

番号

解答用

氏名

秋田 純一

1. 授業で扱った例や教科書の例を参考に、次の回路のVHDL記述を完成させよ。(10点×2)
- (1) Dフリップフロップ (入出力はクロック(clk)、データ(d)、出力(q)とし、リセット機能はもたない)
- (2) 4ビットパリティジェネレータ (教科書の例のようにfunctionを用いる任意ビット数対応でなく、4ビット固定でよい)

```
entity dff is
port (
    clk, d: in std_logic;
    q: out std_logic;
);
end dff;

architecture arch of dff is
begin
    process(clk) begin
        if (clk'event and clk = '1') then
            q <= d;
        end if;
    end process;
end arch;
```

*Handwritten notes:*  
 ☆ 入出力は多ビットの場合 → 矢印  
 ☆ process あり → 矢印  
 ☆ 矢印ミス: 矢印

```
entity parity4 is
port (
    a: in std_logic_vector(3 downto 0);
    x: out std_logic;
);
end parity4;

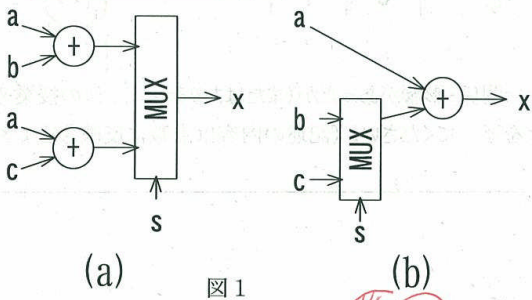
architecture arch of parity4 is
begin
    x <= a(0) xor a(1) xor a(2) xor a(3);
end arch;
```

*Handwritten notes:*  
 ☆ function 提供, 任意ビット対応. 矢印ミスに注意  
 ☆ 矢印ミス: 矢印

2. 以下の記述からは、それぞれ図1・図2に示すように2通りの回路が合成される可能性がある。(1)を参考に、図2(b)中の空欄を埋めよ。また、加算器のシェアリングが行われる、それぞれの右側の回路(図1(b)・図2(b))を明示的に得ることができると思われるVHDL記述を完成させよ。(20点×2)

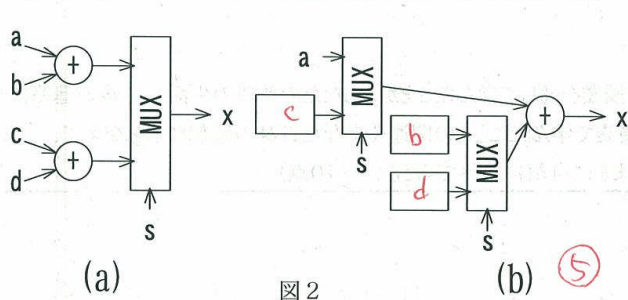
(1)

```
process (a, b, c, s) begin
    if (s = '1') then x <= a + b;
    else x <= a + c;
    end if;
end process;
```



(2)

```
process (a, b, c, d, s) begin
    if (s = '1') then x <= a + b;
    else x <= c + d;
    end if;
end process;
```



```
signal t: std_logic;
process (a, b, c, s) begin
    if (s = '1') then t <= b;
    else t <= c;
    end if;
end process;
x <= a + t;
```

```
signal t1, t2: std_logic;
process (a, b, c, d, s) begin
    if (s = '1') then
        t1 <= a; t2 <= b;
    else
        t1 <= c; t2 <= d;
    end if;
end process;
x <= t1 + t2;
```

☆ 矢印ミス: 矢印  
 ☆ MUXの記述が非明示的  
 (1) 矢印  
 (2) 矢印  
 ☆ 空欄埋められ  
 ☆ Signal 追加  
 矢印



