

番号	氏名 秋田 純
----	---------

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)、リセット(rst)とし、非同期リセット機能をもつ)

(2) 8ビット ALU (算術演算ユニット)。ただし s=00, 01, 10, 11 のときに、出力 q が、それぞれ順に a+b (算術和)、a-b、a・b (論理積)、a+b (論理和) とする。

10 x2
変換ミス
A-B

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

architecture arch of dff is
begin
    process (clk, rst) begin
        if (rst = '1') then
            q <= '0';
        elsif (clk'event and clk = '1') then
            q <= d;
        end if;
    end process;
end arch;
```

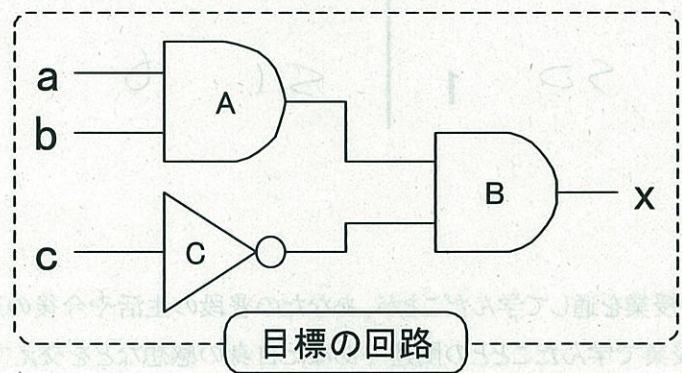
```
entity alu8 is
port (
    a, b: in std_logic_vector(7 downto 0);
    s : in std_logic_vector(1 downto 0);
    q: out std_logic_vector(7 downto 0)
);
end alu8;

architecture arch of alu8 is
begin
    process (a, b, s) begin
        case s is
            when "00" => q <= a + b;
            when "01" => q <= a - b;
            when "10" => q <= a and b;
            when "11" => q <= a or b;
        end case;
    end process;
end arch;
```

2. 以下のような3種類のエリア(回路規模)と遅延時間(いずれも単位なし)をもつ論理ゲートを用いて、以下の「目標の回路」をつくる際に、それぞれの制約条件を満たすために用いる論理ゲートの記号の組を記し、その場合のエリアと遅延時間を求めよ。例えば「目標の回路」中の論理ゲートA、B、Cに、それぞれ順に「AND-1」と「AND-2」と「NOT-1」を使う場合は、「a, b, d」と記せ。なお回路全体の遅延時間は、各入力から出力に至る経路上の各論理ゲートの遅延時間の和のうち、最大の値と定義し、同一論理ゲートを複数回用いてもよい。また制約を満たす回路を作ることが不可能である場合には、「×」を記せ。(5点×4)

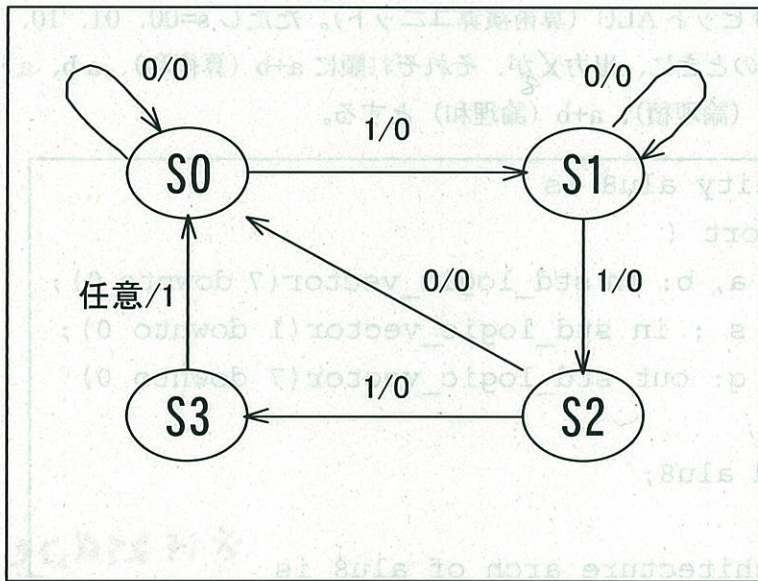
5 x4

	エリア	遅延時間	記号
AND-1	2	10	a
AND-2	4	5	b
AND-3	20	1	c
NOT-1	2	6	d
NOT-2	10	1	e



	用いる論理ゲートの組	エリア	遅延時間
(1) 最もエリアの小さい回路	a, a, d	6	20
(2) 最も遅延時間の短い回路	c, c, e	50	2
(3) エリアが10以下、遅延時間が10以下	×	×	×
(4) エリアが35以下、遅延時間が12以下	{ a, a, e b, c	34	7
	{ c, a, e	32	11
	{ b, b, d	10	11

3. 次のような状態遷移図をもつステートマシンの VHDL 記述を完成させよ。なおこのステートマシンは1つの入力 a と、1つの出力 x をもつ。また状態遷移図において、遷移をあらわす矢印に示されている2つの値は、左側が遷移の条件の入力 a の値、右側が出力 x の値を表わす。なお s0_st~s3_st は、それぞれ状態 S0~S3 を表す状態コードとする。またこのステートマシンに、図のような入力をあたえたときの状態遷移表を完成させよ。ただし、ある遷移の「次状態」が、次の行の遷移での「現状態」となり、順に遷移が続くとする。(50点)



現状態	入力	次状態	出力
S0	0	S0	0
S0	1	S1	0
S1	0	S1	0
S1	1	S2	0
S2	0	S0	0
S2	1	S0	0

```

process (clk, rst) begin
  if (rst = '1') then state <= S0_ST;
  elsif (clk'event and clk = '1') then
    case state is
      when S0_ST =>
        if (s = '1') then state <= S1_ST;
        else state <= S0_ST;
        end if;
      when S1_ST =>
        if (s = '1') then state <= S2_ST;
        else state <= S1_ST;
        end if;
      when S2_ST =>
        if (s = '1') then state <= S3_ST;
        else state <= S0_ST;
        end if;
      when S3_ST => state <= S0_ST;
    end case;
  end if;
end process;

```

```

process (state, a) begin
  case state is
    when S0_ST => g <= '0';
    when S1_ST => g <= '0';
    when S2_ST => g <= '0';
    when S3_ST => g <= '1';
  end case;
end process;

```

state/next-state
2行14行構成
2行15行の process
1に含む構成
for: 変数 +
正しく4行
文法正
← A → B
next-state → state
の5行14行
for: 構成が5行
→ A → B

4. この授業を通して学んだことが、あなたの普段の生活や今後の進路にどのような関係・影響があったか(またはありそうか)、他の授業やこれまでの授業で学んだこととの関連やあなた自身の感想などを交えて、自由に考えを述べてください。(記述の内容は点数に反映させませんので、思うままに自由に述べてください)(10点)

基本的に10