Data: 12.05.2021

**Politechnika Wrocławska**

Układy Cyfrowe i Systemy Wbudowane 2

Dokumentacja projektu:

Organy generujące fale piłokształtne z

wyświetlaniem informacji na kanale tekstowym

Skład grupy:

Artur Sobolewski 248913

Przemysław Rychter 248820

Prowadzący:

Dr inż. Jarosław Sugier

Spis treści

1. **Wprowadzenie**
   1. **Cel projektu**

Celem projektu było wykonanie układu realizującego organki generujące dźwięki w postaci fal piłokształtnych z zakresu jednej oktawy. Obsługa urządzenie miała odbywać się za pomocą klawiatury wysyłającej sygnały przez interfejs PS/2. Miało to pozwolić generować odpowiednie tony w zależności od naciśniętego klawisza. Za odtwarzanie dźwięków odpowiadałby podłączony do płyty głośniczek. Dodatkową formą komunikacji z użytkownikiem ma być wyświetlanie informacji dotyczącej wysokości i długości trwania dźwięku na kanale tekstowym VGA.

* 1. **Wykorzystany sprzęt i narzędzia**

Docelowym sprzętem do zaprogramowania była płyta FPGA Spartan-3E Starter Kit. Najważniejszą właściwością tej płyty, która ma wpływ na sposób generowania dźwięków o odpowiednich częstotliwościach jest oscylator zegarowy pracujący z częstotliwością 50 *MHz*, co daje długość okresu 20 *ns*. Kolejnymi elementami płyty Spartan-3E niezbędnymi do realizacji projektu są 2 porty PS/2 przeznaczone dla klawiatury lub myszy (wykorzystany zostanie tylko jeden z nich) oraz port VGA. Ponadto posiada ona też inne komponenty przydatne pod kątem tworzonego układu: moduł DAC – czterokanałowy konwerter sygnału cyfrowego na analogowy[[1]](#footnote-1). Wykorzystywanym środowiskiem było oprogramowanie Xilinx ISE, które pozwala na realizację kodu w języku opisu sprzętu VHDL oraz jego kompilację dostosowaną do podanej wcześniej płyty i końcowo wykonywać symulacje układu. Do projektu zostały przewidziane dodatkowe urządzenia: klawiatura wyposażona w interfejs PS2, głośniczek oraz wyświetlacz odbierający informacje z portu VGA.

* 1. **Wstęp teoretyczny**

W ramach projektu z portu PS/2 (XILINX, brak daty) przy podłączonej klawiaturze zostaną użyte 2 magistrale tego interfejsu: PS2\_DATA oraz PS2\_CLK. Obie wykorzystują 11-bitowe słowa zwierające znaczniki startu, stopu, bit parzystości. Pierwsza z nich przesyła dane dotyczące naciskanych klawiszy: kod skanowania klawisza, kod czy klawisz jest wciśnięty, czy zwolniony[[2]](#footnote-2).

Modułem, który posłużył do konwersji danych cyfrowych dźwięków na ich analogową reprezentację jest moduł DAC[[3]](#footnote-3). Zapewnia on 4 kanały wyjściowe do wyboru. Komunikacja z modułem odbywa się za pomocą szeregowego interfejsu peryferyjnego (SPI). W ramach projektu najważniejszym sygnałem jest SPI\_MOSI, za pomocą, którego przekazywane są wystarczające informacje definiujące charakterystykę fali dźwiękowej jaka ma być wygenerowana. Jest to 32-bitowe słowo. 4 najniższych i 8 najwyższych bitów są nieistotne, natomiast ze wzrostem pozycji kolejnych bitów: 12-bitowy ciąg jest wartością, od której zależy chwilowa wartość napięcia składająca się na falę dźwiękową; 4-bitowy ciąg określający docelowy kanał wyjściowy modułu DAC; i końcowo 4-bitowy ciąg wskazujący tryb pracy modułu – zazwyczaj jest to wartość „0011”, która oznacza automatyczne odświeżenie sygnału wyjściowego po otrzymaniu nowych danych.

1. **Realizacja projektu**
   1. **Top module schemat**

**TODO**

* 1. **Moduły składowe**
     1. **dac\_signal\_generator**

Moduł dac\_signal\_generator odpowiada za generowanie fali piłokształtnej i sygnałów sterujących, w formacie dostosowanym do danych jakie przyjmuje modułu DACWrite (fala - wartości liczb całkowitych w postaci wektora 12-bitoweg). Wartości przyjmowane na wektorze Key określają częstotliwość fali, jaka ma być generowana. Praca tego modułu dostosowana jest do częstotliwości z jaką pracuje wykorzystana płyta Spartan – 50 MHz – w oparciu o nią dobierane są opóźnienia w zmianie wartości wychodzącej, dzięki czemu otrzymujemy określone częstotliwości fal.

**Schemat dac_signal_generator

Opis wygenerowany automatycznie**

Rysunek Symbol modułu generującego cyfrowe fale piłokształtne

Wejścia:

* Rdy – pełni rolę sygnału sterującego, które powoduje pracę lub spoczynek modułu,
* Clk – sygnał zegarowy,
* Key(3:0) – wartości określające częstotliwość fali do wygenerowania,

Wyjścia:

* Start – impuls wysyłany do modułu DACWrite – informujący by ten przyjął przesyłane dane,
* Command(3:0) – wektor określający pracę modułu DACWrite,
* Address(3:0) – wektor określający, na który port wyjściowy ma być wysyłany sygnał analogowy z modułu DACWrite,
* Data(11:0) – wektor, którego określa wartość sygnału w danym czasie.

Opis realizacji modułu:

Na samym początku architektura modułu dac\_signal\_generator definiuje zmienne:

* clks\_delay – liczba określające ile taktów zegara należy odczekać pomiędzy kolejnymi zmianami wartości sygnału (bazowa wartość 999 – pozwala uzyskać częstotliwość równą 1 kHz). Konkretne częstotliwości są możliwe do uzyskania dzięki kontrolowaniu częstotliwości inkrementacji wartości przechowywanej w zmiennej var – przed każdą inkrementacją proces odczekuje zadaną liczbę taktów, których ilość jest zależna od wartości na wejściu Key. Zależność ta została stabelaryzowana w instrukcji with-select.
* clk\_counter – zmienna, która realizuje zliczanie taktów
* var – zmienna inkrementowana, której wartość przekazywana jest na wyjście Data.

W procesie został zaimplementowany licznik modulo działający w maksymalnym zakresie liczb całkowitych 12-bitowych. Wartość zwiększana jest o dziesiętną wartość 82, po zliczeniu wymaganej liczby okresów. Zależność liczby taktów zegara do odczekania od częstotliwości fali, jaka ma zostać wygenerowana wyznaczana była następująco:

Jeśli długość taktu sygnału zegarowego płyty: , liczba kolejnych wartości składających się na jeden okres fali piłokształtnej: , to długość okresu fali piłokształtnej:

,

Gdzie:

T – długość okresu,

x – liczba taktów zegara, jaką należy odczekać (we wzorze występuje + 1, ponieważ jest to dodatkowy takt potrzebny na inkrementację zmiennej)

Następnie jeśli: to po przekształceniach otrzymujemy, zależność od częstotliwości:

*Fragment 1 Kod VHDL modułu dac\_signal\_generator*

**entity** dac\_signal\_generator **is**

**Port** **(** Rdy **:** **in** STD\_LOGIC**;**

Clk **:** **in** STD\_LOGIC**;**

Key **:** **in** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

Command **:** **out** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

Address **:** **out** STD\_LOGIC\_VECTOR **(**3 **downto** 0**);**

Data **:** **out** STD\_LOGIC\_VECTOR **(**11 **downto** 0**);**

Start **:** **out** STD\_LOGIC**);**

**end** dac\_signal\_generator**;**

**architecture** Behavioral **of** dac\_signal\_generator **is**

**signal** clks\_delay **:** integer **:=** 999**;**

**signal** clk\_counter **:** integer **:=** 0**;**

**signal** var **:** unsigned **(**11 **downto** 0**)** **:=** X"000"**;**

**begin**

**with** Key **select** clks\_delay **<=**

955 **when** "0000"**,** -- 1046.50 c [ HZ ]

901 **when** "0001"**,** -- 1108.73 cis

850 **when** "0010"**,** -- 1174.66 d

802 **when** "0011"**,** -- 1244.51 dis

758 **when** "0100"**,** -- 1318.51 e

715 **when** "0101"**,** -- 1396.91 f

675 **when** "0110"**,** -- 1479.98 fis

637 **when** "0111"**,** -- 1567.98 g

601 **when** "1000"**,** -- 1661.22 gis

567 **when** "1001"**,** -- 1760.00 a

535 **when** "1010"**,** -- 1864.66 b

505 **when** "1011"**,** -- 1975.53 h

0 **when** **others;**

Command **<=** "0011"**;**

Address **<=** "0000"**;**

**process(**Clk**)**

**begin**

**if** **rising\_edge(**Clk**)** **and** Rdy **=** '1' **then**

**if** clks\_delay **/=** 0 **then** --

**if** clk\_counter **<** clks\_delay **then**

clk\_counter **<=** clk\_counter **+** 1**;**

Start **<=** '0'**;**

**else**

Start **<=** '1'**;**

**if** var **<** X"FAD" **then**

var **<=** var **+** 82**;**

**else**

var **<=** X"000"**;**

**end** **if;**

clk\_counter **<=** 0**;**

**end** **if;**

**else**

var **<=** X"000"**;**

clk\_counter **<=** 0**;**

**end** **if;**

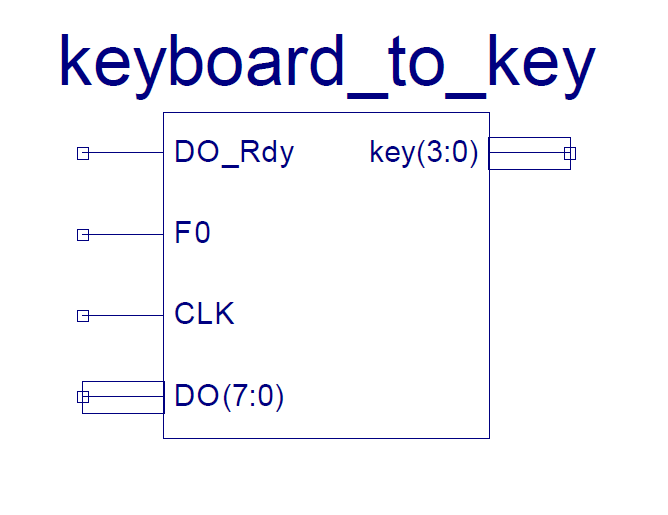
**end** **if;**

**end** **process;**

Data **<=** STD\_LOGIC\_VECTOR**(**var**);**

**end** Behavioral**;**

* + 1. **keyboard\_to\_key**

****

Rysunek Symbol modułu keyboard\_to\_key

* 1. **TODO**

1. **Literatura**

Sugier, J. *Moduły pomocnicze do Spartan-3E Starter Kit - DACWrite*. http://www.zsk.ict.pwr.wroc.pl/zsk\_ftp/fpga/#\_Toc59028436

XILINX. *Spartan-3E FPGA Starter Kit Board User Guide*  https://www.xilinx.com/support/documentation/boards\_and\_kits/ug230.pdf

1. Spis komponentów i funkcjonalności płyty Spartan-3E

   XILINX. *Spartan-3E FPGA Starter Kit Board User Guide - str. 12*. [↑](#footnote-ref-1)
2. Opis protokołu interfejsu PS/2 z klawiaturą

   XILINX. *Spartan-3E FPGA Starter Kit Board User Guide - str. 64*. [↑](#footnote-ref-2)
3. Opis modułu DAC

   XILINX. *Spartan-3E FPGA Starter Kit Board User Guide - str. 69*. [↑](#footnote-ref-3)