

The Development of Reversible and Irreversible Digital Logic Circuits for the Experimental Comparison of Power Consumption

By

Stanley Vass (2306978),
Bryan Skeen (2307828),
Robert Webb (2308625),
Rhys Cheedy (2307075)

Project Submitted in completion of the module: Group
Project ACDA5017

Abstract (RC)

With Koomey's law beginning to slow and the continuous need for ever more demanding energy efficient computation a new paradigm is needed. Reversible logic promises to fulfil this having the ability to overcome Landauer's limit and achieve zero-power computation. Limited physical implementations exist for reversible logic, but to contribute to this a reversible logic circuit is design and fabricated. In this report a reversible logic adder circuit is designed and alongside an irreversible complement using an S2LAL topology to be used as testbeds for the comparison of the power consumption differences between reversible and irreversible logic. Errors in the design of parts of the reversible logic PCB has seen power measurement only for the irreversible S2LAL adder being performed. Although this meant a complete comparison could not be made and impact of reversible logic has on power consumption assessed, an understanding of the success of S2LAL has still been achieved. Also, this report still presents a feasible reversible logic design in S2LAL and paves a way for a successful physical implementation.

Table of Contents

1. Introduction (BW).....	7
2. Background.....	7
2.1 Quantifying Information (BS).....	7
2.2 Landauer's Principle (RC)	9
2.3 Reversible Logic (BS).....	9
2.3.1 What is reversible Logic?.....	9
2.3.2 Reversible Logic gates	10
2.3.3 Reversible Logic Implementations	12
2.4 Digital Logic Hardware Families (SV).....	12
3. Methodology (SV)	14
4. Design and Implementation	14
4.1 4-Bit Adder Design (RC)	14
4.2 S2LAL Logic Gate Design.....	17
4.2.1 The CD4007 (SV)	17
4.2.2 Irreversible gates (SV)	17
4.2.3 Reversible gates (BS).....	17
4.3 Simulations and Proof of Concept Testing (BW/BS).....	18
4.3.1 SPICE Simulations.....	18
4.3.2 Proof-of-Concept Gates	20
4.4 PCB Design (BS)	22
4.4.1 Logic Gates	22
4.4.2 Adder Backplates	24
4.5 Measurement & Interface Board.....	25
5. Results and Analysis (BW)	26
6. Discussion and Evaluation (BS)	28
7. Conclusion and Recommendations (SV)	29
8. Reflections	29
8.1 Project Management (BW)	29
8.2 Stanley Vass	31
8.3 Bryan Skeen.....	31
8.4 Robert Webb	32
8.5 Rhys Cheedy	32
References.....	32
Appendix A: Example Explaining Information Quantity	36
Appendix B: LTSpice Netlists of CD4007 Gate Implementations	37
Appendix C: LTSpice CD4007 Spice Model.....	43

Appendix D: SPICE Simulation Outputs for S2LAL Logic Gates	44
Appendix E: PCB Designs	47

Figure 1. Logic gate symbol of reversible NOT gate.....	10
Figure 2. Logic gate symbol of reversible Feynman gate.....	11
Figure 3. Logic gate symbol of reversible Toffoli gate.....	11
Figure 4. Logic gate symbol of reversible Fredkin gate.	11
Figure 5. Logic gate symbol of reversible Peres gate.	11
Figure 6. Toffoli gate equivalent of a Peres gate.....	12
Figure 7. CMOS transistor constructions of a transmission gate.....	13
Figure 8. Logic gate level implementation of the irreversible 4-bit ripple carry adder.	15
Figure 9. Logic gate level design of the 4-bit reversible adder using Peres gates.	15
Figure 10. Irreversible 4-bit adder implementations using S2LAL Buffers, AND, OR and XOR gates.	16
Figure 11. Reversible 4-bit adder implementation using S2LAL Peres gates and Buffers.	16
Figure 12. T-Cell design for the $A \cdot B \oplus C$ operation of the Peres gate.	18
Figure 13. CD4007 implementation of the Peres gate created within LTSpice.	18
Figure 14. The eight 10kHz trapezoidal waves used in the S2LAL logic gate simulations.....	19
Figure 15. The input signals used with the simulations of the S2LAL logic gates.	19
Figure 16. The proof-of-concept buffer created on matrix board. Only the rising half of the Buffer was created to demonstrate its operation.....	20
Figure 17. Input and output signals seen in the operation of the proof-of-concept buffer.	20
Figure 18. The proof-of-concept AND gate created on matrix board. Both rising and falling halves of this gate were implemented.	21
Figure 19. Input and output signals seen in the operation of the proof-of-concept AND gate.	21
Figure 20. The proof-of-concept XOR gate created on matrix board. Both rising and falling halves of this gate were implemented.	22
Figure 21. Input and output signals seen in the operation of the proof-of-concept XOR gate.	22
Figure 22. Logic gate breakout board constructed with a PCB edge connector reciprocal and breakout header pins to enable the easy testing of the constructed logic gates to identify correct design operation and manufacturing faults.....	23
Figure 23. Change made to the CD4007 implementation of the OR gate to correct its operation.	23
Figure 24. The simulation outputs of the original OR gate design and altered design.	23
Figure 25. Modified OR gate PCB.....	24
Figure 26. Constructed reversible adder backplane filled with logic gates.	24
Figure 27. Constructed irreversible adder backplane, filled with corresponding logic gates.	25
Figure 28. Constructed measurement and interface PCB.	26
Figure 29. Measured current flow for the rising and falling power-clock 0 signal.....	26
Figure 30. Measured current flow for the rising and falling power-clock 2 signal.....	27
Figure 31. Measured current flow for the rising and falling power-clock 1 signal.....	27
Figure 32. Measured current flow for the rising and falling power-clock 3 signal.....	27
Figure 33. Gantt chart created to manage tasks and timelines of each team member for the delivery of the project.....	30

Appendix Figure 1. Left is the SPICE simulation output of the ideal MOSFET implementation of the S2LAL Buffer and right is the CD4007 implementations' outputs.....	44
Appendix Figure 2. Left is the SPICE simulation output of the ideal MOSFET implementation of the S2LAL AND gate and right is the CD4007 implementations outputs.	44
Appendix Figure 3. Left is the SPICE simulation output of the ideal MOSFET implementation of the S2LAL OR gate and right is the CD4007 implementations outputs.....	45
Appendix Figure 4. SPICE simulation outputs of the altered CD4007 implementation of the OR gate.	45

Appendix Figure 5. Left is the SPICE simulation output of the ideal MOSFET implementation of the S2LAL XOR gate and right is the CD4007 implementations outputs.	46
Appendix Figure 6. Buffer gate PCB design.	47
Appendix Figure 7. AND gate PCB design.	48
Appendix Figure 8. OR gate PCB design.	49
Appendix Figure 9. XOR gate PCB design.	50
Appendix Figure 10. Peres gate PCB design.	51
Appendix Figure 11. Reversible adder backplane PCB.	52
Appendix Figure 12. Irreversible backplane PCB design.	53

Table 1. Information transformation of a 2-input AND gate, showing the characterisation of self-information of the inputs and outputs with their differences due to the gate's computation.	8
Table 2. Information transformation of a 2-input OR gate, showing the characterisation of self-information of the inputs and outputs with their differences due to the gate's computation.	8
Table 3. Information transformation of a 2-input XOR gate, showing the characterisation of self-information of the inputs and outputs with their differences due to the gate's computation.	8
Table 4. Key parameters to be considered for the efficient design of reversible logic circuits.	9
Table 5. Key metrics from the measured current flow of each power-clock signal.	28

1. Introduction (BW)

In a rapidly evolving field of digital electronics and computing, the development of efficient computation is seen to be beginning to stall; observed in [1] with the slowing of Koomey's law – a link between time and improvements in computation energy efficiency. This makes way for a new paradigm of computing aiming to keep up with today's extensive computational demands. The idea of reversible computing offers the ability to drastically reduce the power consumption of computation; even offering the possibility to achieve zero-power computation by providing a method for performing functions with zero information loss and overcoming the energy dissipation described by Landauer's Principle.

Limited physical implementations of transistor-based implementations of reversible logic circuits exist, but this report aims to contribute to this by presenting the design and implementation of a reversible adder circuit. This will be used alongside an irreversible adder design as testbeds to assess the power savings resulting from reversible logic and the impact of information loss. First is a review of information quantification and the impact logical functions can have on the information quantity in a system, along with the impact information loss can have on energy dissipation. Next reversible logic is presented with an exploration of the requirements for physical implementation. Following is the design and implementation of the reversible and irreversible adder circuits, which are then used to measure their respective power consumptions and compare the impact reversible logic has.

2. Background

2.1 Quantifying Information (BS)

Measuring information is useful for quantifying the affect a logical operation has on the entropy/energy of a system. This is demonstrated by Shannon in his 1948 paper "A Mathematical Theory of Communication" [2], in which he introduces entropy as a measure of information, characterising a variable's uncertainty. This measure of entropy is defined as the minimum number of binary questions needed to determine the value of a given variable. An example in [3], presented in *Appendix A: Example Explaining Information Quantity*, illustrates how the entropy of a system can be derived from the probability distribution of a variable's value. The average information entropy, expressed in the unit bits, is calculated using the following equation:

$$H(X) = - \sum_{x_i \in X} P(x_i) \log_2(P(x_i))$$

Where X represents an arbitrary variable, which has the possibility to take on a value x_i .

The contribution of a specific outcome to a variable's entropy is called self-information. This value helps to quantify how possible values of a variable contribute to the overall entropy. Self-information is defined as:

$$I(X) = -\log_2(P(x_i))$$

With this definition of information quantity, it becomes possible to characterise the information content of the inputs and outputs of digital logic gates.

Starting with a 2-input AND gate an information transformation was performed in *Table 1*. A loss in self-information between the input and output for input cases x_1, x_2, x_3 can be observed. Whereas for case x_4 (both inputs logical 1), no information was lost. This aligns with expectations, as in the case of output y_1 , multiple input combinations are possible, thereby removing the ability to deduce the original input. Whereas for y_2 , the corresponding input is unique, and the ability to determine

the input remains, preserving information. Similar information transformations can be performed for OR and XOR gates, shown in *Table 2* and *Table 3*.

Table 1. Information transformation of a 2-input AND gate, showing the characterisation of self-information of the inputs and outputs with their differences due to the gate's computation.

Inputs				Outputs			Self-Information		
Case	a	b	$P(x_i)$	Case	$a \cdot b$	$P(y_j)$	$I(x_i)$	$I(y_j)$	$\Delta I_{i,j}$
x_1	0	0	0.25	y_1	0	0.75	2	0.415	1.585 bits
x_2	0	1	0.25				2		
x_3	1	0	0.25				2		
x_4	1	1	0.25	y_2	1	0.25	2	2	0 bits

Table 2. Information transformation of a 2-input OR gate, showing the characterisation of self-information of the inputs and outputs with their differences due to the gate's computation.

Inputs				Outputs			Self-Information		
Case	a	b	$P(x_i)$	Case	$a + b$	$P(y_j)$	$I(x_i)$	$I(y_j)$	$\Delta I_{i,j}$
x_1	0	0	0.25	y_1	0	0.25	2	2	0 bits
x_2	0	1	0.25	y_2	1	0.75	2	0.415	1.585 bits
x_3	1	0	0.25				2		
x_4	1	1	0.25				2		

Table 3. Information transformation of a 2-input XOR gate, showing the characterisation of self-information of the inputs and outputs with their differences due to the gate's computation.

Inputs				Outputs			Self-Information		
Case	a	b	$P(x_i)$	Case	$a \cdot b$	$P(y_j)$	$I(x_i)$	$I(y_j)$	$\Delta I_{i,j}$
x_1	0	0	0.25	y_1	0	0.5	2	1	1 bit
x_2	0	1	0.25	y_2	1	0.5	2	1	1 bit
x_3	1	0	0.25				2		
x_4	1	1	0.25	y_1	0	0.5	2	1	1 bit

This information transformation for basic logic gates can be extended to more complex functions.

In [4] Hänninen et al. quantified the input and output self-information of digital adders, demonstrating that information loss can be greatly impacted by the manner of implementation. For a full adder, information loss is minimal when implemented monolithically and increases when implemented modularly from full-adder blocks. Analysing a more realistic implementation with AND, OR and XOR gates information loss significantly increases with an average of 38 bits for an 8-bit adder [4].

2.2 Landauer's Principle (RC)

At the core of understanding the impact of the information loss that can occur in a logical function is Landauer's Principle. This principle relates loss of information to increasing physical entropy, thus an associated energy dissipation. Landauer showed the increase in thermodynamic entropy from 1-bit of information loss to be [5]:

$$\Delta S = k \ln 2$$

Where k is Boltzmann's constant ($\approx 1.38 \times 10^{-23}$).

The corresponding energy dissipation is therefore seen to be:

$$\Delta E = kT \ln 2$$

Where T is the operating temperature of the system.

This principle presents a fundamental limit to the energy efficiency achievable through traditional computation. Providing an incentive to research reversible computing as a promising method of continuing the development of computational performance. [6]

2.3 Reversible Logic (BS)

2.3.1 What is reversible Logic?

To reduce the energy dissipated through information loss, as described by Landauer's Principle, a method of digital logic which preserves information is needed. In 1973, Bennett showed that any arbitrary computation can be performed reversibly – with no information loss – using an n -tape Turing machine, and with the goal of zero power computation, suggested the concept of thermodynamically reversible computation [7] [8]. Since then, many reversible logic gates have been proposed [9] [10] [11]. Unlike irreversible gates, reversible gates are bijective, meaning each input maps uniquely to an output, forming a $n \times n$ input-output structure for reversible gates [12]. An important characteristic of reversible gates is quantum cost (QC), measuring the number of 1×1 and 2×2 gates needed to build a reversible function. With 1×1 gates contributing a QC of 0 and 2×2 contributing 1. Here it should be noted the close link with quantum computing [13], as some 1×1 and 2×2 need to calculate QC realise quantum functions requiring qubits, something transistor-based implementations cannot replicate. Despite this, QC serves as a useful proxy for estimating transistor count and is widely used to assess design efficiency, with efforts being made to minimise it [14]. However, other metrics such as circuit depth, gate distribution, nearest neighbour cost and transistor cost, should also be considered as doing so can give a varied view from just considering QC alone as has typically been done [15].

Implementing reversible logic involves considerations of additional design parameters as outlined in *Table 4* [12].

Table 4. Key parameters to be considered for the efficient design of reversible logic circuits.

Parameter	Description
Ancilla Inputs (AI)	These are inputs which are held constant and are required in addition to the variable inputs to be able to achieve desired logical functions with reversible gates. The number of ancilla inputs should be minimised as they can lead to high gate

	count and impact the overall complexity of a reversible logic circuit.
Garbage Outputs (Go)	These are outputs which do not provide any useful or needed logical function. However, garbage outputs are required to maintain the bijectivity property of the circuit by ensuring the one-to-one mapping of inputs to outputs as is required for logical reversibility. The number of garbage outputs should be minimised within a circuit to help reduce resource utilisation and overall circuit overhead.
Gate Count	This parameter refers to the total number of individual reversible logic gates that have been used to produce a particular logic function. With the obvious link between gate count and the complexity of a reversible circuit. Effort should be made to select gates that reduce the gate count and thus the circuit complexity for efficient design.
Delay	This parameter refers to the amount of time a gate takes for the output to be produced after an input has been applied. The delay of a circuit is related to the number gates as for each gate a signal passes through the delay will increment. Ideally the delay of the circuit will be minimal, further highlighting the importance of minimising gate count. It should be acknowledged that there are several factors that can impact the delay a gate has, including its physical implementation, size and complexity.

2.3.2 Reversible Logic gates

Now some of the basic gates present in literature will be discussed [9] [10] [11]:

- NOT Gate – this is the simplest reversible gate consisting of 1 input and 1 output with a QC of 0. *Figure 1* show the symbol and logical function of a NOT gate.

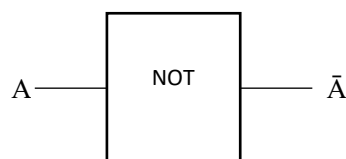


Figure 1. Logic gate symbol of reversible NOT gate.

- CNOT/Feynman Gate – this is also known as the controlled-not and is a 2x2 gate with a QC of 1. *Figure 2* shows the symbol and logical functions of a CNOT gate.

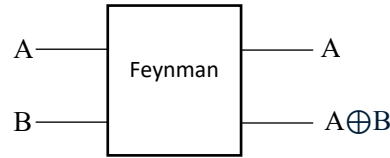


Figure 2. Logic gate symbol of reversible Feynman gate

- Toffoli Gate - this is also known as the CCNOT (Controlled-Controlled-NOT) and is a 3×3 gate with a QC of 5. The Figure 3 shows the symbol and logical functions of the Toffoli gate.

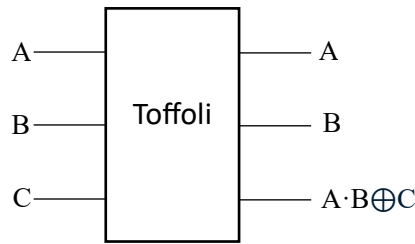


Figure 3. Logic gate symbol of reversible Toffoli gate.

- Fredkin Gate – this is also known as the CSWAP (Controlled-SWAP) and is a 3×3 gate with a QC of 5. The Figure 4 shows the symbol and logical functions of the Fredkin gate.

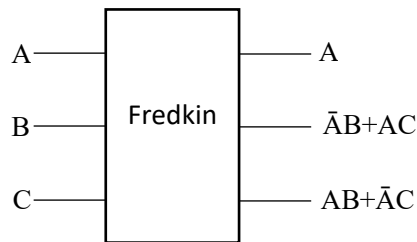


Figure 4. Logic gate symbol of reversible Fredkin gate.

- Peres Gate – this is a 3×3 gate with a QC of 4. The Figure 5 shows the symbol and logical functions of the Peres gate.

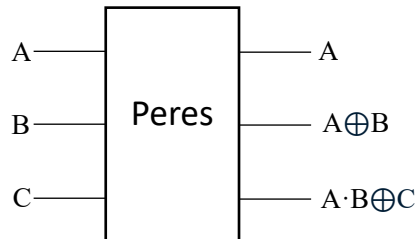


Figure 5. Logic gate symbol of reversible Peres gate.

The Toffoli and Fredkin gates are fundamental gates and are universal, meaning they can be combined to implement any logic function. The Feynman gate is a 2×2 variation of the Toffoli gate, derived from the general Toffoli definition [16]:

$$y_i = x_i \text{ for } 1 \leq i < n$$

$$y_n = x_n \oplus (x_1 \cdot x_2 \cdot \dots \cdot x_{n-1})$$

Where x represents the inputs and y the outputs, with n being the total number of inputs/outputs.

Both the Toffoli and Fredkin gates have the property of self-reversibility, allowing their outputs to be fed through an identical gate to recover inputs. In contrast, the Peres gate is not universal or self-reversible, requiring a different gate for input recovery. However, it offers practical advantage: it can replace two Toffoli gates with a lower QC, shown in *Figure 6*. Toffoli gate equivalent of a Peres gate. [17].

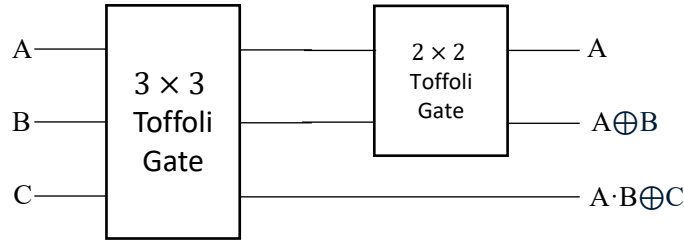


Figure 6. Toffoli gate equivalent of a Peres gate.

Non-fundamental gates, like the Peres can improve circuit efficiency. For example, [16] describes a synthesis approach using only Toffoli gates but notes that incorporating Fredkin gates can enhance performance, as one Fredkin gate is equivalent to three Toffoli gates. Similarly, [18] shows that using Peres gates can further reduce both gate count and QC. Many other gates have been proposed to optimise QC, AI and GO. Notably, the R gate, QC of 4, has shown promise in constructing reversible ALUs by supporting many basic arithmetic operations alone or in pairs [19]. Another example, the DKG gate, QC of 6, has also been used in ALU design with minimal AI, GO and QC [20] [21]. The gates presented here are by no means exhaustive and only give a brief insight into the different reversible logic gates that exist.

2.3.3 Reversible Logic Implementations

Despite significant theoretical progress, the practical implementation of reversible logic remains limited. While many gate and function designs have been simulated using HDL, and some transistor-level designs proposed, few have advanced beyond simulation. Only a few physical implementations exist. Notably MIT's VLSI demonstrator chips produced between 1996 and 1999 [22]. More recently, Vaire Computing developed a new physical approach, with a prototype fabricated in early 2025 [23]. These spares efforts underscore significant practical challenges, particularly achieving thermodynamic reversibility, a key requirement for zero-power computation as outlined by Bennett [7]. Adiabatic circuits offer a potential path forward, but its implementation remains complex.

2.4 Digital Logic Hardware Families (SV)

A digital logic family is a group of compatible devices that perform logic operations and can interconnect with each other. What follows is a brief summary of a number of different relevant or common digital logic families.

1) CMOS (Complementary Metal-Oxide Semiconductor)

CMOS logic operates by using P- and N-type transistors to connect logic lines to V_{dd} or V_{ss} respectively. By connecting these logic lines to the gates of subsequent stages, logical functions can be realised [24]. This works well and forms the basis for most digital circuits available today. However, CMOS logic has a fundamental limit on power [6]. This is because charge is dissipated to V_{ss} during a high-to-low state transition.

2) PTL (Pass Transistor Logic)

PTL is an interesting logic family as despite seeing limited commercial traction, other than very few niche applications [25], it serves somewhat as a stepping-stone towards some of the upcoming adiabatic families discussed.

Unlike CMOS, where transistors are used to tie logic lines to V_{DD} or V_{SS} , PTL operates by using its gates as switches to either pass or block a signal. The innovation arising from PTL is the Transmission Gate (T-Gate), formed by a complementary pair of CMOS transistors in parallel, controlled by a signal and its complement [26].

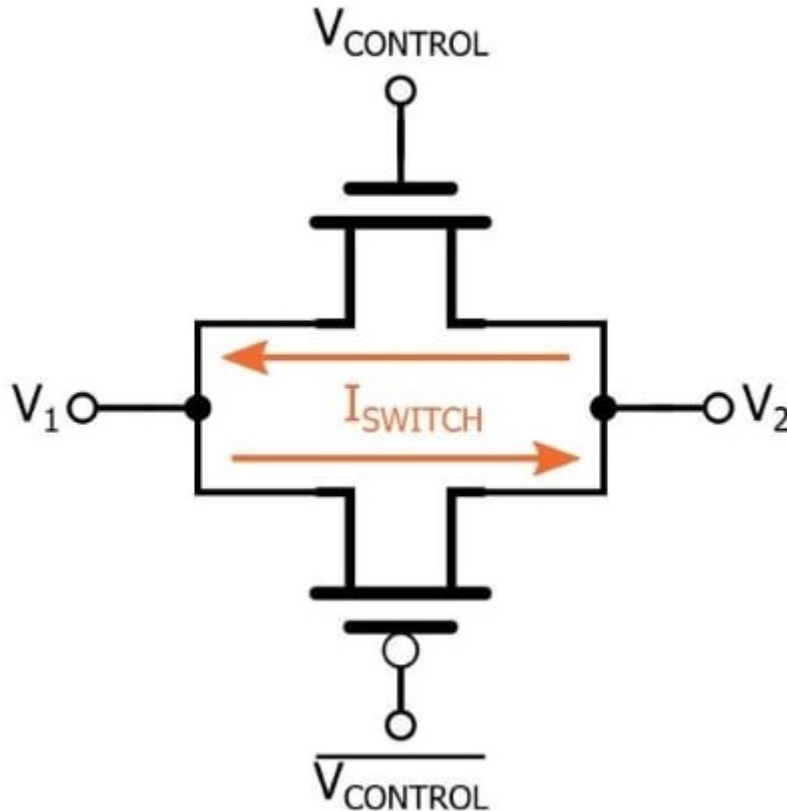


Figure 7. CMOS transistor constructions of a transmission gate [32].

A T-Gate works effectively in isolation, however the primary drawback of PTL is signal degradation when chaining gates together because of the typically high ON-resistance of CMOS transistors [27].

3) 2LAL (2 Level Adiabatic Logic)

2LAL uses previously described T-Gates, also known as the T-Cell, to conduct various power-clock signals, rather than transmitting the signal itself, avoiding signal degradation as successive stages does not degrade the same signal. S2LAL presents an interesting scheme for digital logic, whereby instead of encoding logic levels as high or low voltage levels, they are represented by active switching (tied to the clock) or by remaining idle. This allows energy to be recovered by the clock, rather than dissipating it to ground as when signal switching is timed to align with the low period of the power-clock, there is no requirement to dissipate charge to ground in the idle state. However, as may be apparent from this description, only the line capacitance holds a line in its idle state, making 2LAL inherently dynamic meaning a slow or stopped clock may cause data corruptions from various parasitic effects [22].

4) S2LAL (Static 2 Level Adiabatic Logic)

The issue of floating logic lines in 2LAL led to the development of S2LAL, which adds additional transistors that tie internal nodes to reference voltages, avoiding data corruption. It is extremely important to note that these additional transistors must be configured such that current may never flow across them. Doing so would render the circuit non-adiabatic as charge would be transferred out of the circuit through the reference line and prevent energy being recovered by the power-clock lines. [22] [28]

3. Methodology (SV)

S2LAL was chosen as the basis for both reversible and irreversible test circuits due to its balance between transistor count and the number of required power-clock lines. Alternative adiabatic circuits, like SCRL, use fewer transistors but demand many more clock lines, making measurement more complex and costlier.

The variable to be measured was the current consumption across all eight power-clock lines. With a known input voltage, this allows for the calculation and comparison of circuit power draw. Measurements were to be performed using a 0.5Ω shunt resistor in series with the test circuit, with its differential voltage amplified with a gain of 200 so that 1V output equals 1mA of current. To minimise temperature related drift over time, the measurement circuit was housed in an oven maintained at approximately 50°C.

An important note is that the de-compute pipeline typically employed in an adiabatic circuit had to be omitted from our non-reversible design. This is because the inclusion of the de-compute pipeline would necessarily make the circuit reversible, making information erasure impossible. But to retain the ability to compare the two circuits it was decided to also omit the reverse pipeline from the reversible circuit as well, which will impact the energy efficiency of the circuit, but importantly, retains the comparability of the two circuits.

4. Design and Implementation

4.1 4-Bit Adder Design (RC)

Creating the adder circuit, a 4-bit ripple carry adder topology was to be used. Being chosen for its simplicity over other adder implementations which require higher complexity irreversible gates involving greater than 2 inputs to be designed. The irreversible implementation was to use an AND/OR/ XOR construction to generate a more likely measurable information loss than a monolithic design. This is shown in *Figure 8*.

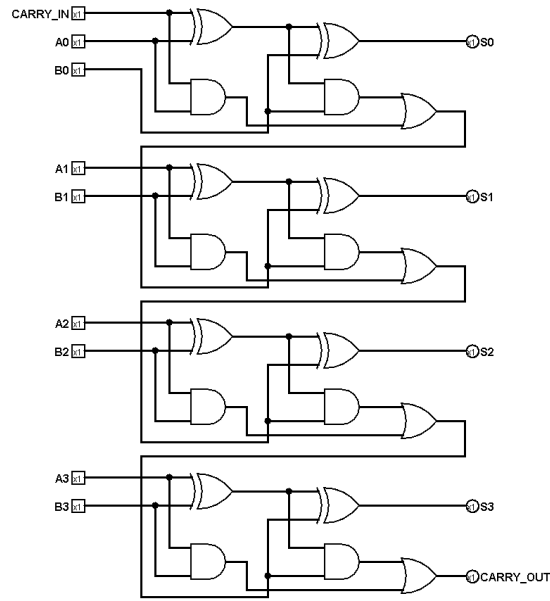


Figure 8. Logic gate level implementation of the irreversible 4-bit ripple carry adder.

For the reversible implementation of the adder the synthesis algorithm in [16] produced an adder consisting of 3×3 and 2×2 Toffoli gates. However, to reduce the number of different gates and to produce a design with optimised reversible parameters a Peres gate design was to be used instead. Figure 9 shows the reversible adder gate design.

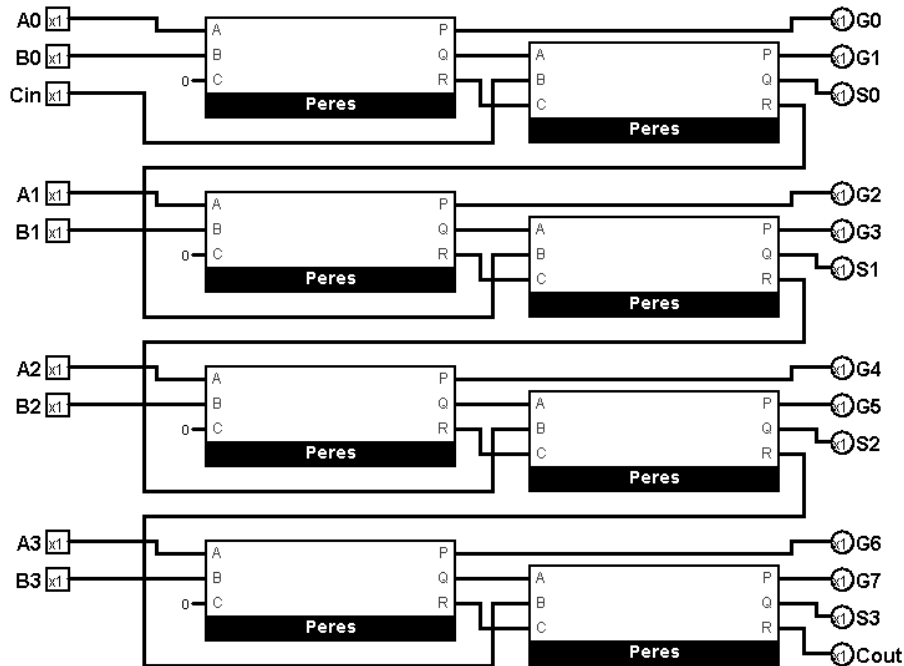


Figure 9. Logic gate level design of the 4-bit reversible adder using Peres gates.

In the operation of both RCAs it is seen logical operations occur in stages. This means for the implementation in S2LAL, with its strict signal transition timing requirements, will require signal lines to be buffered to ensure their phase align with its computation stage. The result of this requirement saw the adder design shown in Figure 10 be produced for the irreversible adder and the

reversible adder in *Figure 11*. In the reversible adder, although logically useless, the GO lines were also buffered to ensure they were not discarded and the reversibility of the function destroyed.

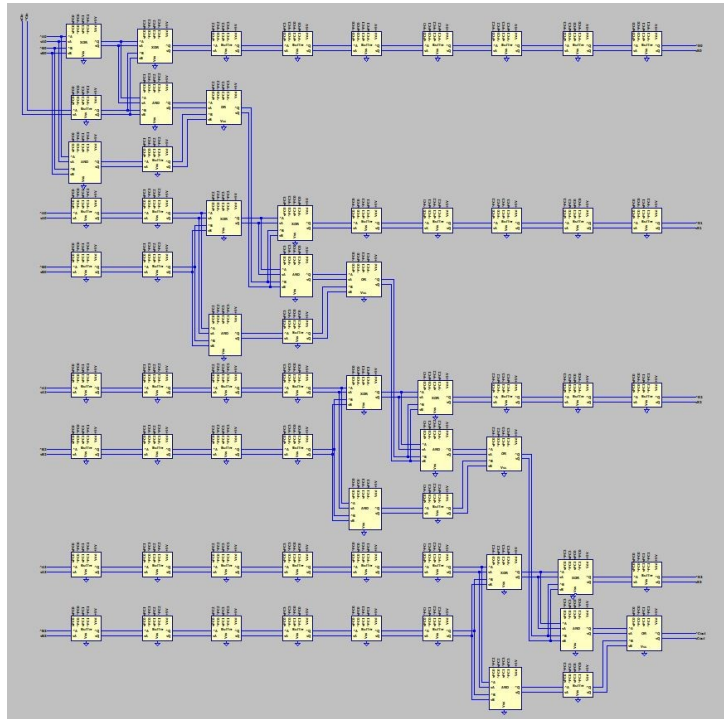


Figure 10. Irreversible 4-bit adder implementations using S2LAL Buffers, AND, OR and XOR gates.

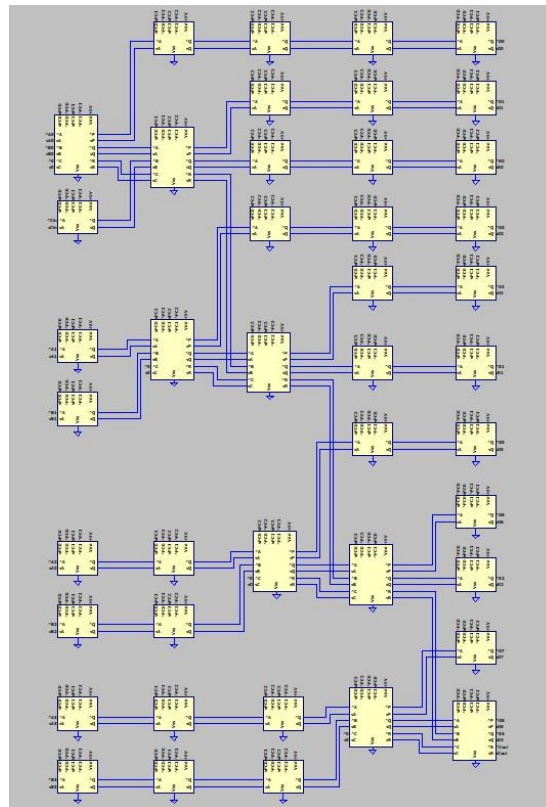


Figure 11. Reversible 4-bit adder implementation using S2LAL Peres gates and Buffers.

4.2 S2LAL Logic Gate Design

4.2.1 The CD4007 (SV)

The CD4007 is a CMOS semiconductor device currently produced by Texas Instruments, it operates as a standard CMOS inverter whilst providing two sets of complementary CMOS transistor pairs [29].

For the design of the S2LAL gates, this device was selected as opposed to discrete transistors because the CMOS manufacturing of the transistors allows the device to operate similarly to a direct silicon implementation. Additionally, as the CD4007 directly connects the body diode to VDD or VSS rather than to the source of the transistor. This allows for the transistor to be configured such that current may flow in either direction across it; this is necessary for the circuit to be thermodynamically reversible.

4.2.2 Irreversible gates (SV)

For the implementation of the irreversible ripple carry adder three gates are required: AND, OR and XOR. The designs of the AND and OR gates follow existing designs described in [28]. However, the XOR gate has no known prior implementation. Designing this gate was achieved by decomposing the fundamental operations of the S2LAL, namely OR and AND, these functions arise from the combination of T-Cells in series and parallel respectively. Then using the XOR equivalent Boolean equation $A \oplus B \equiv \bar{A}B + A\bar{B}$ allows a T-Cell implementation to be realised.

The T-Cell design alone is not static. Addressing this, additional transistors need to be added such that every node is at all times connected to a voltage reference, either a power-clock or one of the two static references. To repair the output node of the rising half of the XOR, this requires a circuit to be constructed that connects the output node to the zero-volt static reference when the output should be idle. It is trivial to see via inspection that this operation can be most easily accomplished by implementing the logical inverse of the XOR. This however still leaves the internal nodes floating, to address this without corrupting the logical function of the device however is tricky as tying the nodes to a clock or to a static reference may corrupt the proper function of the gate. In order to correct this, we may instead connect these internal nodes to the output node, assuming the output node is static, it will prevent the internal nodes from floating, whilst maintaining the correct function of the circuit.

4.2.3 Reversible gates (BS)

For the reversible ripple carry adder, only the Peres gate needs to be implemented using the S2LAL hardware topology. Since the Peres gate performs three different functions – a buffer, an XOR and a 3-input function $a \cdot b \oplus c$ – its design can be divided as such. The buffer and XOR elements can be reused from the existing irreversible gate designs, leaving only the $a \cdot b \oplus c$ function to be developed. Starting with an XOR gate as a base, one input was modified to implement an AND operate instead. Initially, a T-Cell only design was first created for the rising half of the gate. Ensuring there would be no floating nodes, as per the requirements of S2LAL, individual transistors were added to ground nodes when inactive. This was a challenge due the gate's complexity, but it was discovered that in the case of the output node the transistor network should only conduct when the output is intended to be idle. This insight enabled optimisations to minimise transistor count resulting in the final T-Cell level design of the rising side, shown in *Figure 12*. A falling signal complement was also designed to make the full gate function.

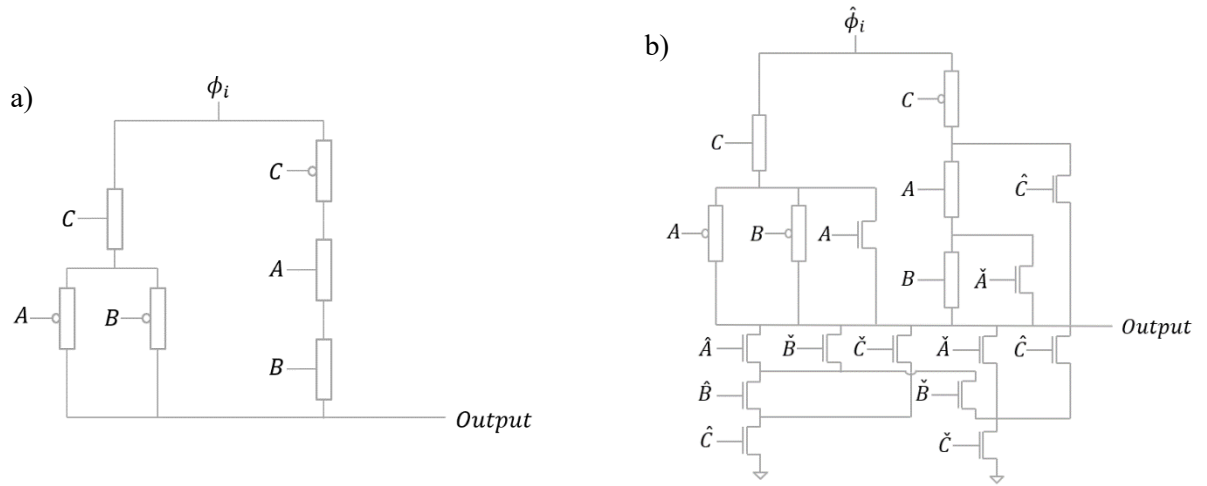


Figure 12. T-Cell design for the $A \cdot B \oplus C$ operation of the Peres gate. a) Initial T-Cell only design created to fulfil the required logic operation. b) T-Cell design with additional MOSFETs being used so no node within the circuit is able to float and the gates operation is static.

Following this T-Cell level design, this was transformed into a CD4007 implementation shown in Figure 13.

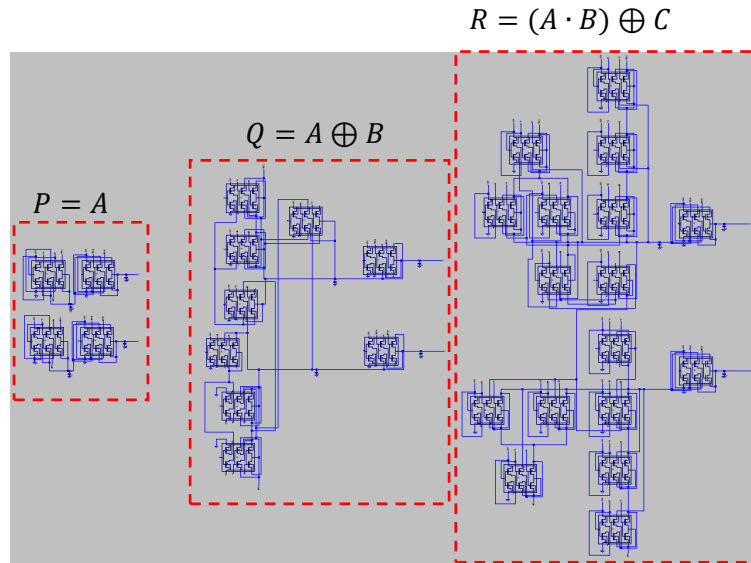


Figure 13. CD4007 implementation of the Peres gate created within LTSpice. Each dashed box highlights the individual sections performing a logical operation.

4.3 Simulations and Proof of Concept Testing (BW/BS)

Validation testing of the logic gate designs and CD4007 implementations began with SPICE simulation for all gates. Physical proof-of-concept prototypes of the Buffer, AND, and XOR gates were then constructed and tested.

4.3.1 SPICE Simulations

For the simulations the following setup configurations were used:

- Power-Clocks were 10kHz trapezoidal waves with a 5V high level and 0V low level, using 12.5 μ s leading and trailing edge times and 37.5 μ s high and low periods. See *Figure 14*.

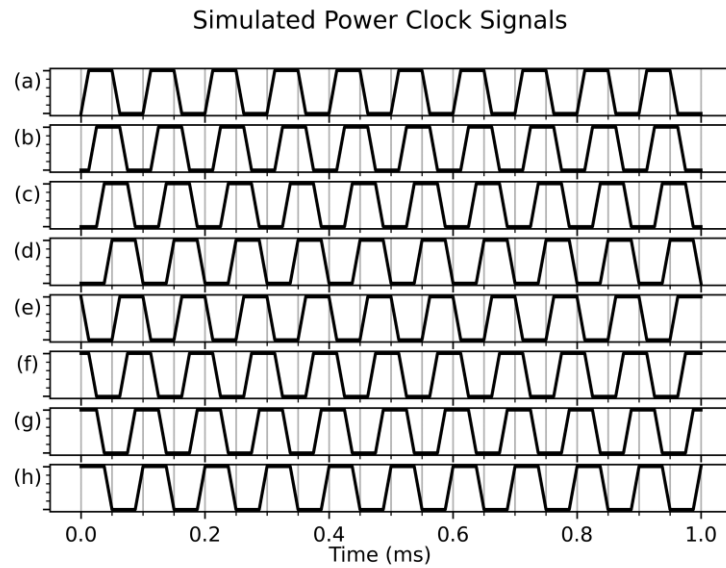


Figure 14. The eight 10kHz trapezoidal waves used in the S2LAL logic gate simulations. Waveforms (a) to (d) show the rising power-clocks 0 to 3 respectively and waveforms (e) to (h) show their respective falling, electrically inverted, complements.

- Inputs signals were applied as shown in *Figure 15* (logic high: 5V, logic low: 0V). These inputs were provided by pulse sources which were carefully timed to ensure the inputs only changed states during power clock idle periods. Also, the timings between the A, B and C inputs ensured every input case would be applied to the gate.

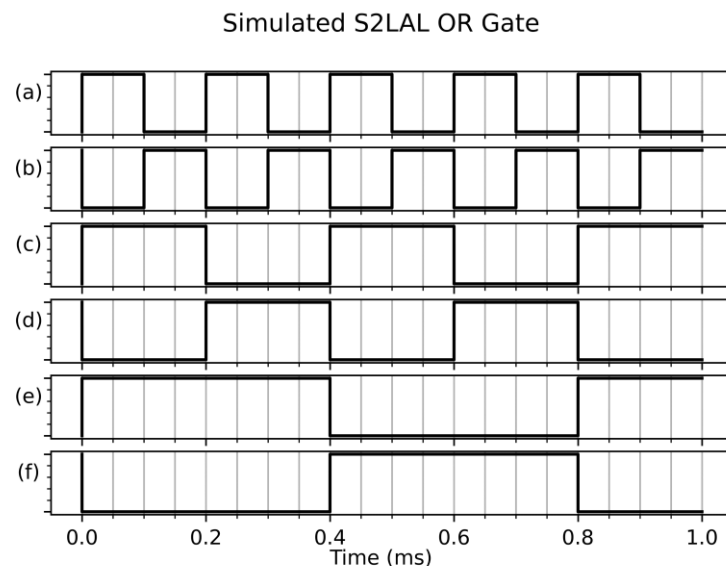


Figure 15. The input signals used with the simulations of the S2LAL logic gates. (a) shows the rising signal of inputs 'A' and (b) it's falling complement. (c) shows the rising signal of input 'B' with its falling complement shown in (d). (e) shows the rising signal of input 'C' with its falling complement shown in (f). Applying these input signals to a gate will result in every input case being applied.

Simulation results, for the LTSpice netlists in *Appendix B: LTSpice Netlists of CD4007 Gate Implementations*, confirmed correct behaviour for each gate, matching expected outputs across all

input cases. Full SPICE outputs are provided in *Appendix D: SPICE Simulation Outputs for S2LAL Logic Gates*.

4.3.2 Proof-of-Concept Gates

The proof-of-concept gates were constructed using CD4007 ICs on matrix board, see Figure 16, Figure 18 and Figure 20. Testing their operation, power-clock signals, similar to those used in simulation, were generated at 100kHz and input signals applied as trapezoidal waves, with stable active periods during the input power-clock's pulses. Figure 17, Figure 19, and Figure 21 present the input and output waveforms of each gate. Testing confirmed that each gate correctly performed its intended logic function, validating the designs for PCB implementation.

Buffer

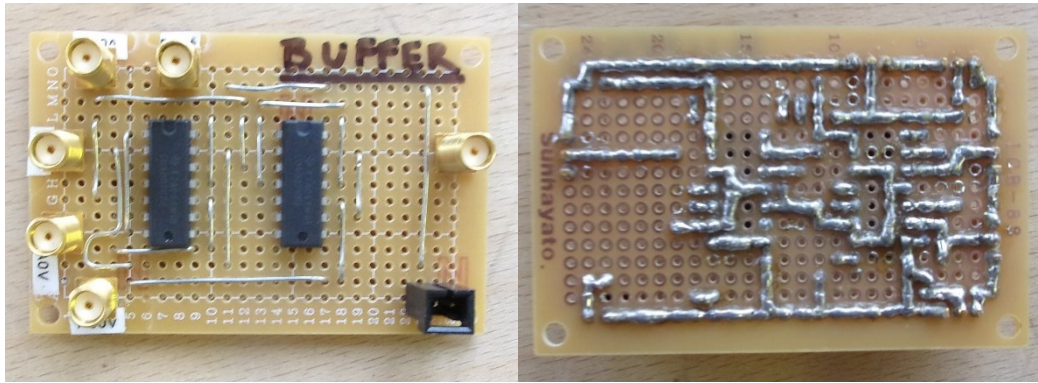


Figure 16. The proof-of-concept buffer created on matrix board. Only the rising half of the Buffer was created to demonstrate its operation.

Proof of Concept Buffer Input/Output Signals

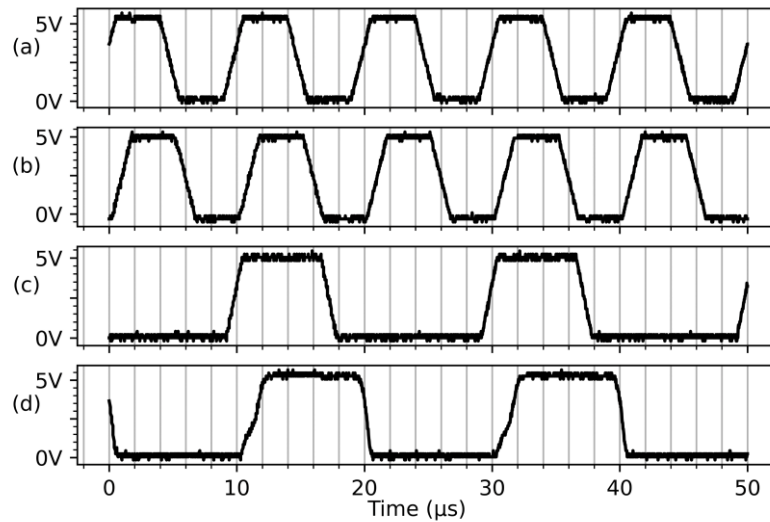


Figure 17. Input and output signals seen in the operation of the proof-of-concept buffer. (a) is the rising power-clock used for the gate's latch. (b) is the power-clock applied to the logic stage of the gate. (c) is the input signal applied to the gate. (d) is the output signal produced by the gate.

AND Gate

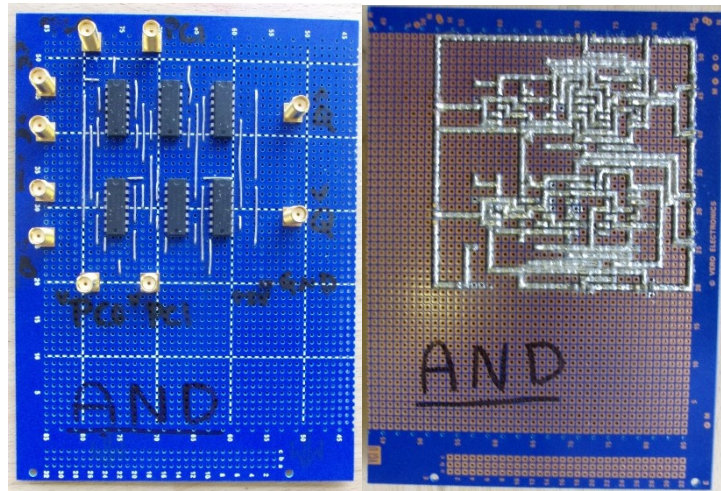


Figure 18. The proof-of-concept AND gate created on matrix board. Both rising and falling halves of this gate were implemented.

Proof of Concept AND Gate Input/Output Signals

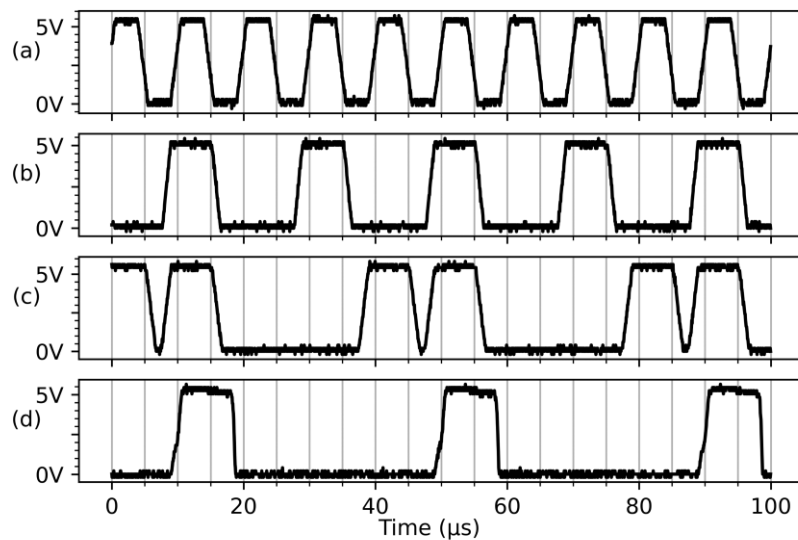


Figure 19. Input and output signals seen in the operation of the proof-of-concept AND gate. (a) is the rising power-clock applied to the logic stage of the gate. (b) is the rising 'A' input signal applied to the gate. (c) is the rising 'B' input applied to the gate. (d) is the rising output signal produced by the gate. In addition to these signals the falling complements were also measured and seen to operate correctly as the electrical inverse of these signals.

XOR Gate

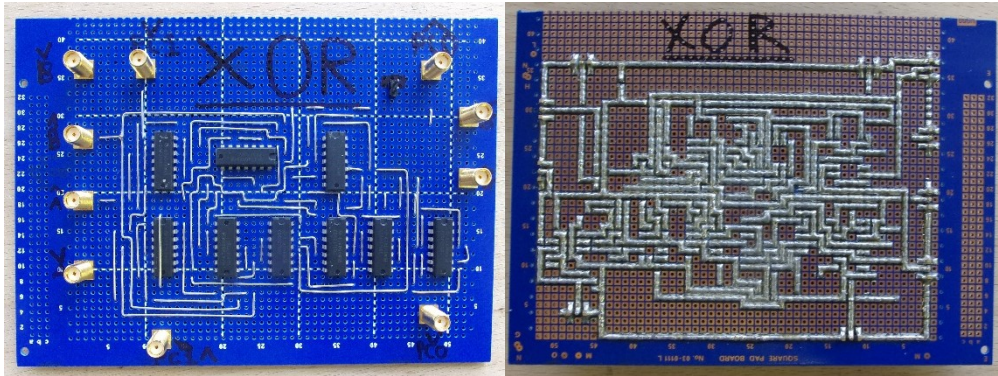


Figure 20. The proof-of-concept XOR gate created on matrix board. Both rising and falling halves of this gate were implemented.

Proof of Concept XOR Gate Input/Output Signals

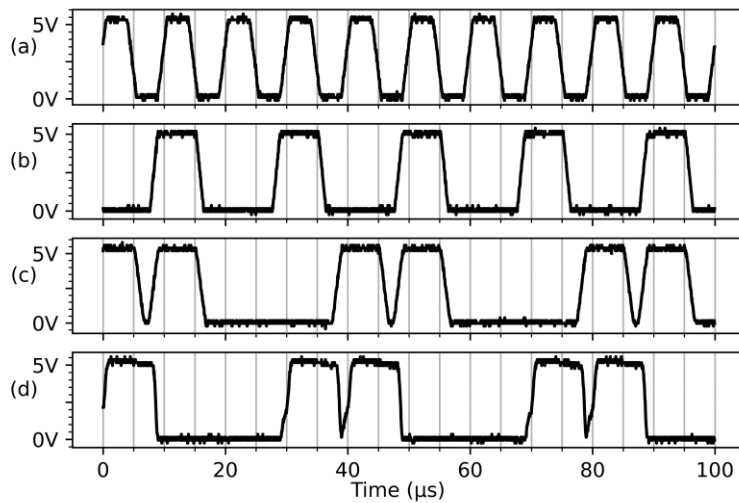


Figure 21. Input and output signals seen in the operation of the proof-of-concept XOR gate. (a) is the rising power-clock applied to the logic stage of the gate. (b) is the rising 'A' input signal applied to the gate. (c) is the rising 'B' input applied to the gate. (d) is the rising output signal produced by the gate. In addition to these signals the falling complements were also measured and seen to operate correctly as the electrical inverse of these signals.

4.4 PCB Design (BS)

To measure power consumption, each adder circuit was constructed using a modular PCB design. Individual logic gates were implemented on separate boards with connectors, allowing them to plug into a shared backplane providing the required interconnections. This modular setup aimed to simplify testing and debugging. All designed PCBs can be seen in *Appendix E: PCB Designs*.

4.4.1 Logic Gates

For the individual logic gates, the CD4007 designs were simply transferred to PCB layouts.

After gate construction, an input/output breakout board was created to facilitate individual gate testing, shown in *Figure 22*. Each gate was tested using two 1-tick out-of-phase power-clocks signals and static logic inputs to apply each input case. Outputs were monitored via an oscilloscope. All gates functioned correctly except the OR gate, where the falling half of the gate failed when both inputs were logic low resulting in the output floating. Debugging involved probing internal nets,

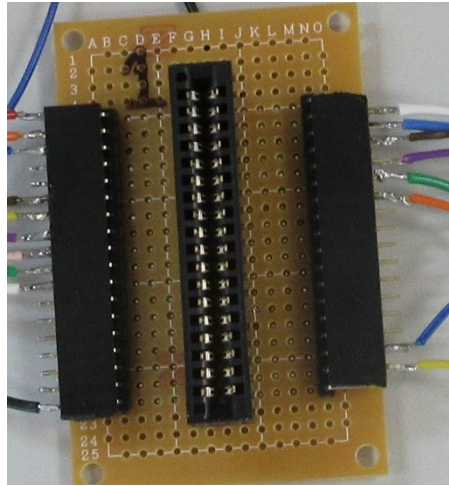


Figure 22. Logic gate breakout board constructed with a PCB edge connector reciprocal and breakout header pins to enable the easy testing of the constructed logic gates to identify correct design operation and manufacturing faults.

working back from the output, which revealed an issue with a transistor connecting the output to the static rail. This issue was replicated in simulation, confirming the design fault. The problem was resolved by changing the transistor used in the IC (Figure 23), with simulations of the revised design being successful as shown in Figure 24. On the PCB, the fix required cutting specific tracks and adding a jumper wire, which can be seen in Figure 25.

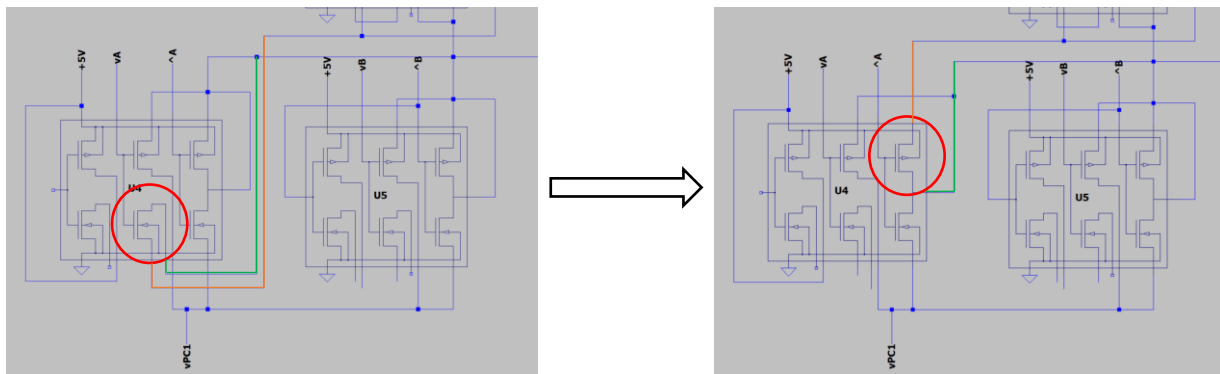


Figure 23. Change made to the CD4007 implementation of the OR gate to correct its operation. Left shows the initial design highlighting the transistor that connects the nets highlighted in green and orange. Right shows to altered design.

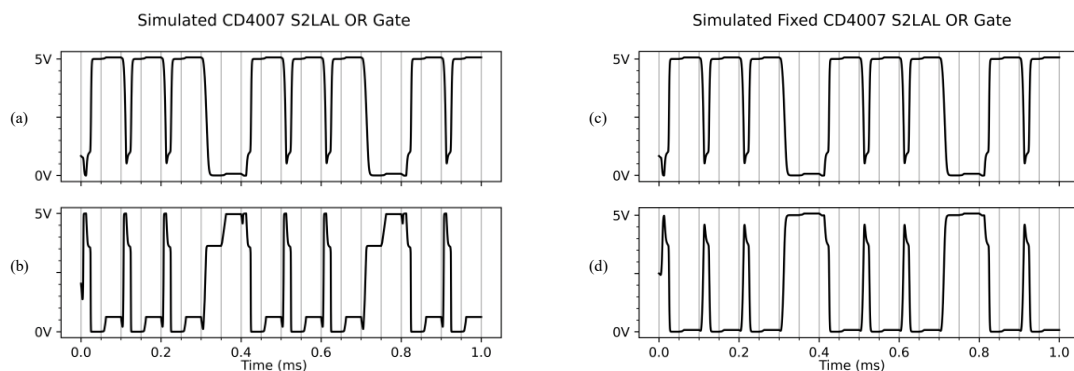


Figure 24. The simulation outputs of the original OR gate design and altered design. (a) and (b) shows the rising and falling signals of the original design. (c) and (d) show the output signals of the altered design.

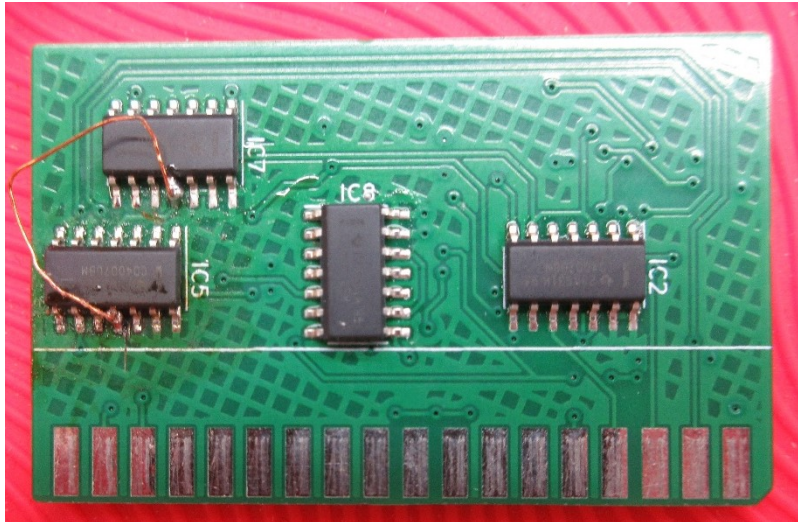


Figure 25. Modified OR gate PCB. 3 tracks connecting IC5 have been and a piece of wire soldered on to create a new connection.

4.4.2 Adder Backplates

Designing the backplate PCBs, gate connectors were arranged to minimise track lengths for inputs, outputs, and power-clocks.

To simplify interfacing and enable external measurement of the power-clocks, all inputs, outputs, and clocks were routed to a common edge connector. This allowed a shared I/O and measurement board to be used for both reversible and irreversible adders. *Figure 26* and *Figure 27* show the respective backplates.

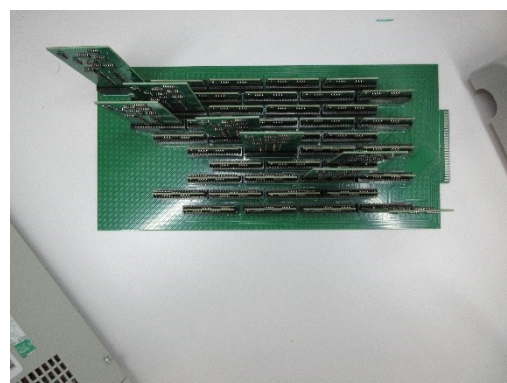


Figure 26. Constructed reversible adder backplane filled with logic gates.

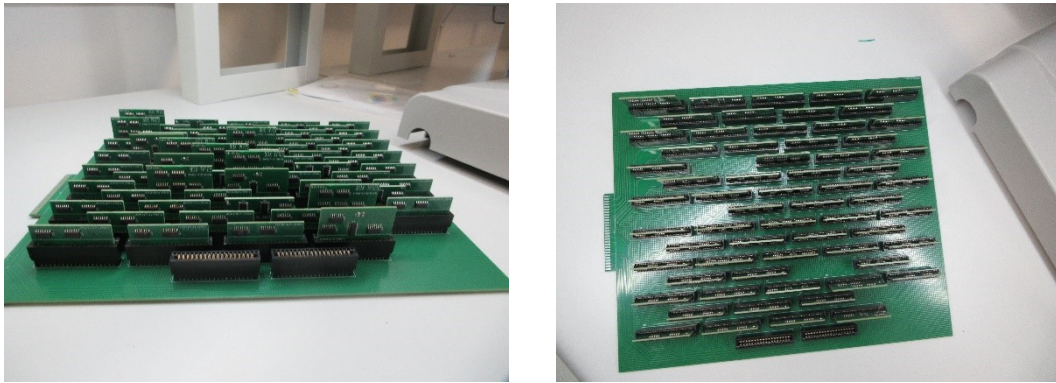


Figure 27. Constructed irreversible adder backplane, filled with corresponding logic gates.

In the testing of the backplates to check correct continuity and ensure no signal shorts were present, it was noticed there were issues with the reversible backplane PCB. Upon investigation it was found there were unconnected nets and a buffer gate was missing from the PCB. Referring to the reversible adder circuit and the PCB design files, a clear error had occurred in the transfer from circuit to PCB resulting the reversible backplane being unusable. With the vast quantity of errors found rectifying the issues would require extensive modifications to the PCB and a new PCB would need to be designed and fabricated. However, due to insufficient time remaining for the project this was unachievable meaning power measurements of the reversible logic adder could not be performed.

4.5 Measurement & Interface Board

To aid in making measurements of the performance of the circuit under test a PCB was designed which features 8 channels of buffering and current measurement for the power clocks.

The buffers were implemented with an LT1810 configured for unity gain, receiving a 50ohm terminated signal from a pulse generator. This was done as it ensures a stable voltage across a wide range of load impedances, but also protects test equipment from any potential faults.

The buffered power clocks are then connected through a 0.5-ohm current shunt resistor to the device under test, creating a voltage across the resistor obeying Ohm's law.

The resulting voltage difference is then measured and amplified by an AD8429 in-amplifier. The gain setting resistance is set by a combination of a fixed and variable resistor to allowing the gain to be adjusted between 200 and 400.

The result of this current measurement and amplification is then directed off the board to allow external test equipment to record the data.

The test and IO board was also designed to provide a TTL level interface to the device under test, allowing it to interface with test equipment for data generation and reading.

The S2LAL logic is converted to TTL by way of D-Type registers which are used to enforce the data input data transitioning in time with a user-selectable power clock

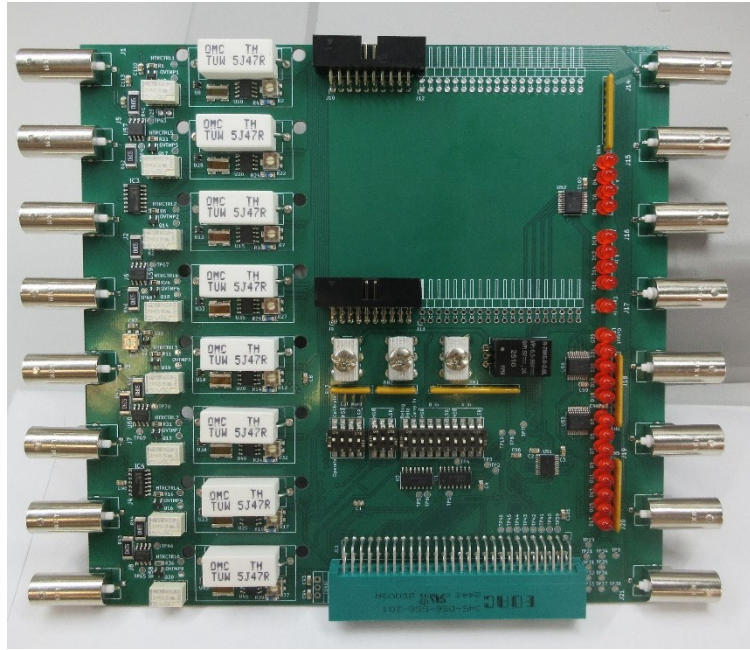


Figure 28. Constructed measurement and interface PCB.

5. Results and Analysis (BW/BS)

This section presents current draw measurements across the eight power-clock phases in the designed S2LAL circuits. The data shows how current consumption varies with clock transitions, providing an insight into energy dissipation patterns during operation. Unfortunately, measurements for the reversible adder circuit were unable to be made due to the aforementioned issues with the backplate.

Current flow was recorded across a 40ms time window for each power-clock during which the adder's operands were iterated on a 50ms cycled through every combination. These measurements reflect the transient behaviour and energy demands associated with switching events in irreversible logic. *Figure 29 - Figure 32* show the current measured over a 5ms portion of the captured data.

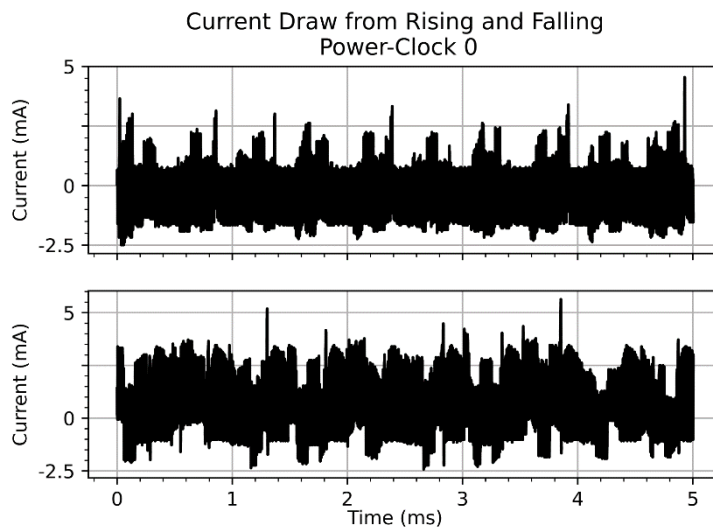


Figure 29. Measured current flow for the rising and falling power-clock 0 signal. Top is the rising variant and bottom is the falling.

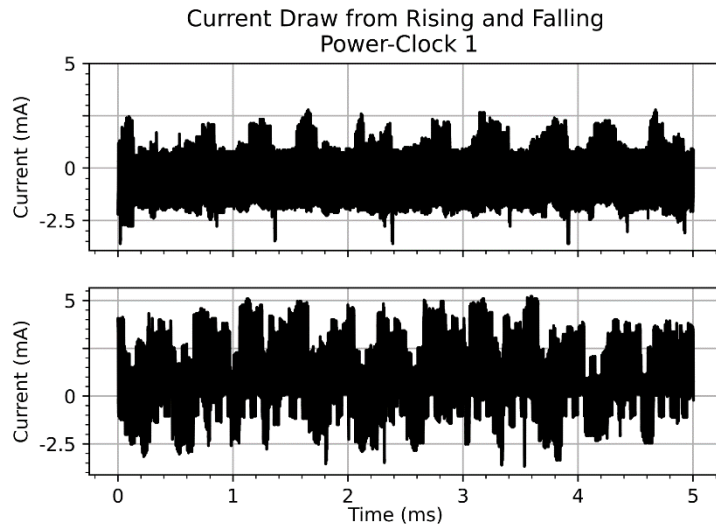


Figure 31. Measured current flow for the rising and falling power-clock 1 signal. Top is the rising variant and bottom is the falling.

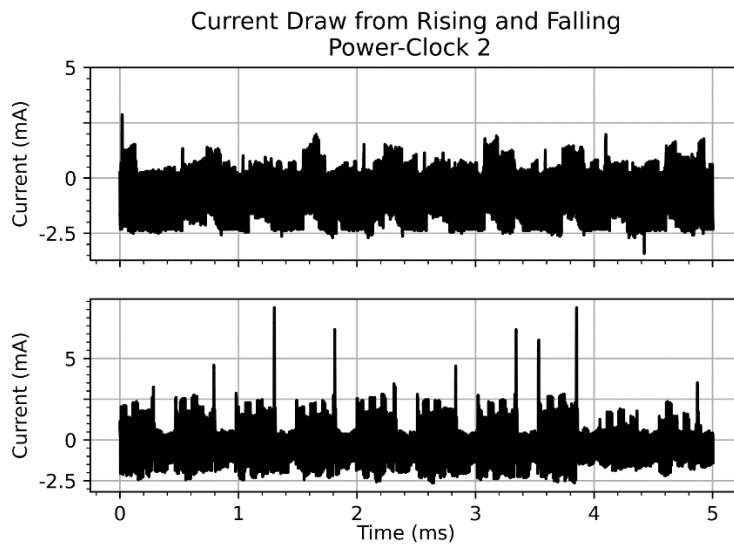


Figure 30. Measured current flow for the rising and falling power-clock 2 signal. Top is the rising variant and bottom is the falling.

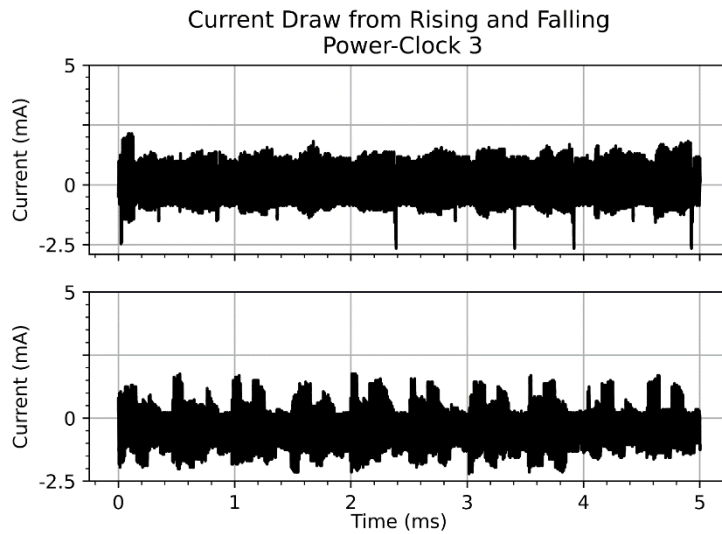


Figure 32. Measured current flow for the rising and falling power-clock 3 signal. Top is the rising variant and bottom is the falling.

Key measures from these results are presented in *Table 5*.

Table 5. Key metrics from the measured current flow of each power-clock signal.

Power-Clock Signal	Average Current (mA)	Max Current (mA)	Min Current (mA)
PC0-Rising	-0.253455	6.20769	-2.94396
PC0-Falling	0.706902	5.75971	-2.62397
PC1-Rising	-0.382208	3.03996	-3.87177
PC1-Falling	1.116039	5.27988	-3.74378
PC2-Rising	-0.700436	3.43994	-3.79178
PC2-Falling	-0.255545	8.23975	-2.76782
PC3-Rising	0.045257	2.26807	-3.61971
PC3-Falling	-0.323136	1.88408	-2.65974

6. Discussion and Evaluation (BS)

Due to the issues arising from the errors made in the design of the reversible adder backplate PCB, the ability to make power measurements of the reversible adder was unachievable. Resulting in a comparison of the power consumption of reversible and irreversible logic circuits being hampered. If a comparison could have been made it would have been expected to see on average an information loss of 17.5-bits (from the method used in [4]) for which at the approximate 21°C temperature the measurements were performed in, this would correspond to an average of $5 \times 10^{-20} J$ per computation and at the 100kHz operating frequency used is a power consumption of $4.92 \times 10^{-15} W$; due to the small magnitudes expected increases in the operating frequency and experiment time period may have been necessary to produce statistically significant results. However, the S2LAL implementation of the irreversible adder has yielded results. Therefore, although a comparison cannot be made here, the significance of implementing a successful S2LAL 4-bit adder in a quasi-adiabatic configuration – negating the de-compute logic required for full adiabatic operation – can be analysed. It should be noted a fully adiabatic implementation cannot be made irreversible as the introduction of de-compute logic transforms an irreversible design into a reversible operation [30].

The power measurement results of the irreversible adder show current flowing in both directions on every power-clock line. Seeing current flow back towards the clock source is intended for S2LAL and reveals the opportunity for energy recovery, potentially by utilising a resonant trapezoidal power-clock source. This has been highlighted as being key in the achievement of zero-power computation where the energy can be recovered and reused within a resonant circuit [28]. In a fully adiabatic implementation the net current flow would be expected to be zero (without accounting for non-ideal parasitic properties), thus for the quasi-adiabatic implementation a non-zero net current would be expected and this seen for the adder circuit with the magnitude of the combined average current across all power-clocks being 46.58µA.

Although the reversible adder cannot be assessed here, understanding the impact using S2LAL has on power consumption can somewhat be compared to documented metrics of CMOS adder ICs. The SN54F283 is a 4-bit CMOS adder IC manufactured by Texas Instruments [31]. At a supply voltage of 5V this IC sees a typical supply current of 36mA. It is clear from this the S2LAL circuit significantly out performs this particular adder IC in its required supply current and demonstrates reasonable power efficiencies can be achieved by a quasi-adiabatic implementation of S2LAL. However, S2LAL has a significant overhead requirement for its operation which will often outweigh any benefits. From this brief comparison it can be seen that S2LAL will likely only find applications

in reversible logic circuits where there is the potential to achieve zero-computation making the required overhead more tolerable.

7. Conclusion and Recommendations (SV)

In this report a practical implementation of a forward computing irreversible 4-bit ripple carry adder in S2LAL has been demonstrated, designed with a modular system of replaceable logic gates, and produced practical modular test and measurement setup. All of which should prove useful as a basis for further experimentation in S2LAL design. Due to the errors that occurred in the design of the reversible backplate and only the irreversible adder being functionally constructed within the time frame of the project, no conclusions about the Landauer's Principle and the energy dissipated due to information loss could be drawn. With additional time we would like to complete construction on the reversible circuit. This would enable a comparison between reversible and irreversible circuit to be made and inferences drawn about the effect of information erasure, based upon the results of such comparison. From such an inference a conclusion may be drawn regarding the value of further work into reversible adiabatic circuits as opposed to irreversible adiabatic circuits, direction future research.

8. Reflections

8.1 Project Management (BW)

Effective planning was essential to the successful progression of this project. To manage our workflow and ensure timely completion, a Gantt chart outlining the planned timeline for each phase of the project was created. This included stages of:

- Planning: Defining objectives, assigning roles, and setting milestones.
- Research: Studying logic gate theory, reversible computing, and hardware components.
- Simulation: Designing and testing gates in digital logic software.
- Prototyping: Building physical gates using the CD4007 IC.
- Testing: Verifying outputs and comparing them to expected results.
- Analysis: Evaluating performance, accuracy, and signal behaviour.
- Report Writing: Documenting our findings and preparing the final report.

As the project progressed, the Gantt chart was updated to reflect actual start and end dates, allowing monitoring of deviations from the original plan. This helped us identify delays, particularly in the simulation and prototyping phases, and adjust workflow accordingly.

Figure 33 shows the Gantt chart outlining project tasks and progress.

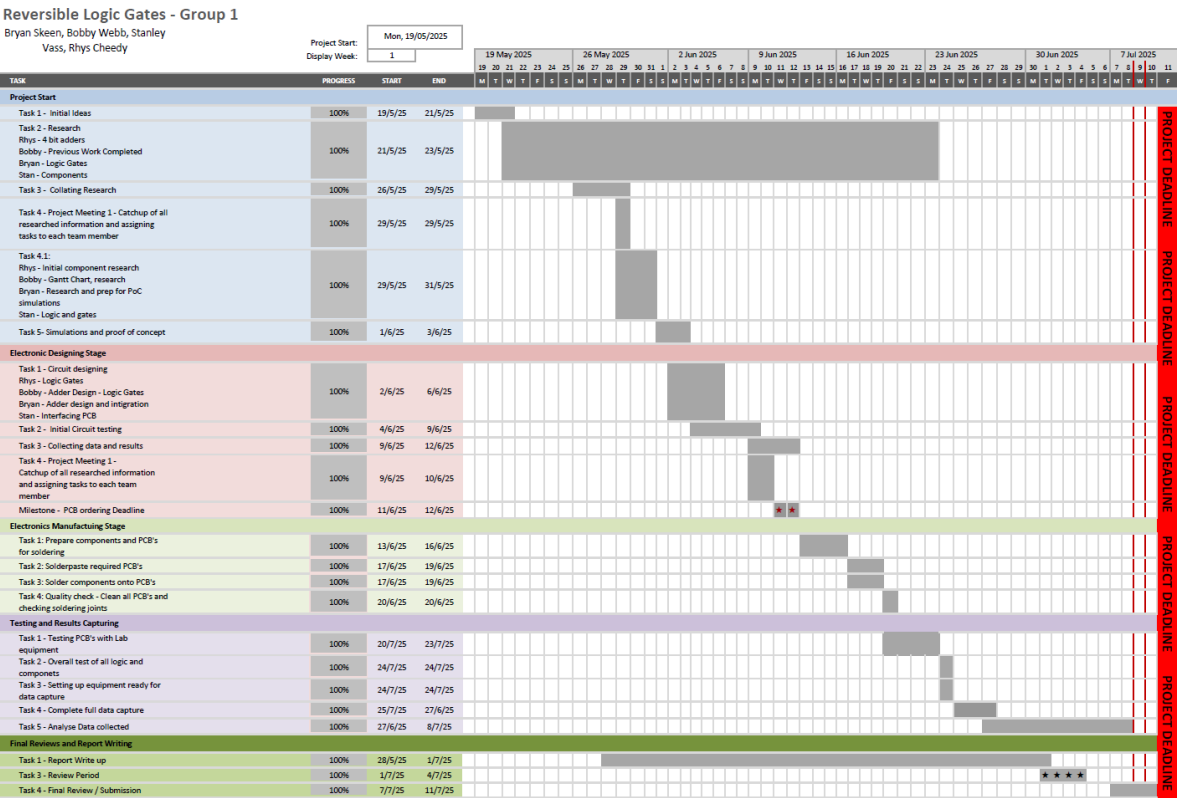


Figure 33. Gantt chart created to manage tasks and timelines of each team member for the delivery of the project.

The chart highlights some phases, such as research and simulation, taking longer than expected due to the complexity of reversible gate design and the need for additional testing. Despite these delays, the team adapted well, and the project was completed within a reasonable timeframe.

This structured approach to planning and tracking progress was crucial in maintaining focus, coordinating tasks, and ensuring all aspects of the project, from concept to execution, were thoroughly addressed.

A budget was calculated for the project; however, due to the nature of the group's research, it did end up being expensive, with the biggest cost being the express delivery due to the tight time allowance, meaning everything was required as soon as possible. See TABLE_X for the breakdown of the products in which were purchased for the project, as well as the other costs, such as delivery and import tax to name but a few.

PCB COSTSINGS		
	Cost \$	Conversion -> £
OR GATE	9.20	£ 6.84
AND GATE	12.40	£ 9.22
BUFFER	31.10	£ 23.12
PERES GATE	12.20	£ 9.07
XOR Gate	12.20	£ 9.07
Irreversible Backplate	63.20	£ 46.98
Reversible Backplate	33.40	£ 24.83
ADDER	9.20	£ 6.86
Stencils		£ 2.05
Overall Total	£	463.65

Breakdown	
Delivery Total -	£ 210.25
Import TAX Total -	£ 77.27
Component Total -	£ 176.13

COMPONENT COSTINGS		
COMPONENT NAME	QTY	Total (£)
571-2-5530843-7	115	£ 185.15
587-345-056-558-201	1	£ 8.12
595-CD4007UBM96	700	£ 196.00
71-VCS1625ZP0R500DSW	8	£ 158.88
584-LT1810CS8#PBF	10	£ 37.60
534-7800	4	£ 3.47
910-SMDLTB35T4	2	£ 39.72
Total QTY		840
Component Total -		£ 628.94
VAT -		£ 125.79
Order Total -		£ 754.73

Project Total	£ 1,218.38
---------------	------------

8.2 Stanley Vass

Throughout this project I have learnt a great deal regarding adiabatic circuit design and implementation. I have also received a great deal of practice in the operation of a variety of test equipment, and ample opportunity to practice matrix board prototyping, PCB design / implementation, 3D CAD, and experiment design.

The project also placed significant strain upon my time management abilities, having to divide my time between such a complex project and my regular work has proven a not insignificant challenge.

I have learnt some valuable lessons across the course of this project about appropriate levels of scope. For instance, initially I was intending to incorporate my own pulse generators into the design, as opposed to generating them externally with test equipment. This was entirely impractical to accomplish within the timespan which I was completely oblivious to, it took Bryan pointing out that we may not have the required time for me to realise the impracticality of building my own pulse generators.

8.3 Bryan Skeen

Through this group project, I have gained a deep understanding of reversible logic-a promising and emerging area of computing. I conducted extensive research into its evolution, from its theoretical foundations to recent advancements aiming to make it a viable computation approach. This knowledge was applied to the transistor-level design and PCB implementation of logic gates based on a proposed circuit topology.

This project allowed me to develop and apply skills in research, collaboration and circuit design. I contributed to answering the research question by reviewing literature, developing a suitable methodology, building test circuits, and analysing the results. Collaboration proved essential, as task had to be distributed efficiently across the team to meet deadlines.

Although there were some challenges which were faced during the project. Time management was a notable challenge due to the project's complexity and time constraints, including 1-week lead times for PCB fabrication and limited debugging time allocated. I also came to appreciate the importance of peer review; regular cross-checking would have helped prevent errors from propagating into the final design of the reversible backplate.

8.4 Robert Webb

Within the project, I worked to help achieve the progress we completed, as well as taking a slight 'managerial role' by completing things like the Gantt chart, keeping a look over the costs throughout the project. Within the project, I believe I helped the team make good progress by helping from research to designing the PCBs ready for manufacturing, soldering the components and testing and meeting minutes as required.

During the project, I have gained a good understanding of reversible logic and the potential it can have in the future if it were to be researched more. I looked into the previous work completed by others and found that not a huge amount made it past the simulation stage, therefore making the project worthwhile.

As well as this, a good range of skills were used and improved during the group project, i.e. working together, communication and the use of CAD systems when designing circuit boards, then testing to ensure the operation worked as intended. Working under pressure was also a well-utilised skill due to the sharp deadlines we were working to and lead times on orders, as well as shipping. Overall, a good project and I have gained new knowledge in this area and skills through working with my fellow team members.

8.5 Rhys Cheedy

During this project I had to develop an understanding in logic gates and Landauer's principle, this is because I did not understand the principle and how it applied to logic gates and data processing. I read multiple papers from research bodies and academic institutions in order to gain this background knowledge which enabled me to make calculate and informed decisions around the project. Regarding the physical application to the project I took the diagram of the layout from the software LTSpice and generated the PCB layout of the connectors in which the Peres gates and buffers would be inserted into. From there I generated the Gerber file for this specific PCB and in the time, it was taking for the external company to print the PCB I started writing my sections of the report. Once the circuit boards came, I then soldered on all of the connectors onto the board and assisted with making the gates using a stencil, solder paste and a heat gun. This entailed me learning how to use solder paste and a stencil in order to ensure every pad has solder on it. Finally, I execute the rework of one of the gates as the tracks were routed wrong on the PCB. After this I photographed and documented all of the PCBs that the group has created for the report.

For future practice in projects similar to this one I would get my work peer reviewed as one of the PCBs that I designed had a missing gate. Second opinions on this may have identified the error and saved time on the project and made it more efficient and given the group the capability to complete the project fully.

References

- [1] A. Prieto, B. Prieto, J. J. Escobar and T. Lampert, "Evolution of Computing Energy Efficiency: Koomey's Lay Revisited," *Cluster Computing*, vol. 28, no. 1, Oct. 2024.

- [2] C. E. Shannon and W. Weaver, "A Mathematical Theory of Communication," *Bell System Technical Journal*, vol. 27, no. 4, pp. 623-656, Oct. 1949.
- [3] P. Saraiva, "On Shannon Entropy and its Applications," *Kuwait Journal of Science*, vol. 50, no. 3, pp. 194-199, Jul. 2023.
- [4] I. K. Hänninen, C. S. Lent and G. L. Snider, "Quantifying Irreversible Information Loss in Digital Circuits," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 11, no. 2, Nov. 2014.
- [5] M. P. Frank, "Physical Foundations of Landauer's Principle," *International Conference on Reversible Computation*, pp. 3-33, 2018.
- [6] M. P. Frank, "The Physical limits of Computing," *Computing in Science & Engineering*, vol. 4, no. 3, pp. 16-26, May-June 2002.
- [7] C. H. Bennett, "Logical Reversibility of Computation," *IBM Journal of Research and Development*, vol. 17, no. 6, pp. 525-532, 1973.
- [8] C. H. Bennet, "Notes on the History of reversible Computation," *IBM Journal of Research and Development*, vol. 44, no. 1.2, pp. 270-277, Jan. 2000.
- [9] R. Khanam, A. Rahman and Pushpam, "Review on Reversible Logic Circuits and its Application," in *International Conference on Computing, Communication and Automation*, May 2017.
- [10] V. Singh and K. B. Ramesh, "Introduction to Reversible Logic Gates adn its Operations," *Ijrasnet Journal For Research in Applied Science and Engineering Technology*, 2022.
- [11] S. Mamatah, D. Saha, N. Banu and A. Prof, "A Review of Reversible Gates and its Application in Logic Design," *American Journal of Engineering Research*, vol. 3, no. 4, pp. 151-161, Jul. 2025.
- [12] S. F. Naz and A. P. Shah, "Reversible Gates: A Paradigm Shift in Computing," *IEEE Open Journal of Circuits and Systems*, vol. 4, pp. 241-257, Sep. 2023.
- [13] R. Wille, A. Chattopadhyay and R. Drechsler, "From Reversible Logic to Quantum Circuits: Loigc Design for an Emerging Technology".
- [14] S. S. Samrin, R. Patil, S. Itagi, S. C. Chetti and A. Tasneem, "Design of Logic Gates using Reversible Gates with Reduced Quantum Cost," *Global Transistions Proceedings*, vol. 3, no. 1, pp. 136-141, Jun. 2022.
- [15] R. Wille, M. Saeedi and R. Drechsler, "Synthesis of Reversible Logic Functions Beyond gate Count and Quantum Cost," arXiv, 2010.
- [16] A. Agrawal and N. K. Jha, "Synthesis of Reversible Logic," *Proceedings Desing, Automation and testing in Europe Conference and Exhibition*, pp. 1384-1385, Jul. 2004.
- [17] P. .. Kadbe and S. D. Markande, "Efficient Design of Reversible Adder and Multiplier Using Peres Gates," *Applied Sciences*, vol. 14, no. 20, p. 9385, Oct. 2024.

- [18] J. Donald and N. K. Jha, "Reversible Logic Synthesis with Fredkin and Peres Gates," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 4, no. 1, Mar. 2008.
- [19] R. Montaser, A. Younes and M. Abdel-Aty, "New Design of Reversible Full Adder/Subtractor Using R Gate," *International Journal of Theoretical Physics*, vol. 58, no. 1, pp. 167-183, Oct. 2018.
- [20] Krishna, "Design and Analysis of Lower Power Reversible Adder/Subtractor Circuits," *International Journal of Engineering Research and Technology*, vol. 9, no. 9, Sep. 2020.
- [21] E. V. Babu and Y. Syamala, "An Efficient VLSI Design of 4-bit ALU," *International Conference on Inventive Computation Technologies*, pp. 736-741, Jul. 2022.
- [22] M. P. Frank, "Perfectly Adiabatic CMOS Logic," 30 April 2021. [Online]. Available: <https://www.sandia.gov/app/uploads/sites/210/2022/06/UKy-talkSAND-v2.pdf>. [Accessed 01 July 2025].
- [23] D. Genkina, "Reversible Computing Escapes the Lab in 2025," *IEEE Spectrum*, 02 Jan. 2025. [Online]. Available: <https://spectrum.ieee.org/reversible-computing>. [Accessed 01 Jul. 2025].
- [24] D. Lancaster, *CMOS Cookbook*, Indiana: SAMS, 1994.
- [25] K. Shirriff, "Reverse-Engineering the Z-80: the Silicon for Two Interesting Gates Explained," Ken Shirriff's Blog, September 2013. [Online]. Available: righto.com/2013/09/understanding-z-80-processor-one-gate.html. [Accessed 01 July 2025].
- [26] N. Sako, "Pass-Transistor Logic Circuit and A Method of Designing Therof". United States of America Patent US7171636B2, 09 May 2003.
- [27] R. Keim, "Restoring Digital Signals in Pass-Transistor Logic," All About Circuits, 02 January 2019. [Online]. Available: allaboutcircuits.com/technical-articles/restoring-digital-signals-in-pass-transistor-logic. [Accessed 01 July 2025].
- [28] M. P. Frank, R. W. Brocato, B. D. Tierney, N. A. Missert and A. H. Hsia, "Reversible Computing with Fast, Fully Static, Fully Adiabatic CMOS," *2020 International Conference on Rebooting Computing (ICRC)*, pp. 1-8, 2020.
- [29] Texas Instruments, "CD4007UB Types Datasheet," September 2003. [Online]. Available: <https://www.ti.com/lit/ds/symlink/cd4007ub.pdf?ts=1751932537020>. [Accessed 01 July 2025].
- [30] W. C. Athas, L. Svensson, J. G. Koller, N. Tzartzanis and E. Ying-Chin Chou, "Lower-Power Digital Systems Based on Adiabatic-Switching Principles," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 2, no. 4, pp. 398-407, Dec. 1994.
- [31] Texas Instruments, "SN54F283, SN74F283 4-Bit Binary Full Adders with Fast Carry," October 1993. [Online]. Available: <https://www.ti.com/lit/ds/symlink/sn54f283.pdf?ts=1752059217866>. [Accessed 01 July 2025].
- [32] R. Keim, "Introduction to Pass-Transistor Logic," All About Circuits, 17 December 2018. [Online]. Available: allaboutcircuits.com/technical-articles/introduction-to-pass-transistor-logic/. [Accessed 01 July 2025].

Appendix A: Example Explaining Information Quantity

{ref: 18} describes the characterisation of the information quantity of a variable in an example of a variable containing a letter which will be presented here. Let X be a random variable, where $X \in \{'A', 'B', 'C', 'D'\}$, with the probability distribution of X 's value being:

$$P(X = 'A') = 0.25,$$

$$P(X = 'B') = 0.25,$$

$$P(X = 'C') = 0.25,$$

$$P(X = 'D') = 0.25$$

Deducing the value of X , a number of questions will be asked. The first question to be asked: *Does X belong to $\{'A', 'B'\}$?* This question will eliminate half of the possible values X could be. If the answer is yes then $\{'C', 'D'\}$ is eliminated, if no $\{'A', 'B'\}$ is eliminated. Assuming the first answer was yes, the next question could be: *Is $X = 'A'$?* From either answer to this question the value of X has now been determined. To achieve the value of X a minimum of 2 binary questions have had to be asked. Therefore, the information, therefore the entropy, of the variable X is determined to be 2 bits. Changing the probability distribution of the X 's possible values:

$$P(X = 'A') = 0.5,$$

$$P(X = 'B') = 0.125,$$

$$P(X = 'C') = 0.125,$$

$$P(X = 'D') = 0.25$$

the value of X can still be determined by asking 2 questions, however as the probability of X being certain values is not uniform so the questions asked can be altered to improve the chance of determining X in fewer questions. Since there is a 50% chance of X being 'A' it is now more suitable to first ask: *Is $X = 'A'$?* If so X has been determined in 1 question, otherwise a second question to be asked: *Is $X = 'D'$?* Again, if true X is determined in 2 questions, otherwise a third question must be asked to determine X from the 2 remaining possible values. As the number of questions needed to determine X is dependent on the value of X , the entropy of X can be found by the average number of questions that would be asked. This is taken as the probability weighted average of the number of questions, for this example seen to be:

$$P(X = 'A') \times 1 + P(X = 'D') \times 2 + P(X = 'B') \times 3 + P(X = 'C') \times 3 = 1.75 \text{ bits}$$

Generalising the entropy associated with an unknown value can be shown to be calculated in bits using:

$$H(X) = - \sum_{x_i \in X} P(x_i) \log_2(P(x_i))$$

Where X represents an arbitrary variable, which has the possibility to take on a value x_i .

Appendix B: LTSpice Netlists of CD4007 Gate Implementations

Buffer:

```
* BUFFER
XU1 0 0 ^A N001 ^PC1 vA 0 N001 N001 vA ^PC1 N001 +5V +5V CD4007
XU2 0 0 ^PC0 ^Q N001 NC_01 0 0 ^Q vPC0 N001 ^Q +5V +5V CD4007
C1 ^Q 0 1nF
XU3 0 0 ^A N002 vPC1 ^A 0 0 N002 vA vPC1 N002 N002 +5V CD4007
XU4 0 0 ^PC0 vQ N002 NC_02 0 0 vQ vPC0 N002 vQ +5V +5V CD4007
C2 vQ 0 1nF
* Power Supply
V1 ^PC0 0 PULSE(0 5 0 12.5u 12.5u 37.5u 100u)
V2 ^PC1 0 PULSE(0 5 12.5u 12.5u 12.5u 37.5u 100u)
V3 ^PC2 0 PULSE(0 5 25u 12.5u 12.5u 37.5u 100u)
V4 ^PC3 0 PULSE(0 5 37.5u 12.5u 12.5u 37.5u 100u)
V5 vPC0 0 PULSE(5 0 0 12.5u 12.5u 37.5u 100u)
V6 vPC1 0 PULSE(5 0 12.5u 12.5u 12.5u 37.5u 100u)
V7 vPC2 0 PULSE(5 0 25u 12.5u 12.5u 37.5u 100u)
V8 vPC3 0 PULSE(5 0 37.5u 12.5u 12.5u 37.5u 100u)
V9 +5V 0 5
* Input Signals
V10 ^A 0 PULSE(0 5 0 1n 1n 100u 200u)
V11 vA 0 PULSE(5 0 0 1n 1n 100u 200u)
.tran 0 1m 0
.lib CD4007.lib
.backanno
.end
```

AND Gate:

** AND Gate*

XU1 N002 ^PC1 vA N001 N002 vA 0 N001 N002 ^A ^PC1 ^PC1 +5V +5V CD4007

XU2 N001 N002 vB 0 N001 0 0 0 N001 ^B N002 N002 +5V +5V CD4007

XU4 0 0 ^PC0 ^Q N001 NC_01 0 0 ^Q vPC0 N001 ^Q +5V +5V CD4007

C1 ^Q 0 1nF

XU7 vPC1 N004 vA N003 N004 ^A 0 +5V vPC1 ^A N004 N004 N003 +5V CD4007

XU8 N004 N003 vB 0 0 ^B 0 0 N004 ^B N003 N003 N003 +5V CD4007

XU9 0 0 ^PC0 vQ N003 NC_02 0 0 vQ vPC0 N003 vQ +5V +5V CD4007

C2 vQ 0 1nF

** Power Supply*

V1 ^PC0 0 PULSE(0 5 0 12.5u 12.5u 37.5u 100u)

V2 ^PC1 0 PULSE(0 5 12.5u 12.5u 12.5u 37.5u 100u)

V3 ^PC2 0 PULSE(0 5 25u 12.5u 12.5u 37.5u 100u)

V4 ^PC3 0 PULSE(0 5 37.5u 12.5u 12.5u 37.5u 100u)

V5 vPC0 0 PULSE(5 0 0 12.5u 12.5u 37.5u 100u)

V6 vPC1 0 PULSE(5 0 12.5u 12.5u 12.5u 37.5u 100u)

V7 vPC2 0 PULSE(5 0 25u 12.5u 12.5u 37.5u 100u)

V8 vPC3 0 PULSE(5 0 37.5u 12.5u 12.5u 37.5u 100u)

V9 +5V 0 5

** Input Signals*

V10 ^A 0 PULSE(0 5 0 1n 1n 100u 200u)

V11 vA 0 PULSE(5 0 0 1n 1n 100u 200u)

V12 ^B 0 PULSE(0 5 0 1n 1n 200u 400u)

V13 vB 0 PULSE(5 0 0 1n 1n 200u 400u)

V14 ^C 0 PULSE(0 5 0 1n 1n 40u 80u)

V15 vC 0 PULSE(5 0 0 1n 1n 40u 80u)

.tran 0 1m 0

.lib CD4007.lib

.backanno

.end

OR Gate:

** OR Gate*

```
XU1 N001 ^PC1 vA N004 N001 NC_01 0 NC_02 N001 ^A ^PC1 ^PC1 +5V +5V CD4007
XU2 N001 ^PC1 vB N003 N004 vB_0 N003 N001 ^B ^PC1 ^PC1 +5V +5V CD4007
XU3 N004 N001 vB N007 NC_03 vB 0 N007 N004 ^B N001 N001 +5V +5V CD4007
XU6 0 0 ^PC0 N002 N001 NC_04 0 0 N002 vPC0 N001 N002 +5V +5V CD4007
```

```
C2 N002 0 1nF
```

```
XU4 vPC1 N009 vA N014 N017 NC_05 0 NC_06 vPC1 ^A N008 N009 +5V +5V CD4007
```

```
XU5 vPC1 N009 vB N016 NC_07 ^B 0 N015 vPC1 ^B N009 N009 N018 +5V CD4007
```

```
XU7 N009 N008 vB N013 NC_08 ^B 0 N013 N009 ^B N008 N008 N008 +5V CD4007
```

```
XU8 0 0 ^PC0 N011 N009 NC_09 0 0 N011 vPC0 N009 N011 +5V +5V CD4007
```

```
C1 N011 0 1nF
```

** Power Supply*

```
V1 ^PC0 0 PULSE(0 5 0 12.5u 12.5u 37.5u 100u)
```

```
V2 ^PC1 0 PULSE(0 5 12.5u 12.5u 12.5u 37.5u 100u)
```

```
V3 ^PC2 0 PULSE(0 5 25u 12.5u 12.5u 37.5u 100u)
```

```
V4 ^PC3 0 PULSE(0 5 37.5u 12.5u 12.5u 37.5u 100u)
```

```
V5 vPC0 0 PULSE(5 0 0 12.5u 12.5u 37.5u 100u)
```

```
V6 vPC1 0 PULSE(5 0 12.5u 12.5u 12.5u 37.5u 100u)
```

```
V7 vPC2 0 PULSE(5 0 25u 12.5u 12.5u 37.5u 100u)
```

```
V8 vPC3 0 PULSE(5 0 37.5u 12.5u 12.5u 37.5u 100u)
```

```
V9 +5V 0 5
```

** Input Signals*

```
V10 ^A 0 PULSE(0 5 0 1n 1n 100u 200u)
```

```
V11 vA 0 PULSE(5 0 0 1n 1n 100u 200u)
```

```
V12 ^B 0 PULSE(0 5 0 1n 1n 200u 400u)
```

```
V13 vB 0 PULSE(5 0 0 1n 1n 200u 400u)
```

```
V14 ^C 0 PULSE(0 5 0 1n 1n 40u 80u)
```

```
V15 vC 0 PULSE(5 0 0 1n 1n 40u 80u)
```

```
.tran 0 1m 0
```

```
.lib CD4007.lib
```

```
.backanno
```

```
.end
```

XOR Gate:

** XOR Gate*

```
XU6 N003 ^PC1 ^A N001 ^PC1 vA 0 N005 N003 vA N001 ^PC1 NC_01 +5v CD4007
XU7 N001 N004 ^B N003 N004 ^B 0 N005 N001 vB N003 N004 NC_02 +5v CD4007
XU8 N007 N002 vA N001 N004 NC_03 0 NC_04 N003 ^A NC_05 N004 NC_06 +5v
CD4007
XU9 NC_07 NC_08 ^PC0 N004 ^Q NC_09 0 NC_10 NC_11 vPC0 N004 ^Q NC_12 +5v
CD4007
XU10 N007 N006 ^A N005 N004 NC_13 0 NC_14 N005 vB NC_15 N004 NC_16 +5v
CD4007
C2 ^Q 0 1nF
XU11 N006 vPC1 ^A N002 vPC1 vB 0 NC_17 N006 vA N002 vPC1 N008 +5v CD4007
XU12 N002 N007 ^B N006 N007 ^A 0 NC_18 N002 vB N006 N007 N008 +5v CD4007
XU2 NC_19 NC_20 ^PC0 N007 vQ NC_21 0 NC_22 NC_23 vPC0 N007 vQ NC_24 +5v
CD4007
XU3 N008 N007 ^B NC_25 NC_26 NC_27 0 NC_28 NC_29 vA N007 N008 NC_30 +5v
CD4007
C4 vQ 0 1nF
* Power Supply
V1 ^PC0 0 PULSE(0 5 0 12.5u 12.5u 37.5u 100u)
V2 ^PC1 0 PULSE(0 5 12.5u 12.5u 12.5u 37.5u 100u)
V3 ^PC2 0 PULSE(0 5 25u 12.5u 12.5u 37.5u 100u)
V4 ^PC3 0 PULSE(0 5 37.5u 12.5u 12.5u 37.5u 100u)
V5 vPC0 0 PULSE(5 0 0 12.5u 12.5u 37.5u 100u)
V6 vPC1 0 PULSE(5 0 12.5u 12.5u 12.5u 37.5u 100u)
V7 vPC2 0 PULSE(5 0 25u 12.5u 12.5u 37.5u 100u)
V8 vPC3 0 PULSE(5 0 37.5u 12.5u 12.5u 37.5u 100u)
V9 +5V 0 5
* Input Signals
V10 ^A 0 PULSE(0 5 0 1n 1n 100u 200u)
V11 vA 0 PULSE(5 0 0 1n 1n 100u 200u)
V12 ^B 0 PULSE(0 5 0 1n 1n 200u 400u)
V13 vB 0 PULSE(5 0 0 1n 1n 200u 400u)
V14 ^C 0 PULSE(0 5 0 1n 1n 40u 80u)
V15 vC 0 PULSE(5 0 0 1n 1n 40u 80u)
.tran 0 1m 0
.lib CD4007.lib
.backanno
.end
```

Peres Gate:

* Peres Gate

```
XU1 N002 N003 ^C N003 ^PC1 ^C 0 N008 N003 vC ^PC1 N003 +5V +5V CD4007
XU2 N009 N002 vA N002 N003 NC_01 0 NC_02 N002 ^A N003 N002 +5V +5V CD4007
XU3 N008 N009 vB N002 N003 N013 0 NC_03 N002 ^B N003 N002 +5V +5V CD4007
XU4 N002 N004 ^A N004 N001 N005 0 NC_04 N004 vA N001 N004 +5V +5V CD4007
XU5 N002 N001 vC N001 ^PC1 vC 0 NC_05 N001 ^C ^PC1 N001 +5V +5V CD4007
XU6 N009 N002 ^B N002 N004 N014 0 NC_06 N002 vB N004 N002 +5V +5V CD4007
XU7 NC_07 N016 vA N022 N002 N020 0 NC_08 N021 vB N017 N021 +5V +5V CD4007
XU8 N022 N002 vC N008 N002 vC 0 N022 N022 vB N009 N009 N017 +5V CD4007
XU9 0 0 ^PC0 ^R N002 NC_09 0 0 ^R vPC0 N002 ^R +5V +5V CD4007
C1 ^R 0 1nF
XU10 N030 N027 ^C vPC1 N030 ^C 0 NC_10 vPC1 vC N030 vPC1 +5V +5V CD4007
XU11 N021 N025 ^B N031 N027 ^B 0 NC_11 N031 vB N027 N031 +5V +5V CD4007
XU12 N027 N021 vA N030 N027 vA 0 NC_12 N030 ^A N027 N030 +5V +5V CD4007
XU13 N027 N021 vB N030 N027 vB 0 NC_13 N030 ^B N027 N030 +5V +5V CD4007
XU14 N031 N027 ^A N032 N031 ^A 0 NC_14 N032 vA N031 N032 +5V +5V CD4007
XU15 N032 N027 vC vPC1 N032 vC 0 NC_15 vPC1 ^C N032 vPC1 +5V +5V CD4007
XU16 N027 N025 ^A NC_16 N029 N026 0 NC_17 N027 ^C +5V N025 +5V +5V CD4007
XU17 0 0 ^PC0 vR N027 NC_18 0 0 vR vPC0 N027 vR +5V +5V CD4007
C3 vR 0 1nF
XU18 N010 ^PC1 ^A N006 ^PC1 vA 0 N012 N010 vA N006 ^PC1 NC_19 +5v CD4007
XU19 N006 N011 ^B N010 N011 ^B 0 N012 N006 vB N010 N011 NC_20 +5v CD4007
XU20 N023 N007 vA N006 N011 NC_21 0 NC_22 N010 ^A NC_23 N011 NC_24 +5v
CD4007
XU21 NC_25 NC_26 ^PC0 N011 Q^ NC_27 0 NC_28 NC_29 vPC0 N011 Q^ NC_30 +5v
CD4007
XU22 N023 N018 ^A N012 N011 NC_31 0 NC_32 N012 vB NC_33 N011 NC_34 +5v
CD4007
C5 Q^ 0 1nF
XU23 N018 vPC1 ^A N007 vPC1 vB 0 NC_35 N018 vA N007 vPC1 N028 +5v CD4007
XU24 N007 N023 ^B N018 N023 ^A 0 NC_36 N007 vB N018 N023 N028 +5v CD4007
XU25 NC_37 NC_38 ^PC0 N023 Qv NC_39 0 NC_40 NC_41 vPC0 N023 Qv NC_42 +5v
CD4007
C7 Qv 0 1nF
XU26 N028 N023 ^B NC_43 NC_44 NC_45 0 NC_46 NC_47 vA N023 N028 NC_48 +5v
CD4007
XU27 0 0 ^A N019 ^PC1 vA 0 N019 N019 vA ^PC1 N019 +5V +5V CD4007
XU28 0 0 ^PC0 ^P N015 NC_49 0 0 ^P vPC0 N015 ^P +5V +5V CD4007
C9 ^P 0 1nF
XU29 0 0 ^A N024 vPC1 ^A 0 0 N024 vA vPC1 N024 N024 +5V CD4007
XU30 0 0 ^PC0 vP N024 NC_50 0 0 vP vPC0 N024 vP +5V +5V CD4007
C11 vP 0 1nF
```

```

* Power Supply
V1 ^PC0 0 PULSE(0 5 0 12.5u 12.5u 37.5u 100u)
V2 ^PC1 0 PULSE(0 5 12.5u 12.5u 12.5u 37.5u 100u)
V3 ^PC2 0 PULSE(0 5 25u 12.5u 12.5u 37.5u 100u)
V4 ^PC3 0 PULSE(0 5 37.5u 12.5u 12.5u 37.5u 100u)
V5 vPC0 0 PULSE(5 0 0 12.5u 12.5u 37.5u 100u)
V6 vPC1 0 PULSE(5 0 12.5u 12.5u 12.5u 37.5u 100u)
V7 vPC2 0 PULSE(5 0 25u 12.5u 12.5u 37.5u 100u)
V8 vPC3 0 PULSE(5 0 37.5u 12.5u 12.5u 37.5u 100u)
V9 +5V 0 5
* Input Signals
V10 ^A 0 PULSE(0 5 0 1n 1n 100u 200u)
V11 vA 0 PULSE(5 0 0 1n 1n 100u 200u)
V12 ^B 0 PULSE(0 5 0 1n 1n 200u 400u)
V13 vB 0 PULSE(5 0 0 1n 1n 200u 400u)
V14 ^C 0 PULSE(0 5 0 1n 1n 40u 80u)
V15 vC 0 PULSE(5 0 0 1n 1n 40u 80u)
.tran 0 1m 0
.lib CD4007.lib
.backanno
.end

```

Appendix C: LTSpice CD4007 Spice Model

```
.SUBCKT CD4007 1 2 3 4 5 6 7 8 9 10 11 12 13 14
*
M1 13 6 14 14 CMOSF L=5u W=100u
M2 1 3 2 14 CMOSF L=5u W=100u
M3 12 10 11 14 CMOSF L=5u W=100u
M4 8 6 7 7 CMOSN L=5u W=20u
M5 5 3 4 7 CMOSN L=5u W=20u
M6 12 10 9 7 CMOSN L=5u W=20u
*
.ENDS

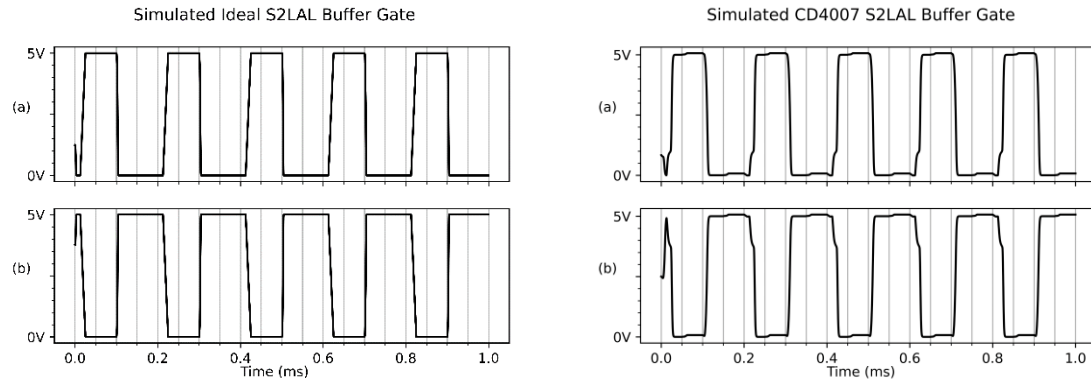
.MODEL CMOSF PMOS (LEVEL = 1 L=5u W=100u
+VTO = -1.40 KP = 3.2e-5 GAMMA = 3.30
+PHI = 0.65 LAMBDA = 9e-2 CBD = 65e-12
+CBS = 2e-14 IS = 1e-15 PB = 0.87
+CGSO = 0 CGDO = 0 CGBO = 1e-5
+CJ = 2e-10 MJ = 0.5 CJSW = 1e-9
+MJSW = 0.33 JS = 1e-8 TOX = 6.89e-10)
*$

.MODEL CMOSN NMOS (LEVEL = 1 L=5u W=20u
+VTO = 1.77 Kp = 2.169e-4 GAMMA = 4.10
+PHI = 0.65 LAMBDA = 1.5e-2 CBD = 20e-12
+CBS = 0 IS = 1e-15 PB = 0.87
+CBS = 2e-14 CGDO = 88e-8 CGBO = 0
+CJ = 2e-10 MJ = 0.5 CJSW = 1e-9
+MJSW = 0.33 JS = 1e-8 TOX = 1.265e-10)
```

Appendix D: SPICE Simulation Outputs for S2LAL Logic Gates

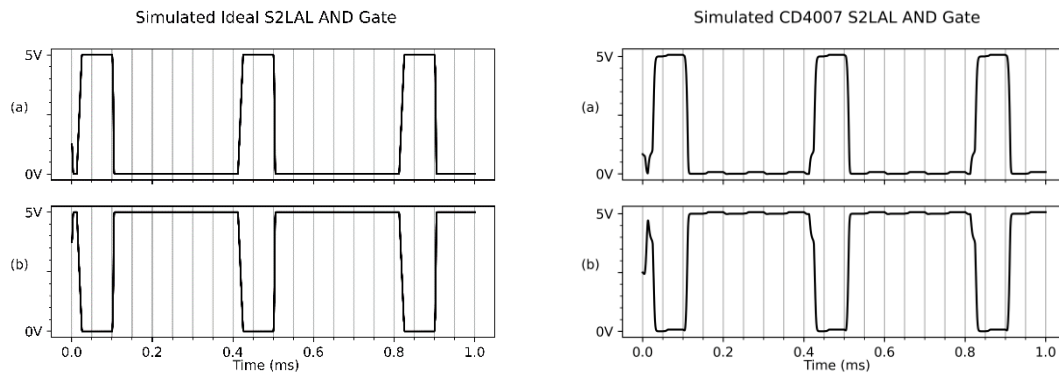
Using the simulated power-clock signals shown in FIGURE_X and the input signals shown in FIGURE_X following are the outputs signals produced by each gate design for its ideal MOSFET implementation and CD4007 implementation.

Buffer:



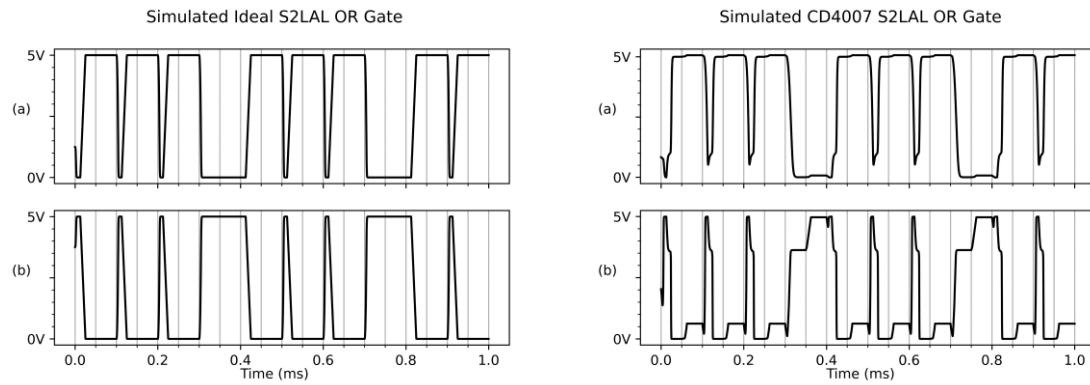
Appendix Figure 1. Left is the SPICE simulation output of the ideal MOSFET implementation of the S2LAL Buffer and right is the CD4007 implementations' outputs. For both graphs (a) shows the rising signal and (b) the falling signal.

AND Gate:



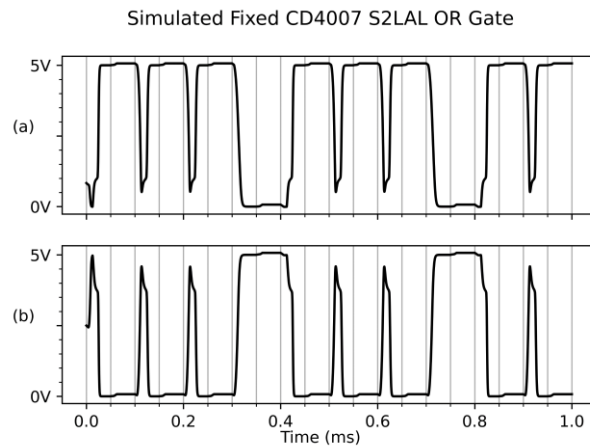
Appendix Figure 2. Left is the SPICE simulation output of the ideal MOSFET implementation of the S2LAL AND gate and right is the CD4007 implementations outputs. For both graphs (a) shows the rising signal and (b) the falling signal.

OR Gate:



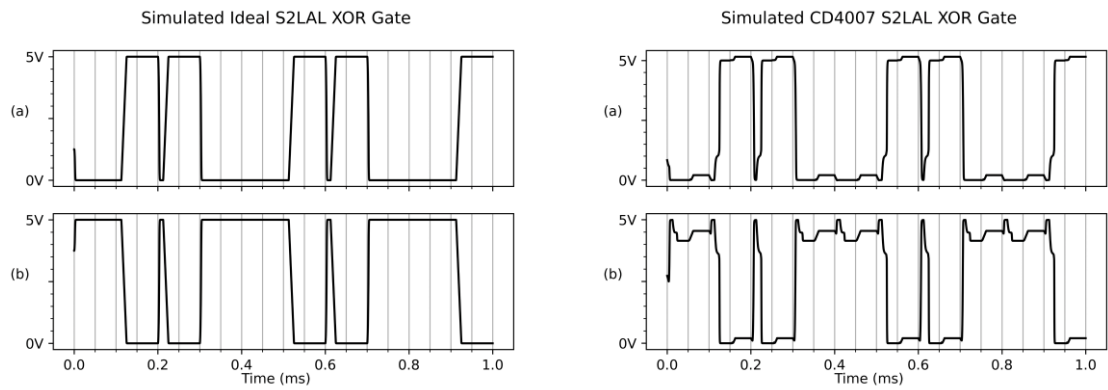
Appendix Figure 3. Left is the SPICE simulation output of the ideal MOSFET implementation of the S2LAL OR gate and right is the CD4007 implementations outputs. For both graphs (a) shows the rising signal and (b) the falling signal.

Following the PCB implementation of the OR gate and identifying an issue with its CD4407 design and the improved simulation outputs for the updated OR gate is shown here in APPENDIX_FIGURE_X.



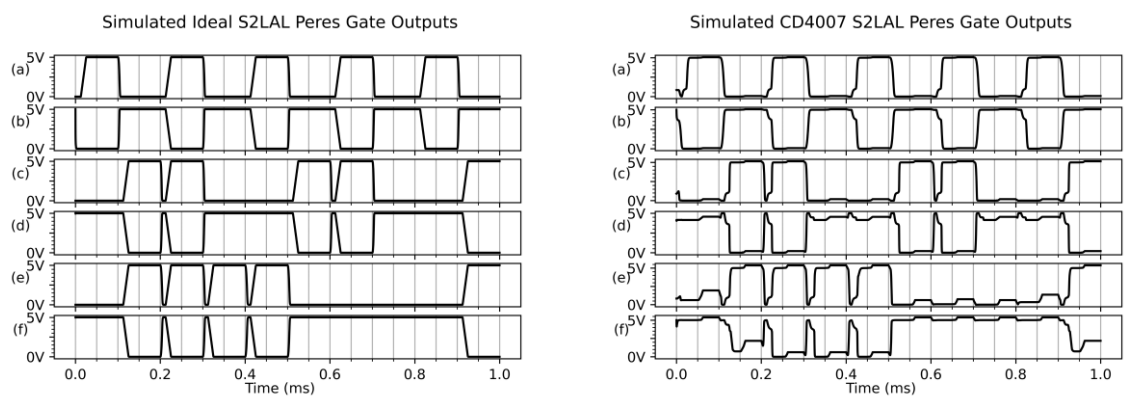
Appendix Figure 4. SPICE simulation outputs of the altered CD4007 implementation of the OR gate. (a) show the rising output signal and (b) shows its falling complement.

XOR Gate:



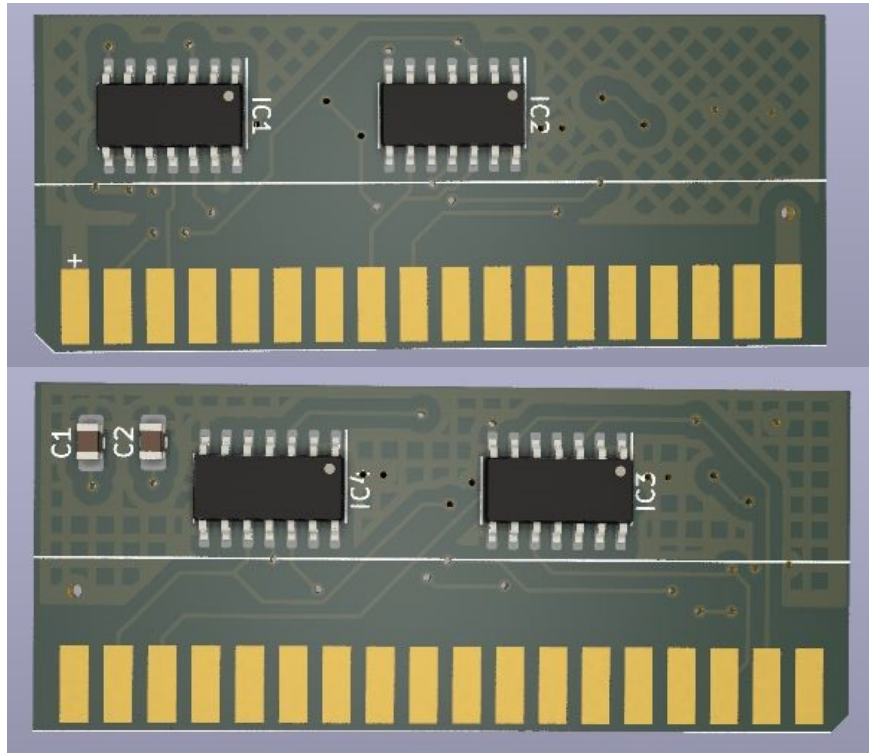
Appendix Figure 5. Left is the SPICE simulation output of the ideal MOSFET implementation of the S2LAL XOR gate and right is the CD4007 implementations outputs. For both graphs (a) shows the rising signal and (b) the falling signal.

Peres Gate:



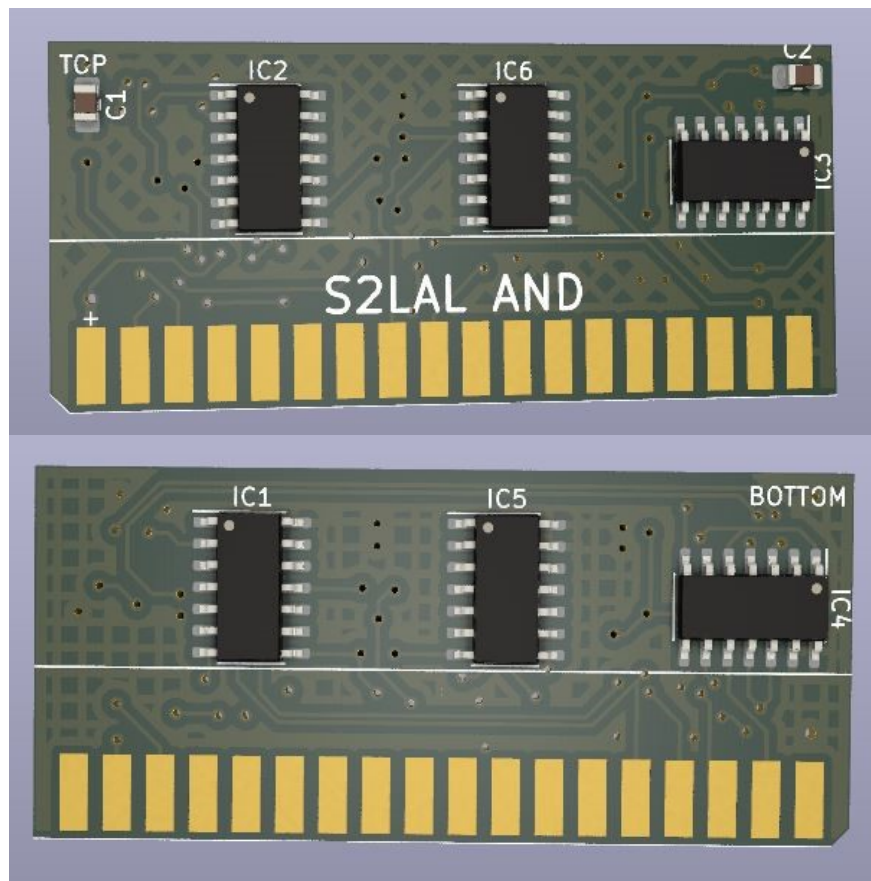
Appendix E: PCB Designs

Buffer



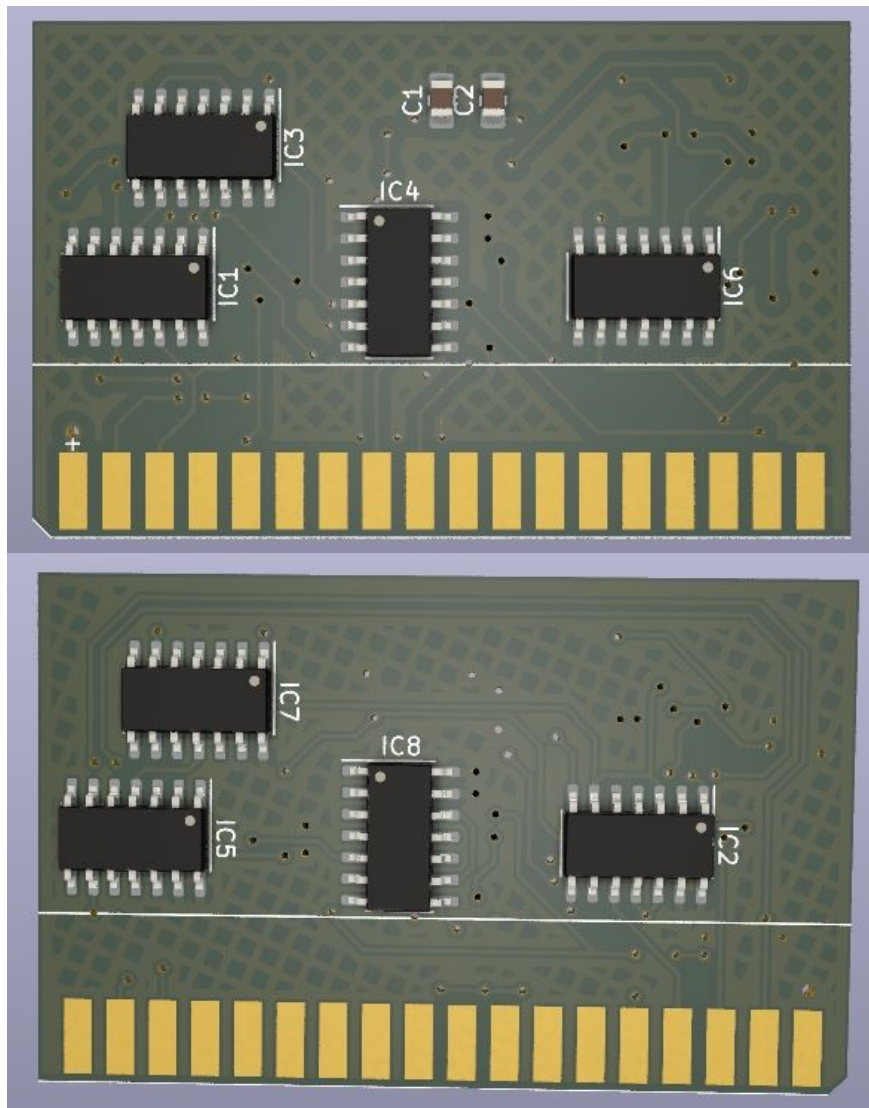
Appendix Figure 6. Buffer gate PCB design.

AND Gate



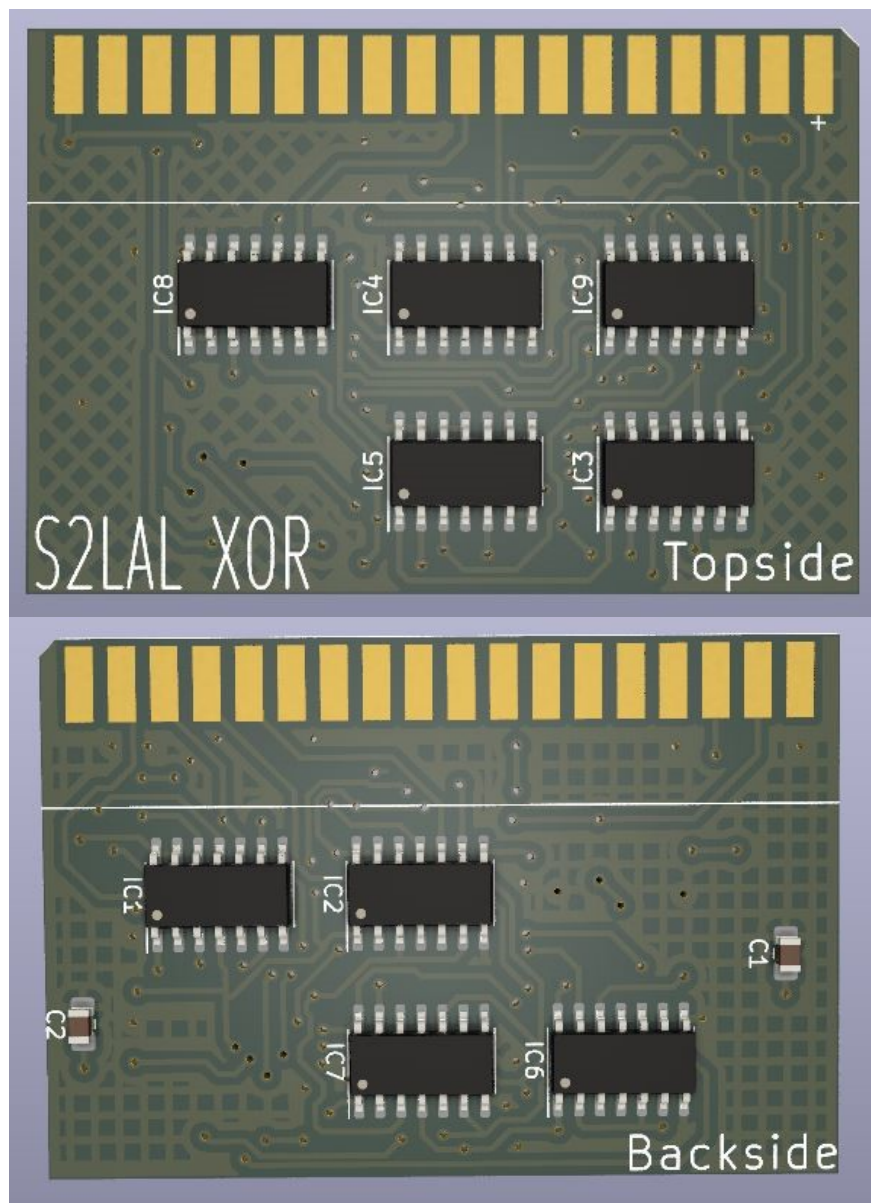
Appendix Figure 7. AND gate PCB design.

OR Gate



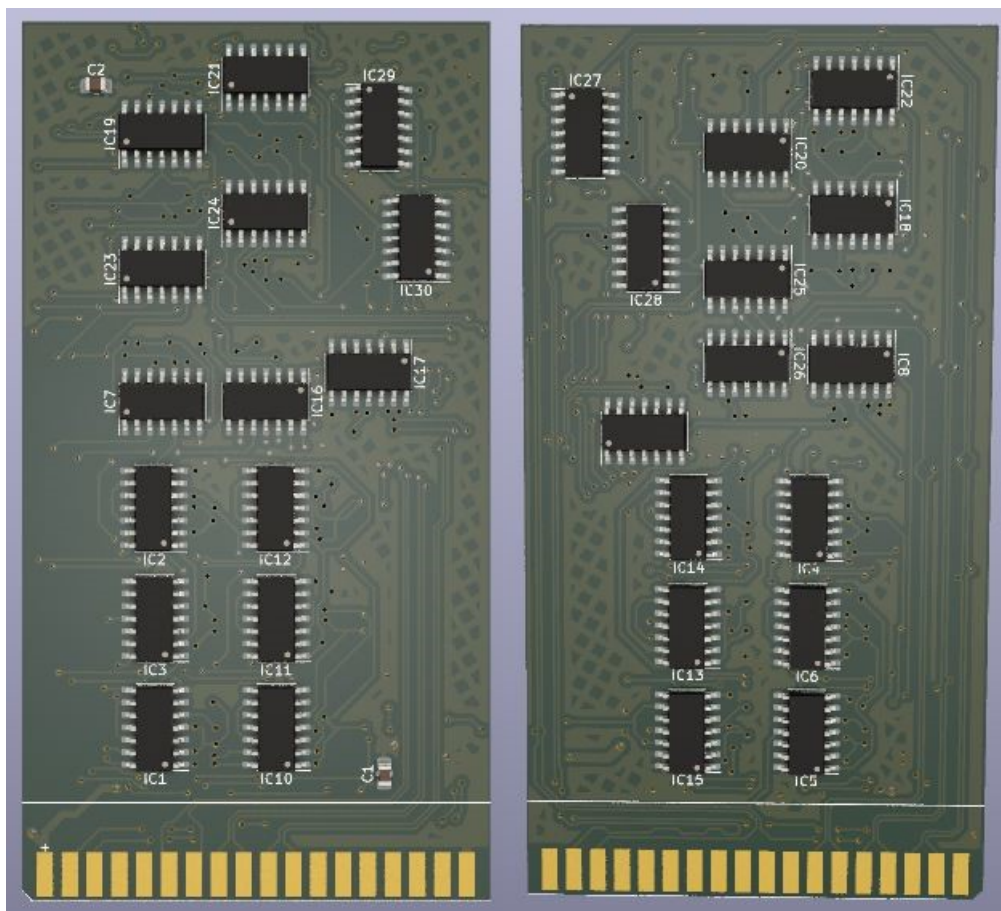
Appendix Figure 8. OR gate PCB design.

XOR Gate



Appendix Figure 9. XOR gate PCB design.

Peres Gate

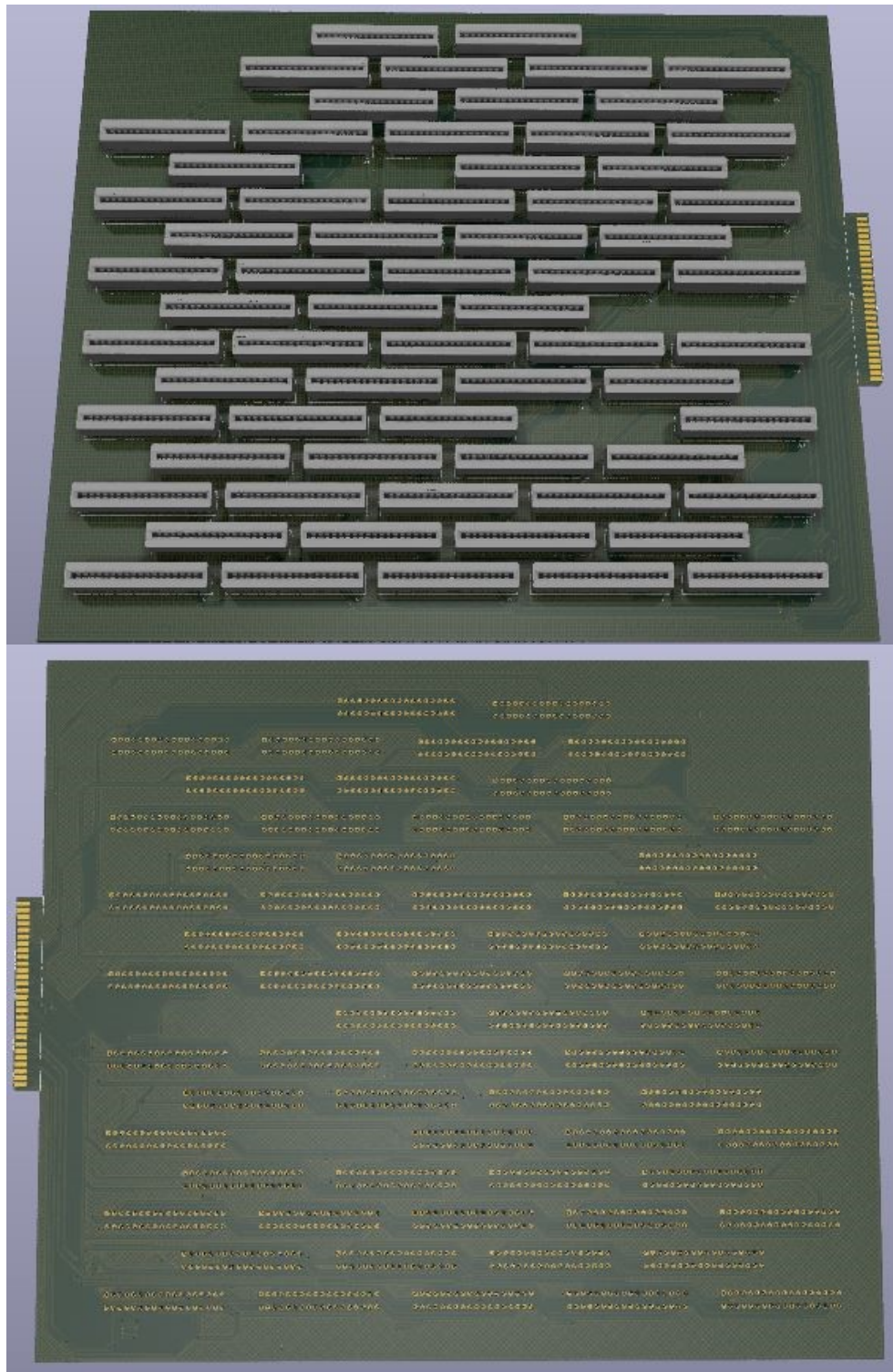


Appendix Figure 10. Peres gate PCB design.

The image displays two views of a dark green printed circuit board (PCB). The left view shows the top surface, which is populated with numerous integrated circuits (chips) of various sizes, along with other surface components. The components are arranged in a structured layout, with some larger chips and many smaller ones. The right view shows the bottom surface of the PCB, highlighting the intricate circuit traces and the gold-plated edge connector, which is used for interfacing with other systems. The gold-plated edge connector is located along the top edge of the board in this orientation.

Appendix Figure 11. Reversible adder backplane PCB.

Irreversible Adder Backplate



Appendix Figure 12. Irreversible backplane PCB design.