PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

The quick RTE inversion on FPGA for DKIST

Cobos Carrascosa, J. P., Ramos Mas, J. L., Aparicio del Moral, B., Hernández Expósito, D., Sánchez Gómez, A., et al.

J. P. Cobos Carrascosa, J. L. Ramos Mas, B. Aparicio del Moral, D. Hernández Expósito, A. Sánchez Gómez, M. Balaguer, A. C. López Jiménez, D. Orozco Suárez, J. C. del Toro Iniesta, "The quick RTE inversion on FPGA for DKIST," Proc. SPIE 10707, Software and Cyberinfrastructure for Astronomy V, 107070L (6 July 2018); doi: 10.1117/12.2312569

Event: SPIE Astronomical Telescopes + Instrumentation, 2018, Austin, Texas, United States

The quick RTE inversion on FPGA for DKIST

J.P. Cobos Carrascosa[1], J.L. Ramos Mas, B. Aparicio del Moral, D. Hernández Expósito, A. Sánchez Gómez, M. Balaguer, A.C. López Jiménez, D. Orozco Suárez, J.C. del Toro Iniesta
{jpedro,ramos,bea, balaguer,dhdez,asgomez,antonio,orozco,jti}@iaa.es
Instituto de Astrofísica de Andalucía, IAA-CSIC. Apdo. de Correos 3004, E-18080, Granada, Spain.

ABSTRACT

In this contribution we present a multi-core system-on-chip, embedded on FPGA, for real-time data processing, to be used in the Daniel K. Inouye Solar Telescope (DKIST). Our system will provide "quick-look" magnetic field vector and line-of-sight velocity maps to help solar physicists to react to specific solar events or features during observations or to address specific phenomena while analyzing the data off line. The stand-alone device will be installed at the National Solar Observatory (NSO) Data Center. It will be integrated in the processing data pipeline through a software interface, and is competitive in computing speed to complex computer clusters.

Keywords: DKIST; RTE inversion; FPGA; SO/PHI;

1. INTRODUCTION

The Daniel K. Inouye Solar Telescope (DKIST) is a next generation ground-based solar telescope located at Haleakala High Altitude Observatory in Maui, Hawaii. It is currently under construction led by the US National Solar Observatory (NSO) with funding from the National Science Foundation (NSF). DKIST is designed to perform high-resolution spectropolarimetric visible and infrared measurements of the Sun. It will support five instruments which will operate in multiple configurations. These instruments will include 2048x2048 and 4096x4096 pixel cameras with frame cadences up to 30Hz. This means that DKIST will annually produce approximately 3 PB of data, via 5x108 images and 2x1011 metadata elements. The whole data set are sent to the Data Center, located at the NSO headquarters in Boulder (Colorado), which will perform post-processing and data reduction, and will make the data available to the user communities [1].

Translating the spectropolarimetric raw data into solar physical quantities is the main task of solar scientists that usually rely on complicated algorithms based on physical models of the solar atmosphere. These algorithms jell into very demanding computer codes that take significant time intervals even when run in big computer clusters. With the advent of instruments such as DKIST, these tasks grow by orders of magnitude and the mere selection of interesting portions of data to be analyzed implies difficult and tedious procedures. Moreover, the experienced observer has traditionally made decisions on line to react to specific events of features appearing during the observation. This decision-taking process has been based on purely raw data that are not specifically well suited to judge on the quality or the scientific discovery potential of given observational periods. The situation would clearly be improved if direct (maybe coarse) diagnostics were available like average values of the three components of the magnetic field and the line-of-sight (LOS) velocity. Our proposal aims at filling this lack of direct diagnostics in modern observations. We present a (multicore, system-on-chip) stand-alone, FPGA-based device capable of carrying out a very fast scientific analysis of some of these data, which will provide almost real-time vector magnetic field and LOS velocity maps. Specifically, we shall employ an inversion method based on the Milne-Eddington (ME) solution of the radiative transfer equation (RTE). The system is smaller, cheaper, and with lower power consumption than any computer cluster. It will allow assessing and classifying the huge amount of data received in a most comfortable way. Our system will be fed with data from the Visible Spectro-Polarimeter (VISP), one of the five instruments at the focus of DKIST.

[1] Corresponding author: J.P. Cobos Carrascosa Email: jpedro@iaa.es Phone: +34 958230656.
Postal Address: Instituto de Astrofísica de Andalucía, IAA-CSIC. Apdo. de Correos 3004, E-18080, Granada, Spain

Software and Cyberinfrastructure for Astronomy V, edited by Juan C. Guzman, Jorge Ibsen, Proc. of SPIE
Vol. 10707, 107070L · © 2018 SPIE · CCC code: 0277-786X/18/$18 · doi: 10.1117/12.2312569

We have already demonstrated the feasibility of using FPGA devices for carrying out the RTE inversion. The stringent requirements of the ESA's Solar Orbiter mission made it mandatory that our Polarimetric and Helioseismic Imager (PHI or SO/PHI) was endowed of an electronic inverter of the RTE working during the flight. The multi-core, system-on-chip FPGA reaches high computing performance by using a relatively slow and old-fashioned (yet space-qualified) FPGA device – Xilinx XQR4VSX55– and under strict time and power constraints [2]. The FPGA technology has evolved quickly and the internal resources have increased significantly in the last years. Therefore, using the latest FPGA device in the market –Xilinx XCVU9P- and without the limitations of a space mission, we can replicate our design up to 50 times in a single FPGA chip. This means that one observation can be inverted faster than with an average cluster of 50 computers. We show in this work how the FPGA design has been updated to the new device and we also introduce the smart data distribution network for feeding more than 500 computing cores placed inside the system on chip.

Section 2 summarizes the proposed device for carrying out the RTE inversion for DKIST. We outline how the initial hardware processing architecture has been improved and how it is integrated in the Data Center. In Section 3 we introduce the software interface to the inversion device for controlling the scientific computation and configuration. Section 5 concludes with brief comments about the proposed device and its main features.

2. INFACT: INVERSION FACTORY

The execution of the RTE inversion consists on an iterative process that requires the processing of a huge amount of data in parallel. In the harsh environment of outer space, we cannot use common devices like clusters of PCs or General Purpose Graphics Processing Unit devices (GPGPU) due to radiation problems and power limitations. Systems based on space-qualified processors are not sufficient either for the stringent requirements of SO/PHI. In [3] we demonstrated that a space-qualified FPGA device can be used as a high-performance computing element that is able of carrying out the RTE inversion aboard the SO/PHI instrument. Specifically, the proposal is focused in exploiting the data parallelism using several processors working together and using different data streams. This processing strategy is known as Single Instruction stream, Multiple data stream (SIMD). It is very suitable to deal with the RTE inversion since it can execute in parallel several spatial pixels of the 4 mega-pixel images of SO/PHI.

The architecture on a Virtex-4 FPGA for SO/PHI squeezes the FPGA internal resources in order to reach the time constraints. Specifically, we proposed in [3] a SIMD architecture composed by 12 processors. One of the most important contributions of that architecture is the ability of saving resources by allocating operation cores in a shared operation block (SOB), which is accessed by every processor. For example, there is only one floating-point division core since it needs around the 3% of the FPGA resources. Another critical core within the SOB is the Singular Value Decomposition block (SVD). SVD is one of the main steps in the RTE inversion and needs around 30% of all the FPGA resources. It basically performs the diagonalization of correlation matrices.

Hence, up to 12 processors together with one SOB are allocated in the SO/PHI RTE inverter. Within the SOB, the SVD core works as a coprocessor supporting the execution of this task in parallel with the rest of the RTE inversion. In this way, the processors can operate over 12 pixels and the unique SVD core does it over the 12 correlation matrices from other 12 pixels previously processed. Therefore, the multiprocessor architecture is also multithreaded since each processor is virtually in charge of two pixels at a given time.

When challenged to extend the space-borne RTE inverter to ground-based applications with much less constraints, a fan of technological possiblities opens. Other devices than FPGAs can be thought of. However, a cost-saving, yet time-efficient development is the replication of the available architecture within state-of-the-art FPGA devices. As mentioned in the Introduction, such a replication into a Xilinx XCVU9P FPGA implies an approximate improvement of a factor 50 with respect to the space-borne application. This improvement is good enough to comply with the minimum needs of DKIST and can eventually be replicated further since the limitations of power consumption do not exist any longer. As a result of a trade-off analysis between costs and capabilities, we have devised INFACT (INversion FACTory) based on a commercial development board from Xilinx as the final hardware platform. Specifically, we use the Xilinx Virtex UltraScale+ FPGA VCU118 Evaluation Kit. The platform offers INFACT the basic elements that it needs like communications and auxiliary memory, in addition to the FPGA itself.

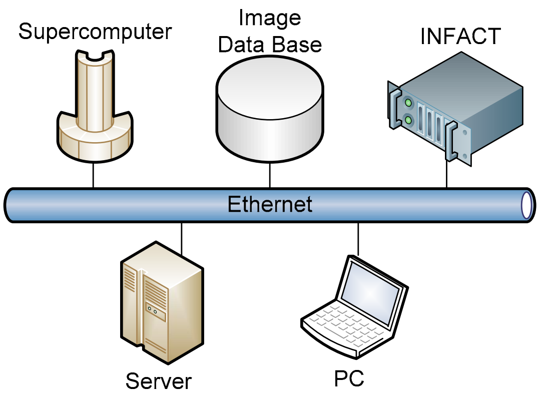


Figure 1 INFACT within the DKIST local network

In its new FPGA, INFACT contains the original SO/PHI architecture up to 40 times (see Figure 2). Each SIMD i-core represents the former one. In SIMD architectures there is only one instruction memory since there is only one instruction stream. We use this memory also for storing some control instructions, as reflected in the Control & Instruction Memory (CIM) of Figure 2. In order to save memory resources, we have extracted the CIM from each original SIMD core and only one CIM distributes the instructions to the 40 SIMD cores.

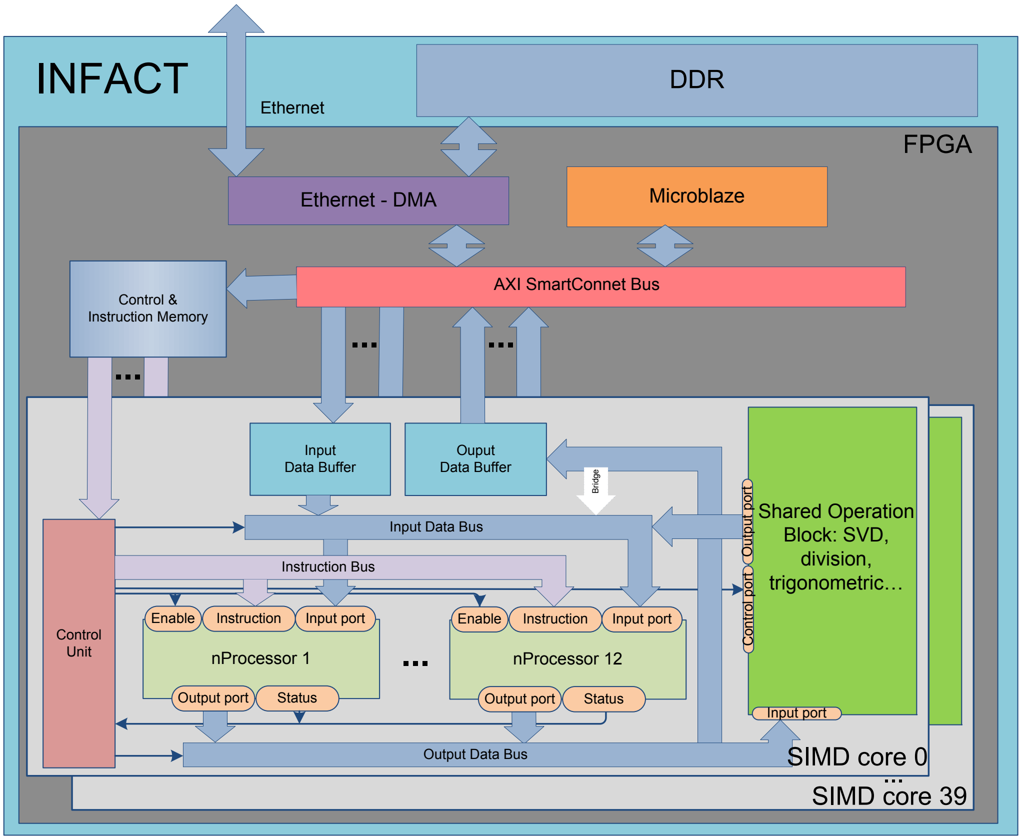


Figure 2 INFACT main-block architecture

The ratio of one SOB per 12 processors has been kept in order to maintain the well-balanced performance of the space-borne RTE inverter. This strategy also eases the use of TAPAS, the software tool we introduced in [2] that automates the entire design process and system settings for the RTE inversion core. This tool uses advanced techniques of software pipelining and parallelizing scientific algorithms in multicore systems. With some modifications, TAPAS is essential to configure INFACT because it has to deal with different RTE flavors according to some important configurable parameters.

INFACT will be allocated in the Ethernet network within the NSO Data Center, as sketched in Figure 1. It admits up to Gigabit Ethernet bandwidth which is enough for the inversion purposes. This organization allows any final user within the network to invoke inversions since our device is another computing element like supercomputers or servers where the entire data processing pipeline is executed. In fact, thanks to a software library it is integrated like an auxiliary software function within the mentioned pipeline. That software library is presented in detail in Section 3.

Figure 2 additionally shows the extra control and communication elements which are needed. The central element which communicates all the elements within INFACT is the AXI Smart Connect Bus, or simply AXI bus. Microblaze is an embedded general-purpose processor [4] which acts as the master in the architecture and it is in charge of controlling and configuring INFACT. It manages the data flow within the architecture. Basically, it distributes the input pixels between the SIMD processing elements. After that, it gathers the results.

The INFACT communications are based on Ethernet. The Ethernet-DMA block manages the communication flows according to the Microblaze commands. The input pixels are stored in the DDR memory while waiting to be processed. The results are also stored in the DDR before being sent by Ethernet. So, the DDR works as an input and output buffer over the data flow.

With this implementation, INFACT is a device 40 times more powerful that the used by SO/PHI. Taking into account that the latter is faster than a desktop computer (approximately) by a factor 10 (see [2]) then we can expect INFACT to provide a very competitive performance with respect to a cluster composed by 50 processors.

3. THE INFACT SOFTWARE LIBRARY

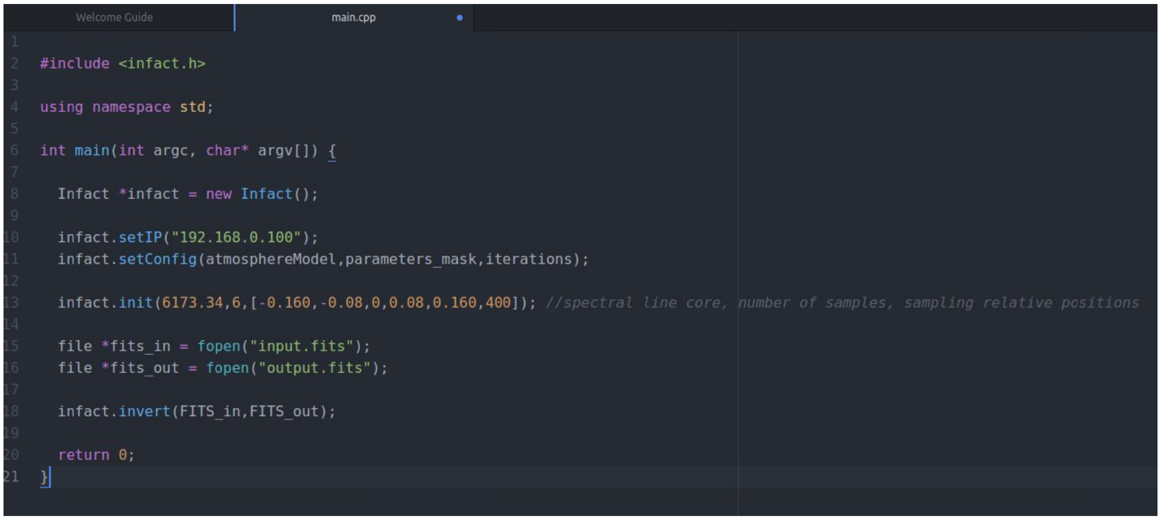
INFACT is aimed to be used as a processing server in the DKIST local network. Then, it is very important to define a very clear software interface that permits to integrate it within the scientific processing pipeline. This software interface manages the INFACT configurations and the data flows from whatever computer in the network.

On the one hand, the images to be processed can proceed from different instruments and, so, INFACT accepts configuration for attending them. The key aspects are the spectral line and the number of samples in that line (see, e.g., [3]). On the other hand, the images have to be accommodated in the order and format that INFACT expects. Finally, an agreed format and packaging of the resulting data has to be respected as well.

The software library (INFACT-LIB) makes most of the commented format and data packaged transparent to the software programmers. Figure 3 shows a program example with uses the INFACT-LIB. This example is written in C++ language, as well as the library itself. It is important that an object-oriented programming methodology is used for developing the library. This methodology relies in class declarations and objet instantiations.

Therefore, once the library is included in the program (note the sentence #include <infact.h>), the Infact class is declared and an infact object can be instantiated. This object manages all the features of INFACT. The first step is to indicate where is allocated INFACT in the local network. The setIP method is in charge of doing that. Note that INFACT is integrated in the network using a static IP address.

Besides the mentioned configurations about the spectral line and number of samples, there are other configurations intrinsic to the RTE inversions. The main parameters are the number of free-parameters in the Milne-Eddington (ME) atmosphere and the number of iterations within the iterative process. This configuration is carried out using the setConfig method, as shown in the Figure 3 example.



```cpp

#include <infact.h>

using namespace std;

int main(int argc, char* argv[]) {

  Infact *infact = new Infact();

  infact.setIP("192.168.0.100");
  infact.setConfig(atmosphereModel,parameters_mask,iterations);

  infact.init(6173.34,6,[-0.160,-0.08,0,0.08,0.160,400]); //spectral line core, number of samples, sampling relative positions

  file *fits_in = fopen("input.fits");
  file *fits_out = fopen("output.fits");

  infact.invert(FITS_in,FITS_out);

  return 0;
}
```

Figure 3 Example of using the INFACT software library

INFACT has to be initialized according to the spectral line and number and position of the samples. Using the init method, a pre-compiled code is sent to the CIM. The pre-compiled codes are prepared by TAPAS in advance; therefore, there is a limited set of possible CIM configurations.

Once the software has configured the INFACT hardware, it indicates the file where the images to be processed are contained and the result storing locations are specified. The invert method is in charge of organizing the spatial pixels in the adequate order and format. Without entering into details, the scheme about how the images are organized in a sequence of spatial pixels (Pixel) is shown in Figure 4 –in this example 6 wavelengths are assumed. As shown, each Pixel is composed by data of a specific spatial position from all the input I, Q, U, V images. This organization is transparent for a final INFACT user but it is important for understanding how it works. INFACT receives a serial flow of raw Pixels and returns a serial flow of inverted Pixels. There is no need to introduce a pixel position identifier in the input data flow because INFACT does not change the order and returns the results following the same order. The results are configured by the setConfig method through the parameters_mask parameter.

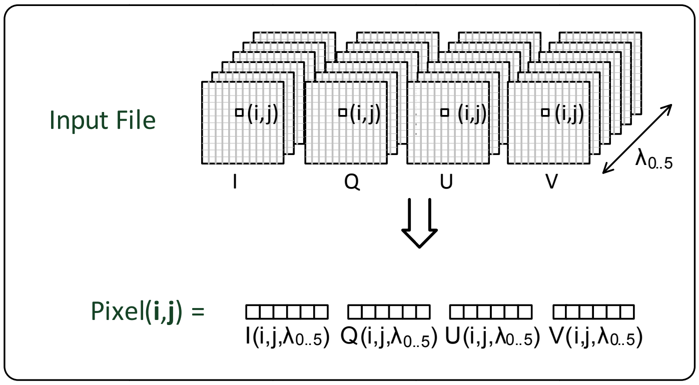


Figure 4 Input file organization and Pixel generation.

4. CONCLUSIONS

We have presented INFACT as a device to be integrated within the NSO Data Center local network as a server to analyze solar images for DKIST. It is aimed at providing “quick-look” maps of the vector magnetic field and the line-of-sight (LOS) velocity.

The scientific analysis is based on the RTE inversion and the processing architecture embedded on FPGA is an evolution of the computing proposal for the SO/PHI instrument. Basically, the architecture is composed by 40 SIMD cores. This model roughly implies 480 processors and 40 SVD cores working in parallel. In addition, since each processor is managing two threads, at a given time INFACT processes up to 960 pixels in parallel.

Together to the hardware device, the software library provides an easy interface which provides the means to configure the hardware and to send and receive data. It is in charge of packing the data flow from and to respective files.

It is very important to remark that since INFACT is integrated in the local network, its performance is totally scalable because several devices can be allocated in such network. In addition, the software library supports this kind of upgrade because the IP address is configurable by software in runtime.

The final stand-alone device will be integrated in the processing data pipeline and it is competitive in computing speed to computer clusters.

ACKNOWLEDGEMENT

This work has been partially funded by the Spanish Ministerio de Economía y Competitividad, through Project No. ESP2016-77548-C5-1-R, including a percentage from European FEDER funds.

REFERENCES

[1] S. Berukoff, T. Hays, K. Reardon, DJ Spiess, F. Watson, S. Wiant, "Petascale cyber infrastructure for ground-based solar physics: approach of the DKIST data center," Proc. SPIE 9913, Software and Cyber infrastructure for Astronomy IV, 99131F (26 July 2016); doi: 10.1117/12.2231899

[2] Cobos Carrascosa, J.P.; Ramos, J.L.; Aparicio del Moral, B.; Balaguer M.; Lopez Jimenez, A.C.; del Toro Iniesta, J.C. “SIMD architecture on FPGA for scientific computing aboard a space instrument”. Journal of Systems Architecture. Volume 62, January 2016, Pages 1-11, ISSN 1383-7621

[3] J.P. Cobos Carrascosa, B. Aparicio del Moral, J.L. Ramos, M. Balaguer, A.C. López Jiménez, J.C. del Toro Iniesta “The RTE inversion on FPGA aboard the solar orbiter PHI instrument”. Proceedings of SPIE - The International Society for Optical Engineering. Volume 9913, 2016, Article number 991342. Software and Cyber infrastructure for Astronomy IV; Edinburgh (UK). ISSN: 0277786X. DOI: 10.1117/12.2232332

[4] Xilinx INC. www.xilinx.com