On the hardware implementation of the arithmetic elements of the pairwise orthogonal transform

Lucana Santos
Ian Blanes
Aday García
Joan Serra-Sagristà
José López
Roberto Sarmiento

SPIE.

On the hardware implementation of the arithmetic elements of the pairwise orthogonal transform

Lucana Santos,[a] Ian Blanes,[b] Aday García,[a,*] Joan Serra-Sagristà,[a] José López,[a] and Roberto Sarmiento[a]

[a]University of Las Palmas de Gran Canaria, Institute for Applied Microelectronics, Campus Universitario de Tafira s/n, Las Palmas de Gran Canaria, Spain

[b]Universitat Autònoma de Barcelona, Group on Interactive Coding of Images, Bellaterra, 08193 Cerdanyola del Vallès, Barcelona, Spain

Abstract. The pairwise orthogonal transform (POT) is an attractive alternative to the Kahrunen–Loève transform for spectral decorrelation in on-board multispectral and hyperspectral image compression due to its reduced complexity. This work validates that the low complexity of the POT makes it feasible for a space-qualified field-programmable gate array (FPGA) implementation. A register transfer level description of the arithmetic elements of the POT is provided with the aim of achieving a low occupancy of resources and making it possible to synthesize the design on a space-qualified RTAX2000S and RTAX2000S-DSP. In order to accomplish these goals, the operations of the POT are fine-tuned such that their implementation footprint is minimized while providing equivalent coding performance. The most computationally demanding operations are solved by means of a lookup table. An additional contribution of this paper is a bit-exact description of the mathematical equations that are part of the transform, defined in such a way that they can be solved with integer arithmetic and implementations that can be easily cross-validated. Experimental results are presented, showing that it is feasible to implement the components of the POT on the mentioned FPGA. © *The Authors. Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI.* [DOI: 10.1117/1.JRS.9.097496]

1 Introduction

While the resolution of the sensors in the remote sensing space missions tends to increase, the bandwidth of the communication channel remains comparatively stable. As a consequence, the reduction of the captured data volume on-board satellites by means of compression is becoming ever more important, and becomes almost imperative for multispectral and hyperspectral sensors, since the amount of data captured is substantial.

When selecting or designing an algorithm for on-board data compression, we have to consider both, its coding efficiency and if the algorithm is suitable to be efficiently executed on the hardware that is available on a satellite. Among the possible hardware technologies that can be used for on-board data compression, field-programmable gate arrays (FPGAs) stand out, yielding a good compromise between performance and cost, as they present multiple advantages, such as the ability to apply parallel processing to increase throughput, provide flexibility, scalability, and data integrity features. However, space-qualified FPGAs, which are tolerant to solar radiation, are more difficult to manufacture, increasing their cost and reducing their performance. As a consequence, the requirement of low complexity for an on-board compression algorithm becomes even more important.

*Address all correspondence to: Aday García, E-mail: agarcia@iuma.ulpgc.es

The Consultative Committee for Space Data Systems (CCSDS) offers currently a solution for lossless multispectral and hyperspectral image compression (CCSDS 123).[1] This solution has been proven to take a reduced amount of hardware resources,[2] fitting in a small space-qualified FPGA.[3] Nevertheless, although lossless compression has been traditionally preferred to preserve all the information in the image, the current bandwidth and on-board storage restrictions make it necessary in some scenarios to resort to lossy compression in order to reduce the data volume appropriately. As a consequence, the CCSDS is currently working toward the definition of a lossy standard as an extension of the existing CCSDS 122 standard for image compression.[4] The latter defines a two-dimensional (2-D) image coder that combines a wavelet transform with a bitplane encoder and that, if coupled with a spectral transform, would take advantage of the spectral correlation present in multispectral and hyperpectral data.

Transform-based methods for compression are known to be well-suited for lossy compression. However, they are also known to be usually computationally complex in terms of the number of operations and memory requirements. In particular, in Ref. 5, several three-dimensional (3-D) transform coding techniques for lossy hyperspectral image compression are compared, showing increased rate-distortion performance when a spectral Kahrunen–Loève transform (KLT) is employed, especially in the low bit-rate region. However, the KLT presents important disadvantages when its implementation on hardware is considered: it has a high computational cost and has high memory requirements. All these factors lead to a higher implementation difficulty, and often become insuperable obstacles when considering an implementation on space-qualified FPGAs.

A divide and conquer strategy was proposed in Ref. 6, in which the pairwise orthogonal transform (POT) is presented. The POT is a simple adaptive transform that only requires a small amount of side information, which, in turn allows a line-based application of the transform not previously available to other divide-and-conquer approaches (or the KLT). In most cases, it has a better coding performance than wavelet transforms and, at the same time, it overcomes the inconveniences of the KLT, achieving a notably lower computational cost and better component scalability, with very low memory requirements (a few orders of magnitude less than a full KLT). The POT has a better performance than the wavelets, although it does not reach the coding performance of a full KLT.

This work aims at validating that the low complexity of the POT makes it suitable for a space-qualified FPGA implementation. A register transfer level (RTL) description of the main arithmetic stages of the POT, namely the mean subtraction, the training of balanced and unbalanced pairwise operations, and the lifting network are provided and optimized in order to of achieve a low occupancy of the hardware resources, making it possible to synthesize the design on a space-qualified RTAX2000S or on an RTAX2000S-DSP, which contains additional dedicated digital signal processor (DSP) blocks. In order to limit the dynamic range expansion of the transform, the variant of the POT introduced in Ref. 7 is the one being implemented.

Despite the POT being significantly less demanding in terms of hardware resources than the KLT, in this paper we make it even more amenable for an implementation on a space-qualified FPGA. In particular, a description of the mathematical operations in the transform is provided, in such a way that they can be solved with integer arithmetic, making the transform fully reversible and bit-exact (i.e., output values are unambiguous down to the bit level), which enables progressive lossy-to-lossless and makes implementations much easier to validate. Moreover, the most computationally demanding operations of the transform are solved by means of lookup tables (LUTs).

We note that, despite the POT being proposed to be included by the CCSDS in the upcoming standard for lossy multispectral and hyperspectral compression, the final standard may be different from the description of the operations in the POT presented in this paper. Moreover, the POT is only a part of the whole compression engine, which is being designed such that existing implementations of CCSDS-122.0, such as the ASICs described in Refs. 8 and 9, can be leveraged to create a complete compression engine.

This paper is structured as follows: Section 2 provides an overview of the arithmetic elements in the POT in a bit-accurate way, the subsequent Sec. 3 reports the hardware design considerations, explaining the methodology followed to create an RTL description of the latter operations, Sec. 4 shows the implementation results and finally, Sec. 5 presents the conclusions.
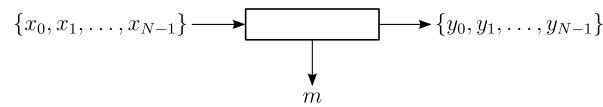
Fig. 1 Mean-subtraction operation.

2 The Pairwise Orthogonal Transform: A Bit-Exact Definition

The POT is an approximation of the KLT at a fraction of its computational cost. It is a composition of three kinds of operations that, when combined into a multilevel structure, are employed to obtain the transformed image. Namely, these operations are the mean-subtraction operation, the balanced pairwise operation, and the unbalanced pairwise operation. For conciseness, this paper does not include a detailed description of the POT or the implemented variant; descriptions which are already available in Refs. 6 and 7.

In the following subsections we present a definition of the arithmetic operations of the transform, which makes it amenable for an implementation on the hardware available on a satellite. Specifically, the operations are described in a bit-exact way, utilizing only integer arithmetic. This will facilitate significantly the subsequent physical implementation of the POT. In this paper, the implementation target is an FPGA of the Microsemi RTAX family, which is currently space-qualified and is flown in current space missions. Reducing the complexity of the arithmetic operations becomes even more important due to the limited computational power of these devices, when compared with other nonqualified FPGA models that can be found nowadays in the market.

2.1 Mean-Subtraction Operation

The mean-subtraction operation maps an input sequence of integer values $\{x_0, x_1, \ldots, x_{N-1}\}$ of size $N \geq 1$ to an output sequence $\{y_0, y_1, \ldots, y_{N-1}\}$, modifying the original values such that the output sequence has an arithmetic mean of approximately zero; see Fig. 1.

For a given input sequence, the value m is an approximation of the arithmetic mean of x_i, obtained as

$$m = \left\lfloor \frac{1}{N} \sum_{i=0}^{N-1} x_i \right\rfloor. \tag{1}$$

Given m, the result of the mean-subtraction operation is $y_i = x_i - m$ for $i = 0, \ldots, N-1$.

2.2 Pairwise Operation

The pairwise operation is defined in such a way that the output of the operation produces sequences that share virtually no correlation among each other.

Let $\{x_0, x_1, \ldots, x_{N-1}\}$ and $\{y_0, y_1, \ldots, y_{N-1}\}$ be two one-dimensional input sequences consisting of N integer samples each, and with an arithmetic mean of approximately zero. Given these sequences, the pairwise operation produces two output sequences, $\{\chi_0, \chi_1, \ldots, \chi_{N-1}\}$ and $\{\phi_0, \phi_1, \ldots, \phi_{N-1}\}$, consisting of N samples each, and with an arithmetic mean also of approximately zero, as shown in Fig. 2

The pairwise operation changes in function of its current inputs. Two training parameters B and C are first derived from the inputs and then used to specialize the operation. These parameters are required to invert the operation and are stored as side information.
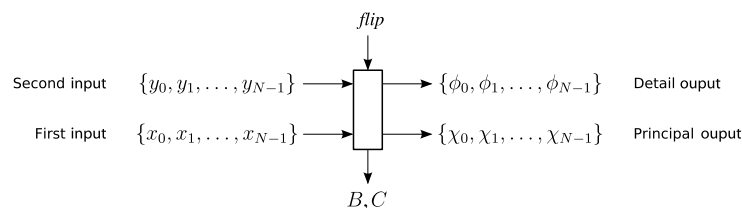

Fig. 2 The pairwise operation.

Additionally, the pairwise operation has a Boolean input, named flip, which changes the sign of all the elements of both output sequences.

The integer constant Ω indicates the precision of pairwise operations. It must be set within the interval [9, 16] and it is constant for the whole compression process of an input image. This constant can be used to regulate the complexity of the implementation at the expense of minor coding performance penalties.

As a prerequisite to calculate the training parameters, B and C the scaled-up variances and covariance of x_i and y_i should be calculated as follows:

$$\widetilde{\sigma_x^2} = \sum_{i=0}^{N-1} x_i^2 - N \cdot \left\lfloor \sum_{i=0}^{N-1} x_i \right\rfloor^2, \tag{2}$$

$$\widetilde{\sigma_y^2} = \sum_{i=0}^{N-1} y_i^2 - N \cdot \left\lfloor \frac{1}{N}\sum_{i=0}^{N-1} y_i \right\rfloor^2, \tag{3}$$

$$\widetilde{\sigma_{x,y}} = \sum_{i=0}^{N-1} (x_i \cdot y_i) - N \cdot \left\lfloor \frac{1}{N}\sum_{i=0}^{N-1} x_i \right\rfloor \cdot \left\lfloor \frac{1}{N}\sum_{i=0}^{N-1} y_i \right\rfloor. \tag{4}$$

The training parameters B and C are then calculated as follows. When $\widetilde{\sigma_{x,y}} \neq 0$,

$$B = \frac{\widetilde{\sigma_{x,y}}}{|\widetilde{\sigma_{x,y}}|} \quad \text{and} \quad C = \min\left(\max\left(\left\lfloor 2^{\Omega-4} \cdot \frac{\widetilde{\sigma_x^2} - \widetilde{\sigma_y^2}}{2 \cdot |\widetilde{\sigma_{x,y}}|} \right\rfloor, -2^{\Omega-2} \right), 2^{\Omega-2} - 1 \right). \tag{5}$$

Otherwise, when $\widetilde{\sigma_{x,y}} = 0$,

$$B = -1 \quad \text{and} \quad C = \begin{cases} 2^{\Omega-2} - 1 & \text{if } \widetilde{\sigma_x^2} \geq \widetilde{\sigma_y^2}, \\ -2^{\Omega-2} & \text{otherwise} \end{cases}. \tag{6}$$

After obtaining B and C, the decorrelated output sequences can be obtained through a lifting network. The lifting network, requires computing the weights $\tilde{w}_1$, $\tilde{w}_2$, and $\tilde{w}_3$, and two flags: one to indicate whether the result shall be sign-changed (flag s) and the another to signal whether the outputs χ_i and ϕ_i shall be permuted. Hence, the result of the pairwise operation (the values χ_i and ϕ_i) is produced by the lifting network shown in Fig. 3, where the values x_i and y_i are affected by three lifting steps (with weights $\tilde{w}_1$, $\tilde{w}_2$, and $\tilde{w}_3$), and the result is conditionally sign-changed and conditionally permuted.
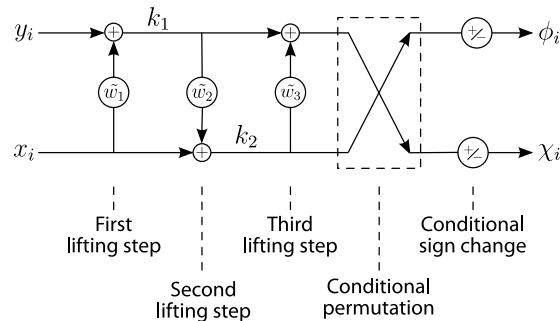

Fig. 3 Lifting network of the pairwise operation.

Weights $\tilde{w}_1$, $\tilde{w}_2$, and $\tilde{w}_3$ are obtained from B and C as follows. Let

$$
\tilde{t} = \begin{cases} 0 & \text{if } C = 2^{\Omega-2} - 1 \\ B \cdot 2^{\Omega-1} & \text{if } C = -2^{\Omega-2} \\ B \cdot \left\lfloor \sqrt{2^{2\Omega-3} - \left\lfloor \dfrac{2^{2\Omega+1} \cdot C}{\lfloor \sqrt{2^8 \cdot C^2 + 2^{2\Omega}} + 0.5 \rfloor} + 0.5 \right\rfloor} + 0.5 \right\rfloor & \text{otherwise} \end{cases}, \tag{7}
$$

$$
\tilde{p} = \left\lfloor (2^{\Omega+1}) \sqrt{1 - \left(\frac{\tilde{t}}{2^{\Omega-1}} \right)^2} + 0.5 \right\rfloor, \tag{8}
$$

and

$$
\widetilde{\sqrt{2}} = \lfloor (2^{\Omega+2}) \sqrt{2} + 0.5 \rfloor. \tag{9}
$$

If $C < 0$, then

$$
\tilde{w}_1 = \left\lfloor (2^{\Omega-4}) \frac{2\tilde{p} - \widetilde{\sqrt{2}}}{\tilde{t}} + 0.5 \right\rfloor, \tilde{w}_2 = \left\lfloor \frac{\widetilde{\sqrt{2}} \cdot \tilde{t}}{2^{\Omega+3}} + 0.5 \right\rfloor, \tilde{w}_3 = \left\lfloor (2^{\Omega-4}) \frac{4\tilde{p} - \widetilde{\sqrt{2}}}{\tilde{t}} + 0.5 \right\rfloor, \tag{10}
$$

and the output shall not be permuted; otherwise, when $C \geq 0$,

$$
\tilde{w}_1 = \left\lfloor (2^{\Omega-3}) \frac{\widetilde{\sqrt{2}} - 16\tilde{t}}{\tilde{p}} + 0.5 \right\rfloor, \tilde{w}_2 = \left\lfloor -\frac{\widetilde{\sqrt{2}} \cdot \tilde{p}}{2^{\Omega+4}} + 0.5 \right\rfloor, \tilde{w}_3 = \left\lfloor (2^{\Omega-3}) \frac{\widetilde{\sqrt{2}} - 8\tilde{t}}{\tilde{p}} + 0.5 \right\rfloor, \tag{11}
$$

and the output shall be permuted. The sign change is determined by

$$
s = \begin{cases} 1 & \text{if } (C \geq 0 \text{ or } B \geq 0) \quad \text{and flip is false} \\ -1 & \text{if } (C \geq 0 \text{ or } B \geq 0) \quad \text{and flip is true} \\ -1 & \text{if } C < 0 \quad \text{and} \quad B < 0 \quad \text{and flip is false} \\ 1 & \text{if } C < 0 \quad \text{and} \quad B < 0 \quad \text{and flip is true} \end{cases}. \tag{12}
$$

Output values χ_i shall be sign-changed when s is -1. Output values ϕ_i shall be sign-changed when $C < 0$ and s is -1, or when $C \geq 0$ and s is 1.

2.3 Unbalanced Pairwise Operation

When the POT is applied to images with a number of spectral bands that is not a multiple of two, some pairwise operations need to be slightly modified, yielding the unbalanced pairwise operation described next (Fig. 4).

An unbalanced pairwise operation is a pairwise operation, as described previously, except for the following two variations.

The training parameters B and C are calculated as follows. When $\widetilde{\sigma_{x,y}} \neq 0$,

$$
B = \frac{\widetilde{\sigma_{x,y}}}{|\widetilde{\sigma_{x,y}}|} \quad \text{and} \quad C = \min \left(\max \left(\left\lfloor 2^{\Omega-4} \cdot \frac{2\widetilde{\sigma_x^2} - \widetilde{\sigma_y^2}}{\lfloor \frac{1448}{1024} \cdot |\widetilde{\sigma_{x,y}}| \rfloor} \right\rfloor, -2^{\Omega-2} \right), 2^{\Omega-2} - 1 \right). \tag{13}
$$

Otherwise, when $\widetilde{\sigma_{x,y}} = 0$,

$$
B = -1 \quad \text{and} \quad C = \begin{cases} 2^{\Omega-2} - 1 & \text{if } 2\widetilde{\sigma_x^2} \geq \widetilde{\sigma_y^2}, \\ -2^{\Omega-2} & \text{otherwise} \end{cases}. \tag{14}
$$

The weights $\tilde{w}_1$, $\tilde{w}_2$, and $\tilde{w}_3$ are defined as follows: if $C < 0$, then

$$\tilde{w}_1 = \left\lfloor (2^{\Omega-3}) \frac{\left\lfloor \frac{\widetilde{\sqrt{2} \cdot \tilde{p}}}{2^{\Omega+2}} + 0.5 \right\rfloor - 2^{\Omega+2}}{\tilde{t}} + 0.5 \right\rfloor, \tilde{w}_2 = \left\lfloor \frac{\tilde{t}}{2} + 0.5 \right\rfloor, \tilde{w}_3$$

$$= \left\lfloor (2^{\Omega-5}) \frac{\left\lfloor \frac{\widetilde{\sqrt{2} \cdot \tilde{p}}}{2^{\Omega-1}} + 0.5 \right\rfloor - 2^{\Omega+4}}{\tilde{t}} + 0.5 \right\rfloor,$$

and the output is not permuted; otherwise, when $C \geq 0$,

$$\tilde{w}_1 = \left\lfloor (2^{\Omega-1}) \frac{\left\lfloor \frac{\widetilde{\sqrt{2}(2^{\Omega-2}-\tilde{t})}}{2^{\Omega}} + 0.5 \right\rfloor}{\tilde{p}} + 0.5 \right\rfloor, \tilde{w}_2 = \left\lfloor -\frac{\widetilde{\sqrt{2} \cdot \tilde{p}}}{2^{\Omega+4}} + 0.5 \right\rfloor, \tilde{w}_3$$

$$= \left\lfloor (2^{\Omega-1}) \frac{\left\lfloor \frac{\widetilde{\sqrt{2}(2^{\Omega}-\tilde{t})}}{2^{\Omega+2}} + 0.5 \right\rfloor}{\tilde{p}} + 0.5 \right\rfloor,$$

and the output is permuted.

3 Hardware Implementation of the Arithmetic Elements of the Pairwise Orthogonal Transform

We present in the following sections the implementation of the operations in the POT in order to estimate its complexity in terms of utilization of hardware resources and assess the feasibility for an implementation on a space-qualified FPGA. More specifically, the following operations are described at RTL: mean-subtraction; computation of the variance and covariance; computation of the training parameters, B and C; the computation of the weights; and the lifting network. A flowchart showing the relationship between the aforementioned operations in the POT is displayed in Fig. 5.

3.1 *Main Design Considerations*

The main objective of the presented hardware implementation is to optimize the amount of FPGA resources taken by the design; hence, we focus on reducing as much as possible the complexity of the operations, scheduling them sequentially when possible. For instance, the integer division operations which are needed to compute the mean values and to obtain the training parameter C are solved by means of serial divisors in order to reduce the amount of hardware resources.

It is decided to split the design in several modules, which are described in VHDL. We designed a module that implements the mean subtraction, variance and covariance calculation, and the necessary operations to obtain the training parameters B and C, as it was described in Sec. 2. Once B and C are available, it is necessary to calculate the weights. From the equations presented in Sec. 2, we can infer that these operations are rather complex and might become
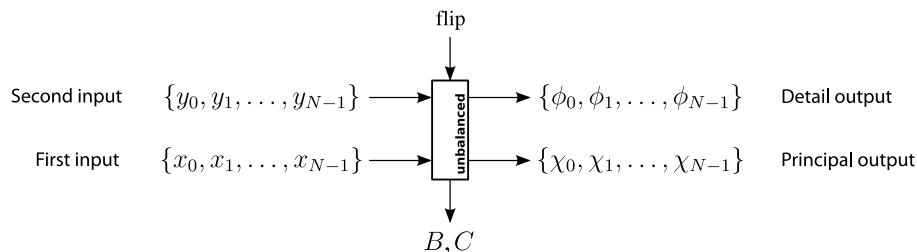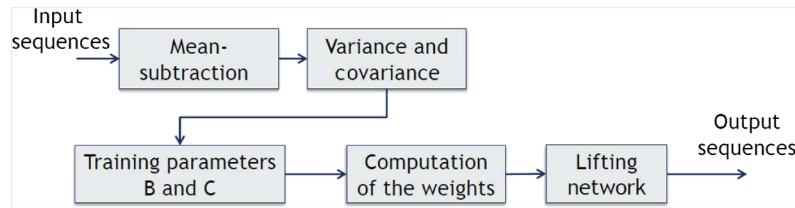

Fig. 4 The unbalanced pairwise operation.

Fig. 5 Flowchart of the different operations in the pairwise orthogonal transform (POT), which are implemented in a field-programmable gate array.

computationally expensive for an FPGA. We have observed, however, that it is possible reduce the complexity and computational load of the overall design by utilizing a LUT that, given the parameters B, C, flip, and a flag that indicates is the transform is balanced or unbalanced, will produce the desired weights ($\tilde{w}_1$, $\tilde{w}_2$, and $\tilde{w}_3$); the sign change; and the flag indicating that the output shall be permuted. Finally, a module which implements the lifting network is also described in VHDL. The basic block diagram of the design is shown in Fig. 6.

In order to find out which is the best implementation solution for the LUT, from which the weights are obtained, we perform an estimation of the amount of resources needed to implement it only with combinatorial logic. The results are presented in Table 1, where it can be observed that the number of combinatorial cells needed to implement such LUT is highly dependant on Ω. Given that the combinatorial cells required for such LUT rise considerably as Ω grows—with, for example, a 14% of the RTAX2000S resources employed for $\Omega = 11$—an alternative solution is proposed in order to reduce the area occupancy, which is to store the necessary LUT values in the internal RAM of the FPGA. The values can be stored in an external electrically erasable/ programmable read-only memory (EEPROM), that would be loaded to RAM upon system start-up.

3.2 *Register Transfer Level Description*

We present the RTL description of the module that performs the POT. This module will receive the input sequences and produce the decorrelated sequences, as specified in Sec. 2.

The designed module, shown in Fig. 7, contains an input interface to a memory where the necessary samples are stored. A memory controller is implemented to access this memory. The decorrelated sequences (detail and principal) are obtained at the output, together with the necessary side information. A valid signal is included to validate the results. First, the input samples are subtracted their average and the variance and covariance values are calculated, and then the training parameters are obtained. From the training parameters B and C, it is possible to obtain the weights, using a LUT. As it was mentioned in Sec. 4, we assume the values in the LUT are in an external EEPROM, that is loaded to the FPGA internal RAM upon system start up. The internal RAM blocks are protected with an error detection and correction scheme able to correct single bit errors and detect double bit errors in the memory content. Finally, the decorrelated values are finally obtained from the weights using the lifting network module.

The following parameters are set as generics in the VHDL code, in such a way that it is possible to estimate how the occupancy of the design varies with them.
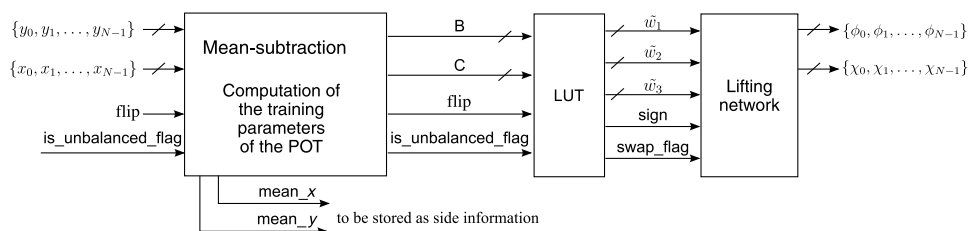

Fig. 6 Block diagram of the implemented elements of the POT: mean-subtraction, computation of training parameters, production of the weights with a lookup table, and lifting network.

Table 1 Combinatorial cells needed to implement the LUTs.

Ω	Combinatorial cells
9	917
10	1685
11	3010
12	5446

- Number of samples in the input sequences (N).
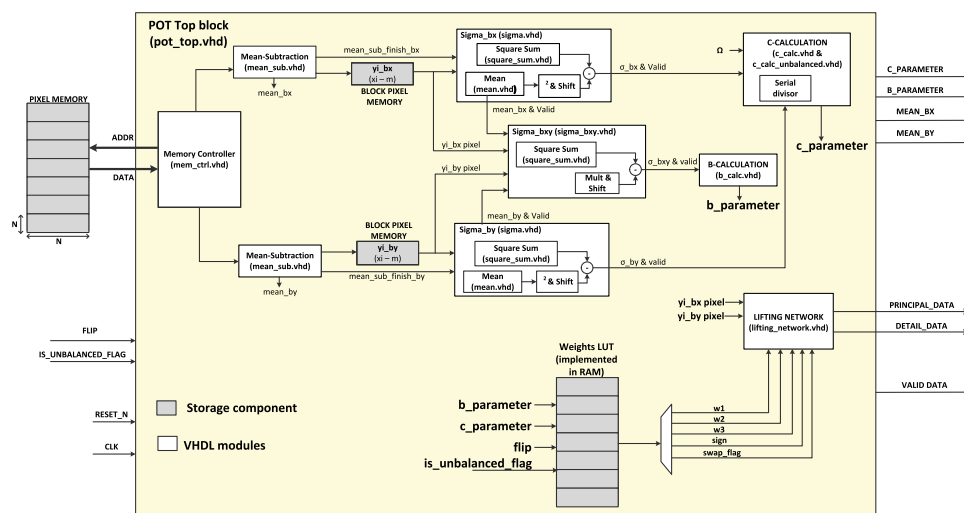- Bit width of the input samples.

Additionally, we include a parameter to indicate if the number of input samples is a power-of-two. In the case the parameter is set, the implemented divisors, which are used to calculate the mean values, are avoided and the division is performed with bit shifts instead.

4 Implementation Results

The synthesis of the previously described module has been performed on an RTAX2000S and an RTAX2000S-DSP, which are manufactured by Microsemi and are space-qualified. The main difference between them is that the RTAX2000S-DSP contains dedicated DSP blocks. Several different configurations of the top module are synthesized, varying the parameter that indicates if the number of input samples is a power-of-two (P2), varying the number of input samples (N), or varying the number of bits per pixel (bpp). In all cases Ω is set to 16. The different combinations of parameters can be observed in Fig. 8. We note that, even when N is a power-of-two, if we do not inform explicitly the core with the corresponding parameter, the value will be interpreted as a nonpower-of-two value, i.e., the integer divisors will be used, they will not be replaced with bit shifts. This is the case of configurations C5 to C8.

Table 2 shows the hardware occupancy and maximum frequency of the implemented module obtained for the different configurations in both target FPGAs. The occupancy results are presented in terms of the percentage of combinatorial cells, sequential cells, and blocks of RAM (BRAM) utilized. For RTAX2000S-DSP device the percentage of used DSP blocks is also displayed.

In order to facilitate the comparison of the presented results, the occupancy values are also displayed in the graphs of Figs. 9 and 10. We can observe that the complete module

Fig. 7 Architecture of the mean-subtraction, balanced, and unbalanced operations.

implementing the POT can be synthesized on an RTAX2000S for any of the presented configurations. The most demanded resource are the combinatorial cells in the RTAX2000S and the internal BRAM in the RTAX2000S-DSP. Comparing both FPGA models, we can see that the number of combinatorial cells occupied is reduced for the RTAX2000S-DSP, which has taken advantage of the DSP blocks that it contains to perform specific computations that, otherwise, would be performed with combinatorial logic. We can also observe that the occupancy of resources is significantly higher when the input samples are represented with 16 bpp. However, there is not a significant increase in the number of resources utilized when we change the number of input samples from 256 to 512.

The frequency results can be observed and compared in Fig. 11. It can be seen that the frequency increases significantly with the RTAX2000S-DSP. The maximum and minimum frequencies for the RTAX2000S are 37.58 and 29.92 MHz, respectively. For the RTAX2000S-DSP the maximum frequency is 73.50 MHz and the minimum frequency is 53.83 MHz. The frequency values decrease as the bit-width of the input samples increases. No significant differences in the frequencies are found when the number of samples increases.

The intent of this paper is to show the feasibility of an FPGA implementation of the POT, and as such the implementation has not been optimized for throughput. Anyhow, a brief discussion follows to illustrate on which could be the throughput of the whole compression engine. The proposed implementation is able to produce a transformed sample every four clock cycles, which
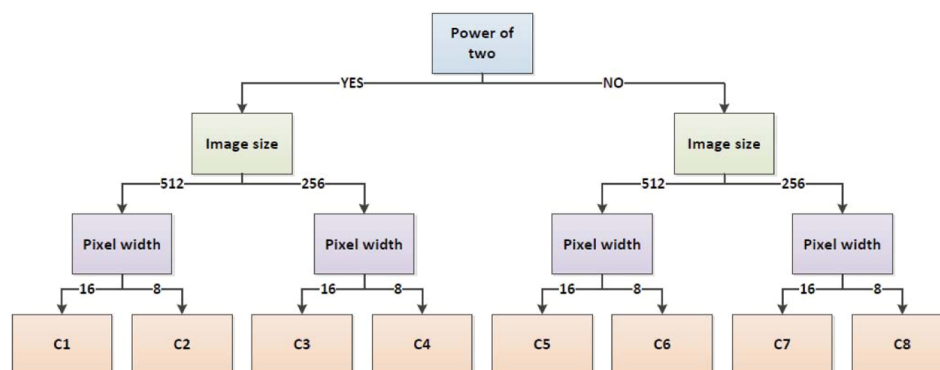

Fig. 8 Structure for parameter setting and resulting configurations.

Table 2 Synthesis results on a RTAX2000S and on an RTAX2000-DSP for different configurations.

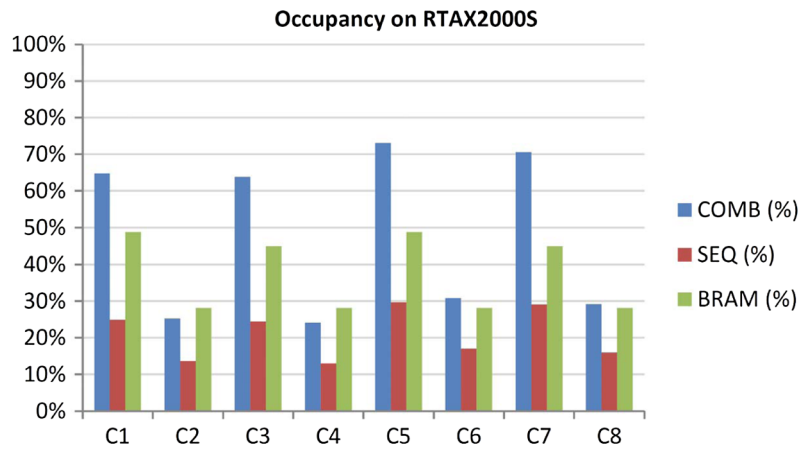|				RTAX2000S				RTAX2000S-DSP				
P2	N	bpp	ID	COMB (%)	SEQ (%)	BRAM (%)	Frequency (MHz)	COMB (%)	SEQ (%)	BRAM (%)	DSP blocks (%)	Frequency (MHz)
Y	512	16	C1	64.82	24.96	48.75	29.92	21.73	18.48	48.75	33.75	53.83
Y	512	8	C2	25.25	13.67	28.13	37.00	14.07	12.74	28.13	22.50	72.00
Y	256	16	C3	63.84	24.48	45.00	30.92	21.07	18.01	45.00	33.75	54.67
Y	256	8	C4	24.13	13.05	28.13	37.58	13.51	12.07	28.13	22.50	73.50
N	512	16	C5	73.04	29.65	48.75	29.92	21.73	18.48	48.75	33.75	53.83
N	512	8	C6	30.81	17.01	28.13	37.00	18.80	15.46	28.13	22.50	72.00
N	256	16	C7	70.60	29.14	45.00	36.08	27.80	22.05	45.00	33.75	54.67
N	256	8	C8	29.22	15.98	28.13	37.58	17.89	14.78	28.13	22.50	73.50

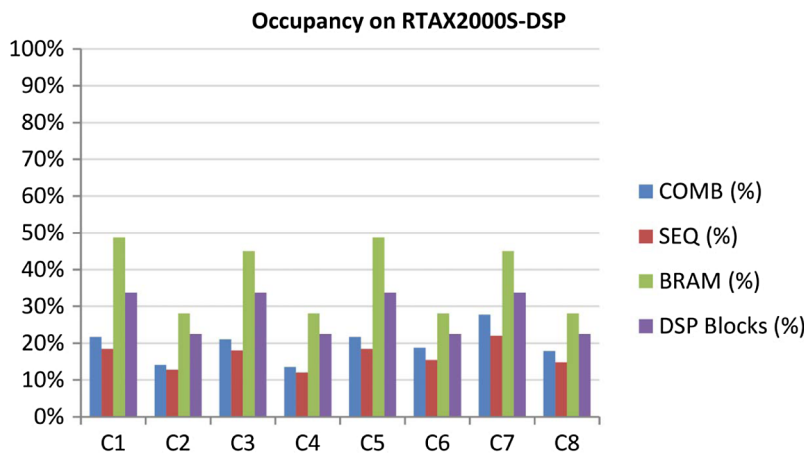**Fig. 9** Occupancy of the different configurations on an RTAX2000S.

Fig. 10 Occupancy of the different configurations on an RTAX2000S-DSP.
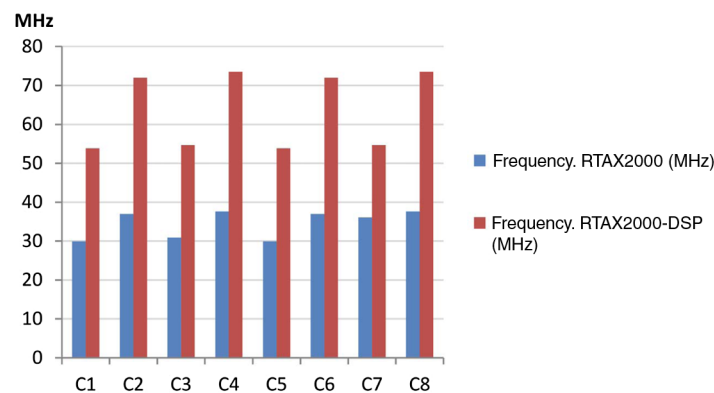

Fig. 11 Maximum frequency of the different configurations on an RTAX2000S and an RTAX2000S-DSP.

yields a minimum throughput of 12.5 MSamples/s for configuration C1 and a maximum of 18.4 MSamples/s for configuration C8.

To complete the whole 3-D compression engine, a 2-D compressor implementing the CCSDS-122.0, such as the CCSDS wavelet image compression (CWICOM) ASIC, is utilized after the POT. The CWICOM is able to operate at up to 60 MSamples/s.[8] While the throughput of both elements is of similar magnitude, future optimizations of the POT implementation will be focused on improving its throughput, in such a way that it can yield one transformed sample per clock cycle.

5 Conclusions

This paper examines the hardware complexity of the arithmetic elements of the POT, namely the mean-subtraction operation, the training of the transform, and the lifting network. A bit-exact description of the operations is provided and utilized to create a hardware implementation with reduced complexity that only utilizes integer arithmetic. A LUT is used to eliminate the most computationally demanding operations of the POT. We perform the synthesis of the described hardware on two space-qualified FPGA, namely the RTAX2000S, and the RTAX2000S-DSP with a different number of input samples represented with different bit depths. We observed that for any of the studied configurations, the complete design fits in the target FPGAs. The worst-case in terms of occupancy shows a 74% of combinatorial cell usage in the RTAX2000S. The best-case for the same FPGA has a maximum occupancy of 24.13% of the combinatorial cells. Furthermore, we observed that the RTAX2000S-DSP offers the possibility of reducing substantially the amount of combinatorial cells used, by utilizing dedicated DSP blocks. This FPGA also yields higher frequency values, displaying a maximum frequency of 73 MHz.

These results show that the POT, which is only a part of a whole compression engine for multispectral and hyperspectral images, presents a low hardware complexity, making it a good candidate for an on-board hardware implementation.

References

1. Consultative Committee for Space Data Systems (CCSDS), *Lossless Multispectral & Hyperspectral Image Compression CCSDS 123.0-B-1*, Blue Book, CCSDS (2012).
2. N. Aranki et al., "Fast and adaptive lossless on-board hyperspectral data compression system for space applications," in *Proc. Aerospace Conf., 2009 IEEE*, pp. 1–8, IEEE (2009).
3. L. Santos et al., "Hyloc: a low-complexity FPGA implementation of the CCSDS-123 standard algorithm for multispectral and hyperspectral compression," in *Proc. 2014 ESA Workshop on Onboard Payload Data Compression (OBPDC)*, ESA (2014).
4. Consultative Committee for Space Data Systems (CCSDS), *Image Data Compression CCSDS 122.0-B-1*, Blue Book, CCSDS (2005).
5. B. Penna et al., "Transform coding techniques for lossy hyperspectral data compression," *IEEE Trans. Geosci. Remote Sens.* **45**, 1408–1421 (2007).
6. I. Blanes and J. Serra-Sagristà, "Pairwise orthogonal transform for spectral image coding," *IEEE Trans. Geosci. Remote Sens.* **49**, 961–972 (2011).
7. I. Blanes et al., "Isorange pairwise orthogonal transform," *IEEE Trans. Geosci. Remote Sens.* **53**(6), 3361–3372 (2014).
8. R. Vitulli and J.-L. Poupat, "CWICOM; the new CCSDS image compression ASIC," in *Proc. 2012 ESA Workshop on Onboard Payload Data Compression (OBPDC)*, ESA (2010).
9. P. Winterrowd et al., "A 320 Mbps flexible image data compressor for space applications," in *Proc. 2010 IEEE Aerospace Conf.*, pp. 1–6 (2010).

Lucana Santos received her PhD degree in 2014 from the University of Las Palmas de Gran Canaria (ULPGC). Currently, she is a postdoctoral researcher at the Integrated Systems Division of the Institute for Applied Microelectronics (IUMA). She was a visiting researcher at the European Space Research and Technology Center in 2011, where she did research on hardware architectures for hyperspectral image compression. From 2012 to 2014 she participated in an industrial project for Thales Alenia Space España.

Ian Blanes received his BS, MS, and PhD degrees in computer science from the Universitat Autònoma de Barcelona (UAB) in 2007, 2008, and 2010, respectively. Since 2003, he has been with the Group on Interactive Coding of Images of the UAB, where he currently holds a postdoctoral position. He was a visiting researcher at Centre National d'Etudes Spatiales. He was the second-place finisher as the best Spanish computer science student.

Aday García received his telecommunication engineer MSc degree from the ULPGC in 2013, where he has been working towards his PhD at the Integrated Systems Design Division of the IUMA. Since 2008, he has been working in the aerospace industry and has been involved in the development of on-board data processing boards for several European Space Agency programs, such as EUCLID and the Solar Orbiter and Meteosat Third Generation missions.

Joan Serra-Sagristà received his PhD degree in computer science from UAB, Spain, in 1999. From 1997 to 1998 he was at the University of Bonn, Germany. Currently, he is an associate professor at UAB. His research interests focus on source coding and data transmission. He serves as an associate editor for *IEEE Transactions on Image Processing*. He has coauthored over 125 publications. He was the recipient of the Spanish Intensification Young Investigator Award in 2006.

José López received his BS degree in physics from the University of Seville in 1989 and his PhD degree in telecommunication engineering from the ULPGC in 1994. He is a full professor in electronic technology and deputy director of the IUMA. In 1992, he was with Thomson-CSF, France. He was an invited researcher at the Technical University of Denmark (1995) and at Edith Cowan University, Australia (1997 and 2000).

Roberto Sarmiento received his PhD degree in electronic engineering from the ULPGC in 1991. He is a full professor at the Telecommunication Engineering Faculty at the ULPGC. He is a cofounder of the IUMA and director of the Integrated Systems Design Division of IUMA. Since 1990, he has published more than 50 journal papers and book chapters and more than 140 conference papers.