MANIC: A 19µW @ 4MHz, 256 MOPS/mW, RISC-V microcontroller with embedded MRAM main memory and vector-dataflow co-processor in 22nm bulk finFET CMOS

Graham Gobieski*, Oguz Atli†, Cagri Erbagci†, Ken Mai†, Nathan Beckmann*, Brandon Lucia†
*Computer Science Department, †Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, USA
gobieski@cmu.edu, {aatli, cerbagci, kenmai, blucia}@andrew.cmu.edu, beckmann@cs.cmu.edu

***Abstract*—Whether powered by a battery or energy harvested from the environment, low-power (LP) sensor devices require extreme energy efficiency. These sorts of devices are becoming pervasive, running increasingly sophisticated applications in inhospitable environments. We present MANIC, an energy-efficient microcontroller (MCU) augmented with a vector-dataflow (VDF) co-processor. The testchip taped out on a 22nm bulk finFET CMOS process demonstrates that MANIC is 60% more energy-efficient than a baseline, scalar, low-power MCU, achieving peak efficiency of 256 MOPS/mW (2.6× prior work) while consuming only 19.1µW (@4MHz). To make the system viable for intermittently powered applications that require non-volatile storage, MANIC includes a 256KB embedded MRAM.**

I. INTRODUCTION

Emerging sensing applications demand extremely energy-efficient data processing in remote environments, e.g., for on-device machine learning or signal processing. Unfortunately, existing digital programmable microcontrollers (MCUs) waste energy on instruction supply (i.e., fetch, decode, and control) and data movement. In this work we describe MANIC, the first digital integrated circuit implementation of the *vector-dataflow* [2] execution model. Vector-dataflow execution substantially reduces the energy costs of instruction supply and data movement, significant improving energy efficiency without compromising programmability.

Fig. 1 illustrates the differences among execution models of a baseline low-power scalar MCU, vector MCU, and MANIC's vector-dataflow. Blue arrows denote data flow, and orange arrows denote control flow.

- A **scalar** MCU performs one operation per instruction, burning energy for instruction fetch, decode, and control and for register-file (RF) accesses to load operands and store results. These instruction and data-movement overheads consume the majority of the entire system's energy.
- A **vector** MCU performs multiple operations per instruction, repeating the same operation to each vector element. Vector operation amortizes the energy cost of instructions across many operations, but each operation still reads from and writes to the vector register file (VRF). In fact, since the VRF is much larger than a scalar RF, data movement energy increases in the vector design.
- Like vectors, MANIC's **vector-dataflow (VDF)** execution model amortizes instruction fetch costs over many operations, but VDF also minimizes VRF energy by avoiding VRF accesses. To avoid VRF accesses, VDF dynamically analyzes a *window* of vector instructions, identifying how data flows between instructions. VDF then forwards values directly from a producer instruction to a consumer instruction, without writing those forwarded, intermediate values into the VRF.

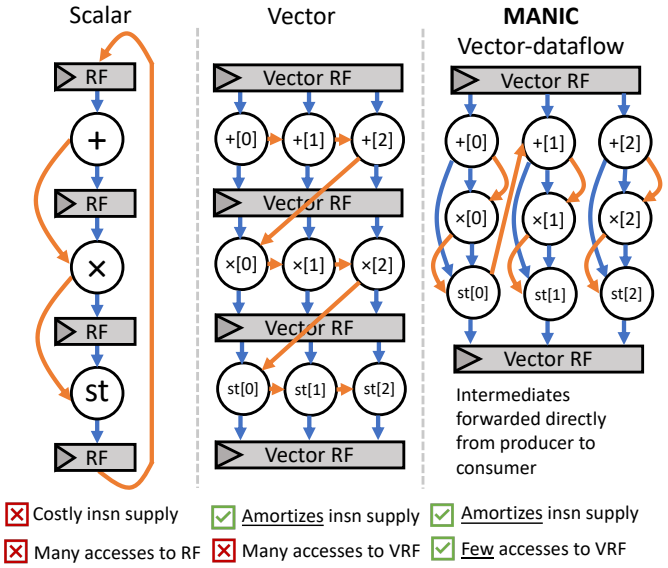


Figure 1: Comparison of execution models. In MANIC's *vector-dataflow execution*, vectors reduce instruction-supply energy and dataflow forwarding reduces data-supply energy.

MANIC is designed for on-device processing in energy-constrained sensor devices. MANIC implements VDF as a coprocessor to a low-power RISCV microcontroller in commercial 22nm bulk finFET CMOS. The design integrates embedded MRAM main memory, providing non-volatility required by batteryless and intermittent computing applications [5]. MANIC achieves an active operating power consumption of 19μW at 4MHz. MANIC is the first silicon implementation of VDF. MANIC's VDF implementation substantially improves over a highly-optimized low-power baseline, eliminating 60% of the base design's energy consumption, with a peak execution efficiency of 256MOPS/mW, a 2.6× improvement over the state-of-the-art.

II. EXECUTION MODEL AND MICROARCHITECTURE

Fig. 2 shows MANIC's microarchitecture, split along the two phases of execution: Decode & Rename and Execute. MANIC executes vector instructions, not scalar instructions, so fetches far fewer instructions from memory compared to a scalar system. To minimize power and energy, MANIC executes

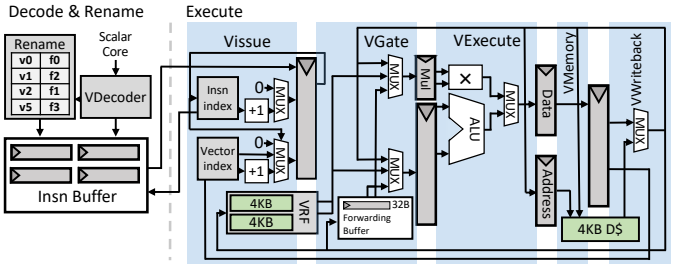


Figure 2: Pipeline diagram of MANIC VDF coprocessor. The Decode&Rename phase identifies dataflow dependencies between operations, buffers decoded instructions in the Insn Buffer, and renames operands. The Execute Phase has five pipelines stages: VIssue (state machine tracking progress), VGate (gathers operands), VExecute, VMemory, and VWriteback.

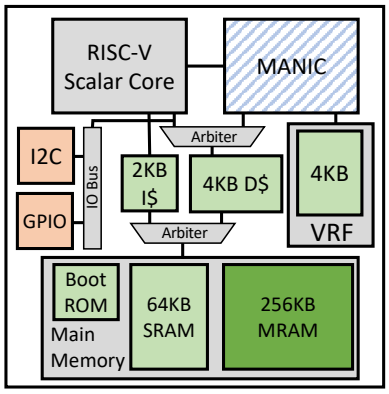


Figure 3: Block diagram of MANIC. MANIC contains a RV32IMEC MCU, the VDF coprocessor connected to a 4KB VRF, instruction and data caches and a 64KB SRAM + 256KB MRAM main memory

vector instructions one element at a time — it does *not* process vector elements in parallel. This design choice is a significant microarchitectural distinction from prior vector designs, which process entire vectors in parallel to maximize performance. The CPI is approximately 1 * vector length, depending on memory.

During the Decode & Rename Phase, MANIC buffers a window of decoded instructions ("Insn Buffer") and identifies dataflow between them. Instructions do not begin execution until a sufficient number have been buffered and analyzed (see next paragraph). Similar to register renaming in high-performance, out-of-order cores, MANIC tracks dataflow using a rename table that records the last instruction to have written each vector register. The rename table is a sixteen entry table composed of flops (144b) that holds the index (the "name") of the register operand. When decoding an instruction, MANIC checks the rename table to identify the producer instructions for the current instruction. If the producer instruction(s) are in the Insn Buffer, MANIC renames the current instruction's operands to point to the "Forwarding Buffer" (a 32B buffer) instead of the 4KB VRF, allocating space in the Forwarding Buffer for producer(s) if necessary. Vector instructions are also annotated with "Kill" hints that indicate that they are the final consumer of a vector register. When a Kill hint is seen, MANIC disables writeback for producer instruction(s), since their outputs need not be recorded in the VRF.

The Execute Phase begins once the instruction buffer is full, a `vfence` instruction is reached, or the forwarding buffer is fully allocated. MANIC has a five-stage execution pipeline. Unlike conventional vector execution, MANIC's VDF executes a *series of operations per element* before going to the next element (Fig. 1). This execution order minimizes the number of in-flight values so that they fit in the 32B Forwarding Buffer. VIssue tracks execution progress and maintains a pointer into the instruction buffer, decodes instructions, and (only if necessary) initiates VRF reads; VGate determines the source for each operand (the VRF, Forwarding Buffer, or bypass paths) and steers operands to the multiplier or ALU; VExecute computes the ALU and multiplier results; VMemory issues loads and

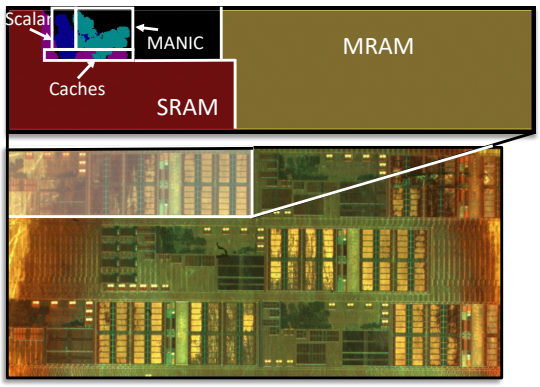


Figure 4: Die shot of the MANIC testchip taped out on a 22nm bulk finFET CMOS process. The dimensions of the MANIC block are 1.600 mm x 0.355 mm, and the logic, SRAM and MRAM power domains take up an area of 0.09 mm^2, 0.15 mm^2 and 0.33 mm^2 respectively.

stores; and VWriteback writes results to the Forwarding Buffer or VRF, as appropriate. VGate reduces switching activity in VExecute by steering operands to dedicated input registers for the ALU or multiplier to, e.g., prevent a VADD from toggling the multiplier. This is important because, unlike conventional vector execution, the active instruction changes every cycle in MANIC, increasing activity on control and data signals.

III. TESTCHIP

Fig. 3 shows a block diagram of the MANIC system-on-chip. MANIC comprises a low-power-optimized RISC-V (RV32IMEC) MCU, 2KB instruction cache, 4KB data cache (shared with the coprocessor), IO-bus supporting I^2C and GPIO, main-memory (1KB boot ROM, 64KB SRAM, and 256KB MRAM), and VDF co-processor with 4KB 1r1w VRF. MANIC can tolerate a single read port in the VRF because dataflow forwarding significantly reduces read pressure on the VRF, since most values come from the Forwarding Buffer.

MANIC was fabricated in a 22nm bulk finFET process with an area of 0.57mm^2. Fig. 4 shows the die photo of the 4mm $\times$ 8mm testchip containing multiple experiments with

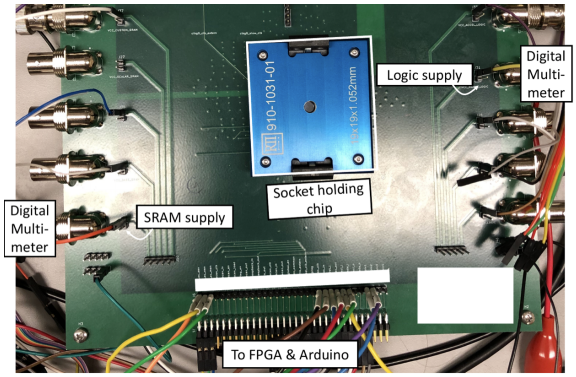


Figure 5: The setup used to test the MANIC testchip. An Arduino and an FPGA are used to drive the signal IO and digital multimeters wired in series to the power supplies are used to measure current.

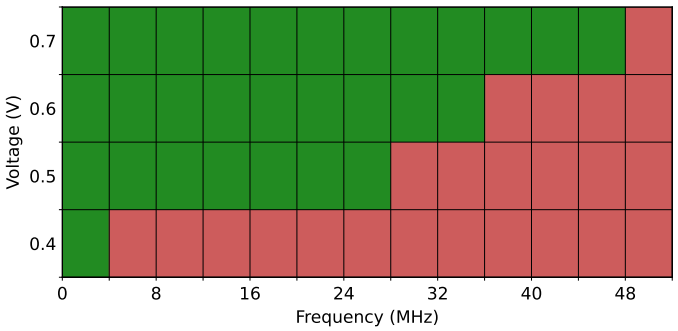


Figure 6: Shmoo plot of the MANIC testchip.

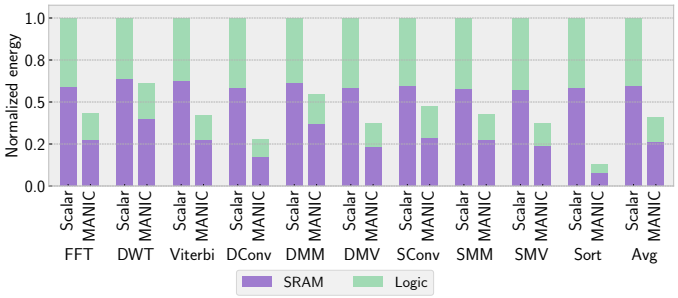
MANIC highlighted. MANIC has separate power domains for SRAM, MRAM, and logic that can be controlled and measured independently. The dimensions of the MANIC block are 1.600 mm x 0.355 mm, and the logic, SRAM and MRAM power domains take up an area of 0.09 mm^2, 0.15 mm^2 and 0.33 mm^2 respectively. MANIC is optimized to run with a 4MHz to 50MHz clock from an on-die clock generator at 0.4V to 1.0V logic, 0.4V to 1.0V SRAM, and 1.10V MRAM.

IV. EVALUATION

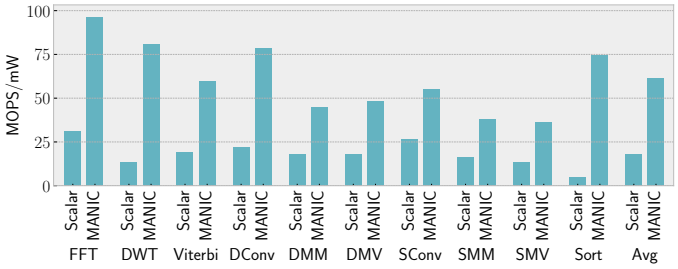
Fig. 5 shows the test setup that we use to verify functionality and gather energy data from the MANIC testchip. We use an Arduino to communicate with the I^2C bus. This interface handles loading the program memory and listen to the prints executed by MANIC. We debug the design and check system state using a PYNQ-Z1 FPGA board through the debug scan interface. We measure power draw with high sensitivity using digital multimeters (Agilent 34405a, Agilent 34410a) wired in series with the MANIC logic and SRAM power supplies.

We evaluate MANIC across ten benchmarks with random 32b inputs. Fig. 7a shows energy normalized to the baseline low-power scalar MCU, and Fig. 7b shows energy efficiency (MOPs/mW). Except where otherwise noted, all results were collected with MRAM disabled and core voltage at 0.4V. On average, MANIC reduces energy by 60% vs. the baseline and achieves 60MOPS/mW at 19µW.

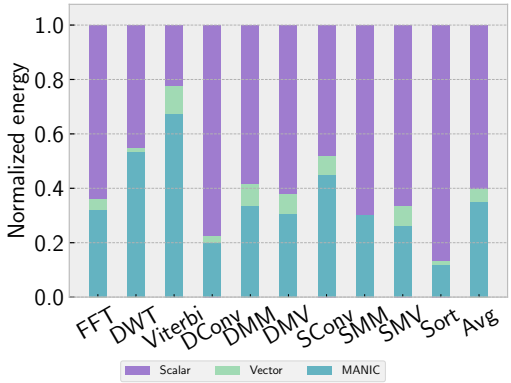
MANIC is energy-efficient: Table I compares MANIC with prior work [1], [3], [4], [6]. MANIC operates from 0.4V to 1.0V with a frequency from 4MHz to 50MHz (Fig. 6). MANIC was designed for energy-minimal, low-power operation: MANIC consumes 19µW at 4MHz, significantly lower than prior work. MANIC is more energy-efficient than prior work (by 2.6×), with a peak efficiency of 256 MOPS/mW (on vector increment) and 3.7pJ/cycle at 0.4V, 4MHz, room temperature, and MRAM disabled. (Note that MANIC performs 32b ops, whereas some prior work performs narrower 16b or 8b ops.) With random inputs, which cause unrealistic, near worst-case toggling of data lines, MANIC gets 45 MOPs/mW on dense matrix-matrix multiplication (DMM).



(a) *Normalized energy consumption of benchmark applications running on the scalar RISC-V core versus on the MANIC VDF co-processor.*



(b) *Energy efficiency (MOPS/mW) of the benchmark applications running on scalar core vs MANIC.*



(c) *Normalized energy consumption of the scalar core, low-energy vector co-processor and MANIC VDF co-processor*

Figure 7: Energy and efficiency evaluation of MANIC.

	2017 [4]	2018 [3]	2019 [1]	2020 [6]	This work
Architecture	Scalar & Vector	Scalar	Scalar	Scalar w/ SIMD ext.	Scalar & Vector-dataflow
ISA	RISC-V	Thumb-2	Thumb-2	Thumb-2	RISC-V
Process (nm)	28 FD-SOI	14 Tri-gate	28 FD-SOI	65 LP	22 bulk FF
Core Area (mm2)	1.07	6.25	0.675	6	0.57
Voltage (V)	0.48-1.0	0.4-1.0	0.4	0.4-0.75	0.4-1.05 Core 1.1 MRAM
Frequency (MHz)	20-797	0.2-950	40-80	0.8-38	4-48.9
Memory (KB)	56KB SRAM	64+64+384 SRAM	32+32 SRAM	128 ROM, 16+4 SRAM	64 SRAM, 256 MRAM
Power Budget (mW)	1-200	1-20	1	1-4	0.019-2 w/o MRAM 1-2 w/ MRAM
Average Power (µW)	50000	80	144	47	19.1 w/o MRAM @ 4MHz[1] 1.7mW w/ MRAM @ 49MHz[1]
Peak Efficiency (MOPS/mW)[2]	41.8 MFlops/mW	Not reported	97	Not reported	256 w/o MRAM 33.2 w/ MRAM
Best Active Energy (pJ/Cycle)	Not reported	6.2	3	10.9	3.7 w/o MRAM 29 w/ MRAM

[1] Over all benchmarks [2] 32b operations

Table I: MANIC vs. prior work. MANIC is 2.6× more efficient than prior work, achieving 256 MOPS/mW (@19μW & 4MHz).

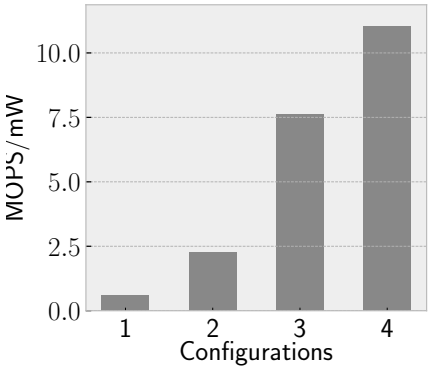
Vector-dataflow uses less energy than vector: To make a fair comparison between VDF and vector execution models, we also taped out an alternative SoC design with an optimized vector co-processor in the same process technology. Fig. 7c overlays the normalized energies of MANIC (Blue), the vector design (Green), and the scalar baseline (Purple). Both the vector design and MANIC achieve state-of-the-art energy-efficiency; Vector execution already achieves state-of-the-art energy-efficiency, using 54% less than the scalar MCU; MANIC's VDF execution reduces energy by a further 12%.

MRAM characterization: MANIC contains a 256KB embedded MRAM to make the system viable for intermittently powered applications that require non-volatile storage. Fig. 8 characterizes the embedded MRAM and presents a case study of designs with MRAM enabled. MRAM leakage is 663µW, reads take 170ns and 13.7pJ/bit, while writes take 8.4µs and 929pJ/bit. Write latency is independent of clock frequency.

A case study of DMM puts these numbers into context. Fig. 8b includes several system configurations: 1) MANIC running out of MRAM with the DCache enabled @49MHz, 2) MANIC running out of MRAM as fast as possible @231MHz (this necessitates the DCache being disabled), 3) MANIC running from SRAM, DCache enabled, and MRAM enabled @49MHz, and 4) MANIC running as fast as possible @166MHz (w/o DCache) and MRAM enabled. Configuration 4 achieves max efficiency with 11MOPS/mW (compared to 46MOPS/mW when MRAM is disabled) and configuration 2 achieves max efficiency for running from MRAM with 2.3MOPS/mW. As found in prior low-power systems, MRAM's high static power is a significant challenge for energy efficiency, requiring fine-grain power gating of MRAM to avoid static power from severely degrading system efficiency. Addressing this challenge is the subject of future work.

Size (KB)	256
Area (mm^2)	0.31
Voltage (V)	1.1
Leakage (µW)	663
32b Read Latency @ 50 MHz (ns)	170
32b Write Latency @ 50 MHz (µs)	8.4
32b Read Energy (pJ)	437
32b Write Energy (nJ)	29.7
Read Energy (pJ/bit)	13.7
Write Energy (pJ/bit)	929

(a) MRAM characterization.



1: Running from MRAM, DCache enabled, 48.9 MHz, 0.64V Core
2: Running from MRAM, DCache disabled, 231 MHz, 1.0V Core
3: Running from SRAM, MRAM enabled, DCache enabled, 48.9 MHz, 0.64V Core
4: Running from SRAM, MRAM enabled, DCache disabled, 166 MHz, 1.0V Core

(b) MOPS/mW for DMM.

Figure 8: MRAM characterization and a case study on running DMM with different DCache and main memory storage configurations.

V. CONCLUSION

Extreme energy-efficiency is a requirement for low-power sensor devices deployed to inhospitable environments. This work presented MANIC, the first silicon implementation of vector-dataflow execution. MANIC-VDF draws just 19µW (@4MHz) and is 2.6× more efficient than prior work.

REFERENCES

[1] D. Bol *et al.*, "19.6 a 40-to-80mhz sub-4μw/mhz ulv cortex-m0 mcu soc in 28nm fdsoi with dual-loop adaptive back-bias generator for 20μs wake-up from deep fully retentive sleep mode," in *ISSCC*, 2019.

[2] G. Gobieski *et al.*, "Manic: A vector-dataflow architecture for ultra-low-power embedded systems," in *MICRO*, 2019.

[3] T. Karnik *et al.*, "A cm-scale self-powered intelligent and secure iot edge mote featuring an ultra-low-power soc in 14nm tri-gate cmos," in *ISSCC*, 2018.

[4] B. Keller *et al.*, "A risc-v processor soc with integrated power management at submicrosecond timescales in 28 nm fd-soi," *JSSC*, 2017.

[5] B. Lucia *et al.*, "Intermittent computing: Challenges and opportunities," in *SNAPL 2*, 2017.

[6] P. Prabhat *et al.*, "27.2 m0n0: A performance-regulated 0.8-to-38mhz dvfs arm cortex-m33 simd mcu with 10nw sleep power," in *ISSCC*, 2020.