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）

```
entity hoge1 is
  port (
    a, b, c, d, e: in std_logic;
    x: out std_logic
  );
end hoge1;

architecture arch of hoge1 is
begin
  process ([1]) begin
    case a is
      when '1'   => x <= a;
      when '0'   => x <= b and c;
    end case;
  end process;

  process ([2]) begin
    case b is
      when '1'   => y <= (a and d) or e;
      when '0'   => y <= '0';
    end case;
  end process;
end arch;
```

[1]：a, b, c
[2]：a, b, d, e

```
entity hoge2 is
  port (
    a, b: in std_logic_vector(1 dowonto 0);
    x, y: out std_logic
  );
end hoge2;

architecture arch of hoge2 is
begin
  process ([3]) begin
    case a is
      when "01"  => x <= '0';
      when "10"  => x <= '1';
      when others => x <= b(1);
    end case;
  end process;

  process ([4]) begin
    case b(0) is
      when '0'  => y <= a(0);
      when '1'  => y <= a(1);
    end case;
  end process;
end arch;
```

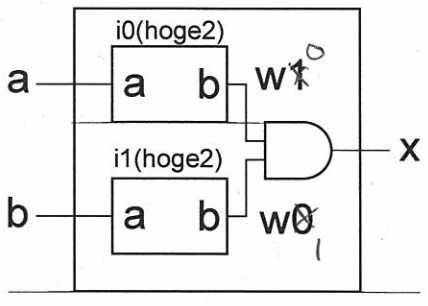
[3]：b (a or b(1)), a
[4]：a, b (a or b(0))

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）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。（２０点）

例：

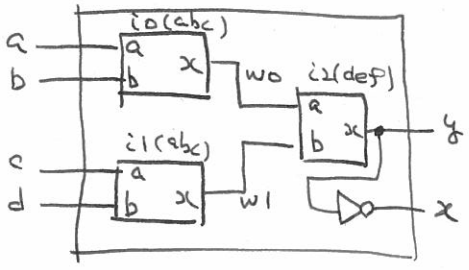
```
entity hoge1 is
  port (a, b: in_std_logic;
        x: out std_logic);
end hoge1;

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, d: in std_logic;
        x, 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(a=>a, b=>b, x=>w0);
  i1: abc port map(a=>c, b=>d, x=>w1);
  i2: def port map(a=>w0, b=>w1, x=>y);
  x <= not y;
end arch;
```



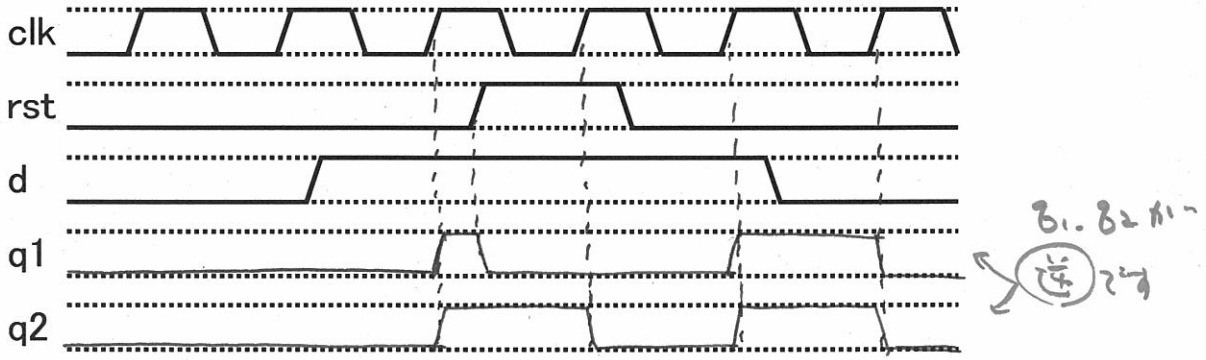
3．次のような 2 種類の回路 hoge1, hoge2 に以下の図のような信号を与えたときの出力 q1, q2 を図示せよ。ただし q1, q2 の初期値は’0’とする。また q1, q2 の変化のタイミングを、対応する信号のタイミングと結ぶ縦の点線で明示すること。（１０点×２）

```
entity hoge1 is
  port (clk, rst, d: 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 <= d;
      end if;
    end if;
  end process;
end arch;
```

```
entity hoge2 is
  port (clk, rst, d: 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 <= d;
    end if;
  end process;
end arch;
```




q1. q2 が 逆 です

4．次の 8 ビットカウンタ count8 の VHDL 記述を参考に、クロックの立ち上がりごとに、入力 dir の値に応じてカウンタの値が 1 ずつ増加(dir=0 のとき)または減少(dir =1 のとき)する回路（アップダウンカウンタ：同期リセットつき）ud_count8 の VHDL 記述を完成させよ。（２０点）

```
entity count8 is
  port (clk, rst: in std_logic;
        q: out std_logic_vector(7 downto 0));
end count8;

architecture arch of count8 is
  signal q_reg : std_logic_vector(7 downto 0);
begin
  process (clk) begin
    if (clk'event and clk = '1') then
      if (rst = '1') then q_reg <= "00000000";
      else q_reg <= q_reg + 1;
    end if;
  end process;
  q <= q_reg;
end arch;
```

```
entity ud_count8 is
  port (clk, rst, dir: in std_logic;
        q: out std_logic_vector(7 downto 0));
end ud_count8;

architecture arch of ud_count8 is
  signal q_reg : std_logic_vector(7 downto 0);
begin
  process (clk) begin
    if (clk'event and clk = '1') then
      if (rst = '1') then q_reg <= "00000000";
      elsif (dir = '0') then
        q_reg <= q_reg + 1;
      else
        q_reg <= q_reg - 1;
      end if;
    end if;
  end process;
  q <= q_reg;
end arch;
```