

番号	氏名
	秋田 悠一

※教科書、自筆ノート、配布プリント（自筆の書き込み含む）のみ持ち込み可です。

1. 次のような論理回路のVHDL記述で、それぞれの process 文のセンシティブティ・リスト(sensitivity list)に記述すべき信号名を過不足なく記せ。(5点×4)

<pre>entity hoge is port (a, b, c, d : 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 b => w <= '0'; when c => w <= '1'; end case; end process; process ([2]) begin if (b = '0') then x <= not a; else x <= a; end if; end process; (右上へ続く)</pre>	<p>(左下から続く)</p> <pre>process ([3]) begin if (c'event and c = '1') then y <= d; end if; end process; process ([4]) begin case (a & b) is when "00" => z <= c and d; when "01" => z <= c or d; when "10" => z <= c xor d; when "11" => z <= not c; when others => z <= '0'; end case; end process; end arch;</pre>
---	--

[1]: a, b, c (5) × 4

[3]: ~~a, b~~ c, d ← cのみでOK

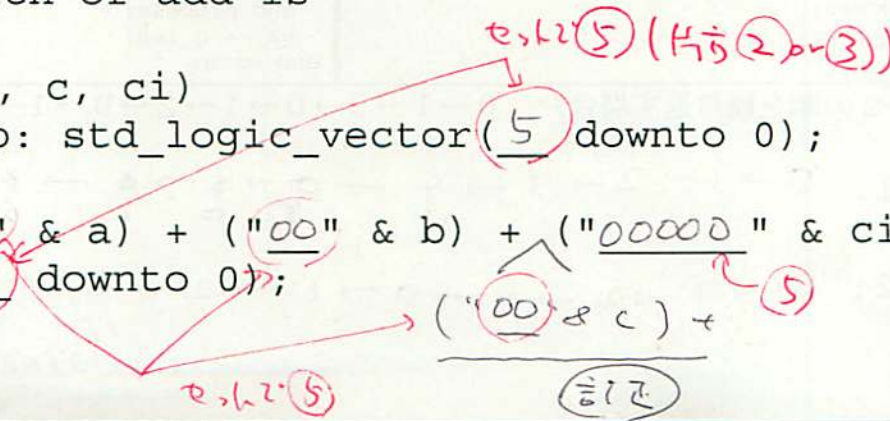
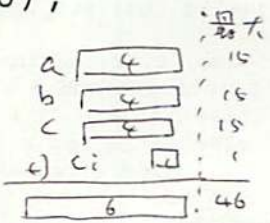
[2]: a, b

[4]: a, b, c, d

2. 授業で扱った8ビット加算器の動作記述を参考に、4ビットの3つの数 a, b, c と1ビットの桁上がり入力 ci の和 x を求める回路のVHDL記述を、以下の下線部に適切な過不足ない数字を埋めて完成させよ。(20点)

```
entity add is
  port (
    a, b, c: in std_logic_vector(3 downto 0);
    ci: in std_logic;
    x: out std_logic_vector(5 downto 0)
  );
end add;

architecture arch of add is
begin
  process (a, b, c, ci)
    variable tmp: std_logic_vector(5 downto 0);
  begin
    tmp := ("00" & a) + ("00" & b) + ("00000" & ci);
    x <= tmp(5 downto 0);
  end process;
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。
(15点×2)

```
entity hoge1 is
  port (c, r, p, d: in std_logic;
        q1: out std_logic);
end hoge1;

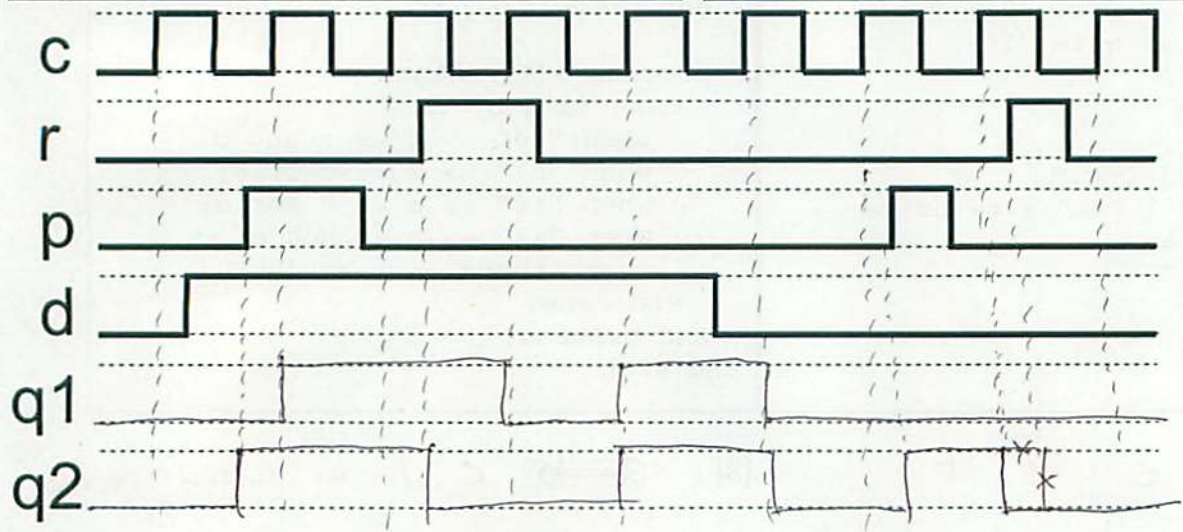
architecture arch of hoge1 is
begin
  process (c, r, p) begin
    if (c'event and c = '1') then
      if (r = '1') then q1 <= '0';
      elsif (p = '1') then q1 <= '1';
      else q1 <= d;
      end if;
    end if;
  end process;
end arch;
```

同期P/V

```
entity hoge2 is
  port (c, r, p, d: in std_logic;
        q2: out std_logic);
end hoge2;

architecture arch of hoge2 is
begin
  process (c, r, p) begin
    if (r = '1') then q2 <= '0';
    elsif (p = '1') then q2 <= '1';
    elsif if (c'event and c = '1') then
      q2 <= d;
    end if;
  end process;
end arch;
```

非同期P/V



(15)×2
(適宜書き込み)

4. 次のような2種類の回路 count1, count2 で、リセット動作(r=1)した後の、クロックの立ち上がりごとの出力 q1, q2 の値を10進数表記した値の変化を、例のように矢印で続けて10クロック分示せ。(15点×2点)

```
entity count1 is
  port (c, r: in std_logic;
        q1: out std_logic_vector(2 downto 0));
end count1;

architecture arch of count1 is
  signal q_reg: std_logic_vector(2 downto 0);
begin
  process (c, r) begin
    if (c'event and c = '1') then
      else if (r = '1') then q_reg <= "000";
      else if (q_reg = 5) q_reg <= 0;
      else q_reg <= q_reg + 1;
      end if;
    end if;
  end process;
  q1 <= q_reg;
end arch;
```

```
entity count2 is
  port (c, r: in std_logic;
        q2: out std_logic_vector(1 downto 0));
end count2;

architecture arch of count2 is
  signal q_reg: std_logic_vector(1 downto 0);
begin
  process (c, r) begin
    if (c'event and c = '1') then
      else if (r = '1') then q_reg <= "00";
      else if (q_reg = 5) q_reg <= 0;
      else q_reg <= q_reg + 1;
      end if;
    end if;
  end process;
  q2 <= q_reg;
end arch;
```

(例: 0~2の順を繰り返す場合) 0→1→2→0→1→2→0→1→2→0

count1: 0→1→2→3→4→5→0→1→2→3

count2: 0→1→2→3→0→1→2→3→0→1

11.6.1.0.2.1
7.12.7.6.2.5
2.0.2
1.7.4.2.0
ok.

(15)×2