Delay Insensitive Circuits

an exercise in formal methods

Dennis Furey

(cancelled) BCS FACS Seminar, 28 May 2020

Plumstead Publishing House

Overview

Delay insensitive circuits are understandable by:

(a) intuitive unconvincing hand-wavy descriptions
(b) formal theories checkable by programs or proofs

Let’s try to find a way from (a) to (b).

The intuitive approach

Executive summary

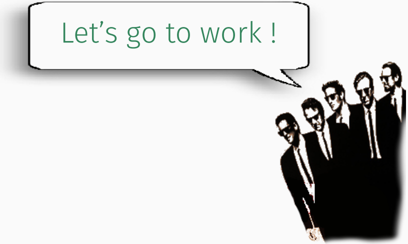
All you need to know about delay insensitive circuits:

- they have no clocks
- handshake signals and causal relationships make them go
- theory taught in school doesn't work on them
- they are feared, reviled, ignored, and misunderstood

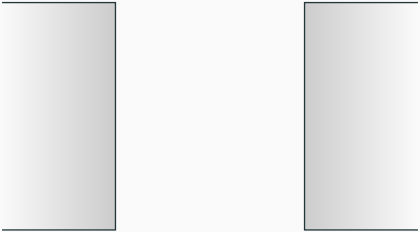
Executive summary

All you need to know about delay insensitive circuits:

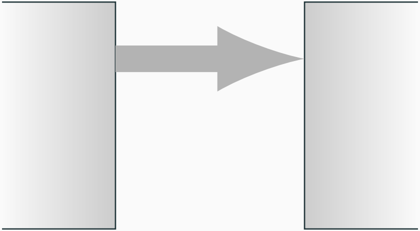
- they have no clocks
- handshake signals and causal relationships make them go
- theory taught in school doesn't work on them
- they are feared, reviled, ignored, and misunderstood



Synchronization by handshakes

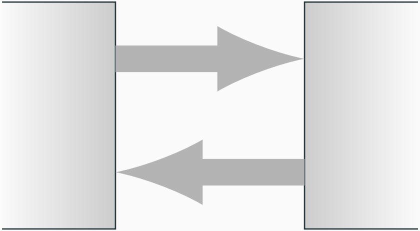


Synchronization by handshakes



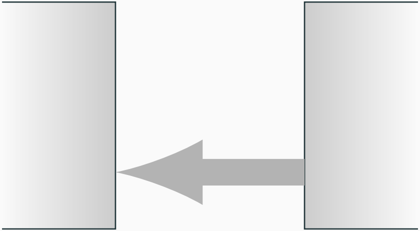
- request

Synchronization by handshakes



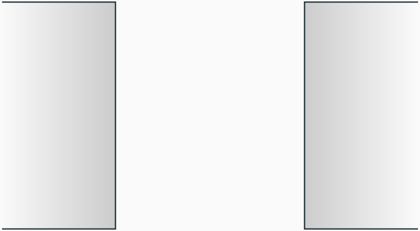
- request
- acknowledge

Synchronization by handshakes



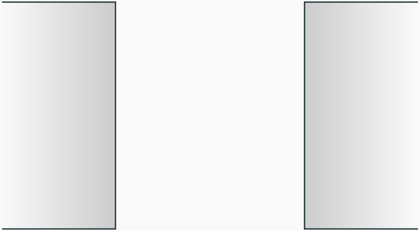
- request
- acknowledge
- release

Synchronization by handshakes



- request
- acknowledge
- release
- unacknowledge

Synchronization by handshakes



- request
- acknowledge
- release
- unacknowledge

} unnecessary ? (release, unacknowledge)

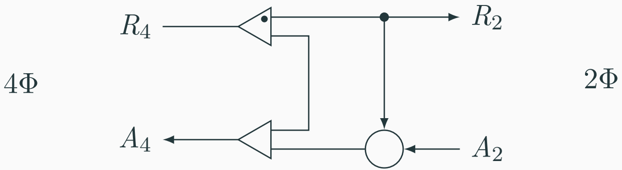
Synchronization by handshakes

- request
- acknowledge
- release
- unacknowledge

(release, unacknowledge) } unnecessary ?

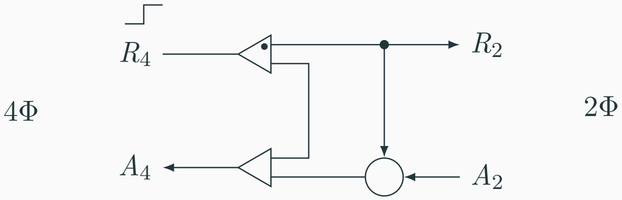
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.

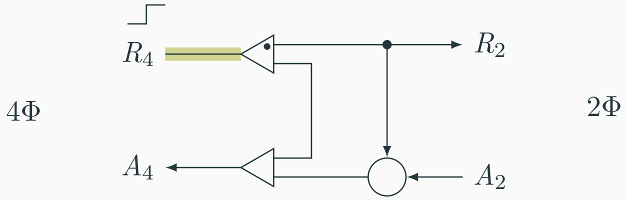


4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.

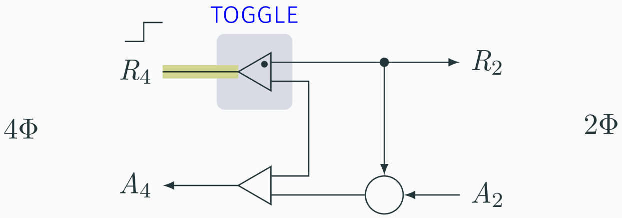


The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



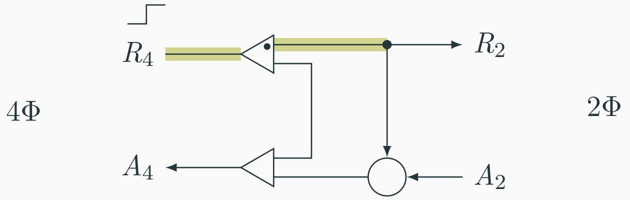
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



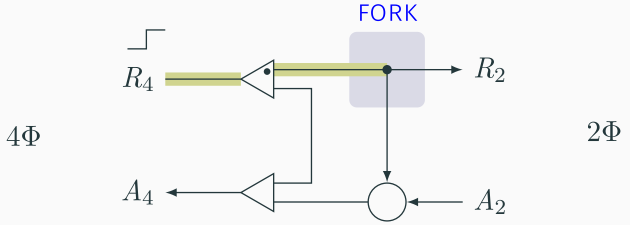
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



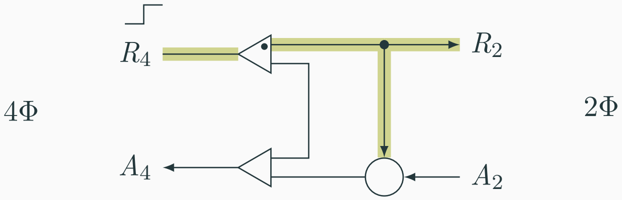
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



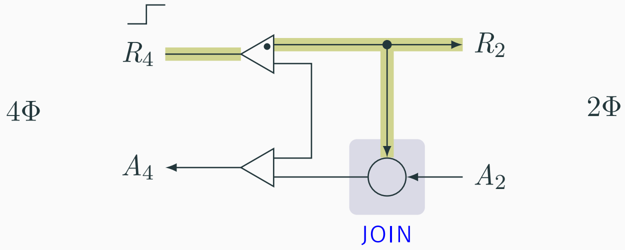
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



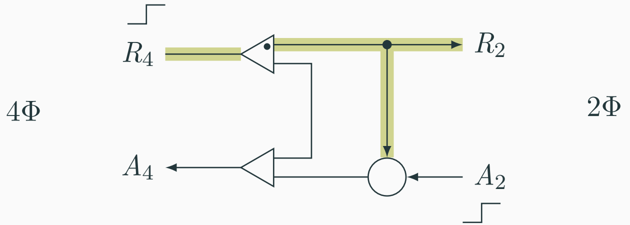
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



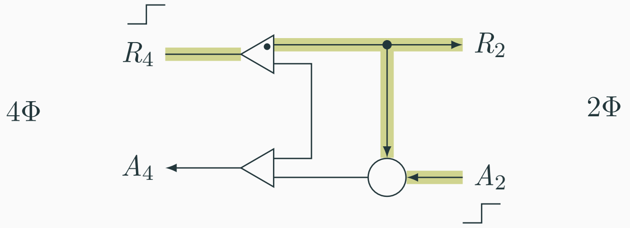
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



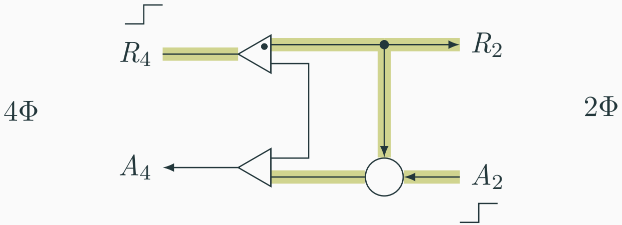
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



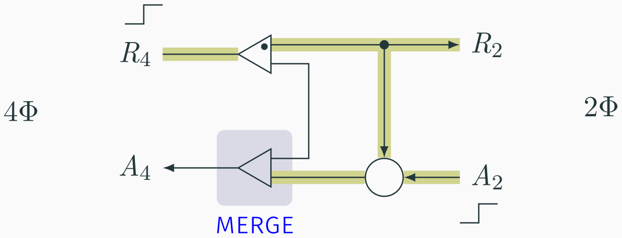
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



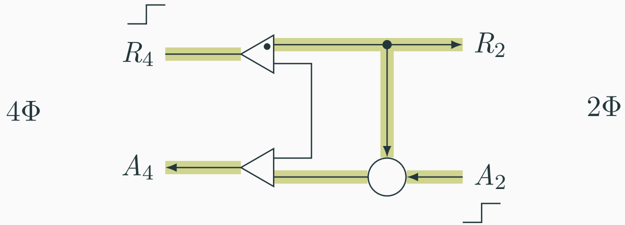
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



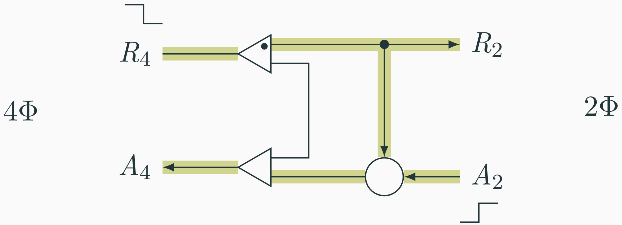
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



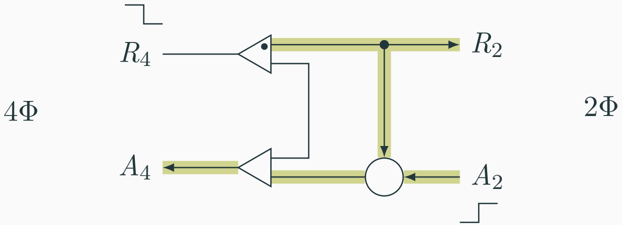
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



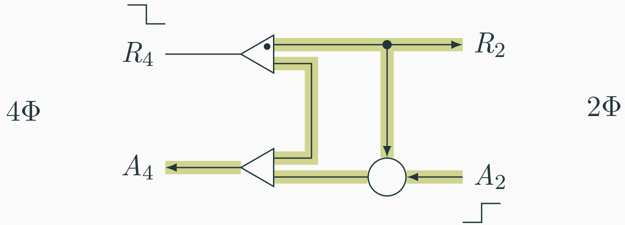
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



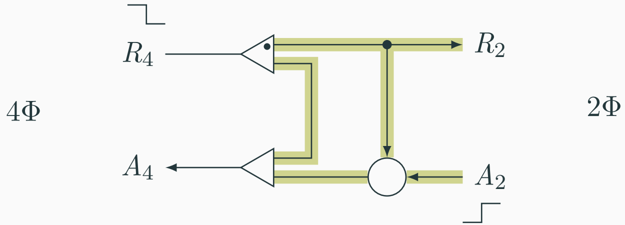
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



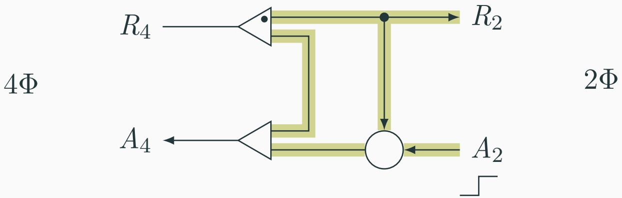
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



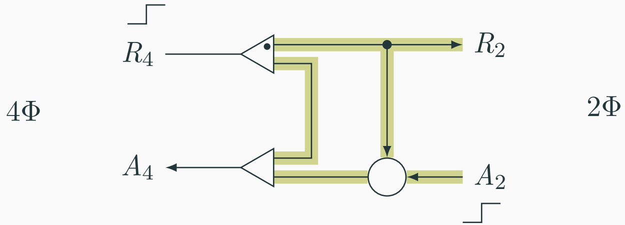
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



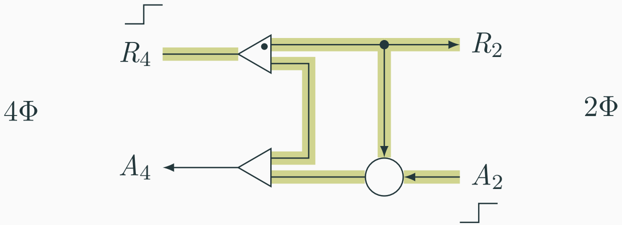
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



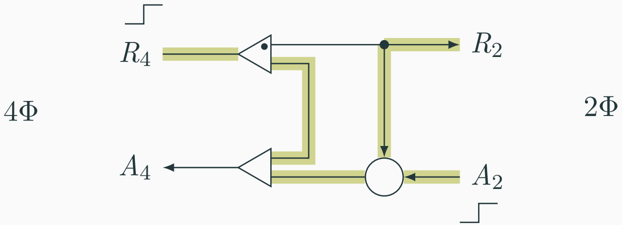
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



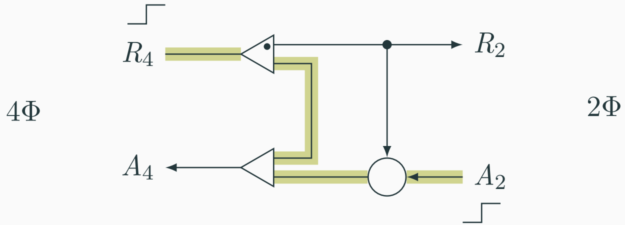
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



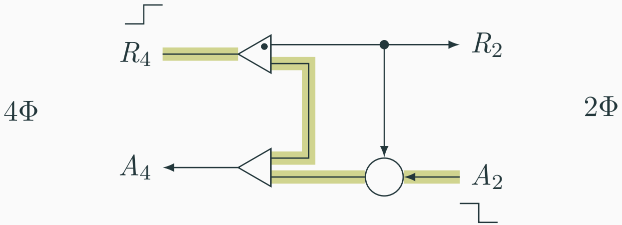
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



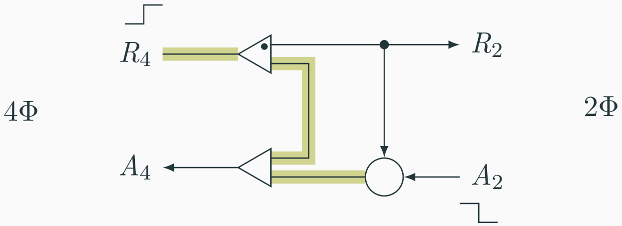
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



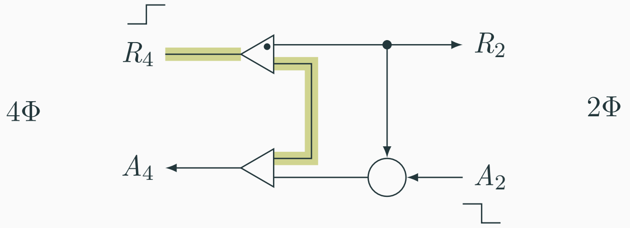
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



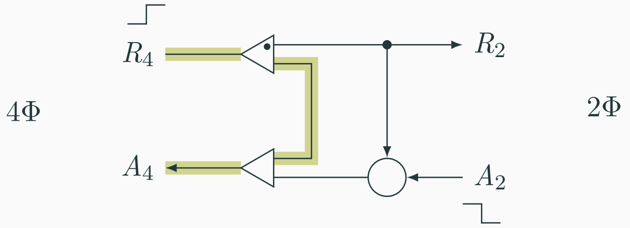
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



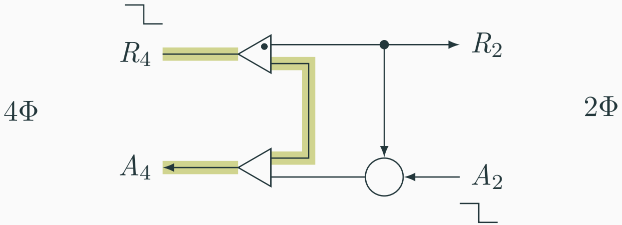
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



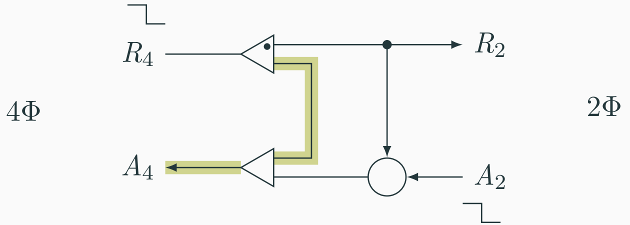
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



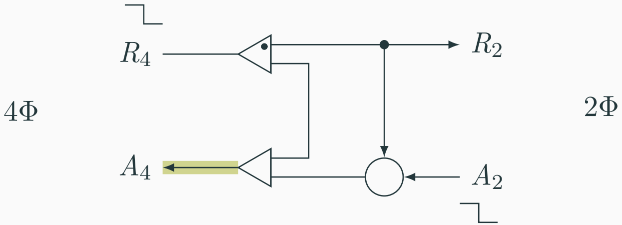
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



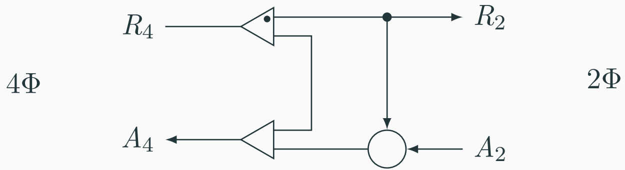
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



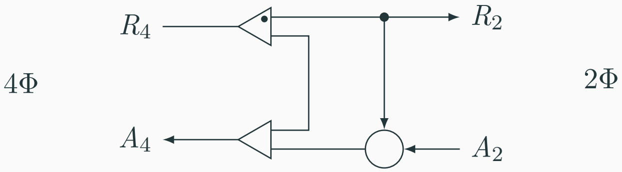
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



4Φ to 2Φ handshake converter

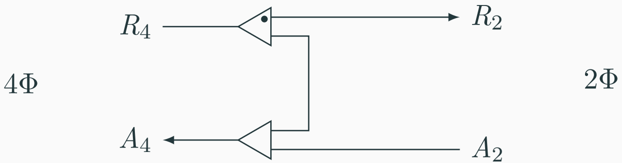
The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.



equivalent without the FORK and JOIN ?

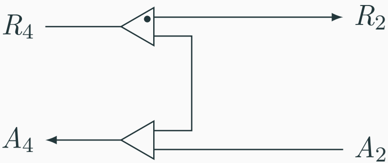
4Φ to 2Φ handshake converter

The active side (left) performs a 4-phase handshake, but the passive side (right) performs a 2-phase handshake.

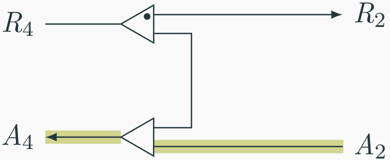


equivalent without the FORK and JOIN ?

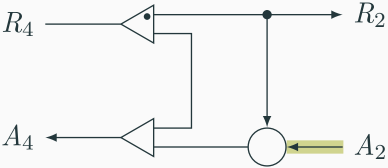
4Φ to 2Φ handshake converter



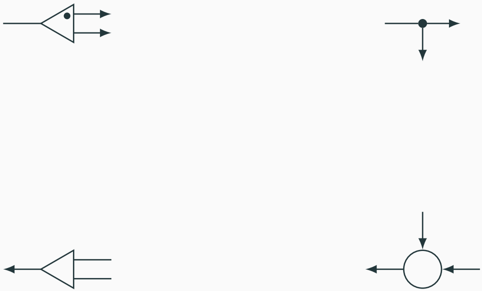
4Φ to 2Φ handshake converter



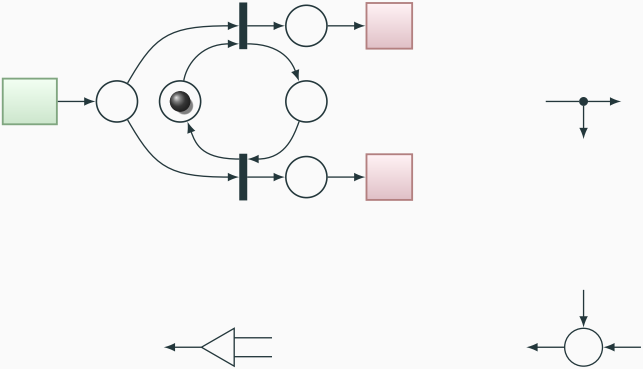
4Φ to 2Φ handshake converter



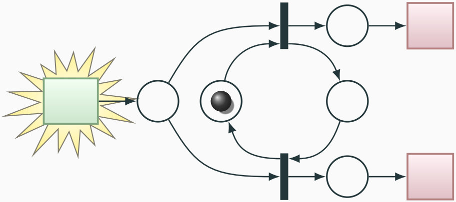
4Φ to 2Φ handshake converter

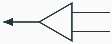


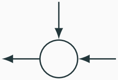
4Φ to 2Φ handshake converter



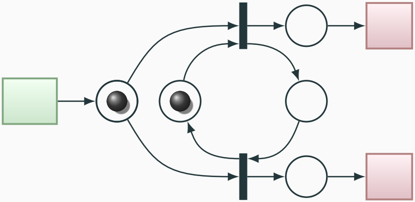
4Φ to 2Φ handshake converter

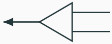


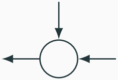




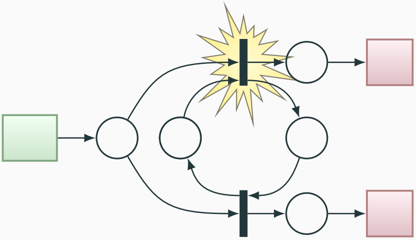
4Φ to 2Φ handshake converter

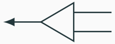


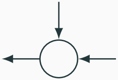




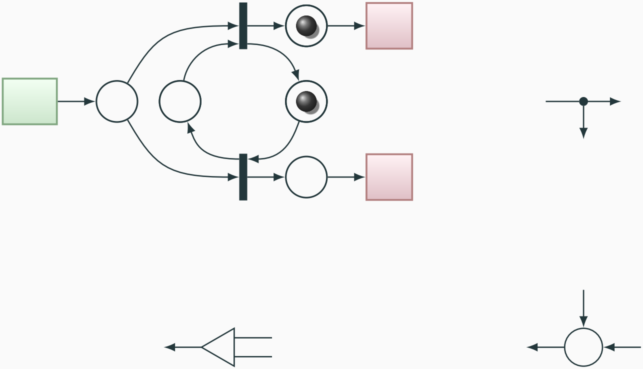
4Φ to 2Φ handshake converter



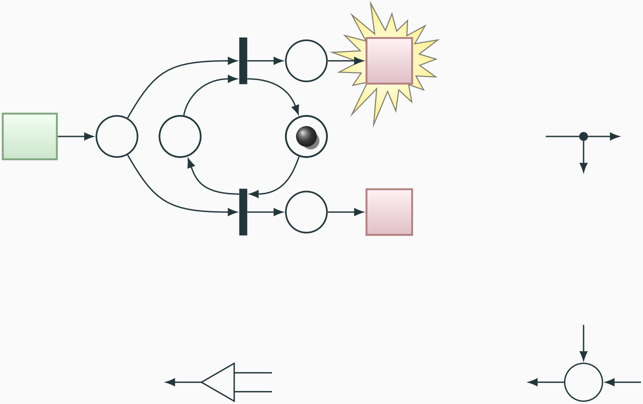




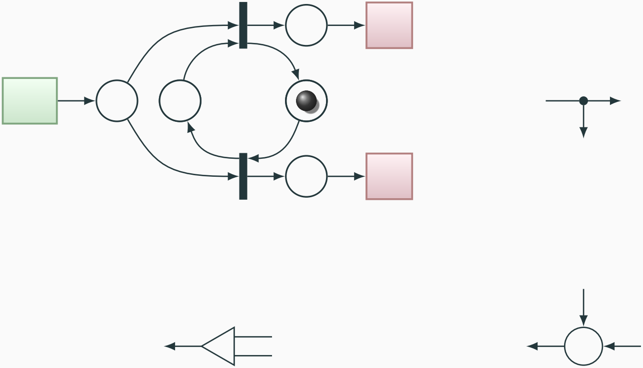
4Φ to 2Φ handshake converter



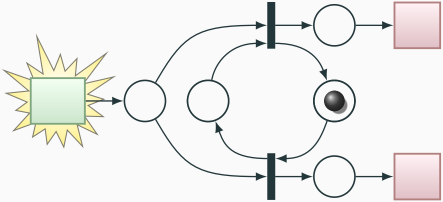
4Φ to 2Φ handshake converter

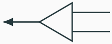


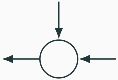
4Φ to 2Φ handshake converter



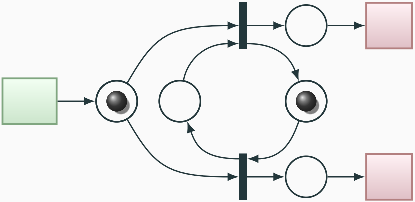
4Φ to 2Φ handshake converter

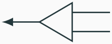


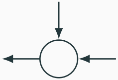




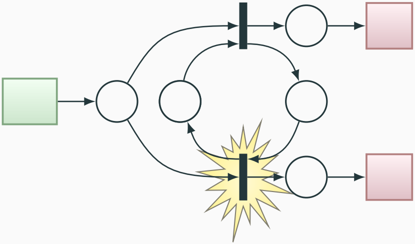
4Φ to 2Φ handshake converter

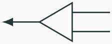


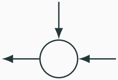




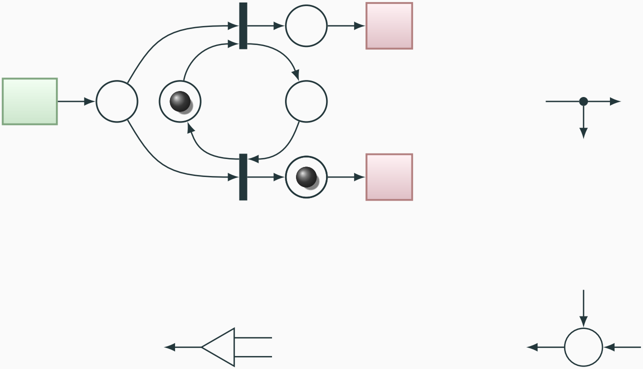
4Φ to 2Φ handshake converter



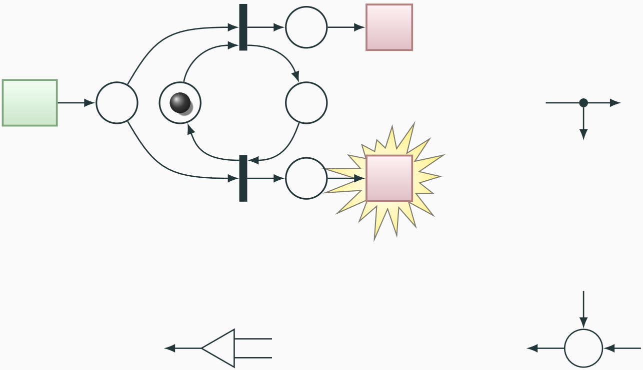




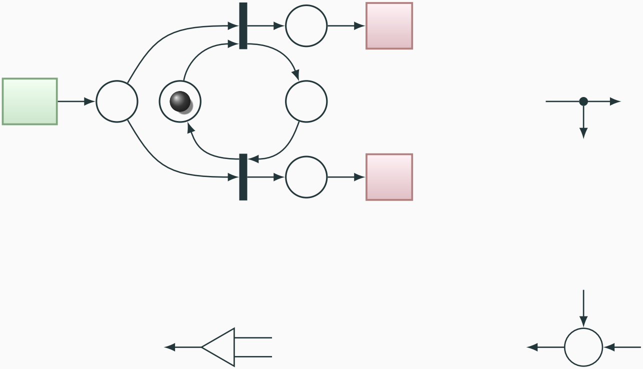
4Φ to 2Φ handshake converter



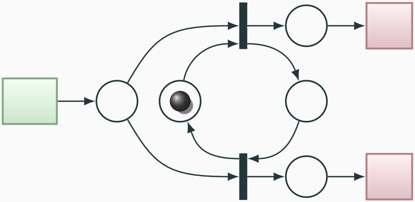
4Φ to 2Φ handshake converter

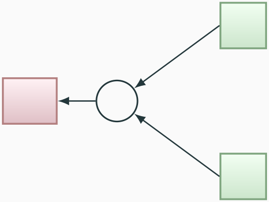


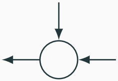
4Φ to 2Φ handshake converter



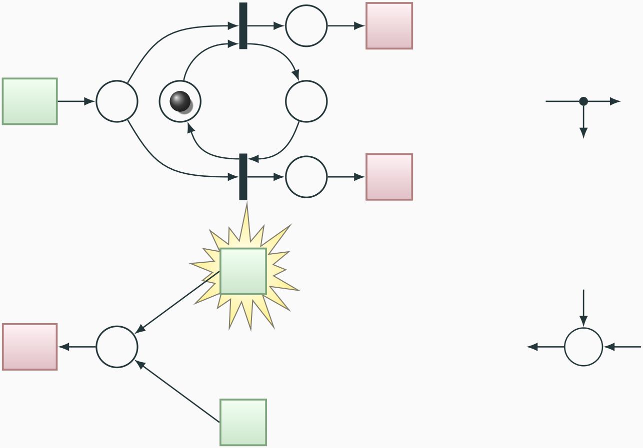
4Φ to 2Φ handshake converter



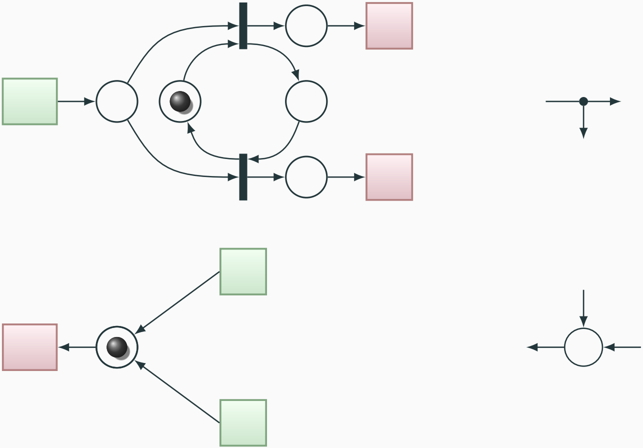




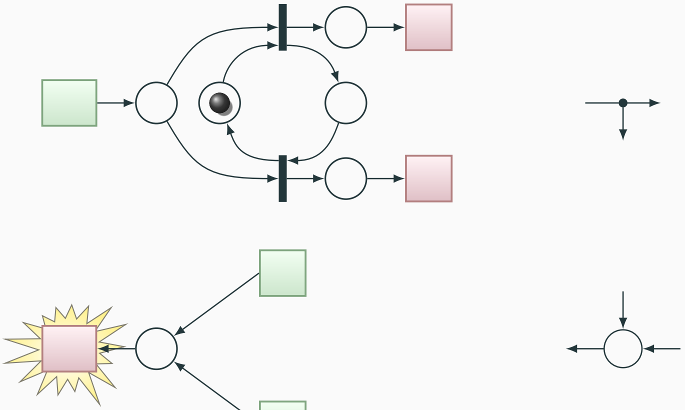
4Φ to 2Φ handshake converter



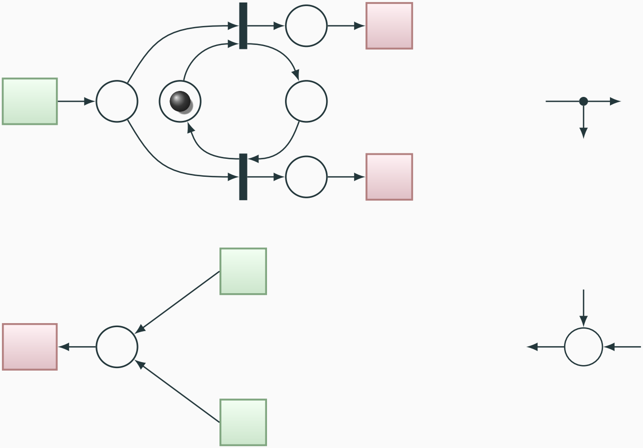
4Φ to 2Φ handshake converter



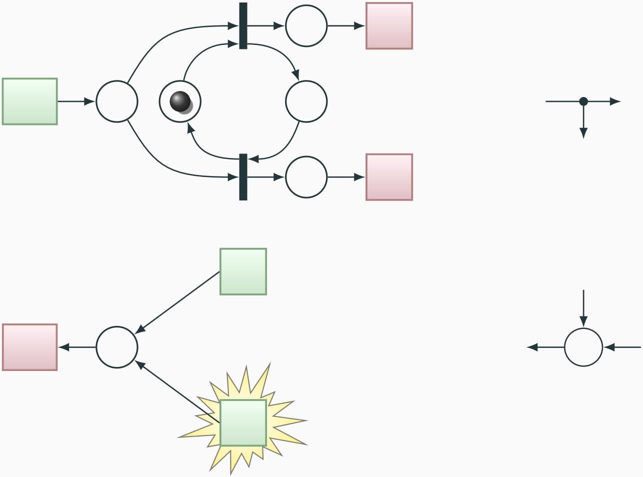
4Φ to 2Φ handshake converter



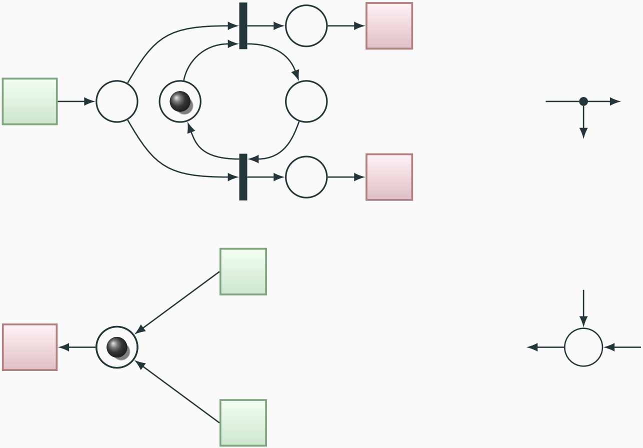
4Φ to 2Φ handshake converter



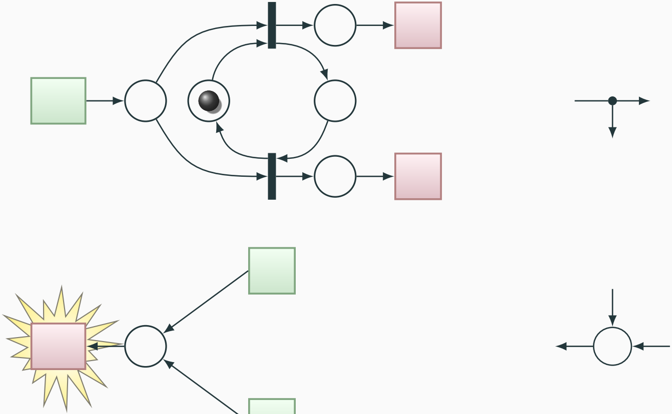
4Φ to 2Φ handshake converter



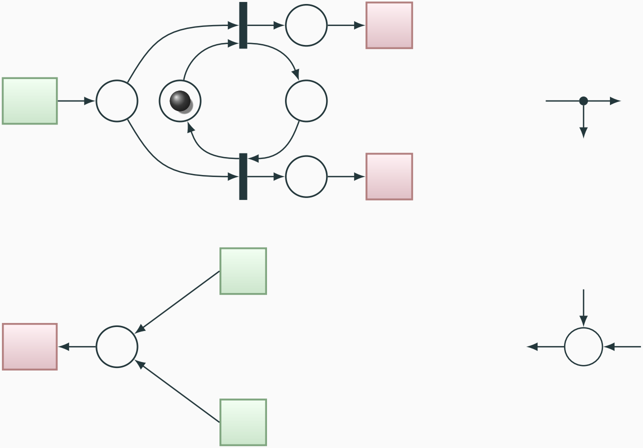
4Φ to 2Φ handshake converter



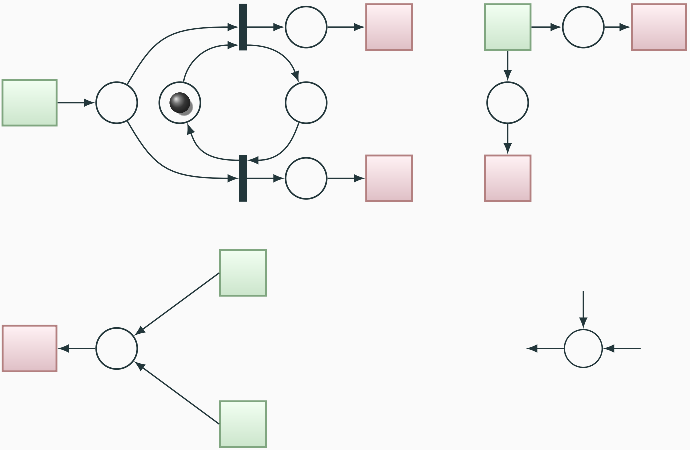
4Φ to 2Φ handshake converter



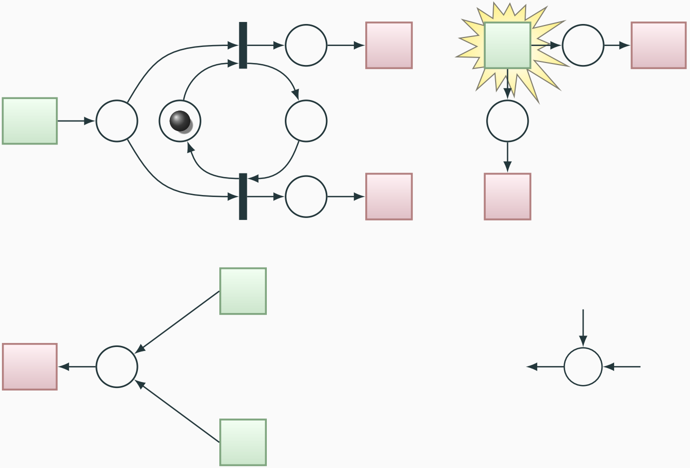
4Φ to 2Φ handshake converter



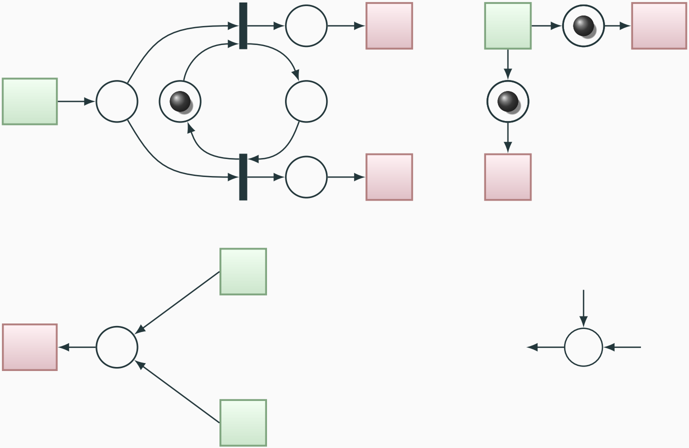
4Φ to 2Φ handshake converter



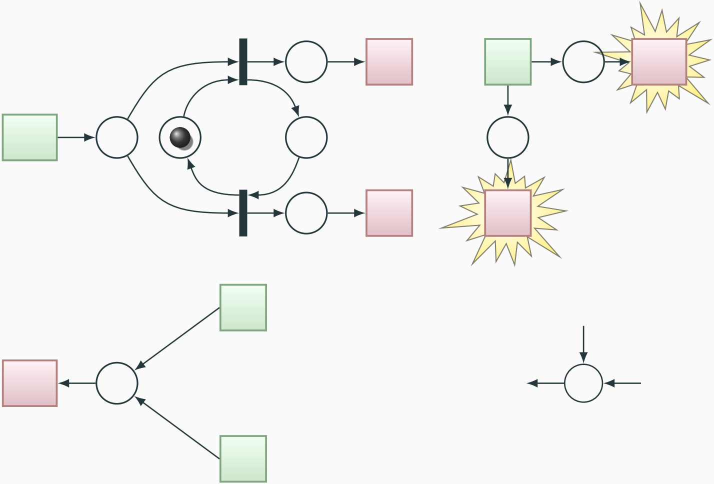
4Φ to 2Φ handshake converter



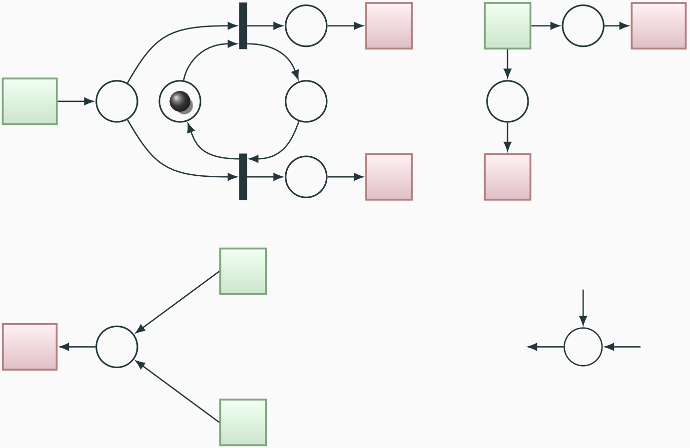
4Φ to 2Φ handshake converter



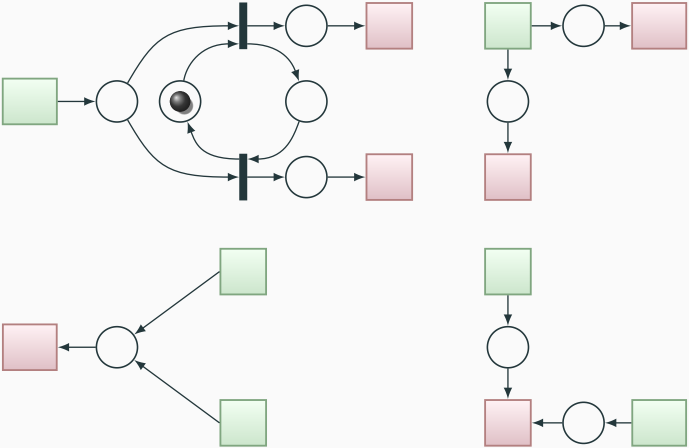
4Φ to 2Φ handshake converter



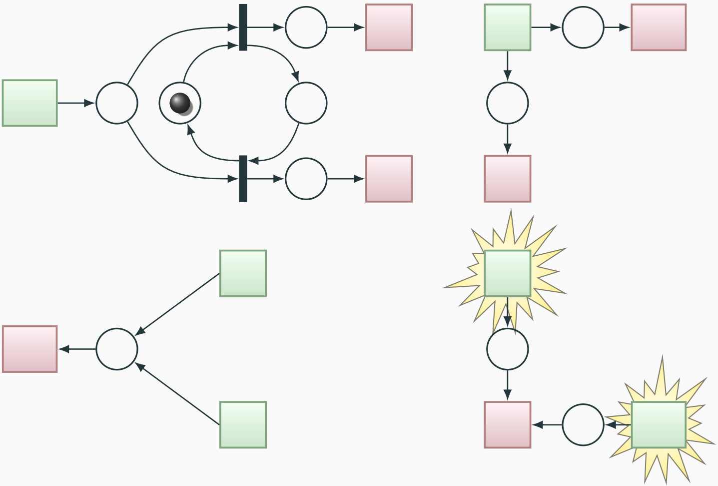
4Φ to 2Φ handshake converter



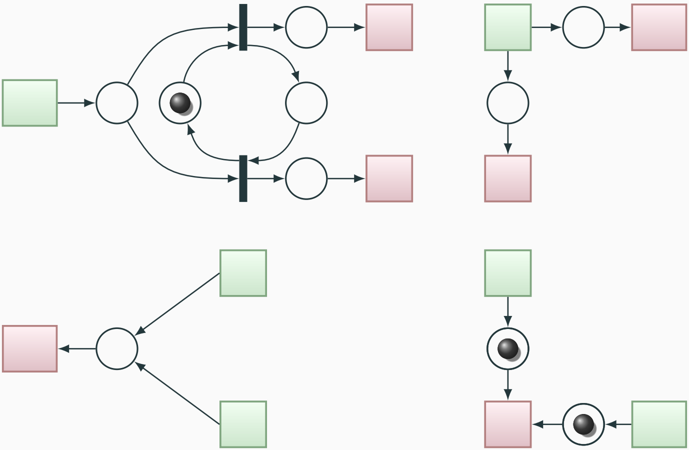
4Φ to 2Φ handshake converter



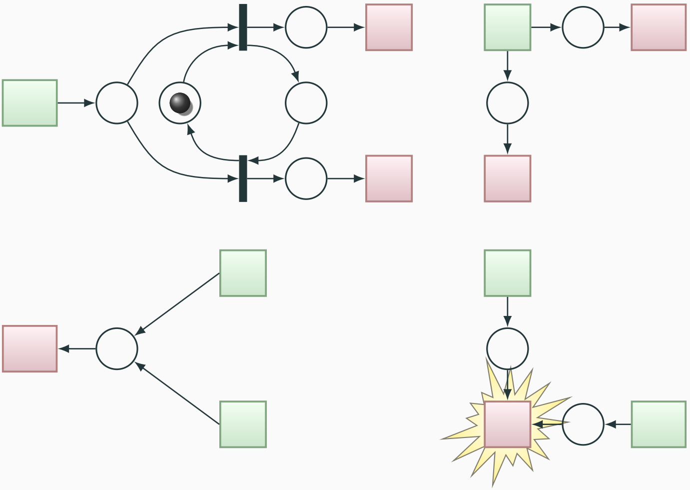
4Φ to 2Φ handshake converter



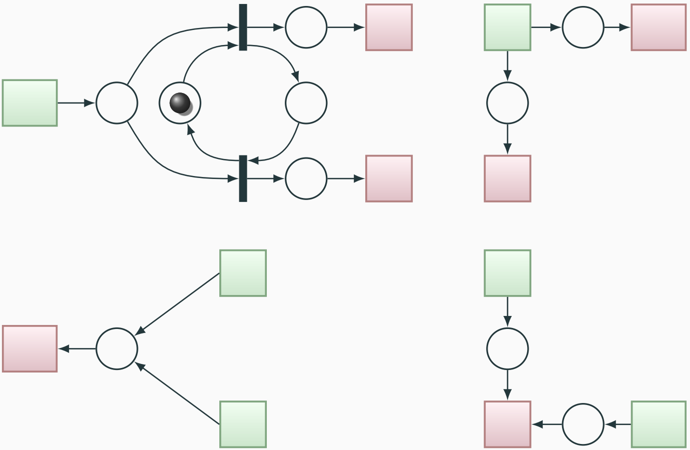
4Φ to 2Φ handshake converter



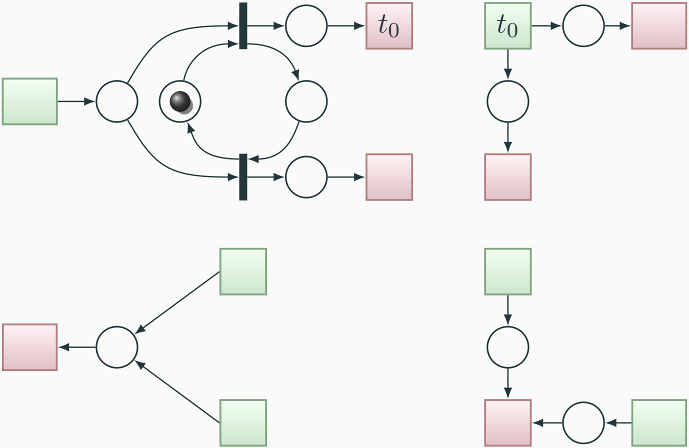
4Φ to 2Φ handshake converter



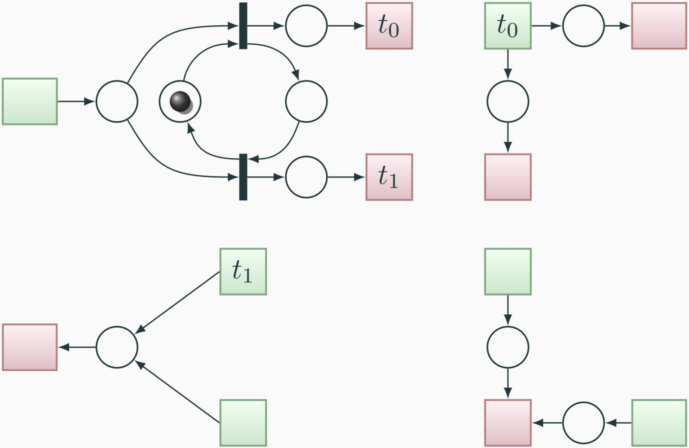
4Φ to 2Φ handshake converter



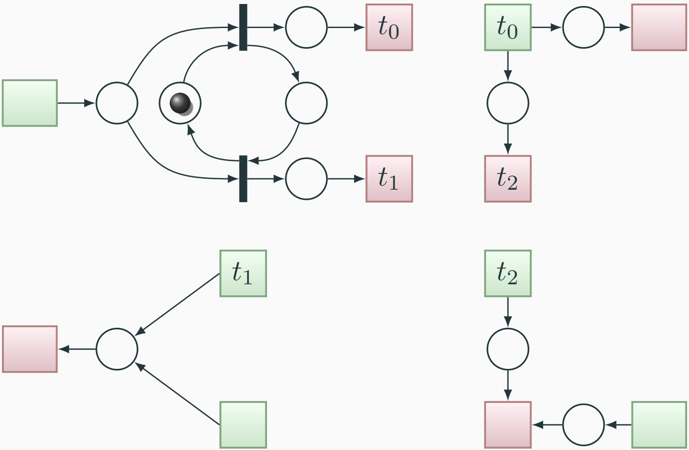
4Φ to 2Φ handshake converter



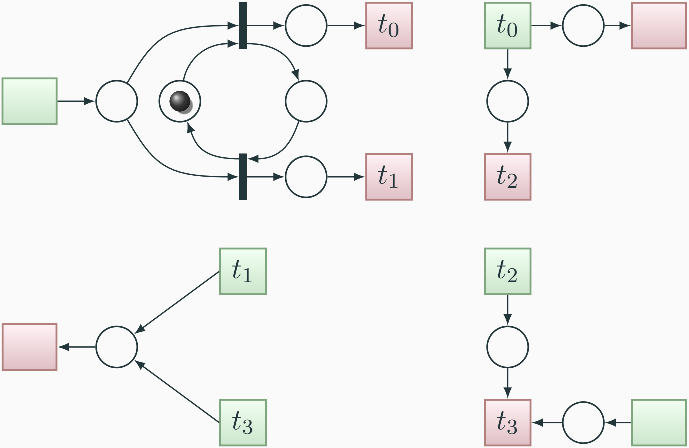
4Φ to 2Φ handshake converter



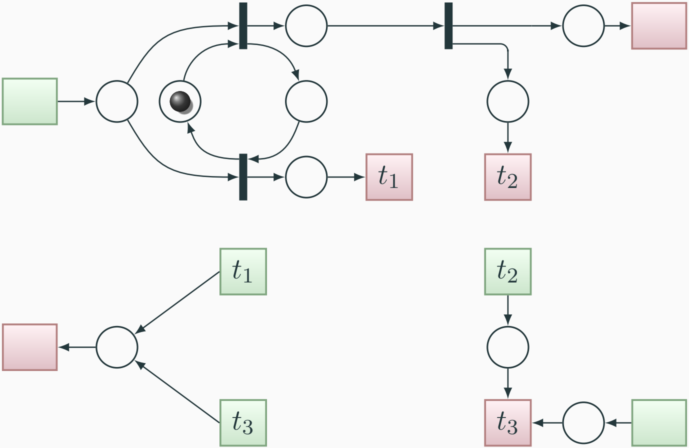
4Φ to 2Φ handshake converter



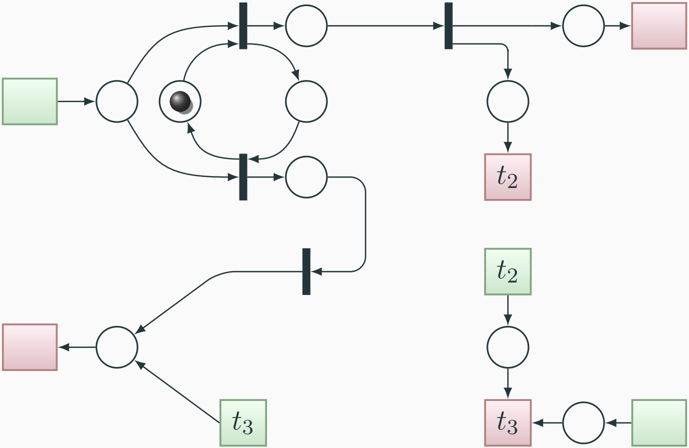
4Φ to 2Φ handshake converter



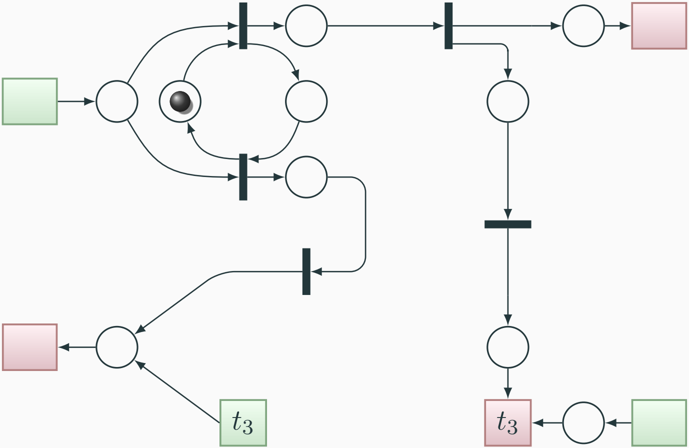
4Φ to 2Φ handshake converter



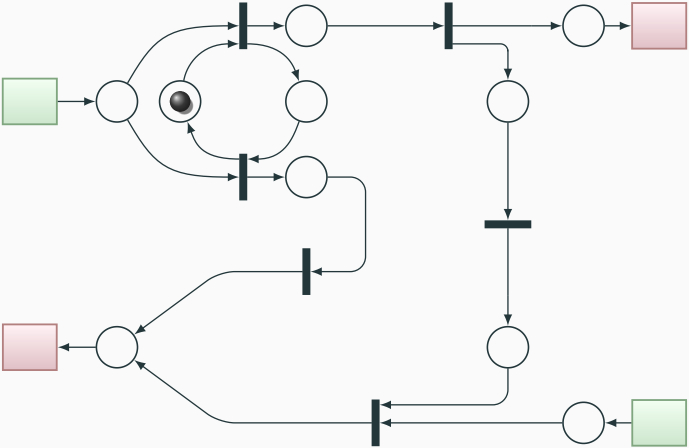
4Φ to 2Φ handshake converter



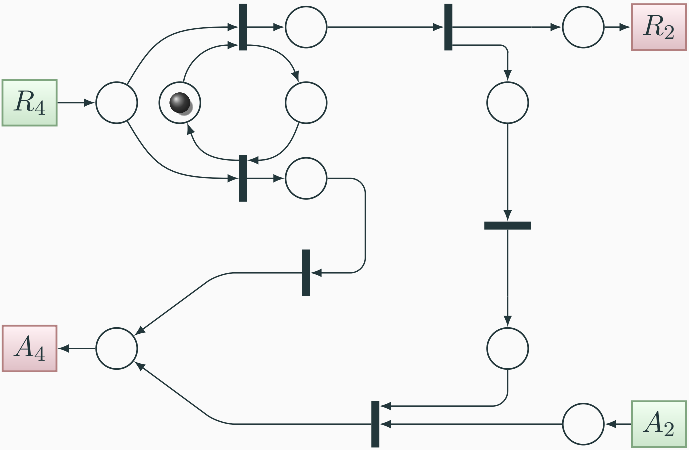
4Φ to 2Φ handshake converter



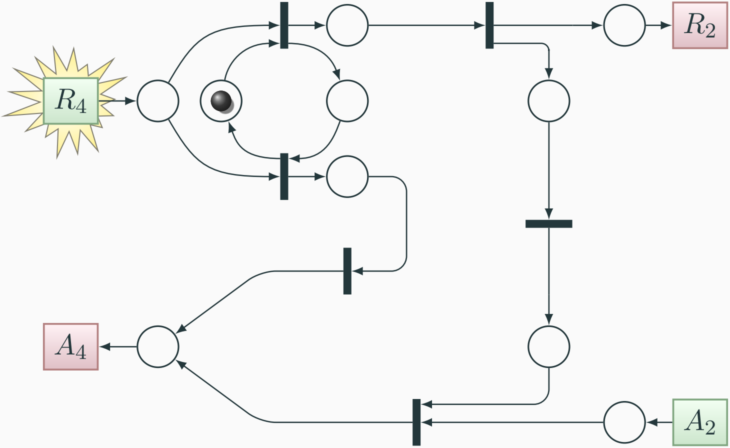
4Φ to 2Φ handshake converter



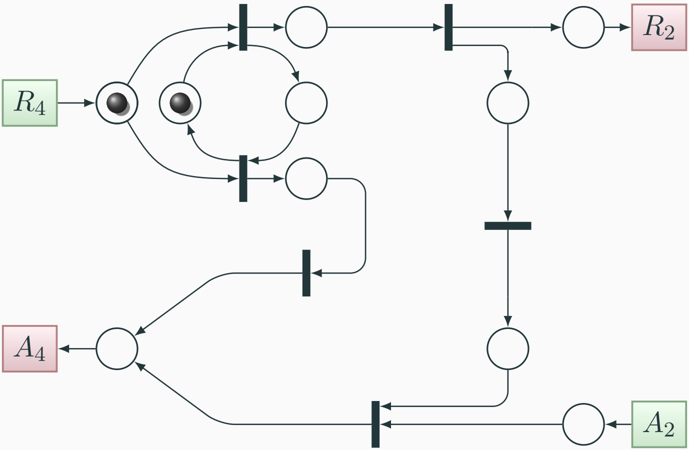
4Φ to 2Φ handshake converter



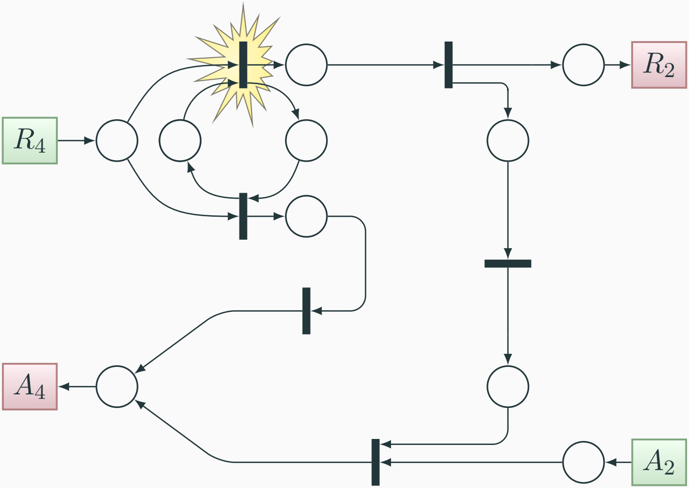
4Φ to 2Φ handshake converter



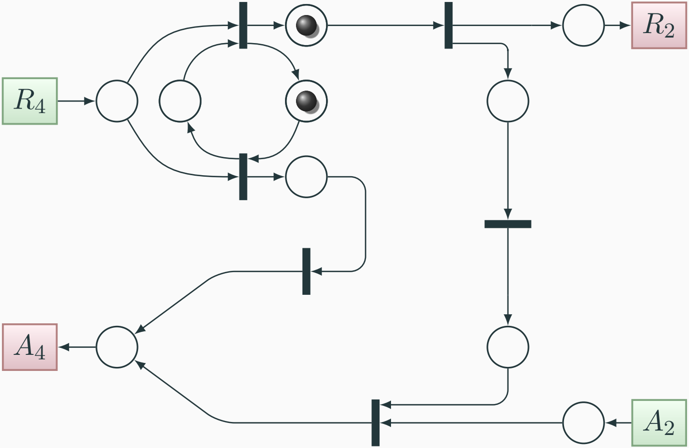
4Φ to 2Φ handshake converter



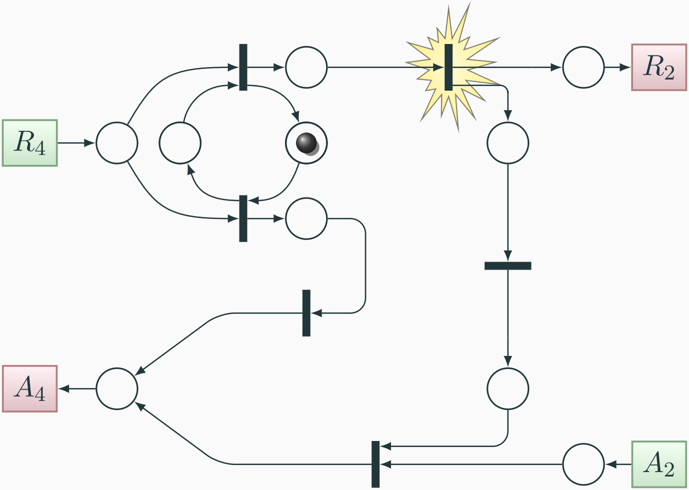
4Φ to 2Φ handshake converter



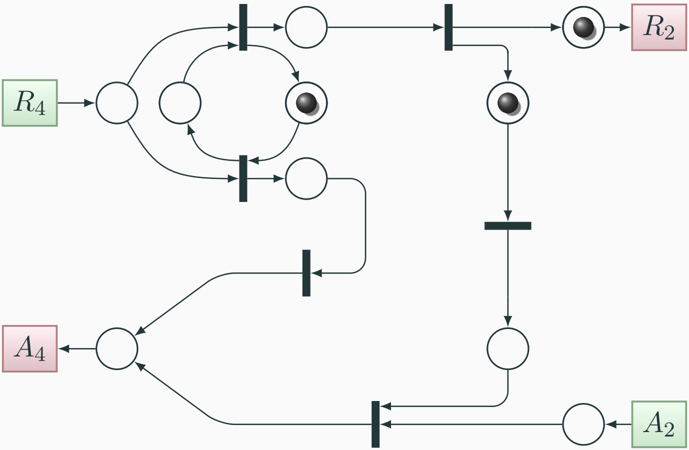
4Φ to 2Φ handshake converter



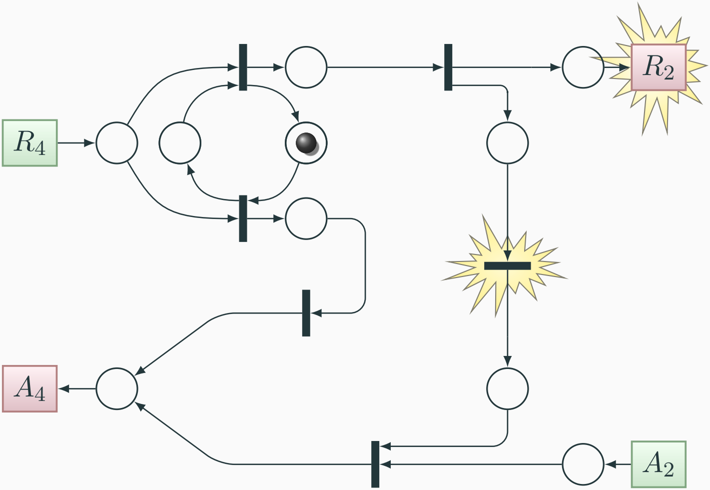
4Φ to 2Φ handshake converter



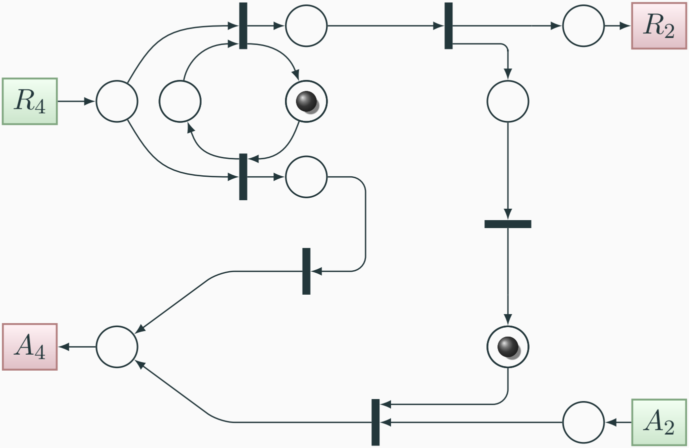
4Φ to 2Φ handshake converter



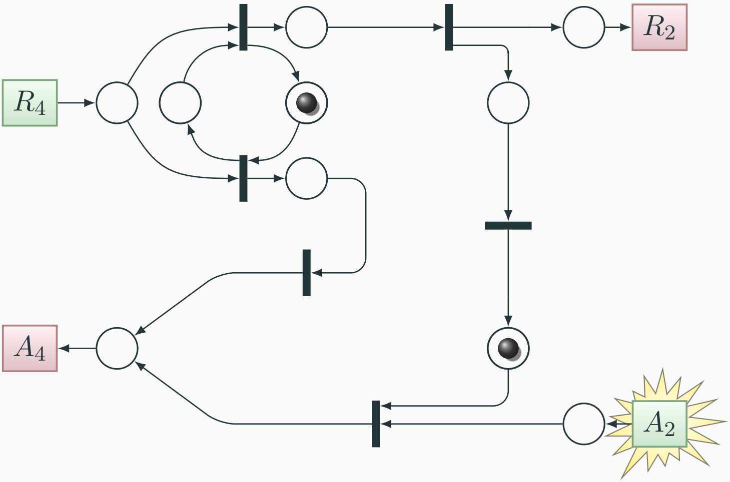
4Φ to 2Φ handshake converter



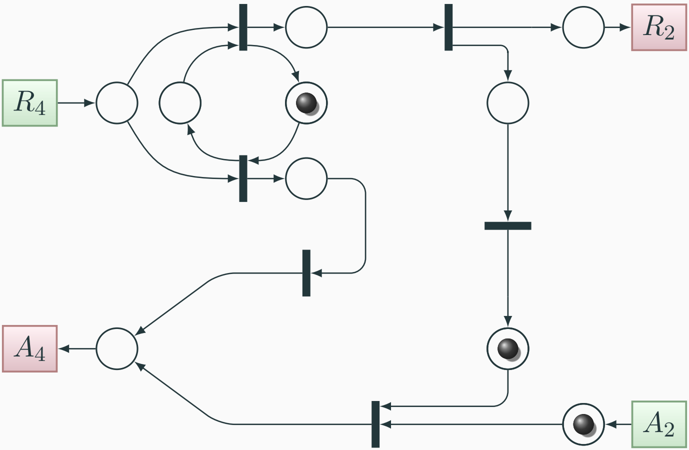
4Φ to 2Φ handshake converter



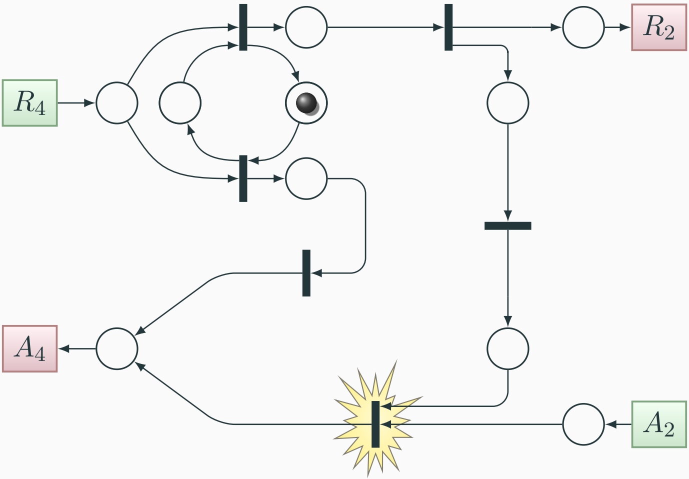
4Φ to 2Φ handshake converter



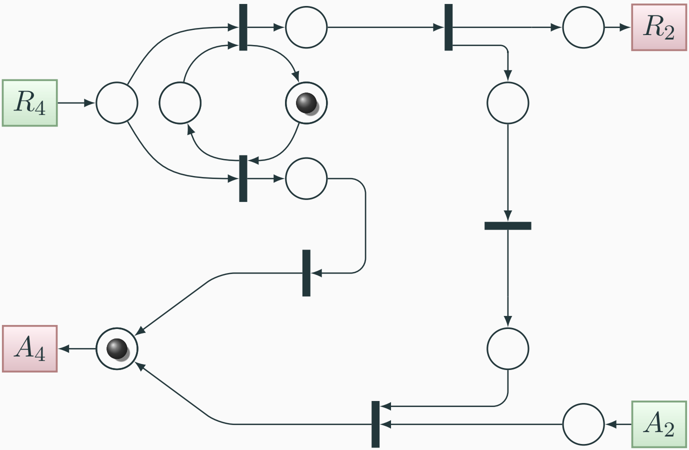
4Φ to 2Φ handshake converter



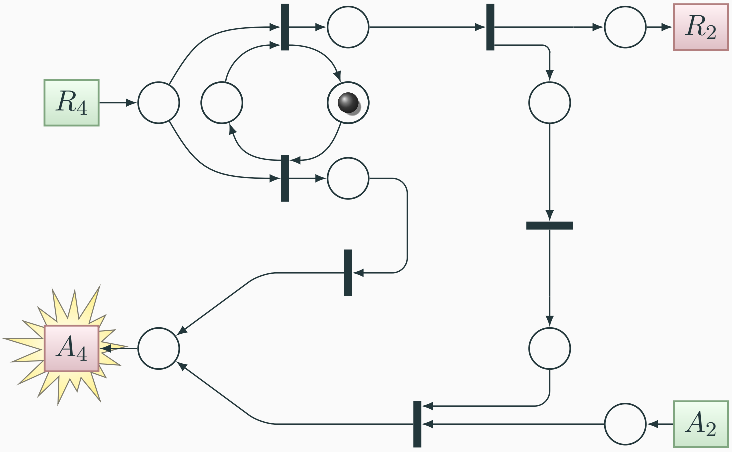
4Φ to 2Φ handshake converter



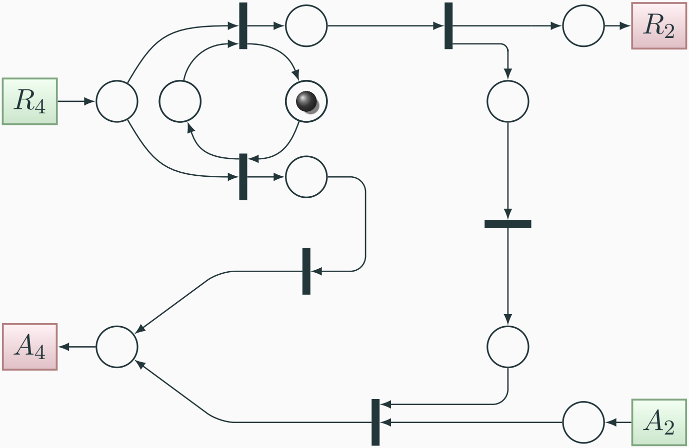
4Φ to 2Φ handshake converter



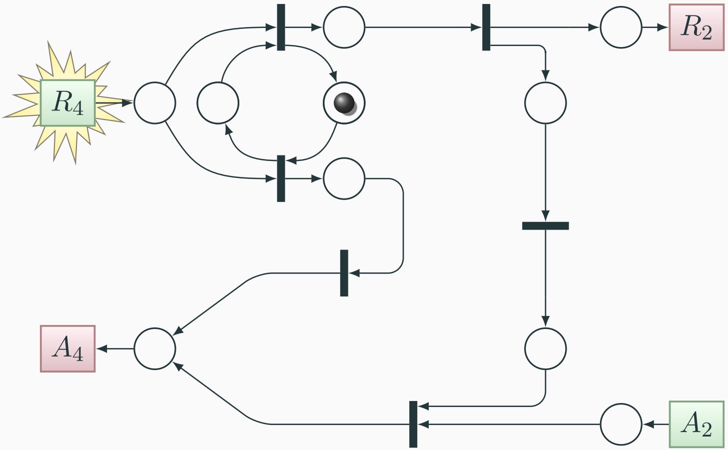
4Φ to 2Φ handshake converter



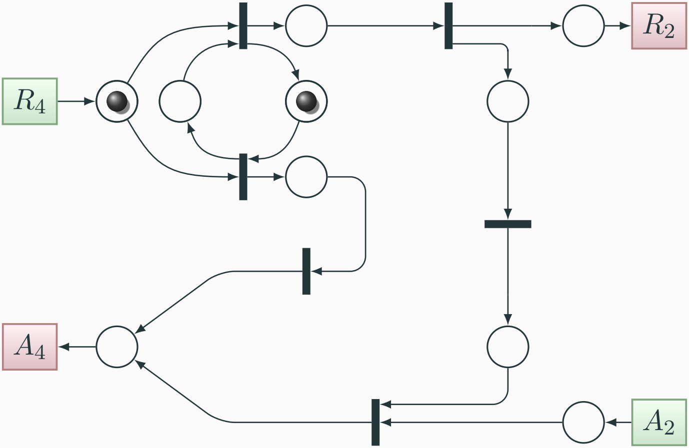
4Φ to 2Φ handshake converter



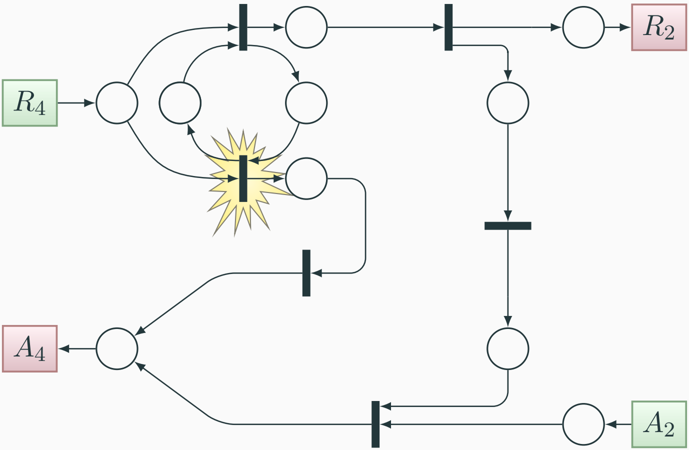
4Φ to 2Φ handshake converter



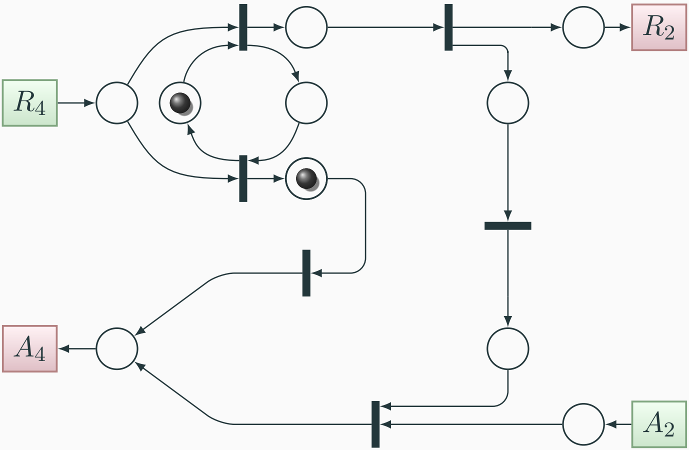
4Φ to 2Φ handshake converter



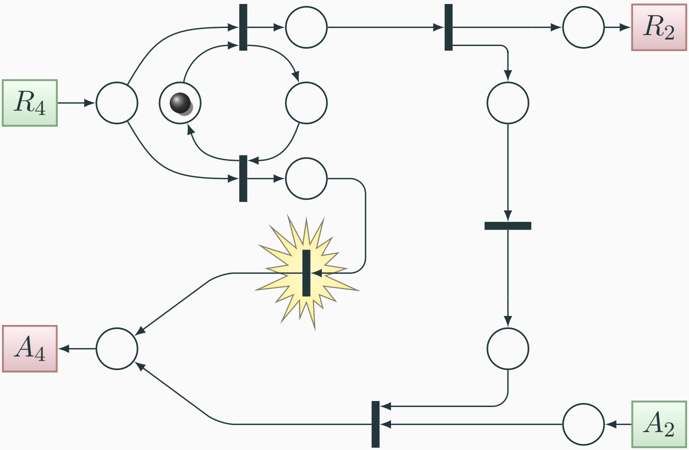
4Φ to 2Φ handshake converter



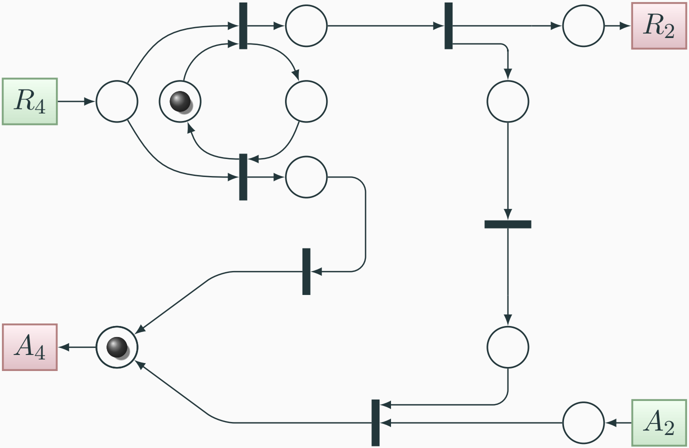
4Φ to 2Φ handshake converter



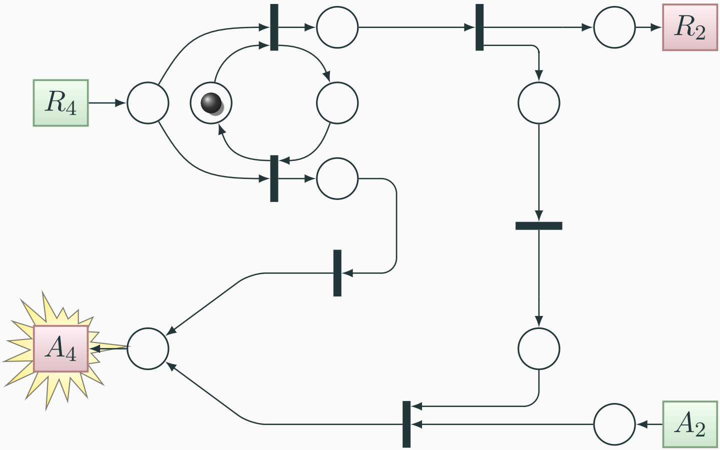
4Φ to 2Φ handshake converter



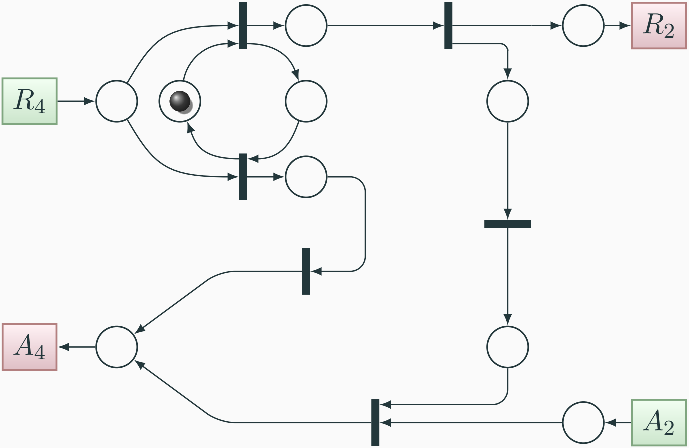
4Φ to 2Φ handshake converter



4Φ to 2Φ handshake converter



4Φ to 2Φ handshake converter



The formal approach

To-do list

Give a formal account of how components

- get connected into a network
- behave individually
- behave collectively when connected into a network

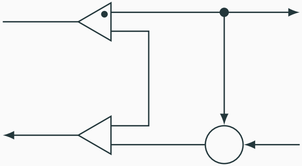
To-do list

Give a formal account of how components

- get connected into a network
- behave individually
- behave collectively when connected into a network

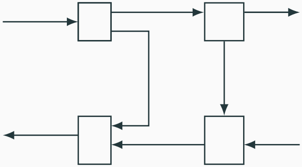
Block diagrams

Disregarding semantics, how can connections be specified ?



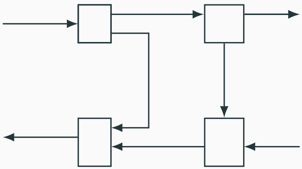
Block diagrams

Disregarding semantics, how can connections be specified ?



Block diagrams

Disregarding semantics, how can connections be specified ?

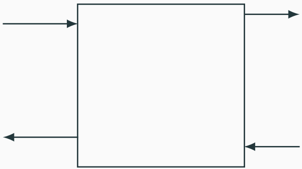


A formalism for block diagrams should enable

- algebraic description

Block diagrams

Disregarding semantics, how can connections be specified ?



A formalism for block diagrams should enable

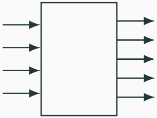
- algebraic description
- hierarchy

A *blockoid* over a set $\mathbf{b}$ is an algebraic structure $(\mathbf{b}, \mathbf{r}, \mathbf{z}, \mathbf{i})$ with

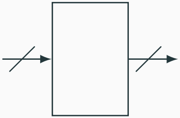
- $\mathbf{i} \in \mathbf{b}$
- $\mathbf{z} : \mathbf{b} \to \mathbf{b}$
- $\mathbf{r} : \mathbf{b} \times \mathbf{b} \to \mathbf{b}$

satisfying the *blockoid axioms*.

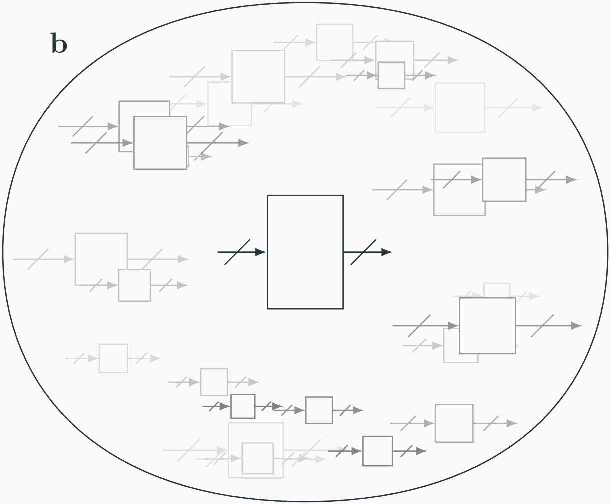
Intuition



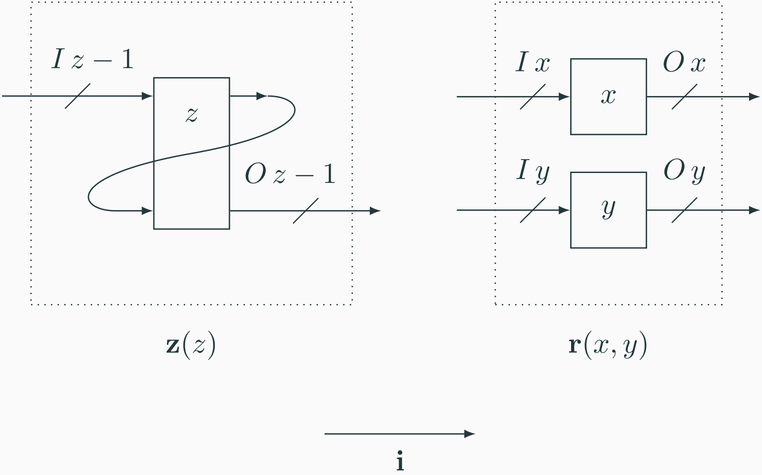
Intuition



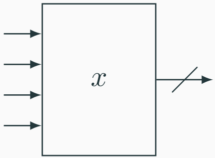
Intuition



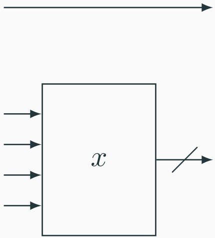
Intuition



Rolling down the inputs

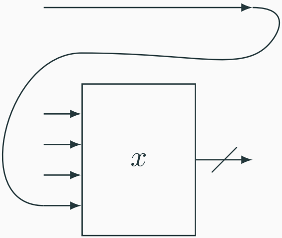


Rolling down the inputs



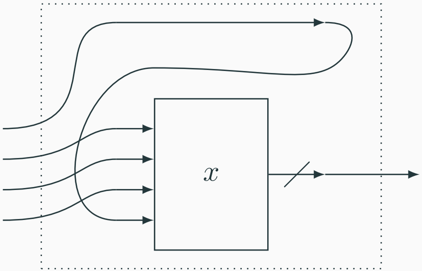
$$\mathbf{r}(\mathbf{i}, x)$$

Rolling down the inputs



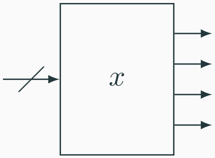
$$\mathbf{zr}(\mathbf{i}, x)$$

Rolling down the inputs

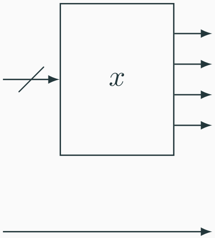


$$\mathbf{zr}(\mathbf{i}, x)$$

Rolling up the outputs

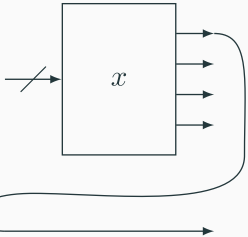


Rolling up the outputs



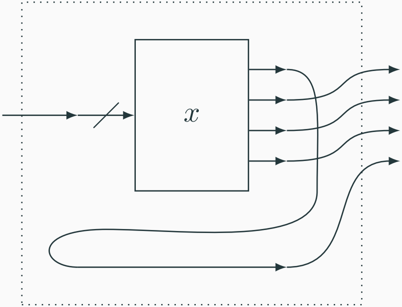
$$\mathbf{r}(x, \mathbf{i})$$

Rolling up the outputs



$$\mathbf{zr}(x, \mathbf{i})$$

Rolling up the outputs



$\mathbf{zr}(x, \mathbf{i})$

Blockoid axioms

There exists a *congruence* on $\mathbf{b}$ such that

- $\forall x \in \mathbf{b}.\ \exists i \in \mathbb{N}.\ \mathbf{t}\ x \equiv (\mathbf{z} \circ \mathbf{t})^{i+1}\, \mathbf{t}\ x$
- $\forall x \in \mathbf{b}.\ \exists o \in \mathbb{N}.\ \mathbf{v}\, x \equiv (\mathbf{z} \circ \mathbf{v})^{o+1}\, \mathbf{v}\, x$

are true, where $\mathbf{t}$ and $\mathbf{v}$ are defined by

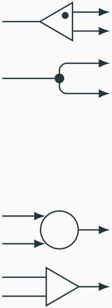
- $\mathbf{t} = \lambda x.\ \mathbf{r}(\mathbf{i}, x)$
- $\mathbf{v} = \lambda x.\ \mathbf{r}(x, \mathbf{i})$

The minimum $i, o \in \mathbb{N}$ satisfying these conditions for a block x are called its *input arity* $I\,x$ and its *output arity* $O\,x$.

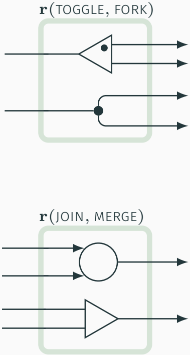
Blockoid axioms

idempotence	$\mathbf{zr}(\mathbf{i}, \mathbf{i}) \equiv \mathbf{i}$
left identity	$\mathbf{r}(\mathbf{z}\,\mathbf{i}, x) \equiv x$
right identity	$\mathbf{r}(x, \mathbf{z}\,\mathbf{i}) \equiv x$
associativity	$\mathbf{r}(x, \mathbf{r}(y, z)) \equiv \mathbf{r}(\mathbf{r}(x, y), z)$
input arity laws	$I\ \mathbf{z}\ x = I\ x - 1$
	$I\ \mathbf{r}(x, y) = I\ x + I\ y$
output arity laws	$O\ \mathbf{z}\ x = O\ x - 1$
	$O\ \mathbf{r}(x, y) = O\ y + O\ y$

Handshake converter blockoid

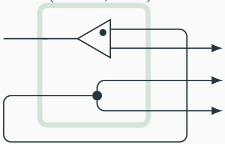


Handshake converter blockoid

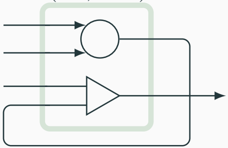


Handshake converter blockoid

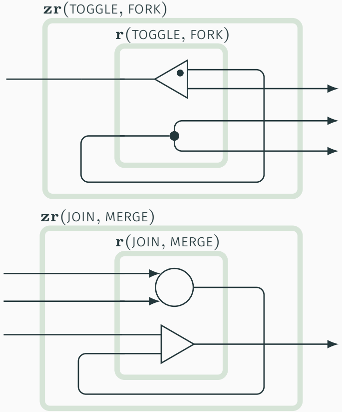
$\mathbf{r}$(TOGGLE, FORK)



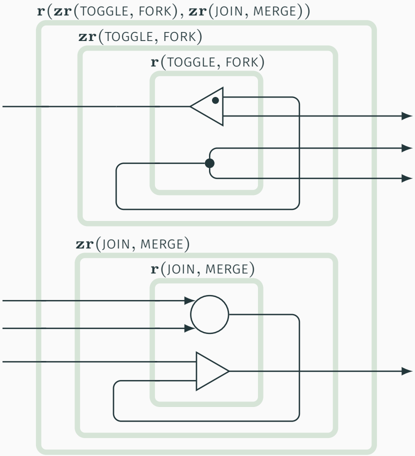
$\mathbf{r}$(JOIN, MERGE)



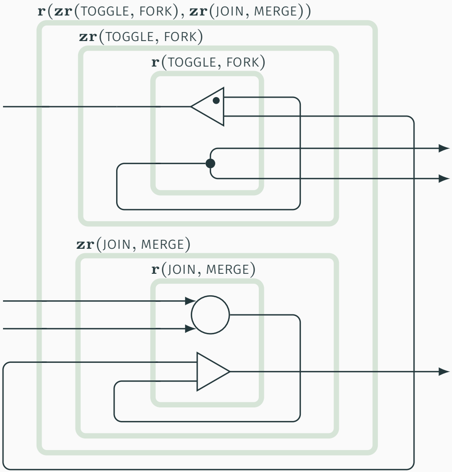
Handshake converter blockoid



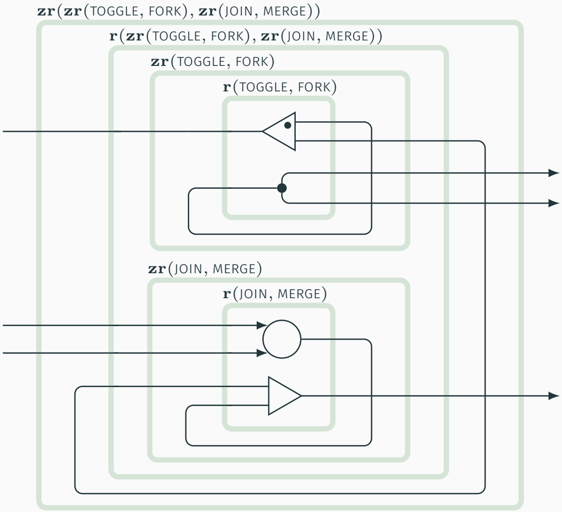
Handshake converter blockoid



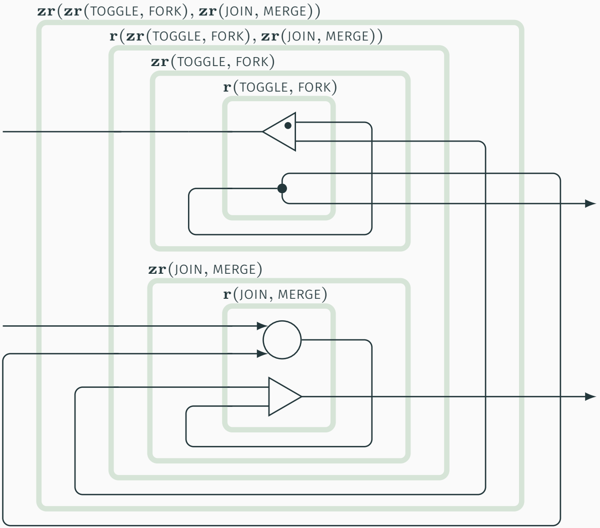
Handshake converter blockoid



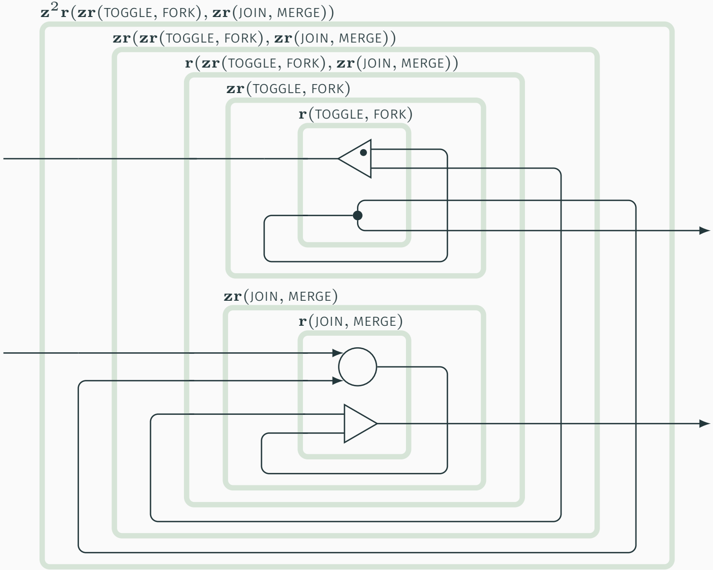
Handshake converter blockoid



Handshake converter blockoid

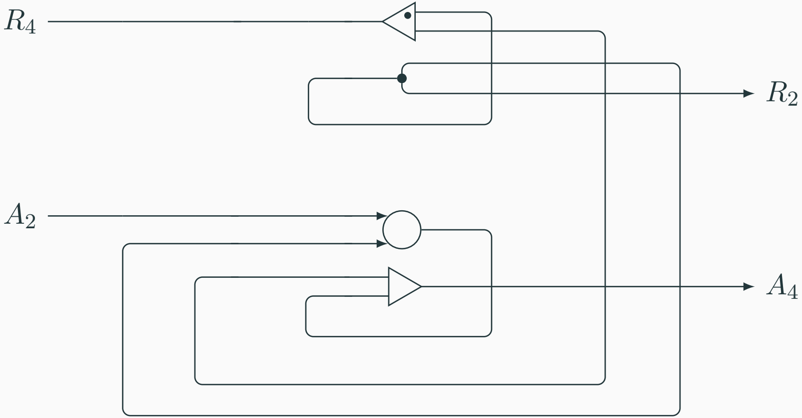


Handshake converter blockoid

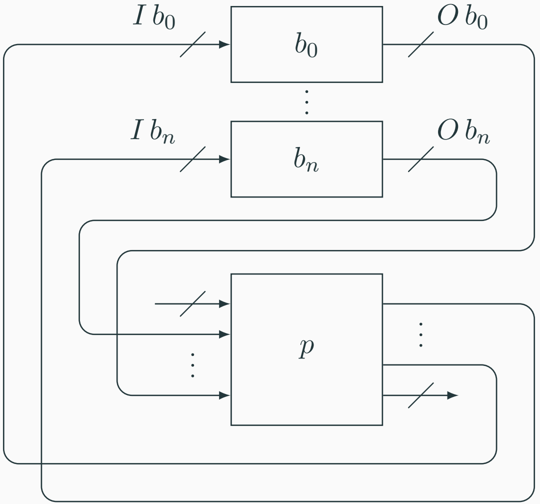


Handshake converter blockoid

$$\mathbf{z}^2\mathbf{r}(\mathbf{zr}(\text{TOGGLE}, \text{FORK}), \mathbf{zr}(\text{JOIN}, \text{MERGE}))$$



Universality of block combinators



Universality of block combinators

Any network of blocks $b = \langle b_0 \ldots b_n \rangle$ is expressible as

- bus from p to b
- exposed input bus
- bus from b to p

$$\mathbf{z}^w \mathbf{s}^v \mathbf{z}^u \, (F \; \mathbf{r}) \, (b \shortparallel \langle p \rangle)$$

$$\mathbf{t} = \lambda x.\, \mathbf{r}(\mathbf{i}, x) \qquad u = \sum_{t=0}^{n} O\, b_t$$

$$\mathbf{s} = \mathbf{z} \circ \mathbf{t} \qquad v = (I\, p) - u$$

$$w = \sum_{t=0}^{n} I\, b_t$$

for some *permutation network* p expressible by $\mathbf{r}$, $\mathbf{z}$ and $\mathbf{i}$.

Universality of block combinators

Any network of blocks $b = \langle b_0 \dots b_n \rangle$ is expressible as

- bus from p to b
- exposed input bus
- bus from b to p

$$\mathbf{z}^w \mathbf{s}^v \mathbf{z}^u \, (\digamma \, \mathbf{r}) \, (b \parallel \langle p \rangle)$$

fold

concatenate

$$\mathbf{t} = \lambda x.\ \mathbf{r}(\mathbf{i}, x) \qquad u = \sum_{t=0}^{n} O\, b_t$$

$$\mathbf{s} = \mathbf{z} \circ \mathbf{t} \qquad v = (I\, p) - u$$

$$w = \sum_{t=0}^{n} I\, b_t$$

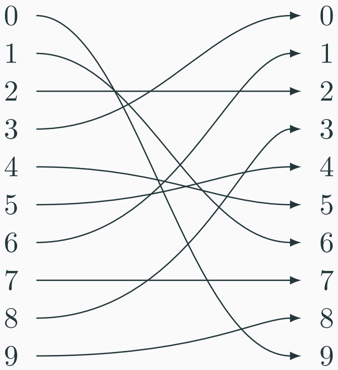
for some *permutation network* p expressible by $\mathbf{r}$, $\mathbf{z}$ and $\mathbf{i}$.

Permutation networks

x

$\langle 9, 6, 2, 0, 5, 4, 1, 7, 3, 8 \rangle$

$\mathbf{p}(x)$

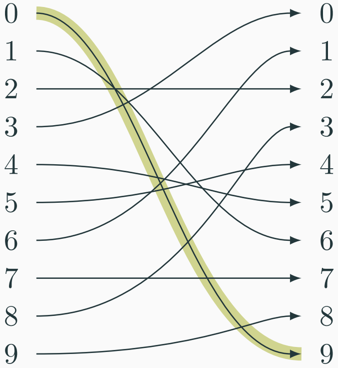


Permutation networks

x

$\langle 9, 6, 2, 0, 5, 4, 1, 7, 3, 8 \rangle$

$\mathbf{p}(x)$

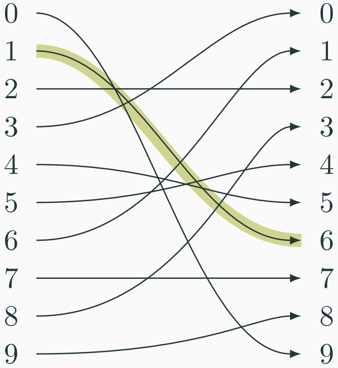


Permutation networks

x

$\langle 9, 6, 2, 0, 5, 4, 1, 7, 3, 8 \rangle$

$\mathbf{p}(x)$

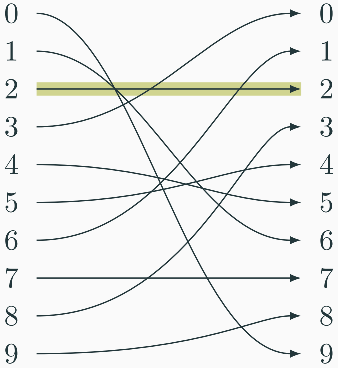


Permutation networks

x

$\langle 9, 6, 2, 0, 5, 4, 1, 7, 3, 8 \rangle$

$\mathbf{p}(x)$



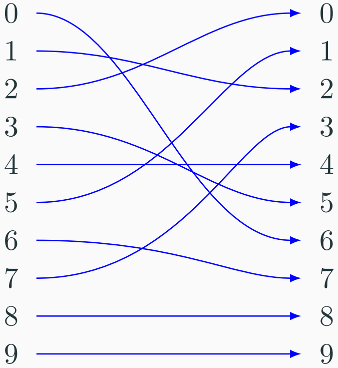
Permutation networks

x

$\langle 9, 6, 2, 0, 5, 4, 1, 7, 3, 8 \rangle$

$\langle 6, 2, 0, 5, 4, 1, 7, 3, 8, 9 \rangle$

$\mathbf{p}(x)$



Permutation networks

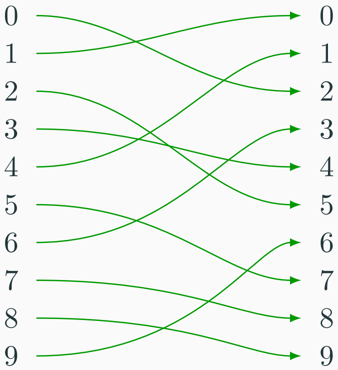
x

$\langle 9, 6, 2, 0, 5, 4, 1, 7, 3, 8 \rangle$

$\langle 6, 2, 0, 5, 4, 1, 7, 3, 8, 9 \rangle$

$\langle 2, 0, 5, 4, 1, 7, 3, 8, 9, 6 \rangle$

$\mathbf{p}(x)$



Permutation networks

x

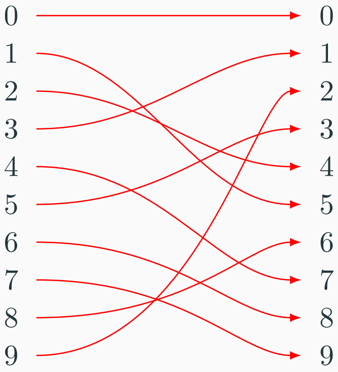
$\langle 9, 6, 2, 0, 5, 4, 1, 7, 3, 8 \rangle$

$\langle 6, 2, 0, 5, 4, 1, 7, 3, 8, 9 \rangle$

$\langle 2, 0, 5, 4, 1, 7, 3, 8, 9, 6 \rangle$

$\langle 0, 5, 4, 1, 7, 3, 8, 9, 6, 2 \rangle$

$\mathbf{p}(x)$



Permutation networks

$$x$$

$$\langle 9, 6, 2, 0, 5, 4, 1, 7, 3, 8 \rangle$$

$$\langle 6, 2, 0, 5, 4, 1, 7, 3, 8, 9 \rangle$$

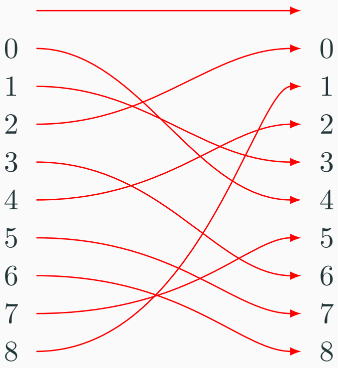
$$\langle 2, 0, 5, 4, 1, 7, 3, 8, 9, 6 \rangle$$

$$\langle \not{0}, 5, 4, 1, 7, 3, 8, 9, 6, 2 \rangle$$

$$\Downarrow -1$$

$$x' = \langle 4, 3, 0, 6, 2, 7, 8, 5, 1 \rangle$$

$$\mathbf{p}(x) = \mathbf{r}(\mathbf{i}, \mathbf{p}(x'))$$



Permutation networks

x

$\langle 9, 6, 2, 0, 5, 4, 1, 7, 3, 8 \rangle$

$\langle 6, 2, 0, 5, 4, 1, 7, 3, 8, 9 \rangle$

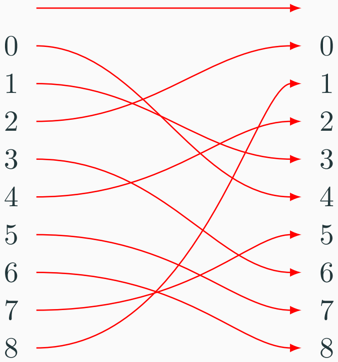
$\langle 2, 0, 5, 4, 1, 7, 3, 8, 9, 6 \rangle$

$\langle \cancel{0}, 5, 4, 1, 7, 3, 8, 9, 6, 2 \rangle$

$\Downarrow -1$

$$x' = \langle 4, 3, 0, 6, 2, 7, 8, 5, 1 \rangle$$
$$= (x \circlearrowleft h \ll 1) - 1$$

$\mathbf{p}(x) = \mathbf{r}(\mathbf{i}, \mathbf{p}(x'))$



Permutation networks

$$x$$

$$\langle 9, 6, 2, 0, 5, 4, 1, 7, 3, 8\rangle$$

$$\langle 6, 2, 0, 5, 4, 1, 7, 3, 8, 9\rangle$$

$$\langle 2, 0, 5, 4, 1, 7, 3, 8, 9, 6\rangle$$

$$\langle \not{0}, 5, 4, 1, 7, 3, 8, 9, 6, 2\rangle$$

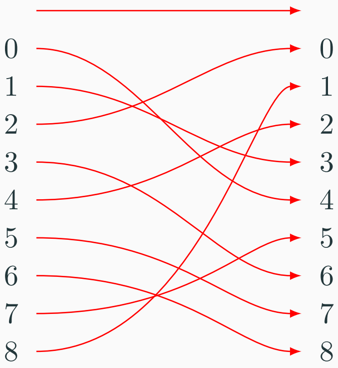
$$\Downarrow -1$$

$$x' = \langle 4, 3, 0, 6, 2, 7, 8, 5, 1\rangle$$

$$= (x \circlearrowleft h \ll 1) - 1$$

roll left *drop* *decrement*

$$\mathbf{p}(x) = \mathbf{r}(\mathbf{i}, \mathbf{p}(x'))$$



Permutation networks

$$x$$

$$\langle 9,6,2,0,5,4,1,7,3,8 \rangle$$

$$\langle 6,2,0,5,4,1,7,3,8,9 \rangle$$

$$\langle 2,0,5,4,1,7,3,8,9,6 \rangle$$

$$\langle \not{0},5,4,1,7,3,8,9,6,2 \rangle$$

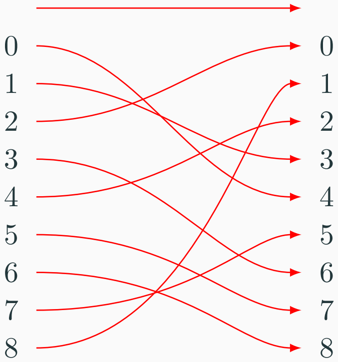
$$\Downarrow -1$$

$$x' = \langle 4,3,0,6,2,7,8,5,1 \rangle$$

$$= (x \circlearrowleft h \ll 1) - 1$$

where $h = (x^{-1})_0$

$$\mathbf{p}(x) = \mathbf{r}(\mathbf{i}, \mathbf{p}(x'))$$



Permutation networks

$$x$$

$$\langle 9,6,2,0,5,4,1,7,3,8\rangle$$

$$\langle 6,2,0,5,4,1,7,3,8,9\rangle$$

$$\langle 2,0,5,4,1,7,3,8,9,6\rangle$$

$$\langle \not{0},5,4,1,7,3,8,9,6,2\rangle$$

$$\Downarrow -1$$

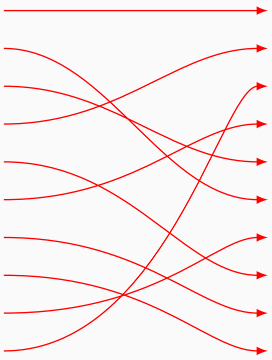
$$x' = \langle 4,3,0,6,2,7,8,5,1\rangle$$

$$= (x \circlearrowleft h \ll 1) - 1$$

where $h = (x^{-1})_0$

$$\mathbf{t} = \lambda a.\ \mathbf{r}(\mathbf{i}, a)$$

$$\mathbf{p}(x) = \mathbf{r}(\mathbf{i}, \mathbf{p}(x'))$$



Permutation networks

x

$\langle 9, 6, 2, 0, 5, 4, 1, 7, 3, 8 \rangle$

$\langle 6, 2, 0, 5, 4, 1, 7, 3, 8, 9 \rangle$

$\langle 2, 0, 5, 4, 1, 7, 3, 8, 9, 6 \rangle$

$\langle \cancel{0}, 5, 4, 1, 7, 3, 8, 9, 6, 2 \rangle$

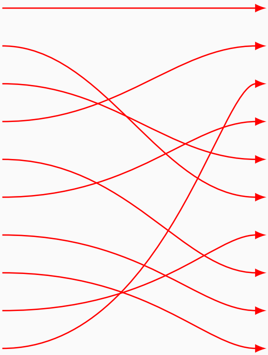
$\Downarrow -1$

$$x' = \langle 4, 3, 0, 6, 2, 7, 8, 5, 1 \rangle$$
$$= (x \circlearrowleft h \ll 1) - 1$$

where $h = (x^{-1})_0$

$\mathbf{t} = \lambda a.\ \mathbf{r}(\mathbf{i}, a)$

$\mathbf{p}(x) = \mathbf{tp}(x')$



Permutation networks

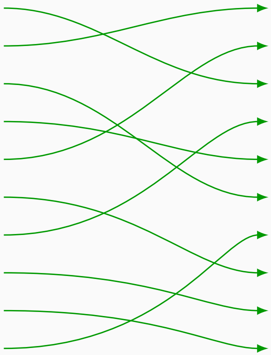
$$x$$

$$\langle 9,6,2,0,5,4,1,7,3,8\rangle$$
$$\langle 6,2,0,5,4,1,7,3,8,9\rangle$$
$$\langle 2,0,5,4,1,7,3,8,9,6\rangle$$
$$\langle \cancel{0},5,4,1,7,3,8,9,6,2\rangle$$
$$\Downarrow -1$$
$$x' = \langle 4,3,0,6,2,7,8,5,1\rangle$$
$$= (x \circlearrowleft h \ll 1) - 1$$

where $h = (x^{-1})_0$

$$\mathbf{t} = \lambda a.\ \mathbf{r}(\mathbf{i}, a)$$
$$\mathbf{s} = \mathbf{z} \circ \mathbf{t}$$

$$\mathbf{p}(x) = \mathbf{stp}(x')$$



Permutation networks

$$x$$

$$\langle 9, 6, 2, 0, 5, 4, 1, 7, 3, 8 \rangle$$

$$\langle 6, 2, 0, 5, 4, 1, 7, 3, 8, 9 \rangle$$

$$\langle 2, 0, 5, 4, 1, 7, 3, 8, 9, 6 \rangle$$

$$\langle \not{0}, 5, 4, 1, 7, 3, 8, 9, 6, 2 \rangle$$

$$\Downarrow -1$$

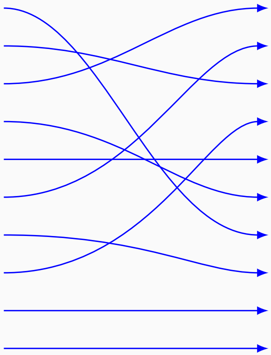
$$\begin{aligned} x' &= \langle 4, 3, 0, 6, 2, 7, 8, 5, 1 \rangle \\ &= (x \circlearrowleft h \ll 1) - 1 \end{aligned}$$

where $h = (x^{-1})_0$

$$\mathbf{t} = \lambda a.\ \mathbf{r}(\mathbf{i}, a)$$

$$\mathbf{s} = \mathbf{z} \circ \mathbf{t}$$

$$\mathbf{p}(x) = \mathbf{s}^2 \mathbf{t} \mathbf{p}(x')$$



Permutation networks

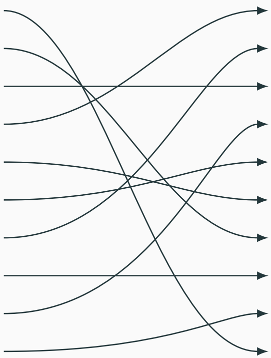
$$x$$

$$\langle 9, 6, 2, 0, 5, 4, 1, 7, 3, 8 \rangle$$
$$\langle 6, 2, 0, 5, 4, 1, 7, 3, 8, 9 \rangle$$
$$\langle 2, 0, 5, 4, 1, 7, 3, 8, 9, 6 \rangle$$
$$\langle \not{0}, 5, 4, 1, 7, 3, 8, 9, 6, 2 \rangle$$
$$\Downarrow -1$$
$$x' = \langle 4, 3, 0, 6, 2, 7, 8, 5, 1 \rangle$$
$$= (x \circlearrowleft h \ll 1) - 1$$

where $h = (x^{-1})_0$

$$\mathbf{t} = \lambda a.\ \mathbf{r}(\mathbf{i}, a)$$
$$\mathbf{s} = \mathbf{z} \circ \mathbf{t}$$

$$\mathbf{p}(x) = \mathbf{s}^h \mathbf{t} \mathbf{p}(x')$$



Permutation network combinator definition

The permutation obtained from $x \in \mathbb{N}^*$ rolled left by

$$h = (x^{-1})_0$$

decapitated and pointwise decremented

$$x' = (x \circlearrowleft h \ll 1) - 1$$

implies the recurrence

$$\mathbf{p}(x) = \begin{cases} \mathbf{i} & \text{if } |x| = 1 \\ (\lambda h.\, \mathbf{s}^h \mathbf{t} \mathbf{p}((x \circlearrowleft h \ll 1) - 1))\,(x^{-1})_0 & \text{if } |x| > 1 \end{cases}$$

implying universality of $\mathbf{r}$, $\mathbf{z}$, and $\mathbf{i}$ for block diagrams !

Permutation network examples

x	$\mathbf{p}(x)$
$\langle 9,6,2,0,5,4,1,7,3,8\rangle$	$\mathbf{s^3ts^2ts^5ts^3ts^4ts^4ts^3t^3i}$
$\langle 5,0,9,3,1,8,7,2,6,4\rangle$	$\mathbf{sts^2ts^2ts^4ts^3t^2s^3ts^2tsti}$
$\langle 4,5,1,7,3,2,8,0,9,6\rangle$	$\mathbf{s^7ts^4ts^2ts^6ts^3t^2s^3t^3i}$
$\langle 1,3,2,7,8,6,9,0,4,5\rangle$	$\mathbf{s^7ts^2tsts^6ts^4t^2s^2tst^2i}$
$\langle 2,9,6,0,1,5,3,7,8,4\rangle$	$\mathbf{s^3t^2s^5ts^3ts^2ts^2ts^3t^3i}$
$\langle 8,3,7,6,0,1,9,5,2,4\rangle$	$\mathbf{s^4t^2s^2ts^2ts^4ts^4ts^2ts^2tsti}$
$\langle 8,9,7,0,5,2,4,1,6,3\rangle$	$\mathbf{s^3ts^3ts^6ts^2ts^4ts^4t^2s^2t^2i}$
$\langle 0,9,2,7,1,4,3,8,5,6\rangle$	$\mathbf{ts^3ts^6ts^2ts^5tst^2st^2i}$
$\langle 5,3,4,8,1,7,0,6,2,9\rangle$	$\mathbf{s^6ts^7ts^2ts^2t^2s^4ts^2ts^2tsti}$
$\langle 0,6,2,4,8,5,3,7,9,1\rangle$	$\mathbf{ts^8tsts^3ts^3tsts^2tstsi}$
$\langle 3,0,7,9,1,8,6,2,5,4\rangle$	$\mathbf{sts^2ts^2ts^2ts^5ts^4ts^3t^2sti}$

To-do list

Give a formal account of how components

- ✓ get connected into a network
- behave individually
- behave collectively when connected into a network

To-do list

Give a formal account of how components

- ✓ get connected into a network
- behave individually
- behave collectively when connected into a network

Component models – first attempt

Model the components as Petri nets.

$\mathbb{T}$	universe of observable transitions
$\mathbb{V}$	universe of places and unobservable transitions
$\mathbb{P}$	Petri nets (P, T, A, M, F)

- places $P \subset \mathbb{V}$
- transitions $T \subset \mathbb{T} \cup \mathbb{V}$
- arcs $A \subseteq (P \times T) \cup (T \times P)$
- initial marking $M \subseteq P$
- final marking $F \subseteq P$

Component models – first attempt

Model the components as Petri nets.

$\mathbb{T}$	universe of observable transitions
$\mathbb{V}$	universe of places and unobservable transitions
$\mathbb{P}$	Petri nets (P, T, A, M, F)

- places $P \subset \mathbb{V}$
- transitions $T \subset \mathbb{T} \cup \mathbb{V}$
- arcs $A \subseteq (P \times T) \cup (T \times P)$
- initial marking $M \subseteq P$
- final marking $F \subseteq P$

but then inputs and outputs are indistinguishable

Component models – second attempt

Model the components as DI processes.

$\mathbb{T}$	universe of observable transitions
$\mathbb{V}$	universe of places and unobservable transitions
$\mathbb{P}$	Petri nets (P, T, A, M, F)
$\mathbb{D}$	delay insensitive processes $p \in \mathscr{P}(\mathbb{T}) \times \mathscr{P}(\mathbb{T}) \times \mathbb{P}$

For $p = (I, O, N) \in \mathbb{D}$

- Petri net $N = (P, T, A, M, F) \in \mathbb{P}$
- input alphabet $I \supseteq T \cap \mathbb{T}$
- output alphabet $O \supseteq T \cap \mathbb{T}$

Refinement over processes

From a delay insensitive process $X = (I, O, N) \in \mathbb{D}$, infer

- a *reachability graph* $\mathbf{RG}(X)$ from the Petri net N.

From the reachability graph, infer

- a *quiescent trace recognizing automaton* $\mathbf{QR}(X)$
- a *divergent trace recognizing automaton* $\mathbf{DR}(X)$

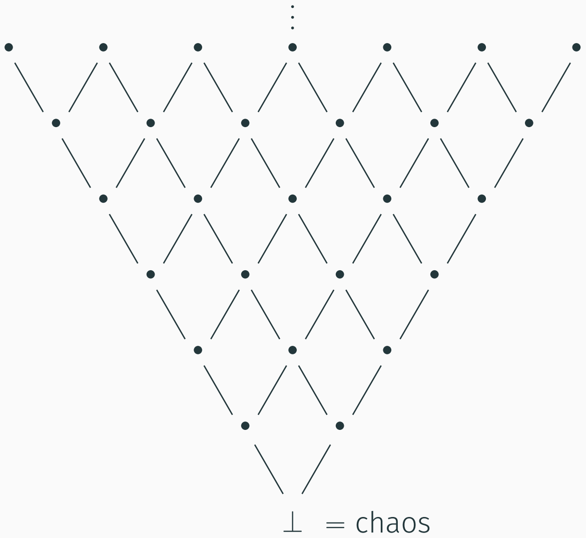
From their languages $\mathcal{L}\,\mathbf{QR}(X), \mathcal{L}\,\mathbf{DR}(X) \in (I \cup O)^*$, infer

- a *relational trace set* $[\![X]\!] = \mathcal{L}\,\mathbf{QR}(X) \cup \mathcal{L}\,\mathbf{DR}(X)$

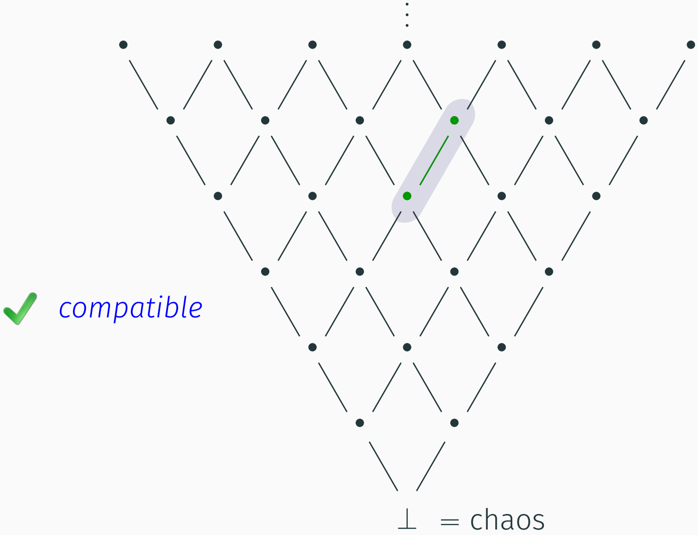
such that the *refinement* relation $X \sqsubseteq Y$ coincides with

$$[\![X]\!] \supseteq [\![Y]\!].$$

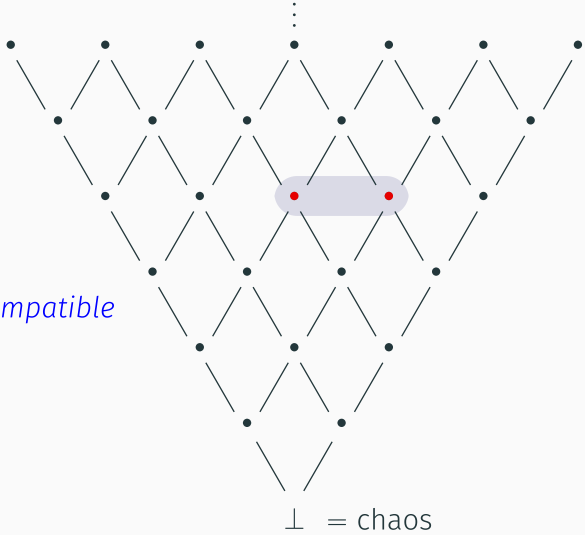
Complete partial ordering by refinement



Complete partial ordering by refinement



Complete partial ordering by refinement



✖ *incompatible*

Component models – second attempt

Model the components as delay insensitive processes.

$\mathbb{T}$	universe of observable transitions
$\mathbb{V}$	universe of places and unobservable transitions
$\mathbb{P}$	Petri nets (P, T, A, M, F)
$\mathbb{D}$	delay insensitive processes $p \in \mathscr{P}(\mathbb{T}) \times \mathscr{P}(\mathbb{T}) \times \mathbb{P}$

For $p = (I, O, N) \in \mathbb{D}$

- Petri net $N = (P, T, A, M, F) \in \mathbb{P}$
- input alphabet $I \supseteq T \cap \mathbb{T}$
- output alphabet $O \supseteq T \cap \mathbb{T}$

Component models – second attempt

Model the components as delay insensitive processes.

$\mathbb{T}$	universe of observable transitions
$\mathbb{V}$	universe of places and unobservable transitions
$\mathbb{P}$	Petri nets (P, T, A, M, F)
$\mathbb{D}$	delay insensitive processes $p \in \mathscr{P}(\mathbb{T}) \times \mathscr{P}(\mathbb{T}) \times \mathbb{P}$

For $p = (I, O, N) \in \mathbb{D}$

- Petri net $N = (P, T, A, M, F) \in \mathbb{P}$
- input alphabet $I \supseteq T \cap \mathbb{T}$
- output alphabet $O \supseteq T \cap \mathbb{T}$

but then name clashes among transitions are inconvenient

Component models – third attempt

Model the components as blocks with terminals.

$\mathbb{T}$	universe of observable transitions
$\mathbb{V}$	universe of places and unobservable transitions
$\mathbb{P}$	Petri nets (P, T, A, M, F)
$\mathbb{D}$	delay insensitive processes $p \in \mathscr{P}(\mathbb{T}) \times \mathscr{P}(\mathbb{T}) \times \mathbb{P}$
$\mathbb{B}$	primitive blocks $b \in \mathbb{N} \times \mathbb{N} \times (((\mathbb{T}^\star \times \mathbb{T}^\star) \to \mathbb{D}))$

for $b = (I, O, B) \in \mathbb{B}$

- input arity $I \in \mathbb{N}$
- output arity $O \in \mathbb{N}$
- process $(I', O', N) = B(i, o)$ has $I' = \mathscr{R}(i), O' = \mathscr{R}(o)$

Component models – third attempt

Model the components as blocks with terminals.

$\mathbb{T}$	universe of observable transitions
$\mathbb{V}$	universe of places and unobservable transitions
$\mathbb{P}$	Petri nets (P, T, A, M, F)
$\mathbb{D}$	delay insensitive processes $p \in \mathscr{P}(\mathbb{T}) \times \mathscr{P}(\mathbb{T}) \times \mathbb{P}$
$\mathbb{B}$	primitive blocks $b \in \mathbb{N} \times \mathbb{N} \times (((\mathbb{T}^\star \times \mathbb{T}^\star) \to \mathbb{D}))$

for $b = (I, O, B) \in \mathbb{B}$

range of a list

- input arity $I \in \mathbb{N}$
- output arity $O \in \mathbb{N}$
- process $(I', O', N) = B(i, o)$ has $I' = \mathscr{R}(i), O' = \mathscr{R}(o)$

Refinement over components

For a standardized infinite list $\mathbb{G} \in \mathbb{T}^*$ of generic symbols, let

$$\mathcal{T}_{\mathbb{BD}} : \mathbb{B} \to \mathbb{D}$$

map components to processes by

$$\mathcal{T}_{\mathbb{BD}}(I, O, B) = B(\mathbb{G} \upharpoonright I, \mathbb{G} \ll I \upharpoonright O)$$

and let refinement among components $X, Y \in \mathbb{B}$ be defined by

$$X \overset{\epsilon}{\sqsubseteq} Y \Leftrightarrow \mathcal{T}_{\mathbb{BD}}(X) \sqsubseteq \mathcal{T}_{\mathbb{BD}}(Y).$$

Refinement over components

For a standardized infinite list $\mathbb{G} \in \mathbb{T}^*$ of generic symbols, let

$$\mathscr{T}_{\mathbb{BD}} : \mathbb{B} \to \mathbb{D}$$

list truncation

map components to processes by

$$\mathscr{T}_{\mathbb{BD}}(I, O, B) = B(\mathbb{G} \mid I, \mathbb{G} \ll I \mid O)$$

and let refinement among components $X, Y \in \mathbb{B}$ be defined by

$$X \overset{\epsilon}{\sqsubseteq} Y \Leftrightarrow \mathscr{T}_{\mathbb{BD}}(X) \sqsubseteq \mathscr{T}_{\mathbb{BD}}(Y).$$

To-do list

Give a formal account of how components

- ✓ get connected into a network
- ✓ behave individually
- behave collectively when connected into a network

To-do list

Give a formal account of how components

- ✓ get connected into a network
- ✓ behave individually
- behave collectively when connected into a network

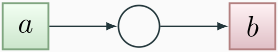
A blockoid $(\mathbb{B}, \mathbf{R}_{\mathbb{B}}, \mathbf{Z}_{\mathbb{B}}, \mathbf{I}_{\mathbb{B}})$ over circuit components

For the universe of primitive blocks

$$\mathbb{B} = \mathbb{N} \times \mathbb{N} \times (((\mathbb{T}^\star \times \mathbb{T}^\star) \to \mathbb{D}))$$

let $\mathbf{I}_{\mathbb{B}} = (I, O, B_{\mathbf{I}}) \in \mathbb{B}$ have $I = O = 1$ and

$$B_{\mathbf{I}}(\langle a \rangle, \langle b \rangle) = (\{a\}, \{b\}, N)$$



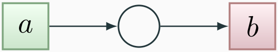
A blockoid $(\mathbb{B}, \mathbf{R}_{\mathbb{B}}, \mathbf{Z}_{\mathbb{B}}, \mathbf{I}_{\mathbb{B}})$ over circuit components

For the universe of primitive blocks

$$\mathbb{B} = \mathbb{N} \times \mathbb{N} \times (((\mathbb{T}^{\star} \times \mathbb{T}^{\star}) \rightarrow \mathbb{D}))$$

let $\mathbf{I}_{\mathbb{B}} = (I, O, B_{\mathbf{I}}) \in \mathbb{B}$ have $I = O = 1$ and

$$B_{\mathbf{I}}(\langle a \rangle, \langle b \rangle) = (\{a\}, \{b\}, (P, T, A, M, F))$$



A blockoid $(\mathbb{B}, \mathbf{R}_\mathbb{B}, \mathbf{Z}_\mathbb{B}, \mathbf{I}_\mathbb{B})$ over circuit components

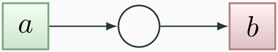
For the universe of primitive blocks

$$\mathbb{B} = \mathbb{N} \times \mathbb{N} \times (((\mathbb{T}^\star \times \mathbb{T}^\star) \to \mathbb{D}))$$

let $\mathbf{I}_\mathbb{B} = (I, O, B_\mathbf{I}) \in \mathbb{B}$ have $I = O = 1$ and

$$B_\mathbf{I}(\langle a \rangle, \langle b \rangle) = (\{a\}, \{b\}, (\{p\}, T, A, M, F))$$

for some arbitrary but fixed place p.



A blockoid $(\mathbb{B}, \mathbf{R}_{\mathbb{B}}, \mathbf{Z}_{\mathbb{B}}, \mathbf{I}_{\mathbb{B}})$ over circuit components

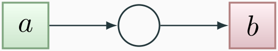
For the universe of primitive blocks

$$\mathbb{B} = \mathbb{N} \times \mathbb{N} \times (((\mathbb{T}^\star \times \mathbb{T}^\star) \to \mathbb{D}))$$

let $\mathbf{I}_{\mathbb{B}} = (I, O, B_{\mathbf{I}}) \in \mathbb{B}$ have $I = O = 1$ and

$$B_{\mathbf{I}}(\langle a \rangle, \langle b \rangle) = (\{a\}, \{b\}, (\{p\}, \{a, b\}, A, M, F))$$

for some arbitrary but fixed place p.



A blockoid $(\mathbb{B}, \mathbf{R}_\mathbb{B}, \mathbf{Z}_\mathbb{B}, \mathbf{I}_\mathbb{B})$ over circuit components

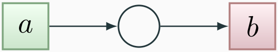
For the universe of primitive blocks

$$\mathbb{B} = \mathbb{N} \times \mathbb{N} \times (((\mathbb{T}^\star \times \mathbb{T}^\star) \to \mathbb{D}))$$

let $\mathbf{I}_\mathbb{B} = (I, O, B_\mathbf{I}) \in \mathbb{B}$ have $I = O = 1$ and

$$B_\mathbf{I}(\langle a \rangle, \langle b \rangle) = (\{a\}, \{b\}, (\{p\}, \{a, b\}, \{(a, p), (p, a)\}, M, F))$$

for some arbitrary but fixed place p.



A blockoid $(\mathbb{B}, \mathbf{R}_{\mathbb{B}}, \mathbf{Z}_{\mathbb{B}}, \mathbf{I}_{\mathbb{B}})$ over circuit components

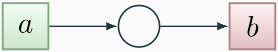
For the universe of primitive blocks

$$\mathbb{B} = \mathbb{N} \times \mathbb{N} \times (((\mathbb{T}^\star \times \mathbb{T}^\star) \to \mathbb{D}))$$

let $\mathbf{I}_{\mathbb{B}} = (I, O, B_{\mathbf{I}}) \in \mathbb{B}$ have $I = O = 1$ and

$$B_{\mathbf{I}}(\langle a \rangle, \langle b \rangle) = (\{a\}, \{b\}, (\{p\}, \{a, b\}, \{(a, p), (p, a)\}, \varnothing, F))$$

for some arbitrary but fixed place p.



A blockoid $(\mathbb{B}, \mathbf{R}_{\mathbb{B}}, \mathbf{Z}_{\mathbb{B}}, \mathbf{I}_{\mathbb{B}})$ over circuit components

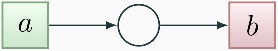
For the universe of primitive blocks

$$\mathbb{B} = \mathbb{N} \times \mathbb{N} \times (((\mathbb{T}^\star \times \mathbb{T}^\star) \to \mathbb{D}))$$

let $\mathbf{I}_{\mathbb{B}} = (I, O, B_{\mathbf{I}}) \in \mathbb{B}$ have $I = O = 1$ and

$$B_{\mathbf{I}}(\langle a \rangle, \langle b \rangle) = (\{a\}, \{b\}, (\{p\}, \{a, b\}, \{(a, p), (p, a)\}, \varnothing, \varnothing))$$

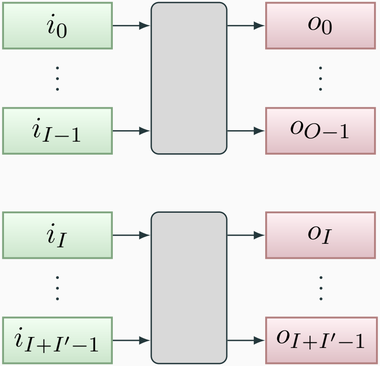
for some arbitrary but fixed place p.



A blockoid $(\mathbb{B}, \mathbf{R}_\mathbb{B}, \mathbf{Z}_\mathbb{B}, \mathbf{I}_\mathbb{B})$ over circuit components

Let $\mathbf{R}_\mathbb{B}((I, O, B), (I', O', B')) = (I + I', O + O', B_\mathbf{R})$ with

$$B_\mathbf{R}(i, o) = \mathsf{par}\ (B(i \shortmid I, o \shortmid O), B'(i \ll I, o \ll O))$$

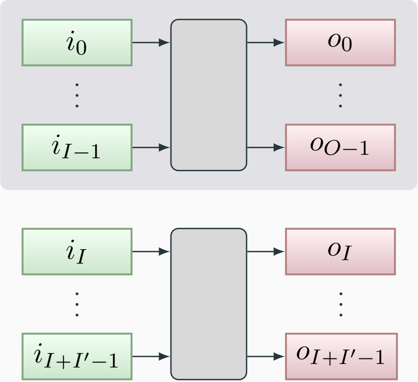


A blockoid $(\mathbb{B}, \mathbf{R}_\mathbb{B}, \mathbf{Z}_\mathbb{B}, \mathbf{I}_\mathbb{B})$ over circuit components

Let $\mathbf{R}_\mathbb{B}((I, O, B), (I', O', B')) = (I + I', O + O', B_\mathbf{R})$ with

$$B_\mathbf{R}(i, o) = \mathsf{par}\ (B(i \restriction I, o \restriction O), B'(i \ll I, o \ll O))$$

$B(i \restriction I, o \restriction I)$

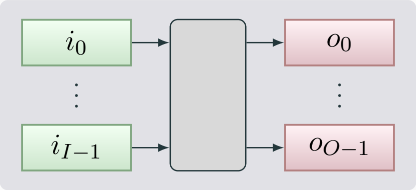


A blockoid $(\mathbb{B}, \mathbf{R}_\mathbb{B}, \mathbf{Z}_\mathbb{B}, \mathbf{I}_\mathbb{B})$ over circuit components

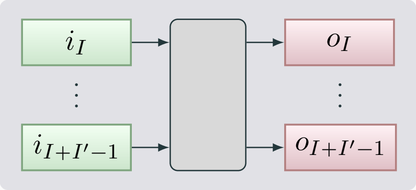
Let $\mathbf{R}_\mathbb{B}((I, O, B), (I', O', B')) = (I + I', O + O', B_\mathbf{R})$ with

$$B_\mathbf{R}(i, o) = \textsf{par}\ (B(i \shortmid I, o \shortmid O), B'(i \ll I, o \ll O))$$

$B(i \shortmid I, o \shortmid I)$



$B'(i \ll I, o \ll I)$

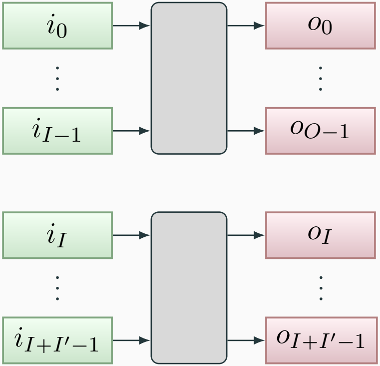


A blockoid $(\mathbb{B}, \mathbf{R}_{\mathbb{B}}, \mathbf{Z}_{\mathbb{B}}, \mathbf{I}_{\mathbb{B}})$ over circuit components

Let $\mathbf{R}_{\mathbb{B}}((I, O, B), (I', O', B')) = (I + I', O + O', B_{\mathbf{R}})$ with

$$B_{\mathbf{R}}(i, o) = \mathsf{par}\ (B(i \upharpoonright I, o \upharpoonright O), B'(i \ll I, o \ll O))$$

Parallel composition of DI processes

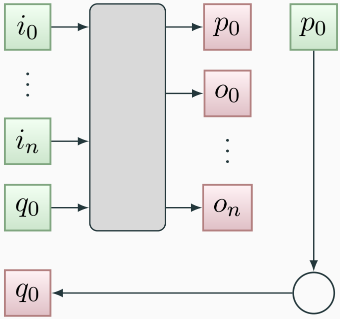


A blockoid $(\mathbb{B}, \mathbf{R}_{\mathbb{B}}, \mathbf{Z}_{\mathbb{B}}, \mathbf{I}_{\mathbb{B}})$ over circuit components

Let $\mathbf{Z}_{\mathbb{B}}(I, O, B) = (I - 1, O - 1, B_{\mathbf{Z}}) \in \mathbb{B}$ with

$$B_{\mathbf{Z}}(i, o) = \mathsf{par}\ (B(i \shortparallel q, p \shortparallel o), B_{\mathbf{I}}(p, q))$$

for arbitrary distinct $p, q \in \mathbb{T}^1$ disjoint from i and o.



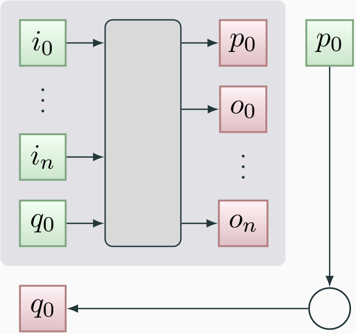
A blockoid $(\mathbb{B}, \mathbf{R}_\mathbb{B}, \mathbf{Z}_\mathbb{B}, \mathbf{I}_\mathbb{B})$ over circuit components

Let $\mathbf{Z}_\mathbb{B}(I, O, B) = (I-1, O-1, B_\mathbf{Z}) \in \mathbb{B}$ with

$$B_\mathbf{Z}(i, o) = \mathsf{par}\ (B(i \shortparallel q, p \shortparallel o), B_\mathbf{I}(p, q))$$

for arbitrary distinct $p, q \in \mathbb{T}^1$ disjoint from i and o.

$B(i \shortparallel q, p \shortparallel o)$

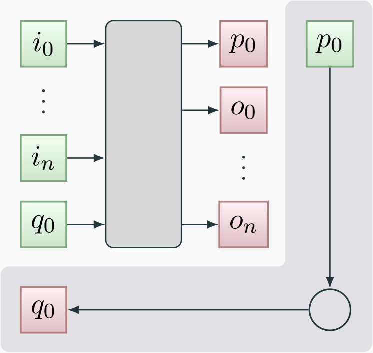


A blockoid $(\mathbb{B}, \mathbf{R}_{\mathbb{B}}, \mathbf{Z}_{\mathbb{B}}, \mathbf{I}_{\mathbb{B}})$ over circuit components

Let $\mathbf{Z}_{\mathbb{B}}(I, O, B) = (I - 1, O - 1, B_{\mathbf{Z}}) \in \mathbb{B}$ with

$$B_{\mathbf{Z}}(i, o) = \mathsf{par}\ (B(i \shortparallel q, p \shortparallel o), B_{\mathbf{I}}(p, q))$$

for arbitrary distinct $p, q \in \mathbb{T}^1$ disjoint from i and o.



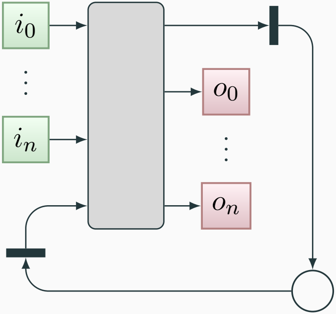
$B_{\mathbf{I}}(p, q)$

A blockoid $(\mathbb{B}, \mathbf{R}_{\mathbb{B}}, \mathbf{Z}_{\mathbb{B}}, \mathbf{I}_{\mathbb{B}})$ over circuit components

Let $\mathbf{Z}_{\mathbb{B}}(I, O, B) = (I - 1, O - 1, B_{\mathbf{Z}}) \in \mathbb{B}$ with

$$B_{\mathbf{Z}}(i, o) = \mathsf{par}\ (B(i \shortparallel q, p \shortparallel o), B_{\mathbf{I}}(p, q))$$

for arbitrary distinct $p, q \in \mathbb{T}^1$ disjoint from i and o.



To-do list

Give a formal account of how components

- ✓ get connected into a network
- ✓ behave individually
- ✓ behave collectively when connected into a network

To-do list

Give a formal account of how components

- ✔ get connected into a network
- ✔ behave individually
- ✔ behave collectively when connected into a network

Now what ?

- A network needs to be physically implemented somehow.
- Technology mapping tools need it in netlist form.
- How to ensure the netlist matches the block expression ?

An abstract representation

Idea !

Use an intermediate source with two possible targets.

- Let $\mathbb{H}$ denote a universe of *hierarchical blocks*.
- Let $\mathbb{L}$ denote a universe of netlists.
- Transform $\mathbb{B} \xleftarrow{\mathcal{T}_{\mathbb{HB}}} \mathbb{H} \xrightarrow{\mathcal{T}_{\mathbb{HL}}} \mathbb{L}$ in either direction.
- Make the transformations $\mathcal{T}_{\mathbb{HB}}$ and $\mathcal{T}_{\mathbb{HL}}$ simple and obvious.

An abstract representation

Idea !

Use an intermediate source with two possible targets.

- Let $\mathbb{H}$ denote a universe of *hierarchical blocks*.
- Let $\mathbb{L}$ denote a universe of netlists.
- Transform $\mathbb{B} \xleftarrow{\mathcal{T}_{\mathbb{HB}}} \mathbb{H} \xrightarrow{\mathcal{T}_{\mathbb{HL}}} \mathbb{L}$ in either direction.
- Make the transformations $\mathcal{T}_{\mathbb{HB}}$ and $\mathcal{T}_{\mathbb{HL}}$ simple and obvious.

Structure-preserving maps between blockoids over $\mathbb{H}$, $\mathbb{B}$, and $\mathbb{L}$?

A blockoid $(\mathbb{H}, \mathbf{R}, \mathbf{Z}, \mathbf{I})$ over hierarchical blocks

Define the universe of hierarchical blocks

$$\mathbb{H} = \mathbb{B} \cup \mathbb{H}^*$$

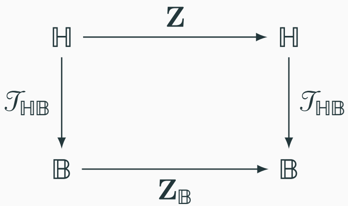
and let nested lists encode them by block combinators

$$\begin{array}{lcl}
\mathbf{I} & = & \mathbf{I}_{\mathbb{B}} \\
\mathbf{Z}(x) & = & \langle x \rangle \\
\mathbf{R}(\langle x \rangle, \langle y \rangle) & = & \langle \langle x \rangle, \langle y \rangle \rangle \\
\mathbf{R}(\langle x \rangle, Y) & = & \langle \langle x \rangle \rangle \shortparallel Y \\
\mathbf{R}(X, \langle y \rangle) & = & X \shortparallel \langle \langle y \rangle \rangle \\
\mathbf{R}(X, Y) & = & X \shortparallel Y
\end{array}$$

for all $x, y \in \mathbb{H}$ and $X, Y \in \mathbb{H}^*$ with $|X|, |Y| > 1$.

Transformation from hierarchical to primitive blocks

A transformation $\mathcal{T}_{\mathbb{HB}} : \mathbb{H} \to \mathbb{B}$ satisfying



determines a refinement relation

$$X \overset{\alpha}{\sqsubseteq} Y \Leftrightarrow \mathcal{T}_{\mathbb{HB}}\, X \overset{\epsilon}{\sqsubseteq} \mathcal{T}_{\mathbb{HB}}\, Y$$

and hence an extensional semantics for all $X, Y \in \mathbb{H}$.

Transformation from hierarchical to primitive blocks

A transformation $\mathcal{T}_{\mathbb{HB}} : \mathbb{H} \rightarrow \mathbb{B}$ satisfying

$$
\begin{array}{ccc}
\mathbb{H} \times \mathbb{H} & \xrightarrow{\quad \mathbf{R} \quad} & \mathbb{H} \\
{\scriptstyle \lambda(x,y).\ (\mathcal{T}_{\mathbb{HB}}\, x,\, \mathcal{T}_{\mathbb{HB}}\, y)} \big\downarrow & & \big\downarrow {\scriptstyle \mathcal{T}_{\mathbb{HB}}} \\
\mathbb{B} \times \mathbb{B} & \xrightarrow[\quad \mathbf{R}_{\mathbb{B}} \quad]{} & \mathbb{B}
\end{array}
$$

determines a refinement relation

$$X \overset{\alpha}{\sqsubseteq} Y \Leftrightarrow \mathcal{T}_{\mathbb{HB}}\, X \overset{\epsilon}{\sqsubseteq} \mathcal{T}_{\mathbb{HB}}\, Y$$

and hence an extensional semantics for all $X, Y \in \mathbb{H}$.

Transformation from hierarchical to primitive blocks

A transformation $\mathscr{T}_{\mathbb{HB}} : \mathbb{H} \to \mathbb{B}$ satisfying

$$\mathscr{T}_{\mathbb{HB}}(h) = \begin{cases} h & \text{if } h \in \mathbb{B} \\ \mathbf{Z}_{\mathbb{B}}\, \mathscr{T}_{\mathbb{HB}}\, h_0 & \text{if } h \in \mathbb{H}^1 \\ (F\ \mathbf{R}_{\mathbb{B}})\, \mathscr{T}^{\star}_{\mathbb{HB}}\, h & \text{otherwise} \end{cases}$$

determines a refinement relation

$$X \overset{\alpha}{\sqsubseteq} Y \Leftrightarrow \mathscr{T}_{\mathbb{HB}}\, X \overset{\epsilon}{\sqsubseteq} \mathscr{T}_{\mathbb{HB}}\, Y$$

and hence an extensional semantics for all $X, Y \in \mathbb{H}$.

Transformation from hierarchical to primitive blocks

A transformation $\mathcal{T}_{\mathbb{HB}} : \mathbb{H} \to \mathbb{B}$ satisfying

map over a list

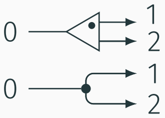
$$\mathcal{T}_{\mathbb{HB}}(h) = \begin{cases} h & \text{if } h \in \mathbb{B} \\ \mathbf{Z}_{\mathbb{B}}\, \mathcal{T}_{\mathbb{HB}}\, h_0 & \text{if } h \in \mathbb{H}^1 \\ (F\ \mathbf{R}_{\mathbb{B}})\, \mathcal{T}^{\star}_{\mathbb{HB}}\, h & \text{otherwise} \end{cases}$$

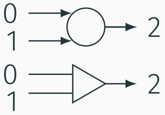
determines a refinement relation

$$X \overset{\alpha}{\sqsubseteq} Y \Leftrightarrow \mathcal{T}_{\mathbb{HB}}\, X \overset{\epsilon}{\sqsubseteq} \mathcal{T}_{\mathbb{HB}}\, Y$$

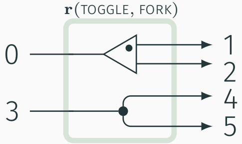
and hence an extensional semantics for all $X, Y \in \mathbb{H}$.

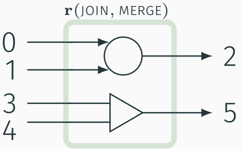
Netlists from blockoid operators



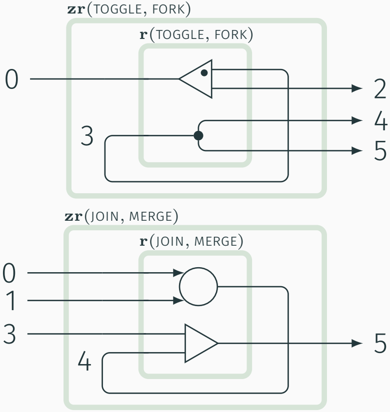


Netlists from blockoid operators

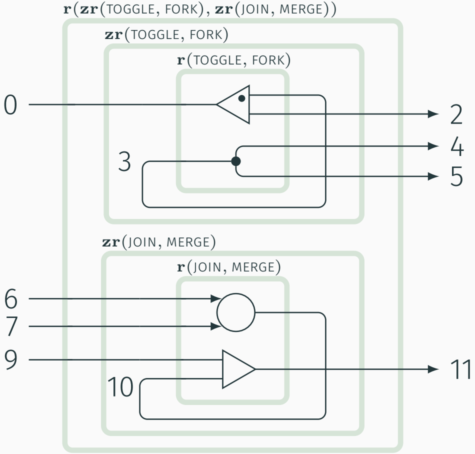




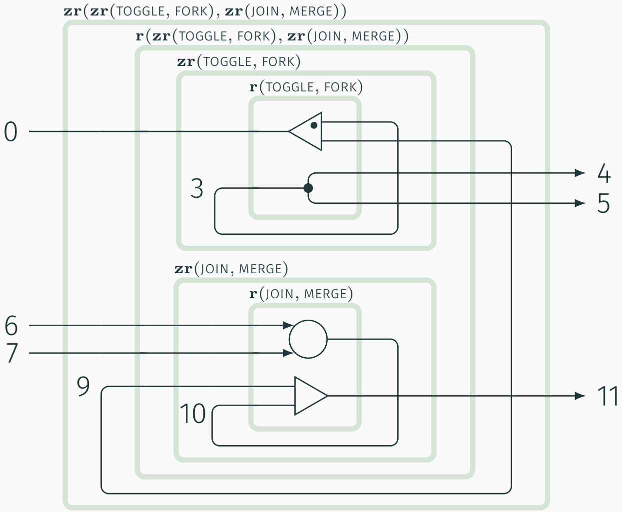
Netlists from blockoid operators



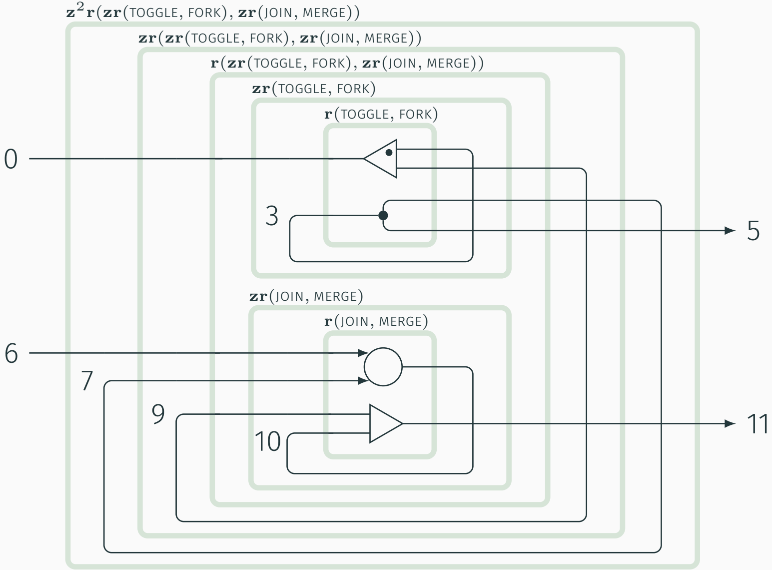
Netlists from blockoid operators

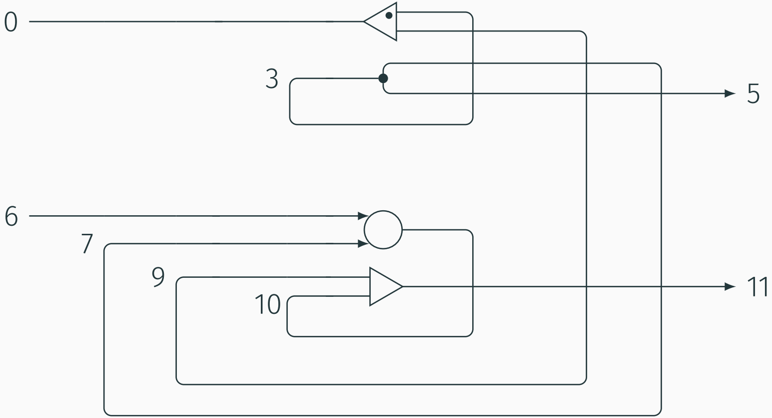


Netlists from blockoid operators

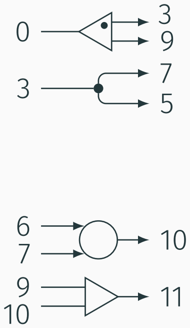


Netlists from blockoid operators

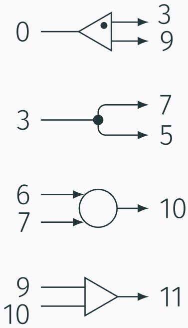



0
3
5
6
7
9
10
11

Netlists from blockoid operators



Netlists from blockoid operators



$(\langle 0 \rangle, \langle 3, 9 \rangle, \textsc{toggle})$

$(\langle 3 \rangle, \langle 7, 5 \rangle, \textsc{fork})$

$(\langle 6, 7 \rangle, \langle 10 \rangle, \textsc{join})$

$(\langle 9, 10 \rangle, \langle 11 \rangle, \textsc{merge})$

Netlists from blockoid operators

Netlists boil down to members of $(\mathbb{N}^* \times \mathbb{N}^* \times \mathbb{B})^*$ such as

$$\begin{aligned}\langle\quad & \\ & (\langle 0\rangle, && \langle 3,9\rangle, && \textsc{toggle}), \\ & (\langle 3\rangle, && \langle 7,5\rangle, && \textsc{fork}), \\ & (\langle 6,7\rangle, && \langle 10\rangle, && \textsc{join}), \\ & (\langle 9,10\rangle, && \langle 11\rangle, && \textsc{merge})\rangle\end{aligned}$$

A blockoid $(\mathbb{L}, \mathbf{R}_{\mathbb{L}}, \mathbf{Z}_{\mathbb{L}}, \mathbf{I}_{\mathbb{L}})$ over netlists

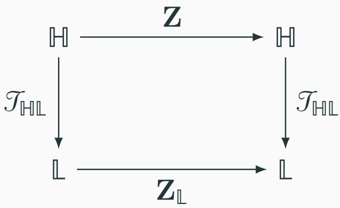
Define the universe of netlists

$$\mathbb{L} = (\mathbb{N}^* \times \mathbb{N}^* \times \mathbb{B})^*$$

with

$$\mathbf{I}_{\mathbb{L}} = \big\langle (\langle 0 \rangle, \langle 1 \rangle, \mathbf{I}_{\mathbb{B}}) \big\rangle$$

and $\mathbf{Z}_{\mathbb{L}}, \mathbf{R}_{\mathbb{L}}$ satisfying



A blockoid $(\mathbb{L}, \mathbf{R}_{\mathbb{L}}, \mathbf{Z}_{\mathbb{L}}, \mathbf{I}_{\mathbb{L}})$ over netlists

Define the universe of netlists

$$\mathbb{L} = (\mathbb{N}^* \times \mathbb{N}^* \times \mathbb{B})^*$$

with

$$\mathbf{I}_{\mathbb{L}} = \Big\langle \big(\langle 0 \rangle, \langle 1 \rangle, \mathbf{I}_{\mathbb{B}} \big) \Big\rangle$$

and $\mathbf{Z}_{\mathbb{L}}, \mathbf{R}_{\mathbb{L}}$ satisfying

$$\begin{array}{ccc}
\mathbb{H} \times \mathbb{H} & \xrightarrow{\quad \mathbf{R} \quad} & \mathbb{H} \\
{\scriptstyle \lambda(x,y).\ (\mathscr{T}_{\mathbb{H}\mathbb{L}}\, x, \mathscr{T}_{\mathbb{H}\mathbb{L}}\, y)} \downarrow & & \downarrow {\scriptstyle \mathscr{T}_{\mathbb{H}\mathbb{L}}} \\
\mathbb{L} \times \mathbb{L} & \xrightarrow[\quad \mathbf{R}_{\mathbb{L}} \quad]{} & \mathbb{L}
\end{array}$$

A blockoid $(\mathbb{L}, \mathbf{R}_{\mathbb{L}}, \mathbf{Z}_{\mathbb{L}}, \mathbf{I}_{\mathbb{L}})$ over netlists

Define the universe of netlists

$$\mathbb{L} = (\mathbb{N}^* \times \mathbb{N}^* \times \mathbb{B})^*$$

with

$$\mathbf{I}_{\mathbb{L}} = \left\langle (\langle 0 \rangle, \langle 1 \rangle, \mathbf{I}_{\mathbb{B}}) \right\rangle$$

and $\mathbf{Z}_{\mathbb{L}}, \mathbf{R}_{\mathbb{L}}$ satisfying

$$\mathcal{T}_{\mathbb{HL}}(h) = \begin{cases} (\lambda(I, O, B).\, \langle (\iota_I, \iota_O^I, h) \rangle)\ h & \text{if } h \in \mathbb{B} \\ \mathbf{Z}_{\mathbb{L}}\, \mathcal{T}_{\mathbb{HL}}\, h_0 & \text{if } h \in \mathbb{H}^1 \\ (\digamma\, \mathbf{R}_{\mathbb{L}})\, \mathcal{T}^{\star}_{\mathbb{HL}}\, h & \text{otherwise} \end{cases}$$

Define the universe of netlists

$$\mathbb{L} = (\mathbb{N}^* \times \mathbb{N}^* \times \mathbb{B})^*$$

with

$$\mathbf{I}_{\mathbb{L}} = \big\langle (\langle 0 \rangle, \langle 1 \rangle, \mathbf{I}_{\mathbb{B}}) \big\rangle$$

and $\mathbf{Z}_{\mathbb{L}}, \mathbf{R}_{\mathbb{L}}$ satisfying

$\langle 0 \ldots I - 1 \rangle$

$\langle I \ldots I + O - 1 \rangle$

$$\mathcal{T}_{\mathbb{HL}}(h) = \begin{cases} (\lambda(I, O, B).\, \langle(\iota_I, \iota_O^I, h)\rangle)\ h & \text{if } h \in \mathbb{B} \\ \mathbf{Z}_{\mathbb{L}}\ \mathcal{T}_{\mathbb{HL}}\ h_0 & \text{if } h \in \mathbb{H}^1 \\ (F\ \mathbf{R}_{\mathbb{L}})\ \mathcal{T}^{\star}_{\mathbb{HL}}\ h & \text{otherwise} \end{cases}$$

A blockoid $(\mathbb{L}, \mathbf{R}_\mathbb{L}, \mathbf{Z}_\mathbb{L}, \mathbf{I}_\mathbb{L})$ over netlists

For input terminal numbers and output terminal numbers

$$i(x) = \bigcup_{(I,O,B) \in \mathscr{R}(x)} \mathscr{R}(I) \qquad o(x) = \bigcup_{(I,O,B) \in \mathscr{R}(x)} \mathscr{R}(O)$$

associated with a netlist $x \in \mathbb{L}$, we have

all terminals	$i(x) \cup o(x)$
all external inputs	$i(x) - o(x)$
all external outputs	$o(x) - i(x)$
first external output	$\min\big(o(x) - i(x)\big)$
last external input	$\max\big(i(x) - o(x)\big)$

A rewrite rule for input terminal numbers

$$w_z(x) = \lambda t.\ \begin{cases} \min(o(x) - i(x)) & \text{if } t = \max(i(x) - o(x)) \\ t & \text{otherwise} \end{cases}$$

specifies the operator $\mathbf{Z}_\mathbb{L} : \mathbb{L} \to \mathbb{L}$

$$\mathbf{Z}_\mathbb{L}(x) = (\lambda(I, O, B).\ ((w_z\, x)^\star\, I, O, B))^\star\, x$$

A blockoid $(\mathbb{L}, \mathbf{R}_{\mathbb{L}}, \mathbf{Z}_{\mathbb{L}}, \mathbf{I}_{\mathbb{L}})$ over netlists

A rewrite rule taking $t \in \mathbb{N}$ to a number outside $i(x) \cup o(x)$

$$w_r(x) = \lambda t.\ t + 1 + \max\big(i(x) \cup o(x)\big)$$

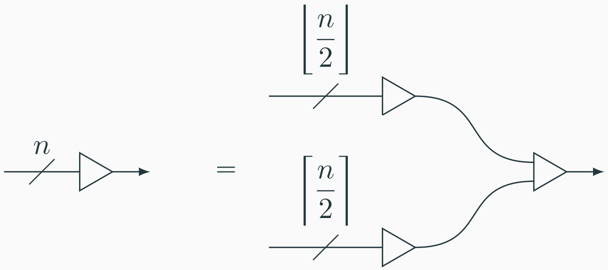
specifies the operator $\mathbf{R}_{\mathbb{L}} : \mathbb{L} \times \mathbb{L} \to \mathbb{L}$

$$\mathbf{R}_{\mathbb{L}}(x, y) = x \shortparallel (\lambda(I, O, B).\ ((w_r\, x)^{\star}\, I, (w_r\, x)^{\star}\, O, B))^{\star}\, y$$

Taking it for a spin

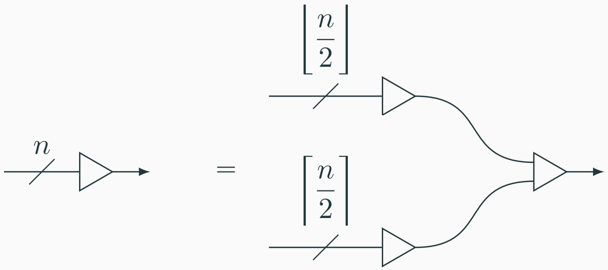
Multiway merge

$$f(n) = \begin{cases} \mathbf{I} & \text{if } n = 1 \\ \mathbf{Z}^2\mathbf{R}\big(\mathbf{R}\big(f\lfloor n/2 \rfloor, f\lceil n/2 \rceil\big), \textsc{merge}\big) & \text{otherwise} \end{cases}$$



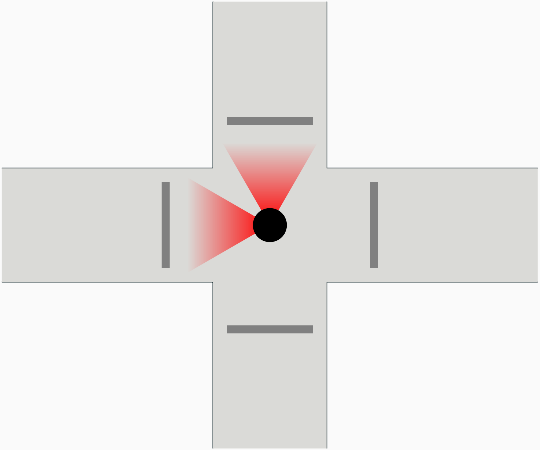
Multiway merge

$$f(n) = \begin{cases} \mathbf{I} & \text{if } n = 1 \\ \mathbf{Z}^2\mathbf{R}\big(\mathbf{R}\big(f\lfloor n/2\rfloor, f\lceil n/2\rceil\big), \textsc{merge}\big) & \text{otherwise} \end{cases}$$

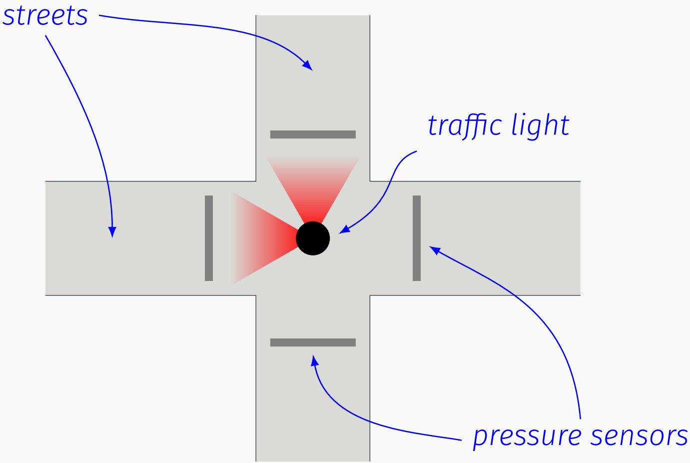


Too easy ?

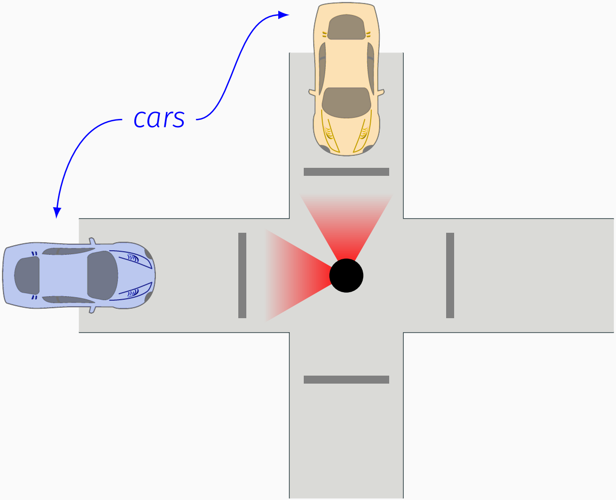
Traffic control



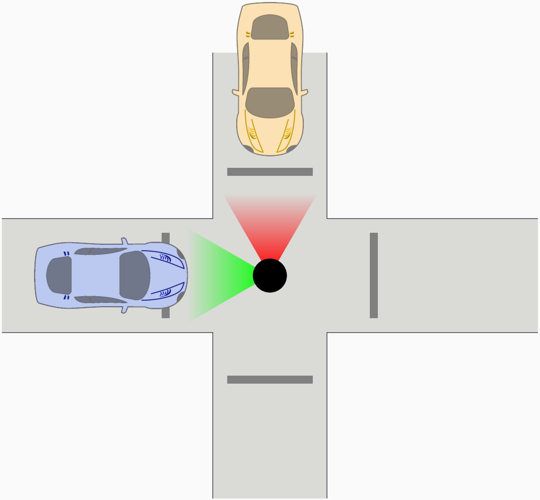
Traffic control



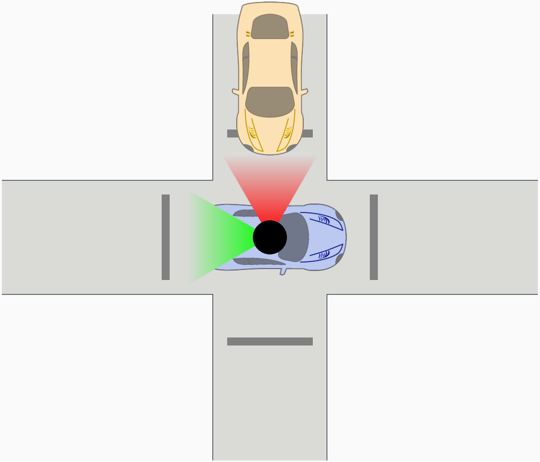
Traffic control



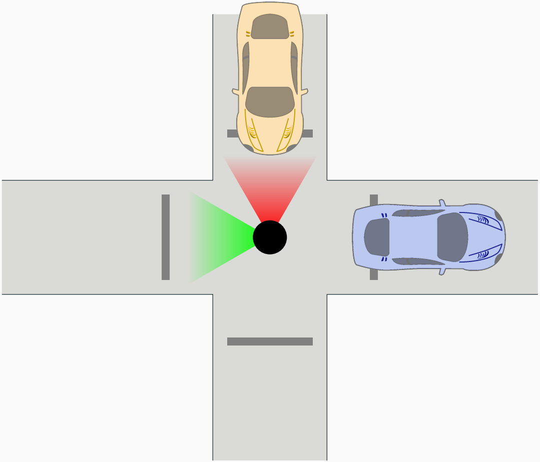
Traffic control



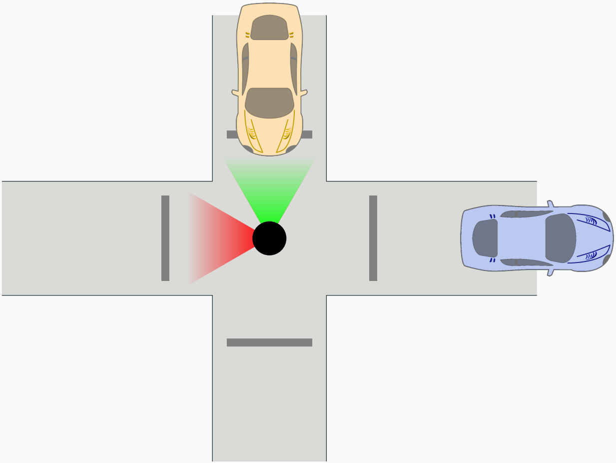
Traffic control



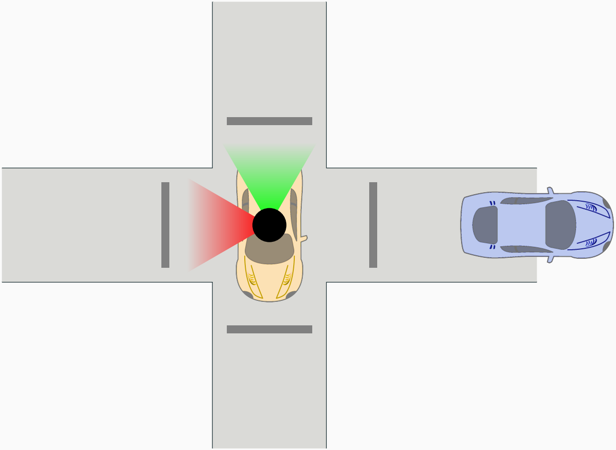
Traffic control



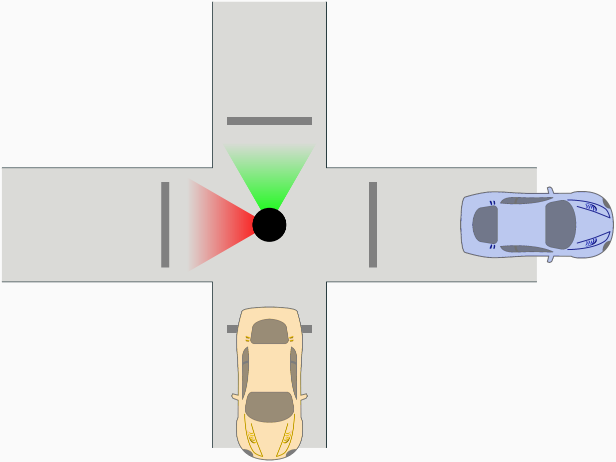
Traffic control



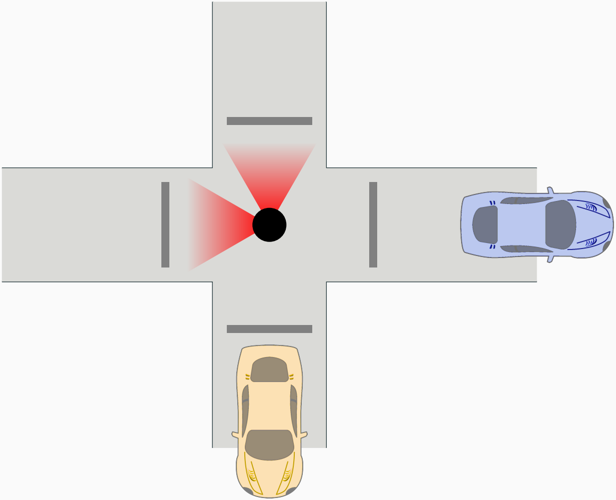
Traffic control



Traffic control

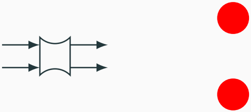


Traffic control

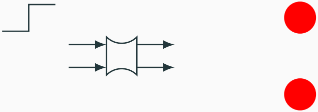


Arbiter circuits

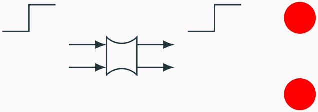
Arbiter circuits



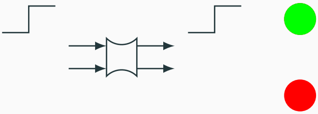
Arbiter circuits



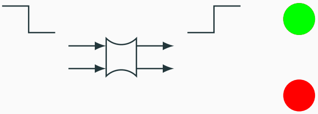
Arbiter circuits



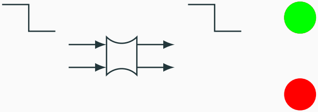
Arbiter circuits



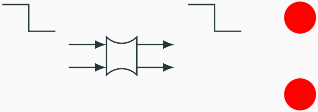
Arbiter circuits



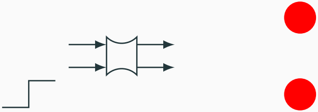
Arbiter circuits



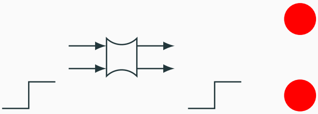
Arbiter circuits



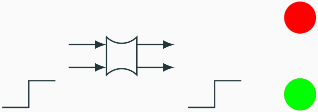
Arbiter circuits



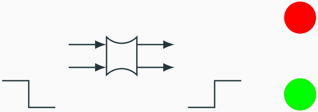
Arbiter circuits



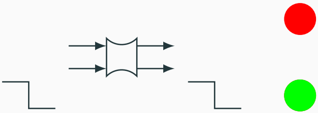
Arbiter circuits



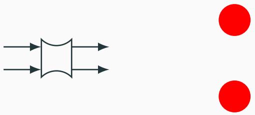
Arbiter circuits



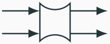
Arbiter circuits



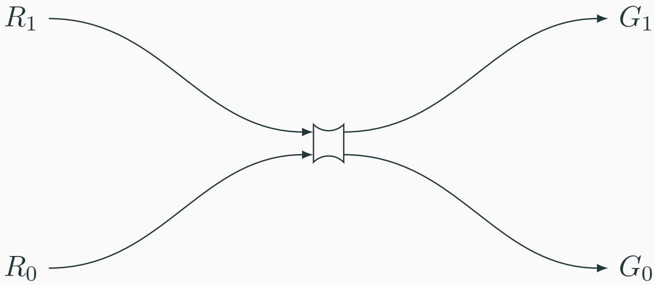
Arbiter circuits



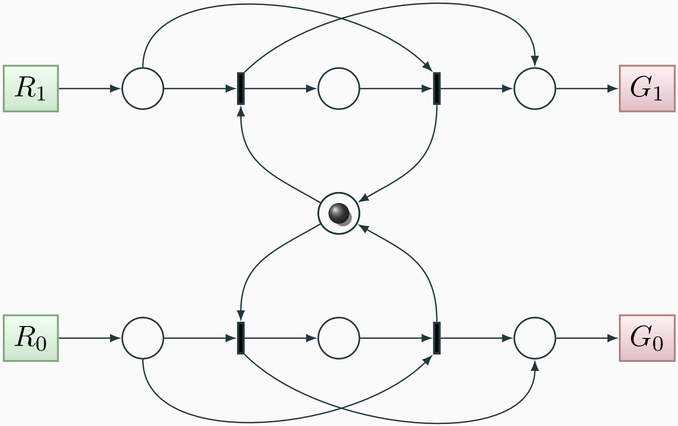
Arbiter circuits



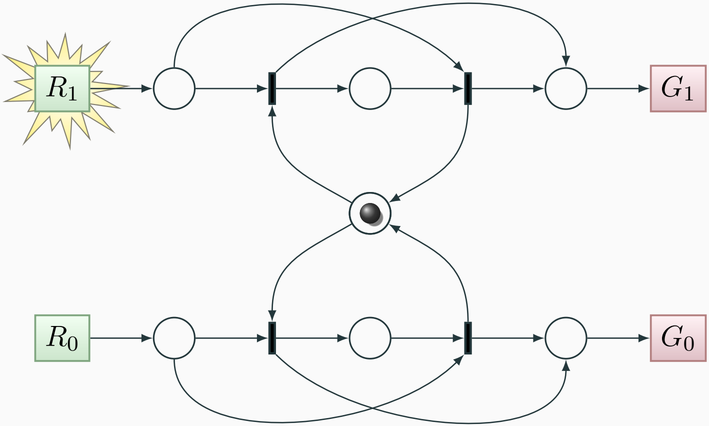
Arbiter circuits



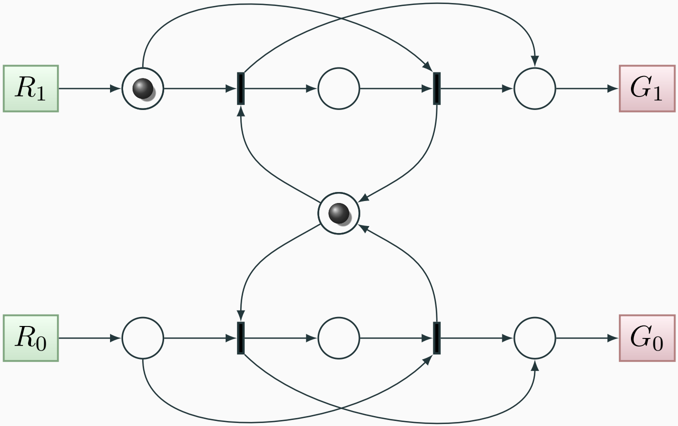
Arbiter circuits



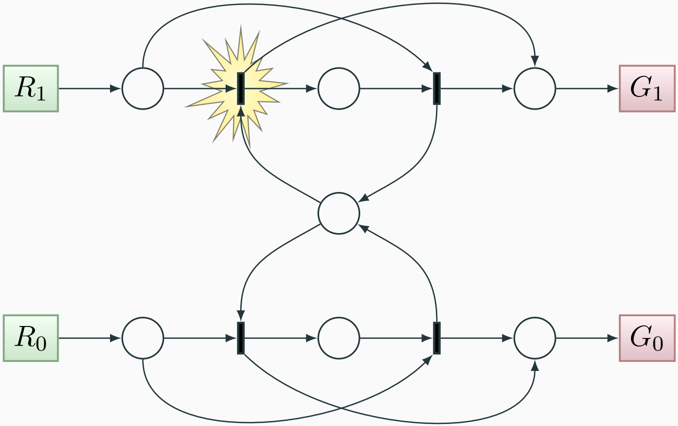
Arbiter circuits



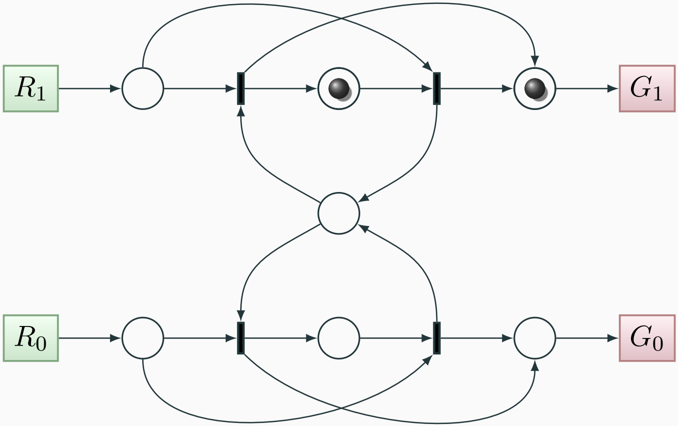
Arbiter circuits



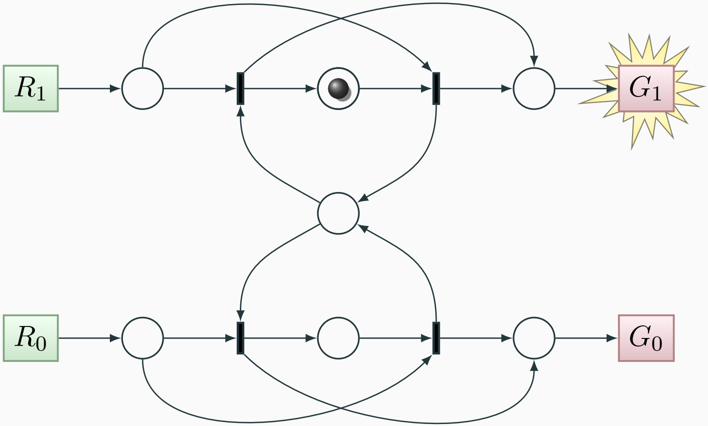
Arbiter circuits



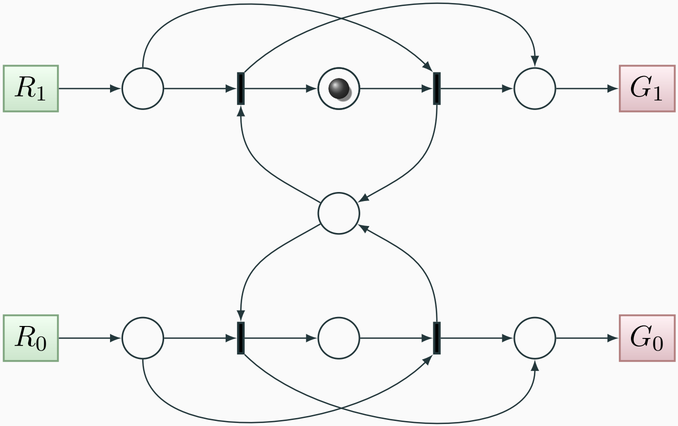
Arbiter circuits



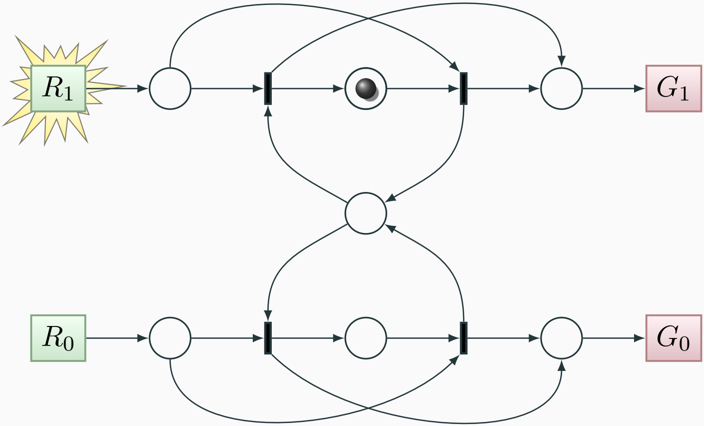
Arbiter circuits

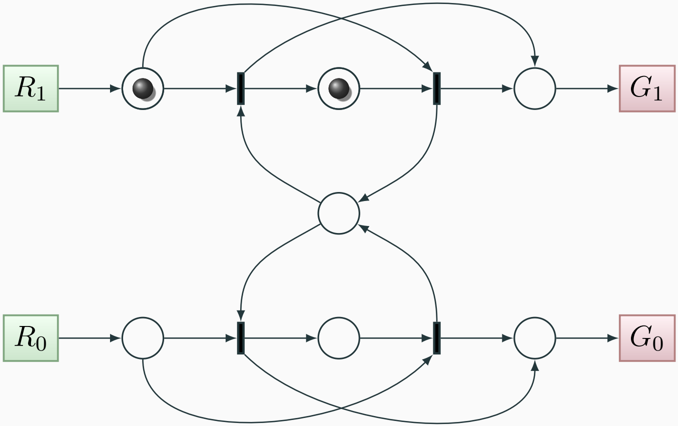


Arbiter circuits

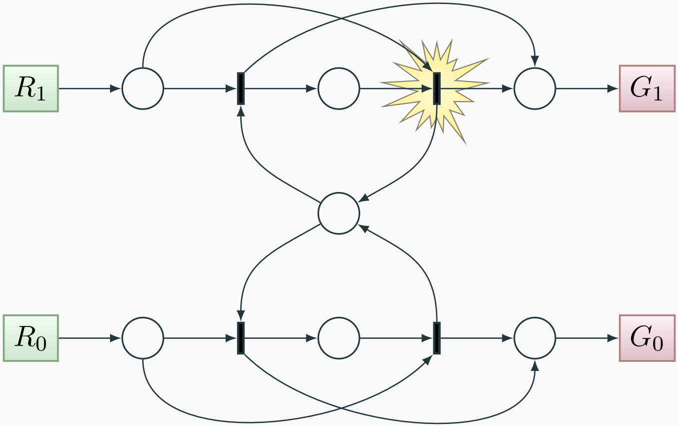


Arbiter circuits

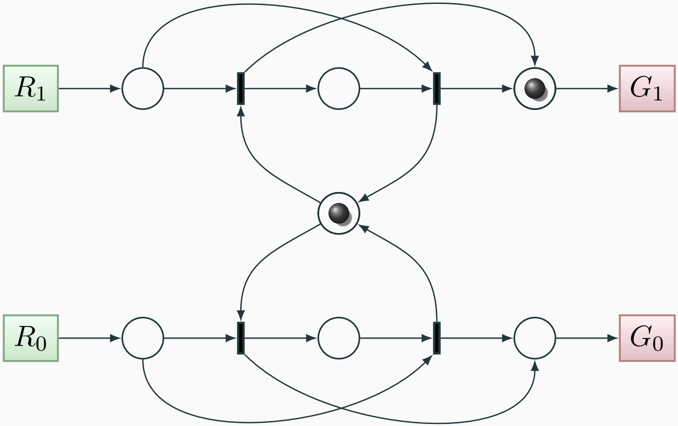



R_1
G_1
R_0
G_0

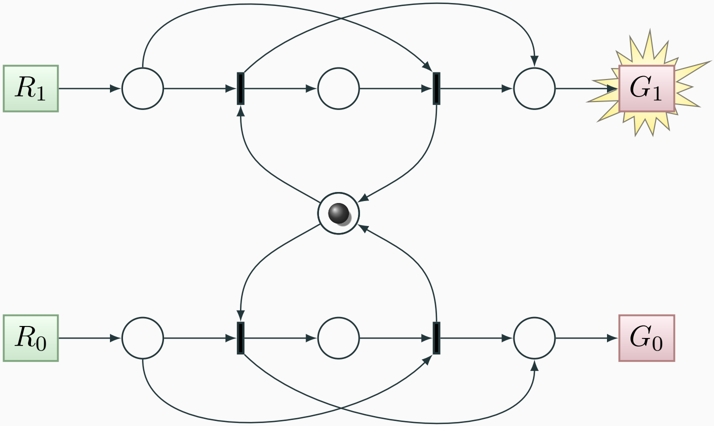
Arbiter circuits



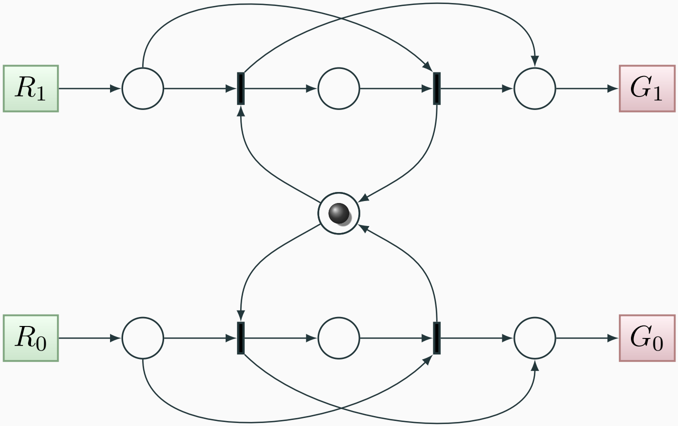
Arbiter circuits



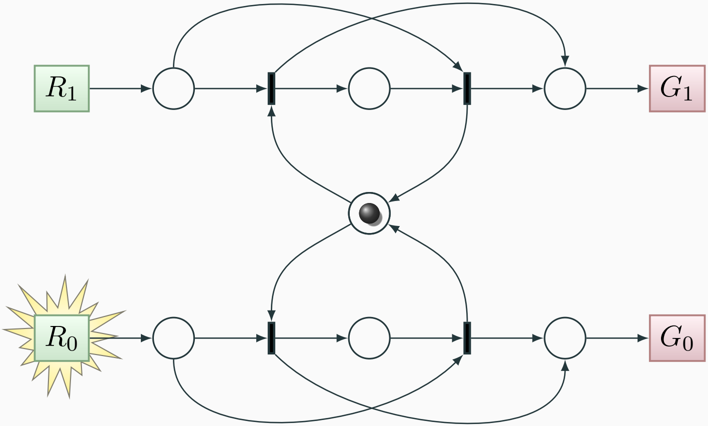
Arbiter circuits



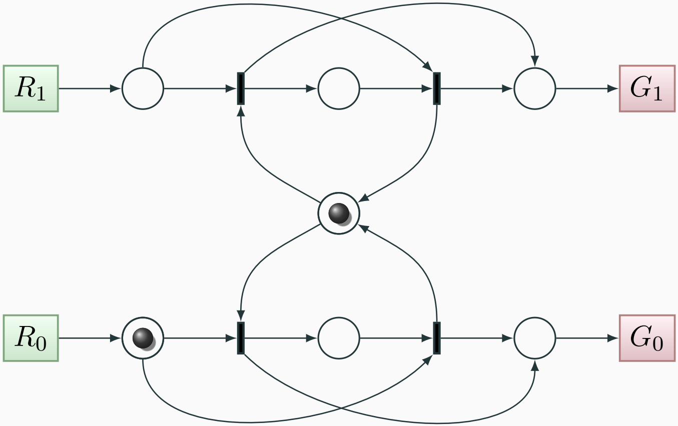
Arbiter circuits



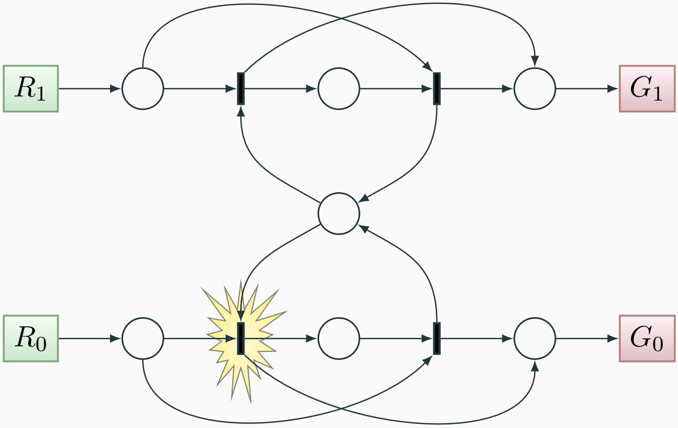
Arbiter circuits



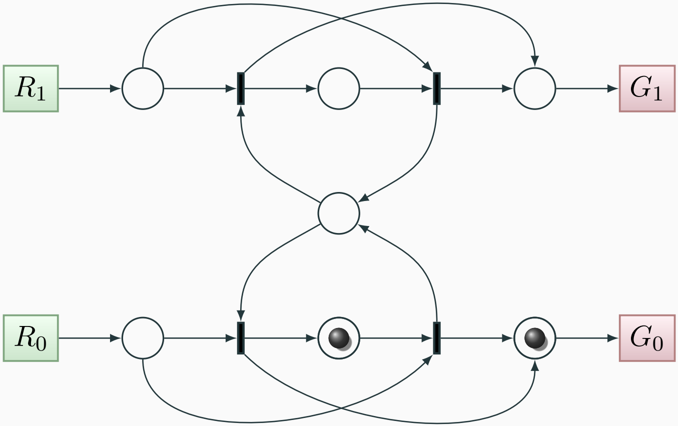
Arbiter circuits



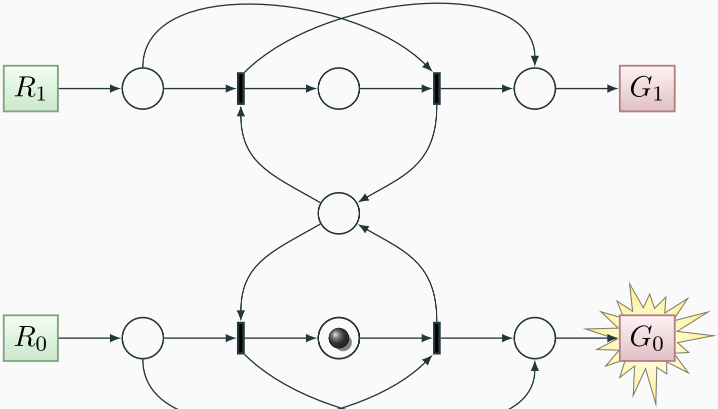
Arbiter circuits



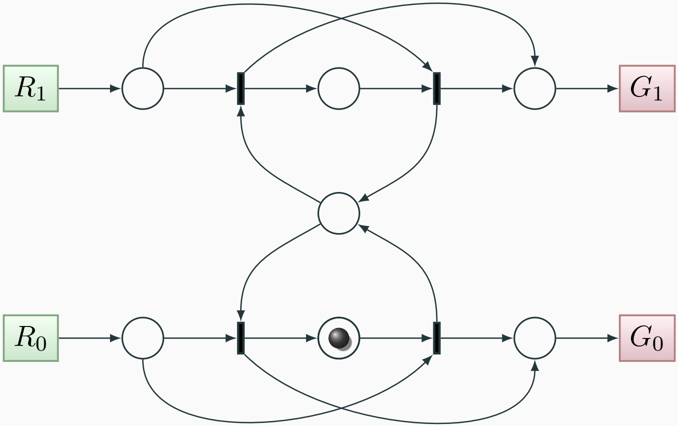
Arbiter circuits



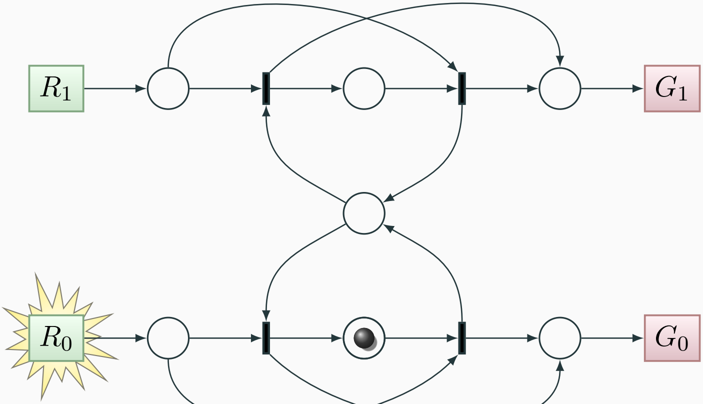
Arbiter circuits



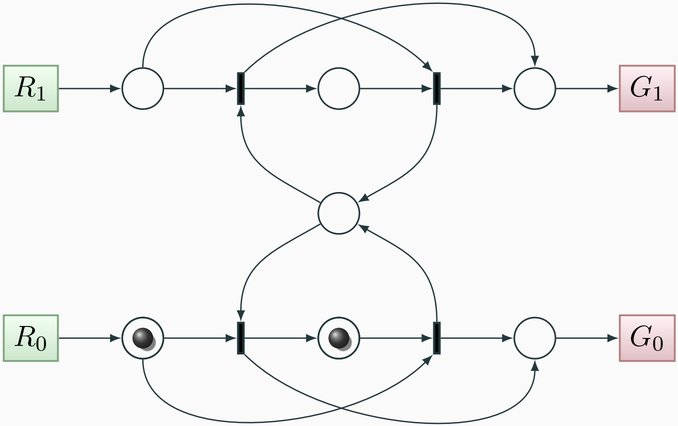
Arbiter circuits



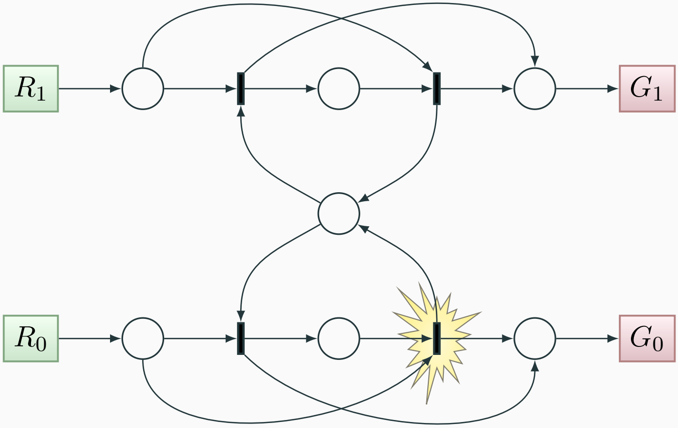
Arbiter circuits



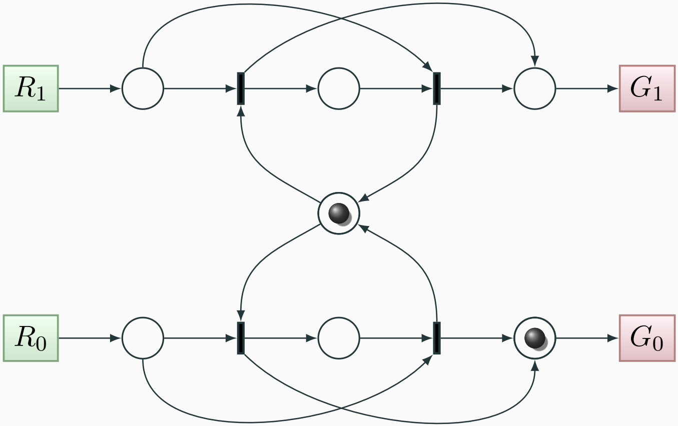
Arbiter circuits



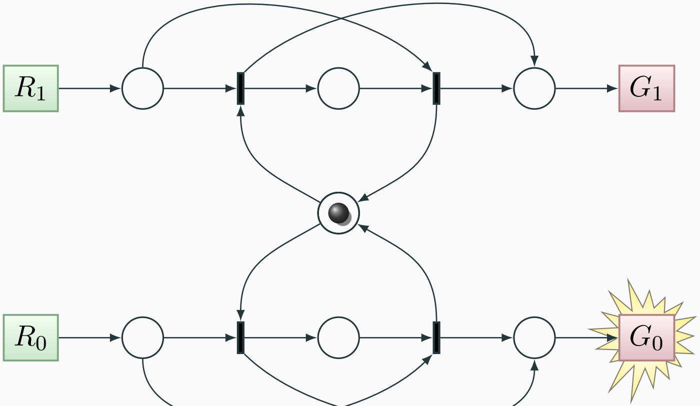
Arbiter circuits



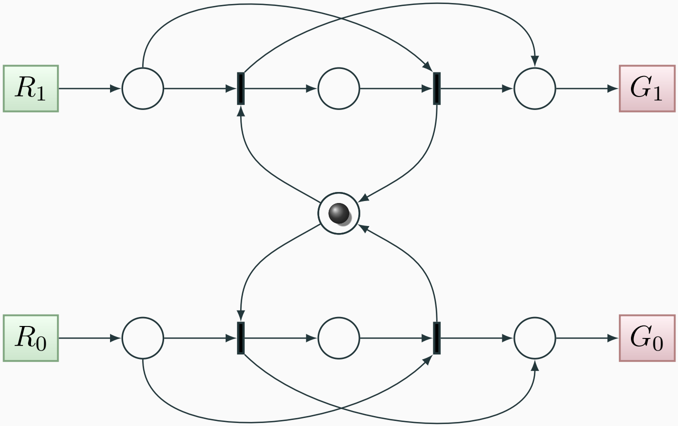
Arbiter circuits



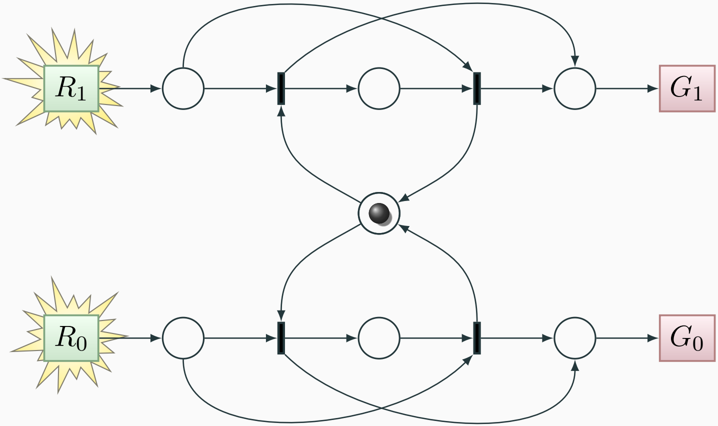
Arbiter circuits



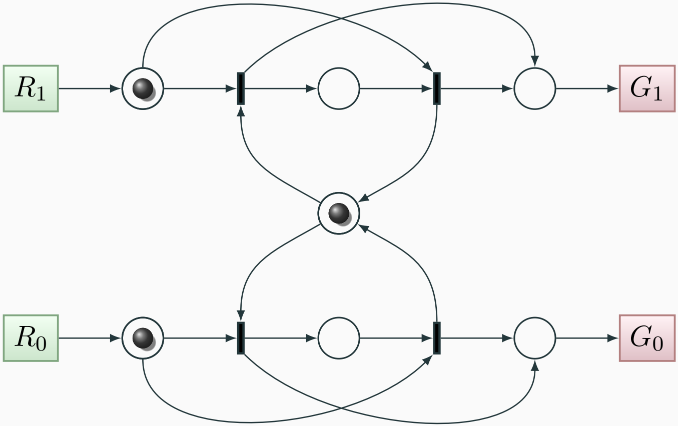
Arbiter circuits



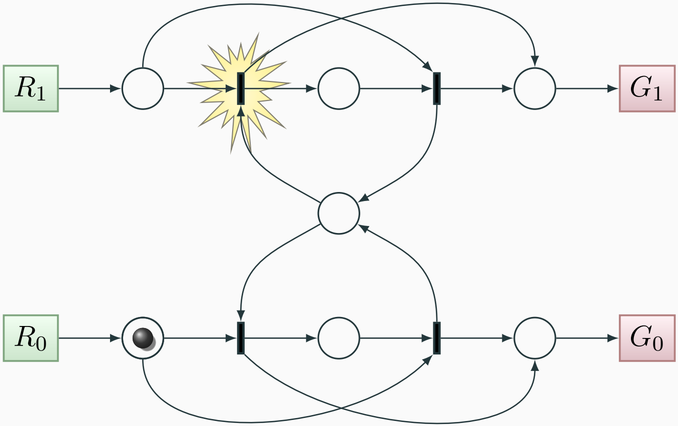
Arbiter circuits



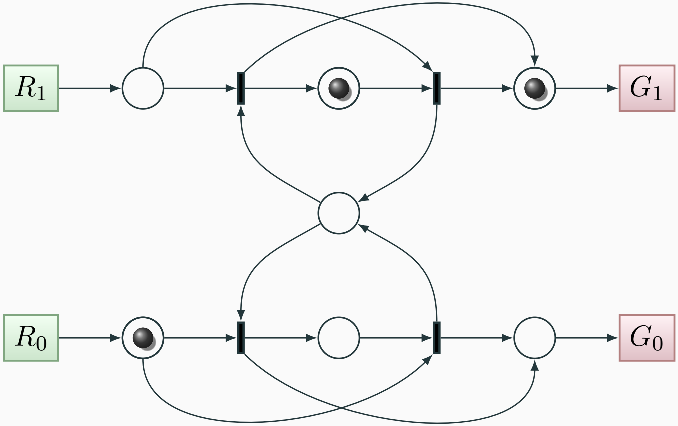
Arbiter circuits



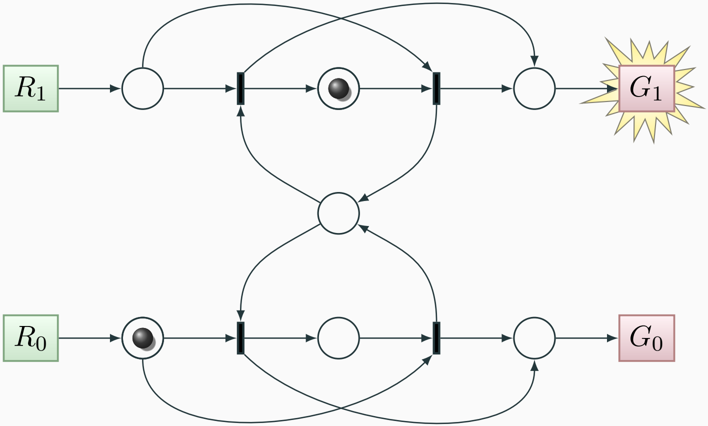
Arbiter circuits



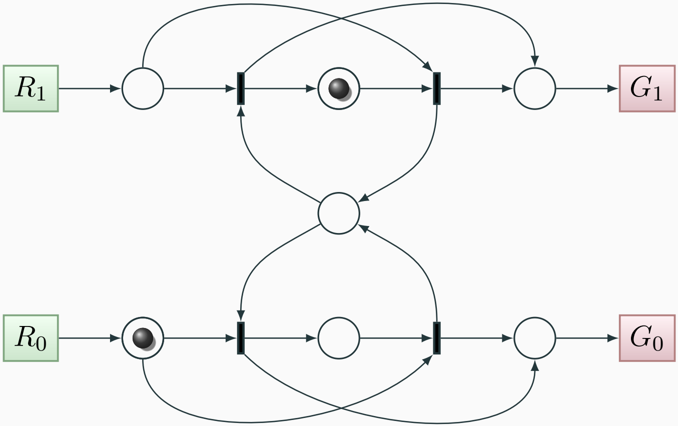
Arbiter circuits



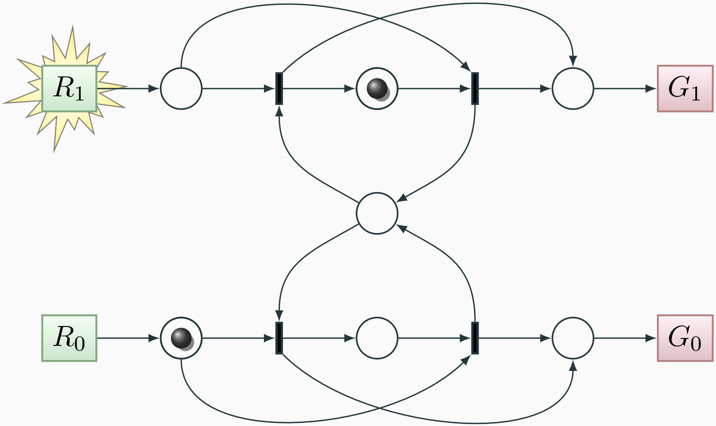
Arbiter circuits



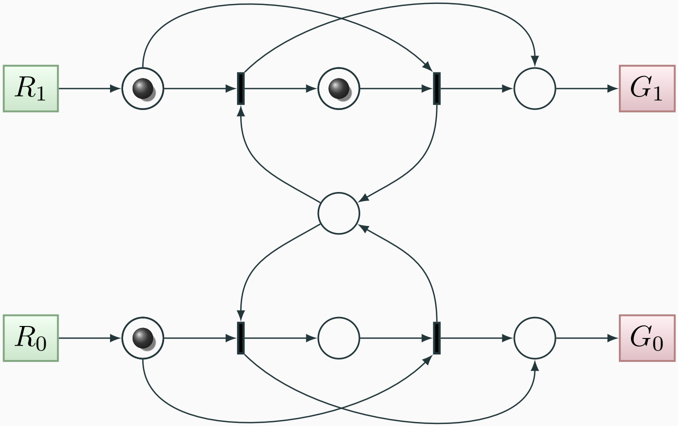
Arbiter circuits



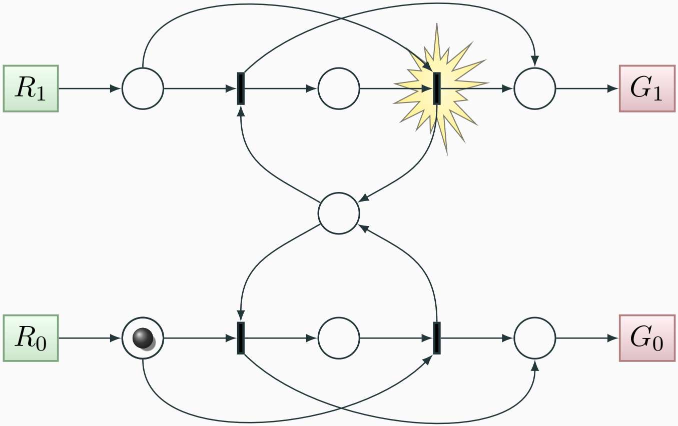
Arbiter circuits



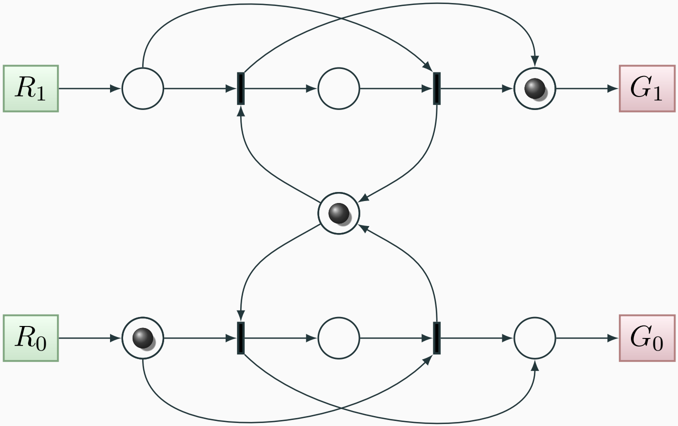
Arbiter circuits



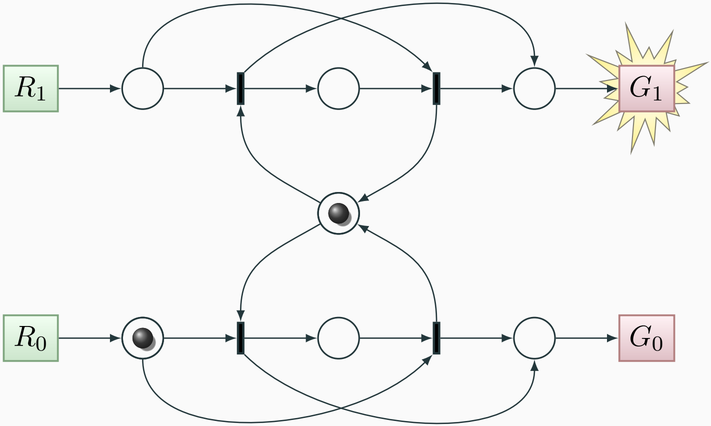
Arbiter circuits



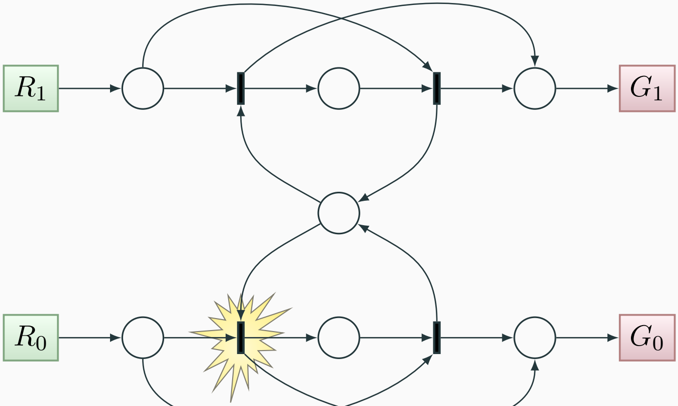
Arbiter circuits



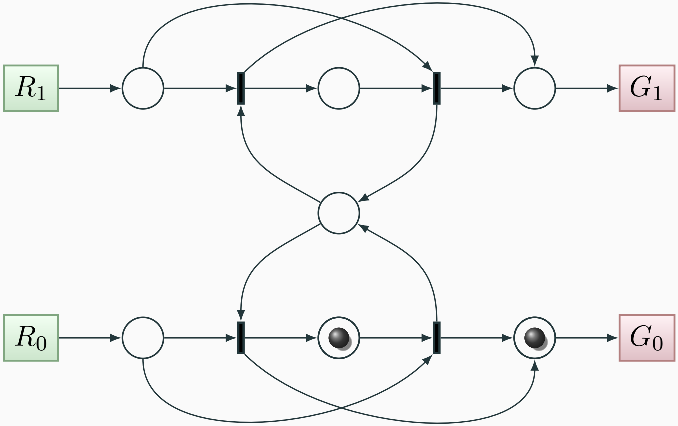
Arbiter circuits



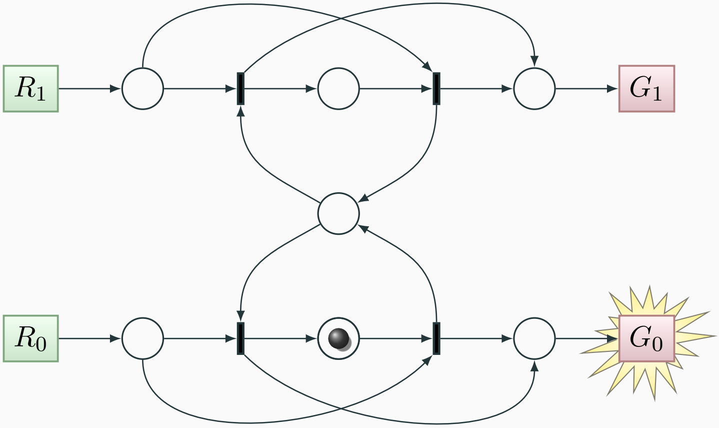
Arbiter circuits



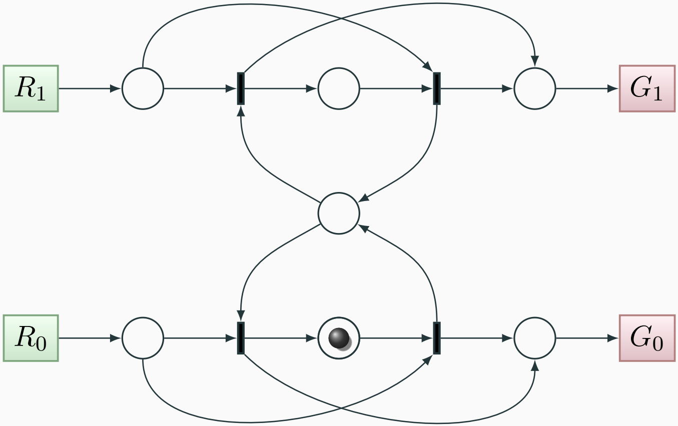
Arbiter circuits



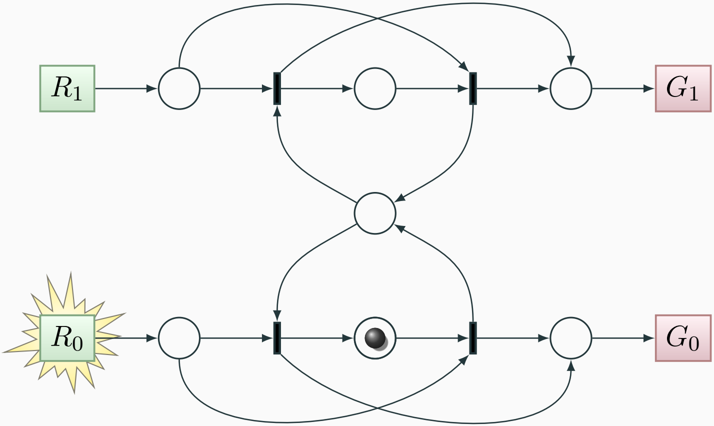
Arbiter circuits



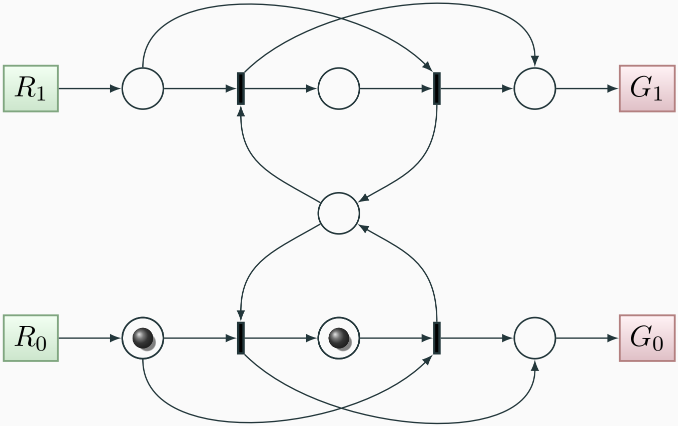
Arbiter circuits



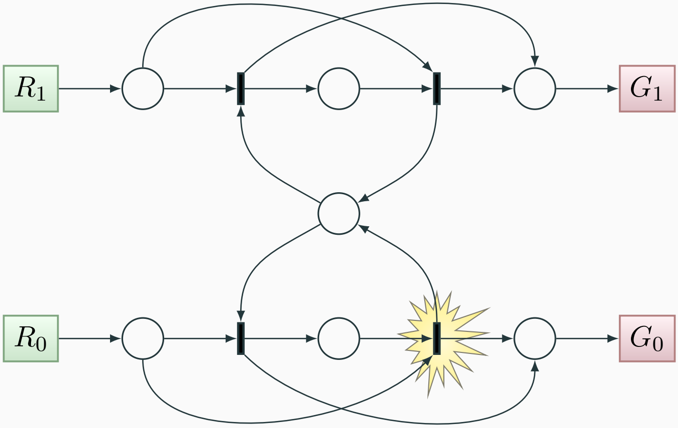
Arbiter circuits



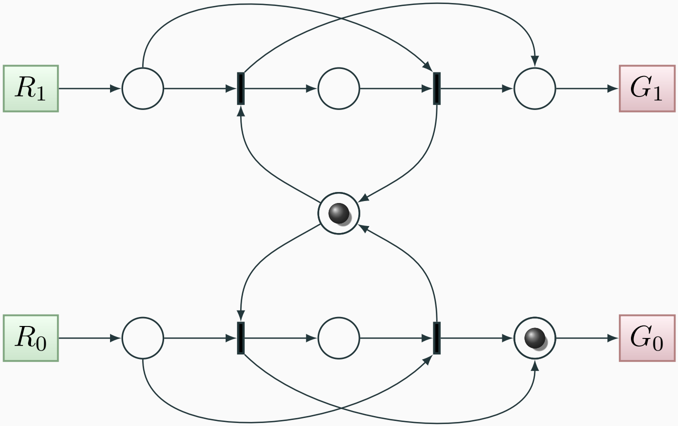
Arbiter circuits



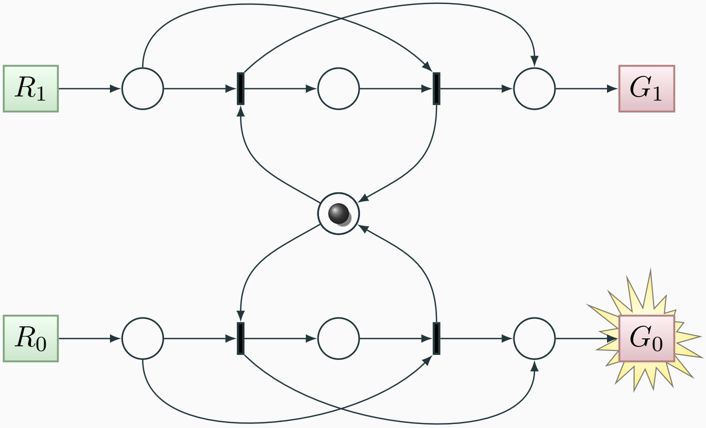
Arbiter circuits



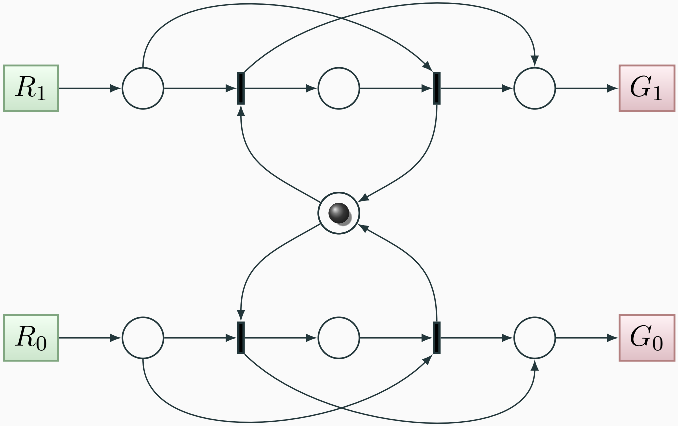
Arbiter circuits



Arbiter circuits



Arbiter circuits



A job for a 4-way arbiter

Each player's button pre-empts the other players.

A job for a 4-way arbiter

Each player's button pre-empts the other players.

A job for a 4-way arbiter

Each player's button pre-empts the other players.

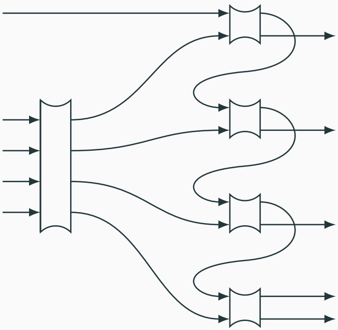
A job for a 4-way arbiter

Each player's button pre-empts the other players.

A job for a 4-way arbiter

Each player’s button pre-empts the other players.

Triangle mesh arbiter



Triangle mesh arbiter

$f(n) \in \mathbb{H}$ expresses an n-way triangle mesh arbiter by

$$\begin{aligned} f(1) &= \mathbf{I} \\ f(n+1) &= \mathbf{S}\mathbf{Z}^n\mathbf{R}\big(f\,n, (\digamma_{\mathbf{I}}\ \lambda(h,t).\,\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{n-h}\,t), \mathbf{I}))\ \iota_n^1\big) \end{aligned}$$

where $\mathbf{S} : \mathbb{H} \to \mathbb{H}$ rolls down the inputs

$$\mathbf{S} = \lambda x.\,\mathbf{ZR}(\mathbf{I}, x)$$

Triangle mesh arbiter

$f(n) \in \mathbb{H}$ expresses an n-way triangle mesh arbiter by

$$\begin{aligned} f(1) &= \mathbf{I} \\ f(n+1) &= \mathbf{S}\mathbf{Z}^n\mathbf{R}\big(f\,n, (\digamma_{\mathbf{I}}\,\lambda(h,t).\,\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{n-h}\,t), \mathbf{I}))\,\iota_n^1\big) \end{aligned}$$

where $\mathbf{S} : \mathbb{H} \to \mathbb{H}$ rolls down the inputs

$$\mathbf{S} = \lambda x.\,\mathbf{ZR}(\mathbf{I}, x)$$

For example, letting $n = 4$ and

$$g = \lambda(h,t).\,\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{4-h}\,t), \mathbf{I})$$

we have

$$f(5) = \mathbf{S}\mathbf{Z}^4\mathbf{R}\big(f\,4, (\digamma_{\mathbf{I}}\,\lambda(h,t).\,\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{4-h}\,t), \mathbf{I}))\,\iota_4^1\big)$$

Triangle mesh arbiter

$f(n) \in \mathbb{H}$ expresses an n-way triangle mesh arbiter by

$$\begin{aligned} f(1) &= \mathbf{I} \\ f(n+1) &= \mathbf{S}\mathbf{Z}^n\mathbf{R}\big(f\,n, (\digamma_{\mathbf{I}}\ \lambda(h,t).\,\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{n-h}\,t), \mathbf{I}))\ \iota_n^1\big) \end{aligned}$$

where $\mathbf{S} : \mathbb{H} \to \mathbb{H}$ rolls down the inputs

$$\mathbf{S} = \lambda x.\,\mathbf{ZR}(\mathbf{I}, x)$$

For example, letting $n = 4$ and

$$g = \lambda(h,t).\,\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{4-h}\,t), \mathbf{I})$$

we have

$$f(5) = \mathbf{S}\mathbf{Z}^4\mathbf{R}\big(f\,4, (\digamma_{\mathbf{I}}\ g)\ \iota_4^1\big)$$

Triangle mesh arbiter

$f(n) \in \mathbb{H}$ expresses an n-way triangle mesh arbiter by

$$f(1) = \mathbf{I}$$

$$f(n+1) = \mathbf{S}\mathbf{Z}^n\mathbf{R}\big(f\,n, (\digamma_{\mathbf{I}}\ \lambda(h,t).\,\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{n-h}\,t), \mathbf{I}))\ \iota_n^1\big)$$

where $\mathbf{S} : \mathbb{H} \to \mathbb{H}$ rolls down the inputs

$$\mathbf{S} = \lambda x.\,\mathbf{ZR}(\mathbf{I}, x)$$

For example, letting $n = 4$ and

$$g = \lambda(h,t).\,\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{4-h}\,t), \mathbf{I})$$

we have

$$f(5) = \mathbf{S}\mathbf{Z}^4\mathbf{R}\big(f\,4, (\digamma_{\mathbf{I}}\ g)\,\langle 1,2,3,4\rangle\big)$$

Triangle mesh arbiter

$f(n) \in \mathbb{H}$ expresses an n-way triangle mesh arbiter by

$$\begin{aligned} f(1) &= \mathbf{I} \\ f(n+1) &= \mathbf{S}\mathbf{Z}^n\mathbf{R}\big(f\,n, (\digamma_{\mathbf{I}}\,\lambda(h,t).\,\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{n-h}\,t), \mathbf{I}))\,\iota_n^1\big) \end{aligned}$$

where $\mathbf{S} : \mathbb{H} \to \mathbb{H}$ rolls down the inputs

$$\mathbf{S} = \lambda x.\,\mathbf{ZR}(\mathbf{I}, x)$$

For example, letting $n = 4$ and

$$g = \lambda(h,t).\,\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{4-h}\,t), \mathbf{I})$$

we have

$$f(5) = \mathbf{S}\mathbf{Z}^4\mathbf{R}\big(f\,4, g(1, g(2, g(3, g(4, \mathbf{I}))))\big)$$

Triangle mesh arbiter

$f(n) \in \mathbb{H}$ expresses an n-way triangle mesh arbiter by

$$f(1) = \mathbf{I}$$
$$f(n+1) = \mathbf{S}\mathbf{Z}^n\mathbf{R}\big(f\,n, (\digamma_{\mathbf{I}}\ \lambda(h,t).\,\mathbf{ZR}(\mathbf{ZR}(\mathsf{ARB}, \mathbf{S}^{n-h}\,t), \mathbf{I}))\ \iota_n^1\big)$$

where $\mathbf{S} : \mathbb{H} \to \mathbb{H}$ rolls down the inputs

$$\mathbf{S} = \lambda x.\,\mathbf{ZR}(\mathbf{I}, x)$$

For example, letting $n = 4$ and

$$g = \lambda(h,t).\,\mathbf{ZR}(\mathbf{ZR}(\mathsf{ARB}, \mathbf{S}^{4-h}\,t), \mathbf{I})$$

we have

$$f(5) = \mathbf{SZZZZR}\big(f\,4, g(1, g(2, g(3, g(4, \mathbf{I}))))\big)$$

Triangle mesh arbiter

evaluating $f(5)$ from the inside out ...

$$\begin{aligned} g(4, \mathbf{I}) &= (\lambda(h,t).\, \mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{4-h}\, t), \mathbf{I})\ (4, \mathbf{I}) \\ &= \mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{0}\, \mathbf{I}), \mathbf{I}) \\ &\overset{?}{\equiv} \textsf{ARB} \end{aligned}$$

Triangle mesh arbiter

evaluating $f(5)$ from the inside out ...

$$\begin{aligned} g(4, \mathbf{I}) &= (\lambda(h,t).\, \mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{4-h}\, t), \mathbf{I})\ (4, \mathbf{I}) \\ &= \mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^{0}\, \mathbf{I}), \mathbf{I}) \\ &\overset{?}{\equiv} \textsf{ARB} \end{aligned}$$

$$\mathbf{I} = \ t = \ \longrightarrow$$

Triangle mesh arbiter

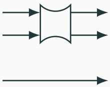
$t =$ ⟶

Triangle mesh arbiter

$$\mathbf{S}^0\,t = \longrightarrow$$

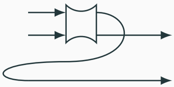
Triangle mesh arbiter

$$\mathbf{R}(\mathsf{ARB}, \mathbf{S}^0\, t) =$$



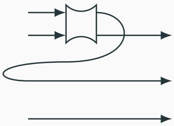
Triangle mesh arbiter

$$\mathbf{ZR}(\mathsf{ARB}, \mathbf{S}^0\, t) =$$



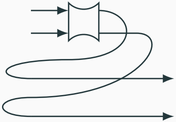
Triangle mesh arbiter

$$\mathbf{R}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^0\, t), \mathbf{I}) =$$



Triangle mesh arbiter

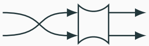
$$\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^0\, t), \mathbf{I}) =$$



Triangle mesh arbiter

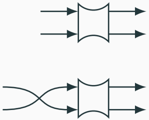
$$g(4, \mathbf{I}) = t =$$ 

Triangle mesh arbiter

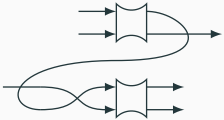
$\mathbf{S}\, t =$ 

Triangle mesh arbiter

$$\mathbf{R}(\mathsf{ARB}, \mathbf{S}\, t) =$$

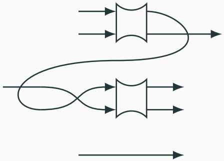


Triangle mesh arbiter

$$\mathbf{ZR}(\mathsf{ARB}, \mathbf{S}\, t) =$$ 

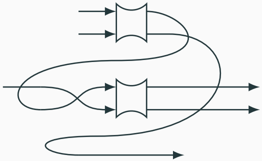
Triangle mesh arbiter

$$\mathbf{R}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}\,t), \mathbf{I}) =$$



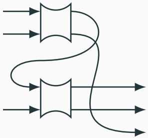
Triangle mesh arbiter

$$\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}\,t), \mathbf{I}) =$$

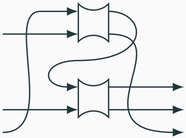


Triangle mesh arbiter

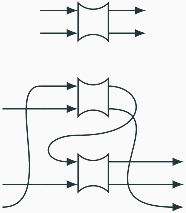
$$g(3, g(4, \mathbf{I})) = t =$$



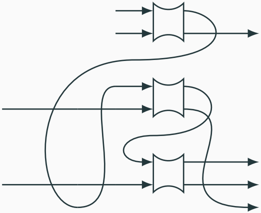
Triangle mesh arbiter

$\mathbf{S}^2\, t =$ 

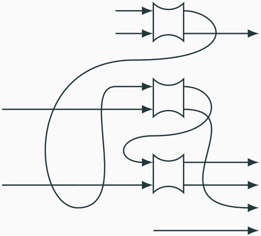
Triangle mesh arbiter

$\mathbf{R}(\mathsf{ARB}, \mathbf{S}^2\, t) =$ 

Triangle mesh arbiter

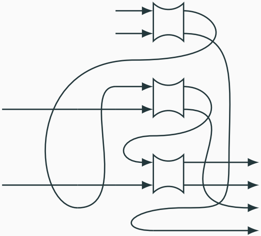
$\mathbf{ZR}(\mathsf{ARB}, \mathbf{S}^2\, t) =$ 

$$\mathbf{R}(\mathbf{ZR}(\mathsf{ARB}, \mathbf{S}^2\, t), \mathbf{I}) =$$



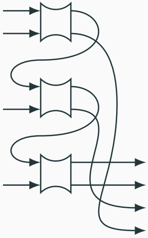
Triangle mesh arbiter

$$\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^2\, t), \mathbf{I}) =$$



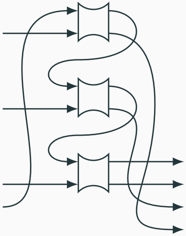
Triangle mesh arbiter

$$g(2, g(3, g(4, \mathbf{I}))) = t =$$



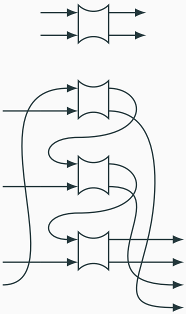
Triangle mesh arbiter

$$\mathbf{S}^3\, t =$$



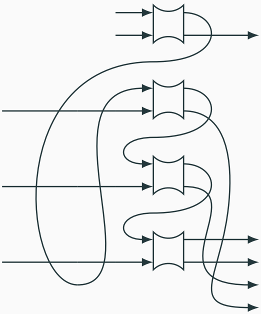
Triangle mesh arbiter

$$\mathbf{R}(\mathsf{ARB}, \mathbf{S}^3\, t) =$$

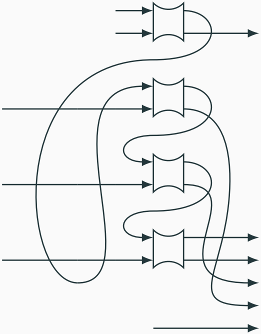


Triangle mesh arbiter

$$\mathbf{ZR}(\mathsf{ARB}, \mathbf{S}^3\, t) =$$

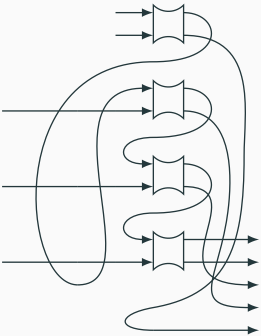


$$\mathbf{R}(\mathbf{ZR}(\mathsf{ARB}, \mathbf{S}^3\, t), \mathbf{I}) =$$



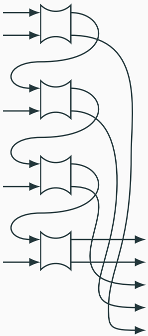
Triangle mesh arbiter

$\mathbf{ZR}(\mathbf{ZR}(\textsf{ARB}, \mathbf{S}^3\, t), \mathbf{I}) =$



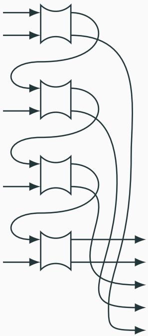
Triangle mesh arbiter

$$g(1, g(2, g(3, g(4, \mathbf{I})))) =$$

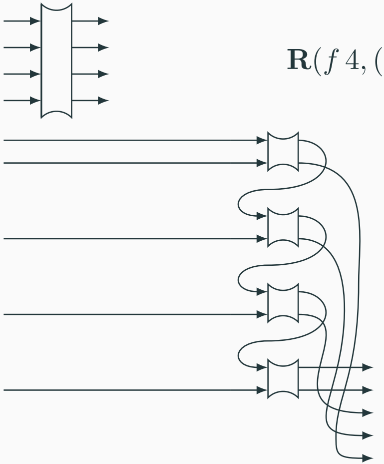


Triangle mesh arbiter

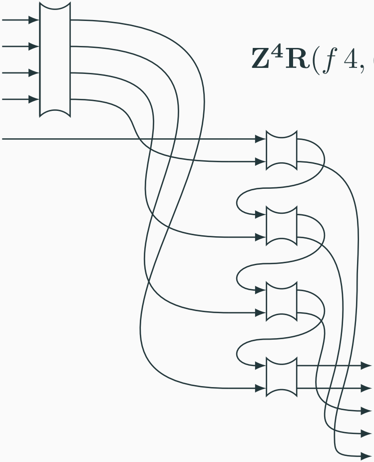
$$(F_{\mathbf{I}}\ g)\ \iota_4^1 =$$



$$\mathbf{R}(f\,4, (F\,\mathbf{I}\,g)\ \iota^1_4)$$

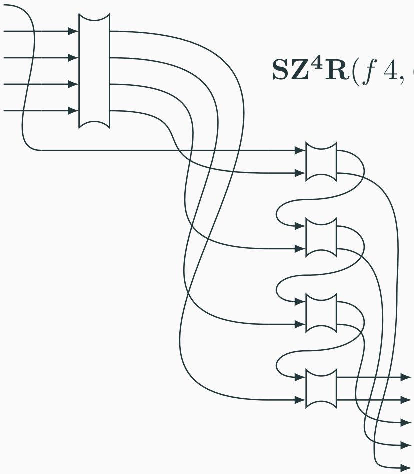


Triangle mesh arbiter



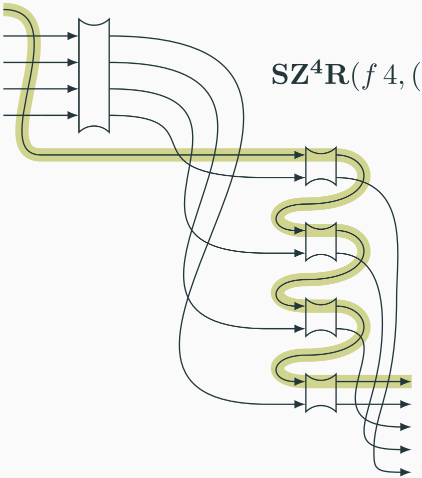
$$\mathbf{Z^4R}(f\,4, (F\;\mathbf{I}\;g)\ \iota_4^1)$$

Triangle mesh arbiter



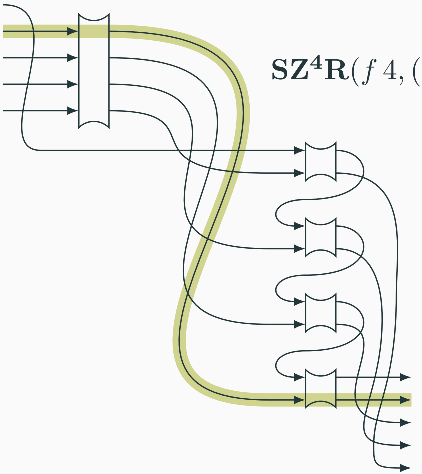
$$\mathbf{SZ^4R}(f\,4, (F\,\mathbf{I}\,g)\ \iota^1_4)$$

Triangle mesh arbiter



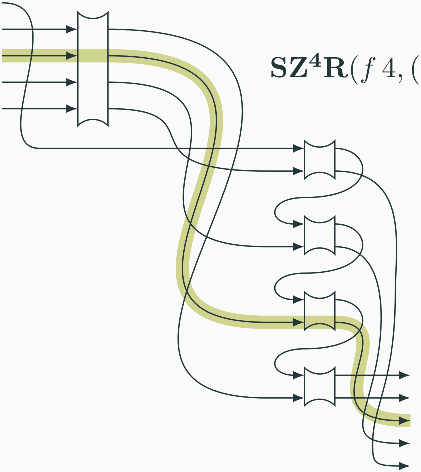
$$\mathbf{S}\mathbf{Z}^{\mathbf{4}}\mathbf{R}(f\,4, (F\ \mathbf{I}\ g)\ \iota_4^1)$$

Triangle mesh arbiter



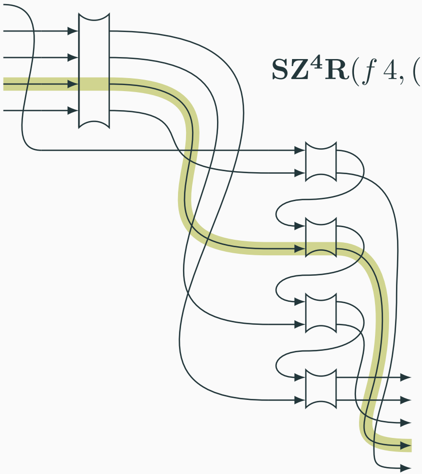
$$\mathbf{SZ^4R}(f\,4, (F\ \mathbf{I}\ g)\ \iota_4^1)$$

Triangle mesh arbiter



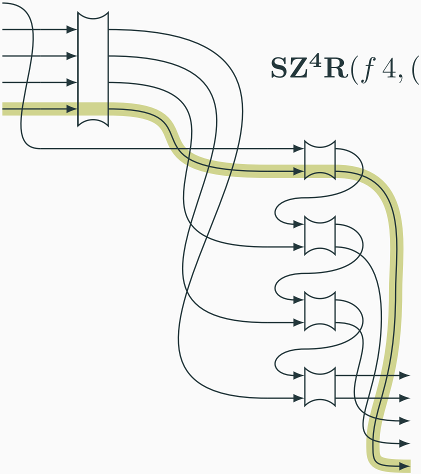
$\mathbf{SZ^4R}(f\,4, (F\,\mathbf{I}\,g)\,\iota_4^1)$

Triangle mesh arbiter



$$\mathbf{SZ^4R}(f\,4, (F\,\mathbf{I}\,g)\ \iota^1_4)$$

Triangle mesh arbiter



$$\mathbf{SZ^4R}(f\,4, (F\,\mathbf{I}\ g)\ \iota_4^1)$$

Summary

- express families of complicated circuits generally
- automatically generate checkable DI semantic models
- automatically generate corresponding netlists

Further reading

- https://www.delayinsensitive.com
  - full details on everything in this presentation
- https://statebox.org
  - overlapping ideas, more ambitious goals
- *Oliver Heaviside : The Life, Work and Times of an Electrical Genius of the Victorian Age*, Paul J. Nahin
  - historical perspective on formal methods