

番号	氏名 秋田 悠一
----	----------

1. 7進カウンタ (クロック信号 `clk` の立ち上がりごとに、出力 `q` (3ビット) が 0 から 6 まで 1 ずつ増え、その後再び 0 に戻る) `count7` を、ステートマシンとして設計した VHDL 記述を、不足している箇所 (下線部) を補って完成させよ。ただしリセット信号 `rst` によるリセット動作は同期リセットとする。またエンティティ記述は省略してある。(30点)

```
architecture arch of count7 is
  signal st, next_st: std_logic_vector(2 downto 0);
begin
  process (rst, clk) begin -- state transition
    if (clk'event and clk='1') then
      if (rst = '1') then st <= "000";
      else st <= next_st;
      end if;
    end process;

    process (st) begin -- next state generation
      case st is
        when "000" => next_st <= "001";
        when "001" => next_st <= "010";
        when "010" => next_st <= "011";
        when "011" => next_st <= "100";
        when "100" => next_st <= "101";
        when "101" => next_st <= "110";
        when "110" => next_st <= "000";
        when others => next_st <= "000";
      end case;
    end process;
    q <= st;
  end arch;
```

②
8/5 ④ x 7

2. 入力された 3 ビットの数 `a` に対して、その 2 の補数 `x` (3 ビット) を出力する回路 `comp2` の VHDL 記述を、不足している箇所 (下線部) を補って完成させよ。ただしエンティティ記述は省略してある。(20点)

```
architecture arch of comp2 is
begin
  process (a) begin
    case a is
      when "000" => x <= "000";
      when "001" => x <= "111";
      when "010" => x <= "110";
      when "011" => x <= "101";
      when "100" => x <= "100";
      when "101" => x <= "011";
      when "110" => x <= "010";
      when "111" => x <= "001";
    end case;
  end process;
end arch;
```

2.5 x 8
(1 bit に 6 bit だけ)

3. 講義で扱った「タイマ」のデータパスの設計例を参考に、以下のような仕様のデータパス datapath の VHDL 記述を、下線部に必要な記述を補って完成させよ。(40点)

- 仕様
- ・ 入力 : op(4bit)、imm(4bit)、z(1bit)
 - ・ 出力 : addr(4bit)
 - ・ 機能 : op="0110"、または、「op="0111"かつ z=1」のときは、addr に imm を代入する (分岐)。それ以外の場合は、現在の addr に 1 加える (次命令フェッチ)。なおこれらの動作はすべてクロック信号 clk の立ち上がりで行われ、rst=1 で addr は clk とは非同期に 0 にリセットされる。

```

entity datapath is
  port( clk, rst, z : in std_logic;
        op, imm      : in std_logic_vector(3 downto 0);
        addr         : out std_logic_vector(3 downto 0));
end datapath;

architecture Behavioral of datapath is
  signal addr_out, addr_in: std_logic_vector(3 downto 0);
begin
  process (op) begin
    if (op = "0110") then addr_in <= imm;
    elsif (op = "0111" and z = '1') then addr_in <= imm;
    else addr_in <= addr_out + "0001";
    end if;
  end process;

  process (clk, rst) begin
    if (rst = '1') then addr_out <= "0000";
    elsif (clk'event and clk = '1') then addr_out <= addr_in;
    end if;
  end process;

  addr <= addr_out;
end Behavioral;

```

} 10
 } 10
 } 8
 } 2
 } 10

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 10 + d
 真の感想は A~A