EENG 426/CPSC 459/ENAS 876
Silicon Compilation

Bubble reshuffling

Computer Systems Lab

http://csl.yale.edu/~rajit

Fall 2018

Bubble reshuffling

The process of converting a production rule set into one that is directly implementable in CMOS is called bubble reshuffling.

- pull-up must only have inverted variables
- pull-down must only have non-inverted variables

Warning: there are production rule sets which cannot be bubble reshuffled! In this case, we have to change the handshaking expansion.

Stability: intuition

We can define a relation that specifies when signal transitions are ordered.

For example:

$$\langle x\uparrow, 0\rangle \prec \langle y\uparrow, 0\rangle \prec \langle x\downarrow, 0\rangle \prec \langle y\downarrow, 0\rangle \prec \langle x\uparrow, 1\rangle$$

There might be concurrent transitions too; the relation only holds between transitions that are truly ordered.

Stability: intuition

What makes a PRS stable?

- every transition is "acknowledged"
- feedback from the acknowledgment of transition to the inputs of the gate

Consider a production rule $G \mapsto t$.

Suppose G becomes **true**.

Eventually the G must become **false** if the opposing transition fires (non-interference).

Stability: intuition

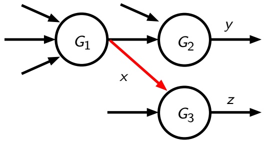
The transition that causes G to become **false** cannot be enabled in the state in which $G \wedge \neg R(t)$ holds (stability).

$\Rightarrow$ there is an intervening transition that changes the state. This transition is said to acknowledge t.

A transition can only be acknowledged by another transition!

Isochronic branches and forks

Consider three operators connected as shown below.



Suppose $x\uparrow$ is acknowledged by $y\downarrow$ and $z\downarrow$, but $x\downarrow$ is only acknowledged by $y\uparrow$.

The connection from x to the input of the operator for z is said to be an isochronic branch, and the fork from x to the inputs of y and z is called an isochronic fork.

Bubble reshuffling

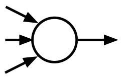
Basic rule:
Inverters can only be placed on non-isochronic branches of a fork.

Otherwise, the output of the inverter will not be acknowledged, and the inverter will be unstable.

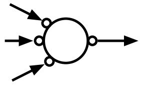
Bubble reshuffling: transformation 1

Basic transformations:

1. Invert sense of a gate
   $\Rightarrow$ invert senses of all inputs



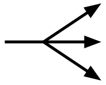
is replaced by:



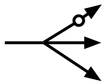
Bubble reshuffling: transformation 2

Basic transformations:

2. Add inverter to a non-isochronic branch



is replaced by:



Example

Production rule set for a buffer reshuffling:

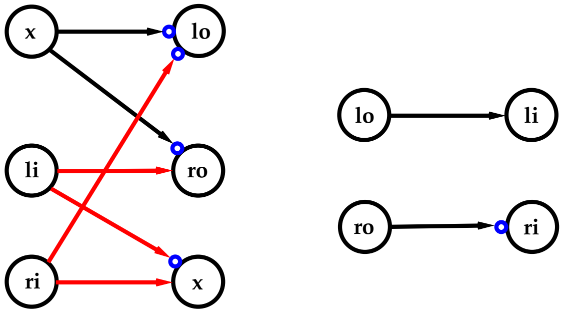
$$\begin{aligned} x &\mapsto lo\uparrow \\ \neg x \wedge \neg ri &\mapsto lo\downarrow \end{aligned}$$

$$\begin{aligned} li &\mapsto x\uparrow \\ ri &\mapsto x\downarrow \end{aligned}$$

$$\begin{aligned} x \wedge \neg li &\mapsto ro\uparrow \\ \neg x &\mapsto ro\downarrow \end{aligned}$$

Example

We can solve this problem graphically.



Example

Observations:

Transformation 1 for bubble reshuffling preserves the parity of the number of bubbles on any cycle. (Proof?)

Bubbles can only remain on non-isochronic branches.

⇒ attempt to move all bubbles from isochronic branches onto non-isochronic branches, possibly eliminating bubbles while doing so.

Example

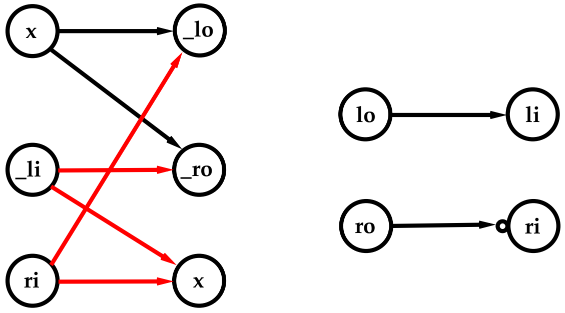
Note:
We cannot move all bubbles to non-isochronic branches when there is a cycle of isochronic branches with an odd number of bubbles on it.

(Why?)

Is this true for our example?

Example

Bubble reshuffling the example:



Example

Production rules:

$$
\begin{aligned}
x &\mapsto \_lo\downarrow \\
\neg x \wedge \neg ri &\mapsto \_lo\uparrow
\end{aligned}
$$

$$
\begin{aligned}
\neg\_li &\mapsto x\uparrow \\
ri &\mapsto x\downarrow
\end{aligned}
$$

$$
\begin{aligned}
x \wedge \_li &\mapsto \_ro\downarrow \\
\neg x &\mapsto \_ro\uparrow
\end{aligned}
$$