A comparison of manycore processors for media processing on mobile platforms

By Gordon Ononiwu

Supervised by Bertil Svensson

Preamble

- More than one billion cell phones sold each year (2006). This does not include the numbers for other mobile devices.
- Today's Mobile devices are expected to be multi-functional, incorporating heavy media applications.
- Bulk of computations today happen in High Performance Embedded Media (EM) Devices.
- Standards are continuously evolving.
- Algorithms are changing and getting ever more complex.
- Results in very little reuse.
- We face environmental and aggregate power consumption challenges.

Implementation Considerations for High Performance EM Applications

- High Performance and Hard Real time targets - They should be fast enough to meet demanding requirements.
- Application Mapping – Take advantage of available logic.
- Design productivity – Provide application developers with suitable abstractions.
- Support and tools.
- Manpower requirements – Require manageable skill set.
- Development time – Application need to be developed quickly enough to meet the market on time.
- Flexibility – Systems that are easily upgraded to provide additional functionalities - Encourage reuse
- Extend product life cycles and component reuse.
- Strict limits on size, power and cost.

Available Processing Options

- Hardwired/Semi-Hardwired solutions.
  - Hardware development productivity for ASICs and FPGAs lags far behind Moore's Law.
  - Long development time.
  - Variety of applications affects area efficiency.
  - Cost is not scaling with design complexity (High NRE cost).
  - For ASICs, a lack of flexibility, affecting reuse.
- Programmable solutions.
  - Highly reusable.
  - General purpose processors can not meet with the required performance.
  - They are not yet energy efficient.

Illustration

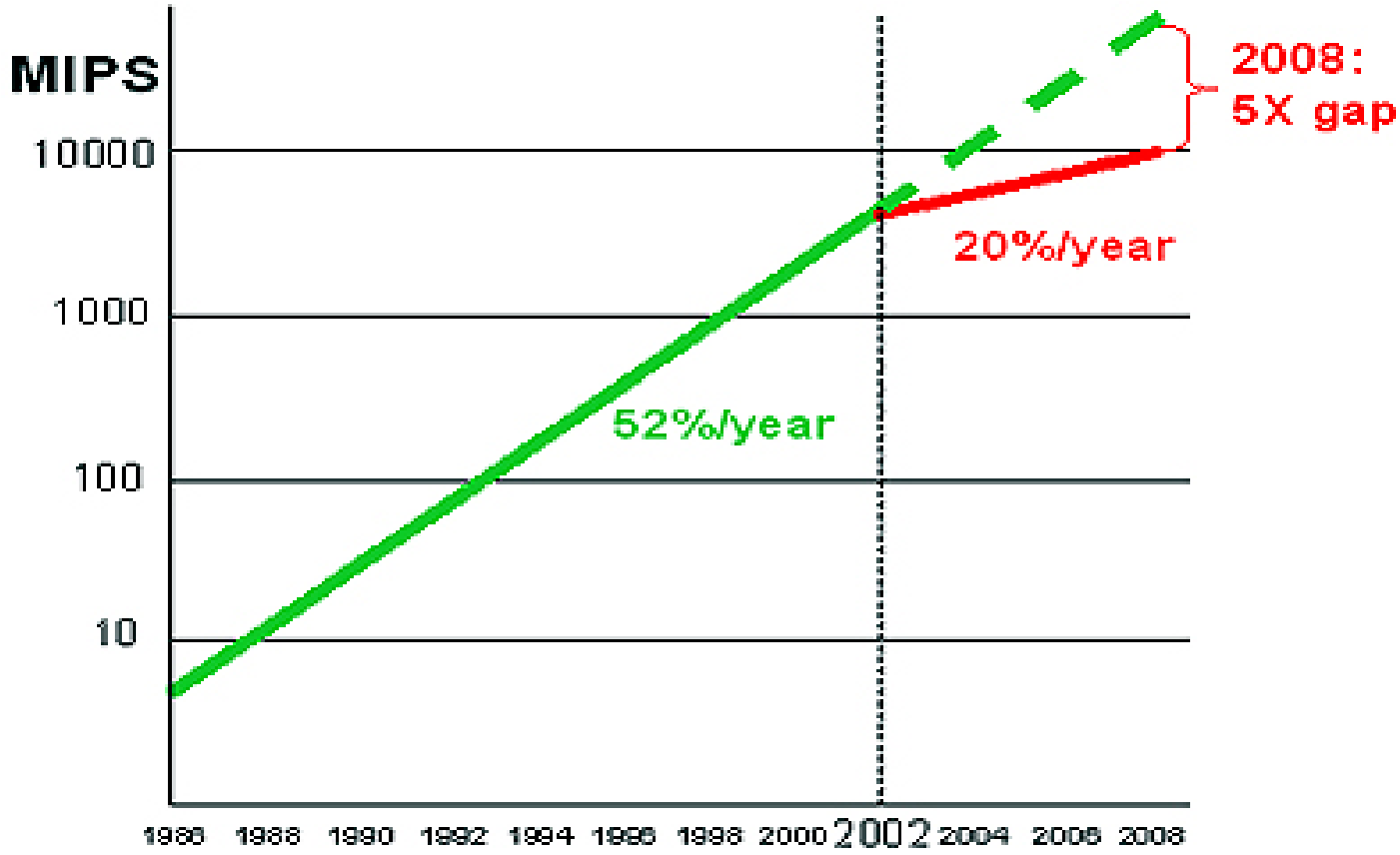

Fig 1: Processor Increase in speed over the past three decades. (From Hennessy & patterson, Comp. Arch. 4th ed., 2006)

Solutions

- Multicore's (A)
  - SMPs – Software developer has to partition the workload so that each core runs a portion of the application at all times.
  - Manage Inter-processor communication overhead.
  - Ensure that threads do not mistakenly affect the behavior of other threads.
  - Synchronization and Cache coherency issues.
  - Possibilities for 100s of cores.
- Manycore's (B)
  - No on-chip sharing of memory resources.
  - Strictly encapsulated processors.
  - High predictability – code produces the same result each time it is run.
  - An application maps into functions which map into separate processors – Functional parallelism.
  - With the right interconnection network, possibilities for energy scalability.
  - Possibilities for 1000s of cores.

Expectations

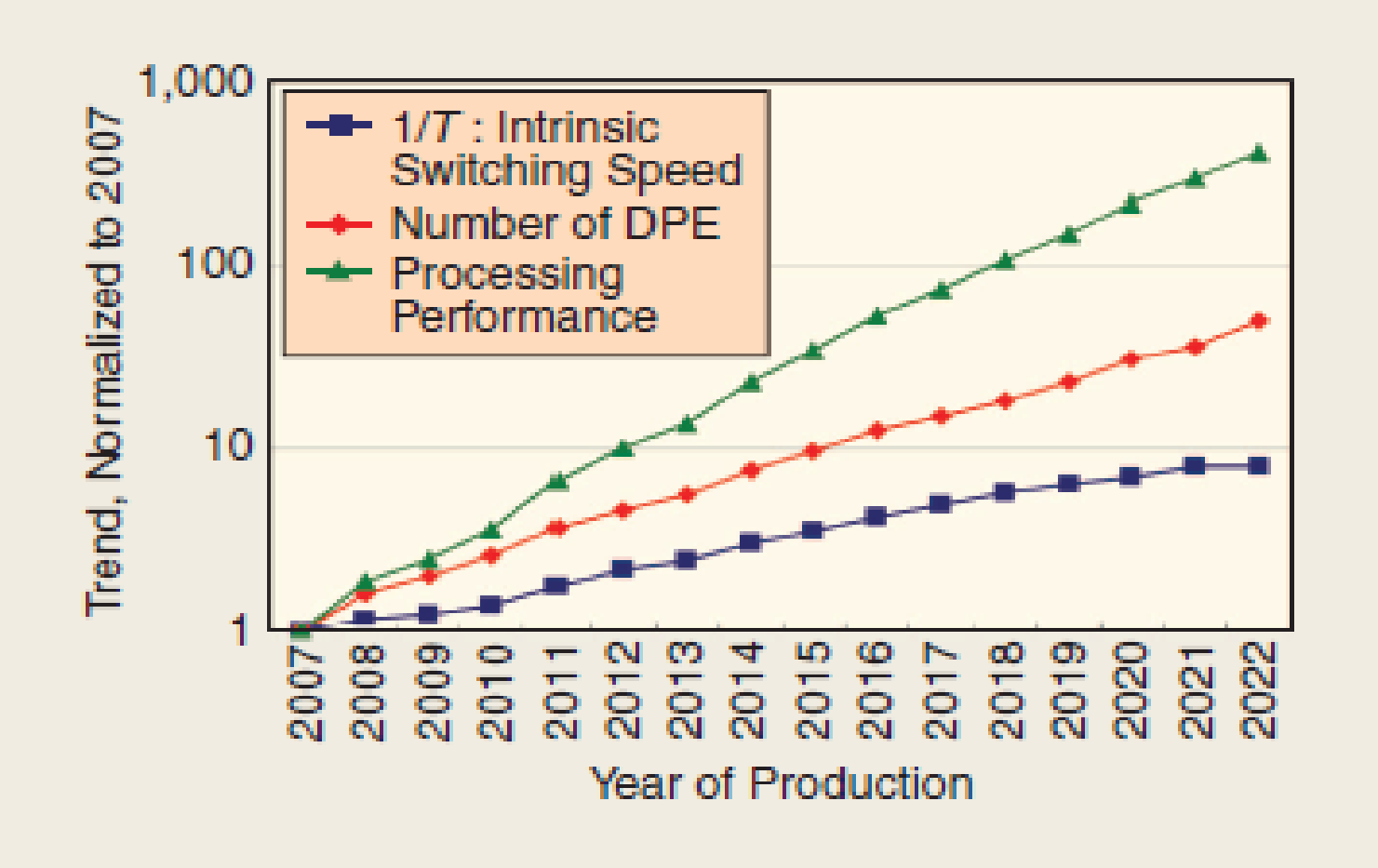

Fig 2: The ITRS Roadmap for frequency, number of data processing elements (DPE) and overall performance.

Interesting Architectures for the Future

- Ambric Massively Parallel Processor Array (MPPA).
- Stanford Efficient Low-Power Microprocessor (ELM).
- Tilera's TILE GX processor family/TILE64 processor.
- Intel's 80-core network-on-a-chip terascale processor.
- Pico Chip

Categorization

- Micro architecture and its realization.
- Number of cores.
- Core complexity.
- Maximum and minimum power.
- Throughput in terms of ops/cycle.
- Operating frequency.
- Interconnection.
- Programming model.
- Communication scheme.
- Synchronization.
- Energy usage pattern.
- Application domain.
- Clocking – Asynchronous or synchronous.
- Long term scalability – 100’s of core/ 1000’s of cores.

Conclusion

- We are not interested in explicitly declaring which processor is better.
- 1000s and not 100s
- What is a manycore processor?
- Energy scalability.
- Maximize the use of available logic.
- Programming models should abstract from the number of available cores
- We argue for greater developer productivity, better mapping tools, higher programmer abstractions from the hardware, faster turn around time, longer product cycles, flexible and software upgradable solutions.

Thank You.