

Sa-TikZ*

Claudio Fiandrino

claudio.fiandrino@gmail.com

October 9, 2012

Contents

Introduction	1
1 Basic usage	2
2 The options	3
2.1 Designing choices	3
2.2 Output customization	5
3 Advanced usage	6
3.1 Identifying front input/output ports	6
3.2 Identifying input/output ports per module	7
4 Didactic purposes	8
Index	12

Introduction

The Sa-TikZ library helps in drawing *switching-architectures*. In particular, one of its aims, is to help students to verify if their exercise are correct, but it could also serve to teachers in preparing lecture notes.

The Sa-TikZ library can be loaded by means of:

```
\usetikzlibrary{switching_architectures}
```

and in this case you should manually load also:

```
\usepackage{tikz}  
\usetikzlibrary{calc, positioning, decorations.pathreplacing}
```

*This package has version number *v0.1* of 07/10/2012; it is released under and subject to the [L^AT_EX Project Public License \(LPPL\)](#).

or by means of:

```
\usepackage{sa-tikz}
```

In the latter case automatically the TikZ package and the libraries `calc`, `positioning` and `decorations.pathreplacing` are loaded.

The version *v0.1* provides a way to define Clos Networks Strictly-non-Blocking (snb) and Rearrangeable (rear). Future implementations will provide a way to draw also Benes and Cantor Networks.

1 Basic usage

The simplest use of the package is to define a

```
\node
```

Basic command definition.

with one of the following options

```
/tikz/clos snb (no value)
```

Option for drawing a Clos Network Strictly-non-Blocking.

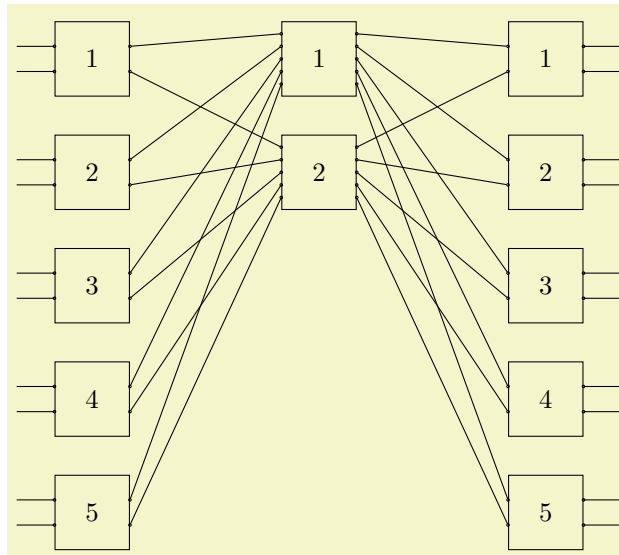
```
/tikz/clos rear (no value)
```

Option for drawing a Clos Network Rearrangeable.

inside a `tikzpicture` environment:

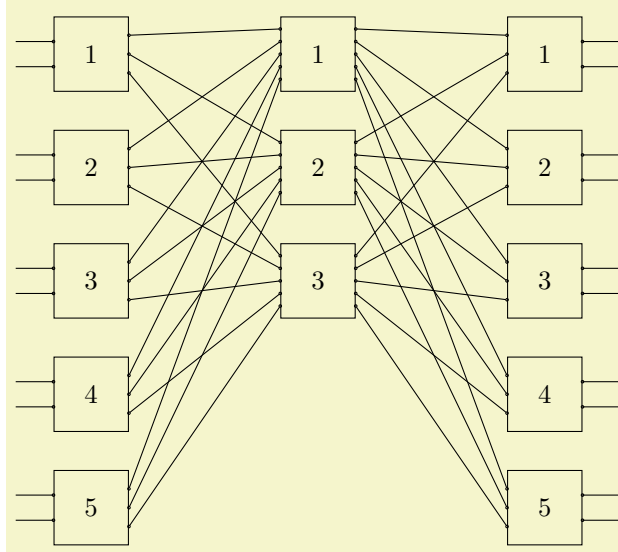
```
\begin{tikzpicture}[\langle options \rangle]
  \langle environment contents \rangle
\end{tikzpicture}
```

The following example shows a Rearrangeable Clos Network.



```
\begin{tikzpicture}
  \node[clos rear] {};
\end{tikzpicture}
```

Consider instead a Strictly-non-Blocking Clos Network:



```
\begin{tikzpicture}
  \node[clos snb] {};
\end{tikzpicture}
```

Notice from the examples that automatically the library is able to compute the constraints that define a Clos Network to be Strictly-non-Blocking or Rearrangeable. Moreover, the network drawn is characterized by:

- the first stage with:
 - a number of modules equal to 5;
 - each one with two input ports;
- the last stage with:
 - a number of modules equal to 5;
 - each one with two output ports.

Each module of the network is numbered according to the stage it belongs to.

2 The options

2.1 Designing choices

Usually, as design parameter one knows the total number of input ports of the first stage and the total number of output ports of the last stage. These two

parameters could be modify by means of:

`/tikz/N={\value}` (no default, initially 10)

This is the number of total input ports in the first stage.

`/tikz/M={\value}` (no default, initially 10)

This is the number of total output ports in the first stage.

A second design parameter usually is the number of modules present in the first and last stage. *Sa-TikZ* defines:

`/tikz/r1={\value}` (no default, initially 5)

This is the number of total input ports in the first stage.

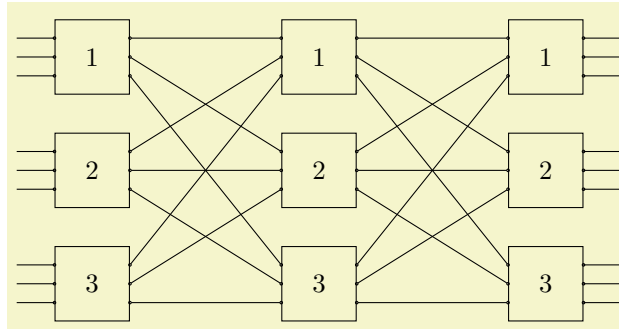
`/tikz/r3={\value}` (no default, initially 5)

This is the number of total output ports in the first stage.

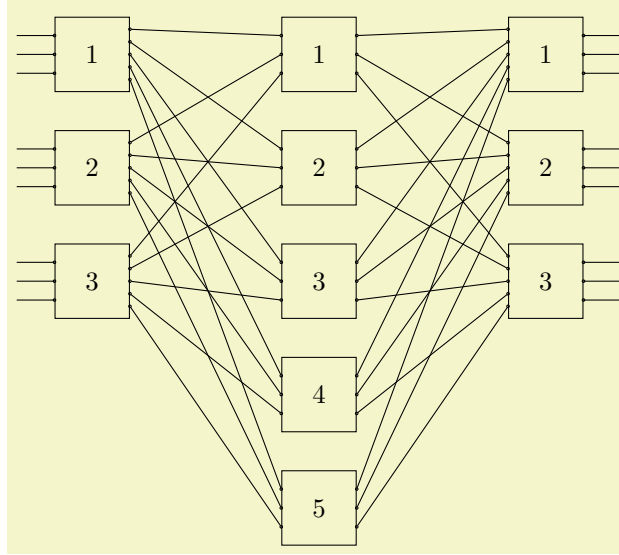
The two design parameter provide the number of ports of each module:

$$m_1 = \frac{N}{r_1} \quad m_3 = \frac{M}{r_3}$$

Some examples considering $N=9$, $r_1=3$, $M=9$ and $r_3=3$.



```
\begin{tikzpicture}
  \node[N=9,r1=3,M=9,r3=3,clos rear] {};
\end{tikzpicture}
```



```
\begin{tikzpicture}
  \node[N=9,r1=3,M=9,r3=3,clos snb] {};
\end{tikzpicture}
```

2.2 Output customization

This subsection focuses on how to customize the aspect of the drawing.

/tikz/module size={ $\langle value \rangle$ } (no default, initially 1cm)

This option allows to set the module dimension.

/tikz/module ysep={ $\langle value \rangle$ } (no default, initially 1.5)

This option allows to set the vertical module distance factor.

/tikz/module xsep={ $\langle value \rangle$ } (no default, initially 3)

This option allows to set the horizontal module distance factor.

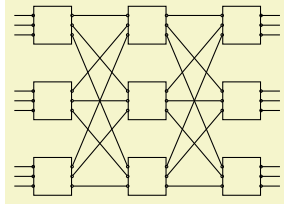
/tikz/module label opacity={ $\langle value \rangle$ } (no default, initially 1)

This option allows to mask the module label when the $\langle value \rangle$ is set to 0.

/tikz/pin length factor={ $\langle value \rangle$ } (no default, initially 1)

This option allows to reduce/increase the length of the pins drawn in input/output. Use a $\langle value \rangle$ [0,1] to reduce the length or, viceversa, a $\langle value \rangle$ greater than 1 to increase the length.

The following example shows a Rearrangeable Clos Network with some options modification. Notice that the `module label opacity` should be given as parameter of the desired network.



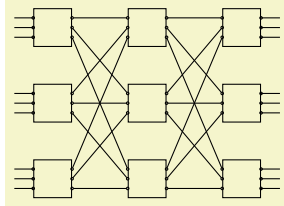
```
\begin{tikzpicture}[N=9,r1=3,M=9,r3=3]
  \node[module size=0.5cm,pin length factor=0.5,
    module ysep=1, module xsep=1.25,
    clos rear={module label opacity=0}] {};
\end{tikzpicture}
```

The options could also be introduced with the standard TikZ syntax:

`\tikzset{<options>}`

Command that process the `<options>`: the should be provided separated by a comma.

Therefore, the previous example could be modified into:



```
\tikzset{module size=0.5cm,pin length factor=0.5,
  module ysep=1, module xsep=1.25}
\begin{tikzpicture}[N=9,r1=3,M=9,r3=3]
  \node[clos rear={module label opacity=0}] {};
\end{tikzpicture}
```

3 Advanced usage

In this section some more advanced examples are shown.

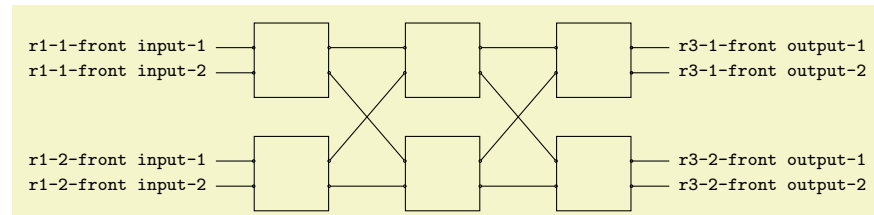
3.1 Identifying front input/output ports

In the first one it is shown how to reference the front input and output ports for the first and last stage. Each front input port could be accessed by means of:

`r1-module number-front input-port number`; example:
`r1-1-front input-1`;

Each front output port could be accessed by means of:

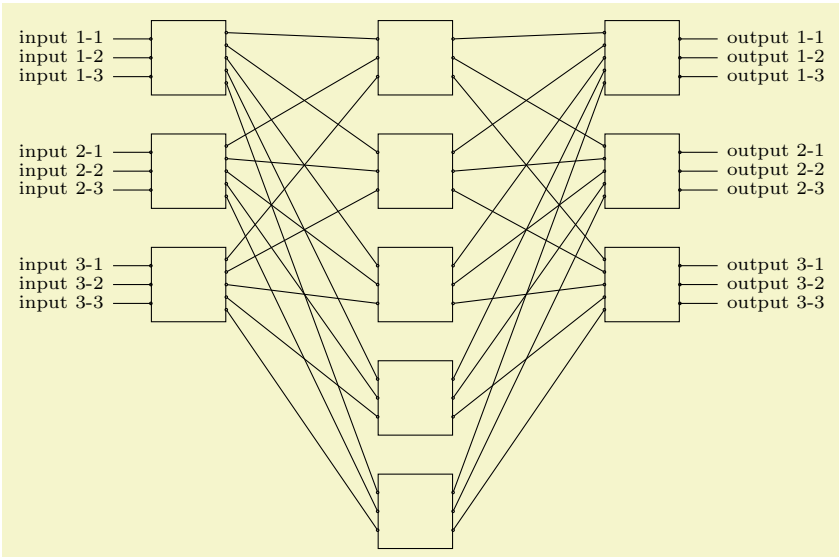
`r3-module number-front output-port number`; example:
`r3-1-front output-1`;



```

\begin{tikzpicture}[module xsep=2]
  \node[N=4,r1=2,M=4,r3=2,clos rear={module label opacity=0}] {};
  \foreach \name
    in {r1-1-front input-1,r1-1-front input-2,
        r1-2-front input-1,r1-2-front input-2}
    \node[left] at (\name) {\scriptsize{\texttt{\name}}};
  \foreach \name
    in {r3-1-front output-1,r3-1-front output-2,
        r3-2-front output-1,r3-2-front output-2}
    \node[right] at (\name) {\scriptsize{\texttt{\name}}};
\end{tikzpicture}

```



```

\begin{tikzpicture}
  \node[N=9,r1=3,M=9,r3=3,clos snb={module label opacity=0}] {};

  \foreach \startmodule in {1,...,3}{
    \foreach \port in {1,...,3}
      \node[left] at (r1-\startmodule-front input-\port)
        {\scriptsize{input \startmodule-\port}};
  }
  \foreach \startmodule in {1,...,3}{
    \foreach \port in {1,...,3}
      \node[right] at (r3-\startmodule-front output-\port)
        {\scriptsize{output \startmodule-\port}};
  }
\end{tikzpicture}

```

3.2 Identifying input/output ports per module

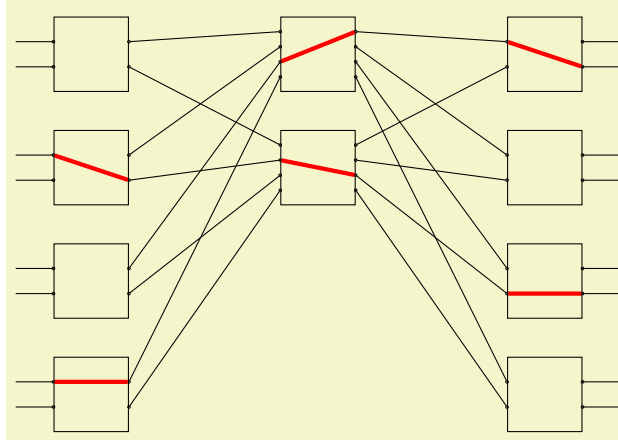
It is also possible to access, for each module, its input and output ports. The syntax is similar to the one used for front input and output ports; each input port could be accessed by means of:

rstage number-module number-input-port number; example: r1-1-input-1;

Each output port could be accessed by means of:

`rstage number-module number-front output-port number`; example:
`r2-1-output-1`;

This allows you to design a connection from the first stage to the last stage. Here is an example.



```
\begin{tikzpicture}
  \node[N=8,r1=4,M=8,r3=4,clos rear={module label opacity=0}] {};
  \draw[red,ultra thick] (r1-2-input-1)--(r1-2-output-2)
    (r2-2-input-2)--(r2-2-output-3)
    (r3-3-input-2)--(r3-3-output-2);
  \draw[red,ultra thick] (r1-4-input-1)--(r1-4-output-1)
    (r2-1-input-3)--(r2-1-output-1)
    (r3-1-input-1)--(r3-1-output-2);
\end{tikzpicture}
```

In the next version it will be implemented an automatic way to design connection paths.

4 Didactic purposes

To quickly draw a Clos Network it is possible to exploit:

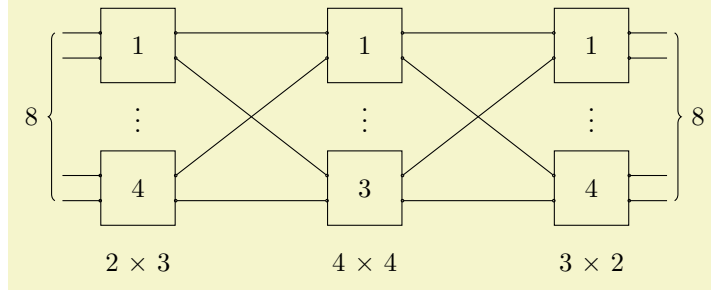
`/tikz/clos snb example` (no value)

Option for quickly drawing a Clos Network Strictly-non-Blocking.

`/tikz/clos rear example` (no value)

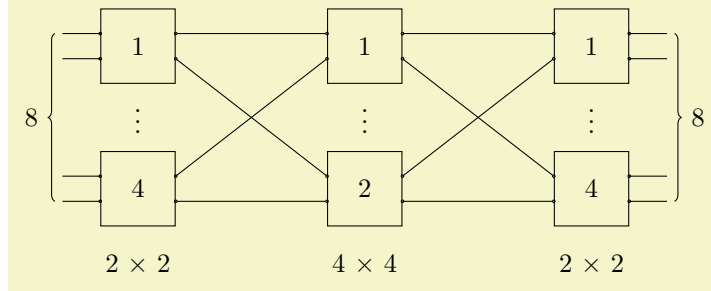
Option for quickly drawing a Clos Network Rearrangeable.

In this way the network is not seen in the whole complexity, but it is synthetically depicted. An example of a Strictly-non-Blocking Clos Network drawn with this approach:



```
\begin{tikzpicture}[N=8,r1=4,M=8,r3=4]
  \node[clos snb example] {};
\end{tikzpicture}
```

Example of a Rearrangeable Clos Network:



```
\begin{tikzpicture}[N=8,r1=4,M=8,r3=4]
  \node[clos rear example] {};
\end{tikzpicture}
```

The networks drawn, automatically display the values at which the input parameters N , M , r_1 and r_3 have been set. However, to let the user to have the possibility of deploying labels rather than the input parameter values, the following options is available:

`/tikz/clos example with labels` (no value)

Option for quickly drawing a Clos Network with custom labels.

The labels that could be customized are:

`/tikz/N label={\value}` (default N)

This options sets the label representing the total number of ports in the first stage.

`/tikz/r1 label={\value}` (default r_1)

This options sets the label representing the number of modules in the first stage.

`/tikz/m1 label={\value}` (default m_1)

This options sets the label representing the number of ports per module in the first stage.

`/tikz/r2 label={\value}` (default r_2)

This options sets the label representing the number of modules in the second stage.

`/tikz/M label={\value}` (default M)

This options sets the label representing the total number of ports in the last stage.

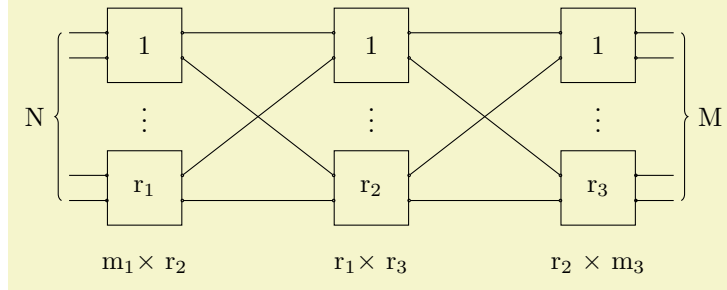
`/tikz/r3 label={\value}` (default r_3)

This options sets the label representing the number of modules in the last stage.

`/tikz/m3 label={\value}` (default m_3)

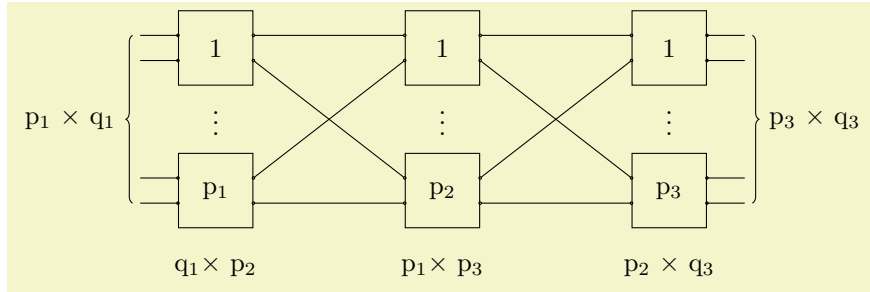
This options sets the label representing the number of ports per module in the last stage.

An example with default values:



```
\begin{tikzpicture}[N=8,r1=4,M=8,r3=4]
  \node[clos example with labels] {};
\end{tikzpicture}
```

Here is an example with custom labels introduced by means of the `\tikzset` syntax.



```

\tikzset{N label={p$_1$ $\times$ q$_1$},M label={p$_3$ $\times$ q$_3$},
r1 label=p$_1$, m1 label=q$_1$, r2 label=p$_2$,r3 label=p$_3$, m3 label=q$_3$}
\begin{tikzpicture}[N=8,r1=4,M=8,r3=4]
  \node[clos example with labels] {};
\end{tikzpicture}

```

Index

`clos example with labels` key, 9
`clos rear` key, 2
`clos rear example` key, 8
`clos snb` key, 2
`clos snb example` key, 8

Environments
 `tikzpicture`, 2

M key, 4
M label key, 10
m1 label key, 9
m3 label key, 10
module label opacity key, 5
module size key, 5
module xsep key, 5
module ysep key, 5

N key, 4
N label key, 9
`\node`, 2

`pin length factor` key, 5

r1 key, 4
r1 label key, 9
r2 label key, 10
r3 key, 4
r3 label key, 10

`/tikz/`
 `clos example with labels`, 9
 `clos rear`, 2
 `clos rear example`, 8
 `clos snb`, 2
 `clos snb example`, 8
 M, 4
 M label, 10
 m1 label, 9
 m3 label, 10
 module label opacity, 5
 module size, 5
 module xsep, 5
 module ysep, 5
 N, 4
 N label, 9
 `pin length factor`, 5

r1, 4
r1 label, 9
r2 label, 10
r3, 4
r3 label, 10
`tikzpicture` environment, 2
`\tikzset`, 6