

Hardware/Software Codesign of a Clustered VLIW Architecture

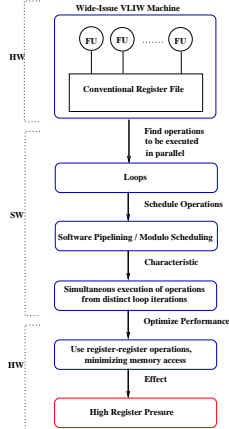
Marcio Merino Fernandes, Josep Llosa, Nigel Topham



Abstract: VLIW machines exploit instruction level parallelism by executing distinct machines operations in a number of functional units (FUs). Operations to be executed in parallel are identified at compile time, resulting in a simple hardware organization when comparing to superscalar architectures. This hardware organization allows the inclusion of a larger number of FUs, however the scalability is still compromised by the register file required by such machines. Our first approach to the problem was to propose a register file organization based on queues (QRF), which presents some advantages for this type of architecture. We are also working on a machine model organized in clusters, each comprising a few FUs and a private register file, all interconnected by a bi-directional ring of queues. Experimental results have shown that this organization is noticeable more efficient than a single cluster machine in terms of silicon area, which might determine the overall execution time. In addition, partitioning and scheduling techniques have been developed in order to exploit the available processing power of this architecture.

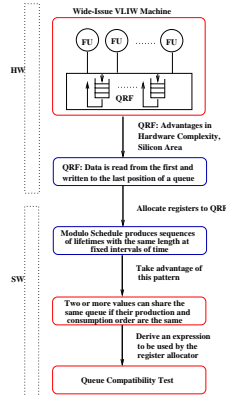
Exploiting ILP

Large amounts of instruction level parallelism (ILP) can be found in loops. It can be exploited by VLIW machines using techniques such as modulo scheduled, however this often results in high register pressure as the number of FUs scales up, requiring alternative hardware and software schemes to deal with the problem.



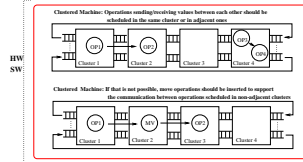
Queue Register File

We have found that a QRF is an interesting solution in terms of hardware complexity and silicon area when compared to a conventional one. However it requires new schemes to perform register allocation, which is accomplished using a queue register allocation condition we have derived. Using a QRF with modulo scheduled loops allows other advantages in terms of register name space, code generation, and register allocation.

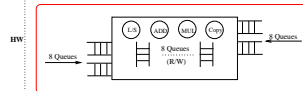


Partitioned Schedules

Clustering of a few FUs and private register files is a more efficient organization for a very wide issue machine. We propose an architecture based on an array of clusters interconnected by a bi-directional ring of queues, which are used as a QRF to provide communication among non-local FUs. The performance of such architecture depends on the ability of the partitioning algorithm to distribute operations among clusters with minimum overhead to perform communication tasks. We have developed some heuristics that results in efficient schedules for machines up to 10 clusters of 3 FUs each (plus a supporting copy FU).



Experimental results using 1258 loops from the *Perfect Club Benchmark* have shown that the following cluster configuration should suffice to schedule over 99% of the loops, for machine models ranging from 1 to 10 clusters.



Further Developments

The hardware and software models developed so far have shown to be a viable alternative to design a scalable VLIW architecture able to achieve high performance in loop execution. Further developments should deal with issues regarding the execution of non-numeric code, in order to deliver a general purpose architecture. Additional information along with detailed experimental results can be found in the references [1, 2, 3, 4].

References

- [1] M. Fernandes, J. Llosa, and N. Topham. Partitioned schedules for clustered VLIW architectures. In *IPPS '98, 12th International Parallel Processing Symposium*, Orlando, USA, 1998.
- [2] M. Fernandes, J. Llosa, and N. Topham. Allocating lifetimes to queues in software pipelined architectures. In *EURO-PAR '97, Third International Euro-Par Conference*, Passau, Germany, 1997.
- [3] M. Fernandes, J. Llosa, and N. Topham. Extending a VLIW architecture model. Technical Report ECS-CSG-34-97, University of Edinburgh, Department of Computer Science, 1997.
- [4] M. Fernandes, J. Llosa, and N. Topham. Using queues for register file organization in VLIW architectures. Technical Report ECS-CSG-29-97, University of Edinburgh, Department of Computer Science, 1997.