

COM515 Advanced Computer Architecture

Lecture 6. Multithreading & Multicore Processors

Prof. Taeweon Suh

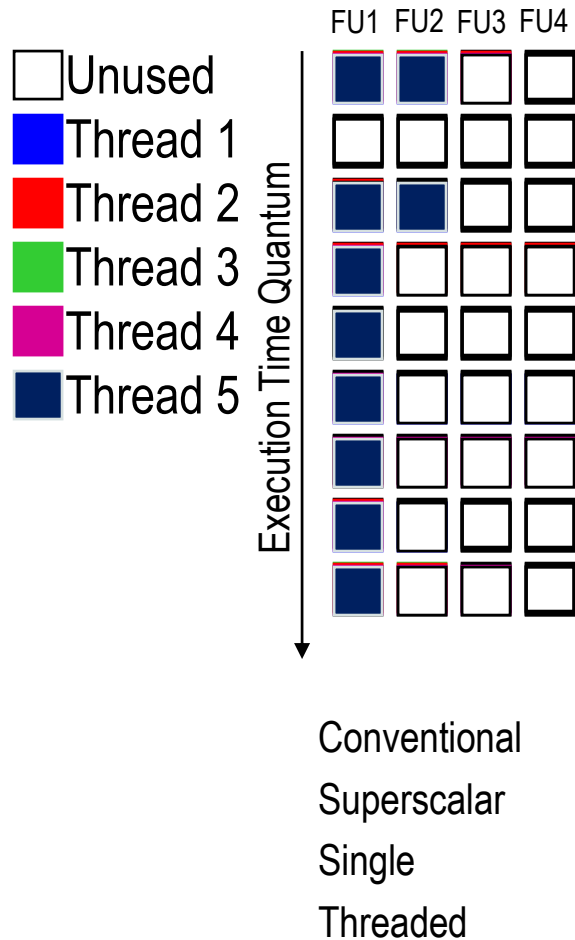
Computer Science Education

Korea University

TLP

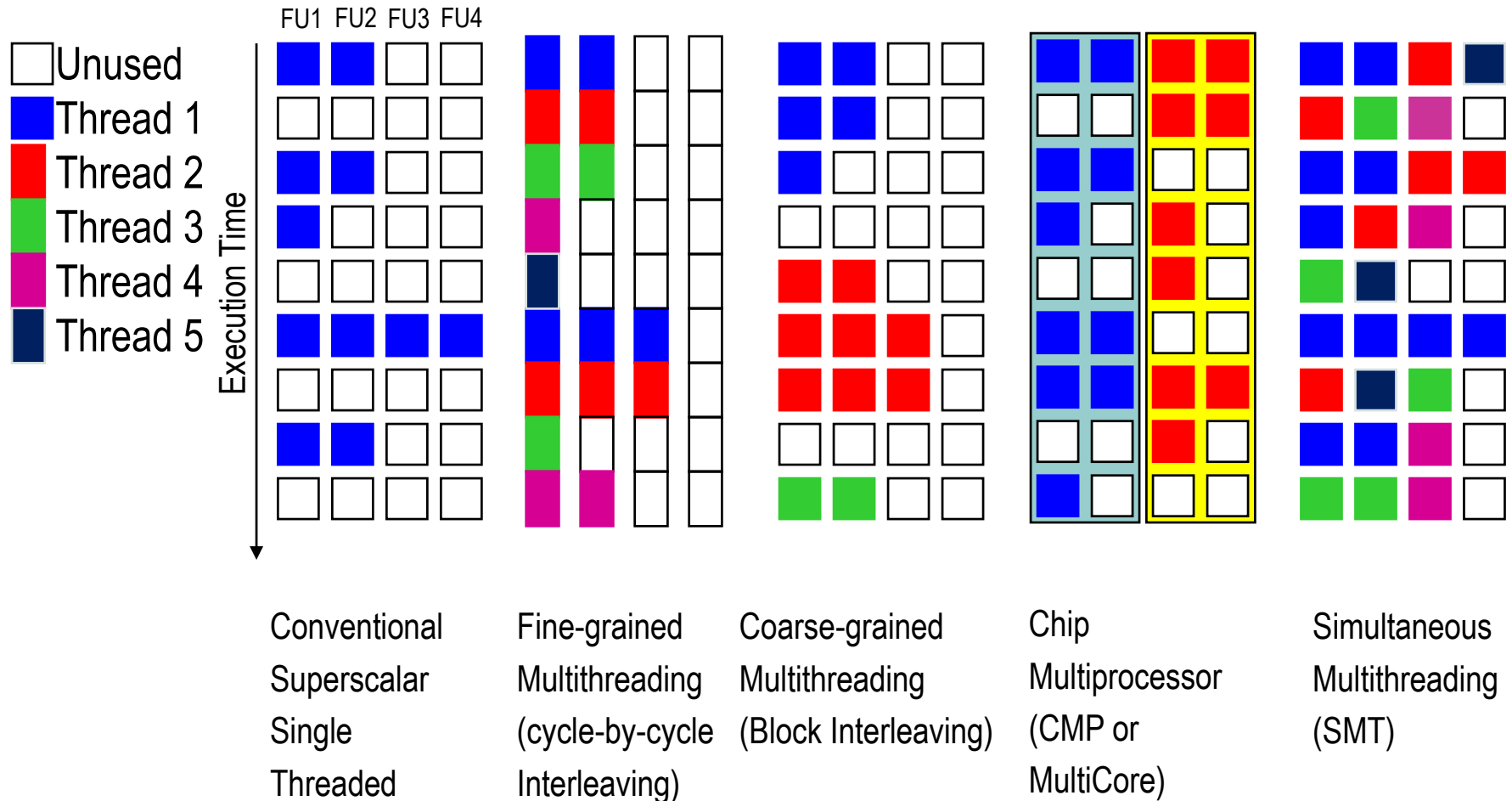
- ILP of a single program is hard
 - Large ILP is Far-flung
 - We are human after all, program w/ sequential mind
- Reality: running multiple threads or programs
- Thread Level Parallelism
 - Time Multiplexing
 - Throughput computing
 - Multiple program workloads
 - Multiple concurrent threads
 - Helper threads to improve single program performance

Multi-Tasking Paradigm



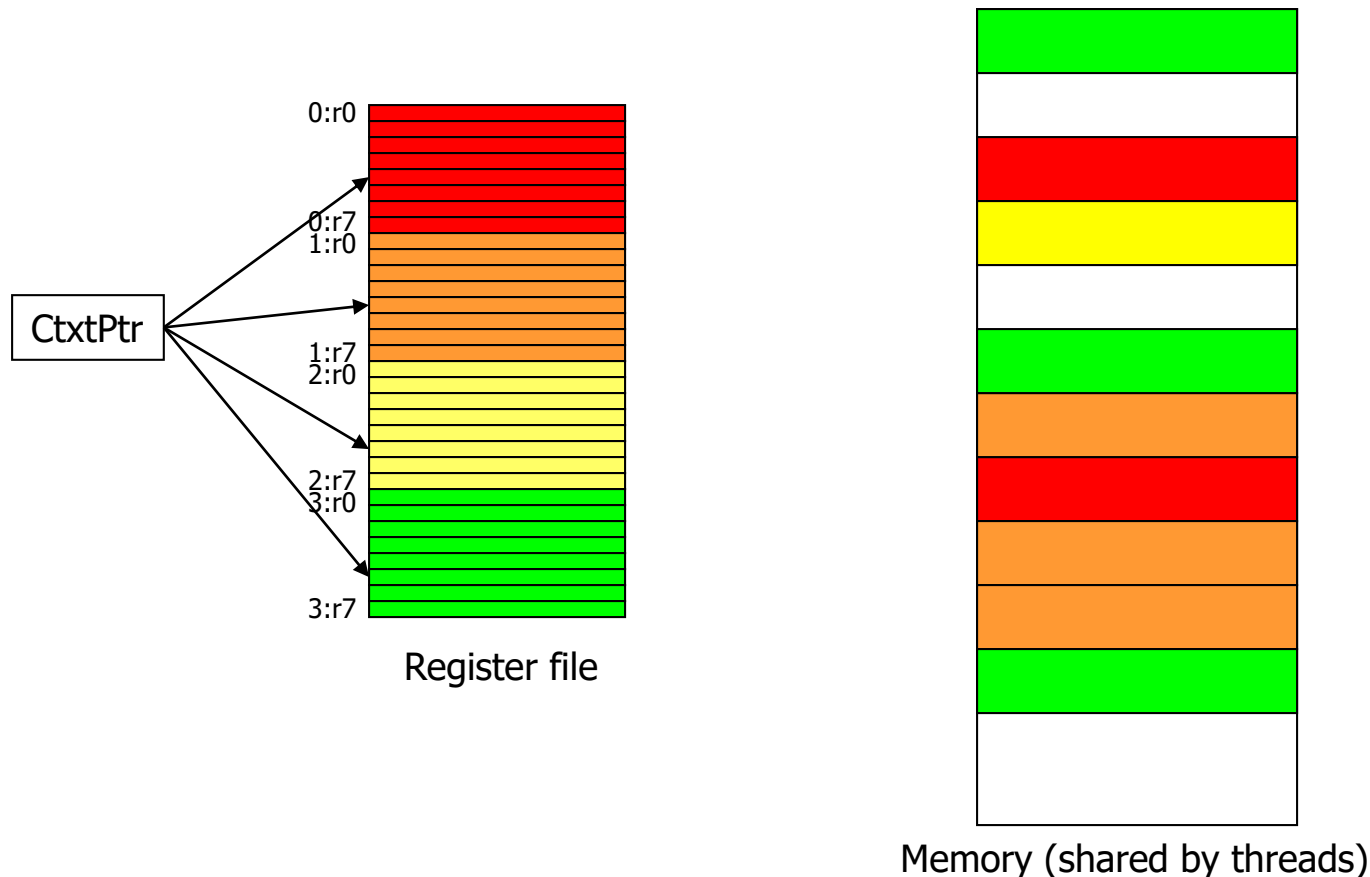
- Virtual memory makes it easy
- Context switch could be expensive or requires extra HW
 - VIVT cache
 - VIPT cache
 - TLBs

Multi-threading Paradigm



Conventional Multithreading

- Zero-overhead context switch
- Duplicated contexts for threads



Cycle Interleaving MT

- Per-cycle, Per-thread instruction fetching
- Examples:
 - HEP (Heterogeneous Element Processor) (1982)
 - http://en.wikipedia.org/wiki/Heterogeneous_Element_Processor
 - Horizon (1988)
 - Tera MTA (Multi-Threaded Architecture) (1990)
 - MIT M-machine (1998)
- Interesting questions to consider
 - Does it need a sophisticated branch predictor?
 - Or does it need any speculative execution at all?
 - Get rid of “**branch prediction**”?
 - Get rid of “**predication**”?
 - Does it need any out-of-order execution capability?

Tera Multi-Threaded Architecture (MTA)

- Cycle-by-cycle interleaving
- MTA can context-switch every cycle (3ns)
- Each processor in a Tera computer can execute multiple instruction streams simultaneously
 - As many as 128 distinct threads (hiding 384ns)
 - On every clock tick, the processor logic selects a stream that is ready to execute
- 3-wide VLIW instruction format (M+ALU+ALU/Br)
- Each instruction has 3-bit for dependence lookahead
 - Determine if there is dependency with subsequent instructions
 - Execute up to 7 future VLIW instructions (before switch)

Loop:

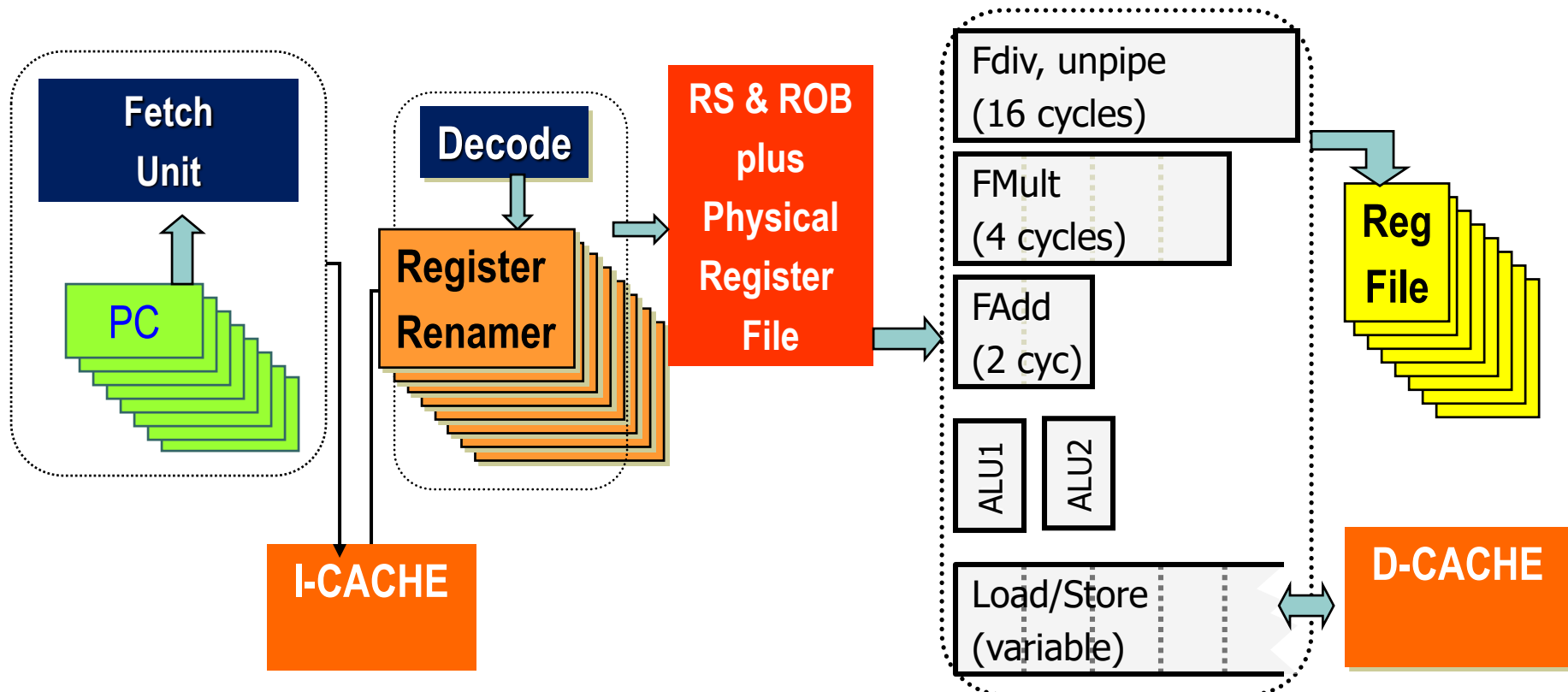
nop	r1=r2+r3	r5=r6+4	lookahead=1
nop	r8=r9-r10	r11=r12-r13	lookahead=2
[r5]=r1	r4=r4-1	bnz Loop	lookahead=0

Block Interleaving MT

- Context switch on a specific event (dynamic pipelining)
 - Explicit switching: implementing a switch instruction
 - Implicit switching: trigger when a specific instruction class fetched
- Static switching (switch upon fetching)
 - Switch-on-memory-instructions: Rhamma processor (1996)
 - Switch-on-branch or switch-on-hard-to-predict-branch
 - Trigger can be implicit or explicit instruction
- Dynamic switching
 - Switch-on-cache-miss (switch in later pipeline stage): MIT Sparcle (MIT Alewife's node) (1993), Rhamma Processor (1996)
 - Switch-on-use (lazy strategy of switch-on-cache-miss)
 - Valid bit needed for each register
 - Clear when load issued, set when data returned
 - Switch-on-signal (e.g. interrupt)
 - Predicated switch instruction based on conditions
- No need to support a large number of threads

Simultaneous Multithreading (SMT)

- SMT name first used by UW; Earlier versions from UCSB [Nemirovsky, HICSS'91] and Matsudshita [Hirata et al., [ISCA-92]
- Intel's HyperThreading (2-way SMT)
- IBM Power7 (4/6/8 cores, 4-way SMT); IBM Power5/6 (2 cores. Each 2-way SMT, 4 chips per package) : Power5 has OoO cores, Power6 In-order cores;
- Basic ideas: Conventional MT + Simultaneous issue + Sharing common resources



Instruction Fetching Policy

- FIFO, Round Robin, simple but may be too naive
- Adaptive Fetching Policies
 - BRCOUNT (reduce wrong path issuing)
 - Count # of br inst in decode/rename/IQ stages
 - Give top priority to thread with the least BRCOUNT
 - MISSCOUNT (reduce IQ clog)
 - Count # of outstanding D-cache misses
 - Give top priority to thread with the least MISSCOUNT
 - ICOUNT (reduce IQ clog)
 - Count # of inst in decode/rename/IQ stages
 - Give top priority to thread with the least ICOUNT
 - IQPOSN (reduce IQ clog)
 - Give lowest priority to those threads with inst closest to the head of INT or FP instruction queues
 - Due to that threads with the oldest instructions will be most prone to IQ clog
 - No Counter needed

Resource Sharing

- Could be tricky when threads compete for the resources
- Static
 - Less complexity
 - Could penalize threads (e.g. instruction window size)
 - P4's Hyperthreading
- Dynamic
 - Complex
 - What is fair? How to quantify fairness?
- A growing concern in Multi-core processors
 - Shared L2, Bus bandwidth, etc.
 - Issues
 - Fairness
 - Mutual thrashing

P4 HyperThreading Resource Partitioning

- TC (or UROM) is alternatively accessed per cycle for each logical processor unless one is stalled due to TC miss
- μop queue (into $\frac{1}{2}$) after fetched from TC
- ROB (126/2)
- LB (48/2)
- SB (24/2) (32/2 for Prescott)
- General μop queue and memory μop queue (1/2)
- TLB ($\frac{1}{2}$?) as there is no PID
- Retirement: alternating between 2 logical processors

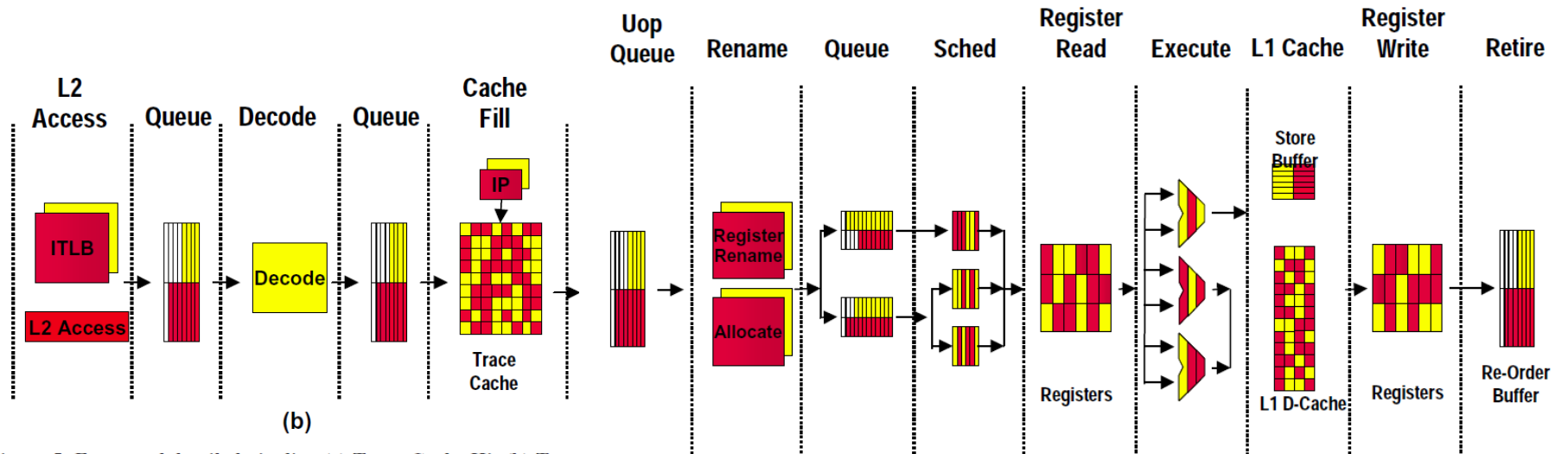


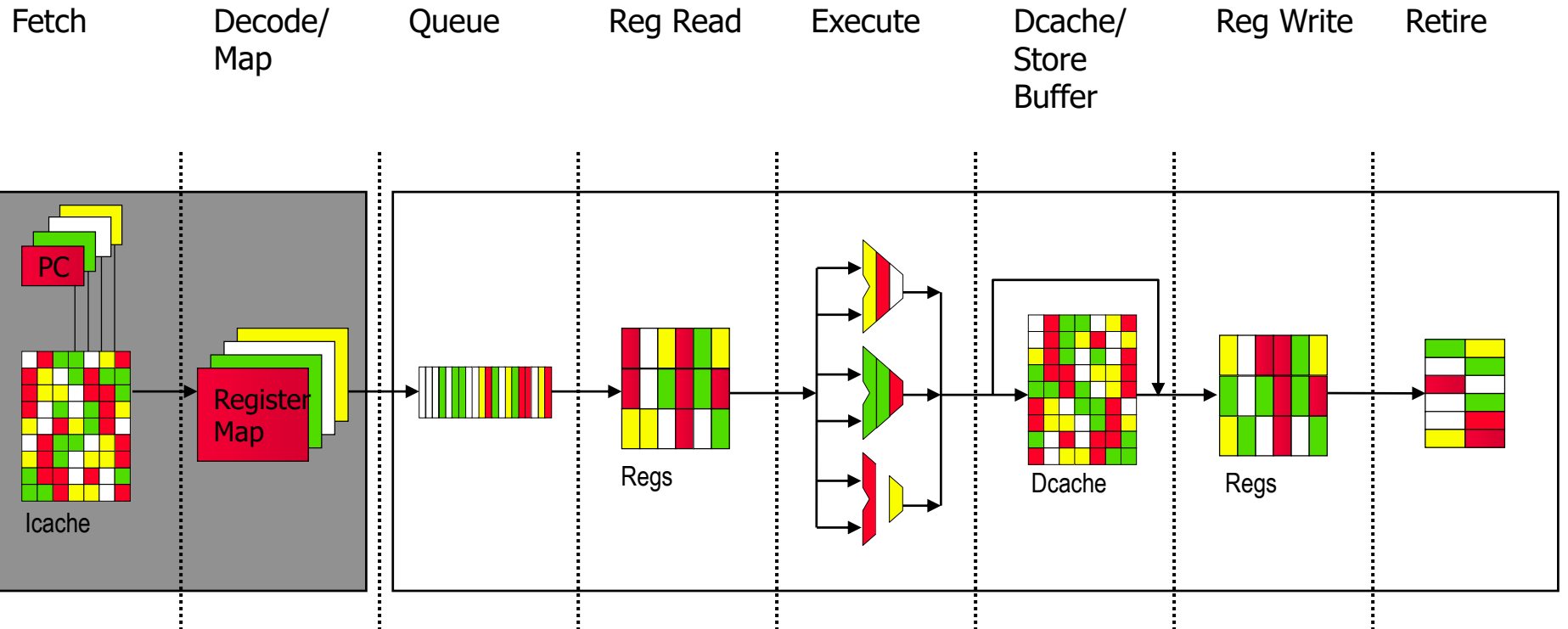
Figure 5: Front-end detailed pipeline (a) Trace Cache Hit (b) Trace Cache Miss

Figure 6: Out-of-order execution engine detailed pipeline

Alpha 21464 (EV8) Processor

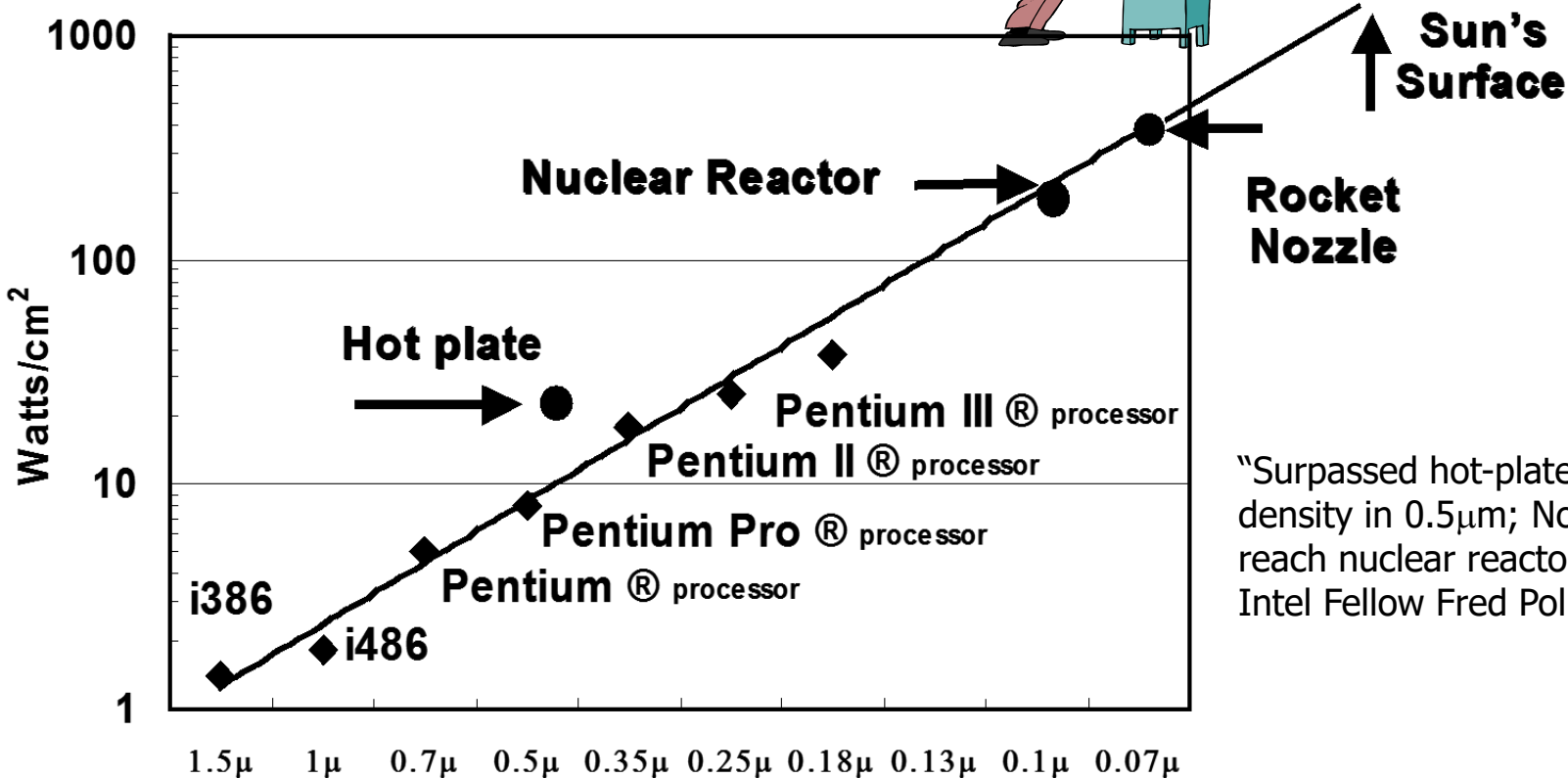
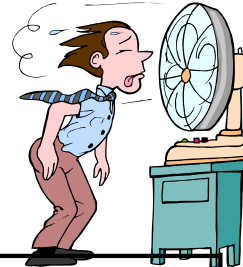
- Enhanced out-of-order execution (that giant 2Bc-gskew predictor we discussed (?) before is here)
- Large on-chip L2 cache
- Direct RAMBUS interface
- On-chip router for system interconnect
- Glueless, directory-based, ccNUMA for up to 512-way SMP
- 8-wide superscalar
- 4-way **simultaneous multithreading (SMT)**
 - Total die overhead $\sim 6\%$ (allegedly)
- Slated for a 2004 release, but canceled on June 2001

SMT Pipeline



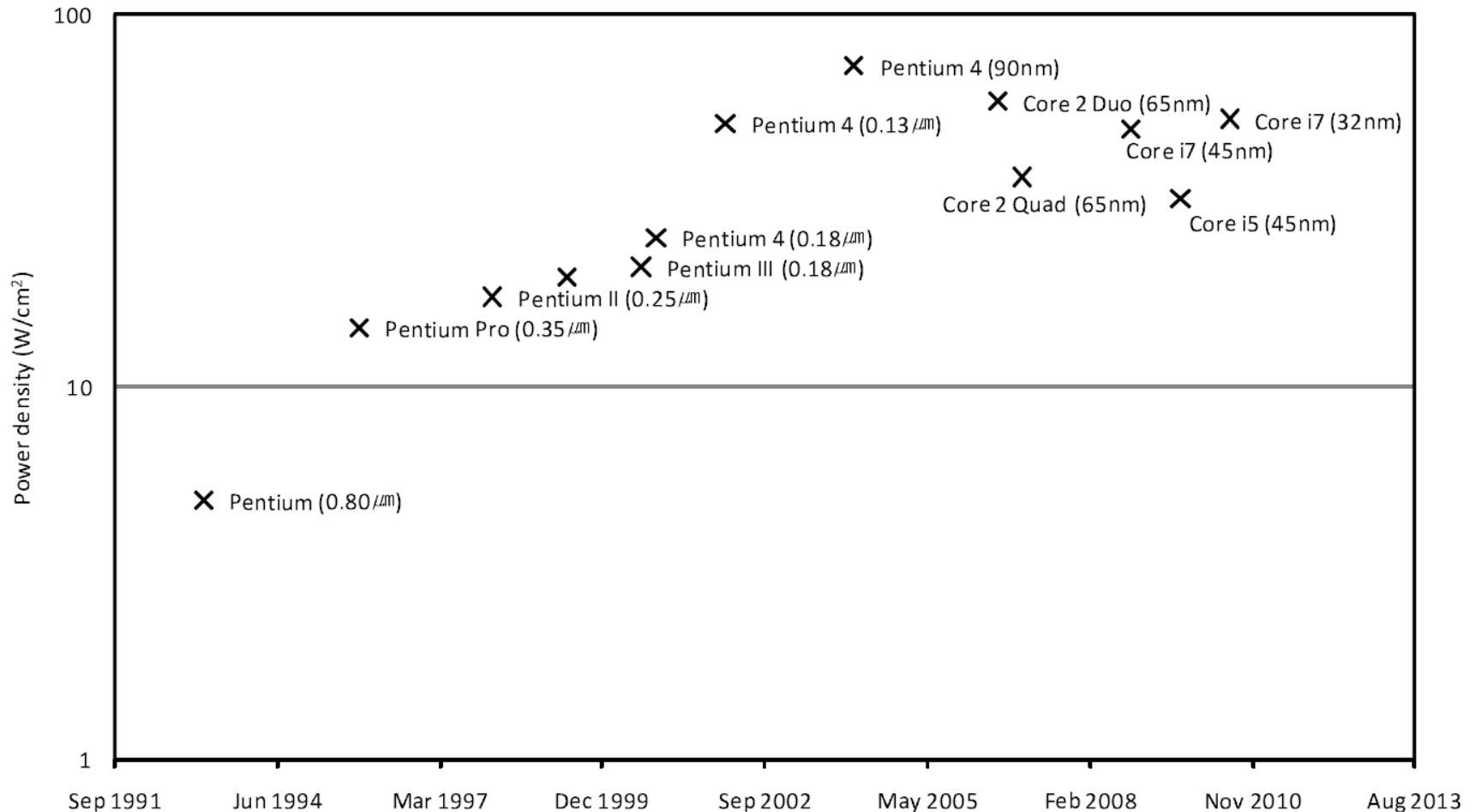
Reality Check, circa 200x

- Conventional processor designs run out of steam
 - Power wall (**thermal**)
 - Complexity (verification)
 - Physics (CMOS scaling)



"Surpassed hot-plate power density in 0.5µm; Not too long to reach nuclear reactor," Former Intel Fellow Fred Pollack.

Latest Power Density Trend

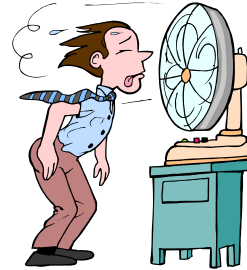


Yeo and Lee, "Peeling the Power Onion of Data Centers," In
Energy Efficient Thermal Management of Data Centers, Springer. To appear 2011

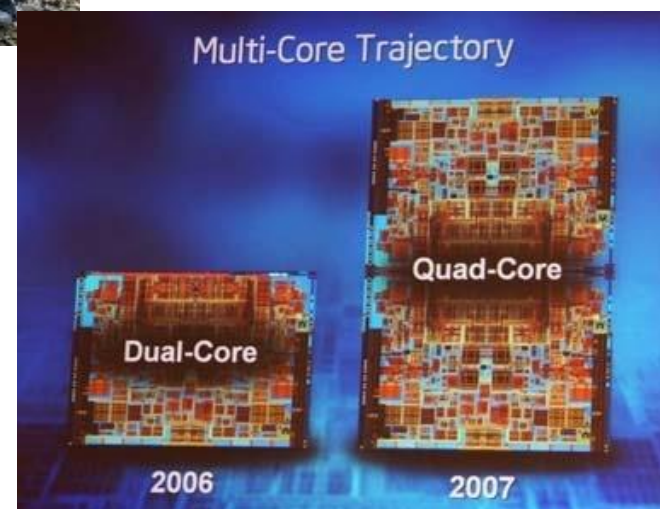
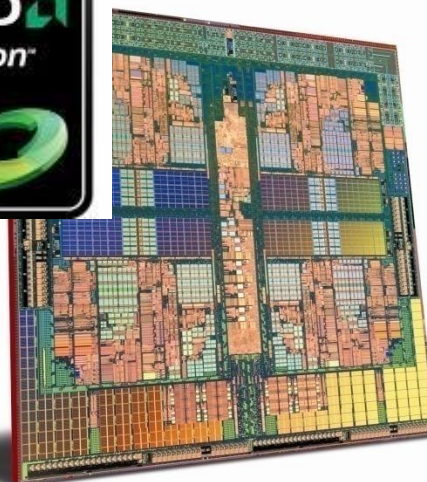
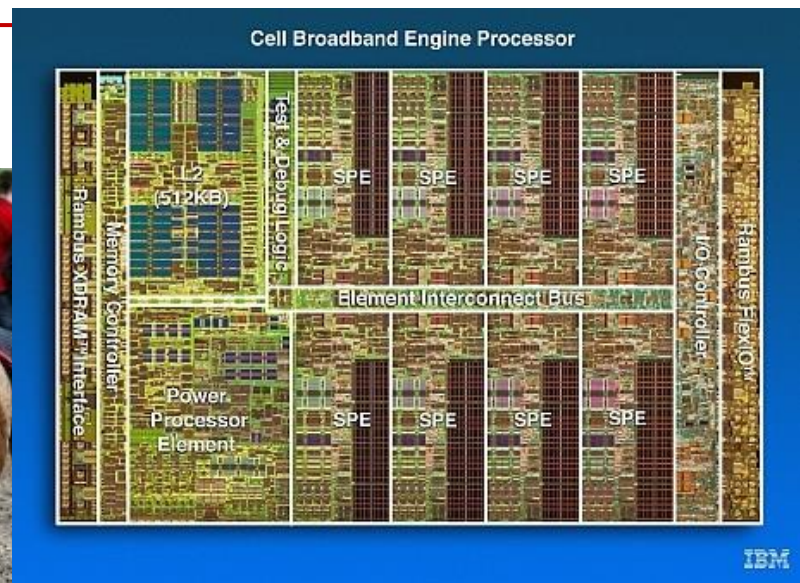
Reality Check, circa 200x

P53 GRILL

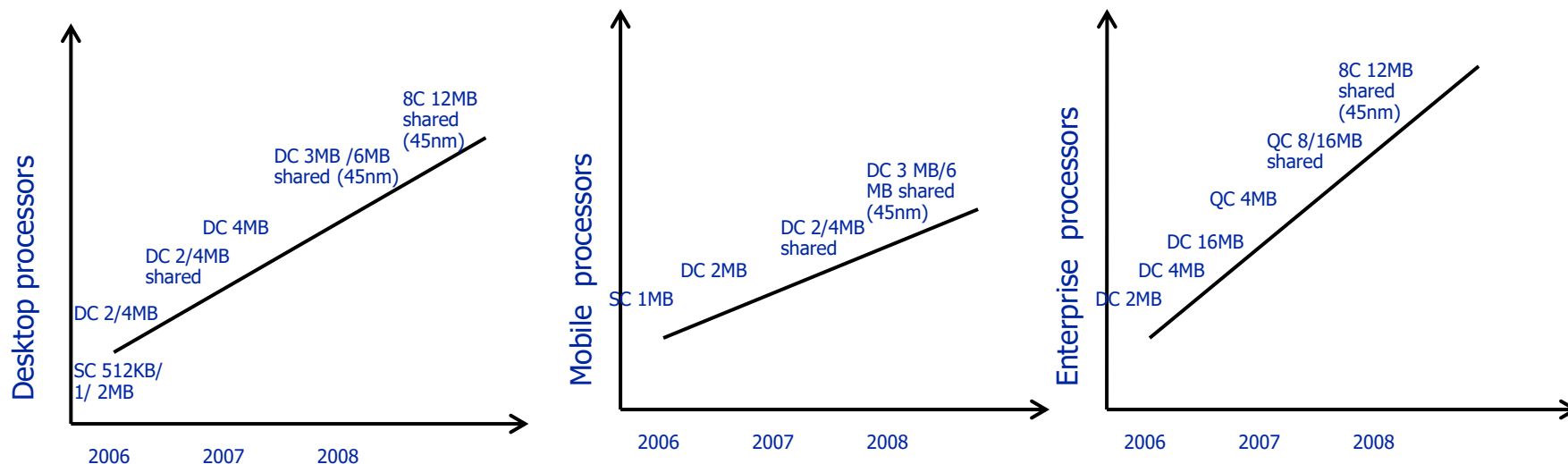
- Conventional processor designs run out of steam
 - Power wall (**thermal**)
 - Complexity (verification)
 - Physics (CMOS scaling)
- Unanimous direction → Multi-core
 - Simple cores (massive number)
 - Keep
 - Wire communication on leash
 - Gordon Moore happy (Moore's Law)
 - Architects' menace: kick the ball to the other side of the court?
- What do you (or your customers) want?
 - Performance (and/or availability)
 - Throughput > latency (turnaround time)
 - Total cost of ownership (performance per dollar)
 - Energy (performance per watt)
 - Reliability and dependability, SPAM/spy free



Multi-core Processor Gala

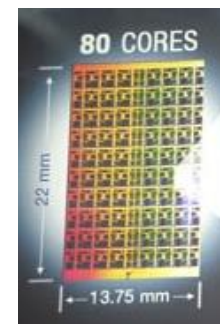


Intel's Multicore Roadmap

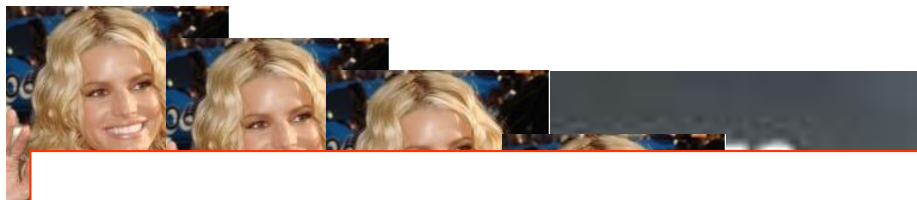


Source: Adapted from Tom's Hardware

- To extend Moore's Law
- To delay the ultimate limit of physics
- By 2010
 - all Intel processors delivered will be multicore
 - Intel's 80-core processor (FPU array)



Is a Multi-core really better off?

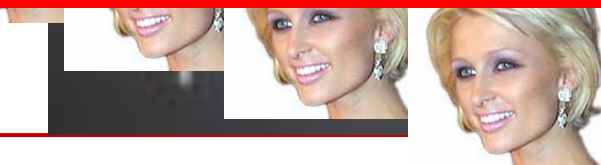


If you were plowing a field,
which would you rather use:

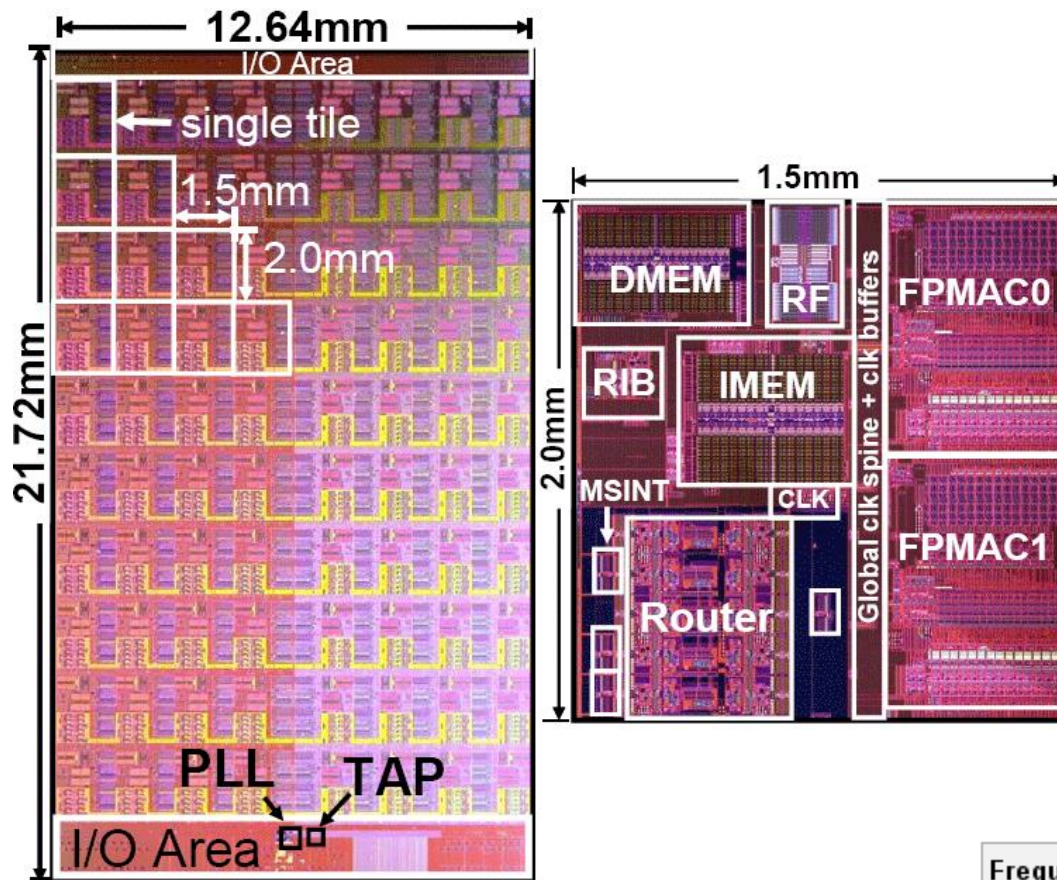
Two strong oxen or 1024 chickens?

--- Seymour Cray

Well, it is hard to say in Computing World



Intel TeraFlops Research Prototype (2007)



- 2KB Data Memory
- 3KB Instruction Memory
- No coherence support
- 2 FMACs (Floating-point Multiply Accumulators)

Frequency	Voltage	Power	Aggregate Bandwidth	Performance
3.16 GHz	0.95 V	62W	1.62 Terabits/s	1.01 Teraflops
5.1 GHz	1.2 V	175W	2.61 Terabits/s	1.63 Teraflops
5.7 GHz	1.35 V	265W	2.92 Terabits/s	1.81 Teraflops

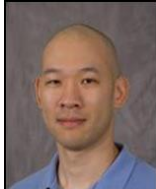
Georgia Tech 64-Core 3D-MAPS Many-Core Chip



LEE, Core/Arch



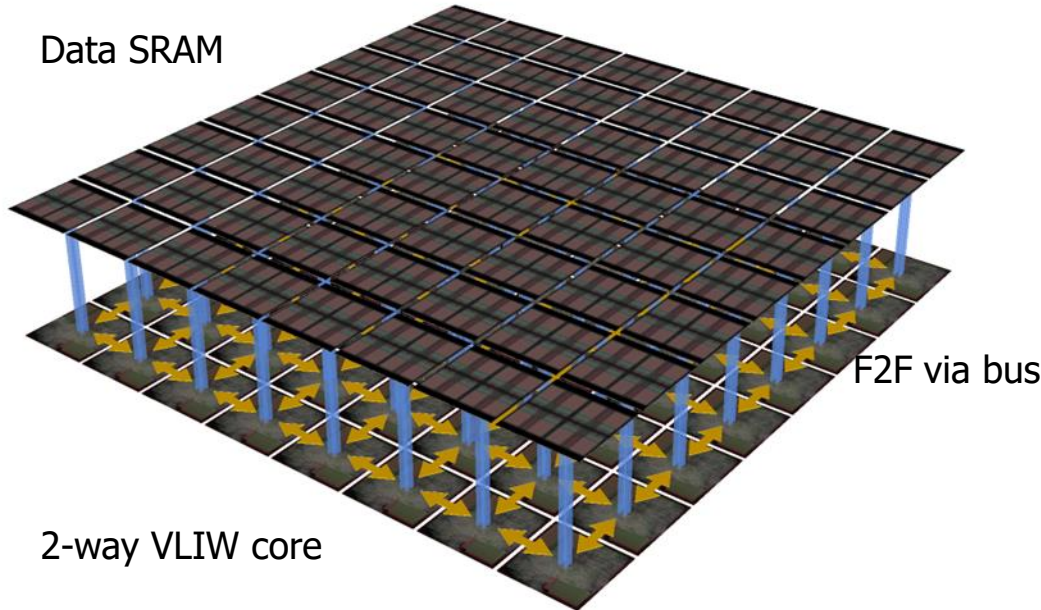
LIM, CAD Tool



LOH, Memory

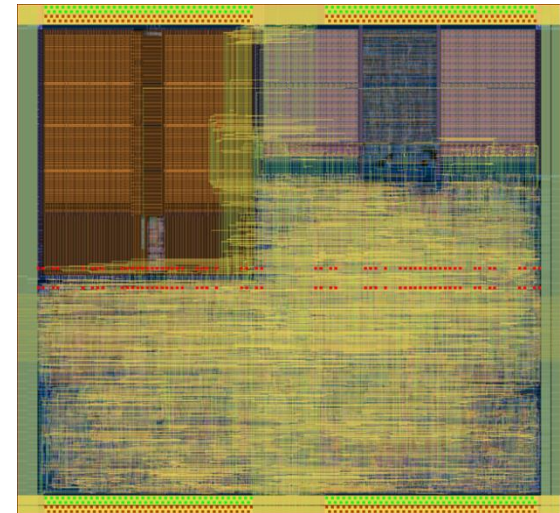
- 3D-stacked many-core processor
- Fast, high-density face-to-face vias for high bandwidth
- Wafer-to-wafer bonding
- @277MHz, peak data B/W $\sim 70.9\text{GB/sec}$

Data SRAM

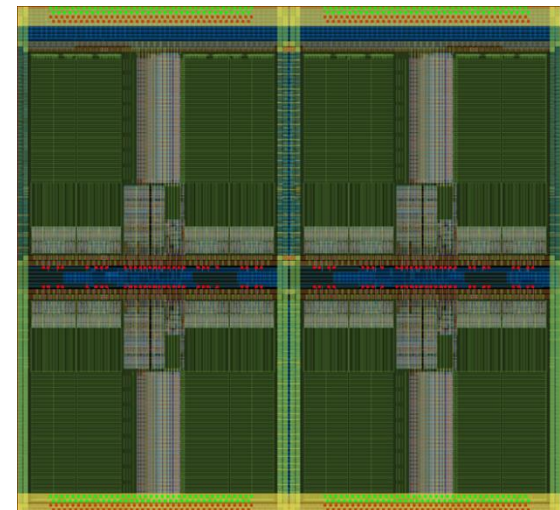


2-way VLIW core

F2F via bus

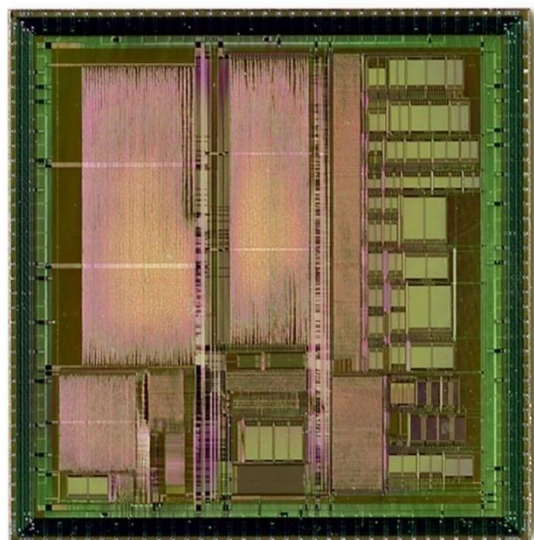


Single Core



Single SRAM tile

Is a Multi-core really better off?



DEEP BLUE



480 chess chips

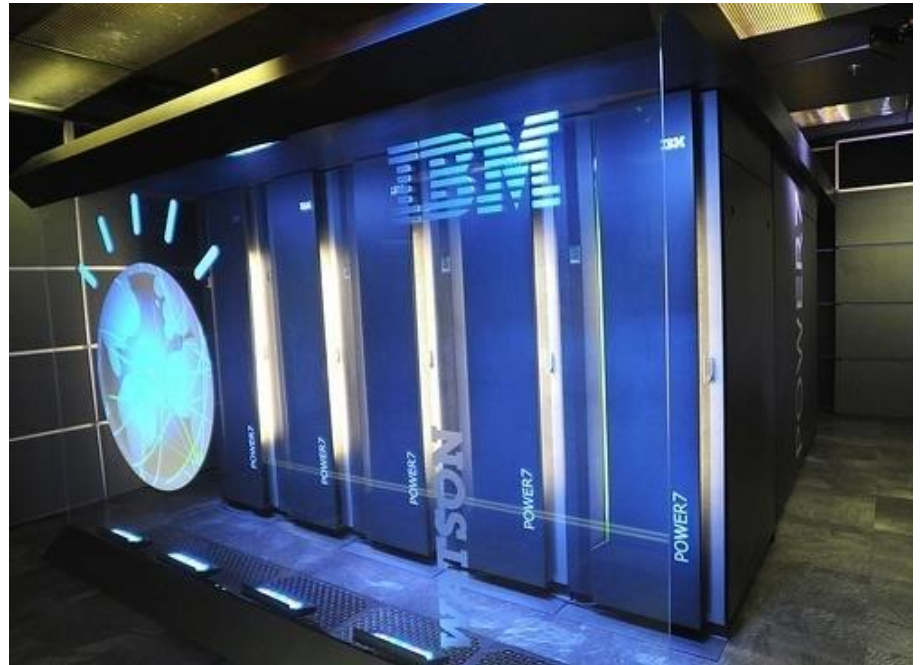
Can evaluate 200,000,000 moves per second!!

<http://www.youtube.com/watch?v=cK0YOGJ58a0>



IBM Watson Jeopardy! Competition (2011.2.)

- POWER7
- Massively parallel processing
- Combine: Processing power, Natural language processing, AI, Search, Knowledge extraction



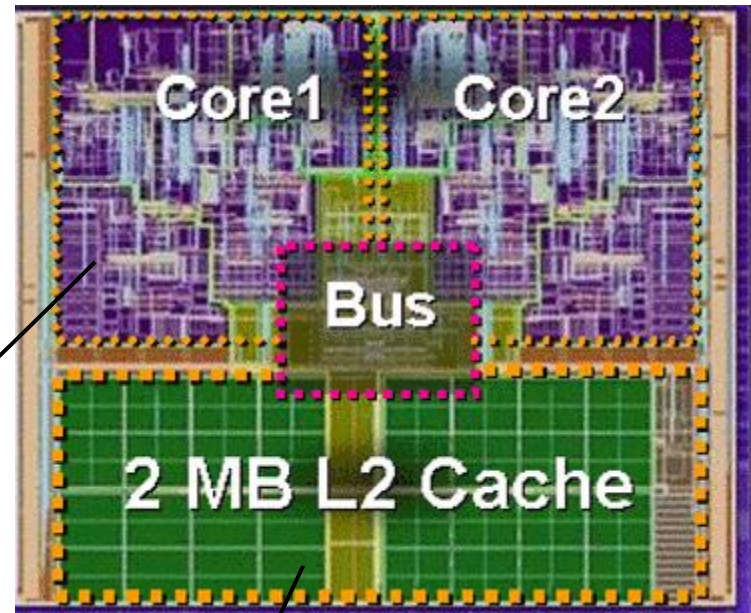
http://www.youtube.com/watch?v=WFR3lOm_xhE

Major Challenges for Multi-Core Designs

- Communication
 - Memory hierarchy
 - Data allocation (you have a large shared L2/L3 now)
 - Interconnection network
 - AMD HyperTransport
 - Intel QPI
 - Scalability
 - Bus Bandwidth, how to get there?
- Power-Performance — Win or lose?
 - Borkar's multicore arguments
 - 15% per core performance drop → 50% power saving
 - Giant, single core wastes power when task is small
 - How about leakage?
- Process variation and yield
- Programming Model

Intel Core 2 Duo

- Homogeneous cores
- Bus based on chip interconnect
- Shared on-die Cache Memory
- Traditional I/O

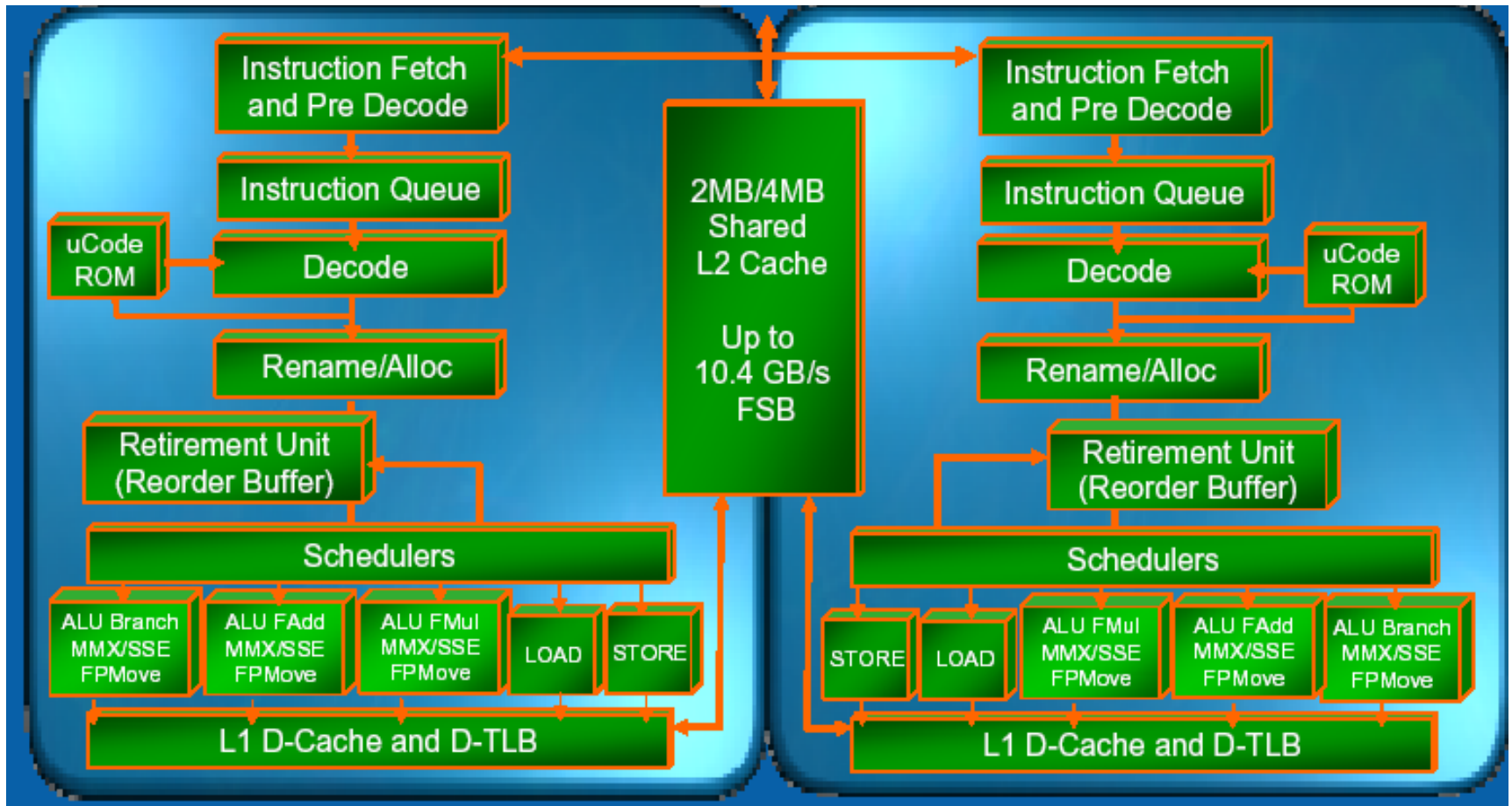


Source: Intel Corp.

*Classic OOO: Reservation Stations,
Issue ports, Schedulers...etc*

*Large, shared set associative, prefetch,
etc.*

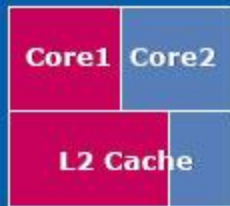
Core 2 Duo Microarchitecture



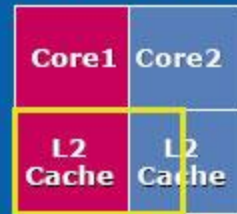
Why Sharing on-die L2?

Advanced Smart Cache Dynamic Cache Allocation

Advanced Smart Cache



Independent Cache (today)

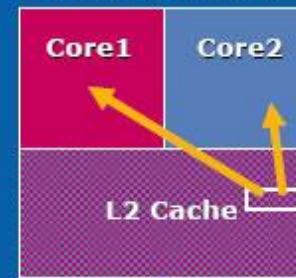


Shared Cache adapts to mismatched loads. Independent Cache can thrash heavy app even when other cache is under-utilized



Advanced Smart Cache Efficient Data Sharing

Advanced Smart Cache



FSB

Chipset MCH

Independent Cache (today)



FSB

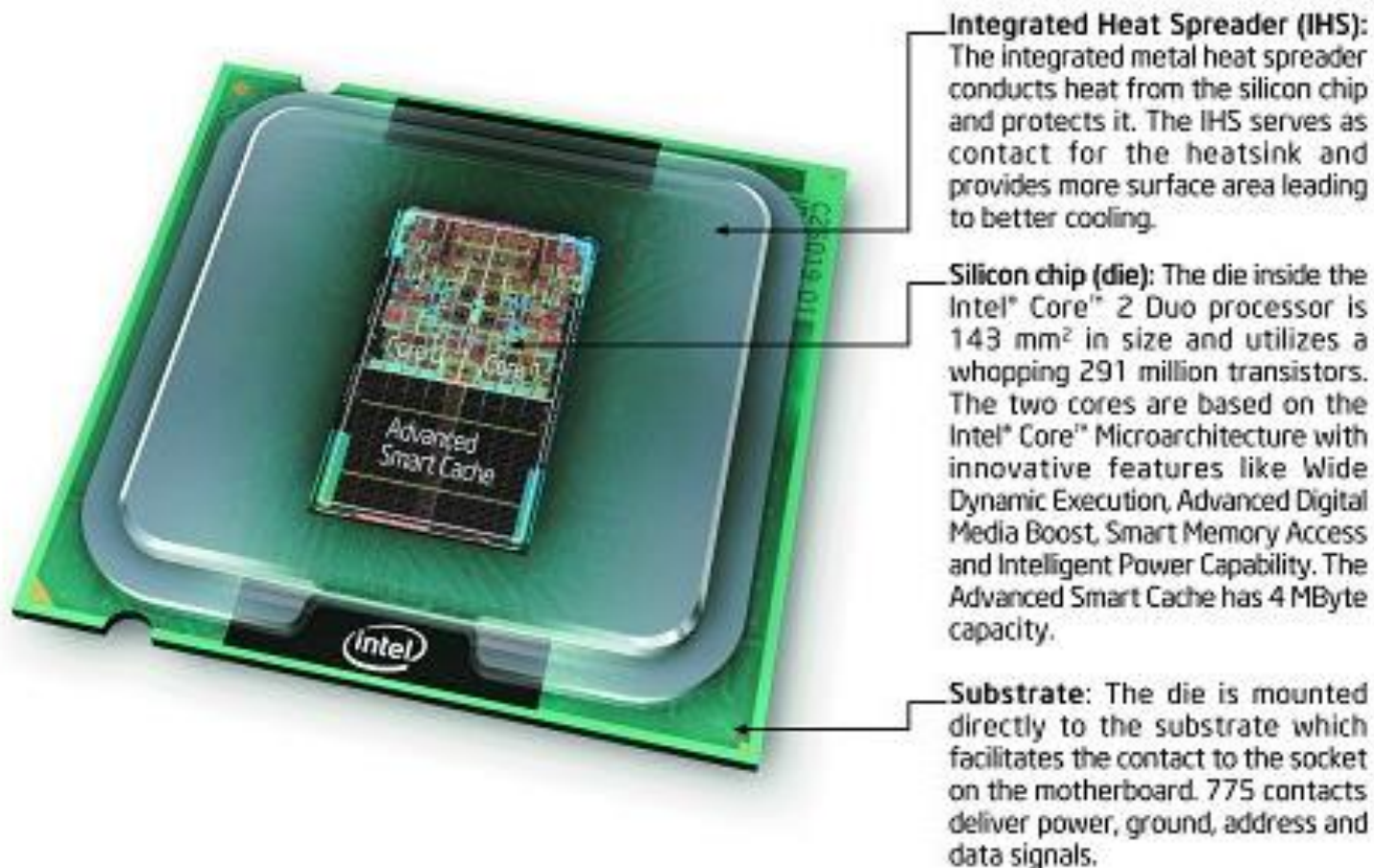
Chipset MCH

Up to 2X L2 to L1 Bandwidth

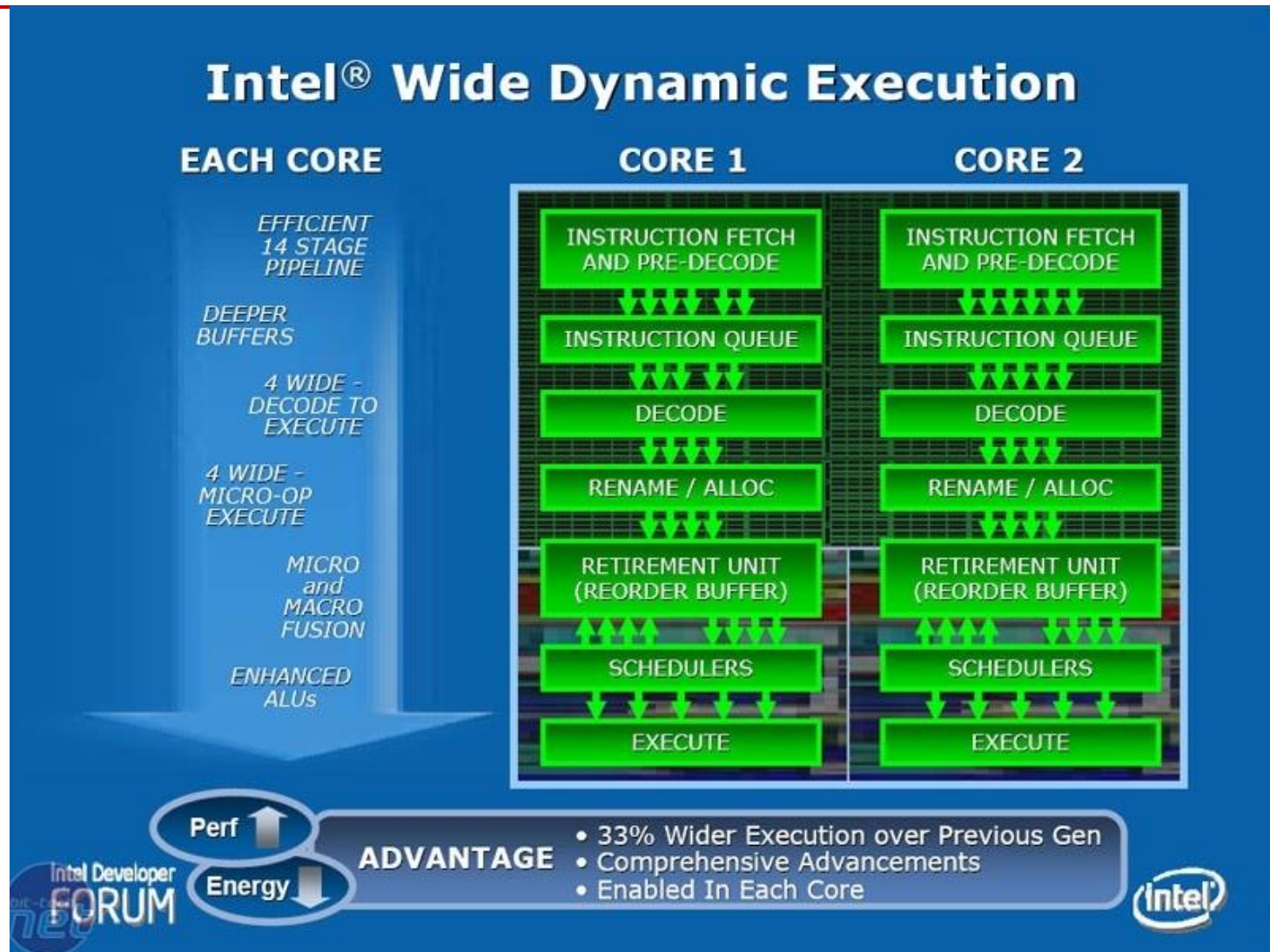


- What happens when L2 is *too* large?

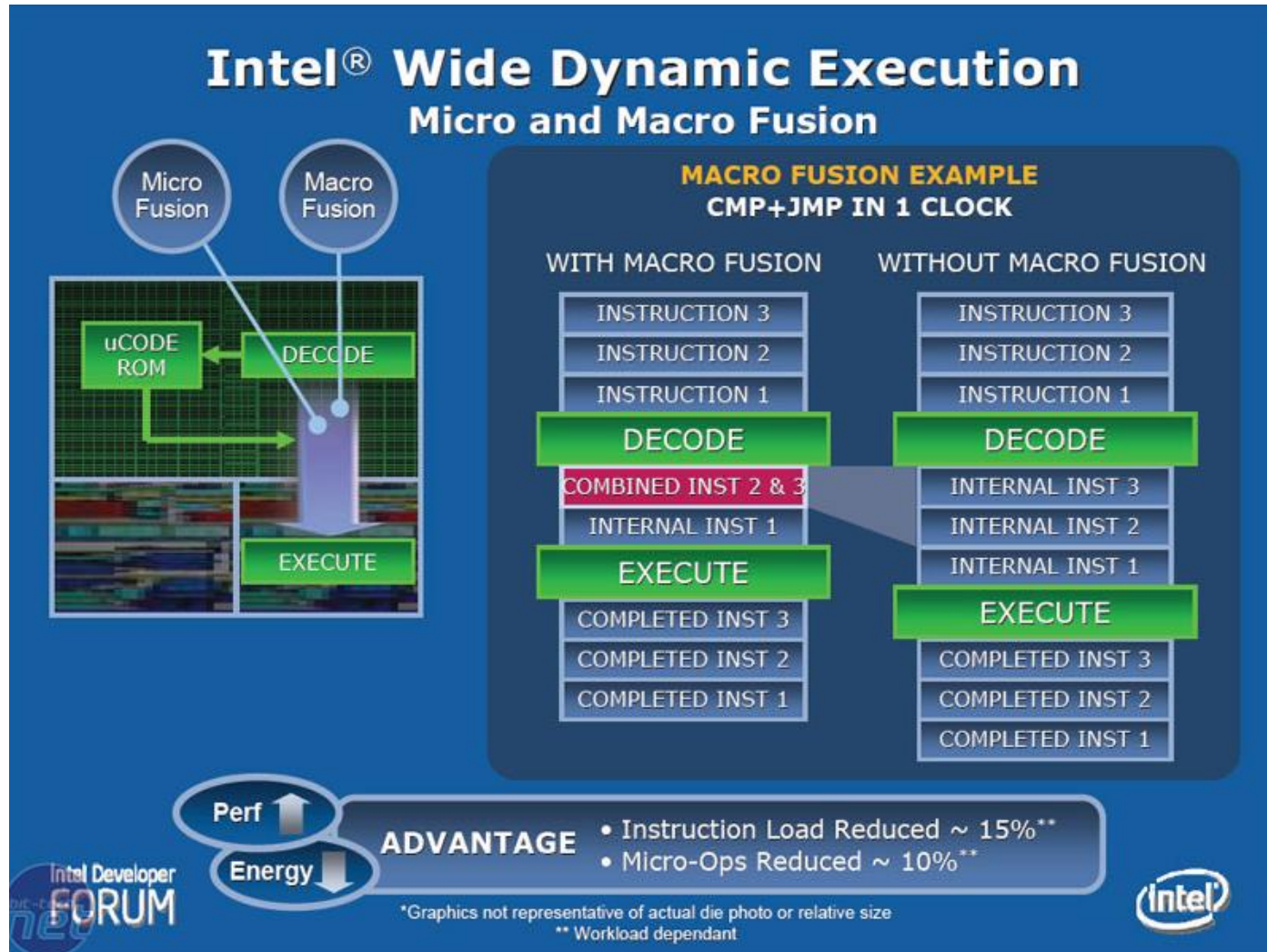
Intel Core 2 Duo (Merom)



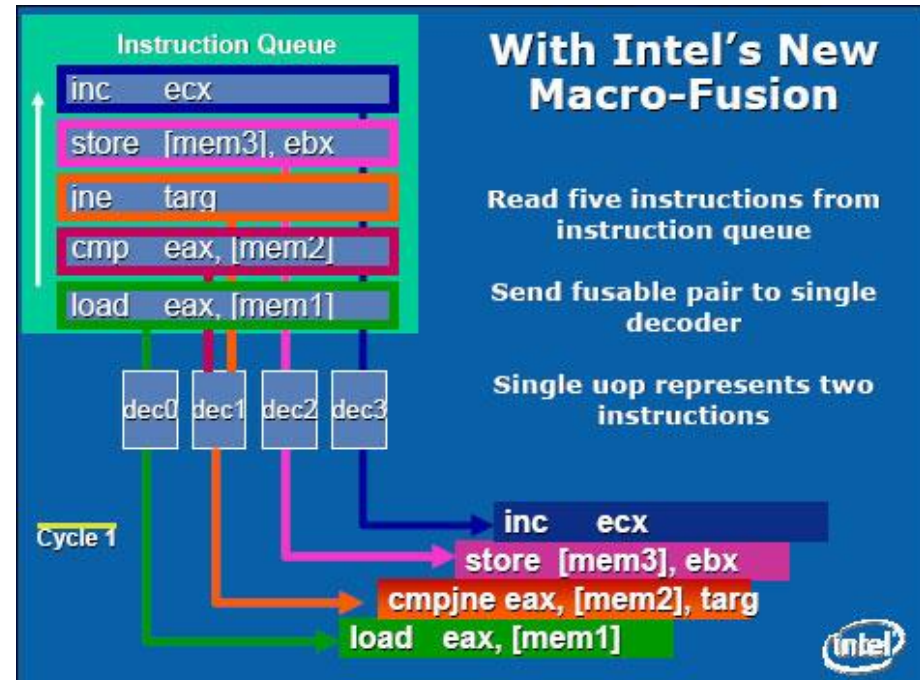
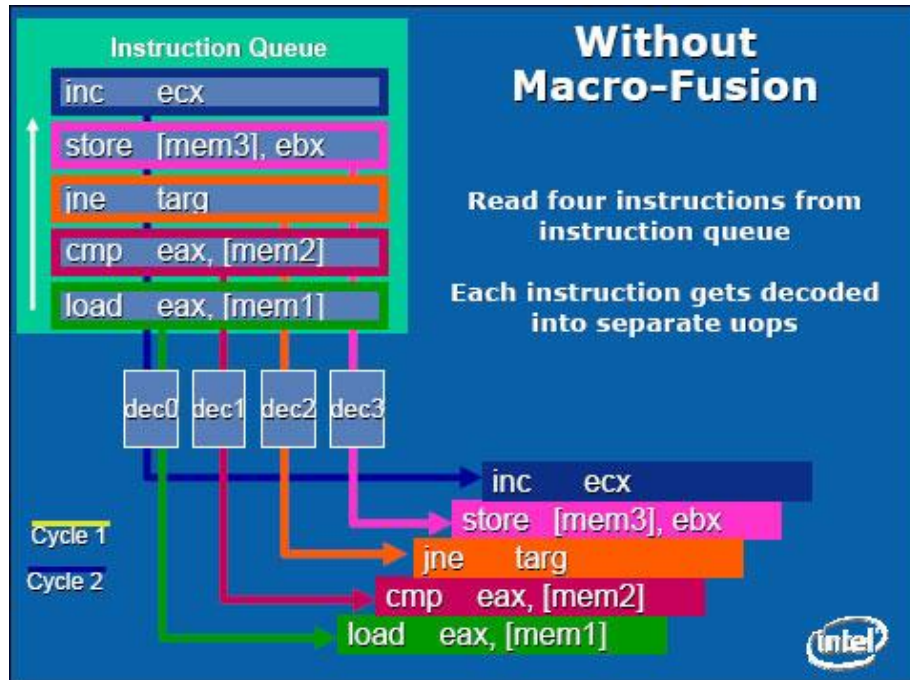
Core™ μ Arch — Wide Dynamic Execution



Core™ μ Arch — Wide Dynamic Execution



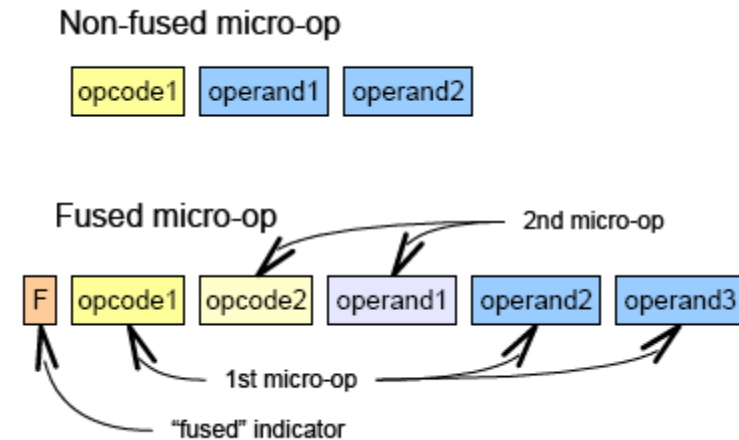
Core™ μ Arch — MACRO Fusion



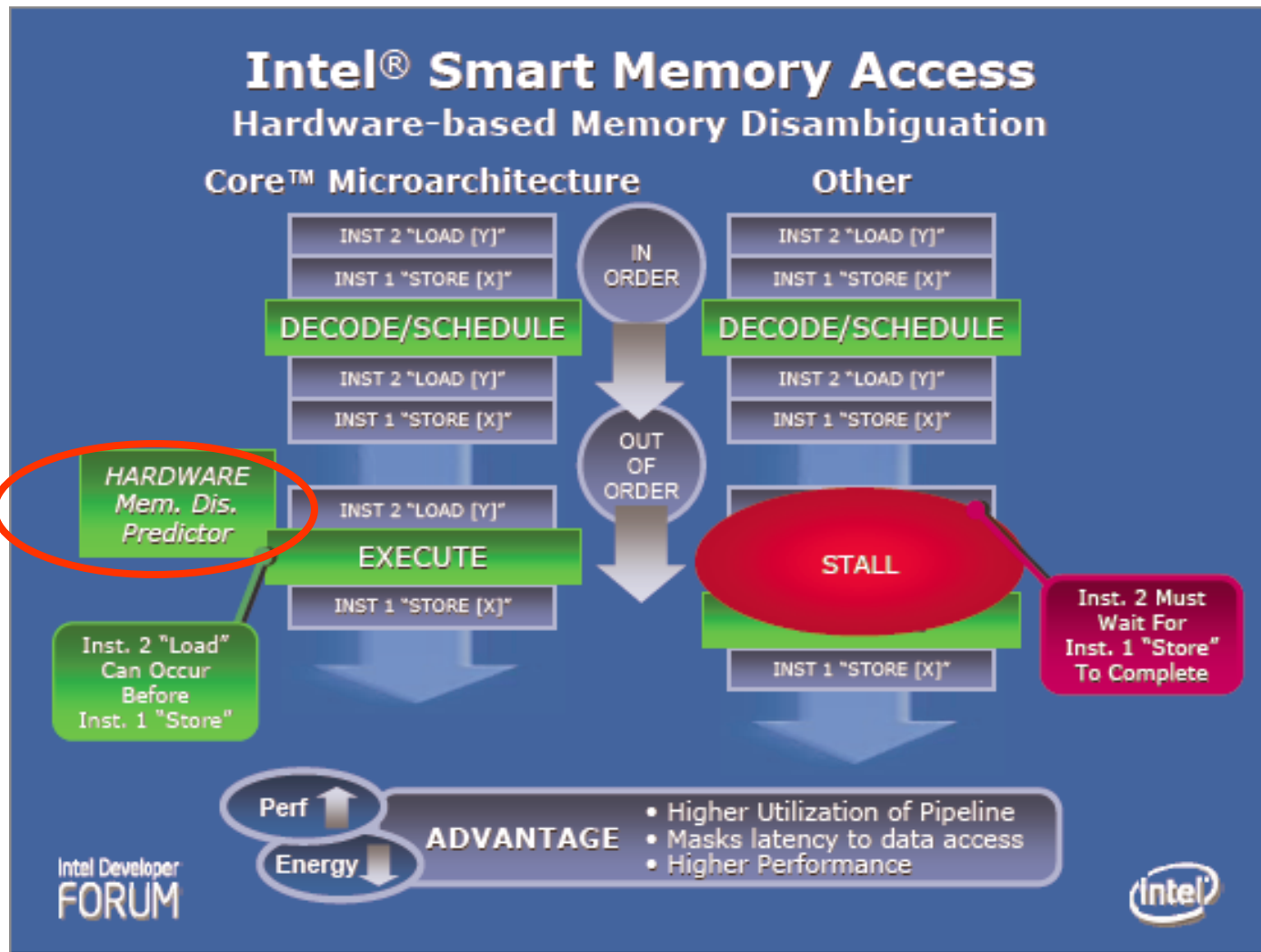
- Common “Intel 32” instruction pairs are combined
- 4-1-1-1 decoder that sustains 7 μ op’s per cycle
- 4+1 = 5 “Intel 32” instructions per cycle

Micro(-ops) Fusion (from Pentium M)

- To fuse
 - Store address and store data μ ops (e.g. `mov [esi], eax`)
 - Load-and-op μ ops (e.g. `add eax, [esi]`)
- Extend each RS entry to take 3 operands
- To reduce
 - micro-ops (10% reduction in the OOO logic)
 - Decoder bandwidth (simple decoder can decode fusion type instruction)
 - Energy consumption
- Performance improved by 5% for INT and 9% for FP (Pentium M data)

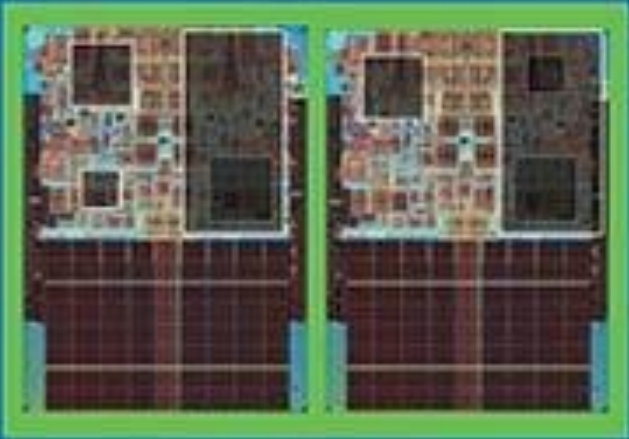


Smart Memory Access



Intel Quad-Core Processor - Kentsfield (Nov. 2006), Clovertown (2006)

Intel® Core™2 Extreme Quad-Core Processor Features



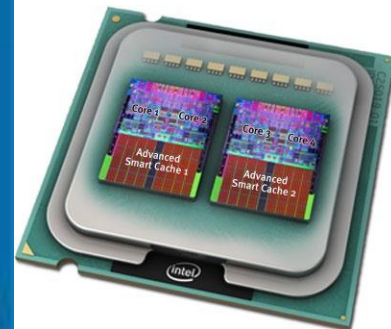
Not representative of relative die sizes

- 2.66 GHz, 1066 MHz front side bus, 8 MB total L2 cache
- Intel® VT, Intel® 64
- Supports Viiv™ technology
- Intel® Intelligent Power Capability
 - Shuts off inactive parts of the core
- Enhanced Halt State
 - Independent idle states per core
- Enhanced Intel SpeedStep® Technology
 - Reduces power in active work loads
- Execute Disable Bit
- Overspeed protection removed¹

Intel's Quad-Core Features Deliver the Latest in Desktop Processor Technology to the Most Demanding Users

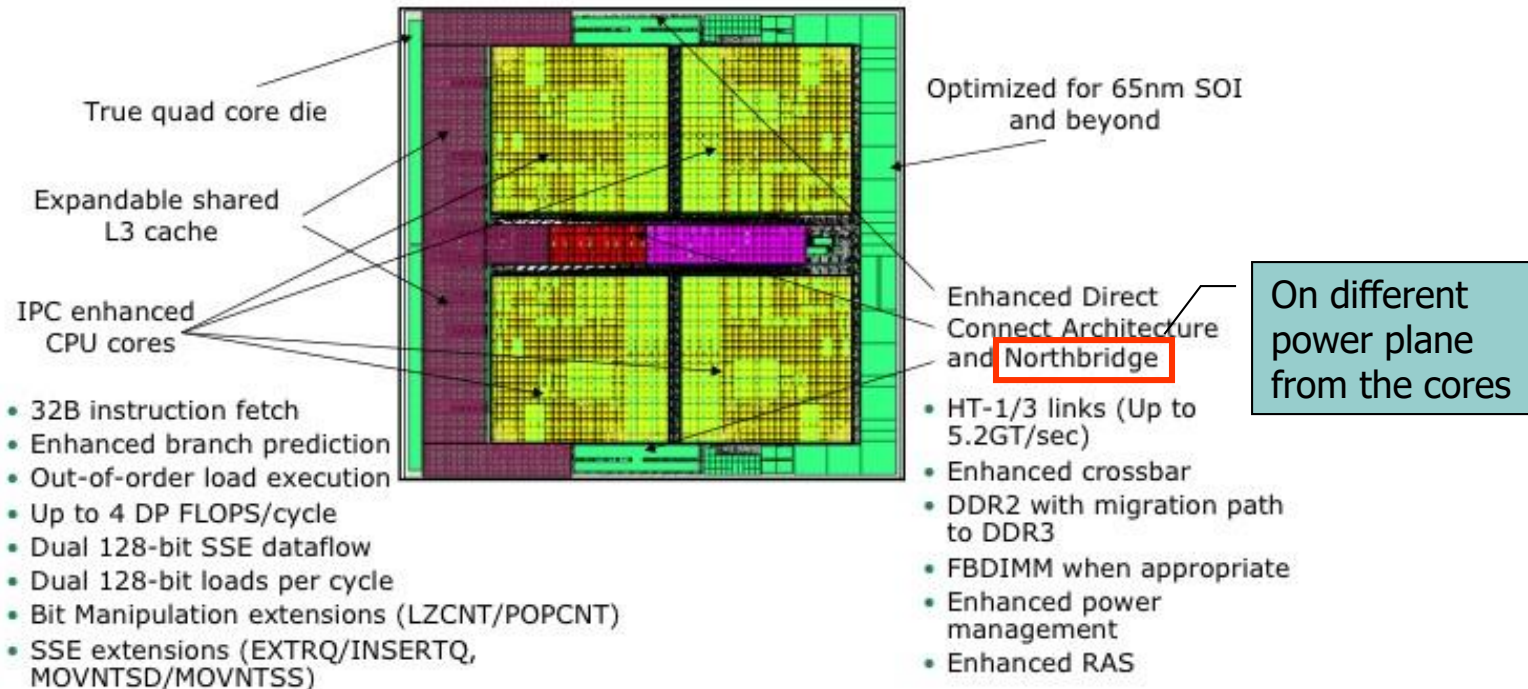
Intel Developer FORUM

Warning: altering clock frequency and/or voltage may (i) reduce system stability and useful life of the system and processor; (ii) cause the processor and other system components to fail; (iii) cause reductions in system performance; (iv) cause additional damage; and (v) affect system data integrity. Intel has not tested, and does not warranty, the operation of the processor beyond its specifications.



AMD Quad-Core Processor (Barcelona) (2007)

A Closer Look at AMD's Quad-Core CPU for '07



- True 128-bit SSE (as opposed 64 in prior Opteron)
- Sideband Stack optimizer
 - Parallelize many POPes and PUSHes (which were dependent on each other)
 - Convert them into pure loads/store instructions
 - No uops in FUs for stack pointer adjustment

Barcelona's Cache Architecture

Balanced, Highly Efficient Cache Structure

Dedicated L1

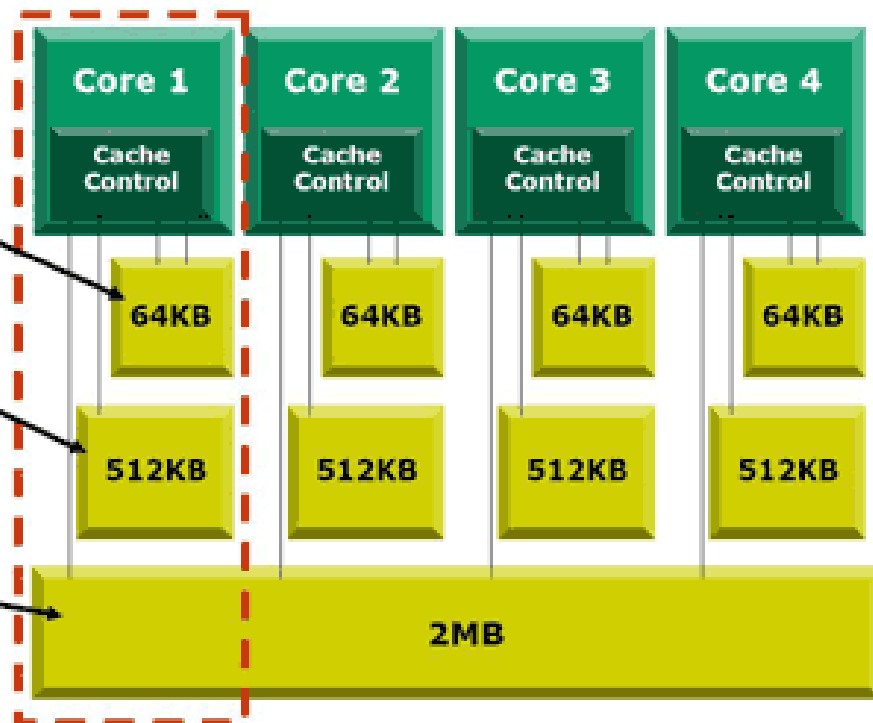
- Locality keeps most critical data in the L1 cache
- Lowest latency
- 2 loads per cycle

Dedicated L2

- Sized to accommodate the majority of working sets today
- Dedicated to eliminate conflicts common in shared caches
 - Better for Virtualization

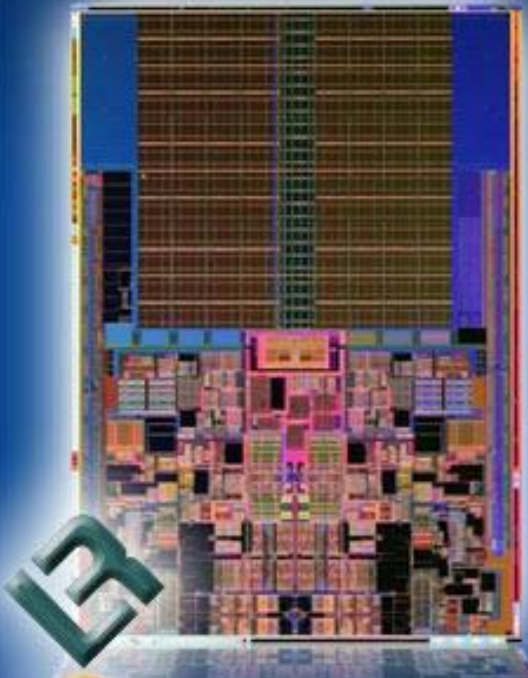
Shared L3 – NEW

- Victim-cache architecture maximizes efficiency of cache hierarchy
- Fills from L3 leave likely shared lines in the L3
- Sharing-aware replacement policy
- Ready for expansion at the right time for customers



Intel Penryn Dual-Core (First 45nm μ processor)

45nm Next Generation Intel® Core™ 2 and Xeon® Family processors (Penryn)



Built Upon Enhanced Intel Core Microarchitecture

Greater Performance at Given Frequency AND Higher Frequencies

Introduces New SSE4 Instructions For Media/Gaming/Graphics

New Levels of Energy Efficiency

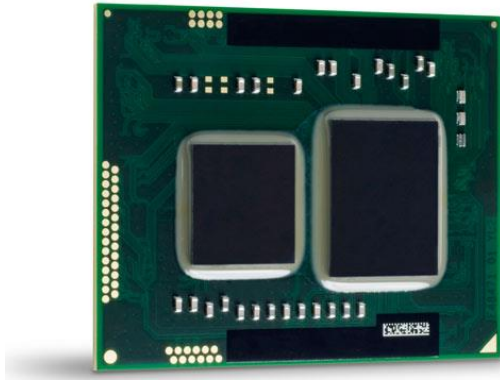
Larger Caches, Faster Buses

 Growing Performance and Energy Efficiency

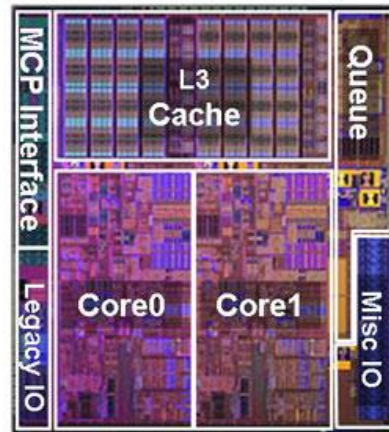
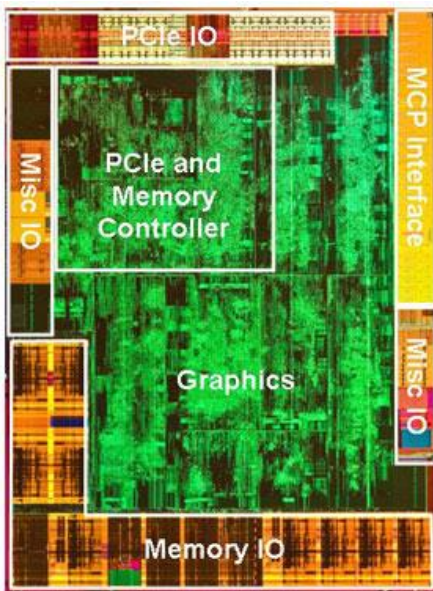
- High K dielectric metal gate
- 47 new SSE4 ISA
- Up to 12MB L2
- > 3GHz

Intel Arrandale Processor (2010)

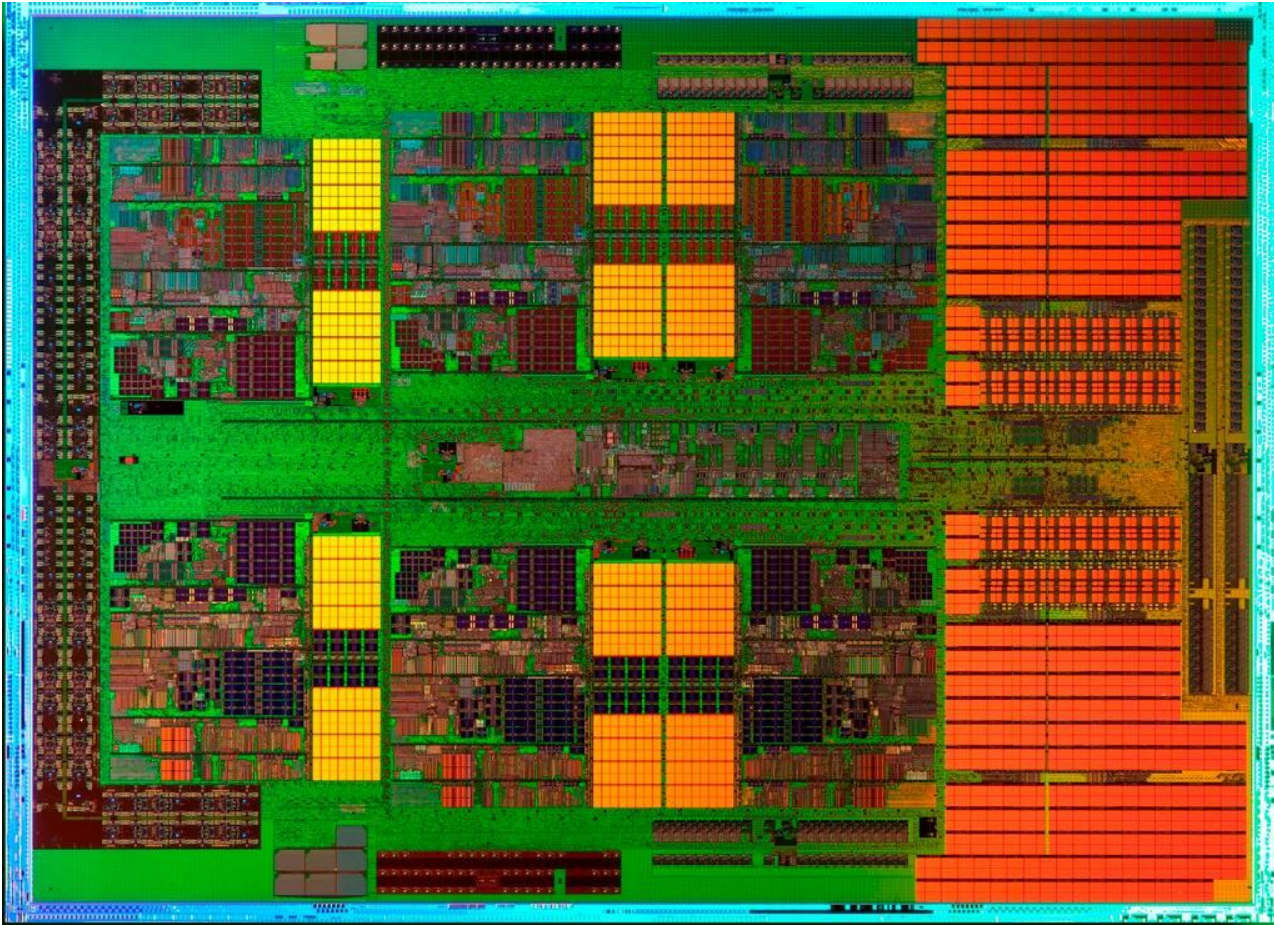
- Arrandale is the code name for a mobile Intel processor, sold as mobile Intel Core i3, i5, and i7 as well as Celeron and Pentium - Wikipedia



- 2 dies in package
- 32nm
- Unified 3MB L3
- Power sharing (Turbo Boost) between cores and gfx via DFS



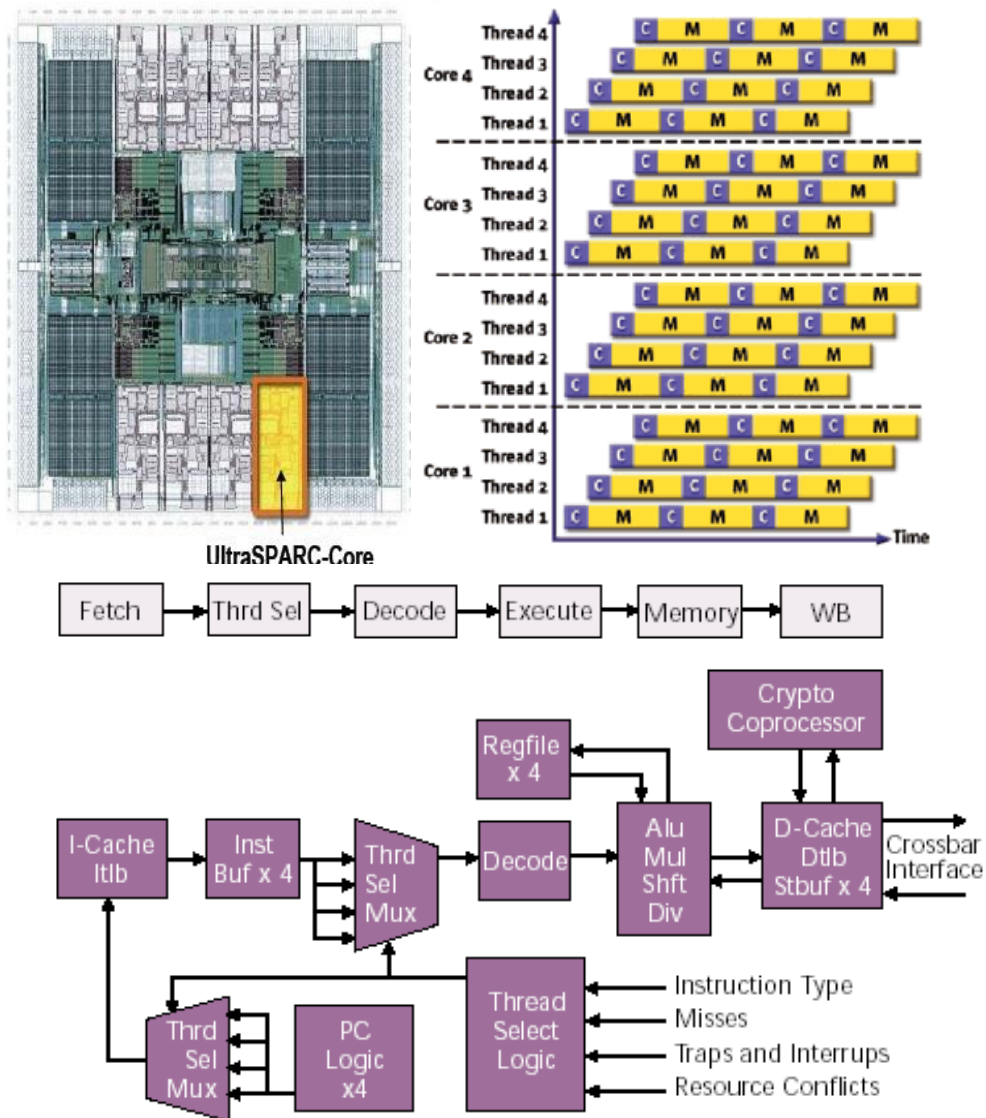
AMD 12-Core "Magny-Cours" Opteron (2010)



- 45nm
- 4 memory channels

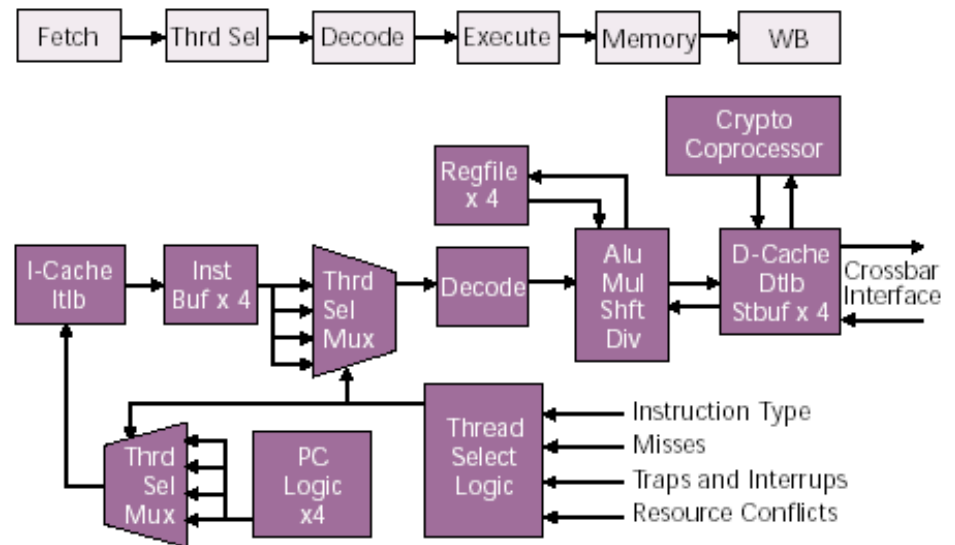
Sun UltraSparc T1 (2005)

- Eight cores, each 4-way threaded
- Fine-grained multithreading
 - a thread-selection logic
 - Take out threads that encounter long latency events
 - Round-robin cycle-by-cycle
 - 4 threads in a group share a processing pipeline (Sparc pipe)
- 1.2 GHz (90nm)
- In-order, 8 instructions per cycle (single issue from each core)
- Caches
 - 16K 4-way 32B L1-I
 - 8K 4-way 16B L1-D
 - Blocking cache (reason for MT)
 - 4-banked 12-way 3MB L2 + 4 memory controllers. (shared by all)
 - Data moved between the L2 and the cores using an integrated crossbar switch to provide high throughput (200GB/s)



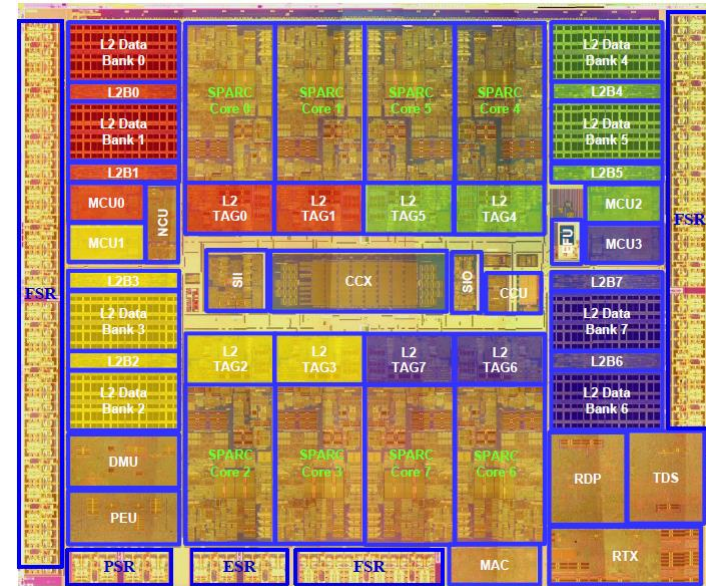
Sun UltraSparc T1 (2005)

- Thread-select logic marks a thread inactive based on
 - Instruction type
 - A predecode bit in the I-cache to indicate long-latency instruction
 - Misses
 - Traps
 - Resource conflicts



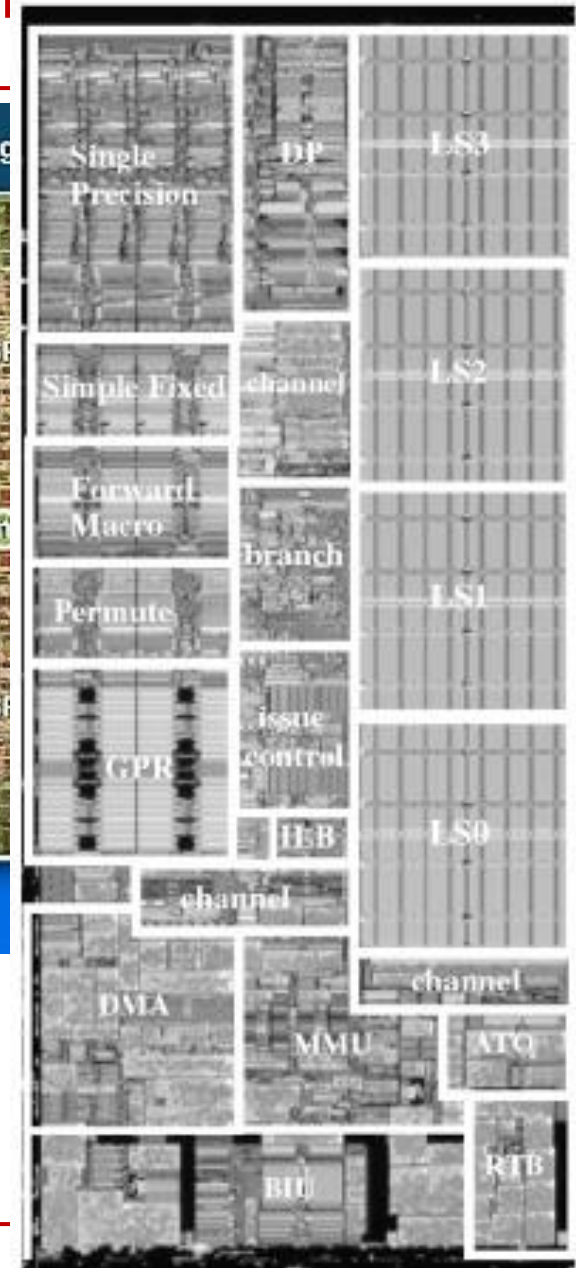
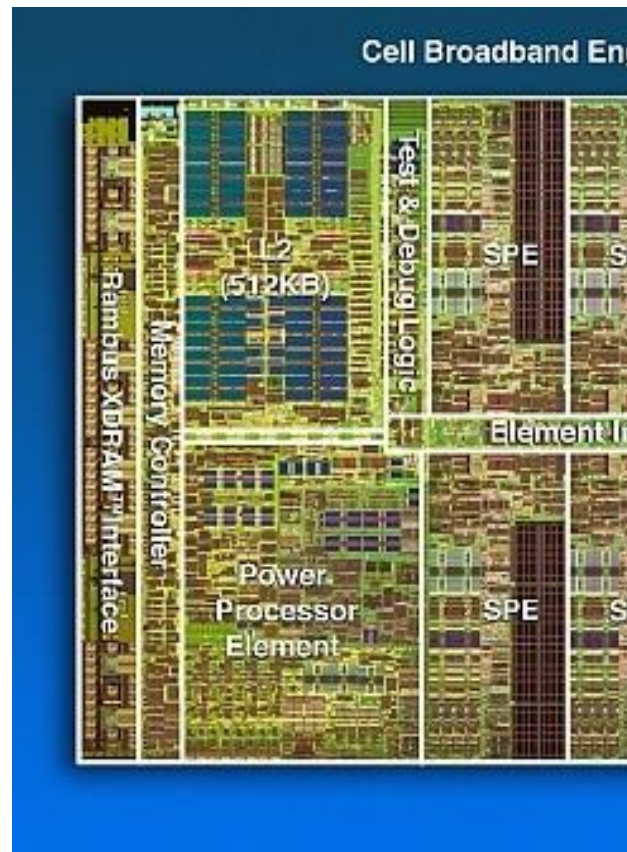
Sun UltraSparc T2 (2007)

- A fatter version of T1
- 1.4GHz (65nm)
- 8 threads per core, 8 cores on-die
- 1 FPU per core (1 FPU per die in T1), 16 INT EU (8 in T1)
- L2 increased to 8-banked 16-way 4MB shared
- 8 stage integer pipeline (as opposed to 6 for T1)
- 16 instructions per cycle
- One PCI Express port (x8 1.0)
- Two 10 Gigabit Ethernet ports with packet classification and filtering
- Eight encryption engines
- Four dual-channel FBDIMM (Fully Buffered DIMM) memory controllers
- 711 signal I/O, 1831 total



STI Cell Broadband Engine (2005)

- **Heterogeneous!**
- 9 cores, 10 threads
- 64-bit PowerPC (2-way multithreaded)
- Eight SPEs (Synergistic Processing Elements)
 - In-order, Dual-issue
 - 128-bit SIMD
 - 128x128b RF
 - 256KB LS (Local Storage)
 - Fast Local SRAM
 - Globally coherent DMA (128B/cycle)
 - 128+ concurrent transactions to memory per core
- High bandwidth
 - EIB (Element Interconnect Bus) (96B/cycle)



Backup Slides

List of Intel Xeon Microprocessors

- The Xeon microprocessor from Intel is a CPU brand targeted at the server and workstation markets
- It competes with AMD's Opteron

1 P6 based Xeons

1.1 Pentium II Xeon

1.1.1 "Drake" (250 nm)

1.2 Pentium III Xeon

1.2.1 "Tanner" (250 nm)

1.2.2 "Cascades" (180 nm)

2 NetBurst based Xeons

2.1 Xeon (UP/DP)

2.1.1 "Foster" (180 nm)

2.1.2 "Prestonia" (130 nm)

2.1.3 "Gallatin" (130 nm)

2.1.4 "Nocona" (90 nm)

2.1.5 "Irwindale" (90 nm)

2.2 Xeon (UP/DP), Dual Core

2.2.1 "Paxville DP" (90 nm)

2.2.2 "Dempsey" (65 nm)

2.3 Xeon MP

2.3.1 "Foster MP" (180 nm)

2.3.2 "Gallatin" (130 nm)

2.3.3 "Cranford" (90 nm)

2.3.4 "Potomac" (90 nm)

2.4 Xeon MP, Dual Core

2.4.1 "Paxville MP" (90 nm)

2.4.2 "Tulsa" (65 nm)

3 Pentium M (Yonah) based Xeons

3.1 Xeon DP, Dual Core

3.1.1 "Sossaman" (65 nm)

4 Core based Xeons

4.1 Xeon (UP/DP), Dual Core

4.1.1 "Allendale" (65 nm)

4.1.2 "Conroe" (65 nm)

4.1.3 "Woodcrest" (65 nm)

4.1.4 "Wolfdale-CL" (45 nm)

4.1.5 "Wolfdale" (45 nm)

4.1.6 "Wolfdale-DP" (45 nm)

4.2 Xeon (UP/DP), Quad Core

4.2.1 "Kentsfield" (65 nm)

4.2.2 "Yorkfield-6M" (45 nm)

4.2.3 "Yorkfield" (45 nm)

4.2.4 "Yorkfield-CL" (45 nm)

4.2.5 "Clovertown" (65 nm)

4.2.6 "Harpertown" (45 nm)

4.3 Xeon MP, Multi Core

4.3.1 "Tigerton" (65 nm)

4.3.2 "Dunnington" (45 nm)

5 Nehalem based Xeons

5.1 Xeon 3000-series (uniprocessor)

5.1.1 "Clarkdale" (32 nm)

5.1.2 "Lynnfield" (45 nm)

5.1.3 "Bloomfield" (45 nm)

5.1.4 "Jasper Forest" (45 nm)

5.1.5 "Gulftown" (32 nm)

5.2 Xeon 5000-series (dual-processor)

5.2.1 "Gainestown" (45 nm)

5.2.2 "Jasper Forest" (45 nm)

5.2.3 "Westmere-EP" (32 nm)

5.3 Xeon 7000-series (multiprocessor)

5.3.1 "Beckton" (45 nm)

5.4 Xeon E7 (multiprocessor)

5.4.1 "Westmere-EX" (32 nm)

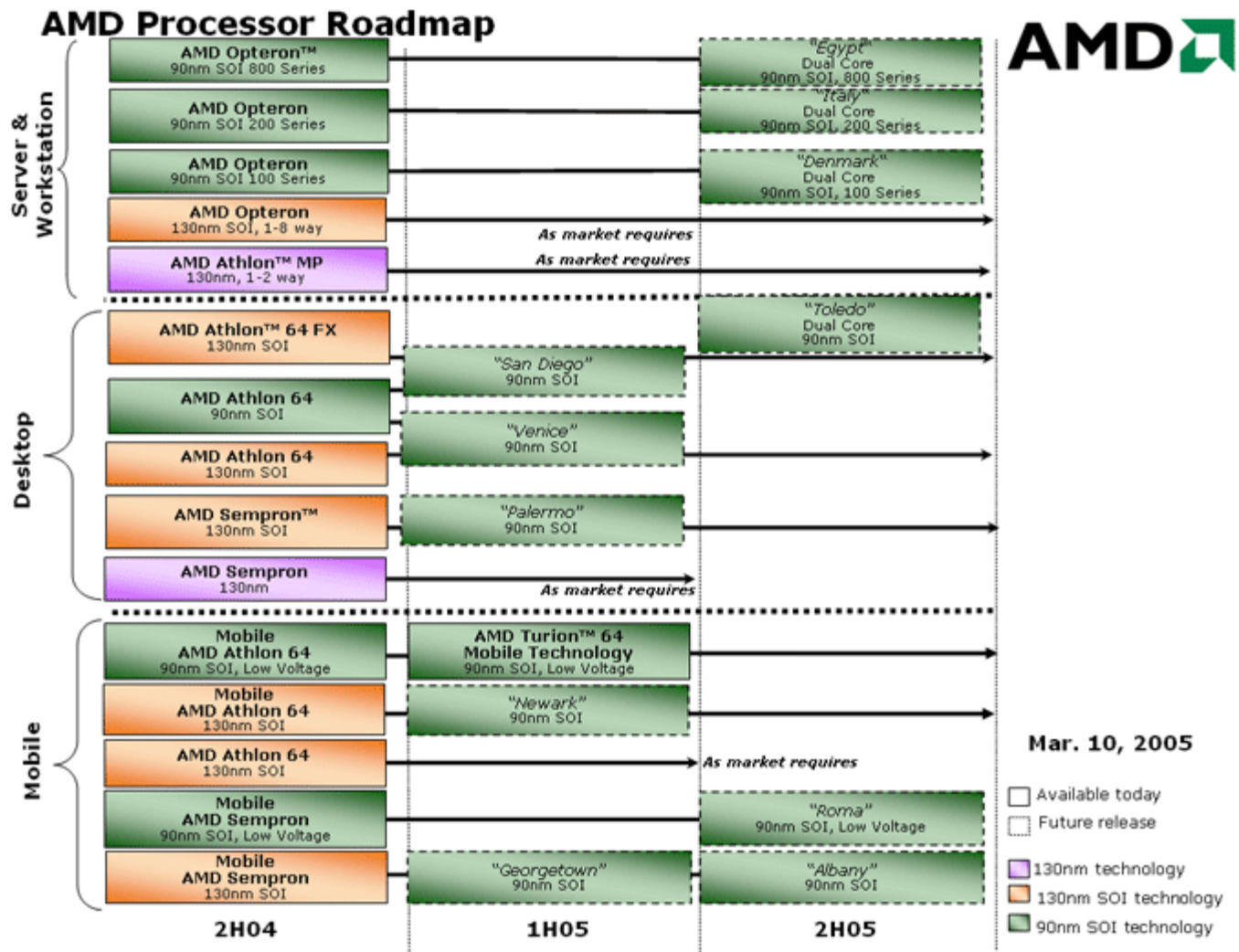
6 Sandy Bridge based Xeons

6.1 Xeon E3 (uniprocessor)

6.1.1 "Sandy Bridge" (32 nm)

Source: Wikipedia http://en.wikipedia.org/wiki/List_of_Intel_Xeon_microprocessors

AMD Roadmap (as of 2005)



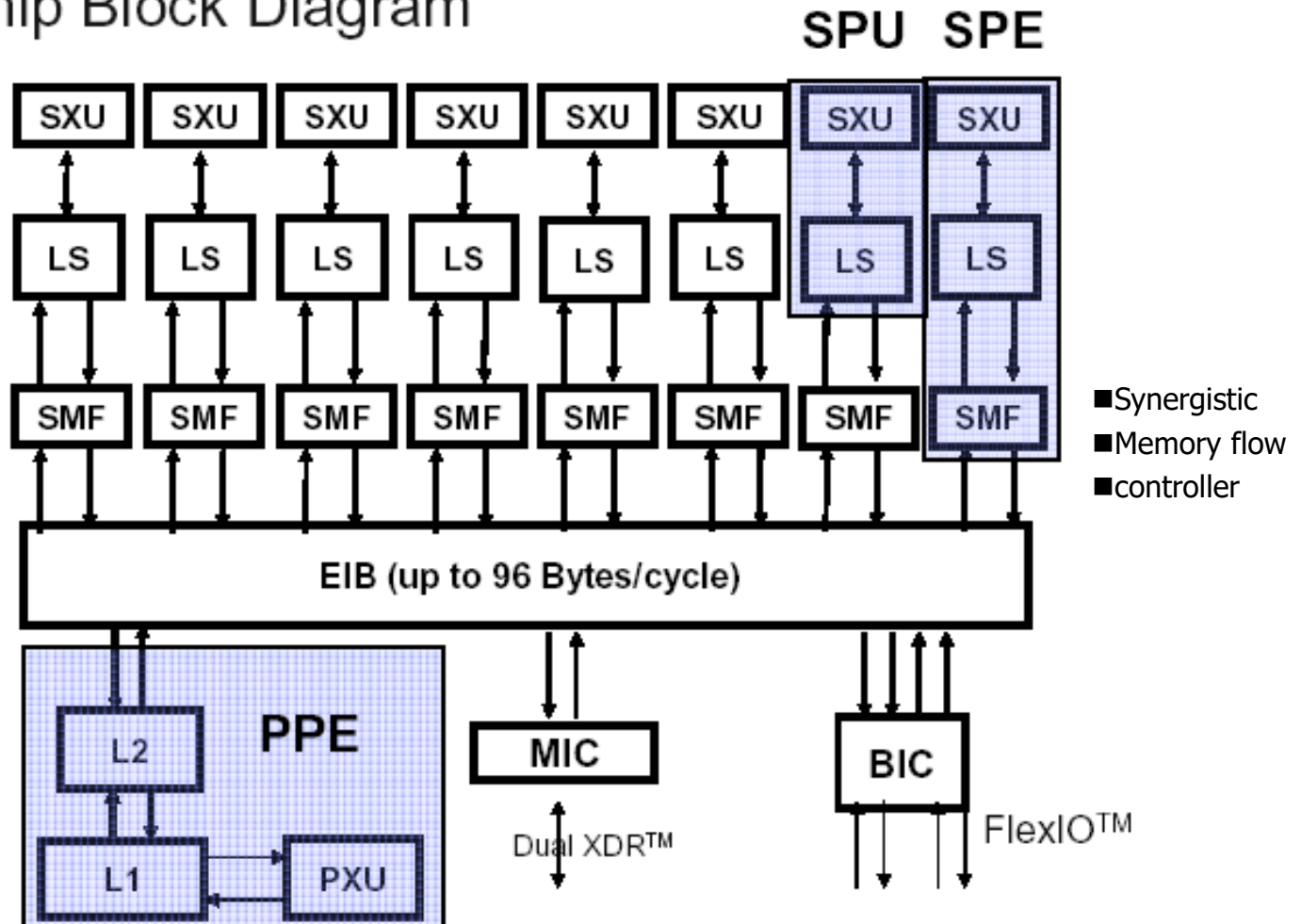
Alpha 21464 (EV8) Processor

Technology

- Leading edge process technology – 1.2 ~ 2.0GHz
 - 0.125μm CMOS
 - SOI-compatible
 - Cu interconnect
 - low-k dielectrics
- Chip characteristics
 - ~1.2V Vdd
 - ~250 Million transistors
 - ~1100 signal pins in flip chip packaging

Cell Chip Block Diagram

Cell Chip Block Diagram

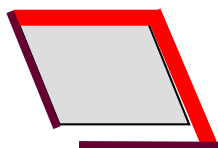


EV8 SMT

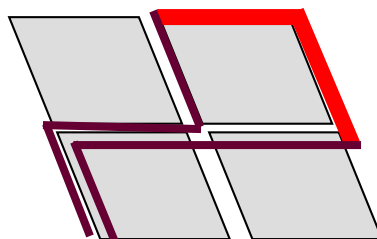
- In SMT mode, it is as if there are 4 processors on a chip that shares their caches and TLB
- Replicated hardware contexts
 - Program counter
 - Architected registers (*actually just the renaming table since architected registers and rename registers come from the same physical pool*)
- Shared resources
 - Rename register pool (larger than needed by 1 thread)
 - Instruction queue
 - Caches
 - TLB
 - Branch predictors
- Deceased before seeing the daylight.

Non-Uniform Cache Architecture

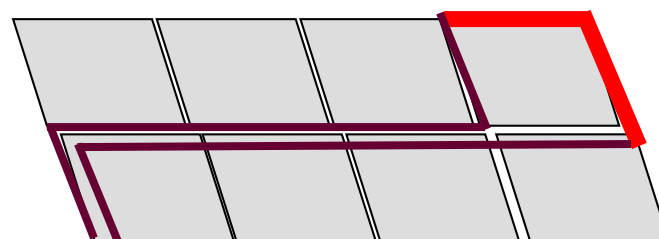
- ASPLOS 2002 proposed by UT-Austin
- Facts
 - Large shared on-die L2
 - Wire-delay dominating on-die cache



3 cycles
1MB
180nm, 1999

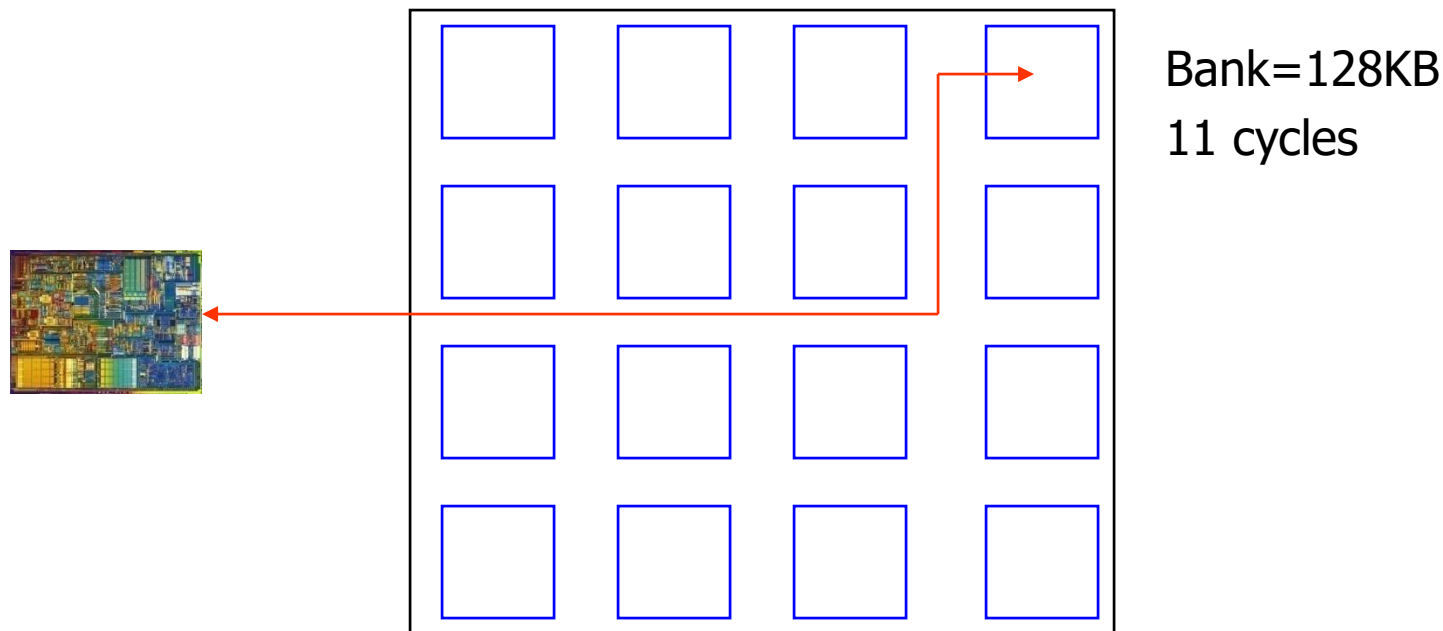


11 cycles
4MB
90nm, 2004



24 cycles
16MB
50nm, 2010

Multi-banked L2 cache

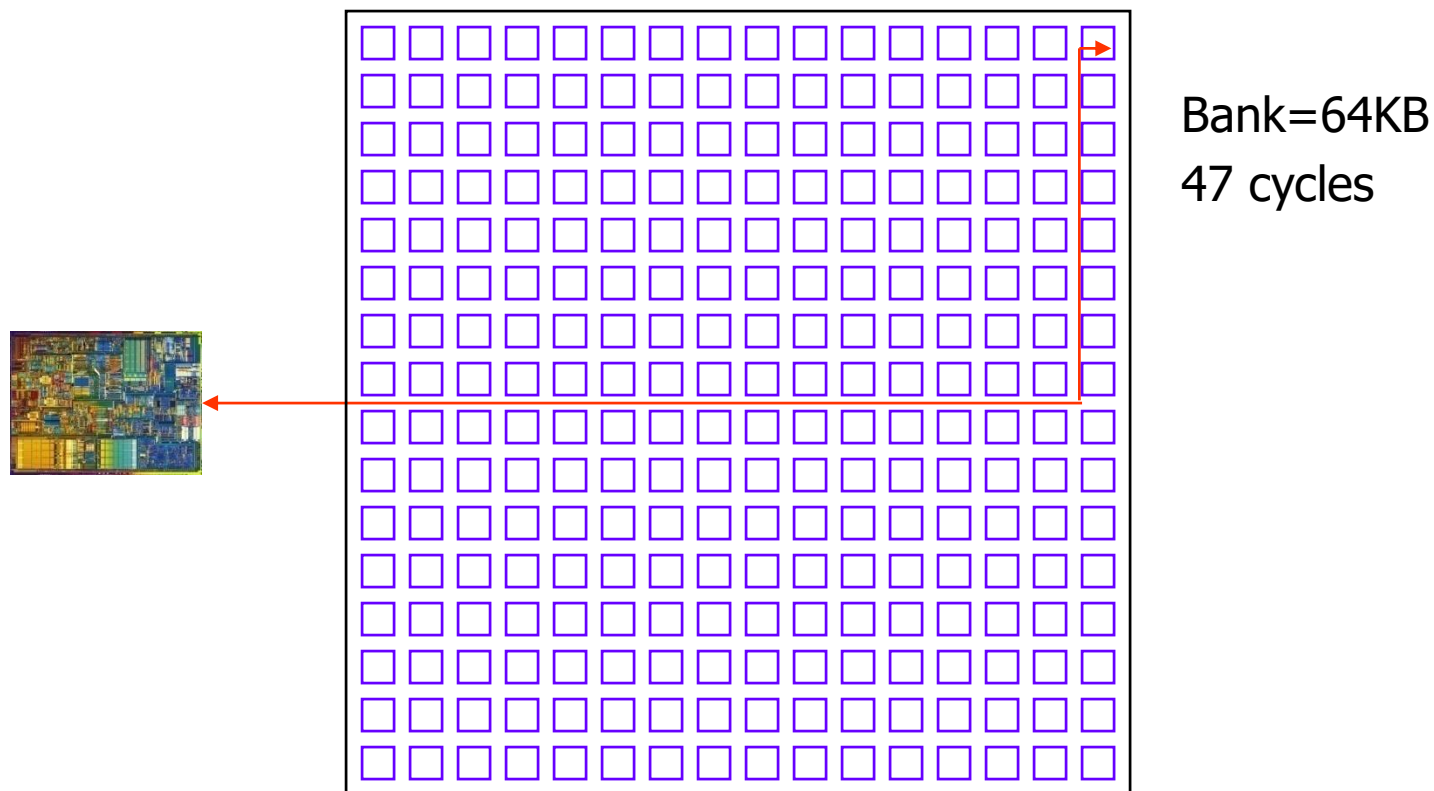


2MB @ 130nm

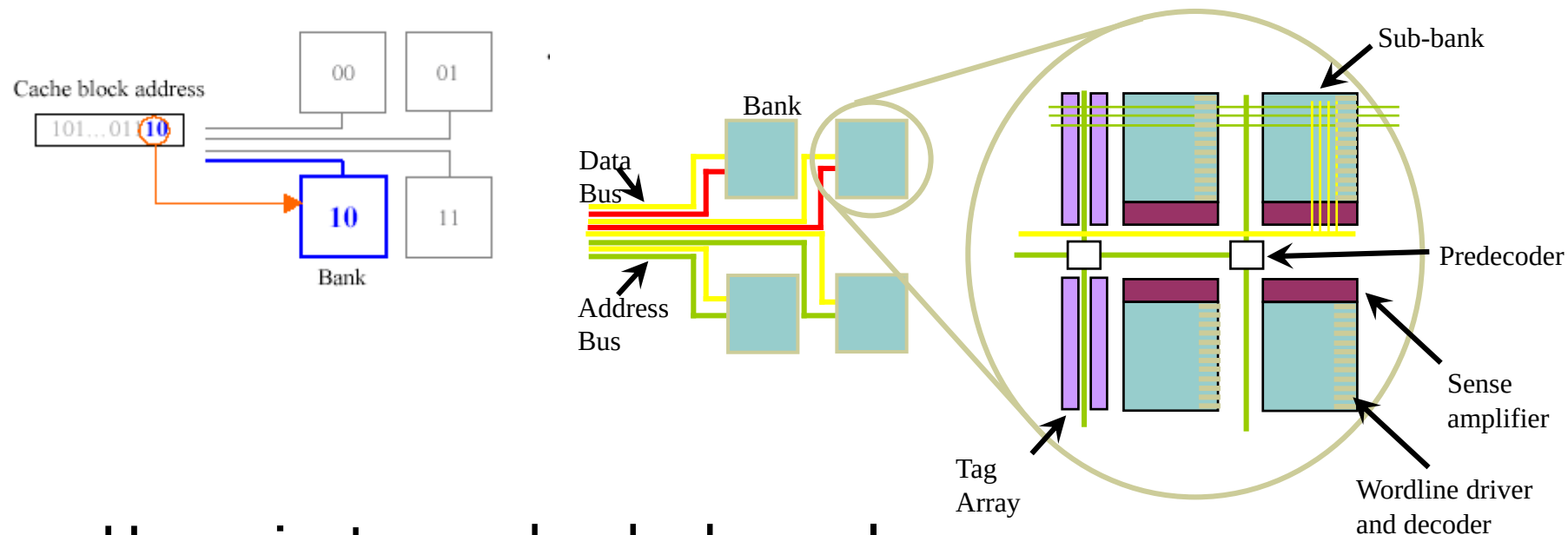
Bank Access time = 3 cycles

Interconnect delay = 8 cycles

Multi-banked L2 cache

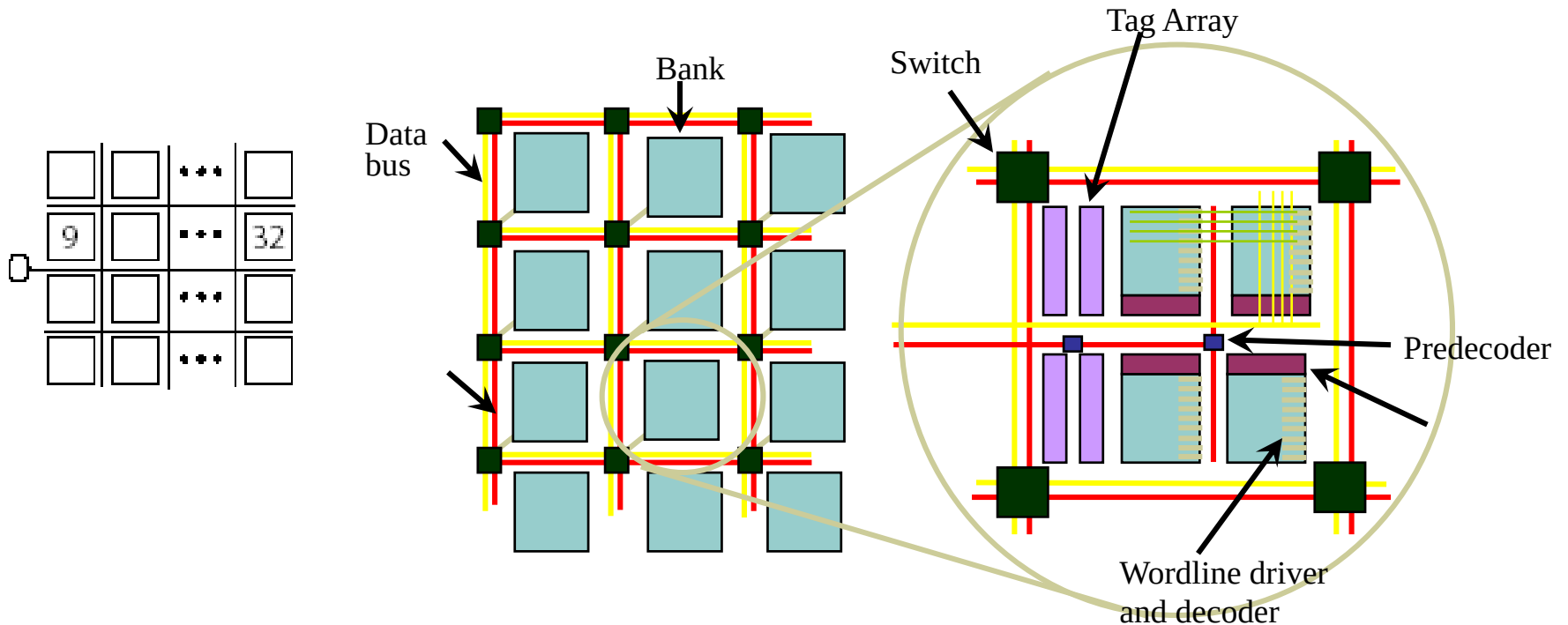


Static NUCA-1



- Use private per-bank channel
- Each bank has its distinct access latency
- Statically decide data location for its given address
- Average access latency = 34.2 cycles
- Wire overhead = 20.9% → an issue

Static NUCA-2



- Use a 2D switched network to alleviate wire area overhead
- Average access latency = 24.2 cycles
- Wire overhead = 5.9%