

Design and Fabrication of an Accelerator for Genetic Local Sequence Alignment

Project A (044167) – Requirements Document

Students: Niv Bar-Tov, Dor Lerman

Supervisor: Goel Samuel

Semester: Spring 2023

Project Description:

Numerous fields currently require DNA sequence database searches based on pair-wise alignment, where a query sequence is compared with a database of sequences to find a highest-similarity sequence. This similarity can provide insights on the functionality of the query protein or the role of a gene. Aligning even several hundred DNA or protein sequences consumes several CPU hours on high performance computer. Hence, there is high motivation to implement hardware accelerators to keep up with the increasing amount of data.

The goal of this project is to perform the design and full backend implementation of Smith-Waterman algorithm for local DNA Alignment Accelerator.

Project Requirements:

- Sequences Length: 16 or 32 residues
- Maximum area efficiency
- Minimal on chip memory
- Performance efficiency
- Synthesis
- Physical design (layout)

Project Specifications:

- Clock Frequency: Up to 300 MHz
- Area: 1.0x1.0 sq. mm
- Maximum number of IOs: 32
- Technology: TSMC65n

Development Stages

Development Stages / Design Flow	Tools
Architectural and Logic Design	draw.io (schemes software)
RTL Implementation	SystemVerilog + Synopsys
Functional Simulations	Ncsim/VCS
Controller Design + Implementation	SystemVerilog + Synopsys
Full Functional Simulations	Ncsim/VCS
Synthesis	Design Compiler
Layout Design	First Encounter

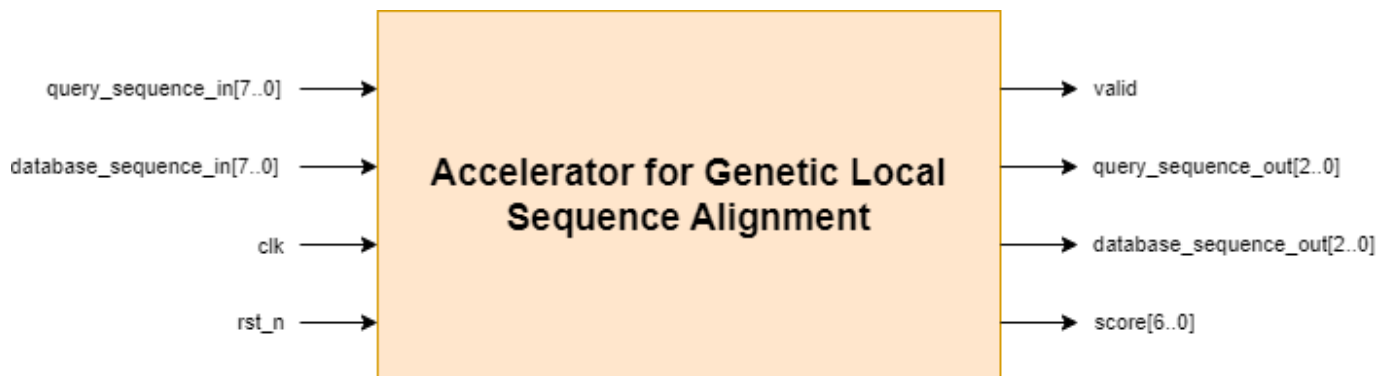
Alternative Solutions

Most of the solutions for the Local Sequence Alignment Problem are using dynamic programming. One of the most famous algorithms is Smith-Waterman algorithm. The main disadvantage of these solutions is that they consume large amount of computing resources as well as computing time (time complexity of $O(n^2)$).

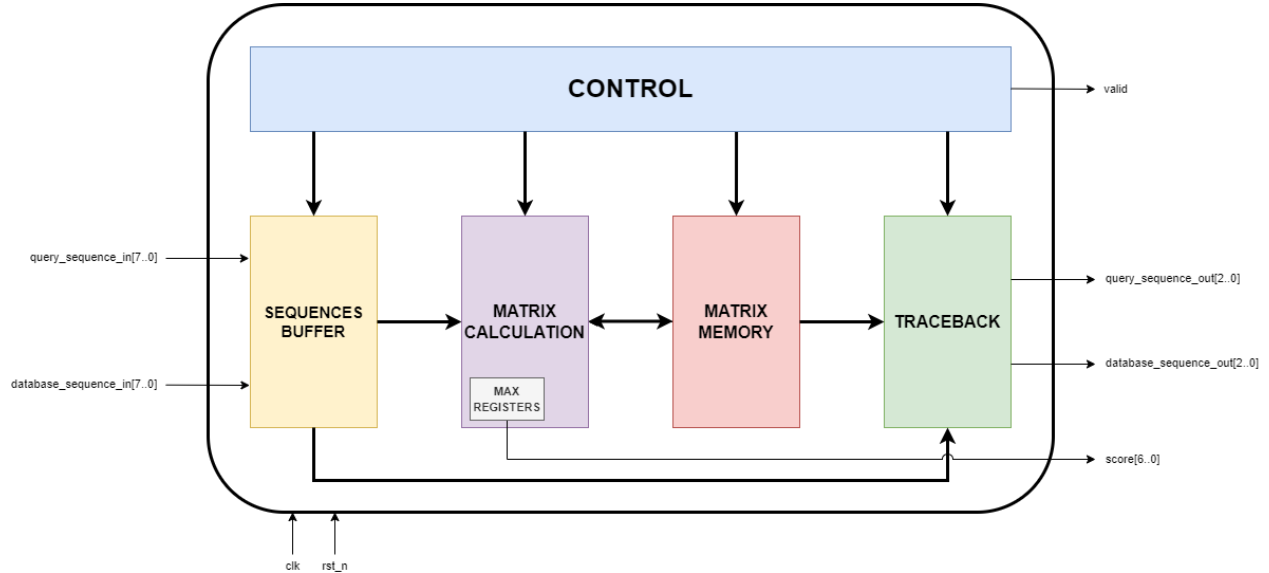
Selected Solution

We decided to implement a hardware accelerator that will solve the Local Sequence Alignment Problem using Smith-Waterman algorithm. The accelerator will be separate from other expensive computing resources and will solve the problem more efficiently because the computation will be done directly on the hardware, instead of complex programming implementations. The architectural design will be pipelined.

Top Level Interface



Top Level Architecture



Project Schedule

Provisional/Preliminary Schedule	Week
Architectural and Logic Design	1-3
RTL Implementation	4-6
Functional Simulations	5-8
Controller Design + Implementation	9-10
Full Functional Simulations	11-13
Exams Period Break	
Report + Final Presentation	14-15
** Backend stages will be done as part of Project B (044169). Detailed plan will be provided in a separate requirements document.	

Gantt Chart

