EDA技术实用教程

第11章

系 统 仿 真

11.1 仿　真

仿真也称模拟（**Simulation**），是对电路设计的一种间接的检测方法，是利用计算机对整个硬件系统进行模拟检测，但却可以不接触具体的硬件系统。

11.2 VHDL源程序仿真

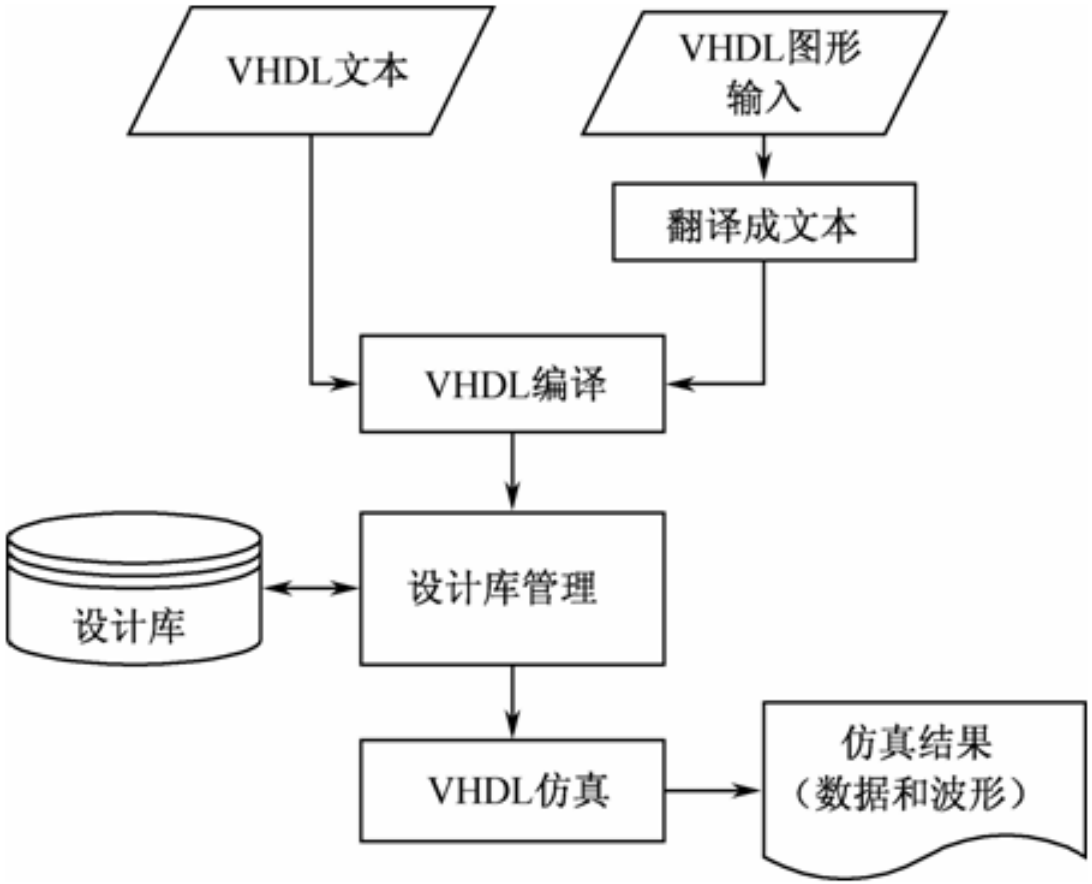

图 11-1　VHDL 仿真流程

11.2 VHDL源程序仿真

【例 11-1】

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY and1 IS
PORT(aaa,bbb : IN STD_LOGIC; ccc: OUT STD_LOGIC);
END and1;
ARCHITECTURE one OF and1 IS
BEGIN
ccc <= aaa AND bbb;
END;
```

11.2 VHDL源程序仿真

【例 11-2】

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
ENTITY TRIBUF_and1 IS
     GENERIC (  ttri: TIME := 1 ns;
                ttxz: TIME := 1 ns;
                ttzx: TIME := 1 ns);
     PORT (     in1 : IN std_logic;
                 oe  : IN std_logic;
                 y   : OUT std_logic);
END TRIBUF_and1;
ARCHITECTURE behavior OF TRIBUF_and1 IS
BEGIN
     PROCESS (in1, oe)
     BEGIN
         IF oe'EVENT THEN
             IF oe = '0' THEN
                 y <= TRANSPORT 'Z' AFTER ttxz;
             ELSIF oe = '1' THEN
```

接下页

11.2 VHDL源程序仿真

接上页

```
                y <= TRANSPORT in1 AFTER ttzx;
            END IF;
        ELSIF oe = '1' THEN
            y <= TRANSPORT in1 AFTER ttri;
        ELSIF oe = '0' THEN
            y <= TRANSPORT 'Z' AFTER ttxz;
        END IF;
    END PROCESS;
END behavior;
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
USE work.tribuf_and1;
ENTITY and1 IS
   PORT (  aaa,  bbb : IN std_logic;
                ccc : OUT std_logic);
END and1;
ARCHITECTURE FPGA1 OF and1 IS
  ......
END FPGA1;
```

11.3 仿真激励信号的产生

【例 11-3】

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY ADDER4 IS
    PORT ( a, b : IN INTEGER RANGE 0 TO 15;
                  c : OUT INTEGER RANGE 0 TO 15 );
END ADDER4;
ARCHITECTURE one OF ADDER4 IS
BEGIN
    c <= a + b;
END one;
```

11.3 仿真激励信号的产生

1. 第一种方法

【例 11-4】

```
ENTITY SIGGEN IS
    PORT ( sig1 : OUT INTEGER RANGE 0 TO 15;
           sig2 : OUT INTEGER RANGE 0 TO 15 );
END;
ARCHITECTURE Sim OF SIGGEN IS
BEGIN
    sig1 <= 10, 5 AFTER 200 ns, 8 AFTER 400 ns;
    sig2 <= 3, 4 AFTER 100 ns, 6 AFTER 300 ns;
END;
```

11.3 仿真激励信号的产生

1. 第一种方法

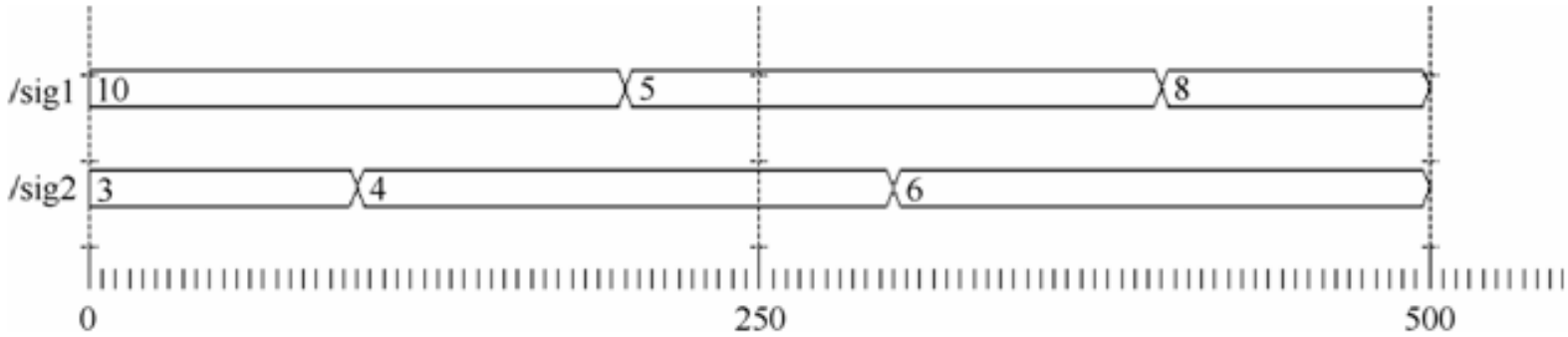

图 11-2　SIGGEN 的仿真输出波形

11.3 仿真激励信号的产生

1. 第一种方法 【例 11-5】

```
ENTITY BENCH IS
END;
ARCHITECTURE one OF BENCH IS
    COMPONENT ADDER4
        PORT ( a, b : integer range 0 to 15;
                 c : OUT INTEGER RANGE 0 TO 15 );
    END COMPONENT;
    COMPONENT SIGGEN
        PORT ( sig1 : OUT INTEGER RANGE 0 TO 15;
               sig2 : OUT INTEGER RANGE 0 TO 15 );
    END COMPONENT;
        SIGNAL a, b, c : INTEGER RANGE 0 TO 15;
BEGIN
    U1 : ADDER4 PORT MAP (a, b, c);
    U2 : SIGGEN PORT MAP (sig1=>a, sig2=>b);
END;
```

11.3 仿真激励信号的产生

1. 第一种方法

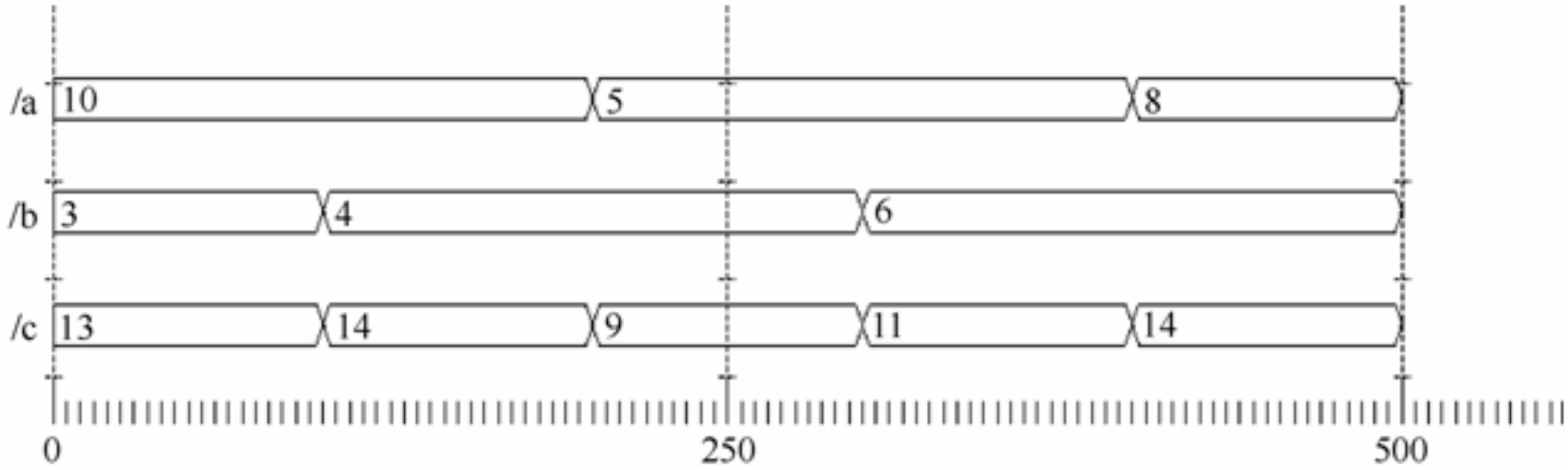

图 11-3　BENCH 仿真波形图

11.3 仿真激励信号的产生

2. 第二种方法

```
force <信号名> <值> [<时间>][, <值> <时间> …] [-repeat <周期>]
```

```
force a 0                    （强制信号的当前值为 0）
force b 0 0, 1 10            （强制信号 b 在时刻 0 的值为 0，在时刻 10 的值为 1）
force clk 0 0, 1 15 -repeat 20   （clk 为周期信号，周期为 20）
```

```
force a 10 0, 5 200, 8 400
force b 3 0, 4 100, 6 300
```

11.4 VHDL测试基准

【例 11-6】

```
Library IEEE;
use IEEE.std_logic_1164.all;
entity counter8 is
  port ( CLK, CE, LOAD, DIR, RESET: in STD_LOGIC;
         DIN: in INTEGER range 0 to 255;
         COUNT: out INTEGER range 0 to 255 );
end counter8;
architecture counter8_arch of counter8 is
begin
process (CLK, RESET)
   variable COUNTER: INTEGER range 0 to 255;
begin
```

接下页

接上页

```
    if RESET='1' then  COUNTER := 0;
    elsif CLK='1' and CLK'event then
       if LOAD='1' then  COUNTER := DIN;
       Else
          if CE='1' then
             if DIR='1' then
              if COUNTER =255 then COUNTER := 0;
                 Else   COUNTER := COUNTER + 1;   end if;
             else
              if COUNTER =0 then  COUNTER := 255;
                Else   COUNTER := COUNTER - 1;
              end if;
             end if;
          end if;
       end if;
    end if;
    COUNT <= COUNTER;
end process;
end counter8_arch;
```

【例 11-7】

```
Entity testbench is end testbench;
Architecture testbench_arch of testbench is
File RESULTS: TEXT open WRITE_MODE is "results.txt";
Component counter8
   port (   CLK , RESET: in STD_LOGIC;
         CE, LOAD, DIR: in STD_LOGIC;
     DIN: in INTEGER range 0 to 255;
     COUNT: out INTEGER range 0 to 255 );
 end component;
shared variable end_sim : BOOLEAN := false;
signal CLK, RESET, CE, LOAD, DIR: STD_LOGIC;
signal DIN: INTEGER range 0 to 255;
signal COUNT: INTEGER range 0 to 255;
procedure WRITE_RESULTS  (
       CLK, CE, LOAD, LOAD, RESET : STD_LOGIC;
                          DIN, COUNT : INTEGER  ) is
Variable V_OUT : LINE;
Begin
```

接上页

```
  write(V_OUT, now, right, 16, ps);                    -- 输入时间
  write(V_OUT, CLK, right, 2);
  write(V_OUT, RESET, right, 2);
  write(V_OUT, CE, right, 2);
  write(V_OUT, LOAD, right, 2);
  write(V_OUT, DIR, right, 2);
  write(V_OUT, DIN, right, 257);
  write(V_OUT, COUNT, right, 257);   --write outputs
  writeline(RESULTS,V_OUT);
  end WRITE_RESULTS;
  begin
  UUT: COUNTER8
  port map (CLK => CLK,RESET => RESET,
                CE => CE,   LOAD => LOAD,
               DIR => DIR, DIN => DIN,
                COUNT => COUNT );
  CLK_IN:  process
  Begin
      if end sim = false then  CLK <= '0';
```

接下页

接上页

```
            Wait for 15 ns;
            CLk <='1';
            Wait for 15 ns;
         Else
            Wait;
        end if;
end process;
STIMULUS: process
Begin
        RESET   <= '1';
        CE      <= '1';          -- 计数使能
        DIR     <= '1';          -- 加法计数
        DIN     <= 250;          -- 输入数据
        LOAD    <= '0';          -- 禁止加载输入的数据
         wait for 15 ns;
         RESET  <= '0';
         wait for 1 us;
         CE     <= '0';          -- 禁止计数脉冲信号进入
         wait for 200 ns;
         CE     <= '1';
```

接下页

接上页

```
        wait for 200 ns;
        DIR    <= '0';
        wait for 500 ns;
        LOAD   <= '1';
        wait for 60 ns;
        LOAD   <= '0';
        wait for 500 ns;
        DIN    <= 60;
        DIR    <= '1';
        LOAD   <= '1';
        wait for 60 ns;
        LOAD   <= '0';
        wait for 1 us;
        CE     <= '0';
        wait for 500 ns;
        CE     <= '1';
        wait for 500 ns;
        end_sim :=true;
    wait;
    end process;
    WRITE_TO_FILE: WRITE_RESULTS(CLK,RESET,CE,LOAD,DIR,DIN,COUNT);
    End testbench_arch;
```

11.4 VHDL测试基准

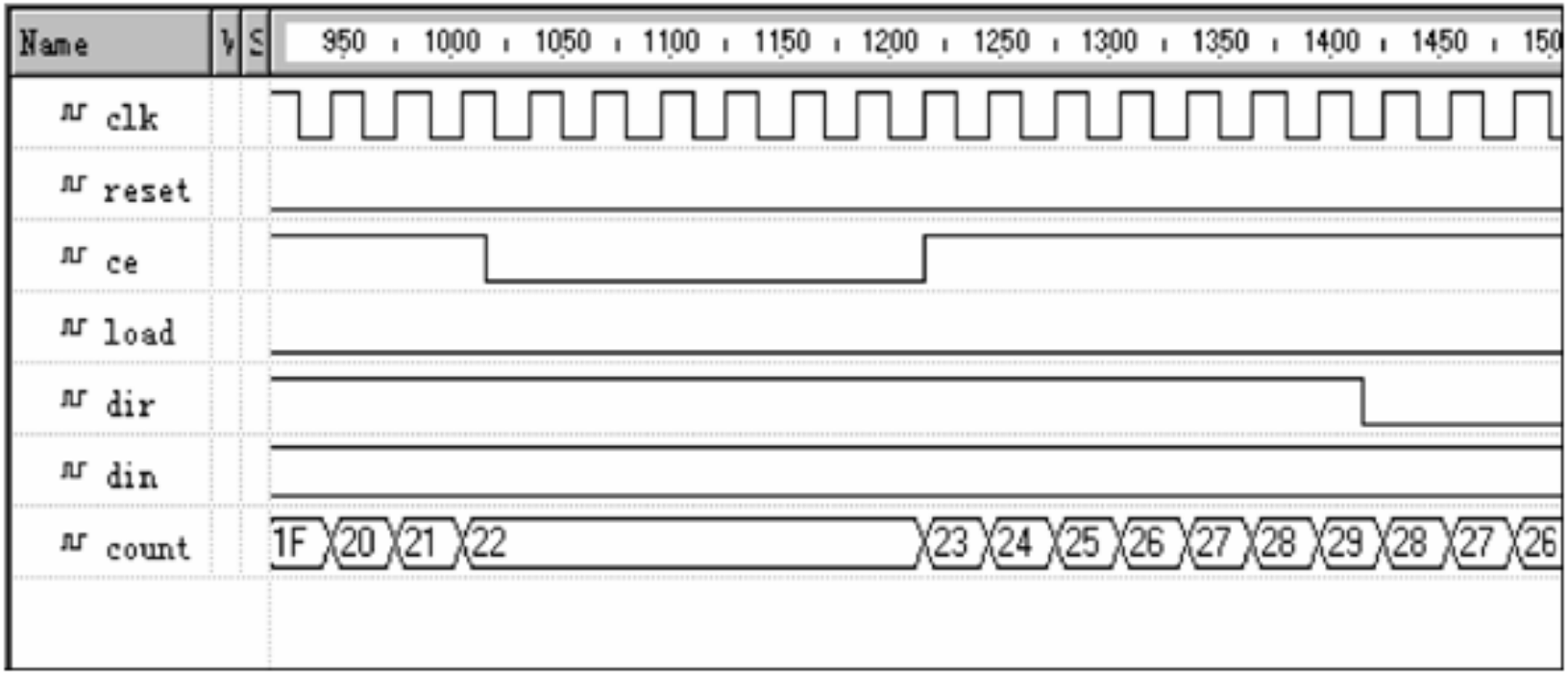

图 11-4　8 位计数器测试基准仿真部分波形图

11.5 VHDL系统级仿真

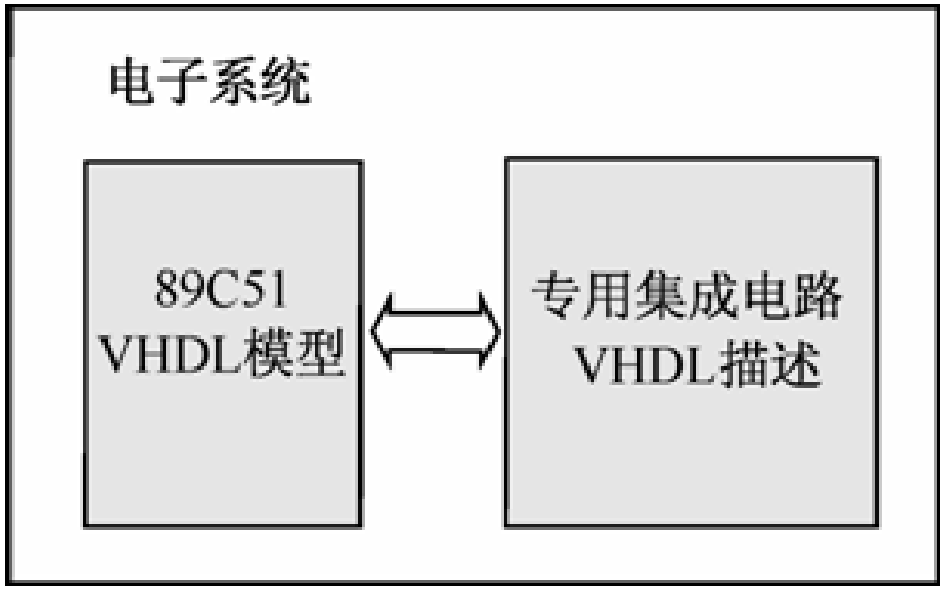

图 11-5 VHDL 系统仿真模型示意图

11.6 使用ModelSim进行仿真

【例 11-8】

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE  ieee.std_logic_unsigned.all;
ENTITY cnt4 IS
 PORT (      rst : IN STD_LOGIC;
      d : IN STD_LOGIC_VECTOR(3 downto 0);
      load : IN STD_LOGIC;
      clk,ce : IN STD_LOGIC;
      q : OUT STD_LOGIC_VECTOR(3 downto 0);
      cout : OUT STD_LOGIC    );
END cnt4;
ARCHITECTURE syn OF cnt4 IS
signal count : std_logic_vector(3 downto 0);
BEGIN
```

接下页

11.6 使用ModelSim进行仿真

接上页

```
cntproc: process(clk,rst) begin
if rst = '1' then  count <= (others => '0');
 elsif rising_edge(clk) then
     if ce = '1' then
         if load = '1' then  count <= d;
             else count <= count + 1;    end if;
     end if;
end if;
end process;
coutproc : process(clk,rst) begin
if rst = '1' then    cout <= '0';
elsif rising_edge(clk) then
     if count = "1111" then      cout <= '1';
         else    cout <= '0';    end if;
end if;
end process;
q <= count;
END syn;
```

11.6 使用ModelSim进行仿真

（1）启动ModelSim

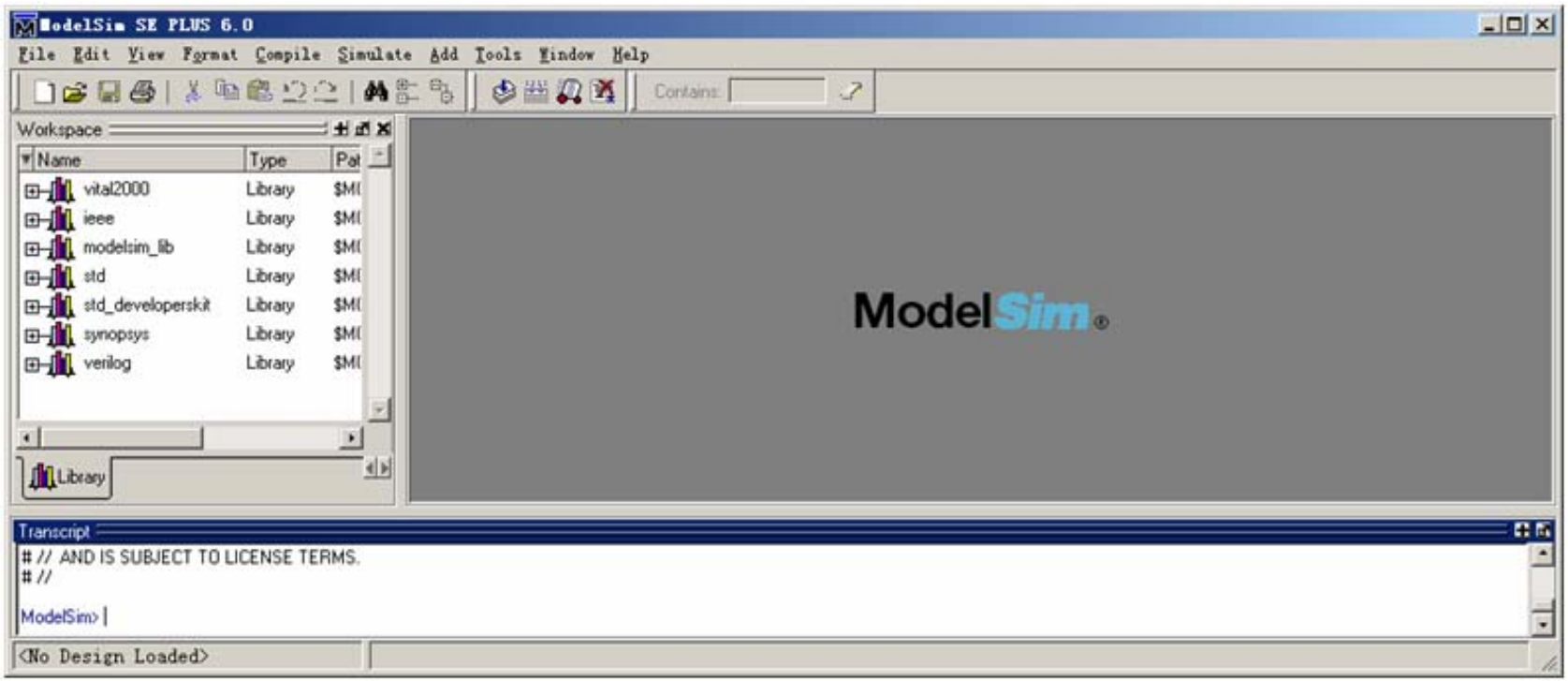

图 11-6 ModelSim 的启动界面

11.6 使用ModelSim进行仿真

（2）建立仿真工程项目

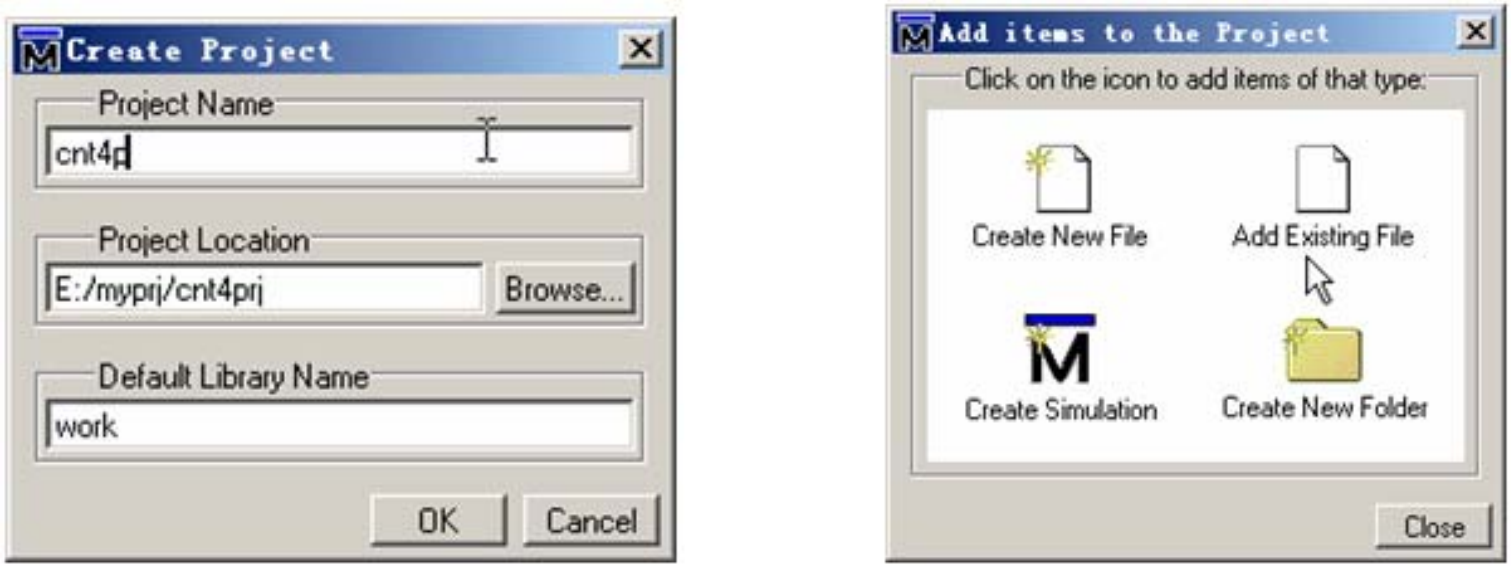

图 11-7　建立工程建立项目

11.6 使用ModelSim进行仿真

（3）编译仿真文件

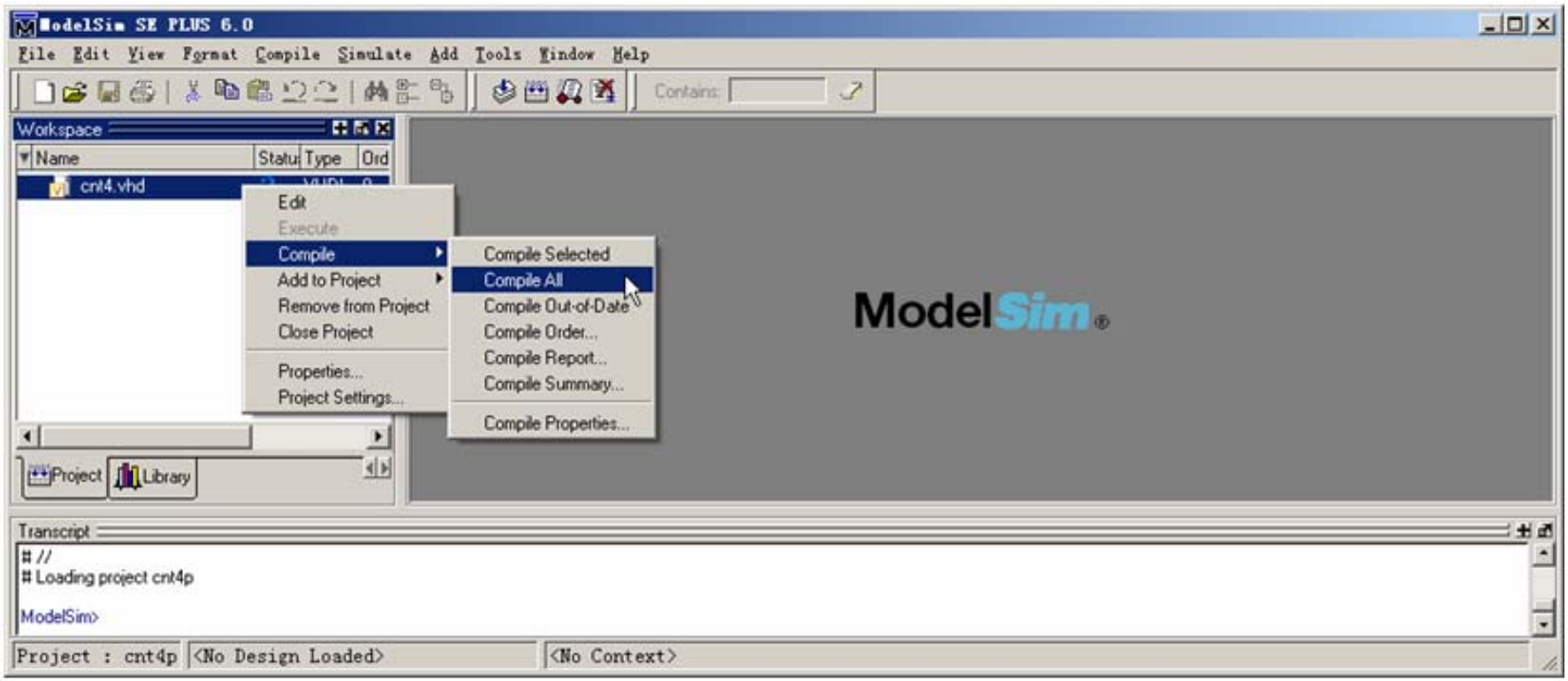

图 11-8　开始编译仿真文件

11.6 使用ModelSim进行仿真

（3）编译仿真文件

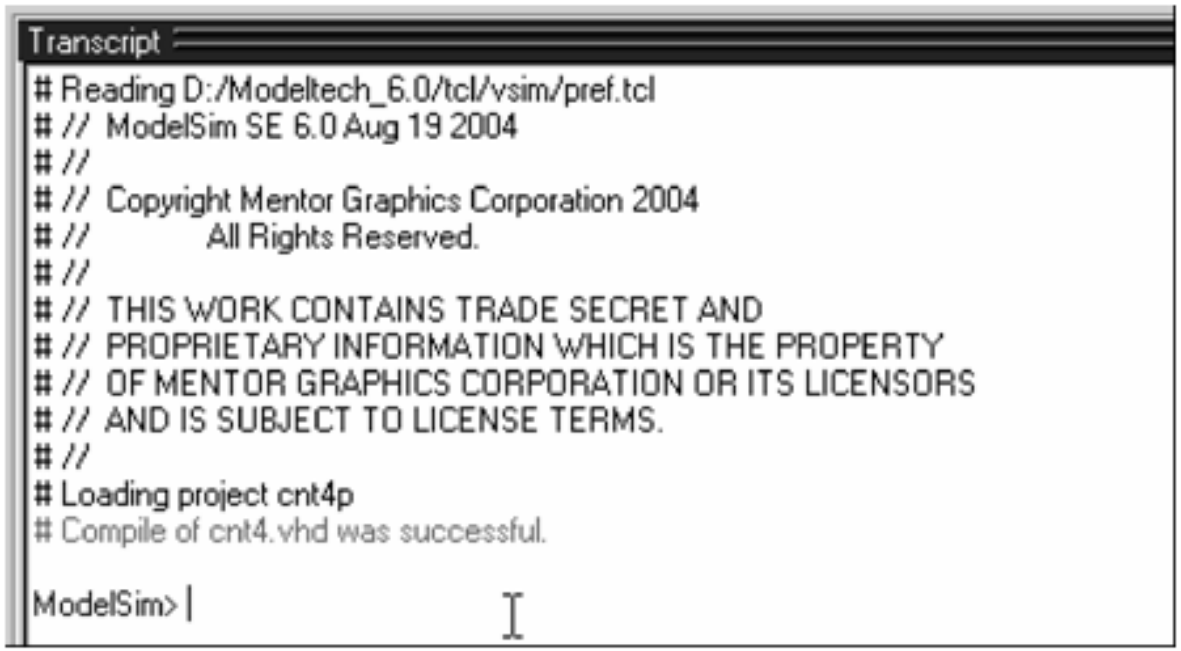

图 11-9 ModelSim 编译时的提示信息

11.6 使用ModelSim进行仿真

（4）装载仿真模块和仿真库

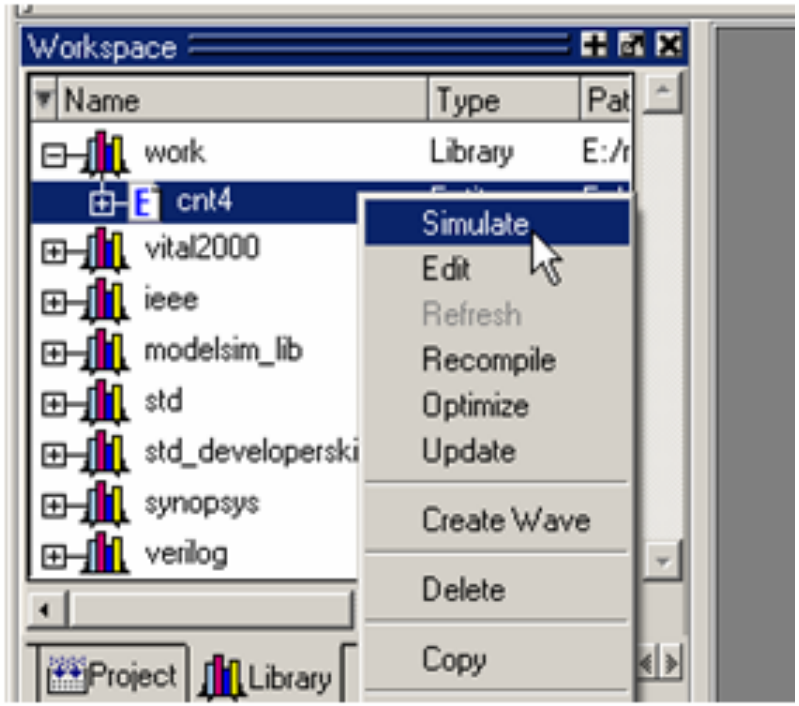

图 11-10　装载设计模块

（5）执行仿真

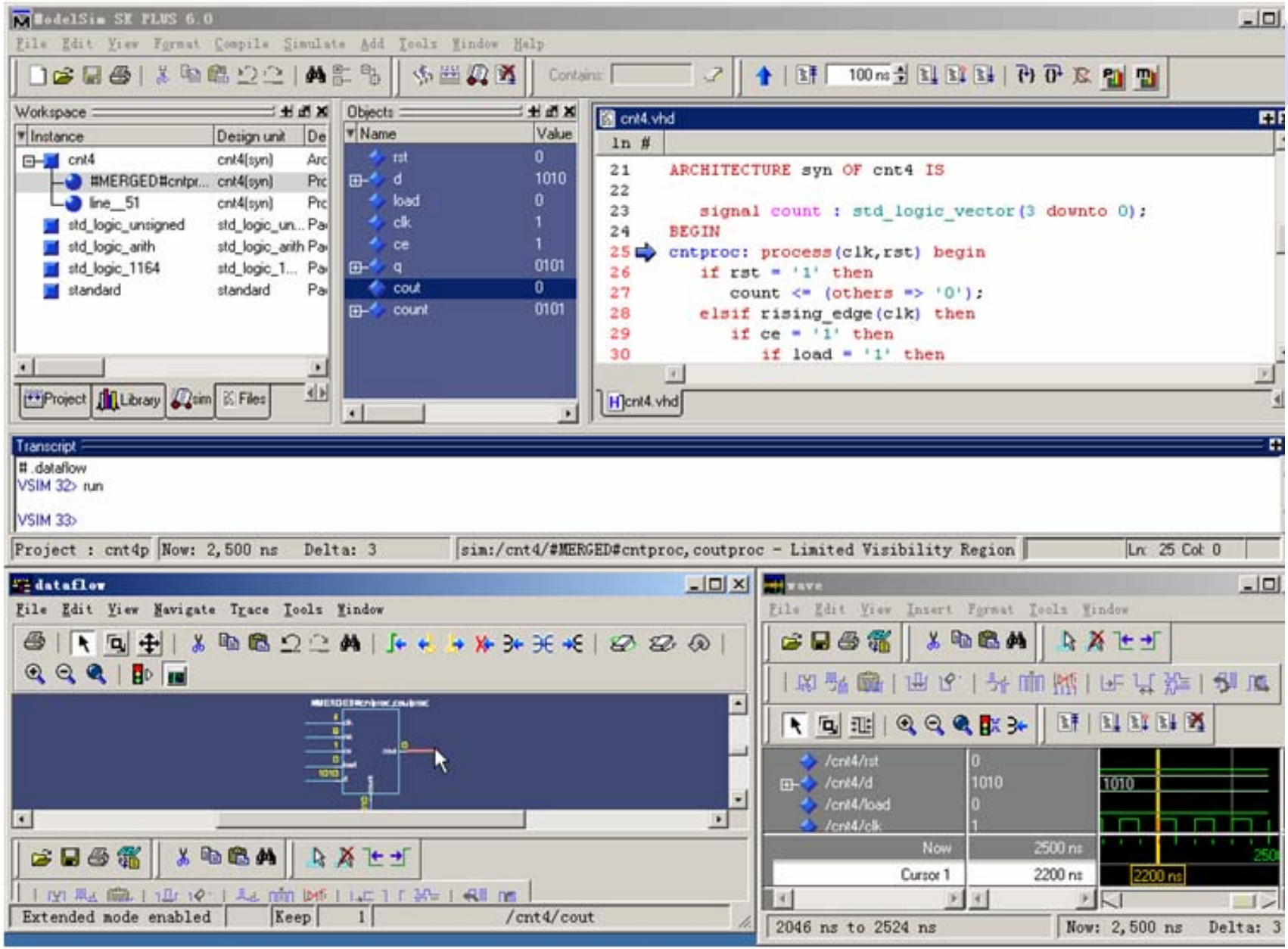

图 11-11　ModelSim 的仿真观察窗口

（5）执行仿真

【例 11-9】

```
library ieee;
use ieee.std_logic_1164.all;
ENTITY wavegen IS
PORT( clk,rst : OUT STD_LOGIC    );
end wavegen;
ARCHITECTURE sim OF wavegen is
constant cycle:Time := 10 ns;
BEGIN
process  begin
     clk <= '0';
     wait for cycle/2;
     clk <= '1';
     wait for cycle/2;
end process;
process  begin
     rst <= '1';
     wait for cycle*5;
     rst <= '0';
     wait;
end process;
END sim;
```

11.6 使用ModelSim进行仿真

（5）执行仿真

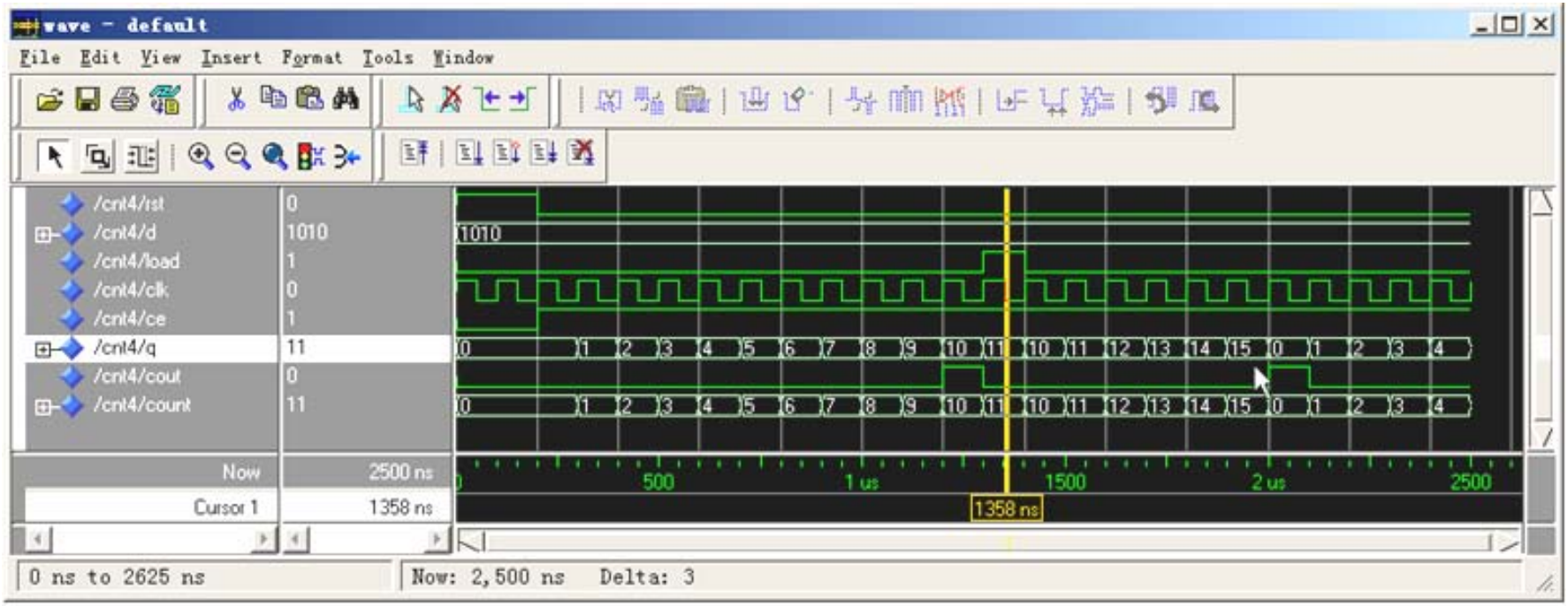

图 11-12 ModelSim 的波形观察窗口

11.6 使用ModelSim进行仿真

（5）执行仿真

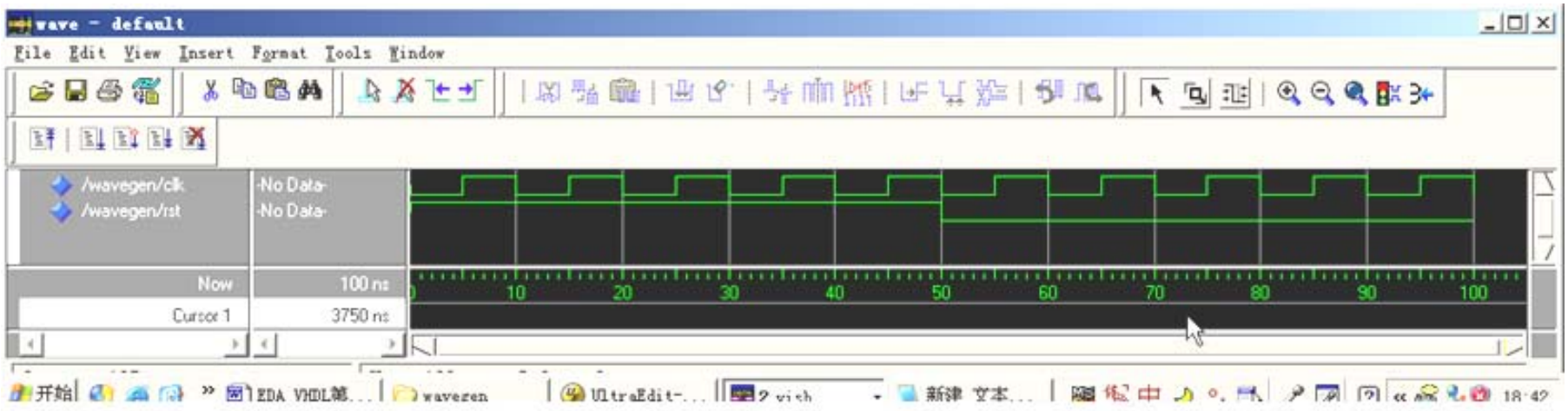

图 11-13　时钟与复位信号生成