Energy Efficiency of Computation in All-spin Logic: Projections and Fundamental Limits

Zongya Chen

Follow this and additional works at: https://scholarworks.umass.edu/masters_theses_2

Part of the Electrical and Electronics Commons, Electronic Devices and Semiconductor Manufacturing Commons, and the Power and Energy Commons

Recommended Citation
https://scholarworks.umass.edu/masters_theses_2/754

This Open Access Thesis is brought to you for free and open access by the Dissertations and Theses at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Masters Theses by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact scholarworks@library.umass.edu.
ENERGY EFFICIENCY OF COMPUTATION IN ALL-SPIN LOGIC: PROJECTIONS AND FUNDAMENTAL LIMITS

A Thesis Presented
by
ZONGLYA CHEN

Submitted to the Graduate School of the University of Massachusetts Amherst in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE IN ELECTRICAL AND COMPUTER ENGINEERING

February 2019
Electrical and Computer Engineering
ABSTRACT

ENERGY EFFICIENCY OF COMPUTATION IN ALL-SPIN LOGIC: PROJECTIONS AND FUNDAMENTAL LIMITS

FEBRUARY 2019

ZONGYA CHEN
B.Sc., HEBEI UNIVERSITY
M.S.E.C.E., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Neal G. Anderson

Built with nanomagnets, a spintronic device called the all-spin logic (ASL) device carries information with only spin currents, resulting in a low power supply—10 mV. This voltage is 100 times smaller than the conventional CMOS devices (usually 0.8~1V). The potential for improved energy efficiency made possible by the low operating voltage of ASL makes it one of the most promising devices among its post-CMOS competitors.

The basic working principles of ASL device are introduced in this thesis and two complementary approaches to studying energy efficiency of computation are applied to a common set of ASL circuits: (1) a circuit simulation approach that provides efficiency estimates for specific ASL circuit realizations, and (2) a physical-information-theoretic approach that reveals fundamental efficiency bounds for ASL circuits as limited by irreversible information loss.
The results of this study support the expectation that the energy efficiency of computation in ASL can far exceed that of CMOS. However, it also reveals that ASL efficiencies—shown to exceed fundamental limits by many orders of magnitude in the ASL implementations studied here—are unlikely to approach fundamental limits because of the unavoidable energetic overhead cost of maintaining spin currents.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 From CMOS to All-spin Logic</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Energy Dissipation in ASL</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Thesis Objectives</td>
<td>6</td>
</tr>
<tr>
<td>1.4 Thesis Overview</td>
<td>7</td>
</tr>
<tr>
<td>2. ALL-SPIN LOGIC DEVICE: WORKING PRINCIPLES AND BASIC UNITS</td>
<td>9</td>
</tr>
<tr>
<td>2.1 Working Principles of ASL Devices</td>
<td>9</td>
</tr>
<tr>
<td>2.2 Digital Computing with ASL</td>
<td>12</td>
</tr>
<tr>
<td>3. ENERGY EFFICIENCY OF COMPUTATION IN ASL: THEORETICAL METHODS</td>
<td>20</td>
</tr>
<tr>
<td>3.1 Energy Efficiency From Circuit Simulation</td>
<td>20</td>
</tr>
<tr>
<td>3.1.1 Landau-Lifshitz-Gilbert Equation and spin storage</td>
<td>20</td>
</tr>
<tr>
<td>3.1.2 Spin Injection</td>
<td>23</td>
</tr>
<tr>
<td>3.1.3 Spin Transport</td>
<td>26</td>
</tr>
<tr>
<td>3.1.4 Energy Efficiency</td>
<td>31</td>
</tr>
<tr>
<td>3.2 Efficiency Limits from Physical Information Theory</td>
<td>31</td>
</tr>
<tr>
<td>3.2.1 Introduction of Physical Information Theory</td>
<td>32</td>
</tr>
<tr>
<td>3.2.2 Physical Decomposition</td>
<td>34</td>
</tr>
</tbody>
</table>
3.2.3 Process Abstraction ................................. 37
3.2.4 Operational Decomposition .......................... 37
3.2.5 Cost Analysis ........................................ 39
3.2.6 Energy Efficiency .................................... 42

4. ASL CIRCUIT STUDY: PRELIMINARY RESULTS AND DISCUSSION ........................................... 44

4.1 Simple Circuits: Buffer, Inverter and Latch ......................... 44
  4.1.1 Buffer Simulation Results ............................ 44
  4.1.2 Inverter .............................................. 49
  4.1.3 Latch ............................................... 49

4.2 Simplified Circuits Simulation Results: Buffer, Inverter, Latch ....... 51
  4.2.1 From Dynamic and Static Power to Projected Energy ............ 51
  4.2.2 Simplified Buffer and Inverter Simulation ..................... 52
  4.2.3 Simplified Latch Simulation ............................. 53
  4.2.4 Comparison of the full-LLG Model and Simplified Model ....... 55

4.3 Simulation of Larger Circuits: Majority Gate, Half-adder, and ALU ................................. 55
  4.3.1 Majority Gate ....................................... 57
  4.3.2 Half Adder .......................................... 58
  4.3.3 Arithmetic Logic Unit ................................ 61

4.4 Energy Efficiency Bounds for ASL Circuits from Physical Information Theory ........................................ 67
  4.4.1 From Projected Energy to Irreversibility Induced Energy ....... 67
  4.4.2 Efficiency Bounds for ASL Circuits: Detailed Application Example ........................................ 69
  4.4.3 Efficiency Bounds for Buffer, Inverter, Latch, Majority Gate, and Half Adder .............................. 77
  4.4.4 Efficiency Bounds for ALU ................................ 79
  4.4.5 Simulation-Based Projections vs Fundamental Bounds .......... 83

5. SUMMARY AND CONCLUSIONS ........................................... 89

APPENDICES

A. LANDAU-LIFSHTIZ-GILBERT EQUATION DERIVATION ........... 91
B. SPIN INJECTION WITH LLG EQUATION ......................... 95
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>The energy cost and throughput of ASL vs CMOS FFT processor.</td>
</tr>
<tr>
<td>2.1</td>
<td>The truth table of three-input majority gate.</td>
</tr>
<tr>
<td>2.2</td>
<td>The truth table of full adder with the illustration of xor/xnor gate design.</td>
</tr>
<tr>
<td>4.1</td>
<td>ASL simulation parameters.</td>
</tr>
<tr>
<td>4.2</td>
<td>The energy cost of different circuits with and without LLG model.</td>
</tr>
<tr>
<td>4.3</td>
<td>The truth table of half adder with the illustration of majority-gate nand gate design.</td>
</tr>
<tr>
<td>4.4</td>
<td>ALU function table.</td>
</tr>
<tr>
<td>4.5</td>
<td>Projected energy results of different ASL devices.</td>
</tr>
<tr>
<td>4.6</td>
<td>State transformation for the ASL adder.</td>
</tr>
<tr>
<td>4.7</td>
<td>Inputs and outputs of the first adder.</td>
</tr>
<tr>
<td>4.8</td>
<td>Outputs of the first adder.</td>
</tr>
<tr>
<td>4.9</td>
<td>Inputs and outputs of the second adder.</td>
</tr>
<tr>
<td>4.10</td>
<td>Outputs of the second adder.</td>
</tr>
<tr>
<td>4.11</td>
<td>Outputs of buffer, inverter and latch.</td>
</tr>
<tr>
<td>4.12</td>
<td>Inputs and outputs of the majority gate.</td>
</tr>
<tr>
<td>4.13</td>
<td>Inputs and outputs of the half adder middle stage.</td>
</tr>
<tr>
<td>4.14</td>
<td>Inputs and outputs of the half adder middle stage.</td>
</tr>
</tbody>
</table>
4.15 State transformation for the buffer, inverter, latch, majority gate, half adder, and ALU. ............................................................. 84

4.16 The truth table of subtractor. .................................................. 84

4.17 Truth table of increment and decrement function. ................. 85

4.18 Irreversibility induced energy results of different ASL devices. ........ 85
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A lab environment fabricated ASL buffer/inverter. 3</td>
</tr>
<tr>
<td>1.2</td>
<td>A 3-input majority gate. 4</td>
</tr>
<tr>
<td>2.1</td>
<td>A typical structure of ASL unit. 10</td>
</tr>
<tr>
<td>2.2</td>
<td>Working principle of ASL. 11</td>
</tr>
<tr>
<td>2.3</td>
<td>Structure of a three-input majority gate. 12</td>
</tr>
<tr>
<td>2.4</td>
<td>A full adder using a three-input majority gate. 14</td>
</tr>
<tr>
<td>2.5</td>
<td>ASL full adder constructed from three majority gates. 15</td>
</tr>
<tr>
<td>2.6</td>
<td>The schematic of latch using ASL. 17</td>
</tr>
<tr>
<td>2.7</td>
<td>The schematic of D flip-flop using ASL (D&lt;sub&gt;1&lt;/sub&gt; and D&lt;sub&gt;2&lt;/sub&gt;). 18</td>
</tr>
<tr>
<td>2.8</td>
<td>The schematic of a pipelined full adder using the 2-phase clock. 19</td>
</tr>
<tr>
<td>3.1</td>
<td>An overview of the circuit simulation model between two magnets and a non-ferromagnet channel. 21</td>
</tr>
<tr>
<td>3.2</td>
<td>Spin storage simulation. 22</td>
</tr>
<tr>
<td>3.3</td>
<td>The charge with a spin direction (spin up or down) flows right, generating a charge flow and a spin flow. 24</td>
</tr>
<tr>
<td>3.4</td>
<td>Magnetization direction of $\vec{m}_2$. 25</td>
</tr>
<tr>
<td>3.5</td>
<td>Spin injection model. 27</td>
</tr>
<tr>
<td>3.6</td>
<td>Spin transport model. 30</td>
</tr>
<tr>
<td>3.7</td>
<td>Physical decomposition of the target circuit, including computationally relevant domain and environmental domain. 35</td>
</tr>
</tbody>
</table>
4.1 Circuit simulation results of a two-magnet buffer. ....................... 47
4.2 Circuit simulation results of a two-magnet inverter. ..................... 48
4.3 Circuit simulation results of three-magnet latches. ...................... 50
4.4 Results of simplified buffer and inverter. ............................... 54
4.5 Results of a simplified three-magnet latch, implementing buffer function. ................................................................. 56
4.6 Results of a simplified 3-input 4-magnet majority gate, implementing NAND gate or NOR gate. ................................. 59
4.7 ASL half adder schematic. .................................................. 60
4.8 Simulation results of the half adder. ................................. 61
4.9 Power simulation results of the half adder. ............................ 62
4.10 ASL based ALU schematic. .............................................. 63
4.11 Results of a ASL based ALU. ........................................... 64
4.12 Fundamental and supporting sources. .................................. 68
4.13 Physical decomposition of ASL magnet and its connecting channel, power supply, ground lead, and substrate. .......... 69
4.14 Data zones in different computational step c1~c5. ...................... 71
4.15 Half adder layout. ......................................................... 81
A.1 A small region $dV_r$ in magnetic field $\vec{H}$. Elementary moments $|\mu_j| > s$ point to random directions. ...................... 92
A.2 Procession and damping. ................................................. 92
B.1 Spin current injection. ..................................................... 96
B.2 Self-consistency of LLG nanomagnet dynamics and spin transport. .... 96
B.3 ASL interver with (a) top view and (b) side view. .................... 97
B.4 Circuit simulation model of an ASL inverter. .......................... 97
1.1 From CMOS to All-spin Logic

In the past decades, integrated circuit engineers have focused on device physics, seeking ways to improve performance, speed, the chip area, energy dissipation, and so on. In 1965, Moore [1] pointed out that the number of transistors in integrated circuits has been doubling every two years, leading to a situation where the size of the Complementary Metal-Oxide Semiconductor (CMOS) devices has aggressively scaled down for the past few decades and enabled the industry to build ultra-dense integrated circuits with higher performance [2].

However, when transistors scaled down into the nanoscale regime in recent years, Moore’s Law slowed down. The power dissipation or the heat dissipation became a significant problem preventing the number of the transistors doubles about every two years. Dynamic power dissipation used to dominate the power consumption in CMOS devices, but as the size of the transistor has kept scaling down, the static power (which is from excessive leakage currents) started to play a more critical role in the power or heat dissipation area.

Energy dissipation, or heat dissipation, is always in the list of problems waiting to be solved. It ended the rapid growth of the transistor count and its historical doubling every two years as per Moore’s Law. Heat dissipation causes severe performance problem among very large scale integrated (VLSI) circuits. Considering the scale and number of transistors in a modern VLSI circuit, the heat dissipation generated from all transistors combined is enormous. The small device scale also increases the density
of defective transistors and limits further scaling. Thus, energy/heat dissipation has become one of the most urgent problems in CMOS integrated circuit technology.

The limitations of CMOS device regarding heat dissipation are mainly caused by leakage currents. There are two kinds of leakage current in CMOS. One is the tunneling current from gate to ground due to the thinner dielectric layer. The other leakage is from source to drain due to the unwanted current flow under threshold voltage. These two leakage currents result in significant energy dissipation and off-state power consumption when the size of transistors is smaller, i.e., the density of transistors is higher. One way to reduce the power is to decrease the supply voltage which in turn lowers the tunneling current at the same time. However, we are unable to reduce the threshold voltage suitable for the reduced supply power, which leads to poor performance [3].

To solve the problem, engineers explored many devices beyond CMOS, among which is the all-spin logic device. Behin-Aein [4] showed that one distinguishable feature of this post-CMOS device is that it conducts both information and energy by only spin current instead of charge current. The spin current is a collection of electron spins. The electron spin is a form of angular momentum which has an possible orientation of spin-up or spin-down [5]. Under this consideration, we name it all-spin logic (ASL) device. Instead of charges in capacitors in conventional CMOS, it uses the direction of the magnetic field of spins in ferromagnets to represent two stable states 0 and 1.

Moreover, the supply voltage in the ASL device has nothing to do with the leakage issue we mentioned above. Thus, it has a significant reduction in $V_{DD}$ (0.01V compared to conventional CMOS 0.8V~1V). Low power from low supply voltage seems to be a promising feature for ASL.

Up to now, a few ASL physical circuits have been fabricated in the lab environment. Figure 1.1 shows an ASL unit with two magnets functioning as a buffer/inverter.
Table 1.1: The energy cost and throughput of ASL vs CMOS FFT processor [8].

<table>
<thead>
<tr>
<th></th>
<th>nJ/FFT</th>
<th>VDD</th>
<th>FFT/s</th>
<th>number of devices</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMOS</td>
<td>155</td>
<td>0.35</td>
<td>3.8</td>
<td>627K</td>
</tr>
<tr>
<td>ASL</td>
<td>23.1</td>
<td>0.1</td>
<td>6.5M</td>
<td>1.5M</td>
</tr>
</tbody>
</table>

[6]. The voltage between the magnet and the conducting channel could be viewed as an output voltage. Figure 1.2 shows a 3-input majority gate [7]. At first, all the three input magnets had spin-up spins. Then at time 1.0ns, the left and upper magnet switched to spin-down state, and the third input magnet (lower magnet) remained the same. The majority gate then chose the majority states from the three magnets. In this case, the spin-down state was chosen then transported to the output magnet (right one). At time 3.0ns, the output magnet switched to spin-down state.

As D. Morris mentioned in his paper [8], when compared with a published CMOS fast Fourier transform (FFT) processor [9], their lab built ASL based FFT processor has a 6.7 times lower energy consumption and 1,700,000 times higher throughput (Table 1.1).

1.2 Energy Dissipation in ASL

Since the issue of the energy dissipation plays a vital role in the device performance and the ASL technique advantages over CMOS this area, we will investigate two
Figure 1.2: A 3-input majority gate. (a) A majority gate fabrication layout. (b)∼(d) Magnetization directions of the majority at time t=0.5, 1, 3ns. Two of the input magnets (left and upper one) were switched from spin-up state to spin-down state and one of the input magnets remained unchanged (spin-up state). Then the output magnet (right one) chose the majority state from all the input states. In this case, the output magnet switched to spin-down state at t=3.0ns [7].

energy dissipation calculation methods and apply them in the same set of ASL circuits to see how this novel post-CMOS device performs under different testing methods.

The first one is based on circuit analysis simulation [4][10]. The energy dissipation obtained from this method could be called as projected energy where a numerical estimate is shown as a final result. The projected energy can be viewed as a simulation of a real ASL circuit specified by particular material parameters, structural dimensions, and geometric configurations. The results could be seen as a prediction or projected energy of the real physical ASL circuit simulation results—this is how the name projected energy comes from. The projected energy is determined by many factors. It describes a situation when the device conditions, interconnection models, physical dimensions, material parameters, environmental effects, etc. are given, how much energy is dissipated by a particular circuit realization executing a particular computation. It is similar to the kind of energy dissipation mentioned in a circuit
performance lab test, i.e. the projected energy would ideally be close to the real physical circuit test result conducted in a lab environment.

The second energy calculation method is related to the fundamental energy limits we will discuss in Section 1.2. This method provides lower bounds on the irreversibility induced energy considering that it is from the irreversible information loss during the logical computation [11]. It is an entirely different calculation method from the first one, and the result is of a very different nature than the projected energy. The irreversibility induced energy has nothing to do with the circuit parameters, such as material properties, conductance channel length, environment temperature, etc.

The irreversibility induced energy is derived from the working principle of a circuit, reveals the inner logic behavior of the computation process. Only the logic computation itself and how the investigated circuit realizes it (the function) determine the final numerical result of irreversibility induced energy. It shows the lowest energy bound of a circuit, i.e. how much energy at minimum is necessarily dissipated by a circuit doing a particular computation in a particular way, no matter what the material, structure parameters, connection models, etc. are.

The relation between the projected and irreversibility induced energy is subtle. The projected energy is more familiar to us when compared with the CMOS simulation using software such as Cadence Virtuoso. The irreversibility induced energy is more theoretical and abstract, depending on Landauer’s Principle (will be introduced in Chapter 3) with formula derivation and calculation. The projected energy is related to the static power and the dynamic power. The integration of the static and dynamic power with respect to time is the projected energy, while the irreversibility induced energy is only related to the dynamic power. The detailed definition and comparison will be explained in Chapter 3 and Section 4.4.5.

These two calculation methods will be both applied to the same sets of ASL circuits. With the results obtained, we will get a basic idea of the low energy (the
projected energy) dissipation of ASL device. Secondly, from the comparison of the two methods, we could get a deeper understanding of the projected and irreversibility induced energy—the projected energy of ASL device is low compared to CMOS, but still far above fundamental limits.

1.3 Thesis Objectives

The goal of this thesis is to apply two kinds of energy dissipation calculation methods to ASL circuits. The detailed information of these two calculation methods will be introduced in Chapter 3.

The detailed working principle of ASL will be introduced in this thesis. Its unique working principle of using only spin currents to transfer information gives it a promising feature, which surpasses its competitors of post-CMOS devices. A clear layout of it, including commonly applied material and structure of spin storage component and conducting channel, is illustrated with words and figures. Because of its non-volatile characteristic, the ASL device could be used for logic computation and information storage. Several basic logic circuits based on ASL, such as the buffer, latch, flip-flop, and adder [3] are shown in Chapter 2. The functions (truth tables), transient simulation with provided pulses and the numerical results (i.e., simulation run time, delay time and power consumption) are also described.

The spin-current-only device, which the all-spin logic device gets named from, has a low supply voltage of 10mV, enabling it to work with high circuit density and low heat dissipation rate. Because of this feature, ASL stands out not only among conventional CMOS devices but also among its post-CMOS competitors. Interested in the low power and energy consumption property of ASL, we will apply these two sets of energy calculation methods to the same ASL circuits. With the results obtained, we will get a basic idea of the low energy (the projected energy) dissipation of ASL device. Secondly, from the comparison of the two methods, we could get a deeper
understanding of the projected and irreversibility induced energy and the difference between them.

1.4 Thesis Overview

The second chapter will introduce the working principles of ASL devices and several computing circuits based on ASL units.

In the third chapter, we will introduce the two calculation methods, the projected energy calculation method, and the irreversibility induced energy calculation method. The projected energy calculation method also can be seen as the circuit simulation method, will build a whole set of circuit components especially suited for the ASL devices. The physical structure and corresponding analog model in the stage of spin storage, spin injection and spin transport will be illustrated and then built under the Matlab environment in Section 3.1. These models will be combined in the later chapter and applied for ASL digital circuit simulation to get the projected energy dissipation.

The calculation method of irreversibility induced energy dissipation will also be proposed in this chapter. The irreversibility induced energy is closely related to another theory—the physical information theory. In this section, we will talk about the Landauer’s Principle [12], which points out that irreversible information loss necessarily induces irreversible energy loss, hence the phrase “irreversibility induced energy dissipation”. Followed by this principle, we will introduce terminologies and four steps by which readers can follow to analyze the circuit and get the irreversibility induced energy results [11].

Chapter 4 will show the preliminary results of the two methods applied in several ASL based circuits, including buffer, inverter, latch, and half-adder. A more complicated circuit (an arithmetic logic unit) based on ASL will be introduced. The projected and irreversibility induced energy dissipation will be calculated, and the
corresponding energy efficiencies obtained from the two methods will be compared and discussed.

Chapter 5 is the summary and conclusions of the thesis.
CHAPTER 2
ALL-SPIN LOGIC DEVICE: WORKING PRINCIPLES AND BASIC UNITS

In this chapter, we introduce the basic idea about ASL device, including the structure of basic ASL function unit, ASL circuit elements, and how the spin currents are excited and flow from one magnet to another. A few basic ASL units are also introduced in this chapter, which gives us a better understanding of how the ASL works.

2.1 Working Principles of ASL Devices

Fig. 2.1 is the structure of a typical ASL device unit [3]. Two ferromagnets are labeled in red color. Ferromagnet is a kind of material contains individual domains. Each domain has its magnetic field with random orientation (high probability in the magnetic easy axis). Once an external magnetic field is applied, the orientation of each domain tends to the direction of the external magnetic field. The gray part is the spacer so that no charge can go from one side of the spacer to the other. The green rectangle is the nonmagnetic channel. For better performance, copper is a better choice for short distance communication channel while graphene is a better choice in the long-distance communication channel [13][14][15]. The ground lead which is shown in blue color is also made of nonmagnetic material. The ground lead is close to one of the ferromagnets, next to the spacer.

In Fig. 2.2, the supply voltage $V_{DD}$ of the left ferromagnet $m_1$ injects current to $m_1$ [16]. The current is then injected to the channel and goes all the way down to
the ground via the ground lead (red arrow in Fig. 2.2). This “red dots” current has the same polarization direction with $m_1$ (detailed discussion is given in Appendix B). Note that positive supply voltage extracts electrons from ground to magnet, i.e. the charge current goes from the magnet to ground, while negative supply voltage injects electrons, i.e. the charge current goes from ground to the magnet. The injection of the electrons accumulates electrons under $m_1$. The accumulated electrons have the same polarization with $m_1$.

On the other hand, the extraction of the electrons accumulates electrons with the opposite polarization direction with $m_1$. These accumulated electrons with either the same or the opposite polarization with $m_1$ diffuse from one side of the channel to the other side, i.e. from beneath $m_1$ to beneath $m_2$. This diffusion current is called the spin current (green dots in Fig. 2.2). When the electrons reach beneath $m_2$, they change the direction of magnetization of $m_2$ to whatever the polarization direction of the electrons is, by torque. This changing process is discussed in detail in Appendix B.

All in all, the negative supply voltage injects majority spins (the ones that have the same magnetization direction with input spins) into the channel. Part of the spin
Figure 2.2: Working principle of ASL [13]. This is an inverter (with the positive supply voltage). The red current generated by supply voltage goes from supply to ground. It is polarized by magnet 1 and accumulates antiparallel polarized spin current near the magnet 1 in the channel. The antiparallel polarized spin current (green current) acts as information current. It torques the magnetization of magnet 2 to the opposite direction of magnet 1, thus realizing a NOT function.

goes all the way down to the ground (the red dots charge current in Fig. 2.2). The rest is accumulated at the left side of the channel to create a diffusion spin current (the green arrow spin current in Fig. 2.2) to the right side (ferromagnet m2 side), thus carrying the input information from input to output. Because the spin current has the same spin direction with that of input, output realizes a COPY from the input.

Vice versa, when the supply voltage is positive, majority spins are extracted from the channel, leading to the accumulation of opposite (minority) spins beneath m2. Thus, the diffusion current (spin current) has the opposite spin direction from that of input. In this case, the charge current and the spin current have different polarization directions. The output operates logic function NOT from the input, i.e. the receiving magnet has the opposite magnetization direction to the transmitting magnet.

Because of the structure and working principle of ASL, we can easily tell that the magnet closer to the ground lead is the sender, and the remote one is the receiver [3].
Figure 2.3: Structure of a three-input majority gate [14]. Magnet A and C have the right pointed magnetization and magnet B has the left pointed magnetization. Output follows the majority direction of inputs, i.e. the right pointed magnetization.

2.2 Digital Computing with ASL

Almost all the logic circuits are built with majority gates. Fig. 2.3 is the basic structure of a three-input majority gate based on ASL [14]. Inputs A, B, and C transmit their spin currents from input to output via copper channels. Two inputs could have either parallel spins or anti-parallel spins. However, for all three inputs, there must be a majority spin that determines the magnetization direction of output. For example, as shown in Fig. 2.3, input A and C have the magnetization direction to the right and B to the left. In this case, the output magnetization direction will appear as the same as the majority magnetization direction, i.e. to the left (the same with input A and C).

A three-input majority gate itself can function as NAND, NOR, AND, and OR gate if the supply voltage is appropriately chosen, as shown in Table 2.1 [3].

Input B and C act as valid inputs which could be customized by the user. Input A and supply voltage $V_{DD}$ act as controllers. When A is 0 and $V_{DD}$ is positive, function B NAND C is realized. Similarly, when A is 0 and $V_{DD}$ is negative, function B AND C is realized; when A is 1 and $V_{DD}$ is positive, function B NOR C is realized; when
Table 2.1: The truth table of three-input majority gate [3].

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Positive $V_{DD}$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2.2: The truth table of full adder with the illustration of xor/xnor gate design [3].

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C$_{in}$</th>
<th>C$_{out}$</th>
<th>Comp.C$_{out}$</th>
<th>Comp.S</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

A is 1 and $V_{DD}$ is negative, function B OR C is realized. In this way, computing elements could be built with three-input majority gates.

The typical building block mentioned in this section is the full adder. Three inputs are needed to build a full adder: two addends A and B, and a carry C$_{in}$. The truth table of a full adder is shown in Table 2.2 [3]. Notice that in a full adder, XOR and XNOR gates can also be realized.

The layout of a full adder is shown in Fig. 2.4 [3]. The supply voltage is positive to create a complementary output C$_{out}$ and Sum S.

A trick is applied to reduce the potential number of majority gates. We can see from the truth table (Table 2.2) that in order to get the complementary S, three inputs
and two complementary $C_{out}$s are needed. The trick is: the channel that transmits complementary $C_{out}$ is made shorter than the channel transmitting inputs $A$, $B$ and $C_{in}$, so that the spin current of the complementary $C_{out}$ is strong enough to act as two complementary $C_{out}$s (the relative strength compared to $A$, $B$ and $C_{in}$).

As shown in Fig. 2.5 (a), the channels connected to the five-input majority gate ($M_3$) is made to have the same length (the wire length in Fig. 2.5 does not represent the real nor relative length); thus the five inputs of the five-input majority gate have the same strength. In this case, two three-input majority gates ($M_3$) are built to generate two $C_{out}$s and one five-input majority gate is built to generate one $S$. Fig. 2.5 (b) is the modified schematic of a full adder, where the trick is used. Only one three-input majority gate is used to generate two $C_{out}$s. Only one $C_{out}$ is generated.
Figure 2.5: (a) ASL full adder constructed from three majority gates. $M_x$ means a majority gate with $x$ inputs, i.e. $M_3$ means a majority gate with three inputs; (b) modified ASL full adder. The five-input majority gate has five inputs: $C_{out}, C_{out}, A, B$ and $C_{in}$. Because we have two $C_{out}$ goes into $M_5$ at the same time, the strength of logical $C_{out}$ is twice as strong as other inputs.)
by one three-input majority gate, but the strength of this $C_{out}$ is as strong as two $C_{out}$s compared to the $C_{out}$ generated in Fig. 2.5 (a). Because the connection channel between this $C_{out}$ and the input side of $M_5$ is made shorter than other connection channels, $C_{out}$ is twice as strong as each input $A$, $B$ and $C_{in}$.

In actual design, instead of going into $M_5$, three inputs $A$, $B$ and $C_{in}$ are combined together and then joined with $C_{out}$, i.e. $M_5$ is not used in this modified design, as shown in Fig. 2.4. The combined input, containing the spins of two $C_{out}$s and one $A$, one $B$, and one $C_{in}$, goes into a single ASL basic unit (i.e. an inverter). Thus, by using the trick, the full adder is made of a three-input majority gate and an inverter, as shown in Fig. 2.4. Because of the routing issue of inputs $A$, $B$ and $C_{in}$, the actual size of this type of full adder is equal to a three three-input majority gates circuit.

A basic ASL unit can act as a latch because when no voltage is supplied, the state of each ferromagnet is stable. When the supply voltage $V_{DD}$ of a basic ASL unit changes to a clock voltage $V_{CLK}$, the basic ASL unit changes to an ASL latch. In Fig. 2.6 [3], the middle ASL unit acts as a latch. We can see in Fig. 2.6 (b) and (c) that the clock voltage $V_{CLK}$ has two phases: $V_{SS}$ and $V_{DD}$. In (b), unit $A$ sends current to $L$ and changes the magnetization of $L$, i.e. $L$ has either the same or opposite magnetization of $A$. However, $L$ will not send current to $B$, because no voltage ($V_{SS}$) is supplied. The latch is “OFF”. Thus, $L$ separates $A$ from $B$. In (c), $V_{DD}$ is supplied to $L$. $L$ is “ON”, and sends any information it received from $A$ to $B$. $L$ is transparent now.

The schematic of D flip-flop (DFF) is shown in Fig. 2.7 [3]. These components have been made physically and tested by M.Alawein’s group. Two ferromagnets $D_1$ and $D_2$ are supplied with $V_{CLK}$ and $V_{CLK}-b$ respectively. $V_{CLK}-b$ is the complement of $V_{CLK}$. $V_{CLK}$ has two phases: $V_{SS}$ and $V_{DD}$. We can see from (b) that when $D_1$ is supplied with $V_{DD}$, the information is transmitted from $A$ to $D_1$, and held there until the supply voltage of $D_2$ changes from $V_{SS}$ to $V_{DD}$, as shown in (c). The information
Figure 2.6: (a) The schematic of latch using ASL; (b) Current cannot flow from A to B via L when the supply voltage of L is low (opaque); (c) Current flows from A to B via L when the supply voltage of L is high (transparent) [3].
then transmits from \(D_2\) to \(B\) in the same clock cycle. It is a positive edge-triggered DFF. A negative-edge-triggered DFF is achieved by exchanging \(V_{CLK}\) and \(V_{CLK}\) of \(D_1\) and \(D_2\).

Fig. 2.8 shows the schematic of pipelined full adder with a 2-phase clock [13]. Three full adders are connected and supplied with different clock voltage. The first and third adder (FA1 and FA3) use \(V_{CLK}\) and the second adder (FA2) uses \(V_{CLK-b}\). \(V_{CLK-b}\) is low when \(V_{CLK}\) is high and vice versa. The width of the clock pulse depends on the critical path of each adder.

When \(V_{CLK}\) is high, the first and third adder start to compute and send information to the next stage, i.e. the first adder sends the computation result to the second adder, and the third adder sends the result to its next stage, which is not shown in
Figure 2.8: The schematic of a pipelined full adder using the 2-phase clock [13]. CLK and CLK-b have opposite clock phases. When CLK is high, and CLK-b is low, FA1 and FA3 work and FA2 is held. When CLK is low, and CLK-b is high, FA1 and FA3 are at hold state and FA2 works.

This figure. FA2 absorbs the output from FA1 (i.e. $C_{out}$ from FA1) but doesn’t do any further computation. Thus, except one magnet keeps the result ($C_{out}$ from FA1) from the current cycle, the other magnets of FA2 keep the results from the previous clock cycle.

Then $V_{CLK}$ changes to low and $V_{CLK-b}$ changes to high. The first adder and the third adder turn to the steady state, keeping the result they got in the last clock cycle (except the receiving magnet, i.e. the magnet receives the previous adder’s $C_{out}$), while the second adder starts to its computation and sends the result to the third adder. The third adder receives the computation result ($C_{out}$) from the second adder but doesn’t do any computation until $V_{CLK}$ changes to high.
This chapter introduces two different theoretical approaches for studying energy dissipation in ASL: circuit simulation and the physical-information-theoretic analysis. Circuit simulation provides energy dissipation estimates for specified circuit realizations of ASL circuits, considering the material parameters, connection models, device dimensions, and supply voltage conditions, etc. The results are numerical projections of energy dissipation [17][10]. The physical information theoretic analyses, on the other hand, provides fundamental lower bounds on energy dissipation that depend on the circuit structure and control but are independent of the particulars of the circuit and device realizations. These analyses quantify the amount of information irreversibly lost during computation from the underlying data processing method and then obtains the energy dissipation of this amount of information loss [11]. This chapter introduces this two methods in detail, letting readers have a better understanding of them and knowing how to apply them in the basic ASL circuits.

3.1 Energy Efficiency From Circuit Simulation

3.1.1 Landau-Lifshitz-Gilbert Equation and spin storage

Landau-Lifshitz-Gilbert Equation (LLG equation) describes the behavior of magnets regarding the changes of the magnetic field and spin current polarization, i.e. the
Figure 3.1: An overview of the circuit simulation model between two magnets and a non-ferromagnet channel. Input voltage applies to magnet 1, affects spins and generates spin currents. Magnet 1 consists of two steps of the simulation, spin injection and spin storage. Then, the spin currents go through a non-ferromagnetic channel, which is another step of simulation called spin transport. When the spin current arrives at magnet 2, it affects the spin direction of Magnet 2, and an output voltage will be detected. The feedback spin current carries feedback information flows through non-ferromagnet channel and reaches magnet 1. The magnetization direction of magnet 1 will be influenced by the feedback current slightly and quickly reach the equilibrium state.

LLG equation illustrates how the information in the magnetization of the ferromagnet and the current influences each other. The LLG equation is written as follow:

$$\frac{\partial \vec{M}}{\partial t} = -\gamma \vec{M} \times \vec{H}_{\text{eff}} + \frac{\alpha}{M_s} \vec{M} \times \frac{\partial \vec{M}}{\partial t}$$  \hspace{1cm} (3.1)$$

where $M_s = |\vec{M}|$ is the magnitude of magnetization vector in a small region $dV$, $t$ is time, and $\vec{H}_{\text{eff}}$ is the effective magnetic field applied to $dV$ and $\alpha > 0$ is the Gilbert constant, depending on the material.

The detailed Landau-Lifshitz-Gilbert equation derivation is in Appendix A. The general LLG equation shown at the end of Appendix A is:

$$\frac{\partial \vec{M}}{\partial t} = -\gamma \vec{M} \times \vec{H}_{\text{eff}} + \frac{\alpha}{M_s} \vec{M} \times \frac{\partial \vec{M}}{\partial t}$$  \hspace{1cm} (3.2)$$

When applying the LLG equation in a spintronic device, we need to take the influence of spin current polarization into consideration, as shown in Appendix B.
We make a step further to break down the modified LLG equation into three equations, each of which stands for the relationship between current and magnetic field or voltage in one direction (x, y, or z-direction) of the magnets [17]:

\[ N_s q (1 + a^2) \frac{dm_x}{dt} = f(\vec{m}, \vec{I}_s, \vec{H}_{eff}) \] (3.3)

\[ N_s q (1 + a^2) \frac{dm_y}{dt} = g(\vec{m}, \vec{I}_s, \vec{H}_{eff}) \] (3.4)

\[ N_s q (1 + a^2) \frac{dm_z}{dt} = h(\vec{m}, \vec{I}_s, \vec{H}_{eff}) \] (3.5)

The formula of these three equations is similar to \( C \frac{dv}{dt} = i \), i.e., the I-V characteristics of charging and discharging a capacitor. Here capacitance \( C = N_s q (1 + a^2) \) and \( i = f(\vec{m}, \vec{I}_s, \vec{H}_{eff}) \) or \( i = g(\vec{m}, \vec{I}_s, \vec{H}_{eff}) \) or \( i = h(\vec{m}, \vec{I}_s, \vec{H}_{eff}) \). Thus we have a spin storage simulation circuit as Fig. 3.2 [17]. Current i is a voltage dependent current source. \( H_{eff} \) is a voltage dependent voltage source.
Function $f(\vec{m}, \vec{I}_s, \vec{H}_{\text{eff}})$, function $g(\vec{m}, \vec{I}_s, \vec{H}_{\text{eff}})$, and function $h(\vec{m}, \vec{I}_s, \vec{H}_{\text{eff}})$ are

$$f(\vec{m}, \vec{I}_s, \vec{H}) = I_{s,y} m_y m_y - I_{s,x} m^2_y + I_{s,z} m_z m_z - I_{s,z} m^2_z - I_s, z m^2_x m_y a$$

$$- I_{s,z} m^3_y a + I_{s,y} m^2_x m_z a + I_{s,y} m^2_y m_z a - I_{s,z} m^2_y m_z a - I_{s,y} m^3_z a + H_{\text{eff}, z} m_y N_s q \gamma \mu_0$$

$$- H_{\text{eff}, y} m_z N_s q \gamma \mu_0 + H_{\text{eff}, y} m_z N_s a q \gamma \mu_0 - H_{\text{eff}, x} m_y N_s q a \gamma \mu_0$$

$$+ H_{\text{eff}, z} m_x m_z N_s q a \gamma \mu_0 - H_{\text{eff}, z} m^2_x N_s q a \gamma \mu_0$$

(3.6)

$$g(\vec{m}, \vec{I}_s, \vec{H}) = - I_{s,y} m^2_x + I_{s,x} m_x m_y + I_{s,z} m_y m_z - I_{s,y} m^2_x + I_{s,z} m^3_x a$$

$$+ I_{s,z} m_x m^2_y a - I_{s,z} m_x m^2_y a - I_{s,z} m^2_y m_a + I_{s,z} m^2_x m_a - I_{s,z} m^3_a$$

$$- H_{\text{eff}, x} m_x N_s q \gamma \mu_0 + H_{\text{eff}, x} m_z N_s q \gamma \mu_0 - H_{\text{eff}, y} m^2_x N_s q a \gamma \mu_0 + H_{\text{eff}, x} m_y N_s q a \gamma \mu_0$$

$$+ H_{\text{eff}, x} m_y N_s q a \gamma \mu_0 - H_{\text{eff}, y} m^2_x N_s q a \gamma \mu_0$$

(3.7)

$$h(\vec{m}, \vec{I}_s, \vec{H}) = - I_{s,z} m^2_x - I_{s,z} m^2_y + I_{s,x} m_x m_z + I_{s,y} m_y m_z - I_{s,y} m^3_x a$$

$$+ I_{s,z} m^2_x m_y a - I_{s,z} m^2_x m_y a - I_{s,z} m^2_x m_y a + I_{s,z} m^2_x m_y a + I_{s,z} m^2_x m_y a$$

$$H_{\text{eff}, y} m_x N_s q \gamma \mu_0 - H_{\text{eff}, x} m_y N_s q \gamma \mu_0 - H_{\text{eff}, x} m^2_x N_s q a \gamma \mu_0 - H_{\text{eff}, y} m^2_x N_s q a \gamma \mu_0$$

$$+ H_{\text{eff}, x} m_x m_z N_s q a \gamma \mu_0 + H_{\text{eff}, y} m_y m_z N_s q a \gamma \mu_0$$

(3.8)

### 3.1.2 Spin Injection

Two-channel resistor model [18][19] is used to build up the spin injection model. In two-channel resistor model, the spin current is sorted into majority spin current $I_\uparrow$ and minority spin current $I_\downarrow$, determined by the majority and minority spin conductance of material $G_\uparrow$ and $G_\downarrow$, respectively [18].
Figure 3.3: The charge with a spin direction (spin up or down) flows right, generating a charge flow and a spin flow. For charge flow, the charge current $\vec{I}_C = \vec{I}_\uparrow + \vec{I}_\downarrow$. For spin flow, the spin current $\vec{I}_s = \vec{I}_\uparrow - \vec{I}_\downarrow$.

Usually, a flow of charges (electrons) only has charge current, because the numbers of spin-up electrons and spin-down electrons are the same, and the spin current $\vec{I}_s = \vec{I}_\uparrow - \vec{I}_\downarrow = 0$. However, if we use a special method to make the number of spin-up and spin-down electrons unequal, i.e. a majority spin current and a minority spin current are created, we get both the charge current and the spin current, which is the case for all-spin logic. Ideally, the charge current only goes from the voltage supply to the ground lead and the spin current to transport from one magnet to the other via channel.

For the charge current $\vec{I}_C$ in all-spin logic, we can use the conventional model to calculate the value:

$$I_C = G_\uparrow (V_{C,N} + \vec{V}_s \cdot \vec{m} - V_{C,F}) + G_\downarrow (V_{C,N} - \vec{V}_s \cdot \vec{m} - V_{C,F})$$  \hspace{1cm} (3.9)

where $G_\uparrow$ and $G_\downarrow$ are the majority and minority spin conductance, respectively. $V_{C,N}$ and $V_{C,F}$ are the charge voltage for the nonmagnetic material and ferromagnetic material, respectively. $\vec{V}_s$ is the spin voltage. By calculating the dot product in the
The equation above, we get

\[ I_C = (G_\uparrow + G_\downarrow)(V_{C,N} - V_{C,F}) + (G_\uparrow - G_\downarrow)V_{s,x} + (G_\uparrow - G_\downarrow)V_{s,y} + (G_\uparrow - G_\downarrow)V_{s,z} \]  
(3.10)

where \( V_{s,x}, V_{s,y} \) and \( V_{s,z} \) are the spin voltage in x, y and z-direction.

Because the magnetization directions on the two sides of the interface are usually non-linear, i.e. \( m_1 \neq \pm m_2 \), the spin current \( I_s \) is then divided into parallel and perpendicular spin current. As shown in Figure 3.4, the parallel current \( I_{s,\parallel} \) is in-plane perpendicular to the magnetization direction of the injected material \( \vec{m}_2 \), thus its magnetization direction is parallel to \( \vec{m}_2 \). The perpendicular current \( I_{s,\perp} \) has two part. One part is parallel to the magnetization direction of the injected material \( \vec{m}_2 \). The other part is out-of-plane perpendicular to the magnetization direction. Both parts have a magnetization direction that is perpendicular to \( \vec{m}_2 \).

Thus, the parallel spin current can be written as

\[ \vec{I}_{s,\parallel} = \vec{m} \cdot [G_\uparrow(V_{C,N} + \vec{V}_s \cdot \vec{m} - V_{C,F}) - G_\downarrow(V_{C,N} - \vec{V}_s \cdot \vec{m} - V_{C,F})] \]  
(3.11)
\[ \vec{I}_s = (G_\uparrow - G_\downarrow) m_x (V_{C,N} - V_{C,F}) \vec{x} + (G_\uparrow - G_\downarrow) m_y (V_{C,N} - V_{C,F}) \vec{y} + (G_\uparrow - G_\downarrow) m_z (V_{C,N} - V_{C,F}) \vec{z} + (G_\uparrow + G_\downarrow) m_x V_{s,x} \vec{x} + (G_\uparrow - G_\downarrow) m_y V_{s,y} \vec{y} + (G_\uparrow - G_\downarrow) m_z V_{s,z} \vec{z} \] (3.12)

The perpendicular spin current can be written as

\[ \vec{I}_{s,\perp} = 2 \text{Re} G_{\uparrow \downarrow} \vec{m} \times (\vec{V}_s \times \vec{m}) + 2 \text{Im} G_{\uparrow \downarrow} \vec{V}_s \times \vec{m} \] (3.13)

3.1.3 Spin Transport

First, it can be inferred from the name of the all-spin logic device that, in the transport channel, all the information is transmitted only by the spin current. Special methodologies are used to generate only spin current but no charge current. However, the charge current does exist in the channel. It is generated only by the charge diffusion, while both spin drift and diffusion generate the spin current.

To present the charge diffusion equation, we have the following expansion. The charge diffusion equation is a continuity equation based on a basic equation \( \frac{Q}{\partial t} = I \)
where $Q = CV$. Substitute $Q$ into the first equation, we get $\frac{CV}{t} = I$. By making the partial derivative of time on left part and the partial derivative of length on the right side (to simulate the charge and discharge process in a limited transmission line $\partial x$), we get the continuity equation of charge diffusion current as follows [20][21]

$$
C_e \frac{\partial V_c}{\partial t} = \frac{\partial I_c}{\partial x} = \sigma A \frac{\partial V_c}{\partial x}
$$

(3.15)

where $C_e$ is the electrostatic capacitance per unit length, $V_c$ is the charge voltage and $I_c$ is the charge current, $\sigma$ is the conductivity and $A$ is the cross-sectional area.
For spin current, as the form of Equation 3.15, we can represent it in both the continuity equation and current density.

For the spin current continuity equation, it is similar to that of the charge continuity equation (Equation 3.15), but with an additional spin relaxation part.

\[
\frac{\partial \vec{s}}{\partial t} = \frac{\partial \vec{J}_s}{\partial x} - \frac{s}{\tau_s}
\]  

(3.16)

where \( \vec{s} \) is the spin accumulation, \( \tau_s \) is the spin relaxation time. \( \frac{\partial \vec{s}}{\partial t} = \frac{\partial \vec{J}_s}{\partial x} \) is the continuity version of \( \frac{Q}{t} = I \).

Driven by both drift and diffusion, the current density can be written as \( \vec{J}_s = \vec{J}_{diff} + \vec{J}_{drift} = D\frac{\partial \vec{s}}{\partial x} + \mu E \vec{s} \), or

\[
\vec{J}_s = D\frac{\partial \vec{s}}{\partial x} + \mu E \vec{s}
\]

(3.17)

where \( \mu \) is the mobility of electrons or holes, \( E \) is the electric field applied to the channel, \( D \) is the diffusion coefficient for the electron, and \( \vec{s} \) is the spin accumulation.

It can also be written as \( s = \mu_s \frac{\partial n_0}{\partial \eta} \) where \( \mu_s \) is the spin quasi-chemical potential, which can also be written as \( \mu_s = V_s q^2 \), \( \eta \) is the chemical potential, and \( n_0 \) is the carrier concentration. Substitute \( \mu_s \) in, we get \( s = V_s q^2 \frac{\partial n_0}{\partial \eta} \). Divide both sides by \( V_s \), we get \( \frac{s}{V_s} = q^2 \frac{\partial n_0}{\partial \eta} \). This is the form of quantum capacitance \( C_q \), i.e. \( \frac{s}{V_s} = q^2 \frac{\partial n_0}{\partial \eta} = C_q \).

And according to the general form of Einsteins relation, \( \sigma = q^2 \frac{\partial n_0}{\partial \eta} \), we get \( C_q = \frac{\sigma}{D} \).

Thus, \( s = \frac{\sigma}{D} V_s \). Putting \( s \) in vector, we get

\[
\vec{s} = \frac{\sigma}{D} \vec{V}_s
\]

(3.18)

Substitute Equation 3.18 into Equation 3.16 and 3.17, we get
\( C_q \frac{\partial \vec{V}_s}{\partial t} = \frac{\partial \vec{J}_s}{\partial x} - \frac{C_q \vec{V}_s}{\tau_s} \)

\( \vec{J}_s = \sigma \frac{\partial \vec{V}_s}{\partial x} + \frac{\mu E \sigma}{D} \vec{V}_s \)  \hspace{1cm} (3.19)

To turn the current density into current, we times the cross-sectional area \( A \) on the both side of Equation 3.19

\( C_q A \frac{\partial \vec{V}_s}{\partial t} = \frac{\partial \vec{I}_s}{\partial x} - \frac{C_q A \vec{V}_s}{\tau_s} \)

\( \vec{I}_s = \sigma A \frac{\partial \vec{V}_s}{\partial x} + \frac{\mu E A}{D} \vec{V}_s \)  \hspace{1cm} (3.20)

where \( E A \sigma = \frac{\partial V_c}{\partial x} A \sigma = I_c \). Thus, Equation 3.20 can be written as

\( C_q A \frac{\partial \vec{V}_s}{\partial t} = \frac{\partial \vec{I}_s}{\partial x} - \frac{C_q A \vec{V}_s}{\tau_s} \)

\( \vec{I}_s = \sigma A \frac{\partial \vec{V}_s}{\partial x} + \frac{\mu I_c}{D} \vec{V}_s \)  \hspace{1cm} (3.21)

Combining Equation 3.15 and Equation 3.21, we get a general equation to describe spin transport.

\( C_s \frac{\partial \vec{V}_s}{\partial t} = \frac{\partial \vec{T}_s}{\partial x} - G_s \vec{V}_s \)

\( \vec{T}_s = \sigma A \frac{\partial \vec{V}_s}{\partial x} + \frac{\mu I_c}{D} \vec{V}_s \)  \hspace{1cm} (3.22)

where \( C_s \) is the unit tensor capacitance of the material

\[
C_s = \begin{bmatrix}
C_e & 0 & 0 & 0 \\
0 & C_q A & 0 & 0 \\
0 & 0 & C_q A & 0 \\
0 & 0 & 0 & C_q A \\
\end{bmatrix}
\]  \hspace{1cm} (3.23)
, and $G_s$ is the spatial tensor conductance of the non-ferromagnetic channel. It is defined as

$$G_s = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & \frac{C_q A}{\tau_s} & 0 & 0 \\
0 & 0 & \frac{C_q A}{\tau_s} & 0 \\
0 & 0 & 0 & \frac{C_q A}{\tau_s}
\end{bmatrix} \quad (3.24)$$

The spatial derivative $\partial_x$ can be simplified as a finite numbers $N$ of $\Delta x$ where $\Delta x$ is a tiny amount of length of the whole channel $L$. Thus between the two ends of $\Delta x$, the voltages are $\Delta V_s(x)$ and $\Delta V_s(x + \delta x)$, respectively. The corresponding currents are $\Delta I_s(x)$ and $\Delta I_s(x + \delta x)$, respectively.
3.1.4 Energy Efficiency

The energy efficiency, derived from input-averaged energy dissipation, is the number of computational operations performed per dissipated Joule of energy. It is a post-processing step for the total energy dissipation obtained either from circuit simulation of physical-information-theoretic analysis.

The energy dissipation is “input averaged” to meaningfully account for the dissipation incurred in processing all computational inputs, which may differ from input to input. Thus the energy dissipation may vary from different input patterns and lengths. For example, we have a circuit with three inputs (eight possible patterns from 000 to 111). By applying the two calculation methods, we could get the projected energy $E_A$ and irreversibility induced energy $E_B$. By the working principle, the projected energy $E_A$ is the energy dissipation from the circuit simulation with an input pattern from 000 to 111, and the results of projected energy may vary from different chosen options. The irreversibility induced energy $E_B$, however, depends only on the statistics of the input, i.e. $E_B$ need not be calculated individually for the inputs 000, 001, 010, 011... The final result is already input averaged.

For this reason, the definition of input averaged energy is used in this thesis, allowing a way for different types of energy dissipation to compare with each other in the same standard.

3.2 Efficiency Limits from Physical Information Theory

The projected energy is a numerical estimate of the energy dissipation per computation obtained for a specific circuit realization from circuit simulations. The fundamental lower bound of the projected energy, which cannot be obtained from circuit simulations, is the dissipation resulting from irreversible information loss associated with the logic function performed by the circuit and the manner in which this function is performed. The irreversibility induced energy is only related to the irreversible in-
formation loss, regardless of the circuit material, connections or working environment. It yields the lower bound of energy dissipation.

In this thesis, we obtain the fundamental lower bound on the energy dissipation using the methodology of Ercan and Anderson [11]. In this methodology, there are four steps to determine the lower boundary of a given circuit: physical decomposition, process abstraction, operational decomposition, and cost analysis. We will first introduce physical information theory, and then explain in detail how to calculate irreversible information loss using the four steps mentioned above and get the corresponding energy dissipation.

### 3.2.1 Introduction of Physical Information Theory

Devices give out heat during the computation steps. Some heat is reversible, and some heat is irreversible. If we define the device and its relative thermodynamic surroundings as a universe, the reversible computation process is, thermodynamically, in this defined universe, the final equilibrium state can be restored to the initial equilibrium state without any energetic cost after a circuit operation [22].

For example, we have input $x$ going through device $D$, realizing function $F(x)$ and getting the output $y$. The initial equilibrium state is when input $x$ is in device $D$ but hasn’t started the computation steps yet. The final equilibrium state is when input $x$ goes through the machine, gets the output $y$, and all the heat is dissipated in the surrounding environment, i.e. the heat in the universe comes to a steady state, not flowing from one end to another. If we could reverse this finally equilibrium state back to the initial equilibrium state without any energy cost outside the universe, i.e. all the heat flow is within the universe; we say this computation operation is reversible [22].

Logically, the reversible computation operation means the inputs and outputs have a one-to-one relation, i.e. for a given output $y$, the input $x$ is determined. For
example, the device $D$ realizes function $F(x)$. Now we have input series $x_1, x_2, ..., x_n$ goes through the device $D$, and gets the output series $y_1, y_2, ..., y_n$. If the number of outputs is the same with inputs and has a one-to-one relation, i.e. one output only has one related input, the computation process is reversible.

On the other hand, the irreversible computation operation is, regarding thermodynamics, when the final equilibrium state of the universe could not be restored to the initial equilibrium state without the additional energy cost from outside the universe. Regarding logic circuit computation, irreversible computation operation means that in the device $D$, there exists at least one output $y$ related to at least two inputs $x_1$ and $x_2$, i.e. for this output $y$, we cannot figure out which input $x$ generates it. In this case, the computation operation is irreversible.

Irreversible computation process is closely related to our final goal—the irreversibility induced energy dissipation. Landauer’s Principle points out that if the computation process is irreversible, the information in the device is erased and can’t be restored, while the reversible computation process means the information can be restored and no information is lost.

If the information is lost, energy will be consumed, and heat will be dissipated into the environment. The relation between information loss and energy/heat dissipation is also shown in the Landauer’s Principle [12]

$$\triangle\langle E \rangle \geq -k_B T \ln(2) \times \triangle I$$

(3.25)

where $k_B$ is the Boltzmann constant and $-\triangle I$ is the information loss from the computation process.

From equation 3.25, we can get a general idea of how physical information theory works: the circuit gives out heat if the computation requires irreversible information loss, i.e. if there exist one or more outputs that can not be referred to a unique input, we say the information is lost from the computing process. In this way, we can get the
irreversibility induced energy dissipation from calculating the irreversible information loss.

The detailed analysis steps [11] are shown in the flowing subsections.

### 3.2.2 Physical Decomposition

The first step in the physical-information-theoretic analysis is the physical decomposition [11], which divides the entire system into two separate parts — computationally relevant domain and environmental domain. These two domains do not have overlap, but they can interact with each other during the computing process. Also, the entire system is carefully defined so that we could treat it as a closed system, i.e. all the heat exchanges is done within the system.

As shown in Fig. 3.7 (a) and (b), the computationally relevant domain has three parts: information processing artifact A (which could be further divided into representational elements C and nonrepresentational elements $\overline{C}$), input referent R, and supporting computational subsystems $\overline{A}$. The detailed definition will be illustrated below.

- **Computationally Relevant Domain** - The universe directly related to the computing process that consists of a physical artifact (device or circuit) and surrounding subsystem.
  
  - **Information Processing Artifact A** - Information processing artifact A is the circuit of interest that realizes expected function $L(x_i)$ where $x_i$ could be any possible argument $x_i \in \{x_i\}$. The artifact A consist of the following elements[11]:
    
    * **Representational Elements C** - representational elements C are the components designed for storing the information during the computation.
Figure 3.7: Physical decomposition of the target circuit, including computationally relevant domain and environmental domain [11].
* Non-representational Element $\overline{C}$ - nonrepresentational elements $\overline{C}$ are the remaining parts of the information processing artifact $A$. Sometimes they store the information during computation as well, but they are not designed for information storage. For example, the connection wires between representational elements. They are designed for transport usage, but sometimes they bear information as well.

– Input Referent $R$ - Input referent its the original inputs without being processed by the information processing artifact $A$, i.e. the “source file” copied from I/O port.

– Supporting Computational Subsystem $\overline{A}$ - Subsystem external to $A$ supports $A$ to complete its computing program. Basically it provides all the information $A$ needs for its function $L(x_\eta^i)$ where $\eta$ is the $\eta$th input of input $x_i$. Thus $\overline{A}$ can be a simple register or buffer that stores $L(x_\eta^i)$. Or it can be the previous or next stage “$A$” that produce $L(x_{\eta-1}^i) = x_i$ or receive $L(x^\eta_i) = x_{\eta+1}^i$.

• Environmental Domain - Environment domain is everything in the closed system except the computationally relevant domain. It consists of two parts, heat bath $B$ and remote environment $\overline{B}$ [11]:

  – Heat Bath $B$ - Heat bath $B$ is the environment that contacts directly to both artifact $A$ and remote environment $\overline{B}$. It is a heat exchanging bridge over the information processing artifact $A$ and environment $\overline{B}$. It usually first exchanges heat with artifact $A$ due to the information loss caused by circuit computation. Second, it exchanges heat with environment $\overline{B}$ to bring its temperature back to the balance temperature $T$.

  – Environment $\overline{B}$ - Remote environment $\overline{B}$ is the greater environment that restores the heat bath $B$ to equilibrium state (temperature $T$).
3.2.3 Process Abstraction

Process abstraction describes how the entire closed system changes and “rethermalized” (i.e. the temperature goes back to the balance state temperature $T$) in the computing process [11]. It contains control operations and a restoration process.

- Control operations — The information in representational elements changes, thus causes the information loss and heat dissipation. The heat dissipation usually involves the information processing artifact $A$, supporting computational subsystem $\overline{A}$ and the heat bath $B$. The process is usually done by a set of control operations $\phi_t \in \{\phi_t\}$ where $\{\phi_t\}$ is determined by the implemented functions of information processing artifact $A$ [11].

- Restoration processes — Remote environment $\overline{B}$ restores heat bath $B$ and computational relevant domain $\overline{A}$. Heat bath $B$ drives artifact $A$ toward the thermal equilibrium state, but the results of the analysis do not presume that the state of $A$ at the conclusion of a computational step is restored to equilibrium.

3.2.4 Operational Decomposition

Operational decomposition divides the entire closed system regarding time and space so that we could analyze the information cost step by step in the next section. Operational decomposition consists of clocking part and computation part. The detailed explanation is as follows [11]:

- clocking

  - Clock zones and subzones - A clock zone is a series of representational elements $C$ that responds to the same control operation $\phi_t \in \{\phi_t\}$ at the same time $t$. Time $t$ could be any desired time. Each of the representational elements $C$ may have different separate physical locations that have no direct interaction with each other, i.e. they may compute independently.
A collection of these physically separated representational elements within the same clock zone is called a subzone. A clock zone may have one or more subzones. The \( u^{th} \) clock zone is denoted as \( C(u) \). The \( l^{th} \) subzone in the \( u^{th} \) subzone is denoted as \( C_l(u) \) [11].

- **Clock step** - Clock step describes what happens in a certain period of time to a certain area. To make it clearer, it denotes that in a period of time, a desired control operation \( \phi_t \) is applied to a desired clock zone \( C(u) \). For example, clock step \( \varphi_v = \{(C(u); \phi_t)\}_v \) indicates that at clock step \( \varphi_v \), control operation \( \phi_t \) applies to clock zone \( C(u) \) [11].

- **Clock cycle** - The control operations are usually periodic, i.e. after a sequence of control operations \( \Phi = \phi_1\phi_2\phi_3\ldots\phi_n \), the next control operation \( \phi_{n+1} \) goes back to \( \phi_1 \). The period of time that a cycle of a periodic sequence control operations takes is called clock time. The clock is called a clock cycle.

- **Computation**
  
  - **Computational step** - To realize the desired function \( L(x_i) \), a function \( L \) applied to the \( \eta^{th} \) input \( x_i^{(\eta)} \) in the input sequence \(...x_i^{(\eta-1)}, x_i^{(\eta)}, x_i^{(\eta+1)}....\). The clock step that this function belongs to is called computational step. The \( k^{th} \) computational step of \( x_i^{(\eta)} \) in \( K \) clock steps is donated as \( c_k \). Clock steps \( \varphi \) apply to the circuit structure, usually involved with artifact \( A \) and other supporting computational subsystem \( \overline{A} \), while computational steps apply to inputs. At the end of each computational step, the collective state of all the computational elements is called computational state.

  - **Computational cycle** - All the computational steps apply to a single input \( x_i \) to realize the entire function \( L(x_i) \) is called a computational cycle. It’s usually denoted as \( \Gamma^{(\eta)} = c_1\ldots c_k\ldots c_K \) where \( c_K \) indicates that it takes \( K \)
clock steps to realize the entire function. In most of the cases, the amount of computational steps in a computational cycle is equal to the amount of the clock steps in the corresponding clock cycle. When comes to the definition of the entire function $L(x_i)$, it means from the first information of $x_i$ starts to appear (when $x_i$ starts to be loaded into artifact A) to the last information loss when the next computational cycle $\Gamma^{(\eta+1)}$ erases all the information of $x_i$ in artifact A [11].

3.2.5 Cost Analysis

Cost analysis is based on the irreversible information loss of data, i.e. it is based on an analysis of the information changes of computational elements along the data flow [11]. Therefore, information dynamics and dissipation bounds are needed in the evaluation, the definitions of which are as follows:

- Information dynamics - In order to analyze the information loss caused by data changes, data zones and subzones are defined to combine the same type of data based on data changing trends.

  - Data zones and subzones - In the $\eta^{th}$ computational clock, a data zone is all the representational elements $C$ that holds the information of input $x_i^\eta$ in the $k^{th}$ computational step, which could be denoted as data zone $D(c_k)$ where $c_k$ stands for the data zone of computational step $c_k$. Data zone consists of one or multiple data subzones depending on user definition. If only one data subzone exists, this data subzone is equal to the corresponding data zone. If multiple data subzones exist, these data subzones together constitute a data zone. Each data subzone is physically disjoint. The size of the data subzone is also defined by the user to make the cost analysis easier. The $w^{th}$ data subzone in data zone $D(c_k)$ is donated as $D_w(c_k)$ [11].
- Information loss - The information loss is the information lost from the information processing artifact A during computational cycles, i.e. we compare the data zones of the $D(c_{k-1})$ and the $D(c_k)$ and find out the representational elements $C$ that was erased in $D(c_k)$ from $D(c_{k-1})$. Moreover, the erased representational elements are the information loss during clock cycle $c_k$. Note that there would be some new representational elements showing up in $D(c_k)$. These new elements, which are not in $D(c_{k-1})$, don’t cause information loss.

- Dissipation bounds - We get the dissipation bounds by using the equation [21]

$$\triangle \langle E^B \rangle_k \geq -k_B T \ln(2) \times \triangle I^{R_\eta A_k} \tag{3.26}$$

where $k_B$ is Boltzmann constant, $T$ is temperature, $\triangle \langle E^B \rangle_k$ represents the energy dissipation from artifact A to the heat bath B in the $k^{th}$ computational step, and $\triangle I^{R_\eta A_k}$ is the information loss about $x_i^{(\eta)}$ in the $k^{th}$ computational step.

We could get the information loss by following steps. Assume we have discrete independent and identically distributed (IID) source

$$\{x\} = \{x_0, x_1, ..., x_{d-1}\} \tag{3.27}$$

with distribution

$$\{p\} = \{p(x_0), p(x_1), ..., p(x_{d-1})\} \tag{3.28}$$

Then it has Shannon entropy

$$H(X) = -\sum_{i=0}^{d-1} p(x_i) \log_2 p(x_i) \tag{3.29}$$
where $I(x_i) = -\log_2 p(x_i)$ is the information of $x_i$. Shannon entropy is a measure of the information in the output of the source $X$.

If this discrete IID source $X$ is sent to a channel $\mathcal{N}$, and get a series of output $Y$. Output $Y$ can be written as a collection of symbol of $\{y\} = \{y_0, y_1, \ldots, y_{r-1}\}$ with distribution $\{q\} = \{q(y_0), q(y_1), \ldots, q(y_{r-1})\}$. The channel $\mathcal{N}$ is not the physical non-ferromagnetic material channel we introduced in Chapter 2. It is the logic computation channel implementing the conditional distribution $p(Y|X)$—when the set of input $X$ is given, the probability of output set $Y$.

It is the information process artifact $A$, including representational element $C$ and non-representational element $\overline{C}$. For the ASL circuit, the channel $\mathcal{N}$ is the combination of the ferromagnetic material input-output magnets, and the non-ferromagnetic connection channel in Figure 2.1. No ground lead and voltage supply connection included.

We can then define the conditional entropy as

$$H(Y|X) = \sum_{i=0}^{d-1} p(x_i) H(Y|x_i) = -\sum_{i=0}^{d-1} \sum_{j=0}^{r-1} p_i q_{j|i} \log_2 q_{j|i}$$

(3.30)

where $p_i$ is short for $p(x_i)$, $q_{j|i}$ is the conditional probability that given the input is $x_i$, the output is $y_j$, and $H(Y|x_i)$ is the entropy of output $Y$ if one of the selected symbol $x_i$ in $X$ is sent to the channel.

The conditional entropy means the information in $Y$ that is not in $X$. Vice versa, if the output $Y$ is given and we need to calculate the entropy of input $X$ distribution, we have

$$H(X|Y) = \sum_{j=0}^{r-1} q(y_j) H(X|y_j) = -\sum_{j=0}^{r-1} \sum_{i=0}^{d-1} q_j p_{i|j} \log_2 p_{i|j}$$

(3.31)
where \( q_j \) is short for \( q(y_j) \), \( p_{ij} = \frac{p_i}{q_j} \) is the conditional probability that given the output is \( y_j \), the input is \( x_i \), \( H(X|y_j) \) is the entropy of input X, if one of the selected symbol \( y_j \) in \( Y \) is received from the channel.

Thus, we have the information loss

\[-\Delta I = H(X|Y) \quad (3.32)\]

, which means the information loss is the uncertainty of the channel input if the channel output is given. Substitute Equation 3.31 and Equation 3.32 to Equation 3.34, we get

\[
\Delta \langle E^B \rangle_k \geq -k_B T \ln(2) \times \Delta I^{R \eta A_k} \\
= k_B T \ln(2) \times H_k(X|Y) \\
= k_B T \ln(2) \times \sum_{j=0}^{r-1} q_j H(X|y_j) = -\sum_{j=0}^{r-1} \sum_{i=0}^{d-1} q_j p_{ij} \log_2 p_{ij} \quad (3.33)
\]

where \( k \) is the \( k^{th} \) computational step; \( \eta \) is the information loss about \( x_i^{(\eta)} \) in the \( k^{th} \) computational step; \( q_j \) is short for \( q(y_j) \); and \( p_{ij} = \frac{p_i}{q_j} \) is the conditional probability. By using this equation, we could calculate the irreversibility induced energy for all the computational steps and the total irreversibility induced energy is

\[
\Delta \langle E^B \rangle_{tot} = \sum_{k=1}^{n} \Delta \langle E^B \rangle_k \quad (3.34)
\]

where \( n \) is the total number of computational steps \( k \) in a computational cycle \( \Gamma^{(\eta)} \) (\( \eta^{th} \) is the \( \eta^{th} \) input \( x_i^{(\eta)} \) in the input sequence \( \ldots x_i^{(\eta-1)}, x_i^{(\eta)}, x_i^{(\eta+1)} \ldots \)).

3.2.6 Energy Efficiency

As it is said in Section 3.1.4, the irreversibility induced energy is an averaged input energy dissipation, i.e. the distribution rate of every input is taken into consideration.
in the calculation method; the result of it does not reflect an actual input pattern but shows a weighted averaged result of all possible patterns.
CHAPTER 4

ASL CIRCUIT STUDY: PRELIMINARY RESULTS AND DISCUSSION

In this chapter, preliminary results are presented for projected and limiting energy efficiencies of the ASL circuits introduced in Chapter 2. The analog circuits for the test of projected energy are built and tested with MATLAB [10]. The standard ASL model simulator is coded and authorized by Alawein and Fariborzi. In their paper [10], an ASL buffer/adder simulation circuit is provided in the MATLAB platform. Based on their codes, circuits with other logic functions such as full-adder and multiplexor are implemented and modified. The detailed simulation results are shown and discussed in this chapter. For the limiting energy efficiencies or the irreversible induced energy, the calculation results are base on physical information theory introduced in Chapter 3.

4.1 Simple Circuits: Buffer, Inverter and Latch

4.1.1 Buffer Simulation Results

As shown in figure 2.1, two basic ASL magnets form a buffer or an inverter. Depending on the supply voltage, the circuit can work as a buffer or an inverter. Negative supply voltage realizes a COPY function, which can be applied in the buffer circuit. Positive supply voltage realizes a NOT function, which could be applied in the inverter circuit.

Figure 4.1 is the projected simulation results of ASL buffer [10].

Figure 4.1 (a) is the magnetization simulation results. The red line indicates a negative voltage supply, which means the circuit function is a COPY. The input (blue
line) changes from -1 to 1 in every 2ns, while the output (green line) copies its status, which builds a basic buffer circuit.

Figure 4.1 (b) is the simulation results of spin currents. In our settings, the x-direction is related to the magnetization storage, so only spin currents in the x-direction are shown in figure (b). Currents in y and z-direction are not shown.

Figure 4.1 (c) shows the spin power changes. The instantaneous results in this simulation model are calculated by the current-voltage product at every time $E = I \times V \times t$. The unstable glitches are caused by the input currents and the feedback currents from the receiving magnet. The feedback currents are small compared to the main spin currents that torque the magnetization direction. The LLG model can calculate them. The integration of the power curve is the total energy of this period (8ns).

$$E = \int_0^t Pdt (t = 8\text{ns}) \quad (4.1)$$

The average value of $p_1$ is higher than $p_2$. It is because magnet 1 needs supply power to drive currents and send information, while magnet 2 only needs to receive information. Only half of magnet 2 is active. The other half it is in the idle state and will be active when the information needs to be sent to its next magnet, i.e. when magnet 2 works as a sending magnet.

The integration of $p_1$ is the energy of sending magnet $m_1$, while the integration of $p_2$ is the energy of receiving magnet $m_2$. The total energy is $e_1 + e_2 = (161.76 + 84.04) \times 10^{-15}J = 0.2458pJ$. This total energy result indicates that as a promising post-CMOS device, the static power of ASL circuits are in $10^{-10}J$ magnitude with a 10mV voltage supply.

Table 4.1 shows the ASL simulation parameters for the buffer [10]. These parameters will also be used in the following circuit simulations.
Table 4.1: ASL simulation parameters [10].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{FM}$</td>
<td>Thickness</td>
<td>3</td>
<td>nm</td>
</tr>
<tr>
<td>$W_{FM}$</td>
<td>Width</td>
<td>50</td>
<td>nm</td>
</tr>
<tr>
<td>$L_{FM}$</td>
<td>Length</td>
<td>100</td>
<td>nm</td>
</tr>
<tr>
<td>$\rho_{FM}$</td>
<td>Resistivity</td>
<td>56</td>
<td>nΩ·m</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Damping factor</td>
<td>0.01</td>
<td>-</td>
</tr>
<tr>
<td>$M_s$</td>
<td>Saturation magnetization</td>
<td>$1.414 \times 10^6$</td>
<td>A/m</td>
</tr>
<tr>
<td>$K_1$</td>
<td>First uniaxial anisotropy constant</td>
<td>$10 \times 10^4$</td>
<td>J/m$^3$</td>
</tr>
<tr>
<td>$H_K$</td>
<td>Anisotropy field</td>
<td>$1.26 \times 10^5$</td>
<td>A/m</td>
</tr>
<tr>
<td>$N_x, N_y, N_z$</td>
<td>Demagnetization factors</td>
<td>0.055, 0.89, 0.055</td>
<td>-</td>
</tr>
<tr>
<td>$P$</td>
<td>Polarization</td>
<td>0.5</td>
<td>-</td>
</tr>
<tr>
<td>$g^{\uparrow}$</td>
<td>Spin-up conductance</td>
<td>$0.42 \times 10^{15}$</td>
<td>S/m$^2$</td>
</tr>
<tr>
<td>$g^{\downarrow}$</td>
<td>Spin-down conductance</td>
<td>$0.38 \times 10^{15}$</td>
<td>S/m$^2$</td>
</tr>
<tr>
<td>Re{$g^{\uparrow\downarrow}$}</td>
<td>Real-part of the mixing conductance</td>
<td>$0.546 \times 10^{15}$</td>
<td>nΩ·S/m$^2$</td>
</tr>
<tr>
<td>Im{$g^{\uparrow\downarrow}$}</td>
<td>Imaginary-part of the mixing conductance</td>
<td>$0.015 \times 10^{15}$</td>
<td>S/m$^2$</td>
</tr>
<tr>
<td>$t_{NM1}$</td>
<td>Thickness</td>
<td>200</td>
<td>nm</td>
</tr>
<tr>
<td>$W_{NM1}$</td>
<td>Width</td>
<td>50</td>
<td>nm</td>
</tr>
<tr>
<td>$L_{NM1}$</td>
<td>Length</td>
<td>100</td>
<td>nm</td>
</tr>
<tr>
<td>$l_{sf,NM1}$</td>
<td>Spin-diffusion length</td>
<td>450</td>
<td>nm</td>
</tr>
<tr>
<td>$\rho_{NM1}$</td>
<td>Resistivity</td>
<td>16.7</td>
<td>nΩ·m</td>
</tr>
<tr>
<td>$t_{NM2}$</td>
<td>Thickness</td>
<td>200</td>
<td>nm</td>
</tr>
<tr>
<td>$W_{NM2}$</td>
<td>Width</td>
<td>50</td>
<td>nm</td>
</tr>
<tr>
<td>$L_{NM2}$</td>
<td>Length</td>
<td>100</td>
<td>nm</td>
</tr>
<tr>
<td>$l_{sf,NM2}$</td>
<td>Spin-diffusion length</td>
<td>600</td>
<td>nm</td>
</tr>
<tr>
<td>$\rho_{NM2}$</td>
<td>Resistivity</td>
<td>26.3</td>
<td>nΩ·m</td>
</tr>
</tbody>
</table>
Figure 4.1: Circuit simulation results of a two-magnet buffer [10]. (a) Magnetization; (b) spin current; (c) spin power.
Figure 4.2: Circuit simulation results of a two-magnet inverter. (a) Magnetization; (b) spin currents; (c) spin power.
4.1.2 Inverter

Figure 4.2 shows the simulation results of an inverter.

Similar to Figure 4.1(a), the red line in Figure 4.2(a) indicates a negative voltage supply, which means the circuit function is a NOT. We could see in the same figure that the input (blue line) changes from -1 to 1 in every 2ns. Moreover, the output (green line) generates states opposite to the blue line, which builds a basic inverter circuit.

The total energy is the integral of $p_1$ and $p_2$ curve. Total energy is $e_1 + e_2 = (130 + 66.75) \times 10^{-15} J = 0.1968pJ$

4.1.3 Latch

As shown in Figure 2.6, a latch is made of three individual basic ASL magnet unit. Depending on the supply voltage $V_{DD}$, the latch can realize a COPY or a NOT function. When the middle magnet supplies low voltage, the information transport (current transport) stops. In this case, the conductance channel between input (A) and output (B) magnet is opaque.

Figure 4.3(a), (c) and (e) are the simulation results when the supply voltage (not shown in the figure) are negative. In (a) we could see outputs copy the states of inputs.

Figure 4.3(b), (d) and (f) shows when the supply voltage (not shown in the figure) is positive, outputs invert the states of inputs.

The average power of $p_1$ and $p_2$ are higher than $p_3$. Because magnet 1 and magnet 2 acts as sending magnet and $p_3$ acts as the receiving magnet, which only half of the magnet is active. The total energy of NOT-function latch is $e_1 + e_2 + e_3 = (126.76 + 129.24 + 66.26) \times 10^{-15} J = 0.3223pJ$. The total energy of COPY-function latch is $e_1 + e_2 + e_3 = (161.37 + 160.57 + 83.46) \times 10^{-15} J = 0.4054pJ$. 


Figure 4.3: Circuit simulation results of three-magnet latches. One implements “buffer” function, the other implements “inverter” function. (a) The magnetization of the buffer; (b) Magnetization of the inverter; (c) Spin currents of the buffer; (d) Spin currents of the inverter; (e) Spin power of the buffer; (f) Spin power of the buffer.
4.2 Simplified Circuits Simulation Results: Buffer, Inverter, Latch

4.2.1 From Dynamic and Static Power to Projected Energy

From the circuit simulation above, we can see there are two contributions to the total power. One is dynamic power. It is caused by the feedback currents and gives slightly unstable fluctuation to the power curve. The other is the static power. It is caused by the supply voltage (the external supply) and generates the majority currents flow of an ASL device. Note here that this static power differs from the static leakage that plagues CMOS, in that it plays a functional rather than a parasitic role.

For the final results, what we concerned is the projected energy dissipation, which could be obtained by the integration of the combined power curve (the combined power curve of both dynamic power and static power). The combined power curves are the ones that we showed in Figure 4.1 and Figure 4.3, the integration of which will be compared with their the irreversibility induced energy counterparts. For convenient reason and decreasing the simulation time, we try to simplify the circuit model. Energy dissipation is clearly dominated by the static contribution to the total power. Calculation of the total power, including the feedback that determines the dynamic component, becomes computationally intensive and induces instabilities in the simulation for more complex circuits. This motivates simplification of the simulator.

From Section 3.1.1, we know that it is the Landau-Lifshitz-Gilbert equation (LLG equation) that describes the relationship between input and output spin currents and how the feedback currents form a self-consistency system. LLG equation is a physical character of ASL circuits, helping us to have a better understanding of spintronic devices. Base on this, we build a simplified model without the influence of the LLG equation and apply it in the following section to the ASL buffer, inverter, and latch. The results are shown and compared with the full-LLG model ones. From
the comparison, we could get an idea of how small the dynamic energy proportion takes in the whole energy cost and the capability of simplified ASL model.

### 4.2.2 Simplified Buffer and Inverter Simulation

Figure 4.4 shows the simulation results of simplified buffer and inverter. (a), (c) and (e) are results of the buffer. (a) is the spin magnetization changes in x direction, which also represents the circuit logic implementation. We could see from the figure that the output magnet $m_2$ just follows the statuses of input magnet $m_1$, which realizes COPY function of a buffer circuit. The spin power of magnet 1 and 2 are around $20\mu W$ and $10\mu W$, respectively.

Figure 4.4(b), (d), and (f) are results of the inverter. (b) is the spin magnetization changes in x direction, from which we could see the output magnet $m_2$ inverts the statuses of input magnet $m_1$.

According to the spin power, the total energy of a simplified buffer is $E = P_1t + P_2t = (163.97 + 85.07) \times 10^{-15}J = 249pJ$; the total energy of a simplified inverter is $E = P_1t + P_2t = (125.7 + 64.1) \times 10^{-15}J = 1898pJ$.

The buffer and the inverter has one input $m_1$; thus input patterns have two possibilities: 0 and 1. We run four clock cycles with input 0, 1, 0, 1, respectively. So the total energy consists of two sets of full input pattern 0 and 1. Divide the total energy by 2, and we get the effective energy or energy dissipation per input pattern for the simplified buffer and latch. For the buffer, effective energy is $E_{\text{effective}} = E_{\text{tot}}/2 = 249pJ/2 = 1245pJ$. For the inverter, effective energy is $E_{\text{effective}} = E_{\text{tot}}/2 = 1898pJ/2 = 949pJ$. The effective energy indicates how much energy an operation (an output pattern) consumes during the computation process. Thus, from the effective energy, we could calculate the energy efficiency, which is how many operations per Joule (or pJ) energy could realize. For the buffer, energy efficiency is $E_{\text{efficiency}} = 1/E_{\text{effective}} = 1/(0.1245pJ) = 8.032\text{operations/pJ}$. 
For the inverter, energy efficiency is \( E_{\text{efficiency}} = 1/E_{\text{effective}} = 1/(0.0949pJ) = 10.537\, \text{operations/pJ}. \)

### 4.2.3 Simplified Latch Simulation

Figure 4.5 shows the simulation results of a simplified latch implementing buffer function.

Figure 4.5 (a) and (b) are the magnetization changes. We could see that the output copies the statuses of input, and the middle magnet \( m_2 \) is transparent when the power supply is high.

Figure 4.5 (d) is the spin power. \( p_1, p_2, \) and \( p_3 \) are the spin power of magnet 1, magnet 2, and magnet 3, respectively. The results are similar to the results of the full-LLG-model buffer circuit (Figure 4.3(e)). The corresponding energy dissipation is \( E = E_1 + E_2 + E_3 = (163.97 + 163.97 + 85.07) \times 10^{-15}J = 0.413pJ. \) Also, we could change the supply voltage to positive and calculate the corresponding energy cost of the inverter function latch. The energy cost of the latch under positive power supply is \( E = E_1 + E_2 + E_3 = (125.71 + 125.71 + 64.10) \times 10^{-15}J = 0.3155pJ. \)

The buffer-function and the inverter-function latch has one input \( m_1. \) Thus their input patterns have two possibilities : 0 and 1. We run four clock cycles with input 0, 1, 0, 1, respectively. So the total energy consists of two sets of full input pattern 0 and 1. Divide the total energy by 2 and then by 4, we get the effective energy or the energy dissipation per pattern for the simplified buffer and latch. 2 is for the full input pattern set is repeated twice during the four clock cycle.

The effective energy for a simplified buffer-function latch is \( E_{\text{effective}} = E/2 = 0.413pJ/2 = 0.2065pJ; \) energy efficiency is \( E_{\text{efficiency}} = 1/E_{\text{effective}} = 1/(0.2065pJ) = 4.843\, \text{operations/pJ}. \) The effective energy for a simplified inverter is \( E_{\text{effective}} = E/2 = 0.3155pJ/2 = 0.1578pJ; \) energy efficiency is \( E_{\text{efficiency}} = 1/E_{\text{effective}} = 1/(0.1578pJ) = 6.337\, \text{operations/pJ}. \)
Figure 4.4: Results of simplified buffer and inverter. (a) The magnetization of the buffer; (b) Magnetization of the inverter; (c) spin currents of the buffer; (d) spin currents of the inverter; (e) spin power of the buffer; (f) spin power of the inverter.
Table 4.2: The energy cost of different circuits with and without LLG model.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>with LLG model</th>
<th>without LLG model</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer</td>
<td>0.2458pJ</td>
<td>0.249pJ</td>
<td>1.32%</td>
</tr>
<tr>
<td>Inverter</td>
<td>0.197pJ</td>
<td>0.19pJ</td>
<td>3.53%</td>
</tr>
<tr>
<td>Latch (COPY function)</td>
<td>0.4054pJ</td>
<td>0.413pJ</td>
<td>1.88%</td>
</tr>
<tr>
<td>Latch (NOT function)</td>
<td>0.3223pJ</td>
<td>0.3155pJ</td>
<td>2.09%</td>
</tr>
</tbody>
</table>

### 4.2.4 Comparison of the full-LLG Model and Simplified Model

In the previous sections, we tested several circuits under the full-LLG model and the simplified model. Table 4.2 shows the results of the energy cost of each circuit under different models.

From the Table 4.2, we can draw two conclusions. First, as shown in the previous calculation, the static power of ASL circuits are in $10^{-13} J$ magnitude with a 10mV voltage supply. It is a small energy cost compared to its CMOS counterparts.

Second, the difference between full-LLG-model circuits and simplified circuits are small (less than 5%). Thus, in the following experiments, we use simplified ASL circuits as samples and compare the results with the irreversibility induced energy dissipation.

### 4.3 Simulation of Larger Circuits: Majority Gate, Half-adder, and ALU

We have established a simplified model of the ASL device and tested several simple ASL circuits. In this section, we are proceeding to more complex circuits with the simplified model and calculate their projected energy dissipation. The circuits are the 3-input majority gate and the half adder, where the half adder is built with the 3-input majority gate unit.
Figure 4.5: Results of a simplified three-magnet latch, implementing buffer function. (a) and (b) Magnetization; (c) spin currents; (d) spin power.
4.3.1 Majority Gate

As discussed in Section 2.2, majority gate consists of four magnets: Three input magnets and one output magnet. According to the truth table 2.1, when the supply voltage is positive, a majority gate can be NOR gate or NAND gate.

Figure 4.6 shows the simulation results of one majority gate act as NOR gate and NAND gate.

Magnet 1, magnet 3, and magnet 4 are input magnets, while magnet 2 is output magnet. Among input magnet, magnet 3 is the control magnet. When the magnetization direction of \( m_3 \) is -1, or logic 0, the majority is a NAND gate. When the magnetization direction of \( m_3 \) is 1, or logic 1, the majority is a NOR gate.

Figure 4.6(c) and (f) are the spin power consumption. \( p_1, p_2, p_3 \) and \( p_4 \) are the spin power of charge currents of magnet \( m_1, m_2, m_3 \) and \( m_4 \), respectively.

According to the spin power, the total energy of a 3-input 4-magnet majority gate implementing NOR function is

\[
E = P_1 t + P_2 t + P_3 t + P_4 t = (213.66 + 213.66 + 213.66 + 93.92) \times 10^{-15} J = 0.735pJ;
\]

the total energy of a 3-input 4-magnet majority gate implementing NAND function is

\[
E = P_1 t + P_2 t + P_3 t + P_4 t = (229.10 + 229.10 + 229.10 + 92.06) = 0.779pJ.
\]

The NOR-function and the NAND-function majority gate has two inputs \( m_1 \) and \( m_4 \) (\( m_3 \) acts as a switch button switching between NAND and NOR function, which is fixed by us once the function is setup), thus their input patterns have four possibilities: 00, 01, 10 and 11. We run four clock cycles with all the possible input patterns 00, 01, 10, 11, respectively. So the effective energy or the energy dissipation per pattern is the total energy divided by 4 (for it has four input patterns and the effective energy is an input averaged value).

the effective energy of a 3-input 4-magnet majority gate implementing NOR function is

\[
E_{\text{effective}} = E/4 = 0.735pJ/4 = 0.1837pJ; \text{ energy efficiency is } E_{\text{efficiency}} = 1/E_{\text{effective}} = 1/(0.1837pJ) = 5.444\text{operations/pJ}.
\]
Table 4.3: The truth table of half adder with the illustration of majority-gate nand gate design.

<table>
<thead>
<tr>
<th>$m_1$</th>
<th>$m_2$</th>
<th>$m_6$ (Sum)</th>
<th>$m_7$ (Carry)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

input 4-magnet majority gate implementing NAND function is $E_{\text{effective}} = E/4 = 0.779pJ/4 = 0.195pJ$; energy efficiency is $E_{\text{efficiency}} = 1/E_{\text{effective}} = 1/(0.195pJ) = 5.128\text{operations/pJ}$.

4.3.2 Half Adder

Section 4.3.1 shows the basic unit of ASL device—the majority gate. Using majority gate as NAND or NOR gate, we could build almost all the logic circuits.

In this section, we build a half adder using NAND-function majority gates. The schematic is shown in Figure 4.7. Magnet 1 and 2 are inputs. Magnate 6 and 7 are outputs. Magnet 6 is sum while magnet 7 is carry. Table 4.3 is the truth table of the half adder. The simulation results are shown in Figure 4.8.

Each NAND gate has three input magnets and one output magnet. With total 5 NAND gates, there are $(3 + 1) \times 5 = 20$ magnets. Each magnet has its own power. Figure 4.9 shows the power consumption of each magnet.

Thus, we could calculate the total energy cost is $E = P_{\text{gate1}} + P_{\text{gate2}} + P_{\text{gate3}} + P_{\text{gate4}} + P_{\text{gate5}} = \sum_{i=1}^{20} P_i t = (221.0 + 221.0 + 93.17 + 214.04) \times 10^{-15} J + (229.09 + 92.82 + 221.15 + 213.25) \times 10^{-15} J + (229.09 + 92.82 + 221.15 + 213.25) \times 10^{-15} J + (211.58 + 98.02 + 219.52 + 219.52) \times 10^{-15} J + (210.68 + 103.67 + 209.70 + 209.70) \times 10^{-15} J = 3.744pJ$

To get the effective energy or the energy dissipation per pattern, we need to divide the total energy 3.744pJ by 4. The half adder has two inputs $m_1$ and $m_2$. Thus their input patterns have four possibilities: 00, 01, 10 and 11. We run four clock cycles with
Figure 4.6: Results of a simplified 3-input 4-magnet majority gate, implementing NAND gate or NOR gate. Input magnet: $m_1$, $m_3$ and $m_4$. Output magnet: $m_2$. $m_3$ acts as a switch button switching between NAND and NOR function. $m_1$ and $m_4$ act as the actual inputs for the NAND or NOR gate. (a), (b) and (c): NOR gate. (d), (e) and (f): NAND gate.
Figure 4.7: (a) Half adder schematic. Number 1~7 are the inputs and outputs of four NAND gates. Black and red lines are the circuit connections. The two colors are to make the schematic clearer, having no difference in function. (b) Half adder unit. The three images are all NAND gate. However, the second and third one are drawn in 3-input majority gate format.
Figure 4.8: Simulation results of the half adder. (a) Inputs. (b) Outputs.

all the possible input patterns 00, 01, 10, 11, respectively. So the energy efficiency is the total energy divided by 4 (for it has four input patterns and the energy efficiency is an input averaged value).

The projected effective energy of a half adder is $E_{eff} = E/4 = 3.744pJ/4 = 0.936pJ$; energy efficiency is $E_{efficiency} = 1/E_{effective} = 1/(0.936pJ) = 1.068 \text{ operations/pJ}$.

4.3.3 Arithmetic Logic Unit

Qi AN’s paper [24][23] reports an ASL implementation of a 5-input ALU based on two 5-input majority gates and one 3-input majority gate, implementing functions of full-adder, subtractor, multiplexor, increment, decrement, NAND and NOR.

Figure 4.10 [23] shows the 5-input majority layout and the ALU schematic. Table 4.4 shows the integrated functions of ALU and the detailed configuration of each terminal. The ALU we discussed here is a 1-bit standard ALU, which can realize the basic function of a 1-bit full adder (1-bit addend A, 1-bit addend B, and 1-bit carry-in $C_{in}$ with 1-bit sum and 1-bit carry-out as outputs), 1-bit subtractor, 1-bit multiplexor, and 1-bit increment/decrement.
Figure 4.9: Power simulation results of the half adder. (a)∼(e): NAND gate 1∼5.
Figure 4.10: ASL based ALU schematic [23]. (a) 5-input majority gate layout. (b) ASL based ALU schematic. (c) ALU symbol.
Figure 4.11: Results of an ASL-based ALU. (a)∼(c): full adder. Input A, B, and carry-in ($C_{in}$) are magnets 1, 3, and 4 respectively; Output sum and carry-out ($C_{out}$) are magnet 8 and 2. (d)∼(f): subtractor. Input A, B, and borrow-in ($B_{in}$) are magnets 1, 3, and 4 respectively; Output difference and borrow-out are magnet 8 and 16. (g)∼(h): multiplexor. Input A, B, and select signal are magnet 3, 4, and 1; Output is magnet 8. When the select signal is logic 1, input A is selected; When the select signal is logic 0, input B is selected.
Each magnet for the ALU has two configurable parts. One is the injected spin current, and the other is the magnetization state of the magnet. $I_{\text{inj}}$ represents injected spin currents. The possibilities are 0 (no spin current), P (positive spin currents), or N (negative spin current). The logic state is represented by the magnetization state of each magnet. The possibilities are logic 0 or 1, i.e. parallel or anti-parallel direction, or Z (don’t care).

For example, to set up a full-adder, we need the following configuration: No current injection for terminal A1, H, Z, and U; negative current injection for terminal A2, A3, B1, B2, and C; positive current injection for terminal M.

Figure 4.11 shows the simulation results of the full adder, subtractor, and multiplexer:

- Figure 4.11(a)∼(c): The simulation results of a full adder. As mentioned in the previous paragraph, it is a 1-bit full adder. One-bit input A (addend), B (addend), and carry-in ($C_{\text{in}}$) are represented by magnets 1, 3, and 4 respectively; Output sum and carry-out ($C_{\text{out}}$) are represented by magnet 8 and 2, respectively. The truth table of a full adder is shown in Table 2.2.

- (d)∼(f): The simulation results of a 1-bit subtractor. One-bit input A, B, and borrow-in ($B_{\text{in}}$) are represented by magnets 1, 3, and 4, respectively; Output difference and borrow-out are represented by magnet 8 and 16, respectively.

- (g)∼(h): The simulation results of a multiplexer. Input A, B, and select signal are represented by magnet 3, 4, and 1, respectively; Output is represented by magnet 8. When the select signal is logic 1, input A is selected; When the select signal is logic 0, input B is selected.

The ALU has 16 magnets. By integrating the power consumption of each magnet by the time, we get the effective energy or the energy dissipation per pattern of each function. For a full adder, the total energy contains eight clock cycles (16ns),
Table 4.4: ALU function table [23].

<table>
<thead>
<tr>
<th>Input</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>B1</th>
<th>B2</th>
<th>C</th>
<th>H</th>
<th>Z</th>
<th>U</th>
<th>M1</th>
<th>E_{tot}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>State</td>
<td>I_{inj}</td>
<td>State</td>
<td>I_{inj}</td>
<td>State</td>
<td>I_{inj}</td>
<td>State</td>
<td>I_{inj}</td>
<td>State</td>
<td>I_{inj}</td>
<td>State</td>
</tr>
<tr>
<td>Full-adder</td>
<td>0/1</td>
<td>0</td>
<td>0/1</td>
<td>N</td>
<td>0/1</td>
<td>N</td>
<td>0/1</td>
<td>N</td>
<td>0/1</td>
<td>N</td>
<td>X</td>
</tr>
<tr>
<td>Subtractor</td>
<td>0/1</td>
<td>P</td>
<td>0/1</td>
<td>N</td>
<td>0/1</td>
<td>N</td>
<td>0/1</td>
<td>N</td>
<td>0/1</td>
<td>N</td>
<td>X</td>
</tr>
<tr>
<td>Multiplexor</td>
<td>0/1</td>
<td>0</td>
<td>0/1</td>
<td>P</td>
<td>0/1</td>
<td>N</td>
<td>0/1</td>
<td>N</td>
<td>0/1</td>
<td>0</td>
<td>0/1</td>
</tr>
<tr>
<td>Increment</td>
<td>0/1</td>
<td>0</td>
<td>0/1</td>
<td>N</td>
<td>0/1</td>
<td>N</td>
<td>1</td>
<td>N</td>
<td>1</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>Decrement</td>
<td>0/1</td>
<td>P</td>
<td>0/1</td>
<td>N</td>
<td>0/1</td>
<td>N</td>
<td>1</td>
<td>N</td>
<td>1</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>Decrement</td>
<td>0/1</td>
<td>P</td>
<td>0/1</td>
<td>N</td>
<td>0/1</td>
<td>N</td>
<td>1</td>
<td>N</td>
<td>1</td>
<td>N</td>
<td>0</td>
</tr>
<tr>
<td>NAND3</td>
<td>0/1</td>
<td>0</td>
<td>0/1</td>
<td>P</td>
<td>0/1</td>
<td>0</td>
<td>0/1</td>
<td>P</td>
<td>0/1</td>
<td>0</td>
<td>0/1</td>
</tr>
<tr>
<td>NOR3</td>
<td>0/1</td>
<td>0</td>
<td>0/1</td>
<td>P</td>
<td>0/1</td>
<td>0</td>
<td>0/1</td>
<td>P</td>
<td>0/1</td>
<td>0</td>
<td>0/1</td>
</tr>
</tbody>
</table>

which allows the input magnets to check all the possible input patterns (from 000 to 111). The energy is $E_{tot} = 3.54pJ$, corresponding effective energy is $E_{effective} = E_{tot}/8 = 0.4425pJ$, and energy efficiency $E_{efficiency} = 1/E_{effective} = 1/(0.4425pJ) = 2.26 operations/pJ$.

Similarly, the subtractor has total energy of $E_{tot} = 3.206pJ$, corresponding effective energy $E_{effective} = E_{tot}/8 = 0.4pJ$, and energy efficiency $E_{efficiency} = 1/E_{effective} = 1/(0.4pJ) = 2.5 operations/pJ$. The multiplexor has total energy of $E_{tot} = 3.222pJ$, corresponding effective energy $E_{effective} = E_{tot}/8 = 0.403pJ$, and energy efficiency $E_{efficiency} = 1/E_{effective} = 1/(0.403pJ) = 2.481 operations/pJ$. The increment/decrement function has total energy of $E_{tot} = 0.795pJ$, corresponding effective energy $E_{effective} = E_{tot}/2 = 0.398pJ$, and energy efficiency $E_{efficiency} = 1/E_{effective} = 1/(0.398pJ) = 2.513 operations/pJ$. Here for the increment/decrement function, we divide total energy by 2 (instead of 8) to get effective energy or the energy dissipation per pattern because the number of effective inputs are two (A and B).

Table 4.5 shows the projected energy results of different ASL devices. Effective energy of half adder (non-ALU based) is higher than the full adder (ALU based) due to the circuit structure difference — the half adder has 20 magnets, and the ALU based full adder has 16 magnets.
Table 4.5: Projected energy results of different ASL devices.

<table>
<thead>
<tr>
<th></th>
<th>Projected Energy</th>
<th>Energy Efficiency (operations/pJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer</td>
<td>0.147pJ</td>
<td>8.032</td>
</tr>
<tr>
<td>Inverter</td>
<td>0.0949pJ</td>
<td>10.537</td>
</tr>
<tr>
<td>Latch (Buffer)</td>
<td>0.2065pJ</td>
<td>4.843</td>
</tr>
<tr>
<td>Latch (Inverter)</td>
<td>0.1578pJ</td>
<td>6.337</td>
</tr>
<tr>
<td>Majority gate (NAND)</td>
<td>0.195pJ</td>
<td>5.444</td>
</tr>
<tr>
<td>Majority gate (NOR)</td>
<td>0.1837pJ</td>
<td>5.128</td>
</tr>
<tr>
<td>Half adder</td>
<td>0.936pJ</td>
<td>1.068</td>
</tr>
<tr>
<td>Full adder</td>
<td>0.4425pJ</td>
<td>2.26</td>
</tr>
<tr>
<td>Subtractor</td>
<td>0.4pJ</td>
<td>2.5</td>
</tr>
<tr>
<td>Multiplexor</td>
<td>0.403pJ</td>
<td>2.481</td>
</tr>
<tr>
<td>increment/decrement</td>
<td>0.398pJ</td>
<td>2.513</td>
</tr>
</tbody>
</table>

4.4 Energy Efficiency Bounds for ASL Circuits from Physical Information Theory

4.4.1 From Projected Energy to Irreversibility Induced Energy

In section 4.1, section 4.2 and section 4.3, we discussed the circuit simulation results of ASL devices. In this section, physical-information-theoretic analysis will be used to the same circuits regarding the energy dissipation calculation. As introduced in Chapter 3, the physical information theory is different from analog circuit simulation. It only concerns the energy dissipation caused by the irreversible information loss during the logic computation. The other energy dissipation sources such as power supply, and transport process are excluded.

In order to apply physical information theory to ASL, we need to distinguish the irreversibility induced and projected sources of energy dissipation in computation via ASL. The irreversibility induced sources of heat dissipation are those sources that cannot be eliminated by improving the structure or material of the circuit, while the
associated practical sources can be minimized only to the extent that the structure, material, and other related can be optimized (which is unknowable). The physical information theory only applies to the heat dissipation caused by the irreversibility induced information loss, which is knowable and unavoidable for a given circuit and clocking scheme.

In ASL circuit, the supporting sources of heat dissipation are the currents tunneling through spacers and the leaky currents in the conductors, i.e. the currents that flow perpendicular in lateral channels and flow in lateral when they are supposed to go perpendicularly, as shown in Fig. 4.12. The irreversibility induced sources consist of two parts. One is the irreversible information loss from the realization of implemented logic function. The other one is the heat dissipation from currents that cannot be eliminated even if the device is perfect, i.e. these currents are fundamental for keeping the device functioning normally.

The physical decomposition of ASL is shown in Fig. 4.13. The green part is the artifact $A$. Orange part is the supporting subsystem $\overline{A}$. The blue part is the heat bath $B$, and the black part is the remote environment $\overline{B}$. 

Figure 4.12: Fundamental and supporting sources. Green arrows are charge currents while pink arrows are spin currents. Solid arrows are currents that make up fundamental sources while dash ones are supporting sources.
Figure 4.13: Physical decomposition of ASL magnet and its connecting channel, power supply, ground lead, and substrate. The green part is the part of the ferromagnetic material magnet, and the connection channel, corresponding to the artifact A. Orange part is the magnet and channel of other computing units, corresponding to the supporting subsystem $A$. The blue part is the non-ferromagnetic material substrate heat bath $B$, and the black part is the power contact and substrate remote environment $B$.

4.4.2 Efficiency Bounds for ASL Circuits: Detailed Application Example

We take the 2-bit ripple adder (Fig. 4.14) as an example to show how the physical information theory applies to ASL devices. The clock is provided at the bottom of Fig. 4.14. Blue line stands for CLK while green line stands for CLK-b. The first five figures of Fig. 4.14 are the schematic of 2-bit ripple adder in the $\eta$th full computational cycle. In computational step $c_1$, addends $A_1$ and $B_1$, and carry $C_1$ receive information from the previous stage, but don’t do any further computation because CLK is low. In computational step $c_2$, CLK is high. The computation in adder1 is done in this step and results are sent to $C_2$. In this scenario, we assume $A_2$ and $B_2$ are loaded at the same time when $C_2$ receives the result from the previous stage. Thus, at the end of $c_2$, information about this computational cycle are in $A_1$, $B_1$, $C_1$, $C_{c_1}$, $S_1$, $A_2$, $B_2$, and $C_2$, as shown in Fig. 4.14 ($c_2$). In computational step $c_3$, $A_1$, $B_1$, and $C_1$ receive inputs of $(\eta + 1)^{th}$ computational cycle, erasing the information of $\eta$th computational cycle, but the $(\eta + 1)^{th}$ inputs don’t do any further
computation because CLK is low, thus C_o1, and S1 still stores the information of \((\eta)^{th}\) computational cycle. Adder 2 completes its logic function in this step and sends its result to an external register which is not shown in Fig. 4.14. In computational step c_4, information in C_o1, S1, A2, B2, and C2 is erased because adder 1 rewrites them with the results of the \(\eta + 1\)th computational cycle. In computational step c_5, information in C_o2, S2, A3, B3, and C3 is replaced with the information from the \(\eta + 1\)th computational cycle. Up to now, all the information of \(\eta\)th computational cycle is erased from artifact A.

Process Abstraction: From the analysis above, the clock cycle has two phases—SWITCH \((\phi_1)\) and HOLD \((\phi_2)\). SWITCH \((\phi_1)\) and HOLD \((\phi_2)\) operate on the selected clock zones. When the supply voltage is high (or negative), SWITCH \((\phi_1)\) acts on the related clock zones to complete the logic function, changing the information of every representational element C according to the logic function \(L : x_i^{(\eta)} \rightarrow L(x_i^{(\eta)})\) and sending the results to next clock zones. When the supply voltage is zero, HOLD \((\phi_2)\) acts on the related clock zones, receiving information from the previous stage and changing the representational elements C contacted directly to the previous stage while other representational elements stay the same.

To understand process abstraction a step further, we can analyze state transformation for 2-bit ASL ripple adder.

First, at the time point that the \(\eta\)th input is about to go into the concerned information processing artifact A, there is no information about the referent \(R\) in the artifact A.

The referent \(R\) is correlated with the supporting computational subsystem \(\overline{A}\), which could be the buffer register connected to the artifact A or the previous stage registers that store the computing results. The state of the \(\eta\)th input referent \(R\) is represented as \(\hat{\rho}^{R_\eta} = \sum_{i=1}^{8} p_i |x_i^{R_\eta}\rangle\langle x_i^{R_\eta}|\) where \(|x_i^{R_\eta}\rangle\) are the orthogonal pure states
Figure 4.14: Data zones in different computational step c1~c5.
encoding the inputs \( \{ x_i \} = \{ A_{1i}, B_{1i}, C_{1i} \} \). Because \( A_{1i}, B_{1i}, \) or \( C_{1i} \) can be 0 or 1, there are \( 2^3 = 8 \) different pure states. \( p_i \) is the probability of each pure state.

After the \( \eta \)th referent \( R_\eta \) goes into the artifact \( A \), the entire closed system evolves according to Shrodinger equation. The evolving procedure of each computational steps \( c_k \) is shown in Table 4.6. For computational step \( c_3 \) and \( c_4 \), information loss, or heat dissipation is occurred. Thus, they have a restoration process \( \hat{U}^{\text{rest}}_{\text{diss}} = \hat{U}^{B\text{B}} \otimes I^{R_\eta A_\eta \overline{A}_k} \), which means the remote environment \( \overline{B} \) restores the heat bath \( B \) to the thermal equilibrium state.

Here we interpret the state transformation for the ASL adder in Table 4.6. At computational step \( c_1 \), in the beginning, inputs are not loaded into artifact \( A \), which means the referent \( R_\eta \) is only correlated with the supporting system \( \overline{A} \). Thus, \( \hat{\rho}^{R_\eta}_i \) only has tensor product with \( \hat{\rho}^{A_k}_i \) for \( i = 1 \) to 8. During the computation process, the data flows into artifact \( A_k \). Thus \( A_k \) is involved into the data zone and \( R_\eta, B, \overline{B} \) remain identical. In computational step \( c_2 \), the initial state involves in \( R_\eta, \overline{A}_k \) and \( A_k \) (\( \hat{\rho}^{0}_0 \) stands for the ending state of \( c_1 \)). Thus, we have \( \hat{\rho}^{R_\eta}_i, \hat{\rho}^{\overline{A}_k}_0, \) and \( \hat{\rho}^{A_k}_0 \) tensor together. The state transformation of \( c_2 \) is the same as \( c_1 \). In computational step \( c_3 \), all the \( x_i \)s are in in \( A_k \) and \( \overline{A}_k \), so \( \hat{\rho}^{R_\eta}_i, \hat{\rho}^{A_k}_{1,i} \) and, \( \hat{\rho}^{\overline{A}_k}_{1,i} \) tensor together for the initial state. In the state transformation, we send information to the next stage (\( \overline{A}_k \)) and information in \( A_1, B_1, \) and \( C_1 \) are erased, leading to an irreversible information loss in this stage. Thus we have heat dissipation from \( A_k \) to \( B \) and \( \overline{A} \) (\( \hat{U}^{A_k B\overline{A}} \)) and from \( B \) to \( \overline{B} \) (\( \hat{U}^{B\overline{B}}_{\text{diss}} = \hat{U}^{B\overline{B}} \otimes I^{R_\eta A_\eta \overline{A}_k} \)). Here, \( \hat{U}^{A_k \overline{A}_k} \) also stands for \( A_k \) is starting to disconnect from \( \overline{A}_k \) and finish the disconnection by the end of \( c_3 \). In \( c_4 \), data \( x_i \) or \( L(x_i) \) is in the referent, artifact \( A \), receiving magnet \( \overline{A} \) and the greater environment. So \( \hat{\rho}^{R_\eta}, \hat{\rho}^{A_k}_{2,i}, \hat{\rho}^{\overline{A}_k}_{2,i} \), and \( \hat{\rho}^{B}_{2,i} \) tensor together. In the state transformation, the data in \( S_1, C_{0,1}, A_2, B_2, \) and \( C_2 \) are erased, leading to an irreversible information loss. Thus we have heat dissipation from \( A_k \) to \( B \) (\( \hat{U}^{A_k B} \)) and from \( B \) to \( \overline{B} \) (\( \hat{U}^{B\overline{B}}_{\text{diss}} = \hat{U}^{B\overline{B}} \otimes I^{R_\eta A_\eta \overline{A}_k} \)). In computational step \( c_5 \), data \( x_i \) or \( L(x_i) \) is in the referent, artifact \( A \), receiving magnet
Table 4.6: State transformation for the ASL adder.

<table>
<thead>
<tr>
<th>Computational Step</th>
<th>Initial State</th>
<th>State Transformation</th>
<th>Control Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>c₁</td>
<td>( \hat{\rho}<em>0 = (\sum</em>{i=1}^{n} \hat{\rho}_0^A \otimes \hat{\rho}_0^{A_k} \otimes \hat{\rho}_0^\beta \otimes \rho_0^\beta) \hat{\xi} )</td>
<td>( \hat{\rho}_1 = \hat{U}^{\text{rest}} \hat{U}_1 \hat{\rho}_0 \hat{U}_1^{\text{rest}} )</td>
<td>( \hat{U}<em>1 = \hat{U}^A \rho_0 \otimes \hat{I}</em>{\rho_0^\beta} \hat{\xi} )</td>
</tr>
<tr>
<td>c₂</td>
<td>( \hat{\rho}<em>1 = (\sum</em>{i=1}^{n} \hat{\rho}_1^A \otimes \hat{\rho}_1^{A_k} \otimes \hat{\rho}_1^\beta \otimes \rho_1^\beta) \hat{\xi} )</td>
<td>( \hat{\rho}_2 = \hat{U}^{\text{rest}} v_1 \hat{\rho}_1 \hat{U}_2^{\text{rest}} )</td>
<td>( \hat{U}<em>2 = \hat{U}^A \rho_0 \otimes \hat{I}</em>{\rho_0^\beta} \hat{\xi} )</td>
</tr>
<tr>
<td>c₃</td>
<td>( \hat{\rho}<em>2 = (\sum</em>{i=1}^{n} \hat{\rho}_2^A \otimes \hat{\rho}_2^{A_k} \otimes \hat{\rho}_2^\beta \otimes \rho_2^\beta) \hat{\xi} )</td>
<td>( \hat{\rho}_3 = \hat{U}^{\text{rest}} v_2 \hat{\rho}_2 \hat{U}_3^{\text{rest}} )</td>
<td>( \hat{U}<em>3 = \hat{U}^A \rho_0 \otimes \hat{I}</em>{\rho_0^\beta} \hat{\xi} )</td>
</tr>
<tr>
<td>c₄</td>
<td>( \hat{\rho}<em>3 = (\sum</em>{i=1}^{n} \hat{\rho}_3^A \otimes \hat{\rho}_3^{A_k} \otimes \hat{\rho}_3^\beta \otimes \rho_3^\beta) \hat{\xi} )</td>
<td>( \hat{\rho}_4 = \hat{U}^{\text{rest}} u_4 \hat{\rho}_3 \hat{U}_4^{\text{rest}} )</td>
<td>( \hat{U}<em>4 = \hat{U}^A \rho_0 \otimes \hat{I}</em>{\rho_0^\beta} \hat{\xi} )</td>
</tr>
<tr>
<td>c₅</td>
<td>( \hat{\rho}<em>4 = (\sum</em>{i=1}^{n} \hat{\rho}_4^A \otimes \hat{\rho}_4^{A_k} \otimes \hat{\rho}_4^\beta \otimes \rho_4^\beta) \hat{\xi} )</td>
<td>( \hat{\rho}_5 = \hat{U}^{\text{rest}} u_5 \hat{\rho}_4 \hat{U}_5^{\text{rest}} )</td>
<td>( \hat{U}<em>5 = \hat{U}^A \rho_0 \otimes \hat{I}</em>{\rho_0^\beta} \hat{\xi} )</td>
</tr>
</tbody>
</table>

\( \overline{A} \) and the greater environment. So \( \hat{\rho}_0^A, \hat{\rho}_A^{A_k}, \hat{\rho}_3^{A_k}, \) and \( \hat{\rho}_3^\beta \) tensor together. In state transformation, data in S2 and \( C_o \) are erased. However, the information loss of this data loss belongs to the next computational cycle. No heat dissipation will be made in this cycle. We only have control operation \( \hat{U}^{A_k \overline{A}} \) to disconnect of \( x_i \) between \( A_k \) and \( \overline{A}_k \).

Operational decomposition: Because the clock cycle \( \Phi \) has two phases \( \phi_1 \) and \( \phi_2 \), it can be written as \( \Phi = \varphi_1 \varphi_2 \) where \( \varphi \) is the clock step, with

\[
\varphi_1 : \{(C(1); \phi_2), (C(2); \phi_1)\} \quad (4.2)
\]

\[
\varphi_2 : \{(C(1); \phi_1), (C(2); \phi_2)\} \quad (4.3)
\]

where \( C(1) \) and \( C(2) \) are the clock zones, representing the representational elements in adder 1 and 2. For a single input \( x_i^{(n)} \), computational cycle \( \Gamma \) requires three full clock cycles, which could be described as

\[
\Gamma = c_1 c_2 c_3 c_4 c_5 = \varphi_1^{(1)} \varphi_2^{(1)} \varphi_1^{(2)} \varphi_2^{(2)} \varphi_1^{(3)} \quad (4.4)
\]

Now, we have a clear understanding of process abstraction. It is a qualitative analysis of information loss. We will move on to the quantitative analysis of information loss — the cost analysis.

Cost Analysis: First, denote receiving magnets A, B, and C as clock subzones \( C_1(u) \) for each clock zone \( C(u) \). Denote magnets S, and \( C_o \) as clock subzones \( C_2(u) \)
for each clock zone $C(u)$. Then the $k^{th}$ data zone $D(k)$ with the input $x^\eta_i$ in the $\eta^{th}$ computational cycle could be written as

$$D(c_1) = C_1(1)$$ (4.5)

$$D(c_2) = C_1(1) \cup C_2(1) \cup C_1(2)$$ (4.6)

$$D(c_3) = C_1(1) \cup C_2(1) \cup C_1(2)$$ (4.7)

$$D(c_4) = C_2(2)$$ (4.8)

$$D(c_5) = \emptyset$$ (4.10)

To calculate the physical information loss, note that only computational step $c_3$ and $c_4$ have non-zero information loss. Assuming inputs have the same rate to be 1 or 0, i.e. 50% input is 1, and 50% input is 0. For the first ripple-adder, we have inputs and outputs status in Table 4.7. Note that because we apply a positive voltage to the magnet, outputs are the NOT logic to their original values.

From Table 4.7, we can see that the outputs $\overline{C_0}$ and $\overline{S_1}$ have four types: 00, 01, 10, 11. Their rates are shown in Table 4.8. From Table 4.7 and Table 4.8 we can see that output pattern 00 only have one corresponding input pattern 000. The same with output pattern 11, which only has one corresponding input pattern 111. So no information loss during the data transport if inputs are 000 or 111. For the other six inputs pattern, we can use the method in Chapter 3 to get the result.

For the second adder, input rate of $C_2$ is fixed by the first stage calculation (see Table 4.7 and Table 4.8). Input rates of A2 and B2 are a uniform distribution, i.e. input A or B has 50% to be 1 and 50% to be 0. Thus, we have inputs and outputs of the second adder as Table 4.9.
Table 4.7: Inputs and outputs of the first adder.

<table>
<thead>
<tr>
<th>Rate</th>
<th>A1</th>
<th>B1</th>
<th>C1</th>
<th>C₀</th>
<th>S²</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1/8</td>
</tr>
<tr>
<td>1/8</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1/8</td>
</tr>
<tr>
<td>1/8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1/8</td>
</tr>
<tr>
<td>1/8</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1/8</td>
</tr>
<tr>
<td>1/8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1/8</td>
</tr>
<tr>
<td>1/8</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1/8</td>
</tr>
<tr>
<td>1/8</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1/8</td>
</tr>
<tr>
<td>1/8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1/8</td>
</tr>
</tbody>
</table>

From Table 4.9, we can get the outputs rates of the second adder as Table 4.10. Similar to the first ripple-adder, output pattern 00 only have one corresponding input pattern 000.

Moreover, output pattern 11 only has one corresponding input pattern 111. No information loss in these two kinds of inputs pattern. For the rest six input patterns, we can do the following calculation.

Table 4.8: Outputs of the first adder.

<table>
<thead>
<tr>
<th>Output</th>
<th>C₀</th>
<th>S²</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>C₀2</td>
<td>0</td>
<td>0</td>
<td>1/8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3/8</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3/8</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1/8</td>
</tr>
</tbody>
</table>

According to physical information theory, the heat dissipating for the \( k \)th computational step is
Table 4.9: Inputs and outputs of the second adder.

<table>
<thead>
<tr>
<th>Rate</th>
<th>A2</th>
<th>B2</th>
<th>C2</th>
<th>Oc2</th>
<th>O2</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1/16</td>
</tr>
<tr>
<td>3/16</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3/16</td>
</tr>
<tr>
<td>3/16</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>3/16</td>
</tr>
<tr>
<td>1/16</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1/16</td>
</tr>
<tr>
<td>1/16</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1/16</td>
</tr>
<tr>
<td>3/16</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3/16</td>
</tr>
<tr>
<td>3/16</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3/16</td>
</tr>
<tr>
<td>1/16</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1/16</td>
</tr>
</tbody>
</table>

Table 4.10: Outputs of the second adder.

<table>
<thead>
<tr>
<th>Output</th>
<th>Oc2</th>
<th>O2</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1/16</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>7/16</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>7/16</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1/16</td>
<td></td>
</tr>
</tbody>
</table>
\[ \Delta \langle E^B \rangle_k \geq \sum_{L_w(k)} -k_B T \ln(2) \Delta I_{R_e L_w} \] (4.11)

In step 3, \( p_i = \frac{1}{8} \), and \( q_{ji} = \frac{p_i}{q_j} = \frac{1}{8} \times \frac{8}{3} = \frac{1}{3} \). The information loss is \( -\Delta I = H(X|Y) = \sum_{j=0}^{r-1} q(y_j)H(X|y_j) = -\sum_{j=0}^{r-1} \sum_{i=0}^{d-1} q_{ji}p_{i|j}\log_2 p_{i|j} = -2 \times \frac{1}{8} \times 3 \times \frac{1}{3} \times 3 \times \log_2 \frac{1}{3} = 1.19 \). Similarly, in step 4, the information loss is \( -\Delta I = H(X|Y) = \sum_{j=0}^{r-1} q(y_j)H(X|y_j) = -\sum_{j=0}^{r-1} \sum_{i=0}^{d-1} q_{ji}p_{i|j}\log_2 p_{i|j} = -2 \times \frac{7}{16} \times (2 \times \frac{3}{4}\log_2 \frac{3}{4} + \frac{1}{7}\log_2 \frac{1}{7}) = 1.27 \). Thus, the information loss, i.e. the heat dissipation in computational step 3 and 4 are

\[ \Delta \langle E^B \rangle_3 \geq 1.19k_B T \ln(2) \] (4.12)

and

\[ \Delta \langle E^B \rangle_4 \geq 1.27k_B T \ln(2) \] (4.13)

The total information loss is

\[ \Delta \langle E \rangle_{TOT} = \sum_{k=1}^{6} \Delta E(c-k) = \sum_{k=1}^{6} \Delta \langle E^b \rangle_k = \Delta \langle E^B \rangle_3 + \Delta \langle E^B \rangle_4 = 2.46k_B T \ln(2) \] (4.14)

For Boltzmann constant \( k_B = 1.38 \times 10^{-23} m^2 kgs^{-2} K^{-1} \), the total energy cost is

\[ \Delta \langle E \rangle_{TOT} = 2.46k_B T \ln(2) = 2.46 \times 1.38 \times 10^{-23} \times \ln(2) = 2.35 \times 10^{-23} J \]

4.4.3 Efficiency Bounds for Buffer, Inverter, Latch, Majority Gate, and Half Adder

We illustrated the physical information calculation steps in in the previous section. Now we can apply it to the ASL circuits in section 4.1, section 4.2 and section 4.3.

- Buffer, inverter and latch: Buffer, inverter and latch implement simple logic function. They only have one input magnet and corresponding one output magnet, which means input X only has one possible output Y related to it. No matter the function is a NOT or a COPY, the input and output are paired
together. No 1 to N or N to 1 situation exists. This means all information from the input to the output is reserved. No information is lost.

Table 4.11 shows the inputs and output of ASL buffer, inverter and latch. We could see that for one certain output, only one input corresponds, which means the conditional probability \( q_{i|j} = \frac{p_i}{q_j} = \frac{1}{2} = 1 \), the entropy \( H(X|y_j) = log_2 p_{i|j} = log_2 1 = 0 \).

Thus the information loss \(-\Delta I = H(X|Y) = -\sum_{j=0}^{r-1} \sum_{i=0}^{d-1} q_{ij} p_{i|j} log_2 p_{i|j} = 0\).

- Majority gate: The 3-input majority has 3 inputs and one output. As shown in Table 2.1, the 3-input majority gate can be a NAND gate or a NOR gate. The input rates and corresponding output rates are shown in Table 4.12, \( m_1 \), \( m_2 \) and \( m_3 \) represent the three inputs magnets, and \( m_4 \) represents the output magnet.

Thus, the information loss for NAND function majority gate is \(-\Delta I = H(X|Y) = -\sum_{j=0}^{r-1} \sum_{i=0}^{d-1} q_{ij} p_{i|j} log_2 p_{i|j} = \frac{3}{4} \times \frac{1}{3} \times 3 \times log_2 \frac{1}{3} = 1.19\). The information loss for NOR gate is \(-\Delta I = H(X|Y) = -\sum_{j=0}^{r-1} \sum_{i=0}^{d-1} q_{ij} p_{i|j} log_2 p_{i|j} = \frac{3}{4} \times \frac{1}{3} \times 3 \times log_2 \frac{1}{3} = 1.19\). The total energy cost is \( \Delta \langle E \rangle_{TOT} = 1.19 k_B T ln(2) = 1.19 \times 1.38 \times 10^{-23} \times ln(2) = 1.14 \times 10^{-23} J \) for both NAND gate and NOR gate.

Because the total irreversible induced energy here is already averaged to the per pattern value, energy efficiency \( E_{efficiency} = 1/E_{tot} = 1/(1.14 \times 10^{-23} J) = 8.772 \times 10^{22} \text{ operations/J} \).

- Half adder: The half adder schematic is shown in Figure 4.7. Exchanging each NAND gate with the 3-input majority gate unit, we get the detailed half adder layout, as shown in Figure 4.15. The half adder is made of five NAND gates. Depending on the information flow, we briefly divide it into three stages.

In the first stage, inputs go into the first NAND gate (colored in blue), and get the first interim output \( m_3 \). In the second stage (colored in blue), interim
output goes into the next three NAND gates, getting one final output Carry and two interim outputs $m_6$ and $m_9$. In the third stage (colored in yellow), the two interim outputs go into the last NAND gate, getting the final output SUM. According to the truth table, all of the majority gates have one magnet served with logic zero (negative voltage) to get the NAND gate function.

The information loss occurs when the information stored in stage 1, 2 and 3 is erased.

For the first stage, the information loss is the same with that of a NAND-function majority gate. 

$$-\Delta I = H(X|Y) = -\sum_{j=0}^{r-1} \sum_{i=0}^{d-1} q_{ji} p_{ij} \log_2 p_{ij} = \frac{3}{4} \times \frac{1}{3} \times 3 \times \log_2 \frac{1}{3} = 1.19.$$ 

For the second stage, the inputs and the corresponding output rates are shown in Table 4.13. The output magnet $mx_6$, $mx_9$ and $mx_{12}$ have one to one output patterns with input, i.e. given one certain output $y$, only one certain input $x$ is related to it. Thus the information loss for the second stage is 0.

For the third stage the inputs and the corresponding output rates are shown in Table 4.14. 

$$-\Delta I = H(X|Y) = -\sum_{j=0}^{r-1} \sum_{i=0}^{d-1} q_{ji} p_{ij} \log_2 p_{ij} = \frac{1}{2} \times \frac{1}{2} \times 2 \times \log_2 \frac{1}{2} = 0.5.$$ 

Thus the total information loss of the three stages is $1.19 + 0.5 = 1.69$. The corresponding energy cost is $\Delta\langle E\rangle_{TOT} = 1.69 k_B T \ln(2) = 1.69 \times 1.38 \times 10^{-23} \times \ln(2) = 1.61 \times 10^{-23} J$. Energy efficiency $E_{\text{efficiency}} = 1/E_{\text{tot}} = 1/(1.61 \times 10^{-23} J) = 6.211 \times 10^{22} \text{operations}/J$.

### 4.4.4 Efficiency Bounds for ALU

For all the circuits calculated above, the state transformation is shown in Table 4.15. To calculate the energy efficiency, we listed the inputs and outputs rates for each function.
Table 4.11: Outputs of buffer, inverter and latch.

<table>
<thead>
<tr>
<th>Rate</th>
<th>Input</th>
<th>buffer</th>
<th>latch</th>
<th>inverter</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 4.12: Inputs and outputs of the majority gate.

<table>
<thead>
<tr>
<th>Rate</th>
<th>Input</th>
<th>Output</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0</td>
<td>0</td>
<td>NAND</td>
</tr>
<tr>
<td>0.25</td>
<td>0</td>
<td>1</td>
<td>NAND</td>
</tr>
<tr>
<td>0.25</td>
<td>1</td>
<td>0</td>
<td>NAND</td>
</tr>
<tr>
<td>0.25</td>
<td>1</td>
<td>1</td>
<td>NAND</td>
</tr>
</tbody>
</table>

Table 4.13: Inputs and outputs of the half adder middle stage.

<table>
<thead>
<tr>
<th>Rate</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0.25</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0.25</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0.25</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 4.14: Inputs and outputs of the half adder middle stage.

<table>
<thead>
<tr>
<th>Rate</th>
<th>$m_{14}$</th>
<th>$m_{13}$</th>
<th>$m_{15}$ (SUM)</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{4}$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>$\frac{1}{4}$</td>
</tr>
<tr>
<td>$\frac{1}{4}$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>$\frac{1}{4}$</td>
</tr>
<tr>
<td>$\frac{1}{2}$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>$\frac{1}{2}$</td>
</tr>
</tbody>
</table>

Figure 4.15: Half adder layout.
• Full adder: According to the ALU schematic Fig. 4.10 and function configurations Table 4.4, the full adder has two computation stages. In the first stage we get the carry-out from majority gate M1 and M3, where M1 has the interim outcome ready for M2 and one of the final outputs \( C_{\text{out}} \) (F1). Here the output of M3 (F2) doesn’t have any logic meaning. In the second stage, we get the final sum of A, B, and \( C_{\text{in}} \). The detailed truth table is shown in Table 2.2. The inputs are evenly distributed with the rate of \( \frac{1}{8} \).

For the first stage, \(-\Delta I = H(X|Y) = -\sum_{j=0}^{r-1}\sum_{i=0}^{d-1} q_{ji}p_{ij}\log_2 p_{ij} = 2 \times \frac{1}{2} \times \frac{1}{4} \times 4\times\log_2 \frac{1}{4} = 2\). For the second stage, \(-\Delta I = H(X|Y) = -\sum_{j=0}^{r-1}\sum_{i=0}^{d-1} q_{ji}p_{ij}\log_2 p_{ij} = 2 \times \frac{1}{2} \times \frac{1}{4} \times 4 \times \log_2 \frac{1}{4} = 2\). Thus the total information loss of the three stages is \(2 + 2 = 4\). The corresponding energy cost is \(\Delta\langle E\rangle_{\text{TOT}} = 4k_BT\ln(2) = 4 \times 1.38 \times 10^{-23} \times \ln(2) = 3.87 \times 10^{-23} \text{J}\). Energy efficiency \(E_{\text{efficiency}} = 1/E_{\text{tot}} = 1/(3.87 \times 10^{-23}) = 2.584 \times 10^{22} \text{operations/J}\).

• Subtractor: According to the ALU schematic Fig. 4.10 and function configurations Table 4.4, the subtractor has two computation stages, which is similar to the configuration of the full adder. In the first stage we get the borrow-out from majority gate M1 and M3, where M1 has the interim outcome ready for M2 and one of the final outputs \( B_{\text{out}} \) (F1). Here the output of M1 (F1) doesn’t have any logic meaning. In the second stage, we get the final difference of A, B, and \( B_{\text{in}} \). The detailed truth table is shown in Table 4.16. The inputs are evenly distributed with the rate of \( \frac{1}{8} \).

For the first stage, \(-\Delta I = H(X|Y) = -\sum_{j=0}^{r-1}\sum_{i=0}^{d-1} q_{ji}p_{ij}\log_2 p_{ij} = 2 \times \frac{1}{2} \times \frac{1}{4} \times 4\times\log_2 \frac{1}{4} = 2\). For the second stage, \(-\Delta I = H(X|Y) = -\sum_{j=0}^{r-1}\sum_{i=0}^{d-1} q_{ji}p_{ij}\log_2 p_{ij} = 2 \times \frac{1}{2} \times \frac{1}{4} \times 4 \times \log_2 \frac{1}{4} = 2\). Thus the total information loss of the three stages is \(2 + 2 = 4\). The corresponding energy cost is \(\Delta\langle E\rangle_{\text{TOT}} = 4k_BT\ln(2) = \)
$4 \times 1.38 \times 10^{-23} \times ln(2) = 3.87 \times 10^{-23} J$. Energy efficiency $E_{efficiency} = 1/E_{tot} = 1/(3.87 \times 10^{-23} J) = 2.584 \times 10^{22} operations/J$.

- **Multiplexor**: According to the ALU schematic Fig. 4.10 and function configurations Table 4.4, the multiplexor has two computation stages. In the first stage, we get the interim results from majority gate M1. Majority gate M3 also process information in this stage, but the output F2 is not needed for this function. In the second stage, we get the final output. When the select signal A is logic 1, input B is selected. When the signal is logic 0, input C is selected. For the first stage, $-\Delta I = H(X|Y) = -\sum_{j=0}^{r-1} \sum_{i=0}^{d-1} q_j p_{ij} \log_2 p_{ij} = \frac{3}{8} \times \frac{1}{3} \times 3 \times \log_2 \frac{1}{3} + \frac{5}{8} \times \frac{1}{5} \times 5 \times \log_2 \frac{1}{5} = 0.594 + 1.451 = 2.045$. For the second stage, $-\Delta I = H(X|Y) = -\sum_{j=0}^{r-1} \sum_{i=0}^{d-1} q_j p_{ij} \log_2 p_{ij} = 2 \times \frac{1}{2} \times \frac{1}{4} \times 4 \times \log_2 \frac{1}{4} = 2$. Thus the total information loss of the three stages is $2.045 + 2 = 4.045$. The corresponding energy cost is $\Delta \langle E \rangle_{TOT} = 4k_B T \ln(2) = 4.045 \times 1.38 \times 10^{-23} \times \ln(2) = 3.87 \times 10^{-23} J$. Energy efficiency $E_{efficiency} = 1/E_{tot} = 1/(3.87 \times 10^{-23} J) = 2.584 \times 10^{22} operations/J$.

- **Increment/Decrement**: The truth table of increment and decrement is shown in Table 4.17. Input B and C are control signals, and A is the actual input. When the implement function is increment, the output F1F0 realizes function $F1F0 = A + 1$. When the implement function is decrement, the output F2F0 realizes function $F1F0 = A - 1$. From the truth table, we could see the input and output pattern has one to one relationship. Thus the information loss of the increment and decrement function is 0.

### 4.4.5 Simulation-Based Projections vs Fundamental Bounds

In previous sections, we used two methods to calculate some ASL circuits and get their energy dissipation (or heat dissipation). Results for projected energy dissipation
Table 4.15: State transformation for the buffer, inverter, latch, majority gate, half adder, and ALU.

<table>
<thead>
<tr>
<th>Computational Step</th>
<th>Initial State</th>
<th>State Transformation</th>
<th>Control Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Buffer and Inverter</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td>$\rho_4 = \bigotimes_{i=1}^{2} \rho_{4_1} \otimes \rho_{4_{A1}} \otimes \rho_{4_{B1}} \otimes \rho_{4_{C1}}$</td>
<td>$\rho_4 = U_{rest} U_4 \rho_4 U_{rest}$</td>
<td>$U_4 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_2$</td>
<td>$\rho_1 = \bigotimes_{i=1}^{2} \rho_{1_2} \otimes \rho_{1_{A1}} \otimes \rho_{1_{B1}} \otimes \rho_{1_{C1}}$</td>
<td>$\rho_1 = U_{rest} U_1 \rho_1 U_{rest}$</td>
<td>$U_1 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_3$</td>
<td>$\rho_2 = \bigotimes_{i=1}^{2} \rho_{2_1} \otimes \rho_{2_{A1}} \otimes \rho_{2_{B1}} \otimes \rho_{2_{C1}}$</td>
<td>$\rho_2 = U_{rest} U_2 \rho_2 U_{rest}$</td>
<td>$U_2 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_4$</td>
<td>$\rho_3 = \bigotimes_{i=1}^{2} \rho_{3_1} \otimes \rho_{3_{A1}} \otimes \rho_{3_{B1}} \otimes \rho_{3_{C1}}$</td>
<td>$\rho_3 = U_{rest} U_3 \rho_3 U_{rest}$</td>
<td>$U_3 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td><strong>Latch</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td>$\rho_4 = \bigotimes_{i=1}^{2} \rho_{4_1} \otimes \rho_{4_{A1}} \otimes \rho_{4_{B1}} \otimes \rho_{4_{C1}}$</td>
<td>$\rho_4 = U_{rest} U_4 \rho_4 U_{rest}$</td>
<td>$U_4 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_2$</td>
<td>$\rho_1 = \bigotimes_{i=1}^{2} \rho_{1_2} \otimes \rho_{1_{A1}} \otimes \rho_{1_{B1}} \otimes \rho_{1_{C1}}$</td>
<td>$\rho_1 = U_{rest} U_1 \rho_1 U_{rest}$</td>
<td>$U_1 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_3$</td>
<td>$\rho_2 = \bigotimes_{i=1}^{2} \rho_{2_1} \otimes \rho_{2_{A1}} \otimes \rho_{2_{B1}} \otimes \rho_{2_{C1}}$</td>
<td>$\rho_2 = U_{rest} U_2 \rho_2 U_{rest}$</td>
<td>$U_2 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_4$</td>
<td>$\rho_3 = \bigotimes_{i=1}^{2} \rho_{3_1} \otimes \rho_{3_{A1}} \otimes \rho_{3_{B1}} \otimes \rho_{3_{C1}}$</td>
<td>$\rho_3 = U_{rest} U_3 \rho_3 U_{rest}$</td>
<td>$U_3 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td><strong>Majority Gate</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td>$\rho_4 = \bigotimes_{i=1}^{2} \rho_{4_1} \otimes \rho_{4_{A1}} \otimes \rho_{4_{B1}} \otimes \rho_{4_{C1}}$</td>
<td>$\rho_4 = U_{rest} U_4 \rho_4 U_{rest}$</td>
<td>$U_4 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_2$</td>
<td>$\rho_1 = \bigotimes_{i=1}^{2} \rho_{1_2} \otimes \rho_{1_{A1}} \otimes \rho_{1_{B1}} \otimes \rho_{1_{C1}}$</td>
<td>$\rho_1 = U_{rest} U_1 \rho_1 U_{rest}$</td>
<td>$U_1 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_3$</td>
<td>$\rho_2 = \bigotimes_{i=1}^{2} \rho_{2_1} \otimes \rho_{2_{A1}} \otimes \rho_{2_{B1}} \otimes \rho_{2_{C1}}$</td>
<td>$\rho_2 = U_{rest} U_2 \rho_2 U_{rest}$</td>
<td>$U_2 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_4$</td>
<td>$\rho_3 = \bigotimes_{i=1}^{2} \rho_{3_1} \otimes \rho_{3_{A1}} \otimes \rho_{3_{B1}} \otimes \rho_{3_{C1}}$</td>
<td>$\rho_3 = U_{rest} U_3 \rho_3 U_{rest}$</td>
<td>$U_3 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td><strong>Half Adder</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td>$\rho_4 = \bigotimes_{i=1}^{2} \rho_{4_1} \otimes \rho_{4_{A1}} \otimes \rho_{4_{B1}} \otimes \rho_{4_{C1}}$</td>
<td>$\rho_4 = U_{rest} U_4 \rho_4 U_{rest}$</td>
<td>$U_4 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_2$</td>
<td>$\rho_1 = \bigotimes_{i=1}^{2} \rho_{1_2} \otimes \rho_{1_{A1}} \otimes \rho_{1_{B1}} \otimes \rho_{1_{C1}}$</td>
<td>$\rho_1 = U_{rest} U_1 \rho_1 U_{rest}$</td>
<td>$U_1 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_3$</td>
<td>$\rho_2 = \bigotimes_{i=1}^{2} \rho_{2_1} \otimes \rho_{2_{A1}} \otimes \rho_{2_{B1}} \otimes \rho_{2_{C1}}$</td>
<td>$\rho_2 = U_{rest} U_2 \rho_2 U_{rest}$</td>
<td>$U_2 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_4$</td>
<td>$\rho_3 = \bigotimes_{i=1}^{2} \rho_{3_1} \otimes \rho_{3_{A1}} \otimes \rho_{3_{B1}} \otimes \rho_{3_{C1}}$</td>
<td>$\rho_3 = U_{rest} U_3 \rho_3 U_{rest}$</td>
<td>$U_3 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_5$</td>
<td>$\rho_5 = \bigotimes_{i=1}^{2} \rho_{5_1} \otimes \rho_{5_{A1}} \otimes \rho_{5_{B1}} \otimes \rho_{5_{C1}}$</td>
<td>$\rho_5 = U_{rest} U_5 \rho_5 U_{rest}$</td>
<td>$U_5 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_6$</td>
<td>$\rho_6 = \bigotimes_{i=1}^{2} \rho_{6_1} \otimes \rho_{6_{A1}} \otimes \rho_{6_{B1}} \otimes \rho_{6_{C1}}$</td>
<td>$\rho_6 = U_{rest} U_6 \rho_6 U_{rest}$</td>
<td>$U_6 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td><strong>ALU</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td>$\rho_4 = \bigotimes_{i=1}^{2} \rho_{4_1} \otimes \rho_{4_{A1}} \otimes \rho_{4_{B1}} \otimes \rho_{4_{C1}}$</td>
<td>$\rho_4 = U_{rest} U_4 \rho_4 U_{rest}$</td>
<td>$U_4 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_2$</td>
<td>$\rho_1 = \bigotimes_{i=1}^{2} \rho_{1_2} \otimes \rho_{1_{A1}} \otimes \rho_{1_{B1}} \otimes \rho_{1_{C1}}$</td>
<td>$\rho_1 = U_{rest} U_1 \rho_1 U_{rest}$</td>
<td>$U_1 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_3$</td>
<td>$\rho_2 = \bigotimes_{i=1}^{2} \rho_{2_1} \otimes \rho_{2_{A1}} \otimes \rho_{2_{B1}} \otimes \rho_{2_{C1}}$</td>
<td>$\rho_2 = U_{rest} U_2 \rho_2 U_{rest}$</td>
<td>$U_2 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_4$</td>
<td>$\rho_3 = \bigotimes_{i=1}^{2} \rho_{3_1} \otimes \rho_{3_{A1}} \otimes \rho_{3_{B1}} \otimes \rho_{3_{C1}}$</td>
<td>$\rho_3 = U_{rest} U_3 \rho_3 U_{rest}$</td>
<td>$U_3 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
<tr>
<td>$c_5$</td>
<td>$\rho_4 = \bigotimes_{i=1}^{2} \rho_{4_1} \otimes \rho_{4_{A1}} \otimes \rho_{4_{B1}} \otimes \rho_{4_{C1}}$</td>
<td>$\rho_4 = U_{rest} U_4 \rho_4 U_{rest}$</td>
<td>$U_4 = U_{A_k \eta_k} \otimes U_{R_k \eta_k}$</td>
</tr>
</tbody>
</table>

Table 4.16: The truth table of subtractor.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>B_{in}</th>
<th>B_{out}</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 4.17: Truth table of increment and decrement function.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increment</td>
<td></td>
</tr>
<tr>
<td>CB (control signal)</td>
<td>A</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
</tr>
<tr>
<td>Decrement</td>
<td></td>
</tr>
<tr>
<td>CB (control signal)</td>
<td>A</td>
</tr>
<tr>
<td>01</td>
<td>0</td>
</tr>
<tr>
<td>01</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4.18: Irreversibility induced energy results of different ASL devices.

<table>
<thead>
<tr>
<th>Device</th>
<th>Irreversibility Induced Energy</th>
<th>Energy efficiency (operations/J)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Inverter</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Latch</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>Majority gate</td>
<td>$1.14 \times 10^{-23} J$</td>
<td>$8.772 \times 10^{22}$</td>
</tr>
<tr>
<td>Half adder</td>
<td>$2.09 \times 10^{-23} J$</td>
<td>$6.211 \times 10^{22}$</td>
</tr>
<tr>
<td>Full adder</td>
<td>$3.87 \times 10^{-23} J$</td>
<td>$2.584 \times 10^{22}$</td>
</tr>
<tr>
<td>Subtractor</td>
<td>$3.87 \times 10^{-23} J$</td>
<td>$2.584 \times 10^{22}$</td>
</tr>
<tr>
<td>Multiplexor</td>
<td>$3.87 \times 10^{-23} J$</td>
<td>$2.584 \times 10^{22}$</td>
</tr>
<tr>
<td>Increment/Decrement</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>
from circuit simulations were shown in Table 4.5. Lower bounds on dissipated energy induced by irreversible information loss are shown in Table 4.18. Both refer to energy dissipation, but the results are quite different. They are from very different approaches that capture complementary aspects of energy dissipation.

The circuit simulation method illustrates the projected energy dissipation, which is numerically closest to what one would expect in actual ASL realizations. It is the combination of both static power and the dynamic power for the full LLG-model circuit simulation (integrated in time to obtain the final projected energy dissipation).

The physical information method, on the other hand, reveals the minimum energy dissipation, or the lowest bound for realizing the logic function via the strategy employed in the ASL circuit according to physical law. It is a reflection of the dynamic power regarding the irreversible information loss, i.e. it’s not about how much (static) power spent on maintaining the circuit running, but the power consumption retrieves when the input information is irreversibly erased from the circuit during a computational step. Thus the physical information method is closely related to the dynamic power. And by using the equation 3.34, we could get the corresponding irreversibility induced energy dissipation.

Table 4.5 and 4.18 shows the comparison of the two methods. We use the simplified ASL circuit model as the projected energy dissipation results.

The irreversibility induced energy for the buffer, inverter, and latch is 0. Because in this two circuits (the buffer and inverter share the same circuit, distinguished by the supply voltage), the computation processes of them are reversible. Inputs and outputs have a one-to-one relation, which means no information is erased from the circuits. In this case, no irreversibility induced energy is dissipated to the environment. The 3-input majority gate realizes NAND and NOR function, which means their computation processes are irreversible, and at least one output has multiple inputs related to it.
The projected energy for buffer and inverter are similar because they use the same circuit. The different logic function between a buffer and an inverter cause the slight differences between the final energy dissipation results. When supplied with a negative voltage, the ASL circuit acts as a buffer, while the positive voltage acts as an inverter. The projected energy of latch, majority gate and have adder get greater successively, which is reasonable because the circuits get more completed successively.

The irreversibility induced energy dissipation is the minimum cost a device would take for a logic computation. Compared to the projected energy, it is an intrinsic characteristic of circuit configuration and control strategy no matter what materials are used in realization of the devices and circuit interconnects. The result of the irreversibility induced energy is obtained from the physical information theory, a theory concerned about the computation steps and how the device fundamental working principle implements the computation steps.

Thus, the results of the irreversibility induced energy eliminate everything but the primary energy needed for realizing a logic computation. It is the reason why some of the results of irreversibility induced energy are low to the \(10^{-23}\) or even zero (buffer, inverter, and latch in our cases). They only show the lowest bounds of a computation step. It is the intrinsic characteristic determined by the logic computation step and the calculation method provided by the implementing device. For the detailed calculation step, please refer to Chapter 3.

The definition of projected energy, on the other hand, is obtained from simulations based on models of actual circuit realization. It depends on constituent material properties, supply voltage, device dimensions, and computing temperature; the target circuit is transformed to an equivalent circuit model that reflects these dependencies and simulated in the virtual environment. The simulation is a digital circuit simulation; all possible inputs are considered and are set manually by the user, and the results would be similar to what we have on a real chip test.
Thus, the results of the projected energy consist of most of the aspects in real circuit simulation, including the irreversibility induced energy. The huge gap between the projected and irreversibility induced energy is as large as $10^{10}$, which shows the difference between the projected energy and the irreversibility induced energy dissipation. Given their complementary nature and the very different aspects of energy dissipation that they capture, one shouldn’t be surprised by the different results and the huge gap.
CHAPTER 5
SUMMARY AND CONCLUSIONS

The goal of this thesis was to understand energy efficiency of computation in a new post-CMOS device, the all-spin logic (ASL) device. By comparing projected energy efficiencies from ASL circuit simulations and irreversibility induced energy dissipation of different ASL circuits, we aimed to provide estimates of the computational energy efficiencies of a variety of ASL circuits and to gauge the gap between the projected and irreversibility induced energy dissipation in these circuits.

The thesis was introduced in Chapter 1. It first pointed out that the limits of CMOS devices make energy dissipation one of the problems that we concerned. Two types of energy dissipation were defined and introduced. The objectives of the thesis were introduced. The working principle and basic computing logic circuits of ASL device were introduced in Chapter 2. The current states, including charge current and spin current, were identified and discussed, illustrating why the ASL device has low supply voltage requirement and how it becomes one of the most promising post-CMOS devices. Chapter 3 discussed in detail about the two energy calculation methods, analog circuits simulation with MATLAB and the physical information theory. The analog circuits simulation got the results of projected energy dissipation, while the physical information theory got the results of the theoretical information energy dissipation. In Chapter 4, the results of these two calculation methods were compared and discussed. Simple circuits of ASL devices such as the buffer, inverter, latch, and half adder were tested. A larger circuit—an ASL arithmetic logic unit (ALU) —
was finally studied. Simulation results and fundamental limits were compared for the various ASL circuits.

The use of complementary theoretical approaches to analyze the same set of ASL circuits provides insight into what might be expected for particular ASL circuit realizations and what is and is not physically achievable with optimization. The low energy dissipation resulting from the low required supply voltage in ASL was observed in the circuit simulation results, supporting the widely recognized advantages in energy efficiency of ASL when compared to conventional CMOS. The irreversibility induced energy, which fixes the lowest bound on the dissipation incurred in a computation process implemented by a particular strategy, is orders of magnitude lower than energy dissipation projected from simulations. The theoretical lower bounds are approachable to differing degrees in various technologies. This raises the question of whether the gap between projected efficiencies from simulations and theoretical lower bounds can be closed in ASL.

Most of the energy cost in ASL goes toward operating circuit, e.g. transporting particles and generating the force of spin torque, as is clear from the calculations of this work. While the associated energy costs are far lower than in CMOS, and can be optimized through improvements in materials and structure design, the energy efficiencies of ASL circuits cannot be expected to approach fundamental efficiency limits associated with the computations they perform. Energetic overheads from static power alone, induced by the currents required to maintain ASL circuit operation, exceed fundamental limits by orders of magnitude as suggested by the projections and limits calculated in this work.
APPENDIX A

LANDAU-LIFSHITZ-GILBERT EQUATION DERIVATION

To get the Landau-Lifshitz-Gilbert equation, we can follow the following steps [25][26].

In a spatially uniform magnetic field, we focus on a small region $dV_r$, which contains $N$ elementary magnetic moments $\vec{\mu}_j$ ($\vec{\mu}_j$ also can be put as $|\mu_j>, j = 1, 2, 3, ... N$), as shown in Fig. A.1. We could get magnetization vector $\vec{M}$ by summing all $\vec{\mu}_j$ together and dividing by $dV_r$,

$$\vec{M}(\vec{r}) = \frac{\sum_j^N \vec{\mu}_j}{dV_r} \quad (A.1)$$

where $\vec{M}$ is magnetization vector and $c$ is the elementary magnetic moment in a small region $dV_r$.

Note that spin moment $\vec{\mu}_j$ and angular momentum $\vec{L}$ can be related by equation

$$\vec{\mu}_j = -\gamma \vec{L} \quad (A.2)$$

where

$$\gamma = \frac{g|e|}{2m_e c} \quad (A.3)$$

$g \approx 2$ is the Lande splitting factor. $e$ and $m_e$ are the electron charge and mass, respectively. $c$ is the speed of light.
Figure A.1: A small region $dV_r$ in magnetic field $\vec{H}$. Elementary moments $|\mu_j>$ point to random directions.

Figure A.2: Procession and damping [27]. The dotted circle indicates the procession without damping while the solid curve shows the real damping path. $T_D$ is the damping torque (shown in equation A.9).
According to momentum theorem, we have

\[ \frac{d\vec{L}}{dt} = \vec{\mu}_j \times \vec{H} \]  \hspace{1cm} (A.4)

where $\vec{L}$ is the angular momentum, $t$ is time, $\vec{\mu}_j$ is the elementary magnetic moment, and $\vec{H}$ is the magnetic field. Substitute equation A.2 into equation A.4, we get

\[ \frac{d\vec{\mu}_j}{dt} = -\gamma \vec{\mu}_j \times \vec{H} \]  \hspace{1cm} (A.5)

Next we sum $|\mu_j|$ from 1 to N and divide both sides by $dV_r$

\[ \frac{d\sum^N_j \vec{\mu}_j}{dtdV_r} = -\gamma \frac{d\sum^N_j \vec{\mu}_j}{dV_r} \times \vec{H} \]  \hspace{1cm} (A.6)

Substitute equation A.1 into A.6, we get continuum precession equation:

\[ \frac{\partial \vec{M}}{\partial t} = -\gamma \vec{M} \times \vec{H} \]  \hspace{1cm} (A.7)

Replacing the spatially uniform magnetic field by effective magnetic field , we get

\[ \frac{\partial \vec{M}}{\partial t} = -\gamma \vec{M} \times \vec{H}_{eff} \]  \hspace{1cm} (A.8)

Equation A.8 describes the undamped gyromagnetic procession shown in dot line in Fig. A.2 [27]. The magnetization vector $\vec{M}$ goes in a circle because damping is not taken into consideration. However, in real life, gyromagnetic procession is usually affected by a torque $\vec{T}_D$. Thus the magnetization vector $\vec{M}$ goes in a spiral line (solid line in Fig. A.2) instead of a circle. The spiral line is known as the damped procession.

This damping torque can be described as

\[ \vec{T}_D = -\lambda \vec{M} \times (\vec{M} \times \vec{H}_{eff}) \]  \hspace{1cm} (A.9)

where $\lambda$ is a constant depending on the material.
By adding the additional torque $\vec{T}_D$ to the original equation A.8, i.e. by adding a damping torque to the undamped procession, we get the equation describing the damped procession, which is also known as the Landau-Lifshitz equation:

$$\frac{\partial \vec{M}}{\partial t} = -\gamma \vec{M} \times \vec{H}_{eff} - \lambda \vec{M} \times (\vec{M} \times \vec{H}_{eff}) \quad (A.10)$$

The first term of this equation describes an undamped procession generated by the interaction of magnetization vector $\vec{M}$ and the effective magnetic $\vec{H}_{eff}$. The second term describes the damped procession.

The damping torque has another form as shown in equation A.11.

$$\vec{T}_D = \alpha \vec{M} \times \frac{\partial \vec{M}}{\partial t} \quad (A.11)$$

where $\alpha > 0$ is the Gilbert constant, depending on the material. This torque is generated by the magnetic field

$$\vec{H} = -\gamma \alpha \frac{\partial \vec{M}}{\partial t}$$

If this new torque $\vec{T}_D$ is added to the undamped procession equation A.8 instead of the torque described in equation A.9, we get a new form to describe the damped procession:

$$\frac{\partial \vec{M}}{\partial t} = -\gamma \vec{M} \times \vec{H}_{eff} + \alpha \vec{M} \times \frac{\partial \vec{M}}{\partial t} \quad (A.12)$$

Equation A.12 is also known as Landau-Lifshitz-Gilbert equation. The first term of this equation describes an undamped procession generated by the interaction of magnetization vector $\vec{M}$ and the effective magnetic $\vec{H}_{eff}$. The second term describes the damping torque that turns the undamped procession into the damped procession.
APPENDIX B

SPIN INJECTION WITH LLG EQUATION

For ASL device, we need to add a magnetization torque to the original LLG equation due to the absorption of incoming spin current [28][18].

\[
\frac{\partial \vec{m}}{\partial t} = -\gamma \mu_0 \vec{m} \times \vec{H}_{\text{eff}} + \alpha \vec{m} \times \frac{\partial \vec{m}}{\partial t} + \frac{\vec{I}_\perp}{eN_s} \tag{B.1}
\]

where \(\mu_0\) is the free space permeability. \(\vec{I}_\perp\) is the interface spin current that is perpendicular to \(\vec{m}\), as shown in Fig. B.1[18]. \(\vec{m}\) is the magnetic moment unit vector of the ferromagnet. \(\vec{I}_\perp\) is the perpendicular part of the current which has the tend to go through the interface of magnet (FM) and channel (NF), i.e. \(\vec{I}_\perp\) is the actual current that goes through the interface of FM and NF, while \(\vec{I}_\parallel\) is the current reflected by the interface. \(N_s\) is the total number of Bohr magnetons per magnet. \(\vec{H}_{\text{eff}}\) is the effective magnetic field applied to ferromagnet and \(e\) is the charge of an electron. \(\alpha > 0\) is the material dependent Gilbert constant.

Thus, the third term in the right side is the spin torque \((\vec{I}_\perp)\) in NF channel. \(\vec{I}_\perp\) can be rewrite as

\[
\vec{I}_\perp = \vec{I}_s - \vec{m}(\vec{m} \cdot \vec{I}_s) = \vec{m} \times (\vec{I}_s \times \vec{m}) \tag{B.2}
\]

where \(\vec{I}_s\) is the spin current.

We can infer a coupled spin transport-magnetization dynamics model [28], as shown in Fig. B.2[29]. The self-consistency can be described as below. The angular momentum of the magnet \(\vec{m}\) determines the conductance of the magnet. The conductance of the magnet determines how many spins would enter the magnet. The spin
Figure B.1: Spin current injection [18]. A nonmagnetic material (red block) is in contact with ferromagnetic magnetic material (green block). For a normal magnetization $\vec{m}$, an incoming spin up current can be written with a sum of right and left spin states $|\to\rangle$ and $|\leftarrow\rangle$ with a scalar $1/\sqrt{2}$. Assuming the conductance $G$ only allows majority spin (right spin) to pass. Only the perpendicular spin current $I_\perp$ (right spin state) is allowed to go through the interface between the nonmagnetic and ferromagnetic material, while $I_\parallel$ (left spin state) is reflected. The magnetization torque generated by $\vec{I}_\perp$ changes the original magnetization $\vec{m}$ of the FM.

Figure B.2: Self-consistency of LLG nanomagnet dynamics and spin transport [29]. It starts with the first pair of coupled spin current ($\vec{I}_{s1}$) and magnetic moment unit vector ($\vec{m}$), and keeps circulating until both of them satell down.
Figure B.3: ASL interver with (a) top view and (b) side view [29].

Figure B.4: Circuit simulation model of an ASL inverter [29]. $G_{FM1}(\vec{m}_1)$ and $G_{FM1}(\vec{m}_2)$ represent the conductance of ferromagnet1 (FM1) and ferromagnet2 (FM2), respectively. $G_{sf\pi}$ and $G_{se\pi}$ represent the conductance of the ground lead and the conducting channel respectively.
current enters the magnet influences the angular momentum of the magnet, which in turn changes the conductance of the magnet. Thus, the spin current entering the magnet would change, too. These kinds of changes will keep going on until every value becomes stable. This self-consistency happens at the receiver side as well as the sender side.

Fig. B.4 [29] shows the circuit model of an ASL inverter which has top view and side view in Fig. B.3. Apply equation B.1 to this special circuit; we can easily get LLG equations for this inverter [29]:

\[
\frac{\partial \vec{m}_1}{\partial t} = -\gamma \mu_0 \vec{m}_1 \times \vec{H}_{eff} + \alpha \vec{m}_1 \times \frac{\partial \vec{m}_1}{\partial t} + \frac{\vec{I}_{13\perp}}{eN_s} \quad (B.3)
\]

\[
\frac{\partial \vec{m}_2}{\partial t} = -\gamma \mu_0 \vec{m}_2 \times \vec{H}_{eff} + \alpha \vec{m}_2 \times \frac{\partial \vec{m}_1}{\partial t} + \frac{\vec{I}_{23\perp}}{eN_s} \quad (B.4)
\]

where \( \vec{I}_{13\perp} \) is the perpendicular current from node 1 to node 3, i.e. the current carries the magnetic information from magnet 1. \( \vec{I}_{23\perp} \) is the current from node 2 to node 3, i.e. the current generates the torque to change the magnetization of the receiving magnet.
BIBLIOGRAPHY


