PushPull: Short Path Padding for Timing Error Resilient Circuits

Yu-Ming Yang
Iris Hui-Ru Jiang
Sung-Ting Ho

IRIS Lab
National Chiao Tung University
Outline

- Introduction
- Problem Formulation
- Algorithm - PushPull
- Experimental Results
- Conclusion
Timing characterization is difficult due to a wide range of dynamic variations
- Supply voltage droops, process variations, etc.
- A timing guardband is reserved to ensure correct functionality
  - Degrade circuit performance, i.e., limit the clock frequency
Resilient Circuit

- Eliminate the guardband by error detection and correction
  - Ex: Razor flip-flop (one error-detection circuit)
  - Correction through instruction replay
  - Cons: short path issue

D. Ernst et al., “Razor: a low-power pipeline based on circuit-level timing speculation” MICRO, 2003
Short Path Issue

- Short paths should exceed the error detection window
  - Otherwise, may detect false timing errors
- Require a significant hold time margin for short paths
- Focus on short path padding in this paper
Previous Work

- **Determine the padding delay path by path**
  - Combine with clock skew scheduling to minimize the clock period at logic synthesis stage
    - J.P. Fishburn, IEEE TC, 1990
    - R.B. Deokar and S.S. Sapatnekar, ISCAS, 1994
    - S.H. Huang et al., DAC, 2007

- **Determine the gate/wire to insert padding delay**
  - Solve by linear programming
    - N. V. Shenoy et al., ICCAD, 1993
  - Heuristics
    - Pad the gate with the largest setup slack
    - Pad the gate passed by most hold violating paths
      - Y. Liu et al., DAC, 2011
Short Path Padding Example

The largest setup slack

Legend:
- ID: gate
- (R/A/S, r/a/H): edge
- : PI/PO
- : Flip-flop

<table>
<thead>
<tr>
<th>ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Short Path Padding Example (cont’d)

Legend:

- **ID** (R/A/S, r/a/H): gate
- **edge**: edge
- **PI/PO**: PI/PO
- **FF**: Flip-flop

<table>
<thead>
<tr>
<th>ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Delay</strong></td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Short Path Padding Example (cont’d)

Optimal padding delay = 0.5

<table>
<thead>
<tr>
<th>ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Our Contributions

- **Find the padding values and locations with a global view**
  - Compute the fanout padding flexibility of each gate

- **Realize delay padding at the post-layout stage**
  - Coarse-grained delay padding: by spare cells
  - Fine-grained delay padding: by dummy metal
Problem Formulation

- **Given**
  - Placed and routed resilient design
  - Cell library
  - Spare cells and dummy metal information
  - Target clock period and error detection window

- **Goal**
  - Pad short paths
  - Satisfy Setup/hold timing constraints
  - Minimize the padding overhead
Delay Padding

- Lengthen short paths by **buffer insertion** and **extra capacitance hook-up**

Buffer insertion

*Padding wire*

Extra cap hook-up

*Padding gate*
Outline

- Introduction
- Problem Formulation
- Algorithm - PushPull
- Experimental Results
- Conclusion
Overview of PushPull

Start

- netlist, DEF/SPEF file
- cell library, timing constraints

Padding Value Determination
- Padding resource collection
- Fanout padding flexibility calculation / Feasibility checking
- Padding value decision

Y
- Improvement?

N
- Padding value refinement

Load/Buffer Allocation
- Spare cell selection
- Dummy metal allocation
- Timing analysis

End
Example: Padding Value Determination

First, pad gates

Legend:

<table>
<thead>
<tr>
<th>ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Example: Padding Value Determination

Second, pad wires

Legend:

<table>
<thead>
<tr>
<th>ID</th>
<th>gate</th>
<th>edge</th>
<th>PI/PO</th>
<th>Flip-flop</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R/A/S, r/a/H)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delay</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Example: Load/Buffer Allocation

Legend:
- **ID** (R/A/S, r/a/H): gate
- **: edge**
- **: PI/PO**
- **: Flip-flop**

<table>
<thead>
<tr>
<th>ID</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
</tr>
<tr>
<td>3</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Padding Value Determination

Start

- netlist, DEF/SPEF file
- cell library, timing constraints

Padding Value Determination

- Padding resource collection
- Fanout padding flexibility calculation / Feasibility checking
- Padding value decision
  - Y: Improvement?
  - N: Padding value refinement

Load/Buffer Allocation

- Spare cell selection
- Dummy metal allocation
- Timing analysis
  - End
Fanout Padding Flexibility Calculation

- **Fanout padding flexibility** $P_F(i)$ of gate $i$
  - Reflect the maximum padding value allowed on its whole fanout cone with a global view

$$P_F(i, j) = \begin{cases} 
0, & \text{if } g_i \in PO \text{ or } H(i) \geq 0; \\
\min\{0, \min\{H'(i, j)|e(i, j) \in E\}\} - H(i), & \text{otherwise.}
\end{cases}$$
Padding Value Decision

- Decide the padding value of each gate in the topological order
  - Only pad the remaining slack
  
  \[ P(i) = \max\{P_{sa}(i) - P_F(i), 0\} \]

- Graphical representation of the padding values and gate connections.
Padding Value Decision

- Iterate until no more improvement
Padding Value Refinement

- Further reduce the total padding values with forked path
  - Push the padding values toward the gates where two or more paths fork

```
A = 0.2
a = 0.0

(0.9/0.6/0.3, 0.4/0.4/0.0)
```

```
P(2) = 0.3
P(3) = 0.3
P(1) = 0.3
```

```
R = 1.0
r = 0.5
```

```
(1.0/0.7/0.3, 0.5/0.5/0.0)
```

```
(0.5/0.5/0.0, 0.1/0.1/0.0)
```

```
(0.9/0.6/0.3, 0.4/0.4/0.0)
```

```
P(1) = 0.2
```

```
P(2) = 0.3
P(3) = 0.3
P(1) = 0.3
```

```
R = 1.0
r = 0.5
```

```
(1.0/1.0/0.0, 0.5/0.5/0.0)
```

```
(1.0/1.0/0.0, 0.5/0.5/0.0)
```

```
R = 1.0
r = 0.5
```

```
FF1
```

```
FF2
```

Joined short paths

```
A = 0.4
a = 0.0
```

```
Total = 0.6
Total = 0.3
```

Forked short paths

```
A = 0.4
a = 0.0
```

```
Total = 0.5
Total = 0.3
```

```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
Buffer Insertion

- When hold violations cannot be fully cleaned by padding on gates
  - Padding on wires (buffer insertion)

![Diagram](image-url)

- **FF1**
  - \( A=0.0 \)
  - \( a=0.0 \)
  - \( P(2)=0.3 \)
  - \((0.6, 0.0)\)

- **i1**
  - \( A=0.6 \)
  - \( a=0.3 \)

- **P(3)=0.2**
  - \( (0.7, 0.0)\)
  - \( (1.0/0.3/0.7, 0.3/0.3/0.0)\)

- **FF2**
  - \( A=0.0 \)
  - \( a=0.0 \)
  - \( P(1)=0.1 \)
  - \( (1.2/1.2/0.0, 0.5/0.5/0.0)\)
Load/Buffer Allocation

Start

- netlist, DEF/SPEF file
  cell library, timing constraints

Padding Value Determination

- Padding resource collection
- Fanout padding flexibility calculation / Feasibility checking
- Padding value decision
- Improvement?
  - Y
  - N
    Padding value refinement

Load/Buffer Allocation

- Spare cell selection
- Dummy metal allocation
- Timing analysis
- End
Spare Cell Selection (1/2)

- Extract the available spare cells located within the bounding box of its fanout net

- Find suitable spare cell candidates for gate/wire
  - Reduce to the subset sum problem

<table>
<thead>
<tr>
<th>Spare cells</th>
<th>Padding value</th>
</tr>
</thead>
<tbody>
<tr>
<td>∅</td>
<td>0.00 0.05 0.10 0.15 0.20 0.25</td>
</tr>
<tr>
<td>{s₁}</td>
<td>0.00 ∅    ∅    ∅    ∅    0.20 {s₁} 0.20 {s₁}</td>
</tr>
<tr>
<td>{s₁, s₂}</td>
<td>0.00 ∅    ∅    ∅    0.15 {s₂} 0.20 {s₁} 0.20 {s₁}</td>
</tr>
<tr>
<td>{s₁, s₂, s₃}</td>
<td>0.00 ∅    ∅    0.10 {s₃} 0.15 {s₂} 0.20 {s₁} 0.25 {s₂, s₃}</td>
</tr>
</tbody>
</table>
Spare Cell Selection (2/2)

- Solve the competition problem between different padding gates/wires
  - Record multiple subset sum solutions
  - Sort the gates/wires in ascending order of the number of spare cell solutions
  - Assign the gates/wires to their best solution in the sorted order
Dummy Metal Allocation

- Fix remaining padding by dummy metal insertion
  - Convert the delay to an amount of capacitance
  - Assign un-overlapped metal resource
  - Resort the unfixed padding to maximum network flow
Outline

- Introduction
- Problem Formulation
- Algorithm - PushPull
- Experimental Results
- Conclusion
## Experimental Results

- **Platform:** Intel® Xeon® CPU E5620 @ 2.40GHz with 16GB memory, CentOS 5.5
- **Compare with two greedy heuristics**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TNS₁(ps)</td>
<td>THS₁(ps)</td>
<td>Padding delay(ps)</td>
</tr>
<tr>
<td>s1196</td>
<td>0.0</td>
<td>0.0</td>
<td>152.5</td>
</tr>
<tr>
<td>s1423</td>
<td>0.0</td>
<td>0.0</td>
<td>4,746.9</td>
</tr>
<tr>
<td>s5378</td>
<td>0.0</td>
<td>0.0</td>
<td>3,722.7</td>
</tr>
<tr>
<td>s9234</td>
<td>0.0</td>
<td>0.0</td>
<td>1,647.5</td>
</tr>
<tr>
<td>s13207</td>
<td>0.0</td>
<td>0.0</td>
<td>371.2</td>
</tr>
<tr>
<td>s15850</td>
<td>0.0</td>
<td>0.0</td>
<td>1,510.8</td>
</tr>
<tr>
<td>s38584</td>
<td>0.0</td>
<td>0.0</td>
<td>143,764.1</td>
</tr>
<tr>
<td>des_perf</td>
<td>0.0</td>
<td>0.0</td>
<td>583,579.0</td>
</tr>
<tr>
<td>b19</td>
<td>0.0</td>
<td>0.0</td>
<td>1,675,688.0</td>
</tr>
</tbody>
</table>

TNS₁: total negative setup slack after padding value determination.
THS₁: total negative hold slack after padding value determination.

Experimental Results (cont’d)

- **Compare with LP solutions**

<table>
<thead>
<tr>
<th>Circuit</th>
<th>#Gate</th>
<th>#FF</th>
<th>#SFF</th>
<th>Conservative clock period (ns)</th>
<th>THS (ps)</th>
<th>LP</th>
<th>Ours: PushPull</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1196</td>
<td>301</td>
<td>19</td>
<td>1</td>
<td>1.0</td>
<td>152.5</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>s1423</td>
<td>486</td>
<td>74</td>
<td>45</td>
<td>1.0</td>
<td>4,916.9</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>s5378</td>
<td>739</td>
<td>162</td>
<td>37</td>
<td>1.0</td>
<td>3,852.8</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>s9234</td>
<td>555</td>
<td>132</td>
<td>24</td>
<td>1.0</td>
<td>1,929.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>s13207</td>
<td>748</td>
<td>213</td>
<td>14</td>
<td>1.0</td>
<td>371.2</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>s15850</td>
<td>428</td>
<td>128</td>
<td>29</td>
<td>1.0</td>
<td>2,114.6</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>s38584</td>
<td>7,890</td>
<td>1,159</td>
<td>194</td>
<td>3.4</td>
<td>108,759.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>des_perf</td>
<td>51,349</td>
<td>8,808</td>
<td>1,190</td>
<td>2.9</td>
<td>583,579.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>b19</td>
<td>72,872</td>
<td>5,541</td>
<td>2,737</td>
<td>3.8</td>
<td>1,481,800.0</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

**TNS**: total negative setup slack after padding value determination.

**THS**: total negative hold slack after padding value determination.

- **The impact of input slew on padding delay capacitance conversion**
  - The average error rate over all cases is only 0.56%.

LP: N. V. Shenoy et al., "Minimum padding to satisfy short path constraints," ICCAD-93.
Experimental Results (cont’d)

- Compare our load/buffer allocation method with LP+Mapping

<table>
<thead>
<tr>
<th>Circuit</th>
<th>LP+Mapping</th>
<th>Ours: PushPull</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TNS₂ (ps)</td>
<td>THS₂ (ps)</td>
</tr>
<tr>
<td>s1196</td>
<td>0.0</td>
<td>0.3</td>
</tr>
<tr>
<td>s1423</td>
<td>0.0</td>
<td>826.4</td>
</tr>
<tr>
<td>s5378</td>
<td>0.0</td>
<td>302.3</td>
</tr>
<tr>
<td>s9234</td>
<td>0.0</td>
<td>631.3</td>
</tr>
<tr>
<td>s13207</td>
<td>0.0</td>
<td>161.2</td>
</tr>
<tr>
<td>s15850</td>
<td>0.0</td>
<td>352.0</td>
</tr>
<tr>
<td>s38584</td>
<td>0.0</td>
<td>29,970.7</td>
</tr>
<tr>
<td>des_perf</td>
<td>0.0</td>
<td>51,146.0</td>
</tr>
<tr>
<td>b19</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

TNS₂: total negative setup slack after load/buffer allocation.
THS₂: total negative hold slack after load/buffer allocation.

Conclusion

- **Focus on the severe short path padding problem in resilient circuits**
  - Enable the timing error detection and correction mechanism of resilient circuits

- **Determine the padding values and locations with a global view**
  - vs. the local view adopted by resent prior work

- **Realize the determined padding values at physical implementation**
  - Propose coarse-grained and fine-grained load/buffer allocation by using spare cells and dummy metal
Future Work

- **Allocate spare cell and dummy metal simultaneously**
  - Spare cells: discrete capacitance resource
  - Dummy metal: continuous capacitance resource

- **Adopt Composite Current Source (CCS) timing model**
  - Use more accuracy delay model
Thank You!

Contact info:
Yu-Ming Yang
yuming.yyang@gmail.com
Backup Slides
The Resilient Circuit Design Flow

- Logic synthesis
- Timing analysis
- $S_{th}/H_{th}$ selection
- Resynthesis
- Timing analysis
- Resilient circuit replacement
- Placement & routing
- Short path padding