POLITECHNIKA WROCŁAWSKA



WYDZIAŁ ELEKTRONIKI

Układy Cyfrowe i Systemy Wbudowane 2

Obsługa interfejsu USB do komunikacji FPGA z PC

*Autorzy: Prowadzący projekt:*

Michał Bańka, 235051 dr inż. Jarosław Sugier

Piotr Kołeczek, 234940

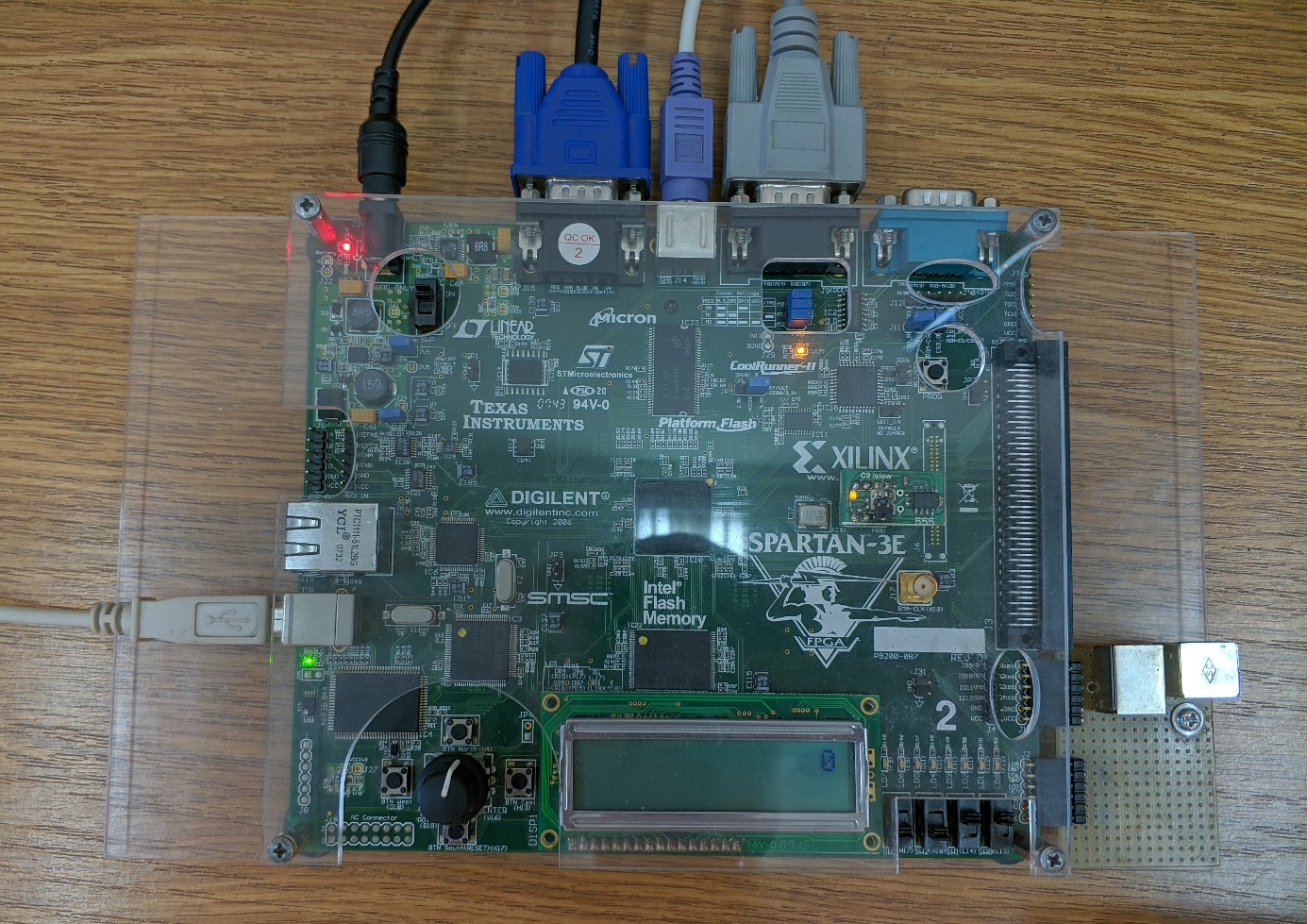
Spis treści

(utwórz spis treści – word umie wygenerować go)

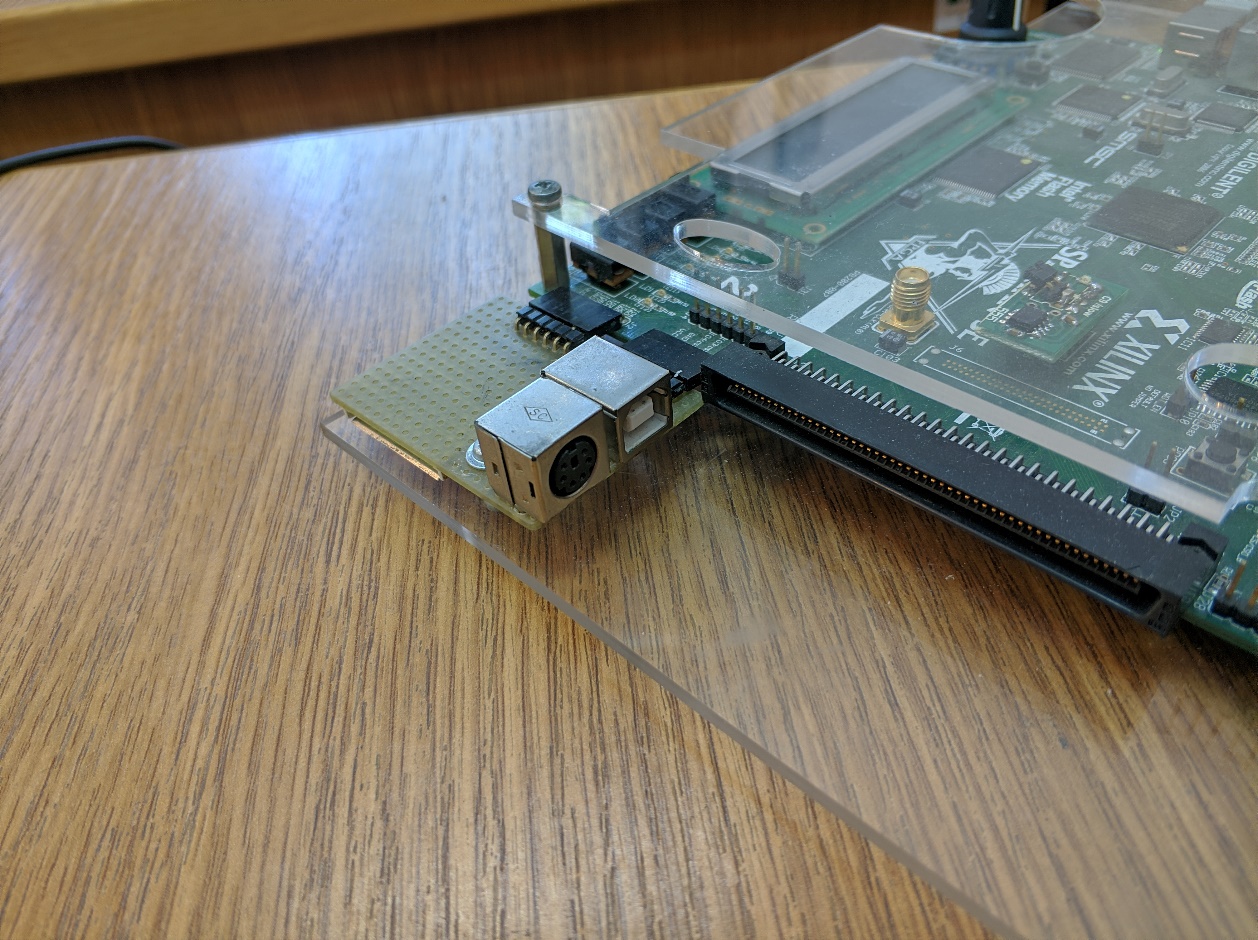
1. **Wprowadzenie**
   1. Cel i zakres projektu

Celem zadania projektowego było napisanie oprogramowania obsługującego interfejs USB 2.0, które umożliwi nawiązanie połączenia i zapewni komunikację urządzenia cyfrowo-logicznego z układem FPGA z komputerem osobistym. Zasięg naszych prac obejmował napisanie kodu obsługującego FPGA w języku ***VHDL*** (maszyny stanów, układy sekwencyjne), przeprowadzenie symulacji behawioralnych w celu zbadania zachowania naszego kodu i ostatecznie przesłania go w wersji wykonywalnej do układu.

* 1. Sprzęt

Do realizacji tematu wraz z wyżej wymienionymi celami projektowymi wykorzystaliśmy układ ***Digilent Spartan-3E Starter Board*** z matrycą logiczną FPGA marki ***Xilinx,***a bezpośrednio do komunikacji tego układu z komputerem wykorzystany został kabel USB.

Rysunek 1: Wykorzystywana płyta rozwojowa Spartan-3E firmy Digilent.

Dodatkowo, w celu realizacji projektu, wykorzystywany był także zewnętrzny moduł z w wlutowanym interfejsem PS/2, USB oraz czytnikiem kart pamięci SD.

Rysunek 2: Dołączona płytka z dodatkowymi interfejsami do komunikacji z FPGA.



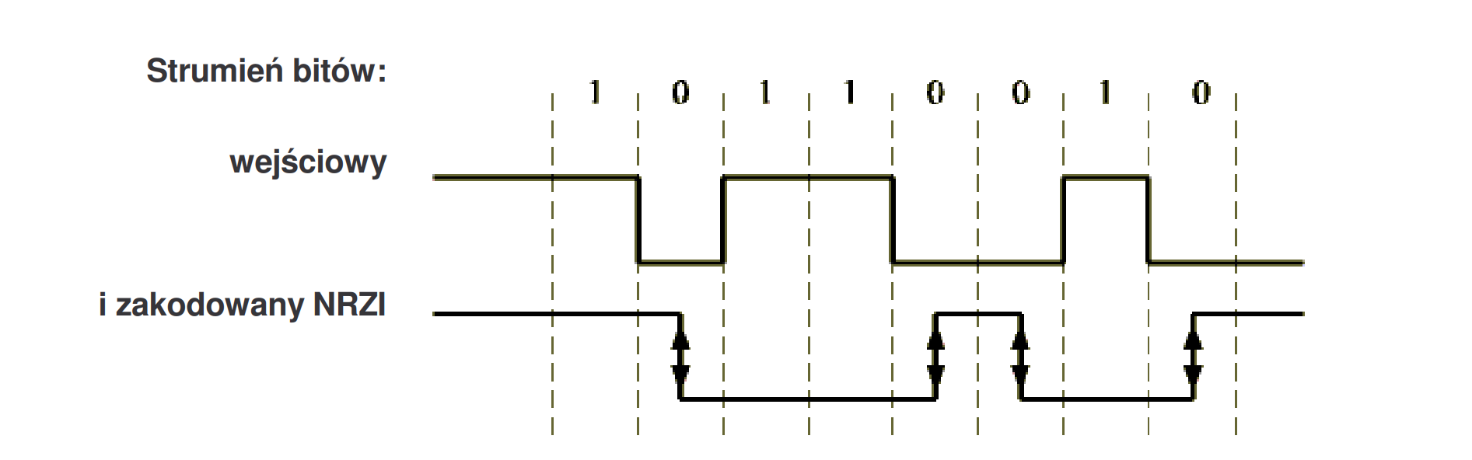
Rysunek 3: Kabel USB do komunikacji Komputera z płytą laboratoryjną.

* 1. Zagadnienia teoretyczne
     1. Wstęp teoretyczny

Interfejs USB (*Universal Serial Bus)*, nazywany **uniwersalną magistralą szeregową** jest komputerowym złączem do wymiany danych, który na dobre wyparł przestarzałe interfejsy szeregowe i równoległe. Sposób przesyłania danych przez USB dzieje się w sposób szeregowy, a zatem dane są przesyłane jednym strumieniem bajtowym. Technologia została opracowana przez światowe korporacje, tj. **IBM**, **Intel**, **Microsoft, DEC** oraz **Compaq**.

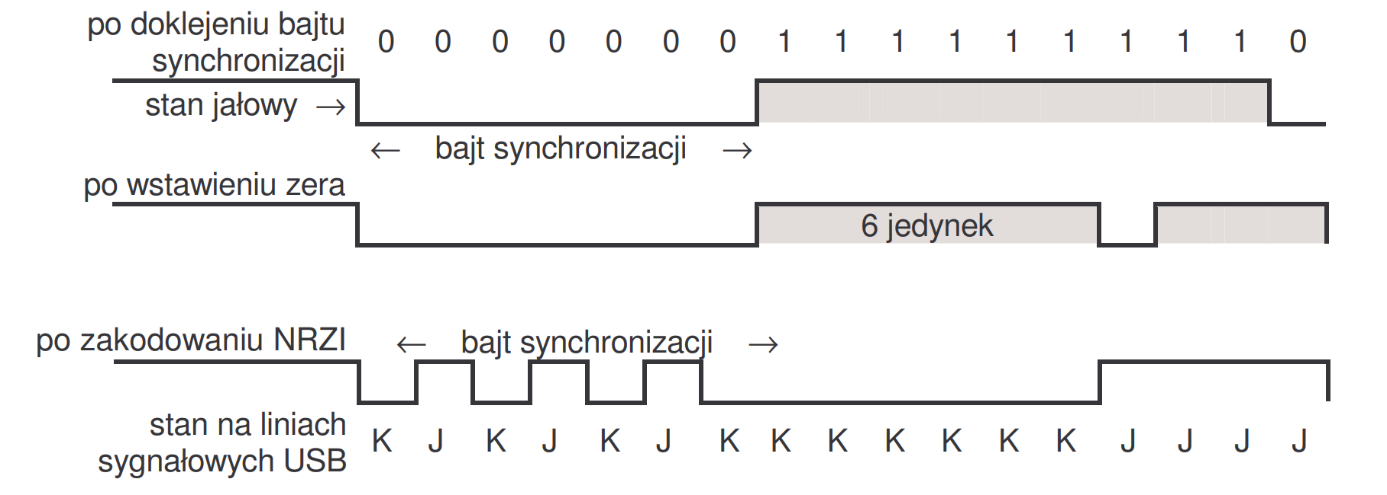
Ze względu na sukces, jaki odniosła ta technologia, interfejs USB doczekał się wielu ulepszeń oraz standaryzacji na przestrzeni ostatnich lat. Aktualnie możemy wyróżnić sześć standardów, które są wykorzystywane:

1. ***USB 1.1*** Low Speed, 1,5 Mbit/s,
2. ***USB 1.1*** Full Speed, 12Mbit/s,
3. ***USB 2.0*** Hi-Speed, 480 Mbit/s,
4. ***USB 3.0*** SuperSpeed, 5Gbit/s,
5. ***USB 3.1*** SuperSpeed +, 10Gbit/s,
6. ***USB 3.2*** SuperSpeed+, 20Gbit/s.
   * 1. Kodowanie danych

Dane wysyłane na magistralę dzielone są na pakiety i kodowane metodą **NRZI** (odwrotny kod bez powrotu do zera - ang. *non-return-to-zero inverted*). Ponieważ kod NRZI jest kodem samo-synchronizującym, nie trzeba oddzielnie przesyłać impulsów zegarowych. W przyjętej metodzie kodowania (NRZI) jedynka jest reprezentowana przez brak zmiany poziomu a zero przez zmianę poziomu, tak jak to podano na poniższym rysunku:

Rysunek 4: Kodowanie metodą NRZI.

Bity danych kierowane są na magistralę USB w kolejności od mniej znaczących do bardziej znaczących. Ciąg bitów, jeszcze przed zakodowaniem NRZI, jest modyfikowany przez wstawianie zera po każdej wykrytej sekwencji 6 jedynek. Operacja wstawiania zer, ma chronić układy pętli PLL (sprzężenia fazo-czułego) przed zablokowaniem. W odbiorniku, po rozkodowaniu danych, wstawione bity są usuwane.



Rysunek 5: Strumień bitów po wstawieniu zera i zakodowaniu NRZI.

Jeszcze przed wstawieniem bitów i kodowaniem NRZI, do każdego wysyłanego na magistralę pakietu danych, zostaje dołączony bajt synchronizacji SYNC (sekwencja 7 zer zakończona jedynką).

Na **rysunku 5**. przedstawiono metodę wstawiania bitów. W tym przykładzie sekwencja 6 jedynek wystąpiła na samym początku pakietu danych. Ostatni bit w polu synchronizacji jest pierwszą jedynką w tej sekwencji.

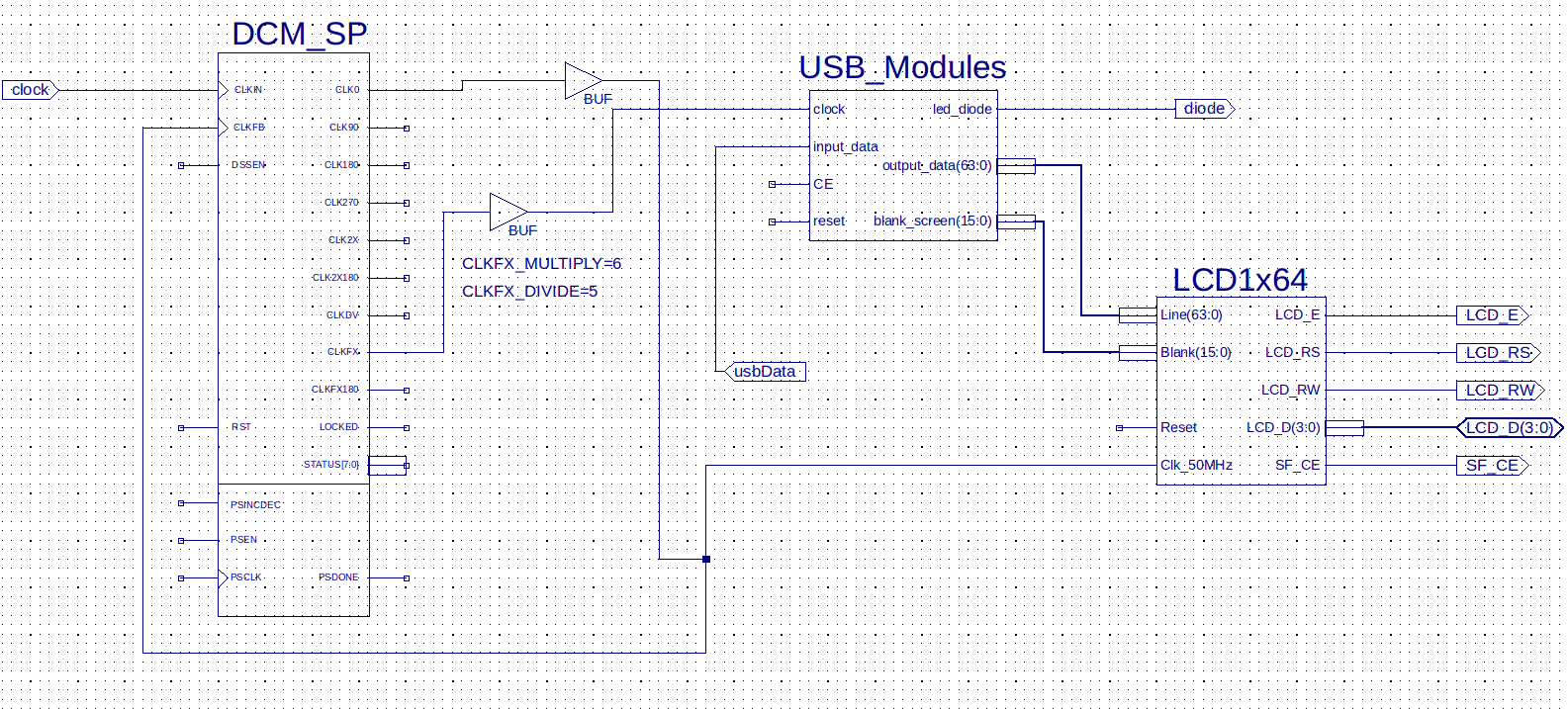
* + 1. Elektryczna reprezentacja bitów

Bity w obrębie pakietu danych na liniach sygnałowych magistrali USB reprezentowane są przez jeden z dwóch stanów:

* **J** - w zakodowanym NRZI strumieniu bitów odpowiada niskiemu poziomowi,
* **K** - jest stanem przeciwnym.

W zależności od częstotliwości z jaką przesyłane są magistralą bity strumienia danych, stany **J** i **K** na liniach sygnałowych magistrali USB reprezentowane są przez różne sygnały różnicowe.

1. **Projekt**
   1. Schemat logiczny



Rysunek 6: Schemat układu naszego projektu.

Nasz projekt składa się z trzech modułów:

* **DCM\_SP**
* **USB\_Modules**
* **LCD1x64**
  1. Moduły
     1. DCM\_SP

**DCM\_SP** (*Digital Clock Manager*)jest modułem odpowiedzialnym za modyfikację standardowej częstotliwości sygnału zegarowego **50MHz**, która jest najczęściej spotykaną częstotliwością taktowania zegara w urządzeniach cyfrowych w rozumieniu systemów wbudowanych. Ze względu na to, że częstotliwość sygnału zegarowego portu USB wynosi **60MHz**, koniecznym było wykorzystanie tego gotowego modułu.

Najważniejsze wejścia tego układu to:

* *CLKIN (clock input)* – wejście, do którego podpina się częstotliwość, którą chcemy zmodyfikować (powinna się zawierać w przedziale od **1** do **280** MHz),
* *CLKFB (clock feedback)* – sprzężenie zwrotne do układu DCM, które jest niezbędne w momencie wykorzystywania wyjść tego układu.

Najważniejsze wyjścia układu:

* *CLK0, 90, 180, 270 (quadrant phase shift outputs)* – wyjścia pozwalające na przesunięcie fazowe sygnału zegarowego o ćwiartkę fazy,
* *CLKFX (synthesied clock output)* – wyjście zegarowe, pozwalające na wykorzystane zmodyfikowanego pierwotnego sygnału zegarowego *CLKIN*, które jest kontrolowane przez parametry *CLKFX\_MULTIPLY*, pozwalające przemnożyć o zadaną wartość częstotliwość oraz *CLKFX\_DIVIDE*, które z kolei umożliwia podzielić częstotliwość wejściową o zadaną wielkość.
  + 1. USB\_Modules

**USB\_Modules** jest modułem przygotowanym na potrzeby realizacji tematu projektowego, zawierającym całą logikę programu, która pozwala na komunikację układu cyfrowego Spartan-3E z innym, zewnętrznym urządzeniem (w przypadku realizacji projektu – komputera stacjonarnego) za pomocą interfejsu USB.

Najważniejsze wejścia:

* *clock* – wejściowy sygnał, do którego doprowadzany jest zmodyfikowany sygnał zegarowy z układu Digital Clock Manager,
* *input\_data –* do wejścia podpinany jest sygnał z interfejsu USB,

Najważniejsze wyjścia:

* *output\_data(63:0)* – wektor 64 bitów (16 bajtów), które są wyprowadzane na wyświetlacz Spartana,
* *blank\_screen(15*:0) – wektor 16 bitów, który pozwala na wygaszenie nieużywanych pozycji wyświetlacza LCD.
  + 1. LCD1x64

**LCD1x64** to moduł pozwalający wyświetlić wartości heksadecymalne na LCD Spartana, który wykorzystuje jedną linię wyświetlacza.

Ważne wejścia:

* *Line(63:0)* – do tego wejścia doprowadzany jest wektor danych, który później jest przed moduł konwertowany do wartości heksadecymalnych,
* *Blank(15:0)* – wektor wygaszający nieużywane pozycje wyświetlacza,
* *CLK\_50MHz* – doprowadzenie sygnału zegarowego 50MHz.

Ważne wyjścia:

* *LCD\_E,*
* *LCD\_RS,*
* *LCD\_RW,*
* *LCD\_D(3:0),*
* *SF\_CE.*
  + 1. Podmoduły
       1. detection

Pierwszym podmodułem, od którego rozpoczęliśmy pracę nad projektem jest proces **detection**, który jest niczym innym jak ośmiobitowym rejestrem przesuwnym **w lewo**. Działanie rejestru przesuwnego w lewo polega na przepisaniu siedmiu młodszych bitów do wektora i „doklejenie” na najmłodszej (ósmej) pozycji wartości *input\_data*, która jest odczytywana z układu Spartan.



Listing 1: kod rejestru przesuwnego.

* + - 1. mod5\_counter

Ten proces pozwala nam na precyzyjne pobranie wartości w kanale. Dzięki licznikowi modulo 5 próbkowany jest sygnał w obrębie jednego bitu.



Listing 2: kod licznika modulo 5.

* + - 1. mod8\_counter

Licznik modulo 8 jest procesem niezbędnym do inicjalizacji połączenia USB – do wykrycia tak zwanego wzoru synchronizującego (sync-pattern). Ten wzór ma długość 8 bitów (jednego bajta) i jest wysyłaną preambułą przez interfejs USB w momencie połączenia urządzeń przez kabel wykorzystujący ten interfejs.



Listing 3: kod licznika modulo 8.

* + - 1. machine\_state

Proces, będący maszyną stanów, która pozwala nam określać w którym momencie przechwytywania preambuły USB znajduje się płytka Spartana – dzięki temu w prosty sposób, wykorzystując powyższe liczniki nawiązujemy połączenie USB.



Listing 4: kod maszyny stanów.

Poniżej przedstawiono graf