Constant-depth circuits for Boolean functions and quantum memory devices using multi-qubit gates

Jonathan Allcock[1], Jinge Bao[2], Joao F. Doriguello[2,3], Alessandro Luongo[2], and Miklos Santha[2,4]

[1]Tencent Quantum Laboratory, Hong Kong, China
[2]Centre for Quantum Technologies, National University of Singapore, Singapore
[3]HUN-REN Alfréd Rényi Institute of Mathematics, Budapest, Hungary
[4]CNRS, IRIF, Université Paris Cité

We explore the power of the unbounded Fan-Out gate and the Global Tunable gates generated by Ising-type Hamiltonians in constructing constant-depth quantum circuits, with particular attention to quantum memory devices. We propose two types of constant-depth constructions for implementing Uniformly Controlled Gates. These gates include the Fan-In gates defined by $|x\rangle|b\rangle \mapsto |x\rangle|b \oplus f(x)\rangle$ for $x \in \{0,1\}^n$ and $b \in \{0,1\}$, where f is a Boolean function. The first of our constructions is based on computing the one-hot encoding of the control register $|x\rangle$, while the second is based on Boolean analysis and exploits different representations of f such as its Fourier expansion. Via these constructions, we obtain constant-depth circuits for the quantum counterparts of read-only and read-write memory devices — Quantum Random Access Memory (QRAM) and Quantum Random Access Gate (QRAG) — of memory size n. The implementation based on one-hot encoding requires either $O(n \log^{(d)} n \log^{(d+1)} n)$ ancillae and $O(n \log^{(d)} n)$ Fan-Out gates or $O(n \log^{(d)} n)$ ancillae and $16d - 10$ Global Tunable gates, where d is any positive integer and $\log^{(d)} n = \log \cdots \log n$ is the d-times iterated logarithm. On the other hand, the implementation based on Boolean analysis requires $8d - 6$ Global Tunable gates at the expense of $O(n^{1/(1-2^{-d})})$ ancillae.

1 Introduction

In this work, we study the power of constant-depth quantum circuits with a focus on circuits designed for quantum memory access and the execution of Boolean functions. Our investigation has two aims: firstly, to fill the theoretical gap in our understanding of quantum memory circuits from a computational complexity perspective and, secondly, to assess the practicality of physically implementing these circuits. We believe that the properties and limitations of these circuits can highlight their feasibility and potential for practical implementations. To obtain constant-depth circuits, we leverage multi-qubit "magic" gates like the Fan-Out gate (a generalization of the CNOT

Jonathan Allcock: jonallcock@tencent.com
Jinge Bao: jbao@u.nus.edu
Joao F. Doriguello: doriguello@renyi.hu, www.joaodoriguello.com
Alessandro Luongo: ale@nus.edu.sg
Miklos Santha: cqtms@nus.edu.sg

that can target multiple output qubits) and the multi-qubit entangling Global Tunable gate (that arises from the time evolution of Ising-type Hamiltonians). This analysis explores the potential for quantum memory to be accessed using specialized hardware (designed, for instance, to implement such multi-qubit gates), which may differ from the hardware in general-purpose quantum computers.

1.1 Quantum memory

Quantum memory, besides being an important component of quantum computers from a theoretical perspective, is also fundamental to many quantum algorithms such as Grover's search [Gro97], solving the dihedral hidden subgroup problem [Kup05], collision finding [BHT98], phase estimation for quantum chemistry [BGB+18], pattern recognition and machine learning algorithms [Tru02, Sch03, SS06, KP17, KLP20], cryptanalysis [CL21], state preparation [GR02], among others.

Traditionally, there are two ways — via a Quantum Random Access Memory (QRAM) or a Quantum Random Access Gate (QRAG) — in which memory (classical or quantum) may be accessed quantumly. A QRAM can be seen as a "read-only" gate, while a QRAG can be interpreted as a "read-write" gate since qubits are swapped from memory into the main part of the quantum computer, acted on, and then swapped back.

Quantum random access memory. A QRAM [GLM08a, GLM08b] is the quantum analogue of a classical Random Access Memory (RAM) device that stores classical or quantum data and allows queries to be performed in superposition. More specifically, a QRAM is a device comprising a memory register $\mathtt{M}$ that stores either classical or quantum information, an address register $\mathtt{A}$ that points to the memory cell to be addressed, and a target register $\mathtt{T}$ into which the content of the addressed memory cell is copied. If necessary, it also includes an auxiliary register supporting the overall operation, which is reset to its initial state at the end of the computation. A call to a QRAM (of size n) implements[1]

$$|i\rangle_{\mathtt{A}}|b\rangle_{\mathtt{T}}|x_0,\dots,x_{n-1}\rangle_{\mathtt{M}} \mapsto |i\rangle_{\mathtt{A}}|b\oplus x_i\rangle_{\mathtt{T}}|x_0,\dots,x_{n-1}\rangle_{\mathtt{M}}, \quad \forall x_0,\dots,x_{n-1},b\in\{0,1\},i\in[n].$$

The bits $x_0,\dots,x_{n-1}$ represent the data to be accessed in superposition, which are separate from the qubits in the *work register* of a fully programmable quantum computer.

Quantum random access gate. Another device for random access to a quantum memory is the so-called QRAG, which performs a swap gate between the target register and some portion of the memory register specified by the address register:

$$|i\rangle_{\mathtt{A}}|b\rangle_{\mathtt{T}}|x_0,\dots,x_{n-1}\rangle_{\mathtt{M}} \mapsto |i\rangle_{\mathtt{A}}|x_i\rangle_{\mathtt{T}}|x_0,\dots,x_{i-1},b,x_{i+1},\dots,x_{n-1}\rangle_{\mathtt{M}}, \quad \forall x_0,\dots,x_{n-1},b\in\{0,1\},i\in[n].$$

While QRAG does not enjoy the same level of publicity as QRAMs, its importance lies in its necessity for quantum algorithms for element distinctness and collision finding [Amb07], as well as other quantum algorithms based on random walks on graphs [ACL+20, BLPS22a].

1.2 Multi-qubit "magic" gates

Uniformly Controlled Gate and Fan-In gate. The f-Uniformly Controlled Gate (f-UCG or simply UCG) is the unitary $\sum_{x\in\{0,1\}^n}|x\rangle\langle x|\otimes f(x)$, where $f:\{0,1\}^n\to\mathcal{U}(\mathbb{C}^{2\times 2})$ is a mapping from n-bit strings onto the set $\mathcal{U}(\mathbb{C}^{2\times 2})$ of single-qubit unitaries. UCGs are also known as SELECT operators [LKS24]. Well-known examples of f-UCGs can be found in quantum state preparation

[1] Define $[n]:=\{0,\dots,n-1\}$.

algorithms [GR02, KP17], quantum Monte Carlo algorithms [Mon15] in finance, and HHL-like algorithms [HHL09] in quantum machine learning. The f-UCG is a generalization of many multi-qubit gates including the f-Fan-In gate (f-FIN) defined by the mapping $|x\rangle|b\rangle \mapsto |x\rangle|b \oplus f(x)\rangle$ for a Boolean function $f : \{0,1\}^n \to \{0,1\}$, where $x \in \{0,1\}^n$ and $b \in \{0,1\}$. Note that an f-FIN is simply an f'-UCG with $f'(x) = \mathsf{X}^{f(x)}$. Special cases of f-FINs include OR, AND, PARITY, MAJORITY, and even QRAM, since it can be implemented with $f : \{0,1\}^n \times [n] \to \{0,1\}$, $f(x,i) = x_i$.

General constructions of f-UCGs and f-FINs using single and two-qubit gates can be framed as a unitary synthesis problem. There are several results in this direction for constructing a general n-qubit unitary [BBC+95, Kni95, VMS04, MV05, SMB04, Ros21b]. Sun et al. [STY+23], Yuan and Zhang [YZ23], and Low et al. [LKS24] proposed circuits specifically for f-UCGs using one and two-qubit gates, and therefore not in constant depth. Regarding constructions for controlled gates of the form $|x\rangle\langle x| \otimes \mathsf{U} + \sum_{y \in \{0,1\}^n \setminus \{x\}} |y\rangle\langle y| \otimes \mathbb{I}_m$, where U is an m-qubit gate, see [BBC+95] (using one and two-qubit gates) and [GHMP02, HŠ05, MMN+16, GKH+21] (using multi-qubit entangling gates defined below). While general sequential implementations for f-FINs are folklore, there have been proposals for specific Boolean functions [BBS00] or based on different models of computation like measurement-based quantum computation [DM22]. See Table 4 for a summary of known results.

The Fan-Out gate. The Fan-Out (FO) gate on $n+1$ qubits implements the quantum operation $|b\rangle|x_0,\ldots,x_{n-1}\rangle \mapsto |b\rangle|x_0 \oplus b,\ldots,x_{n-1} \oplus b\rangle$ for all $x_0,\ldots,x_{n-1},b \in \{0,1\}$. In other words, it is a sequence of CNOT gates sharing a single control qubit. For this reason, unlike classical Fan-Out gates, the ability to implement quantum Fan-Out gates as a primitive is not usually taken for granted. Indeed, the Fan-Out gate is powerful in the sense that several interesting results follow from its use, especially connected to constant-depth complexity classes (more on this below). Moore [Moo99] and Green et al. [GHMP02] proved that Fan-Out is equivalent to the PARITY gate. Høyer and Špalek [HŠ05] proved that EXACT$[t]$ gates (which output 1 if the input's Hamming weight is t and 0 otherwise) can be approximated with a polynomially small error by Fan-Out and single-qubit gates in constant depth. These in turn can simulate AND, OR, and THRESHOLD$[t]$ gates. Later, Takahashi and Tani [TT16] managed to prove that EXACT$[t]$ can be simulated *exactly* by Fan-Out and single-qubit gates in constant depth. See Table 4 for a summary of known results.

Unbounded Fan-Out gates that can act on any number of qubits are used in quantum complexity theory (and in this work) to compile certain circuits in constant depth. Even though unbounded Fan-Out gates are just a theoretical construction, bounded Fan-Out gates are within the reach of next-generation quantum hardware [Fen03, ZZY05, RGG+20, YGZ+20, KMN+22, GDC+22, FW23] and can serve as building blocks in larger Fan-Out gates, since an n-arity Fan-Out gate can be simulated by k-arity Fan-Out gates in $O(\log_k n)$-depth, offering interesting trade-offs for hardware implementations. Another approach to implementing Fan-Out gates is the work of Pham and Svore [PS13], who proposed a constant-depth circuit for Fan-Out gates using $O(n)$ ancillae based on measurement-based quantum computation and classical feedback.

The Global Tunable gate. Another powerful and physically implementable gate is the Global Tunable (GT) gate. In its simplest form, it implements a product of two-qubit controlled-Z gates:

$$\prod_{i \neq j \in S} \mathsf{C}_i\text{-}\mathsf{Z}_{\to j}$$

for some subset S of the physical qubits, where $\mathsf{C}_i\text{-}\mathsf{Z}_{\to j}$ denotes a Z gate applied to qubit j controlled on qubit i being in the $|1\rangle$ state (for the general definition see Section 4.2). The first proposal for this kind of gate dates back to Mølmer and Sørensen [MS99], and several experimental implementations have been reported [NMM+14, LWL+19, FOL+19, GBW+20, GFPV+21].

A few studies have explored the use of GT gates in constructing n-qubit Clifford gates [MN18, vdW21, GMNN22, BZC+23, MZ22]. The state-of-the-art construction [BMN22] requires 4 GT gates and n ancilla or 26 GT gates and no ancilla, plus $O(n)$ single-qubit gates. Similarly to the Fan-Out [HŠ05, TT16], the GT gate has been used to implement the unbounded OR gate. Constructions for 4-AND gates using 7 GT gates and no ancillae were reported in [IIV15, MMN+16, MN18]. Regarding general n-arity AND, several constructions [MN18, GWSG20, GMNN22] have been proposed, and improved to the state-of-the-art implementation of [BMN22] using $O(\log^* n)$ GT gates with $O(\log n)$ ancillae or using 4 GT gates with $O(n)$ ancillae, where $\log^* n$ is the star-log function. See Table 4 for a summary of known results.

1.3 Our results

In this work, we propose new constant-depth quantum circuits, based on Fan-Out and GT gates, for f-UCGs, which include f-FINs and certain quantum memory devices as special cases (see Figure 1). We use two different techniques: the first based on the one-hot encoding of the input (also known as indicator function [LKS24], see definition below), and the second based on Boolean analysis of the function f. In Section 3, we formalize our model of quantum computers with quantum access to memory. A Quantum Memory Device (QMD) of size n (assume n to be a power of 2) comprises a $\log n$-qubit address register $\mathtt{A}$, a single-qubit target register $\mathtt{T}$, a poly(n)-qubit auxiliary register $\mathtt{Aux}$, and an n-qubit memory $\mathtt{M}$ consisting of n single-qubit registers $\mathtt{M}_0, \ldots, \mathtt{M}_{n-1}$. A call to the QMD implements

$$|i\rangle_{\mathtt{A}}|b\rangle_{\mathtt{T}}|x_i\rangle_{\mathtt{M}_i}|0\rangle_{\mathtt{Aux}}^{\otimes \operatorname{poly} n} \mapsto |i\rangle_{\mathtt{A}}\mathsf{V}(i)\big(|b\rangle_{\mathtt{T}}|x_i\rangle_{\mathtt{M}_i}\big)|0\rangle_{\mathtt{Aux}}^{\otimes \operatorname{poly} n}, \quad \text{where } \mathsf{V}:[n]\to\mathcal{V} \text{ and } \mathcal{V}\subset\mathcal{U}(\mathbb{C}^{4\times 4})$$

is a $O(1)$-size subset of two-qubit gates. Our model is general enough to include QRAM and QRAG as subcases (by letting $\mathsf{V}(i)$ equal CNOT or SWAP gates). It also includes QMDs that we named f-QRAM for which $\mathsf{V}(i) = \mathbb{I}_1 \otimes |0\rangle\langle 0|_{\mathtt{M}_i} + f(i) \otimes |1\rangle\langle 1|_{\mathtt{M}_i}$, where $f : \{0,1\}^{\log n} \to \mathcal{U}(\mathbb{C}^{2\times 2})$. As far as we know, a general model for a QMD has not been formally defined before. This model allows us to compare the power of different gates with quantum access to memory. In this direction, we show that a QRAG can simulate a QRAM, but not vice-versa, and we discuss the similarities and differences between QMD and f-UCG. In particular, even though f-UCGs do not contain general QMDs, since $\mathsf{V}(i)$ can act non-trivially on two qubits, an f-QRAM of memory size n (i.e., $f : \{0,1\}^{\log n} \to \mathcal{U}(\mathbb{C}^{2\times 2})$) can be seen as an f'-UCG for some function f' on $\{0,1\}^{n+\log n}$ (see Figure 1), since $\mathtt{A}$ and $\mathtt{M}$ work as control registers.

In Section 4, we discuss the Fan-Out and GT gates in more detail. In Section 5, we develop our quantum circuits based on one-hot encoding for any f-UCG, $\sum_{x\in\{0,1\}^n} |x\rangle\langle x| \otimes f(x)$. The main idea is to use Fan-Out or GT gates to compute, in parallel, the one-hot encoding $e(x) \in \{0,1\}^{2^n}$ of the control register $|x\rangle$, where $e(x)_j = 1$ if and only if $j = x$, and to apply the single-qubit gate $f(j)$ controlled on the qubit $|e(x)_j\rangle$, for all $j \in \{0,1\}^n$. By the definition of the one-hot encoding, the correct gate $f(x)$ is selected. To perform all controlled single-qubit gates $f(j)$ in parallel, we use the well-known Z-decomposition of single-qubit gates stating the existence of functions $\alpha, \beta, \gamma, \delta : \{0,1\}^n \to [-1,1]$ such that

$$f(j) = e^{i\pi\alpha(j)}\mathsf{Z}(\beta(j))\mathsf{H}\mathsf{Z}(\gamma(j))\mathsf{H}\mathsf{Z}(\delta(j)), \quad \text{for all } j \in \{0,1\}^n,$$

where $\mathsf{H} := \frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ and $\mathsf{Z}(\theta) := \left(\begin{smallmatrix} 1 & 0 \\ 0 & e^{i\pi\theta} \end{smallmatrix}\right)$ for $\theta \in [-1,1]$. By a result of Green et al. [GHMP02] (see also [MN01, HŠ05]), m commuting gates can be performed in parallel with the aid of $m-1$ ancillae and 2 Fan-Out gates, or simply 1 GT gate and no ancillae. All the $\mathsf{Z}(\delta(j))$ gates can thus be performed in parallel (and similarly for $\mathsf{Z}(\gamma(j))$, $\mathsf{Z}(\beta(j))$, $e^{i\pi\alpha(j)}$).

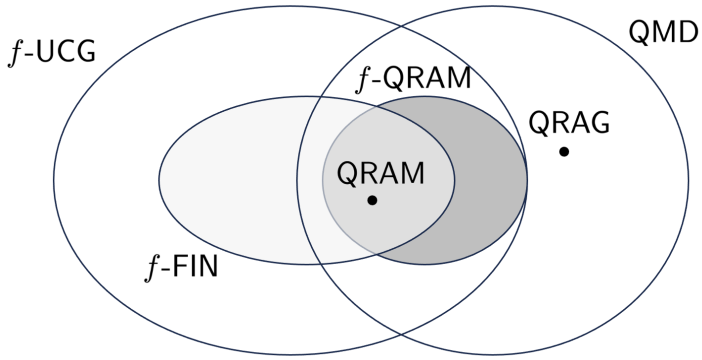


Figure 1: We give constant-depth circuits for f-UCGs, which contain f-FINs and a subset of quantum memory devices (QMD) including QRAM (and its generalization we call f-QRAM) as special cases. Although QRAG is not an f-FIN, our (one-hot-encoding-based) construction for QRAM can be adapted to it.

Naively, one can compute the one-hot encoding of the whole input x. However, if f is a junta, i.e., it depends on only a few coordinates, one only needs to compute the one-hot encoding of the coordinates on which it depends. More generally, this "compression" idea can be extended to a concept we introduce and call (J, r)-junta, where $J \subseteq [n]$ and $r \in \mathbb{N}$. We say $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$ is a (J, r)-junta if, by fixing the coordinates in $\overline{J} := [n] \setminus J$ to any value, the resulting restriction of f to J is an r-junta, i.e., it depends on at most r of its input coordinates. A fine example of a (J, r)-junta is QRAM, since by fixing the coordinates of input i, the resulting restriction is a 1-junta (as it depends only on x_i). It is possible to take advantage of this property and simplify our circuit construction: we partition the input x into sub-strings $x_{\overline{J}}$ and x_J and compute the one-hot encoding of $x_{\overline{J}}$ separately from the one-hot encoding of the coordinates in J that the restriction of f depends on. Both one-hot encodings are then used to select the correct $f(x)$ gate as described above. The resources required for our constructions are as follows (see also Table 1).

Result 1 (Informal version of Theorem 26)**.** *Let $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$ be a (J, r)-junta, $|\overline{J}| = t$. The f-UCG can be implemented in constant depth using either $O(2^{t+r}(t + r)\log(t + r))$ ancillae and $O(2^{t+r}(t + r))$ Fan-Out gates or $O(2^{t+r}(t + r))$ ancillae and 9 GT gates. As a corollary, any f'-QRAM of size n, $f' : \{0,1\}^{\log n} \to \mathcal{U}(\mathbb{C}^{2\times 2})$, can be implemented in constant depth using either $O(n \log n \log\log n)$ ancillae and $O(n \log n)$ Fan-Out gates or $O(n \log n)$ ancillae and 9 GT gates.*

We then tailor Result 1 to f-FINs specifically, given their simpler structure compared to f-UCGs. The number of ancillae and Fan-Out gates are asymptotically the same, and the number of GT gates is reduced to 6 (see Table 2). In particular, we apply the f-FIN results to QRAMs and also show how to implement a QRAG in constant depth[2], even though it is not an f-FIN (see Table 3). In the following, $\log^{(d)} n = \log \cdots \log n$ is the d-times iterated logarithm.

Result 2 (Informal version of Theorem 30)**.** *Let $d \in \mathbb{N}$ be a constant. A QRAM of size n can be implemented in $O(d)$-depth using either $O(n \log^{(d)} n \log^{(d+1)} n)$ ancillae and $O(n \log^{(d)} n)$ Fan-Out gates or $O(n \log^{(d)} n)$ ancillae and $16d-10$ GT gates. A QRAG of size n can be implemented in $O(d)$-depth using either $O(n \log^{(d)} n \log^{(d+1)} n)$ ancillae and $O(n \log^{(d)} n)$ Fan-Out gates or $O(n \log^{(d)} n)$ ancillae and $21d - 12$ GT gates.*

In Section 6, we extend ideas from [HŠ05, TT16] to implement f-UCGs in constant depth using tools from the analysis of Boolean functions. We give three slightly different constructions based on

[2] A $O(1)$-depth and $O(n \log n \log\log n)$-size circuit for QRAG using Fan-Out gates had previously appeared in [Ros21b, Lemma 4.3]. We note that the author missed the $\log\log n$-factor.

different representations of a real-valued Boolean function $g : \{0,1\}^n \to \mathbb{R}$. The first representation is the Fourier expansion (over the reals)

$$g(x) = \sum_{S\subseteq[n]} \widehat{g}(S)\chi_S(x), \quad \text{where } \chi_S(x) := (-1)^{\sum_{i\in S} x_i} \text{ and } \widehat{g}(S) = \frac{1}{2^n} \sum_{x\in\{0,1\}^n} g(x)\chi_S(x)$$

are the Fourier coefficients of g. The PARITY function χ_S over $\{-1,1\}$ is called characteristic function. The second representation is based on the existence of a function $p : \{0,1\}^n \to \mathbb{R}$ with a (potentially) sparse Fourier expansion that approximates g up to an additive error $\epsilon > 0$, i.e., $\max_{x\in\{0,1\}^n} |p(x) - g(x)| \leq \epsilon$. Finally, the third representation is the Fourier expansion of g using AND functions instead of PARITY functions, which is sometimes called a real-polynomial representation over $\{0,1\}$,

$$g(x) = \sum_{S\subseteq[n]} \widetilde{g}(S)x^S, \quad \text{where } x^S := \prod_{i\in S} x_i \text{ and } \widetilde{g}(S) = \sum_{T\subseteq S} (-1)^{|S|-|T|} g(T).$$

In the case of Boolean functions $g : \{0,1\}^n \to \{0,1\}$, the above representation over the reals can be "compressed" into a representation over the $\mathbb{F}_2$ field, also known as algebraic normal form, as

$$g(x) = \bigoplus_{S\subseteq[n]} \widetilde{g}_{\mathbb{F}_2}(S)x^S, \quad \text{where } \widetilde{g}_{\mathbb{F}_2}(S) \in \{0,1\} \text{ is given by } \widetilde{g}_{\mathbb{F}_2}(S) = \widetilde{g}(S) \pmod 2.$$

Such relation is true since $\widetilde{g}(S) \in \mathbb{Z}$ for $g : \{0,1\}^n \to \{0,1\}$. The utility of each of the above representations depends on the Boolean properties of g, e.g., its Fourier support $\mathrm{supp}(g) := \{S \subseteq [n] : \widehat{g}(S) \neq 0\}$, (real) $\{0,1\}$-support $\mathrm{supp}_{\{0,1\}}(g) := \{S \subseteq [n] : \widetilde{g}(S) \neq 0\}$, and, for Boolean functions $g : \{0,1\}^n \to \{0,1\}$, its $\mathbb{F}_2$-support $\mathrm{supp}_{\mathbb{F}_2}(g) := \{S \subseteq [n] : \widetilde{g}_{\mathbb{F}_2}(S) \neq 0\}$. Other relevant properties of g are its Fourier support $\mathrm{supp}^{>k}(g) := \{S \subseteq [n] : |S| > k, \widehat{g}(S) \neq 0\}$ at degree greater than k (similarly for $\mathrm{supp}^{=k}(g)$, $\mathrm{supp}^{>k}_{\{0,1\}}(g)$, and $\mathrm{supp}^{>k}_{\mathbb{F}_2}(g)$), its real degree $\deg(g) := \max\{|S| : S \in \mathrm{supp}(g)\}$, its Fourier 1-norm $\widehat{\|g\|}_1 := \sum_{S\subseteq[n]} |\widehat{g}(S)|$, and $\widehat{\|g^{>k}\|}_1 := \sum_{S\subseteq[n]:|S|>k} |\widehat{g}(S)|$.

We can generalize the above properties to operator-valued functions $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$ in an indirect way by applying Boolean analysis to the functions $\alpha, \beta, \gamma, \delta : \{0,1\}^n \to [-1,1]$ arising from f's Z-decomposition and defining, for instance, $\mathrm{supp}(f) := \mathrm{supp}(\alpha) \cup \mathrm{supp}(\beta) \cup \mathrm{supp}(\gamma) \cup \mathrm{supp}(\delta)$ and $\deg(f) := \max\{\deg(\alpha), \deg(\beta), \deg(\gamma), \deg(\delta)\}$. Similar definitions apply to $\mathrm{supp}^{>k}(f)$, $\mathrm{supp}^{=k}(f)$, $\mathrm{supp}_{\{0,1\}}(f)$, and $\mathrm{supp}^{>k}_{\{0,1\}}(f)$. Note that other extensions of Boolean analysis exist in the literature and had been applied to problems in quantum computation [NV00, FS08, BARdW08, MO10, RWZ24]. However, this extension based on Z-decomposition may be of independent interest.

The idea behind our constructions for f-UCGs in Section 6 is to reconstruct the functions $\alpha, \beta, \gamma, \delta$ using one of the aforementioned representations. Consider the Fourier expansion of $\alpha, \beta, \gamma, \delta$ as an example. First we compute the terms $\chi_S(x)$ in parallel using Fan-Out or GT gates, since $\chi_S(x)$ are PARITY functions. Since $\prod_{S\in\mathrm{supp}(\delta)} \mathsf{Z}(\widehat{\delta}(S)\chi_S(x)) = \mathsf{Z}(\sum_{S\in\mathrm{supp}(\delta)} \widehat{\delta}(S)\chi_S(x)) = \mathsf{Z}(\delta(x))$, it is possible to apply $\mathsf{Z}(\delta(x))$ onto a target qubit by simply applying onto this target qubit a sequence of phases $\mathsf{Z}(\widehat{\delta}(S))$ controlled on $\chi_S(x)$, for $S \in \mathrm{supp}(\delta)$. This sequence of controlled phases $\prod_{S\in\mathrm{supp}(\delta)} \mathsf{Z}(\widehat{\delta}(S)\chi_S(x))$ can be performed in constant depth in the case of GT gates by definition. In the case of Fan-Outs, it can be done by using techniques from Høyer and Špalek [HŠ05]. More precisely, first compute a cat state $(|0\rangle^{\otimes m} + |1\rangle^{\otimes m})/\sqrt{2}$ from the target qubit using one Fan-Out, where $m := |\mathrm{supp}(\delta)|$, followed by applying the controlled phases $\mathsf{Z}(\widehat{\delta}(S))$ onto *different* qubits of the cat state. This yields $(|0\rangle^{\otimes m} + (-1)^{\sum_S \widehat{\delta}(S)\chi_S(x)}|1\rangle^{\otimes m})/\sqrt{2} = \mathsf{Z}(\delta(x))(|0\rangle^{\otimes m} + |1\rangle^{\otimes m})/\sqrt{2}$. Finally, uncompute the cat state with another Fan-Out. The same idea applies to α, β, γ and the other two representations (for the real $\{0,1\}$-representation we compute x^S instead of $\chi_S(x)$). The

resources required for our constructions are stated below (see Table 1). In the following, we say that a quantum circuit implements an f-UCG with spectral norm error at most ϵ if it implements an f'-UCG such that the spectral norm $\|f'(x) - f(x)\|$ is at most ϵ for all $x \in \{0,1\}^n$.

Result 3 (Informal version of Theorem 32, Theorem 34, Theorem 35)**.** *Let $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$ with* Z*-decomposition $\alpha, \beta, \gamma, \delta : \{0,1\}^n \to [-1,1]$. We propose constant-depth quantum circuits that implement f-*UCG

- *exactly using*
  - *either $O(\sum_{S\in\mathrm{supp}(f)} |S|)$ ancillae and $O(|\mathrm{supp}^{>1}(f)| + |\bigcup_{S\in\mathrm{supp}^{>1}(f)} S|)$ Fan-Outs,*
  - *or $O(|\mathrm{supp}^{>1}(f)|)$ ancillae and 5* GT *gates;*
- *with spectral norm error at most $\epsilon > 0$ using*
  - *either $O(s\deg(f) + |\mathrm{supp}^{=1}(f)|)$ ancillae and $O(s + |\bigcup_{S\in\mathrm{supp}^{>1}(f)} S|)$ Fan-Outs,*
  - *or $O(s)$ ancillae and 5* GT *gates,*

  where $s := (n/\epsilon^2)\sum_{\nu\in\{\alpha,\beta,\gamma,\delta\}} \|\hat{\nu}^{>1}\|_1^2$;
- *exactly using*
  - *either $O(\sum_{S\in\mathrm{supp}_{\{0,1\}}(f)} |S|\log(1+|S|))$ ancillae and $O(\sum_{S\in\mathrm{supp}^{>1}_{\{0,1\}}(f)} |S|)$ Fan-Outs,*
  - *or $O(\sum_{S\in\mathrm{supp}^{>1}_{\{0,1\}}(f)} |S|)$ ancillae and 9* GT *gates.*

Similarly to the one-hot-encoding-based constructions, we then simplify our Boolean-based constructions to f-FINs, which mainly reduces the number of GT gates (see Table 2), and apply them to QRAMs, thus showing that it is possible to use fewer GT gates at the price of more ancillary qubits (see Table 3). We say that a quantum circuit implements an f-FIN with spectral norm error at most ϵ if it implements an f'-UCG such that $\max_{x\in\{0,1\}^n} \|f'(x) - \mathsf{X}^{f(x)}\| \leq \epsilon$.

Result 4 (Informal version of Theorem 42)**.** *Let $d \in \mathbb{N}$ be a constant. A* QRAM *of size n can be implemented in $O(d)$-depth using either $O(n^{1/(1-2^{-d})}\log n)$ ancillae and $O(n^{1/(1-2^{-d})})$ Fan-Out gates or $O(n^{1/(1-2^{-d})})$ ancillae and $8d-6$* GT *gates.*

Depending on the properties of f, one construction can be more desirable compared to the others when it comes to implementing an f-UCG (similarly for f-FINs). A (J,r)-junta for small $|\overline{J}|$ and r might call for a one-hot-encoding-based construction, while a function with sparse Fourier expansion could be more easily implementable using a Boolean-based circuit. The four different constructions presented above are thus incomparable. Nonetheless, in the worst case, the Boolean-based implementation using the Fourier expansion (Theorem 32 and Theorem 36) requires fewer resources: either $O(2^n)$ Fan-Out gates and $O(2^n n)$ ancillae, or 5 GT gates and $O(2^n)$ ancillae.

Result 5. *Any f-*UCG *with $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$ can be implemented in constant depth using either $O(2^n n)$ ancillae and $O(2^n)$ Fan-Out gates, or $O(2^n)$ ancillae and 5* GT *gates.*

Result	$O(1)$-depth Fan-Out construction		$O(1)$-depth GT construction	
	#Fan-Out	#Ancillae	#GT	#Ancillae
f-UCG $(*)$ Theorem 26	$O(n + 2^{t+r}(t+r))$	$O(2^{t+r}(t+r)\log(t+r))$	9	$O(2^{t+r}(t+r))$
f-UCG Theorem 32	$O\Big(\lvert\operatorname{supp}^{>1}(f)\rvert + \lvert\bigcup_{S\in\operatorname{supp}^{>1}(f)} S\rvert\Big)$	$O\Big(\sum_{S\in\operatorname{supp}(f)} \lvert S\rvert\Big)$	5	$O(\lvert\operatorname{supp}^{>1}(f)\rvert)$
f-UCG $(\ddagger)$ Theorem 34	$O\Big(s + \lvert\bigcup_{S\in\operatorname{supp}^{>1}(f)} S\rvert\Big)$	$O(s\deg(f) + \lvert\operatorname{supp}^{=1}(f)\rvert)$	5	$O(s)$
f-UCG Theorem 35	$O\Big(\sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} \lvert S\rvert\Big)$	$O\Big(\sum_{S\in\operatorname{supp}_{\{0,1\}}(f)} \lvert S\rvert\log(1+\lvert S\rvert)\Big)$	9	$O\Big(\sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} \lvert S\rvert\Big)$

Table 1: Main results for f-UCG, where $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$ has the Z-decomposition $\alpha, \beta, \gamma, \delta : \{0,1\}^n \to [-1,1]$. In $(*)$, f is a (J,r)-junta with $|\overline{J}| = t$. In $(\ddagger)$, the gate is implemented with spectral norm error at most ϵ and $s := (n/\epsilon^2)\sum_{\nu\in\{\alpha,\beta,\gamma,\delta\}} \hat{\|}\nu^{>1}\hat{\|}_1^2$.

Result	$O(1)$-depth Fan-Out construction		$O(1)$-depth GT construction	
	#Fan-Out	#Ancillae	#GT	#Ancillae
f-FIN $(*)$ Theorem 27	$O(n + 2^{t+r}(t+r))$	$O(2^{t+r}(t+r)\log(t+r))$	6	$O(2^{t+r}(t+r))$
f-FIN Theorem 36	$O\Big(\lvert\operatorname{supp}^{>1}(f)\rvert + \lvert\bigcup_{S\in\operatorname{supp}^{>1}(f)} S\rvert\Big)$	$O\Big(\sum_{S\in\operatorname{supp}(f)} \lvert S\rvert\Big)$	2	$O(\lvert\operatorname{supp}^{>0}(f)\rvert)$
f-FIN $(\ddagger)$ Theorem 37	$O\Big(s + \lvert\bigcup_{S\in\operatorname{supp}^{>1}(f)} S\rvert\Big)$	$O(s\deg(f) + \lvert\operatorname{supp}^{=1}(f)\rvert)$	2	$O(s + \lvert\operatorname{supp}^{=1}(f)\rvert)$
f-FIN Theorem 38	$O\Big(\sum_{S\in\operatorname{supp}^{>1}_{\mathbb{F}_2}(f)} \lvert S\rvert\Big)$	$O\Big(\sum_{S\in\operatorname{supp}_{\mathbb{F}_2}(f)} \lvert S\rvert\log(1+\lvert S\rvert)\Big)$	6	$O\Big(\sum_{S\in\operatorname{supp}_{\mathbb{F}_2}(f)} \lvert S\rvert\Big)$

Table 2: Main results for f-FIN, where $f : \{0,1\}^n \to \{0,1\}$. In $(*)$, f is a (J,r)-junta with $|\overline{J}| = t$. In $(\ddagger)$, the gate is implemented with spectral norm error at most ϵ and $s := n\hat{\|}f^{>1}\hat{\|}_1^2/\epsilon^2$.

Result	$O(d)$-depth Fan-Out construction		$O(d)$-depth GT construction	
	#Fan-Out	#Ancillae	#GT	#Ancillae
QRAG Theorem 30	$O(n\log^{(d)} n)$	$O(n\log^{(d)} n\log^{(d+1)} n)$	$21d - 12$	$O(n\log^{(d)} n)$
QRAM Theorem 30	$O(n\log^{(d)} n)$	$O(n\log^{(d)} n\log^{(d+1)} n)$	$16d - 10$	$O(n\log^{(d)} n)$
QRAM Theorem 42	$O(n^{1/(1-2^{-d})})$	$O(n^{1/(1-2^{-d})}\log n)$	$8d - 6$	$O(n^{1/(1-2^{-d})})$

Table 3: Main results for QRAM/QRAG with memory size n. $d \in \mathbb{N}$ and $\log^{(d)} n$ is the d-times iterated logarithm.

1.4 Related work

Constant-depth complexity classes. Recall the main classical classes computed by constant-depth and polynomial-size circuits:

- NC^0 with NOT and bounded AND, OR gates;
- AC^0 with NOT and unbounded AND, OR gates;
- TC^0 with NOT and unbounded AND, OR, THRESHOLD$[t]$ gates for all t;
- $\mathsf{AC}^0[q]$ with NOT and unbounded AND, OR, MOD$[q]$ gates;
- $\mathsf{ACC}^0 = \bigcup_{q>1} \mathsf{AC}^0[q]$.

The study of shallow quantum circuit classes was initiated in [MN98, MN01], which introduced a definition of QNC^0, the quantum analogue of the class NC^0. The remaining quantum analogs of the above circuit classes such as $\mathsf{QAC}^0, \mathsf{QTC}^0, \mathsf{QAC}^0[q]$, and QACC^0 were later defined in [GHMP02]. In

Work	n-qubit Circuit	#Ancillae	#Fan-Out	#GT	Size	Depth
[BBC+95]	Arbitrary Unitary	0	-	-	$O(n^3 2^{2n})$	$O(n^3 2^{2n})$
[Kni95]	Arbitrary Unitary	0	-	-	$O(n2^{2n})$	$O(n2^{2n})$
[VMS04] [MV05]	Arbitrary Unitary	0	-	-	$O(2^{2n})$	$O(2^{2n})$
[HŠ05]	Approx. Exact$[t]$	$O(n\log^{(d)}n)$	$O(n)$	-	-	$O(d)$
	Exact$[t]$	$O(n)$	$O(n)$	-	-	$O(\log^* n)$
	Approx. Threshold$[t]$	$O(n\log n)$	$O(n)$	-	-	$O(1)$
[TT16]	Exact$[t]$	$O(n\log^{(d)}n)$	$O(n)$	-		$O(d)$
	Threshold$[t \leq \log n]$	$O(n\log n)$	$O(n)$	-	-	$O(1)$
	Threshold$[t \geq \log n]$	$O(\sqrt{t\log n})$	$O(n\sqrt{t\log n})$	-	-	$O(1)$
[MN18]	Arbitrary Clifford	0	-	$12n-18$	-	$O(n)$
	Exact$[t]$	$n/2$	-	$3n-6$	-	$O(n)$
[GWSG20]	Exact$[t]$	0	-	$2n$	-	$O(n)$
[vdW21]	Arbitrary Clifford	0	-	$6n-8$	-	$O(n)$
[GMNN22]	Arbitrary Clifford	$n/2$	-	$6\log n+O(1)$	-	$O(\log n)$
	Exact$[t]$	$O(1)$	-	$3n/2$	-	$O(n)$
[BZC+23]	Arbitrary Clifford	0	-	$2n$	-	$O(n)$
[MZ22]	Arbitrary Clifford	0	-	$2\log n+O(1)$	-	$O(\log n)$
[BMN22]	Arbitrary Clifford	0	-	26	-	$O(1)$
	Arbitrary Clifford	$O(n)$	-	4	-	$O(1)$
	Exact$[t]$	$O(n)$	-	4	-	$O(1)$
	Exact$[t]$	$O(\log n)$	-	$O(\log^* n)$	-	$O(1)$
[Ros21b]	QRAG	$O(n\log n\log\log n)$	$O(n\log n)$	-	-	$O(1)$
	Arbitrary Unitary	$\widetilde{O}(2^{2n})$	-	-	$\widetilde{O}(2^{5n/2})$	$\widetilde{O}(2^{n/2})$
[ZLY22]	f-UCG	$O(2^n)$	-	-	$O(n2^n)$	$O(n)$
[STY+23]	Arbitrary Unitary	$O(m)$	-	-	$O(2^{2n})$	$O(n2^n+2^{2n}/(n+m))$
	f-UCG	$O(m)$	-	-	$O(2^n)$	$O(n+2^n/(n+m))$
[YZ23]	Arbitrary Unitary	$O(m)$	-	-	$O(\sqrt{m}2^{3n/2})$	$O(n2^{n/2}+\sqrt{n/m}2^{3n/2})$
[LKS24]	Arbitrary Unitary	$O(m)$	-	-	$O(2^{2n})$	$O(n2^n+2^{2n}/m)$
	QRAM	$2\lceil\log_2 n\rceil+2$	-	-	$O(n)$	$O(n)$
This work	f-UCG	$O(n2^n)$	$O(2^n)$	-	-	$O(1)$
	f-UCG	$O(2^n)$	-	5	-	$O(1)$
	f-FIN	$O(2^n)$	-	2	-	$O(1)$
	QRAM/QRAG	$O(n\log^{(d)}n\log^{(d+1)}n)$	$O(n\log^{(d)}n)$	-	-	$O(d)$
	QRAG	$O(n\log^{(d)}n)$	-	$21d-12$	-	$O(d)$
	QRAM	$O(n\log^{(d)}n)$	-	$16d-10$	-	$O(d)$
	QRAM	$O(n^{(1/(1-2^{-d})})$	-	$8d-6$	-	$O(d)$

Table 4: Summary of some known constructions in the literature regarding number of ancillae, Fan-Out and GT gates, size (number of single and two-qubit gates), and depth. Here d and m are tunable parameters, $\log^{(d)} n$ is the d-times iterated logarithm, and $\log^* n$ is the star-log function. We disregard the size of circuits with Fan-Out and/or GT gates. In [YZ23], $m \in [\Omega(2^n/n), O(2^{2n})]$, while $m \in [\Omega(n), O(n2^n)]$ in [LKS24]. Ref. [LKS24] focuses on minimising the T-count, which is $O(2^n m + 2^{2n}/m)$ for arbitrary unitary. The result of [MZ22] is implicit.

the same paper, the authors introduced expanded versions of the aforementioned classes in which Fan-Out gates are also allowed. For example, the class QAC^0_f consists of problems solvable by constant-depth and polynomial-size quantum circuits composed by Fan-Out gates and unbounded $\mathsf{AND}, \mathsf{OR}$ gates (and similarly for the remaining classes $\mathsf{QTC}^0_f, \mathsf{QAC}^0_f[q], \mathsf{QACC}^0_f$).

Moore [Moo99] and Green et al. [GHMP02] proved that $\mathsf{QAC}^0_f = \mathsf{QAC}^0[q] = \mathsf{QACC}^0$ for any $q > 1$. This result differs greatly from the classical result [Smo87] that $\mathsf{AC}^0[p] \neq \mathsf{AC}^0[q]$ for primes $p \neq q$. The power of Fan-Out was further explored in [HŠ05] who proved that the bounded-error versions of $\mathsf{QNC}^0_f, \mathsf{QAC}^0_f, \mathsf{QTC}^0_f$ are equal. Later, [TT16] managed to collapse the hierarchy of

constant-depth exact quantum circuits: $\mathsf{QNC}^0_f = \mathsf{QAC}^0_f = \mathsf{QTC}^0_f$. This is in sharp contrast to the classical result $\mathsf{NC}^0 \subset \mathsf{AC}^0 \subset \mathsf{TC}^0$. Still, open problems abound, e.g., whether QAC^0 and QAC^0_f are equal or not. In this direction, see [Piu15, PFGT20, Ros21a, NPVY24, ADOY24].

Regarding the class QNC^0 more specifically, it has been an object of great interest since its proposal. A series of works [TD04, Aar05, AC17, BFNV19, BIS+18] gave evidence that sampling from the output distribution of shallow quantum circuits cannot be simulated by polynomial-time classical computers. Recently, a new line of research starting in [BGK18] is focused in proving unconditional separation between the classical and quantum constant-depth circuits [CSV21, LG19, WKST19, BGKT20, GS20, WP23, CCRK23, BBCSN24, GK24, GKMdO24, HMdOS24].

Quantum state preparation. Quantum state preparation is the problem of constructing an n-qubit quantum state $|\psi\rangle$ starting from the initial state $|0\rangle^{\otimes n}$ and classical knowledge of the amplitudes of $|\psi\rangle$. To our knowledge, the first results for efficient state preparation are [Gro00, GR02], the latter using oracle access (implementable with a QRAM) to a set of pre-computed partial integrals. Since then, several constructions have been proposed [BVMS05, PB11, CB18, SLSB19, APPdS21, Bau22, RK22, MGB22, PB11, ZYY21, Ros21b, ZLY22, STY+23, BFLN23, Ros24].

2 Preliminaries

Denote $\mathbb{N} = \{1, 2, \dots\}$ and $[n] := \{0, \dots, n-1\}$. Given $d \in \mathbb{N}$, let $\log^{(d)} n$ be the d-times iterated logarithm defined recursively by $\log^{(d)} n = \log(\log^{(d-1)} n)$ and $\log^{(1)} n = \log n$. The *star-log function* $\log^* n$ is the maximum number of iterations d such that $\log^{(d)} n$ exists and is real. Let $[n]^m$ and $[n]^{\leq m}$ be the set of sequences of size m and size at most m, respectively. We shall often equate the decimal and binary representations of a given number. Given $x = x_0 x_1 \dots x_{n-1} \in \{0,1\}^n$, let $|x|$ be its Hamming weight and $\overline{x}$ its bit-wise negation, i.e., $\overline{x}_i = x_i \oplus 1$ for all $i \in [n]$. The one-hot encoding $e(x) \in \{0,1\}^{2^n}$ of a string $x \in \{0,1\}^n$ is defined such that $e(x)_j = 1$ if and only if $j = x$, $j \in \{0,1\}^n$, and can be calculated as $e(x)_j = \bigwedge_{k\in[n]}(x \oplus \bar{j})_k$. We take logarithms to the base 2. Given $\mathsf{A} \in \mathbb{C}^{n\times n}$, its spectral norm is $\|\mathsf{A}\| := \max_{v\in\mathbb{C}^n:\|v\|_2=1} \|\mathsf{A}v\|_2$. Let $\mathcal{U}(\mathbb{C}^{n\times n})$ be the set of $n \times n$ unitary matrices. Let $\mathbb{I}_n$ be the $2^n \times 2^n$ identity matrix, $\mathsf{X}, \mathsf{Y}, \mathsf{Z}$ the usual Pauli matrices, and H the Hadamard gate. For $\theta \in [-1, 1]$, define $\mathsf{Z}(\theta) := \left(\begin{smallmatrix} 1 & 0 \\ 0 & e^{i\pi\theta} \end{smallmatrix}\right)$.

- Given an ordered sequence $I \in [n]^m$ of m distinct elements and a unitary $\mathsf{U} \in \mathcal{U}(\mathbb{C}^{2^m\times 2^m})$, let $\mathsf{U}_{\to I} \in \mathcal{U}(\mathbb{C}^{2^n\times 2^n})$ be the unitary that applies U onto qubits in I and the identity onto the remaining qubits, i.e., $\mathsf{U}_{\to I}|x\rangle = (\mathsf{U}|x_I\rangle)|x_{\overline{I}}\rangle$. If $I = (i) \in [n]$, write $\mathsf{U}_{\to i}$.
- Given an ordered sequence $I \in [n]^m$ of m distinct elements, $S \subseteq [n] \setminus I$, and a unitary $\mathsf{U} \in \mathcal{U}(\mathbb{C}^{2^m\times 2^m})$, let $\mathsf{C}_S\text{-}\mathsf{U}_{\to I} \in \mathcal{U}(\mathbb{C}^{2^n\times 2^n})$ be the unitary that applies U onto qubits in I controlled on all qubits in S being in the $|1\rangle$ state and the identity onto the remaining qubits (define $\mathsf{C}_\emptyset\text{-}\mathsf{U}_{\to I} := \mathsf{U}_{\to I}$ if $S = \emptyset$). As an example, $\mathsf{C}_S\text{-}\mathsf{X}_{\to i}$ is the X gate applied onto qubit i controlled on qubits in S being in the $|1\rangle$ state (if $|S| = 1$, this is just a CNOT gate).

Let $\mathsf{SWAP}_{i\leftrightarrow j}$ be the gate that swaps qubits $i, j \in [n]$ and $\mathsf{C}_k\text{-}\mathsf{SWAP}_{i\leftrightarrow j}$ its controlled version on qubit $k \in [n] \setminus \{i, j\}$. In the present work, we use a one and two-qubit universal gate set $\mathcal{G}$, e.g., $\{\mathsf{H}, \mathsf{CNOT}, \mathsf{Z}(\theta)\}$, supplemented with the multi-qubit Fan-Out and GT gates defined in Section 4.

Fact 6 (Z-decomposition, [NC10, Theorem 4.1])**.** *Let $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$ be a function onto single-qubit gates. Then there are functions $\alpha, \beta, \gamma, \delta : \{0,1\}^n \to [-1, 1]$ such that*

$$f(x) = e^{i\pi\alpha(x)}\mathsf{Z}(\beta(x))\mathsf{HZ}(\gamma(x))\mathsf{HZ}(\delta(x)), \quad x \in \{0,1\}^n.$$

We say that the tuple $(\alpha, \beta, \gamma, \delta)$ is the Z-decomposition *of f.*

The size/arity of a gate is the number of qubits on which it depends and effects, e.g., a $\mathsf{C}_S\text{-}\mathsf{U}_{\to i}$ gate has arity $|S|+1$. For clarity, we may explicitly denote the arity k of a gate U by writing $\mathsf{U}^{(k)}$. Circuit diagrams in this paper use the following convention for controlled gates. A black circle ($\bullet$) denotes a control that is active when the qubit is in the $|1\rangle$ state, while a white circle ($\circ$) denotes a control that is active when the qubit is in the $|0\rangle$ state (see Figure 3).

2.1 Boolean analysis

For an introduction to Boolean analysis, see [dW08, O'D14]. In the following, we identify a set $S \subseteq [n]$ with its characteristic vector $S \in \{0,1\}^n$ such that $S_i = 1$ if and only if $i \in S$. Given a real-valued Boolean function $f : \{0,1\}^n \to \mathbb{R}$, its (unique) real-polynomial representation, or Fourier expansion, is

$$f(x) = \sum_{S\subseteq[n]} \widehat{f}(S)\chi_S(x), \quad \text{where } \chi_S(x) := (-1)^{S\cdot x} = (-1)^{\sum_{i\in S} x_i}$$

and its Fourier coefficients $\widehat{f} : 2^{[n]} \to \mathbb{R}$ are given by $\widehat{f}(S) = \frac{1}{2^n}\sum_{x\in\{0,1\}^n} f(x)\chi_S(x)$. The Fourier expansion is a multipolynomial expansion over $\{-1,1\}$, i.e., it uses PARITY functions. It is possible to represent a function over $\{0,1\}$, i.e., using AND functions instead. Given $f : \{0,1\}^n \to \mathbb{R}$, its (unique) real-polynomial $\{0,1\}$-representation is

$$f(x) = \sum_{S\subseteq[n]} \widetilde{f}(S)x^S, \quad \text{where } x^S := \prod_{i\in S} x_i \text{ and } \widetilde{f} : 2^{[n]} \to \mathbb{R} \text{ is } \widetilde{f}(S) = \sum_{T\subseteq S} (-1)^{|S|-|T|} f(T)$$

(the formula $\sum_{T\subseteq S}(-1)^{|S|-|T|}f(T)$ is called Möbius inversion). The Fourier expansion (using AND or PARITY functions) is a representation over the *real* field. In the special case of functions with codomain $\{0,1\}$, it is possible to represent them over the field $\mathbb{F}_2$ instead. Given $f : \{0,1\}^n \to \{0,1\}$, its (unique) $\mathbb{F}_2$-polynomial representation (also called algebraic normal form) is

$$f(x) = \bigoplus_{S\subseteq[n]} \widetilde{f}_{\mathbb{F}_2}(S)x^S, \quad \text{where } \widetilde{f}_{\mathbb{F}_2} : 2^{[n]} \to \{0,1\} \text{ is } \widetilde{f}_{\mathbb{F}_2}(S) = \widetilde{f}(S) \pmod 2 = \bigoplus_{x:\mathrm{supp}(x)\subseteq S} f(x),$$

with $\mathrm{supp}(x) := \{i \in [n] : x_i \neq 0\}$. The $\mathbb{F}_2$-representation can be obtained from the real $\{0,1\}$-representation by changing the sum over the reals to a sum over $\mathbb{F}_2$, i.e., $\widetilde{f}_{\mathbb{F}_2}(S) = \widetilde{f}(S) \pmod 2$.

Given the above representations of a function f, there are several important quantities that can be extracted from them. We list various concepts that will be used in the paper.

1. $\mathrm{supp}(f) := \{S \subseteq [n] : \widehat{f}(S) \neq 0\}$ is the Fourier support of f;
2. $|\mathrm{supp}(f)|$ is the sparsity of f;
3. $\mathrm{supp}^{>k}(f) := \{S \subseteq [n] : |S| > k, \widehat{f}(S) \neq 0\}$ (similarly for $\mathrm{supp}^{\leq k}(f)$ and $\mathrm{supp}^{=k}(f)$);
4. $\deg(f) := \max\{|S| : S \in \mathrm{supp}(f)\}$ is the Fourier degree of f;
5. $f^{>k} := \sum_{S\subseteq[n]:|S|>k} \widehat{f}(S)\chi_S$ is the part of f with degree greater than k (similarly for $f^{\leq k}$);
6. $\hat{\|}f\hat{\|}_1 := \sum_{S\subseteq[n]} |\widehat{f}(S)|$ is the Fourier 1-norm of f;
7. $\mathrm{supp}_{\{0,1\}}(f) := \{S \subseteq [n] : \widetilde{f}(S) \neq 0\}$ is the $\{0,1\}$-support of f;
8. $\mathrm{supp}^{>k}_{\{0,1\}}(f) := \{S \subseteq [n] : |S| > k, \widetilde{f}(S) \neq 0\}$ (similarly for $\mathrm{supp}^{\leq k}_{\{0,1\}}(f)$ and $\mathrm{supp}^{=k}_{\{0,1\}}(f)$);

9. $\deg_{\{0,1\}}(f) := \max\{|S| : S \in \mathrm{supp}_{\{0,1\}}(f)\} = \deg(f)$ is the $\{0,1\}$-degree of f;

10. $\mathrm{supp}_{\mathbb{F}_2}(f) := \{S \subseteq [n] : \widetilde{f}_{\mathbb{F}_2}(S) \neq 0\}$ is $\mathbb{F}_2$-support of f;

11. $\deg_{\mathbb{F}_2}(f) := \max\{|S| : S \in \mathrm{supp}_{\mathbb{F}_2}(f)\} \leq \deg(f)$ is the $\mathbb{F}_2$-degree of f;

12. $\mathrm{supp}^{>k}_{\mathbb{F}_2}(f) := \{S \subseteq [n] : |S| > k, \widetilde{f}_{\mathbb{F}_2}(S) \neq 0\}$ (similarly for $\mathrm{supp}^{\leq k}_{\mathbb{F}_2}(f)$ and $\mathrm{supp}^{=k}_{\mathbb{F}_2}(f)$).

Given $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$, consider its Z-decomposition $\alpha, \beta, \gamma, \delta : \{0,1\}^n \to [-1,1]$. We extend the above Boolean definitions to f by defining $\mathrm{supp}(f) := \mathrm{supp}(\alpha) \cup \mathrm{supp}(\beta) \cup \mathrm{supp}(\gamma) \cup \mathrm{supp}(\delta)$ and $\deg(f) := \max\{\deg(\alpha), \deg(\beta), \deg(\gamma), \deg(\delta)\}$. Similar definitions apply to $\mathrm{supp}^{>k}(f)$, $\mathrm{supp}^{\leq k}(f)$, $\mathrm{supp}^{=k}(f)$, $\mathrm{supp}_{\{0,1\}}(f)$, and $\mathrm{supp}^{>k}_{\{0,1\}}(f)$.

More generally, consider a function $f : \{0,1\}^n \to V$, where V is a complex vector space. Given a partition $(J, \overline{J})$ of $[n]$ and $z \in \{0,1\}^{|\overline{J}|}$, we write $f_{J|z} : \{0,1\}^{|J|} \to V$ for the subfunction of f given by fixing the coordinates in $\overline{J}$ to the bit values z. We say that $f : \{0,1\}^n \to V$ is an r-junta for $r \in \mathbb{N}$ if it depends on at most r of its input coordinates, i.e., $f(x) = g(x_{i_1}, \ldots, x_{i_r})$ for some $g : \{0,1\}^r \to V$ and $i_1, \ldots, i_r \in [n]$. We say that $f : \{0,1\}^n \to V$ is a (J,r)-junta for $J \subseteq [n]$ and $r \in \mathbb{N}$ if $f_{J|z} : \{0,1\}^{|J|} \to V$ is an r-junta for any $z \in \{0,1\}^{|\overline{J}|}$. Note that a (J,r)-junta with codomain $\{0,1\}$ can be computed by a $|J|$-partial depth-r decision tree, see [Kum23, Definition 3.4].

3 Quantum memory architectures

In this section, we formally define a model of a quantum computer with quantum access to memory. A simplified model of classical computers can be thought of as (i) a central processing unit (CPU), (ii) a Random Access Memory (RAM) that serves as a temporary storage medium for the CPU to quickly retrieve data, and (iii) auxiliary permanent storage mediums. A RAM constitutes a memory array, an address/input register, and a target/bus/output register. Data is accessed or modified via address lines. When the CPU requires access to the memory, it sends the value from the address register down the address lines, and, depending on the read or write signal, the content of a memory cell is either copied into the target register or stored from the target register into the memory cell. To define a model of a quantum computer with quantum access to memory, it will first be helpful to formally define the quantum processing unit (QPU).

Definition 7 (Quantum Processing Unit)**.** *A Quantum Processing Unit* (QPU) *of size m is defined as a tuple $(\mathtt{I}, \mathtt{W}, \mathcal{G})$ consisting of*

1. *an $m_{\mathtt{I}}$-qubit Hilbert space called* input register $\mathtt{I}$*;*

2. *an $(m - m_{\mathtt{I}})$-qubit Hilbert space called* workspace $\mathtt{W}$*;*

3. *a constant-size universal gate set $\mathcal{G} \subset \mathcal{U}(\mathbb{C}^{4\times 4})$.*

The qubits in the workspace $\mathtt{W}$ are called ancillary qubits or simply ancillae. An input to the QPU*, or quantum circuit, is a tuple $(T, |\psi_{\mathtt{I}}\rangle, C_1, \ldots, C_T)$ where $T \in \mathbb{N}$, $|\psi_{\mathtt{I}}\rangle \in \mathtt{I}$, and, for each $t \in \{1, \ldots, T\}$, $C_t \in \mathcal{I}(\mathcal{G})$ is an instruction from a set $\mathcal{I}(\mathcal{G})$ of possible instructions. Starting from the state $|\psi_0\rangle := |\psi_{\mathtt{I}}\rangle|0\rangle^{\otimes(m-m_{\mathtt{I}})}_{\mathtt{W}}$, at each time step $t \in \{1, \ldots, T\}$ we obtain the state $|\psi_t\rangle = C_t|\psi_{t-1}\rangle \in \mathtt{I}\otimes\mathtt{W}$. The instruction set $\mathcal{I}(\mathcal{G}) \subset \mathcal{U}(\mathbb{C}^{2^m\times 2^m})$ consists of all m-qubit unitaries on $\mathtt{I}\otimes\mathtt{W}$ of the form*

$$\prod_{i=1}^{k} (\mathsf{U}_i)_{\to I_i}$$

for some $k \in \mathbb{N}$, $\mathsf{U}_1, \ldots, \mathsf{U}_k \in \mathcal{G}$ *and pair-wise disjoint non-repeating sequences* $I_1, \ldots, I_k \in [m]^{\leq 2}$ *of at most* 2 *elements. We say that* $\sum_{i=1}^{k} |I_i|$ *is the* arity/size *of the corresponding instruction. We say that* T *is the* depth *of the input to the* QPU*, while its* size *is the sum of the arities/sizes of the instructions* $C_1, \ldots, C_T$*.*

The extension of this definition to incorporate a quantum memory device (QMD) is then:

Definition 8 (Quantum Processing Unit and Quantum Memory Device)**.** *We consider a model of computation comprising a* QPU *of size* $\operatorname{poly}\log(n)$ *and a Quantum Memory Device* (QMD) *of* n *memory registers, where each register is of* ℓ*-qubit size (for* n *a power of* 2*). A* QPU *and a* QMD *are collectively defined by a tuple* $(\mathtt{I}, \mathtt{W}, \mathtt{A}, \mathtt{T}, \mathtt{Aux}, \mathtt{M}, \mathcal{G}, \mathsf{V})$ *consisting of*

1. *two* $(\operatorname{poly}\log n)$*-qubit Hilbert spaces called* input register $\mathtt{I}$ *and* workspace $\mathtt{W}$ *owned solely by the* QPU*;*
2. *a* $(\log n)$*-qubit Hilbert space called* address register $\mathtt{A}$ *shared by both* QPU *and* QMD*;*
3. *an* ℓ*-qubit Hilbert space called* target register $\mathtt{T}$ *shared by both* QPU *and* QMD*;*
4. *a* $(\operatorname{poly} n)$*-qubit Hilbert space called* auxiliary register $\mathtt{Aux}$ *owned solely by the* QMD*;*
5. *an* $n\ell$*-qubit Hilbert space called* memory $\mathtt{M}$ *comprising* n *registers* $\mathtt{M}_0, \ldots, \mathtt{M}_{n-1}$*, each containing* ℓ *qubits, owned solely by the* QMD*;*
6. *a constant-size universal gate set* $\mathcal{G} \subset \mathcal{U}(\mathbb{C}^{4\times 4})$*;*
7. *a function* $\mathsf{V} : [n] \to \mathcal{V}$*, where* $\mathcal{V} \subset \mathcal{U}(\mathbb{C}^{2^{2\ell}\times 2^{2\ell}})$ *is a* $O(1)$*-size subset of* 2ℓ*-qubit gates.*

The qubits in $\mathtt{W}$*,* $\mathtt{A}$*,* $\mathtt{T}$*, and* $\mathtt{Aux}$ *are called ancillary qubits or simply ancillae. An input to the* QPU *with a* QMD*, or quantum circuit, is a tuple* $(T, |\psi_{\mathtt{I}}\rangle, |\psi_{\mathtt{M}}\rangle, C_1, \ldots, C_T)$ *where* $T \in \mathbb{N}$*,* $|\psi_{\mathtt{I}}\rangle \in \mathtt{I}$*,* $|\psi_{\mathtt{M}}\rangle \in \mathtt{M}$*, and, for each* $t \in \{1, \ldots, T\}$*,* $C_t \in \mathcal{I}(\mathcal{G}, \mathsf{V})$ *is an instruction from a set* $\mathcal{I}(\mathcal{G}, \mathsf{V})$ *of possible instructions. The instruction set* $\mathcal{I}(\mathcal{G}, \mathsf{V})$ *is the set* $\mathcal{I}(\mathcal{G})$ *from* Definition 7 *of instructions on* $\mathtt{I} \otimes \mathtt{W} \otimes \mathtt{A} \otimes \mathtt{T}$ *augmented with the call-to-the-*QMD *instruction that implements the unitary*

$$|i\rangle_{\mathtt{A}}|b\rangle_{\mathtt{T}}|x_i\rangle_{\mathtt{M}_i}|0\rangle_{\mathtt{Aux}}^{\otimes \operatorname{poly} n} \mapsto |i\rangle_{\mathtt{A}}\mathsf{V}(i)\big(|b\rangle_{\mathtt{T}}|x_i\rangle_{\mathtt{M}_i}\big)|0\rangle_{\mathtt{Aux}}^{\otimes \operatorname{poly} n}, \qquad \forall i \in [n], b, x_i \in \{0,1\}^{\ell}.$$

Starting from $|\psi_0\rangle|0\rangle_{\mathtt{Aux}}^{\otimes \operatorname{poly} n}$*, where* $|\psi_0\rangle := |\psi_{\mathtt{I}}\rangle|0\rangle_{\mathtt{W}}^{\otimes \operatorname{poly}\log n}|0\rangle_{\mathtt{A}}^{\otimes \log n}|0\rangle_{\mathtt{T}}^{\otimes \ell}|\psi_{\mathtt{M}}\rangle$*, at each time step* $t \in \{1, \ldots, T\}$ *we obtain the state* $|\psi_t\rangle|0\rangle_{\mathtt{Aux}}^{\otimes \operatorname{poly} n} = C_t(|\psi_{t-1}\rangle|0\rangle_{\mathtt{Aux}}^{\otimes \operatorname{poly} n})$*, where* $|\psi_t\rangle \in \mathtt{I}\otimes\mathtt{W}\otimes\mathtt{A}\otimes\mathtt{T}\otimes\mathtt{M}$*.*

We depict the architecture of a quantum processing unit with access to a quantum memory device in Figure 2. The address register $\mathtt{A}$ (shared by the QPU and QMD) is used to select a unitary from $\mathcal{V}$ and apply it to the target and memory registers $\mathtt{T}$ and $\mathtt{M}$ with the help of the auxiliary register $\mathtt{Aux}$. Even though a call to the QMD might require gates from a universal gate set, we stress that the underlying quantum circuit implementing such a call is *fixed*, i.e., does not change throughout the execution of a quantum algorithm by the QPU, or even between different quantum algorithms. This allows for highly specialized circuits for the QMD.

We did not include measurements in our models of computation, but these can easily be performed on the output state $|\psi_T\rangle$ if desired. We do not fix the position of qubits within our architecture and thus allow for long-range interactions, generally through multi-qubit entangling gates like Fan-Out and GT gates (see Section 4). Our model, nonetheless, could be augmented with algorithms for efficiently moving and addressing qubits within physically realistic devices. In this direction, we point the reader to Ref. [BBG+13].

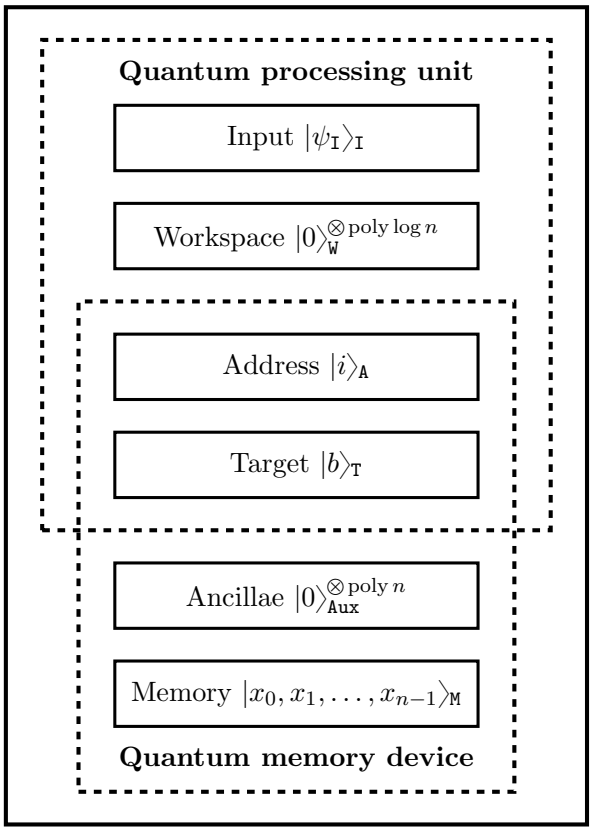


Figure 2: The architecture of a Quantum Processing Unit (QPU) with access to a quantum memory device (QMD). The QPU encompasses a $(\operatorname{poly}\log n)$-qubit input register $\mathtt{I}$ and workspace $\mathtt{W}$, a $(\log n)$-qubit address register $\mathtt{A}$, and an ℓ-qubit target register $\mathtt{T}$, while the QMD encompasses the address register $\mathtt{A}$, the target register $\mathtt{T}$, an $n\ell$-qubit memory array $\mathtt{M}$ composed of n cells $x_0, \ldots, x_{n-1} \in \{0,1\}^\ell$ of ℓ qubits each, and a $(\operatorname{poly} n)$-qubit auxiliary register $\mathtt{Aux}$.

Note that our definition of circuit size in Definition 7 differs slightly from the standard notion of circuit size (number of gates from $\mathcal{G}$) up to a factor of at most 2. In this work, we focus on constant-depth circuits, and since the size of a constant-depth circuit is just a constant times the number of input qubits plus ancillary qubits, we shall specify only the input and the number of ancillae of a circuit, with its size thus being implicit. In the rest of the paper we assume that each memory cell has size $\ell = 1$.

A call to the QMD is defined by the function V and we shall often equate the quantum memory device with the unitary that it implements. In many applications, one is interested in some form of reading a specific entry from the memory, which corresponds to the special cases where the $\mathsf{V}(i)$ unitaries are made of controlled single-qubit gates, and to which the traditional QRAM belongs.

Definition 9 (f-QRAM)**.** *Let $n \in \mathbb{N}$ be a power of 2 and $f : \{0,1\}^{\log n} \to \mathcal{U}(\mathbb{C}^{2\times 2})$. An* f-quantum random access memory (f-QRAM) *of memory size n is a* QMD *with $\mathsf{V}(i) = \mathsf{C}_{\mathtt{M}_i}\text{-}f(i)_{\to \mathtt{T}}$, $\forall i \in [n]$. Equivalently, it is a* QMD *that maps*

$$|i\rangle_{\mathtt{A}}|b\rangle_{\mathtt{T}}|x_0,\ldots,x_{n-1}\rangle_{\mathtt{M}} \mapsto |i\rangle_{\mathtt{A}}(f(i)^{x_i}|b\rangle_{\mathtt{T}})|x_0,\ldots,x_{n-1}\rangle_{\mathtt{M}} \qquad \forall i \in [n], b, x_0, \ldots, x_{n-1} \in \{0,1\}.$$

A special case, normally called simply QRAM*, is when $f(i) = \mathsf{X}$ for all $i \in [n]$, i.e., $\mathsf{V}(i) = \mathsf{C}_{\mathtt{M}_i}\text{-}\mathsf{X}_{\to \mathtt{T}}$.*

Note that f-QRAMs are QMDs that can be implemented via an UCG (see comment at the end of Section 3.1 and Figure 1). Another case of interest is writing content from the workspace into memory using SWAP gates.

Definition 10 (QRAG)**.** *Let $n \in \mathbb{N}$ be a power of 2. A* quantum random access gate QRAG *of memory size n is a* QMD *with $\mathsf{V}(i) = \mathsf{SWAP}_{\mathtt{M}_i \leftrightarrow \mathtt{T}}$, $\forall i \in [n]$. Equivalently, it is a* QMD *that maps*

$$|i\rangle_{\mathtt{A}}|b\rangle_{\mathtt{T}}|x_0,\dots,x_{n-1}\rangle_{\mathtt{M}} \mapsto |i\rangle_{\mathtt{A}}|x_i\rangle_{\mathtt{T}}|x_0,\dots,x_{i-1},b,x_{i+1},\dots,x_{n-1}\rangle_{\mathtt{M}} \quad \forall i \in [n], b, x_0,\dots,x_{n-1} \in \{0,1\}.$$

The following lemma shows that a QRAG is at least as powerful as a QRAM.

Lemma 11 (Simulating QRAM with QRAG)**.** *A query to a* QRAM *of memory size n can be simulated using 2 queries to a* QRAG *of memory size n, 3 two-qubit gates, and 1 workspace qubit.*

Proof. Start with the input $|i\rangle_{\mathtt{A}}|0\rangle_{\mathtt{Tmp}}|b\rangle_{\mathtt{T}}|x_0,\dots,x_{n-1}\rangle_{\mathtt{M}}$ by using an ancillary qubit Tmp from the workspace. Use a $\mathsf{SWAP}_{\mathtt{T}\leftrightarrow\mathtt{Tmp}}$ gate to obtain $|i\rangle_{\mathtt{A}}|b\rangle_{\mathtt{Tmp}}|0\rangle_{\mathtt{T}}|x_0,\dots,0,\dots,x_{n-1}\rangle_{\mathtt{M}}$. A query to the QRAG then leads to $|i\rangle_{\mathtt{A}}|b\rangle_{\mathtt{Tmp}}|x_i\rangle_{\mathtt{T}}|x_0,\dots,0,\dots,x_{n-1}\rangle_{\mathtt{M}}$. Use a $\mathsf{C}_{\mathtt{T}}\text{-}\mathsf{X}_{\to\mathtt{Tmp}}$ gate from register T to register Tmp, and query again the QRAG, followed by a $\mathsf{SWAP}_{\mathtt{T}\leftrightarrow\mathtt{Tmp}}$ gate, to obtain the desired state $|i\rangle_{\mathtt{A}}|b \oplus x_i\rangle_{\mathtt{T}}|x_0,\dots,x_{n-1}\rangle_{\mathtt{M}}$ after discarding the ancillary qubit. □

On the other hand, in our model, the converse is not true. It is possible, though, to simulate a QRAG using a constant number of QRAM queries in a model where single-qubit gates are allowed to be freely applied to the memory register M. The next lemma formalizes these results.

Lemma 12 (Simulating QRAG with QRAM)**.**

- *In the model from Definition 8, a query to a* QRAG *cannot be simulated by any number of queries to a* QRAM*.*
- *Suppose that single-qubit gates can be freely applied onto the memory register* M *of any* QRAM*. Then a* QRAG *of memory size n can be simulated using 3 queries to a* QRAM *of memory size n and $2(n+1)$ Hadamard gates.*

Proof. For the first statement, consider the simplest case of trying to implement a QRAG with zero address qubits (i.e., there is only one memory cell): we are given memory qubit M, target qubit T, and an arbitrary number of workspace qubits W. A single action of the QRAM followed by an arbitrary unitary U acting on the target and workspace maps $|x_0\rangle_{\mathtt{M}}|b\rangle_{\mathtt{T}}|\psi\rangle_{\mathtt{W}} \mapsto |x_0\rangle_{\mathtt{M}}\mathsf{U}(|b\oplus x_0\rangle_{\mathtt{T}}|\psi\rangle_{\mathtt{W}}) = |x_0\rangle_{\mathtt{M}}|\Phi\rangle_{\mathtt{TW}}$ and thus leaves the memory register invariant. As we cannot modify the memory register, it is not possible to swap the state of the memory with the contents of the target.

The second statement follows from the simple fact that three CNOTs can implement a SWAP, i.e., $\mathsf{SWAP}_{\mathtt{B}\leftrightarrow\mathtt{D}} = \mathsf{C}_{\mathtt{B}}\text{-}\mathsf{X}_{\to\mathtt{D}} \cdot \mathsf{C}_{\mathtt{D}}\text{-}\mathsf{X}_{\to\mathtt{B}} \cdot \mathsf{C}_{\mathtt{B}}\text{-}\mathsf{X}_{\to\mathtt{D}}$, and that one can swap control and target registers of a CNOT as $(\mathsf{H}_{\to\mathtt{B}}\cdot\mathsf{H}_{\to\mathtt{D}})\mathsf{C}_{\mathtt{B}}\text{-}\mathsf{X}_{\to\mathtt{D}}(\mathsf{H}_{\to\mathtt{B}}\cdot\mathsf{H}_{\to\mathtt{D}}) = \mathsf{C}_{\mathtt{D}}\text{-}\mathsf{X}_{\to\mathtt{B}}$, for registers B, D. Then, starting from the input $|i\rangle_{\mathtt{A}}|b\rangle_{\mathtt{T}}|x_0,\dots,x_{n-1}\rangle_{\mathtt{M}}$, apply a QRAM followed by the $n+1$ Hadamard gates $\mathsf{H}_{\to\mathtt{T}} \cdot \prod_{j\in[n]} \mathsf{H}_{\to\mathtt{M}_j}$, and then another QRAM query followed by $\mathsf{H}_{\to\mathtt{T}} \cdot \prod_{j\in[n]} \mathsf{H}_{\to\mathtt{M}_j}$, and a final QRAM query. □

Our computational model can be seen as a refined version of the one described in [BLPS22b]. Similar to our Definition 8, the authors divide the qubits of a quantum computer into work and memory qubits. Given M memory qubits, their workspace consists of $O(\log M)$ qubits, of which the address and target qubits are always the first $\lceil \log M \rceil + 1$ qubits. However, address and target qubits are not considered to be shared by the QMD, and there is no mention of ancillary qubits mediating a call to the QMD. The inner structure of the QMD is abstracted away by assuming

access to the unitary of a QRAG as in Definition 10. Our model, in contrast, "opens" the quantum memory device, and allows for general fixed unitaries, including QRAM and QRAG.

The first efficient architectures for QRAM were formalized and proposed in [GLM08a, GLM08b], namely the Fan-Out and bucket-brigade architectures. These architectures can readily be used for QRAGs, with a simple modification: replacing the last layer of CNOT gates with SWAP gates. Both schemes access the memory cells through a binary tree of size $O(n)$ and depth $\log n$. Each qubit of the address register $|i\rangle_{\mathtt{A}}$ specifies the direction to follow from the root to the correct memory cell, i.e., the k-th qubit of the address register tells whether to go left or right at a router (or bifurcation) on the k-th level of the binary tree. The target qubit is sent down the binary tree to the memory cell corresponding to the address register, and the information in the memory cell is copied (QRAM) or swapped (QRAG), and the target qubit is then sent back up the tree to the root.

The Fan-Out and bucket-brigade architectures differ in how the target qubit is routed down the binary tree. In the Fan-Out architecture, the k-th address qubit controls all the 2^k routers on the k-th level via a Fan-Out gate. The drawback of this scheme is that it requires simultaneous control of all $n-1$ routers, even though only $\log n$ routers (in each branch of the wavefunction) are necessary to route the target down the tree. This in turn makes the Fan-Out architecture highly susceptible to noise since each router is maximally entangled with the rest of the system. In the bucket-brigade architecture, on the other hand, all routers are initially in an "idle" state. Each address qubit is sequentially sent down the binary tree and its state is transferred to the first idle router it encounters. This creates a path for the following address qubits to the next idle router and, after all address qubits have been routed down the tree, a path for the target qubits to the correct memory cells. One main advantage of the bucket-brigade architecture is reducing the number of active routers down to $\log n$ in each component of the superposition. Another advantage is its high resilience to noise due to limited entanglement between the memory components [GLM08a, GLM08b, AGJO$^+$15, HLGJ21].

Several other architectures for QRAM have been proposed, including Flip-Flop QRAM [PPR19], Entangling Quantum Generative Adversarial Network QRAM [NZB$^+$22], approximate Parametric-Quantum-Circuit-based QRAM [PLG22], and others [CDEH$^+$21, ZLW19, NZB$^+$22, AP22]. Roughly speaking, one can classify the proposals for QRAM with classical memory in two ways [MGM20]. One way is to explicitly lay out the classical memory in physical hardware at the end of the quantum circuit implementing a QRAM, e.g., at the end of the ancillary binary tree in the Fan-Out and bucket-brigade architectures, and then be copied via a CNOT gate. The advantage of such "explicit" QRAMs is that their underlying circuits must be optimized and compiled just once, while the contents of the memory array can be modified freely. The other way is to encode the memory implicitly in the quantum circuit. This can be achieved by employing multicontrolled CNOT gates controlled by bits representing the memory address containing a 1. The advantage of such "implicit" QRAMs is that in some cases they can be heavily optimized using techniques from Boolean circuits [MP01, SSP13]. Another way to distinguish between QMDs is in the way the routing operation, i.e., the memory cell selection, is implemented: passively or actively. For example, the architecture in [CDEH$^+$21] is passive: when the routers (the ancillary qubits of the device) are configured, a photon gets absorbed into a cavity, and then subsequent incoming photons acquire a phase shift depending on the state of the cavity. Active architectures [HLGJ21], on the other hand, are similar to a traditional gate-based quantum computer, where each SWAP or controlled-SWAP gate is executed by some control pulse. We point the reader to a few recent surveys on the state of the art of QRAMs for more information [Han21, PCG23, JR23].

3.1 Uniformly controlled gates

An f-Uniformly Controlled Gate (f-UCG or simply UCG) is a unitary that, conditioned on the state of a set of control qubits, implements one of a set of single-qubit gates on a target qubit.

Definition 13 (f-Uniformly Controlled Gate)**.** *Let $m, n \in \mathbb{N}$, $n < m$. Let $i \in [m]$. Consider a function $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$, and let $S \in ([m] \setminus \{i\})^n$ be a sequence of n* non-repeating *elements from $[m] \setminus \{i\}$. The Uniformly Controlled Gate $f\text{-}\mathsf{UCG}^{(n)}_{S\to i}$ of arity[3] n is defined as*

$$|x_0\rangle|x_1\rangle \dots |x_{m-1}\rangle \mapsto |x_0\rangle \dots |x_{i-1}\rangle(f(x_S)|x_i\rangle)|x_{i+1}\rangle \dots |x_{m-1}\rangle, \quad \forall x_0, \dots, x_{m-1} \in \{0,1\},$$

where $x_S = x_{S_1} \dots x_{S_n}$. When it is clear from context, we shall omit either the superscript (n), or the subscripts corresponding to the target i and/or control S from $f\text{-}\mathsf{UCG}^{(n)}_{S\to i}$. By f-UCG we mean a generic $f\text{-}\mathsf{UCG}^{(n)}_{S\to i}$ for some n, S, i.

An f-UCG is normally defined in the literature by listing a set $\{\mathsf{U}_0, \dots, \mathsf{U}_{2^n-1}\}$ of single-qubit gates (corresponding to $f(0^n), \dots, f(1^n)$), and writing

$$f\text{-}\mathsf{UCG}^{(n)}_{[n]\to n} = \sum_{x\in\{0,1\}^n} |x\rangle\langle x| \otimes f(x) = \sum_{x\in\{0,1\}^n} |x\rangle\langle x| \otimes \mathsf{U}_x,$$

where we ignored the qubits on which $f\text{-}\mathsf{UCG}^{(n)}_{S\to i}$ does not depend (so $m = n+1$) and took the target qubit i to be the last one. Equivalently, its matrix representation is

$$f\text{-}\mathsf{UCG}^{(n)}_{[n]\to n} = \begin{pmatrix} \mathsf{U}_0 & & & \\ & \mathsf{U}_1 & & \\ & & \ddots & \\ & & & \mathsf{U}_{2^n-1} \end{pmatrix} \in \mathbb{C}^{2^{(n+1)}\times 2^{(n+1)}}.$$

A possible way to implement $f\text{-}\mathsf{UCG}^{(n)}_{[n]\to n}$ is shown in Figure 3a, where each gate $f(x) = \mathsf{U}_x$ is sequentially performed controlled on the state $|x\rangle$. It is possible to simplify such sequential implementation by using Gray codes, see [BBC+95, BM04, STY+23].

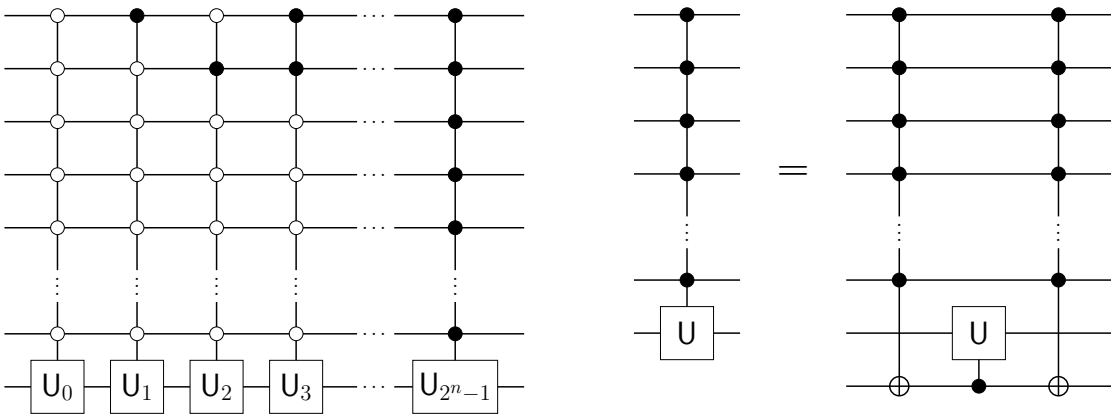
Well-known examples of f-UCGs can be found in [GR02, KP17, Mon15, HHL09]. These algorithms perform a set of controlled rotations $|x\rangle|0\rangle \mapsto |x\rangle(\cos\theta(x)|0\rangle + \sin\theta(x)|1\rangle)$ on a single qubit for a function $\theta : \{0,1\}^n \to [0, 2\pi]$. Another example is the special subclass of f-UCGs known as Fan-In gates (f-FIN), for which $f : \{0,1\}^n \to \{\mathbb{I}_1, \mathsf{X}\}$, i.e., the Z-decomposition of f is simply $f(x) = \mathsf{HZ}(\gamma(x))\mathsf{H} = \mathsf{X}^{\gamma(x)}$ for $\gamma : \{0,1\}^n \to \{0,1\}$. Fan-In gates are thus equivalent to gates for which a Boolean function is computed on a subset of the registers and the result is added to a specified register $|x_i\rangle$. Other f-UCGs include phase oracles for which $f : \{0,1\}^n \to \{\mathbb{I}_1, \mathsf{Z}\}$.

Definition 14 (f-Fan-In gate)**.** *Let $m, n \in \mathbb{N}$, $n < m$. Let $i \in [m]$. Consider a Boolean function $f : \{0,1\}^n \to \{0,1\}$ on n bits, and let $S \in ([m] \setminus \{i\})^n$ be a sequence of n* non-repeating *elements from $[m] \setminus \{i\}$. The Fan-In gate $f\text{-}\mathsf{FIN}^{(n)}_{S\to i}$ of arity n is defined as*

$$|x_0\rangle|x_1\rangle \dots |x_{m-1}\rangle \mapsto |x_0\rangle \dots |x_{i-1}\rangle|x_i \oplus f(x_S)\rangle|x_{i+1}\rangle \dots |x_{m-1}\rangle, \quad \forall x_0, \dots, x_{m-1} \in \{0,1\},$$

where $x_S = x_{S_1} \dots x_{S_n}$. When it is clear from context, we shall omit either the superscript (n), or the subscripts corresponding to the target i and/or control S from $f\text{-}\mathsf{FIN}^{(n)}_{S\to i}$. By f-FIN we mean a generic $f\text{-}\mathsf{FIN}^{(n)}_{S\to i}$ for some n, S, i.

[3] Even though the gate depends on $n+1$ qubits, we define its arity according to f.



(a) (b)

Figure 3: (a) A sequential implementation of f-UCG$^{(n)}_{[n]\to n}$. (b) The gate $|x\rangle\langle x| \otimes \mathsf{U} + \sum_{j\in\{0,1\}^n\setminus\{x\}} |j\rangle\langle j| \otimes \mathbb{I}_1$ can be implemented by employing two AND gates onto an ancillary qubit and controlling $\mathsf{U} \in \mathcal{U}(\mathbb{C}^{2\times 2})$ on it being in the $|1\rangle$ state [BBC+95].

Examples of Boolean functions $f : \{0,1\}^n \to \{0,1\}$ include

$$f(x) = 1 \text{ if and only if } \begin{cases} |x| > 0 & \mathsf{OR}^{(n)}, \\ |x| = n & \mathsf{AND}^{(n)} \text{ (generalized Toffoli)}, \\ |x| \geq n/2 & \mathsf{MAJORITY}^{(n)}, \\ |x| \geq t & \mathsf{THRESHOLD}^{(n)}[t], \\ |x| = t & \mathsf{EXACT}^{(n)}[t], \\ |x| \pmod q = 0 & \mathsf{MOD}^{(n)}[q], \\ |x| \text{ is odd} & \mathsf{PARITY}^{(n)}. \end{cases}$$

Another example of f-FIN is the QRAM itself. Indeed, QRAM is simply the f-FIN with $f : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}$ defined by $f(x,i) = x_i$ (also known as selection function).

The following simple fact is behind our constructions based on one-hot encoding in Section 5.

Fact 15. *Given $x \in \{0,1\}^n$ and $\mathsf{U} \in \mathcal{U}(\mathbb{C}^{2\times 2})$, the gate $|x\rangle\langle x| \otimes \mathsf{U} + \sum_{j\in\{0,1\}^n\setminus\{x\}} |j\rangle\langle j| \otimes \mathbb{I}_1$ can be implemented using two $\mathsf{AND}^{(n)}$ gates and one ancillary qubit.*

Proof. Given n-qubit register $|k\rangle_{\mathtt{I}} = \bigotimes_{j\in[n]} |k_j\rangle_{\mathtt{I}_j}$ and single-qubit register $|b\rangle_{\mathtt{T}}$, simply note that

$$\left(|x\rangle\langle x| \otimes \mathsf{U} + \sum_{j\in\{0,1\}^n\setminus\{x\}} |j\rangle\langle j| \otimes \mathbb{I}_1\right) \otimes \mathbb{I}_1 |k\rangle_{\mathtt{I}} |b\rangle_{\mathtt{T}} |0\rangle_{\mathtt{Tmp}}$$
$$= \left(\prod_{j\in[n]} \mathsf{X}^{\overline{x}_j}_{\to \mathtt{I}_j}\right) \mathsf{AND}^{(n)}_{\mathtt{I}\to\mathtt{Tmp}} \cdot \mathsf{C}_{\mathtt{Tmp}}\text{-}\mathsf{U}_{\to\mathtt{T}} \cdot \mathsf{AND}^{(n)}_{\mathtt{I}\to\mathtt{Tmp}} \left(\prod_{j\in[n]} \mathsf{X}^{\overline{x}_j}_{\to \mathtt{I}_j}\right) |k\rangle_{\mathtt{I}} |b\rangle_{\mathtt{T}} |0\rangle_{\mathtt{Tmp}}$$

for all $k \in \{0,1\}^n$ and $b \in \{0,1\}$ (see Figure 3b). □

Relation between QMD and f-UCG. Uniformly Controlled Gates and quantum memory devices are similar but distinct concepts. Since $\mathcal{V} \subset \mathcal{U}(\mathbb{C}^{4\times 4})$, i.e., $\mathsf{V}(i)$ can act non-trivially on two qubits

for all $i \in \{0,1\}^{\log n}$ (registers $\mathtt{T}$ and $\mathtt{M}_i$), it is clear that f-UCGs cannot simulate general QMDs. However, if, for all $i \in \{0,1\}^{\log n}$, $\mathsf{V}(i)$ is of the form $f(i) \otimes \mathbb{I}_1$ for some $f : \{0,1\}^{\log n} \to \mathcal{U}(\mathbb{C}^{2\times 2})$, then such QMD is simply the f-UCG$^{(\log n)}$. Similarly, an f-QRAM for $f : \{0,1\}^{\log n} \to \mathcal{U}(\mathbb{C}^{2\times 2})$ (which is a QMD such that $\mathsf{V}(i) = \mathbb{I}_1 \otimes |0\rangle\langle 0|_{\mathtt{M}_i} + f(i) \otimes |1\rangle\langle 1|_{\mathtt{M}_i}$) is an f'-UCG$^{(n+\log n)}$ for some $f' : \{0,1\}^n \times \{0,1\}^{\log n} \to \mathcal{U}(\mathbb{C}^{2\times 2})$ that is a $(J = [n], 1)$-junta.

In the other direction, the requirement that $\mathcal{V}$ be a constant-size set limits the kind of f-UCG that can be simulated by a QMD to those where f has a constant range. Relaxing the restriction on the size of $\mathcal{V}$, an f-UCG$^{(1+\log n)}$ can be simulated by a QMD of memory size n.

4 Multi-qubit gates as building blocks

4.1 The Fan-Out gate

The Fan-Out gate copies a specific register $|x_i\rangle$ into a subset of other registers. It can be thought of as a single-control multiple-target CNOT gate.

Definition 16 (Fan-Out gate)**.** *Let $m \in \mathbb{N}$. Let $i \in [m]$ and $S \subseteq [m] \setminus \{i\}$, with $|S \cup \{i\}| =: n$. The Fan-Out gate $\mathsf{FO}^{(n)}_{i\to S}$ of arity n is defined as*

$$|x_0\rangle|x_1\rangle \dots |x_{m-1}\rangle \mapsto \bigotimes_{j\in[m]} \begin{cases} |x_j \oplus x_i\rangle & \text{if } j \in S, \\ |x_j\rangle & \text{if } j \notin S, \end{cases} \qquad \forall x_0, \dots, x_{m-1} \in \{0,1\},$$

which copies the bit x_i into the registers in S. Similarly to f-UCG, we shall sometimes omit either the superscript (n), or the subscripts corresponding to the control i and/or target S from $\mathsf{FO}^{(n)}_{i\to S}$.

The Fan-Out gate is known to be powerful, in that other multi-qubit gates can be efficiently implemented if one has access to Fan-Out. In particular, we have the following fact.

Fact 17 ([Moo99, GHMP02])**.** *The Fan-Out gate is equivalent to the* PARITY *gate up to a Hadamard conjugation, i.e., for $i \in [n]$ and $S \subseteq [n] \setminus \{i\}$,*

$$\mathsf{PARITY}_{S\to i} = \left(\prod_{j\in S\cup\{i\}} \mathsf{H}_{\to j}\right) \mathsf{FO}_{i\to S} \left(\prod_{j\in S\cup\{i\}} \mathsf{H}_{\to j}\right).$$

It is known that the EXACT$^{(n)}$ gate (including OR$^{(n)}$ and AND$^{(n)}$) can be simulated exactly in constant depth using Fan-Out and single-qubit gates [TT16]. Other known constructions with Fan-Out include MAJORITY and THRESHOLD [HŠ05, TT16].[4]

Fact 18 ([TT16, Theorem 1])**.** *The* EXACT$^{(n)}[t]$ *gate can be implemented in $O(1)$-depth using $2n\log n + O(n)$ ancillae and $6n + O(\log n)$ Fan-Out gates with arity $\leq 2n$.*

The above result comes from a useful OR reduction from n to $\lceil \log(n+1) \rceil$ qubits developed in [HŠ05]. We include the proof for completeness, and explicitly count the resources required.

Fact 19 ([HŠ05, Lemma 5.1])**.** *The* OR$^{(n)}$ *gate can be reduced to* OR$^{(p)}$*, $p = \lceil \log(n+1) \rceil$, in $O(1)$-depth using $2n\lceil \log(n+1) \rceil$ ancillae and $2n + 2\lceil \log(n+1) \rceil$ Fan-Out gates with arity at most n. In other words, there is a $O(1)$-depth circuit that maps $|x\rangle|0\rangle^{\otimes p} \mapsto |x\rangle|\psi_x\rangle$ for $x \in \{0,1\}^n$, where $|\psi_x\rangle \in \mathbb{C}^{2^p}$ is such that $\langle 0^p|\psi_x\rangle = 1$ if $\mathsf{OR}(x) = 0$ and $\langle 0^p|\psi_x\rangle = 0$ if $\mathsf{OR}(x) = 1$.*

[4] The power of quantum THRESHOLD has recently been explored in [GM24] one year after our work appeared online and shortly before publication.

Proof. Given the input $|x\rangle|0\rangle$, $x \in \{0,1\}^n$, we first show how to compute $|x\rangle|0\rangle \mapsto |x\rangle|\mu_\theta^{|x|}\rangle$ in constant depth, where $|\mu_\theta^{|x|}\rangle := \frac{1}{2}(1+e^{i\pi\theta|x|})|0\rangle + \frac{1}{2}(1-e^{i\pi\theta|x|})|1\rangle$, $\theta \in [-1,1]$. Attach an ancillary register $|0\rangle^{\otimes(n-1)}$ and apply a Hadamard gate on the first qubit of $|0\rangle^{\otimes n}$ followed by a Fan-Out gate copying this first qubit onto the remaining $n-1$ qubits. This leads to

$$|x\rangle|0\rangle^{\otimes n} \mapsto |x\rangle\frac{|0\rangle+|1\rangle}{\sqrt{2}}|0\rangle^{\otimes(n-1)} \mapsto |x\rangle\frac{|0\rangle^{\otimes n}+|1\rangle^{\otimes n}}{\sqrt{2}}.$$

Apply a $\mathsf{Z}(\theta x_i)$ gate on the i-th qubit of $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes n}+|1\rangle^{\otimes n})$ controlled on $|x_i\rangle$, for $i \in [n]$. Thus

$$|x\rangle\frac{|0\rangle^{\otimes n}+|1\rangle^{\otimes n}}{\sqrt{2}} \mapsto |x\rangle\frac{|0\rangle^{\otimes n}+e^{i\pi\theta|x|}|1\rangle^{\otimes n}}{\sqrt{2}}.$$

Uncomputing the first step leads to $|x\rangle|\mu_\theta^{|x|}\rangle$ as required. In total, we have used $n-1$ ancillae and 2 Fan-Out gates with arity n.

The reduction works by computing in parallel the states $|\psi_k\rangle = |\mu_{\theta_k}^{|x|}\rangle$ with $\theta_k = 1/2^k$, for all $k \in [p]$, which requires copying the register $|x\rangle$ a number of $p-1$ times by using n Fan-Out gates with arity p. The output $|\psi\rangle = |\psi_0\rangle|\psi_1\rangle\ldots|\psi_{p-1}\rangle$ is the desired state. Indeed, if $|x|=0$, then $\langle 0^p|\psi\rangle = 1$, since $|\psi_k\rangle = |0\rangle$ for each $k \in [p]$. On the other hand, if $|x| \neq 0$, then $\langle 0^p|\psi\rangle = 0$, since at least one qubit $|\psi_k\rangle$ is $|1\rangle$ with certainty. Indeed, there are integers $a \in [p]$ and $b \geq 0$ such that $|x| = 2^a(2b+1)$. Then a direct calculation shows that $\langle 1|\psi_a\rangle = 1$. This proves the correctness of the reduction. Finally, the whole reduction uses $n(p-1)+p(n-1) \leq 2np$ ancillae and $2m+2p$ Fan-Out gates with arity at most n. □

Pham and Svore [PS13] showed how to implement an n-arity Fan-Out using a $O(1)$-depth and $O(n)$-size quantum circuit with intermediary measurements, classical feedback, and classical $O(\log n)$-depth computation. Without intermediary measurements and classical feedback, it is folklore that n-arity Fan-Out gates can be implemented by a CNOT circuit of depth $O(\log n)$ and size $O(n)$. If low-arity Fan-Out gates are available, it is possible to improve the number and depth of CNOT gates required as follows.

Lemma 20. *For $y \in \{0,1\}$, the unitary $|y\rangle|0\rangle^{\otimes n} \mapsto |y\rangle^{\otimes(n+1)}$ can be implemented with $\lceil n/(k-1)\rceil$ k-arity Fan-Out gates in depth $\lceil \log_k(n+1)\rceil$.*

Proof. Note that, starting from the initial state at depth $d=0$ until the final state at maximum depth $d=d_c$, the i-th Fan-Out layer maps the i-th state onto the $(i+1)$-th state, for $i \in [d_c]$. We prove by induction that, at depth d, our state is $|y\rangle^{\otimes k^d}|0\rangle^{\otimes(n+1-k^d)}$. The case $d=0$ is obvious. Assume the induction hypothesis for d. Then, after applying one layer of k^d k-arity Fan-Out gates we obtain

$$|y\rangle^{\otimes k^d}|0\rangle^{\otimes(n+1-k^d)} \mapsto |y\rangle^{\otimes k^{d+1}}|0\rangle^{\otimes(n+1-k^{d+1})},$$

as wanted. The circuit depth d_c is the minimum d such that $n+1-k^d \leq 0$, i.e., $d_c = \lceil \log_k(n+1)\rceil$. Regarding the size, from depth $d=0$ to $d=d_c-1$ we require $\sum_{j=0}^{d_c-2} k^j = (k^{d_c-1}-1)/(k-1)$ Fan-Out gates. In the final layer there are only $n+1-k^{d_c-1}$ qubits left in the $|0\rangle$ state, thus another $\lceil \frac{n+1}{k} - k^{d_c-2}\rceil$ Fan-Out gates are required. In total, the number of Fan-Outs is at most

$$\frac{k^{d_c-1}-1}{k-1} + \left\lceil \frac{n+1}{k} - k^{d_c-2}\right\rceil = \left\lceil \frac{n+1}{k} + \frac{k^{d_c-2}-1}{k-1}\right\rceil \leq \left\lceil \frac{n+1}{k} + \frac{(n+1)/k-1}{k-1}\right\rceil = \left\lceil \frac{n}{k-1}\right\rceil. \quad \square$$

4.2 The Global Tunable gate

Definition 21 (Global Tunable gate)**.** *Let* $\Theta \in [-1,1]^{n\times n}$*. The* n*-arity Global Tunable gate* $\mathsf{GT}_\Theta^{(n)}$ *is the unitary operator*

$$\mathsf{GT}_\Theta^{(n)} = \prod_{1\le i<j\le n} \mathsf{C}_i\text{-}\mathsf{Z}(\Theta_{ij})_{\to j}.$$

The GT gate is powerful in that it can perform many Fan-Out gates in parallel.

Lemma 22. *A number* l *of pair-wise commuting Fan-Out gates* $\mathsf{FO}^{(n_0)},\dots,\mathsf{FO}^{(n_{l-1})}$ *can be performed in depth-3 using one* GT *gate with arity at most* n *and at most* $2(n-1)$ *Hadamard gates, where* $n := \sum_{j\in[l]} n_j$*.*

Proof. Let $T \subseteq [l]$. Without lost of generality, for each $i \in [l]$, consider a Fan-Out gate $\mathsf{FO}_{q_i\to S_i}$ controlled on qubit $q_i \in T$ with target qubits in $S_i \subseteq [n]\setminus T$. All Fan-Out gates $\mathsf{FO}_{q_i\to S_i}$ commute since the sets of target and control qubits T and $\bigcup_{i\in[l]} S_i$, respectively, are disjoint. Therefore

$$\prod_{i\in[l]} \mathsf{FO}_{q_i\to S_i} = \prod_{i\in[l]}\prod_{j\in S_i} \mathsf{C}_{q_i}\text{-}\mathsf{X}_{\to j} = \prod_{i\in[l]}\prod_{j\in S_i} \mathsf{H}_{\to j}\cdot \mathsf{C}_{q_i}\text{-}\mathsf{Z}_{\to j}\cdot \mathsf{H}_{\to j} = \left(\prod_{j\in\bigcup_{i\in[l]} S_i} \mathsf{H}_{\to j}\right) \mathsf{GT}_\Theta \left(\prod_{j\in\bigcup_{i\in[l]} S_i} \mathsf{H}_{\to j}\right).$$

Here $\Theta \in \{0,1\}^{n\times n}$ is the matrix $\Theta = \bigoplus_{i\in[l]} \Theta_{q_i}$, where $\Theta_{q_i} \in \{0,1\}^{n\times n}$ is the matrix whose q_i-th row is the characteristic vector of the set S_i, while the remaining rows are zero (the parity is taken entry-wise). In other words, the (i,j)-entry of Θ is the parity of the number of sets $S_0,\dots,S_{l-1}$ that contain $j \in [n]\setminus T$ and are controlled on qubit $i \in T$. This is because a Z gate is applied onto a qubit $j \in [n]\setminus T$ only if it is the target to an odd number of Fan-Out gates. The maximum arity of the GT gate happens when $(S_i \cup \{q_i\}) \cap (S_j \cup \{q_j\}) = \emptyset$ for every $i \neq j \in [l]$, i.e., when each Fan-Out gate acts on a separate set of qubits. The maximum number of Hadamard gates happens when $q_0 = \cdots = q_{l-1}$ and $S_i \cap S_j = \emptyset$ for every $i \neq j \in [l]$, i.e., when all Fan-Out gates share the same control qubit but copy it into separate sets of qubits. □

Thus, up to conjugation by Hadamards, a single GT gate can copy a control register into an arbitrary number of target registers. Moreover, from the above fact follows a simple yet interesting result concerning constant-depth quantum circuits.

Lemma 23. *Consider a constant-depth circuit that uses* l *Fan-Out gates* $\mathsf{FO}^{(n_0)},\dots,\mathsf{FO}^{(n_{l-1})}$*. There is an equivalent constant-depth circuit that uses* $O(1)$ $\mathsf{GT}^{(n)}$ *gates, where* $n \le \sum_{j\in[l]} n_j$*.*

Proof. The circuit consists of a constant number of layers, each of which contains at most l disjoint Fan-Out gates. The result follows from Lemma 22. □

An example of the above result applied to Fact 18 (up to an exact gate count) is the following result concerning the $\mathsf{EXACT}^{(n)}$ gate.

Fact 24 ([BMN22])**.** *The* $\mathsf{EXACT}^{(n)}[t]$ *gate can be implemented in* $O(1)$*-depth using* $2n + O(\log n)$ *ancillae and* 4 GT *gates with arity* $\le n + O(\log n)$*.*

Note that the original construction of [BMN22] is for $\mathsf{OR}^{(n)}$ and requires fewer than $2n$ ancillae because they implement a slightly different gate, $|x\rangle \mapsto (-1)^{\mathsf{OR}(x)}|x\rangle$. Their construction is similar to the one from [TT16] and uses the OR reduction from [HŠ05] adapted to GT gates. We include the proof of the OR reduction for completeness and explicitly count the resources required.

Fact 25 ([BMN22])**.** *The* $\mathsf{OR}^{(n)}$ *gate can be reduced to* $\mathsf{OR}^{(p)}$*,* $p = \lceil \log(n+1) \rceil$*, in* $O(1)$*-depth using* 1 GT *gate with arity* $n + \lceil \log(n+1) \rceil$ *and no ancillae.*

Proof. The general construction is the same as in Fact 19, the difference being the number of ancillary qubits. When constructing the states $|\mu_\theta^{|x|}\rangle$, there is no need for the ancillary register $|0\rangle^{\otimes(n-1)}$, since all the $\mathsf{Z}(\theta x_i)$ gates controlled on $|x_i\rangle$, $i \in [n]$, can be performed using a single GT gate with arity $n+1$ on the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, i.e., the mapping

$$|x\rangle \frac{|0\rangle + |1\rangle}{\sqrt{2}} \mapsto |x\rangle \frac{|0\rangle + e^{i\pi\theta|x|}|1\rangle}{\sqrt{2}}$$

requires only one GT gate and no ancillae. This procedure actually scales to computing $|x\rangle|0\rangle^{\otimes p} \mapsto |x\rangle|\psi_0\rangle \dots |\psi_{p-1}\rangle$, i.e., all states $|\psi_k\rangle = |\mu_{\theta_k}^{|x|}\rangle$ can be computed in parallel with a single GT gate with arity $n+p$ and no ancillary qubits. □

As another example, it was shown in [Ros21b, Theorem 6] that every n-qubit state can be constructed by a QAC_f^0 circuit with $\widetilde{O}(2^n)$ ancillae. It follows from Lemma 23 that every n-qubit state can be constructed with a constant number of GT gates and $\widetilde{O}(2^n)$ ancillae.

Comment on physical implementation of multi-qubit gates. The constant-depth architectures we consider make use of the multi-qubit Fan-Out and GT gates. However, the complexity and time required to implement such gates in practice may differ and may be both hardware and code-dependent. For example, if one considers logical qubits encoded via the surface code, then for a fixed code distance d, Fan-Out gates can be performed in a constant number of surface code cycles via braiding [FMMC12]. On the other hand, in the non-error-corrected ion-trap GT gate implementation proposed in [GBW+20], each of the n qubits is simultaneously acted on by a separate sequence of at least $2n$ constant-duration laser pulses. Assuming a practical lower bound on the duration of any pulse in this sequence, the wall-clock time required to implement a single GT gate according to this scheme scales linearly with n (and uses a linear number of laser sources) and the constant-depth GT gate constructions do not necessarily translate to constant-time constructions. This is not surprising, since the GT gate is strictly more powerful than Fan-Out.

5 Constant-depth circuits based on one-hot encoding

In this section, we provide constant-depth circuits for f-UCGs via our first technique based one-hot encoding. We rely on a simple fact regarding the unitary

$$\mathsf{C}_{\mathtt{E}_{n-1}}\text{-}(\mathsf{U}_{n-1})_{\to\mathtt{T}} \cdots \mathsf{C}_{\mathtt{E}_1}\text{-}(\mathsf{U}_1)_{\to\mathtt{T}} \cdot \mathsf{C}_{\mathtt{E}_0}\text{-}(\mathsf{U}_0)_{\to\mathtt{T}}$$

that sequentially applies the gates $\mathsf{U}_0, \dots, \mathsf{U}_{n-1}$ onto a target qubit $\mathtt{T}$ controlled on the single-qubit registers $\mathtt{E}_0, \dots, \mathtt{E}_{n-1}$, respectively, being in the $|1\rangle$ state (see Figure 4a). Let $|e\rangle_{\mathtt{E}} = \bigotimes_{j\in[n]} |e_j\rangle_{\mathtt{E}_j}$ be the state of the registers $\mathtt{E}_0, \dots, \mathtt{E}_{n-1}$. If $e \in \{0,1\}^n$ has Hamming weight at most 1 (i.e., $|e| \leq 1$), then we can rearrange the gates from the Z-decomposition of U_j as

$$\mathsf{C}_{\mathtt{E}_{n-1}}\text{-}(\mathsf{U}_{n-1})_{\to\mathtt{T}} \cdots \mathsf{C}_{\mathtt{E}_1}\text{-}(\mathsf{U}_1)_{\to\mathtt{T}} \cdot \mathsf{C}_{\mathtt{E}_0}\text{-}(\mathsf{U}_0)_{\to\mathtt{T}} |e\rangle_{\mathtt{E}} |b\rangle_{\mathtt{T}} =$$
$$\left(\prod_{j\in[n]} \mathsf{Z}(\alpha_j)_{\to\mathtt{E}_j}\right) \left(\prod_{j\in[n]} \mathsf{C}_{\mathtt{E}_j}\text{-}\mathsf{Z}(\beta_j)_{\to\mathtt{T}}\right) \mathsf{H}_{\to\mathtt{T}} \left(\prod_{j\in[n]} \mathsf{C}_{\mathtt{E}_j}\text{-}\mathsf{Z}(\gamma_j)_{\to\mathtt{T}}\right) \mathsf{H}_{\to\mathtt{T}} \left(\prod_{j\in[n]} \mathsf{C}_{\mathtt{E}_j}\text{-}\mathsf{Z}(\delta_j)_{\to\mathtt{T}}\right) |e\rangle_{\mathtt{E}} |b\rangle_{\mathtt{T}}.$$

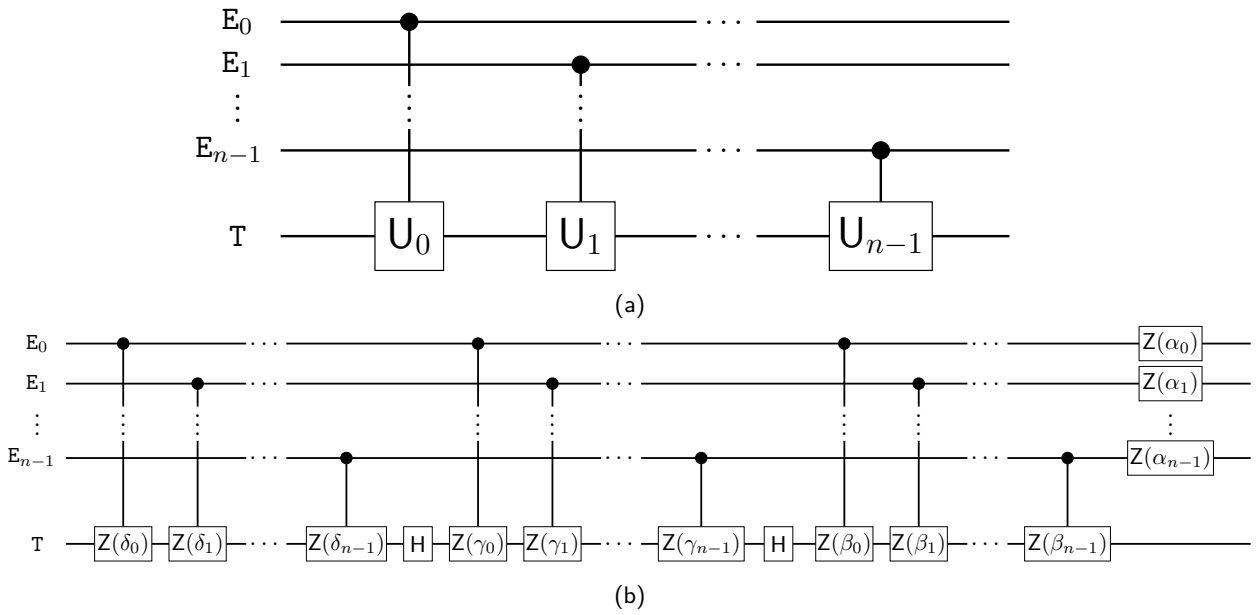


Figure 4: (a) A serial circuit of unitaries $\mathsf{U}_j = e^{i\pi\alpha_j}\mathsf{Z}(\beta_j)\mathsf{HZ}(\gamma_j)\mathsf{HZ}(\delta_j)$ controlled on the single-qubit register $\mathtt{E}_j$ being in the $|1\rangle$ state, $j \in [n]$. (b) If the control qubits $|e\rangle_{\mathtt{E}} = \bigotimes_{j\in[n]} |e_j\rangle_{\mathtt{E}_j}$ are such that $e \in \{0,1\}^n$ has Hamming weight at most 1 ($|e| \leq 1$), then the gates composing $\mathsf{U}_0, \ldots, \mathsf{U}_{n-1}$ can be rearranged as shown in the equivalent circuit.

The above identity holds because at most one controlled gate $\mathsf{C}\text{-}\mathsf{U}_j$ is "active" for the state $|e\rangle_{\mathtt{E}}$. This allows us to group the Z operators from the Z-decomposition of $\mathsf{U}_0, \ldots, \mathsf{U}_{n-1}$ as shown in Figure 4b.

The idea of our quantum circuits is to compute the one-hot encoding of the input string $x \in \{0,1\}^n$, after which the correct phases $\alpha_j, \beta_j, \gamma_j, \delta_j$ can be selected. We note that Low et al. [LKS24] proposed a $O(n^2)$-depth circuit with $O(2^n)$ single and two-qubit gates and no ancillae to compute the one-hot encoding of $x \in \{0,1\}^n$.

5.1 Constant-depth circuits for f-UCGs

Theorem 26 (One-hot-encoding implementation of f-UCG)**.** *Let $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$ be a (J,r)-junta for $J \subseteq [n]$ with $|\overline{J}| = t$ and $r \in \mathbb{N}$. There is a $O(1)$-depth circuit for f-UCG that uses*

- *either $2(t+r)2^{t+r}\log(t+r) + O((t+r)2^{t+r})$ ancillae and $2n + 6(t+r)2^{t+r} + O(2^{t+r}\log(t+r))$ Fan-Out gates with arity $\leq 1 + 2^{t+r}$,*
- *or $3(t+r)2^{t+r} + O(2^{t+r}\log(t+r))$ ancillae and 9 GT gates with arity $\leq n + (t+r)2^{t+r} + O(2^{t+r}\log(t+r))$.*

Proof. Given the initial state $|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}}$ for $x \in \{0,1\}^n$ and $b \in \{0,1\}$, we wish to perform the mapping $|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}} f(x)|b\rangle_{\mathtt{T}}$. For each $z \in \{0,1\}^t$, let $J_z \subseteq J$, with $|J_z| \leq r$, be the subset of coordinates that $f_{J|z}$ depends on. For $z \in \{0,1\}^t$, let $g_z : \{0,1\}^{|J_z|} \to \mathcal{U}(\mathbb{C}^{2\times 2})$ be such that $f_{J|z}(x_J) = g_z(x_{J_z})$. In the following, split the register $\mathtt{I}$ into registers $\overline{\mathtt{J}}$ and $\mathtt{J}$ such that $\overline{\mathtt{J}}$ contains the coordinates of x in $\overline{J}$ and $\mathtt{J}$ contains the coordinates of x in J, i.e., $|x\rangle_{\mathtt{I}} = |x_{\overline{J}}\rangle_{\overline{\mathtt{J}}}|x_J\rangle_{\mathtt{J}}$. For $i \in J$, let $m_i := \sum_{z\in\{0,1\}^t : i\in J_z} 2^{|J_z|} \leq 2^r \cdot |\{z \in \{0,1\}^t : i \in J_z\}|$ and $m := \sum_{z\in\{0,1\}^t} 2^{|J_z|} \leq 2^{t+r}$. Let $f(x) = e^{i\pi\alpha(x)}\mathsf{Z}(\beta(x))\mathsf{HZ}(\gamma(x))\mathsf{HZ}(\delta(x))$ be the Z-decomposition of each single-qubit gate $f(x)$, $x \in \{0,1\}^n$. Equivalently, write $g_z(j) = e^{i\pi\alpha_z(j)}\mathsf{Z}(\beta_z(j))\mathsf{HZ}(\gamma_z(j))\mathsf{HZ}(\delta_z(j))$ for the Z-decomposition of

the single-qubit gate $g_z(j)$, $z \in \{0,1\}^t$ and $j \in \{0,1\}^{|J_z|}$. From $f_{J|z}(x_J) = g_z(x_{J_z})$ we can establish the correspondence that $\alpha(x) = \alpha_z(j)$ for all $z \in \{0,1\}^t$, $j \in \{0,1\}^{|J_z|}$, and $x \in \{0,1\}^n$ such that $x_{\overline{J}} = z$ and $x_{J_z} = j$ (and similarly for β, γ, δ).

The main idea of the circuits is to compute the one-hot encoding of a compressed version of x. Naively, one could compute the one-hot encoding of the whole string x. However, by breaking x into the sub-strings $x_{\overline{J}}$ and x_J indexed by $\overline{J}$ and J, respectively, it is possible to compute the one-hot encoding of $x_{\overline{J}}$ separately from the one-hot encoding of x_J. In principle, this would not offer any advantage, but since $f_{J|z}$ depends only on a few coordinates of x_J, we can compute the one-hot encoding of the much shorter sub-string x_{J_z} for $z = x_{\overline{J}}$ instead. Since we do not know $x_{\overline{J}}$, we must compute the one-hot encoding of x_{J_z} for all $z \in \{0,1\}^t$. We first consider the Fan-Out-based circuit (see Figure 5) and then the one based on GT gates. The algorithm is as follows:

1. For each $z \in \{0,1\}^t$ and $j \in \{0,1\}^{|J_z|}$, attach a t-qubit ancillary register $\overline{\mathtt{J}}_{z,j}$ and copy the contents of the $|x_{\overline{J}}\rangle_{\overline{\mathtt{J}}}$ register onto it. Every qubit of $|x_{\overline{J}}\rangle_{\overline{\mathtt{J}}}$ is thus copied m times.

2. For each $z \in \{0,1\}^t$ and $j \in \{0,1\}^{|J_z|}$, attach a $|J_z|$-qubit ancillary register $\mathtt{J}_{z,j}$. The registers $\mathtt{J}_{z,j}$, for $j \in \{0,1\}^{|J_z|}$, will be used to compute the one-hot encoding of x_{J_z}. For each $i \in J$ in parallel, copy m_i number of times the qubit $|x_i\rangle_{\mathtt{J}}$ onto the registers $\{\mathtt{J}_{z,j}\}_j$, where $j \in \{0,1\}^{|J_z|}$, for all z such that $i \in J_z$. Steps 1 and 2 lead to

$$|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \left(\bigotimes_{z\in\{0,1\}^t} \bigotimes_{j\in\{0,1\}^{|J_z|}} |x_{\overline{J}}\rangle_{\overline{\mathtt{J}}_{z,j}} |x_{J_z}\rangle_{\mathtt{J}_{z,j}} \right).$$

3. For each $z \in \{0,1\}^t$ and $j \in \{0,1\}^{|J_z|}$, apply the gate $\bigotimes_{k\in[t]} \mathsf{X}^{\overline{z}_k}$ onto $|x_{\overline{J}}\rangle_{\overline{\mathtt{J}}_{z,j}}$ and the gate $\bigotimes_{k\in[|J_z|]} \mathsf{X}^{\overline{j}_k}$ onto $|x_{J_z}\rangle_{\mathtt{J}_{z,j}}$. This leads to

$$|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \left(\bigotimes_{z\in\{0,1\}^t} \bigotimes_{j\in\{0,1\}^{|J_z|}} |x_{\overline{J}} \oplus \overline{z}\rangle_{\overline{\mathtt{J}}_{z,j}} |x_{J_z} \oplus \overline{j}\rangle_{\mathtt{J}_{z,j}} \right).$$

4. For each $z \in \{0,1\}^t$ and $j \in \{0,1\}^{|J_z|}$, attach a single-qubit ancillary register $\mathtt{E}_{z,j}$ and apply an $\mathsf{AND}^{(t+|J_z|)}_{\{\overline{\mathtt{J}}_{z,j},\mathtt{J}_{z,j}\}\to\mathtt{E}_{z,j}}$ gate from registers $\overline{\mathtt{J}}_{z,j}$ and $\mathtt{J}_{z,j}$ onto register $\mathtt{E}_{z,j}$ to obtain

$$\begin{aligned}
&|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \left(\bigotimes_{z\in\{0,1\}^t} \bigotimes_{j\in\{0,1\}^{|J_z|}} |x_{\overline{J}} \oplus \overline{z}\rangle_{\overline{\mathtt{J}}_{z,j}} |x_{J_z} \oplus \overline{j}\rangle_{\mathtt{J}_{z,j}} |0\rangle_{\mathtt{E}_{z,j}} \right) \\
&\mapsto |x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \left(\bigotimes_{z\in\{0,1\}^t} \bigotimes_{j\in\{0,1\}^{|J_z|}} |x_{\overline{J}} \oplus \overline{z}\rangle_{\overline{\mathtt{J}}_{z,j}} |x_{J_z} \oplus \overline{j}\rangle_{\mathtt{J}_{z,j}} \Big| \bigwedge\nolimits_{k\in[t]} (x_{\overline{J}} \oplus \overline{z})_k \cdot \bigwedge\nolimits_{l\in[|J_z|]} (x_{J_z} \oplus \overline{j})_l \Big\rangle_{\mathtt{E}_{z,j}} \right) \\
&= |x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \left(\bigotimes_{z\in\{0,1\}^t} \bigotimes_{j\in\{0,1\}^{|J_z|}} |x_{\overline{J}} \oplus \overline{z}\rangle_{\overline{\mathtt{J}}_{z,j}} |x_{J_z} \oplus \overline{j}\rangle_{\mathtt{J}_{z,j}} |e(x_{\overline{J}})_z \cdot e(x_{J_z})_j\rangle_{\mathtt{E}_{z,j}} \right),
\end{aligned}$$

where $e(x_{\overline{J}}) \in \{0,1\}^{2^t}$ and $e(x_{J_z}) \in \{0,1\}^{2^{|J_z|}}$ are the one-hot encodings of $x_{\overline{J}}$ and x_{J_z}, respectively. Note that $e(x_{\overline{J}})_z$ is the z-th bit of $e(x_{\overline{J}})$ and $e(x_{J_z})_j$ is j-th bit of $e(x_{J_z})$.

5a. For each $z \in \{0,1\}^t$ and $j \in \{0,1\}^{|J_z|}$, attach a single-qubit ancillary register $\mathtt{T}_{z,j}$. Apply a $(1+m)$-arity Fan-Out gate $\mathsf{FO}^{(1+m)}_{\mathtt{T}\to\{\mathtt{T}_{z,j}\}_{z,j}}$ from register $\mathtt{T}$ onto registers $\{\mathtt{T}_{z,j}\}_{z,j}$, $z \in \{0,1\}^t$

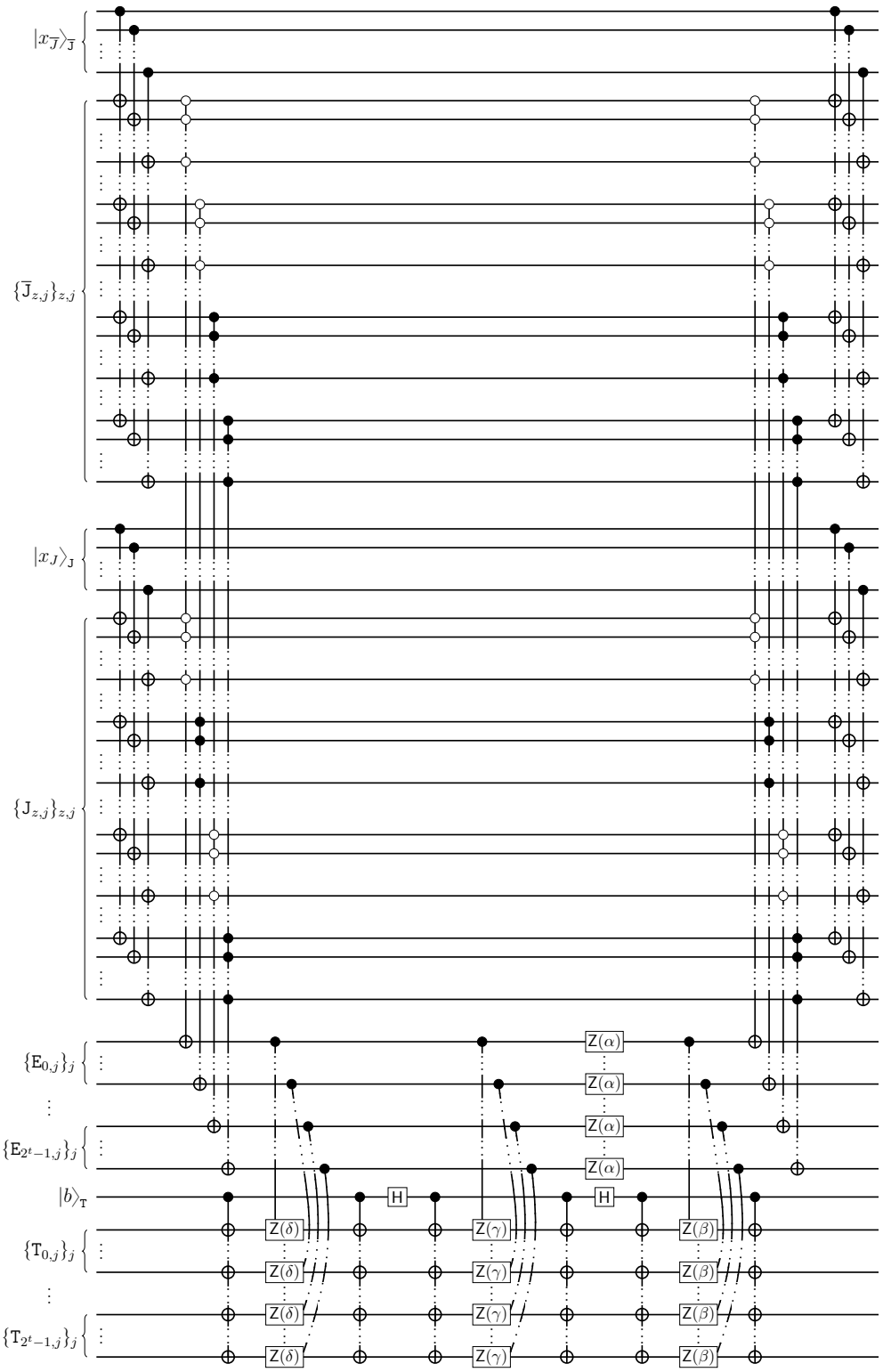


Figure 5: The circuit for an f-UCG$^{(n)}$ using Fan-Out gates, where f is a (J,r)-junta with $|\overline{J}| = t$. For simplicity, we include targets from $|x_i\rangle_{\mathtt{J}}$ onto all registers $\{\mathtt{J}_{z,j}\}_{z,j}$, but in reality x_i is copied onto the registers $\{\mathtt{J}_{z,j}\}_j$ for all $z \in \{0,1\}^t$ such that $i \in J_z$, where J_z is the set of coordinated that $f_{J|z}$ depends on. Moreover, we omit the indices in the parameters $\alpha_z(j)$, $\beta_z(j)$, $\gamma_z(j)$, $\delta_z(j)$.

and $j \in \{0,1\}^{|J_z|}$ (remember that $m := \sum_{z\in\{0,1\}^t} 2^{|J_z|}$). For each $z \in \{0,1\}^t$ and $j \in \{0,1\}^{|J_z|}$, apply a $\mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{Z}(\delta_z(j))_{\to\mathtt{T}_{z,j}}$ gate controlled on register $\mathtt{E}_{z,j}$ onto register $\mathtt{T}_{z,j}$. Finally, apply $\mathsf{FO}^{(1+m)}_{\mathtt{T}\to\{\mathtt{T}_{z,j}\}_{z,j}}$ again. We shall omit the registers $\overline{\mathtt{J}}_{z,j}$ and $\mathtt{J}_{z,j}$ from now on for simplicity. This chain of operations leads to

$$\begin{aligned}
&|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}}\left(\bigotimes_{z\in\{0,1\}^t}\bigotimes_{j\in\{0,1\}^{|J_z|}}|e(x_{\overline{J}})_z e(x_{J_z})_j\rangle_{\mathtt{E}_{z,j}}|b\rangle_{\mathtt{T}_{z,j}}\right)\\
\mapsto\ &|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}}\left(\bigotimes_{z\in\{0,1\}^t}\bigotimes_{j\in\{0,1\}^{|J_z|}}|e(x_{\overline{J}})_z e(x_{J_z})_j\rangle_{\mathtt{E}_{z,j}}\mathsf{Z}(\delta_z(j))^{e(x_{\overline{J}})_z e(x_{J_z})_j}|b\rangle_{\mathtt{T}_{z,j}}\right)\\
\mapsto\ &|x\rangle_{\mathtt{I}}\mathsf{Z}(\delta(x))|b\rangle_{\mathtt{T}}\left(\bigotimes_{z\in\{0,1\}^t}\bigotimes_{j\in\{0,1\}^{|J_z|}}|e(x_{\overline{J}})_z e(x_{J_z})_j\rangle_{\mathtt{E}_{z,j}}\right),
\end{aligned}$$

where we have used the definition of one-hot encoding, i.e., $e(x_{\overline{J}})_z = 1$ if and only if $z = x_{\overline{J}}$ and $e(x_{J_z})_j = 1$ if and only if $j = x_{J_z}$, and also that $\delta_z(j) = \delta(x)$ for $z = x_{\overline{J}}$ and $j = x_{J_z}$.

5b. Apply a $\mathsf{H}_{\to\mathtt{T}}$ gate to register $\mathtt{T}$ followed by a $(1+m)$-arity Fan-Out gate $\mathsf{FO}^{(1+m)}_{\mathtt{T}\to\{\mathtt{T}_{z,j}\}_{z,j}}$ from register $\mathtt{T}$ to registers $\{\mathtt{T}_{z,j}\}_{z,j}$. For each $z \in \{0,1\}^t$ and $j \in \{0,1\}^{|J_z|}$, apply a $\mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{Z}(\gamma_z(j))_{\to\mathtt{T}_{z,j}}$ gate controlled on register $\mathtt{E}_{z,j}$ onto register $\mathtt{T}_{z,j}$. Finally, apply $\mathsf{FO}^{(1+m)}_{\mathtt{T}\to\{\mathtt{T}_{z,j}\}_{z,j}}$ again. For simplicity, write $\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}} = r_{b,x}|0\rangle_{\mathtt{T}} + s_{b,x}|1\rangle_{\mathtt{T}}$. This chain of operations yields

$$\begin{aligned}
&|x\rangle_{\mathtt{I}}\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}}\left(\bigotimes_{z\in\{0,1\}^t}\bigotimes_{j\in\{0,1\}^{|J_z|}}|e(x_{\overline{J}})_z e(x_{J_z})_j\rangle_{\mathtt{E}_{z,j}}\right)\\
\mapsto\ &r_{b,x}|x\rangle_{\mathtt{I}}|0\rangle_{\mathtt{T}}\bigotimes_{z\in\{0,1\}^t}\bigotimes_{j\in\{0,1\}^{|J_z|}}(|e(x_{\overline{J}})_z e(x_{J_z})_j\rangle_{\mathtt{E}_{z,j}}|0\rangle_{\mathtt{T}_{z,j}})\\
&\qquad + s_{b,x}|x\rangle_{\mathtt{I}}|1\rangle_{\mathtt{T}}\bigotimes_{z\in\{0,1\}^t}\bigotimes_{j\in\{0,1\}^{|J_z|}}(|e(x_{\overline{J}})_z e(x_{J_z})_j\rangle_{\mathtt{E}_{z,j}}|1\rangle_{\mathtt{T}_{z,j}})\\
\mapsto\ &r_{b,x}|x\rangle_{\mathtt{I}}|0\rangle_{\mathtt{T}}\bigotimes_{z\in\{0,1\}^t}\bigotimes_{j\in\{0,1\}^{|J_z|}}(|e(x_{\overline{J}})_z e(x_{J_z})_j\rangle_{\mathtt{E}_{z,j}}\mathsf{Z}(\gamma_z(j))^{e(x_{\overline{J}})_z e(x_{J_z})_j}|0\rangle_{\mathtt{T}_{z,j}})\\
&\qquad + s_{b,x}|x\rangle_{\mathtt{I}}|1\rangle_{\mathtt{T}}\bigotimes_{z\in\{0,1\}^t}\bigotimes_{j\in\{0,1\}^{|J_z|}}(|e(x_{\overline{J}})_z e(x_{J_z})_j\rangle_{\mathtt{E}_{z,j}}\mathsf{Z}(\gamma_z(j))^{e(x_{\overline{J}})_z e(x_{J_z})_j}|1\rangle_{\mathtt{T}_{z,j}})\\
\mapsto\ &|x\rangle_{\mathtt{I}}\mathsf{Z}(\gamma(x))\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}}\left(\bigotimes_{z\in\{0,1\}^t}\bigotimes_{j\in\{0,1\}^{|J_z|}}|e(x_{\overline{J}})_z e(x_{J_z})_j\rangle_{\mathtt{E}_{z,j}}\right).
\end{aligned}$$

5c. Apply a $\mathsf{H}_{\to\mathtt{T}}$ gate to register $\mathtt{T}$ followed by a $(1+m)$-arity Fan-Out gate $\mathsf{FO}^{(1+m)}_{\mathtt{T}\to\{\mathtt{T}_{z,j}\}_{z,j}}$ from register $\mathtt{T}$ to registers $\{\mathtt{T}_{z,j}\}_{z,j}$. For each $z \in \{0,1\}^t$ and $j \in \{0,1\}^{|J_z|}$, apply a $\mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{Z}(\beta_z(j))_{\to\mathtt{T}_{z,j}}$ gate controlled on register $\mathtt{E}_{z,j}$ onto register $\mathtt{T}_{z,j}$ followed by a $\mathsf{Z}(\alpha_z(j))_{\to\mathtt{E}_{z,j}}$ gate applied onto register $\mathtt{E}_{z,j}$. Finally, apply $\mathsf{FO}^{(1+m)}_{\mathtt{T}\to\{\mathtt{T}_{z,j}\}_{z,j}}$ again. Similarly to the previous step, we get (consider only registers $\mathtt{I}$ and $\mathtt{T}$ for simplicity)

$$|x\rangle_{\mathtt{I}}\mathsf{Z}(\gamma(x))\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}}e^{i\pi\alpha(x)}\mathsf{Z}(\beta(x))\mathsf{HZ}(\gamma(x))\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}} = |x\rangle_{\mathtt{I}}f(x)|b\rangle_{\mathtt{T}}.$$

6. Uncompute Steps 4, 3, 2, and 1. This leads to the desired state $|x\rangle_{\mathtt{I}}f(x)|b\rangle_{\mathtt{T}}$.

We now consider the GT-gate-based circuit (see Figure 6), which is basically the same as the Fan-Out-based circuit, but replacing Steps 5a, 5b, and 5c with the following Step 5:

5. Apply the gate

$$\left(\prod_{z\in\{0,1\}^t}\prod_{j\in\{0,1\}^{|J_z|}}\mathsf{Z}(\alpha_z(j))_{\to\mathtt{E}_{z,j}}\right)\left(\prod_{z\in\{0,1\}^t}\prod_{j\in\{0,1\}^{|J_z|}}\mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{Z}(\beta_z(j))_{\to\mathtt{T}}\right)\mathsf{H}_{\to\mathtt{T}}$$
$$\cdot\left(\prod_{z\in\{0,1\}^t}\prod_{j\in\{0,1\}^{|J_z|}}\mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{Z}(\gamma_z(j))_{\to\mathtt{T}}\right)\mathsf{H}_{\to\mathtt{T}}\left(\prod_{z\in\{0,1\}^t}\prod_{j\in\{0,1\}^{|J_z|}}\mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{Z}(\delta_z(j))_{\to\mathtt{T}}\right)$$

using 3 GT gates (one for each $\prod_{z\in\{0,1\}^t}\prod_{j\in\{0,1\}^{|J_z|}}\mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{Z}(\cdot)_{\to\mathtt{T}}$). Since $|e(x_{\overline{J}})|=|e(x_{J_z})|=1$, this leads to $|x\rangle_{\mathtt{I}}f(x)|b\rangle_{\mathtt{T}}$ up to the ancillary registers.

We now analyse the resources required for each step:

- Step 1: the registers $\overline{\mathtt{J}}_{z,j}$, for $z\in\{0,1\}^t$ and $j\in\{0,1\}^{|J_z|}$, use at most $t2^{t+r}$ ancillae and copying the register $\overline{\mathtt{J}}$ requires either t Fan-Out gates with arity at most $1+2^{t+r}$ or 1 GT gate with arity at most $t(1+2^{t+r})$;
- Step 2: the registers $\mathtt{J}_{z,j}$, for $z\in\{0,1\}^t$ and $j\in\{0,1\}^{|J_z|}$, use at most $r2^{t+r}$ ancillae and copying the register $\mathtt{J}$ requires either $|J|=n-t$ Fan-Out gates with arity at most $1+\max_{i\in J}m_i\leq 1+2^{t+r}$ or 1 GT (which can be absorbed by the one from Step 1) with arity at most $n-t+\sum_{i\in J}m_i\leq n-t+r2^{t+r}$;
- Step 4: the registers $\mathtt{E}_{z,t}$, for $z\in\{0,1\}^t$ and $j\in\{0,1\}^{|J_z|}$, use at most 2^{t+r} ancillae. The $m\leq 2^{t+r}$ $\mathsf{AND}^{(t+|J_z|)}_{\{\overline{\mathtt{J}}_{z,j},\mathtt{J}_{z,j}\}\to\mathtt{E}_{z,j}}$ gates require either $2(t+r)2^{t+r}\log(t+r)+O((t+r)2^{t+r})$ ancillae by using the construction based on Fan-Out gates (Fact 18) or $2(t+r)2^{t+r}+O(2^{t+r}\log(t+r))$ ancillae by using the construction based on GT gates (Fact 24). Naively, one would expect the $m\leq 2^{t+r}$ $\mathsf{AND}^{(t+|J_z|)}_{\{\overline{\mathtt{J}}_{z,j},\mathtt{J}_{z,j}\}\to\mathtt{E}_{z,j}}$ gates to require either $6(t+r)2^{t+r}+O(2^{t+r}\log(t+r))$ Fan-Out gates with arity at most $2(t+r)$ (Fact 18) or 4 GT gates with arity at most $(t+r)2^{t+r}+O(2^{t+r}\log(t+r))$ (Fact 24), but we can postpone their inner uncomputation part until Step 6 and carry over all the required ancillae. This means that Step 4 requires only $3(t+r)2^{t+r}+O(2^{t+r}\log(t+r))$ Fan-Out gates or 2 GT gates;
- Step 5: the Fan-Out-based circuit requires at most 2^{t+r} ancillae for the registers $\mathtt{T}_{z,j}$, where $z\in\{0,1\}^t$ and $j\in\{0,1\}^{|J_z|}$, and 6 Fan-Out gates with arity at most $1+2^{t+r}$. The GT-gate-based circuit does not require any ancillae, and only 3 GT gates with arity $\leq 1+2^{t+r}$;
- Step 6: uncomputing uses either $n+3(t+r)2^{t+r}+O(2^{t+r}\log(t+r))$ Fan-Outs or 3 GT gates.

In total, we require either $2(t+r)2^{t+r}\log(t+r)+O((t+r)2^{t+r})$ ancillae and $2n+6(t+r)2^{t+r}+O(2^{t+r}\log(t+r))$ Fan-Out gates with arity at most $1+2^{t+r}$, or $3(t+r)2^{t+r}+O(2^{t+r}\log(t+r))$ ancillae and 9 GT gates with arity at most $n+(t+r)2^{t+r}+O(2^{t+r}\log(t+r))$. □

5.2 Constant-depth circuits for f-FIN

The circuits from the previous section can be used to implement an f-FIN, since they are a special case of f-UCGs, as explained before Definition 14. Nonetheless, the circuits from the previous section can be simplified due to their simpler structure, i.e., the Z-decomposition of an f-FIN is

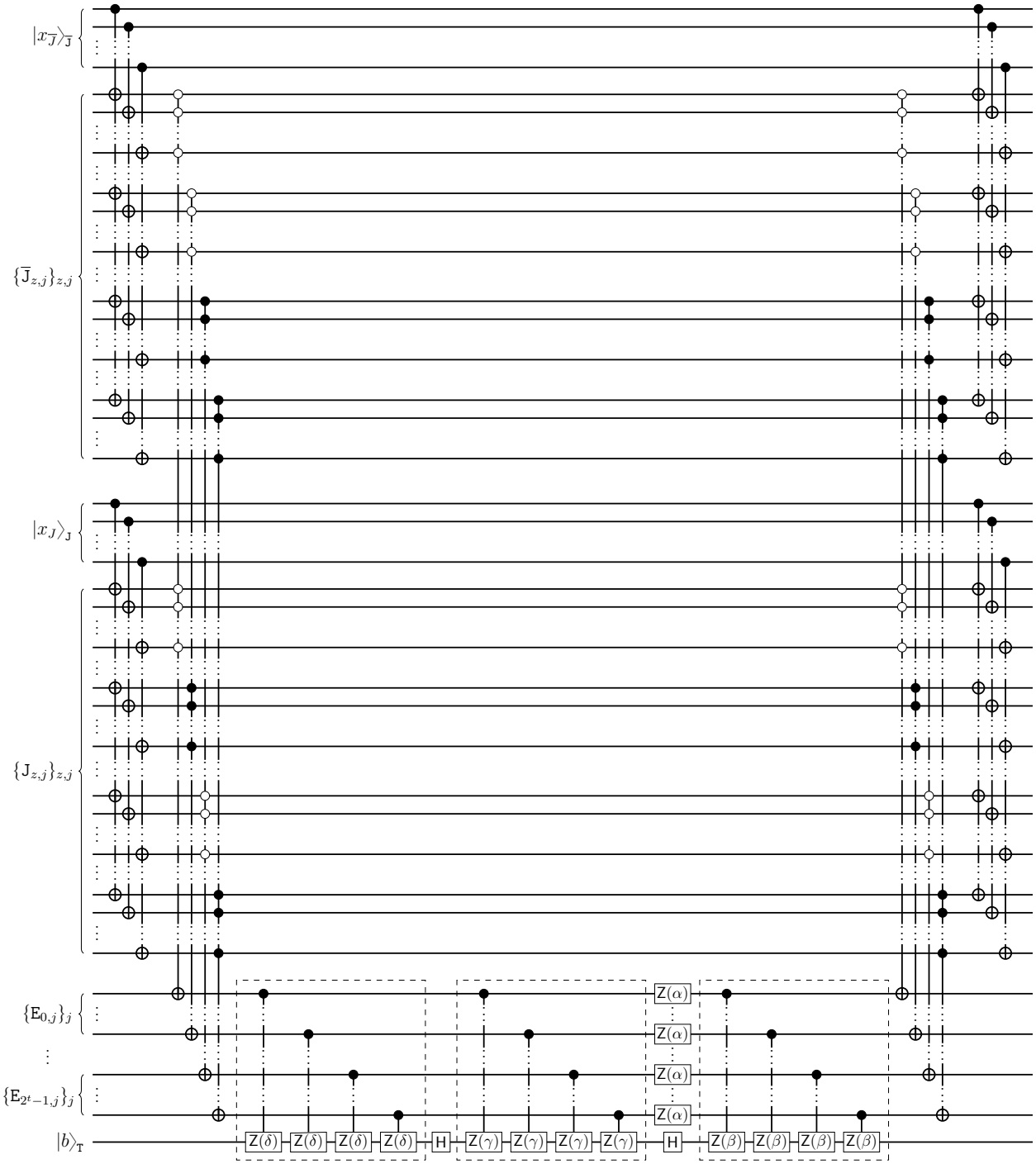


Figure 6: The circuit for an f-UCG$^{(n)}$ using GT gates, where f is a (J, r)-junta with $|\overline{J}| = t$. We highlight the GT gates inside the dashed boxes. For simplicity, we include targets from $|x_i\rangle_{\mathrm{J}}$ onto all registers $\{\mathrm{J}_{z,j}\}_{z,j}$, but in reality x_i is copied onto the registers $\{\mathrm{J}_{z,j}\}_j$ for all $z \in \{0,1\}^t$ such that $i \in J_z$, where J_z is the set of coordinated that $f_{J|z}$ depends on. Moreover, we omit the indices in the parameters $\alpha_z(j)$, $\beta_z(j)$, $\gamma_z(j)$, $\delta_z(j)$.

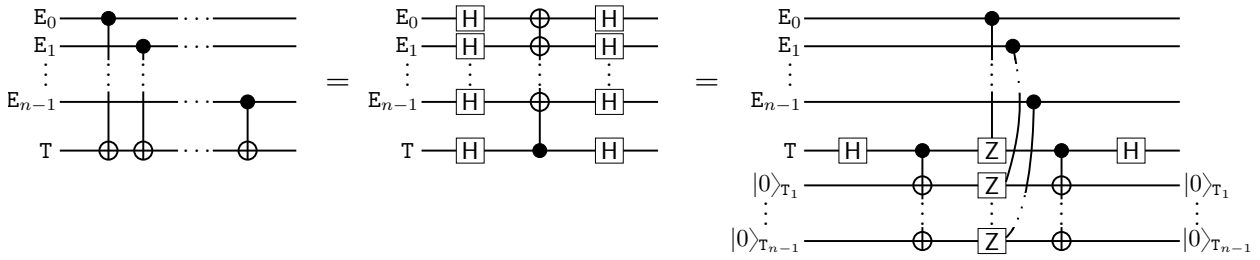


Figure 7: The gate $\prod_{j\in[n]} \mathsf{C}_{\mathtt{E}_j}\text{-}\mathsf{X}_{\to\mathtt{T}}$ (first circuit) is equivalent to $\mathsf{PARITY}_{\{\mathtt{E}_j\}_{j\in[n]}\to\mathtt{T}}$ (second circuit) and to $\mathsf{H}_{\to\mathtt{T}}\mathsf{FO}_{\mathtt{T}\to\{\mathtt{T}_j\}_{j=1}^{n-1}}\big(\prod_{j\in[n]} \mathsf{C}_{\mathtt{E}_j}\text{-}\mathsf{Z}_{\to\mathtt{T}_j}\big)\mathsf{FO}_{\mathtt{T}\to\{\mathtt{T}_j\}_{j=1}^{n-1}}\mathsf{H}_{\to\mathtt{T}}$ (third circuit) (define $\mathtt{T}_0 := \mathtt{T}$), where the ancillary registers $\{\mathtt{T}_j\}_{j=1}^{n-1}$ are initialized in the $|0\rangle$ state.

$\mathsf{HZ}(f(x))\mathsf{H} = \mathsf{X}^{f(x)}$. In particular, the controlled gates $\mathsf{H}_{\to\mathtt{T}}\mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{Z}(\gamma_z(j))_{\to\mathtt{T}}\mathsf{H}_{\to\mathtt{T}} = \mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{X}^{\gamma_z(j)}_{\to\mathtt{T}}$, where $\gamma_z : \{0,1\}^{|J_z|} \to \{0,1\}$, that arise from the Z-decomposition can be replaced by a single PARITY gate (recall that the PARITY gate can be implemented by a single Fan-Out gate (Fact 17)). We show how this can be done in the next result.

Theorem 27 (One-hot-encoding implementation of f-FIN)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a (J,r)-junta for $J \subseteq [n]$ with $|\overline{J}| = t$ and $r \in \mathbb{N}$. There is a $O(1)$-depth circuit for f-FIN that uses*

- *either $2(t+r)2^{t+r}\log(t+r)+O((t+r)2^{t+r})$ ancillae and $2n+6(t+r)2^{t+r}+O(2^{t+r}\log(t+r))$ Fan-Out gates with arity $\leq 1+2^{t+r}$,*
- *or $3(t+r)2^{t+r}+O(2^{t+r}\log(t+r))$ ancillae and 6 GT gates with arity $\leq n+(t+r)2^{t+r}+O(2^{t+r}\log(t+r))$.*

Proof. Construct the state

$$|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}}\left(\bigotimes_{z\in\{0,1\}^t}\bigotimes_{j\in\{0,1\}^{|J_z|}}|x_{\overline{J}}\oplus\overline{z}\rangle_{\overline{\mathtt{J}}_{z,j}}|x_{J_z}\oplus\overline{j}\rangle_{\mathtt{J}_{z,j}}|e(x_{\overline{J}})_z\cdot e(x_{J_z})_j\rangle_{\mathtt{E}_{z,j}}\right)$$

by following the same steps as in Theorem 26. To perform the f-FIN gate, we must apply a $\mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{X}_{\to\mathtt{T}}$ gate for all $z\in\{0,1\}^t$ and $j\in g_z^{-1}(1)$, where $g_z : \{0,1\}^{|J_z|}\to\{0,1\}$ is such that $f_{J|z}(x_J) = g_z(x_{J_z})$, since this leads to (consider only register $\mathtt{T}$)

$$|b\rangle_{\mathtt{T}} \mapsto |b\oplus\bigoplus_{z\in\{0,1\}^t}\bigoplus_{j\in g_z^{-1}(1)}e(x_{\overline{J}})_z\cdot e(x_{J_z})_j\rangle_{\mathtt{T}} = |b\oplus g_{x_{\overline{J}}}(x_{J_{x_{\overline{J}}}})\rangle_{\mathtt{T}} = |b\oplus f(x)\rangle_{\mathtt{T}},$$

where we used (i) the definition of one-hot encoding, $e(x_{\overline{J}})_z = 1$ if and only if $z = x_{\overline{J}}$, and $e(x_{J_z})_j = 1$ if and only if $j = x_{J_z}$; (ii) the fact that $\bigoplus_{j\in g_z^{-1}(1)} e(y)_j = g_z(y)$ for any $z\in\{0,1\}^t$ and $y\in\{0,1\}^{|J_z|}$; (iii) the identity $g_{x_{\overline{J}}}(x_{J_{x_{\overline{J}}}}) = f_{J|x_{\overline{J}}}(x_J) = f(x)$.

There are two methods to apply the $\mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{X}_{\to\mathtt{T}}$ gates in parallel (see Figure 7). The first method is via [Moo99]

$$\prod_{z\in\{0,1\}^t}\prod_{j\in g_z^{-1}(1)}\mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{X}_{\to\mathtt{T}} = \mathsf{PARITY}_{\{\mathtt{E}_{z,j}\}_{z\in\{0,1\}^t, j\in g_z^{-1}(1)}\to\mathtt{T}},$$

i.e., applying an X onto $|b\rangle_{\mathtt{T}}$ controlled on $|e(x_{\overline{J}})_z\cdot e(x_{J_z})_j\rangle_{\mathtt{E}_{z,j}}$ for all $z\in\{0,1\}^t$ and $j\in g_z^{-1}(1)$ is equivalent to applying a PARITY gate onto $|b\rangle_{\mathtt{T}}$ from input $\bigotimes_{z\in\{0,1\}^t}\bigotimes_{j\in g_z^{-1}(1)}|e(x_{\overline{J}})_z e(x_{J_z})_j\rangle_{\mathtt{E}_{z,j}}$. The PARITY gate costs 1 Fan-Out or GT gate with arity $1+\sum_{z\in\{0,1\}^t}|g_z^{-1}(1)|$.

The second method is to use the parallelisation method from [GHMP02, MN01, HŠ05]. More specifically, by using the $\sum_{z\in\{0,1\}^t} |g_z^{-1}(1)| \leq 2^{t+r}$ ancillary registers $\mathtt{T}_{z,j}$, $z \in \{0,1\}^t$ and $j \in g_z^{-1}(1)$, initialized in the $|0\rangle$ state (note that we do not require all registers $\mathtt{T}_{z,j}$ from Theorem 26), then

$$\prod_{z\in\{0,1\}^t} \prod_{j\in g_z^{-1}(1)} \mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{X}_{\to\mathtt{T}}$$

$$= \mathsf{H}_{\to\mathtt{T}}\mathsf{FO}_{\mathtt{T}\to\{\mathtt{T}_{z,j}\}_{z,j}} \left(\prod_{z\in\{0,1\}^t} \prod_{j\in g_z^{-1}(1)} \mathsf{C}_{\mathtt{E}_{z,j}}\text{-}\mathsf{Z}_{\to\mathtt{T}_{z,j}} \right) \mathsf{FO}_{\mathtt{T}\to\{\mathtt{T}_{z,j}\}_{z,j}} \mathsf{H}_{\to\mathtt{T}} \bigotimes_{z\in\{0,1\}^t} \bigotimes_{j\in g_z^{-1}(1)} |0\rangle_{\mathtt{T}_{z,j}}.$$

In more details, we have (consider only registers $\mathtt{T}$ and $\mathtt{T}_{z,j}$)

$$\begin{aligned}
|b\rangle_{\mathtt{T}} &\mapsto \frac{1}{\sqrt{2}}|0\rangle_{\mathtt{T}} \bigotimes_{z\in\{0,1\}^t} \bigotimes_{j\in g_z^{-1}(1)} |0\rangle_{\mathtt{T}_{z,j}} + \frac{(-1)^b}{\sqrt{2}}|1\rangle_{\mathtt{T}} \bigotimes_{z\in\{0,1\}^t} \bigotimes_{j\in g_z^{-1}(1)} |1\rangle_{\mathtt{T}_{z,j}} \\
&\mapsto \frac{1}{\sqrt{2}}|0\rangle_{\mathtt{T}} \bigotimes_{z\in\{0,1\}^t} \bigotimes_{j\in g_z^{-1}(1)} |0\rangle_{\mathtt{T}_{z,j}} + \frac{(-1)^b}{\sqrt{2}}|1\rangle_{\mathtt{T}} \bigotimes_{z\in\{0,1\}^t} \bigotimes_{j\in g_z^{-1}(1)} (-1)^{e(x_{\overline{J}})_z e(x_J)_j} |1\rangle_{\mathtt{T}_{z,j}} \\
&\mapsto \frac{|0\rangle_{\mathtt{T}} + (-1)^{b+f(x)}|1\rangle_{\mathtt{T}}}{\sqrt{2}} \\
&\mapsto |b \oplus f(x)\rangle_{\mathtt{T}}.
\end{aligned}$$

The above requires 2 Fan-Out or GT gates with arity at most $1 + 2^{t+r}$. We crucially remark that the GT gates can be absorbed by the ones from computing and uncomputing registers $\mathtt{E}_{z,j}$.

The rest of the circuit is identical to Theorem 26: uncompute the ancillary registers. The number of ancillae and Fan-Out gates is asymptotically the same as in Theorem 26. The number of GT gates is reduced from 9 to 7 by using the first method or to 6 by using the second method. □

5.3 Constant-depth circuits for quantum memory devices via one-hot encoding

In this section, we apply our previous circuit constructions based on one-hot encoding to the case of QRAM and QRAG. As mentioned before, QRAM is simply an f-FIN with the Boolean function $f : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}$ defined by $f(x,i) = x_i$. Furthermore, this Boolean function is a $(J,1)$-junta with $J = [n]$ and $|\overline{J}| = \log n$. Indeed, by fixing the coordinates of $i \in [n]$, $f_{[n]|i}(x) = x_i$ depends only on one input coordinate. Theorem 27 thus immediately applies to any QRAM by setting $r = 1$ and $t = |\overline{J}| = \log n$. (Actually, there is no need to compute the one-hot encoding of register $\mathtt{J}$ since $r = 1$. This means that we only require registers $\mathtt{E}_z$ for $z \in \{0,1\}^t$. The number of ancillae is thus halved). For completeness we depict the circuit in Figure 8.

Theorem 28 (One-hot-encoding implementation of QRAM)**.** *For every $n \in \mathbb{N}$ a power of 2, a* QRAM *of memory size n can be implemented in $O(1)$-depth using*

- *either $2n\log n\log\log n + O(n\log n)$ ancillae and $6n\log n + O(n\log\log n)$ Fan-Out gates with arity $\leq n+1$,*
- *or $3n\log n + O(n\log\log n)$ ancillae and 6 GT gates with arity $\leq n\log n + O(n\log\log n)$.*

Even though QRAG is not an f-FIN or even an f-UCG, it is possible to use the one-hot encoding framework from the previous circuit constructions to implement QRAG in constant depth. We mention that a similar $O(1)$-depth and $O(n\log n\log\log n)$-size circuit for QRAG based on one-hot encoding and using Fan-Out gates had previously appeared in [Ros21b, Lemma 4.3]. We note that author missed the $\log\log n$-factor by ignoring the $\log n$-size overhead in Fact 18.

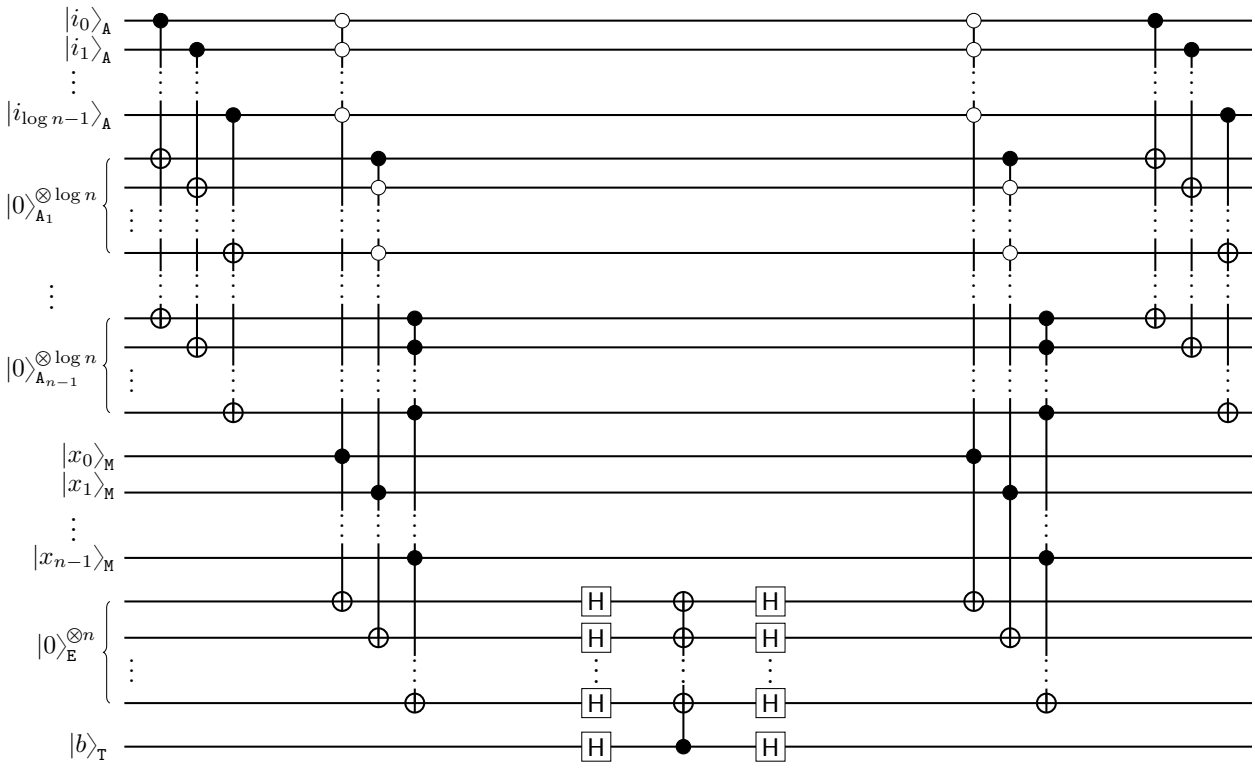


Figure 8: The circuit for the QRAM $|i\rangle_{\mathtt{A}}|b\rangle_{\mathtt{T}}|x\rangle_{\mathtt{M}} \mapsto |i\rangle_{\mathtt{A}}|b \oplus x_i\rangle_{\mathtt{T}}|x\rangle_{\mathtt{M}}$.

Theorem 29 (One-hot-encoding implementation of QRAG)**.** *For every $n \in \mathbb{N}$ a power of 2, a* QRAG *of memory size n can be implemented in $O(1)$-depth using*

- *either* $2n \log n \log\log n + O(n \log n)$ *ancillae and* $6n \log n + O(n \log\log n)$ *Fan-Out gates with arity* $\leq n + 1$,
- *or* $3n \log n + O(n \log\log n)$ *ancillae and* 9 GT *gates with arity* $\leq n \log n + O(n \log\log n)$.

Proof. Given the state $|i\rangle_{\mathtt{A}}|b\rangle_{\mathtt{T}}|x_0, \ldots, x_{n-1}\rangle_{\mathtt{M}}$, we shall compute the one-hot encoding $e(i) \in \{0,1\}^n$ of the address $i \in \{0,1\}^{\log n}$ given by $e(i)_j = \bigwedge_{k \in [\log n]} (i \oplus \bar{j})_k$, where $j \in \{0,1\}^{\log n}$. Since $e(i)_j = 1$ if and only if $j = i$, the one-hot encoding is then used to swap the correct entry x_i from the memory $\mathtt{M}$ onto an n-qubit ancillary register $\mathtt{B}$. The swapped entry in register $\mathtt{B}$ is then mapped onto the target register $\mathtt{T}$ by using a PARITY gate. At this point, both registers $\mathtt{M}$ and $\mathtt{T}$ are in the desired state. The final step is uncomputing register $\mathtt{B}$ with an additional ancillary register $\mathtt{C}$. Consider the following circuit (see Figure 9):

1. Attach an $((n-1)\log n)$-qubit ancillary register $\bigotimes_{j=1}^{n-1} |0\rangle_{\mathtt{A}_j}^{\otimes \log n}$ and copy $n-1$ times the register $|i\rangle_{\mathtt{A}}$ using either $\log n$ Fan-Out gates with arity n or 1 GT gate with arity $n \log n$.
2. Attach an n-qubit ancillary register $|0\rangle_{\mathtt{B}}^{\otimes n} = \bigotimes_{j \in [n]} |0\rangle_{\mathtt{B}_j}$ and apply an $(n+1)$-arity Fan-Out gate $\mathsf{FO}_{\mathtt{T} \to \mathtt{B}}^{(n+1)}$ from register $\mathtt{T}$ to register $\mathtt{B}$ to copy n times the register $|b\rangle_{\mathtt{T}}$.
3. For each $j \in [n]$, apply the gate $\bigotimes_{k \in [\log n]} \mathsf{X}^{\bar{j}_k}$ to $|i\rangle_{\mathtt{A}_j}$ (define $\mathtt{A}_0 := \mathtt{A}$). This leads to

$$\left(\bigotimes_{j \in [n]} |i\rangle_{\mathtt{A}_j} |b\rangle_{\mathtt{B}_j} |x_j\rangle_{\mathtt{M}_j} \right) |b\rangle_{\mathtt{T}} \mapsto \left(\bigotimes_{j \in [n]} |i \oplus \bar{j}\rangle_{\mathtt{A}_j} |b\rangle_{\mathtt{B}_j} |x_j\rangle_{\mathtt{M}_j} \right) |b\rangle_{\mathtt{T}}.$$

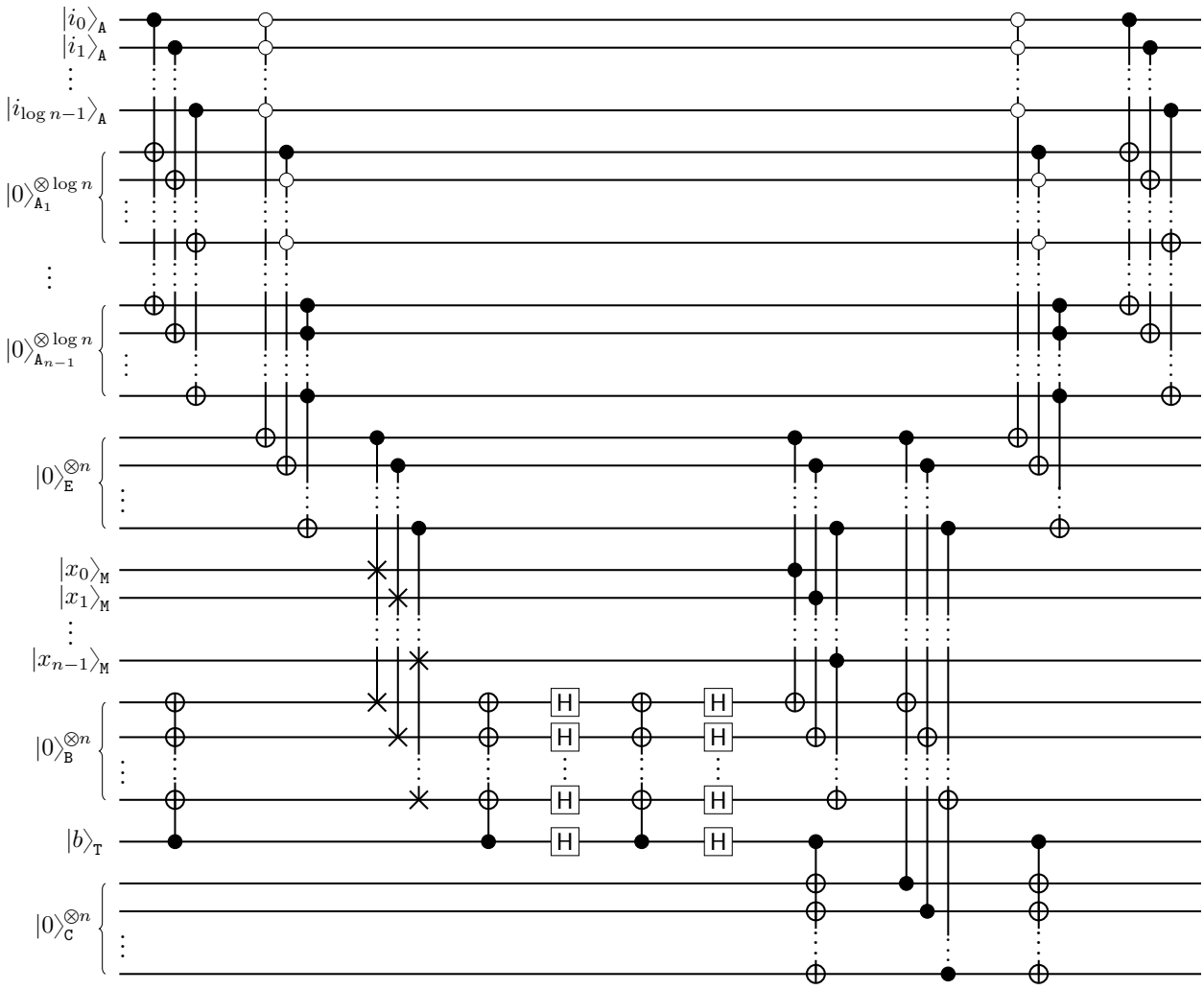


Figure 9: The circuit for the QRAG $|i\rangle_{\mathtt{A}}|b\rangle_{\mathtt{T}}|x\rangle_{\mathtt{M}} \mapsto |i\rangle_{\mathtt{A}}|x_i\rangle_{\mathtt{T}}|x_0,\ldots,x_{i-1},b,x_{i+1},\ldots,x_{n-1}\rangle_{\mathtt{M}}$. The symbol ×—× means a SWAP gate.

4. Attach a new n-qubit ancillary register $|0\rangle^{\otimes n}_{\mathtt{E}} = \bigotimes_{j\in[n]}|0\rangle_{\mathtt{E}_j}$ and apply an $\mathsf{AND}^{(\log n)}_{\mathtt{A}_j\to\mathtt{E}_j}$ gate from register $\mathtt{A}_j$ onto register $\mathtt{E}_j$ for all $j\in[n]$ to obtain

$$\left(\bigotimes_{j\in[n]}|i\oplus\bar{j}\rangle_{\mathtt{A}_j}|b\rangle_{\mathtt{B}_j}|x_j\rangle_{\mathtt{M}_j}|0\rangle_{\mathtt{E}_j}\right)|b\rangle_{\mathtt{T}} \mapsto \left(\bigotimes_{j\in[n]}|i\oplus\bar{j}\rangle_{\mathtt{A}_j}|b\rangle_{\mathtt{B}_j}|x_j\rangle_{\mathtt{M}_j}|e(i)_j\rangle_{\mathtt{E}_j}\right)|b\rangle_{\mathtt{T}},$$

where $e(i)\in\{0,1\}^n$ is the one-hot encoding of i.

5. Apply n $\mathsf{C}_{\mathtt{E}_j}$-$\mathsf{SWAP}_{\mathtt{B}_j\leftrightarrow\mathtt{M}_j}$ gates in parallel for $j\in[n]$, i.e., swap registers $\mathtt{B}_j$ and $\mathtt{M}_j$ controlled on $\mathtt{E}_j$. Since $e(i)_j=1$ if and only if $j=i$, we obtain (ignore the register $\bigotimes_{j\in[n]}|i\oplus\bar{j}\rangle_{\mathtt{A}_j}$ for clarity)

$$|e(i)\rangle_{\mathtt{E}}|b,\ldots,b,x_i,b,\ldots,b\rangle_{\mathtt{B}}|x_0,\ldots,x_{i-1},b,x_{i+1},\ldots,x_{n-1}\rangle_{\mathtt{M}}|b\rangle_{\mathtt{T}}.$$

6. Apply an $(n+1)$-arity Fan-Out gate $\mathsf{FO}^{(n+1)}_{\mathtt{T}\to\mathtt{B}}$ from register $\mathtt{T}$ onto register $\mathtt{B}$ to get

$$|e(i)\rangle_{\mathtt{E}}|0,\ldots,0,b\oplus x_i,0,\ldots,0\rangle_{\mathtt{B}}|x_0,\ldots,x_{i-1},b,x_{i+1},\ldots,x_{n-1}\rangle_{\mathtt{M}}|b\rangle_{\mathtt{T}}.$$

7. Apply a $\mathsf{PARITY}_{\mathtt{B}\to\mathtt{T}}$ gate from register $\mathtt{B}$ onto register $\mathtt{T}$ to obtain

$$|e(i)\rangle_{\mathtt{E}}|0,\dots,0,b\oplus x_i,0,\dots,0\rangle_{\mathtt{B}}|x_0,\dots,x_{i-1},b,x_{i+1},\dots,x_{n-1}\rangle_{\mathtt{M}}|x_i\rangle_{\mathtt{T}}.$$

8. Apply n $\mathsf{C}_{\{\mathtt{E}_j,\mathtt{M}_j\}}\text{-}\mathsf{X}_{\to\mathtt{B}_j}$ gates in parallel for $j\in[n]$, i.e., apply an X gate onto register $\mathtt{B}_j$ controlled on registers $\mathtt{E}_j$ and $\mathtt{M}_j$. This yields

$$|e(i)\rangle_{\mathtt{E}}|0,\dots,0,x_i,0,\dots,0\rangle_{\mathtt{B}}|x_0,\dots,x_{i-1},b,x_{i+1},\dots,x_{n-1}\rangle_{\mathtt{M}}|x_i\rangle_{\mathtt{T}}.$$

9. Attach a new n-qubit ancillary register $|0\rangle_{\mathtt{C}}^{\otimes n}$ and apply an $(n+1)$-arity Fan-Out gate $\mathsf{FO}_{\mathtt{T}\to\mathtt{C}}^{(n+1)}$ from register $\mathtt{T}$ to register $\mathtt{C}$ to get (ignore register $\mathtt{M}$ for clarity)

$$|e(i)\rangle_{\mathtt{E}}|0,\dots,0,x_i,0,\dots,0\rangle_{\mathtt{B}}|0\rangle_{\mathtt{C}}^{\otimes n}|x_i\rangle_{\mathtt{T}}\mapsto|e(i)\rangle_{\mathtt{E}}|0,\dots,0,x_i,0,\dots,0\rangle_{\mathtt{B}}|x_i\rangle_{\mathtt{C}}^{\otimes n}|x_i\rangle_{\mathtt{T}}.$$

10. Apply n $\mathsf{C}_{\{\mathtt{E}_j,\mathtt{C}_j\}}\text{-}\mathsf{X}_{\to\mathtt{B}_j}$ gates in parallel for $j\in[n]$ to uncompute register $\mathtt{B}$ and get

$$|e(i)\rangle_{\mathtt{E}}|0,\dots,0,x_i,0,\dots,0\rangle_{\mathtt{B}}|x_i\rangle_{\mathtt{C}}^{\otimes n}|x_i\rangle_{\mathtt{T}}\mapsto|e(i)\rangle_{\mathtt{E}}|0\rangle_{\mathtt{B}}^{\otimes n}|x_i\rangle_{\mathtt{C}}^{\otimes n}|x_i\rangle_{\mathtt{T}}.$$

11. Uncompute Steps 9, 4, 3, 2, and 1. This leads to $|i\rangle_{\mathtt{A}}|x_i\rangle_{\mathtt{T}}|x_0,\dots,x_{i-1},b,x_{i+1},\dots,x_{n-1}\rangle_{\mathtt{M}}$.

We now analyse the resources for each step:

- Steps 1 and 2: the registers $\mathtt{A}_1,\dots,\mathtt{A}_{n-1}$ and $\mathtt{B}_0,\dots,\mathtt{B}_{n-1}$ use $(n-1)\log n+n$ ancillae. Copying register $\mathtt{A}$ onto $\mathtt{A}_1,\dots,\mathtt{A}_{n-1}$ and register $\mathtt{T}$ onto $\mathtt{B}_0,\dots,\mathtt{B}_{n-1}$ requires either $\log n+1$ Fan-Out gates with arity at most $n+1$ or 1 GT gate with arity $n\log n+n+1$;
- Step 3: the n $\mathsf{AND}_{\mathtt{A}_j\to\mathtt{E}_j}^{(\log n)}$ gates use either $2n\log n\log\log n+O(n\log n)$ ancillae and $3n\log n+O(n\log\log n)$ Fan-Out gates with arity at most $2\log n$ (Fact 18) or $2n\log n+O(n\log\log n)$ ancillae and 2 GT gates with arity $n\log n+O(n\log\log n)$ (Fact 24), where for both cases we pushed the uncomputation part of the $\mathsf{AND}_{\mathtt{A}_j\to\mathtt{E}_j}^{(\log n)}$ gates to Step 11;
- Step 6: copying register $\mathtt{T}$ onto $\mathtt{B}$ uses 1 Fan-Out or 1 GT gate with arity $n+1$;
- Step 7: the $\mathsf{PARITY}_{\mathtt{B}\to\mathtt{T}}$ gate uses 1 Fan-Out or 1 GT gate with arity $n+1$;
- Step 9: copying register $\mathtt{T}$ onto $\mathtt{C}$ uses 1 Fan-Out or 1 GT gate with arity $n+1$;
- Step 11: uncomputing the previous steps uses $3n\log n+O(n\log\log n)$ Fan-Out gates or 3 GT gates, since the GT gate from uncomputing Step 9 can be absorbed by the ones from uncomputing Steps 1, 2, or 3.

In total, we require either $2n\log n\log\log n+O(n\log n)$ ancillae and $6n\log n+O(n\log\log n)$ Fan-Out gates with arity at most $n+1$ or $3n\log n+O(n\log\log n)$ ancillae and 9 GT gates with arity at most $n\log n+O(n\log\log n)$. □

We note that the number of ancillae for QRAM and QRAG can be asymptotically reduced at the expense of at most a constant factor increase in depth. This is achieved by reducing the problem into small blocks, each of which is solved using a small $\mathsf{QRAM}/\mathsf{QRAG}$ circuit. The outcome of all the small circuits is then solved by another small $\mathsf{QRAM}/\mathsf{QRAG}$ circuit. This can be done recursively in a tree-wise fashion, where the output of one level of $\mathsf{QRAM}/\mathsf{QRAG}$ circuits is broken into small blocks and inputted into a new level of $\mathsf{QRAM}/\mathsf{QRAG}$ circuits. This is formalised in the next result. Note that the same idea was used for the OR function, see [HŠ05, Theorem 6.3] and [TT16, Section 3.2].

Theorem 30. *For every $n, d \in \mathbb{N}$, a* QRAM *of memory size n can be performed in $O(d)$-depth using*

- *either $O(n \log^{(d)} n \log^{(d+1)} n)$ ancillae and $O(n \log^{(d)} n)$ Fan-Out gates,*
- *or $O(n \log^{(d)} n)$ ancillae and $16d - 10$ GT gates.*

Moreover, a QRAG *of memory size n can be performed in $O(d)$-depth using*

- *either $O(n \log^{(d)} n \log^{(d+1)} n)$ ancillae and $O(n \log^{(d)} n)$ Fan-Out gates,*
- *or $O(n \log^{(d)} n)$ ancillae and $21d - 12$ GT gates.*

Proof. We first focus on QRAM. The proof is by induction on d. For $d = 1$, the result follows from Theorem 28. Assume that the result is true for $d - 1$. Divide the input $x \in \{0,1\}^n$ into $m := n/\log^{(d-1)} n$ blocks of $b := \log^{(d-1)} n$ qubits each. Given $i \in [n]$, compute $r \equiv i \pmod b$ into an ancillary register $\mathtt{B}_0$ using a $O(1)$-depth $\operatorname{poly}\log n$-size quantum circuit [HŠ05, SBKH93]. We then copy $|r\rangle_{\mathtt{B}_0}$ a number of $m-1$ times to obtain $\bigotimes_{j=0}^{m-1} |r\rangle_{\mathtt{B}_j}$. For each input $|x_{jb}, x_{jb+1}, \dots, x_{j(b+1)-1}\rangle_{\mathtt{M}} |r\rangle_{\mathtt{B}_j}$, $j \in [m]$, we apply the QRAM circuit from Theorem 28 with target qubit $|0\rangle_{\mathtt{T}_j}$, which yields $|x_{jb+r}\rangle_{\mathtt{T}_j}$. This d-th QRAM-level uses either $O(mb \log b \log\log b) = O(n \log^{(d)} n \log^{(d+1)} n)$ ancillae and $O(mb \log b) = O(n \log^{(d)} n)$ Fan-Outs, or $O(n \log^{(d)} n)$ ancillae and 6 GT gates (the GT gates from different blocks can be done in parallel). We are left with $m = n/\log^{(d-1)} n$ output qubits $\bigotimes_{j=0}^{m-1} |x_{jb+r}\rangle_{\mathtt{T}_j}$. Compute now $q := \lfloor i/b \rfloor$ into a separate quantum register using a $O(1)$-depth $\operatorname{poly}\log n$-size quantum circuit [HŠ05, SBKH93]. Using the induction hypothesis, we can input $|q\rangle \bigotimes_{j=0}^{m-1} |x_{jb+r}\rangle_{\mathtt{T}_j}$ into a $O(d)$-depth QRAM circuit (the remaining $d-1$ QRAM-levels) that uses either $O(m \log^{(d-1)} m \log^{(d)} m) = O(n \log^{(d)} n)$ ancillae and $O(m \log^{(d-1)} m) = O(n)$ Fan-Outs, or $O(n)$ ancillae and $16(d-1)-10$ GT gates. The output qubit is $|b \oplus x_{bq+r}\rangle_{\mathtt{T}} = |b \oplus x_i\rangle_{\mathtt{T}}$ as required. We then uncompute all the intermediary steps.

Let us compute the amount of resources. First note that computing and uncomputing the d-th QRAM-level uses 6 GT gates since we can wait and perform the second half of the QRAM circuits (see Figure 8) until after $|b \oplus x_i\rangle_{\mathtt{T}}$ is outputted. Moreover, copying and uncopying $|r\rangle_{\mathtt{B}_0}$ requires 2 GT gates. Finally, we must take into consideration the resources to compute q and r. For the Fan-Out-based construction, it only requires $\operatorname{poly}\log n$ Fan-Outs. The GT-gate-based construction, on the other hand, requires more care. It is well known that q can be computed using a depth-4 polynomial-size threshold circuit, while r can be computed with a depth-2 polynomial-size threshold circuit [SBKH93]. In order to compute THRESHOLD functions, we employ the Boolean construction from Theorem 36. Due to that, we shall postpone the proof till Section 6.3 (see proof of Theorem 42) and just claim for now that computing (plus uncomputing) q uses $O(n)$ ancillae and 16 GT gates, while computing (plus uncomputing) r uses $O(n)$ ancillae and 8 GT gates. Computing q (16 GTs) can be done in parallel to computing plus copying r and performing the d-th QRAM circuit ($8 + 2 + 6 = 16$ GTs), so the d-th level-QRAM uses 16 GT gates. In total, we use $16d - 10$ GT gates.

The analysis is the same for QRAG, the difference being the number of GT gates. Computing and uncomputing the d-th QRAG circuit from Theorem 29 uses 11 GTs instead of 9, since the GT gates next to the Hadamard layers in Figure 9 must be uncomputed. Computing q (16 GTs) can be done in parallel to computing plus copying r and performing the d-th QRAG circuit ($8+2+11 = 21$ GTs), so the d-th level-QRAG uses 21 GT gates. In total, we use $21d - 12$ GT gates. □

Remark 31. *If $d = \log^* n$, then* QRAM *requires only $O(n)$ ancillae and either $O(n)$ Fan-Outs or $16 \log^* n - 10$ GT gates. Similarly for* QRAG*. However, the depth becomes $O(\log^* n)$, which is no longer constant. Nonetheless, the log-star of the estimated number of atoms in the universe is 5, thus one can consider $\log^* n$ to be a constant in practice.*

6 Constant-depth circuits based on Boolean analysis

In this section, we explore the Boolean analysis connection between constant-depth gates and Fan-Outs made by Takahashi and Tani [TT16] and propose several constructions for f-UCGs. Given $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$, consider its Z-decomposition $\alpha, \beta, \gamma, \delta : \{0,1\}^n \to [-1,1]$. Recall that $\mathrm{supp}(f) := \mathrm{supp}(\alpha) \cup \mathrm{supp}(\beta) \cup \mathrm{supp}(\gamma) \cup \mathrm{supp}(\delta)$ and $\deg(f) := \max\{\deg(\alpha), \deg(\beta), \deg(\gamma), \deg(\delta)\}$. Similar definitions apply to $\mathrm{supp}^{>k}(f)$, $\mathrm{supp}^{\leq k}(f)$, $\mathrm{supp}^{=k}(f)$, $\mathrm{supp}_{\{0,1\}}(f)$, and $\mathrm{supp}^{>k}_{\{0,1\}}(f)$.

6.1 Constant-depth circuits for f-UCGs

Theorem 32 (Real implementation of f-UCG)**.** *Given $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$, there is a $O(1)$-depth circuit for f-UCG that uses*

- *either* $\sum_{S\in\mathrm{supp}(f)} |S| + 2|\mathrm{supp}^{>1}(f)|$ *ancillae and* $2|\bigcup_{S\in\mathrm{supp}^{>1}(f)} S| + 2|\mathrm{supp}^{>1}(f)| + 6$ *Fan-Out gates with arity* $\leq 1 + \max\{|\mathrm{supp}^{>0}(f)|, \deg(f)\}$,
- *or* $|\mathrm{supp}^{>1}(f)|$ *ancillae and* 5 GT *gates with arity* $\leq 2\sum_{S\in\mathrm{supp}(f)} |S|$.

Proof. Consider the initial state $|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}}$ for $x \in \{0,1\}^n$ and $b \in \{0,1\}$. We wish to implement $|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}} f(x)|b\rangle_{\mathtt{T}}$. Let $\alpha, \beta, \gamma, \delta : \{0,1\}^n \to [-1,1]$ be the Z-decomposition of f. Let $\alpha(x) = \sum_{S\subseteq[n]} \widehat{\alpha}(S)\chi_S(x)$ be the Fourier expansion of α, and similarly for β, γ, and δ. For ease of notation, write $m := |\mathrm{supp}^{>0}(f)|$. Let also $m_i := |\{S \in \mathrm{supp}^{>1}(f) : i \in S\}|$ be the number of sets of size greater than 1 that contain the coordinate $i \in [n]$.

Consider first the Fan-Out-based circuit (see Figure 10):

1a. Attach an ancillary register $\bigotimes_{S\in\mathrm{supp}^{>1}(f)} |0\rangle^{\otimes|S|}_{\mathtt{R}_S}$. For each $i \in [n]$ in parallel, copy m_i number of times the qubit $|x_i\rangle_{\mathtt{I}}$ using a $(1+m_i)$-arity Fan-Out gate. This leads to

$$|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \bigotimes_{S\in\mathrm{supp}^{>1}(f)} |x_S\rangle_{\mathtt{R}_S}.$$

1b. Attach an ancillary register $|0\rangle^{\otimes|\mathrm{supp}^{>1}(f)|}_{\mathtt{P}} = \bigotimes_{S\in\mathrm{supp}^{>1}(f)} |0\rangle_{\mathtt{P}_S}$. For each $S \in \mathrm{supp}^{>1}(f)$ in parallel, apply a $\mathsf{PARITY}^{(|S|)}_{\mathtt{R}_S\to\mathtt{P}_S}$ gate using a $(1+|S|)$-arity Fan-Out gate. We obtain the state

$$|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \bigotimes_{S\in\mathrm{supp}^{>1}(f)} |x_S\rangle_{\mathtt{R}_S} \mapsto |x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \bigotimes_{S\in\mathrm{supp}^{>1}(f)} |x_S\rangle_{\mathtt{R}_S} \Big|\bigoplus\nolimits_{i\in S} x_i\Big\rangle_{\mathtt{P}_S}.$$

2a. Attach an ancillary register $|0\rangle^{\otimes(m-1)}_{\mathtt{T}'}$ and apply an m-arity Fan-Out gate $\mathsf{FO}^{(m)}_{\mathtt{T}\to\mathtt{T}'}$ from register $\mathtt{T}$ onto register $\mathtt{T}'$. Apply a $\mathsf{Z}(\delta(0^n))_{\to\mathtt{T}}$ gate onto register $\mathtt{T}$. Notice that $\delta(0^n) = \sum_{S\subseteq[n]} \widehat{\delta}(S)$. Then, for each $S \in \mathrm{supp}^{>0}(\delta)$ in parallel, apply a $\mathsf{Z}(-2\widehat{\delta}(S))$ gate controlled on register $\mathtt{P}_S$ onto the S-th qubit in register $\mathtt{T}'$ (if $|S| = 1$, the gate is controlled on $|x_S\rangle_{\mathtt{I}}$). Finally, apply $\mathsf{FO}^{(m)}_{\mathtt{T}\to\mathtt{T}'}$ again. This chain of operations leads to (omit registers $\mathtt{R}_S$ and $\mathtt{P}_S$ for simplicity)

$$\begin{aligned}|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}}|b\rangle^{\otimes m}_{\mathtt{T},\mathtt{T}'} &\mapsto |x\rangle_{\mathtt{I}}\mathsf{Z}\left(\sum_{S\subseteq[n]} \widehat{\delta}(S)\left(1 - 2\bigoplus_{i\in S} x_i\right)\right)|b\rangle^{\otimes m}_{\mathtt{T},\mathtt{T}'} \\ &= |x\rangle_{\mathtt{I}}\mathsf{Z}\left(\sum_{S\subseteq[n]} \widehat{\delta}(S)\chi_S(x)\right)|b\rangle^{\otimes m}_{\mathtt{T},\mathtt{T}'} \mapsto |x\rangle_{\mathtt{I}}\mathsf{Z}(\delta(x))|b\rangle_{\mathtt{T}}.\end{aligned}$$

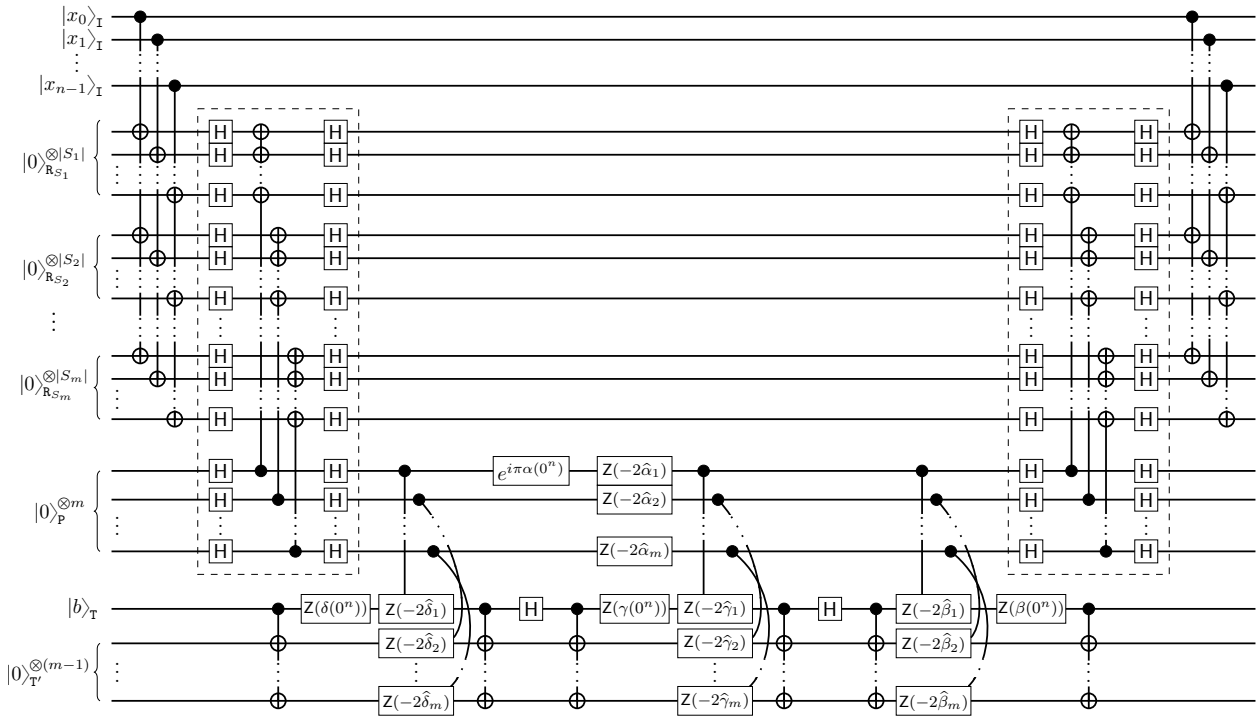


Figure 10: The circuit for an f-UCG$^{(n)}$ using Fan-Out gates. Here $m := |\operatorname{supp}^{>0}(f)|$. We highlight the PARITY operations inside dashed boxes. For simplicity, we write $\widehat{\alpha}(S_j)$ as $\widehat{\alpha}_j$ (and similarly for β, γ, δ). Moreover, we depict $S_1, \dots, S_m \in \operatorname{supp}^{>0}(f)$, but in reality there is no need to compute the parities of sets with size 1 (hence why the register $\mathtt{P}$ is shown with size m). Moreover, we include targets onto all registers $\{\mathtt{R}_S\}_S$ in the Fan-Out gates copying $x_0, \dots, x_{n-1}$, but in reality x_i is copied only onto the registers such that $S \ni i$.

2b. Apply a $\mathsf{H}_{\to\mathtt{T}}$ gate onto register $\mathtt{T}$ followed by an m-arity Fan-Out gate $\mathsf{FO}^{(m)}_{\mathtt{T}\to\mathtt{T}'}$ from register $\mathtt{T}$ onto register $\mathtt{T}'$. Apply a $\mathsf{Z}(\gamma(0^n))_{\to\mathtt{T}}$ gate onto register $\mathtt{T}$. Then, for each $S \in \operatorname{supp}^{>0}(\gamma)$ in parallel, apply a $\mathsf{Z}(-2\widehat{\gamma}(S))$ gate controlled on register $\mathtt{P}_S$ onto the S-th qubit in register $\mathtt{T}'$ (if $|S| = 1$, the gate is controlled on $|x_S\rangle_{\mathtt{I}}$). Finally, apply $\mathsf{FO}^{(m)}_{\mathtt{T}\to\mathtt{T}'}$ again. For simplicity, write $\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}} = r_{b,x}|0\rangle_{\mathtt{T}} + s_{b,x}|1\rangle_{\mathtt{T}}$. This chain of operations leads to

$$\begin{aligned}
|x\rangle_{\mathtt{I}}\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}} &\mapsto |x\rangle_{\mathtt{I}}\left(r_{b,x}|0\rangle^{\otimes m}_{\mathtt{T},\mathtt{T}'} + s_{b,x}|1\rangle^{\otimes m}_{\mathtt{T},\mathtt{T}'}\right) \\
&\mapsto |x\rangle_{\mathtt{I}}\mathsf{Z}\left(\sum_{S\subseteq[n]} \widehat{\gamma}(S)\left(1 - 2\bigoplus_{i\in S} x_i\right)\right)\left(r_{b,x}|0\rangle^{\otimes m}_{\mathtt{T},\mathtt{T}'} + s_{b,x}|1\rangle^{\otimes m}_{\mathtt{T},\mathtt{T}'}\right) \\
&\mapsto |x\rangle_{\mathtt{I}}\mathsf{Z}\left(\sum_{S\subseteq[n]} \widehat{\gamma}(S)\chi_S(x)\right)\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}} \\
&= |x\rangle_{\mathtt{I}}\mathsf{Z}(\gamma(x))\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}}.
\end{aligned}$$

2c. Apply a $\mathsf{H}_{\to\mathtt{T}}$ gate onto register $\mathtt{T}$ followed by an m-arity Fan-Out gate $\mathsf{FO}^{(m)}_{\mathtt{T}\to\mathtt{T}'}$ from register $\mathtt{T}$ onto register $\mathtt{T}'$. Apply a $\mathsf{Z}(\beta(0^n))_{\to\mathtt{T}}$ gate onto register $\mathtt{T}$. Then, for each $S \in \operatorname{supp}^{>0}(\beta)$ in parallel, apply a $\mathsf{Z}(-2\widehat{\beta}(S))$ gate controlled on register $\mathtt{P}_S$ onto the S-th qubit in register $\mathtt{T}'$ (if $|S| = 1$, the gate is controlled on $|x_S\rangle_{\mathtt{I}}$). Finally, apply $\mathsf{FO}^{(m)}_{\mathtt{T}\to\mathtt{T}'}$ again. Similarly to the previous step, this chain of operations leads to

$$|x\rangle_{\mathtt{I}}\mathsf{HZ}(\gamma(x))\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}}\mathsf{Z}(\beta(x))\mathsf{HZ}(\gamma(x))\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}}.$$

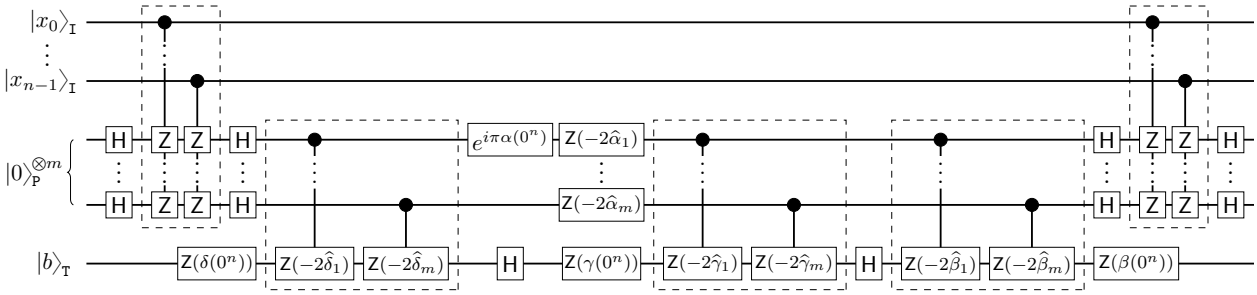


Figure 11: The circuit for an f-UCG$^{(n)}$ using GT gates. Here $m := |\operatorname{supp}^{>0}(f)|$. We highlight the GT gates inside dashed boxes. For simplicity, we write $\widehat{\alpha}(S_j)$ as $\widehat{\alpha}_j$ (and similarly for β, γ, δ). Moreover, we depict $S_1, \ldots, S_m \in \operatorname{supp}^{>0}(f)$, but in reality there is no need to compute the parities of sets with size 1 (hence why the register $\mathtt{P}$ is shown with size m). Moreover, we apply Z gates onto all registers $\{\mathtt{P}_S\}_S$ controlled on $x_0, \ldots, x_{n-1}$ being in the $|1\rangle$ state, but in reality the Z gates controlled on x_i are only applied onto the registers indexed by sets S such that $S \ni i$.

2d. Apply an overall $e^{i\pi\alpha(0^n)}$ phase. For each $S \in \operatorname{supp}^{>0}(\alpha)$ in parallel, apply a $\mathsf{Z}(-2\widehat{\alpha}(S))$ gate onto register $\mathtt{P}_S$ (if $|S| = 1$, apply the gate onto $|x_S\rangle_{\mathtt{I}}$). This leads to

$$|x\rangle_{\mathtt{I}} e^{i\pi\alpha(x)} \mathsf{Z}(\beta(x))\mathsf{H}\mathsf{Z}(\gamma(x))\mathsf{H}\mathsf{Z}(\delta(x))|b\rangle_{\mathtt{T}} = |x\rangle_{\mathtt{I}} f(x)|b\rangle_{\mathtt{T}}.$$

3. Uncompute Step 1.

We now consider the GT-gate-based circuit (see Figure 11), which is basically the same as the Fan-Out-based circuit, the main difference being that it is no longer necessary to copy the register $|x\rangle_{\mathtt{I}}$ several times into $\bigotimes_{S\in\operatorname{supp}^{>1}(f)} \bigotimes_{i\in S} |x_i\rangle_{\mathtt{R}_S}$ as an intermediate step in order to compute the terms $\bigoplus_{i\in S} x_i$ for all $S \in \operatorname{supp}^{>1}(f)$. A single GT gate can compute all the parity terms in parallel according to Lemma 22. In the following, write register $\mathtt{I}$ as $|x\rangle_{\mathtt{I}} = \bigotimes_{i\in[n]} |x_i\rangle_{\mathtt{I}_i}$.

1. Apply Hadamard gates to an ancillary register $|0\rangle_{\mathtt{P}}^{\otimes|\operatorname{supp}^{>1}(f)|} = \bigotimes_{S\in\operatorname{supp}^{>1}(f)} |0\rangle_{\mathtt{P}_S}$. Then, using 1 GT gate with arity $|\bigcup_{S\in\operatorname{supp}^{>1}(f)} S| + \sum_{S\in\operatorname{supp}^{>1}(f)} |S|$, apply, for each $i \in [n]$, a Z gate controlled on $|x_i\rangle_{\mathtt{I}_i}$ to the registers $\mathtt{P}_S$ indexed by the sets $S \in \operatorname{supp}^{>1}(f)$ that contain i. Finally, apply another layer of Hadamard gates to the ancillary register $\mathtt{P}$. We obtain

$$|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \bigotimes_{S\in\operatorname{supp}^{>1}(f)} \Big|\bigoplus\nolimits_{i\in S} x_i\Big\rangle_{\mathtt{P}_S}.$$

2. Apply the gate (write $\mathtt{P}_S := \mathtt{I}_S$ if $|S| = 1$)

$$\left(e^{i\pi\alpha(0^n)} \prod_{S\in\operatorname{supp}^{>0}(\alpha)} \mathsf{Z}(\widehat{\alpha}(-2S))_{\to\mathtt{P}_S}\right)\left(\mathsf{Z}(\beta(0^n))_{\to\mathtt{T}} \prod_{S\in\operatorname{supp}^{>0}(\beta)} \mathsf{C}_{\mathtt{P}_S}\text{-}\mathsf{Z}(\widehat{\beta}(-2S))_{\to\mathtt{T}}\right)\mathsf{H}_{\to\mathtt{T}}$$
$$\cdot\left(\mathsf{Z}(\gamma(0^n))_{\to\mathtt{T}} \prod_{S\in\operatorname{supp}^{>0}(\gamma)} \mathsf{C}_{\mathtt{P}_S}\text{-}\mathsf{Z}(\widehat{\gamma}(-2S))_{\to\mathtt{T}}\right)\mathsf{H}_{\to\mathtt{T}}\left(\mathsf{Z}(\delta(0^n))_{\to\mathtt{T}} \prod_{S\in\operatorname{supp}^{>0}(\delta)} \mathsf{C}_{\mathtt{P}_S}\text{-}\mathsf{Z}(\widehat{\delta}(-2S))_{\to\mathtt{T}}\right)$$

using 3 GT gates (one for each $\prod_{S\in\operatorname{supp}^{>0}(\cdot)} \mathsf{C}_{\mathtt{P}_S}\text{-}\mathsf{Z}(\cdot)_{\to\mathtt{T}}$).

3. Uncompute Step 1.

We now analyse the resources for each step:

- Step 1: in the Fan-Out-based construction we need to copy each x_i, $i \in [n]$, for every[5] $S \in \mathrm{supp}^{>1}(f)$ such that $i \in S$. Registers $\{\mathtt{R}_S\}_S$ thus require $\sum_{i=0}^{n-1} m_i = \sum_{S\in\mathrm{supp}^{>1}(f)} |S|$ ancillae. Such copying can be done with $|\bigcup_{S\in\mathrm{supp}^{>1}(f)} S|$ Fan-Outs with arity $\leq 1+\max_{i\in[n]} m_i$. Moreover, all the parity terms $\bigoplus_{i\in S} x_i$ in registers $\{\mathtt{P}_S\}_S$ for $S \in \mathrm{supp}^{>1}(f)$ require $|\mathrm{supp}^{>1}(f)|$ ancillae and can be computed using either $|\mathrm{supp}^{>1}(f)|$ Fan-Outs with arity $\leq 1+\deg(f)$ or 1 GT gate with arity $|\bigcup_{S\in\mathrm{supp}^{>1}(f)} S| + \sum_{S\in\mathrm{supp}^{>1}(f)} |S| \leq 2\sum_{S\in\mathrm{supp}^{>1}(f)} |S|$;
- Step 2: constructing $f(x)$ requires either $m-1$ ancillae and 6 Fan-Out gates with arity m or no ancillae and 3 GT gates with arity m;
- Step 3: either $|\bigcup_{S\in\mathrm{supp}^{>1}(f)} S| + |\mathrm{supp}^{>1}(f)|$ Fan-Out gates or 1 GT gate.

We use $\sum_{S\in\mathrm{supp}^{>1}(f)} |S| + |\mathrm{supp}^{>0}(f)| + |\mathrm{supp}^{>1}(f)| = \sum_{S\in\mathrm{supp}(f)} |S| + 2|\mathrm{supp}^{>1}(f)|$ ancillae and $2|\bigcup_{S\in\mathrm{supp}^{>1}(f)} S| + 2|\mathrm{supp}^{>1}(f)| + 6$ Fan-Outs with arity $\leq 1+\max\{m, \deg(f)\}$, or $|\mathrm{supp}^{>1}(f)|$ ancillae and 5 GT gates with arity $\leq \max\{m, 2\sum_{S\in\mathrm{supp}^{>1}(f)} |S|\} \leq 2\sum_{S\in\mathrm{supp}(f)} |S|$. □

Instead of exactly simulating an f-$\mathsf{UCG}^{(n)}$ as in the previous result, it is possible to approximate it by a simpler $\mathsf{UCG}^{(n)}$. For such, we can employ real polynomials that equal $\alpha, \beta, \gamma, \delta$ on all inputs up to a small additive error. We shall use the next technical result.

Lemma 33 ([O'D14, Theorem 5.12])**.** *Let $f : \{0,1\}^n \to \mathbb{R}$ be nonzero, $\epsilon > 0$, and $s \geq 4n\hat{\|}f\hat{\|}_1^2/\epsilon^2$ an integer. Then there is a multilinear polynomial $p : \{0,1\}^n \to \mathbb{R}$ of degree and sparsity at most $\deg(f)$ and s, respectively, such that $\max_{x\in\{0,1\}^n} |f(x) - p(x)| < \epsilon$.*

Theorem 34 (Approximate real implementation of f-UCG)**.** *Let $\epsilon > 0$. Given $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$, let $\alpha, \beta, \gamma, \delta : \{0,1\}^n \to [-1,1]$ be its Z-decomposition. For $\nu \in \{\alpha, \beta, \gamma, \delta\}$, define $s_\nu = \min\{\mathrm{supp}^{>1}(\nu), \lceil 64\pi^2 n\hat{\|}\nu^{>1}\hat{\|}_1^2/\epsilon^2\rceil\}$ and $s = s_\alpha + s_\beta + s_\gamma + s_\delta$. There is a $O(1)$-depth circuit that implements an f'-UCG with $f' : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$ such that $\max_{x\in\{0,1\}^n} \|f(x) - f'(x)\| \leq \epsilon$ and uses*

- *either $s(\deg(f)+2) + |\mathrm{supp}^{=1}(f)|$ ancillae and $2s + 2|\bigcup_{S\in\mathrm{supp}^{>1}(f)} S| + 6$ Fan-Out gates with arity $\leq 1 + \max\{s + |\mathrm{supp}^{=1}(f)|, \deg(f)\}$,*
- *or s ancillae and 5 GT gates with arity $\leq 2s\deg(f) + 2|\mathrm{supp}^{=1}(f)|$.*

Proof. Consider the initial state $|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}}$ for $x \in \{0,1\}^n$ and $b \in \{0,1\}$. We wish to perform the operation $|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}} f'(x)|b\rangle_{\mathtt{T}}$ for some $f' : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$ such that $\max_{x\in\{0,1\}^n} \|f(x) - f'(x)\| \leq \epsilon$. Consider, for $\nu \in \{\alpha, \beta, \gamma, \delta\}$, a multilinear polynomial $p_\nu : \{0,1\}^n \to \mathbb{R}$ of degree and sparsity at most $\deg(f)$ and s_ν, respectively, such that $\max_{x\in\{0,1\}^n} |p_\nu(x) - \nu^{>1}(x)| < \sqrt{2\epsilon}/\pi$ according to Lemma 33. Assume without lost of generality that $|\mathrm{supp}(p_\nu)| < |\mathrm{supp}^{>1}(\nu)|$ for $\nu \in \{\alpha, \beta, \gamma, \delta\}$. Our construction is the same as from Theorem 32, the only difference is that we now use p_α in place of $\alpha^{>1}$, so $\alpha(x)$ is replaced with $\alpha^{\leq 1}(x) + p_\alpha(x)$ (and similarly for β, γ, δ). This means that $\mathrm{supp}^{>1}(f)$ is replaced with $\mathrm{supp}(p_\alpha) \cup \mathrm{supp}(p_\beta) \cup \mathrm{supp}(p_\gamma) \cup \mathrm{supp}(p_\delta)$. The number of resources follows from Theorem 32 by replacing

$$
\begin{aligned}
|\mathrm{supp}^{>1}(f)| &\to |\mathrm{supp}(p_\alpha) \cup \mathrm{supp}(p_\beta) \cup \mathrm{supp}(p_\gamma) \cup \mathrm{supp}(p_\delta)|,\\
|\mathrm{supp}^{>0}(f)| &\to |\mathrm{supp}^{=1}(f)| + |\mathrm{supp}(p_\alpha) \cup \mathrm{supp}(p_\beta) \cup \mathrm{supp}(p_\gamma) \cup \mathrm{supp}(p_\delta)|,\\
\Big|\bigcup_{S\in\mathrm{supp}^{>1}(f)} S\Big| &\to \Big|\bigcup_{S\in\mathrm{supp}(p_\alpha)\cup\mathrm{supp}(p_\beta)\cup\mathrm{supp}(p_\gamma)\cup\mathrm{supp}(p_\delta)} S\Big|,\\
\sum_{S\in\mathrm{supp}(f)} |S| &\to |\mathrm{supp}^{=1}(f)| + \sum_{S\in\mathrm{supp}(p_\alpha)\cup\mathrm{supp}(p_\beta)\cup\mathrm{supp}(p_\gamma)\cup\mathrm{supp}(p_\delta)} |S|,
\end{aligned}
$$

[5] We do not need to copy $|x_i\rangle$ for $S \ni i$ if $|S| = 1$ since the state $\big|\bigoplus_{j\in S} x_j\big\rangle$ already equals $|x_i\rangle$.

and bounding

$$\begin{aligned}
&|\operatorname{supp}(p_\alpha) \cup \operatorname{supp}(p_\beta) \cup \operatorname{supp}(p_\gamma) \cup \operatorname{supp}(p_\delta)| \leq s,\\
&\left|\bigcup_{S\in\operatorname{supp}(p_\alpha)\cup\operatorname{supp}(p_\beta)\cup\operatorname{supp}(p_\gamma)\cup\operatorname{supp}(p_\delta)} S\right| \leq \left|\bigcup_{S\in\operatorname{supp}^{>1}(f)} S\right|,\\
&\sum_{S\in\operatorname{supp}(p_\alpha)\cup\operatorname{supp}(p_\beta)\cup\operatorname{supp}(p_\gamma)\cup\operatorname{supp}(p_\delta)} |S| \leq s\deg(f).
\end{aligned}$$

To show correctness of the circuit, define $\overline{p}_\alpha := \alpha^{\leq 1} + p_\alpha$ for simplicity (and similarly for $\overline{p}_\beta$, $\overline{p}_\gamma$, $\overline{p}_\delta$). Then (omit the x dependence for clarity)

$$\begin{aligned}
&\|e^{i\pi\alpha}\mathsf{Z}(\beta)\mathsf{HZ}(\gamma)\mathsf{HZ}(\delta) - e^{i\pi\overline{p}_\alpha}\mathsf{Z}(\overline{p}_\beta)\mathsf{HZ}(\overline{p}_\gamma)\mathsf{HZ}(\overline{p}_\delta)\|\\
&\leq \|(e^{i\pi\alpha} - e^{i\pi\overline{p}_\alpha})\mathsf{Z}(\beta)\mathsf{HZ}(\gamma)\mathsf{HZ}(\delta)\| + \|e^{i\pi\overline{p}_\alpha}\mathsf{Z}(\beta)\mathsf{HZ}(\gamma)\mathsf{HZ}(\delta) - e^{i\pi\overline{p}_\alpha}\mathsf{Z}(\overline{p}_\beta)\mathsf{HZ}(\overline{p}_\gamma)\mathsf{HZ}(\overline{p}_\delta)\|\\
&= 2|\sin(\pi(\alpha-\overline{p}_\alpha)/2)| + \|\mathsf{Z}(\beta)\mathsf{HZ}(\gamma)\mathsf{HZ}(\delta) - \mathsf{Z}(\overline{p}_\beta)\mathsf{HZ}(\overline{p}_\gamma)\mathsf{HZ}(\overline{p}_\delta)\|\\
&\leq 2|\sin(\pi(\alpha-\overline{p}_\alpha)/2)| + \|(\mathsf{Z}(\beta) - \mathsf{Z}(\overline{p}_\beta))\mathsf{HZ}(\gamma)\mathsf{HZ}(\delta)\| + \|\mathsf{Z}(\overline{p}_\beta)\mathsf{H}(\mathsf{Z}(\gamma)\mathsf{HZ}(\delta) - \mathsf{Z}(\overline{p}_\gamma)\mathsf{HZ}(\overline{p}_\delta))\|\\
&= 2|\sin(\pi(\alpha-\overline{p}_\alpha)/2)| + 2|\sin(\pi(\beta-\overline{p}_\beta)/2)| + \|\mathsf{Z}(\gamma)\mathsf{HZ}(\delta) - \mathsf{Z}(\overline{p}_\gamma)\mathsf{HZ}(\overline{p}_\delta)\|\\
&\leq 2|\sin(\pi(\alpha-\overline{p}_\alpha)/2)| + 2|\sin(\pi(\beta-\overline{p}_\beta)/2)| + \|(\mathsf{Z}(\gamma) - \mathsf{Z}(\overline{p}_\gamma))\mathsf{HZ}(\delta)\| + \|\mathsf{Z}(\overline{p}_\gamma)\mathsf{H}(\mathsf{Z}(\delta) - \mathsf{Z}(\overline{p}_\delta))\|\\
&= 2|\sin(\pi(\alpha-\overline{p}_\alpha)/2)| + 2|\sin(\pi(\beta-\overline{p}_\beta)/2)| + 2|\sin(\pi(\gamma-\overline{p}_\gamma)/2)| + 2|\sin(\pi(\delta-\overline{p}_\delta)/2)|\\
&\leq 8|\sin(\epsilon/8)| \leq \epsilon.
\end{aligned}$$

□

Our final construction uses the real-polynomial $\{0,1\}$-representation based on AND functions which can be computed using Fact 18 or Fact 24.

Theorem 35 (Real $\{0,1\}$-implementation of f-UCG)**.** *Given $f : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$, there is a $O(1)$-depth circuit for f-UCG that uses*

- *either $\sum_{S\in\operatorname{supp}^{>0}_{\{0,1\}}(f)} \left(2|S|\log|S| + O(|S|)\right)$ ancillae and $\sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} \left(8|S| + O(\log|S|)\right)$ Fan-Out gates with arity $\leq \max\{1 + |\operatorname{supp}^{>0}_{\{0,1\}}(f)|, 2\deg(f)\}$,*
- *or $\sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} \left(3|S| + O(\log|S|)\right)$ ancillae and 9 GT gates with arity $\leq 2\sum_{S\in\operatorname{supp}_{\{0,1\}}(f)} |S|$.*

Proof. Consider the initial state $|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}}$ for $x \in \{0,1\}^n$ and $b \in \{0,1\}$. We wish to implement $|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}} f(x)|b\rangle_{\mathtt{T}}$. Let $\alpha(x) = \sum_{S\subseteq[n]} \widetilde{\alpha}(S)x^S$ be the real-polynomial $\{0,1\}$-representation of α, and similarly for β,γ,δ. Write $m := |\operatorname{supp}^{>0}_{\{0,1\}}(f)|$ and $m_i := |\{S \in \operatorname{supp}^{>1}_{\{0,1\}}(f) : i \in S\}|$ for the number of sets of size greater than 1 that contain the coordinate $i \in [n]$.

Consider first the Fan-Out-based circuit:

1. Attach an ancillary register $\bigotimes_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} |0\rangle^{\otimes|S|}_{\mathtt{R}_S}$. For each $i \in [n]$ in parallel, copy m_i number of times the qubit $|x_i\rangle_{\mathtt{I}}$ by using a $(1+m_i)$-arity Fan-Out to obtain
$$|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \bigotimes_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} |x_S\rangle_{\mathtt{R}_S}.$$
2. Attach an ancillary register $\bigotimes_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} |0\rangle_{\mathtt{P}_S}$. For each $S \in \operatorname{supp}^{>1}_{\{0,1\}}(f)$ in parallel, apply an $\mathsf{AND}^{(|S|)}_{\mathtt{R}_S\to\mathtt{P}_S}$ gate to get
$$|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \bigotimes_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} |x_S\rangle_{\mathtt{R}_S} \mapsto |x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \bigotimes_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} |x_S\rangle_{\mathtt{R}_S}|x^S\rangle_{\mathtt{P}_S}.$$

3a. Attach an ancillary register $|0\rangle_{\mathtt{T}'}^{\otimes(m-1)}$ and apply an m-arity Fan-Out gate $\mathsf{FO}^{(m)}_{\mathtt{T}\to\mathtt{T}'}$ from register $\mathtt{T}$ onto register $\mathtt{T}'$. Apply a $\mathsf{Z}(\widetilde{\delta}(\emptyset))_{\to\mathtt{T}}$ gate onto register $\mathtt{T}$. Then, for each $S \in \mathrm{supp}^{>0}_{\{0,1\}}(\delta)$ in parallel, apply a $\mathsf{Z}(\widetilde{\delta}(S))$ gate controlled on register $\mathtt{P}_S$ onto the S-th qubit in register $\mathtt{T}'$ (if $|S|=1$, apply the gate onto $|x_S\rangle_{\mathtt{I}}$). Finally, apply $\mathsf{FO}^{(m)}_{\mathtt{T}\to\mathtt{T}'}$ again. This chain of operations leads to (omit registers $\mathtt{P}_S$ and $\mathtt{R}_S$ for simplicity)

$$|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}}|b\rangle^{\otimes m}_{\mathtt{T},\mathtt{T}'} \mapsto |x\rangle_{\mathtt{I}}\mathsf{Z}\left(\sum_{S\subseteq[n]}\widetilde{\delta}(S)x^S\right)|b\rangle^{\otimes m}_{\mathtt{T},\mathtt{T}'} \mapsto |x\rangle_{\mathtt{I}}\mathsf{Z}(\delta(x))|b\rangle_{\mathtt{T}}.$$

3b. Apply a $\mathsf{H}_{\to\mathtt{T}}$ gate onto register $\mathtt{T}$ followed by an m-arity Fan-Out gate $\mathsf{FO}^{(m)}_{\mathtt{T}\to\mathtt{T}'}$ from register $\mathtt{T}$ onto register $\mathtt{T}'$. Apply a $\mathsf{Z}(\widetilde{\gamma}(\emptyset))_{\to\mathtt{T}}$ gate onto register $\mathtt{T}$. Then, for each $S \in \mathrm{supp}^{>0}_{\{0,1\}}(\gamma)$ in parallel, apply a $\mathsf{Z}(\widetilde{\gamma}(S))$ gate controlled on register $\mathtt{P}_S$ onto the S-th qubit in register $\mathtt{T}'$ (if $|S|=1$, apply the gate onto $|x_S\rangle_{\mathtt{I}}$). Finally, apply $\mathsf{FO}^{(m)}_{\mathtt{T}\to\mathtt{T}'}$ again. We obtain

$$|x\rangle_{\mathtt{I}}\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}}\mathsf{Z}\left(\sum_{S\subseteq[n]}\widetilde{\gamma}(S)x^S\right)\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}} = |x\rangle_{\mathtt{I}}\mathsf{Z}(\gamma(x))\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}}.$$

3c. Apply a $\mathsf{H}_{\to\mathtt{T}}$ gate onto register $\mathtt{T}$ followed by an m-arity Fan-Out gate $\mathsf{FO}^{(m)}_{\mathtt{T}\to\mathtt{T}'}$ from register $\mathtt{T}$ onto register $\mathtt{T}'$. Apply a $\mathsf{Z}(\widetilde{\beta}(\emptyset))_{\to\mathtt{T}}$ gate onto register $\mathtt{T}$. Then, for each $S \in \mathrm{supp}^{>0}_{\{0,1\}}(\beta)$ in parallel, apply a $\mathsf{Z}(\widetilde{\beta}(S))$ gate controlled on register $\mathtt{P}_S$ onto the S-th qubit in register $\mathtt{T}'$ (if $|S|=1$, apply the gate onto $|x_S\rangle_{\mathtt{I}}$). Finally, apply $\mathsf{FO}^{(m)}_{\mathtt{T}\to\mathtt{T}'}$ again. Similarly to the previous steps, this chain of operations leads to

$$|x\rangle_{\mathtt{I}}\mathsf{HZ}(\gamma(x))\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}} \mapsto |x\rangle_{\mathtt{I}}\mathsf{Z}(\beta(x))\mathsf{HZ}(\gamma(x))\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}}.$$

3d. Apply an overall phase $e^{i\pi\widetilde{\alpha}(\emptyset)}$. Then, for each $S \in \mathrm{supp}^{>0}_{\{0,1\}}(\alpha)$ in parallel, apply a $\mathsf{Z}(\widetilde{\alpha}(S))$ gate onto register $\mathtt{P}_S$ (if $|S|=1$, apply the gate onto $|x_S\rangle_{\mathtt{I}}$). This yields

$$|x\rangle_{\mathtt{I}}e^{i\pi\alpha(x)}\mathsf{Z}(\beta(x))\mathsf{HZ}(\gamma(x))\mathsf{HZ}(\delta(x))|b\rangle_{\mathtt{T}} = |x\rangle_{\mathtt{I}}f(x)|b\rangle_{\mathtt{T}}.$$

4. Uncompute Steps 1 and 2.

We now consider the GT-gate-based circuit. Steps 3a-d are replaced with the following Step 3. In the following, write register $\mathtt{I}$ as $|x\rangle_{\mathtt{I}} = \bigotimes_{i\in[n]}|x_i\rangle_{\mathtt{I}_i}$.

3. Apply the gate (write $\mathtt{P}_S := \mathtt{I}_S$ if $|S|=1$ and $\mathtt{P}_\emptyset := \emptyset$)

$$\left(e^{i\pi\widetilde{\alpha}(\emptyset)}\prod_{S\in\mathrm{supp}^{>0}_{\{0,1\}}(\alpha)}\mathsf{Z}(\widetilde{\alpha}(S))_{\to\mathtt{P}_S}\right)\left(\prod_{S\in\mathrm{supp}_{\{0,1\}}(\beta)}\mathsf{C}_{\mathtt{P}_S}\text{-}\mathsf{Z}(\widetilde{\beta}(S))_{\to\mathtt{T}}\right)\mathsf{H}_{\to\mathtt{T}}$$
$$\cdot\left(\prod_{S\in\mathrm{supp}_{\{0,1\}}(\gamma)}\mathsf{C}_{\mathtt{P}_S}\text{-}\mathsf{Z}(\widetilde{\gamma}(S))_{\to\mathtt{T}}\right)\mathsf{H}_{\to\mathtt{T}}\left(\prod_{S\in\mathrm{supp}_{\{0,1\}}(\delta)}\mathsf{C}_{\mathtt{P}_S}\text{-}\mathsf{Z}(\widetilde{\delta}(S))_{\to\mathtt{T}}\right)$$

using 3 GT gates (one for each $\prod_{S\in\mathrm{supp}_{\{0,1\}}(\cdot)}\mathsf{C}_{\mathtt{P}_S}\text{-}\mathsf{Z}(\cdot)_{\to\mathtt{T}}$).

We now analyse the resources for each step:

- Step 1: we need to copy each x_i, $i \in [n]$, for every $S \in \operatorname{supp}^{>1}_{\{0,1\}}(f)$ such that $i \in S$. Thus registers $\{\mathtt{R}_S\}_S$ require $\sum_{i\in[n]} m_i = \sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} |S|$ ancillae. Such copying can be done with either $|\bigcup_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} S| \leq \sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} |S|$ Fan-Outs, each with arity at most $1+\max_{i\in[n]} m_i$, or 1 GT gate with arity $|\bigcup_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} S| + \sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} |S| \leq 2\sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} |S|$;

- Step 2: the $|\operatorname{supp}^{>1}_{\{0,1\}}(f)|$ $\mathsf{AND}^{(|S|)}_{\mathtt{R}_S\to\mathtt{P}_S}$ gates can be performed with either

  $$\sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} (2|S|\log|S| + O(|S|)) \text{ ancillae} \quad \text{and} \quad \sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} (3|S| + O(\log|S|)) \text{ Fan-Outs}$$

  with arity at most $2\deg(f)$ (Fact 18), or

  $$\sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} (2|S| + O(\log|S|)) \text{ ancillae} \quad \text{and} \quad 2 \text{ GT gates}$$

  with arity at most $2\deg(f)+O(\log\deg(f))$ (Fact 24) if we postpone their inner uncomputation part until Step 5;

- Step 3: constructing $f(x)$ requires either $m-1$ ancillae and 6 Fan-Out gates with arity m or no ancillae and 3 GT gates with arity m;

- Step 4: uncomputing Steps 1 and 2 requires either $\sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} (4|S|+O(\log|S|))$ Fan-Out gates or 3 GT gates.

We require either $\sum_{S\in\operatorname{supp}^{>0}_{\{0,1\}}(f)} (2|S|\log|S|+O(|S|))$ ancillae and $\sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} (8|S|+O(\log|S|))$ Fan-Out gates with arity $\leq \max\{1+|\operatorname{supp}^{>0}_{\{0,1\}}(f)|, 2\deg(f)\}$ or $\sum_{S\in\operatorname{supp}^{>1}_{\{0,1\}}(f)} (3|S|+O(\log|S|))$ ancillae and 9 GT gates with arity $\leq 2\sum_{S\in\operatorname{supp}_{\{0,1\}}(f)} |S|$. □

6.2 Constant-depth circuits for f-FINs

Similarly to Section 5.2, we now show how the circuits from the previous section can be simplified and used to implement f-FINs.

Theorem 36 (Real implementation of f-FIN)**.** *Given $f : \{0,1\}^n \to \{0,1\}$, there is a $O(1)$-depth circuit for f-FIN that uses*

- *either $\sum_{S\in\operatorname{supp}(f)} |S|+2|\operatorname{supp}^{>1}(f)|$ ancillae and $2|\bigcup_{S\in\operatorname{supp}^{>1}(f)} S|+2|\operatorname{supp}^{>1}(f)|+2$ Fan-Out gates with arity $\leq 1+\max\{|\operatorname{supp}^{>0}(f)|, \deg(f)\}$,*

- *or $2|\operatorname{supp}^{>0}(f)|$ ancillae and 2 GT gates with arity $\leq 3\sum_{S\in\operatorname{supp}(f)} |S|$.*

Proof. Since an f-FIN is simply an f'-UCG whose f' Z-decomposition is $\mathsf{HZ}(f(x))\mathsf{H}$, Step 2 in Theorem 32 only requires 2 Fan-Out gates or 1 GT gate. This gives the resource count for the Fan-Out-based construction and the number of GT gates is reduced to 3. It is possible to further reduce the number of GT gates to 2 by using the $(|\operatorname{supp}^{>0}(f)|-1)$-qubit register $\mathtt{T}'$ and, similarly to the Fan-Out-based construction, to use the parallelisation method depicted in Figure 7 (see proof of Theorem 27). This increases the number of ancillae to $|\operatorname{supp}^{>0}(f)| + |\operatorname{supp}^{>1}(f)| \leq 2|\operatorname{supp}^{>0}(f)|$. The new GT gates' arity is $|\bigcup_{S\in\operatorname{supp}^{>1}(f)} S| + \sum_{S\in\operatorname{supp}^{>1}(f)} |S| + |\operatorname{supp}^{>0}(f)| \leq 3\sum_{S\in\operatorname{supp}(f)} |S|$. □

To obtain the next result on an approximate implementation of f-FINs, the modifications to be made to Theorem 34 are the same that were conducted in the previous theorem. Recall that an f-FIN is an f'-UCG such that $f'(x) = \mathsf{X}^{f(x)}$. Therefore, an approximate circuit for an f-FIN implements an f'-UCG with $f'(x)$ close to $\mathbb{I}_1$ or X.

Theorem 37 (Approximate real implementation of f-FIN)**.** *Let $\epsilon > 0$, $f : \{0,1\}^n \to \{0,1\}$, and $s = \min\{\mathrm{supp}^{>1}(f), \lceil 4\pi^2 n \hat{\|} f^{>1} \hat{\|}_1^2 / \epsilon^2 \rceil\}$. There is a $O(1)$-depth circuit that implements an f'-UCG with $f' : \{0,1\}^n \to \mathcal{U}(\mathbb{C}^{2\times 2})$ such that $\max_{x\in\{0,1\}^n} \|f'(x) - \mathsf{X}^{f(x)}\| \leq \epsilon$ and uses*

- *either $s(\deg(f)+2) + |\,\mathrm{supp}^{=1}(f)|$ ancillae and $2s+2|\bigcup_{S\in\mathrm{supp}^{>1}(f)} S| + 2$ Fan-Out gates with arity $\leq s + |\,\mathrm{supp}^{=1}(f)|$,*
- *or $2s + |\,\mathrm{supp}^{=1}(f)|$ ancillae and 2 GT gates with arity $\leq s(2\deg(f)+1) + |\,\mathrm{supp}^{=1}(f)|$.*

Finally, Theorem 35 can be considerably simplified in the case of f-FINs since we can use the $\mathbb{F}_2$-polynomial representation of $f(x)$ instead of its real $\{0,1\}$-representation.

Theorem 38 ($\mathbb{F}_2$-implementation of f-FIN)**.** *Given $f : \{0,1\}^n \to \{0,1\}$, there is a $O(1)$-depth circuit for f-FIN that uses*

- *either $\sum_{S\in\mathrm{supp}^{\geq 1}_{\mathbb{F}_2}(f)} \left(2|S|\log|S| + O(|S|)\right)$ ancillae and $\sum_{S\in\mathrm{supp}^{\geq 1}_{\mathbb{F}_2}(f)} \left(8|S| + O(\log|S|)\right)$ Fan-Out gates with arity $\leq \max\{1 + |\mathrm{supp}^{>0}_{\mathbb{F}_2}(f)|, 2\deg_{\mathbb{F}_2}(f)\}$,*
- *or $\sum_{S\in\mathrm{supp}^{>0}_{\mathbb{F}_2}(f)} \left(4|S| + O(\log|S|)\right)$ ancillae and 6 GT gates with arity $\leq 3\sum_{S\in\mathrm{supp}_{\mathbb{F}_2}(f)} |S|$.*

Proof. After constructing the state

$$|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \bigotimes_{S\in\mathrm{supp}^{\geq 1}_{\mathbb{F}_2}(f)} |x_S\rangle_{\mathtt{R}_S}|x^S\rangle_{\mathtt{P}_S}$$

as in Theorem 35, apply a $\mathsf{X}^{\widetilde{f}_{\mathbb{F}_2}(\emptyset)}_{\to\mathtt{T}}$ gate onto register $\mathtt{T}$, a $\mathsf{PARITY}_{\{\mathtt{I}_j\}_{j\in\mathrm{supp}^{=1}_{\mathbb{F}_2}(f)}\to\mathtt{T}}$ gate from registers $\{\mathtt{I}_j\}_{j\in\mathrm{supp}^{=1}_{\mathbb{F}_2}(f)}$ onto register $\mathtt{T}$ ($\mathtt{I}_j$ contains x_j), and finally a $\mathsf{PARITY}_{\{\mathtt{P}_S\}_{S\in\mathrm{supp}^{\geq 1}_{\mathbb{F}_2}(f)}\to\mathtt{T}}$ gate from registers $\{\mathtt{P}_S\}_{S\in\mathrm{supp}^{\geq 1}_{\mathbb{F}_2}(f)}$ onto register $\mathtt{T}$ (both PARITY gates can be performed together). We get

$$\begin{aligned}
|x\rangle_{\mathtt{I}}|b\rangle_{\mathtt{T}} \bigotimes_{S\in\mathrm{supp}^{\geq 1}_{\mathbb{F}_2}(f)} |x_S\rangle_{\mathtt{R}_S}|x^S\rangle_{\mathtt{P}_S} &\mapsto |x\rangle_{\mathtt{I}}|b \oplus \widetilde{f}_{\mathbb{F}_2}(\emptyset) \oplus \bigoplus\nolimits_{S\in\mathrm{supp}^{\geq 1}_{\mathbb{F}_2}(f)} x^S\rangle_{\mathtt{T}} \bigotimes_{S\in\mathrm{supp}^{\geq 1}_{\mathbb{F}_2}(f)} |x_S\rangle_{\mathtt{R}_S}|x^S\rangle_{\mathtt{P}_S} \\
&= |x\rangle_{\mathtt{I}}|b \oplus \bigoplus\nolimits_{S\subseteq[n]} \widetilde{f}_{\mathbb{F}_2}(S)x^S\rangle_{\mathtt{T}} \bigotimes_{S\in\mathrm{supp}^{\geq 1}_{\mathbb{F}_2}(f)} |x_S\rangle_{\mathtt{R}_S}|x^S\rangle_{\mathtt{P}_S} \\
&= |x\rangle_{\mathtt{I}}|b \oplus f(x)\rangle_{\mathtt{T}} \bigotimes_{S\in\mathrm{supp}^{\geq 1}_{\mathbb{F}_2}(f)} |x_S\rangle_{\mathtt{R}_S}|x^S\rangle_{\mathtt{P}_S}.
\end{aligned}$$

Uncomputing registers $\{\mathtt{R}_S\}_S$ and $\{\mathtt{P}_S\}_S$ gives the desired state. As in Theorem 35, the total cost of computing and uncomputing $\{\mathtt{R}_S\}_S$ and $\{\mathtt{P}_S\}_S$ is either

$$\sum_{S\in\mathrm{supp}^{\geq 1}_{\mathbb{F}_2}(f)} \left(2|S|\log|S| + O(|S|)\right) \text{ ancillae} \quad \text{and} \quad \sum_{S\in\mathrm{supp}^{\geq 1}_{\mathbb{F}_2}(f)} \left(8|S| + O(\log|S|)\right) \text{ Fan-Outs}$$

with arity $\leq \max\{1+|\mathrm{supp}^{\geq 0}_{\mathbb{F}_2}(f)|, 2\deg_{\mathbb{F}_2}(f)\}$ or

$$\sum_{S\in\mathrm{supp}^{\geq 1}_{\mathbb{F}_2}(f)} (3|S|+O(\log|S|)) \text{ ancillae} \quad \text{and} \quad 6 \ \mathsf{GT} \text{ gates}$$

with arity $\leq 2\sum_{S\in\mathrm{supp}_{\mathbb{F}_2}(f)}|S|$. The PARITY gates cost another $(1+|\mathrm{supp}^{\geq 0}_{\mathbb{F}_2}(f)|)$-arity Fan-Out or GT gate. It is possible to reduce the number of GT gates from 7 to 6 by using $|\mathrm{supp}^{\geq 0}_{\mathbb{F}_2}(f)|$ extra ancillae similarly to Theorem 36. □

6.3 Constant-depth circuits for quantum memory devices via Boolean analysis

In this section, we apply our Boolean-based circuit constructions to the case of QRAM. We first compute the Fourier coefficients of $f(x,i)=x_i$.

Lemma 39. *Let $n\in\mathbb{N}$ be a power of 2 and let $f:\{0,1\}^n\times\{0,1\}^{\log n}\to\{0,1\}$ be the Boolean function $f(x,i)=x_i$. The Fourier coefficients of f are*

$$\widehat{f}(S,T)=\begin{cases}\frac{1}{2} & \text{if } (S,T)=(\emptyset,\emptyset),\\ \frac{-\chi_T(k)}{2n} & \forall (S,T)\subseteq[n]\times[\log n], S=\{k\},\\ 0 & \text{otherwise,}\end{cases}$$

where $\chi_T(k)=(-1)^{\sum_{i\in T}k_i}$ and $k=k_0\ldots k_{\log n-1}\in\{0,1\}^{\log n}$.

Proof. By a straightforward calculation,

$$\widehat{f}(\emptyset,T)=\frac{1}{2^n n}\sum_{i\in\{0,1\}^{\log n}}\chi_T(i)\sum_{x\in\{0,1\}^n}x_i=\frac{1}{2n}\sum_{i\in\{0,1\}^{\log n}}\chi_T(i)=\begin{cases}\frac{1}{2} & \text{if } T=\emptyset,\\ 0 & \text{otherwise.}\end{cases}$$

Moreover,

$$\widehat{f}(S,T)=\frac{1}{2^n n}\sum_{i\in\{0,1\}^{\log n}}\chi_T(i)\sum_{x\in\{0,1\}^n}x_i(-1)^{\sum_{j\in S}x_j}=\begin{cases}\frac{-\chi_T(k)}{2n} & \text{if } S=\{k\},\\ 0 & \text{if } |S|\geq 2,\end{cases}$$

for every $T\subseteq[\log n]$, since $\sum_{x\in\{0,1\}^n}x_i(-1)^{x_j}$ equals 0 if $i\neq j$ and -2^{n-1} if $i=j$. □

Theorem 40 (Real implementation of QRAM)**.** *Let $n\in\mathbb{N}$ be a power of 2. A* QRAM *of memory size n can be implemented in $O(1)$-depth using*

- *either $\frac{1}{2}n^2\log n+O(n^2)$ ancillae and $2n^2+O(n\log n)$ Fan-Out gates with arity $\leq 1+n^2$,*
- *or $2n^2$ ancillae and 2* GT *gates with arity $\leq\frac{1}{2}n^2\log n+O(n^2)$.*

Proof. By Lemma 39, $\mathrm{supp}(f)=\{(S,T)\subseteq[n]\times[\log n]:|S|=1\}\cup\{(\emptyset,\emptyset)\}$. Thus $|\mathrm{supp}^{>0}(f)|=n^2$, $|\mathrm{supp}^{>1}(f)|=n^2-n$, $|\bigcup_{(S,T)\in\mathrm{supp}^{>1}(f)}(S,T)|=n+\log n$, $\deg(f)=1+\log n$, and

$$\sum_{(S,T)\in\mathrm{supp}(f)}|S|+|T|=n\sum_{k=0}^{\log n}\binom{\log n}{k}(1+k)=n^2+\frac{n^2\log n}{2}.$$

By Theorem 36, there is a $O(1)$-depth circuit for QRAM that uses either $\frac{1}{2}n^2\log n+O(n^2)$ ancillae and $2n^2+O(n\log n)$ Fan-Out gates with arity at most $1+n^2$, or $2n^2$ ancillae and 2 GT gates with arity at most $\frac{1}{2}n^2\log n+O(n^2)$. □

Remark 41. *Since the $\mathbb{F}_2$-support of $f : \{0,1\}^n \times \{0,1\}^{\log n} \to \{0,1\}$, $f(x,i) = x_i$, is quite dense, Theorem 38 is not suited for constructing an efficient* QRAM*. Moreover,*

$$\hat{\|} f^{>1} \hat{\|}_1 = \sum_{(S,T)\in \mathrm{supp}^{>1}(f)} |\widehat{f}(S,T)| = \frac{1}{2n} n(2^{\log n} - 1) = \frac{n-1}{2}.$$

Therefore, $s = \lceil 4\pi^2 n \hat{\|} f^{>1} \hat{\|}_1^2 / \epsilon^2 \rceil = \lceil \pi^2 n^3 / \epsilon^2 \rceil - O(n^2)$ and the approximate real constructions in Theorem 37 require many more ancillae compared to Theorem 40.

Similarly to the one-hot encoding construction, it is possible to reduce the number of ancillae by reducing the problem into small blocks and solving each with a small QRAM circuit.

Theorem 42. *For every $n, d \in \mathbb{N}$, a* QRAM *of memory size n can be implemented in $O(d)$-depth using*

- *either $O\big(n^{1/(1-2^{-d})} \log n\big)$ ancillae and $O\big(n^{1/(1-2^{-d})}\big)$ Fan-Out gates,*
- *or $O\big(n^{1/(1-2^{-d})}\big)$ ancillae and $8d-6$* GT *gates.*

Proof. The proof is very similar to Theorem 30, only the size of the blocks is different. For $d = 1$, the result follows from Theorem 40. For the induction step, we divide the input $x \in \{0,1\}^n$ into $m := n^{(2^d-2)/(2^d-1)}$ blocks of $b := n^{1/(2^d-1)}$ qubits each. The d-th QRAM circuit thus uses either $O(mb^2 \log b) = O\big(n^{1/(1-2^{-d})} \log n\big)$ ancillae and $O(mb^2) = O\big(n^{1/(1-2^{-d})}\big)$ Fan-Outs, or $O\big(n^{1/(1-2^{-d})}\big)$ ancillae and 2 GT gates (the GT gates from different blocks can be done in parallel). We are left with $m = n^{(2^d-2)/(2^d-1)}$ output qubits. Using the induction hypothesis, the remaining $d-1$ QRAM-levels uses either $O\big(m^{1/(1-2^{-d+1})} \log m\big) = O\big(n^{1/(1-2^{-d})} \log n\big)$ ancillae and $O\big(m^{1/(1-2^{-d+1})}\big) = O\big(n^{1/(1-2^{-d})}\big)$ Fan-Outs, or $O\big(n^{1/(1-2^{-d})}\big)$ ancillae and $8(d-1)-6$ GT gates. The output qubit is $|b \oplus x_{bq+r}\rangle_{\mathsf{T}} = |b \oplus x_i\rangle_{\mathsf{T}}$ as required.

Regarding the resources, we must take into consideration the computation of the quotient $q := \lfloor i/b \rfloor$ and remainder $r \equiv i \pmod b$ (and copying/uncopying the remainder $m-1$ times). As mentioned in the proof of Theorem 30, q can be computed with a depth-4 polynomial-size threshold circuit and r can be computed with a depth-2 polynomial-size threshold circuit [SBKH93]. In the Fan-Out-based construction, $\mathrm{poly}\log n$ ancillae and Fan-Out gates are sufficient since i is a $\log n$-bit string. Regarding the GT-gate-based construction, we employ Theorem 36 to perform a THRESHOLD function using $O(|\mathrm{supp}^{>0}(\mathsf{THRESHOLD}^{(\log n)})|) = O(n)$ ancillae and 2 GT gates. Each layer of the threshold circuit is made of $\mathrm{poly}\log n$ THRESHOLD functions. It is then possible to either

1. compute all the PARITY functions x_S, $S \in \mathrm{supp}^{>0}(\mathsf{THRESHOLD}^{(\log n)})$, a number of $\mathrm{poly}\log n$ times in parallel, so that each THRESHOLD function has its own set $\{x_S\}_S$, and this requires $O(n\,\mathrm{poly}\log n)$ ancillae and 1 GT gate (plus 1 GT gate for uncomputation). Therefore, computing q uses $O(n\,\mathrm{poly}\log n)$ ancillae and 8 GT gates (2 per layer), and computing r uses $O(n\,\mathrm{poly}\log n)$ ancillae and 4 GT gates;

2. or compute the PARITY functions $\{x_S\}_S$ just once and share them with all THRESHOLD functions of a layer. This means that, in order to apply the gates $\mathsf{Z}(\widehat{f}(S))$ controlled on x_S (cf. Figure 11), we require another GT gate since x_S will serve as control-qubit for all THRESHOLD functions (plus 1 GT for uncomputation). This requires $O(n)$ ancillae and 4 GT gates. Computing q uses $O(n)$ ancillae and 16 GT gates (4 per layer), and computing r uses $O(n)$ ancillae and 8 GT gates.

For the one-hot-based circuit (Theorem 30), we employ the second construction since it only uses $O(n)$ ancillae. Here, for the Boolean-based circuit, we employ the first construction as we already used $O(n^{1/(1-2^{-d})})$ ancillae. We note that computing q (8 GT gates) can be done in parallel to computing plus copying/uncopying r and performing the d-th QRAM circuit ($4 + 2 + 2 = 8$ GT gates), so the d-th level-QRAM uses 8 GT gates. In total, we use $8d - 6$ GT gates. □

7 Acknowledgement

We thank Rainer Dumke, Yvonne Gao, Wenhui Li, and Daniel Weiss for useful discussions on physical implementations of QRAM, Arthur Rattew for interesting conversations on the feasibility of QRAMs, Patrick Rebentrost for initial discussions, Sathyawageeswar Subramanian for Ref. [Kum23], and Shengyu Zhang for general discussions and for clarifying some points in [STY+23, YZ23]. This research is supported by the National Research Foundation, Singapore and A*STAR under its CQT Bridging Grant and its Quantum Engineering Programme under grant NRF2021-QEP2-02-P05. This work was done in part while JFD, AL, and MS were visiting the Simons Institute for the Theory of Computing.

References

[Aar05] Scott Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 461(2063):3473–3482, 2005. doi:10.1098/rspa.2005.1546. 10

[AC17] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. In *32nd Computational Complexity Conference (CCC 2017)*, volume 79 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:67, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2017.22. 10

[ACL+20] Scott Aaronson, Nai-Hui Chia, Han-Hsuan Lin, Chunhao Wang, and Ruizhe Zhang. On the Quantum Complexity of Closest Pair and Related Problems. In Shubhangi Saraf, editor, *35th Computational Complexity Conference (CCC 2020)*, volume 169 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:43, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2020.16. 2

[ADOY24] Anurag Anshu, Yangjing Dong, Fengning Ou, and Penghui Yao. On the computational power of QAC0 with barely superlinear ancillae. *arXiv preprint arXiv:2410.06499*, 2024. doi:10.48550/arXiv.2410.06499. 10

[AGJO+15] Srinivasan Arunachalam, Vlad Gheorghiu, Tomas Jochym-O'Connor, Michele Mosca, and Priyaa Varshinee Srinivasan. On the robustness of bucket brigade quantum RAM. *New Journal of Physics*, 17(12):123010, Dec 2015. doi:10.1088/1367-2630/17/12/123010. 16

[Amb07] Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007. doi:10.1137/S0097539705447311. 2

[AP22] Gabriele Agliardi and Enrico Prati. Optimized quantum generative adversarial networks for distribution loading. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 824–827, 2022. doi:10.1109/QCE53715.2022.00132. 16

[APPdS21] Israel F. Araujo, Daniel K. Park, Francesco Petruccione, and Adenilton J. da Silva. A divide-and-conquer algorithm for quantum state preparation. *Scientific Reports*, 11(1):6329, Mar 2021. doi:10.1038/s41598-021-85474-1. 10

[BARdW08] Avraham Ben-Aroya, Oded Regev, and Ronald de Wolf. A hypercontractive inequality for matrix-valued functions with applications to quantum computing and LDCs. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 477–486, 2008. doi:10.1109/FOCS.2008.45. 6

[Bau22] Johannes Bausch. Fast Black-Box Quantum State Preparation. *Quantum*, 6:773, August 2022. doi:10.22331/q-2022-08-04-773. 10

[BBC+95] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. Elementary gates for quantum computation. *Phys. Rev. A*, 52:3457–3467, Nov 1995. doi:10.1103/PhysRevA.52.3457. 3, 9, 17, 18

[BBCSN24] Jop Briët, Harry Buhrman, Davi Castro-Silva, and Niels M. P. Neumann. Noisy Decoding by Shallow Circuits with Parities: Classical and Quantum (Extended Abstract). In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:11, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2024.21. 10

[BBG+13] Robert Beals, Stephen Brierley, Oliver Gray, Aram W. Harrow, Samuel Kutin, Noah Linden, Dan Shepherd, and Mark Stather. Efficient distributed quantum computing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 469(2153):20120686, 2013. doi:10.1098/rspa.2012.0686. 13

[BBS00] Howard Barnum, Herbert J Bernstein, and Lee Spector. Quantum circuits for OR and AND of ORs. *Journal of Physics A: Mathematical and General*, 33(45):8047, Nov 2000. doi:10.1088/0305-4470/33/45/304. 3

[BFLN23] Harry Buhrman, Marten Folkertsma, Bruno Loff, and Niels M. P. Neumann. State preparation by shallow circuits using feed forward. *arXiv preprint arXiv:2307.14840*, 2023. doi:10.48550/arXiv.2307.14840. 10

[BFNV19] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. "Quantum Supremacy" and the Complexity of Random Circuit Sampling. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:2, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2019.15. 10

[BGB+18] Ryan Babbush, Craig Gidney, Dominic W. Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven. Encoding electronic spectra in quantum circuits with linear T complexity. *Phys. Rev. X*, 8:041015, Oct 2018. doi:10.1103/PhysRevX.8.041015. 2

[BGK18] Sergey Bravyi, David Gosset, and Robert König. Quantum advantage with shallow circuits. *Science*, 362(6412):308–311, 2018. doi:10.1126/science.aar3106. 10

[BGKT20] Sergey Bravyi, David Gosset, Robert König, and Marco Tomamichel. Quantum advantage with noisy shallow circuits. *Nature Physics*, 16(10):1040–1045, Oct 2020. doi:10.1038/s41567-020-0948-z. 10

[BHT98] Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In Cláudio L. Lucchesi and Arnaldo V. Moura, editors, *LATIN'98: Theoretical Informatics*, pages 163–169, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg. doi:10.1007/BFb0054319. 2

[BIS+18] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595–600, Jun 2018. doi:10.1038/s41567-018-0124-x. 10

[BLPS22a] Harry Buhrman, Bruno Loff, Subhasree Patro, and Florian Speelman. Limits of Quantum Speed-Ups for Computational Geometry and Other Problems: Fine-Grained Complexity via Quantum Walks. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:12, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2022.31. 2

[BLPS22b] Harry Buhrman, Bruno Loff, Subhasree Patro, and Florian Speelman. Memory Compression with Quantum Random-Access Gates. In François Le Gall and Tomoyuki Morimae, editors, *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, volume 232 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:19, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.TQC.2022.10. 15

[BM04] Stephen S. Bullock and Igor L. Markov. Asymptotically optimal circuits for arbitrary n-qubit diagonal comutations. *Quantum Info. Comput.*, 4(1):27–47, January 2004. doi:10.26421/QIC4.1-3. 17

[BMN22] Sergey Bravyi, Dmitri Maslov, and Yunseong Nam. Constant-cost implementations of clifford operations and multiply-controlled gates using global interactions. *Phys. Rev. Lett.*, 129:230501, Nov 2022. doi:10.1103/PhysRevLett.129.230501. 4, 9, 21, 22

[BVMS05] Ville Bergholm, Juha J. Vartiainen, Mikko Möttönen, and Martti M. Salomaa. Quantum circuits with uniformly controlled one-qubit gates. *Phys. Rev. A*, 71:052330, May 2005. doi:10.1103/PhysRevA.71.052330. 10

[BZC+23] Pascal Baßler, Matthias Zipper, Christopher Cedzich, Markus Heinrich, Patrick H. Huber, Michael Johanning, and Martin Kliesch. Synthesis of and compilation with time-optimal multi-qubit gates. *Quantum*, 7:984, April 2023. doi:10.22331/q-2023-04-20-984. 4, 9

[CB18] John A. Cortese and Timothy M. Braje. Loading classical data into a quantum computer. *arXiv preprint arXiv:1803.01958*, 2018. doi:10.48550/arXiv.1803.01958. 10

[CCRK23] Libor Caha, Xavier Coiteux-Roy, and Robert Koenig. A colossal advantage: 3D-local noisy shallow quantum circuits defeat unbounded fan-in classical circuits. *arXiv preprint arXiv:2312.09209*, 2023. doi:10.48550/arXiv.2312.09209. 10

[CDEH+21] K. C. Chen, W. Dai, C. Errando-Herranz, S. Lloyd, and D. Englund. Scalable and high-fidelity quantum random access memory in spin-photon networks. *PRX Quantum*, 2:030319, Aug 2021. doi:10.1103/PRXQuantum.2.030319. 16

[CL21] André Chailloux and Johanna Loyer. Lattice sieving via quantum random walks. In Mehdi Tibouchi and Huaxiong Wang, editors, *Advances in Cryptology – ASIACRYPT 2021*, pages 63–91, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-92068-5_3. 2

[CSV21] Matthew Coudron, Jalex Stark, and Thomas Vidick. Trading locality for time: Certifiable randomness from low-depth circuits. *Communications in Mathematical Physics*, 382(1):49–86, Feb 2021. doi:10.1007/s00220-021-03963-w. 10

[DM22] Austin K. Daniel and Akimasa Miyake. Quantum algorithms for classical Boolean

functions via adaptive measurements: Exponential reductions in space-time resources. *arXiv preprint arXiv:2211.01252*, 2022. doi:10.48550/arXiv.2211.01252. 3

[Fen03] Stephen A. Fenner. Implementing the fanout gate by a Hamiltonian. *arXiv preprint quant-ph/0309163*, 2003. doi:10.48550/arXiv.quant-ph/0309163. 3

[FMMC12] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, 86:032324, Sep 2012. doi:10.1103/PhysRevA.86.032324. 22

[FOL+19] C. Figgatt, A. Ostrander, N. M. Linke, K. A. Landsman, D. Zhu, D. Maslov, and C. Monroe. Parallel entangling operations on a universal ion-trap quantum computer. *Nature*, 572(7769):368–372, Aug 2019. doi:10.1038/s41586-019-1427-5. 3

[FS08] Serge Fehr and Christian Schaffner. Randomness extraction via δ-biased masking in the presence of a quantum attacker. In Ran Canetti, editor, *Theory of Cryptography*, pages 465–481, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. doi:10.1007/978-3-540-78524-8_26. 6

[FW23] Stephen Fenner and Rabins Wosti. Implementing the fanout operation with simple pairwise interactions. *Quantum Information and Computation*, 23(13&14):1081–1090, 2023. doi:10.26421/QIC23.13-14-1. 3

[GBW+20] Nikodem Grzesiak, Reinhold Blümel, Kenneth Wright, Kristin M. Beck, Neal C. Pisenti, Ming Li, Vandiver Chaplin, Jason M. Amini, Shantanu Debnath, Jwo-Sy Chen, and Yunseong Nam. Efficient arbitrary simultaneously entangling gates on a trapped-ion quantum computer. *Nature Communications*, 11(1):2963, Jun 2020. doi:10.1038/s41467-020-16790-9. 3, 22

[GDC+22] Andrew Y. Guo, Abhinav Deshpande, Su-Kuan Chu, Zachary Eldredge, Przemyslaw Bienias, Dhruv Devulapalli, Yuan Su, Andrew M. Childs, and Alexey V. Gorshkov. Implementing a fast unbounded quantum fanout gate using power-law interactions. *Phys. Rev. Res.*, 4:L042016, Oct 2022. doi:10.1103/PhysRevResearch.4.L042016. 3

[GFPV+21] Xiu Gu, Jorge Fernández-Pendás, Pontus Vikstål, Tahereh Abad, Christopher Warren, Andreas Bengtsson, Giovanna Tancredi, Vitaly Shumeiko, Jonas Bylander, Göran Johansson, and Anton Frisk Kockum. Fast multiqubit gates through simultaneous two-qubit gates. *PRX Quantum*, 2:040348, Dec 2021. doi:10.1103/PRXQuantum.2.040348. 3

[GHMP02] Frederic Green, Steven Homer, Cristopher Moore, and Christopher Pollett. Counting, fanout and the complexity of quantum ACC. *Quantum Info. Comput.*, 2(1):35–65, Dec 2002. doi:10.26421/QIC2.1-3. 3, 4, 8, 9, 19, 30

[GK24] Sabee Grewal and Vinayak M Kumar. Improved circuit lower bounds with applications to exponential separations between quantum and classical circuits. *arXiv preprint arXiv:2408.16406*, 2024. doi:10.48550/arXiv.2408.16406. 10

[GKH+21] Pranav Gokhale, Samantha Koretsky, Shilin Huang, Swarnadeep Majumder, Andrew Drucker, Kenneth R. Brown, and Frederic T. Chong. Quantum fan-out: Circuit optimizations and technology modeling. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 276–290, 2021. doi:10.1109/QCE52317.2021.00045. 3

[GKMdO24] Alex Bredariol Grilo, Elham Kashefi, Damian Markham, and Michael de Oliveira. The power of shallow-depth Toffoli and qudit quantum circuits. *arXiv preprint arXiv:2404.18104*, 2024. doi:10.48550/arXiv.2404.18104. 10

[GLM08a] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Architectures for a quantum

random access memory. *Phys. Rev. A*, 78:052310, Nov 2008. doi:10.1103/PhysRevA.78.052310. 2, 16

[GLM08b] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, Apr 2008. doi:10.1103/PhysRevLett.100.160501. 2, 16

[GM24] Daniel Grier and Jackson Morris. Quantum threshold is powerful. *arXiv preprint arXiv:2411.04953*, 2024. doi:10.48550/arXiv.2411.04953. 19

[GMNN22] Nikodem Grzesiak, Andrii Maksymov, Pradeep Niroula, and Yunseong Nam. Efficient quantum programming using EASE gates on a trapped-ion quantum computer. *Quantum*, 6:634, January 2022. doi:10.22331/q-2022-01-27-634. 4, 9

[GR02] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002. doi:10.48550/arXiv.quant-ph/0208112. 2, 3, 10, 17

[Gro97] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79:325–328, Jul 1997. doi:10.1103/PhysRevLett.79.325. 2

[Gro00] Lov K. Grover. Synthesis of quantum superpositions by quantum computation. *Phys. Rev. Lett.*, 85:1334–1337, Aug 2000. doi:10.1103/PhysRevLett.85.1334. 10

[GS20] Daniel Grier and Luke Schaeffer. Interactive shallow Clifford circuits: Quantum advantage against NC^1 and beyond. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 875–888, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384332. 10

[GWSG20] Koen Groenland, Freek Witteveen, Kareljan Schoutens, and Rene Gerritsma. Signal processing techniques for efficient compilation of controlled rotations in trapped ions. *New Journal of Physics*, 22(6):063006, Jun 2020. doi:10.1088/1367-2630/ab8830. 4, 9

[Han21] Connor T. Hann. *Practicality of Quantum Random Access Memory*. PhD thesis, Yale University, 2021. URL: https://www.proquest.com/dissertations-theses/practicality-quantum-random-access-memory/docview/2631670801/se-2. 16

[HHL09] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.*, 103:150502, Oct 2009. doi:10.1103/PhysRevLett.103.150502. 3, 17

[HLGJ21] Connor T. Hann, Gideon Lee, S.M. Girvin, and Liang Jiang. Resilience of quantum random access memory to generic noise. *PRX Quantum*, 2:020311, Apr 2021. doi:10.1103/PRXQuantum.2.020311. 16

[HMdOS24] Min-Hsiu Hsieh, Leandro Mendes, Michael de Oliveira, and Sathyawageeswar Subramanian. Unconditionally separating noisy QNC^0 from bounded polynomial threshold circuits of constant depth. *arXiv preprint arXiv:2408.16378*, 2024. doi:10.48550/arXiv.2408.16378. 10

[HŠ05] Peter Høyer and Robert Špalek. Quantum fan-out is powerful. *Theory of Computing*, 1(5):81–103, 2005. doi:10.4086/toc.2005.v001a005. 3, 4, 5, 6, 9, 19, 21, 30, 33, 34

[IIV15] Svetoslav S. Ivanov, Peter A. Ivanov, and Nikolay V. Vitanov. Efficient construction of three- and four-qubit quantum gates by global entangling gates. *Phys. Rev. A*, 91:032311, Mar 2015. doi:10.1103/PhysRevA.91.032311. 4

[JR23] Samuel Jaques and Arthur G. Rattew. QRAM: A survey and critique. *arXiv preprint arXiv:2305.10310*, 2023. doi:10.48550/arXiv.2305.10310. 16

[KLP20] Iordanis Kerenidis, Alessandro Luongo, and Anupam Prakash. Quantum expectation-maximization for Gaussian mixture models. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of

Proceedings of Machine Learning Research, pages 5187–5197. PMLR, 13–18 Jul 2020. URL: https://proceedings.mlr.press/v119/kerenidis20a.html. 2

[KMN+22] Yosep Kim, Alexis Morvan, Long B. Nguyen, Ravi K. Naik, Christian Jünger, Larry Chen, John Mark Kreikebaum, David I. Santiago, and Irfan Siddiqi. High-fidelity three-qubit *i*Toffoli gate for fixed-frequency superconducting qubits. *Nature Physics*, 18(7):783–788, Jul 2022. doi:10.1038/s41567-022-01590-3. 3

[Kni95] Emanuel Knill. Approximation by quantum circuits. *arXiv preprint quant-ph/9508006*, 1995. doi:10.48550/arXiv.quant-ph/9508006. 3, 9

[KP17] Iordanis Kerenidis and Anupam Prakash. Quantum Recommendation Systems. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 49:1–49:21, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2017.49. 2, 3, 17

[Kum23] Vinayak M. Kumar. Tight Correlation Bounds for Circuits Between AC0 and TC0. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference (CCC 2023)*, volume 264 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:40, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2023.18. 12, 45

[Kup05] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005. doi:10.1137/S0097539703436345. 2

[LG19] François Le Gall. Average-Case Quantum Advantage with Shallow Circuits. In Amir Shpilka, editor, *34th Computational Complexity Conference (CCC 2019)*, volume 137 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:20, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2019.21. 10

[LKS24] Guang Hao Low, Vadym Kliuchnikov, and Luke Schaeffer. Trading T gates for dirty qubits in state preparation and unitary synthesis. *Quantum*, 8:1375, June 2024. doi:10.22331/q-2024-06-17-1375. 2, 3, 4, 9, 23

[LWL+19] K. A. Landsman, Y. Wu, P. H. Leung, D. Zhu, N. M. Linke, K. R. Brown, L. Duan, and C. Monroe. Two-qubit entangling gates within arbitrarily long chains of trapped ions. *Phys. Rev. A*, 100:022332, Aug 2019. doi:10.1103/PhysRevA.100.022332. 3

[MGB22] Sam McArdle, András Gilyén, and Mario Berta. Quantum state preparation without coherent arithmetic. *arXiv preprint arXiv:2210.14892*, 2022. doi:10.48550/arXiv.2210.14892. 10

[MGM20] Olivia Di Matteo, Vlad Gheorghiu, and Michele Mosca. Fault-tolerant resource estimation of quantum random-access memories. *IEEE Transactions on Quantum Engineering*, 1:1–13, 2020. doi:10.1109/TQE.2020.2965803. 16

[MMN+16] Esteban A Martinez, Thomas Monz, Daniel Nigg, Philipp Schindler, and Rainer Blatt. Compiling quantum algorithms for architectures with multi-qubit gates. *New Journal of Physics*, 18(6):063029, Jun 2016. doi:10.1088/1367-2630/18/6/063029. 3, 4

[MN98] Cristopher Moore and Martin Nilsson. Some notes on parallel quantum computation. *arXiv preprint quant-ph/9804034*, 1998. doi:10.48550/arXiv.quant-ph/9804034. 8

[MN01] Cristopher Moore and Martin Nilsson. Parallel quantum computation and quantum codes. *SIAM Journal on Computing*, 31(3):799–815, 2001. doi:10.1137/S0097539799355053. 4, 8, 30

[MN18] Dmitri Maslov and Yunseong Nam. Use of global interactions in efficient quantum

circuit constructions. *New Journal of Physics*, 20(3):033018, Mar 2018. doi:10.1088/1367-2630/aaa398. 4, 9

[MO10] Ashley Montanaro and Tobias J. Osborne. Quantum Boolean functions. *Chicago Journal of Theoretical Computer Science*, 2010(1), January 2010. doi:10.4086/cjtcs.2010.001. 6

[Mon15] Ashley Montanaro. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181):20150301, 2015. doi:10.1098/rspa.2015.0301. 3, 17

[Moo99] Cristopher Moore. Quantum circuits: Fanout, parity, and counting. *arXiv preprint quant-ph/9903046*, 1999. doi:10.48550/arXiv.quant-ph/9903046. 3, 9, 19, 29

[MP01] Alan Mishchenko and Marek Perkowski. Fast heuristic minimization of exclusive-sums-of-products. In *RM'2001 Workshop*, 2001. URL: https://pdxscholar.library.pdx.edu/ece_fac/195/. 16

[MS99] Klaus Mølmer and Anders Sørensen. Multiparticle entanglement of hot trapped ions. *Phys. Rev. Lett.*, 82:1835–1838, Mar 1999. doi:10.1103/PhysRevLett.82.1835. 3

[MV05] M. Mottonen and Juha J. Vartiainen. Decompositions of general quantum gates. *arXiv preprint quant-ph/0504100*, 2005. doi:10.48550/arXiv.quant-ph/0504100. 3, 9

[MZ22] Dmitri Maslov and Ben Zindorf. Depth optimization of CZ, CNOT, and Clifford circuits. *IEEE Transactions on Quantum Engineering*, 3:1–8, 2022. doi:10.1109/TQE.2022.3180900. 4, 9

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010. doi:10.1017/CBO9780511976667. 10

[NMM+14] D. Nigg, M. Müller, E. A. Martinez, P. Schindler, M. Hennrich, T. Monz, M. A. Martin-Delgado, and R. Blatt. Quantum computations on a topologically encoded qubit. *Science*, 345(6194):302–305, 2014. doi:10.1126/science.1253742. 3

[NPVY24] Shivam Nadimpalli, Natalie Parham, Francisca Vasconcelos, and Henry Yuen. On the Pauli Spectrum of QAC0. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 1498–1506, New York, NY, USA, 2024. Association for Computing Machinery. doi:10.1145/3618260.3649662. 10

[NV00] Ashwin Nayak and Ashvin Vishwanath. Quantum walk on the line. *arXiv preprint quant-ph/0010117*, 2000. doi:10.48550/arXiv.quant-ph/0010117. 6

[NZB+22] Murphy Yuezhen Niu, Alexander Zlokapa, Michael Broughton, Sergio Boixo, Masoud Mohseni, Vadim Smelyanskyi, and Hartmut Neven. Entangling quantum generative adversarial networks. *Phys. Rev. Lett.*, 128:220505, Jun 2022. doi:10.1103/PhysRevLett.128.220505. 16

[O'D14] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. doi:10.1017/CBO9781139814782. 11, 38

[PB11] Martin Plesch and Časlav Brukner. Quantum-state preparation with universal gate decompositions. *Phys. Rev. A*, 83:032302, Mar 2011. doi:10.1103/PhysRevA.83.032302. 10

[PCG23] Koustubh Phalak, Avimita Chatterjee, and Swaroop Ghosh. Quantum random access memory for dummies. *Sensors*, 23(17), 2023. doi:10.3390/s23177462. 16

[PFGT20] Daniel Padé, Stephen Fenner, Daniel Grier, and Thomas Thierauf. Depth-2 QAC circuits cannot simulate quantum parity. *arXiv preprint arXiv:2005.12169*, 2020. doi:10.48550/arXiv.2005.12169. 10

[Piu15] Einar Pius. *Parallel quantum computing: from theory to practice*. PhD thesis, The University of Edinburgh, 2015. URL: http://hdl.handle.net/1842/15857. 10

[PLG22] Koustubh Phalak, Junde Li, and Swaroop Ghosh. Approximate quantum random access memory architectures. *arXiv preprint arXiv:2210.14804*, 2022. doi:10.48550/arXiv.2210.14804. 16

[PPR19] Daniel K. Park, Francesco Petruccione, and June-Koo Kevin Rhee. Circuit-based quantum random access memory for classical data. *Scientific Reports*, 9(1):3949, Mar 2019. doi:10.1038/s41598-019-40439-3. 16

[PS13] Paul Pham and Krysta M. Svore. A 2D nearest-neighbor quantum architecture for factoring in polylogarithmic depth. *Quantum Info. Comput.*, 13(11&12):937–962, Nov 2013. doi:10.26421/QIC13.11-12-3. 3, 20

[RGG+20] S. E. Rasmussen, K. Groenland, R. Gerritsma, K. Schoutens, and N. T. Zinner. Single-step implementation of high-fidelity n-bit Toffoli gates. *Phys. Rev. A*, 101:022308, Feb 2020. doi:10.1103/PhysRevA.101.022308. 3

[RK22] Arthur G. Rattew and Bálint Koczor. Preparing arbitrary continuous functions in quantum registers with logarithmic complexity. *arXiv preprint arXiv:2205.00519*, 2022. doi:10.48550/arXiv.2205.00519. 10

[Ros21a] Gregory Rosenthal. Bounds on the QAC^0 Complexity of Approximating Parity. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2021.32. 10

[Ros21b] Gregory Rosenthal. Query and depth upper bounds for quantum unitaries via Grover search. *arXiv preprint arXiv:2111.07992*, 2021. doi:10.48550/arXiv.2111.07992. 3, 5, 9, 10, 22, 30

[Ros24] Gregory Rosenthal. Efficient quantum state synthesis with one query. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2508–2534, 2024. doi:10.1137/1.9781611977912.89. 10

[RWZ24] Cambyse Rouzé, Melchior Wirth, and Haonan Zhang. Quantum Talagrand, KKL and Friedgut's Theorems and the Learnability of Quantum Boolean Functions. *Communications in Mathematical Physics*, 405(4):95, Apr 2024. doi:10.1007/s00220-024-04981-0. 6

[SBKH93] K.-Y. Siu, J. Bruck, T. Kailath, and T. Hofmeister. Depth efficient neural networks for division and related problems. *IEEE Transactions on Information Theory*, 39(3):946–956, 1993. doi:10.1109/18.256501. 34, 44

[Sch03] Ralf Schützhold. Pattern recognition on a quantum computer. *Phys. Rev. A*, 67:062311, Jun 2003. doi:10.1103/PhysRevA.67.062311. 2

[SLSB19] Yuval R. Sanders, Guang Hao Low, Artur Scherer, and Dominic W. Berry. Black-box quantum state preparation without arithmetic. *Phys. Rev. Lett.*, 122:020502, Jan 2019. doi:10.1103/PhysRevLett.122.020502. 10

[SMB04] Vivek V. Shende, Igor L. Markov, and Stephen S. Bullock. Minimal universal two-qubit controlled-not-based circuits. *Phys. Rev. A*, 69:062321, Jun 2004. doi:10.1103/PhysRevA.69.062321. 3

[Smo87] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, page 77–82, New York, NY, USA, 1987. Association for Computing Machinery. doi:10.1145/28395.28404. 9

[SS06] Gernot Schaller and Ralf Schützhold. Quantum algorithm for optical-template recognition with noise filtering. *Phys. Rev. A*, 74:012303, Jul 2006. doi:10.1103/PhysRevA.74.012303. 2

[SSP13] Alireza Shafaei, Mehdi Saeedi, and Massoud Pedram. Reversible logic synthesis of k-input, m-output lookup tables. In *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1235–1240, 2013. doi:10.7873/DATE.2013.256. 16

[STY+23] Xiaoming Sun, Guojing Tian, Shuai Yang, Pei Yuan, and Shengyu Zhang. Asymptotically optimal circuit depth for quantum state preparation and general unitary synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(10):3301–3314, 2023. doi:10.1109/TCAD.2023.3244885. 3, 9, 10, 17, 45

[TD04] Barbara M. Terhal and David P. DiVincenzo. Adaptive quantum computation, constant depth quantum circuits and Arthur-Merlin games. *Quantum Info. Comput.*, 4(2):134–145, Mar 2004. doi:10.26421/QIC4.2-5. 10

[Tru02] C. A. Trugenberger. Phase transitions in quantum pattern recognition. *Phys. Rev. Lett.*, 89:277903, Dec 2002. doi:10.1103/PhysRevLett.89.277903. 2

[TT16] Yasuhiro Takahashi and Seiichiro Tani. Collapse of the hierarchy of constant-depth exact quantum circuits. *computational complexity*, 25(4):849–881, Dec 2016. doi:10.1007/s00037-016-0140-0. 3, 4, 5, 9, 19, 21, 33, 35

[VMS04] Juha J. Vartiainen, Mikko Möttönen, and Martti M. Salomaa. Efficient decomposition of quantum gates. *Phys. Rev. Lett.*, 92:177902, Apr 2004. doi:10.1103/PhysRevLett.92.177902. 3, 9

[vdW21] John van de Wetering. Constructing quantum circuits with global gates. *New Journal of Physics*, 23(4):043015, Apr 2021. doi:10.1088/1367-2630/abf1b3. 4, 9

[WKST19] Adam Bene Watts, Robin Kothari, Luke Schaeffer, and Avishay Tal. Exponential separation between shallow quantum circuits and unbounded fan-in shallow classical circuits. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 515–526, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3313276.3316404. 10

[dW08] Ronald de Wolf. *A Brief Introduction to Fourier Analysis on the Boolean Cube*. Number 1 in Graduate Surveys. Theory of Computing Library, 2008. doi:10.4086/toc.gs.2008.001. 11

[WP23] Adam Bene Watts and Natalie Parham. Unconditional quantum advantage for sampling with shallow circuits. *arXiv preprint arXiv:2301.00995*, 2023. doi:10.48550/arXiv.2301.00995. 10

[YGZ+20] Dongmin Yu, Yichun Gao, Weiping Zhang, Jinming Liu, and Jing Qian. Scalability and high-efficiency of an $(n+1)$-qubit Toffoli gate sphere via blockaded Rydberg atoms. *arXiv preprint arXiv:2001.04599*, 2020. doi:10.48550/arXiv.2001.04599. 3

[YZ23] Pei Yuan and Shengyu Zhang. Optimal (controlled) quantum state preparation and improved unitary synthesis by quantum circuits with any number of ancillary qubits. *Quantum*, 7:956, March 2023. doi:10.22331/q-2023-03-20-956. 3, 9, 45

[ZLW19] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Information*, 5(1):103, Nov 2019. doi:10.1038/s41534-019-0223-2. 16

[ZLY22] Xiao-Ming Zhang, Tongyang Li, and Xiao Yuan. Quantum state preparation with optimal circuit depth: Implementations and applications. *Phys. Rev. Lett.*, 129:230504, Nov 2022. doi:10.1103/PhysRevLett.129.230504. 9, 10

[ZYY21] Xiao-Ming Zhang, Man-Hong Yung, and Xiao Yuan. Low-depth quantum state preparation. *Phys. Rev. Res.*, 3:043200, Dec 2021. doi:10.1103/PhysRevResearch.3.043200. 10

[ZZY05] B. Zeng, D. L. Zhou, and L. You. Measuring the parity of an n-qubit state. *Phys. Rev. Lett.*, 95:110502, Sep 2005. doi:10.1103/PhysRevLett.95.110502. 3