

# A Compact Bit Serial Memory Cell for Adiabatic Quantum Flux Parametron Register Files

L. Camron Blackburn , Alex Wynn , Karl K. Berggren , *Fellow, IEEE*, and Neil Gershenfeld

**Abstract**—The Adiabatic Quantum Flux Parametron (AQFP) superconducting logic family is an attractive beyond-CMOS technology due to its extreme energy efficiency. As AQFP circuits become more complex and target applications scale to microprocessors and ASICs, a high performance and area efficient register-level memory design is essential. In this work, we design and implement a 3-bit memory cell based on a simple Set-Reset (SR) latch attached to a shift register feedback loop. This design is more compact than existing AQFP registers that encode data with Data-Enable logic and therefore require larger XOR-style gates for every 1-bit cell (Tsuji et al., 2017). Furthermore, we share layout design and simulation results for scaling this circuit to an 8-bit memory cell and a  $N\text{-row} \times M\text{-col} \times 8\text{-bit}$  register file. By using SR data encoding, we are able to join the cells in an array using more compact AND gates, performing cell-level addressing with two write demuxes and two read decoders. Projecting our initial designs to a 32-word by 8-bit register file, we find that our SR-loop memory array has a 26% decrease in area compared to other state-of-the-art designs.

**Index Terms**—Adiabatic logic, quantum flux parametron, superconducting digital logic, superconducting memory.

## I. INTRODUCTION

ONE of the major limiting constraints on data center compute intensity today is power demand. Superconducting computing hardware is a compelling alternative to CMOS due to near-adiabatic bit-level operations and mature fabrication process technology. In particular, the Adiabatic Quantum Flux Parametron (AQFP) [2] is an extremely energy efficient superconducting logic family with 1.4 zJ switching energy dissipation per Josephson junction [3], without taking into account cryogenic cooling overhead.

The AQFP can be thought of as a DC-SQUID with a shunt inductor,  $L_q$ , through its center that creates two possible superconducting loops to trap a single flux quanta, representing two digital logic states, as shown in Fig. 1(a). The state of the AQFP is controlled by two signals, the data signal,  $I_{in}$ , and the

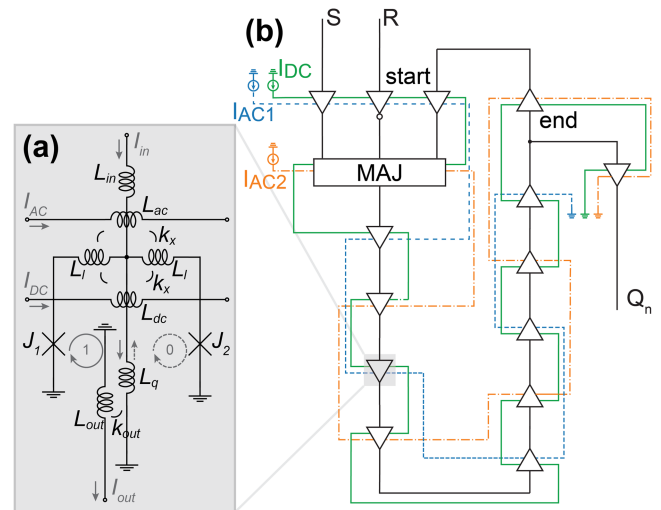


Fig. 1. (a) Schematic of a single AQFP buffer. The 1 (0) logic state is represented by a single fluxon in the left (right) loop which generates a positive (negative) current along the center shunt inductor,  $L_q$ . The device values are:  $L_{in} = 350\text{fH}$ ,  $L_{ac} = L_{dc} = 7.4\text{pH}$ ,  $k_x = 0.18$ ,  $L_l = 1.3\text{pH}$ ,  $L_q = 10\text{pH}$ ,  $L_{out} = 8\text{pH}$ ,  $k_{out} = 0.3$  (b) Schematic diagram of SR-Loop memory cell. A majority gate SR Latch at the input receives the Set (S), Reset (R), and previously stored bit ( $Q_{n-1}$ ) and propagates the output along a shift register storage loop. A four-phase clock is generated from two AC signals 90 degrees out of phase,  $I_{AC1}$  and  $I_{AC2}$ , and a DC bias,  $I_{DC}$ , and 1 b of information is stored across every four buffers. This design contains 34 JJs.

excitation signal, which is divided into a DC component,  $I_{DC}$ , and AC component,  $I_{AC}$ . Information is stored only when the magnitude of the excitation signal is large enough to cause the potential energy of the device to transition from a single-well landscape to a double-well landscape [4]. The AQFP is capable of extreme energy efficiency because it always remains at its potential minimum.

The performance metrics of a single AQFP device make it a compelling candidate for ultra low-power beyond CMOS computing; however, many integration and scaling concerns must be addressed for AQFP logic to be useful in state of the art high performance computing applications. A major concern for scaling superconducting logic systems is the lack of high density cryogenic memory. In the MANA AQFP microprocessor, the 16-word by 4-bit register file contributed the highest JJ count and largest footprint of all the pipeline stages in the chip [5]. There are register files from other superconducting logic families, such as RQL [6] and RSFQ [7] logic, that improve upon the bit density, however these circuits are built from devices that inherently

Received 25 September 2024; revised 26 December 2024 and 31 January 2025; accepted 3 February 2025. Date of publication 11 February 2025; date of current version 3 March 2025. This work was supported in part by the Under Secretary of Defense for Research and Engineering, Air Force under Contract FA8702-15-D-0001, and in part by the MIT Center for Bits and Atoms Consortium. (Corresponding author: L. Camron Blackburn.)

L. Camron Blackburn, Karl K. Berggren, and Neil Gershenfeld are with the Massachusetts Institute of Technology in Cambridge, Cambridge, MA 02139-4307 USA (e-mail: camronb@mit.edu).

Alex Wynn is with MIT Lincoln Laboratory in Lexington, Lexington, MA 02421 USA.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TASC.2025.3540048>.

Digital Object Identifier 10.1109/TASC.2025.3540048

TABLE I  
LOGIC TRUTH TABLE FOR SET-RESET LATCH

S	R	$Q_n$	Action
0	0	$Q_{n-1}$	Hold
0	1	0	Write 0
1	0	1	Write 1
1	1	$Q_{n-1}$	Hold

dissipate more energy than the AQFP and require additional circuitry to interface with AQFP logic.

The present study shares designs and experimental results of a serial memory cell that aims to address the area and bit density scaling concerns for low-level register files in AQFP processors. We then introduce designs and simulation for a hybrid serial and addressable array register file that is projected to provide a 35% increase in bit density by decreasing the area by 26% relative to the MANA register design.

## II. SR-LOOP MEMORY CELL

### A. Design

The SR-Loop memory cell is based on a majority gate SR latch [8], but modified with additional buffers in the feedback loop to store any desired length of data, rather than a single bit. As shown in Table I, when the Set and Reset signal values match, the previously stored data value,  $Q_{n-1}$ , is held. When the Set and Reset signals are different, the stored data is overwritten to match the Set signal.

The SR-Loop can be thought of as a ring oscillator with AQFP buffers that store a string of bits by indefinitely passing them along the loop. A schematic diagram for a 3-bit SR-Loop is shown in Fig. 1(b). The bits are addressable with the SR latch at the start of the loop and read out with a 1-to-2 splitter at the end of the loop. The bit depth is the number of bits simultaneously held in the storage loop, which is set by the number of buffers in the loop. To allow proper read and write operation, each bit must be temporally separated by a single clock cycle; therefore, in the case of a four-phase clock, the bit is spatially spread across 4 buffers. In Fig. 1(b), it takes 12 clock phases for a single bit to propagate from the start to the end of the storage loop and 1 b is written every 4 phases, so this design can hold 3 bits of information with a 3 clock cycle read latency. The read latency is the minimum time delay for a bit written to the loop to be available on the output,  $Q_n$ .

The SR-Loop has a non-destructive readout and a output bit is always available on the splitter. Therefore, one must keep track of timing to ensure that the bit on the output aligns with the bit intended to be read. The readout timing can be handled by a memory controller or hardcoded in the microarchitecture design, but this higher level of system integration design is beyond the scope of this paper and will be addressed in future work.

### B. Experimental Results

The 3-bit SR-Loop memory cell was fabricated in the MITLL SFQ5ee process [9] and tested at 4 K in a liquid Helium

immersion probe using an Octopux measurement system [10]. Although the circuits are simulated up to 5 GHz, the experimental testing was constrained to kHz frequencies due to limitations in testing equipment. The 3-bit SR-Loop has a  $240 \mu\text{m} \times 100 \mu\text{m}$  footprint with 34 JJs, although it was not designed for aggressive area optimization. The measurements were performed in a magnetically shielded environment with ambient field estimated around 100 nT. The experimental results are shown in Fig. 2.

Fig. 2(a) demonstrates the SR-loop writing every possible 3-bit sequence and storing it for two complete passes around the loop. As expected, the data takes 3 clock cycles to propagate to the output, and the output is aligned with the fourth clock phase, *i.e.* the negative peak of  $I_{AC2}$ . The output is read from the splitter buffer into a weakly coupled DC-SQUID amplifier. The readout SQUID is biased such that it is always in its voltage state, so the polarity of the output current pulse from the AQFP shifts the SQUID voltage in the positive or negative direction, which is then detected at room temperature by the Octopux.

Fig. 2(b) displays the clock amplitude margins. The same data pattern was evaluated for proper operation while the amplitudes of  $I_{AC1}$ ,  $I_{AC2}$  and  $I_{DC}$  were varied individually. The nominal operating point and clock margins were  $1.25 \text{ mA} \pm 31\%$ ,  $1.27 \text{ mA} \pm 32\%$ , and  $1.29 \text{ mA} \pm 15\%$  for  $I_{AC1}$ ,  $I_{AC2}$ , and  $I_{DC}$ , respectively.

### C. Scaling; 8-Bit SR-Loop

To motivate how best to scale the SR-Loop memory cell and build intuition for incorporating it into larger system designs, we will discuss its latency and energy trade-offs, and determine a reasonable cell bit depth. As discussed earlier, the bit depth of a single SR-Loop memory cell can be increased by increasing the length of the shift register feedback loop. However, the SR-Loop is a serial memory device, so as the bit depth grows so does the read latency. In many architectural applications where data retrieval is not perfectly random a serial memory can be beneficial for performance and energy efficiency, but the exact data segment length is application specific.

Additionally, the storage loop is actively clocked, with every buffer switching once per cycle, so as the storage loop grows in length, the energy dissipation per read grows at a faster than linear rate because the read latency increases. This is different from a passive storage device, such as an SFQ NDRO based register file [11] or SFQ delay-line memories based on passive transmission lines [12], where a persistent current loop stores a single bit per fluxon. Therefore, we can set a maximum SR-Loop bit depth by estimating when it become energetically favorable to use passive SFQ storage devices rather than the active AQFP storage loops. Assuming that a single SFQ write requires approximately  $\Phi_0 I_c$ ,  $1 \times 10^{-19} \text{ J}$  for  $50 \mu\text{A}$ , the energy dissipation is linear with bit depth,  $x$ , following  $E_{\text{SFQ}} = x \times 100 \text{ zJ}$ . On the other hand, the SR-Loop energy dissipation per bit is dependent on the read latency which is always equal to the bit depth since one bit is stored across one clock cycle. Assuming a 4-phase clock, this means the SR-Loop energy dissipation scales as  $E_{\text{AQFP}} = x \times (4 \text{ QFP/bit} \cdot 1.4 \text{ zJ/cycle/QFP} \cdot x \text{ cycle}) = x^2 \cdot 4 \cdot 1.4 \text{ zJ}$ . At small bit depths, it's energetically

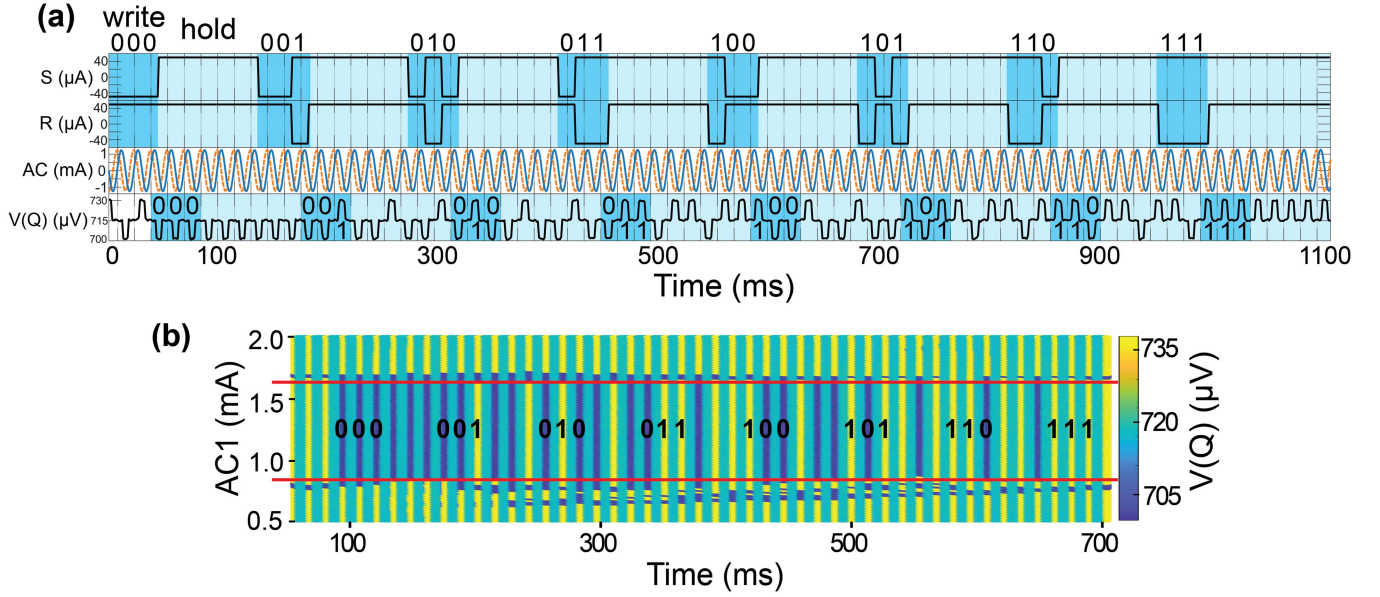


Fig. 2. (a) Basic circuit operation is verified by writing and storing every possible 3-bit sequence. The top two axes show the Set and Reset signal supplied to the circuit. The third axis plots both  $I_{AC1}$  (solid, blue) and  $I_{AC2}$  (dashed, orange) and verifies that the output is aligned with phase four, the minimum peak on  $I_{AC2}$ . The bottom axis plots the SR-Loop readout,  $V(Q)$ , amplified to the Octopus at room temperature with a DC-SQUID voltage readout. As expected, we see each bit sequence repeated 3 times - once when it is written and twice while storing. (b) The margins are obtained by individually sweeping the amplitude of each AC signal and the DC bias for all possible storage sequences (same data sequences as part (a), but modified for only one storage of each sequence to decrease testing time). The clock margin plot for only  $I_{AC1}$  is shown here as an example. Horizontal slices describe a single run of the data pattern for the AC amplitude specified on the y-axis. The output voltage state is read from the color map - blue regions represent a 0 logic state and yellow is 1. The limits of the correct operating region are indicated with a red line, e.g. the circuit works for  $I_{AC1}$  values between 0.859 mA and 1.633 mA. Margins for  $I_{AC2}$  and  $I_{DC}$  were extracted in the same way. The  $I_{AC1}$ ,  $I_{AC2}$ , and  $I_{DC}$  margins are  $\pm 31\%$ ,  $32\%$ , and  $15\%$ , respectively.

favorable to use the SR-Loop cell for storage, but beyond approximately 17 bits a passive SFQ storage device could be more energy efficient.

We will proceed by focusing on an 8-bit SR-Loop, with the target application being an 8-bit processor, such that a single SR-Loop stores one word. We have designed an 8-bit SR-Loop cell with 78 JJs, a  $150 \mu\text{m} \times 390 \mu\text{m}$  footprint, and 68.80 zJ/cycle simulated energy dissipation at 5 GHz clock operation. The energy dissipation is calculated with Cadence SPICE simulations by taking the sum of the continuous time integral of the current times voltage of each excitation line, and averaging over many clock cycles with random data input. This design is currently in fabrication and has not yet been experimentally tested. The 8-bit layout design is shown in Fig. 3(d).

### III. SR-LOOP ARRAY

It is impractical to scale a single SR-Loop memory cell into a large register memory, so we propose scaling by tiling the SR-Loop in a row-column addressable array. A schematic diagram for a 72-bit SR-Loop register file is shown in Fig. 3. In this design, we have a  $3 \times 3$  array of the 8-bit SR-Loop.

The S and R signals for a single bit write operation are passed through row and column demultiplexers to select one cell in the array. The demux schematic is shown in Fig. 3(a). The 2-bit select address in the demux determines which index will receive a 1 value from S or R, while keeping all other values 0. For example, to write a 1 in an SR-Loop at index 2 in the demux,

we set  $SR = 10$  and select  $ab = 10$ ; or to write a 0 to index 0,  $SR = 01$  and  $ab = 00$ . To generate a 2-dimensional array index, we have a demux for the row and column index, then the SR signals are passed through an AND gate at each SR-Loop cell. Proper write addressing was simulated with Cadence SPICE tools and the results are shown in Fig. 3(e). To non-destructively read a value, a readout address (following the same encoding as the write address) selects the cell with column and row decoders and the outputs are propagated through a merger tree composed of OR gates until the single desired output is obtained.

The 72-bit register file requires 1,510 JJs, including the write demuxes and readout decoders. There is a 10 clock cycle write latency, and 5 to 13 clock cycle read latency for the start to end of the word, respectively. 3.18 aJ energy dissipation per cycle at 5 GHz clock operation was extracted from simulation, following the same method described in Section II-C. We can estimate the footprint of a single 72-bit SR-Loop register file from each of the circuit components. Assuming we need approximately  $100 \mu\text{m}$  between each 8-bit cell for routing, the  $3 \times 3$  array has a  $600 \mu\text{m}$  by  $1,320 \mu\text{m}$  footprint. Additionally, the 2-to-4 decoder circuit is  $240 \mu\text{m}$  by  $200 \mu\text{m}$ , and the 1-to-3 demux is  $480 \mu\text{m}$  by  $340 \mu\text{m}$ . Therefore, the 72-bit register file should have an area of approximately  $1.2 \text{ mm}^2$  ( $600 \mu\text{m} \times 1,320 \mu\text{m} + 2 \cdot (480 \mu\text{m} \times 340 \mu\text{m}) + 2 \cdot (240 \mu\text{m} \times 200 \mu\text{m})$ ).

The design can be scaled to an  $N$ -row  $\times$   $M$ -column  $\times$   $K$ -bit register file with the following components: a 1-to- $N$  demux and 1-to- $M$  demux for S and R row and column write; an  $N \times M$  array



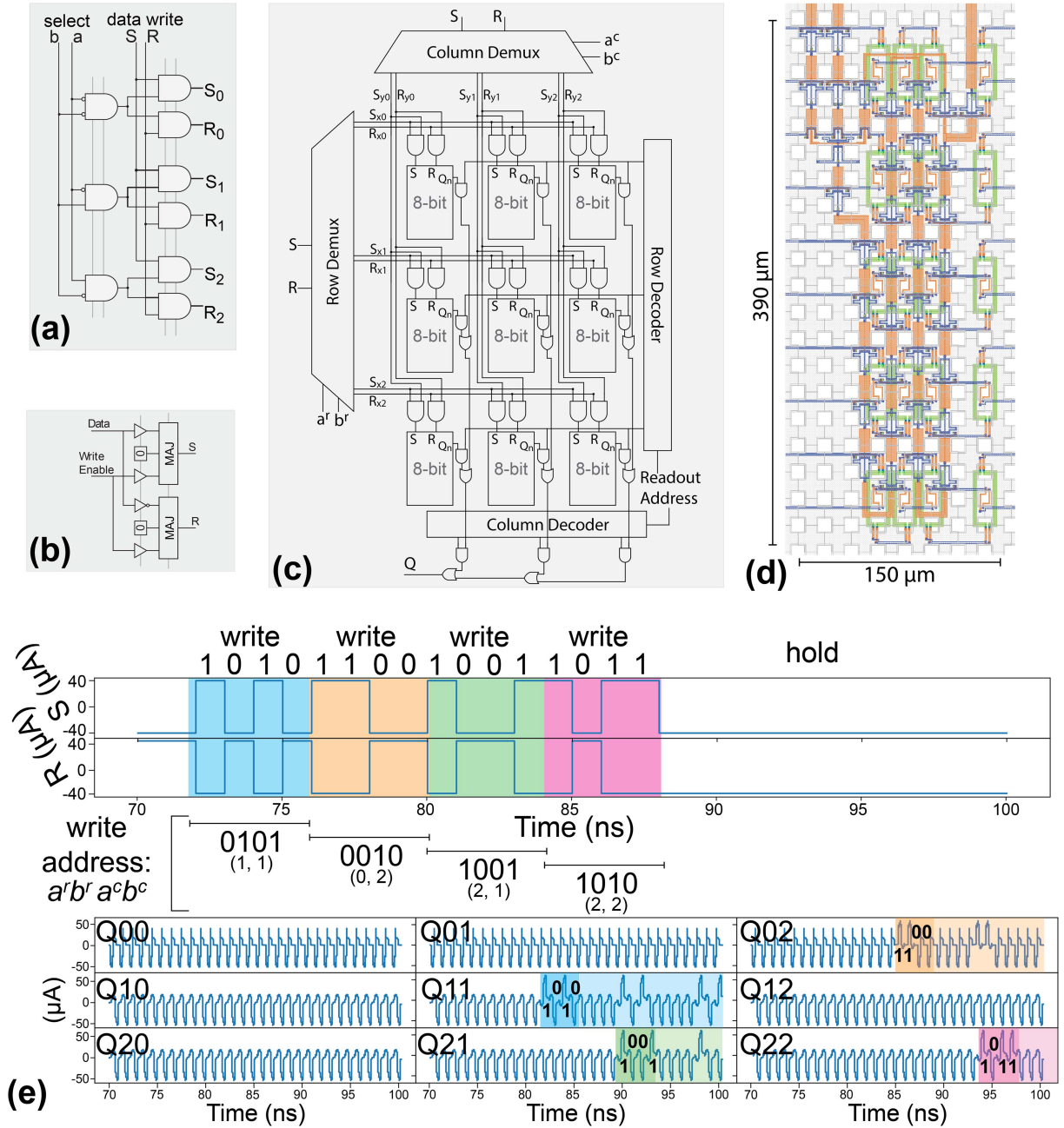


Fig. 3. (a) Schematic diagram for a 1-to-3 demux distributing S and R to the index specified by a and b select lines. The AND gates are composed of AQFP majority gates with a constant 0 cell and the gray vertical nets indicate a single phase for the excitation signal. The demux has a 1 clock cycle (4 phase) latency. (b) Simple circuit for converting from a Data-Enable memory encoding to a Set-Reset memory encoding. The Data-Enable addressing stores the value of the data input whenever the enable signal is high, and holds the existing data when enable is low. Set-Reset will store the value on the Set signal only when the Set and Reset values are not the same. (c) Schematic diagram of a  $3 \times 3 \times 8$ -bit SR-Loop register file. The design is composed of a column and row demux, a  $3 \times 3$  array of 8-bit SR-Loops, and a row and column decoder fed into a merger tree of OR gates for readout. This design requires 1,510 JJs. (d) Layout design for a single 8-bit SR-Loop memory cell. (e) Cadence SPICE simulation of SR-Loop register file. Each color along the S and R signals highlight the data signal sent to a unique SR-Loop within the array. For example, the first bit sequence in blue, 1010, is sent to the cell at index (1, 1) with the 0101 write select address. Then in the bottom plots, we see 1010 read from the correct cell with a 10 clock cycle write latency. The orange, green, and pink regions similarly show writing data to (0, 2), (2, 1), and (2, 2), respectively. Note that this plot demonstrates operation at 1 GHz for visual clarity, but the circuit simulation was also verified at 5 GHz.

of K-bit SR-loop cells; and a  $\log_2(N)$ -to-N decoder,  $\log_2(M)$ -to-M decoder, and merger tree for readout. The exact optimal sizing for N, M, and K depend on the memory array application. For example, if the application requires high throughput and low latency, it may be beneficial to keep K small so that stored data is accessible on shorter time intervals around each SR-Loop.

Whereas, a power or area limited application may benefit from larger K at the expense of read latency because more data can be stored without introducing more costly addressing overhead. Additionally, keeping N unequal to M can change the readout latency depending on the direction of the merger tree. Finally, all of these trade-offs do not have a linear relationship because the

demuxes and decoders size and cost is largely binned in steps of  $2^N$  or  $2^M$ , unlike size and cost for  $K$ , which scales continuously but in smaller increments. We'll note here that the  $3 \times 3$  array design is an example of inefficient utilization of the demuxes and decoders because 2 address bits could easily encode 4 indices at the cost of approximately 100 additional JJs. Future work will explore this optimization in more detail.

From a circuit design perspective, the SR-Loop register file is unique because it combines serial memory devices with addressable array style memory. It can be thought of as a three-dimensional memory array because each of the storage cells in the row-column array has an additional bit depth dimension. Existing AQFP register file designs have used 1-bit storage cells built from D-Latch majority gates with write and read decoders [1]. While these have been demonstrated with impressive size and useful application in an AQFP microprocessor [5], they can be greatly improved upon by switching from Data-Enable addressing (D-Latch [8] storage) to Set-Reset addressing (SR-Latch storage) because the D-Latch requires two stages of majority gates while the SR-Latch only needs one. Therefore, even if we implement our SR-Loop with 1-bit depth, the SR-Loop register will have a smaller footprint and better energy performance than D-Latch based registers. If data applications require Data-Enable address retrieval, we can use the simple circuit in Fig. 3(b) to convert between the two encodings once, in the peripheral addressing logic, rather than embedding the extra logic in each cell, as is done in the D-Latch design.

For direct comparison, a 32-word by 8-bit AQFP register composed of D-Latch 1-bit storage cells has a projected  $14 \text{ mm} \times 1 \text{ mm}$  footprint and 20 clock cycle read latency [1]. Projecting the equivalent SR-Loop register with  $4 \times 8 \times 8$ -bit configuration is roughly  $1.60 \text{ mm} \times 1.76 \text{ mm}$  (excluding addressing logic and maintaining  $100 \text{ }\mu\text{m}$  routing spacing) with 20 clock cycle read latency. For an extremely conservative estimate, we will assume that the demux and decoder addressing overhead will double both dimensions of the SR-Loop array, resulting in a  $3.20 \text{ mm} \times 3.52 \text{ mm}$  footprint. Before comparing the D-Latch and SR-Loop register file area, it's important to note that the D-Latch register was designed with a  $30 \text{ }\mu\text{m}$  by  $40 \text{ }\mu\text{m}$  AQFP standard cell footprint in the AIST STP2 process [13], while our SR-Loop AQFP standard cell is  $20 \text{ }\mu\text{m}$  by  $20 \text{ }\mu\text{m}$ . Therefore, our design benefits from a 66% area reduction purely from AQFP device size; however, the 32-word by 8-bit SR-Loop register design contributes an additional 26% reduction in area compared to the same bit storage D-Latch register. Additionally, the read latency is identical, so converting to the serial memory cell does not incur an unreasonable delay overhead.

#### IV. CONCLUSION

We designed and fabricated a 3-bit SR-Loop memory cell demonstrating proper write operation and storage states and extended this to an 8-bit register file design that is more area efficient than existing AQFP D-Latch registers while maintaining the comparable read latency. Future work will include testing our 8-bit SR-Loop cell which is currently in fabrication, demonstrating the entire SR-Loop register with addressing logic, and evaluating the design and application space for SR-Loop

registers. With a successful SR-Loop register file, the AQFP logic family gains a higher density low-level memory which can help advance the goal of scalable energy efficient processors.

#### ACKNOWLEDGMENT

The authors would like to thank Andrew Wagner and David Russo from MIT Lincoln Laboratory for their design of the AQFP cell library used in this work. Additionally, the authors thank Reed Foster, Donnie Keathley, and Matteo Castellani for helpful discussion and comments during the preparation of this manuscript.

*Distribution Statement A:* Approved for public release. Distribution is unlimited. This material is based upon work supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001, and the MIT Center for Bits and Atoms Consortium. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Under Secretary of Defense for Research and Engineering.

#### REFERENCES

- [1] N. Tsuji, C. L. Ayala, N. Takeuchi, T. Ortlepp, Y. Yamanashi, and N. Yoshikawa, "Design and implementation of a 16-word by 1-bit register file using adiabatic quantum flux parametron logic," *IEEE Trans. Appl. Supercond.*, vol. 27, no. 4, Jun. 2017, Art. no. 1300904.
- [2] N. Takeuchi, D. Ozawa, Y. Yamanashi, and N. Yoshikawa, "An adiabatic quantum flux parametron as an ultra-low-power logic device," *Supercond. Sci. Technol.*, vol. 26, no. 3, Jan. 2013, Art. no. 035010. [Online]. Available: <https://dx.doi.org/10.1088/0953-2048/26/3/035010>
- [3] N. Takeuchi, T. Yamae, C. L. Ayala, H. Suzuki, and N. Yoshikawa, "An adiabatic superconductor 8-bit adder with  $24k_B T$  energy dissipation per junction," *Appl. Phys. Lett.* vol. 114, no. 4, 2019, Art. no. 042602. [Online]. Available: <https://doi.org/10.1063/1.5080753>
- [4] N. Takeuchi, T. Yamae, C. L. Ayala, H. Suzuki, and N. Yoshikawa, "Adiabatic quantum-flux-parametron: A tutorial review," *IEICE Trans. Electron.*, vol. 105, no. 6, pp. 251–263, 2022.
- [5] C. L. Ayala, T. Tanaka, R. Saito, M. Nozoe, N. Takeuchi, and N. Yoshikawa, "MANA: A monolithic adiabatic integration architecture microprocessor using 1.4-zJ/op unshunted superconductor Josephson junction devices," *IEEE J. Solid-State Circuits*, vol. 56, no. 4, pp. 1152–1165, Apr. 2021.
- [6] R. Burnett et al., "Demonstration of superconducting memory for an RQL CPU," in *Proc. Int. Symp. Memory Syst.*, 2018, pp. 321–323. [Online]. Available: <https://doi.org/10.1145/3240302.3270313>
- [7] M. Tanaka, R. Sato, Y. Hatanaka, and A. Fujimaki, "High-density shift-register-based rapid single-flux-quantum memory system for bit-serial microprocessors," *IEEE Trans. Appl. Supercond.*, vol. 26, no. 5, Aug. 2016, Art. no. 1301005.
- [8] N. Tsuji, N. Takeuchi, Y. Yamanashi, T. Ortlepp, and N. Yoshikawa, "Majority gate-based feedback latches for adiabatic quantum flux parametron logic," *IEICE Trans. Electron.*, vol. E99.C, no. 6, pp. 710–716, 2016.
- [9] S. K. Tolpygo et al., "Advanced fabrication processes for superconducting very large-scale integrated circuits," *IEEE Trans. Appl. Supercond.*, vol. 26, no. 3, Apr. 2016, Art. no. 1100110.
- [10] D. Y. Zinoviev and Y. A. Polyakov, "Octopux: An advanced automated setup for testing superconductor circuits," *IEEE Trans. Appl. Supercond.*, vol. 7, no. 2, pp. 3240–3243, Jun. 1997.
- [11] A. F. Kirichenko et al., "Demonstration of an 88-bit RSFQ multi-port register file," in *Proc. 2013 IEEE 14th Int. Superconductive Electron. Conf.*, 2013, pp. 1–3.
- [12] J. Volk, A. Wynn, E. Golden, T. Sherwood, and G. Tzimpragos, "Addressable superconductor integrated circuit memory from delay lines," *Sci. Rep.* vol. 13, no. 1, 2023, Art. no. 16639. [Online]. Available: <https://doi.org/10.1038/s41598-023-43205-8>
- [13] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa, "Adiabatic quantum-flux-parametron cell library adopting minimalist design," *J. Appl. Phys.*, vol. 117, no. 17, 2015, Art. no. 173912. [Online]. Available: <https://doi.org/10.1063/1.4919838>