

Spatially programmable origami networks enable high-density mechanical computing for autonomous robotics

Received: 1 April 2025

Xinyu Hu¹, Ting Tan²✉, Yinghua Chen² & Zhimiao Yan¹✉

Accepted: 26 September 2025

Published online: 20 November 2025

 Check for updates

Mechanical computing enables logic decision-making, allowing direct computational integration into robotics to enhance their autonomy in complex environments. However, current non-universal logic designs hinder reconfigurability in multifunctional mechanical computing systems. Complexity-multifunctionality trade-offs limit mechanical computing materials to single logical operations and low computational density. Here, we address these limitations using origami metamaterials with reconfigurable conductive networks, enabling high-density programmable logic via physical reorganization. By rotating intra-gate elements to modify AND/OR-based Boolean cascades, the design reduces gates by 46.7% compared to standard arrays, executing arithmetic and comparison operations efficiently. Shared tree-like cascades allow multiple functions with minimal redundant gates. The system via Rubik's Cube mechanics supports three-axis reconfiguration of Buffer/NOT elements, achieving reconfigurable full-adder/subtractor and computational densities up to 1728. Integrated robotics demonstrate autonomous right-angled and curved path planning through reprogrammable half-adder/subtractor logic. This framework provides a universal, scalable design-methodology for high-density mechanical computing, with implications for robotics and embodied intelligence.

Mechanical computing is an unconventional information processing method based on mechanical motion or deformation for logical operations^{1–4}. Despite inferior speed and size versus electronic computing, it offers value in niche applications: complex environments, safety-critical systems, and energy autonomy^{5–7}. The mechanical computing can utilize metamaterials as a medium to abstract the unit-cell into information bits, enabling logical operations through force input^{8–10}. Additionally, the mechanical computing marks a revolutionary change in robotic systems. Instead of relying on a central computing chip, the mechanical computing metamaterial is used to be their physical bodies, integrating distributed embodied intelligent modules for autonomous perception, decision-making, and actuation – much like living organisms do^{11–13}. This bio-inspired design makes

robots more flexible, reduces system-wide failures through built-in backups, and helps them adapt to unpredictable environments¹⁴. A key challenge now is building computation directly into structural materials^{15–18}. This would allow physical components to make smart decisions with the robot's intelligence, not just serve as passive parts.

Current research facilitates the deep integration of functional materials and robotic systems composed of simple logic gates, enabling the execution of unitary motions through mechanical computation^{5,6}. For instance, Conrad et al. embedded three NOT logic gates within a fully integrated soft robot, achieving linear locomotion driven by pneumatic actuation⁵. Yan et al. configured an origami multiplexed switch based on bistable structures and artificial muscles, generating digital logic gates and memory units to construct an

¹State Key Laboratory of Ocean Engineering, Department of Engineering Mechanics, School of Ocean & Civil Engineering, Shanghai Jiao Tong University, Shanghai, China. ²Meta-mechanics Lab, Shanghai Jiao Tong University, Shanghai, China. ✉e-mail: tingtan95@outlook.com; zhimiaoy@sjtu.edu.cn

integrated autonomous origami robot. An AND-gate-based pre-catching robot captures “live prey” via force closure; an OR-gate-based self-reversing legged robot retreats upon obstacle encounter; and an origami wheeled car, utilizing a 2-bit memory disk design, executes corresponding motion trajectories sequentially under stored command sequences⁶. Byun et al. implemented a soft machine employing one AND gate and three OR gates to emulate the differential intensity folding response mechanisms of the *Mimosa pudica* plant¹⁹. While these implementations of mechanical computing functional materials confer adaptive capabilities to robots, the inherent simplicity of their logic functions constrains advancement in both physical domains (e.g., lightweight construction) and cognitive domains (e.g., complex data processing). Further advances in mechanical computation within the domains of high integration and self-optimizing neural networks have laid a more solid foundation for the intelligence of embodied robotic systems^{20–22}. Among these, Helou et al. established Boolean-kinematic equivalences, realizing 2-bit mechanically integrated computation (adder/subtractor/multiplier)²⁰. Mei et al. developed an in-memory mechanical computing architecture supporting neuromorphic computation²¹. However, as distributed computing systems, mechanical logic systems within embodied robots are typically customized for single or a few specific tasks^{20–23}. Consequently, reconfigurable approaches have been introduced into mechanical computing system design, aiming to achieve in-situ adjustment of the logical functions of functional materials. For example, the curved-beam-based reconfigurable mechanological metamaterial proposed by Mei et al. realized the reconfiguration of logic functions, such as, half-adders, and latches, based on combinational and sequential logic⁸. The reprogrammable metamaterial processors for soft robots developed by Jiao et al. achieved functional reconfiguration for full adders and full subtractors²⁴. Although the aforementioned studies enrich logical functionalities, a discrepancy exists in the number of unit-cells required for reconfigurable logic gates within such logic operation systems, which rely on the coupling of unit-cell mechanical responses or the Laws of Physical Boolean Algebra. This discrepancy necessitates an increased number of redundant cells for achieving logic reconfigurability, resulting in reduced computational density within the mechanical computing architecture and increasing the complexity of embodied intelligent robotic systems.

Based on the design principle of minimized canonical combinational logic functions, different logic gates can be reconfigured using a minimal and identical number of logic units, thereby reducing the quantity of redundant components within the system to resolve the aforementioned issue. Here, we propose a high-density programmable embodied computing method using origami-inspired conductive network reconfiguration. By leveraging the kinematics and mechanics of reconfigurable origami switching elements, we bridge mechanical robustness with electronic-like computational versatility, enabling complex, adaptive decision-making in dynamic environments. A two-dimensional (2D) reprogrammable meta-array dynamically reconfigures Boolean cascades to reduce logic gate counts and execute arithmetic, comparison operations efficiently. The proposed approach utilizes Buffer/NOT elements as finer-grained reconfigurable unit-cells compared to traditional logic gates in FPGAs/DPGAs. This enables intra-gate reorganization of logic functions, thereby significantly reducing the number of gates reserved for future reconfigurations and increasing computational density through improved unit-cell utilization. To further enhance mechanical computational integration and reduce the complexity of reconfigurable control, the three-dimensional (3D) reprogrammable logic cubes integrate Rubik's Cube-like reconfigurability with Buffer/NOT switching elements are designed for realizing subtractor/adder logic computing and autonomous path planning in robotics. This fusion of mechanical intelligence and electronic-like programmability in structural matter provides

insights into redefining the boundary between computation and physical embodiment, where materials no longer passively execute algorithms – they become the algorithm, ushering in an era of self-evolutionary machines capable of cognitive emergence through embodied environmental interactions.

Results

Design concept

Field/DNA programmable gate arrays (FPGA/DPGA) are composed of single-functional logic gates, such as AND, OR, and NOT (Fig. 1a). Programmability is achieved by re-selecting and rewiring the logic gates within the array. For instance, compared to array A with the same inputs, which performs a III-level logic operation from logic gates 1, 4 to 2, 8 to 6, array B performs a III-level logic operation from logic gates 1, 4 to 5, 8, to 3 by enabling logic gate 5 (OR) and 3 (AND) while disabling logic gates 2 (AND) and 6 (OR), thus changing the output f from $ab(c+d)+\bar{e}$ to $(ab+c+d)\bar{e}$. This reconfigurable computation completes 2^{2n-1} combinatorial logic functions when the array contains $3n-2$ AND, OR, and NOT gates (Supplementary Discussion 1). To enrich the functionality, we propose a reprogrammable gate meta-array (Fig. 1b) constructed by reconfigurable logic gates. The reconfigurable logic gate operating AND and OR is composed of Buffer and NOT unit-cells, and its mutual transformation of logic functions is achieved by relative rotation and reorganization of the two unit-cells, thus enabling the programmability of the gate meta-array. For example, logic gate 2 transforms from AND to OR, resulting in a change of item $ab(c+d)$ in output f to item $ab+c+d$. And logic gate 6 transforms from OR to AND, leading to another change of item $(c+d)+\bar{e}$ in output f to item $(c+d)\bar{e}$. As a result, $ab(c+d)+\bar{e}$ changes to $(ab+c+d)\bar{e}$ accordingly. This reconfigurable computation requires only $2n-1$ reconfigurable AND, OR, and NOT logic gates to ensure the 2^{2n-1} combinatorial logic functions.

In the 2D reprogrammable gate meta-array, reconfiguration occurs within the logic gates. Inspired by embodied intelligence in biological systems capable of dynamically adjusting their functional characteristics to interact with and adapt to various complex and changing environments^{13,25}, a reprogrammable logic cube with space utilization and high function density is proposed. The Buffer and NOT switching elements are not designed to form gate arrays but are arranged in series for AND operations and parallel for OR operations based on sum of product Boolean functions (Fig. 1c). The origami switching elements of the product terms are serially distributed in an inter-layer column of the reprogrammable logic cube, with identical inputs placed in the same layer. The outputs of the serially connected origami switching elements are parallelly connected to complete the sum terms. The block transformation of origami switching elements at the corner or edge positions within layers enhances the combinational freedom between inter-layer origami switching elements, thereby increasing the computational power density. This high computational power density reprogrammable logic cube can serve as a decision-making module, forming part of an embodied autonomous robotic system (Fig. 1d). Combined with perception and enforcement modules, the 3D reprogrammable logic cube can perceive information, make action decisions, and execute commands to facilitate interactions with the physical environment.

Buffer/NOT origami switching element and reconfigurable logic gate

The logical operations and reconfiguration of the logic gate are accomplished by the deformation and reorganization of intra-gate elements. The intra-gate elements are based on the origami unit-cell designed as a fusion of Miura-ori and waterbomb bases, possessing pseudo single degree-of-freedom and flat-foldable deformation characteristics (Fig. 2a). Details on how to design the flat-foldable patterns can be found in the Supplementary Information of ref. 26. The origami

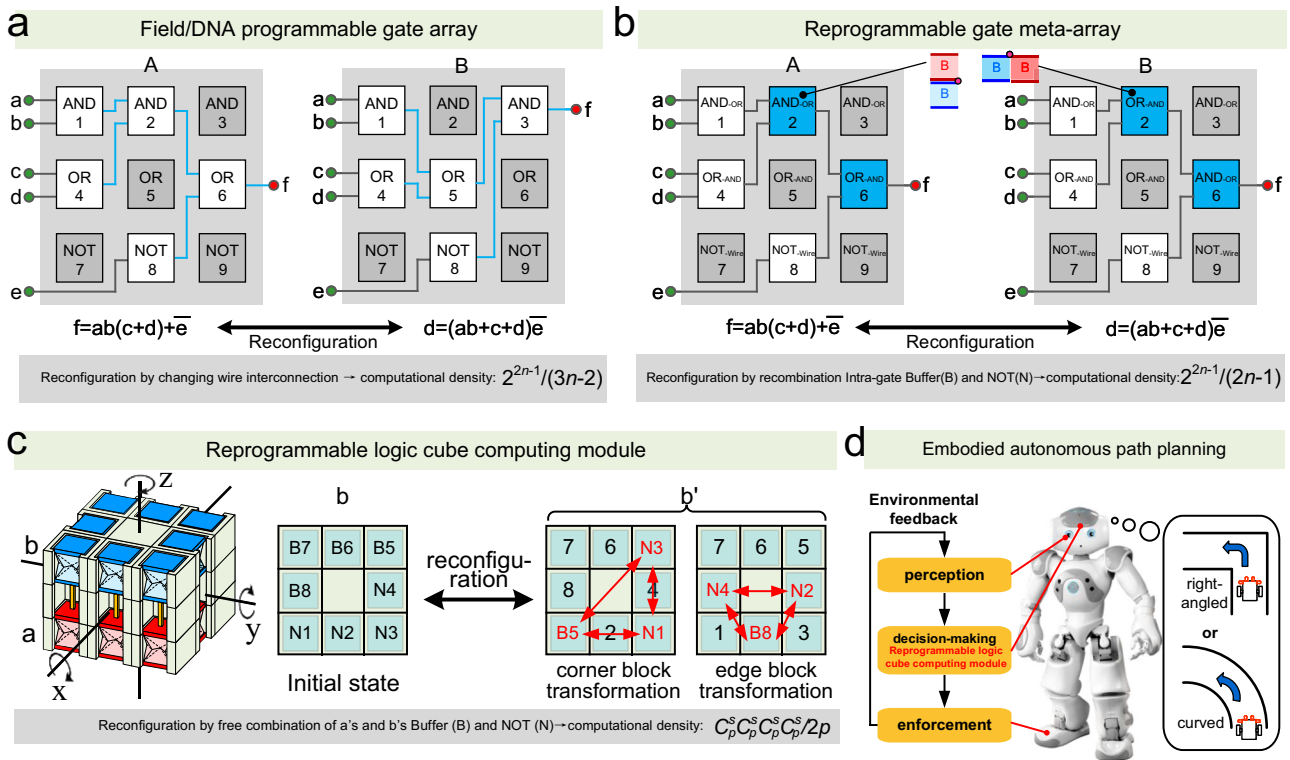


Fig. 1 | Design concept of 2D reprogrammable gate meta-array and 3D reprogrammable logic cube. **a** Reconfiguration of traditional FPGA/DPGA through modifications in wiring connection patterns. In array A, logic gates 1, 2, 4, 6, and 8 are connected to form a logical expression, while in array B, logic gates 1, 3, 4, 5, and 8 are connected to form a logical expression. The following equation represents the maximum computational power density achievable by FPGA/DPGA. n represents the number of operation inputs through I -level logic gates in the array. **b** Reconfiguration of the designed 2D fixed-wiring reprogrammable gate meta-

array (our work) composed of reconfigurable logic gates. The following equation represents its maximum computational power density. **c** 3D reprogrammable logic cube and its logic reconfiguration method (our work). Blue and red unit-cells represent the basic elements, such as Buffer/NOT, as shown in **(b)**. The following equation indicates its maximum computational power density. Where s equals 2, i.e., two origami switching elements. And p equals 4, that is, the number of corner/edge blocks. **d** The schematic of an embodied autonomous robotic system for path planning and three functional parts.

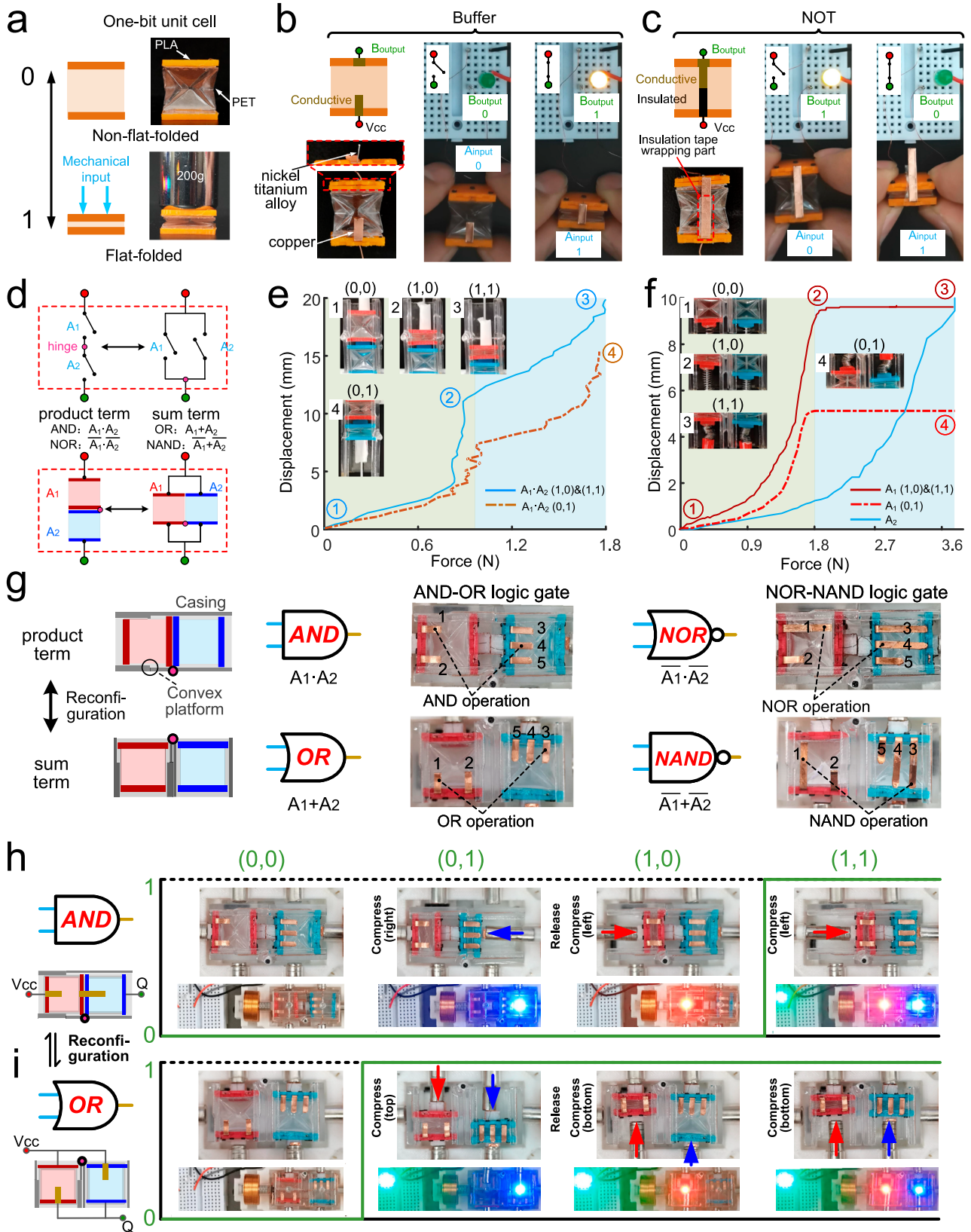
unit-cell with different stable height H_{stable} can present full-height tensile, full-height compressible, and partial tensile/compressible mechanical behaviors. The origami unit-cell with full-height compressible complete-elastic mechanical behavior is used to design logic elements due to its significant deformation from H_{max} to 0 and external force application. Different folding states of the origami unit-cell are abstracted as a 1-bit mechanical input: when subjected to axial compression, the non-flat-folded state represents input 0, and the flat-folded state represents input 1. The origami unit-cell, combined with copper sheets and titanium-nickel alloy wire conductive networks, forms the origami switching element (see “Methods” section for details about its fabrication). A truncated copper sheet enabling exclusive electrical contact in the flat-folded state configures the origami switching element as a Buffer logic gate (Fig. 2b), maintaining direct signal correspondence as validated in Supplementary Movie 1. Conversely, an elongated copper sheet with an insulated base achieves conductive states solely during non-flat-folded transitions, implementing a NOT logic operation (Fig. 2c) that inverts input signals.

According to integrated circuit switch theory²⁰, 2-bit logic gates AND and NOR, represented as mathematical product term, is viewed as two switching elements A1 and A2 in series, while 2-bit logic gates OR and NAND, represented as mathematical sum term, is viewed as two switching elements A1 and A2 in parallel (Fig. 2d). The AND logic gate is implemented through serial interconnection of two Buffer-type origami switching elements, while the NOR gate is achieved via analogous serial integration of NOT-type elements. Parallel arrangement of Buffer elements produces OR gate functionality, whereas NAND operation emerges from equivalent NOT-element serialization. Rotary hinge-

mediated reconfiguration enables topological metamorphosis between series and parallel architectures, permitting dynamic AND \leftrightarrow OR and NOR \leftrightarrow NAND transformations through coordinated element rotation.

The physical implementation of 2-bit logic gates requires two origami switching elements to mechanically encode dual-input states through deformation-controlled parametric modulation. The crease-generated mechanical force follows $F = Elh^3/12b \cdot \Delta\varphi d\varphi/dH^{26}$, where E is Young’s modulus of crease material, l , h , and b are the length, thickness, and width of the crease, and φ is the dihedral angle of the crease. In geometrically uniform switching elements, crease dimensions l and b remain constant, while $\Delta\varphi$ maintains consistency across unit-cell variants (Supplementary Discussion 2). The variations H in deformation are engineered through thickness modulation under iso- E material conditions at constant applied force. This thickness-graded PET implementation enables binary state combinations: (0,0), (0,1), (1,0), and (1,1) through precision-engineered unit-cell fabrication.

Under uniform loading, the displacement-force profile of series-connected origami unit-cells exhibits a characteristic double-J morphology (Fig. 2e, Supplementary Movie 2). The deformation initiates with the red thin unit-cell undergoing progressive folding (first J-phase) while the blue thick unit-cell remains quasi-static, representing the input state (0,0) (states ①-②). Beyond 1 N, the red unit-cell achieves complete flat-folding, transitioning to (1,0) (states ②-③), followed by significant blue unit-cell deformation (second J-phase) culminating in dual flat-folding at 1.8 N, establishing (1,1) (state ③). Structural stabilization is achieved through a casing-integrated design (Supplementary Fig. 7), where a convex platform on the red unit-cell’s casing realizes



directional deformation variation. This constrained kinematics, enabled by asymmetric compressive loading, restricts red unit-cell flat-folding while permitting unimpeded blue unit-cell deformation, thereby realizing the input (0,1) (state ④). Parallel configurations demonstrate independent deformation kinematics under identical loading (Fig. 2f), with sequential thin-to-thick unit-cell folding

transitioning through (0,0) → (1,0) → (1,1) (states ①→③) and (0,0)→(0,1) (states ①→④).

The AND-OR and NAND-NOR reconfigurable logic gates are implemented through bidirectional mechanical-electrical transduction in origami switching elements (Fig. 2g). Each logic function utilizes dedicated conductive pathways, with element A2 incorporating dual

Fig. 2 | Bit abstraction of the origami switching element and the construction and testing of a reconfigurable logic gate. **a** 0-input and 1-input of the origami unit-cell as a logic element. **b** The Buffer origami switching elements and logic function demonstrations. **c** The NOT origami switching elements along with their input and output demonstrations. **d** Construction and reconfigurable schematic of AND, OR, NOR, and NAND logic gates based on integrated circuit switch theory. An inverted input ($\bar{}$) corresponds to a NOT switching element, and an ordinary input corresponds to a Buffer switching element. **e** The physical images of four mechanical inputs and displacement-force curves of the origami unit-cell in series under compression. **f** The physical images of four mechanical inputs and

displacement-force curves of the origami unit-cell in parallel. **g** Designed AND, OR, NOR, and NAND reconfigurable logic gates using Buffer and NOT origami switching elements. Conductive networks 1, 3, and 4 are used for logical operations, while conductive networks 2 and 5 are used for indicating the folding states of the unit-cells, i.e., the inputs. **h** The physical demonstration of four possible input and output results for the designed AND-OR logic gate with the AND logic function. The blue or red light indicates that one of the two inputs is 1, while no light indicates an input of 0. The green light indicates that the logic gate output is 1, and no light indicates an output of 0. **i** The physical demonstration of four possible input and output results for the designed AND-OR logic gate with OR logic function.

copper-titanium-nickel alloy networks (networks 3 and 4, Supplementary Fig. 8). Series AND/NOR operations activate networks 1 and 4, while parallel OR/NAND configurations engage networks 1 and 3. Integrated indicator lights provide real-time input/output state visualization. The system employs electromagnetic actuation for force input generation and electrical signal output, enabling logic gate cascading in combinatorial circuits (see “Methods” section). Electrical inputs energize either a single set of helical coils for AND/NOR or parallel sets of helical coils for OR/NAND, generating magnetic fields that drive embedded N52 magnets to compress the origami elements. Precision voltage modulation from an external power supply facilitates automated cycling through four discrete mechanical inputs (Supplementary Movies 3 and 4). Modular design supports reconfigurable motion control and network integration. Comprehensive state mapping, including folding dynamics and I/O indicators for all input conditions, is presented in Fig. 2h, i and Supplementary Fig. 11, with experimental outputs demonstrating complete agreement with theoretical truth tables.

Reprogrammable gate meta-array

The reprogrammable gate meta-array for integrated circuit operations is constructed through hierarchical cascading of reconfigurable logic gates, enabled by canonical Boolean algebra and optimized network topologies.

This architecture supports dynamic combinational logic reconfiguration through functional metamorphosis of constituent gates within the meta-array (Fig. 3), implemented via the following methodology:

- (1) **Network Optimization:** Combinational logic relationships are first expressed in canonical sum-of-products form, followed by Boolean minimization to reduce gate count and optimize cascading network complexity.
- (2) **Reconfigurable Architecture Design:** The meta-array implements Boolean operations through universal AND-OR gate primitives, with product terms, sum terms, and inversions mapped to AND, OR, and NOT-Wire gate configurations, respectively (Supplementary Fig. 12). Functional reconfiguration is achieved through AND \leftrightarrow OR gate transformation and selective gate activation/deactivation, leveraging shared network resources across different logic functions.
- (3) **Mechanical-Electrical Signal Propagation:** Signal cascading is implemented through electromechanical transduction, where each stage’s origami switching element serves as the control mechanism for the subsequent stage’s electromagnetic actuator, enabling seamless signal transmission across the logic hierarchy.

We showcase the reconfiguration of logic functions within a reprogrammable gate meta-array through four combinational logic operations: 2-bit adder, subtractor, comparator, and multiplier. These operations are facilitated by the input-output relationships of abcd-e, abcd-f, abcd-g, and ac-h networks, incorporating NOT-Wire logic gates and multi-level AND-OR logic gates (I, II, III, IV) as detailed in Supplementary Figs. 14 and 15. NOT-Wire gates 1–4 consistently apply NOT operations to inputs a, b, c, and d, whereas gates 5–8

toggle inputs a and c between their original and negated states. The meta-array’s logic gates are tested using platforms that integrate electromagnetic actuation with origami logic circuits (Supplementary Fig. 14). These circuits, though functionally distinct, interact mechanically: the electromagnetic circuit transforms electrical signals into electromagnetic force, which the origami circuit then utilizes to execute mechanical logic operations, ultimately generating electrical outputs.

The logic operation network of the reprogrammable gate meta-array is structured into four hierarchical control circuits (Supplementary Fig. 15c). The first level integrates a Vcc switch, an input switch, and the actuating coils of NOT-Wire gates in series, which supply the electromagnetic force required for NOT gates to perform logical negation on specific input signals. The second level circuit connects a Vcc switch, an input switch, or a Vcc switch with the origami switching elements of NOT-Wire gates and the actuating coils of Level I AND-OR gates, enabling the provision of electromagnetic force to drive Level I AND-OR logic operations. The third level circuit includes a Vcc switch, the origami switching elements of x -level ($x \geq 1$) AND-OR gates, and the actuating coils for $x+1$ level AND-OR gates, facilitating the control of electromagnetic-mechanical inputs to the $x+1$ level based on the results of x -level logic operations. The fourth level circuit comprises a Vcc switch, the origami switching elements of x_{\max} -level AND-OR gates, and a collecting resistor, which collectively manage the final computational output of the meta-array. This approach leverages electromagnetic force generated by actuation coils to drive origami switching elements, enabling seamless cascading between logic gates within the reprogrammable logic meta-array.

When AND-OR logic gates 15, 21, and 27 are excluded, the reprogrammable logic meta-array executes binary logic operations for a 2-bit adder via the abcd-e, abcd-f, and abcd-g input-output cascade networks (Fig. 3a and Supplementary Movie 5). The cascade network for the 2-bit subtractor differs from the adder only in the participation of NOT operations on inputs to AND-OR logic gates 11, 12, 24, 25, and 26 (Fig. 3b and Supplementary Movie 6). This transformation is achieved by reconfiguring NOT-Wire gates 5 and 7 from Wire to NOT functions, while NOT-Wire gate 6 switches from NOT to Wire, enabling the transition from a 2-bit adder to a 2-bit subtractor. The 2-bit comparator’s cascade network incorporates a subset of the abcd-e network and the entirety of the abcd-f network (Fig. 3c and Supplementary Movie 7). Logical functions of NOT-Wire gates 5, 6, and 7, along with AND-OR gates 9, 10, 12, 16, 18, and 19, are reconfigured, while AND-OR gates 13, 14, 20, 22, 23, and 28 are disconnected, facilitating the shift from a 2-bit adder to a 2-bit comparator. In this configuration, only the red origami switching element and its actuating coil in logic gate 19 remain active, transferring the output of logic gate 12 to logic gate 18 as input. The 2-bit multiplier introduces an additional ac-h cascade network alongside the existing three networks (Fig. 3d and Supplementary Movie 8). NOT-Wire gate 8 and AND-OR gates 14, 26, and 28 undergo functional conversion, while logic gates 10, 13, 22, 23, and 24 are disconnected, and logic gates 15, 21, and 27 are added, reconfiguring the meta-array from a 2-bit adder to a 2-bit multiplier. Logic gates 16, 20, and 29 retain only one origami switching element and actuating coil for signal transmission.

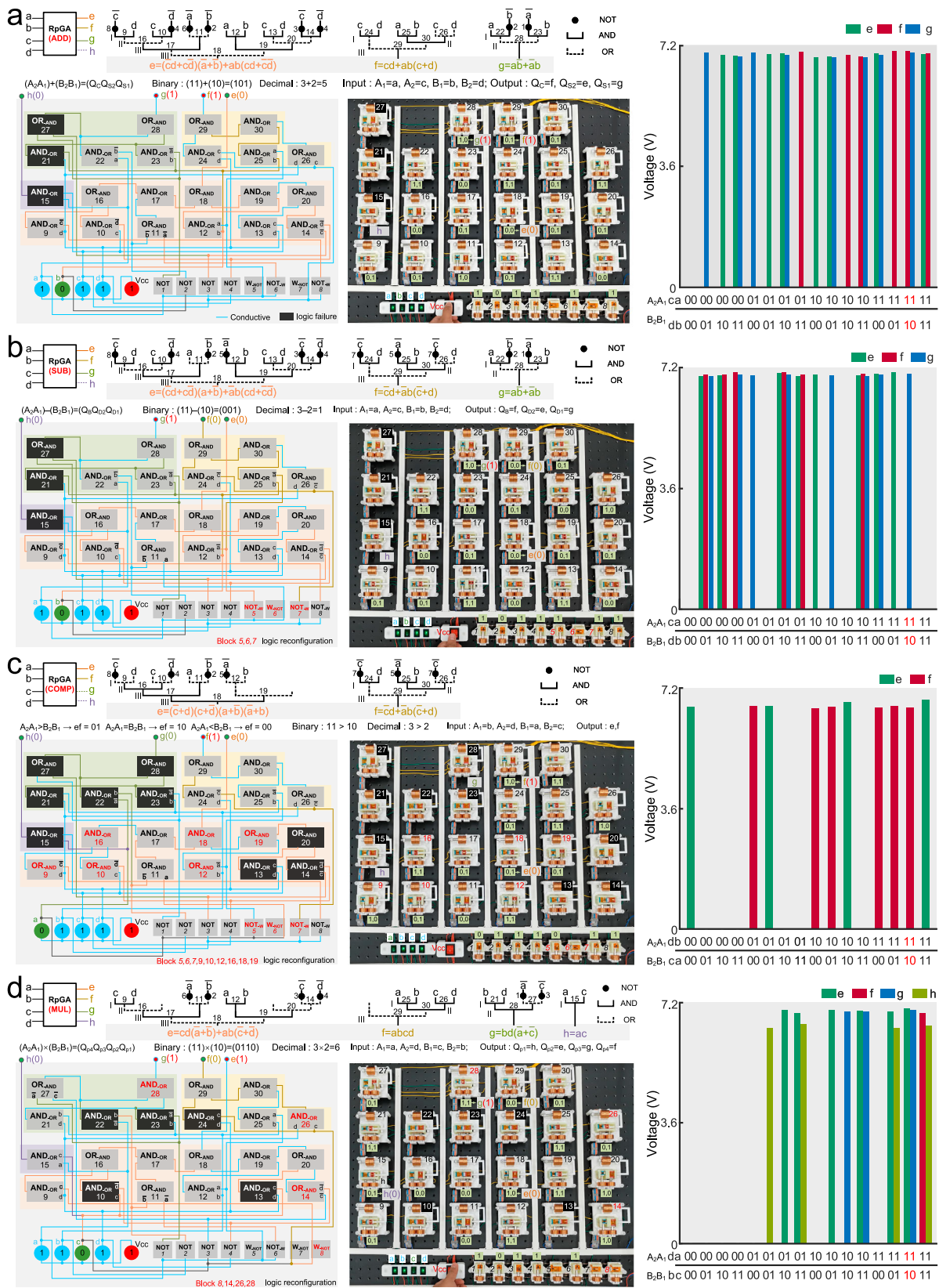


Figure 3 presents experimental results of the reprogrammable logic meta-array for four distinct functions, with inputs $A = 11$ (decimal 3) and $B = 10$ (decimal 2). For the 2-bit adder, inputs a, c , and d are set to 1, while b is 0. NOT-Wire gate 2 and AND-OR gates 22, 28, followed by AND-OR gates 24 and 29, are sequentially activated, producing voltage signals at outputs g and f , yielding the result 101

(decimal 5). For the 2-bit subtractor, inputs a, c , and d remain 1, with b at 0. NOT-Wire gate 2 and AND-OR gates 22 and 28 are activated in sequence, generating a voltage signal at output g , resulting in 001 (decimal 1). For the 2-bit comparator, inputs b, c , and d are 1, while a is 0. NOT-Wire gate 5 and AND-OR gates 25, 26, 30, and 29 are sequentially activated, producing a voltage signal at output f , indicating 01

Fig. 3 | Construction and multitask functionality of a reprogrammable gate meta-array based on reconfigurable AND-OR and NOT-Wire logic gates.

a Cascading network diagram of the reprogrammable gate meta-array as a 2-bit adder, along with a demonstration of the example of $3 + 2 = 5$ and the graph of output electrical signal results for 16 possible inputs. The reprogrammable gate meta-array is constructed by four inputs (a, b, c, d), four outputs (e, f, g, h), 8 NOT-Wire gates (1–8), and 22 AND-OR gates (14–30). Vcc represents high potential input. Orange-red, yellow, green, and purple regions represent abcd-e, abcd-f, abcd-g, and ac-h cascading networks, respectively. Simplified Boolean expressions and cascading network relationships are shown at the bottom. Solid lines represent AND, dashed lines represent OR, and overlines represent NOT operations. Roman numerals I-IV denote logical network levels. The formulas for the 2-bit adder and

corresponding input-output relations for $3 + 2$ in binary and decimal are also shown. Connected paths that result in an output of 1 are highlighted in Cyan. Green-numbered results show the inputs of NOT-Wire and AND-OR logic gate.

b Cascading network diagram of the reprogrammable gate meta-array as a 2-bit subtractor, along with a demonstration of the example of $3 - 2 = 1$ and the graph of output electrical signal results. c Cascading network diagram of the reprogrammable gate meta-array as a 2-bit comparator, along with a demonstration of the example of $3 > 2$ and the graph of output electrical signal results. d Cascading network diagram of the reprogrammable gate meta-array as a 2-bit comparator, along with a demonstration of the example of $3 \times 2 = 6$ and the graph of output electrical signal results.

($A > B$). For the 2-bit multiplier, inputs a, b, and d are 1, with c at 0. NOT-Wire gate 3 and AND-OR gates 14, 19, 20, and 18, followed by NOT-Wire gate 3 and AND-OR gates 21, 27, and 28, are sequentially activated, generating voltage signals at outputs e and g, resulting in 0110 (decimal 6). Comprehensive testing of all 16 possible input combinations for the 2-bit adder, subtractor, comparator, and multiplier confirms that the outputs align with expected decimal operations (Supplementary Figs. 16–31).

The reprogrammable logic meta-array, comprising 30 NOT-Wire and AND-OR logic gates, is capable of executing 2-bit adder, subtractor, comparator, and multiplier operations at least. Our work achieves a 20% reduction in gate count compared to a non-reconfigurable circuit utilizing 36 AND, OR, NOT, and XOR gates. Furthermore, when benchmarked against FPGA/DPGA architectures requiring 44 AND, OR, and NOT gates for equivalent functionality (including the aforementioned 2-bit operations implemented through III level abcd-e, III-level abcd-f, II level abcd-g, and I-level ac-h configurations), this work demonstrates a 46.7% reduction in logic gate utilization.

3D reprogrammable logic cube

The reprogrammable gate meta-array, leveraging its high logic integration, is well-suited for large-scale computing applications. It enables combinational logic function conversion by reconfiguring two Buffer origami switching elements within the logic gates. However, the origami switching elements lack inter-gate combinability, and each logic gate requires individual control of control nodes, resulting in a complex reconfiguration process due to the absence of coordinated control mechanisms. In intelligent systems such as robotics, logic operations are often simple but numerous, distributed, and require parallel processing across diverse tasks. This demands higher-dimensional integration of logic modules to conserve space while preserving functionality, as well as simplification of control nodes in reconfigurable logic modules. Inspired by the Rubik's Cube, which exhibits diverse reconfigurability through 3D block distribution and inter-block reconfiguration enabled by just three control nodes (Supplementary Fig. 32). By integrating the reconfigurable principles of the Rubik's Cube with Buffer/NOT origami switching elements, we develop a reprogrammable logic cube capable of multifunctional logic computing (Fig. 4). The implementation steps are as follows:

- (1) 3D Distribution of Origami Switching Elements: The input-output relationships in sum-of-product Boolean functions consist of product terms connected by a sum term. Each input within a product term corresponds to a switching element: inputs not involved in a NOT operation are mapped to Buffer origami switching elements, while those requiring inversion are mapped to NOT origami switching elements. These origami switching elements are distributed in a serial configuration within each column of the reprogrammable logic cube, with identical inputs positioned in the same layer. The outputs of the

serially connected origami switching elements are then parallelly connected to implement the sum operation.

- (2) Reconfigurability of the Reprogrammable Logic Cube: The origami switching elements in the designed reprogrammable logic cube can be reconfigured through relative rotation between layers and positional changes within layers using the Permutation of Last Layer (PLL) method. By serially connecting origami switching elements across multiple layers, more flexible combinational logic computations are enabled. The reconfiguration process is achieved through coordinated control of rotations along the three degrees of freedom (x, y, and z).

The second-order reprogrammable logic cube enables logic transformation by allowing two corner blocks within each layer to exchange positions via the PLL method, while leaving other blocks unchanged. For instance, swapping N2 and B3 in layer b and reconfiguring with B6 and N7 in layer a modifies the combinational logic outcomes of the two serially connected origami switching elements (Fig. 4a). Through multiple corner block transformations, any two origami switching elements within layer b can be exchanged. With two types of origami switching elements (Buffer/NOT) positioned at four corners, and considering that these arrangements are mutually convertible, the number of possible configurations is given by C_4^2 (Supplementary Discussion 3). This implies that each layer of the second-order reprogrammable logic cube can achieve C_4^2 distinct combinational logic functions.

The third-order reprogrammable logic cube enables logical reconfiguration by repositioning origami switching elements at three corner blocks within each layer via the Ua/Ub PLL method. For instance, swapping N1, N3, and B5 in layer b and reconfiguring with B9, B11, and N13 in layer a modifies the combinational logic outputs (Fig. 4b). Similarly, origami switching elements at three edge blocks can be reconfigured using the Aa/Ab PLL method—e.g., exchanging N2, N4, and B8 in layer b and reconfiguring with B10, B12, and N16 in layer a alters the logic outcomes. Repeated corner or edge block transformations allow origami switching elements to occupy any corner or edge positions, respectively, with no mutual interference between corner and edge transformations. Given the placement of two types (Buffer/NOT) of origami switching elements across four distinct corner or edge positions, and accounting for their mutual convertibility, the number of possible configurations is C_4^2 . Consequently, each layer of the third-order reprogrammable logic cube can implement $C_4^2 C_4^2$ distinct combinational logic functions.

As the order increases to the 4th, the reprogrammable logic cube retains the corner block transformation mechanism of the third order. For example, exchanging positions 1, 4, and 7 in layer b and reconfiguring with positions 13, 16, and 19 in layer a (Fig. 4c). In the fourth-order cube, each edge block comprises two origami switching elements, forming four possible combinations: B-B, B-N, N-B, and N-N. Three combined edge blocks can be reconfigured using the Aa/Ab PLL method to achieve logical reconfiguration. For instance, swapping B2-

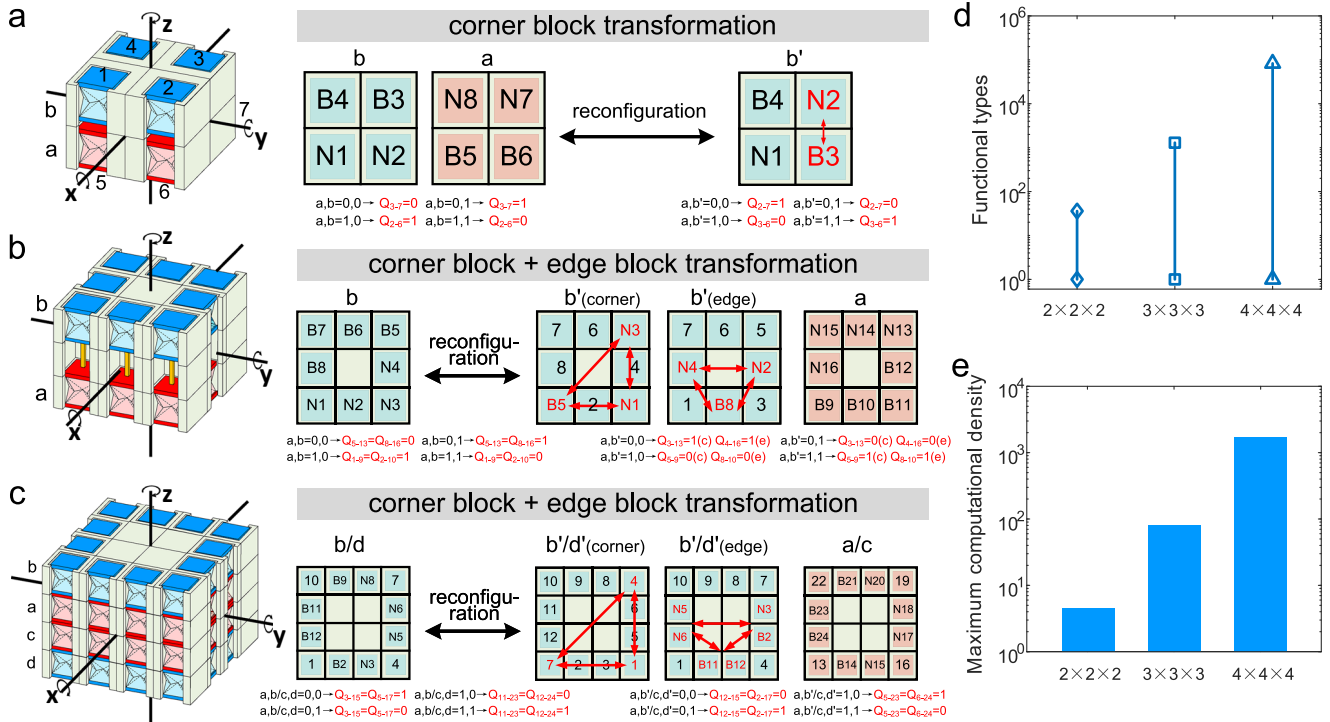


Fig. 4 | Schematic diagrams of the reconfigurable logic transformation methods of second-order, third-order, and fourth-order reprogrammable logic cubes. **a** Schematic diagram of the second-order reprogrammable logic cube and the logic reconfiguration under the corner block transformation method. **b** Schematic diagram of third-order reprogrammable logic cube and the logic reconfiguration under corner block and edge block transformation methods. **c** Schematic diagram of a fourth-order reprogrammable logic cube and the logic

reconfiguration under corner block and edge block transformation methods. **d** Theoretical data graph showing the number range of combinational logic functions achievable by second-order, third-order, and fourth-order reprogrammable logic cubes. **e** Comparison graph of the maximum computational power density achievable by second-order, third-order, and fourth-order reprogrammable logic cubes.

N3, N5-N6, and B11-B12 in layer b and reconfiguring with B14-N15, N17-N18, and B23-B24 in layer a modifies the combinational logic outputs. Given the placement of four distinct combinations of origami switching elements across four edge positions, and accounting for their mutual convertibility, the number of possible configurations is $C_4^1 C_3^1 C_2^1 / 2$. Layer a, as the middle layer, cannot internally exchange origami switching elements but can rotate as a whole around the z-axis and recombine with layer b. Consequently, $4C_4^2 C_4^1 C_3^1 C_2^1 / 2$ reconfigurable logic functions can be realized for layers a and b. Due to the symmetry between layers a, b and layers c, d, their reconfigurability is identical. The maximum functional types for second, third, and fourth-order reprogrammable logic cubes are 36 ($C_4^2 C_4^2$), 1296 ($C_4^2 C_4^2 C_4^2 C_4^2$), and 82944 ($16C_4^2 C_4^1 C_3^1 C_2^1 C_4^2 C_4^1 C_3^1 C_2^1 / 4$), respectively. Correspondingly, their computational density can reach 4.5, 81, and 1728 (Fig. 4d, e).

To validate the logic-processing capability and reprogrammability described above, we fabricated a third-order reprogrammable logic cube prototype incorporating sixteen Buffer/NOT origami switching elements. This cube consists of 3D-printed face-center, edge, and corner blocks—each housing Buffer/NOT switching elements (Fig. 5). All face-center blocks are fixed to a central steel sphere via bearings; edge blocks attach magnetically to both the face-center blocks and the sphere; corner blocks adhere magnetically to the edge blocks (Supplementary Fig. 33f–g). Because each face’s edge and corner blocks rotate in concert with their face-center block, the reprogrammable logic cube exhibits three-dimensional axial reconfiguration of Rubik-cube. Multiple-layer rotations reconfigure its corner/edge blocks (Supplementary Movie 9). When compressive force is applied from input a, the cube successively realizes inputs (0, 0), (1, 0), and (1, 1) (Supplementary Figs. 34–37); force from input b yields the input (0, 1). The upper conductive network is connected to Vcc in parallel, and

each column’s bottom side outputs a voltage signal. Following the initial arrangement illustrated in Fig. 4b, we experimentally test the cube before reconfiguration, after corner block transformation, and after edge block transformation. In all four input combinations, the measured outputs matched theoretical predictions exactly.

Based on the above arrangement, the third-order reprogrammable logic cube removes the NOT13 and NOT16 origami switching elements of the input layer. Additionally, the lower side of conductor networks of the second column (series-connected NOT3 and Buffer11 in corner blocks) and sixth column (series-connected Buffer7 and NOT15) are parallelized to form the output Q'_{diff} . The lower side of conductor networks of the third column (series-connected NOT4 and Buffer12 in edge blocks) and fifth column (series-connected Buffer6 and NOT14) are parallelized to form the output Q_{diff} . Consequently, the half-subtractor can be realized with the output of the first column (series-connected NOT2 and Buffer10 in edge blocks) and output Q'_{diff} . Likewise, another half-subtractor is achieved with the output of the eighth column (series-connected NOT11 and Buffer9 in edge blocks) and output Q_{diff} . Cascading these two half-subtractor stages produces the full-subtractor function (Fig. 5a): The lower side of the conductor networks of the first column and eighth column are parallelized to form the borrow-out output Q_{cin} . While the Q_{diff} serves as the difference output. The minuend and subtrahend inputs are input a_c and b_c of the half-subtractor in the corner blocks, and the borrow-in input C_{in} is input a_e of the half-subtractor in the edge blocks. While the logic operation, the input b_e of the half-subtractor in the edge blocks is determined by output Q'_{diff} . Thus, a reprogrammable logic cube requires two sequential outputs to achieve full subtraction. For example, for inputs (a_c, b_c, a_e) = (0, 1, 0), the half-subtractor in corner blocks processes (0, 1) to yield $Q'_{diff} = 1$, then the

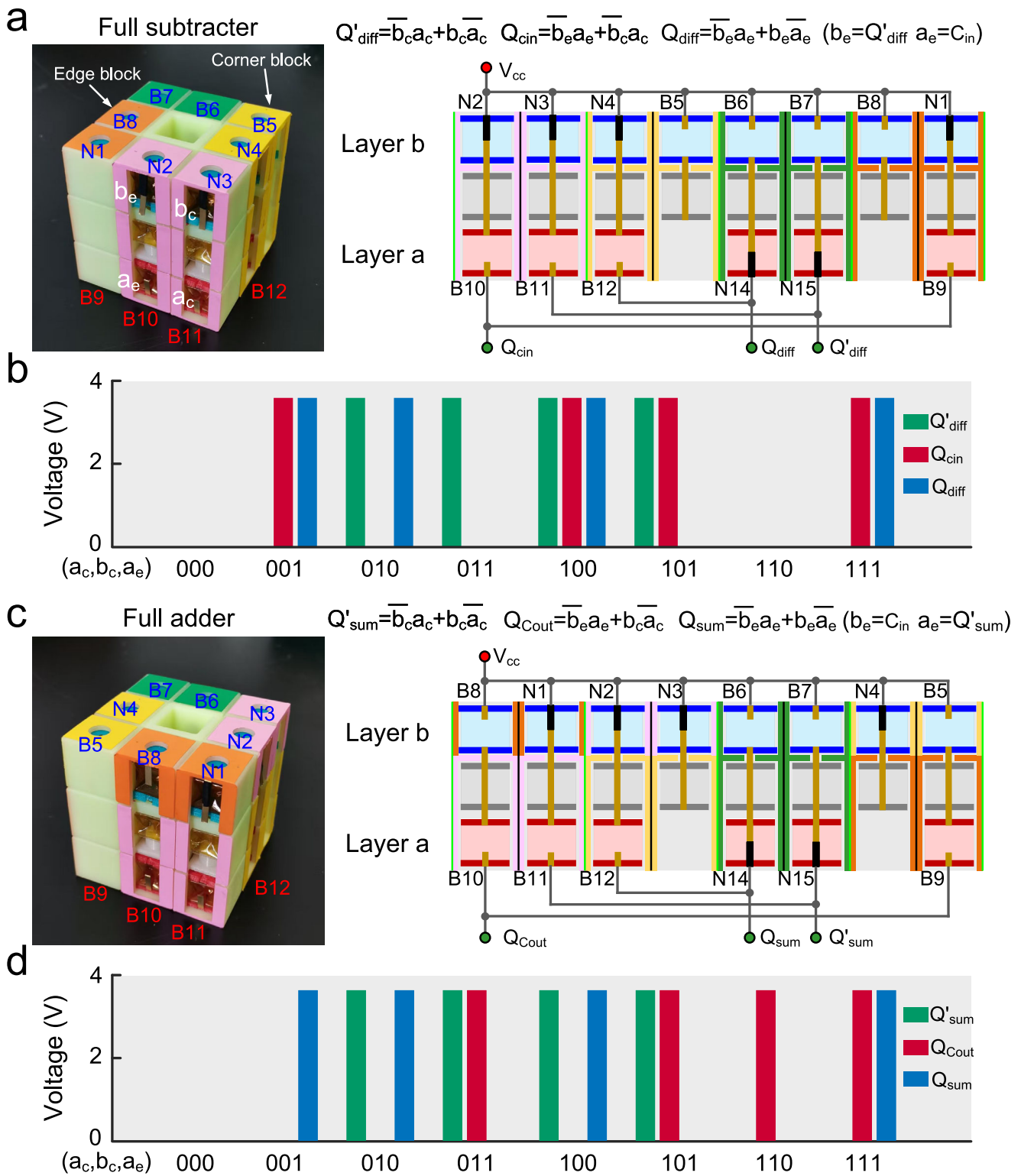


Fig. 5 | Design of a third-order reprogrammable logic cube with the logic functions of full subtractor/full adder and logic reconfiguration. **a** The Physical diagram, wiring network diagram, and logic expression of the third-order reprogrammable logic cube as a full subtractor. **b** The graph of the output electrical signal results of the third-order reprogrammable logic cube as a full subtractor. **c** The Physical diagram, wiring network diagram, and logic expression of the third-order reprogrammable logic cube as a full adder. **d** The graph of the output electrical signal results of the third-order reprogrammable logic cube as a full adder.

half-subtractor in edge blocks processes (0, 1) to co-produce final outputs Q_{cin} and Q_{diff} (Supplementary Movie 10). Experimental results for all eight input combinations agree with the full-subtractor truth table (Fig. 5b; Supplementary Figs. 38 and 39). Upon once corner block transformation and once edge block transformation, the

reprogrammable logic cube reconfigures from full-subtractor to full-adder functionality (Supplementary Movie 9). The parallel combination of Buffer8-Buffer10 in corner blocks and Buffer5-Buffer9 in edge blocks generates Q_{Cout} (Fig. 5c). The parallel combination of NOT3-Buffer11 and Buffer7-NOT15 in corner blocks generates Q'_{sum} .

The parallel combination of NOT2-Buffer12 and Buffer6-NOT14 in corner blocks generates Q_{sum} . For the full adder, the two-bit addend inputs are inputs a_c and b_c of the half-adder input in the corner blocks, respectively. The low-order carry input C_{in} is the edge of the input b_e of the half-adder input in edge blocks, and the input a_e is determined by the output Q'_{sum} . Measured outputs for all input combinations match the full-adder truth table (Fig. 5d; Supplementary Figs. 40 and 41).

The designed third-order reprogrammable logic cube implements both full-subtractor and full-adder logic functions using only fourteen Buffer/NOT origami switching elements. In comparison, the mechanically integrated circuit metamaterial proposed by Helou et al.—based on soft matter and synthesized via the Quine–McCluskey method—requires 18 switching elements to realize a single full-adder²⁰. Considering full-adder performance alone, our work achieves a 29% increase in computational density. The mechanical system designed by Zhang et al., which employs 15 convex–concave modules within a rigid frame and conductive material, also implements a full adder²⁷; our work offers a 7% higher computational density. The reprogrammable mechanical metamaterial proposed by Mei et al., constructed from curved-beam unit-cells, uses 21 elements to realize only a half-adder⁸; relative to this, our work on a half-adder achieves a 200% increase in computational density. The reprogrammable Metamaterial Processors for soft robotics, designed by Jiao et al.—based on the Laws of Physical Boolean Algebra—use 18 pneumatic unit-cells to reconfigure between full-adder and full-subtractor functions²⁴; our work attains a 29% higher mechanical-computing density for these combined reconfigurable operations.

Embodied autonomous robotic system

The reprogrammable logic cube, leveraging the flexible reconfigurability and coordinated control of its origami switching elements, serves as a decision-making module integrated with perception and actuation systems to enable embodied robotic intelligence for environmental interaction in path-planning applications, including boundary recognition (right-angled vs. curved paths) and adaptive motion control (Fig. 6). The inputs (a, b) of the logic cube are determined by the perception module, which employs photoelectric distance sensors a, b and c (Fig. 6a): front sensors b and c measure the distance between the robot and path boundaries, while side sensor a monitors lateral distances. Analog signals ($D \geq \text{threshold } D_0 \rightarrow \text{logic } 0$; $D < D_0 \rightarrow \text{logic } 1$) are digitized via an analog-to-digital converter. Upon reaching a path boundary, the logic cube—configured for half-adder or half-subtractor functions—executes one mechanical compression to process inputs (Sensor b, Sensor c). Right-angled boundaries are identified when the half-subtractor output $d = 0$ triggers a transition to half-adder logic or when the half-adder maintains current logic. Curved boundaries are recognized if the half-subtractor output $d = 1$ retain logic, or half-adder switches to half-subtractor logic. Following boundary recognition, one compression process inputs (Sensor a, Sensor b) to generate motion commands. For right-angled paths, output (0,1) reverses the left wheel and advances the right wheel for in-place left turns, while output (1,0) reverses the right wheel and advances the left wheel for in-place right turns. For curved paths, two-bit binary instructions modulate wheel rotational speeds: outputs (0,0) prioritize the left wheel for rightward arc turns, whereas (1,1) prioritize the right wheel for leftward arc turns.

A second-order reprogrammable logic cube with 2^3 -1 Buffer/NOT elements implements these functions through an origami switching array enclosed in upper/lower casings with copper tape and external wiring for I/O connectivity (Fig. 6b). The casings rotate via a central axis to reconfigure logic between half-adder and half-subtractor modes. The upper casing houses two Buffer and two NOT elements, while the lower casing contains two Buffer and one NOT element. The half-subtractor (Fig. 6c) comprises two-bit input (a, b) and two-bit output (Q_{cin} , Q_{diff}). Sequential compressive force applied to the bottom surface enables the second-order reprogrammable logic cube to

execute mechanical inputs (0,0), (1,0), and (1,1), while top-surface compression activates the input (0,1) (Supplementary Fig. 44). The upper side of the conductive network is connected to V_{cc} . The output Q_{cin} is derived from the output of the first column's series-connected NOT1 and Buffer5. The lower side of conductor networks of the second column (series-connected NOT2 and Buffer6) and fourth column (series-connected Buffer4 and NOT7) are parallelized to form the output Q_{diff} . Experimental validation across all four input combinations confirms alignment with the half-subtractor truth table. Buffer3 occupies the third column to accommodate input b for half-adder functionality. A 90° rotation of the half-subtractor's upper and lower layers reconfigures Buffer4 and Buffer5 into a series connection for Q_{cout} , while the parallel combination of NOT1-Buffer6 and Buffer3-NOT7 generates Q_{sum} , thereby converting the logic function from a half-subtractor to a half-adder (Fig. 6d). Subsequent testing of all input permutations (Supplementary Fig. 45) exhibits full consistency with the half-adder truth table.

The analog signals from the perception system of the robot prototype (Fig. 6e) are digitized to determine mechanical inputs and reconfigurability of the reprogrammable logic cube. The decision module comprises: (1) the reprogrammable logic cube, (2) two thrust servo mechanisms [the first applies input (0,1); the second and third apply inputs (0,0), (1,0), and (1,1)], and (3) a rotating mechanism driving 90° clockwise/counterclockwise rotations of the cube's upper layer, enabling dynamic switching between half-adder and half-subtractor functions. Digital action commands generated by the logic cube are transmitted to the actuation module, either directly or as converted analog signals, to regulate driving wheels. Additional fabrication details are provided in the “Methods” section. The robot autonomously executed two distinct left-turn maneuvers in mixed right-angled and curved boundary paths (Fig. 6f, Supplementary Movie 11), validating the system's adaptive path-planning capabilities through coordinated perception-logic-actuation workflows.

Discussion

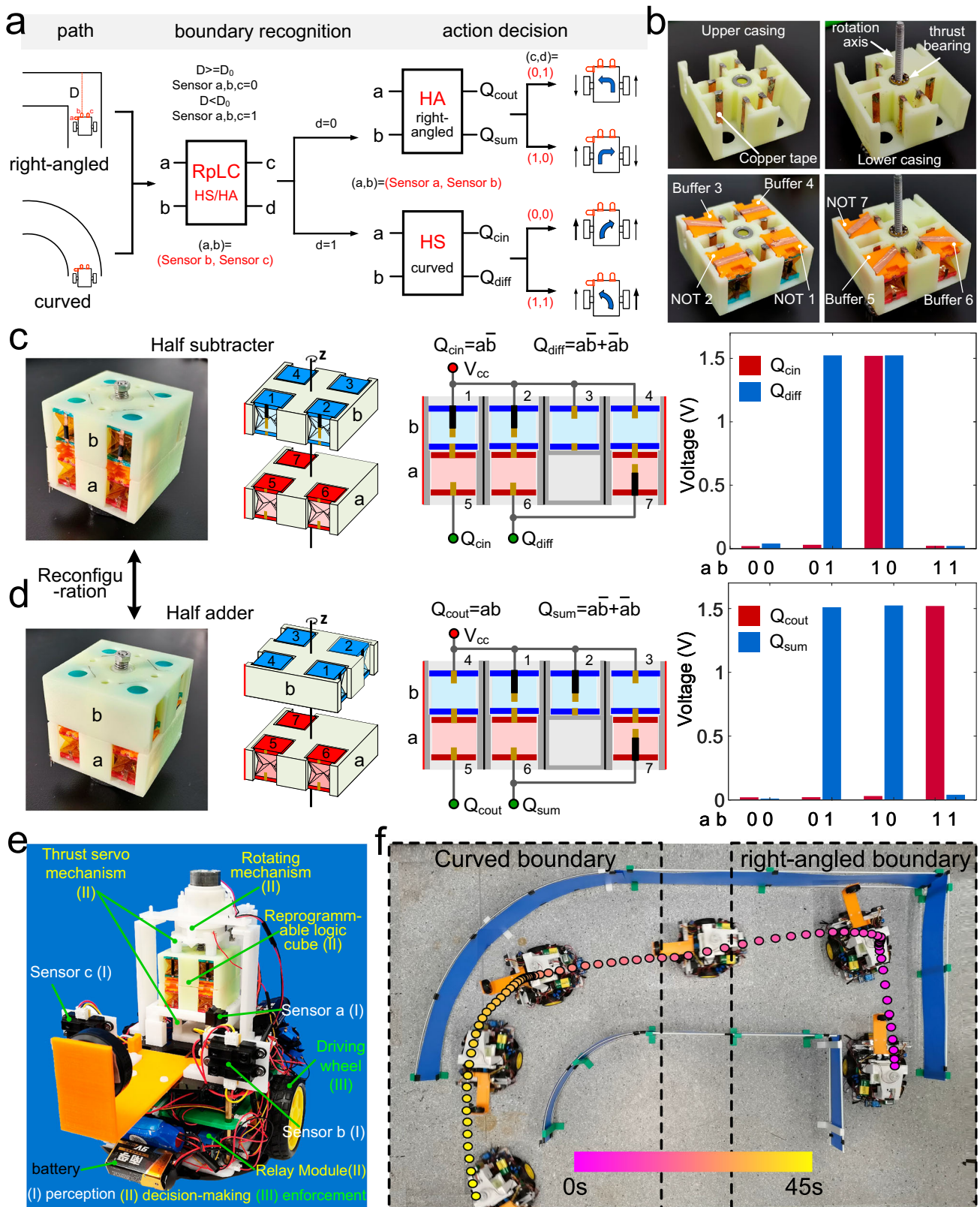
The reprogrammable logic metamaterials introduced here bridge a critical gap in mechanical computing: the inability to dynamically adapt logic functions post-fabrication. By encoding Boolean functions into reconfigurable origami networks, our 2D meta-array and 3D logic cube systems achieve on-the-fly functional transformation—a capability previously restricted to electronic systems. This architectural shift enables 46.7% fewer logic gates than FPGA/DPGA architectures and computational density of 1728 in a fourth-order 3D meta-cube—a trade-off between footprint and functional scalability. When integrated with perception and enforcement modules, these second-order reprogrammable logic metamaterials enable sophisticated decision-making processes, such as path planning, demonstrating their practical applicability in embodied intelligent environmental interaction.

The Boolean function-based design methodology is not constrained by specific structures or materials, making it universally applicable to a wide range of combinational logic functions and their reconfigurable designs. This universality strengthens the interoperability between different mechanical computing systems and significantly enhances the interaction capabilities of hybrid system modules¹³. This work provides design methodologies for the integration of computational intelligence into physical systems and unlocking opportunities for informational materials^{28,29}, embodied intelligent^{30,31}, and autonomous robotics^{2,32}.

Methods

Fabrication of origami unit-cell and origami switching elements

The origami switching elements are based on origami unit-cells (Fig. 1a). The crease pattern of the origami unit-cell (Supplementary Fig. 3a) can be described by the plane angles α , β , and the edge length l . The lengths of the four creases in the pattern are related³². The origami



unit-cells used for reconfigurable logic gates are fabricated with PET film. The surface density of the PET film is 1.38 g cm^{-3} , and its tensile strength is approximately 200 MPa. To facilitate folding, the crease pattern with α , β , and l set to 45° , 90° , and 15 mm, respectively, is engraved on one side of the PET film using a cutting plotter (CX720, Shenzhen Yitu Electromechanics Co., Ltd) equipped with an Indentation knife, creating unpenetrated creases (Supplementary Fig. 3b–d).

The processed origami unit-cell is in the form of a cube, with 1-mm-thick resin plates, 3D-printed using FDM technology, attached to the top and bottom surfaces to assist with force-induced folding (Supplementary Fig. 3e).

The origami unit-cells are combined with 0.3-mm-thick, 2-mm-wide copper sheets and 0.01-mm-thick titanium-nickel alloy wires to form an origami switching element with computational capabilities

Fig. 6 | Reprogrammable logic cube with half-subtractor and half-adder functions as the decision-making module for robotic embodied intelligent environment-interactive path-planning applications. **a** Path-planning logic design for right-angled and curved boundary paths achieved through the reconfiguration of half-subtractor and half-adder logic operations. **b** Physical images of the upper and lower casings of the second-order reprogrammable logic cube used for path planning, and the arrangement of Buffer/NOT origami switching elements within the casings. **c** Physical image, decomposition 3D schematics, wiring

diagrams, and experimental output results for 4 possible inputs of a second-order reprogrammable logic cube as a half-subtractor logic function. **d** Physical image, decomposition 3D schematics, wiring diagrams, and experimental output results of a second-order reprogrammable logic cube as a half-adder logic function. **e** Physical image of the embodied intelligent environment-interactive robot and component description. **f** Overlaid sequential images visualize the trajectories of the robot when navigating paths with right-angled and curved boundaries. The time intervals between adjacent points on the trajectory are equal.

(Supplementary Fig. 4a). The copper sheets and titanium-nickel alloy wires are placed on the upper and lower resin plates of the origami unit-cell, respectively. To ensure secure attachment of the conductive network components to the top and bottom surfaces, the resin plates on the origami unit-cells are divided into two 0.5-mm-thick pieces. Each separated resin plate has a 0.2-mm-wide groove for placing the titanium-nickel alloy wire or a 2-mm-wide slot for placing the copper sheet. The other separated resin piece is tightly fastened together with the resin plate holding the conductive network using M1 self-tapping screws. The copper sheets and titanium-nickel alloy wires are connected to 0.1 mm copper wires for demonstration of the Buffer and NOT origami switching elements' logic functions in series with the power supply and indicator lights (Supplementary Fig. 4b).

Mechanical testing of series/parallel origami unit-cells

Since AND, OR, NAND, and NOR logic gates are composed of two origami switching elements with casings, these switching elements realize the inputs (0,1), (1,0), and (1,1) through their deformation difference. Therefore, two origami unit-cells with thicknesses of 0.038 mm and 0.05 mm are used in the fabrication of the logic gates. For AND and NOR logic gates, the origami unit-cells are arranged in series. The mechanical performance of the series origami unit-cells is tested using an electronic universal testing machine (HY-0580, Shanghai Hengyi Testing Machine Co., Ltd.). During the testing, the series of origami unit-cells is vertically mounted between custom-designed clamps (Supplementary Fig. 7a). To prevent deformation of the series of origami unit-cells in the horizontal direction, an iron bar is placed between them to limit their displacement. The iron bar is relatively thin and has a smooth surface, so its effect on the folding mechanical performance testing is negligible. The upper clamp applies compressive force at a speed of 5 mm/min to induce quasi-static deformation of the series origami unit-cells. The reaction force is measured using a 1000 N force sensor (BSS-100 kg, Transcell Technology, Inc.) at a sampling frequency of 20 Hz. The errors of the force sensor and displacement sensor are less than 0.03% and 0.5% of the actual output, respectively.

For OR and NAND logic gates, the origami unit-cells are arranged in parallel. To test the folding states of the parallel origami unit-cells under the same applied force, each origami unit-cell is connected in series with a spring (Supplementary Fig. 7b). It is important to note that when two origami unit-cells are in a flat-foldable state, the spring cannot be fully compressed. The testing conditions for the parallel origami unit-cells are the same as those for the series origami unit-cells. Data obtained from the electronic universal testing machine shows that the displacement is the sum of the deformation displacements of the origami unit-cell and the spring. Therefore, experimental data relating the deformation displacement of each origami unit-cell and the compressive force were obtained using Microsoft Visio image analysis. The height of each origami unit-cell and the combined height of the origami unit-cell with spring in series are measured by the length of marked lines (Supplementary Fig. 7c). The actual deformation displacement is calculated by multiplying the corresponding data obtained from image analysis by a scale factor.

Fabrication of AND-OR, NOR-NAND Logic Gates

The AND-OR logic gate is composed of two Buffer origami switching elements with casings. Wire networks are arranged on each casing, and those wire networks, along with the copper sheets and titanium-nickel alloy wires on the origami switching elements, form a conductive network that enables the reconfigurability of the logic gate's functionality. In addition to the conductive network for logical operations, a conductive network is also designed and fabricated to display the input status of each origami switching element. Therefore, each origami switching element is equipped with multiple sets of conductive networks made of copper sheets and titanium-nickel alloy wires. For instance, in the AND-OR logic gate, the conductive network 1 on the red origami switching element is used for AND and OR logic operations, while the conductive network 2 displays the input status of the red origami switching element (Supplementary Fig. 8a, b). Since the AND and OR logic operations in the AND-OR logic gate are realized by two separate circuits, the blue origami switching element is equipped with conductive networks 3, 4, and 5 (Supplementary Fig. 8c, d). Conductive network 3 is used for OR logic operations, conductive network 4 is used for AND logic operations, and conductive network 5 displays the input status of the blue origami switching element. When the origami switching elements are connected in series, the two conductive networks involved in the AND logic operation become interconnected due to the contact of the copper foil set in the middle connection. The NAND-NOR logic gate consists of two NOT origami switching elements with casings. The arrangement of the origami switching elements is similar to that of the AND-OR logic gate. In this case, the conductive networks 1 on the red origami switching element are used for NAND and NOR logic operations, while network 2 is used to display the input status of the red origami switching element. The blue origami switching element is equipped with conductive networks 3, 4, and 5, which are used for NAND logic operations, NOR logic operations, and input status display of the blue origami switching element, respectively.

The casings of the AND-OR and NAND-NOR logic gates are identical. Both casings are fabricated from transparent resin material using light-cured 3D printing (Supplementary Fig. 9a and b). The casing is a cubic shape with dimensions of 22 mm × 22 mm × 22 mm. The wall thickness is 2.4 mm, and the top and bottom surfaces are open to facilitate the placement of the origami switching elements and mechanical input loading. The two casings are connected by hinges, formed by two perforated cylinders, designed at the joint allowing the casings to rotate relative to each other for the reconfiguration of the logic gate. At four corners inside the casing, cylindrical passages are set, and the circular protrusions at the four corners of the resin plates of the origami switching elements are designed to fit them to stabilize the folding of the origami unit-cell. LED lights are placed on the front side of the casing to indicate the folding state of the origami switching elements. Internal grooves on the back side of the casing are provided to affix conductive copper tape. The titanium-nickel alloy wire behind the origami switching element sweeps the conductive copper tape to obtain the voltage input. Copper tapes, applied to the internal grooves on the left and right sides of the casing, are used to output electrical signals after the logic operation via the sweeping contact of the titanium-nickel alloy wire on the side of the origami switching element. The left side of the casing of red origami switching element is equipped

with four pin headers, each serving specific functions: positive electrode, negative electrode of input status indication of the origami switching element, AND/NOR signal output, and OR/NAND signal output (Supplementary Fig. 9c). The resin casing is also equipped with grooves for placing 0.7 mm copper wire conductive networks, which connect the conductive copper tapes, LED lights, origami switching elements and pin headers to form two logic operation circuits and two indication circuits for the origami switching units (Supplementary Fig. 9d, e).

Due to the complexity of the combinational logic circuits, we simplified the circuit setup on the origami switching elements and casings, mainly retaining the conductive networks used for logic operations (Supplementary Figs. 12 and 13). Since the output results for the logic gate inputs (0,1) and (1,0) are the same, we replaced both inputs (0,1) and (1,0) for AND and NOR gates with the input (1,0) in the combinational logic design. As a result, the casing only has a circular hole at the top of the red origami switching element for mechanical input.

Electromagnetic Actuation System Design

We use an electromagnetic actuation method to apply mechanical input to the reconfigurable logic gates. The electromagnetic actuation system for testing the logic function of a single reconfigurable logic gate consists of an electromagnetic actuation circuit and origami logic operation circuit (Supplementary Figs. 10 and 14). The electromagnetic actuation circuit is composed of a DC stabilized power supply and four sets of actuation coils (Supplementary Fig. 10a, b). The DC power supply provides a stable input of up to 60 V with a maximum current of 30 V. The actuation coils are wound with 0.5 mm insulated wire, with a thickness of 10.5 mm and inner and outer diameters of 8 mm and 22 mm, respectively. The resistance of a single-wound actuation coil is approximately 1.23 Ω . All the actuation coils are arranged on a test platform fabricated using FDM 3D printing, based on the direction of mechanical input of the reconfigurable logic gate. On the left side, a coil is fixed for AND/NOR operations at the mechanical inputs (0,0), (1,0), and (1,1). On the right side, the coil is used for the (0,1) mechanical input. Since the folding of parallel origami switching elements does not interfere with each other, one mechanical input for OR/NAND logic operations requires two coils. To ensure that both origami switching elements experience the same compression force, the actuation coils for a single input are connected in parallel. The coil group on the upper side of the base is fixed for OR/NAND logic operations at the mechanical inputs (0,0), (1,0), and (1,1). The coil group on the lower side is fixed for the mechanical input (0,1) of OR/NAND logic operations. Each set of actuation coils is connected in parallel to the output terminals of the stabilized DC power supply. An N52 permanent magnet is placed at the center of the actuation coils to drive the deformation of the origami unit-cells. Different mechanical inputs can be implemented by switching circuits on the right side and adjusting the input voltage (Supplementary Fig. 10c–e). The 3D-printed test platform is equipped with a slot with female header connectors for fixing the reconfigurable logic gates and connecting them to the logic operation circuit. The logic operation circuit is composed of the reconfigurable logic gate, a 3 V battery, and an LED indicator light for the logic result in series.

Due to the complexity of the combinational logic circuits, the mechanical input for each reconfigurable logic gate is simplified. We complete the cascading of combinational logic by controlling the actuation coils of the next stage through the origami switching element of the previous level reconfigurable logic gate. For a 2-input, 1-output logic gate, the next stage input (0,0) indicates no signal from the previous stage output, (1,1) indicates two signals from both previous stage outputs, and (0,1) and (1,0) indicate that one signal from one of the previous stage outputs. The states of no signal, signal present in both outputs, and a signal present in only one output can be

considered as three distinct states based on force-electric conversion. Additionally, the AND/NOR logic operation yields the same output for inputs (0,1) and (1,0). Therefore, we can complete the AND/NOR logic operation with only two parallel actuation coils on one side for the input (Supplementary Fig. 14c, e). For the OR/NAND logic operation, the coils that actuate the parallel origami switching elements are connected individually to an origami switching element of the previous level logic gate in series. Thus, OR/NAND logic operations can be completed with two actuation coils on one side. The two sets of coils are positioned on the upper and lower sides of the test platform to reduce interference. The input for series-connected origami switching elements of AND/NOR logic operation is actuated by a C-shaped dowel bar with N52 permanent magnets. Since the combinational logic also requires NOT operations, we have designed a test platform for the NOT-Wire logic gate, which is equipped with vertical NOT logic network headers and horizontal headers that directly connect the inputs a, b, c, and d to the I-level reconfigurable logic gate (Supplementary Fig. 14d, f).

Fabrication of the reprogrammable logic cube

In order to verify the functionality of reprogrammable logic cubes, we fabricated a third-order prototype and completed the logic function testing. The third-order prototype comprises three layers: top/bottom logic input layers and a middle layer for force transmission and electrical conduction. Each layer integrates eight Buffer/NOT origami switching elements matching reprogrammable logic array structures (Fig. 5). Buffer/NOT origami switching elements feature copper sheets connecting to lateral NiTi alloy wires, and front NiTi alloy wires connecting to iron screws for signal conduction (Supplementary Fig. 33a, b). All switching elements are housed in 3D-printed resin casings and can be assembled into the third-order reprogrammable logic cube with deformation characteristics similar to a Rubik's Cube. Casings are classified as face-center (FC), edge, or corner blocks. Six FC blocks anchor via bearings to a perforated iron sphere (Supplementary Fig. 33c). This reprogrammable logic cube accepts binary input exclusively in the vertical direction. FC blocks I and II solely serve as connectors without force transmission capability due to obstruction by the central iron sphere. FC blocks III–VI reside in the middle layer, which also incorporates partial edge blocks. These FC blocks (III–VI) and middle-layer edge blocks mutually adhere through surface-mounted permanent magnets. Protruding cylinders are positioned at the right-rear side of middle-layer edge blocks, accommodating embedded permanent magnets that secure them to the central iron sphere for enhanced structural stability. The top and bottom edges of these edge blocks are additionally configured with permanent magnets to facilitate magnetic engagement with corner blocks. Middle blocks contain perforated origami cells with copper pillars conducting electricity and transmitting downward force (open top/closed bottom enables four mechanical inputs). Top/bottom layers contain edge/corner blocks. The protruding cylinder is configured at the posterior-superior aspect of the edge block, facilitating magnetic attachment to the central iron sphere (Supplementary Fig. 33d). Iron pillars are provided along the top and bottom edges of corner blocks for magnetic fixation to adjacent middle-layer edge blocks (Supplementary Fig. 33e). All block enclosed surfaces are perforated with hole apertures to permit mechanical force application. Copper tapes laminated along internal edges establish electrical connections to external copper wires for testing processed output signals after mechanical logic operations.

The second-order prototype is fabricated as a decision-making module of an embodied robotic. It consists of two layers, and each represents one bit of input (Supplementary Fig. 43a, b). The upper and lower layers contain four and three origami switching elements, respectively. The casing of the prototype is fabricated by photopolymerization 3D printing using high-strength resin materials. Each

layer's casing has a central M6 through-hole to accommodate a rotation axis. Additionally, a protrusion is designed to hold a thrust bearing, allowing smooth relative rotation between the upper and lower layers. Copper tape is adhered to the inner surface of the casing, while copper wires are arranged on the outer surface to work in conjunction with the origami switching elements to perform half-adder and half-subtractor logic functions (Supplementary Fig. 43c, d). The circuit diagram of the reprogrammable logic cube is shown in the Supplementary Fig. 43e–g. At the bottom of the lower-layer casing, pin headers are provided for the detachable networking of the reprogrammable logic cube (Supplementary Fig. 42).

Assembly and fabrication of the embodied intelligent robotic

The embodied intelligent environmental interaction robot is composed of three main modules: perception, decision-making, and enforcement (Supplementary Fig. 50a–c). The perception module consists of four photonic distance sensors (GP2Y0A21YK0F, SHARP Corporation) (Supplementary Fig. 50b). These sensors measure distances within a range of 10–80 cm, with a distance measurement accuracy of 0.1 cm at 10-bit AD resolution. The data update frequency is 25 Hz, with a cycle time of 40 ms. The enforcement module consists of two identical motor-driven wheels. The transformation of the reprogrammable logic cube is achieved by a rotating system consisting of 3D-printed torsion parts and a geared DC motor (Supplementary Fig. 50d). The mechanical inputs of the second-order reprogrammable logic cube are actuated by two thrust servo systems composed of 3D-printed thrust components and push rods (KTP0905, Xiamen Kabasi Electric Co., Ltd). The downward-actuating push rod 1 can retract 10 mm and completes the (0,1) mechanical input under different perceptual signals (Supplementary Fig. 50e). The upward-actuating push rod 2 individually retracts 10 mm and completes the (1,0) mechanical input (Supplementary Fig. 50f). And the (1,1) is achieved by simultaneous 10 mm retraction of both push rods 2 and 3. The photonic distance sensors, thrust servo mechanisms, rotating mechanisms, and the reprogrammable logic cube are supported by 3D-printed auxiliary components mounted on the robot's chassis. Additionally, a camera is installed at the front end of the robot's chassis to record the mechanical input conditions of the reprogrammable logic cube.

To enable information transfer among the perception, decision-making, and actuation modules in embodied intelligent environment-perception robots, it is necessary to convert analog voltage signals with varying amplitudes into digital signals. Thus, we specifically designed three relay-based circuits with analog-to-digital conversion capabilities—Circuit 1 (Supplementary Figs. 50g and 51a), Circuit 2 (Supplementary Fig. 50g, b), and Circuit 3 (Supplementary Figs. 50h and 52)—to facilitate robotic path planning. Circuit 1 is responsible for information transfer between the perception module and the thrust servo system, which processes mechanical input, in the decision-making module. The photoelectric distance sensors are divided into two groups: the first group, consisting of sensor b(ii) and sensor c, is used to read boundary information and determine mechanical input for path recognition; the second group, consisting of sensor a and sensor b(i), is used to read boundary information and determine the mechanical input when making the action decision command required by the execution module. Circuit 2 mediates information transfer between the decision-making module and the logically reconfigurable rotating system. Circuit 3 governs the robot's stop-recognition, action decision, as well as mediates information flow between the decision-making module and the drive wheels that execute motion commands.

Data availability

All the data supporting the findings of this study are available within the main text and its Supplementary Information. Source data are provided with this paper.

References

- Jin, Y. et al. Materials tactile logic via innervated soft thermochromic elastomers. *Nat. Commun.* **10**, 4187 (2019).
- Jiang, Y., Korpas, L. M. & Raney, J. R. Bifurcation-based embodied logic and autonomous actuation. *Nat. Commun.* **10**, 128 (2019).
- Treml, B., Gillman, A., Buskohl, P. & Vaia, R. Origami mechanologic. *Proc. Natl. Acad. Sci. USA* **115**, 6916–6921 (2018).
- Yasuda, H. et al. Mechanical computing. *Nature* **598**, 39–48 (2021).
- Conrad, S. et al. 3D-printed digital pneumatic logic for the control of soft robotic actuators. *Sci. Robot.* **9**, eadh4060 (2024).
- Yan, W. et al. Origami-based integration of robots that sense, decide, and respond. *Nat. Commun.* **14**, 1553 (2023).
- He, Q. et al. A modular strategy for distributed, embodied control of electronics-free soft robots. *Sci. Adv.* **9**, eade9247 (2023).
- Mei, T., Meng, Z., Zhao, K. & Chen, C. Q. A mechanical metamaterial with reprogrammable logical functions. *Nat. Commun.* **12**, 7234 (2021).
- Song, Y. et al. Additively manufacturable micro-mechanical logic gates. *Nat. Commun.* **10**, 882 (2019).
- Yang, N. et al. Bistable soft shells for programmable mechanical logic. *Adv. Sci.* **12**, 2412372 (2024).
- Li, Q., Tan, T., Wang, B. & Yan, Z. Avian-inspired embodied perception in biohybrid flapping-wing robotics. *Nat. Commun.* **15**, 9099 (2024).
- Bartolozzi, C., Indiveri, G. & Donati, E. Embodied neuromorphic intelligence. *Nat. Commun.* **13**, 1024 (2022).
- Howard, D. et al. Evolving embodied intelligence from materials to machines. *Nat. Mach. Intell.* **1**, 12–19 (2019).
- He, Q., Ferracin, S. & Raney, J. R. Programmable responsive metamaterials for mechanical computing and robotics. *Nat. Comput. Sci.* **4**, 567–573 (2024).
- Yasuda, H., Tachi, T., Lee, M. & Yang, J. Origami-based tunable truss structures for non-volatile mechanical memory operation. *Nat. Commun.* **8**, 962 (2017).
- Zhang, H., Wu, J., Fang, D. & Zhang, Y. Hierarchical mechanical metamaterials built with scalable tristable elements for ternary logic operation and amplitude modulation. *Sci. Adv.* **7**, eabf1966 (2021).
- Li, Y. et al. Reprogrammable and reconfigurable mechanical computing metastructures with stable and high-density memory. *Sci. Adv.* **10**, eado6476 (2024).
- El Helou, C., Buskohl, P. R., Tabor, C. E. & Harne, R. L. Digital logic gates in soft, conductive mechanical metamaterials. *Nat. Commun.* **12**, 1633 (2021).
- Byun, J., Pal, A., Ko, J. & Sitti, M. Integrated mechanical computing for autonomous soft machines. *Nat. Commun.* **15**, 2933 (2024).
- El Helou, C., Grossmann, B., Tabor, C. E., Buskohl, P. R. & Harne, R. L. Mechanical integrated circuit materials. *Nature* **608**, 699–703 (2022).
- Mei, T. & Chen, C. Q. In-memory mechanical computing. *Nat. Commun.* **14**, 5204 (2023).
- Li, S. & Mao, X. Training all-mechanical neural networks for task learning through in situ backpropagation. *Nat. Commun.* **15**, 10528 (2024).
- Xia, X., Spadaccini, C. M. & Greer, J. R. Responsive materials architected in space and time. *Nat. Rev. Mater.* **7**, 683–701 (2022).
- Jiao, Z. et al. Reprogrammable metamaterial processors for soft machines. *Adv. Sci.* **11**, 2305501 (2024).
- Gupta, A., Savarese, S., Ganguli, S. & Fei-Fei, L. Embodied intelligence via learning and evolution. *Nat. Commun.* **12**, 5721 (2021).
- Hu, X., Tan, T., Wang, B. & Yan, Z. A reprogrammable mechanical metamaterial with origami functional-group transformation and ring reconfiguration. *Nat. Commun.* **14**, 6709 (2023).
- Zhang, Y. et al. Mechanical system with soft modules and rigid frames realizing logic gates and computation. *Adv. Intell. Syst.* **5**, 2200374 (2023).

28. Helou, C., Hyatt, L., Buskohl, P. & Harne, R. Intelligent electroactive material systems with self-adaptive mechanical memory and sequential logic. *Proc. Natl. Acad. Sci. USA* **121**, e2317340121 (2024).
29. Meng, Z. et al. Encoding and storage of information in mechanical metamaterials. *Adv. Sci.* **10**, 2301581 (2023).
30. Jiao, P., Mueller, J., Raney, J. R., Zheng, X. & Alavi, A. H. Mechanical metamaterials and beyond. *Nat. Commun.* **14**, 6004 (2023).
31. Wu, L. et al. Mechanical metamaterials for handwritten digits recognition. *Adv. Sci.* **11**, 2308137 (2024).
32. Chi, Y. et al. Bistable and multistable actuators for soft robots: structures, materials, and functionalities. *Adv. Mater.* **34**, 2110384 (2022).

Acknowledgements

This work was supported by the National Key R&D Program of China (Grant No. 2023YFF0616800, Z.Y.), the Shanghai Key Technology Research and Development Program “Pioneer Program” (Project No. 25XF3200300, Z.Y.), the National Natural Science Foundation of China (Grant No.12572395, T.T.), and the National Natural Science Foundation of China (Grant No.12572176, Z.Y.).

Author contributions

X. H., T. T. and Z. Y. conceived the ideas and designed the research. X. H., Y. C. and Z. Y. established the theoretical model. X. H. performed the experiments and analyzed the data. X. H., T. T., Y. C. and Z. Y. interpreted the results and wrote the manuscript with input from all authors. Z. Y. and T. T. supervised the study.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41467-025-64956-0>.

Correspondence and requests for materials should be addressed to Ting Tan or Zhimiao Yan.

Peer review information *Nature Communications* thanks Shuangyue Yu and the other anonymous reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025