FPGAs

How a circuit is formed on an FPGA

**Configurable computing element (LUT)**
A look-up table for a simple Boolean operation

**Configurable interconnect element**
Configured to connect any LUT with any other LUT
Circuit design vs Programming

• The difference between using VHDL and C is much more than syntax
• Circuit design works in the physical world of real electronics
  – Like building a bridge
  – Continual judgments about cost and safety
• Software programming works in the abstract world of pure logic
  – Like mathematics
  – Purely logical description of problem
Executing software

• To achieve the abstract world of software, a device that physically embodies a model of execution is required
  – The device operates in the real world
  – Its operations form the pure abstract world of software
  – This is the essence of a computer processor
The Mitrion Platform

1) The Mitrion Virtual Processor (MVP)
   - A configurable processor design for a fine-grain massively parallel, soft-core processor
   - 10-30 times faster than traditional CPUs

2) The Mitrion-C programming language
   - An intrinsically parallel C-family language

3) The Mitrion Software Development Kit
   - Compiler
   - Debugger/Simulator
   - Processor configurator
   - IDE

(5) 05 May 2007, by Stefan Möhl
What the MVP gives you

• No hardware design considerations or judgments about physical electronics

• Real, fully abstract, software programming
  – No clock, the MVP is designed to run at 100MHz
  – The MVP keeps track of data and send it to the right place at the right time
  – No timing considerations whatsoever
  – No gates, drive-high, drive-low, ground, combinatorial paths, flip-flops, clocks, PLLs, DCMs, latches, etc, etc
What you need to do

• You must write a parallel program
• You have to give sufficient parallelism to feed the compute units in parallel
  – If you have 20 data parallel processing units, you need 20 independent data elements
  – If you have 20 pipelined processing units, you need 20 independent data elements
• Mitrion-C is designed to help you achieve this
The Mitrion Platform

```c
foreach (landmark in <0..1W>)
{
    int48 dx = px[landmark] - x;
    int48 dy = py[landmark] - y;
    int48 r2 = dx*dx + dy*dy;
    int48 ext = if[r2 == 0] 0
        else log(r2) * r2;
    distx = foreach (c in ext by 1)
    }
```
NCBI BLAST accelerated on the MVP
**Mitrion-Accelerated NCBI BLAST**

- Built on the NCBI “blastall” application
  - Identical interface and output options
- BLASTN algorithm is accelerated:
  - Critical parts ported
  - Runs on the Mitrion Virtual Processor in FPGA hardware
NCBI BLAST accelerated on the MVP

• Mitrion BLAST is open-source, released under GPL
  – Requires at least MVP on Virtex4-LX200
  – Not a Black Box – fully open-source: Anyone can add to it, change it, add “secret sauce”
  – Total Mitrion-C code is less than 1300 lines
  – Free to download on Sourceforge

• Newly released!

• 60x performance increase vs standard NCBI BLAST run on Opteron 2.8
NCBI BLAST accelerated on the MVP

Current main limitations

- All being worked on

• BLAST-N only
  - BLAST-P in development

• Currently limited to less than ~100K queries (unlimited database sizes)

• Gapped extension not run on MVP
  - Small percentage of run-time for most queries

• Only 11-length kernels
Basic architecture of NCBI BLAST

• Build a hash-table of all WMERs (11-basepair sequences) in query
• Produce all possible WMERs from database
• Remove the produced WMERs that do not exist in query
• Do ungapped extension of WMER hits
• Do gapped extension of WMER hits
Changes in the port to the MVP

• Minimal changes
  – Strong concern regarding repeatability of results with standard NCBI BLAST
  – Changes to parallelize code, but done in a “safe” way

• Works as a pre-filter
  – Inserted before the standard filters of NCBI BLAST, the full standard BLAST is run afterwards (but just on the hits that passed)
  – Removes a bit fewer than optimal number of hits
Mitrion BLASTN Performance

<table>
<thead>
<tr>
<th>Query</th>
<th>CPU Runtime (s)</th>
<th>CPU+FPGA Runtime (s)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>300s</td>
<td>17.28s</td>
<td></td>
</tr>
<tr>
<td>Query 2</td>
<td>300s</td>
<td>19.52s</td>
<td></td>
</tr>
<tr>
<td>Query 3</td>
<td>250s</td>
<td>16.34s</td>
<td></td>
</tr>
<tr>
<td>Query 4</td>
<td>300s</td>
<td>18.58s</td>
<td></td>
</tr>
<tr>
<td>Query 5</td>
<td>300s</td>
<td>16.61s</td>
<td></td>
</tr>
<tr>
<td>Query 6</td>
<td>300s</td>
<td>16.84s</td>
<td></td>
</tr>
</tbody>
</table>
The Mitrion-C Open Bio Project

- Open source development of key Bioinformatics applications for the Mitrion Virtual Processor at Sourceforge.net:

  http://mitc-openbio.sourceforge.net/

- Currently BLAST, others to come...
Contact Information

Mitrionics AB
Ideon Science Park
SE-223 70 Lund
www.mitrionics.com

stefan.mohl@mitrionics.com