

1 Package Options

As of now, just one:

`noextracmds` This will suppress the definition of a few auxiliary commands. See [1.1](#).

1.1 Auxiliary Commands

Those are based on some ideas from Redaelli et al. (CircuiTiKz). Main differences: a variable number of parameters (see below) and it always also adds an empty node `<coord>`

<code>\showcoordtrue</code>	<code>\showcoordtrue</code>
<code>\shoocoordsfalse</code>	<code>\showcoordsfalse</code>

These will affect as the `\coord` will behave, with `\showcoordtrue` a red pin will be added to the newly defined coordinate.

<code>\coord</code>	<code>\coord(<coord>)</code>
<code>\pincoord</code>	<code>\pincoord(<coord>)</code>
	<code>\pincoord(<coord> , <color>)</code>
	<code>\pincoord(<coord> , <color> , <angle>)</code>
	<code>\pincoord(<coord> , <color> , <angle> , <distance>)</code>

The `\coord` always expects a single parameter `<coord>`. A coordinate and node with the same name will be created. If `\showcoordtrue` is en force, it will also add a pin.

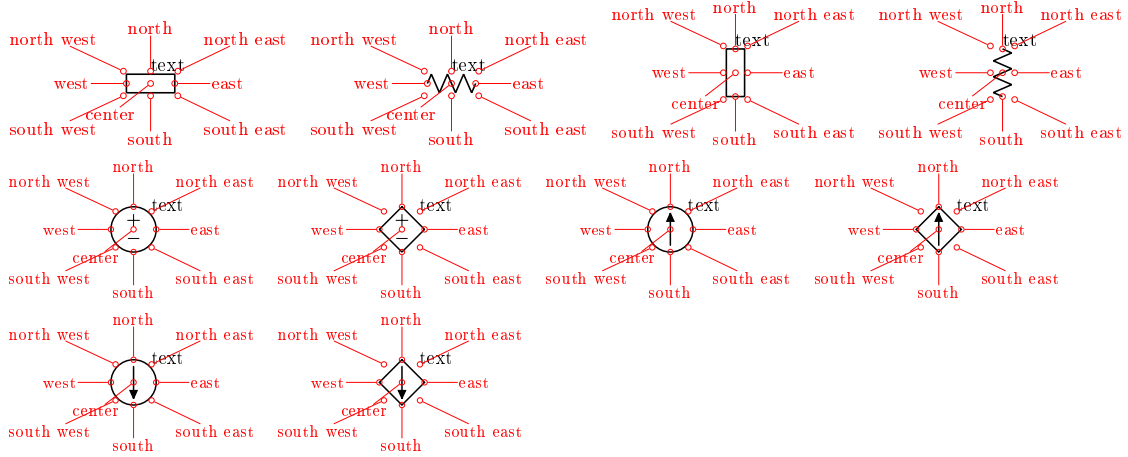
The `\pincoord` expects from one to 4 parameters, as listed. If omitted, the default value for distance is 4 (unit: pt), the default value for the angle is -45 (degrees), the default value for color is blue. In fact, the `\coord(name)` is just a short cut for `\pincoord (name,red,45)`, if `\showcoordtrue`.

2 Auxiliary Shapes and Basic Keys

Those shapes are not intended for end users.

2.1 Auxiliary shapes

A set of auxiliary shapes are defined, but not meant to be used otherwise, though their anchors might be relevant:



Note: The main point being that, regardless of the sub-shape orientation, the intuitive geographical coordinates applies.

2.2 General Keys

The following set of keys allow for shape fine tuning:

<code>outer sep</code>	Text outer separation, initial value: 1.5pt
<code>inner sep</code>	Text inner separation, initial value: 1pt
<code>thickness</code>	Components thickness (relative to the drawing thickness), initial value: 2
<code>tip len</code>	tip len (current source). initial value: 4pt
<code>tip type</code>	possible values: <code>triangle</code> and <code>bezier</code> . initial value: <code>triangle</code>
<code>minussign len</code>	Minus sign len (voltage source). initial value: <code>\pgf@circ@Rlen/14</code>
<code>plussign len</code>	Plus sign len (voltage source). initial value: <code>1.1\pgf@circ@Rlen/14</code>
<code>source radius</code>	The base radius. initial value: <code>0.3\pgf@circ@Rlen</code>
<code>round sources</code>	Sources will be round ones
<code>control sources</code>	Sources will be control/diamond ones
<code>generic, european</code>	Impedances will be generic rectangles
<code>zigzag, american</code>	Impedances will be draw as zigzags

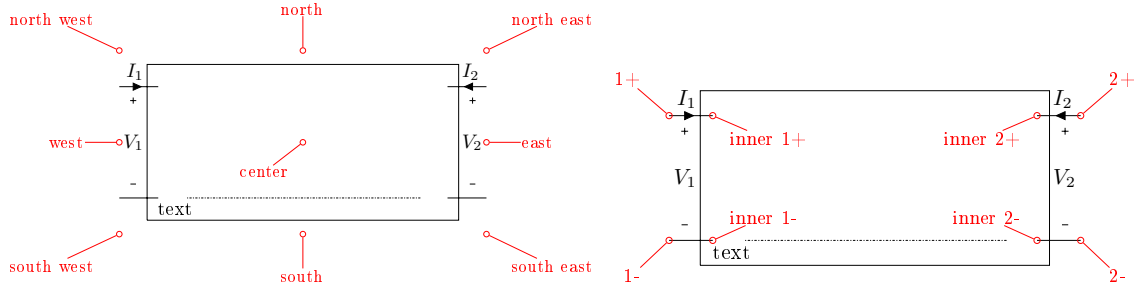
Note: Those keys can be used with all the following components: `<Quad>`, `<Quad Z>`, `<Quad Y>`, `<Quad G>`, `<Quad H>`, `<ToQuad>`, `<ToQuad Z>`, `<ToQuad Y>`, `<ToQuad G>`, `<ToQuad H>`, `<Black Box>`, `<Thevenin>`, `<Norton>`, `<ToBlack Box>`, `<ToThevenin>` and `<ToNorton>`.

3 Z, Y, G, H Quadripoles

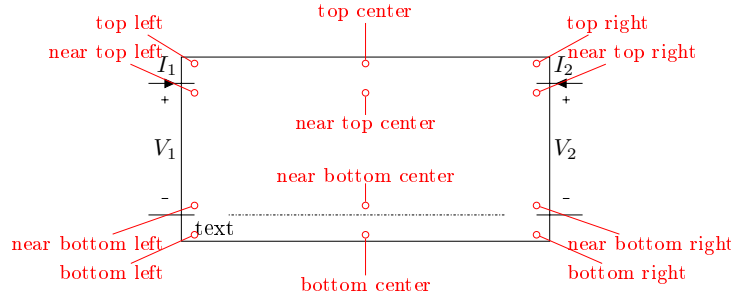
A set of configurable Quadripoles is defined, whereas quadripoles parameters (for instance Z_{11} , Z_{12} , Z_{21} and Z_{22}) are <key-value> parameters.

3.1 The Base Quadripole

The base shape just draws a base box and sets some connection anchors: $1+$, $1-$, *inner* $1+$, *inner* $1-$, $2+$, $2-$, *inner* $2+$ and *inner* $2-$, besides the geographic ones:



And also a set of (meant for) *text* anchors:



3.2 Customization keys

Additionally, one has:

<code>base width</code>	The 'box' width
<code>half base width</code>	Ditto, half width. Initial value: <code>2\pgf@circ@Rlen</code> .
<code>base height</code>	The distance between $1+$ and $1-$. The 'box' full height is equal to $2 * (\text{half base height} + \text{height ext} + \text{height ext+})$.
<code>half base height</code>	Ditto, half height. Initial value: <code>\pgf@circ@Rlen/7</code>
<code>height ext</code>	Initial value: <code>2\pgf@circ@Rlen/7</code>
<code>height ext+</code>	Initial value: 0
<code>inner ext</code>	distance between the 'box' and <code>inner1+/1-/2+/2-</code> . initial value: <code>\pgf@circ@Rlen/7</code>
<code>outer ext</code>	distance between the 'box' and <code>1+/1-/2+/2-</code> . initial value: <code>5\pgf@circ@Rlen/14</code>
<code>inner marks</code>	If set, the inner anchors will be marked.
<code>outer marks</code>	If set, the outer anchors will be marked.
<code>invert</code>	The shape will be inverted, more or less like 'x scale=-1'.
<code>alt, opt</code>	Case a Voltage source is zero, a series impedance will be draw vertically.
<code>outer x fit to</code>	<code>outer x fit={\langle CoordA \rangle}{\langle CoordB \rangle}</code> . The width will be set so that $\langle 1+ \rangle$ and $\langle 2+ \rangle$ (or $\langle 1- \rangle$ and $\langle 2- \rangle$, depending on the used anchor) will fit $\langle CoordA \rangle$ and $\langle CoordB \rangle$

inner x fit to *inner x fit*= $\{\langle\text{CoordA}\rangle\}\{\langle\text{CoordB}\rangle\}$. The width will be set so that $\langle\text{inner } 1+\rangle$ and $\langle\text{inner } 2+\rangle$ (or $\langle\text{inner } 1-\rangle$ and $\langle\text{inner } 2-\rangle$, depending on the used anchor) will fit $\langle\text{CoordA}\rangle$ and $\langle\text{CoordB}\rangle$

y fit to *y fit*= $\{\langle\text{CoordA}\rangle\}\{\langle\text{CoordB}\rangle\}$. The distance between, lets say *1+* and *1-* will be made equal to the distance between *CoordA* and *CoordB*.

Note: For Quadripoles, *outer x fit* and *inner x fit* might result in a shape rotation. *y fit* never will result in a rotation.

Note: Those keys can be used with all the following components: $\langle\text{Quad}\rangle$, $\langle\text{Quad Z}\rangle$, $\langle\text{Quad Y}\rangle$, $\langle\text{Quad G}\rangle$, $\langle\text{Quad H}\rangle$, $\langle\text{ToQuad}\rangle$, $\langle\text{ToQuad Z}\rangle$, $\langle\text{ToQuad Y}\rangle$, $\langle\text{ToQuad G}\rangle$, $\langle\text{ToQuad H}\rangle$, $\langle\text{Black Box}\rangle$, $\langle\text{Thevenin}\rangle$, $\langle\text{Norton}\rangle$, $\langle\text{ToBlack Box}\rangle$, $\langle\text{ToThevenin}\rangle$ and $\langle\text{ToNorton}\rangle$.

A small example of the 'fit to' keys:

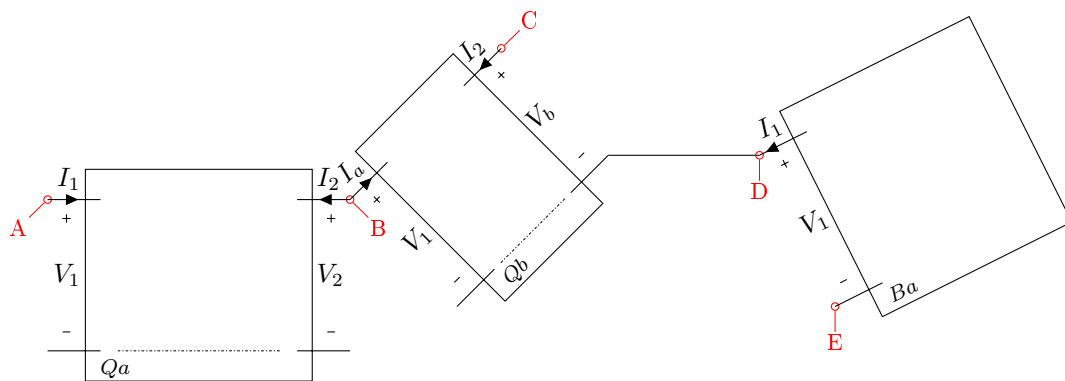
LaTeX Code:

```

1 \begin{tikzpicture}
2 \draw (0,0) coordinate(A) \showcoord(A)<225:0.2> ++(4,0) coordinate(B) \showcoord(B)<-45:0.2> ++(2,2) coordinate(C)
   \showcoord(C)<45:0.2> ;
3
4 \draw (A) node[Quad,anchor=1+,outer x fit to={A}{B}](Qa){\footnotesize$Qa$};
5 \draw (B) node[Quad,anchor=1+,outer x fit to={B}{C},I1=$I_a$,V2=$V_b$](Qb){\footnotesize$Qb$};
6
7 \draw (Qb.2-) -- ++(2,0) coordinate(D) \showcoord(D)<-90:0.2> ++(1,-2) coordinate(E) \showcoord(E)<-90:0.2>;
8
9 \draw (D) node[Black Box,anchor=1+,y fit to={D}{E}](Ba){\footnotesize$Ba$};
10
11 \draw (Qa.1-) ++(0,-1);
12 \end{tikzpicture}

```

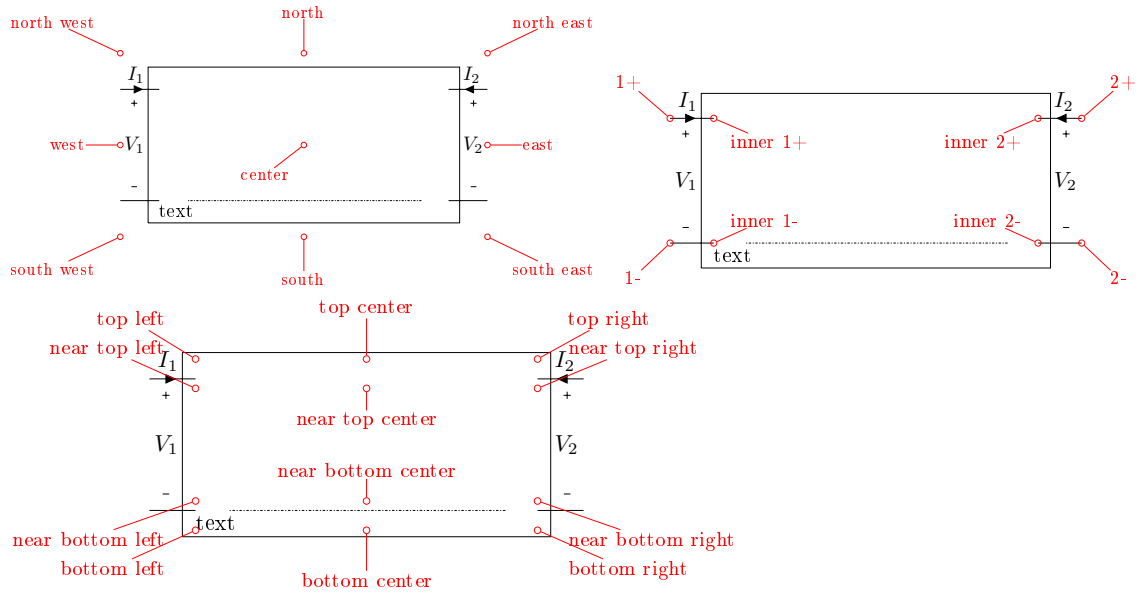
LaTeX Result:



3.3 Quad

This is just the base shape, to be used in cases where one just want to emphasises part of a circuit (using, for instance, the *inner x fit to* key, or just mark a two port black box.

Note: There is also a *ToQuad* to be used in a *to[]* path, in which case the key *outer x fit to* style will be triggered with the starting and ending points of the *to[]* path.



3.3.1 Quad Keys

<i>name</i>	\langle node-name \rangle , when using a to[] path.
<i>I1</i>	Initial value: $\$I_1\$$
<i>I2</i>	Initial value: $\$I_2\$$
<i>V1</i>	Initial value: $\$V_1\$$
<i>V2</i>	Initial value: $\$V_2\$$

3.3.2 Example of fit to uses

Squeezing a Quadripole between two parts of a circuit (nodes C and D):

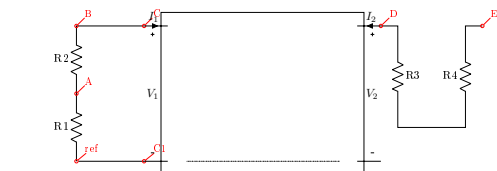
LaTeX Code:

```

1 \resizebox{0.4\textwidth}{!}{
2 \begin{tikzpicture}
3   \draw (0,0) coordinate(ref) \showcoord(ref)<45:0.2> to[R=R1] ++(0,2) coordinate(A) \showcoord(A)<45:0.2> to[R=R
4     2] ++(0,2) coordinate(B) \showcoord(B)<45:0.2>
5     -- ++(2,0) coordinate(C) \showcoord(C)<45:0.2> (C |- ref) coordinate(C1) \showcoord(C1)<45:0.2> -- (ref);
6     \draw (C) ++(7,0) coordinate(D) \showcoord(D)<45:0.2> -- ++(0.5,0) to[R=R3] ++(0,-3) -- ++(2,0) to[R=R4]
7       ++(0,3) -- ++(0.5,0) coordinate(E) \showcoord(E)<45:0.2>;
8     \draw (C) node[Quad,anchor=1+,y fit to={C}{C1},outer x fit to={C}{D}]{};
9 \end{tikzpicture}

```

LaTeX Result:



Fitting some circuit inside the Quadripole (nodes C and E):

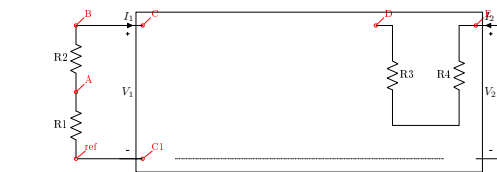
LaTeX Code:

```

1 \resizebox{0.4\textwidth}{!}{
2 \begin{tikzpicture}
3   \draw (0,0) coordinate(ref) \showcoord(ref)<45:0.2> to[R=R1] ++(0,2) coordinate(A) \showcoord(A)<45:0.2> to[R=R
4     2] ++(0,2) coordinate(B) \showcoord(B)<45:0.2>
5     -- ++(2,0) coordinate(C) \showcoord(C)<45:0.2> (C |- ref) coordinate(C1) \showcoord(C1)<45:0.2> -- (ref);
6     \draw (C) ++(7,0) coordinate(D) \showcoord(D)<45:0.2> -- ++(0.5,0) to[R=R3] ++(0,-3) -- ++(2,0) to[R=R4]
7       ++(0,3) -- ++(0.5,0) coordinate(E) \showcoord(E)<45:0.2>;
8     \draw (C) node[Quad,anchor=inner 1+,y fit to={C}{C1},inner x fit to={C}{E}]{};
9 \end{tikzpicture}}

```

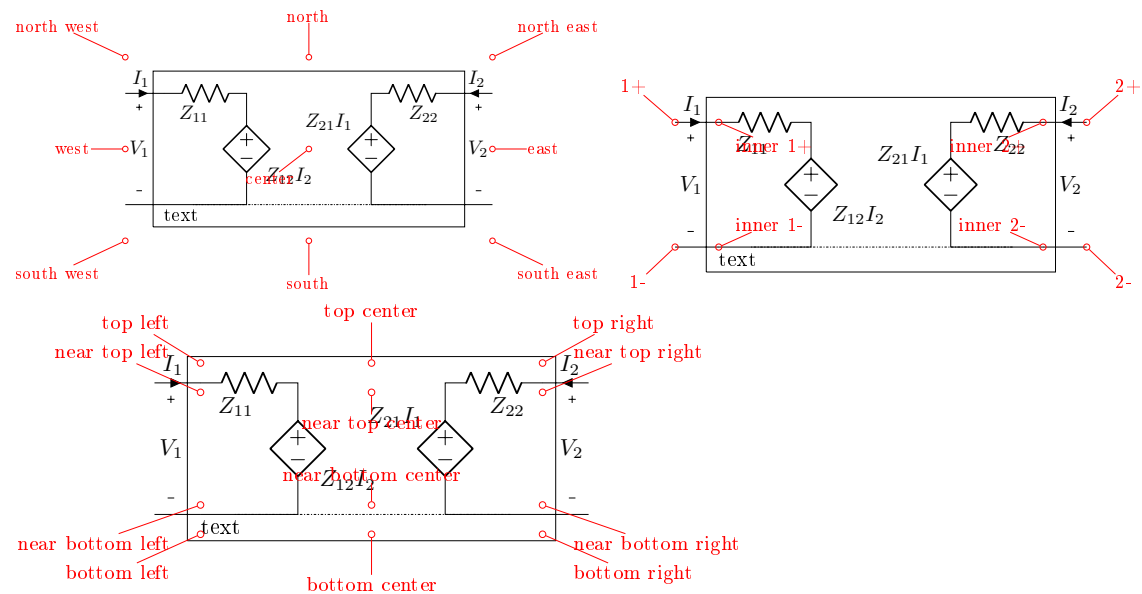
LaTeX Result:



3.4 Quad Z

This shape, besides the base anchors (see 3) it has 4 internal nodes: `<node>-Z11`, `<node>-Z12`, `<node>-Z21` and `<node>-Z22` and each of those sub-nodes has geographic anchors as defined at 2.1.

Note: There is also a `ToQuad Z` to be used in a `to[]` path, in which case the key `outer x fit to` style will be triggered with the starting and ending points of the `to[]` path.



3.4.1 Quad Z keys

name	<code><node-name></code> , when using a <code>to[]</code> path.
<code>I1</code>	Initial value: I_1
<code>I2</code>	Initial value: I_2
<code>V1</code>	Initial value: V_1

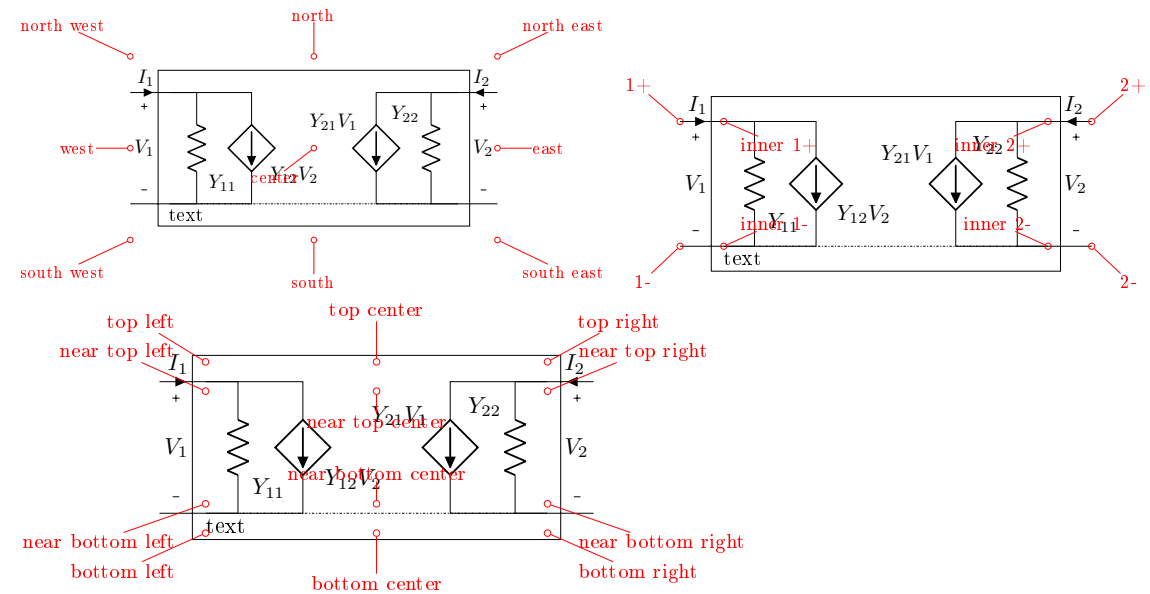
<code>V2</code>	Initial value: <code>\$V_2\$</code>
<code>Z11</code>	Initial value: <code>\$Z_{11}\$</code>
<code>Z12</code>	Initial value: <code>\$Z_{12}\$</code>
<code>Z21</code>	Initial value: <code>\$Z_{21}\$</code>
<code>Z22</code>	Initial value: <code>\$Z_{22}\$</code>
<code>Z11 label pos</code>	changes the label position. Defaults to: {south west}{top left}
<code>Z12 label pos</code>	changes the label position. Defaults to: {south east}{top left}
<code>Z21 label pos</code>	changes the label position. Defaults to: {north west}{bottom right}
<code>Z22 label pos</code>	changes the label position. Defaults to: {south east}{top right}

Note: The label pos keys expects two anchor names (... label pos={⟨anchor A⟩}⟨anchor B⟩}). The first anchors refers the sub-shape node and the second anchor is the text one.

3.5 Quad Y

This shape, besides the base anchors (see 3) it has 4 internal nodes: `<node>-Y11`, `<node>-Y12`, `<node>-Y21` and `<node>-Y22` and each of those sub-nodes has geographic anchors as defined at 2.1.

Note: There is also a *ToQuad Y* to be used in a *to[]* path, in which case the key *outer x fit to* style will be triggered with the starting and ending points of the *to[]* path.



3.5.1 Quad Y keys

<i>name</i>	⟨node-name⟩, when using a <i>to[]</i> path.
<code>I1</code>	Initial value: <code>\$I_1\$</code>
<code>I2</code>	Initial value: <code>\$I_2\$</code>
<code>V1</code>	Initial value: <code>\$V_1\$</code>
<code>V2</code>	Initial value: <code>\$V_2\$</code>
<code>Y11</code>	Initial value: <code>\$Y_{11}\$</code>
<code>Y12</code>	Initial value: <code>\$Y_{12}\$</code>
<code>Y21</code>	Initial value: <code>\$Y_{21}\$</code>

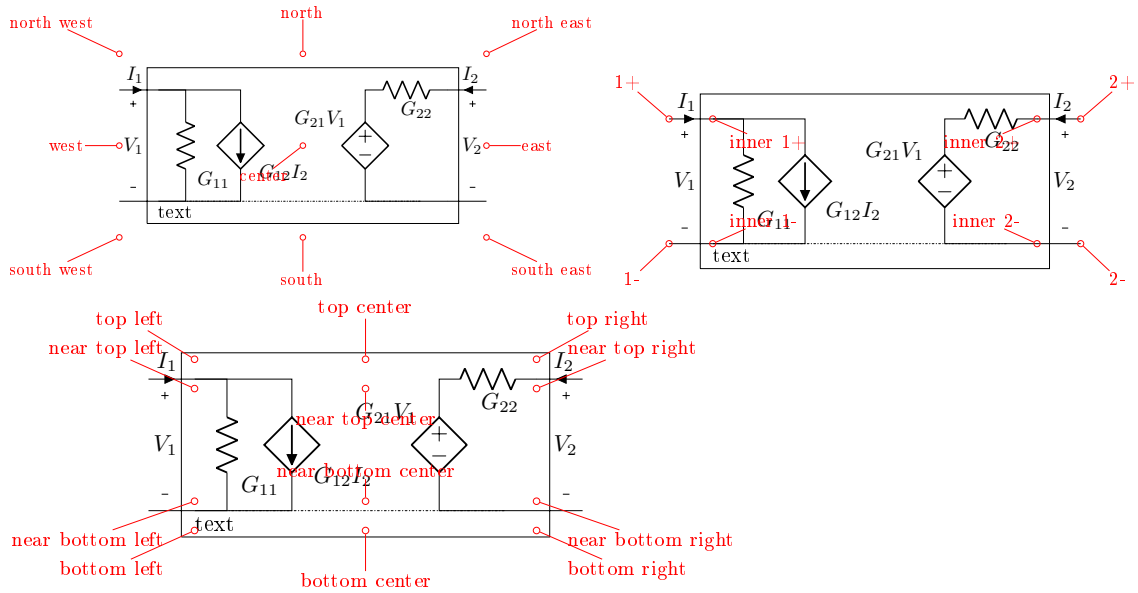
<code>Y22</code>	Initial value: $\$Y_{22}\$$
<code>Y11 label pos</code>	changes the label position. Defaults to: {south west}{top left}
<code>Y12 label pos</code>	changes the label position. Defaults to: {south east}{top left}
<code>Y21 label pos</code>	changes the label position. Defaults to: {north west}{bottom right}
<code>Y22 label pos</code>	changes the label position. Defaults to: {north west}{bottom right}

Note: The label pos keys expects two anchor names (... label pos={⟨anchor A⟩}{⟨anchor B⟩}). The first anchors refers the sub-shape node and the second anchor is the text one.

3.6 Quad G

This shape, besides the base anchors (see 3) it has 4 internal nodes: `<node>-G11`, `<node>-G12`, `<node>-G21` and `<node>-G22` and each of those sub-nodes has geographic anchors as defined at 2.1.

Note: There is also a *ToQuad G* to be used in a `to[]` path, in which case the key *outer x fit to* style will be triggered with the starting and ending points of the `to[]` path.



3.6.1 Quad G keys

<i>name</i>	⟨node-name⟩, when using a <code>to[]</code> path.
<code>I1</code>	Initial value: $\$I_1\$$
<code>I2</code>	Initial value: $\$I_2\$$
<code>V1</code>	Initial value: $\$V_1\$$
<code>V2</code>	Initial value: $\$V_2\$$
<code>G11</code>	Initial value: $\$G_{11}\$$
<code>G12</code>	Initial value: $\$G_{12}\$$
<code>G21</code>	Initial value: $\$G_{21}\$$
<code>G22</code>	Initial value: $\$G_{22}\$$
<code>G11 label pos</code>	changes the label position. Defaults to: {south west}{top left}
<code>G12 label pos</code>	changes the label position. Defaults to: {south east}{top left}
<code>G21 label pos</code>	changes the label position. Defaults to: {north west}{bottom right}

`G22 label pos`

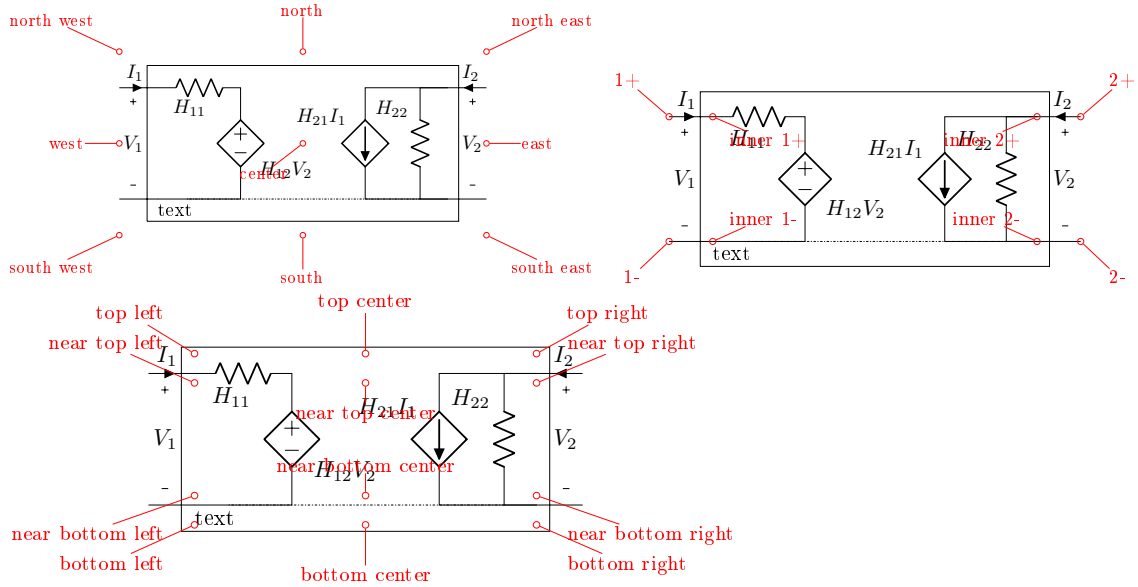
changes the label position. Defaults to: {south east}{top right}

Note: The label pos keys expects two anchor names (... label pos={⟨anchor A⟩}{⟨anchor B⟩}). The first anchors refers the sub-shape node and the second anchor is the text one.

3.7 Quad H

This shape, besides the base anchors (see 3) it has 4 internal nodes: <node>-H11, <node>-H12, <node>-H21 and <node>-H22 and each of those sub-nodes has geographic anchors as defined at 2.1.

Note: There is also a *ToQuad H* to be used in a *to[]* path, in which case the key *outer x fit to* style will be triggered with the starting and ending points of the *to[]* path.



3.7.1 Quad H keys

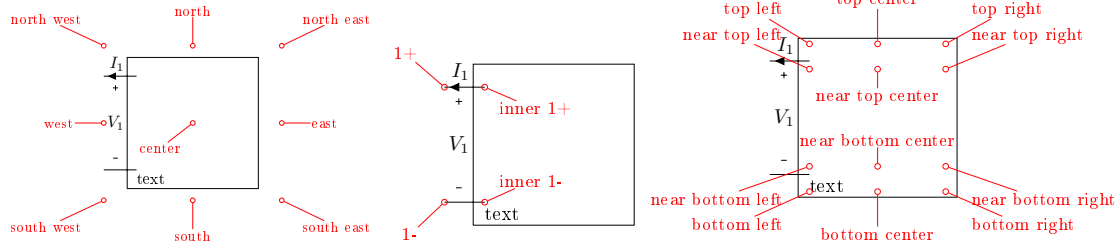
<code>name</code>	⟨node-name⟩, when using a <i>to[]</i> path.
<code>I1</code>	Initial value: $\$I_1\$$
<code>I2</code>	Initial value: $\$I_2\$$
<code>V1</code>	Initial value: $\$V_1\$$
<code>V2</code>	Initial value: $\$V_2\$$
<code>H11</code>	Initial value: $\$H_11\$$
<code>H12</code>	Initial value: $\$H_12\$$
<code>H21</code>	Initial value: $\$H_21\$$
<code>H22</code>	Initial value: $\$H_22\$$
<code>H11 label pos</code>	changes the label position. Defaults to: {south west}{top left}
<code>H12 label pos</code>	changes the label position. Defaults to: {south east}{top left}
<code>H21 label pos</code>	changes the label position. Defaults to: {north west}{bottom right}
<code>H22 label pos</code>	changes the label position. Defaults to: {north west}{bottom right}

Note: The label pos keys expects two anchor names (... label pos={⟨anchor A⟩}{⟨anchor B⟩}). The first anchors refers the sub-shape node and the second anchor is the text one.

4 Thevenin, Norton single port dipoles

4.1 The Base Black Box

The base shape just draws a base box and sets some connection anchors: $1+$, $1-$, *inner* $1+$, *inner* $1-$, besides the geographic and text ones:



4.2 Customization keys

Additionally, one has:

<code>base width</code>	The 'box' width
<code>half base width</code>	Ditto, half width. Initial value: <code>2\pgf@circ@Rlen</code> .
<code>base height</code>	The distance between $1+$ and $1-$. The 'box' full height is equal to $2*(half\ base\ height + height\ ext + height\ ext+)$.
<code>half base height</code>	Ditto, half height. Initial value: <code>\pgf@circ@Rlen/7</code>
<code>height ext</code>	Initial value: <code>2\pgf@circ@Rlen/7</code>
<code>height ext+</code>	Initial value: 0
<code>inner ext</code>	distance between the 'box' and <i>inner</i> $1+/1-/2+/2-$. initial value: <code>\pgf@circ@Rlen/7</code>
<code>outer ext</code>	distance between the 'box' and $1+/1-/2+/2-$. initial value: <code>5\pgf@circ@Rlen/14</code>
<code>inner marks</code>	If set, the inner anchors will be marked.
<code>outer marks</code>	If set, the outer anchors will be marked.
<code>invert</code>	The shape will be inverted, more or less like 'x scale=-1'.
<code>alt, opt</code>	Case a Voltage source is zero, a series impedance will be draw vertically.
<code>outer x fit to</code>	<code>outer x fit={⟨CoordA⟩}{⟨CoordB⟩}</code> . The width will be set so that the distance between $\langle 1+ \rangle$ and $\langle 2+ \rangle$ (or $\langle 1- \rangle$ and $\langle 2- \rangle$, depending on the used anchor) will be the same as $\langle CoordA \rangle$ and $\langle CoordB \rangle$.
<code>inner x fit to</code>	<code>inner x fit={⟨CoordA⟩}{⟨CoordB⟩}</code> . The width will be set so that the distance between <i>inner</i> $1+$ and <i>inner</i> $2+$ (or <i>inner</i> $1-$ and <i>inner</i> $2-$), depending on the used anchor) will be the same as $\langle CoordA \rangle$ and $\langle CoordB \rangle$
<code>y fit to</code>	<code>y fit={⟨CoordA⟩}{⟨CoordB⟩}</code> . $1+$ and $1-$ will fit <i>CoordA</i> and <i>CoordB</i> respectively.

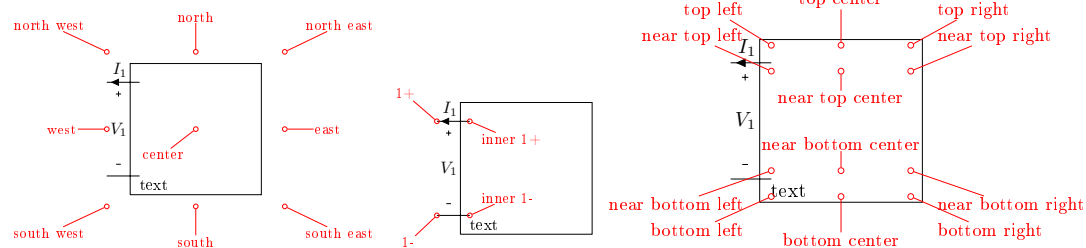
Note: For a Thevenin/Norton/Black Box, `outer x fit` and `inner x fit` will never result in a shape rotation. `y fit` might result in a shape rotation.

Note: Those keys can be used with all the following components: $\langle Quad \rangle$, $\langle Quad\ Z \rangle$, $\langle Quad\ Y \rangle$, $\langle Quad\ G \rangle$, $\langle Quad\ H \rangle$, $\langle ToQuad \rangle$, $\langle ToQuad\ Z \rangle$, $\langle ToQuad\ Y \rangle$, $\langle ToQuad\ G \rangle$, $\langle ToQuad\ H \rangle$, $\langle Black\ Box \rangle$, $\langle Thevenin \rangle$, $\langle Norton \rangle$, $\langle ToBlack\ Box \rangle$, $\langle ToThevenin \rangle$ and $\langle ToNorton \rangle$.

4.3 Black Box, BB

This is just the base shape, to be used in cases whereas one just want to emphasises part of a circuit (using, for instance, the *inner x fit to* key, or just mark a single port black box.

Note: There is also a *ToBlack Box* to be used in a *to[]* path, in which case the key *y fit to* style will be triggered with the starting and ending points of the *to[]* path.



4.3.1 Black Box keys

<i>name</i>	$\langle \text{node-name} \rangle$, when using a <i>to[]</i> path.
<i>I1</i>	Initial value: $\$I_1\$$
<i>V1</i>	Initial value: $\$V_1\$$

4.4 Example of fit to uses

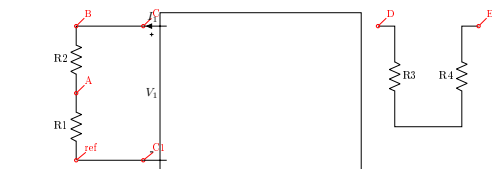
Squeezing a Black Box between two parts of a circuit (nodes C and D):

L^AT_EX Code:

```

1 \resizebox{0.4\textwidth}{!}{
2 \begin{tikzpicture}
3   \draw (0,0) coordinate(ref) \showcoord(ref)<45:0.2> to[R=R1] ++(0,2) coordinate(A) \showcoord(A)<45:0.2> to[R=R
4     2] ++(0,2) coordinate(B) \showcoord(B)<45:0.2>
5   -- ++(2,0) coordinate(C) \showcoord(C)<45:0.2> (C |- ref) coordinate(C1) \showcoord(C1)<45:0.2> -- (ref);
6   \draw (C) ++(7,0) coordinate(D) \showcoord(D)<45:0.2> -- ++(0.5,0) to[R=R3] ++(0,-3) -- ++(2,0) to[R=R4]
7     ++(0,3) -- ++(0.5,0) coordinate(E) \showcoord(E)<45:0.2>;
8   \draw (C) node[Black Box,anchor=1+,y fit to={C}{C1},outer x fit to={C}{D}]{};
9 \end{tikzpicture}
10 }
```

L^AT_EX Result:



Fitting some circuit inside the Black Box (nodes C and E):

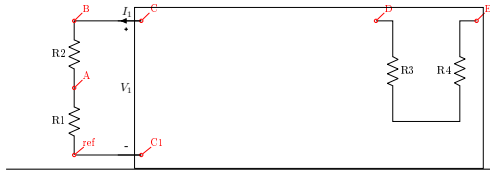
LaTeX Code:

```

1 \resizebox{0.4\textwidth}{!}{
2 \begin{tikzpicture}
3   \draw (0,0) coordinate(ref) \showcoord(ref)<45:0.2> to[R=R1] ++(0,2) coordinate(A) \showcoord(A)<45:0.2> to[R=R
4     2] ++(0,2) coordinate(B) \showcoord(B)<45:0.2>
5   -- ++(2,0) coordinate(C) \showcoord(C)<45:0.2> (C |- ref) coordinate(C1) \showcoord(C1)<45:0.2> -- (ref);
6   \draw (C) ++(7,0) coordinate(D) \showcoord(D)<45:0.2> -- ++(0.5,0) to[R=R3] ++(0,-3) -- ++(2,0) to[R=R4]
7     ++(0,3) -- ++(0.5,0) coordinate(E) \showcoord(E)<45:0.2>;
8   \draw (C) node[Black Box,anchor=inner 1+,y fit to={C}{C1},inner x fit to={C}{E}]{ };
9 \end{tikzpicture}
10 }

```

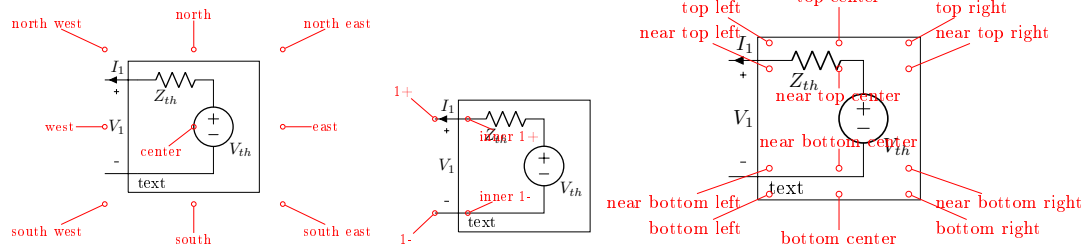
LaTeX Result:



4.5 Thevenin

This is the classical Thevenin circuit. Besides the base anchors (see 4.1) it has 2 internal nodes: `<node>-Zth` and `<node>-Vth` and each of those sub-nodes has geographic anchors as defined at 2.1.

Note: There is also a `ToThevenin` to be used in a `to[]` path, in which case the key `y fit to` style will be triggered with the starting and ending points of the `to[]` path.



4.5.1 Thevenin keys

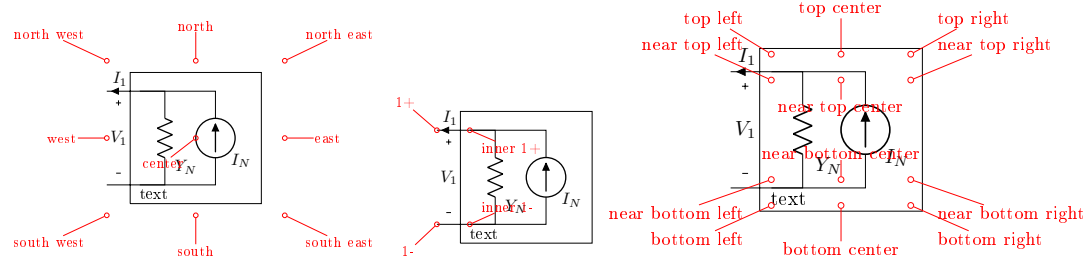
<i>name</i>	<code><node-name></code> , when using a <code>to[]</code> path.
<i>I1</i>	Initial value: <code>\$I_1\$</code>
<i>V1</i>	Initial value: <code>\$V_1\$</code>
<i>Zth</i>	Initial value: <code>\$Z_{th}\$</code>
<i>Vth</i>	Initial value: <code>\$V_{th}\$</code>
<i>Zth label pos</i>	changes the label position. Defaults to: <code>{south west}{top left}</code>
<i>Vth label pos</i>	changes the label position. Defaults to: <code>{south east}{top left}</code>

Note: The label pos keys expects two anchor names (... label pos=`{<anchor A>}{<anchor B>}`). The first anchors refers the sub-shape node and the second anchor is the text one.

4.6 Norton

This is the classical Norton circuit. Besides the base anchors (see 4.1) it has 2 internal nodes: `<node>-Yn` and `<node>-In` and each of those sub-nodes has geographic anchors as defined at 2.1.

Note: There is also a `ToNorton` to be used in a `to[]` path, in which case the key `y fit to` style will be triggered with the starting and ending points of the `to[]` path.



4.6.1 Norton keys

<code>name</code>	<code><node-name></code> , when using a <code>to[]</code> path.
<code>I1</code>	Initial value: <code>\$I_1\$</code>
<code>V1</code>	Initial value: <code>\$V_1\$</code>
<code>Yn</code>	Initial value: <code>\$Y_{N}\$</code>
<code>In</code>	Initial value: <code>\$I_{N}\$</code>
<code>Yn label pos</code>	changes the label position. Defaults to: <code>{south west}{top left}</code>
<code>In label pos</code>	changes the label position. Defaults to: <code>{south east}{top left}</code>

Note: The label pos keys expects two anchor names (... label pos=`{<anchor A>}{<anchor B>}`). The first anchors refers the sub-shape node and the second anchor is the text one.