Concurrency Reduction of Untimed Latch Protocols – Theory and Practice

Async 2010

Santosh N. Varanasi, Kenneth S. Stevens, and Graham Birtwistle

University of Utah    University of Sheffield

Supported by a gift from Sun Microsystems
Concurrency and Protocols

Systems are complex. We want to:

1. Better understand concurrency and synchronization
2. Examine composition of concurrent protocols
3. Verify complex systems
4. Validate concurrency reduction assumptions about hardware

Photo: Intel – 80 core teraflops chip
Theory Overview

1. Created formal abstract model
   - for linear pipelines
   - covers all designs in protocol class

2. Concurrency reduction rules
   - start with maximum state space
   - well defined rules and properties
   - 250 / 4900 possible untimed / timed protocols
Practice Overview

Evaluate complete family of 137 untimed protocols from Theory

- Scripted flow evaluating cycle time, latency, area, energy
- Results from physical P&R design in
  - IBM 65nm 10sf process, Artisan 12T static library

- See how 17 published untimed protocol compare to 114 others
- Evaluation assumptions of concurrency reduction effects
1. Delay Insensitive Abstraction
   - $n$-bit data path can be abstracted as single bit path

2. Bundled Data abstraction
   a. prove that abstraction is correct
   b. prove that concurrency reduction supports abstraction

We have shown this for most published designs
Formal Model for Datapath Abstraction

Executive summary: *Serializing the latch handshake in process L allows us to prove correctness of abstraction and concurrency reduction approach.*

1. Liveness property holds due to concurrency reduction rules (next foils)
2. Process $L$ ensures input data integrity holds
3. Process $V$ ensures output data validity holds
4. Process $S$ ensures output data integrity holds
5. Protocol timing ensures input data validity holds

\[
\begin{align*}
L &= lr↑.gS.\overline{rEn}↑.aEn↑.\overline{rEn}↓.aEn↓.pV.\overline{Ta}↑.lr↓.\overline{Ta}↓.L \\
R &= gV.\overline{rr}↑.ra↑.pS.\overline{rr}↓.ra↓.R \\
S &= \overline{gS}.pS.S \\
V &= \overline{pV}.gV.V \\
LC &= (L \mid R \mid S \mid V) \setminus \{gS, pS, gV, pV\} \\
LATCH &= rEn↑.open.\overline{aEn}↑.rEn↓.closed.\overline{aEn}↓.LATCH \\
max &= (LC \mid LATCH) \setminus \{rEn, aEn\}
\end{align*}
\] (1)
Maximum Concurrency Abstracted Protocol

As a petri net (very much like process on previous foil):

- ra- \[\rightarrow\] rr+ \[\rightarrow\] ra+ \[\rightarrow\] rr-

- lr+ \[\rightarrow\] la+ \[\rightarrow\] lr- \[\rightarrow\] la-

As minimized state graph, in particular configuration called **shape**
Concurrency Reduction (*Cuts*)
from Most Concurrent Design (*Shape*)

- **Shape** is representation of state space in particular configuration
- Arcs not shown for clarity
- Reduce concurrency by removing states from left or right of rows
Concurrency Reduction (*Cuts*) from Most Concurrent Design (*Shape*)

- **Shape** is representation of state space in particular configuration
- Arcs not shown for clarity
- Reduce concurrency by removing states from left or right of rows

Removed (or **cut**) 2 states from right of first and third rows
Concurrency Reduction with Right Cuts

- Remove states from right end of rows.

- Denoted $R_{abcd}$

- Liveness constraints
  - limits extend of cuts (initial state must be reachable)
  - imply “above” states must be removed

Right Cut Constraints:

\[
0 \leq a, b \leq 4 \\
0 \leq c, d \leq 8 \\
a \geq b \land b + 4 \geq c \land c \geq d \land d \geq a
\]
Concurrency Reduction with Left Cuts

- Modify shape representation for convenience
  - top row duplicated below bottom row
  - cut labels start on second row
Concurrency Reduction with Left Cuts

- Remove states from left end of rows
- Denoted $\text{Labcd}$
- Liveness constraints
  - limits extend of cuts
  - imply “below” states must be removed

Left Cut Constraints:

\[
0 \leq a, b, c, d \leq 3
\]

\[
a \leq b \land b \leq c \land c \leq d
\]
Protocol Family with Left and Right Cuts

- Combine Left $L$ and Right $R$ cuts for concurrency reduction on protocol family
- Liveness restricts cuts

Combined Cut Liveness Constraints:

\[
La + Ra < 5 \land Lb + Rb < 5 \land
La + Rb < 5 \land
Lb + Rc < 9 \land Lc + Rd < 9 \land
Lc + Rc < 9 \land Ld + Rd < 9
\]
Correctness of Concurrency Reduction Rules

1. Complete and Coherent Symmetric Lattices

2. Each cut has complement around **self-symmetric axis**:

   \[
   L_{3333} = L(a_0 + d_1)(b_0 + c_1)(c_0 + b_1)(d_0 + a_1) \quad R_{4488} = R(a_0 + b_1)(b_0 + a_1)(c_0 + d_1)(d_0 + c_1)
   \]
Current Status

1. Abstraction Model

2. Concurrency reduction rules
   - complete and coherent
   - liveness
   - occupancy
   - pipeline behavior
     - family equivalence classes
     - delay-insensitive
     - speed-independent regular
     - speed-independent irregular

3. Practice trailing theory

Parallel pipeline equivalence classes
## Practice: Concurrency Reduction on Protocols

Compare all 137 protocols under concurrency reduction.

**What is the impact of concurrency reduction on designs?**

**Can we determine which protocol in the family is best?**

**What is the best implementation for equivalent protocols when pipelined?**

### Coverage of published protocols

<table>
<thead>
<tr>
<th></th>
<th>L0000</th>
<th>L0011</th>
<th>L1111</th>
<th>L0022</th>
<th>L1122</th>
<th>L0033</th>
<th>L1133</th>
<th>L2222</th>
<th>L2233</th>
<th>L3333</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>R0000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R0020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R0040</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R0022</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R0042</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2022</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2042</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R0044</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2044</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R4044</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2222</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2242</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2262</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2244</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2264</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R4244</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R4264</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Practice: Concurrency Reduction on Protocols

Complex interplay between:

- Concurrency reduction simplifies logic
- Simpler logic assumed to be faster
- Concurrency reduction adds protocol delays
- Hazard removal and technology mapping creates “noise”

Our expectation:

- Optimal will lie somewhere in middle of concurrency reduction spectrum

Will this hold globally as well as across protocol equivalent classes?
**Circuit Complexity and Concurrency Reduction**

Number of explicit state variables generated by Petrify

<table>
<thead>
<tr>
<th>L0000 L0011 L1111</th>
<th>L0022 L1122 L0033 L1133</th>
<th>L2222 L2233 L3333</th>
<th>L ∘ R</th>
</tr>
</thead>
<tbody>
<tr>
<td>– – 2</td>
<td>4 – – 2</td>
<td>2 2 2</td>
<td>R0000</td>
</tr>
<tr>
<td>3 – 2</td>
<td>2 2 2</td>
<td>2 1 1</td>
<td>R0020</td>
</tr>
<tr>
<td>2 2 2</td>
<td>2 2 2</td>
<td>2 1 1</td>
<td>R0040</td>
</tr>
<tr>
<td>3 3 2</td>
<td>1 2 2 1</td>
<td>2 1 1</td>
<td>R0022</td>
</tr>
<tr>
<td>2 2 2</td>
<td>– 2 2 1</td>
<td>2 1 1</td>
<td>R0042</td>
</tr>
<tr>
<td>2 2 2</td>
<td>2 2 2</td>
<td>2 1</td>
<td>R2022</td>
</tr>
<tr>
<td>2 1 1</td>
<td>1 1 1 0</td>
<td>1 0</td>
<td>R2042</td>
</tr>
<tr>
<td>3 2 2</td>
<td>2 2 2 1</td>
<td>2 1 1</td>
<td>R0044</td>
</tr>
<tr>
<td>2 2 1</td>
<td>1 1 1 0</td>
<td>1 0</td>
<td>R2044</td>
</tr>
<tr>
<td>2 1</td>
<td>. 1 .</td>
<td>. .</td>
<td>R4044</td>
</tr>
<tr>
<td>2 2 2</td>
<td>2 2 2 1</td>
<td>2 1</td>
<td>R2222</td>
</tr>
<tr>
<td>2 1</td>
<td>2 1 1 0</td>
<td>1 0</td>
<td>R2242</td>
</tr>
<tr>
<td>2 1</td>
<td>. 1 .</td>
<td>.</td>
<td>R2262</td>
</tr>
<tr>
<td>2 1 1</td>
<td>1 1 1 0</td>
<td>1 0</td>
<td>R2244</td>
</tr>
<tr>
<td>2 1</td>
<td>1 0 .</td>
<td>. .</td>
<td>R2264</td>
</tr>
<tr>
<td>2 1</td>
<td>. 1</td>
<td>.</td>
<td>R4244</td>
</tr>
<tr>
<td>1 0</td>
<td>0 . .</td>
<td>. .</td>
<td>R4264</td>
</tr>
</tbody>
</table>

- “–”: no solution found
- “.”: deadlocking agent

**Complexity reduction assumption: generally holds**
Area and Concurrency Reduction

Effect of Concurrency Reduction on Area for Out and In Channels:

Assumption: Concurrency reduction reduces area

- Generally holds on both output ($L$) and input ($R$) channels
- L2233 cuts and R2244 best on average
- R2044 best cut for fully buffered protocol
Area and Concurrency Reduction

Circuit with smallest area and backward latency

- Half-buffered
- Protocol L2233 $\circ$ R2244
- Only three gates
Forward Latency

Effect of Concurrency Reduction on Forward Latency

Assumption: Concurrency reduction optimized away from ends

- Generally holds due to competing tradeoffs
  - concurrency reduction on incoming channel generally reduces latency
  - outgoing channel $L$ produces tradeoff
    - generally decreasing
    - outgoing channel cuts delaying $rr \uparrow$ increase latency
    - other reductions have 2nd order effect of reducing latency
Forward Latency

Circuit with smallest forward latency

L0033 generally best cut for forward latency

- Half buffered protocol L0033°R4244 best
- Full buffered L0033°R2242 only 10ps slower
Assumption: Concurrency reduction optimized away from ends

- Generally holds except for half buffered protocols
  - outgoing channel $L$ latency generally reduced, though not significantly
  - incoming channel $R$ decreases irregularly based on delaying $la$
  - incoming half buffered protocols ultra-fast
    - every other stage stalls in different compatible state
Cycle Time

Effect of Concurrency Reduction on Backward Latency

Assumption: Concurrency reduction optimized away from ends

- Generally holds for output channel cuts $L$
  - $L1133$ generally the best cut
- Generally holds for input channel except for half buffered protocols
  - $R2042$ best full buffered protocol
  - half buffered similar to / affected by the backward latency drop-off
    - every other stage stalls in different *compatible* state
Cycle Time

Circuit with smallest cycle time

- Full buffered protocol
- Protocol L0022 ° R2042
Concurrency: Performance vs Parallelism

Assumption: ∃ a tradeoff between performance and parallelism

- Performance increases with concurrency reduction
- Protocol parallelism decreases with concurrency reduction

Evaluate comparing performance (area) with parallelism (cycle time)
Optimal Cuts vs Published Space

Most of the best cuts for latency (blue) and cycle time (green) have not been explored.

<table>
<thead>
<tr>
<th>L0000</th>
<th>L0011</th>
<th>L1111</th>
<th>L0022</th>
<th>L1122</th>
<th>L0033</th>
<th>L1133</th>
<th>L2222</th>
<th>L2233</th>
<th>L3333</th>
<th>L  \lor  R</th>
</tr>
</thead>
<tbody>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R0000</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R0020</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R0040</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R0022</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R0042</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R2022</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R2042</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R0044</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R0244</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R2044</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R4044</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R2244</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R2242</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R2262</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R2244</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R2264</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R4244</td>
</tr>
<tr>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>■</td>
<td>R4264</td>
</tr>
</tbody>
</table>
Conclusions

1. Abstraction approach correct for all protocols in family
2. Regular concurrency reduction rules covering all designs
   - limited here to untimed (DI & SI) protocols
3. Rules form complete, coherent, symmetric lattice
4. Complete protocol family was synthesized and characterized
5. Assumptions on effects of concurrency reduction evaluated
   - largely shown true: best design near mid range
   - surprising quality found for half buffered protocols
6. Better designs for certain applications uncovered
Backup Foils
Restricting to the Untimed Family

Untimed protocols (delay-insensitive and speed-independent) can not delay input signals.

- Can cut **green** output arcs to remove states
  - cuts states in horizontal and vertical pairs inside dashed boxes
The Speed-Independent (SI) Family

- Allow left and right cut rules to cut output arcs
  - subject to $\mathcal{L}$, $\mathcal{R}$, and liveness rules
- Permits output ordering
  - one output enables another output
    - e.g. for L0011 $la \uparrow$ enables $rr \downarrow$

Speed Independent Constraints:

$L$: $a = b \land c = d$

$R$: $a, b, c, d$ are even
The Delay-Insensitive (DI) Family

Speed-independent constraints plus:

- Output signals may only be delayed by inputs
  - outputs cannot enable other outputs
  - output ordering is disallowed
  - cuts of $2 \times 2$ states (right or bottom pair of output arc square)

Delay-Insensitive Constraints:

$$\forall \{ L, R \}: a = b \land c = d \land a, b, c, d \text{ are even}$$