Retiming and Re-synthesis

R. K. Brayton
University of California
Berkeley
Outline

- Retiming
- Retiming and Resynthesis (RnR)
- Resynthesis of Pipelines
Optimizing Sequential Circuits by Retiming Netlist of Gates

- Netlist of gates and registers:

- Various Goals:
  - Reduce clock cycle time
  - Reduce area
  - Reduce number of latches
Retiming

- **Problem**
  - Pure combinational optimization can be myopic since relations across register boundaries are disregarded

- **Solutions**
  - **Retiming**: Move register so that
    - clock cycle decreases, or number of registers decreases and
    - input-output behavior is preserved
  - **RnR**: Combine retiming with combinational optimization techniques so that can optimize across boundaries
Circuit Representation

- Leiserson, Rose and Saxe (1983)
- Circuit representation: \( G(V,E,d,w) \)
  - \( V \leftrightarrow \) set of gates
  - \( E \leftrightarrow \) set of wires
  - \( d(v) = \) delay of gate/vertex \( v \), \( (d(v) \geq 0) \)
  - \( w(e) = \) number of registers on edge \( e \), \( (w(e) \geq 0) \)
**Circuit Representation**

Example: Correlator

\[ \delta(x, y) = 1 \text{ if } x=y \]
\[ 0 \text{ otherwise} \]

- Every cycle in Graph has at least one register i.e. no combinational loops.

<table>
<thead>
<tr>
<th>Operation</th>
<th>delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \delta )</td>
<td>3</td>
</tr>
<tr>
<td>+</td>
<td>7</td>
</tr>
</tbody>
</table>
preliminaries

• For a path $p$: $V_0 \rightarrow$
  
  $d(p) = \sum_{e \in p} d(e)$  \quad (includes endpoints)

  $w(p) = \sum_{e \in p} w(e)$

• Clock cycle

\[
c = \max_{p : w(p)=0} \{d(p)\}
\]

For correlator $c = 13$

Path with $w(p)=0$
**Basic Operation**

- **Movement of registers from input to output of a gate or vice versa**

- **Does not affect gate functionalities**

- **A mathematical definition: Retardation**
  - \( r: V \rightarrow Z \), an integer vertex labeling
  - \( Wr(e) = w(e) + r(v) - r(u) \) for edge \( e = (u,v) \)
Basic Operation

- Thus in the example, \( r(u) = -1, r(v) = -1 \) results in

- For a path \( p: s \rightarrow t \), \( W_r(p) = w(p) + r(t) - r(s) \)
- A mathematical definition: Retardation
  - \( r: V \rightarrow \mathbb{Z} \), an integer vertex labeling
  - \( W_r(e) = w(e) + r(v) - r(u) \) for edge \( e = (u, v) \)
  - A retiming \( r \) is legal if \( W_r(e) \geq 0, \forall e \in E \)
Retiming for minimum clock cycle

- Problem Statement: (Minimum cycle time)
- Given $G(V, E, d, w)$, find a Legal retiming $r$ so that

$$c = \max_{p\in V} \{d(p)\}$$

is minimized

- Retiming: 2 important matrices
  - Register weight matrix
    $$W(u, v) = \min \{w(p) : u \rightarrow^p v\}$$
  - Delay matrix
    $$D(u, v) = \max \{d(p) : u \rightarrow^p v, w(p) = w(u, v)\}$$
Retiming for minimum clock cycle

\[ \begin{align*}
W & \begin{array}{cccc}
V0 & V1 & V2 & V3 \\
V0 & 0 & 2 & 2 & 2 \\
V1 & 0 & 0 & 0 & 0 \\
V2 & 0 & 2 & 0 & 0 \\
V3 & 0 & 2 & 2 & 0 \\
\end{array} \\
D & \begin{array}{cccc}
V0 & V1 & V2 & V3 \\
V0 & 0 & 3 & 6 & 13 \\
V1 & 13 & 3 & 6 & 13 \\
V2 & 10 & 13 & 3 & 10 \\
V3 & 7 & 10 & 13 & 7 \\
\end{array} \\
\end{align*} \]

\[ C \leq \alpha \iff \forall p, \text{ if } d(p) > \alpha \text{ then } w(p) \geq 1 \]
Conditions for Retiming

- Assume that we are asked to check if a retiming exists for a clock cycle $\alpha$
- Legal retiming: $w_r(e) \geq 0$ for all $e$. Hence
  \[ w_r(e) = w(e) = r(v) - r(u) \geq 0 \text{ or} \]
  \[ r(u) - r(v) \leq w(e) \]
- For all paths $p: u \rightarrow v$ such that $d(p) \geq \alpha$, we require $w_r(p) \geq 1$
  - Thus
    \[
    1 \leq w_r(p) = \sum_{e \in p} w_r(e) = \sum_{e \in p} [w(e) + r(u) - r(v)] = w(p) + r(u) - r(v) = w(p) + r(v) - r(u).
    \]

Or take the least $w(p)$ (tightest constraint) $r(u)-r(v) \leq W(u,v)-1$

Note: this is independent of the path from $u$ to $v$, so we just need to apply it to $u$, $v$ such that $D(u,v) > \alpha$
Solving the constraints

- All constraints in difference of 2 variable form
- Related to shortest path problem

Correlator: $\alpha = 7$

Legal: $r(u) - r(v) \leq w(e)$

$D > 7$: $r(u) - r(v) \leq W(u, v) - 1$

$W$

<table>
<thead>
<tr>
<th></th>
<th>V0</th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>V1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>V3</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

$D$

<table>
<thead>
<tr>
<th></th>
<th>V0</th>
<th>V1</th>
<th>V2</th>
<th>V3</th>
</tr>
</thead>
<tbody>
<tr>
<td>V0</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>V1</td>
<td>13</td>
<td>3</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>V2</td>
<td>10</td>
<td>13</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>V3</td>
<td>7</td>
<td>10</td>
<td>13</td>
<td>7</td>
</tr>
</tbody>
</table>
Solving the constraints

- Do shortest path on constraint graph: \( O(|V|^3) \).
- A solution exists if and only if there exists no negative weighted cycle.

\[
\begin{align*}
\text{Legal: } r(u) - r(v) &\leq w(e) \\
\text{D>7: } r(u) - r(v) &\leq W(u,v) - 1
\end{align*}
\]

A solution is \( r(v_0) = r(v_3) = 0, \ r(v_1) = r(v_2) = -1 \)
# Retiming

To find the minimum cycle time, do a binary search among the entries of the D matrix $O(\sqrt{V \log V})$

$$
\begin{array}{c}
\text{V0} \\
\text{V1} \\
\text{V2} \\
\end{array}
\begin{array}{c}
\text{2} \\
\text{0} \\
\text{0} \\
\end{array}
\begin{array}{c}
\text{v0} \\
\text{v1} \\
\text{v2} \\
\end{array}
\begin{array}{c}
\text{2} \\
\text{0} \\
\text{0} \\
\end{array}
\begin{array}{c}
\text{W} \\
\text{D} \\
\end{array}
\begin{array}{cccc}
\text{V0} & \text{V1} & \text{V2} & \text{V3} \\
0 & 2 & 2 & 2 \\
0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 \\
0 & 2 & 2 & 0 \\
\end{array}
\begin{array}{cccc}
\text{V0} & \text{V1} & \text{V2} & \text{V3} \\
0 & 3 & 6 & 13 \\
13 & 3 & 6 & 13 \\
10 & 13 & 3 & 10 \\
7 & 10 & 13 & 7 \\
\end{array}
$$

Retimed correlator:

Clock cycle $= 3 + 3 + 7 = 13$

Clock cycle $= 7$
Retiming: 2 more algorithms

- Relaxation based: Repeatedly find critical path; retime vertex at end of path by +1
  \[ O(|V| |E| \log |V|) \]

- Mixed Integer Linear Program formulation
Retiming for minimum state

Goal: minimize number of registers used

\[ N_v = \sum_{e \in E} v(e) \]
\[ = \sum_{e \in E} (w(e) + r(v) - r(u)) \]
\[ = \sum_{e \in E} w(e) + \sum_{e \in E} (r(v) - r(u)) \]
\[ = N + \sum_{e \in E} (r(v) - r(u)) \]
\[ = N + \sum_{e \in E} (r(v)(\#fanin(v) - \#fanout(v))) \]
\[ = N + \sum_{e \in E} a_v r(v) \]

Where \( a_v \) is a constant.
**Minimum registers - formulation**

- Minimize:
  \[
  \sum_{e \in V} d_{e} \cdot r(v)
  \]

  Subject to \( W_r(e) = w(e) + r(v) - r(u) \geq 0 \)

- Reducible to a flow problem
- Care to be taken when registers fanout to registers
- Handle multi-way forks with care
Retiming and resynthesis: motivation

- Goal: incorporate cominational optimazation into sequential optimization
- Naïve approach: carve out combinational regions, do optimization on each region Only local gains made.
- Can we do any better?

RnR: a new approach

- Sentovich, Malik, Brayton andSangiovanni-Vincentelli (89)
- 3 steps approach
  - 1. Move registers to boundary of circuit
  - 2. Optimize network
  - 3. Move registers back in an optimal way
**RnR: circuit representation**

- Circuit representation: communication graph
  - internal/peripheral edges
  - edge-weight = register count
Extended Retiming

- Move register to the periphery
- Negative edge-weights permitted

A negative latch has the interpretation that it advances its input by 1 clock cycle instead of delaying it by one clock cycle.
Peripheral Retiming

- A retiming is called a peripheral retiming if it results in all internal edges having zero weight
- Peripheral edges can have negative weight
Peripheral Retiming

• A circuit that undergoes peripheral retiming followed by a legal retiming, i.e. one that results in all weights \( \geq 0 \) is functionally equivalent to the original circuit

• Functional equivalence: equivalence of finite state automata (but we have to be careful about the initial conditions and initializing sequences. The resulting circuit may only exhibit equivalence after an appropriate delay. See Singhal et.al ICCAD 1995. This holds even for regular retiming)
Peripheral Retiming - an example

Not possible for all circuits:
**Path Weight Matrix (PWM)**

- **Matrix W**
  - row: inputs
  - column: outputs

\[
W_{ij} = \begin{cases} 
1 & \text{if paths with different weights} \\
0 & \text{if same weight for all paths}
\end{cases}
\]

\[
W = \begin{pmatrix}
1 & * \\
0 & \sim \\
* & 0
\end{pmatrix}
\]
**PWM and Peripheral Retiming**

- **Satisfiable path weight matrix:**
  - $W_{ij} \neq \sim, \forall i, \forall j$ and
  - $\exists \alpha_i, \beta_j$, such that $W_{ij} = \alpha_i + \beta_j$, $\forall W_{ij} \neq ^*$

- **Peripheral retiming possible $\iff$ matrix is satisfiable**
  - $\alpha_i, \beta_j$ specify registers on peripheral edge
  - some $\alpha_i, \beta_j$ can be negative

- **Complexity:** linear in size of communication graph
Path Weight Matrix - generation

- DFS skeleton
  - generate one output column of $W$ at a time
- Example:

\[
\begin{align*}
\begin{bmatrix} 1 & 0 & * \end{bmatrix} & \circ_1 \\
\begin{bmatrix} 1 & 0 & * \end{bmatrix} & \circ_2
\end{align*}
\]

\[
\begin{align*}
[0^{**}] & \quad 1 & [0^{*}] \\
[0^{*}] & \quad 1 & [0^{*}] \\[1^{**}] & \quad 1 & [0^{*}]
\end{align*}
\]

\[
\begin{align*}
([0^{**}]+1) & \& ([0^{*}]+0) \\
= [1^{**}] & \& [0^{*}] \\
= [1^{0*}]
\end{align*}
\]

\[
\begin{align*}
([0^{**}]+0) & \& ([0^{*}]+0) \\
= [0^{*}] & \& [0^{*}] \\
= [0^{00}]
\end{align*}
\]

\[
\begin{align*}
([0^{*}]+1) & \& ([0^{*}]+0) \\
= [1^{*}] & \& [0^{*}] \\
= [1^{0*}] \\
= [0^{*}]
\end{align*}
\]

\[
\begin{align*}
([0^{*}]+0) & \& ([0^{*}]+0) \\
= [0^{*}] & \& [0^{*}] \\
= [0^{00}]
\end{align*}
\]
Computing $\alpha, \beta$

- Each constraint of the form $\alpha_i + \beta_j = W_{ij}$
- Procedure
  - set $\alpha_1 = 0$
  - use first row to generate $\beta$'s
  - determine $\alpha$'s from $\beta$'s
  - check for consistency
- Example:
Computing \( \alpha, \beta \)

- Example:

\[
\begin{array}{c|cc}
\text{i} & 0 & 0 \\
\text{j} & 0 & 1 \\
\end{array}
\]

\[
\begin{array}{c|cc}
\text{o1} & 0 & 0 \\
\text{o2} & 0 & 1 \\
\end{array}
\]

\[
\begin{align*}
\alpha_1 + \beta_1 &= 0 \\
\alpha_1 + \beta_2 &= 0 \\
\alpha_2 + \beta_1 &= 0 \\
\alpha_2 + \beta_2 &= 1 \\
\beta_1 - \beta_2 &= 0 \\
\beta_1 - \beta_2 &= -1 \\
\end{align*}
\]
Optimizing Acyclic Sequential Circuits

- Acyclic circuits with satisfiable W
  - Do peripheral retiming putting $\alpha_i, \beta_j$ registers at the I/O
  - Resynthesize interior
  - Do a legal retiming (move registers in)
    - May not always be possible

- Acyclic circuits with unsatisfiable W
  - Identify maximal sub-circuits with satisfiable W
  - Cut connections
  - Repeat previous procedure
Acyclic Sequential Circuits

- Note that both of the cut circuits can be peripherally retimed, whereas the first one cannot
Optimizing Cyclic Sequential Circuits

- Cyclic circuits
  - Make circuit acyclic by breaking cycles
  - Different feedback-cuts give different W's
  - Find sub-circuits with satisfiable W's
  - Repeat procedure (1)
FSM Optimization

Break feedback
FSM Optimization

Peripheral retiming

Resynthesize
FSM Optimization

Retime

Reconnect
FSM Optimization

Original circuit

Resynthesized circuit
Resynthesis of Pipelines

- **Goal:** Performance optimization of pipeline circuits
- **Example:** Pipeline circuit $C$

![Diagram showing pipeline circuit and retiming process]
Resynthesis of Pipelines

- **Parameters:**
  - $A$ vector of arrival times for inputs
  - $R$ vector of required times for outputs
  - $c$ target clock cycle
  - $C$ circuit

- **Pipeline performance problem:**
  \[ P_p(C_p, c_p, A_p, R_p) \]

- **Combinational performance problem:**
  \[ P_c(C_c, A_c, R_c) \]
Resynthesis of Pipelines

• Problem Transformation:
  - Relax $P_p(C_p,c_p,A_p,R_p)$ to $P'_{p}(C_p,c+\delta,A_p,R_p)$
    • where $\delta$ = largest possible single gate delay
  - Convert: pipeline $P'_{p}$ to combinational problem $P'_{c}$

\[
\begin{align*}
C_p & \xrightarrow{\text{peripheral retiming}} C_c \\
R_{C_{\text{stage}}} & = (i-1) \times c + R_{p_{\text{stage}}} \\
A_{C_{\text{stage}}} & = (i-1) \times c + A_{p_{\text{stage}}}
\end{align*}
\]
Performance Synthesis of Pipeline

- Arrival and Required Times for $P^c_i$:

- Theoretical Contribution:
  - If $P^c_i$ has a solution then retiming yields a solution to $P^p_i$
  - If there is a solution to $P_p$ then peripheral retiming yields a solution to $P^c_i$