Matrix Multiplication with compressed gadget

Sasan Moradi[1*]

[1]IQOQI, university of Vienna, Senssengasse, Vienna, 1090, Vienna, Austria.

Corresponding author(s). E-mail(s): sassan.moradi@gmail.com;

0.1 Matrix multiplications

There are several methdos for performing matrix-matrix multiplication on quantum computers. One method involves duplicating the ancilla qubits relying on block-encoding [1, 2]. The number of ancilla qubits and two-qubit swap gates required for matrix multiplication increases linearly with the number of matrix multiplications. For example, computing E^p, the power p of matrix E, using the duplicate ancilla qubits technique requires additional $O(p)$ ancilla qubits and $O(p)$ swap gates. Each swap gate is decomposed into 3 $CNOT$ gates. Consequently, the duplicate ancilla qubits technique is inefficient for multiplications of non-unitary matrices. Another method for multiplication of the non-unitary matrices is called compression gadget [3]. The number of extra ancilla qubits needed for computing matrix multiplication increases logarithmically with the number of multiplications. For instance, E^p can be computed with $O(log_2(p))$ ancilla qubits using the compression gadget. First, we explain the compression gadget in a lemma, and then we describe the quantum circuit for the compression gadget.

Lemma 1(compression gadget for the matrix multiplications) [4]. Suppose we are given unitaries $U_1, U_2, \ldots, U_L$, each of which is a $(\alpha'_l, m'_l, 0)$-block encoding of A_l. Then the $(\alpha_{\text{comp}}, m_{\text{comp}}, 0)$-block encoding of $A_L \cdots A_2 A_1$ is constructed, where

$$\alpha_{\text{comp}} = \alpha'_1 \alpha'_2 \cdots \alpha'_L, \quad m_{\text{comp}} = \max_l(m'_l) + \lceil \log_2(L) \rceil + 1 \tag{1}$$

using one application of each U_l. In Eq. 1, m'_l is the number of qubits needed for block encoding of A_l. Fig. 1 shows the quantum circuit for the compression gadget for multiplication of $A_L ... A_2 A_1$. The counter register in Fig. 1 contains $log_2(L) + 1$ qubits. It is used to keep track of how many block encoding of A_l have been applied successfully. The ADD operation in Fig. 1 is a unitary operation. It maps $ADD\,|0\rangle =$

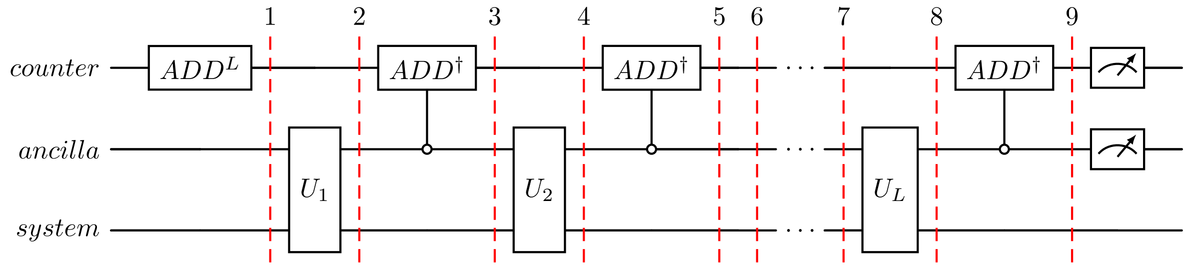

Fig. 1 Quantum Circuit for computation of matrix multiplication with compression gadget. ADD is applied L times on the counter register to maps $ADD^L\left|0\right\rangle = \left|L\right\rangle$ (step 1). At the next step, unitary block encoding U_1 is applied to the ancilla qubits and system qubits. This operation generates a superposition between $\left|0\right\rangle^{\otimes a}$ and $\left|1\right\rangle^{\otimes a}$ (step 2). The next step is to utilize the information in the ancilla register to change the information in the counter register. The controlled $ADD^\dagger$ operation, conditioned on a-ancilla qubits all in the state $\left|0\right\rangle^{\otimes a}$ is applied to change the information in the counter registers (step 3). Steps 2 and 3 are repeated until the quantum state of the counter register is brought back to $\left|0\right\rangle^{\otimes c}$. Finally, the counter register and ancilla registers are measured. The output after postselection is the matrix $A_L ... A_2 A_1$.

$\left|1\right\rangle$, $ADD\left|1\right\rangle = \left|2\right\rangle$,..., $ADD\left|L-1\right\rangle = \left|L\right\rangle$ and $ADD\left|L\right\rangle = \left|0\right\rangle$ and the $ADD^\dagger$ is the hermitian conjugate of ADD. It maps $ADD^\dagger\left|0\right\rangle = \left|L\right\rangle$, $ADD^\dagger\left|L\right\rangle = \left|L-1\right\rangle$, ..., $ADD^\dagger\left|2\right\rangle = \left|1\right\rangle$, and $ADD^\dagger\left|1\right\rangle = \left|0\right\rangle$.

Let's consider that the initial state is $\left|0\right\rangle^{\otimes(c=log(L)+1)}\left|0\right\rangle^{\otimes(a=\frac{m_l'+1}{2})}\left|0\right\rangle^{\otimes(s=\frac{m_l'-1}{2})}$. First ADD operator applies L times on the counter register c to map $\left|0\right\rangle^{\otimes c}$ to $\left|L\right\rangle^{\otimes c}$. The quantum state after step 1 is $\left|L\right\rangle^{\otimes c}\left|0\right\rangle^{\otimes a}\left|0\right\rangle^{\otimes s}$. Then the first unitary block encoding U_1 is applied on $\left|L\right\rangle^{\otimes c}\left|0\right\rangle^{\otimes a}\left|0\right\rangle^{\otimes s}$. This operation generate a superposition between $\left|0\right\rangle^{\otimes a}$ and $\left|1\right\rangle^{\otimes a}$. The quantum state is $\left|L\right\rangle^{\otimes c}(\left|0\right\rangle^{\otimes a} A_1 \left|0\right\rangle^{\otimes s} + \left|1\right\rangle^{\otimes a}\left|*\right\rangle^{\otimes s})$. $\left|*\right\rangle$ is called a junk state. The next step is to utilize the ancilla register to change the information in the counter register. This is achieved by applying a controlled $ADD^\dagger$ operation, conditioned on a-control qubits all in the state $\left|0\right\rangle^{\otimes a}$. The quantum state after this operation (step 3) is $\left|L-1\right\rangle^{\otimes c}\left|0\right\rangle^{\otimes a} A_1 \left|0\right\rangle^{\otimes s} + \left|L\right\rangle^{\otimes c}\left|1\right\rangle^{\otimes a}\left|*\right\rangle^{\otimes s}$. After step 3, another unitary block encoding U_2 is applied. The quantum state after applying U_2 operation is $\left|L-1\right\rangle^{\otimes c}(\left|0\right\rangle^{\otimes a} A_2 A_1 \left|0\right\rangle^{\otimes s} + \left|1\right\rangle^{\otimes a}\left|*\right\rangle^{\otimes s}) + \left|L\right\rangle^{\otimes c}(\left|0\right\rangle^{\otimes a}\left|*\right\rangle^{\otimes s} + \left|1\right\rangle^{\otimes a}\left|*\right\rangle^{\otimes s})$ (step 4). Then another controlled $ADD^\dagger$ by considering the control qubit $\left|0\right\rangle^{\otimes a}$ is applied and a new counter register $\left|L-2\right\rangle^{\otimes c}$ is generated. The new quantum state is $\left|L-2\right\rangle^{\otimes c}\left|0\right\rangle^{\otimes a} A_2 A_1 \left|0\right\rangle^{\otimes s} + \left|L-1\right\rangle^{\otimes c}\left|1\right\rangle^{\otimes a}\left|*\right\rangle^{\otimes s} + \left|L-1\right\rangle^{\otimes c}\left|0\right\rangle^{\otimes a}\left|*\right\rangle^{\otimes s} + \left|L\right\rangle\left|1\right\rangle^{\otimes a}\left|*\right\rangle^{\otimes s}$ (step 5). These steps must be repeated until all the unitary block encoding operations are applied and the counter register is brought back to its initial state $\left|0\right\rangle^{\otimes c}$. The quantum state before measuring the counter qubit and the ancilla qubits is $\left|0\right\rangle^{\otimes c}\left|0\right\rangle^{\otimes a} A_L ... A_2 A_1 \left|0\right\rangle^{\otimes s} + \sum_{j>0}\left|j\right\rangle\left|*\right\rangle\left|*\right\rangle$ (step 9). Measuring the counter qubits and the ancilla qubits on the basis $\left|0\right\rangle^{\otimes c}\left|0\right\rangle^{\otimes a}$ in combination with post-selection guarantee that the multiplication $A_L ... A_2 A_1$ is applied successfully (step 10). The code for matrix multiplication with the compressed gadget technique is available on GitHub.

References

[1] Gilyén, A., Su, Y., Low, G.H., Wiebe, N.: Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing. ACM, ??? (2019). https://doi.org/10.1145/3313276.3316366

[2] Takahira, S., Ohashi, A., Sogabe, T., Usuda, T.S.: Quantum Algorithms based on the Block-Encoding Framework for Matrix Functions by Contour Integrals (2021)

[3] Low, G.H., Wiebe, N.: Hamiltonian Simulation in the Interaction Picture (2019). https://arxiv.org/abs/1805.00675

[4] Fang, D., Lin, L., Tong, Y.: Time-marching based quantum solvers for time-dependent linear differential equations. Quantum **7**, 955 (2023) https://doi.org/10.22331/q-2023-03-20-955