I. INTRODUCTION

This paper presents DeepPump, a novel approach for generating and optimising hardware designs of deep Convolutional Neural Networks (CNNs) with multi-pumping on FPGA platforms. **Multi-pumping** [1] is a promising technique to save hardware resource usage by replacing $M$ parallel units with one clocked at $M$ times the global clock rate. DeepPump aims at automatically adopting multi-pumping when generating hardware designs for CNNs. It has three components: a parameterised CNN accelerator architecture that supports multi-pumping, a design model for trade-off analysis related to multi-pumping, and an optimisation flow for improving the architecture based on the design model.

II. DeepPump Framework

DeepPump contains a multi-pumped streaming architecture, in which an input stream of feature maps is buffered and computed, and the computation units are multi-pumped. Multi-pumping works effectively in this architecture because there are many parallel computing units that can achieve high maximum clock frequencies. We characterise this architecture with three parameters: $N$ is the original number of parallel blocks, $M$ is the multi-pumping factor, and $F$ stands for the global clock frequency. We also devise a parameterised implementation of this architecture that is described as follows. First, the streaming interface of the multi-pumping blocks, i.e. the width and frequency of streams, remains the same. Second, the number of parallel blocks is reduced by a factor of $M$, and both their clock frequencies and number of cycles are increased by a factor of $M$. Third, additional logic to control the behaviour at each multi-pumped cycle, such as multiplexers and counters, is attached to the original design. In this way, we build a parameterised multi-pumped architecture.

Based on this architecture, we derive a design model for trade-off analysis while applying multi-pumping, by predicting the latency, resource usage and power consumption. Latency ($O(NF)$) only relates to $N$ and $F$; multi-pumping normally does not affect the speed. Resource usage ($O(N/M) + O(NM)$) has components which are directly and inversely proportional to $M$. Power consumption consists of static power ($O(N/M)$) and dynamic power ($O(NMF)$) components.

Moreover, we provide an optimisation flow for generating multi-pumped designs with optimised parameters. It first identifies constant parameters within the design model by learning from real hardware builds. It then solves a constrained optimisation problem with minimising latency as objective, and with reduced resource usage and power consumption as constraints.

To evaluate DeepPump, we compare optimised designs generated by DeepPump with designs from previous work [2] [3]. Our implementation targets the Maxeler MAX4 dataflow engine and processes multiple batches of $512 \times 32 \times 32$ (channel $\times$ height $\times$ width) input feature maps by a convolution layer with 512 output filters. Table I shows that the design generated by DeepPump outperforms other designs in both performance (GOP/s) and resource efficiency (GOP/s/Slice). The number of slices for a Stratix V device is estimated by the number of resource groups with equivalent logic capacity.

III. CONCLUSION

This paper presents DeepPump, an approach that generates CNN hardware designs with multi-pumping, which have competitive performance when compared with previous designs. Future work includes integrating DeepPump with other optimisations, and providing further evaluations on various FPGA platforms.

ACKNOWLEDGMENT

The support of UK EPSRC (EP/L00058X/1, EP/L016796/1 and EP/N031768/1), the European Horizon 2020 Research and Innovation Programme under grant agreement number 671653, Maxeler and Intel is gratefully acknowledged.

REFERENCES

