原码二位乘法器设计

一、VHDL 描述

1.1 顶层模块设计

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity multiplier is
    Port(
        clk,start:in std_logic;
        ain,bin:in std_logic_vector(7 downto 0);
        done:out std_logic;
        sout:inout std_logic_vector(15 downto 0)
    );
end multiplier;

architecture Behavioral of multiplier is

component multiplier_ctrl
    Port (
        clk,start:in std_logic;
        clkout,rstall,done:out std_logic
     );
end component;
component multiplier_8bitshiftreg
    Port (
        clk,load:in std_logic;
        din:in std_logic_vector(7 downto 0);
        qb0,qb1:out std_logic
     );
end component;
component multiplier_16bitreg
    Port (
        clk,clr:in std_logic;
        d:in std_logic_vector(8 downto 0);
        q:out std_logic_vector(15 downto 0)
     );
end component;
component multiplier_selector
    Port (
        clk,rst:in std_logic;
```

```
        a0,a1,cin:in std_logic;
        din:in std_logic_vector(7 downto 0);
        cout:out std_logic;
        dout:out std_logic_vector(7 downto 0)
     );
end component;
component multiplier_8bitadder
    Port (
        clk,rst:in std_logic;
        cin:in std_logic;
        ain,bin:in std_logic_vector(7 downto 0);
        sout:out std_logic_vector(8 downto 0)
     );
end component;

signal clk_line:std_logic;
signal rst_line:std_logic;
signal cin_line:std_logic;
signal qb1_line,qb0_line:std_logic;
signal bin_line:std_logic_vector(7 downto 0);
signal sout_line:std_logic_vector(8 downto 0);
signal test_line:std_logic_vector(8 downto 0);

begin
multiplier_ctrl_inst:multiplier_ctrl port
map(clk=>clk,start=>start,clkout=>clk_line,rstall=>rst_line,done=>done);
multiplier_8bitshiftreg_inst:multiplier_8bitshiftreg port
map(clk=>clk_line,load=>rst_line,din=>ain,qb0=>qb0_line,qb1=>qb1_line);
multiplier_16bitreg_inst:multiplier_16bitreg port
map(clk=>clk_line,clr=>rst_line,d=>sout_line,q=>sout);
multiplier_selector_inst:multiplier_selector port
map(clk=>clk_line,rst=>rst_line,a0=>qb0_line,a1=>qb1_line,cin=>sout_line(8),din=>b
in,cout=>cin_line,dout=>bin_line);
multiplier_8bitadder_inst:multiplier_8bitadder port
map(clk=>clk_line,rst=>rst_line,cin=>cin_line,ain=>sout(15 downto
8),bin=>bin_line,sout=>sout_line);

end Behavioral;
```

1.2 控制器设计

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity multiplier_ctrl is
    Port (
```

```
        clk,start:in std_logic;
        clkout,rstall,done:out std_logic
     );
end multiplier_ctrl;

architecture Behavioral of multiplier_ctrl is

signal cnt3b:std_logic_vector(2 downto 0);

begin

process(clk,start)
begin
    rstall<=start;
    if(start='1')then cnt3b<="000";
    elsif clk'event and clk='1'then if cnt3b<4 then cnt3b<=cnt3b+1;end if;
    end if;
end process;

process(clk,cnt3b,start)
begin
    if (start='1')then
        clkout<='0';done<='0';
    elsif(start='0')then
        if cnt3b<4 then clkout<=clk;
        else clkout<='0';done<='1';
        end if;
    end if;
end process;

end Behavioral;
```

1.3 8位移位寄存器设计

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity multiplier_8bitshiftreg is
    Port (
        clk,load:in std_logic;
        din:in std_logic_vector(7 downto 0);
        qb0,qb1:out std_logic
     );
end multiplier_8bitshiftreg;

architecture Behavioral of multiplier_8bitshiftreg is
```

```
signal reg8b:std_logic_vector(7 downto 0);

begin

process(clk,load)
begin
    if load='1'then reg8b≤din;qb0≤'0';qb1≤'0';end if;
    if(load='0'and clk='1')then
        qb0≤reg8b(0);
        qb1≤reg8b(1);
        reg8b(5 downto 0)≤reg8b(7 downto 2);
        reg8b(7 downto 6)≤"00";
    end if;
end process;

end Behavioral;
```

1.4 16 位锁存器设计

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity multiplier_16bitreg is
    Port (
        clk,clr:in std_logic;
        d:in std_logic_vector(8 downto 0);
        q:out std_logic_vector(15 downto 0)
     );
end multiplier_16bitreg;

architecture Behavioral of multiplier_16bitreg is

begin

process(clk,clr)
variable sr16b:std_logic_vector(15 downto 0);
begin
    if clr='1'then
        sr16b:="0000000000000000";
    elsif(clr='0'and clk'event and clk='1')then
        sr16b(15 downto 8):=d(7 downto 0);
        sr16b(13 downto 0):=sr16b(15 downto 2);
        sr16b(15):=d(8);
        sr16b(14):=d(8);
    end if;
```

```
    q<=sr16b;
end process;

end Behavioral;
```

1.5 8 位加法器设计

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity multiplier_8bitadder is
    Port (
        clk,rst:in std_logic;
        cin:in std_logic;
        ain,bin:in std_logic_vector(7 downto 0);
        sout:out std_logic_vector(8 downto 0)
     );
end multiplier_8bitadder;

architecture Behavioral of multiplier_8bitadder is
begin

process(clk,rst,ain,bin,cin)
begin
    if(rst='1')then sout<="000000000";
    elsif(rst='0'and clk='0')then
        sout<=('0'& ain)+(cin & bin);
    end if;
end process;

end Behavioral;
```

1.6 数据选择器设计

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity multiplier_8bitadder is
    Port (
        clk,rst:in std_logic;
        cin:in std_logic;
        ain,bin:in std_logic_vector(7 downto 0);
        sout:out std_logic_vector(8 downto 0)
     );
end multiplier_8bitadder;
```

```
12
13  architecture Behavioral of multiplier_8bitadder is
14  begin
15
16  process(clk,rst,ain,bin,cin)
17  begin
18      if(rst='1')then sout<="000000000";
19      elsif(rst='0'and clk='0')then
20          sout<=('0'& ain)+(cin & bin);
21      end if;
22  end process;
23
24  end Behavioral;
```

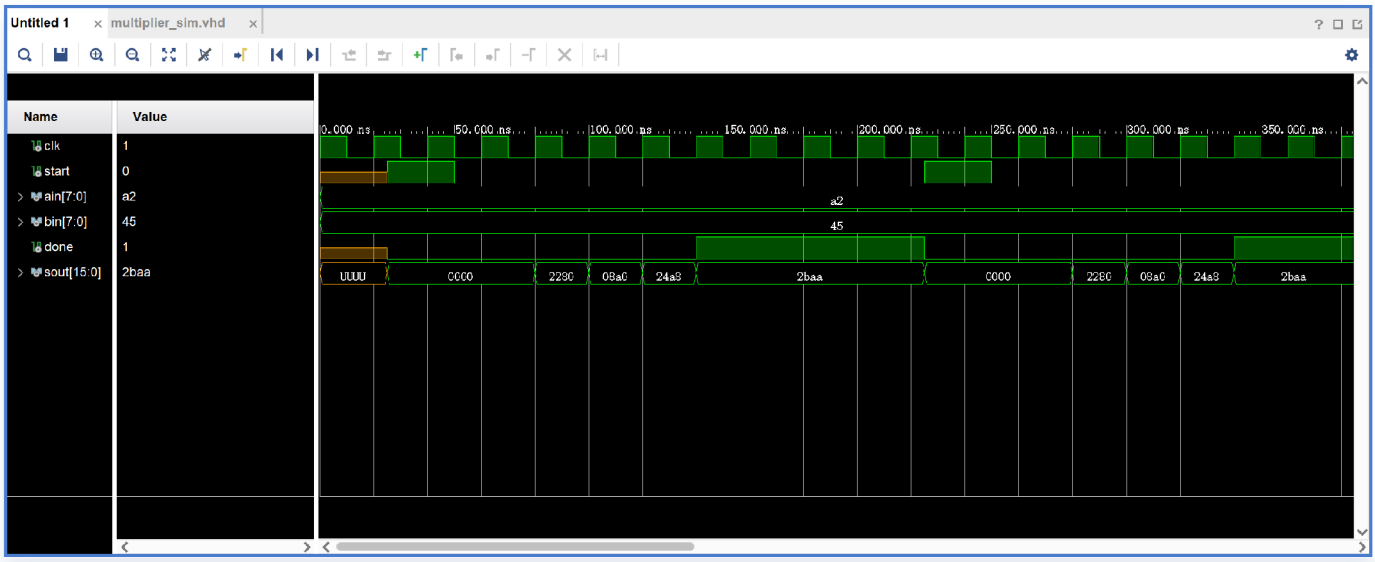
二、仿真配置

```
1   library IEEE;
2   use IEEE.STD_LOGIC_1164.ALL;
3
4   entity multiplier_sim is
5   --  Port ( );
6   end multiplier_sim;
7
8   architecture Behavioral of multiplier_sim is
9   component multiplier
10      Port(
11          clk,start:in std_logic;
12          ain,bin:in std_logic_vector(7 downto 0);
13          done:out std_logic;
14          sout:inout std_logic_vector(15 downto 0)
15      );
16  end component;
17
18  signal clk,start: std_logic;
19  signal ain,bin: std_logic_vector(7 downto 0);
20  signal done: std_logic := '0';
21  signal sout: std_logic_vector(15 downto 0) := (others => '0');
22  begin
23  multiplier_2bit_inst:multiplier port map(clk,start,ain,bin,done,sout);
24
25  clock_gen:process
26  begin
27      clk<='1';
28      wait for 10ns;
29      clk<='0';
30      wait for 10ns;
```

```
31  end process;
32
33  test:process
34  begin
35      ain<="10100010";
36      bin<="01000101";
37      wait for 25ns;
38      start<='1';
39      wait for 25ns;
40      start<='0';
41      wait for 150ns;
42  end process;
43
44  end Behavioral;
```

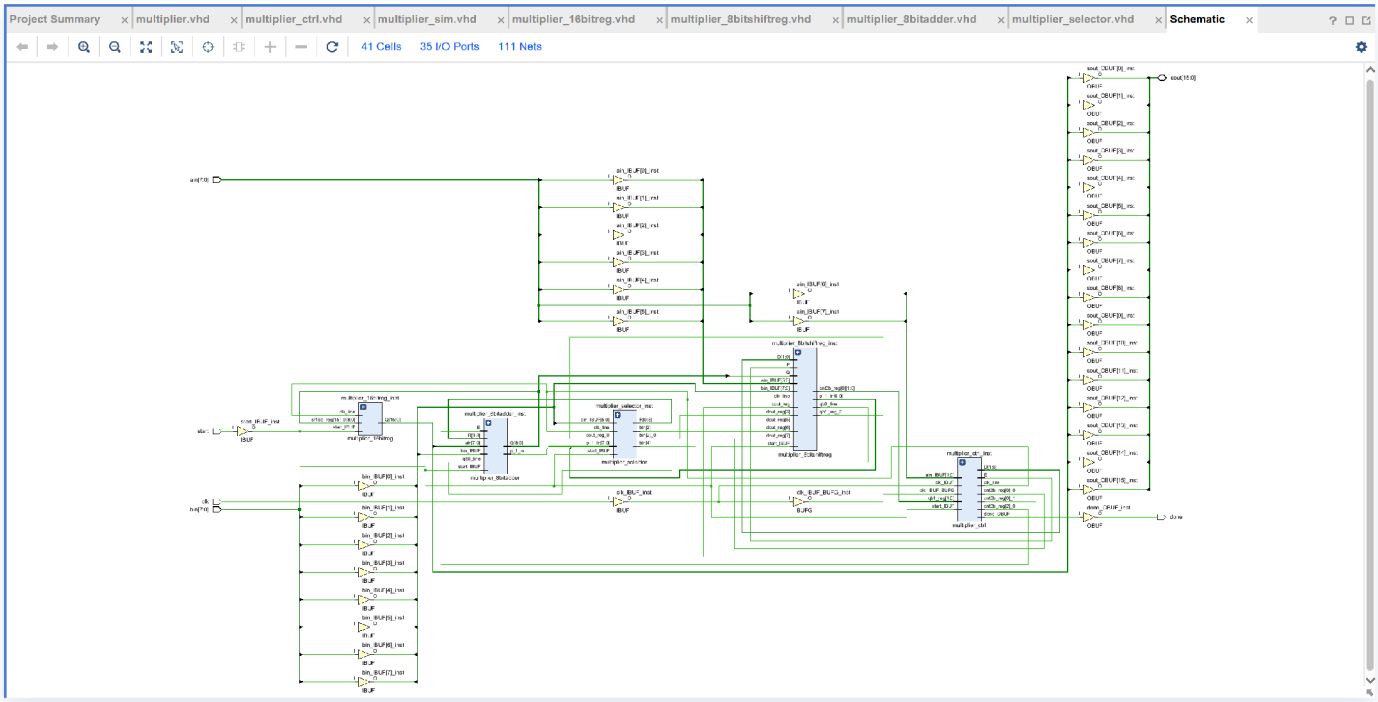
三、功能仿真结果与分析

3.1 仿真电路时序图



- 从仿真结果可以看出，乘法器输入两个 8位原码数：a = a2H = 1010 0010B, b = 45H = 0100 0101B;
- 计算完成后，done = 1, 输出计算结果 sout = 2baaH = 0010 1011 1010 1010B, 经进一步验证结果计算正确;

3.2 电路连接关系图


Project Summary
multiplier.vhd
multiplier_ctrl.vhd
multiplier_sim.vhd
multiplier_16bitreg.vhd
multiplier_8bitshiftreg.vhd
multiplier_8bitadder.vhd
multiplier_selector.vhd
Schematic
41 Cells
35 I/O Ports
111 Nets