

# DrawPauliTransferMap

```
SetDirectory @ NotebookDirectory[];  
Import["../Link/QuESTlink.m"];
```

## Doc

### ? DrawPauliTransferMap

#### Symbol

DrawPauliTransferMap[map] visualises the given PTMap as a graph where nodes are basis Pauli strings, and edges indicate the transformative action of the map.

DrawPauliTransferMap also accepts PTM, circuit and gate

instances, for which the corresponding PTMap is automatically calculated.

DrawPauliTransferMap accepts options "PauliStringForm", "ShowCoefficients" and "EdgeDegreeStyles", in addition to all options accepted by Graph[].

- "ShowCoefficients" -> False hides the map's Pauli string coefficients which are otherwise shown as edge labels.
- "PauliStringForm" sets the vertex label format to one of "Subscript" (default), "Index", "Kronecker", "String" or "Hidden". These (except the latter) are the formats are supported by GetPauliStringReformatted[].
- "EdgeDegreeStyles" specifies a list of styles (default informed by ColorData["Pastel"]) to set upon edges from nodes with increasing outdegree. For example, "EdgeDegreeStyles" -> {Red, Green, Blue} sets edges from Pauli states which are mapped to a single other state to the colour Red, but two-outdegree node out-edges become Green, and three-outdegree become Blue. The list is assumed repeated for higher outdegree nodes than specified.
- Graph[] options override these settings, so specifying EdgeStyle -> Black will set all edges to Black regardless of their node's outdegree.

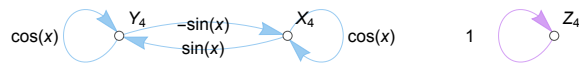
# Correctness

## PTMap

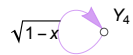
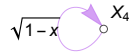
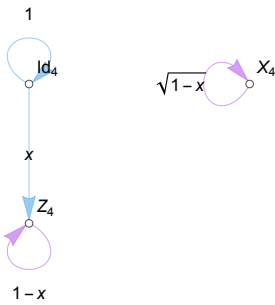
`map = CalcPauliTransferMap @ Rz4[x]`

`DrawPauliTransferMap[map]`

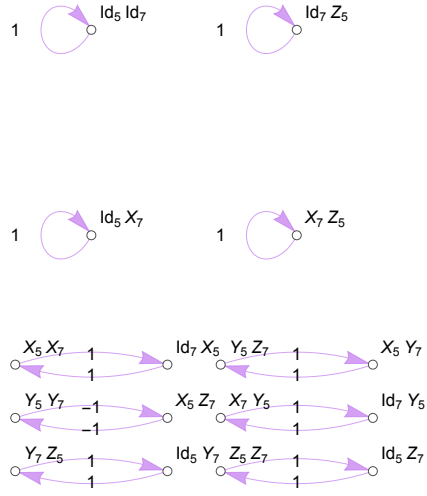
`PTMap4[0 → {{0, 1}}, 1 → {{1, Cos[x]}, {2, Sin[x]}},  
2 → {{1, -Sin[x]}, {2, Cos[x]}}, 3 → {{3, 1}}]`



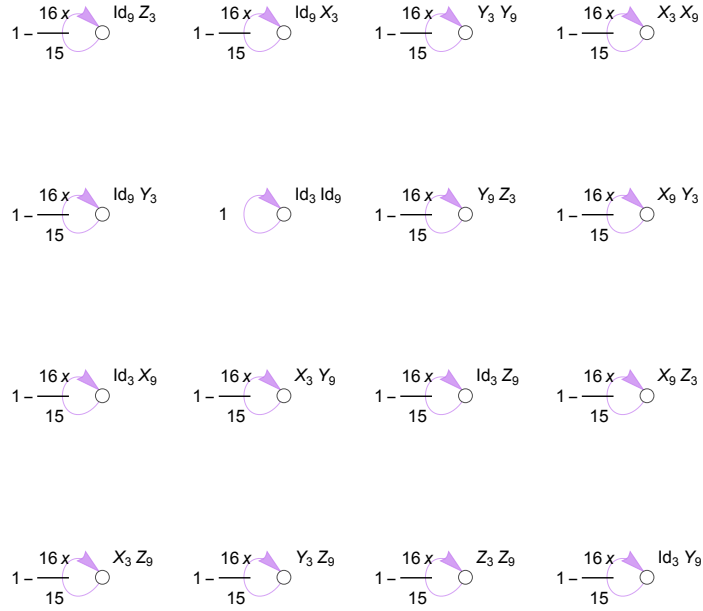
`DrawPauliTransferMap @ CalcPauliTransferMap @ Damp4[x]`



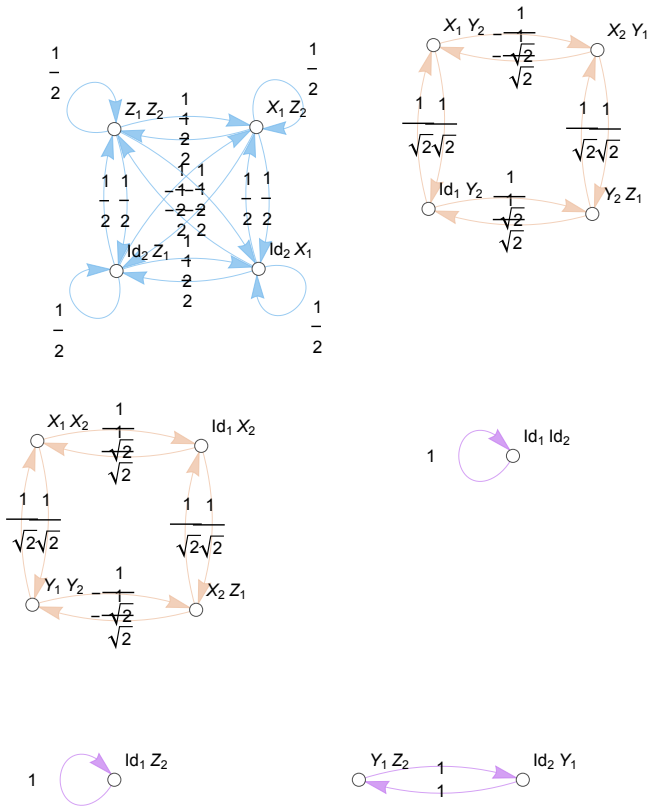
### DrawPauliTransferMap @ CalcPauliTransferMap @ C<sub>5</sub>[X<sub>7</sub>]



### DrawPauliTransferMap @ CalcPauliTransferMap @ Depol<sub>3,9</sub>[x]

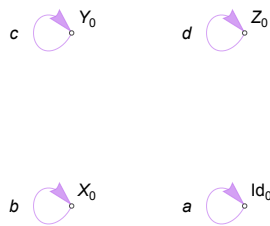


DrawPauliTransferMap @ CalcPauliTransferMap @ C<sub>2</sub>[H<sub>1</sub>]

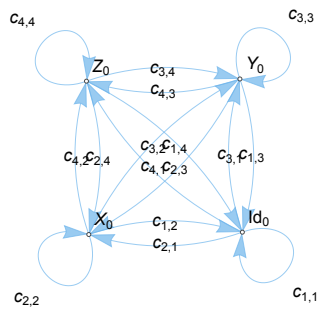


## PTM, gates, circuits

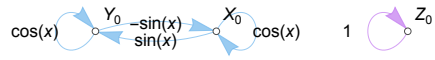
DrawPauliTransferMap @ PTM<sub>0</sub> @ DiagonalMatrix[{a, b, c, d}]



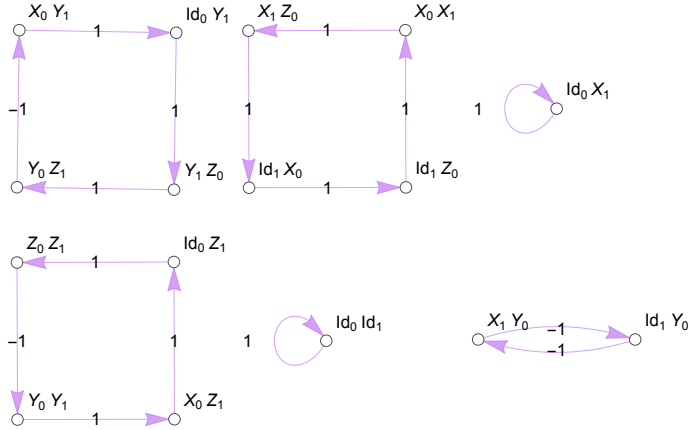
DrawPauliTransferMap @ PTM<sub>0</sub> @ Table[c<sub>i,j</sub>, {i, 4}, {j, 4}]



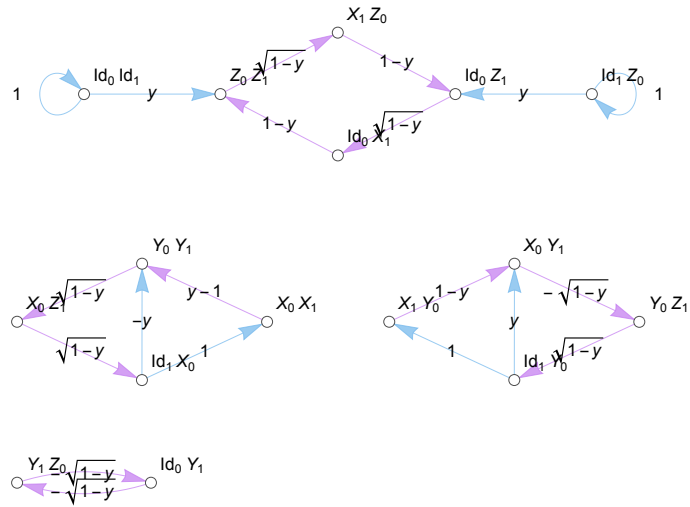
### DrawPauliTransferMap @ Rz<sub>0</sub>[x]



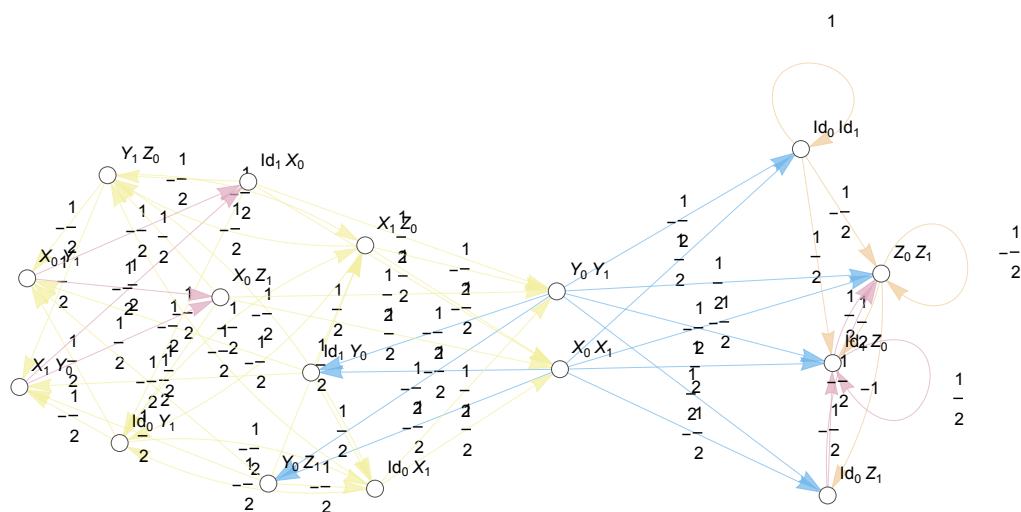
### DrawPauliTransferMap @ Circuit[H<sub>0</sub> C<sub>0</sub>[X<sub>1</sub>]]



### DrawPauliTransferMap @ Circuit[H<sub>1</sub> Damp<sub>1</sub>[y] C<sub>0</sub>[X<sub>1</sub>]]



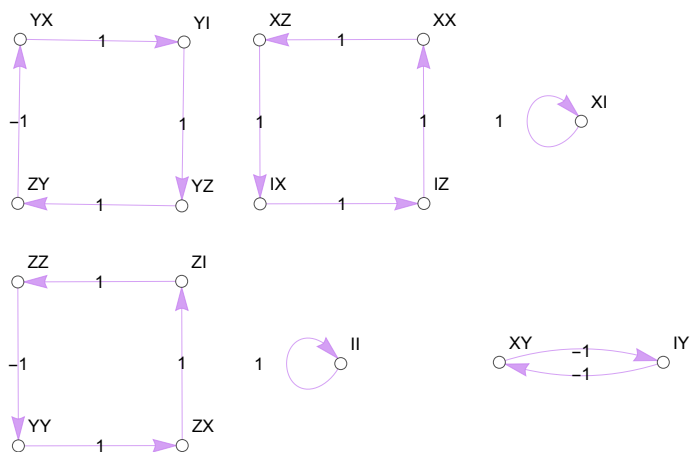
$$\text{DrawPauliTransferMap}\left[U_{0,1} @ \begin{pmatrix} 0 & 0 & i & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}\right]$$



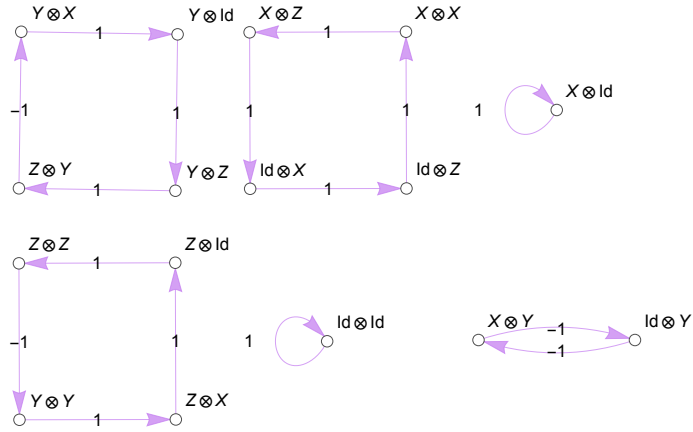
## option “PauliStringForm”

```
circ = Circuit[H0 C0[X1]];
```

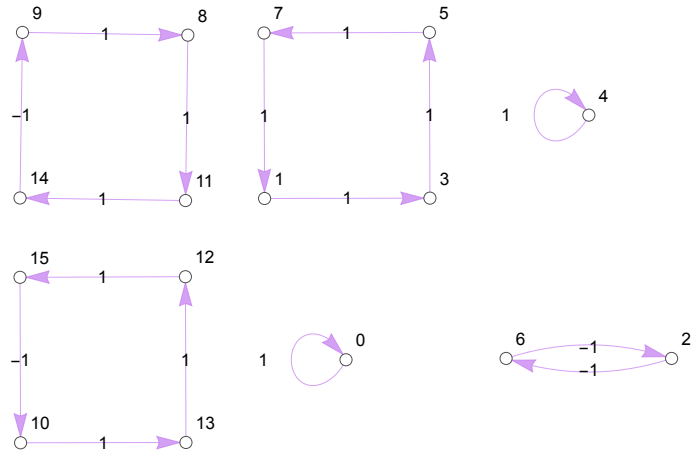
```
DrawPauliTransferMap[circ, "PauliStringForm" → "String"]
```



`DrawPauliTransferMap[circ, "PauliStringForm" → "Kronecker"]`

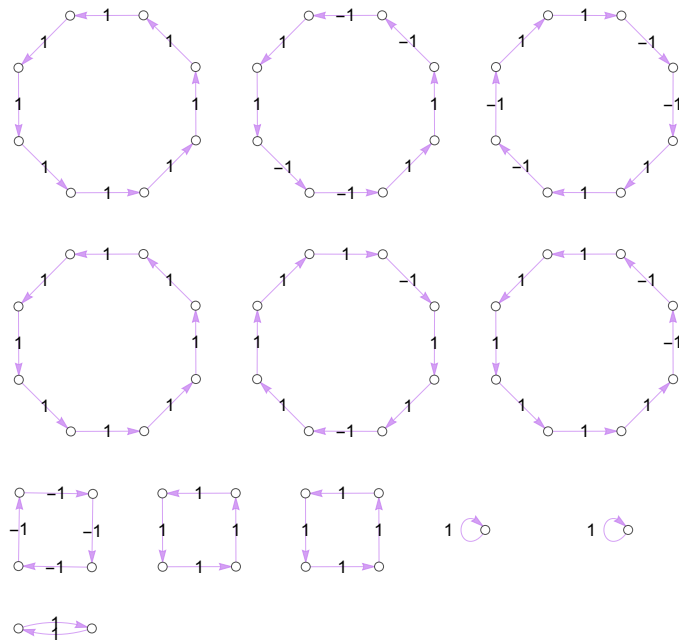


`DrawPauliTransferMap[circ, "PauliStringForm" → "Index"]`



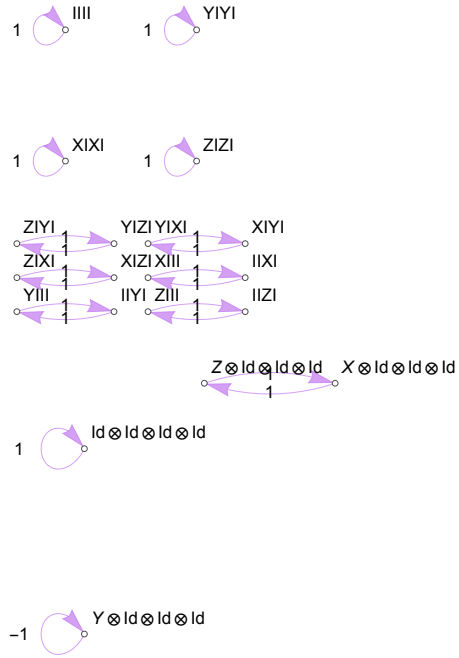
`circ = Circuit[H0 C0[X1] SWAP1,2];`

`DrawPauliTransferMap[circ, "PauliStringForm" → "Hidden"]`



```
DrawPauliTransferMap[SWAP1,3, "PauliStringForm" → "String"]
```

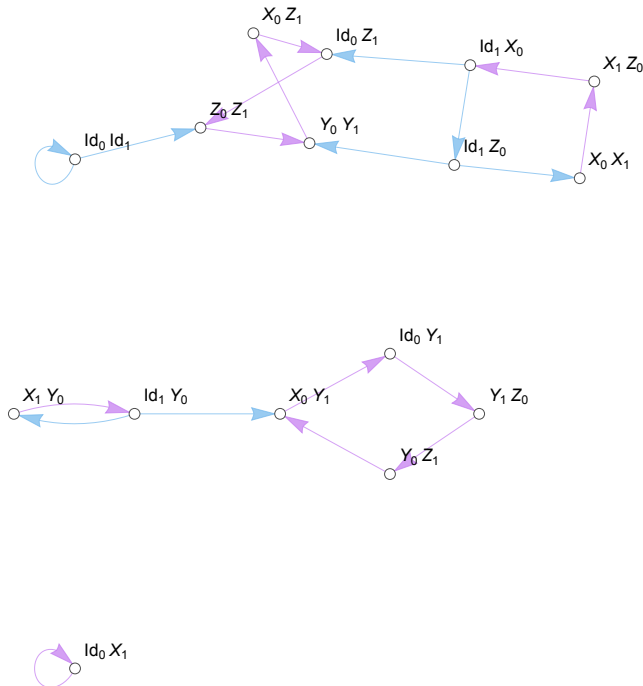
```
DrawPauliTransferMap[H3, "PauliStringForm" → "Kronecker"]
```



## option “ShowCoefficients”

```
circ = Circuit[H0 Depol0,1[x] Damp1[y] C0[X1]];
```

```
DrawPauliTransferMap[circ, "ShowCoefficients" → False]
```

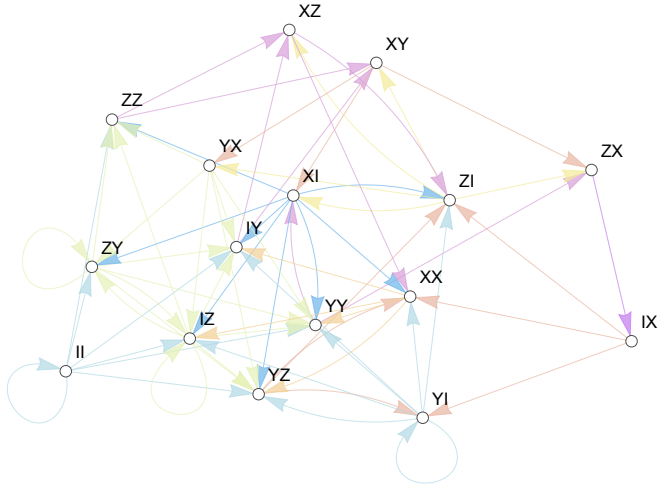




```

circ = Circuit[H1 Damp0[x] Depol0,1[x] C0[X1] C1[Rx0[x]] R[x, X0 Y1]];
DrawPauliTransferMap[circ,
  "ShowCoefficients" → False,
  "PauliStringForm" → "String"]

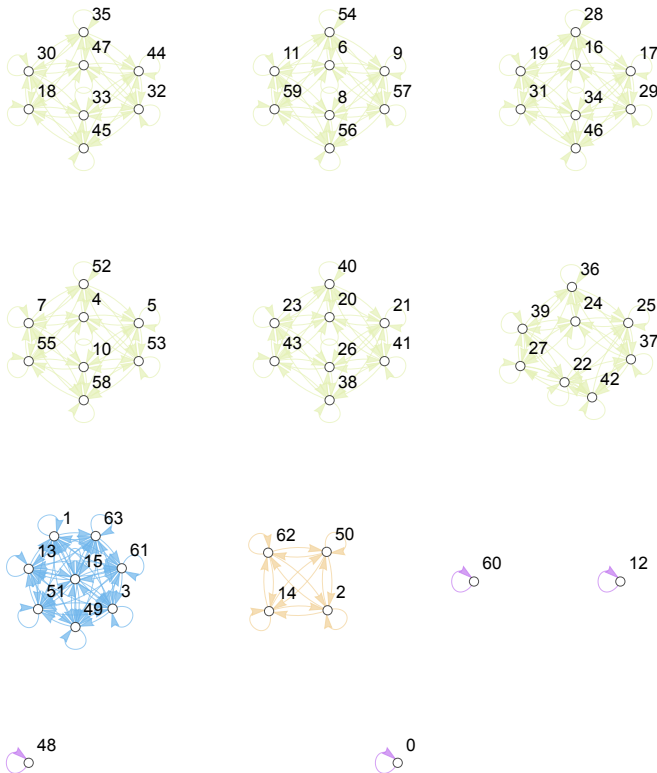
```



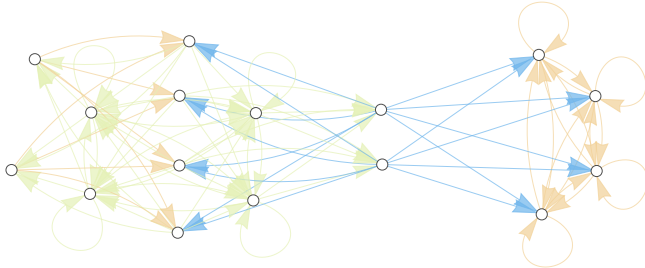
```

DrawPauliTransferMap[C0,2[H1],
  "ShowCoefficients" → False,
  "PauliStringForm" → "Index"]

```

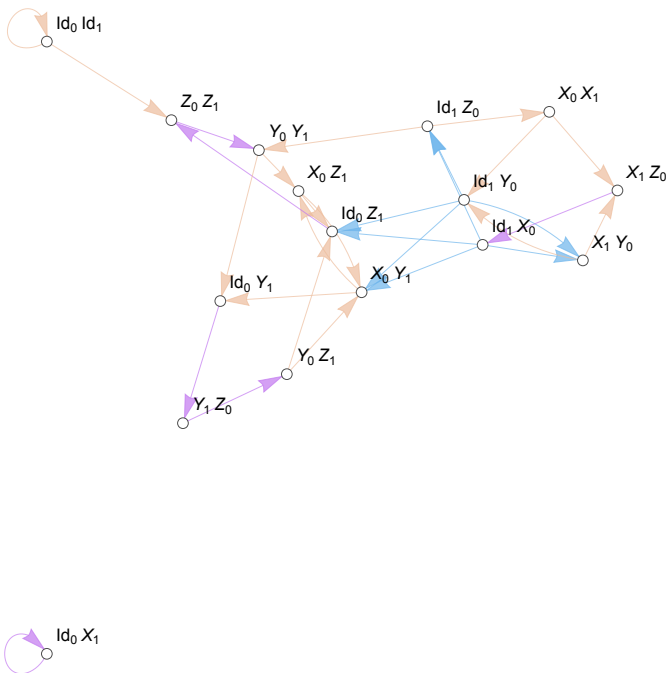


$\text{DrawPauliTransferMap}\left[U_{0,1} @ \begin{pmatrix} 0 & 0 & a & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix},\right.$   
 $\left.\text{"ShowCoefficients"} \rightarrow \text{False}, \text{"PauliStringForm"} \rightarrow \text{"Hidden"}\right]$



## option “EdgeDegreeStyles”

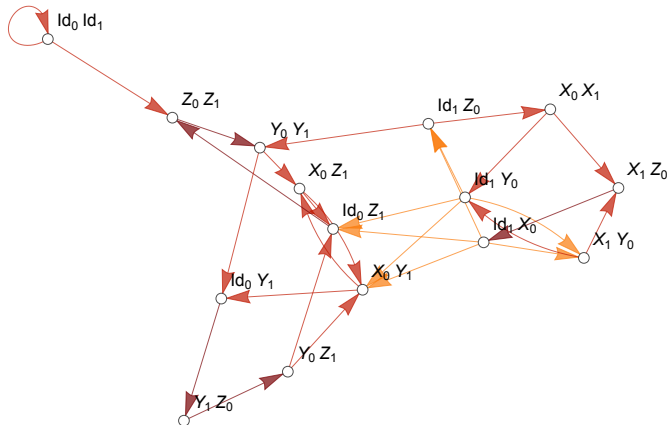
`circ = Circuit[H0 Depol0,1[x] Rx0[a] Damp1[y] C0[X1]];`  
`DrawPauliTransferMap[circ, "ShowCoefficients" → False]`



```
colors = ColorData["SolarColors"] /@ Range[0, 1, .2]
```

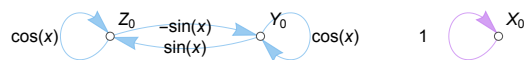
```
DrawPauliTransferMap[circ,  
  "ShowCoefficients" → False,  
  "EdgeDegreeStyles" → colors]
```

```
{  
  , , , , ,    
}
```



## option AssertValidChannels

```
DrawPauliTransferMap[Rx0[x], AssertValidChannels → True]
```



`DrawPauliTransferMap[Rx0[x], AssertValidChannels → False]`

$$\frac{1}{2} \left( 2 \cos^2\left(\frac{x}{2}\right) - 2 \sin^2\left(\frac{x}{2}\right) + 2 \cos^2\left(\frac{x}{2}\right) + 2 \sin^2\left(\frac{x}{2}\right) \right)$$

$$\frac{1}{2} \left( 2 \sin^2\left(\frac{x}{2}\right) + 2 \cos^2\left(\frac{x}{2}\right) \right)$$

## options of Graph

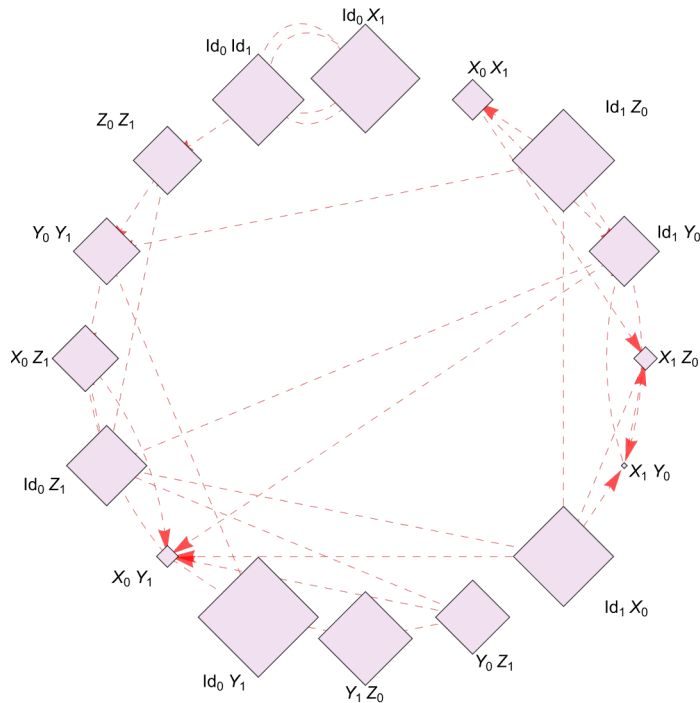
```

circ = Circuit[H0 Depol0,1[x] Rx0[a] Damp1[y] C0[X1]];
DrawPauliTransferMap[circ, "ShowCoefficients" → False,

EdgeStyle → Directive[Red, Dashed],
VertexShapeFunction → "Diamond",
VertexSize → RandomReal[],
VertexStyle → LightPurple,
GraphLayout → "CircularEmbedding"

]

```



# Warnings

**CalcPauliTransferMap[Depol<sub>0</sub>[3 / 4]]**

**PTMap<sub>0</sub>[0 → {{0, 1}}, 1 → {}, 2 → {}, 3 → {}]**

**DrawPauliTransferMap[%]**

**DrawPauliTransferMap** : Warning: The Pauli transfer map produces no Pauli string from some initial strings. These edges (and null strings) are not being plotted. Hide this warning with Quiet[].



# Errors

**DrawPauliTransferMap[CalcPauliTransferMap[X<sub>0</sub>], "BadOption" → "bad"]**

**OptionValue** : Unknown option BadOption for {DrawPauliTransferMap, Graph}.

\$Failed

**DrawPauliTransferMap[CalcPauliTransferMap[X<sub>0</sub>], "PauliStringForm" → "bad"]**

**DrawPauliTransferMap** : Unrecognised value for option "PauliStringForm". See ?DrawPauliTransferMap

\$Failed

**DrawPauliTransferMap[ PTMap<sub>0,-1</sub>[x] ]**

**DrawPauliTransferMap** : Failed to automatically obtain the PTMap due to the below error:

**CalcPauliTransferMatrix** : Circuit contained an unrecognised or unsupported gate: PTMap<sub>0,1</sub>[x]

\$Failed

**DrawPauliTransferMap[ ]**

**DrawPauliTransferMap** : Invalid arguments. See ?DrawPauliTransferMap

\$Failed

**DrawPauliTransferMap @ Bad<sub>0</sub>**

**DrawPauliTransferMap** : Failed to automatically obtain the PTMap due to the below error:

**CalcPauliTransferMatrix** : Circuit contained an unrecognised or unsupported gate: Bad<sub>0</sub>

\$Failed