FP Project Meeting 2013.03.25

Completed tasks:

1. Fixed bug on detectOverlap
2. Code cleanup and added test cases

To Do:

1. Optimize legalization process
2. Tight integration between ParquetFP and ArchFP
3. Framework to call external floorplaner (ex. ParquetFP)
   1. Block Packer (Optimal way to pack rectangles in a bigger one)
   2. ParquetFP
4. Way to implement wires and measuring wire length?
   1. Inter-connects components
   2. Center Coordinate
   3. HPW (Half Perimeter Wire Length)
      1. Rectangle
5. Legalization based on wirelength (Internal justify the decision)

FP Project Meeting 2013.02.28

Completed tasks:

1. Added TopBottomInversion
2. Added Overlap detection
3. Implemented Sort by decreasing area with legalization
4. Added the flag for legalization and all of its features

TO DO:

1. Expand the boundary based on the overlapping area.
2. Come out with the best expansion direction (TOP or RIGHT) and try it out in order.
3. Increase the area of the bounding box.
4. Always valid output.

ddCompleted Tasks:

1. Handled bagLayout (LeftRight/TopBottom)
2. Handled gridLayout (Mutliple elements)
3. ~~Changed the TopBottom order to BottomTop~~ (removed)
4. Added out-of-bound detection based on remaining area
   1. Different cases
      1. AR constraint (isGoodAR = false)
      2. Simply not enough space due to outer deadspace
      3. Note: Order of the components matters
   2. Question: Should we automatically swap the order in which the components are added or prompt user to do so in some out-of-bound cases?
   3. In other words, should we aim to build a more robust automating or user-customizable system?
5. Added optional overlap detection
   1. exception throwing
   2. not used at the moment

Questions:

1. How to properly design overall and suitable test cases? (+corner cases)
2. Does it matter if a few values from the original design are different?
   1. And it only happens when we optimize the compilation
   2. No difference when we compile normally

Next Week:

* Test basic 180 and Mirror cases.
* Deliver a working solution with non-recursive test cases for LeftRight/TopBottom/180/Mirror
* Deliver a working solution with non-recursive test cases for LeftRightMirror, TopBottomMirror and Center.

Later objective: Deliver an algorithm for recursive test cases