3D Field Programmable Gate Arrays

Presentation by Nilangshu
Problems faced by current FPGAs – Interconnect delay

As process size shrinks, transistor delay exceeds gate (logic) delay. Thus the routing fabric dominates overall delay in the FPGA.
FPGA Interconnect issues

- Over 90% of FPGA logic area penalty is due to programmable interconnect
- Performance and power penalty are a direct result of the area
- Transistor as programmable interconnect doesn't scale well

<table>
<thead>
<tr>
<th>Delay Ratio</th>
<th>@100nm</th>
<th>@35nm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing (1mm)</td>
<td>30ps</td>
<td>100ps</td>
</tr>
<tr>
<td>Transistor</td>
<td>5ps</td>
<td>1ps</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power Ratio</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing</td>
<td>5</td>
<td>30</td>
</tr>
<tr>
<td>Transistors</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- Interconnect speed needs to keep up with gate speed in order for denser FPGAs to be economically viable
- In order to reduce interconnect, some FPGAs have started looking at higher granularity – 6-LUT designs as opposed to the traditional 4-LUT design
FPGAs' market share is dwarfed by ASICs because the latter offers higher computational density and lower routing delays.
The FPGA industry hasn't seen a major growth in the last few years. This is mainly due to the inability of the FPGA to substitute ASIC devices.
FPGA – ASIC gap

- FPGA cost-speed-power make it unattractive for many high volume applications
  - FPGA penalty is extremely high
    - Gate Density (ASIC : FPGA): 1: 20-40
    - Power (ASIC : FPGA): 1: 9-12
    - Speed (ASIC : FPGA): 1: 2-4

- Old Process ASIC is preferable in many applications
  - Density ~1:32=> 5 process generations (45nm FPGA ~ 0.25μ ASIC)
    - => 3-4 gen. older ASIC process (0.13μ-0.18μ) is more competitive
    - => 3-4 gen. older ASIC process fab is fully deprecated and hence cheaper

- ASSP + Software in many cases provide a better alternative
Why 3D FPGAs?

- Provide greater flexibility during placement and routing
- Decreased interconnect length, hence shorter propagation delays
- Shorter interconnect length and elimination of IO pins aid in lowering power consumption.
- Logic density is higher than 2D FPGAs
- Might give the much needed push to FPGAs and bring it closer to ASIC level performance
The basic 3D FPGA architecture is a generalization of the 2D model in which each switch block has six immediate neighbors as opposed to four in 2D.

2D FPGAs stacked on top of each other.
3D FPGA Fabrication

- Multi-chip Module (MCM) techniques
- Optoelectronic 3D FPGA
- The Rothko Architecture
- Modified Rothko Architecture
- All of the above assume that the vertical segments can connect only to adjacent layers
- Monolithically stacked FPGAs
Multi-Chip Module 3D FPGA

- Stacked dies
- Architecture realized by solder bumps between dies to make electrical contact
- Usually I/O buffers are unnecessary
- Solder bumps are at least 100 microns in size – affects via density
- Consume a lot of power
The Rothko Architecture

- A Rothko chip is similar to individual FPGAs stacked on top of each other with connection between layers.
- The Rothko layer is inspired by the Triptych architecture – routing and logic resources combined into RLBs.
Monolithically stacked FPGAs

- Not 3D in the traditional sense
- Possible for vias to have high densities
  - In the future may be able to implement fully 3D switch boxes – 50% reduction in channel width and interconnection delay and power consumption
- Companies like TierLogic have made prototypes of such 3D FPGAs
  - Configuration bits are put on a different layer than the user circuit.
  - Achieves very high logic density as compared to traditional 2D FPGA
3D vs. 2D FPGA: Logic density

- **2D FPGA 4-LUT density follows Moore's law**
  - $180\text{nm} = 110/\text{mm}^2$
  - $20\text{nm} = 4500/\text{mm}^2$

- **TierLogic 3D achieves even higher density**
  - $28\text{nm} = 16K/\text{mm}^2$
Recent studies show that Through-Silicon Vias (TSVs) in 3D switchboxes are never utilized more than 10% even when fully connected.

Since switchboxes eat up a lot of chip area, sparsely connected architectures can improve the area occupied.
Sparsely connected architectures

- Categorized as
  - Internally sparse (IS)
  - Externally sparse (ES)
  - Sparse (a combination of the above two) (SP)
Internally and Externally Sparse

Figure 4. SB patterns of (a) BSL, (b) IS, and (c) ES.
Sunny Egg architecture

- TSV demand much larger (and uneven) in the central region than the peripheral zone
- SE divides the plane into two regions, periphery (egg white), central (egg yolk)
- Two regions are implemented by two separate SP architectures
Challenges faced by 3D FPGAs

- Lack of good CAD software
  - The situation is getting better
- Thermal effects
  - Lead to untraceable faults
  - Heat dissipation issues
  - Thermal stress
    - Such problems can be mitigated by using chip-size packages, thermal bumps, pillars and gels
    - Reduction in size of I/O buffers
- Number of vertical vias in most architectures not as scalable as horizontal routing resources
Software support

- Spiffy - Simultaneous placement and global routing. Placement is recursive partition based.

- Three-dimensional Place & Route (TPR) – Placement (recursive partition based and simulated annealing based, routing is based on PathFinder algorithm)

- 3D MEANDER
Architecture description
Architecture description

- 3D architecture similar to stacked 2D FPGAs
- Single segment – All vertical vias are of unit length, i.e. they span one layer
- Multi segment – Vertical vias are of either unit length, span two layers or span all layers
- In both of the above architectures, the horizontal routing resources are multi segment

- Each CLB consists of one 4-LUT
- Assumes fully deployed switches on each layer
TPR - Overview

TPR flow diagram
Placement Algorithms

TPR can place logic blocks using the following algorithms

- Layer partitioning followed by recursive partitioning based placement
- Simulated annealing based placement
- Layer partitioning followed by simulated annealing based placement (combination of the first two)
Layer Partitioning

This step has two objectives:

- Minimize cut-size between adjacent layers
- Minimize total wirelength

Cut-size needs to be reduced since vertical vias are difficult to fabricate and are a scarce resource on the 3D FPGA.

This step is achieved by the Min-cut algorithm.
Min-cut Algorithm

Comprises of two steps

- Build EV Matrix
- Transform EV Matrix to Band Matrix

By bringing 1s closer to the main diagonal reduces the cut-size.

Reducing the bandwidth decreases the total wirelength of the net.
Min-cut Algorithm Result

Comparison between initial and final interlayer connections

Initial: WL=11, Max-cut=3

Final: WL=7, Max-cut=2
Recursive partition based placement

Input:
- Tech mapped netlist .net G(V,E)
- Architecture description file

Algorithm:
1. Initial min-cut partitioning into layers for via minimization
2. For all layers $i = 0$ to $L-1$ from top to bottom
3. Do partitioning based placement of layer $i$
4. Update timing slacks
5. Re-electruncate critical paths
6. Greedy overlap removal
7. Constraint generation for layers below
8. Write .p placement output file
The algorithm begins by partitioning the area of the FPGA into smaller quad sections until each of the smaller sections has less than 4 blocks.

Delay (slack values) and list of critical paths are updated at every partitioning level. This ensures an accurate estimation of circuit delay as placement progresses.
Placement : Node Alignment

- Delay is influenced by number of routing segments
- Align most critical nets
  - Consider a two terminal net X-Y. Assume Y was already placed. It acts as X's anchor point.
  - Less routing segments will be used if node X is locked into partition A than partition B. If X is put in B, two segments will be used between X & Y
As recursive partitioning goes on, nodes are aligned along narrower placement stripes.

Terminals aligned at higher levels may become unaligned at lower levels if their criticality is not preserved throughout the partitioning levels.

Number of nodes in the leaf partitions might exceed its capacity causing overlaps. Least critical nodes from this partition are moved to the closest and best aligned available empty partition.
Delay of the nets depend on:

- Number of segments in the net
- Timing criticalities which are obtained from slack calculations, also used as edge weights
- Distance between terminals of the nets

Initial delay = delay of single-length segment. Delay due to length is more accurately calculated at every next partitioning.
After a layer is placed, the placement constraints are projected on the layer directly below.

It is desirable to keep parts of the same net in different layers directly below one another so as to reduce the total wirelength.

Two cases might arise if the above is not possible. Both are solved by the greedy overlap removal scheme discussed earlier.
Simulated annealing based placement

- Very similar to the 2D simulated annealing based placement
- Intra-layer as well as inter-layer swapping of blocks is allowed
- The cost of the placement is modified as below:

\[
Cost_{3D}(e) = q \cdot Cost_{2D}(e) + \alpha \cdot Span_z + \beta \cdot Num\_layers(e)
\]

Where,

\( q \) = correction factor for 2D bounding box

\( Cost_{2D} \) = half perimeter bounding box of the projection of all terminals of the net

\( Span_z \) = vertical span of the net

\( Num\_layers \) = number of layers in which terminals of net are placed.
3D SA-P cost function parameters

- $\alpha$ restricts nets from being placed in layers that are far from each other
- $\beta$ makes sure vertical vias are not used frequently since they are considered a scarce resource
Delay calculations are carried out by a lookup tables, similar to the one used in VPR.

To calculate the 2D source-sink delays, the net is projected onto the 2D plane and VPR's lookup table is used to find the delay value corresponding to the 2D distance, i.e. value at \((\Delta x, \Delta y)\)

To calculate the 3D source-sink delays, the corresponding z-direction length is looked up in the table with either coordinate assumed to be zero, i.e. value at \((\Delta z, 0)\) or \((0, \Delta z)\)
Comparison between the placement algorithms

- Simulated annealing based placement gives better placements of better quality
- Recursive partition based placement is much faster than the simulated annealing based approach
The routing algorithm employed in TPR is adapted from PathFinder in VPR.

- Rip up and re-route scheme, routing nets by shortest path using a breadth-first search technique.
- Cost of overused resources is gradually increased forcing nets with alternate routes to avoid using overused resources.
- Differs from VPR's PathFinder in that it penalizes vertical via connections.
Minimum Vertical and Horizontal Channel Widths (VCW & HCW)

- Initial value of VCW is taken from architecture file
- VCW is incremented after routing through current HCW fails a certain number of times

Total Area – Calculated by summing the individual areas of logic and routing blocks of all layers

\[
Area_{3D} = [Area_{LUT} + C_1 \cdot HCW + C_2 \cdot Area_{mux}(HCW)] \cdot L \cdot W^2 \\
+ [C_3 \cdot (HCW - VCW) + C_4 \cdot VCW] \cdot L \cdot (W - 1)^2
\]

\[
Area_{mux}(HCW) = 6 \cdot \left[\log_2(HCW)\right] + 2HCW - 2
\]

Footprint Area – Total Area per layer
Simulation: Delay results
Why do single segment architectures have higher delays than multi-segment architectures?
Simulation: Wirelength Results
### TABLE I
**SIMULATED CIRCUITS: STATISTICS, VERTICAL CHANNEL WIDTH (VCW), AND RUN-TIME**

<table>
<thead>
<tr>
<th>Circuit</th>
<th>No. CLBs</th>
<th>No. IOs</th>
<th>TPR</th>
<th>SA-TPR</th>
<th>VCW</th>
<th>CPU (s)</th>
<th>SA-based placement</th>
<th>Detailed Routing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex5p</td>
<td>1064</td>
<td>71</td>
<td>6</td>
<td>5</td>
<td>7</td>
<td>85</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>Apex4</td>
<td>1262</td>
<td>28</td>
<td>6</td>
<td>5</td>
<td>8</td>
<td>105</td>
<td>76</td>
<td></td>
</tr>
<tr>
<td>Misex3</td>
<td>1397</td>
<td>28</td>
<td>6</td>
<td>5</td>
<td>9</td>
<td>55</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>Alu4</td>
<td>1522</td>
<td>22</td>
<td>5</td>
<td>5</td>
<td>16</td>
<td>145</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>Des</td>
<td>1591</td>
<td>501</td>
<td>5</td>
<td>5</td>
<td>11</td>
<td>227</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>Seq</td>
<td>1750</td>
<td>76</td>
<td>5</td>
<td>5</td>
<td>12</td>
<td>212</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>Apex2</td>
<td>1878</td>
<td>42</td>
<td>5</td>
<td>5</td>
<td>13</td>
<td>243</td>
<td>148</td>
<td></td>
</tr>
<tr>
<td>Spla</td>
<td>3690</td>
<td>62</td>
<td>5</td>
<td>5</td>
<td>37</td>
<td>912</td>
<td>532</td>
<td></td>
</tr>
<tr>
<td>Pdc</td>
<td>4575</td>
<td>56</td>
<td>7</td>
<td>7</td>
<td>60</td>
<td>1354</td>
<td>1039</td>
<td></td>
</tr>
<tr>
<td>Ex1010</td>
<td>4598</td>
<td>20</td>
<td>4</td>
<td>5</td>
<td>56</td>
<td>1238</td>
<td>273</td>
<td></td>
</tr>
<tr>
<td>Dsip</td>
<td>1370</td>
<td>426</td>
<td>5</td>
<td>5</td>
<td>28</td>
<td>154</td>
<td>34</td>
<td></td>
</tr>
<tr>
<td>Tseng</td>
<td>1407</td>
<td>174</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>70</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>Diffeq</td>
<td>1497</td>
<td>103</td>
<td>5</td>
<td>5</td>
<td>14</td>
<td>154</td>
<td>46</td>
<td></td>
</tr>
<tr>
<td>Bigkey</td>
<td>1707</td>
<td>426</td>
<td>5</td>
<td>5</td>
<td>22</td>
<td>209</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>S298</td>
<td>1931</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>23</td>
<td>208</td>
<td>53</td>
<td></td>
</tr>
<tr>
<td>Frisc</td>
<td>3556</td>
<td>136</td>
<td>5</td>
<td>5</td>
<td>56</td>
<td>881</td>
<td>227</td>
<td></td>
</tr>
<tr>
<td>Elliptic</td>
<td>3604</td>
<td>245</td>
<td>5</td>
<td>5</td>
<td>74</td>
<td>818</td>
<td>142</td>
<td></td>
</tr>
<tr>
<td>S38417</td>
<td>6406</td>
<td>135</td>
<td>5</td>
<td>5</td>
<td>133</td>
<td>2000</td>
<td>210</td>
<td></td>
</tr>
<tr>
<td>S38584.1</td>
<td>6447</td>
<td>342</td>
<td>5</td>
<td>5</td>
<td>230</td>
<td>2034</td>
<td>268</td>
<td></td>
</tr>
<tr>
<td>Clma</td>
<td>8383</td>
<td>144</td>
<td>5</td>
<td>5</td>
<td>199</td>
<td>892</td>
<td>950</td>
<td></td>
</tr>
</tbody>
</table>

| Sum     | 1016     | 11996   | 4465 |
# Single Segment Architecture

<table>
<thead>
<tr>
<th>Num layers</th>
<th>TPR</th>
<th>SA-TPR</th>
<th>Combined Partitioning + SA Place</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Delay ($\times 10^7$)</td>
<td>WL</td>
<td>Routing area</td>
</tr>
<tr>
<td>1</td>
<td>1.62</td>
<td>107087.6</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.52</td>
<td>98808.3</td>
<td>1.088</td>
</tr>
<tr>
<td>3</td>
<td>1.37</td>
<td>93162.93</td>
<td>1.097</td>
</tr>
<tr>
<td>4</td>
<td>1.3</td>
<td>87410.88</td>
<td>1.096</td>
</tr>
<tr>
<td>5</td>
<td>1.28</td>
<td>84741.11</td>
<td>1.041</td>
</tr>
<tr>
<td>6</td>
<td>1.22</td>
<td>81707.36</td>
<td>1.089</td>
</tr>
<tr>
<td>7</td>
<td>1.22</td>
<td>80143.2</td>
<td>1.065</td>
</tr>
<tr>
<td>8</td>
<td>1.2</td>
<td>78589.86</td>
<td>1.005</td>
</tr>
<tr>
<td>9</td>
<td>1.21</td>
<td>78456.85</td>
<td>1.018</td>
</tr>
<tr>
<td>10</td>
<td>1.19</td>
<td>78643.86</td>
<td>1.072</td>
</tr>
</tbody>
</table>
## Multi-segment architecture

### TABLE III

**Average of Delay, Wire-Length (WL), Horizontal Channel Width (HCW), and Routing Area for Multi-Seg Architecture**

<table>
<thead>
<tr>
<th>Num layers</th>
<th>TPR Delay ($\times 10^{-7}$)</th>
<th>WL</th>
<th>Routing area</th>
<th>HCW</th>
<th>SA-TPR Delay ($\times 10^{-7}$)</th>
<th>WL</th>
<th>Routing area</th>
<th>HCW</th>
<th>Combined Partitioning + SA Place Delay ($\times 10^{-7}$)</th>
<th>WL</th>
<th>Routing area</th>
<th>HCW</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.62</td>
<td>107087.6</td>
<td>1</td>
<td>1</td>
<td>1.28</td>
<td>79410.53</td>
<td>1</td>
<td>1</td>
<td>1.27</td>
<td>79626.34</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1.52</td>
<td>98808.3</td>
<td>1.088</td>
<td>0.92</td>
<td>1.2</td>
<td>76088.11</td>
<td>1.015</td>
<td>0.96</td>
<td>1.18</td>
<td>73663.53</td>
<td>1.113</td>
<td>0.99</td>
</tr>
<tr>
<td>3</td>
<td>1.38</td>
<td>93162.93</td>
<td>1.096</td>
<td>0.89</td>
<td>1.19</td>
<td>75304.16</td>
<td>1.068</td>
<td>0.96</td>
<td>1.14</td>
<td>70729.52</td>
<td>1.112</td>
<td>0.91</td>
</tr>
<tr>
<td>4</td>
<td>1.35</td>
<td>87410.88</td>
<td>1.053</td>
<td>0.83</td>
<td>1.12</td>
<td>73796.21</td>
<td>1.075</td>
<td>0.95</td>
<td>1.07</td>
<td>67383.43</td>
<td>1.107</td>
<td>0.86</td>
</tr>
<tr>
<td>5</td>
<td>1.24</td>
<td>84741.11</td>
<td>1.025</td>
<td>0.78</td>
<td>1.07</td>
<td>72228.37</td>
<td>1.045</td>
<td>0.90</td>
<td>1.06</td>
<td>66798.35</td>
<td>1.113</td>
<td>0.82</td>
</tr>
<tr>
<td>6</td>
<td>1.21</td>
<td>81707.36</td>
<td>1.038</td>
<td>0.74</td>
<td>1.03</td>
<td>70888.9</td>
<td>1.064</td>
<td>0.90</td>
<td>1.05</td>
<td>66682.31</td>
<td>1.115</td>
<td>0.80</td>
</tr>
<tr>
<td>7</td>
<td>1.2</td>
<td>80143.2</td>
<td>1.003</td>
<td>0.70</td>
<td>1.03</td>
<td>70710.4</td>
<td>1.052</td>
<td>0.88</td>
<td>1.03</td>
<td>65605.21</td>
<td>1.115</td>
<td>0.77</td>
</tr>
<tr>
<td>8</td>
<td>1.19</td>
<td>78589.86</td>
<td>0.942</td>
<td>0.70</td>
<td>1.02</td>
<td>69849.53</td>
<td>1.054</td>
<td>0.87</td>
<td>0.989</td>
<td>67049.56</td>
<td>1.113</td>
<td>0.81</td>
</tr>
<tr>
<td>9</td>
<td>1.19</td>
<td>78456.85</td>
<td>0.967</td>
<td>0.67</td>
<td>0.984</td>
<td>69190.21</td>
<td>1.062</td>
<td>0.87</td>
<td>0.987</td>
<td>65536.51</td>
<td>1.114</td>
<td>0.77</td>
</tr>
<tr>
<td>10</td>
<td>1.16</td>
<td>78643.86</td>
<td>0.981</td>
<td>0.68</td>
<td>0.971</td>
<td>68840.17</td>
<td>1.058</td>
<td>0.85</td>
<td>0.987</td>
<td>65439.38</td>
<td>1.115</td>
<td>0.77</td>
</tr>
</tbody>
</table>
Why does routing area decrease with increase in the number of layers for the multi-segment case?
Why does the combination of layer partitioning and simulated annealing placement perform better than the others?
Conclusion

- The analysis carried out by TPR showed decrease in the following (using either algorithm for placement)
  - Wirelength – by 25%
  - Delay – by 35%
- Multisegment 3D architecture is much more efficient than single segment 3D architecture
- TPR does not investigate 3D architectures derived from monolithically stacked FPGAs
- TPR does not allow changing the distribution of 3D switchboxes on the FPGA