High Level Performance Model Based Design Space Exploration for Energy-Efficient Designs on FPGAs

Sanmukh R. Kuppannagari
Ming Hsieh Department of
Electrical Engineering
University of Southern California
Los Angeles, USA 90007
Email: kuppanna@usc.edu

Yusong Hu
School of Electrical
and Electronic Engineering
Nanyang Technological University
Singapore, 639798
Email: ysh_055@usc.edu

Viktor K. Prasanna
Ming Hsieh Department of
Electrical Engineering
University of Southern California
Los Angeles, USA 90007
Email: prasanna@usc.edu

***Abstract*—Energy efficiency has become a key performance metric in implementing application on FPGA. Several parameters such as parallelism, data layout, data re-usability etc. determine energy efficiency. Therefore a parameterized architecture is required to analyze the trade-offs and select the most energy-efficient design. However, increasing the number of parameters exponentially increases the number of possible designs in the design space. It then becomes infeasible to simulate all the designs as time consumed in each simulation is quite significant. In this paper, we perform a high-level performance model based design space exploration for 2D image convolution. This technique allows us to explore the design space quickly and efficiently. We first develop a high level performance model which would allow us to do a parameter sweep of the large design space quickly. The parameter sweep will provide us with a small number of promising designs which can be simulated to arrive at the most energy-efficient design. We prove that a small design subspace will contain the dominating designs with respect to the metric of Energy Efficiency. We simulate the designs from this subspace and show that there is a considerable overlap of 77% between the top designs identified by the performance model and the designs simulated from the dominating design subspace.**

I. Introduction

Due to their high frequency of operation and easy reconfigurability, FPGAs are an attractive option for implementing computation intensive applications such as signal, image and network processing kernels [1], [2]. Because FPGAs can be customized for any specific application, they are high-performance low-power alternatives to general purpose processors. Additionally, due to their ease of programmability, the design and synthesis cycles for FPGAs are far shorter and more economic than that of ASICs.

With the influx of hand-held mobile devices, energy efficiency has become a significant metric in addition to the traditional metrics such as latency and throughput. Power consuming applications exert a strain on the already limited battery and require high packaging cost for cooling infrastructure. To make FPGAs a viable platform for embedded systems, many technological advances are being proposed to improve their energy efficiency such as clock gating [3]. Those must be supplemented at the algorithmic levels to obtain energy efficient designs.

To develop an energy-efficient design, the algorithm designer has to carefully choose among a vast number of available algorithms. Additionally, the various architectural features and limitations of FPGAs need to be considered during mapping process. To analyze trade-offs between various algorithm and architectural choice, a parameterized architecture needs to be developed with parameters denoting the various choices. It is apparent that the number of designs which need to be evaluated will grow exponentially with the number of parameters in the parameterized architecture. Simulating each design is not feasible as each simulation takes considerable amount of time.

The work presented here is part of the TAPAS (Tunable Algorithms for PERFECT ArchitectureS) project [4]. TAPAS is studying a model-driven approach for developing energy-efficient algorithms for FPGAs. The project proposes a design methodology for Hierarchical Design Space Exploration where the designer first chooses an algorithm-architecture pair (as defined in Section IV-A), develops a computational model to quickly evaluate the exponential order design space and arrive at a small number of promising designs, which can then be simulated to get the best designs.

In this paper, we perform a high level performance model based design space exploration of 2D image convolution. We use energy-efficiency as our metric. We develop a parameterized architecture for image convolution, identify various characteristics of the algorithm and the target platform and develop the performance model. Using our knowledge of the algorithm-architecture pair, we identify a small subspace which contains the dominating designs i.e. designs with highest energy-efficiency and prove it mathematically. We simulate the designs of the subspace to calculate their energy-efficiency. Now, using the performance model that we developed, we perform a parameter sweep over the entire design space and arrive at a small number of promising design points. We show that there is a considerable overlap between these promising design points and the subspace containing the dominating designs that we simulated.

Section II gives a background on topics related to this work. In Section III we explore the work done in the community both in the area of image convolution on FPGAs and

This work has been funded by DARPA under the grant number HR0011-12-2-0023.

This material is based upon work supported by the National Science Foundation under Grant No. 1018801

regarding developing performance modeling techniques for FPGA architectures. In Section IV we define the parameterized architecture for image convolution on which we perform a design space exploration in Section V by developing a high-level performance model. We conclude the paper in Section VI giving a brief idea of what we envisage.

II. Background

A. FPGA

Field-Programmable Gate Arrays (FPGAs) are semiconductor devices which can be reprogrammed to achieve desired functionality. They are composed of LUTs which can be configured as logic to implement the functional units. Additionally, they can be configured as memory in form of Distributed RAM (Dist. RAM) to use as small on-chip memory. Dedicated Block RAMS (BRAM) are also provided to cater for on-chip large size memory requirements. FPGAs can be interfaced with external memory when memory requirement is very high.

B. Image Convolution

The problem of image convolution using a 2D Gaussian kernel can be stated as follows: Given a 2D image f of dimensions $M \times N$ and a 2D gaussian kernel $g = h.h^T$ of dimensions $k \times k$, the output image f' of dimensions $M \times N$ is obtained by the Equation [5]:

$$f'(m,n) = \sum_{j=-(k-1)/2}^{j=(k-1)/2} h(j) \sum_{i=-(k-1)/2}^{i=(k-1)/2} f(m+i, n+j)h(i) \quad 0 \leq m \leq M, 0 \leq n \leq N \tag{1}$$

C. Energy-Efficiency

Increasing the number of nodes to improve performance is not scalable as the power consumption increases rapidly. The solution is to rethink the designs focusing on reducing the power consumption. The Green500 List [6] uses *Flops per watt* as their metric for power efficiency/energy efficiency. It can be defined as the number of operations performed per Joule of energy consumed. We use GFLOPS/W as the metric to evaluate energy-efficiency in this work.

III. Related Work

The need for quickly estimating power dissipation at earlier stages to reduce the design times has been studied in the community for a while. In [7] the authors develop an approach to estimate the power of a design on Virtex architectures if the number of logic cells used is known. Authors in [8] focus on high level power estimation by estimating the switching activity given the control data flow graphs of the application. [9] uses device-level simulations to develop coarse-grained architectural models which can be used for power estimations. Several works have focused on estimating the interconnect energy in the designs such as [10], [11].

Several other works focus on RTL based power estimation. [12] presents equation based high level models for RTL level operations on FPGAs. Authors in [13] present a context-based activity propagation to analyse RTL structures and improve the speed of RTL based power estimation. In [14] a dynamic power optimization CAD tool for FPGAs applied at runtime is presented. Both the Gate/Circuit level and RTL level estimation require significant amount of time. They require the design to be synthesized which is quite late in the design cycle and are unsuitable to handle the very large design space that we are dealing with in our parameterized architectures.

There have been some works focusing on high-level modeling using data mining techniques [15], [16]. These however require a lot of training data and might not lead to accurate results for certain application specific optimizations which a designer may incorporate. Xilinx provides a tool named *Xilinx Power Analyzer* [17] which allows us to input the number of FPGA components such as BRAMs, LUTs etc. to estimate the power consumption. However being an excel sheet it requires certain automation to make it useful for design space exploration. In [18] the authors propose a SystemC based algorithmic level estimation. However, for each design the algorithmic model provides the area and activity as input to the system level model and this whole process, although quick, is still not scalable to the large design spaces we are trying to handle. Similarly in [19] a SystemC design is simulated on functional block level instead of gate level or RTL level. This decreases the exploration time but is still not scalable. The work presented in this paper focuses on performance modeling at the design level. We believe that first fixing the domain of the design will greatly help in developing an accurate high-level performance model. A similar approach has been followed in [20] where the authors focus on modeling power for application specific datapath designs. On the similar lines, we focus on application specific modeling to facilitate fast design space exploration of very large design spaces.

Due to its significance in the image processing applications and its computationally intensive nature, 2D image convolution has been extensively studied in the community. On FPGAs, several schemes such as full buffering, partial buffering etc have been developed [21], [22]. Area efficient architecture for shift-variant convolution is proposed in [23]. A systolic architecture tailored for real time windows based operations is proposed in [24]. In this work, to illustrate our design methodology, we use an architecture which uses full buffering technique with pixel reuse to reduce image memory accesses.

IV. Parameterized Architecture and Design Space for Image Convolution

We use the TAPAS methodology of design space exploration to generate energy-efficient designs for the problem of performing image convolution using a 2D Gaussian Kernel. We use two different image sizes of 480P (640x480) and 2160P (3840x2160). Pixel width is 32 bits.

A. Algorithm-Architecture pair

Algorithm-Architecture Pair is defined as an algorithm along with a suitable data layout. The algorithm-architecture pair selected for the illustration uses the symmetry provides by the gaussian kernel and decomposes the 2D kernel convolution process into two 1D kernel convolution processes row-wise and column-wise. The total number of operations required are

$2kMN$ where k is the size of the 1D gaussian kernel and $M \times N$ represent the image size. The input image is laid out in row major order. The size of the gaussian kernel is fixed at $k = 9$ for this illustration.

B. Parameterized Architecture

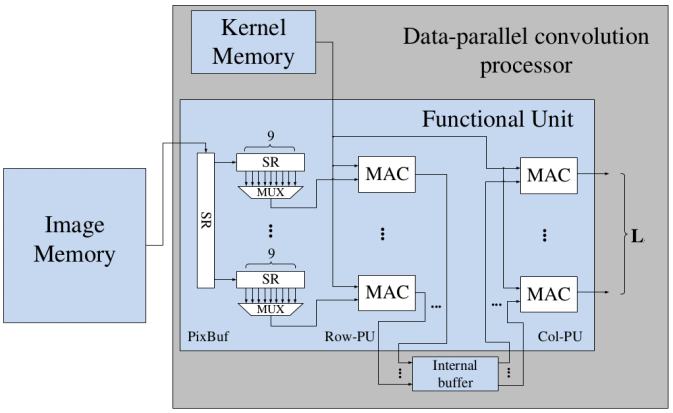

Fig. 1. Parameterized Architecture for Image Convolution

Figure 1 shows the parameterized architecture for convolution for an image of size $M \times N$, where M is the width of the image and N is the height. It is composed of a *Data Parallel Convolution Processor* (DPCP) which performs the convolution on the image stored in the *Image Memory*. The convolution processor can again be sub-divided into functional unit and local memories. The Kernel Memory stores the gaussian kernel. The functional unit consists of a column-wise convolution processing unit (ColPU) and a row-wise convolution processing unit (RowPU) which are a MAC (Multiplier and Accumulator) and perform one computation operation in a clock cycle. L RowPUs work in parallel and produce 1 partial result in k clock cycles which is stored in the internal buffer. L ColumnPUs each read a partial result from the internal buffer and produce the final output pixel in k clock cycles. The internal buffer is composed of $k-1$ shift registers. $k-1$ values received from a RowPU are stored into the shift registers while the kth value is directly fed to the ColumnPU. PixBuf is used to reduce the memory accesses by reusing the input pixels. It is composed of a shift register of size k and a $k - to - 1$ multiplexer. One pixel is shifted into the shift register every k clock cycles, while the multiplexer selects one of the shift register's element every clock cycle for input of RowPU.

The 480P image can be fully stored on-chip. The 2160P image requires to be stored on an external memory as it is larger that the available on-chip memory. The on-chip memory can be implemented using either Dist. RAM or BRAM. The PixBuf can be implemented using flip-flops, BRAM or Dist. RAM. We use DDR3 SDRAM for external memory.

The maximum value of L for a DPCP can be k as mentioned in Section IV-C. We can use several such DPCPs in parallel to process several portions of the image in parallel. Theoretically, we can have N/k such DPCPs running in parallel each one of them processing an image size of $k \times M$. So the maximum possible value of L is $N/k * k$.

We use *Memory Activation Scheduling* [25] to minimize the internal buffer energy consumption. The algorithm-architecture pair performs $2(k-1)MN$ read and write operations in kMN/L clock cycles to the internal buffer with $(k-1)$ shift registers. For $k = 9$, the values of 0.25 and 0.5 are used as the fraction of shift registers active when applying memory activation scheduling for $L = 1$ and $L = 2, 3$ respectively. For greater values, it is not possible to apply memory activation scheduling.

Each shift register of the internal buffer is M words long. It requires $\lceil M/576 \rceil * 8$ BRAMs of size 18Kbits. Theoretically it is possible to have a poor data layout which might result in each word being stored into a single BRAM thereby requiring $\max\{M,\ num_{brams}\}$ where num_{brams} is the number of BRAMs available in the target platform.

So we have the following parameters for our design:

1) **Level of Parallelism**: $1 \leq L \leq N$.
2) **Fraction of Internal Buffer Active**: b. The values are represented by the set $\{1.0, 0.5, 0.25\}$.
3) **PixBuf Architecture Binding**: The technology used to implement PixBuf. Its values are represented by the set $\{$Flip-Flops, Dist. RAM, BRAM$\}$.
4) **Internal Buffer Architecture Binding**: The type of memory used to implement the Internal Buffer. Its values are represented by the set $\{$Dist. RAM, BRAM$\}$.
5) **Kernel Memory Architecture Binding**: The type of memory used to implement the Kernel Memory. Its values are represented by the set $\{$Dist. RAM, BRAM$\}$.
6) **Number of BRAMs for internal buffer**: The number of BRAMS utilized by internal buffer. The value can range from $\lceil M/576 \rceil * 8$ to $\max\{M, num_{brams}\}$.

So the theoretical design space size S is given by the equation

$$S = 24 * N * (max\{M, num_{brams}\} - \lceil M/576 \rceil * 8) \quad (2)$$

The size of the design space for the image sizes of 480P and 2160P are 9,584,640 and 152,340,480 respectively based on the parameters mentioned above. However, these are not the only parameters and there can be more parameters which could increase the design space by many folds.

To enumerate, the architectural components of the algorithm-architecture pair consists of Kernel Memory, Internal Buffer, PixBuf, Row-ColumnPUs and Image Memory.

C. Design Sub-Space Containing Best Designs

We conjecture that the best possible design points will be contained in the design sub-space of $L \leq k$ based on the fact that the maximum level of parallelism of a DPCP is k and the internal memory and kernel memory is increased by 1 for every k increments of L. The proof for the maximum level of parallelism is detailed using the following theorems.

Theorem 1: In a system where elements which need to be stored in a buffer for atleast L cycles arrive at the rate ≥ 1 per clock cycle. The buffer size required is $\geq L$.

Proof: Since we need to consider the minimum bound on the buffer size, we consider the case where the elements arrive at the rate of 1 per cycle and lifetime is L. The maximum occupancy of the buffer will occur when the oldest element is in its last cycle. The number of elements in the buffer at this time is L as can be seen in the Figure 2. So, L is the minimum

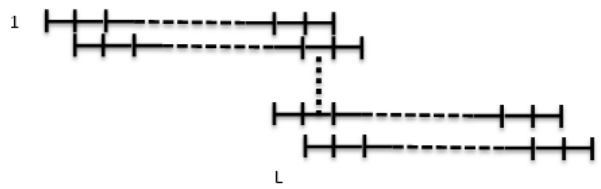

Fig. 2. The maximum occupancy of buffer is L

buffer required to handle this case. And since this is the best possible scenario, its value is $\leq$ the buffer size required.

Corollary: Given a buffer size k, and the lifetime of an element k, the arrival rate cannot be more than 1 element per cycle.

Theorem 2: The maximum value of level of parallelism L in a single DPCP is k.

Proof: Consider a pair of RowPU and ColumnPU. The partial-result produced by a RowPU is consumed by ColumnPU only when it reaches the rightmost position of a shift register. So each partial result is consumed by the ColumnPU k times at the intervals of kM/L cycles with the first consumption directly from the RowPU and the rest from the rightmost position of the $(k-1)$ shift registers. So considering only the clock cycles in which the the partial result is being consumed, the lifetime of the partial result is k such clock cycles. The buffer size is k (including 1 virtual buffer due to direct feeding from RowPU to ColumnPU). So by Theorem 1, the arrival rate of the elements cannot be more than k.

So we can have at most k RowPUs each producing 1 partial-result per cycle. This achieves the maximum arrival rate of k elements per cycle. Any increase in the number of RowPUs will increase the arrival rate which will contradict the theorem. Hence, the maximum level of parallelism of a DPCP is k.

V. Design Space Exploration Using High-Level Performance Model

In Section IV we chose an algorithm-architecture pair for image convolution and identified its parameters. In this section, we develop a high level performance model for the algorithm-architecture pair. We developed a tool that implements design space exploration using the high-level performance model developed and allows us to perform a parameter sweep of the vast design space. We show that there is a considerable overlap between the top results predicted by the performance model and the designs in the subspace containing the dominating designs.

A. Experimental Setup

We use Xilinx Vivado 2013.4 development tools [26] including the Vivado Power Analysis tool for simulating the designs and determining the power dissipation. We use the state-of-the-art Xilinx Virtex 7 XC7VX980 with -2L as our target platform [27]. We fix the processing rate to 0.4 Giga-Operations per second. So with the increase in the level of parallelism, the frequency is decreased accordingly. We use the Micron DDR3 chips with 16 bits output operating at 400MHz. We use Micron DDR3 power calculator for a 1GB target chip [28] to calculate the power dissipation of the image memory. The DDR3 outputs data for 6% of the total time and this gives an average power consumption of 44mW as per the power calculator.

B. Component Power Functions

In Section IV-B, we identified the architectural components of the chosen algorithm-architecture pair. In Table I we enumerate the various architectural binding parameters of these components. For memory components the table lists the size of the components and for logic components it lists the number. Each one of the component is active for $9MN/L$ clock cycles except Internal Buffer which is active for $9MNb/L$ clock cycles.

TABLE I. Architectural Components of Image Convolution on FPGA

Component Name	Architecture-Binding Parameters	Size(words)/ number	Accesses
Kernel Memory	Dist. RAM, BRAM	9	$9MN/L$
Image Memory(480P)	Dist. RAM, BRAM	MN	$18MN$
Image Memory(2160P)	DDR3	MN	$18MN$
Internal Buffer	Dist RAM, BRAM	$8MN$	$18MN$
PixBuf	Flip-Flops	$9L$	NA
Row-Col-PU	MAC	$2L$	NA

TABLE II. Power Functions for Xilinx V7 XC7VX980-2L

Parameter Name	Power Function	Remarks
Dist. RAM	$1.7e^{-5} * s + 0.02$	nJ per access of s bytes
ShiftRegister	.0045	nJ for a word
BRAM Read	0.06	nJ per access
BRAM Activation	0.004	nJ per cycle
MAC	.035	nJ per cycle
DDR3	44mW	average power

Table II enumerates the power functions of the various architectural-binding parameters for the target platform for a single component unless otherwise stated. Note that the power functions we use are in terms of energy consumed per cycle or per access instead of energy consumed per second. We use Micron DDR3 power calculator for a 1GB target chip [28]. We use the power functions for 18Kbit BRAMs. The shiftregister mentioned in the table is used to implement PixBuf and is composed of flip-flops.

C. Performance Model

Based on the power functions mentioned in the previous section, we develop an equation for energy-efficiency in terms of the algorithm-mapping parameters and the architectural-binding parameters mentioned above. The number of operations performed for this algorithm-architecture pair is given by

$$Num\_Ops = 2kMN$$

The performance model generated is not expected to be very accurate as we have not considered the routing power. We refine our model to incorporate the energy consumed by routing as mentioned in the next section.

D. Model Refinement by Routing Power Estimation

Estimating routing power is non-trivial because it varies a lot with the number of components, the place-and-route methodology and fanouts of each component. Inspite of that, we cannot ignore it as it consumes a significant amount of energy.

In order to achieve a higher accuracy in estimating routing power, we propose that such an estimation should be done by fixing an algorithm-architecture pair, analysing the implemented design for a few design points and heuristically arriving at a model.

TABLE III. COMPARISON OF ENERGY-EFFICIENCY OBTAINED BY SIMULATION AND BY PREDICTION FOR 2160P IMAGE

L	b	Energy-Efficiency (GFLOPS/W) Simulation	Energy-Efficiency (GFLOPS/W) Prediction
1	0.25	4.96	6.09
1	0.5	4.60	5.3
1	1	4.0	-
2	0.5	4.90	5.0
2	1	4.49	-
3	0.5	4.70	5.0
3	1	4.70	-
4	1	4.59	4.61
5	1	4.65	4.63
6	1	4.60	4.65
7	1	4.76	4.67
8	1	4.60	4.68
9	1	4.70	4.69
16	1	-	4.57
17	1	-	4.58
18	1	-	4.59

For this work, we use regression analysis to arrive at a model. We use the difference in the energy consumption reported by our performance model and the energy consumption reported by simulations for design points of $L = 1, 3, 5, 7, 9$ to do the regression analysis and arrive at the following equations for 2160P and 480P image sizes respectively.

$$0.89 * L - 1.46 \tag{3}$$

$$2.06 * L - 0.36 \tag{4}$$

Incorporating these equations into our performance model, we run our tool over the vast design space of the image sizes of 480P and 2160P. The analysis of the results is detailed in the next section.

E. Results

We simulate the 13 designs which are included in the sub-space containing the best designs.We compare the results produced by our performance model with these results. We measure the success of our performance model based on the percentage of intersection of the top 13 designs produced by our performance model and the simulated designs. For image size of 2160P, there is 77% overlap with the simulated designs as shown in Table III. The - in an entry represents that either the design point was not simulated as it did not come under the chosen design sub-space containing the best designs or it was not reported among the top 13 designs by the performance model. For the case of 480P the overlap is again 77% as shown in Table IV. The architecture binding parameters corresponding to the table entries are: Dist. RAM for kernel memory, flip-flops for PixBuf and BRAMs for the rest of the memories.

TABLE IV. COMPARISON OF ENERGY-EFFICIENCY OBTAINED BY SIMULATION AND BY PREDICTION FOR 480P IMAGE

L	b	Energy-Efficiency (GFLOPS/W) Simulation	Energy-Efficiency (GFLOPS/W) Prediction
1	0.25	6.27	6.13
1	0.5	6.01	5.99
1	1	5.55	-
2	0.5	6.15	5.96
2	1	5.79	-
3	0.5	6.29	5.95
3	1	5.97	-
4	1	5.88	5.88
5	1	6.06	5.90
6	1	5.63	5.90
7	1	5.97	5.9
8	1	5.97	5.9
9	1	5.88	5.91
16	1	-	4.57
17	1	-	4.58
18	1	-	4.59

The figures show the overlap in a graphical fashion. The top design is correctly predicted by the performance model within the top 3 promising designs. So, the designer can choose the top few points reported by the performance model and simulate them to get the best results as per the requirements.

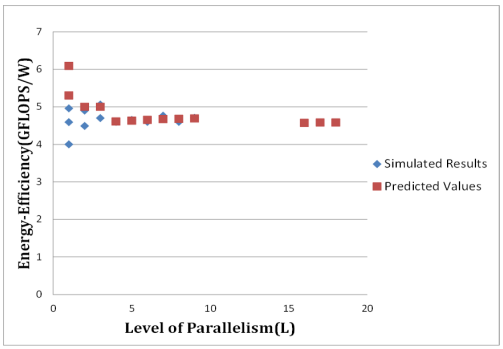

Fig. 3. Comparison of designs predicted vs designs simulated for 2160P

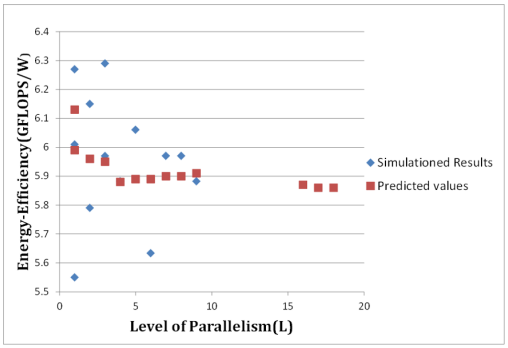

Fig. 4. Comparison of designs predicted vs designs simulated for 480P

The simulations were performed on a Dell Optiplex workstation with 8GB RAM and Intel i-5 processors. The time taken for simulating a single design point was at an average 1.5 hours. If we try to simulate all the design points on 1000 such workstations in parallel, it will take around 27 years to get the results for both the image sizes. However, the tool

which implements our design methodology takes around 4.5 hours to generate the top 13 designs for both the images sizes. Simulating the designs to get the best design will require 39 hours. So the design cycle for analyzing a single algorithm-architecture pair is less than 2 days of computational work. So this allows the designer to compare various algorithm-architecture pairs on various target platforms and choose the best design in a reasonable period of time.

By analyzing just a single algorithm-architecture we cannot claim that we obtained the best possible design for the given problem: 2D Image Convolution for this work. However, several algorithm-architecture pairs can be quickly analyzed using the *Hierarchical Design Space Exploration* based design methodology proposed by TAPAS. This would accelerate the process of generating the best designs for a given target platform.

VI. Conclusion and Future Work

Although we used a very naive method of modeling routing power, the performance model was able to predict the top designs correctly. This was possible because we started by fixing the algorithm-architecture pair and then developed a performance model accordingly. Developing an accurate high-level performance model is critical to evaluate large design spaces quickly and routing power estimation is a major source of inaccuracies. In the future works, we will focus on developing heuristics which would allow us to predict the routing power consumption based on the chosen algorithm-architecture pair.

References

[1] W. J. MacLean, "An evaluation of the suitability of fpgas for embedded vision systems," in *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*. IEEE, 2005, pp. 131–131.

[2] Y. Qu, Y. Yang, and V. K. Prasanna, "Large-scale multi-flow regular expression matching on fpga," in *High Performance Switching and Routing (HPSR), 2012 IEEE 13th International Conference on*. IEEE, 2012, pp. 70–75.

[3] Y. Zhang, J. Roivainen, and A. Mammela, "Clock-gating in fpgas: A novel and comparative evaluation," in *Digital System Design: Architectures, Methods and Tools, 2006. DSD 2006. 9th EUROMICRO Conference on*. IEEE, 2006, pp. 584–590.

[4] TAPAS: Tunable Algorithms for PERFECT ArchitectureS. [Online]. Available: http://ganges.usc.edu/wiki/TAPAS

[5] J. Y. Mori, C. H. Llanos, and P. A. Berger, "Kernel analysis for architecture design trade off in convolution-based image filtering," in *Integrated Circuits and Systems Design (SBCCI), 2012 25th Symposium on*. IEEE, 2012, pp. 1–6.

[6] W.-c. Feng and K. W. Cameron, "The green500 list: Encouraging sustainable supercomputing," *Computer*, vol. 40, no. 12, pp. 50–55, 2007.

[7] K. Weiß, C. Oetker, I. Katchan, T. Steckstor, and W. Rosenstiel, "Power estimation approach for sram-based fpgas," in *Proceedings of the 2000 ACM/SIGDA eighth international symposium on Field programmable gate arrays*. ACM, 2000, pp. 195–202.

[8] D. Chen, J. Cong, Y. Fan, and Z. Zhang, "High-level power estimation and low-power design space exploration for fpgas," in *Proceedings of the 2007 Asia and South Pacific Design Automation Conference*, ser. ASP-DAC '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 529–534. [Online]. Available: http://dx.doi.org/10.1109/ASPDAC.2007.358040

[9] V. Degalahal and T. Tuan, "Methodology for high level estimation of fpga power consumption," in *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*. ACM, 2005, pp. 657–660.

[10] D. Chen, J. Cong, Y. Fan, and L. Wan, "Lopass: A low-power architectural synthesis system for fpgas with interconnect estimation and optimization," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 18, no. 4, pp. 564–577, 2010.

[11] R. Jevtic and C. Carreras, "A complete dynamic power estimation model for data-paths in fpga dsp designs," *Integration, the VLSI Journal*, vol. 45, no. 2, pp. 172–185, 2012.

[12] T. Jiang, X. Tang, and P. Banerjee, "Macro-models for high-level area and power estimation on fpgas," *International Journal of Simulation and Process Modelling*, vol. 2, no. 1, pp. 12–19, 2006.

[13] P. Schumacher, P. Jha, S. Kuntur, T. Burke, and A. Frost, "Fast rtl power estimation for fpga designs," in *Field Programmable Logic and Applications (FPL), 2011 International Conference on*. IEEE, 2011, pp. 343–348.

[14] D. Howland and R. Tessier, "Rtl dynamic power optimization for fpgas," in *Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on*. IEEE, 2008, pp. 714–717.

[15] O. Ulusel, K. Nepal, R. Bahar, and S. Reda, "Fast design exploration for performance, power and accuracy tradeoffs in fpga-based accelerators," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 7, no. 1, p. 4, 2014.

[16] L. Shang and N. K. Jha, "High-level power modeling of cplds and fpgas," in *Computer Design, 2001. ICCD 2001. Proceedings. 2001 International Conference on*. IEEE, 2001, pp. 46–51.

[17] Xilinx Power Estimator. [Online]. Available: http://www.xilinx.com/products/design_tools/logic_design/xpe.htm

[18] N. Abdelli, A.-M. Fouilliart, N. Julien, and E. Senn, "High-level power estimation of fpga," in *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*. IEEE, 2007, pp. 925–930.

[19] S. K. Rethinagiri, R. Ben Atitallah, S. Niar, E. Senn, and J. Dekeyser, "Hybrid system level power consumption estimation for fpga-based mpsoc," in *Computer Design (ICCD), 2011 IEEE 29th International Conference on*. IEEE, 2011, pp. 239–246.

[20] S. Mohanty, S. Choi, J.-w. Jang, and V. K. Prasanna, "A model-based methodology for application specific energy efficient data path design using fpgas," in *Application-Specific Systems, Architectures and Processors, 2002. Proceedings. The IEEE International Conference on*. IEEE, 2002, pp. 76–87.

[21] B. Bosi, G. Bois, and Y. Savaria, "Reconfigurable pipelined 2-d convolvers for fast digital signal processing," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 7, no. 3, pp. 299–308, 1999.

[22] H. Zhang, M. Xia, and G. Hu, "A multiwindow partial buffering scheme for fpga-based 2-d convolvers," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 54, no. 2, pp. 200–204, 2007.

[23] F. Cardells-Tormo and P.-L. Molinet, "Area-efficient 2-d shift-variant convolvers for fpga-based digital image processing," in *Signal Processing Systems Design and Implementation, 2005. IEEE Workshop on*. IEEE, 2005, pp. 209–213.

[24] C. Torres-Huitzil and M. Arias-Estrada, "Fpga-based configurable systolic architecture for window-based image processing," *EURASIP Journal on Applied Signal Processing*, vol. 2005, pp. 1024–1034, 2005.

[25] M. Singh and V. K. Prasanna, "Algorithmic techniques for memory energy reduction," in *Experimental and Efficient Algorithms*. Springer, 2003, pp. 237–252.

[26] Vivado design suite user guide: Design flows overview. [Online]. Available: http://www.xilinx.com/support/documentation/sw_manuals/xilinx2013_4/ug892-vivado-design-flows-overview.pdf

[27] Xilinx. (2013). Virtex-7 FPGA family. [Online]. Available: http://www.xilinx.com/products/silicon-devices/fpga/virtex-7.html

[28] Micron ddr3 sdram system-power calculator. [Online]. Available: http://www.micron.com/products/support/power-calc