VHDL 语言设计 4 位并行加法器

```vhdl
library ieee;
use ieee.std_logic_1164.all;
entity zjw_bingxingjiafaqi is
    port (A,B: in std_logic_vector(3 downto 0);
    Dout: out std_logic_vector(3 downto 0);
    Dout1:out std_logic);
end zjw_bingxingjiafaqi;

architecture bx of zjw_bingxingjiafaqi is
component quanjiaqi_vhdl
    port (X,Y,Z: in std_logic;
    S,C: out std_logic);
end component;
signal D: std_logic_vector(2 downto 0);
begin
    g0: quanjiaqi_vhdl port map(A(0),B(0),'0',Dout(0),D(0));
    g1: quanjiaqi_vhdl port map(A(1),B(1),D(0),Dout(1),D(1));
    g2: quanjiaqi_vhdl port map(A(2),B(2),D(1),Dout(2),D(2));
    g3: quanjiaqi_vhdl port map(A(3),B(3),D(2),Dout(3),Dout1);
end bx;
```

VHDL 语言设计 4 位串行加法器

```
 1  library ieee;
 2  use ieee.std_logic_1164.all;
 3 entity zjw_chuanxingjiafaqi is
 4     port (F,E: in std_logic_vector(3 downto 0);
 5      M4: out std_logic_vector(3 downto 0));
 6  end zjw_chuanxingjiafaqi;
 7
 8 architecture bre of zjw_chuanxingjiafaqi is
 9
10 component quanjiaqi_vhdl
11     port (X,Y,Z: in std_logic;
12      S,C: out std_logic);
13  end component;
14 component yiwei_vhdl
15     port (A: in std_logic_vector(3 downto 0);
16           C: in std_logic;
17           B: out std_logic_vector(3 downto 0));
18  end component;
19  signal  D: std_logic_vector(3 downto 0);
20  signal  N: std_logic_vector(3 downto 0);
21  signal  M0: std_logic_vector(3 downto 0);
22  signal  M1: std_logic_vector(3 downto 0);
23  signal  M2: std_logic_vector(3 downto 0);
24  signal  M3: std_logic_vector(3 downto 0);
25  signal  N0: std_logic_vector(3 downto 0);
26  signal  N1: std_logic_vector(3 downto 0);
27  signal  N2: std_logic_vector(3 downto 0);
28  signal  N3: std_logic_vector(3 downto 0);
29  signal  Dout: std_logic_vector(3 downto 0);
30  begin
31      M0 <= F;
32      N0 <= E;
33  g0: quanjiaqi_vhdl port map(M0(0),N0(0),'0',Dout(0),D(0));
34      M1 <= Dout(0) & M0(3 downto 1);
35      N1 <= Dout(0) & N0(3 downto 1);
36
37  g1: quanjiaqi_vhdl port map(M1(0),N1(0),D(0),Dout(1),D(1));
38      M2 <= Dout(1) & M1(3 downto 1);
39      N2 <= Dout(1) & N1(3 downto 1);
```

```
36
37  g1: quanjiaqi_vhdl port map(M1(0),N1(0),D(0),Dout(1),D(1));
38      M2 <= Dout(1) & M1(3 downto 1);
39      N2 <= Dout(1) & N1(3 downto 1);
40
41  g2: quanjiaqi_vhdl port map(M2(0),N2(0),D(1),Dout(2),D(2));
42      M3 <= Dout(2) & M2(3 downto 1);
43      N3 <= Dout(2) & N2(3 downto 1);
44
45  g3: quanjiaqi_vhdl port map(M3(0),N3(0),D(2),Dout(3),D(3));
46      M4 <= Dout(3) & M3(3 downto 1);
47
48  end ;
```

VHDL 语言设计模型机运算器 ALU

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity zjw_ALU is
port(R1, R2: in std_logic_vector(7 downto 0);
     S: in std_logic_vector(3 downto 0);
     M:in std_logic;
     C, Z: out std_logic;
     Rlout: out std_logic_vector(7 downto 0));
end zjw_ALU;

architecture bev of zjw_ALU is
signal instruct: std_logic_vector(3 downto 0);
signal sum: std_logic_vector(8 downto 0);
signal d: std_logic;
begin
instruct <= S;
C <= sum(8);
Z <= '1' when sum="00000000" else '0';
process(instruct,M)
begin
if M='1' then
    if instruct = "1001" then
        sum <= ('0' & R1) + ('0' & R2);
        Rlout <= sum(7 downto 0);
    elsif instruct = "0110" then
        sum <= ('0' & R1) - ('0' & R2);
        Rlout <= sum(7 downto 0);
    elsif instruct = "1010" or instruct = "0100" then
        Rlout <= R1;
    elsif instruct = "1111" then
        Rlout <= R2;
    else
        Rlout <= "ZZZZZZZZ";
    end if;
else
    if instruct = "1011" then
        Rlout <= R1 or R2;
    elsif instruct = "0101" then
        Rlout <= not R1;
    else
        Rlout <= "ZZZZZZZZ";
    end if;
end if;
end process;
end;
```

VHDL 语言设计移位逻辑

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

entity zjw_yiweiluoji is
port(F,FRL,FRR: in std_logic;
     data_in: in std_logic_vector(7 downto 0);
     FZ,FC: out std_logic;
     data_out: out std_logic_vector(7 downto 0));
end zjw_yiweiluoji;

architecture behave of zjw_yiweiluoji is
signal d: std_logic_vector(7 downto 0);
begin
     data_out<=d;
     process(F,FRL,FRR,data_in)
     begin
     if F='1' then
          d <=data_in;
          FC <= '0';
          FZ <= '0';
     elsif FRL='1' then
          d <=data_in(6 downto 0)&data_in(7);
          FC <= data_in(7);
          if d="00000000" then
               FZ <= '1';
          else
               FZ <= '0';
          end if;
     elsif FRR='1' then
          d <=data_in(7) & data_in(7 downto 1);
          FC <= data_in(0);
          if d="00000000" then
               FZ <= '1';
          else
               FZ <= '0';
          end if;
     else
          d <="ZZZZZZZZ";
          FC <= '0';
          FZ <= '0';
     end if;
     end process;
end;
```