LETTER

https://doi.org/10.1038/s41586-019-1557-9

Integer factorization using stochastic magnetic tunnel junctions

William A. Borders[1,8], Ahmed Z. Pervaiz[2,8], Shunsuke Fukami[1,3,4,5,6,7]*, Kerem Y. Camsari[2]*, Hideo Ohno[1,3,4,5,6,7] & Supriyo Datta[2]

Conventional computers operate deterministically using strings of zeros and ones called bits to represent information in binary code. Despite the evolution of conventional computers into sophisticated machines, there are many classes of problems that they cannot efficiently address, including inference, invertible logic, sampling and optimization, leading to considerable interest in alternative computing schemes. Quantum computing, which uses qubits to represent a superposition of 0 and 1, is expected to perform these tasks efficiently[1–3]. However, decoherence and the current requirement for cryogenic operation[4], as well as the limited many-body interactions that can be implemented, pose considerable challenges. Probabilistic computing[1,5–7] is another unconventional computation scheme that shares similar concepts with quantum computing but is not limited by the above challenges. The key role is played by a probabilistic bit (a p-bit)—a robust, classical entity fluctuating in time between 0 and 1, which interacts with other p-bits in the same system using principles inspired by neural networks[8]. Here we present a proof-of-concept experiment for probabilistic computing using spintronics technology, and demonstrate integer factorization, an illustrative example of the optimization class of problems addressed by adiabatic[9] and gated[2] quantum computing. Nanoscale magnetic tunnel junctions showing stochastic behaviour are developed by modifying market-ready magnetoresistive random-access memory technology[10,11] and are used to implement three-terminal p-bits that operate at room temperature. The p-bits are electrically connected to form a functional asynchronous network, to which a modified adiabatic quantum computing algorithm that implements three- and four-body interactions is applied. Factorization of integers up to 945 is demonstrated with this rudimentary asynchronous probabilistic computer using eight correlated p-bits, and the results show good agreement with theoretical predictions, thus providing a potentially scalable hardware approach to the difficult problems of optimization and sampling.

The field of adiabatic quantum computing[9] (AQC) solves complex optimization problems by constructing networks of qubits in which the inter-qubit interactions are engineered to make the overall energy E reflect the cost function for the problem. One such algorithm[12] frames integer factorization of a given number F as an optimization problem by writing each of its factors X and Y in binary form and defining the cost function $E = (XY - F)^2$

$$E(x_P, \ldots, x_1; y_Q, \ldots, y_1) = \left[\left(\sum_{p=0}^{P} 2^p x_p\right)\left(\sum_{q=0}^{Q} 2^q y_q\right) - F\right]^2 \quad (1)$$

with $x_0 = 1$, $y_0 = 1$ and P, Q denoting the number of bits needed to represent X and Y, respectively, so that the lowest energy state corresponds to the configuration of qubits $\{x_p, \ldots, x_1, y_q, \ldots, y_1\}$ that makes XY equal to F. In general, E involves terms of the form $x_p y_q x_r y_s$, requiring up to four-body interactions. This algorithm does not require coherence, but needs auxiliary bits to represent many-body interactions when implemented using AQC[13,14]. In probabilistic computing, many-body interactions are implemented electrically, removing the need for extra components.

Individual p-bits are stochastic building blocks with a normalized output m_i that takes on the values 0 and 1 with probabilities P_0 and P_1, respectively. These probabilities are controlled by their normalized inputs I_i; for $I_i = 0$ they are equal ($P_0 = P_1 = 0.5$), large $+I_i$ pins the output m_i to 1 ($P_0 = 0$, $P_1 = 1$) and large $-I_i$ pins m_i to 0 ($P_0 = 1$, $P_1 = 0$). This is similar to the behaviour of a binary stochastic neuron, a well known concept in the field of stochastic neural networks and machine learning[15], which has an input–output relation $m_i = \vartheta[\sigma(I_i) - r]$, where ϑ is the unit step function, σ is the sigmoidal function, r is a random number uniformly distributed between 0 and 1, and the input I_i is obtained from the synaptic function (described below). Thus, the p-bit requires a natural element that is substantially unstable but controllable. A magnetic tunnel junction (MTJ), widely recognized as a critical building block of nonvolatile magnetoresistive random-access memory (MRAM)[10,11], has potential to be used as the stochastic element in p-bits[16] if its thermal stability can be sufficiently reduced. In this work, we build stochastic MTJs and demonstrate an experimental proof of concept of probabilistic computing, in which an eight-p-bit network performs integer factorization of values up to 945.

The building block of the p-bit, the MTJ, comprises ferromagnetic free and reference layers separated by an insulating tunnel barrier (Fig. 1a). Previous studies have used the switching probability[17] and fluctuation rate[18] of the free-layer magnetization of separate MTJs to show random-number generation and population coding, respectively. Here we show that complex optimization problems can be generally addressed using the correlation among multiple naturally stochastic MTJs. The stack consists of a CoFeB/MgO structure with a perpendicular magnetic easy axis[10], a de facto system of MRAM technology (see Methods section 'MTJ fabrication'). In general, an MTJ is characterized by its tunnelling magnetoresistance, which switches between high and low values by varying the angle between the magnetization direction of the two ferromagnetic layers[19]. The high (antiparallel, AP) and low (parallel, P) resistance states (R_{AP}, R_P) are separated by an energy barrier E such that stored information is retained for a time $\tau = \tau_0 \exp[E/(k_B T)]$ following Arrhenius' law, where τ_0 is the attempt time ($\tau_0 \approx 1$ ns)[20], k_B is the Boltzmann constant and T is the temperature (Fig. 1b). Nonvolatile memory applications require stable MTJs with a retention time τ of the order of years[11], whereas our p-bit experiments require stochastic MTJs with retention times on the millisecond scale. Figure 1c shows the measured τ as a function of the CoFeB free-layer thickness for different nominal diameters of the MTJ pillar. For each junction diameter D (CoFeB thickness t_{CoFeB}), the timescale of stochasticity decreases with increasing t_{CoFeB} (decreasing D).

[1]Laboratory for Nanoelectronics and Spintronics, Research Institute of Electrical Communication, Tohoku University, Sendai, Japan. [2]School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. [3]Center for Spintronics Integrated Systems, Tohoku University, Sendai, Japan. [4]Center for Innovative Integrated Electronic Systems, Tohoku University, Sendai, Japan. [5]Center for Spintronics Research Network, Tohoku University, Sendai, Japan. [6]Center for Science and Innovation in Spintronics (Core Research Cluster), Tohoku University, Sendai, Japan. [7]WPI-Advanced Institute for Materials Research, Tohoku University, Sendai, Japan. [8]These authors contributed equally: William A. Borders, Ahmed Z. Pervaiz. *e-mail: s-fukami@riec.tohoku.ac.jp; kcamsari@purdue.edu

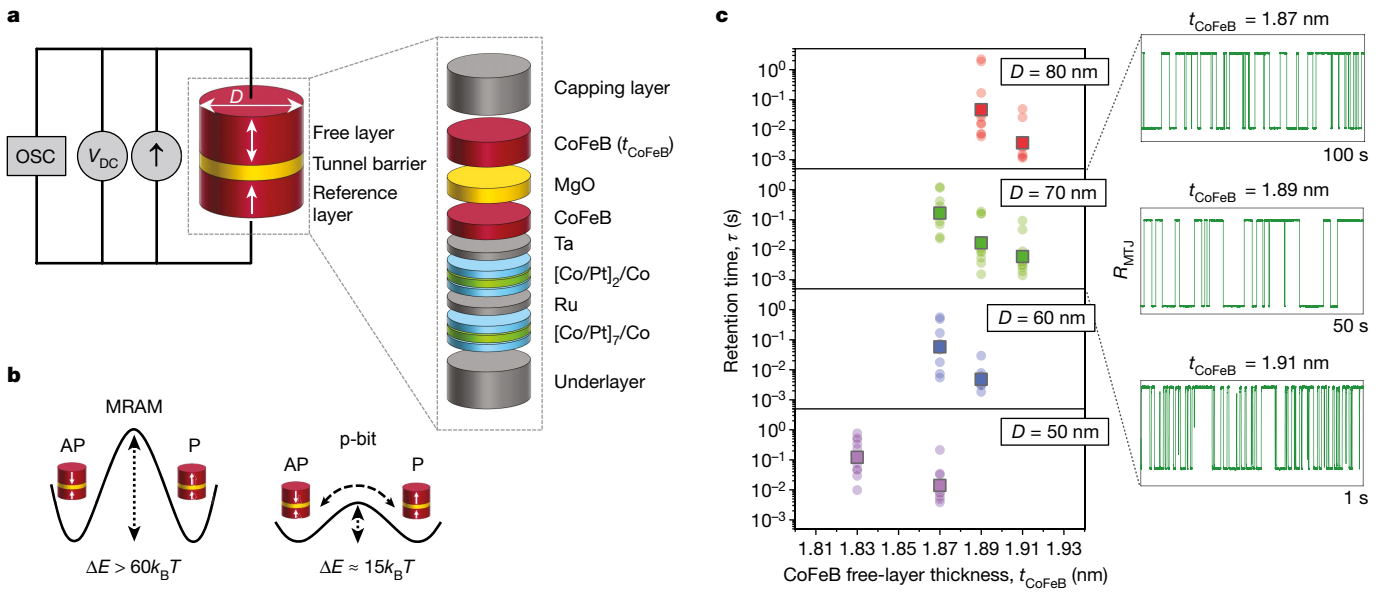


Fig. 1 | Characteristics of stochastic magnetic tunnel junctions. **a**, Measurement setup of a stochastic MTJ, with a stack structure that is only slightly modified from current MRAM technology. A current is passed from the free layer to the reference layer, a time-averaged signal is read on the voltmeter, and a time-domain signal is measured on the oscilloscope. **b**, The energy profile between the P and AP states of the magnetization orientation of the MTJ for typical MRAM technology and for the MTJs used in the p-bits for this work. **c**, Experimental results showing the retention time τ of MTJs with varying thickness of the CoFeB free layer t_{CoFeB} and diameter D. The retention time τ is determined at an applied current of $I_{50/50}$, which induces equal fluctuation time of the MTJ magnetization in the AP and P states. Square symbols represent the average of the retention time for 10 MTJs at each D and t_{CoFeB}. Transparent circles represent the retention time for each device. The right-most panels show the effect of varying the free-layer thickness on the stochasticity for devices of the same size. Note that reducing the thickness below 1.8 nm results in a stable binary device suitable for nonvolatile memory applications[30].

The behaviour is understood by considering the energy barrier for magnetization reversal. Because interfacial magnetic anisotropy is dominant in this system[10], increasing the free-layer thickness will reduce the total perpendicular magnetic anisotropy energy, mainly owing to an increase in the demagnetizing energy, which favours in-plane magnetization. Furthermore, decreasing D also decreases the energy barrier for magnetization reversal, as reported in previous studies[21]. Importantly, by varying only the ferromagnetic free-layer thickness for arbitrary sizes of the MTJs used in typical MRAM fabrication, we are able to manipulate the stochasticity of the MTJ so that it is suitable for p-bit experiments (see Methods section 'MTJ characterization').

To form the building block for stochastic neural networks, we connect the stochastic MTJs with standard n-type metal–oxide–semiconductor (NMOS) transistors to obtain a three-terminal p-bit (Fig. 2a). The output voltage for the ith p-bit, $V_{OUT,i}$, from this composite unit can be written in terms of the input voltage $V_{IN,i}$ in a form similar to the ideal binary stochastic neuron described above:

$$\overbrace{\frac{V_{OUT,i}}{V_{DD}}}^{m_i} \approx \vartheta\left(\sigma\left\{\overbrace{\frac{V_{IN,i}-v_{0,i}}{V_{0,i}}}^{I_i}\right\}-r\right) \quad (2)$$

where V_{DD} is the supply voltage, $V_{0,i}$ is the scaling voltage determined by the transistor, $v_{0,i}$ is the offset voltage (1.95 V in this experiment). Figure 2b shows the time-averaged output voltage as the input voltage is swept from 1.5 V to 2.4 V, where each point is averaged over 700 ms with a fixed input voltage. Figure 2c shows the time-varying output voltage for specific input voltages, displaying stochastic behaviour centred at 1.95 V, but becoming deterministic as the input changes by about ±75 mV, a consequence of spin-transfer torque[22–24] (see Methods section 'p-bit construction').

These p-bits can be used to perform useful functions by interconnecting them so that the ith p-bit is driven by a synaptic input I_i that is a function of all the other outputs $\{m_1, \ldots, m_N\}$. Boltzmann machines represent a subset of such networks for which I_i can be obtained from an energy function E using the relation $I_i = -\partial E(m_1, \ldots, m_N)/\partial m_i$. Such networks will visit different configurations with probabilities given by the Boltzmann law $P(m_1, \ldots, m_N)$, which are proportional to $\exp[-E(m_1, \ldots, m_N)]$, so configurations with the lowest energy E occur with the highest probability. This property makes the networks naturally suited for solving optimization problems, similar to the way that AQC solves them, where the correct solution minimizes a cost function identified for E and is used to calculate the synaptic inputs I_i. Unlike in machine-learning schemes, these synaptic inputs are analytically deduced and not learned.

Experimentally we connect eight p-bits following a general architecture presented previously[25] (Fig. 3a). A microcontroller reads the output voltage of each p-bit and is programmed to calculate the inputs I_i for a given cost function E. The result is converted into analogue voltages using a digital-to-analogue converter (DAC). Together, the microcontroller and DAC function as the synaptic weight logic that determines I_i, reading in digital outputs from the p-bits and feeding back analogue inputs (see Methods section 'p-circuit construction'). Although the main experiment that we describe here demonstrates integer factorization, this methodology can be applied to other optimization problems, such as invertible Boolean logic, for which the objective is to determine all the possible inputs when the logic output is known (see Methods section 'p-bit-based implementation of an invertible AND gate').

In the case of integer factorization, we use the cost function represented by equation (1) to evaluate the input functions. We first test the factorization of 35 using four p-bits ($P = 2$, $Q = 2$) (see Methods section 'Factorization algorithm'). In our algorithm, the synaptic inputs include nonlinear terms that effectively enforce both three p-bit and four p-bit interactions, in addition to the customary linear terms arising from two p-bit interactions. Accordingly, an integer up to 2^{n+2} can be encoded according to equation (1) with n p-bits using the current algorithm, a relation that requires fewer bits than current AQC schemes, mainly owing to the added flexibility provided by nonlinear synapses[14] that could be useful in other optimization problems as well. Figure 3b gives the three-dimensional histograms of the time fluctuations (see Methods section 'Factorization algorithm') for pairs of numbers $\{x_2, x_1, 1\}$ and $\{y_2, y_1, 1\}$, depicted below the uncorrelated state that is obtained when

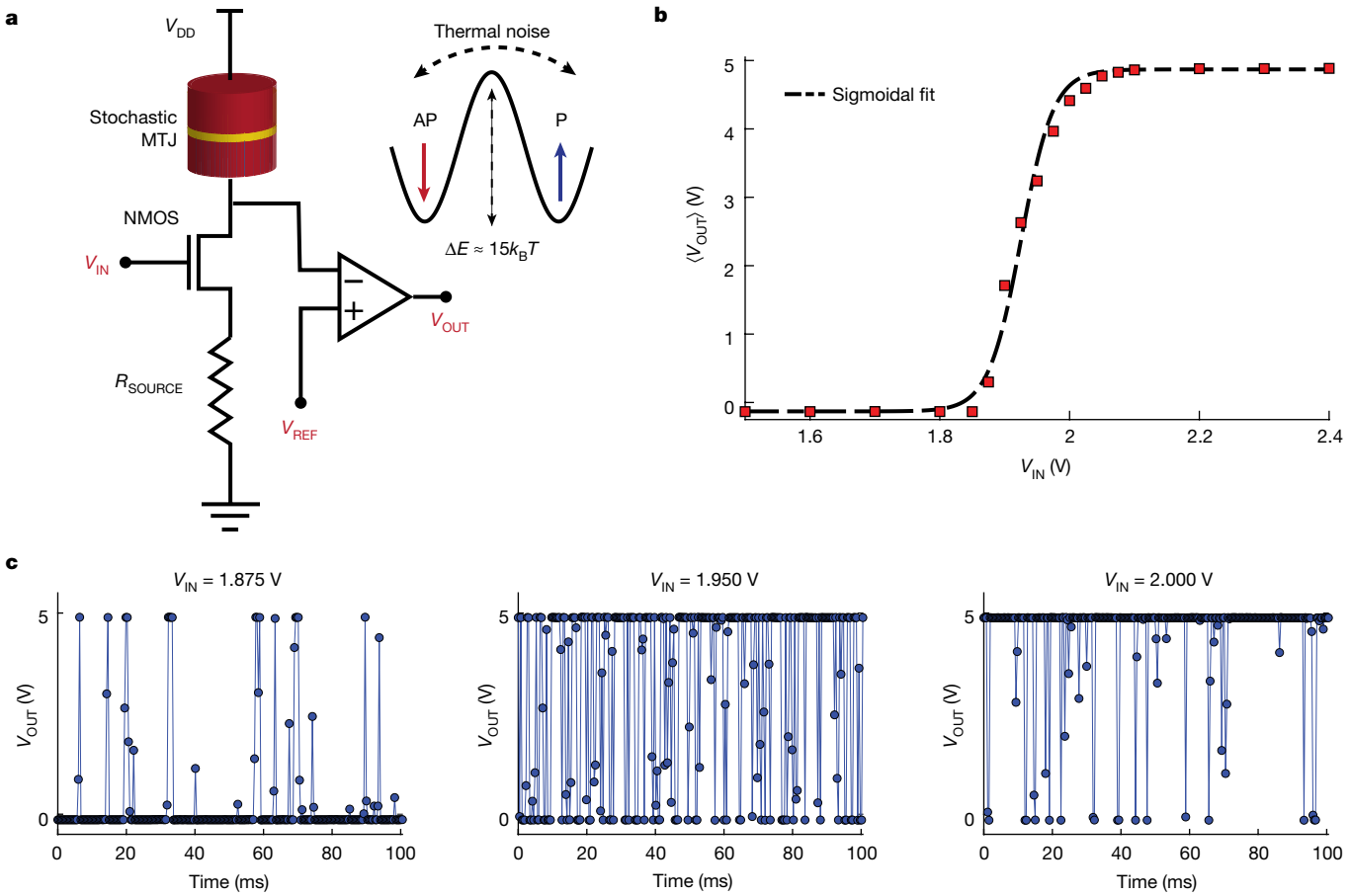


Fig. 2 | Experimental demonstration of a p-bit. a, Electrical schematic of a p-bit using a stochastic MTJ with an NMOS transistor, a comparator and a resistor, extending the design presented in ref. [16] to handle device-specific variations. A stochastic MTJ (S-MTJ) has a free layer with a relatively low energy barrier ($\Delta E \approx 15k_BT$) so that thermal noise makes it fluctuate between its stable states, one being parallel (P) to the fixed layer and the other being anti-parallel (AP). **b**, Time-averaged V_{OUT}, $\langle V_{OUT} \rangle$, as a function of the applied input, fitted to the sigmoidal function. Each point is averaged over 700 ms with 2,000 or more sampling points for each data point shown. **c**, Time snapshots of V_{OUT} for three different inputs V_{IN}, showing the preferred state of a p-bit (high or low) as a function of its input voltage.

all input functions are set to zero. Although the p-bits fluctuate independently in the uncorrelated state (top panel), non-zero input to the network results in two peaks observed at (5, 7) and (7, 5), showing that 35 is factorized into 5 and 7 correctly (bottom panel). Figure 3c shows the three-dimensional histogram obtained with the input functions appropriate for factorizing 161 using six p-bits with $P = 4$ and $Q = 2$, where the correct factor (23, 7) shows a prominent peak (bottom panel). Similarly, Fig. 3d shows an eight-p-bit network factorizing 945 ($P = 5$, $Q = 3$). Using p-bit models, we also simulate the factorization process and obtain agreement with experimental results using a single fitting parameter (see Methods section 'Experiment versus simulation'). We also investigate the influence of varying MTJ parameters such as R_P, R_{AP}, $I_{50/50}$, shift and distortion of the response of MTJs, and retention time τ. Response variations are corrected by adjusting the bias voltage $\nu_{0,i}$ (see Methods section 'Factorization experiment calibration') and variations in the retention time of the MTJs have little effect provided that the synapse is faster than the fastest p-bit (see Methods section 'Effect of p-bit parameter variation on system performance'). Owing to the relative ease of these methods, we expect robust and repeatable results for networks on even larger scales.

Next, we compare the demonstrated probabilistic computing system with its quantum counterpart. The present approach uses an algorithm that is similar to AQC but does not perform annealing, which normally requires coherence. Compared to AQC, the present scheme has a threefold advantage: it operates at room temperature, it can be implemented using existing highly scalable MRAM technology and it is relatively easy to incorporate complex many-body interactions into the scheme. Further, we note that for a subclass of quantum systems, quantum annealing can be approximated with replicated p-bit networks[26]. This class of systems is commonly referred to as 'stoquastic'[9]. The approximation becomes systematically more accurate upon increasing the number of replicas. The increased number of p-bits is offset by their comparably lower implementation costs (see Methods section 'Comparison between p-bit and quantum computing').

Probabilistic computing can also be executed using conventional complementary metal–oxide–semiconductor (CMOS) circuits. Our p-bit implementation uses three transistors and one MTJ, whereas CMOS-based probabilistic computing with digital random-number generators (RNGs) requires more than a thousand transistors to perform the same function. A quantitative comparison shows an energy advantage by a factor of 10 and an area advantage by a factor of 300 (see Methods section 'Comparison between MTJ-based p-bit and CMOS-based alternatives').

We should note that there are deterministic algorithms implemented on a fully digital CMOS system that specializes in performing factorization. However, this system takes a substantially greater amount of time to reach the exact solution as the problem size increases[27]. On the other hand, when algorithms that produce approximate solutions are acceptable, there is interest in hardware that enables probabilistic computing methods. Because the purpose of this study was to establish a system that is suitable for solving optimization problems in general, these factors mentioned above are very attractive, particularly considering the energy and surface area advantages.

In summary, this work serves as a proof-of-concept demonstration of an asynchronous probabilistic computer similar to the one envisioned

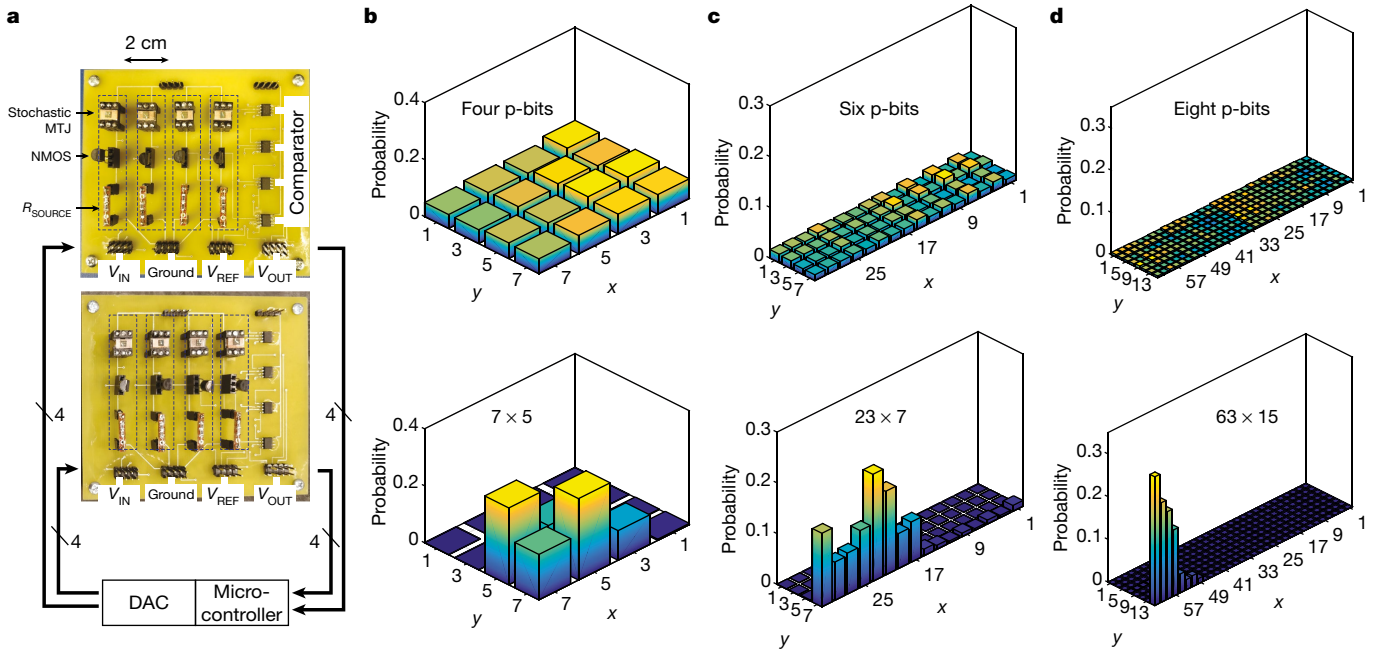


Fig. 3 | Experimental demonstration of integer factorization. **a**, A photograph of a printed circuit board for an eight-p-bit circuit, interconnected through a microcontroller and a DAC. **b**–**d**, The uncorrelated (top) and correlated (bottom) state of the system when four, six and eight p-bits are used to factorize $35 = 5 \times 7 = 7 \times 5$ ($P = 2$, $Q = 2$ with four p-bits; **b**), $161 = 23 \times 7$ ($P = 4$, $Q = 2$ with six p-bits; **c**) and $945 = 63 \times 15$ ($P = 5$, $Q = 3$ with eight p-bits; **d**). The x and y axes show the factors X and Y (see Methods section 'Factorization algorithm'). All statistics are taken over a window of 15 s with over 2,000 sampling points. Each separate factorization experiment was performed more than twice to ensure reproducibility.

by Feynman[1], which is realized through a slight modification of embedded MRAM technology currently at the level of 8 Mb and above[28] and which could find applications in the areas of optimization, sampling, and machine learning. An important aspect of this demonstration is the asynchronous operation of p-bits without any forced sequencing, unlike typical software implementations of Boltzmann machines, which require individual neurons or p-bits to be updated sequentially[29]. This asynchronous feature allows the parallel operation of a large number of p-bits, leading to an unconventional computing paradigm.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-019-1557-9.

Received: 17 January 2019; Accepted: 29 July 2019;
Published online 18 September 2019.

1. Feynman, R. P. Simulating physics with computers. *Int. J. Theor. Phys.* **21**, 467–488 (1982).
2. Shor, P. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **26**, 1484–1509 (1997).
3. Vandersypen, L. M. K. et al. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature* **414**, 883–887 (2001).
4. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
5. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220**, 671–680 (1983).
6. Geman, S. & Geman, D. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 721–741 (1984).
7. Sutton. B., Camsari, K. Y., Behtash, B.-A. & Datta, S. Intrinsic optimization using stochastic nanomagnets. *Sci. Rep.* **7**, 44370 (2017).
8. Camsari, K. Y., Faria, R., Sutton, B. M. & Datta, S. Stochastic p-bits for invertible logic. *Phys. Rev. X* **7**, 031014 (2017).
9. Albash, T. & Lidar, D. A. Adiabatic quantum computation. *Rev. Mod. Phys.* **90**, 015002 (2018).
10. Ikeda, S. et al. A perpendicular anisotropy CoFeB–MgO magnetic tunnel junction. *Nat. Mater.* **9**, 721–724 (2010).
11. Kent, A. D. & Worledge, D. C. A new spin on magnetic memories. *Nat. Nanotechnol.* **10**, 187–191 (2015).
12. Peng, X. et al. Quantum adiabatic algorithm for factorization and its experimental implementation. *Phys. Rev. Lett.* **101**, 220405 (2008).
13. Biamonte, J. Nonperturbative *k*-body to two-body commuting conversion Hamiltonians and embedding problem instances into Ising spins. *Phys. Rev. A* **77**, 052331 (2008).
14. Jiang, S., Britt, K. A., Humble, T. S. & Kais, S. Quantum annealing for prime factorization. *Sci. Rep.* **8**, 17667 (2018).
15. Ackley, D. H., Hinton, G. E. & Seinowski, T. J. A learning algorithm for Boltzmann machines. *Cogn. Sci.* **9**, 147–169 (1985).
16. Camsari, K. Y., Salahuddin, S. & Datta, S. Implementing p-bits with embedded MTJ. *IEEE Electron Device Lett.* **38**, 1767–1770 (2017).
17. Fukushima, A. et al. Spin dice: a scalable truly random number generator based on spintronics. *Appl. Phys. Express* **7**, 083001 (2014).
18. Mizrahi, A. et al. Neural-like computing with populations of superparamagnetic basis functions. *Nat. Commun.* **9**, 1533 (2018).
19. Julliere, M. Tunneling between ferromagnetic films. *Phys. Lett. A* **54**, 225–226 (1975).
20. Brown, W. F. Thermal fluctuations of a single-domain particle. *Phys. Rev.* **130**, 1677–1686 (1963).
21. Chaves-O'Flynn, G. D., Wolf, G., Sun, J. Z. & Kent, A. D. Thermal stability of magnetic states in circular thin-film nanomagnets with large perpendicular magnetic anisotropy. *Phys. Rev. Appl.* **4**, 024010 (2015).
22. Slonczewski, J. C. Current-driven excitation of magnetic multilayers. *J. Magn. Magn. Mater.* **159**, L1–L7 (1996).
23. Berger, L. Emission of spin waves by a magnetic multilayer traversed by a current. *Phys. Rev. B* **54**, 9353–9358 (1996).
24. Brataas, A., Kent, A. D. & Ohno, H. Current-induced torques in magnetic materials. *Nat. Mater.* **11**, 372–381 (2012).
25. Pervaiz, A. Z., Ghantasala, L. A., Camsari, K. Y. & Datta, S. Hardware emulation of stochastic p-bits for invertible logic. *Sci. Rep.* **7**, 10994 (2017).
26. Camsari, K. Y., Chowdhury, S. & Datta, S. Scaled quantum circuits emulated with room temperature p-bits. Preprint at https://arxiv.org/abs/1810.07144 2018).
27. Kleinjung, T. et al. in *Advances in Cryptology – CRYPTO 2010* (ed. Rabin, T.) 333–350 (Springer, 2010).
28. Lee, Y. K. et al. Embedded STT-MRAM in 28-nm FDSOI logic process for industrial MCU/IoT application. In *2018 IEEE Symposium on VLSI Technology* 181–182 (IEEE, 2018).
29. Roberts, G. O. & Sahu, S. K. Updating schemes, correlation structure, blocking and parametrization for the Gibbs sampler. *J. R. Soc. Ser. B* **59**, 291–317 (1997).
30. Endo, M. et al. Electric field effects on thickness-dependent magnetic anisotropy of sputtered MgO/CoFeB/Ta structures. *Appl. Phys. Lett.* **96**, 212503 (2010).

METHODS

MTJ fabrication. The MTJs are fabricated with a stack structure as follows, from the substrate side: Ta(5)/Pt(5)/$[Co(0.3)/Pt(0.4)]_7$/Co(0.3)/Ru(0.45)/$[Co(0.3)/Pt(0.4)]_2$/Co(0.3)/Ta(0.3)/$Co_{18.75}Fe_{56.25}B_{25}$(1)/MgO(1.1)/$Co_{18.75}Fe_{56.25}B_{25}(t_{CoFeB})$/Ta(5)/Ru(5)/Ta(50) (Fig. 1a). The numbers in parentheses are the nominal thicknesses in nanometres. The thickness of the free layer of CoFeB, t_{CoFeB}, is adjusted to view the change in the fluctuation of the MTJ magnetization. All films are deposited on a thermally oxidized silicon substrate by d.c. and radiofrequency magnetron sputtering at room temperature. The stacks are then processed into circular MTJs with nominal junction size varied from 40 to 80 nm in diameter by electron beam lithography and argon ion milling. The samples are annealed at 300 °C in vacuum for an hour under a 1.2 T perpendicularly applied magnetic field. MTJs are then cut out from wafers and bonded with wires to IC sockets to be placed in the p-bit circuit board.

MTJ characterization. First, the MTJ resistance is measured by sweeping the current from negative to positive values, and the time-averaged and high-frequency signals are read across a voltmeter and oscilloscope, respectively (Fig. 1a). We measured an approximate tunnel magnetoresistance ratio of 100% fluctuating between $R_P = 7$–11 kΩ and $R_{AP} = 12$–19 kΩ. The current at which the resistance switches by half is determined to be $I_{50/50}$, which is the bias current at which the MTJs will spend equal time in the AP and P states. To determine τ, we perform retention time measurements[31] when the MTJ is in either the AP or the P state using voltage measurements from the oscilloscope (Fig. 1b). To ensure reliable collection of data, each point is measured with a constant current on the oscilloscope at a sampling rate set ten times faster than the fastest fluctuation time of the MTJ. The retention time values are determined from approximately 1,000 to 10,000 switching events per device. The retention times used in this work range from 1 ms to 100 ms, which is suitable to match with the sampling rate of the microcontroller and DAC used to determine the inputs for each p-bit. For these purposes, we choose a free-layer thickness of 1.9 nm and different MTJ diameters (Fig. 1c).

p-bit construction. The p-bit is constructed following the circuit proposed previously[16] with two changes to the design: First, we use an additional resistance R_{SOURCE} attached to the source of the NMOS transistor to restrict the current through the MTJ branch to values in the stochastic range around $I_{50/50}$ (which is around 5–10 μA). This produces voltage fluctuations $V \approx 30$–50 mV. On the basis of measured values of $I_{50/50}$, and R_P, R_{AP} for every MTJ, an R_{SOURCE} value for each MTJ is calculated according to:

$$R_{SOURCE} = \frac{V_{DD}}{I_{50/50}} - R_{NMOS} - \frac{R_{AP} + R_P}{2} \tag{3}$$

where V_{DD} is the supply voltage and R_{NMOS} represents the drain-to-source resistance of the NMOS transistor.

Extended Data Fig. 1b shows the measured R_{NMOS} versus V_{IN} characteristics for a 2N7000 (T0-92-3 package) NMOS with drain resistance $R_D = 9.8$ kΩ, source resistance $R_S = 9.6$ kΩ and $V_{DD} = 200$ mV to mimic the p-bit circuit used in our experiment. The value of R_{NMOS} is chosen so that the p-bit is centred at $V_{IN} = 1.95$ V, as shown in Fig. 2b. This value of V_{IN} is optimized considering the transistor characteristics. A smaller value of V_{IN} makes the sigmoidal characteristics sharper because the current through the MTJ changes rapidly for small changes in V_{IN}, pinning the MTJ. If we choose values of V_{IN} greater than 1.95 V, the p-bit does not get saturated properly to V_{DD}.

Second, to achieve better gain, we use comparators (AD8692, 8-SOIC package) instead of the inverters used previously[16]. The drain of the NMOS is connected to the negative terminal of the comparator and a voltage V_{REF} is given as an input to the positive terminal. The comparator has a biasing current of 1 pA, which is 3–4 orders of magnitude lower than the current passing through the MTJ, ensuring that it does not load the MTJ branch. V_{REF} is chosen so that when $I_{50/50}$ is flowing through the MTJ branch, V_{REF} is centred at the drain voltage V_{DRAIN}. Under these conditions, V_{REF} can be calculated according to:

$$V_{REF} = V_{DD} - I_{50/50}\left(\frac{R_{AP} + R_P}{2}\right) \tag{4}$$

p-circuit construction. We have constructed our p-circuits following the general architecture described previously[25] which is shown in Extended Data Fig. 2. An Arduino microcontroller (Mega 2560) is used to read the output voltages of each p-bit as binary inputs and is programmed to implement the synaptic weights. These are then converted into analogue voltages using a DAC (PMOD DA4) that has eight channels, each with 12-bit resolution. The DAC also has an internal 2.5 V reference allowing a resolution of $2.5/4096 \approx 6.1$ mV. An important design consideration is to ensure that the interconnect delay—that is, the time it takes to update the inputs—is shorter than the retention time of the p-bits[25] (see Methods section 'Effect of p-bit parameter variation on system performance'). The DAC uses a Serial Peripheral Interface (SPI) protocol to communicate with the microcontroller and has a worst-case interconnect delay of 150 μs for eight p-bits, which is lower than the retention time of the MTJs used in this manuscript. We use an oscilloscope (MSO-X-3014T, Keysight) to collect the output voltages for all p-bits. The oscilloscope can read up to 16 digital voltages and is connected to a computer using the Keysight BenchVue oscilloscope software.

Factorization algorithm. To minimize the cost function E, we construct a network of binary stochastic neurons with the ith neuron driven by an input I_i obtained from evaluating $-\partial E(m_1, \ldots, m_N)/\partial m_i$, where m_i is the output of the ith neuron. This approach is similar in spirit to AQC[12] and a large amount of effort has gone into identifying appropriate cost functions for different problems of interest[32]; many of these formulations can also be adapted to design p-bit networks. The optimization-problem-based approach in this scheme is different from those in previous studies[8,25], in which integer factorization is cast as an inverse multiplication problem, which typically requires more p-bits to factor numbers of the same size.

For each number that is factored, the corresponding function is programmed into the synaptic function I_i, as explained below.

We start from a cost function of the form in equation (1)[14,33–36], which is simplified to:

$$\begin{aligned} E(x_P, \ldots, x_1; y_Q, \ldots, y_1) &= F^2 + \sum_{p,q} x_p y_q \left(2^{2p+2q} - 2^{p+q+1}F\right) \\ &+ \sum_{p,q,s\neq q} 2^{2p+q+s} x_p y_q y_s + \sum_{p,q,r\neq p} 2^{p+2q+r} x_p x_r y_q \\ &+ \sum_{p,q,r\neq p,s\neq q} 2^{p+q+r+s} x_p y_q x_r y_s \end{aligned} \tag{5}$$

using the property of binary digits that $b^2 = b$.

In this cost function, the numbers X and Y are assumed to be odd numbers, because large semiprimes of interest are always odd; this is implemented by setting x_0 and y_0 to 1. For a four-p-bit network, $P = 2$ and $Q = 2$ so that the cost function for $F = 35$ from equation (5) is obtained as below, where I_0, an arbitrary constant that controls overall strength of coupling, is chosen to be 1.

$$\begin{aligned} E = &-0.3x_1 - 0.7x_2 - 0.3y_1 - 0.7y_2 - x_2y_1 - 1.4x_2y_2 - 0.6x_1y_1 \\ &- x_1y_2 + 0.3x_1y_1y_2 + x_2y_1y_2 + 0.3x_1x_2y_1 + x_1x_2y_2 + 0.7x_1x_2y_1y_2 \end{aligned} \tag{6}$$

where the coefficients are rounded off to have one significant digit. By evaluating $-\partial E(m_1, \ldots, m_N)/\partial m_i$, we obtain the input functions I_i:

$$I_{x2} = 0.7 + 1.0y_1 + 1.4y_2 - 1.0y_1y_2 - 0.3x_1y_1 - 1.0x_1y_2 - 0.7x_1y_1y_2 \tag{7a}$$

$$I_{x1} = 0.3 + 0.6y_1 + 1.0y_2 - 0.3y_1y_2 - 0.3x_2y_1 - 1.0x_2y_2 - 0.7y_1x_2y_2 \tag{7b}$$

$$I_{y1} = 0.3 + 0.6x_1 + 1.0x_2 - 0.3x_1y_2 - 1.0x_2y_2 - 0.3x_1x_2 - 0.7x_1x_2y_2 \tag{7c}$$

$$I_{y2} = 0.7 + 1.0x_1 + 1.4x_2 - 0.3y_1x_1 - 1.0y_1x - 1.0x_1x - 0.7x_1y_1x_2 \tag{7d}$$

Similar cost functions—but with many more terms—can be obtained for the eight-p-bit experiment in which $P = 4$ and $Q = 4$. These cost functions and the resulting input functions are not listed here but are available upon request from the authors. Extended Data Fig. 3 shows the output of four p-bits x_2, x_1, y_2, and y_1 as a function of time, which are then used to collect the statistics shown in Fig. 3b.

Factorization experiment calibration. We begin by establishing an uncorrelated state for the p-circuit as a reference for the experiment. To offset variations, we first measure the average sigmoidal response of each p-bit used in our experiment. Extended Data Fig. 4 shows six such responses (15-s averages per point) for the six p-bits used in our experiment. Initially, we choose a value for R_{SOURCE} so that each sigmoid is centred at 1.95 V, and measure the average output. Any shifts in the average outputs from 1.95 V (due to variations in transistor characteristics and MTJ parameters) are adjusted as individual synaptic biases to centre the average response. Once these are set to obtain average responses that are aligned, they are not varied and an uncorrelated state for the system is established, as shown in Fig. 2 and in Extended Data Fig. 4b. After establishing the reference state, only the interconnect strengths between p-bits are changed for the remainder of the experiment.

Comparison between MTJ-based p-bit and CMOS-based alternatives. As noted in equation (2), the MTJ-based p-bit used in this work evaluates the function $m_i = \vartheta(\sigma(I_i) - r)$. Below we compare this evaluation to a digital-CMOS-based evaluation of the same function. As mentioned in the main text, the problem of factorization can be addressed with fully digital deterministic algorithms that do not require this function. However, the aim of this work is to demonstrate a broad approach to optimization and sampling problems using a network of p-bits interacting

asynchronously, in which high precision is not the primary figure of merit. With this in mind, we do not consider the deterministic algorithm and present below a functionality-based comparison between MTJ-based and CMOS-based probabilistic computers. To evaluate the same function $m_i = \vartheta(\sigma(I_i) - r)$ digitally using CMOS, one could use[37,38] an RNG for r, a look-up table for $\sigma(I_i)$ or a comparator for the step function ϑ.

In this section, we compare the energy and area of a CMOS-based pseudo-random-number generator (PRNG) to the MTJ-based p-bit (Extended Data Fig. 5). The look-up table and comparator would further add to the area of the CMOS-based p-bit. However, we note that the MTJ-based p-bit requires a DAC to interface with digital synapses. In principle this would not be needed for synapses implemented with analogue devices.

Extended Data Fig. 5 shows that the CMOS-based PRNG requires an energy consumption an order of magnitude higher and requires an area several orders larger compared to the MTJ-based p-bit in this work. Details of the models used are described below.

CMOS-based RNG. True RNGs operate specialized circuits using thermal noise from CMOS-based sources such as cross-coupled inverter pairs to produce true random bits[39]. However, inducing true randomness in conventional hardware typically requires high levels of energy consumption and large cell area. On the other hand, a PRNG-based approach that uses linear-feedback shift registers (LFSRs) offers a low-cost solution at the expense of reduced random bit quality[37].

We implement a 32-bit LFSR to form the PRNG that is composed of 32 D-type flip flops with three separate two-input XOR gates. Each XOR requires 14 transistors. Each D-type flip flop is composed of 36 transistors, which includes eight NAND gates (four transistors each) and two inverters (two transistors each). Therefore, the 32-bit LFSR requires 1,194 transistors in total. Each transistor is implemented using a minimum size (nfin = 1) 14 nm high performance fin field effect transistor HP-FinFET model obtained from a predictive technology model[40]. The details of the LFSR are shown in Extended Data Fig. 5. This circuit is simulated in the HSPICE circuit-simulator software with a clock frequency of 10 GHz ($\tau_{CLK} = 100$ ps). We note that because we are computing the energy per random bit, we average the active power over many clock cycles and so the exact clock frequency that is used in the circuit becomes irrelevant. The energy per random bit is obtained by integrating the total supply current (multiplied by the supply voltage) over one clock cycle. The energy per random bit for the 32-bit LFSR is about 20 fJ, as shown in Extended Data Fig. 5.

MTJ-based p-bit. For the MTJ-based p-bit simulation, we use the design proposed previously[16] with an MTJ of negligible energy barrier and with an autocorrelation time of about 100 ps for an arbitrarily chosen magnetization direction denoted as $m(t)$. The MTJ is modelled as a variable conductance with $G_{MTJ}(t) = G_0[1 + m(t)\mathrm{TMR}/(2 + \mathrm{TMR})]$, where TMR is the tunnelling magnetoresistance with a value of around 110%, close to the experimental value of TMR in our experiments. The average MTJ conductance G_0 (where ${G_0}^{-1} = 23.4\ \mathrm{k\Omega}$) is chosen to match the transistor conductance when $V_{IN} = 0$. This makes the sigmoidal response of the p-bit symmetric around zero. The instantaneous magnetization $m(t)$ is calculated by a stochastic Landau–Lifshitz–Gilbert (LLG) equation solver as a separate circuit in HSPICE. The stochastic LLG solver takes spin current as an input and produces $m(t)$ at each time step. The spin current is assumed to be proportional to the instantaneous charge current flowing through the MTJ, multiplied by a spin polarization P that in turn is assumed to be related to TMR by $\mathrm{TMR} = 2P^2/(1 - P^2)$ (ref. [41]).

The energy per random bit for the MTJ-based p-bit is calculated by computing the average power drawn from the supplies, $V_{DD} \times (I_{SUPPLY1} + I_{SUPPLY2})$, for a given period ($t = 100$ ns) and multiplying this average by the autocorrelation time of the low-barrier magnet to estimate the energy per random bit to be about 2 fJ per random bit. Extended Data Fig. 5c shows the difference in energy per random bit and the transistor count for the p-bit-based and hardware CMOS-based schemes.

Comparison between p-bit and quantum computing. The optimization algorithm used in this work is similar to an AQC algorithm that can run on quantum computing hardware. It has been shown[26] that a system of x qubits, if they belong to a class of 'stoquastic'[9] systems, can be efficiently emulated with $x \times r$ p-bits when using the Suzuki–Trotter decomposition, where r (about 10–100) is the number of replicas, each comprising x p-bits. Increasing the number of replicas systematically reduces the error compared to the exact solution; the increased number of p-bits is offset by their relative cheapness. Although many groups are working towards implementing 1,000 qubits, p-computers with density around 1 Gb could be a relatively near-term goal using embedded MRAM technology operating at room temperature. However, we note that the replicated p-bit approach to quantum computing is established only for a subset of quantum Hamiltonians that do not suffer from the 'sign-problem' associated with negative probabilities, and are commonly referred to as 'stoquastic'[9].

A recent experiment[14] performed on a D-Wave machine (D2000Q) used the same factorization algorithm—but with additional qubits to reduce the problem to two-body interactions—and factored 15 and 21 using four logical qubits, and factored 143 using 12 logical qubits. In general, $O(\log^2(F))$ logical qubits are required to factor an integer F. The increased number of qubits is a result of additional logical qubits in the Hamiltonian used to reduce the problem. By contrast, our demonstration factors numbers up to 945 with eight p-bits at room temperature and is estimated to be able to factorize 2^{n+2}-sized integers, with n p-bits.

Comparison of AQC and p-bits. We first describe the typical system—the transverse Ising Hamiltonian—that demonstrates an AQC algorithm for factorization and then present an emulation of this system with p-bits.

We show in Extended Data Fig. 6 that the results of an exact solution of the quantum many-body Hamiltonian can be accurately obtained by a replicated network of p-bits. The transverse Ising Hamiltonian for the factorization problem H_Q is given as:

$$H_Q = -\left(\sum_{i<j} J_{ij}\sigma_i^z\sigma_j^z + \sum_{i<j<k} K_{ijk}\sigma_i^z\sigma_j^z\sigma_k^z + \sum_{i<j<k<l} L_{ijkl}\sigma_i^z\sigma_j^z\sigma_k^z\sigma_l^z + \Gamma_X \sum_i \sigma_i^x \right) \tag{8}$$

where J_{ij}, K_{ijk} and L_{ijkl} represent the interactions obtained from the cost function $E = (XY - F)^2$ in equation (1), and Γ_X is the (dimensionless) transverse magnetic field that is used as an annealing parameter. The quantum system described in equation (8) can be mapped to a classical system with networks of p-bits. The classical Hamiltonian H_C for a classical system with r replicas is expressed as:

$$H_C = -\left(\sum_{n=1}^{n=r} \sum_{i<j} \frac{J_{ij}}{r} m_{i,n}m_{j,n} + \sum_{n=1}^{n=r} \sum_{i<j<k} \frac{K_{ijk}}{r} m_{i,n}m_{j,n}m_{k,n} + \sum_{n=1}^{n=r} \sum_{i<j<k<l} \frac{L_{ijkl}}{r} m_{i,n}m_{j,n}m_{k,n}m_{l,n} + \sum_{n=1}^{n=r} \sum_i J_\perp m_{i,n}m_{i,n+1} \right) \tag{9}$$

where $J_\perp$ is the local transverse coupling between replicas with periodic boundary conditions; $J_\perp = -1/(2\beta)\ln[\tanh(\Gamma_X\beta/r)]$ where β is the dimensionless inverse temperature.

In AQC, the system is prepared at a low temperature and the transverse magnetic field starts from a high value to initialize the system in its ground state. The magnetic field is then slowly reduced to keep the system in its ground state so that the ground state of the classical Ising Hamiltonian is reached.

Here, instead of performing annealing that requires a continuous change of the transverse magnetic field, we perform two static simulations, for factoring $161 = 23 \times 7$ using a small Γ_X (corresponding to a 'cold' system close to the ideal solution) and using a large Γ_X (corresponding to a 'hot' system close to thermal equilibrium).

We compare the results obtained by exactly solving the quantum system with those obtained by a classical simulation of p-bits. We note that quasi-static quantum annealing can also be performed using p-bits, but our purpose here is to show the correspondence between the exact quantum and the replicated classical system.

Exact quantum solution. For a small number of qubits, the many-body quantum Hamiltonian described in equation (8) can be solved exactly by methods of equilibrium statistical quantum mechanics:

$$\langle S \rangle = \frac{\mathrm{tr}[S_{op}\exp(-\beta H_Q)]}{\mathrm{tr}[\exp(-\beta H_Q)]} \tag{10}$$

where $\langle S \rangle$ is the expectation value of an observable corresponding to the operator S_{op} 'tr' represents trace. In this case, we choose S_{op} to correspond to all possible spin configurations corresponding to the different factors of the problem. We choose an inverse temperature of $\beta = 25$ and two magnetic fields $\Gamma_X = 0.1$ and $\Gamma_X = 0.5$. For each spin configuration $[y_2\, y_1\, x_4\, x_3\, x_2\, x_1]$, where $(y_i, x_i) \in \{-1, +1\}$, we compute the corresponding operator S_{op} to calculate the equilibrium probability.

Replicated p-bit simulation. The mapped classical system is simulated by first obtaining the current vector I_i for the i^{th} p-bit in the system from the classical Hamiltonian in equation (9) by $I_i = -\partial H_C/\partial m_i$. The same inverse temperature, $\beta = 25$ is chosen with $r = 45$ replicas and all p-bits are sequentially updated according to $m_i = \mathrm{sgn}[\tanh(\beta I_i) - \mathrm{rand}(-1, 1)]$, where rand is a number that that is uniformly distributed between -1 and $+1$. For each magnetic field $\Gamma_X = 0.1$ and $\Gamma_X = 0.5$ that enters $J_\perp$, $N = 2 \times 10^6$ time steps are chosen and a probability of each state is obtained using time averaging of the state of the system for the entire duration N of the simulation over all replicas r. Although the exact solution and the replicated p-bit simulation do not seem to show complete agreement at each state, the error can be systematically reduced by choosing a larger number of replicas; the error of the system scales as $O(1/r^2)$.

Experiment versus simulation. In this section, we compare our experimental work with ideal simulations performed using software. The simulation updates all p-bits every Δt, flipping the ith p-bit with probability $P_i = 1 - \exp(-\Delta t/\tau_i)$, where the dwell time τ_i of the ith p-bit depends on the inputs I_i obtained from the synaptic function: $\tau_i = \tau_{0,i}\exp(\pm I_i)$. Here $\tau_{0,i}$ is the zero-bias dwell time, and I_i is positive if it is parallel to the state of the p-bit and negative if it is anti-parallel. Extended Data Fig. 7a shows six simulated p-bits of an ideal system in which the average outputs versus inputs for all p-bits are identical. The retention times of the p-bits are much greater than the interconnect delay (about 1,000 times greater) such that $\tau_{\text{inter}} \ll \tau_N$, where τ_N is the smallest zero-bias dwell time among all p-bits.

By contrast, Extended Data Fig. 7b shows experimentally observed average behaviour of six p-bits where the device variations of the MTJs affect the alignment and shape of the average response. Using a simple correction in the synaptic weights (see Methods section 'Factorization experiment calibration'), experimental results of factorizing 161 (shown in Extended Data Fig. 7d) are fitted to computer simulations (Extended Data Fig. 7c) using a single fitting parameter $I_0 = 5$.

Effect of p-bit parameter variation on system performance. We investigate simulations using device parameter variations obtained from our experiments and elaborate on how to effectively mitigate them within certain limits. Extended Data Fig. 8 shows the effect of variations in retention times of the free layer on the overall performance of the system. In these simulations, the retention times for p-bits is varied from τ_N to $4\tau_N$ in all of the three cases shown. We conclude from our simulations that in general, for all p-bits that have retention times much slower than the interconnect delay ($\tau_{\text{inter}} = \tau_N$), the system will operate properly.

Extended Data Fig. 8c suggests that when $\tau_N = 10^1 \times \tau_{\text{inter}}$ the system fails to operate correctly. The exact boundary where the system stops working is a function of the type (linear versus nonlinear) and size (fan-in) of the synapse and the overall size (number of p-bits) of the system and in general requires a systematic study using a large number of p-bits.

Extended Data Fig. 9 shows the effect of variations of other MTJ parameters (R_P, R_{AP}, TMR, $I_{50/50}$) that are important for p-bit operation. Variations manifest themselves as either a misaligned average response of the p-bits or a distorted shape of the average behaviour of a p-bit. We correct the former in our experiments by measuring this shift and by adding an appropriate constant d.c. bias to the synaptic weights for each p-bit. The results of this procedure are simulated in Extended Data Fig. 9d–f. For all our experiments this procedure was performed to achieve an 'unbiased reference state', which is the first step of the factorization process. This process can be automated, for example using a control loop feedback mechanism such as a proportional–integral–derivative (PID) controller. The latter variation—the distortions in the shape of the average behaviour—are harder to correct, but in general their adverse effects on system operation seem minimal.

Owing to the ease of implementing compensation for device variations, as well as the recently reported market-ready MRAM showing lower levels of variation[42] compared to the experimental values obtained in this work, variation effects are not expected to become an issue as the size of the p-bit network scales.

p-bit-based implementation of invertible AND gate. A three-p-bit circuit of the type shown in the main text can implement an AND gate using x_2, x_1 as input p-bits and y_1 as an output p-bit with a cost function of the form[7,13]

$$E(x_1, x_2, y_1) = I_0(3y_1 + x_1x_2 - 2x_1y_1 - 2x_2y_1) \tag{11}$$

which minimizes the energy for configurations $\{x_2, x_1, y_1\}$ that satisfy the truth table. We use the same method as the main text to obtain the inputs I_{x2}, I_{x1}, I_{y1}:

$$I_{x2} = I_0(-x_1 + 2y_1)$$

$$I_{x1} = I_0(-x_2 + 2y_1)$$

$$I_{y1} = I_0(-3 + 2x_1 + 2x_2)$$

Extended Data Fig. 10a, b shows the direct mode of operation for the AND gate, with applied inputs leading to an output consistent with the inputs of any CMOS-based Boolean gate. Extended Data Fig. 10a, b shows a time snapshot and statistics for the three p-bits when both inputs are pinned to 1 by adding a large input voltage. The statistics for the direct mode of operation match well with the Boltzmann law (see main text) with the constant I_0 adjusted to 0.25.

A more interesting case is the inverted mode, in which an output is pinned and the inputs resolve themselves to be consistent with the applied output. Extended Data Fig. 10c shows a time snapshot of the p-bits when the output p-bit is pinned to 0. In this case, all three possible combinations of inputs appear, as shown by the statistics in Extended Data Fig. 10d.

The final case is when all p-bits are left floating. Extended Data Fig. 10e shows a time snapshot acquired for such a case, and Extended Data Fig. 10f shows the statistics. In this case the system goes through the four states consistent with the truth table of an AND gate.

Data availability

The datasets generated and analysed during this study are available from the corresponding authors on reasonable request.

31. Enobio, E. C. I., Bersweiler, M., Sato, H., Fukami, S. & Ohno, H. Evaluation of energy barrier of CoFeB/MgO magnetic tunnel junctions with perpendicular easy axis using retention time measurement. *Jpn. J. Appl. Phys.* **57**, 04FN08 (2018).
32. Lucas, A. Ising formulations of many NP problems. *Front. Phys.* **2**, 5 (2014).
33. Xu, N. et al. Quantum factorization of 143 on a dipolar-coupling nuclear magnetic resonance system. *Phys. Rev. Lett.* **108**, 130501 (2012).
34. Burges, C. J. C. *Factoring As Optimization*. Report No. MSR-TR-2002-83 (Microsoft Research Lab, 2002).
35. Henelius, P. & Girvin, S. A statistical mechanics approach to the factorization problem. Preprint at https://arxiv.org/abs/1102.1296 (2011).
36. Dridi, R. & Alghassi, H. Prime factorization using quantum annealing and computational algebraic geometry. *Sci. Rep.* **7**, 43048 (2017); erratum **7**, 44963 (2017).
37. Pervaiz, A. Z., Sutton, B. M., Ghantasala, L. A. & Camsari, K. Y. Weighted p-bits for FPGA implementation of probabilistic circuits. *IEEE Trans. Neural Netw. Learn. Syst.* **30**, 1920–1926 (2018).
38. Zand, R., Camsari, K. Y., Datta, S. & Demara, R. F. Composable probabilistic inference networks using MRAM-based stochastic neurons. *ACM J. Emerg. Technol.* **15**, 17 (2019).
39. Mathew, S. K. et al. μRNG: a 300–950 mV, 323 Gbps/W all-digital full-entropy true random number generator in 14 nm FinFET CMOS. *IEEE J. Solid-State Circuits* **51**, 1695–1704 (2016).
40. Zhao, W. & Yu, C. New generation of predictive technology model for sub-45 nm early design exploration. *IEEE Trans. Electron Dev.* **53**, 2816–2823 (2006).
41. Datta, D. et al. Voltage asymmetry of spin-transfer torques. *IEEE Trans. Nanotechnol.* **11**, 261–272 (2012).
42. Park, C. et al. Low RA magnetic junction arrays in conjunction with low switching current and high breakdown voltage for STT-MRAM at 10 nm and beyond. In *2018 IEEE Symposium on VLSI Technology* 185–186 (IEEE, 2018).

Acknowledgements We thank H. Sato, M. Bersweiler, T. Hirata, H. Iwanuma, K. Goto, C. Igarashi, I. Morita, R. Ono and M. Musya for technical support. We thank O. Hassan and S. Chowdhury for their help with the Methods sections comparing CMOS alternatives and quantum computing, respectively. A portion of this work was supported by ImPACT Program of CSTI, JSPS KAKENHI grant numbers 17H06093 and 19J12206, Cooperative Research Projects of RIEC, and ASCENT, one of six centres in JUMP, an SRC program sponsored by DARPA. W.A.B. acknowledges JST-OPERA.

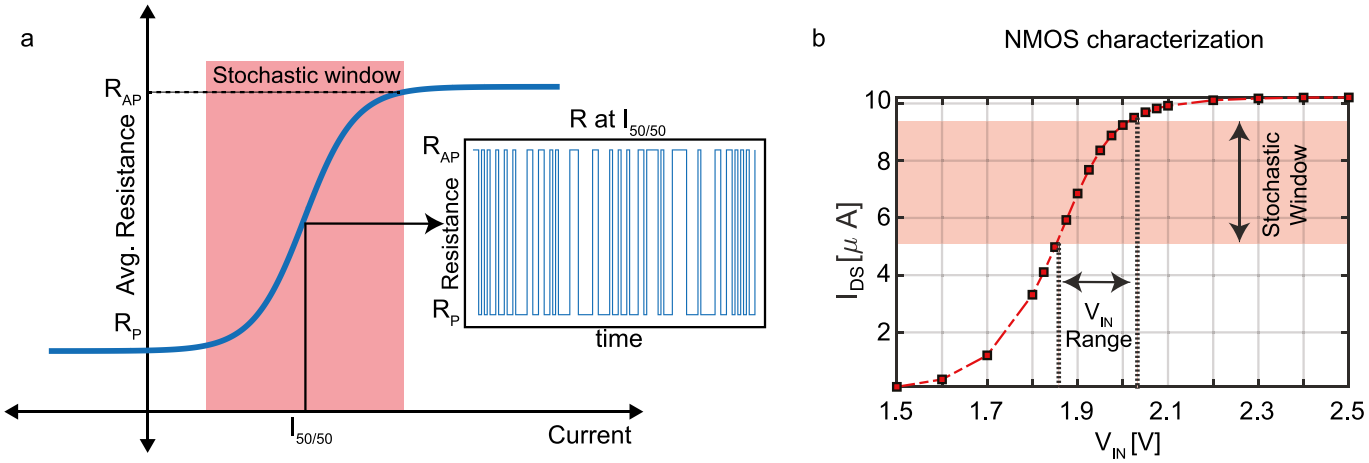
Author contributions S.F., K.Y.C., H.O. and S.D. planned the study. W.A.B. and S.F. prepared and characterized the MTJ devices. A.Z.P., K.Y.C. and S.D. developed the algorithm and experimental setup. A.Z.P. and K.Y.C. conducted factorization experiment and collected results. All authors contributed to the writing of the manuscript. All authors discussed the results.

Competing interests The authors declare no competing interests.

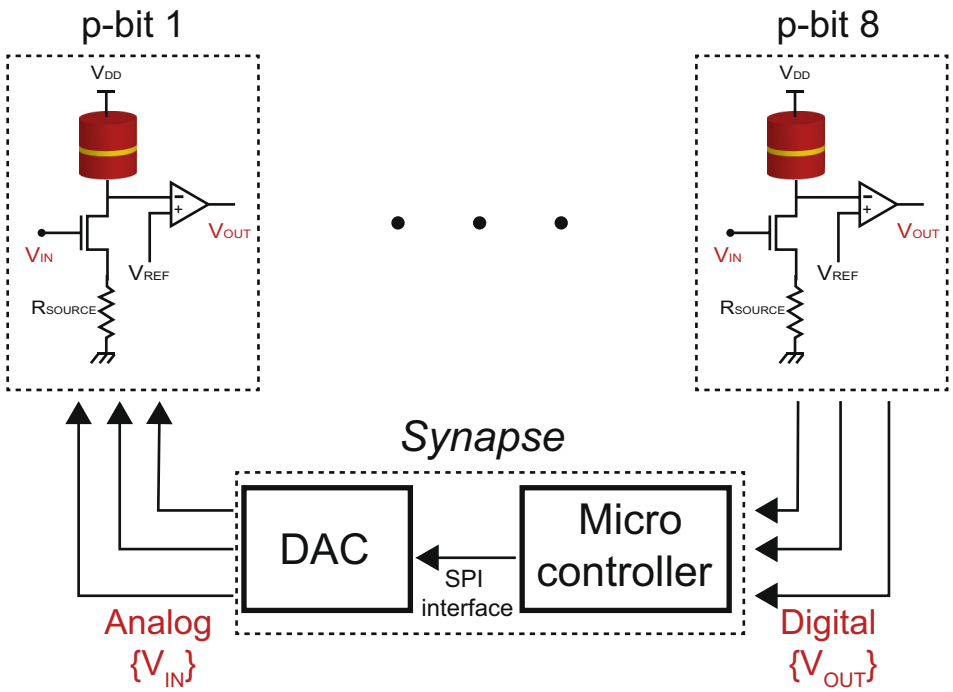
Additional information

Correspondence and requests for materials should be addressed to S.F. or K.Y.C.

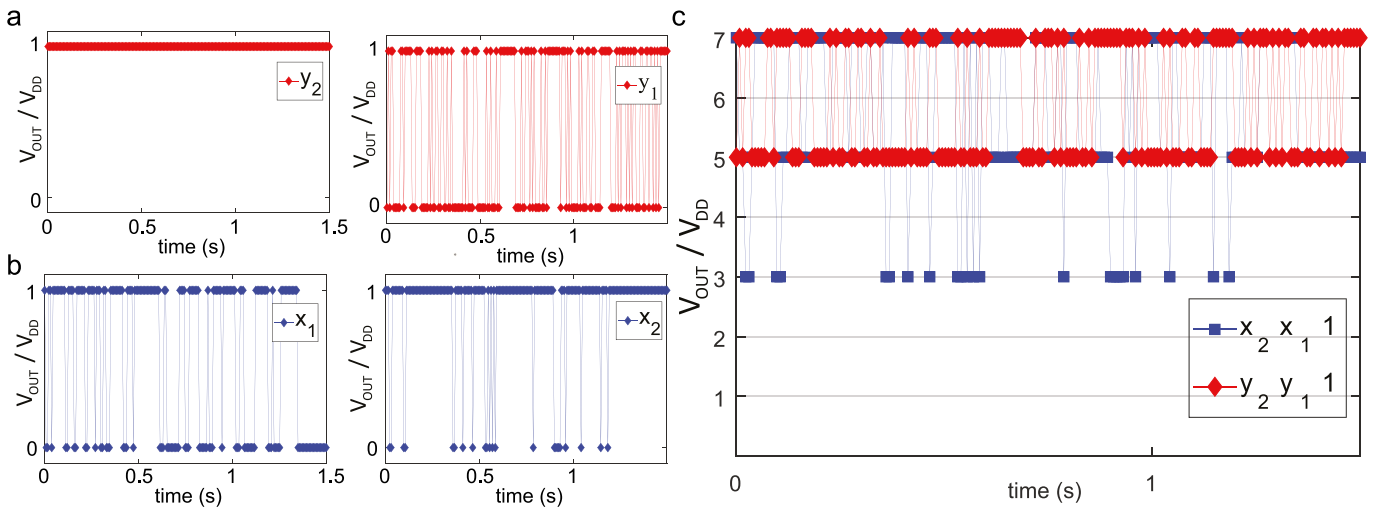
Peer review information *Nature* thanks Kyung-Jin Lee, Dmitri Nikonov and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.



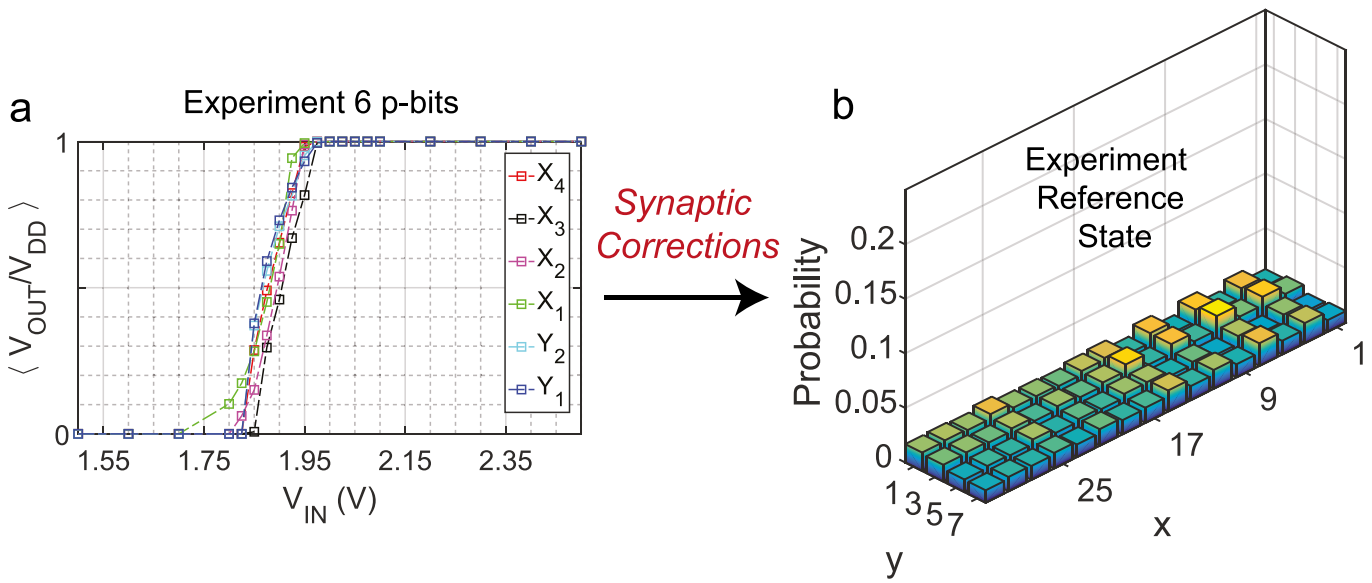
Extended Data Fig. 1 | p-bit construction. a, A diagram of the ideal response of a stochastic MTJ as used in this work and the parameters used to characterize the MTJ. **b**, The measured drain current I_{DS} as a function of V_{IN} of a 2N7000 NMOS transistor used in our p-bit demonstration.



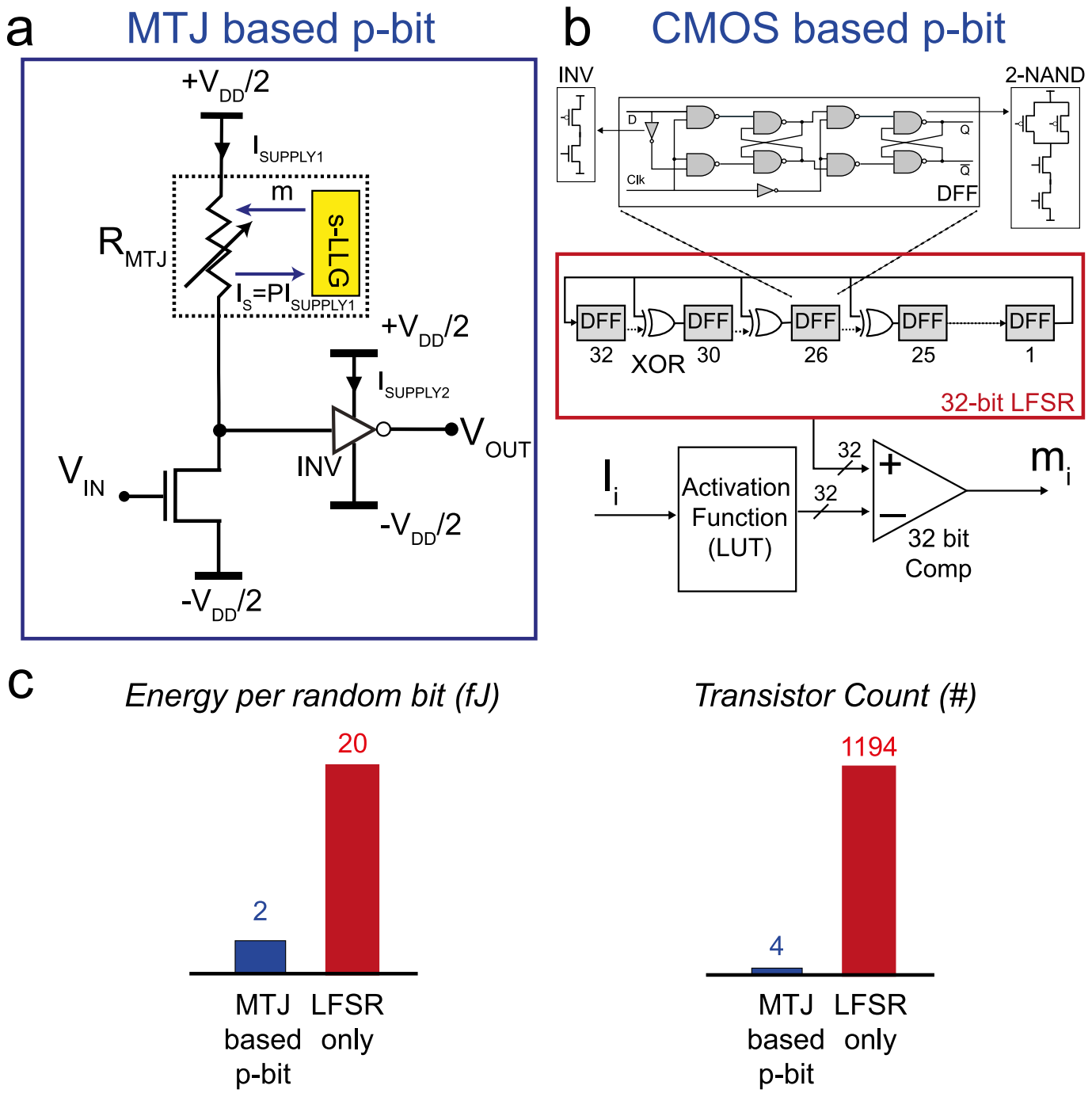
Extended Data Fig. 2 | Block diagram of an asynchronous p-circuit. A microcontroller reads the outputs voltages V_{OUT} of all p-bits and computes the synaptic weights, which are then converted to the analogue input voltages V_{IN} for each p-bit, using a DAC that communicates with the microcontroller.



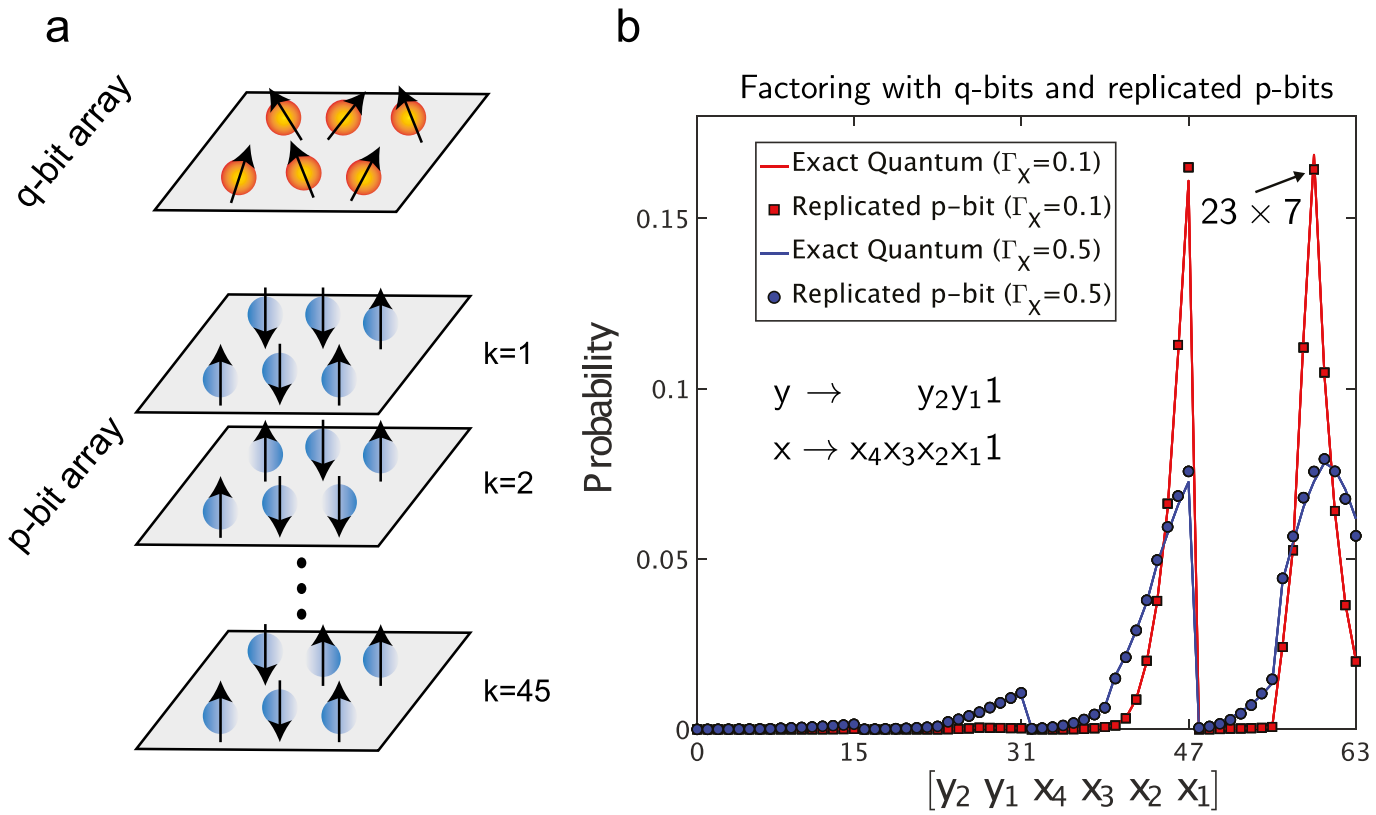
Extended Data Fig. 3 | Experimentally observed time snapshots. a–c, Experimentally observed time snapshots of the four p-bits used to factorize 35 (**a**, **b**). These snapshots are combined to create *x* and *y* (**c**), which fluctuate between 7 × 5 and 5 × 7.



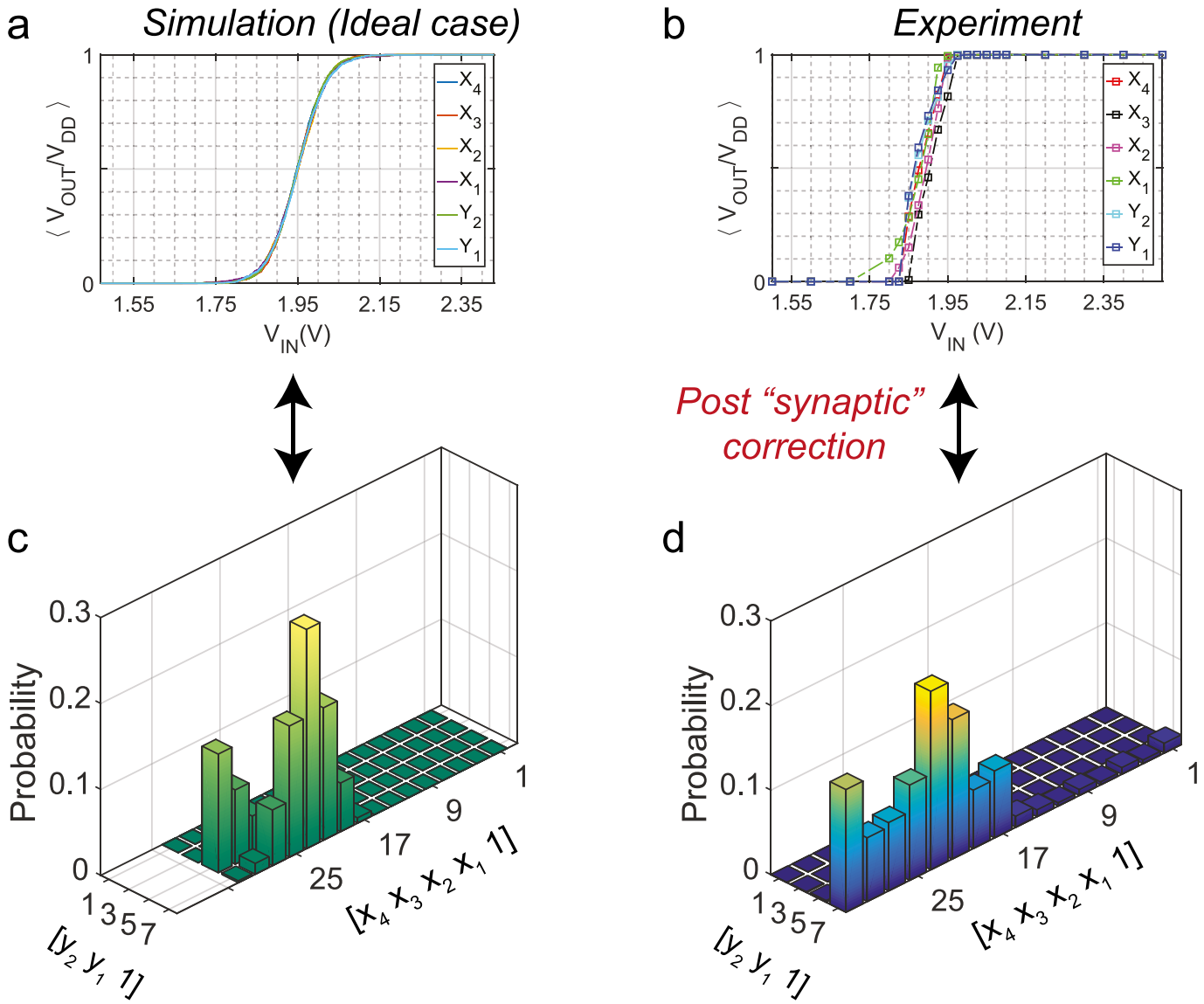
Extended Data Fig. 4 | Calibrating the experimental system. Calibrating a reference state using synaptic weights. **a**, The experimentally observed time-averaged output of six p-bits versus applied inputs (which are misaligned). **b**, The output is corrected using synaptic biases leading to the reference state shown. Each data point in **a** and **b** are taken as an average over a time window of 15 s with 2,000 or more sampling points.



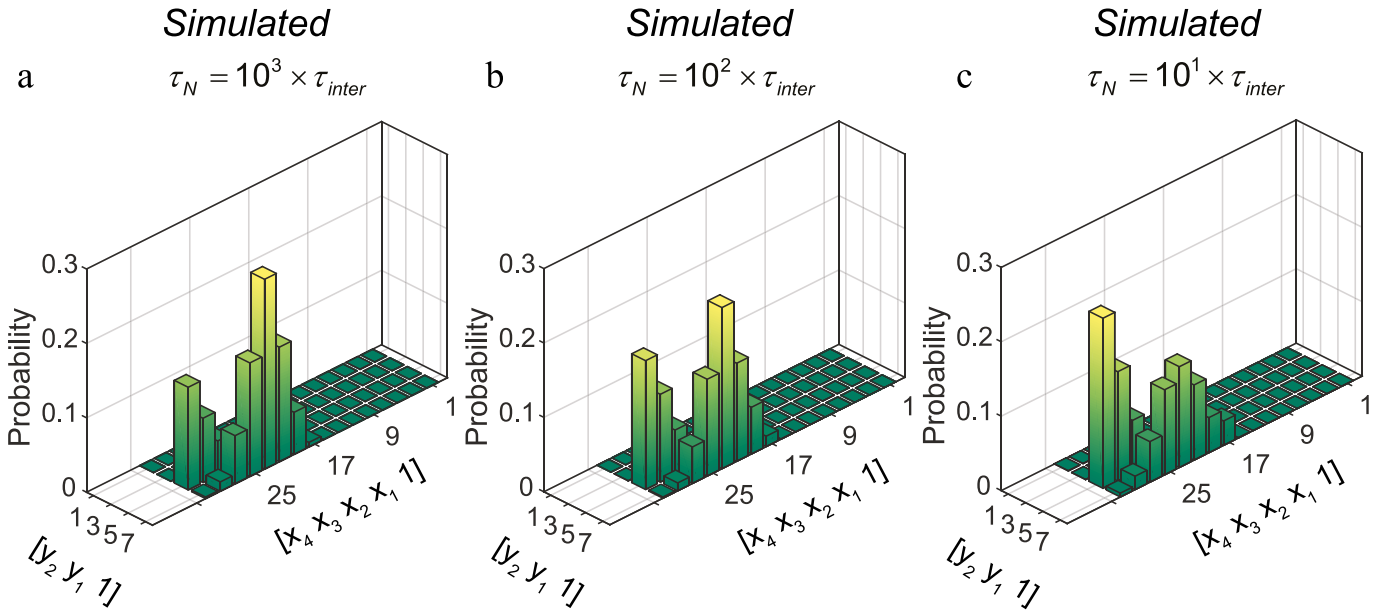
Extended Data Fig. 5 | Comparison between the MTJ- and CMOS-based energy per random bit and cell area. a, An MTJ-based p-bit simulated with the stochastic LLG model (s-LLG, dotted box). **b**, A 32-bit LFSR. The look-up table (LUT) and the digital comparator of the CMOS p-bit are not included in the comparison. INV, inverter; DFF, D-type flip flop.



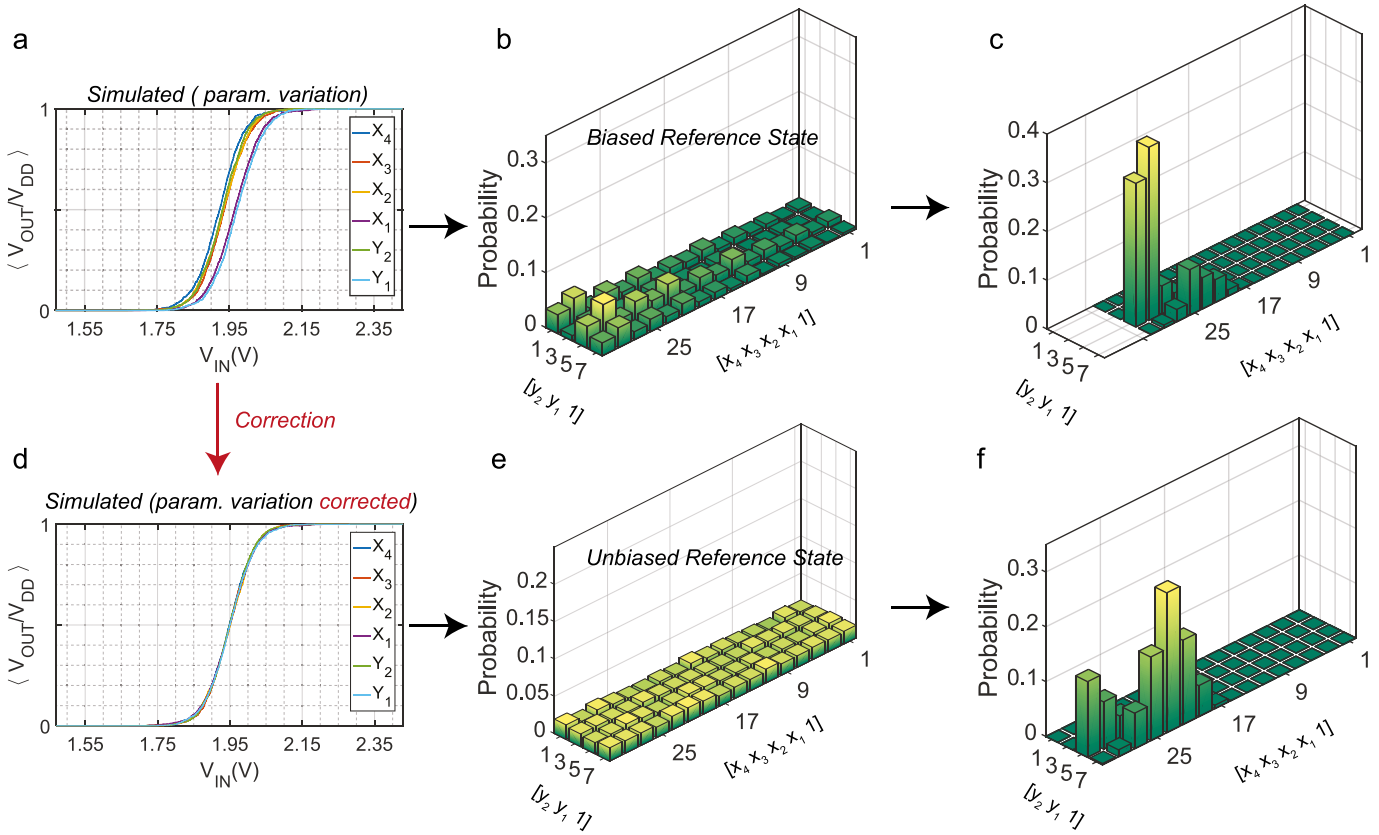
Extended Data Fig. 6 | Computing with p-bits versus AQC. a, A representation of how an array of six Ising spins in a qubit array can be replicated with an array of p-bits. **b**, A comparison of both approaches for factoring $161 = 23 \times 7$. For a system of six Ising spins, there are 64 states. At higher magnetic fields ($\Gamma_X = 0.5$) both systems are 'disordered' and the correct peak is not pronounced. At lower magnetic field ($\Gamma_X = 0.1$) the correct peaks emerge with a high probability. The states (y_i, x_i) have been converted to binary variables s_i from the bipolar variables m_i by defining $s_i = (m_i + 1)/2$ and the states $[y_2\, y_1\, x_4\, x_3\, x_2\, x_1]$ are expressed in decimal on the x axis.



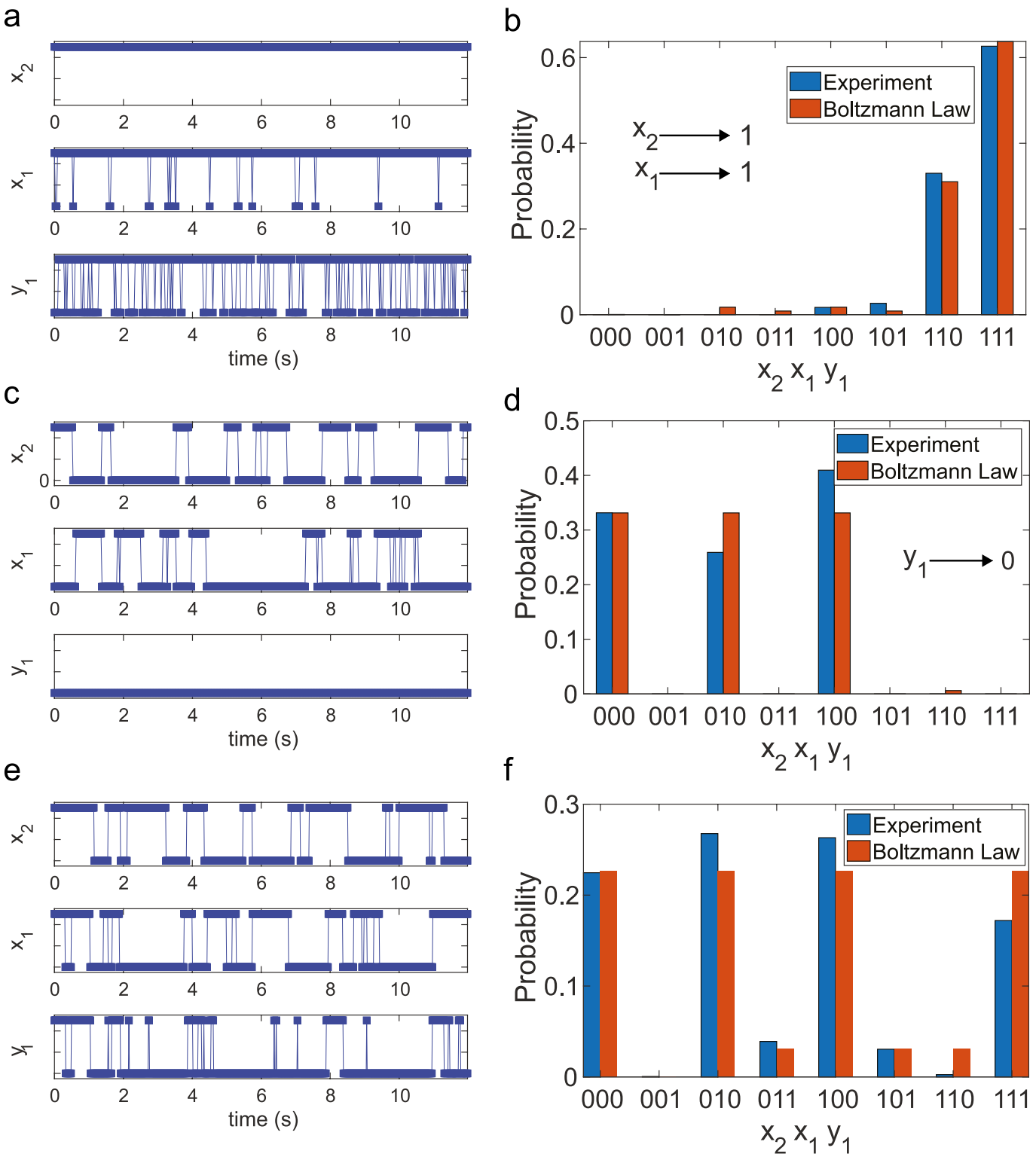
Extended Data Fig. 7 | Simulation versus experiment. a–d, We simulate the ideal experiment when all p-bits are perfectly aligned (**a**), using an idealized p-bit model which produces the results shown in **c**. Each data point is taken as an average over a time window of 15 s with 2,000 or more sampling points. The presence of device variations leads to a non-ideal system of misaligned p-bits (**b**), which is corrected using synaptic biases, allowing the experiment to approach the correct results (**d**). The time-averaged statistics in **b** are collected over a time window of 15 s with 2,000 or more sampling points.



Extended Data Fig. 8 | Simulation of variations of τ_N. The τ of six p-bits is varied from a minimum value of τ_N to a maximum value of $4\tau_N$. Variations between p-bits do not affect system operation providing that $\tau_{\text{inter}} = \tau_N$.



Extended Data Fig. 9 | Simulations of variations of MTJ parameters. **a**–**c**, The variation of MTJ parameters results in the misalignment of the average responses of the p-bits (**a**), which results in a biased reference state (**b**). When such a system is used for factorizing 161 the observed results are incorrect (**c**). **d**–**f**, The shifts in the average responses are corrected using synaptic biases (**d**), which correct the reference state (**e**) and factorization results (**f**).



Extended Data Fig. 10 | Invertible AND gate operation. a, **b**, Time snapshot for the direct mode of operation when the inputs x_2 and x_1 have both been pinned to 1 (**a**); the statistics collected for 60 s (**b**). **c**, **d**, Time snapshot for the p-bits operating the AND gate in inverted mode when the output y_1 is pinned to 0 (**c**); the statistics collected for 60 s (**d**). **e**, **f**, Time snapshot for the p-bits operating the AND gate in floating mode (**e**); the statistics collected for 60 s (**f**). All statistics shown are collected over a time window of 60 s with 2,000 or more sampling points.