More phase ordering problems: Code generation for DSPs

Example: Hitachi SH-3 DSP

<table>
<thead>
<tr>
<th>A</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>X</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>MAYX</td>
<td></td>
</tr>
</tbody>
</table>

mul add/sub store

M A

mul add/sub

1 1 0

M A

mul add/sub

Memory layout for dual-banked memory

Some VLIW architectures have multiple (3+ as in SH3) memory banks simultaneously e.g. load on A, load on B, move A⇒B

Instruction scheduling needs information about the residence of operands and instructions

Heuristic [Leupers'00] iterative optimization with simulated annealing

needs information about concurrent need of resources

+ mapping instructions to clusters

+ explicit parallelism on subwords

+ SIMD parallelism on subwords

+ non-homogeneous register sets

+ clustered VLIW architectures

memory models need different generation units

+ typical features:

  + execution time/throughput, code size, power consumption do matter

  + need as workhorse in high-performance embedded systems

  + optimized for high throughput for special applications

  + Digital Signal Processors (DSPs)

Heuristic (Leupers'00) iterative optimization with simulated annealing

needs information about the residence of operands and instructions

Instruction scheduling needs information about the concurrent need of resources

+ mapping instructions to clusters

+ explicit parallelism on subwords

+ SIMD parallelism on subwords

+ non-homogeneous register sets

+ clustered VLIW architectures

memory models need different generation units

+ typical features:

  + execution time/throughput, code size, power consumption do matter

  + need as workhorse in high-performance embedded systems

  + optimized for high throughput for special applications

  + Digital Signal Processors (DSPs)

+ optimal partitioning is NP-complete [Garey/Johnson'79]
Memory layout for dual-banked memory (3)

Phase 1: construct bank interference graph
- Start with empty bank interference graph
- Weight each edge by nesting depth
- Where $n$ and $v$ access different variables
- All instructions that can be scheduled
- Place instructions from $v$ into the same bank as units available
- At each zero-indegree set $z$

Phase 1: construct bank interference graph

Extension: [Sudarsanam/Malik ICCAD'95, TOSAES'2000]

Minimise-cost labelling of the graph: simulated annealing heuristic

Common interference graphs with different kinds of edges

Integrate bank allocation and register allocation

Memory layout for dual-banked memory (4)

Phase 2: partition the bank interference graph
- Greedy heuristic—here for 2 banks
- Start with partition $P = \{a, n\}$
- Repeat
- For each partition $P$, remove an edge $(u, v)$
- Multiply partition cost $d$ by $w_{u,v}$
- Remove an edge that yields the maximum cost reduction

until cost cannot be decreased further

+ 13-43% improvement in practice

+ Requires alias analysis, especially for array elements

+ Assumption of a large, general-purpose register file is unrealistic

Integration dual data memory banks in digital signal processors

[Saghir/Chow/Lee ASPLOS-VII 1996]
Hardwaredescriptionlanguagesforretargetablecodegeneration

Behavioralmodels
Mixedmodels
Structuralmodels

+instructionsemantics
+assemblyrepresentation
+proceduralorgrammar-based
+highabstractionfromhardware
+incompletestructuralconstraints
+netlistofhardwareblocks
+incompleterelationshipsonnetlists
+structuredhazardsetc.

IBURG/OLIVEspec.lang.
SALTOspec.lang.
ML[Fauthetal.'95]
ISDL[Hamono,Devadas'98]
TDL[Kaster,00]
MINOLA[Fauthetal.'95]
MIMOLA[Marwedel'93]