Proceedings of the Brazilian Decimetric Array Workshop
São José dos Campos, Brazil - July 28 – August 1, 2008

ALTERNATIVE BDA CORRELATION SYSTEMS: FPGA VERSUS SOFTWARE CORRELATION

Gustavo Poli, José H. Saito

Departamento de Computação - Universidade Federal de São Carlos
Rodovia Washington Luis, Km 235 – São Carlos-SP
(saito@dc.ufscar.br, gustavo_silva@dc.ufscar.br)

ABSTRACT

This work is concerned to an analysis of the use of alternative BDA Correlation Systems implementations. The first one is the FPGA (Field Programmable Gate Array) system, using commercial platforms of reconfigurable chips, in this case a Virtex-II Pro (XC2VP30) of Xilinx Company. The second alternative is the software correlation. In this case, a software development platform, explicitly the GPU (Graphic Processing Unit) processor, with multiple cores driven by very high memory bandwidth, is analyzed. The results showed the viability of this type of device as a massively parallel data processing tool, and that smaller the granularity of the parallel processing, and the independence of the processing, better is its performance.

INTRODUCTION

In the BDA correlations, all the interferometer pairs formed by the 39 antennas are measured using one-bit correlators which can be assembled with digital logics, that yields greater stability than analog correlators. In order to obtain all possible multiplications between the 39 antennas, n(n-1)/2 = 741 complex correlations are calculated. At the implementation of the BDA correlators in the phase I, it was used the chips designed for the Nobeyama radioheliograph, Japan (Ramesh, 1998). There, each chip is composed by 4 complex correlator units, which corresponds to 2x2 combinations of antennas, and each unit consists of there 4-bit- parallel Ex-OR circuits, one 4-bit- parallel Ex-NOR circuit, two addition circuits, two integration circuits, a latch, and a multiplexer circuit. It can be noted that in the 22-bit counter used for integration, only the 16 MSBs (most significant bits) are read out with the 6 LSBs (least significant bits) being truncated inside the chip.

The complex correlation of one interferometer pair is obtained as a cosine correlation and sine correlation. Considering that an antenna *i* provides cosine signal, Ci, and sine signal, Si, the output of the cosine correlator is C1(+)C2 + S1(+)S2, and the output of the sine correlator is C1(+)S2+S1(.)C2 where (+) represents the Ex-OR operation and (.) represents the Ex-NOR operation, when we consider the antennas 1 and 2.

In the BDA case, the correlator system may be comprised by 8 boards, each one with 16 correlator chips. At the end of each integration period (about 145 ms), the correlated data from each of these 128 correlators in one board is written into a memory unit. This operation goes on synchronously in all 8 boards. The process of reading correlated data from each chip and writing into the memory unit goes on until we reach 256 integration cycles. At the same time, the data which was written into

another memory unit during earlier 256 integration cycles will be read into a computer. At the end of 256 integration cycles, the operations of the memory units are exchanged.

In the other side, reconfigurable electronic components have been introduced since 1980's, becoming more and more attractive to many applications. A FPGA is a semiconductor device containing programmable logic components called logic blocks, and can be programmed to perform the function of basic logic gates such as AND, and XOR, or more complex combinational functions such as decoders or mathematical functions. In most FPGA's, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory. The inherent parallelism of the logic resources on a FPGA allows for considerable compute throughput even at low clock rate. The current generation of FPGAs can implement around a hundred single precision floating point units, all of which can compute a result in a single clock cycle. The flexibility of the FPGA allows for even higher performance by trading off precision and range in the number format for an increased number of parallel arithmetic units. The adoption of FPGAs in high performance computing is currently limited by the complexity of FPGA's design compared to the conventional software and the extremely long turn-around times of current design tools, where some hours wait is necessary after even minor changes to the source code.

The first part of this work is concerned to an analysis of the use of alternative BDA Correlation Systems implementations, using commercial platforms of reconfigurable chips, in this case a Virtex-II Pro (XC2VP30) of Xilinx Company. with 13969 Slices, 80 × 46 Array Size, 136 Multiplier Blocks, 2448 Kb RAM per block and 2 PowerPC RISC Cores. For the development it is used the ISE 9.2 environment and VHDL as the programming language of the correlation circuit. As it is expected the results are similar to the first implementation for the BDA correlation system, using Nobeyama radioheliograph correlation chip.

The second part is concerned to the software correlation. In this case, a software development platform, explicitly the GPU (Graphic Processing Unit) processor, with multiple cores driven by very high memory bandwidth, is analyzed.

The following sections are concerned to the description of these two alternative implementations, and the conclusions.

FPGA CORRELATION DEVELOPMENT

In this section it is described the FPGA application development system used to the BDA correlation development. The XUP (*Xilinx University Program*) Virtex-II Pro (XILINX, 2006) platform provides advanced resources to the development of reconfigurable applications, which consists of high performance FPGA, XC2VP30, with a high quantity of components that can be used to create complex systems. It follows the main elements of the platform: Virtex-II Pro FPGA with two PowerPCs 405 cores; 512Mb de (DDR) SDRAM; USB connections and interface to FPGA reconfiguration and data store; 10/100 Ethernet network interface; RS-232 DB9 serial port; 2 serial ports PS-2; 4 LEDs conected to Virtex II I/O pins; 4 *switches* connected to Virtex II I/O pins; 5 buttons connected to Virtex II I/O pins; AC-97 audio *CODEC*; XSGA video output with 1200 x 1600 resolution at 70 Hz; and 3 ATA serial port, with 100 MHz clock system, and 75 MHz SATA clock.

The Virtex-II Pro platform provides several configuration methods. The configuration data can be originated from the platform Flash PROM, from a compact-flash memory card, or from a external configuration by a USB or parallel port.

The applications development can be realized using the WEBPack ISE 9.2, the free tool environment from Xilinx, which consists of a set of tools, as Xilinx Plataform Studio (XPS). In this environment the programs are described using ANSI C language and specific libraries to access the

protocol IEEE 802.3 to send data from host computer to the FPGA. All environment is realized on Linux 64-bits operation system (Ubuntu 7.10).

SOFTWARE CORRELATION DEVELOPMENT

Recently, we are attending to a huge evolution on the development of high performance computing platforms. Among these platforms, the GPU (Graphics Processing Units) has evolved into an absolute computing workhorse, with multiple cores driven by very high memory bandwidth. Today's GPUs offer incredible resources for both graphics and non-graphics processing. One of the main reasons is the fact that GPU is specialized to compute-intensive, highly parallel computation – exactly what graphics rendering is about – and therefore is designed such that more transistors are devoted to data processing rather than data catching and flow control. The internal pipelined processing fashion makes the GPU's suitable for stream processing. Furthermore, GPU's speed grows much faster than the famous Moore's law for CPU's, that is, 2.4 times/year versus 2 times per 18 months. Moreover, a GPU usually contains multiple (typically 4 or 8) parallel pipelines and is indeed a SIMD processor. Another important feature of most contemporary GPUs is their programmability of the partial or full graphics pipeline, thanks to the introduction of vertex shaders and pixel shaders in DirextX-8. The powerfulness of SIMD operation and programmability of GPU's have motivated an active research area of using GPU for non-graphics oriented operations such as numerical computations like basic linear algebra subprograms (BLAS) (Harris, 2003) and image/volume processing (Larsen and McAllister, 2001; Rumpf and Strzodka, 2001; Thompson, Hahn, and Oskin, 2002; Colantoni, Boukala, and Rugna, 2003; Moreland and Angel, 2003; Hopf and Ertl, 2000). In Larsen and McAllister (2001), the authors presented a technique for multiplying large matrices quickly using the graphics hardware of a PC. Strzodka and Rumpf have implemented numerical schemes solving parabolic differential equations fully in graphics hardware [5]. Chris Thompson et al. introduced a programming framework of using modern graphics architectures for general purpose computing using GPU (Thompson, Hahn, and Oskin, 2002). In Colantoni, Boukala, and Rugna, 2003), the authors tested five color image processing algorithms using the latest programmability feature available in DirectX-9 compatible GPUs. Performing FFT on GPU was reported by Doggett et al. in Moreland and Angel (2003). Wavelet decomposition and reconstruction was implemented on modern OpenGL capable graphics hardware (Hopf and Ertl, 2000).

For the development of this non-graphic application on GPU it was used CUDA (Compute Unified Device Architecture) as API (Application Programming Interface) between the CPU and the GPU, from GeForce 8800 GTX from NVIDIA, with 128 ALU's. CUDA is a new hardware and software environment for issuing and managing computations on the GPU as a data-parallel computing device without the need of mapping a graphics API. The operating system's multitasking mechanism is responsible for managing the access to the GPU by several CUDA and graphics applications running concurrently.

The objective of this work is to use GPU as an alternative correlation system, and compare the result with the FPGA correlation system. For this, it was developed the application using NVIDIA GeForce 8800 GTX, with 128 processing unit cores.

The CUDA Toolkit target a class of applications whose control part runs as a process on a general purpose computer, and which use an NVIDA GPU as coprocessor for accelerating SIMD parallel jobs. Such jobs are 'self-contained', in the sense that they can be executed and computed by a batch of GPU threads entirely without intervention by the 'host' (CPU) process, thereby gaining optimal benefit from parallel graphics hardware.

Dispatching GPU jobs by the host process is supported by the CUDA Toolkit in the form of remote procedure calling. The GPU code is implemented as a collection of functions in a language that is essentially 'C', but with some annotations of distinguishing them from host code, plus annotations for distinguishing different types of data memory that exists on the GPU. Such functions may have parameters, and they can be 'called' using a syntax that is very similar to regular C function calling, but slightly extended for being able to specify the matrix of GPU threads that must execute the 'called' function. During its life time, the host process may dispatch many parallel GPU tasks.

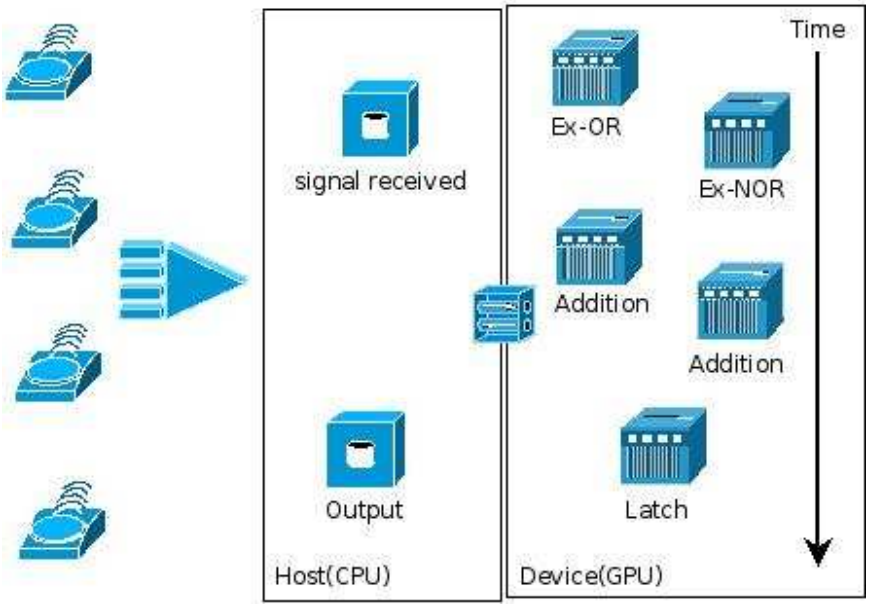

Fig. 1 - GPU/CUDA Software correlation diagram.

This work implements the software correlation to the BDA signals as shown in Figure 1. As the signals are received by the host CPU, it is buffered, until 10 sequential samples are achieved, corresponding to 29.640 antenna signals of 741 interferometers of BDA. When the buffered data is been processed at the GPU, the CPU must be receiving the subsequent signals from the antennas. The GPU processing uses two buffers Ex-OR and Ex-NOR, to compute these correlations in parallel, followed by the latching process to the Output buffer, at the CPU, to complete the process. The processing time in the GPU was 1.6 ms for 29,640 elements (741 x 4 in 10 times), but it can be increased using new versions of GPU.

CONCLUSIONS

It was implemented a subset of the correlation system using the Virtex II Pro platform of FPGA, and preliminary results showed its performance to the application, close to the Nobeyama radioheliograph correlator chip. The results of the use of GPU, showed the feasibility of the software correlation with this processing device, as a massively parallel data processing tool, and that smaller the granularity of the parallel processing, and the independence of the processing, better is its performance.

REFERENCES

Ramesh, R., PhD. Thesis, Bangalore University, 1998.

XILINX. *Xilinx University Program Virtex-II Pro Development System - Hardware Reference Manual*. [S.l.], Março 2006.

Harris, M., http://www.cs.unc.edu/~harrism/gpgpu, 2003.

Larsen E. S., McAllister, D. K. *Proc. IEEE Supercomputing*, 55, 2001.

Rumpf, M., Strzodka, R., *Proc. ICIP*, 3, 1103–1106, 2001.

Thompson, C. J., Hahn, S., Oskin, M., *Proc. ACM/IEEE MICRO*-35, 306–317, 2002.

Colantoni, P., Boukala, N., Rugna, J. D., *8th Int. Fall Workshop: Vision Modeling and Visualization*, Munich, Germany, 2003.

Moreland K., Angel, E., *Proc. SIGGRAPH/Eurographics Workshop Graphics Hardware*, 112–119, 2003.

Hopf, M., Ertl, T., *Proc. EG/IEEE TCVG Symp. Visualization*, 93–103, 2000.