

POLITECHNIKA WROCŁAWSKA

UKŁADY CYFROWE I SYSTEMY WBUDOWANE

Dokumentacja projektu
Organy z możliwością zapisywania i odtwarzania
melodii.

Autorzy:

Łukasz BIESZCZAD

Krzysztof BUCZAK

Prowadzący:

dr inż. Jarosław SUGIER

18 maja 2018

Spis treści

1	Wprowadzenie	2
1.1	Cel i zakres	2
1.2	Zagadnienia teoretyczne	2
1.3	Sprzęt	2
2	Projekt	3
2.1	Schemat i hierarchia projektu	3
2.2	Moduły	4
2.2.1	Modulator2	4
2.2.2	Synth2	8
2.2.3	Switch	8
2.2.4	RAM	8
2.2.5	MsgGenerator	8
3	Implementacja	8
3.1	Zasoby	8
3.2	"User manual" urządzenia	8
4	Podsumowanie	8
5	Literatura	8

1 Wprowadzenie

1.1 Cel i zakres

Celem projektu było stworzenie jednooktawowego "instrumentu" klawiszowego, obsługiwanego za pomocą klawiatury PS/2. Wciskanie poszczególnych klawiszy miało powodować odtwarzanie dźwięków przez podpięty do pinów głośniczek. Dodatkowo częścią zadania było także zaimplementowanie funkcjonalności nagrywania melodii (zapis do pamięci ROM) i odtwarzania nagranych materiału, a także wykorzystanie wyświetlacza LCD do pokazania stanu nagrywania i diody LED do przekazania informacji o trwającym właśnie nagrywaniu.

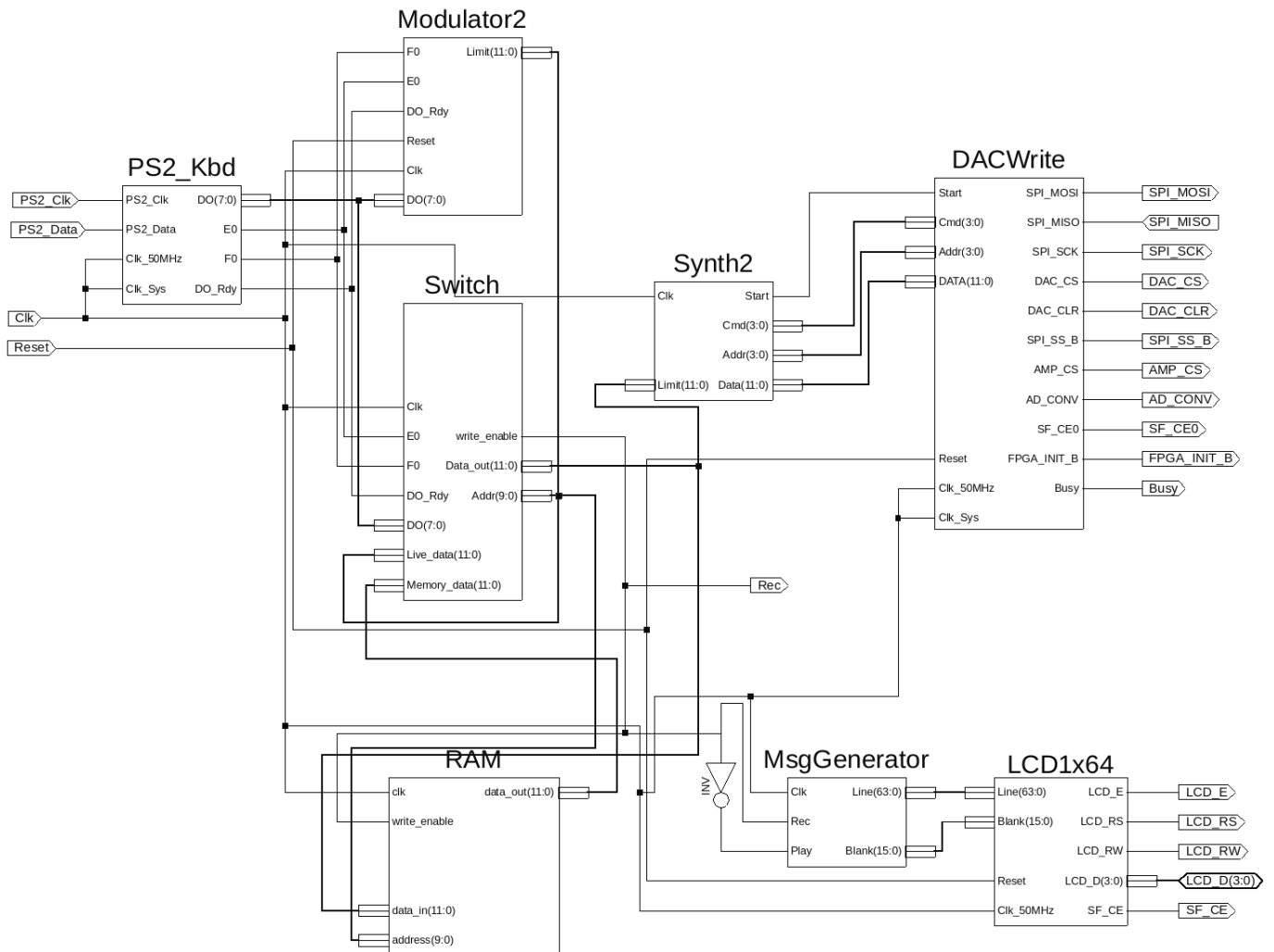
1.2 Zagadnienia teoretyczne

1.3 Sprzęt

Językiem projektu był język opisu sprzętu VHDL. Stanowisko laboratoryjne/projektowe zostało wyposażone w układ Spartan-3E oraz komputer z oprogramowaniem Xilinx ISE, pozwalającym kompilować kod VHDL pod dostarczony sprzęt, a także wykonywać symulacje testujące działanie systemu. Wykorzystaliśmy także port PS/2 (klawiatura symulująca keyboard), piny do podłączenia głośnika, ekran LCD wyświetlający stan nagrywania (pozostały czas) i układ pamięci ROM do przechowywania nagranych melodii.

2 Projekt

2.1 Schemat i hierarchia projektu



Rysunek 1: Schemat całego projektu.

Hierarchia modułów:

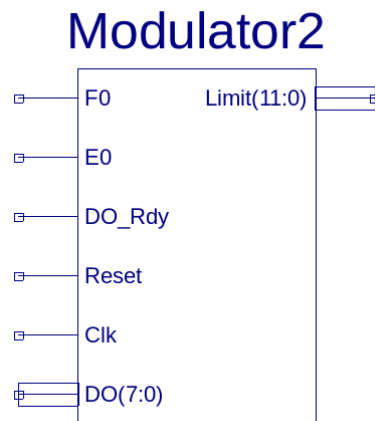
- schemat
 - DACWrite
 - Modulator2
 - PS2_Kdb
 - Synth2
 - Switch
 - RAM
 - LCD1x64
 - MsgGenerator
 - ADC_DAC.ucf
 - GenIO.ucf
 - LCD.ucf

2.2 Moduły

W tym podrozdziale znajdują się opisy i symulacje modułów stworzonych na zajęciach w ramach projektu.

2.2.1 Modulator2

Ten moduł odpowiedzialny jest za wysyłanie wartości ograniczającej licznik w module Synth2 na podstawie sygnału pochodzącego z klawiatury. Jest oparty na maszynie stanów, która składa się 14 stanów odpowiadających różnym dźwiękom w oktawie lub ciszy.



Rysunek 2: Symbol modułu Modulator2.

Wejścia modułu:

- F0 - symbolizuje zwolnienie klawisza klawiatury
- E0 - symbolizuje czy dane to tzw. kod rozszerzony
- DO_Rdy - symbolizuje zakończenie odbierania kodu
- Clk - symbolizuje zegar
- DO[7:0] - symbolizuje otrzymane dane z klawiatury

Wyjścia modułu:

- limit - symbolizuje wartość graniczną dla licznika

Fragmenty kodu VHDL

```
1 architecture Behavioral of Modulator2 is
2     type state_type is (Silence , C, Cis , D, Dis , E, F, Fis , G, Gis , A, Ais , H, C2);
3     signal state , next_state : state_type;
4 begin
5
6     process1 : process( Clk )
7     begin
8         if rising_edge( Clk ) then
9             if Reset = '1' then
10                 state <= Silence;
11             else
12                 state <= next_state;
13             end if;
```

```

15         end if;
16     end process process1;

17 process2: process (state, DO, F0, DO_Rdy)
18 begin
19     next_state <= state;

21     if DO_Rdy = '1' and F0 = '0' then

23         case state is
24             when Silence =>
25                 if DO = X"1C" then
26                     next_state <= C;
27                 elsif DO = X"1D" then
28                     next_state <= Cis;
29                 elsif DO = X"1B" then
30                     next_state <= D;
31                 elsif DO = X"24" then
32                     next_state <= Dis;
33                 elsif DO = X"23" then
34                     next_state <= E;
35                 elsif DO = X"2B" then
36                     next_state <= F;
37                 elsif DO = X"2C" then
38                     next_state <= Fis;
39                 elsif DO = X"34" then
40                     next_state <= G;
41                 elsif DO = X"35" then
42                     next_state <= Gis;
43                 elsif DO = X"33" then
44                     next_state <= A;
45                 elsif DO = X"3C" then
46                     next_state <= Ais;
47                 elsif DO = X"3B" then
48                     next_state <= H;
49                 elsif DO = X"42" then
50                     next_state <= C2;
51                 end if;

53             when C =>
54                 next_state <= state;

55             when Cis =>
56                 next_state <= state;

57             when D =>
58                 next_state <= state;

59             when Dis =>
60                 next_state <= state;

61             when E =>
62                 next_state <= state;

63             when F =>
64                 next_state <= state;

65             when Fis =>
66                 next_state <= state;

67             when G =>
68                 next_state <= state;

69             when Gis =>
70                 next_state <= state;

71             when H =>
72                 next_state <= state;

73             when A =>
74                 next_state <= state;

75             when Ais =>
76                 next_state <= state;

77             when C2 =>
78                 next_state <= state;

79         end case;

```

```

81         when A =>
            next_state <= state;

83         when Ais =>
            next_state <= state;

85         when H =>
            next_state <= state;

87         when C2 =>
            next_state <= state;

89         end case;
91     elsif F0 = '1' then
93         next_state <= Silence;
95     end if;
96 end process process2;

97 with state select
98     Limit <= X"5D4" when C,
99             X"581" when Cis ,
100            X"532" when D,
101            X"4E7" when Dis ,
102            X"4A1" when E,
103            X"45E" when F,
104            X"41F" when Fis ,
105            X"3E4" when G,
106            X"3AC" when Gis ,
107            X"377" when A,
108            X"345" when Ais ,
109            X"316" when H,
110            X"2EA" when C2,
111            X"000" when others;

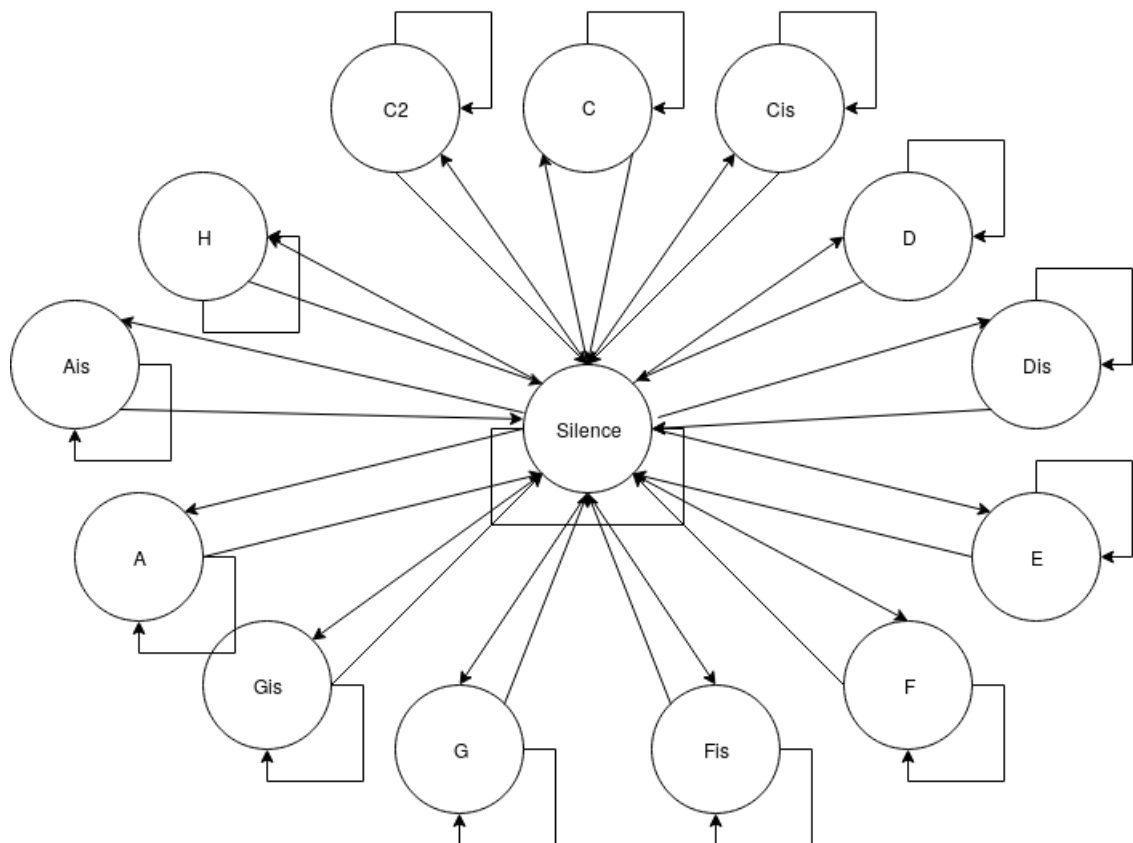
113 end Behavioral;

```

Listing 1: Procesy modułu Modulator2.

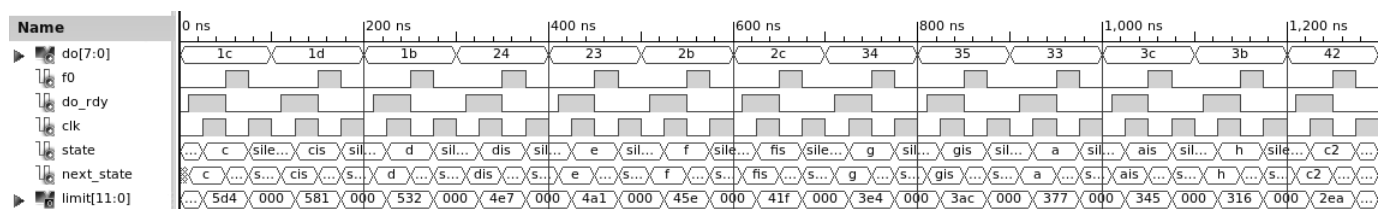
Architektura jednostki na samym początku posiada deklarację wszystkich stanów, które odpowiadają dźwiękom w oktawie lub ciszy. Pierwszy proces jest typowym przykładem procesu odpowiedzialnego za przełączanie stanu w maszynie stanów. Kolejny proces odpowiada za wybranie odpowiedniego stanu w zależności od wciśniętego klawisza. Ostatni element modułu to ustawianie odpowiedniego limitu w zależności od aktualnego stanu.

Graf maszyny stanu



Rysunek 3: Graf maszyny stanów modułu Modulator2.

Symulacja



Rysunek 4: Wyniki symulacji modułu Modulator2.

Na powyższej symulacji widać jak w momencie impulsu sygnału do_rdy z najbliższym taktom zegara zmieniany jest stan maszyny zgodnie z wciśniętym klawiszem oraz ustawiany jest odpowiedni limit. W momencie impulsu f0 (puszczenie klawisza) stan przełączany jest na silence, a limit na wartość zerową.

2.2.2 Synth2

2.2.3 Switch

2.2.4 RAM

2.2.5 MsgGenerator

3 Implementacja

3.1 Zasoby

3.2 "User manual" urządzenia

4 Podsumowanie

Zadanie udało się zrealizować w całości. Instrument jest w pełni działający, a ze względu na jasny podział na moduły można bez trudu dopisywać do niego kolejne funkcjonalności. Także sama wartość "merytoryczna" keyboarda nie pozostawia wiele do życzenia, ponieważ faktycznie pokrywa on całą oktawę, a wysokości dźwięków różnią się od siebie dokładnie tak jak w prawdziwym instrumencie, dzięki czemu mając nuty do utworu muzycznego możemy go zagrać.

5 Literatura