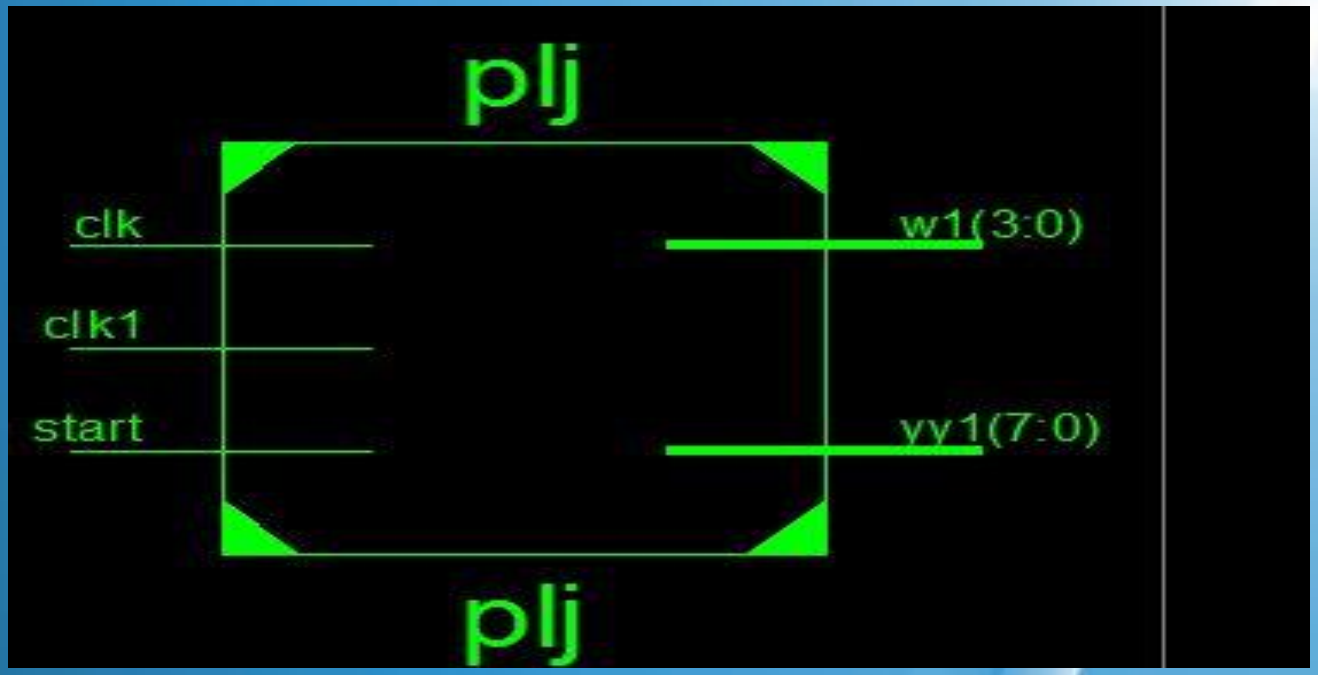
数字频率计设计与仿真

组员：周若华
刘金凯

数字频率计



用一个稳定的频率源作为基准时钟，对比测量其他信号的频率。通常情况下计算每秒内待测信号的脉冲个数，也就是称闸门时间为1秒。闸门时间也可以大于或小于一秒。闸门时间越长，得到的频率值就越准确。

数字频率计

因为在电脑上仿真，所以这里用基准时钟clk 来测量被测时钟clk1。clk clk1 在仿真波形文件里设定

步骤

1、先产生一个持续时间为一秒的的闸门信号

2、获取7位十进制计数器的计数脉冲

3、对被测信号计脉冲数

4、把7位十进制计数器有效的高4位数据送入码寄存器bcd0~3，然后动态显示

5、译码

频率计源代码

```
 library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity plj is
  port ( start:in std_logic;                    --复位信号
      clk :in std_logic;                    --系统时钟
      clk1:in std_logic;                    --被测信号
      yy1:out std_logic_vector(7 downto 0);     --八段码
      w1 :out std_logic_vector(3 downto 0));    --数码管位选信号
end plj;
architecture behav of PLj is
signal b1,b2,b3,b4,b5,b6,b7:std_logic_vector(3 downto 0);  --十进制计数器
signal bcd:std_logic_vector(3 downto 0);              --BCD码寄存器
signal q :integer range 0 to 49999999;               --秒分频系数
signal qq : integer range 0 to 499999;               --动态扫描分频系数
signal en,bclk:std_logic;                         --使能信号，有效被测信号
signal sss : std_logic_vector(3 downto 0);            --小数点
signal bcd0,bcd1,bcd2,bcd3 : std_logic_vector(3 downto 0);寄存7位十位计数器中
有效的高4位数据
```

```
 begin
second:process(clk)                -----产生一个持续时间为一秒的的闸门信号
begin
  if start='1' then q<=0;
  elsif clk'event and clk='1' then
     if q<49999999 then q<=q+1;
     else q<=49999999;
     end if;
  end if;
  if q<49999999 and  start='0' then en<='1';
  else en<='0';
  end if;
end process;

and2:process(en,clk1)              -----获取 7位十进制计数器的计数脉冲
begin
  bclk<=clk1 and en;
end process;
```

```
com:process(start,bclk)                                    ------完成对被测信号计脉冲数
begin
  if start='1' then
b1<="0000";b2<="0000";b3<="0000";b4<="0000";b5<="0000";b6<="0000";b7<="0000";
  elsif bclk'event and bclk='1' then
    if b1="1001" then b1<="0000";
      if b2="1001" then b2<="0000";
        if b3="1001" then b3<="0000";
          if b4="1001" then b4<="0000";
            if b5="1001" THEN b5<="0000";
              if b6="1001" then b6<="0000";
                if b7="1001" then b7<="0000";
                else b7<=b7+1;
                end if;
              else b6<=b6+1;
              end if;
            else b5<=b5+1;
            end if;
          else b4<=b4+1;
          end if;
        else b3<=b3+1;
        end if;
      else b2<=b2+1;
      end if;
    else b1<=b1+1;
    end if;
  end if;
end process;
```

```
 process(clk)                          -----把7位十进制计数器有效的高4位数据送入bcd0~3
begin
 if rising_edge(clk) then
      if b7>"0000" then bcd3<=b7; bcd2<=b6; bcd1<=b5; bcd0<=b4; sss<="1110";
      elsif b6>"0000" then bcd3<=b6; bcd2<=b5; bcd1<=b4; bcd0<=b3; sss<="1101";
      elsif b5>"0000" then bcd3<=b5; bcd2<=b4; bcd1<=b3; bcd0<=b2; sss<="1011";
                else bcd3<=b4; bcd2<=b3; bcd1<=b2; bcd0<=b1; sss<="1111";
                end if;
  end if;
end process;
```

```
weixuan:process(clk)                          -------动态显示
begin
 if clk'event and clk='1' then
      if qq< 9then qq<=qq+1;bcd<=bcd3; w1<="0111";
                  if sss="0111" then yy1(0)<='0';
                              else yy1(0)<='1';
                              end if;
      elsif qq<19 then qq<=qq+1;bcd<=bcd2; w1<="1011";
                  if sss="1011" then yy1(0)<='0';
                              else yy1(0)<='1';
                              end if;
      elsif qq<29 then qq<=qq+1;bcd<=bcd1; w1<="1101";
                  if sss="1101" then yy1(0)<='0';
                  else yy1(0)<='1';
                  end if;
      elsif qq<39 then qq<=qq+1;bcd<=bcd0; w1<="1110";
        if sss="1110" then yy1(0)<='0';
        else yy1(0)<='1';
        end if;
      else  qq<=0;
      end if;
 end if;
end process;
```

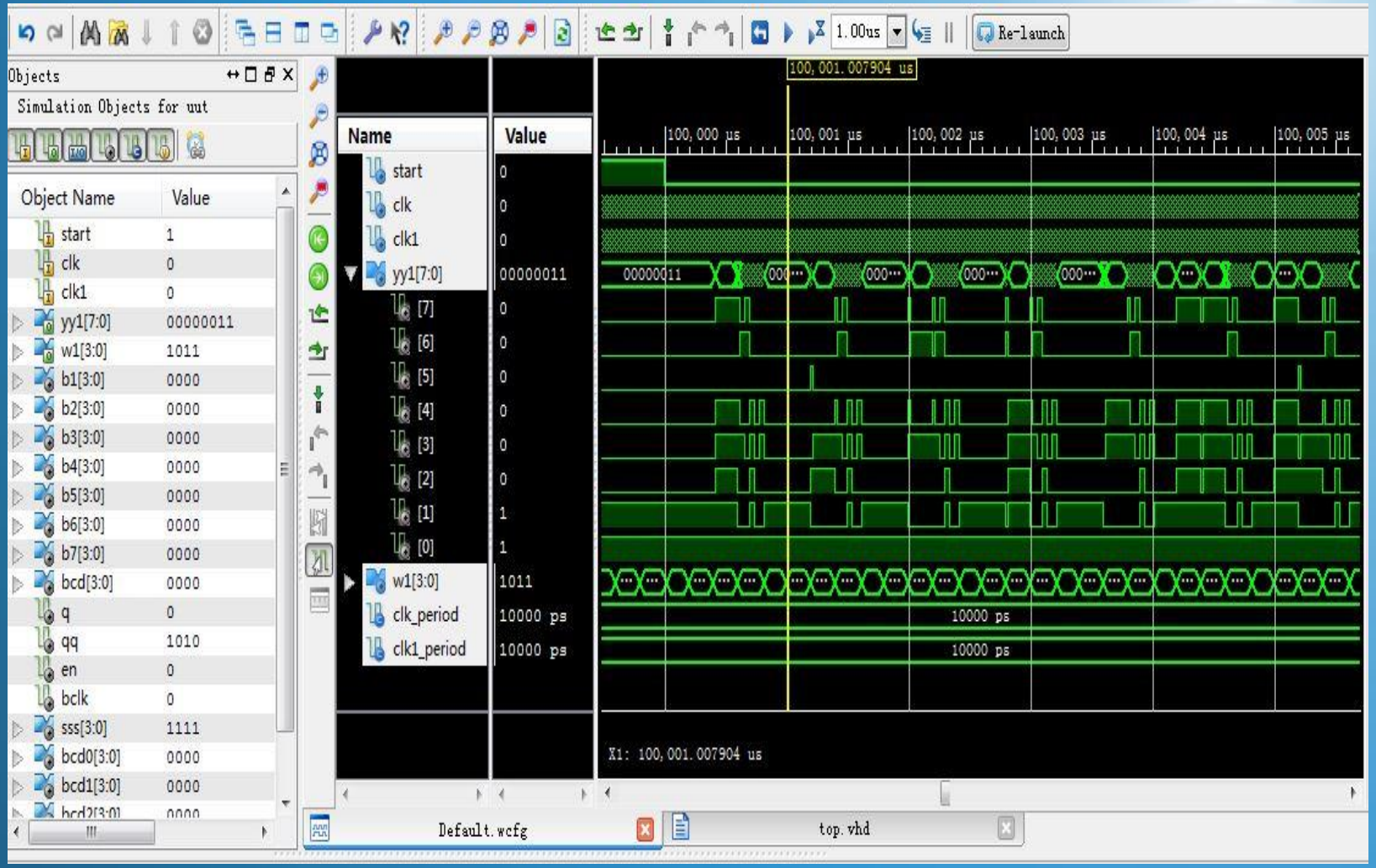
```
m0: process (bcd)            --译码
 begin
 case bcd is
    when "0000"=>yy1(7 downto 1)<="0000001";
    when "0001"=>yy1(7 downto 1)<="1001111";
    when "0010"=>yy1(7 downto 1)<="0010010";
    when "0011"=>yy1(7 downto 1)<="0000110";
    when "0100"=>yy1(7 downto 1)<="1001100";
    when "0101"=>yy1(7 downto 1)<="0100100";
    when "0110"=>yy1(7 downto 1)<="1100000";
    when "0111"=>yy1(7 downto 1)<="0001111";
    when "1000"=>yy1(7 downto 1)<="0000000";
    when "1001"=>yy1(7 downto 1)<="0001100";
    when others=>yy1(7 downto 1)<="1111111";
 end case;
end process;
end behav;
```

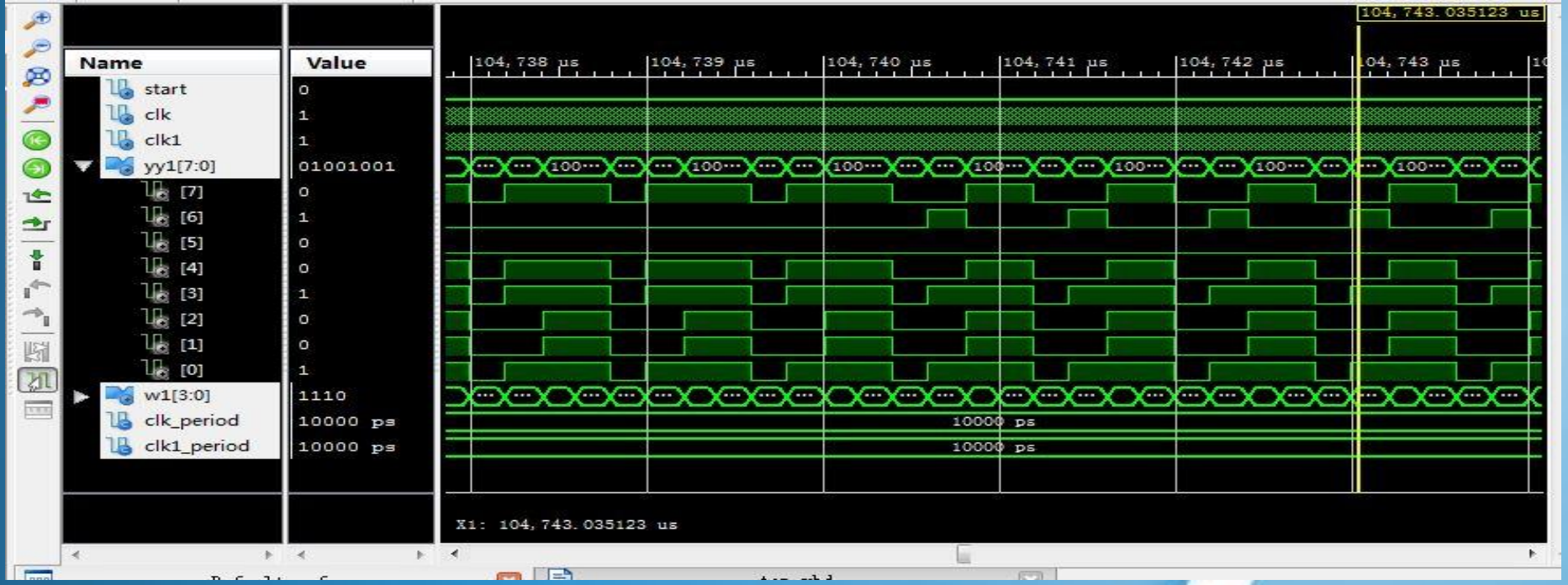
波形驱动代码

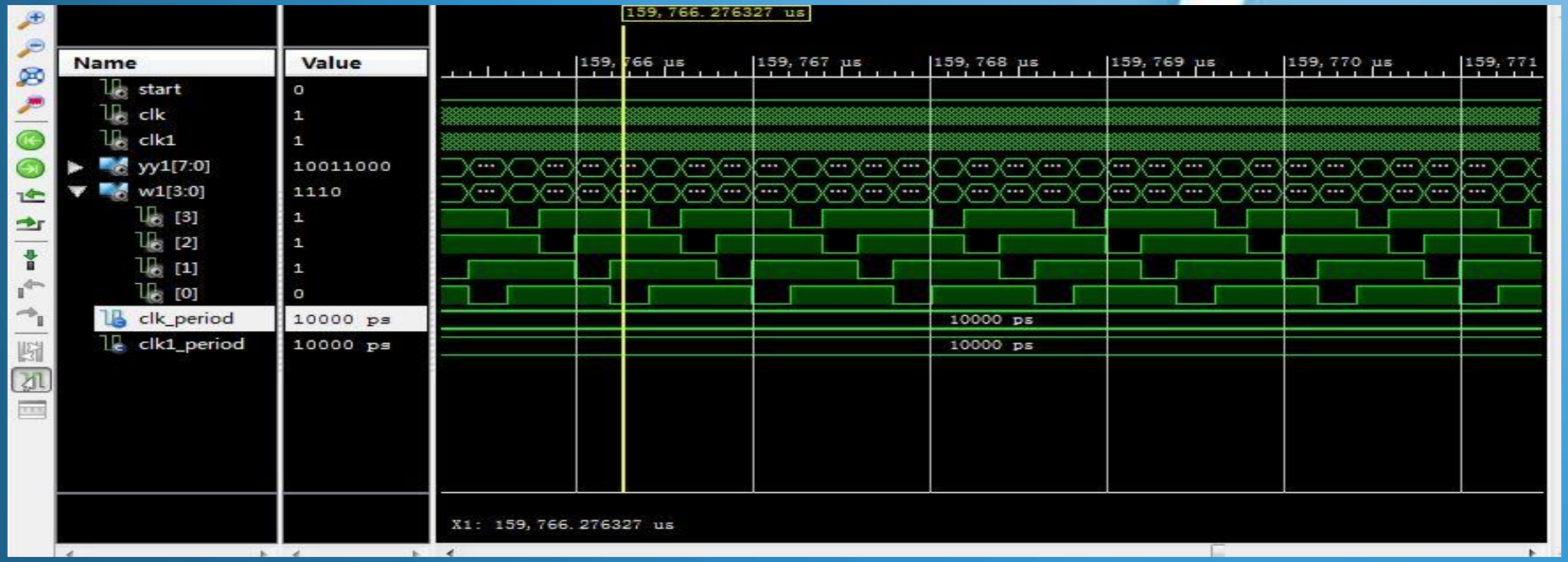
```
 uut: plj PORT MAP (
        start => start,
        clk => clk,
        clk1 => clk1,
        yy1 => yy1,
        w1 => w1
       );
process
begin
clk<='0';
wait for 10 ns;
clk<='1';
wait for 10 ns;
end process;
```

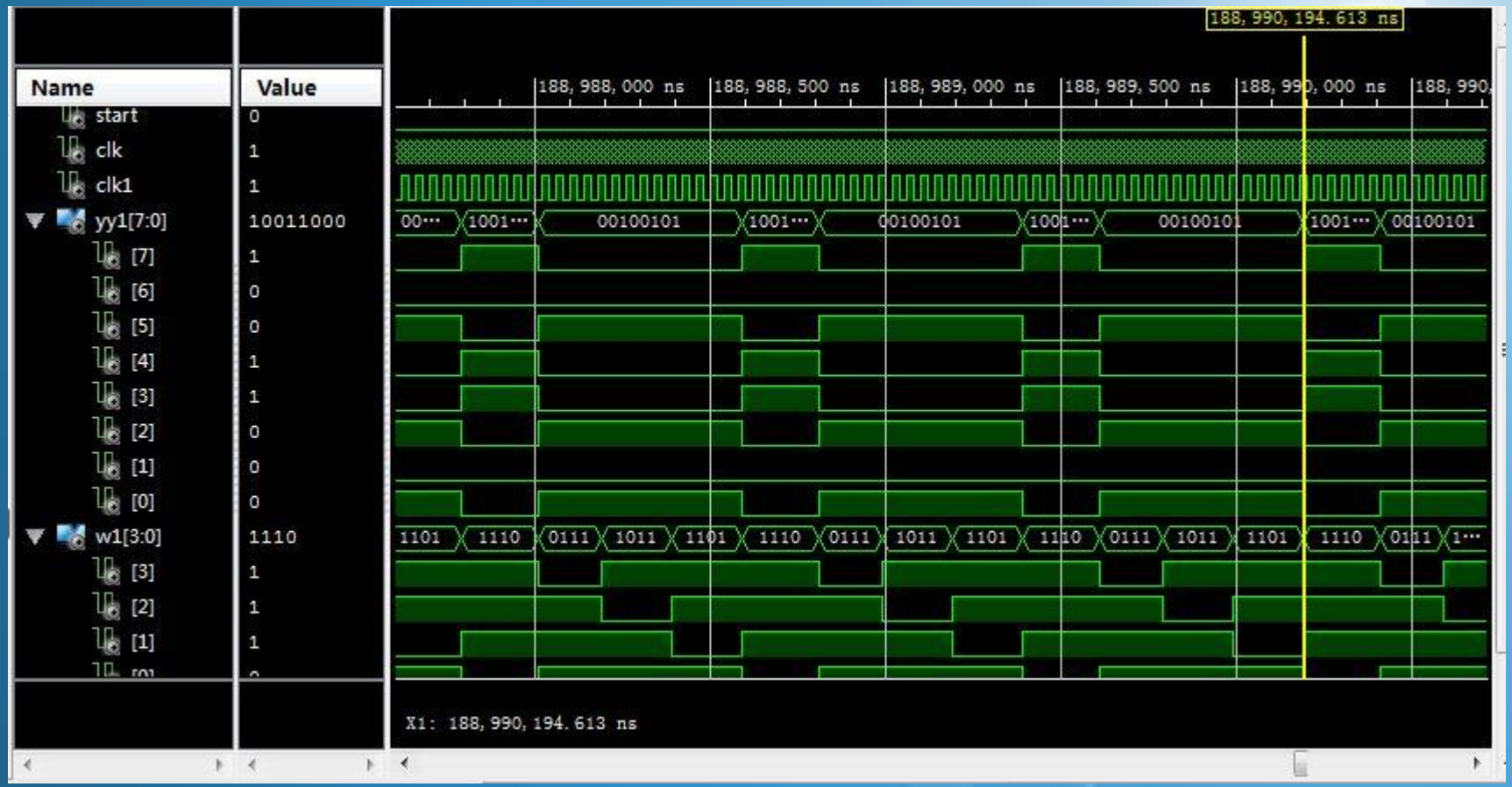
```
  process
begin
start<='1';
wait for 100 ms;
start<='0';
wait for 100 ms;
end process;

process
begin
clk1<='0';
wait for  1 ms;
clk1<='1';
wait for   1 ms;
end process;

END;
```

仿真结果


Re-launch
1.00us
Objects
Simulation Objects for uut
Object Name
Value
start 1
clk 0
clk1 0
yy1[7:0] 00000011
w1[3:0] 1011
b1[3:0] 0000
b2[3:0] 0000
b3[3:0] 0000
b4[3:0] 0000
b5[3:0] 0000
b6[3:0] 0000
b7[3:0] 0000
bcd[3:0] 0000
q 0
qq 1010
en 0
bclk 0
sss[3:0] 1111
bcd0[3:0] 0000
bcd1[3:0] 0000
Name
Value
start 0
clk 0
clk1 0
yy1[7:0] 00000011
[7] 0
[6] 0
[5] 0
[4] 0
[3] 0
[2] 0
[1] 1
[0] 1
w1[3:0] 1011
clk_period 10000 ps
clk1_period 10000 ps
100,001.007904 us
100,000 us
100,001 us
100,002 us
100,003 us
100,004 us
100,005 us
00000011
10000 ps
10000 ps
X1: 100,001.007904 us
Default.wcfg
top.vhd


Name
Value
start
0
clk
1
clk1
1
yy1[7:0]
01001001
[7]
0
[6]
1
[5]
0
[4]
0
[3]
1
[2]
0
[1]
0
[0]
1
w1[3:0]
1110
clk_period
10000 ps
clk1_period
10000 ps
104, 743. 035123 us
104, 738 us
104, 739 us
104, 740 us
104, 741 us
104, 742 us
104, 743 us
10000 ps
10000 ps
X1: 104, 743. 035123 us


Name
Value
start
0
clk
1
clk1
1
yy1[7:0]
10011000
w1[3:0]
1110
[3]
1
[2]
1
[1]
1
[0]
0
clk_period
10000 ps
clk1_period
10000 ps
159, 766. 276327 us
159, 766 us
159, 767 us
159, 768 us
159, 769 us
159, 770 us
159, 771
10000 ps
10000 ps
X1: 159, 766. 276327 us


Name
Value
start
0
clk
1
clk1
1
yy1[7:0]
10011000
[7]
1
[6]
0
[5]
0
[4]
1
[3]
1
[2]
0
[1]
0
[0]
0
w1[3:0]
1110
[3]
1
[2]
1
[1]
1
188, 990, 194.613 ns
188, 988, 000 ns
188, 988, 500 ns
188, 989, 000 ns
188, 989, 500 ns
188, 990, 000 ns
00···
1001···
00100101
1101
1110
0111
1011
X1: 188, 990, 194.613 ns

遇到的困难

根据源程序来写仿真波形

复位时钟频率合适才能使复位信号到来之前计数完毕

还要使仿真时间足够长

多次调整 run ---- time

```
process
begin
clk<='0';
wait for 10 ns;
clk<='1';
wait for 10 ns;
end process;
```

```
process
begin
start<='1';
wait for 100 ms;
start<='0';
wait for 100 ms;
end process;
```

心得体会

一句话：看起来容易　实践起来难

谢谢！