

Aaron Kobayashi

## 7 Reasons MLP can exceed ILP

- 1) Cache misses are extremely expensive. If cache misses are relatively frequent, ILP becomes useless. MLP attempts to maximize cache misses by overlapping the penalty as much as possible.
- 2) MLP simply attempts to maximize the number of outstanding cache misses. These misses can be generated by any processor architecture and therefore will make it possible to design a simpler (and hopefully less expensive) processor.
- 3) A simpler processor can be designed to execute at faster clock cycles. This is similar to the arguments proposed by the RISC architecture.
- 4) This more simple processor can be implemented in as simple a design as an in order execution processor. This will greatly reduce the transistors necessary to implement the design by reducing the logic necessary to complete an instruction. This reduction in transistor count will translate to reduced power consumption.
- 5) While ILP based processors may be able to continue to make gains in an environment where there are many threads going (and the processor can dedicate time spent waiting for I/O to other processes), MLP can make gains in the single thread environment. In many environments, when a user wants to execute something, they are only interested in how fast the requested action takes, not how much work on other processes can be achieved while waiting.
- 6) MLP systems may be able to reduce stalls on speculation. In an MLP system it may be possible to keep fetching and executing branches while forgetting about data depending on cache misses of "older" instructions.
- 7) Linked data structure traversal speed may be greatly increased with MLP. It may be possible to build skip lists in hardware. This can make parallel workloads that are dependent on cache misses thereby increasing the MLP of the processor.

## References:

- A. Glew, "MLP yes! ILP no!", Wild and Crazy Ideas Session, *8th International Conference on Architectural Support for Programming Languages and Operating Systems (ASLOS-VIII)*, October 1998.