

Lecture 9 CPSL 447/547 - Intro to QC

Quantum Compiling II.

Outline

- Proof of universality theorem.
- Other topics in quantum compiling.

* Theorem: Single-qubit and two-qubit gates are computationally universal.

It remains to show "two qubit gates are sufficient".

Recall from last time, Block-Diagonal gate

$$T_i(v) = \begin{bmatrix} & & & \\ \vdots & I_{i-1} & & \\ & & \ddots & \\ & & & v \\ & & & \\ & & & \ddots & \\ & & & & I_{d-i-1} \\ & & & & \vdots & \end{bmatrix}$$

$d = 2^n$,
 I_k = ($k \times k$) Identity matrix
 V = (2×2) unitary matrix
 $1 \leq i \leq d-1$.

$$\begin{cases} T_i(v) |i-1\rangle = v_1|i-1\rangle + v_2|i\rangle & |0\rangle, |1\rangle, |2\rangle, \dots |d-1\rangle. \\ T_i(v) |i\rangle = v_3|i-1\rangle + v_4|i\rangle \\ T_i(v) |j\rangle = |j\rangle, \text{ for } j \neq i, i-1. \end{cases}$$

What if $|v|^2 + |w|^2 \neq 1$

Special V matrix

What if $V \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \sqrt{|\alpha|^2 + |\beta|^2} \\ 0 \end{bmatrix}$. $\xrightarrow{\text{unitary? What's } V?}$

$$V = \begin{bmatrix} \frac{\alpha^*}{c} & \frac{\beta^*}{c} \\ \frac{\beta}{c} & \frac{-\alpha}{c} \end{bmatrix}, \text{ where } c = \sqrt{|\alpha|^2 + |\beta|^2}.$$

"2 amplitudes \Rightarrow 1 amplitude"

Proof of Theorem (*)

Outline of proof:

1. Represent any n-qubit unitary as a product of $O(2^n)$ matrix of **Block-diagonal form**.
2. Decompose each block-diagonal gate into a product of **two-qubit gates**.

Want to find :

$$U = W_{2^n} \cdot \dots \cdot W_2 W_1$$

$$\Rightarrow W_{2^n} \cdot \dots \cdot W_2 W_1 U^{-1} = I$$

Strategy: multiply U^{-1} by W matrices to get I .

Let's take a look at the **first column** of U^{-1}

$$\Gamma_{U^{-1}}^{1,0} \quad 1$$

$$U^{-1} = \begin{bmatrix} u_2 \\ \vdots \\ u_{2^n-3} \\ u_{2^n-2} \\ u_{2^n-1} \\ \dots \end{bmatrix}.$$

Let $v_i \begin{bmatrix} u_{i+1} \\ u_i \end{bmatrix} = \begin{bmatrix} \sqrt{|u_{i+1}|^2 + |u_i|^2} \\ 0 \end{bmatrix}$, $w_i = T_1(v_1) T_2(v_2) \cdots T_{2^n-1}(v_{2^n-1})$

What's $w_i U^{-1} = ?$

- After $T_{2^n-1}(v_{2^n-1})$: $T_{2^n-1}(v_{2^n-1}) U^{-1} = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{2^n-3} \\ \sqrt{|u_{2^n-2}|^2 + |u_{2^n-1}|^2} \\ 0 \end{bmatrix} \dots$

- After $T_{2^n-2}(v_{2^n-2})$: $T_{2^n-2}(v_{2^n-2}) T_{2^n-1}(v_{2^n-1}) U^{-1} = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ \sqrt{|u_{2^n-3}|^2 + |u_{2^n-2}|^2 + |u_{2^n-1}|^2} \\ 0 \\ 0 \end{bmatrix} \dots$
 \vdots

$$\Rightarrow w_i U^{-1} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \dots \end{bmatrix}, \text{ In fact } \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

By same construction,

why? \nearrow

we focus on second column of $w_i U^{-1}$ next.

$$W_2 = T_2(v_2) \cdots T_{2^n-2}(v_{2^n-2}) T_{2^n-1}(v_{2^n-1})$$

where $v_i \begin{bmatrix} u_{i+1} \\ u_i \end{bmatrix} = \begin{bmatrix} \sqrt{|u_i|^2 + |u_{i-1}|^2} \\ 0 \end{bmatrix}$ for u_i from second

$[U_i]$ is a column of $W_i U^{-1}$.

$$W_2 W_1 U^{-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

Question: Does W_2 change values of the first column?

No. Why not?

Finally, continue this construction until:

$$W_2^n W_{2-1}^n \cdots W_2 W_1 U^{-1} = I.$$

Next, implement $T_i(v)$ using two-qubit gates.

Recall from last time, we saw some examples:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & & \\ 0 & 1 & & \\ & & 1 & 0 \\ & & 0 & 1 \end{bmatrix} = \begin{bmatrix} I & \\ & X \end{bmatrix} = T_3(x) = \Lambda(x)$$

$$\text{Toffoli} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = T_7(x) = \Lambda^2(x)$$

$$\text{SWAP} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} = T_2(x)$$

Generalized SWAP and generalized Toffoli:

$$T_i(x) = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & 0 \\ & & 0 & 1 \\ & & & \ddots \\ & & & & 1 & 0 \end{bmatrix}_{\substack{1 \leq i \leq d-1 \\ d=2^n}} \quad \Delta^{n-1}(x) = T_{d-1}(x) = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & 0 \\ & & 0 & 1 \\ & & & \ddots \\ & & & & 1 & 0 \end{bmatrix}$$

- Can we implement $T_i(v)$ using $\Delta^{n-1}(v)$?

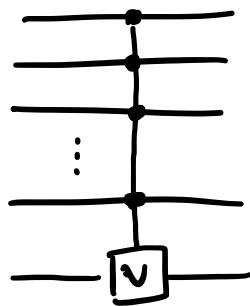
Key: Permutations!

Suppose $P : \begin{cases} |d-2\rangle \rightarrow |i-1\rangle \\ |d-1\rangle \rightarrow |i\rangle \end{cases}$, then

$$T_i(v) = P^\dagger \Delta^{n-1} v P \quad \leftarrow \text{Why?}$$

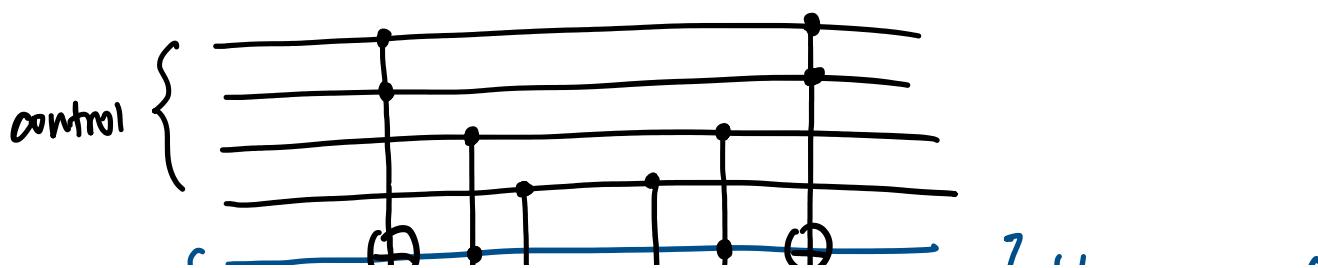
- How to implement P with two-qubit gates? CNOT, Toff.
- How to implement $\Delta^{n-1}(v)$ with two-qubit gates?

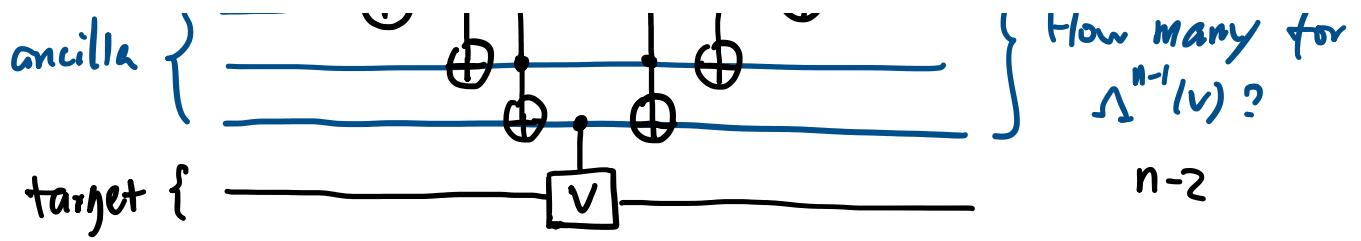
Circuit diagram for $\Delta^{n-1}(v)$:



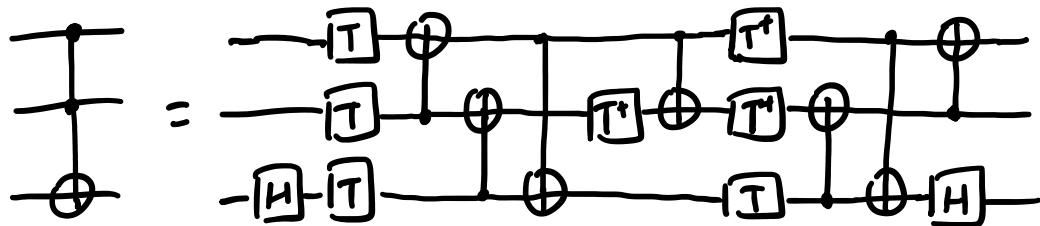
- Can we break up the gate to two-qubit gates?
- Easier task: to three-qubit gates?

Yes, with the help of ancillary (scratch) qubits.





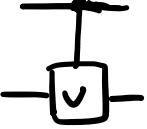
- How to implement Toffoli with two-qubit gates?

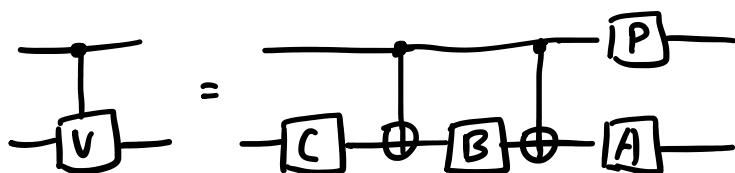


[Amy et al, 2013].

Also see N&C § 4.3

We are done! 😊

Bonus: Implement  with CNOT + single-qubit gates?

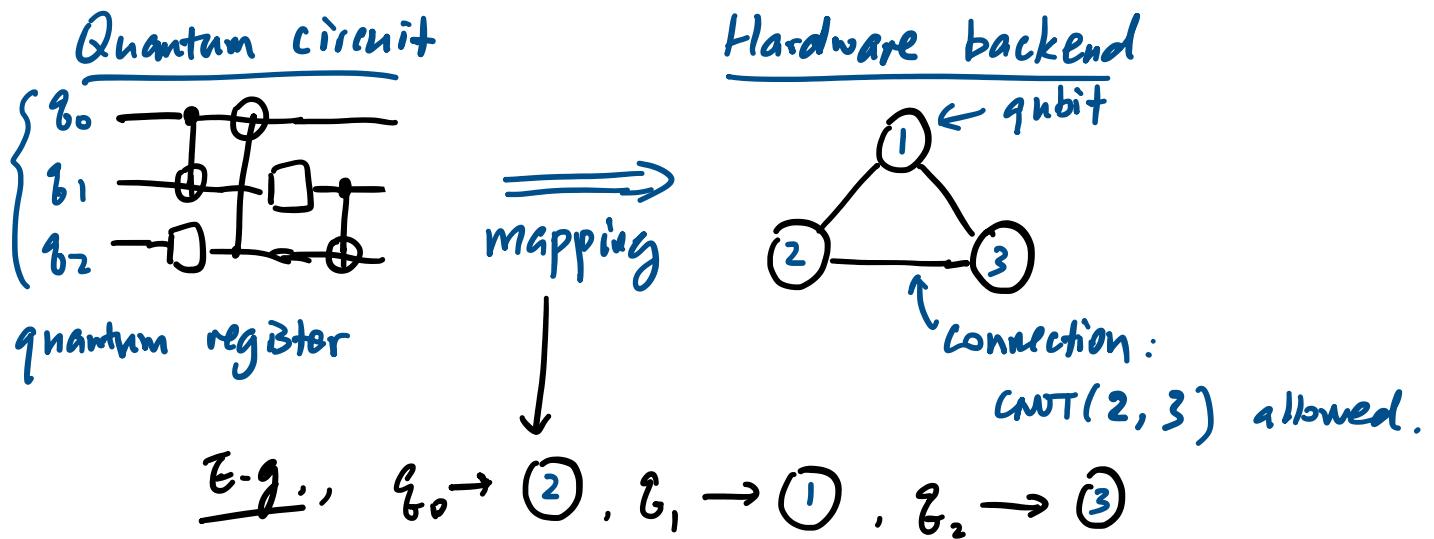


where $V = e^{i\theta}AXBXC$, $ABC = I$, $P = R_z(\theta)$

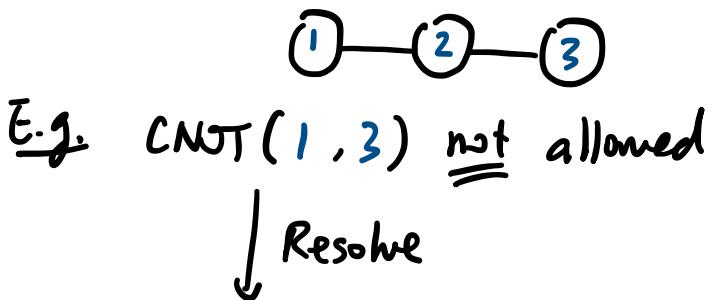
Other topics in quantum compiling:

- Register allocation / Qubit Mapping (HW 2)
- Gate scheduling

- Circuit optimization.



What if backend is not complete graph:



$$\text{SWAP}(2,3) \cdot \text{CNOT}(1,2) \cdot \text{SWAP}(2,3)$$

In general, we need to introduce a chain of swaps to resolve connectivity constraints.