

Aaron Kobayashi

## 10 factors that may limit the performance of the IA-64 architecture

- 1) Lower Frequency – The IA-64 claims to be less complex than traditional superscalar processors, but Intel has made no claims on its ability to support high frequencies that can increase the instruction throughput.
- 2) Predication – Predication depends on the existence of many functional units because approximately half the predicated instructions are discarded effectively wasting hardware, power, and processor calculations.
- 3) Explicit addressing – The IA-64 processors have only base registers and lack offset fields forcing the explicit computation of an address in advance.
- 4) Sign-Extending – The IA-64 has no sign-extended loads greatly increasing the path length
- 5) Integer Operations – The IA-64 has no integer multiply or divide instructions available. Instead it must explicitly copy data to and from the floating-point registers to perform these operations.
- 6) Instruction Bundle Placement – The IA-64 architecture places severe limitations on instruction placement which can cause a larger code footprint and/or more cycles wasted due to less compact code.
- 7) Pollution of Cache – Code bloat can pollute the instruction cache while speculative loads will pollute data cache and use bandwidth.
- 8) In order execution – In-order execution defined by the IA-64 can result in stalling the pipeline when a superscalar processor would not.
- 9) Recovery Code – To recover from speculative operations, recovery code needs to be used. This recovery code may not be in cache and consequently may cause page faults that would not have happened in a superscalar processor.
- 10) Dynamic Information – A superscalar machine makes extensive use of dynamic information while the IA-64 relies almost entirely on decisions made statically at compile time. Dynamic information can provide a wealth of data that is not available at compile time.

### References:

- M. Hopkins, "A Critical Look at IA-64," Microprocessor Report, February 7, 2000.