LOW POWER HIGH PERFORMANCE COMPUTING FOR AUV BASED REAL-TIME SYNTHETIC APERTURE SONAR

Simon Banks*, Andrea Bellettini**, Simon Charles*, Allan Willcox*

*Bloomsbury DSP Ltd.
**NATO Undersea Research Centre

1 INTRODUCTION

Bloomsbury DSP Ltd. has developed and implemented a number of Synthetic Aperture Sonar (SAS) component algorithms aimed at real time operation in Autonomous Underwater Vehicles (AUVs). Algorithms have been implemented using high performance computing solutions utilising a combination of Field Programmable Gate Array (FPGA) and General Purpose Processor (GPP) technology. Such implementations maintain processing flexibility, whilst minimising power consumption and physical volume, making them suitable for use in battery powered AUVs.

Synthetic Aperture Sonar (SAS) provides very high-resolution sonar images suitable for use in a wide range of seabed survey applications including Mine Counter Measures (MCM) operations [4]. Combined with high resolution, the ability to produce multi-aspect images of a target means SAS has the capability to provide a step change in the performance of automatic detection and classification of underwater objects.

There is currently a growing interest in both the defense and commercial sectors to perform surveys remotely using small battery powered autonomous vehicles to reduce survey times, costs and risk to personnel. For MCM operations, the ability to perform automated detection and classification of underwater objects on-board an AUV can potentially lead to significantly reduced missions times, and a large reduction in the amount of data that must be off-loaded from the AUV to the surface ship. Real time on-board processing of SAS imaging and CAD/CAC is therefore highly desirable.

This paper discusses high performance computing architectures suitable for the implementation of real-time SAS algorithms. It is shown that by fusing GPP and FPGA technology, it is possible to build low power and compact re-configurable solutions for real time SAS processing.

SAS imaging involves many processing stages including pulse compression, beam-forming and micro-navigation. This paper concentrates on the implementation of the DPCA acoustic navigation algorithm using a combination of GPP and FPGA technology. A hybrid FPGA / GPP implementation of this algorithm has recently been delivered by Bloomsbury DSP Ltd. to the NATO Undersea Research Center (NURC), La Spezia, Italy.

2 DPCA MICRO-NAVIGATION ALGORITHM

The Displaced Phase Center Algorithm (DPCA), provides a very accurate estimate of vehicle motion using the acoustic returns from an array of receivers for SAS [2]. Not only is an accurate motion estimate essential for the formation of high quality SAS images, it also provides a very accurate estimate of platform motion that can potentially be fed into the navigation system of an AUV. The use of data from the DPCA therefore has the potential to reduce the requirement for very accurate motion sensors on AUVs.

The input to the algorithm is data from two consecutive sonar pings. A requirement of the algorithm is that at least some of the phase centers[1] from consecutive pings overlap. This is shown in figure 1.

---

[1] A location half way between the transmitter and receiver which approximates a monostatic transmit and receive setup.

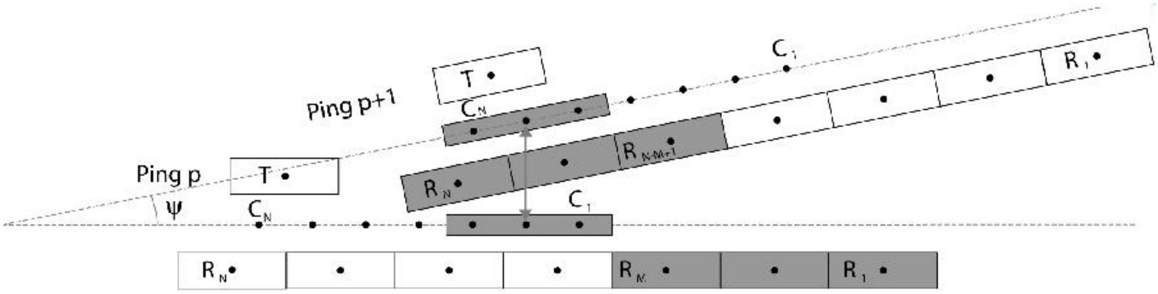

Figure 1 : Displaced phase centres from two consecutive pings

By cross-correlating these overlapping pairs of signals, a sub-array of correlation functions is formed. This sub-array of virtual signals can be beamformed in a direction determined by an external reference. The estimation of the DPCA sway is then possible by measuring the time delay associated with this beam. Iterating this method with varying sub-array sizes, a joint estimation of ping-to-ping sway and surge is possible [1].

The curvilinear integration of these ping-to-ping motions provides an estimate of the across-track deviations of the platform trajectory accurate to well within the wavelength of the transmitted pulse over an integration time sufficient to provide large SAS resolution gains [2].

The across-track deviations of the trajectory being referred with respect to a local slant-range plane, to cover the full swath of the sonar the algorithm above has to be iterated time-windowing the signals at different ranges.

2.1 Implementation of the algorithm

BDSP were asked by NURC to implement DPCA in real time using the combination of PowerPC and Xilinx Virtex II FPGAs provided on the Spectrum Signal Processing SDR-3000 system [3]. The processing was required to operate with data collected from a SAS sonar system mounted on an AUV.

The SDR 3000 system consists of a Pro-3500 GPP board and a Pro-3100 FPGA board. The GPP board provides two PowerPC 7410 processors, with Altivec extensions, operating at 500MHz. The FPGA board provides four Xilinx Virtex II 6 mega-gate FPGAs.

The SAS sonar system is double sided, each side utilizing a 36 element main receiver array with an additional shorter bathymetric array. The maximum expected ping-to-ping overlap is equivalent to 16 phase centers, corresponding to the minimum vehicle speed. The minimum expected overlap is one phase center, corresponding to the maximum vehicle speed. An on-board INS sensor provides a real time update of platform yaw.

Frequency domain cross correlation of data from overlapping phase centers provides an estimate of platform sway. An estimate of surge is obtained by beamforming the cross-correlated data in the yaw direction provided by the INS system, and performing interpolation to find the maximum.

The algorithm divides well into the following two sections :

1. **Systematic processing** including FFT, cross correlations, beam-forming and inverse FFT
2. **Non-systematic processing** including control of data loading and unloading, and interpolation

Systematic algorithms are well suited to FPGA implementation. Configured for a specific task, FPGAs can provide significantly higher processing power per watt, and reduced physical volume, compared to a GPP processor. For this reason, the FFT, cross correlation and beam-forming were implemented using a FPGA device.

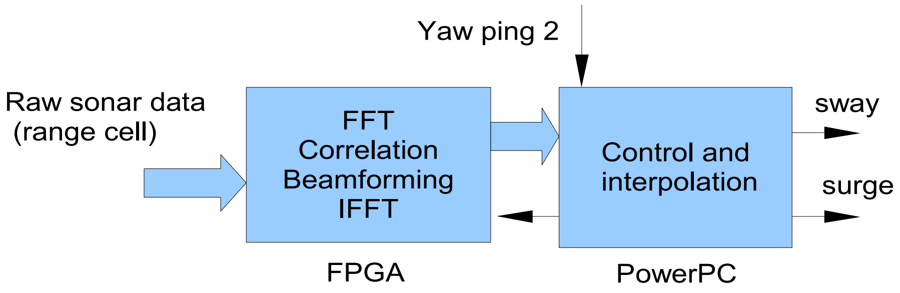

Figure 2 : Split between FPGA and PowerPC

GPP devices, such as the PowerPC provided on the Spectrum SDR 3000 system, are well suited to non-systematic algorithms such as interpolation. Coding using a language such as C means changes to the algorithm are relatively easy to make. For this reason, the interpolation and control of the DPCA process was implemented in a PowerPC processor.

The split between FPGA and PowerPC of the algorithm is shown in figure 2.

2.2 FPGA Implementation

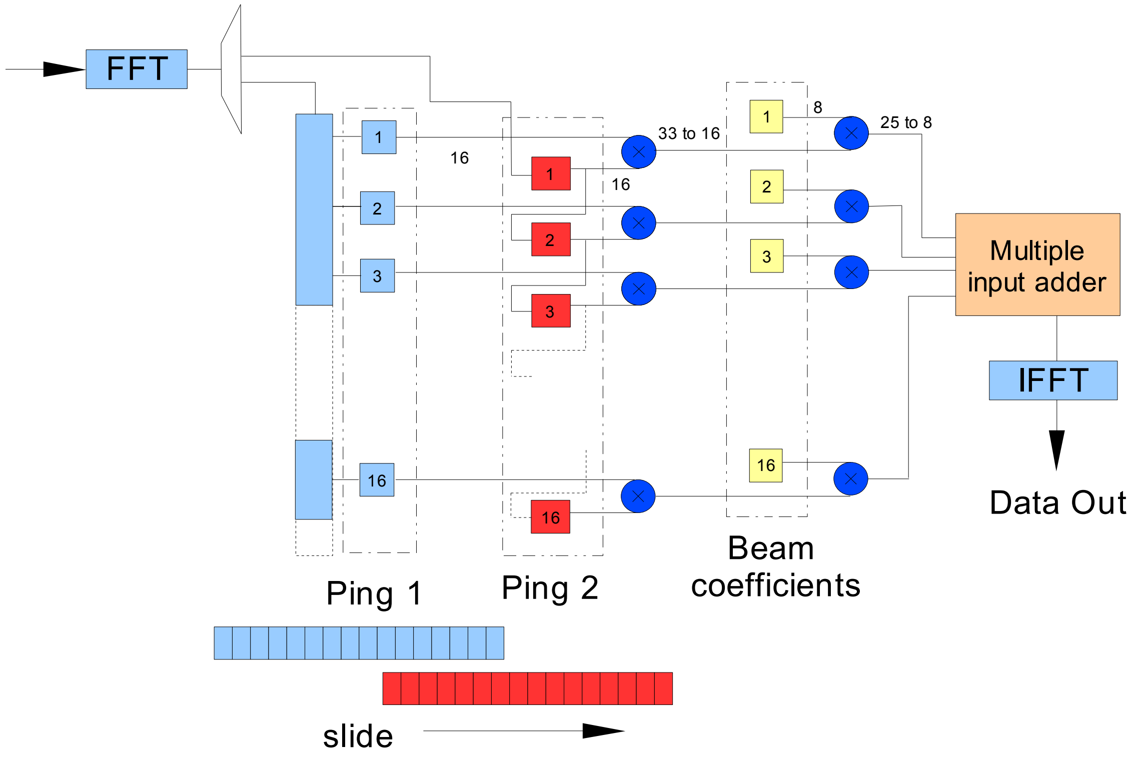

Figure 3 : Simplified architecture of FPGA design

The simplified architecture of the FPGA implementation of the FFT, correlation, and beam-forming, is shown in figure 3. The sonar data is complex fixed point, represented using 16 bits for real and 16 bits imaginary.

The Xilinx Virtex II device used allows for a maximum clock rate of 100 MHz with this design. Whilst it is estimated that a more modern device, such as a Virtex-4, would provide a doubling in clock speed, the processing rate obtained was more than adequate to process data from both sides of the sonar system. A peak processing rate of over 30 Gops/s was achieved. Due to latencies in the control mechanism the mean operations per second is somewhat less than this.

Data is processed using range cells of length 256 samples. This means the data stored at one time in the FPGA is relatively small, amounting to around 32 k bytes. Although a large external RAM of 256 MB is provided on the SDR 3000 system, storing data entirely within the FPGA means large memory bandwidths can be achieved by making use of the many dual-port block-RAM primitives provided in the Virtex II device. Forward and reverse FFTs are performed using a streaming architecture, meaning data can be clocked in and out of the FFT blocks at the full 100 MHz.

Data from the two consecutive pings is stored in two memory structures, formed using block-RAM primitives provided in the Xilinx Virtex II FPGA.

The *ping-1 memory* is shown in *blue* on the left in figure 3. The memory has one 32 bit data input port, and 16x32 bit data output ports. The maximum write data rate is therefore 3.2 Gbits/s and the read data rate is 51.2 Gbit/s. When full, each of the 16 sub memories stores stores one range cell for one channel of data from ping 1.

The *ping-2 memory* is shown in *red* on the right in figure 3. Like the ping-1 memory, the ping-2 memory also has one 32 bit data input and 16x32 bit data output ports. The difference is the ping-2 memory is arranged as a tapped shift register, meaning as data is written to the memory it flows from the top of the memory, location 1, eventually reaching the bottom, location 16. The shift register has 16 taps spaced at 256 32 bit word locations, allowing data from the same sample of each channel to be read simultaneously.

Following the memories are two sets of complex multipliers. The first set multiplies, in the frequency domain, data from ping 1 with data from ping 2 to form a correlation matrix. The second set of multipliers apply a frequency dependant phase factor, stored as 8 bit fixed point complex, to steer the correlation matrix to the direction measured by the on-board inertial navigation system.

Following the beam-steering is a multiple input complex adder tree which performs beam-forming at the angle provided by the INS system. A tree structure is required to enable data from all 16 channels to be summed at once.

Finally, an inverse FFT is performed on the data using a streaming IFFT structure.

Order of operation

Once data is fully loaded into the ping-1 memory, via the forward FFT, each of the 16 sub-memories contains one range cell of data for one of the 16 sonar channels. Data can then be loaded into the ping-2 memory, again via the forward FFT. The correlation process can begin once the *first channel* of ping-2 is loaded into the ping-2 memory. As the ping-2 memory fills up, correlations for each of the 16 possible surge positions are performed. The movement of data down the shift register structure of the ping-2 memory can be compared to the sliding of one ping over the consecutive ping. Once data is loaded, the design can run without additional control input, producing 16 correlated beams corresponding to all 16 possible overlaps of the two ping positions.

The ping to ping surge and sway is calculated using interpolation; this process is carried out in the PowerPC 7410 processor.

2.3 7410 PowerPC implementation

The final stage of the process is an interpolation of the data output by the structure shown in figure 3. This interpolation stage has been implemented using one of the 500 Mhz PowerPC 7410 processor provided on the Spectrum 3500 card. The algorithm has been coded in C. In this case, the PowerPC processor runs the real-time operating system VxWorks[6].

In the current implementation, correlation and interpolation phases are performed sequentially. It is possible to modify the design to operate in a pipelined fashion, with data from pings N and N+1 being processing in the Power PC, whilst data from pings N+1 and N+2 are processed in the FPGA. This pipelined mode of operation reduces idle time and hence increases overall performance.

3 PERFORMANCE: GPU VS FPGA

The system was designed to meet a specified operational speed of less than 6.8ms per 512 sample range cell of sonar data. The section of DPCA implemented in the FPGA, shown in Figure 2, easily meets this specification, taking only 1.1ms per range cell. Using sequential operation, where correlations and interpolation are not performed in parallel, the processing time is 6.1ms, falling within the 6.8ms specification. The real-time operating system VxWorks and predictable performance of the FPGA implementation ensure that the processing time of 6.1ms is consistent for each range cell.

We estimate that the 7410 PowerPC processor would require 10ms to complete the process, around 9 times longer than the FPGA. Although the algorithm was not tested on the PowerPC, we were able to derive this estimate using performance figures for a similar implementation on an Intel platform using IPP[5], and scaling by the clock frequency.

Both the FPGA device and the PowerPC processor used are a number of years old. However, the ratio of achievable performance between FPGA and GPP remains similar when comparing modern devices, say an Intel Core 2 Duo based GPP with a Xilinx Virtex-4 (as used on the Spectrum Signal Processing SDR-4000 platform) or Virtex-5 FPGA. In terms of power consumption, a Virtex-5 device may consume around 3-5 watts, whereas a low power single board computer may consume between 10 and 40 watts.

4 OTHER SAS ALGORITHMS

This paper has concentrated on the implementation of the DPCA using a hybrid FPGA and PowerPC high performance computing architecture. To produce a complete SAS image, a number of additional processing stages are required. These include pulse compression, physical aperture beamforming and SAS stacking. These algorithms use very roughly 10%, 40% and 50% respectively of the total processing time. Bloomsbury DSP has also implemented these algorithms using hybrid FPGA / GPP architectures, leading to a complete set of high performance, low power solutions for real time SAS processing.

The DPCA has been implemented on the Spectrum Signal Processing SDR-3000 system. However, other high performance computing platforms are available incorporating a combination of FPGA and GPP technology. Bloomsbury DSP has also implemented other real-time SAS algorithms using Xilinx Virtex II Pro and Virtex-4 FPGAs, as well as x86 CPUs including the use of the Intel Integrated Performance Primitives (IPP) [5].

5 CONCLUSIONS

In this paper we have discussed the benefits of using real time SAS with small AUVs for applications including MCM operations. To facilitate real time SAS processing, a compact and low power computing platform is required. We have shown that for the DPCA algorithm, such a platform can be made using a combination of a PowerPC and a Xilinx Virtex II FPGA. More generally, the combination of FPGA and GPP technology leads to flexible, compact and low power computing platforms suitable for the implementation of real time SAS for AUVs.

6 ACKNOWLEDGEMENTS

Ben Evans, Reginald Hollett, Peter Munk and Marc Pinto from the NATO Undersea Research Centre, La Spezia, Italy, are gratefully acknowledged for initiating the work to develop a real-time implementation of the DPCA micro-navigation algorithm.

7 REFERENCES

[1] French patent 9700487 Procédé de recalage d'images sonar au moyen de sous-antennes, Thomson Marconi Sonar (Pinto, M.), 17 January 1997 .

[2] A. Bellettini and M. A. Pinto. 'Theroretical accuracy of synthetic aperture sonar micronavigation using displaced phase centre antenna'. IEEE Journal of Oceanic Engineering, Vol. 27, Issue 4 Oct 2002 pp780-789.

[3] Spectrum Signal Processing, Inc. See: http://www.spectrumsignal.com/ *

[4] B. Fletcher, 'UUV master plan: a vision for navy UUV development', Oceans 2000, Volume 1, 11-14 Sept 2000, pp65-71.

[5] Intel Integrated Performance Primitives. See : http://www.intel.com/cd/software/products/asmo-na/eng/perflib/ipp/index.htm *

[6] VxWorks real time operating system from Wind River. See: http://www.windriver.com/ *

*web addresses correct at time of writing.