MA Z X SYMBOL

See Also | Related Guides | URL

MaZX`

ZXDiagram

NEW IN 13.2

ZXDiagram[{$expr_1$, $expr_2$, …}]

constructs a ZX diagram from ZX expressions $expr_1$, $expr_2$, … and stores it as ZXObject.

ZXDiagram[obj_1, obj_2, …, {$expr_1$, $expr_2$, …}]

constructs a ZX diagram on top of the existing ZXObjects obj_1, obj_2, ….

Details and Options

- Valid ZX expressions include Z[k_1, k_2, …][$phase$] for the Z spiders, X[k_1, k_2, …][$phase$] for the X spiders, H[k_1, k_2, …] for the Hadamard gates, B[k_1, k_2, …] for the diamond gates.
- The links between spiders take form Z[i_1, i_2, …][$phase$] → Z[k_1, k_2, …][$phase$], Z[i_1, i_2, …][$phase$] → X[k_1, k_2, …][$phase$], X[i_1, i_2, …][$phase$] → Z[k_1, k_2, …][$phase$], X[i_1, i_2, …][$phase$] → X[k_1, k_2, …][$phase$]. The phase part [$phase$] in spider specifications may be dropped if the same spider appears with a phase value somewhere in the ZX expression.
- The Hadamard gates may also be involved in ZX links. However, it can one and only one incoming and outgoing link.
- To generate ZX links, Q3 function Chain may be useful.
- Within ZXDiagram, Rule [→], is regarded as Chain. This simplifies the DirectedEdge specifications in a ZX expression. For example, $a \to b \to c \to \ldots$ is equivalent to Chain[a, b, c, …], which generates a series of DirectedEdges {$a \to b$, $b \to c$, $c \to \ldots$}.

Examples (22)

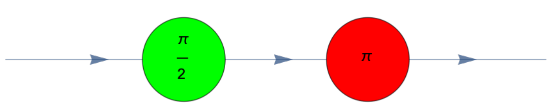
In[1]:=
```
Needs["MaZX`"]
```

Basic Examples (7)

Here is a simple example of ZX diagram. Note that how specifying DirectedEdges have been simplified by using Chain.

In[1]:=
```
ZXDiagram[{Chain[i, Z[1][Pi/2], X[1][Pi], o]}]
```

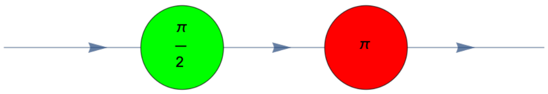
Out[1]=



Specifying the ZX links may be simplified even further by using $a \to b \to c \to \ldots$. Note that this works only within ZXDiagram.

In[2]:=
```
ZXDiagram[{i -> Z[1][Pi/2] -> X[1][Pi] -> o}]
```

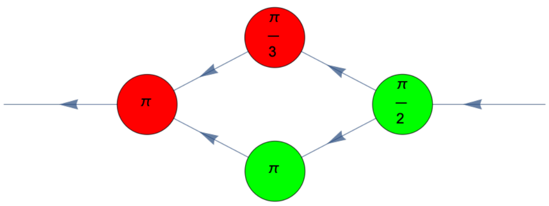
Out[2]=



Using … → {a, b} → … = {… → a → …, … → b → …} is another method for simplifying your specifications.

In[3]:=
```
ZXDiagram[{i -> Z[1][Pi/2] -> {Z[2][Pi], X[2][Pi/3]} -> X[1][Pi] -> o},
 GraphLayout -> "SpringElectricalEmbedding"]
```

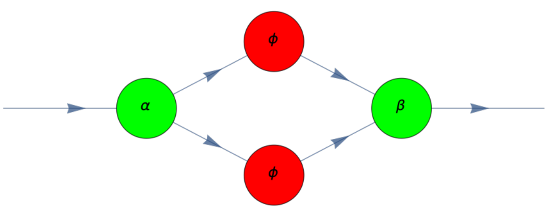
Out[3]=



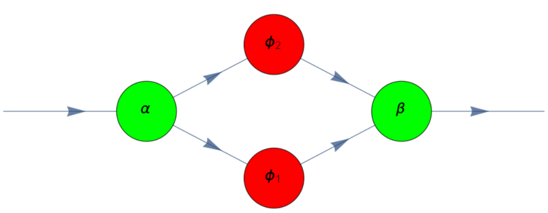
Here is another shortcut: Within ZXDiagram , X[{k_1, k_2, …, k_n}][ϕ] is expanded to {X[k_1][ϕ], X[k_2][ϕ], …, X[k_n][ϕ]}. Similarly, X[{k_1, k_2, …, k_n}][{ϕ_1, ϕ_2, …, ϕ_n}] = {X[k_1][ϕ_1], …, X[k_n][ϕ_n]}. Note that this is only the case within ZXDiagram .

In[4]:=
```
ZXDiagram[{i -> Z[1][α] -> X[{1, 2}][ϕ] -> Z[2][β] -> o}, GraphLayout -> "SpringElectricalEmbedding"]
ZXDiagram[{i -> Z[1][α] -> X[{1, 2}][{ϕ1, ϕ2}] -> Z[2][β] -> o}, GraphLayout -> "SpringElectricalEmbedding"]
```

Out[4]=

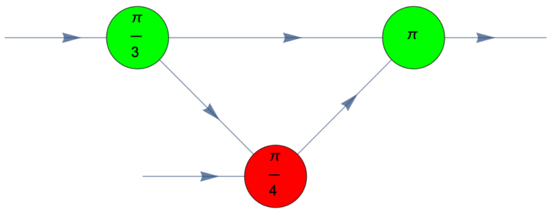


Out[4]=



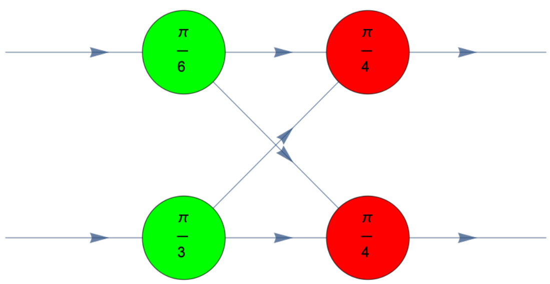
Here, note that the second Z spider does not have a phase specification. This is because the phase value π must be clear from the previous specification `Z[1][Pi]` of the same spider.

In[5]:=
```
ZXDiagram@{
  ii -> X[1][Pi/4] -> Z[1][Pi] -> o,
  i2 -> Z[2][Pi/3] -> Z[1],
  Z[2] -> X[1]
 }
```

Out[5]=


Consider a ZX diagram.

In[1]:=
```
obj = ZXDiagram@{
   i1 -> Z[1][Pi/3] -> {X[1][Pi/4], X[2][Pi/4]},
   i2 -> Z[2][Pi/6] -> {X[1], X[2]},
   X[1] -> o1,
   X[2] -> o2
  }
```

Out[1]=


Calculate the corresponding operator expression.

In[2]:=
```
op = ExpressionFor[obj]
```

Out[2]=

$$\left(\frac{1}{8}+\frac{i}{8}\right)\left(1+\sqrt{2}\right)\left|0_{o1}0_{o2}\right\rangle\left\langle 0_{i1}0_{i2}\right| + \frac{1}{8}(-1)^{1/6}\left((1+i)-2(-1)^{1/4}\right)\left|0_{o1}0_{o2}\right\rangle\left\langle 0_{i1}1_{i2}\right| +$$
$$\frac{1}{8}(-1)^{1/3}\left((1+i)-2(-1)^{1/4}\right)\left|0_{o1}0_{o2}\right\rangle\left\langle 1_{i1}0_{i2}\right| - \left(\frac{1}{8}-\frac{i}{8}\right)\left(1+\sqrt{2}\right)\left|0_{o1}0_{o2}\right\rangle\left\langle 1_{i1}1_{i2}\right| + \left(\frac{1}{8}-\frac{i}{8}\right)\left|0_{o1}1_{o2}\right\rangle\left\langle 0_{i1}0_{i2}\right| +$$
$$\left(\frac{1}{8}-\frac{i}{8}\right)(-1)^{1/6}\left|0_{o1}1_{o2}\right\rangle\left\langle 0_{i1}1_{i2}\right| + \left(\frac{1}{8}-\frac{i}{8}\right)(-1)^{1/3}\left|0_{o1}1_{o2}\right\rangle\left\langle 1_{i1}0_{i2}\right| + \left(\frac{1}{8}+\frac{i}{8}\right)\left|0_{o1}1_{o2}\right\rangle\left\langle 1_{i1}1_{i2}\right| +$$
$$\left(\frac{1}{8}-\frac{i}{8}\right)\left|1_{o1}0_{o2}\right\rangle\left\langle 0_{i1}0_{i2}\right| + \left(\frac{1}{8}-\frac{i}{8}\right)(-1)^{1/6}\left|1_{o1}0_{o2}\right\rangle\left\langle 0_{i1}1_{i2}\right| + \left(\frac{1}{8}-\frac{i}{8}\right)(-1)^{1/3}\left|1_{o1}0_{o2}\right\rangle\left\langle 1_{i1}0_{i2}\right| +$$
$$\left(\frac{1}{8}+\frac{i}{8}\right)\left|1_{o1}0_{o2}\right\rangle\left\langle 1_{i1}1_{i2}\right| - \left(\frac{1}{8}+\frac{i}{8}\right)\left(-1+\sqrt{2}\right)\left|1_{o1}1_{o2}\right\rangle\left\langle 0_{i1}0_{i2}\right| + \left(\frac{1}{8}+\frac{i}{8}\right)(-1)^{1/6}\left(1+\sqrt{2}\right)\left|1_{o1}1_{o2}\right\rangle\left\langle 0_{i1}1_{i2}\right| +$$
$$\left(\frac{1}{8}+\frac{i}{8}\right)(-1)^{1/3}\left(1+\sqrt{2}\right)\left|1_{o1}1_{o2}\right\rangle\left\langle 1_{i1}0_{i2}\right| + \left(\frac{1}{8}-\frac{i}{8}\right)\left(-1+\sqrt{2}\right)\left|1_{o1}1_{o2}\right\rangle\left\langle 1_{i1}1_{i2}\right|$$

Calculate the matrix representation of the ZX diagram.

In[3]:=
```
mat = Matrix[obj] // N;
MatrixForm[mat]
```

Out[3]//MatrixForm=

$$\begin{pmatrix} 0.301777 + 0.301777\,i & -0.0189516 - 0.0707283\,i & 0.0189516 - 0.0707283\,i & -0.301777 + 0.301777\,i \\ 0.125 - 0.125\,i & 0.170753 - 0.0457532\,i & 0.170753 + 0.0457532\,i & 0.125 + 0.125\,i \\ 0.125 - 0.125\,i & 0.170753 - 0.0457532\,i & 0.170753 + 0.0457532\,i & 0.125 + 0.125\,i \\ -0.0517767 - 0.0517767\,i & 0.110458 + 0.412235\,i & -0.110458 + 0.412235\,i & 0.0517767 - 0.0517767\,i \end{pmatrix}$$

Of course, the above matrix must be the same as the one from the operator expression.

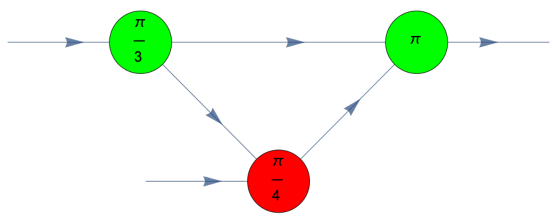
In[4]:=
```
mat - Matrix[op] // N // Chop // MatrixForm
```

Out[4]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

In[1]:=
```
obj = ZXDiagram@{
   ii -> X[1][Pi/4] -> Z[1][Pi] -> o,
   i2 -> Z[2][Pi/3] -> Z[1],
   Z[2] -> X[1]
  }
op = ExpressionFor[obj]
mat = Matrix[obj];
mat // N // MatrixForm
```

Out[1]=



Out[1]=

$$\frac{1}{4}\left((1+i)+\sqrt{2}\right)\left|0_o\right\rangle\left\langle 0_{i2}0_{ii}\right| + \frac{1}{4}\left((-1-i)+\sqrt{2}\right)\left|0_o\right\rangle\left\langle 0_{i2}1_{ii}\right| - \frac{(-1)^{1/3}\left(1+(-1)^{1/4}\right)\left|1_o\right\rangle\left\langle 1_{i2}0_{ii}\right|}{2\sqrt{2}} + \frac{(-1)^{1/3}\left(-1+(-1)^{1/4}\right)\left|1_o\right\rangle\left\langle 1_{i2}1_{ii}\right|}{2\sqrt{2}}$$

Out[1]//MatrixForm=

$$\begin{pmatrix} 0.603553 + 0.25\,i & 0.103553 - 0.25\,i & 0. & 0. \\ 0. & 0. & -0.0852703 - 0.647693\,i & -0.268283 + 0.0353201\,i \end{pmatrix}$$

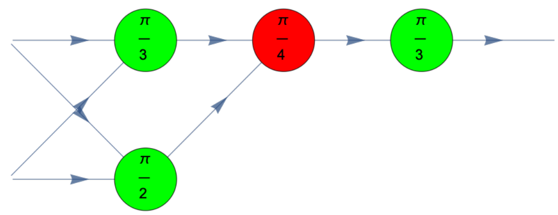
In[2]:=
```
Matrix[op] - Matrix[obj] // N // Chop // MatrixForm
```

Out[2]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

In[1]:=
```
obj = ZXDiagram@{
   Z[2][Pi/3], Z[1][Pi/2], Z[3][Pi/3],
   {i1, i2} -> Z[{1, 3}] -> X[1][Pi/4] -> Z[2] -> o
  }
op = ExpressionFor[obj]
mat = Matrix[obj];
mat // SimplifyThrough // MatrixForm
```

Out[1]=



Out[1]=

$$\frac{1}{4}\left((1+i)+\sqrt{2}\right)\left|0_o\right\rangle\left\langle 0_{i1}0_{i2}\right| + \frac{(-1)^{5/6}\left(1+(-1)^{1/4}\right)\left|0_o\right\rangle\left\langle 1_{i1}1_{i2}\right|}{2\sqrt{2}} - \frac{(-1)^{1/3}\left(-1+(-1)^{1/4}\right)\left|1_o\right\rangle\left\langle 0_{i1}0_{i2}\right|}{2\sqrt{2}} + \frac{(-1)^{1/6}\left(-1+(-1)^{1/4}\right)\left|1_o\right\rangle\left\langle 1_{i1}1_{i2}\right|}{2\sqrt{2}}$$

Out[1]//MatrixForm=

$$\begin{pmatrix} \frac{1}{4}\left((1+i)+\sqrt{2}\right) & 0 & 0 & \frac{1}{4}(-1)^{5/6}\left((1+i)+\sqrt{2}\right) \\ \frac{1}{4}(-1)^{1/3}\left((-1-i)+\sqrt{2}\right) & 0 & 0 & -\frac{1}{4}(-1)^{1/6}\left((-1-i)+\sqrt{2}\right) \end{pmatrix}$$

In[2]:=
```
Matrix[op] - Matrix[obj] // N // Chop // MatrixForm
```

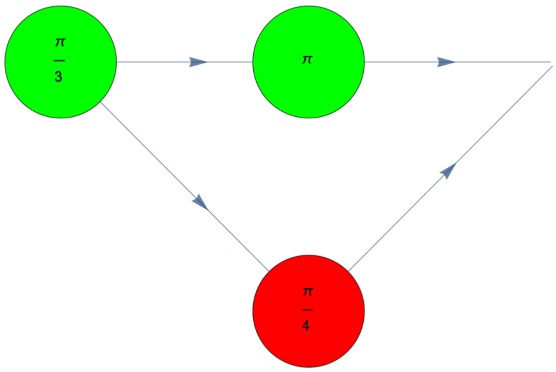
Out[2]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

In this example, the output vertex has two incident edges.

In[1]:=
```
obj = ZXDiagram@{
   Z[1][Pi/3], X[1][Pi/4], Z[2][Pi],
   Z[1] -> {X[1], Z[2]} -> o
  }
op = ExpressionFor[obj]
mat = Matrix[obj];
mat // N // MatrixForm
```

Out[1]=



Out[1]= $\frac{1}{2}\left(1+(-1)^{1/4}\right)\,|0_o\rangle-\frac{1}{2}(-1)^{1/3}\left(1+(-1)^{1/4}\right)\,|1_o\rangle$

Out[1]//MatrixForm=

$$\begin{pmatrix} 0.853553+0.353553\,i \\ -0.12059-0.915976\,i \end{pmatrix}$$

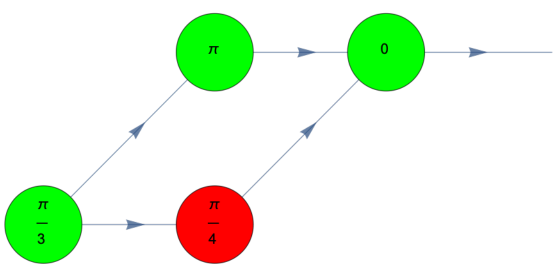
In[2]:=
```
Matrix[op] - Matrix[obj] // N // Chop // MatrixForm
```

Out[2]//MatrixForm=

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

In[1]:=
```
obj = ZXDiagram@{
   Z[1][Pi/3], X[1][Pi/4], Z[2][Pi], Z[3][0],
   Z[1] -> {X[1], Z[2]} -> Z[3] -> o
  }
op = ExpressionFor[obj]
mat = Matrix[obj];
mat // N // MatrixForm
```

Out[1]=



Out[1]= $\frac{1}{2}\left(1+(-1)^{1/4}\right)\,|0_o\rangle-\frac{1}{2}(-1)^{1/3}\left(1+(-1)^{1/4}\right)\,|1_o\rangle$

Out[1]//MatrixForm=

$$\begin{pmatrix} 0.853553+0.353553\,i \\ -0.12059-0.915976\,i \end{pmatrix}$$

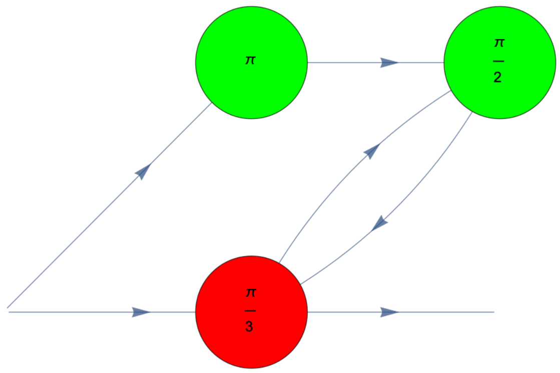
In[2]:=
```
ZXLayers[Graph@obj]
```

Out[2]= $\left\{\left\{Z_1\left(\frac{\pi}{3}\right)\right\},\ \left\{X_1\left(\frac{\pi}{4}\right),\ Z_2(\pi)\right\},\ \{Z_3(0)\},\ \{o\}\right\}$

This example shows a common mistake you can make in ZX expression. Note that just changing the phase of the spider `X[1][Pi/3]` does not fix it because there is a loop of the directed edges.

In[1]:=
```
obj = ZXDiagram@{i -> {Z[1][Pi], X[1][Pi/3]} -> Z[2][Pi/2] -> X[1][Pi/3] -> o}
op = ExpressionFor[obj]
```

Out[1]=

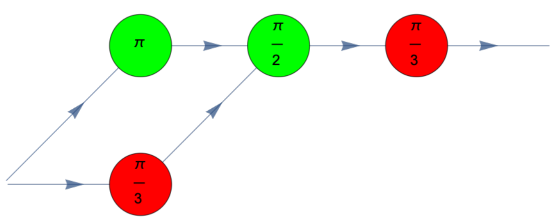


Out[1]= $\frac{1}{2}\left|+_4+_6 0_o\right\rangle\left\langle 0_4 0_6 0_i\right| - \frac{1}{2}\,i\left|+_4+_6 0_o\right\rangle\left\langle 1_4 0_6 1_i\right| + \frac{1}{2}\left|+_4+_6 1_o\right\rangle\left\langle 0_4 1_6 0_i\right| - \frac{1}{2}\,i\left|+_4+_6 1_o\right\rangle\left\langle 1_4 1_6 1_i\right| + \frac{1}{2}(-1)^{1/3}\left|-_4-_6 0_o\right\rangle\left\langle 0_4 0_6 0_i\right| - \frac{1}{2}(-1)^{5/6}\left|-_4-_6 0_o\right\rangle\left\langle 1_4 0_6 1_i\right| + \frac{1}{2}(-1)^{1/3}\left|-_4-_6 1_o\right\rangle\left\langle 0_4 1_6 0_i\right| - \frac{1}{2}(-1)^{5/6}\left|-_4-_6 1_o\right\rangle\left\langle 1_4 1_6 1_i\right|$

Maybe, this was the intended diagram.

In[2]:=
```
obj = ZXDiagram@{i -> {Z[1][Pi], X[1][Pi/3]} -> Z[2][Pi/2] -> X[2][Pi/3] -> o}
op = ExpressionFor[obj] // ToZBasis
```

Out[2]=



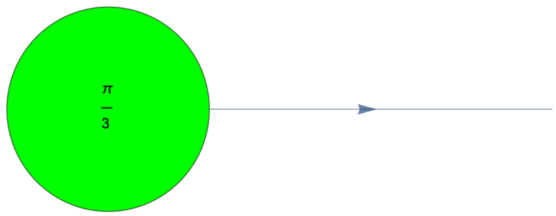
Out[2]= $\frac{1}{4}\left(1+(-1)^{1/3}\right)^2\left|0_o\right\rangle\left\langle 0_i\right| + \frac{1}{4}\,i\left(-1+(-1)^{2/3}\right)\left|0_o\right\rangle\left\langle 1_i\right| + \frac{1}{4}\left(1-(-1)^{2/3}\right)\left|1_o\right\rangle\left\langle 0_i\right| + \frac{3}{8}\left(-i+\sqrt{3}\right)\left|1_o\right\rangle\left\langle 1_i\right|$

Scope (9)

Either input or output (but not both) (1)

In[1]:=
```
obj = ZXDiagram@{Z[1][Pi/3] -> o}
```

Out[1]=

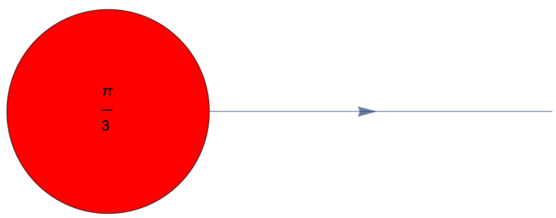


In[2]:=
```
op = ExpressionFor[obj]
```

Out[2]= $|0_o\rangle + (-1)^{1/3}\ |1_o\rangle$

In[3]:=
```
obj = ZXDiagram@{X[1][Pi/3] -> o}
op = ExpressionFor[obj] // ToXBasis
```

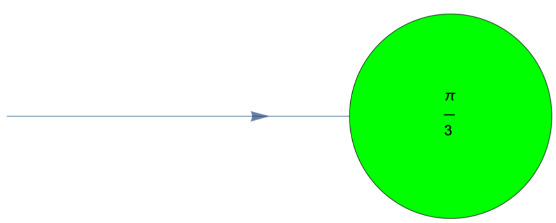
Out[3]=



Out[3]= $|+_o\rangle + \frac{1}{2}\left(1 + i\ \sqrt{3}\right)\ |-_o\rangle$

In[4]:=
```
obj = ZXDiagram@{i -> Z[1][Pi/3]}
op = ExpressionFor[obj]
```

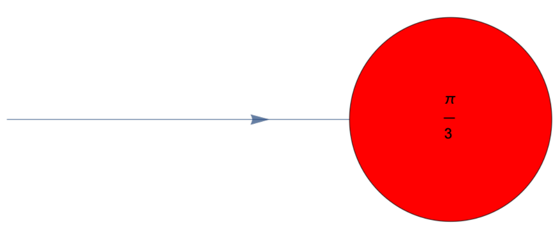
Out[4]=



Out[4]= $\langle 0_i| + (-1)^{1/3}\ \langle 1_i|$

In[5]:=
```
obj = ZXDiagram@{i -> X[1][Pi/3]}
op = ExpressionFor[obj] // ToXBasis
```

Out[5]=

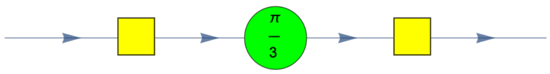


Out[5]= $\langle +_i| + \frac{1}{2}\left(1 + i\ \sqrt{3}\right)\ \langle -_i|$

Hadamard gate (2)

In[1]:=
```
obj = ZXDiagram@{i -> H[1] -> Z[1][Pi/3] -> H[2] -> o}
```

Out[1]=



In[2]:=
```
op = ExpressionFor[obj]
```

Out[2]= $\frac{1}{2}\left(1+(-1)^{1/3}\right)\,|0_o\rangle\langle 0_i| + \frac{1}{2}\left(1-(-1)^{1/3}\right)\,|0_o\rangle\langle 1_i| + \frac{1}{2}\left(1-(-1)^{1/3}\right)\,|1_o\rangle\langle 0_i| + \frac{1}{2}\left(1+(-1)^{1/3}\right)\,|1_o\rangle\langle 1_i|$

In[3]:=
```
ToXBasis[op]
ToXBasis[op, {o}]
```

Out[3]= $|+_o\rangle\langle +_i| + \frac{1}{2}\left(1+i\sqrt{3}\right)\,|-_o\rangle\langle -_i|$

Out[3]= $\frac{|+_o\rangle\langle 0_i|}{\sqrt{2}} + \frac{|+_o\rangle\langle 1_i|}{\sqrt{2}} + \frac{(-1)^{1/3}\,|-_o\rangle\langle 0_i|}{\sqrt{2}} - \frac{(-1)^{1/3}\,|-_o\rangle\langle 1_i|}{\sqrt{2}}$

In[4]:=
```
mat = Matrix[obj] // SimplifyThrough;
mat // MatrixForm
```

Out[4]//MatrixForm=

$$\begin{pmatrix} \frac{1}{2}\left(1+(-1)^{1/3}\right) & \frac{1}{2}\left(1-(-1)^{1/3}\right) \\ \frac{1}{2}\left(1-(-1)^{1/3}\right) & \frac{1}{2}\left(1+(-1)^{1/3}\right) \end{pmatrix}$$

Let us compare the above result with the usual algebraic calculation.

In[5]:=
```
Let[Qubit, S]
op = Phase[Pi/3, S[3]]
```

Out[5]= $S^z\left(\frac{\pi}{3}\right)$

In[6]:=
```
new = S[6] ** op ** S[6] // Matrix // Simplify;
new // MatrixForm
```

Out[6]//MatrixForm=

$$\begin{pmatrix} \frac{1}{4}\left(3+i\sqrt{3}\right) & \frac{1}{4}\left(1-i\sqrt{3}\right) \\ \frac{1}{4}\left(1-i\sqrt{3}\right) & \frac{1}{4}\left(3+i\sqrt{3}\right) \end{pmatrix}$$

In[7]:=
```
new - mat // Simplify // MatrixForm
```

Out[7]//MatrixForm=

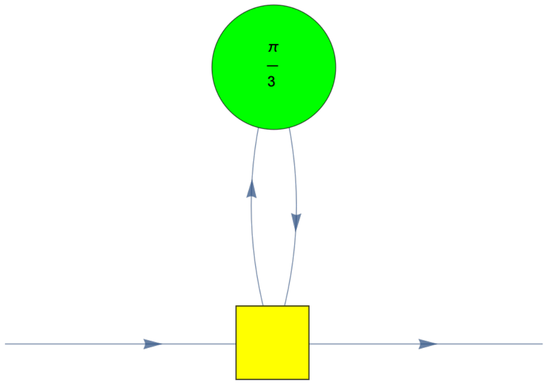
$$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

In a ZX expression, the Hadamard gate can have one and only one input and output links.

In[1]:=
```
obj = ZXDiagram@{i -> H[1] -> Z[1][Pi/3] -> H[1] -> o}
```

ZXDiagram: Wrong arities for some Hadamard gates: $\{H_1 \to \{2, 2\}\}$. Every Hadamard gate should have one and only one input and output link.

Out[1]=

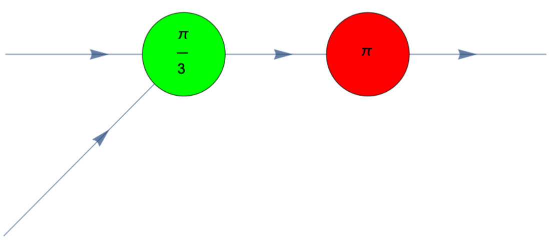


Combining two ZXObjects (3)

Here is one ZX diagram.

In[1]:=
```
obj1 = ZXDiagram[{{i1, i2} -> Z[1][Pi/3] -> X[1][Pi] -> o}]
op1 = ExpressionFor[obj1]
mat1 = Matrix[obj1];
mat1 // MatrixForm
```

Out[1]=



Out[1]= $(-1)^{1/3} \left|0_o\right\rangle\left\langle 1_{i1} 1_{i2}\right| + \left|1_o\right\rangle\left\langle 0_{i1} 0_{i2}\right|$

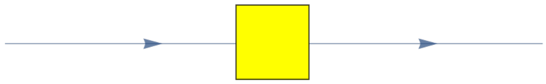
Out[1]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & e^{\frac{i\pi}{3}} \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

Here is another one.

In[2]:=
```
obj2 = ZXDiagram[{i3 -> H[1] -> o2}, VertexLabels -> None]
op2 = ExpressionFor[obj2] // ToXBasis[#, {o2}] &
mat2 = Matrix[obj2];
mat2 // MatrixForm
```

Out[2]=



Out[2]= $\left|+_{o2}\right\rangle\left\langle 0_{i3}\right| + \left|-_{o2}\right\rangle\left\langle 1_{i3}\right|$

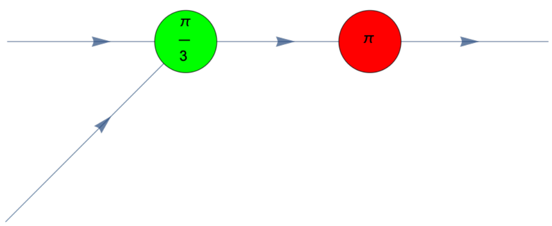
Out[2]//MatrixForm=

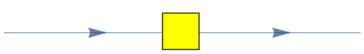
$$\begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}$$

If you want to combine the above two ZX diagrams, usually, you have to put all those ZX expression in the above two diagrams again in the new one.

In[3]:=
```
new = ZXDiagram@{
    {i1, i2} -> Z[1][Pi/3] -> X[1][Pi] -> o,
    i3 -> H[1] -> o2
   }
op = ExpressionFor[new]
```

Out[3]=





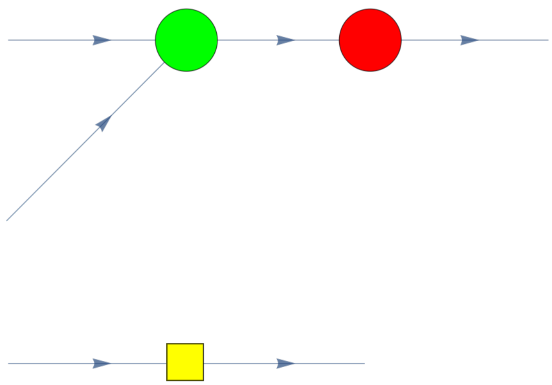
Out[3]=

$$\frac{(-1)^{1/3}\left|0_o 0_{o2}\right\rangle\left\langle 1_{i1}1_{i2}0_{i3}\right|}{\sqrt{2}} + \frac{(-1)^{1/3}\left|0_o 0_{o2}\right\rangle\left\langle 1_{i1}1_{i2}1_{i3}\right|}{\sqrt{2}} + \frac{(-1)^{1/3}\left|0_o 1_{o2}\right\rangle\left\langle 1_{i1}1_{i2}0_{i3}\right|}{\sqrt{2}} - \frac{(-1)^{1/3}\left|0_o 1_{o2}\right\rangle\left\langle 1_{i1}1_{i2}1_{i3}\right|}{\sqrt{2}} + \frac{\left|1_o 0_{o2}\right\rangle\left\langle 0_{i1}0_{i2}0_{i3}\right|}{\sqrt{2}} + \frac{\left|1_o 0_{o2}\right\rangle\left\langle 0_{i1}0_{i2}1_{i3}\right|}{\sqrt{2}} + \frac{\left|1_o 1_{o2}\right\rangle\left\langle 0_{i1}0_{i2}0_{i3}\right|}{\sqrt{2}} - \frac{\left|1_o 1_{o2}\right\rangle\left\langle 0_{i1}0_{i2}1_{i3}\right|}{\sqrt{2}}$$

To avoid, you can just combine them using Join .

In[1]:=
```
joined = Join[obj1, obj2]
```

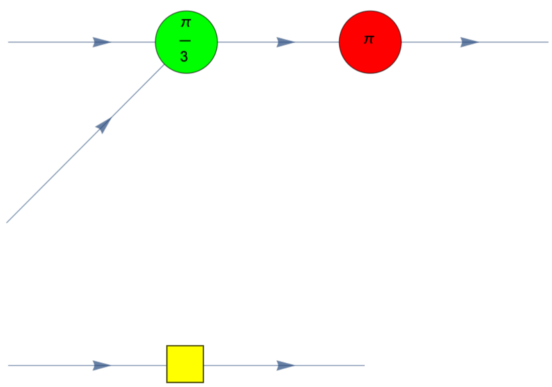
Out[1]=



If you just add to an already existing ZX diagram, then just include the previous one in the ZX expression.

In[1]:=
```
obj3 = ZXDiagram[{obj1, i3 -> H[1] -> o2}]
op3 = ExpressionFor[obj3]
mat3 = Matrix[obj3];
mat3 // MatrixForm
```

Out[1]=



Out[1]=

$$\frac{(-1)^{1/3}\,|0_{o}0_{o2}\rangle\langle 1_{i1}1_{i2}0_{i3}|}{\sqrt{2}}+\frac{(-1)^{1/3}\,|0_{o}0_{o2}\rangle\langle 1_{i1}1_{i2}1_{i3}|}{\sqrt{2}}+\frac{(-1)^{1/3}\,|0_{o}1_{o2}\rangle\langle 1_{i1}1_{i2}0_{i3}|}{\sqrt{2}}-\frac{(-1)^{1/3}\,|0_{o}1_{o2}\rangle\langle 1_{i1}1_{i2}1_{i3}|}{\sqrt{2}}+\frac{|1_{o}0_{o2}\rangle\langle 0_{i1}0_{i2}0_{i3}|}{\sqrt{2}}+\frac{|1_{o}0_{o2}\rangle\langle 0_{i1}0_{i2}1_{i3}|}{\sqrt{2}}+\frac{|1_{o}1_{o2}\rangle\langle 0_{i1}0_{i2}0_{i3}|}{\sqrt{2}}-\frac{|1_{o}1_{o2}\rangle\langle 0_{i1}0_{i2}1_{i3}|}{\sqrt{2}}$$

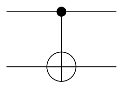
Out[1]//MatrixForm=

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{e^{\frac{i\pi}{3}}}{\sqrt{2}} & \frac{e^{\frac{i\pi}{3}}}{\sqrt{2}} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{e^{\frac{i\pi}{3}}}{\sqrt{2}} & -\frac{e^{\frac{i\pi}{3}}}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Controlled-NOT (CNOT) gate (2)

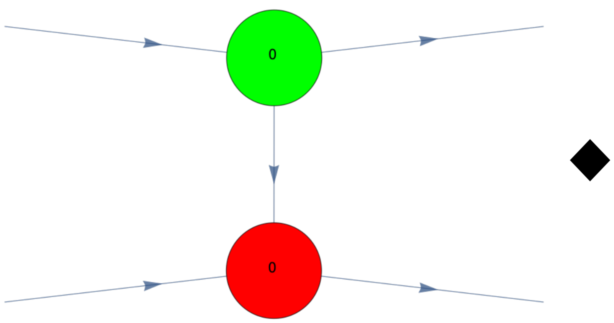
In[1]:=
```
Let[Qubit, S]
qc = QuantumCircuit[CNOT[S[1], S[2]]]
```

Out[1]=



In[2]:=
```
cnot = ZXDiagram[{B[1],
    i1 -> Z[1][0] -> o1,
    i2 -> X[2][0] -> o2,
    Z[1] -> X[2]},
   VertexCoordinates -> {
     i1 -> {-1, 1}, i2 -> {-1, 0},
     o1 -> {1, 1}, o2 -> {1, 0},
     B[1] -> {1, 1/2}}
  ]
op = ExpressionFor[cnot] // ToZBasis
mat = Matrix[cnot];
mat // MatrixForm
```

Out[2]=



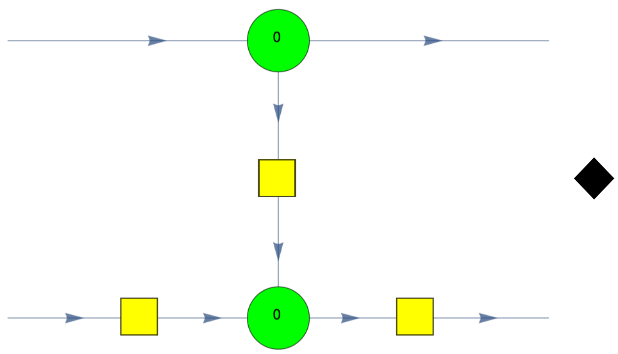
Out[2]= $|0_{o1}0_{o2}\rangle\langle 0_{i1}0_{i2}| + |0_{o1}1_{o2}\rangle\langle 0_{i1}1_{i2}| + |1_{o1}0_{o2}\rangle\langle 1_{i1}1_{i2}| + |1_{o1}1_{o2}\rangle\langle 1_{i1}0_{i2}|$

Out[2]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

In[3]:=
```
cnot = ZXDiagram[
  {Z[1][0], Z[2][0], H[1], B[1],
   i1 -> Z[1] -> o1,
   i2 -> H[2] -> Z[2] -> H[3] -> o2,
   Z[1] -> H[1] -> Z[2]},
  VertexCoordinates -> {
    i1 -> {-2, 1}, i2 -> {-2, -1},
    o1 -> {2, 1}, o2 -> {2, -1},
    Z[1] -> {0, 1}, Z[2] -> {0, -1},
    H[1] -> {0, 0}, H[2] -> {-1, -1}, H[3] -> {1, -1},
    B[1] -> {1, 0}}
 ]
op = ExpressionFor[cnot] // ToZBasis
mat = Matrix[cnot];
mat // MatrixForm
```

Out[3]=



Out[3]= $|0_{o1}0_{o2}\rangle\langle 0_{i1}0_{i2}| + |0_{o1}1_{o2}\rangle\langle 0_{i1}1_{i2}| + |1_{o1}0_{o2}\rangle\langle 1_{i1}1_{i2}| + |1_{o1}1_{o2}\rangle\langle 1_{i1}0_{i2}|$

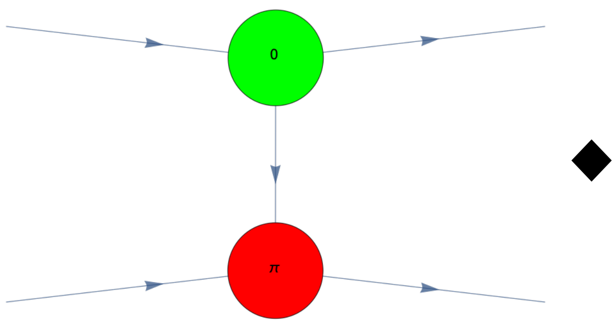
Out[3]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

An interesting variation is the following ZX diagram.

In[1]:=
```
cnot = ZXDiagram[
  {Z[1][0], X[1][Pi], B[1],
   i1 -> Z[1] -> o1,
   i2 -> X[1] -> o2,
   Z[1] -> X[1]},
  VertexCoordinates -> {
    i1 -> {-1, 1}, i2 -> {-1, 0},
    o1 -> {1, 1}, o2 -> {1, 0},
    B[1] -> {1, 1 / 2}}
 ]
op = ExpressionFor[cnot] // ToZBasis
mat = Matrix[cnot];
mat // MatrixForm
```

Out[1]=



Out[1]= $|0_{o1}0_{o2}\rangle\langle 0_{i1}1_{i2}| + |0_{o1}1_{o2}\rangle\langle 0_{i1}0_{i2}| + |1_{o1}0_{o2}\rangle\langle 1_{i1}0_{i2}| + |1_{o1}1_{o2}\rangle\langle 1_{i1}1_{i2}|$

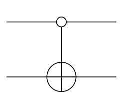
Out[1]//MatrixForm=

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

It corresponds to the following quantum circuit.

In[2]:=
```
Let[Qubit, S]
qc = QuantumCircuit[CNOT[{S[1]} -> {0}, S[2]]]
```

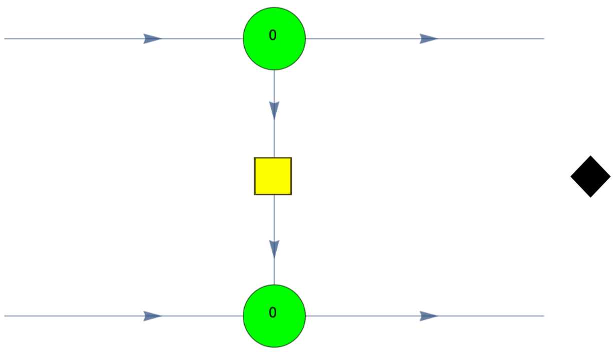
Out[2]=



Controlled-Z (CZ) gate (1)

In[1]:=
```
cz = ZXDiagram[
  {Z[1][0], Z[2][0], B[1],
   i1 -> Z[1] -> o1,
   i2 -> Z[2] -> o2,
   Z[1] -> H[1] -> Z[2]},
  VertexCoordinates -> {
    i1 -> {-1, 1}, i2 -> {-1, 0},
    o1 -> {1, 1}, o2 -> {1, 0},
    Z[1] -> {0, 1}, Z[2] -> {0, 0},
    H[1] -> {0, 1/2}, B[1] -> {1, 1/2}}
 ]
op = ExpressionFor[cz]
mat = Matrix[cz];
mat // MatrixForm
```

Out[1]=



Out[1]= $|0_{o1}0_{o2}\rangle\langle 0_{i1}0_{i2}| + |0_{o1}1_{o2}\rangle\langle 0_{i1}1_{i2}| + |1_{o1}0_{o2}\rangle\langle 1_{i1}0_{i2}| - |1_{o1}1_{o2}\rangle\langle 1_{i1}1_{i2}|$

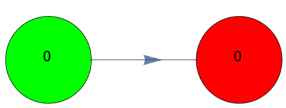
Out[1]//MatrixForm=

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Properties & Relations (6)

Simple diagrams (1)

In[1]:=
```
Row@{
  obj = ZXDiagram[{Z[1][0] -> X[1][0]}, ImageSize -> Small],
  Style[" = ", Large], ExpressionFor[obj]
 }
```

Out[1]= 

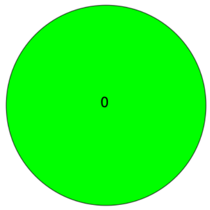
 = $\sqrt{2}$

In[2]:=
```
Row@{
  obj = ZXDiagram[{B[1]}, ImageSize -> Small], Style[" = ", Large],
  ExpressionFor[obj]
 }
```

Out[2]=  = $\sqrt{2}$

In[3]:=
```
Row@{
  obj = ZXDiagram[{Z[1][0]}, ImageSize -> Small], Style[" = ", Large],
  ExpressionFor[obj]
 }
```

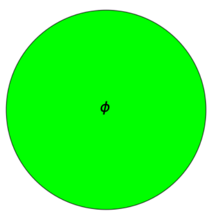
Out[3]=



= 2

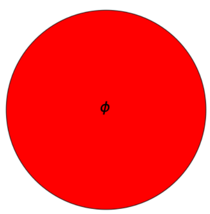
In[4]:=
```
Row@{
  obj = ZXDiagram[{Z[1][ϕ]}, ImageSize -> Small], Style[" = ", Large],
  ExpressionFor[obj]
 }
Row@{
  obj = ZXDiagram[{X[1][ϕ]}, ImageSize -> Small], Style[" = ", Large],
  ExpressionFor[obj]
 }
```

Out[4]=



= $1 + e^{i\,\phi}$

Out[4]=

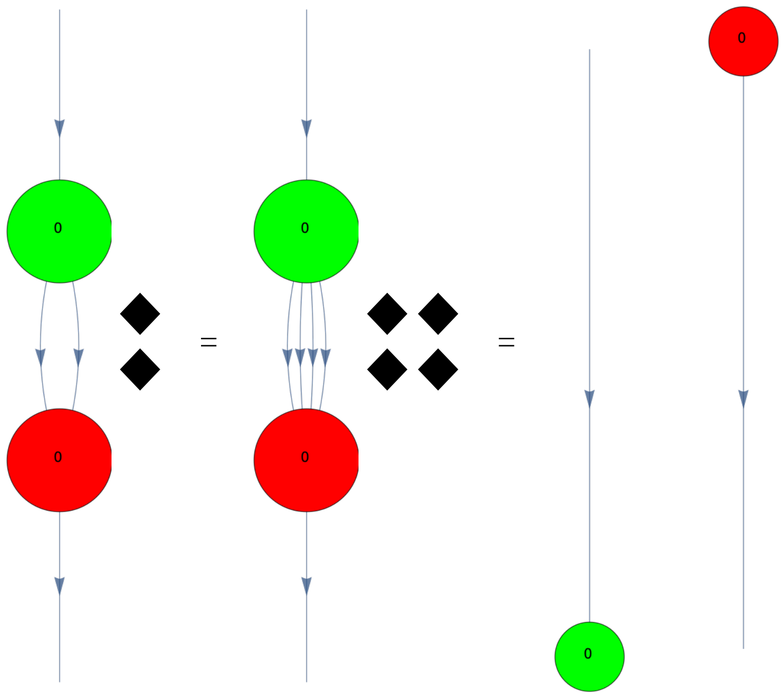


= $1 + e^{i\,\phi}$

Interaction between Z and X spiders (5)

In[1]:=
```
Grid@{{
   obj1 = ZXDiagram[{B[{1, 2}],
      i -> Z[1][0] -> X[1][0] -> o,
      Z[1] -> X[1]},
     GraphLayout -> "LayeredDigraphEmbedding"], Style[" = ", Large],
   obj2 = ZXDiagram[{B[Range@4],
      i -> Z[1][0] -> X[1][0] -> o,
      Z[1] -> X[1], Z[1] -> X[1], Z[1] -> X[1]},
     GraphLayout -> "LayeredDigraphEmbedding"], Style[" = ", Large],
   new = ZXDiagram[{i -> Z[1][0], X[1][0] -> o},
     GraphLayout -> "LayeredDigraphEmbedding"]
  }}
```

Out[1]=



In[2]:=
```
op1 = ToXBasis[ExpressionFor[obj1], {i}]
op2 = ToXBasis[ExpressionFor[obj2], {i}]
op = ToXBasis[ExpressionFor[new], {i}]
```

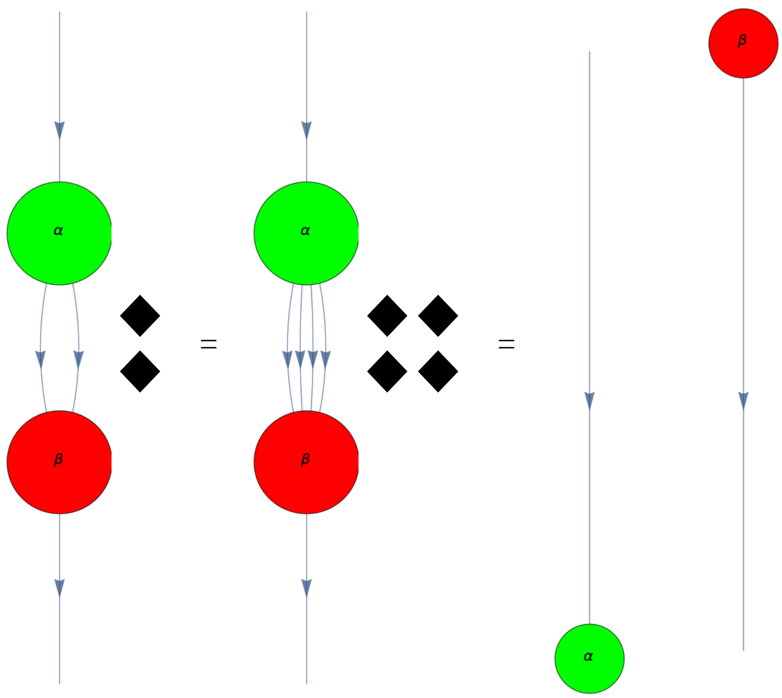
Out[2]= $2\ |0_o\rangle\langle +_i|$

Out[2]= $2\ |0_o\rangle\langle +_i|$

Out[2]= $2\ |0_o\rangle\langle +_i|$

In[3]:=
```
Grid@{{
   obj1 = ZXDiagram[{B[{1, 2}],
      i -> Z[1][α] -> X[1][β] -> o,
      Z[1] -> X[1]},
     GraphLayout -> "LayeredDigraphEmbedding"], Style[" = ", Large],
   obj2 = ZXDiagram[{B[Range@4],
      i -> Z[1][α] -> X[1][β] -> o,
      Z[1] -> X[1], Z[1] -> X[1], Z[1] -> X[1]},
     GraphLayout -> "LayeredDigraphEmbedding"], Style[" = ", Large],
   new = ZXDiagram[{i -> Z[1][α], X[1][β] -> o},
     GraphLayout -> "LayeredDigraphEmbedding"]
  }}
```

Out[3]=



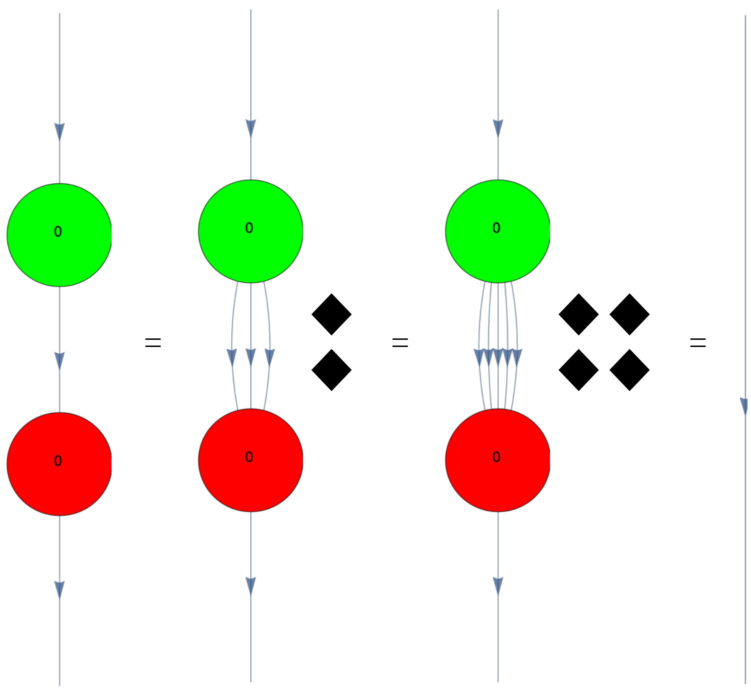
In[4]:=
```
ExpressionFor[obj1] - ExpressionFor[new]
ExpressionFor[obj2] - ExpressionFor[new]
```

Out[4]= 0

Out[4]= 0

In[1]:=
```
Grid@{{
   obj1 = ZXDiagram[{i -> Z[1][0] -> X[1][0] -> o},
     GraphLayout -> "LayeredDigraphEmbedding"], Style[" = ", Large],
   obj2 = ZXDiagram[{B[Range@2],
      i -> Z[1][0] -> X[1][0] -> o,
      Z[1] -> X[1], Z[1] -> X[1]},
     GraphLayout -> "LayeredDigraphEmbedding"], Style[" = ", Large],
   obj3 = ZXDiagram[{B[Range@4],
      i -> Z[1][0] -> X[1][0] -> o,
      Z[1] -> X[1], Z[1] -> X[1], Z[1] -> X[1], Z[1] -> X[1]},
     GraphLayout -> "LayeredDigraphEmbedding"], Style[" = ", Large],
   new = ZXDiagram[{i -> o},
     GraphLayout -> "LayeredDigraphEmbedding"]
  }}
```

Out[1]=



In[2]:=
```
op1 = ExpressionFor[obj1]
op2 = ExpressionFor[obj2]
op3 = ExpressionFor[obj3]
op = ExpressionFor[new]
```

Out[2]= $|0_o\rangle\langle 0_i| + |1_o\rangle\langle 1_i|$

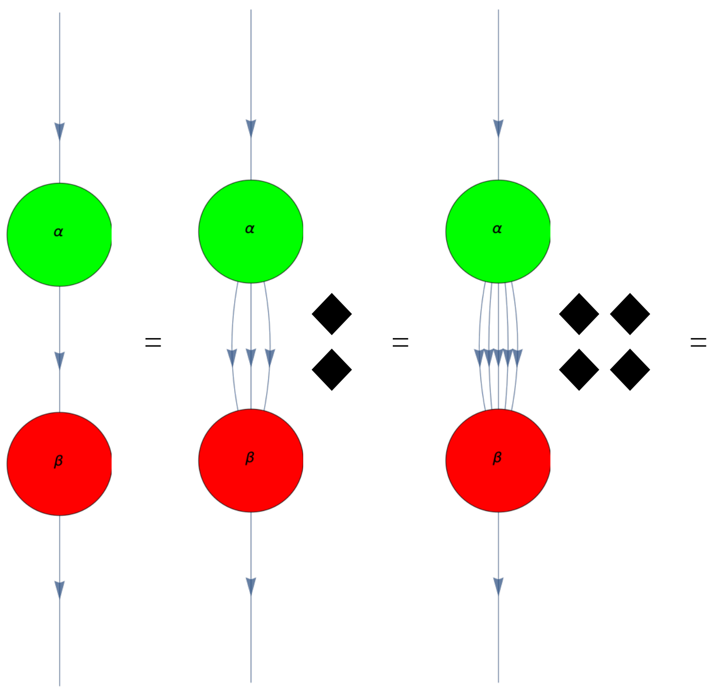
Out[2]= $|0_o\rangle\langle 0_i| + |1_o\rangle\langle 1_i|$

Out[2]= $|0_o\rangle\langle 0_i| + |1_o\rangle\langle 1_i|$

Out[2]= $|0_o\rangle\langle 0_i| + |1_o\rangle\langle 1_i|$

In[3]:=
```
Grid@{{
   obj1 = ZXDiagram[{i -> Z[1][α] -> X[1][β] -> o},
     GraphLayout -> "LayeredDigraphEmbedding"], Style[" = ", Large],
   obj2 = ZXDiagram[{B[Range@2],
      i -> Z[1][α] -> X[1][β] -> o,
      Z[1] -> X[1], Z[1] -> X[1]},
     GraphLayout -> "LayeredDigraphEmbedding"], Style[" = ", Large],
   obj3 = ZXDiagram[{B[Range@4],
      i -> Z[1][α] -> X[1][β] -> o,
      Z[1] -> X[1], Z[1] -> X[1], Z[1] -> X[1], Z[1] -> X[1]},
     GraphLayout -> "LayeredDigraphEmbedding"], Style[" = ", Large]
  }}
```

Out[3]=



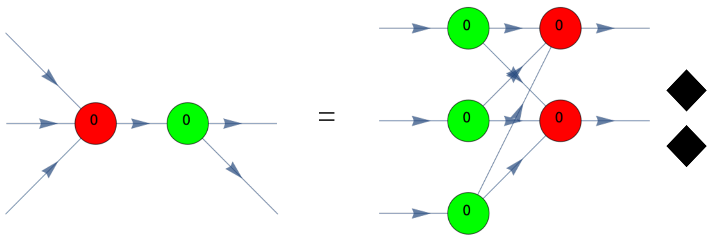
In[4]:=
```
ExpressionFor[obj1] - ExpressionFor[obj2]
ExpressionFor[obj1] - ExpressionFor[obj2]
```

Out[4]= 0

Out[4]= 0

In[1]:=
```
Grid@{{
   obj = ZXDiagram[
     {{i1, i2, o3} -> X[1][0] -> Z[1][0] -> {o1, o2}},
     ImageSize -> Small], Style[" = ", Large],
   new = ZXDiagram[
     {B@{1, 2},
      Z[{1, 2, 3}][0], X[{1, 2}][0],
      i1 -> Z[1], i2 -> Z[2], i3 -> Z[3],
      Z[{1, 2, 3}] -> X[{1, 2}],
      X[1] -> o1, X[2] -> o2},
     ImageSize -> Small]
  }}
```

Out[1]=



In[2]:=
```
op1 = ExpressionFor[obj]
op2 = ExpressionFor[new]
```

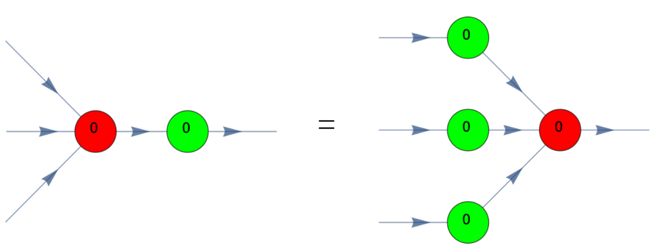
Out[2]= $\frac{1}{2}\left|0_{o1}0_{o2}\right\rangle\left\langle 0_{i1}0_{i2}0_{o3}\right| + \frac{1}{2}\left|0_{o1}0_{o2}\right\rangle\left\langle 0_{i1}1_{i2}1_{o3}\right| + \frac{1}{2}\left|0_{o1}0_{o2}\right\rangle\left\langle 1_{i1}0_{i2}1_{o3}\right| + \frac{1}{2}\left|0_{o1}0_{o2}\right\rangle\left\langle 1_{i1}1_{i2}0_{o3}\right| + \frac{1}{2}\left|1_{o1}1_{o2}\right\rangle\left\langle 0_{i1}0_{i2}1_{o3}\right| + \frac{1}{2}\left|1_{o1}1_{o2}\right\rangle\left\langle 0_{i1}1_{i2}0_{o3}\right| + \frac{1}{2}\left|1_{o1}1_{o2}\right\rangle\left\langle 1_{i1}0_{i2}0_{o3}\right| + \frac{1}{2}\left|1_{o1}1_{o2}\right\rangle\left\langle 1_{i1}1_{i2}1_{o3}\right|$

Out[2]= $\frac{1}{2}\left|0_{o1}0_{o2}\right\rangle\left\langle 0_{i1}0_{i2}0_{i3}\right| + \frac{1}{2}\left|0_{o1}0_{o2}\right\rangle\left\langle 0_{i1}1_{i2}1_{i3}\right| + \frac{1}{2}\left|0_{o1}0_{o2}\right\rangle\left\langle 1_{i1}0_{i2}1_{i3}\right| + \frac{1}{2}\left|0_{o1}0_{o2}\right\rangle\left\langle 1_{i1}1_{i2}0_{i3}\right| + \frac{1}{2}\left|1_{o1}1_{o2}\right\rangle\left\langle 0_{i1}0_{i2}1_{i3}\right| + \frac{1}{2}\left|1_{o1}1_{o2}\right\rangle\left\langle 0_{i1}1_{i2}0_{i3}\right| + \frac{1}{2}\left|1_{o1}1_{o2}\right\rangle\left\langle 1_{i1}0_{i2}0_{i3}\right| + \frac{1}{2}\left|1_{o1}1_{o2}\right\rangle\left\langle 1_{i1}1_{i2}1_{i3}\right|$

This is trivial .

In[1]:=
```
Grid@{{
   obj = ZXDiagram[
     {{i1, i2, o3} -> X[1][0] -> Z[1][0] -> o1},
     ImageSize -> Small], Style[" = ", Large],
   new = ZXDiagram[
     {Z[{1, 2, 3}][0],
      i1 -> Z[1], i2 -> Z[2], i3 -> Z[3],
      Z[{1, 2, 3}] -> X[1][0] -> o1},
     ImageSize -> Small]
  }}
```

Out[1]=



In[2]:=
```
op1 = ExpressionFor[obj]
op2 = ExpressionFor[new]
```

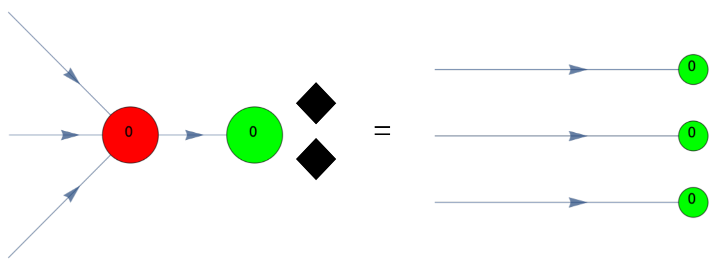
Out[2]= $\frac{1}{2}|0_{o1}\rangle\langle 0_{i1}0_{i2}0_{o3}| + \frac{1}{2}|0_{o1}\rangle\langle 0_{i1}1_{i2}1_{o3}| + \frac{1}{2}|0_{o1}\rangle\langle 1_{i1}0_{i2}1_{o3}| + \frac{1}{2}|0_{o1}\rangle\langle 1_{i1}1_{i2}0_{o3}| + \frac{1}{2}|1_{o1}\rangle\langle 0_{i1}0_{i2}1_{o3}| + \frac{1}{2}|1_{o1}\rangle\langle 0_{i1}1_{i2}0_{o3}| + \frac{1}{2}|1_{o1}\rangle\langle 1_{i1}0_{i2}0_{o3}| + \frac{1}{2}|1_{o1}\rangle\langle 1_{i1}1_{i2}1_{o3}|$

Out[2]= $\frac{1}{2}|0_{o1}\rangle\langle 0_{i1}0_{i2}0_{i3}| + \frac{1}{2}|0_{o1}\rangle\langle 0_{i1}1_{i2}1_{i3}| + \frac{1}{2}|0_{o1}\rangle\langle 1_{i1}0_{i2}1_{i3}| + \frac{1}{2}|0_{o1}\rangle\langle 1_{i1}1_{i2}0_{i3}| + \frac{1}{2}|1_{o1}\rangle\langle 0_{i1}0_{i2}1_{i3}| + \frac{1}{2}|1_{o1}\rangle\langle 0_{i1}1_{i2}0_{i3}| + \frac{1}{2}|1_{o1}\rangle\langle 1_{i1}0_{i2}0_{i3}| + \frac{1}{2}|1_{o1}\rangle\langle 1_{i1}1_{i2}1_{i3}|$

This is a fascinating illustration of how the ZX calculus handles complementarity naturally.

In[1]:=
```
Grid@{{
   obj = ZXDiagram[
     {{i1, i2, o3} -> X[1][0] -> Z[1][0], B[{1, 2}]},
     ImageSize -> Small], Style[" = ", Large],
   new = ZXDiagram[
     {Z[{1, 2, 3}][0],
      i1 -> Z[1], i2 -> Z[2], i3 -> Z[3]},
     ImageSize -> Small]
  }}
```

Out[1]=



In[2]:=
```
op1 = ExpressionFor[obj] // Garner
op2 = ExpressionFor[new]
```

Out[2]= $\langle 0_{i1}0_{i2}0_{o3}| + \langle 0_{i1}0_{i2}1_{o3}| + \langle 0_{i1}1_{i2}0_{o3}| + \langle 0_{i1}1_{i2}1_{o3}| + \langle 1_{i1}0_{i2}0_{o3}| + \langle 1_{i1}0_{i2}1_{o3}| + \langle 1_{i1}1_{i2}0_{o3}| + \langle 1_{i1}1_{i2}1_{o3}|$

Out[2]= $\langle 0_{i1}0_{i2}0_{i3}| + \langle 0_{i1}0_{i2}1_{i3}| + \langle 0_{i1}1_{i2}0_{i3}| + \langle 0_{i1}1_{i2}1_{i3}| + \langle 1_{i1}0_{i2}0_{i3}| + \langle 1_{i1}0_{i2}1_{i3}| + \langle 1_{i1}1_{i2}0_{i3}| + \langle 1_{i1}1_{i2}1_{i3}|$

See Also

ZXObject ▪ Chain ▪ ZXLayers ▪ ZXForm ▪ ZXStandardForm

Related Guides

- MaZX

Related Links

- R. Duncan, A. Kissinger, S. Perdrix, and J. van de Wetering, Quantum 4, 279 (2020) , "Graph–theoretic Simplification of Quantum Circuits with the ZX–calculus."