Experimental evaluation of simulated quantum annealing with MTJ-augmented p-bits

Andrea Grimaldi[‡,1,2], Kemal Selcuk[‡,1], Navid Anjum Aadit[‡,1], Keito Kobayashi[3,4], Qixuan Cao[1], Shuvro Chowdhury[1], Giovanni Finocchio[2], Shun Kanai[3,5], Hideo Ohno[3,5,6], Shunsuke Fukami[3,4,5,6] and Kerem Y. Camsari[†,1]

[1]Department of Electrical and Computer Engineering, University of California Santa Barbara, Santa Barbara, CA, USA

[2]Department of Mathematical and Computer Sciences, Physical Sciences and Earth Sciences, University of Messina, Messina, Italy,

[3]RIEC, Tohoku University, Sendai, Japan, [4]Graduate School of Engineering, Tohoku University, Sendai, Japan,

[5]CSIS, Tohoku University, Sendai, Japan, [6]WPI-AIMR, Tohoku University, Sendai, Japan

[‡]*Equally contributing authors*, [†]email: camsari@ece.ucsb.edu

***Abstract*—The slowing down of Moore's Law has created an exciting new era of electronics, leading to the emergence of various types of CMOS+X devices and architectures. Here, we present the first experimental demonstration of a probabilistic computer where a stochastic magnetic tunnel junction (sMTJ) drives a powerful CMOS-based field programmable gate array (FPGA) in a heterogeneous compute fabric. We use our machine to experimentally evaluate the simulated quantum annealing (SQA) algorithm, known to closely mimic the behavior of D-Wave's quantum annealers which implement the transverse field Ising model (TFIM). Our machine matches the exact solution of the TFIM where p-bits in the FPGA are asynchronously driven by the stochastic dynamics of a magnetic tunnel junction. To compare the performance of SQA against classical annealing (CA) in hard combinatorial optimization at large scale, we also design a fully digital emulator of our asynchronous architecture in the FPGA. Our digital system uses 7,085 p-bits to factor up to 26-bit integers and is about 10X faster than optimized Tensor (TPU) and Graphics Processing Units (GPU) at lower power. Surprisingly, we find that the additional replica networks necessary for SQA do not lead to appreciably better performance over an optimized CA that is using the same computational resources. The systematic evaluation of the SQA algorithm we present will be relevant for other types of accelerators, such as photonic or electronic Ising machines and the integrated scaling of our CMOS + sMTJ architecture could lead to orders of magnitude further improvements over TPU and GPUs, according to experimentally-validated projections.**

I. INTRODUCTION

The slowing down of the Moore's Law has been marked by the rise of domain-specific hardware and architectures. A notable example of this approach is probabilistic computation with p-bits with a wide applications space from accelerating optimization, sampling and Monte Carlo algorithms. p-bits have previously been demonstrated in small-scale prototypes using magnetic nanodevices [1] or in large-scale using all digital CMOS [2]. In this work, we demonstrate a novel heterogeneous architecture combining stochastic magnetic tunnel junctions (sMTJ) asynchronously driving digital p-bits implemented in a powerful CMOS-based field programmable gate array (FPGA) (FIG. 1). Here, the sMTJ is used as an asynchronous and randomized clock to drive digital p-bits for optimization and sampling problems however the same heterogeneous integration architecture could be used for other purposes, for example, to turn low-quality digital random numbers to high-quality true random numbers at massive scale. We start with a perpendicular MTJ whose free layer is designed to have a low energy barrier ($\approx$ 15-20 k_BT) such that the resistance of the MTJ shows telegraphic fluctuations (FIG. 2). Due to the dipolar coupling between the fixed layer and the free layer the MTJ does not show fluctuations at zero current. Typically around $\approx$ 5-15 μA the spin torque effect cancels the dipolar coupling and the sMTJ shows 50/50 fluctuations (FIG. 2a). We then design a circuit where the sMTJ is attached to an NMOS transistor and a source resistor where the NMOS controls the current through the sMTJ by an input voltage (FIG. 2b). The fluctuations at the drain are converted to rail-to-rail voltages using two comparators. Typically one comparator (or inverter) is enough [1], here we use a double comparator setup to drive the peripheral module (PMOD) pins of a Kintex FPGA board. FIG. 2c shows the stochastic fluctuations of the overall p-bit circuit as a function of the input voltage of the NMOS.

II. HETEROGENEOUS ARCHITECTURE

For our heterogeneous architecture, we bias the sMTJ-based p-bit at its midpoint such that the output is a 50/50 fluctuating, rail-to-rail signal used as a stochastic clock to drive digital p-bits in the FPGA. Inside the FPGA, we construct a programmable architecture representing an Ising model:

$$E = -\left(\sum J_{ij} m_i m_j + \sum h_i m_i\right) \tag{1}$$

where J_{ij} are the weights and h_i are the biases whereas the m_i are the p-bit states that are $+1$ or -1, which are converted to binary states, s, to be represented in the FPGA using $m = 2s-1$ [2]. Our main benchmark is a 1D transverse field Ising model problem (TFIM) which translates to a 2D nearest-neighbor classical Ising graph by a Suzuki-Trotter decomposition. The 2D grid has a chessboard pattern and can be colored using 2-colors. This allows p-bits in each color block to be updated in parallel, a trick often used in parallelizing the 2D classical Ising model in GPU and TPUs. We use similar architecture in the FPGA where a graph representing the J_{ij} is colored and each color is updated by a separate clock [2]. We use a single magnetic p-bit where the rising edge triggers an update of one color block and the falling edge triggers the other block (FIG. 2d). These triggers activate low quality pseudorandom number generators, labeled PRNG in the FPGA block view (FIG. 1). The equations of p-computing for optimization (i.e., minimizing E) and sampling (i.e., from $\propto \exp[-\beta E]$) are:

$$m_i = \mathrm{sgn}(\tanh(\beta I_i) - r_U) \quad I_i = \sum J_{ij} m_j + h_i \tag{2}$$

where m_i are the p-bit states (± 1), r_U is a uniform random number between (-1,+1) and $[J], \{h\}$ define the Ising model

of Eq. 1 and β is the inverse pseudo-temperature. Typically a single network describing weights and biases is needed for either optimization or sampling. To mimic quantum annealing through simulated quantum annealing (SQA), however, interacting replicas of the original network is necessary (FIG. 3b). The need for replicas makes SQA computationally more expensive for classical hardware, the graph size for an N-node problem becomes NR, R being the number of replicas. With the same hardware effort, however, classical annealing (CA) algorithm can be run in R parallel replicas [3] (FIG. 3a). We first demonstrate how our sMTJ + FPGA p-computer reproduces the exact quantum average obtained from a 1D-TFIM by its 2D classical Ising model mapping. The TFIM model serves the basis of SQA which is later compared with CA using the computationally hard factorization problem (FIG. 4a,b).

III. QUANTUM PROBLEMS WITH MTJ-BASED P-BITS

FIG. 5 shows an experimental measurement of a sampling problem for the 1D nearest neighbor TFIM of an 8-qubit system. The mapping between the TFIM and classical network is done through: $J_{ij} = J_{ij}^{Q}/R, h_i = h_i^{Q}/R, J_T = 1/(2\beta)\ln[\tanh(\beta\Gamma_x/R)]$, where the ($J$, h) and J_T represent local and transverse terms, respectively. Each are obtained from the quantum Hamiltonian, denoted by the superscript Q (FIG. 3b). We choose $R = 10$ to simulate this system using 80 p-bits divided into two color blocks which are driven by the external sMTJ. FIG. 5a shows two experiments with a longitudinal magnetic field, $\Gamma_z = \pm 1$. We take 100 independent measurements for each Γ_z value where the average magnetization, $\langle m_z \rangle$ is initialized to 0 over 100 runs. For each magnetic field, the probability distribution defined by 100 separate runs evolve to the *exact* quantum average (FIG. 5b) computed from the density matrix, $\rho = 1/Z\ \mathrm{tr.}[\exp(-\beta H_Q)]$ (FIG. 4a). A final histogram obtained after the system has saturated shows excellent agreement with the quantum distribution (FIG. 5c). The relaxation timescales are in hundreds of seconds in these measurements however this is due to the slow sMTJs we used, $>$ GHz frequencies of sMTJs have been demonstrated [4]. This result establishes how our CMOS + sMTJ architecture can solve a truly quantum problem and if scaled in integrated circuits could lead to the faithful simulation of hundreds of thousands of qubits.

IV. HARD OPTIMIZATION: CA OR SQA?

Given the additional costs of building replicas, we systematically investigate the performance of SQA and compare it with CA. To do this, we construct a fully digital emulator of our heterogeneous computer in the FPGA, similar to our earlier result [2], replacing the sMTJ driven clocks by multiple colored digital ones, allowing us to reach up to 7,085 p-bit circuits which can solve up to 26-bit integer factorization problems, far greater than alternative approaches. First, we perform a careful parameter optimization of SQA, finding the best possible combination of β/R at several bit-lengths (with 100 semiprimes/bit). We find that around $\beta/R \approx 2.5$ the time to solution is minimal (FIG. 6a). In the remaining comparisons, we use the same $\beta/R = 2.5$ for all examples. FIG. 6b shows the annealing schedules we used for CA and QA and FIG. 6c-f represent our main algorithmic findings: we show that in fast or slow annealing for solving the integer factorization problem, the time to solution for SQA is much better than a single CA network, in line with earlier observations [3]. When we perform a *replicated* CA (RCA) algorithm where the best among all parallel runs of the simulated annealing is chosen, the advantage of SQA vanishes, and in some cases, becomes inferior to RCA. Given the stringent requirements of finding an optimum (β, R) and the necessity of additional transverse weights, our conclusion is the added difficulty of SQA may not justify its use over a much simpler RCA algorithm using parallel replicas, particularly for classical domain-specific hardware such as Ising machines. Simulating quantum systems may still be a useful application of SQA.

V. PROJECTIONS AND OUTLOOK

In FIG. 7 and Table I, we show a summary of our results and provide experimentally-validated projections. Compared to optimized GPU/TPU implementations [5]–[10], the fully digital FPGA system is already competitive in the main metric of probabilistic flips/second. Given recent experimental breakthroughs in fast sMTJs [4] and the demonstrated integration of millions of sMTJs in embedded CMOS raises the intriguing possibility of orders of magnitude improvement in sampling throughput and energy-efficiency for probabilistic computing applications.

K.Y.C., K.S., N.A.A. acknowledge support from National Science Foundation (CCF 2106260), Samsung GRO and the ONR YIP program. A.G. and G.F. were supported under PRIN 2020LWPKH7 funded by the Italian M.U.R and supported by the PETASPIN Association (www.petaspin.com). S.F. and S.K. are supported by JST-CREST JPMJCR19K3 and JST PRESTO JPMJPR21B2, respectively.

[1] B. et al. William A, "Integer factorization using stochastic magnetic tunnel junctions," *Nature*, vol. 573, no. 7774, pp. 390–393, 2019.

[2] N. A. Aadit et al., "Massively parallel probabilistic computing with sparse ising machines," *Nature Electronics*, pp. 1–9, 2022.

[3] B. Heim et al., "Quantum versus classical annealing of ising spin glasses," *Science*, vol. 348, no. 6231, pp. 215–217, 2015.

[4] K. Hayakawa et al., "Nanosecond random telegraph noise in in-plane magnetic tunnel junctions," *Physical review letters*, vol. 126, no. 11, p. 117202, 2021.

[5] B. Block et al., "Multi-gpu accelerated multi-spin monte carlo simulations of the 2d ising model," *Computer Physics Communications*, vol. 181, no. 9, pp. 1549–1556, 2010.

[6] T. Preis et al., "Gpu accelerated monte carlo simulation of the 2d and 3d ising model," *Journal of Computational Physics*, vol. 228, no. 12, pp. 4468–4477, 2009.

[7] K. Yang et al., "High performance monte carlo simulation of ising model on tpu clusters," in *Proceedings of the International Conference for HPC, Networking, Storage and Analysis*, 2019, pp. 1–15.

[8] J. Romero et al., "High performance implementations of the 2d ising model on gpus," *Computer Physics Communications*, vol. 256, p. 107473, 2020.

[9] Y. Fang et al., "Parallel tempering simulation of the three-dimensional edwards–anderson model with compact asynchronous multispin coding on gpu," *Computer Physics Communications*, vol. 185, no. 10, pp. 2467–2478, 2014.

[10] M. Aramon et al., "Physics-inspired optimization for quadratic unconstrained problems using a digital annealer," *Frontiers in Physics*, vol. 7, p. 48, 2019.

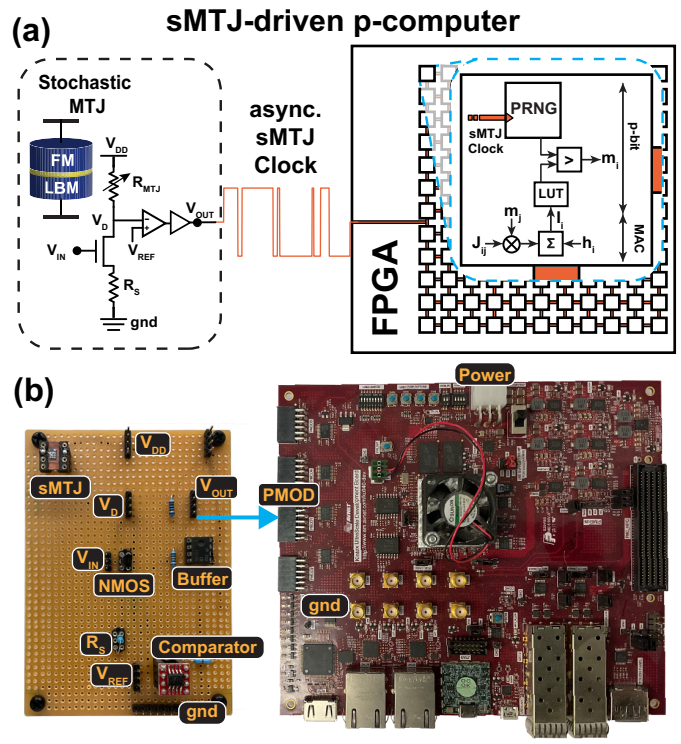


Fig. 1. (a) The sMTJ-based p-bit feeds a stochastic and asynchronous clock to the FPGA. Digital p-bit architecture including a lookup table, comparator, weights and a pseudorandom number generator (PRNG) activated by the sMTJ. (b) Photo of the experimental setup. (Left) Vector board of the mixed-signal p-bit circuit combining sMTJs with NMOS transistors and comparators. (Right) Kintex FPGA board receives the sMTJ clock through the PMOD pin to solve optimization and sampling problems by probabilistic computation.

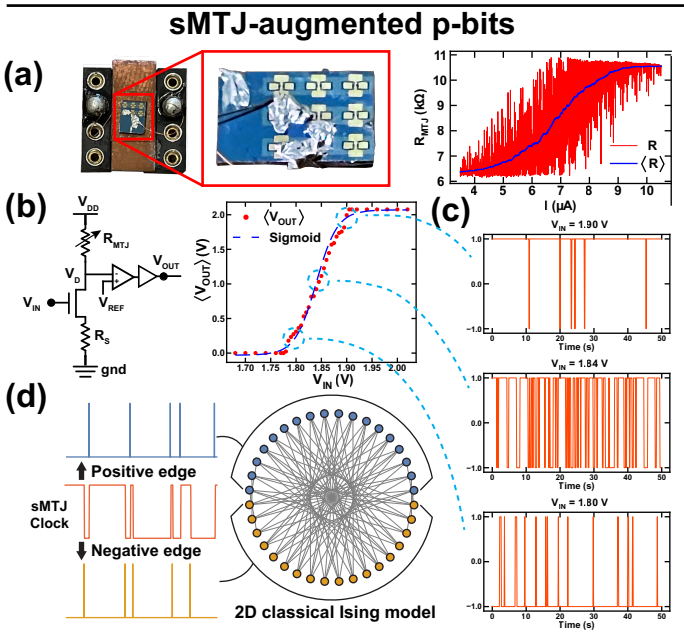
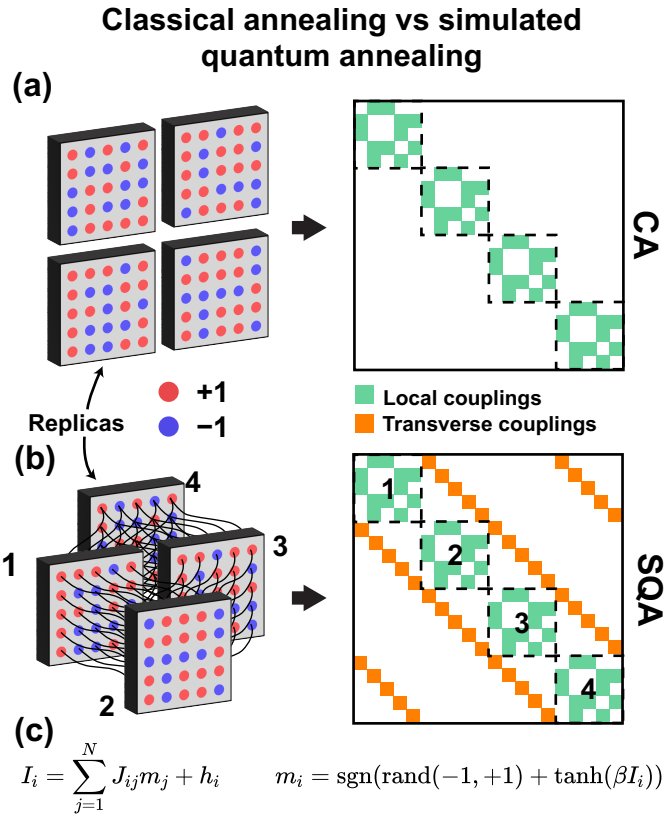
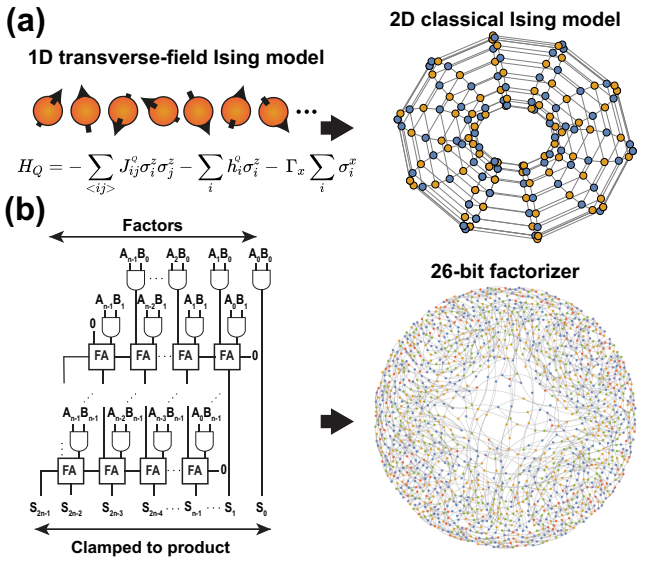


Fig. 2. (a) (Left) Wirebonded sMTJs. (Right) Resistance of sMTJ as a function of current. (b) (Left) sMTJ-based p-bit design with NMOS (2N7000), source resistance (R_S = 10 kΩ) and two comparators. V_{DD} of the sMTJ branch is 500 mV. (Right) Time-averaged V_{OUT} over 200 s as V_{IN} is varied. The first op-amp (AD8692) uses a reference voltage of V_{REF} (= 0.39 V), the second comparator acts as buffer to drive the PMOD pin of FPGA. (c) Normalized V_{OUT} vs time at different V_{IN}. (d) sMTJ-augmented p-bits serve as a clock. Positive edges and negative edges update two separate blocks of p-bits without overlapping with each other. Our sMTJ-based p-bit is biased to produce 50/50 fluctuations.



$$I_i = \sum_{j=1}^{N} J_{ij} m_j + h_i \qquad m_i = \mathrm{sgn}(\mathrm{rand}(-1,+1) + \tanh(\beta I_i))$$

Fig. 3. (a) Replicated classical annealing. Several replicas of the system are working independently and [J] takes the form of a block matrix. All couplings are local. (b) Simulated quantum annealing (SQA). All p-bits are connected to their counterparts in the neighboring replicas with the J_T (transverse) coupling. Periodic boundary conditions are used for transverse couplings. (c) Basic equations for p-bits. (Left) Synaptic equation to calculate the input signal received by a p-bit. (Right) Stochastic activation of a p-bit. β is the inverse pseudo-temperature.



$$H_Q = -\sum_{<ij>} J_{ij}^{0} \sigma_i^z \sigma_j^z - \sum_i h_i^0 \sigma_i^z - \Gamma_x \sum_i \sigma_i^x$$

Fig. 4. (a) (Left) Transverse-field Ising model for a 1D quantum spin chain, along with the corresponding quantum Hamiltonian. (Right) The 2D graph (torus) representing the corresponding classical system using p-bits. (b) Integer factorization problem. (Left) The invertible multiplier circuit for factorization using p-bits. The multiplier circuit is run in reverse by clamping the product to the number to factor and observing the factors. On the right, the graph of a 26-bit factorizer.

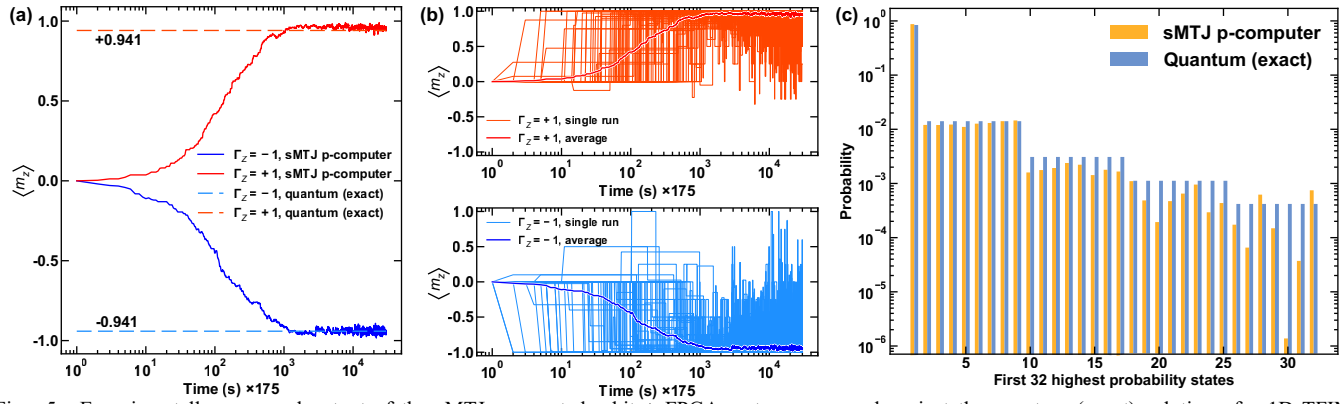


Fig. 5. Experimentally measured output of the sMTJ-augmented p-bit + FPGA system compared against the quantum (exact) solution of a 1D TFIM Hamiltonian: (a) A transverse field ($\Gamma_X = +1$) is applied to a linear chain of eight (8) FM coupled ($J_{ij} = 2$) qubits with periodic boundary operating at $\beta = 0.5$. The qubits are initialized to the state ($|\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow\rangle$) where $\langle m_z \rangle$ is zero. A symmetry breaking field along z-direction (either $\Gamma_Z = +1$ or $\Gamma_Z = -1$) is applied such that the exact average is (± 0.941), respectively. Average output (over 100 different runs) with R = 10 replicas gradually reaches the exact equilibrium value for both Γ_Z. (b) Outputs of each individual run. (c) Measured equilibrium probabilities for the first 32 states (highest probability) show excellent agreement with the theoretical equilibrium probabilities as calculated from the corresponding quantum density matrix.

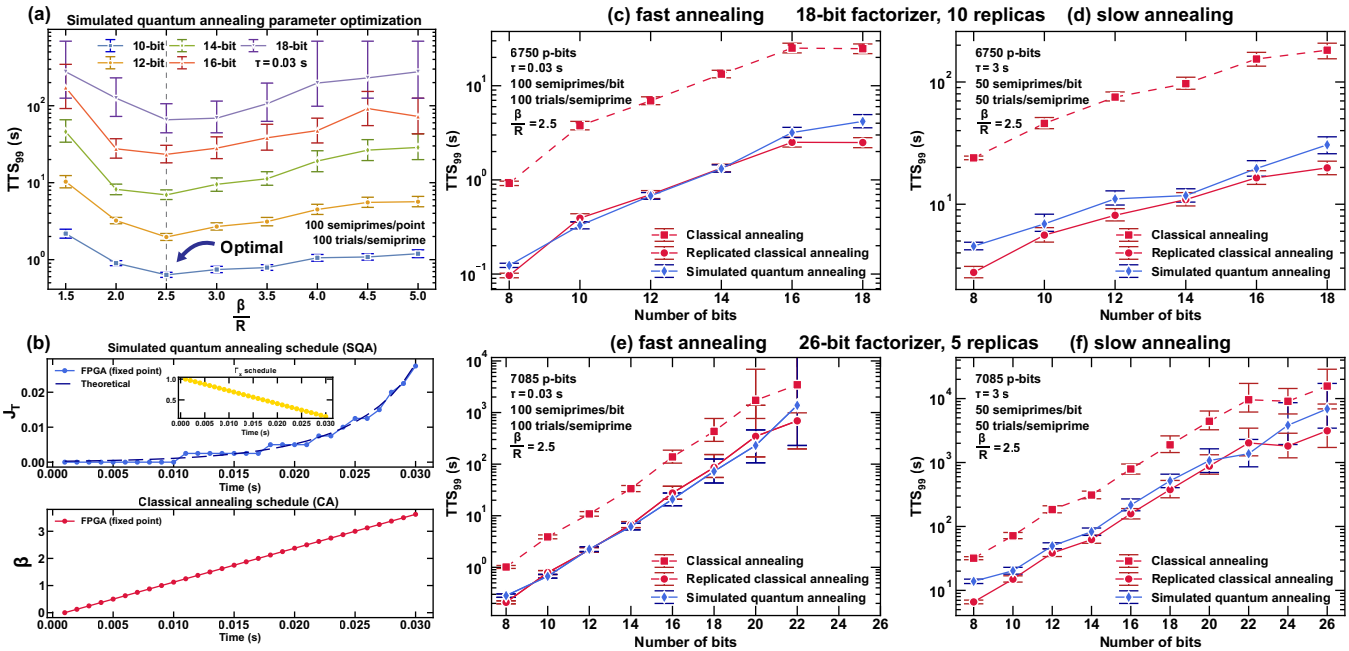


Fig. 6. Fully digital FPGA evaluation of SQA vs RCA. An 18-bit factorizer with 10 replicas and a 26-bit factorizer with 5 replicas are encoded with 6,750 and 7,085 p-bits, respectively. Both circuits are synthesized only once and reconfigured to factor semiprime numbers from 8-bit to the highest bit. (a) For SQA, $\beta/R = 2.5$ shows the optimum TTS_{99} (time to find the exact solution with 99% probability for a given annealing schedule). $\beta/R = 2.5$ is used for all experiments. (b) (Upper panel) The annealing schedule for the SQA for J_T where the transverse field, Γ_X is linearly decreased at fixed β (inset). FPGA approximates J_T with s[6][3] fixed point precision. (Lower panel) The annealing schedule for RCA with β linearly increased from 0.500 to 3.625. (c,e) Fast annealing schedules ($\tau = 0.03$ s) for RCA and SQA on the 18-bit and the 26-bit factorizer circuits to factor 100 random semiprimes with 100 trials for each number of bits. (d,f) Slow annealing schedules ($\tau = 3$ s) for RCA and SQA on the 18 and 26-bit factorizers (50 semiprimes, 50 trials). For all results (c,d,e,f) RCA and SQA perform similarly, both significantly better than the single replica CA. Error bars are obtained using bootstrapping with 95% confidence.

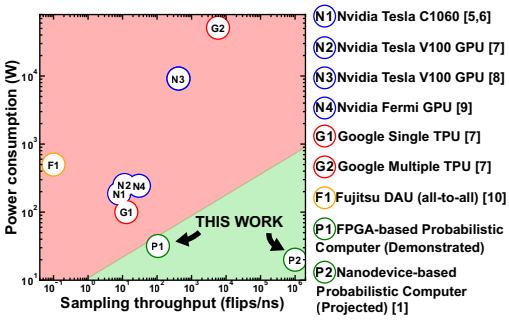


Fig. 7. Comparison of probabilistic accelerators, showing sampling throughput (flips/ns) and power consumption. The performance achieved in this work with digital FPGA (P1) and the projection with sMTJs (P2) fit into the desirable zone, yielding maximum throughput with the least power consumption.

Technology (This Work)	Sampling Throughput (flips/ns)	Power Dissipation (W)
sMTJ Branch + Comparator Branch + FPGA (Demonstrated)	4e-8	6.7 μW + 10.2 mW + 2.1 W
All Digital FPGA (Demonstrated)	106.28	32.72 W
sMTJ + CMOS (Projected)	1,000,000	20 W

Table I. Summary of results and projections. The prototype sMTJ-augmented FPGA has high energy efficiency: The sMTJ branch draws ~6.7 μW power, in good agreement with theoretical predictions [1]. It also has low throughput (~ 4e-8 flips/ns) since slow sMTJs were deliberately used to ease design, compared to the all-digital FPGA reaching ~100 flips/ns. Nanoseconds fluctuations have been demonstrated using sMTJs [4]. Up to 1M sMTJs fluctuating with 1 ns time scale can lead to 1M flips/ns with about 20 W power consumption [1], orders of magnitude better than optimized GPU/TPU (Fig. 7).