Reconfigurable Processing for Satellite On-Board Automatic Cloud Cover Assessment (ACCA)

Esam El-Araby[1], Tarek El-Ghazawi[1], Jacqueline Le Moigne[2], and Richard Irish[2]
[1]*The George Washington University,* [2]*NASA/Goddard Space Flight Center*
{esam, tarek}@gwu.edu, Jacqueline.LeMoigne@nasa.gov, ririsch@pop400.gsfc.nasa.gov

Abstract

Clouds have a critical role in many studies such as weather- and climate- related investigations. However, they represent a source of errors in many applications, and the presence of cloud contamination can hinder the use of satellite data. In addition, sending cloudy data to ground stations can result in an inefficient utilization of the communication bandwidth. This requires satellite on-board cloud detection capability to mask out cloudy pixels from further processing.

Remote sensing satellite missions have always required 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-6. It will be shown that our work provides higher detection accuracy and over one order of magnitude improvement in performance when compared to previously reported investigations.

1. 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 in the 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. In particular, many of the Earth Science missions that have been defined in the recent Decadal Survey [1] will require advanced on-board processing capabilities in order to be successful in producing all the desired Science products without driving the cost out of proportions. One of the recommendations of the report is to "reduce cost risk on recommended missions by investing early in the technological challenges of the missions". We believe that on-board computing is one of the technologies that will enable these future missions. Some examples are the DESDynI (Deformation, Ecosystem Structure, and Dynamic of Ice) and the HYSPIRI (HYperSPectral InfraRed Imager) missions for which on-board computing will enable to concentrate on returning large amount of meaningful information instead of large amount of unprocessed data. This will reduce the space-to-ground communication bottleneck. Such an increased ability of performing on-board pre-processing and "data reduction" operations would also decrease the complexity and the cost of ground systems and increase the availability of processed data products to the users of direct broadcast spacecraft, for example for national emergencies or for educational institutions. The advance of processors such as the one discussed in this paper will make a greater number of algorithm computations feasible on-board, that will not be limited to fixed-point or local operations.

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 [2-4] 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.

Most space systems have a need for reconfigurability for different reasons such as the need of physical upgrade or repair of unmanned spacecraft devices. However, during operation, the systems are physically remote from their operators. Furthermore, space based systems must operate in an environment in which radiation effects have an adverse impact on integrated circuit operation. Ionizing radiation can cause soft-errors in the static cells used to hold the configuration data. This will affect the circuit functionality and can cause system failure.

Reconfigurable Computers (RCs) combine the flexibility of traditional microprocessors with the power of Field Programmable Gate Arrays (FPGAs). The above mentioned requirements of space based systems problems can be fulfilled by using RCs. RC technology allows new hardware circuits (FPGA configurations) to be transmitted over a wireless radio link. This also allows changing the system functionality according to changing mission requirements. Most reconfigurable devices and systems contain SRAM-programmable memory to allow logic and interconnect reconfiguration in the field. For example, the Mars rover mission used a Xilinx FPGA that had not been completely

designed at the time of launch. The FPGA configuration was uploaded to the space craft two months after the launch [5]. In addition, radiation hardened FPGAs are in great demand for military and space applications to reduce cost and cycle time. Actel Corp. has been producing radiation-tolerant antifuse FPGAs for several years for high-reliability space-flight systems. Actel FPGAs have been on board more than 100 launches. Xilinx FPGAs have been used in more than 50 missions [5]. Furthermore, RCs have always been reported to outperform the conventional platforms in terms of throughput and processing power for many different applications such as cryptography, and image processing [6-15]. They are also 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 Signal Data Processor (SDP) and SRC Portable MAPstation [16] are examples of this category of hybrid computers. SDP and SRC Portable MAPstation are based on the SRC-6 line of reconfigurable MAP® processors [17, 18]. Such RC systems can be easily integrated with the Space Cube architecture which has already been space qualified [19, 20].

This paper presents the design and implementation of an RC-based real-time cloud detection system. We investigate the potential and feasibility 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-6. SRC-6 Scalable Systems & Servers [17, 18], which are larger variants of the miniature versions of SDP and Portable MAPstation, are used here as a testbed for this purpose.

Although a reasonable amount of investigations of the ACCA cloud detection algorithm using FPGAs has been reported in the literature [21, 22], very few details/results were provided and/or limited contributions were accomplished. Our work is unique in providing higher performance and higher detection accuracy.

This paper is organized such that section 1 introduces satellite on-board reconfigurable processing, and section 2 provides a detailed description of the ACCA algorithm along with definitions of some of the terminologies used. Section 3 shows the hardware architecture and the programming model of the verification testbed used, SRC-6. Section 3 also includes mathematical derivation of the formulae used in the design of the ACCA architecture. Furthermore, section 4 shows the hierarchical details of the hardware architecture which was designed/implemented for ACCA. In addition, section 5 presents the experimental approaches and results. Section 6 summarizes the paper and provides some observations, while section 7 concludes the paper and suggests work for future considerations.

2. Description of the Automatic Cloud Cover Assessment (ACCA) Algorithm

Providing cloud-free global imaging over the earth's landmasses is the primary goal of the Landsat 7 mission. Success of the mission is determined by the cloud-free nature of each scene acquired. Every image captured by the Landsat 7, is first processed by the ACCA algorithm. This algorithm determines and rates the overall cloud cover. Then these ratios of cloud to non-cloud coverage are reviewed to determine if the data will go on and be processed or discarded. The ACCA is very important to the success of the Landsat 7 mission. It also provides NASA with the ability to determine what signals should be archived and which portions of the Earth's surface must be viewed again. Mission planners use the cloud content of each image to reschedule failed acquisitions.

2.1. Theory

To distinguish between clouds and the terrain below is fairly elementary. Theory of Landsat 7 ETM+ ACCA algorithm is based on the observation that clouds are mainly white, i.e. highly reflective, and colder than the earth's surface. This matches up well with the multispectral response characteristics of the ETM+. 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 1 presents the bands wavelengths and their detection features. However, due to surface and cloud variability, this distinction is not as straightforward as it seems. For this reason a scene dependent approach for identifying clouds was developed. The algorithm handles the cloud population in each scene uniquely by examining the image data twice after a normalization step being performed on the raw data to compensate for temporal data characteristics. The first pass is meant to capture only clouds using eight different filters. The goal of Pass-One is to develop a reliable cloud signature for Pass-Two, where the remaining clouds are identified.

2.2. Normalization Step (Radiometric Formulation)

The images from the Landsat 7 spacecraft are converted from raw data to units that are more useful. Bands 2 through 5 are converted to top-of-atmosphere (TOA) reflectance, and Band 6 is converted to at-sensor temperature. The unit conversions provide a basis for more normalized comparisons of data within a scene and between scenes acquired on different dates. Detector normalization removes within scene responsivity differences, while conversion to reflectance removes the cosine effect of different sun angles across time and space. In the following discussion of this section we include the radiometric formulation and the conversion formulae for the normalization step as well as some related definitions.

The conversion from ETM+ radiometrically raw data to reflectance and temperature is formulated as follows [24]:

Table 1. 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

$$L_{\lambda_i} = \left(\frac{L_{\max_{\lambda_i}} - L_{\min_{\lambda_i}}}{Q_{cal_{\max}} - Q_{cal_{\min}}} \right) \left(Q_{cal_i} - Q_{cal_{\min}} \right) + L_{\min_{\lambda_i}} \quad (1)$$

$$\forall\ i \in \{2,3,4,5,6\}$$

$$\rho_i = \frac{\pi \cdot d^2}{E_{sun_{\lambda_i}} \cos(\theta_s)} \cdot L_{\lambda_i} \ , \ \forall\ i \in \{2,3,4,5\} \quad (2)$$

$$T = \frac{K_2}{\ln\left(\frac{K_1}{L_{\lambda_i}} + 1 \right)} \ , \ i = 6 \quad (3)$$

The parameters for equations (1-3) are defined as follows:

- Captured Data
  - Q_{CAL} = quantized pixel value (e.g. 0-255 for gray-scale)
  - Q_{CALmin} = minimum quantized pixel value (i.e. 0 for gray-scale)
  - Q_{CALmax} = maximum quantized pixel value (i.e. 255 for gray-scale)
- Calibrated Data
  - L_λ = Spectral Radiance at the sensor's aperture in watts/(meter^2 x steradian x µm)
  - ρ = Unitless planetary reflectance calculated for bands 2-5
  - T = Effective at-satellite-temperature in Kelvin calculated for band6 only
- Calibration Constants
  - $L_{MIN\lambda}$ = spectral radiance that is scaled to Q_{CALmin} in watts/(meter^2 x steradian x µm) , see Table 2
  - $L_{MAX\lambda}$ = spectral radiance that is scaled to Q_{CALmax} in watts/(meter^2 x steradian x µm) , see Table 2
  - d = Earth-Sun distance in astronomical units from nautical handbook or interpolated from values shown in Table 3 depending on the Julian day on which the data was acquired
  - $E_{SUN\lambda}$ = Mean solar exoatmospheric irradiances in watts/(meter^2 x µm), see Table 4
  - θ_s = Solar zenith angle in degrees
  - K_1 & K_2 = Thermal calibration constants, see Table 5

Table 2. Spectral Radiance Range ($L_{MIN\lambda}$, $L_{MAX\lambda}$)

ETM+ Spectral Radiance Range watts/(meter squared * ster * µm)								
Band Number	Before July 1, 2000				After July 1, 2000			
	Low Gain		High Gain		Low Gain		High Gain	
	LMIN	LMAX	LMIN	LMAX	LMIN	LMAX	LMIN	LMAX
1	-6.2	297.5	-6.2	194.3	-6.2	293.7	-6.2	191.6
2	-6.0	303.4	-6.0	202.4	-6.4	300.9	-6.4	196.5
3	-4.5	235.5	-4.5	158.6	-5.0	234.4	-5.0	152.9
4	-4.5	235.0	-4.5	157.5	-5.1	241.1	-5.1	157.4
5	-1.0	47.70	-1.0	31.76	-1.0	47.57	-1.0	31.06
6	0.0	17.04	3.2	12.65	0.0	17.04	3.2	12.65
7	-0.35	16.60	-0.35	10.932	-0.35	16.54	-0.35	10.80
8	-5.0	244.00	-5.0	158.40	-4.7	243.1	-4.7	158.3

Table 3. Earth-Sun Distance (d)

Earth-Sun Distance in Astronomical Units									
Julian Day	Distance	Julian Day	Distance	Julian Day	Distance	Julian Day	Distance	Julian Day	Distance
1	.9832	74	.9945	152	1.0140	227	1.0128	305	.9925
15	.9836	91	.9993	166	1.0158	242	1.0092	319	.9892
32	.9853	106	1.0033	182	1.0167	258	1.0057	335	.9860
46	.9878	121	1.0076	196	1.0165	274	1.0011	349	.9843
60	.9909	135	1.0109	213	1.0149	288	.9972	365	.9833

Table 4. Solar Spectral Irradiance ($E_{SUN\lambda}$)

ETM+ Solar Spectral Irradiances	
Band	watts/(meter squared * µm)
1	1969.000
2	1840.000
3	1551.000
4	1044.000
5	225.700
7	82.07
8	1368.000

Table 5. Solar Spectral Irradiance ($E_{SUN\lambda}$)

ETM+ Thermal Band Calibration Constants		
	Constant 1- **K1** watts/(meter squared * ster * μm)	Constant 2 - **K2** Kelvin
Landsat 7	666.09	1282.71

2.3. Pass-One

The Landsat 7 ETM+ ACCA algorithm recognizes clouds by analyzing the scene twice. Pass-One processing is intended to isolate clouds from non-clouds. In order to achieve this, eight filters are utilized for this purpose, see Table 6. For more details and discussion of band threshold selection, refer to [2-4].

The goal of Pass-One is to develop a reliable cloud signature for use in Pass-Two where the remaining clouds are identified. Omission errors, however, are expected. These errors 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. [22, 23] 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 7.

2.4. Pass-Two

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.

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

Band 6 statistical moments (mean, standard deviation, distribution skewness, kurtosis), see equation (4), 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.

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 300K

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

Table 6. 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

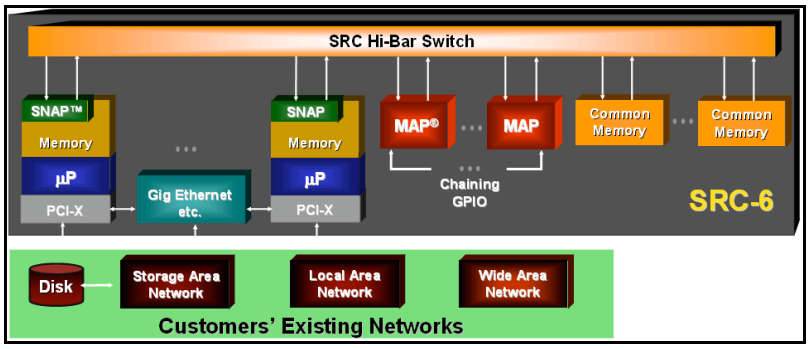
Table 7. Generalized Classification Rules for Pass-One [22]

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

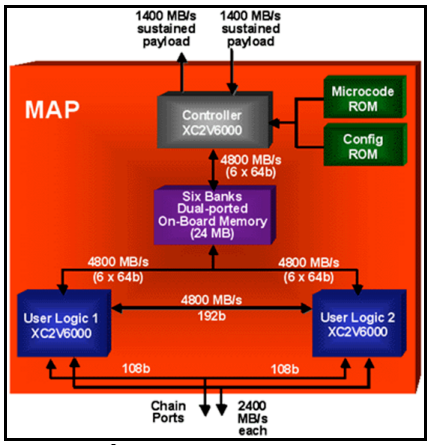
Notes:

[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.



a) System Architecture



b) MAP® Reconfigurable Processor

Figure 1. Hardware Architecture of SRC-6

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.

3. SRC-6 Reconfigurable Computer

3.1. Hardware Architecture

SRC-6 platform consists of one or more general-purpose microprocessor subsystems and one or more MAP® reconfigurable processor subsystems. These subsystems are interconnected through a Hi-Bar Switch communication layer and Global Common Memory (GCM) nodes of shared memory space; see Fig. 1(a). Each microprocessor board is based on a 3GHz Pentium 4 microprocessors. Microprocessors

boards are connected to the MAP boards through the SNAP® interconnect. The SNAP card plugs into the DIMM (Dual In-line Memory Module) slot on the microprocessor motherboard [17, 18] to provide higher data transfer rates between the boards than the inefficient but common PCI (Peripheral Component Interconnect) solution. The peak transfer rate between a microprocessor board and the MAP board is 1600 MB/sec.

Hardware architecture of the SRC-6 MAP processor is shown in Fig. 1(b). The MAP board is composed of 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.

3.2. Programming Model

The SRC-6 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.

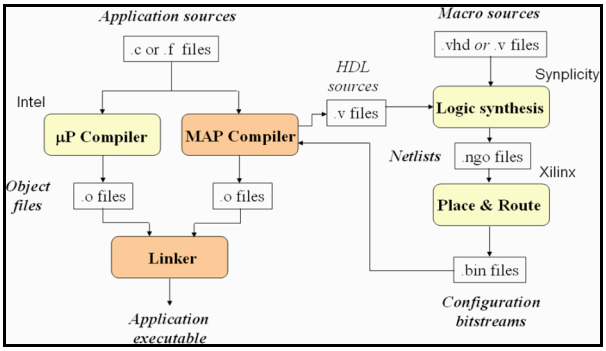


Figure 2. SRC Compilation Process

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.

4. ACCA Hardware Architecture

The ACCA algorithm has been implemented targeting both conventional microprocessor (µP) platforms and reconfigurable computing (RC) platforms. The µP implementation has been performed using a C++ and Matlab programs whose results have been used as a reference against which the RC results are evaluated for both accuracy and speed/performance. The RC implementations have been performed using two designs, namely full-precision fixed-point arithmetic and single-precision floating-point arithmetic.

4.1 Top-Hierarchical Architecture

Fig. 3 shows the main functional/architectural units of the ACCA algorithm. As previously described, the ACCA algorithm handles the cloud population in each scene uniquely by examining the image data twice after a normalization step being performed on the raw data to compensate for temporal data characteristics. The first pass captures clouds using eight different filters. The goal of Pass-One is to develop a reliable cloud signature for Pass-Two. Pass-Two resolves the detection ambiguity resulted from Pass-One where thermal properties of clouds identified during Pass-One are characterized and used to identify remaining cloud pixels.

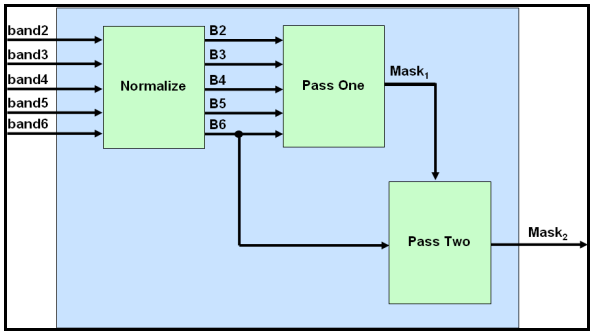


Figure 3. Top-Level Architecture of the ACCA Algorithm

4.2 Normalization Module

ETM+ bands 2-5 are reflectance bands, while band 6 is a thermal band. The reflectance bands are normalized to correct for illumination (solar zenith) angle yielding an estimated reflectance value ρ. The thermal band is calibrated to an equivalent blackbody Brightness Temperature (BT). As can be seen from equations (1-3) in section 2.2, this normalization for the reflectance bands is a linear operation while it is non-linear for the thermal band. In the on-board processing system, these operations are performed by the calibration stage [22]. Therefore, equations (1-3) can be rewritten as:

$$\left.\begin{aligned} \rho_i &= \beta_i \times band_i + \alpha_i \ , \ i = 2,3,4,5,6 \\ B_i &= \rho_i \ , \ i = 2,3,4,5 \end{aligned}\right\} \quad (5)$$

$$B_6 = \frac{K_2}{\ln\left(\frac{K_1}{\rho_6} + 1\right)} \quad (6)$$

The formulation of the normalization function as expressed in equations 5 and 6 can be easily used to derive the corresponding architecture shown in Fig. 4(a).

Due to the high cost in terms of hardware resources required to implement the non-linear normalization function for the thermal band, we will use a piecewise-linear approximation.

Performing algebraic manipulation, equation (6) can be rewritten as:

$$B_6 = \frac{K_2}{\ln(K_1) - \ln(\rho_6) + \ln\left(1 + \frac{\rho_6}{K_1}\right)} \quad (7)$$

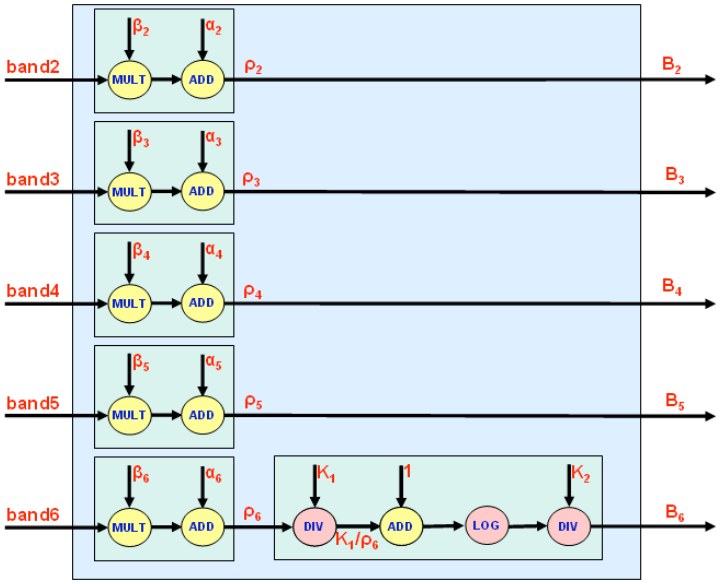
Using Taylor's series first order approximation, the logarithmic function as well as the inversion function can be approximated as follows:

$$when\ |x| < 1 \ \Rightarrow\ \ln(1+x) \cong x\ ,\ \frac{1}{1-x} \cong 1 + x \quad (8)$$

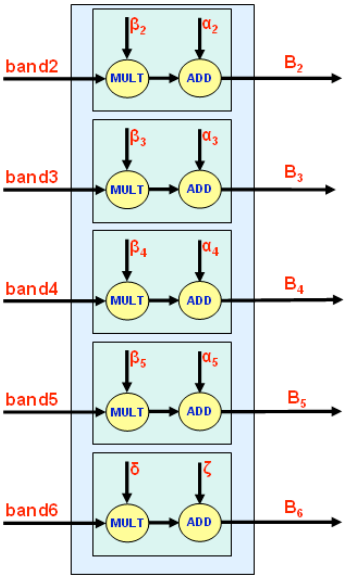
The value of reflectance, ρ, is, by definition, a positive rational number less than unity. As shown in Table 5, the thermal calibration constant K_1 is a positive number much larger than unity. Therefore, the following expression can be derived:

$$\left.\begin{array}{l} \because (0 < \rho_6 < 1)\ and\ (K_1 >> 1) \\ \Rightarrow 0 < \frac{\rho_6}{K_1} < 1 \Rightarrow \ln\left(1 + \frac{\rho_6}{K_1}\right) \cong \frac{\rho_6}{K_1} \end{array}\right\} \quad (9)$$

Applying the results of equations (8) and (9), equation (7) can be approximated as follows:



a) Exact Normalization Operations



b) Approximated Normalization Operations

Figure 4. Normalization Module Architecture

$$B_6 = \frac{K_2}{\ln(K_1) - \ln(1 + (\rho_6 - 1)) + \ln\left(1 + \frac{\rho_6}{K_1}\right)}$$

$$B_6 \cong \frac{K_2}{\ln(K_1) - (\rho_6 - 1) + \frac{\rho_6}{K_1}}$$

$$B_6 \cong \frac{K_2}{1 + \ln(K_1)} \cdot \frac{1}{1 - \rho_6 \cdot \frac{\left(1 - \frac{1}{K_1}\right)}{1 + \ln(K_1)}} \qquad (10)$$

As previously noted about the value of reflectance, ρ, and the thermal calibration constant, K_1, the following expression can be derived:

$$\left.\begin{array}{l} \because (0 < \rho_6 < 1) \; and \; (K_1 >> 1) \\ \Rightarrow 0 < \rho_6 \cdot \frac{\left(1 - \frac{1}{K_1}\right)}{1 + \ln(K_1)} < 1 \end{array}\right\} \qquad (11)$$

Applying equation (8) to equation (11), equation (10) can be rewritten as:

$$B_6 \cong \frac{K_2}{1 + \ln(K_1)} + \frac{K_2 \cdot \left(1 - \frac{1}{K_1}\right)}{(1 + \ln(K_1))^2} \cdot \rho_6 \qquad (12)$$

Substituting expression (5) for ρ_6 in equation (12) the following linear expression approximates expression (6) for the thermal band:

$$\left.\begin{array}{l} \rho_6 = \beta_6 \times band_6 + \alpha_6 \\ B_6 \cong \delta \times band_6 + \zeta \end{array}\right\}, \; where \qquad (13)$$

$$\delta = \left(\frac{K_2 \left(1 - \frac{1}{K_1}\right) \cdot \beta_6}{(1 + \ln(K_1))^2} \right), \; and$$

$$\zeta = \left(\frac{K_2}{1 + \ln(K_1)} + \frac{K_2 \left(1 - \frac{1}{K_1}\right) \cdot \alpha_6}{(1 + \ln(K_1))^2} \right)$$

Therefore, the following equations can be used to express the approximated normalization operations:

$$\left.\begin{array}{l} B_i = \beta_i \times band_i + \alpha_i \;, \; i = 2,3,4,5 \\ B_6 \cong \delta \times band_6 + \zeta \end{array}\right\} \qquad (14)$$

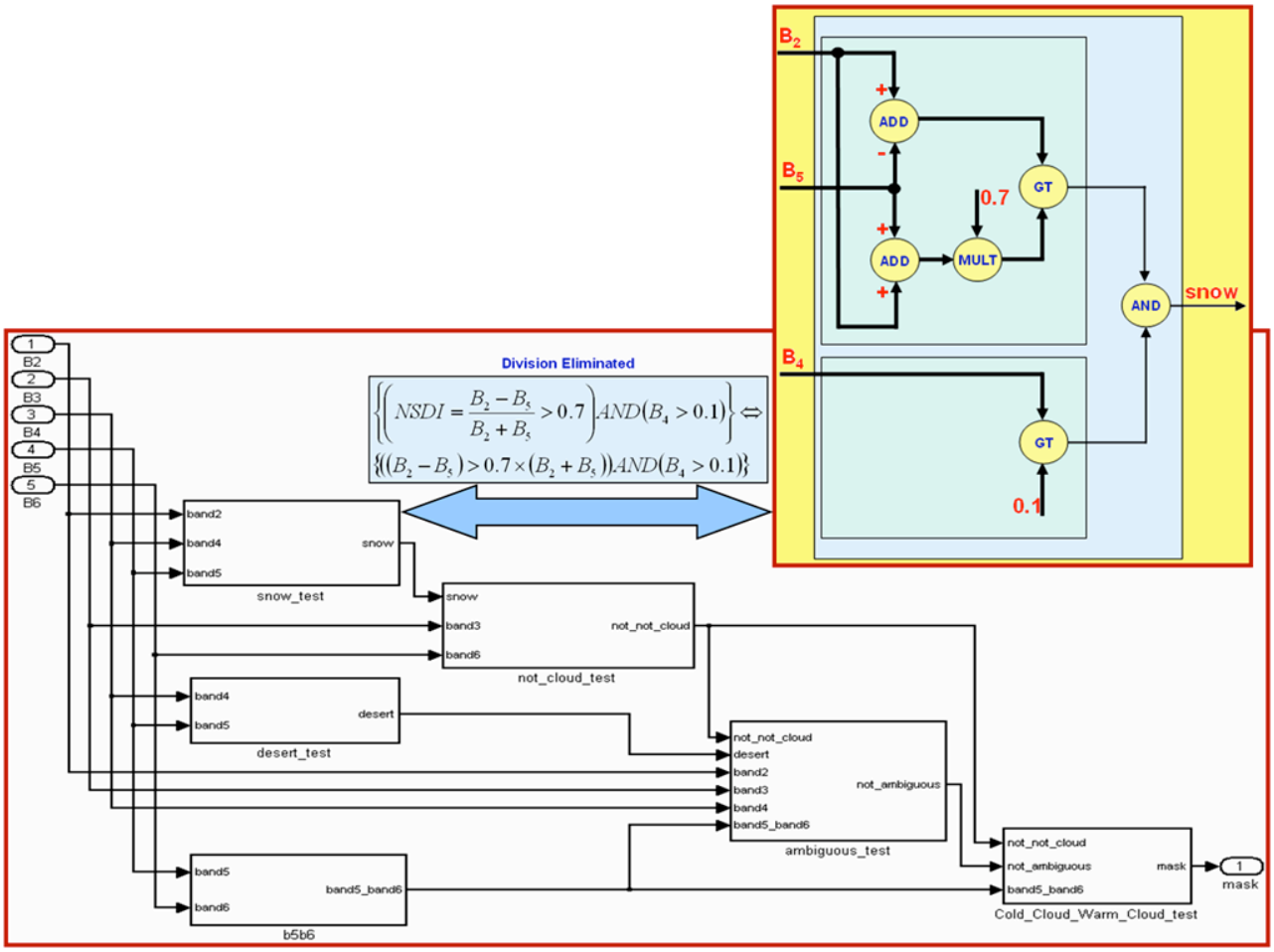


Figure 5. Pass-One Architecture and Snow-Test Algebraic Re-Formulation

The coefficients (α_1-α_6) and (β_1-β_6) are computed from equations (1) and (2), while ζ and δ are computed from equation (13). Fig. 4(b) shows the final implementation architecture derived from equation 14.

4.3 Pass-One Module

The first pass of the ACCA algorithm is a cascading set of eight threshold-based filters, see Table 6. These filters are designed to classify each pixel into one of four classes, ColdCloud, WarmCloud, NotCloud, and Ambiguous, as shown in Table 7. Pixels labeled ambiguous are reprocessed in the second pass as previously discussed. 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. This algebraic re-formulation was applied to most of the filters involved for Pass-One. Fig. 5 shows the equivalent hardware architecture of Pass-One as well as an example of this algebraic re-formulation applied to the snow test.

5. Experimental Work

5.1. Experimental Approaches

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-6.

The goal of the implementation was to achieve a comparable performance/speed and detection accuracy to what has been reported in [21] and [22]. Therefore, the constraints to the design were the processing speed, as measured by throughput, and the hardware resources required for the design. The first constraint is approached through full-pipelining and superscaling of the design. The second constraint was approached through approximating the non-linear normalization step as mentioned in section 4.2. Moreover, because many of the tests in Pass-One are threshold tests of ratio values such as the snow test, see Tables 6 and 7, 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.

The criterion that defines the detection accuracy is based on the absolute error given by equation (15), between the detected cloud mask and a reference mask produced by the software, C++/Matlab, version of the ACCA algorithm.

$$error = \frac{\sum_{i=0}^{(rows-1)} \sum_{j=0}^{(columns-1)} \left| x_{ij} - y_{ij} \right|}{rows \times columns} \quad , \quad where \qquad (15)$$

$$x \equiv output\ \ image \ ,$$

$$y \equiv reference\ \ image$$

In addition, the goal of achieving high detection accuracy has been approached by minimizing the quantization errors through full-precision fixed-point as well as single-precision floating-point arithmetic hardware implementations. Furthermore, saturated arithmetic has also been utilized in order to avoid overflow errors.

5.2 Experimental Results

The fixed-point version of the design was developed in VHDL, synthesized, placed and routed, and was found to occupy approximately a maximum of 14% of the available logic resources (slices) on the chip, see Table 8. Because the utilization of the block RAM, i.e. 7% RAMB16, and the embedded hardware multipliers, i.e. 2% MULT 18X18, typically changes linearly with the number of unit instantiations inside of the chip, these types of resources are factors in deciding how many concurrent processing engines to instantiate in the same chip. The bandwidth of the FPGA local on-board memory (OBM) is another important factor. This enabled the instantiation of eight concurrent processing engines, see equation (16), of the design in the same chip, which increased the performance to eight folds. The total resources utilization for the eight engine version was approximately 64%, which leaves plenty of room for more processing functions to be implemented on the same chip.

$$N_{engines}^{\max} = Min\left(\frac{100}{Max(u_1, u_2, ..., u_n)}, \frac{W_{OBM}}{W_{pixel}} \right) \quad , \quad where \qquad (16)$$

$$N_{engines}^{\max} \equiv Maximum\ \ number\ \ of\ \ engines,$$

$$u_i \equiv Utilization\ \ of\ \ type-i\ \ resource\ \ (\%)$$

$$n \equiv Number\ \ of\ \ resources\ \ whose\ \ utilization\ \ changes\ \ linearly\ \ with\ \ number\ \ of\ \ engines$$

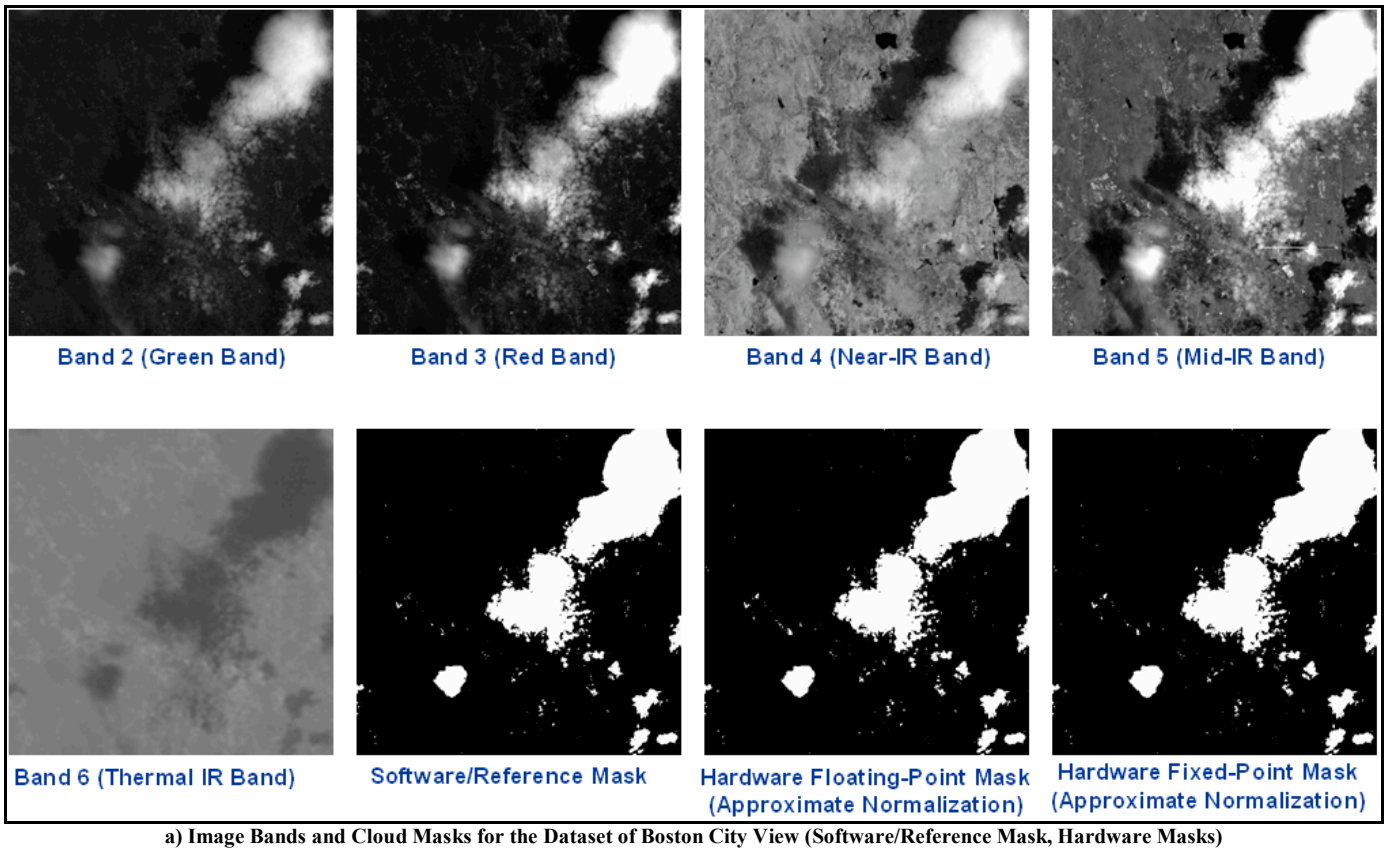
$$W_{OBM} \equiv OBM\ \ word\ \ length\ \ (bits)$$

$$W_{pixel} \equiv Pixel\ \ width\ \ (bits)$$

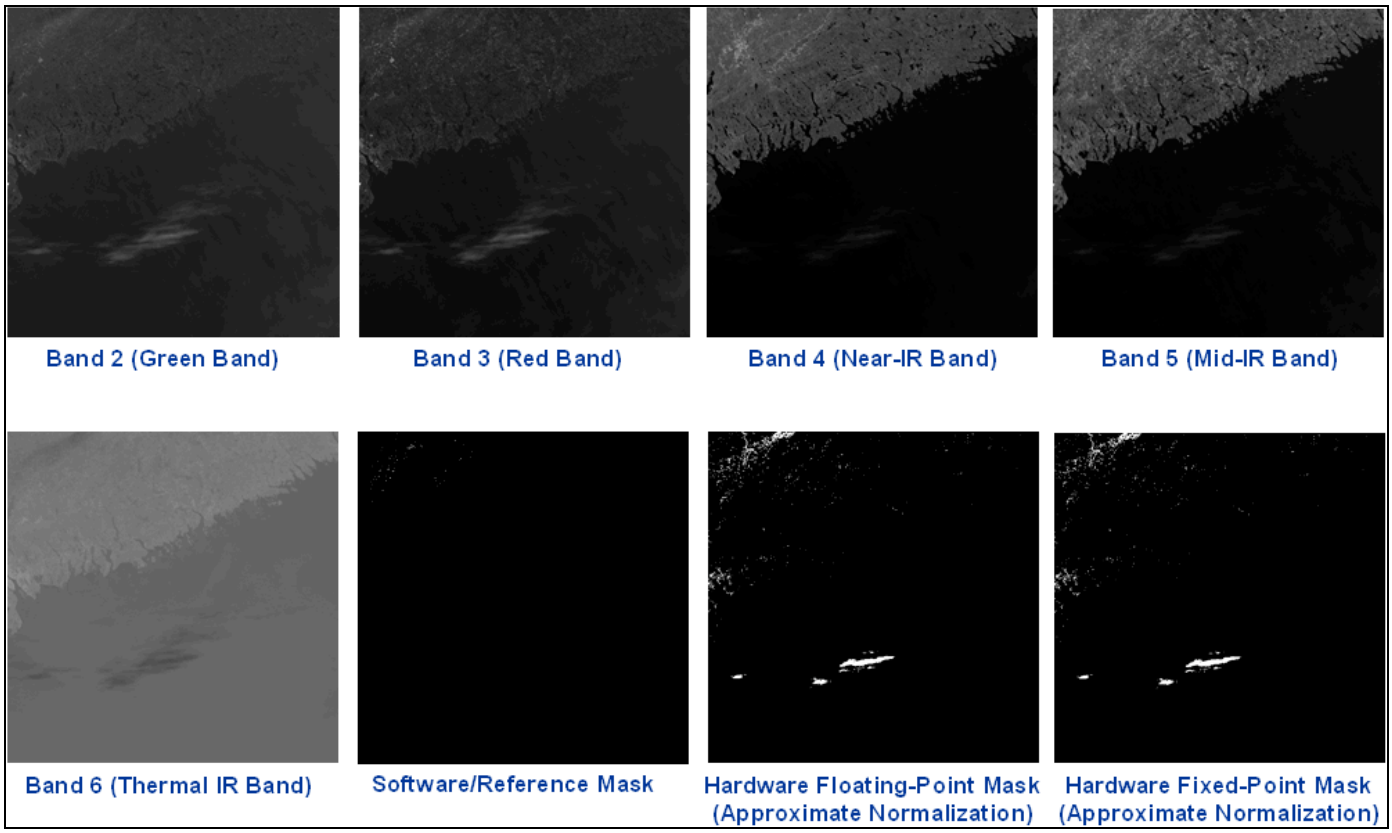
$$\Rightarrow N_{engines}^{\max} = Min\left(\frac{100\%}{Max(2\%, 7\%)}, \frac{64bits}{8bits} \right) = 8\ \ engines$$

Table 8. Hardware Resources Usage

Resource Type	ACCA on SRC-6 (Virtex-II XC2v6000ff1517-4, 100MHz)			
	23-bit Fixed-Point (25 Clock Cycle Latency)		Single-Precision Floating-Point (78 Clock Cycle Latency)	
	1 Engine	8 Engines	1 Engine	2 Engines
Slices	4,820 (14%)	21,694 (64%)	17,565 (51%)	29,660 (87%)
LUTs	3,260 (4%)	13,726 (20%)	20,885 (30%)	39,791 (58%)
Slice Flip Flops	8,384 (12%)	37,826 (55%)	23,005 (34%)	42,526 (62%)
MULT 18X18	4 (2%)	32 (22%)	36 (25%)	72 (50%)
RAMB16	11 (7%)	88 (61%)	0 (0%)	0 (0%)



a) Image Bands and Cloud Masks for the Dataset of Boston City View (Software/Reference Mask, Hardware Masks)



b) Image Bands and Cloud Masks for the Dataset of USA East Coast View (Software/Reference Mask, Hardware Masks)

c) Approximate Normalization and Quantization Errors for the Dataset of Boston City View

Figure 6. Detection Accuracy (Based on the Absolute Error)

Table 9. Approximation and Quantization Errors Based on the Absolute Error, see equation (15)

Dataset	Percentage Error (%)			
	Before July 1, 2000		After July 1, 2000	
	Low Gain	High Gain	Low Gain	High Gain
Boston City	00.404	00.136	00.368	**00.1028**
East Coast of USA	03.4767	**00.9102**	03.1567	00.7036

The floating-point version of the design including a partial implementation of Pass Two was developed in SRC MAP-C, and was found to occupy approximately 51% of the available logic resources (slices) on the chip, see Table 8. Only two concurrent processing engines of the design were instantiated in the same chip. This increased the performance to two folds. The total resources utilization for the two engine version was approximately 87%.

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).

Figs. 6(a) and 6(b) show the image bands for views of the city of Boston and for the east coast of the USA. These views were captured by Landsat 7 ETM+. Figs. 6(a) and 6(b) also show the reference mask produced by the software, C++/Matlab, version of the ACCA algorithm as well as both hardware masks, i.e. fixed-point and floating-point. The results were obtained from a 2.8GHz Intel Xeon processor and from SRC-6.

As shown in Fig. 6(c), the linearization of the normalization step of the algorithm has introduced an error equal to 0.1028% for the dataset of Boston City view. The hardware floating-point implementation has shown identical behavior to the software version of the algorithm. Fig. 6(c)

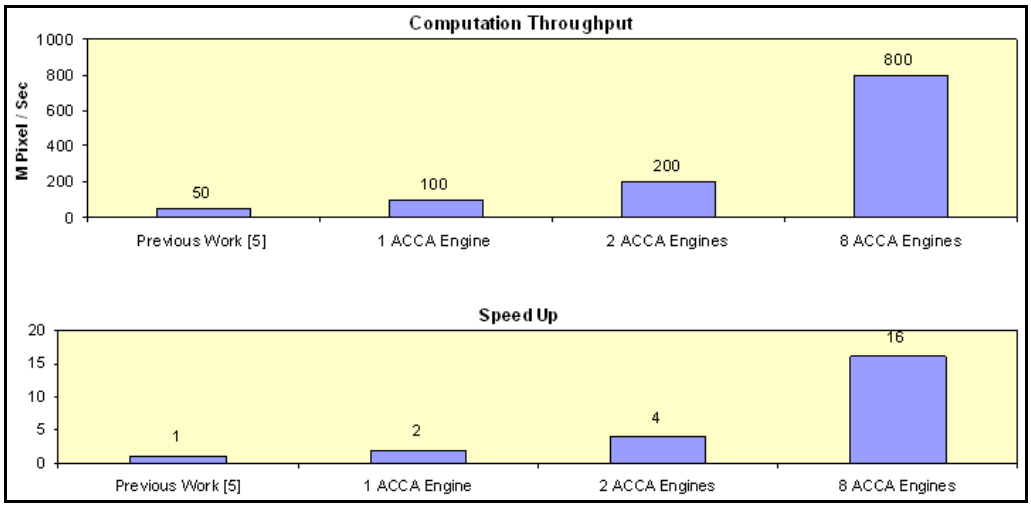


Figure 7. Hardware-to-Hardware Performance

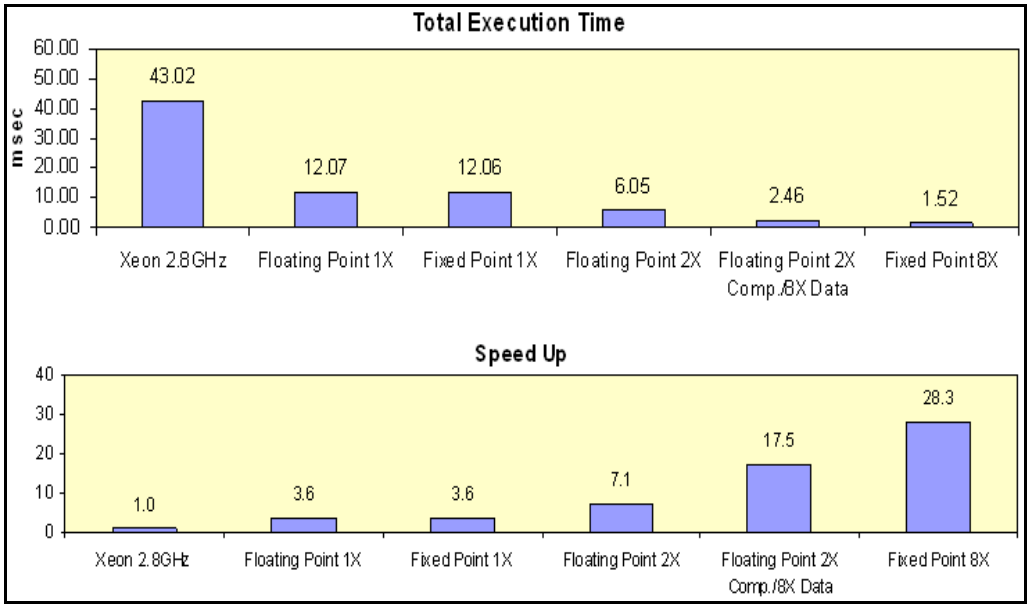


Figure 8. Hardware-to-Software Performance

also shows that the hardware full-precision (23-bit) fixed-point version has improved the error due to quantization effects from 0.2676% to 0.1028% which made it identical to the software/reference version. The latest reported error by Williams et al. [22] is approximately 1.02% which makes the achieved results of this work 9.9x, at best, more accurate for the best case of Boston City view, see Table 9. The results shown in Table 9 were obtained based on the date on which the datasets were acquired as well as on the calibration parameters shown in Table 2. It is worth to mention that the error 0.1028% is due to the approximated normalization step which has not been implemented by Williams [22]. This adds another privilege to this work against previous similar reports. Had we not considered the normalization step, the results of this investigation in terms of the detection accuracy as compared to previous similar works could be considered ideal.

In terms of performance/throughput, the results have also been compared to previous work by Williams et al. [22]. Fig. 7 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). The results show a speedup of up to 16 times compared to those reported in [22], see Fig. 7.

The hardware implementations provided a higher performance (28 times faster) compared to the 2.8GHz Xeon implementation, see Fig. 8. The superiority of RCs over traditional platforms for cloud detection is demonstrated through the performance plots shown in Fig. 8.

6. Summary and Observations

This paper presented the design and implementation of an RC-based real-time cloud detection system. 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-6, has been investigated.

This work was shown 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 implementation 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.

7. Conclusions and Future Work

The contributions of this work, as compared to previous work, can be concluded to achieve an implementation completeness of the ACCA algorithm by prototyping both the normalization step and Pass-One targeting hardware platforms as well as Pass-Two targeting software (Pass-Two has not been fully implemented for hardware due to time constraints). In addition, this implementation is generic and has been parameterized to support both Landsat and MODIS datasets. This work is characterized by higher performance/throughput as well as higher detection accuracy as compared to previous reported works, [22]. Using the SRC-6 reconfigurable architecture, a 16x speedup over previous hardware implementations, and a 28x speedup over microprocessor, Xeon 2.8GHz, implementations is achieved. Furthermore, a 9.9x higher detection accuracy compared to previous hardware implementations has been verified. This high accuracy has been achieved by minimizing the quantization errors as well as the overflow errors. The quantization errors have been minimized through the floating-point and full-precision fixed-point implementations. The overflow errors have been reduced through the adoption of saturated arithmetic.

Although the ACCA algorithm has been fully verified in the software version, this work can be further extended by considering full hardware implementations of Pass-Two. In addition, power consumption issues can also be considered for power-efficient prototypes which can be critical for on-board processing. In parallel to these directions, other on-board pre-processing functionalities, e.g. image registration, should be tackled in order to accomplish a complete set of efficient hardware and/or software solutions covering all the needs for on-board remote sensing.

References

[1] Decadal Survey, “Earth Science and Applications from Space: National Imperatives for the Next Decade and Beyond”, http://www.nap.edu/catalog/11820.html.

[2] 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.

[3] R.R. Irish, J.L. Barker, S.N. Goward, and T.J. Arvidson, “Characterization of the Landsat-7 ETM+ automated cloud-cover assessment (ACCA) algorithm”, Photogrammetric Engineering & Remote Sensing, vol. 72, no. 10, pp. 1179–1188, 2006.

[4] Jih-Jui Su, “Enhanced ACCA Algorithm”, Space Imaging Corporation Technical Memo IT81-LSD-SA&E Memo 274, 1984.

[5] J. T. Johnson, Rad Hard FPGAs, Available online: http://esl.eng.ohio-state.edu/ rstheory/iip/RadHardFPGA.doc.

[6] T. V. Court, and M. C. Herbordt, "Families of FPGA-Based Accelerators for Approximate String Matching", ACM Microprocessors & Microsystems, v. 31, Issue 2, March 2007, pp. 135-145.

[7] V. Kindratenko, and D. Pointer, “A case study in porting a production scientific supercomputing application to a reconfigurable computer”, in Proc. IEEE Symposium on Field-Programmable Custom Computing Machines - FCCM'06, 2006. pp. 13-22.

[8] V. Aggarwal, A. D. George, K. C. Slatton, "Reconfigurable Computing with Multiscale Data Fusion for Remote Sensing", Proceedings of the 2006 ACM/SIGDA 14th International Symposium on Field Programmable Gate Arrays (FPGA 2006), Monterey, California, USA.

[9] E. El-Araby, M. Taher, T. El-Ghazawi, and J. Le Moigne, “Prototyping Automatic Cloud Cover Assessment (ACCA) Algorithm for Remote Sensing On-Board Processing on a Reconfigurable Computer”, IEEE International Conference On Field-Programmable Technology (FPT 2005), Singapore, December, 2005.

[10] J. Harkins, T. El-Ghazawi, E. El-Araby, and M. Huang, “Performance of Sorting Algorithms on the SRC 6 Reconfigurable Computer”, IEEE International Conference On Field-Programmable Technology (FPT 2005), Singapore, December, 2005.

[11] 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.

[12] D. A. Buell, J. P. Davis, G. Quan, S. Akella, S. Devarkal, P. Kancharla, E. A. Michalski, and H. A. Wake, "Experiences with a reconfigurable computer," Proceedings, Engineering of Reconfigurable Systems and Algorithms, Las Vegas, Nevada, 21-24 June 2004.

[13] D. A. Buell and R. Sandhu, "Identity management," IEEE Internet Computing, v. 7, no. 6, November/December 2003, pp. 26-28 (guest editors' introduction).

[14] A. Michalski, K. Gaj, T. El-Ghazawi, “An Implementation Comparison of an IDEA Encryption Cryptosystem on Two General-Purpose Reconfigurable Computers”, Proc. FPL 2003, Lisbon, Sept. 2003, pp. 204-219.

[15] O. Storaasli, “Scientific Applications on a NASA Reconfigurable Hypercomputer”, 5th MAPLD International Conference, Washington, DC, USA, September, 2002.

[16] SRC Computers, Inc., http://www.srccomputers.com/products/embeddedportable.asp

[17] “SRC-6 C-Programming Environment Guide”, SRC Computers, Inc. 2005.

[18] http://www.srccomputers.com/products/docs/SRC-6%20SystemsServers_69223-AD.pdf

[19] R. D. Burns, B. J. Naasz, S. Z. Queen, J. Van Eepoel, J. Hannah, and E. Skeltonk, “The HST SM4 Relative Navigation Sensor System: Overview and Preliminary Testing Results from The Flight Robotics Lab”, 18th AAS/AIAA Space Flight Mechanics Meeting, Galveston, Texas, USA, January 27-31, 2008.

[20] http://gsfctechnology.gsfc.nasa.gov/SpaceCube.htm

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

[22] 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.
[23] 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.
[24] http://ltpwww.gsfc.nasa.gov/IAS/handbook/handbook_htmls/chapter11/chapter11.html

Esam El-Araby (M'05) received his B.Sc. degree in Electronics and Telecommunication Engineering from Assiut University, Egypt, in 1991, his Higher Diploma degree in Automatic Control and Computer Engineering from Assiut University, Egypt, in 1997, and his M.Sc. in Computer Engineering from the George Washington University (GWU), USA, in 2005. He is currently pursuing his Ph.D. degree in Computer Engineering at the George Washington University (GWU), USA. His research interests include reconfigurable computing, hybrid architectures, evolvable hardware, performance evaluation, digital signal/image processing, and hyperspectral sensing.

Tarek El-Ghazawi is a Professor in the Department of Electrical and Computer Engineering at The George Washington University, where he also directs the High Performance Computing Laboratory (HPCL). He is a fellow of the Arctic Region Supercomputing Center and a Visiting Scientist at NASA GSFC. He has received his Ph.D. degree in Electrical and Computer Engineering from New Mexico State University in 1988. His research interests include high-performance computing and architectures, reconfigurable computing, parallel I/O, and performance evaluations. He has published over 100 refereed research papers and book chapters in these areas and his research has been supported by DoD/DARPA, NASA, NSF, and industry including IBM and SGI. He is a senior member of the Institute of Electrical and Electronics Engineers (IEEE), a member of the Association for Computing Machinery (ACM), Phi Kappa Phi National Honor Society.

Jacqueline Le Moigne is Assistant Chief for Technology in the Software Engineering Division at NASA Goddard Space Flight Center as well as a NASA Goddard Senior Fellow. She received a B.S and a M.S. in Mathematics and a Ph.D. in Computer Science (specialty: Computer Vision) from the University Pierre and Marie Curie, Paris, France. As an Assistant Research Scientist at the Computer Vision Laboratory of the University of Maryland, Dr. Le Moigne designed new algorithms and supervised the development of a visual navigation system for the Autonomous Land Vehicle (ALV) project. Jacqueline Le Moigne came to Goddard in 1990 as a National Research Council Senior Research Associate, she then became a Senior Scientist at the Center of Excellence in Space Data and Information Sciences (CESDIS), a Senior Computer Scientist in the Applied Information Sciences Branch of the Earth and Space Data and Computing Division, and more recently Assistant Chief for Technology in the Software Engineering Division. During that time, she has focused her research interests on applying Computer Vision to Earth and Space Science problems such as robotics, land use/land cover assessment, and intelligent data management, and on utilizing high performance parallel computers. Some of her most recent research focuses on Parallel Registration of Multi-Sensor/Multi-Scale Satellite Image Data, for which she has been studying wavelets and their implementation on high performance computers. Other recent work includes the development of an image registration toolbox, the registration of Landsat and EOS Core Sites imagery, the implementation of image processing techniques on reconfigurable computers for application to on-board processing and sensor webs, as well as to in-situ processing for planetary vision systems. Dr. Le Moigne has published over 100 papers. She was appointed NASA Goddard Senior Fellow in 2005, elected IEEE Senior member in 1996 and was Associate Editor of the IEEE Transactions on Geoscience and Remote Sensing from 2001 to 2005. She was also an Associate Editor for Pattern Recognition from 2001 to 2003.

Richard Irish is a Senior Systems Engineer at the Biospheric Science Branch, Goddard Space Flight Center. As he worked on NASA's Landsat 7 program, he was involved with cross calibrating ETM+ with the TM and ALI sensors. Mr. Irish developed the cloud cover recognition algorithm used for Landsat 7 and is working to extend the algorithm's capability. He is also the Landsat 7 science liaison to the user community. He wrote and maintains the online Landsat 7 Science Data Users Handbook. Mr. Irish led the working group that defined the standard Landsat 7 distribution product and authored the associated data format control book. He also prepared the level-one requirements for Landsat's LPGS and IAS. Previously, Mr. Irish worked as the Remote Sensing Applications Manager at Hughes STX Remote Satellite Mapping Technologies Corporation. He was involved in promoting the growth of the company's commercial remote sensing business. There he developed new products, prepared proposals, managed value-added projects, authored trade journal articles, and applied company technologies to marketplace opportunities. Mr. Irish also initiated the company's direct marketing campaign.