

## 1. INTRODUCTION

Computers, which have become a ubiquitous staple of modern society, are projected to consume nearly 20% of all the energy produced worldwide by the year 2040 [1]. While computing implementation technology has been made more energy efficient over the years, the energy required to operate a gate logically has become an increasingly large proportion of the total energy required. A systematic improvement of this power use would result in significant power savings, which would grow even more appreciable as overall efficiency improves. One method for achieving these power savings would be the use of conservative reversible logic (CRL) gates for system design. However, to date, only a few designs using these types of gates have been developed. The lack of an efficient analysis method for CRL gates and circuits is partially responsible for this sporadic development.

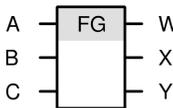
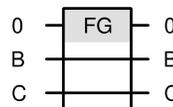
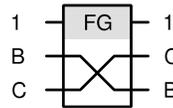
## 2. OBJECTIVES

- Introduce effective design method for CRL gates and circuits
- Demonstrate improved efficiency over traditional methods of analysis

## 3. BACKGROUND

A CRL gate, in order to remain conservative, must have the same number of outputs as inputs. The simplest CRL gate is known as the Fredkin Gate (FG) and has three outputs/inputs. Fig. 1 shows the FG truth table and its circuit element and two operational states.

A	B	C	X	Y	Z
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

**Fig. 1** Truth table, (a) circuit element, and (b) operational states of the FG

While the truth table accurately denotes the operation of any CRL gate, as the number of inputs/outputs increases linearly, the number of elements in the resulting truth table increases exponentially.

To reduce the total number of elements in the truth table of a given CRL gate, the truth table can be reduced to a minimal form based on the permutations created by the given CRL gate. For example, the FG creates the permutations

$$\{(A, B, C), (A, C, B)\}$$

These two permutations can then be encoded in a reduced tabular form based on the control line encodings of the FG.

Tab. 1 shows this reduced form.

A	X	Y	Z
0	A	B	C
1	A	C	B

**Tab. 1** Reduced truth table for a FG

This reduced form is known as a permutation table, and is capable of representing CRL gates of any size. In a permutation table, the columns on the input side represent all of the possible combinations of the control lines of a given CRL gate. In the permutation table shown in Tab. 1, there is only one input column, the leftmost column, which denotes the values '0' and '1', as there is only a single control line for a FG.

The columns on the output side denote the outputs of a given CRL gate. There will only be as many output columns as there are outputs of a CRL gate. For each control line encoding, the corresponding permutation created is written in the corresponding elements of each output column.

The resulting reduced tabular form is a much more compact way of representing CRL gates that keeps the advantages of the truth table but eliminates the way a truth table grows exponentially as the inputs of a CRL gate increase.

## 4. METHODS

Now that any CRL gate can be represented using a minimal tabular representation, CRL gates and circuits can be analyzed.

### A. Analysis of a Fredkin Gate

In order to analyze a CRL gate, each output column of the corresponding permutation table must be mapped to a permutation K-Map (PK-Map). A PK-Map is a compact method of a K-Map that is similar to a Variable Entered K-Map (VEKM) [2].

Each entry in each output column should be mapped to the corresponding entries in a PK-Map whose size corresponds directly to the number of control lines in a given CRL gate. After mapping each entry, values should be grouped in a similar way to regular K-Maps. Fig. 2 shows how the first output column, the 'X' column, should be mapped and grouped.

A	0	1
A	A	A

**Fig. 2** Covered PK-Map of 'X' output of a FG

This single group then produces the output expression

$$f_x = A$$

The second output column maps to the PK-Map shown in Fig. 3.

A	0	1
B	C	

**Fig. 3** Covered PK-Map of 'Y' output of a FG

The second PK-Map shown in Fig. 3 contains two groups and produces the output expression

$$f_y = \bar{A}B + AC$$

The third output column, 'Z', maps to the PK-Map shown in Fig. 4.

A	0	1
C	B	

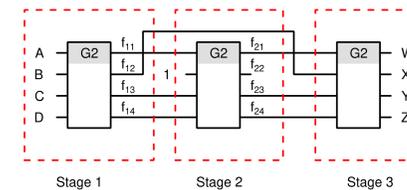
**Fig. 4** Covered PK-Map of 'Z' output of a FG

The third PK-Map shown in Fig. 4 also contains two groups and produces the output expression

$$f_z = \bar{A}C + AB$$

### B. Analysis of a Multistage CRL Gate Circuit

In addition to single CRL gates, PK-Maps are also suitable for analyzing multistage CRL gate circuits. In order to more easily analyze a multistage CRL circuit, the CRL circuit being analyzed should be divided into several stages. Fig. 5 shows how a three-stage circuit comprised of G2 gates should be divided into three stages.



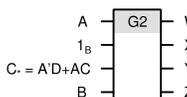
**Fig. 5** Stages of a three-stage CRL gate circuit

After dividing a circuit into stages, each stage can be analyzed as an individual CRL gate, starting with the first stage. The expressions of the first stage become

$$\begin{aligned} f_{11} &= A \\ f_{12} &= \bar{A}C + AD \\ f_{13} &= \bar{A}D + AC \\ f_{14} &= B \end{aligned}$$

Using the expressions determined from the first stage, the second stage can be analyzed. To analyze the second stage, the permutation table should be modified. Fig. 6 shows stage two and its modified permutation table.

A	W	X	Y	Z
0	A	C.	D	1 <sub>B</sub>
1	A	D	C.	1 <sub>B</sub>



**Fig. 6** Modified permutation table and depiction of stage two

With the modified permutation table created, each output column can be mapped to a PK-Map. The first output column, 'W', yields the following expression

$$f_{21} = A$$

For columns that contain the same variable, the output expression will simply be the variable at the input of the control line, which in this case is the Boolean variable A. The second output is not connected, and so its analysis can be skipped. The fourth output column, 'Z', yields the following expression

$$f_{24} = 1$$

because each entry is a constant Boolean '1'. The third output column, 'Y', requires a PK-Map. Fig. 7 shows this completed PK-Map.

A	0	1
D	C.	

**Fig. 7** Covered PK-Map of 'Y' output of the stage in Fig. 6

Using the PK-Map shown in Fig. 7 yields the expression

$$f_{23} = \bar{A}B + AC$$

These expressions can then be used to compute the output expressions of the third stage, which are the final outputs.

## 5. RESULTS

It is also clear that PK-Map method is more efficient than traditional methods which rely on the full truth table of a CRL gate in order to analyze it.

To quantify this improved efficiency, the efficiency of a PK-Map will be compared with the efficiency of a the Quine-McCluskey algorithm for the same generalized CRL gate. A CRL gate with  $n$  inputs/outputs and  $c$  control lines will ultimately require  $n$  K-Maps to be analyzed with a maximum of  $\frac{3^n}{\sqrt{n}}$  prime implicants [3]. Therefore the computational complexity of a traditional CRL analysis method can be estimated to be

$$n \left( \frac{3^n}{\sqrt{n}} \right)$$

Using a PK-Map, however, is much more efficiency. The permutation table effectively reduces all of a CRL gate's outputs to a function of only the CRL gate's control lines. Therefore, the computational complexity of the PK-Map can be estimated to be

$$n \left( \frac{3^c}{\sqrt{c}} \right)$$

and because  $c$  will always be smaller than  $n$ , the computational complexity of a PK-Map will always be less than the computational complexity of traditional methods of CRL gate analysis. Tab. 2 demonstrates how the complexity of a PK-Map is always vastly smaller than the complexity of a traditional method.

n	c	Quine-McCluskey	PK-Map	Complexity Decrease (%)
3	1	47	9	80.25
4	1	162	12	92.59
5	1	543	15	97.24
6	1	1786	32	98.99

**Tab. 2** Computational complexities of common CRL gates

## 6. CONCLUSION

By introducing a permutation table and a PK-Map, CRL gates and circuits can be more efficiently represented and analyzed. Using these contributions, optimal CRL circuits can begin to be more easily studied which will lead to their patterns being better understood, which will ultimately lead to a better design method for arbitrary circuits comprised of CRL gates.

## 7. REFERENCES

- [1] Andrae, A., & Edler, T. (2015). On global electricity usage of communication technology: Trends to 2030. *Challenges*, 6 (1), 117–157. <https://doi.org/10.3390/challe6010117>
- [2] Burgoon, R. (1972). Improve your karnaugh mapping skill. *Electron. Des.*, 21 (Dec), 54–56.
- [3] Chandra, A. K., & Markowsky, G. (1978). On the number of prime implicants. *Discrete Mathematics*, 24 (1), 7–11. [https://doi.org/10.1016/0012-365x\(78\)90168-1](https://doi.org/10.1016/0012-365x(78)90168-1)