Qubits

- states $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ represent the classical bits 0 and 1
- a *qubit* $\in \mathbb{C}^2 = \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = c_0|0\rangle + c_1|1\rangle$, where $|c_0|^2 + |c_1|^2 = 1$
- probability of qubit being measured as $|0\rangle = |c_0|^2$

  probability of qubit being measured as $|1\rangle = |c_1|^2$

Multiple qubits

- $|00\rangle = |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$ $\qquad |01\rangle = |0\rangle \otimes |1\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$

  $|10\rangle = |1\rangle \otimes |0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$ $\qquad |11\rangle = |1\rangle \otimes |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$
- an arbitrary 2-qubit state: $c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle$
- example: $\frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$
- $|00000000\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \ldots \quad |11111111\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$ with $2^8 = 256$ rows

Rules for tensor products

- $\otimes$ distributes over $+$:

  $$|A\rangle \otimes \Big(|B\rangle + |C\rangle\Big) = |A\rangle \otimes |B\rangle + |A\rangle \otimes |C\rangle$$

  $$\Big(|A\rangle + |B\rangle\Big) \otimes |C\rangle = |A\rangle \otimes |C\rangle + |B\rangle \otimes |C\rangle$$

- scalar multiplication "semi-distributes" over $\otimes$:

  $$\alpha\Big(|A\rangle \otimes |B\rangle\Big) = \alpha|A\rangle \otimes |B\rangle = |A\rangle \otimes \alpha|B\rangle$$

- "parallel" operations:

  $$(A \star C) \otimes (B \star D) = (A \otimes B) \star (C \otimes D)$$

  special case:

  $$(A \star V_1) \otimes (B \star V_2) = (A \otimes B) \star (V_1 \otimes V_2)$$

Example of parallel operations

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 4 & -1 \\ 2 & 1 \end{bmatrix} \qquad V_1 = \begin{bmatrix} 3 \\ 5 \end{bmatrix} \qquad V_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

We can apply A to V_1 and B to V_2 seperately:

$$A \star V_1 = \begin{bmatrix} 5 \\ 3 \end{bmatrix} \qquad B \star V_2 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

and then combine the results:

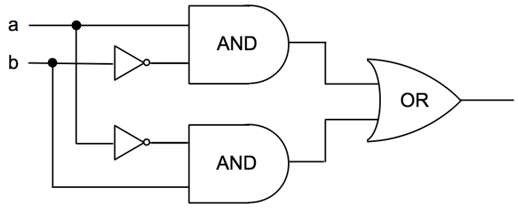
$$(A \star V_1) \otimes (B \star V_2) = \begin{bmatrix} 5 \\ 3 \end{bmatrix} \otimes \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 10 \\ 20 \\ 6 \\ 12 \end{bmatrix}$$

Or: we can combine the operations as $A \otimes B$ and the vectors as $V_1 \otimes V_2$, and apply the combined operation to the combined vectors:

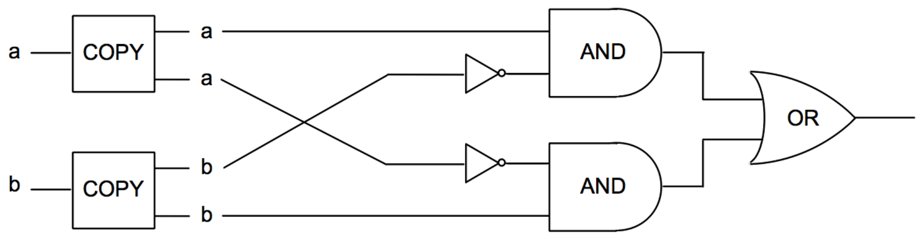
$$A \otimes B = \begin{bmatrix} 0 & 0 & 4 & -1 \\ 0 & 0 & 2 & 1 \\ 4 & -1 & 0 & 0 \\ 2 & 1 & 0 & 0 \end{bmatrix} \qquad V_1 \otimes V_2 = \begin{bmatrix} 3 \\ 6 \\ 5 \\ 10 \end{bmatrix} \qquad (A \otimes B) \star (V_1 \otimes V_2) = \begin{bmatrix} 10 \\ 20 \\ 6 \\ 12 \end{bmatrix}$$

A circuit for XOR

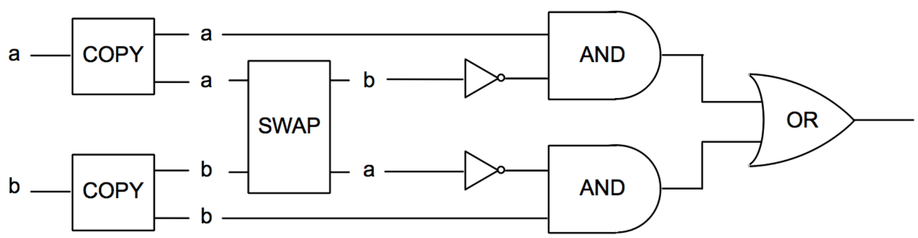
a XOR b = (a AND (NOT b)) OR ((NOT a) AND b)



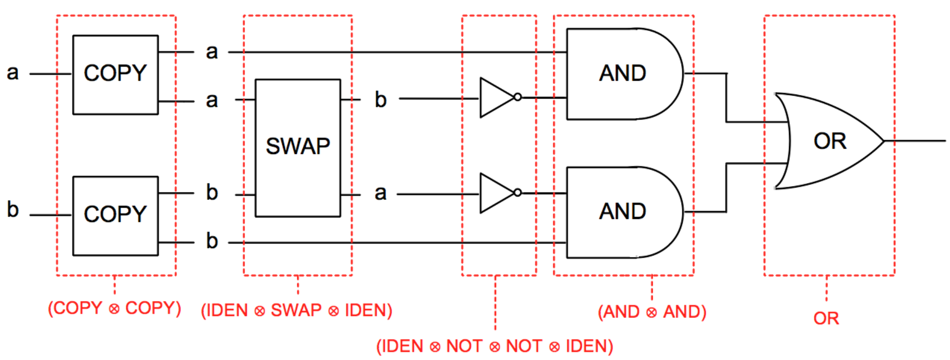
The solid black dots in the above diagram correspond to bit-copying operations (also called “fanout” operations). To write an expression for this circuit in terms of matrix multiplications and tensor products, we first need to make these bit-copying operations explicit. We can use two *COPY* gates:



The above circuit still contains two crossed wires, which corresponds to swapping bits. We can make this explicit with a *SWAP* gate:



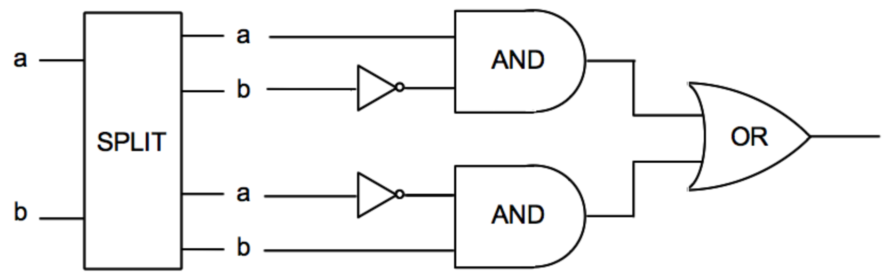
We can now express the circuit as a combination of matrix and tensor product operations:



$OR \star (AND \otimes AND) \star (IDEN \otimes NOT \otimes NOT \otimes IDEN) \star (IDEN \otimes SWAP \otimes IDEN) \star (COPY \otimes COPY)$

Another approach is to use a *SPLIT* operation, which produces two copies of its input bits according to the following truth table:

```
SPLIT:  A  B |  A  B  A  B
        -----+------------
        0  0 |  0  0  0  0
        0  1 |  0  1  0  1
        1  0 |  1  0  1  0
        1  1 |  1  1  1  1
```



This avoids the need for an intervening *SWAP* gate. We can then express the circuit as the combination of matrix multiplications and tensor products below:

$OR \star (AND \otimes AND) \star (IDEN \otimes NOT \otimes NOT \otimes IDEN) \star SPLIT$

Universality of the NAND gate