

# Języki Opisu Sprzętu

## Dokumentacja projektowa

### Generator przebiegu z wykorzystaniem pmod DA1 i interfejsu UART

Łukasz Bartoszek, Piotr Deda

Czerwiec 2022

## 1 Założenia projektu

Celem projektu jest zbudowanie arbitralnego generatora przebiegów na płycie testowej ZedBoard. Projekt został napisany w języku SystemVerilog.

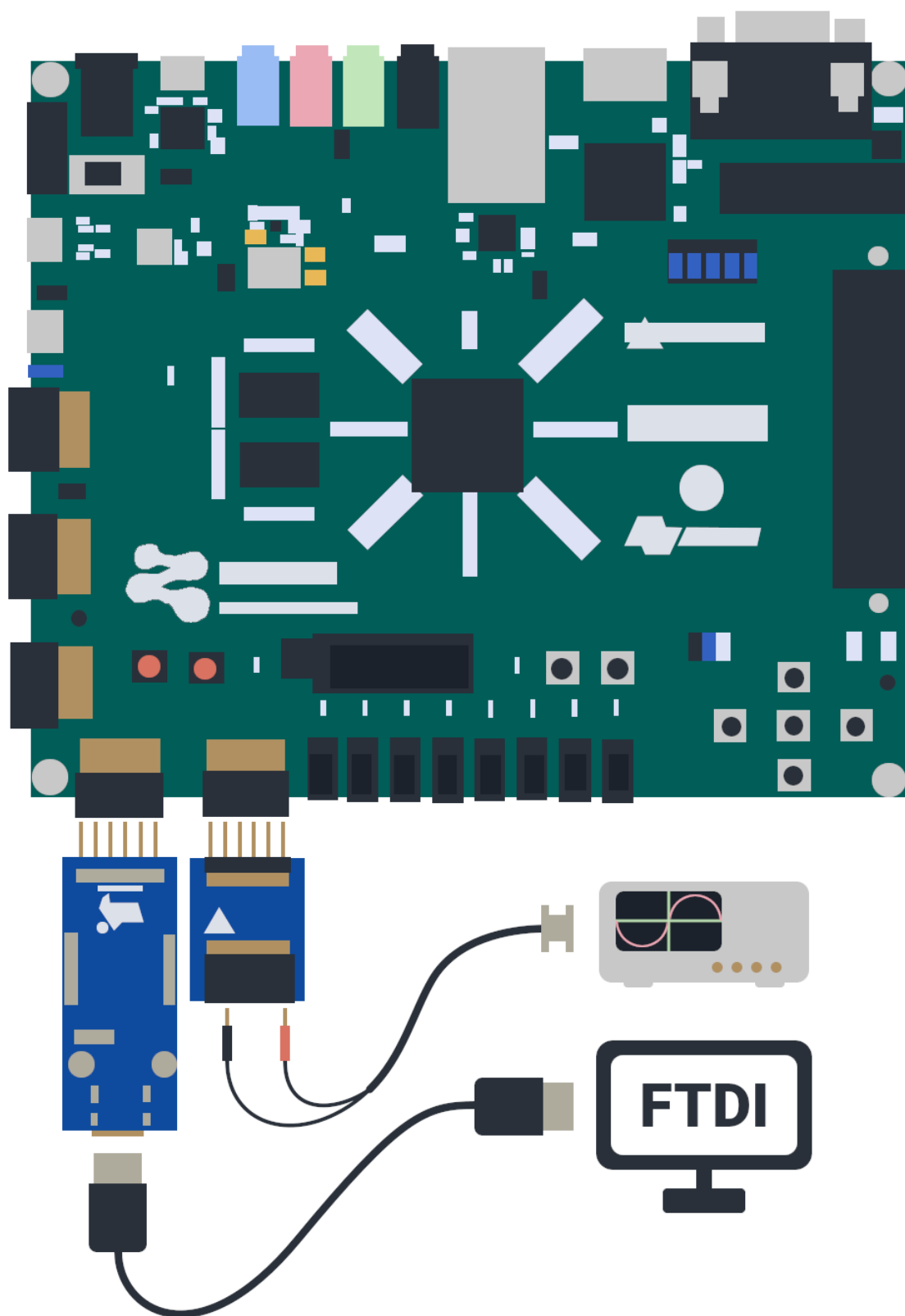
Dane o przebiegach pobierane są z pamięci trwałej urządzenia i przy pomocy SPI przekazywane do modułu pmod DA1. Sterowanie generatorem (zmiana przebiegu, natężenia, amplitudy) odbywa się przy pomocy skryptu napisanego w języku Python, który przesyła komendy interfejsem UART.

## 2 Uruchomienie Projektu

### 2.1 Podłączanie Płytki

W pierwszym kroku należy skompletować płytę Zedboard wraz z peryferiami.

1. Do gniazd JA podłączamy DAC pmod DA1 zgodnie z podanym schematem:
  - SYNC – JA1
  - D0 – JA2
  - D1 – JA3
  - SCLK – JA4
2. Do gniazd JB podłączamy moduł FT232 USB-UART zgodnie z podanym schematem:
  - tx – JB4
  - rx – JB3
3. Płytkę Zedboard i moduł UART podłączamy kablami USB do komputera.
4. Podłączamy zasilanie i włączamy płytkę.
5. Sygnał generowany jest na wszystkich czterech wyjściach pmod DA1. Do testowania można podpiąć do oscyloskopu dowolne z gniazd A1, A2, B1, B2 razem z GND.



Rysunek 1: Schemat połączenia układu

## 2.2 Programowanie Płytki

W programie Vivado otwieramy projekt, generujemy strumień bitowy i programujemy płytke.

## 2.3 Znajdowanie adresu URL FTDI

Z pomocą Pythona uruchamiamy skrypt `ftdi_urls.py` i kopiujemy adres URL oznaczony jako UART. Wklejamy go do skryptu `cmd_uart.py` w pierwszej linijce.

## 2.4 Uruchamianie skryptu i sterowanie przebiegiem

Po uruchomieniu skryptu z pomocą Pythona zobaczymy następujące menu:

```
===== MAIN MENU =====
|| Choose generator properties: ||
|| (1) - waveform           ||
|| (2) - amplitude          ||
|| (3) - frequency          ||
|| (4) - turn current off   ||
|| (0) Exit                 ||
|
```

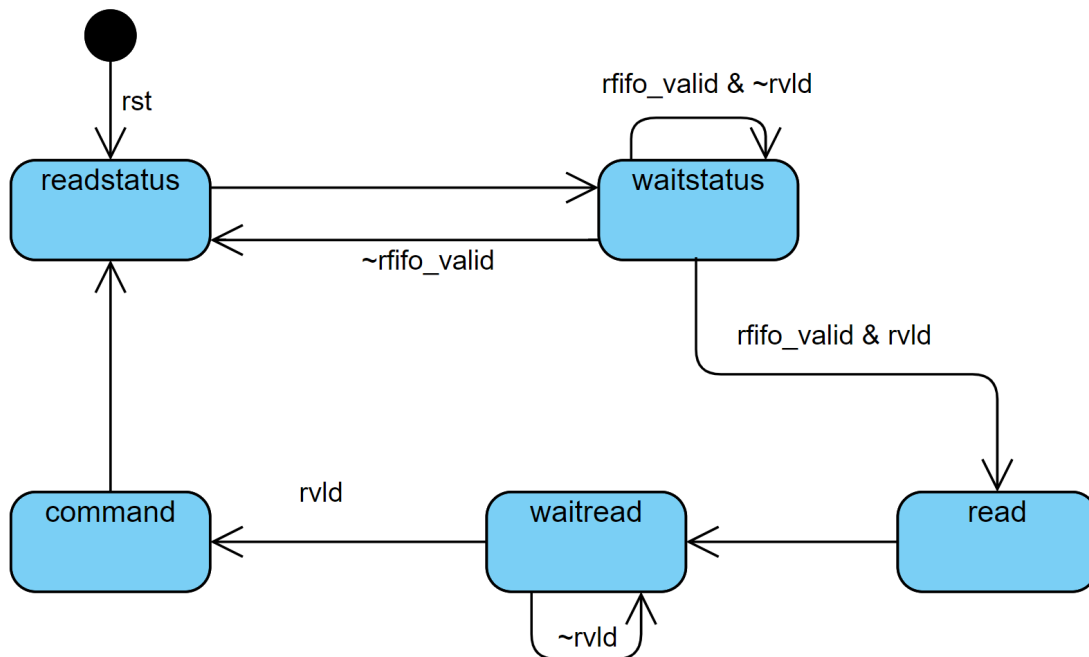
Rysunek 2: Menu główne

Aby wybrać daną opcję należy wpisać w terminalu odpowiadającą jej liczbę i zatwierdzić przyciskiem **Enter**. W przypadku sterowania amplitudą i częstotliwością należy wpisać liczbę z podanych zakresów.

## 3 Implementacja

### 3.1 UART

UART wykorzystywany jest do przesyłania komend z komputera za pomocą FTDI. Wszystkie potrzebne dane zawarte są w komendach, dlatego zaimplementowany układ nie uwzględnia transmisji danych. Parametry układu to baud rate 230400, częstotliwość 100 MHz, transmisja 8 bitów danych, bez sprawdzania parzystości.



Rysunek 3: Maszyna stanów układu master UART

## 3.2 Komendy

### 3.2.1 StartGen

Rozpoczyna generację przebiegu o kształcie zadanym przez dwa ostatnie bity komendy.

1	0	0	0	0	0	1	0	sinus
						0	1	trójkąt
						0	0	piła
						1	1	wykładniczy

### 3.2.2 StopGen

Zatrzymuje generację sygnału do momenty wydania następnej komendy StartGen. W czasie zatrzymania można zmieniać częstotliwość i amplitudę, które generator "zapamiętuje".

0	0	1	1	1	1	0	0
---	---	---	---	---	---	---	---

### 3.2.3 SetFreq

Ustawia częstotliwość generowanego sygnału. 6 bitów (f) jest traktowane jako dzielnik częstotliwości – wartość bitowa 1 oznacza maksymalną częstotliwość 2 KHz, natomiast wartość 2 oznacza  $2\text{KHz}/2 = 1\text{KHz}$ . Dla wartości 0, układ nie będzie generował sygnału.

0	1	f	f	f	f	f	f
---	---	---	---	---	---	---	---

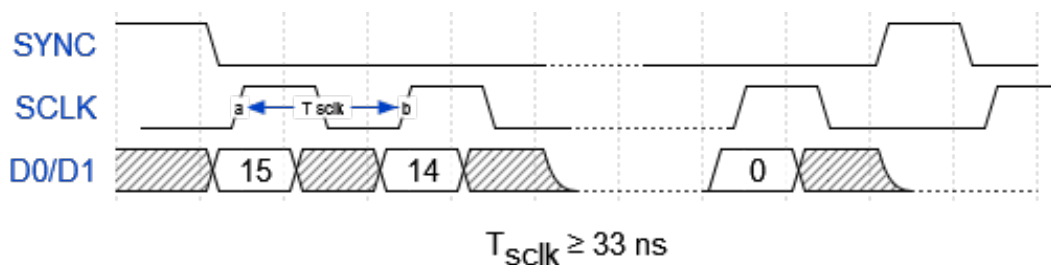
### 3.2.4 SetAmpl

Ustawia amplitudę generowanego sygnału. 6 bitów (a) traktowane jest jako zakres – wartość bitowa 63 oznacza maksymalną amplitudę 3.3 V, natomiast wartość 31 oznacza  $3.3\text{V} * 32/64 = 1.65\text{V}$ .

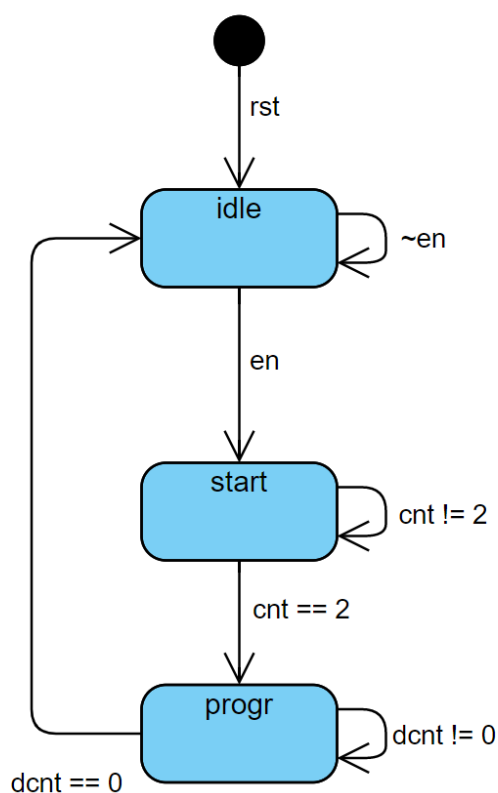
1	1	a	a	a	a	a	a
---	---	---	---	---	---	---	---

### 3.3 SPI

SPI służy w projekcie do przesyłania danych do układu pmod DA1. Ze względu na charakterystykę układu, zaimplementowany jest prostszy model zawierający jedynie transmisję MOSI (z płytki do pmod DA1).



Rysunek 4: Sygnały używane przez SPI do komunikacji z pmod DA1



Rysunek 5: Maszyna stanów układu SPI

## 4 Charakterystyka generowanych przebiegów

Program umożliwia generowanie przebiegu piło-kształtnego, trójkątnego, sinusoidalnego i wykładniczego. Operuje w zakresie częstotliwości  $\langle 32, 2000 \rangle \text{ Hz}$  i zakresie amplitudy  $\langle 0.2, 3.3 \rangle \text{ V}$ .

Dane przekazywane przy pomocy SPI do DAC'a zostały wygenerowane przy pomocy skryptu napisanego w Pythonie. Każdy przebieg generowany jest na podstawie 200 próbek w zakresie  $\langle 0, 255 \rangle$ .

## 4.1 Częstotliwość

Uzyskane wartości częstotliwości są wynikiem wykorzystania 200 próbek i charakterystyką DAC'a, który może przyjmować próbkę co minimalnie 33ns. Zegar sterujący przekazywaniem kolejnych wartości z pamięci ma konfigurowalne taktowanie, a sterowanie nim pozwala na zmianę częstotliwości generowanego sygnału. Jednym z argumentów zegara jest mnożnik ilości jego okresów, który dla wartości 1 ustawia taktowanie zegara na:

$$\frac{100\text{MHz}}{250} = 400\text{KHz}$$

Co przy naszej liczbie próbek daje częstotliwość sygnału na poziomie:

$$\frac{400\text{KHz}}{200} = 2\text{KHz}$$

Tego typu implementacja powoduje, że możliwe do uzyskania wartości częstotliwości sygnału są ilorazami 2000 Hz i liczb z przedziału  $\langle 1, 63 \rangle$ .

## 4.2 Amplituda

Podobne ograniczenie występuje przy sterowaniu amplitudą. Wartości przekazywane do DAC'a muszą mieścić się w przedziale  $\langle 0, 255 \rangle$ . Takie też wartości zostały przez nas wygenerowane i zapisane w pamięci płytki, skąd są pobierane.

Modyfikowanie amplitudy umożliwia istnienie parametru, który używamy do przemnażania wartości przed wysłaniem do DAC'a. Wzór wygląda następująco:

$$output = input * \frac{param + 1}{64},$$

gdzie *param* należy do zbioru  $\langle 0, 63 \rangle$ .

Tego typu rozwiązanie oferuje nam liniowe sterowanie napięciem w przedziale  $\langle 0.2, 3.3 \rangle$  V.

# 5 Raport syntezy

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	224	0	53200	0.42
LUT as Logic	224	0	53200	0.42
LUT as Memory	0	0	17400	0.00
Slice Registers	138	0	106400	0.13
Register as Flip Flop	138	0	106400	0.13
Register as Latch	0	0	106400	0.00
F7 Muxes	34	0	26600	0.13
F8 Muxes	16	0	13300	0.12

Tabela 1: Elementy wynioskowane w trakcie syntezy

Tabela 1 przedstawia ilość wykorzystanych LUT, rejestrów i multiplekserów po syntezie. Uzyskane wartości są zgodne z oczekiwaniami. Żadne rejestry nie zostały wynioskowane jako zatraski.

## 6 Bibliografia

- Instrukcja do zrealizowania zadania,
- ZedBoard Zynq™ Evaluation and Development Hardware User's Guide,
- PmodDA1 Reference Manual, [https://reference.digilentinc.com/pmod/pmod/da1/ref\\_manual](https://reference.digilentinc.com/pmod/pmod/da1/ref_manual)
- AD7303; +2.7 V to +5.5 V, Serial Input, Dual Voltage Output 8-Bit DAC, <http://www.analog.com/media/en/technical-documentation/data-sheets/AD7303.pdf>
- FT232R USB UART IC Datasheet Version 2.07 Clearance No.: FTDI 38, Document No.: FT\_000053, FTDI Ltd., 2010
- PyFtdi Documentation, <https://eblot.github.io/pyftdi/>