ECE 636
Reconfigurable Computing
Lecture 4
FPGA Placement
Outline

• Brief review of important architecture info for homework
• Basic clustering of BLEs into clusters
• Timing-driven analysis and clustering
• Placement techniques (simulated annealing)
• Timing driven placement
Placement metrics

- Quality metrics for layout:
  - Area
  - Delay
  - Dynamic power consumption (relatively recently)

- Ideally placement and routing would be performed together
  - Some have tried!
  - Both problems are NP-hard
  - For practical considerations placement and routing must be performed separately
Before Placement: Clustering

- Need to group BLEs into groups

- Goals:
  - Minimize number of clusters
  - Minimize inter-cluster wiring
  - Minimize critical path (timing-driven)

- How do we do this
  - Take advantage of cluster architecture
Basic Clustering

- **Iterate until all BLEs consumed**
  - Start new cluster by selecting a random BLE
  - Add BLE with most shared inputs with current cluster to cluster
  - Keep adding until either cluster full or input pins used up
  - *Hill climbing – if some cluster BLEs unused*
    - Add another BLE even if cluster input count temporarily overflowed
    - If input count not eventually reduced select best choice from before hill climbing

Does this do anything for timing performance?
Timing Analysis

netlist with delay for each gate

arrival times

Source: David Pan
Lecture 4: FPGA Placement
Timing Analysis

arrival time / required time

slack = required time - arrival time
Example with interconnect delay

 diagram

 Lecture 4: FPGA Placement

 September 20, 2016
Timing-Driven Clustering – T-VPACK

- Cost metric now considers both connectivity and timing criticality

\[ \text{Connection Criticality}(i) = 1 - \frac{\text{slack}(i)}{\text{MaxSlack}} \]

- Perform an analysis of criticality at beginning considering all wires to be inter-cluster

- As clustering progresses consider the following timing weight ratios
  - LUT delay: 0.1
  - Intra-cluster delay
  - Inter-cluster delay

- Determine “Base” BLE criticality
How to break ties?

- Initially, many paths may have the same number of BLEs

- Include “tie-breaking” in performance cost function

\[
\text{Criticality}(B) = \text{Base BLE Criticality}(B) \cdot (\varepsilon \cdot \text{total paths affected}(B))
\]

\[
\text{Attraction}(B) = \alpha \cdot \text{Criticality}(B) + (1 - \alpha) \cdot \frac{|\text{Nets}(B) \cap \text{Nets}(C)|}{G}
\]
Results for T-VPACK versus VPACK

- Timing driven place and route also used for these results.

Figure 9. Critical Path Delay vs. Cluster Size

Figure 10. Area-Delay Product vs. Cluster Size
Wire length measures

- Estimate wire length by distance between components.

- Possible distance measures:
  - Euclidean distance \( \sqrt{x^2 + y^2} \);
  - Manhattan distance \( x + y \).

- Multi-point nets must be broken up into trees for good estimates.
Placement

• Placement has a set of competing goals.
• Can’t optimize locally and globally simultaneously.
• Use heuristic approaches to evaluate quality.
Placement Algorithms

- Constructive methods: begin from netlist and generate an initial placement.
  - Partitioning methods: mincut and Kernighan-Lin methods
  - Clustering
- Iterative improvement
  - Begin with random or constructive placement.
  - Iterate to improve it.
  - Hill-climbing
Iterative Placement Algorithms

• Pairwise interchange methods

• Force-directed methods

• Simulated annealing
  - Generates best results
  - Can be time consuming

• Partitioning-based approaches
  - Recursively break logic into smaller pieces
Iterative Improvement Algorithms

Force-directed: (classical mechanics)
- Force vector computed on each module corresponding to all nets
- Solve set of non-linear differential equations.

Simulated annealing: (statistical mechanics)
- Model a physical annealing process which optimizes energy.
- Similar to “quenching” metal.
Formulating Force Equations

Use Hooke’s Law
Modules 1, 2, … N

\( m_i \)  mass of module i
\( x_i \)  x position of module i
\( K_{ij} \)  Attractive constant between module i and j
\( F_i \)  Net force on module i from rest of modules

\[
\frac{d^2 x_i}{dt^2} = F_i = - \sum_{j=1}^{N} K_{ij} (x_i - x_j)
\]
**Minimization**

- Using previous formulation will collapse all locations to a single point \( X_1 = X_2 \ldots X_n \)

- Need a repulsive force between modules to prevent overlap

\[
F_i = \sum_{j=1}^{N} (\delta_{ij} \ast R) - K_{ij} (x_i - x_j)
\]

- \( R \) too small -> modules too close
- \( R \) too large -> modules far apart

*Pads have fixed* \( X_i \)
Force-directed

- We know that for the steady state
  \[ \frac{d^2 X_i}{dT^2} = 0 \]

- Determine set of non-linear equations and solve simultaneously using Newton’s Method.

- Problem size can grow quite large as size of device increases.

- Interaction between X and Y coords
  - 3D Device?
Example

\[ X_4 = X_5 = 0 \]
\[ X_6 = X_7 = L \]

\[ 0 = R \left[ \frac{1}{X_1} + \frac{1}{X_1-L} + \frac{1}{X_1-X_2} + \frac{1}{X_1-X_3} \right] - \left[ X_1 + (X_1-L) + 3(X_1-X_3) + (X_1-X_2) \right] \]

• Different results for different \( R \) values

• Solve equations simulataneously

Both for X and Y  -> different repulsive forces?
Force-Directed Relaxation

- Start with random placement.
- Compute forces on each module.
- Pick the module with the largest force on it
- Compute zero-force position with Newton’s method
  - Attempt to move to unoccupied position
  - Or swap with existing module
  - Or move to nearest open position

Continue until final locations determined.
Hill Climbing Algorithms

• To avoid getting trapped in local minima, consider “hill-climbing” approach

• Need to accept worse solutions or make “bad” moves to get global minima.

• Acceptance is probabilistic. Only accept cost-increasing moves some of the time.
Physical Annealing

• Take a metal and heat to high temperature
• Allow it to cool slowly; metal is annealed to a low temperature
• Atoms in the metal are at lower energy states after annealing
• Higher the temperature initially and slower the cooling, the tougher the metal becomes.
• Atoms transition to high energy states and then move to low energy.
**Simulated Annealing**

- Optimization strategy based on physical annealing process
- Generate random moves.
  - Initially, accept moves that decrease and increase cost.
- As temperature decreases, the probability of accepting bad moves decreases.
- Eventually, default to greedy algorithm

Only accept positive moves
Determine when to terminate.
Annealing Algorithm

T = StartingT
Moves_per_iteration = BN^{4/3}
While (stopping_criteria(T) = false) {
    While (Move_Count < Moves_per_Iter) {
        swap blocks
        evaluate Δcost
        if (Accept < Δcost)
            Move block to new location
    }
    T = update(T)
}

N = # blocks
B = scaling factor
T = temperature
Accept Function

$\Delta$cost = new cost – initial cost

If ($\Delta$cost <= 0)
    return (yes)

Else {
    y = exp ( - $\Delta$cost/T)
    r = random (0, 1)
    if (r < y)
        return (yes)
    else
        return (no)
}
Annealing Criteria

• Contemporary FPGA packages use the following parameters:
  1. Starting temp – $20 \times \text{stand}_\text{dev}(\text{cost of N swaps})$
  2. Cost function – weighted sum of wire length and delay
  3. Inner loop – $B \times N^{4/3}$
     • Beta cost function
  4. Stopping criteria –
     • $T < [.005 \times \text{Cost/N}_{\text{nets}}]$
Range Limiting

• As temperature drops, limit scope of swaps
• Increased likelihood of acceptance.
• Can also used to secure critical path performance.
Timing-driven Placement

• Take both wire length and critical path into account
• Problem
  - Critical path changes as I move blocks
  - How do I balance the two objectives
• How do we go about modeling routing delay during placement?

\[
Timing\_Cost(i,j) = \text{Delay}(i,j) \cdot \text{Criticality}(i,j)^{\text{Criticality\_Exponent}}
\]
Estimating delays

• Marquardt and Betz approach (T-VPlace)
  - Perform initial delay analysis to determine shortest delays between each pair of X, Y locations
  - Store information in table for quick look-up
  - Assumption that router will probably find the minimum delay path (a leap of faith!)

\[
\text{Timing Cost}(i,j) = \text{Delay}(i,j) \cdot \text{Criticality}(i,j)^{\text{Criticality Exponent}}
\]
Determining Criticality

• Same basic approach as used for clustering criticality
• For each (i, j) connection from source i and sink j
  - Determine arrival times (pre-order BFS)
  - Determine required arrival times (post-order BFS)
  - Determine slack -> required_arrival_time – arrival_time
  - Criticality(i, j) = \[1 - \text{slack}(i, j)\] / (Max slack)

\[
\text{Timing Cost}(i,j) = \text{Delay}(i,j) \cdot \text{Criticality}^{\text{Criticality Exponent}}(i,j)
\]

What is the purpose of the criticality exponent?
Balancing Wiring and Timing Cost

• Need to determine relative changes in timing and wiring based on moves

\[
\Delta C = \lambda \cdot \frac{\Delta \text{Timing\_Cost}}{\text{Previous\_Timing\_Cost}} + (1 - \lambda) \cdot \frac{\Delta \text{Wiring\_Cost}}{\text{Previous\_Wiring\_Cost}}
\]

• Idea: Use relative changes from previous calculation
  - Both values less than 1
  - Helps balance effect based on scaling parameter

This still doesn’t help address changes in delay
Updated Annealing Algorithm

\[ S = \text{RandomPlacement}() ; \]
\[ T = \text{InitialTemperature}() ; \]
\[ R_{\text{limit}} = \text{InitialRlimit}() ; \]
\[ \text{Criticality}_\text{Exponent} = \text{ComputeNewExponent}() ; \]

\[ \text{ComputeDelayMatrix}() ; \]

\[
\text{while (ExitCriterion () == False) \{ /* Outer loop */}
\]
\[
\quad \text{TimingAnalyze}(); /* Perform a timing-analysis and update each connections criticality */
\quad \text{Previous}_\text{Wiring}_\text{Cost} = \text{Wiring}_\text{Cost}(S) ; /* wire-length minimization normalization term */
\quad \text{Previous}_\text{Timing}_\text{Cost} = \text{Timing}_\text{Cost}(S) ; /* delay minimization normalization term */
\]
\[
\text{while (InnerLoopCriterion () == False) \{ /* Inner loop */}
\]
\[
\quad S_{\text{new}} = \text{GenerateViaMove} (S, R_{\text{limit}}) ;
\quad \Delta \text{Timing}_\text{Cost} = \text{Timing}_\text{Cost}(S_{\text{new}}) - \text{Timing}_\text{Cost}(S) ;
\quad \Delta \text{Wiring}_\text{Cost} = \text{Wiring}_\text{Cost}(S_{\text{new}}) - \text{Wiring}_\text{Cost}(S) ;
\quad \Delta C = \lambda \Delta \text{Timing}_\text{Cost}/\text{Prev}_\text{Timing}_\text{Cost} + (1 - \lambda) \Delta \text{Wiring}_\text{Cost}/\text{Prev}_\text{Wiring}_\text{Cost} ; /* new cost fcn */
\]
\[
\quad \text{if } (\Delta C < 0) \{
\quad 
\quad \quad S = S_{\text{new}} ; /* Move is good, accept */
\quad \}
\]
\[
\quad \text{else} \{
\quad \quad r = \text{random} (0,1) ;
\quad \quad \text{if } (r < e^{-\Delta C / T}) \{
\quad \quad \quad S = S_{\text{new}} ; /* Move is bad, accept anyway */
\quad \quad \}
\quad \}
\]
\[
\} /* End "inner loop" */
\]

\[ T = \text{UpdateTemp}() ; \]
\[ R_{\text{limit}} = \text{UpdateRlimit}() ; \]
\[ \text{Criticality}_\text{Exponent} = \text{ComputeNewExponent}() ; \]
\[
1 /* End "outer loop" */
How often to recalculate delay?

- Recalculating delay once per temperature is good.
- Also simplifies programming somewhat

<table>
<thead>
<tr>
<th>Timing-Analysis Interval</th>
<th>Placement Estimated Critical Path (ns) (20 Circuit Geometric Average)</th>
<th>Wiring Cost (20 Circuit Geometric Average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>39.3</td>
<td>529.6</td>
</tr>
<tr>
<td>2</td>
<td>39.5</td>
<td>531.1</td>
</tr>
<tr>
<td>4</td>
<td>40.1</td>
<td>530.5</td>
</tr>
<tr>
<td>8</td>
<td>40.5</td>
<td>531.0</td>
</tr>
<tr>
<td>16</td>
<td>39.5</td>
<td>530.3</td>
</tr>
<tr>
<td>32</td>
<td>41.4</td>
<td>534.5</td>
</tr>
<tr>
<td>64</td>
<td>41.3</td>
<td>528.3</td>
</tr>
<tr>
<td>128</td>
<td>43.0</td>
<td>522.9</td>
</tr>
<tr>
<td>Never</td>
<td>43.0</td>
<td>522.9</td>
</tr>
</tbody>
</table>
### How important is timing-driven placement?

**TABLE 1.6** Post-place-and-route comparison of VPlace and T-VPace (cluster size = 1).

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Post-Place-and-Route Minimum Channel Width ($W_{min}$)</th>
<th>Post-Place-and-Route Critical Path (ns) $W = \infty$</th>
<th>Post-Place-and-Route Critical Path (ns) $W = W_{min} + 20%$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VPlace ((\lambda = 0))</td>
<td>T-VPace ((\lambda = 0))</td>
<td>VPlace ((\lambda = 0))</td>
</tr>
<tr>
<td>alu4</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>apex2</td>
<td>15</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>apex4</td>
<td>17</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>bighkey</td>
<td>13</td>
<td>13</td>
<td>10</td>
</tr>
<tr>
<td>clmu</td>
<td>16</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>des</td>
<td>11</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>difseq</td>
<td>11</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>dsp</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>elliptic</td>
<td>14</td>
<td>16</td>
<td>15</td>
</tr>
<tr>
<td>ex1010</td>
<td>14</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>ex3p</td>
<td>17</td>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>frisc</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>mixex3</td>
<td>14</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>pdc</td>
<td>22</td>
<td>21</td>
<td>24</td>
</tr>
<tr>
<td>s298</td>
<td>11</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>s38417</td>
<td>11</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>s38584.1</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>seq</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>spla</td>
<td>18</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>tseng</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>Geom. Av.</td>
<td>13.78</td>
<td>14.22</td>
<td>14.50</td>
</tr>
</tbody>
</table>

%diff w.r.t VPlace — +3.2% +5.2% — — +1.8% -29.7% — — +1.04% -20.0%

Run time Penalty – 2.5X
Summary

- Placement and clustering of modules critically important for subsequent routing step
- Often initial placement performed and then iteratively improved
- Mincut partitioning approaches sometimes used for initial placement
- Efficient timing analysis a key to successful placement
- Island-style devices benefit from simulated annealing approaches
  - Accurate cost function is the key to success
- Issues related to power consumption remain