



EDA 技术实用教程

第 13 章

电子系统设计实践

13.1 VGA彩条信号显示控制器设计



VGA工业标准要求的频率：

时钟频率(Clock frequency)：25.175 MHz (像素输出的频率)

行频(Line frequency)：31469 Hz

场频(Field frequency)：59.94 Hz (每秒图像刷新频率)

13.1 VGA彩条信号显示控制器设计

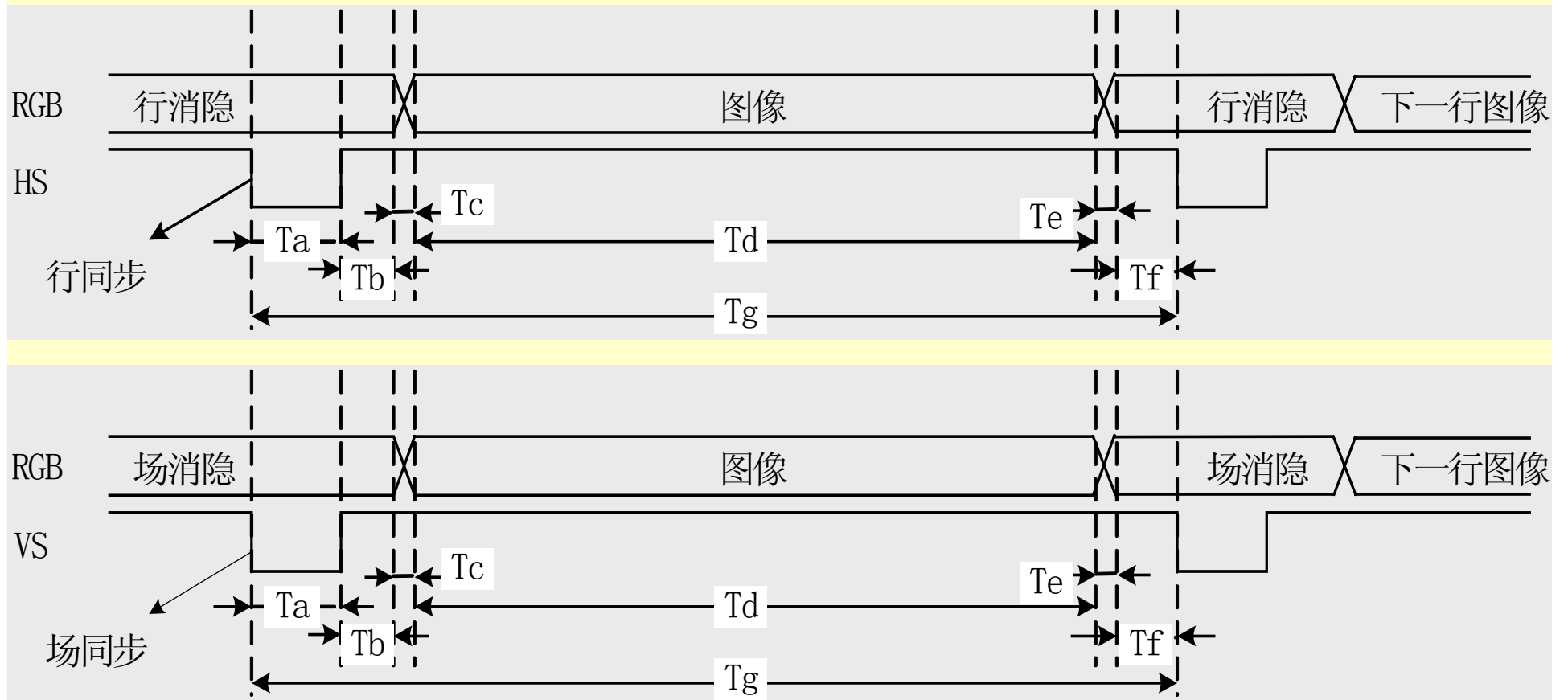


图13-1 VGA行扫描、场扫描时序示意图

13.1 VGA彩条信号显示控制器设计

表13-1 行扫描时序要求：(单位：像素，即输出一个像素Pixel的时间间隔)

		行同步头			行图像		行周期
对应位置	Tf	Ta	Tb	Tc	Td	Te	Tg
时间(Pixels)	8	96	40	8	640	8	800

表13-1 行扫描时序要求：(单位：像素，即输出一个像素Pixel的时间间隔)

		行同步头			行图像		行周期
对应位置	Tf	Ta	Tb	Tc	Td	Te	Tg
时间(Lines)	2	2	25	8	480	8	525

13.1 VGA彩条信号显示控制器设计

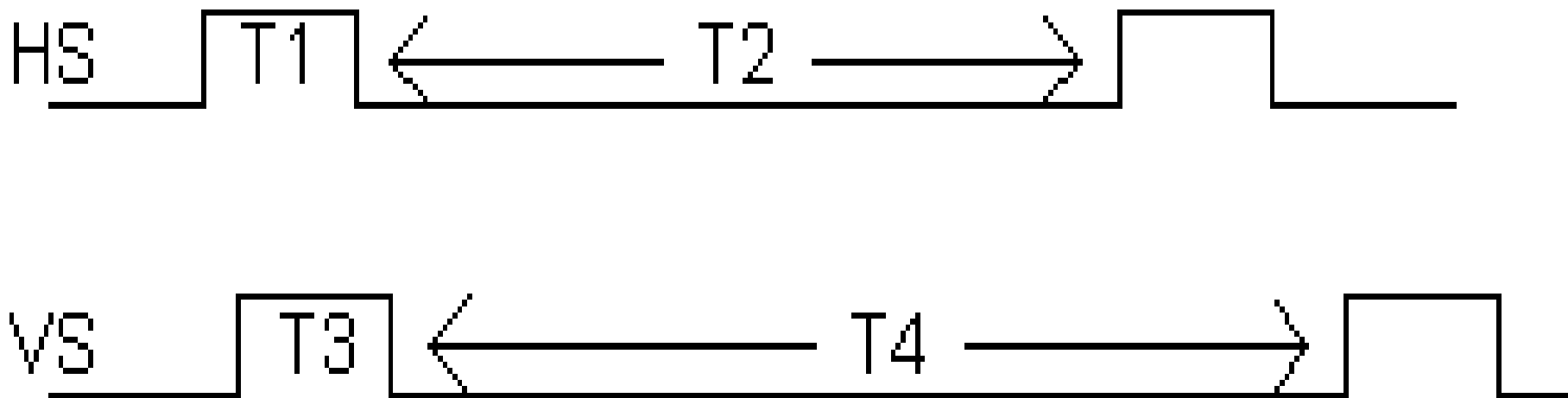


图13-2 HS和VS的时序图

13.1 VGA彩条信号显示控制器设计

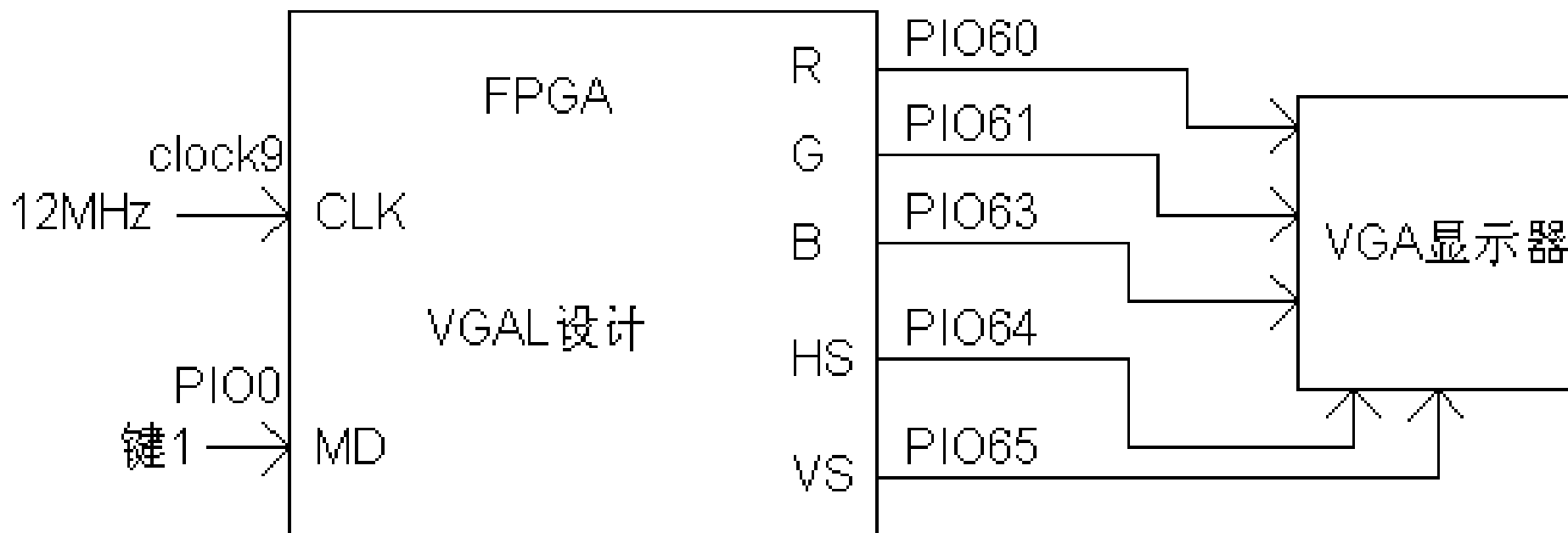


图13-3 例13-7实现电路

13.1 VGA彩条信号显示控制器设计

表13-3 颜色编码:

颜色	黑	蓝	红	品	绿	青	黄	白
R	0	0	0	0	1	1	1	1
G	0	0	1	1	0	0	1	1
B	0	1	0	1	0	1	0	1

表13-4彩条信号发生器3种显示模式

1	横彩条	1: 白黄青绿品红蓝黑	2: 黑蓝红品绿青黄白
2	竖彩条	1: 白黄青绿品红蓝黑	2: 黑蓝红品绿青黄白
3	棋盘格	1: 棋盘格显示模式1	2: 棋盘格显示模式2

13.1 VGA彩条信号显示控制器设计

【例13-1】

```
LIBRARY IEEE;    -- VGA显示器 彩条 发生器
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY COLOR IS
    PORT (
        CLK, MD : IN STD_LOGIC;
        HS, VS, R, G, B : OUT STD_LOGIC ); -- 行场同步/红, 绿, 蓝
END COLOR;
ARCHITECTURE behav OF COLOR IS
    SIGNAL HS1, VS1, FCLK, CCLK      : STD_LOGIC;
    SIGNAL MMD : STD_LOGIC_VECTOR(1 DOWNTO 0); -- 方式选择
    SIGNAL FS : STD_LOGIC_VECTOR (3 DOWNTO 0);
    SIGNAL CC : STD_LOGIC_VECTOR(4 DOWNTO 0); --行同步/横彩条生成
    SIGNAL LL : STD_LOGIC_VECTOR(8 DOWNTO 0); --场同步/竖彩条生成
    SIGNAL GRBX : STD_LOGIC_VECTOR(3 DOWNTO 1); -- X横彩条
    SIGNAL GRBY : STD_LOGIC_VECTOR(3 DOWNTO 1); -- Y竖彩条
    SIGNAL GRBP : STD_LOGIC_VECTOR(3 DOWNTO 1);
    SIGNAL GRB  : STD_LOGIC_VECTOR(3 DOWNTO 1);
BEGIN
```

(接下页)


```

GRB(2) <= (GRBP(2) XOR MD) AND HS1 AND VS1;
GRB(3) <= (GRBP(3) XOR MD) AND HS1 AND VS1;
GRB(1) <= (GRBP(1) XOR MD) AND HS1 AND VS1;
PROCESS( MD )
BEGIN
    IF MD'EVENT AND MD = '0' THEN
        IF MMD = "10" THEN MMD <= "00";
        ELSE MMD <= MMD + 1; END IF;           --三种模式
    END IF;
END PROCESS;
PROCESS( MMD )
BEGIN
    IF MMD = "00" THEN GRBP <= GRBX;           -- 选择横彩条
    ELSIF MMD = "01" THEN GRBP <= GRBY;       -- 选择竖彩条
    ELSIF MMD = "10" THEN GRBP <= GRBX XOR GRBY; --产生棋盘格
    ELSE GRBP <= "000"; END IF;
END PROCESS;
PROCESS( CLK )
BEGIN
    IF CLK'EVENT AND CLK = '1' THEN -- 13MHz 13分频
        IF FS = 13 THEN FS <= "0000";
        ELSE FS <= (FS + 1); END IF;
    END IF;
END PROCESS;
FCLK <= FS(3); CCLK <= CC(4);
PROCESS( FCLK )
BEGIN

```

(接下页)

```

IF FCLK'EVENT AND FCLK = '1' THEN
    IF CC = 29 THEN CC <= "00000";
    ELSE CC <= CC + 1; END IF;
    END IF;
END PROCESS;
PROCESS( CCLK )
BEGIN
    IF CCLK'EVENT AND CCLK = '0' THEN
        IF LL = 481 THEN LL <= "0000000000";
        ELSE LL <= LL + 1; END IF;
    END IF;
END PROCESS;
PROCESS( CC,LL )
BEGIN
    IF CC > 23 THEN HS1 <= '0'; --行同步
    ELSE HS1 <= '1'; END IF;
    IF LL > 479 THEN VS1 <= '0'; --场同步
    ELSE VS1 <= '1'; END IF;
END PROCESS;
PROCESS(CC, LL)
BEGIN
    IF CC < 3 THEN GRBX <= "111"; -- 横彩条
    ELSIF CC < 6 THEN GRBX <= "110";
    ELSIF CC < 9 THEN GRBX <= "101";
    ELSIF CC < 13 THEN GRBX <= "100";
    ELSIF CC < 15 THEN GRBX <= "011";

```

(接下页)

```

ELSIF CC < 18 THEN GRBX <= "010";
  ELSIF CC < 21 THEN GRBX <= "001";
  ELSE GRBX <= "000";
  END IF;
  IF LL < 60 THEN GRBY <= "111"; -- 竖彩条
  ELSIF LL < 130 THEN GRBY <= "110";
  ELSIF LL < 180 THEN GRBY <= "101";
  ELSIF LL < 240 THEN GRBY <= "100";
  ELSIF LL < 300 THEN GRBY <= "011";
  ELSIF LL < 360 THEN GRBY <= "010";
  ELSIF LL < 420 THEN GRBY <= "001";
  ELSE GRBY <= "000";
  END IF;
END PROCESS;
HS <= HS1 ; VS <= VS1 ;R <= GRB(2) ;G <= GRB(3) ; B <= GRB(1);
END behav;

```

13.2 VGA图像显示控制器设计

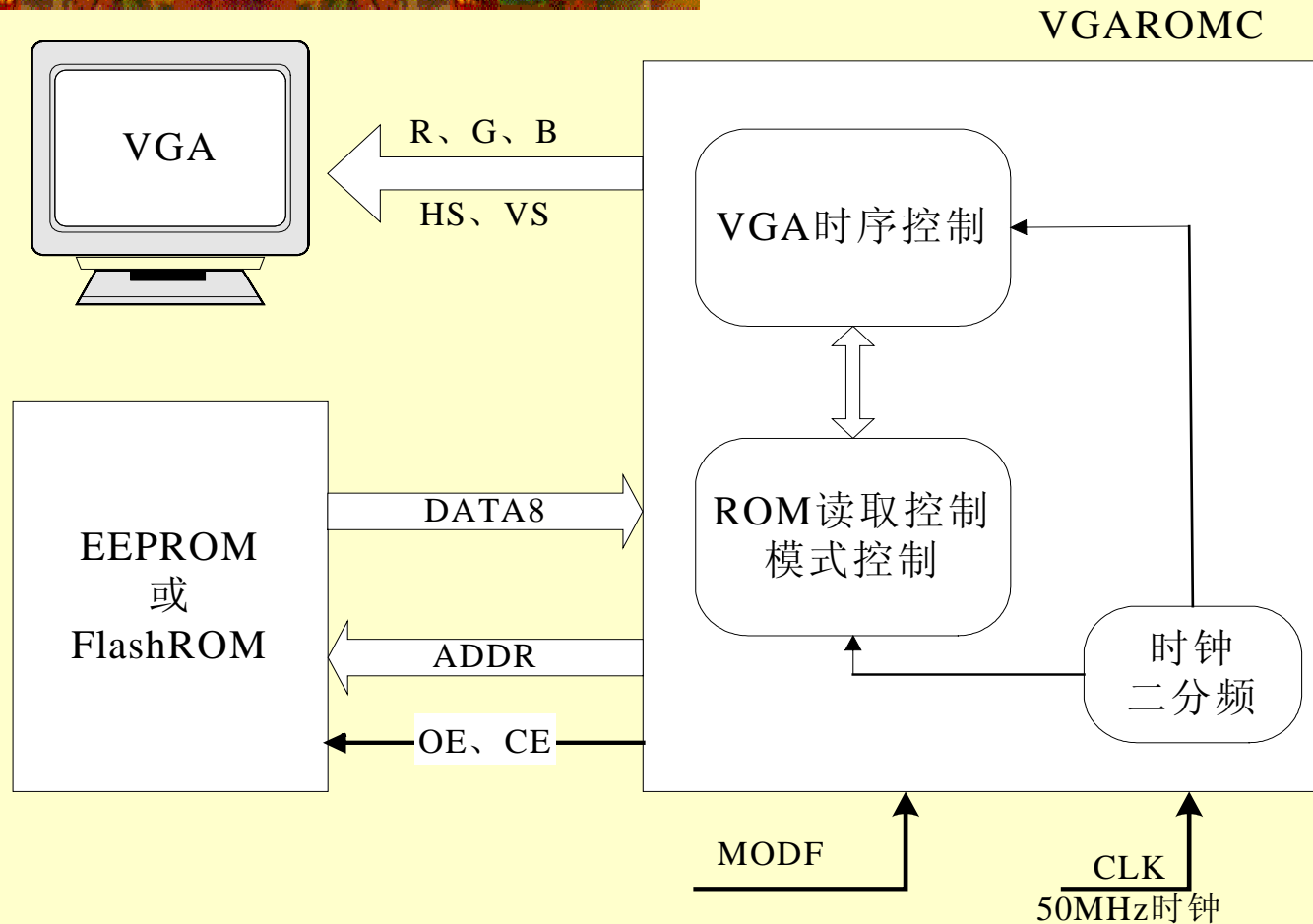


图13-4 VGA图像控制器框图

【例13-2】

```
LIBRARY ieee; --图象显示顶层程序
USE ieee.std_logic_1164.all;
ENTITY img IS
    port
        ( clk50MHz : IN STD_LOGIC;
          hs,      vs,      r, g, b : OUT STD_LOGIC );
END img;
ARCHITECTURE modelstru OF img IS
    component vga640480 --VGA显示控制模块
        PORT(clk : IN STD_LOGIC;
             rgb_in : IN STD_LOGIC_VECTOR(2 downto 0);
             hs, vs, r, g, b : OUT STD_LOGIC;
             hcntout, vcntout : OUT STD_LOGIC_VECTOR(9 downto 0) );
    end component;
    component imgrom --图象数据ROM, 数据线3位; 地址线13位
        PORT(inclock : IN STD_LOGIC;
             address : IN STD_LOGIC_VECTOR(11 downto 0);
             q : OUT STD_LOGIC_VECTOR(2 downto 0) );
    end component;
    signal rgb : STD_LOGIC_VECTOR(2 downto 0);
    signal clk25MHz : std_logic;
    signal romaddr : STD_LOGIC_VECTOR(11 downto 0);
    signal hpos, vpos : std_logic_vector(9 downto 0);
    BEGIN
        romaddr <= vpos(5 downto 0) & hpos(5 downto 0);
        process(clk50MHz) begin
            if clk50MHz'event and clk50MHz = '1' then clk25MHz <= not clk25MHz ; end if;
        end process;
        i_vga640480 : vga640480 PORT MAP(clk => clk25MHz, rgb_in => rgb, hs => hs,
            vs => vs, r => r, g => g, b => b, hcntout => hpos, vcntout => vpos);
        i_rom : imgrom PORT MAP(inclock => clk25MHz, address => romaddr, q => rgb);
    END;
```

【例13-3】

```
LIBRARY IEEE
use IEEE.std_logic_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity vga640480 is
    port ( clk      : in STD_LOGIC;
          hs,      vs,      r,g,b  : out STD_LOGIC;
          rgbina   : in std_logic_vector(2 downto 0);
          hcntout,vcntout      : out std_logic_vector(9 downto 0)
    );
end vga640480;
architecture ONE of vga640480 is
    signal hcnt, vcnt      : std_logic_vector(9 downto 0);
begin
    hcntout <= hcnt; vcntout <= vcnt;
    process(clk) begin
        if (rising_edge(clk)) then
            if(hcnt < 800) then      hcnt <= hcnt + 1;
            else hcnt <= (others => '0'); end if;
        end if;
    end process;
    process(clk) begin
        if (rising_edge(clk)) then
            if (hcnt = 640+8 ) then
                if(vcnt < 525) then      vcnt <= vcnt + 1;
                else vcnt <= (others => '0');      end if;
            end if;
        end if;
    end process;
end architecture ONE;
```

(接下页)

```

        end if;
    end if;
end process;
process(clk) begin
    if (rising_edge(clk)) then
        if((hcnt>=640+8+8) and (hcnt<640+8+8+96 )) then hs<='0';
        else hs <= '1';          end if;
    end if;
end process;
process(vcnt) begin
    if ((vcnt >= 480+8+2) and (vcnt<480+8+2+2)) then vs <= '0';
        else vs<='1'; end if;
end process;
process(clk) begin
    if (rising_edge(clk)) then
        if (hcnt<640 and vcnt<480) then
            r<=rgbin(2); g<=rgbin(1); b<=rgbin(0);
        else r<='0'; g<='0'; b<='0'; end if;
    end if;
end process;
end ONE;

```

13.3 步进电机细分驱动控制

- 1、步进电机细分驱动原理
- 2、步距细分的系统构成

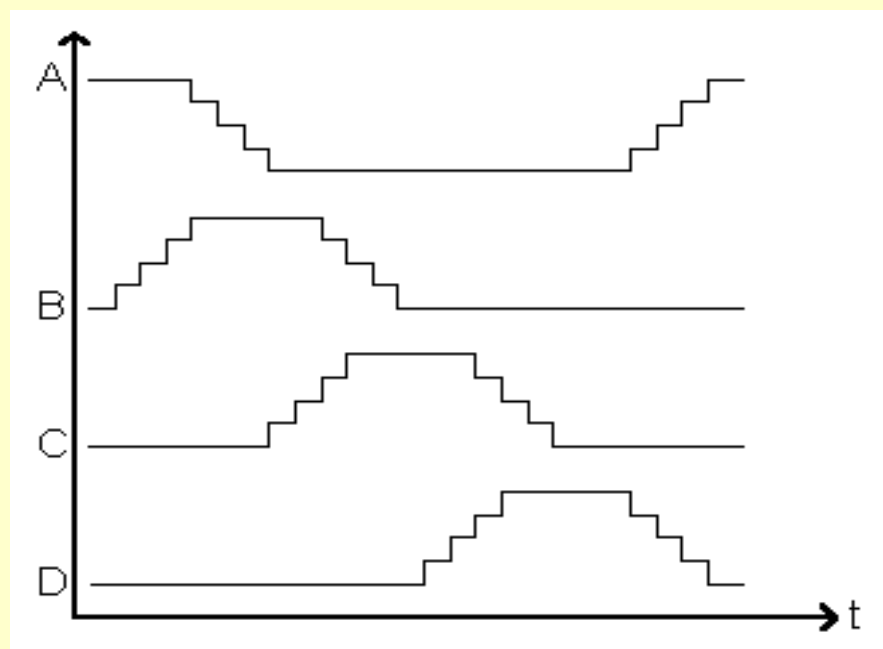


图13-5 四相步进电机8细分电流波形

13.3 步进电机细分驱动控制

2、步距细分的系统构成

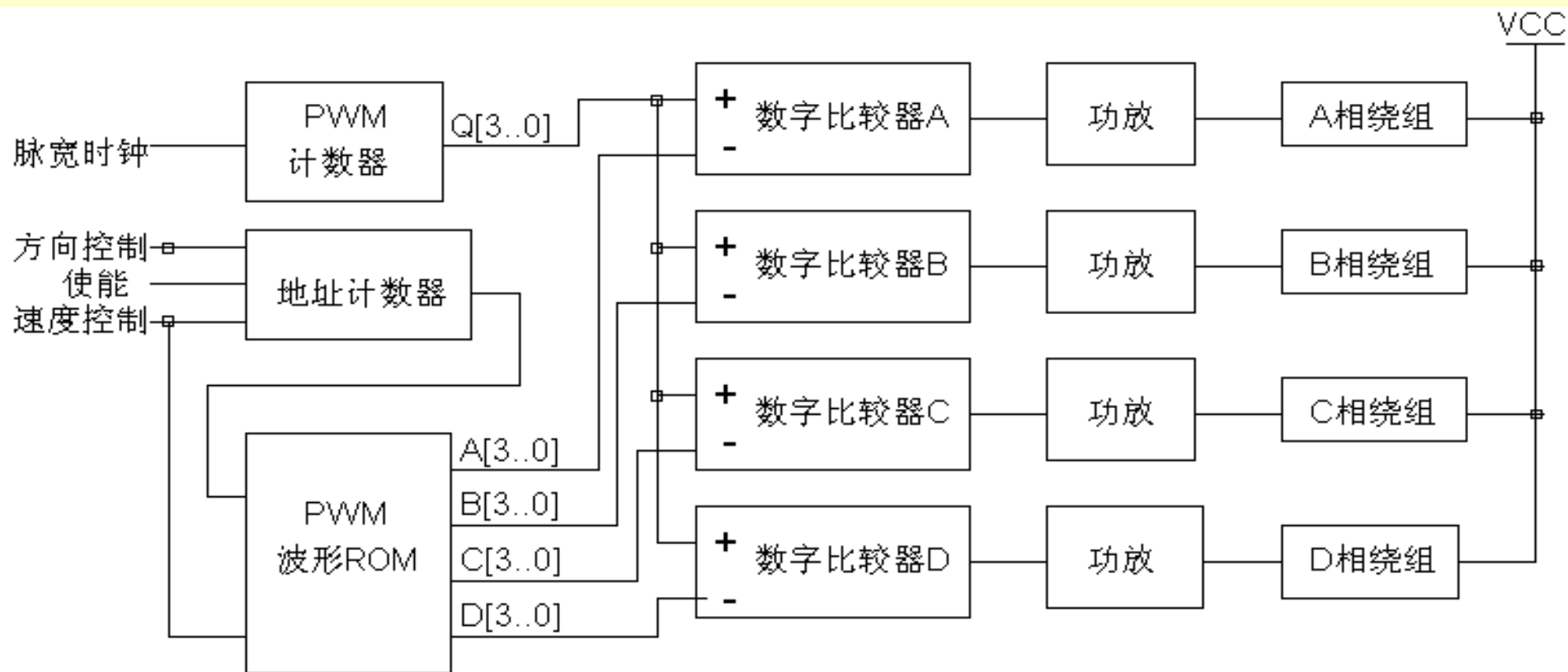


图13-6 步进电机细分驱动电路结构图

13.3 步进电机细分驱动控制

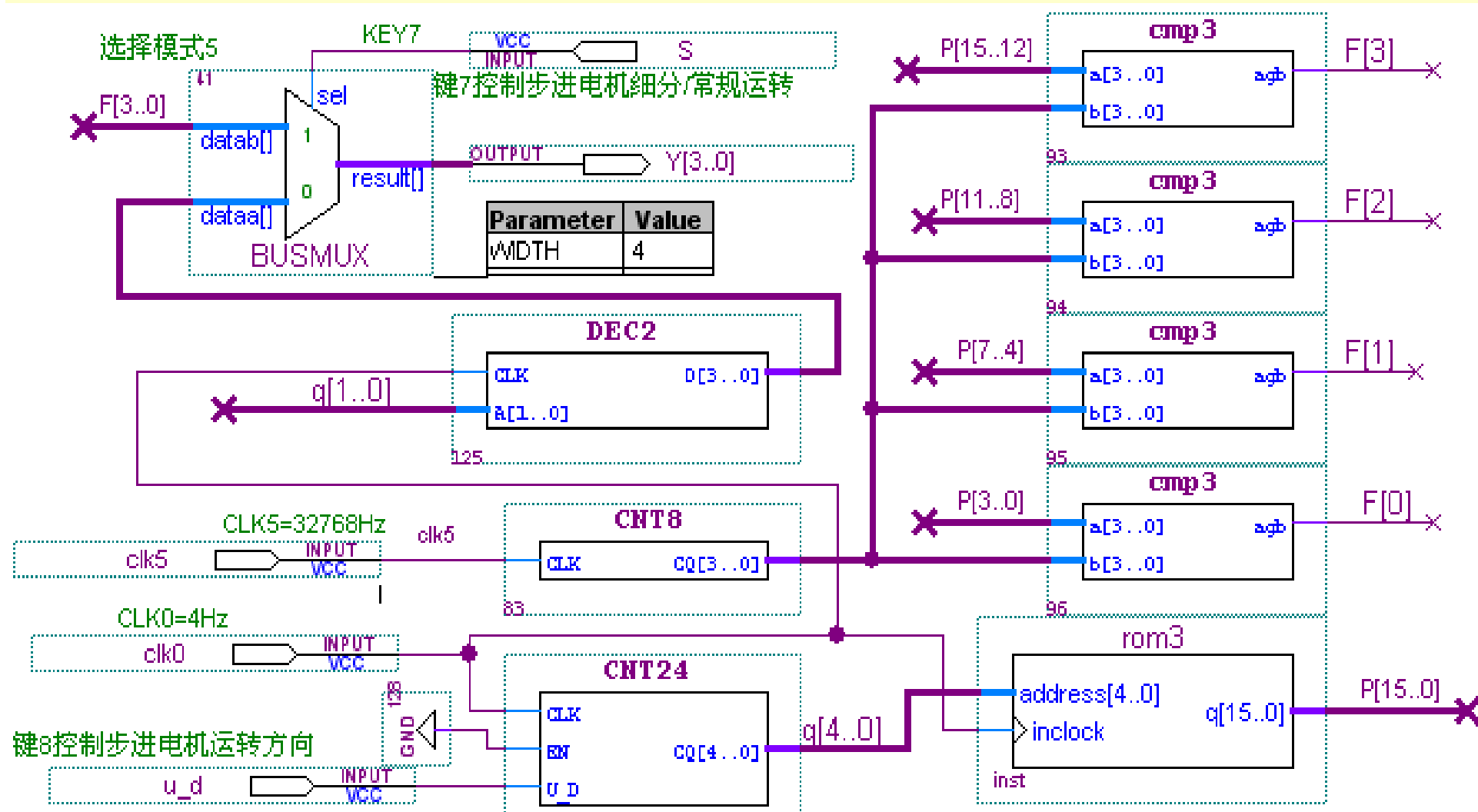


图13-7 步进电机PWM细分控制控制电路图

13.3 步进电机细分驱动控制

2、步距细分的系统构成

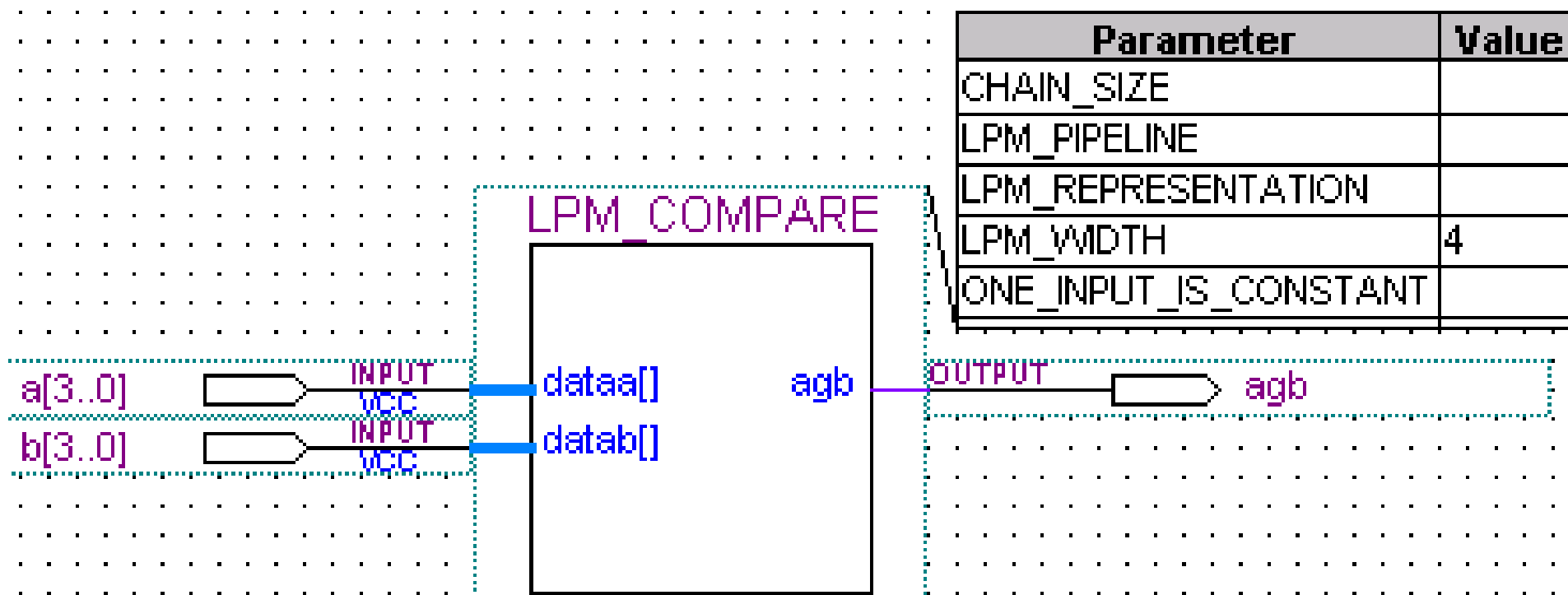


图13-8 图13-7中的cmp3模块

13.3 步进电机细分驱动控制

2、步距细分的系统构成

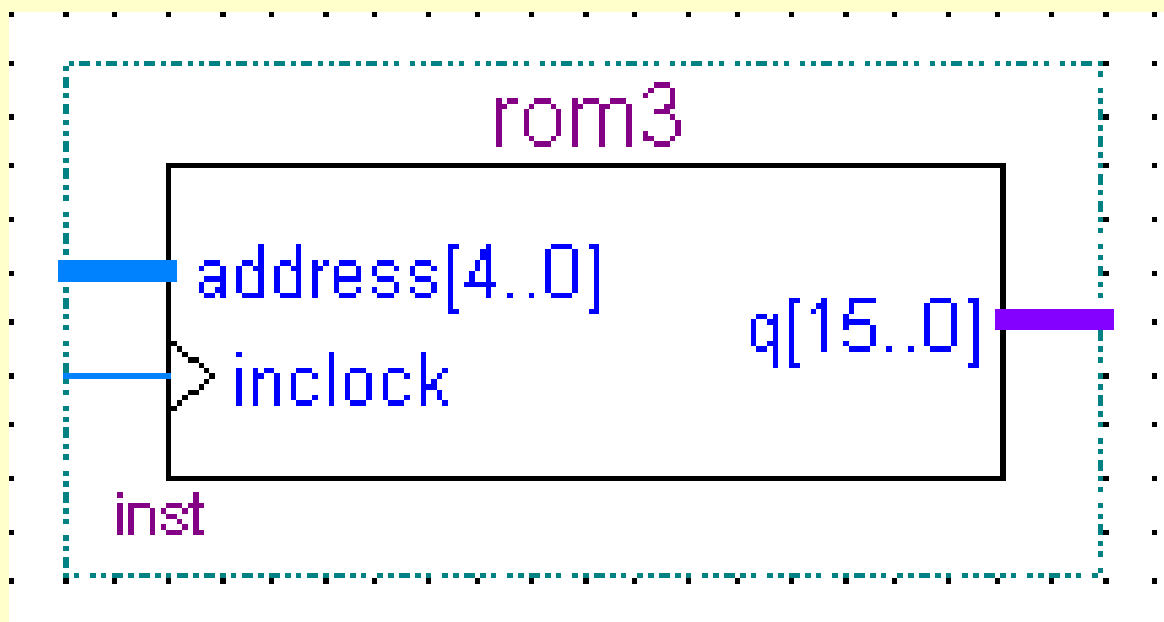


图13-9 PWM波形ROM存储器

13.3 步进电机细分驱动控制

3、细分电流信号的实现

4、细分驱动性能的改善

5、细工作时序分析

13.3 步进电机细分驱动控制

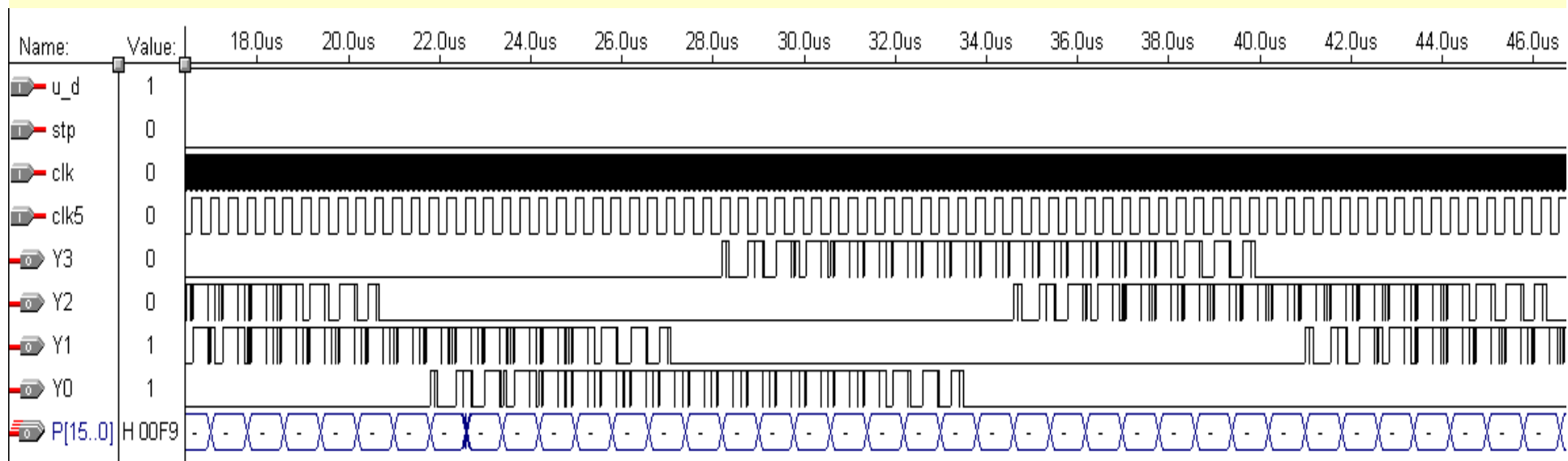


图13-10 步进电机PWM仿真波形图（注意，图中clk与clk5交换）

13.3 步进电机细分驱动控制

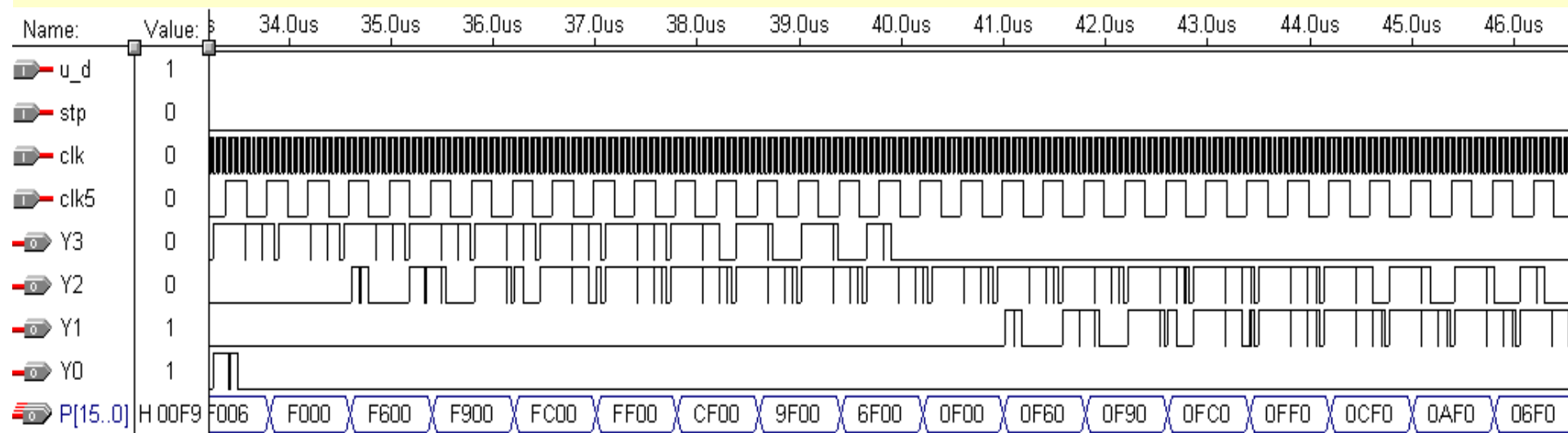


图13-11 展开后的步进电机PWM仿真波形图（注意，图中clk与clk5交换）

13.3 步进电机细分驱动控制

6、硬件验证

【例13-4】--元件CNT8

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY CNT8 IS
    PORT ( CLK : IN STD_LOGIC;
          CQ : OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
END CNT8;
ARCHITECTURE behav OF CNT8 IS
    SIGNAL CQI : STD_LOGIC_VECTOR(4 DOWNTO 0);
BEGIN
    PROCESS(CLK)
    BEGIN
        IF CLK'EVENT AND CLK = '1' THEN CQI <= CQI + 1; END IF;
    END PROCESS;
    CQ <= CQI(4 DOWNTO 1);
END behav;
```

-- 8进制计数器

【例13-5】--元件DEC2

```
LIBRARY IEEE ;
USE IEEE.STD_LOGIC_1164.ALL ;
ENTITY Dec2 IS
    PORT ( CLK : IN STD_LOGIC;
          A  : IN  STD_LOGIC_VECTOR(1 DOWNTO 0) ;
          D  : OUT STD_LOGIC_VECTOR(3 DOWNTO 0) ) ;
END ;
ARCHITECTURE one OF Dec2 IS
    SIGNAL CQ : STD_LOGIC_VECTOR(1 DOWNTO 0);
BEGIN
    PROCESS( CQ )
        BEGIN
            CASE CQ IS
                WHEN "00" => D <= "1001" ;
                WHEN "01" => D <= "1100" ;
                WHEN "10" => D <= "0110" ;
                WHEN "11" => D <= "0011" ;
                WHEN OTHERS => NULL ;
            END CASE ;
        END PROCESS ;
    PROCESS(CLK)
        BEGIN
            IF CLK'EVENT AND CLK = '1' THEN CQ <= A; END IF;
        END PROCESS;
END ;
```

【例13-6】 --元件CNT24

-- 24进制计数器

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY CNT24 IS
    PORT ( CLK,EN, U_D : IN STD_LOGIC;
          CQ : OUT STD_LOGIC_VECTOR(4 DOWNTO 0));
END CNT24;
ARCHITECTURE behav OF CNT24 IS
    SIGNAL CQI : STD_LOGIC_VECTOR(4 DOWNTO 0);
BEGIN
    PROCESS(CLK, EN, U_D)
    BEGIN
        IF EN = '1' THEN    CQI <= CQI;
        ELSIF CLK'EVENT AND CLK = '1' THEN
            IF U_D = '1' THEN    CQI <= CQI + 1;
                                ELSE CQI<= CQI-1;    END IF;
        END IF;
    END PROCESS;
    CQ(4 DOWNTO 0) <= CQI;
END behav;
```

13.3 步进电机细分驱动控制

6、硬件验证

【例13-7】 --5位地址线ROM3中的数据: pwm_1.mif, 注意, 实用每一组数据要占一行

```
WIDTH = 16;  
DEPTH = 32;  
ADDRESS_RADIX = HEX;  
DATA_RADIX = HEX;  
CONTENT BEGIN  
0:f000; 1:f600; 2:f900; 3:fc00; 4:ff00; 5:cf00; 6:9f00; 7:6f00; 8:0f00;  
9:0f60; a:0f90; b:0fc0; c:0ff0; d:0cf0; e:0af0; f:06f0; 10:00f0; 11:00f6;  
13:00f9;13:00fc; 14:00ff; 15:00cf; 16:009f; 17:006f; 18:000f; 19:600f;  
1a:900f;1b:c00f; 1c:f00f; 1d:f00c; 1e:f009; 1f:f006;  
END;
```

13.4 直流电机的PWM控制

直流电机控制电路主要由三部分组成：

- **FPGA中PWM脉宽调制信号产生电路**
- **FPGA中的工作/停止控制和正/反转方向控制电路**
- **由功率放大电路和H桥组成的正反转功率驱动电路**

13.4 直流电机的PWM控制

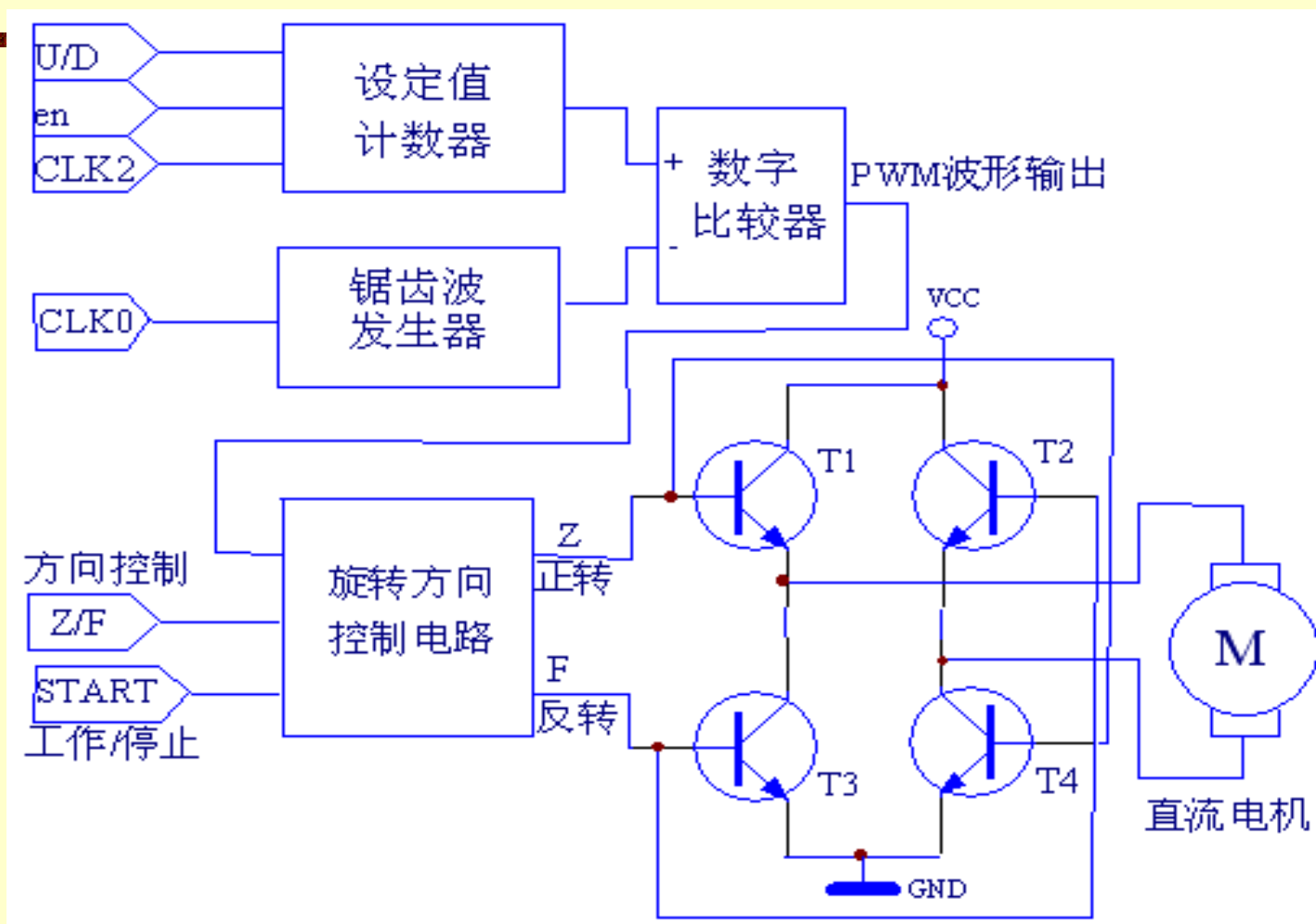


图13-12 FPGA直流电机驱动控制电路

13.4 直流电机的PWM控制

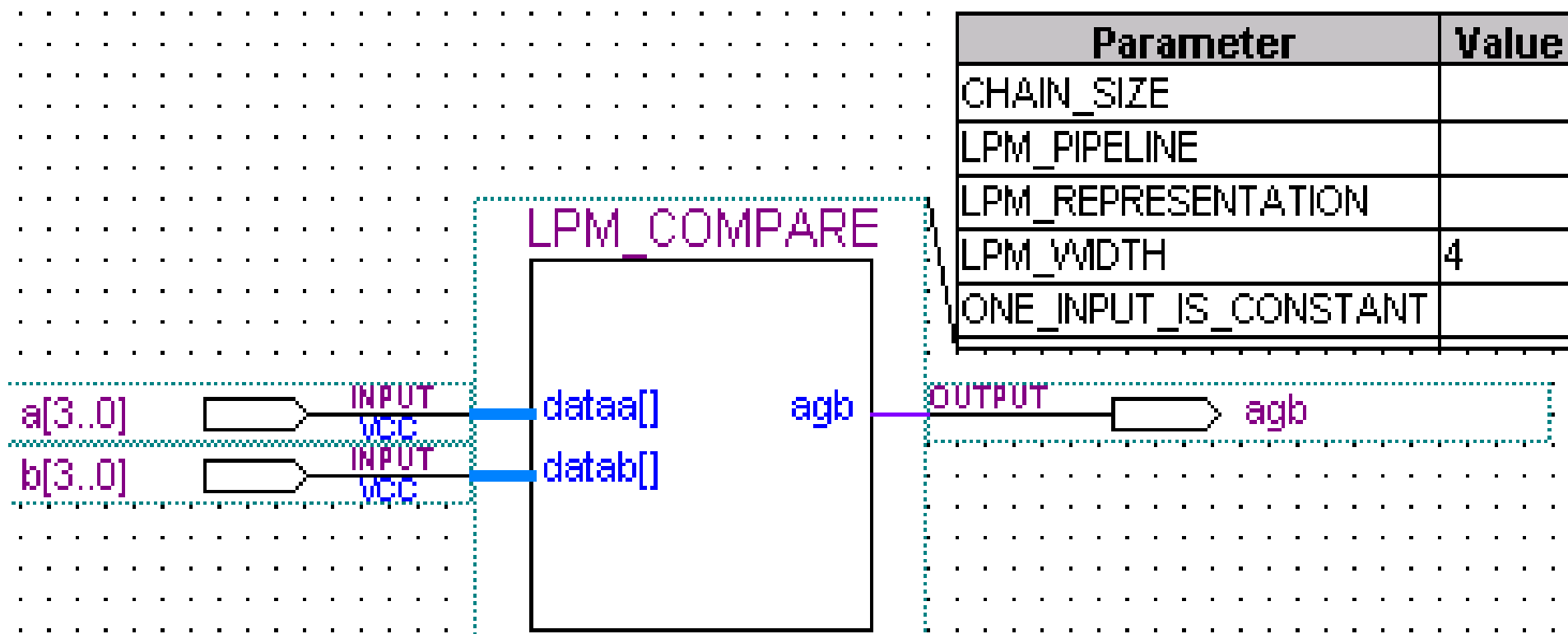


图13-13图13-14中的cmp3模块

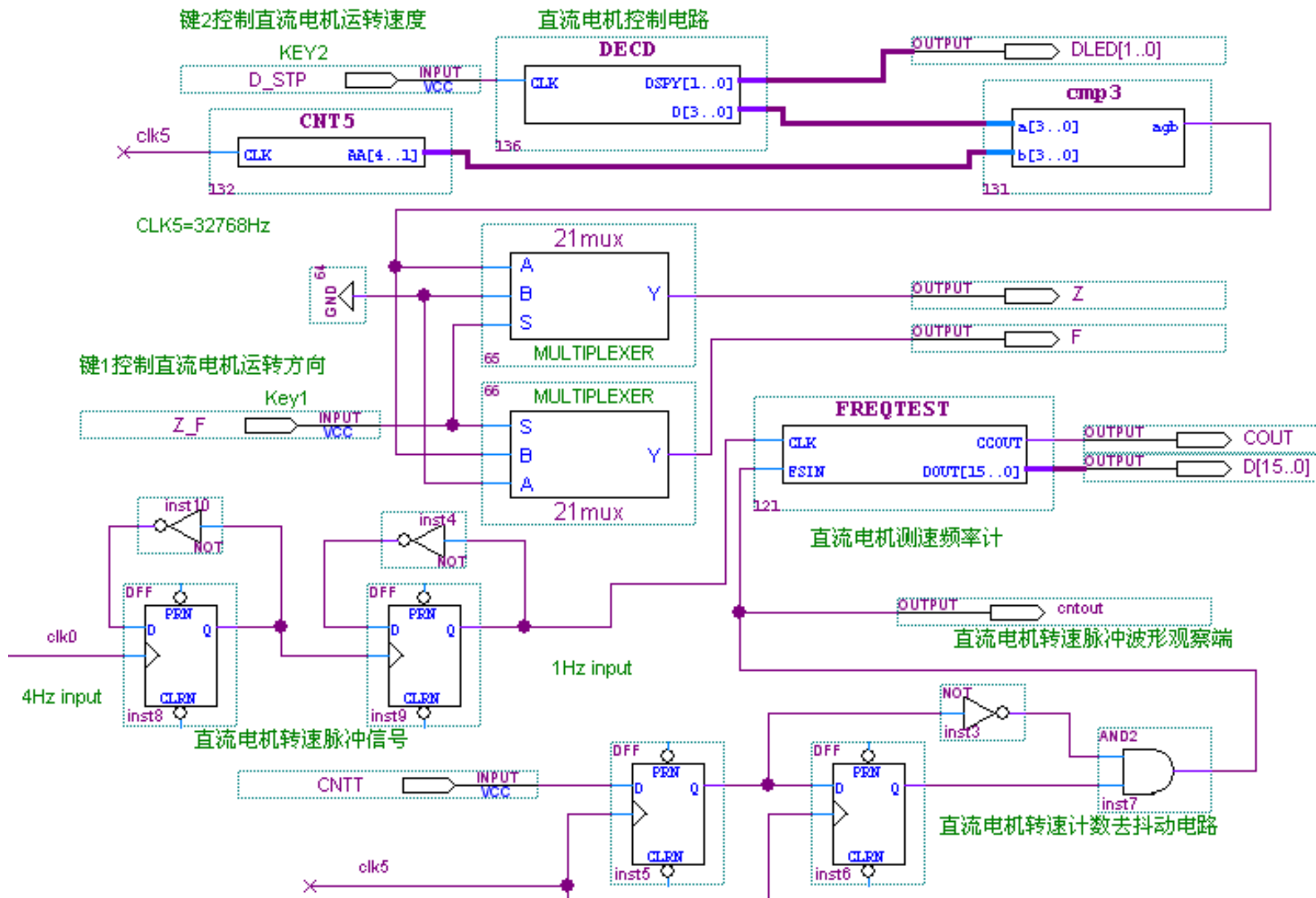


图13-14 FPGA直流电机控制模块

【例13-8】

```
LIBRARY IEEE ;
USE IEEE.STD_LOGIC_1164.ALL ;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY DECD IS
    PORT ( CLK : IN STD_LOGIC;
          DSPY : OUT STD_LOGIC_VECTOR(1 DOWNTO 0) ;
          D : OUT STD_LOGIC_VECTOR(3 DOWNTO 0) ) ;
END ;
ARCHITECTURE one OF DECD IS
    SIGNAL CQ : STD_LOGIC_VECTOR(1 DOWNTO 0);
BEGIN
    PROCESS( CQ )
        BEGIN
            CASE CQ IS
                WHEN "00" => D <= "0100" ;
                WHEN "01" => D <= "0111" ;
                WHEN "10" => D <= "1011" ;
                WHEN "11" => D <= "1111" ;
                WHEN OTHERS => NULL ;
            END CASE ;
        END PROCESS ;
    PROCESS(CLK)
        BEGIN
            IF CLK'EVENT AND CLK = '1' then CQ <= CQ + 1; END IF;
        END PROCESS;
    DSPY<=CQ;
END ;
```


13.4 直流电机的PWM控制

【例13-9】

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY CNT5 IS
    PORT ( CLK : IN STD_LOGIC;
          AA : OUT STD_LOGIC_VECTOR(4 DOWNTO 1));
END CNT5;
ARCHITECTURE behav OF CNT5 IS
    SIGNAL CQI : STD_LOGIC_VECTOR(4 DOWNTO 0);
BEGIN
    PROCESS(CLK)
    BEGIN
        IF CLK'EVENT AND CLK = '1' then CQI <= CQI + 1; END IF;
    END PROCESS;
    AA <= CQI(4 DOWNTO 1);
END behav;
```

-- 4进制计数器



习题

- 13-1.** 详述VGA显示控制原理。
- 13-2.** 试在通用异步收发器UART中加入FIFO，以缓冲接收，发生数据。
- 13-3.** 把VGA控制器模块与UART模块连接起来，实现VGA显示图像的动态更新。
- 13-4.** 简述步进电机转角细分的工作原理，有哪些方法可以实现步进转角细分控制？
- 13-5.** 步进电机相电流的细分与步进转角细分是一回事吗？有何区别？要提高步进电机转角细分的控制精度，可以采取哪些方法？



习题

- 13-6.** 要使步进电机按预先设定的角度转动，控制电路应如何设计？
- 13-7.** 有哪些方法可以对直流电机进行调速控制？如何用FPGA对直流电机进行调速控制？
- 13-8.** 若要使电机转速设置更精确，可以采取哪些措施，控制电路应如何修改？
- 13-9.** 要使直流电机精确地达到设定转速，可以通过检测电机的转速，采用速度闭环控制。如何通过实验台上的光电检测装置检测电机转速，如何用FPGA实现速度闭环控制？



实验与设计

13-1. VGA彩条信号显示控制器设计

(1) 实验目的：学习VGA图像显示控制器的设计。

(2) 实验内容1：根据图13-3和程序13-1，完成VGA彩条信号显示的验证性实验。根据图13-3引脚锁定：**R、G、B**分别接**PIO60、PIO61、PIO63**；**HS、VS**分别接**PIO64、PIO65**；**CLK**接**clock9 (13MHz)**，**MD**接**PIO0**，控制显示模式。

接上VGA显示器，选择模式5，下载**COLOR.SOF**；控制键1，观察显示器工作（如果显示不正常，将GW48系统右侧开关拨以下，最后再拨回到“TO_MCU”）。

(3) 实验内容2：设计可显示横彩条与棋盘格相间的VGA彩条信号发生器。

(4) 实验内容3：设计可显示英语字母的VGA信号发生器电路。

(5) 实验内容4：设计可显示移动彩色斑点的VGA信号发生器电路。



实验与设计

13-2. VGA图像显示控制器设计

- (1) **实验内容1**: 根据图13-4和程序例13-2/3, 设计与生成图象数据; 根据例13-2中元件的接口, 定制放置图象数据的ROM。
- (2) **实验内容2**: 硬件验证例13-2/3, 选择模式5, 引脚连接方式仍同图13-4, 只是时钟输入clk50MHz接clock0, 选择频率50MHz的时钟信号。在EDA系统上接上VGA显示器, 下载后观察图形显示情况。
- (3) **实验内容3**: 为此设计增加一个键, 控制输出图象的正色与补色。
- (4) **实验内容4**: 为了显示更大的图象, 用外部ROM取代FPGA的内部ROM, 即元件, 电路结构参考图13-4, 引脚锁定参考电路结构图NO.5图中的ROM 27C020/27C040与FPGA的引脚连接情况。示例程序可下载 ./VGA88/vgarom.sof, 或 ./VGA88/vgarom.sof, clock0接50MHz, GW48 EDA系统左下角的拨码开关的“ROM使能”拨向下(如果显示不正常, 将EDA系统右侧开关拨以下, 最后再拨回到“TO_MCU”)。选择模式5, 键1控制图象的正色与补色显示。注意, 实验结束后将拨码开关的“ROM使能”拨向上还原。



实验与设计

13-3. 步进电机细分驱动控制实验

(1) 实验目的：学习用FPGA实现步进电机的驱动和细分控制。

(2) 实验内容1：完成以图13-7所示的步进电机控制电路的验证性实验。首先引脚锁定：步进电机的4个相：Ap、Bp、Cp、Dp（对应程序中的Y0、Y1、Y2、Y3）分别与PIO65、PIO64、PIO63、PIO62（见GW48主系统左侧的标注）相接。

CLK0接clock0，选择4Hz；CLK5接clock5，选择32768Hz；S接PIO6（键7），控制步进电机细分旋转（1/8细分，2.25度/步），或不细分旋转（18度/步）；U_D接PIO7（键8），控制旋转方向。

用短路帽将系统左侧的“步进允许(JM0)”短路（注意，电机实验结束后，短路帽插回“禁止”端！

选择模式No.5，用Quartus下载step_1c3中的step_a.sof到EP1C3中，观察电机工作情况。

给出电机的驱动仿真波形，与示波器中观察到的电机控制波形进行比较。



实验与设计

13-3. 步进电机细分驱动控制实验

(3) 实验内容2: 设计2个电路: 1、要求能按给定细分要求, 采用PWM方法, 用FPGA对步进电机转角进行细分控制(利用QuartusII的EAB在系统编辑器实时在系统编辑调试ROM3中的细分控制数据); 2、用FPGA实现对步进电机的匀加速和匀减速控制。

(4) 实验内容3: 为使步进电机能平稳地运行, 并尽快从起点到达终点, 步进电机应按照以下控制方式运行: 启动→匀加速→匀速→匀减速→停止。当给定终点位置(转角)以后, 试用FPGA实现此控制。

(5) 实验内容4: 步进电机在步距角8细分的基础上, 试通过修改控制电路对步距角进一步细分。



实验与设计

13-4. 直流电机PWM控制实验

(1) 实验目的：学习直流电机PWM的FPGA控制。掌握PWM控制的工作原理，对直流电机进行速度控制、旋转方向控制、变速控制。

(2) 实验内容1：完成以图13-14所示的直流电机控制电路的验证性实验。首先引脚锁定：

直流电机模块中的MA2、MA1（对应程序中的Z、F）分别与EP1C3的PIO60/61相接，用于控制直流电机；测直流电机转速的MA-CNT端接PIO66，即CNTT端（见主系统左侧的标注）；

用短路帽分别将主系统左侧的“直流允许（JM1）”和“计数允许（JM2）”短路；CLK5接clock5，选择32768Hz；F1HZ接clock2，选择1Hz，作为转速测量的频率计的门控时钟；

键1（PIO0，接Z_F）控制旋转方向；键2（PIO1，D_STP）控制旋转速度。连续按动此键时，由数码管7显示0、1、2、3指示4个速度级别；转速由数码管4、3、2、1显示。

选择模式No.5，用Quartus下载step_1c3中的step_a.sof到EP1C3中，观察电机工作情况。

给出电机的驱动仿真波形，与示波器中观察到的电机控制波形进行比较。

(3) 实验内容2：实现直流电机的闭环控制，旋转速度可设置。

(4) 实验内容3：说明图13-14中的去抖动电路的工作原理，为了加强去抖动效果，改进图13-14中的去抖动电路和工作时钟频率的选择，如设计一个含4个D触发器的去抖动电路，实测它的性能。并说明此电路的工作时钟频率与被测信号频率的关系。