# Is the Multi-Core Roadmap going to Live Up to its Promises?"

**Tim Mattson,**

**Principal Engineer**

**Intel, Applications Research Laboratory**

# Disclaimers

- Part 1: legal matters
  - These slides are the personal views of the author and do not represent the views of the Intel Corporation.
  - While I’ve tried to be careful, they may include improper trademark usage or other nasty evil things. If that is the case, I apologize. I’m a nice guy … I don’t want to offend anyone or their trademarks.
- Part 2: Opinion Warning
  - These slides where used for a Panel. To keep things interesting, I’ve intentionally overstated opinions and taken some strong, contrary views. Please keep that in mind as you consider the views expressed in these slides.

# Yes, Multi-core will live up to its promises

- The promises of the many core Roadmap:
  - The memory bandwidth crisis will get worse even faster.
  - End users will be frustrated as the applications they care about don’t scale.
  - ISVs will mostly ignore many core since they need features they can sell … and performance isn’t a feature.
  - Programmers will suffer as they attempt in vain to parallelize software.
  - You’ll hear of new languages that make parallel programming easy and automatic parallelization. And as they did before, they will fail to work on real applications.

**In short, the multi-core roadmap is well positioned deliver on its promises of frustration, aggravation and irrelevance.**

# So lets break those promises

- Let’s try something radically new … let’s actually learn from the past.
- Software:
  - Stop creating new languages:
    - Understand what programmers truly need by studying how they think (joint C.S. work with experimental psychologists).
    - And then extend the languages we have today to create parallel languages that “get the job done”.
- Hardware:
  - Multiprocessor systems failed to go mainstream. Nothing’s changed. Shared address spaces don’t scale and are too hard to program. So what else should we do?
    - Stop moving SMP onto CMP. Go back to the drawing board and figure out what will really work. If you ask me, the lesson from the history of parallel computing is NoC* … not SMPoC*.

*NoC = network on a chip *CMP = chip multiprocessing *SMPoC = symm. Multiprocesing on a chip

# Conclusion

- Let's hope the multi-core roadmap doesn't deliver on its promises.
  - If it does, I can't put my children through college.
- We know what didn't work in the past, nothing's changed, so let's do it differently this time:
  - Hardware is irrelevant. Successful software is all that matters.
  - Parallel software research has largely failed, so let's do it different. Let's understand how programmers think, and then design the tools they need.
- Good news: We have some time to get it right.  Multi-tasking and coarse grained threading will be enough through 4-core and maybe to 8 core.  Beyond that, we're in trouble.