FPGA SUPERCOMPUTING SOLUTION FOR PEPTIDE MASS FINGERPRINTING

D. Coca\textsuperscript{1}, I. Bogdan\textsuperscript{1}, R. Beynon\textsuperscript{2}
\textsuperscript{1}Department of Automatic Control & Systems Engineering, The University of Sheffield, Mappin Street, Sheffield S1 3JD, UK
\textsuperscript{2}Department of Veterinary Pre-Clinical Sciences, The University of Liverpool, Brownlow Hill, Liverpool L69 7ZJ.
d.coca@sheffield.ac.uk, i.bogdan@sheffield.ac.uk, r.beynon@liverpool.ac.uk

ABSTRACT
This paper presents a complete high-performance reconfigurable computing solution for real-time Peptide Mass Fingerprinting. The system, which implements both stages of computation associated with this protein identification technique - the processing of the raw mass spectrum and the database search - in FPGA hardware, can deliver the complete search results in less than a quarter of a second. This represents an almost 2000 fold speed up compared with an equivalent software implementation in C, running on a single 3GHz Xeon workstation.

KEY WORDS
Proteomics, Peptide Mass Fingerprinting, Reconfigurable Computers, Field Programmable Gate Arrays

1. Introduction
Peptide Mass Fingerprinting (PMF) is an established technique to identify proteins by matching the molecular masses of a subset of constituent peptide fragments, generated experimentally using mass spectroscopy, against theoretical proteolytic peptide maps derived from protein sequence database by \textit{in silico} digestion.

The Peptide Mass Fingerprinting approach is predicated on the assumption that a pattern of proteolytic peptide masses provide a quasi-unique signature for every protein in the database.

The peptide mass fingerprint is routinely generated with high accuracy using MALDI-TOF mass spectrometers.

The computations associated with the PMF approach involve two distinct processing stages: a) processing the raw MALDI-TOF mass spectrometric data to extract a peptide mass fingerprint and b) searching and matching the protein signature against a comprehensive database of known proteins.

At present, the capability of most mass spectrometers is such that the speed of acquisition of the mass spectrometric data is far greater than the time required for processing of the raw mass spectrum and the subsequent database search. MALDI-ToF mass spectrometers for example, boast acquisition rates of up to 200 spectra per second.

Thus, post-instrument data processing is already a major bottleneck in proteomics workflow, and as experimental design, instrument performance and user skills all increase, we might expect this to become worse.

A reasonable goal, if PMF is to remain a key method in proteomics, would seem to be that the processing and first level (non interpretive) analysis of data should be completed within the same timeframe as the experiment itself - this would give ‘real-time PMF’ (RTPMF).

At the same time, data processing solutions are probably best implemented ‘near-instrument’, where the end user has the option of controlling search strategies and quantification, rather than widely distributed multiprocessor search solutions.

As the goal of near-instrument RTPMF is unlikely to be achieved by computing platforms based on a small number of conventional or multi-core microprocessors, which currently cannot deliver the required processing speed, an effective approach is to implement proteomics data processing algorithms in computer hardware, effectively designing and implementing computer chips that have a specific dedicated function. This approach is based on the use of devices termed field programmable gate arrays (FPGAs) and requires the implementation of existing or new algorithms in the form of highly parallel and reconfigurable digital hardware designs.

FPGAs are large-scale integrated circuits that can be programmed (and re-programmed) to implement a custom digital hardware design after they have been manufactured.

FPGAs are well suited for high-performance, high-bandwidth and parallel processing applications. FPGAs have been successfully employed to speed up DNA sequencing algorithms ([1],[2],[3], [4], [5],[6].
More recently, FPGAs have been used to accelerate sequence database searches with MS/MS-derived query peptides [7].

This article summarizes the results of a successful project which has led to the development of the first complete reconfigurable computing (RC) solution for protein identification based on MALDI-TOF data. The FPGA-hardware solution, which incorporates a raw mass spectra processor and a parallel search engine, delivers a peptide mass fingerprint match in fewer than 240 milliseconds when searching the entire MSDB protein database.

2. Reconfigurable Computing Platform

The FPGA computational engines were designed and optimized to run on a modular and scalable reconfigurable hardware platform, consisting of a reconfigurable FPGA motherboard equipped with a Xilinx Virtex-II XC2V8000 FPGA (8 million gates) and 4Mbytes RAM, communicating with the host PC server via a PCI interface. On the motherboard there are two FPGA devices. The bigger one (Virtex-II XC2V8000 FPGA) is used to implement user designs – in our case the spectrum processor.

The PCI interface (firmware) between the server PC and the user FPGA from the motherboard is coded on a smaller Xilinx Spartan-II FPGA. Communication between these two FPGA devices is at 40MHz on a 32 bits wide data bus. A block diagram of the reconfigurable system is shown in Figure 1.

Figure 1. Block diagram of the multi-FPGA reconfigurable computing system

The motherboard can be configured to have up to three additional FPGA modules that can be plugged into dedicated motherboard slots. One of such modules has been used to implement the database search. Each FPGA module has one Virtex-II XC2V8000 FPGA device and 1GB of DDR SDRAM that can hold the entire MSDB protein database (currently the encoded MSDB database is stored on a single module and takes about 680MB). Each module is connected with the motherboard FPGA and with the other two modules via a 64 bit, 66MHz local bus. This architecture enables the implementation of parallel searches at FPGA level as well as across modules.

The current hardware implementation incorporates a raw mass spectra processor and a parallel search engine which delivers a peptide mass fingerprint match in fewer than 240 milliseconds when searching the entire MSDB protein database.

3. Mass Spectrum Processor

The raw spectra processor [8] implements specific signal processing operations (de-noising, baseline correction, peak identification and de-isotoping) in order to generate the peptide mass fingerprint.

The block diagram of the mass spectrum processor is depicted on Figure 2. The implementation has two major functional blocks: a peak detection unit, which identifies all significant spectral peaks and a peptide identification unit that generates the final list of peptide masses and associated abundances.

The FPGA spectra processor developed by Bogdan et al. [8] was designed to implement, with some variations, the algorithm proposed by Samuelsson et al. [9]. The major difference is the method used by the FPGA processor to implement aggregation of natural isotopomers (due primarily to the natural abundance of $^{13}$C and $^{15}$N) - the algorithm implemented in FPGA uses Poisson distributions to approximate the isotopic patterns for every peptide [10].

![Figure 2. Block diagram of the mass spectra processor](image-url)

4. Database Search Engine

The database search engine traces the peptide fingerprint back to the originating peptide by matching it against the expected (theoretical) peptide masses obtained by digesting in silico - on the fly - all protein sequences in the database. The protein sequence databases (MSDB for example) are in fact flat text files.

In order to fully exploit the benefits of FPGA acceleration, the entire MSDB database was encoded, using a 5-bit representation for all the 20 constituent
aminoacids, and stored in the local on-board memory of the FPGA module in a format that facilitates fast parallel searches.

By encoding the database more efficiently using only 5-bit “characters”, the database size was reduced by about 40%. The resulting “shrunk” database occupies at the moment only 60% of the total 1GB DDR SDRAM memory installed on the FPGA module.

One effect of storing the database in the local module memory is that it eliminates a significant memory access bottleneck, due to the PCI interface, that would otherwise be present if the protein database were stored in the computer memory. However, the most significant reason for encoding and storing the protein database in the local memory is that it enables parallel processing of protein sequences. In the current implementation there are 48 protein sequences that are streamed out, in parallel, from the memory. Each protein sequence is processed sequentially by a search processor implemented in the module’s FPGA. A block diagram of the FPGA search engine is illustrated in Figure 3.

Each search processor performs two basic operations: a) the computation of theoretical peptide masses for every protein in the database by in-silico digestion and b) the subsequent computation, for each protein, of a matching score which indicates the likelihood that the peptide mass fingerprint generated by the mass spectrum processor, based on experimental data, belongs to that particular protein.

The digestion unit can be programmed to implement different cleavage site rules according to the particular digestion enzyme used in practice – typically trypsin.

In the computation of theoretical peptide mass spectrum the digestion unit can be programmed to account for possible Post Translational Modifications (PTMs). In the current implementation, only fixed PTMs are allowed. Variable PTMs will also be implemented in the near future.

The scoring unit calculates the number of peptide masses in the peptide mass fingerprint that are matched for every digested protein in the database. A user defined mass matching tolerance can be programmed by the user to account for MS instrument precision and other sources of errors that may affect the accuracy of the peptide mass fingerprint. The database search engine is presented in more detail in [11].

5. Results

In order to validate the accuracy of the hardware implementation, reference C programs were developed for all the algorithms implemented in hardware to process the raw mass spectra and perform database searching and matching. The first validation step involved checking that both the hardware and software implementation generate identical results. The second validation step involved, in the case of the mass spectra processor, comparing the results with commercial software solutions such as MassLynx and assessing the ability to correctly resolve isotopomer distributions derived from asparagine containing peptides and the deamidated cognate peptide (see [8] for more details).

The database search engine was also tested using theoretical peptide mass fingerprints derived from randomly selected proteins in the database.

The performance gains of the hardware implementation relative to the conventional software solution were evaluated by measuring the execution time of the main computational loops of the reference C implementations and comparing this with the FPGA processing time. It should be noted that the time elapsed for initializations of memory locations before the effective processing of data and disk access time are not included in the software processing time. Only the processing of effective algorithmic steps was measured. The reference design was compiled in C and was simulated on a dual 3.06GHz Xeon processor server. Each C simulation was repeated 30 times and the average processing time was used for comparison.

The speed gains corresponding to the mass spectra processor, for different lengths of the analysed mass spectrum are summarized in Table 1.

<table>
<thead>
<tr>
<th>Spectrum size</th>
<th>Processing time [ms], dual Xeon 3GHz processors</th>
<th>Processing time [ms], Virtex-II FPGA, 180MHz clock</th>
<th>Speed Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>25488</td>
<td>20.27</td>
<td>0.1632</td>
<td>124.20</td>
</tr>
<tr>
<td>50448</td>
<td>31.23</td>
<td>0.3105</td>
<td>100.56</td>
</tr>
<tr>
<td>75168</td>
<td>47.33</td>
<td>0.4557</td>
<td>103.86</td>
</tr>
<tr>
<td>101040</td>
<td>62.50</td>
<td>0.5607</td>
<td>111.46</td>
</tr>
<tr>
<td>125184</td>
<td>79.17</td>
<td>0.7557</td>
<td>104.76</td>
</tr>
<tr>
<td>150144</td>
<td>114.33</td>
<td>0.8547</td>
<td>133.76</td>
</tr>
<tr>
<td>175104</td>
<td>130.20</td>
<td>1.0024</td>
<td>129.88</td>
</tr>
<tr>
<td>200976</td>
<td>188.63</td>
<td>1.1219</td>
<td>168.13</td>
</tr>
</tbody>
</table>
Acknowledgements

This work was funded by the Biotechnology and Biological Sciences Research Council (BBSRC), UK. The authors gratefully acknowledge the support of Xilinx Inc. who donated the devices and design tools used in this study.

References


6. Conclusion

This paper has outlined a complete FPGA hardware solution for Peptide Mass Fingerprinting which significantly outperforms software solutions designed to run on conventional microprocessor platforms. Whilst in the case of the raw spectra processor, we found that speed gains compared with single processor machines are almost 200, in the case of the database search the speed gain is almost 2000. The comparisons are made by implementing in C the same algorithm that is implemented in hardware.

The processing times associated with mass spectra processing are relatively insignificant and do not exceed 1.2 milliseconds even for a very large spectrum. The database search takes about 240 milliseconds so that overall, the FPGA solution can perform peptide mass fingerprinting in less than a quarter of a second. In other words, the FPGA system can process and identify four mass spectra per second so essentially it provides results in real time.

Computational time can be further reduced by distributing the database search over different FPGA modules. With the current board for example it should be possible to achieve processing speeds of 12 spectra per second. Further expansion is possible if more than one FPGA motherboards are configured.