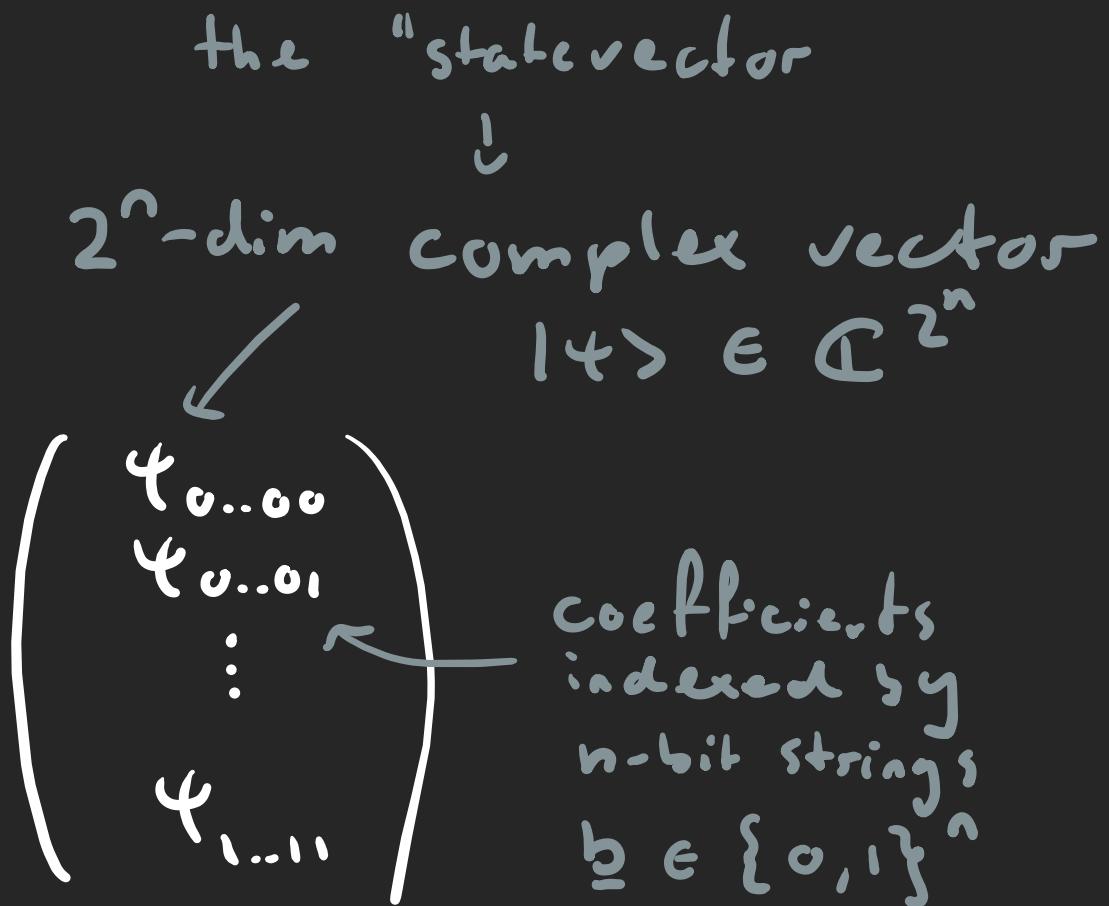


Many-qubit gates

Bracket Notation

States of n qubits:

$$|\psi\rangle =$$



(i) normalised

$$\sum_{b \in \{0,1\}^n} |\psi_b|^2 = 1$$

(ii) up to phase

$$|\psi\rangle = e^{i\theta} |\psi\rangle$$

Bracket Notation

Product states on 2 qubits:

joint state of both qubits

$$|\Psi\rangle \otimes |\Psi\rangle := \begin{pmatrix} \varphi_0(\varphi_0, \varphi_1) \\ \varphi_1(\varphi_0, \varphi_1) \end{pmatrix}$$

Each qubit has its own well-defined state

Diagram illustrating the joint state of two qubits. The expression $|\Psi\rangle \otimes |\Psi\rangle$ is shown with curly braces above and below the tensor product symbol, indicating it represents a joint state of both qubits. Below the expression, curly braces are placed under each qubit state, with arrows pointing to the text "Each qubit has its own well-defined state". To the right, the joint state is represented as a column vector with two components: $\varphi_0(\varphi_0, \varphi_1)$ and $\varphi_1(\varphi_0, \varphi_1)$. The first component is colored red, and the second is colored blue.

Bracket Notation

Product states on 2 qubits:

joint state of both qubits

$$|\Psi\rangle \otimes |\Psi\rangle := \begin{pmatrix} |\Psi_0\rangle & |\Psi_0\rangle \\ |\Psi_0\rangle & |\Psi_1\rangle \\ |\Psi_1\rangle & |\Psi_0\rangle \\ |\Psi_1\rangle & |\Psi_1\rangle \end{pmatrix}$$

Each qubit has its own well-defined state

The diagram illustrates the tensor product of two qubit states. On the left, the expression $|\Psi\rangle \otimes |\Psi\rangle$ is shown, where each $|\Psi\rangle$ is underlined by a bracket. Above this, a larger bracket spans both terms, labeled "joint state of both qubits". To the right, the resulting state is represented as a 2x2 matrix. The columns are labeled with red Ψ_0 and blue Ψ_0 , and the rows are labeled with red Ψ_0 and blue Ψ_1 . Below the matrix, the text "Each qubit has its own well-defined state" is written, with arrows pointing from the individual $|\Psi\rangle$ terms in the original expression to the corresponding columns and rows of the matrix.

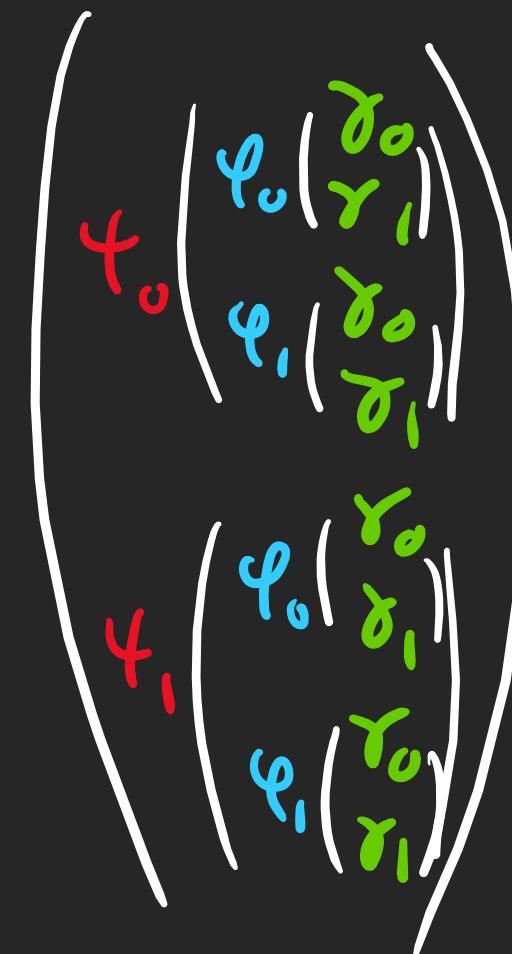
Bracket Notation

Product states on 3 qubits:

joint state of all three qubits

$$|\Psi\rangle \otimes |\Psi\rangle \otimes |\Psi\rangle :=$$

Each qubit has its own
well-defined state



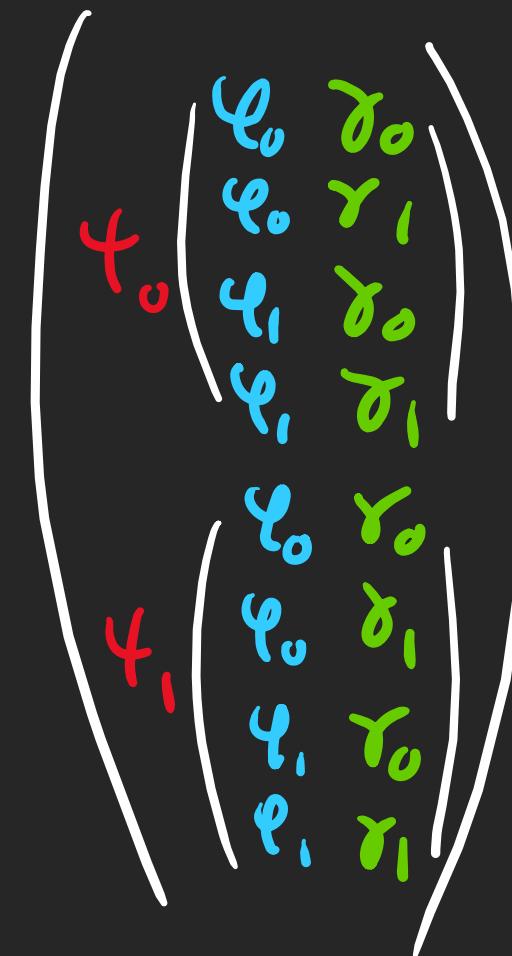
Bracket Notation

Product states on 3 qubits:

joint state of all three qubits

$$|\Psi\rangle \otimes |\Psi\rangle \otimes |\Psi\rangle :=$$

Each qubit has its own
well-defined state



Bracket Notation

Product states on 3 qubits:

joint state of all three qubits

$$|+\rangle \otimes |+\rangle \otimes |+\rangle :=$$

Each qubit has its own
well-defined state

ψ_0	φ_0	γ_0
ψ_0	φ_0	γ_1
ψ_0	φ_1	γ_0
ψ_0	φ_1	γ_1
ψ_1	φ_0	γ_0
ψ_1	φ_0	γ_1
ψ_1	φ_1	γ_0
ψ_1	φ_1	γ_1

Bracket Notation

Computational basis states on n qubits:

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

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

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

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

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

(Orthonormal basis)

Bracket Notation

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

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

$$\begin{aligned} |000\rangle &= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ |001\rangle & \end{aligned}$$

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

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

$$\begin{aligned} |010\rangle & \end{aligned}$$

$$\begin{aligned} |011\rangle &= \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \\ |100\rangle & \end{aligned}$$

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

$$\begin{aligned} |111\rangle &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \\ |1111\rangle & \end{aligned}$$

Bracket Notation

$$\begin{aligned} |\psi\rangle &= \begin{pmatrix} \psi_{0\ldots 0} \\ \psi_{0\ldots 1} \\ \vdots \\ \psi_{1\ldots 1} \end{pmatrix} = \psi_{0\ldots 0} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \psi_{0\ldots 1} \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} + \dots + \psi_{1\ldots 1} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \\ &= \psi_{0\ldots 0} |0\ldots 0\rangle + \psi_{0\ldots 1} |0\ldots 1\rangle + \dots + \psi_{1\ldots 1} |1\ldots 1\rangle \\ &= \sum_{b \in \{0,1\}^n} \psi_b |b\rangle \end{aligned}$$

Bracket Notation

$$\begin{aligned} |\Psi\rangle \otimes |\Psi\rangle &= (\varphi_0|0\rangle + \varphi_1|1\rangle) \otimes (\varphi_0|0\rangle + \varphi_1|1\rangle) \\ &= \varphi_0\varphi_0|00\rangle + \varphi_0\varphi_1|01\rangle + \varphi_1\varphi_0|10\rangle + \varphi_1\varphi_1|11\rangle \end{aligned}$$

Bracket Notation

2-qubit Bell state:

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

Bracket Notation

2-qubit Bell state:

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

This is an entangled state: it cannot be written as $|\Phi^+\rangle = |\psi\rangle \otimes |\psi\rangle$ for any $|\psi\rangle$

Bracket Notation

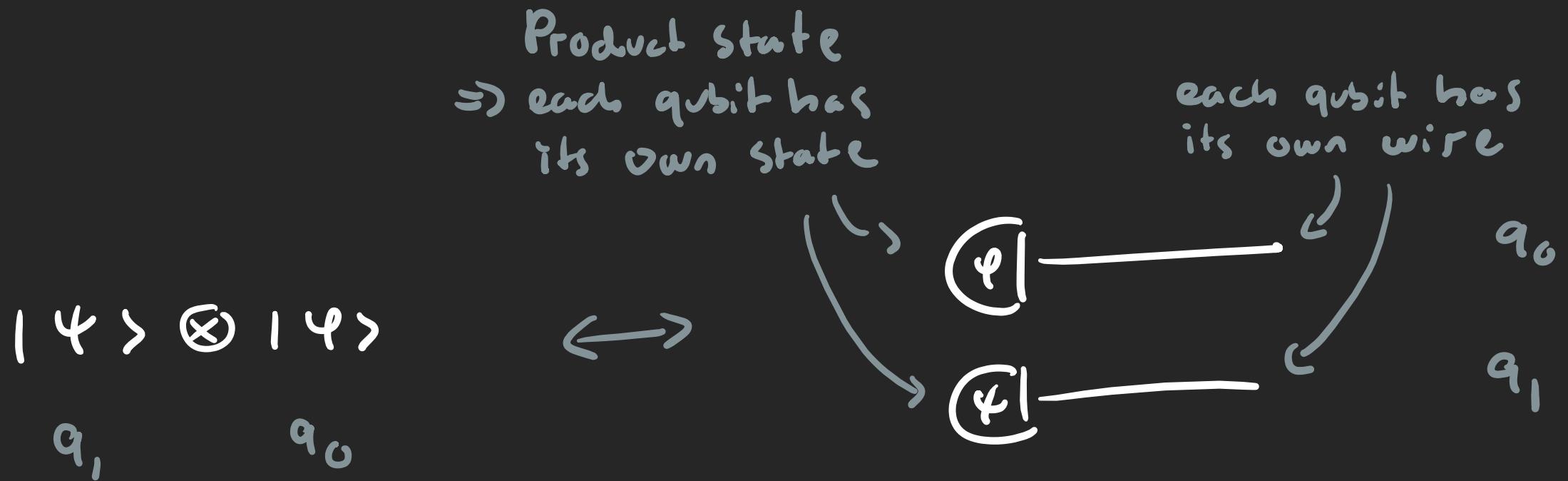
2-qubit Bell state:

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

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

Proof: $|\Phi^+\rangle = |\psi\rangle \otimes |\varphi\rangle$ 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



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.

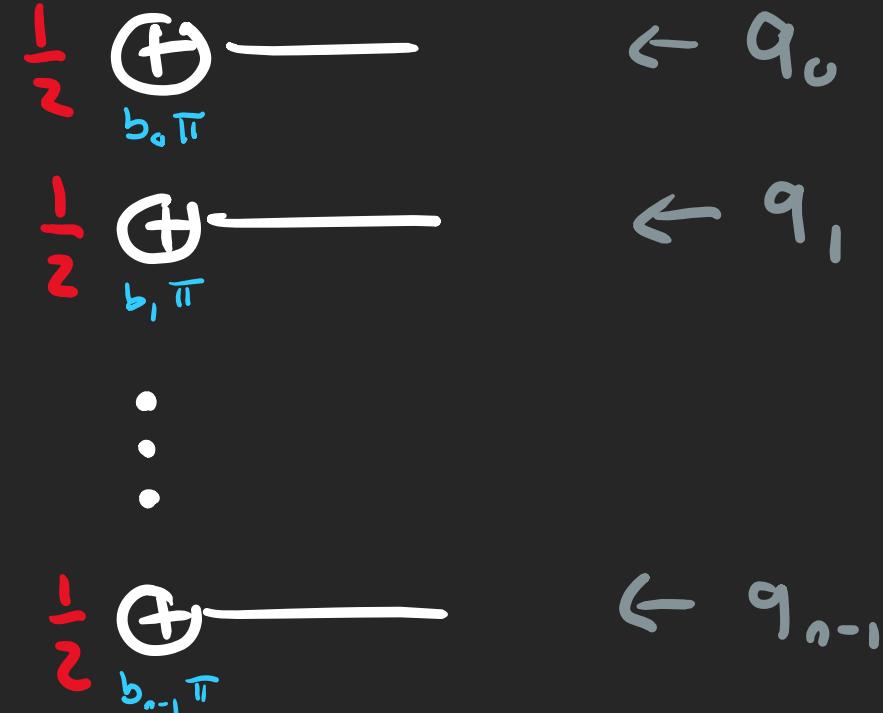
Graphical Notation

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



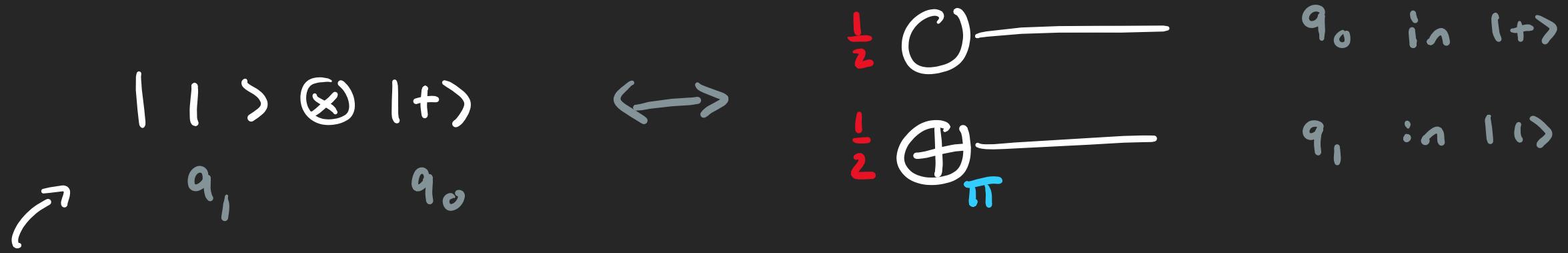
$| \underline{b} \rangle \quad \longleftrightarrow$

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



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

Graphical Notation



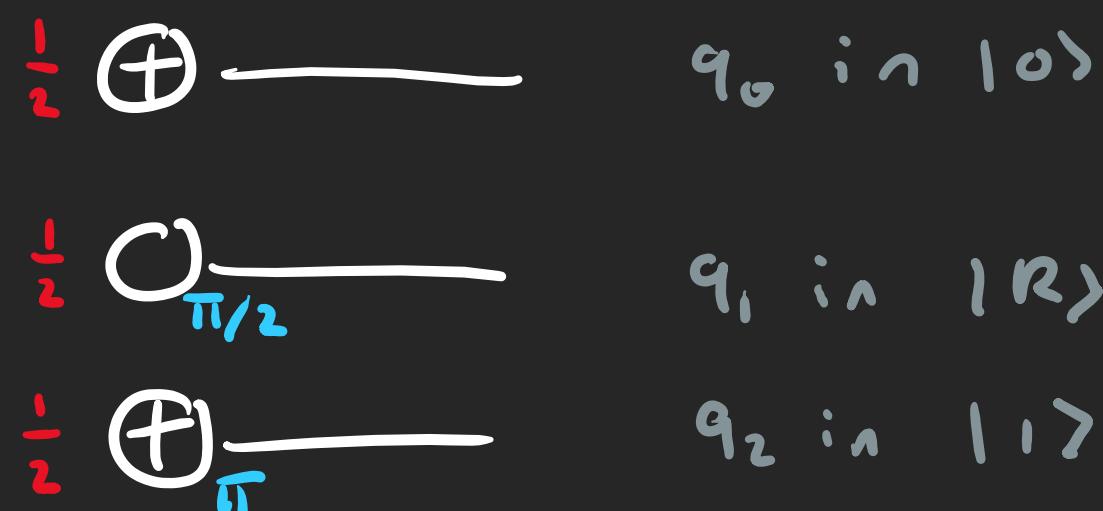
Shorthand: $|1+\rangle$

Shorthand: $|1R0\rangle$



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

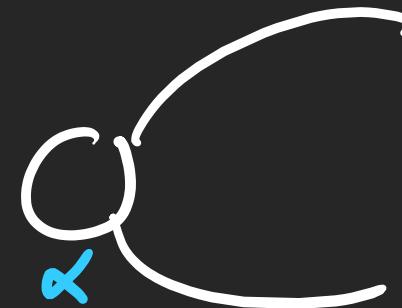
$q_2 \quad q_1 \quad q_0$



Bell State

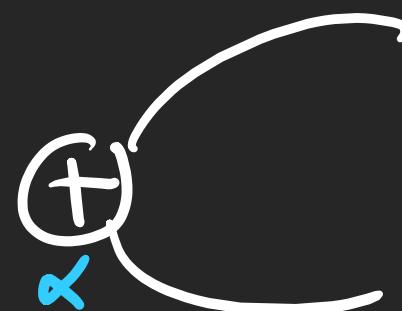
Z basis:

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



X basis:

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

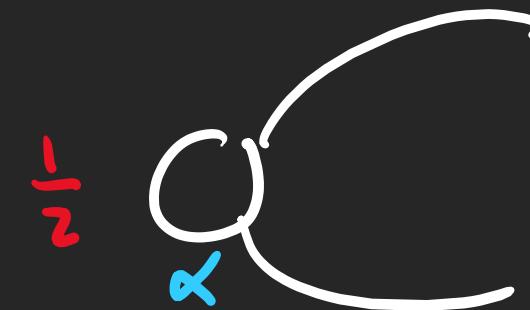


Bell State

Z basis:

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

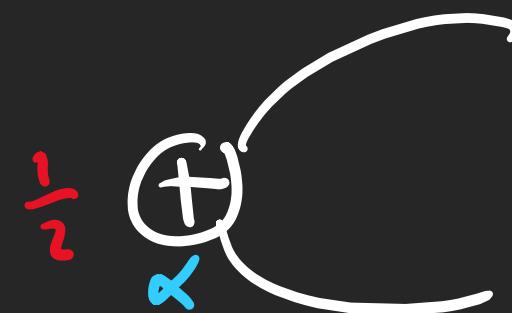
(normalised)



X basis:

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

(normalised)

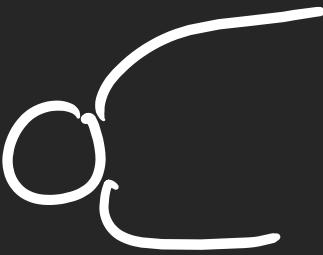


Bell State

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

Bell State

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



But also:

$$\text{Diagram with two overlapping circles} = \text{Diagram with a circle containing a plus sign} \quad (\text{note: } \alpha = 0)$$

Proof: $|++\rangle + |--\rangle = \frac{1}{2}(|00\rangle + |11\rangle)(|00\rangle + |11\rangle) + \frac{1}{2}(|00\rangle - |11\rangle)(|00\rangle - |11\rangle)$

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

Bell State

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \longleftrightarrow \frac{1}{\sqrt{2}} \text{ (open circle)}$$

||

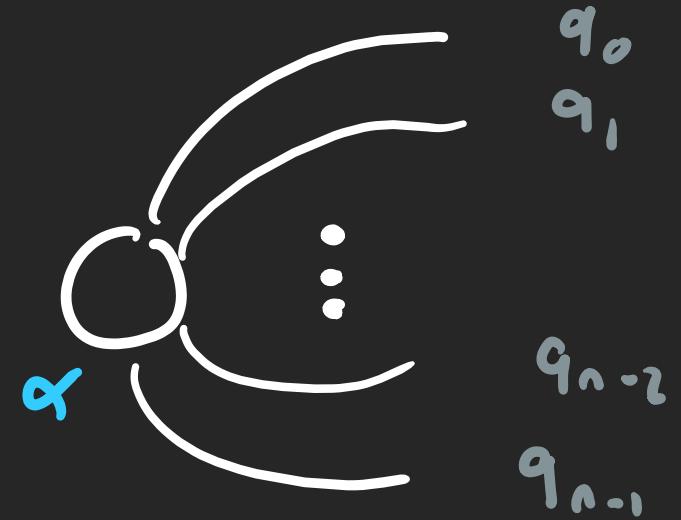
||

$$\frac{1}{\sqrt{2}}(|++\rangle + |--\rangle) \longleftrightarrow \frac{1}{\sqrt{2}} \text{ (circle with +)}$$

GHZ States

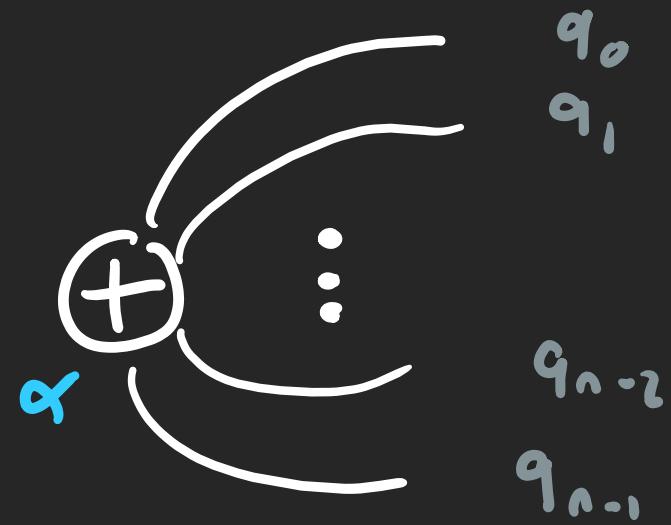
Z basis:

$$|00\dots00\rangle + e^{i\alpha} |11\dots11\rangle \leftrightarrow$$



X basis:

$$|++\dots++\rangle + e^{i\alpha} |--\dots--\rangle \leftrightarrow$$

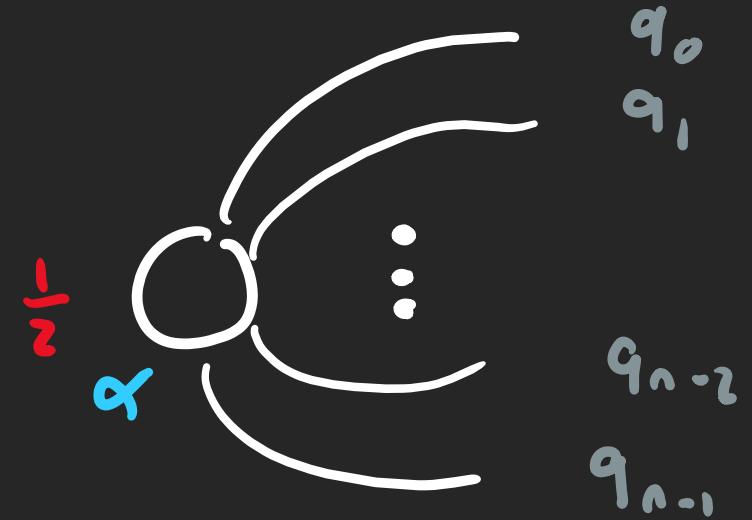


GHZ States

Z basis:

$$\frac{1}{\sqrt{2}} \left(|00\dots00\rangle + e^{i\alpha} |11\dots11\rangle \right) \leftrightarrow$$

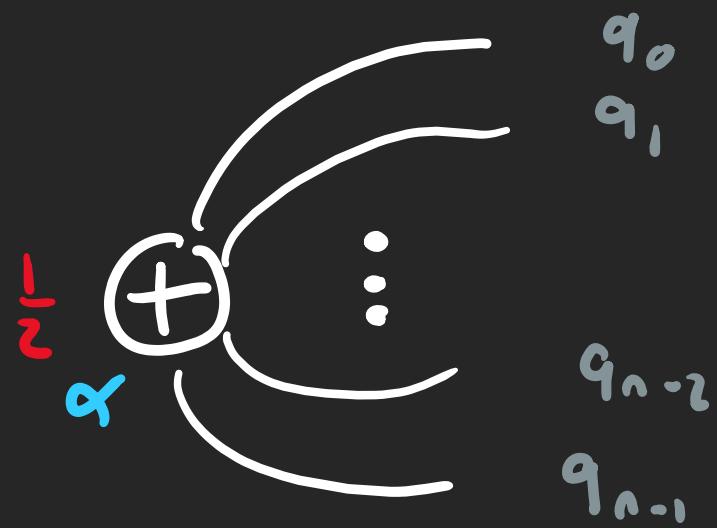
(normalised)



X basis:

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

(normalised)

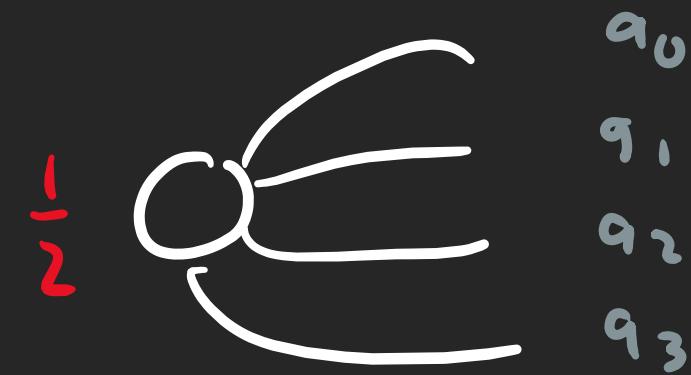


GHZ States

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



$$|GHZ_4\rangle := \frac{1}{\sqrt{2}}(|0000\rangle + |1111\rangle)$$



Braket Notation

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

Transformations of n-qubits:

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

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

Bracket Notation

Outer products:

$$|\underline{b}\rangle \langle \underline{b}'| = \begin{pmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{pmatrix}$$

↑ col \underline{b}'
 ↓
 row \underline{b}

$$|\underline{01}\rangle \langle \underline{10}| = \begin{pmatrix} \langle 001 | & \langle 001 | & \langle 101 | & \langle 111 | \\ |00\rangle & 0 & 0 & 0 \\ |01\rangle & 0 & 0 & 1 \\ |10\rangle & 0 & 0 & 0 \\ |11\rangle & 0 & 0 & 0 \end{pmatrix}$$

Braket Notation

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

Transformations of n-qubits:

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

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

Braket Notation

Product transformations:

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

unitary
 $v_n q_1$

unitary
 $v_n q_0$

each qubit transformed independently

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
 $U \in \mathcal{B}(q_1)$

unitary
 $V \in \mathcal{B}(q_0)$

each qubit transformed independently

Braket Notation

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

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

Braket Notation

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

$$= \begin{pmatrix} U_{00} \Psi_0 \begin{pmatrix} v_{00} & v_{01} \\ v_{10} & v_{11} \end{pmatrix} \begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} + U_{01} \Psi_1 \begin{pmatrix} v_{00} & v_{01} \\ v_{10} & v_{11} \end{pmatrix} \begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} \\ U_{10} \Psi_0 \begin{pmatrix} v_{00} & v_{01} \\ v_{10} & v_{11} \end{pmatrix} \begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} + U_{11} \Psi_1 \begin{pmatrix} v_{00} & v_{01} \\ v_{10} & v_{11} \end{pmatrix} \begin{pmatrix} \varphi_0 \\ \varphi_1 \end{pmatrix} \end{pmatrix}$$

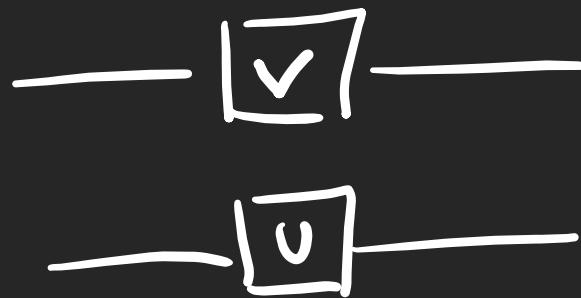
Braket Notation

$$(\textcolor{red}{U} \otimes \textcolor{blue}{V}) (\textcolor{red}{|}\psi\rangle \otimes \textcolor{blue}{|\psi\rangle})$$

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

Graphical Notation

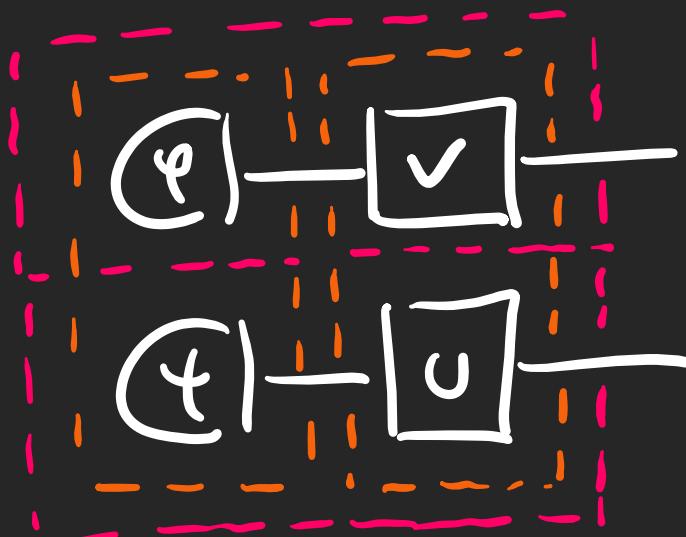
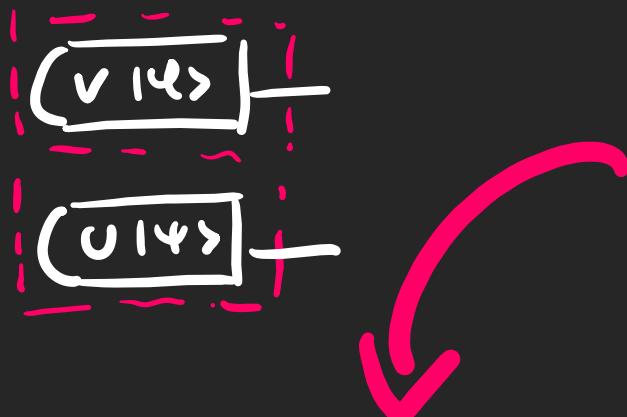
U ⊗ V



q_0
 q_1

Graphical Notation

We get equations "for free" from geometry:



$$(v|\psi\rangle) \otimes (u|\psi\rangle) = (\psi \otimes \psi)(\psi \otimes \psi)$$

Copy Maps

$$|00\rangle\langle 01 + |11\rangle\langle 11 \leftrightarrow \text{---} \circ \text{---}$$

$$\begin{aligned} \frac{1}{2} \oplus \text{---} \circ \text{---} &= (|00\rangle\langle 01 + |11\rangle\langle 11) |10\rangle \\ &= \cancel{|00\rangle\langle 01}^1 + \cancel{|11\rangle\langle 11}^0 = \frac{1}{2} \oplus \text{---} \\ &\qquad\qquad\qquad |00\rangle \end{aligned}$$

Copy Maps

$$|00\rangle\langle 01 + |11\rangle\langle 11 \leftrightarrow \text{---} \circ \text{---}$$

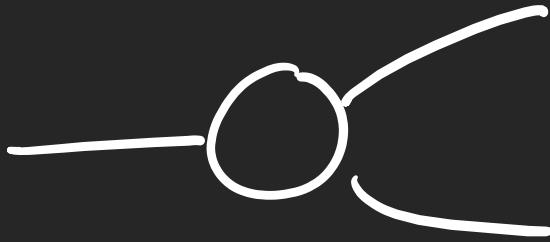
$$\begin{aligned} \frac{1}{2} \oplus \text{---} \circ \text{---} &= (|00\rangle\langle 01 + |11\rangle\langle 11)|1\rangle \\ &= \cancel{|00\rangle\langle 01|1\rangle} + \cancel{|11\rangle\langle 11|1\rangle} = \frac{1}{2} \oplus \text{---} |11\rangle \end{aligned}$$

Copy Maps

$$|00\rangle\langle 0| + |11\rangle\langle 1| \leftrightarrow \begin{array}{c} \text{---} \\ \textcircled{0} \end{array}$$

$$|++\rangle\langle +| + |-+\rangle\langle -| \leftrightarrow \begin{array}{c} \text{---} \\ \textcircled{+} \end{array}$$

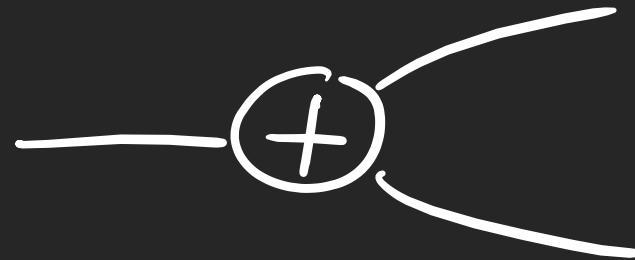
Copy Maps



\mathbb{Z} basis copy

$$\frac{1}{2} \oplus \text{---} = \frac{1}{2} \oplus \text{---}$$
$$\frac{1}{2} \oplus \text{---}$$

$$\frac{1}{2} \oplus \text{---} = \frac{1}{2} \oplus \pi$$
$$\frac{1}{2} \oplus \pi$$



\times basis copy

$$\frac{1}{2} \text{---} \oplus = \frac{1}{2} \text{---}$$
$$\frac{1}{2} \text{---}$$

$$\frac{1}{2} \text{---} \oplus = \frac{1}{2} \pi$$
$$\frac{1}{2} \pi$$

Copy Maps

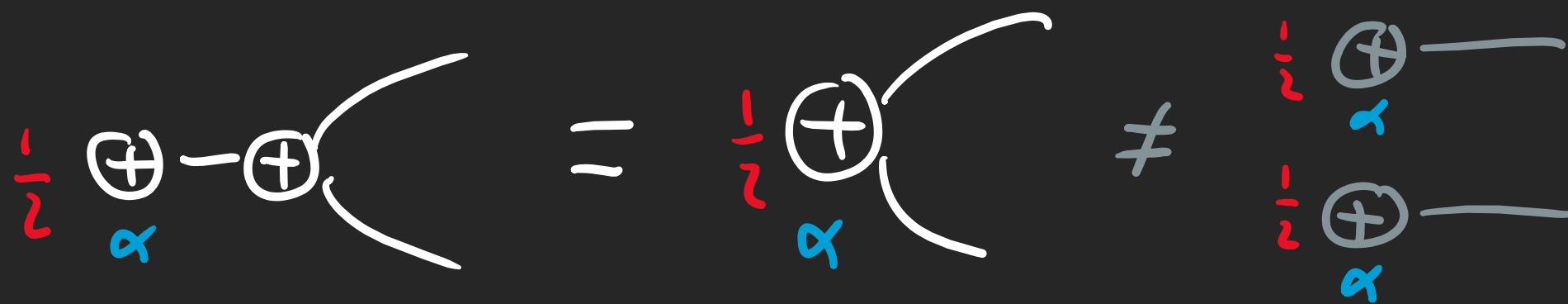
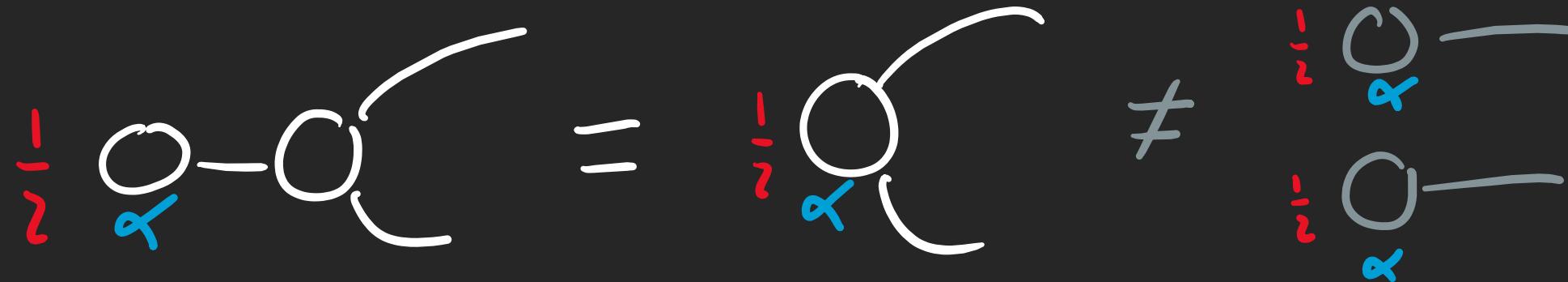
$$\text{Diagram: Two circles connected by a horizontal line, with a blue 'id' label below the left circle.} = (|00\rangle\langle 01 + |11\rangle\langle 11) (|0\rangle + e^{i\alpha}|1\rangle)$$

$$= \cancel{|00\rangle\langle 01} + e^{i\alpha} \cancel{|0\rangle\langle 01}, \\ + \cancel{|11\rangle\langle 11} + e^{i\alpha} \cancel{|1\rangle\langle 11},$$

$$= |00\rangle + e^{i\alpha}|11\rangle = \text{Diagram: One circle with two curved arrows pointing away from it, with a blue 'id' label below the circle.}$$

Copy Maps

BÉVARD! 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

~~Boolean Logic~~ Binary Arithmetic

~~Truth values~~

~~Numbers :~~

$$\mathbb{B} := \{F, T\}$$

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

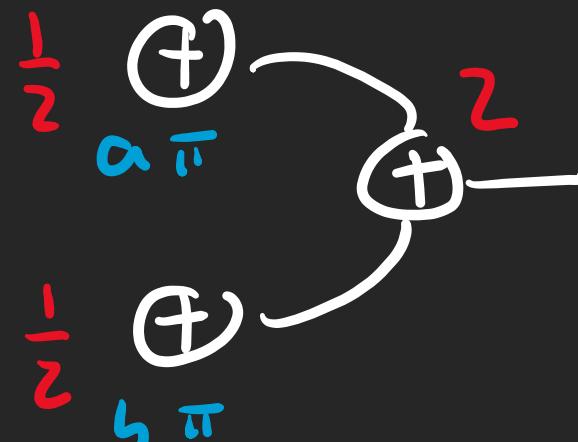
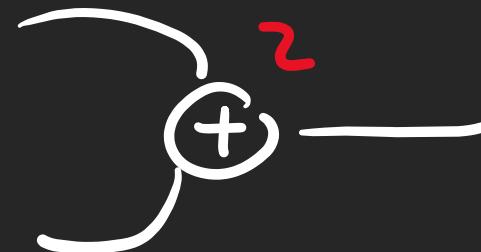
\oplus	F	T
F	F	T
T	T	F

\wedge	F	T
F	F	F
T	F	T

Note: $(a+b)\pi = (a \oplus b)\pi \pmod{2\pi}$

Binary Arithmetic

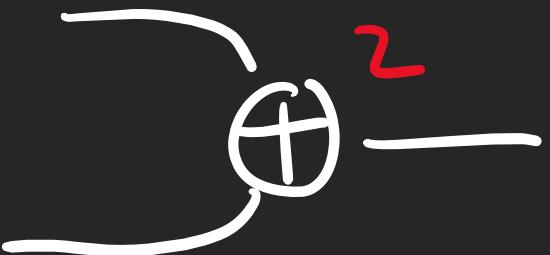
$$\sum_{a,b \in \{0,1\}} |b \oplus a\rangle \langle ba|$$



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

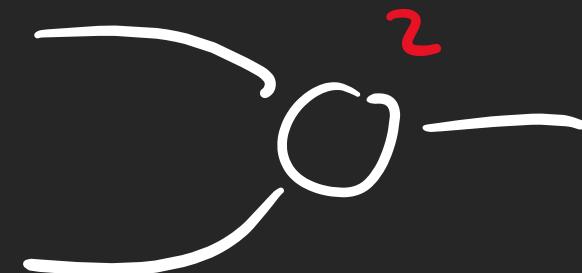
$$= |b \oplus a\rangle = \frac{1}{2} \oplus \begin{matrix} & \\ (a+b)\pi & \end{matrix}$$

Addition Maps



Binary addition (Z basis)

$$\frac{1}{2} \oplus_{a\pi} \oplus^2 - = \frac{1}{2} \oplus_{(a+b)\pi} -$$
$$\frac{1}{2} \oplus_{b\pi} \oplus^2 - = \frac{1}{2} \oplus_{(a+b)\pi} -$$



Binary addition (X basis)

$$\frac{1}{2} \oplus_{a\pi} \leftarrow \oplus^2 - = \frac{1}{2} \oplus_{(a+b)\pi} -$$
$$\frac{1}{2} \oplus_{b\pi} \leftarrow \oplus^2 - = \frac{1}{2} \oplus_{(a+b)\pi} -$$

The NOT Gate

```
circ = QuantumCircuit(1)
circ.x(0) # the NOT gate
circ.draw("mpl")
```

$$a \xrightarrow{\pi} \oplus \quad \gamma a = 1 \oplus a$$



$$\frac{1}{2} \begin{matrix} \oplus \\ \pi \end{matrix} - \begin{matrix} \oplus \\ \pi \end{matrix} - = \frac{1}{2} \begin{matrix} \oplus \\ \pi \end{matrix} -$$

$$\text{NOT } |0\rangle = |1\rangle$$

$$\frac{1}{2} \begin{matrix} \oplus \\ \pi \end{matrix} - \begin{matrix} \oplus \\ \pi \end{matrix} - = \frac{1}{2} \begin{matrix} \oplus \\ \pi \end{matrix} -$$

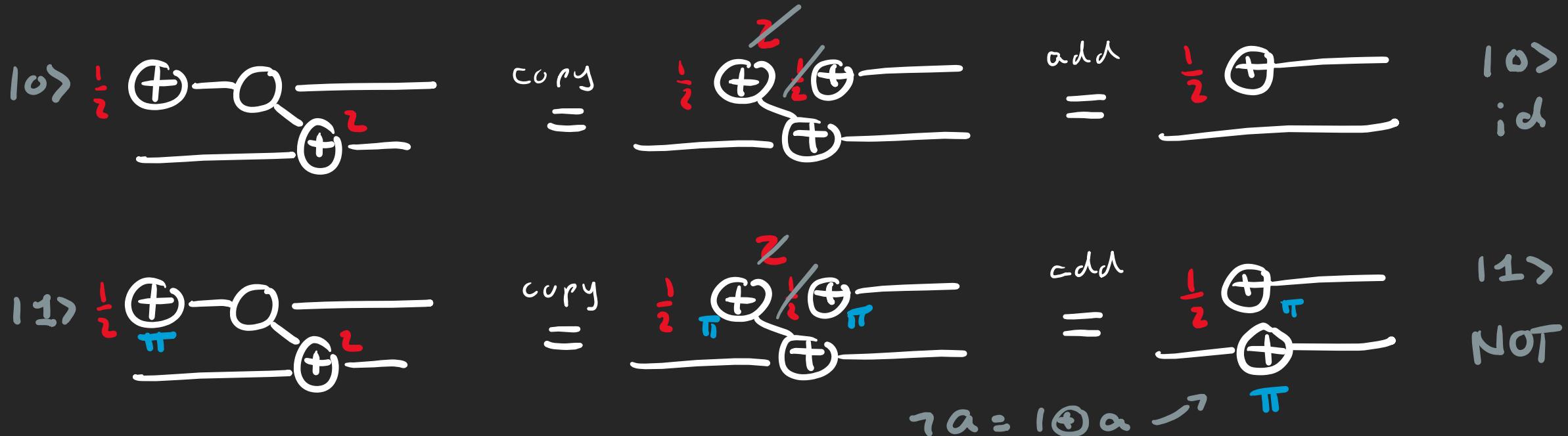
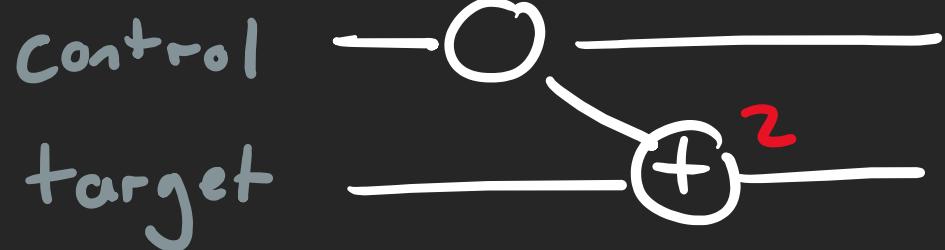
$$\text{NOT } |1\rangle = |0\rangle$$

The NOT Gate eg. $|0\rangle \otimes I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$

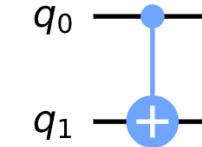
$$\begin{aligned}
 \frac{1}{2} \begin{pmatrix} + \\ b\pi \end{pmatrix}^{\oplus 2} &= \left(\sum_{a', b' \in \{0, 1\}} |b'\oplus a'\rangle \langle b'a'| \right) (|b\rangle \otimes I) \\
 &= \sum_{a' \in \{0, 1\}} |b\oplus a'\rangle \langle a'| \\
 &= \begin{cases} \text{---} & \text{if } b=0 \\ -\frac{1}{\pi} \text{---} & \text{if } b=1 \end{cases} = -\frac{1}{\pi} \begin{pmatrix} + \\ b\pi \end{pmatrix}
 \end{aligned}$$

aubit q_0 ↗
aubit q_1 ↘

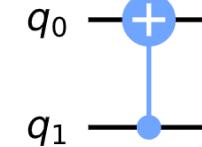
The CNOT Gate



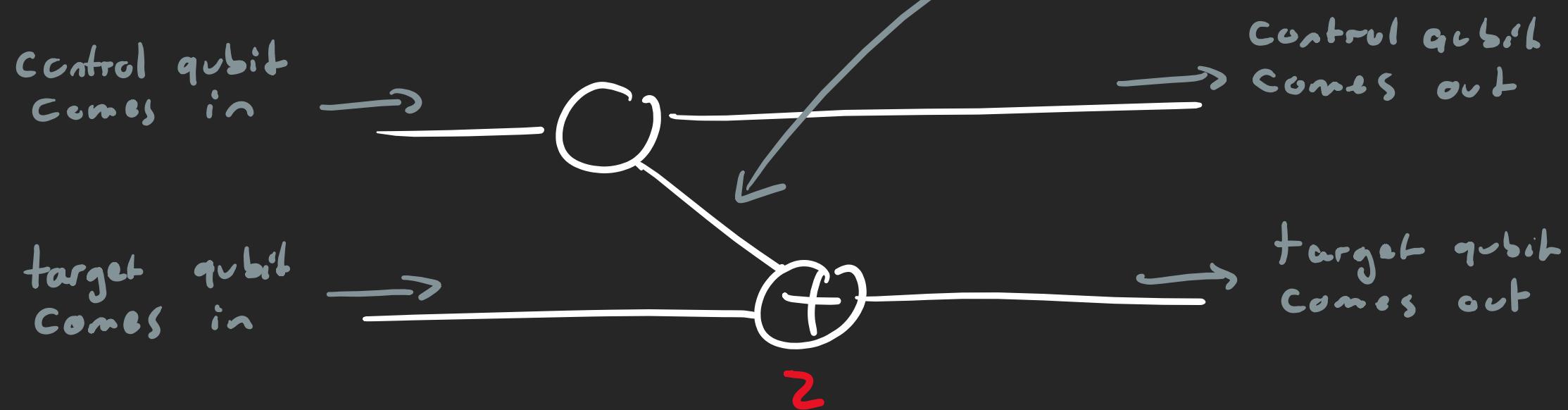
```
circ = QuantumCircuit(2)
circ.cx(0,1) # the CNOT gate
circ.draw("mpl")
```



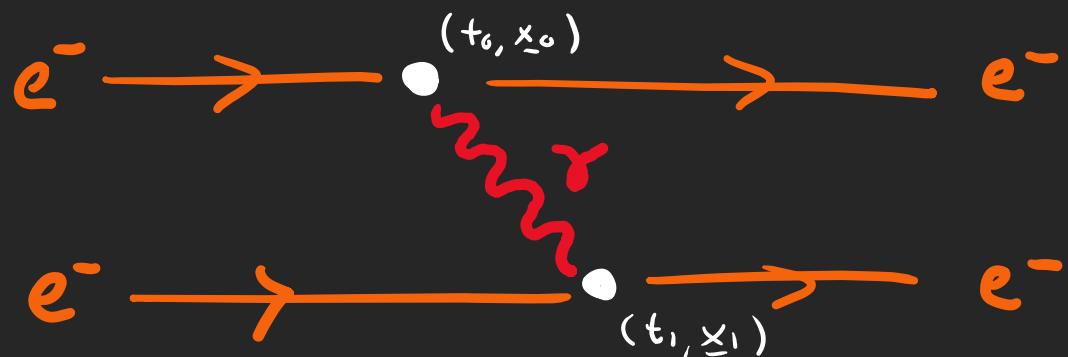
```
circ = QuantumCircuit(2)
circ.cx(1,0) # the CNOT gate
circ.draw("mpl")
```



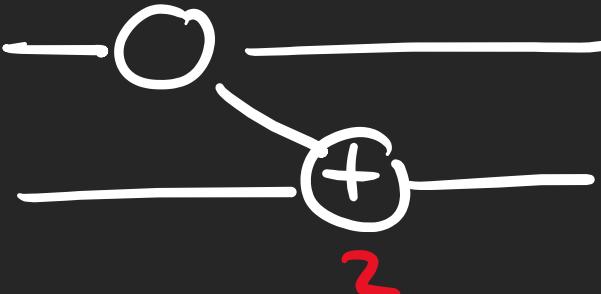
The CNOT Gate



Doesn't look much different from a Feynman Diagram:



The CNOT Gate



$$\begin{array}{c} \frac{1}{2} \oplus \\ \frac{1}{2} \end{array} \otimes \textcircled{C} = \begin{array}{c} \frac{1}{2} \oplus \\ \frac{1}{2} \end{array}$$

\textcircled{Z} $00 \rightarrow 00$

$$\begin{array}{c} \frac{1}{2} \oplus \\ \frac{1}{2} \end{array} \otimes \textcircled{C} = \begin{array}{c} \frac{1}{2} \oplus \\ \frac{1}{2} \end{array}$$

\textcircled{Z} $01 \rightarrow 11$

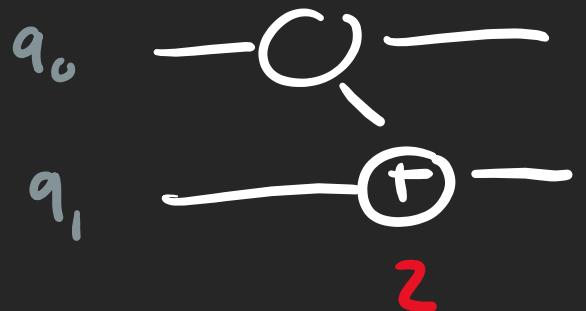
$$\begin{array}{c} \frac{1}{2} \oplus \\ \frac{1}{2} \end{array} \otimes \textcircled{C} = \begin{array}{c} \frac{1}{2} \oplus \\ \frac{1}{2} \end{array}$$

\textcircled{Z} $10 \rightarrow 10$ $\textcircled{\pi}$

$$\begin{array}{c} \frac{1}{2} \oplus \\ \frac{1}{2} \end{array} \otimes \textcircled{C} = \begin{array}{c} \frac{1}{2} \oplus \\ \frac{1}{2} \end{array}$$

\textcircled{Z} $11 \rightarrow 01$ $\textcircled{\pi}$

The CNOT Gate



$$\begin{aligned} & q_1 \quad q_0 \quad q_1 \quad q_0 \\ & \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ & |00\rangle\langle 00| + |11\rangle\langle 01| \\ & + |10\rangle\langle 10| + |01\rangle\langle 11| \end{aligned}$$

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

Diagram illustrating the effect of the CNOT gate on four basis states:

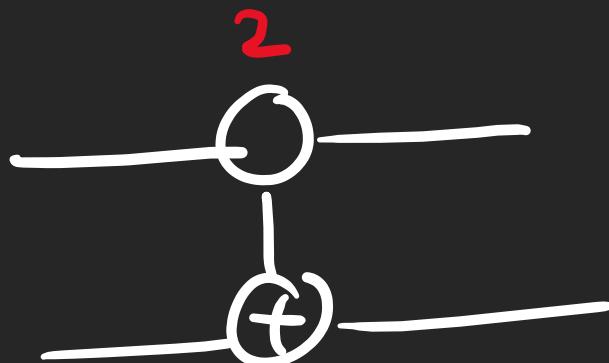
- $|00\rangle \mapsto |00\rangle$
- $|01\rangle \mapsto |10\rangle$
- $|10\rangle \mapsto |10\rangle$
- $|11\rangle \mapsto |01\rangle$

The CNOT Gate

We'll prove later on that :



So the CNOT can be drawn symmetrically :



Exercise

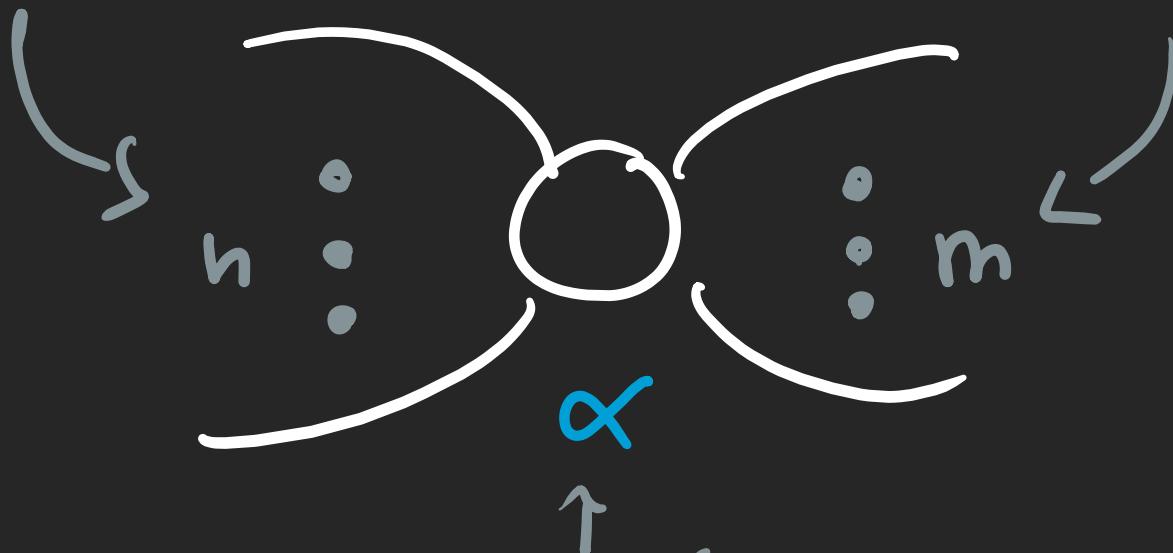
TOP

BALL
BASIS

Spiders

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

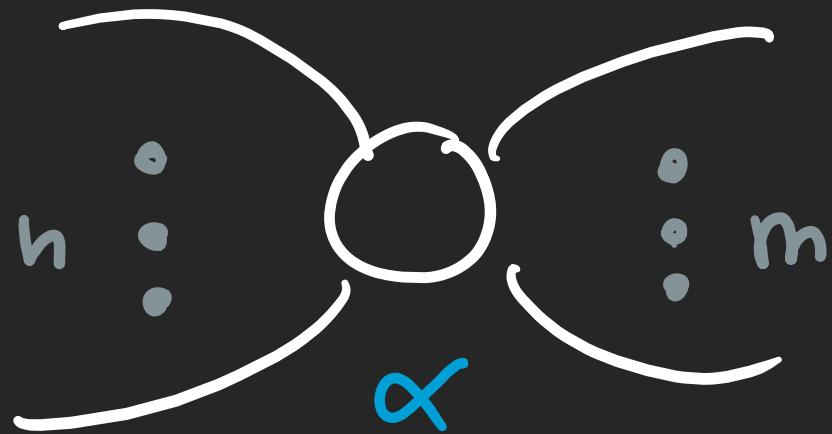
Any number $m \geq 0$ of
output legs / qubits



Any angle (known as the phase)

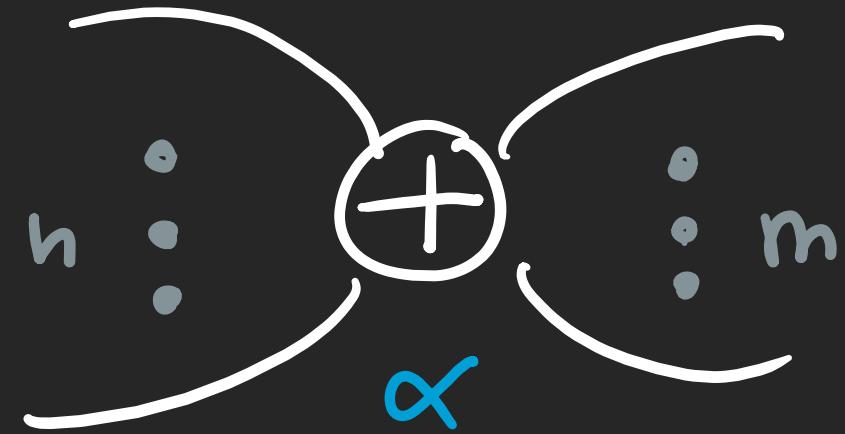
Spiders

Z Spider



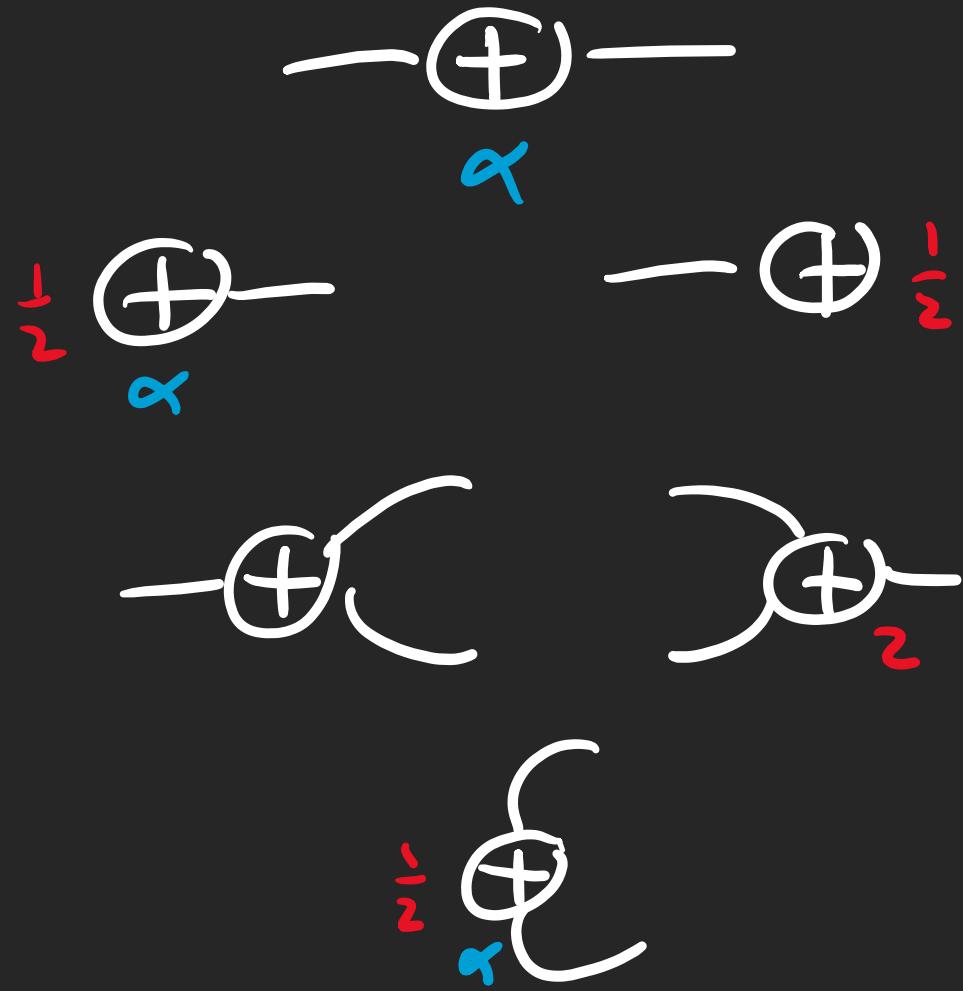
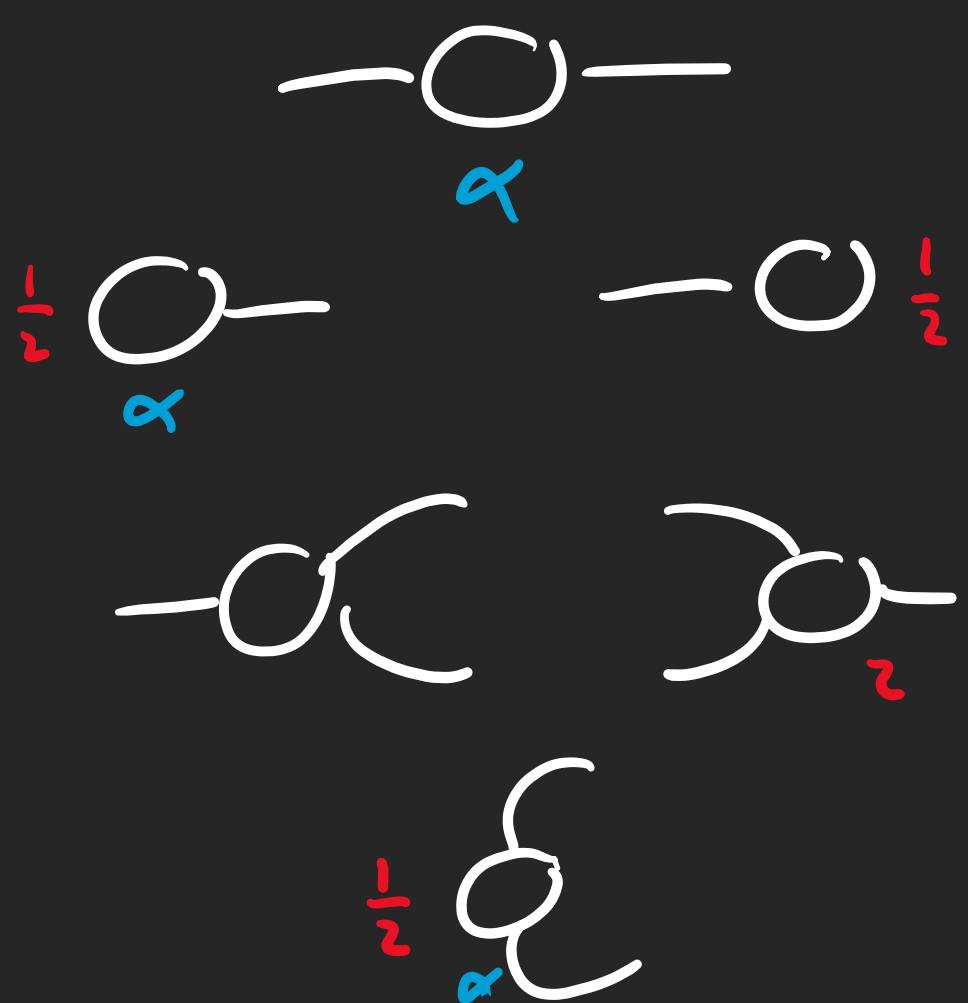
$$|0\dots0\rangle\langle0\dots0| + e^{i\alpha} \underbrace{|1\dots1\rangle\langle1\dots1|}_{m} - \underbrace{|1\dots1\rangle\langle1\dots1|}_{n}$$

X Spider

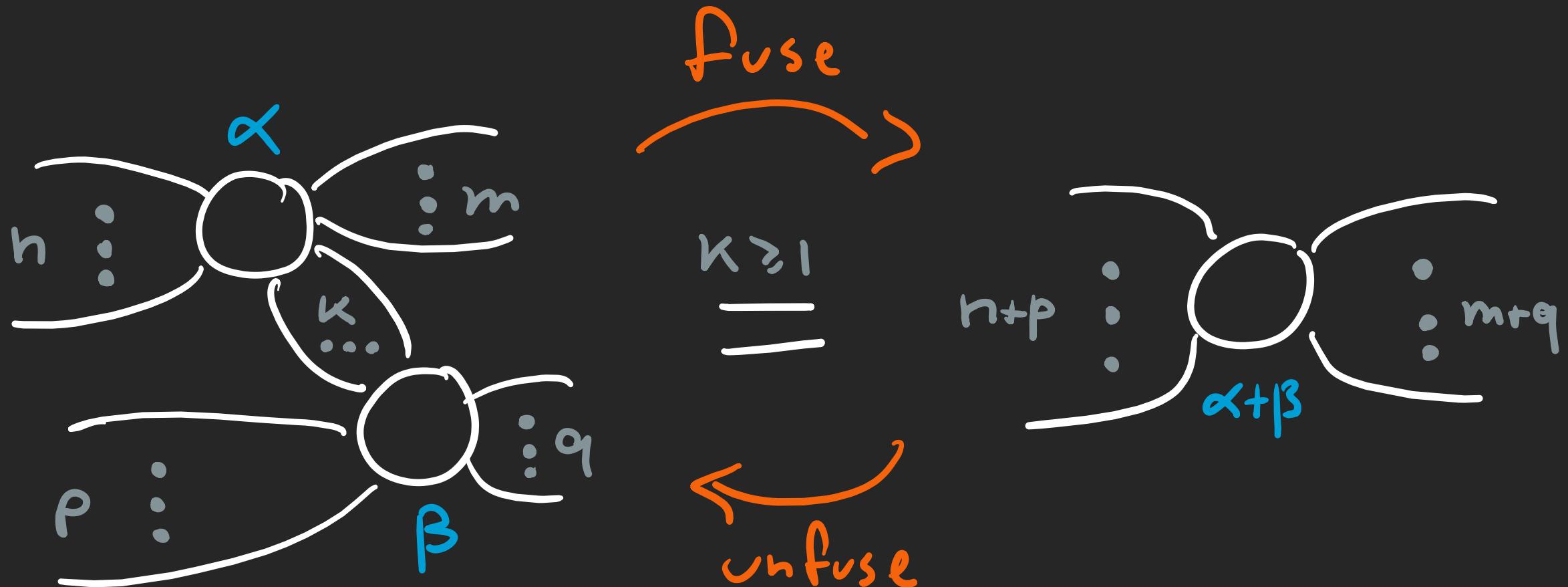


$$\underbrace{|t\dots t\rangle\langle t\dots t|}_{m} - \underbrace{|t\dots t\rangle\langle t\dots t|}_{n} + e^{i\alpha} \underbrace{|-\dots-\rangle\langle-\dots-|}_{m} - \underbrace{|-\dots-\rangle\langle-\dots-|}_{n}$$

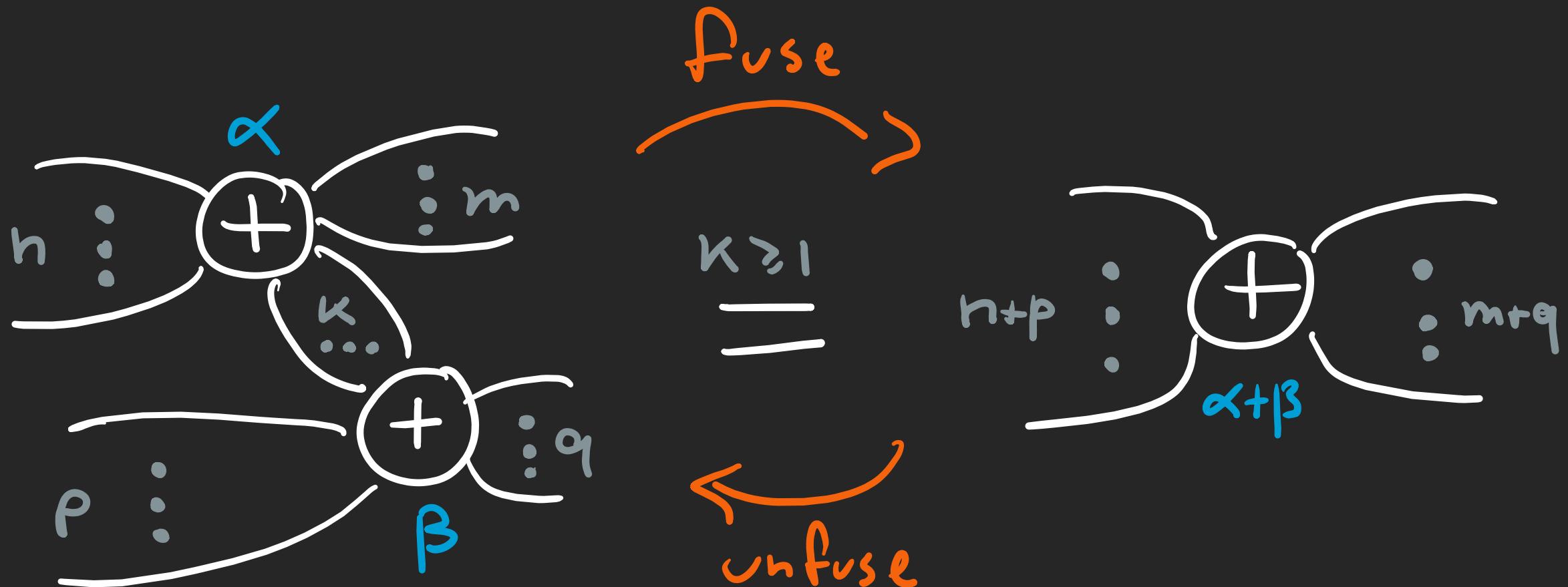
Spiders



Spider Fusion



Spider Fusion



Commutativity

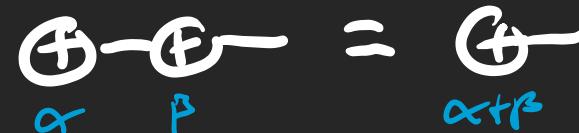
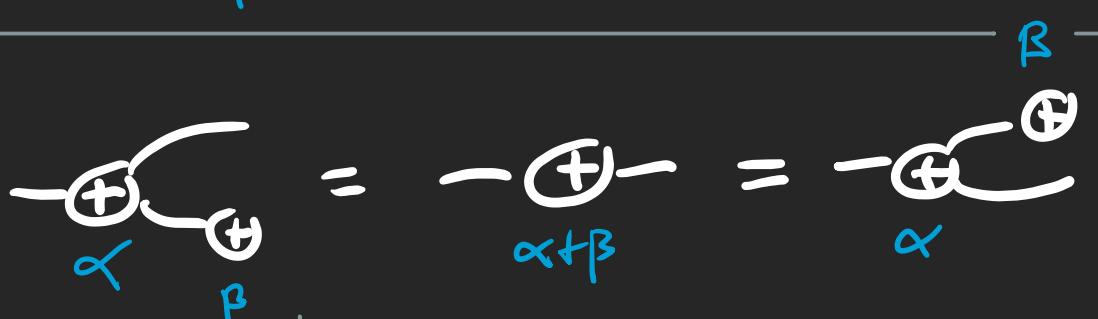
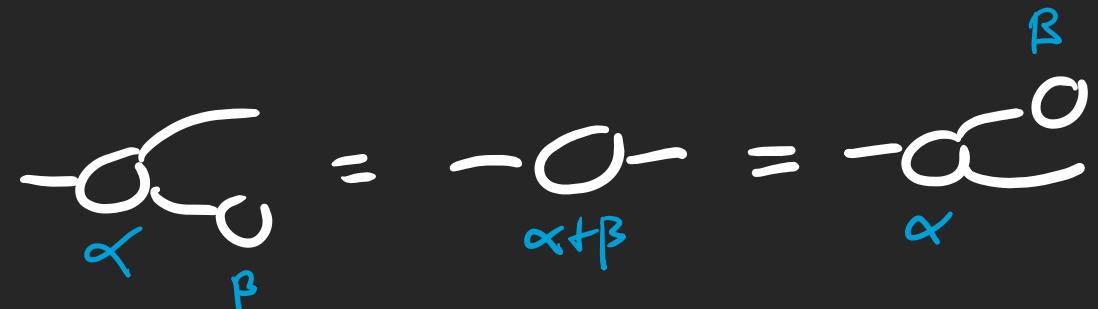
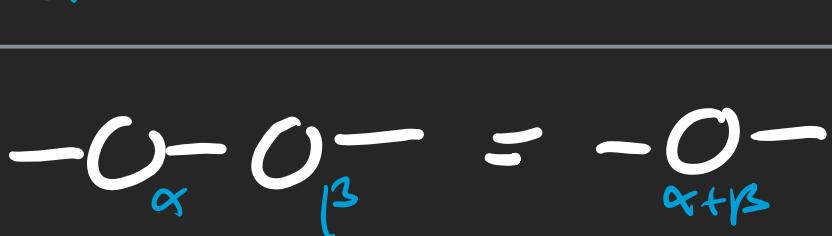
$$* \left\{ \begin{array}{l} x \oplus - = \ominus \\ - \ominus x = - \ominus x \end{array} \right.$$

Leg order in spiders is irrelevant, e.g.

$$\text{fuse } \begin{array}{c} \diagup \quad \diagdown \\ \text{x} \oplus \text{x} \end{array} = \begin{array}{c} \diagup \quad \diagdown \\ \text{x} \oplus \text{x} \end{array} \boxed{\text{x}} \quad * = \begin{array}{c} \diagup \quad \diagdown \\ \text{x} \oplus \text{x} \end{array}$$

$$\text{fuse } \begin{array}{c} \diagup \quad \diagdown \\ \text{x} \oplus \text{x} \end{array} = \begin{array}{c} \diagup \quad \diagdown \\ \text{x} \oplus \text{x} \end{array} \boxed{\text{x}} \quad * = \begin{array}{c} \diagup \quad \diagdown \\ \text{x} \oplus \text{x} \end{array} = \text{fuse } \begin{array}{c} \diagup \quad \diagdown \\ \text{x} \oplus \text{x} \end{array}$$

Spider Fusion



Spider Fusion

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

$$\text{--} \circ \text{--} = \text{--} \circ \text{--} \circ \text{--} = \text{--} \circ \text{--} \quad \text{"}\times\text{"}$$

$$\text{--} \oplus \text{--} = \text{--} \oplus \text{--} \oplus \text{--} = \text{--} \oplus \text{--} \quad \text{"}\times\text{"}$$

$$\text{--} \alpha \text{--} = \text{--} \quad \text{--} \oplus \text{--} = \text{--}$$

$$\text{--} \oplus \text{--} = \text{--} \quad \text{--} \oplus \text{--} = \text{--}$$

Frobenius Law

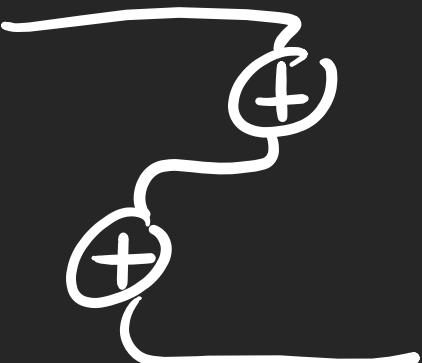
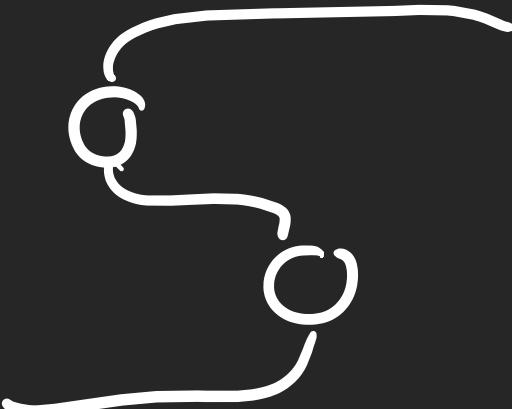
Isometry Law

Snake Equations



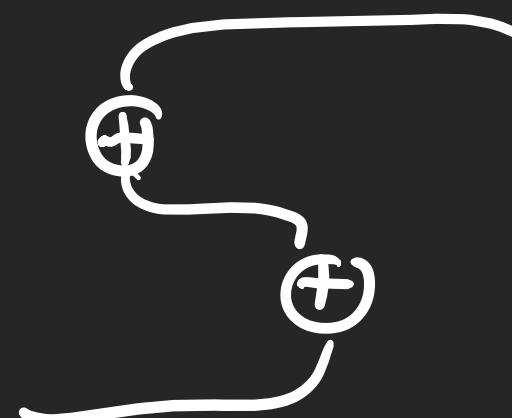
fusion

=



fusion

=



Snake Equations

The diagram shows two snakes, each with a head and a body ending in a tail. They are positioned above a horizontal line with two equals signs. Above the first equals sign is the word "fusion". Above the second equals sign is the word "fusion". Below the second equals sign is a pink handwritten note: "Spider fusion is a really powerful tool...".

$|4\rangle \quad (|00\rangle + |11\rangle) \otimes |4\rangle \quad \left(\sum_b |b\rangle (\langle b_{00}| + \langle b_{11}|) \right) \left(\sum_{b_1} (|00b_1\rangle + |11b_1\rangle) t_{b_1} \right) =$

A pink stick figure with its arms raised in excitement, with three exclamation marks (!!!) floating above its head.

Snake Equations

$\langle 00 \rangle + \langle 11 \rangle$

$\langle ++ \rangle + \langle -- \rangle$

$\langle \omega \rangle + \langle 11 \rangle$

$\langle ++ \rangle + \langle -- \rangle$

$$C := \text{cup} = \text{cap}$$

$$D := \text{cap} = \text{cup}$$

the cup

the cap

$$\text{Z} = \text{---} = \text{S}$$

Snake Equations

The Transpose

$$\left(\begin{array}{c|c} \vdots & \\ \hline n & U \\ \vdots & m \end{array} \right)^T = \text{Diagram showing the transpose of a matrix. The original matrix has dimensions } n \times m. \text{ The transpose is shown as a row vector where the columns of the original matrix are stacked vertically. The diagram uses arrows and labels to indicate the mapping from columns to rows. The top row of the original matrix is labeled } n \text{ and the bottom row is labeled } m. \text{ The rightmost column of the original matrix is labeled } m \text{ and the leftmost column is labeled } n. \text{ Ellipses indicate additional columns and rows.}$$

$$\sum_{\hat{b}, b} |b\rangle \langle U_{\hat{b}, b}| \hat{b}' \mapsto \sum_{\hat{b}, b} |\hat{b}'\rangle \langle b| U_{\hat{b}, b}$$

The Transpose

$$(\text{---})^T = \text{---} = -\text{---}$$

$$(-\text{---})^T = \text{---} = \text{---}$$

$$(-\text{---})^T = \text{---} = -\text{---}$$

The Transpose

$$(\oplus)^\top = \cancel{\oplus} = -\oplus$$

$$(-\ominus)^\top = \cancel{\ominus} = +\ominus$$

$$(-\oplus)^\top = \cancel{\oplus} = -\ominus$$

The Transpose

$$(-\infty)^\top = \text{[hand-drawn diagram of a complex loop]} =$$

$\mathcal{D} = \{p, c, q\}$
 $\underbrace{\quad}_{\text{---}}$
=



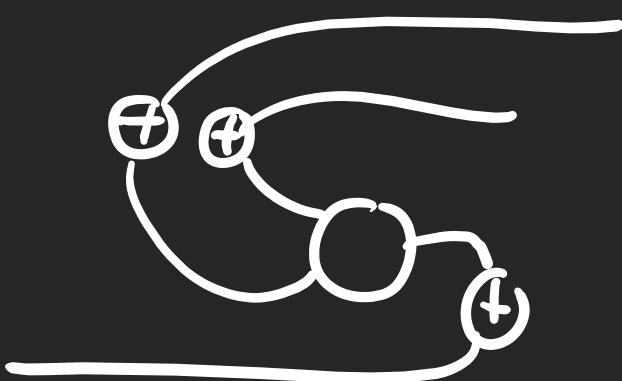
Fusion
=



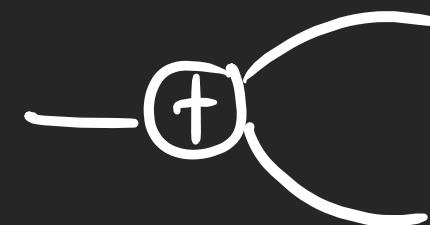
The Transpose

$$\left(\begin{array}{c} \text{X} \\ \text{O} \end{array} \right)^T = \begin{array}{c} \text{X} \\ \text{O} \end{array} = \begin{array}{c} \text{X} \\ \text{O} \end{array}$$

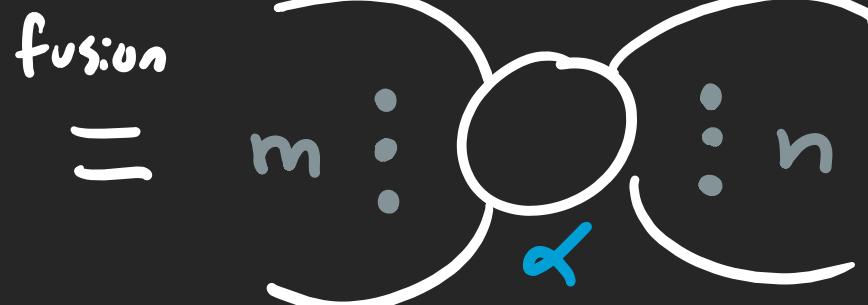
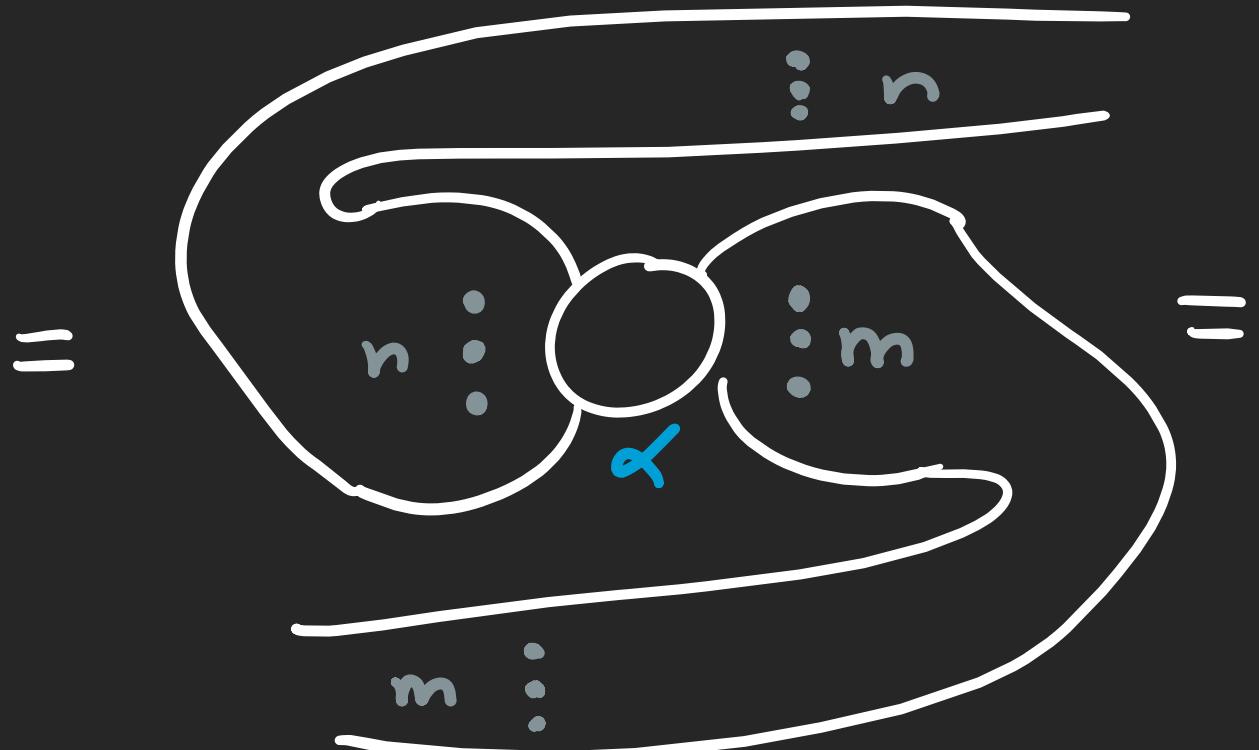
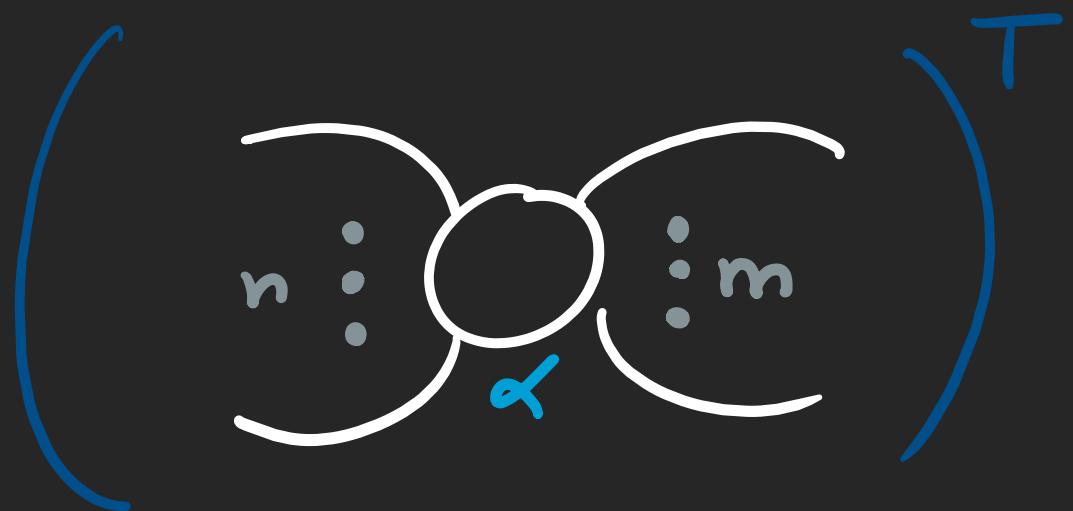
$\mathcal{D} = \{ \mathcal{S}, \mathcal{C}, \mathcal{E} \}$

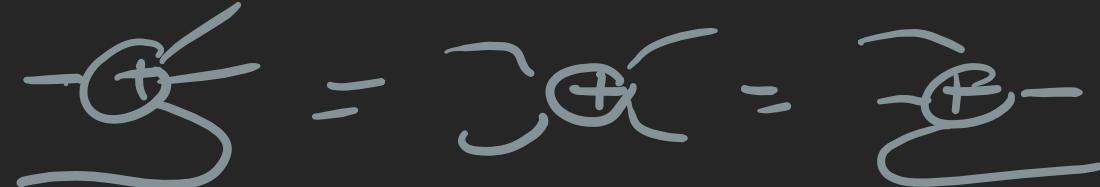


Fusion =



The Transpose



The Transpose e.g. 

We can also perform a partial transpose :

$$\text{---} \circlearrowleft = \text{---} \alpha$$

$$-\alpha = \text{---} \circlearrowright$$

$$\text{---} \oplus \circlearrowleft = \text{---} \oplus \alpha$$

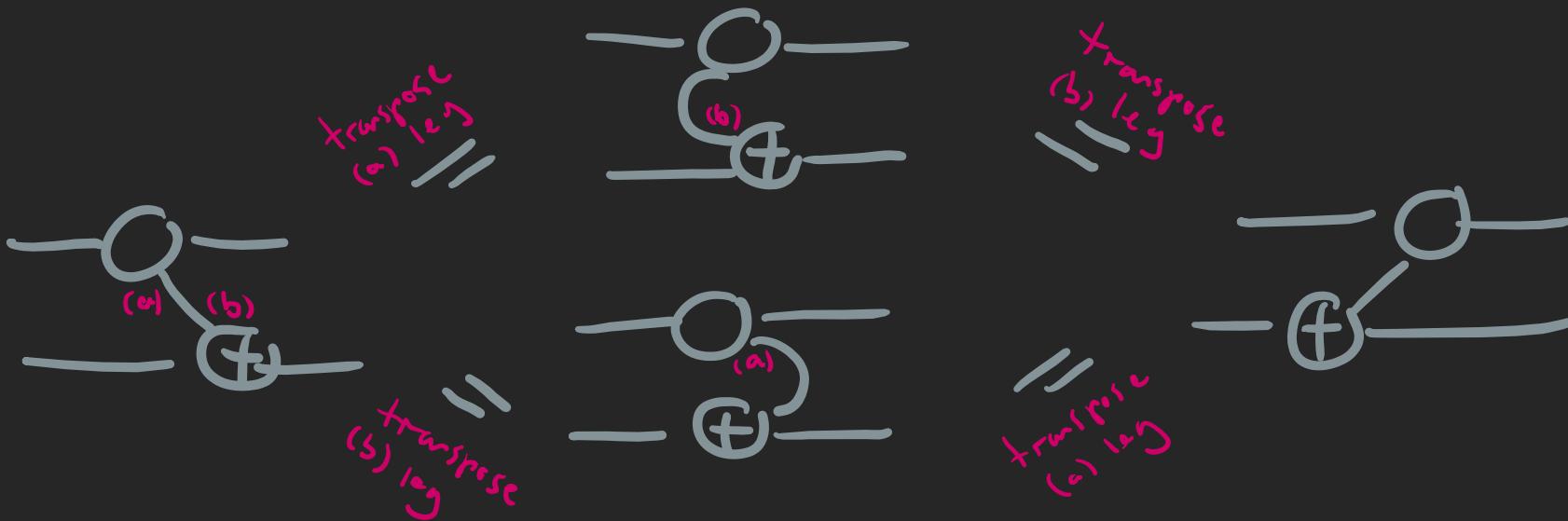
$$-\oplus \alpha = \text{---} \circlearrowright \oplus$$

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

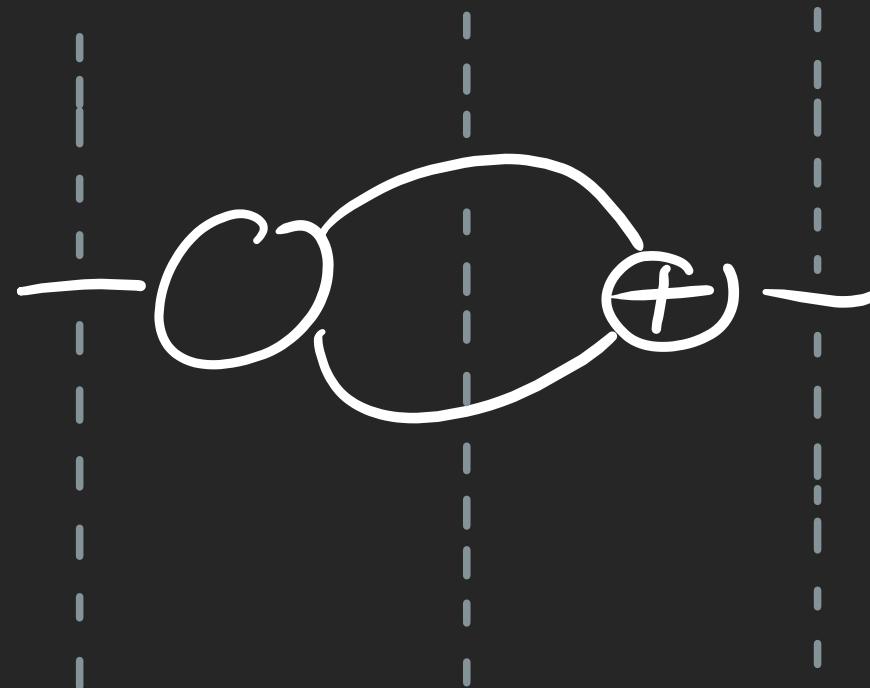
The CNOT Gate



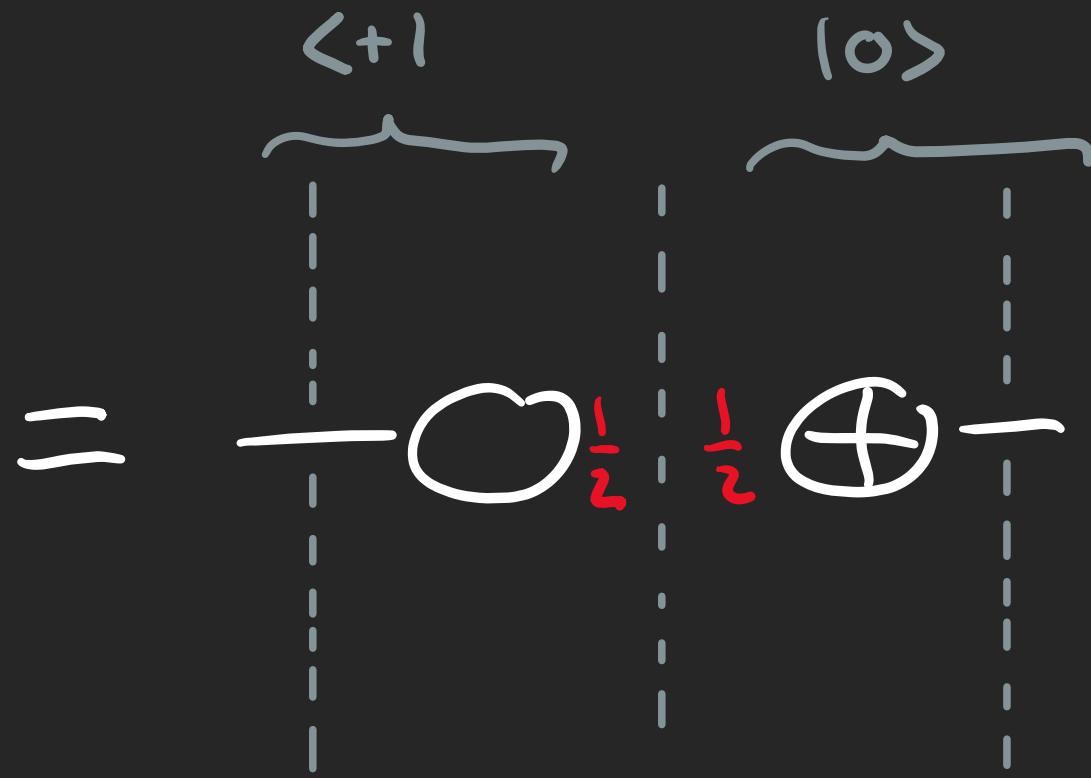
Proof



The Hopf Law



$$|b\rangle \quad |b\rangle \quad \frac{1}{\sqrt{2}} |b\&b\rangle \\ = \frac{1}{\sqrt{2}} |0\rangle$$



$$|b\rangle \quad \langle +|b\rangle \quad \frac{1}{\sqrt{2}} |0\rangle \\ = \frac{1}{\sqrt{2}}$$

The Hopf Law

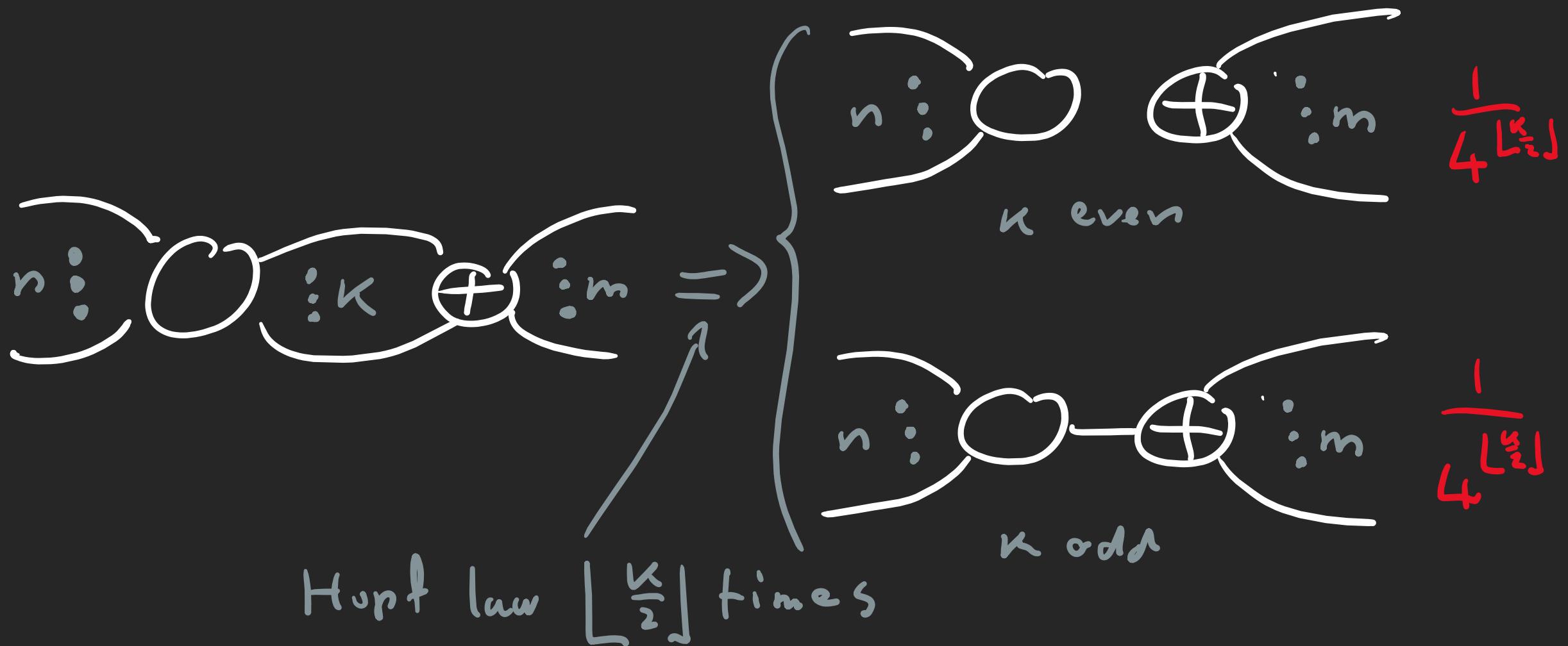
$$\text{Diagram 1} = \frac{1}{4} \text{Diagram 2}$$

Diagram 1: A diagrammatic equation showing two configurations of a loop with legs. The left side shows a loop with three vertices. The top vertex has a dot labeled $\vdots k$. The bottom-right vertex has a dot labeled $\vdots m$. The bottom-left vertex has a dot labeled $\vdots n$. A white circle with a plus sign (+) is at the center. Two red curved arrows point from the bottom-left vertex to the center circle. The right side of the equation is $\frac{1}{4}$ times Diagram 2.

Diagram 2: Similar to Diagram 1, but the red curved arrows point from the center circle to the bottom-left vertex.

\leftarrow pair of legs
is chopped up

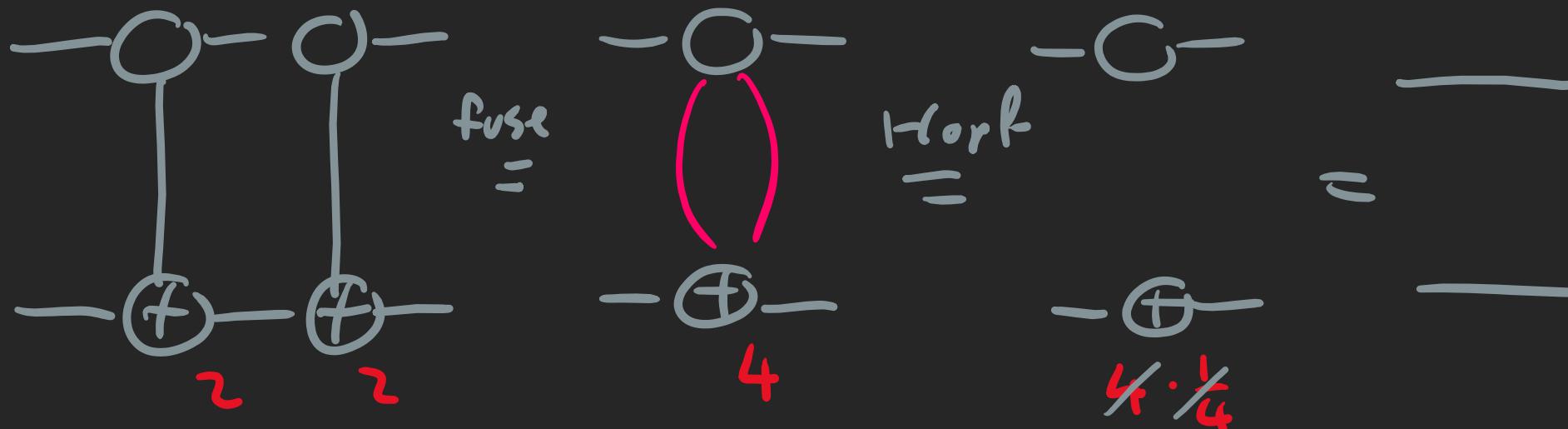
The Hopf Law



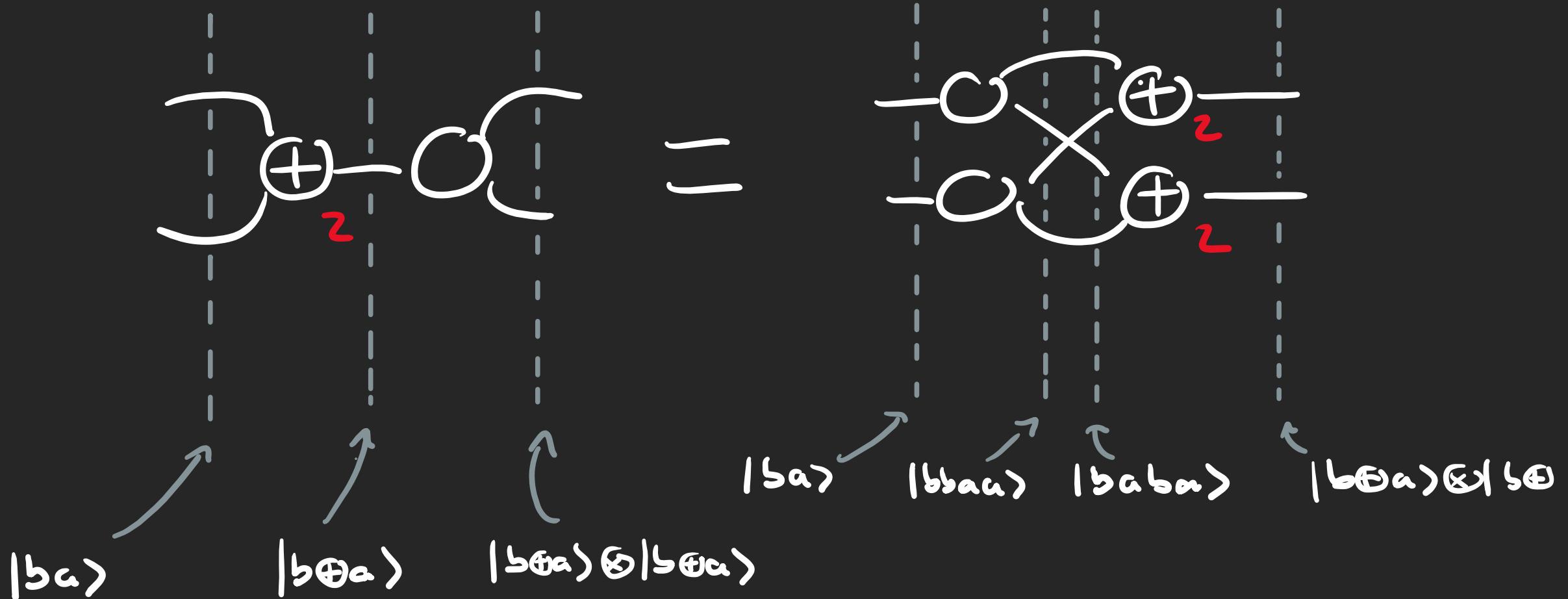
The CNOT Gate



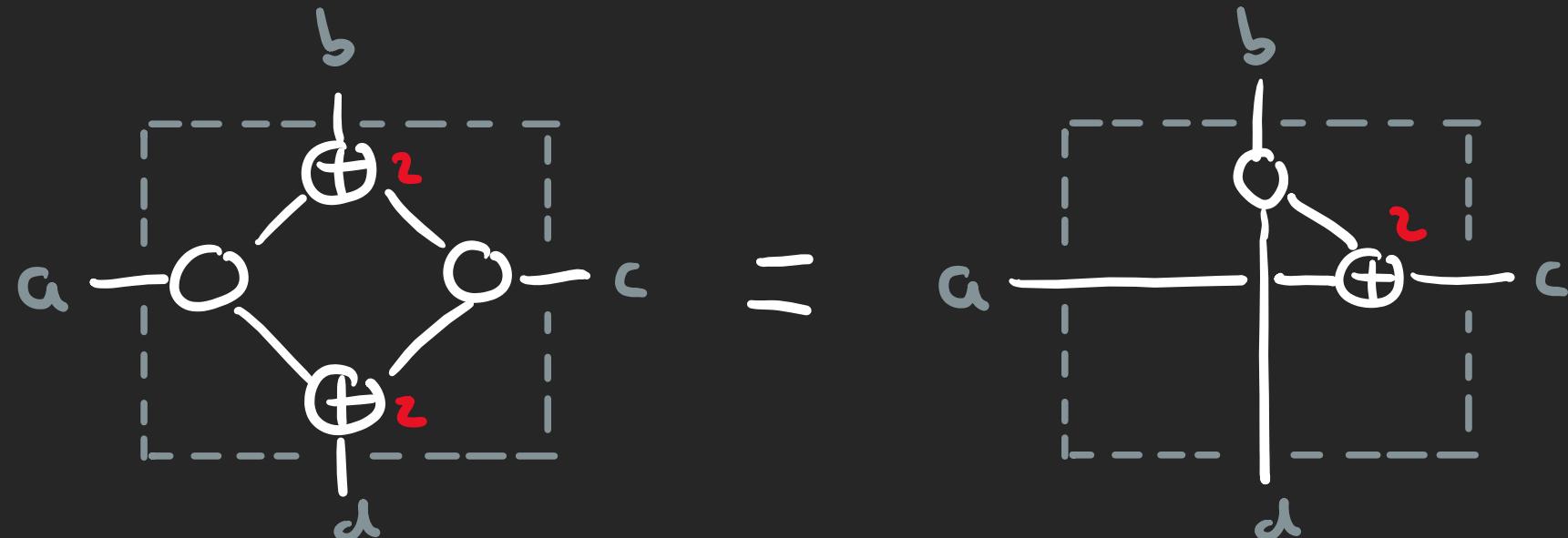
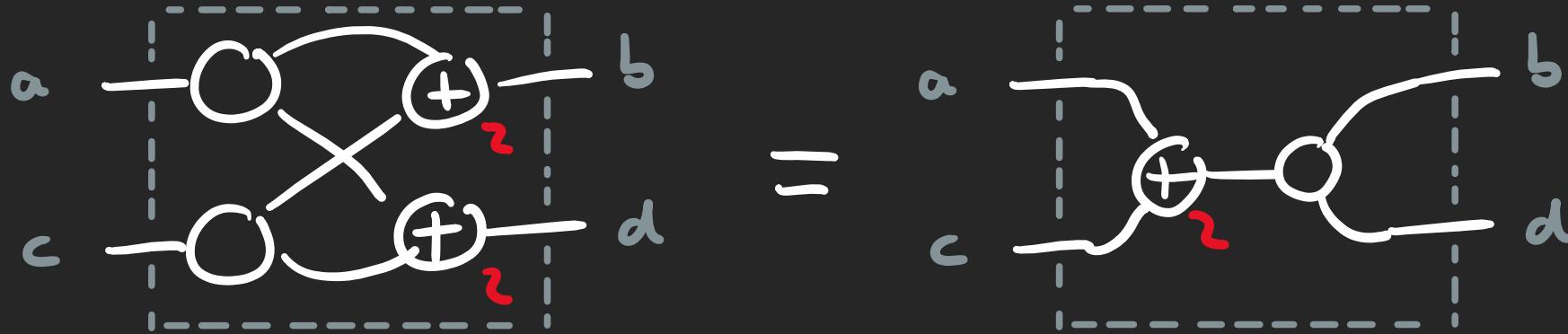
Proof:



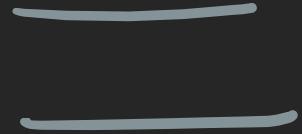
The Bialgebra Law

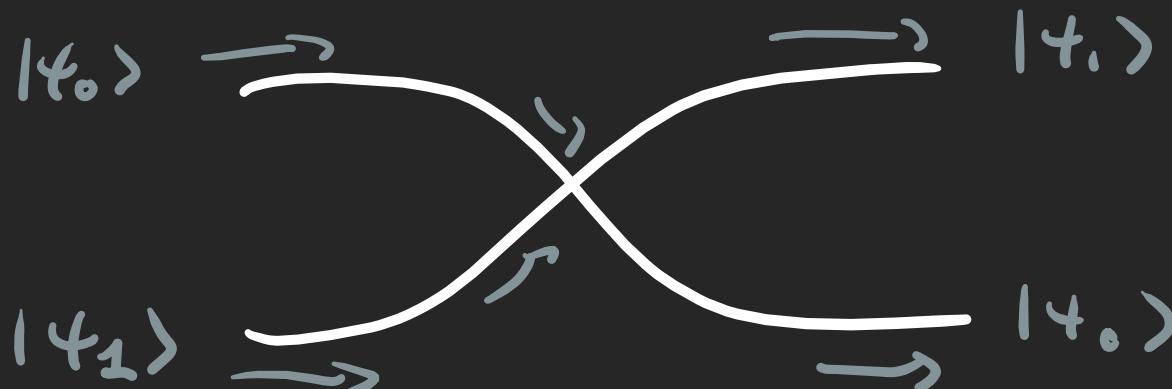
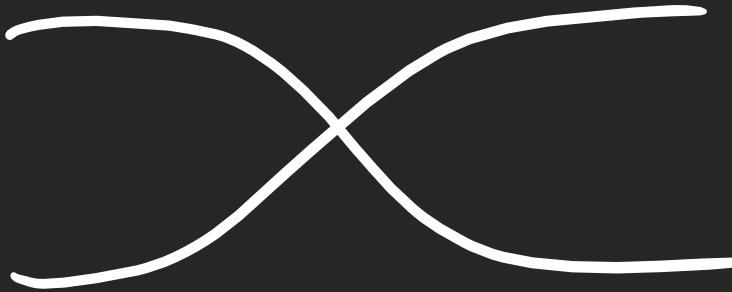


The Bialgebra Law

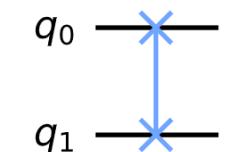


The SWAP Gate

 = 



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

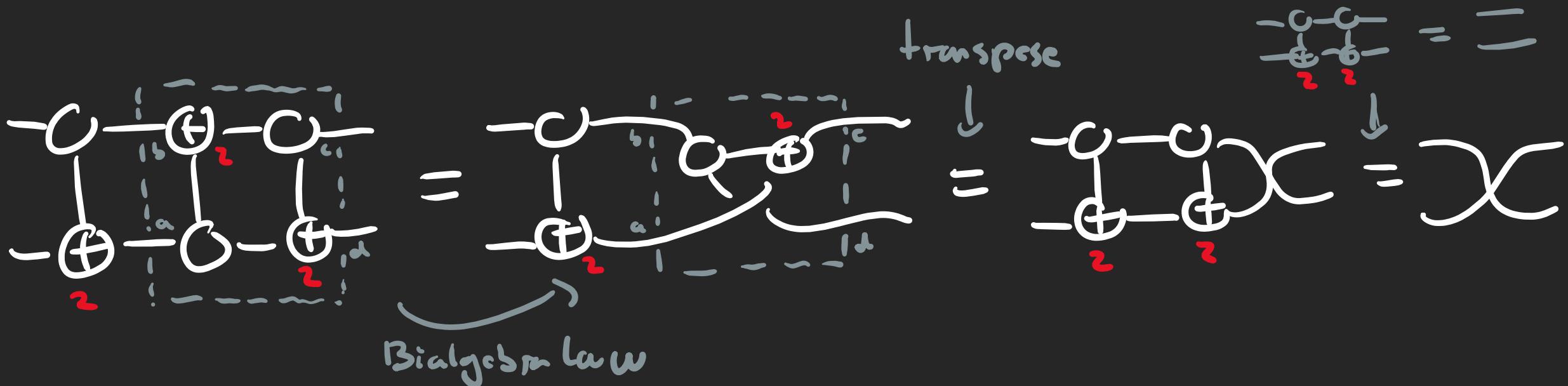


The SWAP Gate

This is how SWAP is actually implemented in QC!



Proof:



The π -commutation Rule

$$\begin{array}{c} \alpha \\ -\oplus- \\ \alpha\pi \end{array} = \begin{array}{c} \alpha \\ -\circ- \\ \alpha\pi \end{array} + \begin{array}{c} \alpha \\ -\oplus- \\ \alpha\pi \end{array}$$

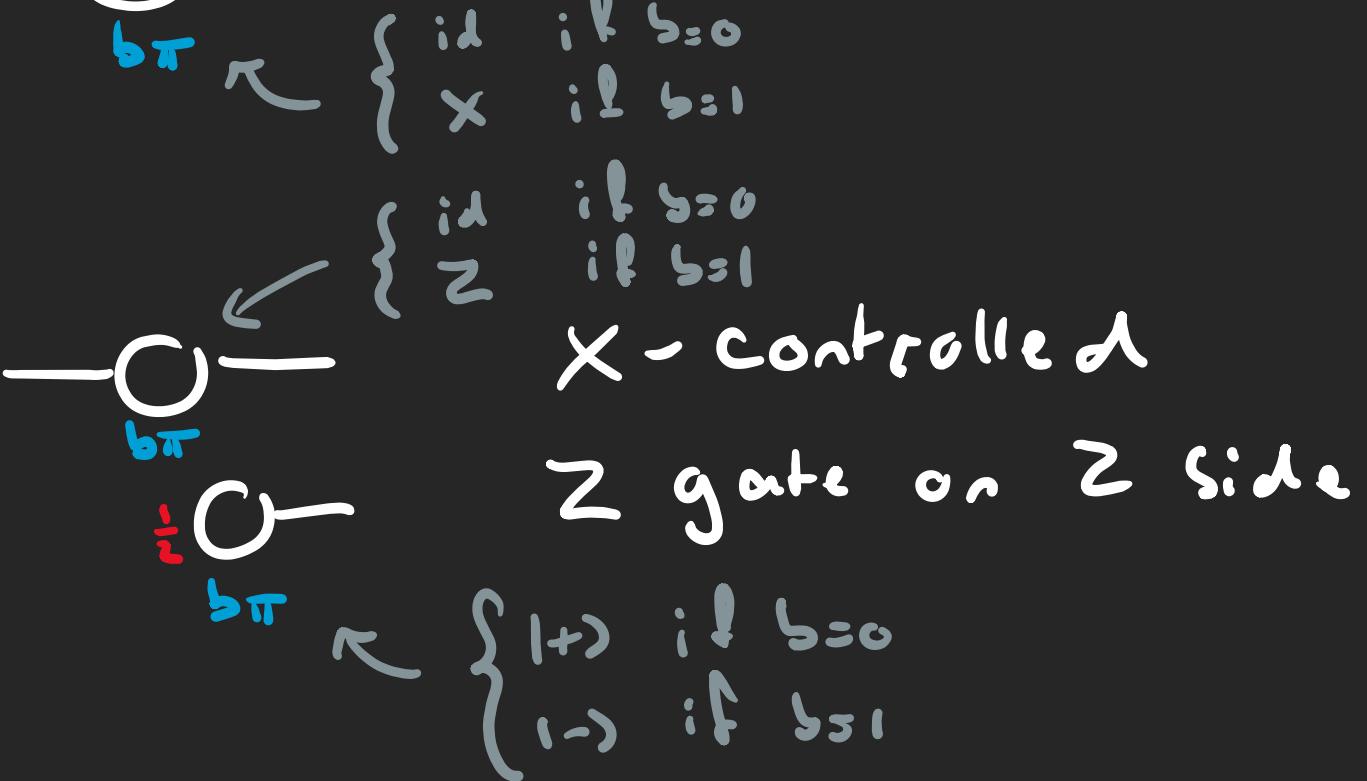
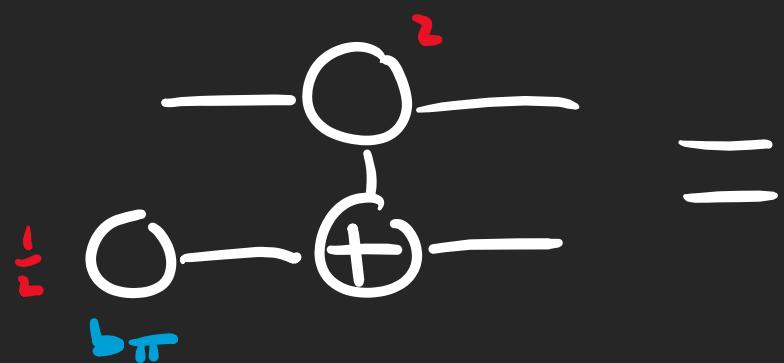
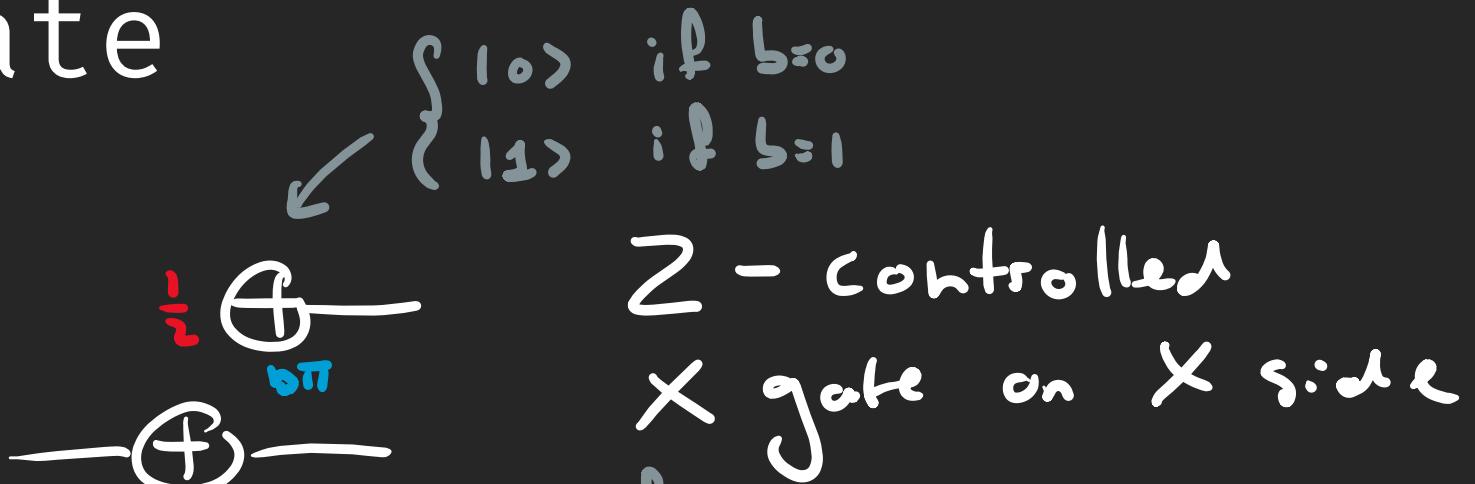
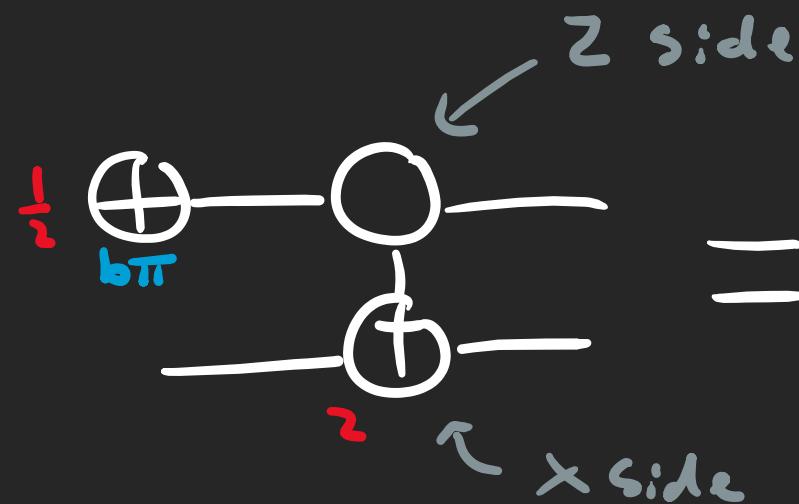
$\alpha \in \{0, 1\}$

Proof:

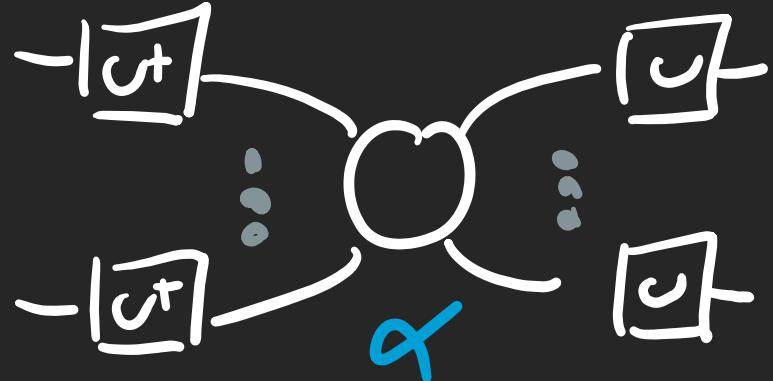
$$\begin{array}{c} \alpha \\ -\oplus- \\ \alpha\pi \end{array} = \begin{array}{c} \text{unfuse} \\ = \\ \alpha\pi \oplus \begin{array}{c} \alpha \\ -\oplus- \\ \alpha\pi \end{array} \end{array}$$

$$\begin{array}{c} \text{bialgebra} \\ = \\ \alpha\pi \oplus \begin{array}{c} \alpha \\ -\oplus- \\ \alpha\pi \end{array} \end{array} \stackrel{\text{copy}}{=} \begin{array}{c} \alpha \\ -\circ- \\ \alpha\pi \end{array} \oplus \begin{array}{c} \alpha \\ -\oplus- \\ \alpha\pi \end{array} \stackrel{\text{fuse}}{=} \begin{array}{c} \alpha \\ -\circ- \\ \alpha\pi \end{array}$$

The CNOT Gate



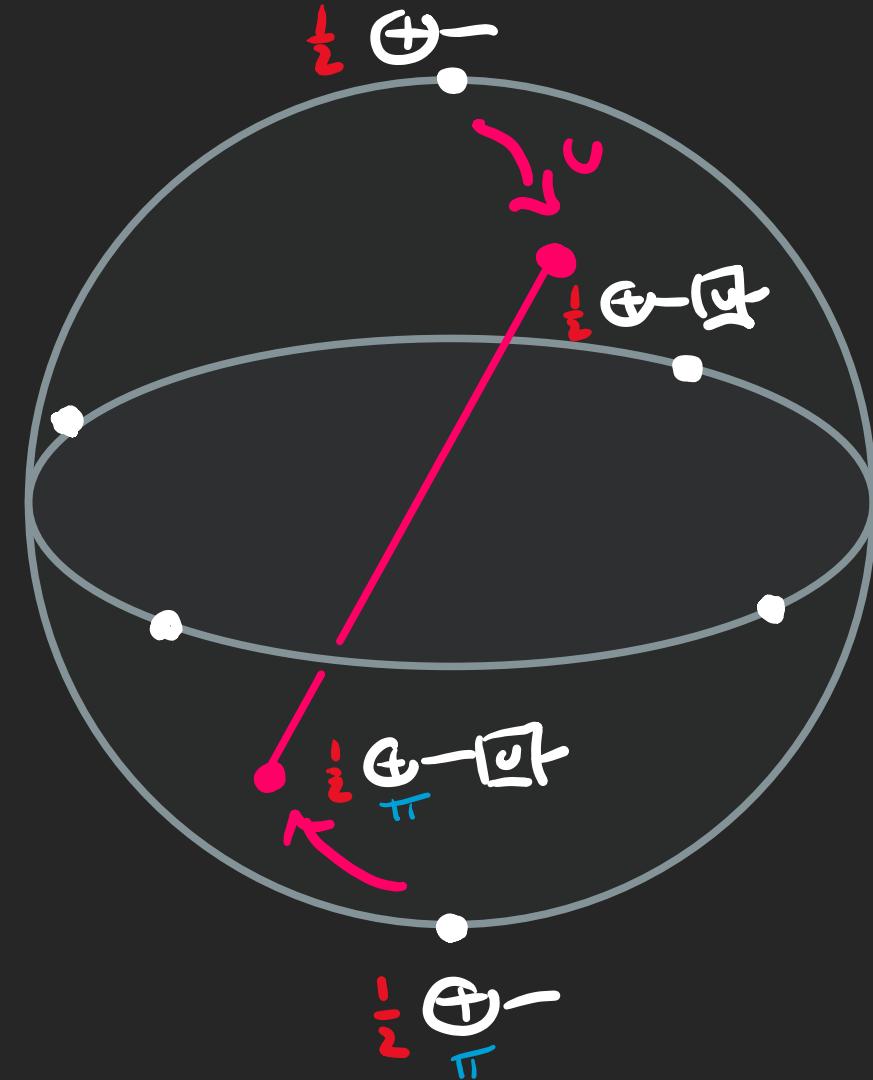
More Spiders



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

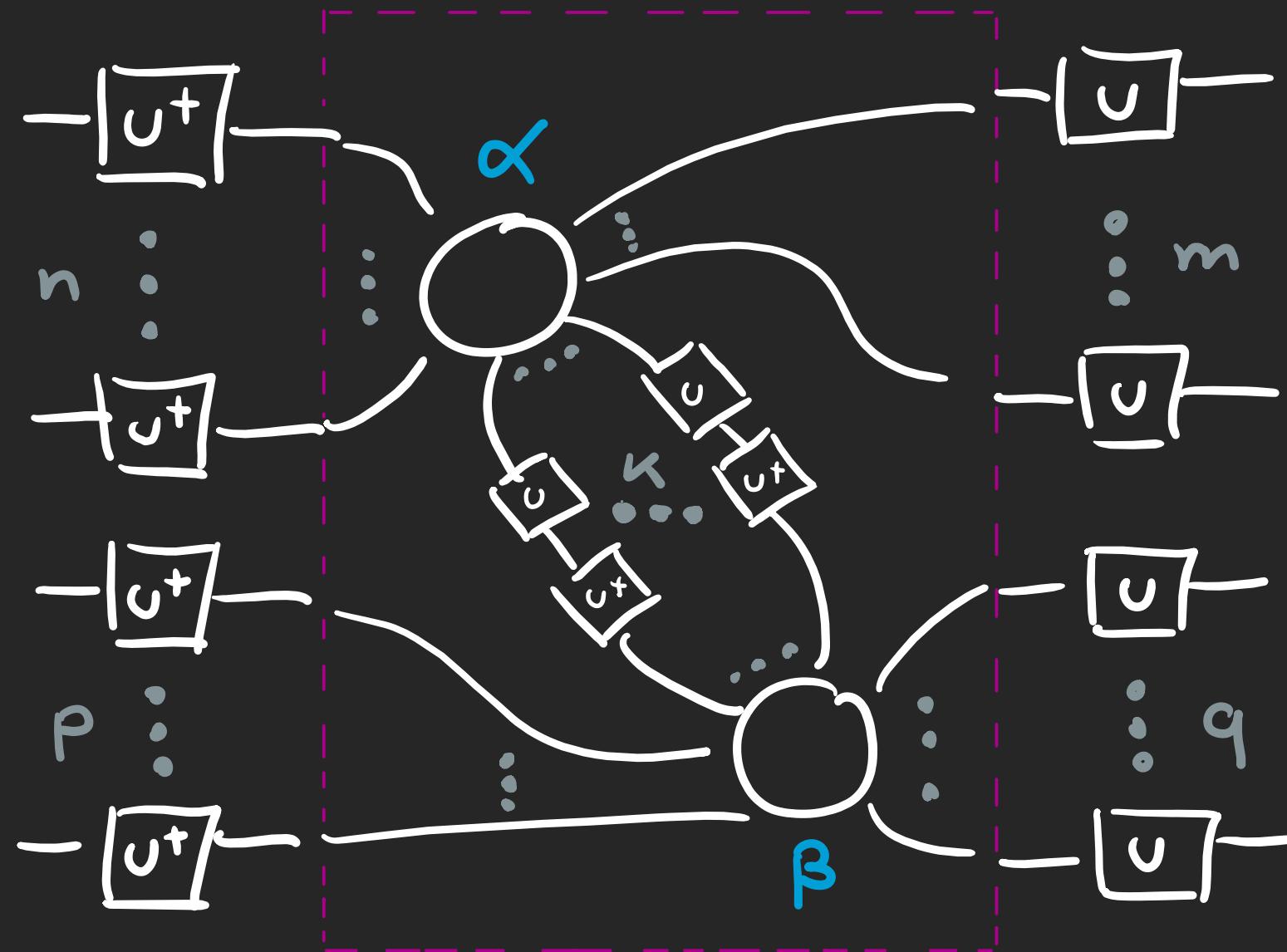
Basis

$$\begin{array}{c} \frac{1}{2} \oplus -\square \\ \frac{1}{2} \oplus -\square \\ \hline \end{array}$$
$$\pi$$



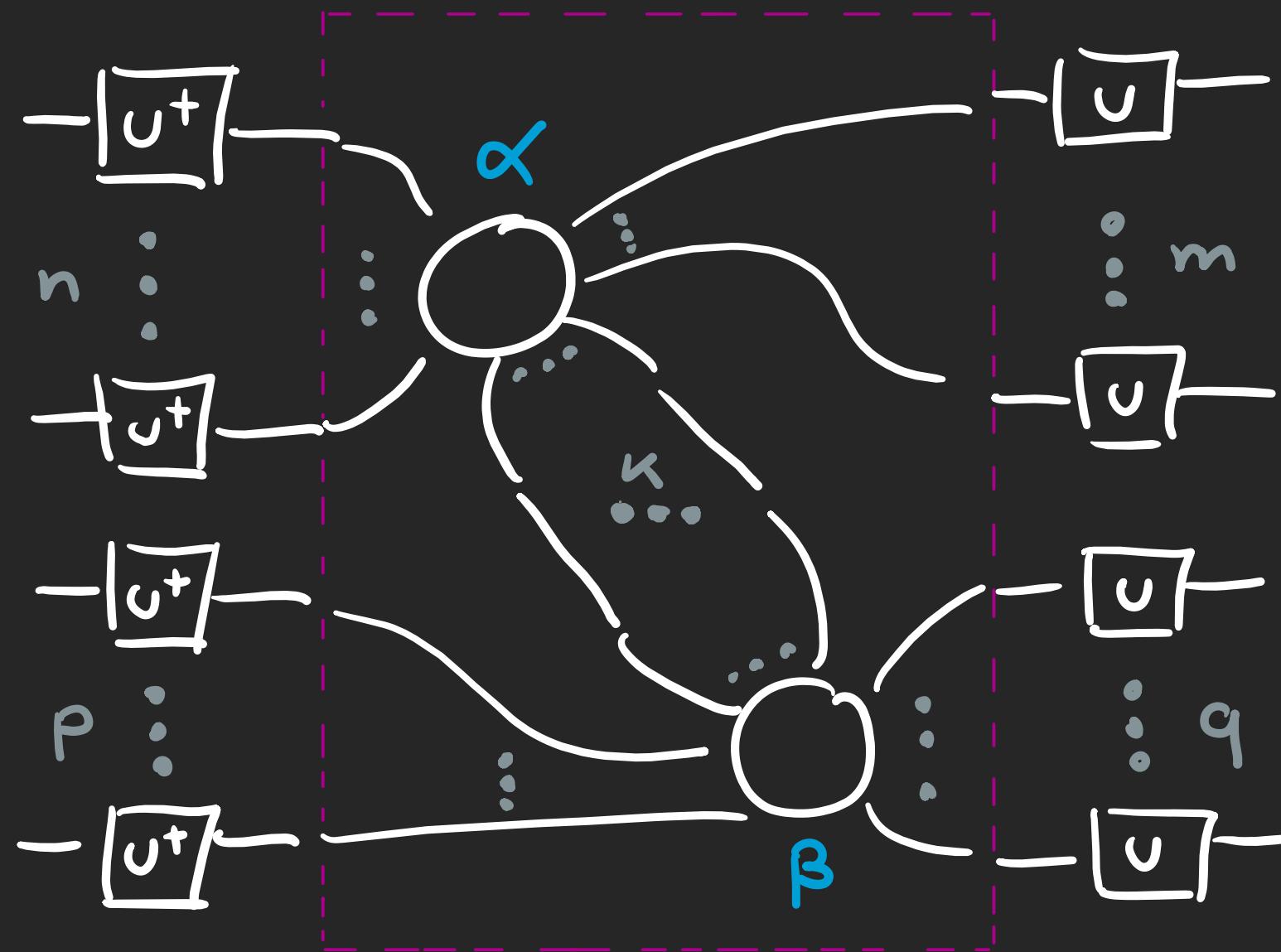
(1. -回収 = →)

More Spiders



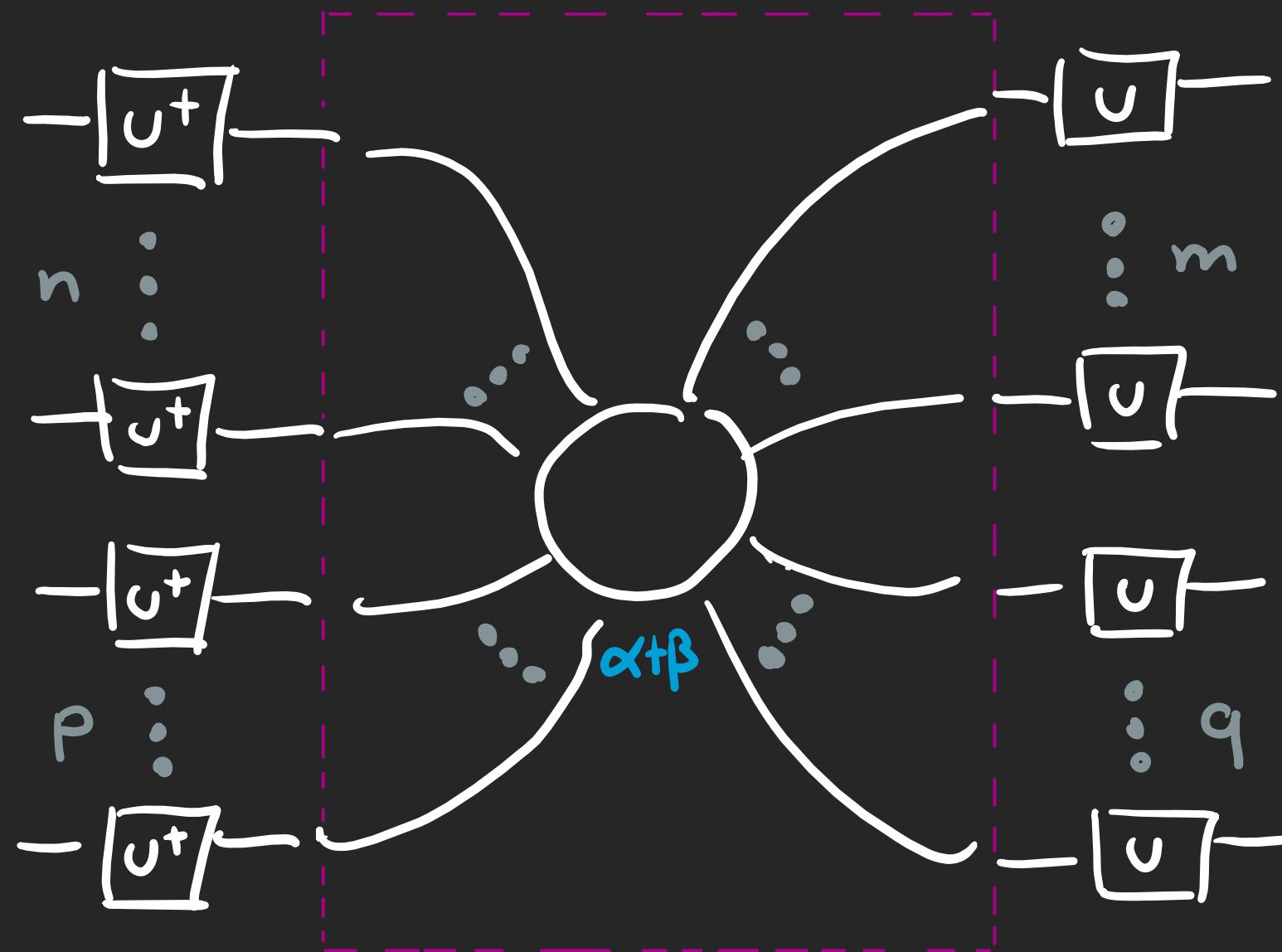
More Spiders

1. $\neg \Box \neg \Diamond = \perp$
(2. Spider fusion)



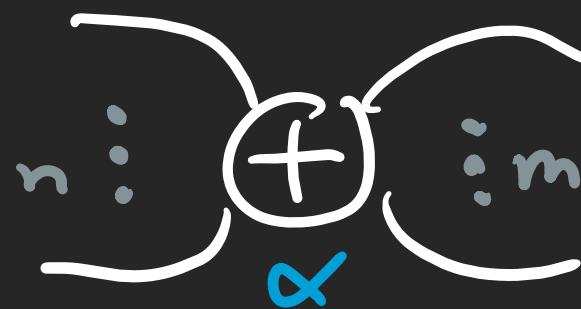
More Spiders

1. $\neg \Box \neg \Diamond = \perp$
2. Spider fusion

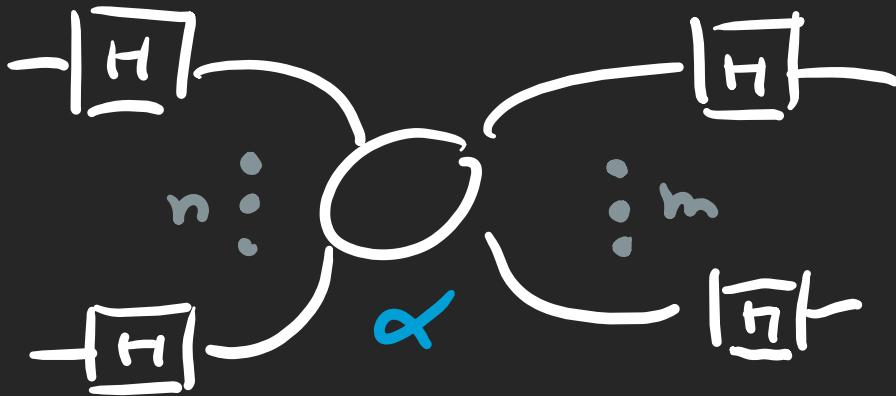


Spiders

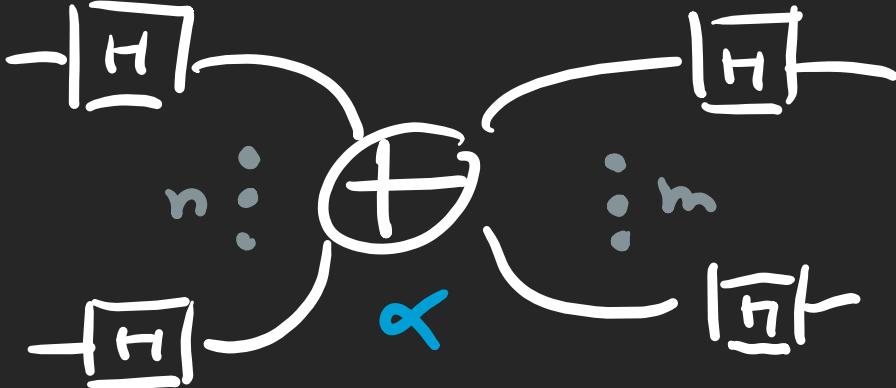
$\mathcal{Z}^{\text{basis}}$  \times^{basis} (because $-\overline{H}H = -$)



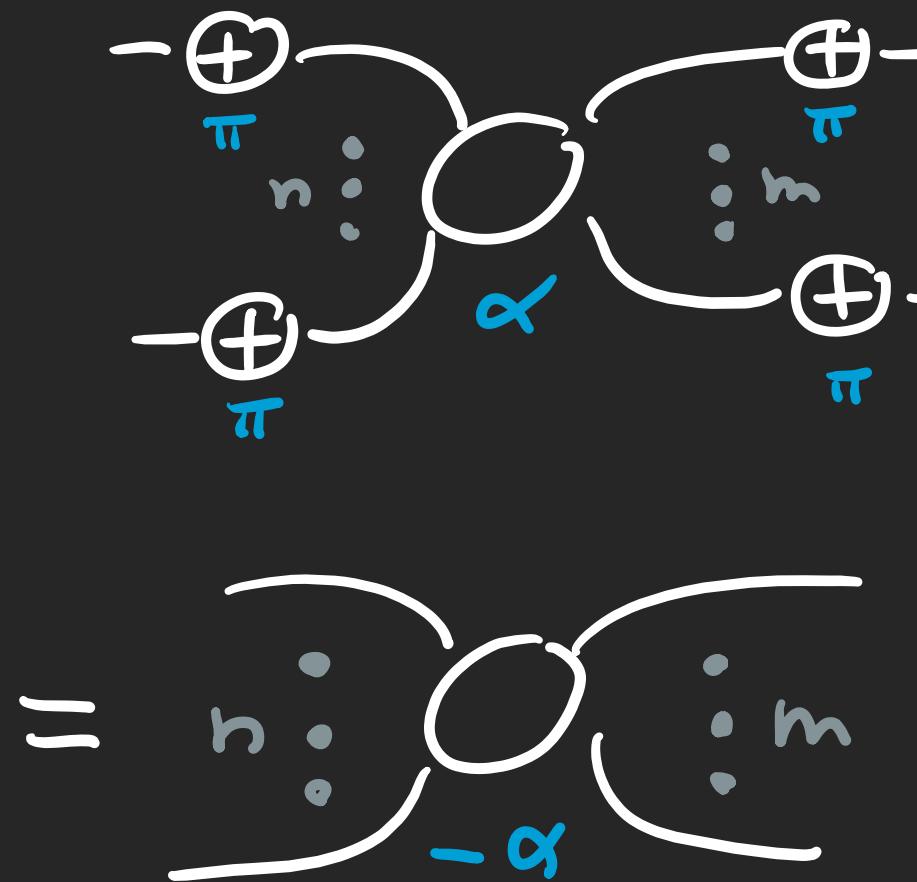
=



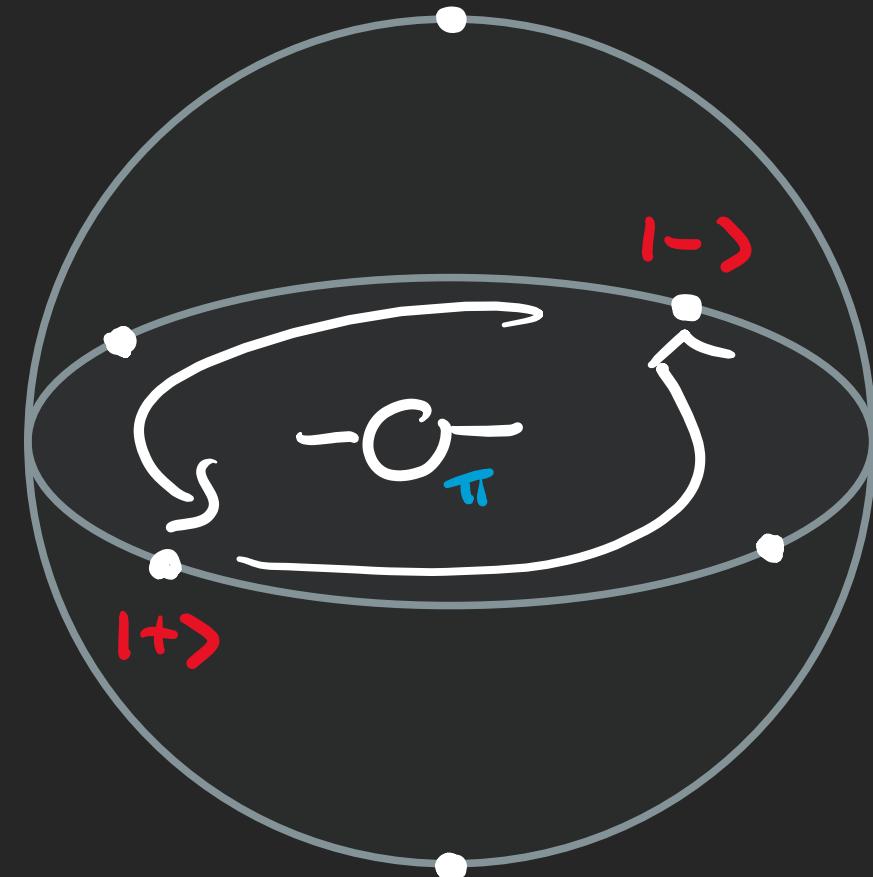
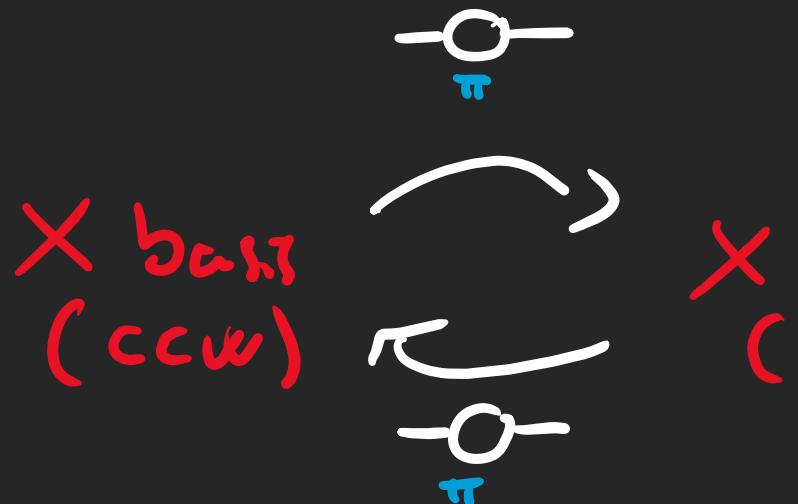
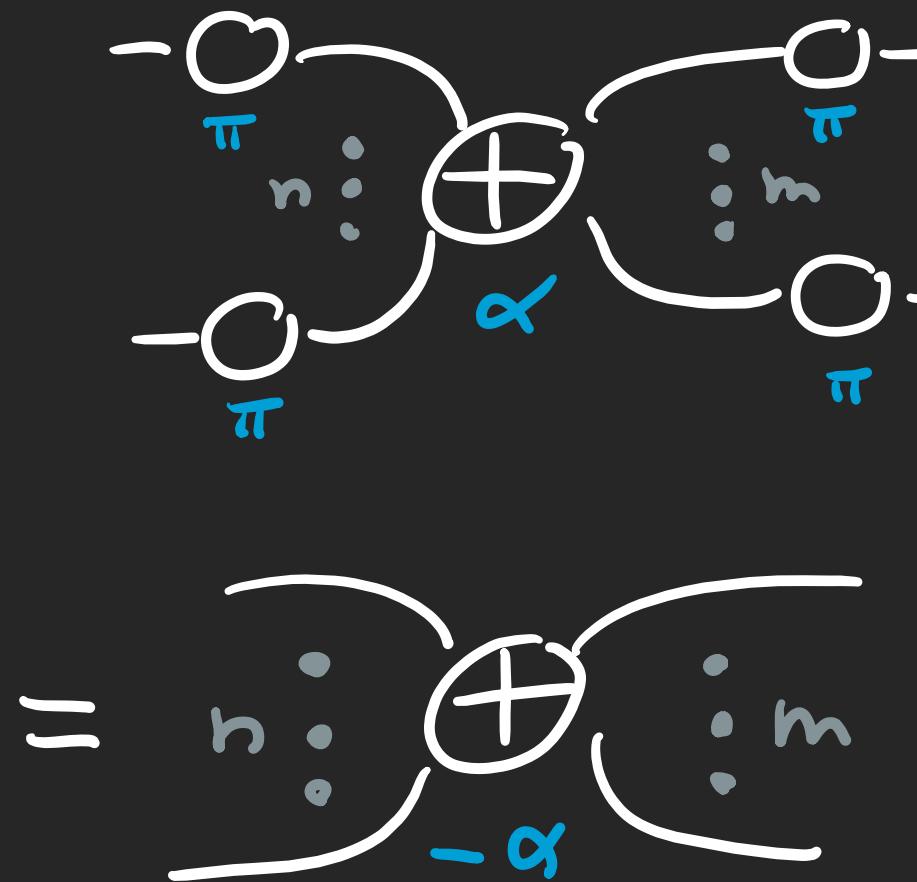
=



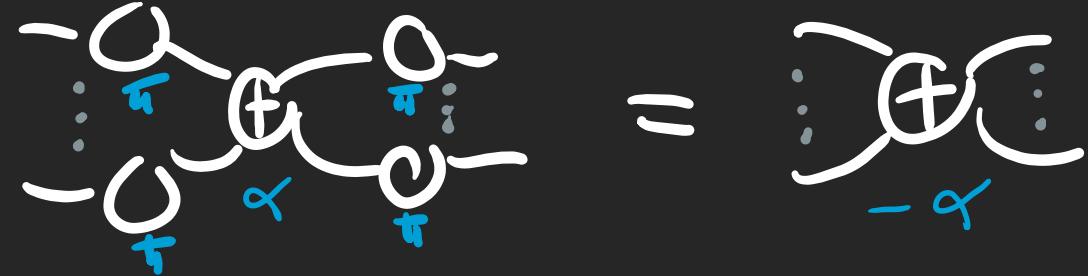
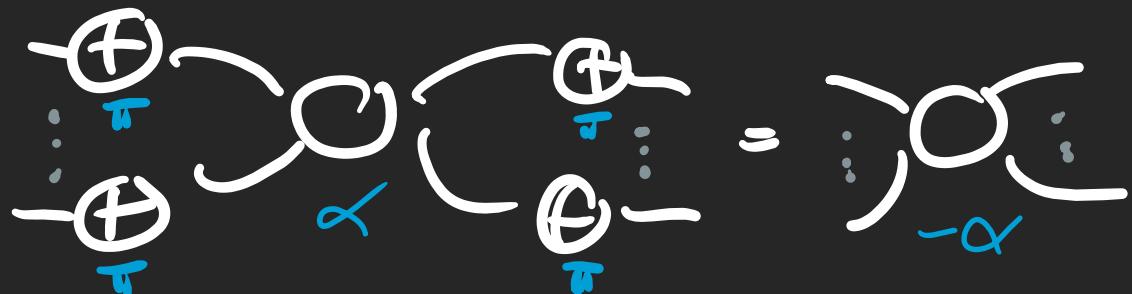
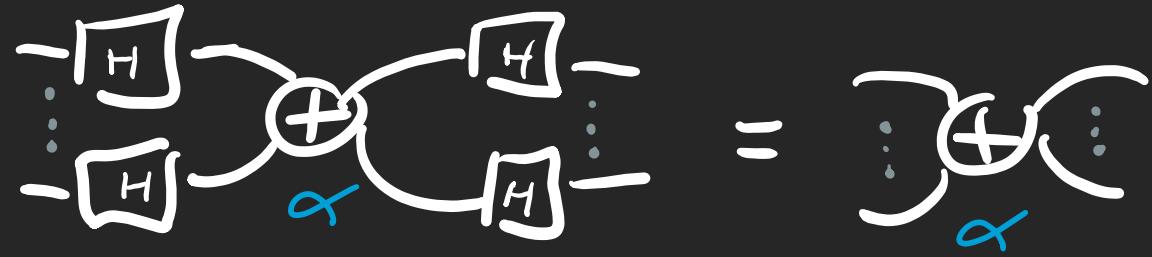
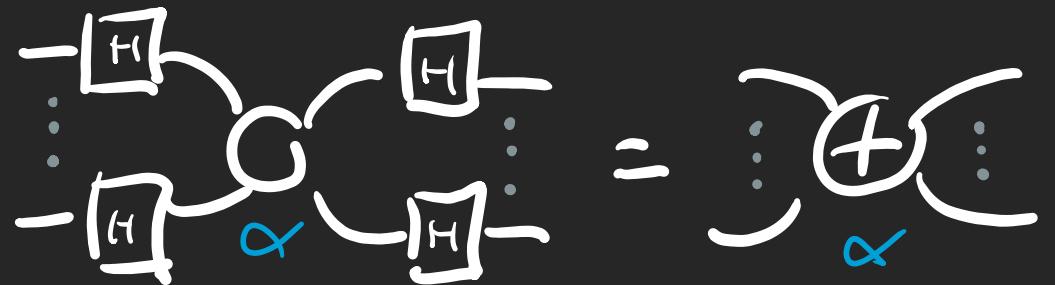
Spiders



Spiders



Conjugation rules



The H-commutation Rule

$$\begin{array}{c} -\square H \\ \vdots \\ -\square \end{array} \alpha \quad = \quad \begin{array}{c} \vdots \\ + \\ \alpha \\ -\square \end{array} \quad \begin{array}{c} H \\ \vdots \\ -\square \end{array}$$

Proof:

$$\begin{array}{c} -\square \\ \vdots \\ -\square \end{array} \alpha \quad = \quad \begin{array}{c} \vdots \\ -\square \square \\ -\square \end{array} \quad \begin{array}{c} \vdots \\ + \\ \alpha \\ -\square \end{array} \quad = \quad \begin{array}{c} \vdots \\ \alpha \\ -\square \end{array}$$

Conjugation

$$\begin{array}{c} \vdots \\ -\square \end{array} = -$$

The π -commutation Rule

Diagram illustrating the π -commutation rule. The left side shows two strands, one labeled π and one labeled α , crossing each other. The right side shows the strands swapped, with the π strand now below the α strand. An equals sign indicates they are equivalent.

Proof:

Diagram showing the proof of the π -commutation rule. It uses conjugation and the rule for crossing strands with π labels to show that the original configuration is equivalent to the swapped configuration.

Conjugation

Commutation rules

$$\begin{array}{c} \text{---} \square \text{---} \\ \vdots \quad \vdots \\ -\square \text{---} \alpha \text{---} \end{array} = \begin{array}{c} \text{---} \square \text{---} \\ \vdots \quad \vdots \\ \beta \oplus \square \text{---} \end{array}$$

$$\begin{array}{c} \text{---} \square \text{---} \\ \vdots \quad \vdots \\ -\square \text{---} \alpha \text{---} \end{array} = \begin{array}{c} \text{---} \square \text{---} \\ \vdots \quad \vdots \\ \beta \oplus \square \text{---} \end{array}$$

$$\begin{array}{c} \oplus \text{---} \\ \vdots \quad \vdots \\ -\oplus \text{---} \alpha \text{---} \end{array} = \begin{array}{c} \text{---} \oplus \text{---} \\ \vdots \quad \vdots \\ -\alpha \text{---} \oplus \text{---} \end{array}$$

$$\begin{array}{c} \text{---} \alpha \text{---} \\ \vdots \quad \vdots \\ -\pi \text{---} \alpha \text{---} \end{array} = \begin{array}{c} \text{---} \alpha \text{---} \\ \vdots \quad \vdots \\ -\alpha \text{---} \pi \text{---} \end{array}$$

H-commutation on CX gate

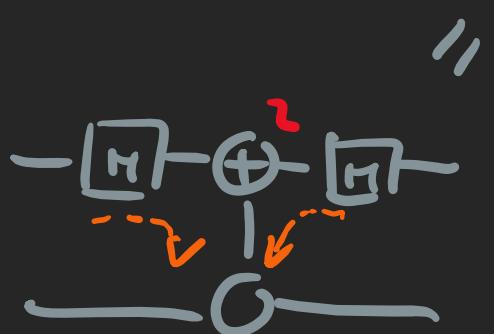
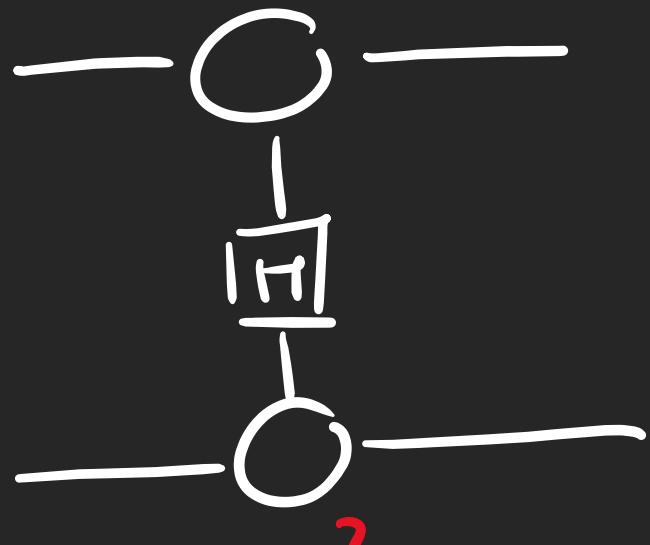


Proof:



The CZ Gate

Acts identically on both qubits:

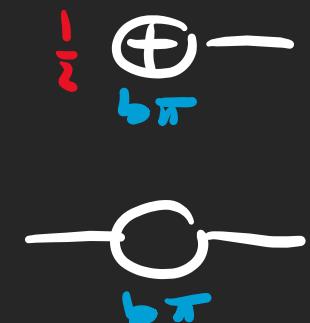
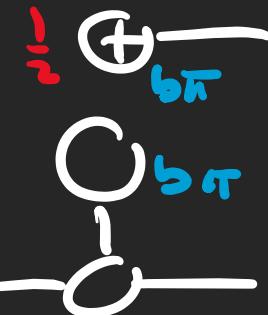
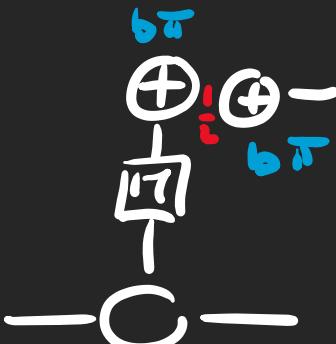
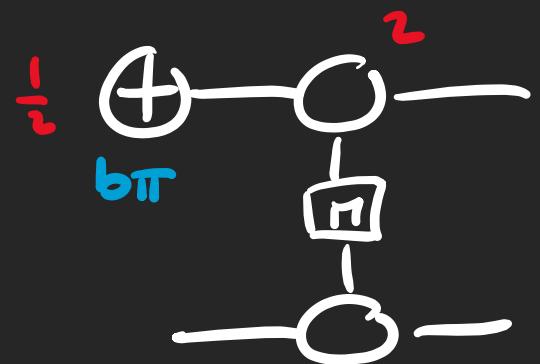


```
circ = QuantumCircuit(2)
circ.cz(0,1)
circ.draw("mpl")
```

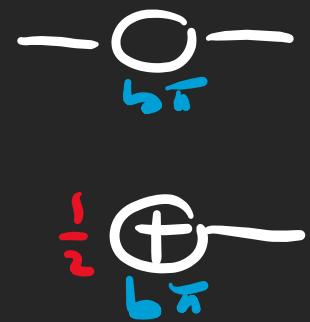
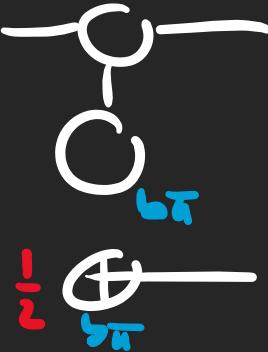
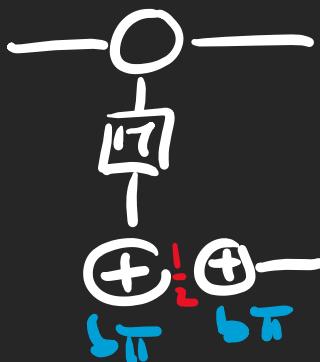
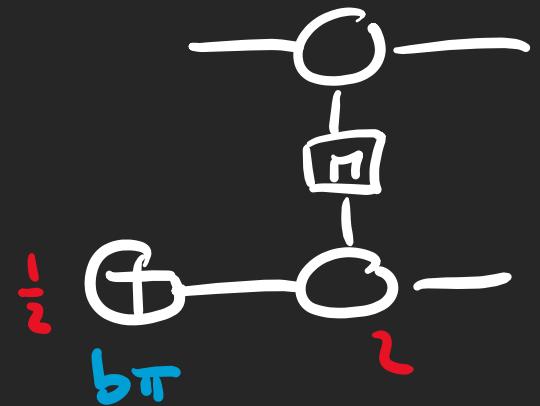
q_0 —————
 q_1 —————

The CZ Gate

Z-controlled Z gate
on both sides

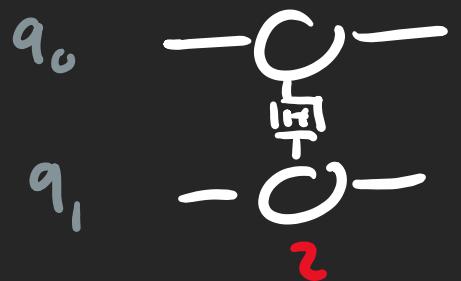


$|0\rangle$ if $b=0$
 $|1\rangle$ if $b=1$



id if $b=0$
 Z if $b=1$

The CNOT Gate



\longleftrightarrow

$$\begin{aligned} & q_1 \quad q_0 \quad q_1 \quad q_0 \\ & \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ & |00\rangle\langle 00| \\ & + |01\rangle\langle 01| \\ & + |10\rangle\langle 10| \\ & - |11\rangle\langle 11| \end{aligned}$$

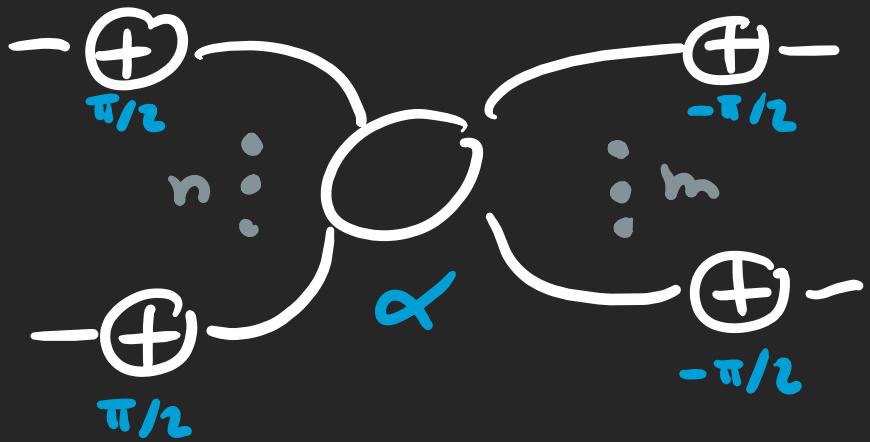
$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

Diagram illustrating the effect of the CNOT gate on four basis states:

- $|01\rangle \mapsto |01\rangle$
- $|00\rangle \mapsto |00\rangle$
- $|10\rangle \mapsto |10\rangle$
- $|11\rangle \mapsto -|11\rangle$

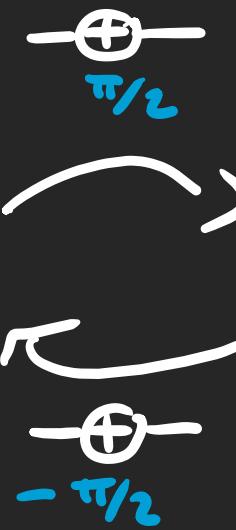
Arrows show the transformation from the initial state to the final state for each basis vector.

Spiders

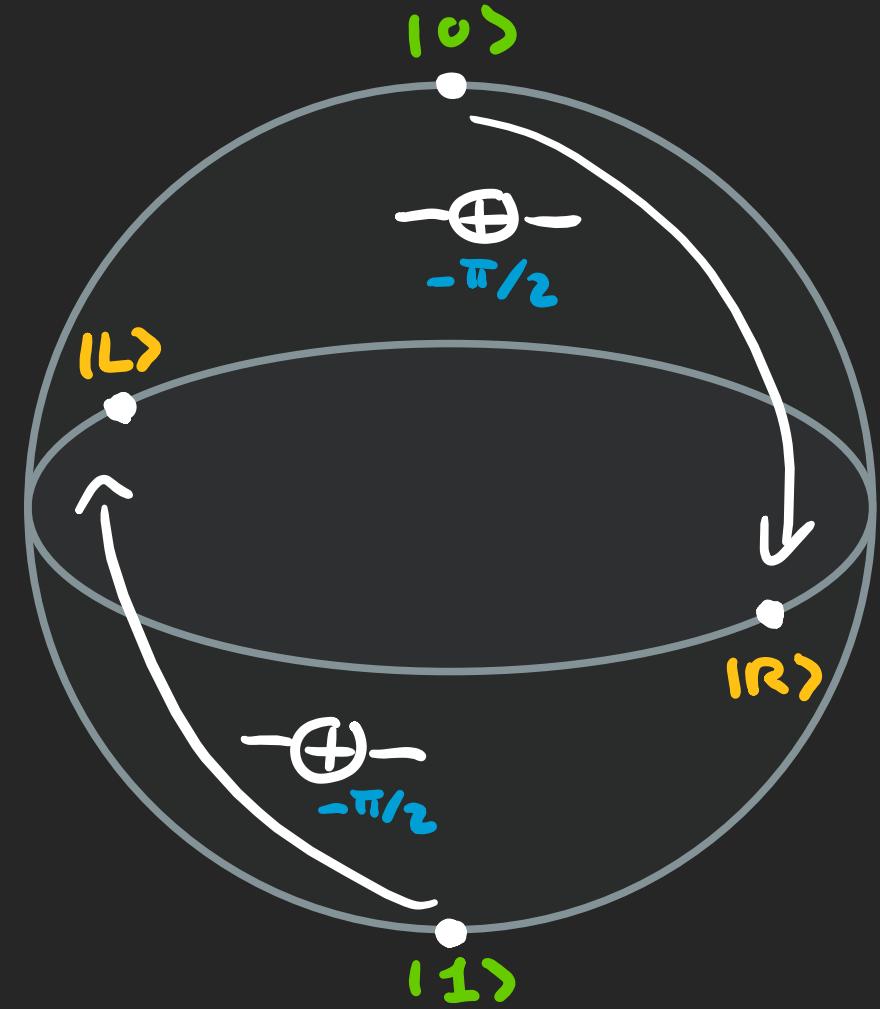


y basis spider

y basis

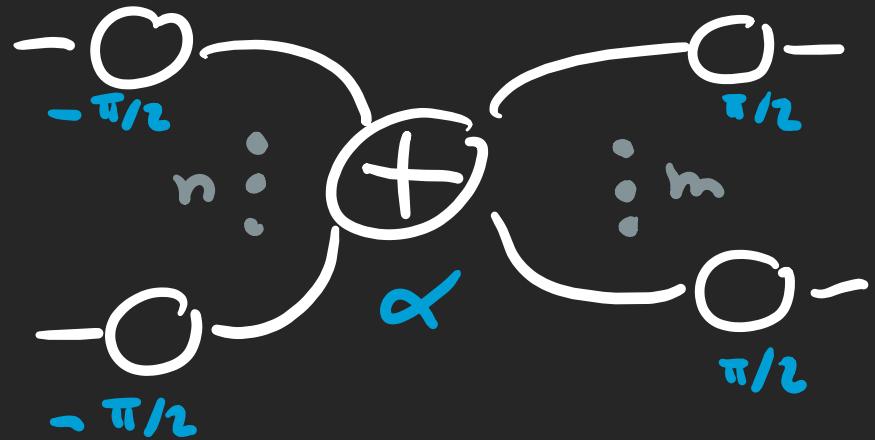


z basis

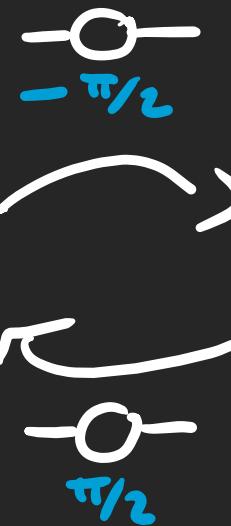


Spiders

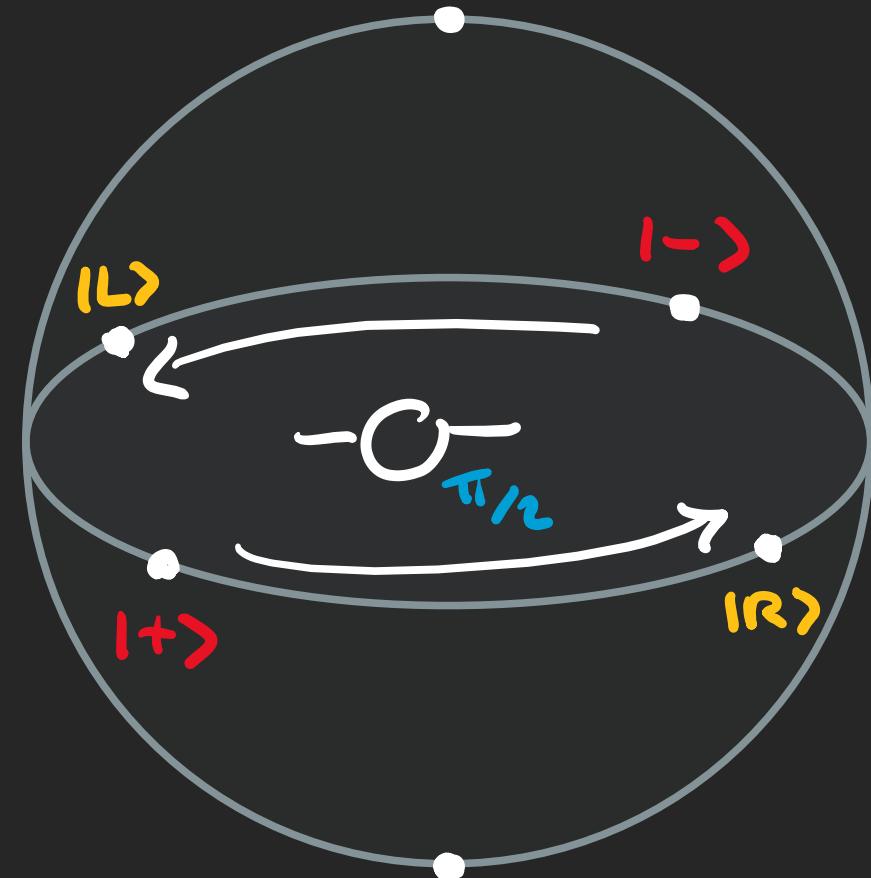
y basis



y basis spider



x basis

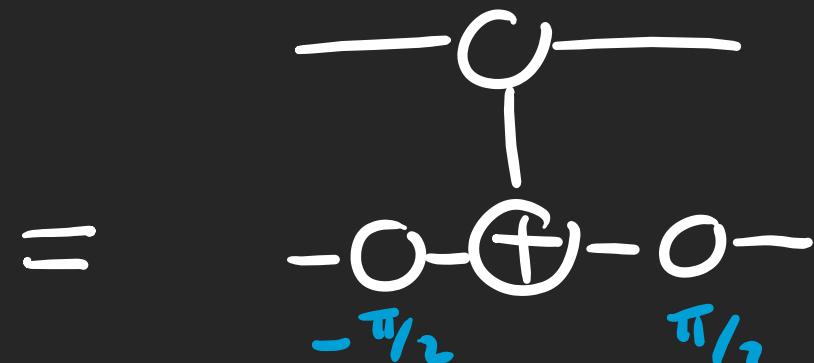
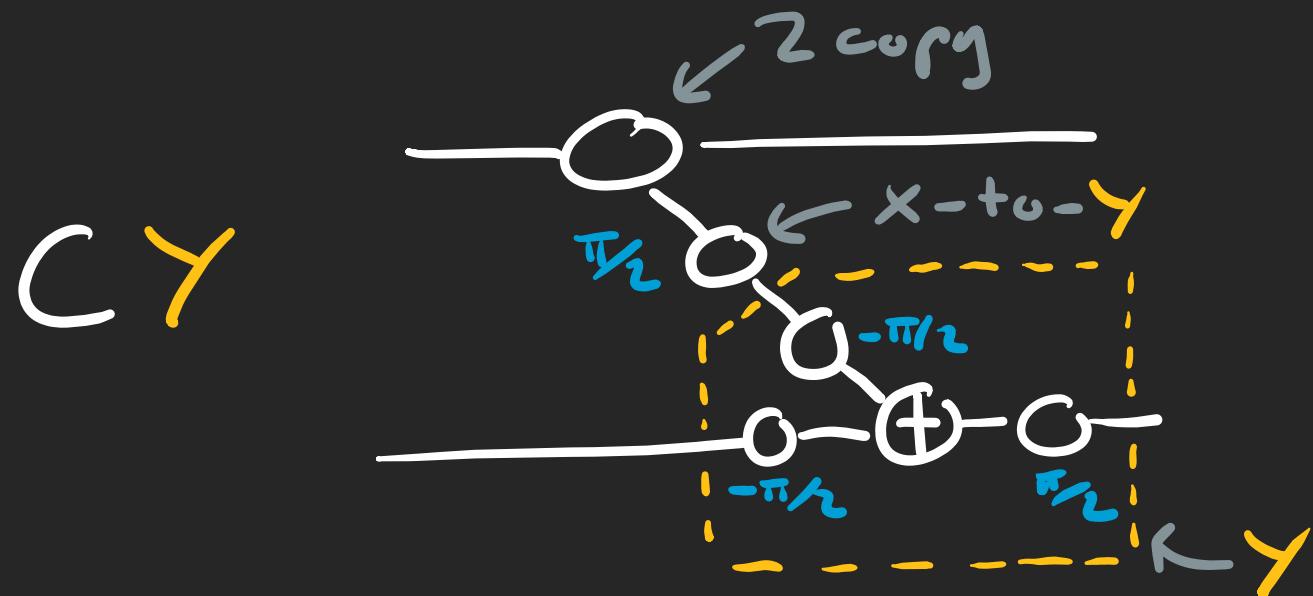
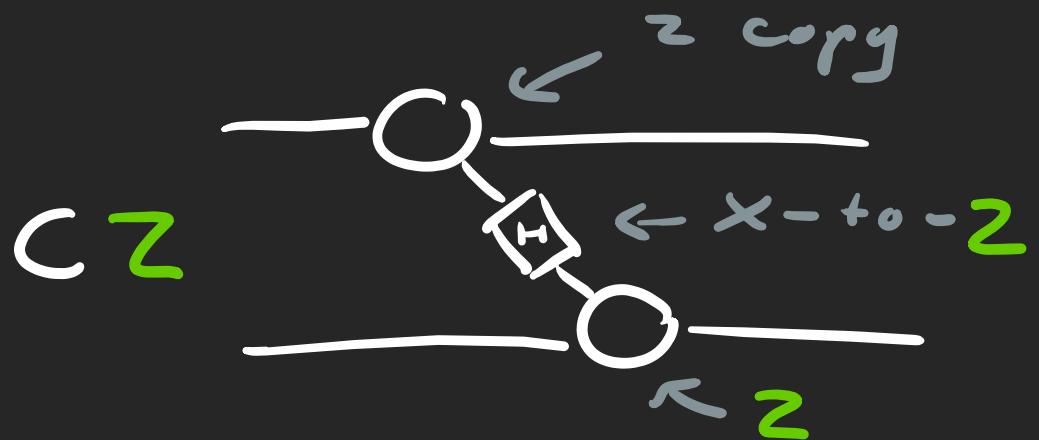
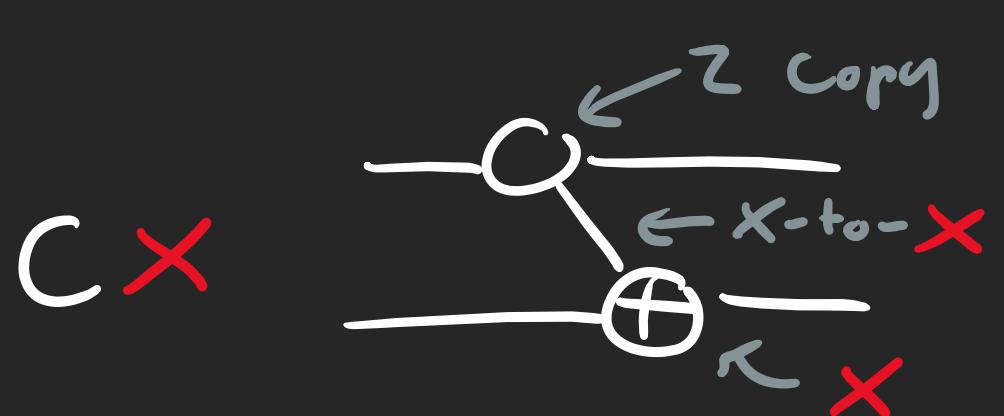


One More Conjugation Rule



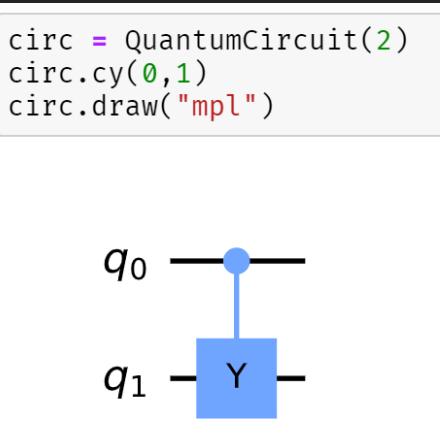
(Z copy copies X basis
 => X-to-? conversion in between)

The CY Gate



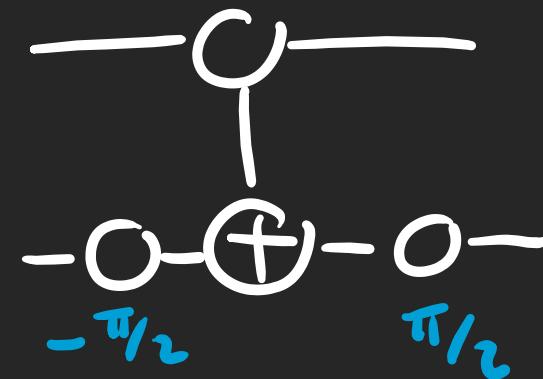
The CY Gate

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



A quantum circuit diagram with two horizontal lines representing qubits. The top line is labeled q_0 and the bottom line is labeled q_1 . A blue rectangular box labeled 'Y' is placed on the q_1 line, with a vertical line connecting it to a blue dot on the q_0 line.

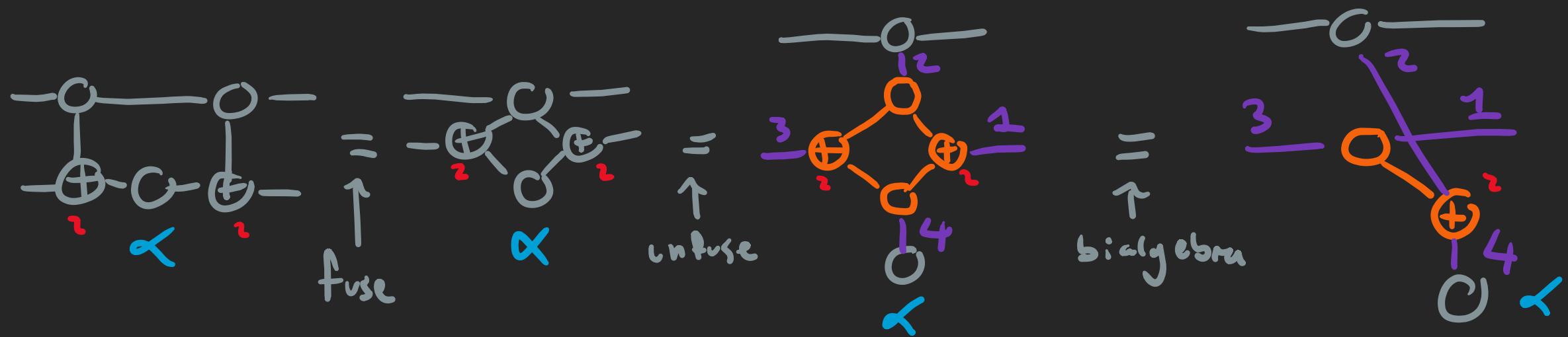
=



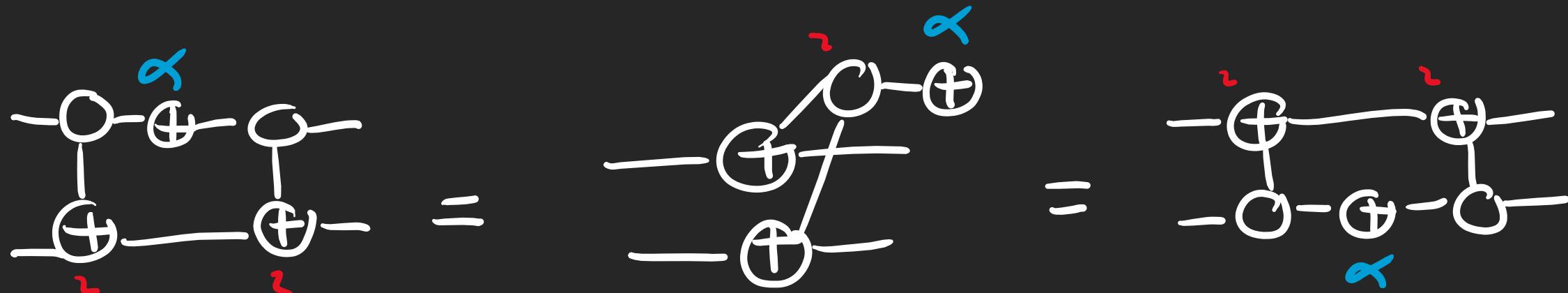
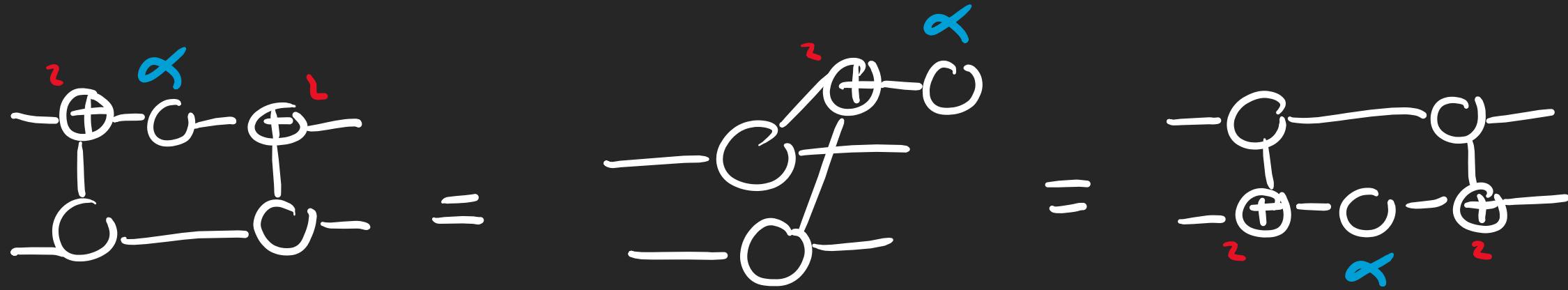
Phase Gadgets



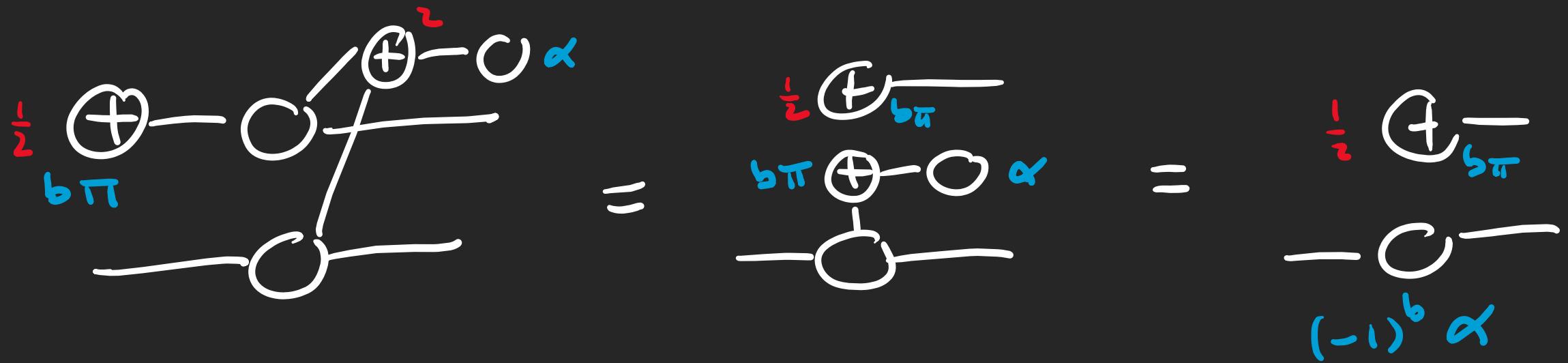
Proof :



Phase Gadgets



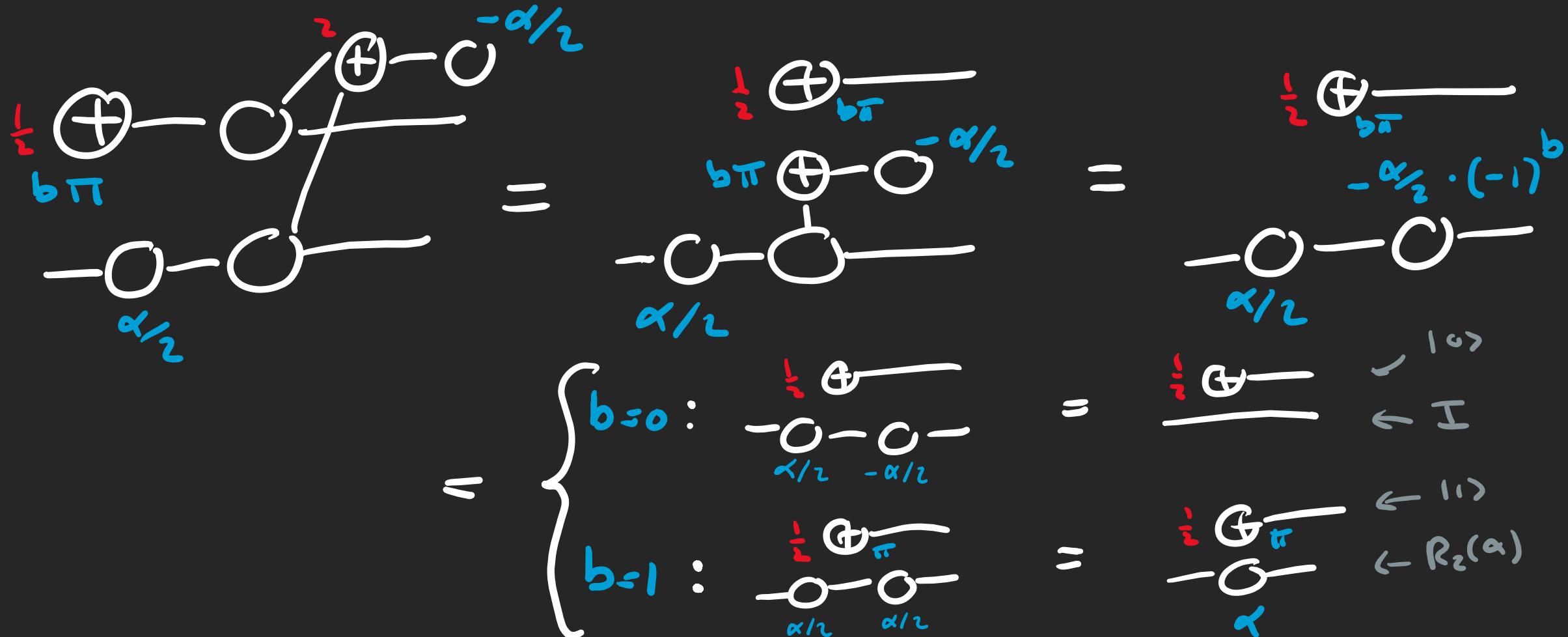
The CRz gate



This gives us an idea ...

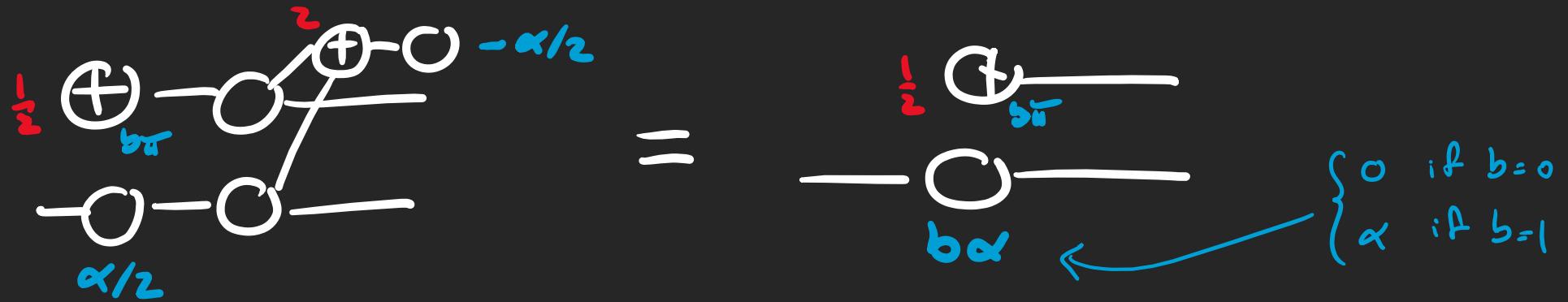
The CRz gate

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

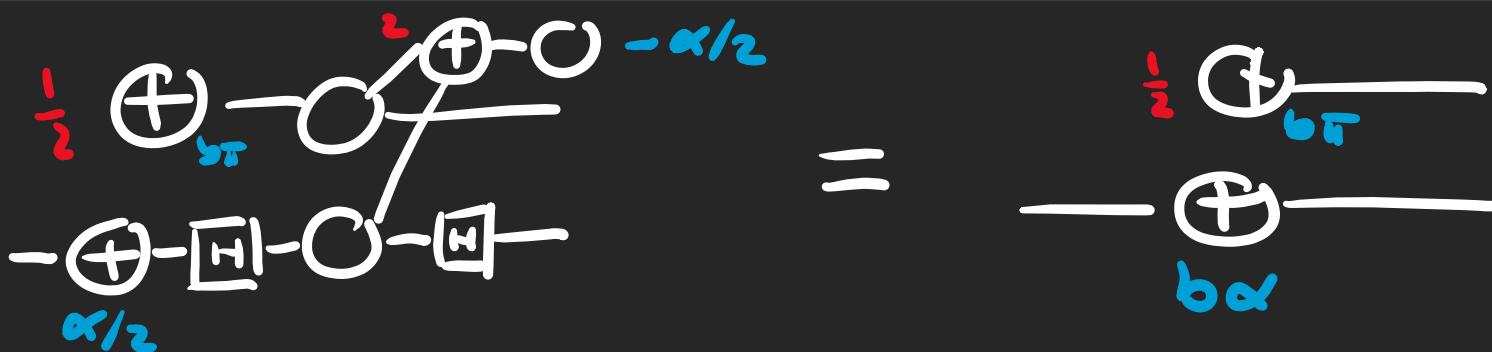


The CRz, CRx and Cry gates

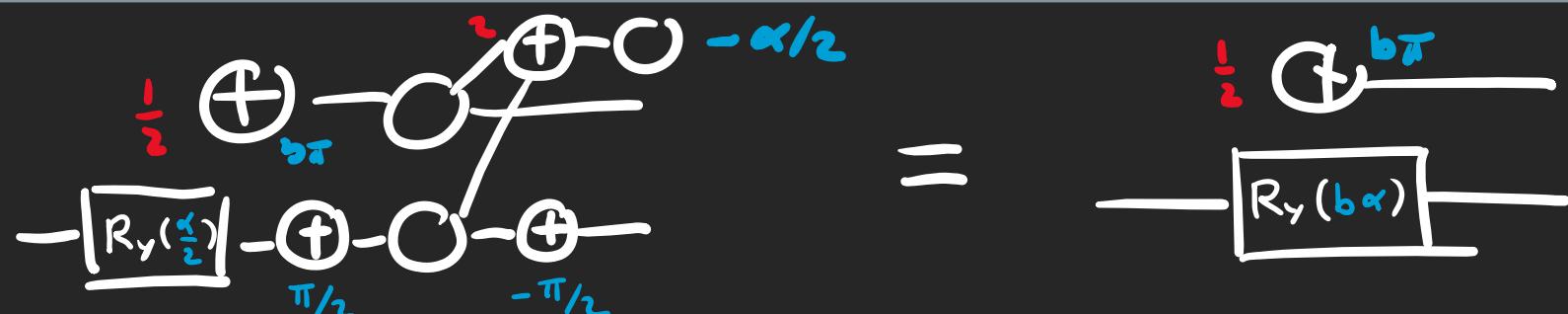
$CR_z(\alpha)$



$CR_x(\alpha)$

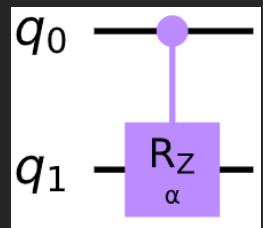


$CR_y(\alpha)$

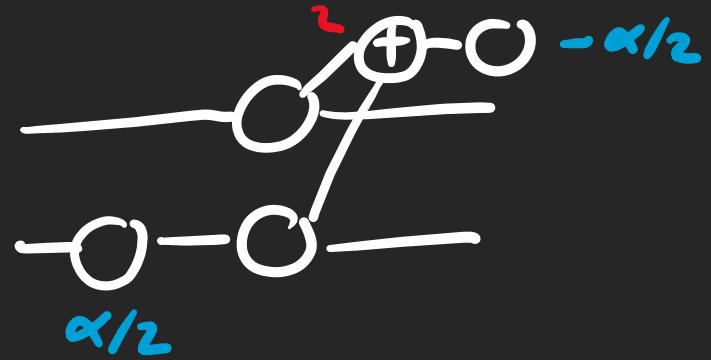


The CRz, CRx and CRY gates

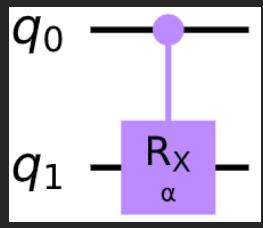
$CR_z(\alpha)$



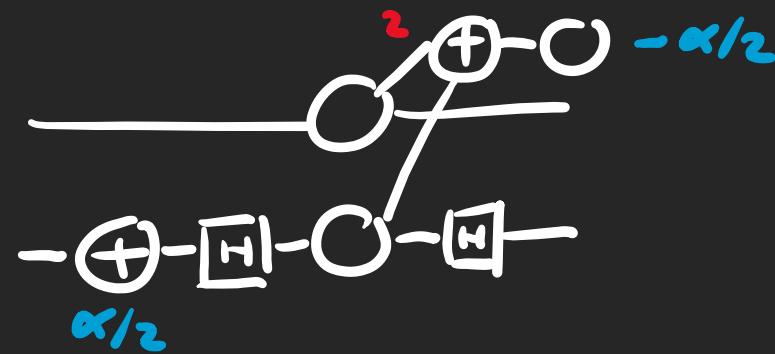
$\hat{=}$



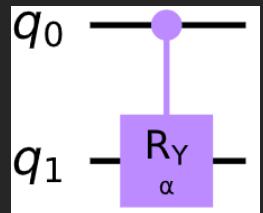
$CR_x(\alpha)$



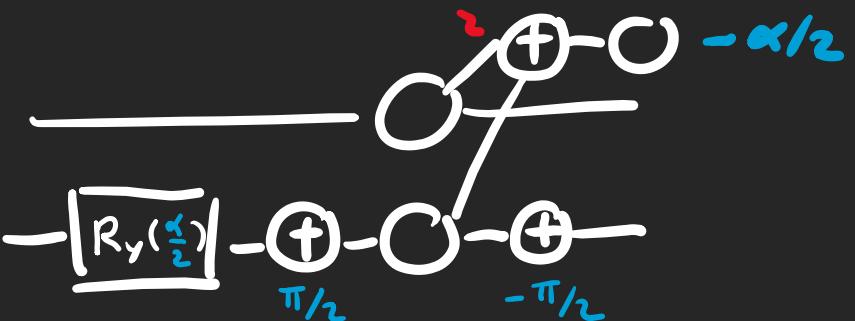
$\hat{=}$



$CR_y(\alpha)$

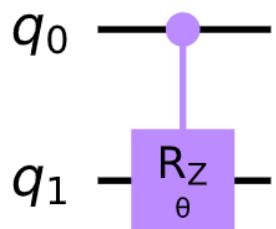


$\hat{=}$

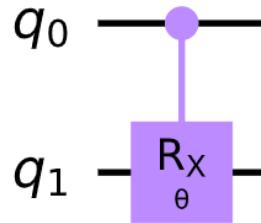


The CRz, CRx and CRY gates

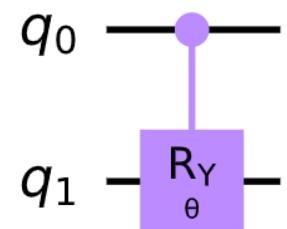
```
theta = Parameter("θ")
circ = QuantumCircuit(2)
circ.crz(theta, 0, 1)
circ.draw("mpl")
```



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



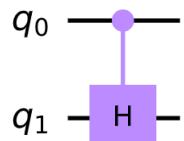
```
theta = Parameter("θ")
circ = QuantumCircuit(2)
circ.cry(theta, 0, 1)
circ.draw("mpl")
```



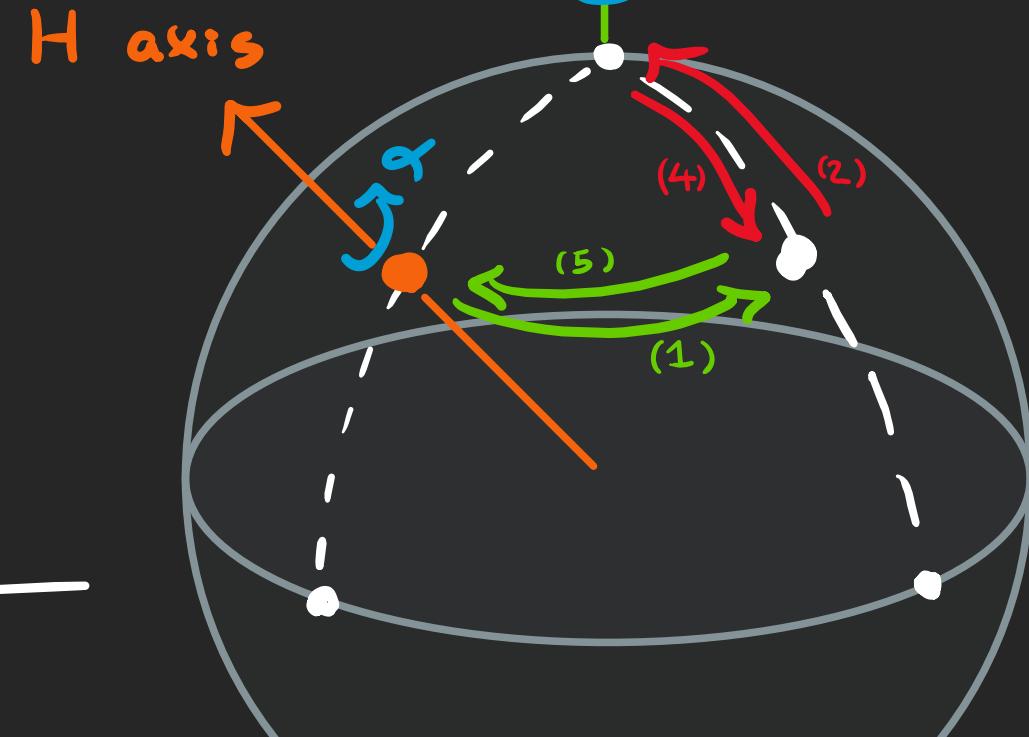
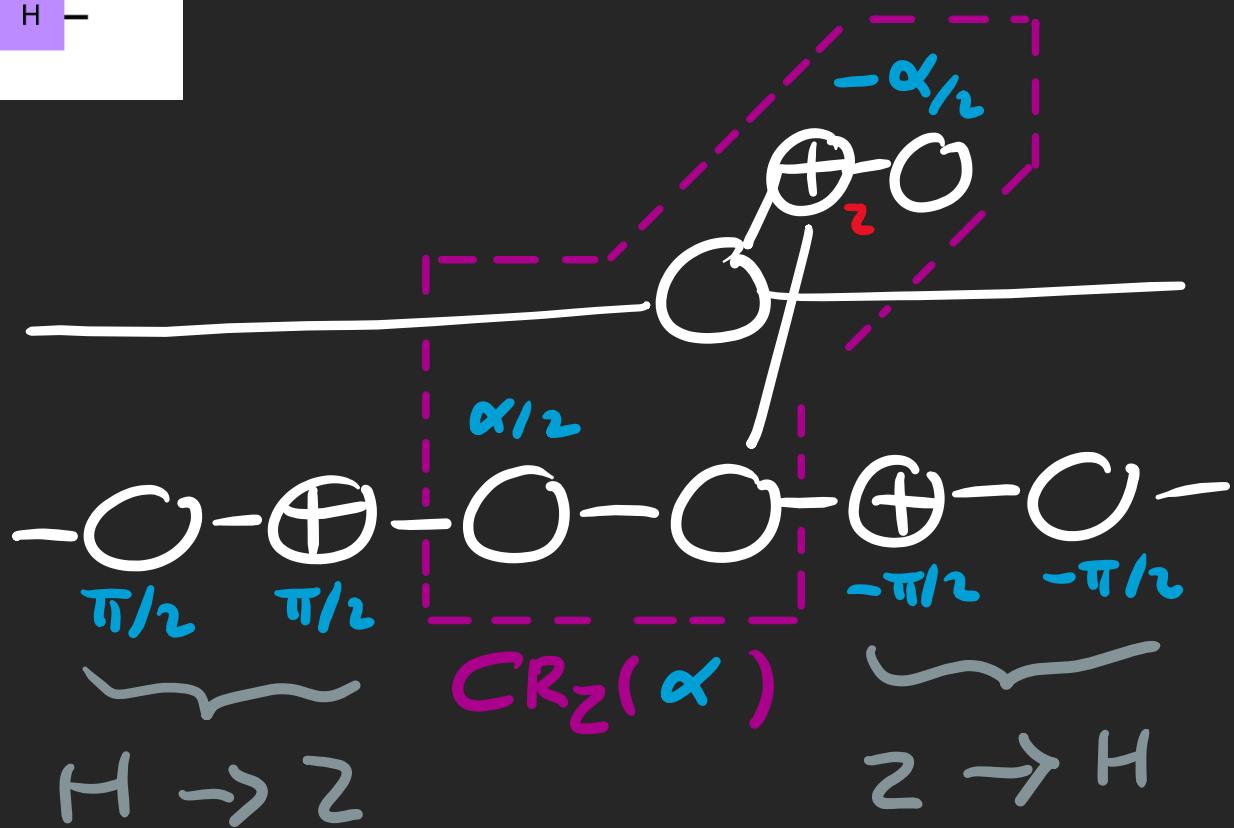
Bonus !!

The CH Gate

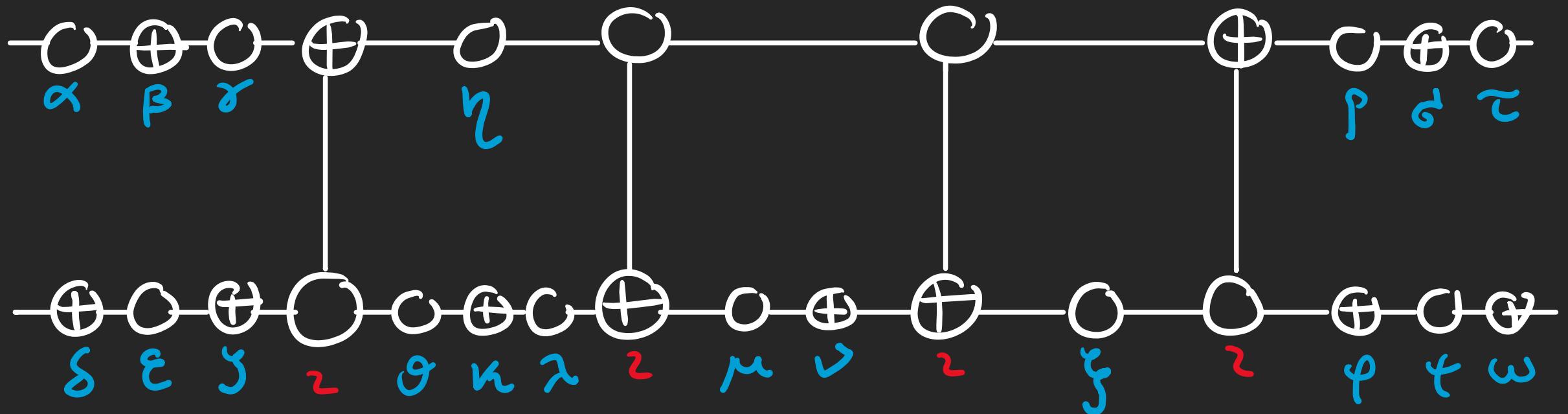
```
circ = QuantumCircuit(2)
circ.ch(0, 1)
circ.draw("mpl")
```



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



Generic 2-qubit Rotation

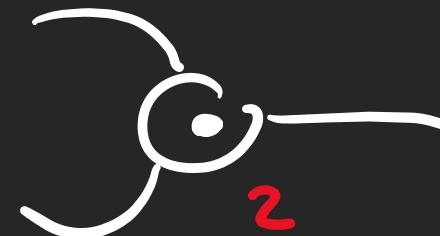


19 single-qubit rotations + 4 CNOT gates

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

The Product Monoid

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



$$\frac{1}{2} \begin{matrix} \oplus \\ a\pi \end{matrix} \circ \overset{\circ}{2} = \frac{1}{2} \begin{matrix} \oplus \\ ab\pi \end{matrix} -$$

A diagram illustrating the multiplication of two monoid elements. It shows two loops, one labeled 'a' and one labeled 'b', connected by a curved arrow. The result is a difference of two terms, each involving a red '1' above a blue '+' sign and a label below it.

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

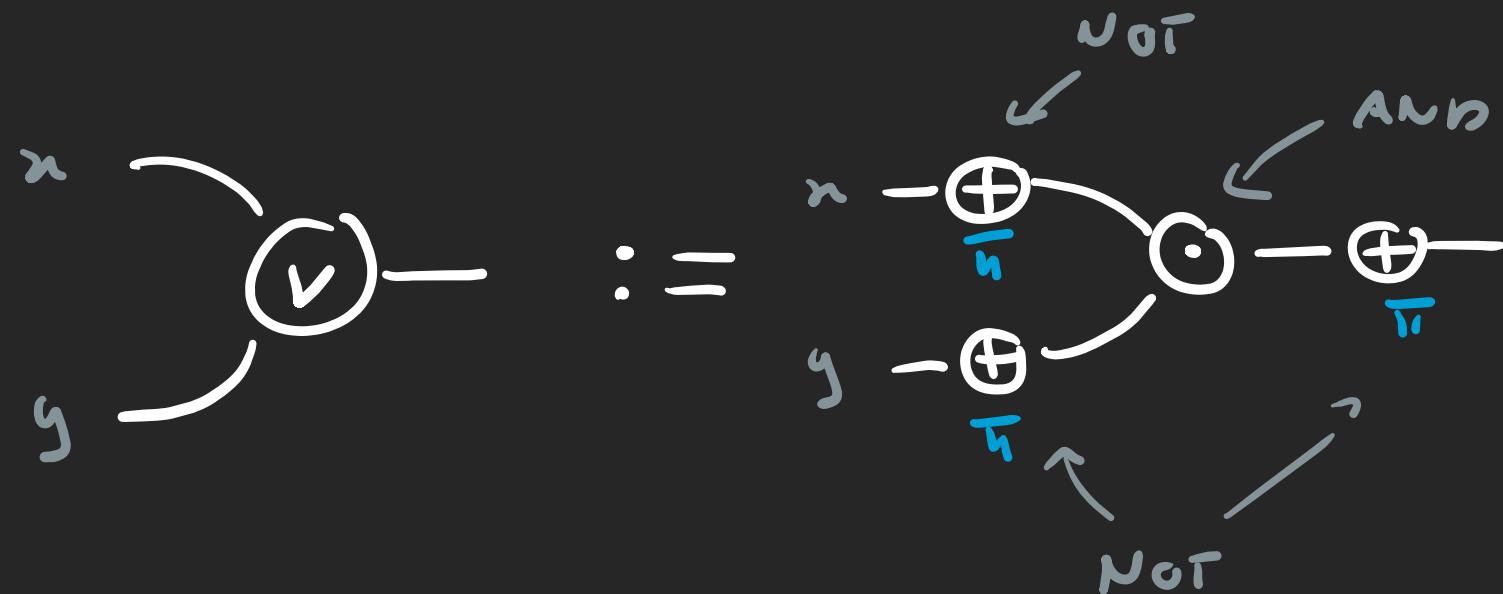
$\neg = \text{NOT}$

$\oplus = \text{xOR}$

$\wedge = \text{AND}$

$\vee = \text{OR}$

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

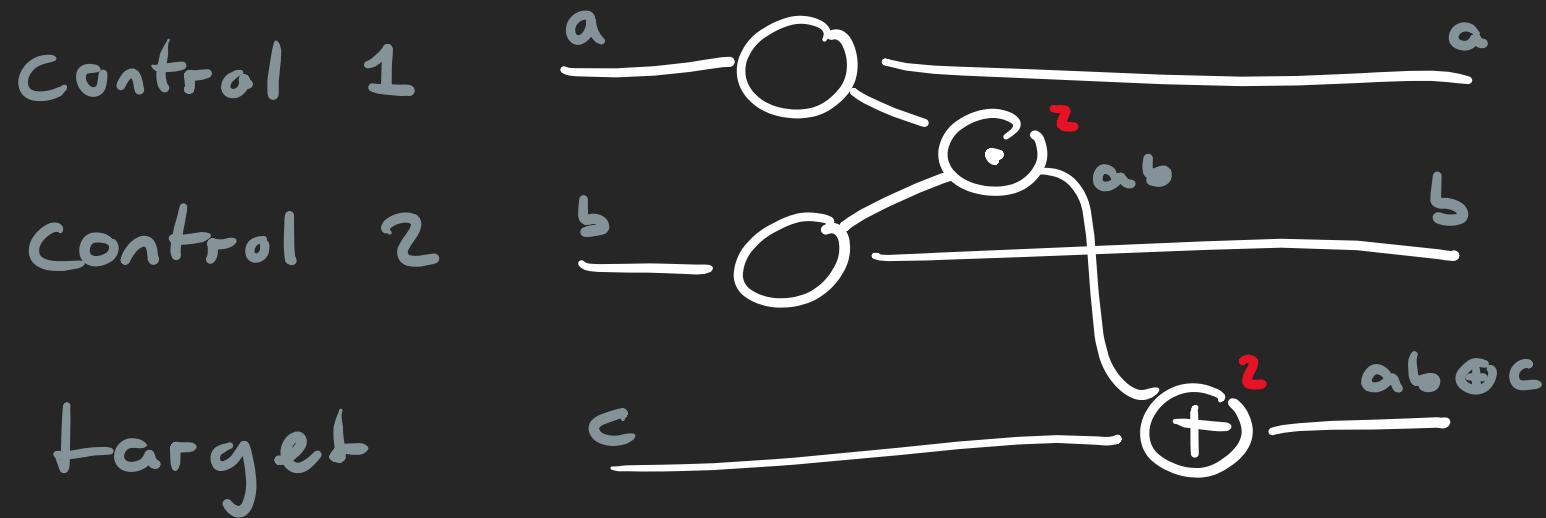


```

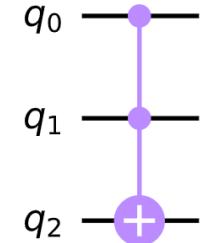
circ = QuantumCircuit(3)
circ.ccx(0,1,2) # The Toffoli gate
circ.draw("mpl")

```

The Toffoli (CCX) Gate

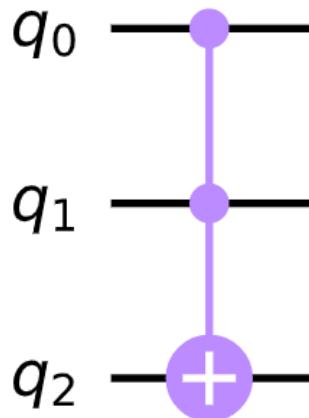


Doubly-controlled NOT/x gate on target

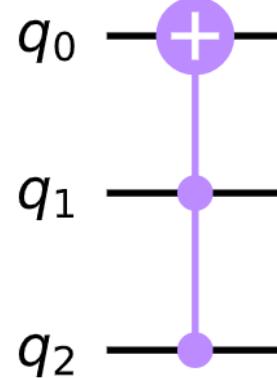


The Toffoli (CCX) Gate

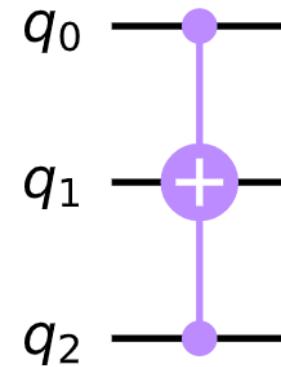
```
circ = QuantumCircuit(3)
circ.ccx(0,1,2)
circ.draw("mpl")
```



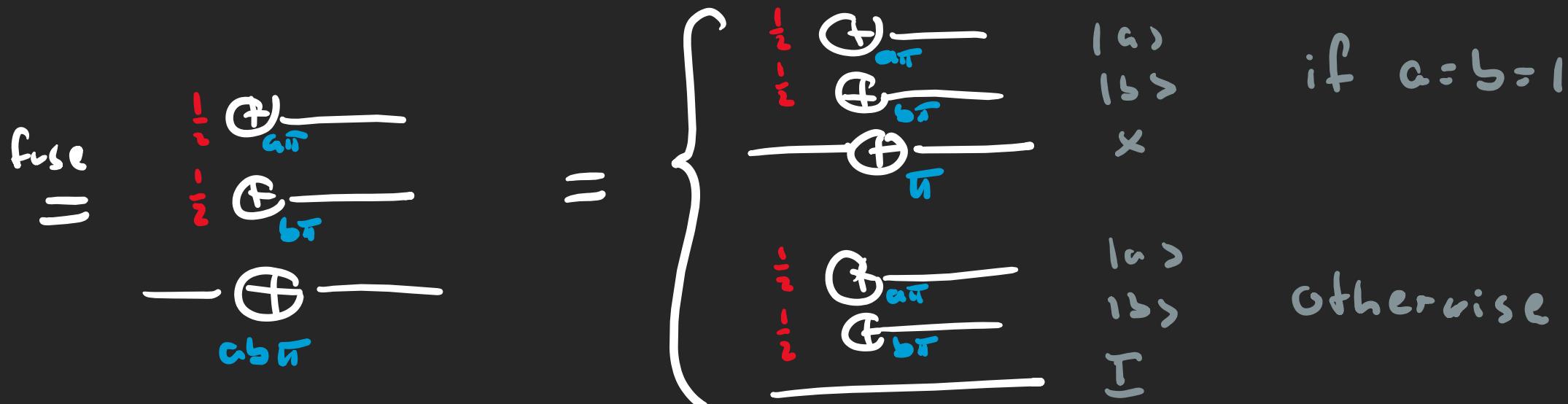
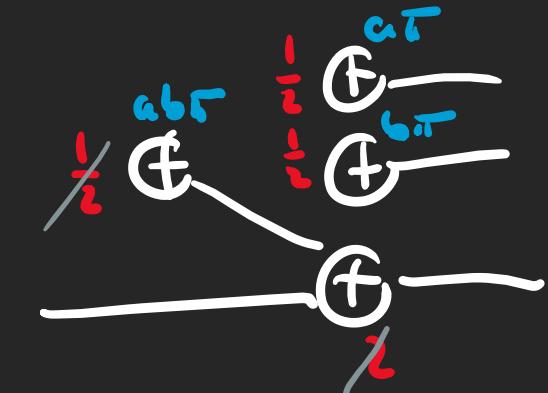
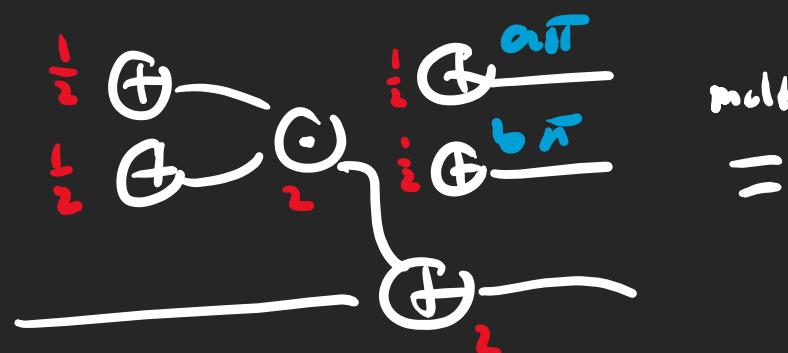
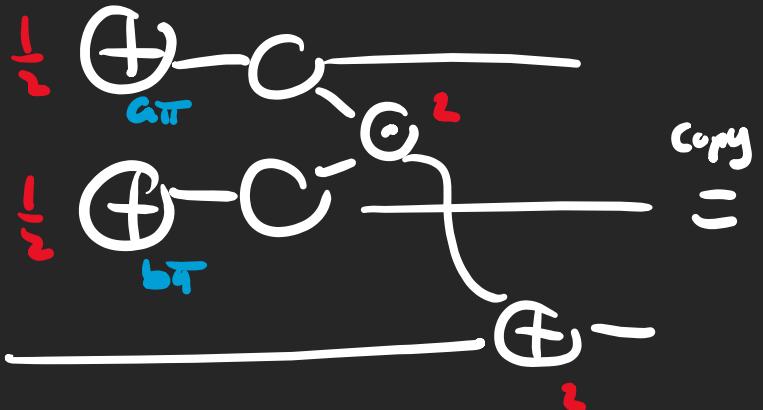
```
circ = QuantumCircuit(3)
circ.ccx(1, # control
         2, # control
         0) # target
circ.draw("mpl")
```



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

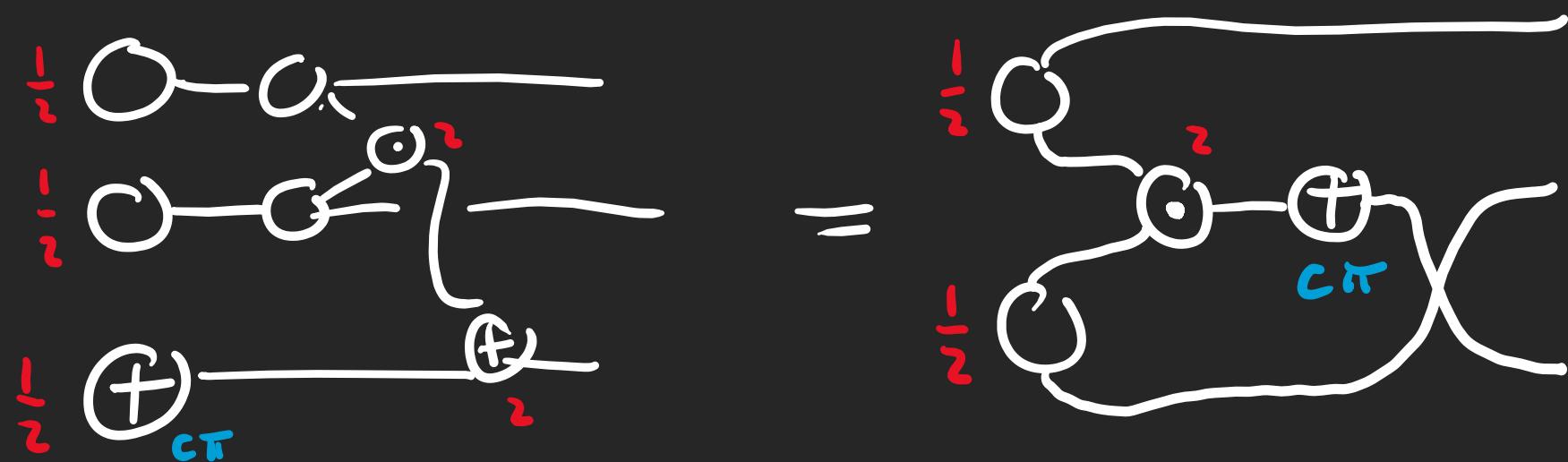


The Toffoli (CCX) Gate



The Toffoli (CCX) Gate

If the controls are not in a $\text{Z}^{\otimes 3}$ 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 :



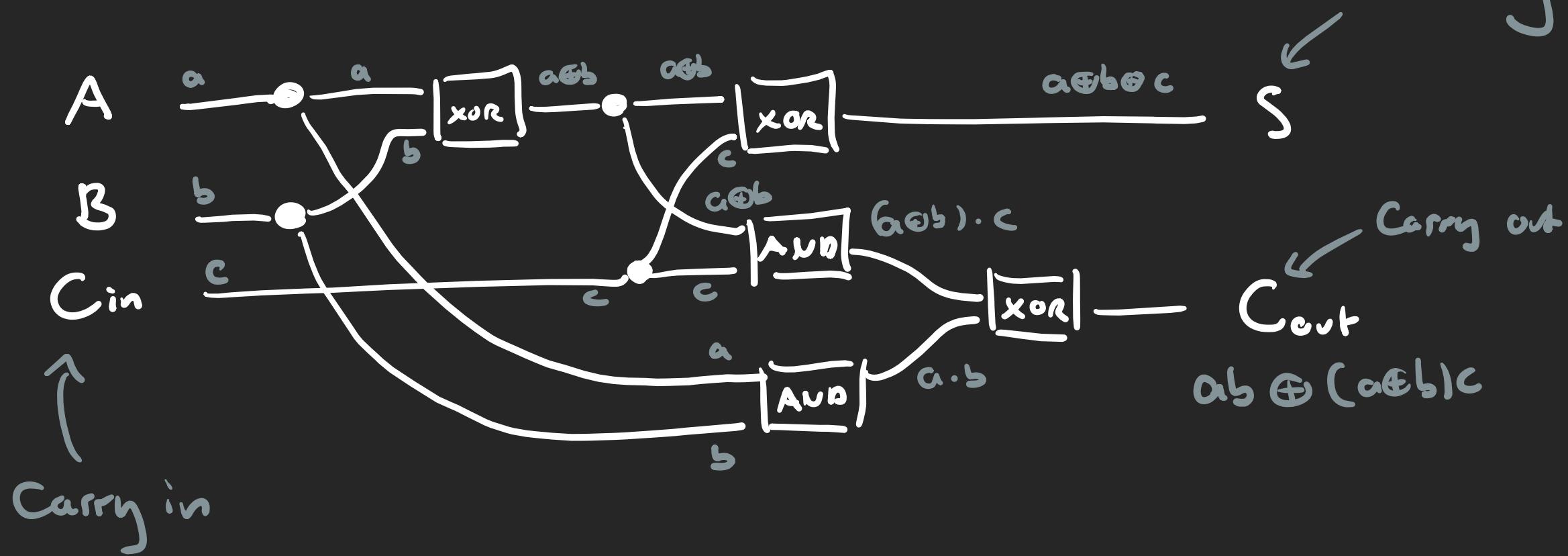
(copy)



Boolean Circuits

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

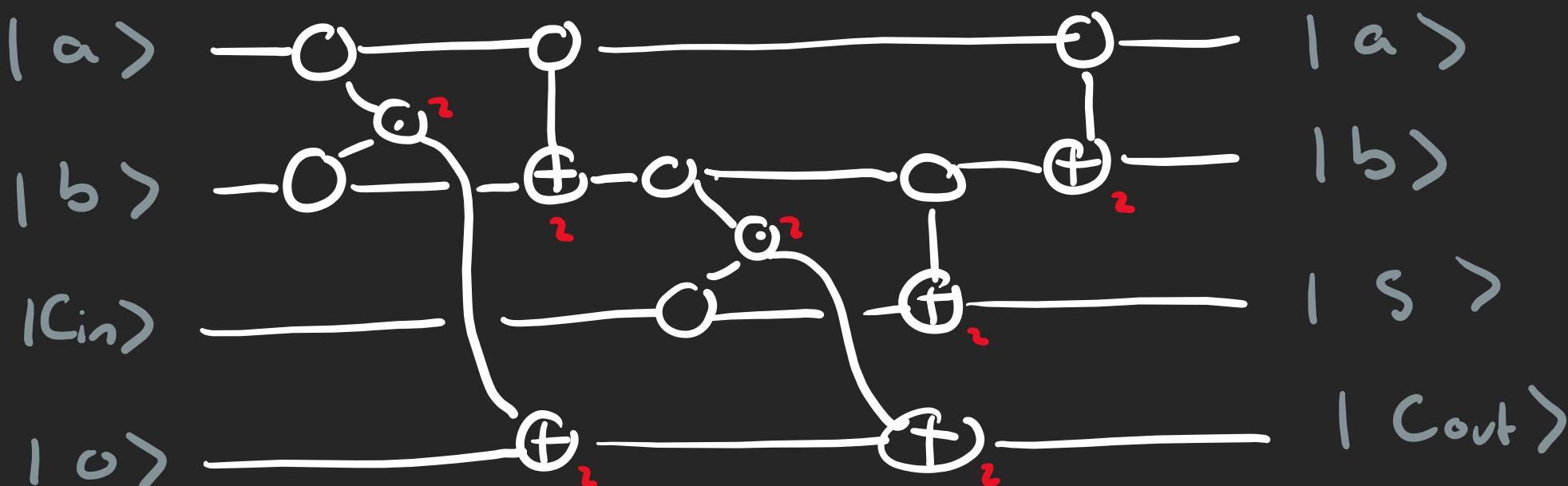
Sum with carry



Boolean Circuits

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

Quantum
Full Adder

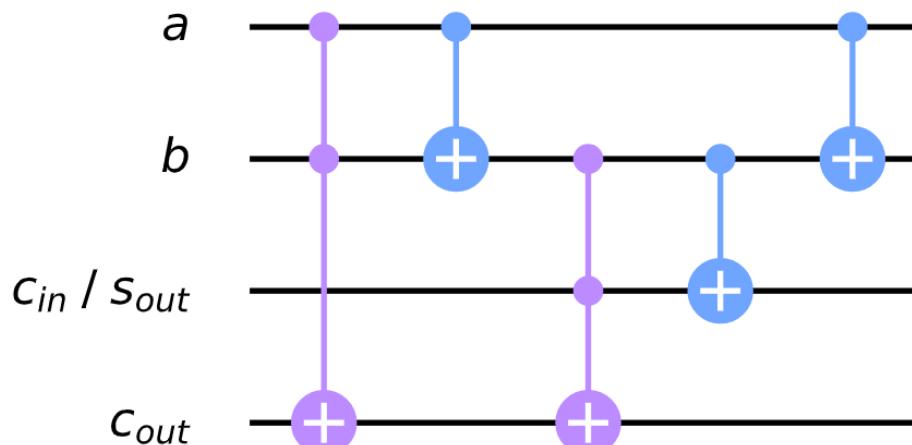


↑ "ancilla", starting at $|o\rangle$

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

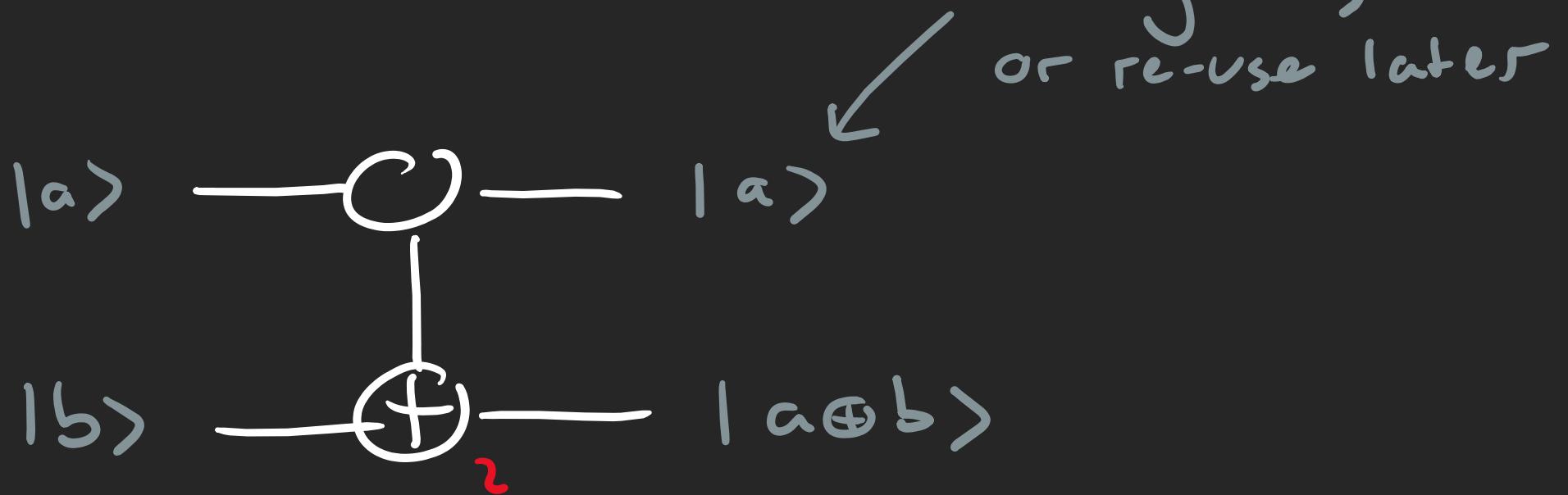
Boolean Circuits

$$\begin{array}{c} \text{Diagram 1: } |b\rangle \xrightarrow{\text{CNOT}} |b\rangle \\ \text{Diagram 2: } |b\rangle \xrightarrow{\text{CNOT}} |b\rangle \end{array} = \frac{1}{2} \begin{array}{c} \text{Diagram 3: } |b\rangle \xrightarrow{\text{CNOT}} |b\rangle \\ \text{Diagram 4: } |b\rangle \xrightarrow{\text{CNOT}} |b\rangle \end{array} + \frac{1}{2} \begin{array}{c} \text{Diagram 5: } |b\rangle \xrightarrow{\text{CNOT}} |b\rangle \\ \text{Diagram 6: } |b\rangle \xrightarrow{\text{CNOT}} |b\rangle \end{array}$$

\nwarrow ancilla in $|0\rangle$

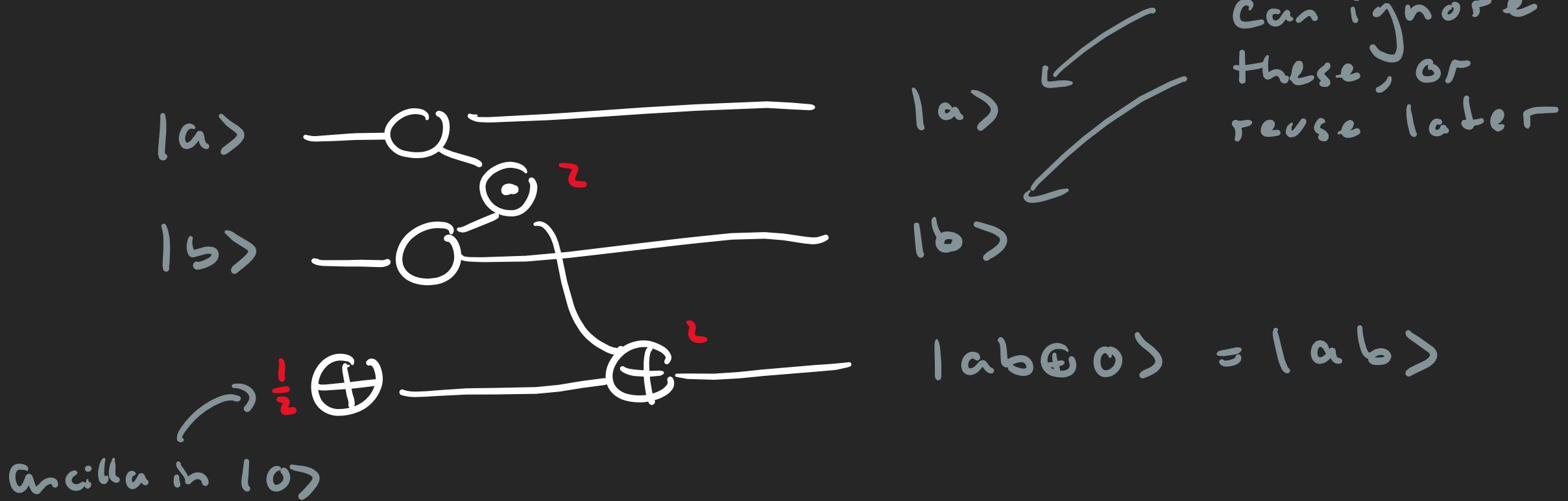
COPY

Boolean Circuits



X OR
 \sim

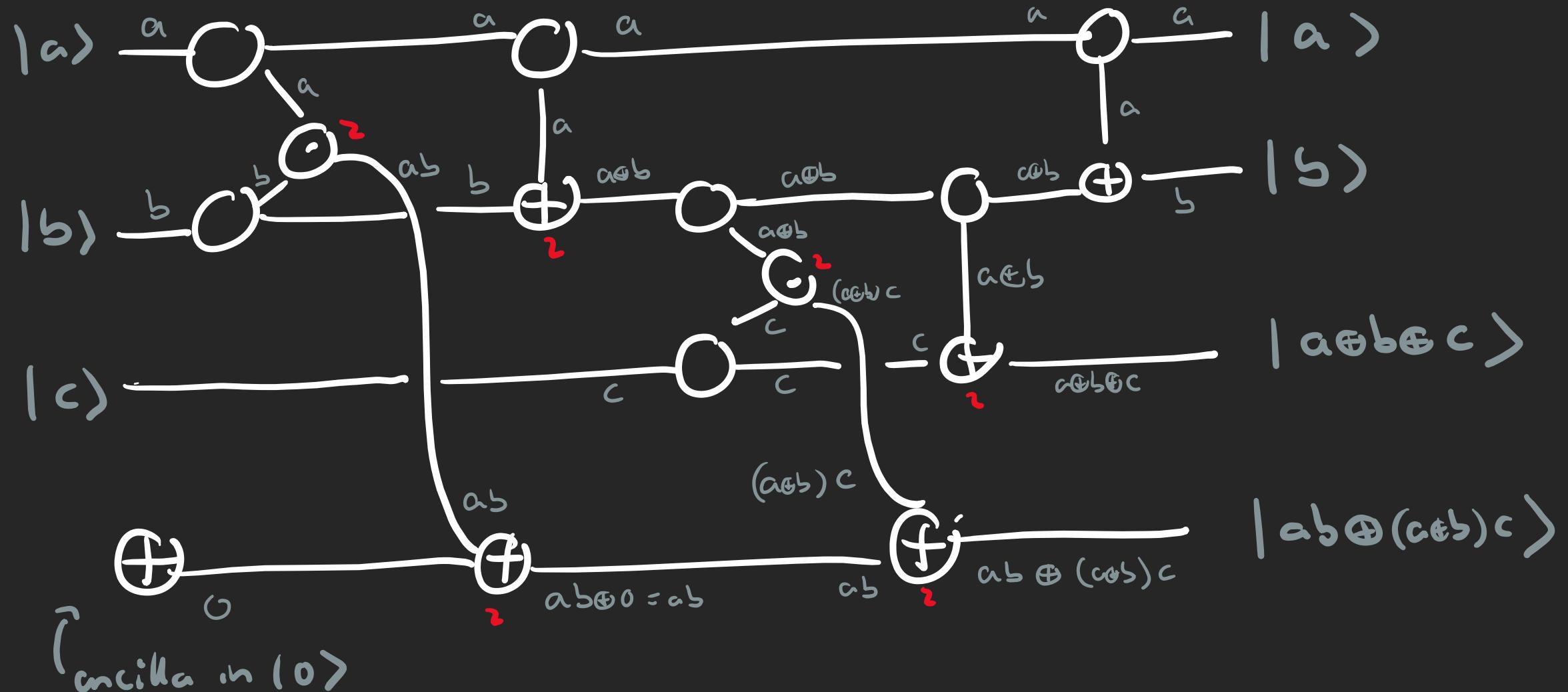
Boolean Circuits



A N D

Boolean Circuits

Quantum Circuit for
a FULL ADDER



Clifford Circuits

Clifford Circuits

= all combinations of

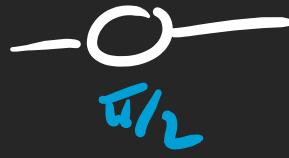
{

- 1-qub.1 stabilizer states
- 1-qubit Clifford gates
- the 2-qub.1 cx gate

}

Clifford Circuits

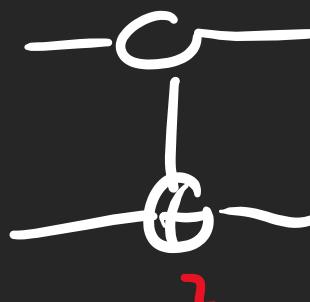
Examples of 2-qubit Clifford Circuits:



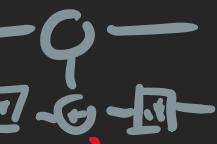
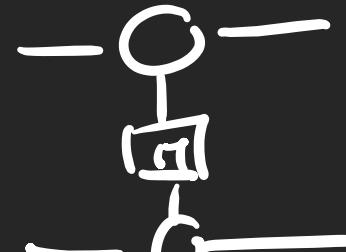
$\pi/2$



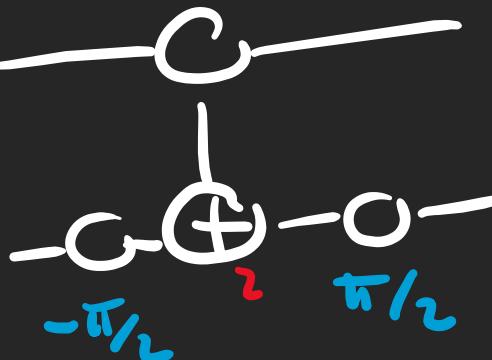
$\pi/2$



C_2



=



$-\pi/2$

$\pi/2$

Sue A P



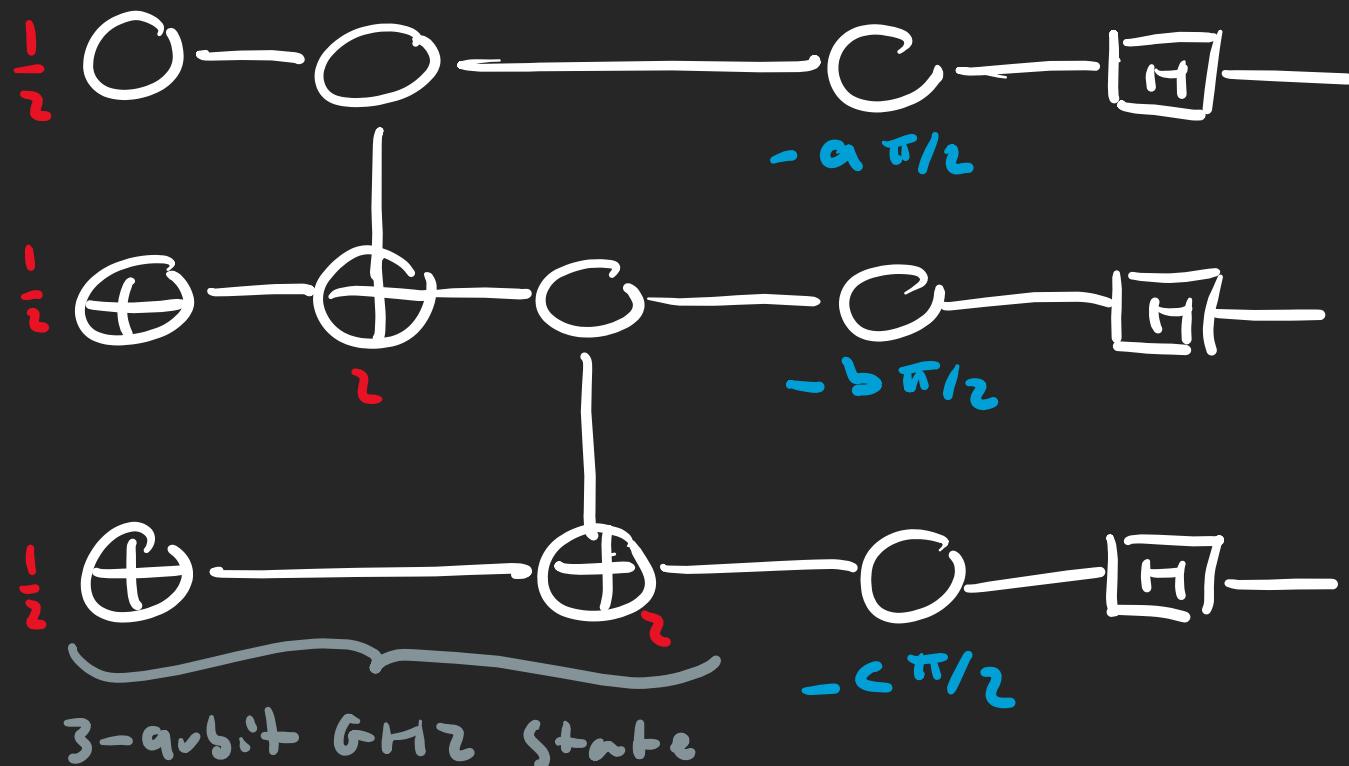
$\pi/2$

$\pi/2$

$\pi/2$

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.

⇒ 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!)



Clifford+T Circuits

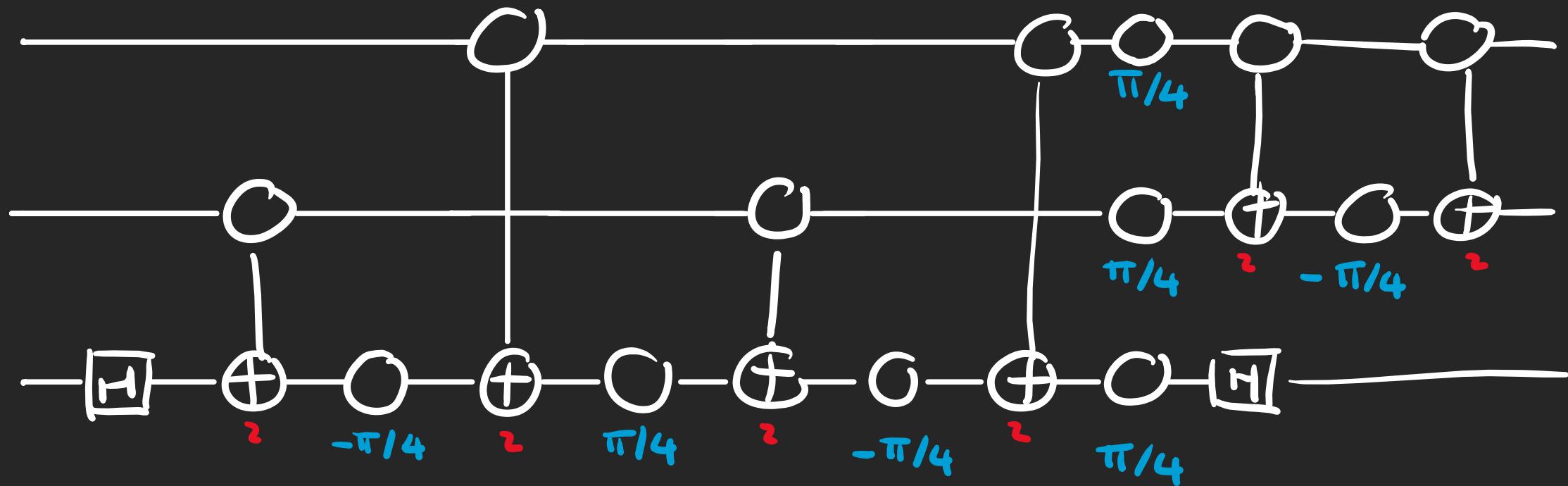
T gates are the typical non-Clifford resource.

\Rightarrow Clifford + \overline{T} circuits

(approximately universal for QC)

of T gates \sim classical hardness

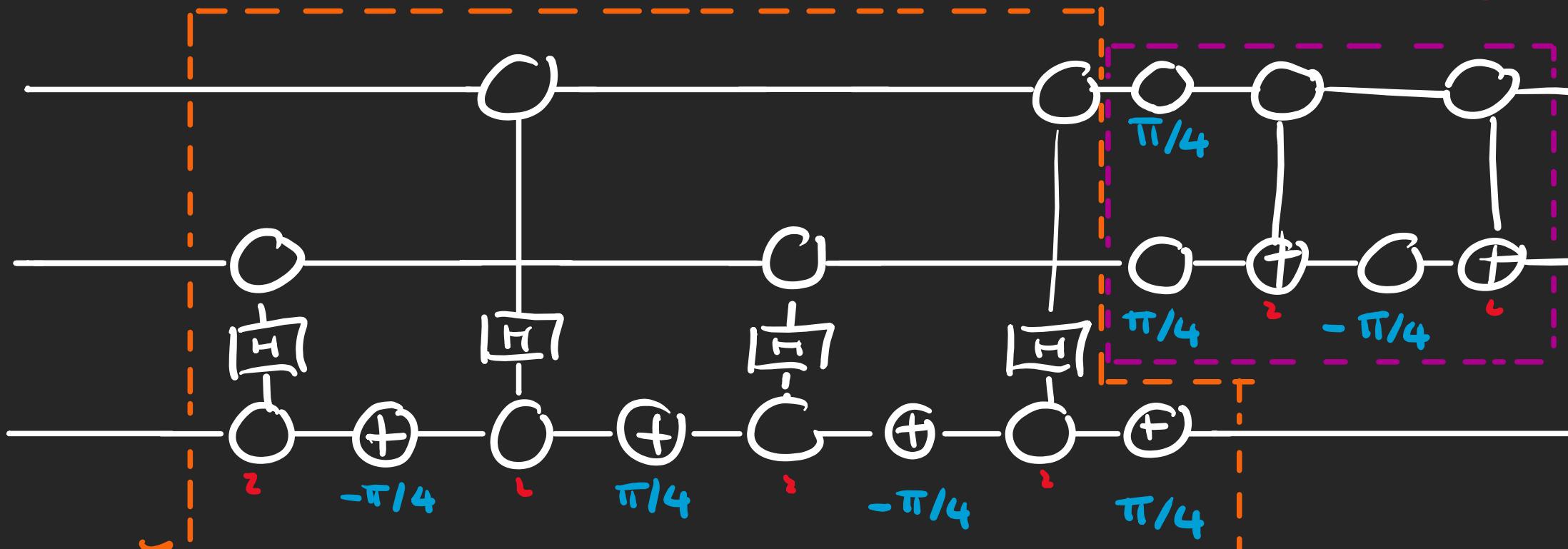
The Toffoli (CCX) Gate



Minimum # of $\pi/4$ (non-Clifford) rotations: 7

The Toffoli (CCX) Gate

Z rotation and
controlled Z rotation
to fix "kickbacks"
from CZ gates,

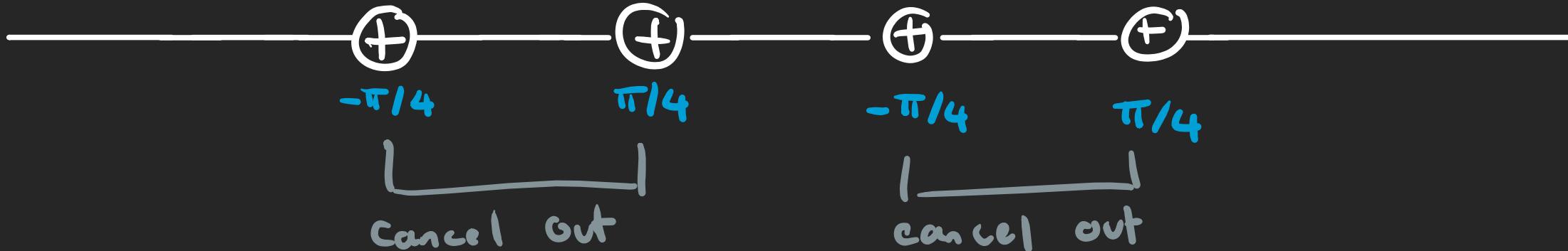
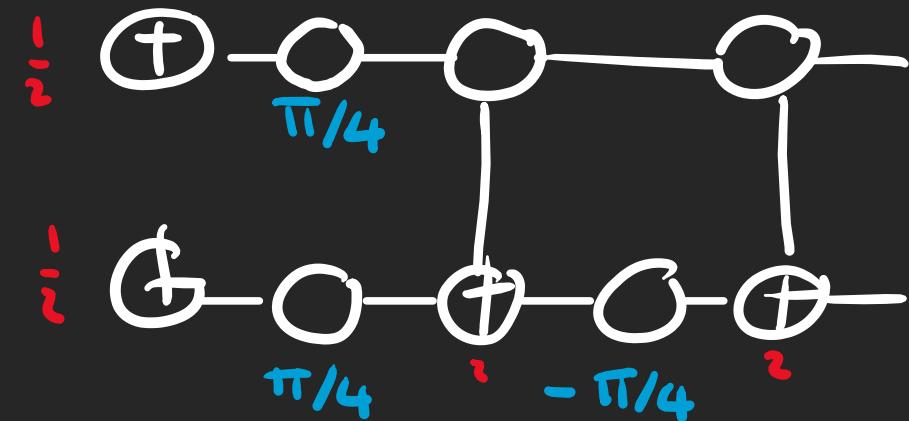


Controlled X rotation

$$\left\{ \begin{array}{ll} 00 : & -\pi/4 + \pi/4 - \pi/4 + \pi/4 = 0 \\ 01 : & -(-\pi/4 + \pi/4) - \pi/4 + \pi/4 = 0 \\ 10 : & -\pi/4 - (\pi/4 - \pi/4) + \pi/4 = 0 \\ 11 : & -(-\pi/4) + \pi/4 - (-\pi/4) + \pi/4 = \pi \end{array} \right.$$

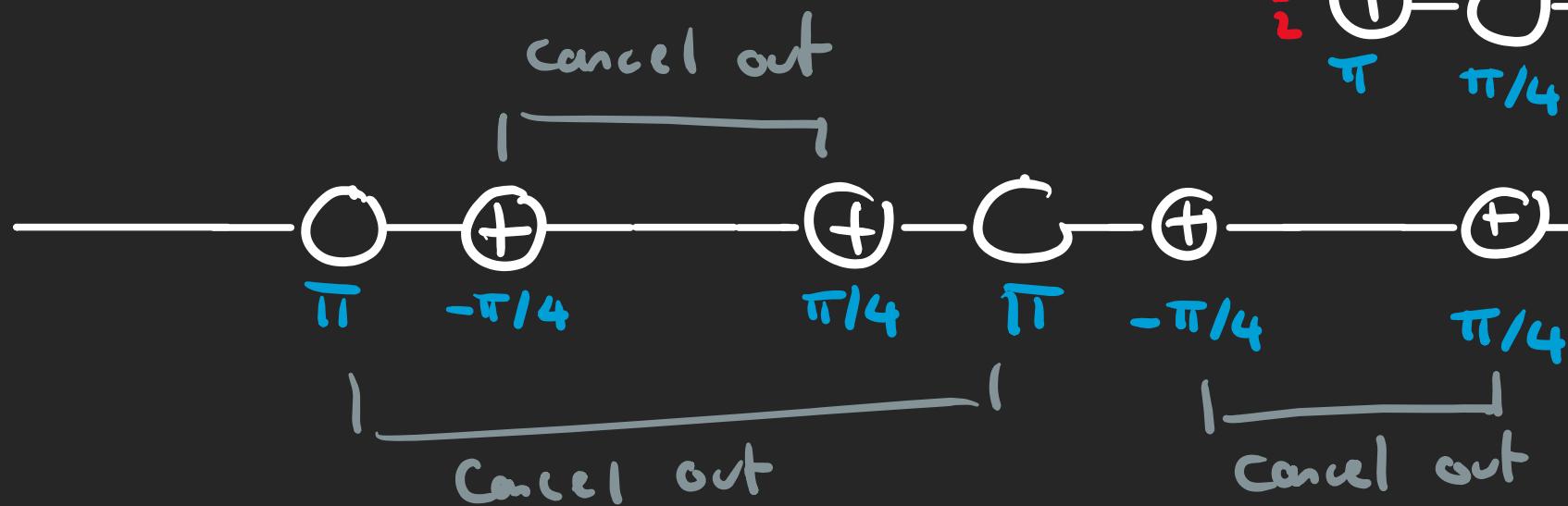
The Toffoli (CCX) Gate

Controls in $|00\rangle$
 $\Rightarrow R_x(0)$ on target



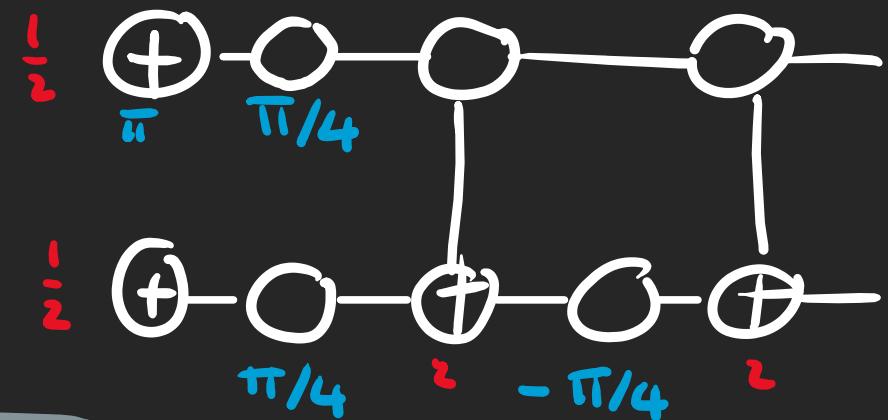
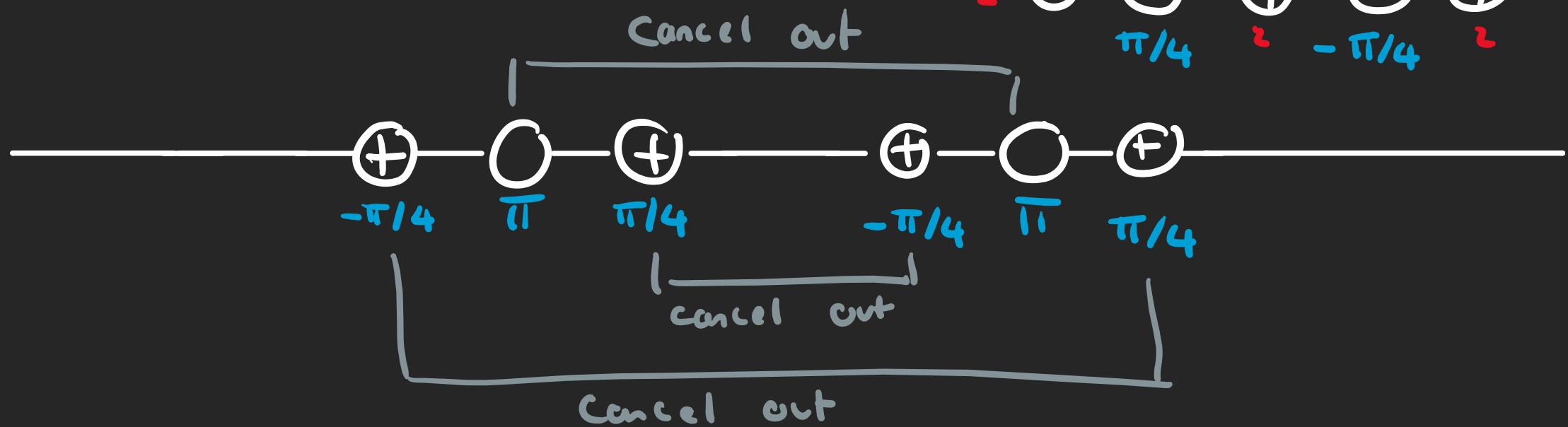
The Toffoli (CCX) Gate

Controls in $|01\rangle$
 $\Rightarrow R_x(0)$ on target



The Toffoli (CCX) Gate

Controls in $|10\rangle$
 $\Rightarrow R_x(0)$ on target



The Toffoli (CCX) Gate

Controls in $|11\rangle$
 $\Rightarrow R_x(\pi)$ on target

