# Performance Scalability of a Remote Sensing Application on High Performance Reconfigurable Platforms

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

## Abstract

*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). Multi-node systems of High-Performance RCs (HPRCs) are becoming more popular recently as dictated by the requirements of many large-scale applications. Therefore, HPRCs are a promising candidate for on-board preprocessing.*

*However, in order to meet the aggressive demands of those large-scale applications such as remote sensing, scalability of such applications on HPRCs may become an issue that can negatively affect the overall performance. In this paper, we investigate the potential of using HPRCs for on-board preprocessing by studying and characterizing the scalability of the Landsat 7 ETM+ ACCA algorithm on two of the state-of-the-art reconfigurable platforms, SRC-6 and Cray-XD1.*

## 1. Introduction

Remote sensing satellite missions have always been characterized by the demand of smaller size, lower cost, and more flexibility. On-board processing, as a solution, permits a good utilization of expensive resources. 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 onboard 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.

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 [1], image processing [2], and remote sensing applications [2], [3]. Multi-node systems of High-Performance RCs (HPRCs) are becoming more popular recently as dictated by the requirements of many large-scale applications. These systems are characterized by lower form/wrap factors compared to parallel platforms, and higher flexibility than ASIC solutions. Therefore, HPRCs are a promising candidate for on-board preprocessing. The SRC-6 Reconfigurable Computer and Cray-XD1 are examples of this category of computers [4], [5], [6] and are used here for this purpose.

However, in order to meet the aggressive demands of those large-scale applications such as remote sensing, scalability of such applications on HPRCs may become an issue that can negatively affect the overall performance. In this paper, we investigate the potential of using HPRCs for on-board preprocessing by studying and characterizing the scalability of the Landsat 7 ETM+ ACCA algorithm [7] on two of the state-of-the-art reconfigurable platforms, SRC-6 and Cray-XD1.

## 2. Systems and Application

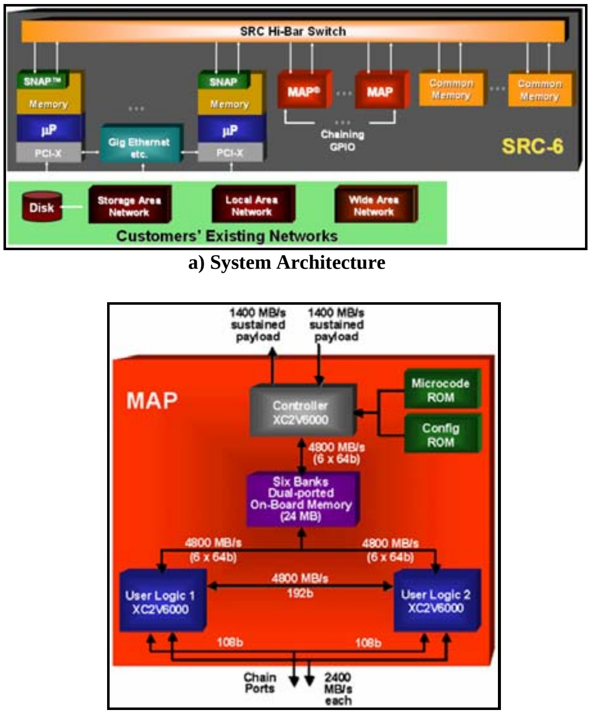
In this work, two main systems have been used. These are the SRC-6 and the Cray-XD1 at The George Washington University, High-Performance Computing Laboratory (HPCL).

Landsat 7 ETM+ ACCA algorithm was selected to determine an almost practical bounds on the potential performance of HPRCs should all architectural challenges be alleviated and to gain an insight into the system level programmability and performance issues apart from those known for general high-performance computers.

### 2.1. SRC-6 Reconfigurable Computer

SRC-6 platform consists of one or more general-purpose microprocessor subsystems and one or more MAP® reconfigurable processor subsystems [4]. These subsystems are interconnected through a Hi-Bar Switch communication layer and Global Common Memory (GCM) nodes of shared memory space; see Fig. 3(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 slot on the microprocessor motherboard 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.

Hardware architecture of the SRC-6 MAP processor is shown in Fig. 3(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.



a) System Architecture

b) MAP® Reconfigurable Processor

Figure 1. Hardware Architecture of SRC-6

## 2.2. Cray XD1 Reconfigurable Computer

The Cray XD1 system is based on the Direct Connected Processor (DCP) architecture, harnessing many processors into a single, unified system [5], [6]. Cray's implementation of the DCP architecture optimizes message-passing applications by directly linking processors to each other through a high performance interconnect fabric, eliminating shared memory contention and PCI bus bottlenecks. The Cray XD1 base unit is a chassis. Up to 12 chassis can be installed in a cabinet.

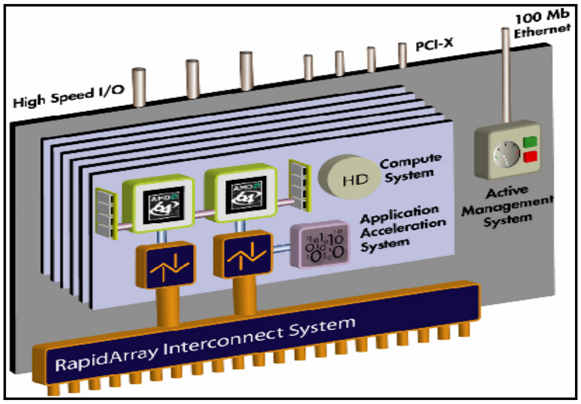


Figure 2. Hardware Architecture of Cray XD1

The Cray XD1 compute subsystem is composed of a Linux operating system and AMD Opteron 64-bit processors. The AMD Opteron processor's integrated memory controller increases memory bandwidth and reduces latencies. Three HyperTransport™ links per processor provide up to 19.2 GB per second I/O bandwidth. Dual Core AMD Opteron processors are available as an option on the Cray XD1 system. Two processing cores exist on a single die, doubling peak performance, without raising power consumption and heat levels.

The Cray XD1 RapidArray interconnect directly connects processors over high-speed, low-latency pathways. Each fully configured chassis includes:

- 6 (or 12) custom communications processors. The communications processors deliver up to 4 GB per second (or 8 GB per second with expansion fabric) bandwidth with 1.7 microsecond MPI latency between nodes.
- A 48 GB per second (optional expansion to 96 GB per second), nonblocking, crossbar switching fabric in each chassis provides two (or four) 2 GB per second links to each node and twelve (or twenty-four) 2 GB per second interchassis links

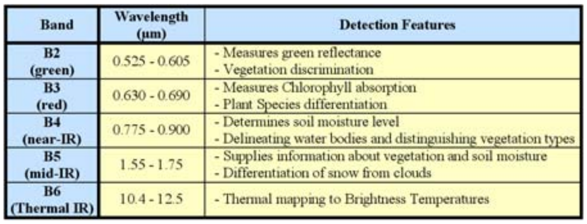
Each Cray XD1 chassis can be configured with six application acceleration processors, based on Xilinx Virtex-IIPro (Virtex-4) FPGAs.

## 2.3. 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 1 presents the bands and their detection features.

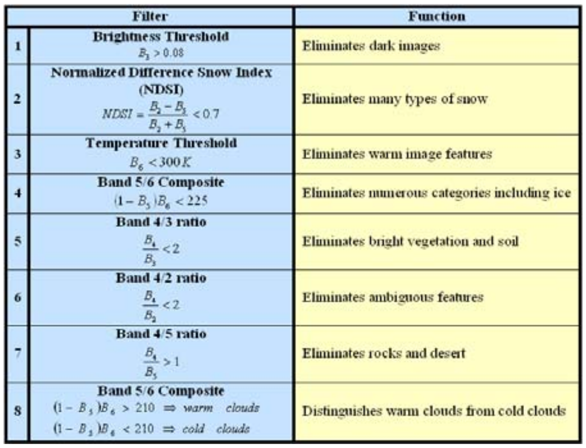
The Landsat 7 ETM+ ACCA algorithm recognizes clouds by analyzing the scene twice. In the first pass eight filters are utilized for this purpose, see Table 2.

**Table 1. Landsat 7 ETM+ Bands**



| Band | Wavelength (µm) | Detection Features |
|---|---|---|
| **B2 (green)** | 0.525 - 0.605 | - Measures green reflectance<br>- Vegetation discrimination |
| **B3 (red)** | 0.630 - 0.690 | - Measures Chlorophyll absorption<br>- Plant Species differentiation |
| **B4 (near-IR)** | 0.775 - 0.900 | - Determines soil moisture level<br>- Delineating water bodies and distinguishing vegetation types |
| **B5 (mid-IR)** | 1.55 - 1.75 | - Supplies information about vegetation and soil moisture<br>- Differentiation of snow from clouds |
| **B6 (Thermal IR)** | 10.4 - 12.5 | - Thermal mapping to Brightness Temperatures |

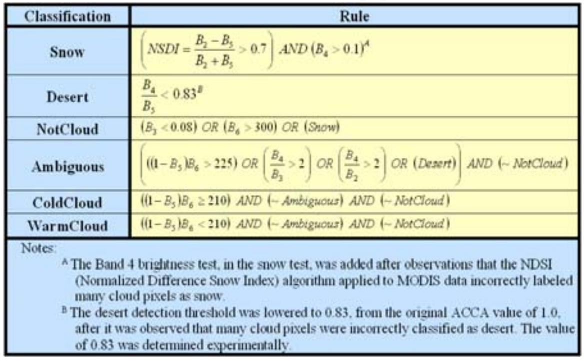
**Table 2. Pass-One Filters**



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

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.

**Table 3. Generalized Classification Rules for Pass-One [8]**



| 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** | $\left(((1 - B_5)B_6 > 225)\ OR \left(\frac{B_4}{B_3} > 2\right) OR \left(\frac{B_4}{B_2} > 2\right) OR\ (Desert)\right) AND\ (\sim NotCloud)$ |
| **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.

John A. Williams et al., [8], [9], 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 3.

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 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.

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

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.

## 2.4. 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 floating-point arithmetic.

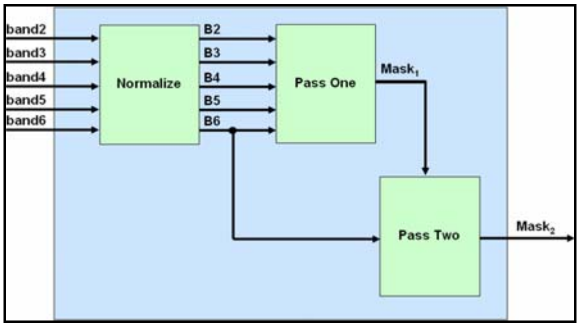


**Figure 3. Top-Level Architecture of the ACCA Algorithm**

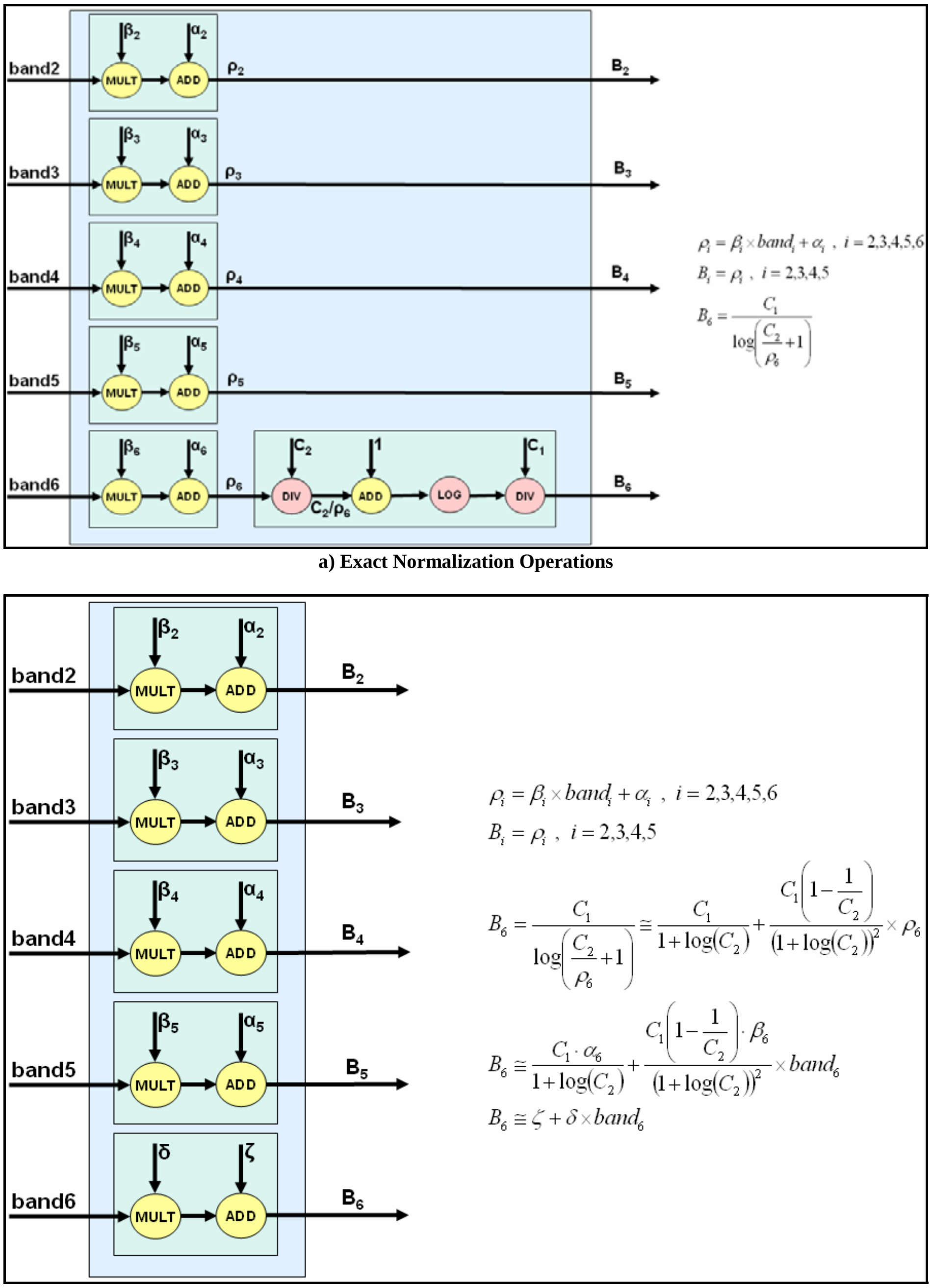
## Top-Hierarchical Architecture

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.

## 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). This normalization for the reflectance bands is a linear operation while it is non-linear for the thermal band, see Fig. 4(a). In the on-board processing system, these operations are performed by the calibration stage [8].

Due to the high cost in terms of hardware resources required, a piecewise-linear approximation is used to


band2
band3
band4
band5
band6
$\beta_2$ $\alpha_2$ $\beta_3$ $\alpha_3$ $\beta_4$ $\alpha_4$ $\beta_5$ $\alpha_5$ $\beta_6$ $\alpha_6$
MULT
ADD
$\rho_2$ $\rho_3$ $\rho_4$ $\rho_5$ $\rho_6$
$C_2$
1
$C_1$
DIV
$C_2/\rho_6$
LOG
$B_2$ $B_3$ $B_4$ $B_5$ $B_6$
$\rho_i = \beta_i \times band_i + \alpha_i \ , \ i = 2,3,4,5,6$
$B_i = \rho_i \ , \ i = 2,3,4,5$
$B_6 = \frac{C_1}{\log\left(\frac{C_2}{\rho_6}+1\right)}$
δ
ζ
$B_6 = \frac{C_1}{\log\left(\frac{C_2}{\rho_6}+1\right)} \cong \frac{C_1}{1+\log(C_2)} + \frac{C_1\left(1-\frac{1}{C_2}\right)}{(1+\log(C_2))^2} \times \rho_6$
$B_6 \cong \frac{C_1 \cdot \alpha_6}{1+\log(C_2)} + \frac{C_1\left(1-\frac{1}{C_2}\right)\cdot\beta_6}{(1+\log(C_2))^2} \times band_6$
$B_6 \cong \zeta + \delta \times band_6$

**a) Exact Normalization Operations**

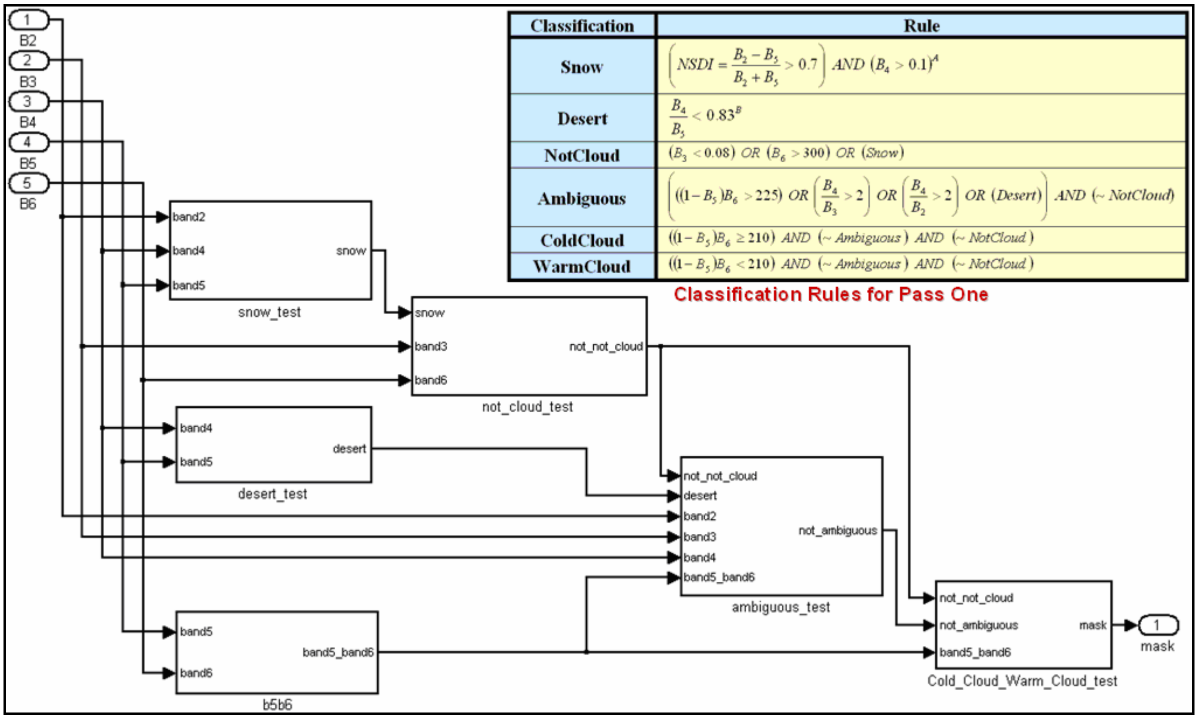
**b) Approximated Normalization Operations**

**Figure 4. Normalization Module Architecture**

implement the non-linear normalization function for the thermal band, see Fig. 4(b).

### Pass-One Module

The first pass of the ACCA algorithm is a cascading set of eight threshold-based filters, see Table 2. These filters are designed to classify each pixel into one of four classes, ColdCloud, WarmCloud, NotCloud, and Ambiguous, as shown in Table 3. 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. Fig. 5 shows the equivalent hardware architecture of Pass-One.



| 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 | $\left(((1 - B_5)B_6 > 225)\ OR \left(\frac{B_4}{B_3} > 2\right) OR \left(\frac{B_4}{B_2} > 2\right) OR\ (Desert)\right)$ 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 $(\sim NotCloud)$ |

Classification Rules for Pass One

**Figure 5. Pass-One Architecture**

## 3. Experimental Work

In a previous work [3], we presented the design and implementation of the ACCA algorithm adapted for Landsat 7 ETM+ data. That work has been proven to provide higher performance and higher detection accuracy than previously reported results. The higher performance was 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, see Fig. 6. On the other hand, the detection accuracy has been verified against software floating-point reference implementation, and the results revealed identical results.

In this work, we investigate the potential of using multi-node HPRCs for on-board preprocessing by studying and characterizing the scalability of the Landsat 7 ETM+ ACCA algorithm on two of the state-of-the-art reconfigurable platforms, SRC-6 and Cray-XD1. The workload was distributed over all nodes using MPI. We scatter the input five bands across all nodes and gather the resulting mask pixels from all nodes at the base node.

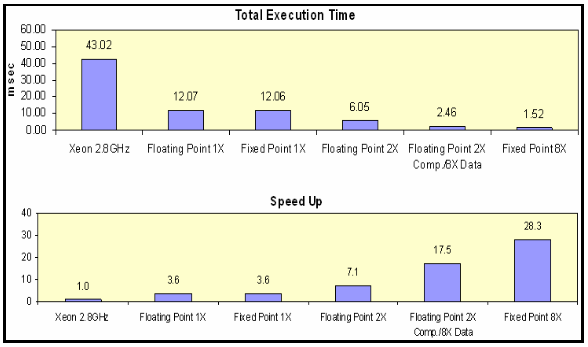
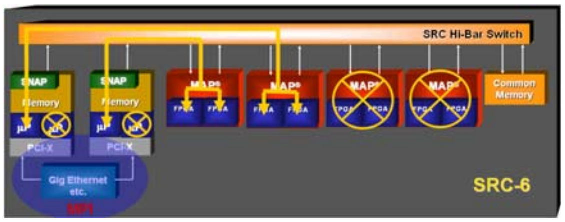
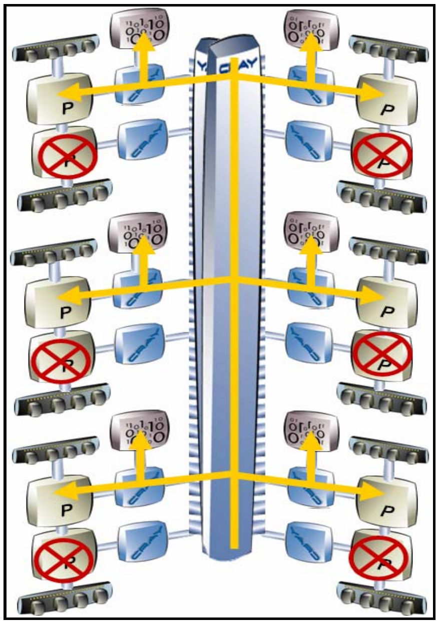


**Figure 6. Hardware-to-Software Performance [3]**



**a) MPI on the SRC-6**



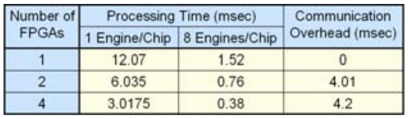
b) MPI on the Cray XD1

**Figure 6. Parallel Programming Issues**

Fig. 6 demonstrates how the parallel applications using MPI were distributed over both platforms. The figure also shows, as a case study, the specific installations of these machines at GWU. On the SRC-6, see Fig. 6(a), MPI processes had to run on the microprocessors and an association between each microprocessor and FPGA(s) is established. One limitation, however, was that the network interfacing cards (SNAPs) were limited to two and since they cannot be efficiently shared only two MPI processes would make sense. This made our SPMD program only limited to running on one microprocessor and one FPGA. Another FPGA can be used with some additional programming effort. At the end two microprocessors and four FPGAs were left unused; otherwise the programmer would have had to exert a large effort.

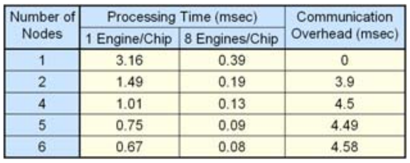
In the case of the XD1, see Fig. 6(b), straightforward application of MPI resulted in using all nodes. However, it only made sense to have the node program running on one microprocessor and its FPGA. The other microprocessor on each node was not useful.

**Table 4. Multi-Node Execution Time on SRC-6**



| Number of FPGAs | Processing Time (msec) | | Communication Overhead (msec) |
|---|---|---|---|
| | 1 Engine/Chip | 8 Engines/Chip | |
| 1 | 12.07 | 1.52 | 0 |
| 2 | 6.035 | 0.76 | 4.01 |
| 4 | 3.0175 | 0.38 | 4.2 |

**Table 5. Multi-Node Execution Time on Cray-XD1**



| Number of Nodes | Processing Time (msec) | | Communication Overhead (msec) |
|---|---|---|---|
| | 1 Engine/Chip | 8 Engines/Chip | |
| 1 | 3.16 | 0.39 | 0 |
| 2 | 1.49 | 0.19 | 3.9 |
| 4 | 1.01 | 0.13 | 4.5 |
| 5 | 0.75 | 0.09 | 4.49 |
| 6 | 0.67 | 0.08 | 4.58 |

Tables 4 and 5 show the performance of the ACCA algorithm on SRC-6 and Cray-XD1. The XD1 had an advantage with its FPGA chips running at 200 MHz, while the SRC machine had a restriction to run the FPGAs only at a 100 MHz speed. This explains the higher performance of XD1 system over SRC-6. The results from Tables 4 and 5 also show that the computation scalability on both machines is close to ideal. In other words, as the number of nodes increases the computation speedup increases with approximately the same factor. On the other hand, the communication overhead is almost constant. This is due to the fact that the same input data bands, i.e. same data size, is distributed/scattered across all nodes which results in almost the same communication overhead irrespective of the number of nodes. One can also note that this overhead is higher than the computation time. This is because the inherent parallelism of the application is fully exploited. This observation results in large deviation in the overall scalability from the ideal. This indicates that with the remarkable speed of FPGAs, overheads, such as communications, must be at a much lower levels than what is accepted in conventional high performance computers. However, the results also show that we may not need very large machines that are characterized with high overhead when HPRCs are used, which is a requirement for on-board preprocessing.

## 4. Conclusions

In this work, we extended our previous effort [3], in which we presented the design and implementation of the ACCA algorithm adapted for Landsat 7 ETM+ data, by investigating the potential of using multi-node HPRCs for on-board preprocessing. Landsat 7 ETM+ ACCA algorithm was selected to determine an almost practical bounds on the potential performance of HPRCs should all architectural challenges be alleviated and to gain an insight into the system level programmability and performance issues apart from those known for general high-performance computers.

We studied and characterized the scalability of the application on two of the state-of-the-art reconfigurable platforms, SRC-6 and Cray-XD1 at The George Washington University, High-Performance Computing Laboratory (HPCL).

The workload was distributed over all nodes using MPI. We scattered the input five bands across all nodes and gather the resulting mask pixels from all nodes at the base node.

The computation scalability on both machines was shown to be close to ideal, and the communication overhead was almost constant irrespective of the number of nodes. The inherent parallelism of the application was fully exploited. However, a large deviation in the overall scalability from the ideal was observed and analyzed.

Our analysis indicated that with the remarkable speed of FPGAs, overheads, such as communications, must be at a much lower levels than what is accepted in conventional high performance computers. However, the results also showed that we may not need very large machines that are characterized with high overhead when HPRCs are used, which is a requirement for on-board preprocessing.

## References

[1] A. Michalski, K. Gaj, D.A. Buell, "High-Throughput Reconfigurable Computing: A Design Study of an IDEA Encryption Cryptosystem on the SRC-6e Reconfigurable Computer", FPL 2005, pp.681-686

[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] 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, 11-14 Dec., 2005.

[4] "SRC-6 C-Programming Environment Guide", SRC Computers, Inc. 2005.

[5] "S-2433-131 Cray XD1$^{TM}$ Programming", Cray Inc., Oct. 2005.

[6] "S-6400-131 Cray XD1$^{TM}$ FPGA Development", Cray Inc., Oct. 2005.

[7] 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.

[8] 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.

[9] 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.

[10] 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.