An Efficient Implementation of Automatic Cloud Cover Assessment (ACCA) on a Reconfigurable Computer

Esam El-Araby
The George Washington University
Washington, DC 20037
USA
esam@gwu.edu

Mohamed Taher
The George Washington University
Washington, DC 20037
USA
mtaher@gwu.edu

Tarek El-Ghazawi
The George Washington University
Washington, DC 20037
USA
tarek@gwu.edu

Jacqueline Le Moigne
NASA Goddard Space Flight Center (GSFC)
Greenbelt, MD 20771
USA
lemoigne@backserv.gsfc.nasa.gov

Abstract – **Clouds have a critical role in many studies, e.g. weather- and climate-related studies. However, they represent a source of errors in many applications, and the presence of cloud contamination can hinder the use of satellite data. This requires a cloud detection process to mask out cloudy pixels from further processing.**

The trend for remote sensing satellite missions has always been towards smaller size, lower cost, more flexibility, and higher computational power. Reconfigurable Computers (RCs) combine the flexibility of traditional microprocessors with the power of Field Programmable Gate Arrays (FPGAs). Therefore, RCs are a promising candidate for on-board preprocessing.

This paper presents the design and implementation of an RC-based real-time cloud detection system. We investigate the potential of using RCs for on-board preprocessing by prototyping the Landsat 7 ETM+ ACCA algorithm on one of the state-of-the art reconfigurable platforms, SRC-6E.

Although a reasonable amount of investigations of the ACCA cloud detection algorithm using FPGAs has been reported in the literature, very few details/results were provided and/or limited contributions were accomplished. Our work has been proven to provide higher performance and higher detection accuracy.

I. Introduction

The trend for remote sensing satellite missions has always been towards smaller size, lower cost, and more flexibility. On-board processing, as a solution, permits a good utilization of expensive resources. Instead of storing and forwarding all captured images, data processing can be performed on-orbit prior to downlink resulting in the reduction of communication bandwidth as well as in simpler and faster subsequent computations to be performed on ground stations. Consequently, on-board processing can reduce the cost and the complexity of the On-The-Ground/Earth processing systems. Furthermore, it enables autonomous decisions to be taken on-board which can potentially reduce the delay between image capture, analysis and action. This leads to faster critical decisions which are crucial for future reconfigurable web sensors missions as well as planetary exploration missions.

The presence of cloud contamination can hinder the use of satellite data, and this requires a cloud detection process to mask out cloudy pixels from further processing. The Landsat 7 ETM+ (Enhanced Thematic Mapper) ACCA (Automatic Cloud Cover Assessment) algorithm [1] is a compromise between the simplicity of earlier Landsat algorithms, e.g. ACCA for Landsat 4 and 5, and the complexity of later approaches such as the MODIS (Moderate Resolution Imaging Spectroradiometer) cloud mask.

Reconfigurable Computers (RCs) combine the flexibility of traditional microprocessors with the power of Field Programmable Gate Arrays (FPGAs). These platforms have always been reported to outperform the conventional platforms in terms of throughput and processing power within the domain of cryptography, and image processing applications [2]. In addition, they are characterized by lower form/wrap factors compared to parallel platforms, and higher flexibility than ASIC solutions. Therefore, RCs are a promising candidate for on-board preprocessing. The SRC-6E Reconfigurable Computer is one example of this category of hybrid computers [3] and is used here for this purpose.

This paper presents the design and implementation of an RC-based real-time cloud detection system. We investigate the potential of using RCs for on-board preprocessing by prototyping the Landsat 7 ETM+ ACCA algorithm on one of the state-of-the art reconfigurable platforms, SRC-6E.

Although a reasonable amount of investigations of the ACCA cloud detection algorithm using FPGAs has been reported in the literature [4], [5], very few details/results were provided and/or limited contributions were accomplished. Our work is unique in providing higher performance and higher detection accuracy.

II. Description of The Automatic Cloud Cover Assessment (ACCA) Algorithm

Theory of Landsat 7 ETM+ ACCA algorithm is based on the observation that clouds are highly reflective and cold. The high reflectivity can be detected in the visible, near- and mid- IR bands. The thermal properties of clouds can be detected in thermal IR band. Table I presents the bands wavelengths and its detection features.

TABLE I
LANDSAT 7 ETM+ BANDS

Band	Wavelength (μm)	Detection Features
2 (green)	0.525 - 0.605	- Measures green reflectance - Vegetation discrimination
3 (red)	0.630 - 0.690	- Measures Chlorophyll absorption - Plant Species differentiation
4 (near-IR)	0.775 - 0.900	- Determines soil moisture level - Delineating water bodies and distinguishing vegetation types
5 (mid-IR)	1.55 - 1.75	- Supplies information about vegetation and soil moisture - Differentiation of snow from clouds
6 (Thermal IR)	10.4 - 12.5	- Thermal mapping to Brightness Temperatures

The Landsat 7 ETM+ ACCA algorithm recognizes clouds by analyzing the scene twice. In the first pass eight filters are utilized for this purpose, see Table II.

TABLE II
PASS-ONE FILTERS

	Filter	Function
1	**Brightness Threshold** $B_3 > 0.08$	Eliminates dark images
2	**Normalized Difference Snow Index (NDSI)** $NDSI = \frac{B_2 - B_5}{B_2 + B_5} < 0.7$	Eliminates many types of snow
3	**Temperature Threshold** $B_6 < 300K$	Eliminates warm image features
4	**Band 5/6 Composite** $(1 - B_5)B_6 < 225$	Eliminates numerous categories including ice
5	**Band 4/3 ratio** $\frac{B_4}{B_3} < 2$	Eliminates bright vegetation and soil
6	**Band 4/2 ratio** $\frac{B_4}{B_2} < 2$	Eliminates ambiguous features
7	**Band 4/5 ratio** $\frac{B_4}{B_5} > 1$	Eliminates rocks and desert
8	**Band 5/6 Composite** $(1 - B_5)B_6 > 210 \Rightarrow warm\ \ clouds$ $(1 - B_5)B_6 < 210 \Rightarrow cold\ \ clouds$	Distinguishes warm clouds from cold clouds

Omission errors are expected. The goal of pass-one is to develop a reliable cloud signature for use in pass-two where the remaining clouds are identified. Commission errors, however, create algorithm failure and must be minimized. Three categories result from pass-one: clouds, non-clouds, and an ambiguous group that are revisited in pass-two.

John A. Williams et al., [5], [6], have used band mapping techniques to implement Landsat-based algorithms on MODIS data. The generalized and modified classification rules for Pass-One are shown in Table III.

Pass-Two resolves the detection ambiguity resulted from Pass-One. Thermal properties of clouds identified during Pass-One are characterized and used to identify remaining cloud pixels. Band 6 statistical moments (mean, standard deviation, distribution skewness, kurtosis), see the equation (1), are computed and new adaptive thresholds are determined accordingly. The 95th percentile, i.e. the smallest number that is greater than 95% of the numbers in the given set of pixels, becomes the new thermal threshold for Pass Two.

$$\left.\begin{aligned} \eta &= \frac{1}{n}\sum_{i=1}^{n} x_i \\ \sigma^2 &= \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \eta)^2 \\ Skewness &= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - \eta}{\sigma}\right)^3 \\ Kurtosis &= \frac{1}{n-3}\sum_{i=1}^{n}\left(\frac{x_i - \eta}{\sigma}\right)^4 \end{aligned}\right\} \quad (1)$$

TABLE III
GENERALIZED CLASSIFICATION RULES FOR PASS-ONE [5]

Classification	Rule
Snow	$\left(NSDI = \frac{B_2 - B_5}{B_2 + B_5} > 0.7\right)\ AND\ (B_4 > 0.1)^A$
Desert	$\frac{B_4}{B_5} < 0.83^B$
NotCloud	$(B_3 < 0.08)\ OR\ (B_6 > 300)\ OR\ (Snow)$
Ambiguous	$\left((1 - B_5)B_6 > 225\right)\ OR\ \left(\frac{B_4}{B_5} > 2\right)\ OR\ \left(\frac{B_4}{B_2} > 2\right)\ OR\ (Desert)\ \ AND\ (\sim NotCloud)$
ColdCloud	$((1 - B_5)B_6 \geq 210)\ AND\ (\sim Ambiguous)\ AND\ (\sim NotCloud)$
WarmCloud	$((1 - B_5)B_6 < 210)\ AND\ (\sim Ambiguous)\ AND\ (\sim NotCloud)$

Notes:
[A] The Band 4 brightness test, in the snow test, was added after observations that the NDSI (Normalized Difference Snow Index) algorithm applied to MODIS data incorrectly labeled many cloud pixels as snow.
[B] The desert detection threshold was lowered to 0.83, from the original ACCA value of 1.0, after it was observed that many cloud pixels were incorrectly classified as desert. The value of 0.83 was determined experimentally.

Image pixels that fall below the new thermal threshold and survive the first three Pass-One filters are classified as cloud pixels. Specifically, the following three conditions must be satisfied:

- Desert index (Filter 7) is greater than 0.5
- Colder cloud population exceeds 0.4 percent of the scene
- Mean temperature of the cloud class is less than 295K

During processing, a cloud mask is created. The final step is processing the cloud mask for holes. After the two ACCA passes, a filter is applied to the cloud mask to fill in cloud holes. This filtering operation works by examining each non-cloud pixel in the mask. If 5 out of the 8 neighbors are clouds then the pixel is reclassified as cloud. Cloud cover results from both Pass-One and Pass-Two are compared. Extreme differences are indicative of cloud signature corruption. When this occurs, Pass-Two results are ignored and all results are taken from Pass-One. The final cloud cover percentage for the image is calculated based on the filtered cloud mask. The cloud pixels in the mask are tabulated and a cloud cover percentage score for the scene is computed.

III. SRC-6E Reconfigurable Computer

A. Hardware Architecture

SRC-6E platform consists of two general-purpose microprocessor boards and one MAP® reconfigurable

processor board. Each microprocessor board is based on two 1GHz (3GHz) Pentium 3 (Pentium 4) microprocessors. The SRC-6E MAP board consists of two MAP reconfigurable processors. Overall, the SRC-6E system provides a 1:1 microprocessor to FPGA ratio. Microprocessors boards are connected to the MAP board through the SNAP® interconnect. The SNAP card plugs into the DIMM slot on the microprocessor motherboard [3] to provide higher data transfer rates between the boards than the inefficient but common PCI solution. The peak transfer rate between a microprocessor board and the MAP board is 1600 MB/sec for the Pentium 4 version of SRC-6E.

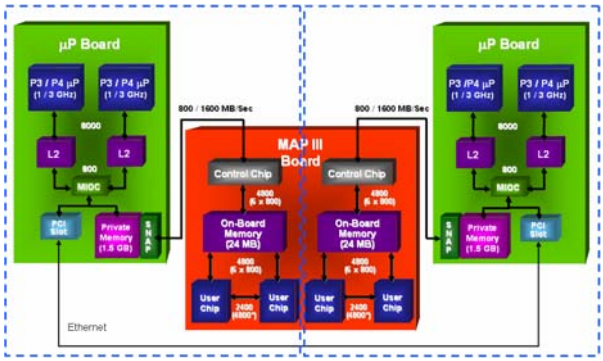


Fig. 1 Hardware Architecture of SRC-6E

Hardware architecture of the SRC-6E MAP processor is shown in Fig. 1. The MAP board is composed of two identical MAP units. Each MAP unit has one control FPGA and two user FPGAs, all Xilinx Virtex II-6000-4. Additionally, each MAP unit contains six interleaved banks of the on-board memory (OBM) with a total capacity of 24 MB. The maximum aggregate data transfer rate among all FPGAs and on-board memory is 4800 MB/s. The user FPGAs are configured in such a way that one is in the master mode and the other is in the slave mode. The two FPGAs of a MAP are directly connected using a bridge port. Furthermore, MAP processors can be chained together using a chain port to create an array of FPGAs

In the typical mode of operation, input data is first transferred through the Control FPGA from the microprocessor memory to OBM. This transfer is followed by computations performed by the User FPGA, which fetches input data from OBM and transfers results back to OBM. Finally, the results are transmitted back from OBM to microprocessor memory.

B. Programming Model

The SRC-6E has a somewhat similar compilation process as a conventional microprocessor-based computing system, but also produces logic for the MAP reconfigurable processor, see Fig. 2.

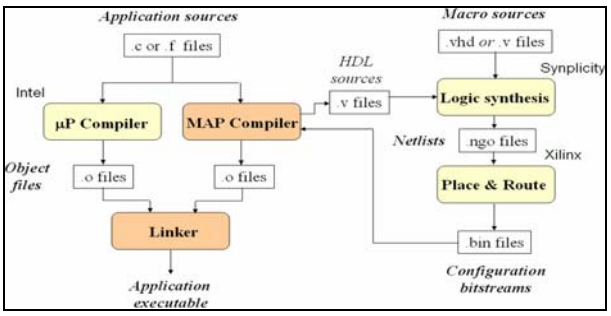


Fig. 2 SRC Compilation Process

There are two types of application source files to be compiled. Source files of the first type are compiled with the Intel processor as the target execution platform. Source files of the second type are compiled with execution on the MAP reconfigurable processor as a target. A file that contains a program to be executed on the Intel processor is compiled using the traditional microprocessor compiler. All files containing functions that call hardware macros and thus execute on the MAP are compiled by the MAP compiler. MAP source files contain MAP functions are mainly composed of macro calls. Here, a macro is defined as a piece of hardware logic designed to implement a certain function. Since users often wish to extend the built-in set of operators, the compiler allows users to integrate their own VHDL/Verilog macros.

IV. Implementation Details and Experimental Results

The ACCA algorithm adapted for Landsat 7 ETM+ data has been implemented both in C and Matlab, and pass-one has been implemented and synthesized for the Xilinx XC2V6000 FPGA on SRC-6E.

Fig 3. shows the implementation architecture of pass-one of the algorithm. This module reads the five band-image and feeds them to six filters generating the output mask. The function of each mask is listed in Table II.

Our goal for the implementation was to achieve comparable performance and detection accuracy to what has been reported in [4] and [5]. Therefore, the only constraint to our design was the processing speed, as measured by throughput. This constraint is approached through full-pipelining of the design.

Many of the tests in pass-one are threshold tests of ratio values, such as the snow test. We found out that it was more efficient, in terms of the required resources, to multiply one value by the threshold, and compare with the other value, instead of performing the division then comparing against the threshold.

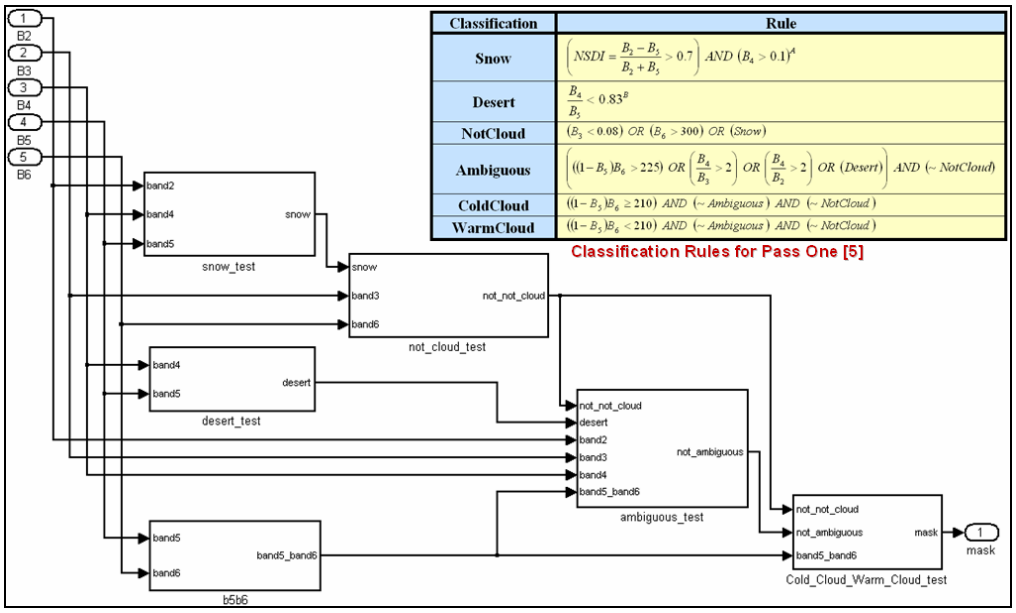


Fig. 3 Pass-One Module

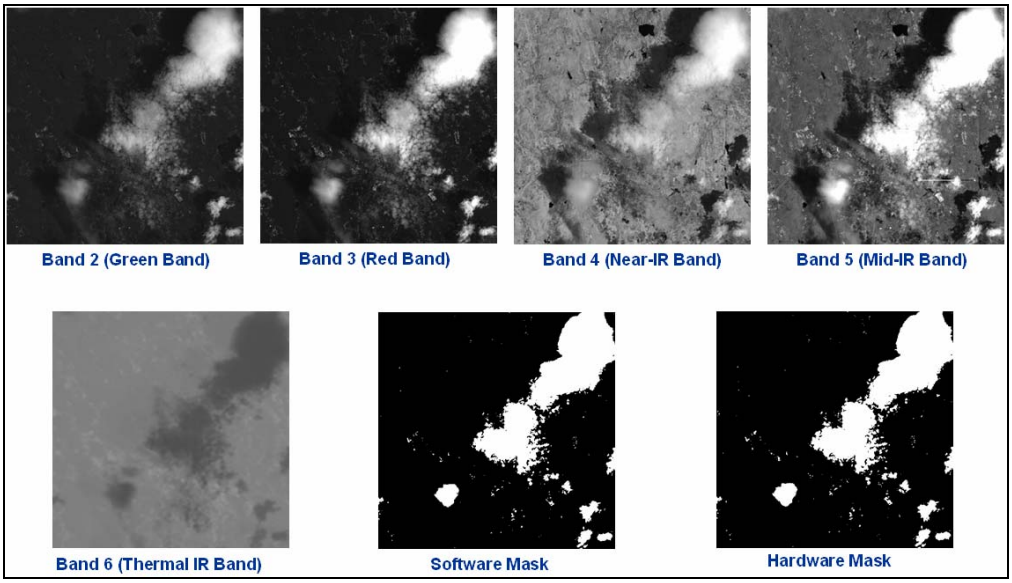
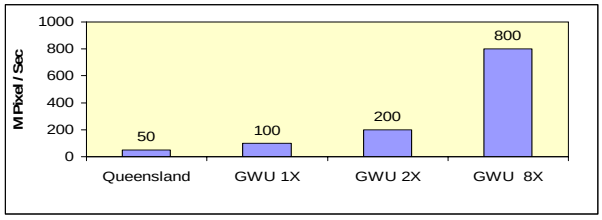


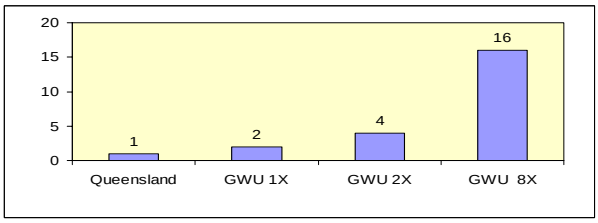
Fig. 4 Detection Accuracy

The design was developed in VHDL, synthesized, placed and routed, and was found to occupy approximately 7% of the available logic resources on the chip. This enabled us to instantiate eight concurrent processing engines of the design in the same chip, which increased the performance to eight folds of what we expected. The total resources utilization for the eight engine version was approximately 57%. This leaves plenty of room for more engines and/or processing functions to be implemented on the same chip. The maximum operational clock speed of the design is 100MHz which resulted in 4000 Megapixels/sec (5 inputs x 8 engines x 100MHz) as data input/consumption rate. Furthermore, the data output/production rate was 800 Megapixels/sec (1 output x 8 engines x 100MHz).

Fig. 4 shows the results obtained from a 2.8GHz Intel Xeon processor and from SRC-6E. The hardware (fixed-point) implementation provided a very high performance (28 times faster) compared to the 2.8GHz Xeon implementation. We have obtained approximately an ideal detection accuracy (0% detection error) as compared to software floating-point reference results. These results were achieved by performing the internal algorithm filtering with full-precision (23-bit) fixed-point arithmetic.

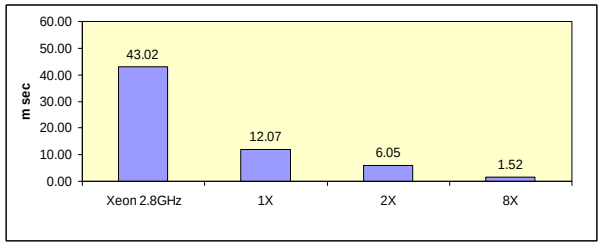


(a) Throughput

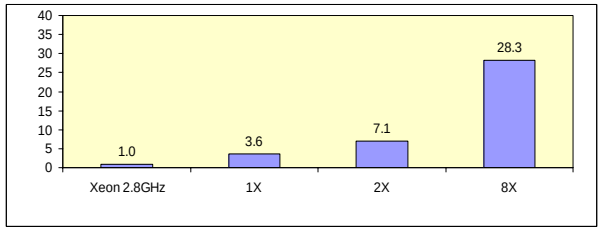


(b) Speedup

Fig. 5 Hardware-to-Hardware Performance



(a) Execution Time



(b) Speedup

Fig. 6 Hardware-to-Software Performance

Our results have been compared to previous work by Williams et al. [5]. Fig. 5 shows those comparisons. Three hardware implementations have been compared: 1X (one engine is instantiated), 2X (2 engines are instantiated), and 8X (8 engines are instantiated). Our implementations achieve a speedup of up to 16 times compared to those reported in [5].

The superiority of RCs over traditional platforms for cloud detection is demonstrated through the performance plots shown in Fig. 6. Our implementations achieve a speedup of up to 28.3 compared to 2.8GHz Intel Xeon processor.

V. Conclusions

This paper presented the design and implementation of an RC-based real-time cloud detection system. We investigated the potential of using RCs for on-board preprocessing by prototyping the Landsat 7 ETM+ ACCA algorithm on one of the state-of-the art reconfigurable platforms, SRC-6E.

Our work has been proven to provide higher performance and higher detection accuracy than previously reported results. The higher performance is achieved through full-pipelining and superscaling (up to 8 concurrent engines), and thus achieving 4000 Megapixels/sec as a data consumption rate and 800 Megapixel/sec as a data production rate. In addition, the performance has been compared to similar hardware implementations and proved to achieve as high as 16 folds speedup. The speedup compared to a 2.8GHz Xeon implementation has been 28 folds higher. On the other hand, the detection accuracy has been verified against software floating-point reference implementation, and the results revealed identical results.

References

[1] R.R. Irish , “Landsat 7 Automatic Cloud Cover Assessment,” Algorithms for Multispectral, Hyperspectral and Ultraspectral Imagery VI, SPIE, Orlando, FL., USA, 24-26 April 2000, pp.348-355.

[2] E. El-Araby, T. El-Ghazawi, J. Le Moigne, and K. Gaj, “Wavelet Spectral Dimension Reduction of Hyperspectral Imagery on a Reconfigurable Computer,” IEEE International Conference on Field-Programmable Technology, FPT 2004, Brisbane, Australia, December 2004.

[3] “SRC-6E C-Programming Environment Guide”, SRC Computers, Inc. 2004.

[4] R.S. Basso, J. Le Moigne, S. Veuella, and R.R. Irish, “FPGA Implementation for On-Board Cloud Detection,” International Geoscience and Remote Sensing Symposium. Hawaii, 20-24 July 2000.

[5] J.A. Williams, A.S. Dawood, S.J. Visser, “FPGA-based Cloud Detection for Real-Time Onboard Remote Sensing,” Proceedings of IEEE International Conference on Field-Programmable Technology (FPT 2002), 16-18 Dec. 2002, pp.110 – 116.

[6] J.A. Williams, A.S. Dawood, S.J. Visser, “Real-Time Wildfire and Volcanic Plume Detection from Spaceborne Platforms with Reconfigurable Logic,” 11th Australasian Remote Sensing and Photogrammetry Conference, Brisbane, Australia, 2-6 September 2002.