Writing Report (QRE 2024 Challange)

Razeen Ud Din and Dr. Jibran Rashid

(Dated: July 2024)

I. PROJECT PART(A)

There are quantum algorithms that start from a specific quantum state. For example, in the HHL algorithm, in which we need to prepare a state that encodes the vector "b" beforehand to solve the problem

$$Ax = b$$

. The process of creating a specific initial quantum state that may encode input data or a problem instance is known as *quantum state preparation.*This project is about the efficient implementation of an algorithm for *sparse quantum state preparation*

INTRODUCTION

State preparation is an essential part of quantum computing and much more complex than their classical counterparts. In classical computing all possible states of a memory-block of size O(n) can be obtained by setting the appropriate bits using O(n) NOT-gates.The possible states of a quantum computer are superpositions of basis states and some of them can only be created with very complex circuits. In fact, it is known that using a gate library of only one-qubit-gates and CNOT-gates, some of the states of an n-qubit quantum computer can only be prepared with circuits of size $\Omega(2^n)$.

Definintion 1. *In contrast to general states, many (if not most) practically relevant states have the property that if S is the set of basis states of a quantum state $|S| << 2^n$ i.e only a small proportion of the basis states have nonzero coefficients. We call states with this property* sparse.

Example 2. *Few examples are*

- *The proportion of basis states with nonzero coefficient among all basis states is $O(\frac{1}{\sqrt{2^n}})$ for generalized Bell states.*
- *$O(\frac{n}{2^n})$ for W-states and $O(\frac{1}{2^n})$ for GHZ-states.*

Niels Gleinig and Torsten Hoefler in their paper [1]show how sparsity can be leveraged to make state preparation asymptotically more efficient than preparation of general states.

[1] An Efficient Algorithm for Sparse Quantum State Preparation

Review of main results of the paper "An efficient algorithm for sparse quantum state preparation"

1. *Calculating parameters of merging gate*

The main crux of this algorithm is the way of choosing the two basis states and then applying the specific controlled-Gate "G" that merge them into one while keeping the other basis states in the form that this "G" isn't affect them. Repeating this eventually leads to the state that have only $|0^n\rangle$ basis state. The mathematical argument for this gone through following steps.

$$G = \begin{bmatrix} \sin(\omega) & e^{ia}\operatorname{Cos}(\omega) \\ e^{-ia}\operatorname{Cos}(\omega) & -\operatorname{Sin}(\omega) \end{bmatrix}, \quad |\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \quad |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

We know that $G|\psi\rangle = e^{i\lambda}|0\rangle$, $|\alpha|^2 + |\beta|^2 = 1$

$$\begin{bmatrix} \sin(\omega) & e^{ia}\operatorname{Cos}(\omega) \\ e^{-ia}\operatorname{Cos}(\omega) & -\operatorname{Sin}\omega \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} e^{i\lambda} \\ 0 \end{bmatrix}.$$

This implies,

$$\alpha \operatorname{Sin}(\omega) + \beta e^{ia} \operatorname{Cos}\omega = e^{i\lambda} \tag{1.1}$$

$$\alpha e^{-ia}\operatorname{Cos}(\omega) - \beta \sin(\omega) = 0 \tag{1.2}$$

Let $\alpha = e^{is}\operatorname{Cos}(\theta)$ and $\beta = e^{it}\operatorname{Sin}(\theta)$ where θ belongs to the first quadrant. Putting above supposition in equations (1.1) and (1.2), we got

$$e^{is}\cos\theta\sin(\omega) + e^{it}\sin\theta e^{ia}\cos\omega = e^{i\lambda} \tag{1.3}$$

$$e^{is}\cos\theta e^{-ia}\cos\omega - e^{it}\sin\theta\sin\omega = 0 \tag{1.4}$$

Which on simplifying give us

$$e^{is}\left(\operatorname{Cos}\theta\sin\omega + e^{i(t-s+a)}\sin\theta\operatorname{Cos}\omega\right) = e^{i\lambda} \tag{1.5}$$

$$e^{it}\left(e^{i(s-t-a)}\cos\theta\operatorname{Cos}\omega - \operatorname{Sin}\theta\operatorname{Sin}\omega\right) = 0 \tag{1.6}$$

Since we are looking for one solution that satisfy the above equations, without loss of generality, we can select

$$a = s - t \tag{1.7}$$

With this selection, the above matrix equations 1.5 and 1.6 takes the form of below equations 1.8 and 1.9 in which we claim, we would find such w which would satisfy the both equations 1.8 and 1.9

$$e^{is}(\operatorname{Cos}\theta\sin\omega + \quad \sin\theta\operatorname{Cos}\omega) = e^{i\lambda} \tag{1.8}$$

$$e^{it}(\operatorname{Cos}\theta\operatorname{Cos}\omega - \sin\theta\sin\omega) = 0 \tag{1.9}$$

Simplifying above 2 equations,

$$e^{is}(\sin(\theta+\omega)) = e^{i\lambda} \tag{1.10}$$

$$e^{it}(\cos(\theta+\omega)) = 0 \tag{1.11}$$

Simplifying equation 1.9,
$\cos(\theta+\omega) = 0$
$\theta+\omega = \operatorname{Cos}^{-1}(0)$
$\theta+\omega = \frac{\pi}{2}$
$\operatorname{Sin}(\theta+\omega) = e^{i(\lambda-s)}$
$= \operatorname{Cos}(\lambda-s) + i\operatorname{Sin}(\lambda-s)$
$\operatorname{Sin}(\lambda-s) = 0 \Rightarrow \lambda = s$
$\operatorname{Cos}(\lambda-s) = \operatorname{Sin}(\theta+\omega)$
$\lambda - s = \operatorname{Cos}^{-1}(1)$
$\lambda = s$
i.e. After merging, merged state would have phase λ
$\alpha = e^{is}\sin\omega$
$\beta = e^{it}\operatorname{Cos}\omega$
$|\alpha| = \left|e^{is}\sin\omega\right|$
$|\alpha| = |\sin\omega|$
$|\beta| = |\operatorname{Cos}\omega|$
We can always select w in first quadrant. Still, above 2 equations are satisfied simultaneously. Dividing one from other gives

$$w = \tan^{-1}\left(\frac{|\alpha|}{|\beta|}\right)$$

This solution of w is consistent with the equations 1.4 and 1.5.

A. Sparse quantum state preparation Algorithm

1. Input

The input of the algorithm is the classical description of the quantum state which we want to prepare, this description contains two parts the basis states and their corresponding probabilities.

2. Output

The outcome of the algorithm is the circuit C having the property when applies to a all zero state provide us the required quantum state.

The algorithm has two parts, classical and quantum.

3. Classical Preprocessing

This part of the sparse state preparation algorithm provide all the necessary classical information needed to proceed for the quantum part of the algorithm. After taking the input the algorithm pass both the basis states and their probabilities to the function named as $Algo\_1()$, this function using the dedicated process of selecting two basis states that will be merged into one along with the piece of information about how to do it i.e by providing control and target qubits for multi-controlled operation G. After getting this information, the next part of this function is to do a merging and return the list of basis states and their probabilities which has one less number of elements than the list provided as an input. In short this part takes the basis states and their probabilities and return the same with one less in numbers. This function run itself multiple number of times unless only one element left in the list. Along the process this function saves history of all gates in form of different lists and pass them to the quantum part of the algorithm.

4. Quantum part of the Algorithm

Sparse State information S is in the form of [[[basis],probability],...]. n is the number of qubits in the desired sparse state. We used Classical Function $Algo\_1()$ to generate history of Gates corresponding to recursively merging of basis states present in the set S and finally getting $|0^n>$ state. We used lists to store required Gates history. Merging Gate itself is the inverse of itself. X,CX Gates also are the inverse of themselves. The inversion of the whole merging circuit would give us the desired Sparse State Preparation circuit. This inversion requires only reading these Gates from the lists in reverse order and applying them as they are. The Sparse$_Sstate() function does this job by reading the lists and creating Gat$

Our codes can be accessed from the Github link - github.com/Razeen-ud-din/Quantum-Sparse-State-Preparation

5. Applying the merging gate part of the algorithm

The algorithm step of applying the gate G that merge the two basis states into one and add up their probabilities into one that remains after merging as discussed above, we are using the result given in Elementary gates using the paper [2]. The result are as follows:

Lemma 3. *$A \wedge_1(W)$ gate can be simulated by a network of the form in figure 1 with $A = \mathrm{R_z}(\alpha)\cdot\mathrm{R_y}\left(\frac{\theta}{2}\right) andB = \mathrm{R_y}\left(-\frac{\theta}{2}\right)\cdot\mathrm{R_z}(-\alpha)$ if and only if W is of the form*

$$W = \mathrm{R_z}(\alpha)\cdot\mathrm{R_y}(\theta)\cdot\mathrm{R_z}(\alpha) = \begin{pmatrix} e^{i\alpha}\cos\theta/2 & \sin\theta/2 \\ -\sin\theta/2 & e^{-i\alpha}\cos\theta/2 \end{pmatrix},$$

where α and θ are real-valued.

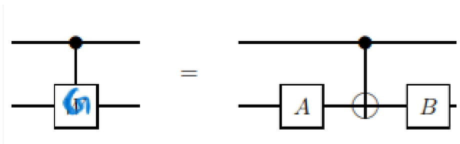

FIG. 1. Lemma for elementary gates

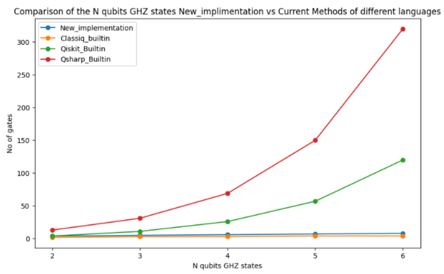

FIG. 2. comparision of N qubit GHZ state preparation between different langauges with this new implementation

B. Simulations Results of the Implementation in different languages

The comparison for different N qubits GHZ and W states of current Classiq QSharp and qiskit state preparation techniques with this new implementation is given in figure 2 and 3

C. Resource Estimation Results

The results from the resource estimation are consistents with the simulation results, the number of qubits in each case wheateher this new implementation of the paper and the builtin methods of generating states remains same throughout. The other parameters (like number of logical depth, runtime and No of T gates etc ...) are much less in the new implementation compare to the builtin methods of langauages (i.e qiskit and Qsharp). The graphs in 4 and 5 support these results.

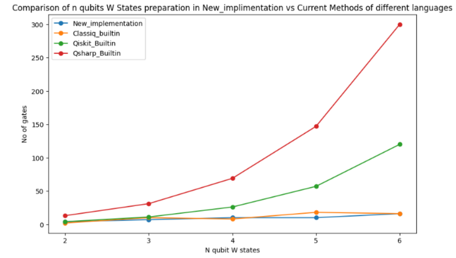

FIG. 3. comparision of N qubit W state preparation between different langauges with this new implementation

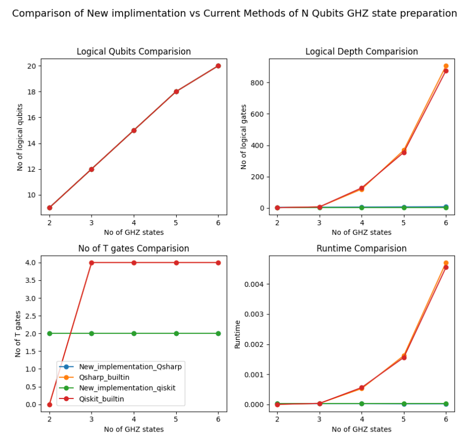

FIG. 4. Resource Estimation of GHZ States

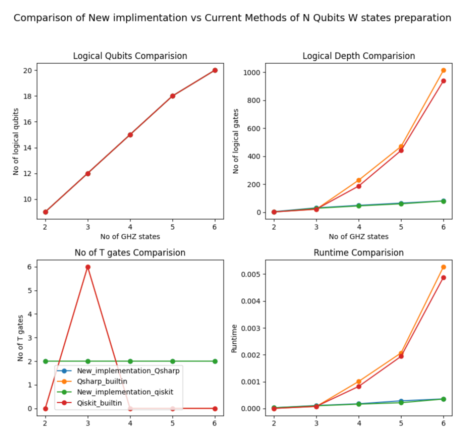

FIG. 5. Resource Estimation of W States

[2] elementary gates for quantum computation