Optimizing Sequential Cycles through Shannon Decomposition and Retiming

Cristian Soviani
Olivier Tardieu
Stephen A. Edwards

Department of Computer Science, Columbia University

{soviani,tardieu,sedwards}@cs.columbia.edu
Shannon transform - review

arbitrary combinational logic

propagation delay through f

multiplexer delay
Retiming - review

Improves performance by re-distributing the registers
Motivation: speed up this sample

Observations:

- Retiming can not improve performance because of the loop
- Shannon seems useless, as all inputs arrive at the same time (0)
Sample after Shannon transform

Period: 9

Observations:
- the performance is worse
- the loop is much smaller
Sample after Shannon and retiming

Shannon transform(s) and retiming: a huge design space

- finding the best combination is not trivial
- we want a **systematic** way to explore it
Presentation outline. Contributions

- new **model** for describing complex combinations of Shannon decompositions
- feasible arrival time (**fat**) sets: generalization of arrival times for Shannon-encoded signals
- **algorithm**: systematic exploration of the Shannon / retiming design space
Modeling Shannon decompositions

unchanged

Shannon with $f$ as sel

Shannon with $g$ as sel

unchanged

Shannon with $h$ as sel

Shannon with $i$ as sel

unchanged

Shannon

start Shannon

stop Shannon

extend Shannon

......
Modeling Shannon decompositions
Exploring variants for $h$ ...
Exploring variants for $h$ ...

unchanged  Shannon  start Shannon  stop Shannon  extend Shannon
Exploring variants for $h$ ...
Exploring variants for $h$ ...
What combination is faster?

Problem: several combinations have “triple” outputs
All feasible arrival times for $h$

- (10,10,14)
- (16,15,11)
- (16,16,6)
- (16,16,7)
- (16,16,8)
- (17)
Pruning $fat(h)$ ...

(10,10,14)

(15)

(16) (15,15,11)

(17) (16,16,6) (16,16,7) (16,16,8)

(15) $\leq$ (16) : (16) can be safely ignored
Pruning $fat(h)$ ...

(10,10,14)

(15)

(16) (15,15,11)

(17) (16,16,6) (16,16,7) (16,16,8)

$(16,16,6) \preceq (16,16,7)$
Pruning \( \text{fat}(h) \) ...
We want to keep only the maximal elements
The pruned $\text{fat}(h)$ set

(10,10,14)

(15)

(15,16,11)

(16,16,6)

(16,16,7)

(16,16,8)
The fat sets

\[
fat(f) = \{(14),(13,13,11)\}
\]

\[
fat(g) = (8)(7,7,7)
\]

\[
fat(h) = \{(15),(10,10,14)\}
\]

\[
fat(i) = \{(15),(14,14,14)\}
\]

\[
fat(h) = \text{combine}(d(h), \{fat(f), \{(6)\}, fat(g)\})
\]
Best delay for combinational logic
Retiming limitation for one cycle

\[ D : \text{total loop combinational delay} \quad (2 + 2 + 1 + 1 + 2 = 8) \]
\[ R : \text{number of registers on the loop} \quad (1 + 3 = 4) \]

\[
\frac{D}{R} \leq c \\
D \leq c \cdot R \\
D + (-c) \cdot R \leq 0
\]

Assign weight \((-c)\) to registers:
Period \(c\) is feasible \(\iff\) the cycle has negative weight
The fundamental limit of retiming

For a given sequential network,

Period $c$ is feasible if $ALL^*$ cycles are negative.

$$f.l.\text{ret}(S) = \min c \text{ such that all cycles in } S \text{ are negative}$$

(*) including an artificial external arc between POs and PIs
Retiming and restructuring

Is A faster than B?

$\max_{\text{comb}} \text{delay}(A) < \max_{\text{comb}} \text{delay}(B) : \text{WRONG}$

$f.l.\text{ret}(A) < f.l.\text{ret}(B)$
The Bellman-Ford algorithm

Bellman-Ford detects positive cycles in polynomial time

\[
\begin{align*}
\text{Period } c \text{ is feasible} \\
\text{ALL cycles are negative} \\
\text{Bellman-Ford converges to a FIX POINT}
\end{align*}
\]

Generalization: Instead of arrival times (real numbers, e.g. 3) we use fat sets (e.g. \{(15), (10, 10, 14)\}).

- \(at(n) = d(n) + \max_{n' \in \text{fanins}(n)} at(n')\)
- \(fat(n) = \text{combine}(d(n), \{fat(n')\}_{n' \in \text{fanins}(n)})\)
Algorithm outline

procedure SeqShannon(S, c)
  (converges, fix_point_fat) = Bellman-Ford (S, c)
  if not converges then
    return NOT_FEASIBLE
  ShannonTransform(S, c, fix_point_fat)
  Retime(S)
  return SUCCESS

• we can approximate the best period c by binary search
desired period : 3
input arrival times: A=1 B=3 C=2
multiplexer delay : 1
output required time(s): D=3
Bellman-Ford : initialization
Bellman-Ford: starting relaxation
Bellman-Ford : relaxing ...
Bellman-Ford: relaxing ...
Bellman-Ford: relaxing ...
Bellman-Ford : relaxing ...
Bellman-Ford: relaxing ...
Bellman-Ford : fix point found
Shannon Transform

period=9
Retiming. Final solution

period=3
## ISCAS89 sequential benchmarks

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Reference period</th>
<th>Reference area</th>
<th>Retimed period</th>
<th>Retimed area</th>
<th>Sh. + ret. period</th>
<th>Sh. + ret. area</th>
<th>Time (s)</th>
<th>Speed up</th>
<th>Area penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>s510</td>
<td>8</td>
<td>184</td>
<td>8</td>
<td>184</td>
<td>8</td>
<td>184</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s641</td>
<td>11</td>
<td>115</td>
<td>11</td>
<td>115</td>
<td>9</td>
<td>122</td>
<td>1.1</td>
<td>22%</td>
<td>6%</td>
</tr>
<tr>
<td>s713</td>
<td>11</td>
<td>118</td>
<td>11</td>
<td>118</td>
<td>10</td>
<td>121</td>
<td>0.9</td>
<td>10%</td>
<td>3%</td>
</tr>
<tr>
<td>s820</td>
<td>7</td>
<td>206</td>
<td>7</td>
<td>206</td>
<td>7</td>
<td>206</td>
<td>0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s832</td>
<td>7</td>
<td>217</td>
<td>7</td>
<td>217</td>
<td>7</td>
<td>217</td>
<td>0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s838</td>
<td>10</td>
<td>154</td>
<td>10</td>
<td>154</td>
<td>8</td>
<td>162</td>
<td>2.6</td>
<td>25%</td>
<td>5%</td>
</tr>
<tr>
<td>s1196</td>
<td>9</td>
<td>365</td>
<td>9</td>
<td>365</td>
<td>9</td>
<td>365</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s1423</td>
<td>24</td>
<td>408</td>
<td>21</td>
<td>408</td>
<td>13</td>
<td>460</td>
<td>3.8</td>
<td>61%</td>
<td>12%</td>
</tr>
<tr>
<td>s1488</td>
<td>6</td>
<td>453</td>
<td>6</td>
<td>453</td>
<td>6</td>
<td>453</td>
<td>0.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s1494</td>
<td>6</td>
<td>456</td>
<td>6</td>
<td>456</td>
<td>6</td>
<td>456</td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s9234</td>
<td>11</td>
<td>662</td>
<td>8</td>
<td>656</td>
<td>8</td>
<td>684</td>
<td>6.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s13207</td>
<td>14</td>
<td>1382</td>
<td>11</td>
<td>1356</td>
<td>9</td>
<td>1416</td>
<td>18.0</td>
<td>22%</td>
<td>4%</td>
</tr>
<tr>
<td>s38417</td>
<td>14</td>
<td>7706</td>
<td>14</td>
<td>7652</td>
<td>13</td>
<td>7871</td>
<td>113</td>
<td>7%</td>
<td>3%</td>
</tr>
</tbody>
</table>
Contributions. Conclusions

- Theoretical model
  - a new way to describe complex combinations of Shannon decompositions
  - *fat* sets: generalized arrival times for "encoded" signals; extension of the Bellman-Ford algorithm

- Algorithm
  - Simultaneously analyses a large design space of Shannon decompositions and retiming.
  - Optimal performance; area heuristics
  - Promising results; short execution time