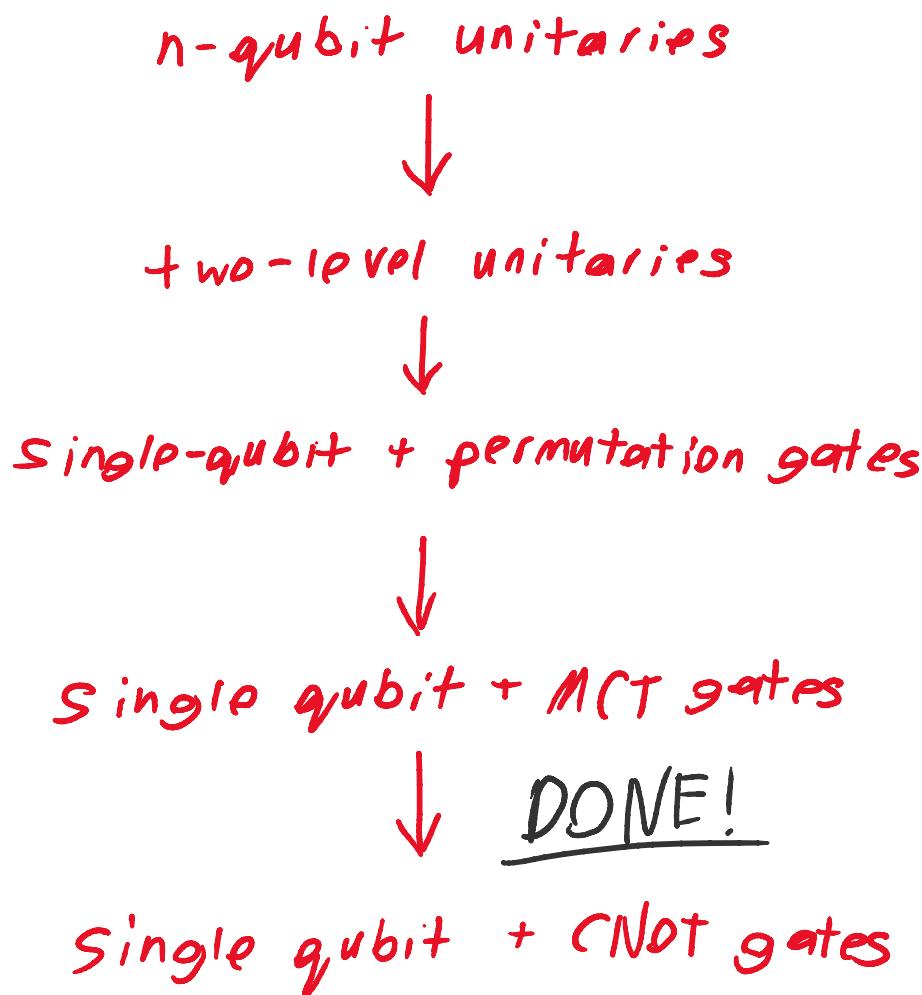


Universal construction of Unitaries

We're now ready to tackle the generic problem of given an n -qubit unitary $U: \mathbb{C}^{2^n} \rightarrow \mathbb{C}^{2^n}$, implement it over a fixed (but infinite, for now) set of unitary gates on 1 or 2 qubits. We will progress down through progressively smaller gate sets as below, showing that each can be constructed over the simpler set.



Construction by two-level unitaries

A **two-level** unitary is a unitary matrix that acts non-trivially on only two rows and columns

E.g.

$$H_{1,3} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

is a two-level H gate
on rows/columns 1 & 3



dimensions worked from notation

Thm.

Any $d \times d$ unitary matrix U can be factorised as a product of $O(d^2)$ two-level unitaries

$$U = U_1 \cdots U_k, \quad U_i \text{ is two-level}$$

To prove the above theorem, it will help to give a **column lemma**.

Lemma. (Column lemma)

Let $v \in \mathbb{C}^d$ be a unit vector. There exists a series of ~~two-level~~ ^{at most d} $d \times d$ unitary matrices U_1, \dots, U_k such that

$$U_k^+ \cdots U_1^+ v = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

PF.

By induction on the non-zero entries of v .

If v has 1 non-zero entry, then

$$\begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

(Pf. cont.)

If v has ≥ 2 non-zero entries, then

$$\frac{1}{\alpha} \begin{bmatrix} \alpha & \dots & 0 \\ \vdots & \ddots & b^* \\ 0 & \dots & -a \\ 0 & \dots & b \end{bmatrix} \begin{bmatrix} v \\ \vdots \\ \alpha \\ \vdots \end{bmatrix} = \begin{bmatrix} \alpha \\ \vdots \\ 0 \\ \vdots \end{bmatrix}$$

Where $\alpha = \sqrt{1|a|^2 + |b|^2}$. Since the matrix on the left acts trivially on every other row of v , we've reduced the number of non-zero rows, so we're done
(Note: d rows so $\leq d$ operators) □

Now we can prove the theorem, namely that every unitary can be factorized as a product of two-level operators.

Pf.

By induction on d (recall U is a $d \times d$ unitary)

If $d=2$, then U is two-level by definition.

If $d > 2$, then let v be the first column of U .

By the column lemma, there exists a two-level unitary U_1 s.t.

$$U_1^* v = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \text{ so } U_1^* U = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & \ddots & & \\ \vdots & & U' & \\ 0 & & & \end{bmatrix}$$

where U' is a $d-1 \times d-1$ unitary. We can use the ind. hyp. to write U' as a product $U_2 \cdots U_k$ of two-level matrices hence $U_1^* U = U_2 \cdots U_k \Rightarrow U = U_1 U_2 \cdots U_k$ O(d^2)

Exact Synthesis

The two-level method extends to a general method of synthesizing unitaries by reducing to the synthesis of individual d -dimensional states. This will be useful later in characterizing certain gate sets in terms of the unitary operators they implement.

Prop.

Given a set of gates G , if for any unit vector $v \in \mathbb{R}^{d^{\otimes 2}}$ where $R \subseteq \mathbb{C}$ there exists a sequence of k -level, $d \times d$ matrix embeddings g_1, \dots, g_ℓ of gates in G such that $g_1 \cdots g_\ell v = e_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, then any unitary $U \in M_{d \times d}(R)$ can be written as a sequence of k -level embeddings of gates in G .

We will see next that such operators correspond to permutations of controlled gates

Decomposing two-level unitaries

Next we need to show how two-level unitaries may be implemented. We do this in stages.

Recall that for $V = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$,

$$\begin{bmatrix} 1 & & \dots & 0 \\ & \ddots & & \vdots \\ & & \ddots & 0 \\ 0 & \dots & & ab \\ & & & cd \end{bmatrix} = \begin{array}{c} \text{---} \\ | \\ | \\ | \\ -U- \end{array}$$

(Decomposition into permutations & single qubit ops)

Let $U_{i,j}$ be an n-qubit unitary acting as V on rows/columns i, j (i.e. $U_{i,j}$ is a two-level U-gate). Then

$$\begin{array}{c} \text{---} \\ | \\ | \\ | \\ -U_{i,j}- \\ | \\ | \\ | \\ \vdots \end{array} \equiv \begin{array}{c} \text{---} \\ | \\ | \\ | \\ -P- \text{---} \text{---} \text{---} \text{---} P^{-1} \text{---} \\ | \\ | \\ | \\ -U- \\ | \\ | \\ | \\ \vdots \end{array}$$

for some permutation $P: |i\rangle \mapsto |2^n-1\rangle$
 $|j\rangle \mapsto |2^n\rangle$

E.g.

$$\begin{bmatrix} 1 & & & 0 \\ & \ddots & & \vdots \\ & & 1 & 0 \\ 0 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & \ddots & & \vdots \\ & & 1 & 0 \\ 0 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & \ddots & & \vdots \\ & & 1 & 0 \\ 0 & \dots & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & & & 0 \\ & & & ab \\ 0 & & & cd \end{bmatrix}$$

Since we know the multiply-controlled U gate can be implemented with permutations (e.g. Toffoli & CNOT) and single-qubit gates, we're done!

Decomposing into MCT gates

Now that we've reduced our gate set to single qubit and permutation operators, we need to show that permutation operators can be compiled to MCT gates.

Let's do the fun algebraic version!

Fact

A $d \times d$ permutation matrix P is a linear representation of a permutation σ in S_d

E.g. Let $P = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$. Then

↑
Symmetric group

$$P|0\rangle = |1\rangle$$

$P|1\rangle = |2\rangle \rightarrow P$ is the permutation

$$P|2\rangle = |0\rangle$$

$0 \leftrightarrow 2$ in S_3
 $3 \leftarrow 2$

Fact

Every permutation in S_d can be written as a product of adjacent transpositions $(i, i+1) \leftarrow$ swaps i & $i+1$

E.g. $\begin{array}{ccc} 0 & \nearrow & 1 \\ 2 & \leftarrow & 1 \end{array} = \underbrace{(0,1)}_{\text{evaluated right to left, i.e.}} \underbrace{(1,2)}$

evaluated right to left, i.e.

$$0 \rightarrow 0 \rightarrow 1$$

$$1 \rightarrow 2 \rightarrow 2$$

$$2 \rightarrow 1 \rightarrow 0$$

The Gray code

The gray code is a ^{binary} ₁ encoding of numbers such that adjacent numbers differ by a single bit

E.g.

$$\begin{array}{l} 0 \rightarrow 00 \\ 1 \rightarrow 01 \\ 2 \rightarrow 11 \\ 3 \rightarrow 10 \end{array}$$

Fact.

Under the gray encoding, an adjacent transposition $(i, i+1)$ can be implemented by an MCT gate with some negative controls

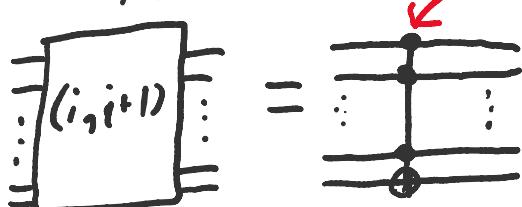
E.g. WLOG assume $i, i+1$ differ in their last bit, i.e.

$$\begin{array}{l} i = x_1 \cdots x_k 0 \\ i+1 = x_1 \cdots x_k 1 \end{array}$$

Then with a single MCT gate we can transpose $i, i+1$ by flipping the final bit if and only if each of the preceding bits are in the state

$$x_1 \cdots x_k$$

Explicitly,



negative if $x_i = 0$

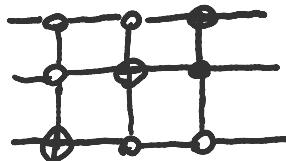
Decomposition into MCT gates

Ihm.

An arbitrary permutation $P: \mathbb{C}^{2^n} \rightarrow \mathbb{C}^{2^n}$ can be implemented using MCT gates

Ex.

To permute $|001\rangle \rightarrow |110\rangle$ we can use the sequence $|001\rangle \rightarrow |000\rangle \rightarrow |010\rangle \rightarrow |110\rangle$, which corresponds to the following circuit



Note:

Generically, implementing a permutation this way would use $O(4^n)$ MCT gates for an n -qubit circuit. A tighter analysis gives $O(n)$ since we can choose **any** permutation that sends

$$|i\rangle \mapsto |i\rangle \quad (\text{gray code order})$$
$$|j\rangle \mapsto |i+1\rangle$$

and in particular we need only flip at most $n-1$ bits. It doesn't really matter anyway because we already have $O(4^n)$ two-level operators, so this won't be efficient in practice!

A note about permutations

Recall that permutations on \mathbb{C}^n are exactly the n -bit reversible functions, so it is useful to study them. A classic theorem relates the even permutations

$$A_d \triangleleft S_d$$

to the existence of ancilla-free circuits over $\{\text{CNOT}, X, \text{Toffoli}\}$

(Note: even permutations are ones with an even number of transpositions. As matrices, P is an even permutation if & only if $\det P = 1$)

Thm.

Let P be a $2^n \times 2^n$ permutation matrix. Then P can be implemented^{without ancillas} over $\{\text{CNOT}, X, \text{Toffoli}\}$ if and only if P is even (i.e. $\det P$ is 1) provided $n > 3$.

Pf sketch.

The backwards direction is easy given that:

$$\det AB = \det A \cdot \det B$$

$$\det A \otimes I_2 = (\det A)^2$$

$$\det X/\text{CNOT}/\text{Toffoli} = -1$$

So any 4 or more qubit circuit over $\{\text{CNOT}, X, \text{Toffoli}\}$ has determinant 1

The other direction can be shown constructively by implementing

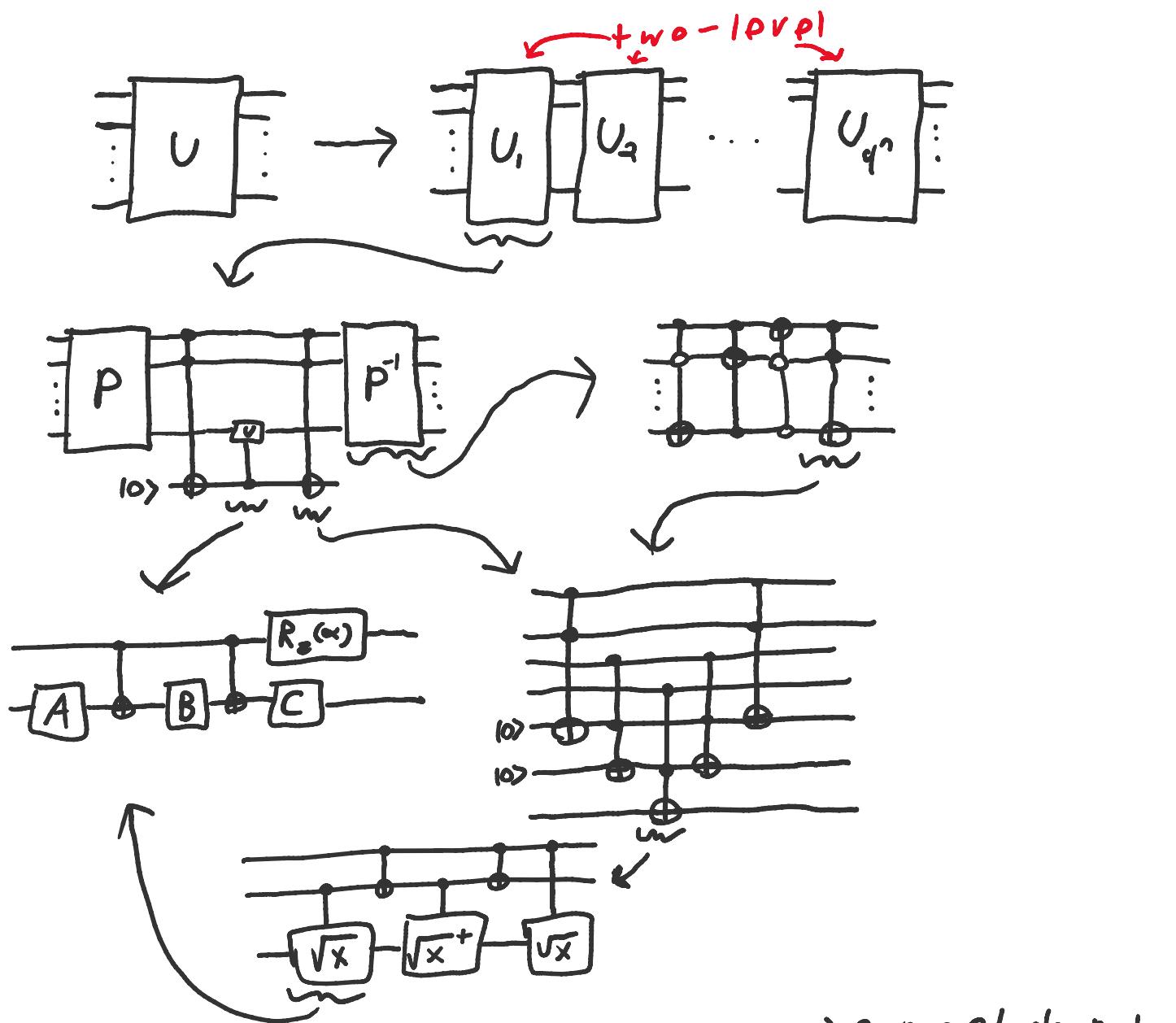
$$\boxed{P} = \boxed{P_1} \text{---} \boxed{P_2} \dots \text{---} \boxed{P_k}$$

where each P_i is on only $n-1$ qubits

best known upper bound is 9
Open question!

Decomposition into single-qubit + CNOT gates

What we've shown



How many gates is that?

→ can get down to $O(n)$

- $O((2^n)^2)$ two-level per Unitary
- $O((2^n)^2)$ MCT per permutation \pm
- $O(n)$ single-qubit + CNOT per MCT

$$\therefore O(n^2 16^n) \text{ (or } O(n^2 4^n))$$

Aside: Optimization

The earliest works in quantum compiler optimization were focused on optimizing this compilation chain. The bottleneck is the decomposition into two-level operators, which Aho & Svore (2003) tackled

dragon book fame Microsoft quantum star

heuristically as a compiler optimization. In a series of papers in 2004, Vartiainen, Mottonen & Salomaa gave more systematic constructions getting this down to $O(4^n)$ (NOT & single-qubit gates). All of this is a bit of a moot point because $O(4^n)$ will kill any speed-up of a quantum algorithm, so attention turned more to efficient constructions of **specific** multi-qubit unitaries. These are however useful (as we will see) **theoretical tools** which are **asymptotically optimal** in the sense that some unitaries require at least $O(4^n)$ (NOT gates).

Thm (Shende, Markov, Bullock 2004)

There exist n -qubit unitaries which cannot be decomposed over CNOT + single qubit gates with fewer than $O(4^n)$ (NOT gates)