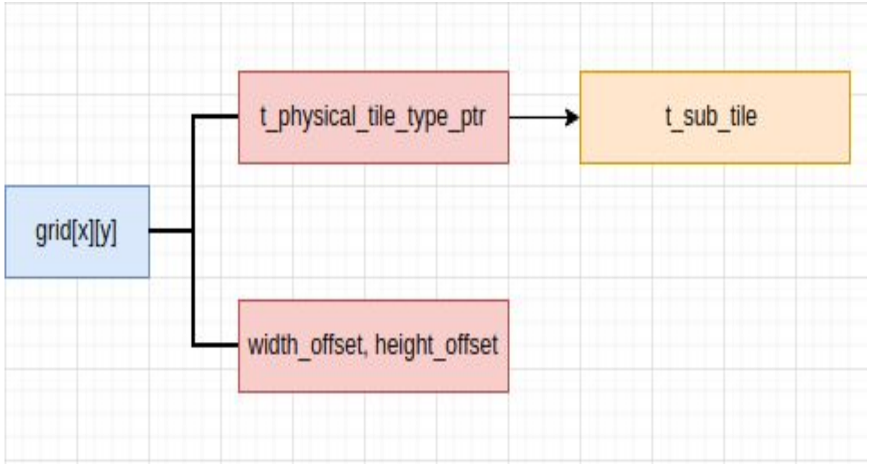
Modeling multi-die stack FPGAs

Sara Mahmoudi

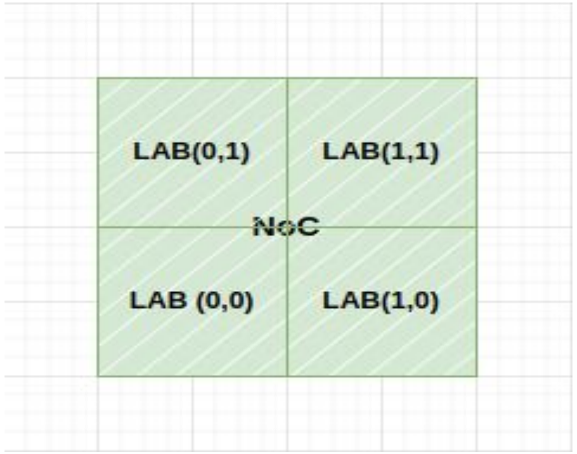
UNIVERSITY OF TORONTO

How VPR currently looks like

Current code data structures

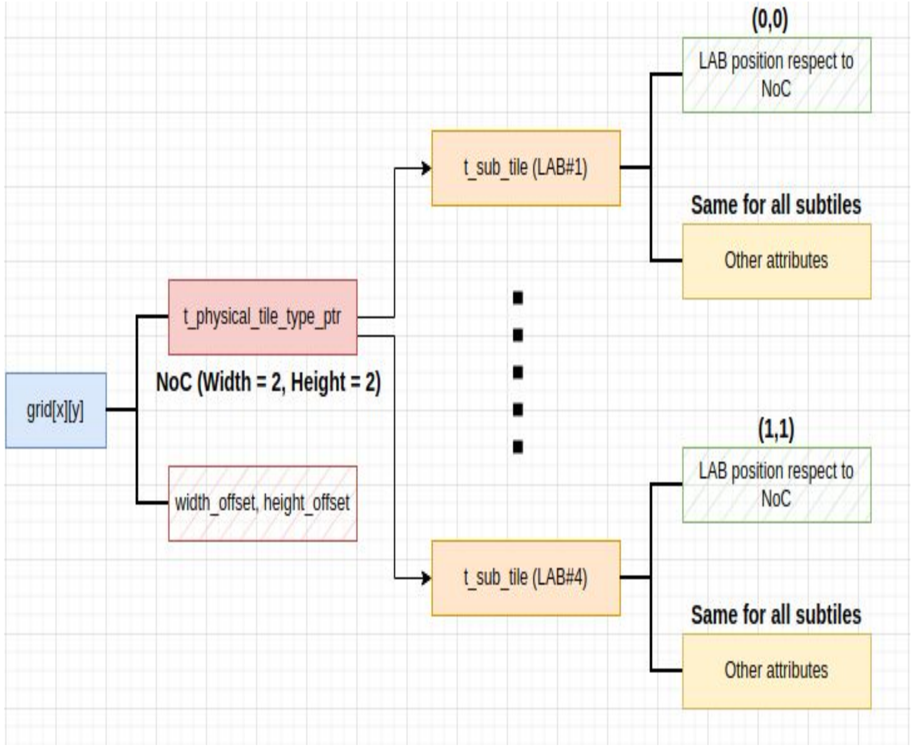


The physical blocks that we are trying to model



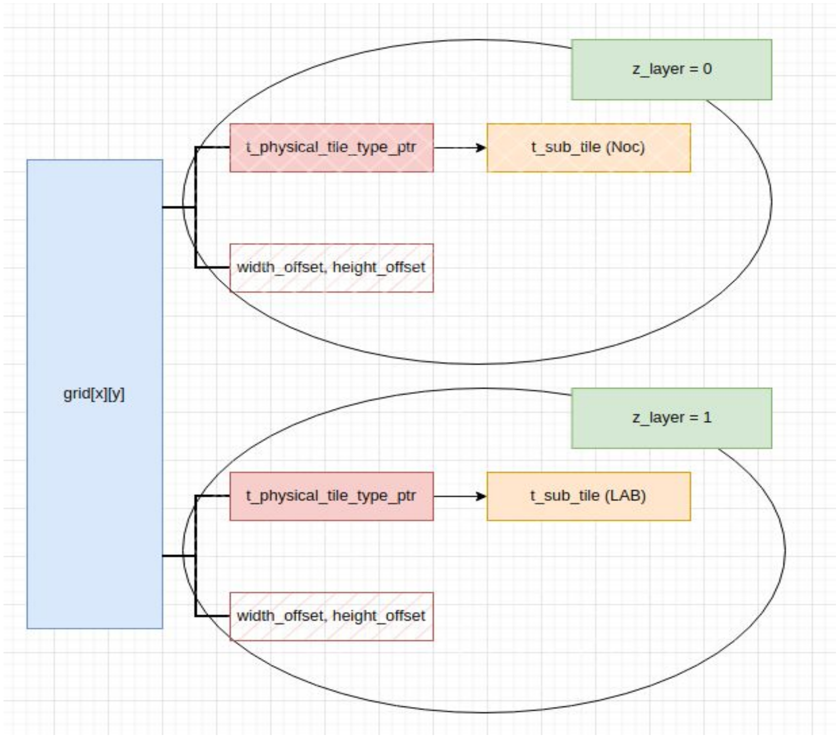
First solution

- Keep the current sub_tiles and tiles data structure (which can not currently model sub_tiles with different size on top of each other)
- Add width_offset, height_offset to sub_tile structure as well as the grid to specify anchor positions for LABs on top of a NoC block
- Very straightforward approach and easy to implement.
- Have to store multiple sub_tiles for LAB type since their anchor positions are different (which might be a bad solution if NoC size is large (10X10))



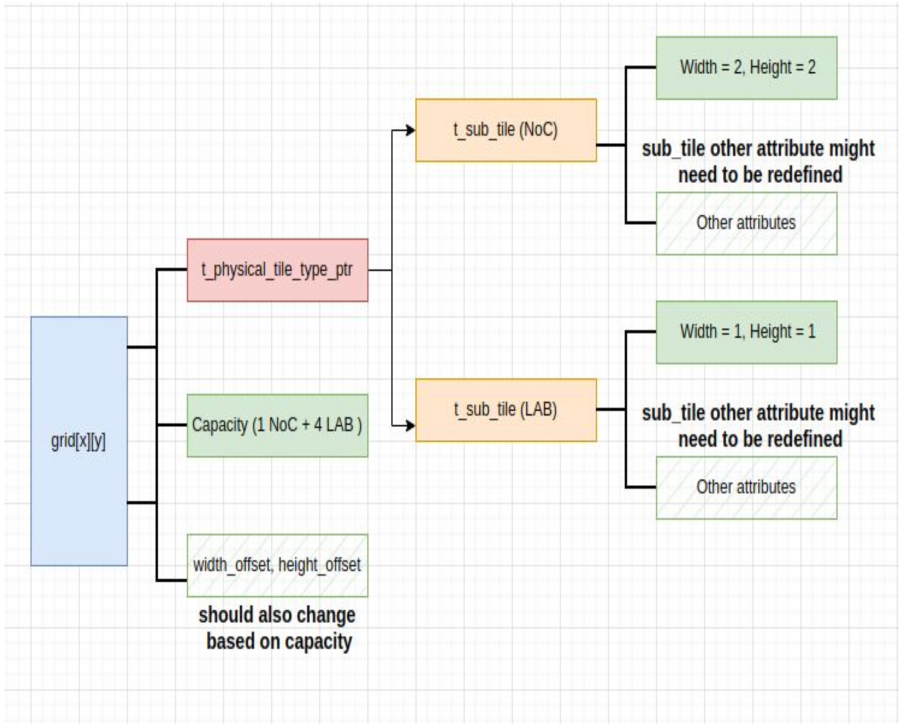
Second solution

- Keep the current sub_tiles and tiles data structure.
- Add an explicit z dimension to grid and a a layer capacity that would specify how many physical tiles were stacked at that point.
- Access to grid[x][y] will be converted to grid[x][y][z_layer].
- Existing code should work since we do not change the current sub_tiles and tiles definitions and meaning.
- A lot of code changes since grid is widely used in different part of VPR (RR graph, placement, and etc)
- Might become slightly slower since code has to work with multiple layers.



Third solution

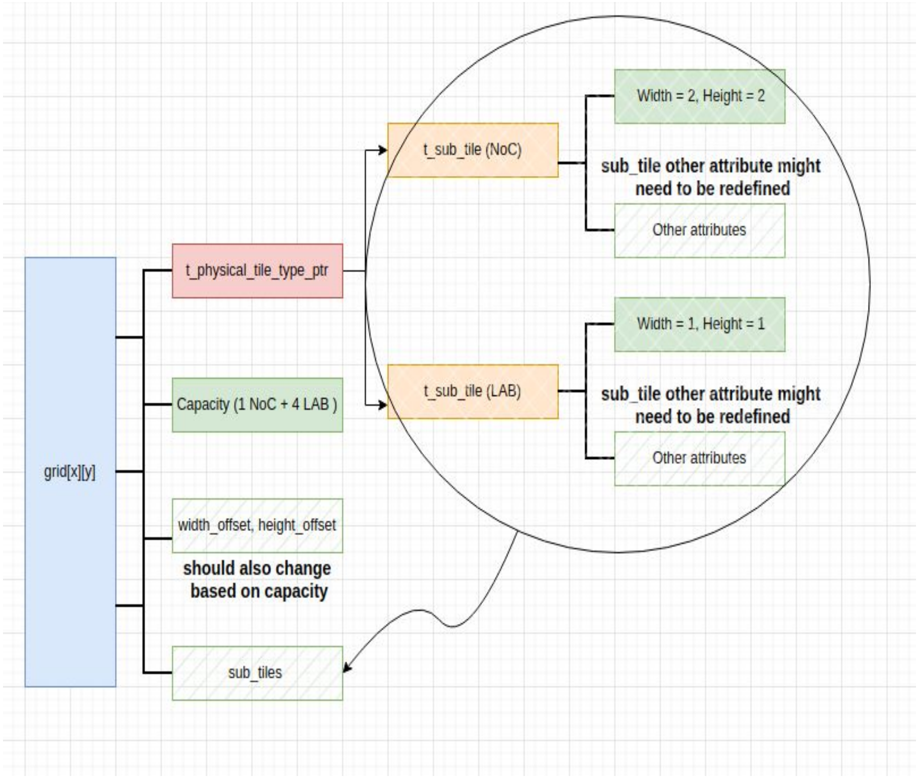
- Redefine sub_tile and possibly tile data structure.
- Add a capacity attribute to grid data structure, and also adding some attribute (e.g. width and height) to sub_tile itself.
- Keep multiple width and height offset for each sub_tile (the exact number would be capacity)
- Less code changes.
- Will probably break the existing code while re-imagining sub_tiles and tiles data structures (e.g. physical_tile currently groups multiple sub_tiles and allows their pins to be grouped and located around the composite physical_tile).



grid[x][y]
t_physical_tile_type_ptr
Capacity (1 NoC + 4 LAB)
width_offset, height_offset
should also change based on capacity
t_sub_tile (NoC)
t_sub_tile (LAB)
Width = 2, Height = 2
sub_tile other attribute might need to be redefined
Other attributes
Width = 1, Height = 1
sub_tile other attribute might need to be redefined
Other attributes

Fourth solution

- Move sub_tile data structure from physical_tile to grid.
- Add a capacity attribute to grid which shows how many sub_tiles we have at the current location.
- Keeps multiple sub_tiles with current attribute (plus some new attributes such as width and height), but no redefinition is required.
- Most of existing code should still be working since we do not change the sub_tile definition with a few changes.
- More code changes (but less than second solution) since grid is widely used throughout the VPR.


grid[x][y]
t_physical_tile_type_ptr
Capacity (1 NoC + 4 LAB)
width_offset, height_offset
should also change based on capacity
sub_tiles
t_sub_tile (NoC)
Width = 2, Height = 2
sub_tile other attribute might need to be redefined
Other attributes
t_sub_tile (LAB)
Width = 1, Height = 1
sub_tile other attribute might need to be redefined
Other attributes

Conclusion

- Adding an explicit z dimension (second solution) seems to be the best proposal.
- Better code readability and maintenance.
- No need to change any existing architecture files.
- Will not break any existing code since main data structures remain untouched.