**cout60\_v.vhd**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

entity cout60\_v is

port(CLRN, LDN, EN, CLK: in std\_logic;

Da: in std\_logic\_vector(3 downto 0);

Db: in std\_logic\_vector(2 downto 0);

Qa: out std\_logic\_vector(3 downto 0);

Qb: out std\_logic\_vector(2 downto 0);

RCO: out std\_logic);

end cout60\_v;

architecture a of cout60\_v is

begin

process(CLK)

variable tmpa: std\_logic\_vector(3 downto 0);

variable tmpb: std\_logic\_vector(2 downto 0);

begin

if CLRN = '0' then tmpb := "000"; tmpa := "0000";

else if (CLK'event and CLK = '1') then

if LDN = '0' then tmpa := Da; tmpb := Db;

elsif EN = '1' then

if tmpa = "1001" then tmpa := "0000";

if tmpb = "101" then tmpb := "000";

else tmpb := tmpb + 1;

end if;

else tmpa := tmpa + 1;

end if;

end if;

end if;

end if;

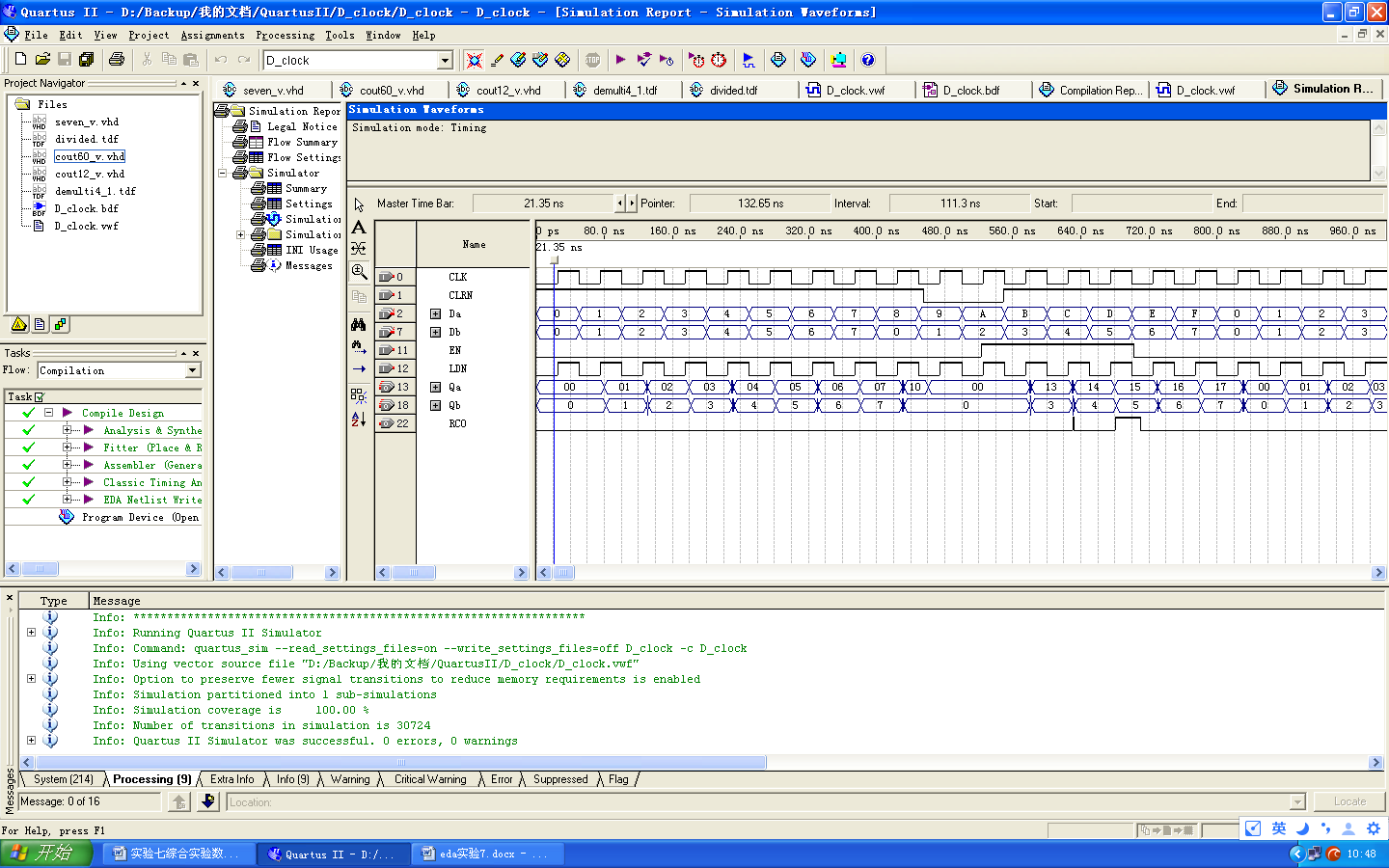
Qa <= tmpa;

Qb <= tmpb;

RCO <= tmpb(0) and tmpb(2) and tmpa(0) and tmpa(3) and EN;

end process;

end a;



**cout12\_v.vhd**

library ieee;

use ieee.std\_logic\_1164.all;

entity cout12\_v is

port(CLRN, LDN, EN, CLK: in std\_logic;

Da: in integer range 0 to 9;

Db: in integer range 0 to 1;

Qa: out integer range 0 to 9;

Qb: out integer range 0 to 1);

end cout12\_v;

architecture a of cout12\_v is

begin

process(CLK)

variable tmpa: integer range 0 to 9;

variable tmpb: integer range 0 to 1;

begin

if CLRN = '0' then tmpb := 0; tmpa := 0;

else

if(CLK'event and CLK = '1') then

if LDN = '0' then

tmpa := Da; tmpb := Db;

elsif EN = '1' then

if tmpa = 9 then

tmpa := 0; tmpb := tmpb + 1;

elsif(tmpb = 1 and tmpa = 1)then

tmpb := 0; tmpa := 0;

else tmpa := tmpa + 1;

end if;

end if;

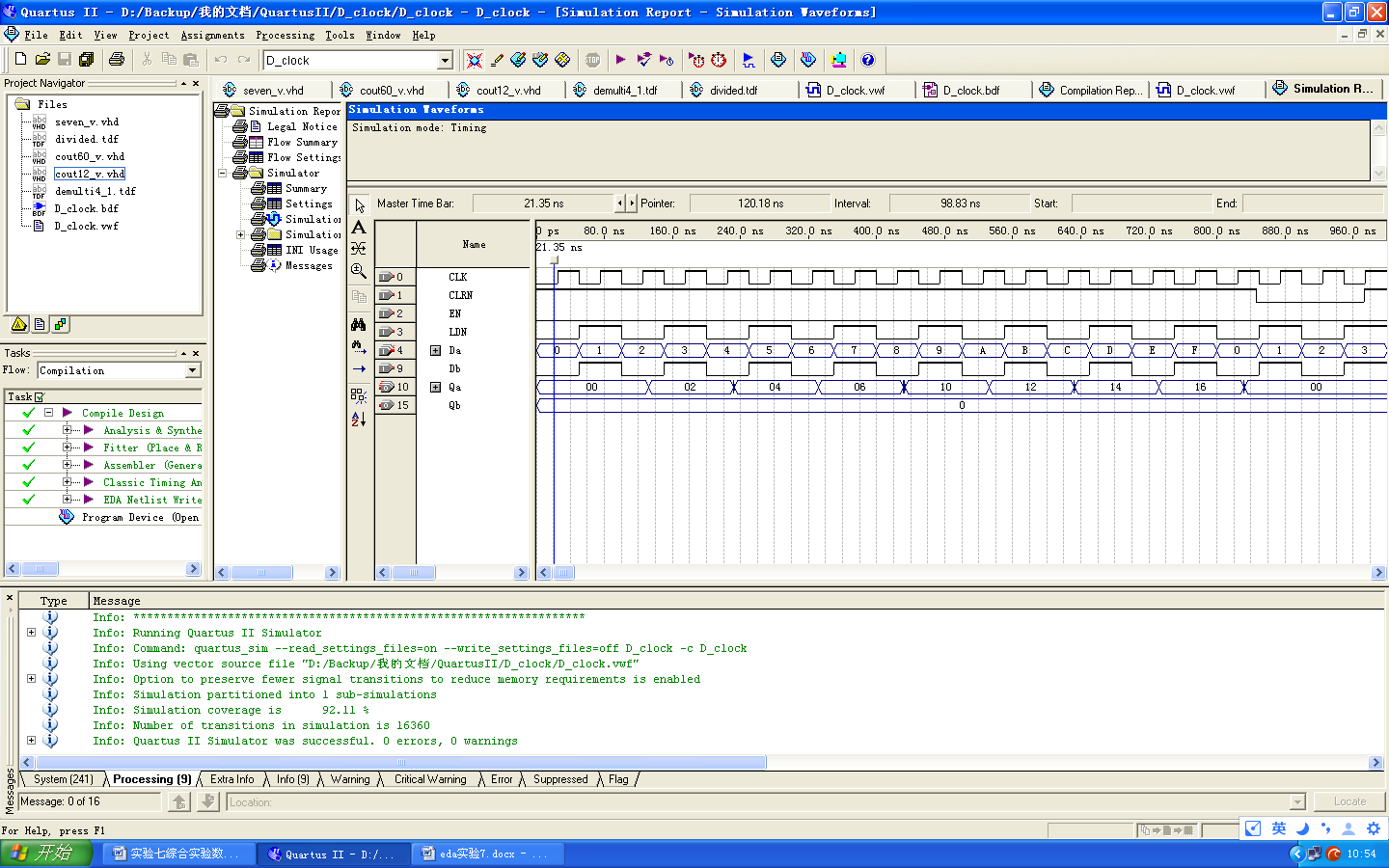
end if;

end if;

Qa <= tmpa; Qb <= tmpb;

end process;

end a;



**seven\_v.vhd**

library ieee;

use ieee.std\_logic\_1164.all;

entity seven\_v is

port(D: in integer range 0 to 9;

S: out std\_logic\_vector(6 downto 0));

end seven\_v;

architecture a of seven\_v is

begin

process(D)

begin

case D is

when 0 => S <= "1111110";

when 1 => S <= "0000110";

when 2 => S <= "1101101";

when 3 => S <= "1111001";

when 4 => S <= "0110011";

when 5 => S <= "1011011";

when 6 => S <= "1011111";

when 7 => S <= "1110000";

when 8 => S <= "1111111";

when 9 => S <= "1111011";

when others => S <= "0000000";

end case;

end process;

end a;

**demulti4\_1.tdf**

subdesign demulti4\_1

(

S[2..0], I[3..0]: input;

SA[3..0], SB[2..0], MA[3..0], MB[2..0], HA[3..0], HB: output;

)

begin

case S[] is

when 0 =>

SA[3..0] = I[3..0];

when 1 =>

SB[2..0] = I[2..0];

when 2 =>

MA[3..0] = I[3..0];

when 3 =>

MB[2..0] = I[2..0];

when 4 =>

MA[3..0] = I[3..0];

when 5 =>

HB = I[0];

when others =>

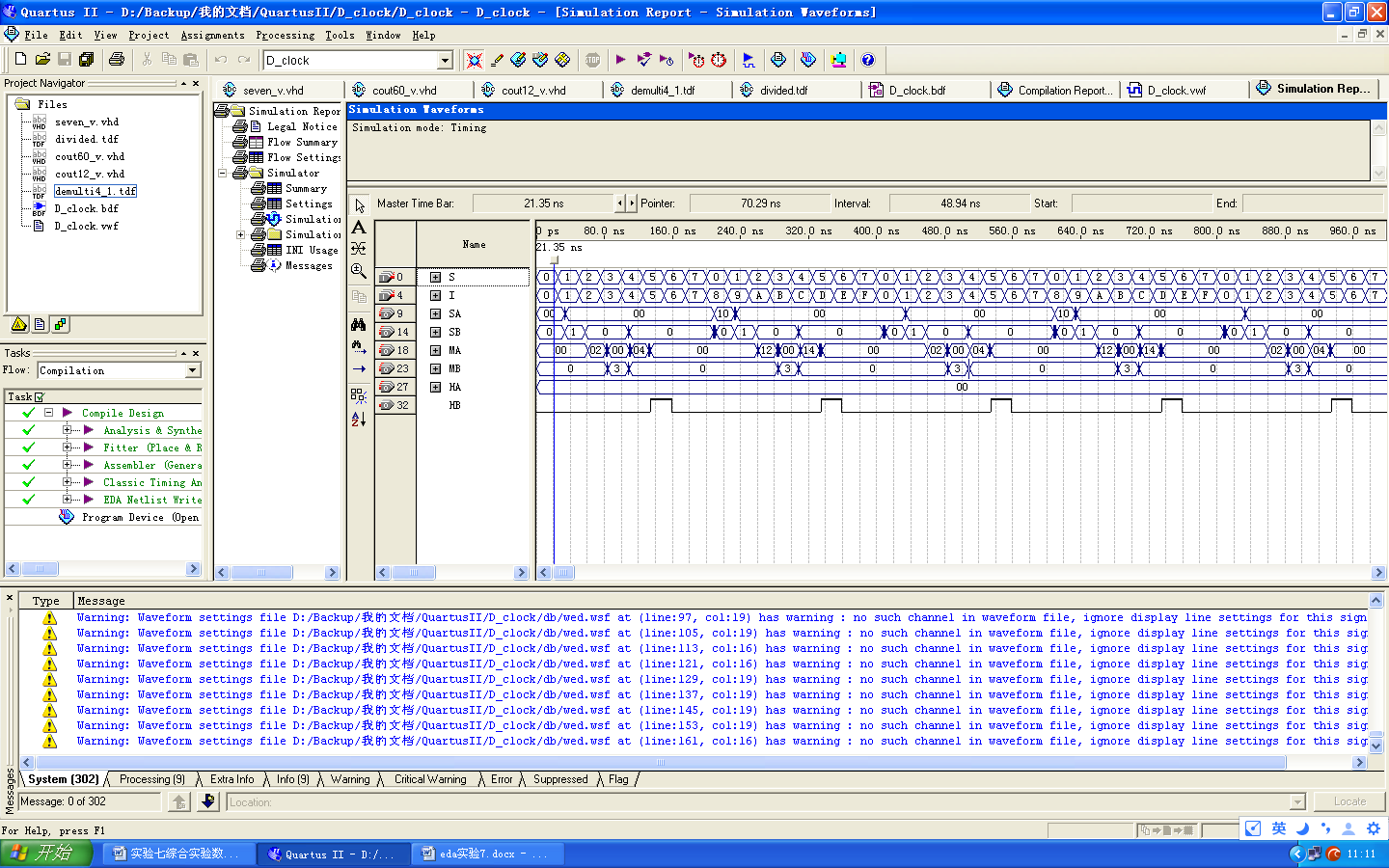
SA[3..0] = 0; SB[2..0] = 0;

MA[3..0] = 0; MB[2..0] = 0;

HA[3..0] = 0; HB = gnd;

end case;

end;



**divided.tdf**

subdesign divided

(

CLKI: input;

CLKO: output;

)

variable CNT[24..0]: DFF;

begin

CNT[].CLK = CLKI;

if CNT[] == 19999999 then

CNT[] = 0;

else

CNT[] = CNT[] + 1;

end if;

CLKO = CNT[24];

end;

**D\_clock.bdf**

