Efficient Priority-Queue Data Structure for Hardware Implementation

Andrew Morton  
University of Waterloo  
Waterloo, Canada  
armorton@uwaterloo.ca

Insop Song  
Dalsa Inc  
Waterloo, Canada  
insop.song@gmail.com

Jeffrey Liu  
University of Waterloo  
Waterloo, Canada  
jc2liu@uwaterloo.ca

Background

What is a Priority-Queue?
- It sorts items according to priority level
- Items are retrieved in FIFO order within each priority level
- Used in switches to sort packets into Quality of Service levels
- Also applicable to fixed-priority scheduling of software tasks

Architecture 1: Multi-FIFO

Advantages:
- fast enqueue
- Disadvantages:
- slow dequeue
- size a N^P

Architecture 2: Shift Register

Advantages:
- simple dequeue
- Disadvantages:
- bus loading for enqueues
- comparator and mux per reg

Architecture 3: Systolic Array

Advantages:
- fast enqueue, dequeue
- 2 comparators and 2 muxes per register

Design

Concept
- all queues share same array
- each queue (priority) has an index to its first element
- example to right:
  - priority 1 – 3 items
  - priority 2 – 1 item
  - priority 0,3 - empty

Architecture
- enqueue operation uses index of next queue to mark insertion point
- dequeue at priority 0 gets item from highest-priority non-empty queue
- no comparators results in fast enqueue/dequeue and small size

Results

Synthesis Results for Virtex II Pro 70

<table>
<thead>
<tr>
<th>Priorities</th>
<th>Array Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>215 498 1194 5748</td>
</tr>
<tr>
<td>16</td>
<td>341 669 1257 5806</td>
</tr>
<tr>
<td>32</td>
<td>858 1300 5799</td>
</tr>
<tr>
<td>64</td>
<td>1753 7023</td>
</tr>
</tbody>
</table>

FPGA Slices

<table>
<thead>
<tr>
<th>Priorities</th>
<th>Array Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>115.2 87.9 56.5 29.7</td>
</tr>
<tr>
<td>32</td>
<td>87.9 71.4 52.1 24.8</td>
</tr>
<tr>
<td>64</td>
<td>70.7 58.5 25.7</td>
</tr>
</tbody>
</table>

Clock Rate (MHz)

<table>
<thead>
<tr>
<th>Priorities</th>
<th>Array Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>44.5 19.5</td>
</tr>
</tbody>
</table>

- compares well with systolic array in both size and speed
- speed much better than multi-FIFO results although size much bigger
- will be used for scheduling and wait queues in hybrid OS