Supplementary Material for Paper "Statistical Testing of Quantum Programs via Fixed-Point Amplitude Amplification" in OOPSLA 2024

CHAN GU KANG[*], Korea University, South Korea
JOONGHOON LEE, Korea University, South Korea
HAKJOO OH[†], Korea University, South Korea

A IMPLEMENTATION OF FPAA

In this section, we describe details on the implementation of FPAA [Yoder et al. 2014]. While much of this section reformulates results from [Yoder et al. 2014], we have included it in the Supplementary Material for the sake of completeness of the paper..

Derivation of Parameters α, β. As stated in Proposition 5.1, the desired amplifier is realized by a l-sequence of the generalized Grover operator $G(\alpha_j, \beta_j)$ for $j = 1, .., l$. Given l and δ, each parameter α_j and β_i is derived as :

$$\alpha_j = -\beta_{l-j+1} = 2\cot^{-1}\left(\tan(2\pi j/L)\sqrt{1-(T_{1/L}(1/\delta))^{-2}}\right),$$

where T_x denotes the Chebyshev polynomial (for the proof, refer to Yoder et al. [2014]).

Constructing Reflection over $|E_\perp\rangle$. Given a target state $|t\rangle$ and a source state $|s\rangle$, FPAA amplifies $|\langle t|s\rangle|^2$, the amplitude of $|t\rangle$ within $|s\rangle$. The implementation of FPAA includes a quantum circuit for reflection over $|t\rangle$, denoted as S_t and presented in (6). Note that our original goal is to amplify $|E_\perp\rangle$, where $|t\rangle = |E_\perp\rangle$ and $|t_\perp\rangle = |E\rangle$. Hence we need to implement reflection over $|E_\perp\rangle$ as follows:

$$S_t(\beta) = S_{E_\perp}(\beta) = I - (1 - e^{i\beta})|E_\perp\rangle\langle E_\perp|.$$

However, while $|E\rangle$ is known from user-provided specification, $|E_\perp\rangle$ may not readily available (as we do not know the exact $|P\rangle$, we cannot calculate $|E_\perp\rangle$). Consequently, we cannot directly implement a quantum circuit for $S_{E_\perp}$. Yet, we know $|E\rangle$ from user-provided specification and hence reflection over $|E\rangle$, $S_E(\beta') = I - (1 - e^{i\beta'})|E\rangle\langle E|$ is directly implementable. Then, we can obtain $S_{E_\perp}(\beta)$ by following relation:

$$S_{E_\perp}(\beta) = S_E(-\beta) \text{ upto global phase.} \tag{26}$$

Therefore, in the implementation details of S_t provided below, we instead describe the implementation for $S_E(\beta)$.

Additional Unitary Transformation on P. We can always set $|E\rangle = |0\rangle$ by assuming the unitary operator V such that $V|E\rangle = |0\rangle$. If such V is obtained, then we can represent the program state as follows:

$$VP|I\rangle = V|P\rangle = \sqrt{a}|0\rangle + \sqrt{b}e^{i\theta}|0_\perp\rangle,$$

where $|0_\perp\rangle = V|E_\perp\rangle$. The unitary transformation V may be provided manually by the user or generated by a state preparation algorithm. For example, if $|E\rangle = |0010\rangle$, the user could provide

[*]Current affiliation : Furiosa AI
[†]Corresponding author

Authors' addresses: Chan Gu Kang, Korea University, Seoul, South Korea, changukang@korea.ac.kr; Joonghoon Lee, Korea University, Seoul, South Korea, joonghoonlee@korea.ac.kr; Hakjoo Oh, Korea University, Seoul, South Korea, hakjoo_oh@korea.ac.kr.

$V = I \otimes I \otimes X \otimes I$[1]. Introducing V ensures that the oracle required for $S_E(\beta)$ (discussed below) is fixed and implementable.

We note that the additional requirement of the unitary operator V also arises in prior work. For instance, Li et al. [2020] required such a unitary due to physical constraints that projective measurements can only be realized in the computational basis.

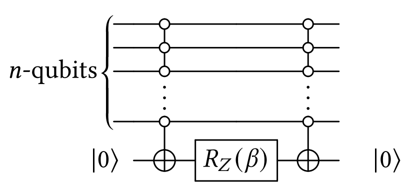
Furthermore, we assume the input preparation unitary operator W such that $W|0\rangle = |I\rangle$ exists. Thereby, we can always assume that the program state $|P\rangle$ is prepared from the input $|0\rangle$ as $|P\rangle = PW|0\rangle$. Providing W will not be an additional overhead for users, since real-world quantum hardware initializes the qubit register state to $|0\rangle$ by default. Thus, users need to prepare such a unitary W, in anyway.

To summarize, whenever the provided states $|E\rangle$ and $|I\rangle$ are not equal to $|0\rangle$, we assume that the additional unitary operator V and W exist. Therefore, we consider the quantum circuit program P to be transformed as follows:

$$P \leftarrow VPW.$$

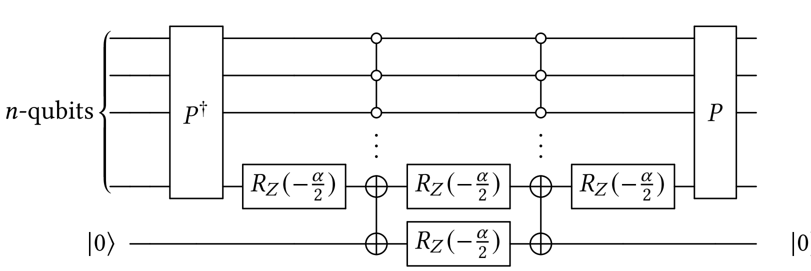
This transformation ensures that the problem is defined with $|E\rangle = |0\rangle$ and $|I\rangle = |0\rangle$, without loss of generality.

Implementation of S_t. As describe through (26), we instead describe how to implement reflection over $|E\rangle$, which is $S_E(\beta) = I - (1 - e^{i\beta})|E\rangle\langle E|$. The implementation of S_E requires oracle O of $|E\rangle$ which is $O|E\rangle|b\rangle = |E\rangle|\neg b\rangle$ and $O|E_\perp\rangle|b\rangle = |E_\perp\rangle|b\rangle$, for boolean $b = 0, 1$, with an additional single ancila bit for $|b\rangle$. Since we assumed to be $|E\rangle = |0\rangle^{\otimes n}$, the desired O is realized in NC^nX gate. The quantum circuit for $S_E(\beta)$ using the oracle is :



where $R_Z(\theta) = e^{-iZ\theta/2}$.

Implementation of S_s. The implementation of S_s (remind that our source state is $|s\rangle = |P\rangle$) requires application of two NC^{n-1} and phases $R_Z(-\alpha/2)$, sandwiched by target program to test P. Note that this P is assumed to prepare program state as $|P\rangle = P|0\rangle$. Quantum circuit for $S_s(\alpha)$ also requires additional single ancila bit. Following circuit implements the S_s operation :



Optimized NC^nX ***by Ancila Bits***. Note that for n-qubit target program, S_t and S_s includes operation of $NC^nX, NC^{n-1}X$ gates, respectively. Originally, NC^kX gives quantum circuit of depth

[1]Here I denotes the identity matrix, not to be confused with input ket-state $|I\rangle$.

quadratic to k when decomposing into $CNOT$+single qubit gates. However, by bringing additional $k-1$ qubits register as ancila qubits, NC^kX can be implemented in linear depth quantum circuit [Nielsen and Chuang 2010].

In the cost analysis of Fpaa (in Section 6), we adopted this method for implementing NC^kX gates. There for, we presumed an additional n qubit space beyond another n-qubit space needed for running the program P. This includes a single qubit for implementing both S_s and S_t, plus $n-1$ qubit space for implementing NC^nX and $NC^{n-1}X$ gates.

B PROOF OF BUG MODEL

Let P and P_{buggy} denote the n-qubit quantum circuit, as illustrated in Fig. 6a and 6b, respectively correct and buggy one. Note that P_{buggy} is dependent to the parameter t through the injection of Z^t. Hence, we denote the buggy program as $P_{\text{buggy}}(t)$.

In this section, we show that the working example (Section 3) and case study (Section 7) illustration by P_{buggy} and setting of $|I\rangle = |0\rangle^{\otimes n}$ is general. Specifically, we show that for any $0 \leq b \leq 1$ there exists $t \in [0,1]$ such that $b = |\langle E_\perp \mid P_{\text{buggy}}(t)\rangle|^2$, where $|E\rangle = P|I\rangle$ for any $I \in \{0,1\}^n$. This shows the existence of at least one possible buggy program for each $b \in [\epsilon, 1]$, supporting the generality of the case study.

Furthermore, we show that for each fixed $t \in [0,1]$, the probability of measuring $|E_\perp\rangle$ over $|P_{\text{buggy}}(t)\rangle$ is invariant for any input $|I\rangle$, where $I \in \{0,1\}^n$ by the corresponding $|E\rangle = P|I\rangle$. That is, $\left|\langle E_\perp \mid P_{\text{buggy}}(t)\rangle\right|^2$ is always the same, regardless of the input $I \in \{0,1\}^n$. This supports that our choice of $|I\rangle = |0\rangle^{\otimes n}$ for the case studies does not loss generality, and the bug detection of $|P_{\text{buggy}}(t)\rangle$ cannot be simply done by giving input specification other than assumed $|I\rangle = |0\rangle^{\otimes n}$.

Altogether, these results are formulated in following Proposition B.1. The result naturally extend to case of Controlled Draper Adder (Fig. 7a and Fig. 7b).

Proposition B.1. *Let $P_{\text{buggy}}(t)$ denote the n-qubit buggy implementation as illustrated in Figure 6b. Then, for any $|I\rangle$ where $I \in \{0,1\}^n$ (and by the corresponding $|E\rangle$),*

$$|\langle E_\perp | P_{\text{buggy}}(t)\rangle|^2 = \sin^2(\frac{\pi}{2}t)$$

(thereby, for any $b \in [0,1]$ there exist $t \in [0,1]$ such that $|\langle E_\perp | P_{\text{buggy}}(t)\rangle|^2 = b$).

Proof. Let $|I\rangle = |y\rangle \otimes |x\rangle$, where $|x\rangle$ and $|y\rangle$ are m-qubit binary state vectors with $n = 2m$. Define $|\psi_k(x)\rangle$ as follows [Draper 2000]:

$$|\psi_k(x)\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\cdot(0.x_k x_{k-1}\ldots x_1)}|1\rangle\right),$$

where x_k is the k-th digit of x in binary representation. If the QFT operates on $|x\rangle$, the state becomes:

$$QFT|x\rangle = |\psi_m(x)\rangle \otimes |\psi_{m-1}(x)\rangle \otimes \cdots \otimes |\psi_1(x)\rangle.$$

Let $z = x + y \mod 2^m$. For $1 \leq k \leq m$, by the controlled phases CR_j appeared in the middle of Draper Adder, the state $|\psi_k(x)\rangle$ evolves to:

$$|\psi_k(z)\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\cdot(0.z_k z_{k-1}\ldots z_1)}|1\rangle\right). \tag{26}$$

The proof proceeds by applying Z^t gate and QFT^{-1} on (26) as shown in Figure 6b. The operation sequence of applying Z^t and QFT^{-1} is illustrated in Fig. 8.

In Fig. 8, M represents a moment in the circuit right before applying the last three gate sequences: H, CR_2^{-1} and another H. At the moment M, the k-th qubit for $1 \leq k \leq m-2$ is already in its final

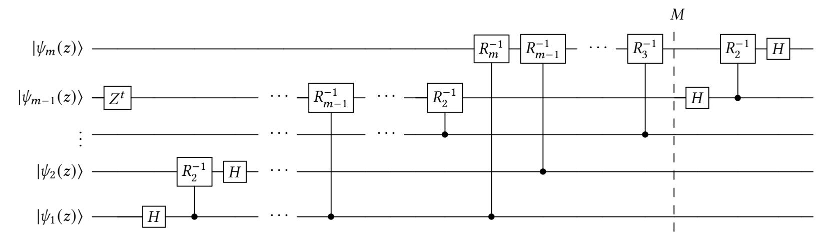


Fig. 8. The Z^t and QFT^{-1} application on $|\phi_k(z)\rangle$ states. Here, R_k^{-1} represents $\begin{pmatrix} 1 & 0 \\ 0 & e^{-2\pi i/2^k} \end{pmatrix}$.

state, correctly derived as $|z_k\rangle$. For the $m-1$-th qubit, due to the effect of the bug gate Z^t, its state is:

$$\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\cdot(0.z_{m-1}+t/2)}\,|1\rangle\right).$$

For the m-th qubit, its state is:

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\cdot(0.z_m z_{m-1})}\,|1\rangle).$$

Summing up, the state at moment M becomes:

$$\underbrace{\frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\cdot(0.z_m z_{m-1})}\,|1\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\cdot(0.z_{m-1}+t/2)}\,|1\rangle\right)}_{=|\phi_1\rangle\otimes|\phi_2\rangle} \otimes |z_{m-2}\rangle \otimes \cdots \otimes |z_1\rangle\,.$$

Since subsequent gates $(I\otimes H), CR_2^{-1}, (H\otimes I)$ only apply to the m-th and $(m-1)$-th qubits, we consider the state evolution on $|\phi_1\rangle, |\phi_2\rangle$. By calculation, we can check that

$$H\,|\phi_2\rangle = \frac{1}{2}\left[(1+e^{\pi i t})\,|z_{m-1}\rangle + (1-e^{\pi i t})\,|\neg z_{m-1}\rangle\right] =: |\phi_2'\rangle$$

where $z_{m-1} \in \{0,1\}$ (which will be decided by input $|I\rangle$). Then, after the application of remaining gates CR_2^{-1} (where the control is applied on the $(m-1)$-th qubit) and $(H\otimes I)$, sequentially, the state ends in

$$|\phi^*\rangle := (H\otimes I)CR_2^{-1}(|\phi_1\rangle \otimes |\phi_2'\rangle) = \frac{1}{2}\left[(1+e^{\pi i t})\,|z_m\rangle\,|z_{m-1}\rangle + (1-e^{\pi i t})\,|?\rangle\,|\neg z_{m-1}\rangle\right] \tag{26}$$

where the part $|?\rangle$ is result of applying phase R_2^{-1} wrongly controlled by $|\neg z_{m-1}\rangle$.

Remind that $|E\rangle = |z_m\rangle \otimes |z_{m-1}\rangle \otimes \cdots \otimes |z_1\rangle$. Then, the final program state vector can be represented in:

$$|P_{\text{buggy}}(t)\rangle = |\phi^*\rangle \otimes |z_{m-2}\rangle \otimes \cdots \otimes |z_1\rangle = \frac{1+e^{i\pi t}}{2}\,|E\rangle + \frac{1-e^{i\pi t}}{2}\,|E_\perp\rangle$$

Hereby, we can check that the probability of measuring $|E_\perp\rangle$ is

$$\left|\frac{1-e^{i\pi t}}{2}\right|^2 = \sin^2(\frac{\pi t}{2}).$$

This derivation was independent to choice of $|I\rangle$, hence the result holds for all $I \in \{0,1\}^n$.

□