MQC – Measurement-based Quantum Computing (R. Jozsa)

Introduction

* So far we have the circuit model of quantum computation, motivated by the obvious classical model. There are also quantum analogues of other classical models (Turing machines, cellular automata etc.)
* measurement-based (or "one-way") quantum computing is an architecture that has no classical analogue. It is universal in the sense that it can simulate the circuit model with only a polynomial overhead in physical resources.
* It emphasises the role of entanglement as a resource that is irreversibly consumed in this model as the computation progresses (hence the name "one way") – computational steps will be (1-qubit) measurements, not unitary gates!

Preliminary notations

"mmt" – abbreviation for "measurement".

Qubit states

$$|\pm_\alpha\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle \pm e^{-i\alpha}|1\rangle\right)$$

$|\pm_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$ also written just as $|\pm\rangle$.

$\mathcal{B}(\alpha) = \{|+_\alpha\rangle, |-_\alpha\rangle\}$ is an orthonormal basis.

1-qubit gates

$$J(\alpha) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & e^{i\alpha} \\ 1 & -e^{i\alpha}\end{bmatrix} = HP(\alpha)$$

$$H = J(0) = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1\end{bmatrix} \qquad P(\alpha) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha}\end{bmatrix}$$

Pauli gates: $X = \begin{bmatrix} 0 & 1 \\ 1 & 0\end{bmatrix}$ $\quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1\end{bmatrix} = P(\pi)$

2-qubit gate

$E = CZ$ (controlled-Z) $= \mathrm{diag}(1,1,1,-1)$ in standard basis.

E for "entangling". E is symmetric $E_{12} = E_{21}$.

We will use E_{ij} only on nearest-neighbour (n.n.) qubit lines
i.e. $j = i \pm 1$ in circuits.

1-qubit measurements

$M_i(\alpha)$: measurement of qubit i in basis $B(\alpha)$
(e.g. rotate $B(\alpha)$ to $\{|0\rangle, |1\rangle\}$ by applying $J(\alpha)$ and measure in std. basis)
Outcomes corresponding to $|+_\alpha\rangle$ (resp. $|-_\alpha\rangle$) denoted 0 (resp. 1).

$M_i(Z)$: measurement of qubit i in std. basis.
outcome $|0\rangle$ (resp. $|1\rangle$) denoted 0 (resp. 1).

Recall extended Born rule:
To find effect of $M_1(\alpha)$ on 1st qubit of 2-qubit state
$$|\psi_{12}\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$$
first write 1st qubit in $B(\alpha)$ basis using (in 1st slot):
$$|0\rangle = \frac{1}{\sqrt{2}}\left(|+_\alpha\rangle + |-_\alpha\rangle\right)$$
$$|1\rangle = \frac{e^{i\alpha}}{\sqrt{2}}\left(|+_\alpha\rangle - |-_\alpha\rangle\right)$$
Then collect all terms with $|+_\alpha\rangle$ resp. $|-_\alpha\rangle$ giving the form
$$|\psi_{12}\rangle = |+_\alpha\rangle_1 \left[|\psi_+\rangle_2\right] + |-_\alpha\rangle_1 \left[|\psi_-\rangle_2\right]$$
Then for mmt outcomes $s = 0$ or 1:

$s=0$: prob $p_0 = \langle\psi_+|\psi_+\rangle$, post-mmt state is $|+_\alpha\rangle_1|\psi_+\rangle_2 / \sqrt{p_0}$

$s=1$: prob $p_1 = \langle\psi_-|\psi_-\rangle$, post-mmt state is $|-_\alpha\rangle_1|\psi_-\rangle_2 / \sqrt{p_1}$ //.

Graphstate $|\psi_G\rangle$:

Let $G = (V, E)$ (with V and E being vertices & edges)
be any graph that has $\quad |V| =$ number of vertices
(i) undirected edges
(ii) no self-loop edges (from a vertex to itself)
(iii) at most one edge between any two vertices.

Then $|\psi_G\rangle$ is the state on $|V|$ qubits obtained as follows:
- for each vertex $i \in V$ introduce a qubit $|+\rangle_i$
- for each edge $\overset{i}{\circ}\!\!-\!\!-\!\!\overset{j}{\circ}$ apply E_{ij} (they all commute)

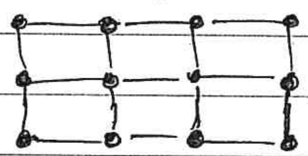
e.g. $G_1 = \overset{1}{\circ}\!\!-\!\!-\!\!\overset{2}{\circ}$ $\quad |\psi_{G_1}\rangle = E_{12}|+\rangle_1|+\rangle_2 = \frac{1}{2}\left[|00\rangle+|01\rangle+|10\rangle-|11\rangle\right]$

$G_2 = \overset{1}{\circ}\!\!-\!\!-\!\!\overset{2}{\circ}\!\!-\!\!-\!\!\overset{3}{\circ}$ $\quad |\psi_{G_2}\rangle = E_{12}\,E_{23}\,|+\rangle_1|+\rangle_2|+\rangle_3$

$$= \frac{1}{2\sqrt{2}}\left[|000\rangle+|001\rangle+|010\rangle-|011\rangle+|100\rangle+|101\rangle-|110\rangle+|111\rangle\right]$$

<u>Cluster state</u>: is graph state $|\psi_G\rangle$ for G being any rectangular 2D grid.

e.g. $G =$



* Later we will need only graphs that are subgraphs of a 2D rectangular grid (obtained by removing some vertices and all associated edges).

* We will often use the edge picture $\circ\!\!-\!\!-\!\!\circ$ to denote E acting on two qubits (not necessarily $|+\rangle|+\rangle$), which are represented by the vertices (blobs).

Measurement-based quantum computing (MQC) — the main result stated.

Let C be any quantum computation given as a quantum circuit C on n qubits i.e. as a sequence of unitary gates $U_1, U_2, \ldots, U_K$ applied in order on a specified input n-qubit state $|\psi_{in}\rangle$ (usually a computational basis state) and followed by final Z-mmts $M_j(Z)$ on specified qubits $j = i_1, \ldots, i_k$, to obtain an output k-bit string.

* Then we can always simulate the result of this quantum computation as follows:

The starting resource: start with a graph state $|\psi_G\rangle$.
Here G is chosen depending on the connectivity structure of the circuit C (or G = a 2D grid suffices too – see later.)
The computational steps: each step is a 1-qubit mmt instruction of the form $M_i(\alpha)$. Here the value of α may depend on the (random) outcomes $s_1, s_2, ..$ of previous mmts i.e. we have an adaptive sequence of mmts.
The computational process: we are given a prescribed sequence of (adaptive) computational steps
$M_{i_1}(\alpha_1), \; M_{i_2}(\alpha_2), .. \quad M_{i_N}(\alpha_N)$
with qubit labels $i_1, i_2, .. i_N$ all distinct. In fact we can discard each qubit i after its mmt, retaining only the mmt outcome s_i for possible use in determining the choice of angles α in future mmts (and in outputs – cf below).
We also retain all unmeasured qubits.
The output: we first obtain the results $s_{i_1}, .., s_{i_k}$ of $M(Z)$ mmts on k specified qubits (which have not previously been measured). Then finally we process these results by further (simple) classical computations involving them as well as previous $M_i(\alpha_i)$-mmt outcomes, to obtain the actual output bits.
Remark:
Mmts are usually regarded as destructive but here they have a constructive role as being our computational steps. We start with a fiducial entangled state $|\psi_G\rangle$ and successively degrade its entanglement by 1-qubit mmts – hence the name "one-way model" – as the entanglement is irreversibly consumed in the process.
For each $M_i(\alpha_i)$ mmt, the outcome s_i is probabilistic and in fact always uniformly random (cf later). Intuitively this randomness in the process is compensated by subsequent α choices being chosen dependent on previous outcomes, to simulate a deterministic unitary evolution up to the final $M(Z)$'s.

How and Why MQC works!

We begin by noting:

FACT: the 1-qubit gates $J_i(\alpha)$ (for all α) together with n.n. $CZ_{ij} = E_{ij}$ (ie. $j = i \pm 1$) comprise a universal set of quantum gates. $\square$

In particular any 1-qubit gate U (up to overall phase) can be written as a product of three J's:

$$U = e^{i\xi}\, J(\alpha)\, J(\beta)\, J(\gamma)$$

(Which can be directly seen using a standard parameterisation of the unitary group $U(2)$ in 2 dimensions)

The n.n. condition $j = i \pm 1$ can be imposed since we can easily construct the SWAP gate of two lines e.g. –

$SWAP_{12} = (CX)_{12}\,(CX)_{21}\,(CX)_{12}$ and $(CX)_{12} = H_2\,(CX)_{12}\,H_2$ with $H_2 = J_2(0)$.

Then distant line actions can be represented using ladders of SWAPs.

* Thus we may assume that the gates of our given circuit C are all of the form

$J_i(\alpha)$ or E_{ij} with $j = i \pm 1$

Next we have the core result:

J-lemma: (how to apply gates by doing mmts!)

For any 1-qubit state $|\psi\rangle = a|0\rangle + b|1\rangle$ consider $E_{12}\,[\,|\psi\rangle, |+\rangle_2\,]$ followed by $M_1(\alpha)$.

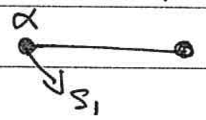
Suppose that the outcome is s_1.

Then after the mmt, the state of qubit 2 is $X^{s_1} J(\alpha)|\psi\rangle$.

Also the two outcomes $s_1 = 0, 1$ always occur with equal probabilities ½ (regardless of the values of a, b, α). $\square$

Proof: an easy calculation using the Born rule. See Exercise Sheet 2. //

We will denote the process in the J-lemma pictorially as



The labels on the left qubit (1) denote the mmt $M_1(\alpha)$ with outcome s_1 and the process leaves the right qubit in state $X^{s_1}J(\alpha)|\psi\rangle$ where $|\psi\rangle$ was the initial state of the left qubit.
• This is sometimes called "1-bit teleportation" as the (altered version of) $|\psi\rangle$ is moved from site 1 to site 2.
Subsequently qubit 1 is left in state $|+_\alpha\rangle$ or $|-_\alpha\rangle$ (for $s_1 = 0$ or 1) and can be discarded.
Similarly a Z-mmt $M(Z)$ with outcome i will be denoted

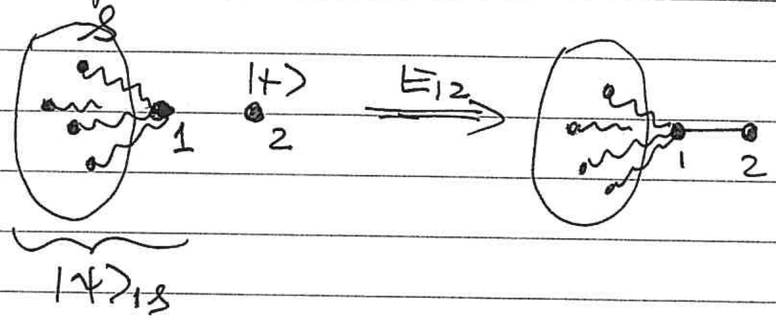
An extension of the J-lemma: the same result holds if $|\psi\rangle$ is an entangled state of many qubits, extending a qubit labelled 1. i.e. $X^{s_1}J(\alpha)$ gets applied to site 1 while keeping the entanglements intact (and site 1 is replaced by a new site) — in this scenario we can write

$$|\psi\rangle_{1S} = |a\rangle_S |0\rangle_1 + |b\rangle_S |1\rangle_1$$

where S represents a system of further qubits i.e. the coeffs a, b in our previous 1-qubit state $|\psi\rangle$, have been replaced by vectors $|a\rangle_S$ and $|b\rangle_S$ from the statespace of S.
Since the Born rule involves just application of a linear projection operator on qubit 1, the calculations go through equally well if the coeffs a, b are complex numbers (= 1-dim vectors) or vectors (states of S).
Thus introducing an extra new qubit 2 (not in S) in state $|+\rangle_2$ and performing $M_1(\alpha)E_{12}|\psi\rangle_{1S}|+\rangle_2$ (and then discarding the measured qubit 1) we get

$$X_2^{s_1} J_2(\alpha)|\psi\rangle_{2S}$$

i.e. $X^{s_1} J(\alpha)$ has been applied to qubit 1 of $|\psi\rangle_{1S}$ and this qubit has been re-labelled as 2 :



then $M_1(\alpha)$ with outcome s_1 gives $X^{s_1} J(\alpha)$ applied to $|\psi\rangle_{1S}$ (and 1 renamed as 2).

We will use the J-lemma to simulate the action of $J(\alpha)$ (up to a possible X "error") using E and the mmt $M(\alpha)$ and we will also want to concatenate such J-lemma applications for sequences of $J(\alpha)$ gates.

<u>Concat lemma</u>: if we concatenate the process of the J-lemma on a row of qubits 1, 2, 3, .. to apply a sequence of $J(\alpha)$ gates then <u>all</u> the entangling operations E_{12}, E_{23}, --- can be done <u>first</u> before any mmts are applied.

<u>FACT</u>: For any composite quantum system AB, any local actions (unitary gates or mmts) done on A always commute with any done on B.

<u>Proof</u>: If $|\psi\rangle_{AB}$ is any (generally entangled) state of AB then local unitary operations U_A and V_B done on A and B respectively correspond to operators $U_A \otimes I_B$ and $I_A \otimes V_B$ on the <u>full</u> system, and these clearly commute:

$$(U_A \otimes I_B)(I_A \otimes V_B) = (I_A \otimes V_B)(U_A \otimes I_B) = U_A \otimes V_B .$$

Similarly for local mmts, represented by actions of linear projection operators P_A and Q_B at A and B, replacing U_A and V_B above. //

Concat lemma Proof:

For $|\psi\rangle_1 |+\rangle_2 |+\rangle_3 \ldots$ the sequence of J-processes is the sequence of operations (from left to right):

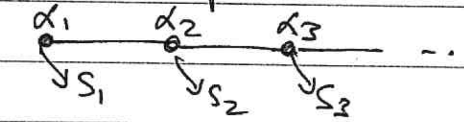
$E_{12}\ M_1(\alpha_1)\ E_{23}\ M_2(\alpha_2)\ E_{34}\ M_3(\alpha_3) \ldots$

Each $E_{i,i+1}$ acts on qubits disjoint from all previous measurements (and E's all commute)

So by (FACT), all E's can be moved to the left over all M's there to give $E_{12}, E_{23}, E_{34}, \ldots, M_1(\alpha_1), M_2(\alpha_2), M_3(\alpha_3) \ldots$ //

Remarks:

- We denote this process as



which implements the 1-qubit circuit

$$|\psi\rangle — \boxed{J(\alpha_1)} — \boxed{X^{s_1}} — \boxed{J(\alpha_2)} — \boxed{X^{s_2}} — \ldots$$

- the $E_{i,i+1}$'s all commute (even on overlapping qubits) so can physically be applied in any order or even simultaneously.
- the $M_i(\alpha_i)$ mmts are all on disjoint qubits so can be done in any order unless the choice of angle α_i depends on the outcome of previous mmts (i.e. adaptive choice of mmts).

Determining the MQC process corresponding to a given circuit

Consider now any circuit C (on n qubits) comprising a sequence of gates $U_1, U_2, \ldots U_K$ applied in order, in which each U_i is either a $J_i(\alpha)$ gate or a n.n. E_{ij} gate.

We will always take the input state to be $|+\rangle|+\rangle \ldots |+\rangle$. This is without loss of generality as any 1-qubit state $|\psi\rangle$ may be written $|\psi\rangle = U|+\rangle$ for a suitable U which may then be represented using at most three $J(\alpha)$'s (by universality). Thus for a general product state input $|\psi_1\rangle \ldots |\psi_n\rangle$ we first prefix C by this extra construction on each line eg. for the computational basis state $|j\rangle$ ($j = 0, 1$) we have $|j\rangle = X^j H|+\rangle$ and $H = J(0)$, $X = J(\pi)J(0)$.

We write the input qubits as a vertical row of blobs. Note:

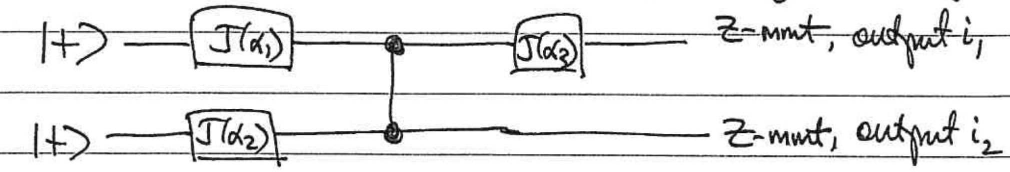
(i) all $J(\alpha)$ gates will be implemented by the J-process. (and we'll see later how to deal with the extra unwanted X^{s_i} gates that arise)

(ii) all n.n. E_{ij}'s will be applied by exploiting the E gates used to make an initial graph state (like the E's used in the J-lemma & Concat lemma re-ordering) - cf below.

(iii) the final outputs will be obtained by $M_i(Z)$ mmts.

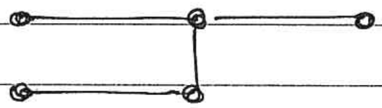
By the concat lemma, all the E's in (i) & (ii) can be done first (before any mmts). This results in a graph state on a graph G that's a subgraph of an $(n \times l)$ rectangular grid D, where l is the depth of the circuit C (not counting the E gates in C). This graph state $|\psi_G\rangle$ can always be made by applying Z mmts to the graph state D to cull vertices (cf Sheet 2 Q8(vi)). Having made this graph state, the whole computation is translated into just a sequence of 1-qubit mmts on $|\psi_G\rangle$ (or equivalently, on $|\psi_D\rangle$ by first preparing $|\psi_G\rangle$ via Z-mmts).

Example: Consider the circuit C given by the diagram:

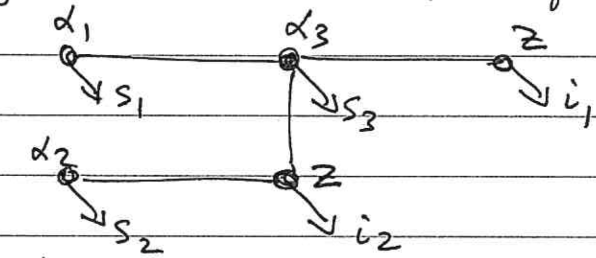


(where represents an E gate as usual)

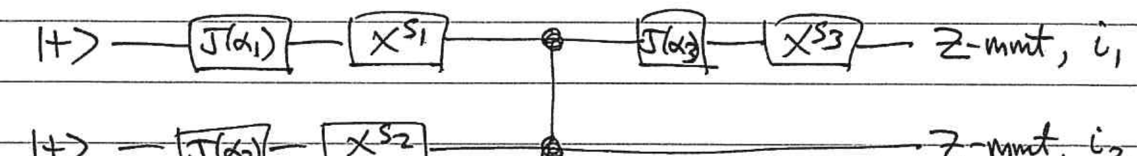
Each $J(\alpha_i)$ gate will be implemented using the J-lemma. Thus for each such gate we'll make the entangled pair, and as noted above, all these entangling operations can be done ab initio, including also the E gates of the circuit itself. Thus we'll use the graph state



If we just measure all the qubits for the J-processes & outputs i.e.



we would effect the circuit:



where s_1, s_2, s_3 have been chosen randomly. But — only $s_1 = s_2 = s_3 = 0$ (occurring with probability $1/2^k$, k = number of $J(\alpha)$ gates) would give the desired simulation!

To deal with these unwanted X "errors" we will use commutation relations between our basic gates and X and Z gates e.g. a simple calculation shows (reading gate applications from left to right as is usual in circuit diagram pictures), that up to an (irrelevant) overall phase $e^{-i\alpha}$:

$$—\boxed{X}—\boxed{J(\alpha)}— \equiv —\boxed{J(-\alpha)}—\boxed{Z}—$$

The full list of relations that we'll need is: (easily verified, and reading gate applications from right to left now, as is usual in algebraic notation):

- $J_i(\alpha)\, X_i^{\,s} = e^{-i\alpha s}\, Z_i^{\,s}\, J_i\big((-1)^s \alpha\big)$ (COM1)
- $J_i(\alpha)\, Z_i^{\,s} = X_i^{\,s}\, J_i(\alpha)$ (COM2)
- $E_{ij}\, X_i^{\,s} = X_i^{\,s}\, Z_j^{\,s}\, E_{ij}$ (COM3)
- $E_{ij}\, Z_i^{\,s} = Z_i^{\,s}\, E_{ij}$ (COM4)

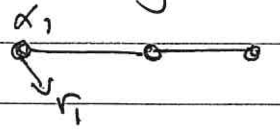
Henceforth we'll omit the (irrelevant) overall phase factor $e^{-i\alpha s}$ when using (COM1).
Note in particular that (COM1) leaves the angle dependent on s (as in the above picture) while E_{ij} propagates an X error on either line i, or j into and additional Z error on the other line (recalling also that E_{ij} is symmetric).

To illustrate how these relations help to deal with errors, consider a simpler circuit with just one qubit line (for the previous example with E_{ij} see exercise sheet 2):

$$|+\rangle —\boxed{J(\alpha_1)}—\boxed{J(\alpha_2)}— Z\text{ mmt},\ i$$

We first prepare the 3-qubit graph state

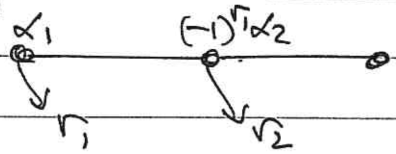
Measuring the first qubit we get

 i.e. we have done $|+\rangle - \boxed{J(\alpha_1)} - \boxed{X^{r_1}} -$

To deal with the unwanted X^{r_1} "error", before measuring the second qubit to apply $J(\alpha_2)$, we note from (COM1) that (up to phase):

$$- \boxed{X^{r_1}} - \boxed{J(\alpha_2)} - \quad \equiv \quad - \boxed{J((-1)^{r_1}\alpha_2)} - \boxed{Z^{r_1}} -$$

! So we adapt the sign of the second mmt angle to depend on the previous mmt result viz.:



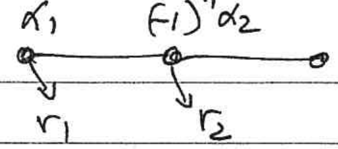
giving then, after this adapted second mmt:

$$|+\rangle - \boxed{J(\alpha_1)} - \boxed{X^{r_1}} - \boxed{J((-1)^{r_1}\alpha_2)} - \boxed{X^{r_2}} -$$

$$\equiv |+\rangle - \boxed{J(\alpha_1)} - \boxed{J(\alpha_2)} - \boxed{Z^{r_1}} - \boxed{X^{r_2}} -$$

- If we had a further $J(\alpha_3)$ gate we'd now need to adapt its angle for both X and Z errors. From (COM1) and (COM2) we see that in propagation across J, X turns into Z, and Z into X, and only X changes the sign of the angle. Thus the next angle would be adapted to $(-1)^{r_2}\alpha_3$, not depending on r_1.
- The order of X and Z on a line is irrelevant as
  $XZ = -ZX$ i.e. same up to overall phase (-1).
  Also multiple X's & Z's on a line can be collapsed using $X^2 = Z^2 = I$.

Now back to our simple example we have so far:



i.e. $|+\rangle$ — $\boxed{J(\alpha_1)}$ — $\boxed{J(\alpha_2)}$ — $\boxed{Z^{r_1}}$ — $\boxed{X^{r_2}}$ —

and it remains to do the final Z-mmt. Having moved all the errors to the end of the circuit (just before the Z mmts) they can now be dealt with by simply <u>re-interpreting</u> the results of the final actual Z-mmts, because the x's & z's have a very simple effect on Z-mmt outcomes —

- a Z gate does not affect the outcome or probability of a Z-mmt result viz.:

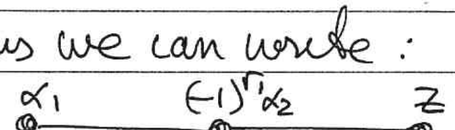
  if $|\psi\rangle = a|0\rangle + b|1\rangle$ then $Z|\psi\rangle = a|0\rangle - b|1\rangle$ and both have same $pr(0) = |a|^2$ $pr(1) = |b|^2$.

- an X gate simply interchanges the labels while leaving the probabilities the same viz:

  if $|\psi\rangle = a|0\rangle + b|1\rangle$ then $X|\psi\rangle = a|1\rangle + b|0\rangle$ so the probs are simply interchanged.

* So for each X^r error we just modify the seen Z mmt outcome i by $i \oplus r$.

Thus we can write:



and output $i \oplus r_2$

which is:

$|+\rangle$ — $\boxed{J(\alpha_1)}$ — $\boxed{J(\alpha_2)}$ — $\boxed{Z^{r_1}}$ — $\boxed{X^{r_2}}$ — Z mmt, i
and output $i \oplus r_2$

$\underset{\text{probs}}{\overset{\text{for output}}{\equiv}}$ $|+\rangle$ — $\boxed{J(\alpha_1)}$ — $\boxed{J(\alpha_2)}$ — Z mmt, i, output i.

as required!

In the literature the accumulating $X^a Z^b$ ($a, b = 0$ or 1) "errors" are sometimes called by-product operators.
Note that E's in a circuit also propagate these errors across to the second line involved via (COM3).

Logical depth of a measurement pattern.

Mmts can always be done from "left to right" (i.e. corresponding to actual order of J gates in C). But recall that the $M_i(\alpha)$ mmts on different qubits can be physically performed simultaneously if we know the angles α, since they are quantum operations on disjoint subsystems. This gives a novel (intrinsically quantum) way of parallelising a computation — any mmt pattern of an MQC process can be performed in layers (instead of left to right along $|\psi_G\rangle$):

layer 1: all mmts that require no adaptation
layer 2: all mmts adapted using outcomes from layer 1 only
layer 3: all mmts adapted using outcomes from layers 1 & 2 only .. etc.

The total number of layers (before the final Z-mmts which are always nonadaptive!) is called the logical depth of the computation.

Example: for our simple example above, logical depth = 2 (layer 1 ~ two end qubits, layer 2 ~ middle qubit)

- Somewhat paradoxically (!) the final Z-mmts giving the outputs can always be done first before the J gates and the Z mmt outcomes later just re-interpreted in the light of the emerging $M(\alpha)$-mmts done later.

Conclusion

The above MQC model allows us to reproduce the output result of any quantum circuit exactly, using only a sequence of single-qubit mmts on a graph state, and we get a new kind of computational parallelism. Any computation with poly(n) gates can be simulated using a graph state with poly(n) qubits, and a poly(n) amount of classical side-processing (which is only ever sums mod 2 of bit values) to deal with accumulating errors and re-interpretation of (final) Z-mmt outcomes.