Experiment of FPGA

Name:李谨杰 ID:16231235 Table number:27

1. Aim

1. Familiar with the use of programmable logic devices (Altera Corporation FPGA Cyclone series EP1C6Q).
2. Familiar with the hardware description language VHDL.
3. Master the development process of FPGA integrated environment (Altera FPGA Quartus II 9.0).
4. Familiar with the working principle of the core target system board and interface circuit and its function module binding information.
5, familiar with and master the choice of download line mode and download files.

2. Principle

1. Learn and master input and timing and function simulation methods such as text and graphics.
2. Learn and be familiar with single module functions such as gate circuit, combination circuit and sequential circuit.
3. Learn and design various logic functions of different state machines.
4. Learn and design from a single module → more functional module integration → system integration method.
5. learn and choose a variety of mode display (LED display, m-shaped digital tube display, seven-segment digital tube → dynamic scanning or static scanning display, LED dot matrix display various static or mobile characters and graphics , and the LCD displays various static or mobile characters and graphics.
6. According to your own interests and wishes, you can select or set the function title from the experimental catalog given below.
7. the number of experiments is not required, the key is to look at the quality, whether it is self-written, debugged, and realized.

3. Content

1. According to the instruction book integrated development environment chapter, the whole process of text programming example 1 and graphic programming example 2 is realized.
2. Optional gate circuit, combination circuit, and sequential circuit experiment each complete a logic function, and its implementation scheme is self-defined. When the input and output pins of the FPGA target device are bound, the input pins are bound with high/low level, single pulse, various divided continuous pulses and other signals. The output pins can be bound to LEDs and seven segments. Digital tube, LED dot matrix and other display modes.
3. Based on completing the 1-digit decimal counter, the logic function of multi-digit decimal counters such as 2 or 3 can be added and displayed by a multi-digit seven-segment digital tube.
4. Display any character, graphic and other information with LED dot matrix.

4. Experiment code and result

1.4-bits binary addr

1)Code

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity wenben is
    port(
          clk:in std_logic;
           rst:in std_logic;
           q:out   std_logic_vector(3   downto
0));
                          end;
architecture b1 of wenben is
signal q1:std_logic_vector(3 downto 0);
   begin
      process(clk,rst)
         begin
            if(clk'event and clk='1')then
                q1<= q1+1;
            end if;
         end process;
         q<= not q1;
end;
```

2) Pin assignment

Node Name	Location
clk	PIN_P20
rst	PIN_N18
q[3]	PIN_U12
q[2]	PIN_V12
q[1]	PIN_W13
q[0]	PIN_P20

3)Operation

Connect input signal ‘clk’ to FRQ_Q21 (1Hz) by wire. Give hign level to input signal ‘rst’. Make sure the mode is ‘00XX’.

4)Result

LED1-LED2-LED3-LED4 display in the order of ‘0000’,’0001’,’0010’,’0011’……’1111’,’0000’……. If I input an reset signal, all LEDs change to ‘0’ level.

2. 2-bit decimal counter

1)Code

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity count_100 is
port(
      clk,en,rst:in std_logic;
      seg:out std_logic_vector(6 downto 0);
      xuan:out std_logic_vector(7 downto 0);
      cout:out std_logic;
                  dec:out std_logic);
end entity count_100;

architecture bhv of count_100 is
signal count: std_logic_vector(6 downto 0);   --max 128
signal q1,q2,q3: std_logic_vector(3 downto 0);
signal neijin:std_logic;
signal jin: std_logic;
dec<='0';
      begin
      process(clk)
            begin
```

```
        if(clk'event and clk='1') then
            count <= count+1;
            if(count > 100) then
                count <= "0000000";
                jin <= '1';
            else
                jin <= '0';
            end if;
        end if;
        if(count(0)='0') then
                q3 <= q1;
                xuan <= "11111110";
        else
                q3 <= q2;
                xuan <= "11111101";
        end if;
        case q3 is
            when "0000"=> seg<="0000001";--1111110
            when "0001"=> seg<="1001111";--0110000
            when "0010"=> seg<="0010010";--1101101
            when "0011"=> seg<="0000110";--1111001
            when "0100"=> seg<="1001100";--0110011
            when "0101"=> seg<="0100100";--1011011
            when "0110"=> seg<="0100000";--1011111
            when "0111"=> seg<="0001111";--1110000
            when "1000"=> seg<="0000000";--1111111
            when "1001"=> seg<="0001100";--1110011
            when others=> seg<="1111111";--0000000
        end case;
    end process;

--计数
    process(jin,en,rst)
        begin
            if(jin'event and jin='1') then
                if(rst = '1')then
                    q1 <= (others=>'0');
                    q2 <= (others=>'0');
                    neijin <= '0';
                    cout <= '0';
                elsif(en='1')then
                    if(q1 < 9)then q1 <= q1+1;
                    else q1 <= (others=>'0');
                    end if;
                    if(q1 = 8)then neijin <= '1';
                    else neijin <= '0';
                    end if;
                    if(neijin = '1')then
                        if(q2 < 9)then q2 <= q2+1;
                        else q2 <= (others=>'0');
                        end if;
                    end if;
                    if(q2=9 and q1=9)then cout <= '1';
                    else cout <='0';
                    end if;
                end if;
            end if;
        end process;
    end architecture bhv;
```

2)Pin assignment

Node Name	Location
clk	PIN_P20
count	PIN_U12
dec	PIN_M19
en	PIN_N18

rst	PIN_M20
seg[6]	PIN_M21
seg[5]	PIN_N20
seg[4]	PIN_N21
seg[3]	PIN_P21
seg[2]	PIN_R21
seg[1]	PIN_W20
seg[0]	PIN_AA20
xuan[7]	PIN_V16
xuan[6]	PIN_AA17
xuan[5]	PIN_U22
xuan[4]	PIN_V22
xuan[3]	PIN_W22
xuan[2]	PIN_Y22
xuan[1]	PIN_Y21
xuan[0]	PIN_AB20

3)Experimental operation

Connect the input signal ‘clk’ clock (Pin P20) to FRQ_Q21 (1Hz) by wire. Set the input signal ‘rst’ to "0". Set the input signal ‘en’ to "1". Change the mode to ‘10XX’.

4）Result

Digital tubes display number from ‘00’ to ‘99’ Cyclically. When I input reset signal, the number turns to ‘00’.

3. Multi-person responder

1) Code

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity qiangdaqi is
        port(rst,clk2:in std_logic;
             s0,s1,s2,s3:in std_logic;
             states:buffer     std_logic_vector(3
downto 0);
             light:buffer      std_logic_vector(3
downto 0);
warm:out std_logic);
end qiangdaqi ;

architecture one of qiangdaqi     is
signal st:std_logic_vector(3 downto 0);
begin

p1:
process(s0,rst,s1,s2,s3,clk2)
   begin
      if rst='0'    then
         warm<='0';st<="0000";
         elsif clk2'event and clk2='1' then
     if (s0='1' or st(0)='1')and not( st(1)='1' or
st(2)='1' or st(3)='1' )
              then st(0)<='1';
     end if ;
  if (s1='1' or st(1)='1')and not( st(0)='1' or
st(2)='1' or st(3)='1' )
              then st(1)<='1';
     end if ;
  if (s2='1' or st(2)='1')and not( st(0)='1' or
st(1)='1' or st(3)='1' )
              then st(2)<='1';
       end if ;
  if (s3='1' or st(3)='1')and not( st(0)='1' or
st(1)='1' or st(2)='1' )
              then st(3)<='1';
       end if ;
warm<=st(0) or st(1) or st(2) or st(3);
```

```
end if ;
end process p1;

p2:
process(states(0),states(1),states(2),states(3),light)
  begin
  if (st="0000") then states<="0000";
elsif (st<="0001") then states<="0001";
elsif (st<="0010") then states<="0010";
elsif (st<="0100") then states<="0011";
elsif (st<="1000") then states<="0100";
end if;
light<=st;
end process p2;

end one;
```

2) Pin assignment

Node Name	Location
clk2	PIN_P20
light[3]	PIN_W13
light[2]	PIN_V15
light[1]	PIN_V12
light[0]	PIN_U12
rst	PIN_C3
s0	PIN_N18
s1	PIN_M20
s2	PIN_AA15
s3	PIN_V13
states[3]	
states[2]	
states[1]	
states[0]	
warm	
Node Name	Location
clk2	PIN_P20
light[3]	PIN_W13
light[2]	PIN_V15
light[1]	PIN_V12
light[0]	PIN_U12
rst	PIN_C3
s0	PIN_N18
s1	PIN_M20
s2	PIN_AA15
s3	PIN_V13
states[3]	
states[2]	
states[1]	
states[0]	
warm	

3) Experiment operation

Connect the input signal ‘clk’ clock (Pin P20) to FRQ_Q6 (32768 Hz) by wire. Change the mode to ‘00XX’.

4) Experiment phenomenon

If I pressed any switch of four, the LED of that switch would light, and be locked, which meant no matter how I change the condition of four switches, there was no change of LED lights. If I wanted to start again, I should input reset signal by SW11.

4. 4x4 keyboard circuit

1） Code

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
entity jianpan is
port(clk: in    std_logic;
start:in    std_logic;
KBCol: in std_logic_vector(3 downto 0);
KBRow: out std_logic_vector(3 downto 0);
seg7: out std_logic_vector(6 downto 0);
scan: out std_logic_vector(7 downto 0) );
end jianpan;
architecture bev of jianpan is
signal count: std_logic_vector(1 downto 0);
signal sta: std_logic_vector(1 downto 0);
begin
scan<="00000001";
a:
process(clk)
begin
if(clk' event and clk='1')    then
count<=count+1 ;
end if;
end process a;
b:
process(clk)
begin
if(clk' event and clk='1')    then
case count(1 downto 0) is
when    "00"=> KBRow<="0111";
sta<="00" ;
when    "01"=> KBRow<="1011";
sta<="01";
when    "10"=> KBRow<="1101";
sta<="10";
when "11"=>KBRow<="1110";
sta<="11";
when others=>KBRow<="1111";
end case;

end if;
end process b;

process(clk,start)
begin
if start='0' then
seg7<="0000000";
else if(clk' event and clk='1')    then
case sta is
when"00"=>
case KBCol is
when    "1110"=>seg7<="1001110";
When    "1101"=>seg7<="0111101";
When    "1011"=>seg7<="1001111";
When    "0111"=>seg7<="1000111";
When others=>seg7<="0000000";
end case;
when"01"=>
case KBCol is
when     "1110"=>    seg7<="1111111";
when     "1101"=>    seg7<="1110011";
when     "1011"=>    seg7<="1110111";
when     "0111"=>    seg7<="0011111";
When others=>       seg7<="0000000";
end case;
when"10"=>
case KBCol is
when    "1110"=>    seg7<="0110011";
when    "1101"=>    seg7<="1011011";
when     "1011"=>    seg7<="1011111";
when     "0111"=>    seg7<="1110000";
when others=>       seg7<="0000000";
```

```
end case;
when"11"=>
case KBCol is
when    "1110"=>    seg7<="1111110";
when    "1101"=>    seg7<="0110000";
when    "1011"=>    seg7<="1101101";
when    "0111"=>    seg7<="1111001";
when others=>       seg7<="0000000";
end case;
when others=> seg7<="0000000";
end case;
end if;
  end if;
end process c;
end bev;
```

2) Pin assignment

Node Name	Location
clk	PIN_P20
KBCol[3]	PIN_A13
KBCol[2]	PIN_F9
KBCol[1]	PIN_D10
KBCol[0]	PIN_B10
KBRow[3]	PIN_C4
KBRow[2]	PIN_A16
KBRow[1]	PIN_A15
KBRow[0]	PIN_A14
scan[7]	PIN_AB20
scan[6]	PIN_Y21
scan[5]	PIN_Y22
scan[4]	PIN_W22
scan[3]	PIN_V22
scan[2]	PIN_U22
scan[1]	PIN_AA17
scan[0]	PIN_V16
seg7[6]	PIN_AA20
seg7[5]	PIN_W20
seg7[4]	PIN_R21
seg7[3]	PIN_P21
seg7[2]	PIN_N21
seg7[1]	PIN_N20
seg7[0]	PIN_M21
start	PIN_N18

3) Experiment operation

Change the mode to ‘10xx’

4) Result

Press 4x4 keyboard button, the digital tube would show letters printed on the board.

5. Display letters “BUAA” by LED dot matrix

1) Code

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_Unsigned.all;
use ieee.std_logic_ARITH.all;
```

```
ENTITY led is
port(clk:in std_logic;
        en:in std_logic;
        hang:out std_logic_vector(15 downto 0);
count: out std_logic_vector(3 downto 0));
End led;

Architecture dianzhen of led is
signal osc:std_logic;
signal x:std_logic_vector(0 downto 0);
signal y:std_logic_vector(0 downto 0);
signal count1:std_logic_vector(3 downto 0);
signal count2:std_logic_vector(25 downto 0);
signal data:std_logic_vector(15 downto 0);
signal
d0,d1,d2,d3,d4,d5,d6,d7,d8,d9,d10,d11,d12,d1
3,d14,d15:std_logic_vector(15 downto 0);
signal
a0,a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,
a14,a15:std_logic_vector(15 downto 0);
signal
b0,b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b1
3,b14,b15:std_logic_vector(15 downto 0);

Begin
hang<=data;
  a0<="0000000000000000";
  a1<="0111111111111110";
  a2<="0100000010000010";
  a3<="0010000101000100";
  a4<="0001001000101000";
  a5<="0000110000010000";
  a6<="0000000000000000";
  a7<="0000000000000000";
  a8<="0001111111111110";
  a9<="0010000000000000";
a10<="0100000000000000";
a11<="0100000000000000";
a12<="0010000000000000";
a13<="0001111111111110";
a14<="0000000000000000";
a15<="0000000000000000";
b0<="0000000000000000";
b1<="0111000000000000";
b2<="0000001110000000";
b3<="0000001000011000";
b4<="0000001000000110";
b5<="0000001000011000";
b6<="0000001110000000";
b7<="0111000000000000";
b8<="0000000000000000";
b9<="0111000000000000";
b10<="0000001110000000";
b11<="0000001000011000";
b12<="0000001000000110";
b13<="0000001000011000";
b14<="0000001110000000";
b15<="0111000000000000";

process(osc,en,clk,x,y,count2)
begin
   osc<=not clk;
x<="1";
   if(osc='1' and osc'event) then
count2<=count2+1;
if (count2=100000) then
y<=y+1;
count2<=(others=>'0');
IF(y="1")then
y<="0";
end if;
end if;
if(x="0")then
       if(en='1')then
          if count1<="0000" then
            data<=d0; count1<="0001";
          elsif count1<="0001"then
            data<=d1; count1<="0010";
          elsif count1<="0010"then
          data<=d2; count1<="0011";
          elsif count1<="0011"then
            data<=d3; count1<="0100";
          elsif count1<="0100"then
            data<=d4; count1<="0101";
          elsif count1<="0101"then
            data<=d5; count1<="0110";
          elsif count1<="0110"then
```

```
            data<=d6; count1<="0111";
          elsif count1<="0111"then
            data<=d7; count1<="1000";
elsif count1<="1000"then
            data<=d8; count1<="1001";
elsif count1<="1001"then
            data<=d9; count1<="1010";
elsif count1<="1010"then
            data<=d10; count1<="1011";
elsif count1<="1011"then
            data<=d11; count1<="1100";
elsif count1<="1100"then
            data<=d12; count1<="1101";
elsif count1<="1101"then
            data<=d13; count1<="1110";
elsif count1<="1110"then
            data<=d14; count1<="1111";
elsif count1<="1111"then
            data<=d15; count1<="0000";
          end if;
       end if;
elsif (x="1")then
    if(en='1')then
if (y="1") then
          if count1<="0000" then
            data<=b0; count1<="0001";
          elsif count1<="0001"then
            data<=b1; count1<="0010";
          elsif count1<="0010"then
          data<=b2; count1<="0011";
          elsif count1<="0011"then
            data<=b3; count1<="0100";
          elsif count1<="0100"then
            data<=b4; count1<="0101";
          elsif count1<="0101"then
            data<=b5; count1<="0110";
          elsif count1<="0110"then
            data<=b6; count1<="0111";
          elsif count1<="0111"then
            data<=b7; count1<="1000";
          elsif count1<="1000"then
            data<=b8; count1<="1001";
          elsif count1<="1001"then
            data<=b9; count1<="1010";
          elsif count1<="1010"then
            data<=b10; count1<="1011";
          elsif count1<="1011"then
            data<=b11; count1<="1100";
          elsif count1<="1100"then
            data<=b12; count1<="1101";
          elsif count1<="1101"then
            data<=b13; count1<="1110";
          elsif count1<="1110"then
            data<=b14; count1<="1111";
          elsif count1<="1111"then
            data<=b15; count1<="0000";
            end if;
elsif (y="0") then
          if count1<="0000" then
            data<=a0; count1<="0001";
          elsif count1<="0001"then
            data<=a1; count1<="0010";
          elsif count1<="0010"then
            data<=a2; count1<="0011";
          elsif count1<="0011"then
            data<=a3; count1<="0100";
          elsif count1<="0100"then
            data<=a4; count1<="0101";
          elsif count1<="0101"then
            data<=a5; count1<="0110";
          elsif count1<="0110"then
            data<=a6; count1<="0111";
          elsif count1<="0111"then
            data<=a7; count1<="1000";
          elsif count1<="1000"then
            data<=a8; count1<="1001";
          elsif count1<="1001"then
            data<=a9; count1<="1010";
          elsif count1<="1010"then
            data<=a10; count1<="1011";
          elsif count1<="1011"then
            data<=a11; count1<="1100";
          elsif count1<="1100"then
            data<=a12; count1<="1101";
          elsif count1<="1101"then
            data<=a13; count1<="1110";
          elsif count1<="1110"then
            data<=a14; count1<="1111";
```

```
        elsif count1<="1111"then
            data<=a15; count1<="0000";
      end if;
  end if;
end if;
end if;
```

```
end if;

end process;
count<=count1;

    end;
```

2) Pin assignment

Node Name	Location
clk	PIN_P20
count[3]	PIN_C4
count[2]	PIN_A16
count[1]	PIN_A15
count[0]	PIN_A14
en	PIN_N18
hang[15]	PIN_A13
hang[14]	PIN_F9
hang[13]	PIN_D10
hang[12]	PIN_B10
hang[11]	PIN_B9
hang[10]	PIN_B8
hang[9]	PIN_B7
hang[8]	PIN_E14
hang[7]	PIN_C15
hang[6]	PIN_F11
hang[5]	PIN_C13
hang[4]	PIN_E11
hang[3]	PIN_B6
hang[2]	PIN_A6
hang[1]	PIN_A5
hang[0]	PIN_A4

3) Experiment operation

Connect the input signal ‘clk’ clock (Pin P20) to FRQ_Q6 (32768 Hz) by wire. Change the mode to ‘00XX’.

4) Result

Letters ‘B U’, ‘A A’ shown on the dot matrix cyclically.

6. Display information by LCD

1) Code

```
library ieee;
use IEEE.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity lcd1602 is
    port(clk, reset: in std_logic;
         LCD_Data:  out  std_logic_vector(8
downto 0);
         en: out std_logic;
         rw: out std_logic);
end lcd1602;

architecture gongneng of   lcd1602 is
signal LCD_Clk: std_logic;
signal s : integer range 0 to 10000000;
```

```
begin
process(clk, reset)
begin
      if reset = '0' then
            LCD_Clk <= '0';
            s <= 0;
      elsif clk'event and clk = '1' then
            if s = 5000000 then
                  s <= 0;
                  LCD_Clk <= not LCD_Clk;
            else
                  s <= s + 1;
end if;
end if;
end process;

rw <='0';
en <= LCD_Clk;

process(LCD_Clk)
variable cnt: std_logic_vector(4 downto 0):="00000";
begin
    if Reset='0'then
LCD_Data<="000000001";              -- Reset 清屏  Left to right D8-D0
cnt:="00000";          --计数器清零
elsif rising_edge(LCD_Clk) then
            if cnt<"01111" then cnt:=cnt+1;
         else cnt:="00000";
         end if;
--设计计数器，每次计数间隔 LCD_CLK 定义的一个周期
case cnt is
  when
"00000"=>LCD_Data<="000111000";--/*  设置 8 位格式,2 行,5*7*/  ,
  when "00001"=>LCD_Data<="000001100";
--/*整体显示,关光标,光标闪烁/
  when
"00010"=>LCD_Data<="000000001";--清屏
  when "00011"=>LCD_Data<="000000110";
--/*显示移动格式，看最后两位，10 表示光标右移
  when
"00100"=>LCD_Data<="010000000";-- 设 定显示的位置在 00H+80H, 即显示屏第一行第一个位置
      when
"00101"=>LCD_Data<="101001100"; --L
      when
"00110"=>LCD_Data<="101101001";--i
      when
"00111"=>LCD_Data<="101101010";--j
      when
"01000"=>LCD_Data<="101101001";--i
      when
"01001"=>LCD_Data<="101101110";--n
      when
"01010"=>LCD_Data<="101101010";--j
      when
"01011"=>LCD_Data<="101101001";--i
      when
"01100"=>LCD_Data<="101100101";--e
      when
"01101"=>LCD_Data<="011000000";-- 设 定显示的位置在 10H+80H，即显示屏第 2 行第一个位置
      when
"01110"=>LCD_Data<="101011010";---Z
      when
"01111"=>LCD_Data<="101101000";---h
      when
"10000"=>LCD_Data<="101100001";---a
      when
"10001"=>LCD_Data<="101101111";---o
      when
"10010"=>LCD_Data<="101110001";---q
      when
"10011"=>LCD_Data<="101101001";---i
      when
"10100"=>LCD_Data<="101100001";---a
      when
"10101"=>LCD_Data<="101101110";---n
   when others =>LCD_Data<="101000100";
  end case;
  end if;
```

end process; end gongneng;

2) Pin assignment

Clk	P20
En	A4
LCD_Data[8]	A6
LCD_Data[7]	A3
LCD_Data[6]	F7
LCD_Data[5]	E6
LCD_Data[4]	C7
LCD_Data[3]	E5
LCD_Data[2]	C3
LCD_Data[1]	AB18
LCD_Data[0]	AB17
reset	N18
rw	A5

3) Experiment operation

Connect the input signal ‘clk’ clock (Pin P20) to FRQ_Q0 (24MHz) by wire.

4) Result

Word ‘Lijinjie’ was displayed on the first row, and then word ‘Zhaoqian’ was displayed on the second row one by one. After that, letters vanished, and began to display again.

5. Summary of this experiment

I finished this experiment with my partner Zhaoqian. It’s very hard to know FPGA very well in a short time, so most code is based on the code on the instruction book. From these two experiments and several simple programs, I have learned basic grammar of VHDL, and known how to design FPGA to have some functions.

The most important point is that coding FPGA is a process of hardware design. We can’t equal this process to advanced language programming, like C.

References

[1] VHDL_LCD 1602 https://wenku.baidu.com/view/00e9526e7e21af45b307a8b0.html
[2] LCD 1602 百度百科 https://baike.baidu.com/item/LCD1602/6014393?fr=aladdin
[3] 百度文库：2016年北航FPGA实验报告 https://wenku.baidu.com/view/db5018322379168884868762caaedd3383c4b5e4.html
[4] FPGA 指导书----北航电工电子中心