High level optimization by Retiming and Shannon decomposition

Cristian Soviani, Olivier Tardieu, Stephen A. Edwards

Department of Computer Science, Columbia University, 2005

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

Improves performance if $x_5$ is late compared to $x_1...x_4$
Retiming - review

Improves performance by re-distributing the registers
Motivation. Let’s 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)
Motivation sample after Shannon transform

Period: 9

Observations:

- the performance is worse
- the loop is much smaller
Motivation 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

- Shannon transform and retiming — review
- motivation sample
- proposed view of complex combinations of Shannon xforms — “feasible arrival times”
- critical cycles: the fundamental limit of retiming — review
- proposed algorithm: systematic exploration of the Shannon / retiming design space
  - simulation for a small sample
- experimental results and conclusions
Shannon decompositions seen as covering

unchanged

Shannon with \( \cdot \) as sel

Shannon with \( \cdot \) as sel

unchanged

Shannon

start Shannon

stop Shannon

extend Shannon

\( f \)

\( g \)

\( h \)

\( i \)
Shannon decompositions seen as covering
Exploring variants for $h$: “unchanged”
Exploring variants for $h$: “Shannon”
Exploring variants for $h$: “Start Shannon”
All possible “Feasible arrival times”

\[
\text{fat}(h) = \text{combine}(d(h), \{ \text{fat}(f), \{ (6) \}, \text{fat}(g) \}) \\
= \{ (15), (10, 10, 14) \}
\]
Solution
Retiming limitation for one cycle

\[ c \geq \frac{\text{delay}_{\text{cyc}}}{\text{regs}_{\text{cyc}}} \]
\[ \text{delay}_{\text{cyc}} \leq c \cdot \text{regs}_{\text{cyc}} \]
\[ (\text{delay}_{\text{cyc}} - c \cdot \text{regs}_{\text{cyc}}) \leq 0 \]

Assign weight \((-c)\) to registers:

Period \(c\) is feasible \(\Leftrightarrow\) the cycle has negative weight
Retiming limitation for all the circuit

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

Bellman-Ford detects positive cycles in polynomial time

$$\begin{cases} 
\text{Period } c \text{ is feasible} \\
\text{ALL cycles are negative} \\
\text{Bellman-Ford converges to a FIX POINT}
\end{cases}$$

Key: fix point equation (holds only if BF converges !!!)

$$fat(n) = \text{combine}(d(n), \{fat(n')\}_{n' \in \text{fanins}(n)})$$
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
Sequential sample - original

desired period : 3
input arrival times: A=1 B=3 C=2
multiplexer delay : 1
output required time(s): D=3
Bellman-Ford : initialization

```
(1) A
B (3)

2 2

(2) C

2 2

(-Inf) (-Inf)

(-Inf) (-Inf)

D
```

Diagram showing the initialization process with vertices A, B, C, and D, and edges with weights 2.
Bellman-Ford : starting relaxation
Bellman-Ford: relaxing ...

A → f → B (3)

B → g → C (2)

C → h → (Inf) (2)

D (1)
Bellman-Ford: relaxing ...
Bellman-Ford: relaxing ...
Bellman-Ford: relaxing...
Bellman-Ford : relaxing ...
Bellman-Ford: fix x point found

Diagram showing connections and labels for nodes A, B, C, and D. Edges and labels are marked with numbers and coordinates.
Shannon Transform
Retiming
Performance / area tradeoff — sanity check

Ripple carry adders $\rightarrow \log(n)$ delay carry select adders

add128. all above 120 points computed in 22s.
### ISCAS89 sequential benchmarks

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