ARC 2014 Over-Clocking KLT Designs on FPGAs under Process, Voltage, and Temperature Variation

RUI POLICARPO DUARTE and CHRISTOS-SAVVAS BOUGANIS, Imperial College London, Electrical and Elecctronic Engineering

Karhunen-Loeve Transformation is a widely used algorithm in signal processing that often implemented with high-throughput requisites. This work presents a novel methodology to optimise KLT designs on FPGAs that outperform typical design methodologies, through a prior characterisation of the arithmetic units in the datapath of the circuit under various operating conditions. Limited by the ever-increasing process variation, the delay models available in synthesis tools are no longer suitable for extreme performance optimisation of designs, and as they are generic, they need to consider the worst-case performance for a given fabrication process. Hence, they heavily penalise the maximum possible achieved performance of a design by leaving safety margin. This work presents a novel unified optimisation framework which contemplates a prior characterisation of the embedded multipliers on the target FPGA device under process, voltage, and temperature variation. The proposed framework allows a design space exploration leading to designs without any latency overheads that achieve high throughput while producing less errors than typical methodologies, operating with the same throughput. Experimental results demonstrate that the proposed methodology outperforms the typical implementation in three real-life design strategies: high performance, low power, and temperature variation; and it produced circuit designs that performed up to 18dB better when over-clocked.

Categories and Subject Descriptors: B.6.3 [**Design Aids**]: Optimization; B.8.1 [**Reliability, Testing, and Fault-Tolerance**]

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Design methodologies, digital signal processing, reconfigurable applications, optimisation, over-clocking

ACM Reference Format:
Rui Policarpo Duarte and Christos-Savvas Bouganis. 2015. ARC 2014 over-clocking KLT designs on FPGAs under process, voltage, and temperature variation. ACM Trans. Reconfigurable Technol. Syst. 9, 1, Article 7 (November 2015), 17 pages.
DOI: http://dx.doi.org/10.1145/2818380

1. INTRODUCTION

The Karhunen-Loeve Transformation (KLT) algorithm, also known as linear projection, is used in many scientific areas to compress data, that is, face recognition [Ngo et al. 2005], Electroencephalogram (EEG) [Ke and Li 2009], Electromyography (EMG) [Chu et al. 2006], Synthetic Aperture Radar (SAR) [Nascimento and Bioucas Dias 2005]. Additionally, the advent of *big data* and near real-time performance requirements has propelled an increased demand in performance for implementations of this algorithm.

This algorithm benefits from being implemented on Field-Programmable Gate Arrays (FPGAs), as they offer high performance along with low power consumption,

Authors' addresses: R. P. Duarte; email: rui.duarte@ist.utl.pt; C.-S. Bouganis; email: christos-savvas.bouganis@imperial.ac.uk.

DOI: http://dx.doi.org/10.1145/2818380

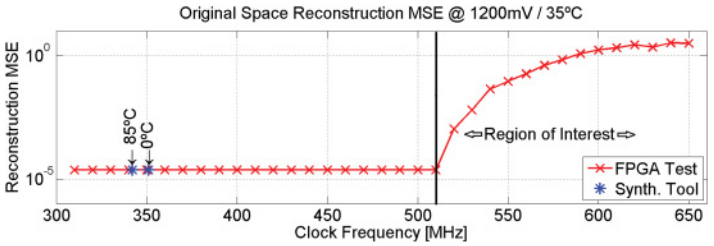


Fig. 1. Mean squared error of the reconstruction of the KLT in the original space *versus* its clock frequency, as well as the maximum operating clock frequencies provided by the conservative models of the synthesis tools. The region of interest corresponds to the clock frequencies for which the KLT circuit generates errors when ran on a DE0 board from Terasic.

parallelism, and highly specialised embedded blocks. The latter is particularly relevant as the KLT algorithm relies heavily on the computation of multiplications and additions.

In real-life implementations, the most common performance-limiting factors of the KLT algorithm are hard area requirements, more specifically when the KLT targets a large number of dimensions, unrolling or deeply pipelining the design is unfeasible; and synthesis tools that utilise models without information about the actual device that is being targeted. Thus, they are conservative in their estimate on the performance of the components.

In this work, to overcome the performance limitation of the embedded multipliers, which cannot be deeply pipelined, the KLT circuit is over-clocked to extreme frequencies beyond those reported by the synthesis tool in order to push further its overall performance.

A novel methodology is proposed to optimise KLT designs under a unified framework for over-clocking, which considers different sources of variation simultaneously. A key step of the proposed framework is the performance characterisation of the embedded multipliers in a given device when they are clocked with clock frequencies well above from what is reported by the synthesis tool. The main idea is to have a prior characterisation of the device with respect to the degradation of the performance of the computational units when operate at specific operational regime. Such offline characterisation is possible in an FPGA device due to their reconfigurability capabilities.

Forasmuch as the performance of designs implemented using FPGAs are affected by Process, Voltage, and Temperature (PVT) variation, like any other silicon device. Moreover, process variation affects the physical dimensions of the transistors on the device, thus changing its threshold voltages and timing constraints, consequently making them with uneven performances across the device. Then as well, embedded multipliers on different locations will have different timing limitations. To address this problem, the proposed methodology supports design strategies with different operating conditions, such as low power and high performance. Moreover, in implementations where voltage and clock frequency are fixed and do not change over time, the temperature may vary, as it is very expensive to fine control. The proposed methodology can optimise designs for graceful degradation over a wide range of temperatures.

The proposed framework optimises extreme over-clocking of KLT designs into error-prone operation, designated as *Region of Interest* in Figure 1. This figure shows the evolution of errors, on a Cyclone III FPGA, with the increase of clock frequency, in a KLT design from a Z^6 space to a Z^3 using the embedded multipliers on the FPGA. It also shows the maximum clock frequencies reported by the synthesis tool for different temperatures, which demonstrate the performance penalty due to the tool's conservative models, and not the actual delay in the targeted device.

The proposed framework captures the performance characterisation of the targeted FPGA device and exposes it to the algorithm specification in order to perform the design space exploration, resulting in implementations of KLT designs with improved performance, while minimising errors without the expense of extra circuit resources. The obtained performance information (i.e., errors that are expected at the output of the multipliers) is injected into a Bayesian formulation of the problem in order to improve the performance of the KLT designs. This framework uses an error model (for specific operating conditions), and later automatically combines this information with high-level parameter selection of the algorithm, generating designs less prone to error, when compared to typical implementations of the KLT algorithm.

The main contributions in this article are:

—Extension of the characterisation and optimisation framework prototypes, in Duarte and Bouganis [2012, 2014b], to support embedded multipliers;
—Introduction of the support for PVT variation in the characterisation framework and in the optimisation algorithm, including support for different device families;
—Development of an error model for the embedded multipliers under a range of operating conditions;
—Optimisation of KLT designs for performance targeting different scenarios (i.e., low-power, high-performance, temperature variation resilience).

2. BACKGROUND

Usually the implementation and performance of KLT designs in a digital system is bound to the hardware resources occupied and its maximum performance. Typical design optimisation techniques often translate them into a tradeoff between the number of bits used in quantisation and the depth of pipelining. However, such methodologies are unable to cope with variation of the operating conditions, and the intra-die and inter-die variation.

Moreover, the ever increasing process variation and the fact that circuits need to support different operating voltages and temperatures, makes the designs to operate at lower clock frequencies than the maximum offered by the fabrication process. Therefore, synthesis tools use conservative models which set the maximum performance of a circuit below the performance of the worst transistor for the family of the device.

Knowing that this margin accounts for variation of the operating conditions and aging of the device, one can over-clock the design as means to increase the throughput at the expense of becoming prone to timing errors.

One of the most well-known techniques that can be applied to address the timing errors problem in a datapath is Razor [Ziesler et al. 2003]. It is a generic time-redundant method proposed for Dynamic Voltage Scaling (DVS) of CPUs. In this work, a shadow register is used to validate the actual data, and in case of mismatch, to correct the data. More recently, Das et al. [2009] proposed an extension to this methodology to recover from errors due to PVT. Both strategies can be applied to any path prone to errors due to timing violations, and the recovery is performed at the expense of extra circuit area and latency, which penalise applications of the KLT algorithm for processing data streams.

Sedcole and Cheung [2008] present two strategies to compensate for *intra-die* performance variability by providing a generic characterisation step for the performance of the device followed by a reconfiguration step, where parts of the design are mapped to specific locations of the device given their performance requirements. Notwithstanding a small improvement in performance is obtained, it is not resilient to timing errors.

A novel approach to improve the performance of KLT designs, using Constant Coefficient Multipliers (CCMs), relied on over-clocking of the design [Duarte and Bouganis 2012]. Notwithstanding, this work is restricted to CCMs and does not consider

different operating conditions demanded by different design strategies. In this article, this constraint is lifted to make the proposed framework practical in a wider range of real-life applications. Furthermore, a new optimisation framework that utilises information from a prior characterisation for a KLT design optimisation is presented. It includes an algorithm for design space exploration that utilises an objective function tuned for the utilisation of embedded multiplier modules within that KLT framework under different operating conditions.

The proposed framework breaks new ground proposing:

—graceful degradation of results at the output of the KLT circuit with the increase in variation of the working conditions,
—a methodology to push forward the performance of embedded multipliers without using extra circuitry,
—a per device optimisation, and
—a methodology to optimise a design over a set of varying conditions performing better than accounting for the worst-case scenario.

The following sections of this article detail the proposed methodology to accelerate KLT designs, while being resilient to PVT variation, based on the prior characterisation of the device.

2.1. KLT Revisited

The KLT, also known as linear projection, or Principal Component Analysis (PCA), transform is formulated as follows. Given a set of N data $x^i \in R^P$, where $i \in [1, N]$ an orthogonal basis described by a matrix Λ with dimensions $P \times K$ can be estimated that projects these data to a lower dimensional space of K dimensions. The projected data points are related to the original data through the formula in Equation (1), written in matrix notation, where $X = [x^1, x^2, \ldots, x^N]$ and $F = [f^1, f^2, \ldots, f^N]$, where $f^i \in R^K$ denote the factor coefficients.

$$F = \Lambda^T X. \tag{1}$$

The original data is described from the lower dimensional space via Equation (2):

$$X = \Lambda F + D, \tag{2}$$

where D is the error of the approximation. The objective of the transform is to find a matrix Λ such as the Mean-Square Error (MSE) of the approximation of the data is minimised. A standard technique is to evaluate the matrix Λ iteratively as described in steps (3) and (4), where λ_j denotes the j^{th} column of the Λ matrix:

$$\lambda_j = arg\ max\ \mathrm{E}\left\{\left(\lambda_j^T X_{j-1}\right)^2\right\} \tag{3}$$

$$X_j = X - \sum_{k=1}^{j-1} \lambda_k \lambda_k^T X, \tag{4}$$

where $X = [x^1 x^2 \ldots x^N]$, $X_0 = X$, $\|\lambda_j\| = 1$ and $\mathrm{E}\{.\}$ refers to expectation.

2.2. KLT Implementations

The KLT algorithm is based on the dot-product operation, which can be implemented using different circuits. Figure 2 shows (a) rolled and (b) unrolled architectures of a dot-product circuit to implement the datapath of one projection vector from a Z^p to Z^k KLT. This work focuses on the rolled architecture, instead of the unrolled one, due to the savings in circuit resources, and the limitation of the unrolled design when

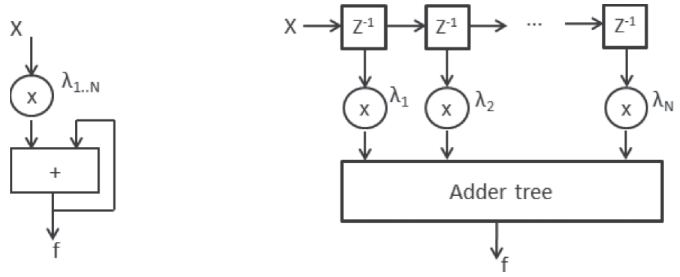


Fig. 2. Schematics of the datapath of a dot-product, for one projection vector of a KLT circuit: (a) rolled architecture and (b) unrolled architecture.

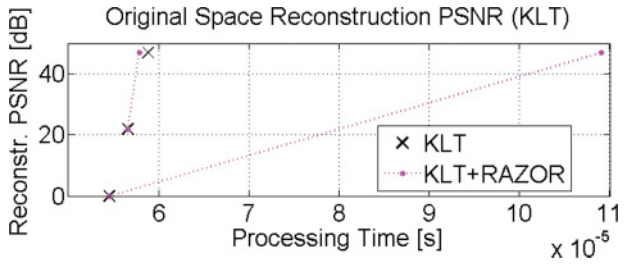


Fig. 3. Errors *versus* processing time tradeoff.

targeting large KLT designs. Moreover, since embedded multipliers cannot be deeply pipelined, there would not be any advantage using the unrolled architecture. The rolled architecture also reduces the number of embedded multipliers to be characterised. In this circuit, sign-magnitude representation was adopted because negative numbers, represented in 2's complement, tend to be more deviated from the expected result than the positive ones. Furthermore, data was quantised with 9 bits, but different word-lengths are supported.

The circuit receives data from the input stream, identified with X. The samples, from the input stream, for each dimension p, are multiplied by the corresponding projection vector λ_{pk}. The output of the multiplier is connected to an adder to do the accumulation. The final result is placed in the output stream, identified with f_k.

2.3. Errors vs. Processing Time Tradeoff

As mentioned earlier, Razor [Ziesler et al. 2003] is recovery mechanism at the expense of extra circuit area and clock latency, which is not suitable to perform computations on data streams such as the case of the KLT algorithm. In the interest of comparing both methods, and without accounting for extra circuit resources and power required by Razor, their time to completion of a linear projection was computed, when compared to the time overhead introduced by Razor on KLT designs.

Figure 3 shows the values for the reconstruction PSNR, of linear projections over 5,000 points from six to four dimensions, for KLT only designs, and respective processing times required to compute the result. Different processing times correspond to different clock frequencies (510, 530, and 550MHz). Points with a reconstruction PSNR close to 50dB represent no over-clocking errors in the computations. The dotted lines make the correspondence between the time required by a design at a specific clock

frequency and its implementation with Razor, without any reconstruction errors. From this plot, it is possible to conclude that as the error rate increases with the increase of over-clocking, Razor takes longer to recover than performing computations at a lower clock frequency. Hence, in cases where performance matters, it may be preferable to produce an approximate result on time rather than waiting for the correct one.

2.4. KLT Optimisations

The key idea in Bouganis et al. [2010] is to inject information about the hardware (i.e., in this case about the required hardware resources of a CCM) as a prior knowledge in the Bayesian formulation of the above optimisation problem. In more detail, the proposed framework in Bouganis et al. [2010] estimates the basis matrix Λ, the noise covariance Ψ, and the factors using Gibbs sampling algorithm [Geman and Geman 1984] from the posterior distribution of the variables, having injecting knowledge about the required hardware recourses for the implementation of the CCMs through a prior distribution. Thus, a probability density function is generated for the unknown Λ matrix, which is used to generate samples, where the prior distribution tunes this posterior distribution, and thus accommodating the impact of the required hardware resources.

Duarte and Bouganis [2012] provides an extension of the above work for the optimisation of KLT designs using CCMs combating the effects for circuit area as well as performance variation due to over-clocking. This work is focused on the extension of the previous work to support embedded multipliers and PVT variation in the characterisation, error modeling, and generation of designs to implement Equation (1). The framework selects the multipliers used for the implementation of each dot-product in Equation (1) along with the coefficients of the Λ matrix that define the lower dimension space.

3. OPTIMISATION OF KLT DESIGNS FOR OVER-CLOCKING

Notwithstanding, a device can operate at higher clock frequencies than the ones reported by the synthesis tools, which can be determined via experimentation. Nevertheless, a margin in performance needs to be preserved to contemplate variation due to process variation, operating conditions (i.e., temperature and voltage), and aging of the device. The proposed methodology tries to close this gap by pushing the clock frequency further into the error-prone regime.

In the circuit to implement the KLT design, the datapath holds the most critical paths. The main purpose of this work focuses on over-clocking embedded multipliers, as they are the components with the largest delay in the datapath of the design. Other FPGAs with more advanced embedded arithmetic blocks, such as DSP blocks capable of producing multiplications followed by addition, can be targeted as the multiplier has a longer critical path when compared to the critical path of the adder.

The calculation of the projection matrix Λ and its hardware mapping onto FPGAs are often considered as two independent steps in the design process. However, considerable area savings can be achieved by coupling these two steps as shown in Bouganis et al. [2007, 2010]. The Bayesian formulation presented in Bouganis et al. [2010] considers the subspace estimation and the hardware implementation simultaneously, allowing the framework to efficiently explore the possibilities of custom design offered by FPGAs. This framework generates KLT designs that minimise timing errors and circuit resources, when compared to the standard approach of the KLT transform application followed by the mapping to the FPGA.

The framework treats the Λ matrix as a random matrix and generates a probability density function for it, which is used to sample its values from. The framework allows to insert prior information in the sampling of the Λ matrix, changing the posterior distribution to accommodate the impact of uncertainty on it. The prior distribution of the Λ matrix, $p(\Lambda, f, l, v, T)$, is the product of the probabilities of the individual

elements at a clock frequency f, location on the FPGA l, core voltage v, and the temperature of the device T,

$$p(\Lambda, f, l, v, T) = \prod_{p=1}^{P} \prod_{k=1}^{K} p(\lambda_{pk}, f, l, v, T). \tag{5}$$

This prior distribution expresses the uncertainty of each Λ matrix and has to meet the following properties:

$$p(\lambda_{pk}, f, l, v, T) \geq 0, \forall \lambda_{pk}, \tag{6}$$

$$\sum_{\lambda_{pk}} p(\lambda_{pk}, f, l, v, T) = 1. \tag{7}$$

3.1. Objective Function

Timing errors are a consequence of over-clocking the design and they are observed in the projection factors. They are represented as ε and are assumed to have a Gaussian distribution with variance and mean obtained from the characterisation process. Thus, the reconstructed data can be expressed as a function of the Λ matrix and as the timing error:

$$\hat{X} = \Lambda(F + \varepsilon) \tag{8}$$

Errors from the approximation are measured in terms of MSE, and it is defined as follows:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\hat{X}_i - X_i)^2. \tag{9}$$

The objective function is formed by the MSE of the reconstructed data in the original space, and errors that are produced due to the over-clocking, under PVT variation, of the utilised embedded multipliers.

Let us denote with $\hat{X}$ the result of the reconstruction of the projected data in a matrix form. Then, the objective function U is defined as in Equation (10), where both reconstruction errors and variation errors are captured. E denotes the expectation and Tr the trace operator. The matrix formulation is defined as:

$$U = \text{Tr}(\text{E}[(X - \hat{X})^T (X - \hat{X})]). \tag{10}$$

By imposing ε to have zero mean, which is achieved by subtracting a constant in the circuit, and using the fact that the Λ matrix is orthogonal and orthonormal, the objective function is expressed as:

$$\begin{aligned} U &= \text{Tr}(\text{E}[(X - \Lambda F)^T (X - \Lambda F)]) + \text{Tr}(\text{E}[\varepsilon^T \varepsilon]) \\ &= \text{Tr}(\text{E}[(X - \Lambda F)^T (X - \Lambda F)]) + \sum_j \text{var}(\varepsilon_j). \end{aligned}$$

Here, j denotes the columns of the Λ matrix. By assuming that the errors at the output of the multipliers are uncorrelated, the first term in the final expression relates to the approximation of the original data from the KLT without any variation error, where the second term relates to the variance of the errors at the output of the embedded multipliers due to over-clocking, under PVT variation. Thus, the errors due to dimensionality reduction and variation are captured by one objective function without any need to formulate a problem using a multi-objective function.

Other applications could be supported as long as they can be formulated by an objective function which can be minimised.

3.2. Prior Distribution Formation

The proposed framework utilises information regarding the performance characterisation of the embedded multipliers for a given device and their respective resource utilisation, by suitably constructing a prior distribution function for the coefficients of the Λ matrix. The utilised models for the over-clocking errors under PVT variation are described below.

3.2.1. Error Models. The proposed framework utilises the performance characterisation framework for CCMs, introduced in Duarte and Bouganis [2012] and now extended to support embedded multipliers and capture PVT variation. By executing that framework, a profile of the errors expected at the output of the embedded multipliers when one of the operands is fixed (i.e., representing a coefficient of the Λ matrix) for various operating conditions can be obtained. As indicated by the objective function formulation, the objective is to capture the variance of the error at the output of the multiplier that models the uncertainty of the result. As such, $Err(m, f, P, v, T)$, holds information regarding the error variance at the output of a multiplier when a stream of data is multiplied by a constant m, the circuit is clocked at frequency f, placed on P coordinates on the FPGA, using core voltage v, and temperature T. Moreover, it assumes Independent and Identically Distributed (IID) input data.

3.2.2. Prior Distribution. The formation of the prior distribution $p(\cdot)$ of the Λ matrix is a key part of the framework, as it injects hardware information to the framework for the estimation of the Λ matrix. The aim of the prior distribution is to penalise Λ matrix instances with high uncertainty, as a result of the use of coefficients that generate high errors due to over-clocking by assigning low probabilities to them. As no information regarding the distribution of the coefficients is available on their suitability in representing the original space, this part of the prior distribution is uninformative and results to a flat prior. Thus, the prior distribution reflects solely information about the errors at the output of the over-clocked multipliers as $p(\lambda_{pk}, f, P, v, T) = g(Err(\lambda_{pk}, f, P, v, T))$, where the performance of every coefficient in the Λ matrix is dictated by the targeted clock frequency, the placement on the FPGA, the core voltage, and the temperature of the device; and $g(\cdot)$ denotes a user defined function. In this work, the following $g(\cdot)$ function is selected, as it provides good results, without any claim on its optimality.

$$g(Err(\lambda_{pk}, f, P, v, T)) = c_E(1 + Err(\lambda_{pk}, f, P, v, T))^{-\beta} \tag{11}$$

c_E is a constant used to ensure that $\sum_{\lambda_{pk}} g(Err(\lambda_{pk}, f, P, v, T)) = 1$. β is a *Hyper-Parameter* that allows the tuning of the contribution of errors in the prior distribution. $Err(\lambda_{pk}, f, P, v, T)$ is the variance of the error observed from the performance characterisation of the multiplier.

3.3. Characterisation Process

The aim of the proposed characterisation framework is to capture the errors at the output of arithmetic units when placed on various locations on an FPGA, clock frequencies, placement and routing configurations, temperatures, and voltages. The information produced by this characterisation framework is to be utilised by the optimisation framework. Such an approach is only possible due to the reconfigurability that is offered by FPGA devices.

Prototypes of the characterisation frameworks for CCMs and generic Look-Up Table (LUT)-based multipliers were introduced in earlier contributions [Duarte and Bouganis 2012, 2014a]. The present framework extends all previous contributions as it supports

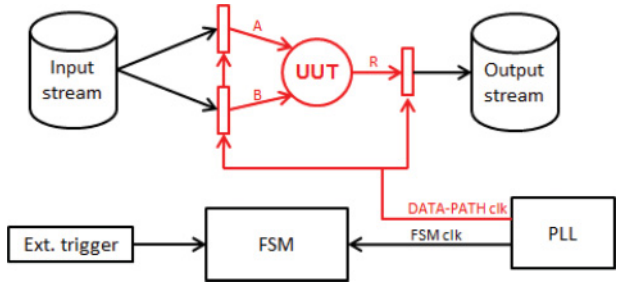


Fig. 4. Schematic of the characterisation circuit.

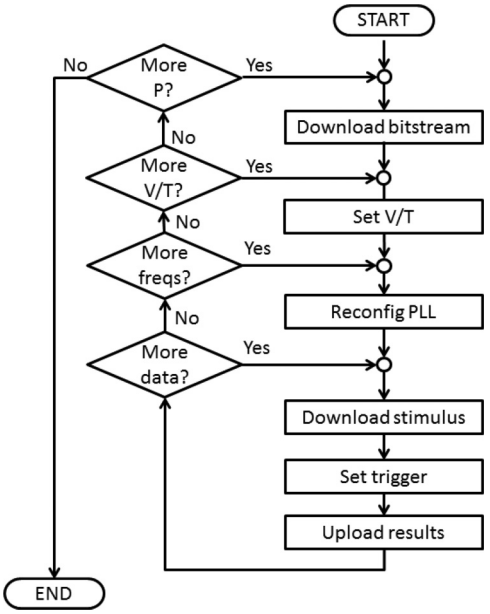


Fig. 5. Flowchart of the characterisation process executed by the software side on the host computer.

embedded multipliers and the test of a wider range of designs under different operating conditions (clock frequency, voltage, and temperature) and enhances its functionality.

Figure 4 shows the schematic of the circuit to do the characterisation of arithmetic units. It is formed by the unit under test, that is, embedded multipliers, and the test supporting blocks, that is, Block RAMs (BRAMs), Finite State Machine (FSM), and Phase-Locked Loop (PLL). The BRAMs hold the input stimulus and the output results, the FSM controls the test execution, and the PLL generates the clock signals for the datapath and the control logic. This circuit implemented on a Cyclone III FPGA requires 1034 Logic Elements (LEs) and 13 BRAMs, and its FSM can be clocked up to 910MHz while operating correctly.

The interactions between the test circuit and the host computer are performed by a Tool Command Language (TCL) script written to handle the characterisation process. The flowchart in Figure 5 illustrates the sequence of actions executed by the TCL script written to interact with the characterisation circuit on the FPGA. The characterisation process starts with the download of the bitstream with the characterisation circuit with the unit under test in place. It then proceeds to set the operating conditions

(i.e., voltage, temperature, clock frequency) via their interfaces with the host computer. After that, the script downloads the stimulus data and starts the test of the arithmetic unit with a constant value in one of its inputs. Once the test finishes, the results are uploaded to the host computer. Later, a Matlab script produces statistics on the errors for each constant multiplicand (i.e., error variance and mean error).

The core voltage of the FPGA was provided by a digital power supply PL303QMD-P [Aim & Thurlby Thandar Instruments 2013] from TTI. The temperature of the FPGA was set by a thermoelectric cooler placed on top of the FPGA. The temperature controller was calibrated using a digital thermometer from Lascar Electronics [2012] and its deviation is below 1°C. The location of the embedded multipliers on the FPGA was set using LogicLock Regions inside Chip Planner in Quartus II.

3.3.1. Characterisation Runtime. The time required by the characterisation framework depends on the number of operating parameters (clock frequency, voltage, temperature) considered for the arithmetic units, as well as the length of the test vector v. The approximate runtime, in seconds, for a single set of operating conditions, to characterise a constant coefficient of an embedded multiplier is given by:

$$T_{char} = 1.7143 \left\lceil \frac{v}{2000} \right\rceil. \tag{12}$$

3.4. Design Generation

The proposed framework uses Gibbs sampling [Geman and Geman 1984] to extract, from the design space, a set of designs that minimise the selected objective function U. The resulting designs are the ones that minimise the value of the objective function. The proposed framework estimates each dimension (i.e., column) of the Λ matrix in a sequential manner. The user supplies the targeted dimensions K, the targeted clock frequency f, the coordinates on the FPGA P, the core voltage v, the temperature T, and the β parameter. The pseudo-code for the optimisation algorithm is given in Algorithm 1.

ALGORITHM 1: KLT Design Unified Framework for Over-Clocking

Require: $K \geq 1 \wedge \beta > 0 \wedge f, p, v, T > 0$
Ensure: 1 KLT design
  $X \leftarrow input$ {original data N cases}
  for $d = 1$ **to** K **do**
    Create new empty $Candidate\_Projs$ list
    $prior \leftarrow generate\_prior(\beta, f, p, v, T)$
    $\lambda_d \leftarrow sample\_projection(X, prior)$
    $F \leftarrow (\lambda_d^T \lambda_d)^{-1} \lambda_d^T X$
    $error \leftarrow X - \sum_{j=1}^{d} \lambda_j F$
    $MSE_d \leftarrow \sum \sum error^2 / PN$
    $Proj \leftarrow (\lambda_d, MSE_d)$
    Add $Proj$ to $Candidate\_Projs$ list
    Extract candidate projections {min MSE}
  end for
  return The KLT design with minimum MSE

4. EVALUATION OF OVER-CLOCKED KLT CIRCUITS UNDER PVT VARIATION

The performance of the proposed methodology was compared against the performance of the reference design, which is based on a typical implementation of the KLT

algorithm. All designs were implemented on a Cyclone III EP3C16 FPGA from Altera [2012], attached to a DE0 board from Terasic Technologies [2009].

The aim of this case study is to demonstrate that an optimisation of a KLT design targeting different design strategies (e.g., low-power, high-performance), under different operating conditions, using the same framework is achievable. The effectiveness of the framework is demonstrated with a case study implementing a linear projection from Z^6 to Z^3. The characterisation of the FPGA, the training of the framework and the test used different sets of data from a uniform pseudo-random distribution, quantised with 9 bits. After synthesis, the tool reported a resource usage of 126 LEs and three 9×9 embedded multipliers, and a maximum clock frequency of 342MHz. Examining the timing report revealed that the critical paths belong to the embedded multiplier and the delay for the remaining components in the datapath, that is, accumulator, and the FSM, are out of reach for the selected over-clocking frequencies. The results from the characterisation of the embedded multipliers were validated using Transition Probability from Wong et al. [2008]. To better demonstrate the impact of variation for each design strategy, only one setting (voltage/temperature/location) has been changed for each test. Nevertheless, the framework supports variation of many operating conditions simultaneously. For the rest of this article, the results for the reference implementation without information about the characterisation of the device are identified with KLT, whereas the results for the proposed framework are identified with NEW. They are compared in terms of Peak Signal-to-Noise Ratio (PSNR) of the reconstructed data in the original space.

4.1. Characterisation of Process Variation in Arithmetic Units

Initially the characterisation was performed on three embedded multipliers on two Cyclone III FPGAs, on different DE0 boards, to assess the impact of intra- and inter-die variation. Figure 6 shows the error variance for all constant coefficients of a 8×8 unsigned multiplier on three embedded multipliers at different clock frequencies on different locations of the same Cyclone III FPGA. This figure shows that the errors are different for each embedded multiplier tested, that is, the same constant coefficient exhibits different levels of error or no error at all. Consequently, since each multiplier has a different error profile, a design optimised for a given location, or a specific FPGA, will not achieve the same performance on other locations. Figures 7 and 8 show the error variance for different embedded multipliers on different locations of the same device, Cyclone III and IV FPGAs.

4.2. Optimisation Targeting Maximum Performance

Performance of an FPGA can be improved by increasing its core voltage and decreasing its package temperature. For this design strategy, the device was kept at 5°C and supplied with 1,400mV, instead of the 1,200mV specified by the manufacturer.

With a clock frequency twice as much as the maximum specified by the synthesis tool, the designs generated by the proposed framework exhibited a reconstruction PSNR up to 15dB better than the KLT designs for the same working conditions, as can be observed in Figure 9. On the other hand, if a target PSNR of 30dB is to be met, then the designs generated by the framework can operate up to 20MHz higher than the KLT designs.

4.3. Optimisation Targeting Low Voltage

KLT circuits operating under limited power budgets, or battery operated, tend to operate using the least core voltage possible and be without any active cooling components. Figure 10(a) shows the results for the KLT designs when operating at 35°C with different FPGA core voltages. This design strategy considered 900mV as the minimum core

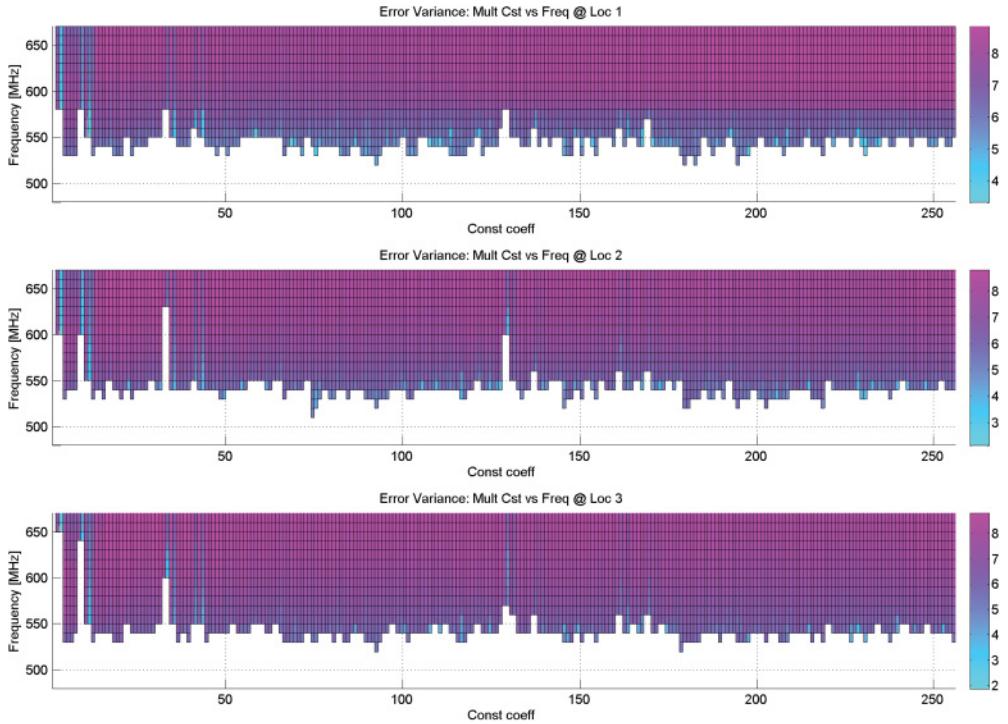
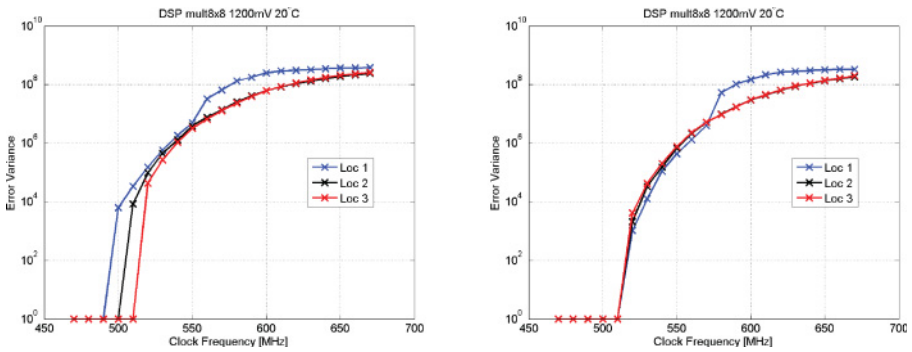


Fig. 6. Error variance for all constant coefficients of a 8×8 unsigned multiplier on three embedded multipliers at different clock frequencies on a Cyclone III FPGA.



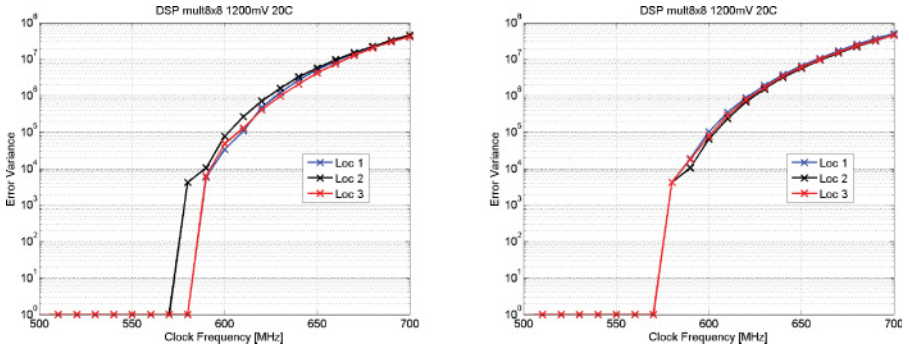
(a) (b)

Fig. 7. Error variance for three embedded multipliers on the same three locations on two different Cyclone III FPGAs.

voltage for the FPGA. Figure 10(b) shows that the designs created by the framework achieve a better PSNR up to 10dB for the same clock frequency, or for similar PSNR, a clock frequency up to 10MHz higher than the reference designs.

4.4. Optimisation Targeting Process Variation

Previously it was shown, through the characterisation, that different embedded multipliers perform differently as a consequence of process variation. To demonstrate that the optimisation performed addresses the impact of variability of the device, in the



(a) (b)

Fig. 8. Error variance for three embedded multipliers on the same three locations on two different Cyclone IV FPGAs.

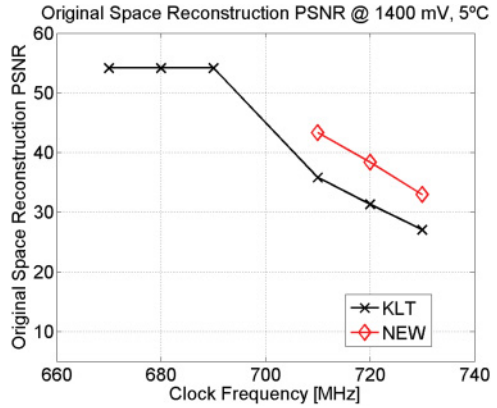
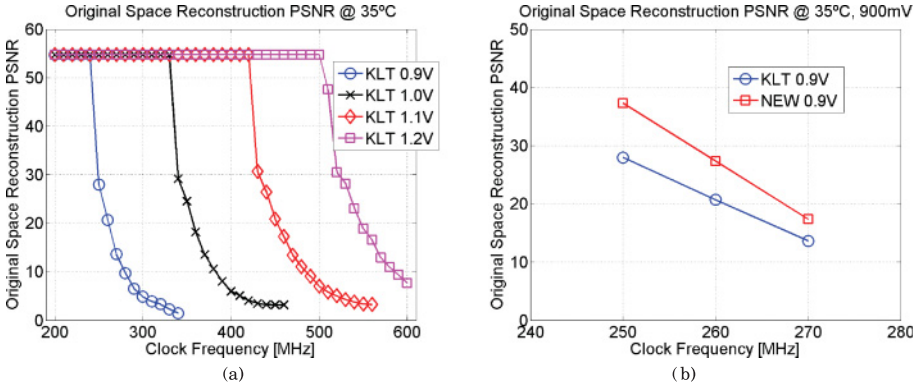


Fig. 9. Comparison of the performance of the two methodologies for the particular case of 1,400mV and 5°C.



(a) (b)

Fig. 10. Performance of the KLT application under different core voltages (a), and a comparison between the two methods for the particular case of 900mV (b).

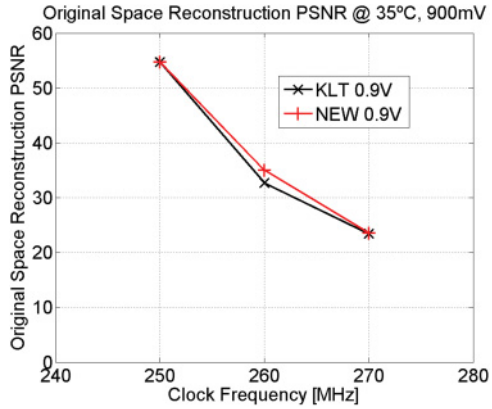


Fig. 11. Performance of the KLT optimised designs for low power, tested on a different Cyclone III FPGA.

design process, a previously optimised, and synthesised, design was placed on a different DE0 board and tested. Figure 11 demonstrates the aforementioned as NEW (design optimised using the characterisation from a different DE0 board) performs similar to the KLT implementation, which has no information about the targeted device. Notwithstanding, all of these designs (KLT and NEW) perform with increased reconstruction PSNR in the second board (Figure 11) than in the first DE0 board (Figure 10(b)). This is due to fact that the second board, on average, has embedded multipliers with smaller delay, consequence of inter-die variation.

4.5. Optimisation Targeting Device Temperature Tolerance

It is well established that temperature increase degrades the performance of silicon devices. Implementing KLT designs without any active cooling components, and operating them in environments prone to large temperature variation can compromise their correct functioning. Usually, if an implementation of the KLT has to consider a wide range of temperatures, then it will have to cope with the worst performance of them. Thus, this places a ceiling on the best reconstruction MSE of the optimised designs even when operating at lower temperatures.

On this account, to enhance the optimisation methodology, it was considered a scenario where the optimisation framework uses a temperature profile for the operating temperatures of the device. This temperature profile detains the temperatures and the percentage of the time the device operates under those temperatures, e.g. 30% at 20°C, 50% at 35°C, and 20% at 50°C

The new idea is focused on sampling KLT designs using the information from a weighted average of the characterisation errors for a range of operating temperatures in the generation of KLT designs. As follows, the prior distribution from (13) is now:

$$g(Err(\lambda_{pk},\, f, P, v, T)) = \sum_i \alpha_i c_E (1 + Err(\lambda_{pk},\, f, P, v, T_i))^{-\beta} \tag{13}$$

Here i iterates over all contributing temperatures ($i = 1, 2, 3$), and $\sum_i \alpha_i = 1$. The different weights represent the significance of the errors at a particular temperature. This particular test case used temperatures 20°C, 35°C, and 50°C and $\alpha_{20} = 0.3$, $\alpha_{35} = 0.5$, and $\alpha_{50} = 0.2$. In practice, the proposed framework generates circuit designs per clock frequency, covering all the temperatures within the expected range. They are identified with NEW WAVG in the results.

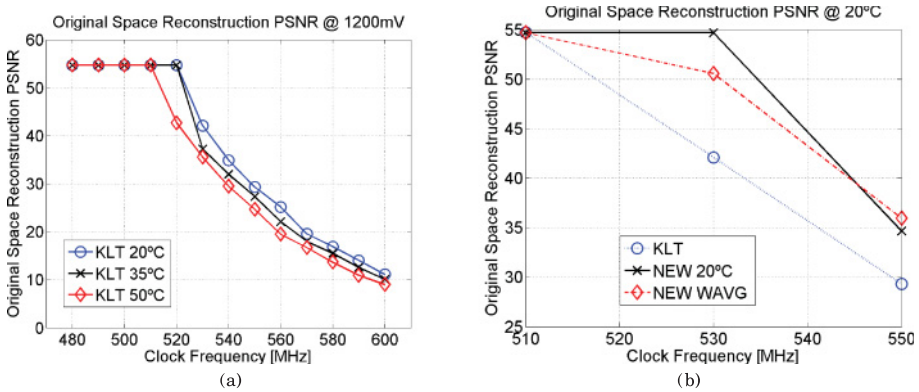


Fig. 12. Performance of the KLT application depending on the temperature of the device (a), and a comparison between the three methods at 20°C (b).

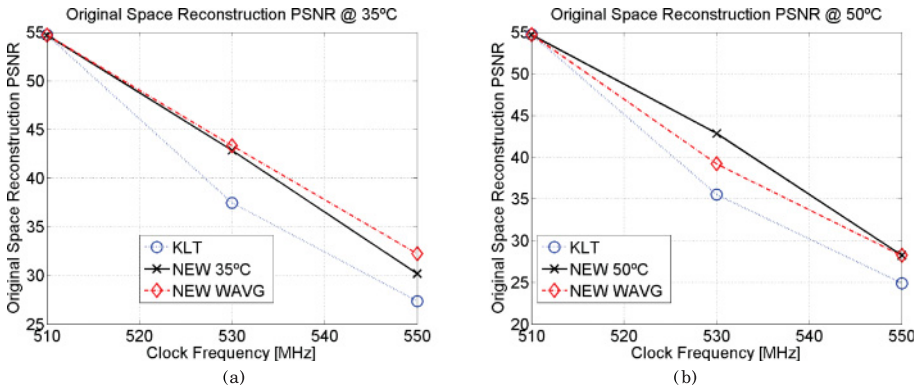


Fig. 13. Performance of the three methods at 35°C (a) and 50°C (b).

Figure 12 (left) shows the dependency of the performance of the reference KLT circuit with the temperature of the device, with a supply voltage of 1,200mV. Figure 12 (right) shows in detail the comparison between the reference and the optimised designs for a specific temperature and a range of temperatures. Figure 13 holds the results for 35°C and 50°C.

These figures show that the designs generated by the framework always outperformed the KLT designs for all temperatures. Furthermore, at 510MHz, the PSNR is more than 10dB better than the KLT design, and at 530MHz, the performance of the NEW design at 35°C is better than the KLT design at 20°C. The NEW WAVG designs perform significantly worst than the NEW ones because they incorporate more information about uncertainty of the results under variation.

This formulation could be used to create models considering many devices from the same family. Such models could be used in High-Level Synthesis to reduce the impact of process variation, avoiding a per-device optimisation.

4.6. Optimisation Process Runtime

The duration, in seconds, of the optimisation process executed by the proposed framework, for each optimisation evaluation of the Z^6 down to Z^3 linear projection, is

approximated by:

$$T_{opti} = (1 + Q(K-1)) \sum_{h=1}^{HPs} \sum_{f=1}^{Freqs} \sum_{wl=WL_{min}}^{WL_{max}} 0.4266e^{(0.6427 \times wl)}, \quad (14)$$

where WL represents the word lengths of the projection coefficients, HPs the number of hyper-parameters considered, $Freqs$ the number of different clock frequencies tested, Q the number of designs generated, and K the size of the projected space. The aforementioned evaluation considered $Q = 1$, $HP = [1, 8]$, $WL = 9$, and $K = 3$.

5. CONCLUSION

This article proposes a novel unified methodology for implementation of extremely over-clocked KLT designs on FPGAs. It couples the problem of data approximation and error minimisation under variation of the operating conditions. It was demonstrated that the proposed methodology optimises KLT designs for performance and resilience simultaneously, by inserting information regarding the performance of the arithmetic units when operating under variation, on a specific device. Results show that the new optimised designs utilise coefficients that produce less errors, when compared to the typical implementation of the KLT algorithm.

REFERENCES

Aim & Thurlby Thandar Instruments. 2013. The New PL-P Series - Advanced Bus Programmable DC Power Supplies. Retrieved from http://www.tti-test.com/products-tti/pdf-brochure/psu-npl-series-8p.pdf.

Altera. 2012. Cyclone III Device Handbook. Retrieved from http://www.altera.co.uk/literature/hb/cyc3/cyclone3_handbook.pdf.

C.-S. Bouganis, I. Pournara, and P. Cheung. 2010. Exploration of heterogeneous FPGAs for mapping linear projection designs. 18, 3 (2010), 436–449. DOI:http://dx.doi.org/10.1109/TVLSI.2009.2012510

C. S. Bouganis, I. Pournara, and P. Y. K. Cheung. 2007. Efficient mapping of dimensionality reduction designs onto heterogeneous FPGAs. In *Proceedings of the 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'07)*. 141–150. DOI:http://dx.doi.org/10.1109/FCCM.2007.50

Jun-Uk Chu, Inhyuk Moon, and Mu seong Mun. 2006. A real-time EMG pattern recognition system based on linear-nonlinear feature projection for a multifunction myoelectric hand. *IEEE Transactions on Biomedical Engineering* 53, 11 (2006), 2232–2239. DOI:http://dx.doi.org/10.1109/TBME.2006.883695

S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw. 2009. RazorII: In situ error detection and correction for PVT and SER tolerance. 44, 1 (2009), 32–48. DOI:http://dx.doi.org/10.1109/JSSC.2008.2007145

R. P. Duarte and C. Bouganis. 2012. High-level linear projection circuit design optimization framework for FPGAs under over-clocking. In *Proceedings of the 2012 22nd International Conference on Field Programmable Logic and Applications (FPL'12)*. 723–726. DOI:http://dx.doi.org/10.1109/FPL.2012.6339162

Rui Policarpo Duarte and Christos-Savvas Bouganis. 2014a. Over-clocking of linear projection designs through device specific optimisations. In *Proceedings of the 21st Reconfigurable Architectures Workshop (RAW'14)*. 9–60.

Rui Policarpo Duarte and Christos-Savvas Bouganis. 2014b. A unified framework for over-clocking linear projections on FPGAs under PVT variation. In *Proceedings of the 2014 10th International Symposium on Applied Reconfigurable Computing (ARC'14)*. 49–60.

Stuart Geman and Donald Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, 6 (Nov. 1984), 721–741. DOI:http://dx.doi.org/10.1109/TPAMI.1984.4767596

Li Ke and Rui Li. 2009. Classification of EEG signals by multi-scale filtering and PCA. In *Proceedings of the International Conference on Intelligent Computing and Intelligent Systems (ICIS'09)*. Vol. 1. 362–366. DOI:http://dx.doi.org/10.1109/ICICISYS.2009.5357825

Lascar Electronics. 2012. EL-USB-TC Thermocouple Data Logger with USB Interface. Retrieved from http://www.lascarelectronics.com/pdf-usb-datalogging/data-logger01585500013493565 70.pdf.

J. M. P. Nascimento and J. M. Bioucas Dias. 2005. Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing* 43, 4 (April 2005), 898–910. DOI:http://dx.doi.org/10.1109/TGRS.2005.844293

H. T. Ngo, R. Gottumukkal, and V. K. Asari. 2005. A flexible and efficient hardware architecture for real-time face recognition based on eigenface. In *Proceedings of the 2005 IEEE Computer Society Annual Symposium on VLSI*. 280–281. DOI: http://dx.doi.org/10.1109/ISVLSI.2005.5

Pete Sedcole and Peter Y. K. Cheung. 2008. Parametric yield modeling and simulations of FPGA circuits considering within-die delay variations. *ACM Transactions on Reconfigurable Technology and Systems* 1, 2, Article 10 (June 2008), 28 pages. DOI: http://dx.doi.org/10.1145/1371579.1371582

Terasic Technologies. 2009. Terasic DE0 Board User Manual v. 1.3. Retrieved from http://www.terasic.com.tw.

J. S. J. Wong, P. Sedcole, and P. Y. K. Cheung. 2008. A transition probability based delay measurement method for arbitrary circuits on FPGAs. In *Proceedings of the International Conference on ICECE Technology (FPT'08)*. 105–112. DOI: http://dx.doi.org/10.1109/FPT.2008.4762372

Conrad Ziesler, David Blaauw, Todd Austin, Krisztian Flautner, and Trevor Mudge. 2003. Razor: A Low-Power Pipeline Based on Circuit-Level Timing Speculation. (2003).

Received June 2014; revised May 2015; accepted August 2015