A course on Parallel Computer Architecture with Projects

Subramaniam Ganesan
Oakland University, Rochester, MI 48309
Ganesan@oakland.edu

Abstract:

This paper describes integration of simple design projects as part of an advanced course titled "Parallel computer architecture". The field of computer architecture is rapidly evolving and the current technology dictates the popularity of Parallel computer architecture. This paper describes how to introduce the fundamental Parallel architecture concepts, utilize simulation techniques, and group design projects, to make this course interesting. Co-operative learning in the projects provides a good impact on student learning. Two types of projects are assigned: minor uncomplicated projects to emphasize basics, and larger realistic projects with multiple solutions. Evaluation of the course by students to survey questionnaire on the content is done. It indicates that the students achieved a better understanding of the principles and had sustained interest throughout the semester. Brief descriptions of some of the projects namely, reconfigurable parallel architecture, instruction level parallelism simulator, multiple level cache design and simulation, analysis of VLIW architecture, and distributed architecture with CAN are also given.

1. Introduction:

The past few years have seen a steady change in computer architecture. Dual core processors for Personal computers, multi-core processors for cell phone and other applications are emphasizing parallel computer architecture in a different dimension. A number of traditional textbooks on Parallel computer architecture, Parallel processing, Grid computing are available [1 -6]. This paper describes how to introduce the fundamental Parallel architecture concepts, utilize simulation techniques, and group design work, to make this course interesting. Group design makes the course interesting. The teams of two or three students work on the design or solution and present it briefly in the class in addition to submitting a detailed report. The projects are of two types, simple and large. The simple projects aim to design a sub module or interface or software in different architectures. The large projects aim to design fully a parallel architecture for specific applications. Co-operative learning in the projects provides a good impact on student learning. The section two gives the list of topics covered in the course, the section three lists some of the projects done in this course by different teams, the sections four and five describe briefly some of the projects done. The conclusion indicates the benefits in this approach.

2. Details of the topics covered:

This section gives the list of topics covered in this course. The topic list consists of many material from a typical parallel computer architecture book and material from a number of research papers and details of existing parallel computers.

The topics covered in the course can be grouped under four areas: Parallel architecture concepts, design issues, application and performance, and emerging parallel architectures. The groups are not taught sequentially, but relevant topics from different groups are covered sequentially.

Some of the topics covered under the Parallel architecture concepts are: Parallelism in Uniprocessor, Parallel computer structures: Pipelined, Array, Multiprocessor; Architectural Classification: SISD, SIMD, MIMD, MISD; Data Parallel Processing, Shared address space; Message passing, distributed memory, multi level caches, Cache coherency, Parallel Computer Models: Shared- Memory, distributed memory, Multivector and SIMD computers, PRAM, Grain sizes, Latency, Scheduling, CISC, RISC, and Superscalar Processors.

Some of the topics covered under the design issues are: Design of multiple bus, multiprocessor and multiple memory module system, bus arbitration logic, use of multi-port memory, interconnection network design, crossbar network, cluster gateway, SIMD, MIMD design with multiple microprocessors, reconfigurable architecture with FPGA, orthogonal memory design, parallel programs and parallelization process, Snooping Cache protocols and controller design, timing analysis for real time system applications of multiprocessor.

Some of the topics covered under the applications and performance are: Parallel processing applications: Simulate Ocean currents, Parallelization of example programs, Principles of Scalable Performance: Performance Metrics and Measure, speed-up performance laws, Scalability analysis, Partitioning for performance: Load balance, Reducing communication, Data Access in a Multi memory system,

The topics covered under the emerging architectures are: Comparison of commercial parallel architectures including Intel paragon, Clusters and networks of Workstations, Vector processors and SIMD: Cray Y-MP, MPP, VP2000, CM-2, Connection Machine CM5, Dataflow Architectures, System on a chip with multiple processors, dual core and multi-core architectures.

3. Projects:

In addition to the lectures on the above topics, the students do a number of simple projects as a team work during the course. A list of some of these projects is given here. Design and build a scalable bus arbitration logic for a multiple bus multiple shared memory module system [7,8]; Design a SIMD system for a simple application using DSP processors [9]; Analyze the software for a VLIW DSP processor which is capable of doing eight instructions at a time (four nanosecond) [10]; study the multi-core processor "Tri-Core" processor by infineon and design the software flow; Use Trimaran simulation software to study Instruction Level Parallel (ILP) Architecture and Explicitly parallel instruction computing (EPIC) [11]; design of multiprocessor with Orthogonal memory system [12], multiprocessor system design using multiport memory ICs [13-16],

Multiprocessor system with Crossbar switch ICs [17,], Use of VHDL to implement a RISC machine with specific features, design reconfigurable switches in a multiprocessor system [18], a distributed message passing architecture using CAN (controller area network) protocol [19]. These projects make the course interesting to learn. The student teams make detailed presentations on these projects.

In addition, each student does a detailed study on another topic (large project) and makes a presentation. Some of the large project titles are: Reconfigurable parallel architecture and applications, SIMD and image processing applications, Image understanding and processing architectures, Pipelined multiple CPU architecture, and Multiple multi-core CPU with multiple level caches.

Next subsections describe a few small non complex projects done as a group, which can also later be done as a detailed large project.

SIMD design

A simple SIMD (single instruction multiple data stream) machine using DSP processors is described in the paper [9]. The students designed a SIMD machine using the latest DSP TMS 320C6713 processor. It gave them design experience for SIMD, data sharing using shared multiport memory system and a need for another processor system to load the memory with the required data for use by SIMD. The time spent in data initialization at the beginning and data integration at the end taught them that the performance of the SIMD machine has limitation, i.e. setup time and final integration or Input/output time.

4. Distributed Architecture with CAN network

CAN (Controller Area Network) protocol [19] is a networking protocol intended for microcontroller *distributed system***s** such that to run *multiple* (different) tasks on *multiple* processors without shared memory. With few modifications, the distributed architecture of CAN networks becomes adequate for *parallel processing* as well such that to run *one* task, with multiple data, concurrently in parallel on *multiple* processors, or other types of parallel processing that suit the distributed systems architecture. The freescale 9SHCS12X with CAN ports are connected together by CAN network and the distributed system for computation is studied.

5. VLIW DSP processor system

The students study VLIW (very large instruction word) architecture of TI DSP TMS320C6713 processor [10]. They also use the Code composer studio development tools to write simple matrix multiplication programs and other algorithms in C language. The selected DSP processor is capable of executing eight instructions in one instruction cycle. They see the parallel assembly code compiled by the optimized C compiler. They also look at the degree of parallelism at every instruction cycle. Depending on the application, the degree of parallelism varies from 1 to 8. They compute the average degree of parallelism for a typical application.

6. Cache Simulation

The students simulate a shared memory system with multiple caches and study the performance of the system. They also study the effect of one cache coherence protocol for a multiprocessor system. Cache simulators are available at various university web sites [20-22]. Dinero IV is a cache simulator from University of Wisconsin [20] for memory reference traces. Subroutine-callable interface in addition to trace-reading program

- simulation of multi-level caches
- simulation of dissimilar I and D caches (instruction and data caches)
- better performance, especially for highly associative caches
- classification of compulsory, capacity, and conflict misses
- support for multiple input formats
- cleaned up and modernized code, improved portability

Cache simulator from Georgia Tech [21] models the effectiveness of various cache algorithms and configurations. Currently, the system supports simulation of static trace files. Several trace files are available on the host server, where the cache simulations will be based upon one of the trace files. One can specify all the parameters and submit the configurations to generate the output simulation on their machine. The Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB) [22] has the simulator to predict program execution times with a flexible memory-hierarchy.

7. Parallel Programming

Students during the next offering will write parallel programs to run on a small grid computer system with multiple PCs and study the performance. It is essentially a set of Linux/windows PCs, interconnected with a separate network using gigabit technology. One of the PC will be a “head node” (login node and scheduling master), with a scheduler (such as openPBS, or PBSPro). The class also can use the Teragrid[23].

8. Conclusion

This paper discusses ways to teach advanced computer architecture with simple projects and simulations. The Projects and simulations make the course more interesting and challenging. The topics covered in the course are grouped under four areas: Parallel architecture concepts, design issues, application and performance, and emerging parallel architectures. Co-operative learning in the projects provides a good impact on student learning. Two types of projects are assigned: minor uncomplicated projects to emphasize basics, and larger realistic projects with multiple solutions. Evaluation of the course by students to survey questionnaire on the content is done. It indicates that the students achieved a better understanding of the principles and had sustained interest throughout the semester.

References:

1. David Culler, J.P. Singh and A.Gupta, “Parallel Computer Architecture” Margan Kaufman, 1999, ISBN: 1-55860-343-3.

2. Kai Hwang, “ Advanced Computer Architecture- Parallelism, Scalability Programmability”, McGraw Hill, 1993, ISBN- 0-07-031622-8

3. Dezso Sima, Terance Fountain, Peter Kacsuk, “Advanced computer architectures- a design space approach”, Addison Wesley, 1997, ISBN 0-201-42291-3.

4. Hwasahm El-Rewini, M Abd-El-Barr, “Advanced computer architecture and parallel processing”, John Wiley 2005, ISBN: 0-471-46740-5.

5. Dan I Moldovan, “Parallel Processing from Applications to systems”, Morgan Kaufman, 1993, ISBN 1-55860-254-2.

6. R. Buyya, “High performance Cluster Computing”, Prentice Hall, 1999.

7. D. Sundararajan, M.O.Ahmad and S.Ganesan, "Interface Links 8 Bit Chips to Provide Multi-processing," Electronics, Oct 20, 1983, pp 140-141.

8. Sundararajan, M.O.Ahmad and S.Ganesan, "Multiprocessors", IEEE Trans on Circuits and systems, June 1985, pp.620-622.

9. P.V.R.Raja and S.Ganesan, "An SIMD multiple DSP micro-processor system for image processing", Microprocessor and Microsystems journal, Vol 15, No 9, November 1991, pp 493-50

10. Nat Seshan, “ High VelociTI Processing”, IEEE Signal Processing Magazine, March 1998.

11. www.trimaran.org Simulation software

12. P.Raja, S.Ganesan, "An Orthoganal Multiprocessor with Snooping Caches", Proceedings of Fifth Distributed Memory Computing Conference, S.Carolina, April 1990:

13. P.V.Raja, S. Ganesan, “ A Hardware based Snooping Cache Coherence Protocol “, Proceedings of 36th Midwest Symposium on Circuit and Systems, August 1993, pp. 181-184

14. S.Ganesan, " A dual DSP microprocessor system for real-time digital correlation, " Microprocessor and Microsystems Journal, Vol 15, No 7, September 1991, pp 379- 384.

15. S.Ganesan, " Multiprocessor architectures using the DSP TMS 320C40 processors ", The computer applications journal, The Circuit Cellar Project File, Volume 2, 1993, pp 1-12.

16. S. Ganesan, M.O. Ahmad and M.N.S. Swamy, "A Multi Micro-processor System with Distributed Common Memory for Real Time Digital Correlation and Spectrum Analysis", IEEE ICASSP proceedings, March 1985, pp 1609-1612.

17. T. Ramesh, S.Ganesan, “ Reconfigurable Shared and dedicated bus multiprocessor for parallel computing”, Computer and Electrical Engineering- an international journal by Pergamon Press, vol. 19, No. 5, pp 377- 386, 1993.

18. T.Ramesh, S.Ganesan, " Reconfigurable Reduced Bus Multi-processor Interconnection Network", Proceedings of Fifth Distributed Memory Computing Conference, S.Carolina, April 1990.

19. Mansoor Mahfoud and S.Ganesan, “Potentials of Parallel Processing on CAN Networks” SAE world congress, 2006.

20. http://www.cs.wisc.edu/~markhill/DineroIV
21. http://www.ece.gatech.edu/research/labs/reveng/cachesim/
22. http://www.zib.de/schintke/ldasim/
23. www.teragrid.org

When porting a parallel program on computers with a different architecture a programmer faces the problem of invalidity of once developed parallel procedures. At present there are no universal means of adapting programs to a concrete architecture of supercomputers and that's why this problem has to be mostly solved manually what makes the process very labor-intensive [15].Â Besides analysis of the structure and properties of the developed programs on all the projecting stages, modeling can help describe all the peculiarities of interaction between parallel processes at the level of a simulation model. Two different projects implemented for the purposes of the course Parallel Computer Architecture. In order to test both projects results download folder and execute run.sh. Smith Waterman. In our project implementation the following are covered: Smith-Waterman algorithm implemented in C. Pthread implementation fine grained granularity. Specialized parallel computer architectures are sometimes used alongside traditional processors, for accelerating specific tasks. In some cases parallelism is transparent to the programmer, such as in bit-level or instruction-level parallelism, but explicitly parallel algorithms, particularly those that use concurrency, are more difficult to write than sequential ones,[7] because concurrency introduces several new classes of potential software bugs, of which race conditions are the most common.Â However, for a serial software programme to take full advantage of the multi-core architecture the programmer needs to restructure and parallelise the code. Hardware architecture of parallel computing â€“ The hardware architecture of parallel computing is disturbed along the following categories as given below : 1. Single-instruction, single-data (SISD) systems 2. Single-instruction, multiple-data (SIMD) systems 3. Multiple-instruction, single-data (MISD) systems 4. Multiple-instruction, multiple-data (MIMD) systems.Â Hardware computing â€“ Computer hardware is the collection of physical parts of a computer system. This includes the computer case, monitor, keyboard, and mouse.Â Get hold of all the important CS Theory concepts for SDE interviews with the CS Theory Course at a student-friendly price and become industry ready. My Personal Notes arrow_drop_up. Save.