Many-qubit gates

Bracket Notation

the "statevector
↓
2^n-dim complex vector
$|\psi\rangle \in \mathbb{C}^{2^n}$

States of n qubits:

$$|\psi\rangle = \begin{pmatrix} \psi_{0..00} \\ \psi_{0..01} \\ \vdots \\ \psi_{1..11} \end{pmatrix}$$

coefficients indexed by n-bit strings $\underline{b} \in \{0,1\}^n$

(i) normalised $\quad \sum_{\underline{b} \in \{0,1\}} |\psi_{\underline{b}}|^2 = 1$

(ii) up to phase $\quad |\psi\rangle = e^{i\theta} |\psi\rangle$

Bracket Notation

Product states on 2 qubits:

joint state of both qubits

$$\overbrace{\underbrace{|\psi\rangle}\otimes\underbrace{|\varphi\rangle}} := \begin{pmatrix} \psi_0 \begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} \\ \psi_1 \begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} \end{pmatrix}$$

each qubit has its own well-defined state

Bracket Notation

Product states on 2 qubits:

joint state of both qubits

$$\overbrace{\underbrace{|\psi\rangle} \otimes \underbrace{|\varphi\rangle}} := \begin{pmatrix} \psi_0 \varphi_0 \\ \psi_0 \varphi_1 \\ \psi_1 \varphi_0 \\ \psi_1 \varphi_1 \end{pmatrix}$$

each qubit has its own well-defined state

Bracket Notation

Product states on 3 qubits:

$$\overbrace{\underbrace{|\psi\rangle} \otimes \underbrace{|\varphi\rangle} \otimes \underbrace{|\gamma\rangle}}^{\text{joint state of all three qubits}} := \begin{pmatrix} \psi_0 \begin{pmatrix} \varphi_0 \begin{pmatrix} \gamma_0 \\ \gamma_1 \end{pmatrix} \\ \varphi_1 \begin{pmatrix} \gamma_0 \\ \gamma_1 \end{pmatrix} \end{pmatrix} \\ \psi_1 \begin{pmatrix} \varphi_0 \begin{pmatrix} \gamma_0 \\ \gamma_1 \end{pmatrix} \\ \varphi_1 \begin{pmatrix} \gamma_0 \\ \gamma_1 \end{pmatrix} \end{pmatrix} \end{pmatrix}$$

each qubit has its own well-defined state

Bracket Notation

Product states on 3 qubits:

$$\overbrace{\underbrace{|\psi\rangle} \otimes \underbrace{|\varphi\rangle} \otimes \underbrace{|\gamma\rangle}}^{\text{joint state of all three qubits}} := \begin{pmatrix} \psi_0 \begin{pmatrix} \varphi_0 \gamma_0 \\ \varphi_0 \gamma_1 \\ \varphi_1 \gamma_0 \\ \varphi_1 \gamma_1 \end{pmatrix} \\ \psi_1 \begin{pmatrix} \varphi_0 \gamma_0 \\ \varphi_0 \gamma_1 \\ \varphi_1 \gamma_0 \\ \varphi_1 \gamma_1 \end{pmatrix} \end{pmatrix}$$

each qubit has its own well-defined state

Bracket Notation

Product states on 3 qubits:

$$\overbrace{\underbrace{|\psi\rangle} \otimes \underbrace{|\varphi\rangle} \otimes \underbrace{|\gamma\rangle}}^{\text{joint state of all three qubits}} := \begin{pmatrix} \psi_0 \varphi_0 \gamma_0 \\ \psi_0 \varphi_0 \gamma_1 \\ \psi_0 \varphi_1 \gamma_0 \\ \psi_0 \varphi_1 \gamma_1 \\ \psi_1 \varphi_0 \gamma_0 \\ \psi_1 \varphi_0 \gamma_1 \\ \psi_1 \varphi_1 \gamma_0 \\ \psi_1 \varphi_1 \gamma_1 \end{pmatrix}$$

each qubit has its own well-defined state

Bracket Notation

$$\langle \underline{b}' | \underline{b} \rangle = \begin{cases} 1 & \text{if } \underline{b} = \underline{b}' \\ 0 & \text{if } \underline{b} \neq \underline{b}' \end{cases}$$

(orthonormal basis)

Computational basis states on n qubits:

$$|\underline{b}\rangle := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

for $\underline{b} \in \{0,1\}^n$

coefficient indexed by $\underline{b}$ is 1, all other are 0.

Bracket Notation

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$|00\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|000\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|001\rangle$$

$$|01\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|010\rangle$$

$$|011\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$|10\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$|100\rangle$$

$$|101)$$

$$|11\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$|110\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$|111\rangle$$

Bracket Notation

$$|\psi\rangle = \begin{pmatrix} \psi_{0\ldots0} \\ \psi_{0\ldots1} \\ \vdots \\ \psi_{1\ldots1} \end{pmatrix} = \psi_{0\ldots0}\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \psi_{0\ldots1}\begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} + \ldots + \psi_{1\ldots1}\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}$$

$$= \psi_{0\ldots0}|0\ldots0\rangle + \psi_{0\ldots1}|0\ldots1\rangle + \ldots + \psi_{1\ldots1}|1\ldots1\rangle$$

$$= \sum_{\underline{b}\in\{0,1\}^n} \psi_{\underline{b}}\,|\underline{b}\rangle$$

Bracket Notation

$$|\psi\rangle \otimes |\varphi\rangle = \left(\psi_0|0\rangle + \psi_1|1\rangle\right) \otimes \left(\varphi_0|0\rangle + \varphi_1|1\rangle\right)$$

$$= \psi_0\varphi_0|00\rangle + \psi_0\varphi_1|01\rangle + \psi_1\varphi_0|10\rangle + \psi_1\varphi_1|11\rangle$$

Bracket Notation

2-qubit Bell state:

$$|\Phi^+\rangle := \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

Bracket Notation

2-qubit Bell state:

$$|\Phi^+\rangle := \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

This is an <u>entangled</u> state: it cannot be written as $|\Phi^+\rangle = |\psi\rangle \otimes |\varphi\rangle$ for any $|\psi\rangle, |\varphi\rangle$

Bracket Notation

2-qubit Bell state:

$$|\Phi^+\rangle := \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \frac{1}{\sqrt{2}}\begin{pmatrix}1\\0\\0\\1\end{pmatrix}$$

This is an <u>entangled</u> state: it cannot be written as $|\Phi^+\rangle = |\psi\rangle \otimes |\varphi\rangle$ for any $|\psi\rangle, |\varphi\rangle$

<u>Proof</u>: $|\Phi^+\rangle = |\psi\rangle \otimes |\varphi\rangle$ would imply $\psi_0\varphi_1 = 0 = \psi_1\varphi_0$, which would imply that either $\psi_0\varphi_0 = 0$ or $\psi_1\varphi_1 = 0$ (both are $\frac{1}{\sqrt{2}}$) $\square$

Graphical Notation

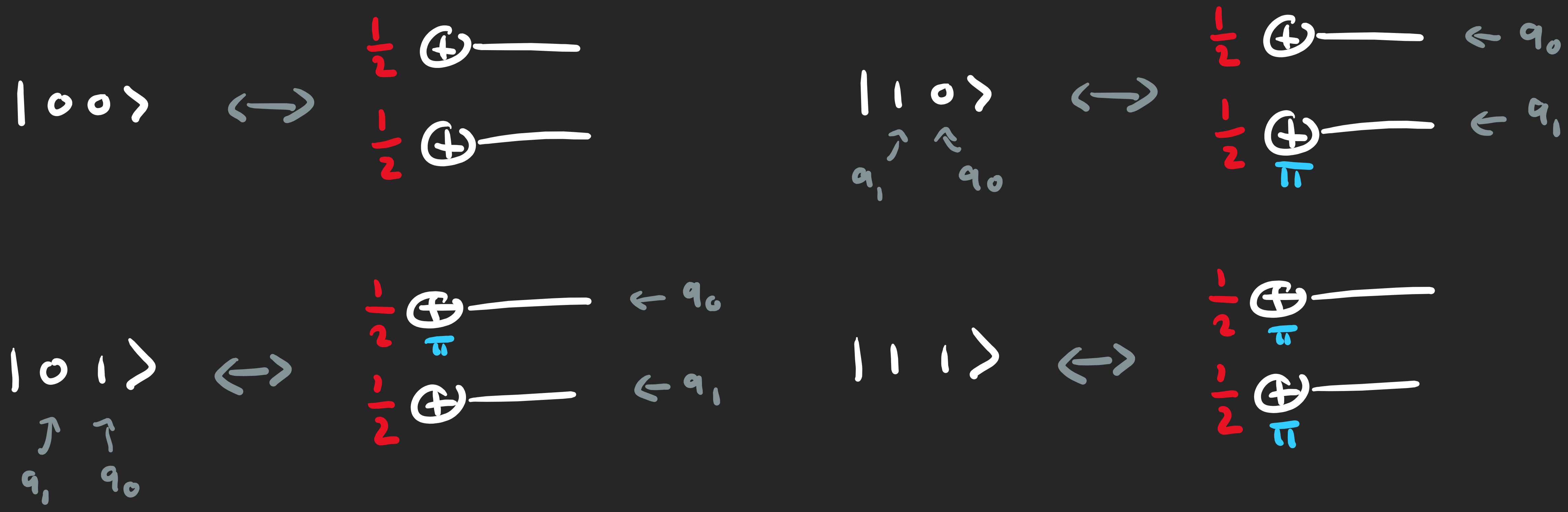
Product state
$\Rightarrow$ each qubit has its own state

each qubit has its own wire

$$\underbrace{|\psi\rangle}_{q_1} \otimes \underbrace{|\varphi\rangle}_{q_0} \longleftrightarrow$$

Note: we index qubits right-to-left because this is the convention used by Qiskit for bitstrings.

Graphical Notation



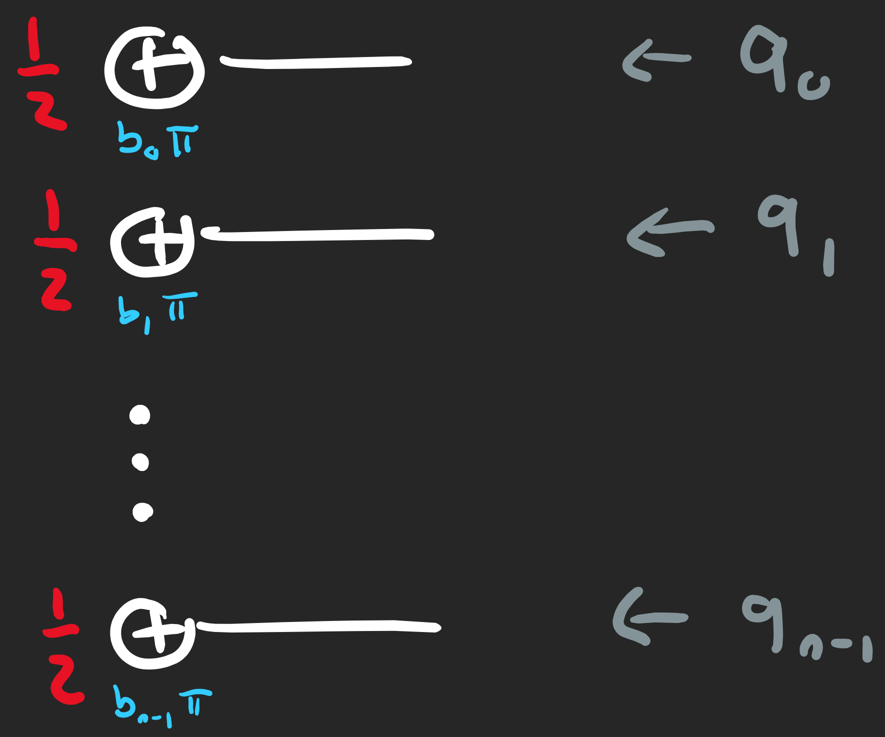
Note: we index qubits right-to-left because this is the convention used by Qiskit for bitstrings.

Graphical Notation

$$|b_{n-1}\rangle \otimes \cdots \otimes |b_0\rangle$$

$$\downarrow$$

$$|\underline{b}\rangle$$



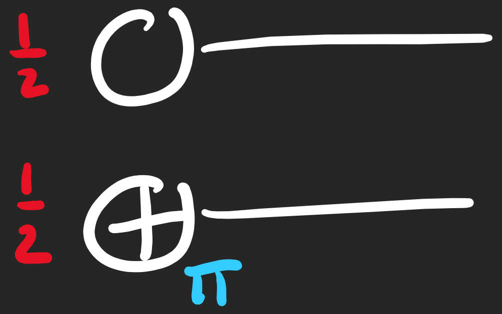
$$\underline{b} = b_{n-1}\cdots b_1 b_0 \in \{0,1\}^n$$

Note: we index qubits right-to-left because this is the convention used by Qiskit for bitstrings.

Graphical Notation

$|1\rangle \otimes |+\rangle$ $\longleftrightarrow$

q_1 $\quad q_0$



q_0 in $|+\rangle$

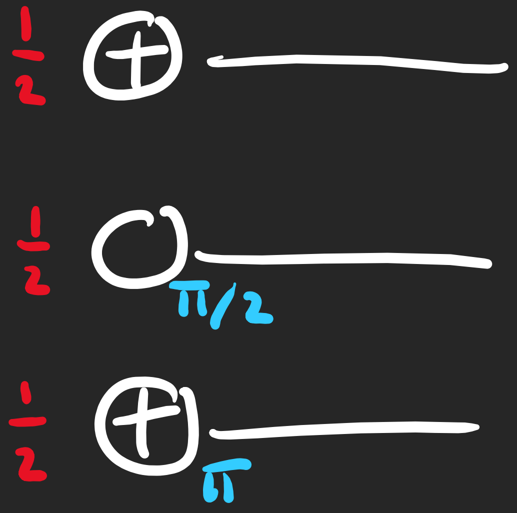
q_1 in $|1\rangle$

Shorthand: $|1+\rangle$

Shorthand: $|1R0\rangle$

$|1\rangle \otimes |R\rangle \otimes |0\rangle$ $\longleftrightarrow$

q_2 $\quad q_1$ $\quad q_0$



q_0 in $|0\rangle$

q_1 in $|R\rangle$

q_2 in $|1\rangle$

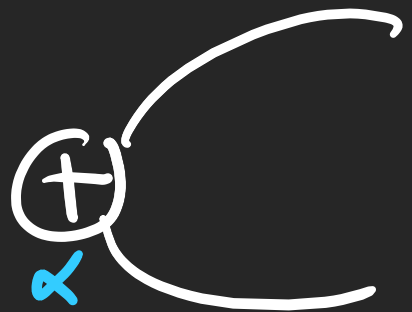
Bell State

Z basis:

$$|00\rangle + e^{i\alpha}|11\rangle \quad \longleftrightarrow$$

X basis:

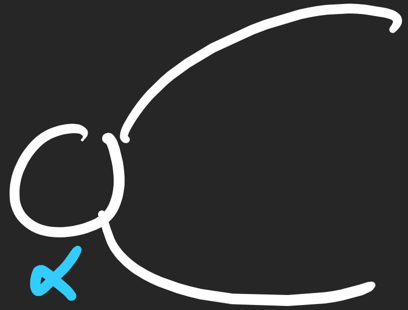
$$|{++}\rangle + e^{i\alpha}|{--}\rangle \quad \longleftrightarrow$$



Bell State

Z basis:

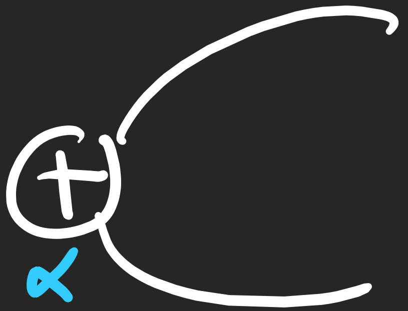
$$\frac{1}{\sqrt{2}}\left(|00\rangle + e^{i\alpha}|11\rangle\right) \longleftrightarrow \frac{1}{2}$$



(normalised)

X basis:

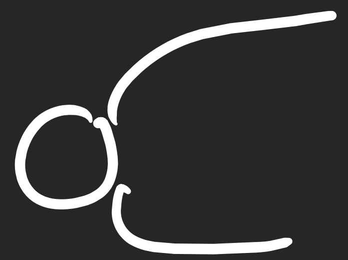
$$\frac{1}{\sqrt{2}}\left(|{+}{+}\rangle + e^{i\alpha}|{-}{-}\rangle\right) \longleftrightarrow \frac{1}{2}$$



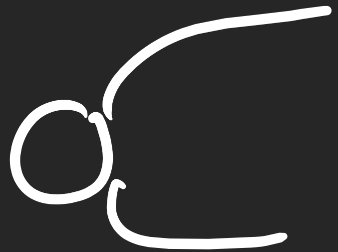
(normalised)

Bell State

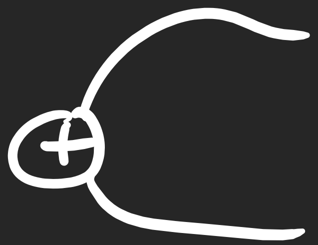
$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \quad \longleftrightarrow \quad \frac{1}{2}$$



Bell State

$|\Phi^+\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \longleftrightarrow \frac{1}{2}$ 

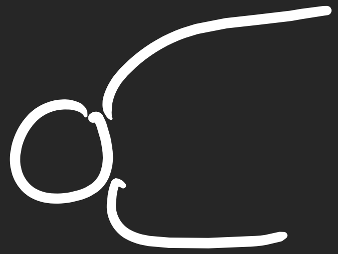
But also:

 =  (note: $\alpha = 0$)

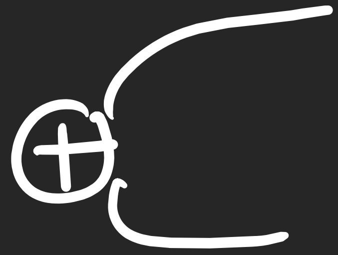
Proof: $|{+}{+}\rangle + |{-}{-}\rangle = \frac{1}{2}(|0\rangle+|1\rangle)(|0\rangle+|1\rangle) + \frac{1}{2}(|0\rangle-|1\rangle)(|0\rangle-|1\rangle)$

$= \frac{1}{2}\left(|00\rangle+|01\rangle+|10\rangle+|11\rangle\right) + \frac{1}{2}\left(|00\rangle-|01\rangle-|10\rangle+|11\rangle\right) = |00\rangle+|11\rangle$

Bell State

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \quad \longleftrightarrow \quad \frac{1}{2}$$ 

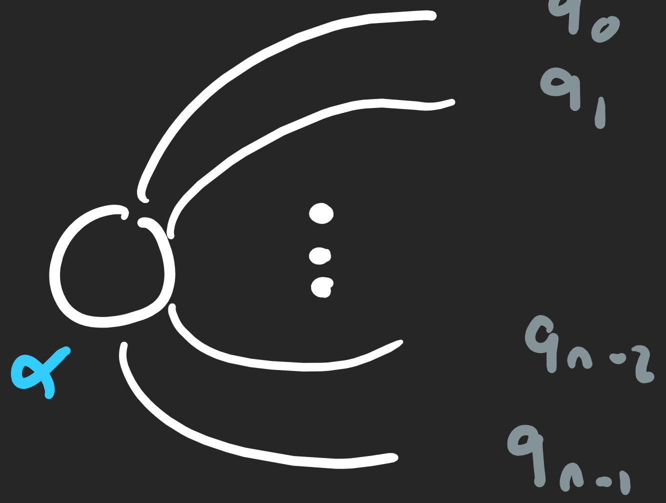
$$\parallel \qquad\qquad\qquad\qquad \parallel$$

$$\frac{1}{\sqrt{2}}\left(|{+}{+}\rangle + |{-}{-}\rangle\right) \quad \longleftrightarrow \quad \frac{1}{2}$$ 

GHZ States

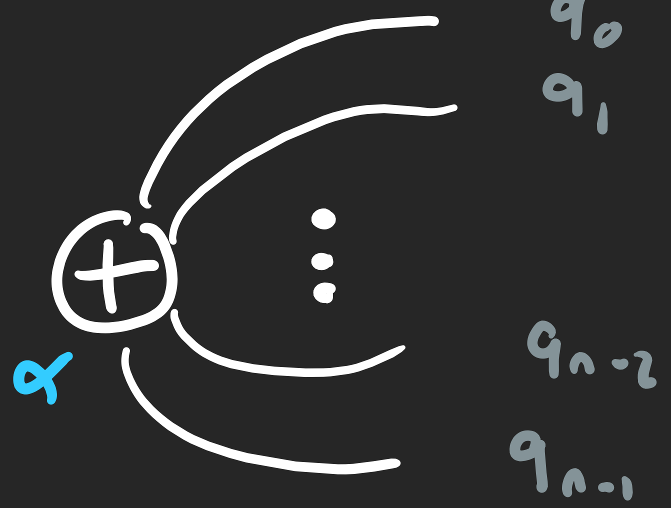
Z basis:

$$|00\ldots00\rangle + e^{i\alpha}|11\ldots11\rangle \longleftrightarrow$$



X basis:

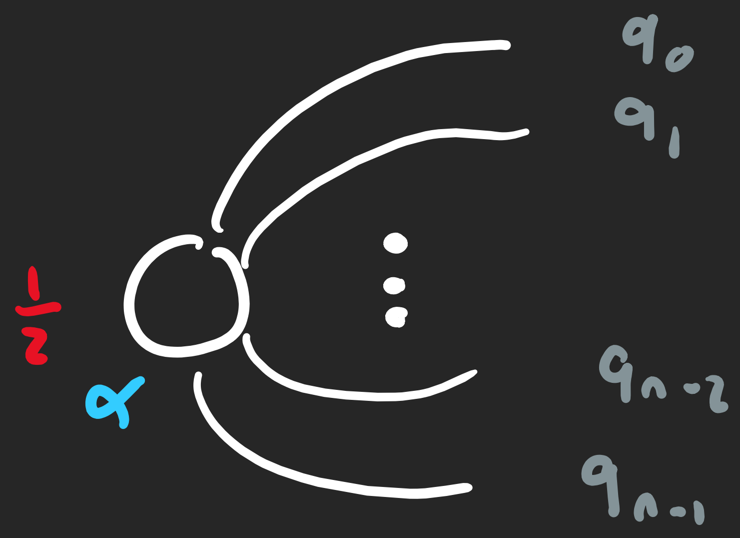
$$|++\ldots++\rangle + e^{i\alpha}|++\ldots++\rangle \longleftrightarrow$$



GHZ States

Z basis:

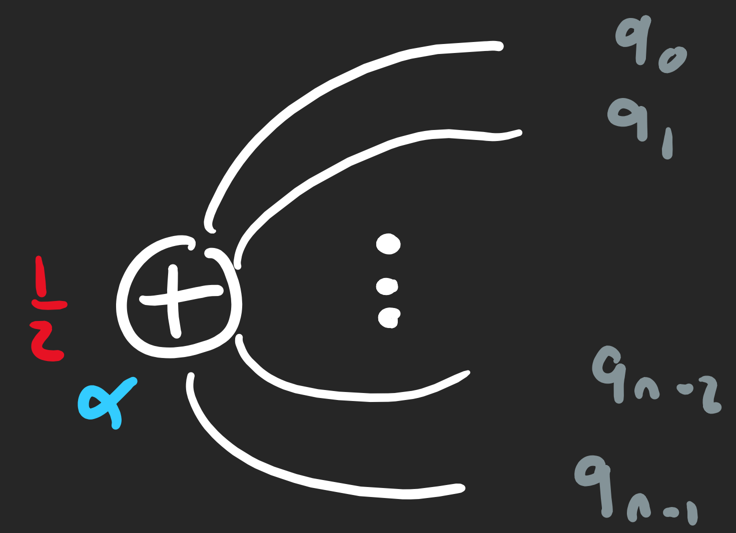
$$\frac{1}{\sqrt{2}}\left(|00\ldots00\rangle + e^{i\alpha}|11\ldots11\rangle\right) \longleftrightarrow$$



(normalised)

X basis:

$$\frac{1}{\sqrt{2}}\left(|++\ldots++\rangle + e^{i\alpha}|++\ldots++\rangle\right) \longleftrightarrow$$



(normalised)

GHZ States

$$|GHZ_3\rangle := \frac{1}{\sqrt{2}}\left(|000\rangle + |111\rangle\right) \longleftrightarrow$$

$$|GHZ_4\rangle := \frac{1}{\sqrt{2}}\left(|000\rangle + |111\rangle\right) \longleftrightarrow$$

Braket Notation

Rows/Columns indexed by n-bit strings $\underline{b} \in \{0,1\}^n$

Transformations of n-qubits:

$$U = \begin{pmatrix} U_{0\ldots0,0\ldots0} & U_{0\ldots0,1\ldots1} \\ U_{1\ldots1,0\ldots0} & U_{1\ldots1,1\ldots1} \end{pmatrix} \in \mathbb{C}^{2^n \times 2^n}$$

which are unitary: $U^{\dagger}U = I = UU^{\dagger}$

Bracket Notation

Outer products:

$$|\underline{b}\rangle\langle\underline{b}'| = \begin{array}{cccccl} & & \text{col } \underline{b}' & & & \\ & & \downarrow & & & \\ \left(\begin{matrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{matrix}\right. & \begin{matrix} \cdots \\ \\ \cdots \\ \\ \cdots \end{matrix} & \begin{matrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{matrix} & \begin{matrix} \cdots \\ \\ \cdots \\ \\ \cdots \end{matrix} & \left.\begin{matrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{matrix}\right) & \begin{matrix} \\ \\ \leftarrow \text{row } \underline{b} \\ \\ \\ \end{matrix} \end{array}$$

$$|01\rangle\langle10| = \begin{array}{c|cccc} & \langle 00| & \langle 01| & \langle 10| & \langle 11| \\ \hline |00\rangle & 0 & 0 & 0 & 0 \\ |01\rangle & 0 & 0 & 1 & 0 \\ |10\rangle & 0 & 0 & 0 & 0 \\ |11\rangle & 0 & 0 & 0 & 0 \end{array}$$

Braket Notation

Rows/Columns indexed by n-bit strings $\underline{b} \in \{0,1\}^n$

Transformations of n-qubits:

$$U = \sum_{\underline{b}, \underline{b}' \in \{0,1\}^n} |\underline{b}\rangle U_{\underline{b},\underline{b}'} \langle \underline{b}'|$$

which are unitary: $U^\dagger U = I = U U^\dagger$

Braket Notation

each qubit transformed independantly

Product transformations:

$$U \otimes V = \begin{pmatrix} U_{00}\begin{pmatrix} V_{00} & V_{01} \\ V_{10} & V_{11} \end{pmatrix} & U_{01}\begin{pmatrix} V_{00} & V_{01} \\ V_{10} & V_{11} \end{pmatrix} \\ U_{10}\begin{pmatrix} V_{00} & V_{01} \\ V_{10} & V_{11} \end{pmatrix} & U_{11}\begin{pmatrix} V_{00} & V_{01} \\ V_{10} & V_{11} \end{pmatrix} \end{pmatrix}$$

$\uparrow$ unitary on q_1 (under U)

$\uparrow$ unitary on q_0 (under V)

each qubit transformed independantly

Braket Notation

Product transformations:

$$U \otimes V = \begin{pmatrix} U_{00}V_{00} & U_{00}V_{01} & U_{01}V_{00} & U_{01}V_{01} \\ U_{00}V_{10} & U_{00}V_{11} & U_{01}V_{10} & U_{01}V_{11} \\ U_{10}V_{00} & U_{10}V_{01} & U_{11}V_{00} & U_{11}V_{01} \\ U_{10}V_{10} & U_{10}V_{11} & U_{11}V_{10} & U_{11}V_{11} \end{pmatrix}$$

unitary on q_1

unitary on q_0

Braket Notation

$$(U \otimes V)(|\psi\rangle \otimes |\varphi\rangle)$$

$$= \begin{pmatrix} U_{00}\begin{pmatrix} v_{00} & v_{01} \\ v_{10} & v_{11} \end{pmatrix} & U_{01}\begin{pmatrix} v_{00} & v_{01} \\ v_{10} & v_{11} \end{pmatrix} \\ U_{10}\begin{pmatrix} v_{00} & v_{01} \\ v_{10} & v_{11} \end{pmatrix} & U_{11}\begin{pmatrix} v_{00} & v_{01} \\ v_{10} & v_{11} \end{pmatrix} \end{pmatrix} \begin{pmatrix} \psi_0\begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} \\ \psi_1\begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} \end{pmatrix}$$

Braket Notation

$$(U \otimes V)(|\psi\rangle \otimes |\varphi\rangle)$$

$$= \begin{pmatrix} U_{00}\psi_0 \begin{pmatrix} v_{00} & v_{01} \\ v_{10} & v_{11} \end{pmatrix}\begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} + U_{01}\psi_1 \begin{pmatrix} v_{00} & v_{01} \\ v_{10} & v_{11} \end{pmatrix}\begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} \\ U_{10}\psi_0 \begin{pmatrix} v_{00} & v_{01} \\ v_{10} & v_{11} \end{pmatrix}\begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} + U_{11}\psi_1 \begin{pmatrix} v_{00} & v_{01} \\ v_{10} & v_{11} \end{pmatrix}\begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} \end{pmatrix}$$

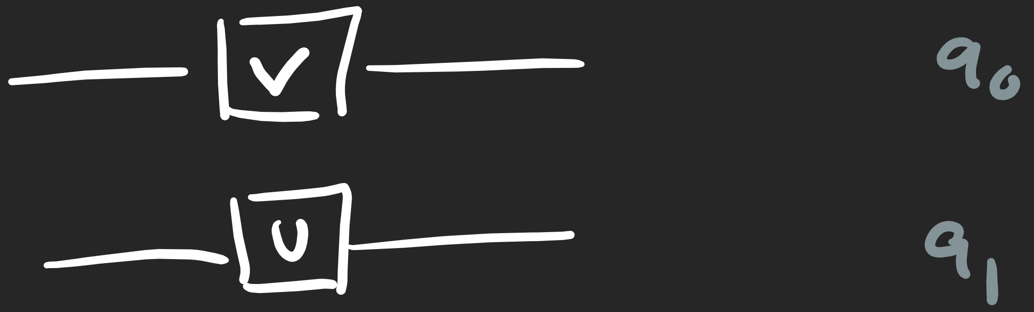
Braket Notation

$$(U \otimes V)(|\psi\rangle \otimes |\varphi\rangle)$$

$$= \begin{pmatrix} (U_{00}\psi_0 + U_{01}\psi_1)\begin{pmatrix} V_{00} & V_{01} \\ V_{10} & V_{11} \end{pmatrix}\begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} \\ \\ (U_{10}\psi_0 + U_{11}\psi_1)\begin{pmatrix} V_{00} & V_{01} \\ V_{10} & V_{11} \end{pmatrix}\begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} \end{pmatrix} = (U|\psi\rangle) \otimes (V|\varphi\rangle)$$

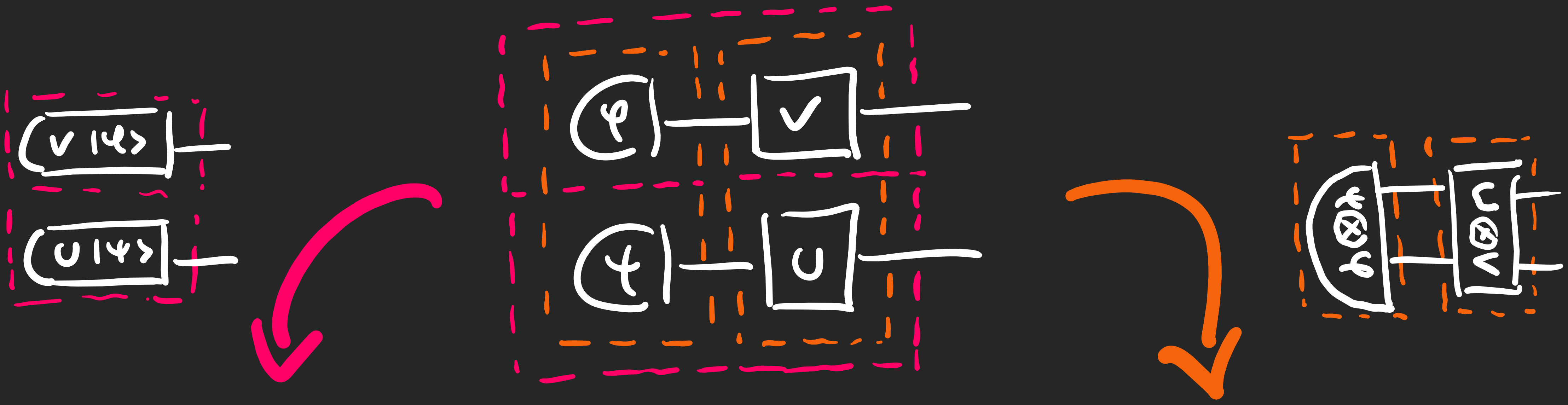
Graphical Notation

$$U \otimes V \longleftrightarrow$$



Graphical Notation

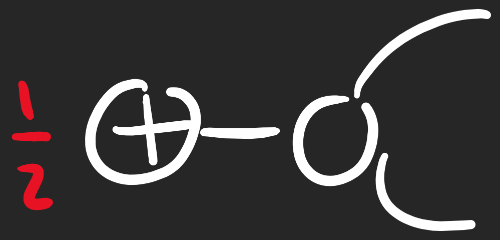
We get equations "for free" from geometry:

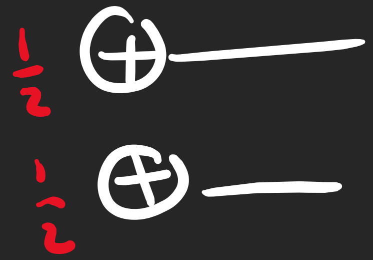


$$(U|\psi\rangle)\otimes(V|\varphi\rangle) = (U\otimes V)(|\psi\rangle\otimes|\varphi\rangle)$$

Copy Maps

$|00\rangle\langle 0| + |11\rangle\langle 1| \longleftrightarrow$ 

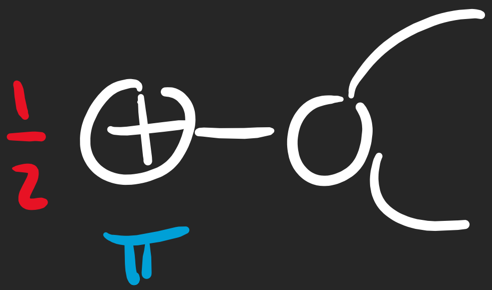
 $= \left(|00\rangle\langle 0| + |11\rangle\langle 1|\right)|0\rangle$

$= |00\rangle\cancel{\langle 0|0\rangle}^{1} + |11\rangle\cancel{\langle 1|0\rangle}^{0} =$ 

$|00\rangle$

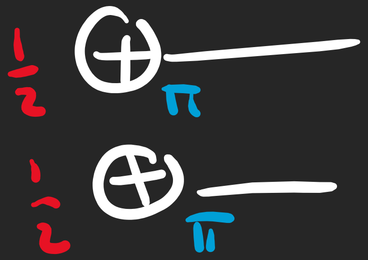
Copy Maps

$|00\rangle\langle 0| + |11\rangle\langle 1| \longleftrightarrow$ 



$= (|00\rangle\langle 0| + |11\rangle\langle 1|)|1\rangle$

$= |00\rangle\cancel{\langle 0|1\rangle} + |11\rangle\cancel{\langle 1|1\rangle}$ =

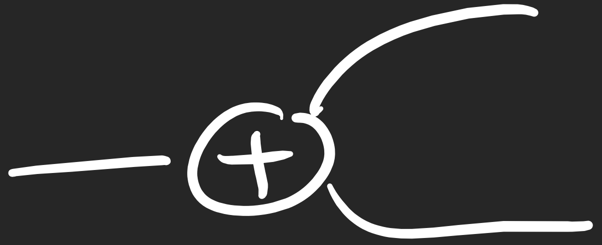


$|11\rangle$

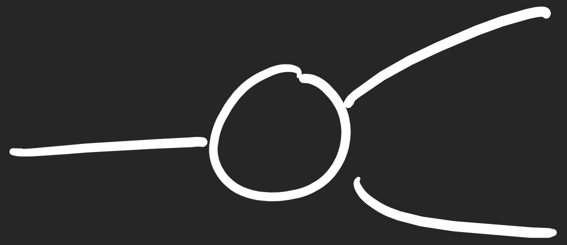
Copy Maps

$|00\rangle\langle 0| + |11\rangle\langle 1| \quad \longleftrightarrow$

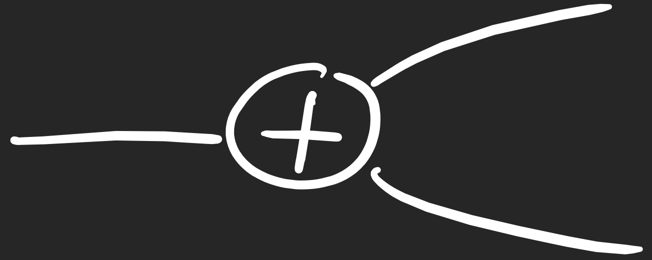
$|{+}{+}\rangle\langle {+}| + |{-}{-}\rangle\langle {-}| \quad \longleftrightarrow$



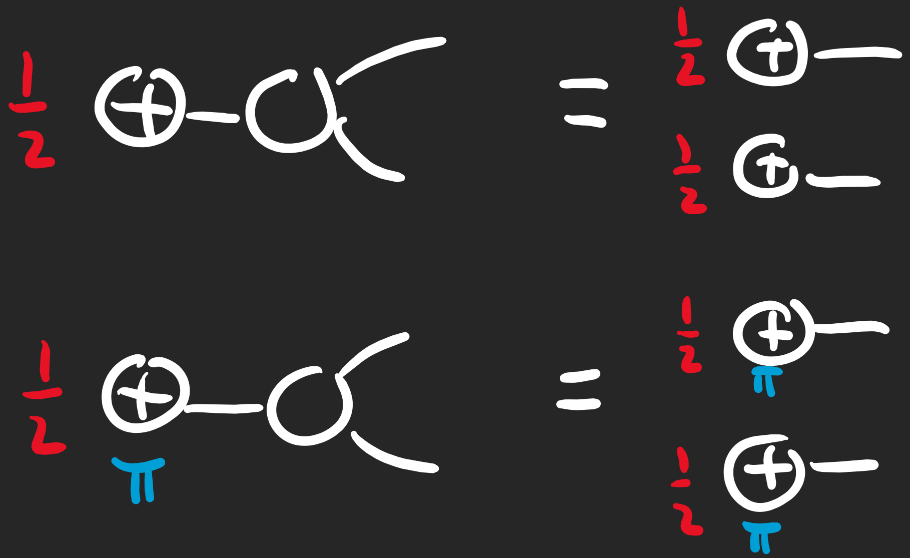
Copy Maps

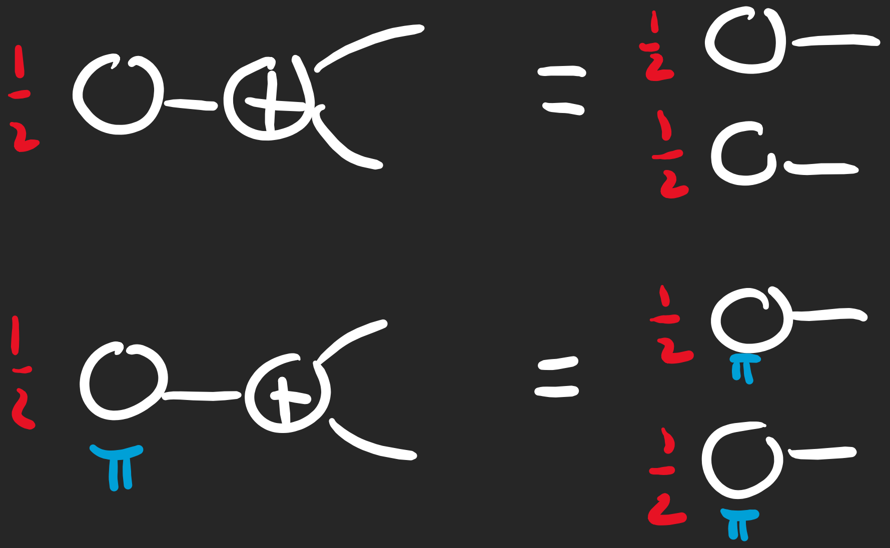


Z basis copy

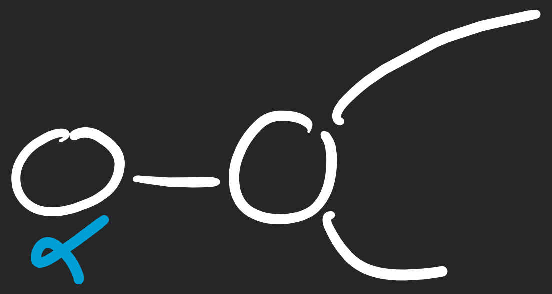


X basis copy





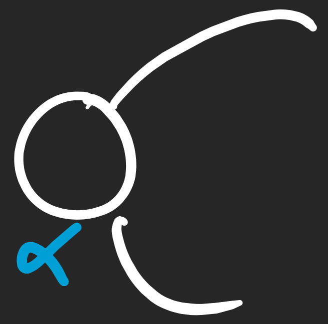
Copy Maps



$$= \left(|00\rangle\langle 0| + |11\rangle\langle 1|\right)\left(|0\rangle + e^{i\alpha}|1\rangle\right)$$

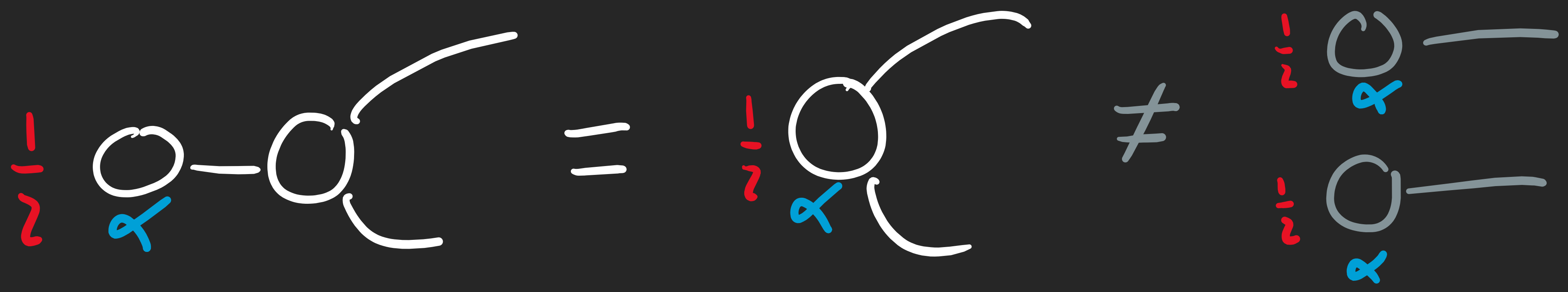
$$= |00\rangle\overset{1}{\cancel{\langle 0|0\rangle}} + e^{i\alpha}|00\rangle\overset{0}{\cancel{\langle 0|1\rangle}}$$
$$+ |11\rangle\overset{0}{\cancel{\langle 1|0\rangle}} + e^{i\alpha}|11\rangle\overset{1}{\cancel{\langle 1|1\rangle}}$$

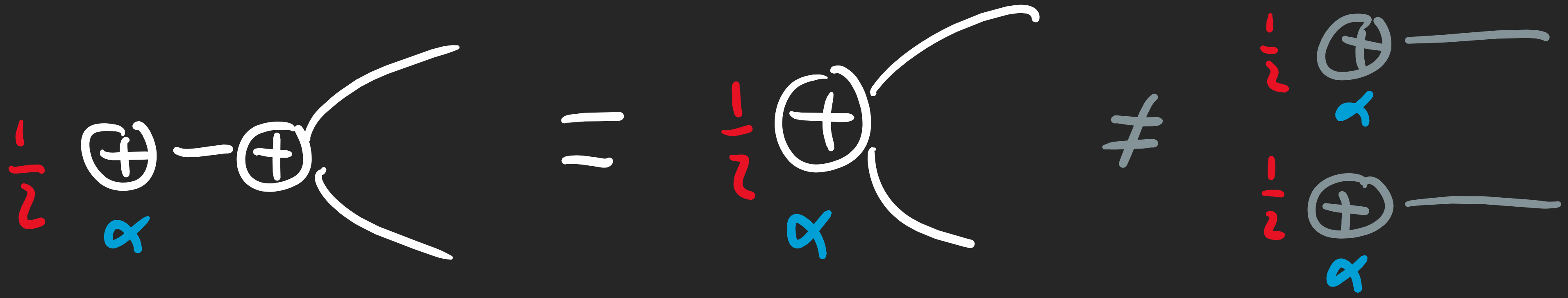
$$= |00\rangle + e^{i\alpha}|11\rangle =$$



Copy Maps

BEWARE! Not all states are copied!





Binary Arithmetic

Numbers : $\mathbb{Z}_2 := \{0, 1\}$

Operations : $\oplus$ (addition mod 2) , $\cdot$ (multiplication)

$\oplus$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

~~Binary~~ Boolean ~~Arithmetic~~ Logic

~~Numbers~~ Truth values : $\mathbb{B} := \{F, T\}$

Operations : $\oplus$ (logical xor), $\wedge$ (logical and)

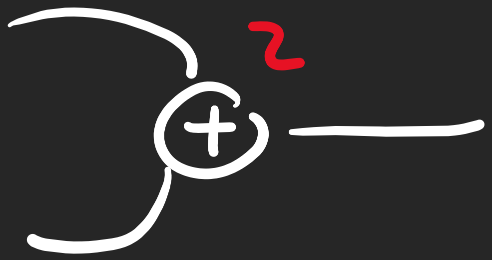
$\oplus$	F	T
F	F	T
T	T	F

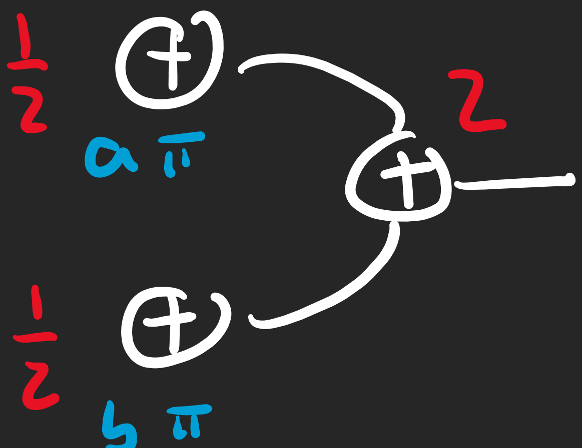
$\wedge$	F	T
F	F	F
T	F	T

Note: $(a+b)\pi = (a \oplus b)\pi \mod 2\pi$

Binary Arithmetic

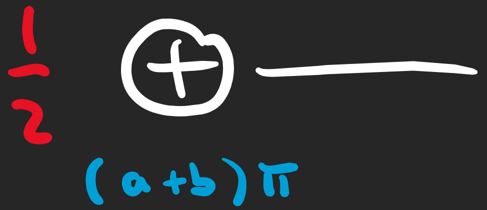
$$\sum_{a,b\in\{0,1\}} |b \oplus a\rangle\langle ba| \longleftrightarrow$$





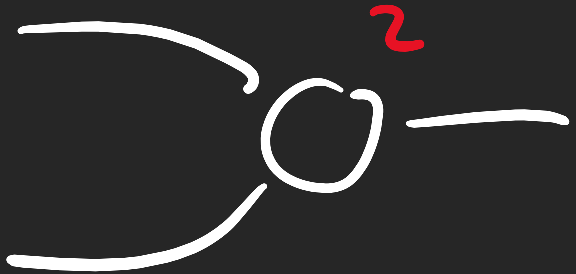
$$= \sum_{a',b'\in\{0,1\}} |b' \oplus a'\rangle\langle b'a'|ba\rangle$$

$$= |b \oplus a\rangle =$$

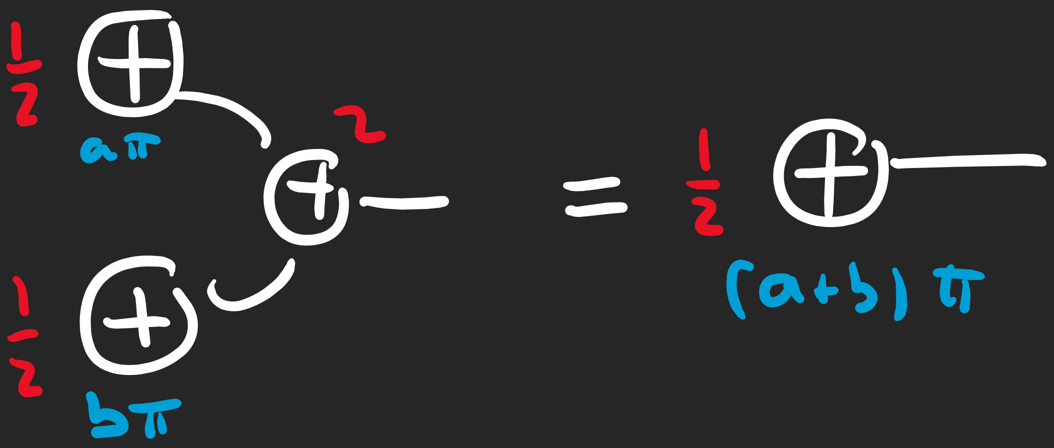


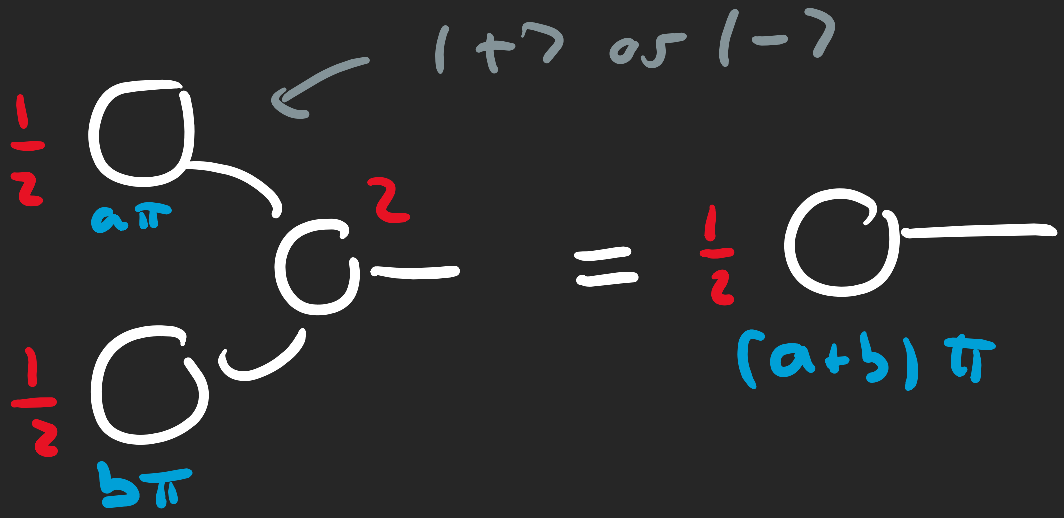
Addition Maps

Binary addition (Z basis)

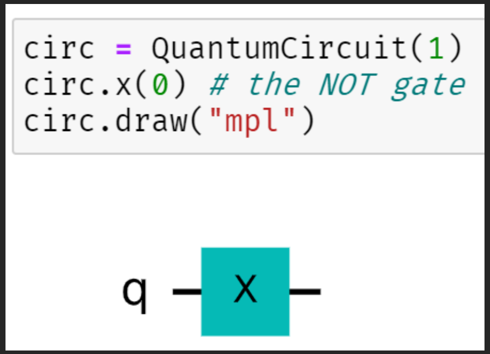


Binary addition (X basis)





The NOT Gate

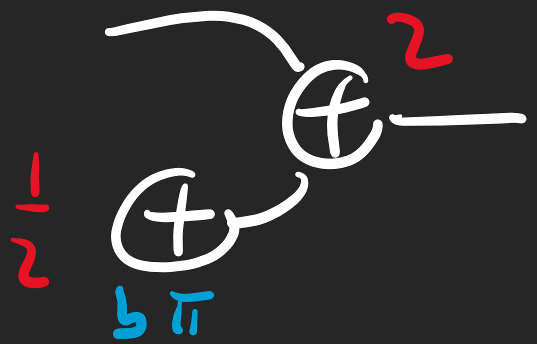


$\mathrm{NOT}\,|0\rangle = |1\rangle$

$\mathrm{NOT}\,|1\rangle = |0\rangle$

The NOT Gate

e.g. $|0\rangle \otimes I = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}1&0\\0&1\end{pmatrix} = \begin{pmatrix}1&0\\0&1\\0&0\\0&0\end{pmatrix}$



qubit q_0

$$= \left(\sum_{a',b'\in\{0,1\}} |b'\oplus a'\rangle\langle b'a'|\right)\left(|b\rangle \otimes I\right)$$

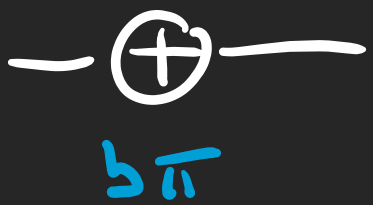
qubit q_1

$$= \sum_{a'\in\{0,1\}} |b\oplus a'\rangle\langle a'|$$

= {

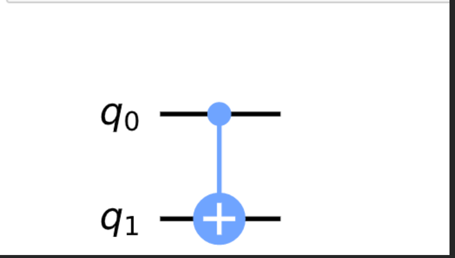
 if $b=0$

if $b=1$

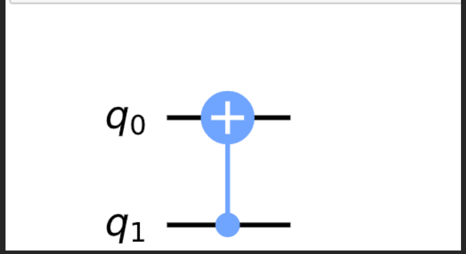
= 

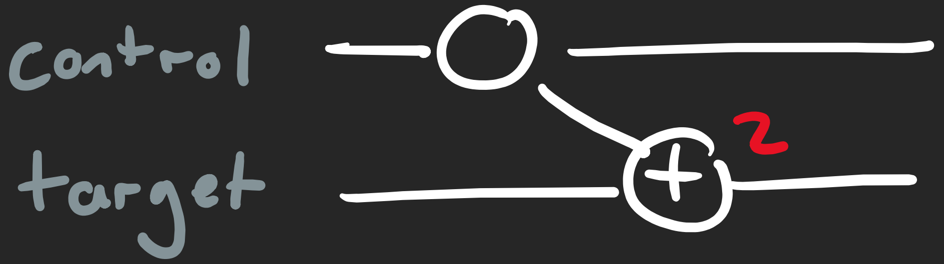
The CNOT Gate

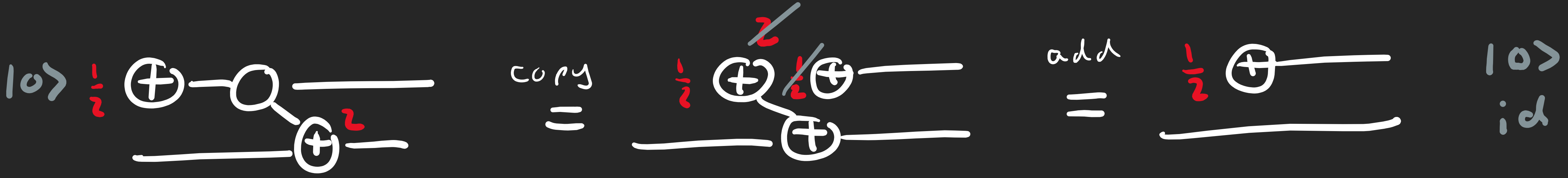
```
circ = QuantumCircuit(2)
circ.cx(0,1) # the CNOT gate
circ.draw("mpl")
```

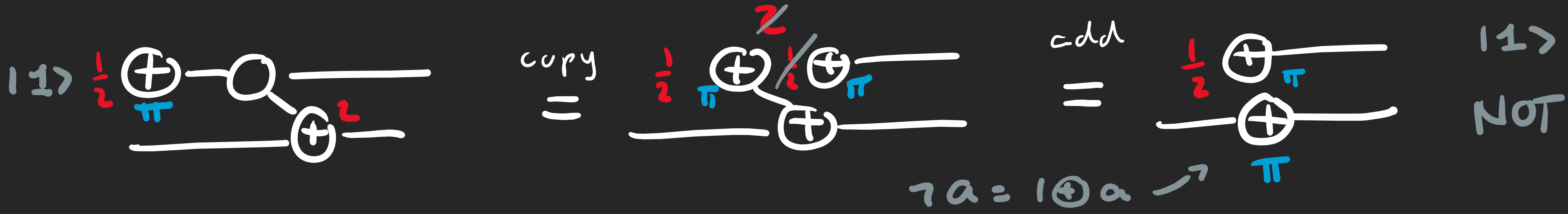



```
circ = QuantumCircuit(2)
circ.cx(1,0) # the CNOT gate
circ.draw("mpl")
```

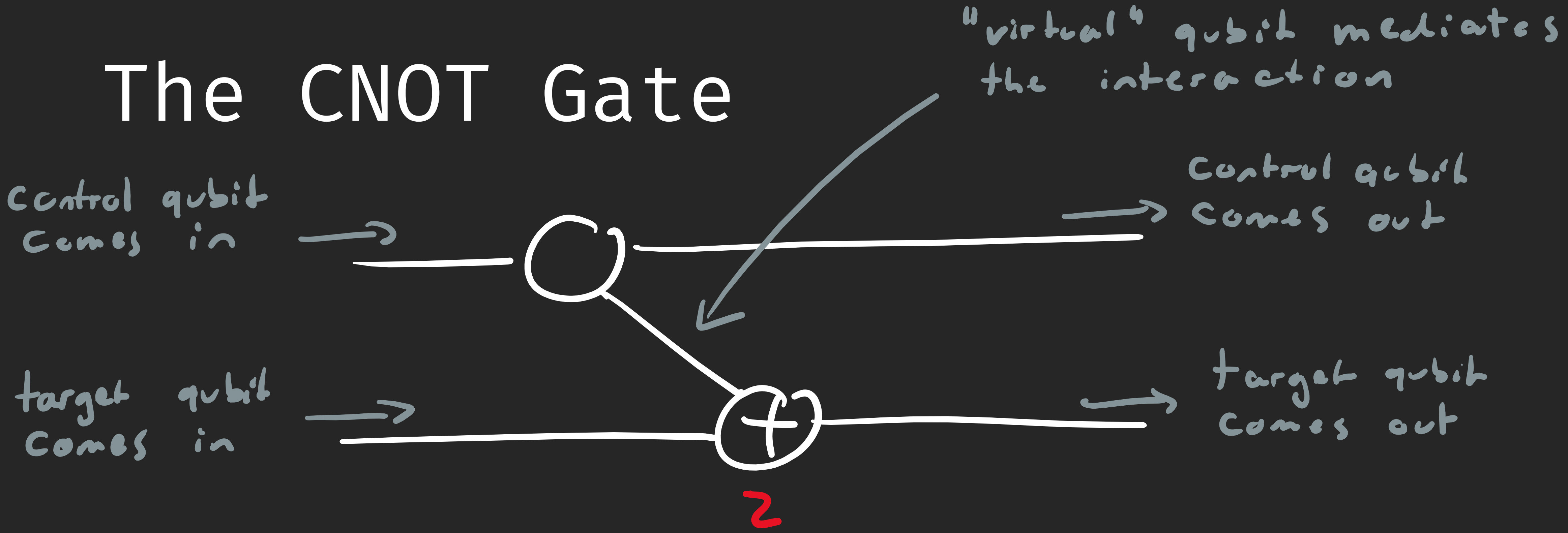




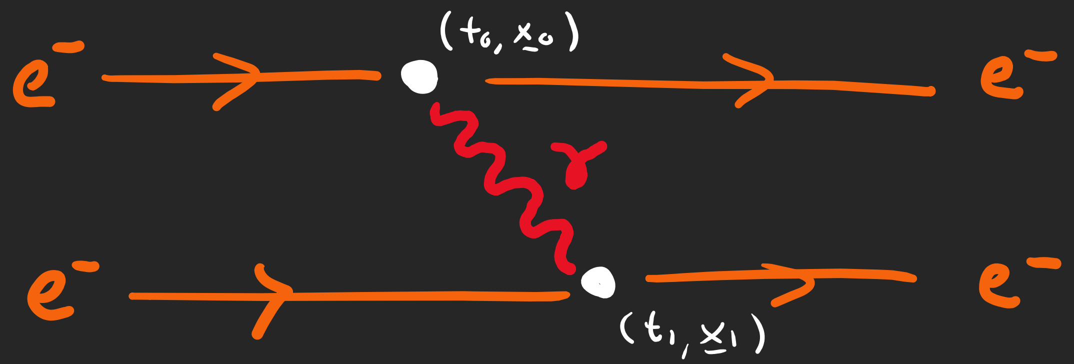




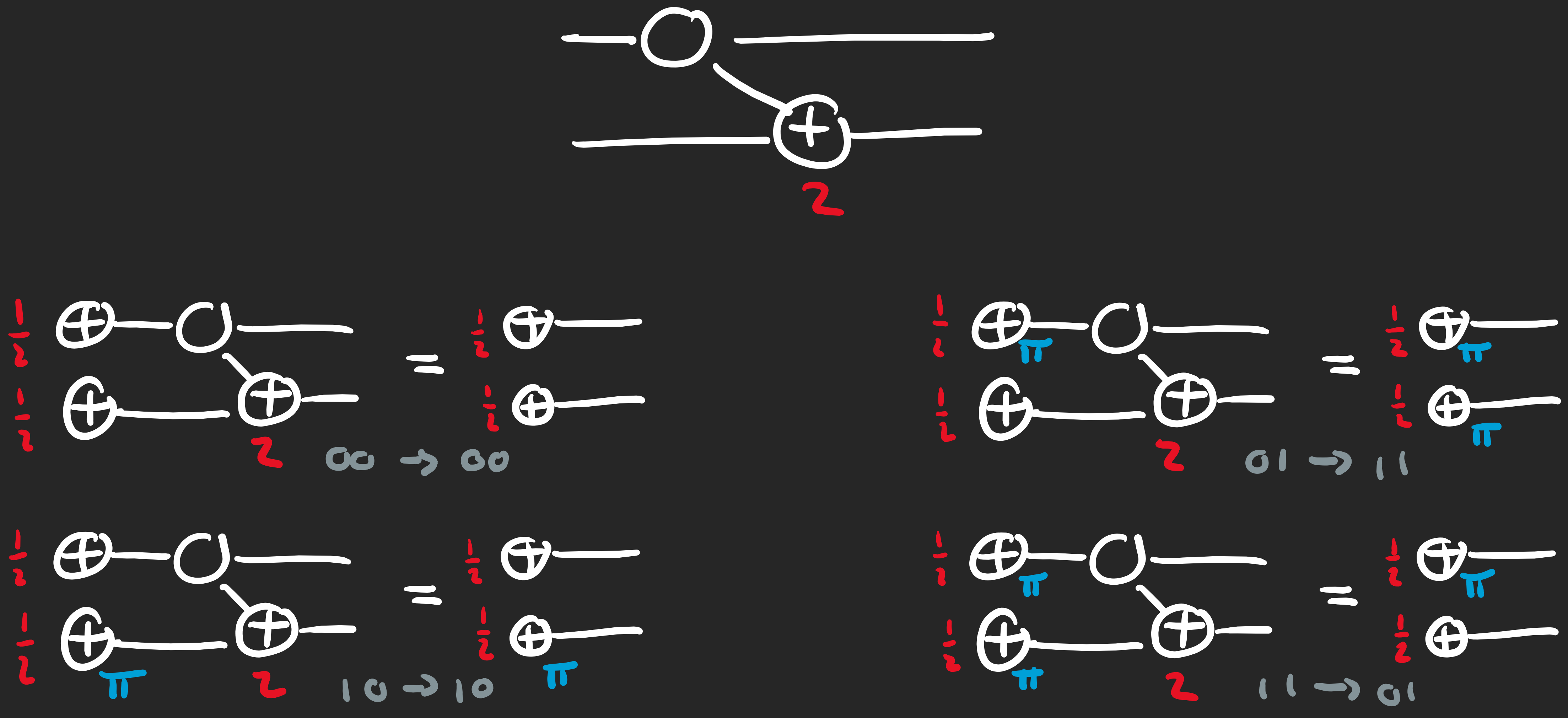
The CNOT Gate



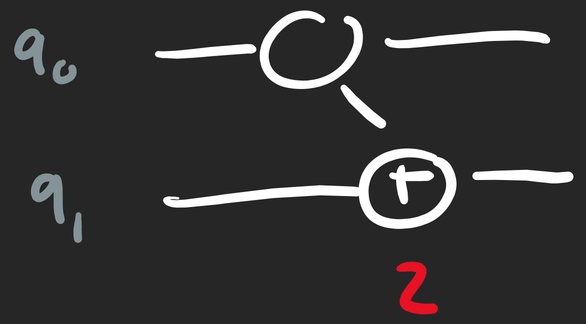
Doesn't look much different from a Feynman Diagram:



The CNOT Gate



The CNOT Gate



$\longleftrightarrow$

$$\begin{array}{cc} q_1\, q_0 & q_1\, q_0 \\ \downarrow\ \downarrow & \downarrow\ \downarrow \end{array}$$

$$\begin{aligned} & |00\rangle\langle 00| \\ + & |11\rangle\langle 01| \\ + & |10\rangle\langle 10| \\ + & |01\rangle\langle 11| \end{aligned} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$|00\rangle \mapsto |00\rangle$

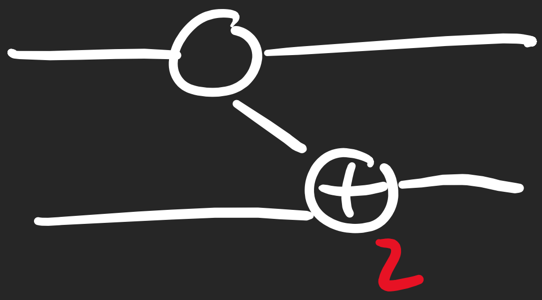
$|01\rangle \mapsto |11\rangle$

$|10\rangle \mapsto |10\rangle$

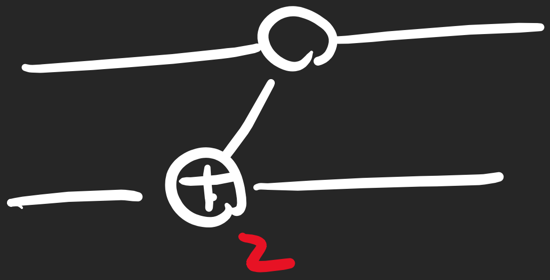
$|11\rangle \mapsto |11\rangle$

The CNOT Gate

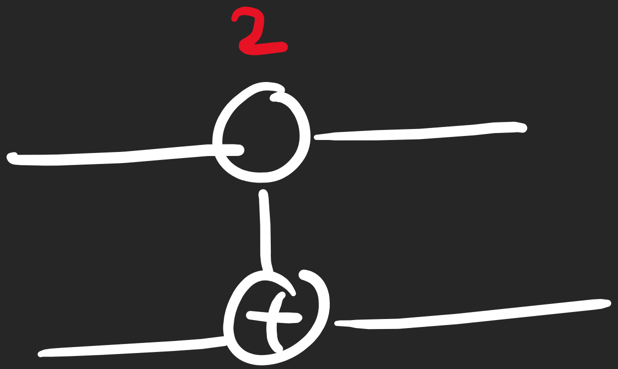
We'll prove later on that:



=



So the CNOT can be drawn symmetrically:



Exercise

TODO

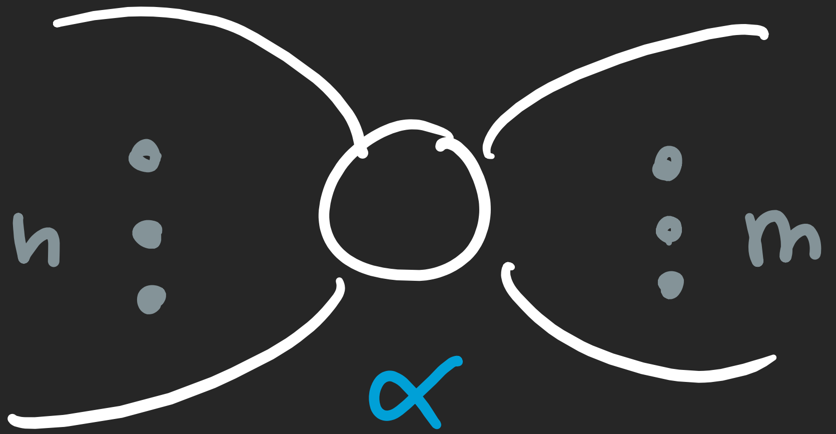
Bell

Basis

Spiders

Any number $n \geq 0$ of input legs/qubits

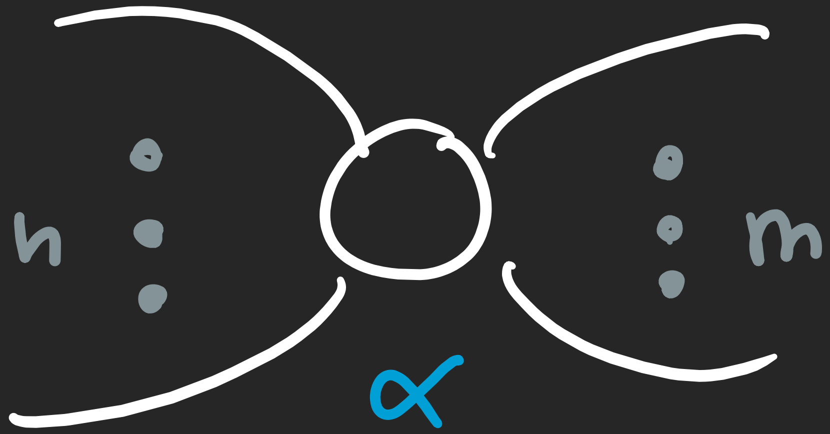
Any number $m \geq 0$ of output legs/qubits



Any angle (known as the **phase**)

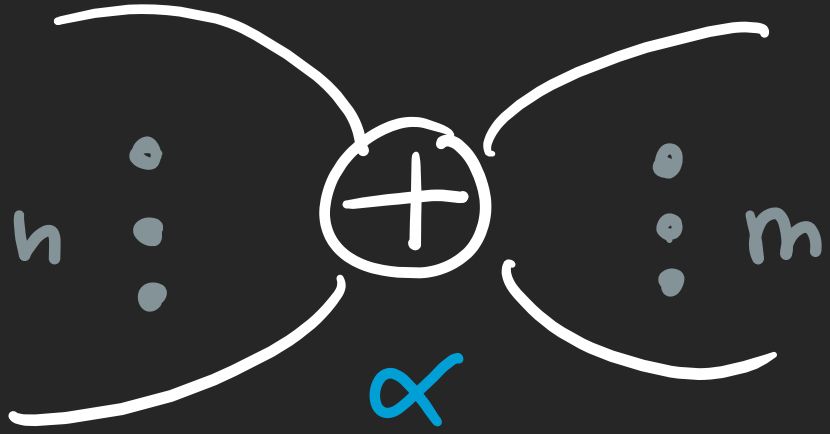
Spiders

Z Spider



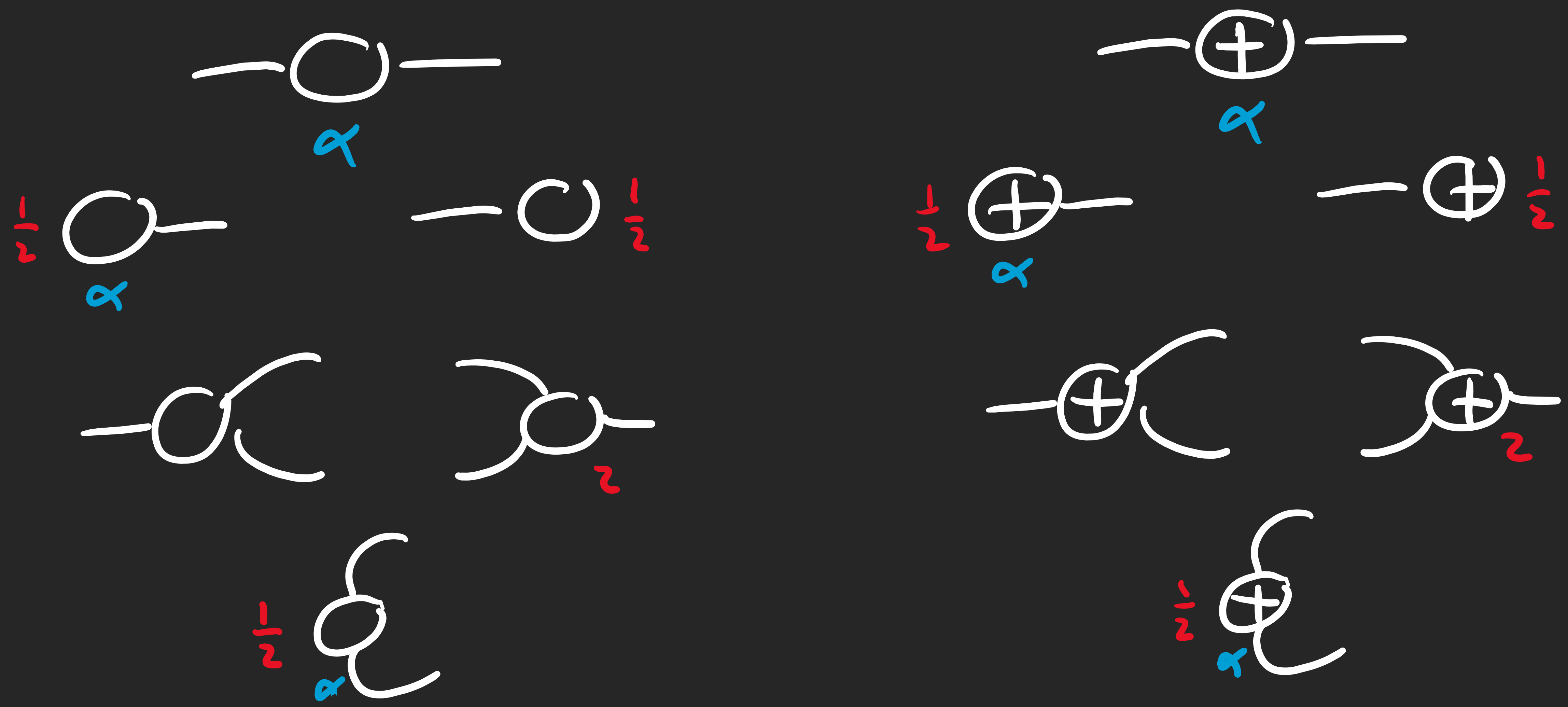
$$\underbrace{|0\ldots0\rangle}_{m}\underbrace{\langle 0\ldots0|}_{n} + e^{i\alpha}\underbrace{|1\ldots1\rangle}_{m}\underbrace{\langle 1\ldots1|}_{n}$$

X spider

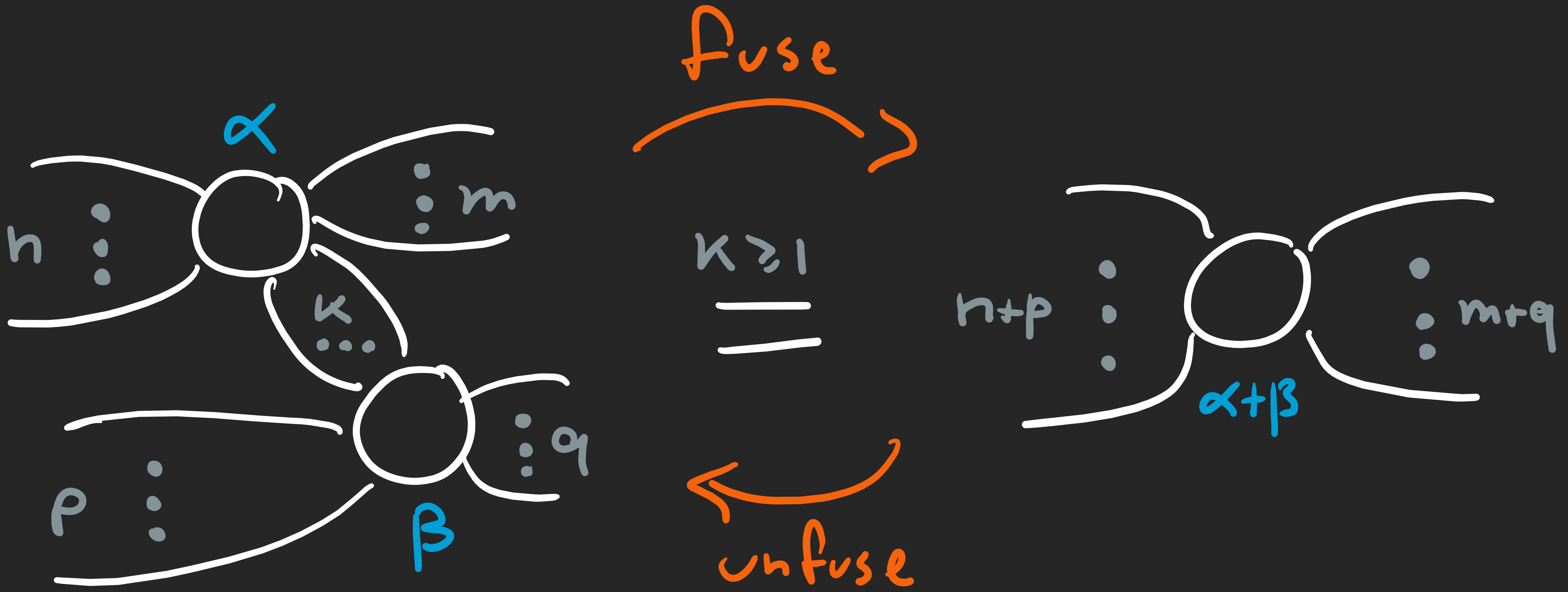


$$\underbrace{|+\ldots+\rangle}_{m}\underbrace{\langle +\ldots+|}_{n} + e^{i\alpha}\underbrace{|-\ldots-\rangle}_{m}\underbrace{\langle -\ldots-|}_{n}$$

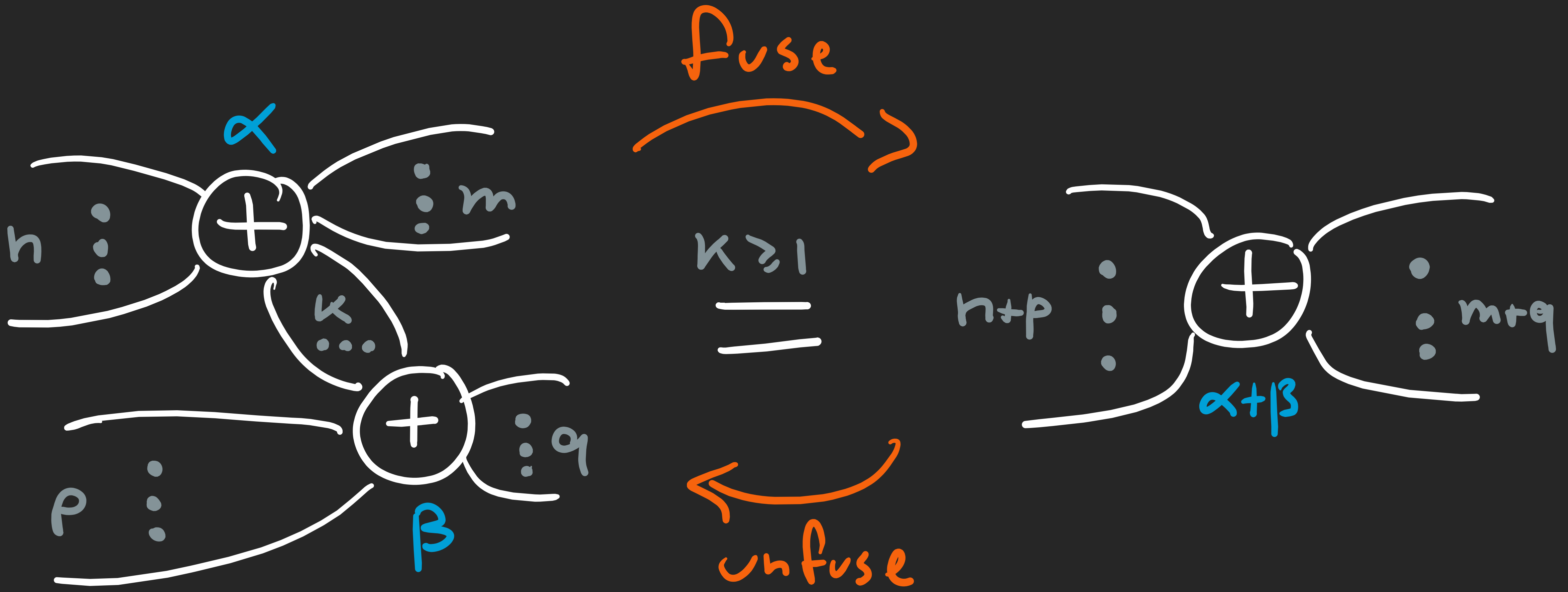
Spiders

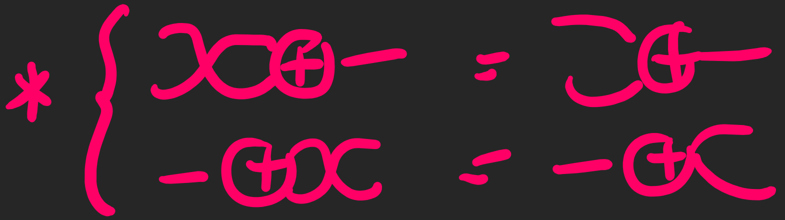


Spider Fusion



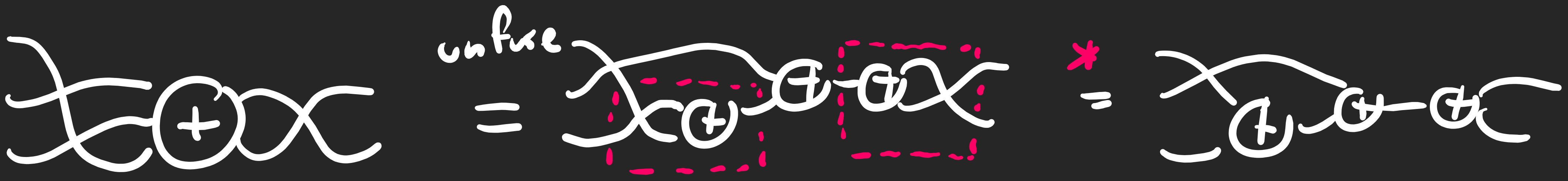
Spider Fusion



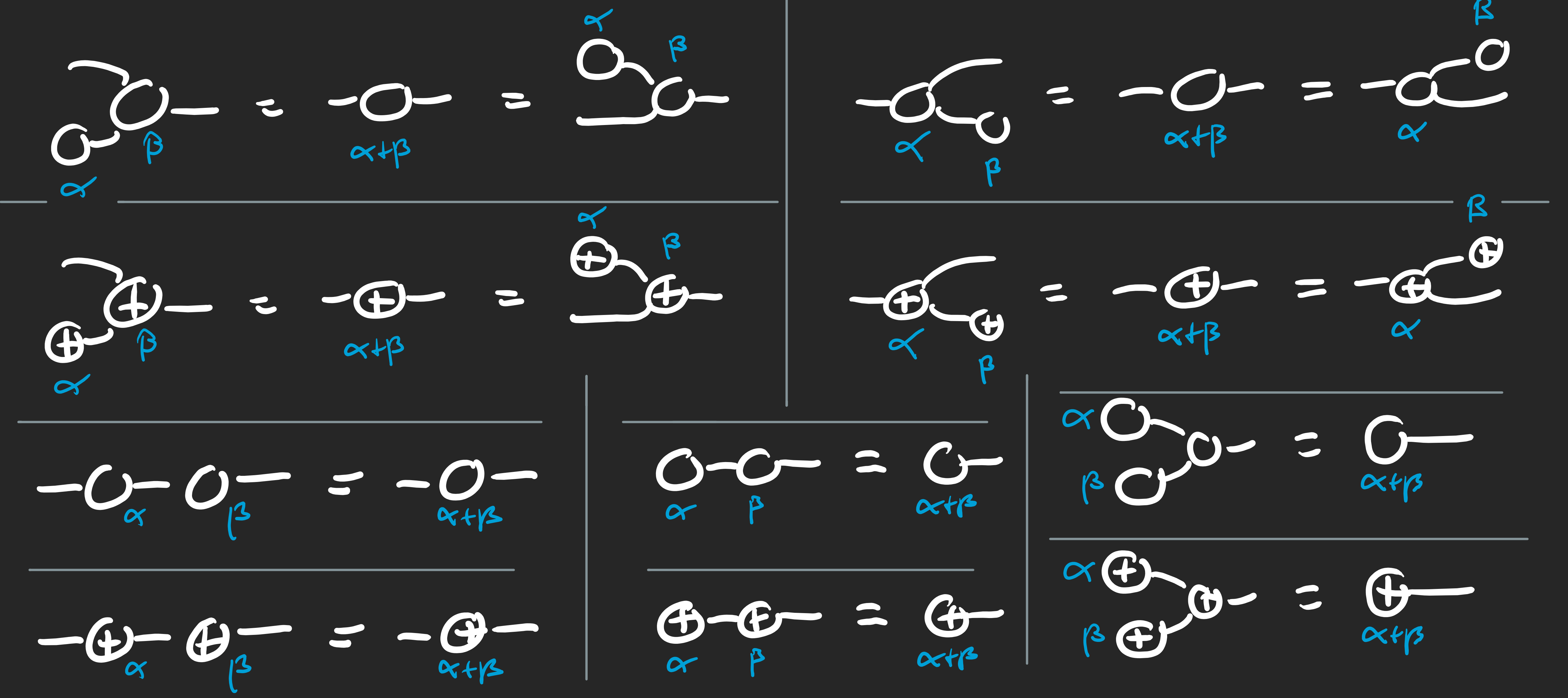


Commutativity

Leg order in spiders is irrelevant, e.g.

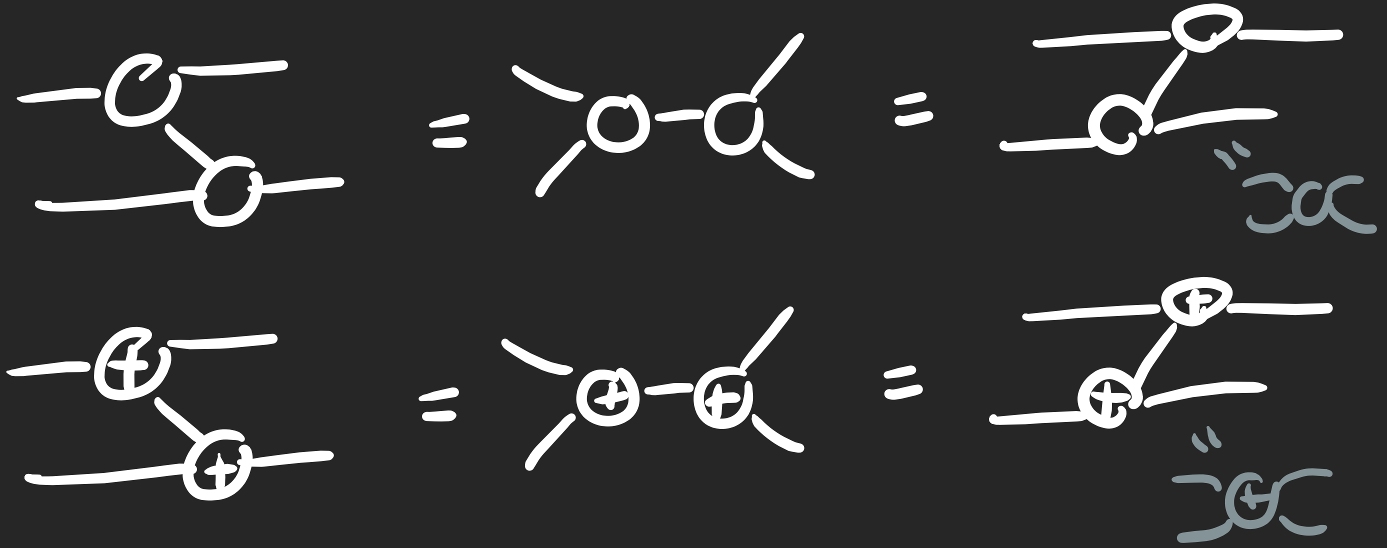


Spider Fusion

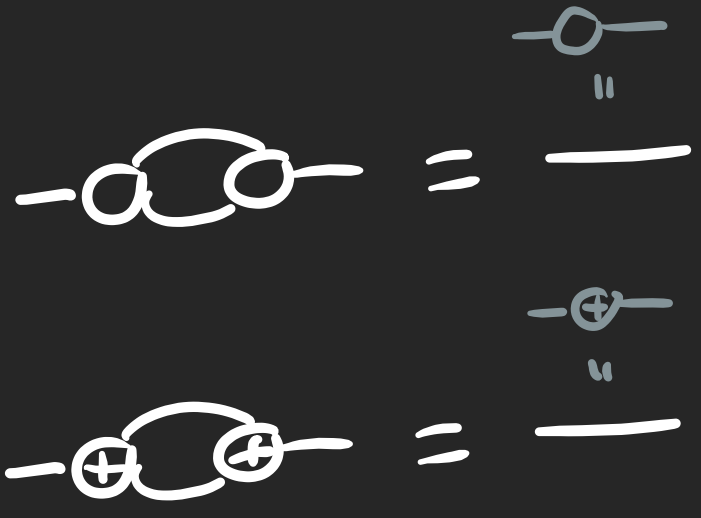


Spider Fusion

Spider fusion also relates copy maps of one basis to addition maps of the other:

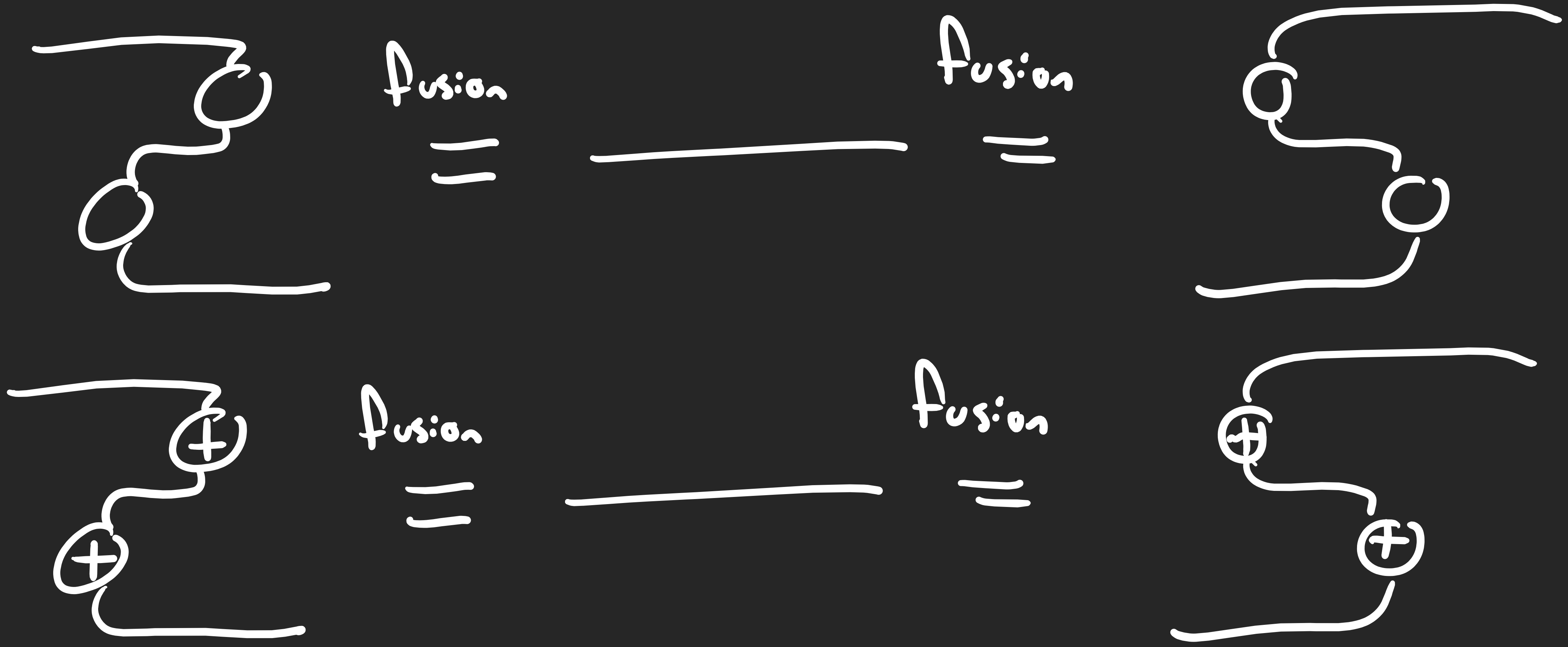


Frobenius Law

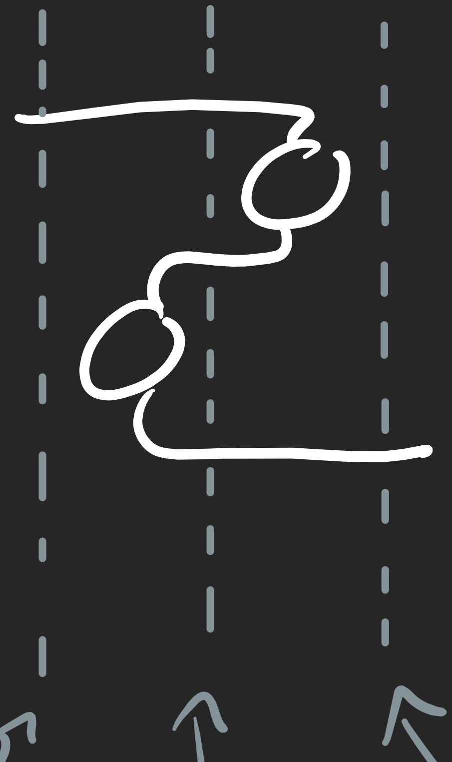


Isometry Law

Snake Equations

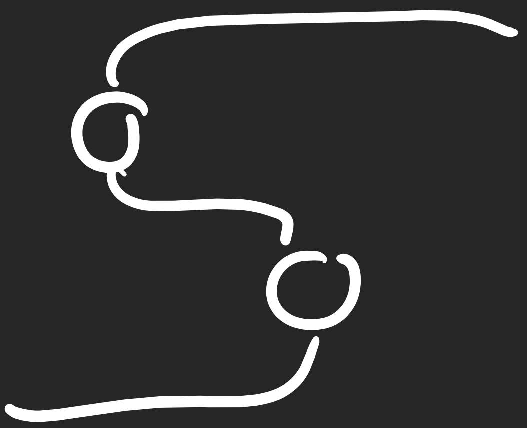


Snake Equations

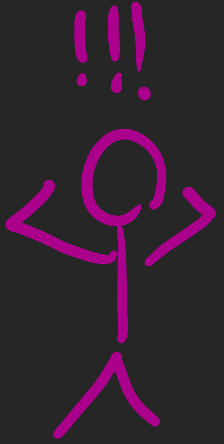


fusion $=$

fusion $=$

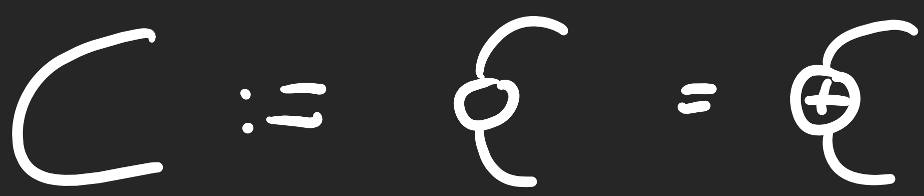


Spider fusion is a really powerful tool...



$|\psi\rangle$ $\quad (|00\rangle + |11\rangle) \otimes |\psi\rangle$ $\quad \left(\sum_b |b\rangle \left(\langle b00| + \langle b11|\right)\right)\left(\sum_{b'} \left(|00b'\rangle + |11b'\rangle\right)\psi_{b'}\right) =$

Snake Equations

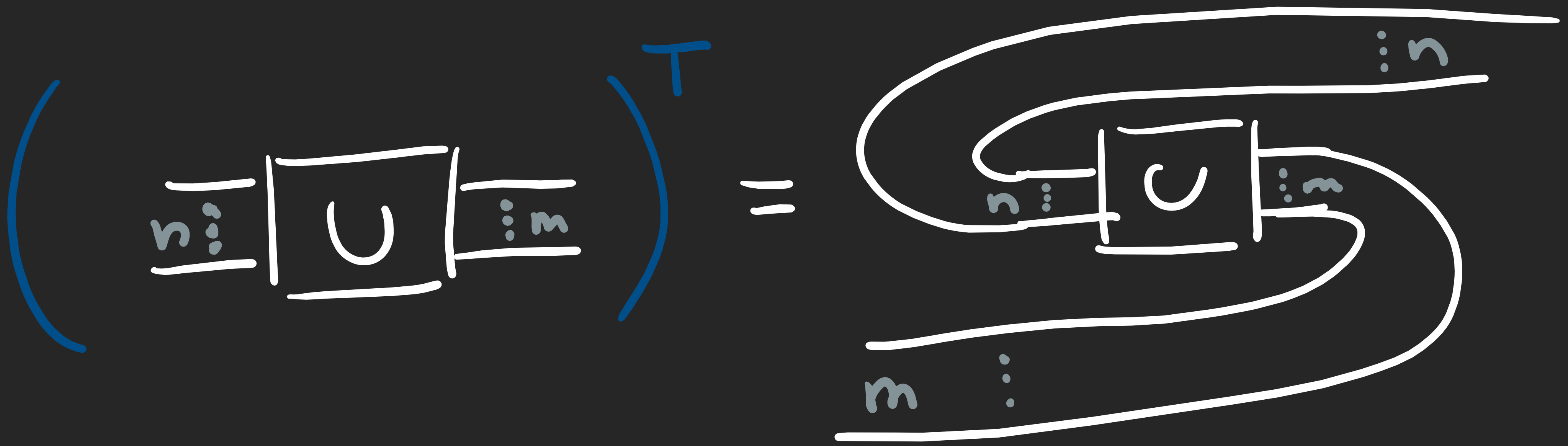


the cup

the cap

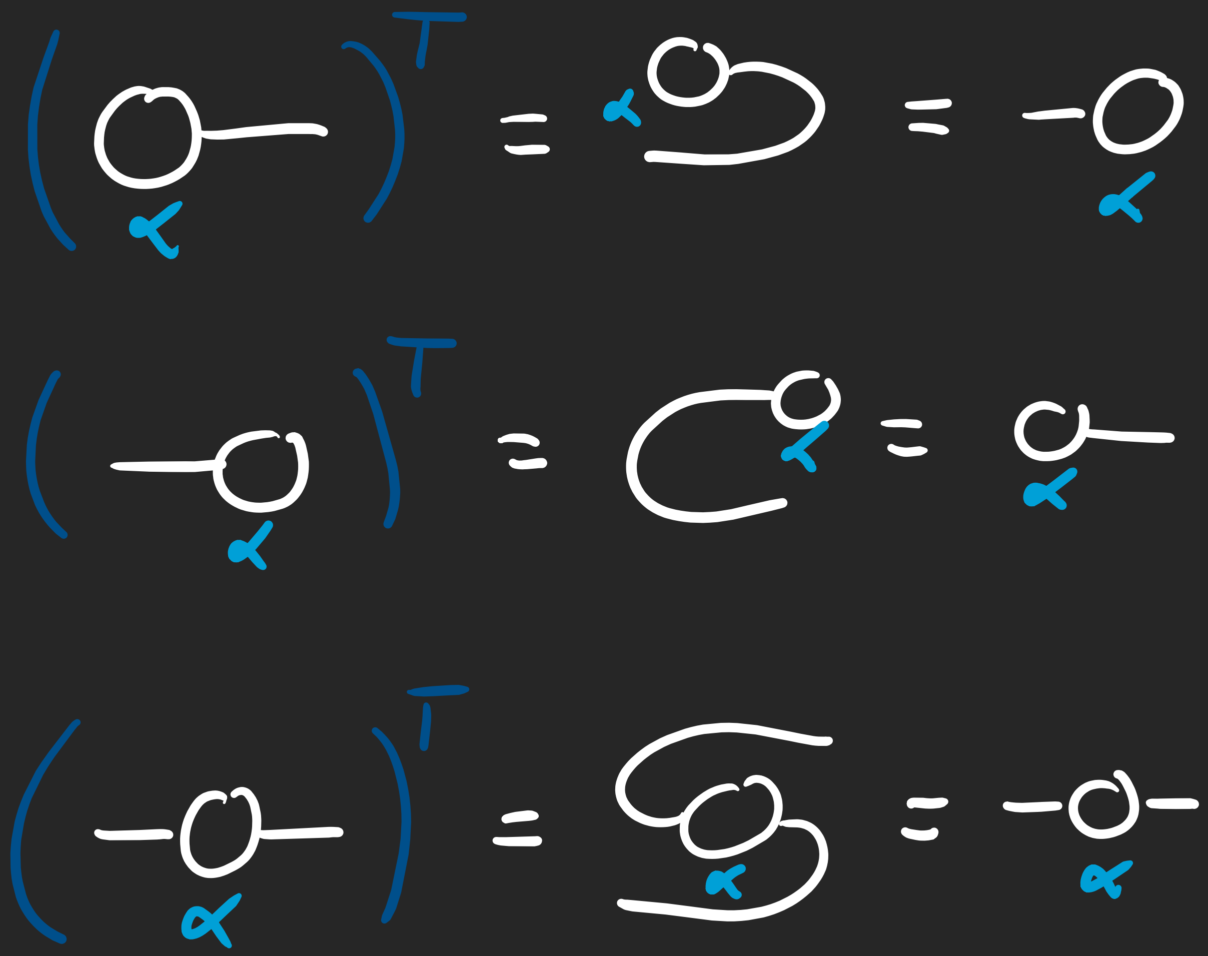
Snake Equations

The Transpose

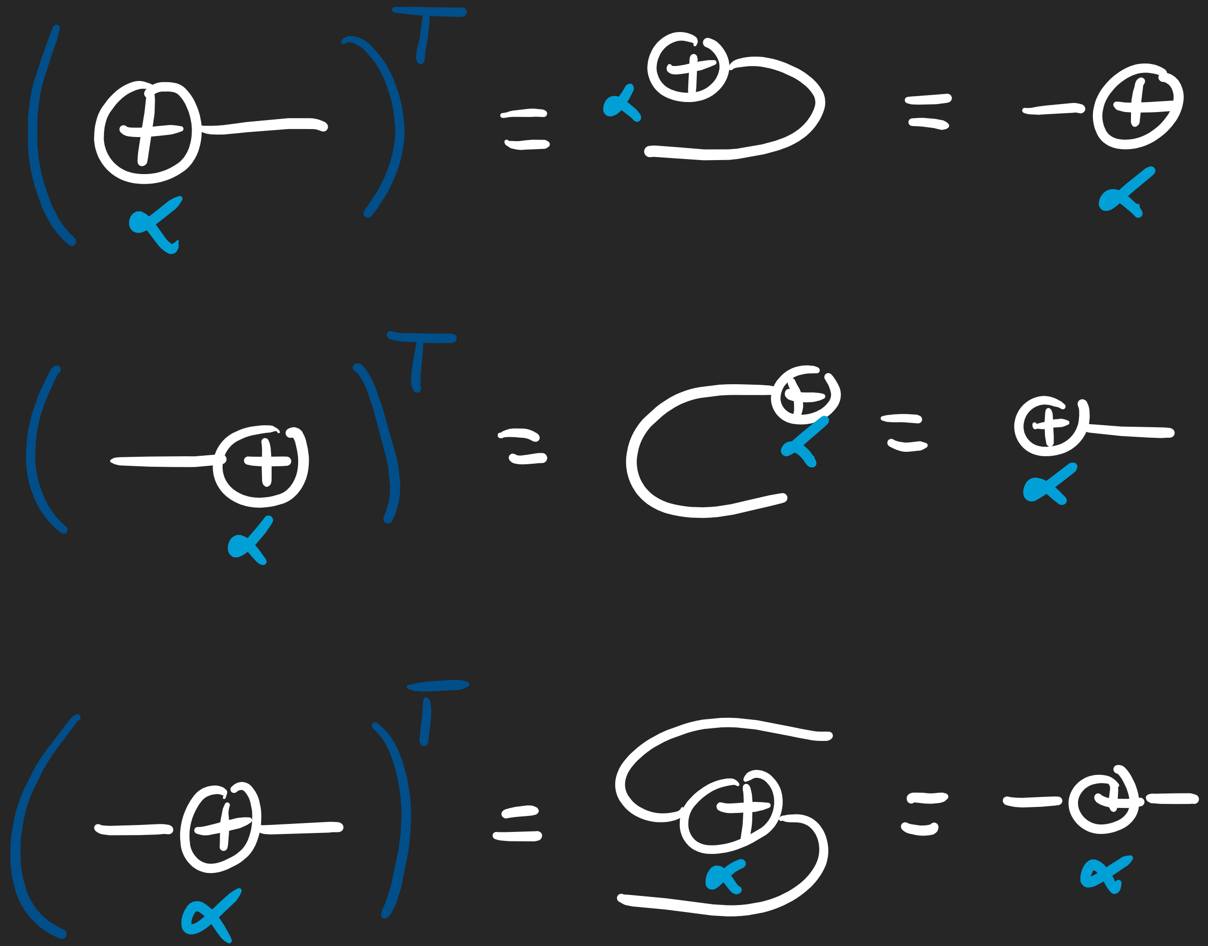


$$\sum_{\underline{b},\underline{b}'} |\underline{b}\rangle U_{\underline{b},\underline{b}'} \langle \underline{b}'| \longmapsto \sum_{\underline{b},\underline{b}'} |\underline{b}'\rangle U_{\underline{b},\underline{b}'} |\underline{b}\rangle$$

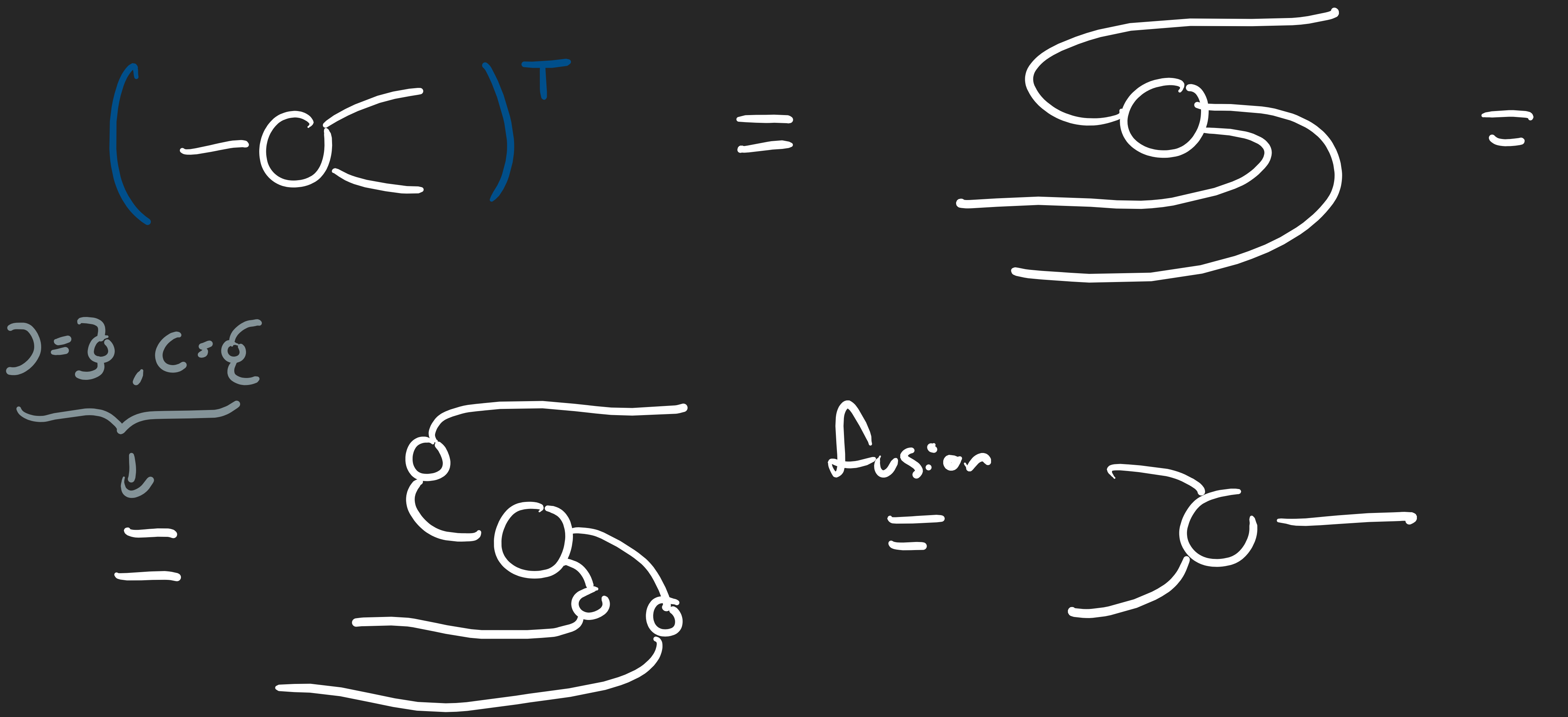
The Transpose



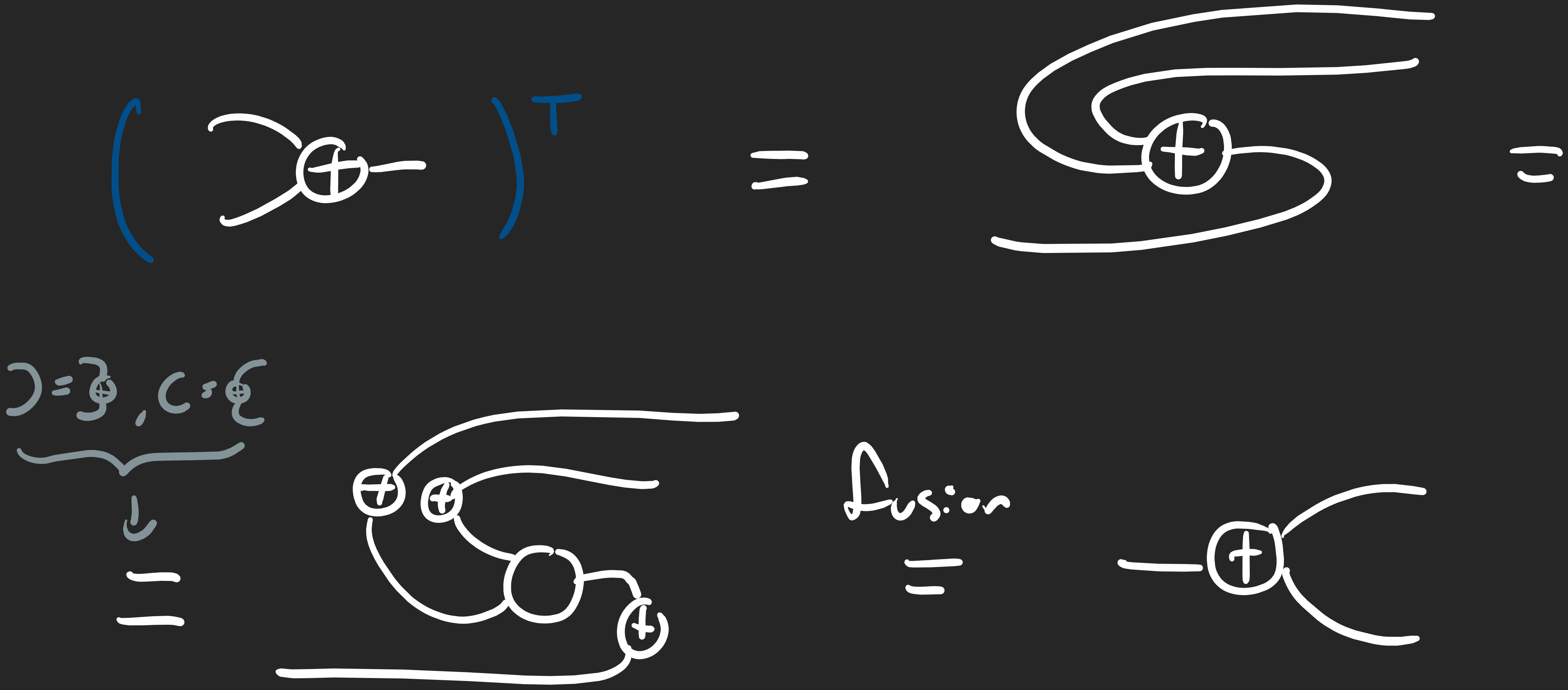
The Transpose



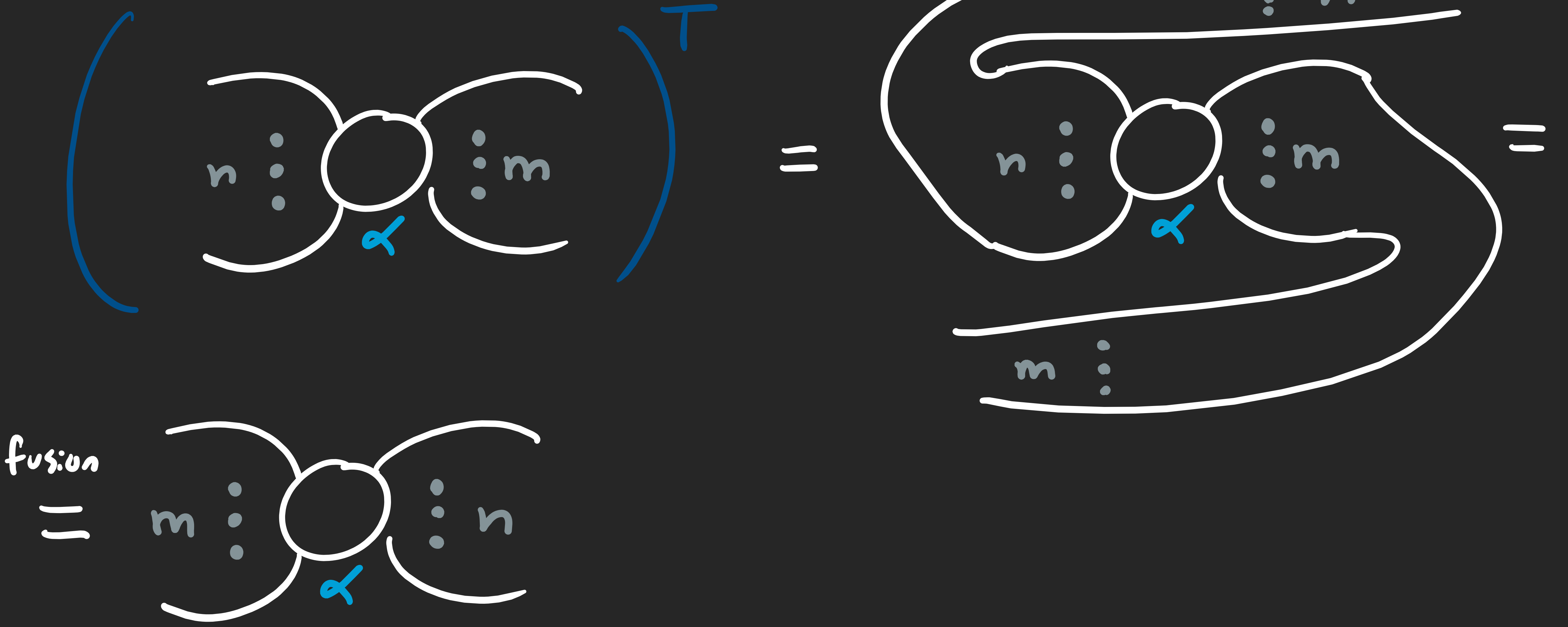
The Transpose



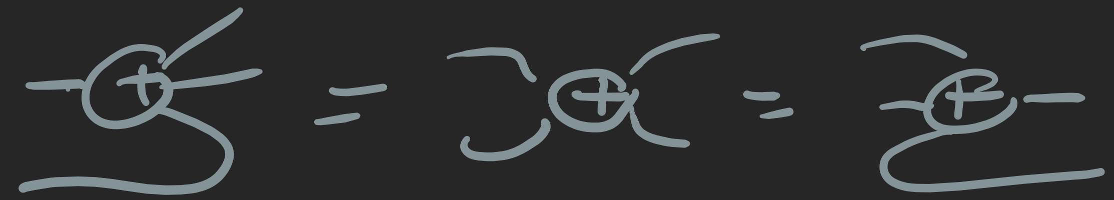
The Transpose



The Transpose



The Transpose

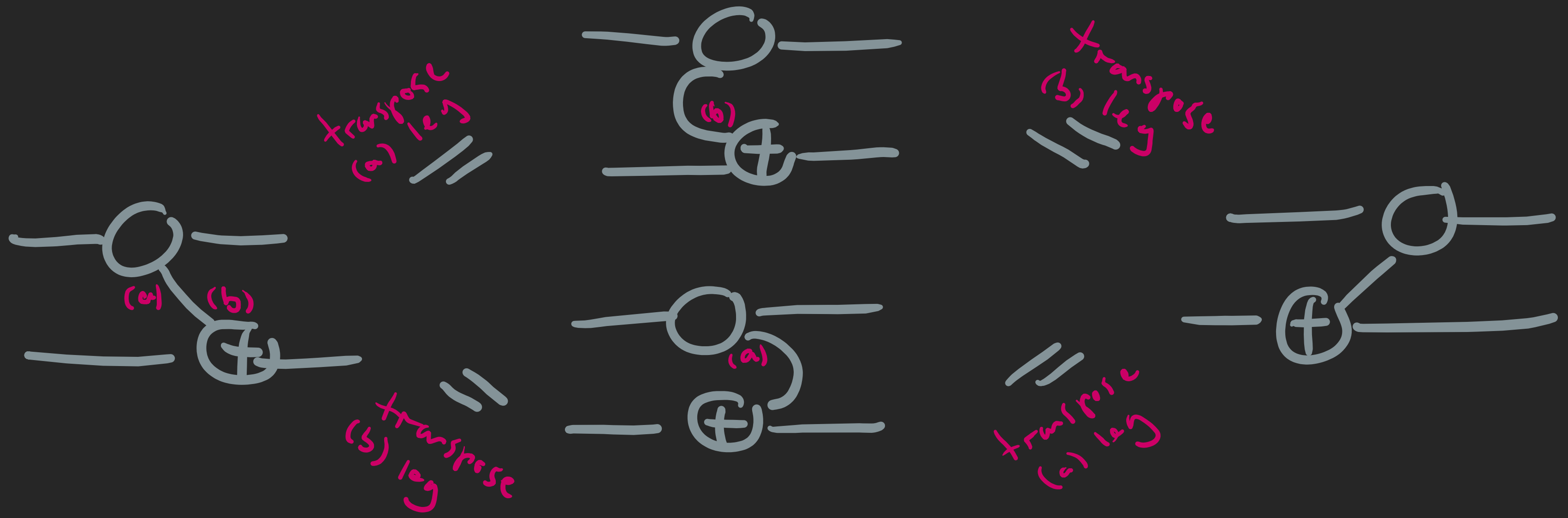
e.g. 

We can also perform a partial transpose:

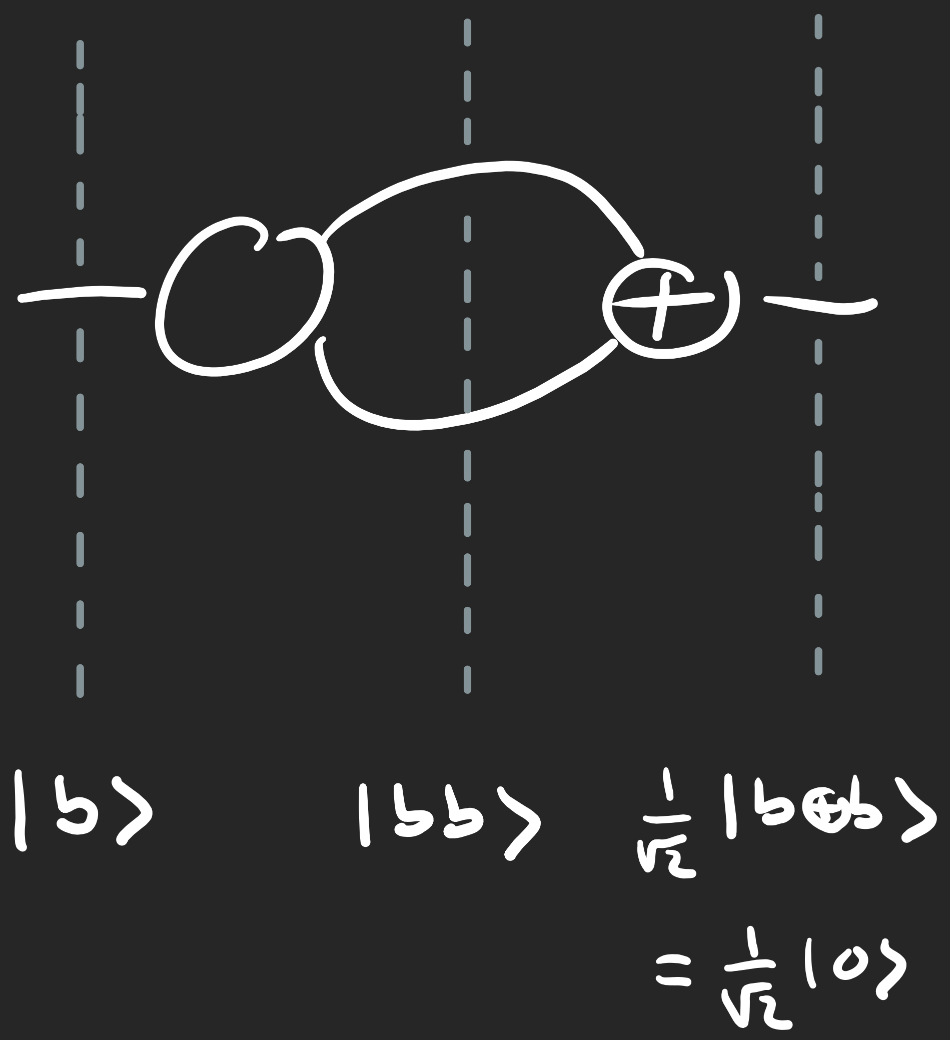
(Good luck describing this in braket notation...)

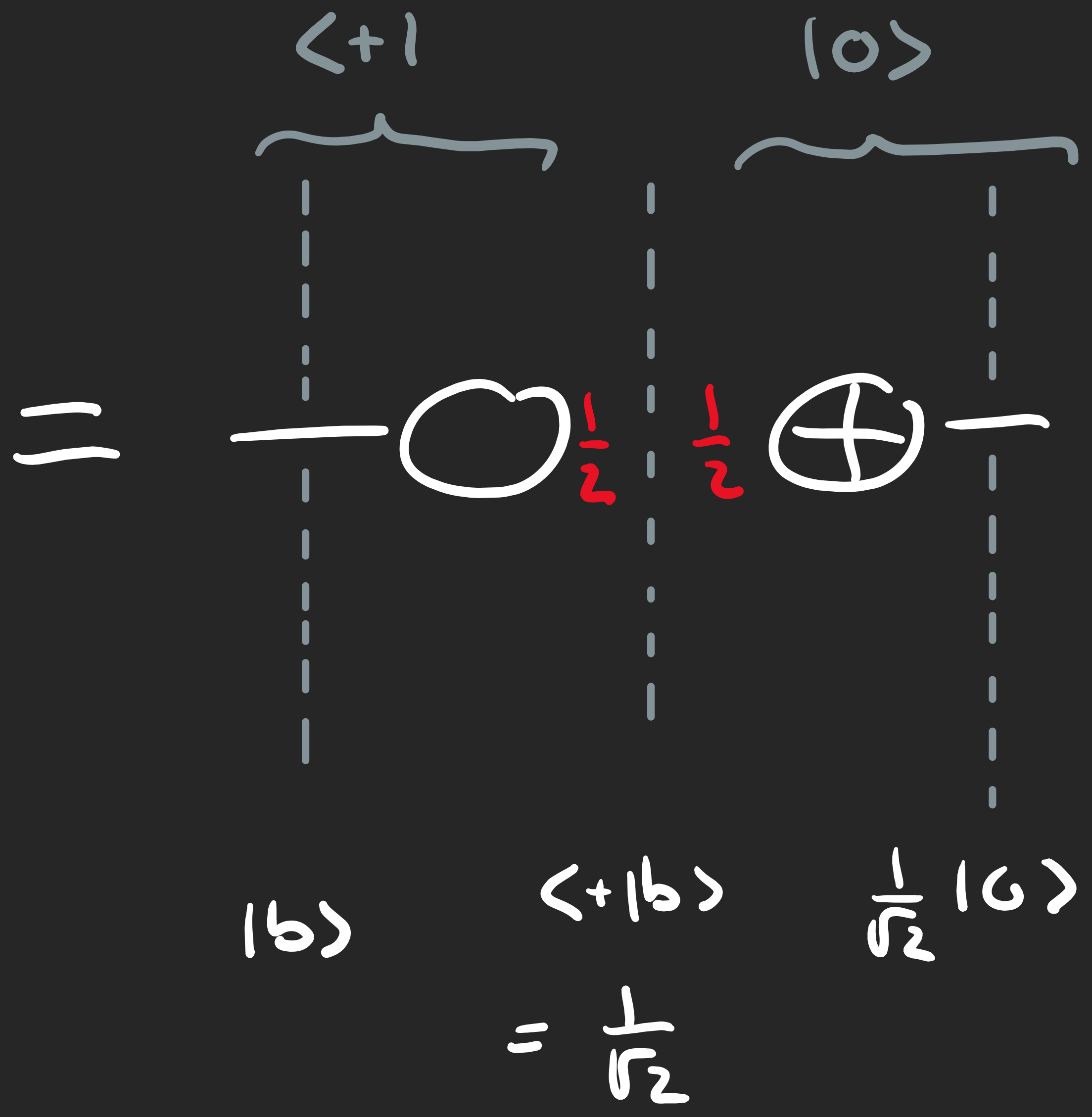
The CNOT Gate

Proof

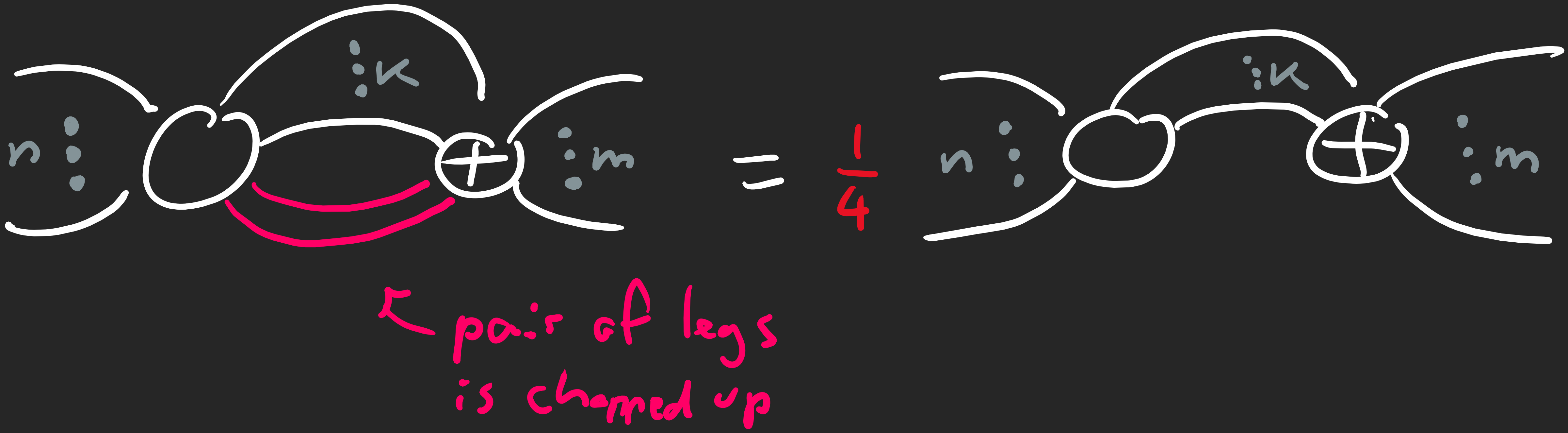


The Hopf Law

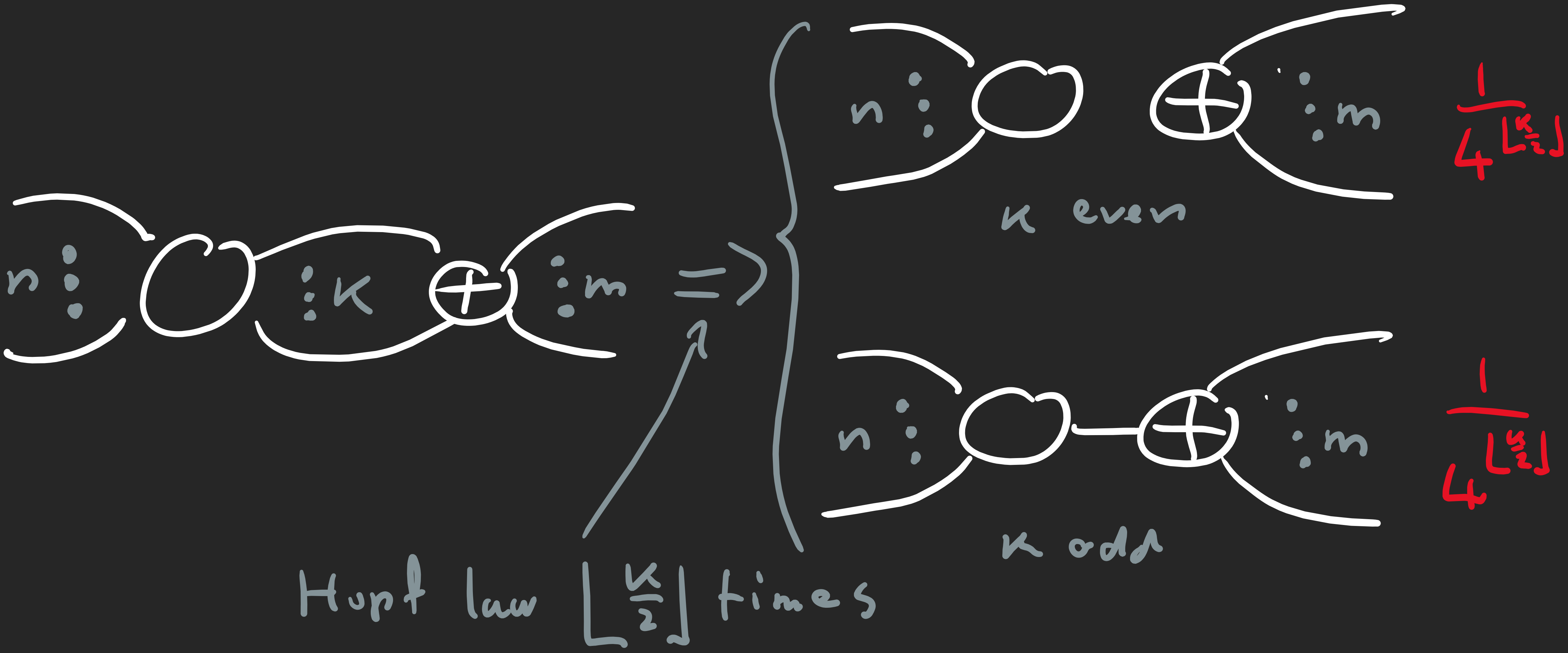




The Hopf Law



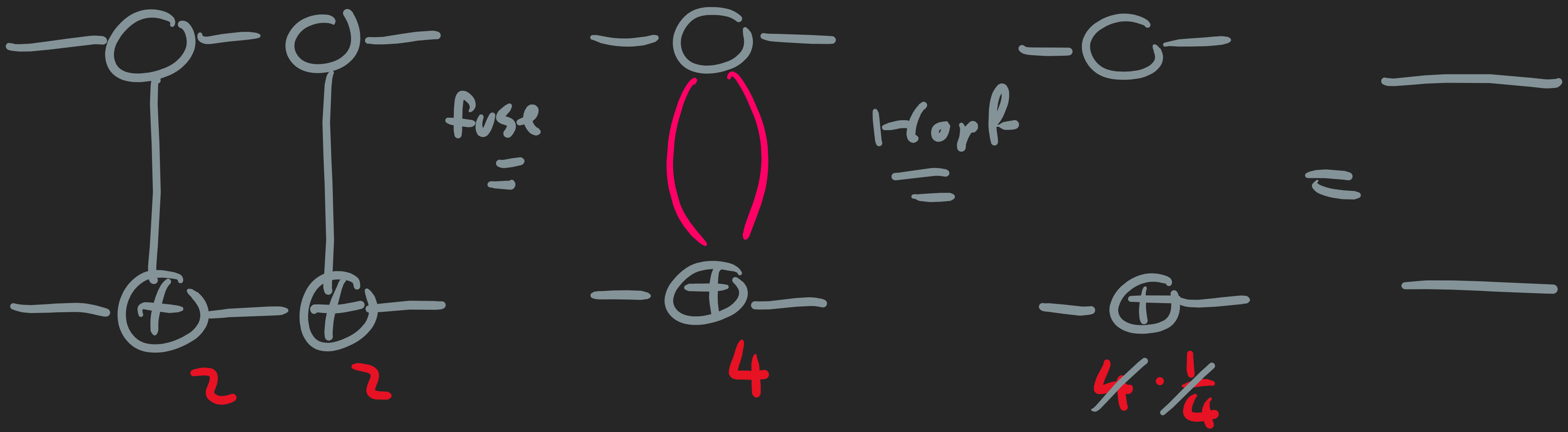
The Hopf Law



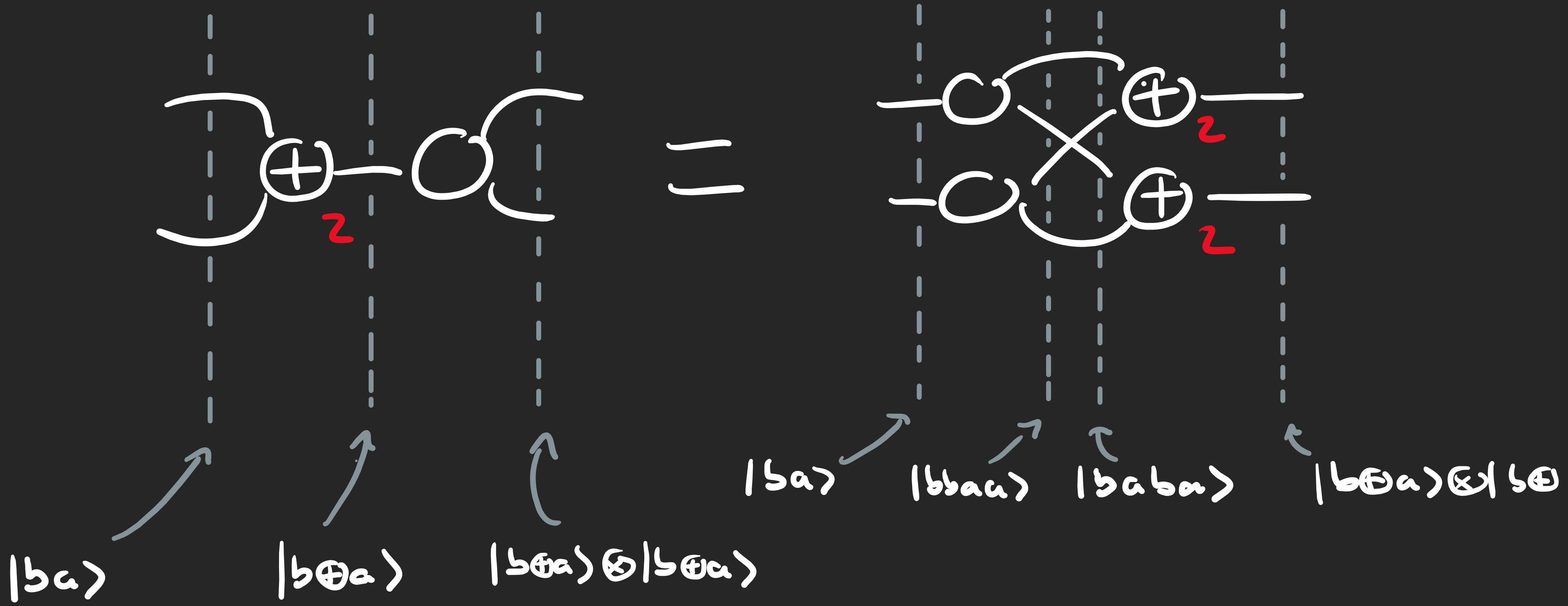
n
K
m
Hopf law ⌊K/2⌋ times
K even
K odd
$\frac{1}{4^{\lfloor K/2 \rfloor}}$
$\frac{1}{4^{\lfloor K/2 \rfloor}}$

The CNOT Gate

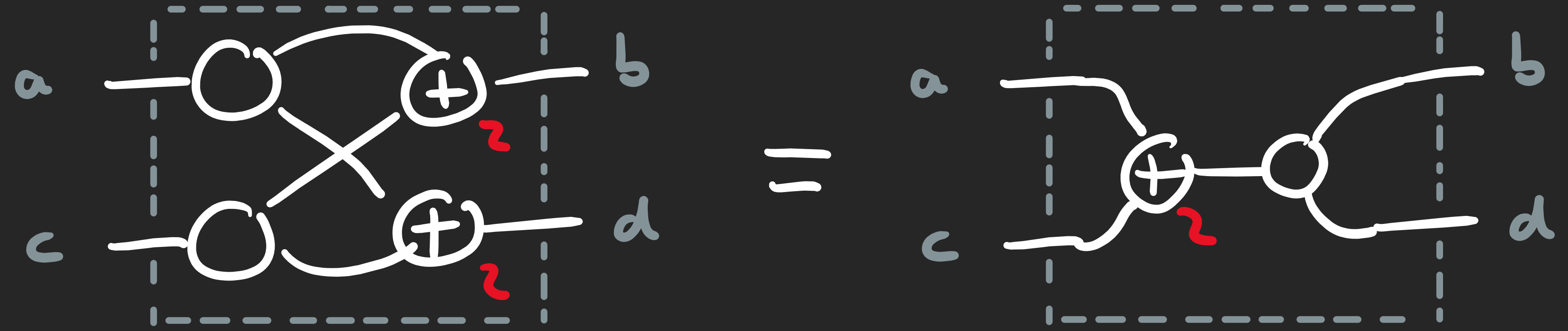
Proof:

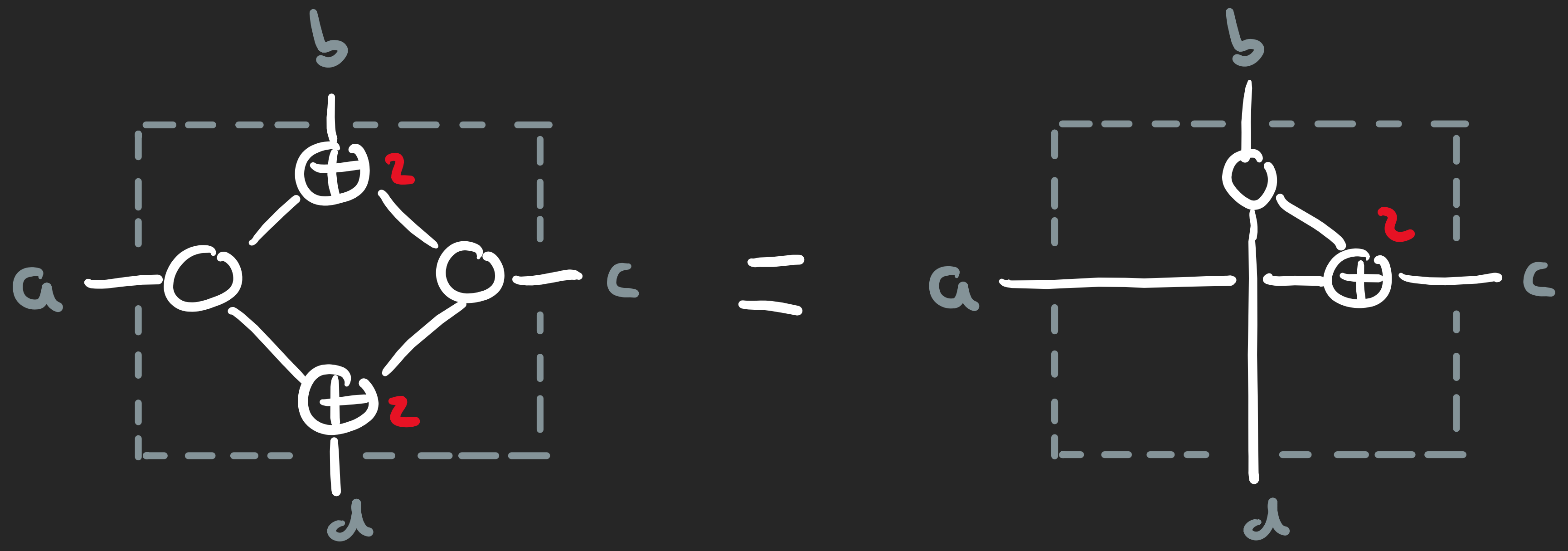


The Bialgebra Law

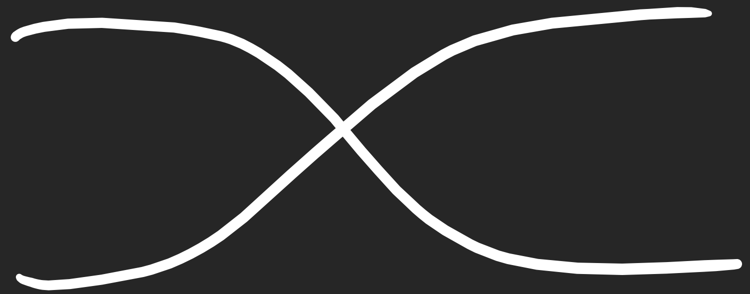


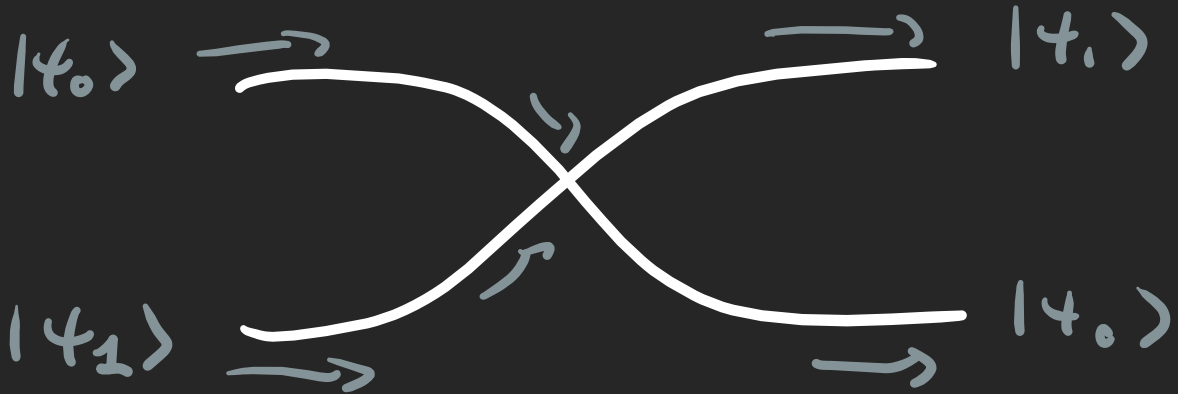
The Bialgebra Law



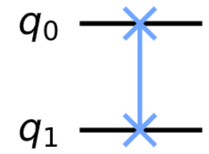


The SWAP Gate



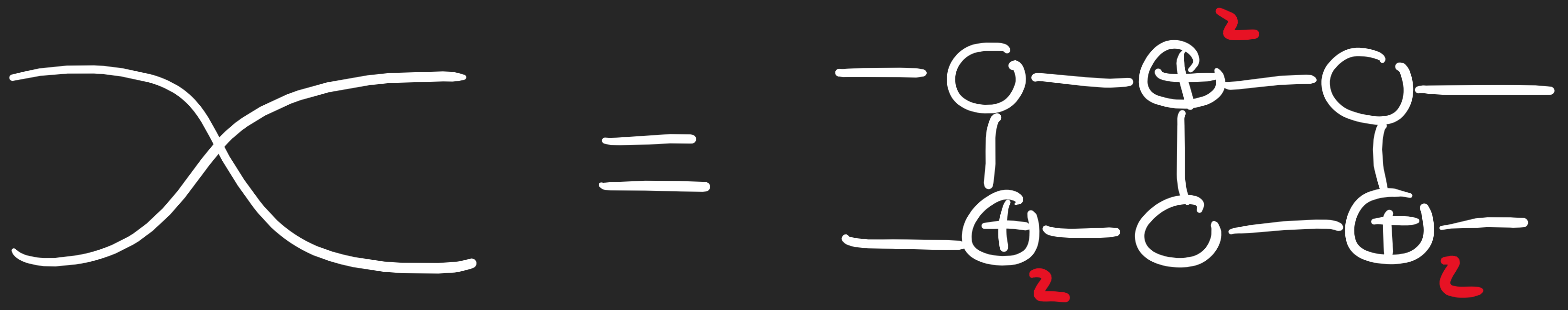



```
circ = QuantumCircuit(2)
circ.swap(0,1)
circ.draw("mpl")
```

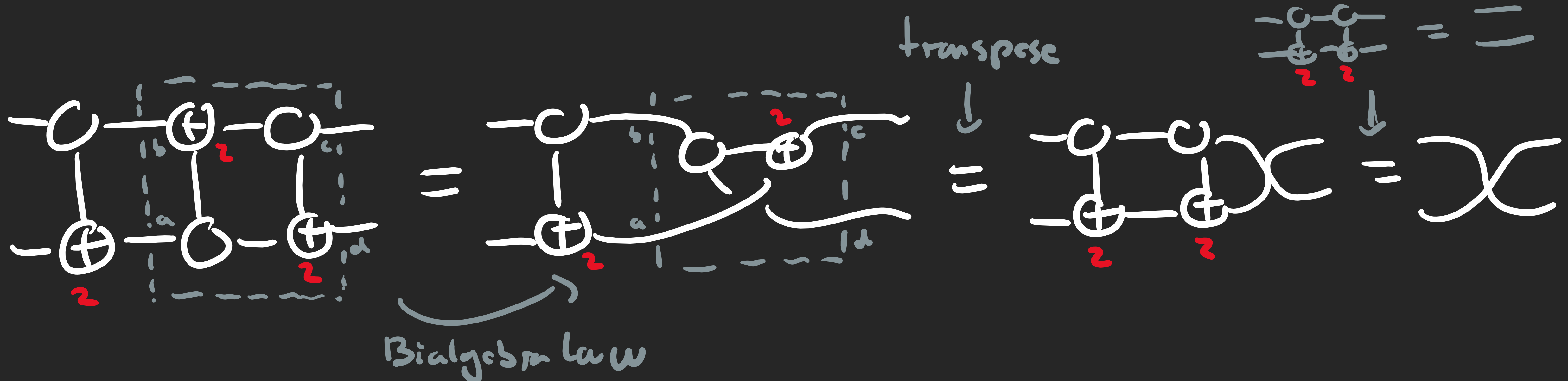


The SWAP Gate

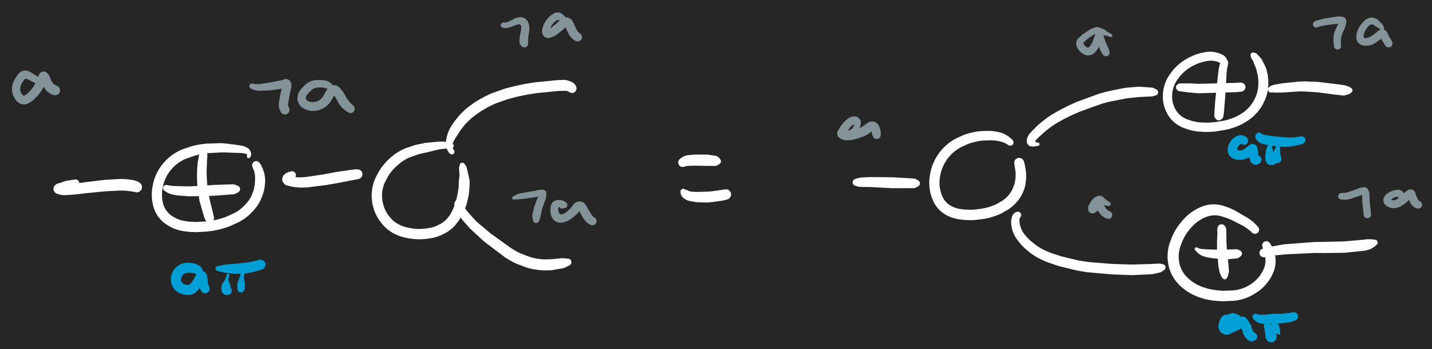
This is how SWAP is actually implemented in QC!



Proof:

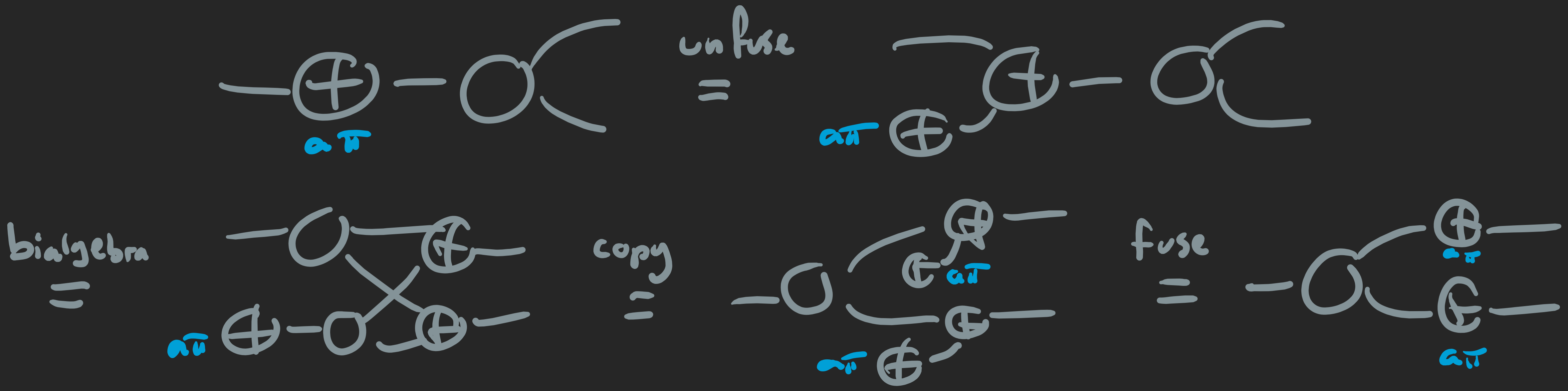


The π-commutation Rule



$a \in \{0,1\}$

Proof:

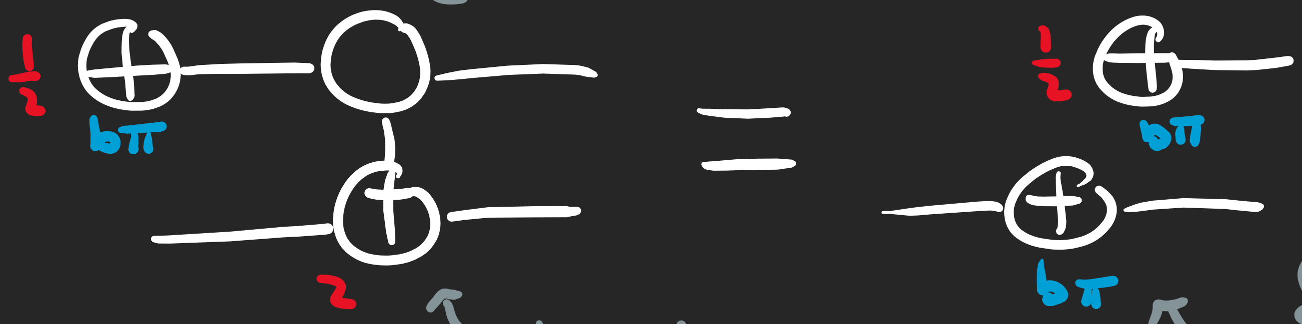


The ~~CNOT~~ Gate

CX

Z side

$\begin{cases} |0\rangle & \text{if } b=0 \\ |1\rangle & \text{if } b=1 \end{cases}$



Z - controlled X gate on X side

X side

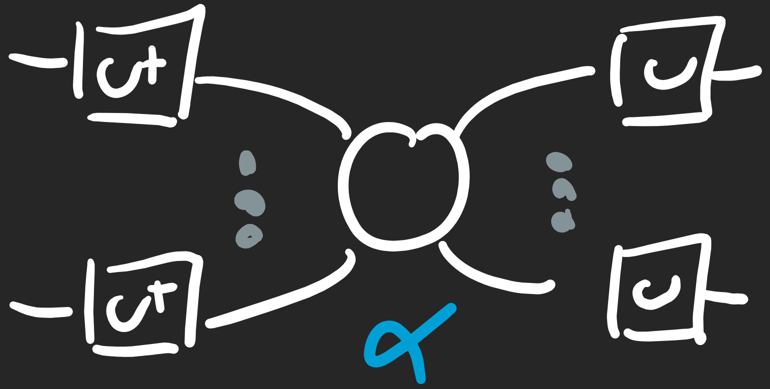
$\begin{cases} id & \text{if } b=0 \\ X & \text{if } b=1 \end{cases}$

$\begin{cases} id & \text{if } b=0 \\ Z & \text{if } b=1 \end{cases}$

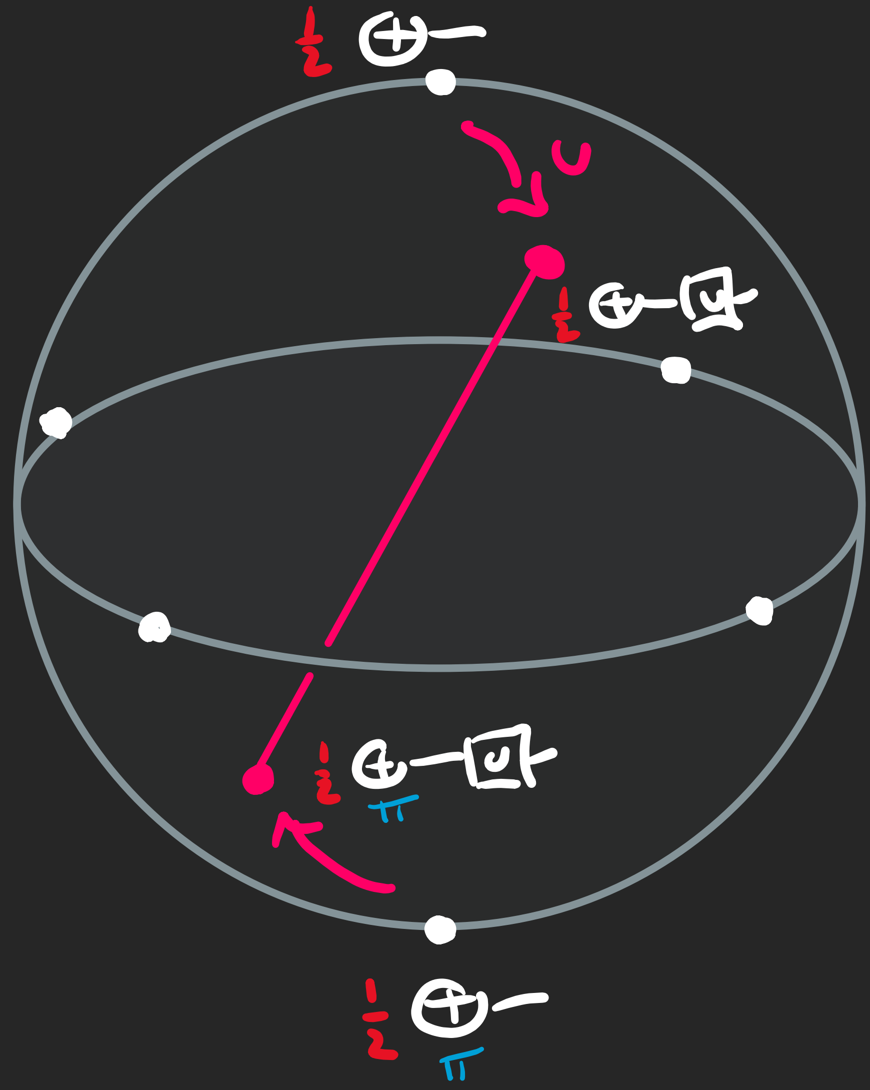
X - controlled Z gate on Z side

$\begin{cases} |+\rangle & \text{if } b=0 \\ |-\rangle & \text{if } b=1 \end{cases}$

More Spiders



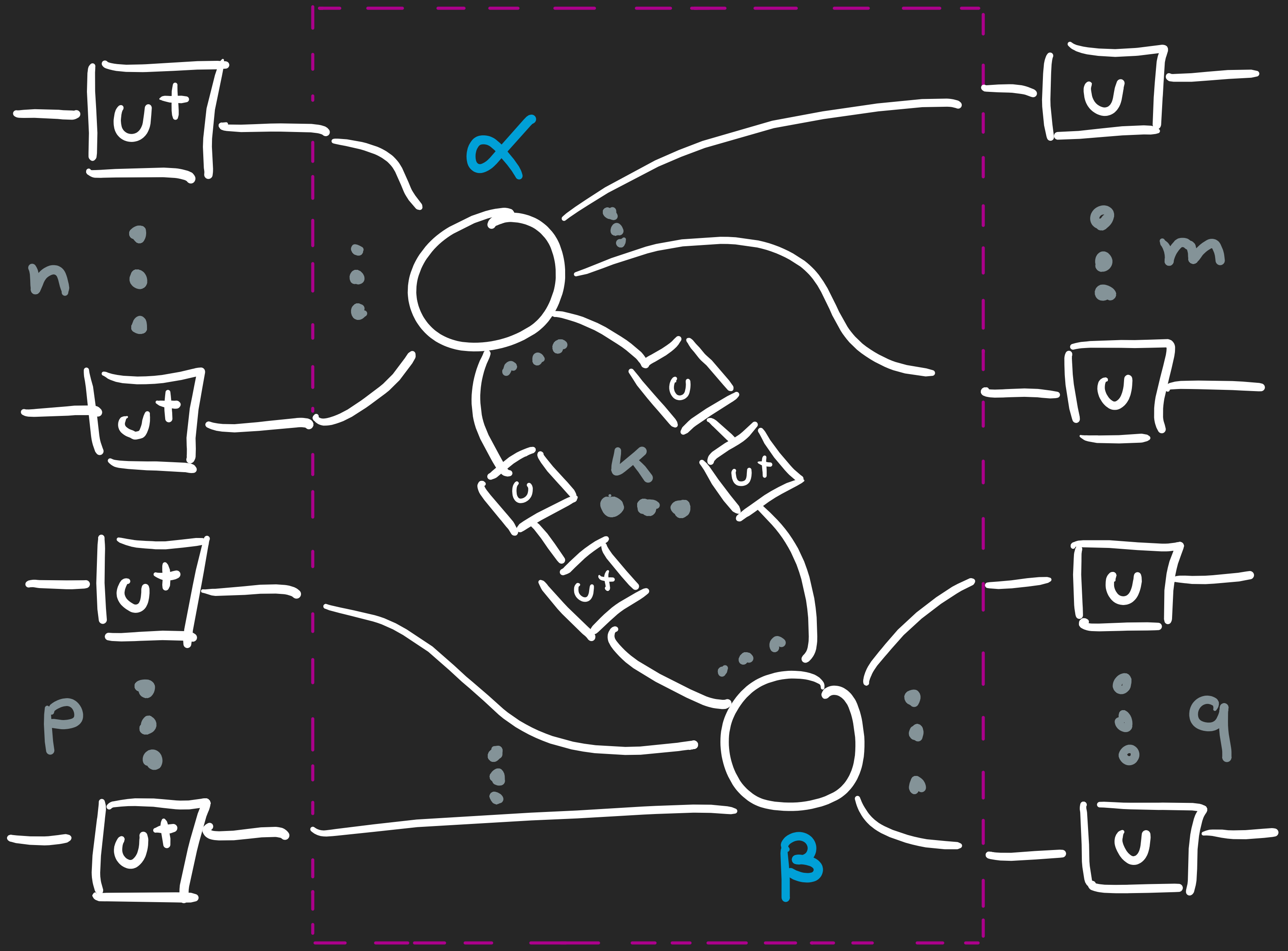
Basis



Spider for basis obtained by rotating the Z basis using rotation U.

$(1.\ -\boxed{U}-\boxed{U^\dagger}- \; = \; ——)$

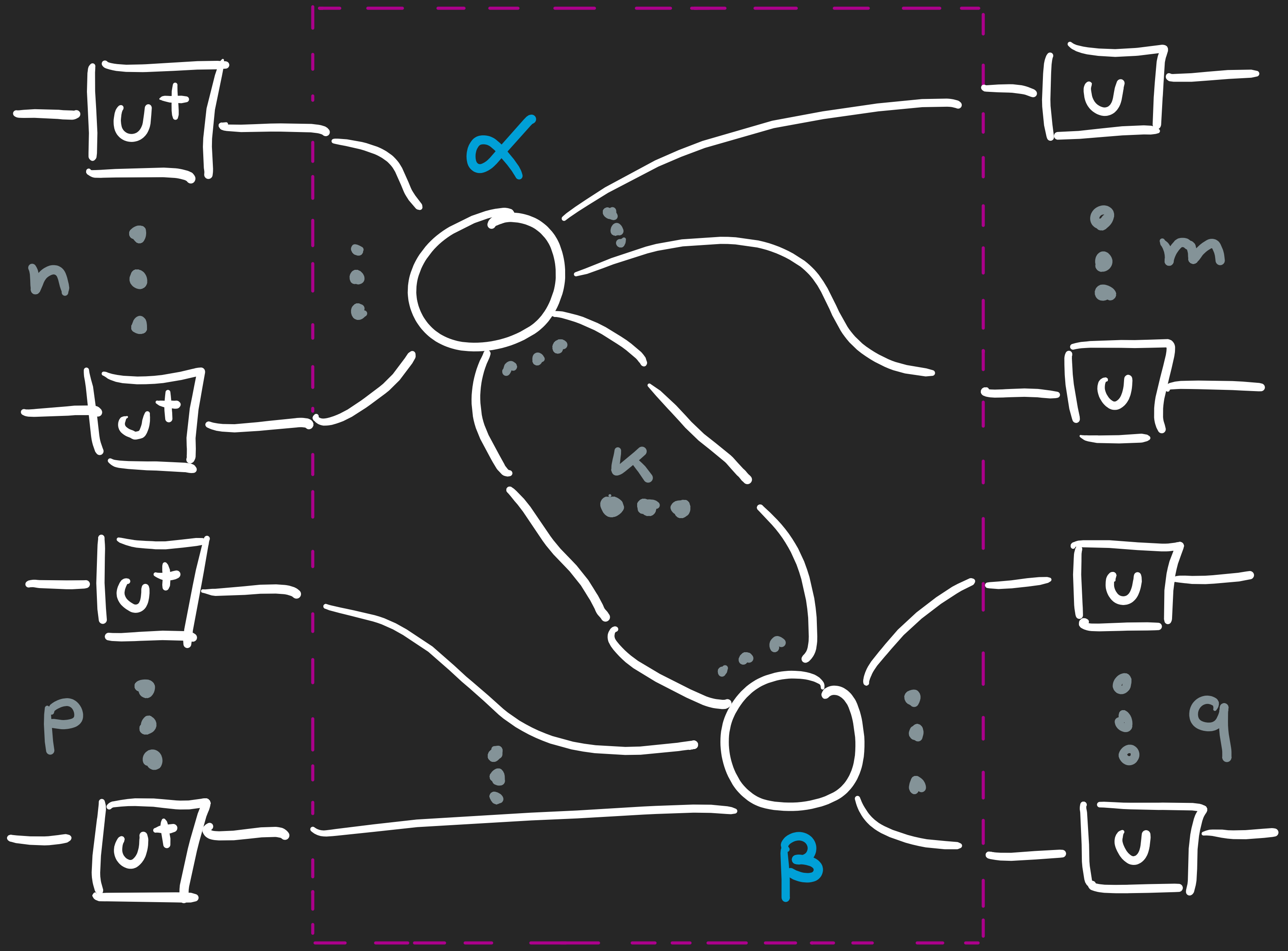
More Spiders



More Spiders

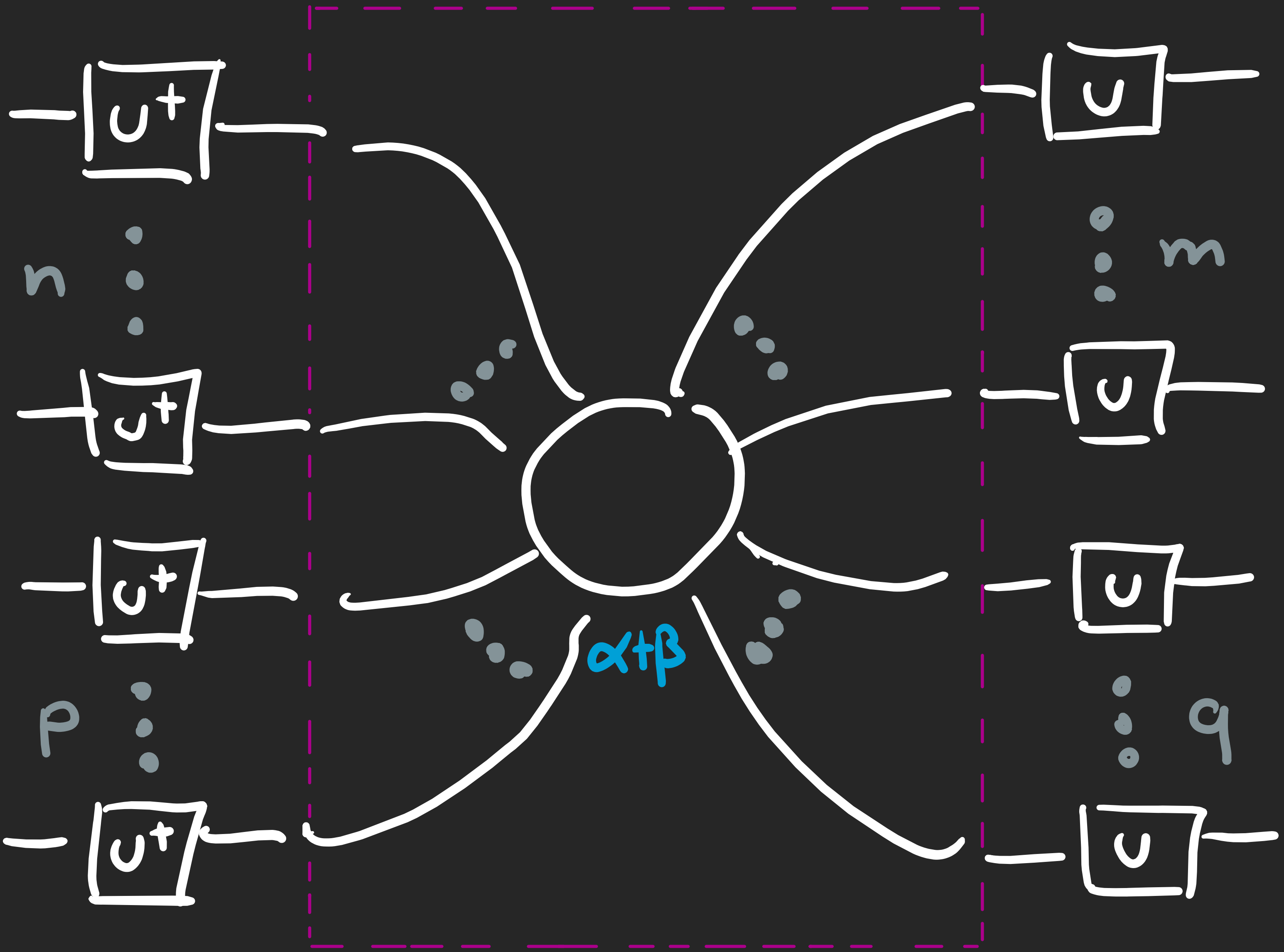
1. $-\boxed{U}-\boxed{U^\dagger}- \; = \; ——$

(2. spider fusion)



More Spiders

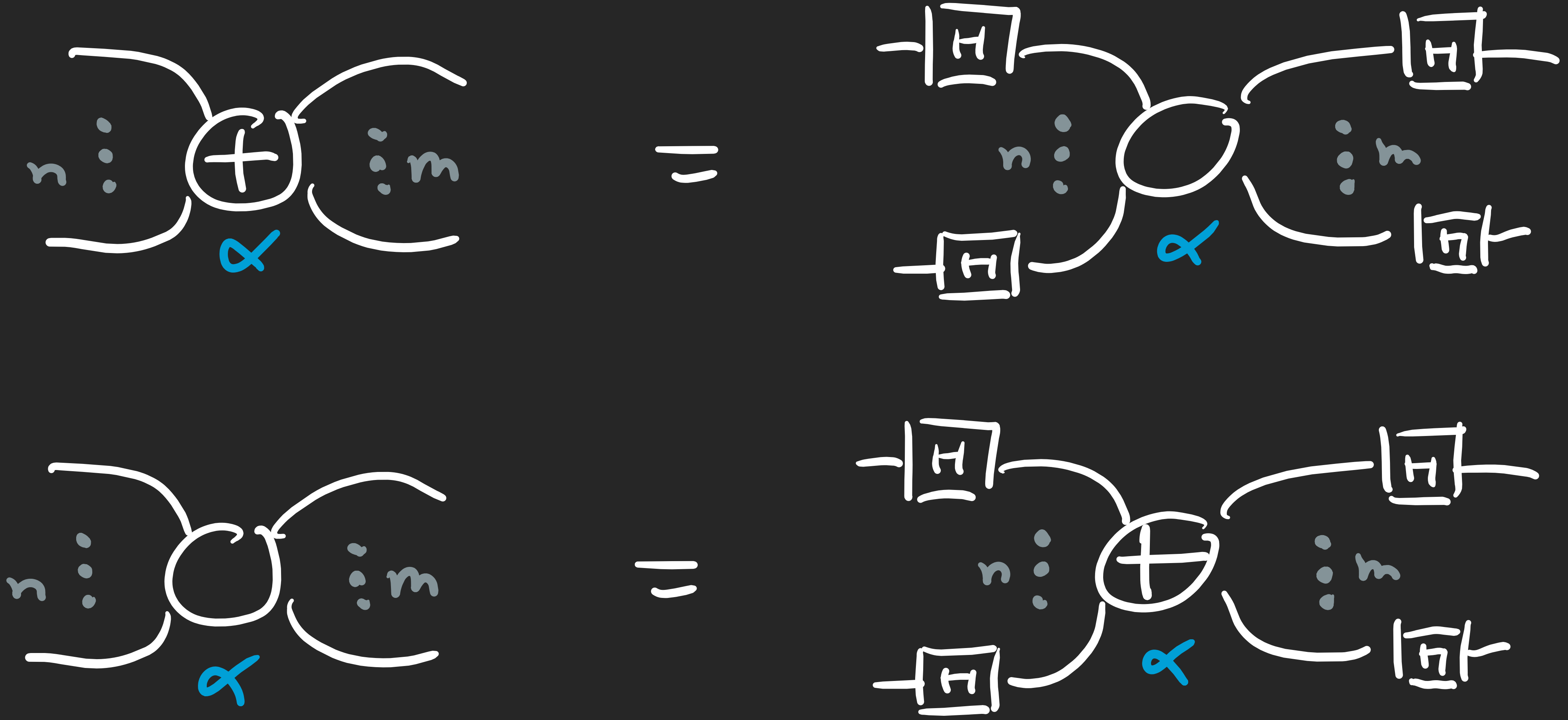
1. $-\boxed{U}-\boxed{U^\dagger}- \;=\; ——$
2. spider fusion



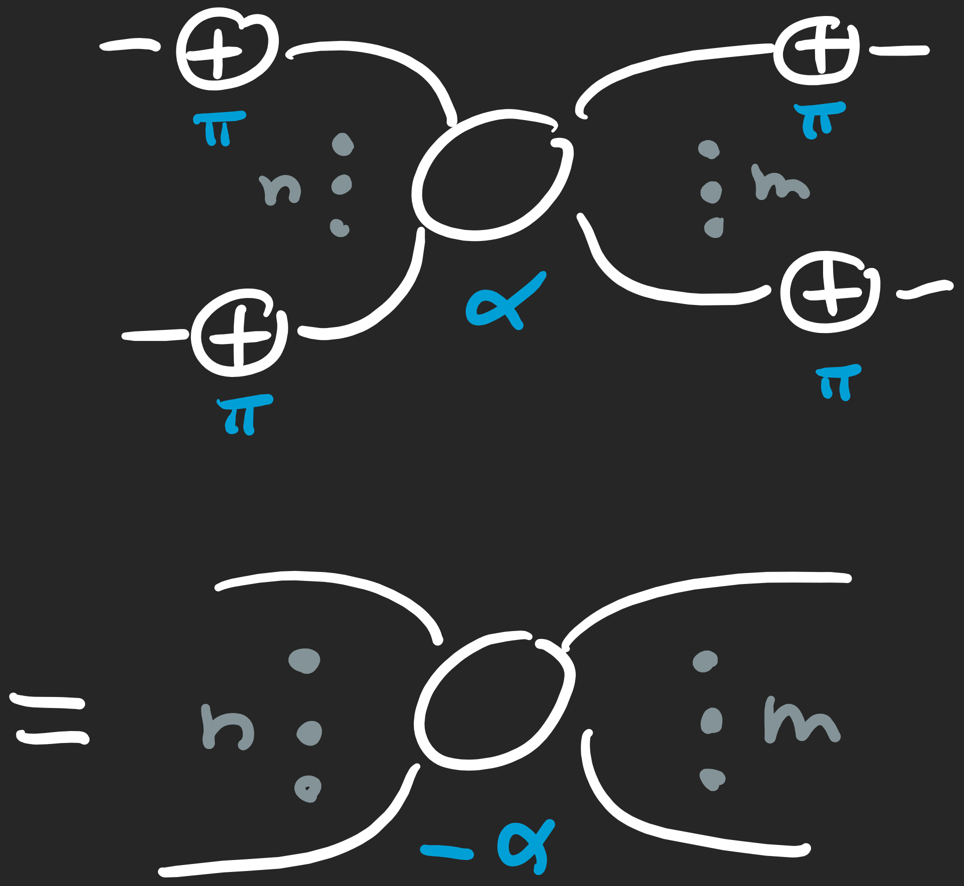
Spiders

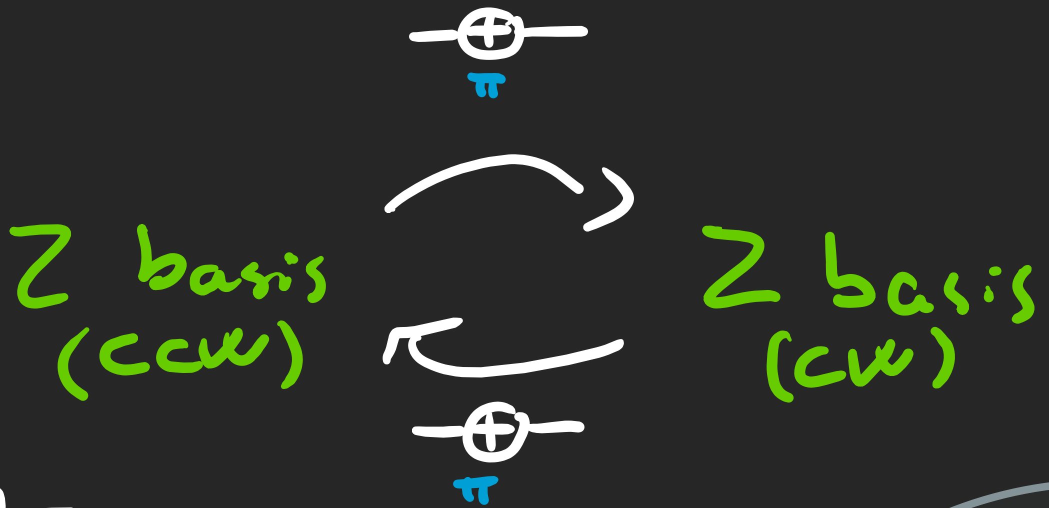
Z basis $\underset{H}{\overset{H}{\rightleftarrows}}$ X basis

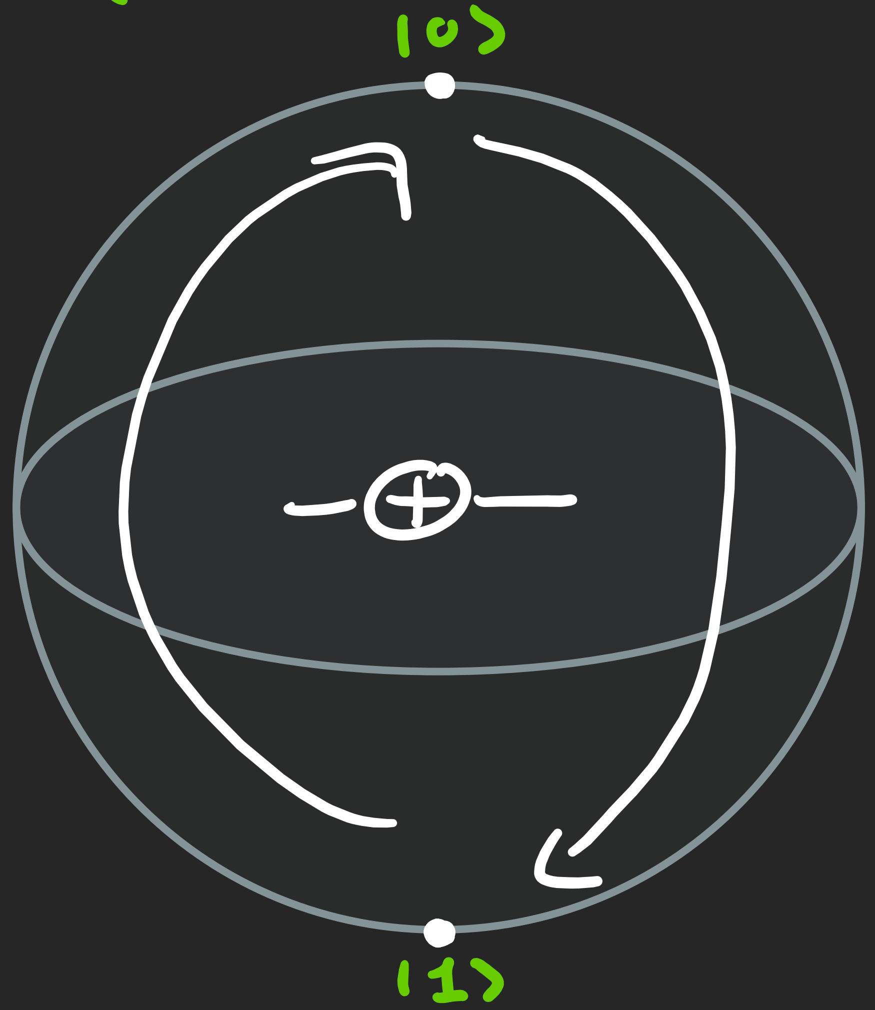
$-\boxed{H}- = -\boxed{H^{\dagger}}-$ (because $-\boxed{H}\boxed{H}- = -$)



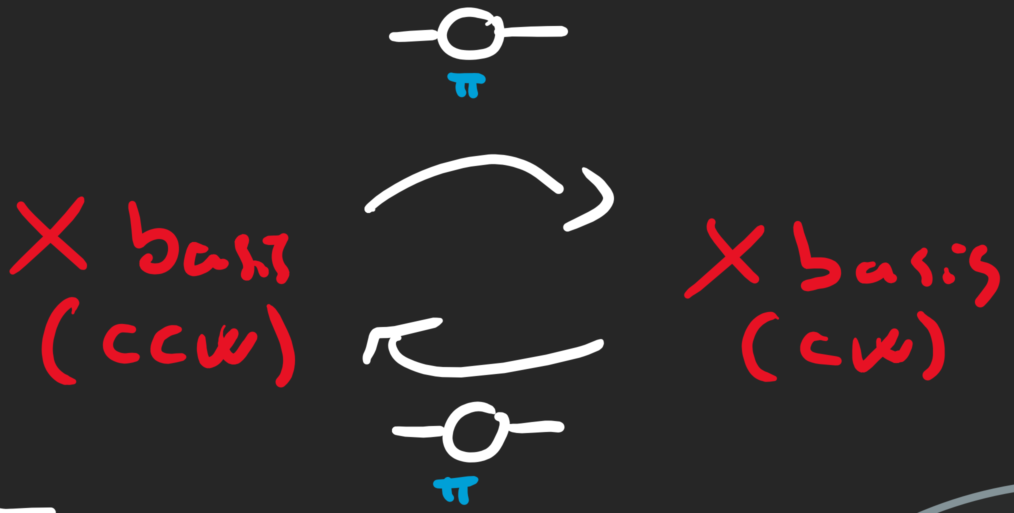
Spiders

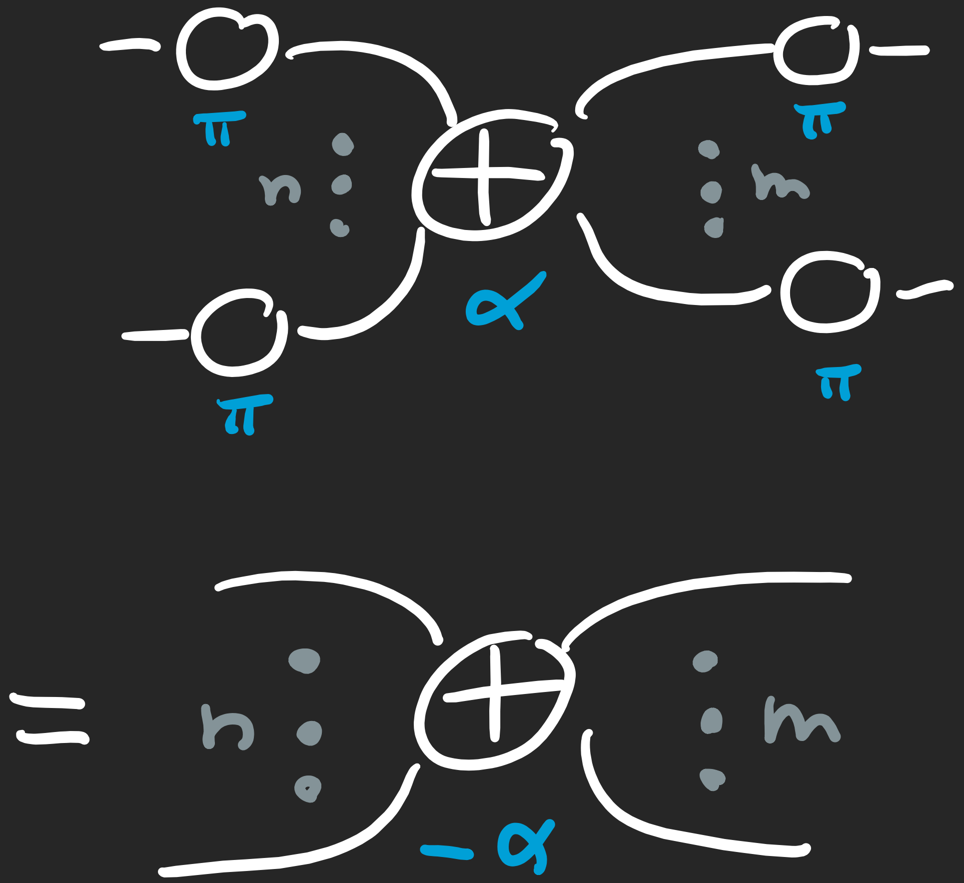


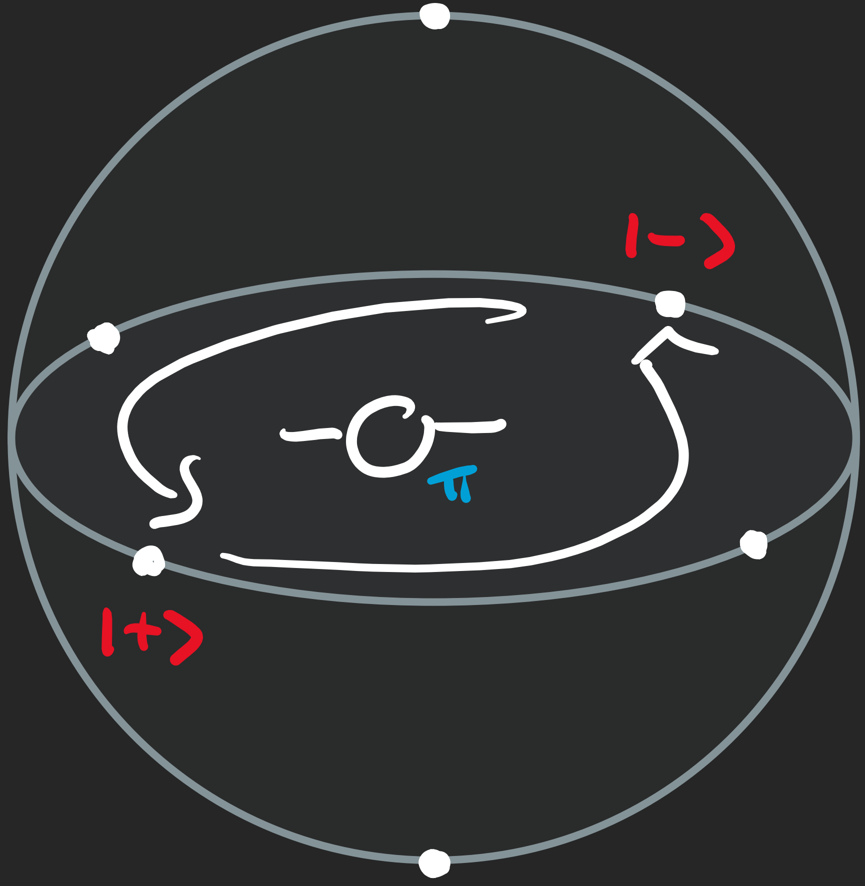




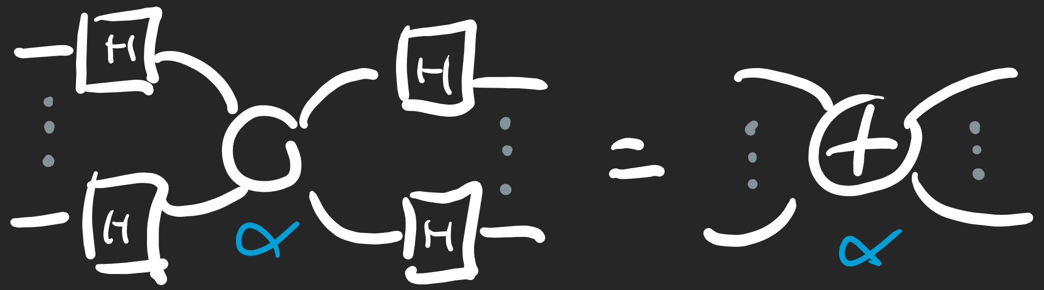
Spiders

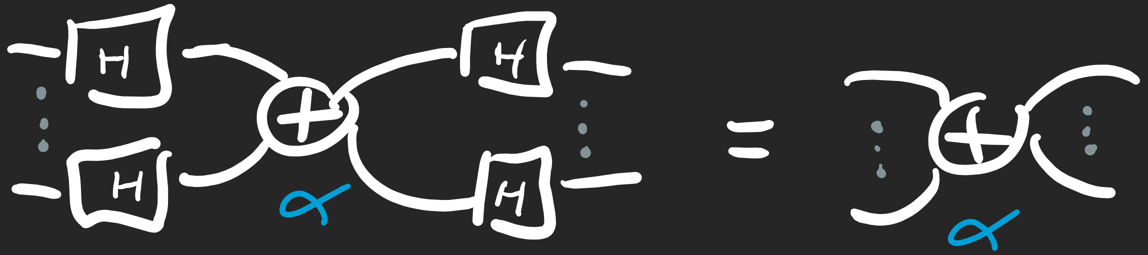


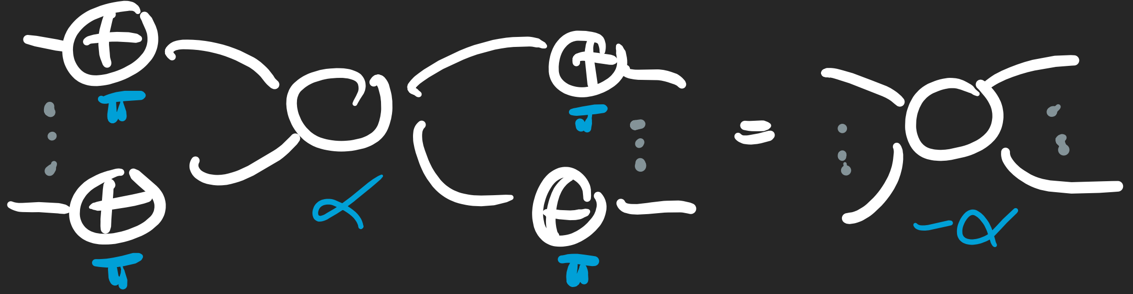


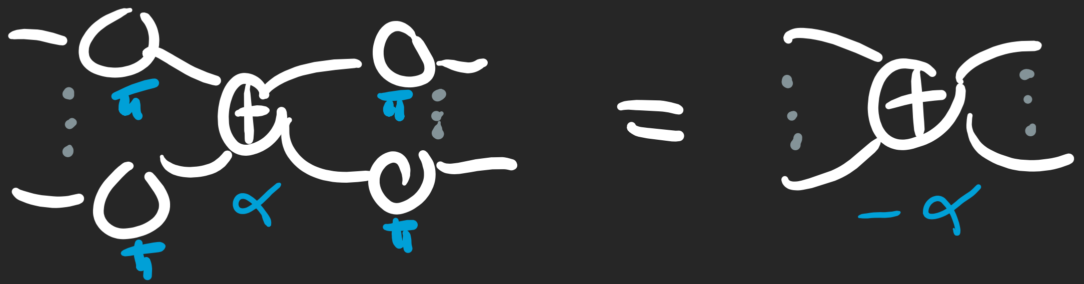


Conjugation rules

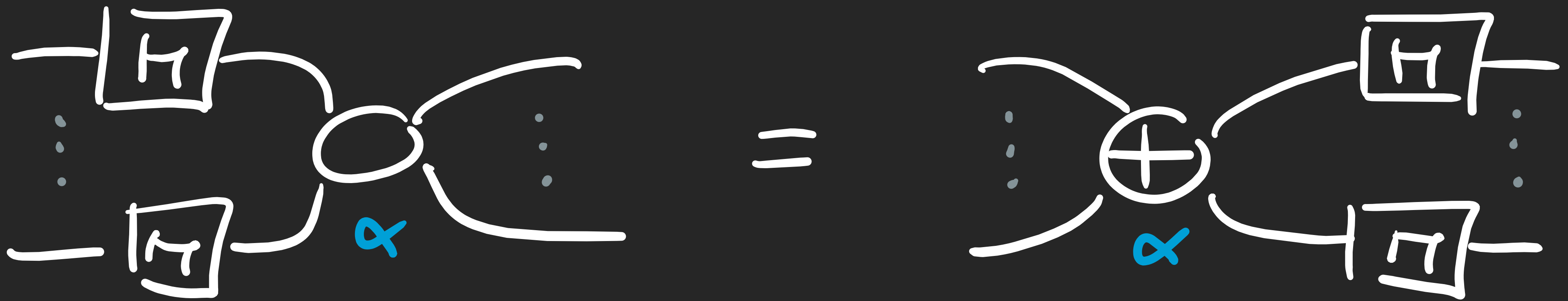




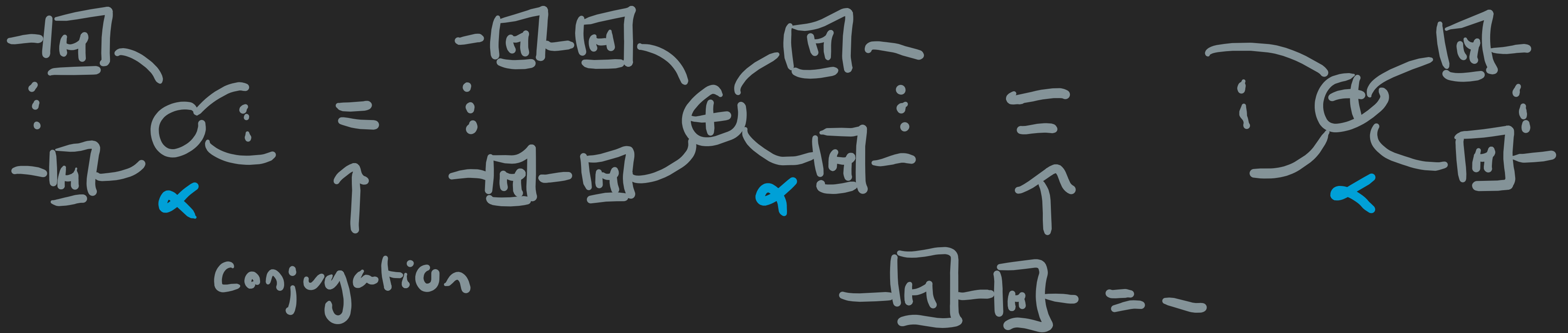




The H-commutation Rule

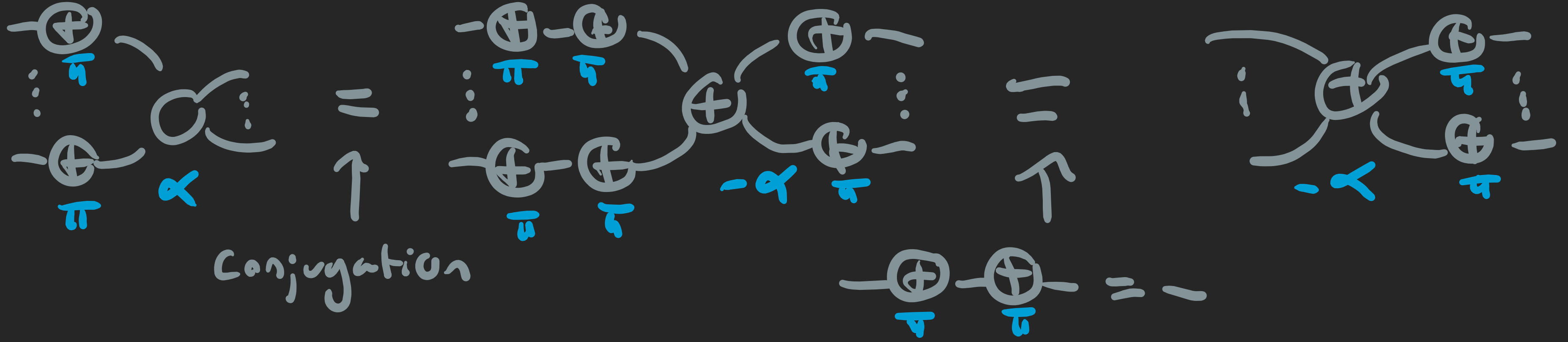


Proof:

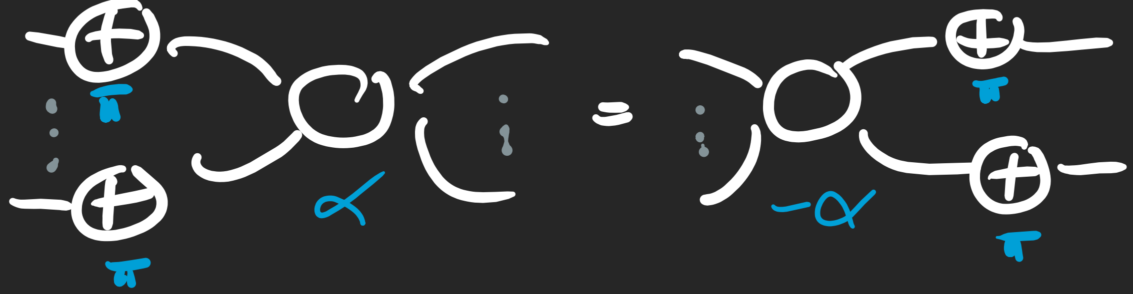


The π-commutation Rule

Proof:



Commutation rules

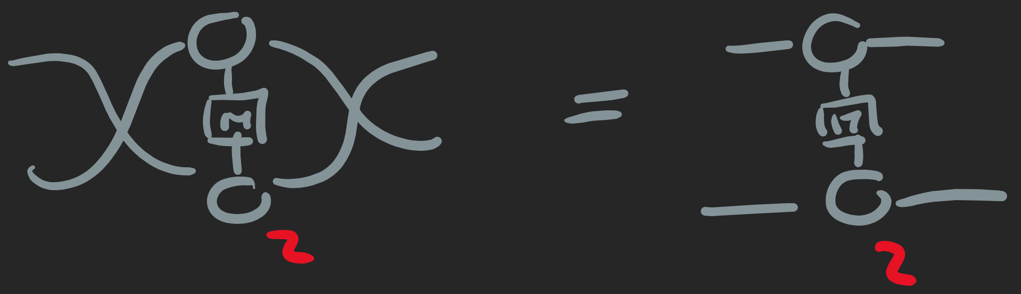


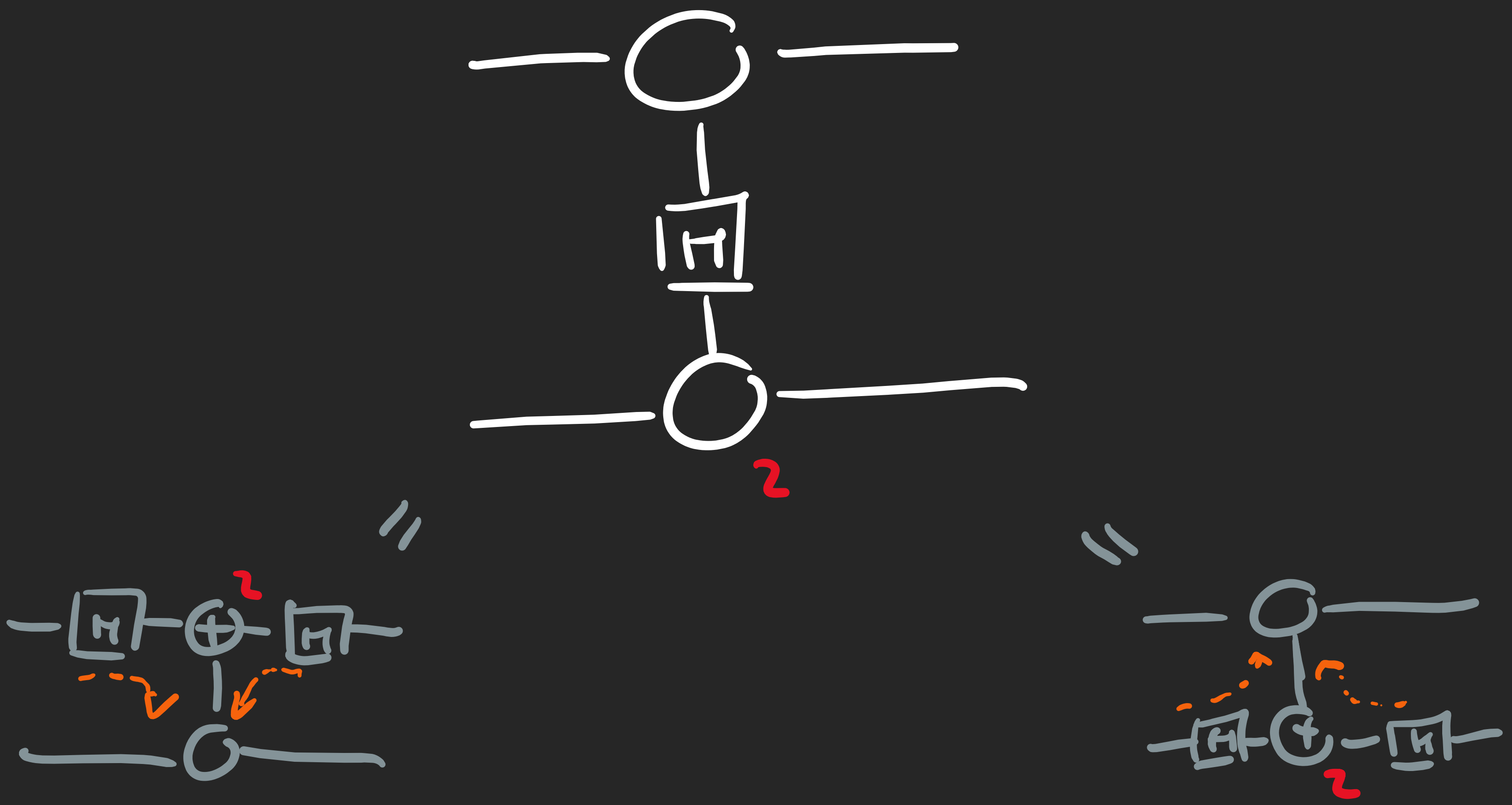
H-commutation on CX gate

Proof:

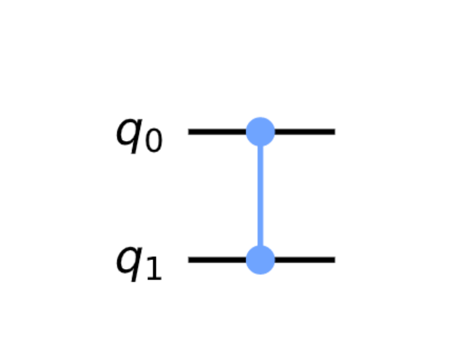
The CZ Gate

Acts identically on both qubits:



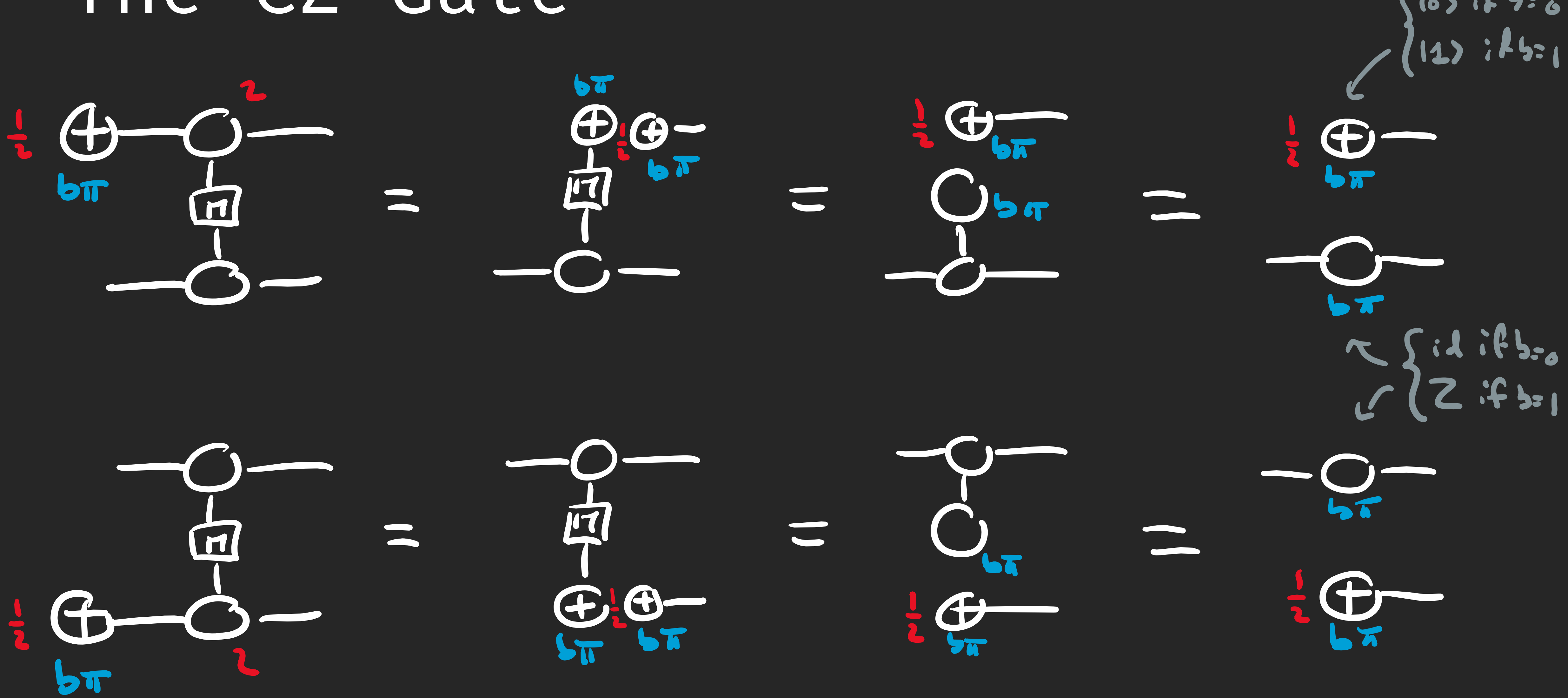


```
circ = QuantumCircuit(2)
circ.cz(0,1)
circ.draw("mpl")
```

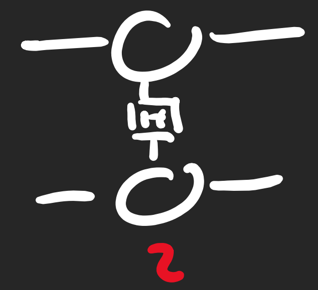


The CZ Gate

Z-controlled Z gate on both sides



The CNOT Gate

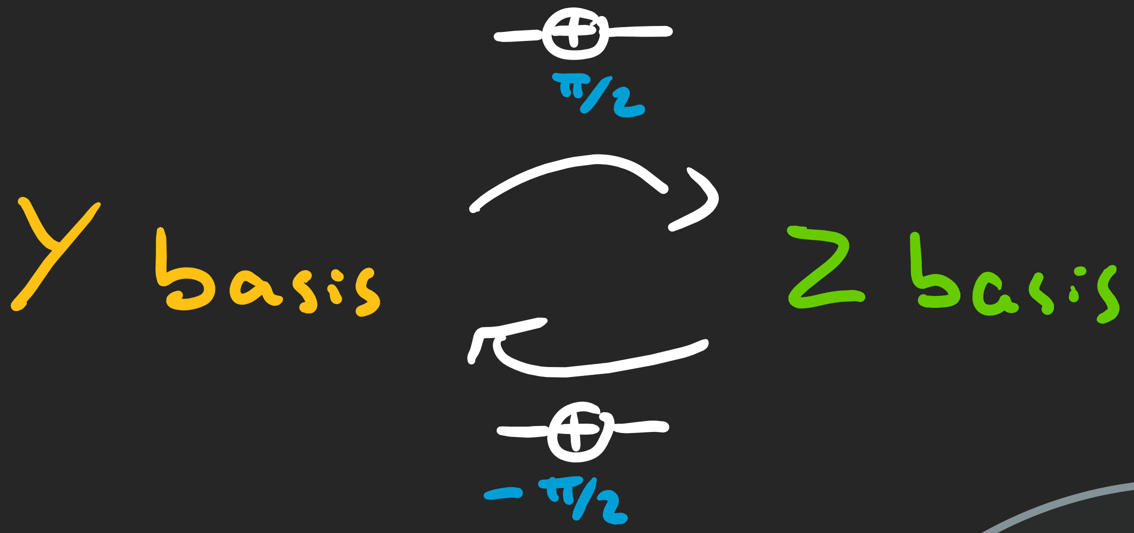


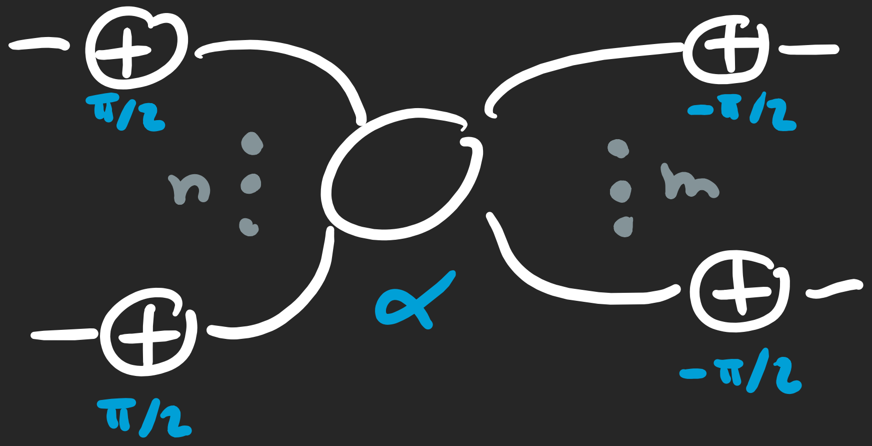
$$\longleftrightarrow$$

$$\overset{q_1 q_0}{|00\rangle}\overset{q_1 q_0}{\langle 00|} + |01\rangle\langle 01| + |10\rangle\langle 10| - |11\rangle\langle 11| = \begin{array}{cccc} |00\rangle \mapsto |00\rangle & |01\rangle \mapsto |01\rangle & |10\rangle \mapsto |10\rangle & |11\rangle \mapsto -|11\rangle \\ \end{array}$$

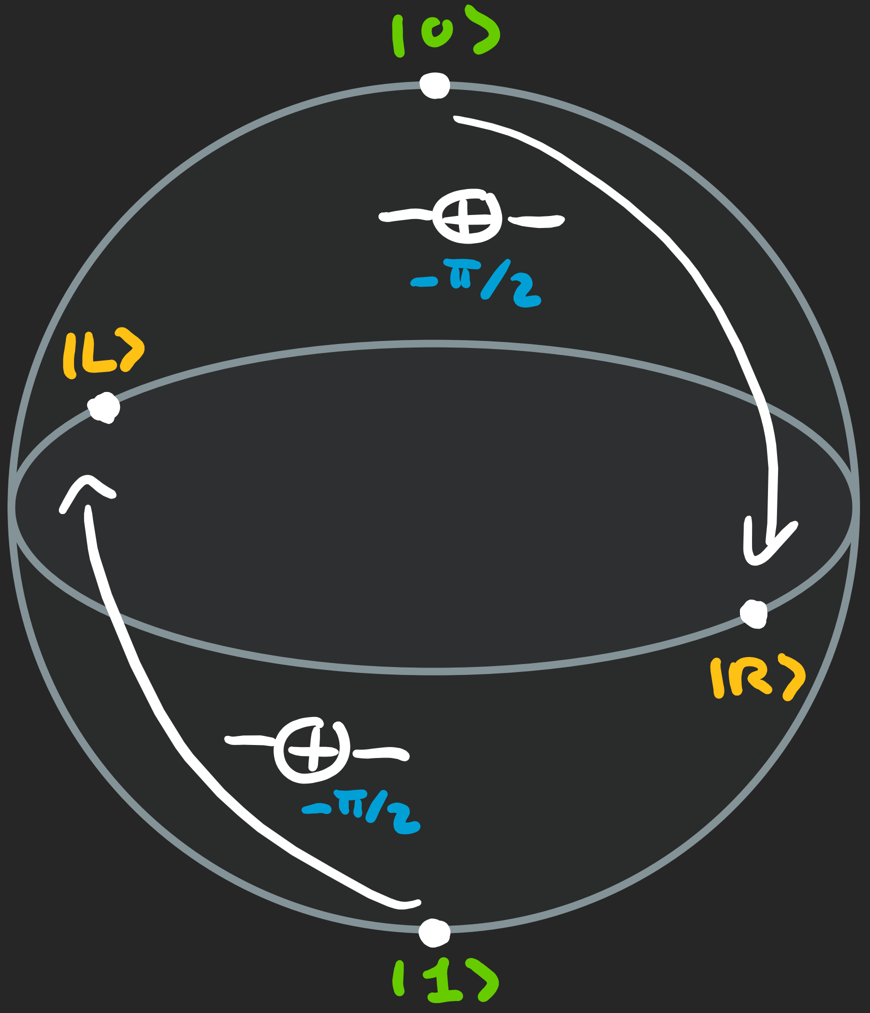
$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Spiders

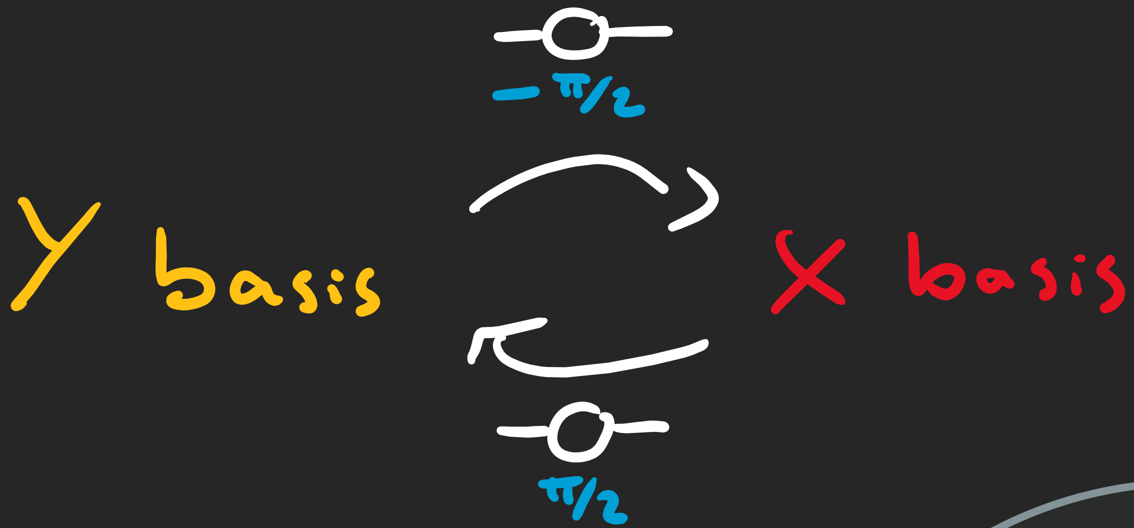


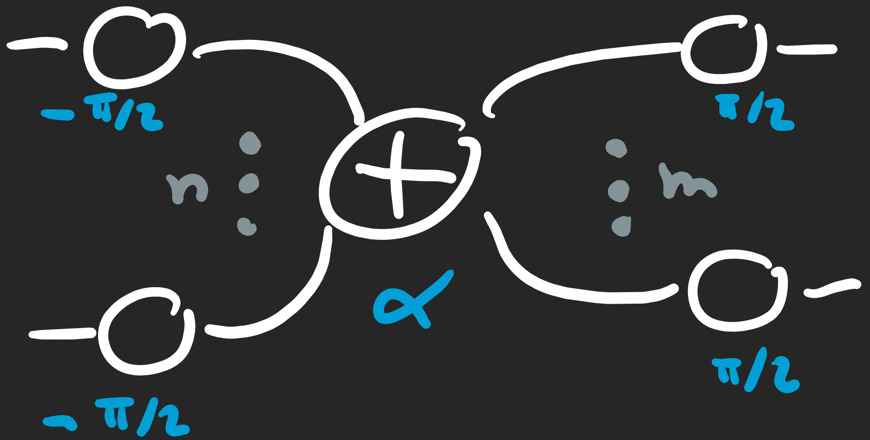


Y basis spider

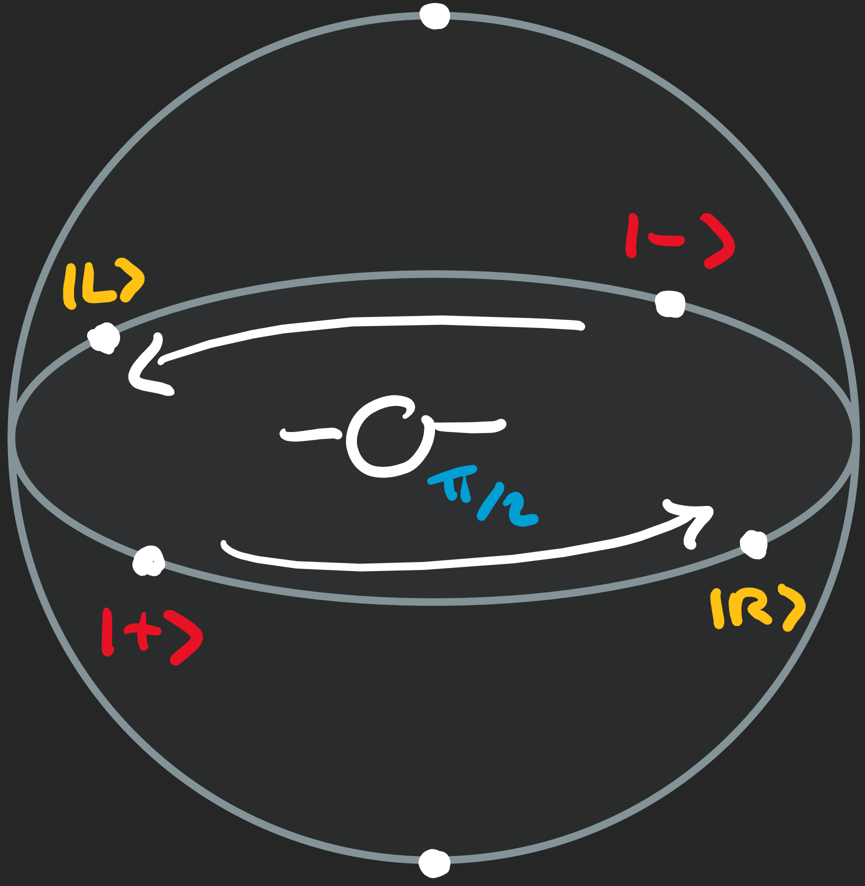


Spiders





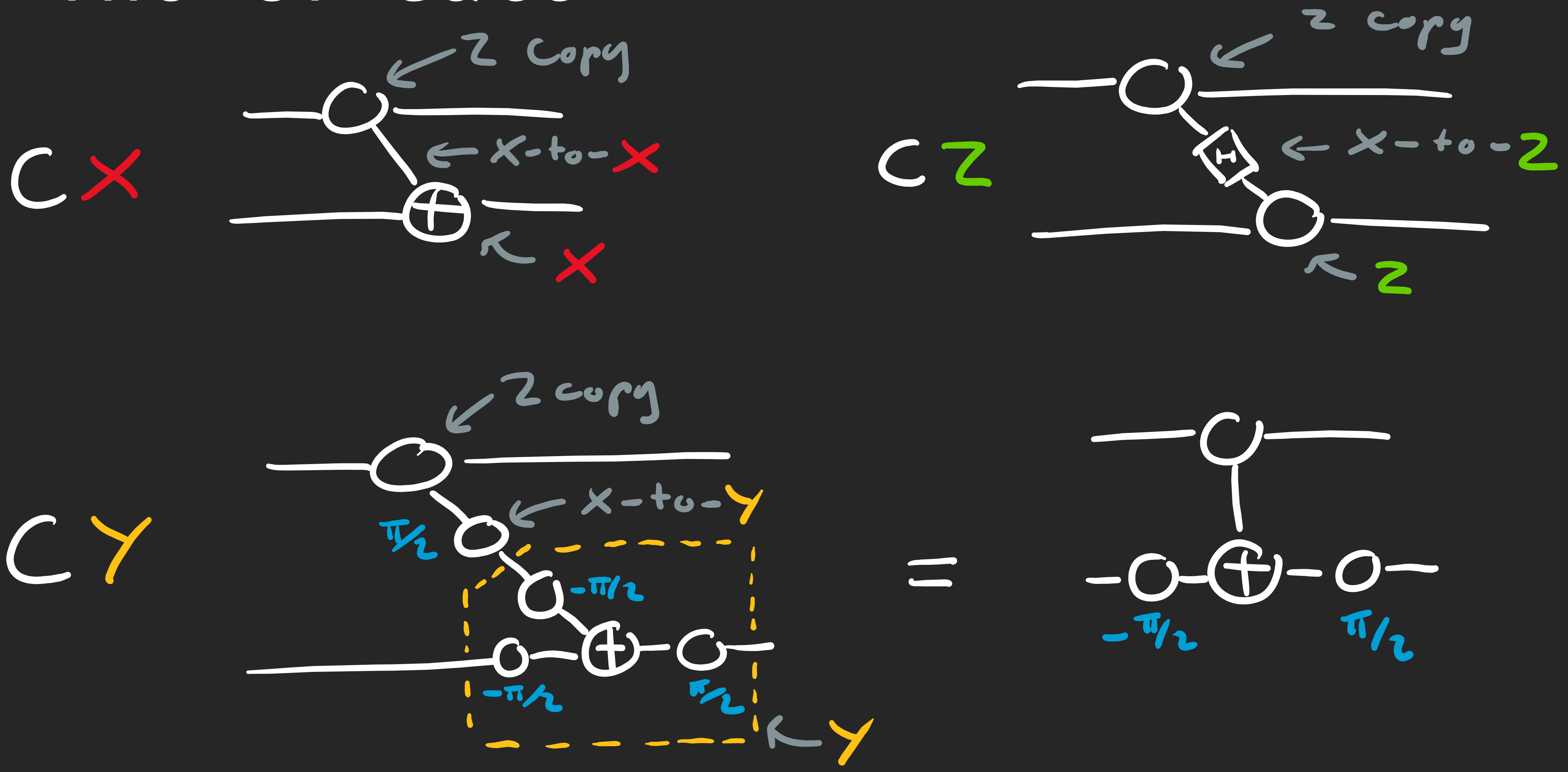
Y basis spider



One More Conjugation Rule

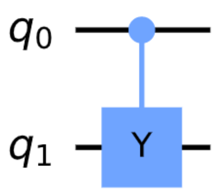
The CY Gate

(Z copy copies X basis
=> X-to-? conversion in between)

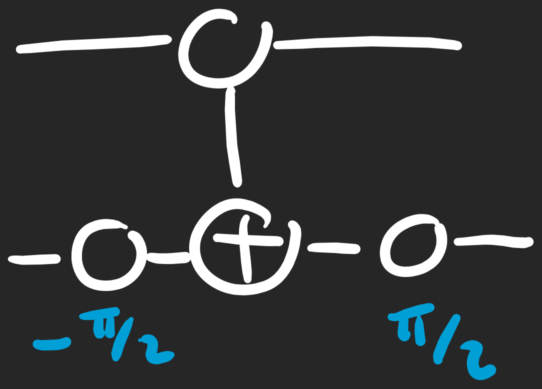


The CY Gate

```
circ = QuantumCircuit(2)
circ.cy(0,1)
circ.draw("mpl")
```

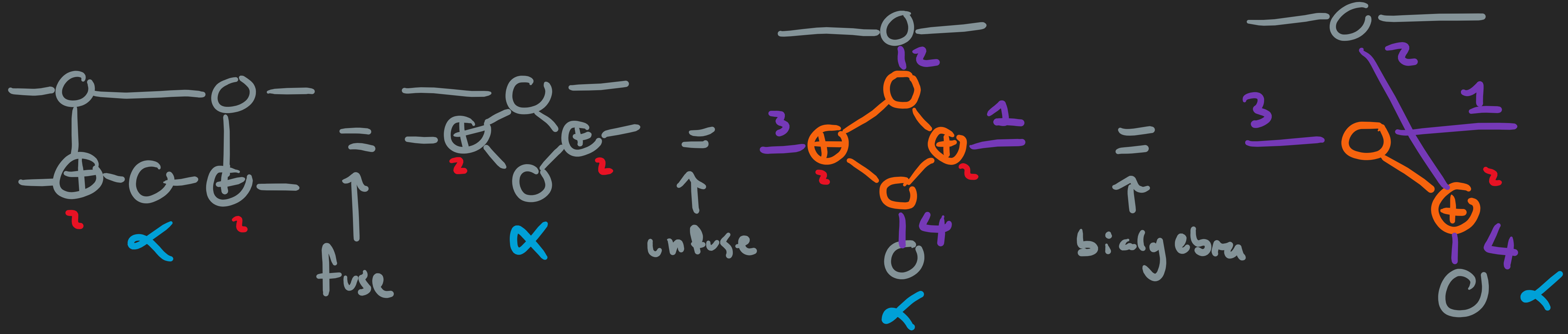


=

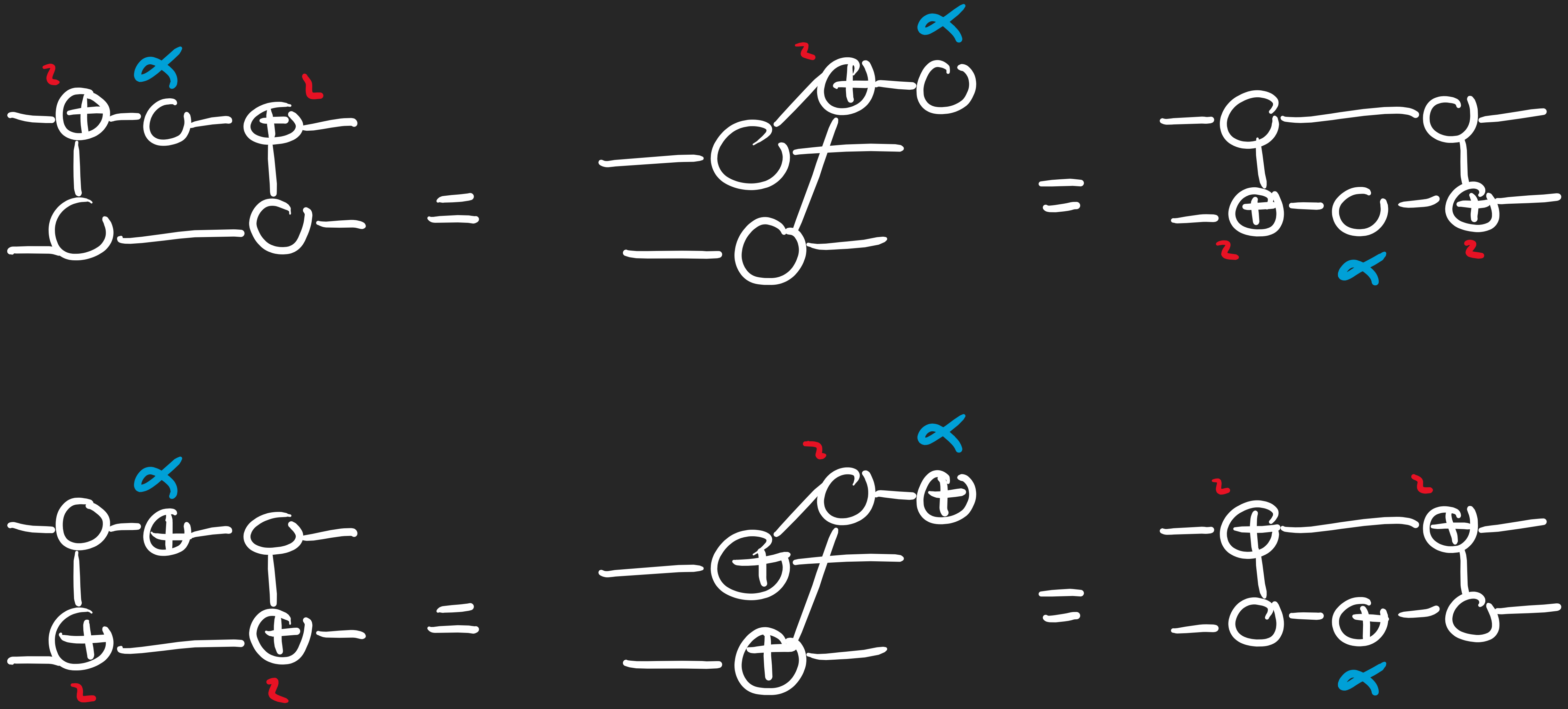


Phase Gadgets

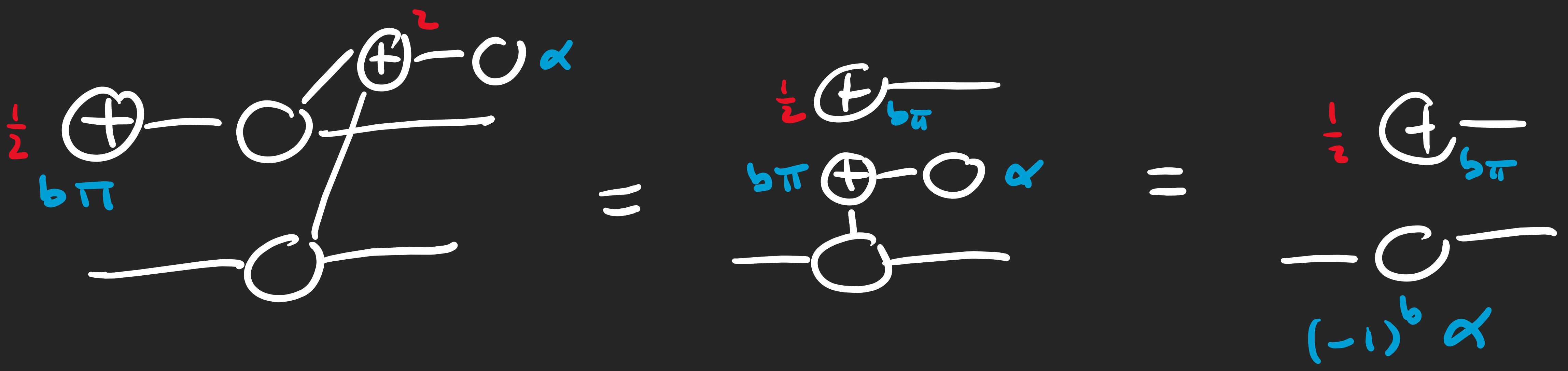
Proof:



Phase Gadgets



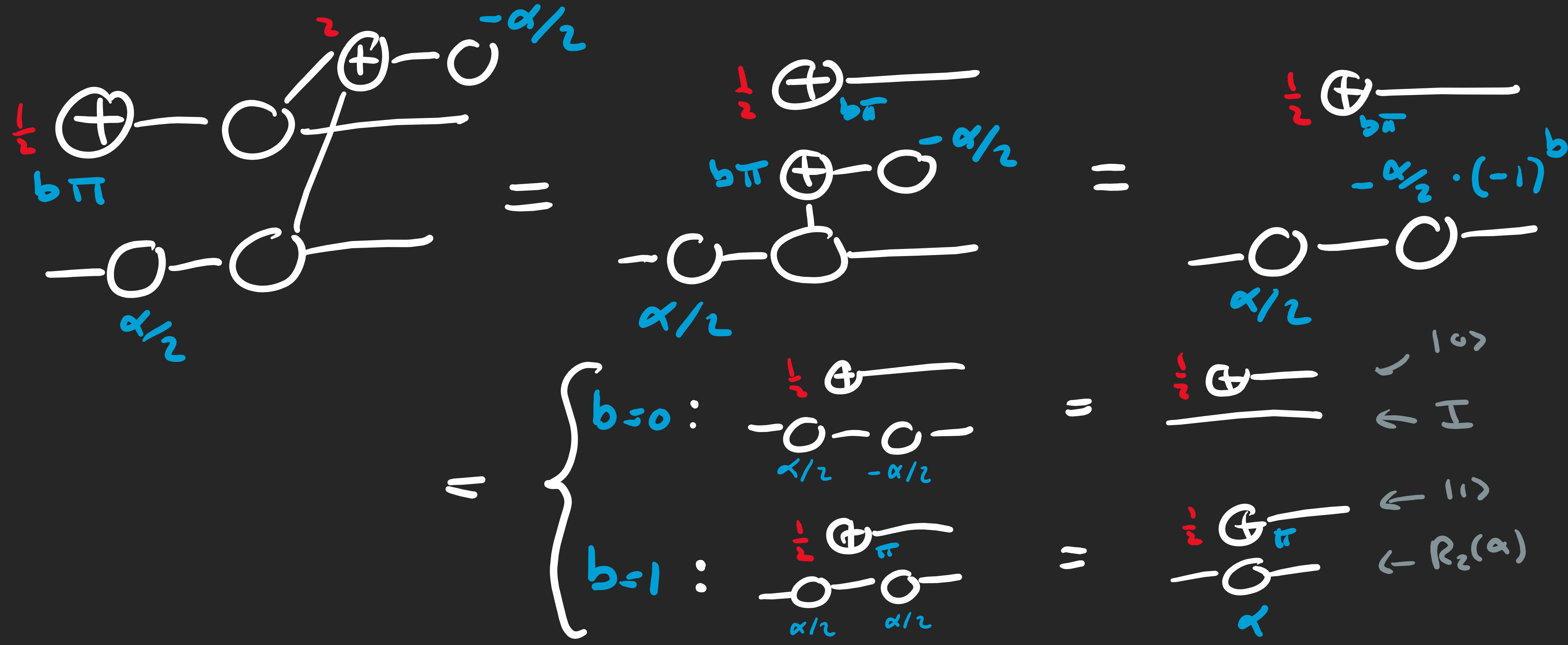
The CRz gate



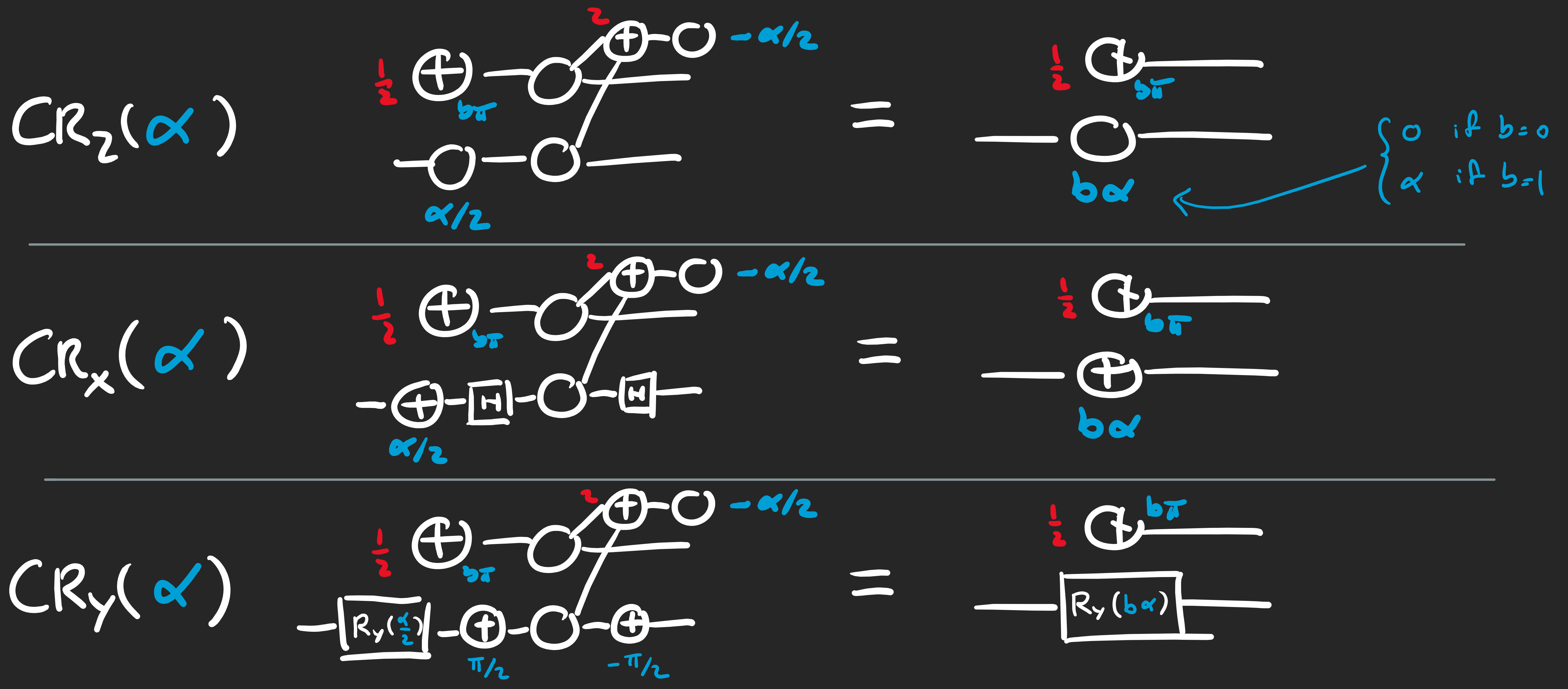
This gives us an idea ...

The CRz gate

Trick: $\dfrac{x-(-1)^b x}{2} = \begin{cases} 0 & \text{if } b=0 \\ x & \text{if } b=1 \end{cases}$

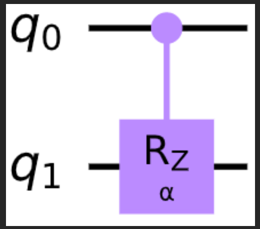


The CRz, CRx and Cry gates

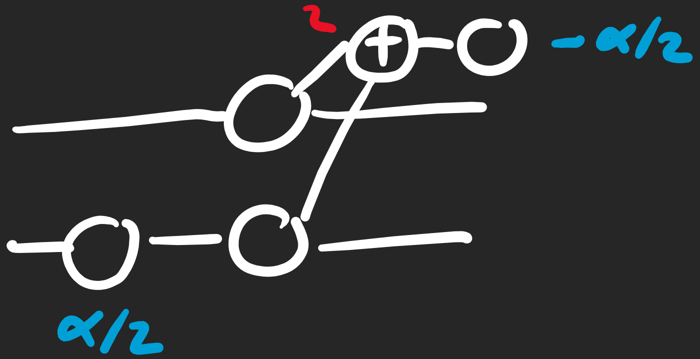


The CRz, CRx and CRy gates

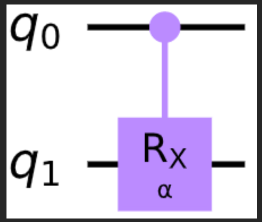
$CR_z(\alpha)$



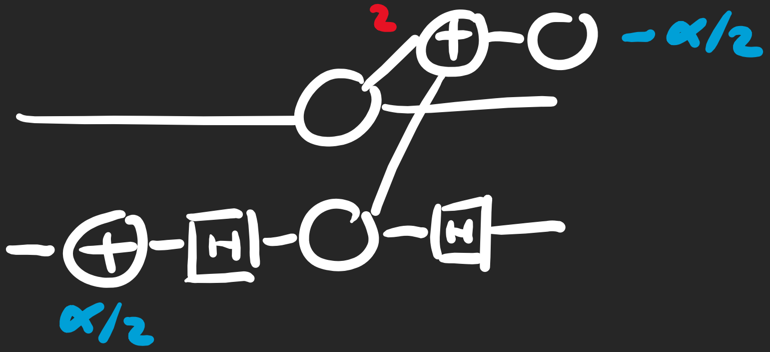
$:=$



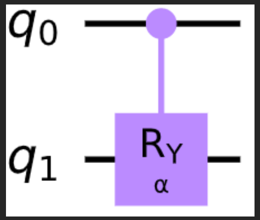
$CR_x(\alpha)$



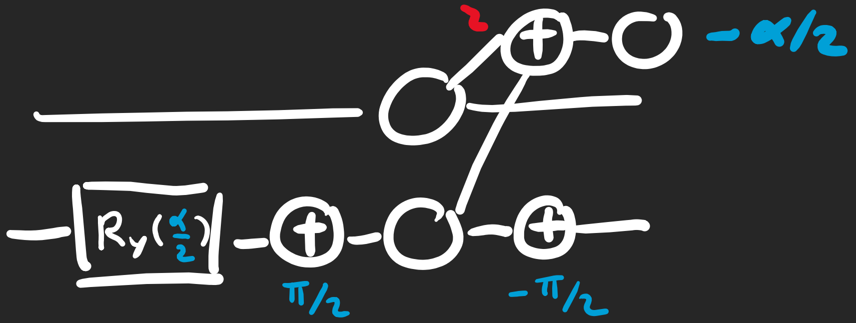
$:=$



$CR_y(\alpha)$

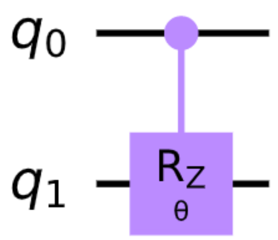


$:=$

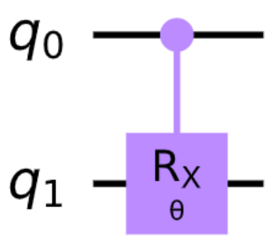


The CRz, CRx and CRy gates

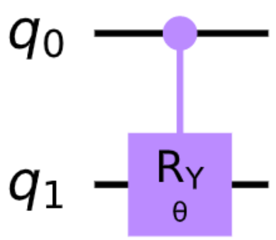
```python
theta = Parameter("θ")
circ = QuantumCircuit(2)
circ.crz(theta, 0, 1)
circ.draw("mpl")
```




```python
theta = Parameter("θ")
circ = QuantumCircuit(2)
circ.crx(theta, 0, 1)
circ.draw("mpl")
```

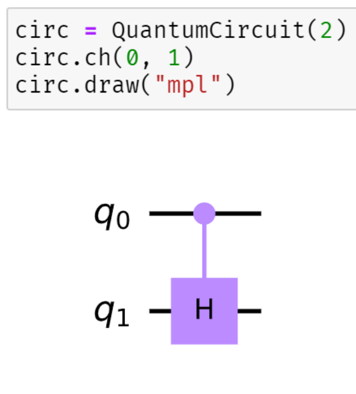



```python
theta = Parameter("θ")
circ = QuantumCircuit(2)
circ.cry(theta, 0, 1)
circ.draw("mpl")
```



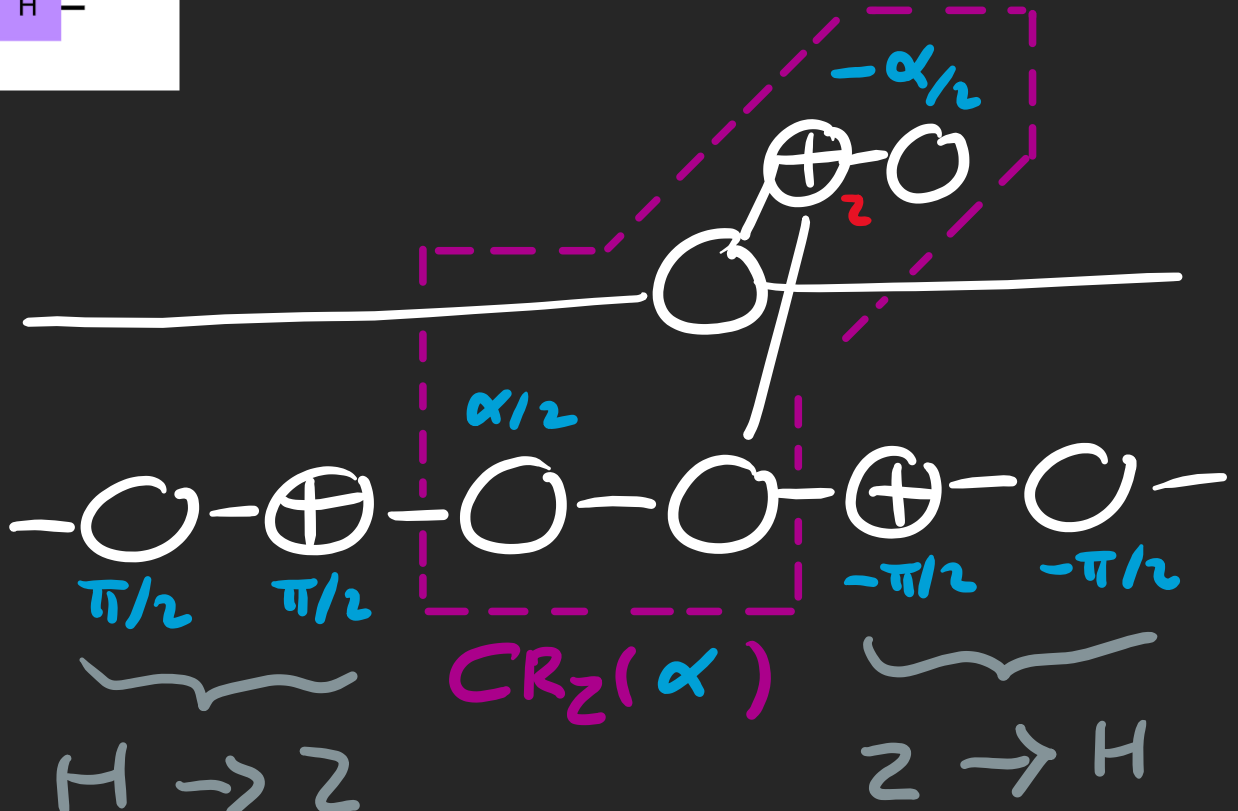
Bonus!!

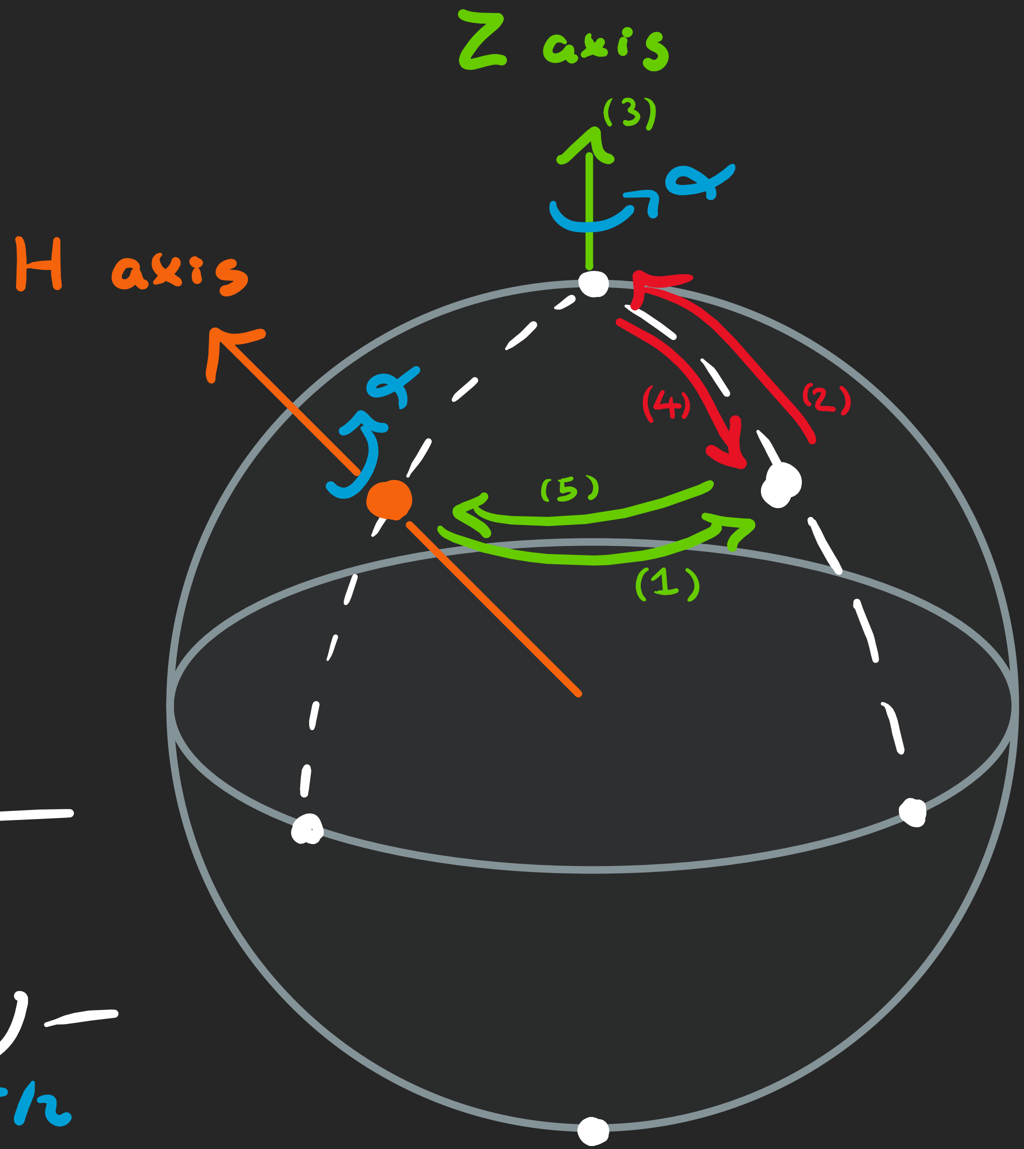
The CH Gate



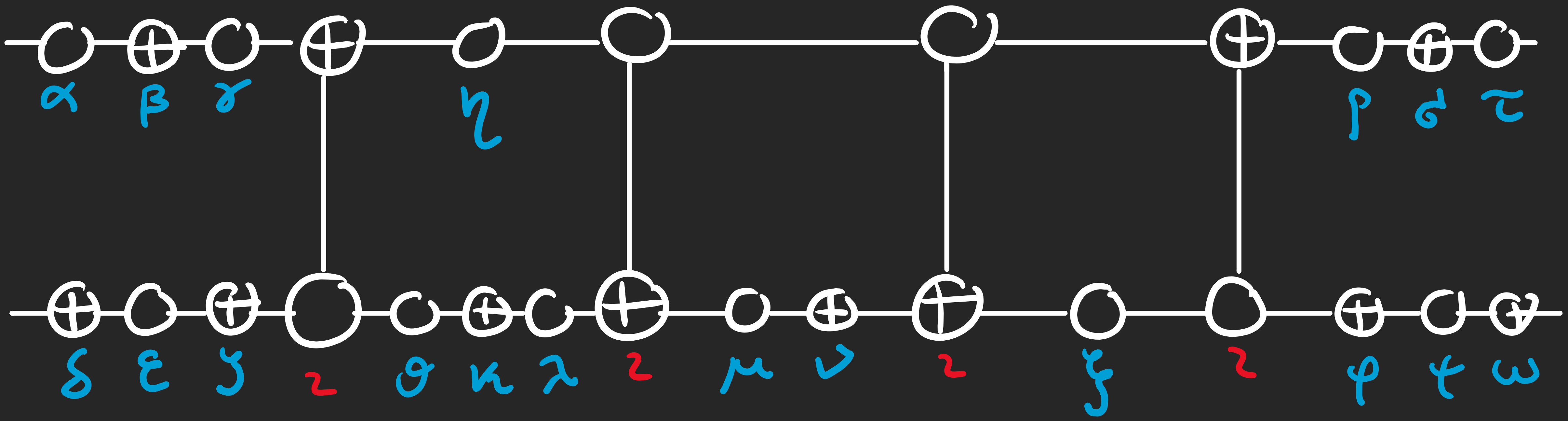
```
circ = QuantumCircuit(2)
circ.ch(0, 1)
circ.draw("mpl")
```


The CH gate is the rotation below in the case $\alpha = \pi$:





Generic 2-qubit Rotation

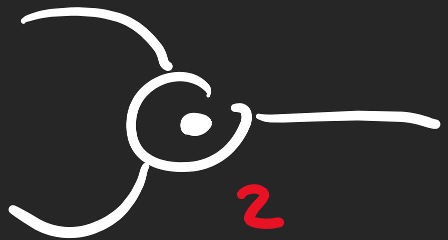


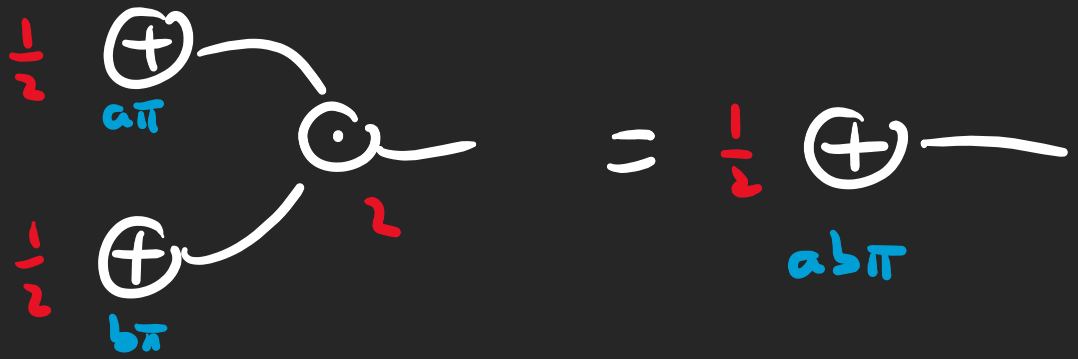
19 single-qubit rotations + 4 CNOT gates

(slightly redundant: 19 parameters for a 15-dim Lie group)

The Product Monoid

$$\sum_{a,b\in\{0,1\}} |a\cdot b\rangle\langle ba| \longleftrightarrow$$





The Product Monoid

·	0	1
0	0	0
1	0	1

Binary Arithmetic

AND	F	T
F	F	F
T	F	T

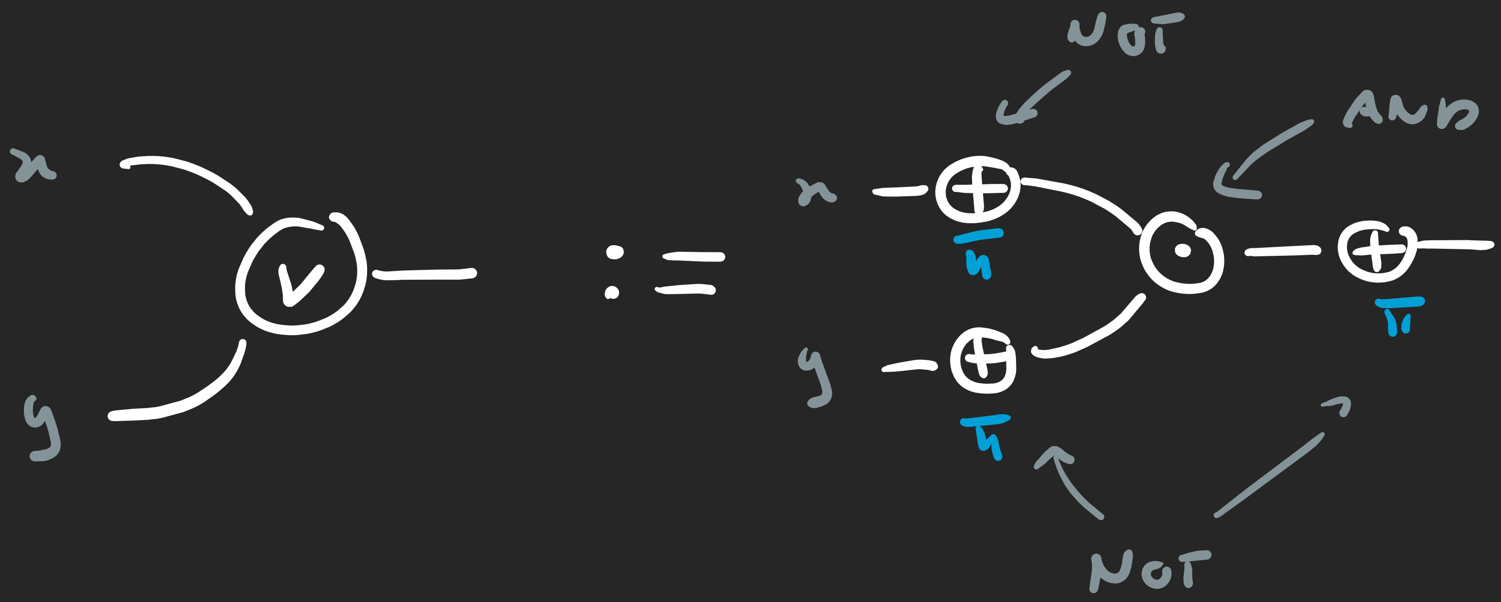
Boolean Logic

The Product Monoid

$\neg$ = NOT $\qquad \wedge$ = AND

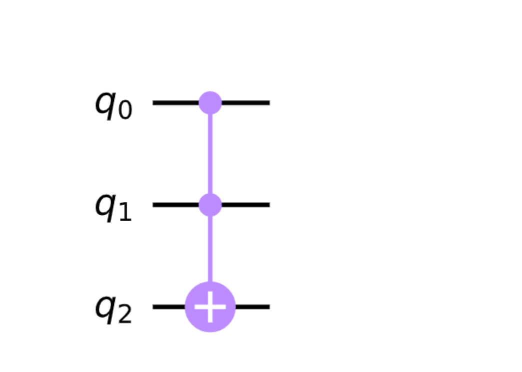
$\oplus$ = XOR $\qquad \vee$ = OR

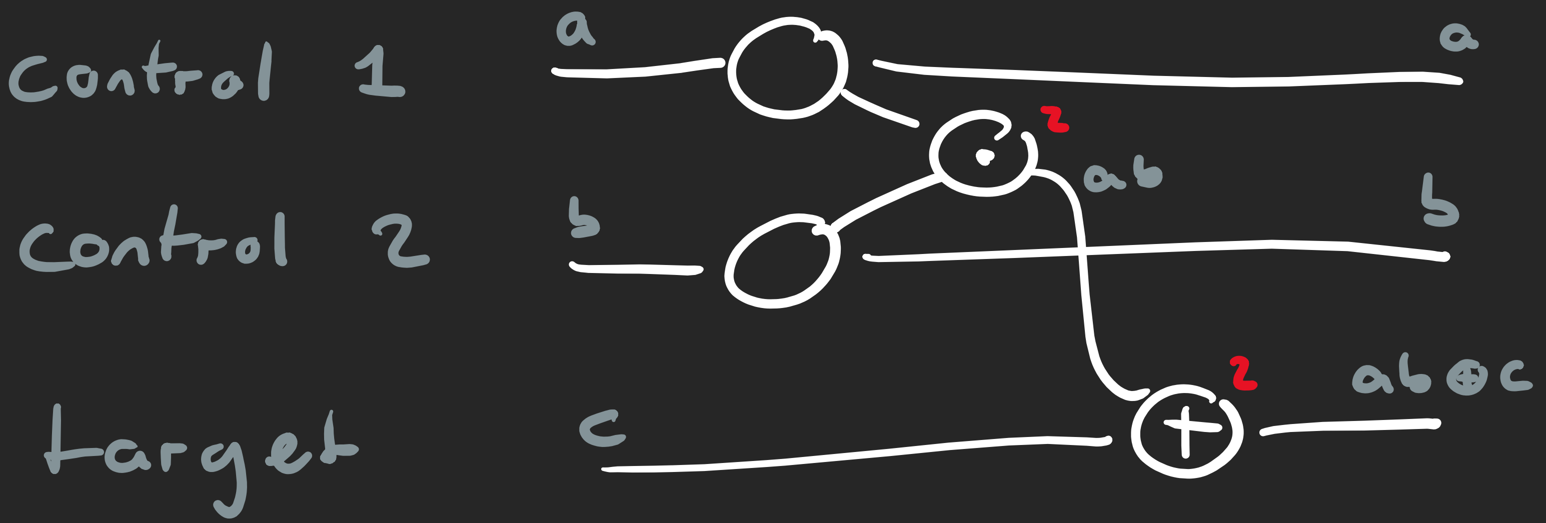
$$x \vee y = \neg((\neg x) \wedge (\neg y))$$



The Toffoli (CCX) Gate

```
circ = QuantumCircuit(3)
circ.ccx(0,1,2) # The Toffoli gate
circ.draw("mpl")
```

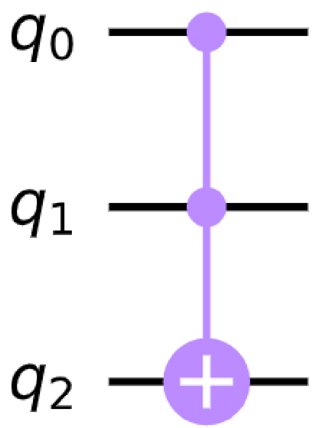




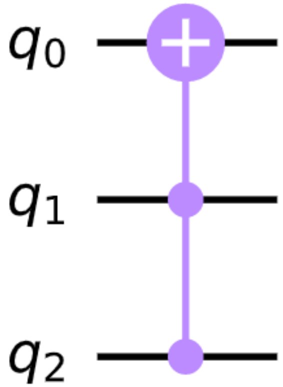
Doubly-controlled NOT/x gate on target

The Toffoli (CCX) Gate

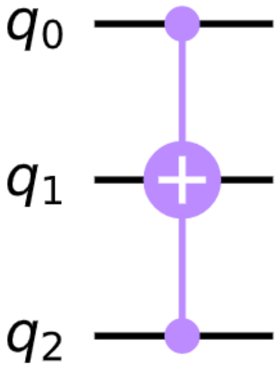
```
circc = QuantumCircuit(3)
circ.ccx(0,1,2)
circ.draw("mpl")
```



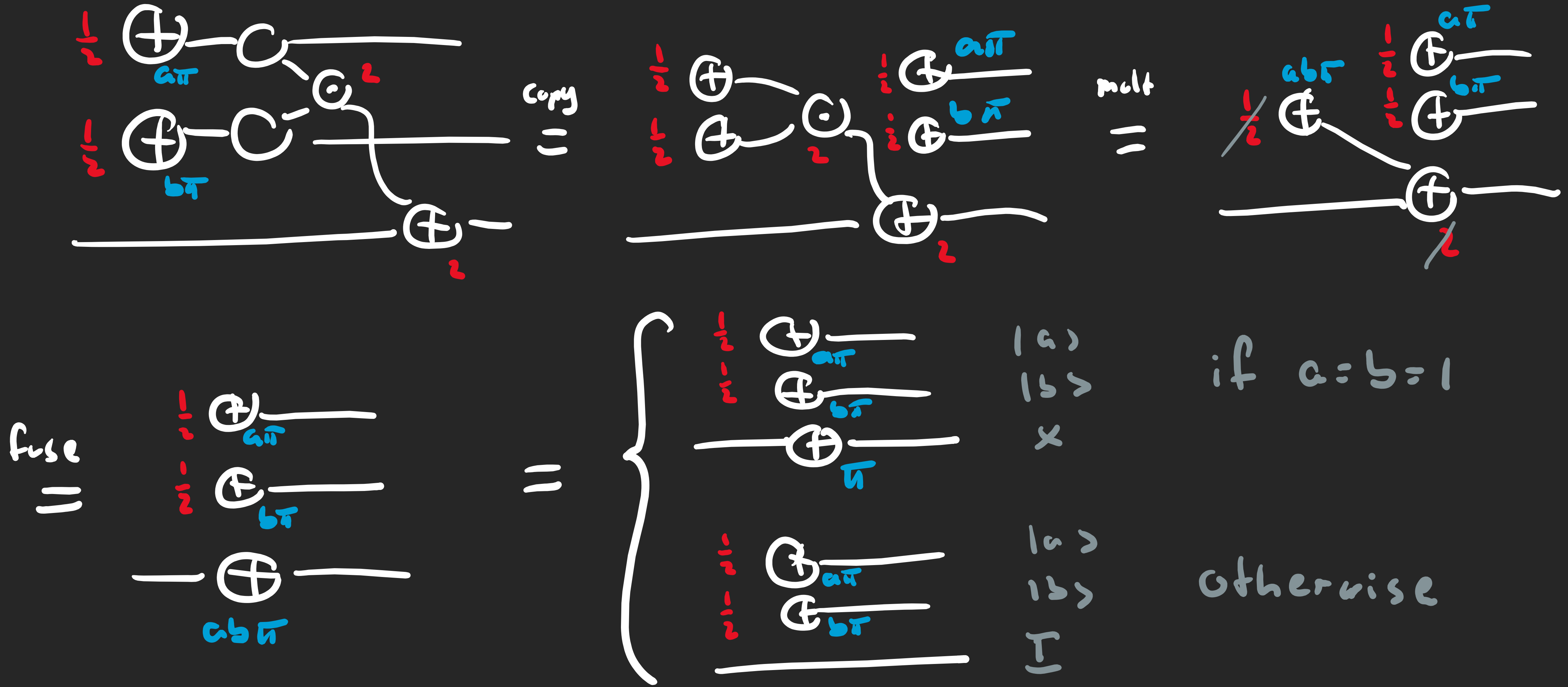

```
circ = QuantumCircuit(3)
circ.ccx(1, # control
         2, # control
         0) # target
circ.draw("mpl")
```




```
circ = QuantumCircuit(3)
circ.ccx(2, # control
         0, # control
         1) # target
circ.draw("mpl")
```

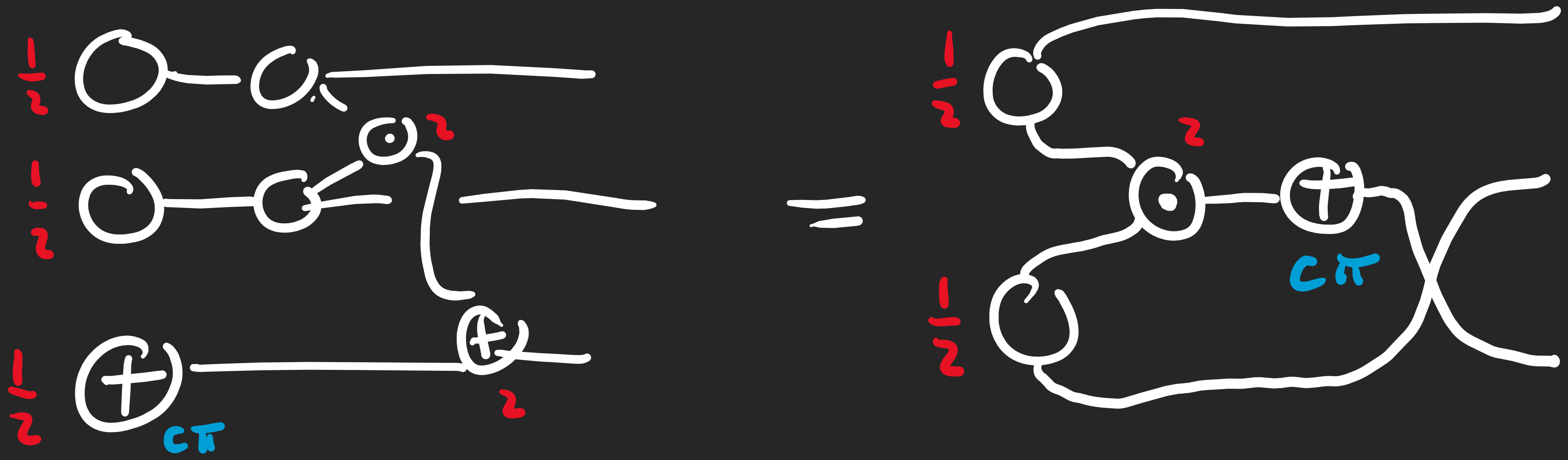


The Toffoli (CCX) Gate



The Toffoli (CCX) Gate

If the controls are not in a Z-basis state, the CCX gate results in an entangled state:



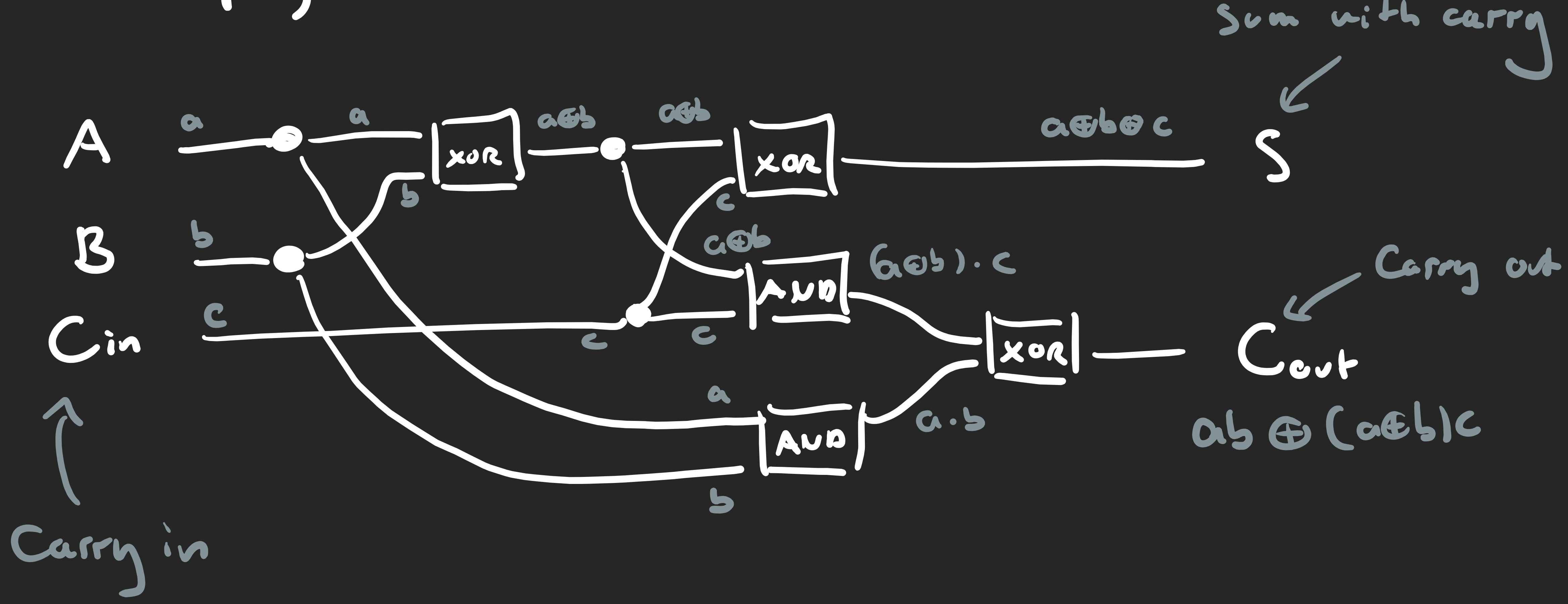
(an odd 3-qubit entangled state)

Boolean Circuits

Boolean circuit are classical binary circuits using logical gates such as :

Boolean Circuits

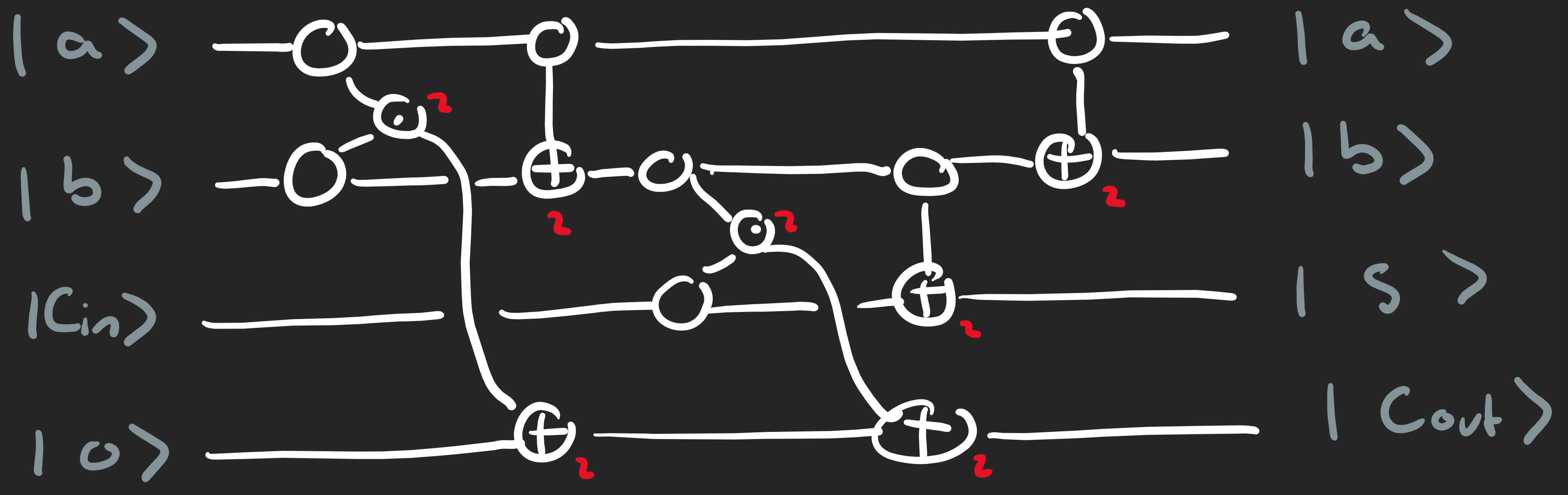
For example, this is the circuit for a FULL ADDER:



Boolean Circuits

Logical gates are not available natively in QC, but they can be realised using X, CX and CCX:

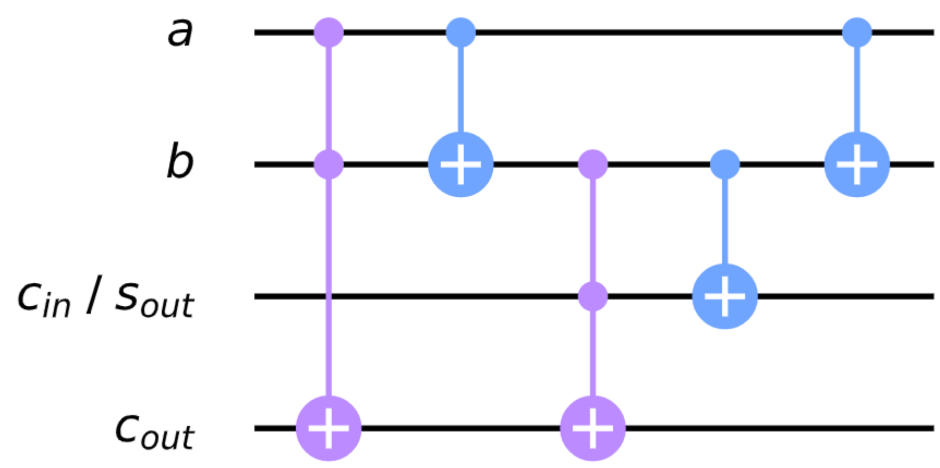
Quantum Full Adder



"ancilla", starting at |0⟩

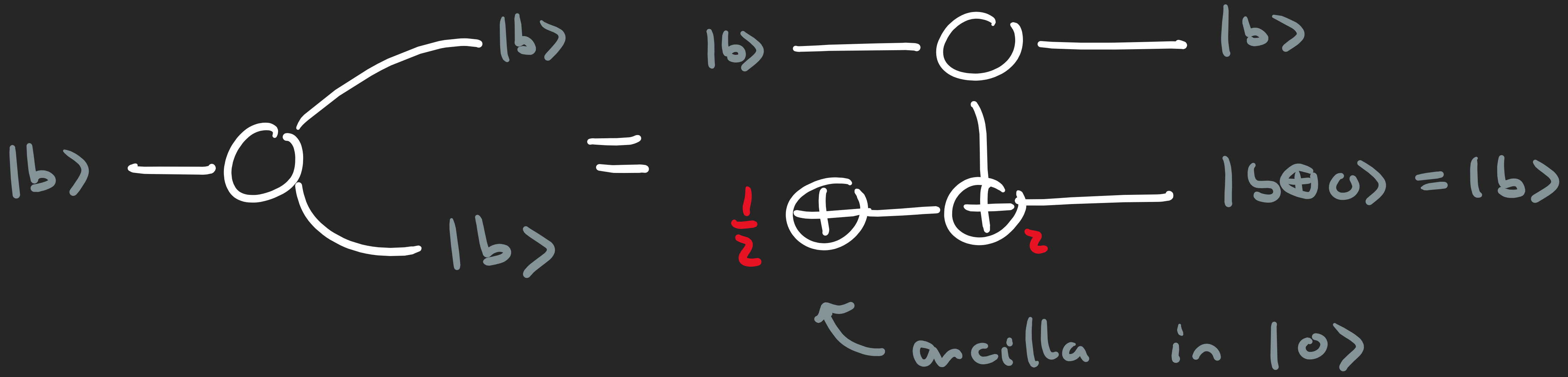
Boolean Circuits

```
from qiskit.circuit import QuantumRegister
circ = QuantumCircuit(QuantumRegister(1,"a "),
                      QuantumRegister(1,"b "),
                      QuantumRegister(1,"c_{in} / s_{out} "),
                      QuantumRegister(1,"c_{out} "))
circ.ccx(0, 1, 3)
circ.cx(0, 1)
circ.ccx(1, 2, 3)
circ.cx(1, 2)
circ.cx(0, 1)
circ.draw("mpl")
```



Quantum Full Adder in Qiskit

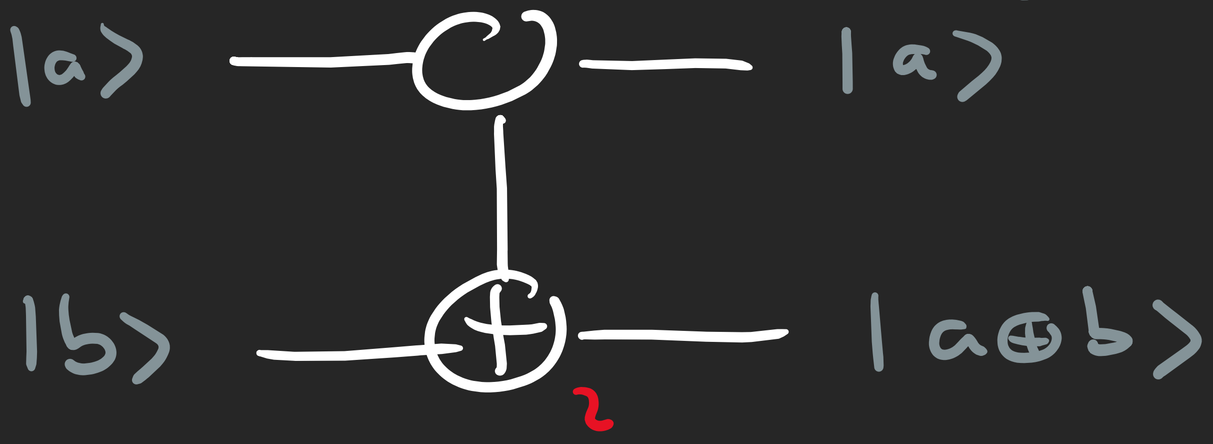
Boolean Circuits



COPY

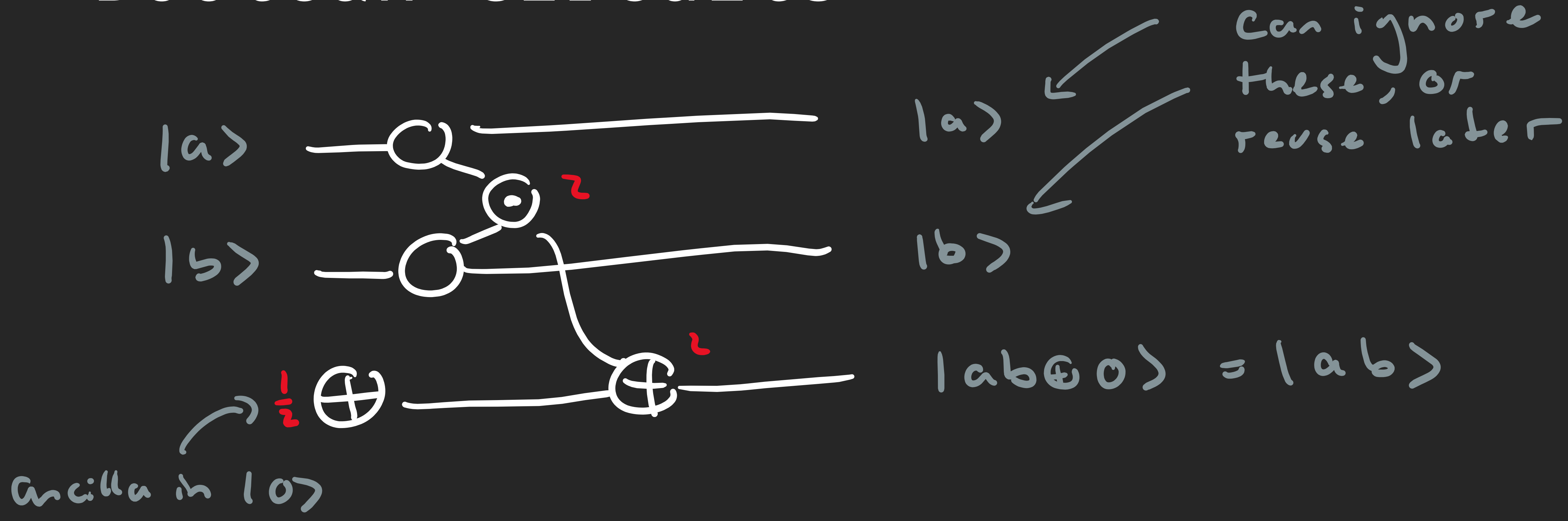
Boolean Circuits

Can ignore, or re-use later



XOR

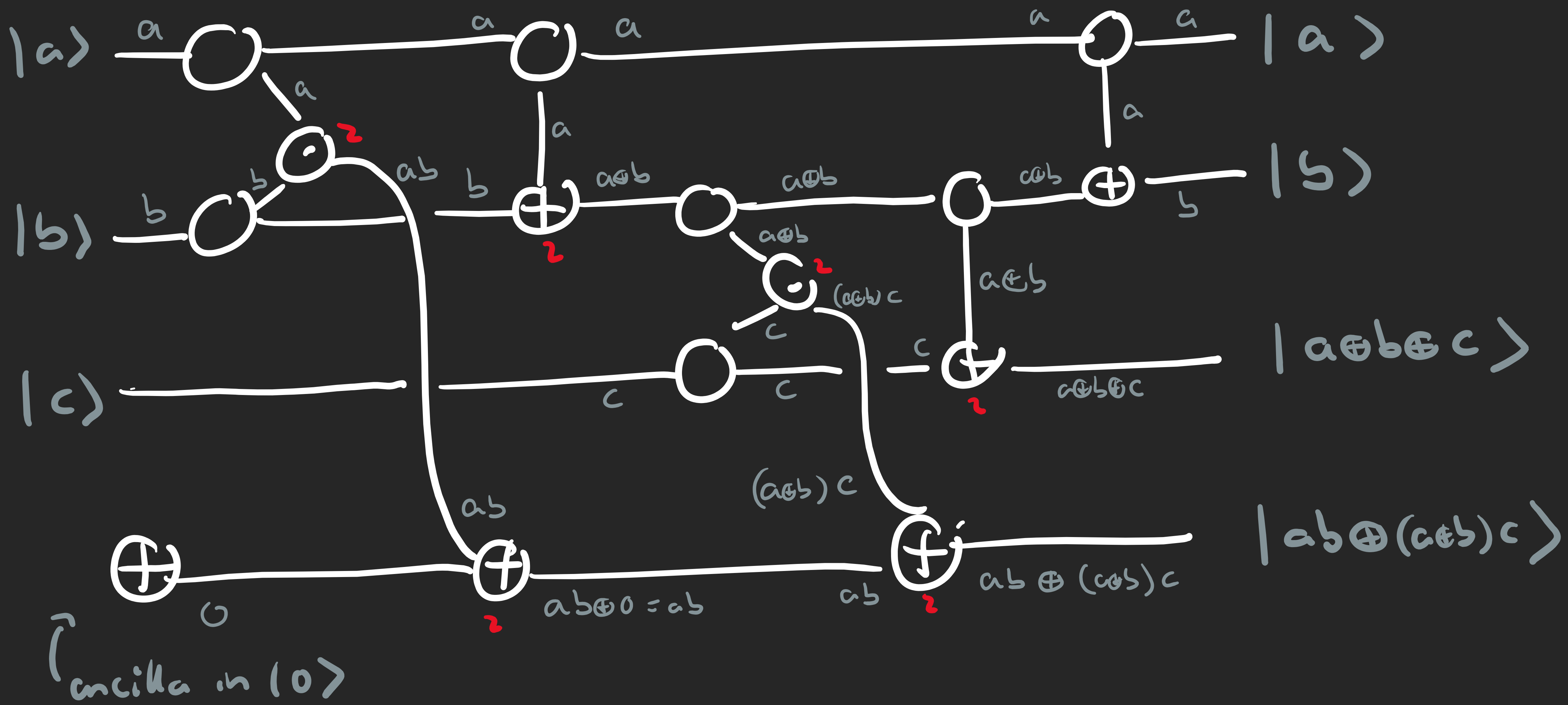
Boolean Circuits



AND

Boolean Circuits

Quantum circuit for a FULL ADDER



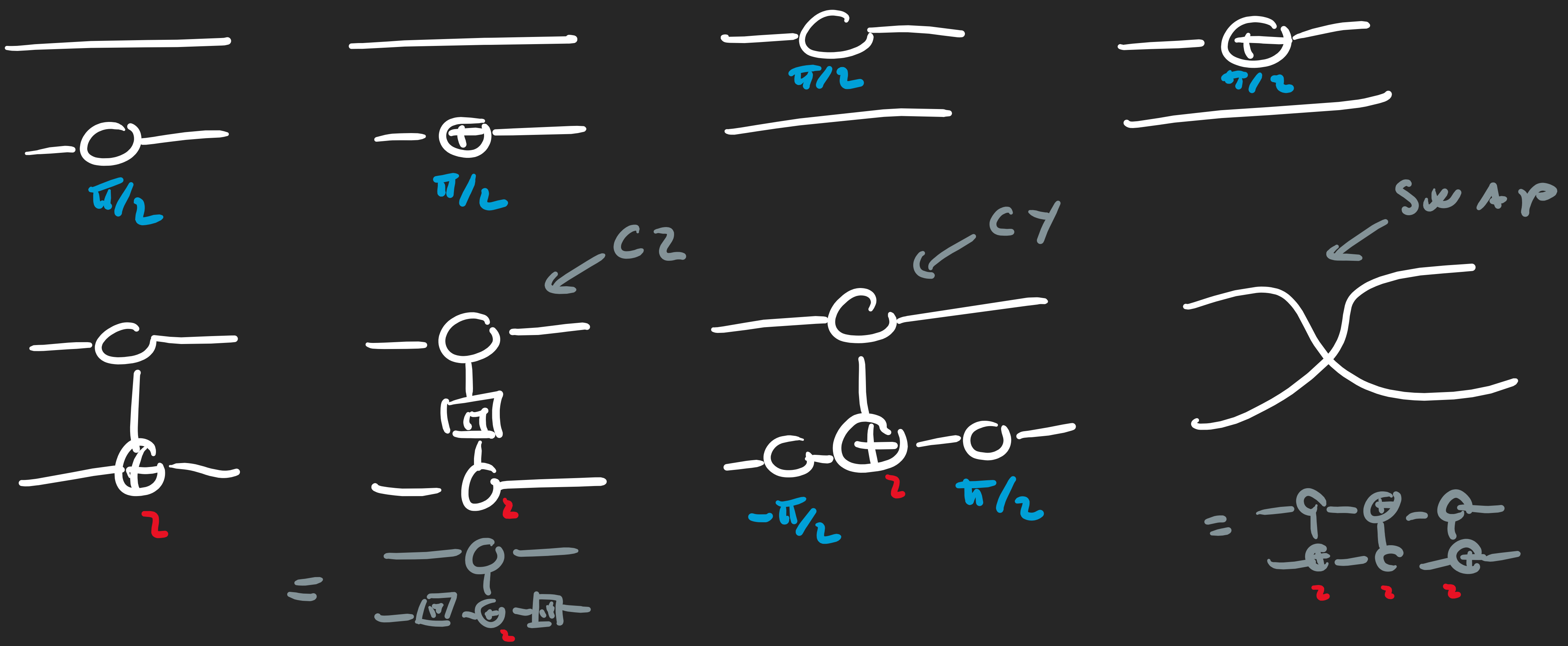
Clifford Circuits

Clifford Circuits

= all combinations of
- 1-qubit stabiliser states
- 1-qubit Clifford gates
- the 2-qubit cx gate

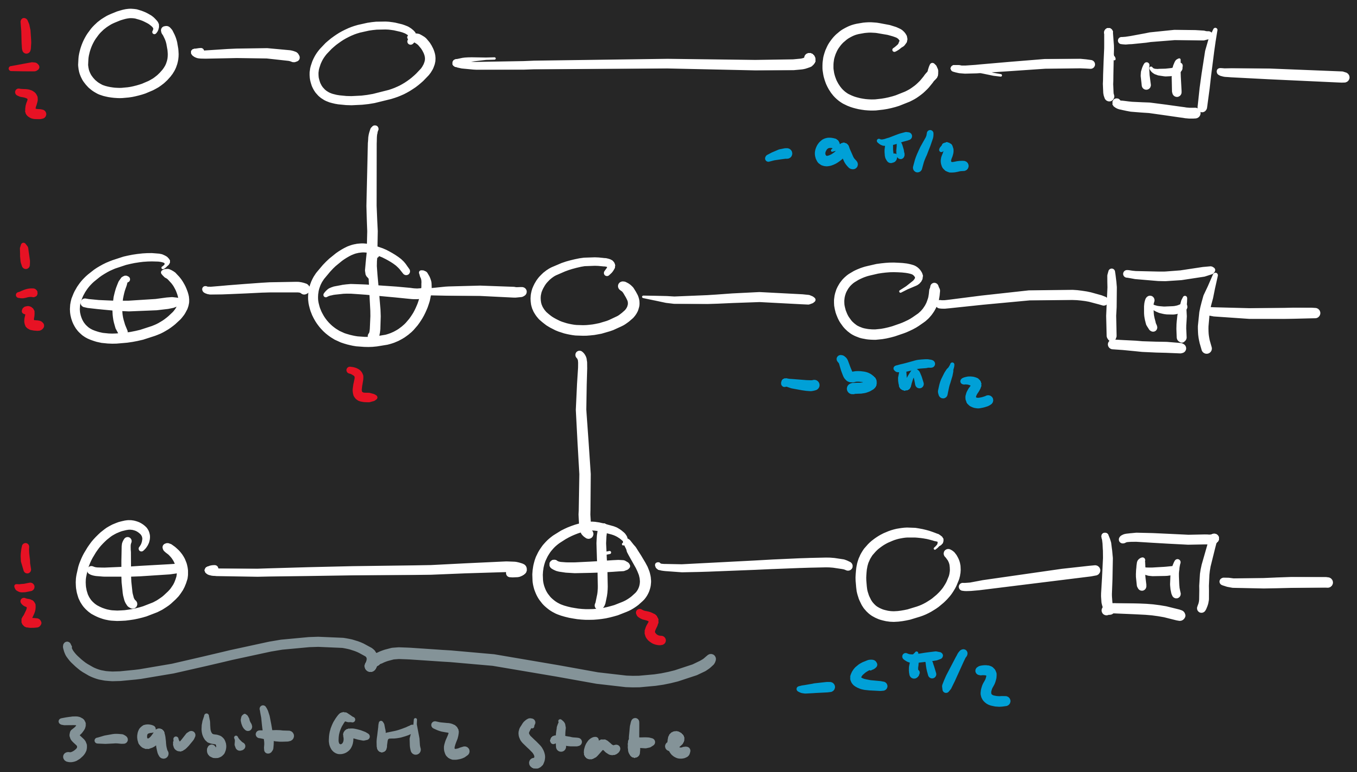
Clifford Circuits

Examples of 2-qubit Clifford Circuits:



Clifford Circuits

Examples of 3-qubit Clifford Circuit:



This family of $8 = 2^3$ 3-qubit entangled states is used in the famous Mermin non-locality argument.

$a, b, c \in \{0, 1\}$

Clifford Circuits

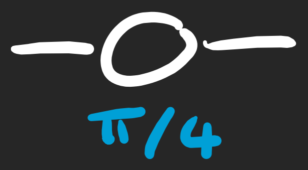
The Gottesman-Knill Theorem says that Clifford circuits are always easy to simulate using classical computers.

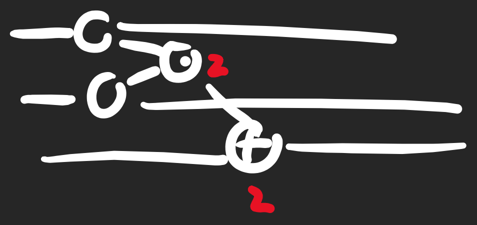
$\Rightarrow$ no quantum advantage!

Clifford Circuits

If we wish to achieve quantum advantage, we need at least one non-Clifford gate (ANY non-Clifford gate will do!)

Examples: T gate, Toffoli gate





Clifford+T Circuits

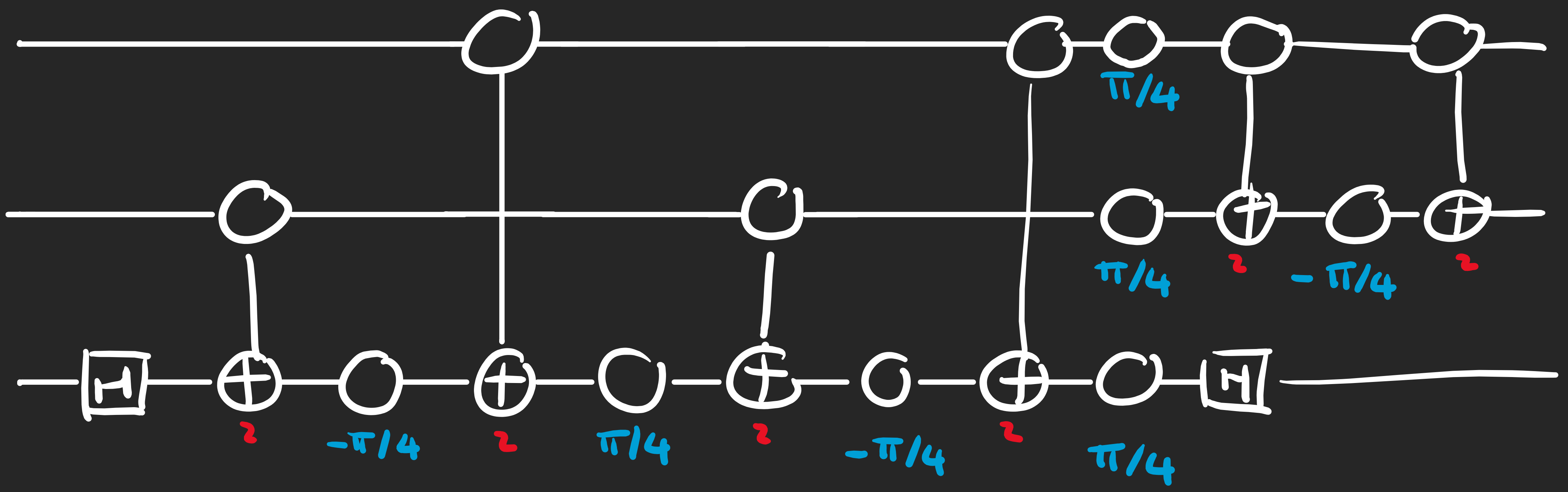
T gates are the typical non-Clifford resource.

$\Rightarrow$ Clifford + T circuits

(approximately universal for QC)

# of T gates $\sim$ classical hardness

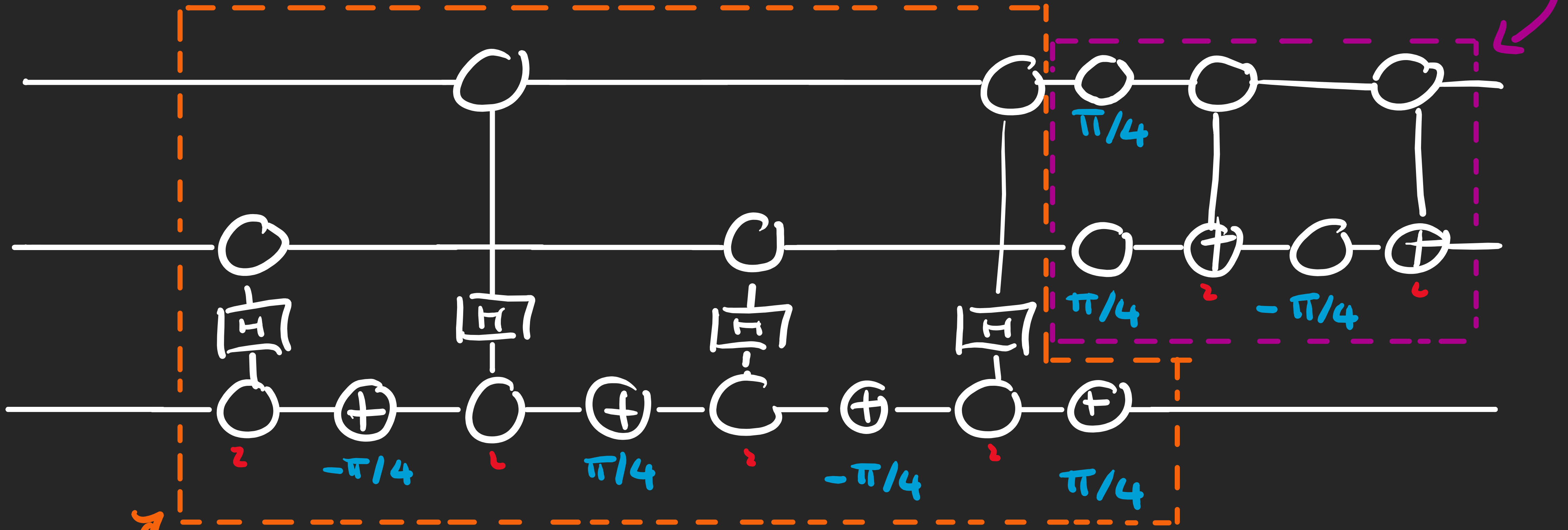
The Toffoli (CCX) Gate



Minimum # of π/4 (non-Clifford) rotations: 7

The Toffoli (CCX) Gate

Z rotation and Controlled Z rotation to fix "Kickbacks" from CZ gates



Controlled X rotation

00 : $-\pi/4 + \pi/4 - \pi/4 + \pi/4 = 0$

01 : $-(-\pi/4 + \pi/4) - \pi/4 + \pi/4 = 0$

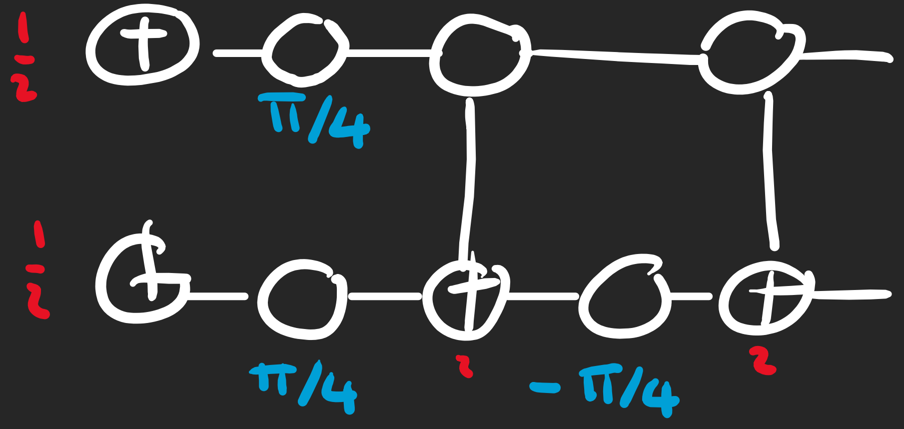
10 : $-\pi/4 - (\pi/4 - \pi/4) + \pi/4 = 0$

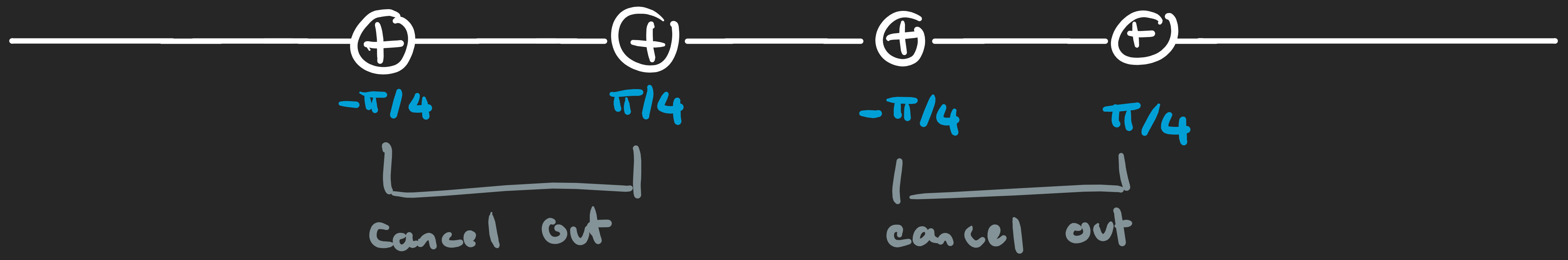
11 : $-(-\pi/4) + \pi/4 - (-\pi/4) + \pi/4 = \pi$

The Toffoli (CCX) Gate

Controls in $|00\rangle$

$\Rightarrow R_x(0)$ on target

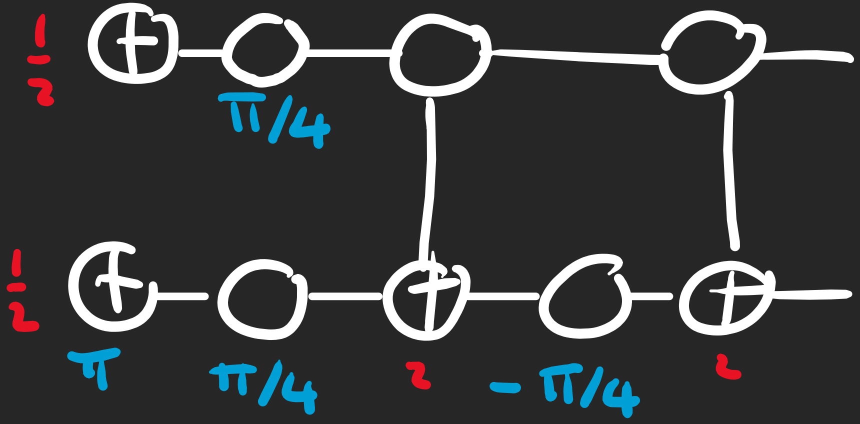


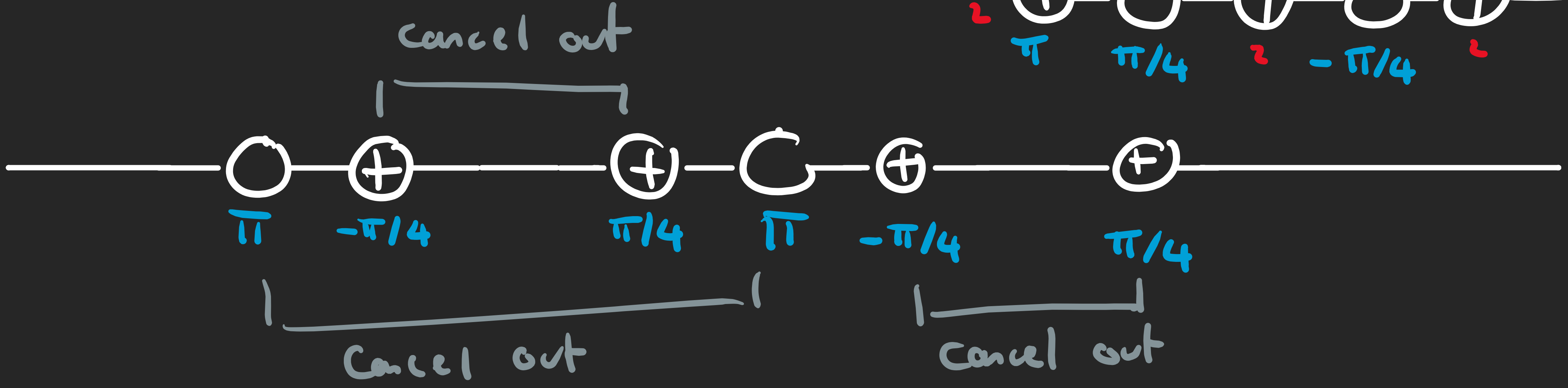


The Toffoli (CCX) Gate

Controls in $|01\rangle$

$\Rightarrow R_x(0)$ on target

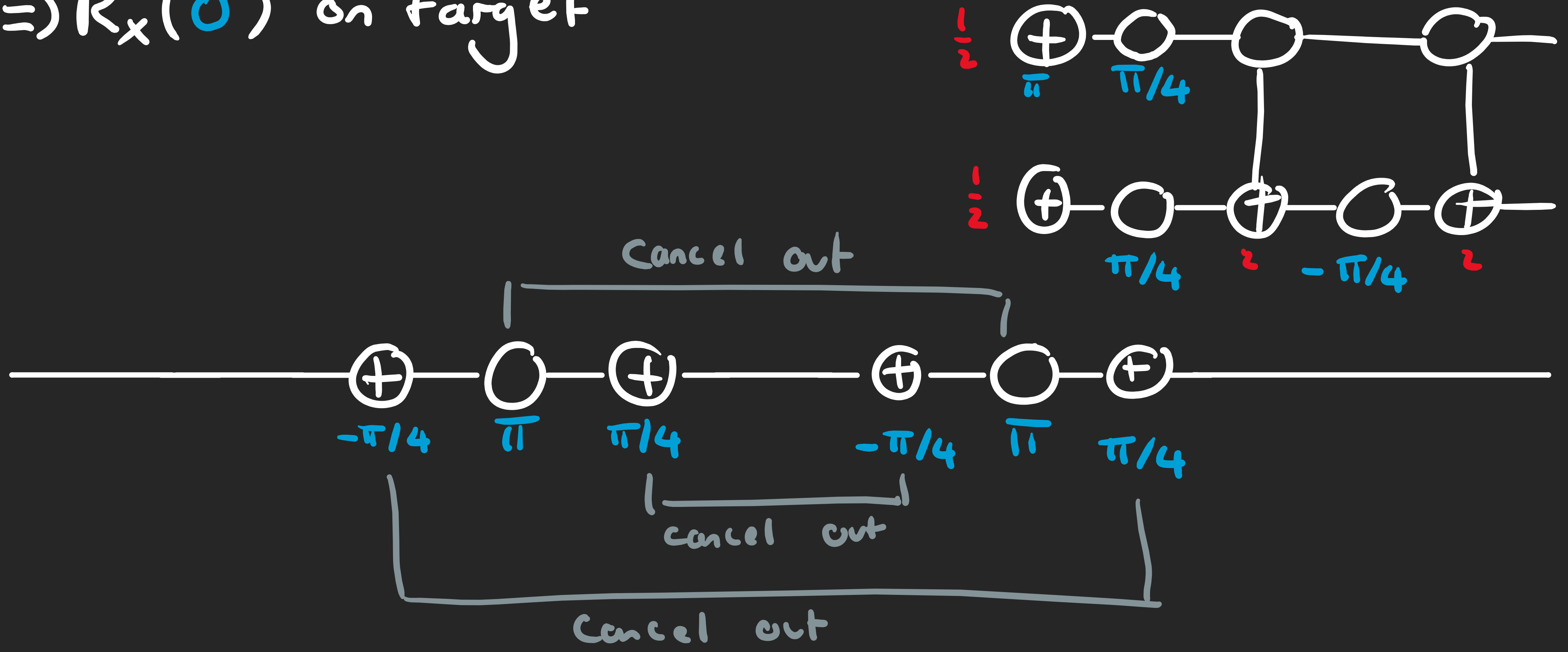




The Toffoli (CCX) Gate

Controls in $|1\ 0\rangle$

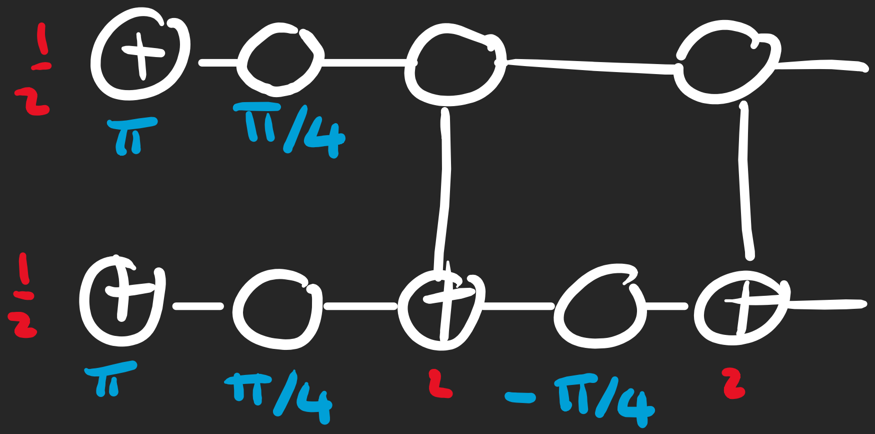
$\Rightarrow R_x(0)$ on target

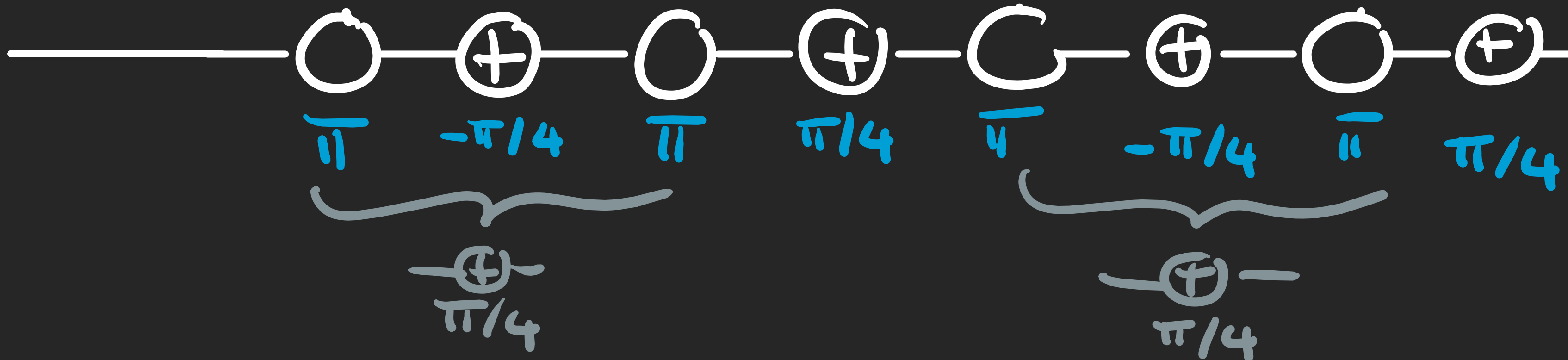


The Toffoli (CCX) Gate

Controls in $|11\rangle$

$\Rightarrow R_x(\pi)$ on target





Add all together: