SCIENTIFIC REPORTS

OPEN

Analog Approach to Constraint Satisfaction Enabled by Spin Orbit Torque Magnetic Tunnel Junctions

Received: 20 November 2017
Accepted: 9 April 2018
Published online: 02 May 2018

Parami Wijesinghe, Chamika Liyanagedera & Kaushik Roy

Boolean satisfiability (*k*-SAT) is an NP-complete (k $\geq$ 3) problem that constitute one of the hardest classes of constraint satisfaction problems. In this work, we provide a proof of concept hardware based analog k-SAT solver, that is built using Magnetic Tunnel Junctions (MTJs). The inherent physics of MTJs, enhanced by device level modifications, is harnessed here to emulate the intricate dynamics of an analog satisfiability (SAT) solver. In the presence of thermal noise, the MTJ based system can successfully solve Boolean satisfiability problems. Most importantly, our results exhibit that, the proposed MTJ based hardware SAT solver is capable of finding a solution to a significant fraction (at least 85%) of hard 3-SAT problems, within a time that has a polynomial relationship with the number of variables(<50).

The Boolean satisfiability problem investigates whether there exists an assignment for the input variables that satisfies a given Boolean formula. *k*-SAT is widely used in many practical applications including automated planning[1], test pattern generation[2], hardware model checking[3], software program testing[4] and timing analysis[5]. *k*-SAT problems are NP-complete ($k \geq 3$)[6,7]. *i.e.*, there are no known algorithms that can guarantee a solution for a SAT problem in polynomial time, making it extremely difficult to solve most satisfiability problems with reasonable computational resources. Numerous research efforts have been directed towards realizing improved SAT solvers[8–13], since a polynomial time solution to *k*-SAT implies efficient solutions to a large number of hard optimization problems. The standard conjunctive normal form (CNF) of any Boolean *k*-SAT problem with *N* variables can be written as

$$\mathcal{F} = (x_1 \bigcup x_2 \bigcup \overline{x}_3) \bigcap (x_2 \bigcup \overline{x}_1 \bigcup x_5) \bigcap \ldots (\overline{x}_4 \bigcup \overline{x}_5 \bigcup x_3) \quad (1)$$

where $x_i \in \{0, 1\}$ is a variable and each clause is the disjunction (OR, $\cup$) of k ($k = 3$ in this case) such variables or their negation ($\overline{x}_i$). The propositional formula $\mathcal{F}$ is a conjunction (AND, $\cap$) of M number of such clauses. The hardness of a SAT problem can be measured as the ratio between the number of clauses and the variables, known as the constraint density α_c (Supplementary section S1).

Analog computational approaches have recently demonstrated promising results in a diverse array of applications[14] including aforementioned constraint satisfaction[8,9,15]. A recent analog formulation of a *k*-SAT solver has demonstrated its potential on locating a solution for the Boolean satisfiability problem in polynomial continuous-time[8]. However, implementing this set of analog formulae using a digital computer will diminish the polynomial time benefits, due to varying computational complexities between different time steps. Also, a hardware implementation of this analog *k*-SAT solver[8] is not ideal[16], due to the exponential energy fluctuations in the system. Consequently, a Cellular Neural Network (CNN) based analog SAT solver with bounded variables was proposed[9], and it is more appealing for hardware implementations. Although this bounded system does not have polynomial time complexity, noise effects in the analog hardware can potentially reduce the long transient times[9] in the system, as we demonstrate in this work. The dynamics of this analog SAT solver, which is also the framework of our work, can be defined by the following set of equations[9].

$$\dot{s}_i(t) = \frac{ds_i(t)}{dt} = -s_i(t) + Af(s_i(t)) + \sum_m c_{mi}\, g(a_m(t)) \quad (2)$$

Purdue University, School of Electrical and Computer Engineering, West Lafayette, Indiana, 47907, USA. Correspondence and requests for materials should be addressed to P.W. (email: pwijesin@purdue.edu)

$$\dot{a}_m(t) = \frac{da_m(t)}{dt} = -a_m(t) + Bg(a_m(t)) - \sum_i c_{mi} f(s_i(t)) + 1 - k \tag{3}$$

Here the variable s_i represents the state of the i^{th} ($i = 1, 2, ..., N$) Boolean variable (x_i) and a_m represents the "satisfiedness" of the m^{th} ($m = 1, 2, ..., M$) clause of the Boolean function. C is the problem specific 'interconnection matrix' of size $M \times N$ (Supplementary section S1). The functions $f()$ and $g()$ are the thresholding functions applied on variables s_i and a_m, respectively, as follows

$$f(s_i) = \frac{1}{2}(|s_i + 1| - |s_i - 1|) \tag{4}$$

$$g(a_m) = \frac{1}{2}(1 + |a_m| - |1 - a_m|) \tag{5}$$

This system is explained in detail in the Supplementary section S1. It is mathematically shown[9] that these set of equations satisfy three theorems that demonstrate the properties of the model. The same theorems are used in Supplementary section S4, to show that our hardware SAT solver demonstrates the same properties. Following are the three theorems.

- Theorem 1: Variables s and a remain bounded.
- Theorem 2: Every k-SAT solution has a corresponding stable fixed point.
- Theorem 3: A stable fixed point always corresponds to a solution.

We present a hardware platform built on nano-scale spintronic devices, which can successfully emulate the behaviour of the aforementioned SAT solver. As a matter of fact, recent studies have demonstrated efficient hardware models that utilize the underlying device physics of nano-electronic structures to perform computationally intensive calculations[17–21]. In this work, each of the above differential equations is modeled by a single MTJ with an underlying heavy metal (*Ta*, *Pt*, etc.) layer. Our numerical results demonstrate that, the MTJ based SAT solver exhibits a polynomial dependency, between the number of variables and the real time for convergence, even for SAT problems that are known to be hardest to solve (note that this is not a mathematical proof that shows a guaranteed polynomial time complexity). We conjecture that, this is due to the non-deterministic nature of our system caused by the random thermal noise, and also due to the added complexities associated with MTJs.

Using the behaviour of Magnetic Tunnel Junctions for a SAT solver. The proposed hardware based SAT-solver is a collection of heavy metal-MTJ (HM-MTJ) structures, interfaced through simple CMOS peripheral circuitry as described in the next section. The device-circuit structure we propose is generic and can be adapted to solve a given k-SAT problem. The HM-MTJ structure is composed of two ferromagnetic layers called the Pinned Layer (PL) and the Free Layer (FL), separated by a thin tunneling oxide (*MgO*) layer and an HM under-layer (Fig. 1(c)). The PL magnetization direction ($\hat{p}$) is fixed and acts as a reference. In contrast, the magnetization direction $\hat{m}$ of the FL can be switched by passing a current through the HM under-layer, using the Spin Orbit Torque (SOT) phenomenon. Such a technique has emerged as an energy-efficient mechanism for magnetization reversal[22–24]. Furthermore, the three terminal structure of this MTJ with a heavy metal under layer is beneficial in this work due to the possibility of simultaneous read and write to an MTJ[25]. This is impossible to achieve in a two terminal MTJ structure that requires the write current to flow through the tunnel junction.

The resistance measured across the MTJ varies with the magnetization of the FL, and shows two stable states; high resistive (R_{AP}) anti-parallel (AP) state, and low resistive (R_P) parallel (P) state. Equations 6 and 7 depict how the resistance across the MTJ (R_{MTJ}) varies with the direction of the FL magnetization, where θ_{fp} is the angle between the directions of FL and PL magnetizations[26]. The magnetization reversal dynamics of an MTJ with an applied current is explained in the Supplementary section S2 using the Landau-Lifshitz-Gilbert-Slonczewski (LLGS) equations[27]. The speed of this magnetization reversal can be controlled by the magnitude of the current passing through the HM layer. However, due to the effect of random thermal noise on nano-scale magnets, the MTJ switching speed follows a Gaussian distribution.

$$R_{MTJ} = \left(\frac{1}{R_P}\left(cos\left(\frac{\theta_{fp}}{2}\right)\right)^2 + \frac{1}{R_{AP}}\left(sin\left(\frac{\theta_{fp}}{2}\right)\right)^2\right)^{-1} \tag{6}$$

$$\theta_{fp} = cos^{-1}(\hat{m} \cdot \hat{p}) \tag{7}$$

The resistance across an MTJ, R_{MTJ}, can be easily converted into a voltage by using a simple resistor divider circuit. In the proposed hardware implementation of the SAT solver, each s_i and a_m variable from equations 2 and 3 are represented using a single HM-MTJ structure. The state of these variables at a particular time instant are given by the resistance across the corresponding MTJ device. The couplings between the s and a variables (terms $\sum_m c_{mi}\, g(a_m(t))$ and $-\sum_i c_{mi} f(s_i(t))$ in equations 2 and 3) are mapped as currents through the HM layer using the interface circuitry explained in the next section.

In addition to the mathematical explanation that can be found in Supplementary section S4, we now intuitively explain how our MTJ based SAT solver mimics the system elaborated in equations (2–3). One main feature of this system is that, the current values of the variables depend on their previous states as well as some inputs.

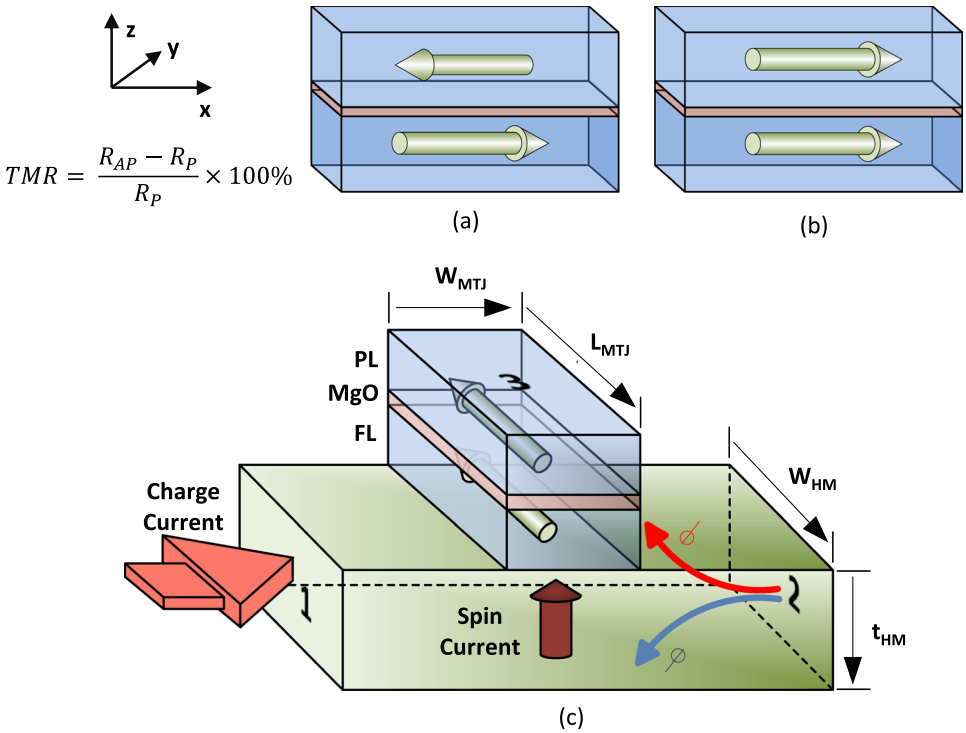

Figure 1. The Heavy Metal-Magnetic Tunnel Junction structure. (**a**) High resistive (R_{AP}) anti-parallel state of an MTJ (**b**) Low resistive (R_P) parallel state of an MTJ. The Tunnel Magneto-Resistance (TMR) is a measure of the normalized difference of these resistances. Typical values of the TMR ranges from 150%–600%[41,42]. (**c**) An HM-MTJ structure. The charge current through the HM layer underneath the MTJ, gets split into up and down spins, inducing a perpendicular spin current which can reverse the magnetization of the free layer through the Spin Orbit Torque phenomenon.

Similarly, the FL magnetization of an MTJ depends on its previous magnetization as well as the driving current. Another feature of the system in (2-3) is that, when the feedback from a_m towards the dynamics of s_i (*i.e.*, $\sum_m c_{mi}\, g(a_m(t_0))$) is zero after a particular time t_0, s_i will move towards $+A$ if $s_i(t_0) > 0$, and $-A$ if $s_i(t_0) < 0$, provided $A > 1$. Similarly, in an MTJ, an instantaneous removal of a current through the HM layer, will lead the free layer magnetization to settle down either to the parallel state or to the anti-parallel state. The state to which the magnet settles down is highly dependent upon the resistance it had at the time of removal of the current, in the absence of thermal noise. When the angle between the PL and FL magnetization directions $\theta_{fp} > \frac{\pi}{2}\left(\theta_{fp} < \frac{\pi}{2}\right)$, and if the drive current is zero, then the final FL magnetization will settle down to the anti-parallel (parallel) state. However, it should be noted that this phenomenon occurs under certain conditions. In this work, we optimized the FL thickness according to the following equation to exhibit the above behaviour.

$$t_{ss} = \frac{2K_i}{(N_{zz} - N_{yy})\mu_0 M_s^2} \tag{8}$$

where K_i is the energy density constant for interface perpendicular anisotropy and N_{zz}, N_{yy} are the demagnetization factors along y and z directions. The derivation of this FL thickness is explained in detail in the supplementary documentation (section S3). The new traversal of the magnetization of a device with a thickness of t_{ss} is illustrated in Fig. 2(a),(b). Figure 2(c),(d) depicts the magnetization traversal of an MTJ with a thickness larger than t_{ss} for reference. Note the smooth transition of the FL magnetization component along the easy axis (denoted as M_x) in Fig. 2(a) in contrast to the oscillatory transition of that in Fig. 2(c). In the light of this observation, we name t_{ss} as the seamless switching thickness of an MTJ.

Structure of the SAT solver. In this section, we will elaborate how we mapped the system in (2-3) to an array of HM-MTJs with a CMOS control interface. Each *s* and *a* variable in (2-3) is represented by the resistance of an MTJ. The resistance of the MTJ can be read as a voltage difference, when a constant read current (I_{READ}) flows through the MTJ. Note that this read current must be sufficiently small ($<1\mu A$) to not to interfere with the proper operation of the system. The functions $f()$ and $g()$ can be generated efficiently using a differential amplifier shown in Fig. 3(a). Note that the amplifier is connected to the bottom of the magnet (not the heavy metal layer). This is to avoid the small variable voltage (ΔV) induced across the HM layer due to the varying current that flows through it. These amplifiers will increase the voltage differences incurred due to the changing resistance of the MTJs. The state AP results in a larger voltage difference between nodes *A* and *B* with respect to that resulting from P state. In our structure, the AP and the P states of an MTJ that represents an *s* variable, gets mapped in to $+1$ and -1 states in equation (2) respectively. In a *k*-SAT problem,

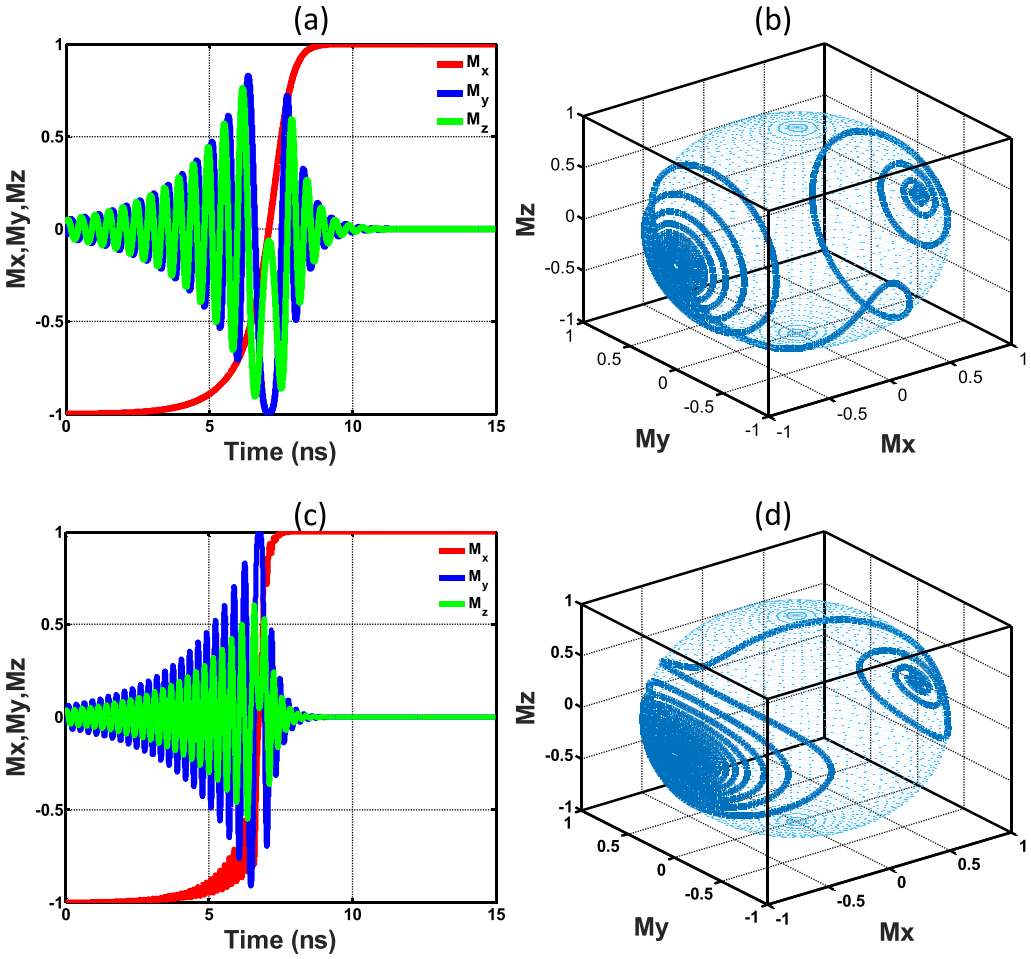

Figure 2. An MTJ changing its state from P to AP due to an applied current. (**a**) Time evolution of the components of unit magnetization vector in an MTJ with a FL thickness of t_{ss} (**b**) Unit magnetization traversal (in 3 dimensional space) of an MTJ with a FL thickness of t_{ss} (**c**) Time evolution of the components of unit magnetization vector in an MTJ with a FL thickness larger than t_{ss} (**d**) Unit magnetization traversal (in 3 dimensional space) of an MTJ with a FL thickness larger than t_{ss}. Notice the lack of oscillations in M_x in (**a**), during the magnetization reversal of the FL, in contrast to (**c**).

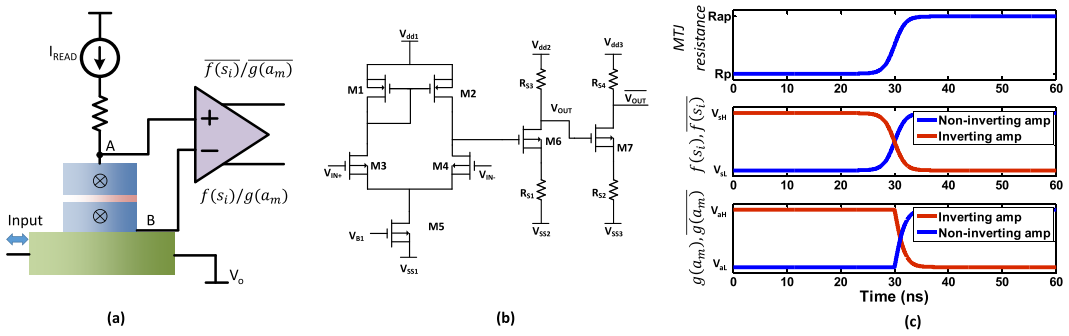

Figure 3. (**a**) The MTJ circuit for an *s*/*a* variable. The input current through the HM layer will change the state of the MTJ and this change will be measured by the amplifiers. The read current is assumed to be constant and its magnitude should be small so that it does not hinder the proper operation of the system. The MTJ resistance changes with the input current. The non-inverting output generates the 'state' ($f(s_i)/g(a_m)$) and the inverting output generates the 'inverse-state' ($\overline{f(s_i)}/\overline{g(a_m)}$) of an MTJ. (**b**) The reference circuit of the differential amplifier. V_{OUT} is the non-inverting output and $\overline{V_{OUT}}$ is the inverting output. (**c**) The outputs ($f(s_i)/g(a_m)$), varying with the MTJ resistance from R_P to R_{AP}.

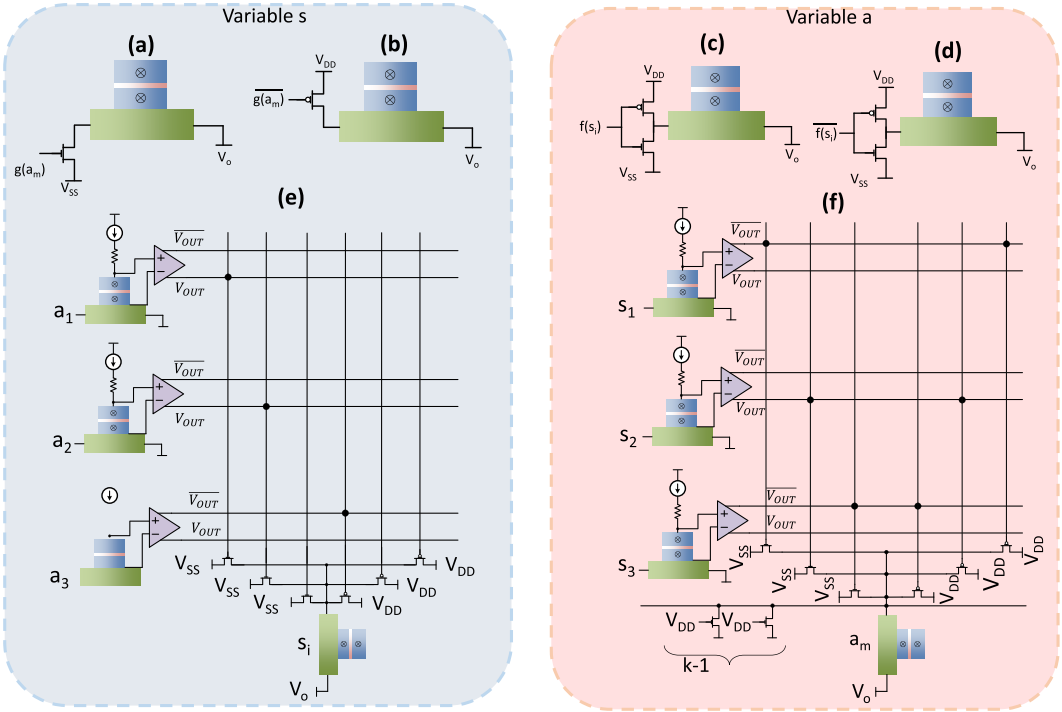

Figure 4. The input connection diagram of the SAT solver. The input connection to an s_i node from a_m, if (**a**) c_{mi} is negative (**b**) c_{mi} is positive. The input connection to an a_m node from s_i, if (**c**) c_{mi} is positive (**d**) c_{mi} is negative. Here the charge current from left to right through the HM layer drives the MTJ towards the AP state. The value of V_0 is smaller than V_{DD} but larger than V_{SS}. The sizing of the transistors must be done appropriately. (**e**) Outputs of three a nodes connected to an s_i node. The connection parameters (c_{1i}, c_{2i} and c_{3i}) between s_i and a_1, a_2 and a_3 are -1, -1 and 1, respectively. (**f**) Outputs of three s nodes connected to an a_m node. The connection parameters (c_{m1}, c_{m2} and c_{m3}) between a_m and s_1, s_2 and s_3 are -1, 1 and -1, respectively.

a variable can appear as x_i, or its negation ($\overline{x_i}$) in the m^{th} clause. This information is encoded in the elements of the connection matrix c_{mi}, as explained in Supplementary section S1. In order to account for different values of c_{mi} at the circuit level, we generate the 'state' ($f(s_i)/g(a_m)$) and the 'inverse-state' ($\overline{f(s_i)}$ /$\overline{g(a_m)}$) signals of an MTJ. These signals are produced at the amplification stage outputs, as shown in Fig. 3(b). A differential amplifier is employed to read the voltage difference across an MTJ. Additionally, a source degenerated common source amplifier is used as a second amplification stage, to boost the voltage to the desired levels. A third amplifier is employed in the design to generate the aforementioned inverse functions ($\overline{f()}$ and $\overline{g()}$). The complete schematic of the amplification stages used in this work is shown in Fig. 3(b). The same amplifier architecture with different control voltages was used for interfacing with MTJs representing both a and s variables. The outputs V_{OUT} ($f(s_i)$ or $g(a_m)$) and $\overline{V_{OUT}}$ ($\overline{f(s_i)}$ or $\overline{g(a_m)}$) are used to drive the MOSFETs controlling the current through the heavy metal layers (Fig. 4).

Figure 3 (c) elaborates how the above mentioned 'states' and 'inverse states' vary with the resistance of an MTJ. Each differential amplifier output will vary between a predefined high voltage (V_{sH}, V_{aH}), and a low voltage (V_{sL},V_{aL}). Therefore, the state -1 and $+1$ of variable s will be mapped to V_{sL} and V_{sH} at the non-inverting (V_{sH} and V_{sL} at the inverting) output. For the variable a, the non-inverting (inverting) output will be V_{aL} (V_{aH}) when the resistance of the MTJ is less than $(R_{ap}+R_p)/2$ and V_{aH} (V_{aL}) when the resistance is R_{ap}.

The term $\sum_m c_{mi}\, g(a_m(t))$ in equation 2, and the term $-\sum_i c_{mi} f(s_i(t)) + 1 - k$ in equation 3 (the coupling between variable s_i and a_m) are mapped as currents through the HM layers of MTJs, that represent s variables and a variables, respectively. At a particular time instant when the m^{th} clause is not satisfied, if the connection parameter c_{mi} is positive, the current should drive the MTJ that represents variable s_i towards the AP state. Similarly, when c_{mi} is negative, the current should drive that MTJ towards the P state, and when c_{mi} is zero, the current through the HM should be zero. Figure 4(a,b) graphically explains how this is realized at the circuit level. Figure 4(c,d) shows the circuit realization of the feedback from the s_i variable acting on the a_m variable, depending upon the connection parameter c_{mi}. When c_{mi} is positive (negative), $-c_{mi}f(s_i(t))$ should drive the MTJ that represents a_m towards the P (AP) state (for a case where s_i is in AP state). The two transistor structures (heavy metal current controllers) in Fig. 4(c,d) should provide an output voltage of V_o, when the input $f(s_i(t)) = \overline{f(s_i(t))} = (V_{DD} + V_{SS})/2$. This is to make sure that there is no current through the HM layer when $s_i(t) = 0$. For the Fig. 4, we assume that a charge current from left to right through an HM layer, drives the MTJ on top, towards the AP state. Figure 4(e,f) illustrates the final structure of the SAT solver with all the control logic. The connections in the 'network' depends on the SAT problem to be solved. Therefore, the connecting switches must be initialized depending upon the problem. Note that when the number of clauses of a problem increases, the connections become more complex.

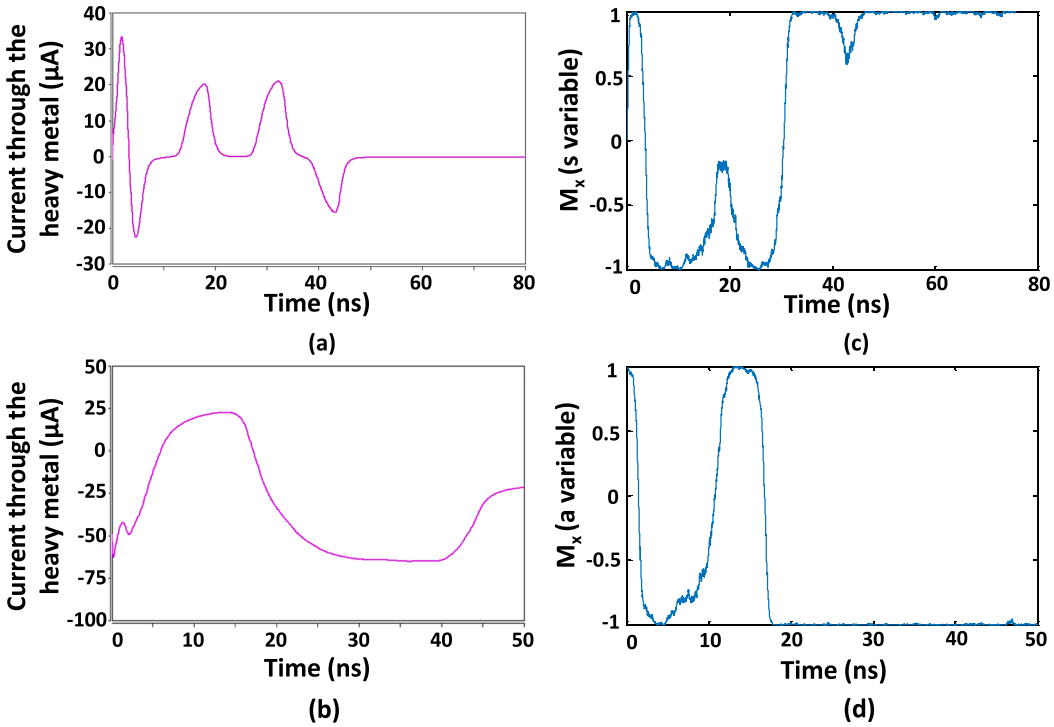

Figure 5. The varying current through the heavy metal layer of two MTJs that represent (**a**) *s* variable and (**b**) *a* variable of a 10-variable SAT instance ($\alpha_c = 4.25$). The currents were obtained via HSPICE following the circuits explained in Fig. 4, simulated in IBM 45 nm technology. The resultant time evolution of the free layer magnetization along the $\hat{x}$ direction is shown on right for (**c**) *s* variable and (**d**) *a* variable.

Results

In order to observe the functionality of our SAT solver, we conducted circuit level simulations. Figure 5 illustrates the currents through the heavy metal layers of the two MTJs representing *s* variable and *a* variable, that correspond to a 10-variable hard SAT instance. The results were obtained from HSPICE simulations using IBM 45 nm technology node. The resultant evolution of the free layer magnetization along the $\hat{x}$ direction (M_x) is shown on the right (Fig. 5(c) and (d)). Note that a positive current drives the MTJ towards AP (+1) state and a negative current drives the MTJ towards P (−1) state in the figure.

As elaborated in the Supplementary section S1, the constraint density (α_c) is an indicator of the hardness to solve a particular SAT instance. In order to observe the functionality of our solver for SAT instances with different hardness levels, we solved randomly generated 3-SAT problems with different constraint densities, and different number of variables. Figure 6 shows the magnetization dynamics of three MTJs that correspond to three variables in two 20 variable 3-SAT problems, each having a constraint density of 4.25 and 3.00, respectively. The colour of the trajectories in Fig. 6(c,d) indicates the normalized energy of the system at that particular point. This energy of the system can be defined by the following equations.

$$E(a, s) = \sum_{m=1}^{M} a_m K_m^2 \tag{9}$$

$$K_m = 2^{-k} \prod_{i=1}^{N} (1 - c_{mi} s_i) \tag{10}$$

where M and N are the number of clauses and the number of variables in the k-SAT problem, respectively. The energy is a function of the number of clauses not satisfied at a particular instant. This can be used as a cost function to determine the "satisfiedness" of a particular problem at a given instant. Notice that the trajectories in Fig. 6(c),(d) pass through higher energy states as the system tries to converge to a solution. This shows that our system escapes local minimum points naturally, unlike other algorithms[28] where simulated annealing is necessary to escape from such local minimum points.

Approximate polynomial time solution from the proposed SAT solver. We also solved randomly generated satisfiable 3-SAT problems in the hard regime ($\alpha_c = 4.25$) for different number of variables (20, 30, 40, 50). We have calculated the number of problems solvable within 10 μs for the purpose of illustration. However, since our system has no limit cycles owing to thermal noise (more details are available in the next section), we argue that our proposed method will probably reach a solution if sufficient time has been provided, given that

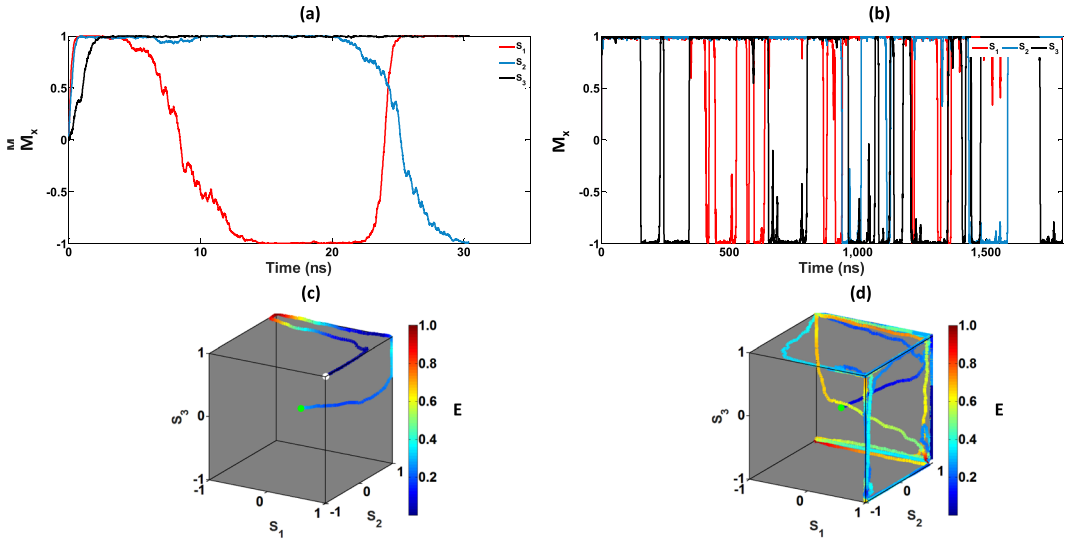

Figure 6. The time evolution of three variables in a 20 variable 3-SAT problem with different constraint densities. (**a**) and (**c**) correspond to a SAT problem with a constraint density $\alpha_c = 3$ whereas (**b**) and (**d**) correspond to a SAT problem with a constraint density $\alpha_c = 4.25$. (**c**) and (**d**) show the trajectories of the same 3 variables in (**a**) and (**b**) respectively, while converging to a solution inside a hypercube Q_3. The colour presents the energy of the system at a given state. The starting point is a green circle and the end point (solution) is the vertex with a white circle.

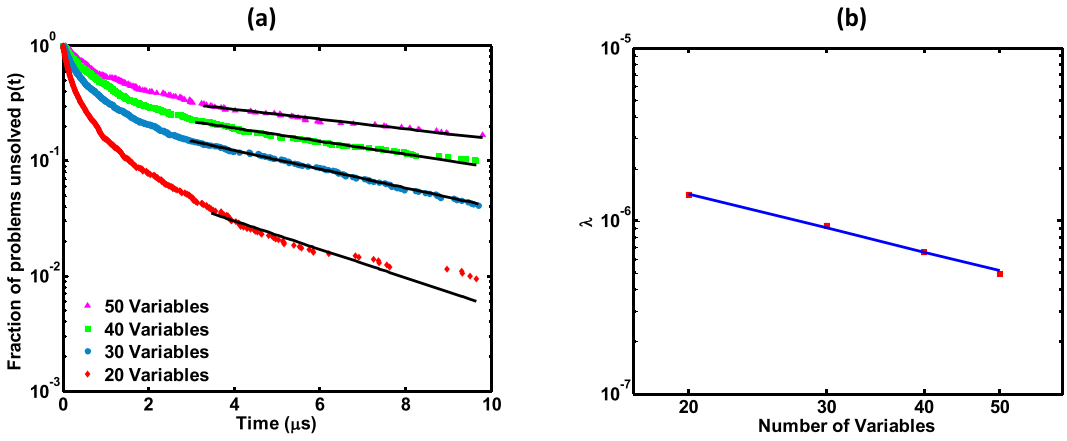

Figure 7. Computation time of the SAT solver (**a**) The fraction of problems $p(t)$ not yet solved at real time t, for 3-SAT problems with $\alpha_c = 4.25$, for $N = 20$, 30, 40 and 50. Averages were calculated with 10^3 instances for each N. (**b**) The decay rate λ for different number of variables. λ takes the form $\lambda(N) = bN^{-\beta}$ with β approximately 1.1 (note the log scale).

a solution exist. We monitored the fraction of problems not solved by the algorithm at time t and the result is depicted in Fig. 7. It is evident that the fraction of problems not solved $p(t)$, has an exponential decay with time t. The relationship between $p(t)$ and t can be approximated by

$$p(t) = re^{-\lambda(N)t+\gamma} \tag{11}$$

where r and γ are constants. The decay rate λ obeys $\lambda(N) = bN^{-\beta}$, with $\beta \approx 1.1$. Therefore the continuous time t needed to solve a $(1-p)$ fraction of problems can be written as

$$t(p, N) = (\gamma + \ln(r/p))b^{-1}N^{\beta} \tag{12}$$

This implies that the time to solve a $(1-p)$ fraction from a set of k-SAT problems is of polynomial complexity (for the range of N we have considered). This polynomial relationship still holds when the fraction of problems left unsolved is a fixed number (irrespective of the number of variables)[8]. That is, the time taken to solve all possible k-SAT formulae for a given N and α_c ($\Theta(k, N, \alpha_c)$), except for a constant amount of problems c ($p(t) = c/\Theta(k, N, \alpha_c)$), would follow a relationship as shown below when $N \to \infty$[8] (if we assume that the relationship in equation (11) and (12) still holds as $N \to \infty$).

$$t(p, N) \sim N^{\beta+1} ln(N) \tag{13}$$

A previous proposal[8] shows a similar relationship for the time required to solve k-SAT problems. However, the relationship is not valid in real time if a digital computer is to be used for the calculations. It is because, the complexity of the analog system varies over time, and when solving each step, it would take different real time values depending upon the complexity. Since our method is purely based on hardware, we argue that the above approximate polynomial time relationship is valid in real time for our proposed system. We conjecture that this behaviour is due to multiple reasons including the thermal noise associated with the MTJs. It has also been predicted[9] that, the noise effects may avoid long transient oscillations. Our results too suggest that higher amounts of noise leads to faster convergence (explained in the next section). It must be noted that, this proposed MTJ based SAT solver does not behave identical to the cellular neural network based solver in equations (2–3). Our solver has some added complexities not present in the CNN based system (refer to the set of equations in Supplementary section S4). We conjecture that such complexities offered by the device physics, acts favorable to give faster solutions to SAT problems as well. However, these benefits come at the cost of handling the circuit limitations (fan-out etc.) that can arise when solving problems with larger N.

Effect of thermal noise. Thermal noise has significant impact on the switching dynamics of nano-magnets. Equation 14 explains the renowned Brown's model[29] that captures the behaviour of thermal noise which can be used as a random magnetic field in the LLGS equations.

$$\overrightarrow{H_{Thermal}} = \overrightarrow{\varsigma}\sqrt{\frac{2\alpha k_B T}{|\gamma| M_s V}} \tag{14}$$

$\overrightarrow{\varsigma}$ is a vector with components that are zero mean, unity standard deviation, Gaussian random variables. V is the volume of the free layer, T is the temperature, and k_B is the Boltzmann's constant. The time discretization value dt must be included in the numerator when solving the equation numerically. The existence of thermal noise is mandatory for the proper operation of our SAT solver. This is due to the tilt of the FL magnetization with the easy axis, induced by thermal noise without which a magnet cannot be switched. Results indeed show that, under zero thermal noise, the magnetizations evolve to frozen non-solution states (Supplementary section S5). In the next two subsections, we will explain how the thermal noise assists in avoiding limit cycles and the impacts of larger thermal noise on the time to converge to a solution.

Not behaving as a chaotic dynamic system and absence of limit cycles. We define the states of all the MTJs that represent variable s as $H_N = [-1, 1]^N$. The P state is mapped to -1 and the AP state is mapped to $+1$. The boundary of H_N is the N hypercube Q_N, with vertices $V_N = \{-1, 1\}^N$. The solution space to a particular problem can be a subset of these V_N. Let us denote such a solution by $V_N^{sol} = \{V_N^i, V_N^j, \ldots\}$. Due to the effect of thermal noise, the solution to which the system will ultimately converge has minimal dependency with the initial states of the MTJs. For example, let us consider a SAT problem that has multiple solutions V_N^{sol} and solving it in two trials with the same initial states of MTJs. The output solutions in the two trials may not be the same even though the starting conditions were identical. This implies that our system does not show any chaotic behaviour (it is not deterministic) in contrast to the system given by equations 2 and 3.

If the states of the MTJs that represent variable s continuously change in a periodic manner (it has entered a limit cycle), it is possible that the system never reaches a solution (even if there exists one). That is, $s_i(t) = s_i(t + nT)$ for $\forall n = 1, 2, \ldots$ and $s_i(t)$ is not a solution of the system. This is known as the system getting trapped in a limit cycle[8]. It is shown that for the system explained in equations 2–3, this can occur for certain coupling parameter (A, B) choices[9]. However, due to the added stochasticity from the thermal noise, we argue that our MTJ based SAT solver does not get trapped in limit cycles. It is highly probable that our system reaches a solution if sufficient amount of time is provided.

Increasing temperature resulting in faster solution convergence. Now let us consider how different amounts of thermal noise will affect the operation of our MTJ based SAT solver. Temperature can affect the amount of thermal noise applied on an MTJ device (equation 14). Sufficiently higher temperatures on MTJs with smaller switching energy barriers, can cause the state of an MTJ to oscillate over time, even without any input current or magnetic field[25]. We solved randomly generated 20 variable, 3-SAT problems at different temperatures and observed the percentage of problems that can be solved within 10 μs. As Fig. 8(a) illustrates, it is evident that in the range of $20°C - 130°C$, there is no significant degradation in the percentage of problems solved within 10 μs. However, according to Fig. 8(b), it appears that the average time to solve a k-SAT problem decreases by $\sim 100\, ns$ with increasing temperature in the range $20°C$–$130°C$.

Effects of process variations. In our design, we selected a particular thickness (t_{ss}) for the free layers of the MTJs. In reality, it is impossible to achieve the exact thickness due to number of reasons including atomic limitations, process variations, fabrication limitations *etc*. How much precession is present along the easy axis is dependent upon how much the actual thickness has deviated from t_{ss}. In order to observe the effects of thickness variations on the operation of our proposed SAT solver, we consider two scenarios.

1. Global variation of thickness.
2. Local variation of thickness.

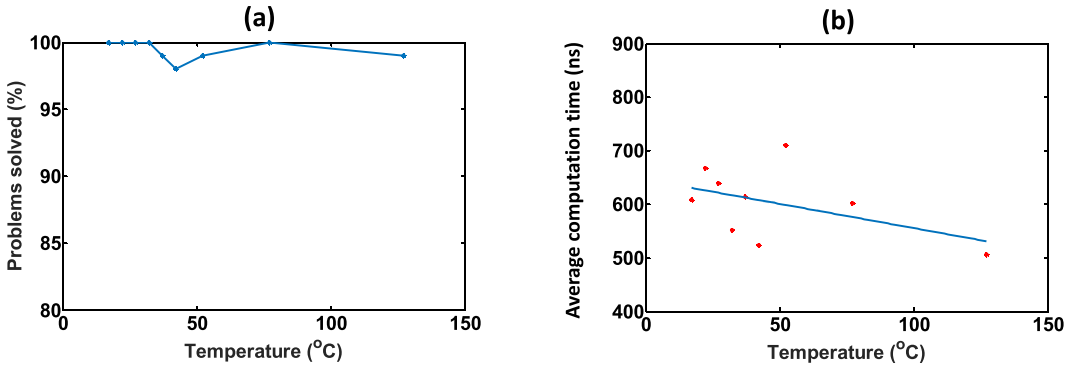

Figure 8. (**a**) The percentage of randomly generated 20 variable SAT problems ($\alpha_c = 4.25$) solved at different temperatures within 10 μs. (**b**) Average time to solve a 20 variable SAT problem at different temperatures.

In the first case, we perturb all the thicknesses of MTJs in our system by some constant percentage from t_{ss}. For a particular percentage global variation in thickness, we solved randomly generated 20 variable, SAT problems. Then the percentage of problems solvable within 10 μs, and the average time to solve a single problem was observed. Figure 9(a) illustrates that there is no significant change in the percentage of solvable problems when the global variations in thickness is changed from −5% to +10%. However, as Fig. 9(b) shows, the average convergence time increases when the deviations in thickness increases. We also observed that the solver no longer works if the thickness is less than a particular value. This is the limit at which the perpendicular magnetic anisotropy (PMA) becomes dominant and the FL magnetization stabilizes in $\hat{z}$ axis instead of $\hat{x}$ axis. We observed this when the actual thickness is ∼−10% deviated from the t_{ss}, for the choice of materials and dimensions used in this work. In the second case, we change thicknesses of all the MTJs according to a Gaussian distribution with a 3σ value (where σ is the standard deviation) of 10% from the t_{ss}. The solver gave an average convergence time of 588.31ns and was able to solve 97% of randomly generated 20 variable, 3-SAT instances within 10 μs.

As described in the previous section, the temperature increments inversely affect the computation time of the MTJ based system. To see the effect of both thickness and temperature variations simultaneously, we conducted the above experiment at different temperatures. The results are summarized in Fig. 9(c) and it shows that the changes in computation time due to thickness deviation is more prominent than that due to temperature drift.

During the fabrication process, other non-idealities such as edge damages[30,31] can be present in the MTJs. We introduced variations in interface anisotropy constant, width and length of the free layer of MTJs, to observe the effect of aforementioned non-idealities. The effects of anisotropy constant variations on computation time follow a similar trend as the 'computation time vs thickness' curve (Fig. 9). We further noticed that the computation time decreases when the free layer dimensions decrease. The percentage of problems solved remained almost at 100%. More detailed results are included in the Supplementary section S6.

Power consumption and computation time of the SAT solver. In this section, we will present the power consumption and computation time of our proposed system, and compare with existing methodologies. In order to calculate the power consumption, we used SPICE simulations in IBM 45 nm technology. The measured average power consumption over solving 1,000, 20-variable (our software based LLGS solving framework could not handle bigger benchmarks such as 'ais8'[32]) hard SAT problems (constraint density $\alpha_c = 4.25$) was 1.37 mW. The power requirements of each section of the solver are presented in Table 1. We observed that the peripheral circuits such as amplifiers and voltage controlled current drivers consume a significant portion of power. The power consumption of the total MTJ and heavy metal layer structures is approximately about 20% of the total power consumption of the system. Similarly, a recent work of a hardware realization[16] of an analog approach[8], also explains that their overall power consumption to be significant (numerical value not specified), due to the usage of op-amps. In another design[15], the CNN based SAT solving algorithm[9] used in our work was realized using op-amp based integrators. The power consumption was reported as 140 μW for a 4-variable, 4-clause problem. For comparison, we evaluated the average power consumption of our MTJ based solver for similar 4-variable 4-clause SAT instances, and witnessed a power consumption of 84 μW, which is about 40% smaller than the aforementioned analog hardware design[15].

Digital hardware for realizing typical SAT solving algorithms such as GRASP[11], DPLL[33] *etc.* can be found in literature[34–36]. A custom IC[34] designed for such an algorithm reported a power consumption of 871 μW. This is about 37% lower than our design. However, these digital approaches are slower than analog solvers due to the step-by-step decision making and backtracking required for solving a problem. It is also noteworthy that our proposal is an asynchronous method in contrast to the above synchronous methods. Therefore, the need for a clock signal and the power associated with it is not present in our design. Our system automatically goes to a minimum energy point (which is a solution) and stabilizes there as explained mathematically in Supplementary section S4.

In order to obtain the computation time of our system, we conducted system level simulations using random 1,000, 20-variable hard SAT instances. The time taken to solve all 1,000 problems (100%) were evaluated and the average computation time per instance was 553 ns. This computation time was then compared with a state-of-the-art software SAT solver, Minisat[37]. The same 1,000 problems were solved using Minisat in a 3.6 GHz processor and the average computation time was evaluated (1.44 ms). We observed an average speed-up

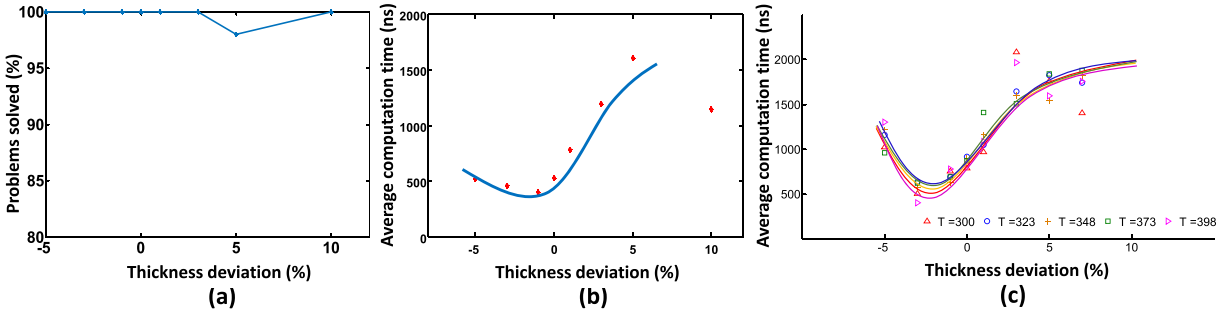

Figure 9. (**a**) The percentage of randomly generated 20 variable SAT problems ($\alpha_c = 4.25$) solved at different percentage variations in thickness (from the seamless switching thickness, t_{ss}) of the free layer within 10 μs. (**b**) Average time to solve a 20 variable SAT problem at different percentage variations in thickness of the free layer. The variations are considered as global. *i.e.*, all the devices in the SAT solver has the thickness with same deviation from t_{ss} (**c**) Same experiment in (**b**) performed for different temperatures. Temperature indicated in Kelvin.

Component	Power consumption
Read circuitry	300 nW
Amplifiers	315 μW
Voltage controlled HM current drivers	827 μW
Driving current through HM	235 μW
Total	1377 μW

Table 1. Power consumption of the MTJ based SAT solver.

Hardware solver	Speed-up with respect to software based solvers
Reconfigurable SAT solver[35]	90 (3.6 GHz processor)
BCP accelerator[38]	6.7 (3.6 GHz processor)
BCP accelerator[36]	4 (3.3 GHz processor)
MTJ based solver	2.6×10^3 (3.6 GHz processor)

Table 2. Speed-up of hardware based SAT solvers with respect to purely software based solvers.

of 2.6×10^3 with respect to Minisat. Prior hardware SAT solver designs in literature have also presented their speed-up with respect to Minisat, and the values are summarized in Table 2 for comparison. Note that the Boolean Constraint Propagation (BCP) accelerator[38] in Table 2 has reported the speed up with respect to a purely software based algorithm (modified zChaff[10]) which has the same performance for BCP when compared with Minisat. Furthermore, the analog CNN based system[9] built using op-amp based integrators[15] has an average computation time of 15 μs for 10 variable, 2-SAT problems. Our MTJ based proposal has an average run time (over 1,000 instances) of 186 ns for 10 variable 3-sat ($\alpha_c = 4.25$) problems (showing that the MTJ based solver is $\sim 30\times$ faster).

Conclusion

Boolean satisfiability is an NP-complete problem ($k \geq 3$) that finds utility in vast array of applications[1–5]. Analog solutions to the satisfiability problem has recently appeared attractive[8,9,15] due to the massive parallelism available when solving, in contrast to the stepwise search algorithms. In this work, we provide a proof of concept hardware based analog SAT-solver using Magnetic Tunnel Junctions driven by the Spin Orbit Torque Phenomenon. We have mathematically shown how the inherent device physics of MTJs closely mimics an existing analog approach[9] to solving the Boolean satisfiability problem. Device and circuit level simulations were conducted to solve hard satisfiability problems in order to observe the performance and functionality of our proposed system. According to the observations, we witnessed that the proposed SAT solver is capable of finding a solution to a significant fraction (>85%) of hard SAT problems in polynomial time. We conjecture that this is due to the inherent thermal noise present in MTJs and the device complexities added on top of the existing analog approach[9]. The SAT solver automatically comes out of local minimum points and limit cycles due to thermal noise. Therefore, it is highly probable that the system reaches a solution if sufficient time is provided, given that the SAT problem has a solution. Further, the variation analysis illustrates that our proposed solver is robust to variations in the MTJ thickness in the range of -5% to 10%. Larger variations result in higher average convergence time. The proposed MTJ based SAT solver is 2.6×10^3 times faster than a state-of-the-art software solver, Minisat.

Methods

The set of equations involved in modelling the HM-MTJ structures is provided in Supplementary sections S2, S3 and S4. The t_{ss} of the MTJs was calculated by self consistently solving the equation (8) and the analytical equations for the demagnetization factors[39]. IBM 45 nm technology node was used to simulate the CMOS interface circuitry. The material parameters (selected according to the experimental papers[24,40]) and the device dimensions are summarized in supplementary Table S1.

References

1. Bao, F. S. *et al.* Accelerating boolean satisfiability (sat) solving by common subclause elimination. *Artificial Intelligence Review* 1–15 (2017).
2. Saha, S. *et al.* Improved test pattern generation for hardware trojan detection using genetic algorithm and boolean satisfiability. In *International Workshop on Cryptographic Hardware and Embedded Systems*, 577–596 (Springer, 2015).
3. Vizel, Y., Weissenbacher, G. & Malik, S. Boolean satisfiability solvers and their applications in model checking. *Proceedings of the IEEE* **103**, 2021–2035 (2015).
4. Stanley, J., Liao, H. & Lafortune, S. Sat-based control of concurrent software for deadlock avoidance. *IEEE Transactions on Automatic Control* **60**, 3269–3274 (2015).
5. Chung, Y.-T. & Jiang, J.-H. *Functional timing analysis method for circuit timing verification US Patent 8*, **671**, 375 (2014).
6. Karp, R. M. Reducibility among combinatorial problems. In *Complexity of computer computations*, 85–103 (Springer, 1972).
7. Cook, S. A. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, 151–158 (ACM, 1971).
8. Ercsey-Ravasz, M. & Toroczkai, Z. Optimization hardness as transient chaos in an analog approach to constraint satisfaction. *Nature Physics* **7**, 966–970 (2011).
9. Molnár, B. & Ercsey-Ravasz, M. Asymmetric continuous-time neural networks without local traps for solving constraint satisfaction problems. *PloS one* **8**, e73400 (2013).
10. Moskewicz, M. W. *et al.* Chaff: Engineering an efficient sat solver. In *Proceedings of the 38th annual Design Automation Conference*, 530–535 (ACM, 2001).
11. Marques-Silva, J. P. & Sakallah, K. A. Grasp: A search algorithm for propositional satisfiability. *IEEE Transactions on Computers* **48**, 506–521 (1999).
12. Paturi, R., Pudlák, P., Saks, M. E. & Zane, F. An improved exponential-time algorithm for k-sat. *Journal of the ACM (JACM)* **52**, 337–364 (2005).
13. Guo, L., Hamadi, Y., Jabbour, S. & Sais, L. Diversification and intensification in parallel sat solving. *Principles and Practice of Constraint Programming–CP* 2010 252–265 (2010).
14. Siegelmann, H. T. Computation beyond the turing limit. In *Neural Networks and Analog Computation*, 153–164 (Springer, 1999).
15. Basford, D. A. *et al.* The impact of analog computational error on an analog boolean satisfiability solver. In *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, 2503–2506 (IEEE, 2016).
16. Yin, X. *et al.* Efficient analog circuits for boolean satisfiability. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **26**, 155–167 (2018).
17. Fan, D., Sharad, M., Sengupta, A. & Roy, K. Hierarchical temporal memory based on spin-neurons and resistive memory for energy-efficient brain-inspired computing. *IEEE transactions on neural networks and learning systems* **27**, 1907–1919 (2016).
18. Sengupta, A. *et al.* Magnetic tunnel junction mimics stochastic cortical spiking neurons. *Scientific reports* **6**, 30039 (2016).
19. Sengupta, A., Shim, Y. & Roy, K. Proposal for an all-spin artificial neural network: Emulating neural and synaptic functionalities through domain wall motion in ferromagnets. *IEEE transactions on biomedical circuits and systems* **10**, 1152–1160 (2016).
20. Behin-Aein, B., Datta, D., Salahuddin, S. & Datta, S. Proposal for an all-spin logic device with built-in memory. *Nature nanotechnology* **5**, 266–270 (2010).
21. Wijesinghe, P., Ankit, A., Sengupta, A. & Roy, K. An all-memristor deep spiking neural computing system: A step towards realizing the low power, stochastic brain. *arXiv preprint arXiv:1712.01472v3* (2017).
22. Yu, G. *et al.* Switching of perpendicular magnetization by spin-orbit torques in the absence of external magnetic fields. *Nature nanotechnology* **9**, 548–554 (2014).
23. Liu, L. *et al.* Spin-torque switching with the giant spin hall effect of tantalum. *Science* **336**, 555–558 (2012).
24. Pai, C.-F. *et al.* Spin transfer torque devices utilizing the giant spin hall effect of tungsten. *Applied Physics Letters* **101**, 122404 (2012).
25. Liyanagedera, C. M., Sengupta, A., Jaiswal, A. & Roy, K. Stochastic spiking neural networks enabled by magnetic tunnel junctions: From nontelegraphic to telegraphic switching regimes. *Physical Review Applied* **8**, 064017 (2017).
26. Fong, X. *et al.* Knack: A hybrid spin-charge mixed-mode simulator for evaluating different genres of spin-transfer torque mram bit-cells. In *Simulation of Semiconductor Processes and Devices (SISPAD), 2011 International Conference on*, 51–54 (IEEE, 2011).
27. Slonczewski, J. C. Conductance and exchange coupling of two ferromagnets separated by a tunneling barrier. *Physical Review B* **39**, 6995 (1989).
28. Jonke, Z., Habenschuss, S. & Maass, W. Solving constraint satisfaction problems with networks of spiking neurons. *Frontiers in neuroscience* **10** (2016).
29. Brown, W. F. Jr. Thermal fluctuations of a single-domain particle. *Physical Review* **130**, 1677 (1963).
30. Song, K. & Lee, K.-J. Spin-transfer-torque efficiency enhanced by edge-damage of perpendicular magnetic random access memories. *Journal of Applied Physics* **118**, 053912 (2015).
31. Sun, Z., Retterer, S. & Li, D. The influence of ion-milling damage to magnetic properties of co80pt20 patterned perpendicular media. *Journal of Physics D: Applied Physics* **47**, 105001 (2014).
32. Hoos, H. H. & Stützle, T. Satlib: An online resource for research on sat. *Sat* **2000**, 283–292 (2000).
33. Davis, M., Logemann, G. & Loveland, D. A machine program for theorem-proving. *Communications of the ACM* **5**, 394–397 (1962).
34. Gulati, K. & Khatri, S. P. Accelerating boolean satisfiability on a custom ic. In *Hardware Acceleration of EDA Algorithms*, 33–61 (Springer, 2010).
35. Gulati, K., Paul, S., Khatri, S. P., Patil, S. & Jas, A. Fpga-based hardware acceleration for boolean satisfiability. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* **14**, 33 (2009).
36. Thong, J. & Nicolici, N. Fpga acceleration of enhanced boolean constraint propagation for sat solvers. In *Proceedings of the International Conference on Computer-Aided Design*, 234–241 (IEEE Press, 2013).
37. Eén, N. & Sörensson, N. Minisat 2.2. http://minisat.se (2013).
38. Davis, J. D., Tan, Z., Yu, F. & Zhang, L. A practical reconfigurable hardware accelerator for boolean satisfiability solvers. In *Design Automation Conference, 2008. DAC 2008. 45th ACM/IEEE*, 780–785 (IEEE, 2008).
39. Aharoni, A. Demagnetizing factors for rectangular ferromagnetic prisms. *Journal of applied physics* **83**, 3432–3434 (1998).
40. Ikeda, S. *et al.* A perpendicular-anisotropy cofeb–mgo magnetic tunnel junction. *Nature materials* **9**, 721–724 (2010).
41. Sakuraba, Y. *et al.* Co-concentration dependence of half-metallic properties in co–mn–si epitaxial films. *Applied Physics Letters* **96**, 092511 (2010).
42. Yuasa, S. & Djayaprawira, D. Giant tunnel magnetoresistance in magnetic tunnel junctions with a crystalline mgo (0 0 1) barrier. *Journal of Physics D: Applied Physics* **40**, R337 (2007).

Acknowledgements
The authors would like to thank A. Jaiswal of Purdue University for his help while discussing the ideas, and for critical reading of the manuscript. The work was supported in part by, Center for Brain Inspired Computing (C-BRIC), a MARCO and DARPA sponsored JUMP center, by the Semiconductor Research Corporation, the National Science Foundation, Intel Corporation and by the Vannevar Bush Fellowship.

Author Contributions
C.L. built the models for HM-MTJs. P.W. implemented the SAT solver using the MTJ models and performed relevant simulations. All authors assisted in discussing the ideas results and writing the manuscript.

Additional Information
Supplementary information accompanies this paper at https://doi.org/10.1038/s41598-018-24877-z.

Competing Interests: The authors declare no competing interests.

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