A Multi-Ported Memory Compiler Utilizing True Dual-port BRAMs

Ameer Abdelhadi and Guy Lemieux

Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, Canada

May 3rd, 2016
Motivation (1): FPGAs as parallel accelerators

- Used as parallel accelerators
  - Have dual-ported memories only

- 1,000’s Dual-Ported Block RAMs
- 1,000,000’s Logic Elements
- 1000’s Multipliers/DSPs
Motivation (2)
Mixed port requirements

- Multi-porting approaches provide simple (fixed) ports only
- Waste of resources if these ports are not active simultaneously
Live-Value Table (LVT)

Multi-read

- Replication

Multi-write

- LVT
Data Banks Optimization

• LVT-based multi-ported RAM is composed of:

- LVT-tracks changes
- Data banks - stores data copies

• Our previous work (I-LVT/FPGA'14) optimizes LVT only
• This work optimizes the data banks (not the LVT!)
• The first technique that requires a CAD tool
Data Banks Optimization

- LVT-based multi-ported RAM is composed of:
  1) LVT - tracks changes
Data Banks Optimization

• LVT-based multi-ported RAM is composed of:
  1) LVT - tracks changes
  2) Data banks - store data copies
Data Banks Optimization

- LVT-based multi-ported RAM is composed of:
  1) LVT - tracks changes
  2) Data banks - store data copies
- Our previous work (I-LVT/FPGA'14) optimizes LVT only
- This work optimizes the data banks (not the LVT!)
- The first technique that requires a CAD tool
Data Banks Optimization

- LVT-based multi-ported RAM is composed of:
  1) LVT - tracks changes
  2) Data banks - store data copies
- Our previous work (I-LVT/FPGA’14) optimizes LVT only
- This work
  - Optimizes the data banks (not the LVT!)

![Data Bank Diagram]
Data Banks Optimization

- LVT-based multi-ported RAM is composed of:
  1) LVT - tracks changes
  2) Data banks - store data copies
- Our previous work (I-LVT/FPGA’14) optimizes LVT only
- This work
  - Optimizes the data banks (not the LVT!)
  - The first technique that requires a CAD tool
Data Banks Optimization

• LVT-based multi-ported RAM is composed of:
  1) LVT - tracks changes
  2) Data banks - store data copies

• Our previous work (I-LVT/FPGA’14) optimizes LVT only

• This work
  • Optimizes the data banks (not the LVT!)
  • The first technique that requires a CAD tool

This work solves the final step and most important problem of Block RAM allocation
Mixed Port Requirements (1): Fixed ports

Fixed (simple) ports: The majority of multi-ported memories supports fixed ports only.
Mixed Port Requirements (2): True ports

True ports: Some techniques support the construction of multi-true-ports.

BRAMs in FPGAs are true dual-ported.
Mixed Port Requirements (3): Switched ports

Switched ports: A number of writes are switched with a number of reads

True ports are special case of switched ports
Switched Ports (1)

Example

Objectives:
Optimize the construction of multi-switched ports

Key Observation: BRAMs’ true ports can be utilized to optimize switched ports
Switched Ports (2)

Fixed ports abstraction
Switched Ports (3)
Fixed data banks

Fixed Ports

\[
\begin{array}{c}
W_0 & W_2 & W_3 \\
R_0 & R_2 & R_1 \\
\end{array}
\]

ALU

\[
\begin{array}{c}
f \\
\sqrt{x} \quad 1/x \\
\gg \\
g \\
f/g \\
en \\
0 \\
1 \\
en \\
\end{array}
\]

Shared bus

\[
\begin{array}{c}
bus \\
r/w \\
\end{array}
\]

I-LVT

\[
\begin{array}{c}
W_0 & BRAM0 \\
W_1 & BRAM1 \\
W_2 & BRAM3 \\
W_3 & BRAM9 \\
\end{array}
\]

\[
\begin{array}{c}
W_0 & BRAM2 \\
W_1 & BRAM4 \\
W_2 & BRAM6 \\
W_3 & BRAM10 \\
\end{array}
\]

\[
\begin{array}{c}
W_0 & BRAM5 \\
W_1 & BRAM7 \\
W_2 & BRAM8 \\
W_3 & BRAM11 \\
\end{array}
\]
Switched Ports (4)
DFG modeling

![Diagram of switched ports in DFG modeling](image)
Switched Ports (5)
Switched DFG
Switched Ports (6)

DFG Covering

\[
W_{1,0} \rightarrow W_1 \rightarrow R_1 \rightarrow R_{1,0} = W_{1,0} + R_{1,0}
\]

\[
W_{2,1} \rightarrow W_2 \rightarrow R_2 \rightarrow R_{2,0} = W_{2,1} + R_{2,0}
\]
Switched Ports (6)
DFG Covering

\[ \begin{align*}
W_{1,0} & \rightarrow W1 \rightarrow R1 \rightarrow R_{1,0} \\
W_{2,1} & \rightarrow W2 \rightarrow R2 \rightarrow R_{2,0}
\end{align*} \]

\[ W_{2,0} \rightarrow W1 \rightarrow R1 \rightarrow R_{2,0} = \]

Complete Bigraph
Switched Ports (6)
DFG Covering

Vertex → Port
Bidlique pattern → BRAM

Complete Bigraph
Switched Ports (6)

DFG Covering

<table>
<thead>
<tr>
<th>W2,0</th>
<th>W</th>
<th>R</th>
<th>R1,0 = W2,0</th>
<th>R1,0</th>
</tr>
</thead>
<tbody>
<tr>
<td>W0,0</td>
<td>W</td>
<td>R</td>
<td>R1,0 = W0,0</td>
<td>R1,0</td>
</tr>
<tr>
<td>W1,0</td>
<td>W1</td>
<td>R1</td>
<td>R1,0</td>
<td></td>
</tr>
<tr>
<td>W2,1</td>
<td>W2</td>
<td>R2</td>
<td>R2,0</td>
<td></td>
</tr>
<tr>
<td>W2,0</td>
<td>W1</td>
<td>R1</td>
<td>R2,0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>R2</td>
<td></td>
<td>R0,0</td>
<td></td>
</tr>
</tbody>
</table>

Vertex → Port
Bidique pattern → BRAM

Complete Bigraph
Switcheed Ports (6)

DFG Covering

\[ W_{20} \rightarrow W \rightarrow R \rightarrow R_{1,0} = W_{20} \rightarrow R_{1,0} \]
\[ W_{00} \rightarrow W \rightarrow R \rightarrow R_{1,0} = W_{00} \rightarrow R_{1,0} \]
\[ W_{10} \rightarrow W_{1} \rightarrow R_{1} \rightarrow R_{1,0} = W_{10} \rightarrow R_{1,0} \]
\[ W_{21} \rightarrow W_{2} \rightarrow R_{2} \rightarrow R_{2,0} = W_{21} \rightarrow R_{2,0} \]
\[ W_{00} \rightarrow W \rightarrow R \rightarrow R_{2,0} = W_{00} \rightarrow R_{2,0} \]
\[ W_{20} \rightarrow W_{1} \rightarrow R_{2} \rightarrow R_{2,0} = W_{20} \rightarrow R_{2,0} \]

Complete Bigraph

Vertex \rightarrow Port
Bidlique pattern \rightarrow BRAM

Complete Bigraph

R_{0,0}
W_{0,0}
W_{1,0}
W_{2,0}
W_{2,1}
R_{1,0}
R_{2,0}
Switched Ports (6)

DFG Covering

Switched Ports

- $W_{20} \rightarrow W \rightarrow R \rightarrow R_{1,0} = W_{20} \rightarrow R_{1,0}$
- $W_{00} \rightarrow W \rightarrow R \rightarrow R_{1,0} = W_{00} \rightarrow R_{1,0}$
- $W_{10} \rightarrow W_1 \rightarrow R_1 \rightarrow R_{1,0} = W_{10} \rightarrow R_{1,0}$
- $W_{21} \rightarrow W_2 \rightarrow R_2 \rightarrow R_{2,0} = W_{21} \rightarrow R_{2,0}$
- $W_{00} \rightarrow W \rightarrow R \rightarrow R_{2,0} = W_{00} \rightarrow R_{2,0}$
- $W_{20} \rightarrow W_1 \rightarrow R_1 \rightarrow R_{2,0} = W_{20} \rightarrow R_{2,0}$
- $W_{00} \rightarrow W \rightarrow R \rightarrow R_{0,0} = W_{00} \rightarrow R_{0,0}$

Vertex → Port
Bidlique pattern → BRAM

Complete Bigraph
Switched Ports (6) DFG Covering

\[
\begin{align*}
W_{20} & \rightarrow W \rightarrow R \rightarrow R_{1,0} = W_{20} \rightarrow R_{1,0} \\
W_{00} & \rightarrow W \rightarrow R \rightarrow R_{1,0} = W_{00} \rightarrow R_{1,0} \\
W_{10} & \rightarrow W1 \rightarrow R1 \rightarrow R_{1,0} = W_{10} \rightarrow R_{1,0} \\
W_{21} & \rightarrow W2 \rightarrow R2 \rightarrow R_{2,0} = W_{21} \rightarrow R_{2,0} \\
W_{00} & \rightarrow W \rightarrow R \rightarrow R_{2,0} = W_{00} \rightarrow R_{2,0} \\
W_{20} & \rightarrow W1 \rightarrow R1 \rightarrow R_{2,0} = W_{20} \rightarrow R_{2,0} \\
W_{20} & \rightarrow W2 \rightarrow R2 \rightarrow R_{0,0} = W_{20} \rightarrow R_{0,0} \\
W_{00} & \rightarrow W \rightarrow R \rightarrow R_{0,0} = W_{00} \rightarrow R_{0,0} \\
W_{10} & \rightarrow W \rightarrow R \rightarrow R_{0,0} = W_{10} \rightarrow R_{0,0}
\end{align*}
\]
Switched Ports (6)

DFG Covering

\[
\begin{align*}
W_{20} &\rightarrow W \rightarrow R_{1,0} = W_{20} \rightarrow R_{1,0} \\
W_{00} &\rightarrow W \rightarrow R_{1,0} = W_{00} \rightarrow R_{1,0} \\
W_{10} &\rightarrow W_{1} \rightarrow R_{1,0} = W_{1} \rightarrow R_{1,0} \\
W_{21} &\rightarrow W_{2} \rightarrow R_{2,0} = W_{21} \rightarrow R_{2,0} \\
W_{00} &\rightarrow W \rightarrow R_{2,0} = W_{00} \rightarrow R_{2,0} \\
W_{20} &\rightarrow W_{1} \rightarrow R_{2,0} = W_{20} \rightarrow R_{2,0} \\
W_{00} &\rightarrow W \rightarrow R_{0,0} = W_{00} \rightarrow R_{0,0} \\
W_{10} &\rightarrow W \rightarrow R_{0,0} = W_{10} \rightarrow R_{0,0} \\
W_{21} &\rightarrow W \rightarrow R_{0,0} = W_{21} \rightarrow R_{0,0}
\end{align*}
\]
Switched Ports (7)
Switched data banks

W_{20} \rightarrow W \rightarrow R \rightarrow R_{1,0}
W_{00} \rightarrow W \rightarrow R \rightarrow R_{1,0}
W_{10} \rightarrow W_1 \rightarrow R_1 \rightarrow R_{1,0}
W_{21} \rightarrow W_2 \rightarrow R_2 \rightarrow R_{2,0}
W_{00} \rightarrow W \rightarrow R \rightarrow R_{2,0}
W_{20} \rightarrow W_1 \rightarrow R_1 \rightarrow R_{2,0}
W_{20} \rightarrow W_1 \rightarrow R_1 \rightarrow R_{2,0}
W_{00} \rightarrow W \rightarrow R \rightarrow R_{0,0}
W_{00} \rightarrow W \rightarrow R \rightarrow R_{0,0}
W_{10} \rightarrow W \rightarrow R \rightarrow R_{0,0}
W_{21} \rightarrow W \rightarrow R \rightarrow R_{0,0}
Switched Ports (7)
Switched data banks
Switched Ports (7)
Switched data banks

Switched Ports (Optimized Bigraph)

Fixed Ports (Complete Bigraph)

8 BRAMs (33% reduction)

12 BRAMs
Multi-switched-ports Compiler

• A RAM compiler optimizes data banks construction
  • Generates DFG from port requirements
  • Solves set-covering problem on all edges
    • Covers are predefined biclique patterns
    • Solved as BLP problem
  • Generates Verilog modules based on optimal covering

Supports bypassing (RAW & RDW) and Initialization

Available as open source contribution
https://github.com/AmeerAbdelhadi
http://www.ece.ubc.ca/~lemieux/downloads/
Graphical User Interface (GUI)
Source of inspiration: Multi-True-Ports by Choi et al. / UofT

- Provides true ports only (no simple/fixed ports)
- Is a special case of our generalized approach
- Doesn't need a CAD tool
Experimental Results

• Run-in-batch flow manager
  • Uses Altera’s Quartus II for synthesis on Stratix V
  • Uses Altera’s ModelSim for verification with:
    • Random vectors
    • Over a million RAM access cycles

• Results on random test-cases
  • Up to 8 switched ports
  • Up to 4 writes and 4 reads per switched port
  • Up to 28 writes/reads per test-case

<table>
<thead>
<tr>
<th></th>
<th>Average BRAM Reduction</th>
<th>Average ALM Reduction</th>
<th>Average Fmax Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best of Previous</td>
<td>18%</td>
<td>-3%</td>
<td>-1%</td>
</tr>
<tr>
<td>True Ports</td>
<td>42%</td>
<td>53%</td>
<td>15%</td>
</tr>
</tbody>
</table>
Conclusions

• A methodology to support switched write/read functionality
• True dual-ported BRAMs are utilized to optimize the RAM allocation
• A RAM compiler optimizes the problem
• An additional 18% average BRAM reduction compared to the best of other approaches
• Practical solution:
  • Initialization
  • Bypassing
  • Available as open source
Future Directions

• Applications
  • Parallel computation
  • HLS – storage binding

• Optimization of switched ports port assignment
  • Extraction of mutually-exclusive functions from HDL

• Statistical approach
  • Ports which are mutually-exclusive in most cases can use a switched port
  • Access conflicts will be rare
Thank You!