**Raport z projektu**

**Temat:** Transformata Burrowsa-Wheelera

**Autorzy:** Karolina Bocian – Simulation engineer  
Angelina Kowalik – Hardware engineer  
Weronika Potaczek – Software engineer

Link do repozytorium: <https://github.com/sdupbwt/BWT>

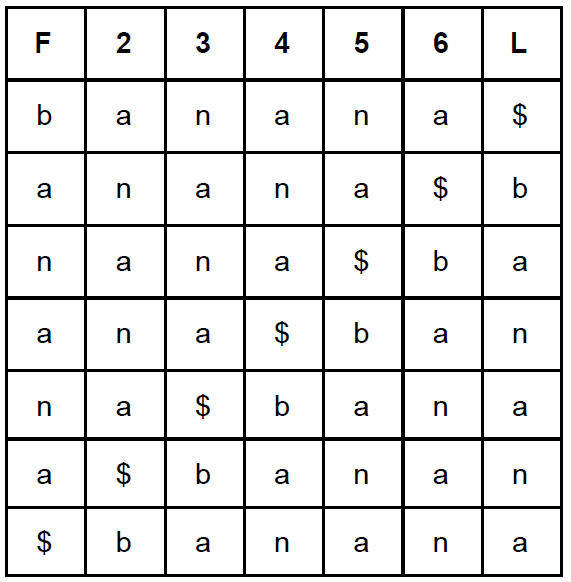
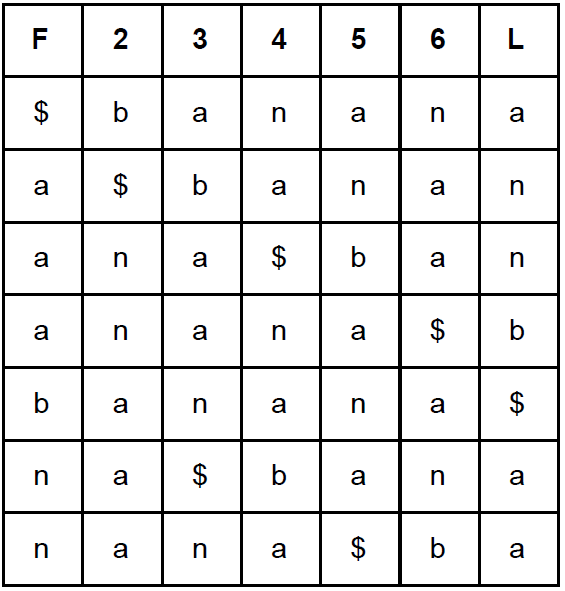
1. **Cel i założenia projektu**

Celem projektu była implementacja transformaty Burrowsa-Wheelera na FPGA.

1. **Opis algorytmu**

Kompresja danych to działanie mające na celu zmianę sposobu zapisu informacji w taki sposób by zmniejszyć objętość danych przy jednoczesnym zachowaniu przenoszonych informacji. Transformata Burrowsa-Wheelera (BWT) jest jedną z najefektywniejszych metod dekompozycji strumienia danych do postaci pośredniej, którą da się znacznie lepiej skompresować za pomocą klasycznych algorytmów kompresji. Sposób działania transformacji BWT polega na pobraniu bloku danych wejściowych, odpowiednim poprzestawianiu jego elementów przy użyciu algorytmu sortowania, co w rezultacie daje na wyjściu blok danych zawierający te same elementy, co blok wejściowy, ale występujące w innej kolejności (występują dłuższe, zgrupowane sekwencje tych samych znaków). Transformacja jest procesem odwracalnym - oryginalna kolejność występowania elementów w bloku może zostać w pełni przywrócona. Działanie algorytmu BWT można podzielić na 3 etapy, pokazane na przykładzie słowa „banana”:

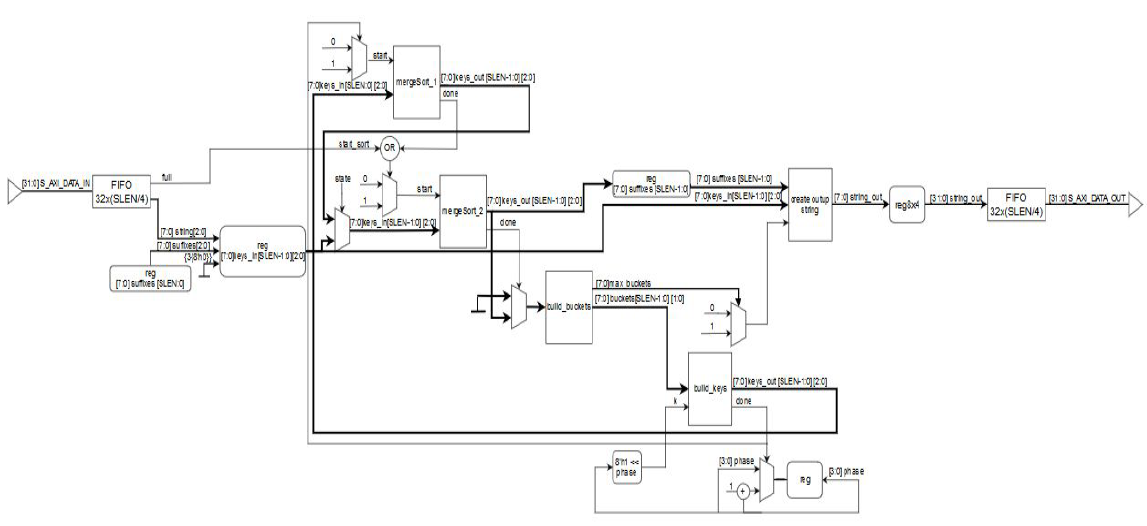
* 1. Dla bloku danych o rozmiarze N generujemy N rotacji ciągu wejściowego. Na koniec ciągu dopisujemy też unikalny symbol terminalny (“$”). (Rys. 1)
  2. Sortujemy łańcuchy leksykograficznie. Symbol terminalny traktujemy jako alfabetycznie mniejszy od każdego innego symbolu. (Rys. 2)
  3. Wyjściowa postać danych po transformacji to sekwencja z ostatniej kolumny (L) posortowanej tabeli, czyli “annb$aa”.



1. **Przebieg pracy nad projektem.**

Projekt rozpoczęto od zebrania podstawowych informacji na temat działania algorytmu i jego wykorzystania. Następnie było przygotowanie modelu behawioralnego algorytmu w języku Python. Po pozytywnych wynikach rozpoczęłyśmy pracę nad projektem schematu blokowego ogólnej struktury sprzętowej. Kolejnym krokiem było przygotowanie syntezowalnego modelu strukturalnego procesora. Następnie zbudowałyśmy połączenie magistralą AXI IPCore pomiędzy procesorem dedykowanym (FPGA) i procesorem ogólnego przeznaczenia (Microblaze). Potem przygotowałyśmy oprogramowanie w języku C na procesor Microblaze. Na koniec uruchomiłyśmy wszystko na płytce rozwojowej Zedboard

1. **Wyniki etapów projektu**

Schemat ogólny procesora dedykowanego.

1. **Podsumowanie**

Podczas wykonywania projektu wykonałyśmy …, który spełniał zadane wymagania. Przeprowadziłyśmy również wszystkie symulacje, żeby udowodnić, że nasz projekt działa. Podczas przeprowadzenia całego projektu nie pojawiły się żadne poważne błędy zaburzające przebieg poszczególnych etapów.