# Power-Aware FPGA Mapping of Convolutional Neural Networks

Alexander Montgomerie-Corcoran*, Stylianos I. Venieris†, Christos-Savvas Bouganis*

*Dept. of Electrical and Electronic Engineering, Imperial College London, UK
†Samsung AI Center, Cambridge, UK
{alexander.montgomerie-corcoran15,christos-savvas.bouganis}@imperial.ac.uk, s.venieris@samsung.com

***Abstract*—With an unprecedented accuracy in numerous AI tasks, convolutional neural networks (CNNs) are rapidly deployed on power-limited mobile and embedded applications. Existing mapping approaches focus on achieving high performance without explicit consideration of power consumption, leading to suboptimal solutions when power is considered in a subsequent stage. In this context, there is an emerging need for power-aware methodologies for the design of custom CNN engines. In this work, a methodology is presented for modelling the power consumption of FPGA-based CNN accelerators using a high-level description of modules, together with a power-centric search strategy for exploring power-performance trade-offs within the CNN-to-FPGA design space. By integrating into an existing CNN-to-FPGA toolflow, the proposed power estimation method can yield a prediction accuracy of 93.4% for total system power consumption. Furthermore, it is demonstrated that the associated power-oriented exploration approach can generate CNN accelerators with a 20.1% power reduction over a purely throughput-driven design for AlexNet, maintaining the design's throughput.**

## I. Introduction

Driven by the recent successes of convolutional neural networks (CNNs) and their diversity of applications, major industrial companies have begun to integrate CNN models within their products [1], prominently targeting mobile and embedded applications. To execute CNN workloads without the overhead of cloud offloading, *on-device inference* [1], [2] is commonly employed. In this setting, the CNN computations are performed locally using the computational resources of the device, with all data remaining local. Nevertheless, battery-powered mobile and embedded platforms are typically severely constrained in terms of their power budget [3]. In this respect, power efficiency becomes a primary objective in the deployment of CNNs on resource-limited settings.

A promising platform that balances high performance with power efficiency are FPGAs [2]. The customisation capabilities of FPGAs offer the opportunity to tailor the generated hardware with respect to the given CNN and the available power budget. The highly customisable aspect of FPGAs creates a large design space for mapping CNNs on the device and in turn allows for specific performance metrics to be targeted explicitly. This work approaches the problem of mapping CNNs to FPGAs as a Design Space Exploration (DSE) task, and utilises a power-modelling technique to rapidly traverse the design space and yield power-efficient designs. In particular, the main contribution of this work is a high-level power-modelling technique tailored to CNN hardware modules which enables the identification of power-efficient designs during the DSE phase of CNN-to-FPGA toolchains.

## II. Background & Related Work

A number of works have focused on the automated mapping of CNNs to FPGAs [4]–[7]. By exploring the design space within the constraints imposed by the platform, these tools are able to target specific performance objectives. The existing CNN-to-FPGA toolflows typically employ performance models, either using a roofline [8], [9] or graph representation [4], [5], in order to estimate the attainable performance as a function of the configurable hardware parameters and explore various candidate designs. A toolflow that demonstrates significant flexibility with respect to the architectural space is fpgaConvNet [4], [10]. The expressivity of fpgaConvNet's graph representation enables the description of diverse CNN engines including both single computation engines and streaming architectures [7]. In this respect, without loss of generality, fpgaConvNet has been used as the backbone for this work.

Modelling of power consumption has been covered at different levels of abstraction for FPGA-based designs [11]–[14], where knowledge of the design as well as evaluation time affect the model's accuracy. Based on its level of abstraction, a power model trades off predictive accuracy for generality and speed of estimation. Alongside levels of abstraction, pattern-dependence [15], [16] trades generality with accuracy.

Power and energy efficiency have been investigated in the area of CNN accelerator design. Two prominent accelerators with significant power and energy reductions are MINERVA [17] and EYERISS [18]. MINERVA employs quantisation, pruning and on-chip memory voltage scaling to reduce energy consumption. EYERISS identifies off-chip memory accesses as the primary source of excessive power consumption and as such exploits the data-reuse patterns of CNN workloads with the goal to decrease the memory transfers needed for weights, feature maps and partial sums deterministically.

## III. Power Consumption Modelling

Despite the extensive existing efforts in power consumption modelling, there is still a gap between accuracy and estimation speed when targeting CNN accelerators on FPGAs. To

978-1-7281-2943-3/19/$31.00 ©2019 IEEE
DOI 10.1109/ICFPT47387.2019.00059

this end, a novel power modelling methodology is proposed tailored to FPGA-based CNN systems. Our method overcomes the limitations of existing tools and combines high accuracy with fast estimation by exploiting two key observations: *1)* strong statistical patterns in the feature maps, and *2)* parametrisation of commonly used CNN hardware modules.

### *A. Dynamic Power Modelling*

With dynamic power being a significant source of power consumption in FPGA-based CNN designs, the structure of a typical CNN architecture and the characteristic propagation of data through it is investigated first. Following the streaming paradigm, each module in the CNN hardware mapping is pipelined with data passing through the various modules sequentially.

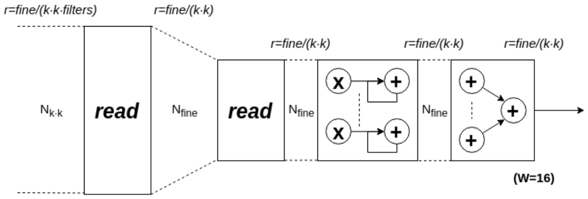

Fig. 1. Example of mapping a convolution module to hardware.

Alongside the high-level topology of the accelerator, each of the modules is parametrised individually to enable a tunable level of parallelism. As an example, Fig. 1 shows a parametrised convolution module. This diagram highlights the operations and the parameters that govern them, showing how high-level parameters can affect the hardware. In particular, streams, rates, data-widths, and operations are given. All modules within the fpgaConvNet framework are designed using High Level Synthesis (HLS) tools and so are abstracted from the hardware implementation. Therefore, in lieu of basic hardware building blocks such as LUTs and DSPs, operations are instead used to form the power model.

To start with, the dynamic power consumption for the CNN design is expressed in terms of its hardware modules as

$$P^{\text{D}}_{\text{total}} = P_{\text{routing}} \; + \sum_{\text{module} \in \mathcal{M}} P^{\text{D}}_{\text{module}} \tag{1}$$

where $\mathcal{M}$ is the set of instantiated hardware modules of the CNN architecture, $P^{\text{D}}_{\text{module}}$ is the dynamic power consumed by the particular module and $P_{\text{routing}}$ is the power consumed by the routing logic between the modules. Usually the amount of resources used to implement each module is considerably larger than the routing between modules. Thus, the module power consumption dominates the routing logic and hence $P_{\text{routing}}$ can be omitted from the power estimation model. To estimate the power of individual modules $P^{\text{D}}_{\text{module}}$, a per-module power model is derived as a function of the type, number and organisation of operators within it, as given by Eq. (2).

$$P^{\text{D}}_{\text{module}} = \frac{f_{\text{clk}}}{2} \cdot V_{DD}^2 \cdot r(\text{input}) \sum_{i=1}^{N_{\text{op}}} P^{\text{D}}_{\text{op},i}(\text{param}_i) \tag{2}$$

where $f_{\text{clk}}$ is the clock frequency, $V_{DD}$ is the supply voltage, $N_{\text{op}}$ describes the number of operators within the module such as adders and multipliers, $r(\text{input})$ is the input rate for the module and $P_{\text{op},i}(\text{param}_i)$ describes the contribution of the i-th operator to the power consumption of the module. The rate $r(\text{input})$ also describes the output rate of each preceding module in the hardware design. In a streaming-based architecture, such as the one described in [10], the rates of each module are propagated through the design.

A relationship can be derived between the operation's parameters and it's overall dynamic power consumption, in a similar fashion to the power factor approximation technique [13], leading to the model given in Eq. (3).

$$P^{\text{D}}_{\text{module}} = \frac{1}{2} f_{\text{clk}} \cdot V_{DD}^2 \cdot r(\text{input}) \sum_{i=1}^{N_{\text{op}}} k_i \cdot W_i \cdot N_i \cdot r_i \cdot s_i \tag{3}$$

where $W_i$ is the wordlength, $N_i$ is the number of input streams, $r_i$ is the processing rate of the i-th operator and $s_i$ is the switching activity entering the i-th operator. $s_i$ is found through statistical analysis of feature maps from a given dataset. The coefficient $k_i$ is introduced in order to capture the unknown relationship between the i-th operator's parameters and its power. With this approach, the power of the module is modelled as the sum of the operator power, and scaled by the rate at the input of the module. The value of $k_i$ is obtained using a regression method based on empirical data of power consumption to populate the per-module power models.

### *B. Static Power Modelling*

To complement dynamic power estimation, the proposed methodology includes a predictive model for static power consumption. The foundation of the static power model is based on the consumption of power through bias currents in resources. This suggests a relationship between the resource usage and static power consumption. Alongside operations, variables also contribute to the static power consumption, as they equate to instantiated hardware such as registers and BRAM. The proposed model draws a linear relationship between static power and resource usage for each of the separate modules, and is described as

$$P^{\text{S}}_{\text{module}} = \sum_{i}^{N_{\text{op}}} a_{T_{\text{op}}} \cdot T_{\text{op},i} + \sum_{j}^{N_{\text{var}}} b_{T_{\text{var}}} \cdot T_{\text{var},j} \tag{4}$$

where $N_{\text{op}}$ is the number of operations in the module, $T_{\text{op},i}$ is the type of operation for the i-th operation, $N_{\text{var}}$ is the number of variables and $T_{\text{var},j}$ is the variable type for the j-th variable. Both $a_{T_{\text{op}}}$ and $b_{T_{\text{var}}}$ are scalar coefficients for each variable type and operation type. As a result, Eq. (4) captures the static power contribution of each operation and variable used.

### *C. Memory Interface Modelling*

The final aspect to the power estimation model is the power consumption from off-chip memory accesses, which contributes to the overall power consumption. This is due to the fact that capacitances of data-lines from IO connections are significantly larger than on-chip routing capacitances. The derived model is given in Eq. (5).

$$P_{DDR} = k_{\text{idle}} + k_{\text{dynamic}} \cdot f_{\text{clk}} \cdot V_{\text{DDR}}^2 \cdot N_{\text{ports}} \cdot W_{\text{data}} \cdot r \cdot s \quad (5)$$

where $f_{\text{clk}}$ is the clock frequency of the design, which is typically less than the clock frequency of DDR, $V_{\text{DDR}}$ the supply voltage of the DDR which drives the data lines, $N_{\text{ports}}$ the number of memory ports used, $W_{\text{data}}$ the data width of the memory port, $r$ the rate of data into the design, and $s$ the average switching activity of the data entering. There are two coefficients to this model: $k_{\text{idle}}$ describing the idle power of the DDR, and $k_{\text{dynamic}}$ capturing the coefficient for the dynamic power consumption from activity on the DDR's data line. Overall, the total power consumption is given by

$$\tilde{P} = \sum_{\text{module} \in \mathcal{M}} P^{\text{D}}_{\text{module}} + \sum_{\text{module} \in \mathcal{M}} P^{\text{S}}_{\text{module}} + P_{DDR} \quad (6)$$

## IV. Power-Constrained Optimisation

With a power-modelling methodology in place, power-driven designs can be explored. Within the embedded space, power is a crucial aspect, and characterising and limiting power consumption can play a key role to the configuration of the final design. To guide the space exploration towards designs that comply with the available power budget, the following optimisation problem is defined:

$$\begin{aligned} &\underset{\gamma}{\text{maximise}} && T(\gamma) \\ &\text{subject to} && rsc(\gamma) \leq rsc_{\text{Avail.}} \ , \ \tilde{P}(\gamma) \leq P_{\text{max}} \end{aligned} \quad (7)$$

where $\gamma$ represents the current design point, $T$, $\tilde{P}$ and $rsc$ return the throughput in GOp/s, the power in watts and the resource utilisation of the current design point, and $P_{\text{max}}$ is the available power budget in watts. Under this formulation, Eq. (7) provides an objective function for obtaining a high-throughput design under power constraints.

## V. Evaluation

The proposed power estimation model is evaluated first for the fpgaConvNet framework [10] with coefficients generated for the Xilinx ZC702 board. The model is evaluated against board-level power measurements for LeNet [19] and AlexNet [20] CNNs, and comparisons of evaluation time are made with existing power estimation tools. Finally, the power model is incorporated into a DSE tool to expose power-efficient designs within the fpgaConvNet framework.

### A. Evaluation of Power Consumption Model

In this section, the accuracy of the developed power consumption model is evaluated. This is investigated by selecting four different designs for LeNet on ZC702. This board allows the separate measurement of power for the programmable logic ($\boldsymbol{P}_{PL}$) and off-chip memory ($\boldsymbol{P}_{DDR}$) through the power management bus. Each hardware design is generated via fpgaConvNet and run on the target FPGA at 125 MHz for a batch size of 254 inputs with a 16-bit fixed-point precision.

| Design | T (*fps*) | Rsc. (%) | $\boldsymbol{P}_{PL}$ (W) |  | $\boldsymbol{P}_{DDR}$ (W) |  |
|---|---|---|---|---|---|---|
|  |  |  | Actual | Model | Actual | Model |
| *1* | 59.06 | 14.84 | 0.1458 | 0.1889 | 0.6422 | 0.642 |
| *2* | 1475.89 | 24.85 | 0.2792 | 0.2187 | 0.6322 | 0.6421 |
| *3* | 4761.01 | 56.41 | 0.6192 | 0.6886 | 0.6436 | 0.6423 |
| *4* | 5234.95 | 57.11 | 0.6521 | 0.5638 | 0.6395 | 0.6423 |

TABLE I
Board-level Measurements of LeNet on ZC702

Table I lists the estimated and measured power, together with the measured throughput and average resource utilisation.

In terms of power estimation, the power predictions lie fairly close to the actual measured power on the target platform with an average error of less than 18% across the four designs. In the case of $\boldsymbol{P}_{PL}$, the accumulation of errors across individual hardware modules contributes to the model's small error, but stays within 50 mW for designs 1 and 2 and below 100 mW for 3 and 4. With respect to off-chip memory power, as all four designs only use 16 bits of a single port, memory bandwidth utilisation is low and hence $\boldsymbol{P}_{DDR}$ is dominated by idle power, leading to the little variation in both measured and estimated $\boldsymbol{P}_{DDR}$ across designs.

### B. Comparison with Existing Power Estimation Tools

To evaluate all aspects of the proposed methodology, this section presents a comparison with widely used vendor tools for power estimation. Table II presents the comparison of our method with the high-level Xilinx Power Estimator (XPE), Vivado power estimation after synthesis and implementation *with* (Vivado (*saif*)) and *without* (Vivado) activity information.

| Tool | Evaluation Time | Error (%) |
|---|---|---|
| *XPE* | <1 second | 624.37 |
| *Vivado* | >30 minutes | 559.86 |
| *Vivado (saif)* | >1 hour | 5.01 |
| ***Proposed solution*** | **<1 second** | **21.50** |

TABLE II
Comparison of power modelling tools

The proposed method overcomes the limitations of both the high error of XPE and the large evaluation time of Vivado (*saif*). By offering rapid power estimation, the design space can be traversed efficiently and many different alternative designs can be explored. Furthermore, the low error allows the proposed power consumption model to be used to guide DSE towards power-efficient designs.

### C. Impact of Estimation Accuracy on the Design Space

To visualise how the error between the model and measurements affects the design space, the predicted and actual power are plotted against latency for the first layer of AlexNet in Fig. 2. The legend indicates the different designs that were run, with the blue measurements indicating the model and the red indicating the actual measurements. Parallel and sequential conv refer to the amount of parallelism used for the dot-product units in the convolution module. As shown in Fig.

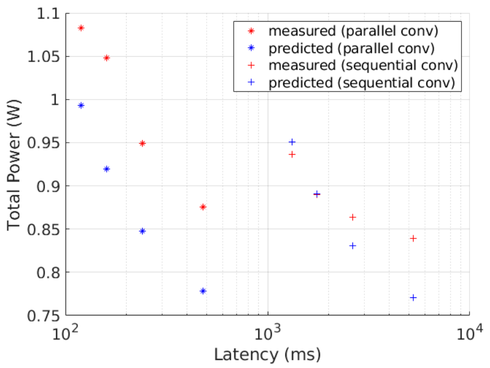

Fig. 2. Measured and predicted power for the $1^{st}$ layer of AlexNet.

2, our model is able to predict power within 100 mW of error, across a range of valid designs within the resource constraints of the device. This figure highlights the existence of power-efficient designs within the design space.

### *D. Design Space Exploration*

Having demonstrated the accuracy of the power modelling framework as well as the existence of power-efficient designs within the design space, the optimiser is now evaluated on its ability to identifying power-efficient designs. Initially, the throughput-power design space is depicted in Fig. 3 by exploring an unconstrained throughput objective. This design exploration is done for the ZC706 platform.

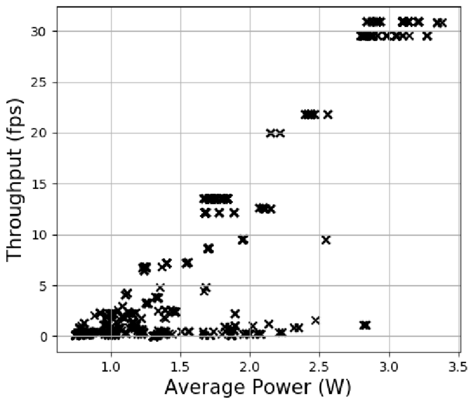

Fig. 3. DSE with a throughput objective for AlexNet on ZC706.

A clear pareto-optimal front can be seen, where average power has a linear relationship with the throughput of the design. An interesting observation is the occurrence of design points on the same throughput plane, yet with larger average power consumption. This indicates the existence of power-efficient designs which achieve high throughput at a reduced power consumption. For example, the highest throughput can be achieved through a range of designs. However, the most power-efficient yields a 20.1% power reduction over the most power-consuming design.

## VI. Conclusion

This paper presents a method of modelling the power consumption of an FPGA-based CNN accelerator system from a high-level description. This power model is then integrated within a DSE-based optimiser to expose power-efficient designs within a CNN-to-FPGA mapping framework. This work brings power consumption to the forefront of the fgpaConvNet framework, and promotes methods which can be used across other frameworks. In this way, low-power implementations of CNNs will be realisable for a host of platforms with harsh power constraints.

## References

[1] C.-J. Wu *et al.*, "Machine Learning at Facebook: Understanding Inference at the Edge," in *HPCA*, 2019.

[2] S. I. Venieris, A. Kouris, and C.-S. Bouganis, "Deploying Deep Neural Networks in the Embedded Space," in *2018 International Workshop on Embedded and Mobile Deep Learning (EMDL), MobiSys*, 2018.

[3] X. Xu, Y. Ding, S. X. Hu, M. Niemier, J. Cong, Y. Hu, and Y. Shi, "Scaling for edge inference of deep neural networks," *Nature Electronics*, vol. 1, no. 4, pp. 216–222, 2018.

[4] S. I. Venieris and C.-S. Bouganis, "fpgaConvNet: A Framework for Mapping Convolutional Neural Networks on FPGAs," in *FCCM*, 2016.

[5] H. Sharma *et al.*, "From High-level Deep Neural Models to FPGAs," in *MICRO*, 2016.

[6] H. Zeng, R. Chen, C. Zhang, and V. Prasanna, "A Framework for Generating High Throughput CNN Implementations on FPGAs," in *FPGA*, 2018.

[7] S. I. Venieris, A. Kouris, and C.-S. Bouganis, "Toolflows for Mapping Convolutional Neural Networks on FPGAs: A Survey and Future Directions," *ACM Comput. Surv.*, 2018.

[8] N. Suda, V. Chandra, G. Dasika, A. Mohanty, Y. Ma, S. Vrudhula, J.-s. Seo, and Y. Cao, "Throughput-Optimized OpenCL-based FPGA Accelerator for Large-Scale Convolutional Neural Networks," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '16. ACM, 2016, pp. 16–25.

[9] X. Wei, C. H. Yu, P. Zhang, Y. Chen, Y. Wang, H. Hu, Y. Liang, and J. Cong, "Automated Systolic Array Architecture Synthesis for High Throughput CNN Inference on FPGAs," in *Proceedings of the 54th Annual Design Automation Conference 2017*, ser. DAC '17. ACM, 2017, pp. 29:1–29:6.

[10] S. I. Venieris and C. Bouganis, "fpgaConvNet: Mapping Regular and Irregular Convolutional Neural Networks on FPGAs," *TNNLS*, 2018.

[11] E. Macii, M. Pedram, and F. Somenzi, "High-level power modeling, estimation, and optimization," *TCAD*, 1998.

[12] L. Benini, A. Bogliolo, M. Favalli, and G. De Micheli, "Regression Models for Behavioral Power Estimation," *Integr. Comput.-Aided Eng.*, 1998.

[13] P. M. Chau and S. R. Powell, "Power dissipation of VLSI array processing systems," *IJVSP*, 1992.

[14] A. Lakshminarayana, S. Ahuja, and S. Shukla, "High level power estimation models for fpgas," in *2011 IEEE Computer Society Annual Symposium on VLSI*, July 2011, pp. 7–12.

[15] C.-T. Hsieh, Q. Wu, C.-S. Ding, and M. Pedram, "Statistical sampling and regression analysis for rt-level power evaluation," in *ICCAD*, 1996.

[16] S. R. Powell and P. M. Chau, "A model for estimating power dissipation in a class of DSP VLSI chips," *TCAS*, 1991.

[17] B. Reagen *et al.*, "Minerva: Enabling Low-Power, Highly-Accurate Deep Neural Network Accelerators," in *ISCA*, 2016.

[18] Y. Chen, J. S. Emer, and V. Sze, "Eyeriss v2: A flexible and high-performance accelerator for emerging deep neural networks," *CoRR*, 2018.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. of the IEEE*, 1998.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS*, 2012.