Journal Club

James Wootton

29th November 2011

Surface code quantum computing by lattice surgery

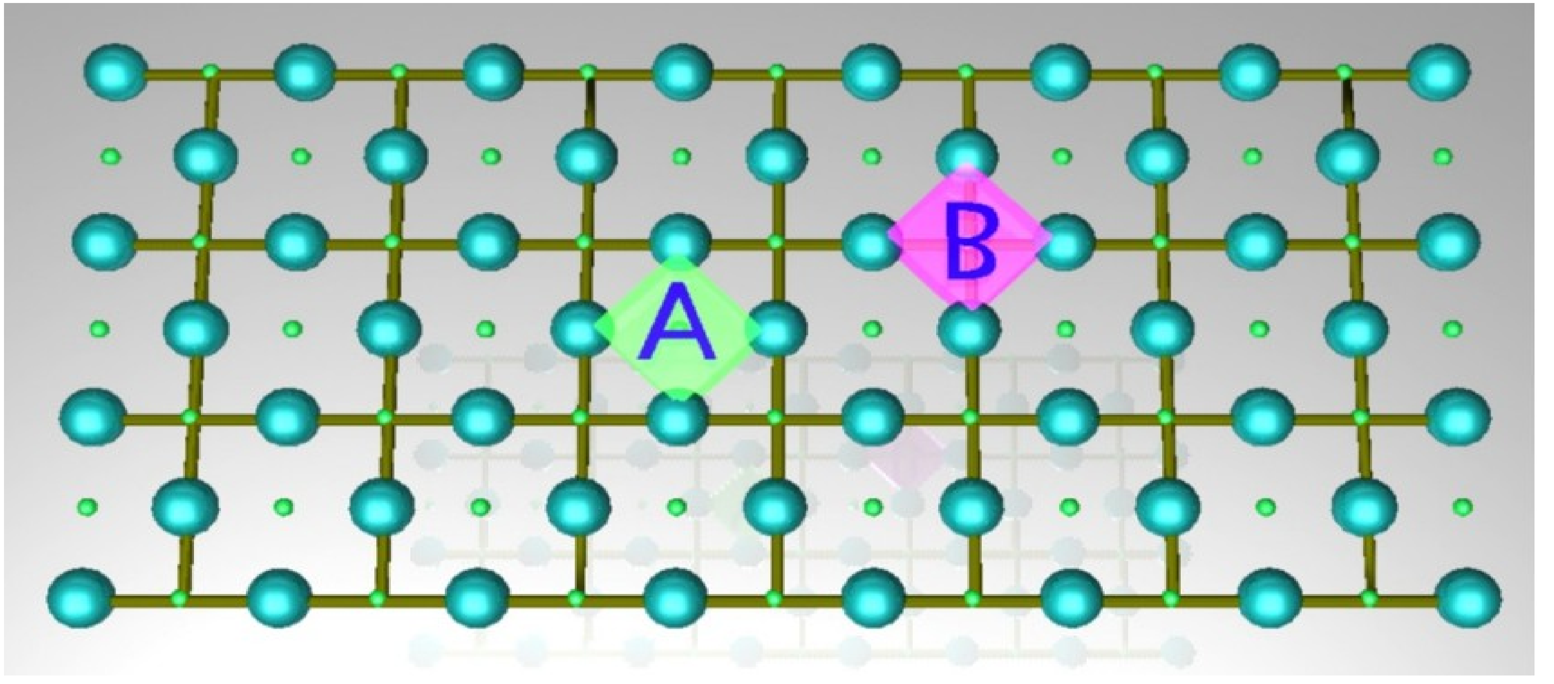
Clare Horsman[1], Austin G. Fowler[2], Simon Devitt[3] and Rodney Van Meter[4]

[1] Keio University Shonan Fujisawa Campus, Fujisawa, Kanagawa 252-0882, Japan
[2] CQC2T, School of Physics, University of Melbourne, VIC 3010, Australia
[3] National Institute for Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan.
[4] Faculty of Environment and Information Studies, Keio University, Fujisawa, Kanagawa 252-0882, Japan

E-mail: `clare@sfc.wide.ad.jp`

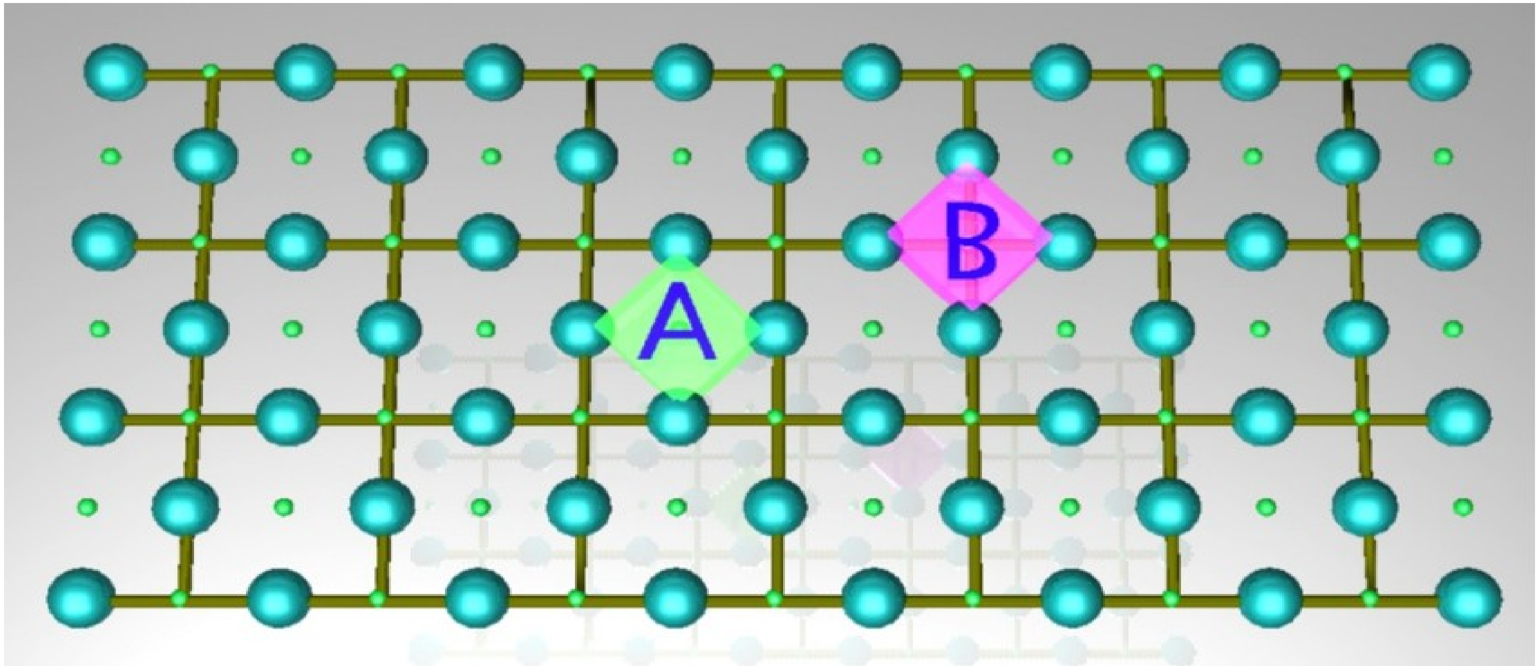
Abstract. In recent years, surface codes have become the preferred method for quantum error correction in large scale computational and communications architectures. Their comparatively high fault-tolerant thresholds and their natural 2-dimensional nearest neighbour (2DNN) structure make them an obvious choice for large scale designs in experimentally realistic systems. While fundamentally based on the toric code of Kitaev, there are many variants, two of which are the planar- and defect- based codes. Planar codes require fewer qubits to implement (for the same strength of error correction), but are restricted to encoding a single qubit of information. Interactions between encoded qubits are achieved via transversal operations, thus destroying the inherent 2DNN nature of the code. In this paper we introduce a new technique enabling the coupling of two planar codes without transversal operations, maintaining the 2DNN of the encoded computer. Our lattice surgery technique comprises splitting and merging planar code surfaces, and enables us to perform universal quantum computation (including magic state injection) while removing the need for braided logic in a strictly 2DNN design, and hence reduces the overall qubit resources for logic operations. We show how lattice surgery allows us to distribute encoded GHZ states in a more direct (and overhead friendly) manner, and how a demonstration of an encoded CNOT between two distance 3 logical states is possible with 53 physical qubits, half of that required in any other known construction in 2D.

Surface Code



- Introduced by Dennis et al. (2002), based on Kitaev's Toric Code (1997)
- Quantum error correcting code, used to encode one logical qubit
- Stabilizer operators $A_F = \prod_{i \in F} \sigma_i^z$ $B_V = \prod_{i \in V} \sigma_i^x$
- Can be measured using nearest neighbour interactions
- Outcome of -1 on a plaquette or vertex can be associated with an anyon

Surface Code



- Information encoded within 2 dimensional space

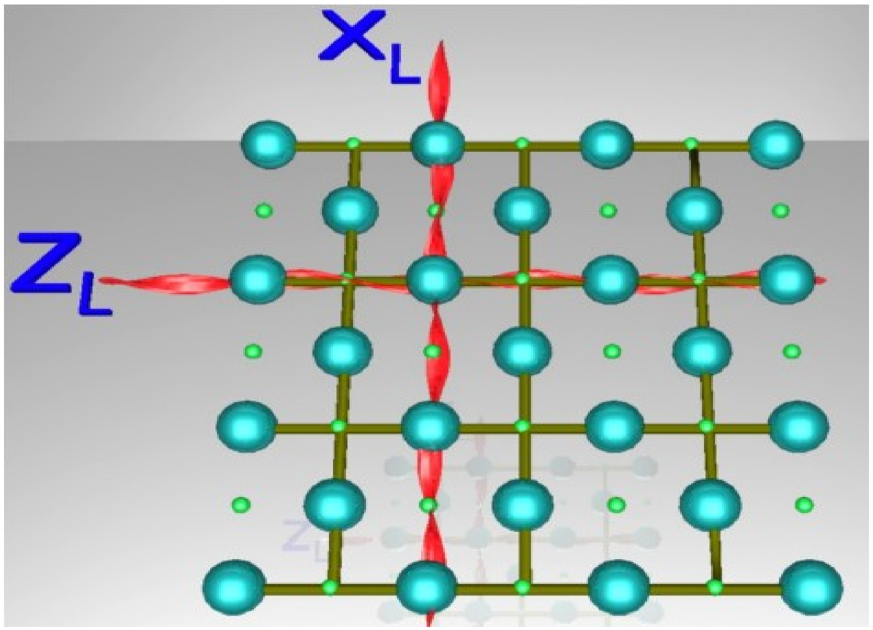
$$A_F|\psi\rangle = |\psi\rangle \qquad B_V|\psi\rangle = |\psi\rangle$$

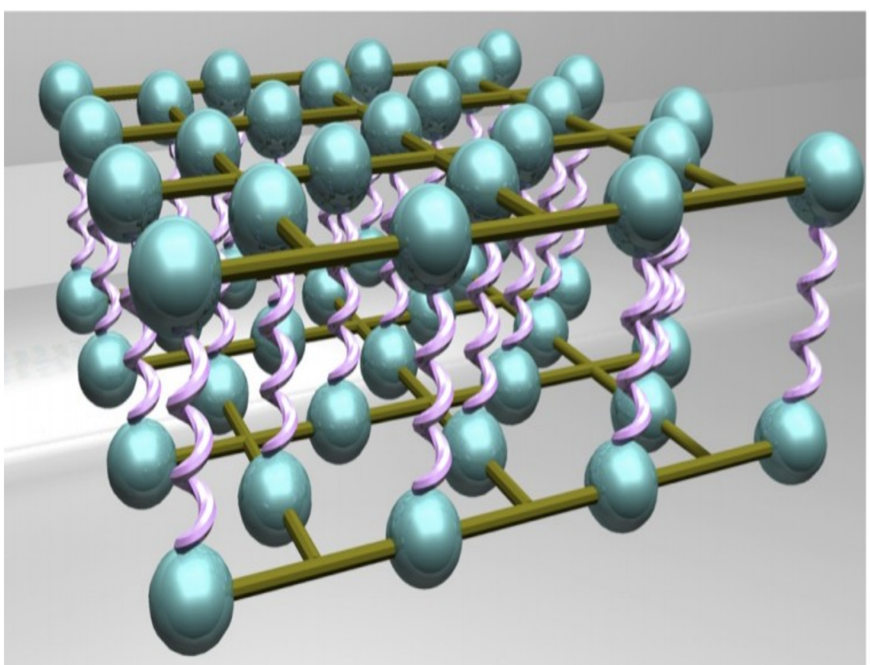
- This is the space with no anyons on any plaquette or vertex

- Performing operations on the code can create pairs of anyons

•These can be reannihilated, but care must be done to do this without causing logical errors

Surface Code





• Logical Pauli operators for logical qubits correspond to products of single spin Pauli operators along lines that traverse the code

• These can be interpreted as operators which create anyons pairs, and place them off opposite edges

• The logical |+> state is that where no vertex anyon exists off the left and right edges

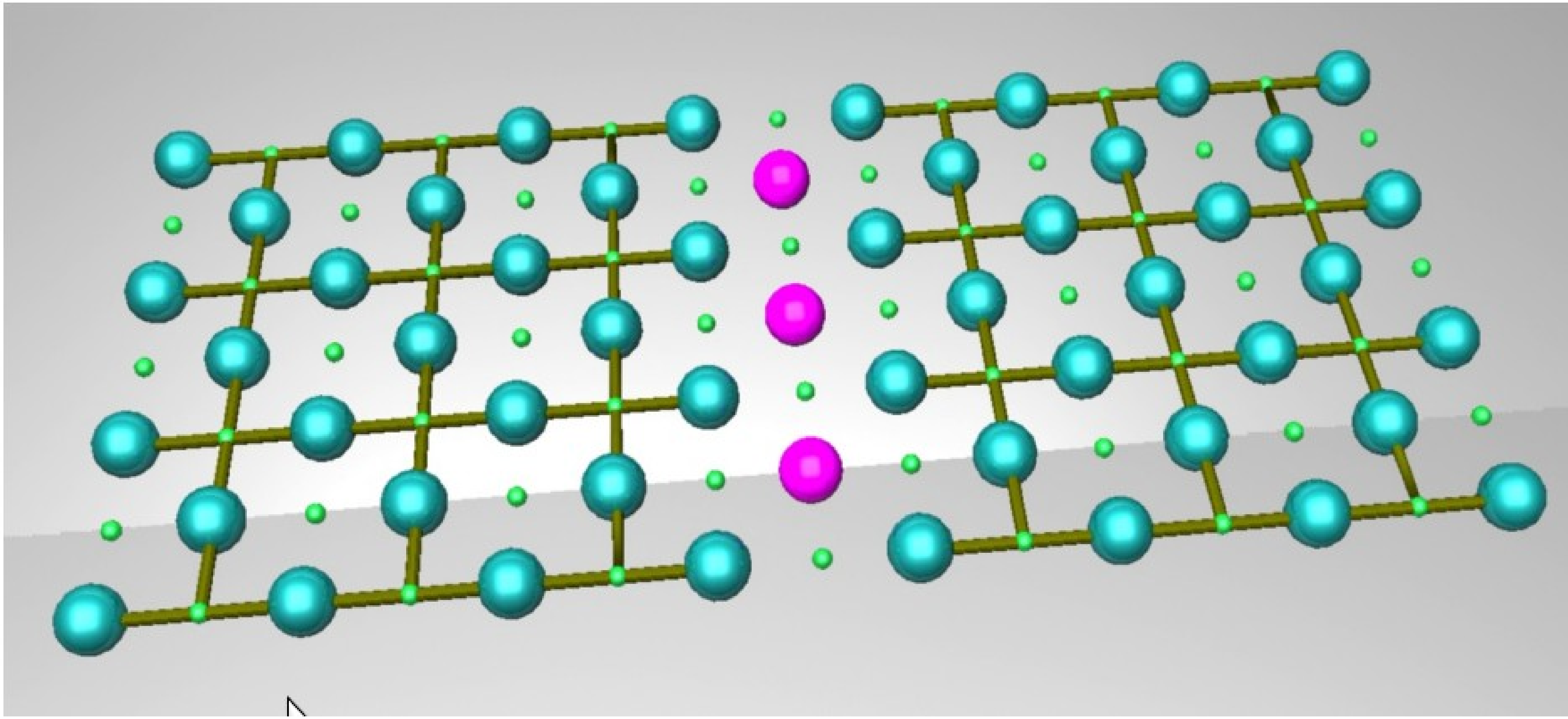
•The |-> state is that where there is a vertex anyon off each

• Single qubit operations and entangling operations must be performed transversally, but this can be hard to realize

• Defect based approaches are an alternative (Raussendorf et al. (2007)), but these need larger codes

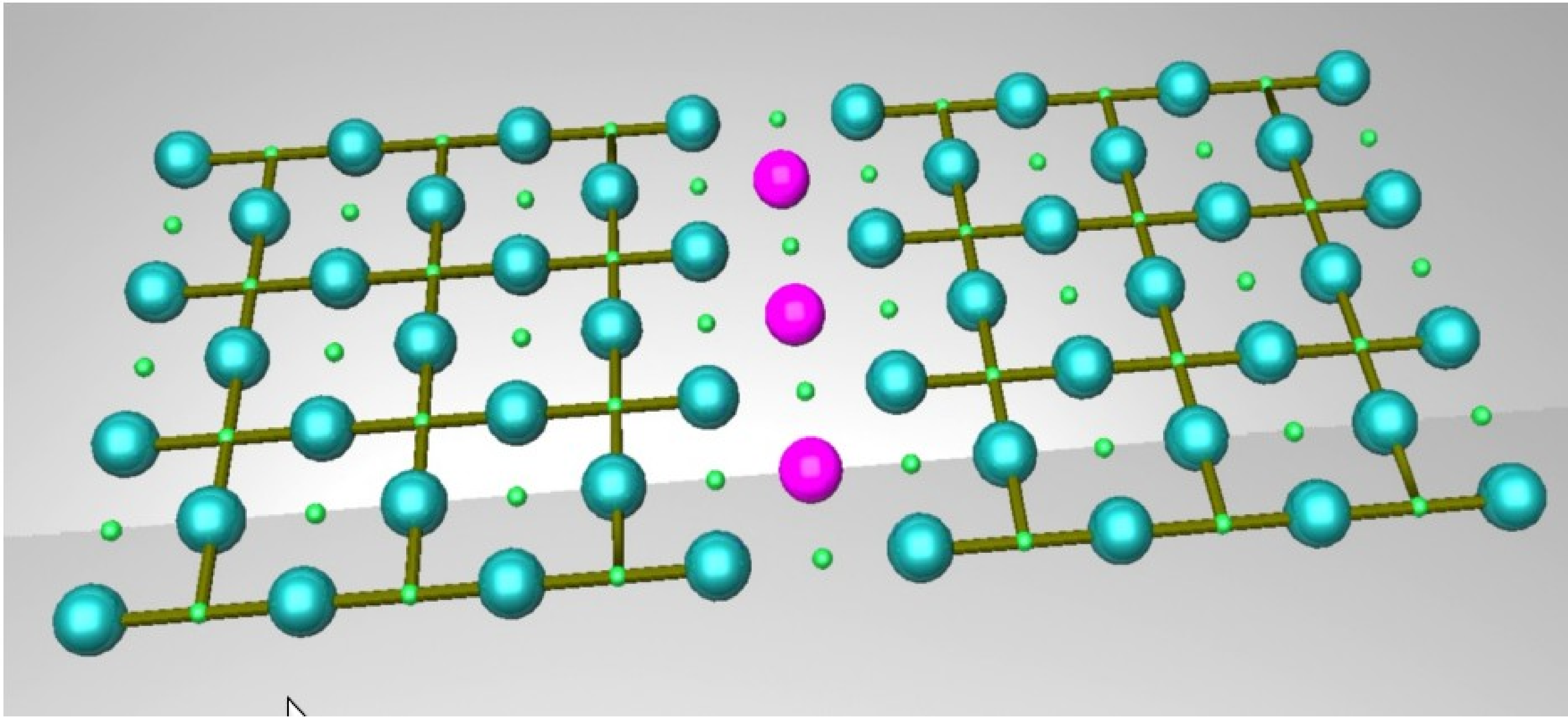
• Is there another way?

Lattice merging



- Lattice merging takes two stabilizer codes and makes one
- Can be done either along the 'smooth' or 'rough' edges
- For a rough merge, a line of ancilla spins in state |0> are placed along the join
- The newly formed vertex stabilizers are measured
- Each will yield +1 (vacuum) or -1 (anyon) with equal probability, except with a parity constraint

Lattice merging



• If both codes are initially in state |+> (or |->), there are no net anyons along the join. Stabilizer measurements must therefore yield an even number of anyons

• If one is in |+> and the other in |->, there is one anyon along the join. Stabilizer measurements must therefore yield an odd number

• Parity of anyon number therefore corresponds to a logical XX measurement

• Any spare anyon can be moved off the right side of the new code

Lattice merging

- In terms of the logical states, such a rough merge corresponds to:
  - Measurement of XX
  - Z on second qubit iff XX=-1

- Use M=0 to denote the XX=+1 outcome and M=1 to denote XX=-1

- Consider arbitrary states

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle = a|+\rangle + b|-\rangle \qquad |\phi\rangle = \alpha'|0\rangle + \beta'|1\rangle = a'|+\rangle + b'|-\rangle$$

- This process results in to the transformation

$$|\psi\rangle|\phi\rangle \rightarrow \alpha|\phi\rangle + (-1)^M \beta X|\phi\rangle = \alpha'|\psi\rangle + (-1)^M \beta' X|\psi\rangle$$

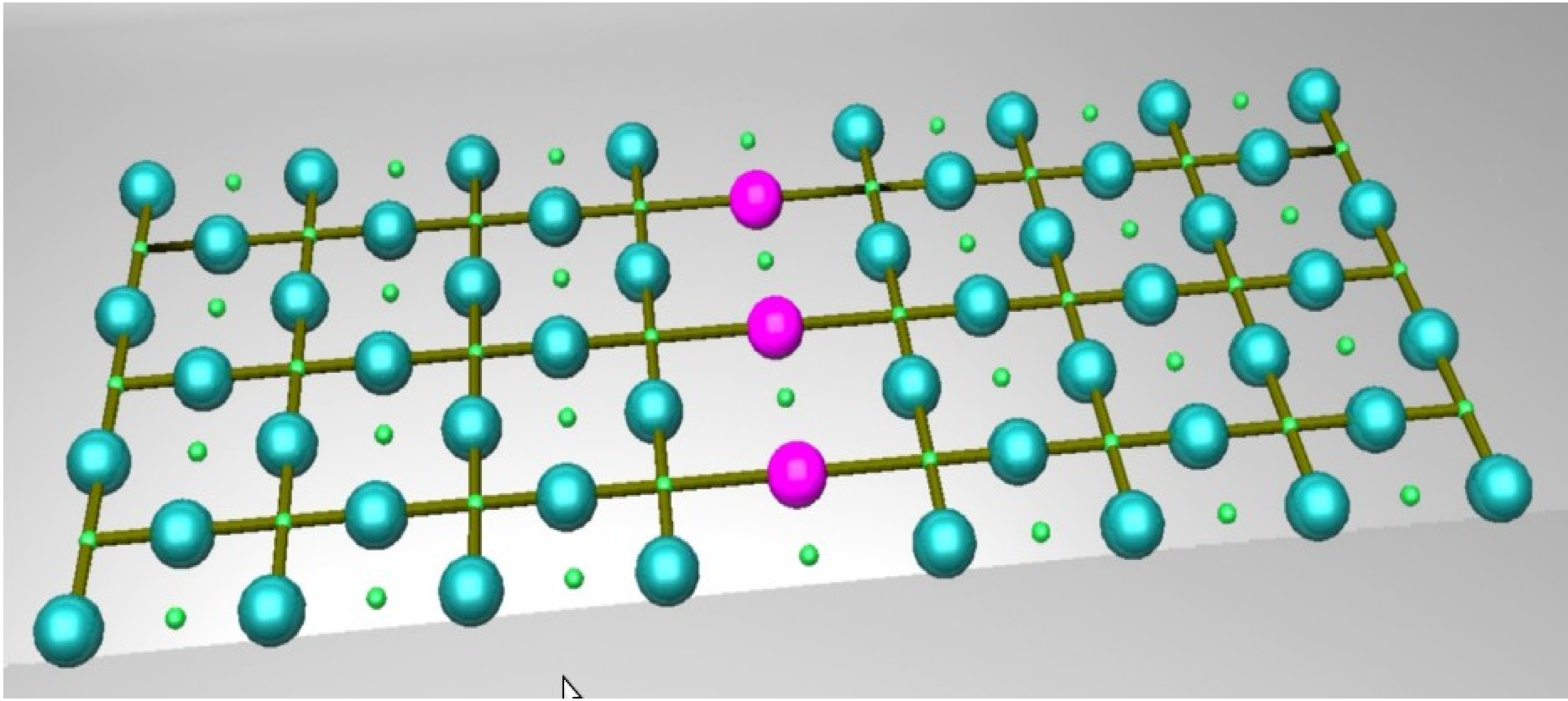
- In terms of the logical states, a smooth merge corresponds to:
  - Measurement of ZZ
  - X on second qubit iff ZZ=-1

- This process results in the transformation

$$|\psi\rangle|\phi\rangle \rightarrow a|\phi\rangle + (-1)^M b X|\phi\rangle = a'|\psi\rangle + (-1)^M b' X|\psi\rangle$$

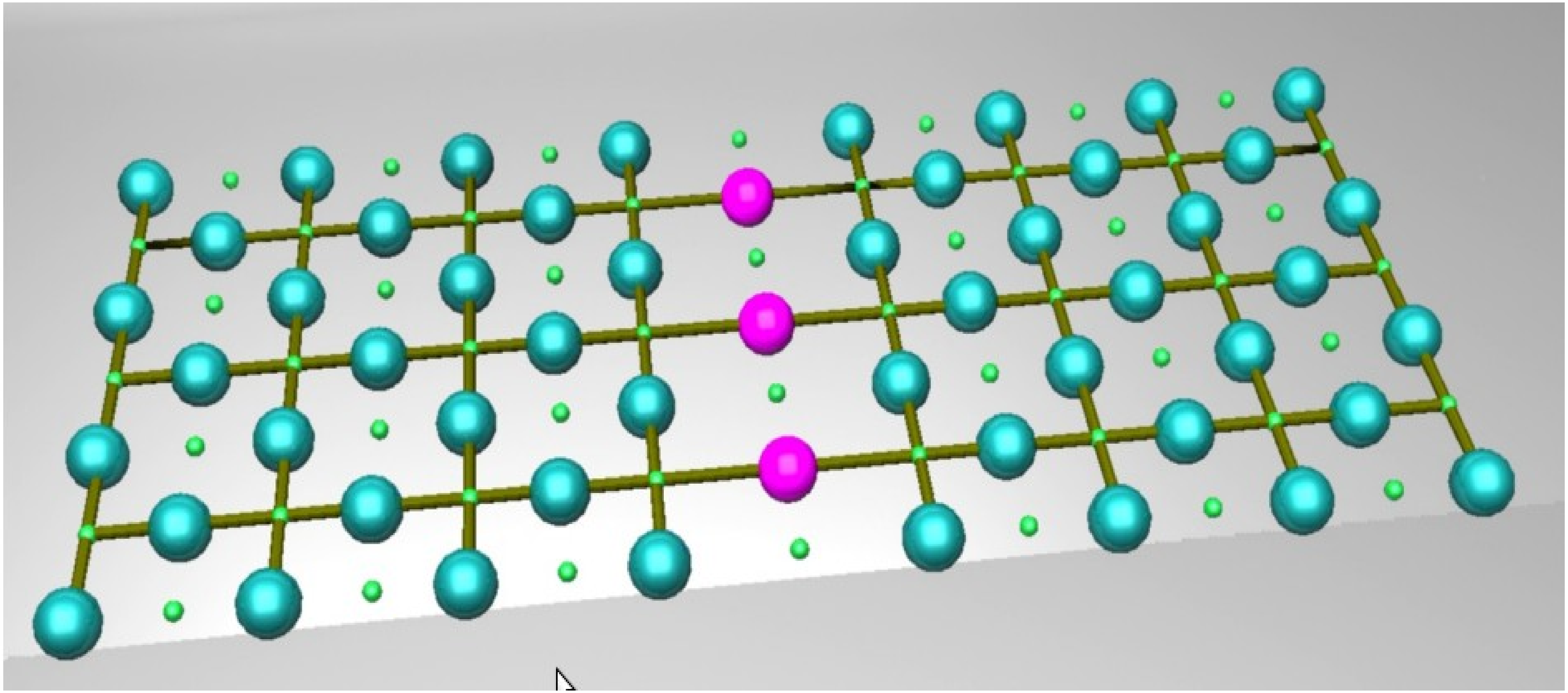
(this transformation does not seem to be correct)

Lattice splitting



- Lattice splitting takes one stabilizer code and makes two
- Can be done either along the 'smooth' or 'rough' edges
- For a smooth merge, a line of spins are measured in the |+>/|-> basis to remove them from the codes
- Border plaquettes will have random anyon occupation, and spare anyons can be disposed of in any way (but consistent for each code)

Lattice splitting



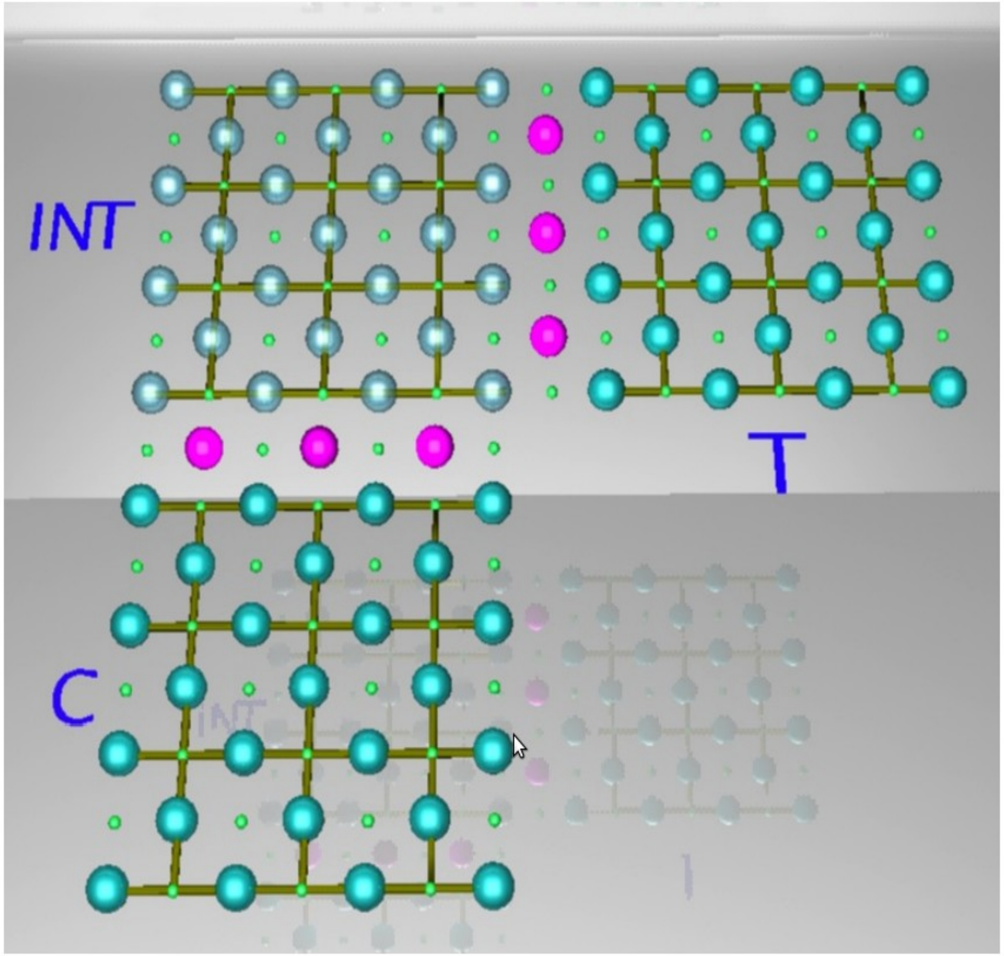
- Process results in two qubits with the same state in the |0>/|1> basis

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|00\rangle + \beta|11\rangle$$

- Similarly for a rough split

$$|\psi\rangle = a|+\rangle + b|-\rangle \rightarrow a|++\rangle + b|--\rangle$$

CNOT



- Merging and splitting can be used to implement a CNOT

- First the control qubit is smooth split

$$|\psi\rangle|0\rangle \rightarrow \alpha|00\rangle + \beta|11\rangle$$

- The ancilla is then rough merged with the target

$$\rightarrow \alpha|0\rangle|\phi\rangle + (-1)^M \beta|1\rangle(X|\phi\rangle)$$

$$|0\rangle|\phi\rangle \rightarrow |\phi\rangle$$
$$|1\rangle|\phi\rangle \rightarrow (-1)^M X|\phi\rangle$$

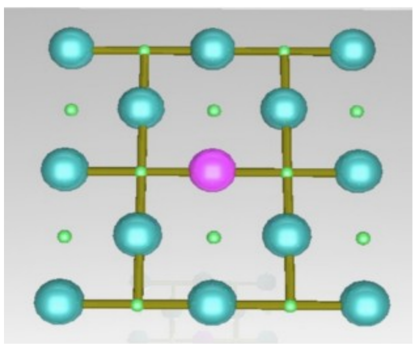
- Since M is known, the spurious -1 can be removed (if present) with a Z operation

- The result is a CNOT between the control and target qubits

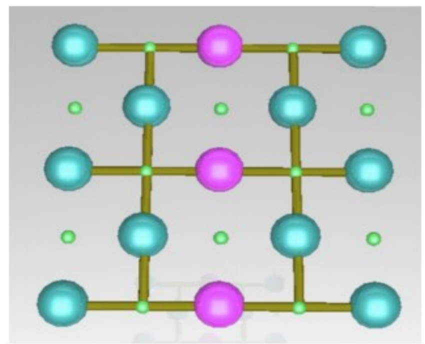
State injection

- For universal quantum computation, the ability to prepare arbitrary logical qubit states must be shown
- These can be used to perform arbitrary single spin rotations
- The means to map a state from a single spin to a planar code is shown below

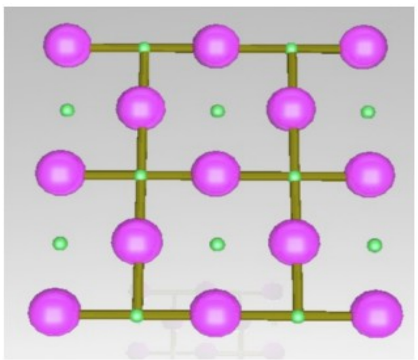
$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad |\psi\rangle = \alpha|000\rangle + \beta|111\rangle$$



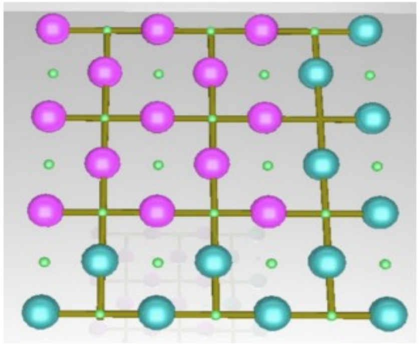
(a)



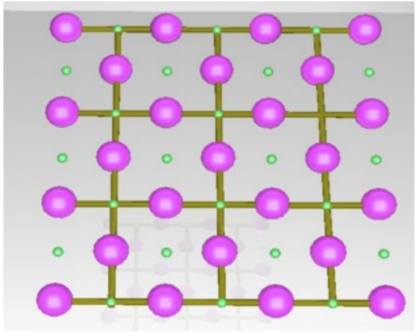
(b)



(c)



(d)



(e)

Conclusions

- A new way of performing operations on qubits stored in planar codes is proposed

- This removes the need for transversal interactions between codes

- It is also more efficient than defect based schemes, using less spins to achieve the same code distance

- However, issues of fault-tolerance are not fully explored
  - Do known thresholds apply?
  - How are errors removed from ancilla states? What is the overhead?

- So lattice surgery could be the new way to do planar code based quantum computation, but more needs to be known about its capabilities