Paper: ASAT-17-018-RS

17th International Conference on
AEROSPACE SCIENCES & AVIATION TECHNOLOGY,
ASAT - 17 – April 11 - 13, 2017, E-Mail: asat@mtc.edu.eg
Military Technical College, Kobry Elkobbah, Cairo, Egypt
Tel: +(202) 24025292 – 24036138, Fax: +(202) 22621908

An Approach of Using General Purpose Graphics Processing Units in Modern Systems for Electronic Warfare

{N. Kozic, I.P. Pokrajac* and P. Okiljevic}†

Abstract: Modern systems for electronic warfare, especially systems for electronic support based on fast scanning wideband direction finders, are usually designed to achieve high scanning speed in order to provide high probability of detection of signals of interest. To achieve high scanning speed two parameters are important: the first is instantaneous bandwidth of a receiver and the second is processing speed. By increasing instantaneous bandwidth of the receiver, it is possible to increase scanning speed. However, increasing of instantaneous bandwidth has a direct impact at amount of processing power to keep the same or to increase processing speed. In modern systems for electronic support, scanning speed depends on instantaneous bandwidth of the receiver, frequency resolution, implemented methods for estimation of direction of arrival and required preprocessing and processing power of hardware. The latest computer system architectures being developed by the computer industry offers more processing power in a smaller footprint. Parallel computing has been rejuvenated with the development of multicore technologies such as Multicore Processors, Multicore Digital Signal Processors and General Purpose Graphics Processing Units. In this paper an approach of using GPGPUs in the wideband fast scanning direction finder has been considered. Some of the obtained results are also presented.

Keywords: antenna array, correlative interferometer, graphics processing unit, wideband direction finding.

1. Introduction

Estimation of direction of arrival (DOA) has various applications in both civil and defence oriented fields. In the defence application, the estimation of DOA is very important in Electronic Warfare (EW) systems and systems for gathering intelligence data such as the Communication Intelligence (COMINT) systems [1]. Modern EW and COMINT systems are based on fast-scanning wideband direction finders (WDF) that provide simultaneously estimation of DOA in the wider instantaneous frequency bandwidth. The basic requirement that has been set to modern WDF is the capability to achieve high probability of detection of all signals of interest (SOI), especially of emission with low probability of interception (LPI). This requirement could be fulfilled by increasing instantaneous frequency bandwidth and/or by decreasing processing time (increasing scanning speed of WDF). These two parameters are mutually connected because increasing the instantaneous bandwidth results in increasing amount of data that should be processed or in the other words it is necessary to increase processing power of processor. In order to provide high probability of detection, modern WDFs usually uses correlative interferometer (CI) method for DOA estimation.

* ivan.pokrajac@vs.rs
† Department for Electronic Systems, Military Technical Institute, MOD R. SERBIA.

This algorithm is serial based, mathematically intensive, and requires significant processing power to be realized in real-time. Comparing correlative interferometer method with some high-resolution method such as Multiple Signal Classification (MUSIC) method, it is obviously that correlative interferometer method has some limitations, however using this method it is possible to achieve high scanning speed in modern DF systems if is used appropriate processor.

In order to fulfill requirement for high processing power modern WDFs could use multicore technologies such as Multicore Processors, Multicore Digital Signal Processors (DSPs) and General Purpose Graphics Processing Units (GPGPU) that operate concurrently and form a WDF processor. Achieved processing power on one WDF processor formed of more multi-core GPUs could be enough for some application. Graphics processing units have been intensively used in general purpose computations for several years. In the last decade, GPU architecture and organization changed dramatically to support ever-increasing demand for computing power [2]. Contemporary GPUs are many core processors, offering abundant data-level parallelism with hundreds or thousands of cores available for execution. Although they can achieve very high number of Floating Points Operations Per Second (FLOPS), they are suitable only for certain set of problems that show high regularity. Therefore, GPUs are used as accelerators to central processing units (CPUs). In heterogeneous systems, CPU is in charge of I/O, management tasks, or smaller portions of work, while compute-intensive parts are offloaded to the GPU.

In this paper, the multi-core NVIDIA Tesla K20X GPU is chosen as the basic processing element, for its low power consumption and high-performance fixed/floating point calculations [3]. NVIDIA Tesla K-series accelerators are based on the NVIDIA Kepler compute architecture and powered by Compute Unified Device Architecture (CUDA), the world's most pervasive parallel computing model. Specifically, Tesla K20X is designed to be performance leader in double precision applications and the broader supercomputing market. The GPU is especially well-suited to address problems that can be expressed as data-parallel computations. The same program is executed on many data elements in parallel with high arithmetic intensity. Data-parallel processing maps data elements to parallel processing threads. Because the same program is executed for each data element, there is a lower requirement for sophisticated flow control, and because it is executed on many data elements and has high arithmetic intensity, the memory access latency can be hidden with calculations instead of big data caches.

2. Correlative Interferometer Method for DOA Estimation

DOA estimation in wideband direction finders is based on non-coherent wideband processing. The non-coherent approach processes each frequency bin in the instantaneous frequency bandwidth of WDF independently. Signal acquired in the instantaneous frequency bandwidth is decomposed at defined number of frequency bins and each frequency bin is approximated as a narrowband signal. In this case any narrowband DOA estimation method is applicable [4]. Estimation of DOA at each frequency bin is separately processed and the final result is average over J frequency bin those bins that corresponding to the same signal. The number of frequencies bins H depends from the instantaneous frequency bandwidth and frequency resolution. Increasing the number of frequencies bins H directly increase the amount of data for processing in WDF processor. In each frame there is H frequency bins for DOA estimation, under assumption that DOA estimation is performed without any detector and without any averaging. The number of frames (frame rate) that should be processed in WDF processor for defined time also depends from requested scanning speed of WDF. For example for instantaneous frequency bandwidth of 20 MHz and scanning speed of 10 GH/s at least 500 frames have to be processed in 1 s in the WDF processor.

In spite of the fact that different approaches in parallelization could be applied on algorithm, there is an eye-catching fact that processing any frequency bin in the frame does not influence processing of other bins. DOA could be determined just for that frequency bin without need to know anything about another one. That is the great potential for parallelization of algorithm execution and implementation at GPU. With more partitioning of processed spectrum and allocating hardware to partition, faster processing should be achieved.

Phase interferometry method for DOA estimation is based on measuring the phase differences among the responses of the antenna array elements to the impinging signal. This method can be realized based on measuring phase difference between the responses of the antenna array elements or by correlation based method [5]. The basic principle of the correlative interferometer method consists in comparing the measured phase differences with the phase differences of the reference spatial signal (RSS) obtained for the known configuration of DF antenna system at known DOA and frequency. The comparison is performed by calculating the quadratic error or the correlation coefficient of the two data sets. If the comparisons made for different values of DOA of the reference data set, the DOA is obtained from the data for which the correlation coefficient is at a maximum.

The correlative interferometry DF method can be realized in two steps [5]. The first step is used for forming the correlation coefficients of the two data sets. The second step is used for precise DOA estimation by parabolic approximation. In the first step correlation coefficients are estimated forming correlation matrix:

$$\mathbf{R} = \frac{\mathbf{X}_M \cdot \mathbf{X}_R^*}{\left[E\left\{ \left| \mathbf{X}_M \right|^2 \right\} \cdot E\left\{ \left| \mathbf{X}_R^* \right|^2 \right\} \right]^{1/2}} \tag{1}$$

where $\mathbf{X}_M$ is measured phase difference vector and $\mathbf{X}_R$ is calculated phase difference for known frequency and DOA or already calculated and stored phase difference from the lookup table.

Although knowing correlation coefficients of **R** continuums makes it easy to determine global maximum, there's still the problem of significant usage of memory space and processing time. Instead of that, it is more efficient to set correlation coefficients in discreet samples and then to locate maximums of correlation coefficients of these discreet data by using interpolation or curve fitting. Algorithm that is used for curve fitting of correlation coefficients implies that the curve can be approximated with parabola near its maximum value. In the case of correlation coefficients parabolic approximation of elevation, additional transformation of its coordinates is needed, because the curve is not symmetric near its maximum value. Formulas for precise estimation of azimuth A_{ZK} and elevation E_{LK} of arriving signal are presented in following:

$$\begin{aligned} A_{ZK} &= A_{ZP} + \Delta\theta * A_{ZQ} \\ A_{ZQ} &= \frac{1}{2}\left(\frac{\varepsilon}{\delta}\right), \quad \varepsilon = \frac{1}{2}\left(R_R - R_L\right); \delta = R_{TR} - \frac{1}{2}\left(R_R + R_L\right) \end{aligned} \tag{2}$$

$$E_{LK} = \cos^{-1}\left(A + \frac{B * Q_E}{C * Q_E + 1} \right), \quad Q_E = \frac{1}{2}\left(\frac{\varepsilon}{\delta}\right) \tag{3}$$

where A_{ZK} and E_{LK} are estimated values of azimuth and elevation; A_{ZP} estimated azimuth in first step; $\Delta\theta$-azimuth resolution; R_{TR} maximum value of correlation coefficients; R_R and R_L-adjacent maximum correlation coefficients; A, B, C – constants for different values of estimated elevation in first step.

3. General Purpose Graphic Processor Tesla K20X

GPU has a multicore architecture consisting of thousands of cores. When parallelizing any algorithm, it should be decomposed into various parts, and each part is broken down into tasks. Each task is executed on a different core of the GPU, and each task contains threads which are executed simultaneously.

Implementation of the algorithm for DOA estimation based on the correlative interferometer method is done on NVIDIA Tesla K20X. The NVIDIA Tesla K20X GPU accelerator is a PCI Express, dual-slot computing module comprising of a single GK110 GPU. The Tesla K20X is designed for servers and offers a total of 6 GB of GDDR5 on-board memory and supports PCI Express Gen2. It is a device with 2688 CUDA cores with 732 MHz for the core clock. In Fig. 1 is shown block diagram for the Tesla K20X GPU dual-slot computing processor module [3].

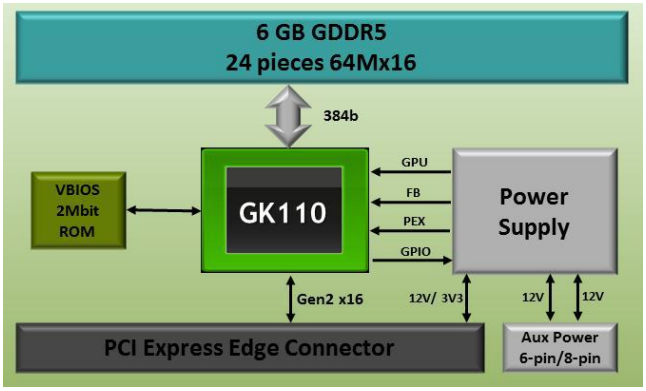

Fig. 1. Tesla K20X Block Diagram

Tesla K20X GPU accelerator is a performance optimized, high-end product and uses power from the PCI Express connector as well as external power connectors. Board power is 235 W and idle power is 25 W. Tesla K20X is designed to be the performance leader in double precision applications and the broader supercomputing market. It provides a peak performance of 3.95 TFLOP for a single precision floating point and 1.17 TFLOP for a double precision floating point. For example, Tesla K20X manages to score 1.22 TFLOPs in double precision general matrix multiply (DGEMM), which puts it at almost three times faster than the previous generation Tesla M2090 accelerator. In figure 1234 are shown graphics for comparison in precision performances.

NVIDIA Tesla K-series Accelerators are based on the NVIDIA Kepler compute architecture and powered by CUDA. CUDA is a general purpose parallel computing platform and programming model that leverages the parallel compute engine in NVIDIA Tesla K-series GPUs to solve many complex computational problems in a more efficient way than on a CPU. CUDA comes with a software environment that allows developers to use C as a high-level programming language. The massively parallel hardware architecture and high performance of floating point arithmetic and memory operations on GPUs make them particularly well-suited to many of the scientific and engineering workloads.

4. Implementation of Correlative Interferometer Method for DOA Estimation at GPU Tesla K20X

GPUs are mainly programmed through extensions of commodity programming languages such as C, C++, and Fortran. Low-level API-s, such as CUDA and OpenCL offer significant control over program execution and performance optimization, but they pose a problem for non-experts in the field of computer science. On the other hand, high-level, directive-based APIs exist, such as OpenACC, in which compiler is responsible for code generation. In both cases, some knowledge of GPU architecture is desirable, since performance optimization is a challenging task on the GPU. Performance can vary greatly depending on the resource constraints of the particular device architecture, and it is much on the developer to exploit all parallelism available [6].

A typical GPU consists of several streaming multiprocessors (SMs) that are able to execute a large number of lightweight threads. Each SM consists of a number of scalar processors. Typically, CUDA programs are executed in co-processing fashion. Sequential parts are executed on the host CPU, while compute-intensive parts are offloaded to the GPU for parallel execution, as special function called kernels. Kernel execution is organized as a grid of thread blocks. Threads are executed in a SIMD fashion. Threads in a thread block can be synchronized using a barrier, but global synchronization is achieved only when kernel execution terminates. Threads from different thread blocks cannot cooperate, since they may or may not execute on the same SM. Available resources (registers, shared memory) are shared by all thread blocks executed on particular SM.

GPU memory architecture is designed to support high throughput and execution of a number of threads in parallel. It consists of a hierarchy of memories that differ in speed and capacity. Global DRAM memory accesses are slow, so threads can utilize other smaller memories to speed up the execution, such as registers, shared memory, constant memory, etc. Each thread has an exclusive access to allocated registers and local memory. Threads in a block could share data through, user-managed, per-block shared memory. Typically, CPU and GPU use different, physically separated memory spaces, so explicit transfers of data between CPU and GPU are needed. GPU based the correlative interferometer method for DOA estimation in the wideband direction finder is implemented as a dynamic linking library (DLL) for National Instruments LabVIEW software package. The correlative interferometer method is performed through library calls that are provided with all necessary data: frequency, antenna steering array, measured phase difference, etc. Estimated DOAs are stored in memory as 1D, $1 \times H$ vector.

The correlative interferometer method is implemented in few steps:

- Initialization step is used to allocate CUDA objects, initialize the data structure used in other library calls, and transfer the data to the GPU.
- Starting kernel function *kernel_DOA_estimation<<<dim3(8), dim3(1024)>>>* that provide DOA estimation at each of H frequency bin, $H=8192$ in the instantaneous frequency bandwidth of 21 MHz.
- Based on known antenna array and selected central frequency of the instantaneous bandwidth, steering vectors for predefined set of DOAs are calculated for each frequency bin. In our case there 756 possible DOAs in the defined set .
  The number of threads is 756 that corresponding to the defined set of possible DOAs. In each threads correlation coefficients are estimated forming correlation matrix. Measured phase difference and calculated steering vectors for predefined set of DOAs are used for estimation of the correlation coefficients. Correlation coefficients are matrix dimension, $\mathbf{R} \in C^{36\times21}$.

- These estimated matrices with correlation coefficients are used in the next step for estimation of DOAs at each frequency bin by parabolic approximation.
- Finalization phase is used for cleanup purposes and transferring of results to the host. In all other steps, several CUDA kernels have been implemented.

In the simulations we use two different software packages. One is Mathworks MATLAB® software package when CPU is used as DF processor and the other is LABVIEW software package, which is used for graphical user interface, when GPU is used as DF processor. Aim was to compare results obtained with the same simulation parameters on two different processors. While implementation of the algorithm for DOA estimation when GPU is used as DF processor, is done on NVIDIA Tesla K20X, implementation of the algorithm for DOA estimation when CPU is used as DF processor, is done on Intel(R) Core(TM) i7-6500U CPU @ 2.5 GHz.

In the first simulation is investigated estimation of Root Mean Square Error (RMSE) of DOA estimation depending on Signal to Noise Ratio (SNR). SNR is changed in range from 0 to 25 dB with step of 5 dB. Fixed frequency signal is generated with MATLAB based simulator for signal generation at antenna array. In the first part of simulator is generated scenario at selected five-element circular antenna array in the central frequency of 390 MHz and instantenous bandwidth of 21 MHz. Next step is adding Additive White Gaussian Noise (AWGN) on signal with specific SNR value. The statistical analysis for the DOA estimation has been estimated by 50 independent Monte-Carlo trials. Obtained signal is processed using algorithm for DOA estimation and estimations of RMSE are generated. In Fig. 2 are shown obtained results. The curves of RMSE of DOA estimation using correlative interferometer method, implemented on the i7-6500U CPU and implemented in on NVIDIA Tesla K20X have almost perfect matches in all cases. It proves that practical implementation of the correlative interferometer method on the GPU NVIDIA Tesla K20X is qualitatively done.

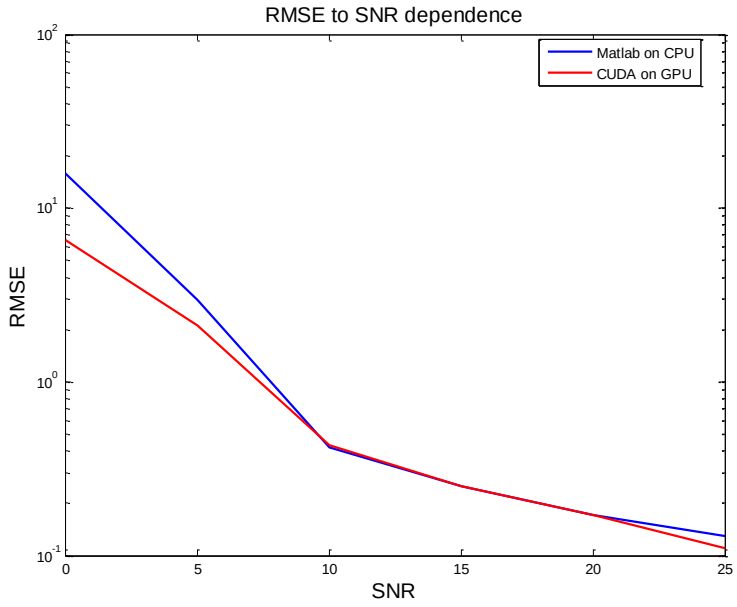

Fig.2 RMSE to SNR dependence

In the next scenario, using the same simulations, time for signal processing is measured. Time is measured when number of processing frames is changed in range from 10 to 100, while SNR is fixed at 15 dB. In Fig. 3 is shown dependence of time for signal processing from number of processing frames. Comparing curves on Fig. 3 can be concluded that for same frame number, more time is needed when CPU is used as DF processor than GPU NVIDIA Tesla K20X.

By increasing frame number, thereby increasing the amount of data, time for processing grow up faster when CPU is used as DF processor. GPU becomes more effective for a large amount of data. Also, at Fig. 3 is shown time for processing at GPU when real signals are used. When algorithm for DOA estimation is performed on real signals, signals acquired from the DF receiver are fed to polyphase filter bank implemented at field programmable gate array (FPGA) and after that, data are fed to GPU. In this case time for processing, for example 10 frames of signal, is less than 20 ms using GPU NVIDIA Tesla K20X. For DOA estimation on simulated signal, it takes time to read data from file in which signal is saved.

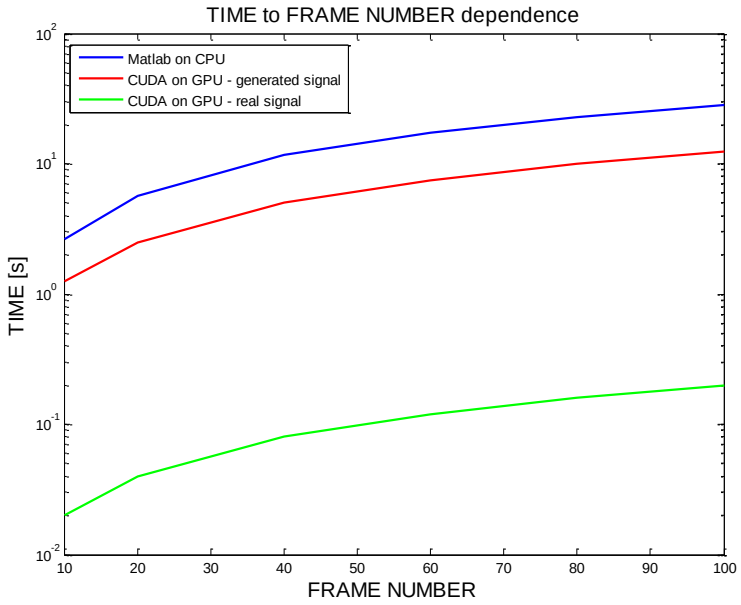

Fig.3 Time to frame number dependence

5. Conclusion

Achieving high processing speed in modern WDF is very important in order to provide high scanning speed. In this paper, we presented our experience with implementation of one method for DOA estimation in wideband direction finder – correlative interferometer method on the GPU. In this paper using GPU NVIDIA Tesla K20X as WDF processor has been proposed. Obtained results give some promising facts that real-time processing and estimation of DOAs in modern WDF using GPU as processor is achievable.

6. References

[1] A.Rembovsky, A.Ashikhmi, V.Kozmin, S.Smolskiy, *Radio Monitoring, Springer*, New York, 2009.

[2] M. Misic, D. Djurdjevic, M. Tomasevic, "Evolution and Trends in GPU computing", *Proc. of the 35th International Convention MIPRO*, Abbazia, Croatia, 2012., pp. 289-294

[3] http://www.nvidia.com/content/PDF/kepler/Tesla-K20X-BD-06397-001-v05.pdf

[4] Tuncer T E and Friedlander B. *Classical and modern direction-of-arrival estimation*, Academic Press, USA, 2009.

[5] Guo, Dong Liang, and Zhong Hua Li. "A fast direction finding algorithm based on correlation processing." *Applied Mechanics and Materials*. Vol. 373. Trans Tech Publications, 2013.

[6] D. B. Kirk, and W. M. Hwu, "*Programming Massively Parallel Processors: A Hands-on Approach*", Morgan Kaufmann, 2010.