Program In – Chip Out

CSE 291E / EE260C
Spring 2002
Overview

• Quick review of basic architectures
  – What is Single Issue, Super Scalar, VLIW,
• Overview of Systolic Arrays
• Overview of PICO Project
• DataWidth Reduction Algorithm
Architecture Review

• Code Segment

```
For(n=0; n<100; n++)
{
    A[n+1] = A[n]*x[n];
    C[n+1] = C[n]*z[n] + B[n];
}
```

• How does this map on different architectures?
  – In-order Single Issue
  – Superscalar
  – VLIW
In-Order Single Issue

1) \(A[n+1] = A[n] \times x[n]\)
2) \(r1 = B[n] \times y[n]\)
3) \(B[n+1] = r1 + A[n]\)
4) \(r2 = C[n] \times z[n]\)
5) \(C[n+1] = r2 + B[n]\)
Superscalar

1) \( A[n+1] = A[n] \times x[n] \)
2) \( r1 = B[n] \times y[n] \)
3) \( B[n+1] = r1 + A[n] \)
4) \( r2 = C[n] \times z[n] \)
5) \( C[n+1] = r2 + B[n] \)
VLIW

1:2) \( A[n+1] = A[n] \times x[n] : r1 \)
3:4) \( B[n+1] = r1 + A[n] : r2 \)
5) \( C[n+1] = r2 + B[n] : \text{NOP} \)

---

<table>
<thead>
<tr>
<th>Time</th>
<th>1 : 2</th>
<th>3 : 4</th>
<th>5 : NOP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Systolic Arrays

- Where does name “Systolic Array” come from?
  - Array: to set or place in order
  - Systolic: a rhythmically recurrent contraction; especially the contraction of the heart by which the blood is forced onward and the circulation kept up

- What is a Systolic Array?
  - A network of PEs that rhythmically compute and pass data through the system
Systolic Arrays

- All PEs are uniform and fully pipelined (usually)
- Only local interconnection (nearest neighbor)
- Some relaxations are introduction to increase the utility of systolic arrays
  - Neighbor interconnection (near, but not nearest)
  - Data broadcast operations
  - Different PEs, especially at the boundaries
Data Graphs for Systolic Arrays

• Example: dynamic programming
Walking the Data Graph
Building the Array
PICO

- Program In Chip Out (PICO)
  - Architecture synthesis system from HP
  - Work done by Bob Rau’s group
  - Input: Application written in subset of C
    - No complex pointer
    - No wacky array indexing
  - Metric: Chip area and performance
  - Output: H/W as VHDL & S/W as binary
  - Generates Pareto-optimal architecture
Paretto Optimality

• For a set of design points, a given design is pareto optimal if:
  – No other design is better with respect to every evaluation metric
  – This means there can be multiple pareto optimal points
PICO Architecture
PICO Design Framework

- Workload and requirements specification
- Spacewalker (design space explorer)
- Design specification (parameters)
- Constructor Hardware, software, simulators, and so on
- Evaluator
- Design
- Parameter ranges
- Architecture framework
- Parameterized design space
- Component library
- Execution time
- Area
- Pareto optimal designs
PICO Design Flow
PICO NPA Design

Diagram showing the process of NPA (Nano Parallel Architecture) design, involving steps such as exploring range, loop nesting, dependence analysis, tiling, iteration scheduling, parallel loop nest to hardware, and generating NPA design (VHDL) with considerations for performance, area, and cost.
PICO Analysis

L1: \( x = a + 1 \)

L2: \( y = x \times b \)

Loop:

L3: \( y = y + 1 \)

If () goto loop

L4: \( z = y + c \)
PICO Datawidth Analysis

original code

after forward prop

after backward prop