ISPD 2011 Contest Benchmark Format Description

Natarajan Viswanathan
IBM Research, Austin, TX

Introduction

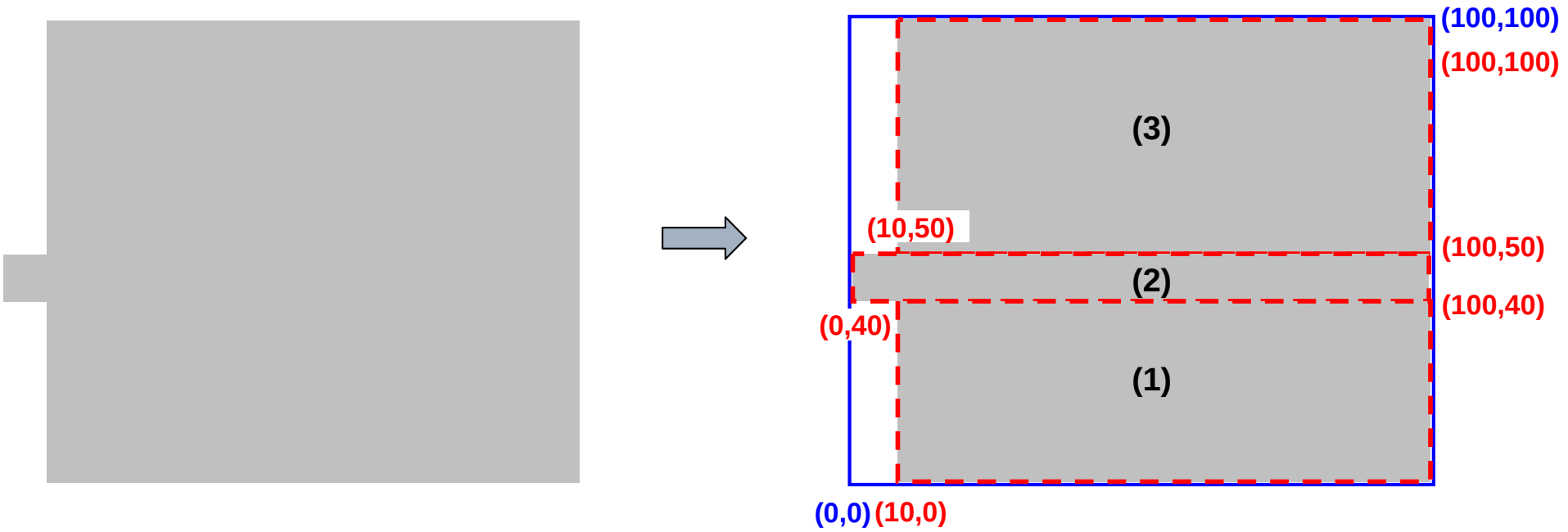
- Extend the Bookshelf format to handle routability-driven placement
- Benchmark files
  - circuit.aux
  - circuit.nodes
  - circuit.nets
  - circuit.wts
  - circuit.pl
  - circuit.scl
  - circuit.shapes
  - circuit.route

Original Files in Bookshelf format with some extensions

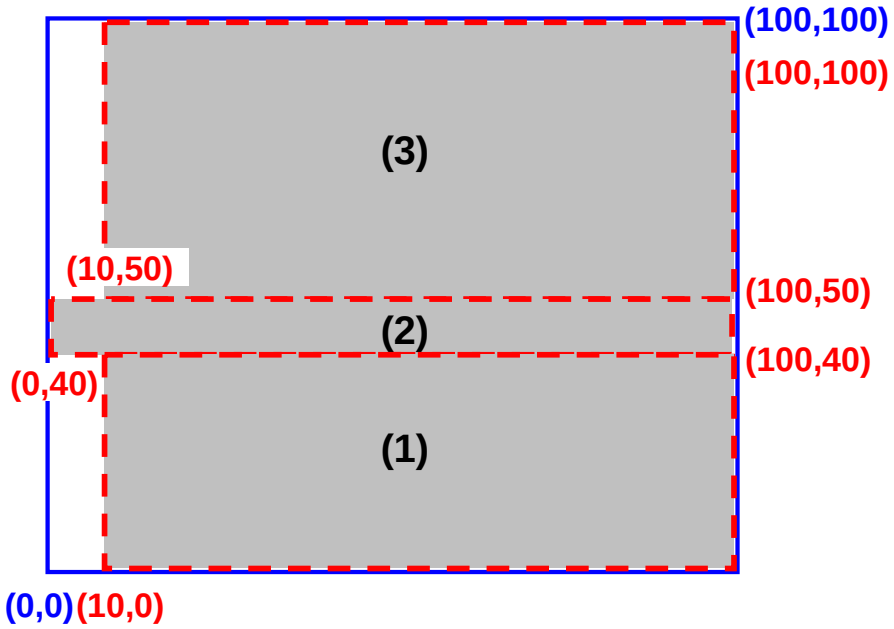
New Files with extensions for both placement and routing

New Features (1)

- Non-rectangular fixed objects
  - A subset of the fixed objects in the design are not rectangular
  - This affects placement density, routing capacity, etc.,
  - Represented as:
    - Enclosing rectangle (blue box in figure)
    - Set of rectangular component shapes (red dashed boxes in figure)



Non-rectangular Fixed Objects



Blue: Enclosing rectangle of non-rectangular object
Red: Set of component shapes (3 in number)

```
== circuit.nodes ==
#ObjName   width  height    movetype
   o25      100     100     terminal

== circuit.pl ==
#ObjName    llx    lly  : orientation  movetype
   o25       0      0   :       N       /FIXED

== circuit.shapes ==
#ObjName : NumComponentShapes
#Shape_id   llx   lly   width   height
o25 : 3
 Shape_0     10     0      90     40
 Shape_1      0    40     100     10
 Shape_2     10    50      90     50
```

Bookshelf Representation:

- circuit.nodes gives dimensions of the enclosing rectangle
- circuit.pl gives the lower-left coordinate of the enclosing rectangle
- circuit.nets gives pin-offsets from the center of the enclosing rectangle
- New file circuit.shapes gives the component shape definitions for the non-rectangular object

New Features (2)

- RLM Pins on "terminal_NI" objects
  - RLM pins are fixed pins that reside on a metal layer above the metal layer(s) that are used within standard-cells for pins/internal routing
  - **For placement**, the corresponding objects are:
    - Fixed (terminal_NI objects)
    - Appear to reside "above" the placement image
      - In other words, standard-cells can be placed "below" the terminal_NI objects without resulting in an overlap
  - **For routing**, the pin(s) associated with the terminal_NI objects will be on a higher metal layer as opposed to M1

RLM Pins (terminal_NI Objects)

```
== circuit.nodes ==
#ObjName   width  height   movetype
   p25       1      1      terminal_NI

== circuit.pl ==
#ObjName    llx    lly  : orientation  movetype
   p25       30     30  :       N       /FIXED_NI

== circuit.route ==
NumNiTerminals  :  NumTerminalNIObjects

#Metal layer for ALL the pins on an object
#ObjName   Layer_ID
   p25        3
```

Bookshelf Representation :

- New movetype terminal_NI in circuit.nodes file (overlap is allowed with associated object)
  - Represented as FIXED_NI in circuit.pl file
- New file circuit.route gives the metal layer for all the pins on the terminal_NI objects
  - The pins for any object not given in this section of circuit.route will be on layer 1

New Features (3)

- 9 metal layer stack for routing
- No routing resources on M1

```
VerticalCapacity   : 0 80  0 80  0 80  0  80  0
HorizontalCapacity : 0  0 80  0 80  0 80   0 80
```

- 2x and 4x wire width and spacing on higher metal layers

```
MinWireWidth   : 1 1 1 1 2 2 2 4 4
MinWireSpacing : 1 1 1 1 2 2 2 4 4
```

Detailed Description of Benchmark Files

- Benchmark Files
  - circuit.aux
  - circuit.nodes
  - circuit.nets
  - circuit.pl
  - circuit.scl
  - circuit.shapes
  - circuit.route
  - circuit.wts
- For additional information:
  - http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/Placement/plFormats.html
  - http://archive.sigda.org/ispd2008/contests/ispd08rc.html

circuit.aux

- Auxiliary file listing the files that need to be read by the placer/router

- Single line giving all the file names

```
RowBasedPlacement : pnr18.nodes pnr18.nets pnr18.wts
pnr18.pl  pnr18.scl  pnr18.shapes  pnr18.route
```

circuit.nodes

- NumNodes : Total number of nodes (movable + fixed)
- NumTerminals : Number of terminal (fixed) nodes
  - NumTerminals = #terminal + #terminal_NI
- For each node:

  `node_name  width  height  movetype`
  - If there is no movetype, then the node is movable

```
UCLA nodes 1.0
# File header with version information, etc.,
# Anything following “#” is a comment, and should be ignored

NumNodes      :  5
NumTerminals  :  2

    o0     4      9                    # movable node
    o1     4      9
    o2    24      9
    o3   414   2007   terminal         # terminal node (fixed node)
    p0     1      1   terminal_NI      # terminal_NI node (fixed node, but overlap is
                                          allowed with this node)
```

circuit.nets

- NumNets : Total number of nets
- NumPins : Total number of pins in the netlist
- For each net:

  `NetDegree : degree_of_this_net  net_name`

  `node_name  pin_dir : pin_xoffset  pin_yoffset`
- Pin offsets are from the center of the node

```
UCLA nets 1.0
# File header with version information, etc.,
# Anything following “#” is a comment, and should be ignored

NumNets  :  2
NumPins  :  5

NetDegree  :  3    n0
        o0   I  :      0.0000     -1.5000
        o1   I  :     -5.0000      0.5000
        p1   I  :      0.0000      0.0000
NetDegree  :  2    n1
        o3   O  :     10.5000     -1.5000
        o2   I  :     -1.0000      0.5000
```

circuit.pl

- Gives the coordinates for each node
- For each node:

```
node_name  lowerleft_xcoordinate  lowerleft_ycoordinate :
orientation  movetype
```

- Orientation of all the nodes will always be N (default)

```
UCLA pl 1.0
# File header with version information, etc.,
# Anything following "#" is a comment, and should be ignored

# node_name  ll_xcoord  ll_ycoord     orientation   movetype
    o0           40          90    :       N
    o1           35         117    :       N
    o2           24           9    :       N
    o3         7831        7452    :       N         /FIXED
    p0         1215        7047    :       N         /FIXED_NI
```

circuit.scl

- NumRows : Number of circuit rows for placement
- Description of each row
  - Coordinate : y-coordinate of the bottom edge of the row
  - Height : row-height
  - Sitewidth / Sitespacing : width / spacing of each placement site
  - SubrowOrigin : x-coordinate of the left edge of the subrow
  - NumSites : number of placement sites in this subrow
  - Hence, x-coordinate of the right edge of the subrow = SubrowOrigin + NumSites*Sitespacing

```
UCLA scl 1.0
# File header with version information, etc.,

NumRows  :  1

CoreRow Horizontal
   Coordinate     :  18
   Height         :  9
   Sitewidth      :  1
   Sitespacing    :  1
   Siteorient     :  N
   Sitesymmetry   :  Y
   SubrowOrigin   :  18          NumSites  :  11605
End
```

circuit.shapes

- Gives the component shapes for non-rectangular nodes
- Any node not in this file is a regular rectangular node
- NumNonRectangularNodes : Number of non-rectangular nodes
- For each node:

```
node_name   :  number_of_component_shapes
  shape_id  lowerleft_xcoord  lowerleft_ycoord  width  height
```

```
shapes 1.0
# File header with version information, etc.,

NumNonRectangularNodes  :  2

o25 : 3                          # Non-rectangular node with three component shapes
 Shape_0   10    0    90  40
 Shape_1    0   40   100  10
 Shape_2   10   50    90  50
o32 : 4
 Shape_0   30   2259   963  9
 Shape_1   30   2268  1024  9
 Shape_2   30   2277  1024  9
 Shape_3   30   2286   963  9
```

circuit.route (1)

- Gives information for routing
- Similar to ISPD 2008 routing contest format
  - http://archive.sigda.org/ispd2008/contests/ispd08rc.html

- Header Section

```
Grid :                num_x_grids  num_y_grids  num_layers
VerticalCapacity :    vertical capacity on each layer
HorizontalCapacity :  horizontal capacity on each layer
MinWireWidth :        minimum metal width on each layer
MinWireSpacing :      minimum spacing on each layer
ViaSpacing :          via spacing per layer
GridOrigin :          grid_lowerleft_x  grid_lowerleft_y
TileSize :            tile_width  tile_height
BlockagePorosity :    porosity for routing blockages
                      (a value of zero implies the blockage
                       completely blocks overlapping tracks)
```

circuit.route (2)

- Terminal_NI section
  - NumNiTerminals : Number of terminal_NI nodes
  - For each node:

    ```
    node_name    layer_id_for_all_pins
    ```

- Blockage Section
  - NumBlockageNodes : Number of blockage nodes
  - For each blockage:

    ```
    node_name  num_blocked_layers  list_of_blocked_layers
    ```

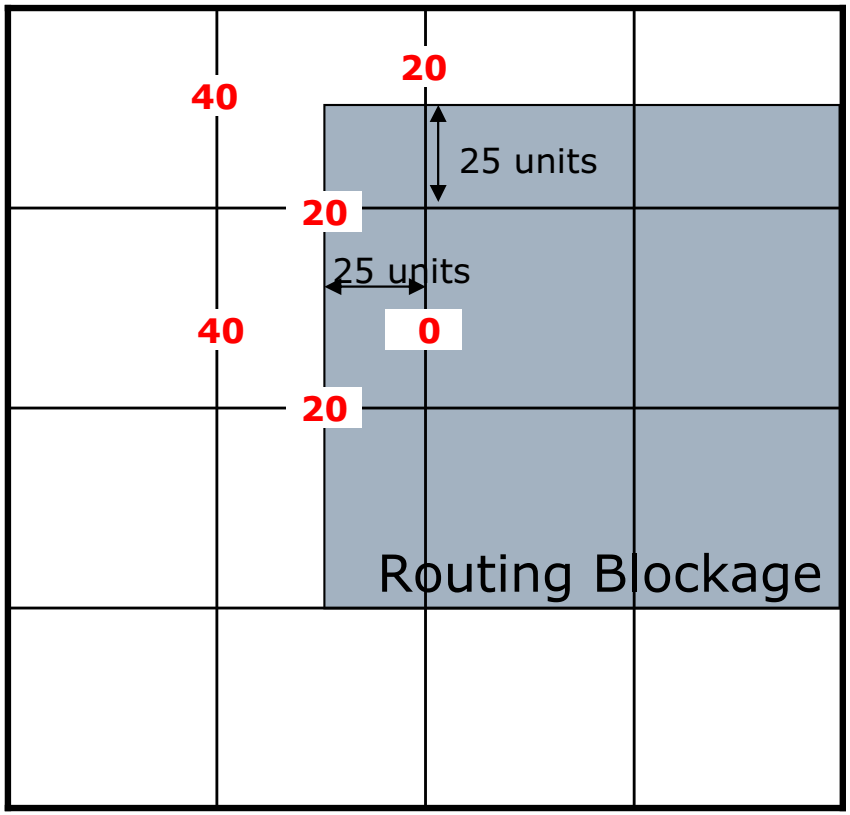
  - The tiles overlapping with a blockage can be determined using placement information from the other files in the benchmark

circuit.route (3)

```
route 1.0
# File header with version information, etc.,

Grid               :  304  403  9
VerticalCapacity   :  0  80   0  80   0  80   0  80   0
HorizontalCapacity :  0   0  80   0  80   0  80   0  80
MinWireWidth       :  1  1  1  1  2  2  2  4  4
MinWireSpacing     :  1  1  1  1  2  2  2  4  4
ViaSpacing         :  0  0  0  0  0  0  0  0  0
GridOrigin         :  18  18
TileSize           :  40  40
BlockagePorosity   :  0

NumNiTerminals  :  2

  p0     4      # all the pins belonging to p0/p1 are on layer 4 for routing
  p1     4

NumBlockageNodes  :  2

  o44  4  1 2 3 4   # o44/o2407 block 4 metal layers within all the routing
o2407  4  1 2 3 4   # tiles that they overlap. These are layers 1 to 4.
```

Example Routing Blockage Map

- One method to construct a blockage map is given below



Max H Capacity : 40 tracks
Max V Capacity : 40 tracks
Tile Width : 50 units
Tile Height : 50 units

Values in Red are the actual capacities of the edges

circuit.wts

- Currently unused
  - All nets have the same net-weight