Design and Simulation of Reversible Molecular Mechanical Logic Gates and Circuits

Ian Seet

The Queen's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2022

Acknowledgements

I would like to thank my supervisor Jonathan Doye and my collaborator Tom Ouldridge for their invaluable assistance in ensuring that my writing remains within the realm of human comprehensibility.

I would also like to thank the various members of the Doye group, particularly the members who actually set foot in the office (aka Doye's Boyes) - Mike, Navoneel, Akie, Hannah, Colin, Don, Monty, Chuyan; the members of Ard Louis' group including Ard himself, Wilber, Joakim and others, as well as the other denizens of the PTCL and adjacent buildings over the past few years, including but not limited to Izzy, Annina, Filippo, Alex, Daniel, Andrew, Matt, Jenny, Ollie, Dilhan, Tim, Tristan, Chloe, Matthew, Tomasz, Matina, Henry, Shaoqi, Jim, Aidan, Zak, Kieran, Astrid, Ava, Gayatri and many others who have tolerated my various eccentricities (such as my tendency to randomly perform calisthenics or break into stirring renditions of "All Star" on my guitar in the PTCL tea room) over the past few years.

I would also like to extend special thanks to the scout of the college accommodation I resided in during the pandemic, Andrea, for taking extra effort to keep us safe and being my only reliable in-person conversation partner during a very trying period. Finally, I would like to thank my parents for believing in me when few others would (which in fairness is entirely understandable).

Abstract

In this thesis, I outline a framework for constructing molecular mechanical circuits from logic gates and components that I have designed and optimised, and describe the results of simulations conducted to determine the thermodynamic properties of these circuits with a coarse-grained rigid body simulator. The molecular mechanical logic gates I have designed can be used to construct logically reversible circuits, which have the potential to be far more efficient than their irreversible counterparts due to the fact that they can easily be operated in a thermodynamically reversible manner.

Using the rigid body simulator, I first demonstrate Landauer's principle, a fundamental principle of information theory which relates the change in information entropy of a given process to the work required to complete it, thus showing that my rigid body simulator is thermodynamically self-consistent. I then simulate a novel molecular mechanical NAND gate design and use it to construct complex logically and thermodynamically reversible combinatorial circuits, including a half-adder. Subsequently, I investigate the use of pipelining to increase the throughput of a circuit at the cost of potential logical irreversibility, and show that it is possible to perform pipelining efficiently under certain conditions. Finally, I use a genetic algorithm to optimise various circuit components, demonstrating a novel technique to reduce the number of costly simulations that must be run in order to calculate the fitness function.

Contents

Chapter 1

Introduction

1.1 The energy cost of electronic computing

Power consumed by electronic computing devices is, as of the time of writing, a major contributor to global CO_2 emissions, with data centres and high performance computing contributing 100 Mt of CO_2 per year and cryptocurrency mining another 60 Mt [1]. As this power consumption is only likely to rise in both absolute and relative terms, methods of reducing the per-operation energy cost of computational operations will become increasingly important in mitigating the impact of climate change.

Virtually all modern computing devices make use of CMOS (Complementary Metal–Oxide Semiconductor)-based transistor technology in order to perform logical operations. Metal-oxide-semiconductor field-effect (MOSFET) transistors, of which CMOS transistors are a subset, rely on the application of a voltage across a doped semiconductor which results in the depletion of charge carriers in the semiconductor, thus triggering a change in the conductivity across the transistor. The process of depleting of charge carriers in a semiconductor in response to an applied voltage is directly analogous to the process of charging and discharging a capacitor, and there-

fore requires the application of work to move the charges against a force exerted by an electric field.

The work done in this process, referred to as the switching energy of a transistor, is directly proportional to the size of the transistor and proportional to the square of voltage. It is not affected by the clock speed (frequency) of the processor in the event that voltage is held constant. However, the rate at which a capacitor can be charged or discharged is itself proportional to voltage; therefore, in the limit of high clock speeds, the work required per clock cycle increases quadratically with clock speed in an ideal transistor [2]. In practice, the heat dissipated will raise the temperature of the device, thereby increasing electrical resistance, resulting in a super-quadratic increase in energy dissipated per clock cycle. It has also been estimated that the switching energy of a standard CMOS transistor has a lower bound of $100\,k_{\mathrm{B}}T$ [3] where k_B is the Boltzmann constant and T is the ambient temperature. This limit holds true regardless of clock speed, and will be reached by 2030 at the current rate of circuit miniaturisation.

1.2 Thermodynamic limits on computation

1.2.1 Boolean logic

Boolean logic is fundamental to all digital computing. In Boolean algebra, variables, also known as bits, may only take on values of true (1) and false (0). The three fundamental Boolean operators are the negation, conjunction and disjunction operations, which correspond to the NOT, AND and OR logic gates respectively.

A Boolean logic gate is an operator which receives at least one Boolean variable as an input and returns at least one Boolean variable as an output. A Boolean logic gate can be represented as a truth table, which outlines all possible inputs for a given logic gate and indicates the output(s) returned for a given input. Tables 1.1a, 1.2a

Input	Output
0	1
1	0

(a) NOT

Input	Output
0	0
1	0

(b) Erasure

Table 1.1: Boolean truth tables of the NOT gate and bit erasure operations.

and 1.2b are the truth tables for the NOT, AND and OR gates respectively.

From these three fundamental logic gates, all other logical operations may be constructed. Furthermore, there are two universal logic gates, the NAND and NOR gates (Table 1.3), from which all Boolean logic gates may be constructed. The NAND gate is the AND gate with negated outputs; analogously, the NOR gate is the OR gate with negated outputs. Each of the three fundamental logical gates can be constructed from only NAND and NOR gates. For instance, the NOT gate can be constructed from the NAND gate by feeding the same bit into both input channels. Since the AND gate is equivalent to the NAND gate with inverted outputs and the OR gate is equivalent to the NAND gate with inverted inputs, all three fundamental logic gates can be generated from the NAND gate, and hence any logical operation may be constructed from an arbitrary number of NAND gates.

More sophisticated logic gates and circuits may be constructed from feeding the output of a basic logic gate into the input of another such gate. These constructs are known as combinatorial circuits. An example of a combinatorial circuit constructed

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

(a) AND

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

(b) OR

Table 1.2: Boolean truth tables of the AND and OR logic gates.

Input 1	Input 2	Output
0	0	1
0	1	1
1	0	1
1	1	0

(a) NAND

Input 1	Input 2	Output
0	0	1
0	1	0
1	0	0
1	1	0

(b) NOR

Table 1.3: Boolean truth tables of the NAND and NOR logic gates.

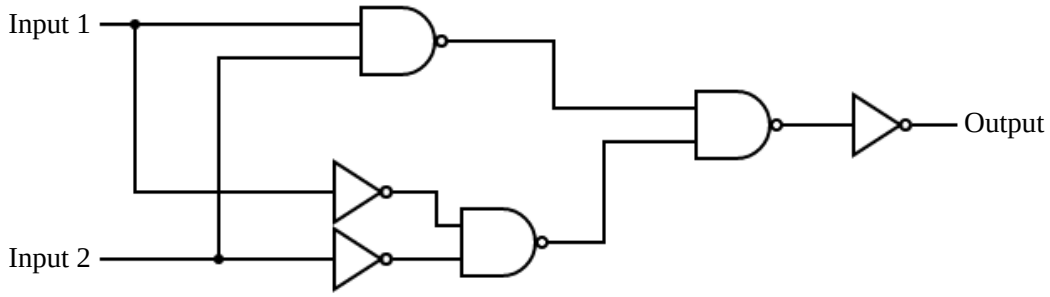


Figure 1.1: The circuit diagram of an XOR gate constructed from only NAND gates and inverters.

from basic logic gates which executes a useful operation is the XOR gate (Table 1.4a), which returns '1' if and only if one of its input bits is equal to '1', else it returns '0'. The XOR gate is a key component of addition operations, since it effectively returns the result of the least significant digit of the addition of two Boolean variables. Combinatorial logic gates can be graphically represented in the form of a circuit diagram, such as in Fig. 1.1, which illustrates the construction of an XOR gate from three NAND gates and inverters. (See Appendix A for a legend of commonly used symbols in circuit diagrams.)

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

(a) XOR

Input 1	Input 2	Output 1	Output 2
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

(b) CNOT

Table 1.4: Boolean truth tables of the XOR and CNOT logic gates.

1.2.2 Logical reversibility

Even if improvements in transistor technology would allow the previously mentioned $100\,k_{\mathrm{B}}T$ limit on the switching energy of a CMOS-based transistor to be bypassed, the laws of thermodynamics nonetheless impose a fundamental limit on the minimum amount of work required to execute a logically irreversible operation. A logically reversible operation is a type of Boolean operation in which there is a bijection (one-to-one mapping) between all possible input and output states. A simple example of a logically reversible operation is the NOT gate or inversion operation (Table 1.1a), where it is always possible to reconstruct the input bit from the output bit. In contrast, bit erasure (Table 1.1b) is a logically irreversible operation, as it is impossible to recover the input from the output bit due to the fact that both possible inputs map to the same output.

It is also possible for logic gates to be conditionally reversible; that is, the input can be reconstructed only for some subset of all possible outputs. For instance, the AND gate (Table 1.2a) is only logically reversible if its output is '1'. In the event that its output is '0', three possibilities exist for the input state ('00, '01' and '10'), making the AND gate logically irreversible for this particular output.

A logically irreversible operation can be rendered logically reversible by ensuring that sufficient information from the input bits is preserved in the output such that the input can always be reconstructed. For instance, the XOR gate (Table 1.4a) is clearly

Input 1	Input 2	Output 1	Output 2	Output 3
0	0	0	0	0
0	1	0	1	0
1	0	1	0	0
1	1	1	1	1

Table 1.5: Boolean truth table of the input-preserving AND gate.

not logically reversible as it is impossible to achieve a bijective mapping between two input bits and one output bit. It can, however, be converted into the reversible CNOT gate (Table 1.4b) by returning the state of the first input bit unchanged as the first output bit. The CNOT gate can be described as a conditional inversion operation, as it inverts the value of the second input bit if the first input bit is '1', or else it leaves it unchanged.

In the case of the AND gate, it is only possible to achieve logical reversibility by preserving both input bits, with the inclusion of two additional output bits (Table 1.5) which return values identical to the input bits. Another method of converting the AND gate into a reversible form is to take advantage of the natural reversibility of the conditional inversion operation, as in the case of the Toffoli gate (Table 1.6), where if the first two input bits are equal to '1', the third input bit is inverted.

Logically irreversible systems require a minimum of $k_{\mathrm{B}}T \ln 2$ of work (or equivalent

Input 1	Input 2	Input 3	Output 1	Output 2	Output 3
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Table 1.6: Boolean truth table of the Toffoli gate.

resources) spent for every bit erased [4, 5, 6]. This limit, also known as Landauer's principle, is a direct consequence of the second law of thermodynamics as applied to information entropy, and has been demonstrated experimentally [7, 8]. Information entropy is a measure of the uncertainty contained in a random variable, and is directly analogous to thermodynamic entropy in that a system with high information entropy, like a system with high thermodynamic entropy, has a probability distribution spread over a large number of possible states. Although the information entropy of the data stored by a particular system is not generally equal to the thermodynamic entropy of the entire system, when the changes in entropy corresponding to the degrees of freedom of the system which are not used to represent data are negligible, the change in thermodynamic entropy of the system can be approximated by the change in information entropy of the data stored by the system. The information entropy $H(X)$ for a discrete random variable X with N possible states x_i is defined in Eq. 1.1 [9]:

$$H(X) = \sum_{i=1}^{N} -p(x_i) \ln p(x_i) \tag{1.1}$$

where $p(x_i)$ is the probability that the discrete variable is occupying the state x_i. Hence, the information entropy of a single bit can be expressed with the following equation:

$$H(X) = -p(0) \ln p(0) - p(1) \ln p(1) \tag{1.2}$$

where $p(0)$ is the probability that the bit is in the state '0' and $p(1)$ is the probability that the bit is in the state '1'. Therefore, the information entropy of a single fully randomised bit where $p(0) = p(1) = 0.5$ is $-\ln 0.5 = \ln 2$. On the other hand, the entropy of an erased bit where $p(0) = 1$ and $p(1) = 0$ is $-\ln 1 = 0$. Thus, the erasure of one bit of information results in the information entropy of the computing system decreasing by $\ln 2$, a decrease that must be compensated for by an equal or greater

increase in entropy in the surroundings per the second law of thermodynamics. The minimum amount of work E which must be expended in order to execute a logically irreversible operation can be expressed in the following form:

$$E = k_B T[H(X_i) - H(X_f)] \tag{1.3}$$

where $H(X_i)$ is the information entropy of the initial state and $H(X_f)$ is the information entropy of the final state after the operation. It should be noted that executing a logically irreversible process does not necessarily increase the thermodynamic entropy of the universe, and the work expended is recoverable as long as there are no sources of additional entropy production (Section 1.2.3); however, it is not always practical to recover this work (Section 1.2.4).

Logically reversible gates are currently of great interest owing to their relevance to quantum computing. A fundamental axiom of quantum mechanics is that the time evolution of a quantum state must be represented by a unitary operator, with the exception of measurement operations which irreversibly destroy any information about phase or entanglement and are therefore not considered quantum logic gates. The unitarity of quantum mechanics ensures that quantum logic gates must always be logically reversible [10].

Although quantum logic gates are logically reversible in theory, quantum phase is highly fragile and prone to decoherence; thus, quantum logic gates generally have a much higher error rate than their classical counterparts. While phase errors can be mitigated by an appropriate quantum error correcting code [11], using a quantum error correcting code requires the execution of measurement operations on several additional qubits, and is therefore logically irreversible. Thus, quantum logical operations are often logically irreversible in practice.

1.2.3 Thermodynamic reversibility

The concept of thermodynamic reversibility is distinct from that of logical reversibility. A process is thermodynamically reversible if the sum of the work done during both forward and reverse processes converges to zero in the quasistatic limit, and thus does not increase the total entropy of the universe. A quasistatic process is defined as one that progresses at an infinitesimally slow rate [12]. A quasistatic process need not be thermodynamically reversible and may be subject to dissipation [12, 13]; for instance, in systems with velocity-independent sources of dissipation such as friction [14]. However, thermodynamically reversible processes must necessarily be quasistatic. A logically reversible process need not be thermodynamically reversible; for instance, the NOT gate implemented using traditional CMOS-based transistors is logically but not thermodynamically reversible due to the fact that the switching energy is not recovered.

Conversely, it is possible for a logically irreversible process to be thermodynamically reversible. For instance, although bit erasure is logically irreversible, it is possible to recover the non-equilibrium free energy contained in the low-entropy erased state. An example of this process is the thought experiment proposed by Szilard to convert information into work, the eponymous Szilard engine [15]. Fig. 1.2 illustrates the operation of the engine. In (a), a box containing a single particle of gas is prepared. If it is known on which side of the box the particle is at a given moment, work can be extracted from the set-up by inserting a frictionless partition in the centre of the box as illustrated in (b). A weight can be attached to the partition on the same side containing the gas particle as demonstrated in (c), and the chamber containing the gas particle allowed to expand isothermally until it reaches the state depicted in (d). As this expansion lifts the weight against gravity, it allows work to be extracted from the system. The entire process does randomise the position of the particle and therefore cannot be repeated; thus, a perpetual motion machine is not possible. In

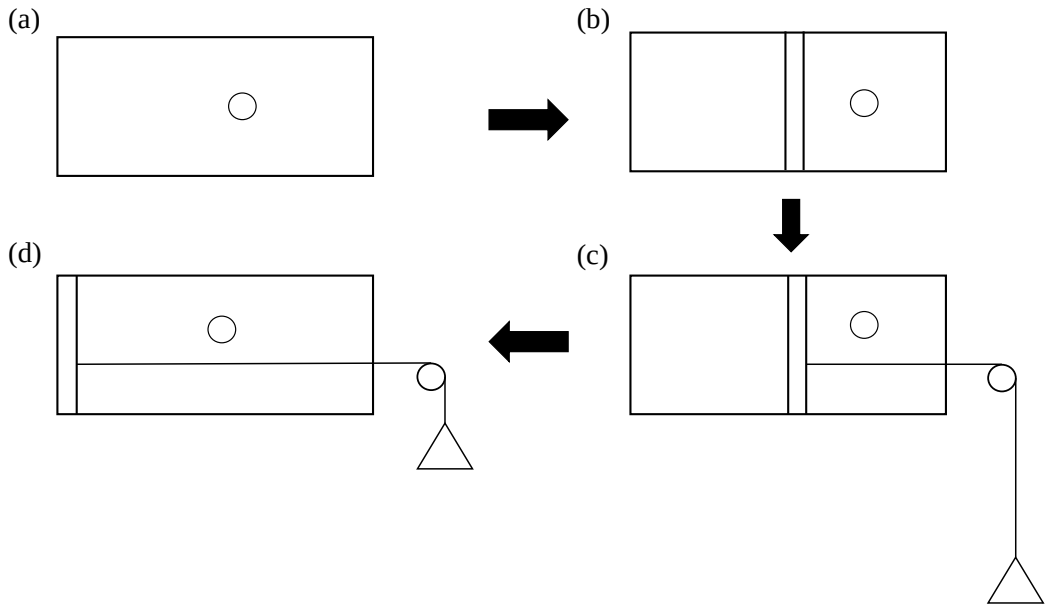


Figure 1.2: Operation of the Szilard engine demonstrating the conversion of information to work. In (a), a system is set up consisting of a gas particle trapped in a box. In (b), a frictionless partition is inserted into the box. If it is known on which side of the box the particle is trapped in a weight can be attached to the side of the partition containing the particle (c), and useful work can be extracted from the system as the chamber containing the gas particle expands isothermally, lifting the weight (d).

this specific example, the information which allows for the extraction of usable work is the correlation between the position of the particle and the position of the weight, which is lost when the particle is randomised. Brittain *et al.* have demonstrated a means of physically realising a biochemical Szilard engine [16].

By making use of the ability of the Szilard engine to extract work from a bit of known state, bit erasure can be executed in a thermodynamically reversible manner as it is theoretically possible to apply the Szilard engine to the erased bit and completely recover the $k_{\mathrm{B}}T\ln 2$ of work originally expended to overwrite the bit. It is not necessary to actually apply the Szilard engine protocol to the bit immediately after it is overwritten; so long as the reverse process could in theory fully recover the work invested in the forward process, the forward process can be considered thermodynamically reversible despite being logically irreversible since there is no increase in

the total entropy of the system and its surroundings.

1.2.4 Thermodynamic costs of logical irreversibility

Although the laws of thermodynamics do not prevent a logically irreversible operation from being executed in a thermodynamically reversible manner, it is generally only possible to do so for a limited subset of optimal distributions of starting states [17, 18, 19]. Other distributions of starting states will incur an entropic penalty known as the mismatch cost, defined as the additional entropy produced during a given process when a non-optimal distribution of starting states is used instead of an optimal distribution.

The mismatch cost for a given system can be calculated via the following expression [19]:

$$\sigma(p) = D(p \,||\, q) - D(Pp \,||\, Pq) \tag{1.4}$$

where $\sigma(p)$ is the mismatch cost for a initial distribution p subject to the process P, $D(p \,||\, q)$ is the Kullback-Leibler divergence between the initial distribution p of states and the optimal distribution q, and $D(Pp \,||\, Pq)$ is the Kullback-Leibler divergence between the final distribution Pp that is the result of the initial distribution p being subject to process P, and the final distribution Pq that is the result of the optimal distribution q being subject to process P.

The Kullback-Leibler divergence is a measure of the statistical difference between two probability distributions [20]. For a pair of discrete probability distributions p and q over the domain X, the Kullback-Leibler divergence between the two distributions $D(p \,||\, q)$ is defined as follows:

$$D(p \,||\, q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} \tag{1.5}$$

The mismatch cost is therefore equal to the drop in Kullback-Leibler divergence between p and q as a result of process P.

The mismatch cost can be intuitively understood with a simple example consisting of a system which can exist in only three states, A, A' and B. A and A' can both be converted to B and vice versa, but cannot interconvert between each other. Suppose that states A and A' are of equal free energy and are almost entirely favoured by the initial set of conditions, and there exists an isothermal process P which causes the equilibrium to shift almost entirely in favour of state B. When process P is executed in reverse on state B, it yields an equal mixture of states A and A'. Thus, process P operating on an equal mixture of states A and A' would be thermodynamically reversible in the quasistatic limit, and would generate no additional entropy.

However, if the initial distribution consists purely of state A, process P executed on this state will be irreversible, as it would yield a mixture of A and A' once reversed. Thus, when process P is applied to the pure state A, it will result in an increase in universal entropy, and the initial distribution consisting of state A yields a mismatch cost relative to an initial distribution consisting of an equal mixture of states A and A'.

In computer engineering terms, the mismatch cost ensures that it is usually much more difficult to create a practical computing system where logically irreversible operations can be rendered thermodynamically reversible since optimality is dependent on the initial distributions of starting states. Thus, if the exact distribution of input states is beyond the control of the computer engineer, which will generally be the case for a practical computing system, it will not always be possible for the logically irreversible operation to be executed in a thermodynamically reversible manner.

1.3 Physical implementations of reversible logic

1.3.1 Electronic logic gates

It is possible to modify traditional thermodynamically irreversible CMOS-based circuits to scavenge a portion of the switching energy. This technique, known as adiabatic logic, attempts to recover the charge contained in the depletion zone of the doped semiconductor after the electric field is turned off, and hence the potential energy associated with said charge. Although effective at substantially reducing power consumption at low clock speeds [21], adiabatic logic circuits are significantly larger and more complex than non-adiabatic circuits and still suffer from the same quadratic increase in energy dissipated per clock cycle with voltage even under ideal circumstances [22], negating the advantages of using such gates at higher clock speeds.

Another proposal for the physical realisation of reversible electronic logic gates revolves around the use of superconducting Josephson junctions [23, 24]. The Josephson effect is a phenomenon by which an electric current may flow without resistance between two superconductors which are placed in close proximity to each other via quantum tunneling. Such currents are extremely sensitive to magnetic fields, including the magnetic fields generated by another similar current passing through an inductor, and can be therefore be used to construct reversible logic gates. Although capable of very high efficiencies, superconducting logic gates suffer from two weaknesses: firstly, superconductors require cryogenic temperatures to function; secondly, the use of inductors in superconducting logic gates substantially increases the difficulty of miniaturising such systems, with state-of-the-art superconducting logic circuitry having a much lower density than CMOS-based circuits [25].

1.3.2 Molecular logic gates

Logically reversible circuits can also be constructed from traditional molecular logic systems, such as DNA logic gates [26, 27] (See Fig. 1.3 [1] for an example of such a gate). DNA logic gates operate by exploiting the specificity of base-pairing interactions between fragments of DNA. For a DNA logic gate to execute a logical operation, the input, generally consisting of a single-stranded fragment of DNA, must first diffuse toward and bind to a toehold on the gate, which is typically composed of several DNA fragments held together by base-pairing interactions. In order to produce the output, the input strand must typically displace the output strand from the gate, a process which requires the disruption of the hydrogen bonding interactions between the base pairs of the output strand and the gate. This reliance on diffusion to transmit information and bond-breaking processes to execute logical operations results in DNA logic gates being far slower than electronic logic gates, with typical execution times measured in minutes [28].

Beyond the inherent sluggishness of DNA logic gates, it is not trivial to design a thermodynamically reversible DNA computing system. In an electronic computer, taking the system to the quasistatic limit is in principle relatively easily accomplished by reducing the clock speed to an arbitrarily small value[2]. It is, however, far more difficult to reduce the rates of diffusion and bond-breaking in a DNA logic gate or other similar system to an arbitrarily small value in a similar manner. Brittain *et al.* have proposed a method to accomplish this via a series of chemical buffers [29]; however, the proposed protocol is conceptually and physically far more complex than the simple reduction in clock speed of an electronic logic gate.

[1] Reprinted with permission from A. Genot, J. Bath, and A. Tuberfield. Reversible logic circuits made of DNA. J. Am. Chem. Soc., 133:20080–20083, 2011. Copyright 2011 American Chemical Society.

[2] As previously mentioned, a quasistatic process need not be thermodynamically reversible. Thus, a conventional electronic gate where the switching energy is not recovered can still be brought to the quasistatic limit.

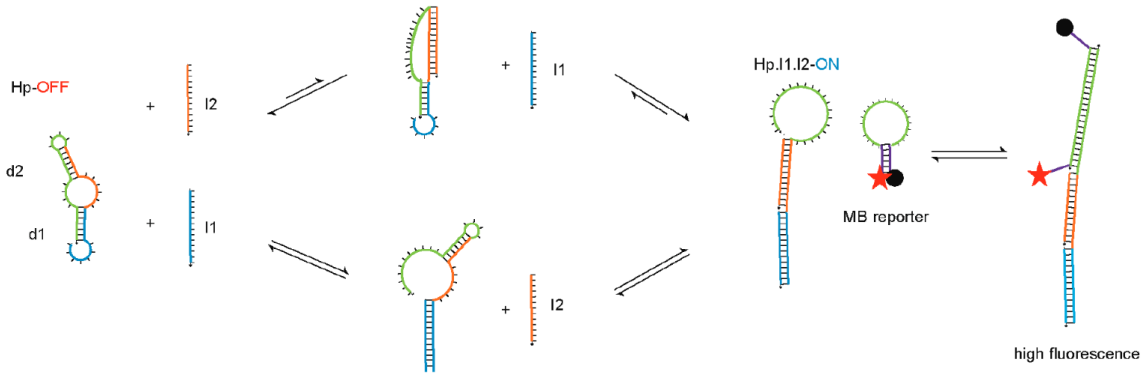


Figure 1.3: The logically reversible AND gate implemented by Genot *et al.* [26]. In this DNA logic gate, the two input strands I1 and I2 must both diffuse towards and bind to the target complementary domain d1 and d2 (coloured orange and blue respectively) via a single-stranded toehold in order to release the green domain. This in turn allows a strand containing a fluorescent reporter (labelled MB reporter) to diffuse toward and bind to it, enabling the output to be reported via fluorescence. This gate is logically reversible since the input bits are not destroyed during the operation of the gate. However, the authors do not describe a protocol for implementing this gate in a thermodynamically reversible manner.

DNA logic gates are not the only class of molecular logic gates. Other biomolecules such as proteins can be used to construct enzymatic [30] and non-enzymatic [31] logic gates. However, such gates are still very slow relative to electronic gates due to a reliance on diffusion to transmit information and bond-breaking to perform computations, and are generally harder to design and synthesise than DNA logic gates. Supramolecular logic gates where logical operations are executed via the binding of ions to chelating groups [32] or changes in the binding conformation of rotaxanes [33] have been designed, but as these are also reliant on diffusion and bond-breaking, they too are subject to the same problems as DNA logic gates.

One potential method to mitigate the slow speed of information transmission is to rely on light to transmit information between gates. Reversible molecular logic gates based on molecules which fluoresce in a given wavelength only when bound to specific substrates have been designed [34, 32]. However, the fact that the output of the gate (light) differs fundamentally from the input (matter) renders impossible any attempt

to construct any complex combinatorial circuit from photonic molecular logic gates, reversible or otherwise.

1.3.3 Optical logic gates

Unlike the photonic molecular logic gate design described in Subsection 1.3.2, optical logic gates receive, process and return purely photonic inputs and outputs; thus, there are no inherent physical reasons why large combinatorial optical circuits cannot be constructed. The most common type of optical logic gate processes information by exploiting the interactions of photons with materials possessing a non-linear refractive index such as a photonic crystal [35]. Unlike everyday materials with linear refractive indices, the passage of photons through materials with non-linear refractive indices will alter the refractive index of the medium, thereby altering the path of other photon beams passing through the material and allowing logical operations to be executed. Reversible optical logic gates based on the Mach-Zehnder interferometer have been designed [36].

Because optical logic gates transmit information at the speed of light, they may at first inspection seem to possess a major speed advantage over any other proposed logic gate design. This simplistic analysis, however, ignores that there are two potential rate-limiting steps in information processing in a logic circuit. The first is the time taken for signals to pass from the output of one gate to the input of another, while the second is the time taken for the input to sufficiently alter the state of the processing elements of the gate such that it is capable of generating the desired output with a low error rate (gate delay). While the speed of signal transmission between optical logic gates cannot be surpassed, photons interact far more weakly with matter than electric currents or other molecules. Thus, in order to achieve gate delays on a par with electronic logic gates while maintaining a similar error rate, very high intensities of light must be used [37]; the energy contained in a photon beam used in an optical logic

gate far exceeds the electrical potential energy of a CMOS-based transistor or the net hybridization free energy associated with a DNA logic gate [38]. This high intensity presents problems in creating a thermodynamically reversible optical computer, since the energy contained in the photon beam is significantly more difficult to recover than the switching energy of a transistor.

1.3.4 Mechanical logic gates

Mechanical logic gates provide yet another alternative for implementing reversible logic. The concept of executing logical operations via the interaction of mechanical components is ancient; the two-thousand-year-old Antikythera mechanism [39] was a mechanical analogue computer capable of predicting the positions of astronomical bodies, and Charles Babbage's analytical engine [40], designed in 1837, was the first described Turing-complete computing device. More recently, attempts have been made to design micro- and nanoscale mechanical logic devices. Several such designs have been proposed [41]. These proposals range from performing logical operations via the flexion of stiff micro- or nanoscale rods [42, 43] to reprogrammable mechanical metamaterials [44] and hybrid electro-mechanical devices [45]. Fig. 1.4 [3] illustrates the key components of several of these mechanical computing schemes.

Perhaps the example that best illustrates both the strengths and weaknesses of nanoscale mechanical computing is the design proposed by Merkle *et al.* [46] In this design, logical processing is carried out via rigid rods and rotary joints only. As there are no flexible components in the system, it is claimed that the primary source of energy dissipation is the drag experienced at each joint. In order to ensure that this is the case, the logic gates proposed by Merkle *et al.* appear to have been inten-

[3] Fig. 1.4(a) is reproduced from [43] under the Creative Commons Attribution 4.0 International License. Fig. 1.4(b) is reproduced from [44] under the Creative Commons Attribution 4.0 International License. Fig. 1.4(c) is reprinted with permission from J. Wenzler, T. Dunn, T. Toffoli, and P. Mohanty. A nanomechanical Fredkin gate. Nano Lett., 14:89–93, 2011. Copyright 2011 American Chemical Society.

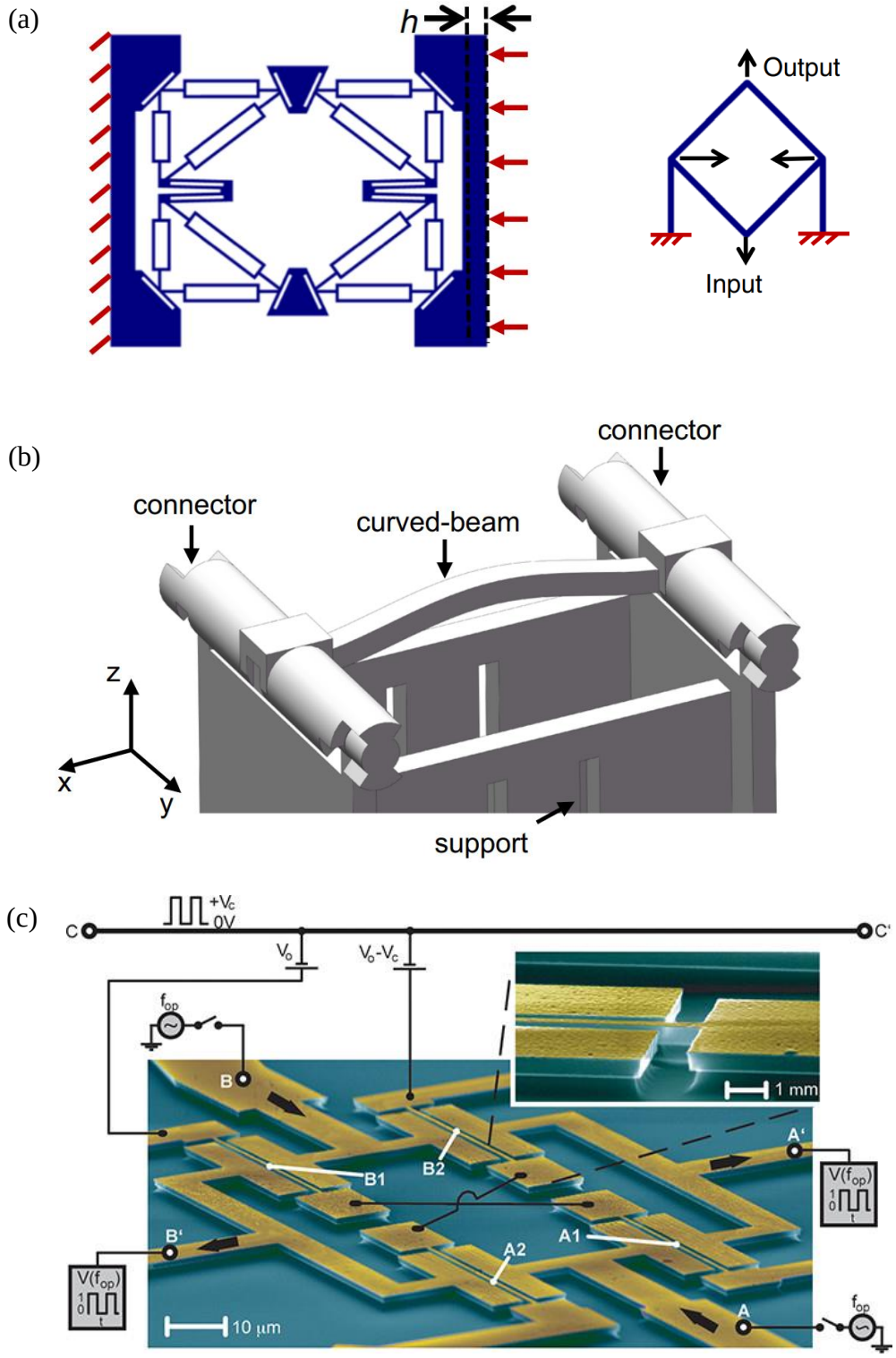


Figure 1.4: A gallery of various mechanical logic gate designs. (a) The micro-flexture-based gates designed by Song *et al.* [43] (b) A mechanical metamaterial-based gate proposed by Mei *et al.* [44] (c) A nanomechanical Fredkin gate proposed by Wenzler *et al.* [45].

tionally designed to have sufficient clearance such that the dispersion force between components would be negligible. However, this simplistic drag-based analysis omits the contribution of the imperfect conversion of kinetic energy between clock cycles towards the dissipation of energy. Based on the figures provided in their paper, the kinetic energies of the proposed gates would be several orders of magnitude greater than $k_{\mathrm{B}}T$ when operating at 100 MHz and 300 K, a direct consequence of designing the gate components to be massive enough such that there would be sufficient clearance to eliminate the dispersion force between components. As all the mechanical logic designs considered thus far require constant starting and stopping of moving components during particular phases of the clock cycle and are at least as massive as the designs proposed by Merkle *et al.*, failing to recover the kinetic energy contained in the gates would result in efficiency losses far greater than $k_{\mathrm{B}}T$ per clock cycle.

Attempting to recover the kinetic energy with nanoscale springs is unlikely to be very efficient. The Q factor is the ratio between the energy contained within a damped harmonic oscillator at the start of a cycle and the energy lost after one radian of oscillation and is inversely proportional to the damping ratio; thus, it functions as an upper bound on the efficiency of kinetic energy scavenging. The Q factor of an atomic force microscopy cantilever in ultra-high vacuum is of the order 10^3 to 10^4 in the 10–100 MHz range [47]. The Q factor is much lower at regular atmospheric pressure, being of the order 10^2 in the 5–50 kHz range [48]. These values, however, cannot be treated as indicative of the energy recovery potential of a realistic mechanical logic device since the logic gates themselves are neither perfectly rigid nor perfectly elastic, and will probably have significantly lower Q factors than purpose-built cantilevers. Furthermore, Q factors generally decrease with increasing frequency; coupled with the quadratic increase in kinetic energy with clock speed in mechanical systems, this would result in a super-quadratic increase of energy dissipation per clock cycle with respect to clock speed. In addition, the restoring force imposed by the springs would

likely increase friction between components, leading to a further decrease in efficiency.

1.3.5 Molecular mechanical logic gates

It is possible to avoid the losses associated with kinetic energy recovery by ensuring that the mechanical logic gates have sufficiently low mass that their kinetic energy does not significantly exceed that predicted by equipartition even at high clock speeds. The equipartition theorem relates the amount of kinetic energy in a system at thermal equilibrium to the number of degrees of freedom it possesses. For a classical system in thermal equilibrium containing N rigid bodies and possessing only harmonic potentials, the average kinetic energy E of the system is given by the following equation:

$$E = 3Nk_BT. \tag{1.6}$$

For the purposes of this thesis, a molecular mechanical logic gate shall be defined as a mechanical logic gate with an average kinetic energy that does not significantly exceed equipartition over the desired range of operational clock speeds. As kinetic energy is directly proportional to mass and proportional to the square of velocity, a major reduction in both the mass and overall dimensions of the logic gate is necessary to reduce its kinetic energy at high clock speeds, greatly increasing the difficulty of avoiding inter-component drag (analogous to friction in macroscale systems) as components will often impinge on the van der Waals radius of other components. In addition, the fact that the entire system must be close to thermal equilibrium precludes any reliance on momentum to ensure the correct positioning of moving components. Therefore, every step of every transition must be a local minimum in free-energy with a sufficiently high barrier to prevent unwanted transitions into erroneous states.

Despite these additional constraints, molecular mechanical logic gates do possess some major advantages over their larger counterparts. Other than the obvious advantage of being smaller and more compact, they would in addition be capable of higher efficiencies at high clock speeds due to inter-component drag and/or viscous drag dominating over the imperfect conversion of kinetic energy as the primary mode of energy loss. These gates would also perform much more efficiently under normal atmospheric pressure or liquid media due to experiencing far less viscous drag due to their smaller size, though it would be unlikely they could achieve the same efficiencies as larger, low-friction gates at very low clock speeds and ultra-high vacuum.

As of the time of writing, the only example of a molecular mechanical logic system which has been designed as such from the outset is the design proposed by the author [49], which will be described in detail throughout the rest of this thesis.

1.4 The importance of an explicit physical model

At present, the techniques of chemical synthesis do not allow for the efficient construction of molecular mechanical logic gates. However, semi-realistic coarse-grained simulations of molecular mechanical circuits would nonetheless allow exploration of the thermodynamic limits of computation in such systems. In coarse-grained models, atoms are not modelled individually; instead, a group of atoms is treated as a rigid body. This has the benefit of both greatly reducing the computational resources required to model a given system, and also reducing the difficulty of designing a viable reversible molecular mechanical logic system. Coarse-grained models are currently used widely in the modelling of complex molecular systems, finding, for example, success in predicting the thermodynamic properties of DNA nanostructures [50], proteins [51] and metal-organic frameworks [52].

The majority of work on the thermodynamics of computing hitherto has either

considered abstract models or simple operations with explicit systems. For instance, Kolchinsky *et al.* [53] consider an abstract model of a reversible Turing machine in order to investigate the relationship between heat generation and the Kolgomorov complexity (a measure of compressibility) of its input-output map. Karel *et al.* [54] use a simple double well potential to derive the minimum energy cost for bit erasure under finite-time conditions. Deshpande *et al.* [55] use a similar double-well potential to derive trade-offs between energy cost, accuracy and time during an optimal bit copying process. Chu *et al.* derive trade-offs between accuracy and energy dissipation of a reading machine [56] and a finite-state machine [57], an abstract machine that can exist in a number of given states with transition rules and hence can be modelled as a Markov chain. Finally, Ouldridge *et al.* [19] consider the thermodynamics of entropy production in deterministic finite automata, a type of finite-state machine that accepts or rejects a given string based on a set of transition rules that trigger the system to switch between its various states.

Although the models described above have produced useful results, the degree of abstraction within them often limits the understanding of the physical constraints on more complex computational systems. A coarse-grained model which explicitly simulates a physical system may be able to highlight phenomena not observable within a more abstract model while avoiding the expense of physical experiments or explicit all-atom simulations. Coarse-grained models are also useful for providing sanity checks against erroneous physical interpretations of purely abstract mathematical models.

Despite these advantages, there are costs associated with adopting a coarse-grained model compared to a purely abstract model. Coarse-grained models still require significant computational resources and time to produce useful results as compared to an abstract model, and there may be major difficulties in designing a physical system within the parameters of the coarse-grained model capable of demonstrating a particular concept.

1.5 Thesis outline

The remaining nine chapters in this thesis describe the design, optimisation and simulation of various molecular mechanical logic gates and combinatorial circuits with a coarse-grained rigid body simulator.

Chapter 2 outlines the specifics of the coarse-grained model and the thermostatted rigid body simulator used to simulate all systems described in this thesis.

Chapter 3 describes the basic circuit components from which all complex circuits in this thesis are constructed, including a novel NAND gate design, the driver and the switch.

Chapter 4 describes the successful demonstration of Landauer's principle and the Szilard engine with a double-bit system. These results show that non-information entropy production can be sufficiently minimised in a molecular mechanical system such that Landauer's principle and the Szilard engine can be demonstrated, and also help confirm that the thermostat of the rigid body simulator is working correctly.

Chapter 5 describes the error rate and thermodynamic reversibility simulations performed on the novel winged NAND gate design. I show that the winged NAND gate has a low error rate even at elevated temperatures across all possible transitions and tends toward thermodynamic reversibility in the limit of clock speeds tested.

Chapter 6 investigates the problems of chaining gates into circuits. I demonstrate a novel mechanism by which gates may be chained in a thermodynamically reversible manner with a low error rate.

Chapter 7 describes the successful design and simulation of a half-adder; these results show that complex combinatorial circuits can be constructed using the components I have designed. I demonstrate the trade-offs between error rate, circuit complexity and speed of a phase-chained combinatorial circuit.

Chapter 8 investigates the advantages and disadvantages of pipelining when applied to simple and complex combinatorial circuits. I show that pipelining results in

an increase in throughput, but may result in logical and thermodynamic reversibility and a large increase in error rate. By modifying the driver design and simulating the effect of persistent information storage, I am able to sufficiently reduce the error rate to successfully demonstrate thermodynamically reversible pipelining of up to four NAND gates.

Chapter 9 describes the application of a genetic algorithm to optimise the various circuit components such as the driver and NAND gate under a variety of conditions. I demonstrate the use of a novel error-triggering potential technique to reduce the otherwise exorbitant computational cost of calculating the fitness function, and explore the effects of high temperature on optimisation.

Chapter 10 summarises the results of all the simulations described in this thesis and outlines possible avenues for future research, including speculations on potential physical realisations of practical molecular mechanical circuitry.

Chapter 2

The Rigid-Body Simulator

A coarse-grained rigid-body simulator, Rigor Mortis (see Appendix B), was written to simulate the logic gates and circuits outlined in this thesis. A rigid-body simulator numerically solves the equations of motion for a system of rigid bodies with specified inter-particle interaction potentials in order to determine their positions, orientations, momenta and angular momenta for a given timestep and set of initial conditions.

2.1 The Langevin thermostat

Numerically solving the equations of motion for a rigid body requires integrating over six independent variables: three translational degrees of freedom along each Cartesian axis and three rotational degrees of freedom about each Cartesian axis. For a system devoid of thermal noise, the solutions are entirely deterministic and can be solved via Verlet integration [58]. However, molecular mechanical logic gates are intended to function in a non-deterministic thermally noisy environment. Circuits composed of such gates immersed in a thermal bath of sufficient size that its temperature would be unaffected by any heat dissipated by the gates would therefore constitute a canonical ensemble, a type of statistical ensemble in which the number of particles in the system N, the volume of the system V, and the temperature T is held constant. While

keeping the number of particles and the volume of the system constant is trivial, maintaining the system at a given temperature while kinetic energy is constantly injected into the system via the driving mechanism of the logic gates requires some form of thermostatting.

Thus, a Langevin thermostat [59] was chosen to maintain the temperature of the canonical ensemble. A Langevin thermostat relies on the Langevin equation, a stochastic differential equation that describes the evolution of a system subject to both deterministic inter-particle potentials and stochastic thermal noise, to model the effects of thermal noise on the canonical ensemble. When applied to rigid-body dynamics, the motion of particles is governed by the Langevin equation of both translational and rotational coordinates and momenta.

The Hamiltonian of the system is $H(\mathbf{r}^N, \mathbf{p}^N, \mathbf{q}^N, \mathbf{\Pi}^N)$ where $\mathbf{r}$ and $\mathbf{p}$ are the positions and momenta of each of the N rigid bodies in the system, while $\mathbf{q}$ and $\mathbf{\Pi}$ are the orientation and angular momenta of each rigid body expressed in terms of quaternions, an extension of the complex numbers into four-dimensional vectors that can be used to efficiently represent angular displacement. Using this Hamiltonian formalism, the translational and rotational Langevin equations can be expressed as

$$\dot{\mathbf{p}}_i = -\frac{\partial H}{\partial \mathbf{r}_i} - \gamma_i \mathbf{p}_i + \sqrt{\frac{2m_i\gamma_i}{\beta}}\mathbf{w}_i(t), \tag{2.1}$$

$$\dot{\mathbf{\Pi}}_i = -\frac{\partial H}{\partial \mathbf{q}_i} - \Gamma_i \mathbf{G}(\mathbf{q}_i, \mathbf{\Pi}_i) + \sqrt{\frac{2M_i\Gamma_i}{\beta}}\mathbf{W}_i(t), \tag{2.2}$$

$$M_i = \frac{4}{\mathrm{Tr}(I_i^{-1})}, \tag{2.3}$$

where β is $1/k_BT$, m_i is the mass, $\mathrm{Tr}(I_i^{-1})$ is the trace of the inverse of the moment

of inertia tensor in the principal axis frame, and γ_i and Γ_i are the translational and rotational damping constants of the i^{th} rigid body. $\mathbf{G}(\mathbf{q}_i, \mathbf{\Pi}_i)$ is a four-dimensional vector that couples the angular momentum quaternion with the orientation quaternion, while $\mathbf{w}_i(t)$ is a three-dimensional Gaussian random variable and $\mathbf{W}_i(t)$ a four-dimensional quaternion Gaussian random variable representing noise due to Brownian motion. Langevin Integrator 'C' from Davidchak et al. [59] was the algorithm used to numerically solve the aforementioned Langevin equations. This integrator is therodynamically self-consistent; that is, it is capable of reproducing the expected Maxwell-Boltzmann distribution of kinetic energies for a system in thermal equilibrium. All simulations were run with a timestep of 3.40 fs.

The translational and rotational damping constants for single beads were calculated using the equations for Stokes' drag of a spherical object:

$$\gamma = 6\pi\mu(\sigma + d)/m, \tag{2.4}$$

$$\Gamma = 8\pi\mu(\sigma + d)^3 \mathrm{Tr}(I^{-1}), \tag{2.5}$$

where μ is the viscosity of the solvent. For non-spherical agglomerations of multiple beads, the value of $\sigma + d$ was approximated using the radius for a solid sphere of identical mass and density.The dynamic viscosity of air at 298 K (1.81×10^{-5} kg m^{-1}s^{-1}) was used in all simulations. At this viscosity, the system is overdamped within the range of clock speeds tested, and the kinetic energy of a single logic gate and its associated drivers does not exceed $0.3\, k_{\mathrm{B}}T$ beyond what would be expected from equipartition even at the highest clock speed tested (2.87 GHz). The damping term applied is only an approximation of the true drag experienced by the rigid bodies as it neglects the asphericity of the bodies and does not take into account their hydrodynamic interactions. However, a highly accurate model of drag is neither necessary

nor useful given that the system is designed to be a coarse-grained representation of large rigid molecules and not a precise model of a specific molecular system.

2.2 Inter-particle interactions

Efficiently evaluating all the pairwise interactions within a large multi-particle system is a non-trivial matter. The number of possible pairwise interactions $f(N)$ between N particles is given by Eq. 2.6:

$$f(N) = \frac{N(N-1)}{2}. \tag{2.6}$$

Because the number of pairwise interactions scales quadratically with the number of particles in the system, the brute-force method of evaluating all possible interactions is not practical for large systems. Therefore, a more sophisticated algorithm is necessary to mitigate the asymptotic complexity of evaluating the pairwise interactions.

2.2.1 The Cell List

One method of reducing the computational cost of evaluating the list of pairwise interactions is to introduce a cut-off beyond which the interaction is not evaluated. Although this modification will speed up the algorithm by a constant term, it does not reduce the asymptotic complexity of the brute-force algorithm, since the distance component of each pairwise interaction must still be evaluated.

However, the implementation of a cell list allows all but the nearest interactions to be ignored, reducing the asymptotic complexity from $O(N^2)$ to $O(N)$ [60, 61]. In this data structure, the region of space over which the system is simulated is divided into cubic cells with an edge length greater than or equal to the cut-off distance beyond which the interaction is not evaluated. In such a scenario, only interactions between particles in cells which meet along an face, edge or corner have to be evaluated, since

any interactions between particles which do not meet the aforementioned criteria must have inter-particle distances exceeding the cut-off. As there are only 26 such neighbouring cells at most for every given cell, if it is assumed that there is an upper bound on the number of particles each cell can realistically contain (a reasonable assumption for virtually all systems), the number of particles, and hence, the number of pairwise interactions between such particles must be bounded by a constant k. Therefore, the number of pairwise interactions to be evaluated is at most kN, which is of $O(N)$ complexity.

Although the cell list allows for a substantial reduction in asymptotic complexity, it is still costly to evaluate if naively implemented. In particular, it is necessary to search all neighbouring cells for potentially interacting particles during each timestep. It is possible to further improve the efficiency of the pairwise interaction evaluation algorithm such that the cost of this search is greatly reduced via the implementation of a Verlet list.

2.2.2 The Verlet List

The Verlet list is a data structure generated for each particle which includes all particles within a radius d_c+2s, where d_c is the cut-off distance and s is a distance referred to as the Verlet skin [62]. In order to evaluate the pairwise interaction potential, only interactions in the Verlet list need to be considered. Provided that s is reasonably small, there are generally far fewer interactions to be considered for a Verlet list than for a naively implemented cell list, leading to a large speedup in the evaluation for the average timestep.

However, the Verlet list will need to be updated whenever a particle's total displacement exceeds the magnitude of the Verlet skin, as there is now a possibility that two particles formerly outside the cut-off radius will now have travelled a sufficient distance to be inside it. Updating the Verlet list by naively considering every pair

of interactions will take $O(N^2)$ time, much like the case of the brute-force algorithm. Fortunately, it is possible to greatly speed up the process of updating the Verlet list by using a cell list with an edge length equal to $d_c + 2s$ to find all particles which could potentially belong to the Verlet list. This improvement reduces the asymptotic complexity of the update operation to just $O(N)$, while maintaining the superior speed of the Verlet list during an average timestep [63].

The algorithm implemented by Rigor Mortis to evaluate pairwise interactions is a Verlet list updated with a cell list as described above, albeit with a few modifications to take advantage of the fact that pairwise interactions between particles which are components of the same rigid body can be ignored. This is accomplished by grouping particles which belong to the same rigid body into distinct categories before the Verlet list update algorithm is executed. Subsequently, the algorithm simply ignores all interactions between particles in the same category. This allows for a reduction in the size of the Verlet list and a consequent increase in efficiency.

2.2.3 Boundary conditions

The boundary conditions of a dynamics simulation refer to the dimensions of the space over which the cell list is constructed and the potentials are evaluated. It is common to use periodic boundary conditions in most molecular dynamics simulations [64], wherein an infinitely large system is simulated by treating cells at one edge of the simulation space as neighbours of cells at the other edge, and allowing particles to interact and pass between these cells.

However, all systems simulated by Rigor Mortis are molecular mechanical logic gates which are constrained in space by strong potentials and are thus unlikely to diffuse any significant distance. Furthermore, no explicit solvent molecules are modelled, with the damping and thermal noise exerted by the solvent on the system being simulated by the damping and noise terms of the Langevin equation. Thus, it is sufficient

for Rigor Mortis to implement only fixed non-periodic boundary conditions.

2.2.4 The Lennard-Jones and WCA potentials

A widely-used pairwise potential for interactions between non-bonded atoms implemented by most molecular dynamics software packages such as AMBER and GROMACS is the Lennard-Jones potential. The Lennard-Jones potential is an easily-computed approximation of the forces experienced between two atoms not directly bonded together and takes the following form:

$$V_{\mathrm{LJ}}(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right], \tag{2.7}$$

where r is the distance between the two atoms and ϵ and σ are the potential depth and van der Waals radius respectively [65].

The standard Lennard-Jones potential is, however, inadequate for modelling objects consisting of multiple atoms as opposed to the single atoms. In such cases, a step-off distance d can be added to the inter-particle distance r to simulate the cores of the multi-atom particles which interact negligibly with each other:

$$V_{\mathrm{LJm}}(r) = 4\epsilon \left[\left(\frac{\sigma}{r-d}\right)^{12} - \left(\frac{\sigma}{r-d}\right)^{6} \right]. \tag{2.8}$$

The fact that the Lennard-Jones potential is never zero for any positive value of r forces the use of an arbitrary cut-off distance to reduce the computational cost. If computational efficiency is desired over accuracy, the Weeks-Chandler-Anderson (WCA) potential (Eq. 2.9) may be used instead:

$$V_{\mathrm{WCA}}(r) = \begin{cases} 4\epsilon[(\frac{\sigma}{r-d})^{12} - (\frac{\sigma}{r-d})^{6}] + \epsilon, & r-d \leq 2^{\frac{1}{6}}\sigma \\ 0, & r-d > 2^{\frac{1}{6}}\sigma. \end{cases} \tag{2.9}$$

The WCA potential is a modified Lennard-Jones potential which is increased by ϵ and truncated at the Lennard-Jones potential minimum [66]. This modification ensures that there is no long-range attractive term, which allows the cut-off radius of the Verlet list to be significantly reduced, speeding up calculations. All simulations in this thesis use the WCA potential to model inter-particle interactions.

The large, medium and small coloured beads in the figures and animations represent three distinct particle types (see Fig. 2.1 for an example containing all three types of particles), corresponding approximately to masses of 120, 41 and 14 amu, respectively, corresponding roughly to groups containing 10, 3 and 1 carbon atom(s). The beads are not meant to represent any particular organic moiety, as it is not my intention to create an accurate model of a specific structural motif.

Physically reasonable values of ϵ and σ were obtained by fitting to appropriate intermolecular potentials. For the large beads, these values were obtained by plotting the Lennard-Jones potential of two adamantane molecules from a distance of 0.5 to 2.0 nm using the Generalized Amber Force Field (GAFF) parameters for carbon and hydrogen [67], while d was determined by least-squares fitting. Values of $\epsilon = 3.33$, $\sigma = 0.2\,\text{nm}$ and $d = 0.15\,\text{nm}$ were obtained; ϵ values are given in units of $k_\text{B}T$ at 298 K. The step-off distance ensured a more accurate fit to the plotted values than would otherwise be possible with the standard Lennard-Jones potential calculated using the distance between the centres of two beads. For the small beads, values of $\epsilon = 0.2$, $\sigma = 0.19\,\text{nm}$ and $d = 0\,\text{nm}$ were obtained directly from the GAFF parameters for the methyl group. The radius parameters $\sigma = 0.14\,\text{nm}$ and $d = 0.11\,\text{nm}$ for the medium-sized beads were obtained by scaling down the constants for the large beads by assuming a cubic relation between van der Waals radius and mass, while the value of $\epsilon = 0.8$ was found by assuming that the mass of a bead m is approximately related to ϵ via a simple polynomial equation $\epsilon = Am^n$, where A and n are constants to be solved for using the known values of m and ϵ of the small and large beads. The values

Table 2.1: Table of values of ϵ, σ and d for the three bead types.

Bead mass	ϵ	σ	d
14	0.2	0.19	0
41	0.8	0.14	0.11
120	3.33	0.2	0.19

of ϵ, σ and d for each bead type are recorded in Table 2.1. To calculate the values of σ, d and ϵ between two different bead types, the expressions

$$\sigma_{ij} = (\sigma_i + \sigma_j)/2, \tag{2.10}$$

$$d_{ij} = (d_i + d_j)/2, \tag{2.11}$$

$$\epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j}, \tag{2.12}$$

were used to find the mixed parameters σ_{ij}, d_{ij} and ϵ_{ij} for each bead type i and j. These equations are identical to those used by standard molecular dynamics packages such as AMBER and GROMACS, albeit with the modification that $\sigma + d$ is used in place of σ.

Within the rigid bodies, beads are distributed in a manner such that their van der Waals radii overlap to the extent that other rigid bodies are unable to pass through or approach too closely without experiencing a large repulsive force. This distance is generally less than 0.375 nm for the 120 amu beads and less than 0.25 nm for the 41 amu beads. The smallest beads are generally not integrated into larger rigid bodies but are instead used as individual components of flexible chains. Repulsive forces between two rigid bodies directly bonded together are suppressed.

2.2.5 Harmonic bond and angular potentials

In order to efficiently approximate the potential energies of interactions between bonded atoms without resorting to costly quantum mechanical calculations, molecular dynamics software packages generally rely on two-, three- and four-atom simple harmonic potentials. The two-atom potentials are intended to restrain directly-bonded atoms to a set distance of each other, and are referred to as bond potentials. These potentials take the following form:

$$V_{\text{bond}} = k_{\text{bond}}(r - r_0)^2/2, \tag{2.13}$$

where k_{bond} is the bond harmonic constant, r_0 the equilibrium bond distance and r the distance between the two particles which are bonded together. For Rigor Mortis, the bond harmonic constants k_{bond} were all set to $214\,\text{N}\,\text{m}^{-1}$, approximately half the strength of a sp^3 carbon-carbon single bond from GAFF.

The three-atom potentials serve to constrain the angle between three atoms, two of which are directly bonded to a central atom. These potentials take the following form:

$$V_{\text{angle}} = k_{\text{angle}}(\theta - \theta_0)^2/2, \tag{2.14}$$

where k_{angle} is the angular harmonic constant, θ_0 the equilibrium angle and θ the angle between the relevant interparticle vectors. For Rigor Mortis, the angular harmonic constants k_{angle} were set to $520\,\text{kJ}\,\text{mol}^{-1}\text{rad}^{-2}$, approximately equal to the bending potential of a chain of three sp^3 carbon atoms; however, it should be noted that as there are multiple angular potentials centred on a realistic carbon atom (typically two to three depending on the hybridization state of the central atom), the angular potential I have used would be weaker than the actual angular potential felt by two

Table 2.2: Table of values of the bond and angle harmonic constants and the dipole interaction constant.

Potential type	Strength
Bond	$214\,\mathrm{N\,m^{-1}}$
Angle	$520\,\mathrm{kJ\,mol^{-1}rad^{-2}}$
Dipole	$496\,\mathrm{kJ/mol}$

rigid sp^3 carbon-based structures bonded together. The bond and angular potentials used are recorded in Table 2.2.

Two types of bonds were used to connect the rigid-body systems. Both used the same bond harmonic potential, but differ in the angular potentials used to restrain the relative positions of their parent rigid bodies. The first type, coloured silver in all figures, represents a bonding interaction between two particles A and B, and also implies two sets of angular potentials $\measuredangle XAB$ and $\measuredangle ABY$ where X is an arbitrary position on the rigid body to which particle A belongs that is fixed with nonzero distance relative to particle A, and Y a similar position on the rigid body to which particle B belongs. The second type, coloured yellow, represents only a bonding interaction between A and B, without any angular potential. In Fig. 2.1, the first type of bond is labelled as "Bond" and the second type as "Tether". Note that the silver spheres in Fig. 2.1 are not standard particles and instead represent a scaffold to which the rigid bodies are constrained. The scaffold particles are fixed in space and particles directly bound to the scaffold are constrained by a harmonic potential centred on their initial position.

2.3 Assessment of the model

It is acknowledged that the model as presented is not a realistic representation of any specific molecular system. However, as mentioned in Section 2.2.4, it is not my intention to create a realistic model of a specific structural motif. The bond

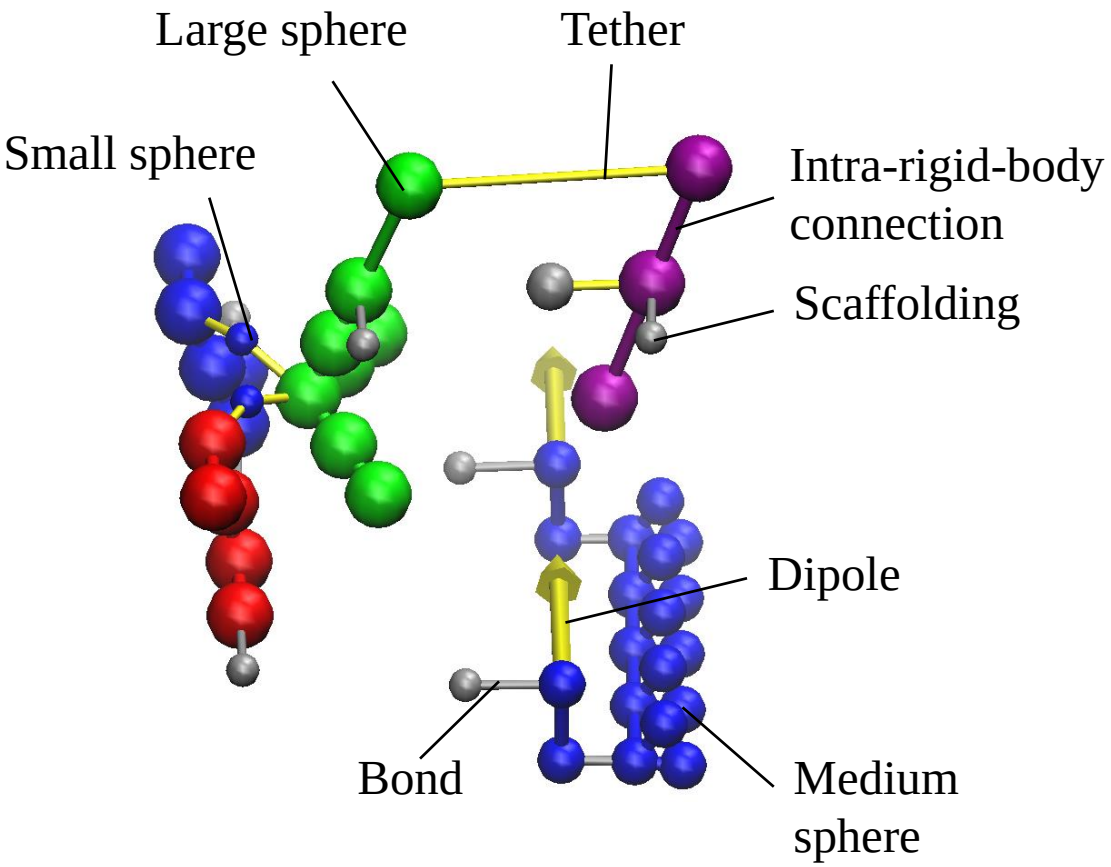


Figure 2.1: An example of a rigid body system containing the various types of bonds and particles described in the previous two sections. A similar scheme is used to illustrate other systems in this thesis.

and angle harmonic constants used are intended to roughly approximate what would be possible with an extremely rigid organic molecule; for instance, one constructed around an adamantane [68] or polyyne [69] motif. It should be noted that although the spring constant of harmonic oscillators in series is inversely proportional to the number of oscillators, the spring constant of harmonic oscillators in parallel is directly proportional to the number of oscillators [70]; thus, reaching an arbitrary desired stiffness in tension should theoretically be possible by increasing the cross-sectional area of the components, though this will of course result in an increase in the size and complexity of the system.

Even under these greatly simplified conditions, substantial difficulties were encountered designing and optimising the molecular mechanical circuit components by hand, to the point that algorithmic optimisation via a genetic algorithm was investigated (Chapter 9). Constraining the model to a specific organic motif would not only greatly increase the difficulty of designing the structure by severely constraining the available solution space, but would also significantly increase the number of particles and interactions involved and thus the computational cost of simulations, which would lead to far longer design cycles for questionable benefit as the selected motif would likely be impractical to synthesise in the near future regardless.

Thus, I judged that not selecting a specific organic motif to construct a coarse-grained model around would be a prudent decision. Given the difficulties encountered in optimising these molecular mechanical systems even with algorithmic assistance (see Chapter 9) due in no small part to the heavy computational cost associated in estimating error rate and thermodynamic efficiency in thermally noisy systems, I believe that my decision to not overly constrain the available solution space has been retrospectively vindicated.

Although the model is simplistic, it nonetheless remains useful for demonstrating the viability of the NAND gate design and other circuit components by allowing their

error rate to be estimated. For instance, the error rate of the winged NAND gate design (Section 5.1) remains sufficiently low even at highly elevated temperatures such that the basic design would likely remain viable even if the maximum stiffness of the bond and angular potentials that could be practically attained was only one-fourth of the harmonic constants used in the model. Therefore, it is likely to be an efficient molecular mechanical gate design that warrants further investigation and optimisation.

Furthermore, the findings I have obtained which pertain to the thermodynamics of logical irreversibility (Chapters 4 and 8) would still hold regardless of the practicality of the logic gate designs as the model is thermodynamically self-consistent and the same simulation parameters are used to compare different protocols. The same holds true for the results pertaining to the mechanics of chaining logic gates together and the trade-offs between ensuring a low error rate at the price of increasing complexity (Chapters 6, 7 and 8).

2.4 External controls

An important part of our approach is that the external control used to drive the computations should be simple [29, 16]. Two reasons underlie this philosophy; firstly, it is common within the field of stochastic thermodynamics to assume that an arbitrary time-varying potential can be applied to the system's coordinates with essentially no external cost. However, it is unclear how efficient transfer of work to and from the computational system can actually occur in this setting. Secondly, the external costs of even a simple control can only be neglected if the system being controlled is relatively large [29]. Any control externally applied must therefore be applicable in parallel to many systems, such that the cost of application is amortised over all systems being driven in parallel.

Therefore, our systems are driven via one or more external dipoles that rotate at constant angular velocity until a change of direction is needed, at which point the angular velocity is negated. The dipole potential takes the form

$$V_{\text{dipole}} = k_{\text{dip}} \hat{\mathbf{d}}_{\text{ext}} \cdot \hat{\mathbf{d}}_{\text{int}}, \tag{2.15}$$

where $\hat{\mathbf{d}}_{\text{ext}}$ and $\hat{\mathbf{d}}_{\text{int}}$ are the direction vectors of the external and internal system dipoles respectively, and $k_{\text{dip}} = 496\,\text{kJ/mol}$ for all dipoles. This interaction provides a simple generic means of rotary mechanical transmission that need not necessarily be implemented using electromagnetic dipoles. In keeping with this simplified approach we do not specify the position of the external dipole or include any distance dependence in the interaction. Each internal dipole interacts with the external dipole independently, and its phase relative to other internal dipoles within the system can be controlled.

Such a simple control mechanism that requires no feedback from the computational state of the machine makes the efficient transfer of work to and from the computational system more plausible. Moreover, a control of this kind allows, at least in principle, for a single external protocol that rotates multiple dipoles within one or more circuits together, provided that they maintain a fixed relative phase, thus allowing the external cost of the control to be neglected. More sophisticated protocols with variations in relative phase or frequency could be physically implemented while maintaining only a single external dipole via the use of reduction gearing.

The following expression was used to calculate the work done on the gate by the driving force of a single dipole:

$$E_{\text{dipole}}(t, \Delta t) = k_{\text{dip}}(\hat{\mathbf{d}}_{\text{ext}}(t + \Delta t) - \hat{\mathbf{d}}_{\text{ext}}(t)) \cdot \hat{\mathbf{d}}_{\text{int}} \tag{2.16}$$

where $\hat{\mathbf{d}}_{\text{ext}}(t)$ is the direction vector of the external dipole at time t, Δt is the length

of one timestep, and E_{dipole} the work done by the external dipole on a given internal dipole across the timestep Δt. Dipoles are marked with yellow arrows in figures (see Fig. 2.1 for an example structure containing a dipole).

The expression above is based on that used by Sekimoto [71] to calculate the work done by an external potential in a system subject to a Langevin thermostat. Intuitively, it numerically integrates over the change in potential resulting from altering the control parameter (in this case, the position of the external dipole) for each timestep. A similar expression has also been used to calculate the work done in a physical realisation of Landauer's principle [8]. This expression converges on the true value of the work done on the system in the limit of infinitesimally small timesteps. The work done on the system by external forces cannot be calculated by measuring heat dissipated from the canonical ensemble as the Langevin thermostat maintains thermal equilibrium of the system indirectly through the noise term and the average kinetic energy of the solvent molecules is never explicitly calculated.

Chapter 3

Circuit Components

In order to construct an arbitrary combinatorial circuit, in addition to at least one type of universal Boolean logic gate (typically a NAND gate), a mechanism for connecting the input of one gate to the output of another and a device capable of converting signals from the clock into a means of selectively transducing information between connected gates (henceforth referred to as a driver) is also required. I shall now introduce these components in detail.

3.1 The Driver

The driver is responsible for converting the mechanically-driven clock signal into a means of transducing information between a fixed input bit and a gate or between two gates via a clutch whose engagement is a function of the phase of the clock. Fig. 3.1 illustrates the driver design.

The driver arms function in a manner similar to a piston rod. The external dipole rotates in the plane henceforth arbitrarily defined as the xz-plane with the x-axis corresponding to the alignment of the dipole at a state of minimal/maximal engagement. When the external dipole rotates, it forces the internal dipole to follow synchronously, moving the purple arm of the driver into the position of greatest

distance between it and the clutch (see Fig. 3.1). The cyan and orange segments are pulled into a position such that they lie parallel to the purple arm, bringing the red clutch into a position where it can transfer via steric interactions the state of the blue input bit to the green output bit.

When the purple arm is rotated to the position of minimal distance from the clutch, it forces the orange arm to take on a pitch of roughly 45° from the plane of the scaffold, forcing the clutch away and decoupling the output and input bits from each other (Fig. 3.1a). In addition to providing a rigid attachment point, the scaffolding also prevents the driver arms from rotating below the plane defined by the scaffold, eliminating any possibility of the driver taking two different states, a scenario that complicates reversibility simulations.

The blue input bit is restrained to an angle of 45° from the z-axis by a dipole potential identical in strength to the one used for the interaction between the internal and external dipoles (Eq. 2.15); bits with a positive slope in the yz-plane (pointing away from the viewer) are arbitrarily defined as '1' and bits with a negative slope as '0'. In Fig. 3.1, all input bits have been set to '1'. As a circuit component, the driver does not always connect a restrained input bit to an output bit, but can also be inserted between the output bit of one gate and the input bit of another, allowing the selective transduction of information based on the clock cycle.

Due to the geometry of other circuit components, two distinct driver types are used throughout this thesis, one arbitrarily designated as a right-handed driver (Fig. 3.1) and the other a left-handed driver. Although the rigid bodies of the two driver types are mirror images of each other, the interaction potentials are not, as the internal dipole for each driver points in the same direction. Therefore, both drivers rotate in the same direction with the same phase.

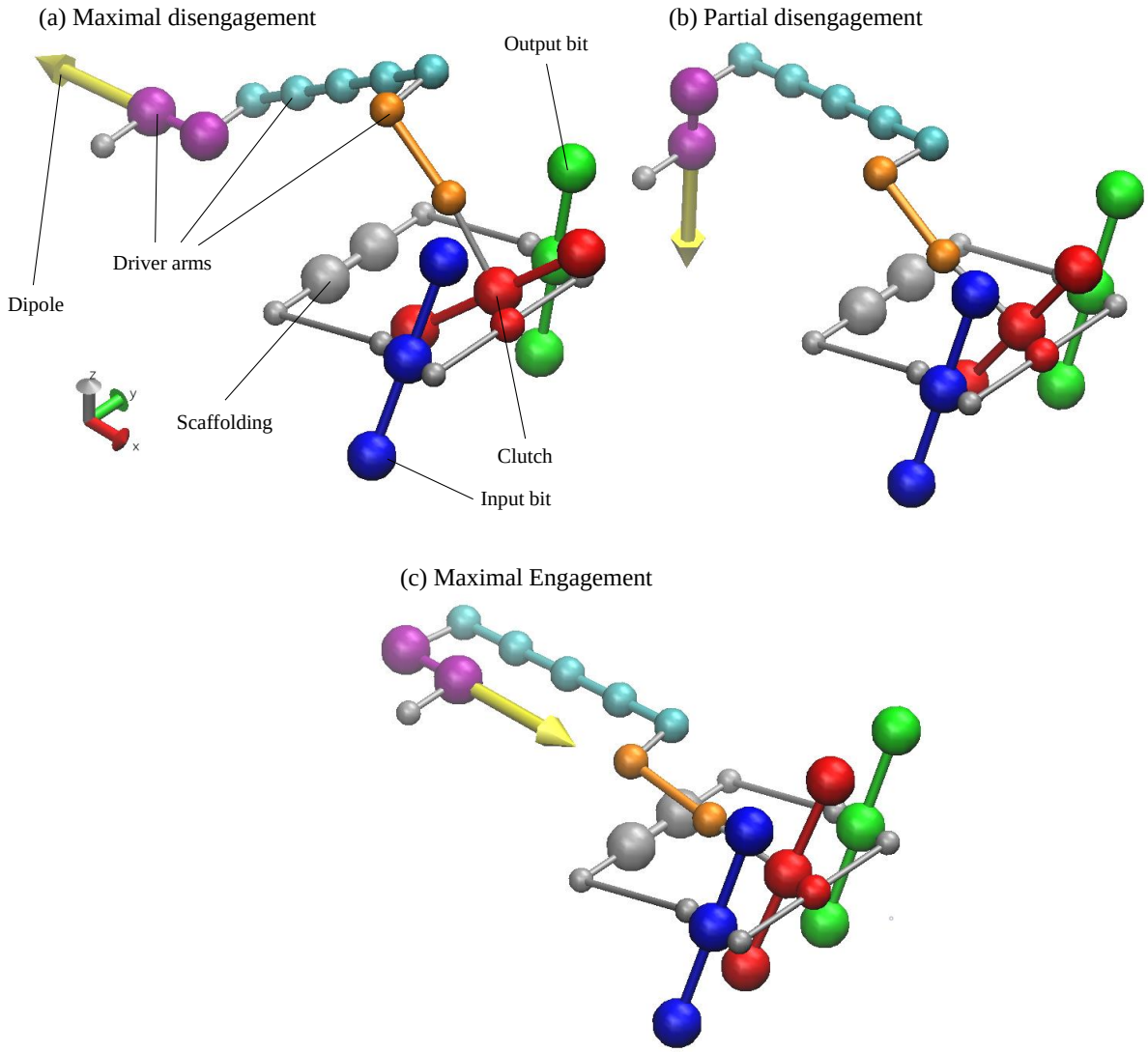


Figure 3.1: The driver system used to selectively transduce information depending on the phase of the clock. Clockwise from the upper left, these diagrams depict a right-handed driver in states of increasing degrees of engagement. Supplemental Video 1 provides an animation of this process. A link to a playlist containing all Supplemental Videos in this thesis may be found in Appendix C.

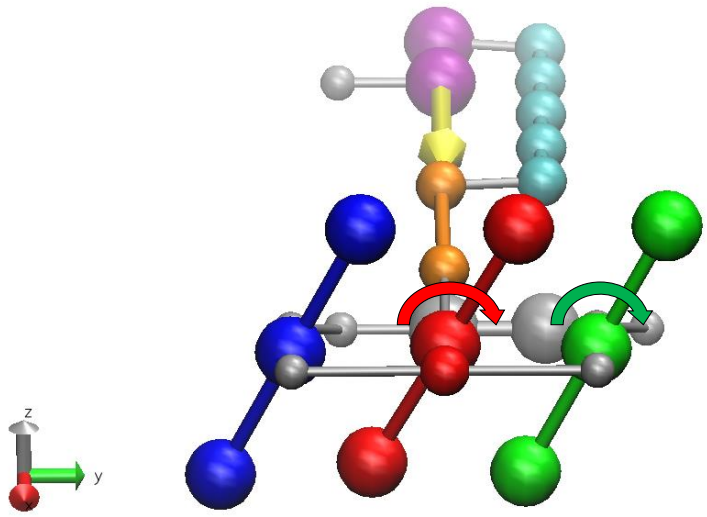


Figure 3.2: The driver system at maximal engagement visualised along the x-axis. The coloured arrows illustrate the ability of the clutch and output bit to rotate about the x-axis to match the state of the input bit. Note that the clutch is capable of rotating along two axes (x and y).

3.2 The NAND gate

As mentioned in Section 1.2.1, any combinatorial Boolean logic gate may be constructed from NAND gates alone. One of the most intuitive methods of designing a molecular mechanical NAND gate is to conceptualise it as a two-bit majority-voter gate. The N-bit majority voter gate accepts N inputs and returns 1 if a majority of the N inputs are set to 1; otherwise, it returns 0 [72]; therefore, a two-bit majority voter gate is synonymous with an AND gate. NAND gates may in turn be trivially constructed from AND gates by inverting the output bit. A two-bit mechanical majority voter gate may be implemented in a fairly simple manner by using one or more springs to restrain the gate to a default position corresponding to the '0' state. These springs are countered by an opposing force that is applied by the input bits when shifted into the '1' state; however, this force is only sufficient to overcome the restraining force supplied by the springs when both input bits are shifted into the '1' state (Fig. 3.3).

Input 1	Input 2	Input 3	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Table 3.1: Boolean truth table of the 3-bit majority voter gate.

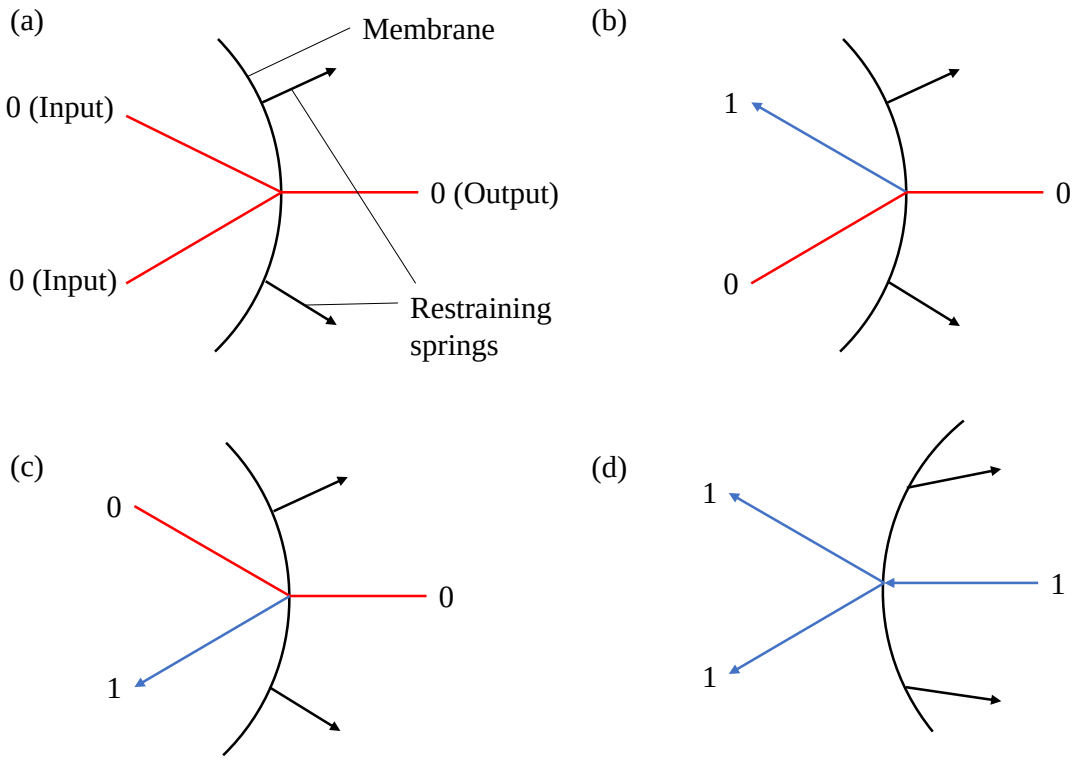


Figure 3.3: A simplified illustration of the operation of the two-bit majority voter AND gate. When the input bits are switched from '0' to '1', they exert a pulling force on the membrane; this force is represented by an arrow. The membrane is restrained by springs in such a manner that it favours a convex orientation viewed from the left. The restraining potential can only be overcome by both input bits being set to '1', which forces the membrane into a concave orientation when viewed from the left. This deflection pulls the output bit into the '1' state.

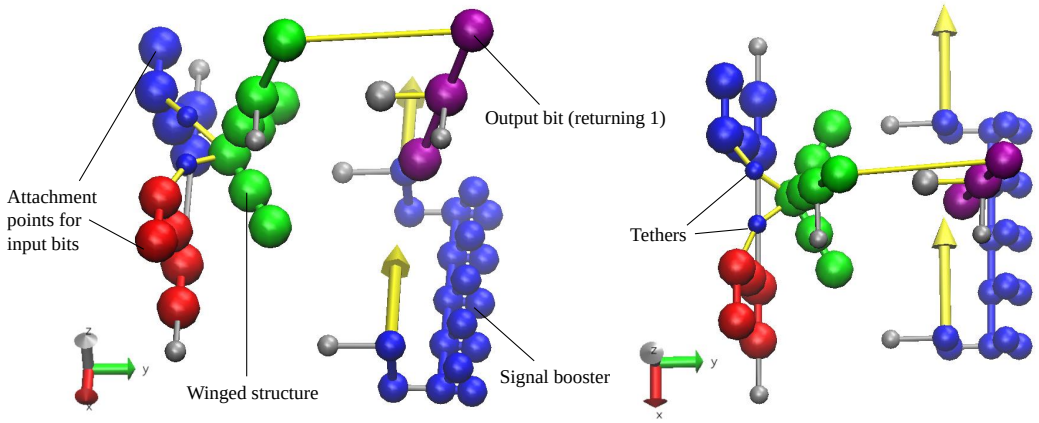


Figure 3.4: The winged NAND gate, which constitutes the basic universal Boolean gate used in circuit construction, viewed from two different orientations. Note that the winged structure can rotate freely about the x- and z-axes. The orientation of the winged structure is that for when the inputs are both zero.

Despite its simplicity, the two-bit majority voter-based NAND gate does bear the disadvantage that its four possible states ('00', '01', '10' and '11') will have significantly different energies due to the strain caused by the force exerted by the input bits acting against the restraining spring. This strain, which is intrinsic to the operation of the gate, will reduce the efficiency of the design if the potential energy stored in the spring cannot be fully recovered.

To avoid the potential problems associated with the majority voter gate, I designed the molecular mechanical NAND gate outlined in Figs. 3.4 and 3.5. The output bit of the gate is connected to a freely-rotating pair of prong- or wing-like structures, hence the term "winged" NAND gate. The tip of the output bit is tethered to the tips of the input bits via the winged structure.

When the input bits are fixed to the states '00', the output bit is pulled by the input bits into the '1' configuration. When the input bits are fixed to the states '00', the winged structure is pulled by the input bits into the '1' configuration, and the output bit with it (see Fig. 3.6 for a clearer illustration of this operation).

When the input bits are set to '01' or '10', the winged structure is simultaneously pulled forward by the '0'-set bit while being pushed forward by the '1'-set bit; however,

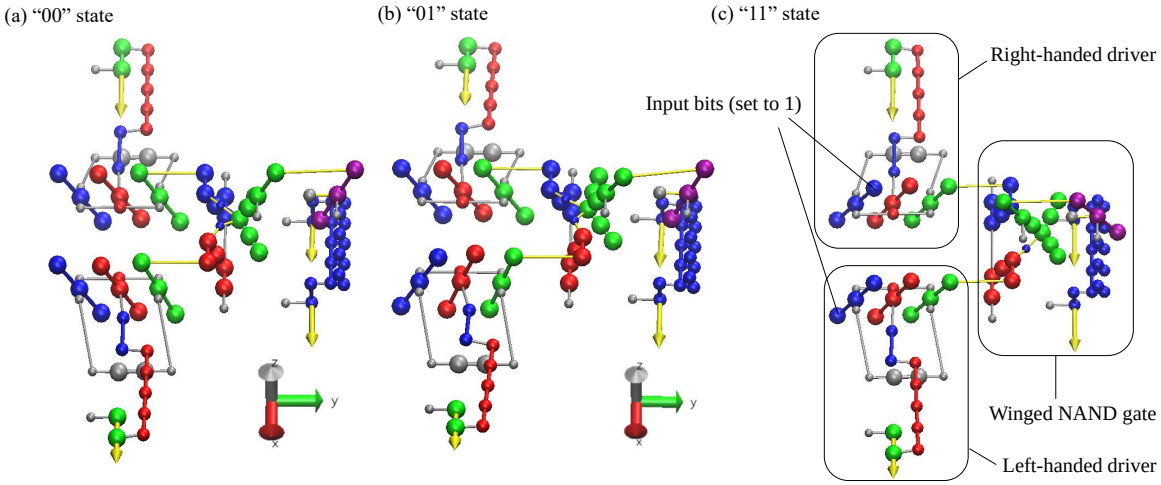


Figure 3.5: The winged NAND gate connected to two engaged drivers with inputs (a) ‘00’, (b) ‘01’ and (c) ‘11’, illustrating how the states of the inputs affect the orientation of the winged structure and hence the output.

the ability of the wings to freely rotate about the axis of the winged structure is forced into the ‘0’ configuration (see Fig. 3.7 for a clearer illustration of this operation).

When the input bits are set to ‘11’, this rotation is not possible and the output bit is forced into the ‘0’ configuration (Fig. 3.8).

The winged NAND gate design has the benefit of not intrinsically relying on internal strain to achieve the desired output like the majority voter gate, though it is acknowledged that practically designing a winged NAND gate with minimal strain is not trivial.

As the angular deflection of the gate output induced by the gate inputs is smaller than the deflection of the input bits, a signal booster is required to increase the deflection of the output bit. The signal booster is a wedge-shaped block that is driven by the external dipoles and remains in phase with the gate drivers. When the drivers are maximally engaged, it forces the output bit away from the y-axis, thereby boosting the signal. The signal booster at maximal engagement forcing the output bit away from the y-axis is visible at $t^* = 1$ in Figs. 3.6, 3.7 and 3.8.

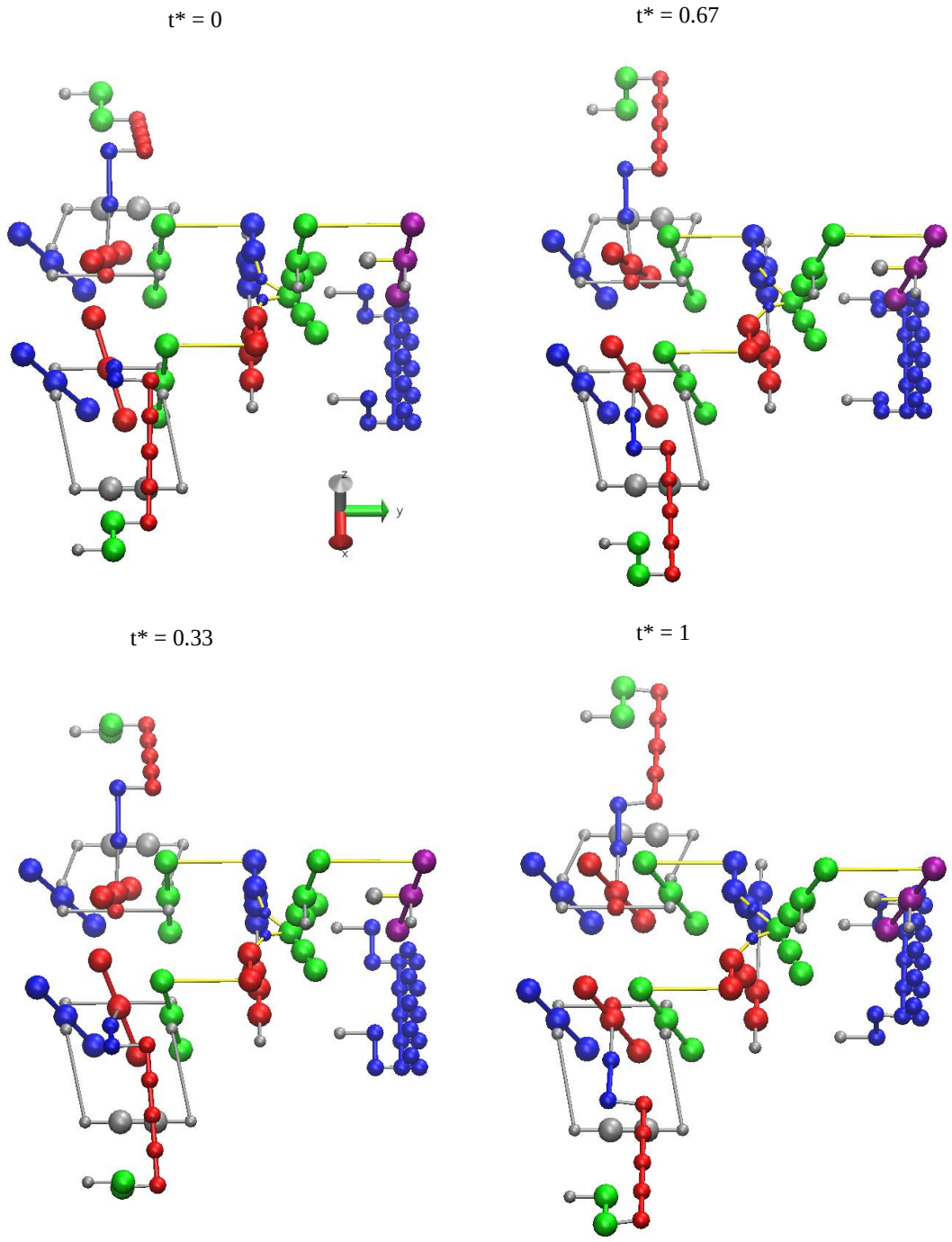


Figure 3.6: The NAND gate with inputs '00' connected to a pair of drivers being driven from a disengaged to an engaged (signal-boosted) state. t^* is the current time fraction of the total time required to complete the full process. The full process in the absence of thermal noise may be viewed in Supplemental Video A.

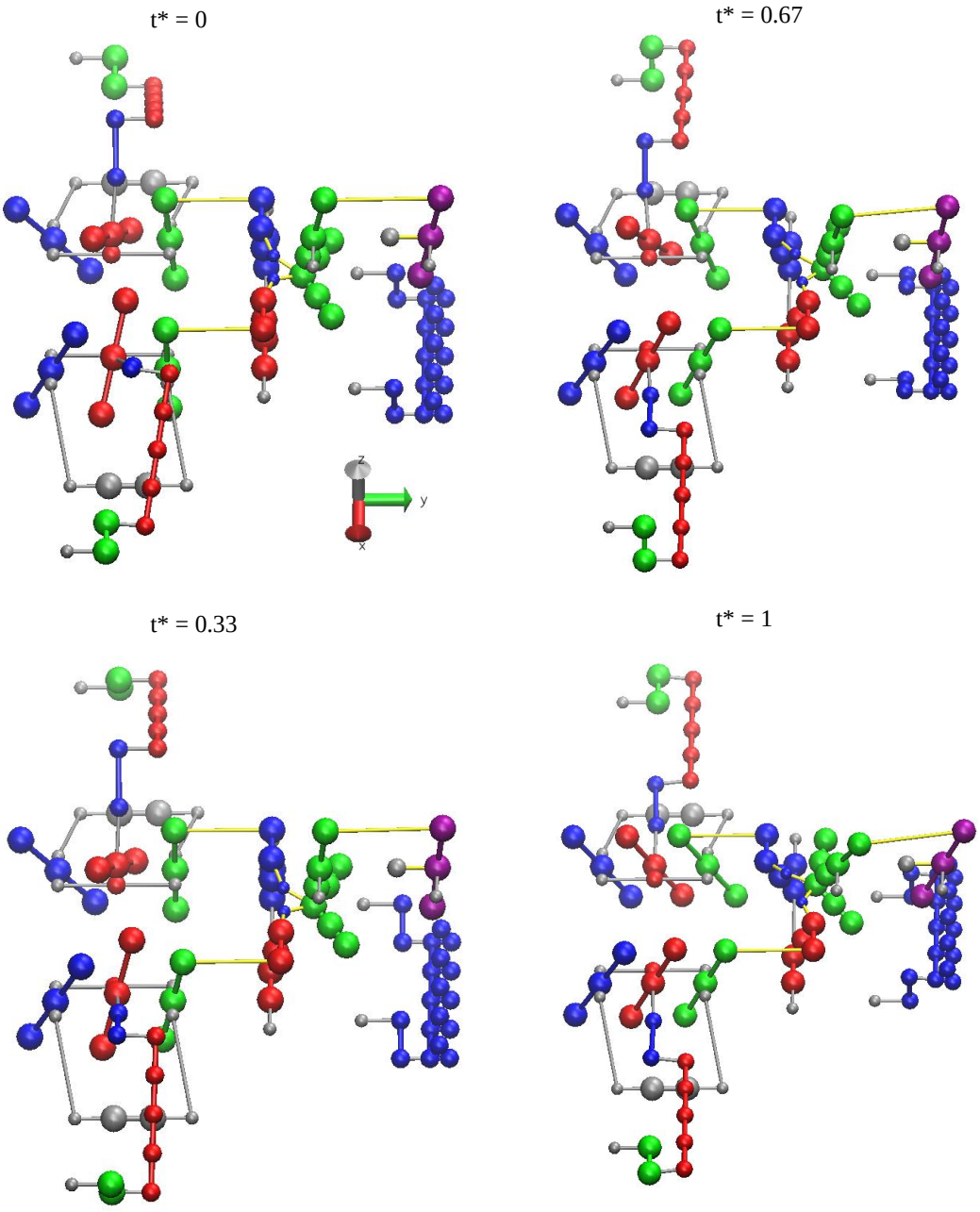


Figure 3.7: The NAND gate with inputs '01' connected to a pair of drivers being driven from a disengaged to an engaged (signal-boosted) state. The full process in the absence of thermal noise may be viewed in Supplemental Video B.

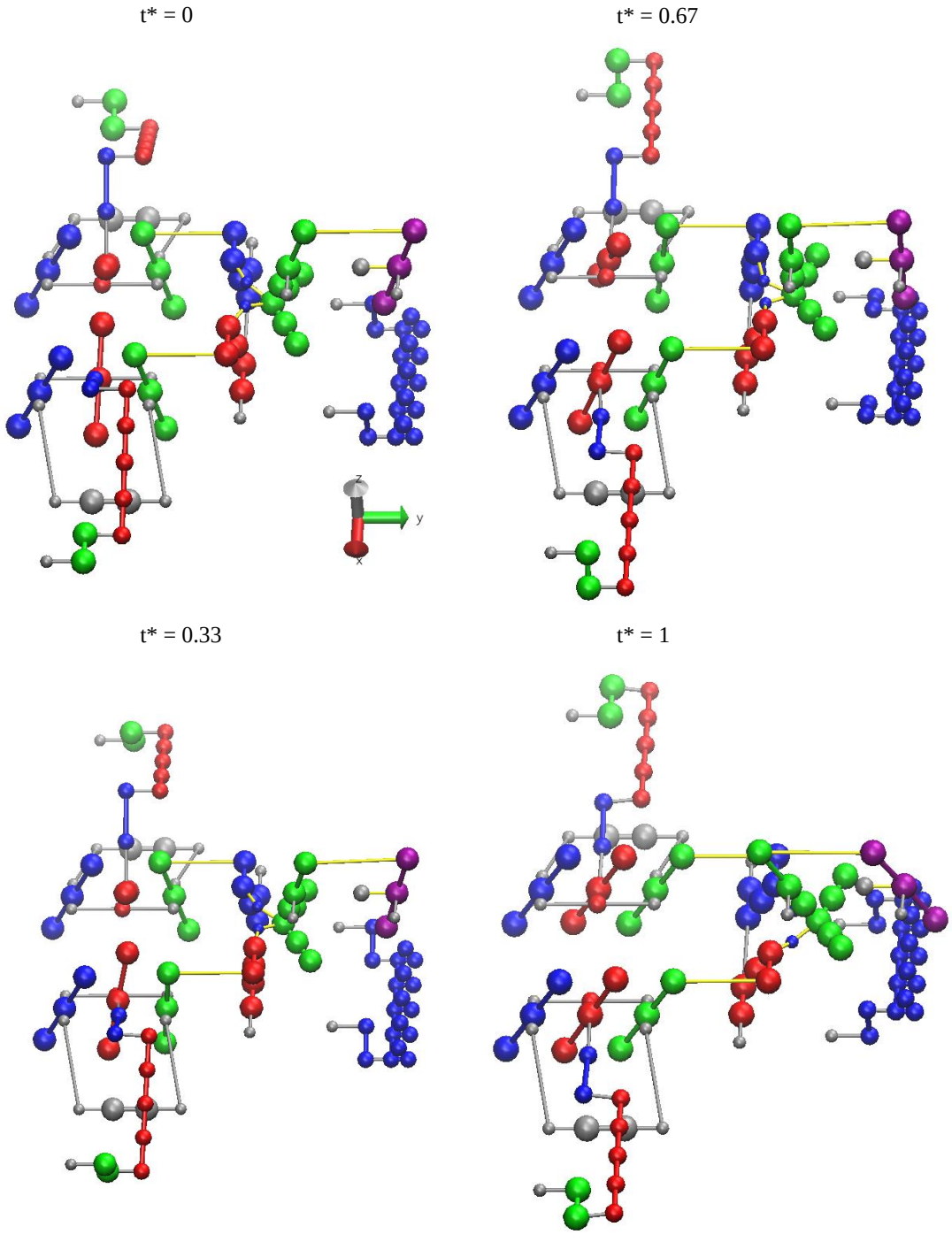


Figure 3.8: The NAND gate with inputs '11' connected to a pair of drivers being driven from a disengaged to an engaged (signal-boosted) state. The full process in the absence of thermal noise may be viewed in Supplemental Video C.

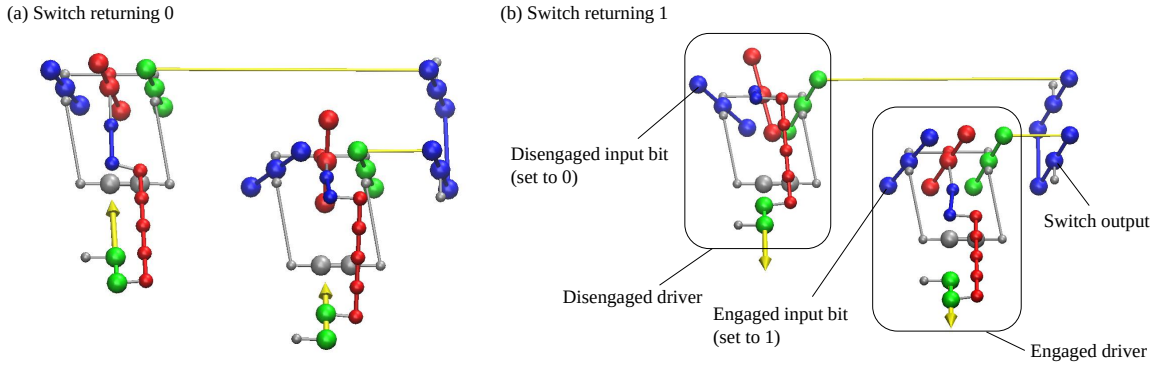


Figure 3.9: The switch design used to selectively transmit information from a given bit depending on the phase of the external dipole. The switch in (a) is currently in the '0' state and will transition to the '1' state if the external dipole advances by 180°; the reverse is true for the switch in (b).

3.3 The Switch

The switch provides a means of selectively relaying information to an output from two different input bits via a change in the phase of the external dipole without relying on a multiplexer. It consists of two drivers with a phase offset of 180° such that the internal dipoles of the two drivers point in opposite directions, tethered to a common switch output. By altering the phase of the external dipole, one of the input bits of the drivers can be selectively expressed at the output (Fig. 3.9). Fig. 3.10 illustrates the process of the switch transitioning from the '1' state to the '0' state.

By connecting two pairs of switches to the NAND gate, the inputs of the NAND gate can be selectively varied with the phase of the external dipole (and hence, the clock). This allows for transitions between arbitrary pairs of states.

3.4 Circuit construction

Several methods of chaining individual logic gates into useful combinatorial circuits are possible. The simplest method is to directly tether the output of one gate to the input of the other. Direct tethering can also be used to fulfil the role of an

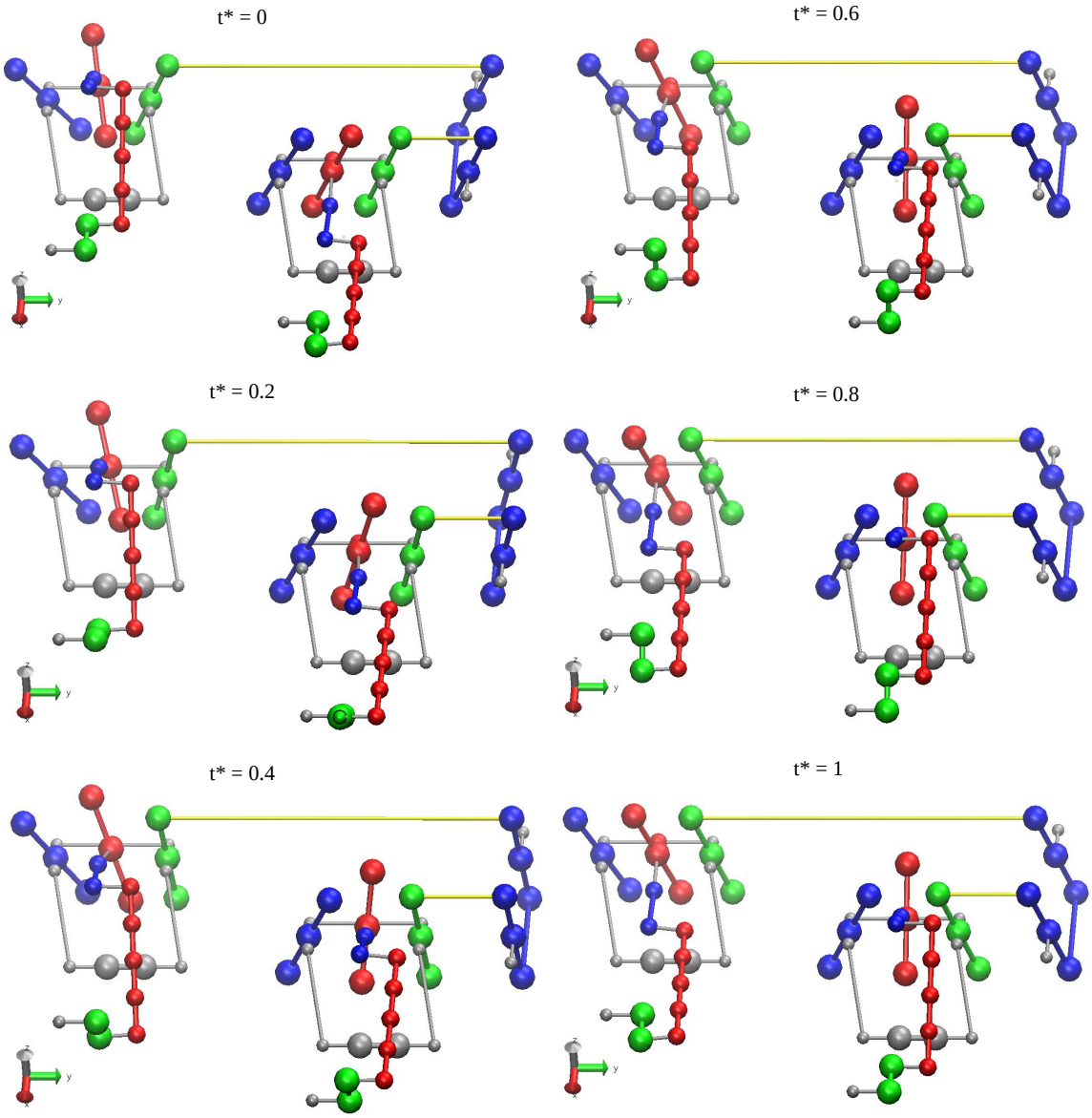


Figure 3.10: The switch transitioning from the '1' state to the '0' state in the absence of thermal noise. The full process may be viewed in Supplemental Video D.

inverter. Although the inverter can be constructed from a NAND gate by feeding the inverter input into both inputs of the NAND gate, this is generally an inefficient use of resources. It is more efficient to implement an inverter within this particular molecular mechanical logic system by tethering the output of one gate to another in a manner such that an increase in the slope of the output bit results in a decrease in the slope of the input bit, and vice versa (Fig. 3.11).

The simple inverting connection does suffer from the disadvantage that the downstream gate or bit is not in the same plane, being displaced in the z-axis. When retaining both gates in the same plane is strongly desired for engineering reasons; for instance, in the half-adder circuit (Ch. 7), an alternative design where both upstream and downstream elements remain in the same plane can be used (Fig. 3.12). By tethering the top of the upstream bit to the bottom of the downstream bit, increasing the slope of the input bit decreases the slope of the output bit via a pulling motion. However, decreasing the slope in a similar manner results in a pushing motion, which is less efficient at force transmission. Mitigating this problem is possible if a second tether is attached from the bottom of the output bit to the top of the input bit, converting deflections in both directions into pulling motions.

Although direct tethering is the simplest way of constructing combinatorial circuits, it also constrains the relative geometry of the two gates, with only a single degree of freedom along the y-axis. A free geometric relationship between the two logic gates in the xy-plane is relatively simple to implement in principle by extending the rigid body which receives the input along the x-axis.

A free geometric relationship in all three axes is somewhat more complicated to achieve, as it requires the use of a bellcrank (Fig. 3.13), a mechanism for converting a displacement along one axis into displacement along another orthogonal axis. The ability to construct three-dimensional circuits remove AI safety papers the fundamental constraint that the circuit graph be planar which afflicts two-dimensional circuits

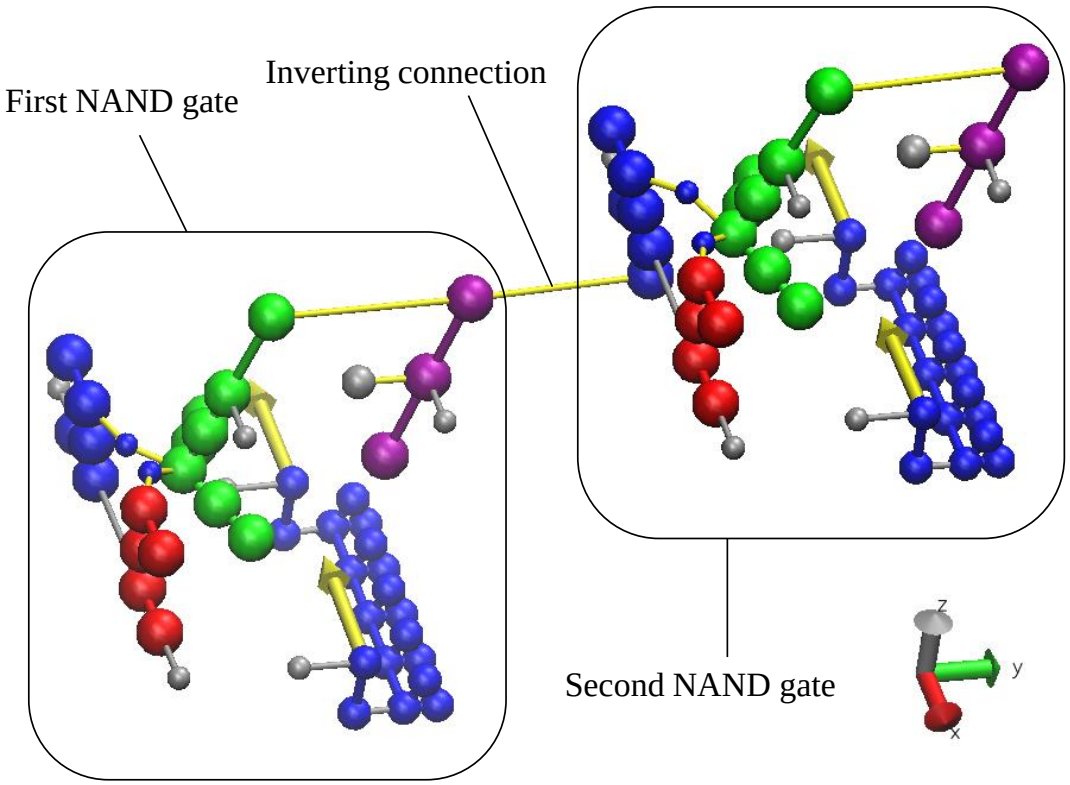


Figure 3.11: The output of one NAND gate fed into the input of another via an inverting connection.

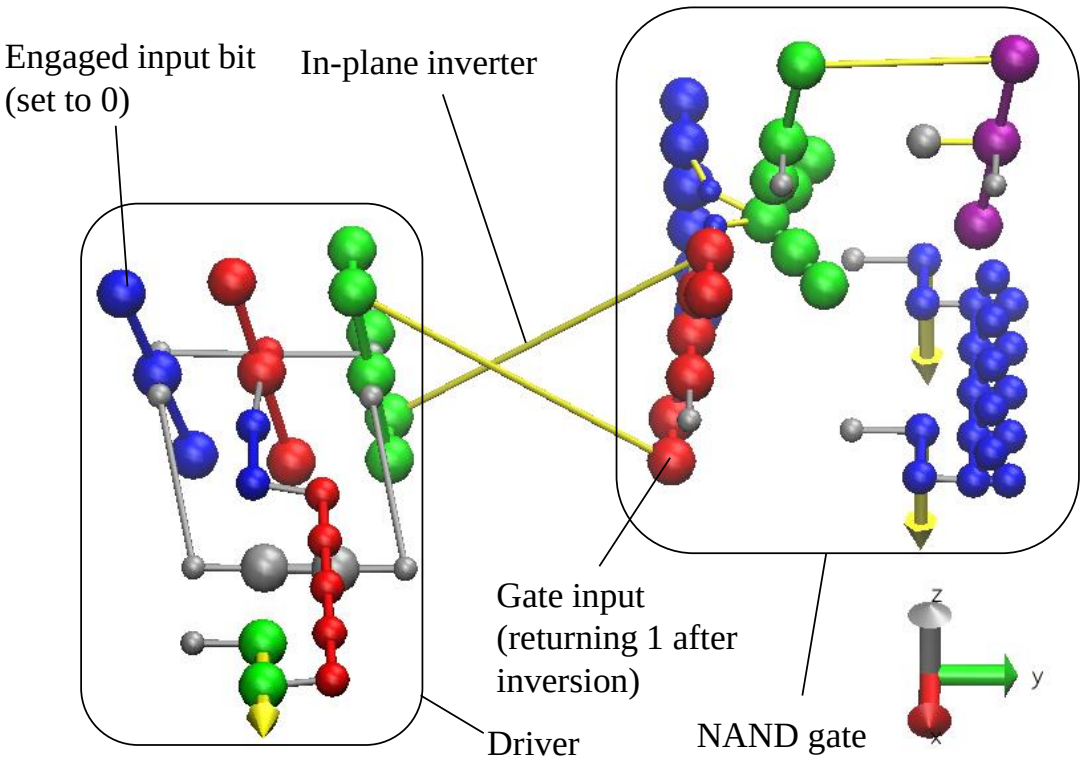


Figure 3.12: The output of a driver fed into the input of a NAND gate via an in-plane inverter.

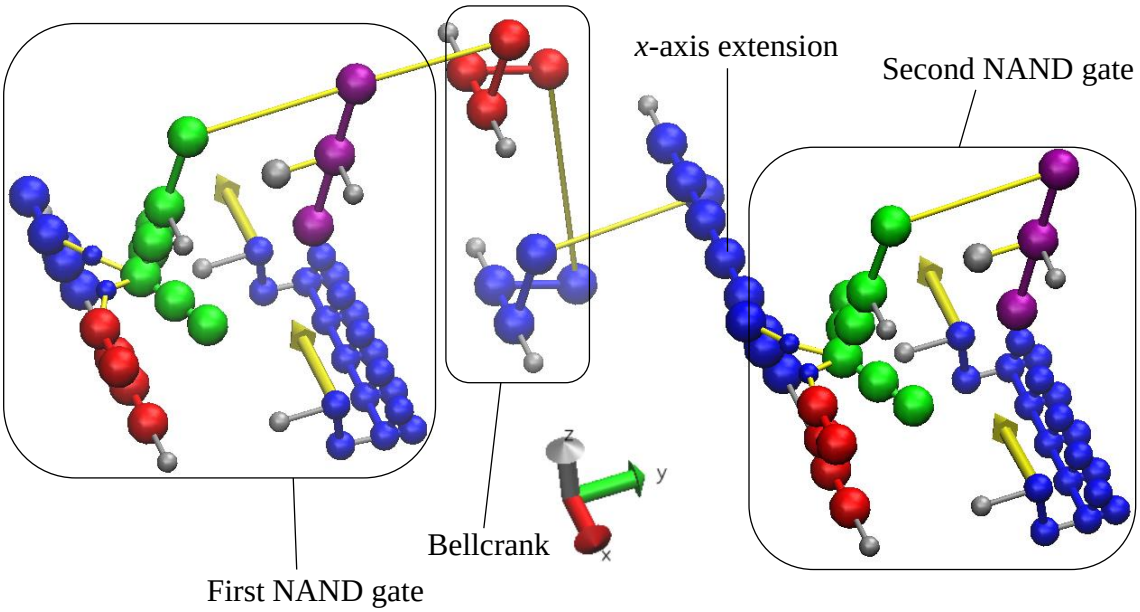


Figure 3.13: The NAND-inverter-NAND circuit from Fig. 3.11 further extended in the y and z-axes by the use of a bellcrank.

such as most CMOS-based designs. It should be noted that three-dimensional CMOS-based integrated circuits do exist; however, their practical use is generally limited to low-power applications due to heat dissipation issues [73]. The far greater theoretical efficiency of molecular mechanical logic gates would allow them to circumvent this limitation, potentially favouring three-dimensional molecular-mechanical logic circuits over two-dimensional circuits.

Chapter 4

Demonstrating Landauer's Principle and the Szilard Engine

Before any simulations of more complex systems were executed, I decided to first design a system capable of demonstrating Landauer's principle and the Szilard engine using the driver from Section 3.1, and subsequently verify Landauer's principle by running simulations using Rigor Mortis. Other than demonstrating a fundamental physical principle, this set of simulations also serves to prove that the Langevin thermostat of Rigor Mortis functions in a reasonable manner under the given conditions, and that persistent storage of information in a molecular mechanical system is at least theoretically possible.

4.1 The double bit

An information storage system based on a simplified version of the driver outlined in Section 3.1 was designed for this purpose. It consists of two drivers coupled to a single storage bit (Fig. 4.1), though only one of the drivers responds to the movement of the external dipole. The driver design used is less complex than the one discussed in Section 3.1, lacking the scaffold to force the driver arms to travel in only one direction;

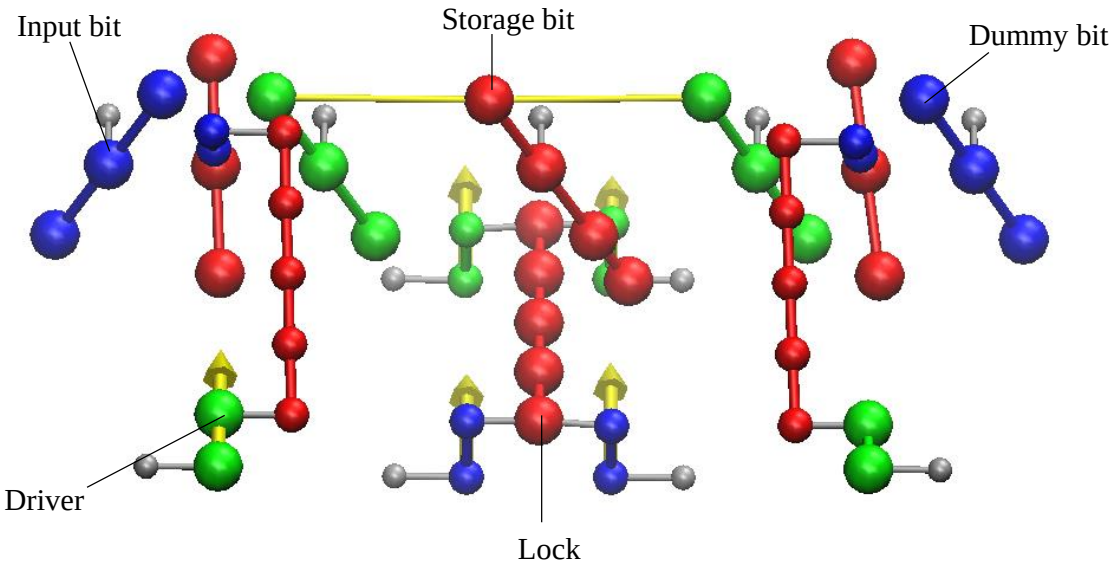


Figure 4.1: The driver-bit system used to demonstrate the Landauer bound and molecular Szilard engine with input bit initialised to '1' and storage bit to '0' at $t^* = 0$ where t^* is the current time fraction of the total time required to complete the full process.

this was possible because there was no significant deviation from the most favourable path for the drivers of this particular system (a phenomenon that did not apply to the other systems simulated). The storage bit receives information from the input bit via the left-handed driver only. The presence of the seemingly redundant right-handed driver ensures that the initial and final states in the case where the storage bit is overwritten are mirror images of each other. Therefore, the energy of the initial and final states are identical, a fact which greatly simplifies the task of demonstrating the Landauer limit.

Another significant component of the double-bit system is the lock. This is a rod which when engaged acts to restrict the motion of the storage bit, forcing it to remain in one of two possible states until the lock is disengaged. The motion of the lock is driven by a pair of synchronised internal dipoles. Despite superficially resembling the signal booster used to augment the signal of the winged NAND gate, the internal

dipoles which drive the lock do not rotate synchronously with the internal dipole of the left-handed driver. Therefore, lock engagement and disengagement is not a logically reversible operation. This logical irreversibility is a direct consequence of the fact that during the reverse process, the lock becomes engaged before the driver does, causing it to force the bit into a random state.

Figs. 4.2 and 4.3 demonstrate the process of writing information from the input bit to the storage bit from time $t^* = 0.25$ where t^* is the ratio of the current time t over the time required to complete the entire operation. The system is initialised with lock engaged and driver disengaged, with the internal dipole of both driver and lock $\hat{\mathbf{d}}_{\text{lock}} = \hat{\mathbf{d}}_{\text{driver}} = (-1, 0, 0)$. As the system moves from $t^* = 0$ to $t^* = 0.25$, $\hat{\mathbf{d}}_{\text{lock}}$ is rotated by 90° to $(0, 0, -1)$, causing the lock to become partially disengaged. Between $t^* = 0.25$ to $t^* = 0.5$ the internal dipole of the driver rotates by 180° to $(1, 0, 0)$, causing information to be transduced from the input bit to the storage bit. Simultaneously, the internal dipole of the lock rotates by 90° to $(1, 0, 0)$, though this has no effect on the storage bit since it is already free to move. Between $t^* = 0.5$ to $t^* = 0.75$ the internal dipole of the lock is rotated back to $(0, 0, -1)$, though as before this has no effect on the storage bit. Finally, between $t^* = 0.75$ to $t^* = 1$ the lock is engaged as its internal dipole is rotated back to its initial position, preventing the storage bit from moving freely. Simultaneously, the internal dipole of the driver is rotated back to its initial position, disengaging the input and storage bits.

Note that the dipoles of the driver and the lock are driven by separate external dipoles in this system, unlike the signal booster of the NAND gate, hence the reason they are out of phase. This process causes data in the storage bit to be overwritten, and is thus logically irreversible due to reducing the information entropy of the system. Therefore, in the absence of entropy production from any other sources, it is expected that the process will require $k_{\text{B}}T \ln 2$ of work to complete in the quasistatic limit. In addition to demonstrating the work required to overwrite a bit, the driver-bit system

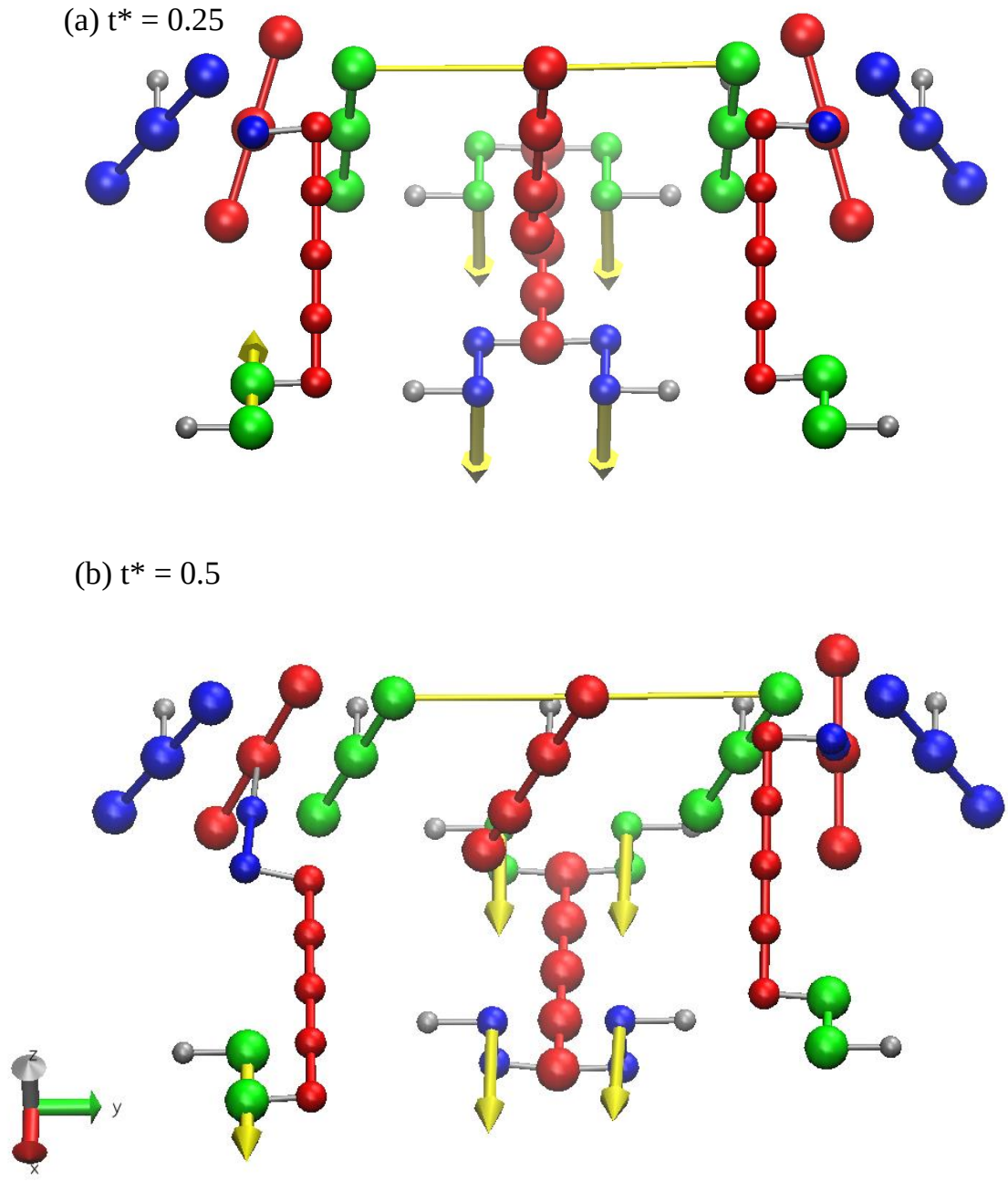
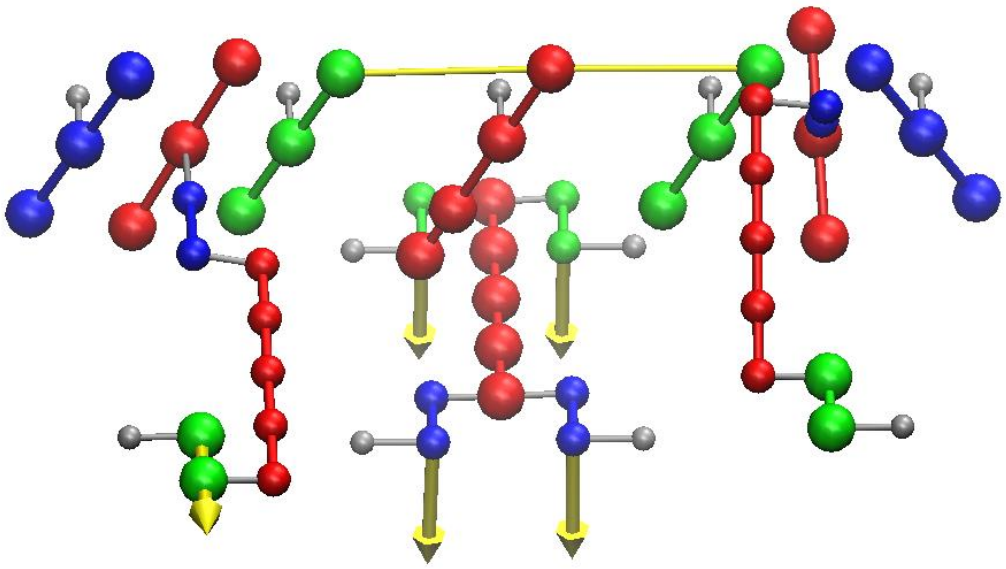


Figure 4.2: The driver-bit system between $t^* = 0.25$ and $t^* = 0.5$ during an overwriting process. Supplemental Video 2 provides an animation of the process described in this figure and Fig. 4.3

(c) t* = 0.75



(d) t* = 1

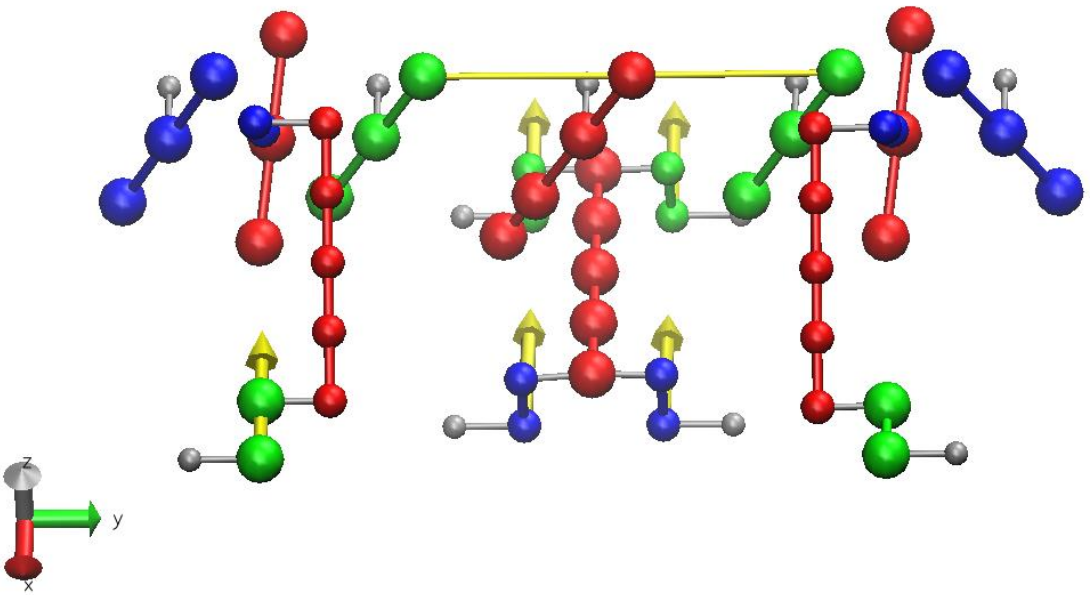


Figure 4.3: The driver-bit system between $t^* = 0.75$ and $t^* = 1$ during an overwriting process.

can also be used to extract work from a system whose state is known. This process resembles the second phase of the set-up of the thought experiment mentioned in Section 1.2.3, the Szilard engine, in which usable work is extracted from information. By reversing the process in which information is written into the storage bit, usable work may be extracted from the system at the cost of randomising the state of the storage bit due to the thermodynamically reversible transfer of the potential energy contained in the high-information (low-entropy) state into the external dipoles. Thus, the full process of overwriting a bit of randomised state by expending work followed by the recovery of of the work done at the cost of overwriting the bit is an example of a logically irreversible process being executed in a thermodynamically reversible manner.

4.2 Simulations on the double bit system

Two sets of simulations were carried out, corresponding to the situation where the input bit is initialised to '0' and '1'. The work done for both sets of simulations was averaged together and the results plotted in Fig. 4.4. From the graph, it is clear that reducing the clock speed causes the work done to converge to $k_{\mathrm{B}}T\ln 2$ at both room temperature ($T_r = 298\,\mathrm{K}$) and 596 K. In addition, the work recovered by the reverse process also converges to $k_{\mathrm{B}}T\ln 2$ as the clock speed is reduced, indicating that the system is capable of recovering the free energy contained in high-information states.

If the system would converge toward thermodynamic reversibility in the quasistatic limit, the primary source of dissipation would be the drag imposed by the Langevin thermostat. This drag is proportional to velocity; thus, the work that must be consumed to overcome it is directly proportional to the clock speed. We observe that this is indeed the case in Fig. 4.5.

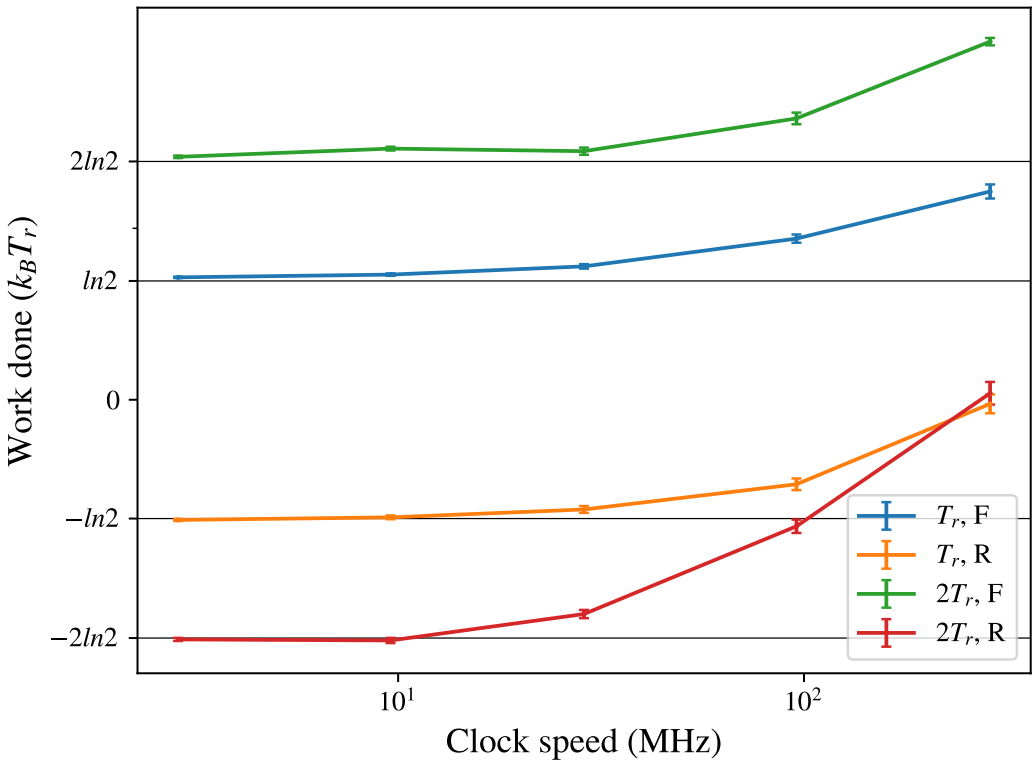


Figure 4.4: Work done against clock speed at two different temperatures for the forward and reverse bit-writing processes of the double-bit system. The blue and orange lines correspond to the forward (F) and reverse (R) simulations carried out at T_r (298 K), respectively, while the green and red lines correspond to those carried out at $2T_r$ (596 K). Each data point corresponds to an average over 240 simulations.

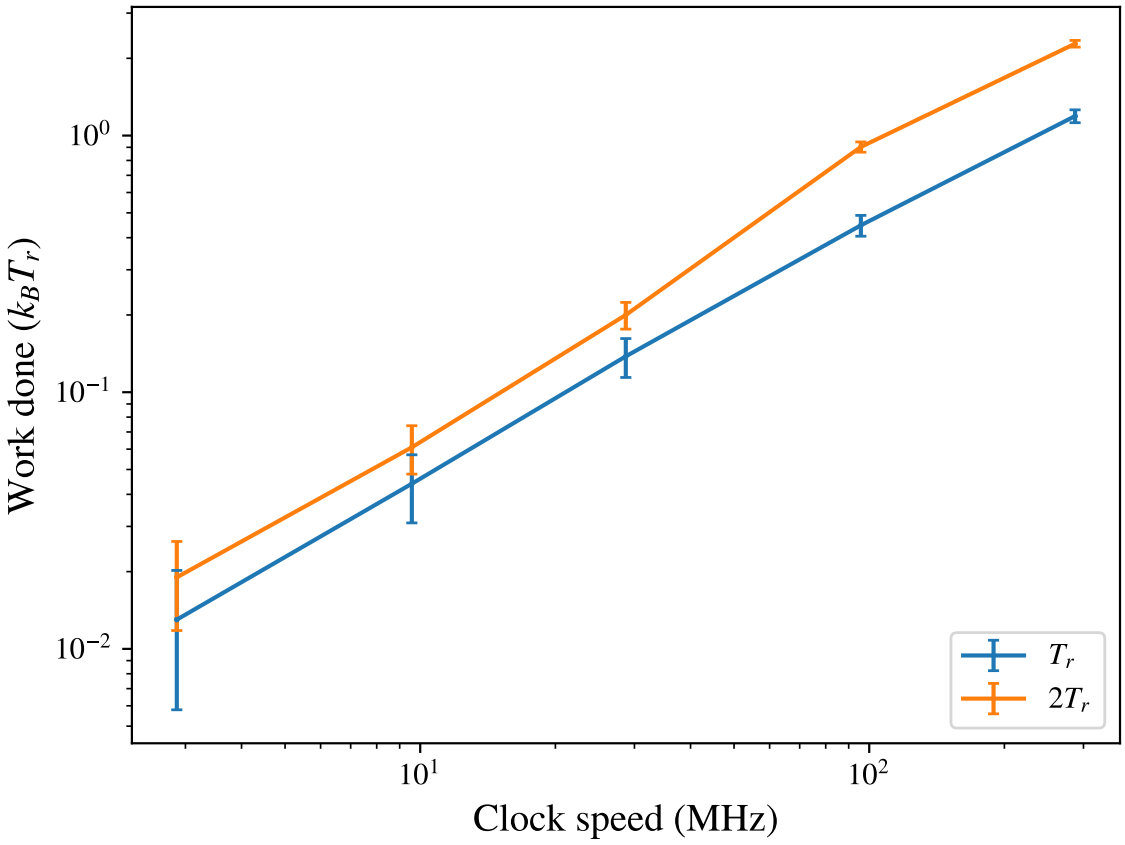


Figure 4.5: Sum of work done against clock speed at two different temperatures over both the forward and reverse bit-writing processes of the double-bit system.

Chapter 5

Simulations on the winged NAND gate

The most important characteristic for a practical logic gate is arguably its error rate. In the context of designing energy-efficient reversible logic gates, the thermodynamic efficiency as a function of clock speed is also an important property to characterise. In this chapter, I outline simulations demonstrating the winged NAND gate's ability to function with a low error rate even at elevated temperatures and in a thermodynamically reversible manner across all possible sets of transitions between pairs of inputs.

5.1 Transitions from a neutral to a defined state

In order to determine the viability of the winged NAND gate design, reversibility and error rate simulations were run on a system consisting of a single NAND gate with each of its inputs linked to a driver constrained to a particular Boolean value. This set-up is identical to the one illustrated in Fig. 3.5. In these simulations, the system was initialised with both drivers at a point of minimum engagement. Before each simulation was run, the system was driven over one full cycle and then re-equilibrated to

ensure that the starting state was as close to thermodynamic equilibrium as possible. The drivers were then rotated by the external dipole to the point of maximum engagement at $t^* = 0.5$. The system was then returned to the initial state by a reverse rotation of the external dipole at the same rate; the full process constitutes one clock cycle. Animations of the NAND gate in the '00', '01' and '11' states driven via the process described above can be viewed in Supplemental Videos 3-5.

Due to the lack of potential energy barriers in the disengaged (neutral) state, it can be assumed that the free energy of the initial and final states are practically equal. The error rate was calculated by taking the ratio of simulations where the output bit returned an incorrect value at the point of maximal driver engagement to the total number of simulations carried out. For the definition of maximal driver engagement, please see Sections 3.1 and 3.2. 120 000 error-rate simulations were run at 1.44 GHz and at temperatures of 1, 2, 4 and $6T_r$; of these, only the simulations run at $6\,T_r$ had non-zero error rates. The results of these simulations are summarised in Table 5.1. The error rate was below 8.33×10^{-6} at 1, 2 and $4T_r$, and still relatively low at $6T_r$. If it is assumed that the probability of the system entering an erroneous state follows a Boltzmann distribution, the error rate at 298 K would be far below 10^{-20}; thus, it is almost certain that the error rate at 298 K would be sufficiently low so as not to interfere with demonstrations of thermodynamic reversibility.

A second set of simulations was run to determine the thermodynamic reversibility of the design. The results are summarised in Fig. 5.1. As with the double bit system, the primary source of entropy production should be the drag imposed by the Langevin thermostat if the system would converge towards thermodynamic reversibility in the quasistatic limit, and the work done by the drag term should decrease linearly with clock speed. If this were not the case, we would observe a situation similar to that summarised in Fig. 4.4, where the work done tapers to a nonzero value as the clock speed is decreased. As this does not occur and the work done by the external dipole

Table 5.1: Results obtained from error-rate simulations of the NAND gate performed at $6T_r$ (1788 K) for the evolution from a neutral state to a given target state at a clock speed of 1.44 GHz. No errors were detected over 120 000 simulations at $T \leq 4T_r$.

State	Error rate
00	3.33×10^{-5}
01	2.17×10^{-4}
10	5.83×10^{-5}
11	5.00×10^{-5}

is roughly proportional to the clock speed, it can be concluded that this system converges toward thermodynamic reversibility in the limit of clock speeds tested.

5.2 Switching between defined states

Switching between defined states (states in which the bits are constrained in place due to driver engagement) and monitoring the error rate of both the initial and final states of the system after a round-trip process provides an even stronger guarantee that the initial and final states are identical in free energy, so long as the error rate remains extremely low at the desired operating temperature. In order to carry out these simulations, the set-up in Fig. 5.2, consisting of two pairs of switches coupled to a NAND gate, was used to drive the NAND gate between all twelve possible pairs of inputs and outputs.

In this protocol, the system is initialised with the drivers of the switch engaged towards the initial state and the signal booster of the NAND gate fully engaged. The starting error in Table 5.2 corresponds to the error rate of this initial state (measured after equilibration); thus, if it is assumed that the system is transitioning from '11' to '00' as in Fig. 5.3, this corresponds to the error rate of the NAND gate at '11'. The drivers of the switch rotate in the same direction, but at half the rate of the signal booster of the NAND gate.

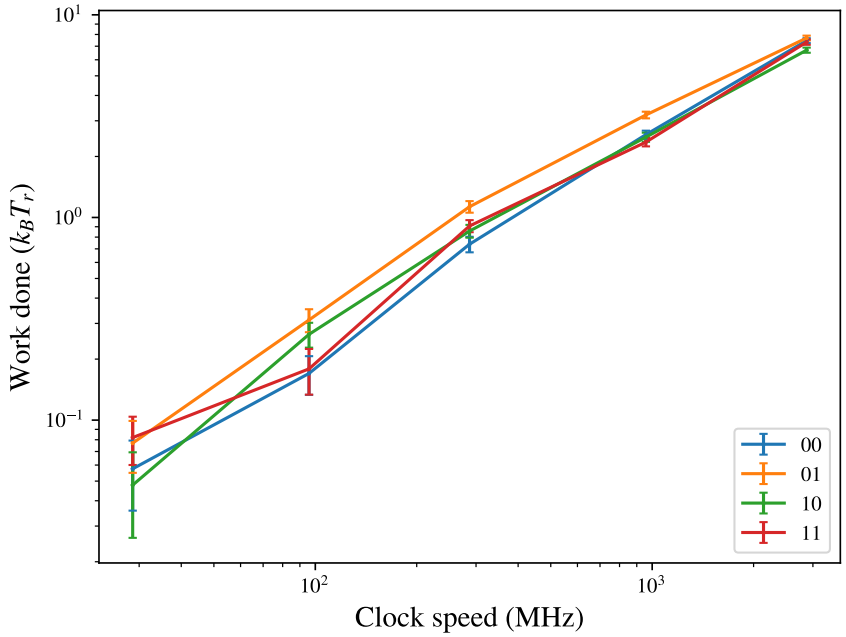


Figure 5.1: Work done against clock speed for the NAND gate during thermodynamic reversibility simulations from a neutral state to the state to which the input bits are set. Each data point corresponds to an average over 120 simulations.

Between $t^* = 0$ and $t^* = 0.25$, the signal booster of the NAND gate is rotated by 180° to become fully disengaged while the switch drivers rotate by 90°, leaving the input bits in an indeterminate state. Between $t^* = 0.25$ and $t^* = 0.5$, the signal booster is rotated in the same direction by another 180°, becoming re-engaged; simultaneously, the switch drivers rotate by another 90° causing the drivers of the switch to engage the second set of bits. It is at this point that the midpoint error is measured; this corresponds to the error rate of the '00' state in Fig. 5.3. This process is then reversed, with the endpoint error corresponding to the error rate of the '11' state for this system after it has been returned to its initial configuration at $t^* = 1$. Supplemental Video 6 provides an animation of such a process. As with the neutral transition simulations, the energy-minimised system was driven over one full cycle and then re-equilibrated before any observables were measured.

The results of reversibility and error rate simulations, recorded in Fig. 5.3 and Ta-

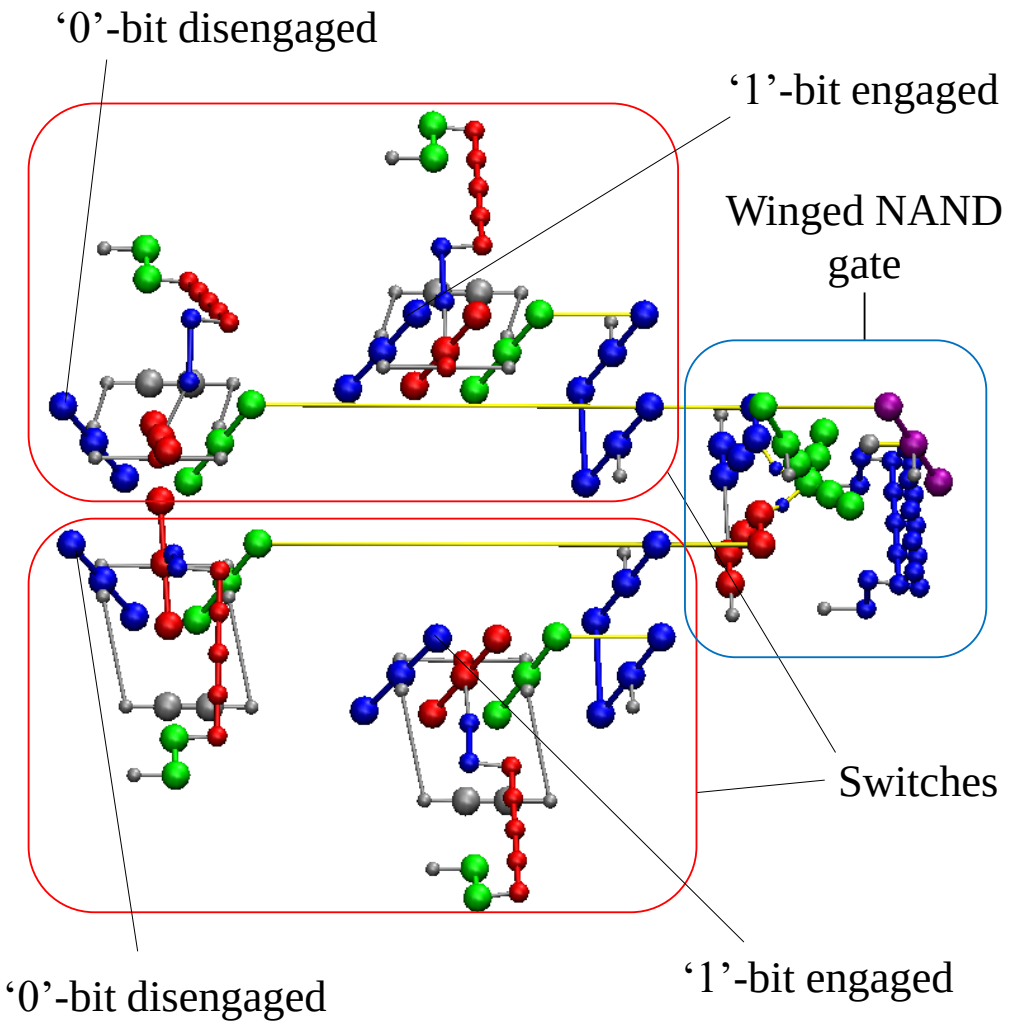


Figure 5.2: Two pairs of switches (red outline) connected to a winged NAND gate (blue outline).

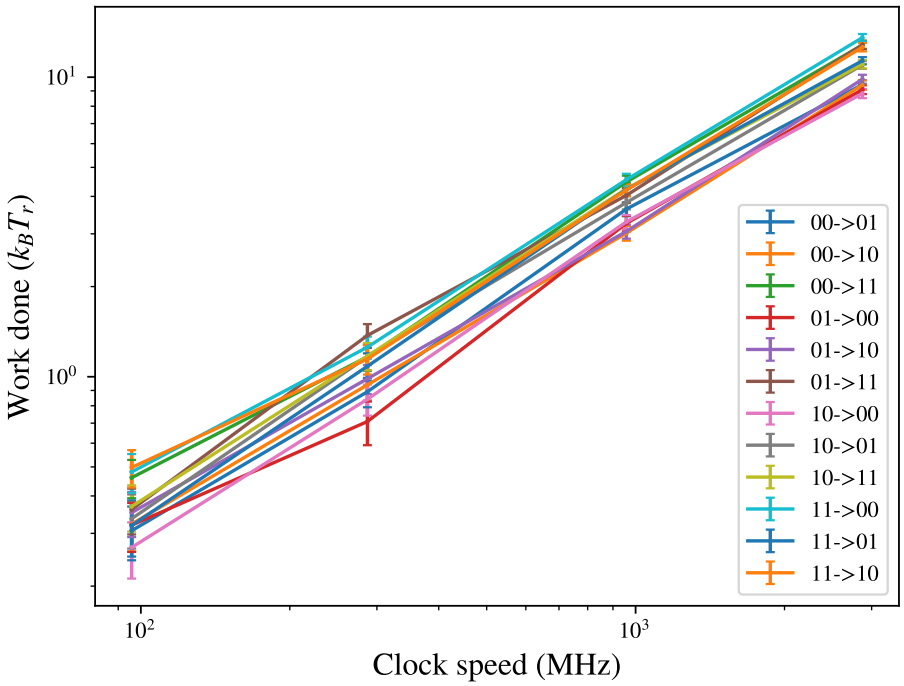


Figure 5.3: Work done against clock speed for the NAND gate during thermodynamic reversibility simulations of transitions from one defined state to another. Each data point corresponds to an average over 120 simulations.

ble 5.2 respectively, show that the system tends towards thermodynamic reversibility across all possible unique transitions in the limit of clock speeds tested and has a reasonably low error rate even at $4T_r$, sufficiently low that it is fair to assume that the initial states of all transitions at 298 K are in equilibrium.

Table 5.2: Results obtained from error-rate simulations of the NAND gate performed at $4T_r$ (1192 K) for the evolution from a defined starting state to a given target state at a clock speed of 1.44 GHz. No errors were detected over 60 000 simulations at $T \leq 3T_r$.

Transition	Error (Start)	Error (Mid)	Error (End)
00→01	1.667×10^{-5}	5.000×10^{-5}	1.667×10^{-5}
00→10	0	3.333×10^{-5}	0
00→11	0	0	0
01→00	0	0	0
01→10	3.333×10^{-5}	5.000×10^{-5}	6.667×10^{-5}
01→11	1.500×10^{-4}	3.333×10^{-4}	1.167×10^{-4}
10→00	0	0	3.33×10^{-5}
10→01	6.667×10^{-5}	6.667×10^{-5}	6.667×10^{-5}
10→11	1.000×10^{-4}	0	1.000×10^{-4}
11→00	0	0	0
11→01	3.167×10^{-4}	5.000×10^{-4}	3.333×10^{-4}
11→10	0	1.000×10^{-4}	0

Chapter 6

Chaining Gates

A lone NAND gate is not particularly useful regardless of its error rate or efficiency; logic gates must be connected into circuits to perform computationally useful tasks. Chaining molecular mechanical gates together such that they deliver the desired output with a low error rate and in a thermodynamically efficient manner is non-trivial. In this chapter, I shall compare two methods by which molecular mechanical logic gates may be chained together to form a circuit, and show that while directly chaining gates together is not practical, phased chaining allows for the construction of thermodynamically reversible combinatorial circuits with a low error rate.

6.1 Direct chaining

The simplest way to chain two gates together is to directly feed the output of one into the input of another, a process henceforth referred to as direct chaining. Despite its simplicity, directly chaining input-preserving molecular mechanical gates together to create logical circuits is not always practical due to the high error rate which may result.

In order to demonstrate the error rate of direct chaining, the circuit shown in Fig. 6.1 was used. This particular circuit can be easily extended by the addition of more

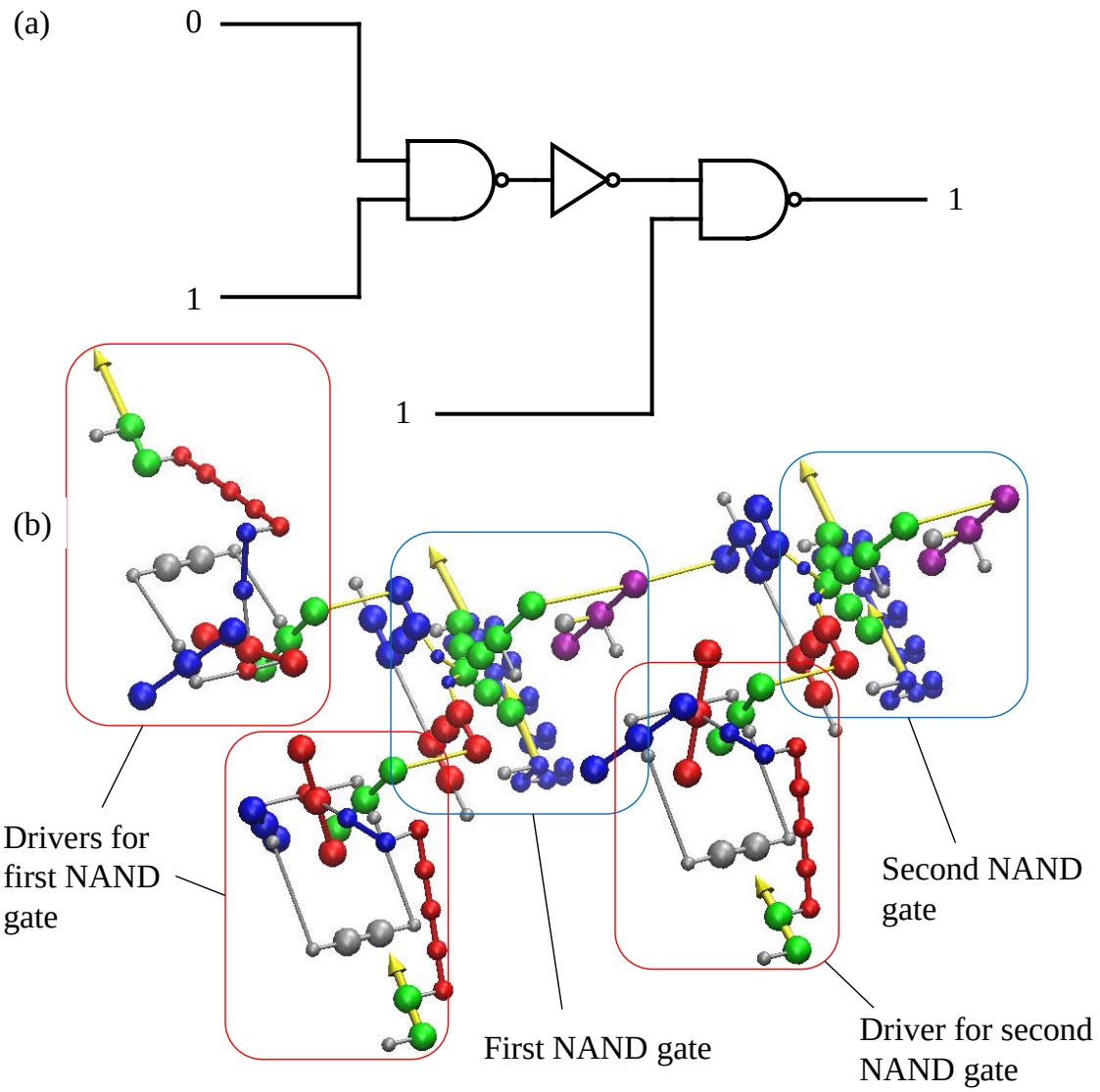


Figure 6.1: The circuit layout used to demonstrate direct gate chaining.

inverter-NAND units since each NAND gate receives an input of '01' and outputs '1', which is inverted and passed along to the next NAND gate in line.

Only one set of error-rate simulations were run for the direct-chaining system. In these simulations, the system was initialised with all drivers and signal boosters fully disengaged. As all drivers and signal boosters are by definition in phase, they simultaneously become fully engaged at $t^* = 0.5$, at which point the error rate of the final gate in the chain is measured. The process is then reversed.

Error rates of 4.6% and 12.9% were obtained for directly-chained systems of two and three gates, respectively. The error rates, measured at room temperature, are far higher than that for a single NAND gate, and demonstrate the predicted problems of direct chaining. No thermodynamic reversibility simulations were attempted due to the high error rate.

I propose two major causes for the large increase in error rate as a result of directly chaining two gates together. Firstly, the spring constant of a number of identical simple harmonic oscillators in series is inversely proportional to the number of oscillators linked together. Similarly, a long series of chained gates becomes increasingly less able to effectively transmit information from the first gate to the last without errors caused by thermal fluctuations. Secondly, due to the finite rigidity of the system, there is insufficient time for the information from the first gate in the chain to reach subsequent gates in the chain before the associated signal boosters is activated (the activation occurring simultaneously with that of the first gate), causing the signal booster to occasionally force the output of the second gate into an erroneous position.

6.2 Phased chaining

A more sophisticated approach is required to ensure acceptable error rates and reversibility when chaining input-preserving gates together. One method of reducing

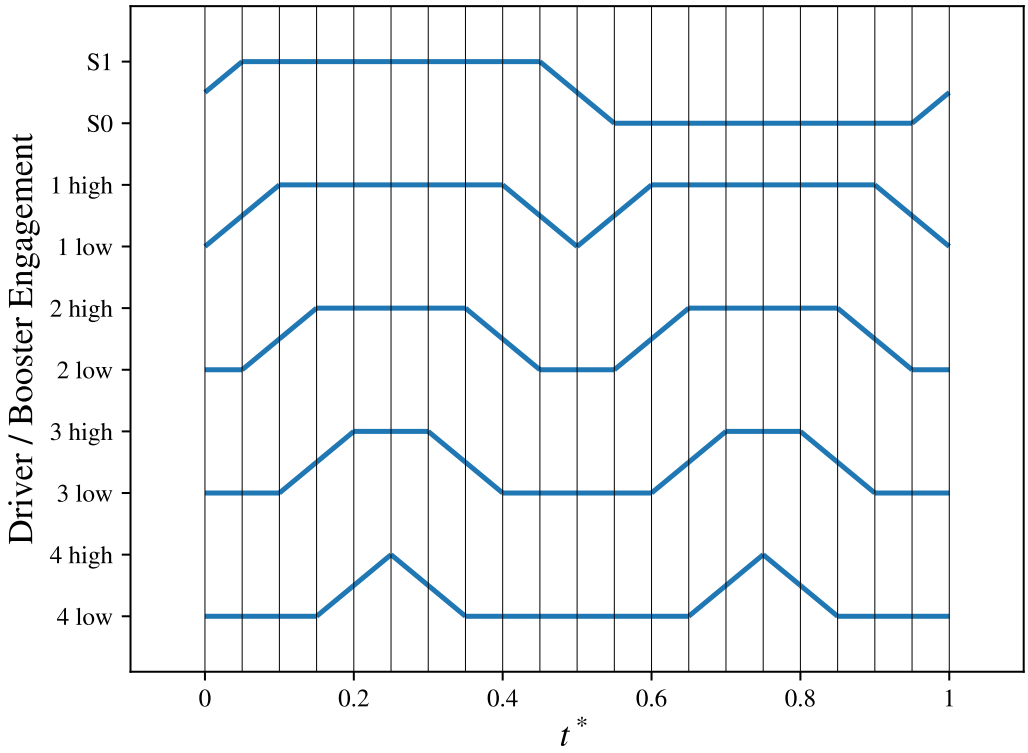


Figure 6.2: Illustration of the phased chaining protocol for a system containing four gates. The n-high/low labels on the y-axis indicate whether the drivers or signal boosters of the nth gate are engaged. S0 and S1 indicate the state of the switches when fully toggled toward '0' and '1' respectively. Each gate is engaged in turn and only disengaged after all succeeding gates have already been disengaged.

error rate is to boost the signal at each gate while staggering the boosts so they are out of phase as shown in Fig. 6.2. In addition, a driver is placed between the output of an upstream gate and the input of the downstream one, allowing the two to be physically separated until the upstream gate's signal has been fully boosted. This protocol should, in principle, allow each gate to be treated as an independent unit.

This protocol does require individual control over each distinct level of logic gates, making it impossible to control with a single external dipole. In addition, the rotation associated with each level must be halted at the point of maximal engagement; this precludes the use of a continuously rotating dipole. This limitation is not necessarily

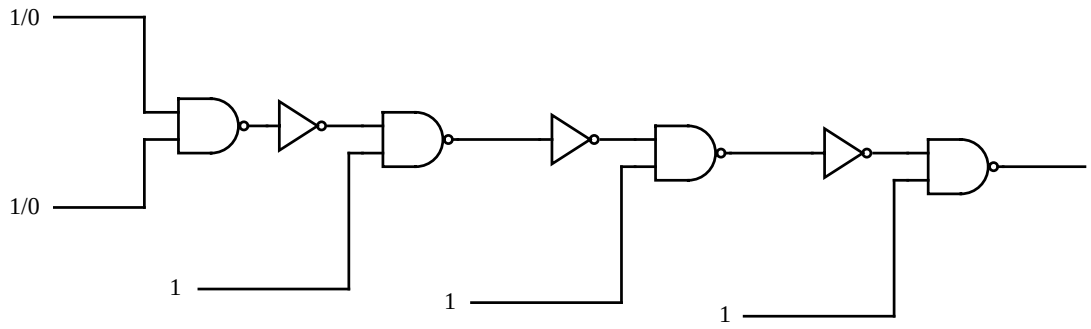


Figure 6.3: A circuit used to demonstrate phased chaining. This particular circuit has a chain length of 4 and can be extended indefinitely by the addition of more inverter-NAND units with the output of each NAND gate switching from '0' to '1' as the inputs of the starting NAND gate are switched from '11' to '00'.

less realistic; extant molecular motors, both biological (e.g. the mitochondrial proton pump [74]) and artificial [75] generally rotate in discrete steps as opposed to supplying a continuous torque.

The circuit layout in Fig. 6.3 was implemented to demonstrate this protocol, as it, like the circuit in Fig. 6.1 has the advantage of allowing additional inverter-NAND units to be added to extend it indefinitely. The inputs of the first NAND gate in the chain is initialised to the state '11', and is shifted to the state '00' at the midpoint of the simulation by the switch; this process is then reversed to return the inputs to '11'.

To illustrate the phased chaining operation, I will use the simplest example with only two NAND gates. The protocol proceeds in two phases, the first in which the switches return an output of '1', and the second where the switches return an output of '0'. The system is initialised with the switch in the midpoint position (biased toward neither bit) and every gate and driver maximally disengaged at $t^* = 0$. As soon as the first phase is initiated, the dipoles of the switches and the signal booster of the first NAND gate begin to rotate, such that at $t^* = 1/12$, both switches now return '1'. At this point, the switch is halted. The dipole of the signal booster of the first NAND gate continues to rotate until it reaches maximal engagement at

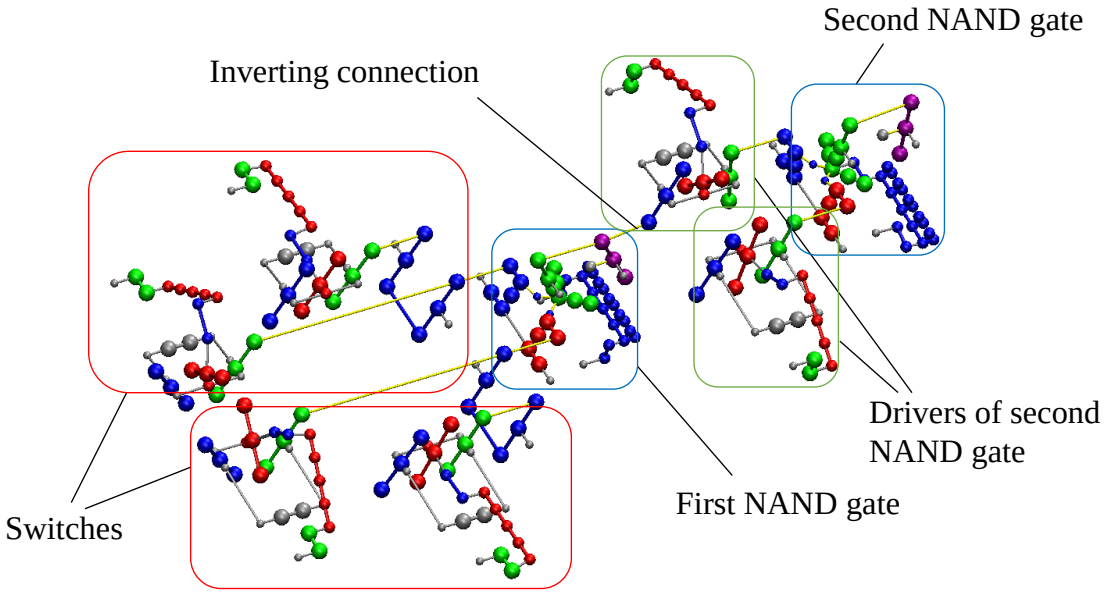


Figure 6.4: A phased-chained system consisting of one initial NAND gate and one inverter-NAND gate unit (chain length 2).

$t^* = 1/6$. Thus, the first NAND gate now outputs '0'. It is at this point that the first-phase error rate of gate 1 is measured. The driver and the signal booster of the second NAND gate begin and end their rotation in phase at $t^* = 1/12$ and $t^* = 1/4$ respectively, returning an output of '0'. It is at this point that the first-phase error rate of gate 2 is measured. Between $t^* = 1/4$ and $t^* = 1/2$, this process is reversed, returning the system to its initial state. The second phase is initiated at this point. It is identical to the first phase except that the switches are rotated in the opposite direction such that they return '0'. The second-phase error of gate 1 is thus measured at $t^* = 2/3$ and that of gate 2 at $t^* = 3/4$.

A similar protocol applies for chains containing three or more gates, with the nth NAND gate reaching maximal engagement in the first phase at $t^* = (2n-1)/4(x+1)$ where x is the number of gates in the chain, and those in the second phase at $t^* = (2x+2n+1)/4(x+1)$. It is at these points that the error rate is measured. A chain with n gates requires $n+1$ independent external dipoles to drive it. One clock cycle

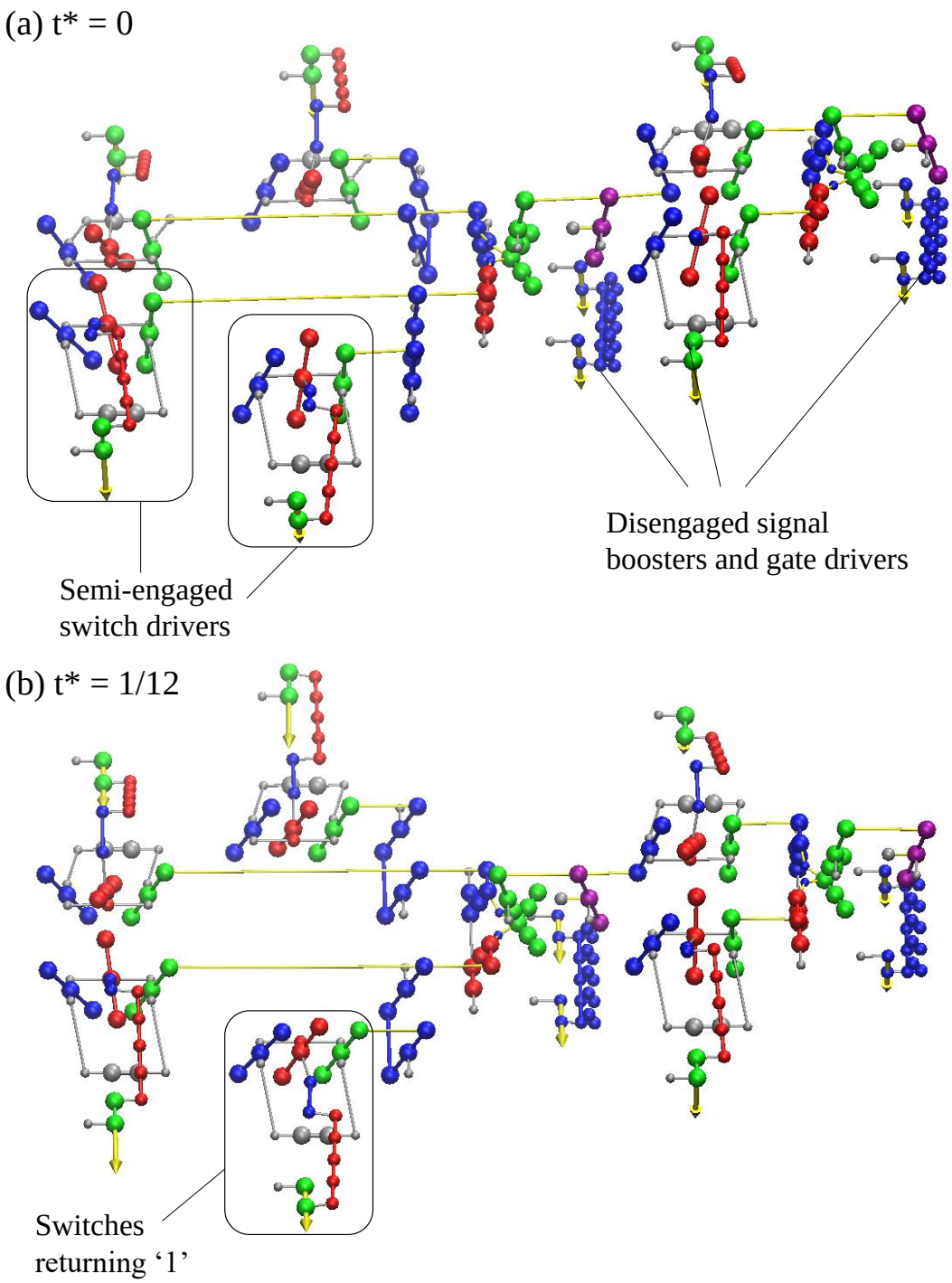


Figure 6.5: The two-gate phase-chained system at $t^* = 0$ and $t^* = 1/12$.

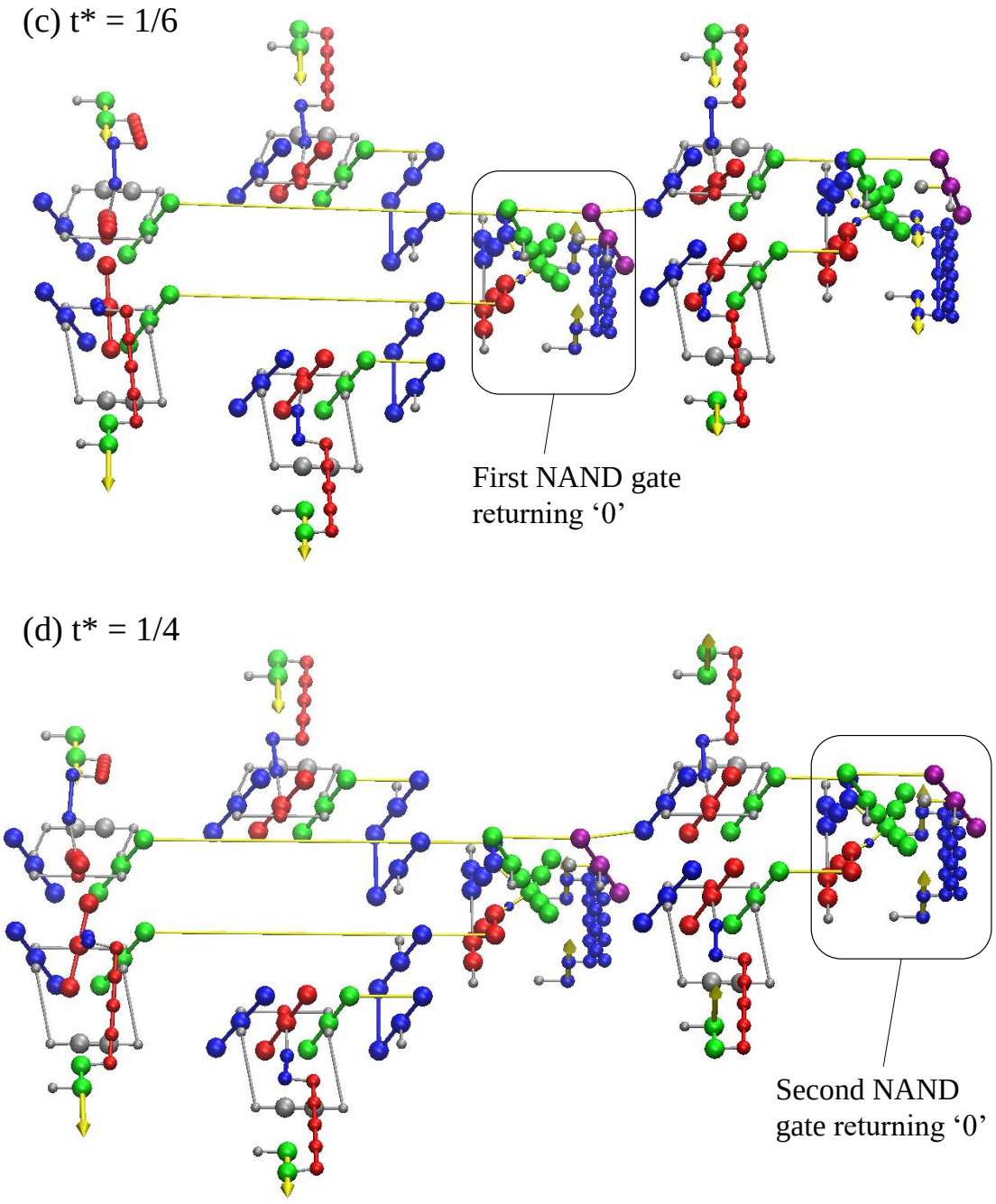


Figure 6.6: The two-gate phase-chained system at $t^* = 1/6$ and $t^* = 1/4$.

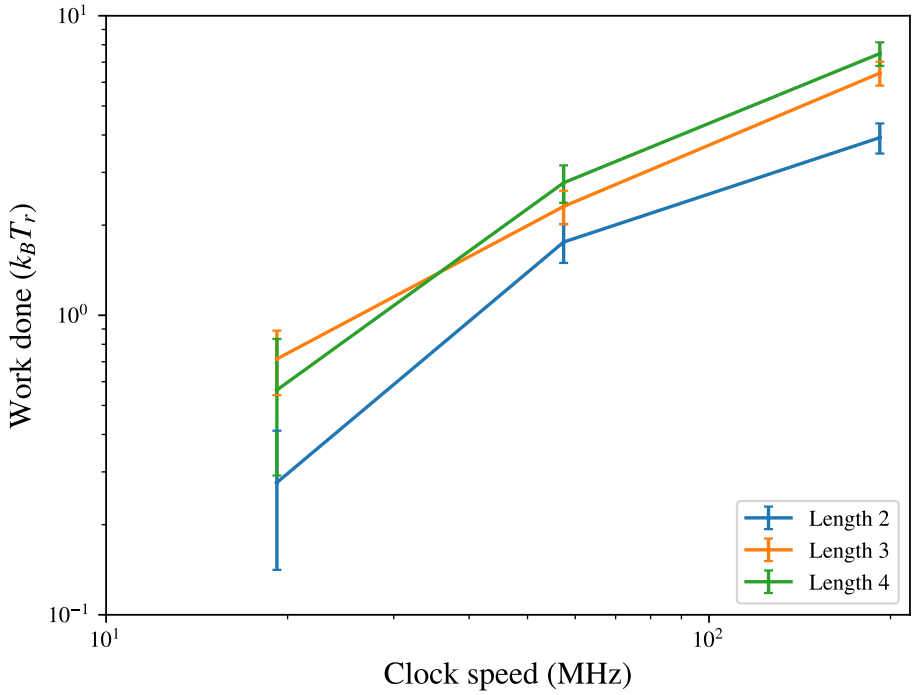


Figure 6.7: Plot of work done against clock speed for different chain lengths in phased chaining. Each data point corresponds to an average over 120 simulations.

is defined as $t/2(x+1)$, that is, twice the time needed for the signal booster of one gate to rotate from the disengaged to engaged position; this corresponds well with the clock cycle definitions of earlier simulations. Animations of the phased-chained systems of lengths 2, 3 and 4 can be viewed in Supplemental Videos 7, 8 and 9, respectively.

The results of the reversibility and error rate simulations are summarised in Fig. 6.7 and Table 6.1. The reversibility simulations show an approximately linear relationship between work done and clock speed, while the error rate remains low even at $6T_r$. Overall, it appears reasonable to conclude that this method of chaining input-preserving NAND gates is tending towards thermodynamic reversibility in the limit of clock speeds tested.

Table 6.1: Results obtained from error-rate simulations of four phase-chained NAND gates at $6T_r$ (1788 K) and 2.87 GHz. No errors were detected over 60 000 simulations at $T \leq 3T_r$.

Gate number	1^{st} phase error	2^{nd} phase error
1	5.90×10^{-3}	6.00×10^{-4}
2	1.00×10^{-3}	1.40×10^{-3}
3	6.67×10^{-4}	5.20×10^{-3}
4	5.67×10^{-4}	5.43×10^{-3}

Chapter 7

The Half-Adder

To demonstrate the ability of phased chaining to transmit information through complex combinatorial circuits, I designed a half-adder circuit consisting of three phase-chained NAND gates. The half-adder is a combinatorial circuit which receives two binary inputs and returns the sum and carry of the inputs (see Table 7.1 for the truth table of the half-adder gate). The sum is equivalent to the Boolean XOR operation performed on the inputs, and the carry equivalent to the Boolean AND operation performed on the inputs. Multiple half-adders can be chained together to create a circuit capable of adding two numbers of arbitrary size together; thus, the half-adder is a critically important combinatorial circuit for any digital computer design.

Two half-adders may be combined to create a full adder by feeding the carry bit of the first half-adder into the first output bit of the second. The full adder is

Input 1	Input 2	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table 7.1: Boolean truth table of the half-adder gate.

Input 1	Input 2	Input 3	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Table 7.2: Boolean truth table of the full adder gate.

a combinatorial gate which receives three binary inputs and returns the sum and carry of the inputs (see Table 7.2 for the truth table of the full adder gate). n full adders may be chained together to create an n-bit adder which returns the sum of two n-bit numbers. Therefore, the half-adder is an extremely important component of a working arithmetic logic unit, a key component of all modern central processing units.

7.1 Physical implementation of the half-adder

The half-adder circuit was constructed from three NAND gates as depicted in Fig. 7.1. Operation of the half-adder via the phased chaining protocol proceeds in a manner similar to the chained NAND gates in Section 6.2. As with the NAND gates, the system is initialised with the switch in the midpoint position and every gate and driver maximally disengaged at $t^* = 0$ (Fig. 7.2(a)). In this particular image, the switches will transition from '01' in the first phase to '11' in the second phase. Thus, the half-adder will return a sum value of '1' and carry value of '0' in the first phase, transitioning to a sum value of '0' and a carry value of '1' in the second phase.

At $t^* = 1/12$ (Fig. 7.3(a)), the switches now return '01' and the switch is halted. It should be noted that the design of the switch differs slightly from the design used

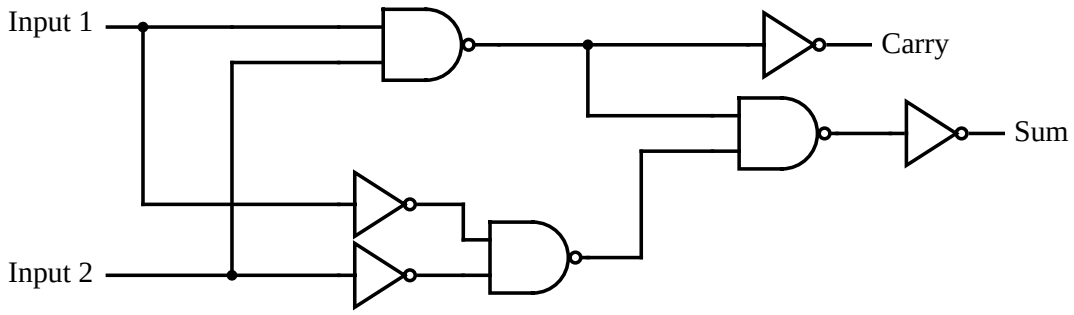


Figure 7.1: The circuit layout of the half-adder.

in the previous section as a booster has been added to increase the strength of the switch output. This modification is necessary as each switch output is coupled to two separate inputs, which increases the error rate thus necessitating an additional booster to mitigate this problem. Information from the switch outputs then passes through the inverting and non-inverting connections to the NAND gates. The upper NAND gate, receiving inverted inputs, is functionally identical to an OR gate. For this particular set of inputs, both NAND gates will return '1' at $t^* = 1/6$ with their signal boosters fully engaged. Information is then transmitted to the final NAND gate via the drivers, with the signal booster of this gate becoming fully engaged at $t^* = 1/4$. It is at this point that the first-phase error rate of both the cyan-coloured sum bit and orange-coloured carry bit (labelled as 'Sum 1' and 'Carry 1' respectively in Tables 7.3 to 7.6) is measured; for this particular example the sum is '1' and the carry '0'.

Between $t^* = 1/4$ and $t^* = 1/2$, this process is reversed, returning the system to its initial state. The second phase is initiated at this point. It is identical to the first phase except that the switches are rotated in the opposite direction such that they return '11'. The second-phase error of both sum and carry bits (labelled as 'Sum 2' and 'Carry 2' respectively) is therefore measured at $t^* = 3/4$; in this example the second-phase sum is '0' while the carry is '1'. Supplemental Video 10 provides an

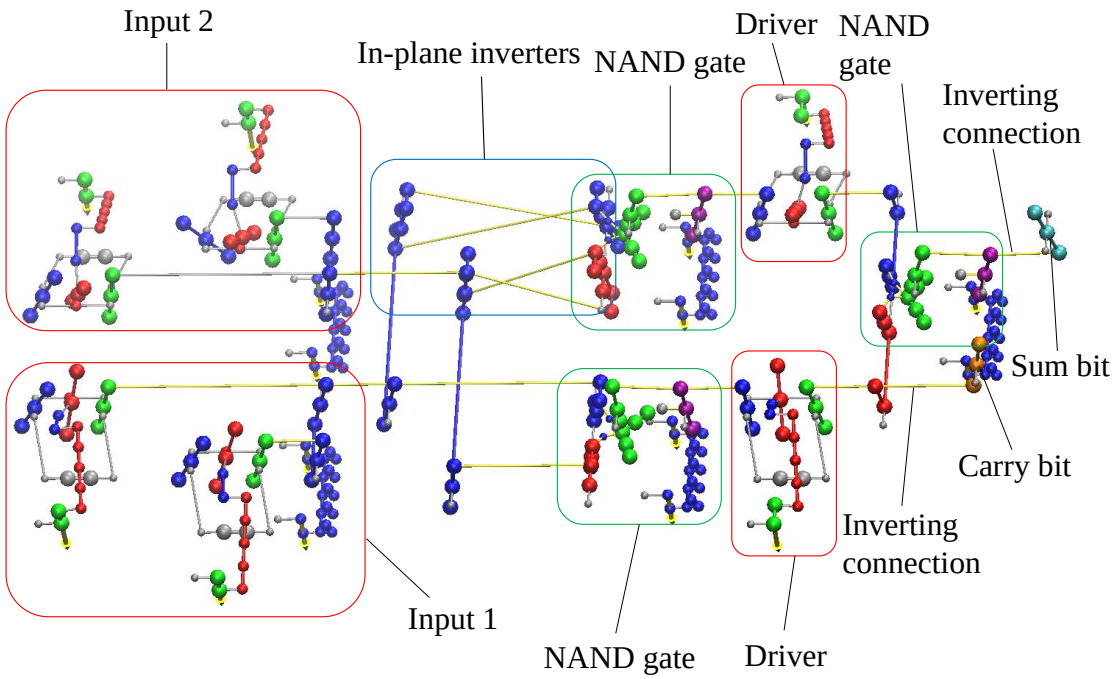


Figure 7.2: The basic half-adder at $t^* = 0$.

animation of this complete process.

In order to simplify the physical construction of the circuit, the in-plane inverter from Section 3.4 was used in place of the simpler inverting connection. However, adopting this type of inverter results in a higher error rate than the directly-tethered connection, as can be observed with the results of the error-rate simulations in Table 7.3, which are much higher than the results for the NAND gate in Table 5.2. Due to the lack of symmetry, there are twelve possible transitions which must be considered for the half-adder.

7.2 Boosting the signal

In order to mitigate the problems posed by the use of the in-plane inverter, an additional booster was placed under the inputs of the topmost NAND gate (Fig. 7.2(b)); this modification serves to effectively boost the inverter outputs. The external dipoles

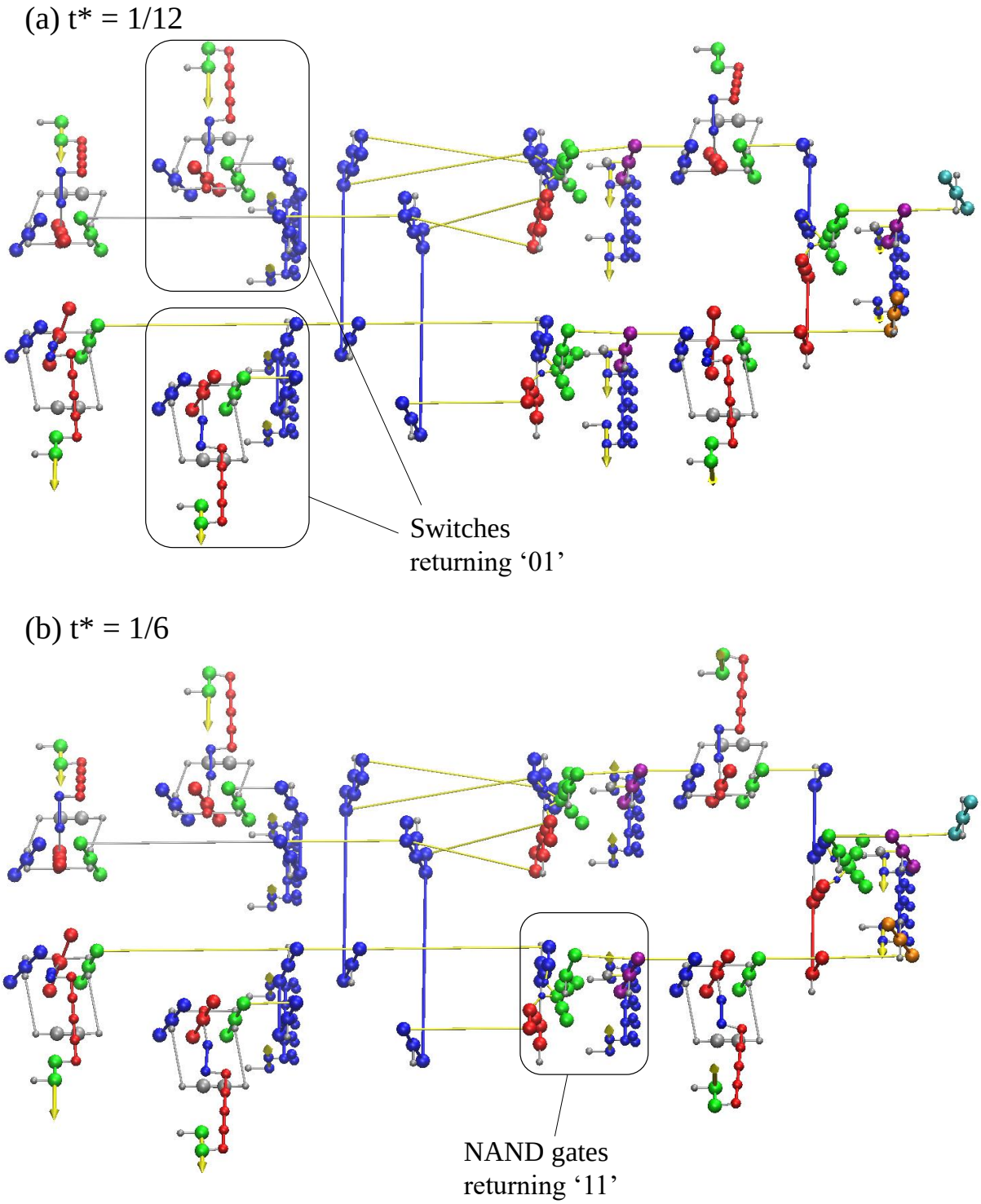


Figure 7.3: The basic half-adder between $t^* = 1/12$ and $t^* = 1/6$.

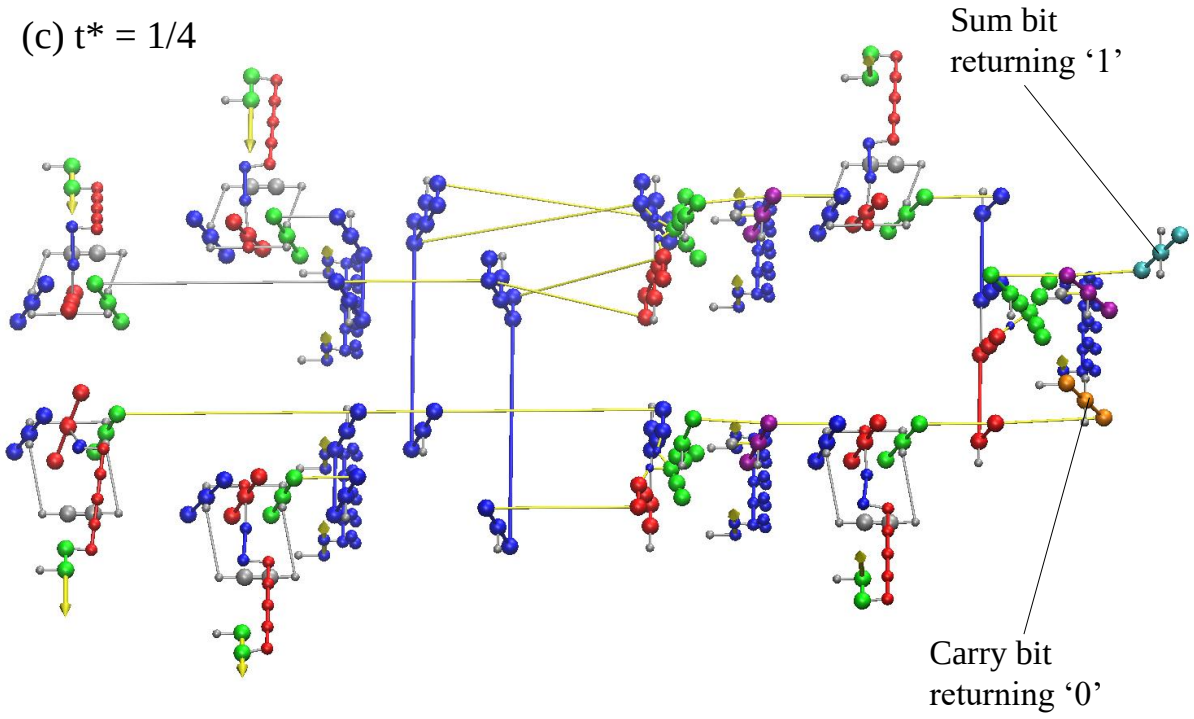


Figure 7.4: The basic half-adder at $t^* = 1/4$.

Table 7.3: Results obtained from error-rate simulations of the basic half-adder gate performed at $2T_r$ (596 K) for the evolution from a defined starting state to a given target state at a clock speed of 2.87 GHz. 4800 simulations were carried out for each transition. No errors were detected over 4800 simulations at $T \leq T_r$.

Transition	Carry 1	Carry 2	Sum 1	Sum 2
00→ 01	0	0	0	0
00→ 10	0	0	2.08×10^{-4}	0
00→ 11	0	2.08×10^{-3}	6.25×10^{-4}	0
01→ 00	0	0	2.08×10^{-4}	4.17×10^{-4}
01→ 10	0	0	0	2.08×10^{-4}
01→ 11	0	1.67×10^{-3}	0	0
10→ 00	0	0	2.08×10^{-4}	0
10→ 01	0	2.08×10^{-4}	2.08×10^{-4}	0
10→ 11	0	2.29×10^{-3}	0	2.08×10^{-4}
11→ 00	2.92×10^{-3}	0	0	2.08×10^{-4}
11→ 01	6.25×10^{-4}	2.08×10^{-4}	0	0
11→ 10	1.67×10^{-3}	0	0	0

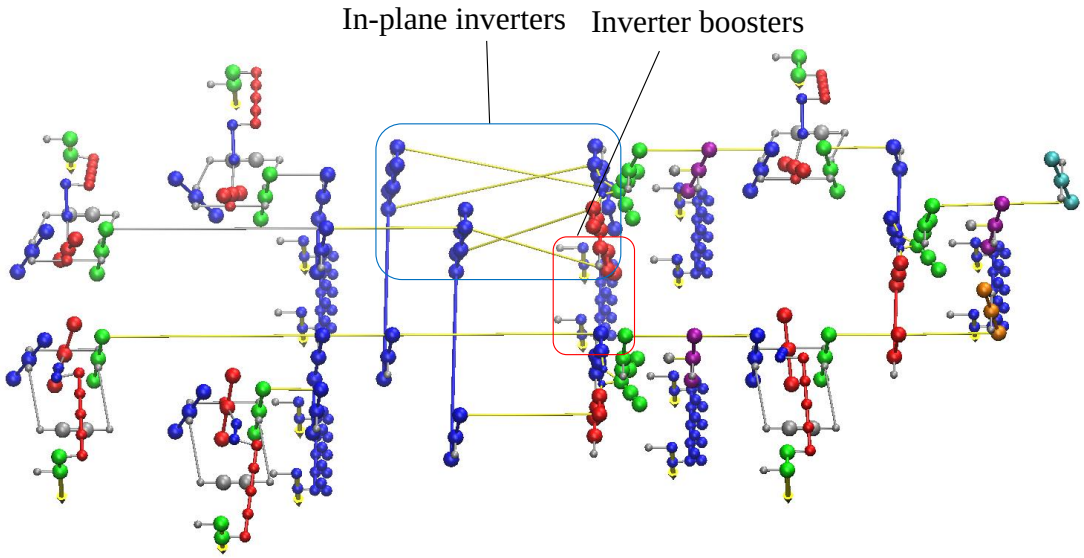


Figure 7.5: The half-adder with boosted inverters at $t^* = 0$.

which drive this signal booster have a 90° lag in phase relative to the switch. This, in turn forces the phases of all external dipoles downstream from the inverters to be shifted back by 90°. Thus, it takes 20% longer for information to propagate from the switches to the sum and carry bits in this system. In exchange for this reduction in speed, there is a sharp decrease in error rate as can be observed from the results in Table 7.4.

Further improvements in error rate can be obtained by adding more signal boosters to the drivers which separate the first two NAND gates from the final NAND gate and outputs (Fig. 7.2c). As before, addition of these signal boosters results in a 17% increase in the time needed for information to propagate along the length of the circuit; this is compensated for by a further large reduction in error rate such that there were no longer any significant errors detected over 4800 simulations at 596 K. An animation of this half-adder can be viewed in Supplemental Video 11. Comparison of the inverter-boosted system at 894 K (Table 7.5) and the inverter and driver-boosted system at the same temperature (Table 7.6) demonstrates the large reduction in error rate.

Table 7.4: Results obtained from error-rate simulations of the half-adder gate with boosted inverter outputs performed at $2T_r$ (596 K) at a clock speed of 2.87 GHz. 4800 simulations were carried out for each transition. No errors were detected over 4800 simulations at $T \leq T_r$.

Transition	Carry 1	Carry 2	Sum 1	Sum 2
00→ 01	0	0	0	0
00→ 10	0	0	0	0
00→ 11	0	4.17×10^{-4}	0	0
01→ 00	0	0	0	0
01→ 10	0	0	0	2.08×10^{-4}
01→ 11	0	6.25×10^{-4}	0	0
10→ 00	0	0	0	0
10→ 01	0	0	0	0
10→ 11	0	6.25×10^{-4}	0	0
11→ 00	4.17×10^{-4}	0	0	0
11→ 01	6.25×10^{-4}	0	0	0
11→ 10	4.17×10^{-4}	0	0	0

Table 7.5: Results obtained from error-rate simulations of the half-adder with boosted inverter outputs performed at $3T_r$ (894 K) at a clock speed of 2.87 GHz. 4800 simulations were carried out for each transition.

Transition	Carry 1	Carry 2	Sum 1	Sum 2
00→ 01	0	2.08×10^{-4}	2.08×10^{-4}	8.33×10^{-4}
00→ 10	0	6.25×10^{-4}	4.17×10^{-4}	2.08×10^{-4}
00→ 11	0	3.96×10^{-3}	4.17×10^{-4}	0
01→ 00	4.17×10^{-4}	0	2.08×10^{-4}	6.25×10^{-4}
01→ 10	6.25×10^{-4}	0	1.04×10^{-3}	8.33×10^{-4}
01→ 11	1.25×10^{-3}	3.13×10^{-3}	1.46×10^{-3}	0
10→ 00	2.08×10^{-4}	0	6.25×10^{-4}	2.08×10^{-4}
10→ 01	2.08×10^{-4}	6.25×10^{-4}	6.25×10^{-4}	1.25×10^{-3}
10→ 11	0	4.38×10^{-3}	1.04×10^{-3}	0
11→ 00	2.92×10^{-3}	0	0	4.17×10^{-4}
11→ 01	3.96×10^{-3}	1.25×10^{-3}	2.08×10^{-4}	1.25×10^{-3}
11→ 10	4.38×10^{-3}	4.17×10^{-4}	0	1.04×10^{-3}

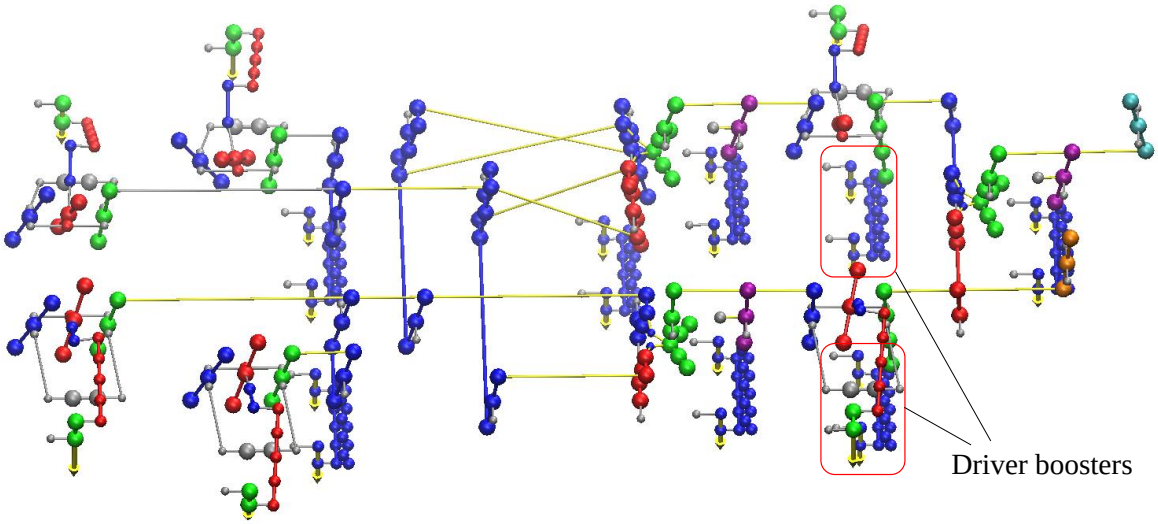


Figure 7.6: The half-adder with boosted inverters and drivers at $t^* = 0$.

Thermodynamic reversibility simulations were carried out on the inverter and driver-boosted half-adder using a similar protocol to the chained NAND gates in Section 6.2. The results are summarised in Fig. 7.7 and show that the half-adder is tending towards thermodynamic reversibility within the limit of clock speeds tested.

Table 7.6: Results obtained from error-rate simulations of the half-adder with boosted inverter and driver outputs performed at $3T_r$ (894 K) at a clock speed of 2.87 GHz. 4800 simulations were carried out for each transition. No errors were detected over 4800 simulations at $T \leq 2T_r$.

Transition	Carry 1	Carry 2	Sum 1	Sum 2
00→ 01	0	0	1.46×10^{-3}	0
00→ 10	0	0	1.04×10^{-3}	0
00→ 11	0	6.25×10^{-4}	8.33×10^{-4}	6.25×10^{-4}
01→ 00	0	0	0	2.08×10^{-3}
01→ 10	1.25×10^{-3}	0	1.25×10^{-3}	0
01→ 11	8.33×10^{-4}	0	8.33×10^{-4}	0
10→ 00	0	0	0	4.17×10^{-4}
10→ 01	0	4.17×10^{-4}	0	4.17×10^{-4}
10→ 11	2.08×10^{-4}	0	2.08×10^{-4}	0
11→ 00	0	0	0	0
11→ 01	0	0	6.25×10^{-4}	6.25×10^{-4}
11→ 10	4.17×10^{-4}	0	4.17×10^{-4}	0

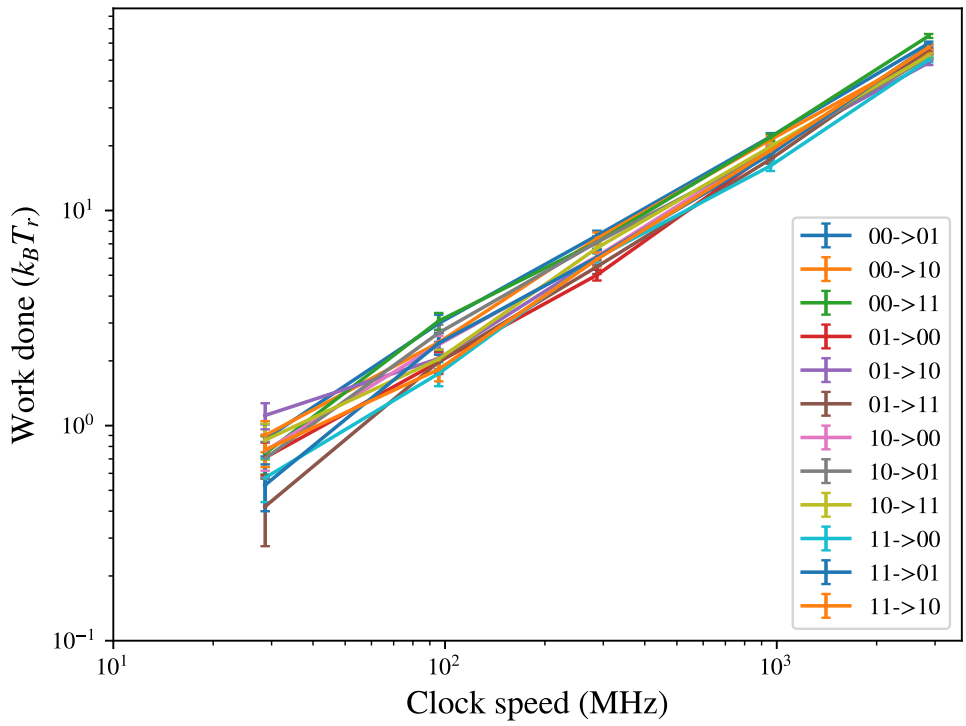


Figure 7.7: Plot of work done against clock speed in the inverter and driver-boosted half-adder system. Each data point corresponds to an average over 96 simulations.

Chapter 8

Pipelining

Although phased chaining is demonstrably effective at reducing the error rate while remaining thermodynamically reversible when compared to direct mechanical coupling of logic gates, it requires the use of multiple external dipoles to drive the system, one for each distinct level of the circuit. In addition, phased chaining also necessitates that every gate in the circuit process its input and return its output before any upstream gates can be released to accept new inputs. As a result, phase-chained systems both have a low throughput and are difficult to scale.

A simple method to reduce the complexity and increase the speed of a combinatorial circuit is to drive the system via a continuously rotating single external dipole while maintaining the staggering of phases. This technique, known as pipelining [76], allows each gate in the chain to simultaneously process information once per clock cycle, as opposed to the phased chaining protocol where each gate can only process information once during the time it takes for information to pass from the first level of the circuit to the last, regardless of how many clock cycles are required. As our framework currently lacks any method of preserving information beyond a single clock cycle, the pipelining implementation implemented in this chapter more closely resembles the wave-pipelining approach [77, 78, 79] than the standard register-based

pipelining approach described in circuit design textbooks, which does require persistent information storage (sequential logic) [80]; however, I will refer to the protocol used in this chapter as pipelining for the sake of simplicity.

Although pipelining allows for a large increase in throughput, a pipelined system does not possess intrinsic logical reversibility in the same manner that a phase-chained system does. In the absence of a mechanism that enables or simulates the storage of information beyond a single clock cycle, a pipelined circuit will be logically irreversible for a given set of inputs if any level of the circuit is not itself logically reversible. As mentioned in section 1.2.4, it is generally not practical to perform logically irreversible operations in a thermodynamically reversible manner as there is only a limited subset of optimal distributions of starting states for which this is possible [17, 18, 19], with all other distributions incurring a mismatch cost. However, even in the case where the system is intialised within the set of optimal starting distributions, there may still be residual entropy production as a result of mechanical inefficiencies in the precise physical implementation of the process. This resdiual entropy production is specific to a particular system and is not generalisable.

Averting the loss in thermodynamic efficiency associated with logical irreversibility is possible through either storing enough information beyond one clock cycle to render the circuit logically reversible, or simulating the effect of information storage in the event that persistent memory storage is not available. In this chapter, I shall demonstrate the use of simulated information storage to create a simple logically and thermodynamically reversible pipelined circuit, and show that it is comparable in thermodynamic efficiency to phased chaining. I will then use the same technique to create a logically and thermodynamically reversible four-NAND gate pipelined system.

8.1 Phased Chaining vs. Pipelining

In phased chaining, each level of logic gates is engaged only after the previous level has become engaged. A logic gate will also never become disengaged until every gate in every level ahead of it has become disengaged. Thus, reversing a phase-chained system requires time equal to that required to engage it fully. In the case of the circuit depicted in Fig. 8.1, should the output of the circuit given inputs of both '0' and both '1' to the initial gates of the circuit be desired, the circuit first has to be fully reversed, taking $t = 3$, and then executed in the forward direction with all inputs set to '0', taking nine units of time in total. If the system is to be returned to its original state, as is necessary in order to demonstrate thermodynamic reversiblility (see Section 8.2), a further three units of time will be required, for a total of twelve units in total.

In contrast, for a pipelined system, information is passed through the circuit without waiting for all levels of logic gates to receive an input. In the case of the circuit depicted in Fig. 8.2, the '0' inputs are fed into the initial gate of the circuit before the result of the '1' inputs have been fully processed by the final gate. This results in a reduction of the time requirement to receive the results of both computations to only five units, and a reduction of the time requirement to return the circuit to its original state to only ten units. While the gains in speed are fairly minor in this particular case, the speed advantage of pipelining relative to phased chaining grows as the length of the circuit and the number of potential inputs increases.

The increase in throughput and speed does come at the cost of potentially rendering the circuit logically irreversible. If the final outputs of the circuit in Fig. 8.8 are not stored after $t = 2$ and $t = 5$, and the circuit is run in reverse, there will be no way to ensure that the final NAND gate receives the correct input (via the former output channel) during the reverse pipelining process. This results in the circuit becoming logically irreversible. This irreversibility occurs despite the fact that each individual

level of the circuit is invertible for the given input.

8.2 Logical Irreversibility in Pipelined Processes

In order to properly determine the thermodynamic efficiency of a pipelined process, it is necessary to run the process in both forward and reverse directions such that the system returns to its original state. This is due to the fact that the final state may have a very different potential energy from the starting state. However, as previously mentioned, a pipelined circuit may become logically irreversible if its inputs are not stored beyond one clock cycle.

This phenomenon can be demonstrated with a simple system, consisting of two bits chained to a switch, with each element being separated by a driver (Fig. 8.3). The two bits are equipped with signal boosters, which force the bit into either the '0' or '1' position when maximally engaged. The signal booster of the first bit is staggered 90° out-of-phase with the first driver while the signal booster of the second bit is in phase with the second driver and 90° out-of-phase with the signal booster of the first bit. The fact that the signal booster of the first bit is 90° out-pf-phase with its driver allows the first bit to more effectively retain and transmit information to the driver of the second bit, since the information contained in the first bit is preserved for some time even after the driver of the first bit disengages.

In contrast to phased chaining, all internal dipoles of the system rotate at the same speed throughout the simulation regardless of initial starting position, with the exception of the dipoles of the switch which travel at only half the speed of the other dipoles due to the fact that the drivers of the switch only have to move through 180° to engage with the other set of bits. In theory, both sets of dipoles can be driven by a single external dipole through the use of reduction gearing.

The two-bit system is designed to receive an input from the switch to the first

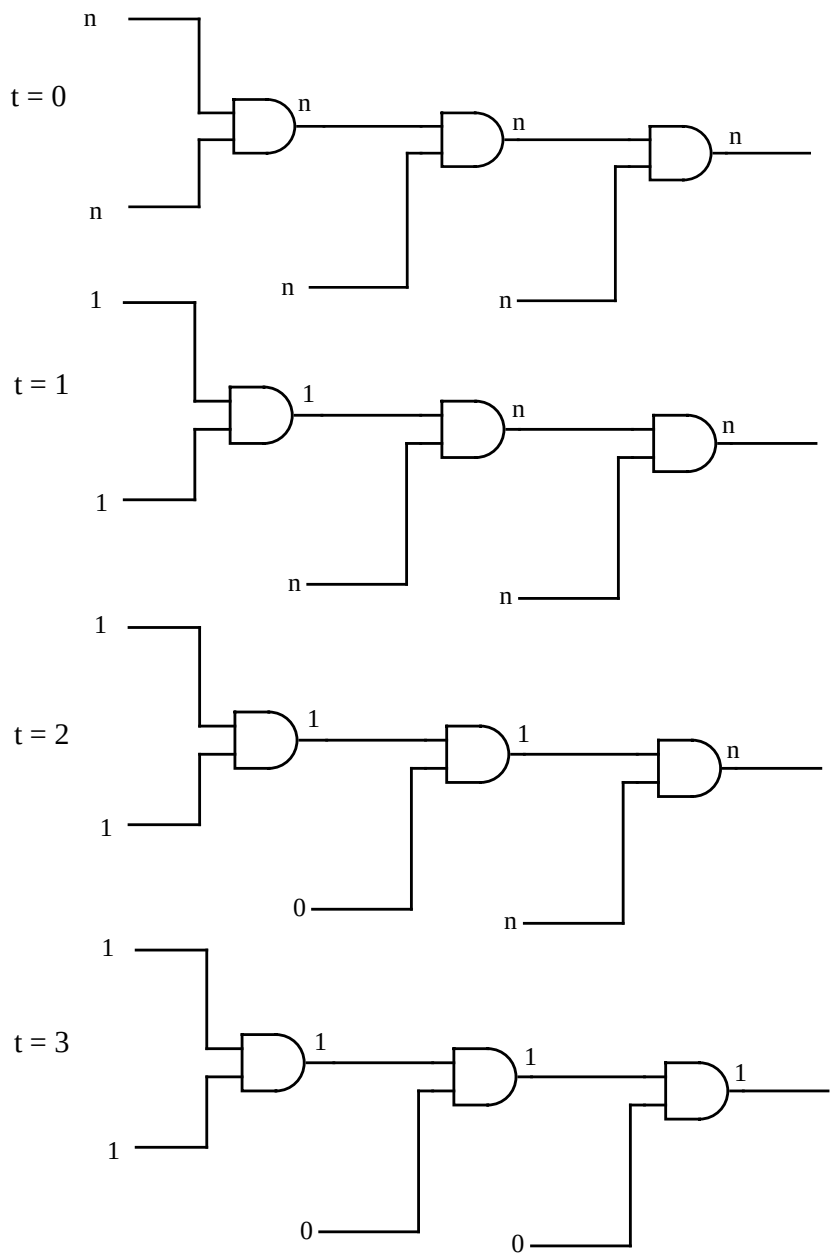


Figure 8.1: A simplified demonstration of phased chaining in a circuit containing three chained AND gates. Units of time are arbitrary. States labelled 'n' are neutral disengaged states set to neither '0' or '1'.

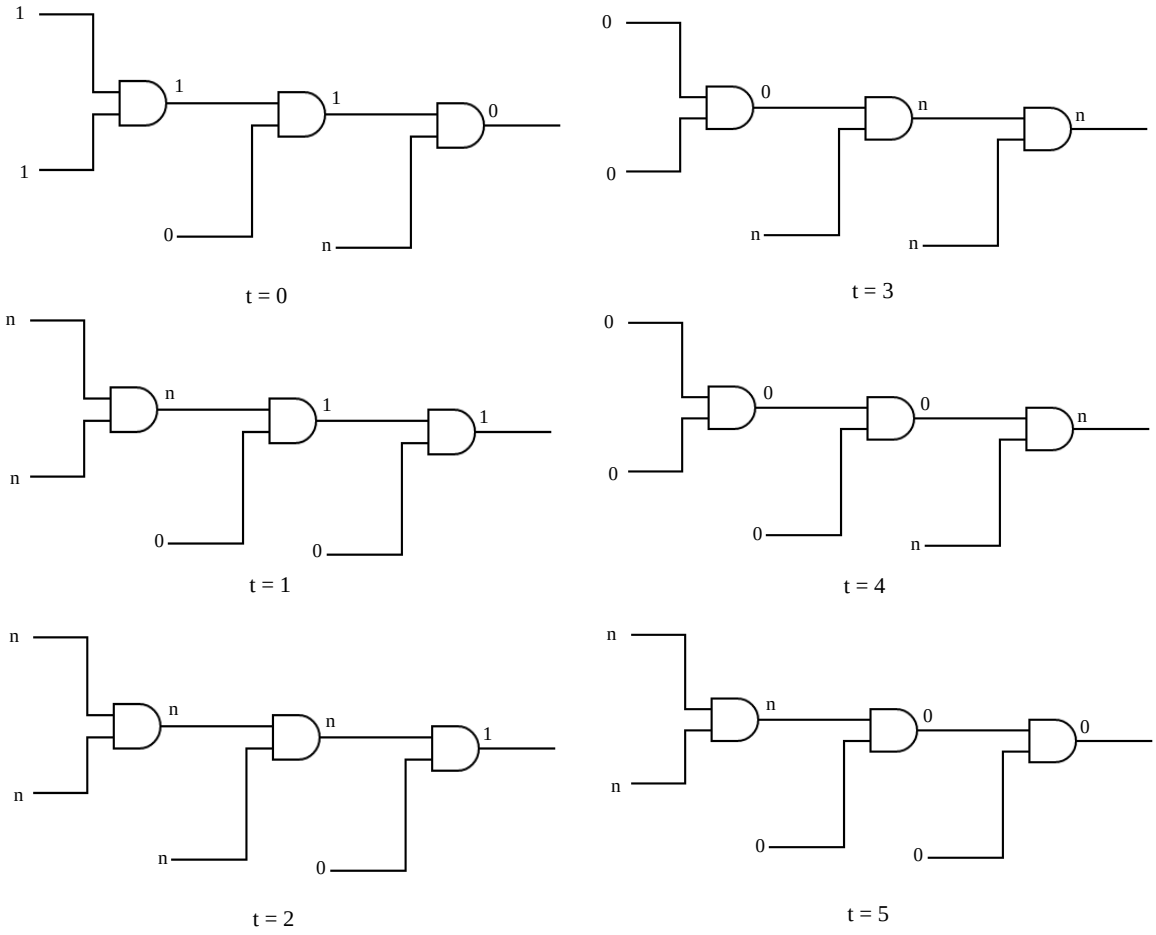


Figure 8.2: A simplified demonstration of pipelining in a circuit containing three chained AND gates. Units of time are arbitrary. States labelled ‘n’ are neutral disengaged states set to neither ‘0’ or ‘1’.

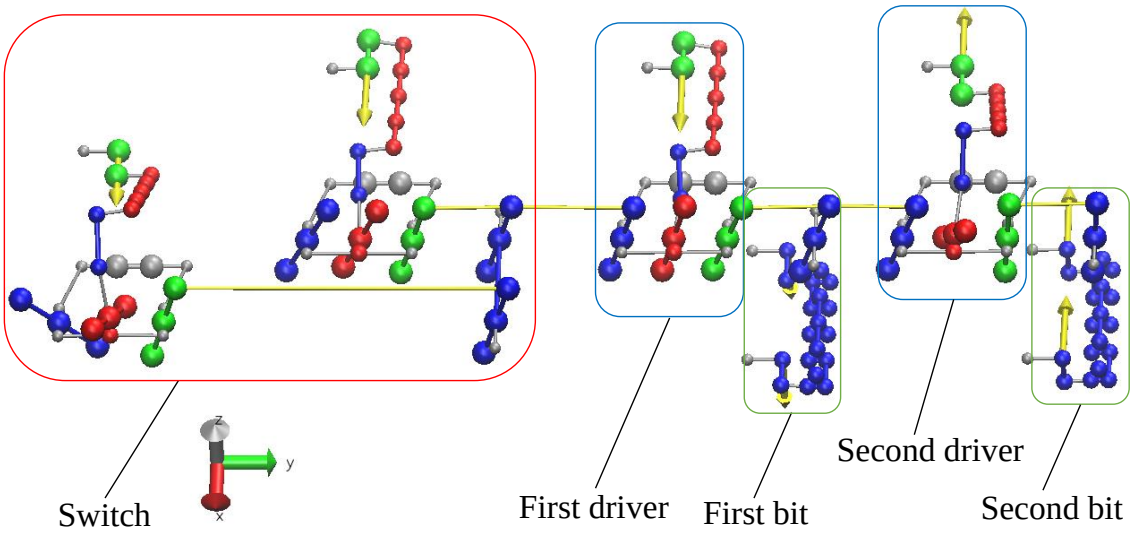


Figure 8.3: The two-bit pipelined system described in Section 8.2 at $t^* = 0$.

bit and then transfer it to the second bit. The switch will be toggled between both possible inputs ('1' and '0'); this state will be transduced into the first bit, followed by the second bit after a short delay. As previously mentioned, these processes are driven by the rotation of an external dipole, which interacts with the internal dipoles in the system. After both inputs have reached the second bit, the dipoles driving the system will then be rotated in reverse for an identical duration. If the system were logically and thermodynamically reversible, every pair of points in the forward and reverse trajectories that are the same distance in time from the midpoint of the simulation should occupy the same ensemble of states. As I shall demonstrate, this is not the case for the pipelined two-bit system.

At $t^* = 0$, the system is initialised with the signal boosters of both bits disengaged and the switch toggled to the '1' state. From $t^* = 0$ to $t^* = 1/6$, the signal boosters of the first and second bits become engaged in turn, effectively copying the '1' state from the switch to the second bit (Fig. 8.4). Between $t^* = 1/6$ and $t^* = 1/3$, the locks of the two bits disengage and the switch toggles between the '1' and '0' states. At $t^* = 1/3$, the switch is now toggled to the '0' state. From $t^* = 1/3$ to $t^* = 1/2$, the signal boosters of the first and second bits become engaged in turn, copying the

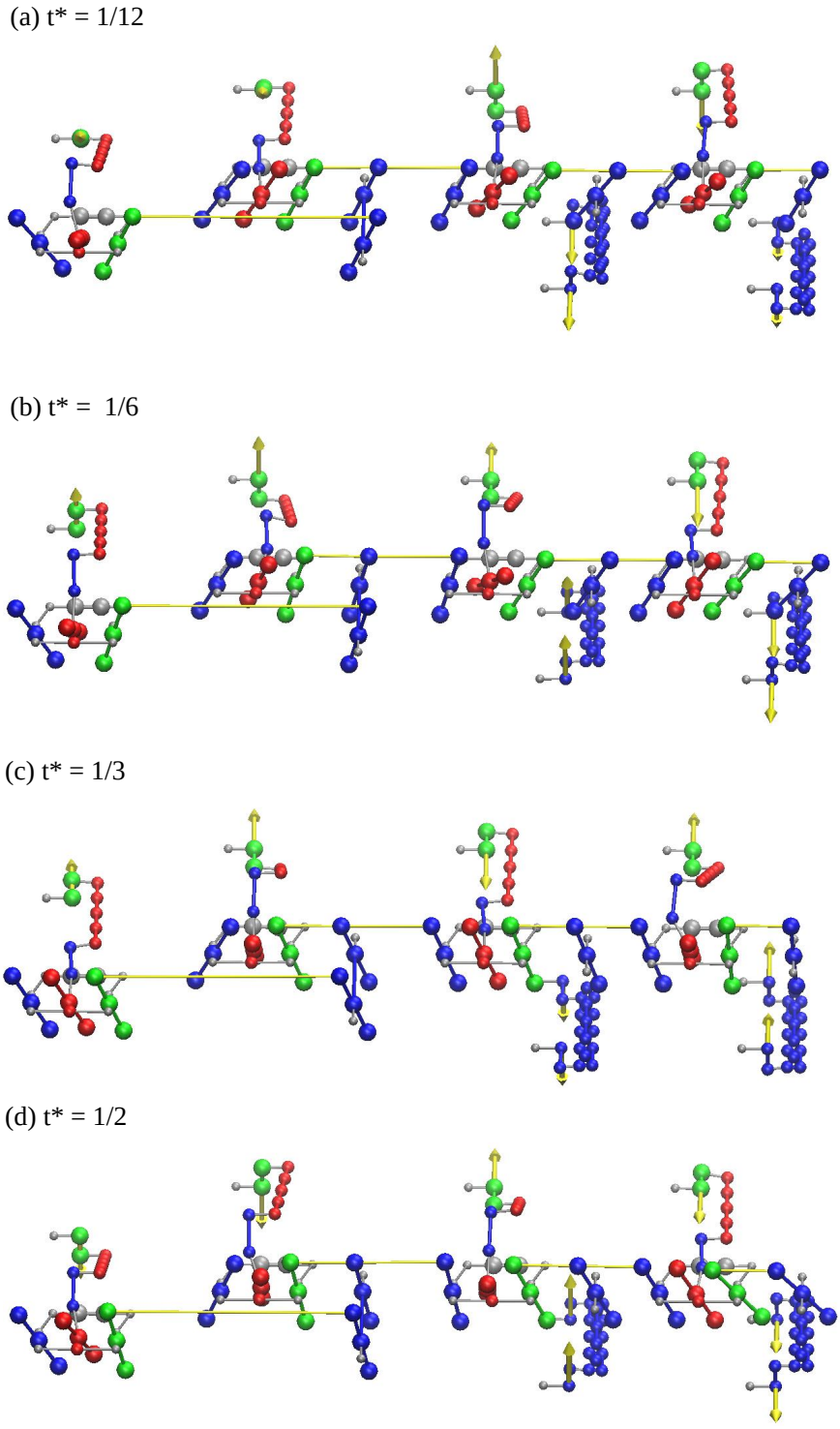
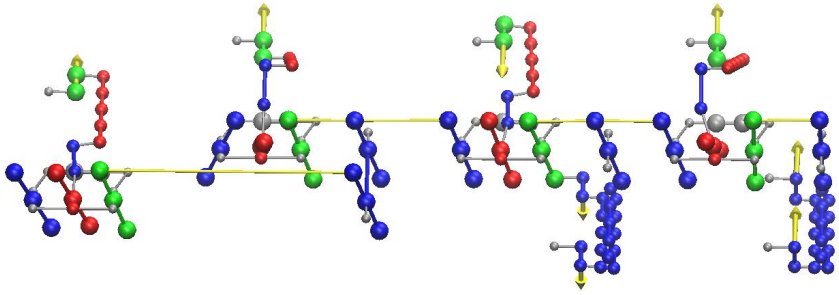
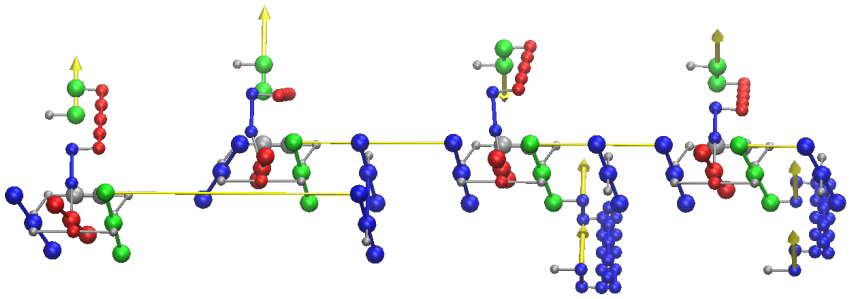


Figure 8.4: The two-bit pipelined system described in section 8.2 between $t^* = 1/12$ and $t^* = 1/2$.

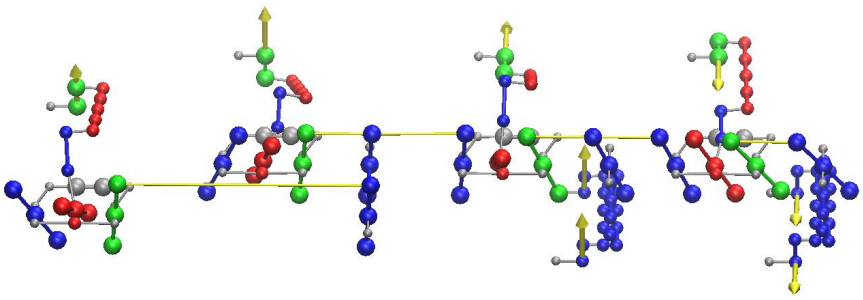
(e) t* = 2/3



(f) t* = 3/4



(g) t* = 5/6



(h) t* = 11/12

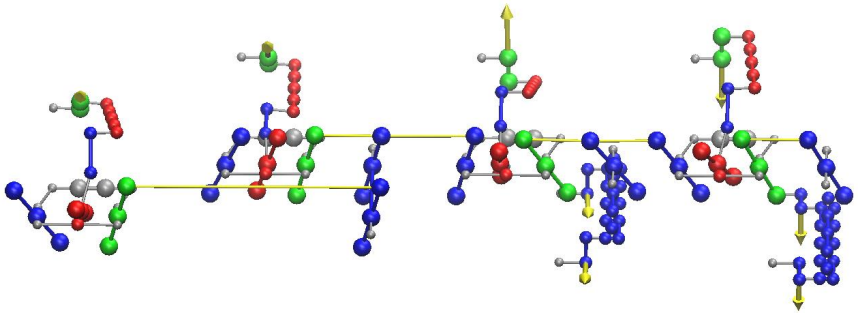


Figure 8.5: The two-bit pipelined system described in section 8.2 between $t^* = 2/3$ and $t^* = 11/12$.

'0' state from the switch to the second bit. The process is now reversed.

Between $t^* = 1/2$ and $t^* = 2/3$, information now flows from the second bit to the first bit (Fig. 8.5). As the second bit was still locked into the '0' state at $t^* = 1/2$, the '0' state is correctly transduced to the first bit. However, a different situation develops at $t^* = 5/6$, when the signal booster of the second bit completes a 360° revolution. As the second bit was not constrained into any particular state before its signal booster is engaged, there is a chance it is forced into the '0' state instead of the '1' state that it was in at $t^* = 1/6$, a direct consequence of the logical irreversibility of the system. The chance of this occurring, henceforth referred to as the failure rate, is approximately 50% at the highest clocks speeds and decreases with clock speed (Table 8.1). This erroneous '0' state is transduced to the first bit at $t^* = 11/12$.

As the clutch of the driver located between the switch and the first bit engages and brings the two components into mechanical contact, it causes a clash between the first bit, forced into to the '0' state by its signal booster, and the switch, currently toggled to the '0' state. This clash is resolved by the first bit being forced into the '1' state despite the engagement of its signal booster, a process which is highly thermodynamically inefficient and results in the conversion of a large amount of potential energy into irrecoverable kinetic energy, as can be observed from the results recorded in Fig. 8.6. Regardless of clock speed, a failed reverse process requires approximately an additional $75\,kT$ of work to complete compared to a successful reverse process. An animation of a failed reverse process can be viewed in Supplemental Video 1b, while an animation of a successful reverse process can be viewed in Supplemental Video 1a. The reduction in failure rate allows for a slight increase in thermodynamic efficiency as the clock speed is decreased, but the reduction in work done as a function of clock speed is still far less than the 1:1 ratio which would be expected if viscous drag were the primary cause of efficiency losses.

Notably, this clash does not occur when the system is driven via phased-chaining,

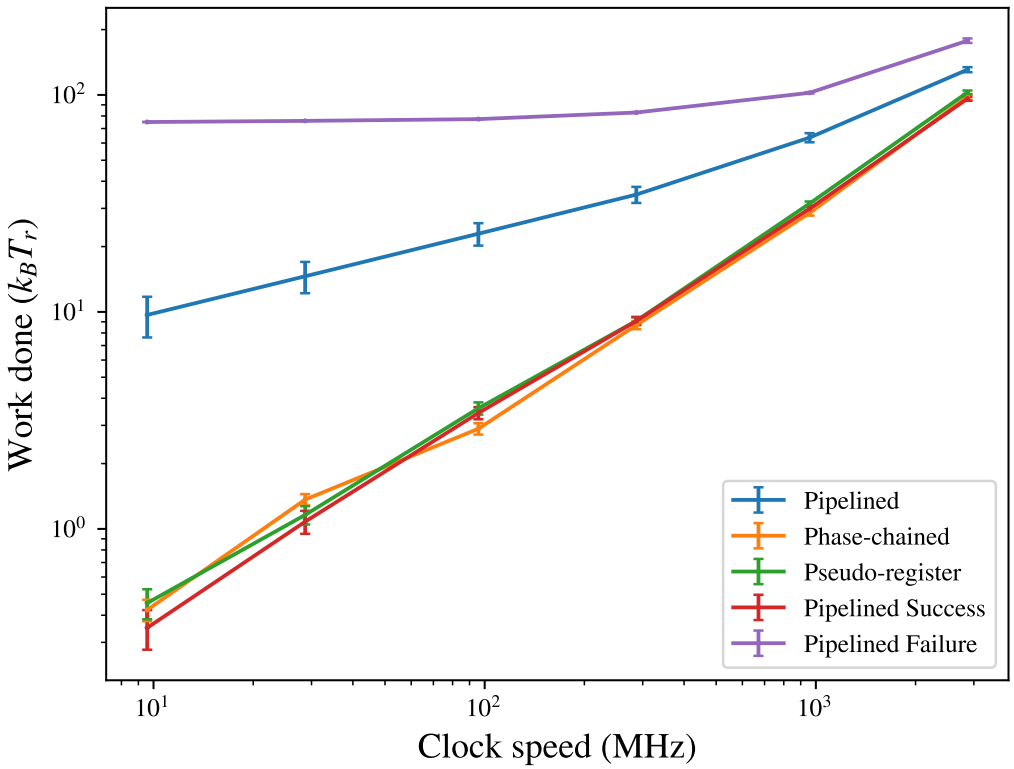


Figure 8.6: Work done against clock speed for the two-bit system with staggered locks with the pipelining, phase-chaining and pseudo-register protocols, with separate plots for successful and failed reverse pipelining operations. 144 simulations were run for each data point. Before all protocols described in this chapter were executed, the system involved was subject to energy minimisation followed by thermal equilibration. All simulations were run at 298 K.

Table 8.1: Failure rate of the two-bit pipelined system relative to clock speed. A reverse pipelining process is defined as a failure if the second bit is in a '0' state rather than a '1' state at $t^* = 3/4$.

Clock speed (MHz)	Failure rate
2870	0.493
957	0.465
287	0.347
95.7	0.264
28.7	0.181
9.57	0.125

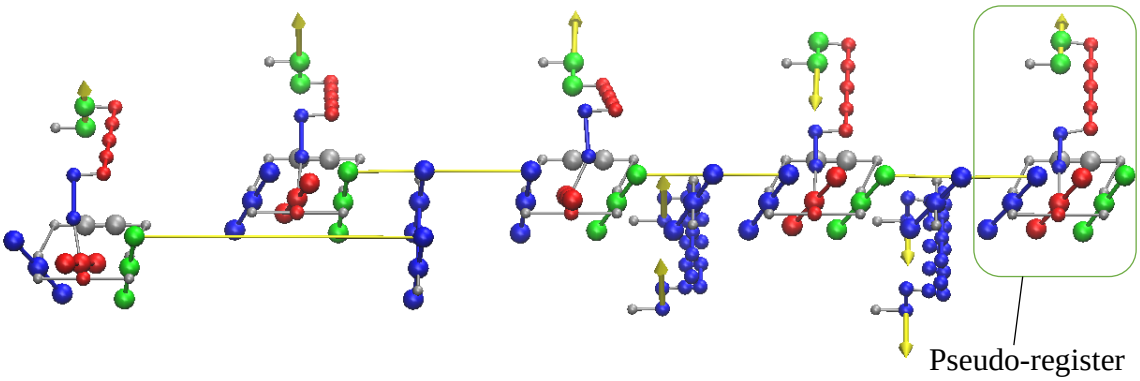


Figure 8.7: The two-bit pipelined system chained to a pseudo-register at $t^* = 5/6$.

which is both logically and thermodynamically reversible as the signal booster of the second bit is never engaged when the signal booster of the first bit is disengaged. As can be observed from Fig. 8.6, phased chaining is comparable in thermodynamic efficiency to a successful pipelining operation. An animation of the two-bit system driven via phased chaining is available in Supplemental Video 1c.

Mitigating the loss of efficiency associated with logical irreversibility is possible if the circuit is rendered logically reversible by preserving the information contained in the second bit at $t^* = 3/4$. In the absence of sequential logic, this can be accomplished by the chaining of an additional driver set to '1' to the second bit (Fig. 8.7). The dipole of this driver was synchronised to the signal booster of the second bit such that the driver returns '1' at the same time that the signal booster of the second bit reaches its maximal engagement. Thus, this driver will invariably force the second bit into the state '1'. As this driver simulates the information from the output bit being stored in a register even after the switches are no longer engaged before being subsequently re-written into the output bit during the reverse process, it shall henceforth be referred to as a pseudo-register. An animation of the two-bit pipelined system chained to a pseudo-register can be viewed in Supplemental Video 2.

The results of the simulations with the switch-linked two-bit pipelined system chained to the pseudo-register are summarised in Fig. 8.6. From these results, it

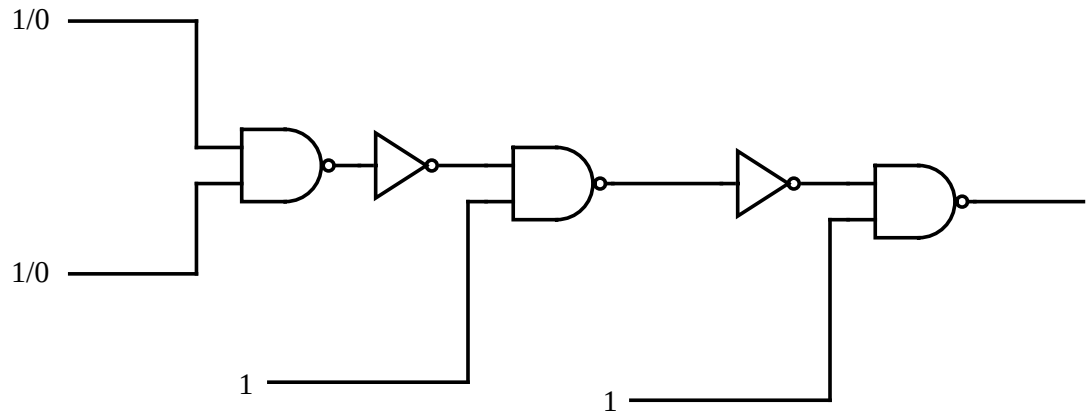


Figure 8.8: One of three circuits used to demonstrate reversible pipelining. This particular circuit has a chain length of 3 and can be extended or truncated by the addition or removal of inverter-NAND units. The other two circuits have lengths of 2 and 4 NAND gates. This particular circuit is equivalent to the circuit outlined in Fig. 8.2 since the NAND-inverter units are equivalent to an AND gate.

can be observed that preserving the input and thus rendering the circuit logically reversible is an effective method of greatly reducing the inefficiencies associated with pipelining.

8.3 Reversible Pipelining

A series of circuits identical to the one used to demonstrate phased chaining in chapter 7 [49] was used to implement a logically reversible pipelined circuit (Fig. 8.8). From the standpoint of ensuring logical reversibility, this circuit has the useful feature of being conditionally invertible; that is, it is possible to fully reconstruct the input of a given NAND gate from its output as long as the other input bit (in this case, '1') is preserved. However, as mentinoed in section 8.1, the circuit in Fig. 8.8 does not have a pseudo-register attached to preserve the final input bit and is therefore logically irreversible in its current form.

In a manner analogous to the two-bit pipelined system, the phases of each gate are

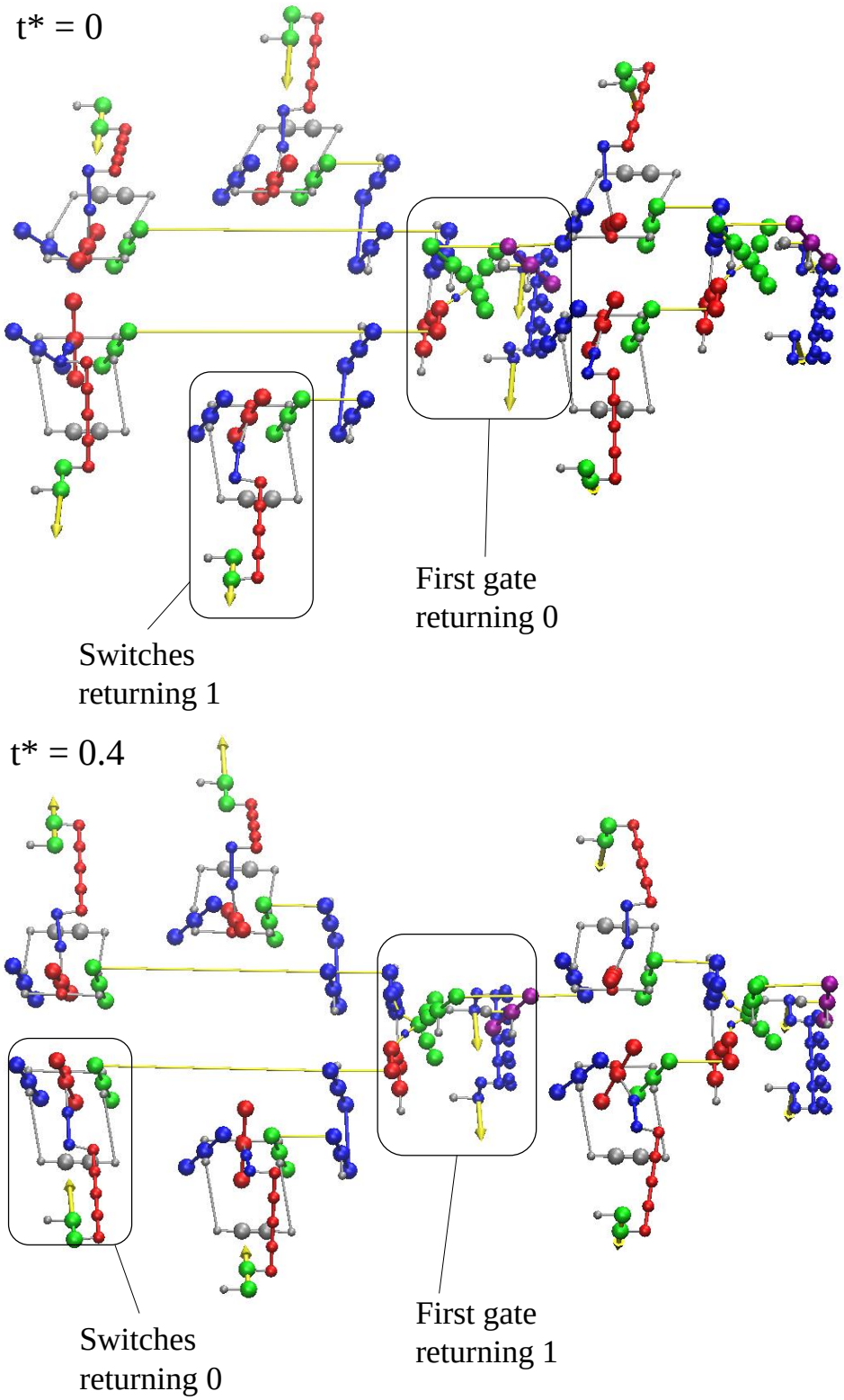


Figure 8.9: The two-gate pipelined system between $t^* = 0$ and $t^* = 0.4$.

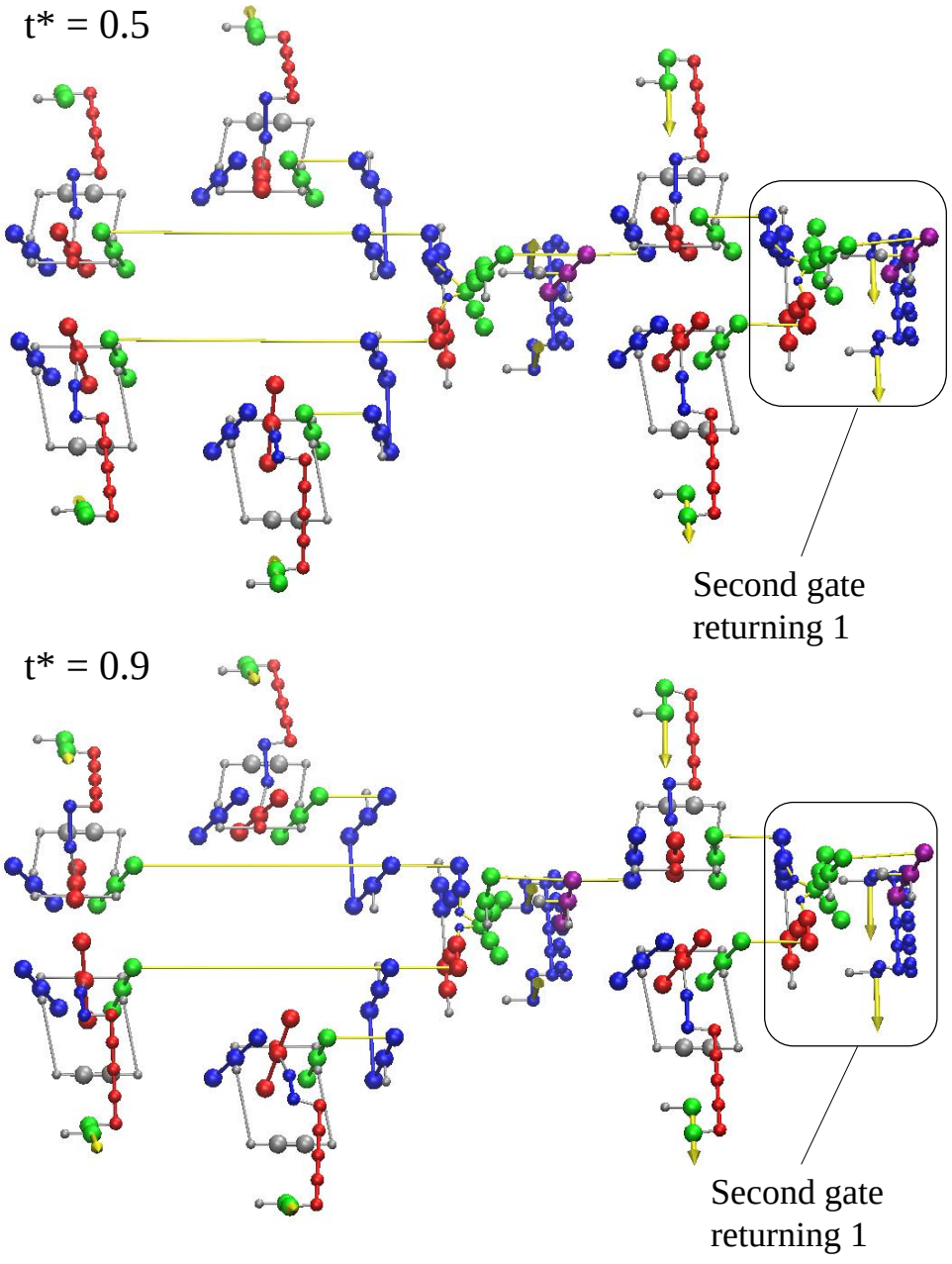


Figure 8.10: The two-gate pipelined system between $t^* = 0.5$ and $t^* = 0.9$.

offset by 90° such that the signal booster of any given gate begins to engage when the signal booster of the previous gate becomes maximally engaged. The protocol used is equivalent to that described in Fig. 8.2 since each NAND-inverter unit is equivalent to an AND gate.

To better illustrate the details of the implementation of this specific pipelining protocol, I shall use the example of two chained NAND gates (Fig. 8.9). At $t^* = 0$, the dipoles of the switches are initialised such that the switches return '1' and the signal booster of the first gate is initialised at the fully engaged position. The drivers and signal boosters of the second gate are initialised 90° behind the signal booster of the first gate. It is at this point that the first-phase error rate of the first gate is measured. At $t^* = 0.1$, the drivers and signal booster of the second gate become fully engaged and the first-phase error rate of the second gate is measured. At $t^* = 0.4$, the switches have fully transitioned to the '0' state and the signal booster of the first gate is fully engaged, returning '1'. It is at this point that the second-phase error rate of the first gate is measured. At $t^* = 0.5$, the drivers and signal booster of the second gate become fully engaged and the second-phase error rate of the second gate is measured.

At $t^* = 0.5$, the dipoles driving the circuit are reversed. Between $t^* = 0.5$ and $t^* = 0.9$, the drivers and signal boosters of the second gate rotate one full cycle. Simultaneously, the signal booster of the first gate rotates one full cycle between $t^* = 0.6$ and $t^* = 1$. However, at $t^* = 0.9$, the first gate has already been disengaged for three quarters of a clock cycle and no longer carries any useful information. Therefore, as the signal booster of the second gate engages, it falls on a randomised rather than a well-defined state. This loss of information correlation results in the circuit becoming logically irreversible, as was the case with the two-bit pipelined system in the absence of a pseudo-register. Animations of for the two, three and four NAND gate pipelined circuits can be viewed in Supplemental Videos 3-1, 3-2 and 3-3 respectively.

Simulations to determine the thermodynamic efficiency for the pipelined NAND circuit were executed according to the protocol described above. The results are summarised in Figs. 8.11. The work done in the pipelining processes is an order of magnitude higher than that of the analogous system driven via phased chaining described in chapter 8, reflecting the inefficiencies introduced by logical irreversibility due to the failure to preserve the state of the final output bit. Failure rates for the reverse pipelining processes for the pipelined NAND circuit are recorded in Table 8.2 and the work done for successful and failed reverse pipelining processes are summarised in Fig. 8.12. Unlike the two-bit pipelined system, the differences in thermodynamic efficiency between successful and failed reverse pipelining processes, while still present, are less clear, and the successful pipelining processes are still significantly less efficient than their phase-chained counterparts.

In order to render the circuit logically reversible, a pseudo-register set to '0' was chained to the end of the final NAND gate, as illustrated in Fig. 8.14. Such a system should, in theory, be logically reversible. Animations of for the two-, three- and four-NAND gate pipelined circuits chained to a pseudo-register can be viewed in Supplemental Videos 4-1, 4-2 and 4-3a respectively. The results of the reversibility simulations were run for this modified system are summarised in Fig. 8.13. While this modification does restore approximate thermodynamic reversibility in the two- and three-gate systems, results for the four-gate system do not differ significantly from the unmodified example.

The failure rate of reverse pipelining in pseudo-register equipped circuits was not detectable over 144 simulations regardless of length or clock speed; thus, failed reverse pipelining cannot be the reason the four-gate pipelined system is far less efficient than its phase-chained counterpart. Instead, the most likely reason for the irreversibility of the four-gate system is its high error rate, as can be observed from the results of error-rate simulations performed on the system with an additional driver in Table 8.3

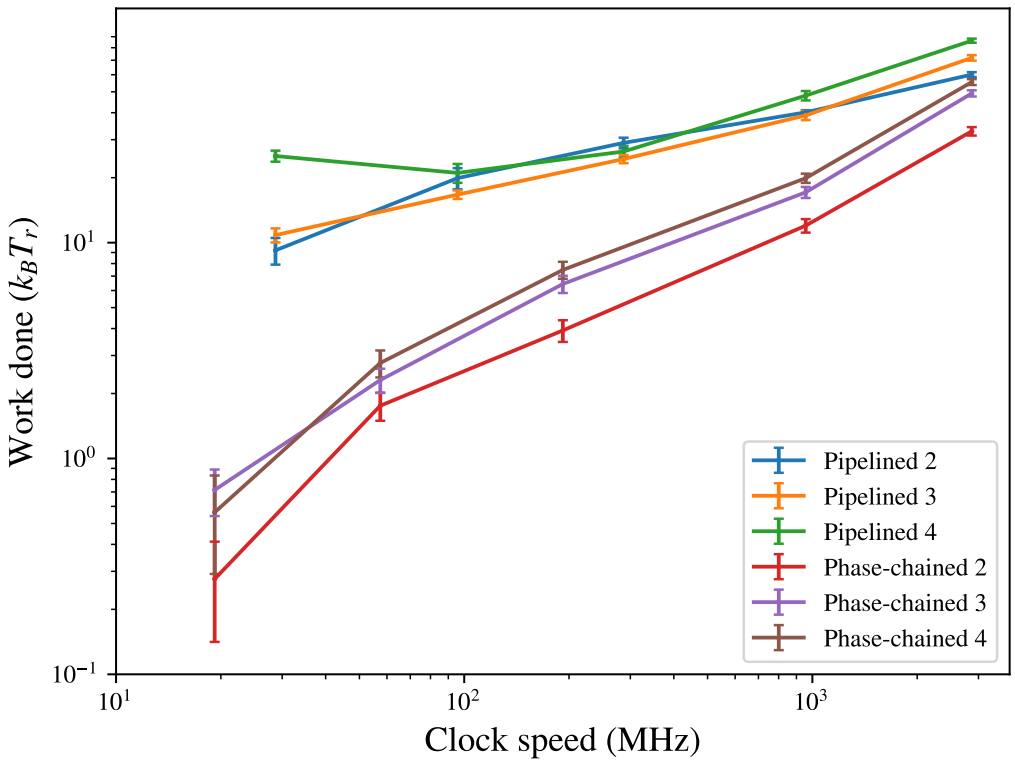


Figure 8.11: Work done against clock speed for different chain lengths in NAND gates driven via the pipelining and phased chaining protocols. Each data point corresponds to an average over 144 simulations.

in the forward pipelining process. Relative to phased-chained circuits of the same type, pipelined circuits generally have more simultaneous moving parts, increasing the difficulty of optimisation to reduce error rate. In this particular case, the high error rate is likely due to the inability of the clutch to fully prevent the upstream gates from influencing the downstream gates even when fully disengaged; an animation of this erroneous state can be viewed in Supplemental Video 4-3b. The thermodynamic irreversibility induced by the high error rate highlights the importance of optimising the clutching mechanism of the driver if effective pipelining is desired.

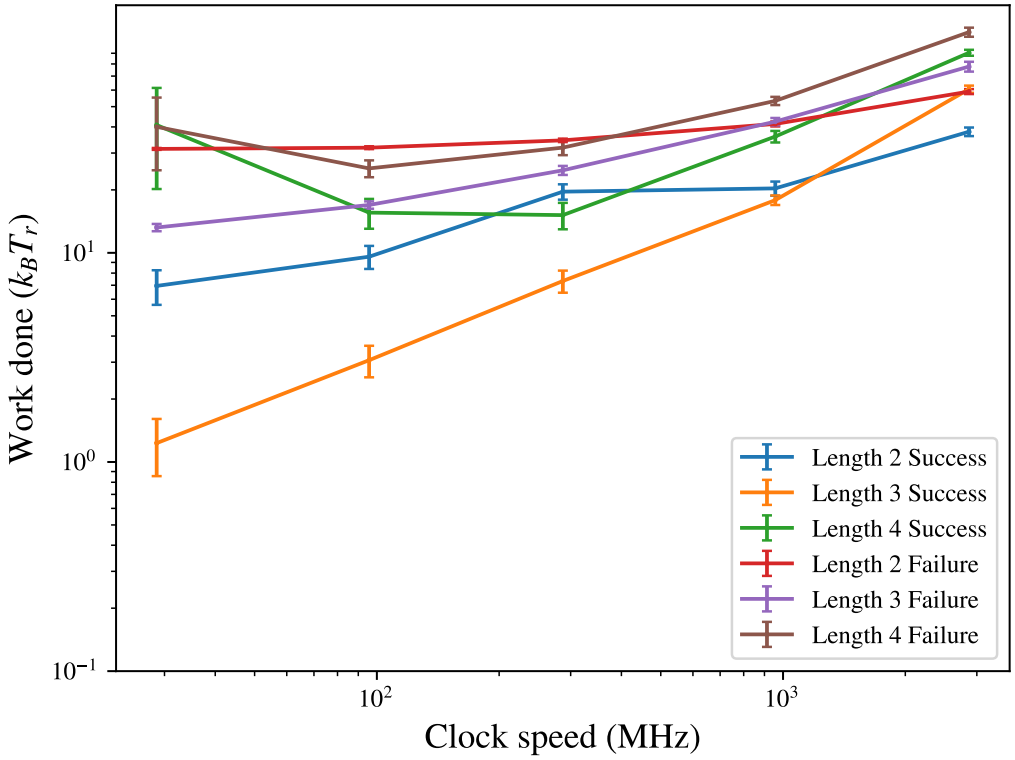


Figure 8.12: Work done against clock speed for different chain lengths in NAND gates driven via the pipelining protocol for both failed and successful reverse pipelining processes.

Table 8.2: Failure rate of the pipelined NAND circuits of length n =2, 3 and 4 relative to clock speed. A reverse pipelining process is defined as a failure if the last NAND gate in the chain is forced into '1' state by its signal booster during the second phase of the reverse pipelining process. This occurs at $t^* = 9/10$, $10/12$ and $11/14$ for $n = 2$, 3 and 4, respectively.

Clock speed (MHz)	Failure rate		
	$n = 2$	$n = 3$	$n = 4$
2870	0.958	0.812	0.576
957	0.924	0.833	0.646
287	0.757	0.944	0.757
95.7	0.424	0.896	0.66
28.7	0.056	0.799	0.486

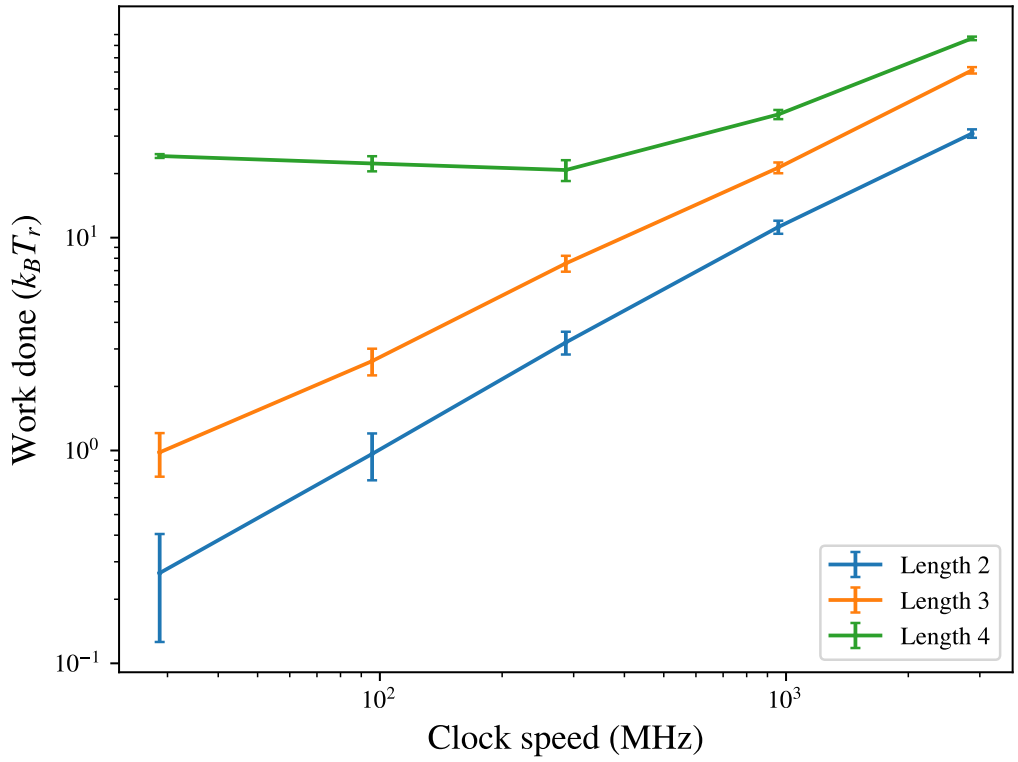


Figure 8.13: Work done against clock speed for different chain lengths in NAND gates driven via the pipelining protocol where the final NAND gate is chained to a pseudo-register. Each data point corresponds to an average over 144 simulations.

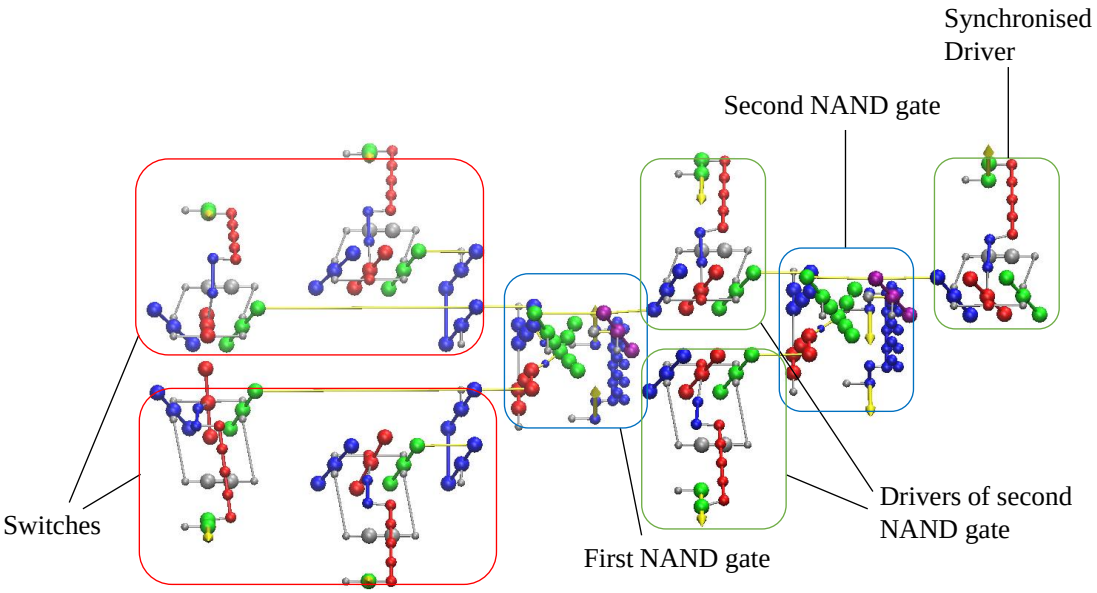


Figure 8.14: Illustration of the pipelining protocol for a system containing two gates modified such that the final gate is chained to a pseudo-register.

Table 8.3: Results obtained from error-rate simulations at T_r (298 K) of the pseudo-register equipped two, three and four chained NAND gates driven via the pipelining protocol at 2.87 GHz. 6000 simulations were run for each data point. Similar results are obtained in the absence of a pseudo-register.

Chain Length	Gate number	1^{st} phase error	2^{nd} phase error
2	1	0	0
2	2	0	0
3	1	0	0
3	2	0	0
3	3	0	5.11×10^{-3}
4	1	0	0
4	2	0	0
4	3	0	4.00×10^{-4}
4	4	0	0.0339

8.3.1 The modified driver

In order to reduce the error rate of the pipelined circuit, a new driver was designed with a different clutch mechanism (Fig. 8.15). The new driver design incorporates two major modifications: a pair of decouplers which are connected via tethers to the input and output bits, and a pair of signal boosters similar to those used in the NAND gates. The purpose of the decouplers is to prevent information from the upstream bits from leaking through to the downstream bits by applying a pulling force through the tethers (coloured red in Fig. 8.15), forcing the bit to which it is connected into a neutral position. Decouplers and signal boosters are required for both the input and output bits due to the fact that the circuit must be operated in both forward and reverse in order to demonstrate reversibility. The decouplers for the input and output bits are offset by 120° and the signal boosters by 60°.

Fig. 8.16 illustrates the operation of the modified driver. In this example, the values of t^* correspond to fractions of exactly one clock cycle. Intuitively, in order to maximise the probability of error-free information transduction, the point at which the driver reaches maximum engagement ($t^* = 0.5$) should correspond to the midpoint

of the phases of the gates chained to its input and output bits. Therefore, given the 90° phase difference between an upstream and downstream gate, the peak engagement of the upstream gate occurs 45° before the peak engagement of the driver at $t^* = 3/8$; correspondingly, the downstream gate reaches peak engagement 45° after the driver does so at $t^* = 5/8$.

Before the upstream gate reaches peak engagement, the decoupler for the output bit engages at $t^* = 1/3$. This forces the output bit into an approximately neutral state and thus improves the ability of the clutch to prevent information from flowing from the downstream gate to the upstream gate or vice versa when not desired. As previously stated, the upstream gate reaches peak engagement at $t^* = 3/8$, followed by the signal booster of the input bit at $t^* = 5/12$, the driver itself at $t^* = 1/2$. By this time the decoupler for the output bit has become sufficiently disengaged that it no longer exerts a significant force on the bit, allowing the signal booster of the output bit to engage fully at $t^* = 7/12$, finally transducing information to the downstream gate which reaches peak engagement at $t^* = 5/8$. This process allows information to flow smoothly from the upstream gate to the downstream gate. At $t^* = 2/3$, the decoupler for the input bit becomes fully engaged. This process, which grants the driver symmetry regardless of the direction the external dipole rotates, is unnecessary for the transduction of information in the forward direction; however, it is critical component of the reversible pipelining process.

Fig. 8.17 illustrates the four-gate pipelined system with the modified drivers replacing all drivers that connect two NAND gates. Drivers which connect fixed input bits to gates and the pseudo-register have been left unchanged. The results of error rate and reversibility simulations run on the system are recorded in Table 8.4 and Fig. 8.18. Based on the high temperature needed to induce errors, it can be concluded that the error rate of the system with modified drivers is far lower than that of the unmodified system at room temperature. In addition, as the work done by the exter-

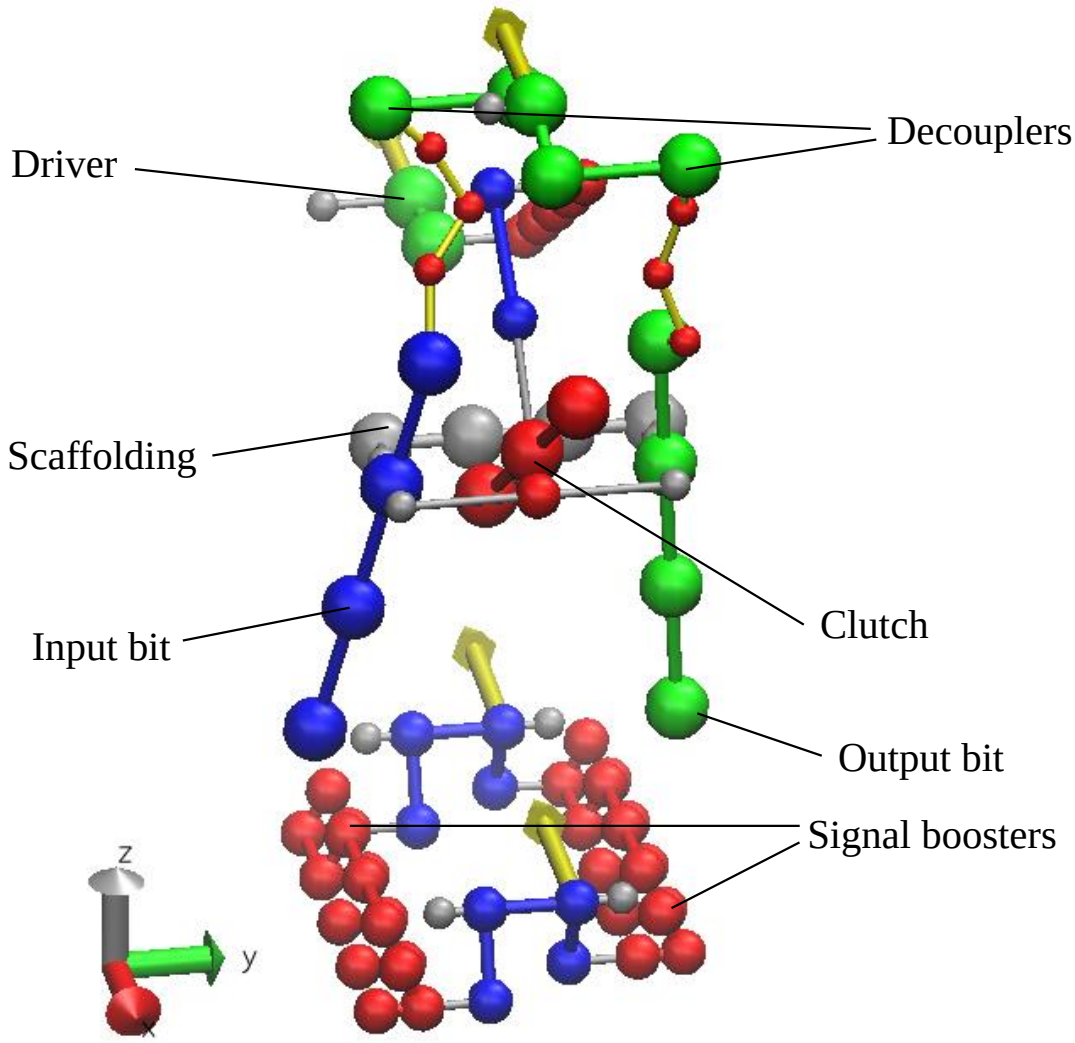


Figure 8.15: The modified driver used to lower the error rate in order to demonstrate reversible pipelining.

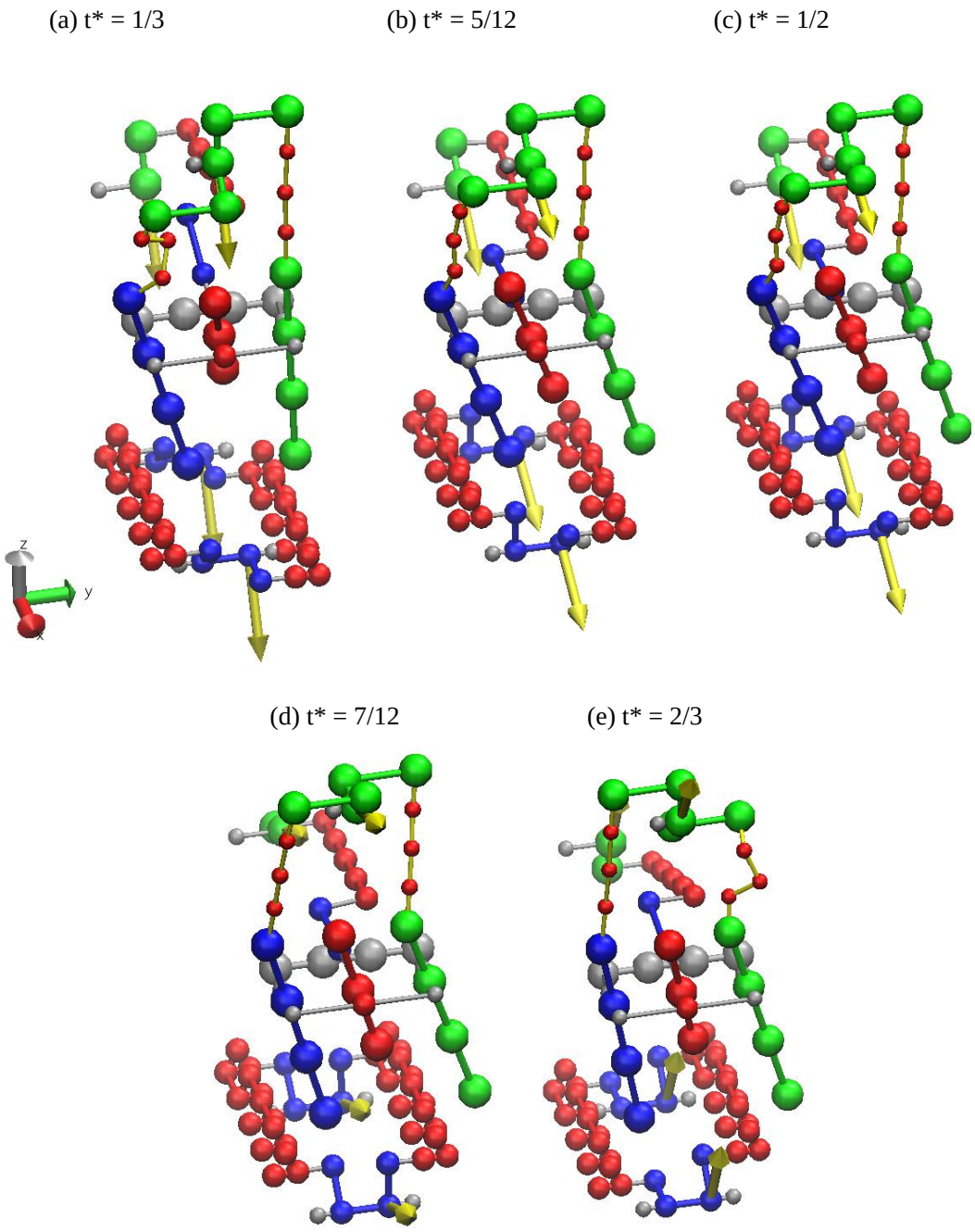


Figure 8.16: The modified driver at five critical values of t^* during the process of transducing the state '1' from the input bit to the output bit.

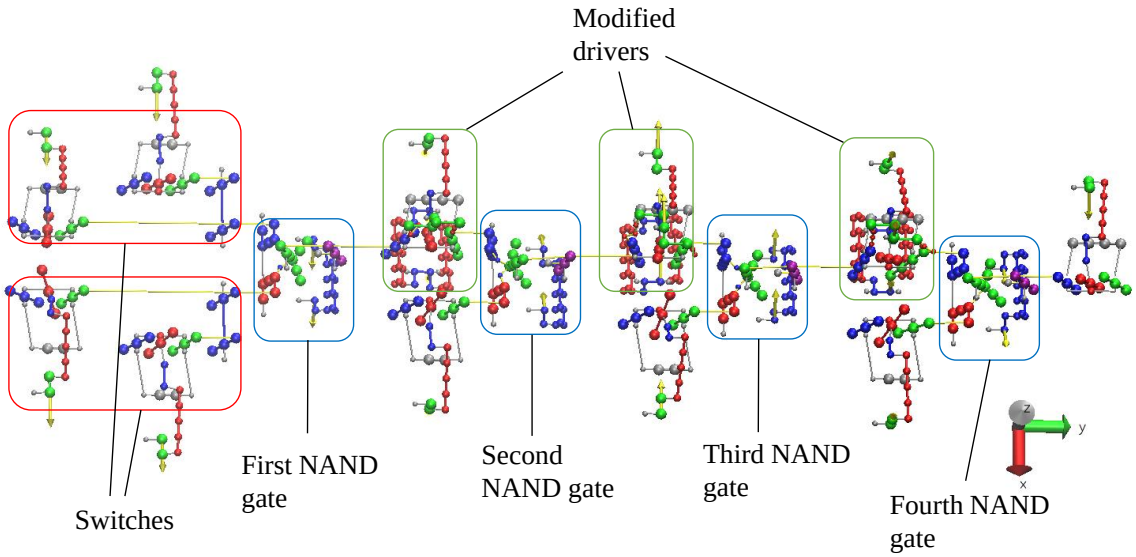


Figure 8.17: The four-gate pipelined system with all drivers between two NAND gates having being replaced by modified drivers.

Table 8.4: Results obtained from error-rate simulations at $6T_r$ (1788 K) of four chained NAND gates connected by modified drivers at 2.87 GHz. 1200 simulations were run for each data point. No errors were detected at simulations run at $T \leq 3T_r$

Gate number	1^{st} phase error	2^{nd} phase error
0	7.50×10^{-3}	0
1	2.50×10^{-3}	1.67×10^{-3}
2	3.33×10^{-3}	7.50×10^{-3}
3	5.83×10^{-3}	1.00×10^{-2}

nal dipoles of the modified system is approximately proportional to clock speed for all chain lengths, it can be concluded that this system converges toward thermodynamic reversibility in the limit of clock speeds tested.

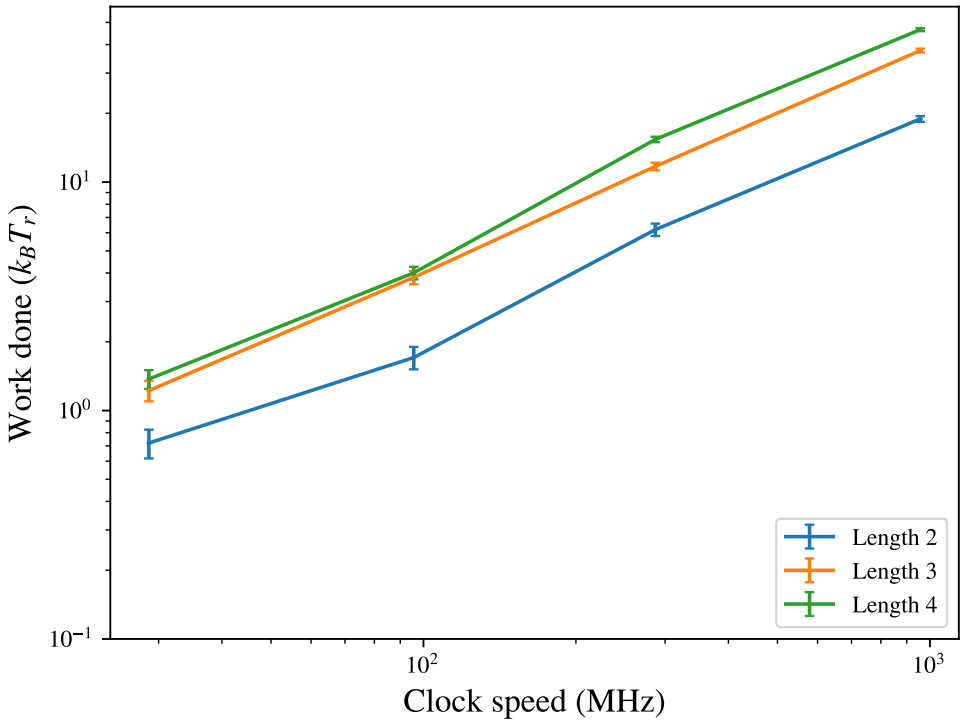


Figure 8.18: Work done against clock speed for the pipelined system with two, three and four chained NAND gates with the modified driver. Each data point corresponds to an average over 96 simulations.

Chapter 9

Optimisation of Logic Gates via a Genetic Algorithm

Thus far, every circuit component outlined in this thesis was designed manually by the author, with no algorithmic assistance. The process of designing a molecular mechanical device by hand is time-consuming and complex, particularly as human mechanical intuition is generally far less effective at designing nanoscale devices intended to operate in thermally noisy stochastic environments than it is at designing macroscopic machines that adhere to purely deterministic Newtonian mechanics.

The complexities of designing and optimising complex mechanical systems in a thermally noisy environment by hand may be overcome with optimisation algorithms. The choice of optimisation algorithm must be informed by the difficulty of calculating the cost function, which will include error rate and efficiency for a logic gate. In order to calculate error rate and efficiency in a thermally noisy system, it is necessary to run a large number of simulations to obtain an accurate result. This places a heavy computational burden on gradient descent algorithms, as the gradient cannot be found by symbolic or automatic differentiation and must be empirically estimated by directly calculating the cost function averaged over multiple simulations. As this

estimate must be calculated for each degree of freedom, such methods would result in a large computational cost. Furthermore, as the systems to be optimised are thermally noisy, two systems which are identical or which differ only infinitesimally may yield very different costs if the number of simulations over which error rate is relatively small. Thus, attempting to empirically estimate the gradient may result in large inaccuracies due to effects of thermal noise unless an impractical number of simulations are run.

Fortunately, there exists a class of optimisation meta-heuristics which operate entirely over discrete variables and thus do not require calculating the gradient of the cost function. Among these meta-heuristics are the genetic algorithms, which currently find widespread use in the optimisation of logic gates and circuits [81, 82, 83], as well as mechanical systems [48, 84, 85]. Genetic algorithms represent a given system as a collection of genes which are subject to random mutations to create a number of distinct alleles. The combinations of different alleles affect the fitness of the system they represent, thus generating a population of individuals with varying levels of fitness. Subsequently, the fittest individuals are selected and the alleles of the next generation are extracted from the gene pool of these fittest individuals in a process analogous to biological reproduction. This process is then repeated until a desired level of fitness is reached or the level of fitness converges.

In this chapter, I shall outline the process of optimising various molecular mechanical circuit elements via a genetic algorithm ("Alabama") which I have written specifically for this task. A link to the code may be found in Appendix B. I also demonstrate the successful optimisation of the driver and winged NAND gate and investigate the effectiveness of high-temperature optimisation of this system.

9.1 The Genetic Algorithm

Two distinct gene types were used in each structure; henceforth referred to as particle-type genes and interaction-type genes. Particle-type genes code for the position of a single particle relative to the centre of the rigid body of which they are a part. Mutations to these genes alter the relative position of the particle in at least one of the Cartesian axes by a fixed constant of either 0.01, 0.02 or 0.03 nm, with an equal chance for each to occur should a mutation be triggered. Interaction-type genes code for the equilibrium value (either x_i for bonding potentials or θ_i for angular potentials) of a single interaction potential, but do not affect the harmonic constant k of the potential. Mutations to these genes alter the equilibrium value of the interaction potential by 2, 4 or 6% in either direction, with an equal chance for each to occur should a mutation be triggered.

The genetic diversity of each offspring was increased via the implementation of uniform crossover [86]. From each generation, the fittest 25% of individuals were selected. For each offspring in the subsequent generation, two parents were randomly selected from this pool, and one allele from each parent was randomly selected to be passed on to the offspring. In addition, an elitism heuristic was also implemented, where the fittest 5% of individuals from each generation were copied unchanged into the next generation.

The cost function, which is the negative of the fitness function, consists of a linear combination of several distinct components. The first component is a threshold for a given observable (typically an angle) such that a cost is imposed when the angle of the observable exceeds a given threshold as quantified by Eq. 9.1:

$$F_i = \sum_{j=a}^{b} k_i[\max(\theta_j - \theta_0, 0)]^2/(b-a), \tag{9.1}$$

where a and b are the limits of the range of the trajectory for which the cost function

is calculated, θ_j is the value of the observable in degrees at the j^{th} frame of the trajectory, θ_0 is the threshold value in degrees, k_i is the weight of the i^{th} member of this component of the overall cost function. Analogously, (Eq. 9.2) describes the case where a cost is imposed if the observable is below a given threshold:

$$G_i = \sum_{j=a}^{b} k_i[\min(\theta_j - \theta_0, 0)]^2/(b-a). \tag{9.2}$$

Both of these functions will henceforth be referred to as observable threshold functions in the remainder of this chapter. The second component, termed the efficiency cost function, imposes a cost for the work done over the entire trajectory beyond a given threshold, and is intended to penalise less thermodynamically efficient systems:

$$H_i = k_i[\max(W - W_0, 0)], \tag{9.3}$$

where W is the work done over the whole process, W_0 the threshold and k_i the weight. The third component, termed the potential cost function, is quantified by Eq. 9.4:

$$J_i = k_i(V_{\text{initial}} - V_{\text{final}})^2, \tag{9.4}$$

where V_{initial} is the potential energy of the initial configuration in a given trajectory, V_f the potential energy of the final configuration and k_i the weight. This component imposes a cost for any difference between the potential energies of the initial and final configurations for a given trajectory, and is intended to penalise situations in which the final configuration is much lower in energy than the initial configuration. In such cases, Eq. 9.3 could return a misleadingly low cost due to a portion of the potential energy of the system having been converted into work. Eq. 9.4 is intended to heavily penalise such systems, to a greater extent than Eq. 9.3 rewards them.

Due to the fact that all systems tested are subject to thermal noise, there is a large

random component in the observables penalised by the cost functions. Therefore, the average value of each component across multiple simulations (typically 10) is used to calculate the overall fitness.

9.2 Optimisation of the Driver

Optimising the error rate and efficiency of the modified driver poses unique challenges due to the fact that it is not possible to accurately calculate the error rate without running a large number of simulations. In order to create a fitness function which allows for the evolution of a system with a lower error rate than the initial seed, a temporary artificial potential opposing the desired output state was applied to the modified driver during the simulations used to calculate the fitness of a given mutant system. The effect of this potential is to increase the chance that the system will encounter an error during a small handful of distinct runs, forcing the system to evolve in a direction that minimises error rate to a far greater extent than would be the case without such a potential.

Eq. 9.5 describes the cost function used to optimise the modified driver. It is a linear combination of four distinct sub-functions, $S_{0,+}$, $S_{0,-}$, $S_{1,+}$ and $S_{1,-}$ each of which corresponds to a simulation with distinct initial conditions and measured observables. Note that the more positive this sum is, the higher the cost and the lower the fitness.

$$T = S_{0,+} + S_{0,-} + S_{1,+} + S_{1,-}\,, \tag{9.5}$$

$$S_{0,+}, S_{0,-} = F_{0,a} + G_{1,a} + G_{2,a} + H + J\,, \tag{9.6}$$

$$S_{1,+}, S_{1,-} = G_{0,b} + F_{1,b} + F_{2,b} + H + J\,. \tag{9.7}$$

$S_{0,+}$ corresponds to a simulation in which the driver is required to transduce infor-

mation from an input bit set to the state '0' to the output bit, while $S_{0,-}$ corresponds to the same process executed in reverse, with information being transduced from the output to the input bit. Henceforth, the bit from which information is to be transduced shall be referred to as the fixed bit, and the bit which receives information as the receiving bit. A time-dependent angular constraint potential (Eq. 9.9) is applied to the fixed bit:

$$V_{\text{fixed}} = k\hat{\mathbf{d}}_{\text{tdp}} \cdot \hat{\mathbf{d}}_{\text{bit}}, \tag{9.8}$$

$$\hat{\mathbf{d}}_{\text{tdp}} = \min(\sin\left[\pi(t^* + 0.125)\right](0, -0.7, 0.7)^T, 0), \tag{9.9}$$

where $k = 496\,\text{kJ/mol}$ and $\hat{\mathbf{d}}_{\text{tdp}}$ is the time-dependent angular constraint potential which is a function of t^*. The potential peaks at $t^* = 0.375$ and declines to zero thereafter, but does not become negative. $\hat{\mathbf{d}}_{\text{bit}}$ refers to the vector corresponding to the slope of the fixed bit.

As previously mentioned, a time-independent potential is applied to the receiving bit for the purposes of deliberately triggering errors, henceforth referred to as an error-triggering potential:

$$V_{\text{etp}} = k\hat{\mathbf{d}}_{\text{etp}} \cdot \hat{\mathbf{d}}_{\text{bit}}, \tag{9.10}$$

$$\hat{\mathbf{d}}_{\text{etp}} = (0, 0.7, 0.7)^T, \tag{9.11}$$

where $k = 99\,\text{kJ/mol}$ and $\hat{\mathbf{d}}_{\text{etp}}$ is the time-dependent angular constraint potential which is a function of t^*.

The first of the observable-threshold cost functions $F_{0,a}$ represents a cost imposed when the receiving bit fails to attain an angle greater than the desired magnitude ($\theta_0 = 115°$) over the period $a = 0.5$ to $b = 0.625$. The angle θ_0 is measured between the direction vector of the principal axis of the green output bit in Fig. 8.15 and the vector $(0, 1, 0)^T$. The second observable-threshold function $G_{1,a}$ represents a cost

imposed when the receiving bit attains an angle greater than the desired magnitude ($\theta_0 = 100°$) over the period $a = 0$ to $b = 0.25$. The third observable-threshold function $G_{2,a}$ is similar, but is active over the period $a = 0.75$ to $b = 1$. The purpose of these last two potentials is to prevent the system from "cheating" by simply adopting a geometry where the receiving bit adopts an angle of the desired magnitude at all times, rather than simply when the fixed bit is constrained. All three cost functions have the same weight $k = 1$ and number of observations $N = (b - a) \times 10^4$. The efficiency cost function H has threshold $W_0 = 0$ and weight $k = 3$, and the potential cost function J has weight $k = 1$.

$S_{1,+}$ and $S_{1,-}$ are analogous to $S_{0,+}$ and $S_{0,-}$, with the only difference that the fixed bit is set to '1' rather than '0', with corresponding changes to the constraint potential (Eq. 9.12), error-triggering potential (Eq. 9.13) and cost functions.

$$\hat{\mathbf{d}}_{\text{tdp}} = \min(\sin\left[2\pi(t^* - 0.125)\right](0, 0.7, 0.7)^T, 0), \tag{9.12}$$

$$\hat{\mathbf{d}}_{\text{etp}} = (0, -0.7, 0.7)^T. \tag{9.13}$$

The observable-theshold functions operate over the same range and serve the same purpose as their '0' counterparts, but are mirrored with respect to the yz-plane. $G_{1,a}$ has a threshold of $\theta_0 = 65°$, while $F_{1,b}$ and $F_{1,c}$ have thresholds of $\theta_0 = 80°$. H and J are identical for all S.

Not all rigid bodies and bonds which compose the driver were modifiable by the genetic algorithm. Only the clutch, driver arms and decouplers and any bonds between these components were converted into genes subject to mutation. Each distinct starting state used to calculate the cost function was simulated ten times with the simulation temperature being $6T_r$; this high temperature was intended to force the system to enter erroneous states it would only rarely enter at room temperature, compensating for the small number of simulations. The genetic algorithm was executed

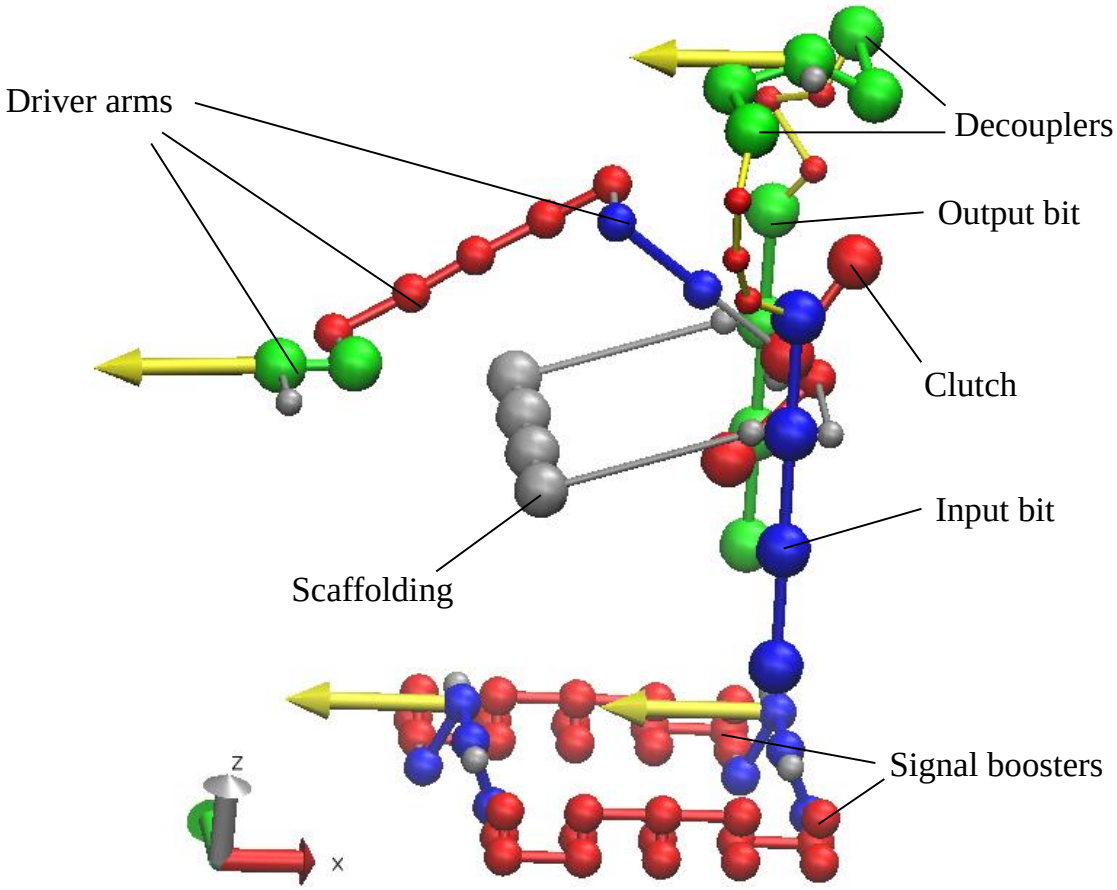


Figure 9.1: The fittest driver obtained after optimsation over 20 generations of the genetic algorithm.

over 20 generations of 48 individual genotypes at a constant mutation rate of 20%. Fig. 9.2 demonstrates the reduction in cost (increase in fitness) over each generation for the fittest individual and the average of the top 25% of each generation. The fittest individual, found in generation 18, is depicted superimposed on the original modified driver in Fig. 9.1.

Overall, the geometry of the fittest individual does not differ significantly from the original parent. Small changes have occurred to the equilibrium angles of the angular potentials between the driver arms, and overall the bond lengths of the tether between the decoupler and bit have increased by roughly 10%. The drivers of the four-gate pipelined system were substituted for their genetically-optimised versions.

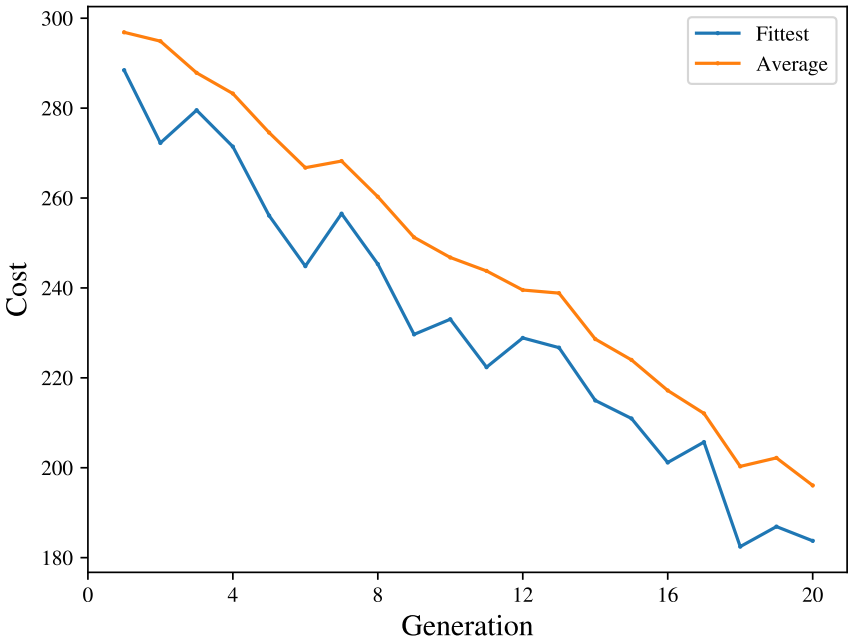


Figure 9.2: Plot of cost over generation for the modified driver system optimised via genetic algorithm.

Error rate and reversibility simulations were run on this augmented system and the results summarised in Table 9.1 and Fig. 8.18.

From these results, it is clear there is a significant improvement in error rate, and the system remains reversible. There is no significant improvement in efficiency; however, this is not unexpected as the main contributors of drag are the NAND gates and switch. Overall, it appears reasonable to conclude that the genetic algorithm with the given parameters is effective at reducing the error rate of this particular system.

9.3 The NAND gate revisited

Application of the genetic algorithm to the winged NAND gate is not as simple as it is for the driver. There are four possible inputs, two possible outputs and twelve

Table 9.1: Results obtained from error-rate simulations at $6T_r$ (1788 K) of four chained NAND gates connected by modified drivers optimised via genetic algorithm at 2.87 GHz. 1200 simulations were run for each data point. No errors were detected at simulations run at $T \leq 4T_r$

Gate number	1^{st} phase error	2^{nd} phase error
0	3.33×10^{-3}	0
1	0	8.33×10^{-4}
2	8.33×10^{-4}	1.67×10^{-3}
3	0	8.33×10^{-4}

possible transitions between distinct states for the NAND gate. The cost function for the NAND gate is accordingly more complex:

$$T = \sum_{j=0}^{12} S_j \,, \tag{9.14}$$

$$S_j = \{S_{00},\ S_{01},\ S_{10}\} \,, \tag{9.15}$$

$$S_{00} = \{S_{0001},\ S_{0010},\ S_{0100},\ S_{1000},\ S_{0110},\ S_{1001}\} \,, \tag{9.16}$$

$$S_{01} = \{S_{0011},\ S_{0111},\ S_{1011}\} \,, \tag{9.17}$$

$$S_{10} = \{S_{1100},\ S_{1101},\ S_{1110}\} \,. \tag{9.18}$$

The total cost function T is composed of a sum of twelve distinct components, each of which corresponds to one of the twelve possible non-trivial transitions. In the notation above, the first two numbers of the subscripts of each component function (for example, '00' for S_{0011}) correspond to the starting states of the input bits (in this case, '00'), and the last two numbers the midpoint states of the input bits ('11'). These twelve transitions can be grouped into three categories S_{00}, S_{01} and S_{10} where the first number is the desired starting state and the second number the desired midpoint state of the output bit. Each of these three categories is represented by a

unique cost function.

$$S_{00} = F_s + F_m + F_e + H + J\,, \tag{9.19}$$

$$S_{01} = F_s + G_m + F_e + H + J\,, \tag{9.20}$$

$$S_{10} = G_s + F_m + G_e + H + J\,. \tag{9.21}$$

The subscript of the observable threshold functions (s, m and e) designate whether the function is active at the start, midpoint or end of the simulation respectively. s-type functions are active over the period $a = 0$ to $b = 0.1$, m-type functions over $a = 0.4$ to $b = 0.6$, and e-type functions over $a = 0.9$ to $b = 1$. The function type G represents a cost imposed when the receiving bit fails to attain an angle greater than ($\theta_0 = 115°$) over the aforementioned periods. The angle θ_0 is measured between the vector represented by the direction vector of the purple output bit in Fig. 9.5 and the vector $(0, 1, 0)^T$. Conversely, F-type functions represent a cost imposed when the receiving bit fails to reach an angle smaller than ($\theta_0 = 65°$) over the period indicated by the subscript. Both F and G have weights of $k = 1$ and number of observations $N = (b - a) \times 10^4$. The efficiency cost function H has threshold $W_0 = 0$ and weight $k = 1$, and the potential cost function J has weight $k = 1$.

In a manner similar to the modified driver, two time-dependent angular constraint error-triggering potentials (Eq. 9.23) are applied to the fixed bit in order to increase the likelihood that an error will occur during the simulation, forcing the gate to adapt to prevent it. The potentials take the following form:

$$V_{se} = k\hat{\mathbf{d}}_{\mathrm{se}} \cdot \hat{\mathbf{d}}_{\mathrm{bit}}, \tag{9.22}$$

$$\hat{\mathbf{d}}_{\mathrm{se}} = \min(\sin\left[2\pi(t^* + 0.25)\right](0, \pm 0.7, 0.7)^T, 0), \tag{9.23}$$

$$V_m = k\hat{\mathbf{d}}_{\mathrm{m}} \cdot \hat{\mathbf{d}}_{\mathrm{bit}}, \tag{9.24}$$

$$\hat{\mathbf{d}}_{\mathrm{m}} = \min(\sin\left[2\pi(t^* - 0.25)\right](0, \pm 0.7, 0.7)^T, 0). \tag{9.25}$$

where $k = 99\text{kJ/mol}$ and $\hat{\mathbf{d}}_{\text{se}}$ and $\hat{\mathbf{d}}_{\text{m}}$ are the time-dependent angular constraint potentials which are functions of t^*. The y-components of $\hat{\mathbf{d}}_{\text{se}}$ and $\hat{\mathbf{d}}_{\text{m}}$ are positive when the desired state of the output bit is '0' (thus opposing it by forcing the bit into the '1' configuration) and negative when the desired state is '1'. $\hat{\mathbf{d}}_{\text{se}}$ reaches peak magnitude at the start and end of the simulation while $\hat{\mathbf{d}}_{\text{m}}$ reaches peak magnitude at the midpoint. $\hat{\mathbf{d}}_{\text{bit}}$ refers to the vector corresponding to the slope of the output bit. All rigid bodies and bonds of the NAND gate were subject to mutation except for the output bit and the scaffold attachment points. Each distinct starting state used to calculate the cost function was simulated ten times with the simulation temperature being T_r.

The genetic algorithm was executed over 20 generations of 48 individual genotypes at a constant mutation rate of 20%. Fig. 9.4 demonstrates the reduction in cost (increase in fitness) over each generation for the fittest individual and the average of the top 25% of each generation. The fittest individual, found in the final generation, is depicted superimposed on the original NAND gate in Fig. 9.3. Error rate simulations at $4T_r$ were run on the optimised system. Overall, there does not appear to be a significant improvement over the original NAND gate (Table 5.2). Given that the original NAND gate has a low error rate to begin with, intuitively, it appears probable that the genetic algorithm would be less effective on this system compared to one with a higher initial error rate.

9.3.1 The modified NAND gate

In order to test the aforementioned hypothesis about the greater relative effectiveness of the genetic algorithm in systems with a higher initial error rate, a modification of the original NAND gate design was subjected to optimisation via the genetic algorithm. As previously mentioned in chapter 2, particles directly bound to the scaffold are constrained by a harmonic potential centred on their initial position instead of a

Table 9.2: Results obtained from error-rate simulations at $4T_r$ (894 K) of the optimised NAND gate at 1.44 GHz.. No errors were detected over 12 000 simulations at $T \leq 3T_r$.

Transition	Error (Start)	Error (Mid)	Error (End)
00→01	0	0	0
00→10	0	0	0
00→11	0	0	0
01→00	0	0	0
01→10	0	0	0
01→11	0	0	8.33×10^{-5}
10→00	0	0	0
10→01	0	1.67×10^{-4}	3.33×10^{-4}
10→11	0	1.42×10^{-3}	0
11→00	0	0	0
11→01	0	8.33×10^{-5}	6.67×10^{-4}
11→10	0	0	0

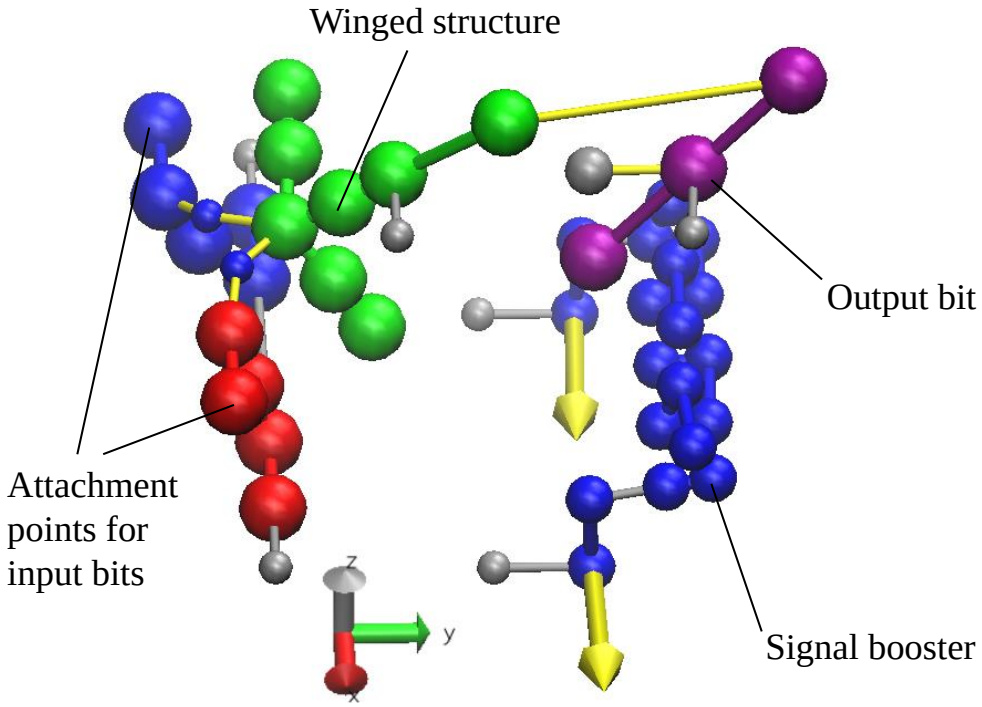


Figure 9.3: The fittest NAND gate obtained after optimisation over 20 generations of the genetic algorithm.

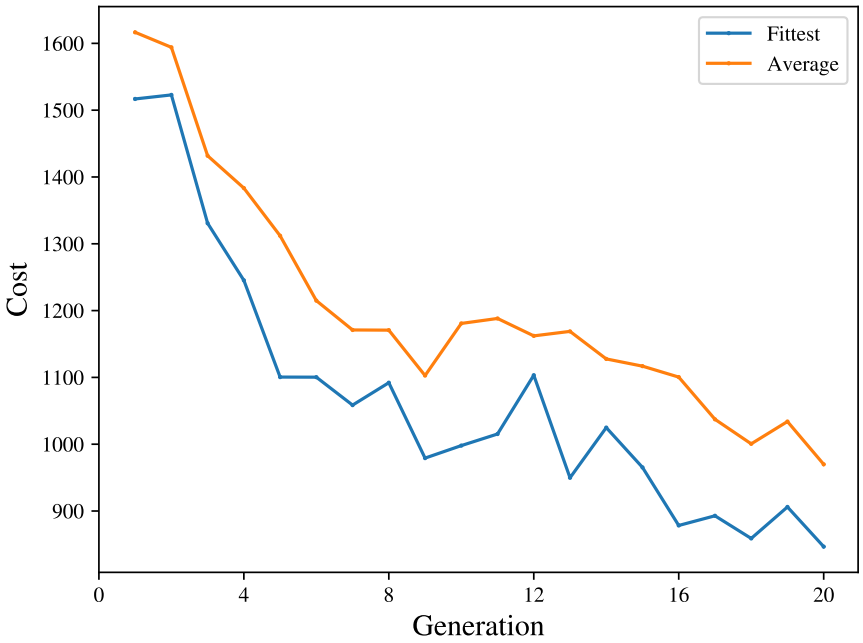


Figure 9.4: Plot of cost over generation for the NAND gate optimised via genetic algorithm.

standard bonding interaction in order to compensate for the lack of angular potentials between rigid bodies bonded to the scaffold and any additional atoms which would be bonded to the scaffold atom as would be the case in a realistic system. A modification of the original NAND gate in which some of these potentials were removed was designed (Fig. 9.5).

The harmonic potentials were removed from the scaffolding of the attachment point for the input bits, the winged structure and the output bit, with only the signal booster retaining the harmonic potentials. The reason for this exception was that the removal of the potentials would leave the signal booster able to rotate about the x-axis since additional scaffolding points cannot be placed without obstructing the path of the signal booster or introducing additional rigid bodies.

The effect of these modifications on the performance of the NAND gate is highly significant, as they greatly increase the flexibility of the structure, particularly the at-

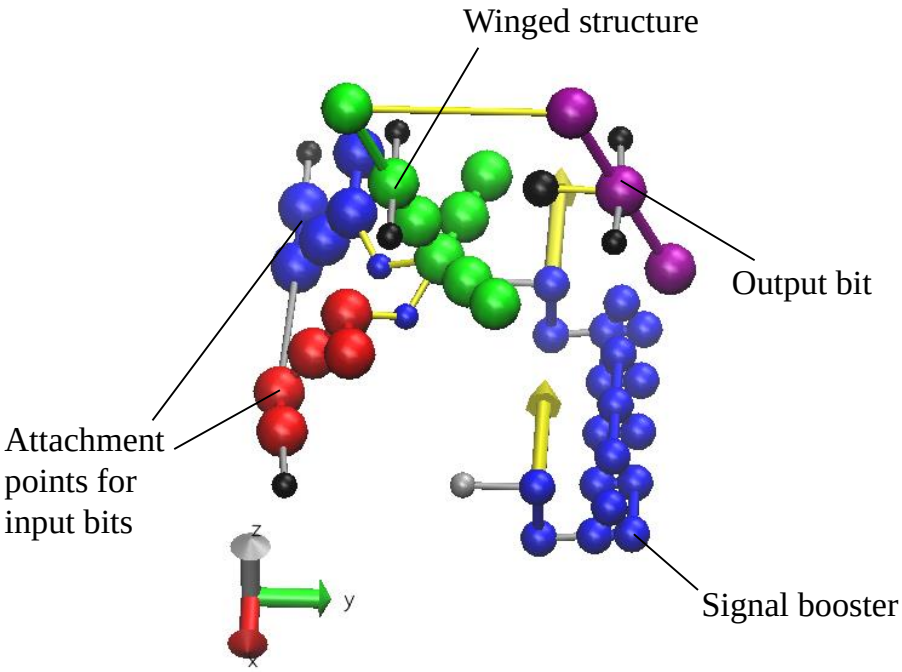


Figure 9.5: A modification of the NAND gate in which several constraining harmonic potentials were removed. The scaffold locations where such potentials were removed are coloured black instead of silver.

tachment points for the input bits. In order to quantify the effect of the modifications on the performance of the gate, two pairs of switches were connected to it and error rate simulations were run between all possible transitions (Fig. 9.6). This protocol is identical to one used to determine the error rate of switching between defined states described in Chapter 5. The results of these simulations are summarised in Table 9.3.

From these results, it is clear that the performance of the modified gate is significantly inferior to the original, which did not produce any erroneous results during any transition over 60 000 simulations at the same temperature (596K). This high error rate is likely due to the aforementioned increased flexibility of critical components of the NAND gate.

The set of parameters used to optimise the modified NAND gate was identical to that used to optimise the original NAND gate. Fig. 9.10 demonstrates the reduction

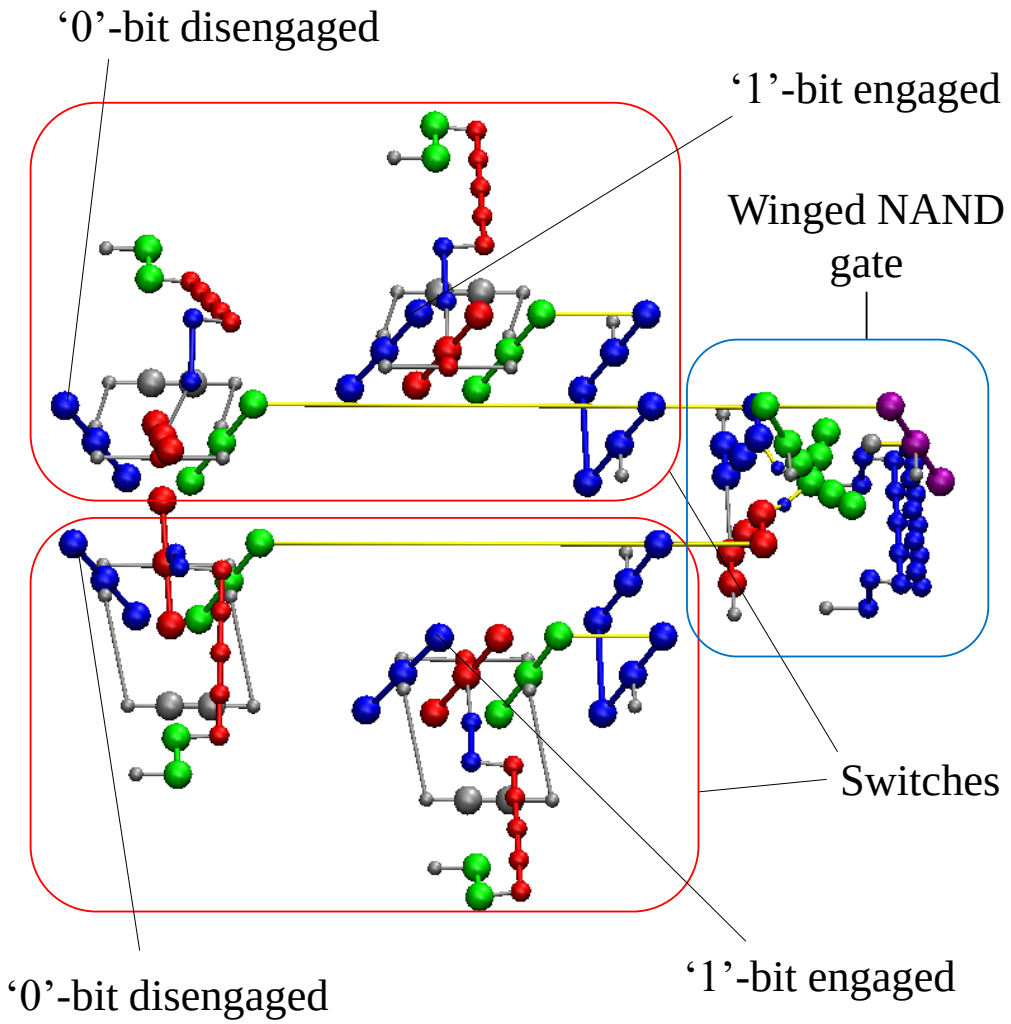


Figure 9.6: The modified NAND gate connected to a pair of switches.

Table 9.3: Results obtained from error-rate simulations at $2T_r$ (596 K) of the modified NAND gate at 2.87 GHz. 12 000 simulations were run for each data point.

Transition	Error (Start)	Error (Mid)	Error (End)
00→01	0	0	0
00→10	0	0	0
00→11	0	0	0
01→00	0	0	0
01→10	8.33×10^{-5}	8.33×10^{-5}	1.67×10^{-4}
01→11	4.17×10^{-4}	1.25×10^{-3}	8.33×10^{-5}
10→00	0	8.33×10^{-5}	0
10→01	1.67×10^{-4}	4.25×10^{-3}	2.67×10^{-3}
10→11	5.83×10^{-4}	1.46×10^{-2}	0
11→00	0	0	0
11→01	0	0	1.21×10^{-2}
11→10	0	0	1.33×10^{-3}

in cost (increase in fitness) over each generation for the fittest individual and the average of the top 25% of each generation. The fittest individual, found in the final generation, is depicted superimposed on the original NAND gate in Fig. 9.7.

The optimised NAND gate differs from the original design in several respects. Firstly, the bond connecting the two attachment points for the input bits has shortened, resulting in a stiffening of the structure. The angle between the two "wings" of the winged structure has become more acute, and the equilibrium angle between the point of the wedge of the signal booster and its arms have increased. Error rate simulations at both $2T_r$ and $3T_r$ were run on the optimised system. No errors were detected over 12 000 simulations at $2T_r$, indicating a large improvement over the unoptimised system. The results of thermodynamic reversibility simulations show that the system approximately tends towards thermodynamic reversibility across all possible unique transitions in the limit of clock speeds tested, albeit with significant differences in efficiency between individual transitions. These results are in line with the intuitive hypothesis that systems with a higher error rate benefit to a greater extent from optimisation via the genetic algorithm than systems that have a lower

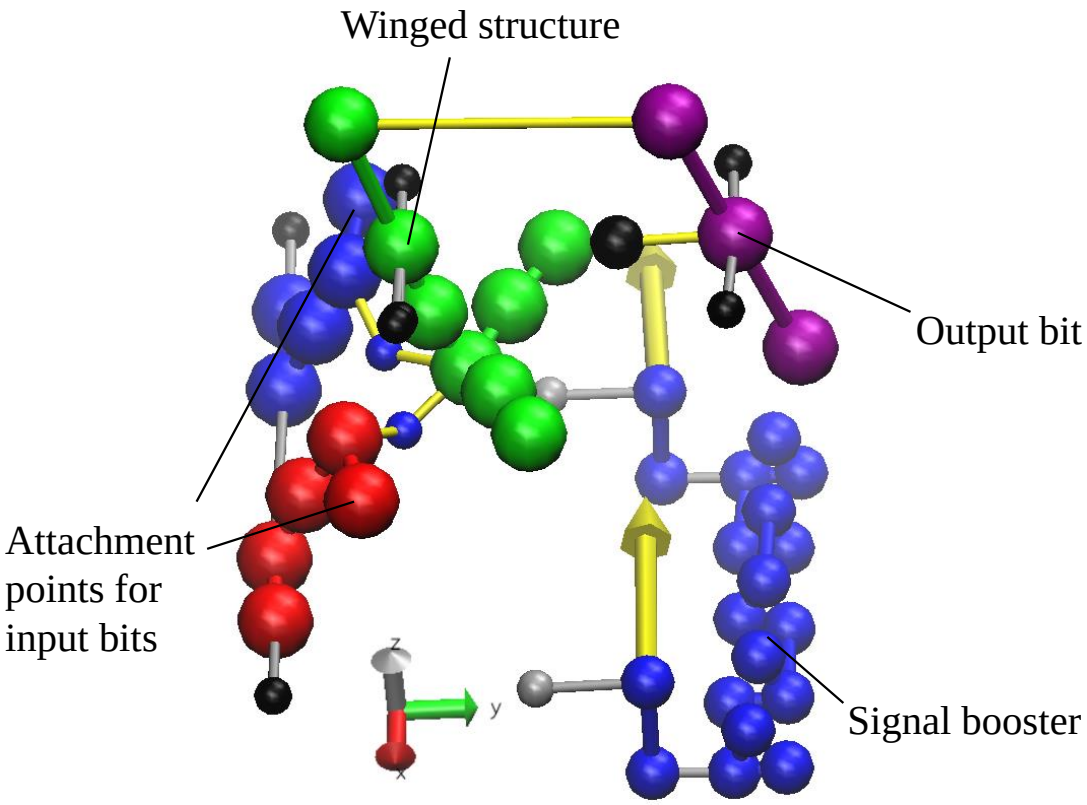


Figure 9.7: The fittest modified NAND gate obtained after optimisation over 20 generations of the genetic algorithm.

error rate to begin with.

9.3.2 High temperature optimisation

Optimisation of the NAND gate was also carried out at high temperature ($6T_r$.) The parameters of the cost function used were identical; however, the room temperature-optimised gate was used as an initial seed instead of the original. Fig. 9.10 demonstrates the reduction in cost (increase in fitness) over each generation for the fittest individual and the average of the top 25% of each generation. The fittest individual, found in generation 15, is depicted superimposed on the original modified driver in Fig. 9.11.

Error rate simulations were run on this new gate at both $6T_r$ and $2T_r$. This high

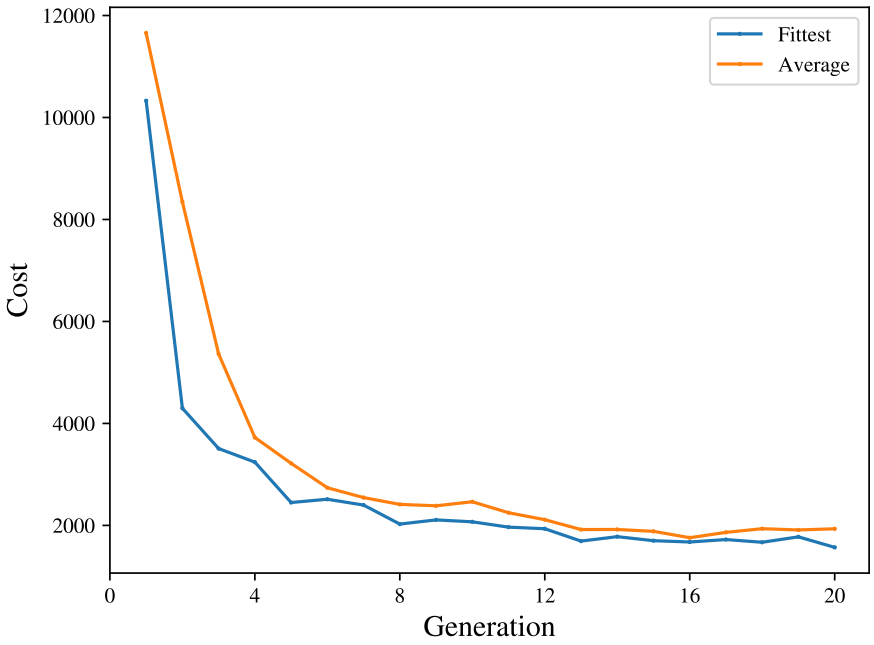


Figure 9.8: Plot of cost over generation for the NAND gate optimised via genetic algorithm.

Table 9.4: Results obtained from error-rate simulations at $3T_r$ (894 K) of the room temperature-optimised NAND gate at 2.87 GHz. No errors were detected over 12 000 simulations at 596 K.

Transition	Error (Start)	Error (Mid)	Error (End)
00→01	0	0	0
00→10	0	0	0
00→11	0	0	0
01→00	0	0	0
01→10	0	0	0
01→11	0	0	3.33×10^{-4}
10→00	0	0	0
10→01	0	0	0
10→11	0	0	5.17×10^{-3}
11→00	0	1.67×10^{-4}	0
11→01	0	0	0
11→10	0	2.17×10^{-3}	0

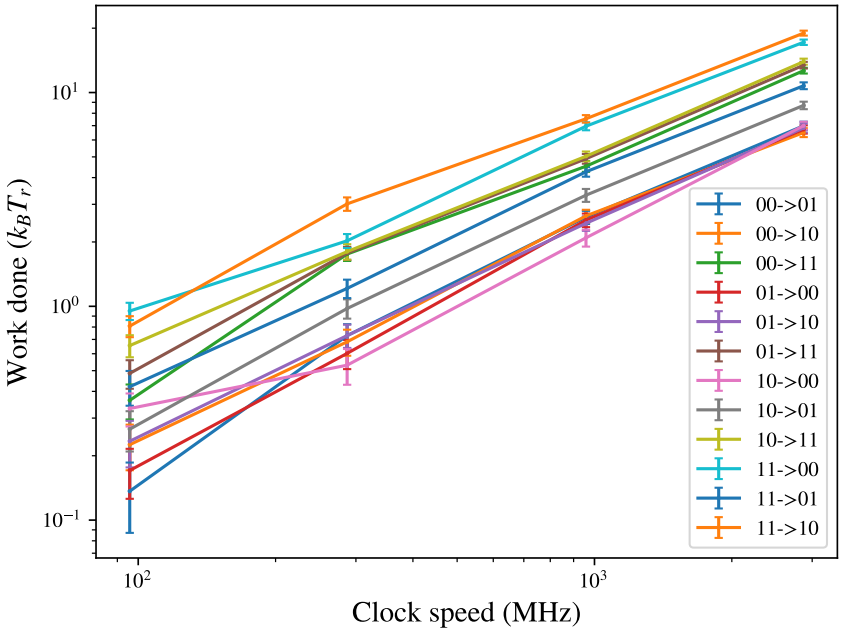


Figure 9.9: Work done against clock speed for the optimised NAND gate during thermodynamic reversibility simulations of transitions from one defined state to another. Each data point corresponds to an average over 96 simulations.

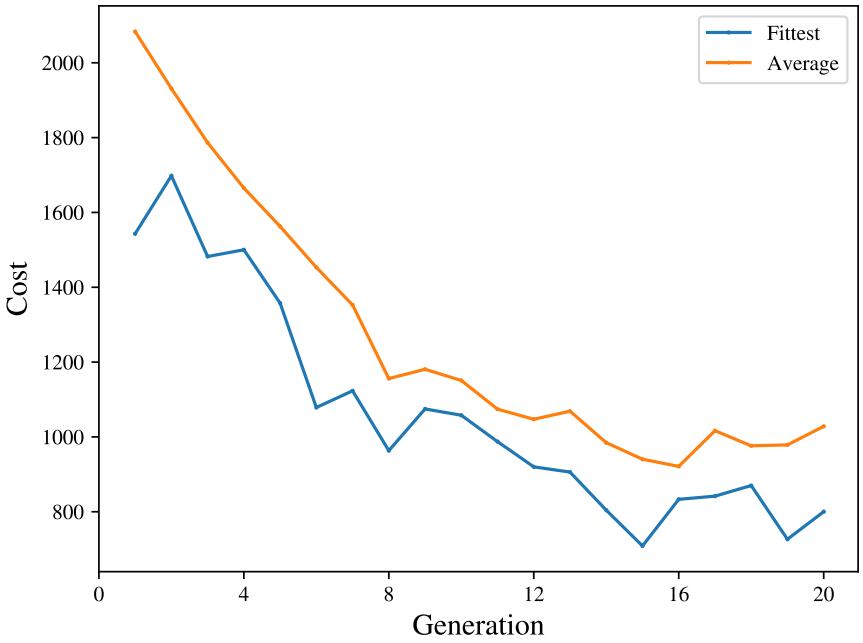


Figure 9.10: Plot of cost over generation for the NAND gate optimised via genetic algorithm at high temperature.

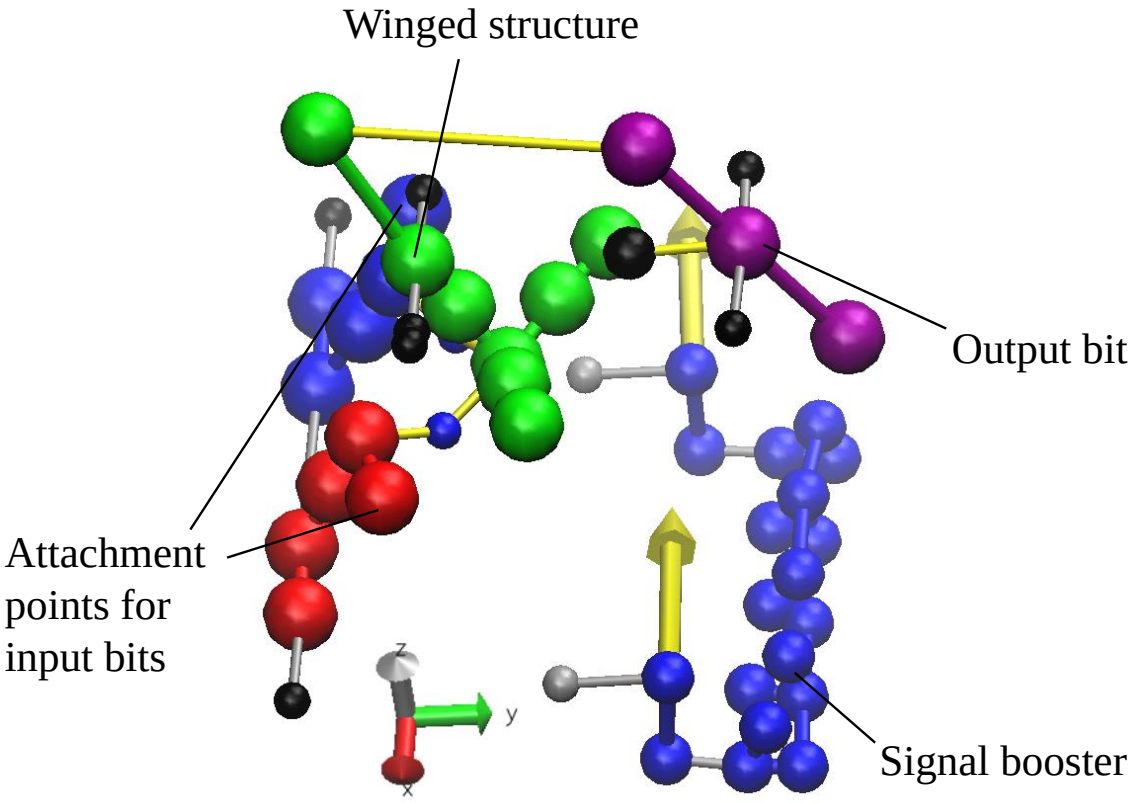


Figure 9.11: The fittest modified NAND gate obtained after high-temperature optimisation over 20 generations of the genetic algorithm.

Table 9.5: Results obtained from error-rate simulations at $6T_r$ (1788 K) of the room temperature-optimised NAND gate at 2.87 GHz.

Transition	Error (Start)	Error (Mid)	Error (End)
00→01	0	0	0
00→10	0	0	0
00→11	0	0	5.83×10^{-3}
01→00	0	0	0
01→10	0	1.67×10^{-3}	0
01→11	0	0	1.67×10^{-3}
10→00	0	0	0
10→01	0	0	0
10→11	8.33×10^{-4}	8.33×10^{-4}	2.12×10^{-1}
11→00	5.00×10^{-3}	1.08×10^{-2}	0
11→01	1.25×10^{-3}	2.42×10^{-3}	6.67×10^{-4}
11→10	3.75×10^{-2}	6.92×10^{-2}	8.33×10^{-4}

temperature-optimised driver performs significantly better at $6T_r$ than the room-temperature optimised driver in terms of the worst-case error rate (0.0142 vs 0.212 respectively). However, its performance is vastly inferior at $2T_r$ with worst-case error rates of 0.479 vs 0; indeed, it is inferior to the original NAND gate in this regard. This demonstrates that simply increasing the temperature of the simulations used to calculate the cost function in the hope it will force low-frequency erroneous states to manifest during the small number of simulations is not a panacea and must be used judiciously, as a high temperature-optimised system may become over-adapted to that temperature.

Table 9.6: Results obtained from error-rate simulations at $6T_r$ (1788 K) of the high temperature-optimised NAND gate at 2.87 GHz.

Transition	Error (Start)	Error (Mid)	Error (End)
00→01	0	0	8.33×10^{-4}
00→10	0	0	0
00→11	0	0	1.67×10^{-3}
01→00	0	0	0
01→10	0	8.33×10^{-4}	0
01→11	1.42×10^{-2}	8.33×10^{-4}	9.17×10^{-3}
10→00	0	0	0
10→01	0	0	1.25×10^{-2}
10→11	0	5.83×10^{-3}	0
11→00	0	8.33×10^{-4}	0
11→01	0	3.33×10^{-3}	3.83×10^{-2}
11→10	0	0	8.33×10^{-4}

Table 9.7: Results obtained from error-rate simulations at $2T_r$ (596 K) of the high temperature-optimised NAND gate at 2.87 GHz.

Transition	Error (Start)	Error (Mid)	Error (End)
00→01	0	0	0
00→10	0	0	0
00→11	0	0	1.67×10^{-4}
01→00	0	0	0
01→10	0	0	0
01→11	0	0	8.33×10^{-5}
10→00	0	0	0
10→01	0	8.00×10^{-3}	1.24×10^{-2}
10→11	4.79×10^{-1}	3.83×10^{-1}	0
11→00	2.03×10^{-2}	6.67×10^{-4}	0
11→01	0	0	4.66×10^{-1}
11→10	3.90×10^{-2}	8.33×10^{-5}	0

Chapter 10

Conclusion

Despite being subject to a large number of design constraints, molecular mechanical logic gates potentially allow for the design of compact reversible logical systems capable of functioning efficiently at room temperature and under standard atmospheric conditions. In Chapter 3, I present a general framework for constructing thermodynamically reversible combinatorial circuits, which allows for the investigation of thermodynamic phenomena in logical systems that are not purely abstract. In Chapter 4, I demonstrate Landauer's principle, a molecular mechanical Szilard engine and thus demonstrate a fundamental physical principle while also verifying the thermodynamic self-consistency of the Langevin thermostat. In Chapter 5, I demonstrate an efficient novel input-preserving molecular mechanical NAND gate design which is thermodynamically reversible and has a low error rate. Although the experimental synthesis of a molecular-scale system based on this framework is likely to remain impractical in the near future, my work nevertheless demonstrates the theoretical soundness of using molecular mechanical logic gates to construct thermodynamically reversible circuits within this framework.

Using the aforementioned framework, I demonstrate the importance of staggering the phases of chained gates to maintain a low error rate when comparing directly-

chained and phase-chained protocols in Chapter 6. I also demonstrate a thermodynamically reversible phase-chained circuit of up to four NAND gates. In Chapter 7, I demonstrate the use of signal boosting to reduce the error rate in more complex systems such as the half-adder. In comparing the performance of the boosted and unboosted systems, I find a trade-off between the complexity of the control protocol that drives a computation, and the accuracy of that computation. Such a trade-off is only visible when a concrete — albeit idealised — computational architecture is modelled.

Subsequently, in Chapter 8, I demonstrate the utility of pipelining in providing a method of greatly increasing the throughput and reducing the number of external controls of a reversible combinatorial circuit; however, these advantages are offset by a number of limitations. I show the thermodynamic inefficiency caused by logically irreversibility in pipelined circuits by comparing the thermodynamic efficiency of the same circuit driven via pipelining and phased chaining, but that it is also possible to convert a pipelined circuit into a logically reversible form even in the absence of persistent memory storage via the use of a pseudo-register, at least for simple systems. I show that it is possible to use this technique to render relatively complex pipelined circuits such as the four-NAND-gate pipelined system thermodynamically reversible, albeit with modifications to the driver and clutching mechanism to reduce the error rate.

Finally, in Chapter 9, I demonstrate the ability of a genetic algorithm to optimise key components of the molecular mechanical circuits, including the driver and winged NAND gate, using the novel technique of an artificial error-triggering potential to reduce the number of simulations required to compute the fitness function. I also investigate the effects of high temperature on the optimisation ability of the genetic algorithm, and show that optimisation at high temperature does not always translate to fitness at low temperature.

10.1 Future work

A great deal of latitude remains for future work to be conducted on the design, optimisation, simulation, and possible synthesis of molecular mechanical logic gates and circuits.

Currently, the capabilities of the framework outlined in this thesis are limited to the construction of combinatorial circuits, with no means of persistent information storage. A Turing machine is capable of executing any classical algorithm [87], and a computational system described as Turing-complete is capable of simulating a Turing machine. In order for a particular computational architecture to be considered Turing-complete, it must have the capability to store information over an arbitrary number of clock cycles. Thus, an obvious addition to the current array of circuit components would be some form of register that can store information on demand. A functional register would not only allow for the construction of a Turing-complete device, but also allow more careful investigation of complex information-entropic phenomena such as the mismatch cost.

The algorithmic optimisation of the logic gates is also another possible area for future research and improvement. The genetic algorithm generates a large amount of randomised data for each gate that it optimises. In theory, a neural network could be trained on the data to identify gate designs that would be probable failures with respect to a particular fitness function, allowing a large amount of time to be saved not simulating these gates. A sufficiently sophisticated neural network could even allow a Bayesian search to replace or augment the genetic algorithm [88], although this would likely only be possible for structures close to the training set.

Further work may also be carried out to convert the current coarse-grained model into a finer-grained model designed to simulate well-defined chemical structures as opposed to an abstract sphere. Before this can be done, the question of which specific molecular structures are most suited for constructing a practical molecular mechan-

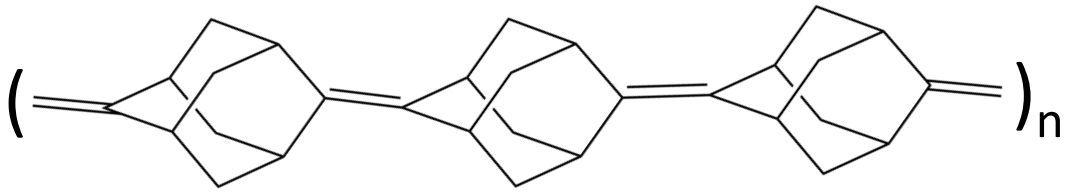


Figure 10.1: The poly-adamantylidene structural motif.

ical circuit must first be answered. These structures would need to be both highly rigid and be relatively easy to synthesise, preferably via controlled self-assembly. Metal-organic frameworks are a group of compounds which consist of organic ligands coordinated to metal ions. They are generally quite rigid and can be synthesised by self-assembly [89], and would likely make a good building material for the scaffold. However, the undirected self-assembly capabilities of metal-organic frameworks are generally limited to simple crystalline structures, and would thus be unsuitable for the logic gates themselves. In addition, the typical organic linkers used in metal-organic frameworks are generally not rigid with respect to rotation about their long axis, and are therefore unsuitable for the long-range transmission of torque which is required in many of the circuit components I have designed.

The lack of rotational rigidity can be resolved by replacing the typical phenyl-based organic linkers with a much more rigid structure based on the poly-adamantylidene motif (Fig. 10.1). The poly-adamantylidene motif consists of adamantane molecules linked by double bonds, and is thus rigid with respect to rotation about its principal axis. The crystalline adamantane structure is highly rigid, being responsible for the extraordinary hardness of diamond, but is extremely difficult to synthesise. While perhaps not quite as rigid, the poly-adamantylidene structure would probably be much easier to synthesise, as bi-adamantylidene can been synthesised via multiple methods [90, 91] and is in fact commercially available.

I shall now consider the problem of directed self-assembly. The poly-adamantylidene molecules must be terminated with the appropriate metal-ligating functional groups,

and a mechanism must be created that ensures the capped poly-adamantylidene molecules will only bind to specific metal atoms during the self-assembly process so as to create the complex aperiodic structures which comprise most of the circuit components. I propose that selective protection of binding sites on the metal atom via DNA or a similar molecule with binding specificity could be used to control the self-assembly process and create highly complex structures.

Currently, DNA has been used in the self-assembly of large, complex structures; these structures are known as DNA origami [92]. The self-assembly of DNA origami exploits the specificity of binding between complementary DNA base pairs. However, DNA is not a suitable material for the construction of molecular mechanical logic gates due to its low structural rigidity for its mass and inability to form stable base-pair interactions in the absence of a polar solvent containing dissolved ions as the presence of polar solvent increases the drag. Hence, my proposal only uses DNA in the self-assembly process but does not rely on it as a structural element.

In my scheme, binding sites on the metal atoms should be occupied by ligands attached to a short length of DNA or another analogous molecule. These ligands prevent the capped poly-adamantylidene molecules from binding, hence functioning as protecting groups. The ligands can be displaced from the metal atom in a controlled fashion by introducing a complementary strand of DNA bound to a moiety which is capable of disrupting the binding between the ligand and the metal atom, henceforth known as a deprotecting agent. The complementary strand ensures that the deprotecting agent binds only to a specific ligand, which it will displace from the metal atom, leaving it free to be attacked by the capped poly-adamantane molecule. A similar strategy can be used to protect the capped poly-adamantane molecules, allowing only the unprotected sites to bind to the metal atom. The scheme I have proposed could potentially be automated in a manner similar to the Merrifield peptide synthesis, which also relies on selective protection and deprotection to synthesise long

peptide chains from individual amino acids (admittedly via a much simpler process which does not rely on the base-pairing specificity of DNA) [93].

It must be acknowledged that much of what I have outlined here is highly speculative, and it is unlikely that my proposed schemes for synthesising molecular mechanical circuits will be viable in the near future. However, simulating and optimising the systems I have designed would still give valuable insight not only into the kinetics and thermodynamics of molecular mechanical computing, but also into the general operation of molecular mechanical systems, and perhaps eventually contribute to the physical realisation of a working molecular mechanical computer.

Bibliography

[1] Loïc Lannelongue, Jason Grealey, and Michael Inouye. Green algorithms: Quantifying the carbon footprint of computation. *Adv. Sci.*, 8:2100707, 2021.

[2] Michael McCool, Arch D. Robison, and James Reinders. Chapter 2 - background. In Michael McCool, Arch D. Robison, and James Reinders, editors, *Structured Parallel Programming*, pages 39–75. Morgan Kaufmann, Boston, 2012.

[3] D. Mamaluy and X. Gao. The fundamental downscaling limit of field effect transistors. *Appl. Phys. Lett.*, 106:193503, 2015.

[4] R. Landauer. Irreversibility and heat generation in the computing process. *IBM J. Res. Dev.*, 5:183–191, 1961.

[5] R. Fredkin and T. Toffoli. Conservative logic. *Int. J. Theor. Phys.*, 21:219–253, 1982.

[6] Charles H Bennett. The thermodynamics of computation—a review. *Int. J. Theor. Phys.*, 21:905–940, 1982.

[7] John PS Peterson, Roberto S Sarthour, Alexandre M Souza, Ivan S Oliveira, John Goold, Kavan Modi, Diogo O Soares-Pinto, and Lucas C Céleri. Experimental demonstration of information to energy conversion in a quantum system at the Landauer limit. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 472(2188):20150813, 2016.

[8] Yonggun Jun, Momčilo Gavrilov, and John Bechhoefer. High-precision test of Landauer's principle in a feedback trap. *Physical review letters*, 113(19):190601, 2014.

[9] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006.

[10] Colin P Williams, Scott H Clearwater, et al. *Explorations in quantum computing*. Springer, 1998.

[11] Peter W Shor. Scheme for reducing decoherence in quantum computer memory. *Phys. Rev. A*, 52:R2493, 1995.

[12] John S. Thomsen. Thermodynamics of an Irreversible Quasi-Static Process. *American Journal of Physics*, 28(2):119–122, 1960.

[13] Klaus Schmidt-Rohr. Expansion work without the external pressure and thermodynamics in terms of quasistatic irreversible processes. *Journal of Chemical Education*, 91(3):402–409, 2014.

[14] João P. S. Bizarro. Entropy production in irreversible processes with friction. *Phys. Rev. E*, 78:021137, Aug 2008.

[15] L. Szilard. Über die Entropieverminderung in einem thermodynamischen System bei Eingriffen intelligenter Wesen. *Z. Phys.*, 53:11–12, 1929.

[16] Rory A Brittain, Nick S Jones, and Thomas E Ouldridge. Biochemical Szilard engines for memory-limited inference. *New J. Phys.*, 21:063022, 2019.

[17] David H Wolpert. The stochastic thermodynamics of computation. *J. Phys. A Math. Theor.*, 52:193001, 2019.

[18] David H Wolpert and Artemy Kolchinsky. Thermodynamics of computing with circuits. *New J. Phys.*, 22:063047, 2020.

[19] Thomas E Ouldridge and David H Wolpert. Thermodynamics of deterministic finite automata operating locally and periodically. *arXiv preprint arXiv:2208.06895*, 2022.

[20] S. Kullback and R. A. Leibler. On Information and Sufficiency. *Ann. Math. Stat*, 22:79 – 86, 1951.

[21] Nazrul Anuar, Yasuhiro Takahashi, and Toshikazu Sekine. Adiabatic logic versus CMOS for low power applications. *Proc. ITC-CSCC 2009*, pages 302–305, 2009.

[22] Jitendra Kanungo and S. Dasgupta. Performance analysis of a complete adiabatic logic system driven by the proposed power clock generator. *Journal of Semiconductors*, 35:095001, sep 2014.

[23] V. K. Semenov, G. V. Danilov, and D. V. Averin. Classical and quantum operation modes of the reversible Josephson-junction logic circuits. *IEEE Trans. Appl. Supercond.*, 17:455–461, 2007.

[24] N. Takeuchi, Y. Yamanashi, and N. Yoshikawa. Reversibility and energy dissipation in adiabatic superconductor logic. *Sci. Rep. 7*, 7:75, 2017.

[25] V. K. Semenov, Y. A. Polyakov, and S. K. Tolpygo. AC-biased shift registers as fabrication process benchmark circuits and flux trapping diagnostic tool. *IEEE Trans. App. Supercond.*, 27:4, 2017.

[26] A. Genot, J. Bath, and A. Tuberfield. Reversible logic circuits made of DNA. *J. Am. Chem. Soc.*, 133:20080–20083, 2011.

[27] Tao Li, Finn Lohmann, and Michael Famulok. Interlocked DNA nanostructures controlled by a reversible logic circuit. *Nat. Commun.*, 5:4940, 2014.

[28] Tianqi Song, Abeer Eshra, Shalin Shah, Hieu Bui, Daniel Fu, Ming Yang, Reem Mokhtar, and John Reif. Fast and compact DNA logic circuits based on single-stranded gates using strand-displacing polymerase. *Nature nanotechnology*, 14:1075–1081, 2019.

[29] Rory A. Brittain, Nick S. Jones, and Thomas E. Ouldridge. What would it take to build a thermodynamically reversible Universal Turing machine? Computational and thermodynamic constraints in a molecular design, 2021.

[30] Evgeny Katz, Arshak Poghossian, and Michael J Schöning. Enzyme-based logic gates and circuits—Analytical applications and interfacing with electronics. *Analytical and bioanalytical chemistry*, 409(1):81–94, 2017.

[31] Takafumi Miyamoto, Shiva Razavi, Robert DeRose, and Takanari Inoue. Synthesizing biomolecule-based Boolean logic gates. *ACS synthetic biology*, 2(2):72–82, 2013.

[32] A Prasanna de Silva, Seiichi Uchiyama, Thomas P Vance, and Boontana Wannalerse. A supramolecular chemistry basis for molecular logic and computation. *Coordination chemistry reviews*, 251(13-14):1623–1632, 2007.

[33] David A. Leigh, M. Ángeles F. Morales, Emilio M. Pérez, Jenny K. Y. Wong, Carlos G. Saiz, Alexandra M. Z. Slawin, Adrian J. Carmichael, David M. Haddleton, A. Manfred Brouwer, Wybren Jan Buma, George W. H. Wurpel, Salvador León, and Francesco Zerbetto. Patterning through Controlled Submolecular Motion: Rotaxane-Based Switches and Logic Gates that Function in Solution and Polymer Films. *Angewandte Chemie International Edition*, 44(20):3062–3067, 2005.

[34] Patricia Remón, Rita Ferreira, Jose-Maria Montenegro, Rafael Suau, Ezequiel Pérez-Inestrosa, and Uwe Pischel. Reversible molecular logic: A photophysical example of a Feynman gate. *ChemPhysChem*, 10:2004–2007, 2009.

[35] Ye Liu, Fei Qin, Zi-Ming Meng, Fei Zhou, Qing-He Mao, and Zhi-Yuan Li. All-optical logic gates based on two-dimensional low-refractive-index nonlinear photonic crystal slabs. *Opt. express*, 19:1945–1953, 2011.

[36] Chinmoy Taraphdar, Tanay Chattopadhyay, and Jitendra Nath Roy. Mach–Zehnder interferometer-based all-optical reversible logic gate. *Opt. Laser Technol.*, 42:249–259, 2010.

[37] Ciyuan Qiu, Huifu Xiao, Liheng Wang, and Yonghui Tian. Recent advances in integrated optical directed logic operations for high performance optical computing: A review. *Front. Optoelectron.*, 15:1–17, 2022.

[38] Jie Hou, Liao Chen, Wenchan Dong, and Xinliang Zhang. 40 GB/s reconfigurable optical logic gates based on FWM in silicon waveguide. *Opt. Express*, 24:2701–2711, Feb 2016.

[39] T. Freeth, Y. Bitsakis, X. Moussas, J. H. Seiradakis, A. Tselikas, H. Mangou, M. Zafeiropoulou, R. Hadland, D. Bate, A. Ramsey, M. Allen, A. Crawley, P. Hockley, T. Malzbender, D. Gelb, W. Ambrisco, and M. G. Edmunds. Decoding the ancient Greek astronomical calculator known as the Antikythera Mechanism. *Nature*, 444:587–591, 2006.

[40] A.G. Bromley. Charles Babbage's analytical engine, 1838. *IEEE Ann.*, 20:29–45, 1998.

[41] Hiromi Yasuda, Philip R. Buskohl, Andrew Gillman, Todd D. Murphey, Susan Stepney, Richard A. Vaia, and Jordan R. Raney. Mechanical computing. *Nature*, 598:39–48, 2021.

[42] Ralph C. Merkle. Two types of mechanical reversible logic. *Nanotechnology*, 4:114, 1993.

[43] Yuanping Song, Robert M. Panas, Samira Chizari, Lucas A. Shaw, Julie A. Jackson, Jonathan B. Hopkins, and Andrew J. Pascall. Additively manufacturable micro-mechanical logic gates. *Nat. Commun.*, 10:882, 2019.

[44] Tie Mei, Zhiqiang Meng, Kejie Zhao, and Chang Qing Chen. A mechanical metamaterial with reprogrammable logical functions. *Nat. Commun.*, 12:1–11, 2021.

[45] J. Wenzler, T. Dunn, T. Toffoli, and P. Mohanty. A nanomechanical Fredkin gate. *Nano Lett.*, 14:89–93, 2011.

[46] Ralph C. Merkle, Robert A. Freitas, Tad Hogg, Thomas E. Moore, Matthew S. Moses, and James Ryley. Mechanical computing systems using only links and rotary joints. *J. Mechanisms Robotics.*, 10:061006, 2018.

[47] H. Kawakatsu, S. Kawai, D. Saya, M. Nagashio, D. Kobayashi, H. Toshiyoshi, and H. Fujita. Towards atomic force microscopy up to 100 MHz. *Rev. Sci. Instrum.*, 73:2317–2320, 2002.

[48] A. Perez-Cruz, A. Dominguez-Gonzalez, I. Stiharu, and R. A. Osornio-Rios. Optimization of Q-factor of AFM cantilevers using genetic algorithms. *Ultramicroscopy*, 115:61–67, 2012.

[49] Ian Seet, Thomas E. Ouldridge, and Jonathan P. K. Doye. Simulation of reversible molecular mechanical logic gates and circuits. *Phys. Rev. E*, 107:024134, Feb 2023.

[50] Petr Šulc, Flavio Romano, Thomas E. Ouldridge, Lorenzo Rovigatti, Jonathan P. K. Doye, and Ard A. Louis. Sequence-dependent thermodynamics of a coarse-grained DNA model. *J. Chem. Phys.*, 137:135101, 2012.

[51] Sebastian Kmiecik, Dominik Gront, Michal Kolinski, Lukasz Wieteska, Aleksandra Elzbieta Dawid, and Andrzej Kolinski. Coarse-grained protein models and their applications. *Chem. Rev.*, 116:7898–7936, 2016. PMID: 27333362.

[52] Johannes P. Dürholt, Raimondas Galvelis, and Rochus Schmid. Coarse graining of force fields for metal–organic frameworks. *Dalton Trans.*, 45:4370–4379, 2016.

[53] Artemy Kolchinsky and David H. Wolpert. Thermodynamic costs of Turing machines. *Phys. Rev. Research*, 2:033312, Aug 2020.

[54] Karel Proesmans, Jannik Ehrich, and John Bechhoefer. Optimal finite-time bit erasure under full control. *Phys. Rev. E*, 102:032105, Sep 2020.

[55] Abhishek Deshpande, Manoj Gopalkrishnan, Thomas E. Ouldridge, and Nick S. Jones. Designing the optimal bit: balancing energetic cost, speed and reliability. *Proc. Math. Phys.*, 473:20170117, 2017.

[56] Dominique Chu. Thermodynamics of quasideterministic digital computers. *Phys. Rev. E*, 97:022121, 2018.

[57] Dominique Chu and Richard E Spinney. A thermodynamically consistent model of finite-state machines. *Interface focus*, 8:20180037, 2018.

[58] Petr Krysl and Lance Endres. Explicit Newmark/Verlet algorithm for time integration of the rotational dynamics of rigid bodies. *Int. J. Numer. Methods Eng.*, 62:2154–2177, 2005.

[59] R. L. Davidchack, T. E. Ouldridge, and M. V. Tretyakov. New Langevin and gradient thermostats for rigid body dynamics. *J. Chem. Phys.*, 142:144114, 2015.

[60] Michael P Allen and Dominic J Tildesley. *Computer simulation of liquids*. Oxford university press, 2017.

[61] William Mattson and Betsy M Rice. Near-neighbor calculations using a modified cell-linked list method. *Comput. Phys. Commun.*, 119:135–148, 1999.

[62] Loup Verlet. Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules. *Phys. Rev.*, 159:98–103, Jul 1967.

[63] Daan Frenkel and Berend Smit. *Understanding molecular simulation: From algorithms to applications*, volume 1. Elsevier, 2001.

[64] Sumit Sharma, Pramod Kumar, and Rakesh Chandra. Chapter 2 - overview of BIOVIA Materials Studio, LAMMPS, and GROMACS. In *Molecular Dynamics Simulation of Nanocomposites Using BIOVIA Materials Studio, LAMMPS and GROMACS*, Micro and Nano Technologies, pages 39–100. Elsevier, 2019.

[65] John Edward Jones. On the determination of molecular fields.—I. From the variation of the viscosity of a gas with temperature. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 106:441–462, 1924.

[66] J. D. Weeks, D. Chandler, and H. C. Andersen. Role of repulsive forces in determining the equilibrium structure of simple liquids. *J. Chem. Phys.*, 54:5237, 1971.

[67] J.W. Ponder and D.A. Case. Force fields for protein simulations. *Adv. Prot. Chem.*, 66:27–85, 2003.

[68] Ralph C Merkle. Molecular building blocks and development strategies for molecular nanotechnology. *Nanotechnology*, 11(2):89, 2000.

[69] Martin R Bryce. A review of functional linear carbon chains (oligoynes, polyynes, cumulenes) and their applications as molecular wires in molecular electronics and optoelectronics. *Journal of Materials Chemistry C*, 2021.

[70] David C Weggel, David M Boyajian, and S Chen. Modelling structures as systems of springs. *World Transactions on Engineering and Technology Education*, 6(1):169, 2007.

[71] Ken Sekimoto. Kinetic characterization of heat bath and the energetics of thermal ratchet models. *Journal of the Physical Society of Japan*, 66(5):1234–1237, 1997.

[72] Md. Asif Nashiry and Jacqueline E. Rice. A reversible majority voter circuit and applications. In *2017 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, pages 1–6, 2017.

[73] Cristiano Santos, Pascal Vivet, Jean-Philippe Colonna, Perceval Coudrain, and Ricardo Reis. Thermal performance of 3D ICs: Analysis and alternatives. In *2014 International 3D Systems Integration Conference (3DIC)*, pages 1–7, 2014.

[74] J. Friedman, L. Mourokh, and M. Vittadello. Mechanism of Proton Pumping in Complex I of the Mitochondrial Respiratory Chain. *Quantum Rep.*, 3:425–434, 2021.

[75] J. Hernandez, E.R. Kay, and D. R. Leigh. A reversible synthetic rotary molecular motor. *Science*, 306:1532–1537, 2004.

[76] Miloš Bečvář. Teaching basics of instruction pipelining with HDLDLX. In *Proceedings of the 2004 workshop on Computer architecture education: held in conjunction with the 31st International Symposium on Computer Architecture*, pages 16–es, 2004.

[77] D.C. Wong, G. De Micheli, and M.J. Flynn. Designing high-performance digital circuits using wave pipelining: Algorithms and practical experiences. *IEEE TCAD*, 12:25–46, 1993.

[78] Wayne P Burleson, Maciej Ciesielski, Fabian Klass, and Wentai Liu. Wave-pipelining: A tutorial and research survey. *IEEE Transactions on Very Large Scale Integration (VLSI) systems*, 6:464–474, 1998.

[79] C Thomas Gray, Wentai Liu, Ralph K Cavin III, and Ralph K Cavin. *Wave Pipelining: Theory and CMOS Implementation: Theory and Cmos Implementation*, volume 248. Springer Science & Business Media, 1994.

[80] J. Shen and M.H. Lipasti. *Modern Processor Design: Fundamentals of Superscalar Processors*. Electrical and Computer Engineering. McGraw-Hill Companies, Incorporated, 2005.

[81] Carlos A Coello, Alan D Christiansen, and Arturo Hernández Aguirre. Automated design of combinational logic circuits using genetic algorithms. In *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*, pages 335–338. Citeseer, 1997.

[82] Toufik Bendib and Fayçal Djeffal. Electrical performance optimization of nanoscale double-gate MOSFETs using multiobjective genetic algorithms. *IEEE T. Electron Dev.*, 58:3743–3750, 2011.

[83] Trailokya Nath Sasamal, Ashutosh Kumar Singh, and Anand Mohan. Reversible logic circuit synthesis and optimization using adaptive genetic algorithm. *Procedia Comput. Sci.*, 70:407–413, 2015.

[84] Mitsuo Gen and Runwei Cheng. *Genetic algorithms and engineering optimization*, volume 7. John Wiley & Sons, 1999.

[85] Federico Millo, Pranav Arya, and Fabio Mallamo. Optimization of automotive diesel engine calibration using genetic algorithm techniques. *Energy*, 158:807–819, 2018.

[86] Gilbert Syswerda et al. Uniform crossover in genetic algorithms. In *ICGA*, volume 3, pages 2–9, 1989.

[87] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proc. London Math. Soc.*, s2-42:230–265, 1937.

[88] AkshatKumar Nigam, Pascal Friederich, Mario Krenn, and Alán Aspuru-Guzik. Augmenting genetic algorithms with deep neural networks for exploring the chemical space. *arXiv preprint arXiv:1909.11655*, 2019.

[89] Qi-Long Zhu and Qiang Xu. Metal–organic framework composites. *Chem. Soc. Rev.*, 43:5468–5512, 2014.

[90] Genrikh A Tolstikov, Berta M Lerman, and Tatjana A Belogaeva. A convenient synthesis of adamantylideneadamantane. *Synth. Commun.*, 21:877–879, 1991.

[91] A Paul Schaap and Gary R Faler. Convenient synthesis of adamantylideneadamantane. *J. Org. Chem.*, 38:3061–3062, 1973.

[92] Swarup Dey, Chunhai Fan, Kurt V Gothelf, Jiang Li, Chenxiang Lin, Longfei Liu, Na Liu, Minke AD Nijenhuis, Barbara Saccà, Friedrich C Simmel, et al. DNA origami. *Nat. Rev. Methods Primers*, 1:1–24, 2021.

[93] RB Merrifield. Automated peptide synthesis. In *Hypotensive Peptides*, pages 1–13. Springer, 1966.

Appendix A

Circuit Diagrams

Combinatorial logic circuits composed of smaller logic gates are most conveniently illustrated with circuit diagrams. These figures depict the connections between the inputs and outputs of the various component logic gates which constitute the combinatorial circuit. Fig. A.1 contains the symbols of the most commonly used logic gates, including all gates used in this thesis.

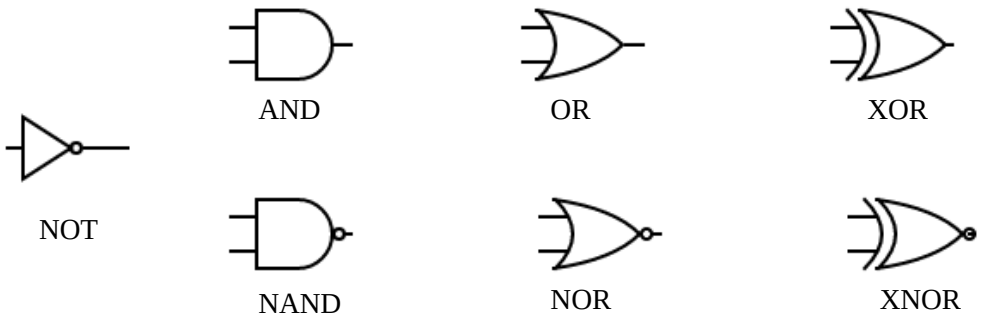


Figure A.1: A legend of the symbols used to represent common one- and two-input logic gates in circuit diagrams.

Appendix B

Rigor Mortis and Alabama code

The C++ code for Rigor Mortis may be found at this clickable link. The C++ code for Alabama may be found at this clickable link. Note that both programs require the Eigen linear algebra library to successfully compile.

Appendix C

List of Supplemental Animations and Videos

In order to improve the comprehensibility of the diagrams, a number of VMD animations and videos of the various circuit components in action have been prepared. The Supplemental Videos can be found at this clickable link. VMD-viewable trajectory files of the processes illustrated in the Supplemental Videos are available at the Oxford University Research Archive. The following are a list of descriptions of the Supplemental Videos by chapter:

C.1 Chapter 3

Supplemental Video 1: The animation depicts one full cycle of the driver. The driver is initially in the fully disengaged state and the input bit (blue) is in state '1'. It reaches maximal engagement at 0.08 allowing information transfer from the input bit to the output bit (green) via the clutch (red). By the end of the animation the driver has returned to the fully disengaged state.

Supplemental Video A: The animation depicts the NAND gate linked to two drivers in the '00' state. The drivers transition from a disengaged state to a fully

engaged state before the process is reversed. At the point of maximum engagement, the purple output bit of the NAND gate returns ‘1’. This process is executed in the absence of thermal noise.

Supplemental Video B: The animation depicts the NAND gate linked to two drivers in the ‘01’ state. The drivers transition from a disengaged state to a fully engaged state before the process is reversed. At the point of maximum engagement, the purple output bit of the NAND gate returns ‘1’. This process is executed in the absence of thermal noise.

Supplemental Video C: The animation depicts the NAND gate linked to two drivers in the ‘11’ state. The drivers transition from a disengaged state to a fully engaged state before the process is reversed. At the point of maximum engagement, the purple output bit of the NAND gate returns ‘0’. This process is executed in the absence of thermal noise.

Supplemental Video D: The animation depicts the switch initalised in the ‘1’ state. Over the course of the animation, the switch is transitioned to the ‘0’ state. This process is executed in the absence of thermal noise.

C.2 Chapter 4

Supplemental Video 2: The animation depicts the process by which the storage bit is overwritten by the input bit for the double-bit system. The input bit (blue; left) is initialized to ’1’ and the storage bit (red; centre) is initialised to ’0’. When the system is initialised, the lock is engaged and the driver associated with the input bit is disengaged. The driver associated with the dummy bit (blue; right) is always in the disengaged state. At 0.09, the lock becomes disengaged, allowing the storage bit to move freely. At 0.16, the driver becomes fully engaged, transducing information from the input bit to the storage bit. At 0.30, the lock re-engages, locking the storage

bit in place. Finally, the driver becomes disengaged, completing the process.

C.3 Chapter 5

Supplemental Videos 3, 4 and 5: The animations depict the neutral transitions, where the NAND gate is linked to a pair of drivers with input bits set to the states '00', '01' and '11', respectively. The drivers are initialised in the fully disengaged state, reach maximal engagement at 0.08 and are returned to the fully disengaged state at the end of the animation. These videos illustrate how the engaged input bits affect the orientation of the winged structure and hence the state of the output bit. Supplemental Video 6: The animation depicts the process described in Section V B where the NAND gate is linked to a pair of switches initialised to the state '11' and driven to the state '00'. The switches reach state '00' at 0.08 and are returned to state '11' at the end of the animation.

C.4 Chapter 6

Supplemental Video 7: The animation depicts the phase-chained system consisting of 2 NAND gates. The system is initialised with the switch in an indeterminate state. By the 1s mark, the switches are toggled to the '11' state. At 0.03, the signal booster of the first NAND gate is engaged and the gate returns '0'. At 0.04, the signal booster of the second NAND gate is engaged and the gate returns '0'. The external dipole is immediately reversed, returning the system to the starting state by 0.08.

At 0.09, the switches are toggled to the '00' state. At 11s, the signal booster of the first NAND gate is engaged and the gate returns '1'. At 0.12, the signal booster of the second NAND gate is engaged and the gate returns '1'. The external dipole is once again reversed, returning the system to the starting state by the end of the animation.

Supplemental Video 8: The animation depicts a process analogous to that depicted in Supplemental Video 7; this particular example is a phase-chained system consisting of 3 NAND gates. The system is initialised with the switch in an indeterminate state. By the 0.01 mark, the switches are toggled to the '11' state. At 0.02, the signal booster of the first NAND gate is engaged and the gate returns '0'. At 0.03, the signal booster of the second NAND gate is engaged and the gate returns '0'. At 0.04, the signal booster of the third NAND gate engages as the gate returns '0'. The external dipole is immediately reversed, returning the system to the starting state by 0.08.

At 0.09, the switches are toggled to the '00' state. At 0.11, the signal booster of the first NAND gate is engaged and the gate returns '1'. At 0.12, the signal booster of the second NAND gate is engaged and the gate returns '1'. At 0.13, the signal booster of the third NAND gate engages as the gate returns '1'. The external dipole is once again reversed, returning the system to the starting state by the end of the animation.

Supplemental Video 9: The animation depicts a process analogous to that depicted in Supplemental Videos 7 and 8; this particular example is a phase-chained system consisting of 4 NAND gates. By the 0.01 mark, the switches are toggled to the '11' state. At 0.02, the signal booster of the first NAND gate is engaged and the gate returns '0'. At 0.04, the signal booster of the second NAND gate is engaged and the gate returns '0'. At 0.05, the signal booster of the third NAND gate engages as the gate returns '0'. At 0.06, the signal booster of the fourth NAND gate engages as the gate returns '0'. The external dipole is immediately reversed, returning the system to the starting state by 0.12.

At 0.14, the switches are toggled to the '00' state. At 0.15, the signal booster of the first NAND gate is engaged and the gate returns '1'. At 0.16, the signal booster of the second NAND gate is engaged and the gate returns '1'. At 0.18, the signal

booster of the third NAND gate engages as the gate returns ‘1’. At 0.19, the signal booster of the fourth NAND gate engages as the gate returns ‘1’. The external dipole is once again reversed, returning the system to the starting state by the end of the animation.

C.5 Chapter 7

Supplemental Video 10: The animation depicts the half adder initialised to the state ’01’ and driven to the state ’11’. The system is initialised with the switch in an indeterminate state. By the 0.01 mark, the switches are toggled to the ’01’ state. At 0.02, the boosters of the switch become engaged. At 0.03, the boosters of the first set of NAND gates become engaged with both returning ‘1’. At 0.04, the drivers linking the first set of NAND gates to the final NAND gate become engaged, forcing the orange carry bit into the ‘0’ state. Shortly after, the signal booster of the final NAND gate becomes engaged, forcing its output bit into the ‘0’ state and therefore the cyan sum bit into the ‘1’ state via the inverting connection. The external dipole is immediately reversed, returning the system to the starting state by 0.08.

At 0.09, the switches are toggled to the ‘11’ state. At 0.10 the boosters of the switch become engaged. At 0.11, the boosters of the first set of NAND gates become engaged with the top NAND gate returning ‘1’ and the bottom NAND gate returning ‘0’. At 0.12, the drivers linking the first set of NAND gates to the final NAND gate become engaged, forcing the orange carry bit into the ‘1’ state. Shortly after, the signal booster of the final NAND gate becomes engaged, forcing its output bit into the ‘1’ state and therefore the cyan sum bit into the ‘0’ state via the inverting connection. The external dipole is once again reversed, returning the system to the starting state by the end of the animation.

Supplemental Video 11: The animation depicts the half adder with boosted in-

verter and drivers initialised to the state '01' and driven to the state '11'. The system is initialised with the switch in an indeterminate state. By the 0.01 mark, the switches are toggled to the '01' state, followed shortly by the boosters of the switch engaging. At 0.02, the booster of the inverter becomes engaged, followed shortly by the boosters of the first set of NAND gates engaging, with both returning '1'. At 0.03, the drivers linking the first set of NAND gates to the final NAND gate become engaged, followed shortly by the boosters for the drivers, forcing the orange carry bit into the '0' state. At 0.04, the signal booster of the final NAND gate becomes engaged, forcing its output bit into the '0' state and therefore the cyan sum bit into the '1' state via the inverting connection. The external dipole is immediately reversed, returning the system to the starting state by 0.08.

At 0.09, the switches are toggled to the '11' state, followed shortly by the boosters of the switch engaging. At 0.10, the booster of the inverter becomes engaged, followed shortly by the boosters of the first set of NAND gates engaging, with the top NAND gate returning '1' and the bottom NAND gate returning '0'. At 0.11, the drivers linking the first set of NAND gates to the final NAND gate become engaged, followed shortly by the signal boosters of the drivers, forcing the orange carry bit into the '1' state. At 0.12, the signal booster of the final NAND gate becomes engaged, forcing its output bit into the '1' state and therefore the cyan sum bit into the '0' state via the inverting connection. The external dipole is once again reversed, returning the system to the starting state by the end of the animation.

C.6 Chapter 8

Supplemental Video 12a: The animation depicts the pipelined two-bit circuit executing a forward pipelining process followed by a successful reverse pipelining process. The switch is initialised in the '1' state. The signal booster of the first bit fully en-

gages at 0.01, locking the bit into the '1' state. Subsequently, the driver and signal booster of the second bit fully engages at 0.03, locking that bit into the '1' state. The signal boosters of both bits slowly disengage over the next few seconds. At 0.05, the switch has toggled over to the '0' state and the process is repeated with the '0' state being transduced to the first bit at 0.07 and the second at 0.08. At this point, the external dipoles driving the system are reversed. At 0.12, the signal booster of the second bit begins to engage. At this point, there is no information retained by the second bit, at the state it takes is largely due to random chance. In this example, the second bit correctly enters the '1' state, which is then transduced to the first bit at 0.15, in time for the switch to toggle back to the '1' state. As there is no clash between the states of the first bit and the switch, this process proceeds smoothly.

Supplemental Video 12b: The animation depicts the pipelined two-bit circuit executing a forward pipelining process followed by a failed reverse pipelining process. Up to 0.12, the failed and successful reverse pipelining processes are identical. However, in this video, the second bit enters the '0' state, which is then transduced to the first bit at 0.15. This causes the first bit (in the '0' state) to violently clash with the switch (in the '1') state. The switch forces the first bit into the '1' state at 0.16 despite its signal booster being mostly engaged. This is a thermodynamically costly process that generates a large amount of entropy.

Supplemental Video 12c: The animation depicts the phase-chained two-bit circuit. The switch is initialised in an indeterminate state, toggling to the '1' state only at 0.01. The driver of the first bit engages at 0.02 and the signal booster at 0.03, and the driver and signal booster of the second bit at 0.04. The external dipoles driving the system are then reversed, returning the system to its initial state at 0.08. Note that the driver and signal booster of the first bit only disengage after the driver and signal booster of the second bit is disengaged. The switch is then toggled to the '0' state at 0.09 and the process described above is repeated, causing both bits to adopt

the ‘0’ state before the system is once again driven in reverse from 0.13, returning it to its initial state at 0.16.

Supplemental Video 13: The animation depicts the pipelined two-bit circuit chained to a pseudo-register set to ‘1’. This process is similar to the one depicted in Supplemental Video 12a. However, a key difference occurs at 0.03, when the pseudo-register engages at the same time as the signal booster of the second bit. While this makes no difference in the forward process, in the reverse process at 0.12, the engagement of the pseudo-register prevents the second bit from entering the ‘0’ state, preventing the situation depicted in Video 12b from occurring.

Supplemental Video 14-1: The animation depicts the pipelined NAND circuit of length 2. The system is initialised with the switches toggled to ‘1’ and the signal booster of the first NAND gate fully engaged. At 0.02, the driver between the output of the first NAND gate and the input of the second becomes engaged, along with the signal booster of the second NAND gate. The driver and signal boosters of these gates then disengage over the next few seconds. Subsequently, the switches toggle over to the ‘0’ state at 0.06, in time for the signal booster of the first NAND gate to fully engage, followed by the driver and signal booster of the second NAND gate at 0.08. The external dipoles driving the system are then reversed. As with Video 12b, problems occur when the system is reversed. At 0.14, the signal booster of the second NAND gate falls on the output bit, forcing it into the ‘1’ state instead of the ‘0’ state it was in at 0.02. This erroneous state is partially transduced to the first NAND gate at 0.14 before the switches toggling to the ‘1’ state force the output of the first NAND gate into the ‘0’ state.

Supplemental Videos 14-2 and 14-3: These animations depict the pipelined NAND circuit of length 3 and 4 respectively. The process depicted is analogous to that of Video 14-2. The critical point at which the signal booster of the second NAND gate forces the output bit of the final NAND gate into the ‘1’ state occurs at 0.13 for Video

14-2 and 0.12 for Video 14-3.

Supplemental Videos 15-1, 15-2 and 15-3a: These animations depict the pipelined NAND circuits of length 2, 3 and 4 respectively equipped with a pseudo-register. In a situation analogous with Video 13, the engagement of the pseudo-register at 0.14 and 0.13 respectively forces the output bit of the final NAND gate into the '0' state, making the process thermodynamically reversible.

Supplemental Video 15-3b: This animation depicts the pipelined NAND circuit of length 4 equipped with a pseudo-register encountering an error during the forward pipelining process. The forward pipelining proceeds in a manner similar to Video 15-3a up until the 0.07 mark, when the output bits of the last two gates in the chain enter the '0' state when they should instead be returning '1'. It should be noted that the failure of the second-last gate in the chain appears to be due to interference from the final gate, hence the need for an improved clutching mechanism for the drivers to reduce this interference.

Supplemental Video 16: This animation depicts the pipelined NAND circuit of length 4 equipped with a pseudo-register and modified drivers. The process depicted is analogous to that shown in Video 15-3a.