2014 IEEE 28th International Parallel & Distributed Processing Symposium Workshops

Over-Clocking of Linear Projection Designs Through Device Specific Optimisations

Rui Policarpo Duarte
Department of Electrical and Electronic Engineering
Imperial College London
United Kingdom
Email: R.Duarte09@imperial.ac.uk

Christos-Savvas Bouganis
Department of Electrical and Electronic Engineering
Imperial College London
United Kingdom
Email: Christos-Savvas.Bouganis@imperial.ac.uk

***Abstract*—Frequently, applications such as image and video processing rely on implementations of the Linear Projection algorithm with high throughput and low latency requirements. This work presents a framework to optimise Linear Projection designs that excel typical design implementations via a pre-characterisation of over-clocked arithmetic units. It is well known that the delay models used by synthesis tools are generic and tuned for the worst performance possible of a given fabrication process. Hence, they impose a heavy penalty in the possible maximum performance offered by the fabrication process. The proposed optimisation framework focuses on the optimisation of the generic multipliers, as they are the arithmetic operators with the most critical paths in the data path of a linear projection design, by performing a performance characterisation step on the target device. Experiments demonstrate that the proposed framework is able to generate Linear Projection designs that achieve higher throughput (up to 1.85 times) while producing less errors than typical implementation methodologies.**

I. Introduction

In recent years, Digital Signal Processing (DSP) engineers designing circuits for applications with high throughput, low latency and low power constraints consider Field-Programmable Gate Arrays (FPGAs) as an alternative technology to standard DSP processors due to their key characteristics. Modern FPGA devices offer highly specialised blocks, such as embedded DSP blocks and distributed memory blocks that are highly utilised by DSP applications, where their reconfigurability property allows for fast prototyping avoiding the fabrication of specialised chips which is an expensive and lengthy process.

Common strategies to maximise the performance of designs targeting high throughput requirements include word-length optimisation [1], quantisation, pipelining, loop unrolling, and retiming techniques just to name a few [2]. All of these techniques assume constant performance *intra-die* and *inter-die*, making use of a high-level performance model that characterises the whole family of the targeted FPGA device.

However, nowadays, the continuous scaling of the fabrication process has led to devices with increased performance characteristics by supporting higher clock frequencies with less power consumption, but also to an increased variation in the performance characteristics for *intra-die* and *inter-die* [3]. This process variation affects the characteristics of transistors on a device, changing physical dimensions, altering the timing threshold and finally affecting their overall performance. This leads to devices that exhibit uneven performance across its area and as the technology continues to scale down, the fabricated devices will be even more susceptible to such variations. As such, modern devices are no longer limited by their process technology performance, but by the performance of their worst transistor.

To ensure that the implemented designs operate without errors once placed on an FPGA device, the synthesis tools use conservative models to determine the maximum error-free performance of circuits for a family of devices. As a consequence, there is a significant gap between the performance that can be achieved as dictated by the models used within the synthesis tools and what actually can be achieved by the actual device where the circuit will be placed on.

The work presented here aims to close this performance gap, establishing a per device optimisation concept, allowing the design of Linear Projection designs to exploit extra performance capabilities from arithmetic operators on a specific FPGA device. The key enabler to the success of such Optimisation Framework (OF) is the reconfigurability property of FPGA devices. Thus, a device characterisation step is performed, resulting in the collection of information that is utilised in the high-level specification of the design in the targeted system, leading to a final performance closer to what is achievable by the targeted FPGA device rather than what is reported by the conservative models of the synthesis tools.

Moreover, motivated by the fact that Linear Projections aren't critical to errors in many parts of their designs, the work presented here operates beyond the error-free limits of the arithmetic operators, and considers also the region of operation where errors appear in the data path allowing the exploitation of the trade-off between performance and error-prone calculations pushing even further the achieved performance of the system.

Figure 1 illustrates the proposed concept when a multiplier is clocked for different clock frequencies. The maximum performance of the operator as reported by the synthe-

978-1-4799-4116-2/14 $31.00 © 2014 IEEE
DOI 10.1109/IPDPSW.2014.25

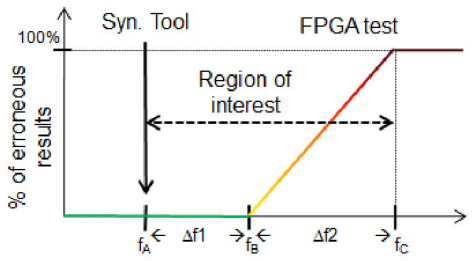

Figure 1. Percentage of erroneous results at the output of a generic multiplier *vs* its clock frequency. The error-free ($\Delta f1$) and error-prone ($\Delta f2$) regimes are depicted as well as the conservative operational limit imposed by the synthesis tool (f_A).

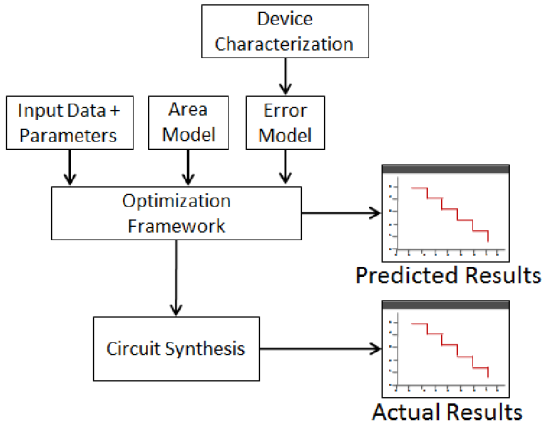

Figure 2. Design flow using the proposed optimisation framework.

sis tools (f_A) is illustrated along with the operational regions where the module can operate in a specific device under error-free ($\Delta f1$) and under error-prone regimes ($\Delta f2$). f_B and f_C represent the clock frequencies up to which the design operates on the FPGA without errors and with errors, respectively. Above f_C the design doesn't produce meaningful results.

A novel methodology here presented exposes the obtained performance characterisation information of generic multipliers to a high-level design of a Linear Projection system, targeting to maximise its throughput while minimising errors. The focus of this work is on Look-Up Table (LUT)-based generic multipliers, as they allow optimisations over the FPGA area taken by the design. However, the proposed framework can be easily extended to accommodate embedded DSP blocks currently available in modern FPGAs. Figure 2 depicts the design flow of the proposed optimisation framework. A key step of this framework is the performance characterisation of the generic multipliers in a given device when they are operated with clock frequencies well above from what is reported by the synthesis tool. The obtained performance information (i.e. errors that are expected at the output of the multipliers) as well as information regarding the estimated area of the generic multiplier are injected to a Bayesian formulation of the problem in order to obtain a high-level specification of the design to implement. This formulation is capable of producing Linear Projection designs that avoid, or minimise, over-clocking errors outperforming typical implementations in terms of trade-off between performance and errors.

More specifically, the paper makes the following two key contributions:

- It presents a framework for the performance characterisation of generic multipliers based on LUTs. The proposed framework characterises the error at the output of such multiplier component when it is clocked under various clock frequencies and placed in various locations across the device.
- It presents a framework that utilises such information for the optimisation of a specific DSP application namely a Linear Projection system. A novel approach is proposed for the design an optimisation of such application taking into account information regarding the performance of generic multipliers even when they are clocked beyond the clock frequency reported by the synthesis tool. As a result, a significant performance boost is achieved while minimising errors.

The presented work deviates significantly from other works in the field of FPGA design optimisations as it considers device specific performance characteristics and demonstrates how such information can be utilised during the design and optimisation of a Linear Projection system. In addition, even though increased performance gives rise to increased power consumption, this work assumes that the power budget isn't a limitation, therefore no power measurements are reported.

The rest of the paper is structured as follows. Section II provides a brief elaboration on the existent techniques that aim to mitigate performance variation across a device or across a family of devices. Section III presents details on the performance characterisation framework that is applied to capture the characteristics of a generic multiplier on a target FPGA device. Section IV introduces the Linear Projection design, which is the design under consideration. Section V demonstrates how the obtained information from the performance characterisation of a generic multiplier can be utilised in the high level design and optimisation of the Linear Projection digital circuit, where Section VI evaluates the performance of the proposed optimisation framework. Section VII draws conclusions on the proposed work and includes remarks on future work.

II. Background

The impact to the overall performance of the device due to process variation within the device and across the devices of the same family has motivated several works in the design

optimisation of circuits that target FPGAs. Most of the works propose frameworks that aim to explore such variability in order to obtain the maximum performance of a design when it is mapped to a specific FPGA device, where only few of them aim to allow errors in parts of the design in order to push the performance limit even further.

One of the most well known techniques that can be applied to address the problem of performance variability within a device is Razor [4]. It is a generic time-redundant method which can be applied to any path prone to errors due to timing violations, where the recovery is performed at the expense of extra latency. However, even though this is a generic technique, it does not "hide" the performance variability in the design as the designer needs to consider in the design process the impact of the extra latency in the functionality and performance of the targeted application.

In [5] the authors 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.

In [6], the authors demonstrate online and off-line techniques for characterising variation and degradation of the FPGA fabric without the need for dedicated hardware support. Furthermore, they present a method to overcome process variation by tuning applications to fit delay patterns measured in a clustered batch of the device. Within the same work, a group of techniques that preempt or adapt to transistor degradation by making changes to the device configuration are reported and evaluated.

All the works above do not consider the design and optimisation of the algorithm that is implemented by the specific circuit, therefore they are context-unaware. Their aim is to map the given design to the targeted device without allowing any timing violations in the design, avoiding the production of erroneous data. However, significant performance gains can be obtained when such low level information i.e. the performance of the circuit (frequency, error at the output, etc) are taken into account in the high-level design/optimisation of the algorithm. Towards this direction, [7] pioneered the idea of collecting performance information from an off-line characterisation of a specific device for a specific functionality, and utilising the information for the design and optimisation of a Linear Projection system. The work considered only the characterisation/utilisation of Constant Coefficient Multipliers (CCMs), which does not allow their system to scale to large problems.

Contrary to the previous approach of collecting actual performance data from the targeted FPGA device, [8] considers the approach of developing models for errors in a data path that are generated due to operating the device beyond from what is reported by the synthesis tool. The authors conclude that in certain applications (i.e. image processing) it may be beneficial to allow for timing violations to occur, and thus errors to appear at the output of the system, but with a significance gain in the overall performance due to higher clocking frequency.

The present work borrows the Baesyan formulation from from [9] resembling the contribution in [7], but deviates from it in many aspects. The presented work lifts the limitations imposed by the CCMs by considering generic multipliers components leading to the applicability of the framework to real large scale applications. As such, by reducing the number of circuits, a significant speed up of the performance characterisation step is obtained leading to a better characterisation of the device under a specific module for a given time frame. Moreover, this work presents a performance characterisation framework targeting a generic multiplier, by addressing the specific challenges imposed by the structure of such component, which differs from the case of CCMs. Also a new optimisation framework that utilises such information for a Linear Projection design optimisation is presented, including a new algorithm for design space exploration that utilises an objective function tuned for the utilisation of generic multipliers within that Linear Projection framework.

Notwithstanding FPGA vendors state conservative timing parameters because of device aging effects, the reconfigurability offered by FPGAs allows future re-characterisations of the device and update the design to compensate slow degradation over the years.

III. Multiplier Characterisation Framework

A. Introduction

A key element of the presented work is the performance characterisation circuit of a generic multiplier under various word-length supports, locations, clock frequencies, and placement and routing configurations on an FPGA. Such approach is possible only due to the reconfigurability that is offered by FPGA devices. A basic framework for the characterisation of CCMs under over-clocking was introduced in previous work [7]. The new framework has been created to support the characterisation of more complex arithmetic operators in many locations simultaneously.

It is worth pointing out that the supporting modules are independent from the design under test, which in this case is a generic multiplier, and thus the proposed framework can be utilised for other arithmetic components. As such, a characterisation of the design under test can be achieved and the collected information can be utilised by other design frameworks.

B. Architecture and Characterisation Process

A schematic of the architecture of the characterisation circuit is shown in Figure 3. It is composed by the following modules: "input stream" Block Random Access Mememory (BRAM), the multiplier under characterisation, the "output

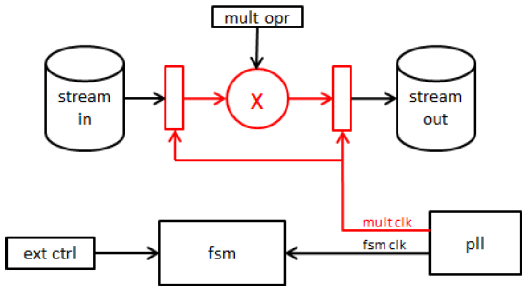

Figure 3. Architecture of the multiplier characterisation circuit.

stream" BRAM, a Phase-Locked Loop (PLL) to set the clock frequencies and the Finite State Machine (FSM) to control the circuit.

The PLL generates the clock signals for two clock domains: the "mult clk" clock drives the design under test (i.e. generic multiplier), where the "fsm clk" drives the FSM, BRAMs, and the other supporting modules. It is necessary to mention that special care has been given to the design of BRAMs interface and to the rest of the supportive modules to ensure that the critical path is always within the design under test. Thus, the clock frequency limit of the supportive modules is well above the region where the design under test generates erroneous results, for the framework to be able to operate without introducing errors.

The whole characterisation process has been automated and integrated in the proposed optimisation framework. The process is initiated by the host computer with the configuration of the FPGA, the transfer of the data from and to the FPGA device takes place through the Joint Test Action Group (JTAG) interface. Once the characterisation process completes, the results are retrieved by the host computer.

C. Results

In this work, the Cyclone III device from Altera hosted in a DE0 board is used. The developed framework is very lean regarding its requirements and can be executed to any available board/device with some small changes. As the target is to utilise such generic multipliers for a Linear Projection design that has been pushed to its performance limits through over-clocking, in the characterisation of the generic multipliers one of the inputs was enumerated through all possible values, (i.e. finite number of values for a given word-length), where the other input was stimulated by a random steam of input data following a uniform distribution. In order to reduce variation due to operating conditions, the temperature of the device was kept constant at 14 degrees Celsius through the use of a cooling element on top of the FPGA. The multipliers were synthesised multiple times at multiple locations in the device, to contain the performance of generic multipliers with different placements and routings.

The obtained results demonstrate that as the clock frequency increases, more erroneous data appears at the output of the multipliers, demonstrating that the presence of errors is cumulative as the clock frequency increases, which is expected. Moreover, the test results show that at high frequencies there is a variation on the observed errors at the output every time the performance characterisation test is executed, something that is attributed to clock jitter.

Also, it was observed that placement of the performance characterisation circuit at various locations of the device gives rise to different patterns for errors at the output of the multiplier. However, this effect is not only attributed to the performance difference in the fabric of the FPGA device, but also due to the different routing that the design under test exhibits.

Figure 4 shows the error at the output of a generic 8x8 unsigned multiplier for the first 100 values from a performance characterisation of a total of $29,400$ values, when it is placed at two different locations in the device (i.e. loc 1 and loc 2). The target clock frequency is set at 320 MHz and one of the operands of the generic multipliers was fixed to a value of 222. The error histograms for the two locations are depicted in the same figure, for the whole test. Important to note that the high error values are expected as the generic multipliers are constructed though LUTs and the Most Significant Bits (MSbs) exhibit the longest paths.

In summary, a framework has been proposed that characterises the performance of a design, in this case a generic multiplier, under various settings (i.e. clock frequency, location on the device, etc.). This framework is scalable considering that it supports many units under characterisation simultaneously, and can be extended to support an automatic framework for performance characterisation of a design under other controllable parameters like voltage and temperature allowing the user to obtain the behaviour of a design under different sets of conditions.

The rest of the paper focuses on how such information can be utilised for the design and optimisation of a DSP design, namely a Linear Projection design that makes heavily use of generic multipliers, allowing the user to clock the design beyond the clock frequency reported by the synthesis tool in a way that minimises the manifested errors, but at the same time improving the overall latency/throughput.

IV. Over-Clocking a Linear Projection Circuit

One of the most widely used dimensionality reduction techniques is the linear projection based on Principal Component Analysis (PCA) or Karhunen-Loeve Transformation (KLT) transform. Such techniques aim to find a low dimension space that captures the main modes of variation of the original high dimensional data from where certain features can be extracted. A large number of applications can be found in computer vision, image processing, and in general in the signal processing arena. When real-time requirements are imposed, a hardware solution is often desired considering

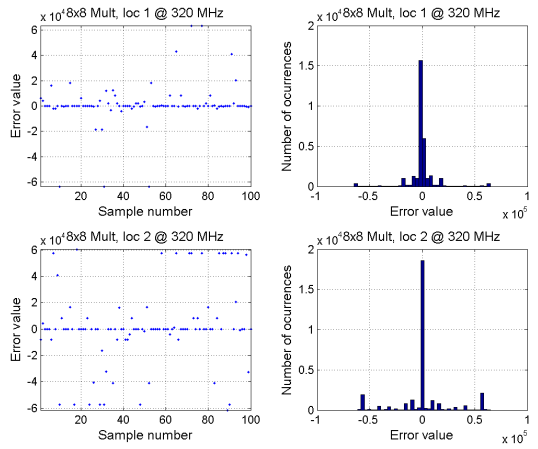

Figure 4. Errors from a 8x8 bit multiplier, for constant multiplicand 222 in 2 locations of the FPGA at 320 MHz.

the trade off between required area, achieved throughput, and reconstruction error.

In this work, the design and optimisation of such Linear Projection circuit for an FPGA is considered, aiming at the same time to push its performance beyond what is reported by the synthesis tool, in order to achieve higher performance than what is predicted by the synthesis tool. It is important to remember that the proposed optimisation is possible as the Linear Projection circuit allows by definition certain errors at the output of the system, which is also what is explored by the proposed framework in order to "hide" errors due to over-clocking.

A. Background

The linear projection, also known as KLT, or 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 (1), written in matrix notation, where $X = [x^1, x^2, ..., x^N]$ and $F = [f^1, f^2, ..., f^N]$, where $f^i \in R^K$ denote the factor coefficients.

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

The original data can be recovered from the lower dimensional space via (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 squared error 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.

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

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

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

The calculation of the Λ matrix 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 [10], [9]. The Bayesian formulation presented in [9] 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 linear projection designs which minimise errors and circuit resources, when compared to the standard approach of the KLT transform application followed by the mapping to the FPGA.

The key idea in [9] is to inject information about the hardware (i.e. in this case about the required hardware resources of a constant coefficient multiplier) as a prior knowledge in the Bayesian formulation of the above optimisation problem.

In more detail, the proposed framework in [9] estimates the basis matrix Λ, the noise covariance Ψ, and the factors using Gibbs sampling algorithm [11] 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 for generation of samples, where the prior distribution tunes this posterior distribution, and thus accommodating the impact the required hardware resources.

[7] provides an extension of the above work for the optimisation of Linear Projection designs using CCMs combating the effects for circuit area as well as performance variation due to over-clocking. This work generalises on the previous work by supporting the use of generic multipliers in the design and optimisation of linear projection circuits under over-clocking. The rest of the paper presents how the challenges raised by the above transition have been addressed.

V. Linear Projection Optimisation Framework Targeting Generic Multipliers

The proposed optimisation framework for Linear Projection designs targets the utilisation of LUT-based generic multipliers for the implementation of (1), along with the coefficients of the Λ matrix that define the lower dimension space. The objective of the framework is to optimise the word-length of these multipliers in order to boost the performance of the overall system. The key characteristic of

this work is that the obtained circuit will be over-clocked to frequencies beyond those reported by the synthesis tool in order to push further its overall performance. Such scenario can be faced when there are hard area requirements and unrolling of the design is not possible, or when applications with high dimensions (i.e. face recognition) are addressed.

A. Objective Function

The objective function utilised in the proposed optimisation framework is formed by the mean squared error of the reconstructed data in the original space, and errors that are produced due to the over-clocking of the utilised generic multipliers. The objective of the proposed optimisation framework is to minimise the objective function by sampling the Λ matrix from its posterior distribution.

Let's denote with $\hat{X}$ the result of the reconstruction of the projected data in a matrix form. Then, the objective function T is defined as in (5), where both reconstruction errors are errors due to over-clocking are captured. E denotes the expectation, where operator tr is used as the matrix formulation is utilised.

$$T = tr\{E[(X-\hat{X})^T(X-\hat{X})]\} \quad (5)$$

By expressing the reconstructed data as a function of the Λ matrix, and the error due to over-clocking with ε such as $\hat{X} = \Lambda(F+\varepsilon)$, the objective function is expressed as:

$$\begin{aligned} T &= tr\{E[(X-\Lambda(F+\varepsilon))^T(X-\Lambda(F+\varepsilon))]\} \\ &= tr\{E[((X-\Lambda F)^T-(\Lambda\varepsilon)^T)((X-\Lambda F)-(\Lambda\varepsilon))]\} \\ &= tr\{E[((X-\Lambda F)^T(X-\Lambda F)- \\ &\quad -(X-\Lambda F)^T(\Lambda\varepsilon)-(\Lambda\varepsilon)^T(X-\Lambda F))+ \\ &\quad +(\Lambda\varepsilon)^T(\Lambda\varepsilon)]\} \end{aligned}$$

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 becomes as follows:

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

where 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 linear projection without any over-clocking errors, where the second term relates to the variance of the errors at the output of the generic multipliers due to over-clocking. Thus, the errors due to dimensionality reduction and the over-clocking of the hardware are captured by one objective function without any need to formulate a problem using a multi-objective function.

B. Prior Distribution Formation

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

1) Over-Clocking Error Model: The proposed framework utilises the performance characterisation framework for the multipliers described before. By executing that framework, a profile of the errors expected at the output of the generic multipliers when one of the operands is fixed (i.e. representing a coefficient of the Λ matrix) for various frequencies 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 which models the uncertainty of the result. As such, a data structure is formed, $E(m, f)$, that holds information regarding the variance at the output of a multiplier when a stream of data is multiplied by a constant m and the circuit is clocked at frequency f. Hence, the error model reflects the characteristics of the data in the problem considered, in preference of an excessive characterisation that ensures all critical paths are exercised even though it doesn't introduce any additional benefit.

Figure 5 depicts the variance of the error at the output of an 8x8 bit multiplier for all multiplicands at different clock frequencies when they are stimulated with a pseudo-random sequence of data, obtained by running the performance characterisation circuit on a Altera Cyclone III 3C16 device. For each region, defined by the pair multiplicand-clock frequency, the shade represents the amount of variation the multiplier exhibits. As expected, it is noticeable that multiplicands with few '1' bits in their binary representation have less over-clocking errors.

2) Area Model: The proposed framework utilises area models for the estimation of the area of the designs without having to synthesise them, accelerating as such the design-space exploration process. The information about the area required to implement a LUT based generic multiplier of a specific word-length is based on information extracted from the synthesis tools when a specific device is targeted. This is possible due to the finite number of word-lengths that are considered. The overall area of the design is estimated through a high-level model. Figure 6 shows the obtained synthesis results regarding the word-length supported by the LUT-based generic multiplier and its required area for multiple placement and synthesis steps.

3) Prior Distribution Formation: The formation of the prior distribution $p(\cdot)$ of the Λ matrix is a key part of the optimisation framework as it injects hardware information to the optimisation framework for the estimation of the Λ matrix. The aim of the prior distribution is to penalise Λ matrix instances with high errors, due to the use of coefficients that generate high errors due to over-clocking or

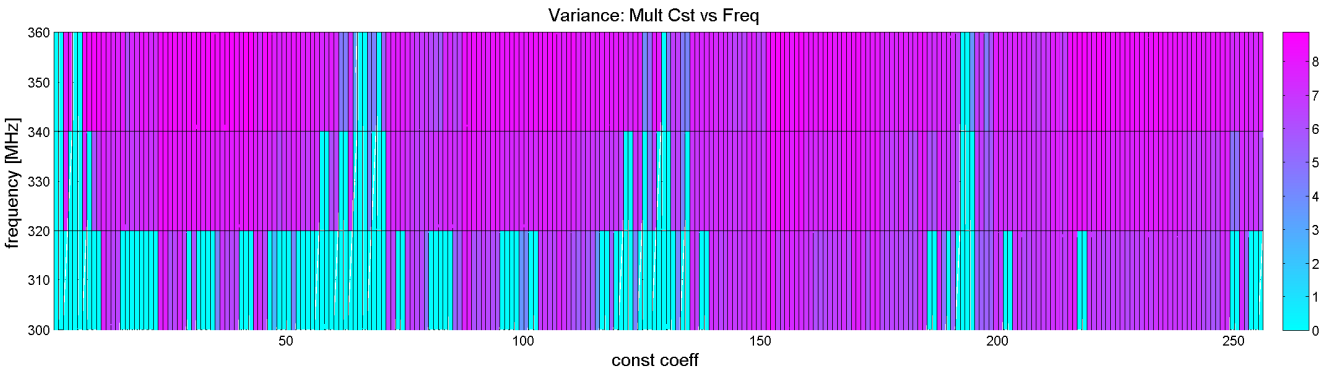

Figure 5. Error model data structure of a 8x8 bit Multiplier.

Figure 6. Example of the data collected to generate the area model for generic multipliers with different word-lengths across many locations on the FPGA.

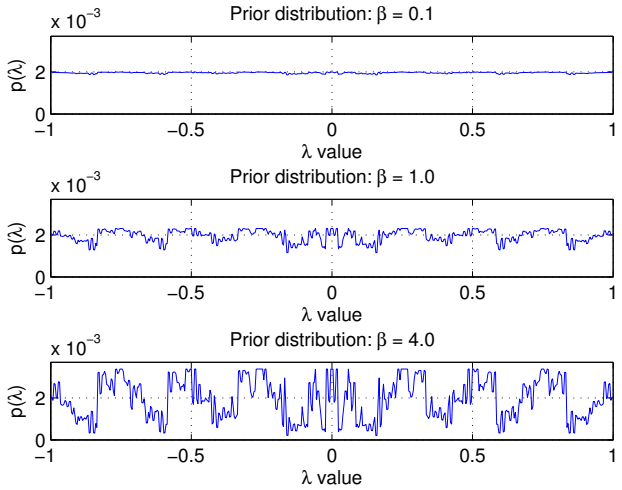

Figure 7. Prior distribution for $\beta = [0.1, 1.0, 4.0]$ for 8 bit word-length multipliers at 340 MHz.

due to poor description of the original space, 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 $p(\lambda_{pk}, freq) = g(E(\lambda_{pk}, freq))$, where the performance of every coefficient in the Λ matrix is dictated by the targeted clock frequency, 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.

$$g(E(\lambda_{pk}, freq)) = c_E(1 + E(\lambda_{pk}, freq))^{-\beta} \quad (6)$$

c_E is a constant used to ensure that $\sum_{\lambda_{pk}} g(E(\lambda_{pk}, freq)) = 1$ for a given clock frequency $freq$. β is a *Hyper-Parameter* that allows scale of the contribution of over-clocking errors in the prior distribution. $E(\lambda_{pk}, freq)$ is the variance of the error observed from the performance characterisation of the multiplier for a given λ_{pk} value tested at $freq$ clock frequency.

In figure 7 there is an example of the prior distributions for 3 different values of β for the same multiplier characterisation at a 310 MHz. The figure shows that for $\beta = 0.1$ all λ_{pk} values will have almost the same probability of being sampled, whereas for $\beta = 4.0$, λ_{pk} values with high over-clocking errors have low probability of being sampled.

C. Design Generation

The proposed optimisation framework generates a number of designs that minimise the selected objective function T for a given FPGA area due to its sampling process. The resulting designs are the ones that fall on the Pareto curve. 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 $freq$, the β parameter, and an internal optimisation parameter of the system that is the number of designs to be passed form

one iteration to another Q. The pseudo-code of the algorithm is given in Alg. 1.

Algorithm 1 Linear Projection Design optimisation framework

Require: $Q \geq 1 \wedge K \geq 1 \wedge \beta > 0 \wedge freq > 0$
Ensure: Q designs
  $X \leftarrow input$ {original data N cases}
  for $d = 1$ **to** K **do**
    Create new empty $Candidate\_Projs$ list
    for $wl = wl_{min}$ **to** wl_{max} **do**
      $prior \leftarrow generate\_prior(wl, \beta, freq)$
      $\lambda_{d,wl} \leftarrow sample\_projection(X, prior, wl)$
      $area_{d,wl} \leftarrow estimate\_area(\lambda_d, wl)$
      $F \leftarrow (\lambda_d^T \lambda_d)^{-1} \lambda_d^T X$
      $error \leftarrow X - \sum_{j=1}^{d} \lambda_{j,wl} F$
      $MSE_{d,wl} \leftarrow \sum \sum error^2 / PN$
      $Proj \leftarrow (\lambda_{d,wl}, area_{d,wl}, MSE_{d,wl})$
      Add $Proj$ to $Candidate\_Projs$ list
    end for
    Extract candidate projections in the Pareto points {min MSE for a given area}
    Create Q bins $\in (MSE_{min}, MSE_{max})$
    Extract 1 candidate projection, with the least MSE, from each bin
    Create new Q candidate projections from the Q extracted
  end for
  Create Q designs from the extracted Q projections
  return Q linear projection designs

VI. Performance Evaluation

A Z^6 to Z^3 linear projection case study is conducted to evaluate the performance of the proposed optimisation framework. The incentive to present a small problem relies on the fact that, when using generic multipliers, the regularity of the architecture to implement a linear projection design makes its performance independent of the dimensions of the problem. Moreover, given the novelty of the methodology proposed, it is intentional to keep the problem as simple as possible to demonstrate the impact of the proposed methodology.

To evaluate the performance of the proposed optimisation framework in the generation of circuit designs for dimensionality reduction problems based on linear projection, a reference design is implemented that relies on the KLT transformation, considering different word-lengths, modeling the existing approach to the above problem.

The performance evaluation of the proposed work is done in three domains, namely the *predicted*, *simulated* and *actual*. The *predicted* performance is the performance expected by the designs generated by the proposed optimisation framework using the described error model. As the error model is obtained by a performance characterisation step, it is expected that a certain deviation from running the design on the actual FPGA device to be observed. The latter is the *actual* performance.

Parameter	Values
P	6
K	3
Characterization	4900 cases
OF training	100 cases
Test	5000 cases
β	4.0, 8.0
Q	5
Clock Frequency	310 MHz
Input data word-length	9 bits
λ_{pk} word-length	3 to 9 bits
Burn-in period	1000 samples
Projection vector	3000 samples

Table I
SETTINGS USED IN THE CASE STUDY.

The *simulated* performance is the performance of a design under simulation using the same data as the ones that are utilised for the *actual* performance evaluation of the design in the device. This intermediate result provides an insight of the quality of the error model due to over-clocking. Deviations between this performance characterisation and the *actual* performance of the designs on the device still exist due to placement and routing variation.

In addition, the word-length of the projection vectors was kept between 3 and 9 bits to assess the trade-off between precision, performance and errors undertaken by the proposed optimisation framework. Other elements present on the FPGA, namely embedded multipliers, perform multiplications with large word-lengths faster, but they are out of scope of the present work.

The generated designs are evaluated in terms of reconstruction mean squared error, required hardware resources, and their operating clock frequency. All tests used a DE0 board from Terasic equipped with a Cyclone III 3C16 FPGA device from Altera. Table VI summarises the parameters used and their values.

A. Performance Limits

The performance limitations of the generated Linear Projections designs following the existing methodology of applying the KLT transform and then mapping the design on an FPGA are depicted in Figure 8. The design space exploration is performed along the word-length employed by the generic multipliers, which corresponds to the quantisation step of the design process. The figure shows the maximum clock frequency reported by the synthesis tool (Tool Fmax - green), the data-path maximum frequency when the design is placed to the targeted FPGA without observing any errors in the calculations in the data path (Data-path Fmax - yellow), and the range of the frequencies where the design starts generating errors due to over-clocking (FSM Fmax - red).

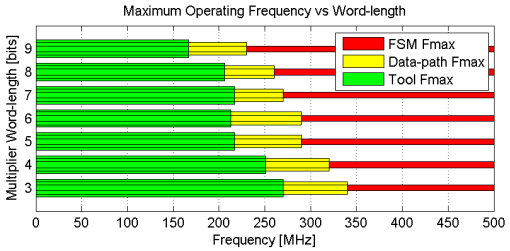

Figure 8. Maximum clock frequencies *vs* word-length for a Z^6 to Z^3 linear projection circuit designed by the KLT transform.

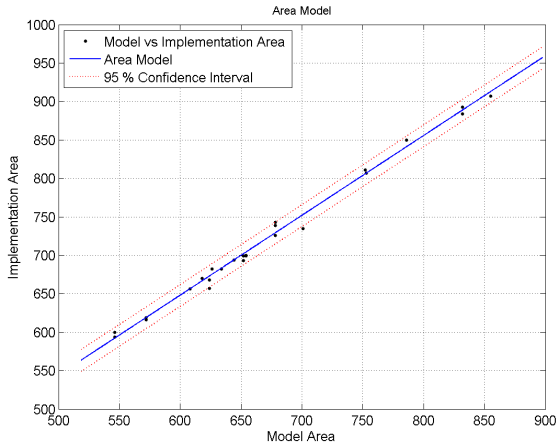

Figure 9. Evaluation of the area model against the actual circuit area.

B. Area Model Evaluation

The area model is created to give a quick, but accurate, estimate on the resources taken by a candidate design without having to actually synthesise it. Area is measured in terms of number of Logic Elements (LEs) required to implement the circuit. This estimate is based on the resources reported by the synthesis tool for each Multiply-Accumulate (MAC) block in the characterisation circuit, however small deviations are expected due to further optimisations performed by the synthesis tool. Figure 9 shows the relation between the predicted and the actual resources occupied on the FPGA. The figure also shows that most of the data points fall inside the 95% confidence interval for the area estimation.

C. Error Model Evaluation

This subsection evaluates the validity and impact of the introduced error model. Figure 10 depicts design points under *predicted*, *simulated* and *actual* performances. It should be noted that all area results refer to the actual area utilised by the design. For designs with small area, the simulation and actual results are very close, as the variation due to placement and routing is minimum. As the designs become larger, an increased discrepancy in the performance is observed. Also, the obtained results show that the *predicted* performance follow better the *simulated* and *actual* performance for larger designs, as small modeling

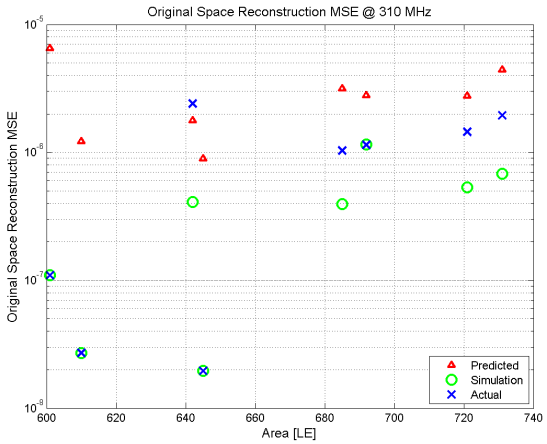

Figure 10. *Predicted*, *simulated* and *actual* performance reconstruction vs. area of the Linear projection designs produced by the proposed optimisation framework. The target clock frequency is 310MHz.

errors in the performance characterisation of the multipliers are more evident in smaller designs. Thus, the validity of the utilised error model is confirmed.

D. Linear Projection Targeting a Specific Clock Frequency

The performance of the proposed optimisation framework is evaluated against the existing design methodology (i.e. KLT) where no information about the errors due to over-clocking of the design is taken into account. The target clock frequency is set to 310 MHz, which is 1.85 times higher than the possible frequency reported by the synthesis tool for a KLT design employing 9 bits coefficient word-length. At this clock frequency some KLT-based designs (i.e. the ones with large area footprint) will operate with errors in their data path (Figure 8).

Figure 11 shows the actual performance of the designs produced by the proposed optimisation framework and the KLT approach, along with their predicted performances. The predicted performance for the KLT-based designs is based on the extension of the existing methodology and the adoption of the objective function T. However, no optimisation with respect to over-clocking characterisation has been performed in the KLT-based designs. The results show that the proposed optimisation framework produces designs that behave as expected under over-clocking, as well as they produce around an order of magnitude on average lower reconstruction error for the same area.

E. Run-Time Investigation

The run-time requirements of the proposed optimisation framework has also been investigated, when the proposed framework is executed in a Intel Core-i7 processor. A model that provides the run-time (i.e. in seconds) of the optimisation framework under difference settings is provided

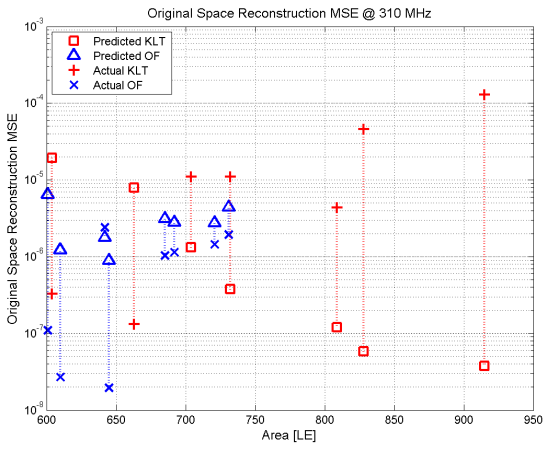

Figure 11. MSE for the reconstruction of the projected data in the original space at 310 MHz. The design points of the KLT correspond to 3-9 bit coefficient word-length.

below.

$$Time = (1+Q(K-1)) \sum_{1}^{\#HP} \sum_{1}^{\#Freqs} \sum_{1}^{\#wl} R(wl) \quad (7)$$

$$R(wl) = 0.4266 \times exp^{(0.6427 \times wl)} \quad (8)$$

Equation (8) models the time to sample one projection vector of a given word-length, where (7) models the required time to sample a complete set of designs for a given number of clock frequencies ($\#Freqs$), projected dimensions (K), maintained designs (Q), values of *Hyper-Parameter* β ($\#HP$), and word-lengths (wl). The results for both equations are in seconds. As an example, the execution of the proposed optimisation framework using ($\#Freqs = 1$, $K = 3$, $Q = 5$, $\#HP = 2$, $wl = [3..9]$), the processing time is 1 hour and 44 minutes, which is considered acceptable for design an optimisation of digital circuits.

VII. Conclusions and Future Work

This work proposes a novel approach for acceleration of Linear Projections by introducing the idea of device specific performance characterisation to address the impact of variation. As such, a framework that characterises the performance of generic multipliers on a specific FPGA device is proposed and described. Moreover, a novel approach is introduced for the utilisation of such information for the design and optimisation of a Linear Projection circuit design. The work shows that high performance improvements can be achieved when considering such device oriented optimisations, specific to FPGA devices due to their reconfigurability properties, that are not possible through the available synthesis tools. Future work envisages applying similar methodology to improve power efficiency by lowering the voltage and tolerating the associated increase in errors.

Acknowledgments

RPD would like to acknowledge Fundação para a Ciência e Tecnologia (Foundation for Science and Technology in Portugal) for their support through PhD grant SFRH/BD/69587.

References

[1] G. A. Constantinides, P. Y. K. Cheung, and W. Luk, "Wordlength optimization for linear digital signal processing," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 10, pp. 1432–1442, 2003.

[2] K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation*. Wiley, 1999.

[3] E. A. Stott, J. S. Wong, N. P. Sedcole, and P. Y. K. Cheung, "Degradation in fpgas: measurement and modelling," in *FPGA*, 2010, pp. 229–238.

[4] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: a low-power pipeline based on circuit-level timing speculation," in *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*, Dec 2003, pp. 7–18.

[5] P. Sedcole and P. Y. K. Cheung, "Parametric yield modeling and simulations of FPGA circuits considering within-die delay variations," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 1, no. 2, pp. 10:1–10:28, Jun. 2008. [Online]. Available: http://doi.acm.org/10.1145/1371579.1371582

[6] E. Stott, Z. Guan, J. Levine, J. Wong, and P. Cheung, "Variation and reliability in FPGAs," *Design Test, IEEE*, vol. 30, no. 6, pp. 50–59, Dec 2013.

[7] R. P. Duarte and C.-S. Bouganis, "High-level linear projection circuit design optimization framework for FPGAs under overclocking," in *FPL*, 2012, pp. 723–726.

[8] K. Shi, D. Boland, and G. A. Constantinides, "Accuracy-performance tradeoffs on an FPGA through overclocking," in *IEEE International Symposium on Field-Programmable Custom Computing Machines*, 2013, pp. 29–36.

[9] C.-S. Bouganis, I. Pournara, and P. Cheung, "Exploration of heterogeneous FPGAs for mapping linear projection designs," *IEEE Trans. VLSI Syst.*, vol. 18, no. 3, pp. 436–449, 2010.

[10] C. S. Bouganis, I. Pournara, and P. Y. K. Cheung, "Efficient mapping of dimensionality reduction designs onto heterogeneous FPGAs," in *Proc. 15th Annual IEEE Symp. Field-Programmable Custom Computing Machines FCCM 2007*, 2007, pp. 141–150.

[11] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-6, no. 6, pp. 721 –741, nov. 1984.