Experimentation and Analysis of Embedding Quantum Dot Cellular Automata on a Quantum Annealing Processor

Melvin Mathews

December 18, 2020

1 Introduction

In this project various problems were analyzed through the use of the D-Wave Quantum Processing Unit (QPU) and its associated online platform D-Wave Leap. The project involved the use of QCA Designer to design various QCA circuits, embedding these designs onto the QPU and analyzing the results generated. Specific investigations into the performance and design of an XOR circuit as well as a coplanar system are included in this document.

A QCA Designs + Embedding

This section outlines the design of two XOR circuit designs and highlights several other QCA circuits used in the later sections of this report.

A .1 Relevant Files

- `CIRCUITS/*`: All the files within this folder contain the various QCA Designer files used to model various QCA circuits. A few of the more relevant files are described below.
- `CIRCUITS/COP1`: Co-planar crossing circuit with width 1.
- `CIRCUITS/COP3`: Co-planar crossing circuit with width 3.
- `CIRCUITS/XOR_MAIN`: XOR circuit design sourced from reference [].
- `CIRCUITS/XOR`: XOR circuit designed as a composition of NAND, AND and OR gates.
- `CIRCUITS/INV`: Inverter circuit.
- `CIRCUITS/MAJ`: Majority gate circuit.
- `CIRCUITS/WIRE10`: A wire circuit composed of 10 consecutive QCA cells.

A .2 XOR QCA Design

A main task of this section was to accurately and effectively simulate an XOR circuit in QCA Designer. AN Xor gate is characterized by outputting a 1 when the inputs are different values and a 0 otherwise. Several designs were implemented and experimented with but two files `CIRCUITS/XOR_MAIN`, `CIRCUITS/XOR` in particular were chosen for further analysis. We begin by discussing the construction of the `CIRCUITS/XOR` design seen in FIG. 1

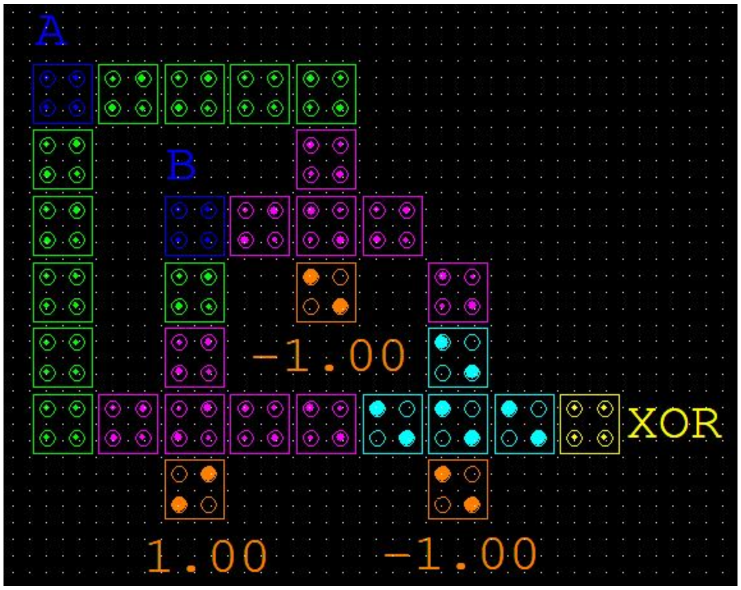


Figure 1: QCA Designer implementation of the XOR Gate as a composition of NAND, AND, and OR gates with appropriate clocking regions.

The design is based on the logic gate schematic seen in FIG. 2. The schematic was taken from reference [1] and shows how an XOR logic operation can be created using a combination of NAND, OR and AND gates.

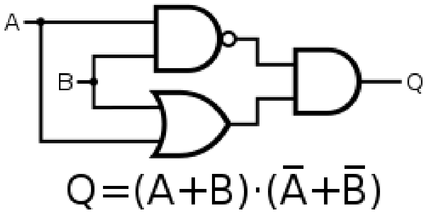


Figure 2: Schematic of a classical XOR gate used to design an XOR gate in QCA Designer. [1]

Running the circuit seen in FIG. 1, the output seen in FIG. 3 is generated. While the correct output is certainly visible there is an offset present in the output waveform when compared with the input waveforms. This is due to the different clocking regions present in the design. The FIG. 4 shows the output for the same circuit with the clocking regions removed. It can be observed that the output and input waveforms are in sync but the output is entirely non representative of an XOR gate.

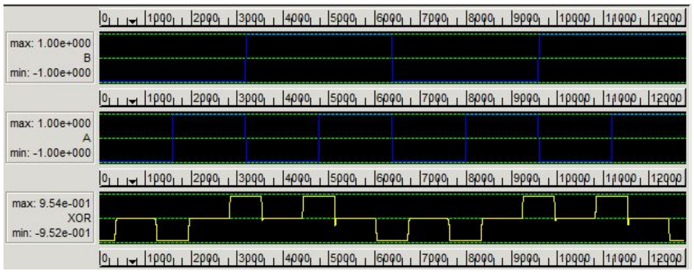


Figure 3: Input and output waveforms of the XOR circuit seen in FIG. 1 with appropriate clocking regimes.

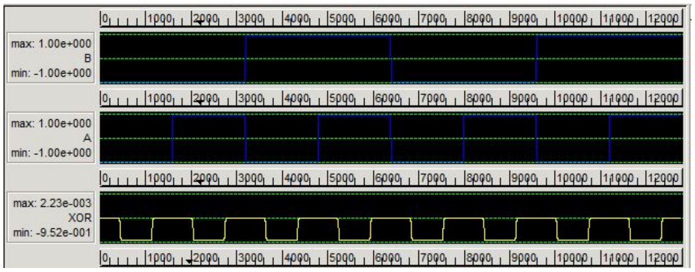


Figure 4: Input and output waveforms of the XOR circuit seen in FIG. 1 with a single clocking regimes.

While this design certainly accurately modelled an XOR gate, the team investigated the possibility of a fully functioning design with a single clocking regime. A paper by K. Navi *et. al.* [2] was relied upon in the development of the circuit seen in FIG. 5 and in the `CIRCUITS/XOR_MAIN` file.

The design is rather unintuitive but presents an elegant solution to the efficient simulation of XOR gates. It can be seen from FIG. 6 that the output follows an XOR truth table and is in phase with the input waveforms.

Many of the designs described in the relevant files section will be referenced in later parts of this report but these two XOR gate designs are of particular importance.

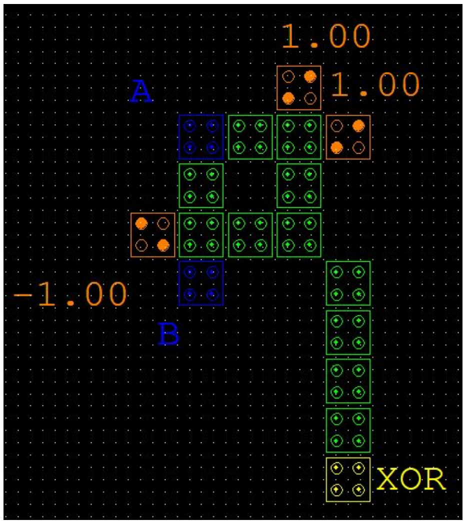


Figure 5: QCA Designer implementation of the XOR Gate with a single clocking regime. Design adapted from reference [2].

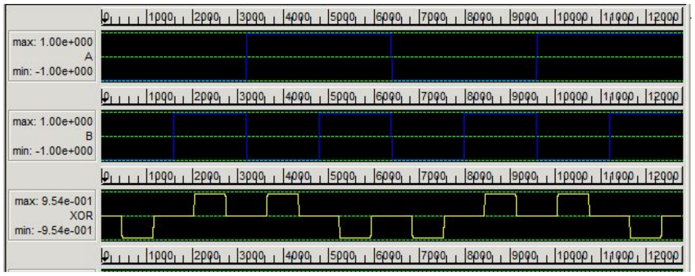


Figure 6: Input and output waveforms of the XOR circuit seen in FIG. 5.

A .3 Embedding

In this section we discuss embedding, and more specifically the embedding onto the D-wave Leap QPU architecture.

B Code Architecture

Here we outline some major design decisions on the development and structure of the code used to interface with the QCA Designer circuit designs as well as the D-Wave Leap API. The root folder contains the complete collection of the new code written for this project with the exception of small modifications to the code provided which is stored within the `util` folder.

B .1 Relevant Files

- `parse_qca.py`: Wrapper code utilizing the `qcadtrans.py` file to parse QCA Designer files into useful data structures.
- `run_qca.py`: Calculates the appropriate self-bias array and coupling matrix for an input matrix and calls the `qca_leap_minimal.py` file to interface with the D-Wave Leap API.
- `process_response.py`: For an input SampleSet data type (output from D-Wave Leap), generates a histogram associating energy levels with count of shots at that energy level. Also calculates the number of polarization values that are equal to the expected value in all the energy levels and the ground energy level exclusively.

B .2 Methods and Approach

An important principle in the development of the code base for this project was to maintain generality. It is for this reason that the mentioned files are able to parse and handle any (reasonable) size of input/output/driver cells. The typical procedure in using this architecture can be seen in any of the python notebook files that will be elaborated upon in the subsequent section. The `parse_qca` function is called for a given QCA Designer file. The output is a series of lists which are used as input into the `run_qca` function as well as the number of reads to perform and the processor (classical or QPU) to use. The output of this is fed into the `process_response` function to generate the appropriate metrics.

C Analysis

In this section we use the QCA designs developed in the first section as well as the code written in the previous section to investigate the efficacy of using a QPU for simple device simulation tasks.

C .1 Relevant Files

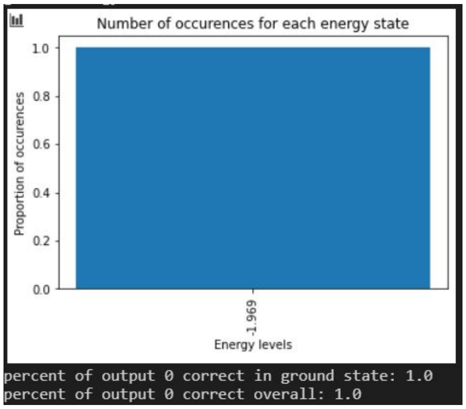
- `verification.ipynb`: Various simple QCA circuits are used to verify the implementation is working as expected. Some examples include the majority gate (`CIRCUITS/MAJ`), an inverter (`CIRCUITS/INV`) as well as a 10-cell wire (`CIRCUITS/WIRE10`).

- `crossover_analysis.ipynb`: An investigation into the performance of coplanar crossover circuits. The spectra of provided energy eigenvalues are plotted with respect to time followed by the generation of histograms for the crossover circuits `CIRCUITS/COP1`, `CIRCUITS/COP3`, and `CIRCUITS/COP5`.

- `xor_analysis.ipynb`: The XOR circuits (`CIRCUITS/XOR_MAIN`, `CIRCUITS/XOR`) are simulated with varying inputs on the QPU and a histogram of the results is generated. Comparison of the states output by the QPU and the metastable states seen in QCA Designer simulation is also conducted.

C .2 Verification

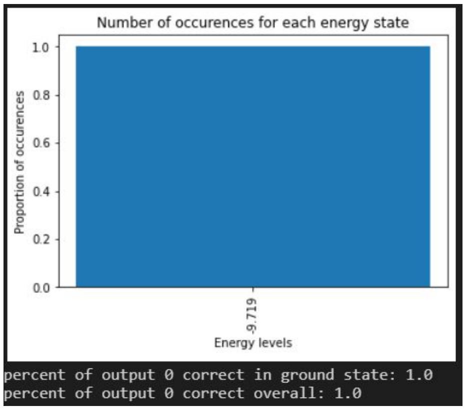
Here, the `verification.ipynb` file is run to generate histograms and percent output values for various simple devices such as the majority gate (`CIRCUITS/MAJ`), an inverter (`CIRCUITS/INV`) as well as a 10-cell wire (`CIRCUITS/WIRE10`). FIG. 7, FIG. 8, and FIG. 9 summarize the respective results.



```
percent of output 0 correct in ground state: 1.0
percent of output 0 correct overall: 1.0
```

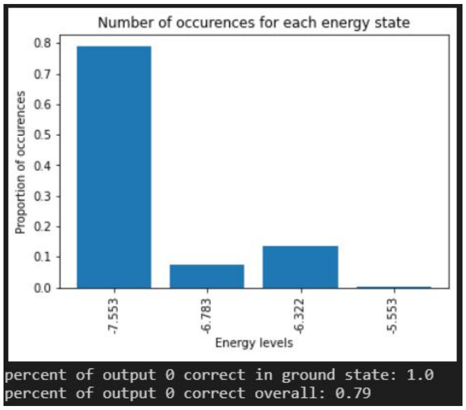
Figure 7: Histogram generated for the majority gate circuit run on the D-Wave QPU. Percentage of ground state output values that are equivalent to the expected value as well as the percentage of overall output states that are equivalent to the expected value is also included. Similar percentages were seen for all possible inputs.

We see that we achieve 100% correct values for the majority and wire circuits providing us confidence in our implementation. We also see that the inverter



```
percent of output 0 correct in ground state: 1.0
percent of output 0 correct overall: 1.0
```

Figure 8: Histogram generated for 10 cell wire circuit run on the D-Wave QPU. Percentage of ground state output values that are equivalent to the expected value as well as the percentage of overall output states that are equivalent to the expected value is also included. Similar percentages were seen for all possible inputs.



```
percent of output 0 correct in ground state: 1.0
percent of output 0 correct overall: 0.79
```

Figure 9: Histogram generated for the inverter circuit run on the D-Wave QPU. The circuit schematic for this inverter is equivalent to that provided in part a of the project description. Percentage of ground state output values that are equivalent to the expected value as well as the percentage of overall output states that are equivalent to the expected value is also included. Similar percentages were seen for all possible inputs.

circuit has the correct value for all runs that were in the ground state but develops errors as other states are introduced. This makes sense since the inverter presents a circuit with more complex interactions that can give rise to stray interactions that are likely one source of the error. In general however, all 3 circuits provide a large enough proportion of the expected output values that we can be confident in our implementation and move on to the analysis of more complex systems.

C .3 Crossover Analysis

In this section we begin by graphing the provided computed spectra values with respect to time to develop a better understanding of the behaviour of crossovers and quantum annealing systems as a whole. The plotted spectra for a coplanar crossover of width 1 and a coplanar crossover of width 3 can be seen in FIG. 10 and FIG. 11.

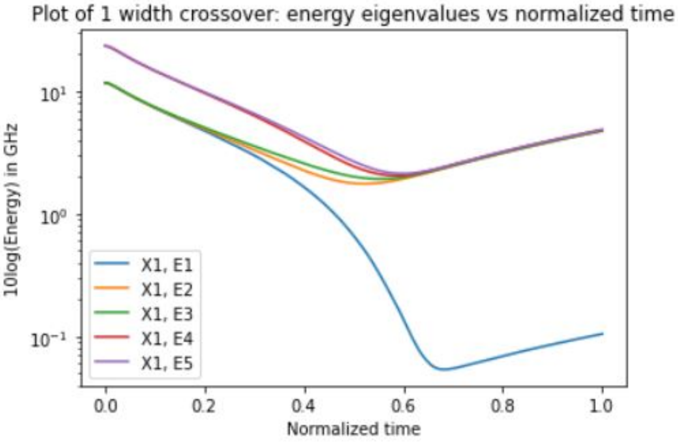


Figure 10: Plot of the provided energy spectra with respect to time for a width 1 coplanar crossing on a log Y scale.

We understand that these plotted values represent energy gaps between the ground state and respective higher energy states. The E1 on both plots denotes the energy gap between the ground state and the first excited state. As these plots both have minimum, we see that the minimum energy gap for the coplanar crossing of width 1 is approximately 0.0538 and the minimum energy gap for the coplanar crossing of width 3 is approximately 0.106. It makes sense that the minimum energy gap for the coplanar crossing of width 3 is larger as it represents a stronger coupling across the crossing portion of the circuit and thus presents a more difficult state transition. This means that the we would expect a higher proportion of runs to remain in the ground state for the width 3 coplanar crossing when compared to the width 1 coplanar crossing. This can be seen explicitly in FIG. 12

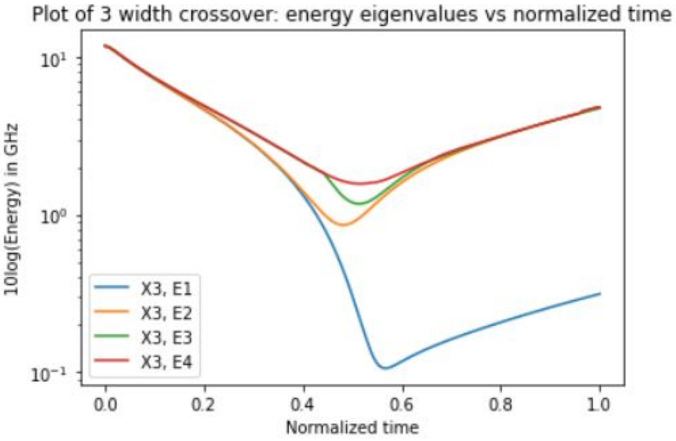


Figure 11: Plot of the provided energy spectra with respect to time for a width 3 coplanar crossing on a log Y scale.

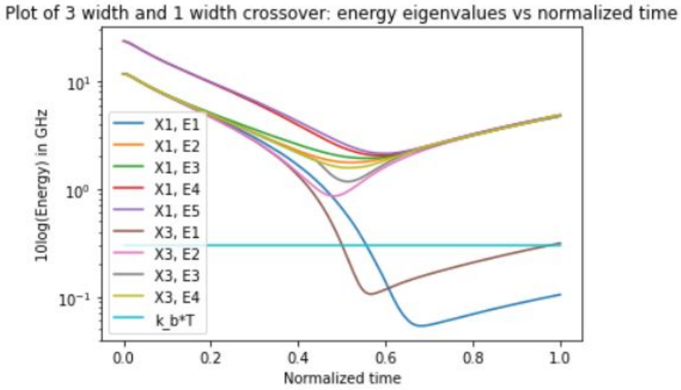
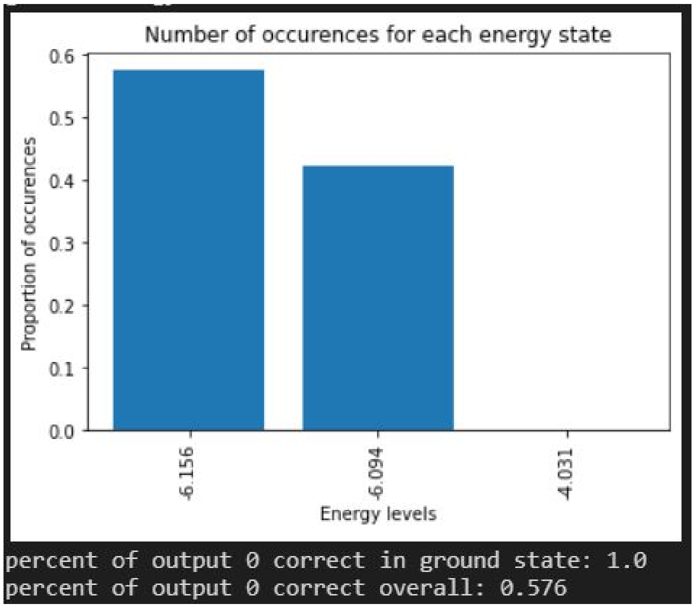


Figure 12: Plot of the provided energy spectra with respect to time for a width 1 and width 3 coplanar crossing on a log Y scale. The D-Wave Thermal Energy $(k_b T = 0.3GHz)$ value is also seen as a vertical line.

Next, we use the same procedure outlined previously to generate histograms and probabilities for the coplanar crossing circuits simulated on the QPU. The circuits can be found in These are summarized in `CIRCUITS/COP1`, `CIRCUITS/COP3`, and `CIRCUITS/COP5` these files and the results can be seen in FIG. 13, FIG. 14, and FIG. 15, respectively.



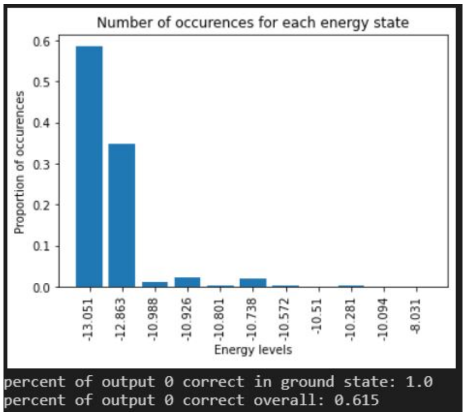
```
percent of output 0 correct in ground state: 1.0
percent of output 0 correct overall: 0.576
```

Figure 13: Histogram generated for coplanar crossing of width 1 circuit run on the D-Wave QPU. Percentage of ground state output values that are equivalent to the expected value as well as the percentage of overall output states that are equivalent to the expected value is also included. Similar percentages were seen for all possible inputs.

It is as expected that as the width of the coplanar crossing increases, there is a larger proportion of the ground state making up the overall solution. This is as predicted since it was observed in FIG. 12 that the width 3 crossover had a larger energy gap between the ground and first excited state and thus it would be in general more difficult for a state transition out of the ground state to occur for the larger width.

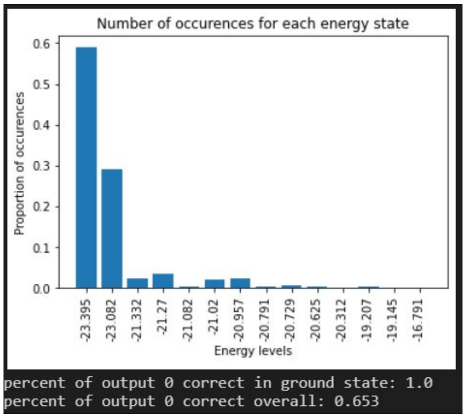
We see that the overall (all energy states including ground state) probabilities of achieving the expected value is still over 50% for all widths with a slight increase being observed as the width is increased. As this circuit was run for over a 1000 reads per run multiple times, this trend was not entirely consistent and can be attributed to the thermal excitations and thermal relaxations caused at an avoided level crossing.

We also observe that in all three graphs, the overall probability of ending up in the ground state is generally not at a level that can be used for effective,



```
percent of output 0 correct in ground state: 1.0
percent of output 0 correct overall: 0.615
```

Figure 14: Histogram generated for coplanar crossing of width 3 circuit run on the D-Wave QPU. Percentage of ground state output values that are equivalent to the expected value as well as the percentage of overall output states that are equivalent to the expected value is also included. Similar percentages were seen for all possible inputs.



```
percent of output 0 correct in ground state: 1.0
percent of output 0 correct overall: 0.653
```

Figure 15: Histogram generated for the coplanar crossing of width 5 circuit run on the D-Wave QPU. Percentage of ground state output values that are equivalent to the expected value as well as the percentage of overall output states that are equivalent to the expected value is also included. Similar percentages were seen for all possible inputs.

deterministic computation. Looking back at FIG. 12 and taking special note of the k_bT plot, we see that several excited energy state energy gaps exist above the threshold energy value. Many sources of error may exist within a QPU that would cause this threshold energy to increase allowing for increased state transition to states beyond just the first excited state. This presents one reason for the decreased ground state probability.

One final observation is that there is an effect of diminishing returns as the width is increased. While there was an increase in the proportion of ground states between the width 3 and width 5 circuits, the difference was far less then the difference between the width 1 and width 3 circuits.

C .4 XOR Analysis

In this section, we simulate the designed XOR gates in the first section of this document on a QPU processor. By running the circuit with various inputs we are able to generate the following tables showing the expected logical output, the proportion of states overall that are logically correct (P_{out}) as well as the proportion of outcomes in the ground state (P_{ground}). The first table represents the results of the `XOR_MAIN` circuit while the second table represents the results of the `XOR` circuit.

A	B	A ⊕ B	P_{out}(`XOR_MAIN`)	P_{ground}(`XOR_MAIN`)
0	0	0	0.69	0.35
0	1	1	0.69	0.40
1	0	1	0.65	0.60
1	1	0	0.54	0.50

A	B	A ⊕ B	P_{out}(`XOR`)	P_{ground}(`XOR`)
0	0	0	0.47	0.15
0	1	1	0.44	0.40
1	0	1	0.45	0.15
1	1	0	0.46	0.50

Many of the observations regarding noise and low ground state proportionality made in the crossover section can be applied here as well. Something to note is that we see results that are much more in line with the expected values for the `XOR_MAIN` circuit. This is likely due to the fact that the circuit is able to function correctly on a single clocking regime. When a circuit that uses clocking regimes is simulated on a QPU, some of the lock-step behaviour that it relies on is removed and stray interactions are more likely. Despite the QPU using no approximations in its Hamiltonian calculations, the expected output is less likely since these stray interactions can lead to unintended polarization states. Further characterization of these clocking regimes is discussed below.

Next, we want to develop a better understanding of how the metastable states seen in the QCA Designer simulations compare to the polarization values output from the QPU. We begin by comparing `XOR_MAIN` first as there is only one clocking regime and then move on to `XOR` to try and characterize the effect of the clocking regimes.

We increase the number of samples in the QCA Simulation and take a screenshot mid-simulation to produce FIG. 16 while FIG. 17 is just a screenshot of the output of the QPU.

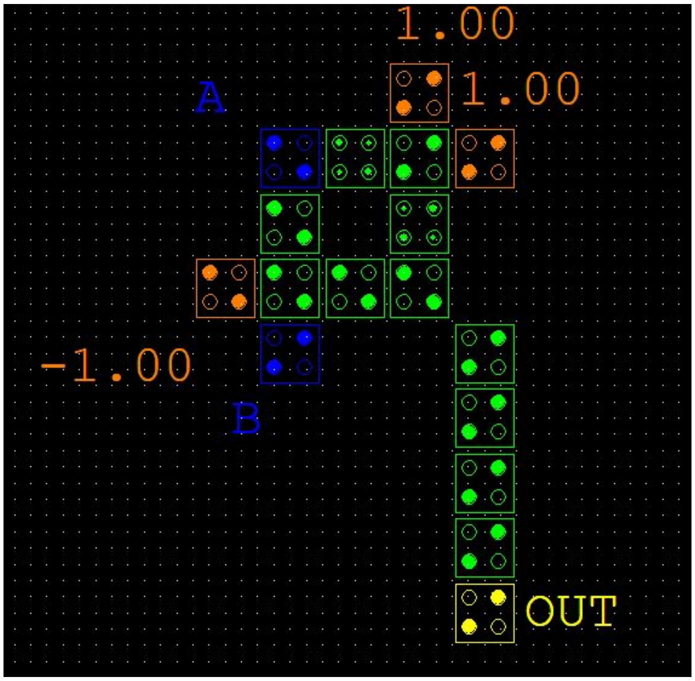
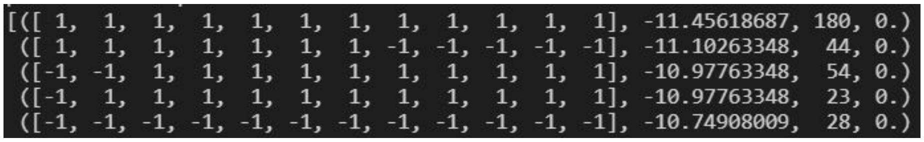


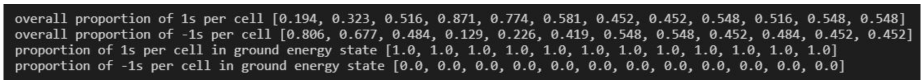
Figure 16: Mid-simulation screenshot of the circuit seen in FIG. 5. Specifically, we illustrate the polarization values mid-execution for the input state $A = -1$, $B = 1$



```
[([ 1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1], -11.45618687, 180, 0.)
 ([ 1,  1,  1,  1,  1,  1,  1, -1, -1, -1, -1, -1], -11.10263348,  44, 0.)
 ([-1, -1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1], -10.97763348,  54, 0.)
 ([-1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1], -10.97763348,  23, 0.)
 ([-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], -10.74908009,  28, 0.)
```

Figure 17: Representative selection of the output states with highest number of occurrences from the SampleSet.record data type provided by D-Wave Leap for the circuit seen in FIG. 5. The first column represents the cell polarizations, the second column the energy value, and the third column is the number of occurrences.

We see that there are significant differences between the polarization states produced by the QPU and the metastable states seen in QCA Designer just from a quick glance at FIG 16 and FIG. 17. To effectively compare these outputs further we realize that the ordering of states is top to bottom and left to right so that we can compare values cell by cell. We also generate percentage values for the proportion of outputs that are -1 or 1 at each cell and these results can be found within the `xor_analysis.ipynb` file. This result is also captured in FIG. 21. This allows us to perform a more direct comparison. The overall proportion metric shows us that the states are not consistent for higher energy levels and there seems to be a near even distribution of 1/-1 polarization's for most cells. It is interesting to note that for the cells right beside the inputs as well as the output cells, we see high probabilities across all states. Looking at the proportion of ground state polarization's we see that the values do not really match the metastable states seen in FIG. 16. While there is some overlap it is only for states close to the input or output cells



```
overall proportion of 1s per cell [0.194, 0.323, 0.516, 0.871, 0.774, 0.581, 0.452, 0.452, 0.548, 0.516, 0.548, 0.548]
overall proportion of -1s per cell [0.806, 0.677, 0.484, 0.129, 0.226, 0.419, 0.548, 0.548, 0.452, 0.484, 0.452, 0.452]
proportion of 1s per cell in ground energy state [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]
proportion of -1s per cell in ground energy state [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
```

Figure 18: proportion of 1 and -1 polarizations across all energy states as well as just for the ground state.

Now we move onto the `XOR` circuit. The same figures are captured for this circuit.

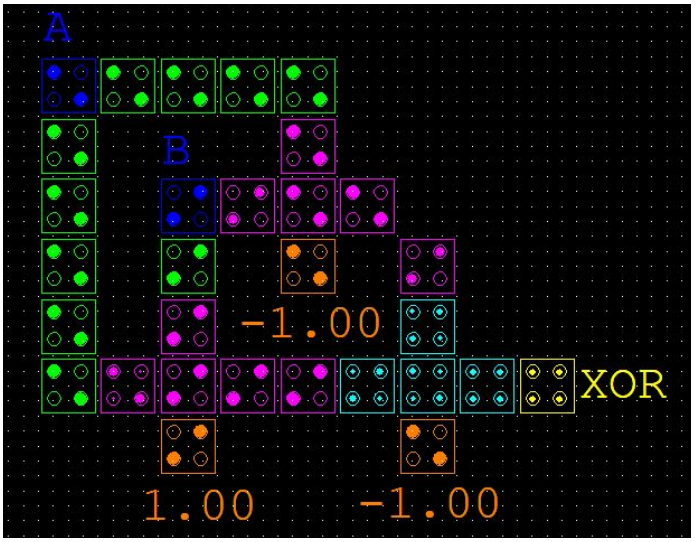
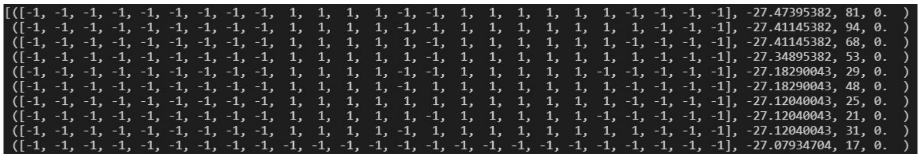


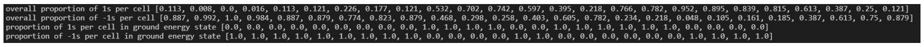
Figure 19: Mid-simulation screenshot of the circuit seen in FIG. 1. Specifically, we illustrate the polarization values mid-execution for the input state $A = -1$, $B = 1$

Focussing on just the first and second clocking regimes present in the `XOR`



```
[([-1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1,  1, -1, -1,  1,  1,  1,  1,  1,  1, -1, -1, -1, -1], -27.47395382, 81, 0.  )
 ([-1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1,  1, -1, -1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1], -27.41145382, 94, 0.  )
 ([-1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1,  1,  1, -1,  1,  1,  1,  1,  1,  1, -1, -1, -1, -1], -27.41145382, 68, 0.  )
 ([-1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1,  1,  1, -1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1], -27.34895382, 53, 0.  )
 ([-1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1,  1, -1, -1,  1,  1,  1,  1,  1, -1, -1, -1, -1, -1], -27.18290043, 29, 0.  )
 ([-1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1,  1, -1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1, -1], -27.18290043, 48, 0.  )
 ([-1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1, -1], -27.12040043, 25, 0.  )
 ([-1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1,  1,  1, -1,  1,  1,  1,  1,  1, -1, -1, -1, -1, -1], -27.12040043, 21, 0.  )
 ([-1, -1, -1, -1, -1, -1, -1, -1, -1,  1,  1,  1,  1, -1,  1,  1,  1,  1,  1,  1,  1,  1, -1, -1, -1], -27.12040043, 31, 0.  )
 ([-1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1], -27.07934704, 17, 0.  )
```

Figure 20: Representative selection of the output states with highest number of occurrences from the SampleSet.record data type provided by D-Wave Leap for the circuit seen in 1. The first column represe nts the cell polarizations, the second column the energy value, and the third column is the number of occurrences.



```
overall proportion of 1s per cell [0.113, 0.008, 0.0, 0.016, 0.113, 0.121, 0.226, 0.177, 0.121, 0.532, 0.702, 0.742, 0.597, 0.395, 0.218, 0.766, 0.782, 0.952, 0.895, 0.839, 0.815, 0.613, 0.387, 0.25, 0.121]
overall proportion of -1s per cell [0.887, 0.992, 1.0, 0.984, 0.887, 0.879, 0.774, 0.823, 0.879, 0.468, 0.298, 0.258, 0.403, 0.605, 0.782, 0.234, 0.218, 0.048, 0.105, 0.161, 0.185, 0.387, 0.613, 0.75, 0.879]
proportion of 1s per cell in ground energy state [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0]
proportion of -1s per cell in ground energy state [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0]
```

Figure 21: proportion of 1 and -1 polarizations across all energy states as well as just for the ground state.

circuit, we see that many of the same observations made for a single clocking regime can be applied to two. We see that the -1 polarization present at the top and left of the circuit are well represented in the QPU output. However, once the second clocking regime comes into play (pink cells in FIG. 19) the polarizations do not match at all. In fact, the disparity in the polarizations of the QCA Designer simulation and the QPU output seem entirely random once a second clocking regime is introduced.

5 Conclusion and Recommendations

In this report, we have designed and simulated various circuits on QCA Designer as well as run them on a quantum annealing processor. We have developed a streamlined and generalized approach to taking QCA Designer files and running the circuit on a QPU. We have also conducted extensive testing into the behaviour of coplanar crossings with varying widths. The XOR gate was another main focus of the report in that circuit designs were developed and run on the QPU. The subsequent output was processed and the behaviour of clocking regimes on this output was characterized. It was seen that circuit designs that functioned correctly without clocking regimes performed better on the QPU than ones that required clocking regimes.

With these conclusions, we can go further and provide a recommendation for the factors to minimize in order to best simulate large clock-zone QCA networks. Larger clock zones present the introduction of more possible energy eigenstates. This means that more of these states have the ability to be below the thermal energy threshold and thus presents a viable transition of the system from the ground state. We have seen the effect of k_bT on a system in the coplanar crossover analysis. Maintaining this value as low as possible by ensuring low operating temperature can have a large effect in increasing overall accuracy.

This thermal energy can be influenced by many variables but having many large concurrent processes on a single QPU has been shown to increase this energy dramatically [3]. Another source of error is hardware imperfections present in a processor. Minimizing this value is important through the use of better manufacturing processes. Finally, limiting the number of complex interactions is seen to be highly effective as well. Looking back at the 10 cell wire that was simulated with 100% expected output compared to the 50-70% correct output of the `XOR_MAIN` circuit, we see that while the number of cells in the system may be similar, the simple interactions in a wire are easily simulated correctly while the QPU struggles to some degree with the geometry of `XOR_MAIN`.

References

[1] File:3 gate XOR.svg - Wikimedia Commons, Dec 2020. [Online; accessed 17. Dec. 2020].

[2] Amir Chabi, Arman Roohi, Hossein Khademolhosseini, Shaahin Angizi, Ronald DeMara, and Keivan Navi. Cost-efficient QCA reversible combinational circuits based on a new reversible gate. *ResearchGate*, Oct 2015.

[3] Shuntaro Okada, Masayuki Ohzeki, Masayoshi Terabe, and Shinichiro Taguchi. Improving solutions by embedding larger subproblems in a D-Wave quantum annealer. *Sci. Rep.*, 9(2098):1–10, Feb 2019.