



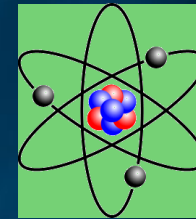
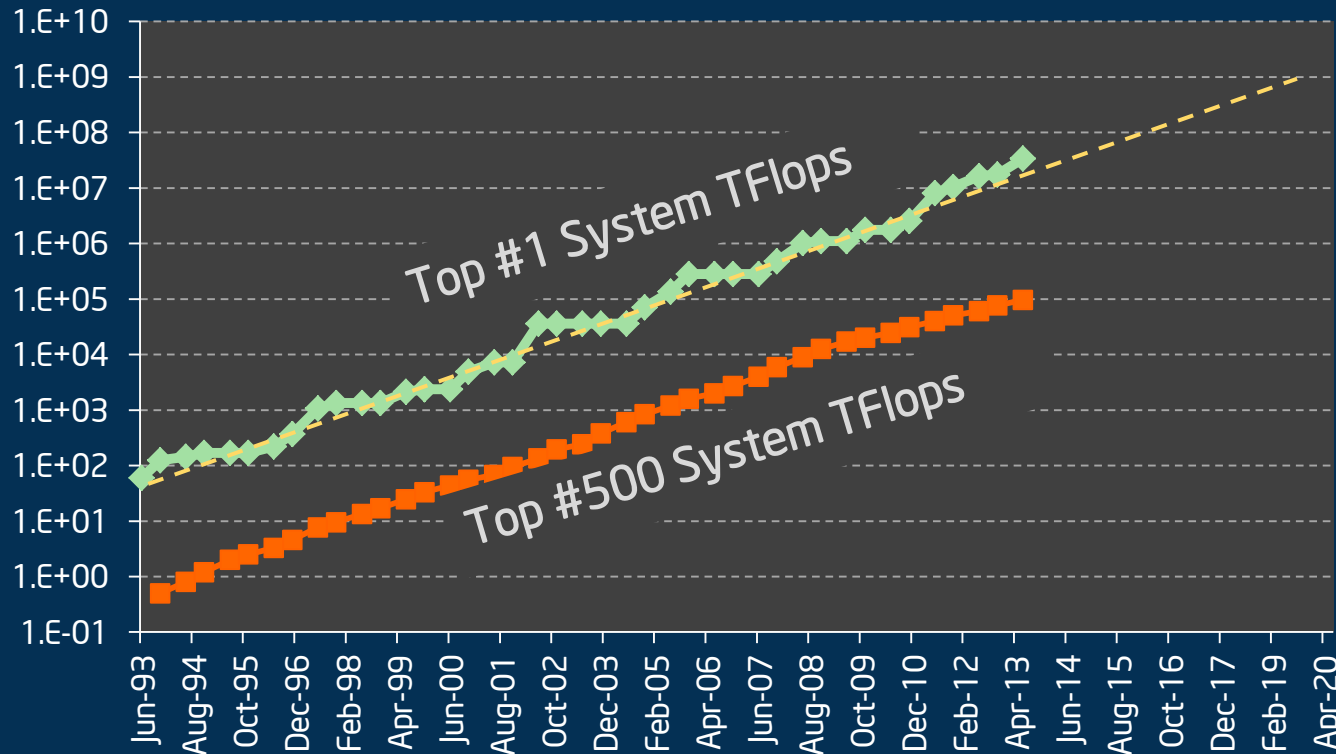
# Exascale Challenges and General Purpose Processors

Avinash Sodani, Ph.D.

Chief Architect, Knights Landing Processor

Intel Corporation

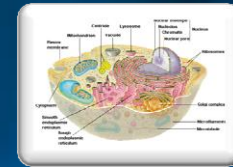
# Exponential Compute Growth



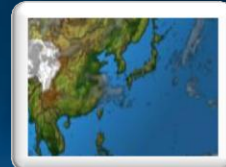
Understand  
Universe



Energy



Health



Weather

- Appetite for compute will continue to grow exponentially
- Fueled by the need to solve many fundamental and life changing problems.

# Many Challenges to Reach Exascale

- **Power efficiency** → Fit in Data Center power budget
- **Space efficiency** → Fit in available floor space
- **Memory technology** → Feed compute power-efficiently
- **Network technology** → Connect nodes power-efficiently
- **Reliability** → Control the increased probability of failures
- **Software** → Utilize the full capability of hardware

And more ...



# Challenge 1: Compute Power

At System Level:

Today: 33 PF, 18 MW → 550 pJ/Op

Exaflop: 1000 PF, 20 MW → 20 pJ/Op

Needs improvements in all system components

Processor-subsystem needs to reduce to 10 pJ/Op

~28x improvement needed for Exascale by 2018/19

# Challenge 2: Memory

Memory bandwidth fundamental to HPC performance

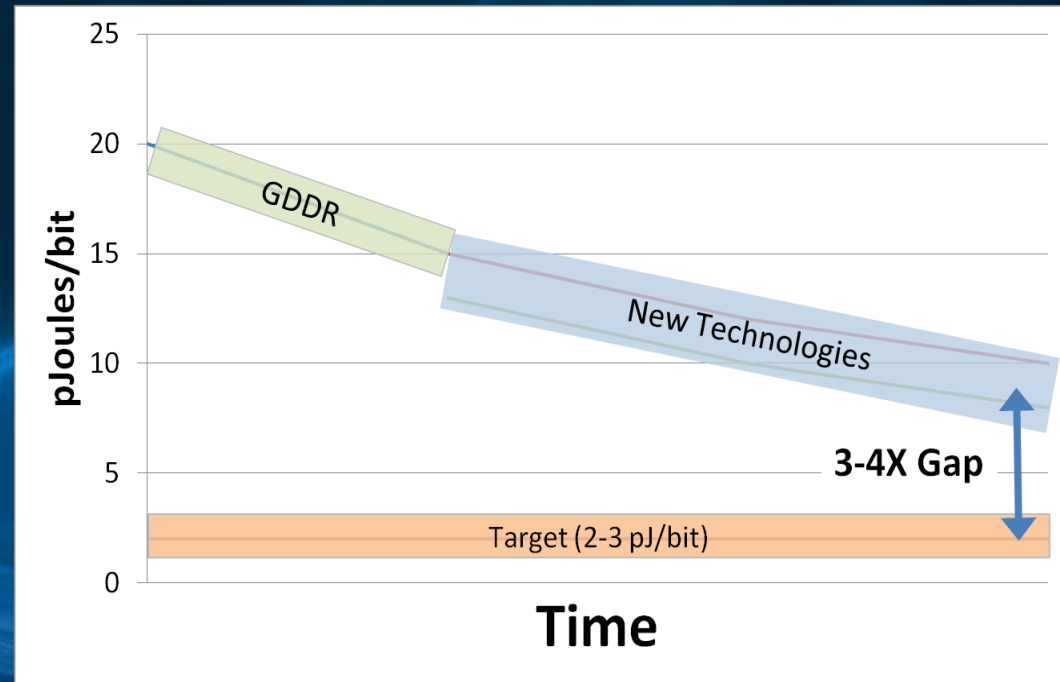
- Need to balance with capacity and power

1 ExaFlop Machine

→ ~200-300 PB/sec

→ ~2-3 pJ/bit

- GDDR will be out of steam
- Periphery connected solutions will run into pin count issues



Existing technology trends leave 3-4x gap on pJ/bit

# Power: Commonly Held Myths

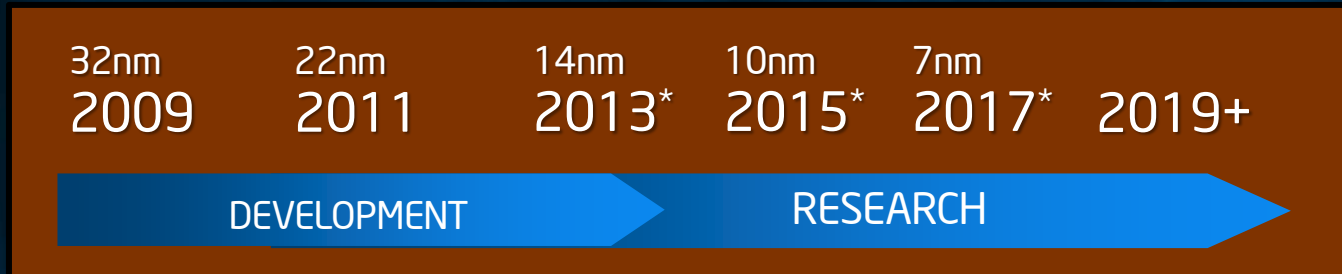
- General-purpose processors cannot achieve the required efficiencies. Need special-purpose processors.
- Single thread performance features and legacy features too power hungry
- IA memory model, Cache Coherence too power hungry
- Caches don't help for HPC → They waste power



## Myth 1:

General-purpose processors cannot achieve the required efficiencies. Need special-purpose processors

# Performance/Power Progression



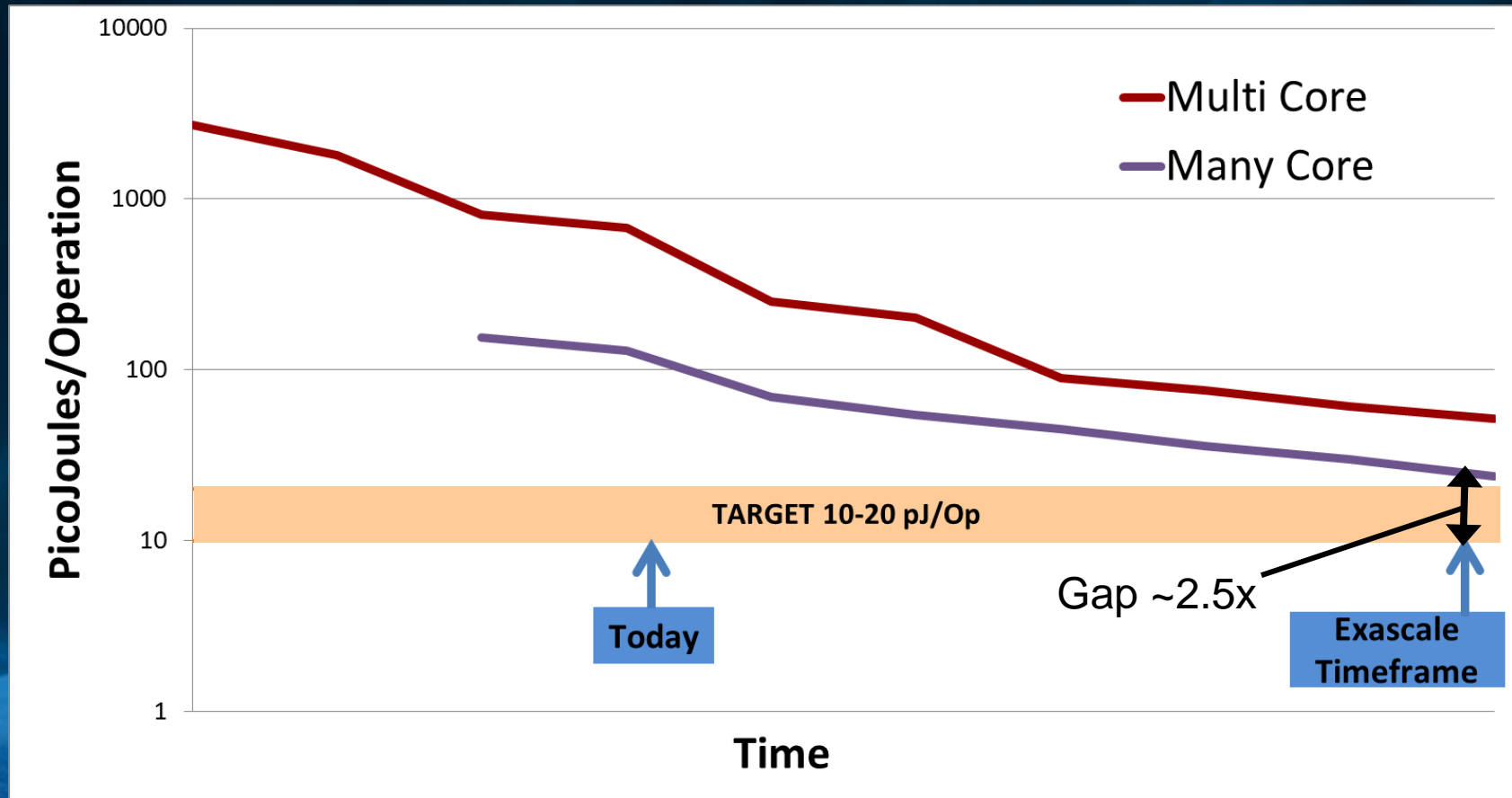
Moore's Law scaling continues to be alive and well

- Process: 1.3x - 1.4x (per generation)
- Arch/Uarch: 1.1x - 2.0x (per generation)

Recurring improvement: 1.4 - 3.0x every 2 years



# Energy/Op Reduction over Time

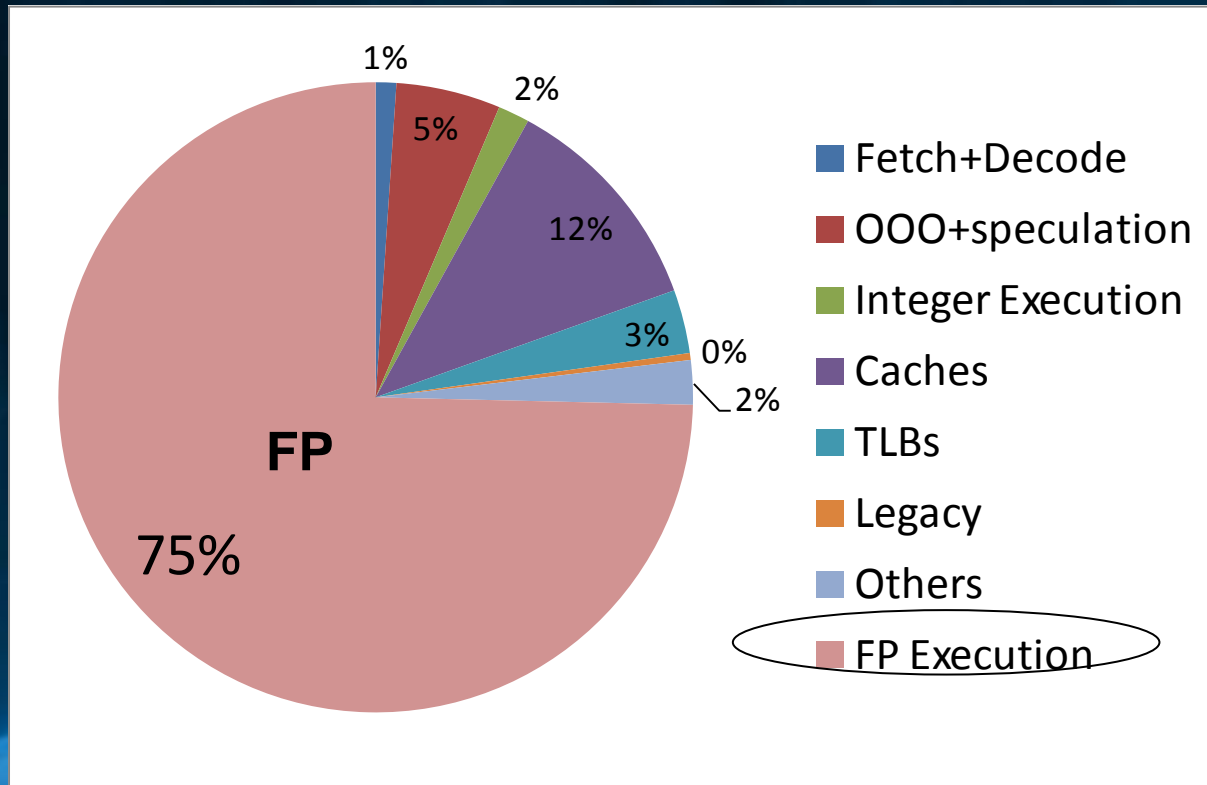


**Gap reduces to ~2.5x from ~30x with existing techniques!**  
 Do not need special purpose processing to bridge this gap

## Myth 2:

Single thread performance features and legacy features too power hungry

# Typical Core-Level Power Distribution

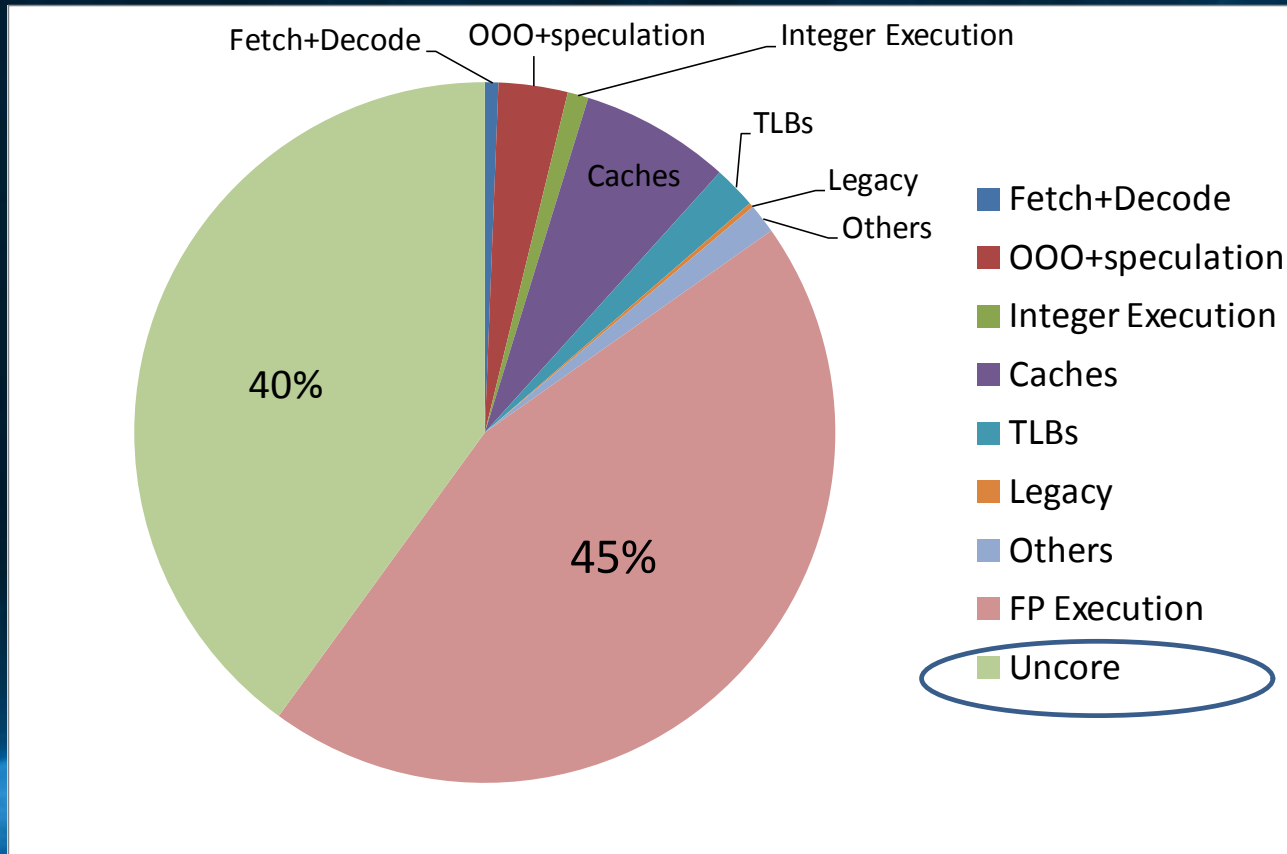


Floating Pt  
Compute-heavy  
Application

- Power dominated by compute – as should be the case
- OOO/Speculation/TLB: < 10%
- X86 Legacy+Decode = ~1%



# Typical Chip-level Power Distribution

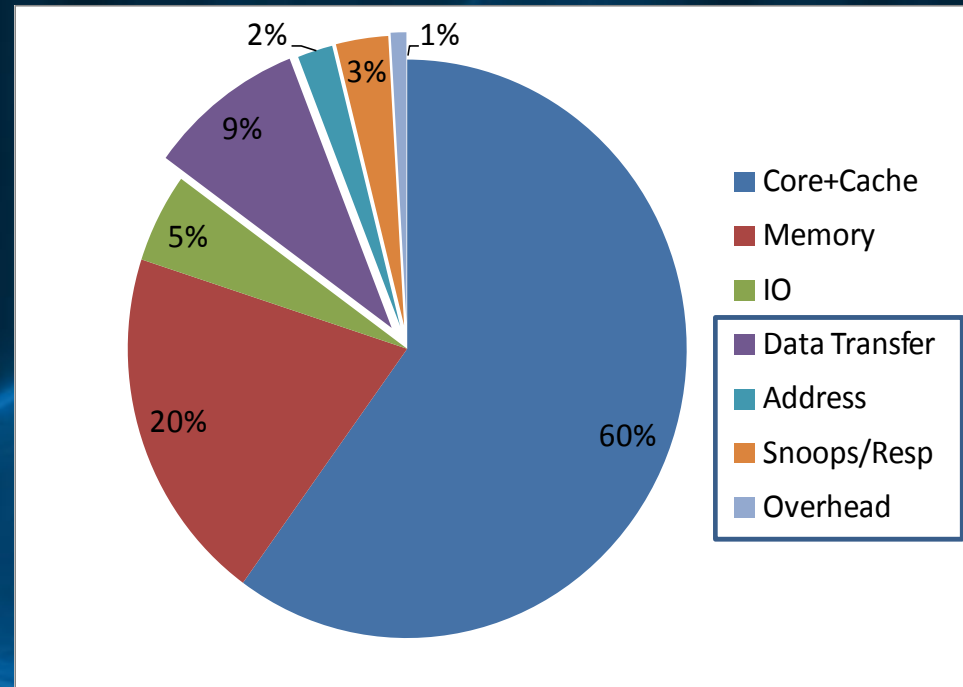
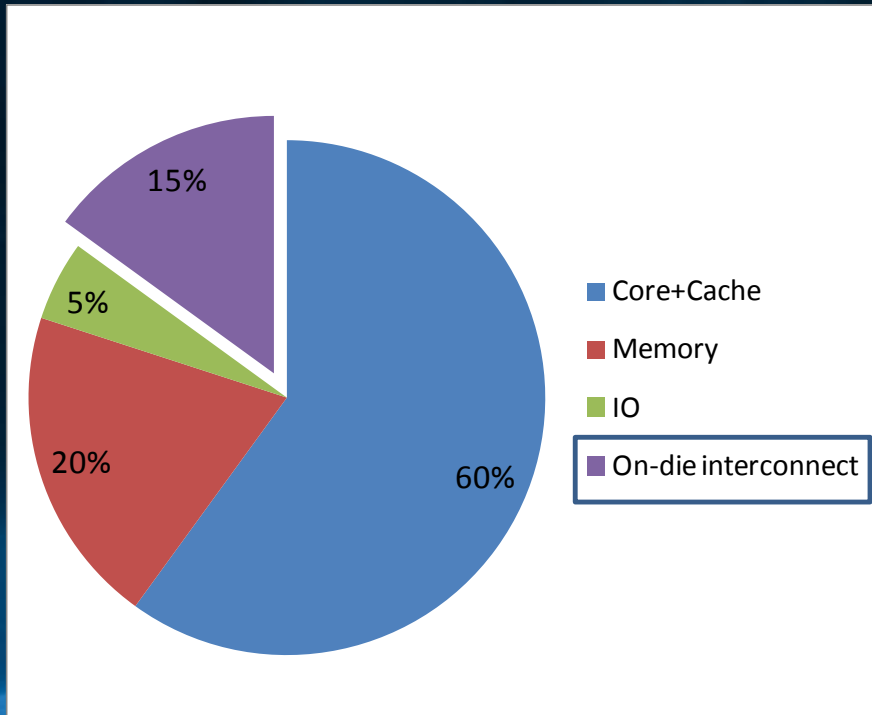


At chip level core power is even smaller portion (~15%).  
X86 support, OOO, TLBs ~6% of the chip power  
Benefits outweigh the gains from removing them

Myth 3:

IA memory model, Cache Coherence too power hungry

# Coherency Power Distribution



- Typically coherency traffic is 4% of total power
- Programming benefits outweigh the power cost

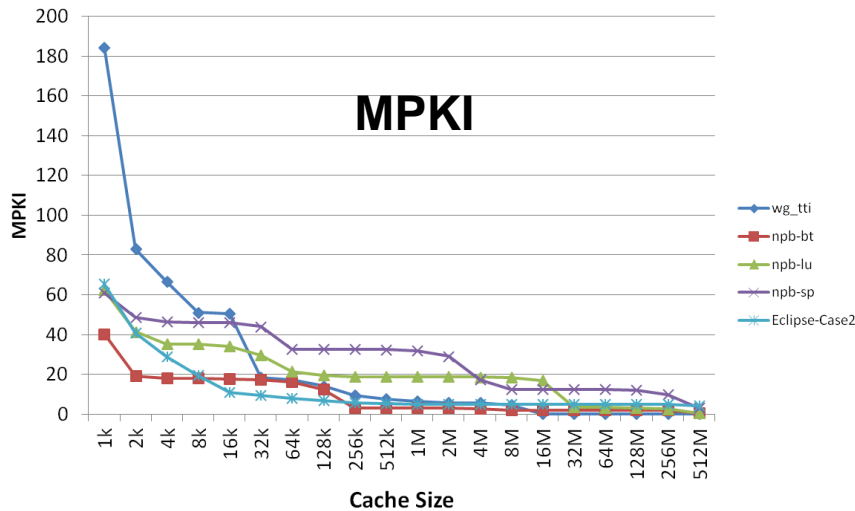


Myth 4:

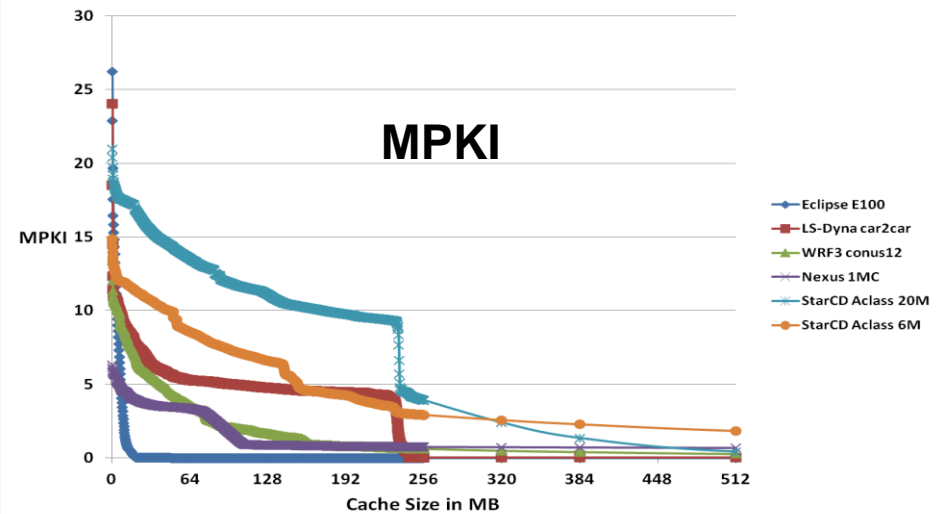
Caches don't help for HPC → They waste power

# MPKI in HPC Workloads

Miss per Thousand Instructions

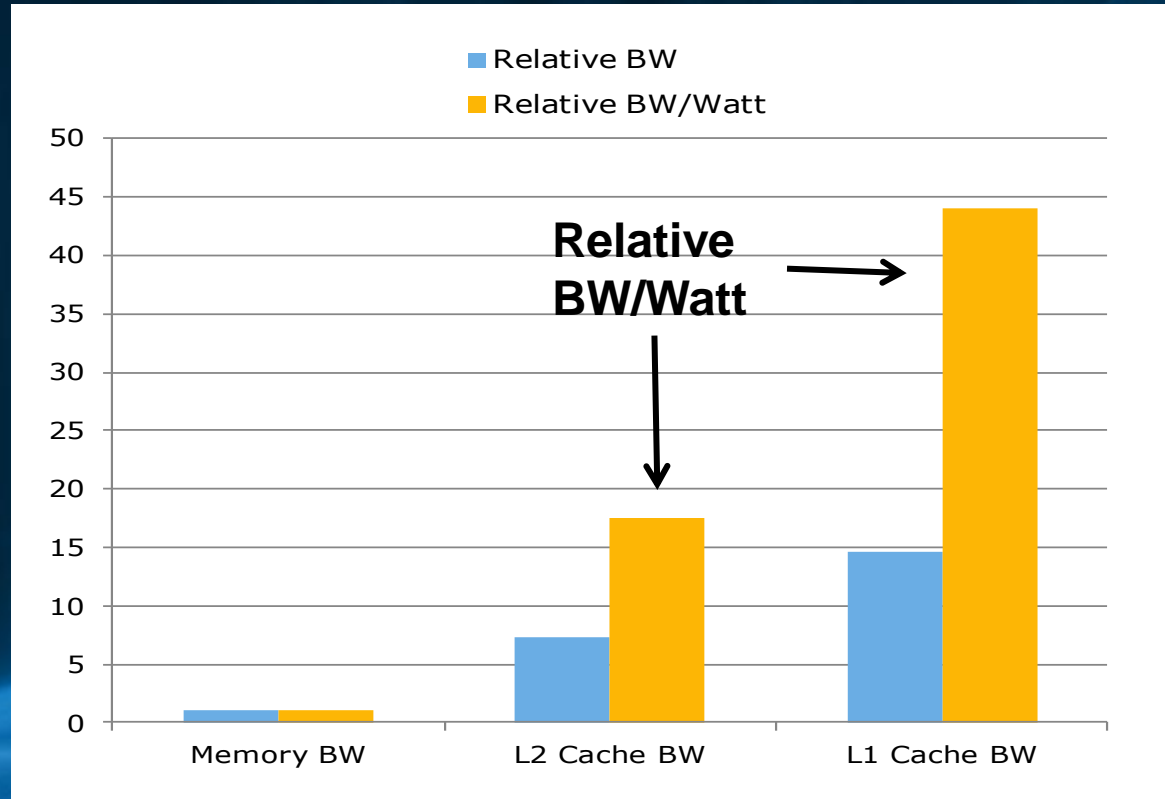


Miss per Thousand Instructions (MPKI) vs. Cache Size  
values beyond 256MB are estimated using trend analysis



- Most HPC workloads benefit from caches
- Less than 20 MPKI for 1M-4M caches

# Caches save power



- Caches save power since memory communication avoided
- Caches 8x-45x better at BW/Watt compared to memory
- Power break-even point around 11% hit rate (L2 cache)



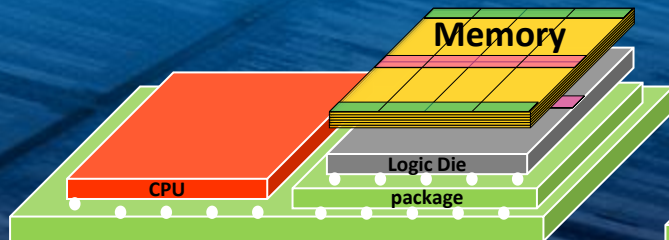


# General purpose processors can achieve Exascale power efficiencies

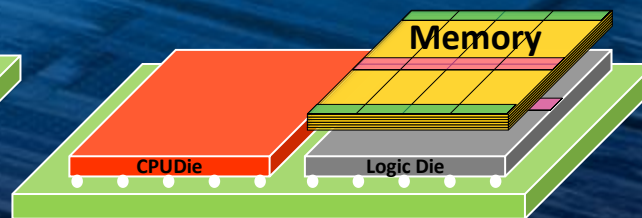
# Memory: Approach Fwd

- Significant power consumed in Memory
  - Need to drive 20 pJ/bit to 2-3 pJ/bit
- Balancing BW, capacity and power is hard problem
- More hierarchical memories
- Progressively more integration

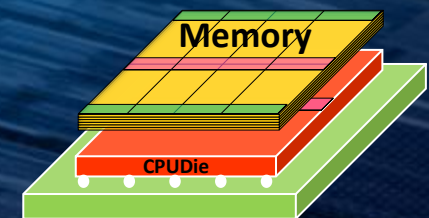
Multi-package Usage



Multi-chip Package Usage



Direct Attach Usage



# Next Intel® Xeon Phi™ Processor:

## Knights Landing



Designed using  
Intel's cutting-edge  
**14nm process**

Not bound by "offloading"  
bottlenecks  
***Standalone CPU or  
PCIe coprocessor***

Leadership compute & memory  
bandwidth  
**Integrated  
on-package memory**



# Benefits of General Purpose Programming

## Familiar SW tools

- New languages/models not required

## Familiar programming model

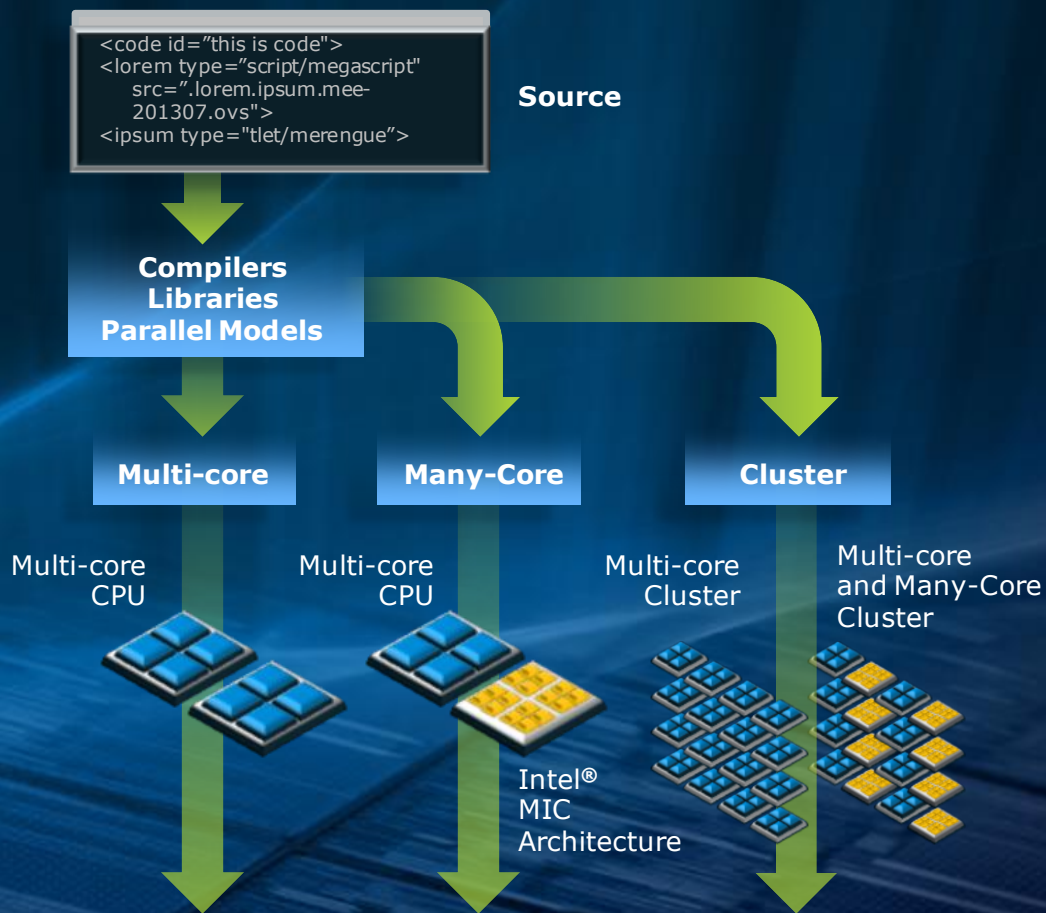
- MPI, OpenMP

## Maintain single code base

- Same SW can run for multi-cores and many-cores™

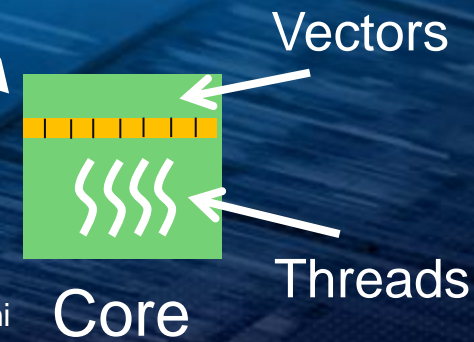
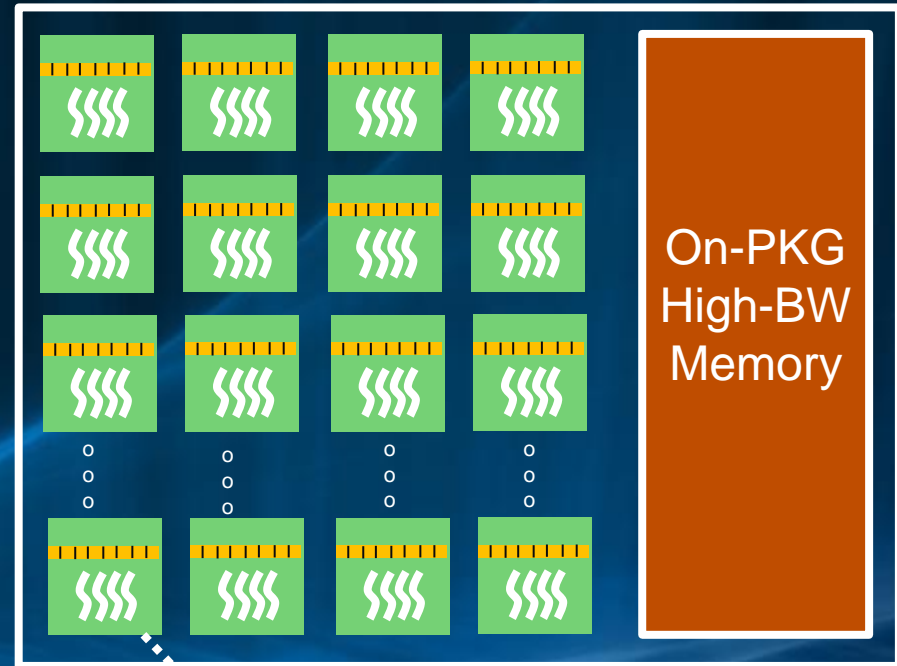
## Optimize code just once

- Optimizations for many cores improve performance for multi-core as well



# Future Xeon-Phi

- Lots of Wide Vectors
- Many IA Cores
- Lots of IA Threads
- Coherent Cache Hierarchy
- Large on-PKG high-bandwidth Memory in addition to DDR
- Standalone general purpose CPU – No PCIe overhead



# What Does It Mean For Programmers

Existing CPU SW will work, but effort needed to prepare SW to utilize Xeon-Phi's full compute capability.

- Expose parallelism in programs to use all cores
  - MPI ranks, Threads, Cilk+
- Remove constructs that prevent compiler from vectorizing
- Block data in caches as much as possible → Power efficient
- Partition data per node to maximize on-Pkg memory usage

Code remains portable. Optimization improves performance on Xeon processor as well.



# Summary

Many challenges to reach Exascale – Power is one of them

General purpose processors will achieve Exascale power efficiencies

- Energy/op trend show bridgeable gap of ~2x to Exascale (not 50x)

General purpose programming allows use of existing tools and programming methods.

Effort needed to prepare SW to utilize Xeon-Phi's full compute capability. But optimized code remains portable for general purpose processors.

More integration over time to reduce power and increase reliability



