (AGSAW) |
| 19 | 500 | 250 | 15 | 3 | 10 | 100 | PER/Tce (AGNMT) |
| 20 | 5000 | 100 | 20 | 4 | 11 | 10 | FMx (ACCM) |

3.Задание к выполнению работы

3.1 Создать модуль (V.1.1) ждущего генератора импульса **WGMNT**. Выполнить синтез, провести моделирование работы генератора. Зарисовать эскизы временных диаграмм.

3.2 Создать модуль **AGNT** или **AGNTD** (V.1.2.1 или V.1.2.2) выбранного варианта синтезатора периода. Выполнить синтез, провести моделирование работы генератора. Зарисовать эскизы временных диаграмм.

3.3 Создать модуль **AGNMT** (V.1.3) генератора периодической последовательности импульсов с длительностью $MT \cdot T_{ce}$ и периодом $N \cdot T_{ce}$. Выполнить синтез, провести моделирование работы генератора. Зарисовать эскизы временных диаграмм.

3.4 Создать модуль **AGSAW** (V.1.4) генератора “пилы от Y до X”. Выполнить синтез, провести моделирование работы генератора «пилы». Зарисовать эскизы временных диаграмм.

3.5 Создать модуль (V.1.5.1) **ACC2mE** аккумулятора с емкостью 2^m . Выполнить синтез, провести моделирование работы генератора. Зарисовать эскизы временных диаграмм.

3.6 Для заданных параметров (см.табл.1) создать символы всех отлаженных модулей.

3.6.1 Создать модуль **ACCM** (V.1.5.2). Для заданных ΔF_{Mx} и $NTclk$ определить M. Создать символ этого модуля.

3.7 Для заданных параметров создать модуль и символ **DAT_BL**.

3.8 Создать модуль и символ **BUTTON_BL** (см. приложение 5.1).

3.9 Создать модуль и символ **Display** (см. приложение 5.2).

3.10 Из созданных символов составить схему Sch_Lab402 (рис.9). Без модуля **MEG_BL** на вход HB[7:0] старшего байта семи сегментного индикатора Display можно подать 8 бит шины N[7:0] или MT[7:0]. Выполнить синтез схемы Sch_Lab402. Создать для неё файл Sch_Lab402.ucf (Implementation Constraints File, см. приложение 5.3) и файл конфигурации (Generate Target PROM/ACE File), загрузить в ПЛИС или ПЗУ макета.

3.11 Проверить работу всех генераторов импульсов. Получить и сохранить осциллограммы выходных сигналов:

- PW и end_PW - ждущего генератора **WGMT**,
- TC – синтезатора периода **AGNT(D)**,
- PW – генератора **AGNMT**,
- Udac – генератора «пилы» **AGSAW**,
- CO и Mx – синтезатора частоты **ACCM**.

4.Задание к сдаче работы

4.1. Составить схему модуля **MEG_BL** измерителя параметров импульсов заданного варианта генератора (см.табл.1). Модуль **MEG_BL** должен иметь 4 порта входов: clk, st, Inp, REF и один 8-и битный выходной порт Q[7:0].

Выход Q[7:0] результатов измерения модуля **MEG_BL** предназначен для соединения с входом старшего байта HB[7:0] модуля **Display**.

Запускаться измеритель должен по входу **st** сигналом с выхода модуля **BUTTON_BL**.

На вход Inp должен подаваться измеряемый сигнал, а на вход REF эталон времени:

- **се** – при измерениях длительности или периода,
- **PW** – при измерении частоты.

4.2 Создать модуль **MEG_BL** провести моделирование его работы. Зарисовать эскиз полученных временных диаграмм.

4.3 Создать символ **MEG_BL**, вставить его в лист схемы Sch_Lab402. Создать файл конфигурации (Generate Target PROM/ACE File) загрузить в ПЛИС или ПЗУ макета. Продемонстрировать работу измерителя.

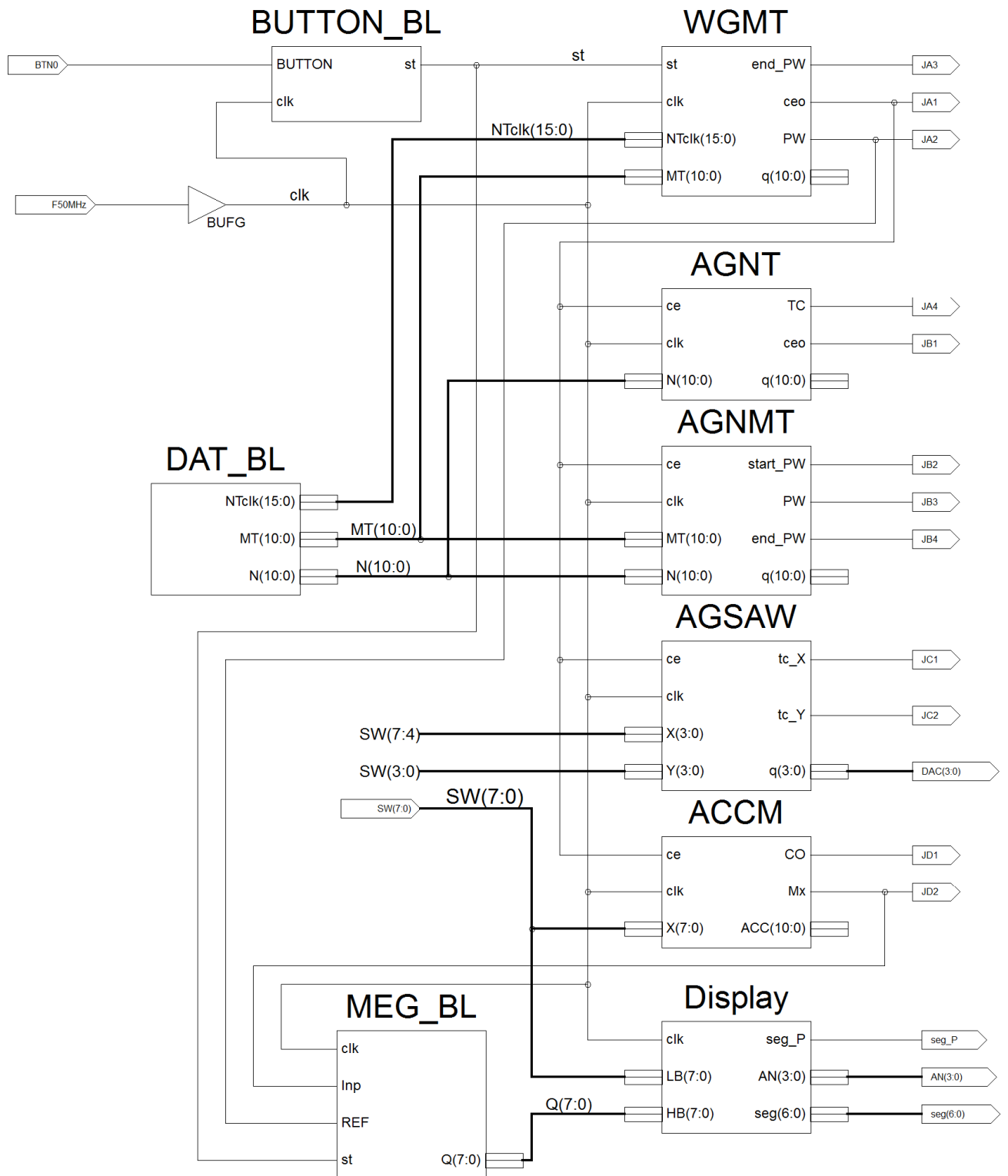


Рис. 9 Пример схемы Sch_Lab402 лабораторной работы №402N

В этом примере модуль **MEG_BL** предназначен для измерения частоты сигнала меандра модуля **ACCM**.

5. Приложения

5.1 Модуль кнопки **BUTTON_BL**

```
module BUTTON_BL(      input BUTTON,      output st,
                    input clk);
```

```

reg [15:0]cb_ce=0 ;
wire ce= (cb_ce==50000);
reg q1=0, q2=0 ;
assign st= q1 & !q2 & ce ;
always @ (posedge clk) begin
q1 <= ce? BUTTON : q1 ;   q2 <= ce? q1 : q2 ;
cb_ce <= ce? 1 : cb_ce+1 ;
end
endmodule

```

5.2 Семи сегментный индикатор (модуль **Display**)

```

module Display(      input clk,                output wire[3:0] AN, //Аноды
                    input[7:0]LB,             output wire[6:0] seg, //Сегменты
                    input[7:0]HB,             output wire seg_P); //Точка
parameter Fclk=50000 ;      //50000 kHz
parameter F1kHz=1 ;        //1 kHz
wire [1:0]ptr_P=2'b10 ;    //Точка в центре
reg [15:0] cb_1ms =0 ;
wire ce = (cb_1ms==Fclk/F1kHz) ;
always @ (posedge clk) begin
cb_1ms <= ce? 1 : cb_1ms+1 ;
end
reg [1:0]cb_an=0 ;//Счетчик анодов
//-----
always @ (posedge clk) if (ce) begin
cb_an <= cb_an+1 ;
end
//-----Переключатель анодов-----
assign AN = (cb_an==0)? 4'b1110 ://включение цифры 0 (младшей)
            (cb_an==1)? 4'b1101 ://включение цифры 1
            (cb_an==2)? 4'b1011 ://включение цифры 2
            4'b0111 ;//включение цифры 3 (старшей)
//-----Переключатель тетрад (HEX цифр)-----
wire[3:0] dig=(cb_an==0)? LB[3:0]:
            (cb_an==1)? LB[7:4]:
            (cb_an==2)? HB[3:0]: HB[7:4];
//-----Семи сегментный 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_an) ;
endmodule

```

5.3 Распределение портов схемы по контактным площадкам ПЛИС (*.ucf файл) (не используемые в данной работе выводы ПЛИС закомментированы символом #)

```

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

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

#NET "led0" LOC = "J14" ; #LD0
#NET "led1" LOC = "J15" ; #LD1
#NET "led2" LOC = "K15" ; #LD2
#NET "led3" LOC = "K14" ; #LD3
#NET "led4" LOC = "E17" ; #LD4
#NET "led5" LOC = "P15" ; #LD5
#NET "led6" LOC = "F4" ; #LD6
#NET "led7" LOC = "R4" ; #LD7

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" ; #SWT0
NET "SW<1>" LOC = "H18" ; #SWT1
NET "SW<2>" LOC = "K18" ; #SWT2
NET "SW<3>" LOC = "K17" ; #SWT3
NET "SW<4>" LOC = "L14" ; #SWT4
NET "SW<5>" LOC = "L13" ; #SWT5
NET "SW<6>" LOC = "N17" ; #SWT6
NET "SW<7>" LOC = "R17" ; #SWT7

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

NET "JA1" LOC = "L15" ; #Pin1

```

```

NET "JA2" LOC = "K12" ;#Pin2
NET "JA3" LOC = "L17" ;#Pin3
NET "JA4" LOC = "M15" ;#Pin4
#NET "JA7" LOC = "K13" ;#Pin7
#NET "JA8" LOC = "L16" ;#Pin8
#NET "JA9" LOC = "M14" ;#Pin9
#NET "JA10" LOC = "M16" ;#Pin10

```

```

NET "JB1" LOC = "M13" ;#| PULLUP
NET "JB2" LOC = "R18" ;#| PULLUP
#NET "JB3" LOC = "R15" ;#
NET "JB4" LOC = "T17" ;#
#NET "JB7" LOC = "P17" ;#Pin7
#NET "JB8" LOC = "R16" ;#Pin8
#NET "JB9" LOC = "T18" ;#Pin9
#NET "JB10" LOC = "U18" ;#Pin10

```

```

NET "JC1" LOC = "G15" ;#Pin1
NET "JC2" LOC = "J16" ;#Pin2
#NET "JC3" LOC = "G13" ;#Pin3
#NET "JC4" LOC = "H16" ;#Pin4
#NET "JC7" LOC = "H15" ;#Pin7
#NET "JC8" LOC = "F14" ;#Pin8
#NET "JC9" LOC = "G16" ;#Pin9
#NET "JC10" LOC = "J12" ;#Pin10

```

```

NET "JD1" LOC = "J13" ;#Pin1
NET "JD2" LOC = "M18" ;#Pin2
#NET "JD3" LOC = "N18" ;#Pin3
#NET "JD4" LOC = "P18" ;#Pin4
#NET "JD7" LOC = "K14" ;#LD3
#NET "JD8" LOC = "K15" ;#LD3
#NET "JD9" LOC = "J15" ;#LD3
#NET "JD10" LOC = "J14" ;#LD3
# Цифро-аналоговый преобразователь (порт VGA)
NET "DAC<0>" LOC = "N8" ;#GRN0
NET "DAC<1>" LOC = "R9" ;#RED0
NET "DAC<2>" LOC = "T8" ;#RED1
NET "DAC<3>" LOC = "R8" ;#RED2

```