石川工業高等専門学校　電子情報工学科　システム設計演習　前期中間テスト

番号	氏名 秋田 純一

1．次のような論理回路の VHDL 記述で、それぞれの process 文のセンシティビティ・リスト(sensitivity list)に記述するべき信号名を過不足なく記せ。（１０点×４）

(40)

```
entity hoge is
  port (
    a, b, c : in std_logic;
    w, x, y, z : out std_logic
  );
end hoge;

architecture arch of hoge is
begin
  process ([1]) begin
    case a is
      when '1'   => w <= a;
      when '0'   => w <= b and c;
    end case;
  end process;

  process ([2]) begin
    case b is
      when '1'   => x <= '1';
      when '0'   => x <= '0';
    end case;
  end process;
 （右上へ続く）
```

```
（左下から続く）
 process ([3]) begin
   case a is
     when '1'   => y <= x;
     when '0'   => y <= z;
   end case;
 end process;

 process ([4]) begin
   case a is
     when '1'   => z <= b;
     when '0'   => z <= not b;
   end case;
 end process;
end arch;
```

[1]：a, b, c

[2]：b

[3]：a, x, z

[4]：a, b

※部分点なし

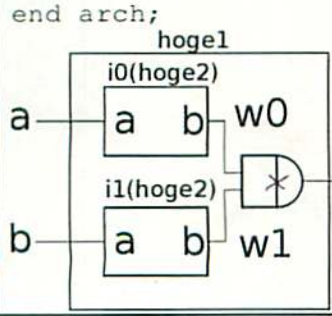
(10)×4

2．次のような VHDL 記述された論理回路の論理ブロック（インスタンス：四角で囲って図示）の名称や論理ゲートの接続関係を、例にならって必要であれば論理ゲートと各信号線の名称とともに図示せよ。（１０点）

(10)

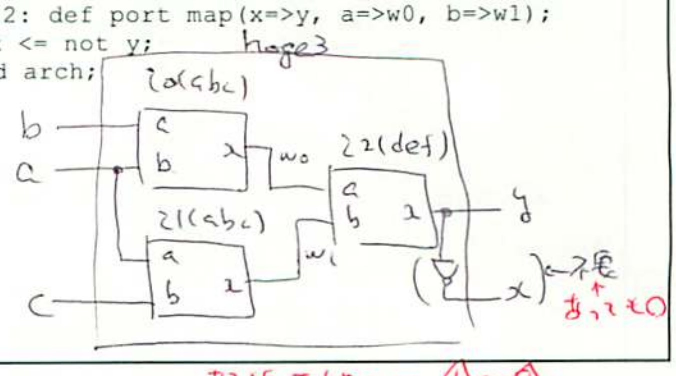
例：

```
entity hoge1 is
  port (a, b: in_std_logic;
       x: out std_logic);
end hoge1;

architecture arch of hoge1 is
  component hoge2
    port (a: in std_logic,
          b: out std_logic);
  end component;
  signal w0, w1: std_logic
begin
  i0: hoge2 port map(a=>a, b=>w0);
  i1: hoge2 port map(a=>b, b=>w1);
  x <= w0 and w1;
end arch;
```




```
entity hoge3 is
  port (a, b, c: in std_logic;
       y: out std_logic));
end hoge3;

architecture arch of hoge3 is
  component abc
    port (a, b: in std_logic, x: out std_logic);
  end component;
  component def
    port (a, b: in std_logic, x: out std_logic);
  end component;
  signal w0, w1: std_logic;
begin
  i0: abc port map(b=>a, a=>b, x=>w0);
  i1: abc port map(a=>a, b=>c, x=>w1);
  i2: def port map(x=>y, a=>w0, b=>w1);
  x <= not y;
end arch;
```

※記述に不備あり
↓
※出力x, yまわりは基本としてすべて○



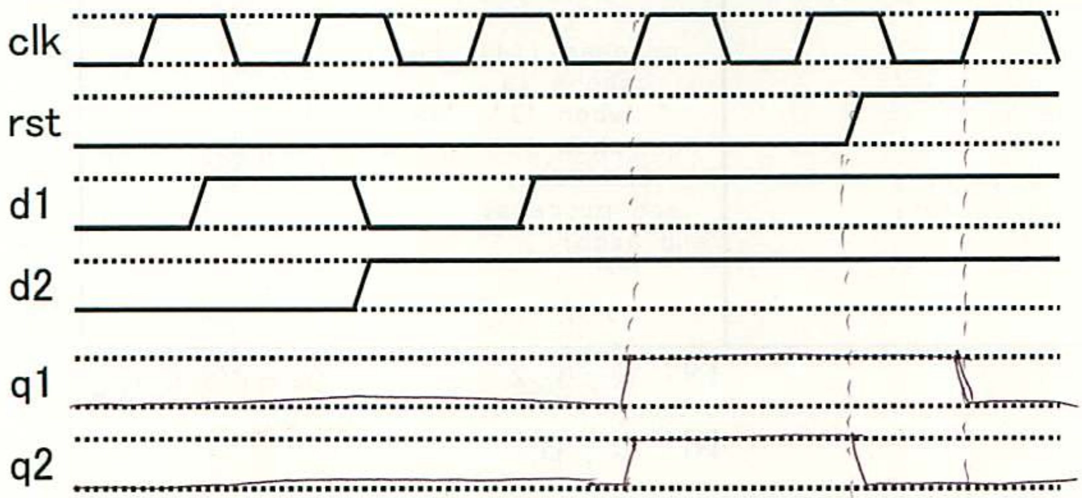
←不備 ↑あってる○

配線ちがい → ①〜②

3．次のような２種類の回路 hoge1, hoge2 に以下の図のような信号を与えたときの出力 q1, q2 を図示せよ。ただし q1, q2 の初期値は'0'とする。また q1, q2 の変化のタイミングを、対応する信号のタイミングと結ぶ縦の点線で明示すること。（１５点×２）

```
entity hoge1 is
  port (clk, rst, d1, d2: in std_logic;
        q1: out std_logic));
end hoge1;

architecture arch of hoge1 is
begin
  process (clk, rst) begin
    if (clk'event and clk = '1') then
      if (rst = '1') then q1 <= '0';
      else q1 <= d1 and d2;
      end if;
    end if;
  end process;
end arch;
```

```
entity hoge2 is
  port (clk, rst, d1, d2: in std_logic;
        q2: out std_logic));
end hoge2;

architecture arch of hoge2 is
begin
  process (clk, rst) begin
    if (rst = '1') then q2 <= '0';
    elsif if (clk'event and clk = '1') then
      q2 <= d1 and d2;
    end if;
  end process;
end arch;
```



q1, q2の ↑, ↓のタイミング 各 5×2 → 15×2

必部分点なし

4．講義で扱った10進カウンタの VHDL 記述を参考に、出力値が 0〜27 の範囲で順番に増える 28 進カウンタ count28 の VHDL 記述を完成させよ。ただし信号のビット数は必要十分な数とし、リセット動作は同期リセットとする。（２０点）

```
entity count28 is
  port (clk, rst, dir: in std_logic;
        q: out std_logic_vector( 4 downto 0));
end count28;

architecture arch of count28 is
  signal q_reg : std_logic_vector( 4 downto 0);
begin
  process (clk) begin
    if (clk'event and clk = '1') then
      if (rst = '1') then q_reg <= " 00000 ";
      else
        if (q-reg = 27) then  q-reg <= "00000";
        else
          q-reg <= q-reg + 1 ;

      end if;
    end if;
  end process;
  q <= q_reg;
end arch;
```

動作が正しければ 0 〜 20
ビット数ちがい → -5 〜 -3
文法まちがい → -1 〜 -10
単なるカウンタ → -5