

Dynamic Core Processors

Kyle Daruwalla, David McNeil, and Ben Schmidt
Rose-Hulman Institute of Technology

Abstract—As both general purpose, desktop-grade processors and embedded processors mature, the distinguishing gap between them continues to diminish. Out of this narrowing market, a need for a versatile, low-power core emerges. Multicore embedded processors are just over the horizon. Dynamic core processors attempt harness these low-power cores to maximize both parallel throughput and minimize serial latency.

Keywords—*IEEEtran, journal, L^AT_EX, paper, template.*

I. INTRODUCTION AND MOTIVATION

Mobile computing devices have grown to require processors that can support a dynamic workload. The typical serial tasks such as voice compression, speech transcoding (for communications), and image compression must perform well. However, newer mobile devices need to render complex graphics and run advanced background scheduling, all while maintaining serial performance and power. The obvious answer might be GPUs, but they are known to be power-hungry chips. Ideally, if most of the serial and parallel tasks could be performed on a single chip, the GPU would only need to be used to perform high TLP tasks like displaying graphics. A smaller GPU workload means a lower power GPU. Thus, power consumption can be kept minimal, while performance gains are still attained.

Dynamic core processors attempt solve precisely these issues. By adding an interconnection network, some additional hardware, and control logic, the symmetric multicore embedded processors can be reconfigured into a modern, super-scalar, out-of-order processor. Thus, by identifying serial and parallel program sections, the processor can be dynamically changed to perform efficiently.

Obviously, creating such a processor, though possible, is not a trivial task. So, we attempt to analyze the theoretical performance gains of a dynamic core as described above. The following work will show whether a dynamic core processor performs better than a symmetric multicore processor and a modern, super-scalar, out-of-order processor. Furthermore, for a fixed length program, we determine how often the program must switch from serial to parallel processing before a dynamic core processor realizes performance gains. Finally, we determine the maximum number of cores in a dynamic processor before critical path length negatively impacts serial performance.

II. PRIOR WORK

Summarize relevant related work with appropriate citations.

III. METHODOLOGY

Describe your methodology in implementing your project. If a simulator is used, cite it accordingly and detail what changes

you made. If a structure was implemented, give details as to what was done. If an algorithm was studied, give details as to its operation. Block diagrams, flow charts, and detailed examples can be very helpful.

IV. RESULTS

Provide and justify any quantitative results, preferably using graphs and/or tables.

V. CONCLUSION AND FUTURE WORK

Summarize your findings, identify remaining open problems or issues, and suggest the next steps for this project.

VI. STATEMENT OF WORK

The project report must also include a statement of work that identifies the contributions of each individual on the team. The statement of work must reflect a team consensus and must be signed by all team members. I recommend that you structure this statement as a table with a row for each project milestone, a column for each team member, and the percentage contribution of each team member to each milestone in the entries in the table. Please do not give a vanilla 50/50 in every column.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

John Doe

