Design Space Exploration for the physical design of a 12nm AI processor using Relative Placement methodology

Mohit Sharma
Dept. of ECE
University of Windsor
Windsor, ON, Canada
sharma88@uwindsor.ca

Pavel Sinha
Aarish Technologies Inc.
Brossard, QC, Canada
pavelsinha@aarish.ai

Don Dattani
Aarish Technologies Inc.
Brossard, QC, Canada
don.dattani@get2silicon.com

Mohammed Khalid
Dept. of ECE
University of Windsor
Windsor, ON, Canada
mkhalid@uwindsor.ca

***Abstract*—This paper presents the results of design space exploration (DSE) for the memory-macro placement of a low-power Artificially Intelligent (AI) processor provided by our industry partner in a collaborative project. The commercially available physical design CAD tools for automatic placement of macros often give unoptimized macro placements that have a large area and total wire length when compared to manual placement of macros. The goal of this research is to obtain an optimized memory macro placement for the AI processor that provides the lowest possible area, wirelength, and utilization through manual placement in small amount of time comparatively. A relative memory macro placement methodology is developed to ease the process of placing the macros. The idea behind the relative placement is that a rectangular object can be placed at 16 different positions around another rectangular object depending on its length and width. Using this methodology for macro placement resulted in a 25% improvement in area requirements, 40% improvement in utilization and 50% decrease in the wirelength when compared with the results for tool placement. Further, the methodology reduced the hand-placement time for memory macros from 2-3 weeks for part of the design to 3-4 days for the complete design.**

***Index Terms*—Low Power, Hardware acceleration, Design space exploration, Macro placement, Physical Design**

I. Introduction

Modern System on Chips(SoCs) can consist of millions of standard cells and hundreds of macros. This makes fast prototyping of blocks difficult and makes efficient DSE of the SoC a time consuming process. A key task in the prototyping process is floorplan generation, i.e., macro placement. Macro placement can have a significant impact on final design quality of results (QoR), and without a thorough DSE, a floorplan may not achieve desired performance or lead to design convergence issues. Current EDA floor-planning solutions are fast but do not produce good enough macro-placements for large designs. For circuits with more than 200 macros, it usually takes 2 to 4 weeks for the floorplan to reach the desired QoR even after using automated floor-planning solutions, which has to be augmented with manual effort from physical designers. Hence, automated macro-placements achieved by commercially available physical design CAD tools are unoptimized.

This paper presents the relative macro-placement methodology for faster manual physical design process for designs with large number of macros. DSE of a 12nm CNN-Processor architecture provided by our industry partner is carried out by the use of the aforementioned methodology to improve metrics such as Area, Power, Utilization and Timing.

25% area improvements, 40% increase in utilization and 50% decrease in wire length , and 30% decrease in total power are achieved using the relative memory macro placement methodology. This methodology exploits the fact that memory macros are rectangular blocks and can be placed relative to each other. As a result of using the methodology based script, hand placement time reduces from 2 weeks for part of design to 3-4 days for the complete design while considering precise placement with respect to the grid.

Following sections are organized as follows. Section II reviews the related research. DSE of the AI processor is presented in section III. Section IV evaluates and compares the physical placement achieved by the relative memory macro placement methodology with the physical placement achieved by the tool. Finally, section V concludes the DSE.

II. Related Work

Existing automated macro-placement methods either have bad QoR or are only capable of handling small number of macros as compared to manual placement of macros. For e.g., as presented in Table III of [1], hand-placement provides better macro-placement results as compared to automated flows. Similarly, as presented in Table 2 of [2], designs with maximum macro count of about 50 have been considered. These drawbacks are curtailed to get better results, with the help of hand-placement with semi-automated placement script based on relative-placement methodology. This methodology reduces DSE time for manual placement while achieving the QoR for manual placement.

Several CNN processors have been designed for the low-power AI edge processing [3]–[5]. The processor provided by our industry partner [6] is another low-power AI processor that has the following features:

1) Optimal memory access of both coefficients and intermediate data for low power profile.

Manuscript 8 submitted to 2023 IEEE 66th International Midwest Symposium on Circuits and Systems. Received April 6, 2023.

2) A non-load-store architecture where control flow is embedded into data flow.
3) No-off chip memory access.
4) Scalable across multiple chips with zero overhead and linear scaling of computation with multiple chips.
5) The system architecture provides optimal latency (i.e. Minimum Input to Output delay).

III. Design Space Exploration

The physical design space of the CNN sub-processor was explored as part of this research. The DSE consisted of (i) selecting and generating memory macros of given depth and memory bitwidth based on their technology type, power, timing, and area, and (ii) Placing these generated memory macros so that the placement-driven synthesis gave optimal placement of these and the logic cells in terms of area, wirelength and congestion. Memory placement is discussed in more detail below and was guided by the following ideas:

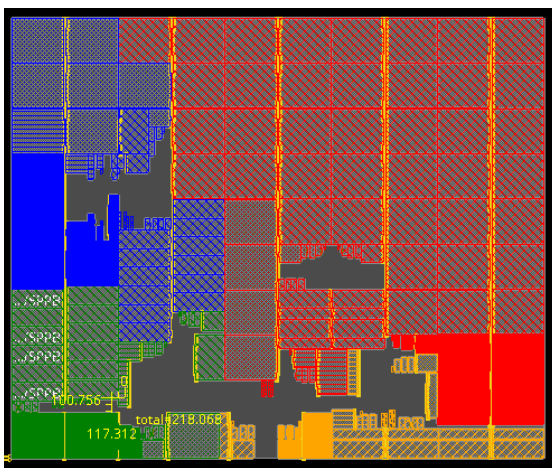


Fig. 1: Placement image demonstrating that same core compute memories were placed closer to each other

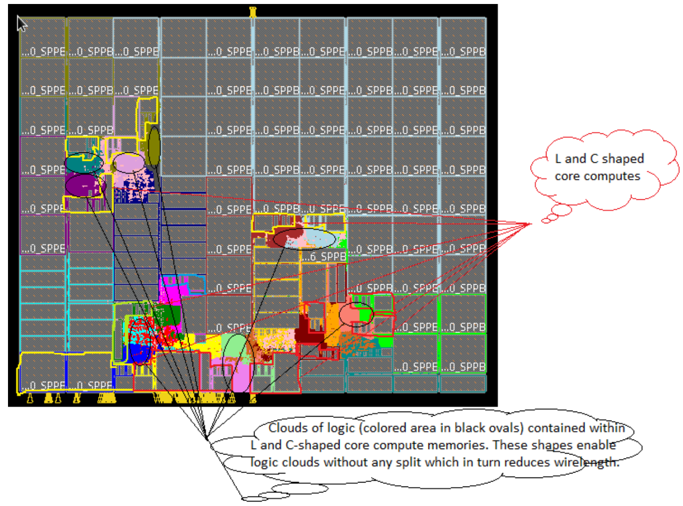


Fig. 2: Placement image showing CC memories placed in the form of closed C-shape or L-shape

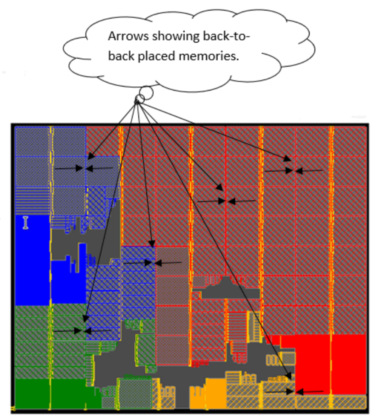


Fig. 3: Placement image showing back to back placed memories

- Memory macros that communicate with each other more often, were placed close to each other to ensure shorter wire-lengths and positive slack or better timing. In the figure 1, the image of PFA with four PFUs are shown with red, green, blue and orange colors. Each PFU has four core computes that can be distinguished by the pattern: dotted, crossed, striped and solid. As can be seen in the image, similar patterned core compute elements were placed closer to each other.
- CC memories were placed in the form of a closed C-shape or L-shape to make sure the corresponding logic cells that are part of the logic cloud were placed by the tool in the area enclosed by the memory macros. This allowed logic clouds to be a uniform entity without splits which again ensured shorter wirelengths and better timing.
- Let the pin face of the macro be called P (Pin face) and non-pin face of the macro be called B (Back Face). Memory macros were placed with B side facing for two macros without any gap as much as possible. This is because when the non-pin sides (B) of two macros are placed back to back, the macros could be aligned so that no space/channel was left in between them. This made sure that maximum area could be utilized and no logic cells could be placed in between the macros. The standard cells would have otherwise been placed into the channel regions left in between the non-pin facing side of the macros.
- However, enough space was left between macros whose pin sides (P) were facing each other so that standard cells could be placed in between for easier connection with pin facing side of the macros. Macros were also placed to ensure that enough space was left for halo and endcap cells. Memory macros facing each other were placed keeping enough distance between them so that two stripes of power supply lines, each consisting of VDD and VSS, could run between the macros. This ensured that power was available to all the endcap cells.

The iterative improvement as part of the design space ex-

Manuscript 8 submitted to 2023 IEEE 66th International Midwest Symposium on Circuits and Systems. Received April 6, 2023.

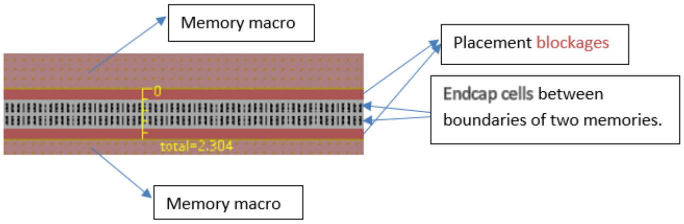


Fig. 4: Placement image showing endcap cells placed between memory macros

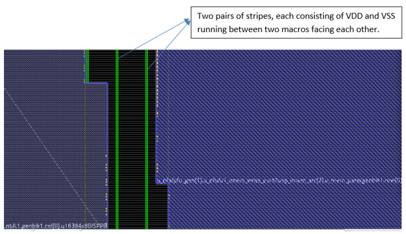


Fig. 5: Placement image showing stripes of VDD and VSS lines between facing memory macros

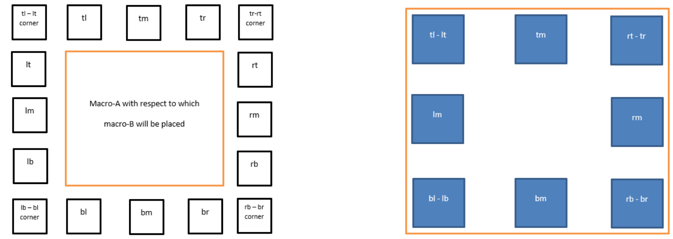
ploration helped in making sure that no hollow spots were left that didn't contain any memory macros or logic cells by compacting the core. This helped in reducing the area of the chip.

Relative Placement

To make sure that memory macro placement principles described above were followed, the relative methodology was used. As per this methodology, rectangular objects like the memory macros are placed relative to one another. This has two benefits: (i) User does not need to identify the exact coordinates for each macro. (ii) This stitching of macros allows faster incremental changes.

Identifying exact coordinates for a macro becomes difficult due to the preciseness of the placement needed which is equal to the width and height of an endcap cell or a halo cell. These cells are 200 – 400X smaller than the size of a macro. If the macro cell count is big and a user needs to calculate exact coordinates or zoom in while placing using a GUI for every macro, then macro placement becomes a time-consuming and difficult task. On the other hand, with relative placement, once a starting macro has been placed with respect to a core boundary, other memory macros can be placed relative to it with a pre-defined margin so that sections of macros can be stitched together. That is, multiple macros can then be placed relative to each other to build stitched sections. The stitched macros can be grouped so that incremental changes, which would otherwise require moving multiple macros, can be achieved simply by moving the pivot macro around which all other macros are placed.

Consider a rectangular shape object as shown in Figure 6a. A second rectangular object (B) can be placed at sixteen

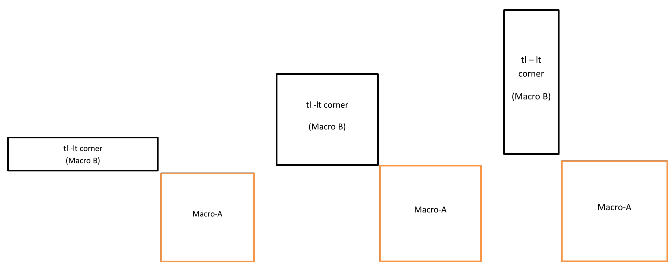


(a) Relative positions of macro B around macro A

(b) Placement of macro within the core

Fig. 6: Relative Placement

different locations around the first object. These locations are left-top (lt), left-middle (lm), left-bottom (lb), left-bottom corner or bottom-left corner (lb or bl corner), bottom-left (bl) , bottom-middle (bm), bottom-right (br), bottom-right corner or right-bottom corner (br or rb corner), right-bottom (rb), right-middle (rm), right-top (rt), right-top corner or top-right corner (rt or tr corner), top-right (tr), top-middle (tm), top-left (tl) and top-left corner or left-top corner (tl or lt corner). Figure 7 shows a possible configuration for the placement of macro B with respect to macro A where macro B can be wider or taller or the same size as macro A.



(a) (b) (c)

Fig. 7: Relative Placement image showing wider/same size/taller macro B, top-left corner or left-top corner, placement w.r.t. macro A

Methodology for the case of memory macro placement: In case of memory macro placement, the objects are memory macros and the methodology is a script. The primary objective of this script was to help reduce time per cycle of design space exploration by reducing the time for hand placement.

The script has two placement modes: (i) within the core and (ii) around another memory macro. Features include: (i) Adding a default space around the boundary of the macros for endcap cells., (ii) Adding space for power line stripes, (iii) Aligning the memory macros to two orthogonal grid lines.

IV. Experimental Results

In this section, we present the results of the placement done by the tool and the placement achieved after DSE. Table I lists the metrics measured, including Worst Negative Slack, Total Negative Slack, Area, Utilization, Total Power and

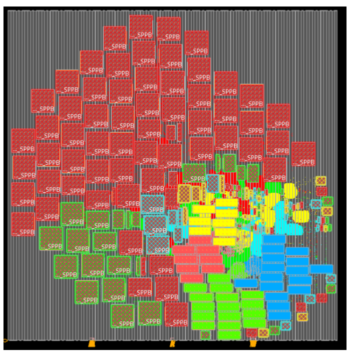
Manuscript 8 submitted to 2023 IEEE 66th International Midwest Symposium on Circuits and Systems. Received April 6, 2023.

TABLE I: Table comparing metrics after placement from industrial tool and after design space exploration

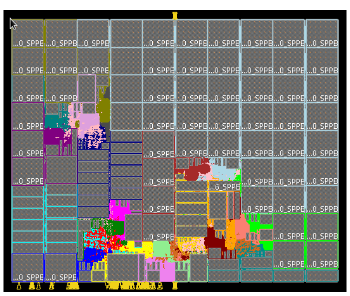
Build	**WNS**(ns)	**TNS**(ns)	**Wirelength**(μm)	**Utilization**(%)	**Area**(μm^2)	**Total Power** (mW)	**Placement Image**
Placement1 (Tool placement)	-0.01	-0.82	40495153	5.29	17119490	13.21	Image 8a*
Placement2 (DSE placement)	-0.03	-2.91	38248240	21.73	19172197	9.83	-
Placement3 (DSE placement)	0.00	-0.02	23138843	23.3	13320277	9.81	-
Placement4 (DSE placement)	-0.08	-32.85	25093516	42.61	13343707	9.88	-
Placement5 (DSE placement)	-0.03	-4.52	25021953	42.24	13336693	9.83	-
Placement6 (DSE placement)	-0.04	-3.87	24719999	25.34	13321562	9.81	-
Placement7 (DSE placement)	0.00	0.00	23528766	30.47	12799161	9.51	-
Placement8 (DSE placement)	-0.01	-0.10	24753101	36.16	12694027	9.43	-
Placement9 (DSE placement)	0.00	0.00	21730693	44.68	12684283	9.35	Image 8b**

*Figure 8a gives the memory macro placement done by the tool

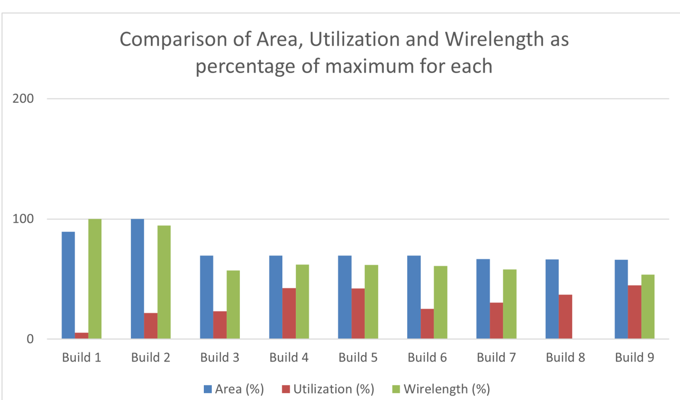
**Figure 8b gives the memory macro placement achieved after design space exploration



(a) Initial placement achieved by the tool



(b) Final placement achieved by DSE



(c) Bar graph comparing Area, Utilization and Wirelength

Fig. 8: Experimental Results

Wirelength for different placement DEF (Design Exchange Format) files for the memory macro placement that were created as DSE progressed. As can be seen from Table I, DSE led to an area improvement of about 25%, a utilization improvement of 8X and wirelength requirement that reduced to half of the tool placed wirelength. Table I also presents the worst negative slack (WNS) and total negative slack which shows that the timing is not achieved for the tool placement (negative slack) but DSE leads to timing being met. Images for the initial tool placement and the final placement achieved after DSE are presented in figures 8a and 8b. On a visual inspection of these images, it can be inferred that DSE leads to a much better memory macro placement as compared to memory macro placement achieved by the tool. Figure 8c presents the comparison of Area, Utilization, and Wirelength as a percentage of the maximum value of each metric. It shows that the area and wirelength kept decreasing steadily as iterative improvements were done in the memory macro placement (DEF). Utilization shows a similar trend but with a peak at placements 4 and 5 because of a congested placement which led to better utilization but with poor timing. All the above results were obtained in a span of two months due to the faster design space exploration time possible because of faster hand-placement. The faster hand-placement was enabled by the relative-placement script which improved DSE design cycle time from 2-3 weeks for part of the design to 3-4 days for the complete design.

V. Conclusion

This paper presented a DSE that showed that improvements in the area, wirelength, and utilization can be achieved through manual placement in shorter amount of time. In our research, DSE led to a 25% area improvement achieved within 3-4 days which is a large saving in terms of cost.

ACKNOWLEDGMENT

The authors would like to thank Aarish Technologies, Canadian Microelectronics Corporation, and Mitacs for supporting this research.

References

[1] A. Vidal-Obiols, J. Cortadella, J. Petit, M. Galceran-Oms and F. Martorell, ”RTL-Aware Dataflow-Driven Macro Placement,” 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2019, pp. 186-191, doi: 10.23919/DATE.2019.8714812.

[2] A. B. Kahng, R. Varadarajan, and Z. Wang, ‘RTL-MP: Toward Practical, Human-Quality Chip Planning and Macro Placement’, in Proceedings of the 2022 International Symposium on Physical Design, Virtual Event, Canada, 2022, pp. 3–11.

[3] Enrico, R.; Marco, R.; Anna, M. N. et al.: Pareto Optimal Design Space Exploration for Accelerated CNN on FPGA. In: 2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)

[4] Ghimire, D.; Kil, D.; Kim, S.-h. A Survey on Efficient Convolutional Neural Networks and Hardware Acceleration. Electronics 2022, 11, 945. https://doi.org/10.3390/electronics11060945

[5] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, ”Survey of Machine Learning Accelerators,” 2020 IEEE High Performance Extreme Computing Conference (HPEC), 2020, pp. 1-12, doi: 10.1109/HPEC43674.2020.9286149.

[6] Configurable Processor for implementing convolution neural networks, by P. Sinha (2021, Jan 28) Patent PCT/IB2020/000609 [Online]. Available: https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2021014215