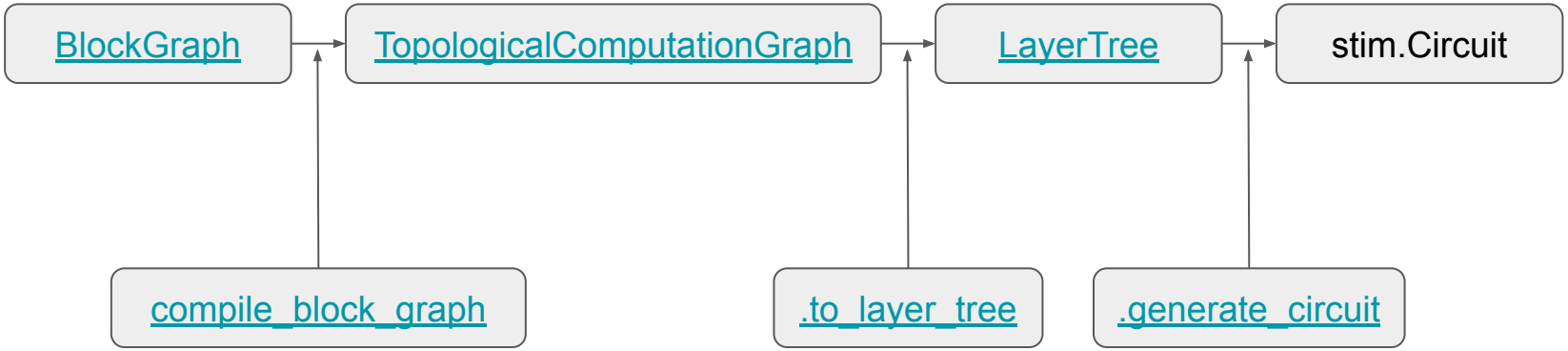
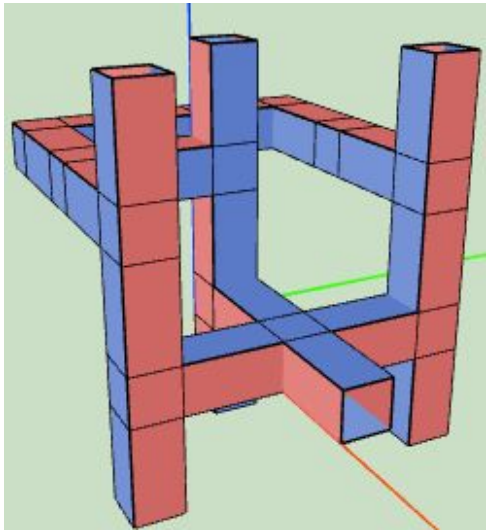
TQEC update

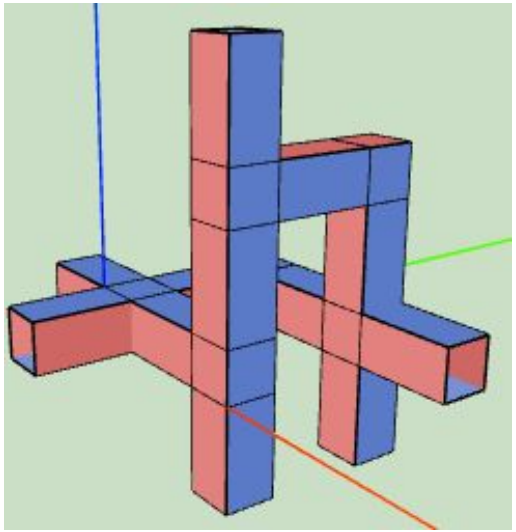
Compilation pipeline



BlockGraph

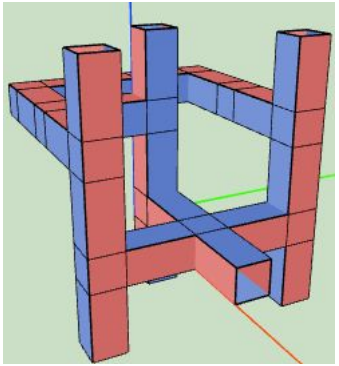
Did not change

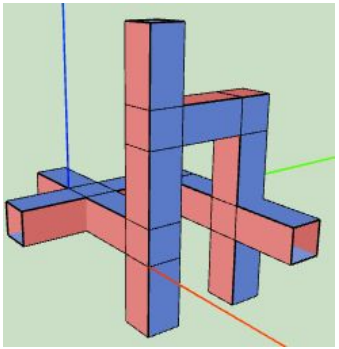




TopologicalComputationGraph

- Keeps the BlockGraph structure
- Replaces each cube by a Block
- Replaces (nearly) each pipe by a Block



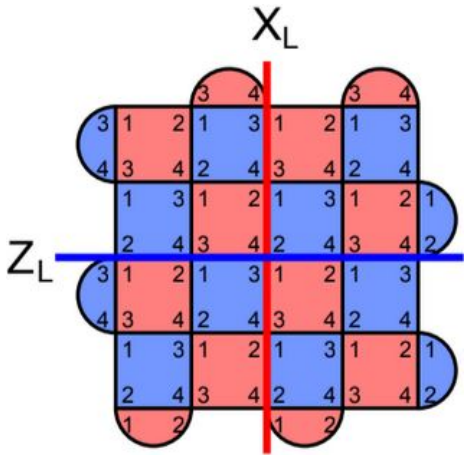


Layers and Blocks

There are BaseLayers and BaseComposedLayers.

A BaseLayer represents **one** QEC round. Can currently be instantiated with:

- RawCircuitLayer, using a raw user-provided and scalable quantum circuit,
- PlaquetteLayer, using the usual Template + Plaquettes.



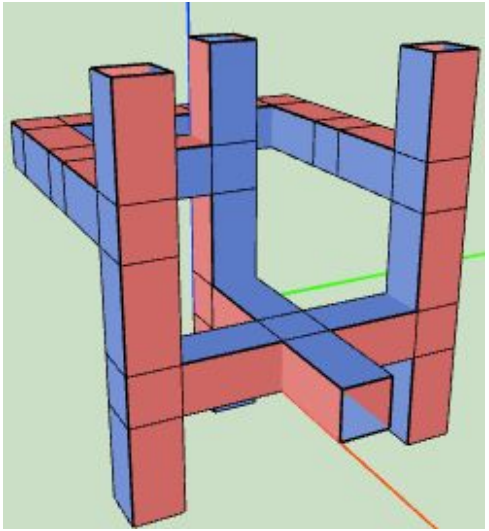
Layers and Blocks

A BaseComposedLayer represents more than one QEC round by composing BaseLayers and potentially other BaseComposedLayers. Can currently be:

- RepeatedLayer, representing repeated rounds (REPEAT block in stim),
- SequencedLayers, representing a sequence of layers applied one after the other.

Blocks are simply defined as an instance of SequencedLayers most of the time with 3 sub-layers: initialisation, memory, measurement.

LayerTree



Issue:

1. the instructions should be grouped by timeslice,
2. TopologicalComputationGraph does not check 1.

Solution: flatten out the Blocks and get global slices.

Done with a new BaseLayer:

- LayoutLayer, representing several BaseLayers on a 2D grid.

LayerTree – why this name?

The layers can be seen as a tree with:

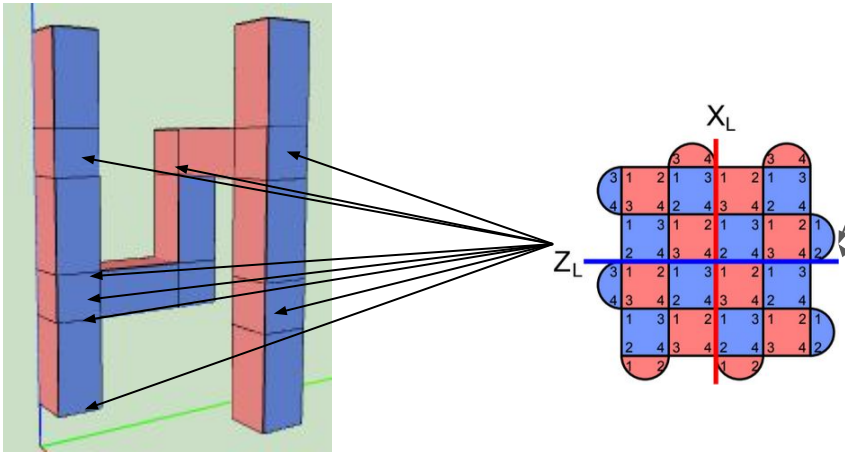
- instances of BaseLayer (i.e., LayoutLayer) being the leaves,
- instances of BaseComposedLayer being internal nodes.

Detector computation

- How is it done?
- Why a database?
- What can/should be improved?

Going into details: layers, templates and plaquettes

Using tqec.templates and tqec.plaquette:



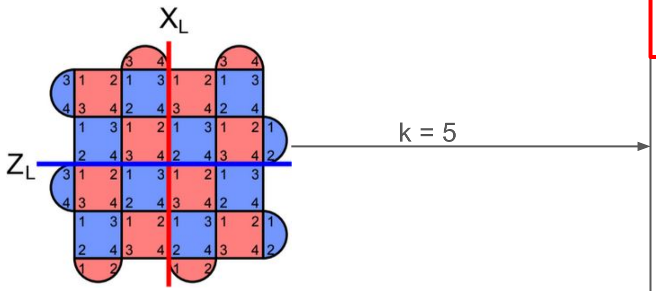

```
1   5   6   5   6   2
7   9  10   9  10  11
8  10   9  10   9  12
7   9  10   9  10  11
8  10   9  10   9  12
3  13  14  13  14   4
```

=> Template

```
{
   6: Plaquette(...),
   7: Plaquette(...),
   9: Plaquette(...),
   10: Plaquette(...),
   12: Plaquette(...),
   13: Plaquette(...),
   default: Plaquette.empty(),
}
```

=> Plaquettes

Subtemplates – Detectors are local in space

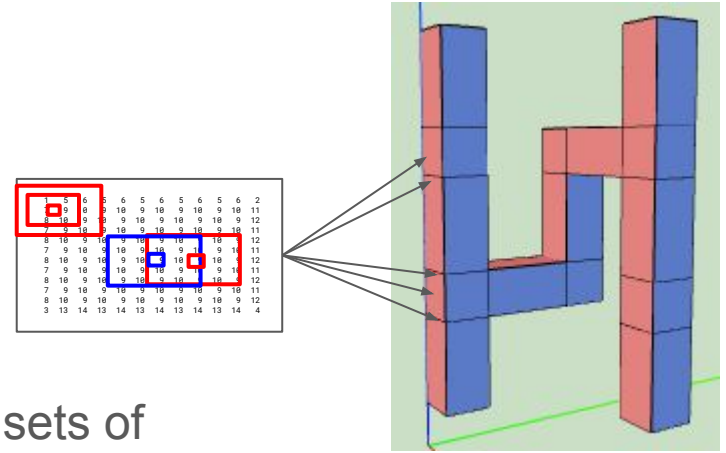


1	5	6	5	6	5	6	5	6	5	6	2
7	9	10	9	10	9	10	9	10	9	10	11
8	10	9	10	9	10	9	10	9	10	9	12
7	9	10	9	10	9	10	9	10	9	10	11
8	10	9	10	9	10	9	10	9	10	9	12
7	9	10	9	10	9	10	9	10	9	10	11
8	10	9	10	9	10	9	10	9	10	9	12
7	9	10	9	10	9	10	9	10	9	10	11
8	10	9	10	9	10	9	10	9	10	9	12
7	9	10	9	10	9	10	9	10	9	10	11
8	10	9	10	9	10	9	10	9	10	9	12
3	13	14	13	14	13	14	13	14	13	14	4

Subtemplates should be stacked

Most detectors are spawning **several** QEC rounds. We limit ourselves to **2** rounds for the moment.

That means that we need **2** subtemplates and **2** sets of Plaquettes to represent fully a region in which a detector may live.



Detector database key (_DetectorDatabaseKey)

To fully represent a “situation” we need:

- **n** subtemplates
- **n** Plaquettes instances to instantiate the subtemplates

For the moment **n <= 2**.

Improvements to the DetectorDatabase

- Add versioning to invalidate automatically,
- Lower the storage space needed to store it (to include it with tqec directly),
- Implement tools to view the database interactively?
- Allow hand-modification of the database?