Synthesis of Reversible Synchronous Counters

Marek Perkowski
Portland State University, marek.perkowski@pdx.edu

Mozammel H.A. Khan
East West University, Bangladesh
Synthesis of Reversible Synchronous Counters

Mozammel H. A. Khan
East West University, Bangladesh
mhakhan@ewubd.edu

Marek Perkowski
Portland State University, USA
mperkows@cecs.pdx.edu

ISMVL 2011, 23-25 May 2011, Tuusula, Finland
Agenda

• Motivation
• Background
• Previous Works on Reversible Sequential Logic
• Reversible Logic Synthesis using PPRM Expressions
• Synthesis of Synchronous Counters
• Conclusion

ISMVL 2011, 23-25 May 2011, Tuusula, Finland
Motivation

• Reversible circuits dissipate less power than irreversible circuits
• Reversible circuits can be used as a part of irreversible computing devices to allow low-power design using current technologies like CMOS
• Reversible circuits can be realized using quantum technologies
Motivation (contd.)

• Reversible circuits have been implemented in ultra-low-power CMOS technology, optical technology, quantum technology, nanotechnology, quantum dot, and DNA technology

• Most of the reversible logic synthesis attempts are concentrated on reversible combinational logic synthesis
Motivation (contd.)

• Only limited attempts have been made in the field of reversible sequential circuits

• Most papers present reversible design of latches and flip-flops and suggest that sequential circuits be constructed by replacing the latches and flip-flops of traditional designs by the reversible latches and flip-flops
Motivation (contd.)

• In this paper, we concentrate on design of synchronous counters directly from reversible gates.
A gate (or a circuit) is reversible if the mapping from the input set to the output set is bijective.

The bijective mapping from the input set to the output set implies that a reversible circuit has the same number of inputs and outputs.
Background (contd.)

Figure 1. Commonly used reversible gates – symbols and truth tables
• Toffoli gate may have **more than three** inputs/outputs.

• In an $n \times n$ Toffoli gate, the first $(n – 1)$ inputs (say $A_1, A_2, \ldots, A_{n-1}$) are control inputs and the last input (say $A_n$) is the target input.

• The value of the target output is $P = A_1A_2\ldots A_{n-1} \oplus A_n$
The $1 \times 1$ and $2 \times 2$ gates are technology realizable primitive gates and their realization costs (quantum costs) are assumed to be one.

The $3 \times 3$ Toffoli gate can be realized using five $2 \times 2$ primitive gates.

The $3 \times 3$ Fredkin gate can be realized using five $2 \times 2$ primitive gates.
Background (contd.)

Figure 2. Realizations of (a) $4 \times 4$ ($\text{cost} = 10$, garbage = 1) and (b) $5 \times 5$ Toffoli gates ($\text{cost} = 15$, garbage = 2)
Previous Works on Reversible Sequential Logic


Previous Works on Reversible Sequential Logic (contd.)

- All the above works present reversible design of latches and flip-flops.
- They suggest that reversible sequential circuit can be constructed by replacing flip-flops and gates of traditional design by their reversible counterparts.
- The (non-clocked) latches have limited usefulness in practical sequential logic design.
Previous Works on Reversible Sequential Logic (contd.)

- **Level-triggered** flip-flops and **edge-triggered/master-slave** flip-flops have usefulness in sequential logic design
Previous Works on Reversible Sequential Logic (contd.)

TABLE I. Comparison of realization costs and number of garbage outputs (separated by comma) of level-triggered flip-flop and edge-triggered/master-slave flip-flop designs

<table>
<thead>
<tr>
<th>Ref</th>
<th>Level-triggered flip-flop</th>
<th>Edge-triggered/master-slave flip-flop</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RS</td>
<td>JK</td>
</tr>
<tr>
<td>[24]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[25]</td>
<td></td>
<td>12,4</td>
</tr>
<tr>
<td>[26]</td>
<td></td>
<td>6,2</td>
</tr>
<tr>
<td>[27]</td>
<td></td>
<td>26,5</td>
</tr>
<tr>
<td>[28]</td>
<td>18,3</td>
<td>12,3</td>
</tr>
</tbody>
</table>

ISMVL 2011, 23-25 May 2011, Tuusula, Finland
Rversible Logic Synthesis using PPRM Expression

- Positive Davio expansion on all variables results into PPRM expression

\[
f(x_1, x_2, \cdots, x_n) = f_0 \oplus x_i f_2
\]

\[
f_0 = f(x_1, \cdots, x_{i-1}, 0, x_{i+1}, \cdots, x_n)
\]

\[
f_1 = f(x_1, \cdots, x_{i-1}, 1, x_{i+1}, \cdots, x_n)
\]

\[
f_2 = f_0 \oplus f_1
\]
• An \( n \)-variable PPRM expression can be represented as

\[
\begin{align*}
    f(x_1, x_2, \cdots, x_n) &= f_{00\cdots0} \oplus f_{00\cdots0} x_n \oplus f_{00\cdots1} x_{n-1} \oplus \\
    &\quad f_{00\cdots1} x_{n-1} x_n \oplus \cdots \oplus \ f_{11\cdots1} x_1 x_2 \cdots x_{n-1} x_n \\
    (\forall i \in \{0,1\}^n) \ f_i &\in \{0,1\}
\end{align*}
\]

ISMVL 2011, 23-25 May 2011, Tuusula, Finland
Rversible Logic Synthesis using PPRM Expression (contd.)

Figure 3. Computation of PPRM coefficients from output vector

<table>
<thead>
<tr>
<th>$ABC$</th>
<th>$F$</th>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0</td>
<td>0 $f_0$</td>
<td>0 $f_0$</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>0</td>
<td>0 $f_0$</td>
<td>0 $f_0$</td>
<td>0</td>
</tr>
<tr>
<td>010</td>
<td>0</td>
<td>0 $f_0$</td>
<td>0 $f_0$</td>
<td>0</td>
</tr>
<tr>
<td>011</td>
<td>1</td>
<td>1 $f_2$</td>
<td>1 $f_2$</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>1 $f_0$</td>
<td>0 $f_0$</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>0</td>
<td>0 $f_1$</td>
<td>0 $f_1$</td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>1</td>
<td>1 $f_1$</td>
<td>0 $f_1$</td>
<td>0</td>
</tr>
<tr>
<td>111</td>
<td>1</td>
<td>0 $f_2$</td>
<td>0 $f_2$</td>
<td>0</td>
</tr>
</tbody>
</table>
Figure 3. Computation of PPRM coefficients from output vector

\[ F(A, B, C) = BC \oplus A \oplus AC \]
Rversible Logic Synthesis using PPRM Expression (contd.)

• The PPRM expression is written from the final coefficient vector \([00011100]^T\)

• The resulting PPRM expression for the given function in Figure 3 is

\[ F(A, B, C) = BC \oplus A \oplus AC \]
Rversible Logic Synthesis using PPRM Expression (contd.)

• The PPRM expression can be realized as a cascade of Feynman and Toffoli gates

Figure 4. Realization of PPRM expression as cascade of Feynman and Toffoli gates

\[ F = BC \oplus A \oplus AC \]
Synthesis of Synchronous Counter

• We construct truth table considering the clock input and the present states as the inputs and considering the next states as the outputs

• Then we calculate PPRM expression of all the outputs and realize them as cascade of Feynman and Toffoli gates

• The feedback from the next state output to the present state input is done by making a copy of the next state output using Feynman gate
Synthesis of Synchronous Counter (contd.)

- The synthesized counter is a level-triggered sequential circuit and clock pulse width has to determined based on the total delay of the circuit
- SHOULD WE DO THIS FOR REVERSIBLE SIMULATE?
- Quantum?
- Quantum is different
### Synthesis of Synchronous Counter (contd.)

Table II. Truth table and PPRM coefficients of the next state outputs for mod 8 up counter

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
<th>PPRM Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>CQ2tQ1tQ0t</td>
<td>Q2t+1Q1t+1Q0t+1</td>
<td>Q2t+1Q1t+1Q0t+1</td>
</tr>
<tr>
<td>0000</td>
<td>000</td>
<td>000</td>
</tr>
<tr>
<td>0001</td>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>0010</td>
<td>010</td>
<td>010</td>
</tr>
<tr>
<td>0011</td>
<td>011</td>
<td>000</td>
</tr>
<tr>
<td>0100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>0101</td>
<td>101</td>
<td>000</td>
</tr>
<tr>
<td>0110</td>
<td>110</td>
<td>000</td>
</tr>
<tr>
<td>0111</td>
<td>111</td>
<td>000</td>
</tr>
<tr>
<td>1000</td>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>1001</td>
<td>010</td>
<td>010</td>
</tr>
<tr>
<td>1010</td>
<td>011</td>
<td>000</td>
</tr>
<tr>
<td>1011</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>1100</td>
<td>101</td>
<td>000</td>
</tr>
<tr>
<td>1101</td>
<td>110</td>
<td>000</td>
</tr>
<tr>
<td>1110</td>
<td>111</td>
<td>000</td>
</tr>
<tr>
<td>1111</td>
<td>000</td>
<td>000</td>
</tr>
</tbody>
</table>
The PPRM expressions for the next state outputs are

\[ Q^{2}_{t+1} = Q^{2}_{t} \oplus CQ^{1}_{t}Q^{0}_{t} \]

\[ Q^{1}_{t+1} = Q^{1}_{t} \oplus CQ^{0}_{t} \]

\[ Q^{0}_{t+1} = Q^{0}_{t} \oplus C \]
Synthesis of Synchronous Counter by direct method

\[ Q_{2t+1} = Q_2_t \oplus CQ_1_tQ_0_t \]
\[ Q_{1t+1} = Q_1_t \oplus CQ_0_t \]
\[ Q_{0t+1} = Q_0_t \oplus C \]

Cost = 19
Garbage = 2

Figure 5. Reversible circuit for mod 8 up counter
Figure 5. Reversible circuit for mod 8 up counter.

\[ Q_{2t+1} = Q_{1t} Q_{0t} C \oplus Q_{2t} \]
\[ Q_{1t+1} = Q_{0t} C \oplus Q_{1t} \]
\[ Q_{0t+1} = C \oplus Q_{0t} \]
Synthesis of Synchronous Counter (contd.)

Figure 6. Traditional circuit for mod 8 up counter
mod 8 up counter by replacement method:

\[ Q_{0_{t+1}} = Q_{0_{t}} \oplus C \]
\[ Q_{1_{t+1}} = Q_{1_{t}} \oplus CQ_{0_{t}} \]
\[ Q_{2_{t+1}} = Q_{2_{t}} \oplus CQ_{1_{t}}Q_{0_{t}} \]

Cost = 24
Garbage = 4

- Figure 7. Reversible circuit for mod 8 up counter after replacement of the T flip-flops and AND gates of Figure 6 by their reversible counter parts.
Direct Synthesis of Mod 16 Synchronous Counter

We can determine the PPRM expressions for the next state outputs of mod 16 up counter as follows:

\[ Q^{3}_{t+1} = Q^{3}_t \oplus CQ^{2}_t Q^{1}_t Q^{0}_t \]
\[ Q^{2}_{t+1} = Q^{2}_t \oplus CQ^{1}_t Q^{0}_t \]
\[ Q^{1}_{t+1} = Q^{1}_t \oplus CQ^{0}_t \]
\[ Q^{0}_{t+1} = Q^{0}_t \oplus C \]
Direct Synthesis of Mod 16 Synchronous Counter

Cost = 35
Garbage = 4

\[ Q_{3_{t+1}} = Q_3_t \oplus C Q_2_t Q_1_t Q_0_t \]
\[ Q_{2_{t+1}} = Q_2_t \oplus C Q_1_t Q_0_t \]
\[ Q_{1_{t+1}} = Q_1_t \oplus C Q_0_t \]
\[ Q_{0_{t+1}} = Q_0_t \oplus C \]

Figure 8. Reversible circuit for mod 16 up counter

ISMVL 2011, 23-25 May 2011, Tuusula, Finland
Synthesis of classical Mod 16 Synchronous Counter

Figure 9. Traditional circuit for mod 16 up counter
Direct Synthesis of Reversible circuit for mod 16 up counter.

\[ Q_{3_{t+1}} = Q_{3_t} \oplus CQ_{2_t} Q_{1_t} Q_{0_t} \]

\[ Q_{2_{t+1}} = Q_{2_t} \oplus CQ_{1_t} Q_{0_t} \]

\[ Q_{1_{t+1}} = Q_{1_t} \oplus CQ_{0_t} \]

\[ Q_{0_{t+1}} = Q_{0_t} \oplus C \]
Flip-flop replacement method for Reversible circuit for mod 16 up counter.

Figure 10. Reversible circuit for mod 16 up counter after replacement of the T flip-flops and AND gates of Figure 9 by their reversible counter parts.
Synthesis of Synchronous Counter (contd.)

TABLE III. Comparison of our direct design and replacement technique for mod 8 and mod 16 up counters

<table>
<thead>
<tr>
<th>Counter</th>
<th>Our direct technique</th>
<th>Replacement technique</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cost</td>
<td>Garbage</td>
</tr>
<tr>
<td>mod 8</td>
<td>19</td>
<td>2</td>
</tr>
<tr>
<td>mod 16</td>
<td>35</td>
<td>4</td>
</tr>
</tbody>
</table>

CONCLUSION: Our method creates counters of smaller quantum cost and number of garbages than the previous methods.
Synthesis of Synchronous Counter (contd.)

• PPRM expressions of the next state outputs can be written in general terms as follows

\[ Q_{i+1} = Q_i \oplus CQ(i-1)Q(i-2)\cdots Q0 \quad \text{for } i > 0 \]
\[ Q_{0+1} = Q0 \oplus C \quad \text{for } i = 0 \]

• These generalized PPRM expressions allow us to implement any up counter directly from reversible gates very efficiently
Conclusions

1. Reversible logic is very important for low power and quantum circuit design.

2. Most of the attempts on reversible logic design concentrate on reversible combinational logic design [9-22].

3. Only a few attempts were made on reversible sequential circuit design [23-28, 32-35].

4. The major works on reversible sequential circuit design [23-27] propose implementations of flip-flops and suggest that sequential circuit be constructed by replacing the flip-flops and gates of the traditional designs by their reversible counter parts.
Conclusions 2

- These methods produce circuits with high realization costs and many garbages.

- We present a method of synchronous counter design directly from reversible gates.

- This method produces circuit with lesser realization cost and lesser garbage outputs.

- The proposed method generates expressions for the next state outputs, which can be expressed in general terms for all up counters.
Conclusions 3

• This generalization of the expressions for the next state outputs makes synchronous up counter design very easy and efficient.

• Traditionally, state minimization and state assignment are parts of the entire synthesis procedure of finite state machines.

• The role of these two processes in the realization of reversible sequential circuits [32,34] has been investigated by us.

• It should be further investigated.
Conclusions 4

• We showed a method that is specialized to certain type of counters.

• We created a similar method for quantum circuits which is specialized to other types of counters.

• T flip-flops are good for counters.

• T flip-flops are good for arbitrary state machines realized in reversible circuits.

• Excitation functions of T ffs are realized as products of EXORs of literals and Inclusive Sums of literals.

• Don’t’ cares should be used to realize functions of the form:

\[ Q_{i_{t+1}} = Q_{i_t} \oplus a \cdot b \cdot c \cdot d \cdot e \]

Linear variable decomposition – Kerntopf Habilitation


• M. Lukac, M. Kameyama, and M. Perkowski, *Quantum Finite State Machines - a Circuit Based Approach, Quantum Information Processing*, accepted with revisions