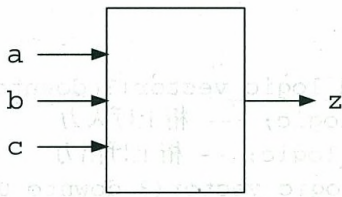


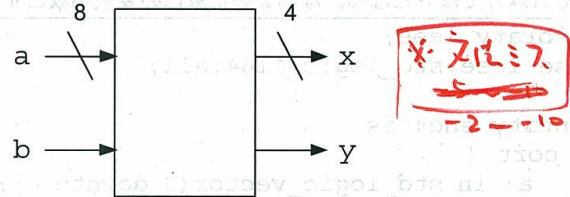
番号	氏名 秋田純一
----	----------------

1. 次のような入出力を持つ回路を VHDL で記述する際のエンティティ記述を完成させよ。なお回路の名称はブロック図の上に記されている。(10点×2)

(1) and3



(2) hoge



```
entity and3 is
  port (
    a, b, c: in std_logic;
    z: out std_logic
  );
end and3;
```

```
entity hoge is
  port (
    a: in std_logic_vector(7 downto 0);
    b: in std_logic;
    x: out std_logic_vector(3 downto 0);
    y: out std_logic
  );
end hoge;
```

2. 2つの入力 a, b から、s で指定したものを出力 x とするセレクタを 2通りの方法によって VHDL で記述した。ただし一方の動作は正しくない。両者にそれぞれの下に示したグラフのような入力を与えた場合に予想されるそれぞれの出力を、それぞれの下グラフに示せ。(10×2点)

```
library ieee;
use ieee.std_logic_1164.all;

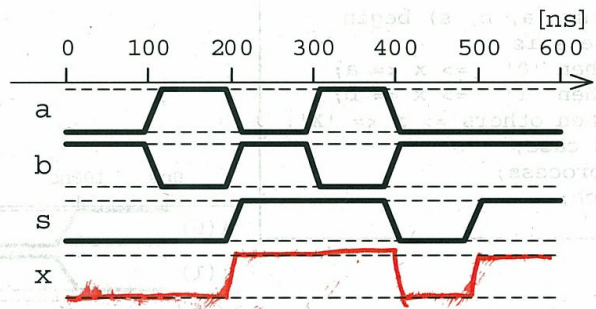
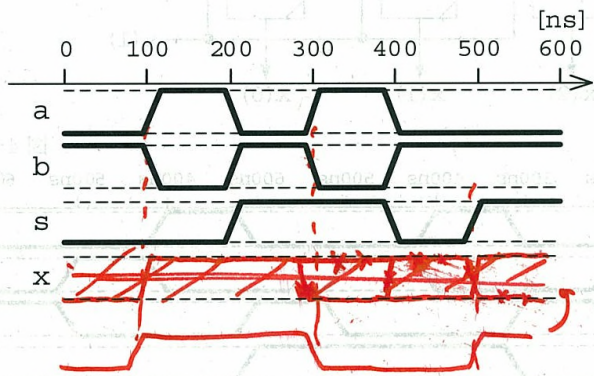
entity sel1 is
  port (
    a, b: in std_logic;
    s: in std_logic;
    x: out std_logic
  );
end sel1;

architecture arch of sel1 is
begin
  process (a, b, s) begin
    case s is
      when '0' => x <= a;
      when '1' => x <= b;
      when others => x <= 'X';
    end case;
  end process;
end arch;
```

```
library ieee;
use ieee.std_logic_1164.all;

entity sel2 is
  port (
    a, b: in std_logic;
    s: in std_logic;
    x: out std_logic
  );
end sel2;

architecture arch of sel2 is
begin
  process (s) begin
    case s is
      when '0' => x <= a;
      when '1' => x <= b;
      when others => x <= 'X';
    end case;
  end process;
end arch;
```



3. 教科書や授業で扱った回路の VHDL 記述を参考に、指定された機能をもつ回路の VHDL 記述を、欠けている箇所を補って完成させよ。(20点×2)

(1) 4ビットプライオリティエンコーダ enc4

※入力の優先度は a(0)が最も高く、以下順に a(3)が最も低い
 ※1つ以上の入力がある場合は出力 x の最上位は 1 となり、
 それ以外では 0 となる。なお x の桁数は必要最小数とする

```

library ieee;
use ieee.std_logic_1164.all;

entity enc4 is
  port (
    a: in std_logic_vector(3 downto 0);
    x: out std_logic_vector(2 downto 0)
  );
end enc4;

architecture arch of enc4 is
begin
  process (a) begin
    if (a(0) = '1')
      then x <= "100";
    elsif (a(1) = '1')
      then x <= "101";
    elsif (a(2) = '1')
      then x <= "110";
    elsif (a(3) = '1')
      then x <= "111";
    else
      x <= "000";
    end if;
  end process;
end arch;

```

* 62x24 - 5
 7x24 - 10 - 5
 10 - 5

(2) 4ビット桁上げ先見加算器 cla4

```

library ieee;
use ieee.std_logic_1164.all;

entity cla4 is
  port (
    a, b: in std_logic_vector(3 downto 0);
    ci: in std_logic; -- 桁上げ入力
    co: out std_logic; -- 桁上げ出力
    s: out std_logic_vector(3 downto 0) -- 和
  );
end cla4;

architecture arch of cla4 is
  signal g0, g1, g2, g3; -- 生成項
  signal p0, p1, p2, p3; -- 伝搬項 (XOR 型)
  signal c0, c1, c2, c3; -- 各桁の桁上げ
begin
  g0 <= a(0) and b(0); p0 <= a(0) xor b(0);
  g1 <= a(1) and b(1); p1 <= a(1) xor b(1);
  g2 <= a(2) and b(2); p2 <= a(2) xor b(2);
  g3 <= a(3) and b(3); p3 <= a(3) xor b(3);
  c0 <= g0 or (p0 and ci);
  c1 <= g1 or (p1 and c0);
  c2 <= g2 or (p2 and c1);
  c3 <= g3 or (p3 and c2);
  co <= c3;
  s(0) <= p0 xor ci; s(1) <= p1 xor c0;
  s(2) <= p2 xor c1; s(3) <= p3 xor c2;
end arch;

```

* a(0) → a0, b0 - 5
 ci 0, 1 - 5
 c0 - 3x
 c1 - 10
 c2 - 5
 c3 - 5

4. 以下のような VHDL 記述による回路 hoge を図 4-1 のように接続した回路に対して、図 4-2 のような入力を与えた場合に得られる出力を図 4-2 中に示せ。(20点)

```

library ieee;
use ieee.std_logic_1164.all;

entity hoge is
  port (
    a, b, s, in std_logic;
    x: out std_logic
  );
end hoge;

architecture arch of hoge is
begin
  process (a, b, s) begin
    case s is
      when '0' => x <= a;
      when '1' => x <= b;
      when others => x <= 'X';
    end case;
  end process;
end arch;

```

