A Transduction-based Framework to Synthesize RSFQ Circuits

Shigeru Yamashita (NAIST), Katsunori Tanaka (NEC), Hideyuki Takada (Kyoto Univ.), Koji Obata and Kazuyoshi Takagi (Nagoya Univ.)
Outline of the Talk

A Transduction-based Framework to Synthesize RSFQ Circuits

- Background & Preliminaries
  - RSFQ Logic Circuits
  - RSFQ Logic Elements
- Framework for logic design of RSFQ circuits
  - Transduction Method
- Experimental Results
- Concluding Remarks
Background: Next Generation Tech.

- Reaching the Limit of Conventional Semiconductor Technology.
  - Increasing Power Dissipation
  - Increasing Relative Delay
  - Difficult Implementation, etc.
- Beyond Conventional Semiconductor Technology.
  - SET: Single Electron Transistor
  - RSFQ: Rapid Single Flux Quantum
- QC: Quantum Computer
Background: RSFQ Circuits (1/2)

- Based on Superconductor Technology
- **Very Low Energy** for Representing a Single Bit
- **Very High Speed**
  - Near to the Light Speed
- Proven Technology
  - Experimental Implementation of a Network Switch Circuit
- **High Cost for Cooling**
Background: RSFQ Circuits (2/2)

- Pulse Representation for Information Propagated
  - Single Flux Quantum = Minimal Unit of Flux
- Basic Element = Superconducting Ring + Josephson Junction (JJ)
- JJ: Switch to Bring a Pulse Into or Out of the Ring.
Background: RSFQ Logic Primitive (1/2)

One Pulse to Each Pair of Inputs

- $A_t$: 00
- $A_f$: 01
- $B_t$: 10
- $B_f$: 11

One Pulse from Four Outputs (Minterm)

2x2-Join

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_t$</td>
<td>$A_f$</td>
</tr>
<tr>
<td>![Graph]</td>
<td>![Graph]</td>
</tr>
</tbody>
</table>
Background: RSFQ Logic Primitive (2/2)

- In Our Framework:
  - SPL (Splitter)
    - Split One Pulse to Two
  - CB (Confluence Buffer)
    - Converging Two Paths
    - No Simultaneous Input of Two Pulses
  - 2x2-Join
    - Dual-Rail, Two-Input
    - One-of-Four Output

Our Motivation

How to design logic circuits by these primitives?
Proposed Framework

- Our Policies
  - Modification of Conventional Logic Design
  - Dual-Rail

- Our Framework
  - Application of Conventional 2-Input Mapper
  - 2x2-Join+CB+SPL → 2-AND/XOR Cell
  - Optimization by the Transduction Method
Dual-Rail Logic Design

- Pulse Representation of bits
  - Single-Rail is Insufficient.
- Dual-Rail
  - Data Line + Clock Line
    - Difficulty on Timing
  - 0-Data Line + 1-Data Line
    - Paired Data Lines (0-Data, 1-Data)
    - (Pulse, No) = Valid-0
    - (No, Pulse) = Valid-1
    - (No, No) = Invalid
    - (Pulse, Pulse) = Prohibited.
  - Can be realized easily by 2x2-Joins
Dual-Rail, 2-Input Cell

Arbitrary 2-Input Function Realized by 2x2-Join & CBs
RSFQ Logic Design Framework

1. Synthesize 2-input node (= 2x2-Join+CB) Circuit
   - By Conventional Method
   - With 2-input Mapping
2. Logic Optimization by Transduction Method

- RSFQ Logic Design
  - Efficient Sharing 2x2-Joins in Logic Optimization
  - Multi-Output Cell with SPLs
Multi-Output 2-Input Cell (1/2)
Multi-Output 2-Input Cell (2/2)

Element Reduction by Sharing:
2x2-Join: -1, CB: -1, SPL: -1
Generalized Sharing

- Many Sharings $\rightarrow$ Large Reduction
- Determine whether Possible to Share.
  - Completely Common Inputs $\rightarrow$ Easy
  - One or Two Different Inputs $\rightarrow$ ?

- Use of the Transduction Method
- Don’t-care-based Logic Optimization
- Useful for Finding Sharings of Cells with Uncommon Inputs
The Transduction Method (Permissible Functions)

Wire Replacement

0 → 0
1 → 1
* → 0/1
The Transduction Method (Wire Replacement)

Repeat until No Reduction

1. Extracting PFs
   - Don’t-Care condition for alternative wire

2. Replacement of Wires Satisfying the Conditions Extracted in 1.

Gate Count Reduction by Wire Replacement:
Many Candidates: the Gate with the Maximal Fanout
The Transduction Method for RSFQ (1/2)

- Initial circuit: 2-input gate circuit
- The same input gates can be considered as a group
  
  2-AND/XOR Cell
  
  - Can realize arbitrary 2-input function
  - 4 \( \land \) AND (w/ NOTs)
  - 1 \( \oplus \) XOR

![Diagram](image-url)
The Transduction Method for RSFQ (2/2)

- Removing a cell if all the gates has lost fan-out.
- Total # of fanouts of the virtual gates are used for the wire selection priority
**Experimental Results**

- **Initial Circuit:** by SIS as 2-input node circuit
- **32 % Cell Count Reduced by Proposed Method**

<table>
<thead>
<tr>
<th>Circuits</th>
<th>PI</th>
<th>PO</th>
<th>2x2</th>
<th>Conn.</th>
<th>Lev.</th>
<th>2x2</th>
<th>Conn.</th>
<th>Lev.</th>
<th>Time (s)</th>
<th>Shared</th>
<th>(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1355</td>
<td>41</td>
<td>32</td>
<td>234</td>
<td>468</td>
<td>17</td>
<td>174</td>
<td>348</td>
<td>14</td>
<td>1.17</td>
<td>6</td>
<td>3.4</td>
</tr>
<tr>
<td>C7552</td>
<td>207</td>
<td>108</td>
<td>1504</td>
<td>3059</td>
<td>32</td>
<td>994</td>
<td>2049</td>
<td>32</td>
<td>122.14</td>
<td>136</td>
<td>13.7</td>
</tr>
<tr>
<td>alu2</td>
<td>10</td>
<td>6</td>
<td>386</td>
<td>773</td>
<td>42</td>
<td>236</td>
<td>473</td>
<td>28</td>
<td>3.35</td>
<td>41</td>
<td>17.3</td>
</tr>
<tr>
<td>alu4</td>
<td>14</td>
<td>8</td>
<td>667</td>
<td>1334</td>
<td>43</td>
<td>479</td>
<td>958</td>
<td>36</td>
<td>11.26</td>
<td>84</td>
<td>17.5</td>
</tr>
<tr>
<td>cmb</td>
<td>16</td>
<td>4</td>
<td>44</td>
<td>88</td>
<td>6</td>
<td>25</td>
<td>51</td>
<td>5</td>
<td>0.01</td>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>dalu</td>
<td>75</td>
<td>16</td>
<td>1172</td>
<td>2344</td>
<td>36</td>
<td>809</td>
<td>1618</td>
<td>18</td>
<td>36.59</td>
<td>84</td>
<td>10.4</td>
</tr>
<tr>
<td>f51m</td>
<td>8</td>
<td>8</td>
<td>113</td>
<td>227</td>
<td>10</td>
<td>64</td>
<td>129</td>
<td>9</td>
<td>0.11</td>
<td>10</td>
<td>15.6</td>
</tr>
<tr>
<td>i8</td>
<td>133</td>
<td>81</td>
<td>1260</td>
<td>2520</td>
<td>19</td>
<td>971</td>
<td>1942</td>
<td>15</td>
<td>155.37</td>
<td>127</td>
<td>13.1</td>
</tr>
<tr>
<td>lal</td>
<td>26</td>
<td>19</td>
<td>88</td>
<td>177</td>
<td>8</td>
<td>61</td>
<td>123</td>
<td>8</td>
<td>0.06</td>
<td>7</td>
<td>11.5</td>
</tr>
<tr>
<td>adder</td>
<td>33</td>
<td>17</td>
<td>96</td>
<td>192</td>
<td>48</td>
<td>64</td>
<td>128</td>
<td>48</td>
<td>0.08</td>
<td>16</td>
<td>25</td>
</tr>
<tr>
<td>t481</td>
<td>16</td>
<td>1</td>
<td>1690</td>
<td>3380</td>
<td>20</td>
<td>1073</td>
<td>2146</td>
<td>19</td>
<td>110.66</td>
<td>57</td>
<td>5.3</td>
</tr>
<tr>
<td>term1</td>
<td>34</td>
<td>10</td>
<td>259</td>
<td>519</td>
<td>16</td>
<td>116</td>
<td>234</td>
<td>11</td>
<td>0.68</td>
<td>15</td>
<td>12.9</td>
</tr>
<tr>
<td>ttt2</td>
<td>24</td>
<td>21</td>
<td>182</td>
<td>364</td>
<td>10</td>
<td>127</td>
<td>254</td>
<td>10</td>
<td>0.41</td>
<td>19</td>
<td>15</td>
</tr>
<tr>
<td>x3</td>
<td>135</td>
<td>99</td>
<td>724</td>
<td>1448</td>
<td>14</td>
<td>548</td>
<td>1096</td>
<td>12</td>
<td>14.19</td>
<td>41</td>
<td>7.5</td>
</tr>
<tr>
<td>z4ml</td>
<td>7</td>
<td>4</td>
<td>44</td>
<td>88</td>
<td>9</td>
<td>12</td>
<td>25</td>
<td>8</td>
<td>0.01</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td><strong>68.0%</strong></td>
<td><strong>68.2%</strong></td>
<td><strong>82.7%</strong></td>
<td><strong>6.08</strong></td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2006/1/25
Concluding Remarks

- RSFQ Logic Design Framework
  - Initial circuit: Synthesize a 2-input node Circuit
  - Optimize by the Transduction Method for Multi-Output Cells
  - 2-AND/XOR $\rightarrow$ 2x2-Join+CB+SPL
- Conventional Logic Design Methods are applicable.
- Size Reduction by Sharing 2x2-Joins: 32.0% on Avg.
- Future work
  - Combination with BDD-based Synthesis
  - Extension for Other Types of RSFQ Logic Element