MINI-PROJECTS ON QUANTUM COMPUTING

1. CONSTRAINED OPTIMIZATION
2. STATE PREPARATION
3. ERROR CORRECTION

Keshav Dahiya

Erdős quantum computing bootcamp
Summer 2025

Mini-Project 1: Objectives

- Write a qiskit function that takes three inputs: a pseudo-boolean function $f : \mathbb{F}_2^n \to \mathbb{Z}$, a threshold $t \in \mathbb{Z}$, and a constraint function $C : \mathbb{F}_2^n \to \mathbb{Z}$, and output marker oracle $U_{f,t,C}$ such that

$$U_{f,t,C}|x\rangle_n |y\rangle_1 = \begin{cases} |x\rangle_n |y \oplus 1\rangle_1 & \text{if } f(x) > t \text{ and } C(x) \geq 0 \\ |x\rangle_n |y\rangle_1 & \text{otherwise.} \end{cases}$$

- The construction may use any number of ancillas, arbitrary 1-qubit gates and multi-controlled X gates, (the number of controlled may be arbitrarily large).
- No classical bit and measurements allowed.

ENCODING INTEGER-VALUED FUNCTIONS

- Let $f : \mathbb{F}_2^n \to \mathbb{Z}$ be the function to be encoded. Such a function has to be a polynomial of degree at most n since $x_i^2 = x_i$ for a boolean variable. Moreover, we have $\binom{n}{k}$ monomials of degree k, $0 \leq k \leq n$.
- On the input side we use binary numbers to represent arbitrary monomials. A total of n-qubits are needed for the 2^n possible inputs. A monomial
$$m_{i_0 \cdots i_{n-1}} = x_0^{i_0} x_1^{i_1} \cdots x_{n-1}^{i_{n-1}} \ ,$$
where $i_k \in \{0,1\}$ is represented by the number corresponding to the binary string $i_{n-1} \cdots i_1 i_0$. Then
$$f = \sum_{i_0,\ldots,i_{n-1} \in \mathbb{F}_2} f(i_{n-1} \ldots i_1 i_0)\, m_{i_0 i_1 \ldots i_{n-1}} \equiv \sum_{j=0}^{2^n-1} f(j)\, m_j \ . \qquad (1)$$
- Thus, we can encode f as a list $[f(0), f(1), \ldots, f(2^n - 1)]$ of length 2^n, and $f(j)$ is the coefficient of the monomial corresponding to the binary string representing j.

REPRESENTING INTEGERS USING QUANTUM GATES

- The $U_G(\theta)$ gate when composed after the Hadamard gate generates a geometric sequence

$$U_G(\theta)\, H^{\otimes m}|0\rangle_m = \frac{1}{\sqrt{2^m}} \sum_{a=0}^{2^m-1} e^{\mathrm{i}a\theta}\, |a\rangle_m \ .$$

- Using the inverse quantum Fourier transform we have

$$\mathrm{QFT}^\dagger\, U_G\left(\frac{2\pi f(j)}{2^m}\right) H^{\otimes m}|0\rangle_m = \frac{1}{\sqrt{2^m}} \sum_{a=0}^{2^m-1} e^{\mathrm{i}a2\pi f(j)/2^m}\, |a\rangle_m = |f(j)\rangle_m \ .$$

  This gives representation of an integer on m-qubits.

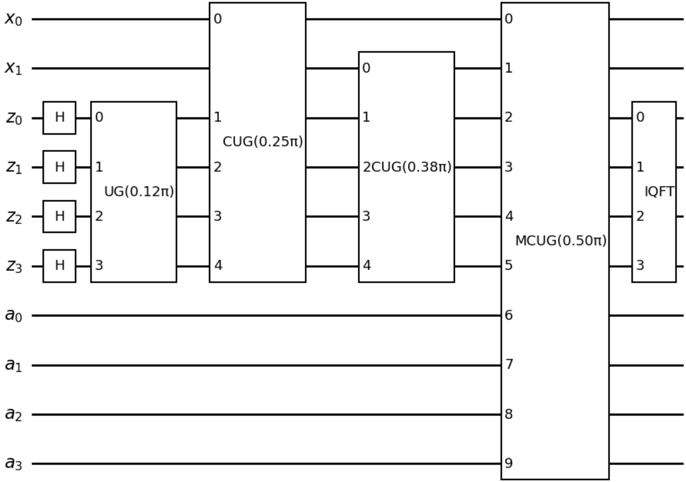
- The representation is understood to be in 2's complement. Thus we have $-2^{m-1} \leq f(j) < 2^{m-1}$, with the understanding that

$$2^{m-1} + z = 2^{m-1} + z - 2^m = -(2^{m-1} - z)\ , \quad 0 \leq z \leq 2^{m-1} - 1\ .$$

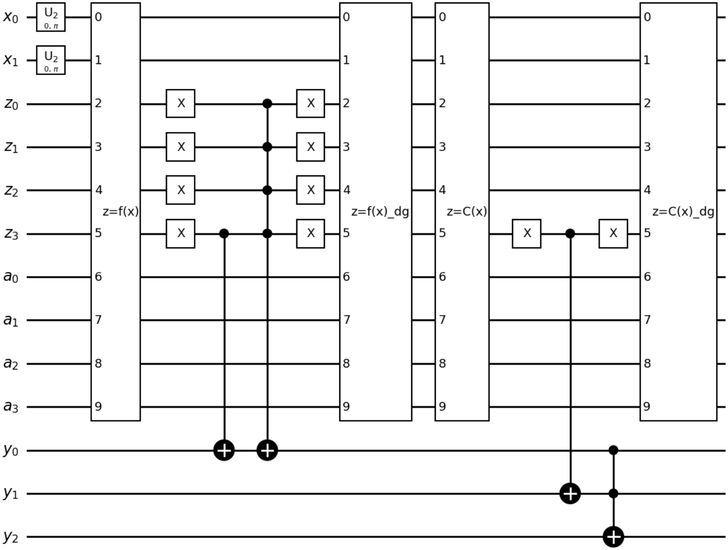
IMPLEMENTING MONOMIALS USING QUANTUM GATES

- A monomial of degree k can be implemented by a k-controlled $U_G(\theta)$ gate.
- The controlled version of $U_G\big(2\pi f(j)/2^m\big)$ is used to implement a monomial $f(j)\, m_j$ in (1) by controlling the m-qubit gate $U_G\big(2\pi f(j)/2^m\big)$ from the input register qubits which correspond to a qubit value of 1.
- For instance, the constant monomial will be not controlled at all, the monomial x_i will be controlled by 1-qubit, namely qubit i, the monomial $x_i x_j$, $(i \neq j)$, will be controlled by qubits i, j, and so on.
- In general a k-degree monomial will be controlled by k-qubits in the input register, precisely those which would be 1 in the binary string representing the corresponding monomial.

Function Encoder Circuit



MARKER ORACLE CIRCUIT



Analysis of Space and Time Complexity

Module	1-qubit Gates	mcx gates	Ancillas
$U_G(\theta)$	m p-gates	0	0
$CU_G(\theta)$	$3m$ p-gates	$2m$	0
$MCU_G(\theta)$	$3m$ p-gates	$3m$	m
IQFT(m)	m h-gates $3m(m-1)/2$ p-gates	$m(2m+1)/2$	0

The overall complexity is as follows:

- $O(n+m)$ H gates and $O(m2^n)$ phase gates,
- $O(m2^n)$ MCX gates,
- $O(n+m)$ qubits and $O(m2^n)$ ancillas,
- $O(2^n+m)$ gate depth.

Thank You !