

Tytuł: Drukarka

Autorzy: Jakub Róg (JR), Jan Strączek (JS)

Ostatnia modyfikacja: 01.09.2024

Spis treści

| | |
|--|---|
| 1. Repozytorium git..... | 1 |
| 2. Wstęp | 1 |
| 3. Specyfikacja | 1 |
| 3.1. Opis ogólny algorytmu..... | 1 |
| 3.2. Tabela zdarzeń | 2 |
| 4. Architektura..... | 2 |
| 4.1. Moduł: top | 2 |
| 4.1.1. Schemat blokowy | 2 |
| 4.1.2. Porty..... | 3 |
| a) input | 3 |
| b) output..... | 3 |
| 4.1.3. Interfejsy | 3 |
| a) vga_if | 3 |
| 4.2. Rozprowadzenie sygnału zegara | 3 |
| 5. Implementacja..... | 4 |
| 5.1. Lista zignorowanych ostrzeżeń Vivado..... | 4 |
| 5.2. Wykorzystanie zasobów | 4 |
| 5.3. Marginesy czasowe | 4 |
| 6. Film. | 4 |

1. Repozytorium git

https://github.com/Jan-Straczek/UEC_Projekt_1.git

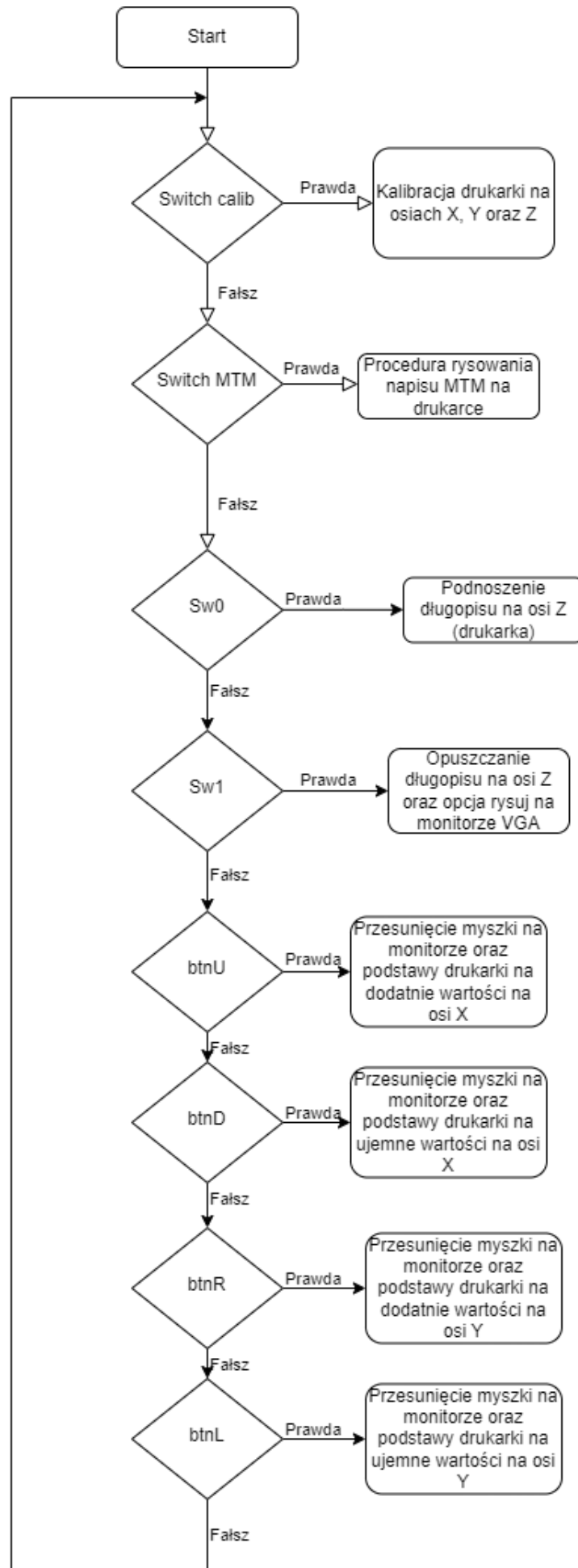
2. Wstęp

W ramach niniejszego projektu podjęliśmy się realizacji urządzenia, które łączy w sobie elementy mechaniki, elektroniki i programowania. Pierwotnie miała to być drukarka 3d, ale niestety zatrzymał nas budżet, dlatego wykonaliśmy urządzenie które zamiast ekstrudera do filamentu wykorzystuje przymocowany długopis umożliwiający pisanie na kartce papieru, działanie i program w obu przypadkach byłoby praktycznie identyczne. Oprócz tworzenia napisu na kartce wyświetlamy go w czasie rzeczywistym na monitorze, inspiracją dla nas był popularny program graficzny 'Paint'. Zbudowane przez nas urządzenie może poruszać się w 3 osiach, oś X oraz Y zbudowana została za pomocą silników krokowych i osi pozyskanych z czytników dysków DVD, oś Z to platforma z przymocowanym narzędziem do pisania która także kontrolowana jest za pomocą silnika krokowego. „Mózgiem” naszej drukarki jest układ fpga BASYS3, który wysyła impulsy na sterowniki silników krokowych DRV8825, oraz odbiera sygnał z czujników krańcowych umożliwiających kalibrację współrzędnych wszystkich osi. Urządzenie może pracować w

trybie manualnym tzn. kontroli za pomocą przycisków, oraz w trybie automatycznym w którym urządzenie samo wypisuje zaprogramowany wcześniej napis.

3. Specyfikacja

3.1. Opis ogólny algorytmu



3.2. Tabela zdarzeń

| Zdarzenie | Kategoria | Reakcja systemu |
|----------------------------------|---------------------|--|
| Przyciski btnU,btnD,btnL,btnR | Interface VGA | Poruszanie myszką po ekranie |
| Przyciski btnU,btnD,btnL,btnR | Drukarka | Poruszanie osiami X oraz Y |
| Switch sw1 | Interface VGA | Rysowanie w białym obszarze |
| Switch sw1 | Drukarka | Podnoszenie długopisu na osi Z |
| Switch sw0 | Drukarka | Opuszczanie długopisu na osi Z |
| Switch sw13 | Drukarka | Rysowanie napisu MTM |
| Switch sw14 | Drukarka | Kalibracja osi X,Y,Z do punktu (0,0,0) |
| Switch sw15 | Drukarka, Interface | Reset |

4. Architektura

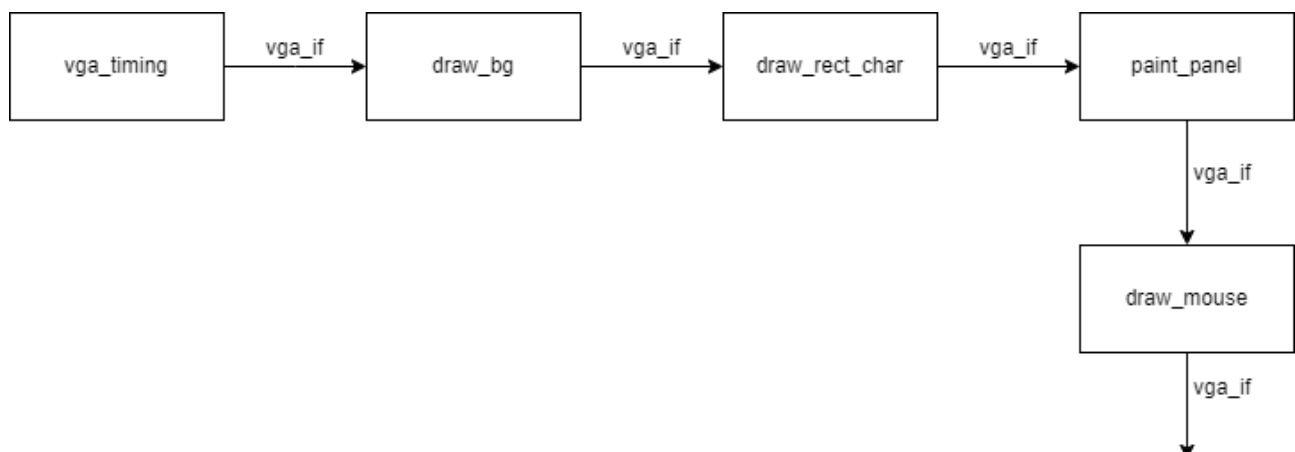
Uwaga: dobrze zrobiony projekt zawiera tylko moduły strukturalne (zbudowane z innych modułów) i funkcjonalne (zawierające bloki proceduralne always @). Staramy się nie generować bloków mieszających te dwa typy, o ile to możliwe.

Uwaga: opisujemy architekturę **tylko głównego modułu oraz rozprowadzenie sygnału zegara**.

4.1. Moduł: top

Osoba odpowiedzialna: JR

4.1.1. Schemat blokowy



4.1.2. Porty

) **Input**

| nazwa portu | opis |
|-------------|--|
| sw0 | Switch do podnoszenia długopisu (oś Z) |
| sw1 | Switch do opuszczania długopisu (oś Z) |
| MTM | Uruchamia procedurę rysowania napisu MTM |

| | |
|---------|---|
| calib | Uruchamia procedurę kalibracji do pozycji (0,0,0) |
| btnU | Sterowanie myszką na ekranie oraz osią Y na drukarce |
| btnD | Sterowanie myszką na ekranie oraz osią Y na drukarce |
| btnL | Sterowanie myszką na ekranie oraz osią X na drukarce |
| btnR | Sterowanie myszką na ekranie oraz osią X na drukarce |
| x_calib | Odczytuje stan kalibracji (jeżeli stan logiczny 1 to pozycja 0) |
| y_calib | Odczytuje stan kalibracji (jeżeli stan logiczny 1 to pozycja 0) |
| z_calib | Odczytuje stan kalibracji (jeżeli stan logiczny 1 to pozycja 0) |

) **Output**

| nazwa portu | opis |
|--------------------|--|
| x_sup | Pin stanu logicznego 1 doprowadzony do przycisku kalibrującego |
| y_sup | Pin stanu logicznego 1 doprowadzony do przycisku kalibrującego |
| z_sup | Pin stanu logicznego 1 doprowadzony do przycisku kalibrującego |
| JB1 | Sygnał direction dla osi X |
| JB2 | Sygnał stepper dla osi X |
| JB3 | Sygnał direction dla osi Y |
| JB4 | Sygnał stepper dla osi Y |
| JB5 | Sygnał direction dla osi Z |
| JB6 | Sygnał stepper dla osi Z |
| vgaBlue | Sygnał dot. koloru niebieskiego w systemie wyświetlania obrazu VGA |
| vgaGreen | Sygnał dot. koloru zielonego w systemie wyświetlania obrazu VGA |
| vgaRed | Sygnał dot. koloru czerwonego w systemie wyświetlania obrazu VGA |
| Vsync | Sygnał synchronizacji pionowej VGA |
| Hsync | Sygnał synchronizacji poziomej VGA |

4.1.3. Interfejsy

) **Vga_if**

| nazwa sygnału | opis |
|----------------------|--|
| vcount[10:0] | Licznik linii – aktualna pozycja pionowa na obrazie |
| vsync | Sygnał synchronizacji pionowej |
| vblnk | Sygnał wyłączający sygnały R,G,B – pozycja pionowa |
| hcount[10:0] | Licznik linii – aktualna pozycja pozioma na obrazie |
| hsync | Sygnał synchronizacji poziomej |
| hblnk | Sygnał wyłączający sygnały R,G,B – pozycja pozioma |
| rgb[11:0] | Informacja o kolorze, który ma zostać wyświetlony na monitorze |

4.2. Rozprowadzenie sygnału zegara

Osoba odpowiedzialna: JR

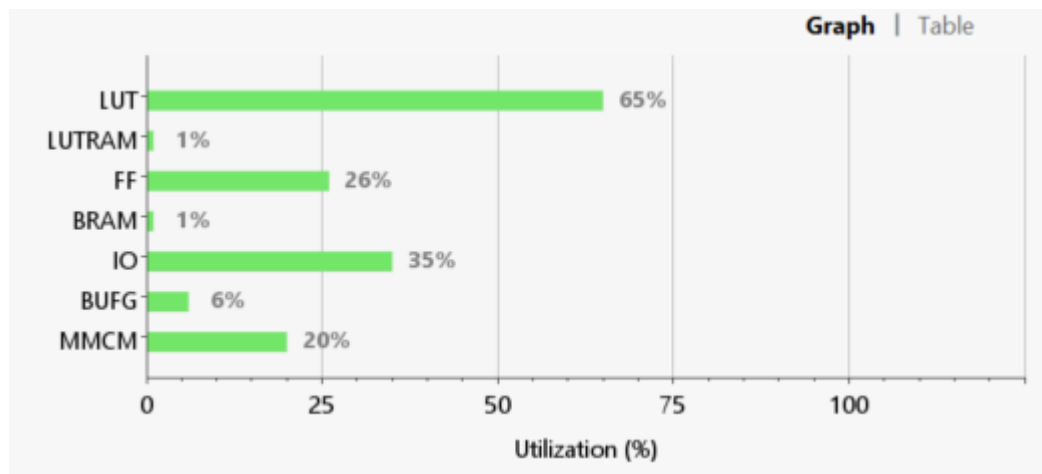
Sygnal zegarowy generowany jest przez moduł Clocking Wizard. Generowany sygnał zegara ma częstotliwość wynoszącą 40MHz i jest używany w modułach związanych z modułem top_vga. Jeżeli chodzi o drukarkę (top_printer) to z powodu konieczności użycia niższych częstotliwości impulsów sterujących pracą silnika (od 1kHz do 100kHz), w module printer_movement został zastosowany dzielnik częstotliwości a impulsy zostały wysłane przez piny wyjściowe Basys3 do wejść drukarki.

5. Implementacja

5.1. Lista zignorowanych ostrzeżeń Vivado.

| Identyfikat or ostrzeżeni a | Liczba wystąpi eń | Uzasadnienie |
|-----------------------------|-------------------|--|
| DRC PDRC-153 | 2 | Głównym problemem mogącym się pojawić po implementacji algorytmu z tym ostrzeżeniem jest pojawienie się opóźnień i niestabilności w generowaniu zegara. Jednak po analizie i testach doszliśmy do wniosku, że ostrzeżenie nie ma wpływu na działanie naszego projektu i nie wprowadza żadnych negatywnych skutków. |
| | | |

5.2. Wykorzystanie zasobów



5.3. Marginesy czasowe

Marginesy czasowe (WNS) dla setup i hold.

| Timing | Setup | Hold | Pulse Width |
|---|----------|------|-------------|
| Worst Negative Slack (WNS): | 7.492 ns | | |
| Total Negative Slack (TNS): | 0 ns | | |
| Number of Failing Endpoints: | 0 | | |
| Total Number of Endpoints: | 10949 | | |
| Implemented Timing Report | | | |

| Timing | | Setup | | Hold | | Pulse Width |
|---|----------|-------|--|------|--|-------------|
| Worst Hold Slack (WHS): | 0.085 ns | | | | | |
| Total Hold Slack (THS): | 0 ns | | | | | |
| Number of Failing Endpoints: | 0 | | | | | |
| Total Number of Endpoints: | 10949 | | | | | |
| Implemented Timing Report | | | | | | |

6. Film.

Link do ściągnięcia filmu:

https://drive.google.com/drive/folders/1fgzBcnyS2KhsAx5ag-2SkpeAqtEANhGZ?usp=drive_link