Abstract
This paper describes an FPGA-based accelerator for maze routing
applications such as integrated circuit detailed routing. The accelerator
efﬁciently supports multiple layers, multi-terminal nets, and rip and
recut. By time-multiplexing multiple layers over a two-dimensional array
of processing elements, this approach can support grids large enough for
practical detailed routing while providing at least 5 orders of magnitude
speedup over software running on a modern desktop computer. The
current implementation supports 32 × 32 routing grids with up to 16 layers
in a single Xilinx XC2V6000 FPGA. Up to 64 × 64 routing grids are
feasible in larger commercialy available FPGAs. Performance
measurements (including interface overhead) show a speedup of 200x
over software running on a 3.7GHz Pentium Xeon desktop computer
depending on the number of layers used. An improved interface design
could yield signiﬁcantly larger speedups.

Additional Features
- Multi-terminal Net Routing
- Etching - Identiﬁcation of Ripup Sets

L4 Organization
- Key idea: 2-D Array of Simple PEs
- Control unit broadcasts commands to PEs
- Reductions O(N) expansion step to O(N)

PE Connection Detail

PE Implementation for Multilayer Routing
- Layers processed from bottom to top
- Expands all grids on current layer in parallel

PE States (for each grid)
- EMPTY: Cell unoccupied and unprocessed
- BLOCKED: Cell occupied by routing net
- AE: Expanded - shortest backtrace path to east
- AX: Expanded - shortest backtrace path to north
- AS: Expanded - shortest backtrace path to south
- AX: Expanded - shortest backtrace path to west
- AD: Expanded - shortest backtrace path to diad

PE Commands
- READ: Return state of selected cell(s)
- WRITE: Write state of selected cell(s)
- EXPAND: If EMPTY cell and a neighboring cell is EXPANDable,
  then a neighboring cell is expanded
- CLEAR: Reset expanded cells to EMPTY state

Implementation
- 32 × 32 accelerator supports 1-16 layers
- Host FPGA: Xilinx XC2V6000
- Board: Dini 3000K10S w/ PCI Interface
- Host: 1.8GHz Pentium 4 Linux PC

Implementation Results
- "Adjacent" route (0.0, 0 - 0.0, 1)
- "Corner" route (0.0, 0 - 31.31, 1)
- 90 Random 2-Terminal Nets
- 40 Random Multi-Terminal Nets

Software measurements: Using P4 cycle counter on a 3.7GHz P4 with
no PCI interface overhead
Using P4 cycle counter on a 1.8GHz P4 with PCI interface overhead
Using PCI interface overhead

Table 1. Implementation Results

<table>
<thead>
<tr>
<th>Category</th>
<th>Average Execution Time (µs)</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjacent Route</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0,0</td>
<td>2.13 ± 0.15</td>
<td>0.04</td>
</tr>
<tr>
<td>Ripup</td>
<td>0.79 ± 04</td>
<td>0.06</td>
</tr>
<tr>
<td>Cmd</td>
<td>2.17 ± 04</td>
<td>0.06</td>
</tr>
<tr>
<td>1</td>
<td>8.38 ± 04</td>
<td>0.06</td>
</tr>
<tr>
<td>Ripup</td>
<td>0.79 ± 04</td>
<td>0.06</td>
</tr>
<tr>
<td>Cmd</td>
<td>2.17 ± 04</td>
<td>0.06</td>
</tr>
<tr>
<td>Corner Route</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0,0</td>
<td>2.13 ± 015</td>
<td>0.04</td>
</tr>
<tr>
<td>Ripup</td>
<td>0.79 ± 0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>Cmd</td>
<td>2.17 ± 0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>1</td>
<td>8.38 ± 0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>Ripup</td>
<td>0.79 ± 0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>Cmd</td>
<td>2.17 ± 0.04</td>
<td>0.04</td>
</tr>
</tbody>
</table>

Table 2. Additional Features

- Multi-terminal Net Routing
- Etching - Identification of Ripup Sets