10. High-level synthesis steps

Scheduling III – scheduling for minimum power

- Each resource has an additional attribute: power consumption
- Modification of Hu’s algorithm for minimum power consumption:
  1. Start with only one type of each resource
  2. Compute latest start times for each node using ALAP with specified latency constraint
     - Latest start times must take into account multicycle operations
  3. For each cycle and for each resource type
     3.1 Determine candidate nodes
     3.2 Compute slack for each candidate
        Slack = current cycle - latest possible cycle
     3.3 If there are nodes with 0 slack, schedule all of them
        - Ask for extra resources if necessary
     3.4 If there are no nodes with 0 slack, schedule one node from those with the lowest slack; arbitrary, but first try a node with the highest power consumption
  4. Repeat loop 3. until all nodes scheduled

Example: minimum power schedule with latency constraints

1. Power: ALU – 1mW, MUL – 5mW
2. Find ALAP schedule

Power constrained scheduling: based on scheduling for minimum resources with latency constraints

3. For cycle and for each resource type
   3.1 Determine candidate nodes
   3.2 Compute slack for each candidate
      Slack = current cycle - latest possible cycle
   3.3 If there are nodes with 0 slack, schedule all of them
   3.4 If there are no nodes with 0 slack, schedule one node from those with the lowest slack; arbitrary, but first try a node with the highest power consumption
   4. Repeat loop 3. until all nodes scheduled

Example: minimum resources with latency constraints

3. For cycle and for each resource type
   3.1 Determine candidate nodes
   3.2 Compute slack for each candidate
      Slack = current cycle - latest possible cycle
   Candidates = {1, 2, 3, 4}
   Slack

Initial Resources = 1 MUL, 1 ALU

Cycle 1

Node | LPC | Slack | Cycle
--- | --- | --- | ---
1   | 1   | 0    | 1
2   | 0   | 1    | 2
3   | 1   | 0    | 3
4   | 3   | 2    | 4
5   | 2   | 2    | 5
6   | 2   | 2    | 6
7   | 3   | 3    | 7
3.3. Schedule candidate nodes with 0 slack
If necessary, update required number of resources

Candidates = \{1,2,3,4\}
Resources = 1 Mul, 2 ALU (1 ALU added to initial list of resources), Power: 7mW

Cycle 1

Example: minimum power with latency constraints

3. For each resource type
3.1. Determine candidate nodes C
3.2. Compute slack for each candidate
   = Slack = current cycle - latest possible cycle

Candidates = \{4,5,6\}
Resources = 1 Mul, 2 ALU

Cycle 2

Example: minimum power with latency constraints

2.3. Schedule the nodes with 0 slack
If necessary, update required number of resources

Candidates = \{4,5,6\}
Resources = 2 Mul, 2 ALU, (1 Mul added)
Power = 10mW

Cycle 2

Example: minimum power with latency constraints

There are spare ALUs -- but we do not schedule node 4!
Example: minimum power with latency constraints

- Final Schedule, power 10mW or less

<table>
<thead>
<tr>
<th>Required Resources = 2 Mult, 2 ALU</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Node</th>
<th>LPC</th>
<th>Stack</th>
<th>Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

Node 1: Slack  Cycle 1
Node 2: Slack  Cycle 2
Node 3: Slack  Cycle 3
Node 4: Slack  Cycle 2
Node 5: Slack  Cycle 2
Node 6: Slack  Cycle 2
Node 7: Slack  Cycle 2

Total Power:
- Cycle 1: 7mW
- Cycle 2: 10mW
- Cycle 3: 1mW

Total Power: 18mW