BIOINFORMATICS ORIGINAL PAPER

Vol. 23 no. 6 2007, pages 724–731
doi:10.1093/bioinformatics/btl656

Gene expression

Hardware acceleration of processing of mass spectrometric data for proteomics

Istvan Bogdan[1], Daniel Coca[1,*], Jenny Rivers[2] and Robert J Beynon[2]

[1]Department of Automatic Control & Systems Engineering, The University of Sheffield, Mappin Street, Sheffield S1 3JD, UK, and [2]Proteomics and Functional Genomics Group, Faculty of Veterinary Science, University of Liverpool, Crown Street, Liverpool L69 7ZJ, UK

Received on September 25, 2006; revised on November 23, 2006; accepted on December 21, 2006

Advance Access publication February 3, 2007

Associate Editor: Golan Yona

ABSTRACT

Motivation: High-resolution mass spectrometers generate large data files that are complex, noisy and require extensive processing to extract the optimal data from raw spectra. This processing is readily achieved in software and is often embedded in manufacturers' instrument control and data processing environments. However, the speed of this data processing is such that it is usually performed off-line, post data acquisition. We have been exploring strategies that would allow real-time advanced processing of mass spectrometric data, making use of the reconfigurable computing paradigm, which exploits the flexibility and versatility of Field Programmable Gate Arrays (FPGAs). This approach has emerged as a powerful solution for speeding up time-critical algorithms. We describe here a reconfigurable computing solution for processing raw mass spectrometric data generated by MALDI-ToF instruments. The hardware-implemented algorithms for de-noising, baseline correction, peak identification and deisotoping, running on a Xilinx Virtex 2 FPGA at 180 MHz, generate a mass fingerprint over 100 times faster than an equivalent algorithm written in C, running on a Dual 3 GHz Xeon workstation.

Contact: D.Coca@sheffield.ac.uk

1 INTRODUCTION

The phenomenal advances in proteomics that have been made in recent years are readily attributed to advances in mass spectrometry (MS), notably soft ionization modes and tandem instrumentation, coupled with new tools for processing spectral data and database searching. The sensitivity and selectivity of the current generation of mass analysers is notable, and useable mass spectra can be recovered from vanishingly small amounts of material. Perhaps the simplest MS method in proteomics is that of peptide mass fingerprinting (PMF). PMF is a protein identification technique in which a protein is proteolyzed using an endopeptidase of defined specificity (usually trypsin) and the masses of the ensuing limit peptide fragments are measured. The proteins are identified by matching the measured molecular masses of these peptide fragments against theoretical peptide mass profiles generated from protein sequence database. PMF is readily delivered at high sensitivity through routine instrumentation such as MALDI-ToF mass spectrometers and although tandem MS approaches can recover more information from single peptides, PMF still plays an important role. Indeed, as more genomes are sequenced, and cross-species matching methods are developed, PMF may assume greater importance for many sub-proteome studies.

PMF involves two basic operations. The first is processing of the raw mass spectrum to derive a mass fingerprint, generating a data set in which the only variable is the mass of each peptide (relative intensities of different ions are not routinely used in PMF). When the mass spectrum has been processed, the list of masses are first filtered to remove spurious masses such as those derived from trypsin or matrix clusters, and the remaining peptide masses constitute the fingerprint that is used to search the protein databases for a possible match. A correlation score is computed between the database entries and the unknown peptide fragment mass list. The matches with the highest score form the final candidate protein list to be returned to the user.

At present, the time required for processing of the raw mass spectrum and the subsequent database search can exceed that of acquisition of the mass spectrometric data, especially by MALDI-ToF, which boasts acquisition rates of up to 200 spectra/s. If PMF is to remain as a key method in proteomics, one compelling gain would be a system in which the raw spectra are processed and searched against the protein database in the same time frame as acquisition—this would give 'real-time PMF' (RTPMF). However, for the goal of RTPMF to be realized, there remains the need for substantial acceleration of these two stages (spectrum processing and database searching).

A very effective approach to speed up the computations is based on the development of dedicated hardware processors that are optimized to perform specific algorithms. The acceleration, compared with the standard sequential microprocessor, is achieved by concurrent implementation of different arithmetic and logic operations that make up a computational loop and by concurrent execution of several computation loops. A major drawback of this approach used to be the prohibitive costs associated with manufacturing a dedicated integrated circuit (ASIC).

*To whom correspondence should be addressed.

However, the hardware implementation has become a much more cost effective solution due to the availability of high-density field programmable gate arrays (FPGAs) and of high-level system design and development tools, which make possible the implementation of very complex hardware designs with almost the same ease as the software implementation. An FPGA is a large-scale integrated circuit that can be programmed (and re-programmed) after it has been manufactured. Early attempts to use FPGA devices in bio-computation were made to accelerate gene sequence analysis (Fagin *et al*., 1993). FPGAs, which are well suited for high-performance, high-bandwidth and parallel processing applications, have been successfully employed to speed up DNA sequencing algorithms (Hughey 1996; Guerdoux-Jamet *et al*., 1997; Wozniak 1997; Lavenier, 1998; Guccione *et al*., 2002; Simmler *et al*., 2004). FPGAs were also used in the attempt to accelerate search of substrings similar to a template in a proteome (Marongiu *et al*., 2003). More recently, FPGAs have been used to accelerate sequence database searches with MS/MS-derived query peptides (Anish *et al*., 2005). This hardware-based solution can reportedly locate a query within the human genome about 32 times faster than a software implementation running on a 2.4 GHz processor. A hardware sequence alignment tool implemented in FPGA is also available (Oliver *et al*., 2005).

In addition to developing approaches for real-time database searching, we have implemented FPGA solutions for processing of raw mass spectra. This article describes the design and hardware implementation of a mass spectrum processor which performs all computational tasks involved in the generation of a mass signature from a raw spectrum namely, smoothing, peak detection and the coalescence of natural isotopomers into a single mass ('deisotoping'). The mass spectrum processor, which is implemented on a Xilinx XC2V8000 FPGA and runs at 180 MHz, achieves more than 100-fold speed-up compared with a C software implementation running on a dual 3 GHz Xeon Server with 4 GBytes of memory.

2 METHODS

A MALDI-ToF mass spectrum of a typical tryptic digest of a protein generates pairs of mass-to-charge (m/z) and abundance values. Typically, the number of points in the spectrum ranges from a few thousand to a few hundred thousand. The determination of experimental peptide masses (the so called peptide mass fingerprint) requires relatively complex processing of the raw mass spectrum in order to discriminate between spectral peaks that correspond to digested peptides and the associated isotopomer peaks and the spurious peaks caused by noise and sample contamination.

To create the raw data used to evaluate the FPGA implementation, single proteins and complex protein mixtures were diluted with 50 mM ammonium bicarbonate and digested with trypsin at a ratio of protein: enzyme of 50:1. One protein that was used was an artificial QconCAT protein chosen designed so that all tryptic fragments fell within the range 1000–2500 m/z (Beynon *et al*., 2005; Pratt *et al*., 2006). Digestion was carried out at 37°C for 24 h after which time, 1 µl digested material was spotted onto a MALDI target. This was mixed with 1 µl α-cyano hydroxycinnamic acid matrix and analysed using a Micromass M@LDI mass spectrometer (Waters, Manchester, UK) typically over the m/z range 800–4000.

The FPGA spectra processor was designed to implement, with some variations, the algorithm proposed by Samuelsson *et al*. (2004). 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 (Breen *et al*. 2000).

The algorithm described in Samuelsson *et al*. (2004) has several steps. First the baseline and noise levels are estimated over an arbitrary interval adjusted by user parameters, (ω and Ω) which divides the raw spectrum. The data points in the spectrum are classified compared with the level of noise and the baseline into noise, baseline and signal points. Then, peaks are constructed from a group of data points where at least one point has to be signal. In the next step the constructed peaks are grouped into clusters that are further processed to identify the monoisotopes.

The deisotoping algorithm proposed by (Breen *et al*., 2000), was preferred for FPGA implementation. Here Poisson modelling is applied to determine monoisotopic masses (deisotoped peaks) from isotopically resolved groups (clusters). The abundances of the higher isotopic contributions for a monoisotopic peak are computed using Poisson distribution models that have been shown to match very well theoretical distributions (Breen *et al*., 2000).

A good test of such an algorithm is in the deconvolution of the overlapping mass spectra of a peptide containing an asparagine residue ('amide') and its cognate acid product in which the side chain amide residue has been deamidated. The two mass spectra overlap by 1 Da, and effective deconvolution would be able to apportion the signal into the relative proportions of acid and amide.

The block diagram of the hardware processor is depicted in Figure 1. 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 peak detection unit implements smoothing, baseline and noise level estimation in order to discriminate between signal and noise peaks. The first block implements a Savitzky–Golay smoothing filter (Savitzky *et al*., 1964) with a user-defined window that can be chosen according to the instrument resolution (number of data points recorded per 1 Da). The smoothing operation is optional; the user can specify if the data is pre-processed or not. The Savitzky–Golay smoothing operation is implemented as a convolution of the filter coefficients with the abundance input stream. A delay equal to the filter latency (DELAY A in Fig. 1) is applied to the mass values data stream in order to preserve synchronicity between the mass and abundance values. The number of coefficients depends on the chosen window size. The maximum window size allowed is 43 which correspond to 43×43 coefficients. The coefficient matrix is loaded into the FPGA memory before processing operation starts. The smoothing operation is implemented as a single channel parallel filter using a Xilinx Logicore block (Xilinx, 2004). Smoothing improves the shapes of peaks which helps peak detection but slightly degrades the abundance values. Minor corrections can however be implemented to compensate the small diminution in abundance. Figure 2 shows the effect of smoothing a segment of real data with a Savitzky–Golay filter that has a polynomial order of 11 and window (or frame) size of 23.

The Y, Z, W computation block computes the minimum (Z), maximum (Y) abundances and their difference ($W = Y - Z$) over a small sliding window of maximum length of $\omega = 16$ points, as described in Samuelsson *et al*., (2004). It has a structure similar to that of a median filter that sorts in ascending order its input data stream over its filter length. Instead of computing the median, the maximum and minimum values are found.

The baseline and noise estimation block uses Y,Z,W values to compute the baseline (Y base) and noise level (Y noise) over a larger moving window of length $\Omega \times \omega$ points, where Ω is a user-defined

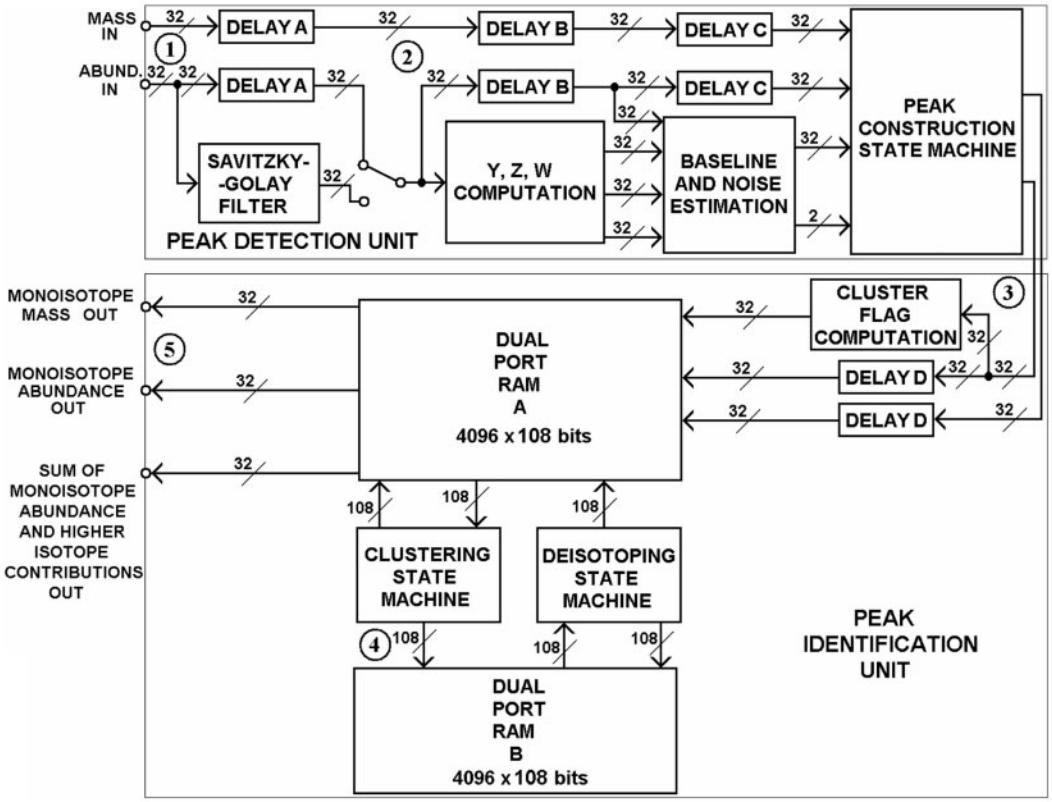

Fig. 1. Block diagram of the mass spectra processor implemented in FPGA.

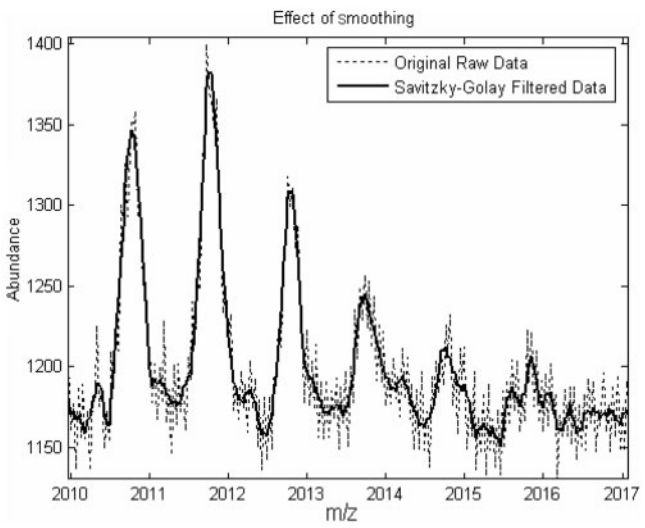

Fig. 2. Raw (dotted line) and Savitsky–Golay smoothed (solid) mass spectrum segment.

parameter (Samuelsson *et al.*, 2004). The (Y noise) and (Y base) values are used to compute the signal-to-noise ratio (S_n) for every data point in the spectrum. This is compared with a threshold (τ_1), which can be set by the user. Each abundance value is classified as noise, support or signal depending on whether the associated S_n value is $S_n < 1$, $1 < S_n < \tau_1$ or $S_n > \tau_1$, respectively. The outputs of the baseline and noise estimation block are a 2-bit classification flag (flag = 1-noise, flag = 2-support and flag = 3-signal) and the estimated baseline (32 bits) calculated for each abundance value. This data stream is aligned with the original mass-abundance pairs of the input spectrum (Fig. 1).

The peak construction state machine generates a list of valid peaks based on the 2-bit classification flag. A peak is defined as the set of mass/abundance data pairs that are either support or signal (flag = 2 or flag = 3) and are bounded by noise (flag = 1). The mass and abundance associated with each identified signal peak are calculated as the centred mass and the (baseline subtracted) peak maximum, respectively. The effect of the baseline subtraction is shown in Figure 3. The resulting peak list is written in a dual port RAM block for further analysis.

The peptide identification unit consists of a clustering and a deisotoping unit. The peaks in a cluster correspond to the isotopes of one or more singly charged chemical compounds, separated by the mass of a neutron. Clustering involves grouping together peaks so that the m/z distance between two successive peaks is between $1 - \tau_2$ and $1 + \tau_2$ where τ_2 is another user selectable value, typically set to 0.2 (Samuelsson *et al.*, 2004).

The first block of the clustering unit ('Cluster flag computation') computes the distance between all consecutive signal peaks from a distance of $1 + \tau_2$ starting with the lowest mass value m_1. To speed up computations, there are p circuits that compute mass differences $m_2 - m_1, \ldots, m_{p+1} - m_1$ between m_1 and the following p consecutive mass values $m_1 < m_2 < \ldots < m_p < m_{p+1}$ in parallel. In our design, p is an adjustable parameter, which is selected according to mass spectrometer resolution, to be equal to the maximum number of signal peaks that are registered within a window of $1 + \tau_2$. The parallel processing of these peaks is implemented by a FIFO (first in first out queue) structure of length p. The data flow through this circuit is depicted in Figure 4.

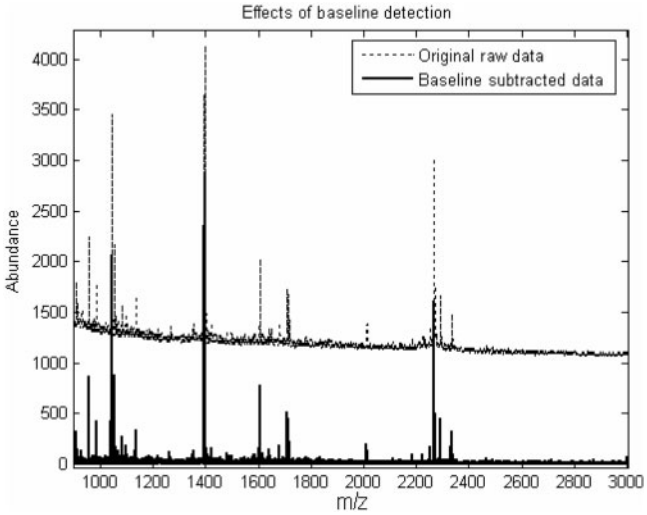

Fig. 3. Effects of baseline detection.

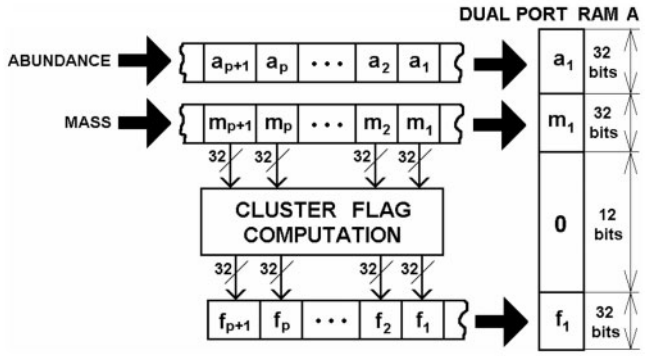

Fig. 4. Data flow for cluster flag generation.

This is the only example of explicit algorithm parallelization. The main approach used to speed-up computations is through instruction pipelining. This technique is particularly suited for mass spectra processing where the same sequence of operations is applied to a long data stream.

In the current configuration, for each mass value that is being processed, the clustering unit generates a p-bit cluster flag f. Typically about 50–100 samples per m/z unit are taken, so the FIFO length is greater or equal to the number of possible peaks in 1.2 m/z distance. Assuming that at least three measurement points define a signal peak, and there are 100 measurements in a unit of m/z, the maximum number of constructed peaks in one m/z unit is 33 and in 1.2 m/z unit is 40. In the current design, p is 32.

If the distance between m_1 and m_k is within the range $1 \pm \tau_2$, the k-1 bit of this word is set to 1 indicating that m_k is a potential isotopomer of m_1. The mass values, abundance values and associated cluster flags are concatenated and stored in RAM (A) at consecutive memory locations (increasing mass) as a $32+32+32+12$ bit word (32 for mass, 32 for abundance, 32 cluster flag and 12 bits reserved for further processing) as shown in Figure 5. These records are processed to group signal peaks into clusters. If all the bits of the f_1 flag corresponding to m_1 are zero this indicates that m_1 has no isotopes. If the kth bit of this word is 1 this indicates that the peak corresponding to the m_{k+1} mass value is part of the cluster having m_1 as the monoisotopic ion. Next, by analysing the

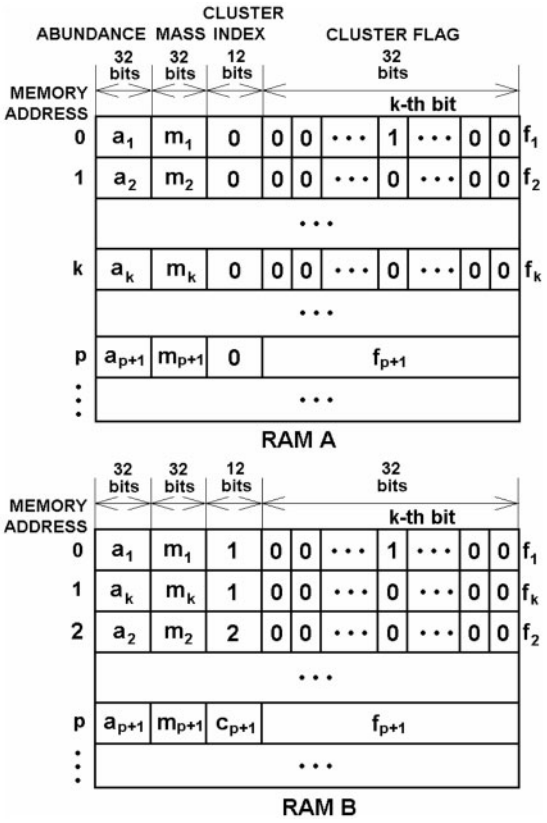

Fig. 5. Memory operations during clustering.

index f_k associated with m_k, it is possible to identify other peaks that belong to the same cluster. The process continues until the cluster flag associated with the last signal peak added to the cluster has only zero entries.

Clustering is implemented as a state machine that sequentially analyses the data stored in the first dual port RAM(A) and generates clusters that are stored in the second dual-port RAM(B). Each RAM block was configured to have 432 Kbits storage space (4 Kb address space and 108 bits word length) which can store 4096 peaks. The cluster flag associated with the latest signal peak added to the cluster is used to compute the memory address of the next peak to be included in the cluster from RAM(A). The peaks identified as belonging to one cluster are stored at consecutive memory locations in RAM(B). Each cluster is also indexed with a number of 12 bits called cluster index, a unique identification value for every cluster, stored with the member peaks. The memory content of RAM(A) before clustering and RAM(B) after clustering is shown in Figure 5. The first peak from address 0 has a single bit set to one at the k-th position of its cluster flag. This indicates that the peak stored at address k is part of the same cluster as the first peak. As a result the peaks 1 and k+1 will be stored in RAM(B) at successive memory locations. In addition, to indicate that they are part of the first cluster, both peaks will have their 12 bits cluster indexes set to 1, The cluster will not include more peaks because the peak at address k has its cluster flags null. The clustering process continues until all the peaks from RAM(A) are visited. The result in RAM(B) will be the same list of peaks, this time ordered by increasing cluster indexes and increasing mass values for peaks belonging to the same cluster.

For example, a cluster of eight peaks is shown in Figure 6. The peaks are situated at $\sim$1 Da distance with m/z values: $m_1 = 2265.35$,

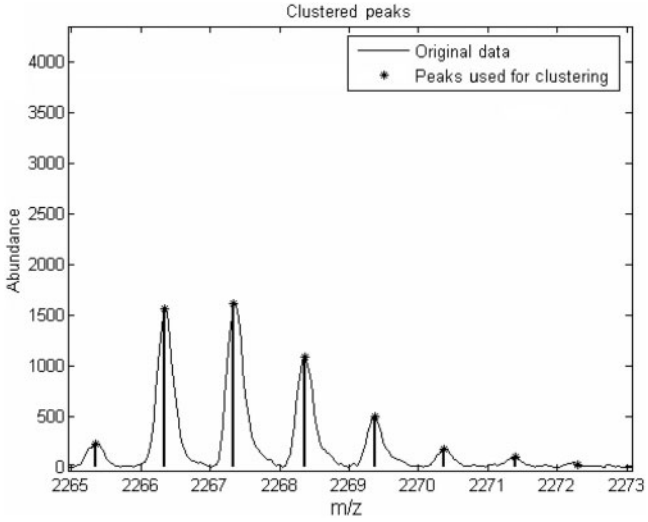

Fig. 6. Results of clustering.

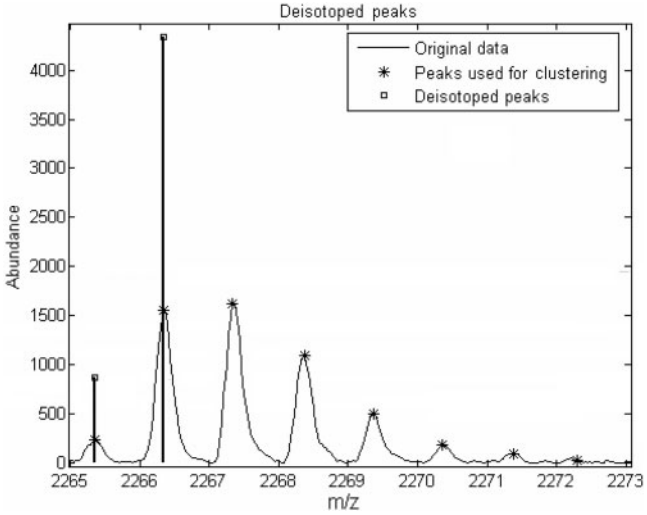

Fig. 7. Deisotoping results. The cluster of peptide ions in the Figure are deconvoluted into two species with overlapping isotopomer profiles.

$m_2 = 2266.35$ Da, $m_3 = 2267.34$ Da, $m_4 = 2268.37$ Da, $m_5 = 2269.37$ Da, $m_6 = 2270.36$ Da, $m_7 = 2271.39$ Da, $m_8 = 2272.29$ Da. In practice such a cluster has to be processed further since one cluster may contain more than one monoisotope, that is, the peaks in a cluster can be viewed as a superposition of isotopic distributions.

The deisotoping unit isolates all monoisotopic masses in a cluster and calculates the total abundance of the peptide by summation of the intensities of all of the isotopomers. The hardware algorithm implements deisotoping based on an approximation of isotopic patterns by a Poisson distribution as proposed by Breen *et al*., (2000). Starting with the first peak in a cluster (usually the monisotopic peptide), the algorithm generates the theoretical isotopic distribution based on peak height (abundance) and mass value. The computed abundance values are then subtracted from the original peaks at the corresponding m/z values. Following subtraction, any abundance value below a user-specified threshold is set to zero. The step is then repeated, with the remaining (height adjusted) peaks. At each step, the monoisotopic mass value, the original detected abundance and the total abundance are recorded in the final peak list. The deisotoping unit processes previously computed clusters from the dual port RAM (B), writes back partial results in RAM (B) and the final peak list in RAM (A).

When the last peak from the cluster is visited, the total peptide abundance for every monoistope detected is stored back to RAM(A), in the 32 bit area previously used to store the cluster flag f_{k+1}. At the same time, the cluster index information (last 12 bits) is overwritten with a flag set to 1 or 0 depending on whether the monoistopic peak at that address is above or below the threshold τ_3.

For example, the monoisotope peaks recovered from the previously clustered fragment displayed in Figure 6 are shown in Figure 7. In this example, two consecutive overlapping monoisotopomers are detected. The abundances of all the higher isotopes of each monoisotope are added to its original monoisotopic abundance. The first monoisotope has its summed abundance of 867.7146 while for the second monoisotope the corresponding overall abundance is 4333.5933. When processing ends, the harvested peak list in RAM(A) is ready to be used to search the protein database.

3 RESULTS

The processor was implemented on a FPGA motherboard equipped with a Xilinx Virtex-II XC2V8000 FPGA (8 million gates) and 4 Mbytes ZBT RAM, communicating with the host PC server via a PCI interface (32 bit, 33 MHz).

On the motherboard there are two FPGA devices (Fig. 8). The bigger one (Virtex-II XC2V8000 FPGA) is used to implement user designs—in our case the spectrum processor. The Xilinx Spartan-II FPGA implements the PCI interface between the server PC and the user FPGA from the motherboard. Communication between these two FPGA devices is at 40 MHz on a 32 bits wide data bus. The motherboard has 4 MB of ZBT RAM connected to the user FPGA as shown in Figure 9. This is enough to store 512 K samples of mass-abundance pairs on 32 bits each.

The actual design occupies about 70% of the FPGA's logic resources and 18% of the FPGA's I/O resources. The server is a Dual 3.06 GHz Xeon processor machine with 4 GBytes RAM. The block scheme of the system is given in Figure 8.

The mass spectrum is transferred into the ZBT RAM via the PCI interface in two steps, first the mass and second the abundance data vectors.

The design can be easily ported, however, to a new version of the motherboard that supports PCI-X standard (64 bits at 133 MHz) which will allow transfer of the abundance and mass data streams at the same time.

All arithmetic operations are performed using 32-bit signed fixed-point binary number representation of mass and abundance values, with 12 bits after the radix point.

3.1 Spectral processing

The basic steps in processing a mass spectrum are largely the same, irrespective of the software used: smoothing, baseline subtraction and centroiding/deisotoping. For the FPGA based approach to be useful, the quality of the processed spectra should be at least as acceptable as those processed by software. A recombinant protein designed as an internal standard for multiplexed absolute protein quantification (Beynon *et al*., 2005; Pratt *et al*., 2006) was digested with trypsin to release

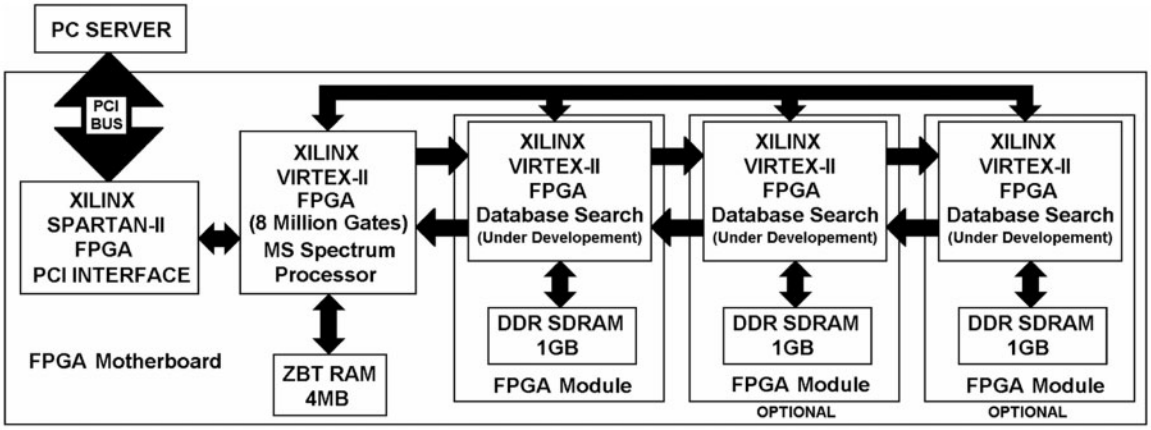

Fig. 8. Block scheme of the system.

20 limit peptides of known identity. Digested material was analysed by MALDI-ToF MS and raw data was processed separately using MassLynx, a commercial mass spectrometry software, and FPGA.

Data was processed using MassLynx software to remove background noise using polynomial order 10 with 40% of the data points below this polynomial curve and a tolerance of 0.01. Spectral data was also smoothed by performing two mean smooth operations with a window of three channels. The processed spectra were compared as a scatterplot of the centroid intensity values (relative to base peak) for data analysed in each way. The centroided spectra are highly comparable and the FPGA identifies the same peaks as the commercial product (Masslynx). Moreover, the intensities of the different peaks correlated well, irrespective of the method used to process the spectrum (Fig. 10) identified by the software.

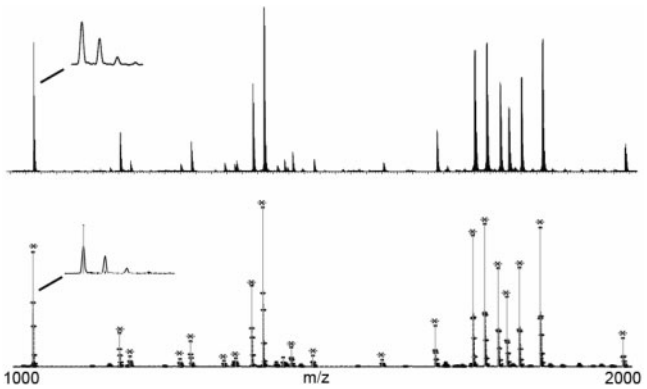

Fig. 9. A recombinant protein designed as an internal standard for multiplexed absolute protein quantification (Beynon *et al.*, 2005; Pratt *et al.*, 2006) was digested with trypsin to release 20 limit peptides of known identity. Digested material was analysed by MALDI-ToF MS and raw data was processed separately using MassLynx software and FPGA.

3.2 Deconvolution

A valuable test of effective deisotoping is provided by the resolution of isotopomer distributions derived from asparagine containing peptides and the deamidated cognate peptide. If a peptide contains the sequence Asn-Gly in particular there is a marked propensity for this to be converted non-enzymically to Asp-Gly, with the result that the deamidated peptide is 1Da heavier ($-NH_2$ to $-OH$). We tested this part of the analysis using MALDI-ToF peptide mass spectra derived from in-gel digestion of glyceraldehyde 3 phosphate dehydrogenase. A peptide generated by tryptic digestion has the sequence VKVGV**N**GFGR (monoisotopic mass 1031.59 Da, creating a singly charged ion $[M+H]^+$ of 1032.59 m/z) which is readily deamidated to VKVGV**D**GFGR (monoisotopic mass 1032.57 Da, creating a singly charged ion $[M+H]^+$ of 1033.57 m/z). To assess the ability of the FPGA implementation to deconvolute complex and overlapping spectral data, we generated a set of spectra for this peptide. The 1Da mass shift on deamidation generates a series of mass spectra that are strongly overlapping (Fig. 11).

Previously, we have assessed the proportion of acid and amide by a non-linear least squares iterative curve fitting procedure that explains the observed mass spectrum by optimizing the proportion of acid and amide variants, from theoretical spectra for the acid and amide species generated using the Protein Prospector MSIsotope tool (http://prospector.ucsf.edu/ucsfhtml4.0/msiso.htm). The correlation between the calculation of acid:amide proportion was exactly the same, irrespective of whether the FPGA implementation or the non-linear least squares method was used (Fig. 12). Thus, the hardware solution was able to deconvolute overlapping spectra with ease and yield the same results as previous methods.

3.3 Speed gains

The impact of the spectral length on processing time was measured using spectra with various lengths but constant isotopic composition and noise levels. The reference design was compiled in C and was simulated on a dual processor server using 3.06 GHz Xeon devices running Windows XP Professional operating system. The FPGA processor had an internal clock frequency of 180 MHz. In order to evaluate how the number of

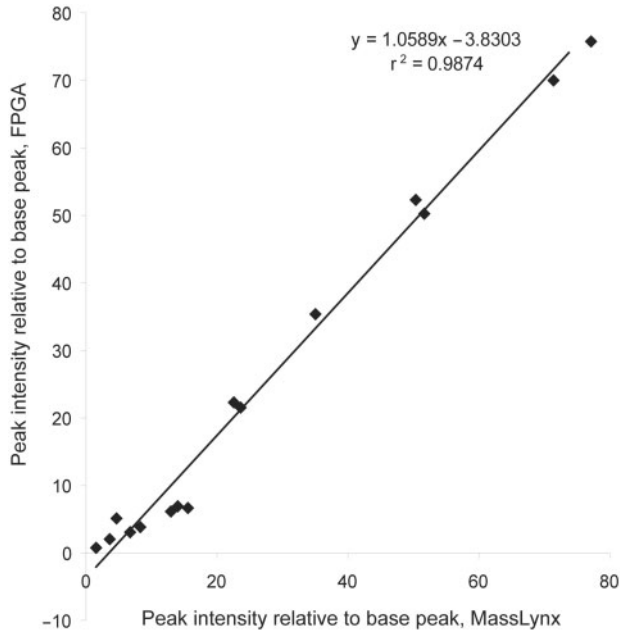

Fig. 10. Measurement of peak intensity by FPGA and commercial software.

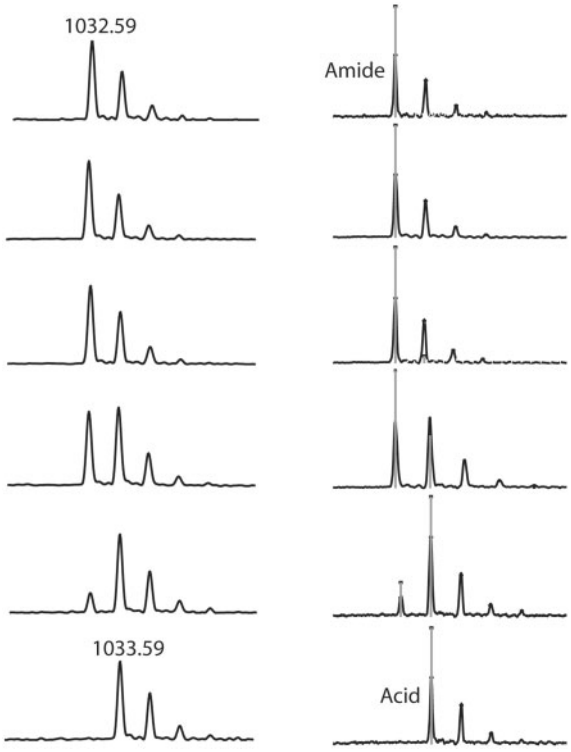

Fig. 11. Test data for deconvolution of mass spectra. A series of spectra were acquired for a peptide that undergoes deamidation using MALDI-ToF mass spectrometry (left hand column). After processing the data with the FPGA implementation, the proportions of acid ($[M+H]+=1032.59\,m/z$ and amide ($[M+H]+=1033.59\,m/z$) were calculated, and indicated on the processed spectra by vertical drop lines, headed by asterisks (right hand column).

data points in a spectrum relates to speed gain, spectra with different number of mass-abundance pairs were processed. The software processing routine was repeated 30 times for each mass spectrum data set and timed. The average time was used to calculate speed gains. It should be mentioned that the software processing time does not account for data transfers and memory initialization operations. Only the main computational loop was timed. Initializations for example, add on average 30 ms to the C processing time. The results are summarized in Table 1.

The average speed gain for processing spectra of different lengths is 122. Of course, implementations in instrument manufacturers' software are somewhat slower, and spectral processing such as obtained here can take several tens of seconds.

It is interesting to note that on a single processor server having the same configuration as the dual processor—except of the number of processors—the average time of processing the largest spectrum of 200 976 mass-abundance pairs was 204.71 ms which corresponds to a speed gain of about 180.

Processing time is less dependent on the number of signal peaks in the mass spectrum. Although, clustering and deisotoping processes are time consuming and depend on the spectral composition (i.e. the higher the isotopic abundance, the larger the number of iterations that have to be performed), the number of monoisotopes and their isotopic contributions is far less then the entire spectrum data. As a consequence, peak identification, which involves processing the entire spectrum, represents the most time consuming operation, giving the bulk of the total processing time.

4 DISCUSSION

We have successfully demonstrated that processing of a mass spectrum can be very effectively implemented as a hardware solution in a high-density FPGA. The performance is comparable in terms of quality of the processed spectrum, and spectra can be processed at much higher rates than obtained through software alone. For example, our FPGA implementation of the PMF algorithm can process in 1 s over 900 mass spectra consisting of 200 000 mass-abundance pairs. When implemented alongside a hardware implemented database search algorithm, which should deliver a match in <100 ms, the goal of real-time peptide mass fingerprinting seems eminently achievable. Another very exciting prospect is that FPGAs will enable the fast execution of 'intelligent' optimization protocols of instrument settings and spectrum processing, which take prohibitively long time to run even on a high-end workstation. For example, the closed-loop multi-objective optimization approach proposed recently by O'Hagan *et al.* (2005), which employs Genetic Algorithms and Genetic Programming to determine optimal instrument settings and remove noise, reportedly takes from 20 min and up to 118 h to run.

The motherboard can be configured to have up to three additional FPGA modules that can be plugged into dedicated motherboard slots. These modules will be used to implement the database search. Each FPGA module has one Virtex-II XC2V8000 FPGA device and 1GB of DDR SDRAM that can

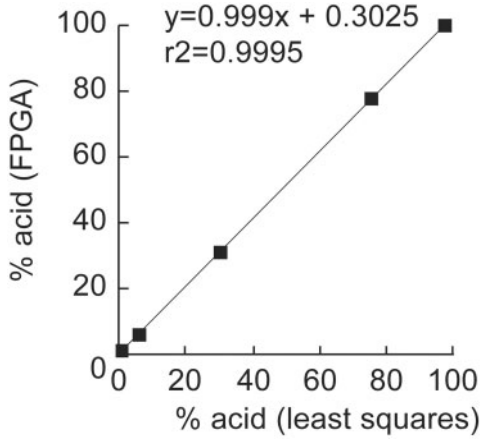

Fig. 12. Performance of the FPGA implementation in spectra deconvolution. The set of mass spectra described in Figure 7 were used as source data for the calculation of the proportion of acid and amide whether assessed by a least squares method or using the FPGA implementation.

Table 1. Benchmark results of the spectrum processor implementations

Spectrum size	Timing [ms]: Dual Xeon 3GHz processors	Timing [ms]: Virtex-II FPGA, 180MHz clock	Speed gain
25 488	20.27	0.1632	124.20
50 448	31.23	0.3105	100.56
75 168	47.33	0.4557	103.86
101 040	62.50	0.5607	111.46
125 184	79.17	0.7557	104.76
150 114	114.33	0.8547	133.76
175 104	130.20	1.0024	129.88
200 976	188.63	1.1219	168.13

easily hold the entire protein database. Each module is connected with the motherboard user FPGA implementing the spectrum processor and with other two modules via a 64 bit, 66 MHz local bus. This architecture will enable the implementation of parallel searches at FPGA level as well as across modules.

There are three types of proteomics: identification proteomics, characterization proteomics and quantitative proteomics. The core technology in many of these applications is mass spectrometry, but power of modern instrumentation brings with it the penalty of highly information-rich data streams at very high rates. As such, the bottleneck is moving from data acquisition to data processing. In identification proteomics and in quantification proteomics in particular, hardware solutions such as described here, could solve this bottleneck, and increase proteomics throughput considerably.

ACKNOWLEDGEMENTS

This work was supported by BBSRC. The authors gratefully acknowledge the support of Xilinx Inc. who donated the devices and design tools used in this study.

Conflict of Interest: none declared.

REFERENCES

Anish,T.A. *et al.* (2005) Hardware-accelerated protein identification for mass spectrometry. *Rapid Commun. Mass Spectr.*, **19**, 833–837.

Beynon,R.J. *et al.* (2005) Multiplexed absolute quantification in proteomics using artificial QCAT proteins of concatenated signature peptides rates. *Nat. Methods*, **2**, 587–589.

Breen,E.J. *et al.* (2000) Automatic Poisson peak harvesting for high throughput protein identification. *Electrophoresis*, **21**, 2243–2251.

Fagin,B. *et al.* (1993) A special-purpose processor for gene sequence analysis. *Comput. Appl. BioSci.*, **9**, 221–226.

Guccione, A.S. and Keller, E. (2002) Gene Matching Using Jbits, Proceedings of the Reconfigurable Computing Is Going Mainstream, 12th International Conference on Field-Programmable Logic and Applications, 1168–1171.

Guerdoux-Jamet,P. and Lavenier,D. (1997) SAMBA: hardware accelerator for biological sequence comparison,. *Comput. Appl. BioSci.*, **13**, 609–615.

Hughey,R. (1996) Parallel hardware for sequence comparison and alignment. *Comput. Appl. BioSci.*, **12**, 473–479.

Lavenier,D. (1998) Speeding up genome computations with systolic accelerator. *SIAM News*, **31**, 1–8.

Marongiu,A. *et al.* (2003) Designing hardware for protein sequence analysis. *Bioinformatics*, **19**, 1739–1740.

O'Hagan,S. *et al.* (2005) Closed-Loop, Multiobjective Optimization of Analytical Instrumentation: Gas Chromatography/Time-of-Flight Mass Spectrometry of the Metabolomes of Human Serum and Yeast Fermentation. *Analytical Chem.*, **77**, 290–303.

Oliver,T. *et al.* (2005) Using reconfigurable hardware to accelerate multiple sequence alignment with ClustaIW. *Bioinformatics*, **21**, 3431–3432.

Pratt,J.M. *et al.* (2006) Multiplexed absolute quantification for proteomics using concatenated signature peptides encoded by QconCAT genes. *Nat. Protocols*, **1**, 1029–1043.

Samuelsson,J. *et al.* (2004) Modular, scriptable and automated analysis tools for high-throughput peptide mass fingerprinting,. *Bioinformatics*, **20**, 3628–3635.

Savitzky,A. and Golay,M.J.E. (1964) Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chem.*, **36**, 1627–1639.

Simmler,H. *et al.* (2004) Real-Time Primer Design for DNA Chips. *Interscience Concurr. Comput.: Pract. Exper.*, **16**, 855–872.

Wozniak,A. (1997) Using video-oriented instructions to speed up sequence comparison. *Comput. Appl. BioSci.*, **13**, 145–150.

XILINX (2004) Distributed Arithmetic FIR Filter V9.0, DS240, Xilinx Inc.