



US011539365B2

(12) **United States Patent**
Friedman et al.

(10) **Patent No.:** US 11,539,365 B2
(45) **Date of Patent:** Dec. 27, 2022

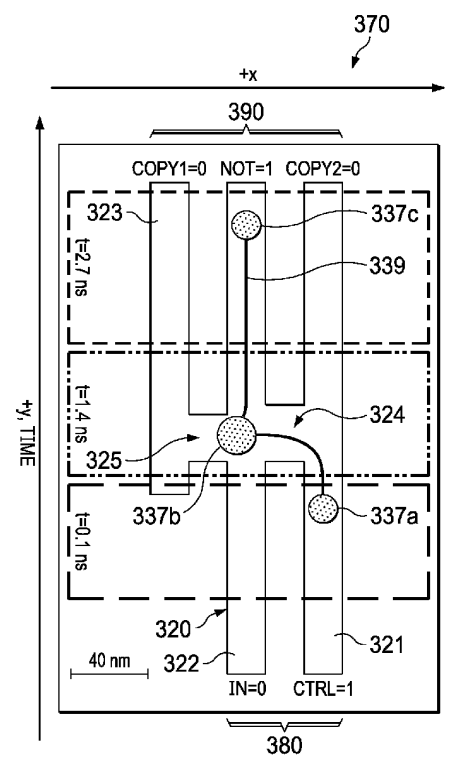
- (54) **REVERSIBLE COMPUTING SYSTEM AND METHOD BASED ON CONSERVATIVE MAGNETIC SKYRMION LOGIC**
- (71) Applicant: **Board of Regents, The University of Texas System**, Austin, TX (US)
- (72) Inventors: **Joseph S. Friedman**, Dallas, TX (US); **Xuan Hu**, Plano, TX (US); **Maverick Alisier Mathis Chauwin**, Saint-Paul-lès-Dax (FR)
- (73) Assignee: **Board of Regents, The University of Texas System**, Austin, TX (US)
- (*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 257 days.
- (21) Appl. No.: **16/914,845**
- (22) Filed: **Jun. 29, 2020**
- (65) **Prior Publication Data**
US 2020/0412366 A1 Dec. 31, 2020
- Related U.S. Application Data**
- (60) Provisional application No. 62/867,641, filed on Jun. 27, 2019.
- (51) **Int. Cl.**
H03K 19/00 (2006.01)
G06F 7/501 (2006.01)
H03K 19/20 (2006.01)

- (52) **U.S. CL.**
CPC **H03K 19/0008** (2013.01); **G06F 7/501** (2013.01); **H03K 19/20** (2013.01)
- (58) **Field of Classification Search**
None
See application file for complete search history.
- (56) **References Cited**
U.S. PATENT DOCUMENTS
2021/0367141 A1* 11/2021 Xiao H03K 19/18
FOREIGN PATENT DOCUMENTS
CN 106877858 A * 6/2017
CN 107332555 A * 11/2017
CN 108521275 A * 9/2018
CN 110445490 A * 11/2019
* cited by examiner

Primary Examiner — Tuan T Lam
(74) *Attorney, Agent, or Firm* — Yee & Associates, P.C.

(57) **ABSTRACT**
A skyrmion logic gate is provided. The logic gate comprises a first track configured for propagation of magnetic skyrmions and a second track configured for propagation of magnetic skyrmions. A junction links the first and second tracks. A continuous current flows through the logic gate, wherein skyrmions propagate due to the current.

24 Claims, 7 Drawing Sheets



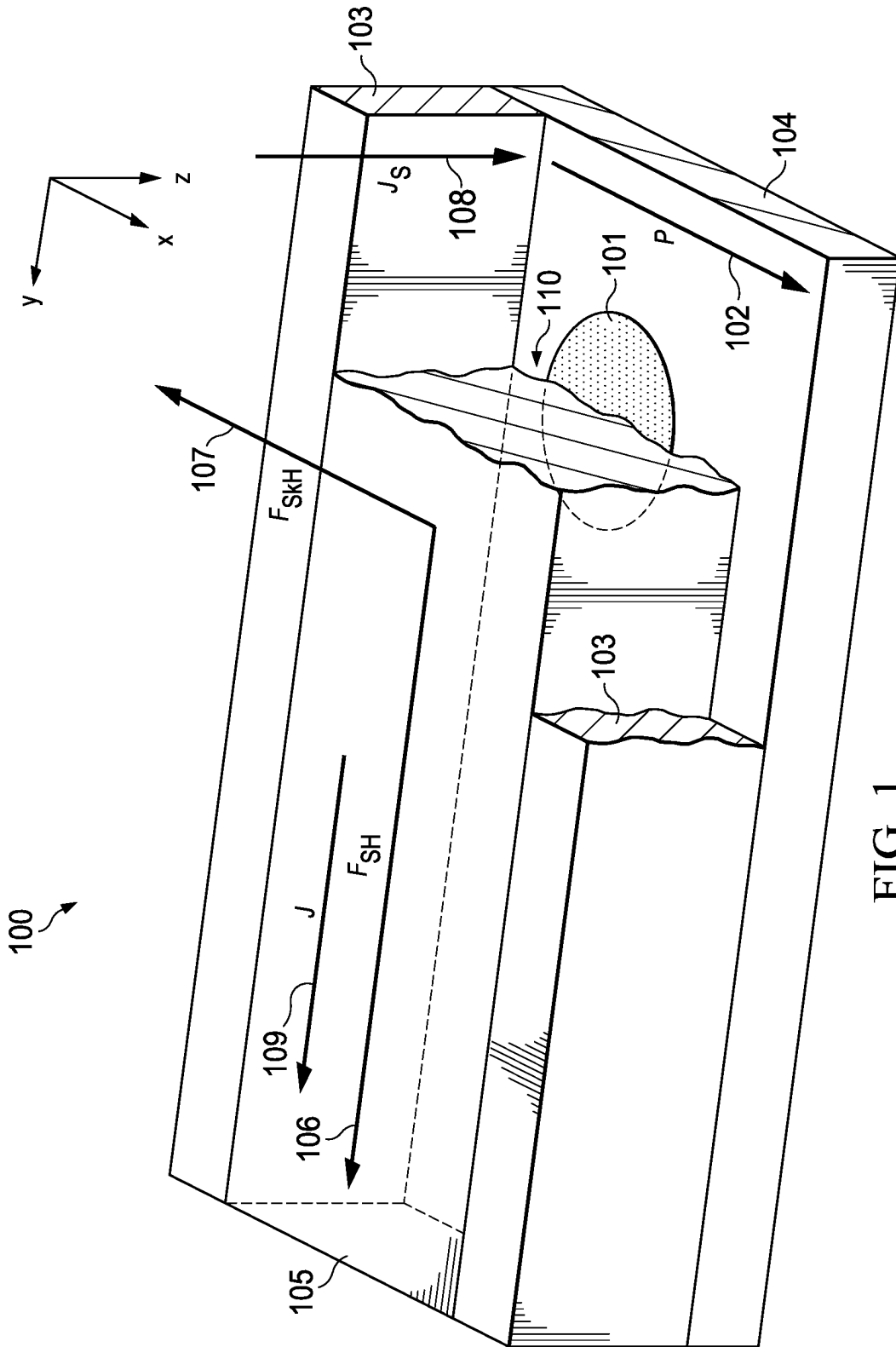


FIG. 1

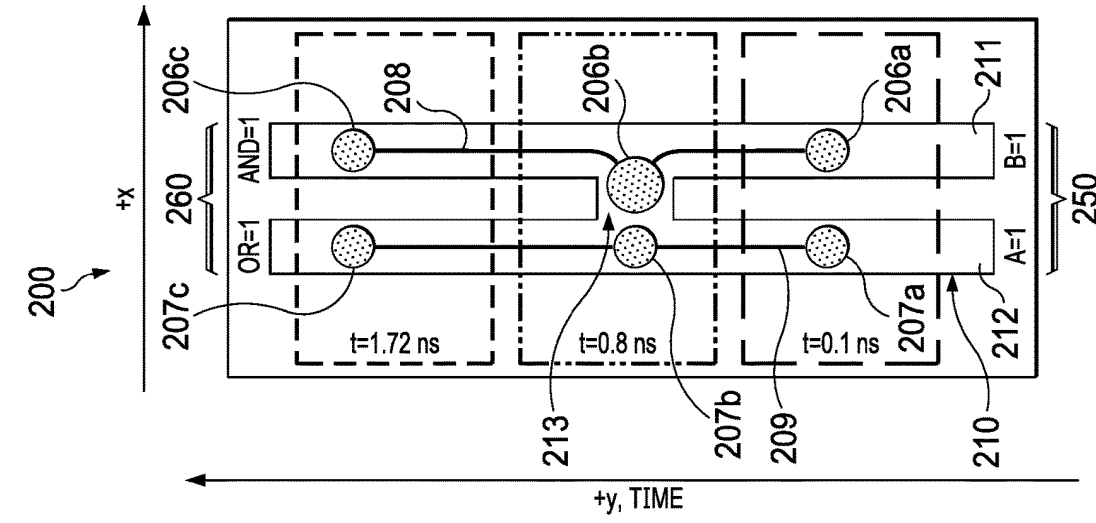


FIG. 2C

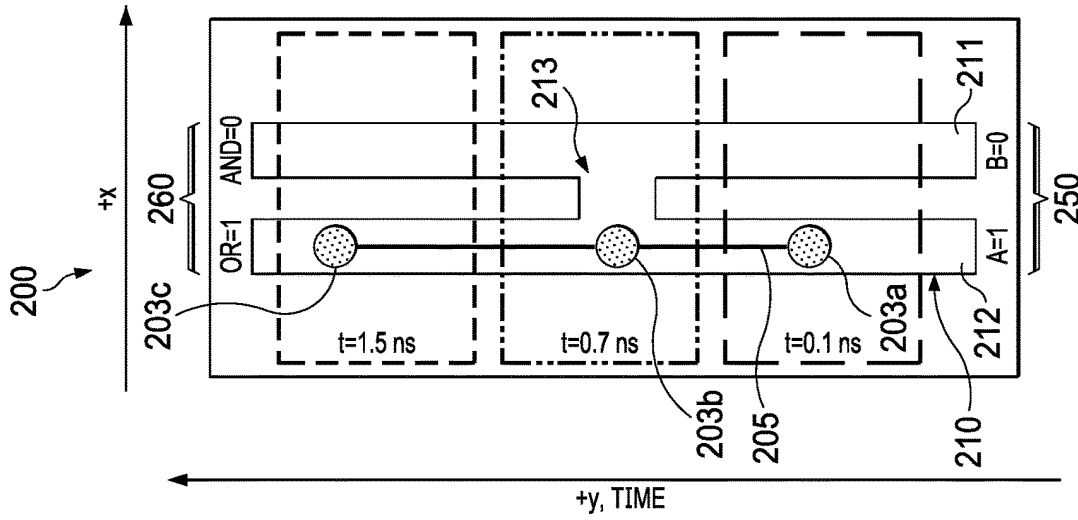


FIG. 2B

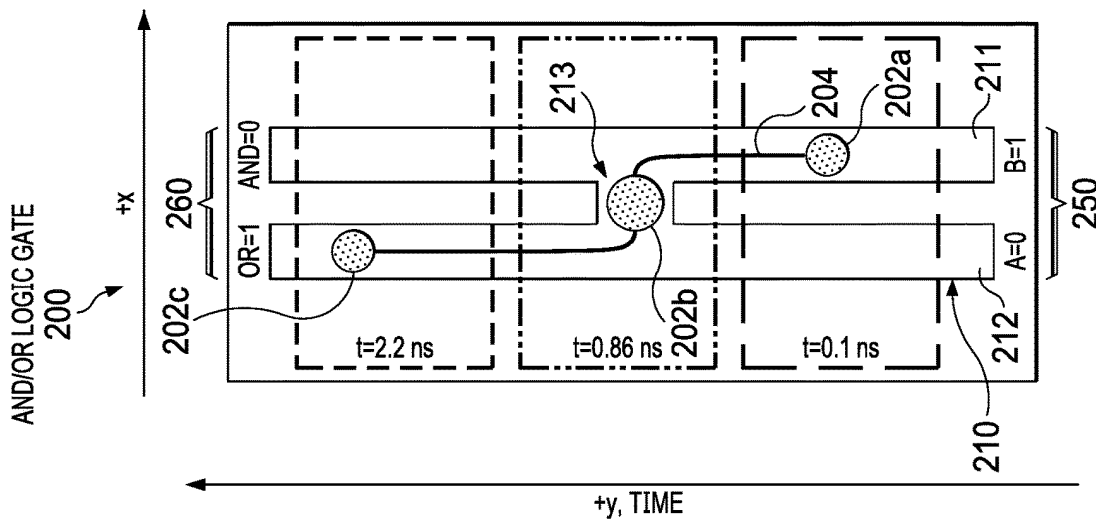


FIG. 2A

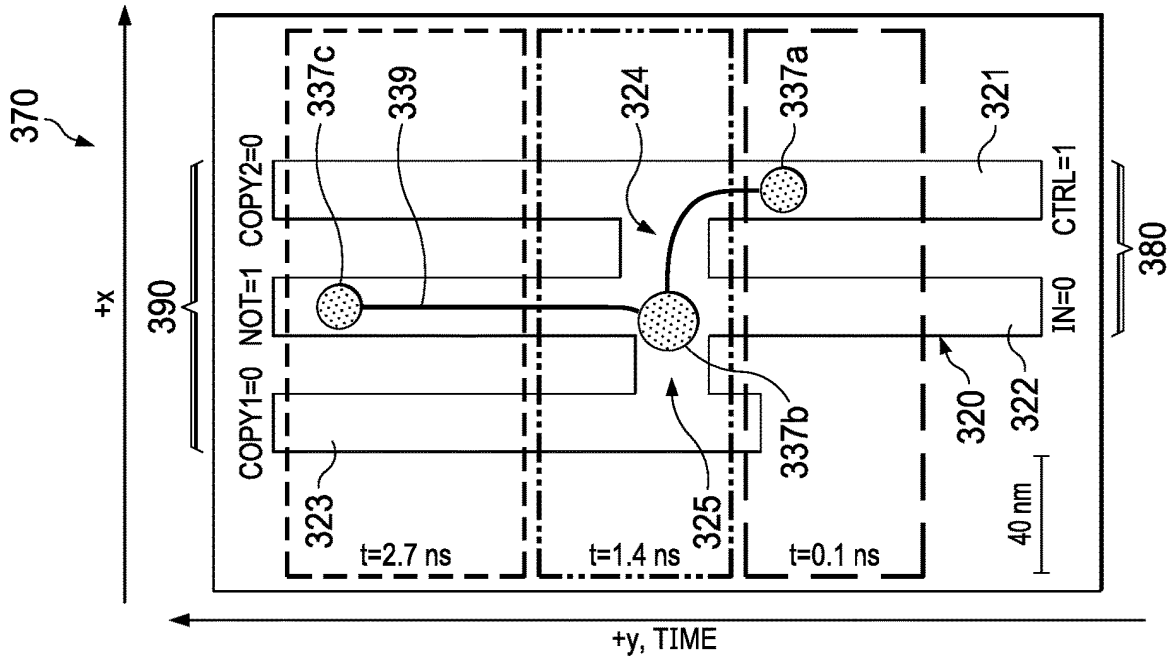


FIG. 3

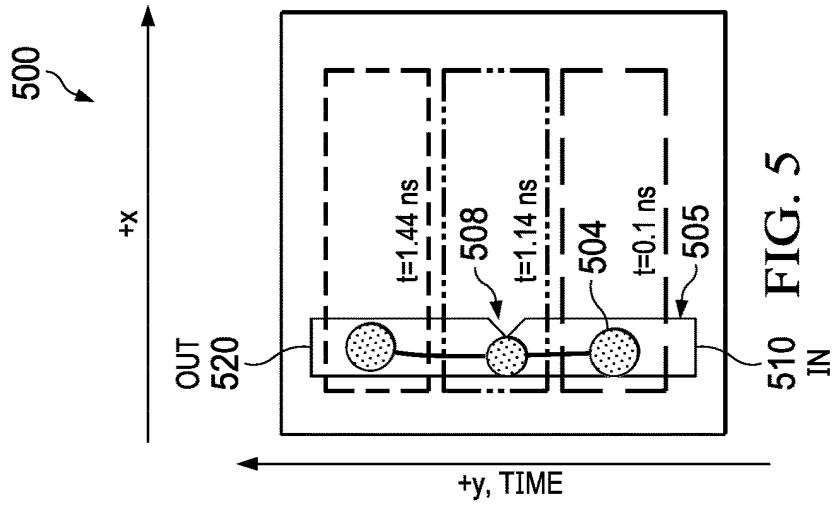
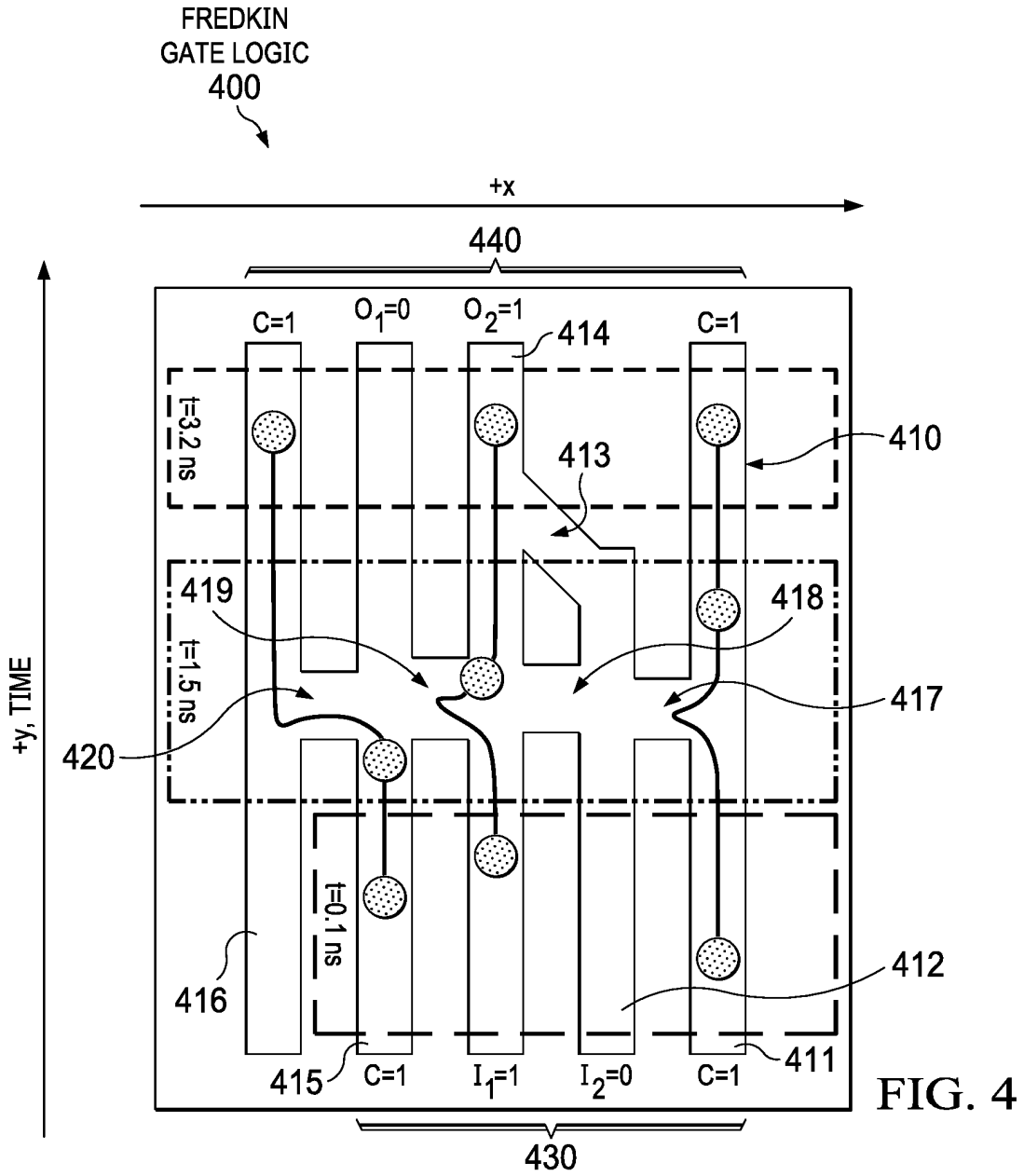
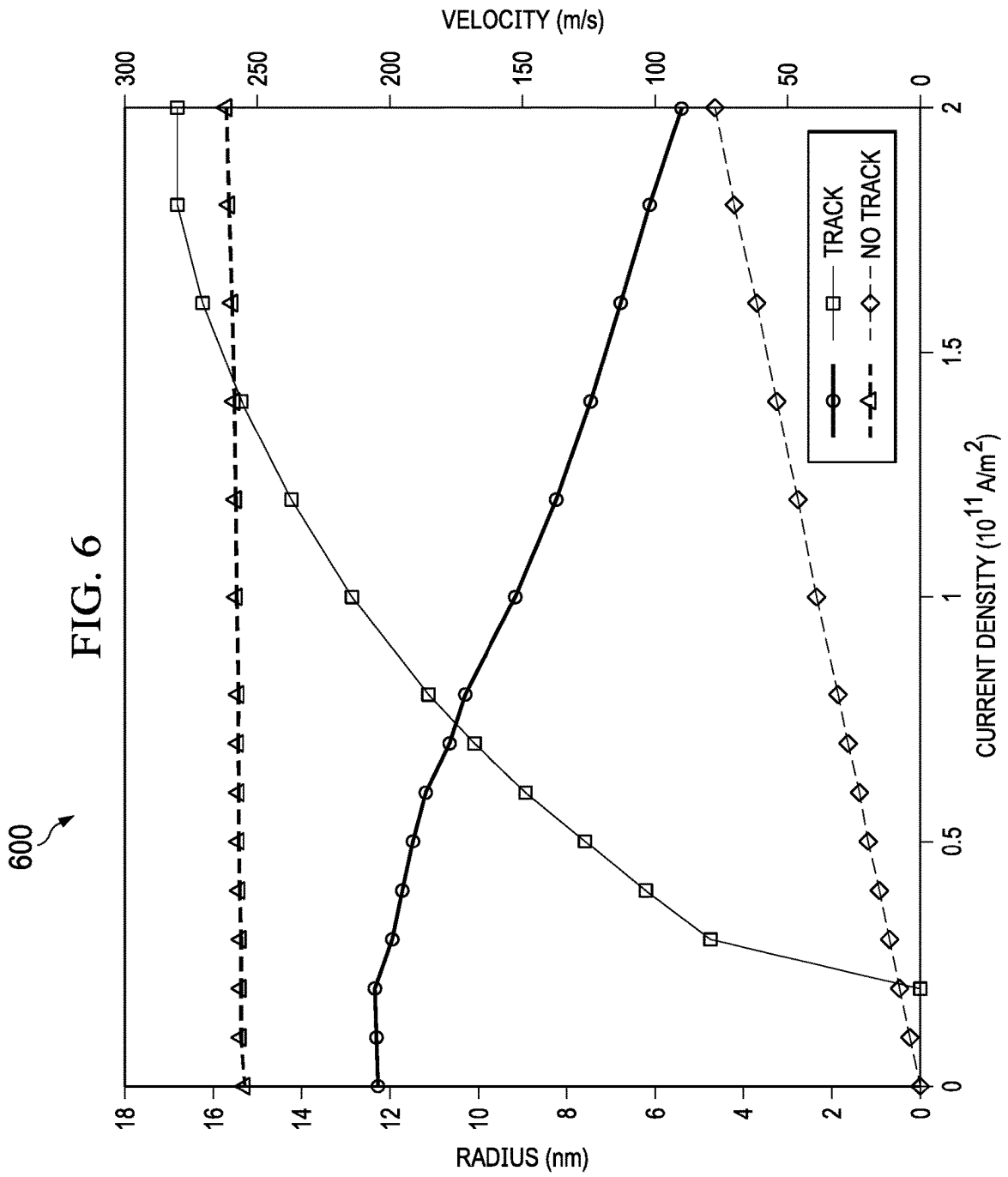


FIG. 5





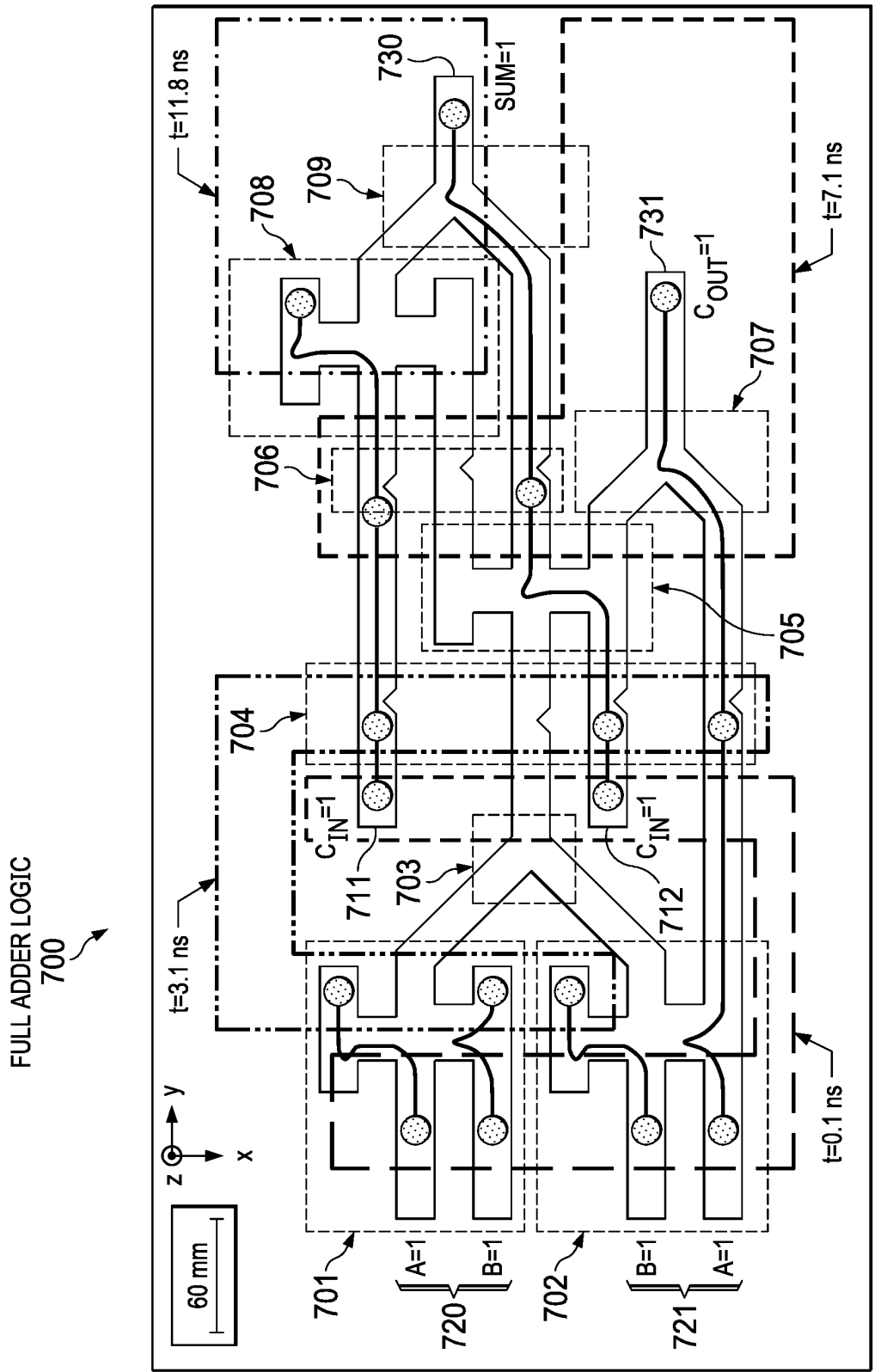


FIG. 7

PIPELINE FULL
ADDER LOGIC
800

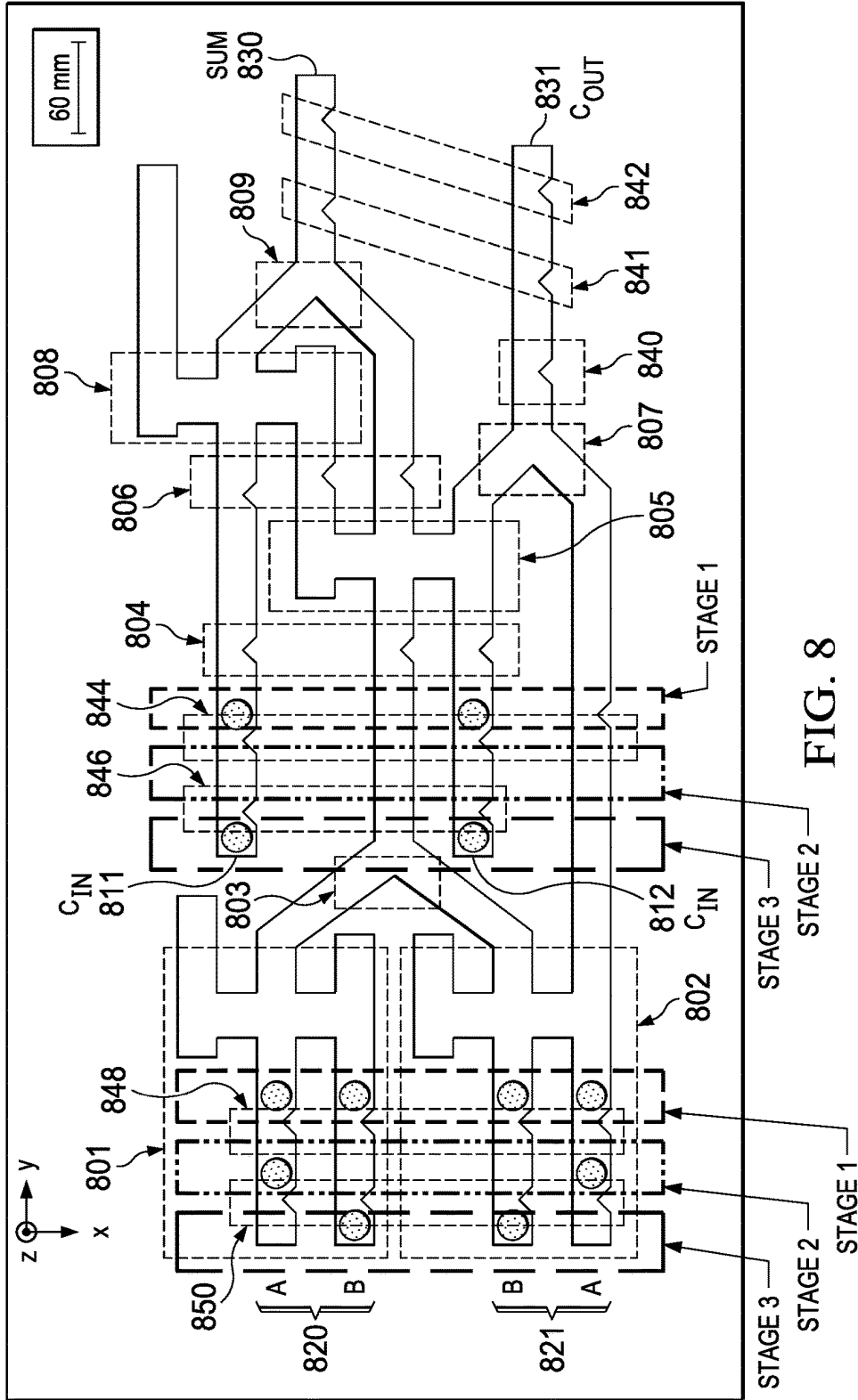


FIG. 8

1

REVERSIBLE COMPUTING SYSTEM AND METHOD BASED ON CONSERVATIVE MAGNETIC SKYRMION LOGIC

BACKGROUND INFORMATION

1. Field

The present invention relates in general to a novel computing system implementing conservative logic based on magnetic skyrmions for improved speed, power consumption, and energy efficiency for next generation computing.

2. Background

There is a fundamental minimum quantity of energy dissipated by a logic gate in which information-carrying signals are continuously created and destroyed. Reversible computing aims to circumvent this limitation by conserving energy as signals propagate through a logic circuit. In this scheme, conservative logical operations are executed through dissipation-free elastic interactions among information carriers that conserve momentum and energy.

A recently developed method combines conservative logic experimentally with micron-sized droplets driven through planar computing structures by pressure and magnetism. However, the large dimensions of the information carriers in these demonstrations detract from the system efficiency and limit potential utility. Therefore, a nanoscale information carrier for reversible computing remains elusive.

Magnetic skyrmions are intriguing information carriers for reversible computing due to their small diameter (~20 nm) and the small current required to induce skyrmion motion. These quasiparticles are topologically stable regions of magnetization comprising a central core oriented antiparallel to the bulk of a magnetic structure. Skyrmion motion involves the propagation of magnetization rather than the transport of physical particles and can be induced by the spin-Hall effect through the application of an electrical current. Magnetic skyrmions propagating along ferromagnetic nanowire tracks have been proposed for memory storage and individual logic gates. However, the development of a scalable skyrmion computing system has been impeded by the need to directly cascade skyrmion logic gates without control and amplification circuitry that significantly reduces the computing system efficiency. In addition, previous proposed skyrmion logic required the continual creation and annihilation of skyrmions, which is an energetically expensive process that requires an external control system.

Therefore, it would be desirable to have an apparatus and system that take into account at least some of the issues discussed above, as well as other possible issues.

SUMMARY

An illustrative embodiment provides a skyrmion logic gate. The logic gate comprises a first track configured for propagation of magnetic skyrmions and a second track configured for propagation of magnetic skyrmions. A junction links the first and second tracks. A continuous current flows through the logic gate, wherein skyrmions propagate due to the current.

Another illustrative embodiment provides a logic circuit comprising a number of skyrmion logic gates. Each skyrmion logic gate comprises two or more tracks configured for

2

propagation of magnetic skyrmions at least one junction linking multiple tracks together. A continuous current flows through the logic gates, wherein skyrmions propagate due to the current. Output skyrmions from logic gates in the circuit act as input skyrmions for other logic gates in the circuit.

Another illustrative embodiment provides a method of performing logical computations with magnetic skyrmions. The method comprises inputting a number of magnetic skyrmions into a logic gate. The logic gate comprises two or more tracks configured for propagation of magnetic skyrmions and at least one junction linking multiple tracks together. A continuous electrical current is applied through the logic gates, wherein the skyrmions propagate due to the current.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the illustrative embodiments are set forth in the appended claims. The illustrative embodiments, however, as well as a preferred mode of use, further objectives and features thereof, will best be understood by reference to the following detailed description of an illustrative embodiment of the present disclosure when read in conjunction with the accompanying drawings, wherein:

FIG. 1 illustrates skyrmion propagation along a track with which illustrative embodiments can be implemented;

FIG. 2A illustrates skyrmion propagation through a logic AND/OR gate when an input skyrmion is only present in the B track in accordance with an illustrative embodiment;

FIG. 2B illustrates skyrmion propagation through a logic AND/OR gate when an input skyrmion is only present in the A track in accordance with an illustrative embodiment;

FIG. 2C illustrates skyrmion propagation through a logic AND/OR gate and skyrmion-skyrmion repulsion when input skyrmions are present in both tracks in accordance with an illustrative embodiment;

FIG. 3 illustrates an invert/copy logic gate with skyrmion implementation in accordance with an illustrative embodiment;

FIG. 4 illustrates a Fredkin gate with skyrmion implementation in accordance with an illustrative embodiment;

FIG. 5 illustrates a signal synchronization structure in accordance with an illustrative embodiment;

FIG. 6 depicts a graph illustrating skyrmion radius as a function of applied electrical current density in accordance with illustrative embodiments;

FIG. 7 illustrates a cascaded one-bit full adder logic circuit with skyrmion implementation in accordance with an illustrative embodiment; and

FIG. 8 illustrates a pipelined full adder logic circuit with skyrmion implementation in accordance with an illustrative embodiment.

DETAILED DESCRIPTION

The illustrative embodiments recognize and take into account one or more different considerations. The illustrative embodiments recognize and take into account that as transistor scaling approaches its ultimate limits, alternative computing paradigms are being explored that circumvent the challenges of conventional computing systems. In particular, reversible computing provides an approach for information processing with zero energy dissipation. However, previous theoretical and experimental implementations of reversible computing use information carriers that cannot be transported in a scalable and efficient manner.

3

The illustrative embodiments also recognize and take into account that conventional information carriers are not energy efficient. The illustrative embodiments also recognize and take into account that the size of conventional information carriers is limited and not desirable.

As used herein, “magnetic skyrmion” or “skyrmion” refers to topologically non-trivial spin textures with integer topological charges that occur in helimagnetic materials in which broken inversion symmetry can give rise to the Dzyaloshinskii-Moriya interaction.

The disclosed skyrmion logic system of the illustrative embodiments provides a nanoscale solution to the challenge of reversible computing, which enables large-scale logic circuits with minimal energy dissipation.

The illustrative embodiments provide a reversible skyrmion logic system in which skyrmions are conserved as they flow through nanowire tracks. The skyrmions do not disintegrate or disappear from the system over time and may enter, propagate through, remain in a stable track position, or exit many tracks over time. Logical operations are performed by thoroughly leveraging the rich physics of magnetic skyrmions, for example, the spin-Hall effect, the skyrmion-Hall effect, skyrmion-skyrmion repulsion, repulsion between skyrmions and the track boundaries, and electrical current-control of notch depinning.

Binary information may be encoded by the presence (‘1’) or absence (‘0’) of magnetic skyrmions, with the skyrmions flowing directly from the output nanowire track of one logic gate to the input track of another logic gate without an external control or amplification circuit. These reversible skyrmion logic gates may provide fan-out functions and be integrated into a large-scale system, with signal integrity provided by simple electronic clock pulses applied to the entirety of the system. The logic-in-memory computing system of the illustrative embodiments may be non-volatile due to the topological stability and ferromagnetic nature of skyrmions, providing efficient pipelining that enhances the potential for high speed and low power.

Micromagnetic simulations are performed using mumax3, an open-source GPU-accelerated micromagnetic simulation software, which integrates the Landau-Lifshitz-Gilbert (LLG) equation of motion with a Finite Difference approach. The sample may be discretized into cuboid cells whose dimensions were set to 1 nm×1 nm×0.4 nm, and the thermal fluctuations may be neglected by setting the temperature to OK.

The LLG equation of motion describes magnetization dynamics in ferromagnetic materials:

$$\frac{\partial M}{\partial t} = -\gamma(M \times H_{\text{eff}}) + \frac{\alpha}{M_s} \left(M \times \frac{\partial M}{\partial t} \right) + \tau_{CPP} \quad (1)$$

where M is the magnetization vector, γ is the gyromagnetic ratio, M_s is the saturated magnetization, and α is the Gilbert damping parameter. H_{eff} is the effective field which includes exchange, anisotropy, magnetostatic, Dzyaloshinskii-Moriya and external magnetic fields. τ_{CPP} implements the injection of spin-Hall current perpendicularly to the sample and is described by:

$$\tau_{CPP} = -\beta e' (M \times m_p) - \frac{\beta}{M_s} (M \times (m_p \times M)) \quad (2)$$

4

with m_p as the spin-Hall polarization direction and

$$\beta = \frac{\theta_{SH} \hbar J}{2M_s e' t_{Co,track}},$$

where θ_{SH} is the spin-Hall angle, e is the electronic charge, J is the electrical current density, and $t_{Co,track}$ is the thickness of the Co track. e' is the field-like torque, which is here considered to be zero.

Magnetic parameters may be selected to model a multi-layer of Pt and Co, with saturation magnetization $M_s=5.80 \times 10^5$ A/m, exchange stiffness $A_{ex}=1.5 \times 10^{-11}$ J/m, Gilbert damping coefficient $\alpha=0.1$, DMI constant $D_{ind}=3.0 \times 10^{-3}$ J/m², magneto-crystalline anisotropy constants $Ku_1=6 \times 10^5$ J/m³ and $Ku_2=1.5 \times 10^5$ J/m³, and spin polarization in the transverse direction $M_p=(1,0,0)$. The anisotropy direction points upwards. The spin-Hall angle θ_{SH} is considered to be equal to 1. The thickness of the Pt layer is $t_{Pt}=0.4$ nm and the thickness of the Co layer varies between $t_{Co,track}=0.4$ nm and $t_{Co,sample}=0.8$ nm elsewhere.

In a fully-reversible skyrmion logic system, skyrmions must only be generated once, at the beginning of the system operation. After these initial skyrmions are generated, they are continually propagated through the logic gates such that the output skyrmions of each conservative skyrmion logic gate are used as the input skyrmions of other conservative skyrmion logic gates. These skyrmions generated at system initialization are therefore sufficient for long-term use of this system, and their non-volatility enables the skyrmions to maintain their states even when the power supply is removed.

Alternatively, ease-of-demonstration as well as optimization of the conventional metrics of speed, power, and delay may require a compromise regarding the conservation of skyrmions. Another approach to skyrmion generation is to continually generate skyrmions at specific points within the system with a dedicated skyrmion generation structure. This may be achieved with homogeneous currents, or with nanosecond electrical current pulses for a device-compatible stripline geometry. This skyrmion generator would generate a skyrmion in each clock cycle, which then can be provided as an input to a particular logic gate where an input skyrmion is always required.

To read the binary outputs of this conservative logic system, it is necessary to detect the presence (‘1’) or absence (‘0’) of skyrmions at various points throughout the circuit.

While standard magnetic force microscopy imaging can be used to detect skyrmions in a laboratory setting, computing applications require transformation of the skyrmion information into electrical signals. Determination of the presence or absence of a skyrmion at a particular location can be achieved by placing a tunneling barrier and hard ferromagnet above the Co ferromagnet to form a magnetic tunnel junction (MTJ). The magnetoresistance of this MTJ indicates the presence or absence of a skyrmion within the free layer, as the skyrmion modifies the local magnetization within the free layer and therefore the current through the MTJ tunneling barrier.

Each of the skyrmion logic gates in the illustrative embodiments conserves skyrmions by propagating each input skyrmion to an output port. The operation of a complete conservative logic system requires the conservation of skyrmions throughout the system. Every skyrmion transmitted to the output port of a skyrmion logic gate is then used as an input to another skyrmion logic gate.

5

Skyrmions propagate to several superfluous output ports that contain logical by-products of the computation of the sum and carry-out signals. For example, the full adder logic gate produces the signals A , B , $A \wedge B$, C_{IN} , and $C_{IN} \wedge (A \oplus B)$. Because these are by-products of the full adder computation that are not the primary objectives of the full adder circuit, they are therefore available for use in other logic gates.

To enable a reversible system, the skyrmion signals must be able to propagate to other logic gates. As the lateral flow of information through this two-dimensional structure is unidirectional (left-to-right), it is necessary to provide a technique by which skyrmions are provided to the left side of the circuit. This provision of skyrmions can be achieved by using an additional circuit layer, which also enables the interaction-free cross-over of skyrmion tracks. Furthermore, the direction of the spin current can be modified through static or dynamic modulation of the electrical current direction or the ferromagnetic Co magnetization. Moreover, it should be noted that it may be worthwhile to shed the requirement of complete skyrmion conservation in order to maximize the primary metrics of a computing system, for example, energy consumption, processing speed, and area footprint.

FIG. 1 illustrates skyrmion propagation along a track with which illustrative embodiments can be implemented. Skyrmions **101** propagate along a track **100** comprised of a Co Ferromagnet wall **104** and Co Ferromagnet walls **103** with Pt heavy metal **105** between the walls. The Co ferromagnet material has polarization P **102**, where an interfacial spin-orbit coupling induces a Dzyaloshinskii-Moriya interaction. This spin-orbit coupling also causes the externally-applied electrical current (J) **109** flowing through the heavy metal in the $+y$ -direction to create a spin current (J_s) **108** polarized in the $+z$ -direction via the spin-Hall effect. The skyrmion **101** lies in the ferromagnetic layer at the interface **110** with the heavy metal **105**, and the surrounding ferromagnet walls **103** prevent the skyrmion **101** from leaving the track **100**. Spin current **108** in the $+z$ -direction produces a force F_{SH} **106** on the skyrmion **101** in the $+y$ -direction, the direction of electrical current **109**. The track constriction prevents the $-x$ -directed skyrmion-Hall force F_{SKH} **107** from influencing the skyrmion trajectory. It should be noted that the axes are inverted for visual clarity; Pt heavy metal **105** is generally below Co ferromagnet wall **104**.

FIGS. 2A-2C illustrates a skyrmion propagation through an AND/OR logic gate in accordance with an illustrative embodiment. The AND/OR logic gate **200** comprises an H-shaped set of tracks **210**. FIGS. 2A-2C are top-down views of tracks having structure as indicated in FIG. 1. Set of tracks **210** have input ports **250** and output ports **260**. Input ports **250** comprise an A port and a B port in which skyrmions enter the logic gate (input skyrmions), and output ports **260** comprise an OR port and an AND port where skyrmions exit the logic gate (output skyrmions). The set of tracks comprise a right track B **211**, a left A track **212**, and a central track/junction **213** connecting the tracks **211** and **212**. Skyrmion trajectories may change near central junction **213** based on skyrmion-track and skyrmion-skyrmion repulsion. The states of the AND/OR logic gate are shown for different input combinations: $A=0$, $B=1$ (FIG. 2A); $A=1$, $B=0$ (FIG. 2B); and $A=B=1$ (FIG. 2C) at the input ports **250**. Table 1 is an example truth table for the AND/OR gate, where N is the number of skyrmions in the gate for each logic combination.

6

TABLE I

Truth table for the AND/OR gate.				
Inputs		Outputs		
A	B	N	AND	OR
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	2	1	1

FIG. 2A illustrates skyrmion propagation through a logic AND/OR gate when the skyrmion is only provided to the B track **211**. The presence of a skyrmion in a track represents a binary 1 in the track at a position and time. When a skyrmion enters the B track **211** of input ports **250** of the AND/OR gate, this logic gate geometry forces the skyrmion to propagate via trajectory **204** to the OR port of output ports **260**, representing binary 1. The spin current J_s pushes skyrmions in the $+y$ -direction, with a skyrmion-Hall force directed in the $-x$ -direction. The positions of skyrmions as a function of time are shown as position **202a** at $t=0.1$ ns, position **202b** at $t=0.86$ ns, and positions **202c** at about $t=2.2$ ns. The travel times for the skyrmions, and therefore the position times, may be shown for a constant current J where $J=5 \times 10^{10}$ A/m².

In an embodiment, the total number of skyrmions N provided to the inputs A and B are equal to the total number of skyrmions emitted by the AND and OR outputs. The spin-Hall effect pushes the skyrmions in the $+y$ -direction through the tracks, while the skyrmion-Hall effect introduces a $-x$ -directed force that is mediated by repulsion from the track boundaries. The skyrmions are therefore free to move laterally within the central junction, where the skyrmion-Hall effect causes leftward skyrmion propagation unless repulsed by a second skyrmion.

FIG. 2B illustrates skyrmion propagation through a logic AND/OR gate when the skyrmion is only provided to the A track. When a skyrmion enters the A track **212** of input ports **250** of the AND/OR gate, the logic gate geometry forces the skyrmion to propagate via trajectory **205** to the OR port of output ports **260** to represent binary 1. The positions of skyrmion as a function of time are shown as position **203a** at $t=0.1$ ns, position **203b** at $t=0.7$ ns, and positions **203c** at about $t=1.5$ ns. Applying a different current will change the times of each position shown. Since it is confined laterally by the tracks, the input skyrmion travels directly in the $+y$ -direction until it reaches the central junction **213**. In the lateral opening of the constrictive tracks at the central junction **213**, the skyrmion-Hall force induces a $-x$ -directed component to the skyrmion trajectory that keeps the skyrmion in the A track **212**.

FIG. 2C illustrates skyrmion propagation through a logic AND/OR gate and skyrmion-skyrmion repulsion when skyrmions are provided to both tracks. When two skyrmions enter input ports **250**, one in A track **212** and one in B track **211**, one skyrmion exits the OR output port and the other skyrmion exits the AND output port such that both output ports produce binary 1. Due to skyrmion-skyrmion repulsion at central junction **213**, the skyrmion entering track B **211** remains in track B rather than crossing the central junction **213** as in FIG. 2A. The respective positions of the skyrmions as a function of time are shown as position **206a** and **207a** at $t=0.1$ ns, position **206b** and **207b** at $t=0.8$ ns, and positions **206c** and **207c** at about $t=1.72$ ns.

In another embodiment, when no skyrmions enter either input port, no skyrmions are emitted by either output port, representing binary 0 outputs. With skyrmions entering both input ports **250** the combined forces resulting from the spin-Hall effect, the skyrmion-Hall effect, skyrmion-skyrmion repulsion, and the boundaries cause this structure to simultaneously calculate the logical functions $A \setminus B$ and $A \wedge B$ while conserving the skyrmions.

FIG. 3 illustrates an invert/copy logic gate with skyrmion implementation in accordance with an illustrative embodiment. In concert with the AND/OR gate, the inversion operation enables the generation of all possible Boolean logic functions. Such a logic operation cannot be achieved by the AND and OR operations alone. The INVERT/COPY gate of the illustrative embodiments has an additional output port and the requirement that a skyrmion always be provided to the control (CTRL) input. This reversible INVERT/COPY gate simultaneously duplicates and inverts the skyrmion input signal.

The INVERT/COPY logic gate **370** includes a set of tracks **320**, wherein each track has the structure as described in FIG. 1. Tracks **320** have input ports **380** and output ports **390**. Input ports **380** comprise an IN port and a CTRL port into which skyrmions enter the logic gate (input skyrmions). Output ports **390** comprise a COPY1 port, a COPY2 port, and a NOT port where skyrmions exit the logic gate (output skyrmions). The set of tracks comprise a right track **321**, a middle track **322** and a left track **323**. Tracks **321** and **322** are connected by junction **324** near the center of the logic gate. Tracks **322** and **323** are connected by junction **325** also near the center of the logic gate. Skyrmion trajectories may change near the center of the logic gate at junctions **324** and **325** based on skyrmion-track and skyrmion-skyrmion interactions.

In an embodiment, the states of the INVERT/COPY logic gate may be designed to have different input combinations, for example, IN=1 with CTRL=1, or IN=0 with CTRL=1 at the input ports **380**. Table 2 is an example truth table for an INVERT/COPY gate.

TABLE II

Truth table for the INV/COPY gate.					
Inputs			Outputs		
CTRL	IN	N	COPY1	NOT	COPY2
1	0	1	0	1	0
1	1	2	1	0	1

In FIG. 3, the spin current J_s pushes skyrmions in the +y-direction, with a skyrmion-Hall force directed in the -x-direction. The positions of skyrmions as a function of time are shown as position **337a** at $t=0.1$ ns, position **337b** at $t=1.4$ ns, and positions **337c** at $t=2.7$ ns. The travel times for the skyrmions, and therefore the position times, may be shown for a constant current J where $J=5 \times 10^{10}$ A/m². A different current may be applied to change the times of each position shown. Since skyrmions are confined laterally by their tracks, the various input skyrmions travel directly in the +y-direction along the various trajectory **339** as shown.

As shown in FIG. 3, when the skyrmion input at the CTRL input reaches junction **324**, the skyrmion-Hall force pushes it in the -x-direction into track **322**, resulting in an output of COPY1=0, NOT=1, and COPY2=0.

If skyrmions are input at both input ports **380** wherein IN=1 and CTRL=1 (not shown), the skyrmion-skyrmion

repulsion at junction **324** counteracts the skyrmion-Hall force to keep the CTRL input skyrmion in track **321**. However, the combination of skyrmion-skyrmion repulsion and skyrmion-Hall forces do push the IN input skyrmion across junction **325** and into track **323**, resulting in an output of COPY1=1, NOT=0, and COPY2=1.

Therefore, the NOT output is 1 whenever the IN input is 0, and 0 whenever the IN input is 1. The reversible logic gate may also perform the fan-out function, where skyrmions are conserved such that the IN signal is duplicated to the two COPY outputs. This signal duplication is an essential component of a large-scale computing system and can be performed repeatedly by cascaded INVERT/COPY gates to generate numerous copies of a signal.

FIG. 4 illustrates a Fredkin gate with skyrmion implementation in accordance with an illustrative embodiment. The Fredkin logic gate **400** comprises a set of tracks **410** wherein each track has the structure as described in FIG. 1. The set of tracks **410** comprise a right track **411**, a right-middle track **412**, a center track **414**, a connecting track **413** which connects track **413** to center track **414** at an angle, a left-middle track **415**, and left track **416**. The tracks **410** further comprise junction **417** connecting track **411** to track **412** near the center of the logic gate, junction **418** connecting track **412** to track **414**, junction **419** connecting track **414** to track **415**, and junction **420** connecting track **415** to track **416**.

Tracks **410** further comprise input ports **430** and output ports **440**. Input ports **430** comprise left and right control inputs C , I_1 , and I_2 . Skyrmions enter the logic gate (input skyrmions) via input ports **430**. Output ports **440** comprise left and right outputs C and outputs O_1 and O_2 . Skyrmions exit the logic gate (output skyrmions) via output ports **440**. Skyrmion trajectories may change near the center of the logic gate at the junctions based on the skyrmion-Hall force and skyrmion-skyrmion interactions.

Skyrmions provided to the control inputs C respectively propagate to the C outputs and determine whether or not the I_1 and I_2 input signals are swapped as they travel to the O_1 and O_2 outputs. Again, the spin current J_s pushes skyrmions in the +y-direction, with a skyrmion-Hall force directed in the -x-direction.

In the example shown in FIG. 4, the inputs are $C=1$, $I_1=1$, and $I_2=0$. As a result of skyrmion-skyrmion repulsion, the inputs I_1 and I_2 swap paths such that $O_1=I_2$ and $O_2=I_1$. In another example (not shown), when $C=0$, the input signals I_1 and I_2 travel directly to the output ports without swapping; that is, $O_1=I_1$ and $O_2=I_2$. The C signal is duplicated, with both skyrmions at the C input propagating directly to the C output.

Table 3 is an example truth table for a Fredkin logic gate.

TABLE III

Truth table for the Fredkin gate.						
Inputs				Outputs		
C	I_1	I_2	N	C	O_1	O_2
0	0	0	0	0	0	0
0	0	1	1	0	0	1
0	1	0	1	0	1	0
0	1	1	2	0	1	1
1	0	0	2	1	0	0
1	0	1	3	1	1	0
1	1	0	3	1	0	1
1	1	1	4	1	1	1

9

In the illustrative embodiments, the output skyrmions emitted by one logic gate are used as input skyrmions for another gate. As the logic gate functionality is based on skyrmion interactions at the central junctions, a synchronization mechanism can be provided to ensure that skyrmions arriving from different input paths reach the central junction simultaneously.

FIG. 5 illustrates a signal synchronization structure in accordance with an illustrative embodiment. A skyrmion track system 500 is shown with an input port IN 510, and an output port OUT 520. Skyrmion 504 enters the track 505 through input port 510 and is shown at different times as it travels along the track in the +y direction. Track 505 includes a notch structure 508 near the center of the track.

In practice, synchronization is achieved with the notch structure 508 which prevents the skyrmion from moving farther down the track unless a large current is applied. A large spin-Hall current pulse enables the skyrmion 504 to traverse the notch 508 by causing a decrease in skyrmion diameter while also increasing the skyrmion velocity.

FIG. 6 depicts a graph illustrating skyrmion radius as a function of applied electrical current density in accordance with illustrative embodiments.

In an embodiment, the notch 508 may be, e.g., 7 nm wide in a 20 nm-wide nanowire track to create a constriction that permits skyrmion passage only when a large current is applied. The electrical current applied to the entirety of the computing system may maintain a constant low magnitude of $J=5 \times 10^{10}$ A/m² that is periodically amplified to $J=2 \times 10^{11}$ A/m² for 150 ps in order to enable skyrmions to traverse notches throughout the system. The skyrmion 504 traverses the notch 508 when this large clock pulse is applied at $t=1$ ns.

In an embodiment, a small spin-Hall current may be continuously applied to the entire system to propagate the skyrmions through the tracks and logic gate junctions. This current magnitude is below the threshold required for skyrmions to traverse various notches throughout the system. At regular intervals, such as at time $t=1.0$ ns in the examples shown, a large spin-Hall current pulse is provided to the entire system to drive the skyrmions past the respective notches, which represents the global system clock that synchronizes the computing system. These notch synchronizers may be placed between logic gates, following the output of a logic gate, with the IN port of the notch synchronizer connected to an output port of a logic gate. The OUT port of the notch synchronizer is connected to an input port of a cascaded logic gate. Notches may be inserted between every logic gate input and output where synchronization is required, with each notch synchronizer handling zero or one skyrmion during each clock cycle. The notch synchronizers ensure that the skyrmions are synchronized with one another as they enter each logic gate, thereby providing proper skyrmion-skyrmion repulsion and logical functionality.

In an alternate embodiment, synchronization can be similarly achieved through clocked electrical control of the magnetic anisotropy. For such synchronization based on voltage control of magnetic anisotropy (VCMA), one gate voltage would be continuously applied to a region of a skyrmion track such that skyrmions would not be able to pass that region of the skyrmion track. At regular intervals, a different gate voltage would be applied to this region of the skyrmion track for a brief period of time, enabling a skyrmion to move past this region. In this clocking paradigm, a constant current would be continuously applied to the entire

10

system, and the different gate voltages would be periodically applied to the VCMA synchronizers simultaneously to represent the global clock.

FIG. 7 illustrates a cascaded one-bit full adder logic circuit with skyrmion implementation in accordance with an illustrative embodiment. In the present example, one-bit full adder 700 is constructed from the logic gates explained above. One-bit full adder 700 computes the binary sum and carry-out of two one-bit binary numbers A and B and a carry-in bit. Input ports 720 are connected to invert/copy logic gate 701 (B to CTRL and A to IN). Input ports 721 are connected to invert/copy logic gate 702 (A to CTRL and B to IN). The NOT output port of logic gate 701 and the NOT output port of logic gate 702 are merged at a first Y-junction 703. COPY2 output port of logic gate 702 is connected to a second Y-junction 707 through a synchronization notch structure at 704.

Output of Y-junction 703 connects to the input ports of invert/copy logic gate 705 through notch structure 704. C_{IN} input ports 711 and 712 are for the carry-in bit input to the adder. Track 711 connects to an IN port of another invert/copy logic gate 708 through synchronization notch structure 704, and synchronization notch structure 706. Track 712 connects to the CTRL port of invert/copy logic gate 705.

The COPY1 output port of logic gate 705 connects to the CTRL port of invert/copy logic gate 708 through synchronization notch structure 706. The COPY2 output port of logic gate 705 connects to Y-junction 707, which has an output of carry-out port C_{OUT} 731. The NOT output port of invert/copy logic gate 708 is connected to Y-junction 709. The NOT output of invert/copy logic gate 705 is also connected to Y-junction 709 through synchronization notch structure 706.

The output of Y-junction 709 is the SUM port 730. Full adder functions to sum the inputs A, B, and C_{IN} at the SUM port with the output carry bit registering at the C_{OUT} port.

Integrating the basic logic gates with the cascading and synchronization mechanisms enables the scaling of the reversible computing paradigm to large systems that efficiently perform complex functions. In an embodiment, the input A, B, and carry-in skyrmion signals interact as they propagate through the circuit to produce the sum and carry-out skyrmion signals, thus executing the one-bit full addition function with two half adders.

In an embodiment, a 150 ps-wide clock pulse is provided every 5 ns to synchronize the skyrmions to ensure proper conservative logic interactions within each component logic gate. The sum output may be produced within three clock cycles, while the carry-out output is produced within two clock cycles, and the carry-in to carry-out delay is only one clock cycle.

The clocked skyrmion signals may provide a natural means for pipelining, enabling the execution of n-bit addition within $n+2$ clock cycles. For example, though the 200 MHz clock frequency and the electrical current magnitudes used in simulation provide inferior efficiency as compared to conventional computing systems, the non-volatility and pipelining inspire a vision for highly-efficient computing with alternative materials and improvements in the Rashba coefficient and spin-Hall angle.

FIG. 8 illustrates a pipelined full adder logic circuit with skyrmion implementation in accordance with an illustrative embodiment. In the present example, pipelined full adder logic circuit 800 is constructed from clocked full adder structure 700. In an embodiment, the clocking scheme used in the full adder micromagnetic simulation is extended for five cycles to perform three separate full adder operations.

11

Input ports **820** are connected to invert/copy logic gate **801** (B to CTRL and A to IN). Input ports **821** are connected to invert/copy logic gate **802** (A to CTRL and B to IN).

The NOT output port of logic gate **801** and the NOT output port of logic gate **802** are merged at a first Y-junction **803** through a synchronization notch structure at **848** and **850**. COPY2 output port of logic gate **802** is connected to a second Y-junction **807** through a synchronization notch structure at **844**. Output of Y-junction **803** connects to the input ports of invert/copy logic gate **805** through notch structure **804**.

Input C_{IN} ports **811** and **812** are for the carry-in bit input to the adder. Track **811** connects to an IN port of another invert/copy logic gate **808** through synchronization notch structures **844**, **846**, **804**, **806**. Track **812** connects to the CTRL port of invert/copy logic gate **805** through synchronization notch structures **844**, **846**, and **804**.

The COPY1 output port of logic gate **805** connects to the CTRL port of invert/copy logic gate **808** through synchronization notch structure **806**. The COPY2 output port of logic gate **805** connects to Y-junction **807**, which connects to its output of carry-out port C_{OUT} **831** through synchronization notch structures **840**, **841**, and **842**. The NOT output of invert/copy logic gate **805** is also connected to Y-junction **809** through synchronization notch structure **806**.

The NOT output port of invert/copy logic gate **808** is also connected to Y-junction **809**. Y-junction **809** is connected to its output SUM port **830** through synchronization notch structures **841** and synchronization notch structure **842**.

In an example, the pipelined full adder logic circuit **800** may have an initial state of three-stage pipeline at $t=0$ ns with inputs of $A=B=C_{IN}=1$ for stage one, $A=1$ and $B=C_{IN}=0$ for stage two, and $A=0$ and $B=C_{IN}=1$ for stage three. The corresponding final state after five clock cycles at $t=24.8$ ns would have an output of $SUM=C_{OUT}=1$ for stage 1, $SUM=1$ and $C_{OUT}=0$ for stage two, and $SUM=0$ and $C_{OUT}=1$ for stage three.

As used herein, the phrase “a number” means one or more. The phrase “at least one of”, when used with a list of items, means different combinations of one or more of the listed items may be used, and only one of each item in the list may be needed. In other words, “at least one of” means any combination of items and number of items may be used from the list, but not all of the items in the list are required. The item may be a particular object, a thing, or a category.

For example, without limitation, “at least one of item A, item B, or item C” may include item A, item A and item B, or item C. This example also may include item A, item B, and item C or item B and item C. Of course, any combinations of these items may be present. In some illustrative examples, “at least one of” may be, for example, without limitation, two of item A; one of item B; and ten of item C; four of item B and seven of item C; or other suitable combinations.

The descriptions of the various embodiments of the present invention have been presented for purposes of illustration, but are not intended to be exhaustive nor is the present invention limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiment. The terminology used herein was chosen to best explain the principles of the embodiments, the practical application or technical improvement over technologies found in the marketplace, or to enable others of ordinary skill in the art to understand the embodiments disclosed here.

12

Further, different illustrative embodiments may provide different features as compared to other illustrative embodiments. The embodiment or embodiments selected are chosen and described in order to best explain the principles of the embodiments, the practical application, and to enable others of ordinary skill in the art to understand the disclosure for various embodiments with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A logic gate, comprising:

a first track configured for propagation of magnetic skyrmions;

a second track configured for propagation of magnetic skyrmions;

a junction linking the first and second tracks;

a continuous current flowing through the logic gate, wherein skyrmions propagate due to the current; and

additional tracks configured for propagation of magnetic skyrmions, wherein one or more additional junctions link together multiple tracks in the logic gate.

2. The logic gate of claim 1, wherein the logic gate is configured for both AND OR operations.

3. The logic gate of claim 1, wherein all input skyrmions entering the logic gate exit the logic gate as outputs.

4. The logic gate of claim 1, wherein skyrmions in the first track move through the junction to the second track due to a skyrmion-Hall force in a direction from the first track to the second track.

5. The logic gate of claim 4, wherein movement of skyrmions through the junction is prevented if other skyrmions are present at an intersection of the junction and second track, thereby producing skyrmion-skyrmion repulsion that counteracts the skyrmion-Hall force.

6. The logic gate of claim 1, wherein:

the presence of a skyrmion at a given time and place within the logic gate represents a binary 1; and

the absence of a skyrmion at a given time and place within the logic gate represents a binary 0.

7. The logic gate of claim 1, wherein the logic gate is configured to perform at least one of:

an inversion function;

a duplication function; or

a Fredkin gate function.

8. A logic circuit, comprising:

a number of skyrmion logic gates, wherein each skyrmion logic gate comprises:

three or more tracks configured for propagation of magnetic skyrmions;

at least two junctions linking multiple tracks together; and

a continuous current flowing through the logic gates, wherein skyrmions propagate due to the current, and wherein output skyrmions from logic gates in the circuit act as input skyrmions for other logic gates in the circuit.

9. The logic circuit of claim 8, wherein the logic gates in the logic circuit are configured to perform at least one of:

AND and OR operations;

an inversion function;

a duplication function; or

a Fredkin gate function.

13

10. The logic circuit of claim 8, further comprising a number of skyrmion synchronizers configured to synchronize skyrmions with one another as the skyrmions enter logic gates, thereby providing skyrmion-skyrmion repulsion at logic gate junctions to ensure logical functionality.

11. The logic circuit of claim 10, wherein each synchronizer comprises a notch or non-uniform material in a track configured to impede motion of the skyrmions.

12. The logic circuit of claim 11, wherein:
the continuous current flowing through the logic gates comprises a first strength that drives the skyrmions to the synchronizers but not past the synchronizers; and a periodic current pulse at a second, greater strength that permits the skyrmions to move past the synchronizers.

13. The logic circuit of claim 12, wherein the logic gates are pipelined wherein each logic gate performs a logical operation during a clock period between periodic applications of the current pulse at the second, greater strength.

14. The logic circuit of claim 10, wherein the skyrmion synchronizers use voltage gating to impede the motion of skyrmions.

15. The logic circuit of claim 14, wherein:
the continuous current flowing through the logic gates comprises a first strength that drives the skyrmions to the synchronizers but not past the synchronizers; and a periodic modulation of a gate voltage that permits the skyrmions to move past the synchronizers.

16. The logic circuit of claim 8, wherein the circuit is configured to perform a half adder or full adder logical function.

17. The logic circuit of claim 8, wherein all input skyrmions entering the logic gates exit the logic gates as outputs.

14

18. A method of performing logical computations with magnetic skyrmions, the method comprising:

inputting a number of magnetic skyrmions into a logic gate, wherein the logic gate comprises:

three or more tracks configured for propagation of magnetic skyrmions;

at least two junctions linking multiple tracks together; and

applying a continuous current through the logic gates, wherein the skyrmions propagate due to the current.

19. The method of claim 18, wherein the logic gate is configured for both AND and OR operations.

20. The method of claim 18, wherein:

the presence of a skyrmion at a given time and place within the logic gate represents a binary 1; and the absence of a skyrmion at a given time and place within the logic gate represents a binary 0.

21. The method of claim 18, wherein all input skyrmions entering the logic gate exit the logic gate as outputs.

22. The method of claim 18, wherein the logic gate is configured to perform at least one of:

an inversion function;
a duplication function; or
a Fredkin gate function.

23. The method of claim 18, further comprising feeding output skyrmions from the logic gate as inputs into another logic gate.

24. The method of claim 23, further comprising synchronizing skyrmions with one another as the skyrmions enter the logic gates.

* * * * *