1 Flip Flops

The syntax for all logical base circuits is

`logic[<options>](<originX,originY>){Label}`

where the options and the origin are optional. If they are missing, then the default options, described in the next section and the default origin $(0,0)$ is used. The origin specifies the lower left corner of the logical circuit.

```
logic{Demo}
logic[logicType=and]{Demo}
logic(0,0){Demo}
logic[logicType=and](0,0){Demo}
```

The above four ,,different“ calls of the `logic` macro give the same output, because they are equivalent.

1.1 The Options

`logicShowNode` **(boolean):** (*default: false*)
`logicShowDot` **(boolean):** (*default: false*)
`logicNodestyle` **(command):** (*default: \footnotesize*)
`logicSymbolstyle` **(command):** (*default: \large*)
`logicSymbolpos` **(value):** (*default: 0.5*)
`logicLabelstyle` **(command):** (*default: \small*)
`logicType` **(string):** (*default: and*)
`logicChangeLR` **(boolean):** (*default: false*)
`logicWidth` **(length):** (*default: 1.5*)
`logicHeight` **(length):** (*default: 2.5*)
`logicWireLength` **(length):** (*default: 0.5*)
`logicNInput` **(number):** (*default: 2*)
`logicJInput` **(number):** (*default: 2*)
`logicKInput` **(number):** (*default: 2*)

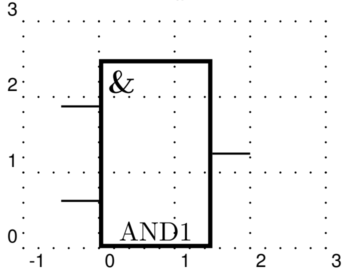
1.2 Basic Logical Circuits

At least the basic objects require a unique label name, otherwise it is not sure, that all nodes will work well. The label may contain any alphanumerical character and most of all symbols. But it is save using only combinations of letters and digits. For example:

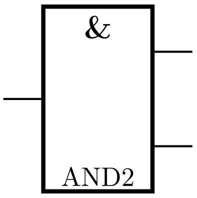
```
And0
a0
a123
12
NOT123a
```

`A_1` is not a good choice, the underscore may causes some problems.

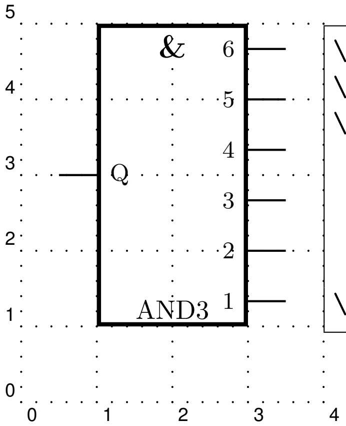
1.2.1 And



```
\begin{pspicture}(-1,0)(3,3)
\psgrid
\logic[logicSymbolpos=0.2]{AND1}
\end{pspicture}
```

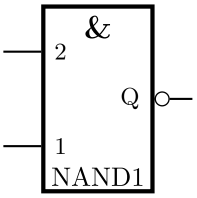


```
\begin{pspicture}(-0.5,0)(3,3)
\logic[logicChangeLR=true]{AND2}
\end{pspicture}
```

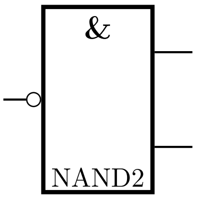


```
\begin{pspicture}(-0.5,0)(4,5)
\psgrid
\logic[logicShowNode=true,%
   logicWidth=2,%
   logicHeight=4,%
   logicNInput=6,%
   logicChangeLR=true](1,1){AND3}
\end{pspicture}
```

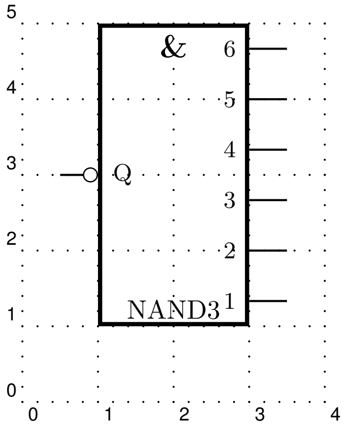
1.2.2 NotAnd



```
\begin{pspicture}(-0.5,0)(3,3)
\logic[logicType=nand,%
   logicShowNode=true]{NAND1}
\end{pspicture}
```

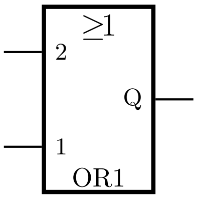


```
\begin{pspicture}(-0.5,0)(3,3)
\logic[logicType=nand,%
   logicChangeLR=true]{NAND2}
\end{pspicture}
```

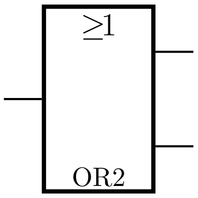


```
\begin{pspicture}(4,5)
\psgrid
\logic[logicType=nand,%
   logicShowNode=true,%
   logicWidth=2,%
   logicHeight=4,%
   logicNInput=6,%
   logicChangeLR=true](1,1){NAND3}
\end{pspicture}
```

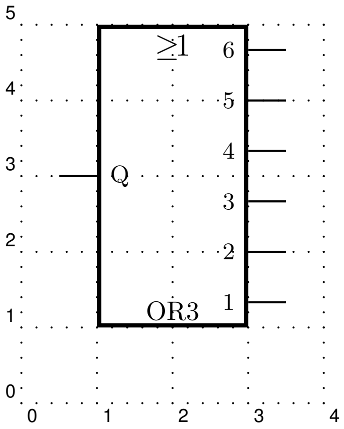
1.2.3 Or



```
\begin{pspicture}(-0.5,0)(3,3)
\logic[logicType=or,%
   logicShowNode=true]{OR1}
\end{pspicture}
```

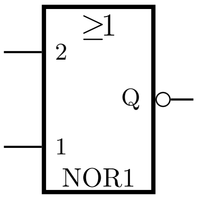


```
\begin{pspicture}(-0.5,0)(3,3)
\logic[logicType=or,%
   logicChangeLR=true]{OR2}
\end{pspicture}
```

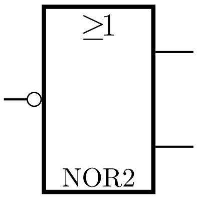


```
\begin{pspicture}(4,5)
\psgrid
\logic[logicType=or,%
   logicShowNode=true,%
   logicWidth=2,%
   logicHeight=4,%
   logicNInput=6,%
   logicChangeLR=true](1,1){OR3}
\end{pspicture}
```

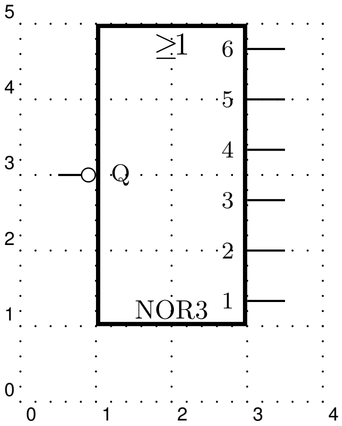
1.2.4 Not Or



```
\begin{pspicture}(-0.5,0)(3,3)
\logic[logicType=nor,%
   logicShowNode=true]{NOR1}
\end{pspicture}
```

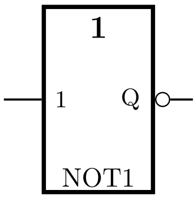


```
\begin{pspicture}(-0.5,0)(3,3)
\logic[logicType=nor,%
   logicChangeLR=true]{NOR2}
\end{pspicture}
```

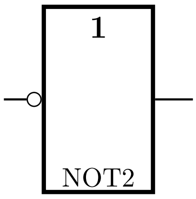


```
\begin{pspicture}(4,5)
\psgrid
\logic[logicType=nor,%
   logicShowNode=true,%
   logicWidth=2,%
   logicHeight=4,%
   logicNInput=6,%
   logicChangeLR=true](1,1){NOR3}
\end{pspicture}
```

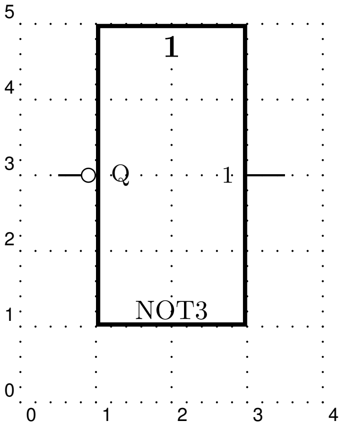
1.2.5 Not



```
\begin{pspicture}(-0.5,0)(3,3)
\logic[logicType=not,%
   logicShowNode=true]{NOT1}
\end{pspicture}
```

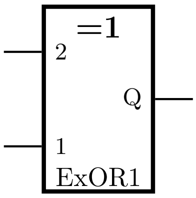


```
\begin{pspicture}(-0.5,0)(3,3)
\logic[logicType=not,%
   logicChangeLR=true]{NOT2}
\end{pspicture}
```

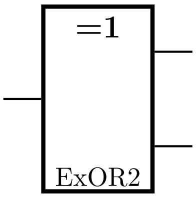


```
\begin{pspicture}(4,5)
\psgrid
\logic[logicType=not,%
   logicShowNode=true,%
   logicWidth=2,%
   logicHeight=4,%
   logicChangeLR=true](1,1){NOT3}
\end{pspicture}
```

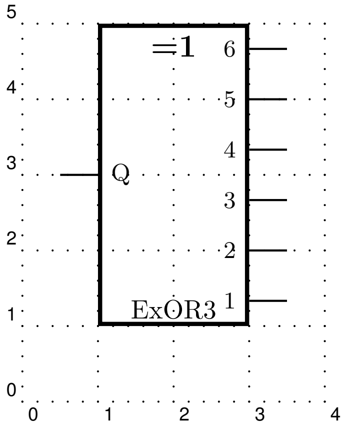
1.2.6 Exclusive OR



```
\begin{pspicture}(-0.5,0)(3,3)
\logic[logicType=exor,%
   logicShowNode=true]{ExOR1}
\end{pspicture}
```

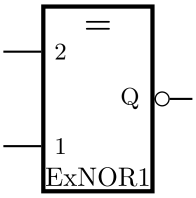


```
\begin{pspicture}(-0.5,0)(3,3)
\logic[logicType=exor,%
   logicChangeLR=true]{ExOR2}
\end{pspicture}
```

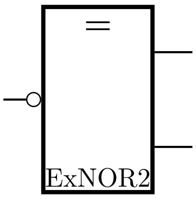


```
\begin{pspicture}(4,5)
\psgrid
\logic[logicType=exor,%
   logicShowNode=true,%
   logicNInput=6,%
   logicWidth=2,%
   logicHeight=4,%
   logicChangeLR=true](1,1){ExOR3}
\end{pspicture}
```

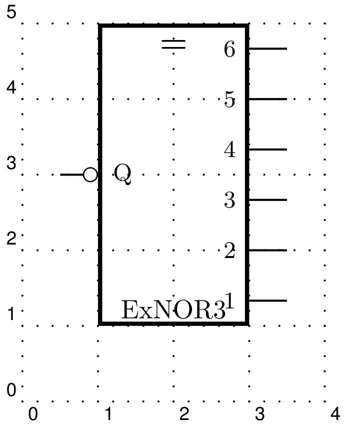
1.2.7 Exclusive NOR



```
\begin{pspicture}(-0.5,0)(3,3)
\logic[logicType=exnor,%
   logicShowNode=true]{ExNOR1}
\end{pspicture}
```

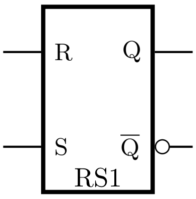


```
\begin{pspicture}(-0.5,0)(3,3)
\logic[logicType=exnor,%
   logicChangeLR=true]{ExNOR2}
\end{pspicture}
```

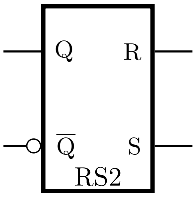


```
\begin{pspicture}(4,5)
\psgrid
\logic[logicType=exnor,%
   logicShowNode=true,%
   logicNInput=6,%
   logicWidth=2,%
   logicHeight=4,%
   logicChangeLR=true](1,1){ExNOR3}
\end{pspicture}
```

1.3 RS Flip Flop

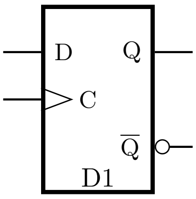


```
\begin{pspicture}(-1,-1)(3,3)
\logic[logicShowNode=true,%
   logicType=RS]{RS1}
\end{pspicture}
```

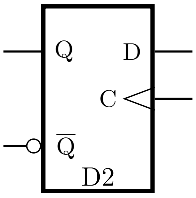


```
\begin{pspicture}(-1,-1)(3,3)
\logic[logicShowNode=true,%
   logicType=RS,%
   logicChangeLR=true]{RS2}
\end{pspicture}
```

1.4 D Flip Flop

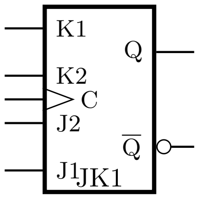


```
\begin{pspicture}(-1,-1)(3,3)
\logic[logicShowNode=true,%
   logicType=D]{D1}
\end{pspicture}
```

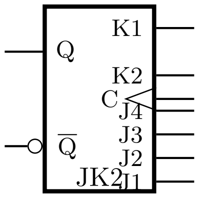


```
\begin{pspicture}(-1,-1)(3,3)
\logic[logicShowNode=true,%
   logicType=D,%
   logicChangeLR=true]{D2}
\end{pspicture}
```

1.5 JK Flip Flop

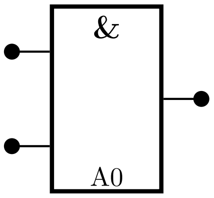


```
\begin{pspicture}(-1,-1)(3,3)
\logic[logicShowNode=true,%
   logicType=JK,%
   logicKInput=2,%
   logicJInput=2]{JK1}
\end{pspicture}
```

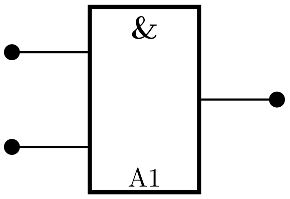


```
\begin{pspicture}(-1,-1)(3,3)
\logic[logicShowNode=true,%
   logicType=JK,%
   logicKInput=2, logicJInput=4,%
   logicChangeLR=true]{JK2}
\end{pspicture}
```

1.6 Other Options



```
\begin{pspicture}(-0.5,0)(3,2.5)
\logic[logicShowDot=true]{A0}
\end{pspicture}
```



```
\begin{pspicture}(-1,0)(3,2.5)
\logic[logicWireLength=1,%
   logicShowDot=true]{A1}
\end{pspicture}
```

The unit of `logicWireLength` is the same than the actual one for pstricks, set by the `unit` option.

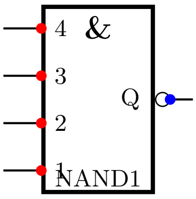
1.7 The Node Names

Every logic circuit is defined with its name, which should be a unique one. If we have the following NAND circuit, then `pst-circ` defines the nodes

`NAND11, NAND12, NAND13, NAND14, NAND1Q`

If there exists an inverted output, like for alle Flip Flops, then the negated one gets the appendix `neg` to the node name. For example:

`NAND1Q, NAND1Qneg`

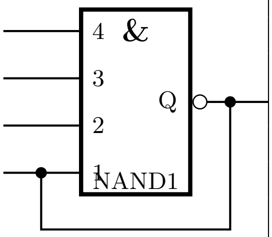


```
\begin{pspicture}(-0.5,0)(2.5,3)
\logic[logicShowNode=true,%
    logicLabelstyle=\footnotesize,%
    logicType=nand,%
    logicNInput=4]{NAND1}
\multido{\n=1+1}{4}{%
   \pscircle*[linecolor=red](NAND1\n){2pt}%
}
\pscircle*[linecolor=blue](NAND1Q){2pt}
\end{pspicture}
```

Now it is possible to draw a line from the output to the input

```
\ncbar[angleA=0,angleB=180]{<Node A>}{<Node B>}
```

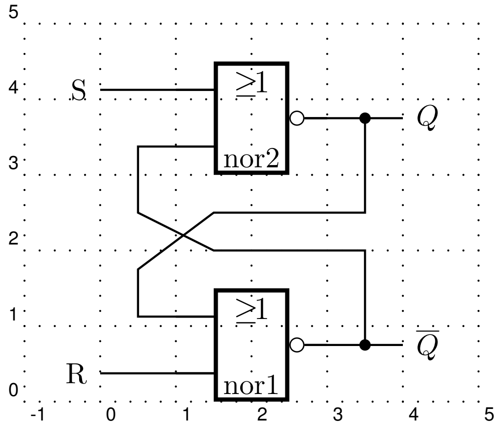
It may be easier to print a grid since the drawing phase and then comment it out if all is finished.



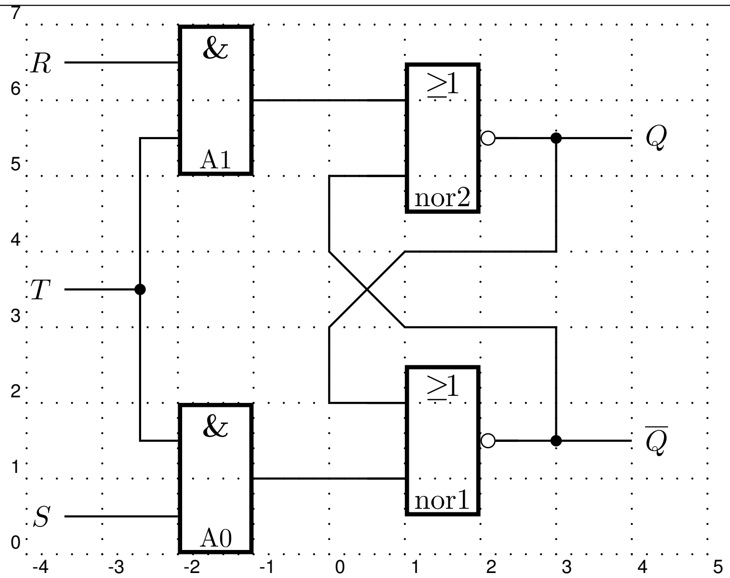
```
\begin{pspicture}(-1,-1)(2.5,3)
\logic[logicShowNode=true,%
    logicLabelstyle=\footnotesize,%
    logicType=nand,%
    logicWireLength=1,%
    logicNInput=4]{NAND1}
    \pnode(-0.5,0|NAND11){tempA}
    \pnode(2,0|NAND1Q){tempB}
\end{pspicture}
\ncbar[angleA=-90,angleB=0,arm=0.75,%
    arrows=*-*, dotsize=0.15]{tempA}{tempB}
```

1.8 Examples

```
\begin{pspicture}(-1,0)(5,5)
  \psgrid
  \psset{logicType=nor, logicLabelstyle=\normalsize,%
       logicWidth=1, logicHeight=1.5, dotsize=0.15}
  \logic(1.5,0){nor1}
  \logic(1.5,3){nor2}
  \psline(nor2Q)(4,0|nor2Q)
  \uput[0](4,0|nor2Q){$Q$}
  \psline(nor1Q)(4,0|nor1Q)
  \uput[0](4,0|nor1Q){$\overline{Q}$}
  \psline{*-}(3.50,0|nor2Q)(3.5,2.5)(1.5,2.5)
      (0.5,1.75)(0.5,0|nor12)(nor12)
  \psline{*-}(3.50,0|nor1Q)(3.5,2)(1.5,2)
      (0.5,2.5)(0.5,0|nor21)(nor21)
  \psline(0,0|nor11)(nor11)\uput[180](0,0|nor11){R}
  \psline(0,0|nor22)(nor22)\uput[180](0,0|nor22){S}
\end{pspicture}
```



```
\begin{pspicture}(-4,0)(5,7)
   \psgrid
   \psset{logicWidth=1, logicHeight=2, dotsize=0.15}
   \logic[logicWireLength=0](-2,0){A0}
   \logic[logicWireLength=0](-2,5){A1}
   \ncbar[angleA=-180,angleB=-180,arm=0.5]{A11}{A02}
   \psline[dotsize=0.15]{-*}(-3.5,3.5)(-2.5,3.5)
   \uput[180](-3.5,3.5){$T$}
   \psline(-3.5,0.5)(A01)\uput[180](-3.5,0.5){$S$}
   \psline(-3.5,6.5)(A12)\uput[180](-3.5,6.5){$R$}
   \psset{logicType=nor, logicLabelstyle=\normalsize}
   \logic(1,0.5){nor1}
   \logic(1,4.5){nor2}
   \psline(nor2Q)(4,0|nor2Q)
   \uput[0](4,0|nor2Q){$Q$}
   \psline(nor1Q)(4,0|nor1Q)
   \uput[0](4,0|nor1Q){$\overline{Q}$}
   \psline{*-}(3,0|nor2Q)(3,4)(1,4)(0,3)(0,0|nor12)(nor12)
   \psline{*-}(3,0|nor1Q)(3,3)(1,3)(0,4)(0,0|nor21)(nor21)
   \psline(A0Q)(nor11)
   \psline(A1Q)(nor22)
\end{pspicture}
```


7
6
5
4
3
2
1
0
-4
-3
-2
-1
0
1
2
3
4
5
R
T
S
&
A1
&
A0
≥1
nor2
≥1
nor1
Q
$\overline{Q}$