ECE 697J – Advanced Topics in Computer Networks

Design Trade-Offs in Network Processors

10/23/03

NP Architectures

- Numerous different design goals
  - Performance
  - Cost
  - Functionality
  - Programmability
- Numerous different system choices
  - Use of parallelism
  - Types of memories
  - Types of interfaces
  - Etc.
- We consider
  - Design tradeoffs on high level (qualitative tradeoffs)
  - Impact of different configurations on one particular architecture (quantitative tradeoffs)

Design Tradeoffs (1)

- Low development cost vs. performance
  - ASICs give higher performance, but take time to develop
  - NPs allow faster development, but might give lower performance
- Programmability vs. processing speed
  - Similar to tradeoff between ASIC and NP
  - Co-processors pose the same tradeoffs
  - Complexity of instruction set
- Performance: packet rate, data rate, and bursts
  - Difficult to assess the performance of a system
  - Even more difficult to compare different systems
- Per-interface rate vs. aggregate data rate
  - NP usually limited to one port

Design Tradeoffs (2)

- NP speed vs. bandwidth
  - How much processing power per bandwidth is necessary?
  - Depends on application complexity
- Coprocessor design: lookaside vs. flow-through
  - Lookaside: “called” from main processor, need state transfer
  - Flow-through: all traffic streams through coprocessor
- Pipelining: uniform vs. synchronized
  - Pipeline stages can take different times
  - Tradeoff between slowing down or synchronization
- Explicit parallelism vs. cost and programmability
  - Hidden parallelism is easier to program
  - Explicit parallelism is cheaper to implement

Design Tradeoffs (3)

- Parallelism: scale vs. packet ordering
  - Why is packet order important?
  - Giving up packet order constraint gives better throughput
- Parallelism: speed vs. stateful classification
  - Shared state requires synchronization
  - Limits parallelism
- Memory: speed vs. programmability
  - Different types of memories give performance
  - Increases difficulty in programming
- I/O performance vs. pin count
  - Packaging can be major cost factor
  - More pins give higher performance

Design Tradeoffs (4)

- Programming languages
  - Ease of programming vs. functionality vs. speed
- Multithreading: throughput vs. programmability
  - Threads improve performance
  - Threads require more complex programs and synchronization
- Traffic management vs. blind forwarding at low cost
  - Traffic management is desirable but requires processing
- Generality vs. specific architecture role
  - NPs can be specialized for access, edge, core
  - NPs can be specialized towards certain protocols
- Memory type: special-purpose vs. general-purpose
  - SRAM and DRAM vs. CAM

Design Tradeoffs (5)

- Backward compatibility vs. architectural advances
  - On component level: e.g., memories
  - On system level: NP needs to fit into overall router system
- Parallelism vs. pipelining
  - Depends on usage of NP
- Summary:
  - Lots of choices
  - Most decisions require some insight in expected NP usage
  - Tradeoffs are all qualitative
- Consider quantitative impact of NP configuration!

A Network Processor Performance and Design Model with Benchmark Parameterization

Mark A. Franklin
Tilman Wolf

Challenges in NP Design

- Need for powerful network processors
  - Increasing link speeds
  - Increasing application complexity
- NPs different from other processors
  - NPs can exploit much more parallelism
- Vast design space
  - How many processors, how much cache, how many I/O and memory channels?
  - General-purpose vs. specialized processors
- Performance models of traditional processors do not apply

- => We propose performance model specific to NPs

Contribution

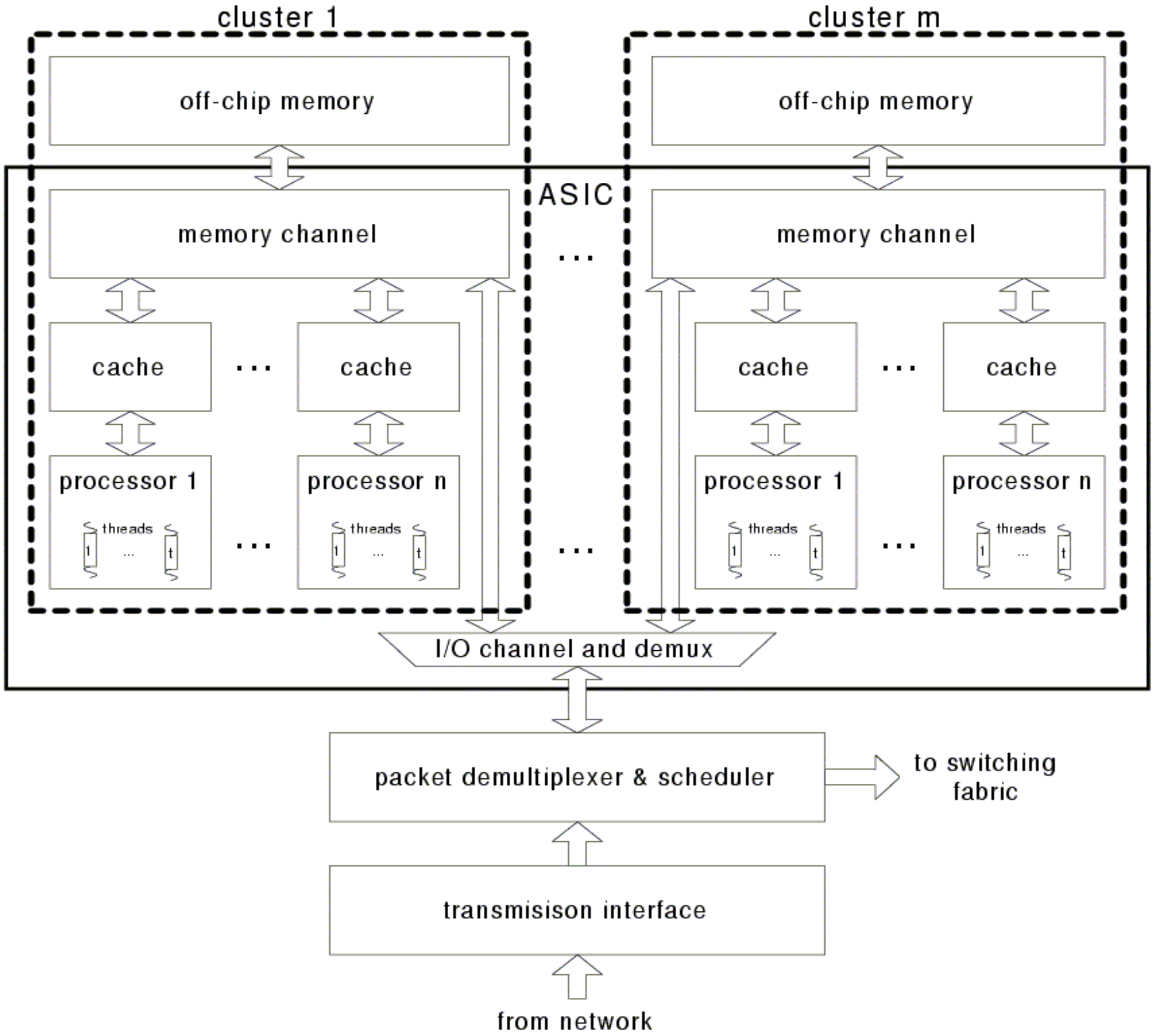
- General NP system model
  - Makes use of parallelism in network workloads
  - Applicable to a broad range of NPs
- Analytical performance model
  - Measure of processing power of NP configuration
  - Measure of cost in terms of chip area used
- Optimization of NP configuration
  - Model used to maximize MIPS/area
  - General design tradeoffs (e.g., # threads, cache sizes, …)
- General philosophy:
  - "If area is not used efficiently, then it might as well be used for another parallel processing engine instead."

Outline

- General NP system
- Performance model
  - Performance definition
  - Processor utilization
  - Memory system
  - I/O channel
  - Area cost
- Application benchmark
  - Parameterization of model
- Optimization results
- Summary

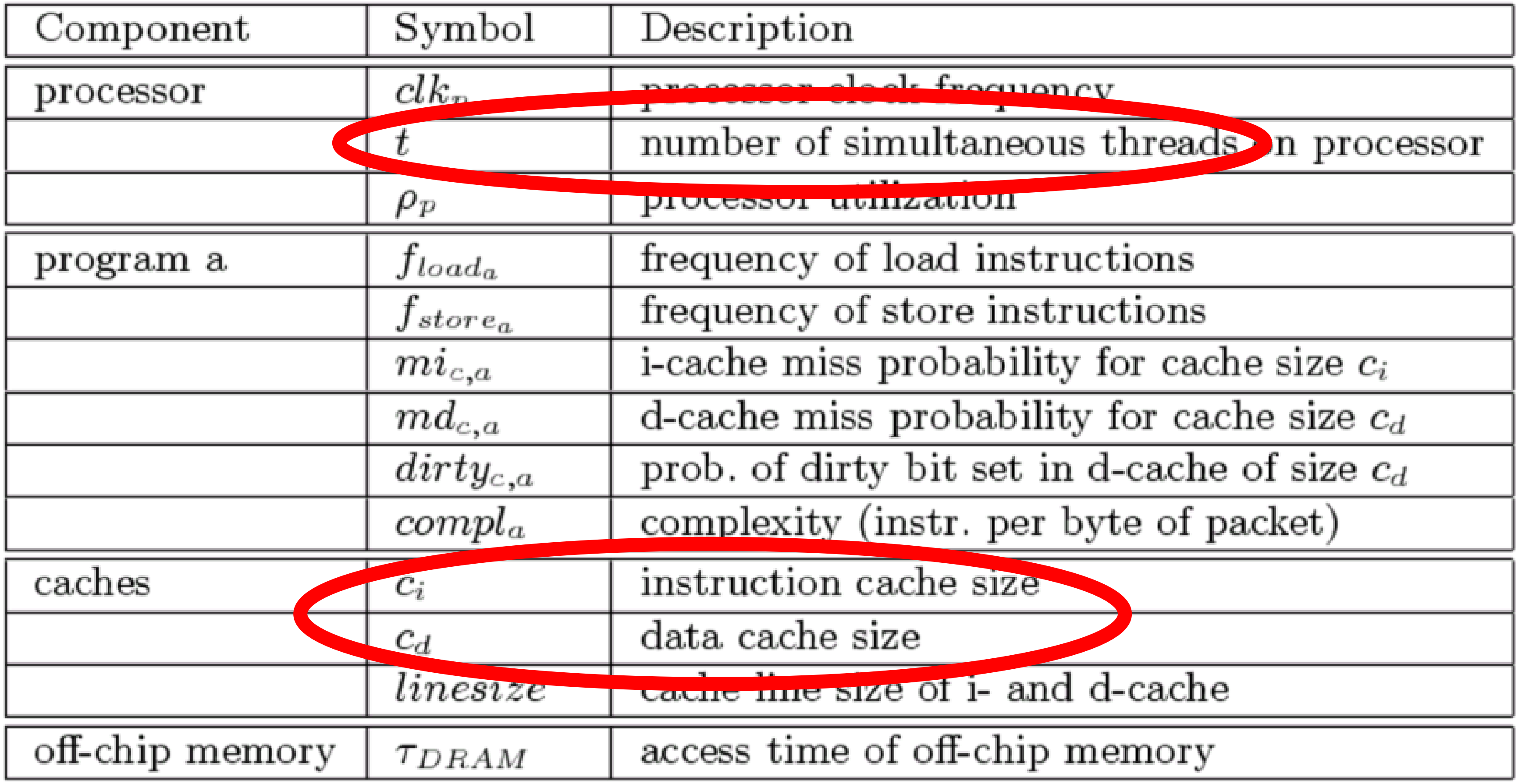
NP System Model

- Single Chip Multi-processor
- Clusters:
  - Processors
  - Per-proc cache
  - Memory channel
- Processors are simple RISC cores
- Off-chip router functions:
  - Queuing
  - Packet demux



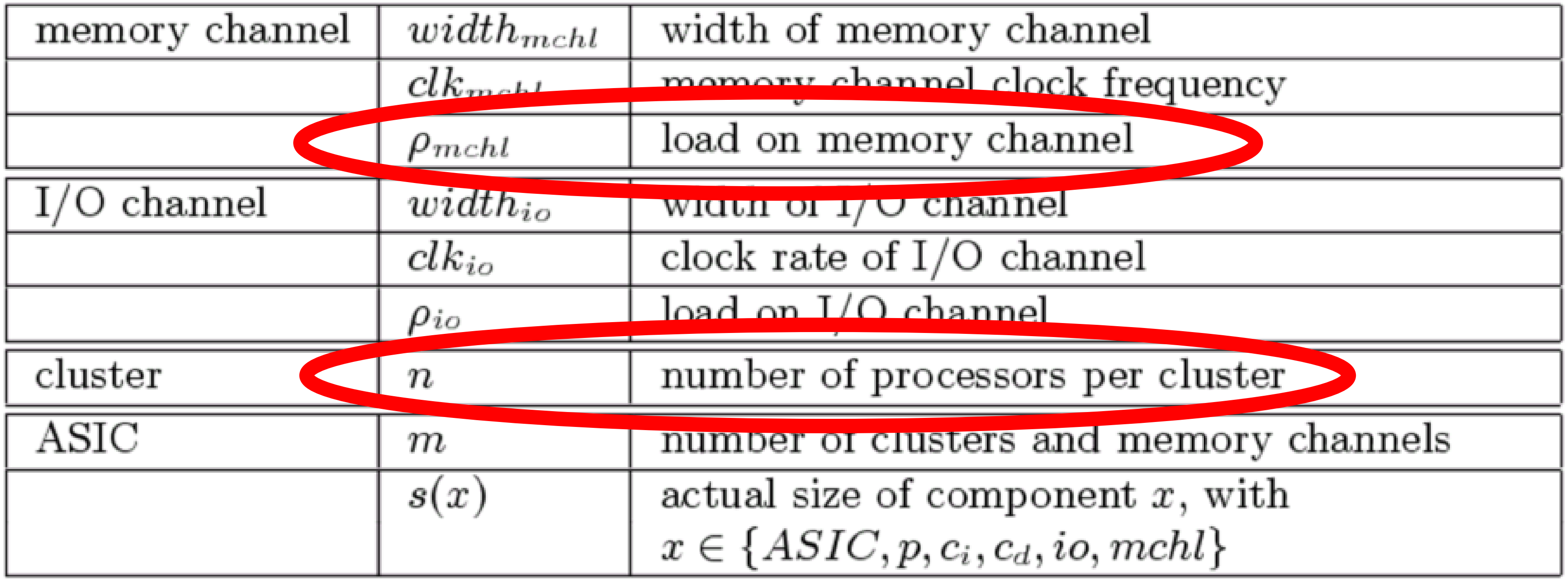
Design Parameters (1)

- Parameters that are considered in model:



Component	Symbol	Description
processor	clk_p	processor clock frequency
	t	number of simultaneous threads on processor
	ρ_p	processor utilization
program a	f_{load_a}	frequency of load instructions
	f_{store_a}	frequency of store instructions
	$mi_{c,a}$	i-cache miss probability for cache size c_i
	$md_{c,a}$	d-cache miss probability for cache size c_d
	$dirty_{c,a}$	prob. of dirty bit set in d-cache of size c_d
	$compl_a$	complexity (instr. per byte of packet)
caches	c_i	instruction cache size
	c_d	data cache size
	$linesize$	cache line size of i- and d-cache
off-chip memory	τ_{DRAM}	access time of off-chip memory

Design Parameters (2)



memory channel	$width_{mchl}$	width of memory channel
	clk_{mchl}	memory channel clock frequency
	ρ_{mchl}	load on memory channel
I/O channel	$width_{io}$	width of I/O channel
	clk_{io}	clock rate of I/O channel
	ρ_{io}	load on I/O channel
cluster	n	number of processors per cluster
ASIC	m	number of clusters and memory channels
	$s(x)$	actual size of component x, with $x \in \{ASIC, p, c_i, c_d, io, mchl\}$

- Develop performance model:
  1. Processor utilization
  2. Cache miss rate and memory access time
  3. Memory channel utilization
  4. Cluster configuration

Processing Power

- RISC: one instruction every cycle unless stalled
- Utilization ρ_p gives fraction of "useful" cycles
- Total processing power:

$$IPS = \sum_{j=1}^{m} \sum_{k=1}^{n} \cdot \rho_{p_{j,k}} \cdot clk_{p_{j,k}}$$

- If all processors are identical in configuration and workload:

$$IPS = m \cdot n \cdot \rho_p \cdot clk_p$$

- Question: How to determine ρ_p?

Processor Utilization

- Cache misses cause processor stalls
  - Reduce utilization
- Multithreading hides memory access latencies
- Processor utilization [Agarwal 1992]:

$$\rho_p(t) = 1 - \frac{1}{\sum_{i=0}^{t} \left(\frac{1}{p_{miss} \cdot \tau_{mem}}\right)^i \frac{t!}{(t-i)!}}$$

- Utilization decreases with
  - more cache misses (p_{miss})
  - longer memory accesses (τ_{mem})
  - Fewer threads (t)
- Need to determine τ_{mem} and p_{miss}

Memory System

- Memory access time has three components:
  - Queuing time until request is served
  - DRAM access time
  - Memory line transmission time

$$\tau_{mem} = \tau_Q + \tau_{DRAM} + \tau_{transmit}$$

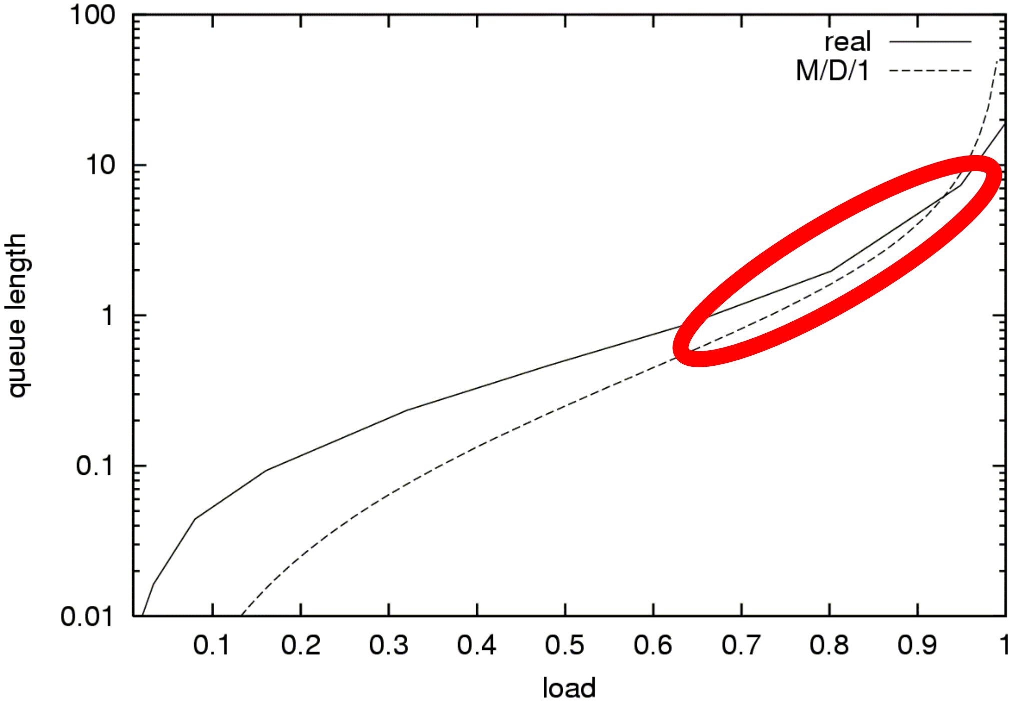
- DRAM access time fixed by technology used.
- Transmission time:

$$\tau_{transmit} = \frac{linesize}{width_{mchl}} \cdot \frac{clk_p}{clk_{mchl}}$$

- Queuing time depends on load on memory channel.

Queuing Approximation

- Processors in cluster generate memory requests
  - Single server queuing system
  - Deterministic service time
  - Geometrically distributed inter-request time



- Approximation with waiting time in M/D/1 queue:

$$\tau_Q = \frac{\rho_{mchl}^2}{2(1-\rho_{mchl})} \cdot \frac{linesize}{width_{mchl}} \cdot \frac{clk_p}{clk_{mchl}}$$

On-Chip Caches

- Miss rate is combination of i-cache and d-cache misses:

$$p_{miss,a} = mi_{c,a} + (f_{load_a} + f_{store_a}) \cdot md_{c,a}$$

- Miss rates of application depend on effective cache size.
- Threads compete for cache => cache pollution
- Cache is effectively split among threads.
  - Effective cache size:

$$c_{i,\text{eff}} = \frac{c_i}{t}, \quad c_{d,\text{eff}} = \frac{c_d}{t}$$

- We now have expression for processor utilization.

Memory and I/O Channel

- How many processor can share one memory channel?
- Processor utilization and miss rates gives memory bandwidth bw$_{mchl,1}$ of one processor.
- Number of processors that can share memory channel:

$$n = \left\lfloor \frac{width_{mchl} \cdot clk_{mchl} \cdot \rho_{mchl}}{bw_{mchl,1}} \right\rfloor$$

- Bandwidth for I/O channel depends on application:
  - Complex applications: little I/O
  - Simple applications: more I/O
  - Formal definition of “complexity” in paper
- Performance equation complete.

Chip Area

- Summation over all chip components:

$$area_{NP} = s(io) + \sum_{j=1}^{m}(s(mchl) + \sum_{k=1}^{n}(s(p_{j,k}, t) + s(c_{i_{j,k}}) + s(c_{d_{j,k}})))$$

- Processor size depends on number of thread contexts:

$$s(p,t) = s(p_{basis}) + t \cdot s(p_{thread})$$

- Memory channel size depends on channel width:

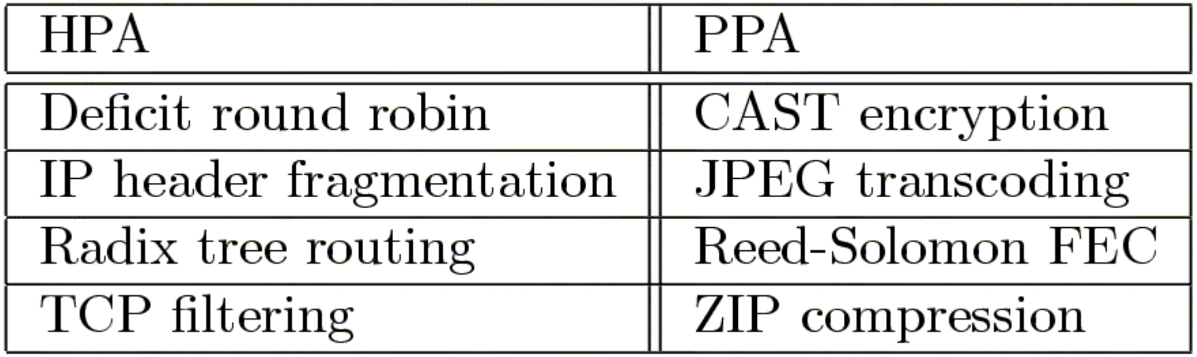
$$s(mchl) = s(mchl_{basis}) + width_{mchl} \cdot s(mchl_{pin})$$

Model Summary

- With IPS performance of system and chip area:
  - Compute IPS/area
- Necessary parameters:
  - Application parameters (load/store freq., cache miss rates, …)
  - Technology parameters (processor clock, component sizes, …)
- => Benchmark for application parameters

CommBench

- Network processor benchmark
- Benchmark applications:
  - Header-processing applications (HPA)
  - Payload-processing applications (PPA)



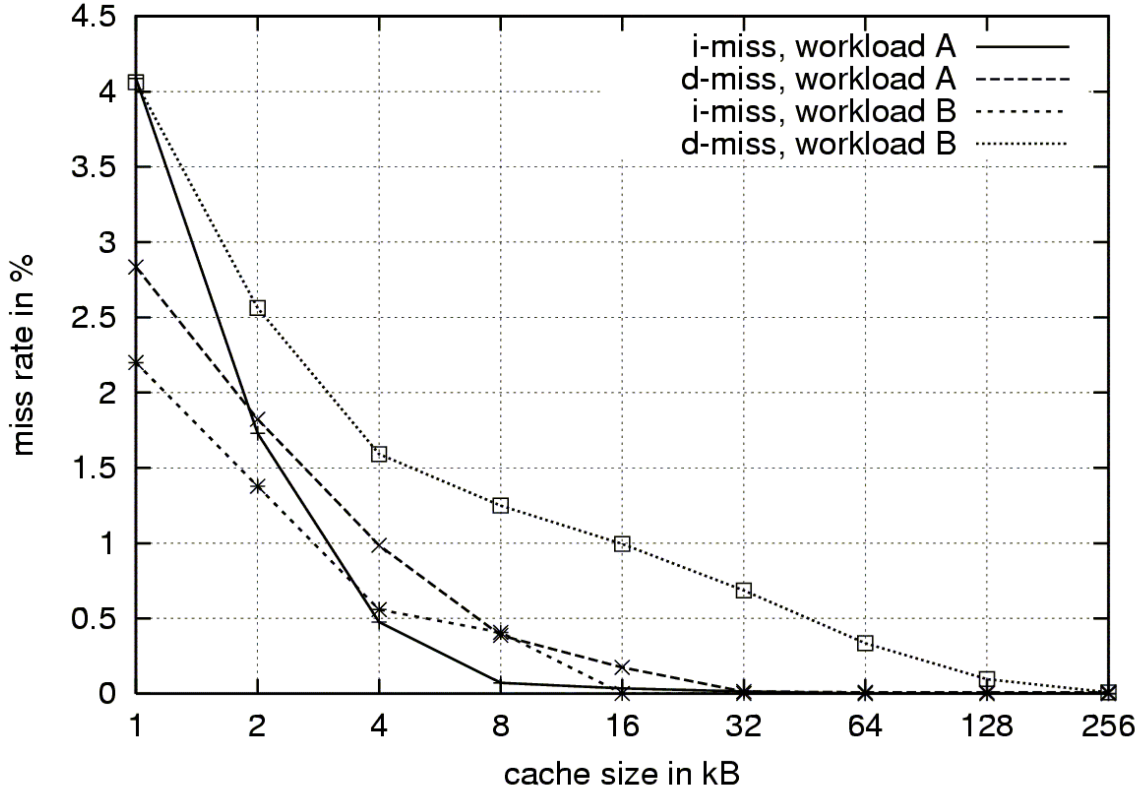
HPA	PPA
Deficit round robin	CAST encryption
IP header fragmentation	JPEG transcoding
Radix tree routing	Reed-Solomon FEC
TCP filtering	ZIP compression

- Two workloads:
  - A: HPA
  - B: PPA
- More details in [Wolf, Franklin 2000].

Application Parameters

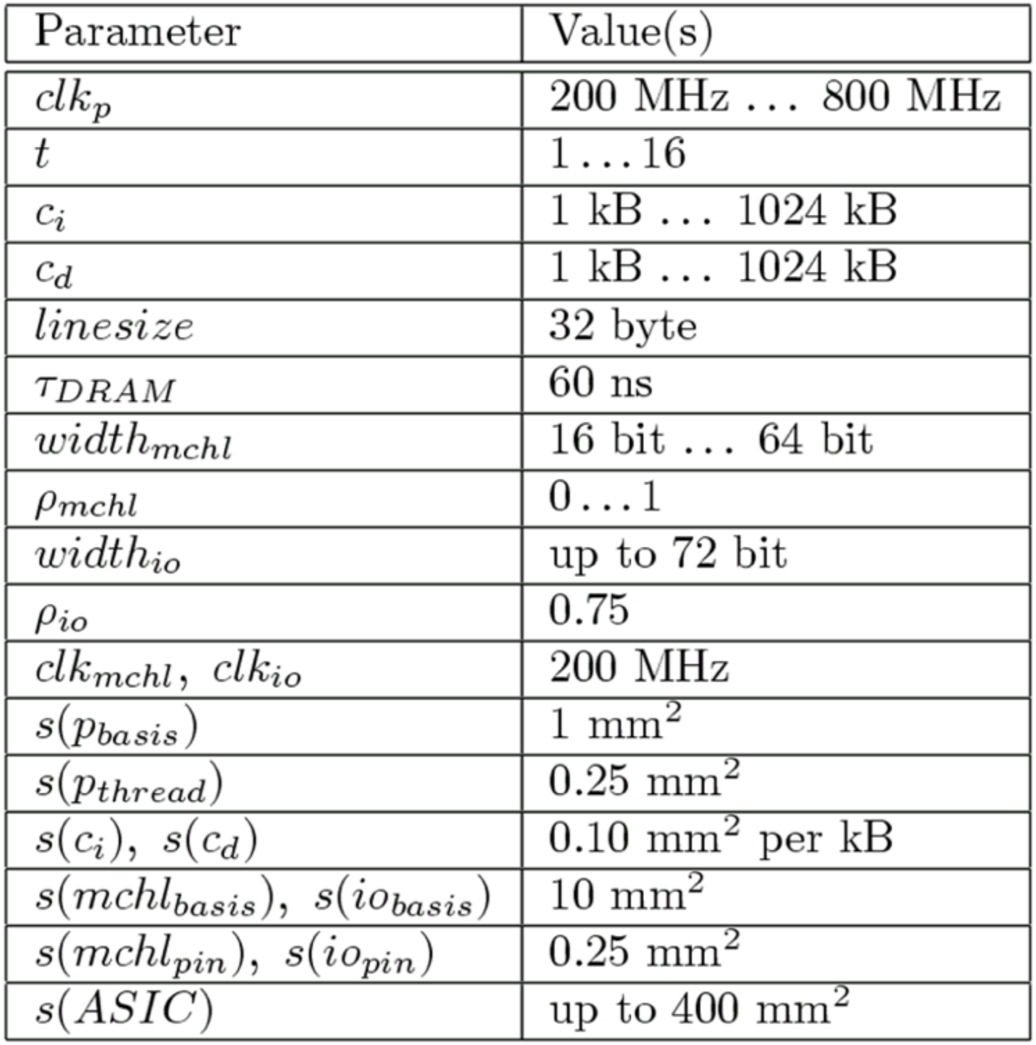
- Workload characteristics for model evaluation
- Simple parameters
  - Can easily be measured
  - Easily adaptable to other workloads

Workload W	$compl_W$	$f_{load,W}$	$f_{store,W}$
A - HPA	9.1	0.2319	0.0650
B - PPA	249	0.1691	0.0595



Technology Parameters

- 0.18 µm CMOS technology
- Exact values are hard to get from industrial sources
  - Performance model also works with more accurate parameters
- Varied parameters:
  - Processor clock
  - # of threads
  - Cache sizes
  - Memory channel bandwidth and load



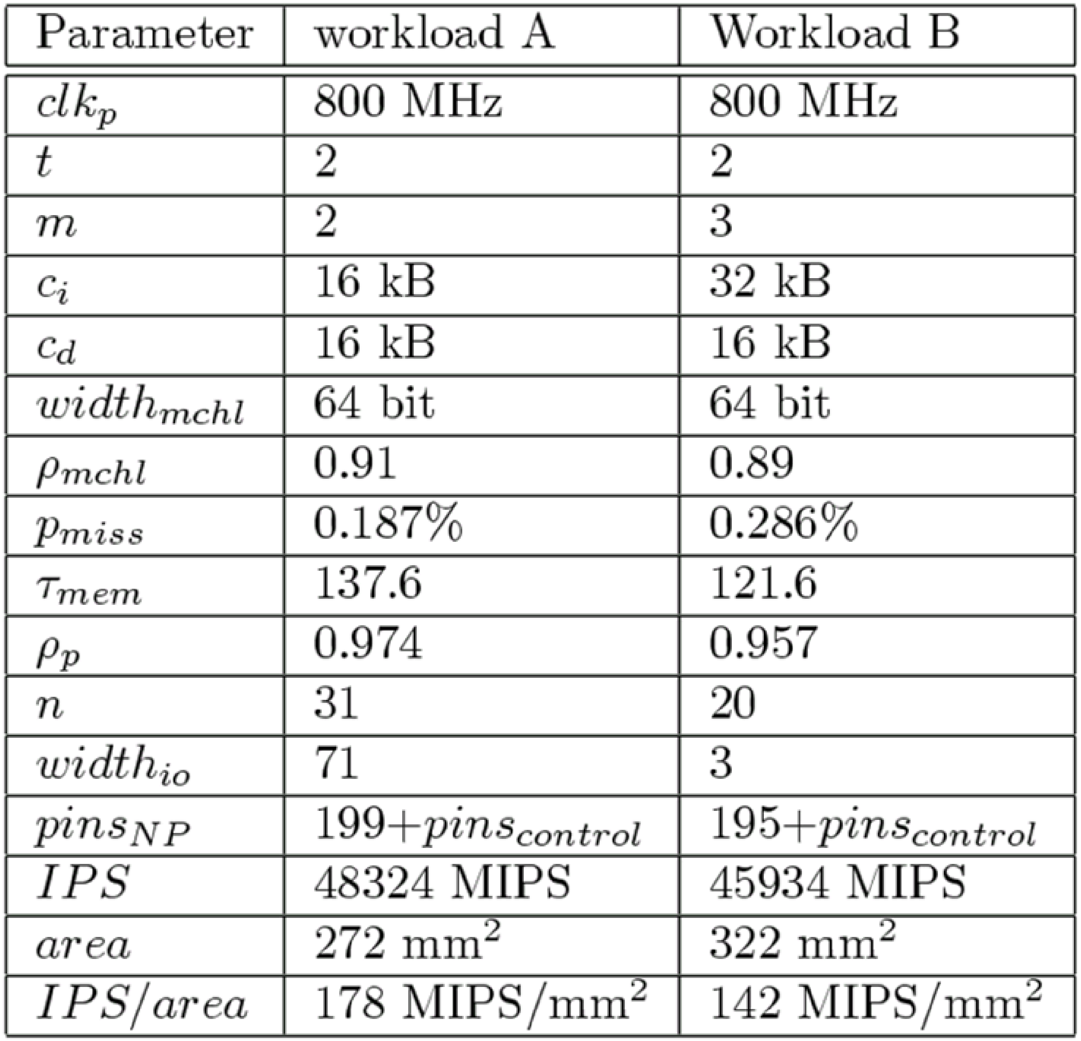
Parameter	Value(s)
clk_p	200 MHz … 800 MHz
t	$1 \dots 16$
c_i	1 kB … 1024 kB
c_d	1 kB … 1024 kB
$linesize$	32 byte
τ_{DRAM}	60 ns
$width_{mchl}$	16 bit … 64 bit
ρ_{mchl}	$0 \dots 1$
$width_{io}$	up to 72 bit
ρ_{io}	0.75
$clk_{mchl},\ clk_{io}$	200 MHz
$s(p_{basis})$	1 mm^2
$s(p_{thread})$	0.25 mm^2
$s(c_i),\ s(c_d)$	0.10 mm^2 per kB
$s(mchl_{basis}),\ s(io_{basis})$	10 mm^2
$s(mchl_{pin}),\ s(io_{pin})$	0.25 mm^2
$s(ASIC)$	up to 400 mm^2

Results

- Optimal configurations
- Performance trends (take optimal configuration and vary parameter)
  - Memory channel
  - Processor clock and threads
  - Caches
- Note: performance metric is MIPS/mm^2

Optimal Configuration

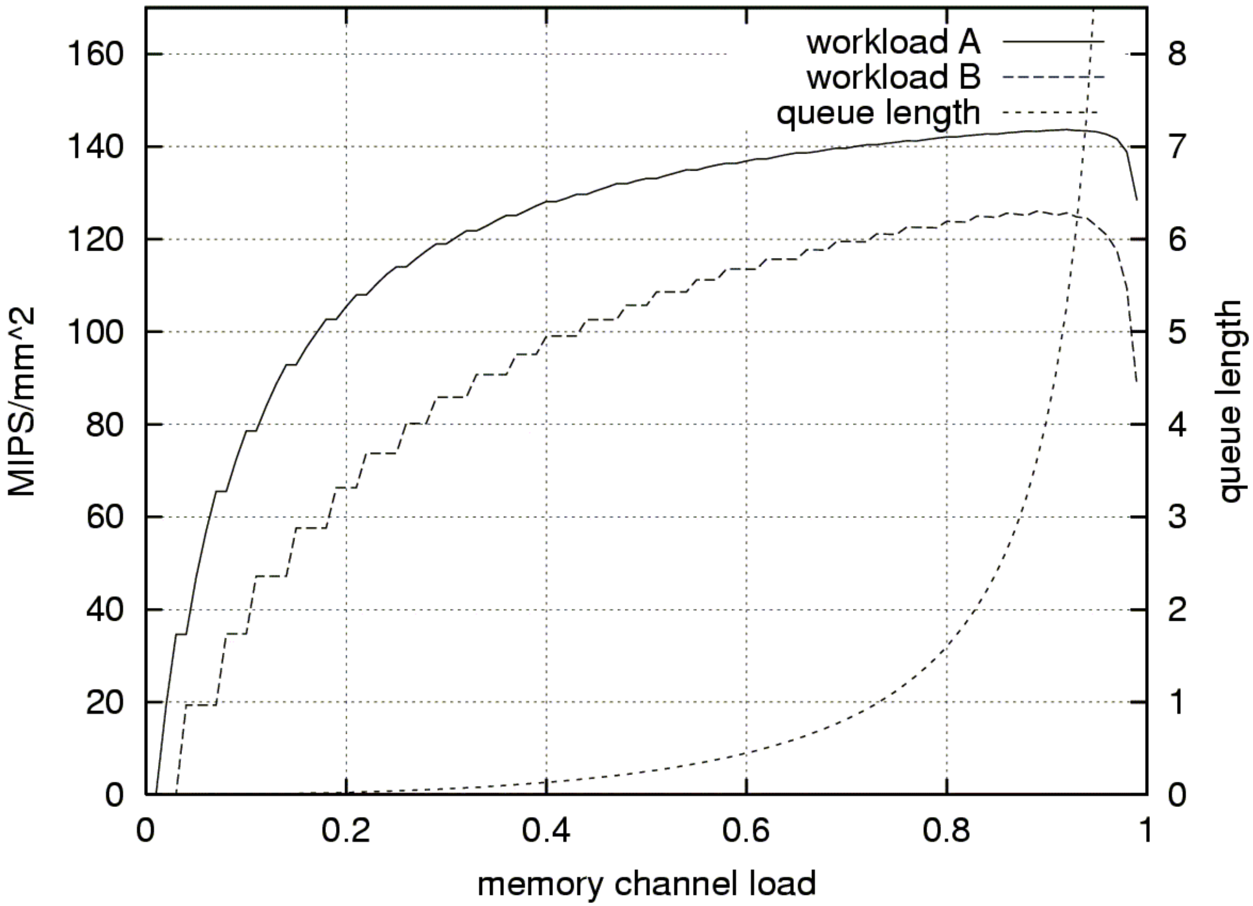
- Processor:
  - 800 MHz
  - 2 threads
  - ~96% utilization
- Chip configuration:
  - 2-3 clusters with 20-30 processors
  - 16-32 kB caches (instruction and data)
- Memory channel:
  - ~90% load
  - 64 bit width
- Off-chip memory:
  - 120-140 cycles access time
- Area:
  - 140-180 mm²



Parameter	workload A	Workload B
clk_p	800 MHz	800 MHz
t	2	2
m	2	3
c_i	16 kB	32 kB
c_d	16 kB	16 kB
$width_{mchl}$	64 bit	64 bit
ρ_{mchl}	0.91	0.89
p_{miss}	0.187%	0.286%
τ_{mem}	137.6	121.6
ρ_p	0.974	0.957
n	31	20
$width_{io}$	71	3
$pins_{NP}$	199+$pins_{control}$	195+$pins_{control}$
IPS	48324 MIPS	45934 MIPS
$area$	272 mm^2	322 mm^2
$IPS/area$	178 MIPS/mm^2	142 MIPS/mm^2

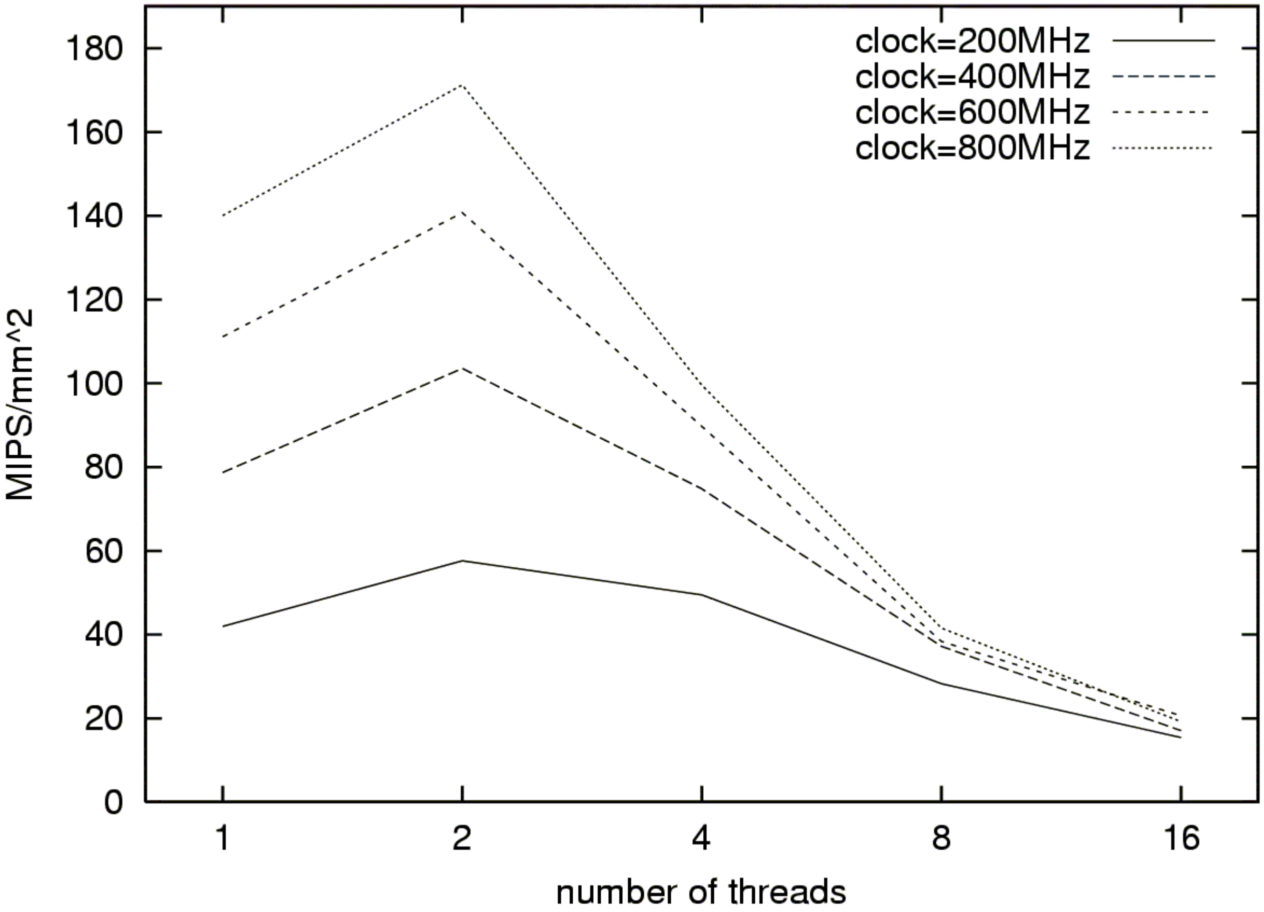
Memory Channel

- Best load ~90%
- Low load:
  - Waste of area for memory channel
- High load:
  - Very long queue length in M/D/1 model
  - High memory access time



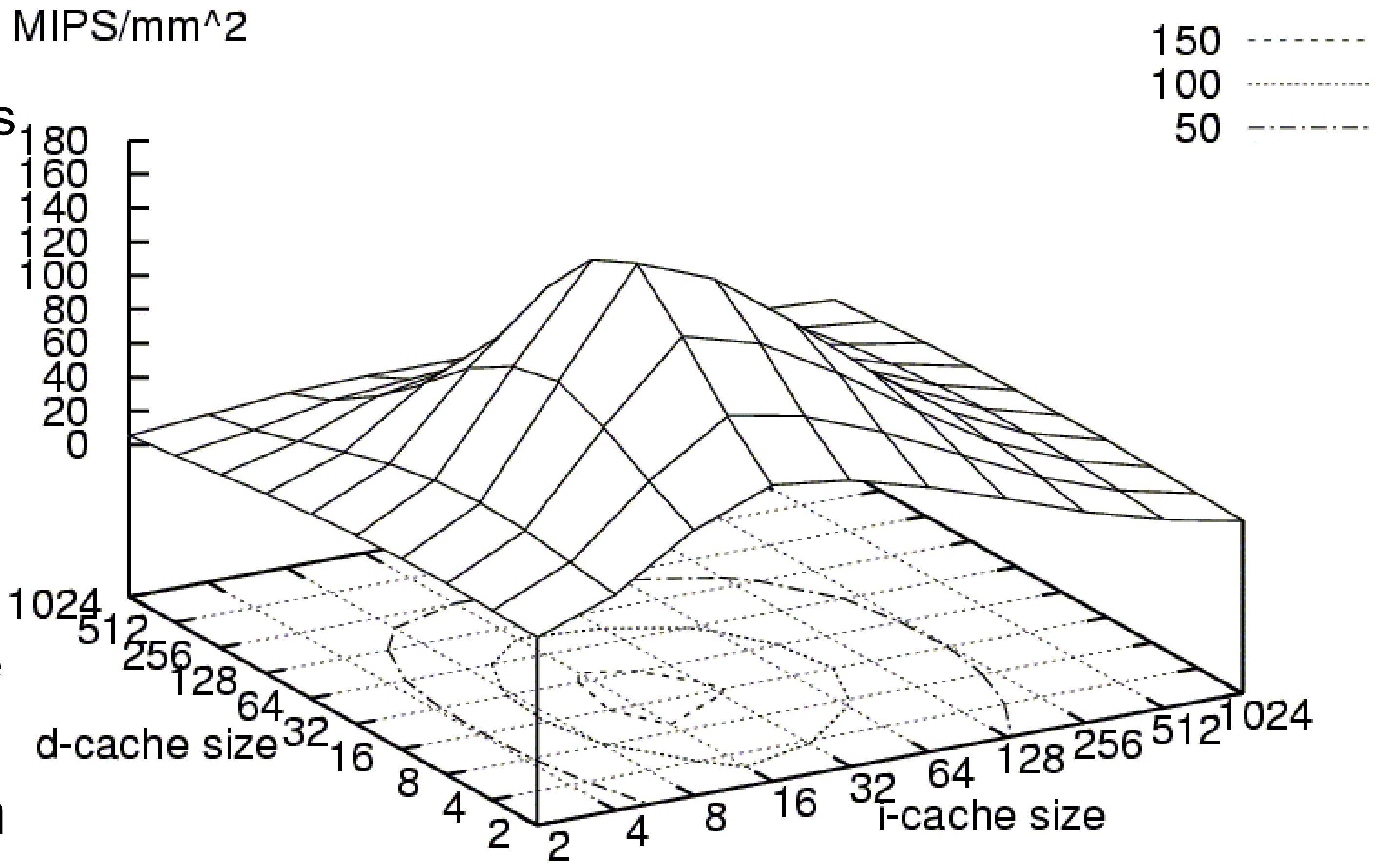
Processor Clock

- Practically linear growth with processor clock speed
- Less significant with more threads
  - Less cache per thread
  - Performance limited by off-chip memory access time



Cache Configuration

- Cache size
  - Small caches cause inefficient execution
  - Large caches waste space
- Performance very sensitive to deviations from optimum



Summary

- NP performance model
  - Determines processing performance of NP configuration
  - Relates processing power to area of system-on-a-chip
  - Uses simple workload characteristics and technology parameters
- Optimal configuration for given scenario
- Performance trends as "rules of thumb:"
  - Cache configuration has big impact on performance
  - Two to four thread contexts is optimal
  - Higher processor clock rates and memory channel directly translate into higher performance
- Model can aid in first-order NP design

Next Class

- Do you want help session on homework?
- If yes:
  - Next Tuesday: help session
  - Thursday: Introduction to Intel IXA, read chapter 18
- If no:
  - Tuesday: Introduction to Intel IXA, read chapter 18