APPLICATION OF ADAPTIVE COMPUTING IN SATELLITE TELEMETRY PROCESSING

Marco Figueiredo
Stinger Ghaffarian Technologies, Inc.
NASA-Goddard Space Flight Center, Code 564

Terry Graessle
Lockheed Martin Space Mission Systems & Services
NASA-Goddard Space Flight Center, Code 564

ABSTRACT

The advent of adaptive computers built from re-programmable logic devices presents a potential solution for meeting the data processing requirements of the new era of Earth monitoring satellites to be launched by the National Aeronautics and Space Administration (NASA) Earth Science Enterprise project. The Earth Observing System (EOS) AM-1 spacecraft, the first satellite of this new era, will produce in only six months as much data as NASA has collected to this date. As a consequence, the Earth Science Data and Information System (ESDIS) project is building high performance and highly costly parallel processing systems to address the real-time data production requirements. Together with the high performance front-end ingest and level 0 processing microcircuits developed in-house at the Goddard Space Flight Center's (GSFC) Data Systems Technology Division (DSTD), adaptive computers present a possible alternative to traditional CPU-based systems to increase the performance while reducing the cost of satellite telemetry processing systems.

The Adaptive Scientific Data Processing (ASDP) project has been investigating the use of adaptive computers in the implementation of space borne scientific data processing systems. An order of magnitude processing speed acceleration over high-end workstations has been demonstrated for both level 1 and level 3 algorithms. This paper discusses the use of adaptive computing in satellite telemetry processing systems, level 1 and beyond. Primarily, it describes the efforts and presents the results of two prototypes developed by the ASDP project. The limitations of the current state of the technology are discussed and the expected improvements to facilitate the adoption of adaptive computers are presented. Finally, future work of the ASDP project is discussed.

KEY WORDS

FPGA, Adaptive Computing, Reconfigurable, Telemetry Processing, NASA

THE EOS PROCESSING PROBLEM

Beginning with the EOS era of satellites, NASA's data processing requirements will significantly increase. Current NASA-supported data holdings total about 125,000 Gigabytes [1]. Table 1 displays the average daily data volume for the EOS AM-1 spacecraft [3], which is scheduled for launch in 1999, along with the average processing load required to generate this data. The EOS AM-1 spacecraft will produce more than 900 Gigabytes of data daily and require over 11,400 million floating-point operations per second (Mflops) of computing power. Therefore, in less than 6 months, the volume of EOS AM-1 spacecraft data will surpass all previous NASA data holdings.

To solve this problem, the Earth Science Data and Information System (ESDIS) project is creating several Distributed Active Archive Centers (DAACs) around the country to process and archive the data. Each DAAC uses numerous powerful workstations operating simultaneously on different data sets. By using these parallel-processing techniques, the DAACs are able to manage the vast amounts of data.

Level	Average Daily Data Volume (Gbytes)	Average Processing Load (Mflops)
1A	254.735	99.922
1B	349.4	2704.212
2	245.514	5628.791
3	68.417	3051.828
4	0.230	.050
Total:	918.3	11484.803

Table 1. Average Daily Data Volume and Processing Load for the EOS AM-1 Spacecraft [3]

THE EOS SCIENCE DATA FLOW

Figure 1 illustrates the flow of science data for a typical satellite. Data from each instrument is encoded into packets and frames and transmitted to Earth. At the ground station, level 0 processing is performed for the entire spacecraft data stream. Higher levels of processing are then performed for each instrument as required.

The Earth Observing System (EOS) project defines the various levels of telemetry processing in the following manner [2]:

Level 0 - Reconstructed, unprocessed instrument payload data at full resolution; any and all communications artifacts, e.g., synchronization frames, communications headers, duplicate data removed.
Level 1A - Reconstructed, unprocessed instrument data at full resolution, time-referenced, and annotated with ancillary information, including radiometric and geometric calibration coefficients and georeferencing parameters, e.g., platform ephemeris, computed and appended but not applied to the Level 0 data.
Level 1B - Level 1A data that have been processed to sensor units (not all instruments will have a Level 1B equivalent).
Level 2 - Derived geophysical variables at the same resolution and location as the Level 1 source data.
Level 3 - Variables mapped on uniform space-time grid scales, usually with some completeness and consistency.
Level 4 - Model output or results from analyses of lower level data, e.g., variables derived from multiple measurements.

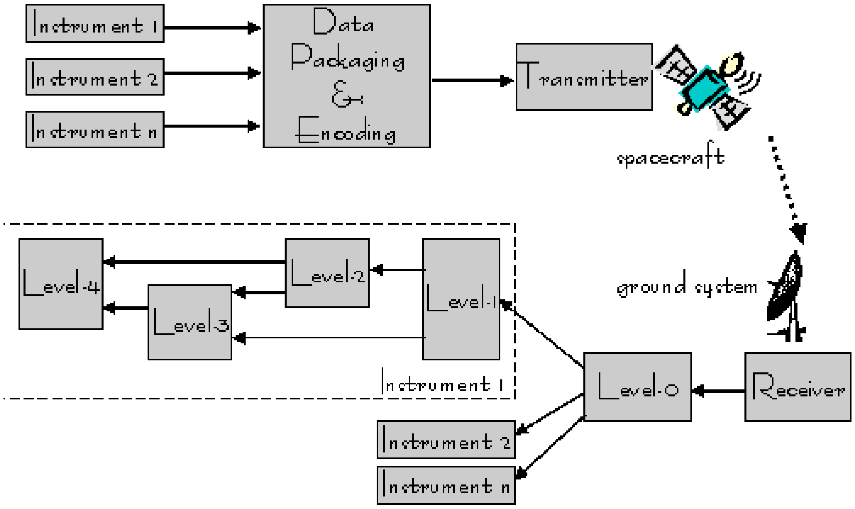

Figure 1. Typical Satellite Data Flow

THE ASDP PROJECT MISSION

The Adaptive Scientific Data Processing project is part of the Microelectronics and Signal Processing branch at NASA-Goddard Space Flight Center, Greenbelt, Maryland. Funded by ESDIS, ASDP is a research and development project aimed at investigating adaptive computing technology and its application to satellite telemetry processing. A primary goal of the ASDP project is to develop low cost, high performance processing solutions to help meet NASA’s growing computing needs.

In the early 1990's, the Goddard Space Flight Center's Microelectronics Systems Branch (Code 521) started the utilization of Very Large Scale Integration (VLSI) Application Specific Integrated Circuits (ASIC) to accelerate and reduce the cost of Level 0 satellite telemetry data processing. Currently, the Desktop Satellite Data Processing (DSDP) system is being developed to provide low-cost integrated ground system solutions from the antenna to the network. Due to the fact that level 1 processing is instrument dependent, an acceleration approach utilizing ASICs is not feasible. The advent of field-programmable gate array (FPGA) based computing provides processing performance close to ASIC levels while maintaining much of the programmability of traditional microprocessor based systems. As a result, the incorporation of adaptive computers to the DSDP is a natural path for the acceleration of science data processing (levels 1 to 4). Over the past two years, the ASDP team has completed two demonstrations of adaptive computing technology applied to the field of telemetry processing. The first, a multispectral image classification algorithm for a current generation satellite, Landsat-2, classifies as a level 3 process. The second is a level 1, instrument calibration for a new EOS generation satellite.

MULTISPECTRAL IMAGE CLASSIFICATION

The Multispectral image classification algorythm processes each pixel of an image to determine which type of terrain is being represented (i.e. water, forest, urban, etc.). The algorithm used was developed by Chettri, et al ([4], [5]) and is based on Probabilistic Neural Networks (PNN). The algorithm takes each image pixel, represented by a vector, and compares the pixel against sets of training vectors, called weights, which are known to belong to certain classes (water, forest, urban, etc.). For each class, the PNN calculation derives a value indicating the probability that the pixel belongs to that particular class. After all classes are processed, the class with the highest probability is the class to which the pixel most likely belongs. This calculation is represented by equation 1 [5].

$$f\left(\vec{X} \mid S_k\right) = \frac{1}{(2\pi)^{d/2}\sigma^d}\frac{1}{P_k}\sum_{i=1}^{P_k}\exp\left[-\frac{\left(\vec{X}-\vec{W}_{ki}\right)^T\left(\vec{X}-\vec{W}_{ki}\right)}{2\sigma^2}\right] \text{, where}$$

$\vec{X}$ = Pixel Vector

$\vec{W}_{ki}$ = weight vector i of class k

P_k = number of weight vectors for class k

d = number of bands

k = number of classes

σ = smoothing parameter

Figure 2 displays the board used to implement the PNN calculation, the G900 Spectrum System board from Giga Operations Corporation. The board shown in Figure 2 has four modules installed, each of which contains 2 FPGA devices. The PNN algorithm was small enough to be implemented on a single module (2 Xilinx XC4013 FPGA's). To increase the performance of the system, two modules were configured with the PNN algorithm, which operated on separate pixels in parallel.

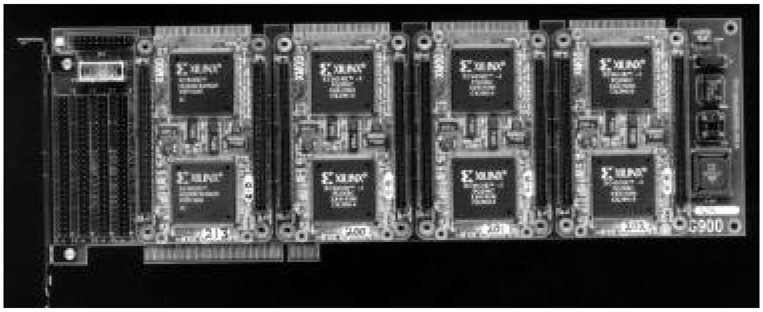

Figure 2. Giga Operations G900 Spectrum System Board

Figure 3 represents the data flow for the PNN algorithm. The host processor is responsible for reading in the image and downloading the FPGA configuration files and image pixels to the adaptive coprocessor. After processing, data is returned to the host and the results are displayed. Refer to the ASDP- Fall 1996 Report [6] for further information regarding the implementation of the PNN algorithm.

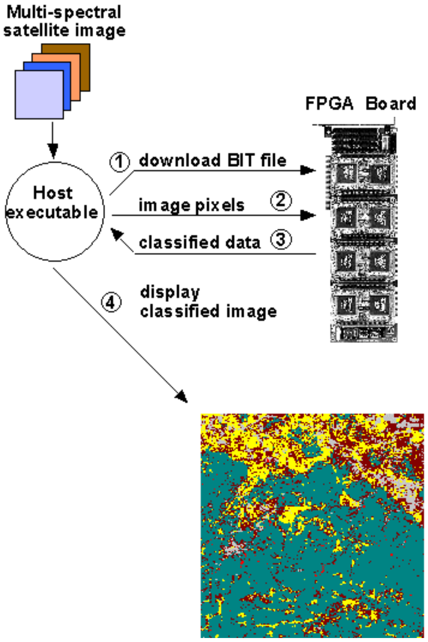

Figure 3. Data Flow for PNN Calculation Implementation

The results obtained for the multispectral image classification algorithm are listed in Table 2. Results for the adaptive coprocessor are shown using both 1 and 2 modules. The version using the FPGA coprocessor board is significantly faster than the software only version. Using 2 FPGA modules hosted in an inexpensive Pentium PC, the FPGA version provides more than 8 times faster performance than a 200 MHz DEC Alpha machine (1 min, 26 sec. vs. 12 min). Higher performance can be achieved through the utilization of up to 16 FPGA modules.

System	Time to calculate a 512x512 image	x Acceleration
VLB Pentium PC 100 MHz	34 min 3 sec	1
+1 FPGA Coprocessor module @ 16 MHz	3 min 40 sec	9.29
+2 FPGA Coprocessor module @ 16 MHz	2 min 30 sec	13.62
PCI Pentium PC 166 MHz	17 min 02 sec	1
+1 FPGA Coprocessor module @ 16 MHz	2 min 34 sec	6.64
+2 FPGA Coprocessor module @ 16 MHz	1min 26 sec	11.88
1 Processor DEC ALPHA 200 MHz	12 min	-

Table 2. Multispectral Image Classification Results

MODIS REFLECTIVE CALIBRATION

The Moderate Resolution Imaging Spectroradiometer (MODIS) instrument is one of the five instruments on board the EOS AM-1 spacecraft. As shown in Table 3, more than ½ of the 918 Gigabytes of data generated daily by the EOS AM-1 spacecraft (Table 2) will come from the MODIS instrument.

Level	Average Daily Data Volume (Gbytes)	Average Processing Load (Mflops)
1A	115.099	99.917
1B	193.6	1339.854
2	161.859	1293.981
3	66.312	3005.218
4	0.230	.050
TOTAL:	537.1	5739.02

Table 3. Average Daily Data Volume and Processing Load for the MODIS Instrument [3]

The MODIS processing will consume nearly ½ of the 11484 Mflops of computing resources required to generate this data. Figure 3 describes the data flow for MODIS Level 1B telemetry processing. After extensive analysis of the L1A-to-L1B software, it was determined that the reflective calibration block was the most computationally intensive portion of the L1A-to-L1B algorithm and therefore the most likely to benefit from hardware acceleration. For more details on the reflective calibration algorithm refer to the “Algorithm Theoretical Basis Document” for the MODIS instrument [7].

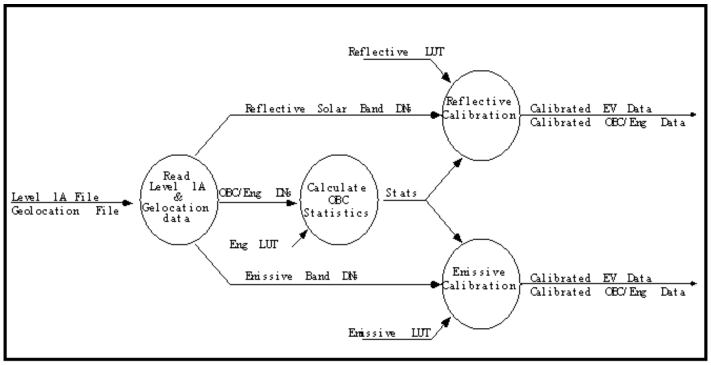

Figure 3. MODIS Level 1B Data Flow

Figure 4 displays the Wildforce board from Annapolis Micro Systems, which was used to implement the MODIS reflective calibration. The Wildforce was chosen because it is PCI compatible, contained sufficient FPGA resources for the application and was available with drivers for a Silicon Graphics workstation. The host platform was a Silicon Graphics Origin 200 workstation with a 180 MHz R10000 RISC microprocessor and 128 Mbytes of RAM. This platform was chosen because the workstations in the DAAC use identical processor and memory configurations. This provides a benchmark to evaluate the impact of inserting adaptive computing into the DAAC environment.

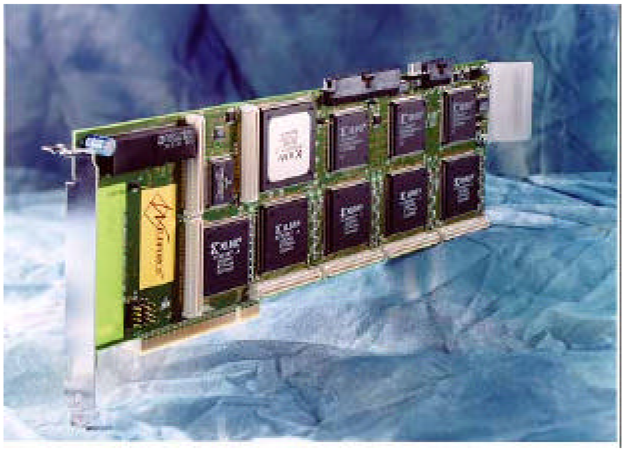

Figure 4. Annapolis Micro Systems Wildforce Board

To implement the reflective calibration algorithm, the design was partitioned among the available devices and implemented and tested in an incremental fashion. For further details on the MODIS reflective calibration algorithm and our implementation, refer to the ASDP – Fall 1997 Report [8].

The results from the MODIS reflective calibration demonstration are displayed in Table 4. The software version, the software that will be running in the DAAC, took 16.51 seconds to complete the reflective calibration function; while the FPGA version only took 1.78 seconds. This demonstrated a performance increase 9 times greater than the software only version.

Reflective Calibration Implementation Method	Time (seconds)
Software Only	16.51
FPGA Coprocessor Card	1.78
X Acceleration	9.28

Table 4. MODIS Reflective Calibration Results

STATE OF ADAPTIVE COMPUTING TECHNOLOGY

The field of adaptive computing is in its infancy. The technology began with the creation of SRAM-based FPGA devices by Xilinx, Inc. in 1985. These devices can be reconfigured to perform a different function without removing the device from a circuit card or removing power from the system. Since then, FPGA device architectures have continuously improved. Today, devices are available containing more than 100,000 logic gates, which operate above 80 MHz. As semiconductor fabrication techniques continue to improve, larger and faster FPGAs will become available.

The most common use for FPGA devices is to prototype Application Specific Integrated Circuits (ASICs), and in embedded systems having low volume requirements. Adaptive computing is considered a small segment of the FPGA market, which hardly represents an income to device manufacturers. However, adaptive computing is seen as the market segment with the most potential for growth and the most potential of establishing FPGAs as mainstream computing elements. Applications developed using adaptive computing have demonstrated several orders of magnitude acceleration over microprocessor based solutions.

The development of adaptive computing applications requires designers who have an in-depth knowledge of both the hardware and software. Algorithms must be thoroughly analyzed to determine their fitness for hardware implementation. The algorithms must then be partitioned between hardware and software. Once the hardware portion of the algorithm is determined, the adaptive coprocessor must be selected. The hardware design must be partitioned among the available FPGA devices on the board. Care must be taken to insure that all necessary data can be passed between FPGA devices, and that all input parameters required by each FPGA can be input to the board. If the FPGA board contains insufficient resources for the design, additional boards can be added or the limited FPGA resources can be repeatedly reconfigured to implement successive pipelined processing stages, in a process known as run-time reconfiguration, a technique that increases the complexity of the design. Reconfiguring an FPGA device requires several milliseconds, so run-time reconfiguration can greatly impact performance. Also, the board must have sufficient memory to store all data between configurations, or bandwidth limitations between the host and FPGA board will eliminate any performance gains.

A characteristic of current adaptive computers is insufficient bandwidth between the host processor and the FPGA board, which is usually the limiting factor in application acceleration. As a consequence, applications performing a large amount of computation on a small amount of data will derive the greatest benefit from adaptive computing, whereas I/O intensive applications will derive lesser benefit.

To allow adaptive computing to move into the realm of mainstream computing, more advanced tools, which simplify the design process, will be required. Ideally, a tool that allows the specification of all system parameters using a high-level graphical interface would be available. This tool would provide the means for a scientist to enter the specification or description of the physics of the application under development. A computer engineer would then utilize the same tool to automatically partition the algorithm between the host and the adaptive coprocessor and implement the design.

Improvements in the architecture of FPGA devices are also required to facilitate the widespread adoption of adaptive computing. Devices should fully support partial reconfiguration at the bit level. Device configuration time should be reduced to the microseconds range, allowing easier and more efficient run-time reconfiguration. Finally, FPGA architectures providing forward compatibility, to allow for easy system upgrades, should become commercially available.

The sole justification for the utilization of adaptive computing today is the capability to attain orders of magnitude performance gains over microprocessor based systems. The major hurdle to overcome in the process of realizing such potential is the difficulties involved in the application development. Several studies are being supported by the Defense Advanced Research Projects Agency’s (DARPA) Adaptive Computing Systems study to address these technology limitations [9]. The ASDP group is collaborating with such efforts by providing an interface between the research community and the application challenges under question at NASA.

FUTURE WORK

The ASDP team is currently working on the level 1 processing stage of a direct broadcast receiving system for the MODIS instrument, the Adaptive Level One Accelerator (ALOA). The goal of this project is to achieve near real-time performance using adaptive computing technology. This work is an expansion of the previous work on MODIS reflective calibration to do the entire MODIS level 1 processing.

Another possible area for future work concerns the use of adaptive computing technology onboard a spacecraft. A radiation hard FPGA is currently being developed by Honeywell, Inc. under a contract with the Goddard Space Flight Center, which is expected to be available before the turn of the century. This device will make possible an adaptive computer capable of being flown in space. This would allow level 1 and higher processing to be done directly on the spacecraft, which in turn would reduce the cost of ground systems, allowing data to be transmitted directly to the end user.

CONCLUSION

This paper discussed NASA’s growing need for increased data processing capabilities. The two prototypes of the ASDP project were described. These prototypes have demonstrated an order of magnitude performance increase over traditional computing methods. The current state of adaptive computing technology was discussed, along with the future work of the ASDP project.

REFERENCES

[1] R. Obenschain, “EOS Data and Information System (EOSDIS), Status and Plan”, presentation, January 16, 1998.

[2] G. Asrar (Ed.), R. Greenstone (Ed.) “*1995 MTPE EOS Reference Handbook*”, NASA Goddard Space Flight Center, Mission to Planet Earth, Earth Observing System, pp. 44, 1995.

[3] “*EOS Output Data Products, Processes, and Input Requirements, Version 4.0”*, Science Office, Earth Science Data Information System Project, Goddard Space Flight Center, Appendix B, April 1996.

[4] S. R. Chettri, R. F. Cromp, M. Birmingham, “Design of neural networks for classification of remotely sensed imagery”*, Telematics and Informatics,* Vol. 9, No 3, pp. 145-156, 1992.

[5] S. R. Chettri and R. F. Cromp, "Probabilistic neural network architecture for high-speed classification of remotely sensed imagery", *Telematics and Informatics*, Vol. 10, No 4, pp. 187-198, 1993.

[6] M. Figueiredo, N. Speciale, "*Adaptive Scientific Data Processing – Fall 1996 Report*", http://fpga.gsfc.nasa.gov/asdp/fall96rep/fall96rep.htm

[7] "Draft of the MODIS Level 1B Algorithm Theoretical Basis Document Version 2.0", ATB MOD-01, SAIC/GSC MCST Document, February 13, 1997.

[8] M. Figueiredo, T. Graessle, K. Winiecki, D. Greenspan, "*Adaptive Scientific Data Processing – Fall 1997 Report*", http://fpga.gsfc.nasa.gov/asdp/fall97rep/fall97rep.htm

[9] DARPA ITO Adaptive Computing Systems Study http://www.darpa.mil/ito/research/acs/index.html