# A 22nm 3.5TOPS/W Flexible Micro-Robotic Vision SoC with 2MB eMRAM for Fully-on-Chip Intelligence

Qirui Zhang, Hyochan An, Zichen Fan, Zhehong Wang, Ziyun Li, Guanru Wang, Hun-Seok Kim, David Blaauw and Dennis Sylvester
University of Michigan, Ann Arbor, MI, USA; Email: qiruizh@umich.edu

2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits) | 978-1-6654-9772-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/VLSITechnologyandCir46769.2022.9830340

**Abstract** We present a highly flexible micro-robotic vision SoC featuring a hybrid Processing Element (PE) for efficient processing of both Convolutional Neural Network (CNN) and non-CNN vision tasks with 2MB embedded MRAM for retentive fully-on-chip weight storage. Fabricated in 22nm, the design achieves 0.22nJ/pix for Harris corner detection (a non-CNN vision task) and 3.5TOPS/W (INT16) for CNN, a 60% efficiency improvement over state-of-the-art NVM-based NN ASICs.

**Introduction:** Autonomous *micro-robots* rely on navigation systems (Fig. 1) that are typically based on cameras and inertial sensors. This involves heavy vision workloads [1] such as feature detection/tracking and depth estimation. On CPUs and GPUs, these tasks consume > 30W [1], greatly exceeding actuation power of *micro-robots* [2]. To achieve low-power robotic vision, prior work [3][4][5] targets each vision task with a dedicated accelerator, making them unable to adapt to algorithm changes and requiring a substantial collection of accelerators at the cost of high design effort. In addition, with the rise of CNN-based vision [6], it is now essential that vision processors execute both non-CNN and CNN vision tasks efficiently.

Targeting efficiency of four major vision tasks in Fig. 1 that are sequentially executed in a navigational vision pipeline, we propose a novel hybrid PE in which non-CNN vision tasks achieve high efficiency by using a *2D-mapping* architecture [7] while CNN is executed in an efficient *output-channel-parallel* systolic manner [8]. Combining both processing domains into a single PE array future-proofs the architecture, facilitating next generation CNN-heavy vision algorithms, while saving 40% area and leakage with <0.5% power overhead and no throughput loss, compared to two separate array implementations. To further improve energy efficiency, the design implements a number of key features: 1) 2MB MRAM for non-volatile fully-on-chip weight storage; 2) A unified Image-Activation Memory (IAMEM) with block-swapping-based input/output image buffering that reduces buffer footprint by 50% and eliminates data copy for multi-frame buffering; 3) A combination of weight buffering and CNN loop ordering that reduces measured MRAM read power by 88.4%.

**Chip Architecture:** Fig. 2 shows the top-level architecture consisting of a programmable Neural Vision Processing Unit (NVPU) that accelerates vision tasks, 2MB embedded MRAM, 936kB IAMEM, a Cortex-M33 sub-system with 256kB I/D memory, a SPI slave, and an input image interface.

**Unified Image-Activation Memory:** Conventionally, an image processor uses ping-pong buffers for image input/output and separate scratchpads for intermediate results. However, in vision tasks such as feature tracking, multiple frames need to be buffered and accessible to computation. With ping-pong buffering, the first frames need to be loaded and then copied elsewhere to free up buffers for subsequent frames, resulting in copying overhead. Instead, we propose an architecture with unified buffers and scratchpads. It consists of six physical memory blocks that can be swapped (through MUX-based re-routing) to any of six different logical blocks (Fig. 2, top right). Logical block 0 is dedicated to input and block 5 to output while the other four are accessible to the NVPU. In the case of 2-frame buffering, each frame can be loaded in-turn through logical block 0 and then swapped to logical blocks 1 and 3, after which both are available for processing. This flexible scheme avoids data copying, allows memory reuse, and reduces overall buffer size by 50% in our design.

**Neural Vision Processing Unit:** The bottom of Fig. 2 details the NVPU architecture, which centers around a hybrid PE array tightly coupled with a custom RISC core, making it instruction-programmable. The PE array combines two modes: *2D-mapping* -based Image Processing Mode (IPM) and Systolic CNN Mode (SCM). In IPM (Fig. 3), image blocks are loaded/stored row-by-row to/from the PE array, where each PE maps one input/output pixel. IPM is optimized for efficient image filtering or other pixel-block-based kernels, where output pixels stay stationary in the PE while input pixels are shifted around in a zig-zag fashion. Each cycle, one weight from the 2D filter is broadcast to all PEs and multiplied with input pixels, with the results accumulated in output pixels. Unlike IPM, in SCM (Fig. 4 top) each PE row maps weights of different output channels, while it still shares similar IAMEM access and remains output stationary. Input activations and weights are unrolled across input channels and kernel sizes and streamed in systolic (non-zig-zag) fashion to the array. In contrast to the image LD/ST latency in IPM, SCM convolution is fully pipelined and incurs no memory latency once set up.

**Hybrid Systolic 2D-Mapping PE:** Fig. 5 shows the combined 2D-Mapping and systolic convolution PE. By merging the Shift-Reg with ActOut-Reg and reusing MAC and Acc-Reg, area and leakage are reduced by 40% with <0.5% power overhead compared to a design using two separate PE arrays. To enhance flexibility, the MAC unit is extended to serve as an ALU with programmable functions (MAC/add/mult/shift/cmp, etc.) for pixel-wise operations. A Local Register File (LRF) enables intermediate results to remain in the PE array for fused computation, which significantly reduces IAMEM access and footprint. For example, Harris corner detection requires multiple filters and pixel-wise operations; for this task IAMEM accesses and footprint can be reduced by at least 14× and 3×, respectively, using fused computation. The mask register (Msk-Reg) enables SIMT-style masked image processing, where masked output pixels are not processed. For a VGA image with 60 tracked features and feature masking radius of 30 pixels, 43% of pixels can be masked for new feature detection on average. Gating their ALUs, LRFs, and Acc-Regs yields 17% power reduction for Harris corner detection at 0.52V.

**MRAM Read Activity Minimization:** The bottom of Fig. 4 depicts the proposed MRAM-read minimization technique to reduce MRAM power. To keep PEs busy in SCM, different weights need to be streamed in every cycle, potentially leading to 100% MRAM read activity and correspondingly high MRAM power, especially in the MRAM's I/O supply domain (1.6V ~ 3.6V). Ping-pong weight buffers (Fig. 4, top left) allow weights for a set of 16 output channels to be reused. By moving the output channel to the outer loop, weights can also be reused for the input feature map and MRAM read activity is minimized. For instance, for a 96×96×16 to 48×48×64 convolution (3×3 kernel), MRAM read activity is reduced by ~95×.

**Measurements**: The proposed chip was fabricated in 22nm and measured at room temperature. In non-NN vision tasks, the design achieves 0.22nJ/pix for Harris corner, 0.22nJ/pix for sparse LK flow, and 0.055nJ/pix for stereo local matching. The chip exhibits 207GOPS INT16 peak image processing performance. Fig. 8 shows a demonstration of the chip detecting Harris corners from an image in the EuRoC dataset. Shown in Fig. 6 (top left), the proposed read activity minimization technique reduces MRAM read power by 88.4% at 0.8V. Fig. 6 (top right) shows the voltage-frequency-efficiency scaling of SCM measured by iteratively running 48×48×32 to 24×24×32 (3×3 kernel) CONV-BN layers. Our design achieves peak 146GOPS INT16 (511GOPS INT8 after normalization) CNN performance. It achieves peak energy efficiency of 3.5TOPS/W INT16 (12.1TOPS/W INT8 after normalization) at 0.5V, 16MHz. Fig. 6 (bottom) shows the performance, energy efficiency, and demonstration of mapping DroNet [6] fully on-chip for learning-based collision avoidance. Fig. 7 shows the performance and energy efficiency of mapping a complete navigational vision pipeline on chip with the four major vision tasks executed sequentially. Fig. 10 gives a chip summary and detailed comparison with state-of-the-art designs across four major robotic vision tasks. Despite its high flexibility, our chip exhibits better or similar energy efficiency when compared to accelerators for non-NN tasks. For NN inference, our design achieves a 60% inference efficiency improvement over a state-of-the-art NVM-based NN ASIC (with INT8 MACs), even when performing INT16 MACs. After normalization to INT8, our chip achieves 5.5× efficiency, showing the potential advantage of the proposed hybrid architecture. Fig. 9 provides the die photo with floorplan.

**Acknowledgement:** We thank Arm Ltd. for supporting this work and TSMC University Shuttle Program for chip fabrication.

**References:**

[1] Y. Lin *et al.*, J. Field Robot. 2018. [2] N. T. Jafferis *et al.*, Nature 2019.
[3] A. Suleiman *et al.*, JSSC 2019. [4] J. Narinx *et al.*, VLSI 2017.
[5] Z. Li *et al.*, ISSCC 2019. [6] A. Loquercio *et al.*, RA-L 2018.
[7] Z. Du *et al.*, ISCA 2015. [8] X. Wei *et al.*, DAC 2017.
[9] S. Smets *et al.*, ISSCC 2019. [10] D. Rossi *et al.*, ISSCC 2021.
[11] M. Giodano *et al.*, VLSI 2021. [12] M. Horowitz *et al.*, ISSCC 2014.

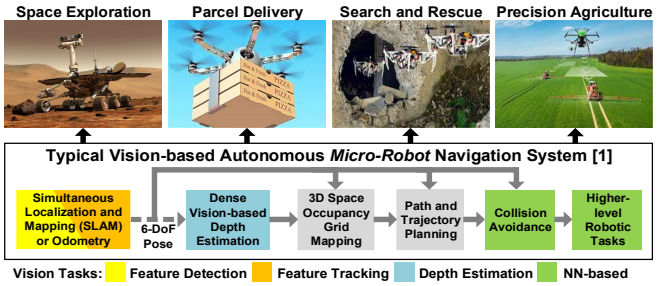



Fig. 1. Four major vision tasks for *micro-robot* navigation.

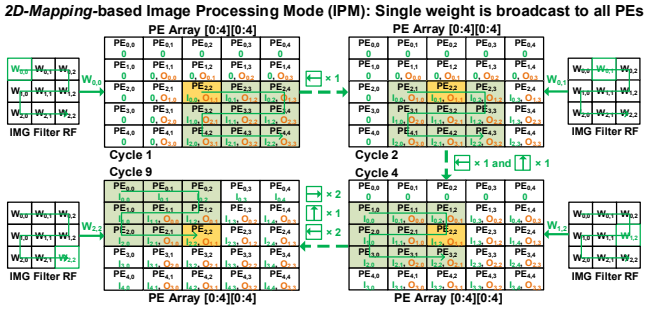



Fig. 3. *2D-Mapping*-based IPM architecture and data flow.

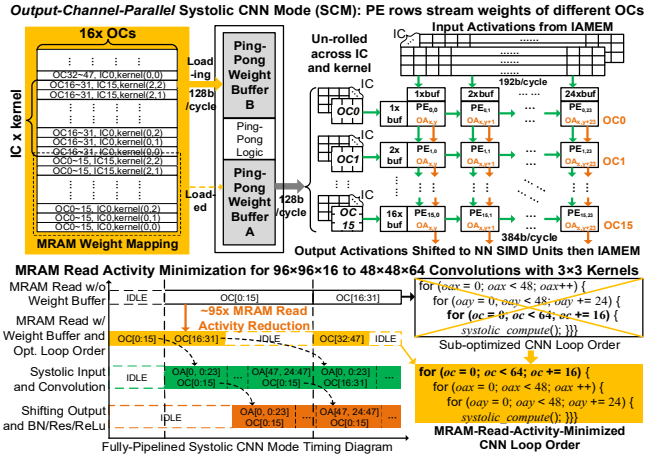
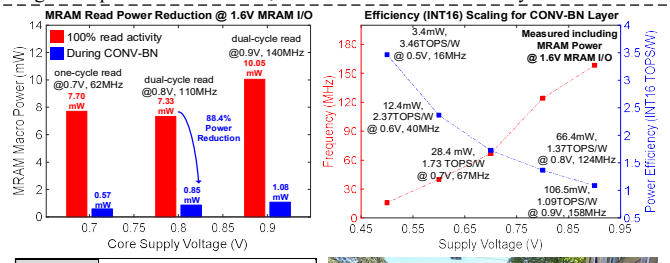



Fig. 4. Top: SCM architecture; Bottom: MRAM read activity minimization.




| | |
|---|---|
| Application | DroNet [6]: Learning-based Collision Avoidance |
| Network Architecture | ResNet-8 variant with full pre-activation residual blocks |
| Parameter Size | 337KB mapped fully on MRAM with pre-trained model |
| Input Size | 200×200 |
| Output (Regression results) | $P_{coll}$: collision probability $\theta_{steer}$: steering angle to avoid collision |
| Best Efficiency | 53fps, **2.6mW** and **49.06uJ/frame**, @ 0.52V Core, 1.6V MRAM I/O and 18MHz |
| Highest Performance | **554fps**, 78.38mW and 141.5uJ/frame @ 1.0V Core, 1.6V MRAM I/O and **187MHz** |

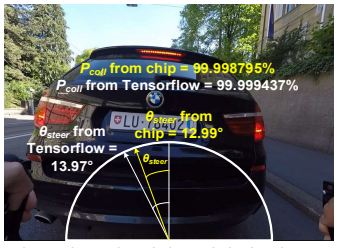



Fig. 6. Top: MRAM power reduction through read activity minimization and voltage-frequency-efficiency scaling for CONV-BN layer (80% weight sparsity, 50% activation sparsity); Bottom: performance, efficiency and demonstration of mapping DroNet [6] fully on chip.

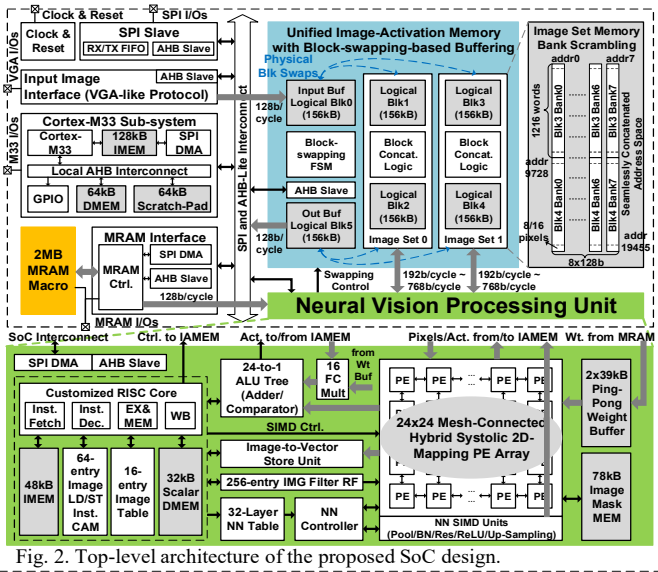






Fig. 2. Top-level architecture of the proposed SoC design.

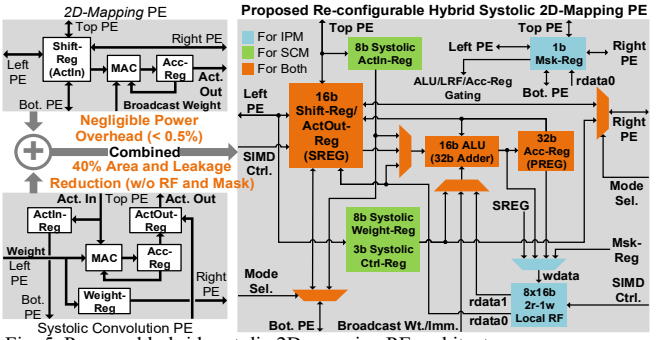
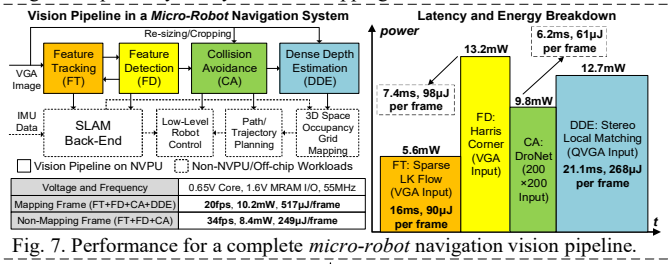



Fig. 5. Proposed hybrid systolic 2D-mapping PE architecture.




| | |
|---|---|
| Voltage and Frequency | 0.65V Core, 1.6V MRAM I/O, 55MHz |
| Mapping Frame (FT+FD+CA+DDE) | **20fps**, **10.2mW**, **517μJ/frame** |
| Non-Mapping Frame (FT+FD+CA) | **34fps**, **8.4mW**, **249μJ/frame** |

Fig. 7. Performance for a complete *micro-robot* navigation vision pipeline.

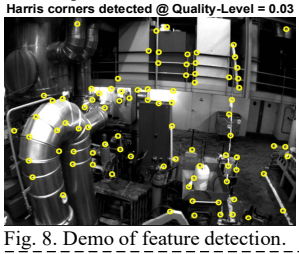



Fig. 8. Demo of feature detection.

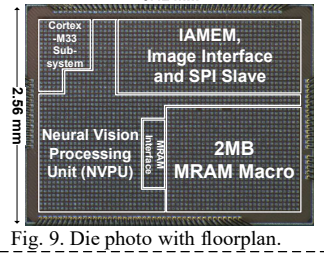



Fig. 9. Die photo with floorplan.

| | VLSI '17 [4] | JSSC '19 [3] | ISSCC '19 [9] | ISSCC '21 [10] | VLSI '21 [11] | This Work |
|---|---|---|---|---|---|---|
| Technology | 28nm CMOS | 65nm CMOS | 22nm FDSOI | 22nm FDSOI | 40nm CMOS | 22nm CMOS |
| Application | Multi-View Depth | Visual-Inertial Odometry | Image Processing | DNN for IoT | Edge NN Inference and Weight Tuning | **CNN and Non-CNN Vision for *Micro-Robot* Navigation** |
| Processing Architecture | Dedicated Accelerator | Dedicated Accelerator | Streaming-based CGRA | CMP and NN Accelerator | Systolic Array | Hybrid Systolic 2D-Mapping PE Array |
| Programmability | Not programmable | Not programmable | Only for Image Processing | General Purpose and NN | Only for NN | **General Purpose, Image Processing and NN** |
| Die Area | 5.96mm² | 20mm² | 4.9mm² | 12mm² | 29.2mm² | 8.76mm² |
| SRAM | 582.5 kB | 854 kB | 690 kB | 1728 kB | 0.5 MB | 1428 kB |
| On-Chip NVM | N/A | N/A | N/A | 4MB MRAM | 2MB RRAM | **2MB MRAM** |
| Voltage | 0.9V | 1.0V | 0.8V | 0.5 ~ 0.8V | 1.1V | 0.5 ~ 1.0V |
| Frequency | 300MHz | 62.5MHz | 5 ~ 220MHz | 32kHz ~ 450MHz | 200MHz | 56kHz ~ 190MHz |
| Power | 380mW | 24mW | 10.7 ~ 101.4mW | 1.7μW ~ 49.4mW | 126mW | 468μW ~ 158mW |
| Peak Image Proc. Perf. | Not Reported | **59.1GOPS**[1] | **145GOPS**[2] | Not Reported | N/A | **207GOPS**[2] @ 1.0V, 180MHz |
| INT8 Peak NN Perf. | N/A | N/A | | **32.2GOPS**[3] **(NN Accelerator)** | **920GOPS**[3] | **511GOPS**[3] @ 1.0V, 190MHz (146GOPS[2] INT16) |
| Feature Detection Efficiency | N/A | **0.33nJ/pix**[1] Shi-Tomasi corner and **1.40nJ/pix**[1] sparse LK flow, 71fps Wide-VGA (752x480) | Not Reported | Not Reported | N/A | **0.22nJ/pix**, 3.4mW, 50fps VGA, Harris corner @ 0.52V, 20MHz |
| Feature Tracking Efficiency | N/A | | **1.16nJ/pix**, 30fps VGA, dense LK flow | Not Reported | N/A | **0.22nJ/pix**, 1.6mW, 23fps VGA, sparse LK flow (≤ 100 features) @ 0.52V, 20MHz |
| Dense Depth Estimation Efficiency | **0.042nJ/pix**, 32fps 2K, stereo local matching | N/A | Not Reported | Not Reported | N/A | **0.055nJ/pix**, 10.9mW, 10fps VGA, stereo local matching (depth level = 64) @ 0.62V, 50MHz |
| INT8 NN Inference Efficiency | N/A | N/A | Not Reported | **1.3TOPS/W**[3, 4] **(NN Accelerator)** | **2.2TOPS/W**[3, 4] | **12.1TOPS/W**[3, 5] (3.5TOPS/W[2, 5] INT16) |

[1] OP definition not specified. Energies calculated from Fig. 11 and Table III in [3]. [2] 16b/32b mult/add = 1OP. [3] 8b MAC = 2OPs. Our 16b MAC (16b mult, 32b add) is normalized to 7OPs as per [12]. [4] Sparsity not specified. [5] 80% weight sparsity and 50% input activation sparsity.

Fig. 10. Chip summary and comparison with state-of-the-art.