電子工学科 実験報告書

実験題目：　VHDL によるディジタル回路の設計 (自由課題)
担当教員：　木場 隼介 先生
実験開始日：　令和 5 年 10 月 19 日
実験終了日：　令和 5 年 11 月 2 日
提出日：　令和 5 年 11 月 7 日
再提出日：

学年：　5 年
出席番号：
実験班：　B 班
氏名：　河合 将暉

共同実験者名：

コメント欄

1　目的

自由課題を通して VHDL の構文規則や処理方法について詳しく知るとともに、プレゼンテーションを行い、課題に対する説明および発表能力を養うことを目的とする。

2　自由課題

2.1　仕様

本実験の自由課題のコンセプトとして、2 人で対戦できるルーレットを FPGA で構成することを目標とした。仕様としては、FPGA デバッグボードに搭載されている 10 個の LED のうち、左右から 3 個ずつの LED を用いてルーレットを 2 組構成した。ルーレットのストップ・リセットには FPGA ボードに標準搭載されているタクトスイッチ 4 個を使用して 1 人あたりストップ・リセット用に 2 個スイッチを割り当てた。

対戦のルールとして、ルーレットが揃う (LED3 個が同色になる) と 1 点加点され、先に 2 点獲得したプレイヤーの勝利というようなルールを提案する。設計はこのルールをベースに設計を行ったが、ルール変更による拡張性にも視野に含めて設計した。対戦型ルーレットシステム稼働時の FPGA ボードを図 1 に示す。

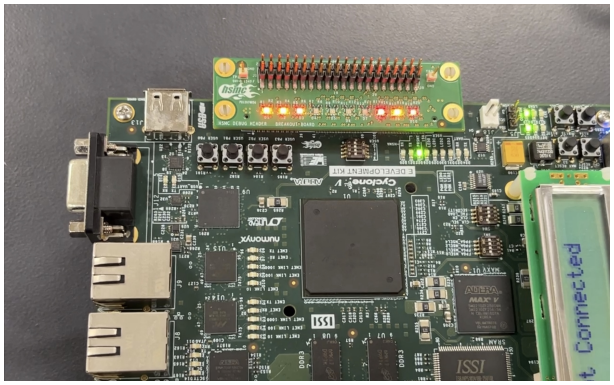

図 1: FPGA ボード上面図

2.2　参考にしたプログラム

自由課題の対戦型ルーレットシステムを構成する際に [1] の Sample08vhd を参考に作成した。Sample08 ではデバックボード上の LED を 3 個、FPGA ボード上のタクトスイッチを 2 個用いてルーレットを構成されていた。また、Sample08 ではルーレット機能のみを記述した roulettevhd をポートとして呼び出し、マッピングを行ってルーレット機能を動作させていた。

2.3　使用器具

以下に、本課題で使用した器具を表 1 に示す。

表 1: 使用器具

No	機器名	型番	シリアル No	備考
1	FPGA ボード	Cyclone V E FPGA Development Kit	2	シリアル No は外箱の番号を記載
2	PC	ASUS TAF-Gaming		

3 プログラム解説

3.1 ピン割当

本課題でのデバッグボード上 LED のピン割当を表 2 に示す。

表 2: デバッグボード LED のピン割当

ピン名称	入出力	ピン番号
led_out0[0]	出力	PIN_AF21
led_out0[1]	出力	PIN_AJ20
led_out0[2]	出力	PIN_AG22
led_out0[3]	出力	PIN_AK20
led_out1[0]	出力	PIN_AF20
led_out1[1]	出力	PIN_AJ19
led_out1[2]	出力	PIN_AG21
led_out1[3]	出力	PIN_AK18
led_out2[0]	出力	PIN_AF18
led_out2[1]	出力	PIN_AJ17
led_out2[2]	出力	PIN_AF19
led_out2[3]	出力	PIN_AJ18
led_out3[0]	出力	PIN_AG18
led_out3[1]	出力	PIN_AG24
led_out3[2]	出力	PIN_AG19
led_out3[3]	出力	PIN_AH25
led_out4[0]	出力	PIN_AK16
led_out4[1]	出力	PIN_AH19
led_out4[2]	出力	PIN_AK17
led_out4[3]	出力	PIN_AH20
led_out5[0]	出力	PIN_AF16
led_out5[1]	出力	PIN_AG17
led_out5[2]	出力	PIN_AG16
led_out5[3]	出力	PIN_AH17
led_out6[0]	出力	PIN_AE16
led_out6[1]	出力	PIN_AJ15
led_out6[2]	出力	PIN_AF15
led_out6[3]	出力	PIN_AK15
led_out7[0]	出力	PIN_AD17
led_out7[1]	出力	PIN_AH14
led_out7[2]	出力	PIN_AE17
led_out7[3]	出力	PIN_AH15
led_out8[0]	出力	PIN_AD18
led_out8[1]	出力	PIN_AE15
led_out8[2]	出力	PIN_AE18
led_out8[3]	出力	PIN_AF14
led_out9[0]	出力	PIN_Y15
led_out9[1]	出力	PIN_AG23
led_out9[2]	出力	PIN_AA15
led_out9[3]	出力	PIN_AH22

本課題での FPGA ボード上スイッチのピン割当を表 3 に示す。

表 3: FPGA ボードのピン割当

部品名	ピン名称	入出力	ピン番号
LED	led_check1	出力	
LED	led_check2	出力	
スイッチ	sw_in1	入力	PIN_AB12
	sw_in2	入力	PIN_AG12
	resetn1	入力	PIN_AB13
	resetn2	入力	PIN_AF13
クロック発振器	clk	入力	PIN_P22

3.2 実装した機能

3.2.1 ルーレットの独立化

本課題で実装したシステムの独自性として、サンプルプログラムで構成されていたルーレットは 1 つのみであり、複数個を同時に動作させることは不可能であったが、本システムでそれを可能にした。単純に LED を増設しただけではストップ・リセットの動作が全て同期したままであり、揃える LED が増えただけの大きなルーレットが構成されてしまう。これを解消するために、[1] から参照したサンプルプログラムの Sample08vhd と roulettevhd を一部変更した。まず、Sample08 におけるポート設定の変更箇所をソースコード 1 に示す。

ソースコード 1: Sample08 のポート設定

```
1       entity sample08 is
2       generic (
3         div_bits : integer := 15);
4
5       port (
6         clk : in std_logic;
7         resetn1 : in std_logic;
8         resetn2 : in std_logic;
9         sw_in1 : in std_logic;
10        sw_in2 : in std_logic;
11        led_check1 : out std_logic :='1';
12        led_check2 : out std_logic :='1';
13        led_out0, led_out1, led_out2, led_out3, led_out4, led_out5 ,led_out6, led_out7, led_out8
              , led_out9 : out std_logic_vector(3 downto 0));
14
15        end sample08;
```

ルーレット機能を複製するために resetn,sw_in を 2 つずつ用意し、led_out を 10 個用意した。ルーレットを実装した段階では led_out は 0〜2,7〜9 の 6 個分しか使用していないが今後のシステム拡張を想定し、デバッグボード上のすべての LED に割当を行った。

次に、roulettevhd におけるルーレット機能のポート設定をソースコード 2 示す。

ソースコード 2: roulettevhd::ルーレット機能のポート設定

```
1       entity roulette is
2
3       port (
4         clk : in std_logic;
5         resetn : in std_logic;
6         sw_in1 : in std_logic;
7         sw_in2 : in std_logic;
8         led_out : out std_logic_vector(3 downto 0));
9
10      end roulette;
```

サンプルプログラムでは、sw_in は 1 個しかなかったため、2 個用意し、どちらからの入力にもルーレットをストップさせる割当にした。

Sample08 でのルーレット機能のマッピングをソースコード 3 に示す。

ソースコード 3: Sample08vhd::ルーレット機能のマッピング

```
      component roulette

      port (
        clk : in std_logic;
        resetn  : in std_logic;
        sw_in1 : in std_logic;
        sw_in2  : in std_logic;
        led_out : out std_logic_vector(3 downto 0));

      end component;

      signal clk_div, clk_roulette, sw_node1, sw_node2 : std_logic := '0';
      signal div_counter : std_logic_vector(22 downto 0) := (others => '0');
      signal sw0, sw1, sw2, sw3, sw4, sw5, sw6, sw7, sw8, sw9: std_logic;
      signal sw_latch1, sw_latch_on1,sw_latch2, sw_latch_on2 : std_logic := '0';
      begin

      roulette_unit0 : roulette port map (
        clk => clk_roulette,
        resetn => resetn1,
        sw_in1 => sw0,
        sw_in2 => '0',
        led_out => led_out0);

      roulette_unit1 : roulette port map (
        clk => clk_roulette,
        resetn => resetn1,
        sw_in1 => sw1,
        sw_in2 => '0',
        led_out => led_out1);

      roulette_unit2 : roulette port map (
        clk => clk_roulette,
        resetn => resetn1,
        sw_in1 => sw2,
        sw_in2 => '0',
        led_out => led_out2);

      roulette_unit7 : roulette port map (
        clk => clk_roulette,
        resetn => resetn2,
        sw_in1 => '0',
        sw_in2 => sw7,
        led_out => led_out7);

      roulette_unit8 : roulette port map (
        clk => clk_roulette,
        resetn => resetn2,
        sw_in1 => '0',
        sw_in2 => sw8,
        led_out => led_out8);

      roulette_unit9 : roulette port map (
        clk => clk_roulette,
        resetn => resetn2,
        sw_in1 => '0',
        sw_in2 => sw9,
        led_out => led_out9);
```

roulette_unit0〜2 では sw_in1 の入力に合わせてルーレットをストップさせるため、sw_in1 の入力を sw0〜2 に代入した。また、sw_in2 の入力に影響を受けないように sw_in2 の入力を 0 にした。反対に、roulette_unit7〜9 では sw_in2 の入力に合わせて代入を行い、sw_in1 の入力を 0 にした。

ルーレットを止める機能のプログラムをソースコード**??**に示す。

ソースコード 4: sample08::ルーレットを止める機能

```
      -- スイッチによるパルスを受け取ってルーレットを止めていく回路
    process (clk, resetn1,resetn2)
    begin -- process
      if resetn1 = '0' then
        sw0 <= '1';
        sw1 <= '1';
        sw2 <= '1';

      elsif clk'event and clk = '1' then -- rising clock edge
        if sw_latch1 = '1' then
          if (sw0 = '1' and sw1 = '1' and sw2 = '1') then
            sw0 <= '0';
          elsif (sw0 = '0' and sw1 = '1' and sw2 = '1') then
            sw1 <= '0';
          elsif (sw0 = '0' and sw1 = '0' and sw2 = '1') then
            sw2 <= '0';
          end if;
        end if;
      end if;

      if resetn2 = '0' then
        sw7 <= '1';
        sw8 <= '1';
        sw9 <= '1';
      elsif clk'event and clk = '1' then -- rising clock edge
        if sw_latch2 = '1' then
          if (sw7 = '1' and sw8 = '1' and sw9 = '1') then
```

```
28              sw7 <= '0';
29            elsif (sw7 = '0' and sw8 = '1' and sw9 = '1') then
30              sw8 <= '0';
31            elsif (sw7 = '0' and sw8 = '0' and sw9 = '1') then
32              sw9 <= '0';
33            end if;
34          end if;
35          end if;
36      end process;
```

スイッチ入力 (sw_latch) が 1 になる場合，つまりスイッチ入力があったときに sw0〜sw2 に 0 を代入することによって LED のルーレットをストップしている。sw7〜sw9 も同様の動作を行っている。

3.3 実装できなかった機能

3.3.1 得点表示機能

本課題では 2 点先取のゲームを想定していたため，ルーレットが 1 回揃った段階で FPGA ボード上の LED を点灯させ，現在の得点を表示する必要があると考えた。手法として，LED 出力に用いている led_out の値を参照して 3 つの LED が同じ出力値の場合に FPGA ボード上の LED 出力を HIGH にして実装を試みたが，led_out の値を参照して条件分岐をしようとすると変数型に関するエラーが出力された。

3.3.2 改善方法の検討

先述のエラーの原因は条件分岐として参照する値が出力型だったためと考えられる。改善するためにはポート設定を led_out : out std_logic…ではなく led_out : buffer std_logic…とすると入出力の両方で用いることができるため，エラーが出力されないと考えられる。

4 質疑回答

- 3 人対戦は可能か
  現在構成しているシステムでは FPGA ボードに搭載されているスイッチの数が足りないため、不可能である。

- LED を 3 個しか使用していない理由
  FPGA ボード上で 2 人のプレイヤーがスイッチを押そうとすると相手の手で遮られて 3 個以上は見えづらいためルーレットとしてある程度難しい 3 個で妥協している。

- プレイヤー表示を下の LED の色分けで実装できないか
  FPGA ボード上の LED は単色 LED で緑色しか表示できないため、その方式では実装できない。デバックボード上では LED が 4 個余っているため、その LED を用いれば可能である。

- リセットボタンを用意するのではなくストップボタンの 4 回目でリセットにすればよいのではないか
  対戦人数を増やすという観点ではいい提案だと感じた。今回の課題の設計思想では、2 人対戦をメインとしており、ユーザビリティの観点でルーレットが揃わなかった際にボタンを数回連打するのと 1 回別のボタンを押すのでは別のボタンを押したほうがすぐにルーレットをやり直すことができてユーザビリティが高いと感じたため、ストップとリセットのスイッチを分けて実装した。

参考文献

[1]「実験実習指導書『各種計算ハードウェアの活用〜VHDL によるディジタル回路の設計〜』」
神戸高専電子工学科 pp.28-36