RC-BLAST: Towards a Portable, Cost-Effective Open Source Hardware Implementation

Authors: Krishna Muriki, Keith D. Underwood, Ron Sass

Presented by: Siddhartha Datta
OUTLINE

• Explanation of BLAST
• Implementation Board Description
• RCBLAST implementation
• Current implementation
• Conclusion and future work
WHAT IS BLAST?

• Algorithm used to compare DNA sequences

• Developed at the National Center for Biotechnology Information (NCBI)

• Inputs to BLAST are:
  – Known DNA sequence Databases
  – Unknown DNA sequence queries

• Outputs of BLAST is the similarity score between the query and the database
BLAST TYPES

- 5 flavors – BLASTp, BLASTn, BLASTx, BLASTtx and PSI-BLAST
- BLASTn deals with nucleotide based queries and databases
- Most challenging to speedup due to short lengths of the queries
- BLASTn consists of only 4 letter types – A, C, G, T
BLASTn SOFTWARE PIPELINE
CREATION OF WORDS

- Database and queries are divided into words
- Also known as W-mers
- Length of each word is 8 letters \((16 \text{ bits})\)
STAGE 1 – OUTPUT (HITS)

• Stage 1 (Seed Matching Stage) yields the number of hits
• A hit is a word match between the query and the database
• A hit has a score of 8
• The hits are stored and sent to the next stage
BLASTn SCORING TECHNIQUE

Seed Matching Output

Ungapped Extension Output
THE LOOKUP TABLE

- Size: (65536 x 8) Bytes
- Query is laid out in ascending and descending order
- Length of the query is doubled
- Stores the query offsets for every 8-letter word formed
- In case of any repetition of a word:
  - all offsets are stored in another table
  - Pointer to that offset is stored in the lookup table
IMPLEMENTATION OF THE LOOKUP TABLE

<table>
<thead>
<tr>
<th></th>
<th>1 bit</th>
<th>2 bits</th>
<th>13 bits</th>
<th>16 bits</th>
<th>16 bits</th>
<th>16 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>flag</td>
<td>#hits</td>
<td>PTR</td>
<td>Off1</td>
<td>Off2</td>
<td>Off3</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$2^{16}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
WORD FINDER FUNCTION (STAGE 1)

• This function is identified as the most computationally intensive (critical_code)
• Streams the database and compares each word to the lookup table entry for a hit
• Stores hits when found
• This is the function ported to hardware
IMPLEMENTATION BOARD - THE ACE-2 CARD
THIS LOOKS MORE REAL!!
IMPLEMENTATION OF BLAST ON THE ACE2 CARD

- Host machine is an INTEL architecture
- Red 9.0 Linux is run on this machine
- The Ace2 card is a PCI based card
- Two XC4085XLA FPGAs are on the board
- Two blocks of SRAM blocks 1MB each
- Only the one FPGA and SRAM block is used for the implementation of BLASTn
HARDWARE DESIGN OF STAGE 1

- Design made using VHDL
- `addr_smc` and `hitinfo_smc` are two FSMs
- Subject buffer is a 64 bit register
- `addr_smc` sends 16 bit data to the `hitinfo_smc`
- `hitinfo_smc` makes lookups to the lookup table for hits
- Hit information is stored in the `hit_info_data`
INITIAL RESULTS ON THE ACE2 CARD

- **SW-BLAST**: 0.4 sec
- **RC-BLAST**: 1.93 sec
COMPARISONS OF INCREASE IN DATABASE SIZES, PROCESSOR AND MEMORY SPEEDS

1717

Released by NCBI Genbank in 2004
NEED FOR PORTABILITY

- As the FPGA size increases, the number of cores on it increases
- Cost of each FPGA is reducing
- Very important to design a generic core which can be replicated on the FPGA
- Memory speeds are not increasing creating bottlenecks
LATEST IMPLEMENTATION BOARD – ML410
CURRENT IMPLEMENTATION – SYSTEM LEVEL
CURRENT IMPLEMENTATION – HARDWARE DESIGN
WHY IS THIS IMPLEMENTATION FASTER?

• 64 bit database data is streamed in instead of 32 bits
• DMA is setup for streaming in the database
• The flag bit of the lookup table is stored on chip
• Reads are made to main memory only when a hit is identified
• The entire word finder functionality is pipelined in this core
CURRENT PORTABLE SYSTEM
SPECIFICATIONS OF THE CURRENT PORTABLE DESIGN

• Multiple cores implemented on the FPGA
• Each core is loaded with different query lookup tables’ flag bits
• Maximum number of cores that can be fit onto the FPGA is limited to eight
• This is due to limited number of Block RAMs available onchip
• Large parallelism is achieved
CURRENT RESULTS

![Graph showing current results with different system configurations and memory sizes.]

- 1 Core System
- 2 Core System
- 4 Core System
- 8 Core System
- Intel Pentium
- 2-Dual Core AMD Opteron

**Speedup**
- swissprot (400MB)
- wgs.38 (1GB)
- nt.00 (3GB)
- env.nt.00 (4GB)
CONCLUSION AND FUTURE WORK

• The initial implementation was very slow
• Multiple cores of the BLAST Word Finder function results in speedups
• With new FPGA boards comes more slices and more speedup
• Can think of porting the ungapped extension stage to Hardware
QUESTIONS