习题解答

11.1　什么是 VHDL？它具有那些特点？

答：

11.2　一个完整的 VHDL 程序包括哪几部分？其中哪些部分是可以单独编译的源设计单元？

答：

11.3　一个设计实体由哪几部分组成？它们的作用如何？

答：

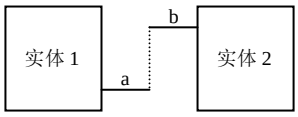
11.4　VHDL 数据对象中常量、变量和信号的实际物理含义是什么？变量和信号的主要区别有哪些？

答：

11.5　判断 VHDL 的如下操作是否正确，若不正确，请修改。

（1）字符串 op1 和 op2 数据类型是 bit，result 是整数，做 VHDL 程序加法 result <=op1+op2；

（2）实体 1 的输出 a 是无符号数 unsigned，实体 2 的输入 b 是实数，两个实体能否按题图 11.5 所示相连接。



题图 11.5

答：(1)的操作是错误的。因为 VHDL 是一种强类型语言，要求运算关系与赋值关系中各量必须有确定的数据类型，并且相同的数据类型的量才能相互传递和作用。(1)中可将 op1 和 op2 改为整数(integer)数据类型或将 result 改为位(bit)数据类型。

（2）两个实体不能按题图 11.5 所示相连接，因为 a 和 b 的数据类型不一致。

11.6　端口模式 in 和 inout 有什么不同？out 和 buffer 有什么区别?

答：

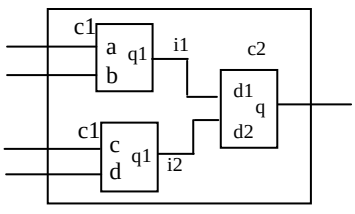
11.7　什么是库？在 VHDL 中有哪几种库？如何使一个库对于设计可见？

答：

11.8　VHDL 为什么提出程序包的结构？在 VHDL 中有哪几种程序包？如何使程序包中的内容对于设计可见？

答：

11.9　用 VHDL 描述题图 11.9 所示的方框图，即在元件 top 中例化元件 c1（两次）和元件 c2。



题图 11.9

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY top IS
PORT(a,b,c,d:IN STD_LOGIC;
     Q:OUT STD_LOGIC);
END top;
ARCHITECTURE a OF top IS
  SIGNAL i1,i2:STD_LOGIC;
  COMPONENT c1
  PORT(a:IN STD_LOGIC;
```

```
        b:IN STD_LOGIC;
        q1:OUT STD_LOGIC);
    END COMPONENT;
    COMPONENT c2
    PORT(d1:IN STD_LOGIC;
        d2:IN STD_LOGIC;
       q:OUT STD_LOGIC);
    END COMPONENT;
    BEGIN
    U1:c1 PORT MAP(a,b,i1);
    U2:c1 PORT MAP(c,d,i2);
    U3:c2 PORT MAP(i1,i2,q);
END a;
```

11.10 请分析下面两个进程，然后回答问题。

```
p1: process(a,b,c)
    variable d :std_logic;
    begin
       d:=a;
       x<=b+d;
       d:=c;
       y<=b+d;
    end process p1;
p2: process(a,b,c,d)
    begin
       d<=a;
       x<=b+d;
       d<=c;
       y<=b+d;
    end process p2;
```

（1）进程 1 执行后 x 和 y 的结果是什么？

（2）进程 2 执行后 x 和 y 的结果是什么？

（3）根据（1）和（2）中的结果，你可以得出什么结论？

答：（1）进程 1 执行后 x=b+a，y=b+c

（2）进程 2 执行后 x<=b+c，y<=b+c

（3）由进程 1 和进程 2 得出信号量的值将进程语句最后所代入的值作为最终值代入。变量的值一经赋值就变成新的值。

11.11 设计带异步置位/复位功能的 JK 触发器的 VHDL 程序。

设复位端子为 reset，置位端子为 set，低电平有效，JK 触发器输出端子为 q 和 qb，程序为：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY jkdff IS
PORT(clk:IN STD_LOGIC;
     reset,set:IN STD_LOGIC;
     j,k:IN STD_LOGIC;
     q,qb:OUT STD_LOGIC);
```

```
END jkdff;
ARCHITECTURE a OF jkdff IS
  SIGNAL q_s,qb_s:STD_LOGIC;
  BEGIN
  PROCESS(reset,set,clk,j,k)
  BEGIN
    IF(reset=‘1’)AND(set=‘0’)THEN
      q_s<=‘1’;qb_s<=‘0’;
    ELSIF(reset=‘0’)AND(set=‘1’)THEN
      q_s<=‘0’;qb_s<=‘1’;
    ELSIF(clk’EVENT AND clk=‘1’)THEN
       IF(k=‘1’)AND(j=‘0’)THEN
         q_s<=‘0’;qb_s<=‘1’;
       ELSIF(j=‘1’)AND(k=‘0’)THEN
         q_s<=‘1’;qb_s<=‘0’;
       ELSIF(j=‘1’)AND(k=‘1’)THEN
         q_s<=NOT q_s;qb_s<=NOT qb_s;
       END IF;
    END IF;
  q<=q_s;
  qb<=qb_s;
 END PROCESS;
END a;
```

11.12 用 VHDL 语言设计一个 4 位二进制并行加法器，再用两个 4 位二进制并行加法器组成 8 位二进制加法器。

四位加法器的低位进位为 cin，加数为 a,被加数为 b,和为 s,进位为 cout,四位加法器的 VHDL 程序为：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY adder4b IS
PORT(cin:IN STD_LOGIC;
     a,b:IN STD_LOGIC_VECTOR(3 DOWNTO 0);
     s:OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
     cout:OUT STD_LOGIC);
END adder4b;
ARCHITECTURE a OF adder4b IS
  SIGNAL sint:STD_LOGIC_VECTOR(4 DOWNTO 0);
  SIGNAL aa,bb:STD_LOGIC_VECTOR(4 DOWNTO 0);
  BEGIN
   aa<=‘0’&a;
   bb<=‘0’&b;
   sint<=aa+bb+cin;
```

```
    s<=sint(3 DOWNTO 0);
    cout<=sint(4);
 END a;
```

八位加法器的低位进位为 cc，加数为 aa,被加数为 bb,和为 ss,进位为 f,由两个四位二进制并行加法器级联而成的 8 位二进制加法器的 VHDL 程序为：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY adder8b IS
PORT(cc:IN STD_LOGIC;
     aa,bb:IN STD_LOGIC_VECTOR(7 DOWNTO 0);
     ss:OUT STD_LOGIC_VECTOR(7 DOWNTO 0);
     f:OUT STD_LOGIC);
END adder8b;
ARCHITECTURE a OF adder8b IS
COMPONENT adder4b
PORT(cin:IN STD_LOGIC;
     a,b:IN STD_LOGIC_VECTOR(3 DOWNTO 0);
     s:OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
     cout:OUT STD_LOGIC);
END COMPONENT;
SIGNAL carry_out:STD_LOGIC;
 BEGIN
  u1:adder4b PORT MAP(cin=>cc,a=>aa(3 DOWNTO 0),b=>bb(3 DOWNTO 0),
  s=>ss(3 DOWNTO 0),cout=>carry_out);
  u2:adder4b PORT MAP(cin=>carry_out,a=>aa(7 DOWNTO 4),b=>bb(7 DOWNTO 4),
  s=>ss(7 DOWNTO 4),cout=>f);
END a;
```

11.13 用 VHDL 语言实现四位通用移位寄存器 74LS194。

74194 是双向并入并出移位寄存器，它的内容可以根据控制输入信号的状态，向两个方向中的一个方向移动。这两个方向就是所谓的“左”和“右”。左移意味着“从 QD 移动 QA”，而右移意味着“从 QA 移到 QD”。下表为 4 位通用移位寄存器 74194 的功能表。

RIN 是右移位时的串行输入，LIN 是左移位时的串行输入。74194 的 VHDL 语言描述如下：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY V74194 IS
PORT( CLK,CLR,RIN,LIN: IN STD_LOGIC;
       A,B,C,D:IN STD_LOGIC;
       S:IN STD_LOGIC_VECTOR(1 DOWNTO 0);
       QA,QB,QC,QD:OUT STD_LOGIC);
END ;
```

```
ARCHITECTURE V74194_ARCH OF V74194 IS
SIGNAL IQ:STD_LOGIC_VECTOR(3 DOWNTO 0);
BEGIN
PROCESS(CLK,CLR)
BEGIN
IF CLR='0' THEN IQ<=(OTHERS=>'0');
ELSIF CLK'EVENT AND CLK='1' THEN
CASE S IS
    WHEN "00"=>NULL;
    WHEN "01"=>IQ<=RIN&IQ(3 DOWNTO 1);
    WHEN "10"=>IQ<=IQ(2 DOWNTO 0)&LIN;
    WHEN "11"=>IQ<=A&B&C&D;
    WHEN OTHERS=>NULL;
END CASE;
END IF;
QA<=IQ(3);QB<=IQ(2);QC<=IQ(1);QD<=IQ(0);
END PROCESS;
END ;
```

4 位通用移位寄存器 74194 的功能表

功能	输入		下一状态			
	S1	S0	QA*	QB*	QC*	QD*
保持	0	0	QA	QB	QC	QD
右移位	0	1	RIN	QA	QB	QC
左移位	1	0	QB	QC	QD	LIN
置数	1	1	A	B	C	D

11.14 用 VHDL 语言实现具有异步清零同步置数的模十计数器 74LS160。

74LS160 端子定义如下：CLK 为时钟输入，CLR 为异步清零端，

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY V74160 IS
PORT( CLK,CLR,LDN,ENT,ENP: IN STD_LOGIC;
       A:IN STD_LOGIC_VECTOR(3 DOWNTO 0);
       Q:OUT STD_LOGIC_VECTOR(3 DOWNTO 0);
       RCO:OUT STD_LOGIC);
 END ;
ARCHITECTURE V74160_ARCH OF V74160 IS
SIGNAL IQ:STD_LOGIC_VECTOR(3 DOWNTO 0);
BEGIN
PROCESS(CLK,CLR)
BEGIN
  IF CLR='0' THEN IQ<=(OTHERS=>'0');
    ELSIF (CLK'EVENT AND CLK='1') THEN
```

```
    IF LDN='0' THEN IQ<=A;
     ELSIF (ENT AND ENP)='1' THEN
      IF IQ=9 THEN IQ<="0000";
          ELSE IQ<=IQ+1;
      END IF;
    END IF;
  END IF;
  IF IQ=9 AND ENT='1' THEN RCO<='1';
   ELSE RCO<='0';
  END IF;
  Q<=IQ;
   END PROCESS;
  END ;
```

11.15 用 VHDL 语言设计一个有异步清零、同步置数的可逆 60 进制计数器。

程序中 clr 为异步清零端，set 为同步置位端，datain 为数据输入端，undp 为可逆计数器控制端，当 undp 为‘1’时，作加法计数器，当 undp 为‘0’时，作减法计数器。其 VHDL 程序为：

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY count60 IS
PORT (clk,set,clr:IN STD_LOGIC;
      undp:IN STD_LOGIC;
      datain:INTEGER RANGE 0 TO 60;
      count:OUT INTEGER RANGE 0 TO 60);
END count60;
ARCHITECTURE rt OF count60 IS
SIGNAL counter:INTEGER RANGE 0 TO 60;
BEGIN
  count<=counter;
  PROCESS(clk,clr)
     BEGIN
       IF clr= ‘1’ THEN counter<=0;
          ELSIF(clk ‘EVENT AND clk= ‘1’)THEN
              IF set= ‘1’ THEN
                 counter<=datain;
              ELSIF undp= ‘1’THEN
                 IF counter=59 THEN counter<=0;
                       ELSE counter<=counter+1;
                 END IF;
              ELSIF counter=0 THEN counter<=59;
                     ELSE counter<=counter-1;
              END IF;
```

```
            END IF;
    END PROCESS;
END rt;
```

11.16 按题表 11.16 给定花样用 VHDL 语言设计 8 路彩灯控制器。

题表 11.16

clk	花样	clk	花样
1	00011000	20	00000011
2	00111100	21	00000111
3	01111110	22	00001111
4	11111111	23	00011111
5	01111110	24	00111111
6	00111100	25	01111111
7	00011000	26	11111111
8	00000000	27	00000000
9	11111111	28	10000001
10	10000001	29	01000010
11	11000000	30	00100100
12	11100000	31	00011000
13	11110000	32	10011001
14	11111000	33	01011010
15	11111100	34	00111100
16	11111110	35	10111101
17	11111111	36	01111110
18	00000000	37	11111111
19	00000001	38	00000000

系统有三个输入端，分别为 en,clk 和 pul。当使能端 en 为 1 时，系统开始工作。

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
ENTITY clight IS
PORT(en,clk,pul:IN STD_LOGIC;
     d:OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
END clight;
ARCHITECTURE a OF clight IS
  SIGNAL sa:STD_LOGIC_VECTOR(3 DOWNTO 0);
  SIGNAL sd:STD_LOGIC_VECTOR(7 DOWNTO 0);
  SIGNAL mode:STD_LOGIC_VECTOR(1 DOWNTO 0);
  BEGIN
  PROCESS(pul)
    BEGIN
     IF(pul'EVENT AND pul= '1')THEN
       IF (mode="11") THEN
```

```
        mode<=“00”;
       ELSE
        mode<=mode+1;
       END IF;
     END IF;
END PROCESS;
PROCESS(clk)
BEGIN
  IF(clk'EVENT AND clk=‘1’)THEN
     IF(mode=“00”)THEN
      IF(sa=“0111”)THEN
         sa<=“0000”;
      ELSE
         sa<=sa+1;
      END IF;
     ELSIF(mode=“01”)THEN
       IF (sa=“0111”)THEN
         sa<=“0000”;
        ELSE
         sa<=sa+1;
       END IF;
     ELSIF(mode=“10”)THEN
        IF(sa=“1000”)THEN
           sa<=“0000”;
        ELSE
           sa<=sa+1;
        END IF;
      ELSIF(mode=“11”)THEN
       IF(sa=“1001”)THEN
          sa<=“0000”;
       ELSE sa<=sa+1;
       END IF;
    END IF;
  END IF;
 END PROCESS;
 PROCESS(en,sa)
 BEGIN
   IF(en=‘1’)THEN
     IF (mode=“00”)THEN
       IF(sa=0)THEN
        sd<=“00011000”;
       ELSIF(sa=1)THEN
        sd<=“00111100”;
       ELSIF(sa=2)THEN
```

```
        sd<=“01111110”;
       ELSIF(sa=3)THEN
        sd<=“11111111”;
       ELSIF(sa=4)THEN
        sd<=“01111110”;
       ELSIF(sa=5)THEN
        sd<=“00111100”;
       ELSIF(sa=6)THEN
        sd<=“00011000”;
       ELSIF(sa=7)THEN
        sd<=“00000000”;
       ELSE
        sd<=“11111111”;
       END IF;
 ELSIF(mode=“01”)THEN
     IF(sa=0)THEN
        sd<=“10000000”;
       ELSIF(sa=1)THEN
        sd<=“11000000”;
       ELSIF(sa=2)THEN
        sd<=“11100000”;
       ELSIF(sa=3)THEN
        sd<=“11110000”;
       ELSIF(sa=4)THEN
        sd<=“11111000”;
       ELSIF(sa=5)THEN
        sd<=“11111100”;
       ELSIF(sa=6)THEN
        sd<=“11111110”;
       ELSIF(sa=7)THEN
        sd<=“11111111”;
       ELSE
        sd<=“00000000”;
       END IF;
ELSIF(mode=“10”)THEN
     IF(sa=0)THEN
        sd<=“00000001”;
       ELSIF(sa=1)THEN
        sd<=“00000011”;
       ELSIF(sa=2)THEN
        sd<=“00000111”;
       ELSIF(sa=3)THEN
        sd<=“00001111”;
       ELSIF(sa=4)THEN
```

```
        sd<=“00011111”;
       ELSIF(sa=5)THEN
        sd<=“00111111”;
       ELSIF(sa=6)THEN
        sd<=“01111111”;
       ELSIF(sa=7)THEN
        sd<=“11111111”;
       ELSE
        sd<=“00000000”;
       END IF;
ELSIF(mode=“11”)THEN
     IF(sa=0)THEN
        sd<=“10000001”;
       ELSIF(sa=1)THEN
        sd<=“01000010”;
       ELSIF(sa=2)THEN
        sd<=“00100100”;
       ELSIF(sa=3)THEN
        sd<=“00011000”;
       ELSIF(sa=4)THEN
        sd<=“10011001”;
       ELSIF(sa=5)THEN
        sd<=“01011010”;
       ELSIF(sa=6)THEN
        sd<=“00111100”;
       ELSIF(sa=7)THEN
        sd<=“10111101”;
       ELSIF(sa=8)THEN
        sd<=“01111110”;
       ELSIF(sa=9)THEN
        sd<=“11111111”;
       ELSE
         sd<=“00000000”;
       END IF;
    ELSE
        sd<=“11111111”;
       END IF;
       d<=sd;
    END IF;
  END PROCESS;
 END a;
```