The Journal of Supercomputing (2022) 78:8056–8080
https://doi.org/10.1007/s11227-021-04247-9

Design and implementation of efficient QCA full-adders using fault-tolerant majority gates

J. A. Bravo-Montes[1] · A. Martín-Toledano[1] · A. Sánchez-Macián[1] · O. Ruano[1] · F. Garcia-Herrero[1]

Accepted: 19 October 2021 / Published online: 7 January 2022

Abstract

CMOS technology is facing physical limitations in scaling the manufacturing process. Therefore, to deepen the development of better designs in a smaller area, it is necessary to look for other alternatives. One of the most studied approaches is Quantum Cellular Automata (QCA). However, it has the disadvantage of its reliability during the manufacturing processes, with high error rates that are difficult to improve. To contribute to the design of more reliable operators based on this technology, new fault-tolerant full-adders are presented in this paper. The proposed solutions improve area up to 57.14%, total energy dissipation up to 36.27%, and average energy dissipation per cycle up to 36.22% compared to those previously proposed. This reduction in power consumption is especially important to make QCA more competitive as it has to operate in low-temperature environments.

Keywords Design · Fault-tolerant · QCADesigner · Quantum cellular automata

✉ F. Garcia-Herrero
fragarh2@gmail.com

J. A. Bravo-Montes
jbravom1@alumnos.nebrija.es

A. Martín-Toledano
amartintoledanog@alumnos.nebrija.es

A. Sánchez-Macián
asanchep@nebrija.es

O. Ruano
oruano@nebrija.es

[1] ARIES Research Center, Universidad Antonio de Nebrija, Madrid 28049, Spain

1 Introduction

Moore's law describes an exponential growth in the number of transistors per area unit over time. However, current Complementary Metal–Oxide–Semiconductor (CMOS) technology is encountering physical limitations such as lithography, making it difficult to reduce the area and, therefore, continue this law [1]. At this point, it is essential to find reliable alternatives for the construction of larger circuits in a smaller area that could replace or even improve CMOS manufacturing processes. Currently, there are several technologies whose objective is to become an alternative.

One possible solution is Nanomagnetic Logic (NML), based on the use of nanomagnets for the construction of circuits, whose main advantage is their high-integration density and their ability to operate at room temperature [2, 3].

Another solution is Silicon Dangling Bonds (Si-DB) based upon quantum dots that allow the creation of ultra-low energy schemes [4, 5].

Finally, Quantum Cellular Automata (QCA), the technology that we will focus on in this work, is an efficient process studied for almost 30 years as a possible foundation for a future generation of integrated circuits. It was proposed by Lent *et al.* in 1993 [6] and developed in 1997 [7], and it has grown to become a relevant alternative technology [8] due to the maturity of its development kits such as QCADesigner [9], and QCADesigner-E [10].

1.1 Related works

Due to the QCA reliability disadvantage during the manufacturing process, a fault-tolerant full-adder using QCA technology will be introduced in this work. The full-adder is a fundamental piece for digital processing and, therefore, for elaborating more complex circuits. Some examples of this can be seen in the creation of a microprocessor, whose Arithmetic Logic Unit (ALU) requires an adder [11], or in other components such as a Digital Signal Processing (DSP) [12], which contains an adder among its basic operations. Proof of this is the numerous works where adders are used to develop more complex circuits [13, 14]. Regarding the creation of full-adders in QCA, other approaches have been made prioritizing speed by creating a three-layer full-adder [15] or sacrificing the circuit's total area in search of fault tolerance, with the use of five-input majority gates [16].

One of the most used full-adder architectures in the literature is [17], which was recently improved by [18] reducing the number of cells, area, and energy dissipation. However, this design is not fault-tolerant.

The current work shows a full-adder with different protection levels partially based on the unprotected architecture from [18] compared to the fully protected one from [19], trying, at the same time, to limit and reduce energy overheads to improve cooling constraints. The main contribution of this work is the generation of a set of full-adders with different layouts and levels of protections depending on the (normal or fault-tolerant) configuration of their majority-logic gates, with a good trade-off in

Table 1 Comparison to existing works

Work	Fault-tolerant adder	Area-Power figures	Delay	Fault-tolerant config.
[15]	−	++	+	−
[16]	+	+	+	−
[17]	−	+++	+	−
[18]	−	++++	+	−
[19]	+	+	+	−
This work	+	++	+	++

terms of area, delay and power consumption when comparing to previous solutions. To the authors' knowledge, no previous work proposes these different options that can be used in different applications depending on the level of reliability required by the system, and the area and power restrictions. An example of application is approximate adders [20], used in error-tolerant applications where some accuracy can be sacrificed to provide better circuit-based metrics.

Table 1 shows a comparison with a set of previous works that implement full-adders. Considering those that provide fault tolerance, the proposed adder delivers better circuit-based metrics. Additionally, a set of partial fault tolerance configurations is presented, optimizing the area and layout of each one based on the required fault-tolerant majority gates, where "+" indicates higher optimization and "-" less optimized.

The paper's organization is the following: Sect. 2 describes the different elements that make up the QCA technology, progressing from a basic circuit to a fault-tolerant full-adder. Section 3 shows the proposed solution for the fault-tolerant full-adder explaining its architecture and the differences concerning the reference model [19]. Section 4 shows the results obtained and finally, Sect. 5 concludes the paper.

2 Background

To understand QCA technology, it is necessary to know the basic components: cells, wires, inverters, majority gates, logic gates, clocks, full-adders, and fault-tolerant design. These are detailed in the following subsections.

2.1 Cells

The main unit of this technology is the cell. Each one is made up of four quantum dots (semiconductor nanostructure that confines the movement of electrons) and two electrons that can be positioned on the two diagonals; thanks to the opening of the existing potential barrier and the Coulomb repulsion between the electrons [21].

In this way, the cell is in the $+1$ or -1 polarization state, representing a logical *1* or *0*, respectively, as shown in Fig. 1.

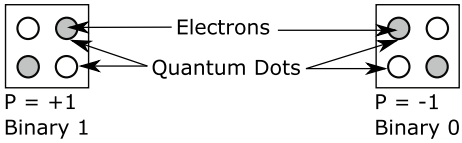


Fig. 1 QCA Cells [22]

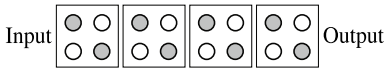


Fig. 2 Simple Wire [23]

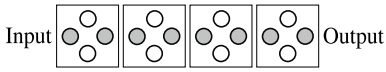


Fig. 3 45^o Wire [23]

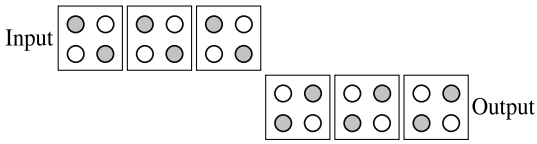


Fig. 4 Inverter [23]

2.2 Wire

Thanks to the Coulomb repulsion, the proximity between cells causes them to be affected by the other cells' state. In this way, a cell could be set as input waiting for a value, and that state could be transmitted to other areas of the circuit. This can be done in two different ways due to the cells' placement, showing two different behaviors depending on whether the cells are positioned normally or rotated 45^o.

This can be seen in Fig. 2, which shows a simple wire that would behave like CMOS connections. Also, Fig. 3 shows how the quantum dots are rotated 45^o, which causes the data transmission to vary. Due to Coulomb repulsion, the value would change from *1* to *0* as it progresses through the wire. These changes could result in the effect that, if the wire is composed of an input cell, an output cell, and an intermediate n cells, if n is odd, the input and output will be the same, while if it is even, the result will be the opposite of the input.

2.3 Inverter

Due to these characteristics, an inverter can be made very easily. Figure 4 shows the simplest example to implement.

It has an input where the value to be inverted will be inserted and an output where the operation's result will be checked. However, instead of being a straight wire, the last three cells are set down in one position. When reaching the point

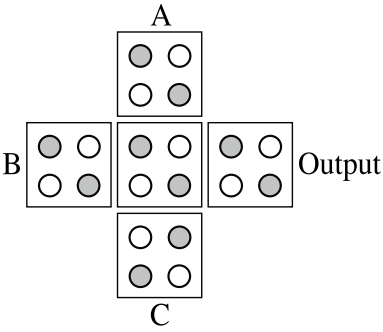


Fig. 5 Majority Gate [18]

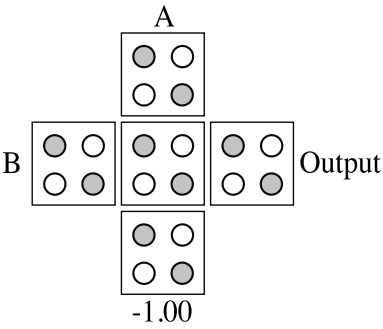


Fig. 6 Logic gate AND [24]

where the third cell value must pass to the fourth, it changes the polarization to the opposite, obtaining the expected result in the output cell.

2.4 Majority gate

The majority gate is one of the main tools for creating logic circuits [18]. Knowing the properties of this element, it can be modified to achieve the basic logic gates, i.e., AND, OR, etc. Given three inputs, these propagate to the central cell, which will take the most repeated polarization and be transmitted to the output.

The example in Fig. 5 shows a majority gate with three inputs and one output, in which the desired inputs (A, B, C) will be set, and the expected value will be observed at the output. Following the majority gate function:

$$M(A, B, C) = AB + AC + BC.$$

2.5 Logic gates

Majority gate's structure can be manipulated to create the AND and OR logic gates [24].

Figure 6 shows how the AND logic gate is formed. To achieve this behavior, one of the three entries of the majority gate would have to be set to -1. Due to

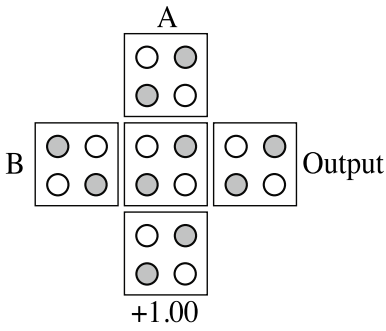


Fig. 7 Logic gate OR [24]

this, we can alter the function described by the majority gate since the input previously called C would be replaced by -1. Applying Boolean logic, this logic gate would be described by the following function $M(A, B, \text{-}1) = AB$. That is the same as the AND logic gate. Similarly, in Fig. 7, the third input is set to $+1$ to achieve an OR logic gate's behavior. Resulting in the following equation: $M(A, B, 1) = A + B$.

2.6 QCA clock

QCA has a multiphase clock [21]. This allows multi-stage synchronization to be performed. It has four clocks, each one of them shifted in phase by 90^o to the previous one, giving the possibility of crossing wires in opposing clocks without influencing each other.

In this way, if two wires are to be crossed, they are in zones 0 and 2 or 1 and 3, it can be done without the values transmitted by one wire affecting those of the other wire [25]. Due to this, it is possible to create larger circuits in a smaller area.

As can be seen in Figs. 8 and 9, there are two opposing phases, while one is in the "release phase," the other one is in the "switch phase," and while the first one is in the relax phase, the other one is in the hold phase, and *vice versa*. A clock cycle is completed when a clock zone passes through four different phases.

2.7 Full-adder

As can be seen in Fig. 10, the QCADesigner tool has been used for the implementation of a full-adder from [17].

The full-adder has A and B which are the two one-bit inputs and *Cin* is the carry input. On the other hand, *Sum* and *Cout* correspond to the two outputs that provide the result of the addition and the carry, respectively. This circuit consists of 44 QCA cells. The circuit is delimited by clock zones that are reflected in the different colors of the cells that make up the circuit.

For this full-adder, four clock zones have been used following a grayscale that represents the clock zones from 0 to 3 as shown in Fig. 11.

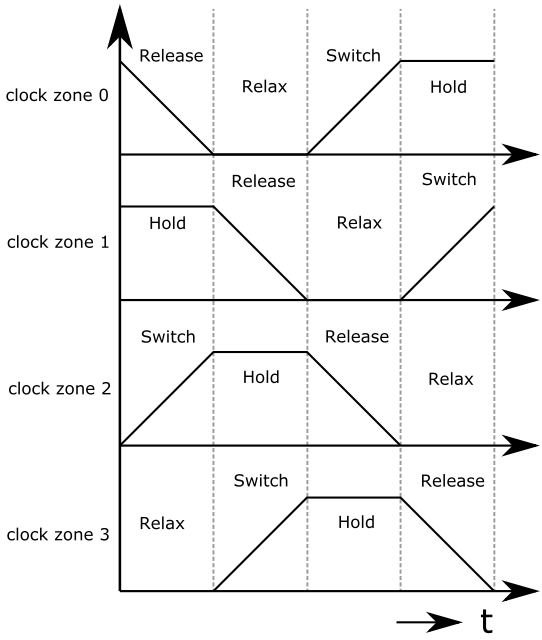


Fig. 8 Clock zones [21]

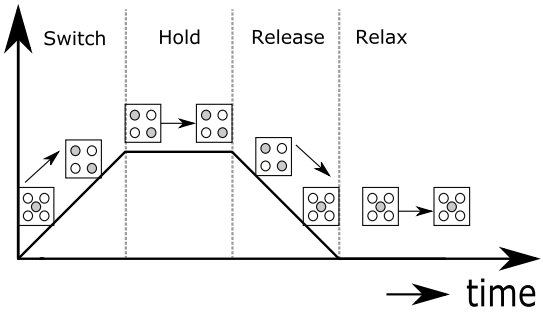


Fig. 9 Clock phases [21]

2.8 Fault-tolerant design QCA

The majority gates are the critical points where a small error can produce erratic behavior of the circuit as demonstrated in [19]. For this reason, it has been proposed to replace all simple majority gates with others with a larger area but greater reliability, exposed in the same reference document.

The structure implemented in the majority gate is equivalent to a traditional one detailed in Sect. 2. For the fault tolerance analysis, different errors were taken into account in [19], such as single-cell omission, extra-cell deposition, and cell misalignment were investigated using QCADesigner 2.0.3. The results obtained were 100% tolerant against a single-cell omission defect and 90% fault-tolerant against an extra-cell deposition defect [19].

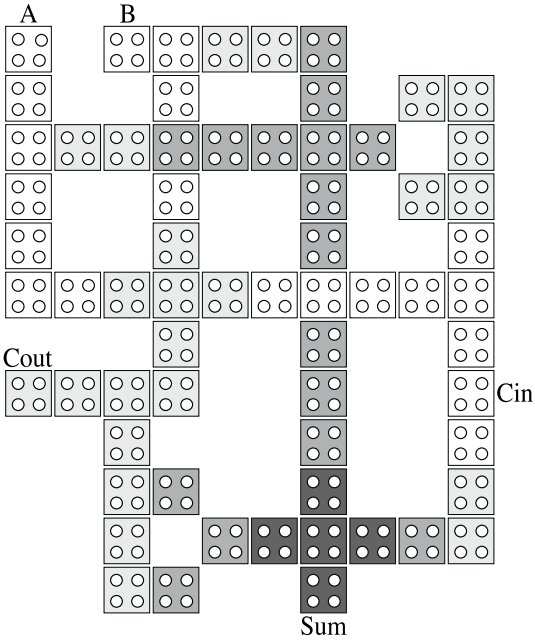


Fig. 10 Full-Adder from [17]

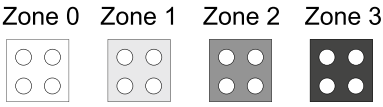


Fig. 11 Clock zones key

This fault-tolerant majority gate has a total of 10 cells (seven normal cells and three rotated cells) and overcomes the error caused by cell omission, extra-cell deposition, and cell displacement defects [19].

In particular, the proposal improved above the previous ones in single-cell omission, double-cell omission, triple-cell omission, quadruple-cell omission, extra-cell, and deposition permissible cell displacement defects.

This structure has been chosen as a reference for this paper after evaluating different majority gates [26–33] since it is optimal in terms of the number of cells, area, latency, cell displacement defects, and energy consumption.

Table 2[1] shows a comparison between a majority gate [18] and a fault-tolerant one such as the one in Fig. 12. As can be seen, the greater the number of cells used, the higher the area and energy dissipation are.

[1] To obtain the results, QCADesigner has been used. The QCADesigner-E (QD-E) is an extension of the QCADesigner. It calculates the estimation of the power dissipation of QCA circuits. It is integrated as an additional simulation module that is based on the Coherence Vector Simulation (CVSE) [9, 10]

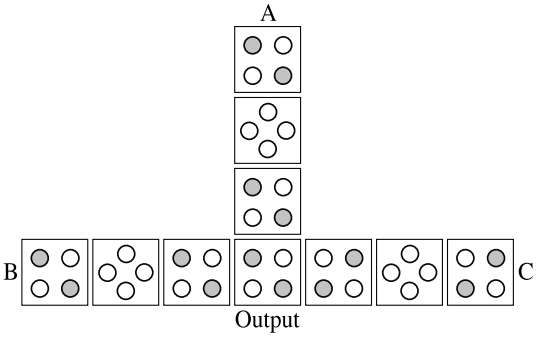


Fig. 12 Fault-Tolerant Majority Gate [19]

Table 2 Majority Gate Comparison

	Majority Gate from [18]	Fault-Tolerant Majority Gate from [19]
Num Cells	5	10
Area (μm^2)	0.01	0.02
Total Energy Dissipation (eV)	2.59^{-3}	6.82^{-04}
Average Energy Dissipation per Cycle (eV)	2.36^{-04}	6.20^{-05}

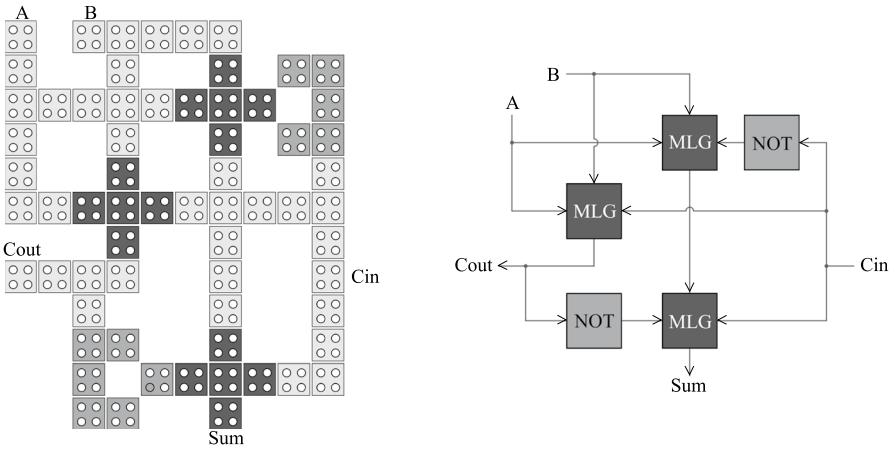


Fig. 13 QCA design (left) vs architectural diagram (right)

3 Proposed solution

This work introduces a new fault-tolerant full-adder architecture. However, the solutions provided in this section can be applied to any full-adder with a different degree of protection against errors, obtaining improvements in area and power consumption without raising latency compared to the state of the art. In Fig. 13, we include the

architectural diagram of the proposal. As it can be seen, the full-adder is built of inverters (NOT) and majority gates (MLG). As we explained, majority gates are the main source of error, for this reason in this work, we will exchange the MLG blocks by its fault-tolerant implementations and we will provide customized solutions for different combinations of protected and unprotected MLG blocks, providing designers with a wide and optimized catalog of adders with different levels of reliability, area consumption, energy dissipation and latency, depending on their needs.

In the following subsections, the parameters that have been taken into account for the proposed full-adder architecture design will be exposed. Later, the general and specific differences between both the existing architectures and the full-adders described here will be presented.

3.1 Architecture design criteria

For the proposed full-adders, parameters such as area, energy dissipation, latency, and connection with external circuits have been considered. Next, a brief explanation of each of these factors and how they directly affect the proposed full-adder architecture is included.

3.1.1 Area

To optimize the area, it has been necessary to reduce the number of cells used in the circuit, reducing the wires' length. On the other hand, the crossover of wires implemented within the circuit itself allows us to avoid using a large number of cells by having to surround the entire circuit, especially when it is necessary to use the same input of the full-adder proposed in different majority gates. With this idea in mind, it has been possible to considerably reduce the number of cells used in the circuit and thus reduce the proposed full-adder's total area.

3.1.2 Energy dissipation

This parameter directly depends on the total area of the circuit. For this reason, by reducing the size of the proposed full-adder, energy dissipation is improved as well. However, it is also important to take into account the number of clocks implemented in the circuit. Due to this, the transitions in a wire have been optimized using different clocks, thus achieving a direct reduction in energy dissipation.

3.1.3 Latency

Another critical factor that has been taken into account is latency. One of the main challenges in the design of the proposed full-adder is to maintain the same latency as the original one. This has been achieved thanks to the wire crossover within the proposed full-adder, obtaining a circuit of equal latency and area reduction.

3.1.4 Input and output connection to external circuits

Finally, it has been taken into account that the inputs and outputs must be easy to connect to other independent circuits. To achieve this, it was necessary to consider that both the three inputs and the two outputs must be located in the borders, thus providing the possibility of connection with the pinout of different circuits.

3.2 Reference architectures

The reference architectures used in this paper are the unprotected full-adder of [18] and the fault-tolerant full-adder of [19] in Fig. 16.

We define three different zones related to the three majority gates involved in the design. To provide flexibility with the level of protection against errors, we analyze all the possible comparisons using majority gates or fault-tolerant majority gates.

The proposed full-adder offers a flexible implementation that allows configuring the degree of protection to apply to a circuit. As it is demonstrated in [19], the majority gates are the critical points and main source of errors in QCA designs, where a small error can produce erratic behavior of the circuit. For this reason, increasing the level of fault-tolerant majority gates increases the reliability of the whole logic unit, at a cost of a larger area, compared to unprotected majority gates. This proposal is intended for architectures that do not need total protection but where partial protection is enough to satisfy their goals, as there are other secondary sources of error apart from the majority gates. An example is the approximate adders that are architectures implemented for image processing [20].

To name the different architectures, a three-letter code is used, one letter for each zone. If the majority gate of a specific zone is not protected is represented with *U* (Unprotected) in the corresponding letter and with *P* (Protected) if it is fault-tolerant. Hence, the unprotected full-adder will be represented by the code *UUU*, and *PPP* is the code for the fully protected circuit.

To compare the rest of the semi-protected architectures, despite not being found in the original article [19], the circuits that would correspond to the literature (UUP in Fig. 18, UPU in Fig. 20, UPP in Fig. 14, PUU in Fig. 22, PUP in Fig. 24, PPU in Fig. 26) have been generated.

3.3 Proposed architectures

The architectures proposed to compare with work [19] and its semi-protected representations are UUP in Fig. 19, UPU in Fig. 21, UPP in Fig. 15, PUU in Fig. 23, PUP in Fig. 25, PPU in Fig. 27, and PPP in Fig. 17.

The development of the UUP, UPU, and UPP designs has used the UUU circuit of [18] as a reference. However, the rest of the circuits (PUU, PUP, PPU, PPP)

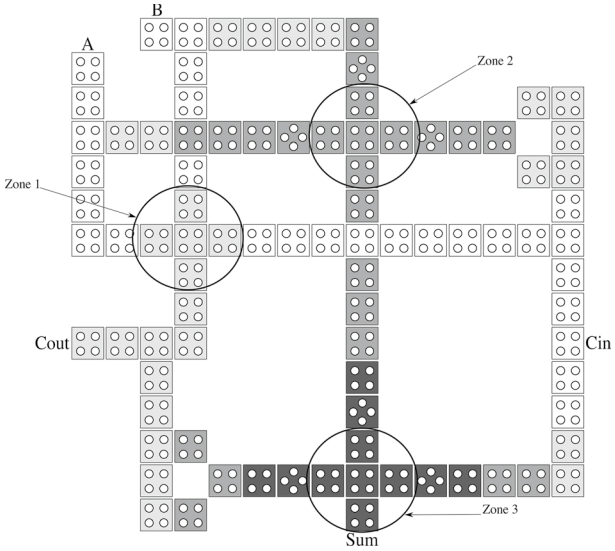


Fig. 14 Full-Adder UPP from [19]

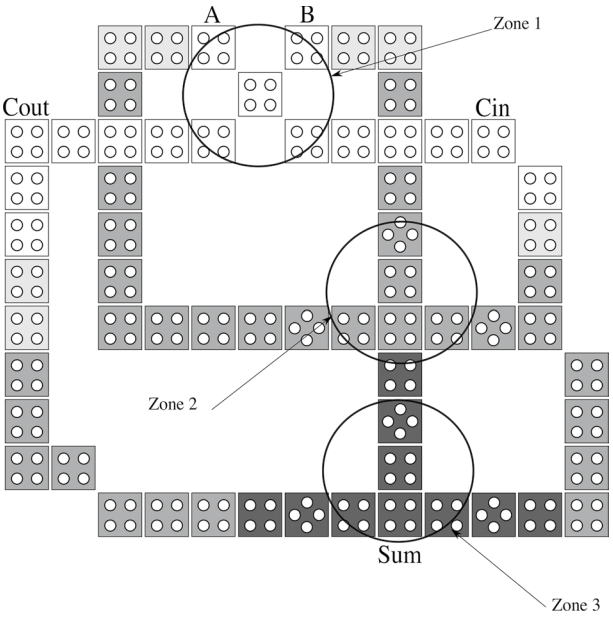


Fig. 15 Proposed Full-Adder UPP

have been created specifically for this paper to provide better results in terms of overhead reduction.

Models are compared below to show the differences between the circuit from [19] and the proposed architecture.

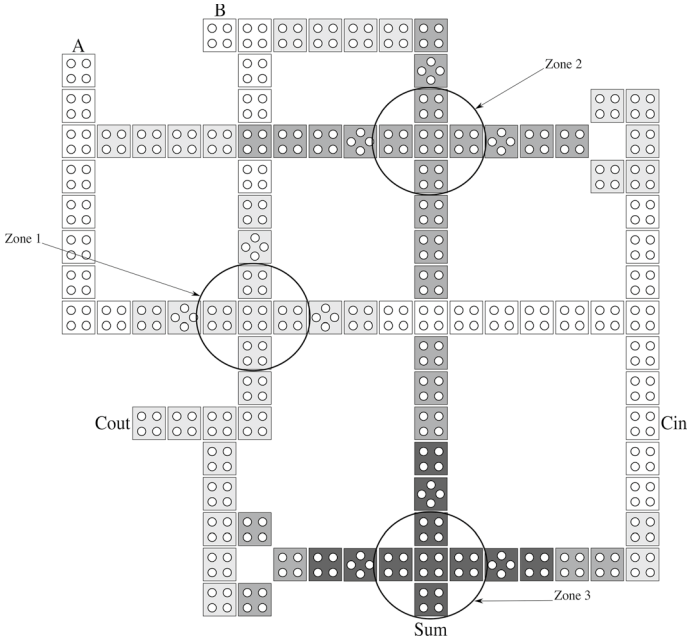


Fig. 16 Full-Adder *PPP* from [19]

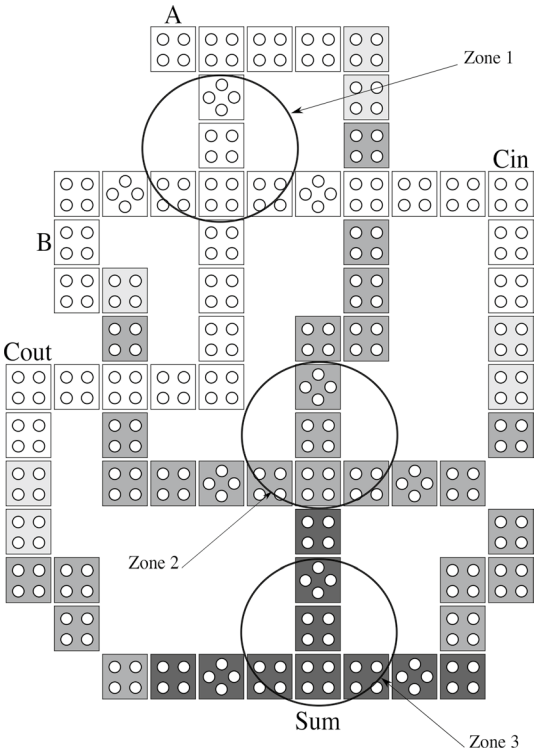


Fig. 17 Proposed Full-Adder *PPP*

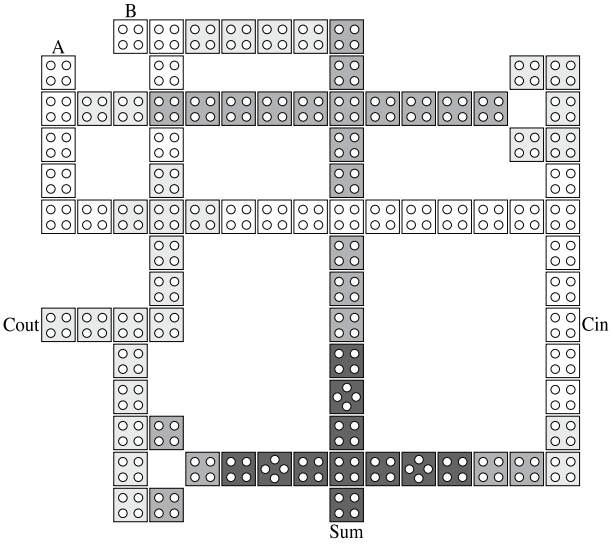
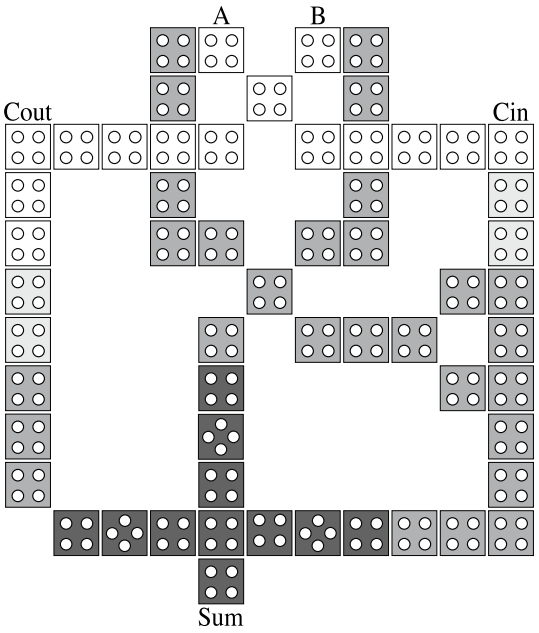


Fig. 18 Full-Adder *UUP* from [19]

Fig. 19 Proposed Full-Adder *UUP*



A particular focus has been set on UPP and PPP architectures due to UPP being the circuit in which the highest percentage of energy dissipation improvement has been achieved and PPP architecture being the design most protected against failures.

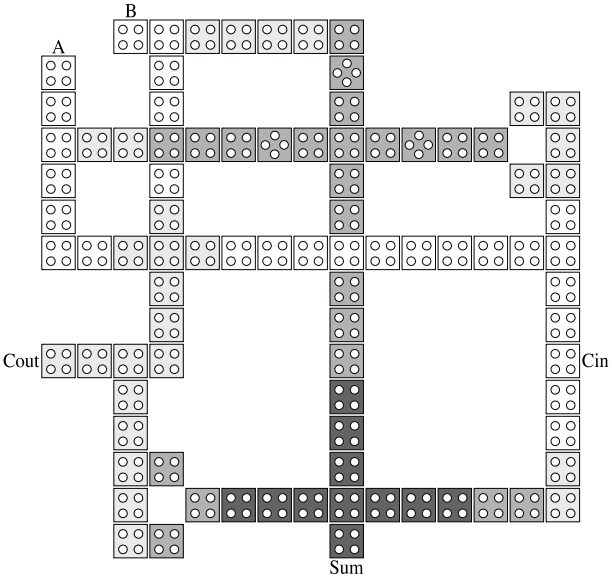


Fig. 20 Full-Adder *UPU* from [19]

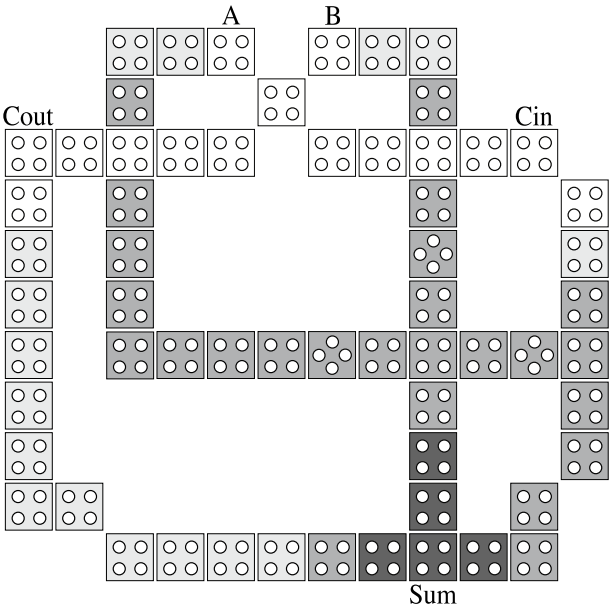


Fig. 21 Proposed Full-Adder *UPU*

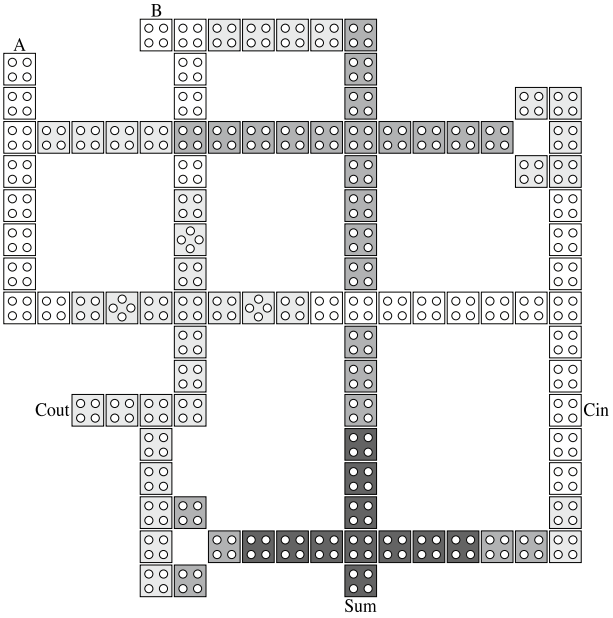


Fig. 22 Full-Adder *PUU* from [19]

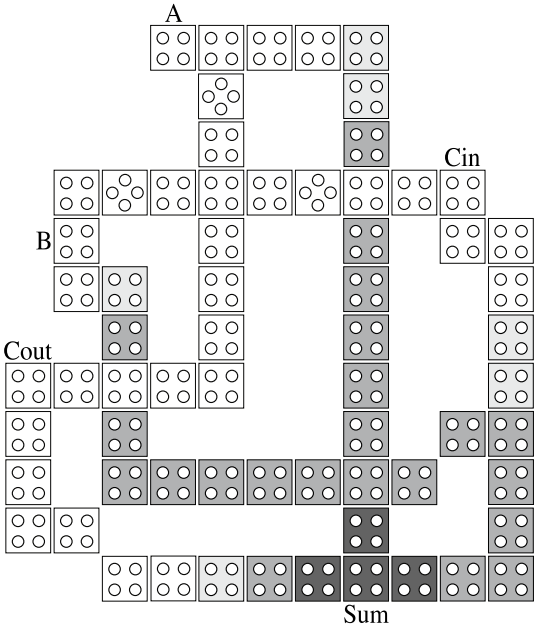


Fig. 23 Proposed Full-Adder *PUU*

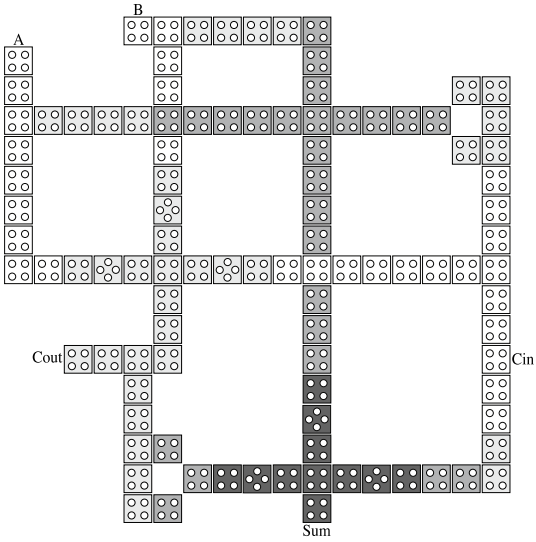


Fig. 24 Full-Adder *PUP* from [19]

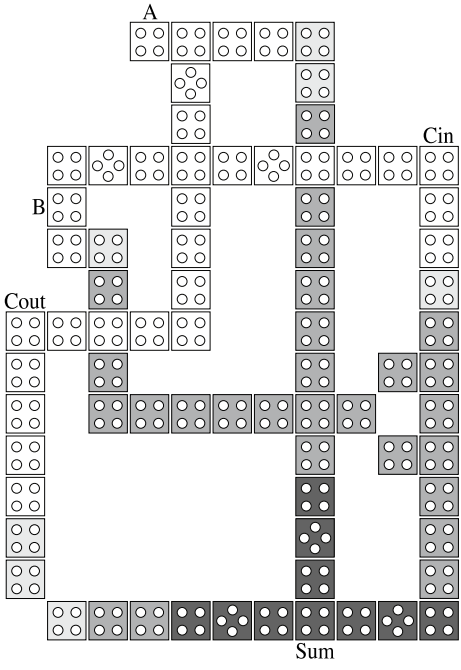


Fig. 25 Proposed Full-Adder *PUP*

3.3.1 Architecture UPP

Figure 14 corresponds to the full-adder from [19] with two fault-tolerant majority gates implemented in zones 2 and 3. The proposed full-adder with the same protection can be found in Fig. 15 where it can be seen that the inputs of the

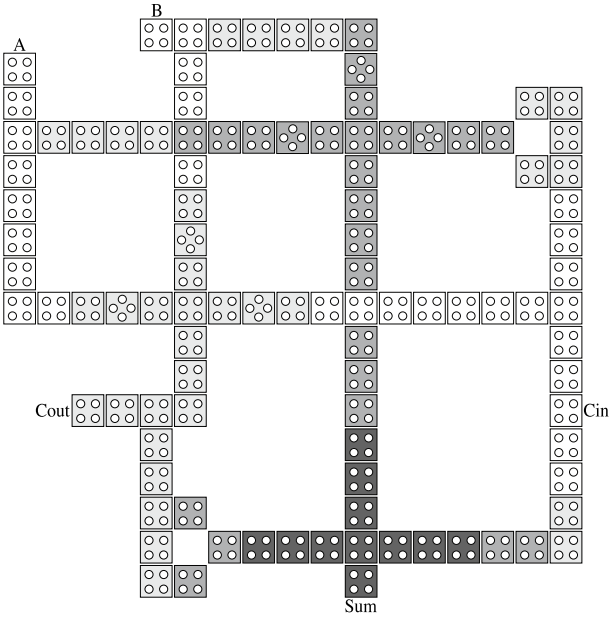
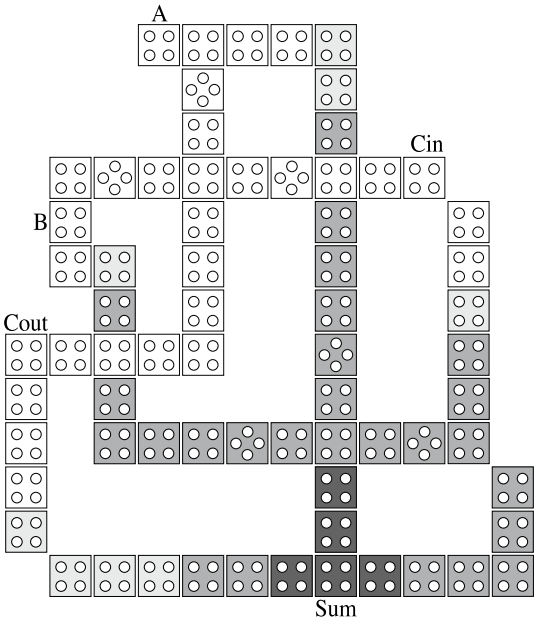


Fig. 26 Full-Adder *PPU* from [19]

Fig. 27 Proposed Full-Adder *PPU*



majority gate of zone 2 have been redistributed to take advantage of the area in an optimal way. It can also be seen that, by distributing the regions in which the majority gates are found in a descending way, it is possible to reduce the total

Table 3 fault-tolerant full-adders with different protection levels compared to full-adder from [19] and [32]

Faul-Tolerant Protection		UUP	UPU	UPP	PUU	PUP	PPU	PPP
Num cells	[19]	87	90	90	102	102	102	102
	[32]	72	73	78	80	85	91	92
	Proposed	58	61	63	66	76	69	72
Area (μm^2)	[19]	0.11	0.11	0.11	0.14	0.14	0.14	0.13
	[32]	0.06	0.06	0.07	0.08	0.08	0.09	0.08
	Proposed	0.07	0.06	0.07	0.06	0.08	0.07	0.08
Total Energy Dissipation (eV)	[19]	2.2^{-2}	2.6^{-2}	2.0^{-2}	2.8^{-2}	2.2^{-2}	2.6^{-2}	1.9^{-2}
	[32]	2.4^{-2}	2.3^{-2}	2.2^{-2}	2.8^{-2}	2.9^{-2}	2.9^{-2}	2.9^{-2}
	Proposed	2.0^{-2}	1.7^{-2}	1.3^{-2}	2.4^{-2}	2.1^{-2}	2.4^{-2}	1.6^{-2}
Average Energy Dissipation (eV)	[19]	2.0^{-3}	2.3^{-3}	1.8^{-3}	2.5^{-3}	2.0^{-3}	2.3^{-3}	1.7^{-3}
	[32]	2.2^{-3}	2.1^{-3}	2.0^{-3}	2.5^{-3}	2.6^{-3}	2.6^{-3}	2.6^{-3}
	Proposed	1.8^{-3}	1.5^{-3}	1.1^{-3}	2.1^{-3}	1.9^{-3}	2.2^{-3}	1.5^{-3}

area of the circuit even more, resulting in a more compact full-adder with a certain level of protection.

3.3.2 Architecture PPP

To make a comparison of the fully protected version of the full-adder from [19] and the proposed protected full-adder, the fault-tolerant majority gates have been implemented in all the three zones, as can be seen in Figs. 16 and 17, respectively.

The main difference between both approaches is the distribution of the majority gates. The majority gates in zones 1 and 2 are redistributed in Fig. 17, reducing the number of cells in the circuit. In zone 3, the majority gate corresponding to Fig. 12 is also implemented in Fig. 17.

Finally, the clock zones in the full-adder from [19] are distributed as follows: clock 1 (zone 1), clock 2 (zone 2) and clock 3 (zone 3), as can be seen in Fig. 16. The proposed full-adder has the following distribution: clock 0 (zone 1), clock 2 (zone 2) and clock 3 (zone 3). This change allows clock 1 only to be used for clock transitions in a wire, as shown in Fig. 17, impacting power consumption.

3.3.3 Semi-protected architectures

The architectures that can be implemented according to the desired level of protection are described as follows. The following figures can be seen in the rest of the designs: UUP, UPU, PUU, PUP, and PPU. First, it was necessary to modify the reference architecture (PPP) from [19] to create and simulate the different partially protected reference designs. Subsequently, an architecture with partial protection was designed for each of the cases mentioned above. Reference and proposed designs have been compared to obtain the values in terms of number of cells, area, total energy dissipation, and average energy dissipation per cycle, which can be seen in

Table 4 improvement percentages of fault-tolerant full-adders with different protection levels compared to full-adder from [19]

Faul-Tolerant Protection	UUP	UPU	UPP	PUU	PUP	PPU	PPP
Reduction of num cells %	33.33	32.22	30.00	35.29	25.49	32.35	29.41
Reduction of area (μm^2) %	36.36	45.45	36.36	57.14	42.85	50.00	38.46
Reduction of total energy dissipation (eV) %	6.36	33.33	36.27	15.79	7.05	6.54	14.36
Reduction of average energy dissipation per Cycle (eV) %	6.50	33.61	36.22	15.83	7.25	6.36	14.12

Table 5 Latency values of fault-tolerant full-adders with different protection levels compared to full-adder from [19]

Faul-Tolerant Protection		UUP	UPU	UPP	PUU	PUP	PPU	PPP
Latency	[19]	4	4	4	4	4	4	4
	This work	4	4	4	4	4	4	4

Table 3. In addition, the two architectures that can be found in the literature with a better trade-off between the number of cells, area, and energy dissipation are also included in the table for comparison purposes. For the sake of simplicity, we only analyze in the text the metrics of [19], as it is the most efficient. Similar conclusions can be reached with [32].

4 Results

The results obtained after simulating all the possible levels of protection against errors allow us to observe that the proposed architectures improve in the number of cells, area, total energy dissipation, and average energy dissipation per cycle, compared to the protected full-added from [19] and the derived semi-protected architectures.

To compare each combination's values, it has been necessary to perform 14 simulations (7 simulations with the adapted architectures from [19] and 7 simulations of the proposed full-adders).[2]

Table 4 shows the percentage of improvement obtained with our proposal compared to the fault-tolerant full-adder of [19]. The margin for improvement of each parameter is as follows: the number of cells from 25.49% in version PUP to 35.29% in version PUU, area from 36.36% in version UUP and UPP to 57.14% in PUU, total

[2] Note that, comparisons to CMOS technology are omitted as it is outside of this work's scope and is already analyzed in the literature by other authors.

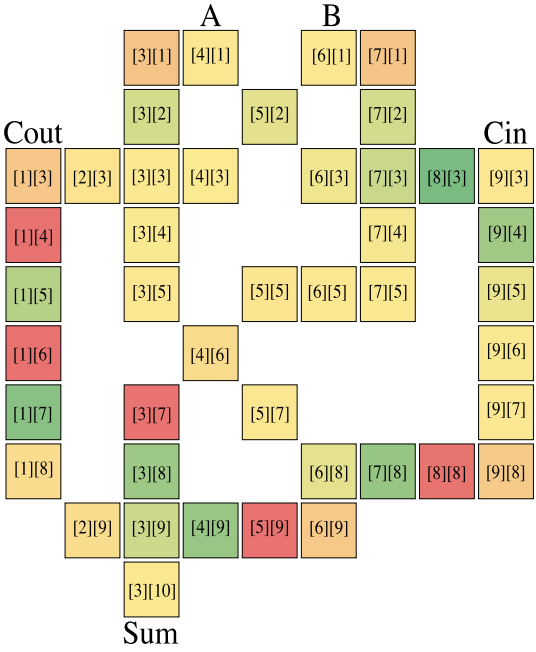


Fig. 28 Thermal energy dissipation maps for the full-adder UUU from [19]

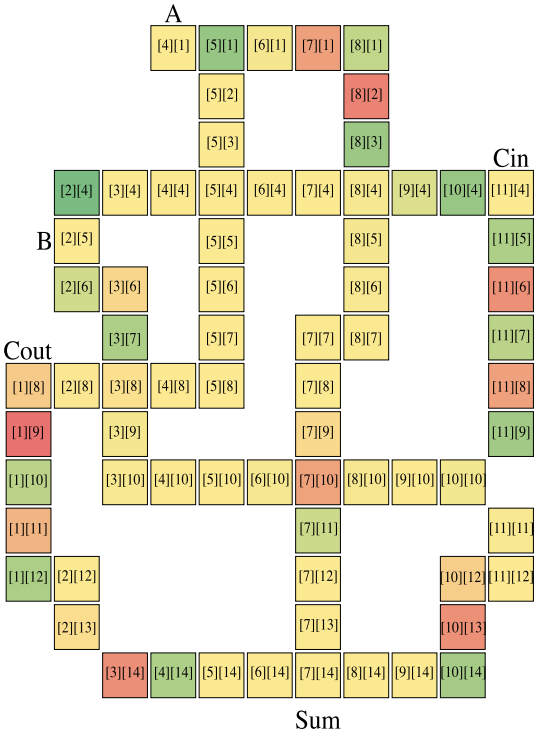


Fig. 29 Thermal energy dissipation maps for the proposed fault-tolerant full-adder PPP

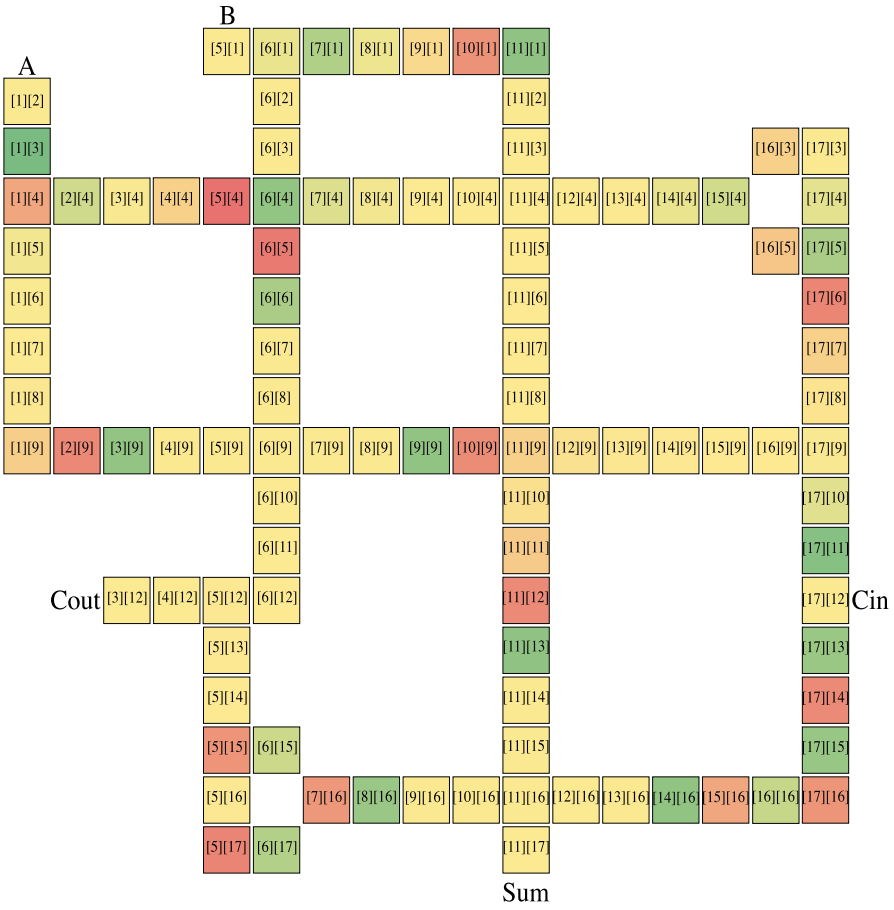


Fig. 30 Thermal energy dissipation maps for the fault-tolerant full-adder PPP from [19]

energy dissipation from 6.36% in version UUP to 36.27% in UPP and the average energy dissipation per cycle from 6.36% in combination PPU to 36.22% in UPP.

Table 5 shows that latency is the same for all the designs, so the area and power savings do not penalize speed.

Next, we include the thermal energy dissipation maps in Figs.28, 29 and 30, obtained with QCADesigner for the standard full-adder (unprotected), the fault-tolerant proposal with complete protection (PPP), and the reference proposed in the literature also for the fault-toleration version (PPP). As it can be seen for the regular full-adder, there are five critical cells from a thermal energy dissipation point of view (red cells), for the fault-tolerant proposal with complete protection (PPP) the number of critical cells (between red and dark orange cells) increase to eight, which is coherent with the increase in cells. The same happens with the reference proposed in the literature, as it has a larger number of cells, it has a larger number of critical cells as well, about thirteen (between red and dark orange cells).

Finally, to analyze the robustness of the proposed architectures, we simulated the errors of single-cell omission, extra-cell deposition, and cell displacement defects in the fault-tolerant majority gates integrated in the different designs obtaining: 100% tolerance against the single-cell omission error, 100% tolerance against extra-cell deposition error and 100% tolerance with cell displacement smaller than 7nm north, 5nm south, and 6nm east and west.

5 Conclusion

This paper introduces efficient full-adder architecture designs with different degrees of fault-tolerant protection. The architectures described here go from a design in which all the majority logic gates are fault-tolerant, which is the most robust solution, to a design in which just one gate is protected. This flexibility will allow other designers to choose an adder based on the trade-off between reliability and area/energy dissipation, which can be useful in larger QCA circuits that do not require full accuracy in their results or need larger protection for certain bits. All the architectures included here are based on full-custom designs validated with QCADesigner. Results show an area improvement between 36.36% and 57.14%, with a total energy dissipation between 6.36% and 36.27%, average energy dissipation per cycle between 6.56% and 36.22% compared to previous works found in the literature. As this circuit is widely used in digital processing, all the savings introduced here will have a greater impact on other structures such as ALUs or DSPs designed with QCA especially in applications that need different levels of reliability in their processing.

References

1. Powell JR (2008) The Quantum Limit to Moore's Law. Proc IEEE 96(8):1247. https://doi.org/10.1109/JPROC.2008.925411
2. Soares T, Nizer Rahmeier J, Lima V, Augusto Lascasas Freitas L, Melo L, Vilela Neto O (2018) NMLSim: a nanomagnetic logic (NML) circuit designer and simulation tool. J Comput Electron 17:1370. https://doi.org/10.1007/s10825-018-1215-8
3. Gu Z, Nowakowski ME, Carlton D, Storz R, Hong J, Chao W, Lambson B, Bennett P, Alam MT, Marcus MA, Doran A, Young A, Scholl A, Bokor J (2014) Speed and reliability of nanomagnetic logic technology, arXiv : Mesoscale and Nanoscale Physics
4. Wolkow RA, Livadaru L, Pitters J, Taucer M, Piva P, Salomons M, Cloutier M, Martins BVC (2014) Silicon Atomic Quantum Dots Enable Beyond-CMOS Electronics, Field-Coupled Nanocomputing: Paradigms, Progress, and Perspectives (Springer Berlin Heidelberg), pp. 33–58. https://doi.org/10.1007/978-3-662-43722-3-3
5. Karunaratne DK, Bhanja S (2012) Study of single layer and multilayer nano-magnetic logic architectures. J Appl Phys 111(7):07A928. https://doi.org/10.1063/1.3676052
6. Lent CS, Tougaw PD, Porod W, Bernstein GH (1993) Quantum cellular automata. Nanotechnol 4(1):49. https://doi.org/10.1088/0957-4484/4/1/004
7. Orlov A, Amlani I, Bernstein G, Lent CS, Snider GL (1997) Realization of a functional cell for quantum-dot cellular automata. Sci 277(5328):928. https://doi.org/10.1126/science.277.5328.928

8. Farazkish R, Sayedsalehi S, Navi K (2012) Novel design for quantum dots cellular automata to obtain fault-tolerant majority gate. J Nanotechnol. https://doi.org/10.1155/2012/943406
9. Walus K, Dysart TJ, Jullien GA, Budiman RA (2004) QCADesigner: a rapid design and Simulation tool for quantum-dot cellular automata. IEEE Trans Nanotechnol 3(1):26. https://doi.org/10.1109/TNANO.2003.820815
10. Torres FS (2015) Manual for QCADesigner-Energy (QD-E). github.com/FSillT/QCADesigner-E/blob/master/ManualQDE.pdf
11. Patterson JLHD (2018) Computer Organization and Design: The Hardware / Software Interface (RISC-V Edition) (Morgan Kaufmann)
12. Hauck ADS (2008) Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation (Morgan Kaufmann)
13. Safoyev N, Jeon J (2018) Coplanar QCA adders for arithmetic circuits. Int J Eng Technol (UAE) 7:15. https://doi.org/10.14419/ijet.v7i4.4.19597
14. Safoyev N, Jeon JC (2020) Design and evaluation of cell interaction based vedic multiplier using quantum-dot cellular automata, Electron 9(6). https://doi.org/10.3390/electronics9061036
15. Rasouli Heikalabad S, Salimzadeh F, Zirak Barughi Y (2020) A unique three-layer full adder in quantum-dot cellular automata. Comput Electr Eng.
16. Ahmadpour SS, Mosleh M, Rasouli Heikalabad S (2019) Robust QCA full-adders using an efficient fault-tolerant five-input majority gate. Int J Circuit Theory Appl 47:1037. https://doi.org/10.1002/cta.2634
17. Abedi D, Jaberipur G, Sangsefidi M (2015) Coplanar full adder in quantum-dot cellular automata via clock-zone-based crossover. IEEE Trans Nanotechnol 14(3):497. https://doi.org/10.1109/TNANO.2015.2409117
18. Mokhtari D, Rezai A, Rashidi H, Rabiei F, Emadi S, Karimi A (2018) Design of novel efficient full adder architecture for Quantum-dot Cellular Automata technology. FACTA UNIVERSITATIS Ser Electron Energ. https://doi.org/10.2298/FUEE1802279M
19. Ahmadpour SS, Mosleh M, Rasouli Heikalabad S (2020) The design and implementation of a robust single-layer QCA ALU using a novel fault-tolerant three-input majority gate. J Supercomput 76:10155. https://doi.org/10.1007/s11227-020-03249-3
20. Liu W, Zhang T, McLarnon E, O'Neill M, Montuschi P, Lombardi F (2019) Design and Analysis of Majority Logic Based Approximate Adders and Multipliers, IEEE Transactions on Emerging Topics in Computing pp. 1–5. https://doi.org/10.1109/TETC.2019.2929100
21. Niemier M (2004) Designing digital systems in quantum cellular automata. Ph.D. thesis, University of Notre Dame
22. Reis D (2016) Robustness analysis and enhancement strategies for quantum-dot cellular automata structures. Ph.D. thesis
23. Danehdaran F, Bagherian Khosroshahy M, Navi K, Bagherzadeh N (2018) Design and power analysis of new coplanar one-bit full-adder cell in quantum-dot cellular automata. J Low Power Electron 14:38. https://doi.org/10.1166/jolpe.2018.1529
24. Kavitha SS, Kaulgud N (2017) Quantum dot cellular automata (QCA) design for the realization of basic logic gates, pp. 314–317. https://doi.org/10.1109/ICEECCOT.2017.8284519
25. Shin S, Jeon JC, Kee-Young Y (2014) Design of wire-crossing technique based on difference of cell state in quantum-dot cellular automata. Int J Control Autom. https://doi.org/10.14257/ijca.2014.7.4.14
26. Hosseinzadeh H, Heikalabad SR (2018) A novel fault tolerant majority gate in quantum-dot cellular automata to create a revolution in design of fault tolerant nanostructures, with physical verification. Microelectron Eng 192(C):52. https://doi.org/10.1016/j.mee.2018.01.019
27. Du H, Hongjun L (2016) Design and analysis of new fault-tolerant majority gate for quantum-dot cellular automata. J Comput Electron 15:1484. https://doi.org/10.1007/s10825-016-0918-y
28. Kumar D, Mitra D (2016) Design of a practical fault-tolerant adder in QCA. Microelectron J 53:90. https://doi.org/10.1016/j.mejo.2016.04.004
29. Sun M, Lv H, Zhang Y, Xie GJ (2018) the fundamental primitives with fault-tolerance in quantum-dot cellular automata. J Electron Test 34:1. https://doi.org/10.1007/s10836-018-5723-z
30. Farazkish R (2018) Novel efficient fault-tolerant full-adder for quantum-dot cellular automata. Int J Nano Dimens 9(1):58
31. Wang X, Xie G, Deng F, Quan Y, Lü H (2018) Design and comparison of new fault-tolerant majority gate based on quantum-dot cellular automata, J Semicond 39(8). https://doi.org/10.1088/1674-4926/39/8/085001

32. Ahmadpour SS, Mosleh M (2019) New designs of fault-tolerant adders in quantum-dot cellular automata. Nano Commun Netw 19:10. https://doi.org/10.1016/j.nancom.2018.11.001
33. Moghimizadeh T, Mosleh M (2019) A novel design of fault-tolerant RAM cell in quantum-dot cellular automata with physical verification. J Supercomput 75:5688. https://doi.org/10.1007/s11227-019-02812-x