Abstract—Energy dissipation of future integrated systems, consisting of a myriad of devices, is a challenge that cannot be solved solely by emerging technologies and process improvements. Even though approaches like Field-Coupled Nanocomputing allow computations near the fundamental energy limits, there is a demand for strategies that enable the recycling of bits’ energy to avoid thermalization of information. In this direction, we propose a new kind of partially reversible systems by exploiting fan-outs in logic networks. We have also introduced a computationally efficient method to evaluate the gain obtained by our strategy. Simulation results for state-of-the-art benchmarks indicate an average reduction of the fundamental energy limit by 17% without affecting the delay. If delay is not the main concern, the average reduction reaches even 51%. To the best of our knowledge, this work presents the first post-synthesis strategy to reduce fundamental energy limits for Field-Coupled Nanocomputing circuits.

I. INTRODUCTION

Recent studies indicate that there might be only four to eight generations left before reaching the energetic boundary for Complementary Metal-Oxide Semiconductor (CMOS) technology [1], [2]. Despite the tremendous progress of integrated technologies, and even with the uprise of promising emerging technologies, the advancement of future systems could be strongly restricted by energy dissipation issues. Surprisingly, the cause of this problem is not a device property, but a direct consequence of the Second Law of Thermodynamics called Landauer’s limit [3]. This is the fundamental energy limit and it is related to the information loss in the form of heat, which occurs when bits are irreversibly erased within logic operations. It is on the order of $k_B T$ Joules per erased bit, where $k_B$ is the Boltzmann constant and $T$ is the temperature of the system’s thermal environment. Despite the controversies around it, Landauer’s idea was experimentally verified, confirming that there is a physical limit in irreversible computation [4]–[7].

CMOS technology can never reach Landauer’s minimum due to the Landauer-Shannon limit that is about 50× higher than Landauer’s limit [8]. However, several emerging technologies, like Quantum-dot Cellular Automata (QCA) or Nanomagnetic Logic (NML), are not restricted by the Landauer-Shannon bound [6], [9]. Consequently, the fundamental energy limits of these nanotechnologies are defined by irreversible logic operations.

The main contribution of this paper is the exploration of the architecture of Field-Coupled Nanocomputing (FCN) circuits in order to replace conventional logic gates by its reversible equivalents such that the fundamental energy limit is reduced. The proposed algorithm permits the designer to prioritize energy dissipation or delay and has been implemented and tested for QCA circuits. To the best of our knowledge, this work presents the first post-synthesis strategies to reduce fundamental energy limits for FCN designs.

The remaining of this paper is organized as follows. Section II introduces QCA and Reversible Computing techniques. Section III presents the proposed approach, while Section IV discusses the simulation results for state-of-the-art benchmarks [10]. Finally, Section V concludes this work.

II. BACKGROUND

A. Quantum-dot Cellular Automata

QCA is a Field-Coupled nanotechnology and a potential alternative to traditional CMOS technologies [9], [11]–[21]. Each QCA cell consists of four quantum dots, which are structures able to confine electric charges. These quantum dots are arranged in a square-like fashion such that free mobile electrons can move between them. Since these electrons impose mutual repulsion due to Coulomb interaction [11], they tend to locate themselves at opposite corners of the square. Thus, these electrons assume stable states, called polarizations, which are energetically equal and interpreted as binary 0 and 1. Further, when placed close to each other, the polarization of one QCA cell influences the polarization of the other—again by Coulomb interaction. The exploitation of this effect allows for the realization of logic gates.

B. Reversible Computing Techniques

A system is physically reversible if it can return to any previous state in reverse order as explained by Lent et al [12]. Nevertheless, turning a circuit reversible demands modifications of its architecture that usually penalize timing and/or area [22]. Hence, it is relevant to find a balanced trade-off between reduction of energy losses due to irreversible logic operations, and costs regarding area and delay. This requires a formal quantification of those losses, as proposed by Landauer [3]. He showed that the fundamental energy dissipation appears when devices irreversibly erase information, i.e., bits. His observation results from a statistical mechanics’ argument, which reveals that this loss is the result of the locally merging of states, happening within the conventional logic gates, e.g., AND, OR or Majority. In these gates, $n$-bit inputs are merged in a certain way to produce a 1-bit output.
The calculation of the Landauer limit is often erroneously understood. This led to recent efforts by the research community towards clarification of this energy assessment [23]–[26]. The minimum loss is the addition of the difference between Shannon’s entropies of gate’s input(s) and output(s). The main implication is that irreversible functions can be conditionally reversible with an adequate input combination subset (one that guarantees a 1-to-1 relation between input(s) and output(s)) [26].

A decade after Landauer’s groundbreaking work, Bennett presented a way to handle Landauer’s limit [27]. He showed that any logically irreversible function can be embedded in a logically reversible circuit. Lent and collaborators implemented this idea in the QCA technology by changing the clock timing [12]. However, this technique comes with the cost of strong degradation of the design throughput. Based on this initial approach, Ottavi et al. [15] proposed an intermediate solution that allows balancing energy dissipation and throughput by adding memory stages and a pipeline-like control scheme. Nevertheless, there remains a considerable performance penalty.

In their work, Lent et al. indicated a different approach in order to exploit Bennett’s idea [12]. The authors proposed echoing the inputs of QCA gates to the output, turning these gates reversible. This shall be explained by help of Fig. 1, which depicts three different implementations of a QCA AND gate: conventional, recovering information of one input (1-bit recycling gate) and fully reversible (2-bit recycling gate).

Lent et al. showed in [12] that the conventional QCA AND (see Fig. 1a) dissipates energy above \( k_B T \ln(2) \) Joules, when the value of one input differs from the output. However, if the input that differs from the output is echoed, as in the case of the 1-bit recycling (Fig. 1b) and fully reversible (Fig. 1c) versions, the dissipation remains below \( k_B T \ln(2) \) Joules. The drawbacks of this method are an increased complexity of the circuit and the requirement to treat intermediate results accordingly.

### III. Reducing Fundamental Energy Limits

Unfortunately, the evaluation of Landauer’s limit comes at high computational cost with exponential time and space complexity, turning it ineligible for comprehensive designs. Thus, we propose a simpler but practicable approach. The main idea of our method is the summation of the upper loss for each gate in the design, i.e., the maximum entropy that a gate can receive. Note that by this choice we are overestimating the losses and underestimate the gains since we are not considering conditionally reversible cases. We consider that this is a fair trade since we are now solving a polynomial problem instead of an exponential one and, thus, enable the evaluation of large circuits.

In case of conventional gates, e.g., the one depicted in Fig. 1a, we count the number of variable input bits as losses. In the other cases we consider each embedded fan-out as a recycled bit and are not considered as losses. Based on this method, we propose the exploitation of signals that are used more than once in the circuit. This gives us the ability to add gates that recycle input bits, in a way that we can reduce the

---

**Fig. 1:** Symbols and layouts of conventional as well as \( n \)-bits recycling QCA gates [12]

**Fig. 2:** Application of proposed method for Half-Adder circuit.
Algorithm 1: Building chains of n-bits recycling gates

```
data : netlist ← circuit’s netlist
1 foreach fo ∈ sort_ranks_desc(netlist.fanouts) do
2     repeat
3         chain ← ∅
4         foreach rank ∈ fo.ranked_children_asc do
5             children ← choose(rank)
6             chain ← chain ∪ children
7             rank ← rank ∖ children
8         fo.make_chain(chain)
9     until |chain| ≤ 1
```

Fig. 3: Operation of algorithm for two circuits

(a) Generation of chains of 1-bit recycling gates (b) Integration of fully reversible gates

The examples in Fig. 3a demonstrate the operation of the algorithm for two benchmark circuits developed by Zografos et al. [28].

Example 2. In the initial version of the circuit depicted in Fig. 5a, the output of node N connects to all remaining nodes. During the first iteration, the algorithm selects and merges nodes A1, B1, C1 and D1 into a chain that starts at node N (marked as blue lines). Further, nodes A1, B1 and C1 are changed to its 1-bit recycling versions. In the following iteration, a second chain consisting of nodes A2 and D2 is created (marked as red lines), and node A2 is modified to a 1-bit recycling gate.

The circuit depicted in Fig. 5b illustrates how fully reversible gates are integrated. In the first iteration, the connection between nodes N1 and B1 (blue lines) is placed between nodes A and B1 and node A is modified to a 1-bit recycling gate. In the second iteration, the connection between nodes N2 and B2 (red lines) is moved between nodes A and B2 and node A is changed to a fully reversible gate.
TABLE I: Benchmark results

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Size</th>
<th>Inputs</th>
<th>Outputs</th>
<th>Original Delay</th>
<th>Original Energy</th>
<th>Diff</th>
<th>Energy</th>
<th>Diff</th>
<th>Energy</th>
<th>Diff</th>
<th>Energy</th>
<th>Diff</th>
<th>Delay</th>
<th>Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>router</td>
<td>113</td>
<td>60</td>
<td>30</td>
<td>9</td>
<td>242</td>
<td>-12%</td>
<td>212</td>
<td>-12%</td>
<td>171</td>
<td>-29%</td>
<td>14</td>
<td>1.56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>priority</td>
<td>498</td>
<td>128</td>
<td>8</td>
<td>10</td>
<td>1060</td>
<td>-13%</td>
<td>920</td>
<td>-13%</td>
<td>620</td>
<td>-42%</td>
<td>21</td>
<td>2.10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>voter</td>
<td>14078</td>
<td>1001</td>
<td>1</td>
<td>60</td>
<td>29313</td>
<td>-13%</td>
<td>25400</td>
<td>-13%</td>
<td>15464</td>
<td>-47%</td>
<td>132</td>
<td>2.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>arbiter</td>
<td>2179</td>
<td>256</td>
<td>129</td>
<td>14</td>
<td>4945</td>
<td>-23%</td>
<td>3792</td>
<td>-23%</td>
<td>2518</td>
<td>-49%</td>
<td>88</td>
<td>6.29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sqrt</td>
<td>52344</td>
<td>128</td>
<td>64</td>
<td>690</td>
<td>111752</td>
<td>-24%</td>
<td>84935</td>
<td>-24%</td>
<td>56889</td>
<td>-49%</td>
<td>6027</td>
<td>8.73</td>
<td></td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>7202</td>
<td>512</td>
<td>130</td>
<td>27</td>
<td>16056</td>
<td>-17%</td>
<td>13397</td>
<td>-17%</td>
<td>8121</td>
<td>-49%</td>
<td>1216</td>
<td>45.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>zc</td>
<td>1108</td>
<td>147</td>
<td>142</td>
<td>9</td>
<td>2285</td>
<td>-19%</td>
<td>1847</td>
<td>-19%</td>
<td>1140</td>
<td>-50%</td>
<td>114</td>
<td>12.67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bar</td>
<td>3054</td>
<td>135</td>
<td>128</td>
<td>14</td>
<td>6108</td>
<td>0%</td>
<td>6108</td>
<td>0%</td>
<td>3061</td>
<td>-50%</td>
<td>1027</td>
<td>73.36</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sin</td>
<td>7895</td>
<td>24</td>
<td>25</td>
<td>91</td>
<td>17265</td>
<td>-19%</td>
<td>13982</td>
<td>-19%</td>
<td>8678</td>
<td>-50%</td>
<td>607</td>
<td>6.67</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mem Ctrl</td>
<td>10362</td>
<td>1204</td>
<td>1231</td>
<td>18</td>
<td>21067</td>
<td>-19%</td>
<td>17158</td>
<td>-19%</td>
<td>10445</td>
<td>-50%</td>
<td>457</td>
<td>25.39</td>
<td></td>
<td></td>
</tr>
<tr>
<td>square</td>
<td>19201</td>
<td>64</td>
<td>128</td>
<td>36</td>
<td>39711</td>
<td>-13%</td>
<td>34702</td>
<td>-13%</td>
<td>19922</td>
<td>-50%</td>
<td>391</td>
<td>10.86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log2</td>
<td>37584</td>
<td>32</td>
<td>32</td>
<td>181</td>
<td>78889</td>
<td>-19%</td>
<td>63860</td>
<td>-19%</td>
<td>39790</td>
<td>-50%</td>
<td>1654</td>
<td>9.14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>adder</td>
<td>2982</td>
<td>256</td>
<td>129</td>
<td>12</td>
<td>7332</td>
<td>-21%</td>
<td>5813</td>
<td>-21%</td>
<td>3611</td>
<td>-51%</td>
<td>72</td>
<td>6.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mult</td>
<td>41936</td>
<td>128</td>
<td>128</td>
<td>111</td>
<td>92915</td>
<td>-17%</td>
<td>77351</td>
<td>-17%</td>
<td>45130</td>
<td>-51%</td>
<td>1477</td>
<td>13.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>int2float</td>
<td>246</td>
<td>11</td>
<td>7</td>
<td>9</td>
<td>529</td>
<td>-21%</td>
<td>420</td>
<td>-21%</td>
<td>256</td>
<td>-52%</td>
<td>53</td>
<td>5.89</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cavlc</td>
<td>745</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>1625</td>
<td>-25%</td>
<td>1212</td>
<td>-25%</td>
<td>775</td>
<td>-52%</td>
<td>119</td>
<td>10.82</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ctrl</td>
<td>127</td>
<td>7</td>
<td>26</td>
<td>5</td>
<td>264</td>
<td>-25%</td>
<td>198</td>
<td>-25%</td>
<td>110</td>
<td>-58%</td>
<td>37</td>
<td>7.40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dec</td>
<td>420</td>
<td>8</td>
<td>256</td>
<td>4</td>
<td>840</td>
<td>0%</td>
<td>840</td>
<td>0%</td>
<td>172</td>
<td>-80%</td>
<td>259</td>
<td>64.75</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

AVERAGE 11226 228 145 73 24011 19564 -17% 12048 -51% 765 17.34

* Delay is given as number of stages.
* Energy means the approximation of the fundamental energy limit, calculated with the proposed method.
* Compared to the original netlist.

IV. Simulation Results

We applied our approach to the EPFL benchmarks used by Testa [29]. The results are summarized in Table II. Here, the first column shows the benchmark names. The second, third and fourth columns display the size of the circuit (number of gates) and the amount of inputs and outputs, respectively. The fifth column lists the delay of the original circuits, which is identical to the number of stages. The sixth column presents the energy dissipation of the original circuits, which means the fundamental energy limit calculated with the method proposed in section III. The next two columns refer to the delay-oriented results, i.e., when no delay penalty is allowed. The seventh column shows the achieved fundamental energy limits, while column eight lists the difference to the original versions. The remaining four columns relate to the energy-oriented results, listing the new fundamental energy limits and the reduction compared to the original version as well as the new delay and the increase.

The results indicate that the fundamental energy limits could be reduced by up to 25% and on average by 17% if the delay had to remain constant. If delay was not restricted, fundamental energy limits decreased by up to 80% and on average by 51%, at the costs of a maximum delay increase of 73.36 $\times$ and on average by 17.34 $\times$. For both benchmarks bar and dec no energy reduction was possible if delay was prioritized. In these cases, our method could not use any of their fan-outs. On the other hand, when the delay is not a problem our method was able to reduce the energy for these circuits by 50% and 80%, respectively, at the cost of high delay degradation.

V. Conclusions

We proposed in this work a simple, but practicable approach to estimate energy dissipation by reversible gates in a complex circuit. The previous exact method comes at high computational cost with exponential time and space complexity, turning it ineligible for large designs. Then, we developed a new method for design partially reversible circuits in order to reduce the fundamental energy limits of Field-Coupled Nanocomputing based architectures. Simulation results for an established benchmark suite indicate that the presented technique can reduce the fundamental energy limits on average by 17% without any delay penalty. When delay is not a concern, energy can be decreased by up to 80% and on average by 51%, at an average performance penalty of 17.34 $\times$.

Acknowledgement

The authors would like to thank CAPES, CNPq and Fapemig for supporting this work.

References


