



Massively Parallel Short-Read Mapping on FPGAs for the Backend Processing of Next-Generation Sequencing

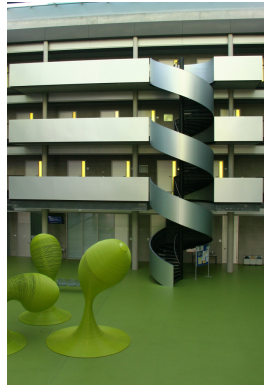
Thomas B. Preußner
Oliver Knodel

Jena, Feb 17, 2011



Itinerary

- *Research Overview*
- Alignment Problem
- State-of-the-Art Software Solutions
Next-Generation Sequencing Conflict
- Field-Programmable Gate Arrays
Hardware Acceleration Approach
Evaluation



Research at VLSI-EDA, TU Dresden

Processor Design

- SHAP Bytecode Processor (Java-programmed embedded platform)
→ <http://shap.inf.tu-dresden.de/>.
- Arrive (RISC + reconfigurable datapath extension).

Architecture Simulation

- Jahris (re-targetable JIT-compiling architectural simulator).

Test and Diagnostics

- embedded kernel & application debugging (ARM, PowerPC)
(cooperation with pls Development Tools → <http://www.pls-mc.com/>).

Efficient FPGA Mapping

- automated carry-chain mapping → <http://dx.doi.org/10.1109/FPL.2010.70>,
- streaming DSP applications (cooperation with Nuclear Physics Department),
- **sequence alignment** (cooperation with Max Planck Institute).

Alignment Problem

Definition

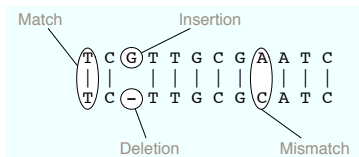
Here: **Local Alignment / Sequence Mapping**:

is a generalized string search that maps an input word $w \in \Sigma^*$ to positions in a text $t \in \Sigma^*$, $|t| \gg |w|$ with tolerating errors.

For $t = rw's$, $|r|$ is considered a matched position iff w is similar to w' as determined by small bounds of:

- differing symbols (mismatches),
- missing symbols (deletions/gaps),
- additional symbols (insertions).

In bioinformatics, the alphabet typically consists of the nucleotide bases $\Sigma = \{C, G, A, T(\equiv U)\}$ (or sometimes of the amino acids):



Alignment Problem

Input Words

- are obtained experimentally from DNA sequencers, and
- are often referred to as *sequences*, *reads* or *queries*.

High-throughput sequencing techniques have been developed since the 1990s. Sequencing machines are commercially available:

454 Genome Sequencer FLX	1 Mio. × 400 bp / 10h
Polonator G.007	150 Mio. × 26 bp / 4d
Illumina HiSeq 2000	750 Mio. × 100 bp / 4d

They produce *many* but rather *short* reads by parallel processing of different sections.

→ Chopped up results!

Alignment Problem

The Text

- is a complete reference genome *assembled*, i.e. linearized, from the reads of a fully sequenced individual, and
- is typically referred to as the *database*.

Genome assembly is another important problem of bioinformatics. It is typically approached as a shortest common superstring (SCS) problem. This is, indeed, NP-complete but (greedy) heuristics work well in practice.

Genome databases make the read mapping complex by sheer size.

NCBI Databases:

Human	9 . 011 . 418 . 319 bp
Mouse	2 . 973 . 905 . 113 bp
Drosophila	124 . 326 . 318 bp

Alignment Problem

Purpose

Genetic Analysis:

- identification of genes, coding vs. non-coding regions, ...
(often based on RNA reads)
- reveal ancestral relationships of species and mutations (→ genetic clock)

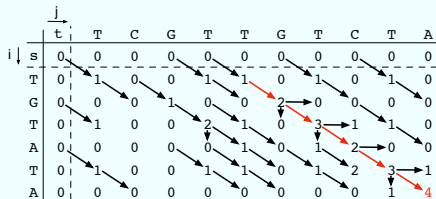
Genetic Diagnostics:

- identification of pathologically relevant genotypes
(risk for allergies, cancer, ...)

State-of-the-Art

SMITH-WATERMAN – the Exact Algorithm [1]

Score Computation within Matrix in $O(|s| \cdot |t|)$:



T C G T T G T C T A
 - - - T G T A T A

$$M(i, j) = \max \begin{cases} M(i-1, j-1) + c(s_i, t_j) & \text{-(Mis-)Match} \\ M(i-1, j) + c_D & \text{- Insertion} \\ M(i, j-1) + c_I & \text{- Deletion} \\ 0 & \text{- Restart} \end{cases}$$

State-of-the-Art

BLAST [2]

Mapping of long reads:

1. Compute seeds as short subsequences from the query including typical mutations and excluding low-complexity regions.
2. Locate seeds in the database by an exact search.
3. Try to extend or join seed matches to form an alignment.



Empirical thresholds are used to filter the partial result sets after each processing step according to their assumed relevance.

BLAST becomes less appealing with decreasing read lengths.

State-of-the-Art

Bowtie [3]

- Exact search optimized by pre-computed BURROWS-WHEELER-Index.
- Expensive mismatch tolerance by backtracking search, unfeasible for more than three (3).

Text	BW-Index
ATTGCGGTA	A ATTGCGG T
TTGCGGTAA	A TTGCGGT A
TGCGGTAAT	C GGTAAT G
GCGGTAATT	G CCGGTAAT T
CGGTAATTG	G GTAATTG C
GGTAATTGC	G TAATTGC G
GTAATTGCG	T AATTGCG G
TAATTGCGG	T GCGGTAA T
AATTGCGGT	T TGCGGTA A

Search for **TGC**:

- starts from the end, and
- avoids scanning the whole database.

Most practical for *short* reads.

State-of-the-Art Maq [4]

- *Huge* (\gg database!) index of paired short seeds for *each* position:

AATC	GCTA	TTGC	ATAG
AATC	GCTA	****	****
AATC	****	TTGC	****
AATC	****	****	ATAG
****	GCTA	TTGC	****
****	GCTA	****	ATAG
****	****	TTGC	ATAG

- Mismatches can only be tolerated outside the seeds.
(Tolerating more than two is unfeasible).

State-of-the-Art

Mappers struggle to keep up with high-throughput sequencers:

- SMITH-WATERMAN is exact but rather slow.
- Parallelizing with general-purpose processors is expensive.
- Heuristics yield feasible run times but deteriorate the result quality.

Many users have adapted to accepting doubtful statistic results from tempting heuristics. A frequent *quality measure* is how many reads, an algorithm was able to map to a database:

- without asking for the actual similarity,
- without checking the correctness of the position, and
- without guarding from false positives.

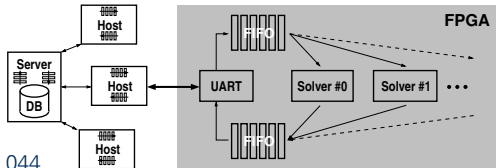
State-of-the-Art Need

... an *exact but fast* sequence mapper → massive, fine-grained parallelism!

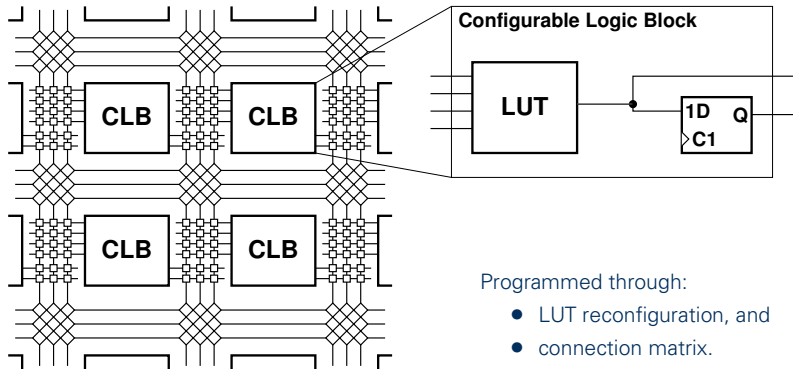
This is how we beat the
26-Queens-Puzzle *before*:

- the cloud computation NQueens@Home and
- two Russian Top-500 supercomputers.

$Q(26) = 22, 317, 699, 616, 364, 044$



FPGA – Field-Programmable Gate Array



FPGA – Field-Programmable Gate Array

FPGA Design vs. Software

Configurable Hardware:

- greater design effort (VHDL, Verilog),
- 10× lower clock frequency than standard CPUs,
- + fine-grained custom concurrency,
- + high computational power per Watt,
- + no instruction overhead.



[5]

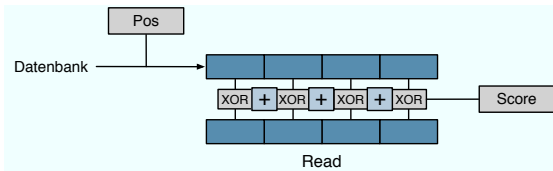
FPGAs offer:

- affordable custom hardware performance (beyond a mass market),
- revisable designs (prototyping, communication protocols), and
- tremendous low-level concurrency.

FPGA – Field-Programmable Gate Array

Trivial Short-Read Mapping

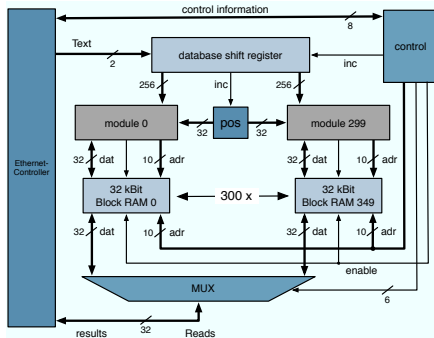
Näive string search with sophisticated hardware mapping:



- trivial tolerance of mismatches through score threshold,
- simple, regular structure, which is easy to parallelize,
- exact results.

FPGA – Field-Programmable Gate Array

Top-Level Module



- 300 parallel search modules on a Xilinx Virtex-6 LX240T [5] running at 200 MHz.
- Each module maps one read of up to 128 bp or two reads of up to 64 bp to the same globally streamed database.
- The system performs up to $300 \times 2 \times 200 \text{ MHz} = 120 \text{ G}$ base pair comparisons per second.

FPGA – Field-Programmable Gate Array

Results

Algorithm	Transformation Time (human genome)	Mapping Time	Mapped Reads (up to 3 errors)
Maq	0:05	16:15	77,7%
Bowtie	8:21	0:54	83,1%
FPGA (V6)	0:04	2:54	100,0%

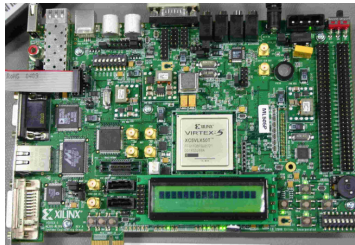
- Reads taken as database probes with injected errors uniformly distributed from 0 to 3.
- SW heuristics fail nearly completely on higher mismatch counts: Reads with 3 errors are almost certainly *not* found!

FPGA – Field-Programmable Gate Array

System Setup

This may be your private HPC cluster for your office:

- PC integration as PCIe extension card, or
- Gigabit Ethernet P2P connection to external box.



Alternatively, we might offer a webservice for mapping against a mirrored set of public databases.

FPGA – Field-Programmable Gate Array

Costs

- High-end devices start at about EUR 2000.
- Power consumption is dominated by communication ($\ll 2$ W for computation on Virtex-6 with 600 search units).
- It takes 20 high-end GPUs to match the FPGA performance:
20 × NVidia GTX for EUR 200 → EUR 4000.
20 × 200 W = 4 kW.
⇒ Each hour of GPU computation costs you an extra Euro in electricity.

FPGA – Field-Programmable Gate Array

User Experience

Yet to collect!






FPGA – Field-Programmable Gate Array

Future Work

- Appealing system integration with smooth frontend interface.
- Moderate extension to allow gaps and deletions (bounded SMITH-WATERMAN).
- Multi-FPGA systems for even greater concurrency.

Thank you!

References

-  T. Smith and M. Waterman, "Identification of common molecular subsequences." *Journal of Molecular Biology*, Vol. 147, No. 1, pp. 195–197, 1981.
-  S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman, "Basic Local Alignment Search Tool." *Journal of Molecular Biology*, Vol. 215, No. 3, p. 403–410, 1990.
-  B. Langmead, C. Trapnell, M. Pop and S. Salzberg, "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome." *Genome Biology*, Vol. 10, No. 3, p. R25, 2009.
-  C. Trapnell and S. Salzberg, "How to map billions of short reads onto genomes." *Nature Biotechnology*, Vol. 27, No. 5, p. 455–457, 2009.
-  Xilinx, → <http://www.xilinx.com/>.