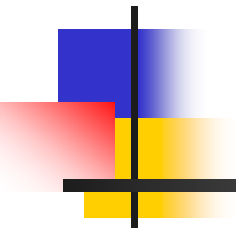

第9章

VHDL设计优化

9.1 资源优化

9.1.1 资源共享

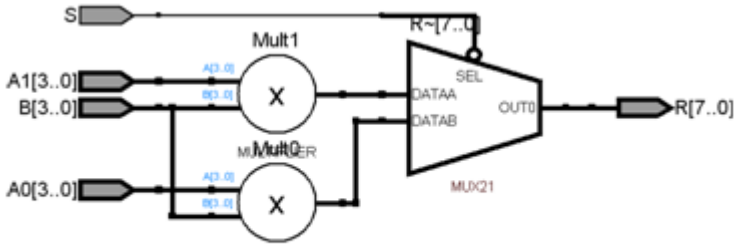

图 9-1 先乘后选择的设计方法 RTL 结构

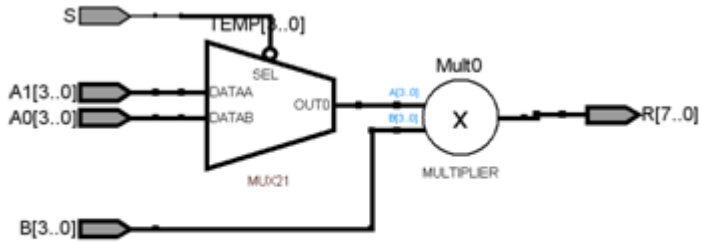

图 9-2 先选择后乘设计方法 RTL 结构

9.1 资源优化

9.1.1 资源共享

【例 9-1】

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;
USE ieee.std_logic_arith.all;
ENTITY multmux IS
    PORT (A0, A1, B  : IN  std_logic_vector(3 downto 0);
                  s  : IN  std_logic;
                  R  : OUT std_logic_vector(7 downto 0));
END multmux;
ARCHITECTURE rtl OF multmux IS
BEGIN
    process(A0,A1,B,s)    begin
        if(s='0') then   R<=A0 * B;  else  R<=A1 * B;  end if;
    end process;
END rtl;
```

9.1 资源优化

9.1.1 资源共享

【例 9-2】

```
ARCHITECTURE rtl OF muxmult IS  --以上部分与例 9-1 相同
   signal temp : std_logic_vector(3 downto 0);
BEGIN
   process(A0,A1,B,s)    begin
      if(s='0') then    temp<=A0; else  temp<=A1;   end if;
      R <= temp * B;
   end process;
END rtl;
```

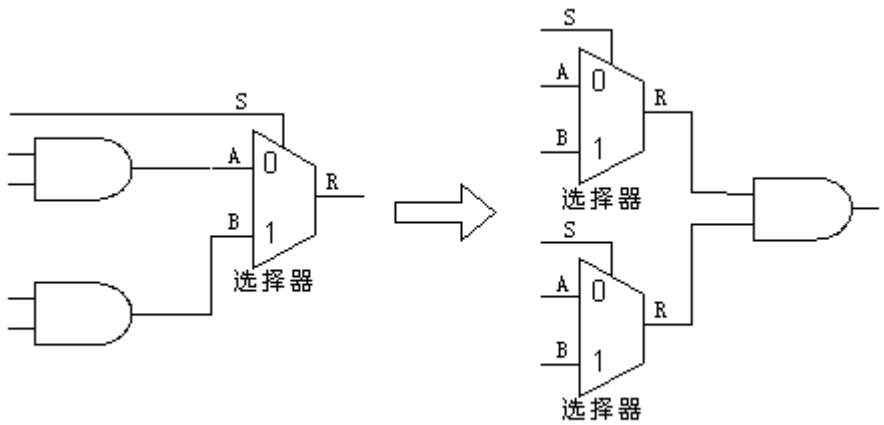

图 9-3　资源共享反例

9.1 资源优化

9.1.2 逻辑优化

【例 9-3】

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
ENTITY mult1 IS
    PORT(clk : in std_logic;
         ma : In std_logic_vector(11 downto 0);
         mc : out std_logic_vector(23 downto 0));
END mult1;
ARCHITECTURE rtl OF mult1 IS
   signal ta, tb : std_logic_vector(11 downto 0);
BEGIN
process(clk)  begin
   if(clk'event and clk='1') then
      ta<=ma;  tb<="100110111001";   mc<=ta * tb;   end if;
end process;
END rtl;
```

9.1 资源优化

9.1.2 逻辑优化

【例 9-4】

```
... --以上同例 9-3
ARCHITECTURE rtl OF mult1 IS
     signal ta : std logic vector(11 downto 0);
   constant tb : std logic vector(11 downto 0):="100110111001";
BEGIN
process(clk) begin
   if(clk'event and clk='1') then  ta<=ma; mc<=ta*tb;  end if;
end process;
END rtl;
```

9.1 资源优化

9.1.3 串行化

【例 9-5】

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
ENTITY pmultadd IS
    PORT(a0,a1,a2,a3 : in std_logic_vector(7 downto 0);
         b0,b1,b2,b3 : in std_logic_vector(7 downto 0);
         yout : out std_logic_vector(15 downto 0);
          clk : in std_logic);
END pmultadd;
ARCHITECTURE p_arch OF pmultadd IS
BEGIN
process(clk) begin
    if(clk'event and clk = '1') then
       yout <= ((a0*b0)+(a1*b1))+((a2*b2)+(a3*b3));  end if;
end process;
END p_arch;
```

【例 9-6】

```
...--以上部分与例 9-5 相同
          clk,  start : in std_logic);
END smultadd;
ARCHITECTURE s_arch OF smultadd IS
   signal cnt : std_logic_vector(2 downto 0);
   signal tmpa, tmpb : std_logic_vector(7 downto 0);
   signal  tmp,  ytmp : std_logic_vector(15 downto 0);
  BEGIN
tmpa <= a0 when cnt = 0 else
        a1 when cnt = 1 else
        a2 when cnt = 2 else
        a3 when cnt = 3 else    a0;
tmpb <= b0 when cnt = 0 else
        b1 when cnt = 1 else
        b2 when cnt = 2 else
        b3 when cnt = 3 else    b0;
tmp <= tmpa * tmpb;
process(clk) begin
   if(clk'event and clk = '1') then
      if (start='1') then  cnt<="000"; ytmp<=(others=>'0');
      elsif (cnt<4) then  cnt<=cnt+1;  ytmp<=ytmp+tmp;
      elsif (cnt=4) then  yout<=ytmp;
      end if;   end if;
end process;
END s_arch;
```

9.2 速度优化

9.2.1 流水线设计

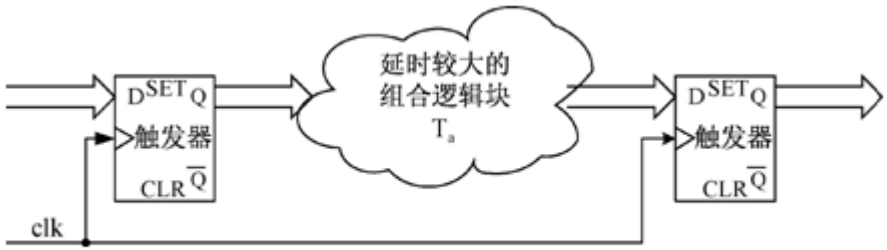

图 9-4 未使用流水线

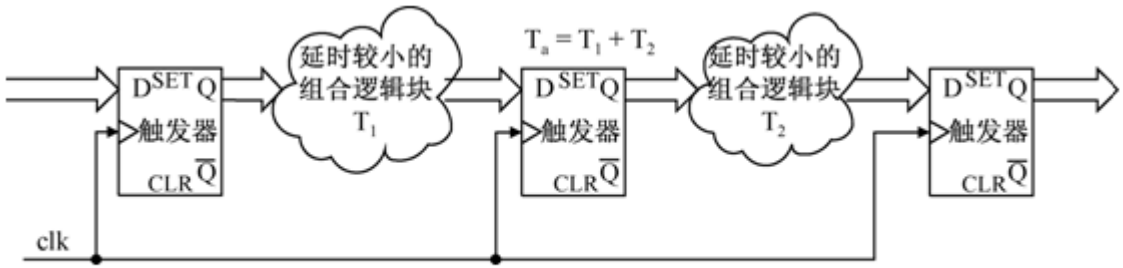

图 9-5 使用流水线结构

9.2 速度优化

9.2.1 流水线设计

$$F_{max} \approx F_{max1} \approx F_{max2} \approx 1 / T_1$$

图 9-6 流水线工作图示

9.2 速度优化

9.2.1 流水线设计

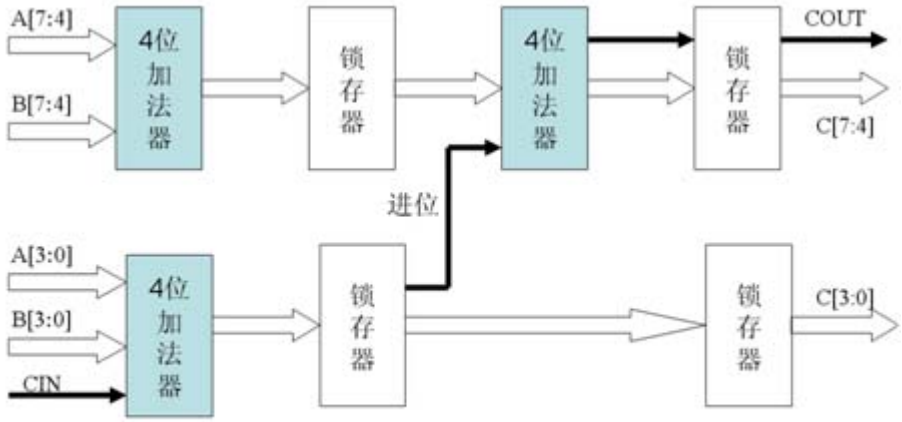

图 9-7　8 位加法器流水线工作图示

【例 9-7】普通加法器，EP3C10 FPGA 综合结果：LCs=10,REG=0,T=7.748ns.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;
USE ieee.std_logic_arith.all;
ENTITY ADDER8 IS
    PORT (A, B : IN std_logic_vector(7 downto 0);
          CLK,CIN  : IN  std_logic;       COUT  : OUT  std_logic;
             SUM  : OUT std_logic_vector(7 downto 0));
END ADDER8 ;
ARCHITECTURE rtl OF ADDER8  IS
SIGNAL SUMC,A0,B0 : std_logic_vector(8 downto 0);
BEGIN
     A0<='0'& A ; B0<='0' & B ;
  process(CLK)   begin
     IF (RISING_EDGE(CLK)) THEN   SUMC <= A0+B0+CIN;  END IF;
  end process;
     COUT<=SUMC(8) ; SUM<=SUMC(7 downto 0);
END rtl;
```

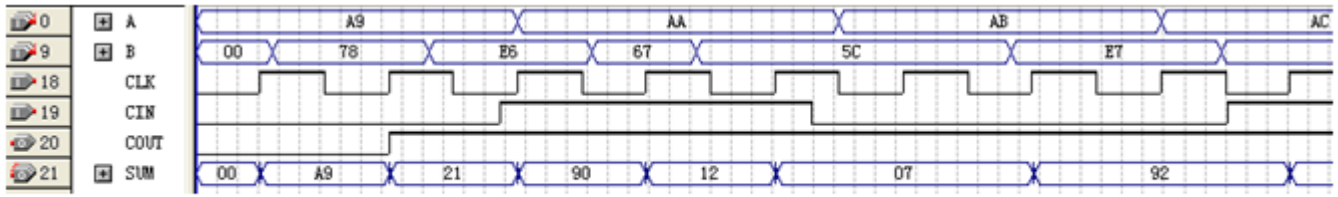

图 9-8　例 9-7 的时序仿真波形

【例 9-8】流水线加法器，EP3C10 综合结果：CLK=275MHz,T=3.63ns，LCs=24，REG=22。

```
...--以上部分与例 9-7 相同
ARCHITECTURE rtl OF ADDER8  IS
SIGNAL SUMC,A9,B9 : std_logic_vector(8 downto 0);
SIGNAL AB5,A5,B5,TA,TB,S : std_logic_vector(4 downto 0);
BEGIN
     A5<='0'& A(3 downto 0); B5<='0'& B(3 downto 0);
 process(CLK)   begin
     IF (RISING_EDGE(CLK)) THEN
       AB5<=A5+B5+CIN;  SUM(3 downto 0)<=AB5(3 downto 0); END IF;
   end process;
 process(CLK)   begin
     IF (RISING_EDGE(CLK)) THEN
     S<=('0'&A(7 downto 4))+('0'&B(7 downto 4))+ AB5(4); END IF;
   end process;
     COUT<=S(4) ; SUM(7 downto 4)<=S(3 downto 0);
END rtl;
```

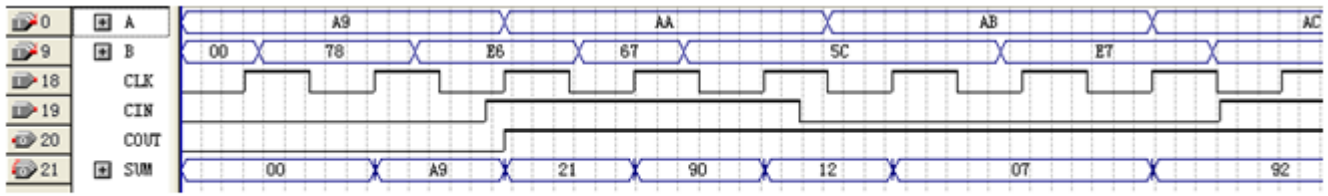

图 9-9　例 9-8 的时序仿真波形

9.2 速 度 优 化

9.2.2 寄存器配平

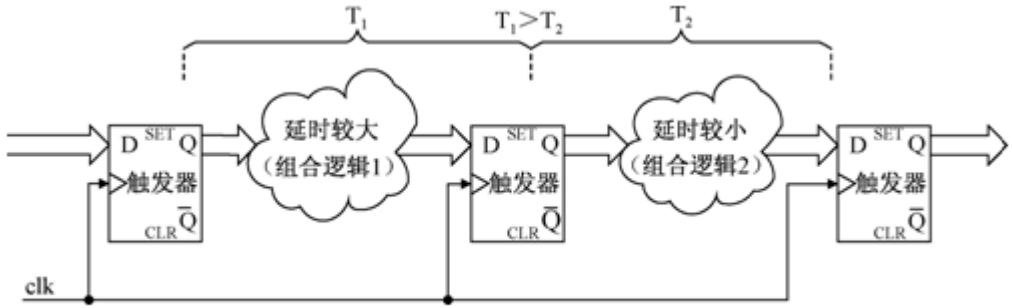

图 9-10 不合理的电路结构

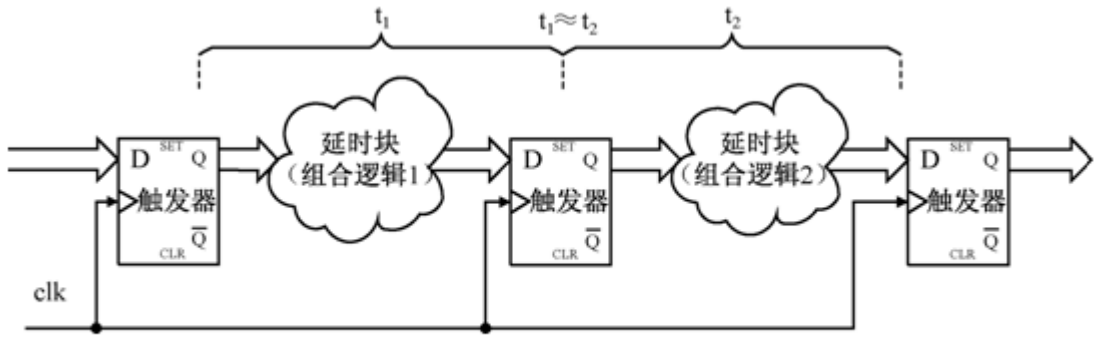

图 9-11 寄存器配平后的结构

9.2 速度优化

9.2.3 关键路径法

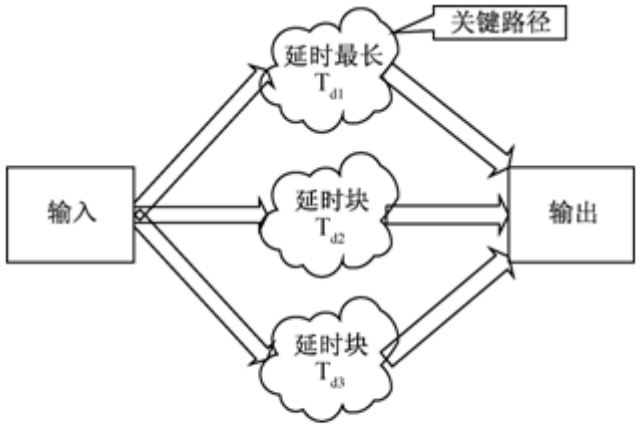

图 9-12 关键路径示意

9.2 速度优化

9.2.4 乒乓操作法

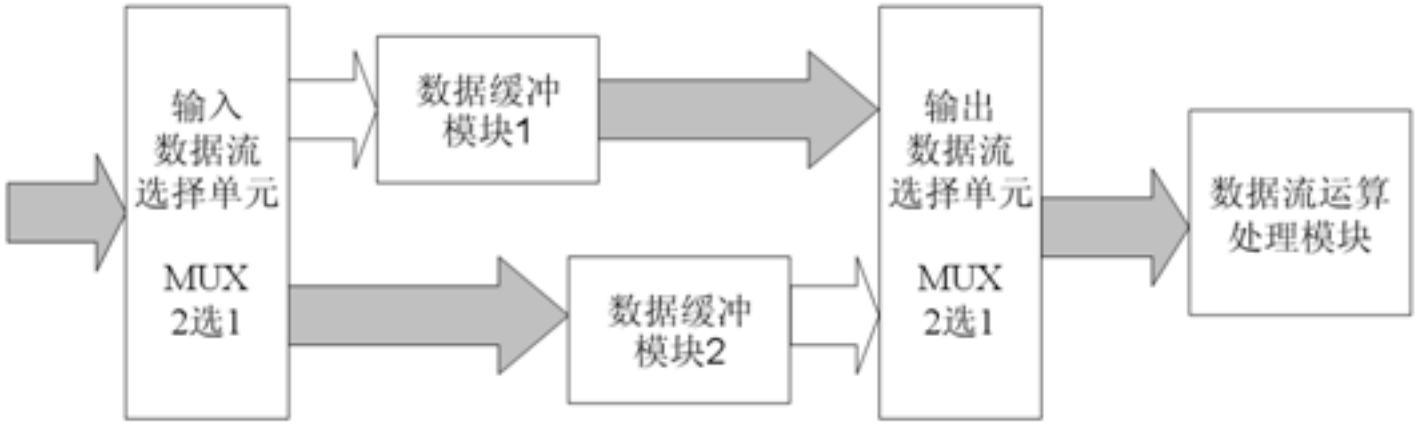

图 9-13　乒乓操作数据缓存结构示意图

9.2.5 加法树法

习　　题

【例 9-9】

```
LIBRARY ieee;
USE ieee.std logic 1164.all;
USE ieee.std logic unsigned.all;
USE ieee.std logic arith.all;
ENTITY addmux IS
   PORT (R : OUT std_logic vector(7 downto 0);  sel : IN  std logic;
         A,B,C,D  : IN  std logic vector(7 downto 0) );
END addmux;
ARCHITECTURE rtl OF addmux IS
BEGIN
    process(A,B,C,D,sel)    begin
       if(sel='0') then  R<=A+B;  else   R<=C+D;  end if;
    end process;
END rtl;
```

习　　题

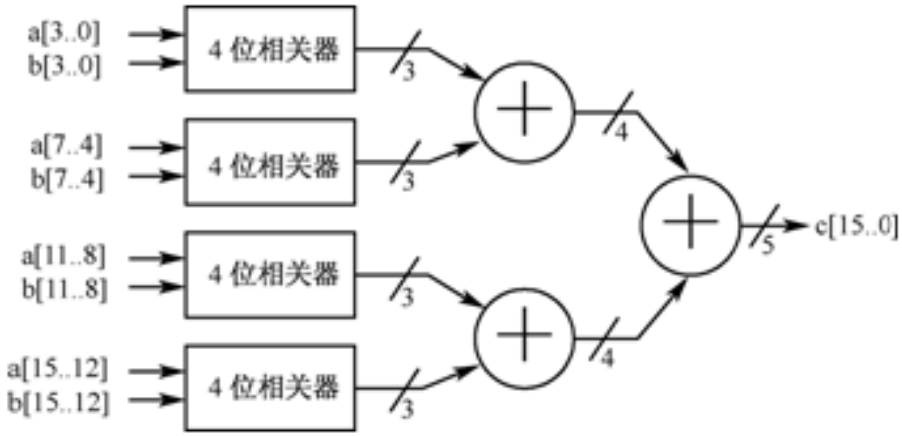

图 9-14　习题 9-9 图

实验与设计

9-1 采用流水线技术设计高速数字相关器

【例 9-10】

```
stemp <= a XOR b;
PROCESS(stemp)   BEGIN
 CASE stemp IS
 WHEN "0000" => c <= "100";                                    --4
 WHEN "0001"|"0010"|"0100"|"1000" => c <= "011";               --3
 WHEN "0011"|"0101"|"1001"|"0110"|"1010"|"1100" =>c<= "010"; --2
 WHEN "0111"|"1011"|"1101"|"1110" => c <= "001";    --1
 WHEN "1111" => c <= "000";  -- 0;
 WHEN OTHERS => c <= "000";
 END CASE;
END PROCESS;
```

实验与设计

9-2 线性反馈移位寄存器设计

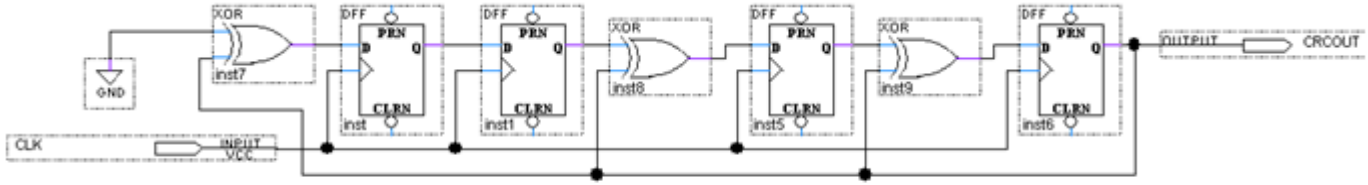

图 9-15 LFSR 举例

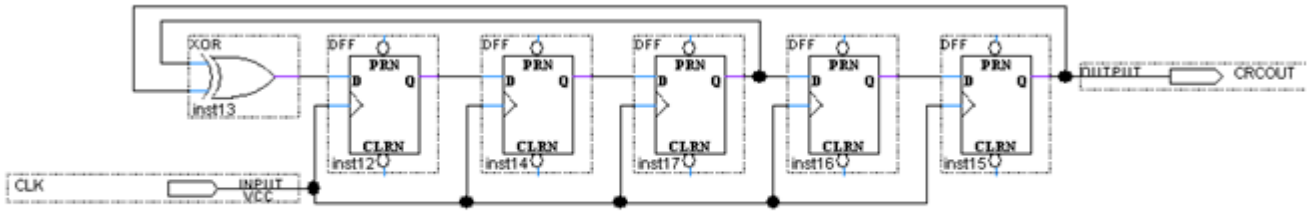

图 9-16 另一种 LFSR 结构

实验与设计

9-3 循环冗余校验（CRC）模块设计

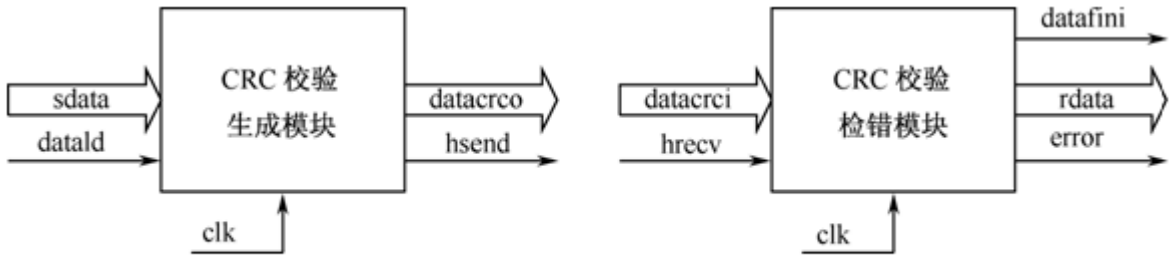

图 9-17　CRC 模块

【例 9-11】

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.ALL;
USE ieee.std_logic_arith.ALL;
ENTITY crcm IS
    PORT (clk, hrecv, datald : IN std_logic;
          sdata     : IN std_logic_vector(11 DOWNTO 0);
          datacrco  : OUT std_logic_vector(16 DOWNTO 0);
          datacrci  : IN std_logic_vector(16 DOWNTO 0);
            rdata   : OUT std_logic_vector(11 DOWNTO 0);
          datafini  : OUT std_logic;
          ERROR0, hsend  : OUT std_logic);
END crcm;
ARCHITECTURE comm OF crcm IS
    CONSTANT multi_coef : std_logic_vector(5 DOWNTO 0) := "110101";
           -- 多项式系数, MSB一定为'1'
    SIGNAL  cnt,rcnt : std_logic_vector(4 DOWNTO 0);
    SIGNAL  dtemp,sdatam,rdtemp   : std_logic_vector(11 DOWNTO 0);
    SIGNAL  rdatacrc: std_logic_vector(16 DOWNTO 0);
    SIGNAL  st,rt  : std_logic;
BEGIN
PROCESS(clk)
```

接下页

```
    VARIABLE crcvar : std_logic_vector(5 DOWNTO 0);
BEGIN
    IF(clk'event AND clk = '1') THEN
        IF(st = '0' AND datald = '1') THEN  dtemp <= sdata;
  sdatam <= sdata; cnt <= (OTHERS => '0');  hsend <= '0';  st <= '1';
        ELSIF(st = '1' AND cnt < 7) THEN  cnt <= cnt + 1;
        IF(dtemp(11)='1')THEN crcvar:=dtemp(11 DOWNTO 6)XOR multi_coef;
           dtemp <= crcvar(4 DOWNTO 0) & dtemp(5 DOWNTO 0) & '0';
            ELSE  dtemp <= dtemp(10 DOWNTO 0) & '0';   END IF;
        ELSIF(st='1' AND cnt=7) THEN datacrco<=sdatam & dtemp(11 DOWNTO 7);
           hsend <= '1';  cnt <= cnt + 1;
        ELSIF(st='1' AND cnt=8) THEN   hsend<= '0';    st<='0';
        END IF;
    END IF;
END PROCESS;
PROCESS(hrecv,clk)
    VARIABLE rcrcvar : std_logic_vector(5 DOWNTO 0);
BEGIN
    IF(clk'event AND clk = '1') THEN
      IF(rt='0' AND hrecv = '1') THEN  rdtemp <= datacrci(16 DOWNTO 5);
       rdatacrc <= datacrci;  rcnt <= (OTHERS => '0');
       ERROR0 <= '0';    rt <= '1';
       ELSIF(rt='1' AND rcnt<7) THEN  datafini<='0';  rcnt <= rcnt + 1;
```

接下页

实验与设计

9-3 循环冗余校验（**CRC**）模块设计

```
            rcrcvar := rdtemp(11 DOWNTO 6) XOR multi_coef;
            IF(rdtemp(11) = '1') THEN
             rdtemp <= rcrcvar(4 DOWNTO 0) & rdtemp(5 DOWNTO 0) & '0';
            ELSE  rdtemp <= rdtemp(10 DOWNTO 0) & '0';
            END IF;
        ELSIF(rt = '1' AND rcnt = 7) THEN  datafini <= '1';
            rdata <= rdatacrc(16 DOWNTO 5);  rt <= '0';
            IF(rdatacrc(4 DOWNTO 0) /= rdtemp(11 DOWNTO 7)) THEN
              ERROR0 <= '1'; END IF;
        END IF;
    END IF;
END PROCESS;
END comm;
```

实验与设计

9-4 SPWM脉宽调制控制系统设计

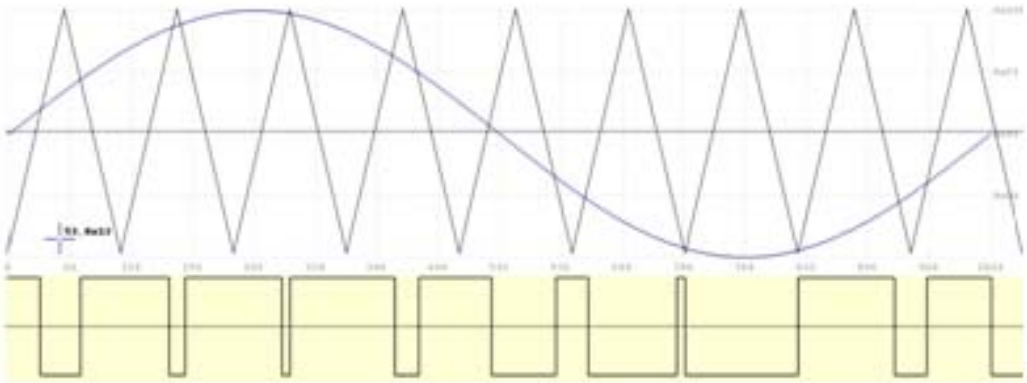

图 9-18 SPWM 波生成原理图

【例 9-12】三角波发生模块

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_unsigned.all;
USE ieee.std_logic_arith.all;
ENTITY TRANG3 IS
    PORT ( ADR : IN std_logic_vector(9 downto 0);
         OUTD : OUT std_logic_vector(9 downto 0));
END TRANG3 ;
ARCHITECTURE rtl OF TRANG3  IS
SIGNAL OT1 : std_logic_vector(9 downto 0);
SIGNAL CC  : std_logic_vector(10 downto 0);
BEGIN
 process(ADR,CC)   begin
     IF (ADR<"1000000000") THEN
       OT1(9 downto 1) <= ADR(8 downto 0); OT1(0)<='0';
       ELSE CC<="10000000000" + (NOT ADR) ;
           OT1(9 downto 1)<=CC(8 downto 0); OT1(0)<='0';  END IF;
   end process;
   OUTD<=OT1;
 END rtl;
```

实验与设计

9-4 SPWM脉宽调制控制系统设计

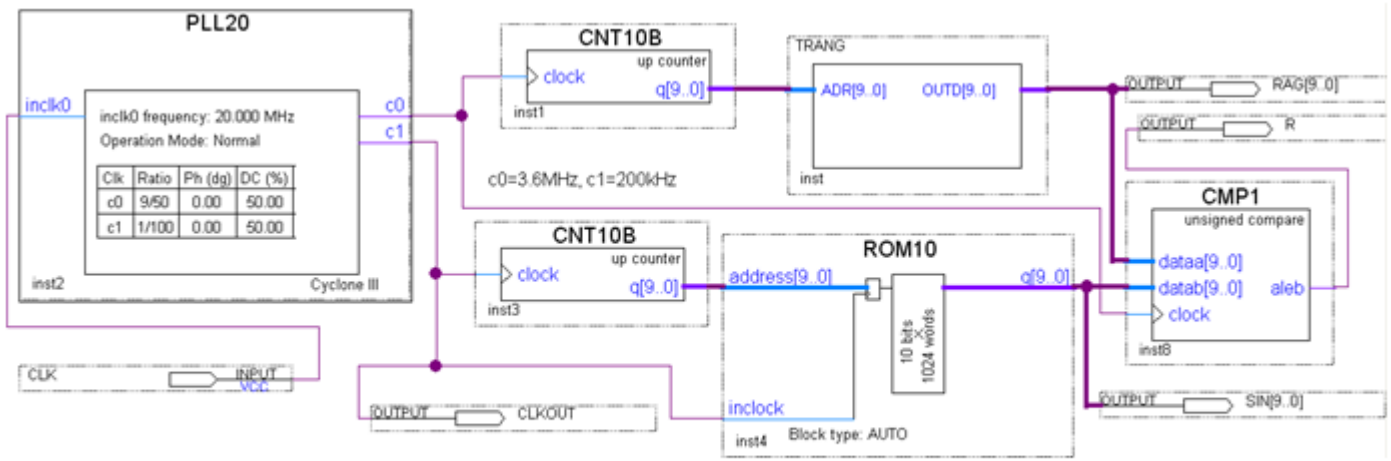

图 9-19　SPWM 波发生器基本电路图

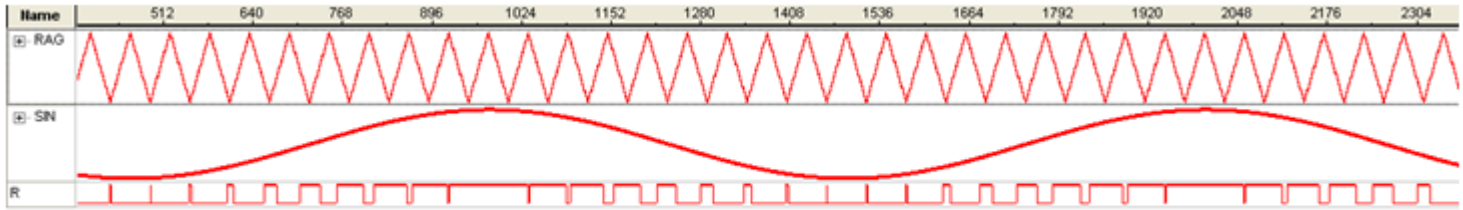

图 9-20　图 9-19 电路的 SignalTap II 实测波形

实验与设计

9-4 SPWM脉宽调制控制系统设计

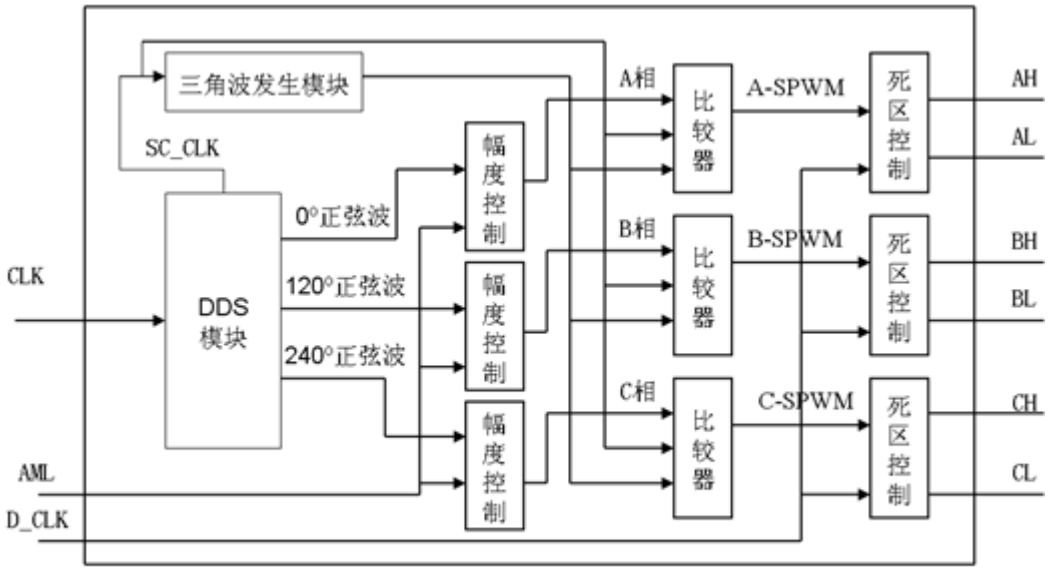

图 9-21 三相 SPWM 控制器电路模块图

实验与设计

9-5 步进电机细分控制电路设计

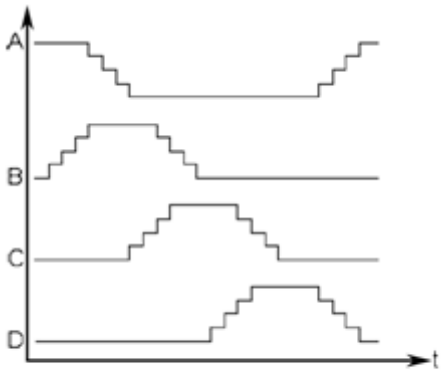

图 9-22 四相步进电机八细分电流波形

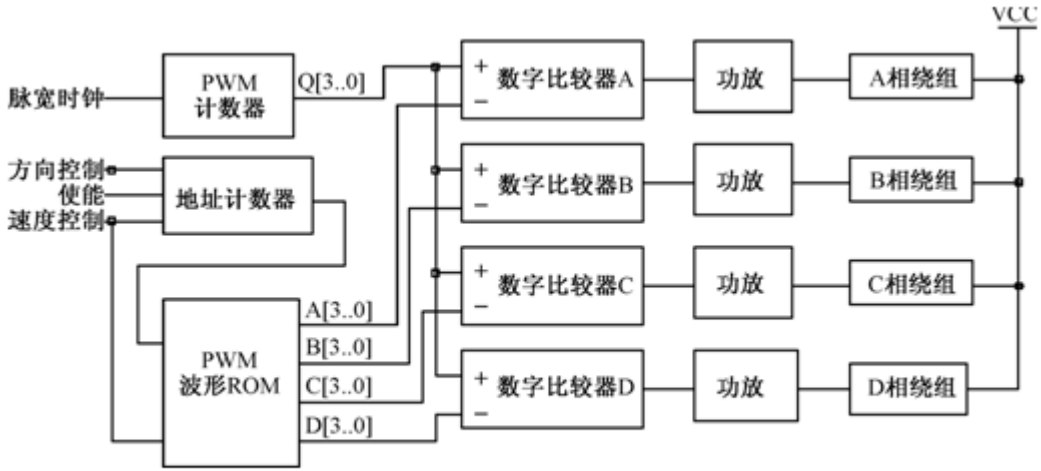

图 9-23 步进电机细分驱动电路结构图

实验与设计

9-5 步进电机细分控制电路设计

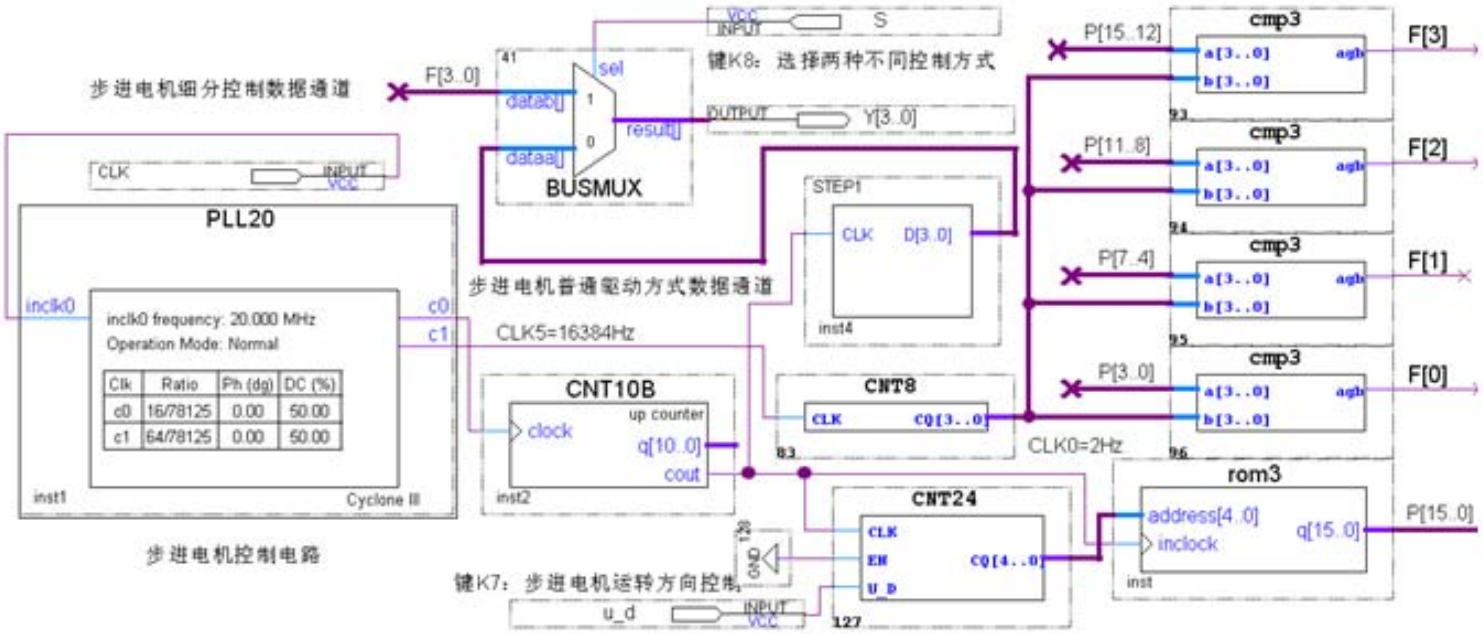

图 9-24 步进电机细分控制电路

9-6 数字彩色液晶显示控制电路设计