Hitting Pollack's Law for Improving MPSoC Programmability and Efficiency*

Toshinori Sato Hideki Mori Rikiya Yano Takanori Hayashida
Fukuoka University
toshinori.sato@computer.org

This study aims to bridge the gap between multi-processor systems-on-chip (MPSoC) and software development techniques. While building parallelizing compilers is the ideal solution, some alternative but practical solutions are urgently required in developments of embedded systems. This is because time-to-market is an important factor to have a big success in the market.

Pollack's law explains performance is proportional to the square-root of the number of transistors. Hence, MPSoCs seem to be more efficient in aggregating performance than single-processor-based systems when transistor budget is large. On the other hand, Amdahl's law explains performance is dominated by sequential portions of the problem. MPSoC efficiency becomes worse as the number of cores increases. We found 17 times larger core achieves better performance than 17 small cores do when sequential portions occupy 20%.

Both programmability and Amdahl's law prefer moderate number of cores. This means single-thread performance is still important. Pollack's law also explains power is proportional to the number of transistors. To keep the advantage of MPSoC in power, we are investigating to improve single-core performance over Pollack's law by adapting core and chip configurations according to parallelism in programs.

The adaptability can consider the trade-off between performance, power, and dependability as well.

* This work is partially supported by the CREST (Core Research for Evolutional Science and Technology) program of Japan Science and Technology Agency (JST), and by Grant-in-Aid for Scientific Research (B) #20300019.

Hitting Pollack's Law

for Improving MPSoC Programmability and Efficiency

Toshinori Sato Hideki Mori Rikiya Yano Takanori Hayashida
Fukuoka University

Goal : To bridge the gap between MPSoC and software development

Background

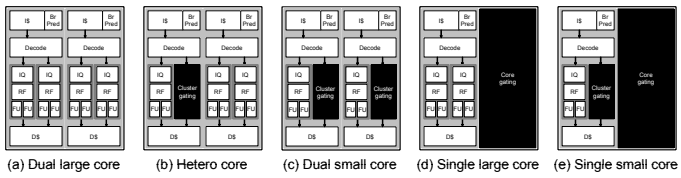
- Practical solutions for multicore programing is urgently required in development of enbedded systems.
- Pollack's law explains performance is proportional to the number of transistors.
- Amdahl's law explains performance is dominated by sequential portions of the program.

=> Single thread performance is still important.

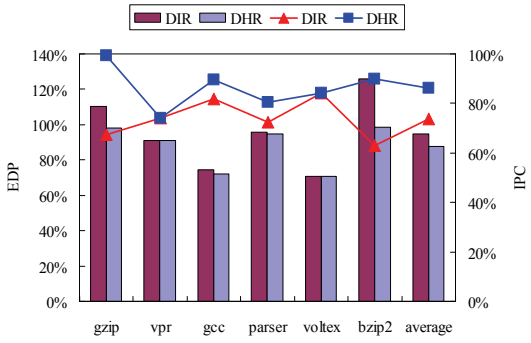
Ongoing work

- Multiple clustered-core processor
- Turbo boost for single core

Multiple Clustered-Core Processor

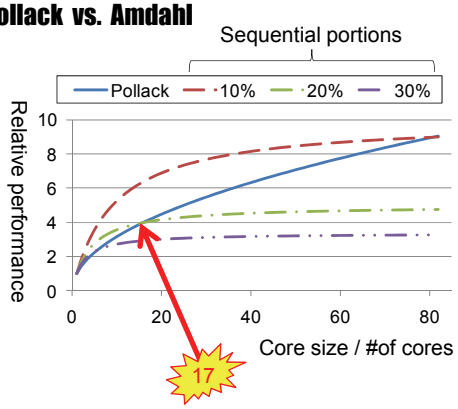


- Cluster gating & Core gating
  + Different scales and different configurations
- Consideration on trade-off between power & performance
  + Providing just required performance achieves power reduction
- DMR with 2 configurations
  + Thread-level and instruction-level redundancies



+ Hybrid DMR improves EDP by 13%.

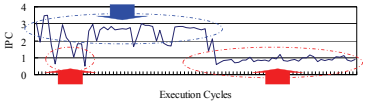
Pollack vs. Amdahl

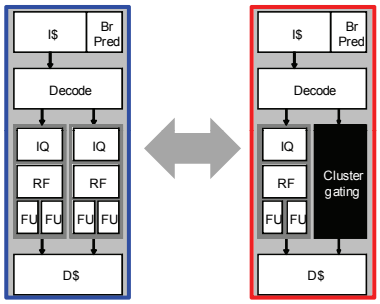


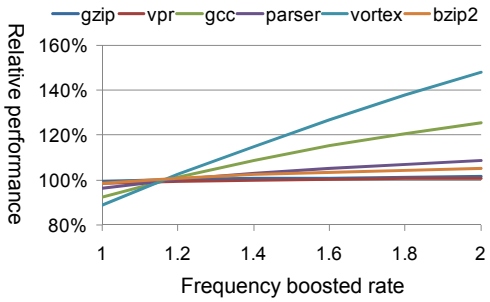
17 times larger core is better than 17 small cores when sequential portions occupy 20%.

Turbo boost for single core

- Scaling down configuration for high frequency
- Hybrid mode
  + Large core mode
    for large ILP but low frequency
  + Small core mode
    for high frequency but low ILP







Small core requires 1.6 times higher frequency to match large core.

+ Average of 5.0% performance gain is achieved when frequency is 1.4 time increased on the small core mode.

New directions

- Virtual single-core processor
  + Utilizing multiple cores as a single-core processor
- Reconfigurable fabric
  + Dynamically generating accelerators
- Critical path reduction
  + Exploiting hot spot
- Area and energy-aware memory modules
  + Utilizing ILP with low cost hardware

References

[1] Multiple clustered core processors, SASIMI, 2006.
[2] Power-performance trade-off on multiple clustered core processors, PRDC, 2007.
[3] Dependability, power, and performance trade-off on a multicore processor, ASP-DAC, 2008.
[4] Boosting Single Thread Performance by Utilizing Multiple Scale Core, IPSJ Kyushu Chapter Symposium, 2011.

These studies are supported in part by JST CREST and JSPS Kakenhi #20300019.