COMP4611: Design and Analysis of Computer Architectures

Memory System

Memory Hierarchy

Lin Gu

CSE, HKUST

Memory System

- Main memory generally uses Dynamic RAM (***DRAM***), which uses a single transistor to store a bit, but requires a periodic data refresh (~every 8 ms).
- Cache uses ***SRAM***: Static Random Access Memory
  - No refresh (6 transistors/bit vs. 1 transistor/bit for DRAM)
- *Size*: DRAM/SRAM *4-8*,
  Cost & Performance: SRAM/DRAM *8-16*
- Performance metrics
  - **Latency** is concern of cache
  - **Access time:** The time it takes between a memory access request and the time the requested information is available to cache/CPU.
  - **Cycle time:** The minimum time between unrelated requests to memory (greater than access time in DRAM to allow address lines to be stable)
  - **Memory bandwidth:** The maximum sustained data transfer rate between main memory and cache/CPU.

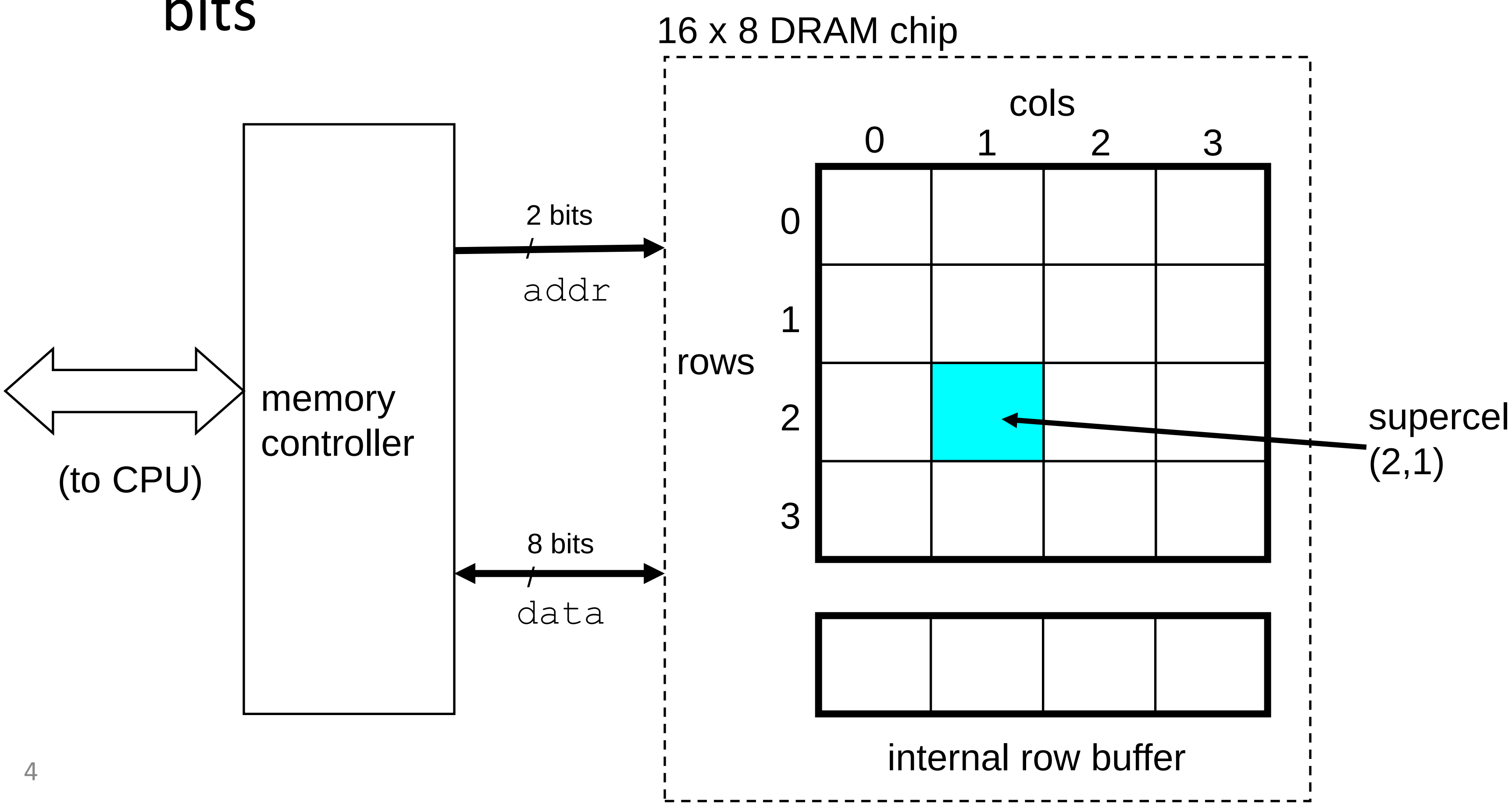
Memory Technology

- SRAM
  - Requires lower power to retain bit than DRAM
  - Requires 6 transistors/bit

- DRAM
  - Must be re-written after being read
  - Must also be periodically refeshed
    - Every ~ 8 ms
    - Each row can be refreshed simultaneously
  - One transistor/bit
  - Address lines are multiplexed:
    - Upper half of address: row access strobe (RAS)
    - Lower half of address: column access strobe (CAS)

Conventional DRAM Organization

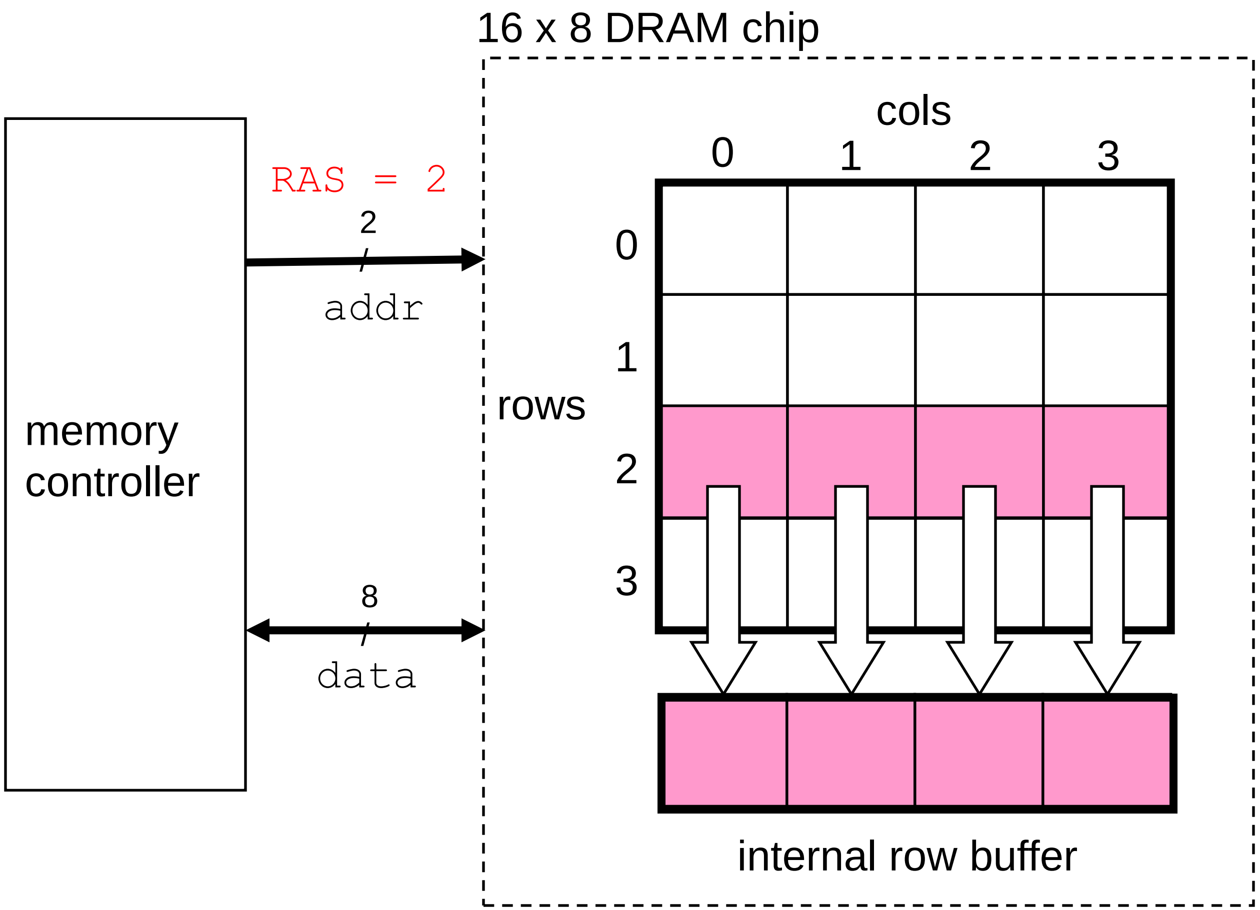
- d x w DRAM:
  - dw total bits organized as d supercells of size w bits



Reading DRAM Supercell (2,1)

Step 1(a): Row access strobe (RAS) selects row 2.

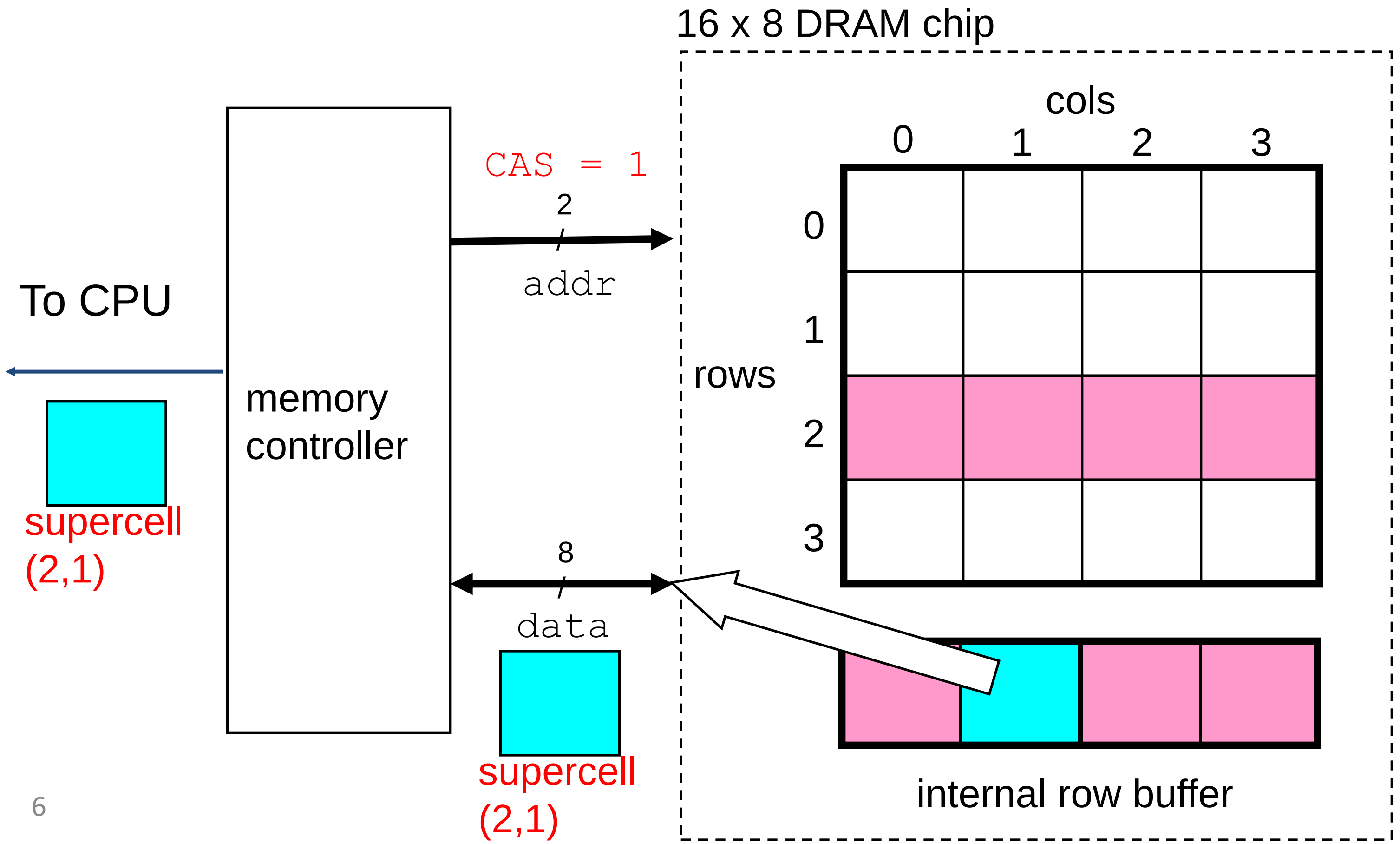
Step 1(b): Row 2 copied from DRAM array to row buffer.



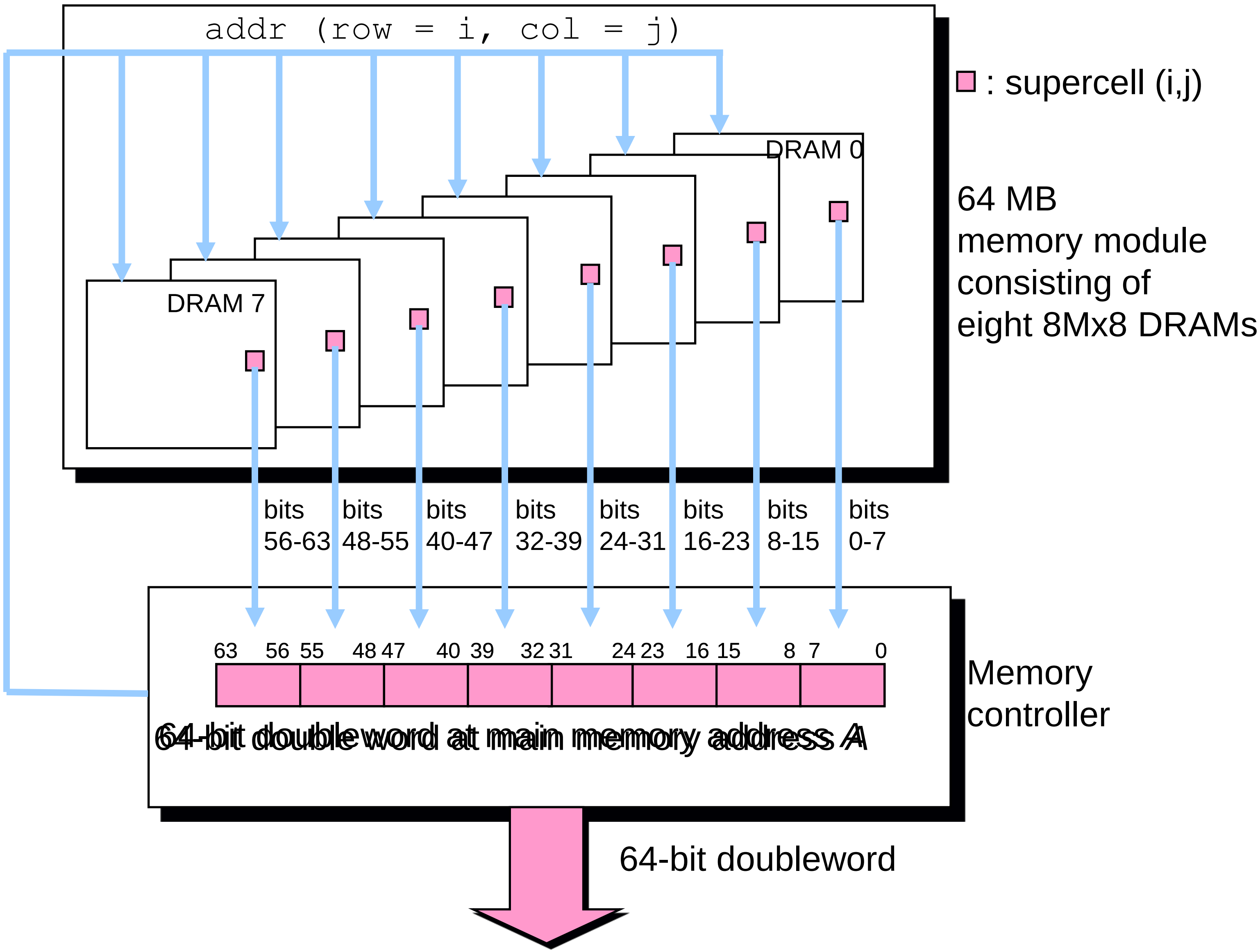
Reading DRAM Supercell (2,1)

Step 2(a): Column access strobe (CAS) selects column 1.

Step 2(b): Supercell (2,1) copied from buffer to data lines, and eventually back to the CPU.



Memory Modules



Memory Optimizations

Production year	Chip size	DRAM Type	Row access strobe (RAS) Slowest DRAM (ns)	Row access strobe (RAS) Fastest DRAM (ns)	Column access strobe (CAS)/ data transfer time (ns)	Cycle time (ns)
1980	64K bit	DRAM	180	150	75	250
1983	256K bit	DRAM	150	120	50	220
1986	1M bit	DRAM	120	100	25	190
1989	4M bit	DRAM	100	80	20	165
1992	16M bit	DRAM	80	60	15	120
1996	64M bit	SDRAM	70	50	12	110
1998	128M bit	SDRAM	70	50	10	100
2000	256M bit	DDR1	65	45	7	90
2002	512M bit	DDR1	60	40	5	80
2004	1G bit	DDR2	55	35	5	70
2006	2G bit	DDR2	50	30	2.5	60
2010	4G bit	DDR3	36	28	1	37
2012	8G bit	DDR3	30	24	0.5	31

Figure 2.13 Times of fast and slow DRAMs vary with each generation. (Cycle time is defined on page 95.) Performance improvement of row access time is about 5% per year. The improvement by a factor of 2 in column access in 1986 accompanied the switch from NMOS DRAMs to CMOS DRAMs. The introduction of various burst transfer modes in the mid-1990s and SDRAMs in the late 1990s has significantly complicated the calculation of access time for blocks of data; we discuss this later in this section when we talk about SDRAM access time and power. The DDR4 designs are due for introduction in mid- to late 2012. We discuss these various forms of DRAMs in the next few pages.

Memory Optimizations

Standard	Clock rate (MHz)	M transfers per second	DRAM name	MB/sec /DIMM	DIMM name
DDR	133	266	DDR266	2128	PC2100
DDR	150	300	DDR300	2400	PC2400
DDR	200	400	DDR400	3200	PC3200
DDR2	266	533	DDR2-533	4264	PC4300
DDR2	333	667	DDR2-667	5336	PC5300
DDR2	400	800	DDR2-800	6400	PC6400
DDR3	533	1066	DDR3-1066	8528	PC8500
DDR3	666	1333	DDR3-1333	10,664	PC10700
DDR3	800	1600	DDR3-1600	12,800	PC12800
DDR4	1066–1600	2133–3200	DDR4-3200	17,056–25,600	PC25600

Figure 2.14 Clock rates, bandwidth, and names of DDR DRAMS and DIMMs in 2010. Note the numerical relationship between the columns. The third column is twice the second, and the fourth uses the number from the third column in the name of the DRAM chip. The fifth column is eight times the third column, and a rounded version of this number is used in the name of the DIMM. Although not shown in this figure, DDRs also specify latency in clock cycles as four numbers, which are specified by the DDR standard. For example, DDR3-2000 CL 9 has latencies of 9-9-9-28. What does this mean? With a 1 ns clock (clock cycle is one-half the transfer rate), this indicate 9 ns for row to columns address (RAS time), 9 ns for column access to data (CAS time), and a minimum read time of 28 ns. Closing the row takes 9 ns for precharge but happens only when the reads from that row are finished. In burst mode, transfers occur on every clock on both edges, when the first RAS and CAS times have elapsed. Furthermore, the precharge in not needed until the entire row is read. DDR4 will be produced in 2012 and is expected to reach clock rates of 1600 MHz in 2014, when DDR5 is expected to take over. The exercises explore these details further.

Memory Optimizations

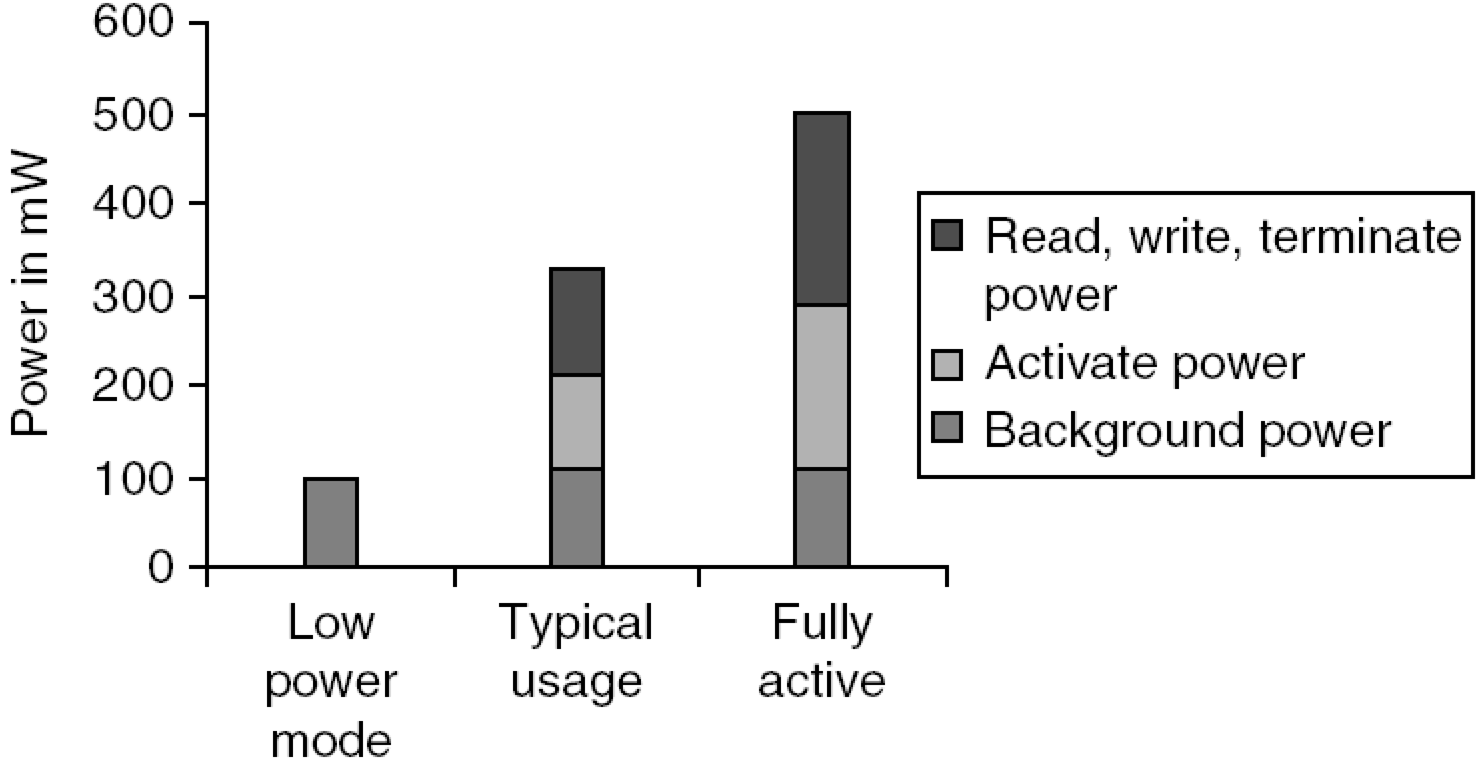
- DDR (Double Data Rate):
  - DDR2
    - Lower power (2.5 V -> 1.8 V)
    - Higher clock rates (266 MHz, 333 MHz, 400 MHz)
  - DDR3
    - 1.5 V
    - 800 MHz
  - DDR4
    - 1-1.2 V
    - 1600 MHz
- GDDR5 is graphics memory based on DDR3

Memory Optimizations

- Graphics memory:
  - Achieve 2-5 X bandwidth per DRAM vs. DDR3
    - Wider interfaces (32 vs. 16 bit)
    - Higher clock rate
      - Possible because they are attached via soldering instead of socketted DIMM modules

- Reducing power in SDRAMs:
  - Lower voltage
  - Low power mode (ignores clock, continues to refresh)

Memory Power Consumption



Flash Memory

- Type of EEPROM
- Must be erased (in blocks) before being overwritten
- Non-volatile
- Limited number of write cycles
- Cheaper than SDRAM, more expensive than disk
- Slower than SRAM, faster than disk

Memory Dependability

- Memory is susceptible to cosmic rays
- *Soft errors*: dynamic errors
  - Detected and fixed by error correcting codes (ECC)
- *Hard errors*: permanent errors
  - Use sparse rows to replace defective rows

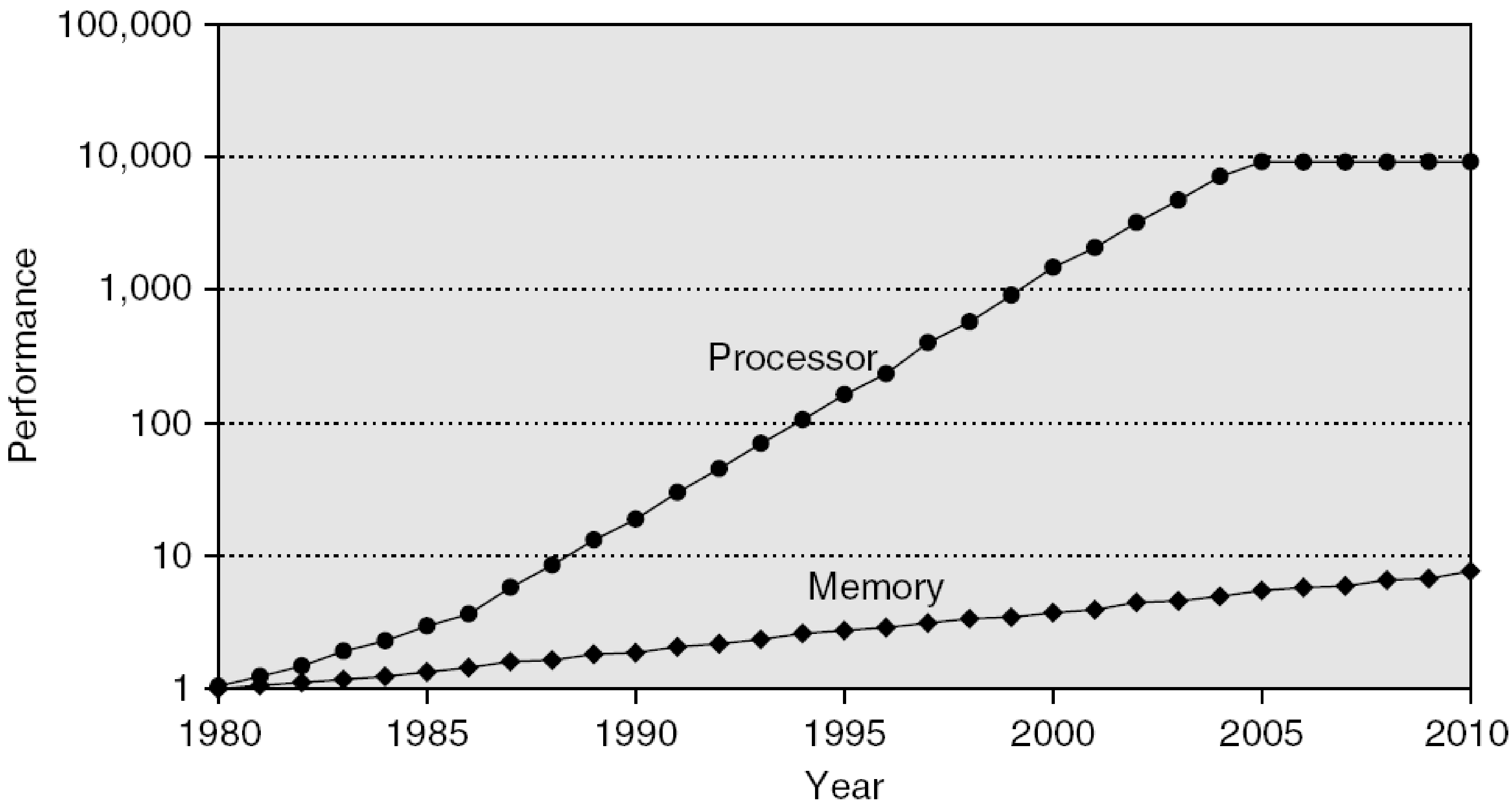
- Chipkill: a RAID-like error recovery technique

Memory Technology

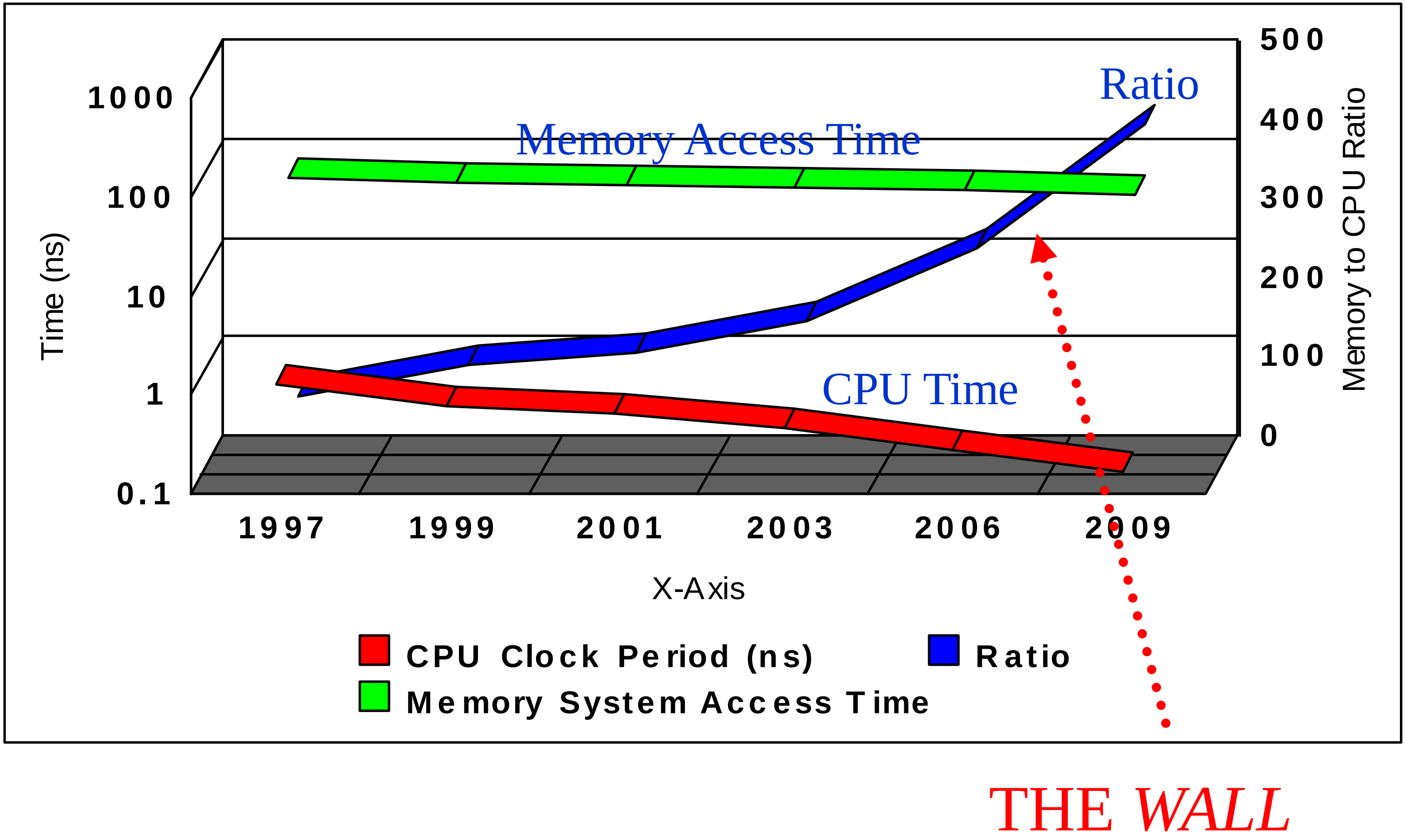
- Amdahl:
  - Memory capacity should grow linearly with processor speed
  - Unfortunately, memory capacity and speed has not kept pace with processors

- Some optimizations:
  - Multiple accesses to same row
  - Synchronous DRAM
    - Added clock to DRAM interface
    - Burst mode with critical word first
  - Wider interfaces
  - Double data rate (DDR)
  - Multiple banks on each DRAM device

Memory Performance Gap



Latency in a Single PC



Technology Trends

	Capacity	Speed (latency)
Logic:	2x in 3 years	2x in 3 years
DRAM:	4x in 3 years	2x in 10 years
Disk:	4x in 3 years	2x in 10 years

DRAM Generations

Year	Size	Cycle Time
1980	64 Kb	250 ns
1983	256 Kb	220 ns
1986	1 Mb	190 ns
1989	4 Mb	165 ns
1992	16 Mb	120 ns
1996	64 Mb	110 ns
1998	128 Mb	100 ns
2000	256 Mb	90 ns
2002	512 Mb	80 ns
2006	1024 Mb	60ns
	16000:1 (Capacity)	4:1 (Latency)

Processor-DRAM Performance Gap Impact: Example

- To illustrate the performance impact, assume a single-issue pipelined CPU with CPI = 1 using non-ideal memory.
- The minimum cost of a full memory access in terms of number of wasted CPU cycles:

Year	CPU speed MHZ	CPU cycle ns	Memory Access ns	Minimum CPU cycles or instructions wasted
1986:	8	125	190	190/125 - 1 = 0.5
1989:	33	30	165	165/30 -1 = 4.5
1992:	60	16.6	120	120/16.6 -1 = 6.2
1996:	200	5	110	110/5 -1 = 21
1998:	300	3.33	100	100/3.33 -1 = 29
2000:	1000	1	90	90/1 - 1 = 89
2003:	2000	.5	80	80/.5 - 1 = 159
2006:	3700	0.27	60	60/.27 – 1 = 221

How to make memory system better?

- Programmers want unlimited amounts of memory with low latency
- Fast memory technology is more expensive per bit than slower memory
- Solution: organize memory system into a hierarchy
  - Entire addressable memory space available in largest, slowest memory
  - Incrementally smaller and faster memories, each containing a subset of the memory below it, proceed in steps up toward the processor
- Temporal and spatial locality insures that nearly all references can be found in smaller memories
  - Gives the illusion of a large, fast memory being presented to the processor

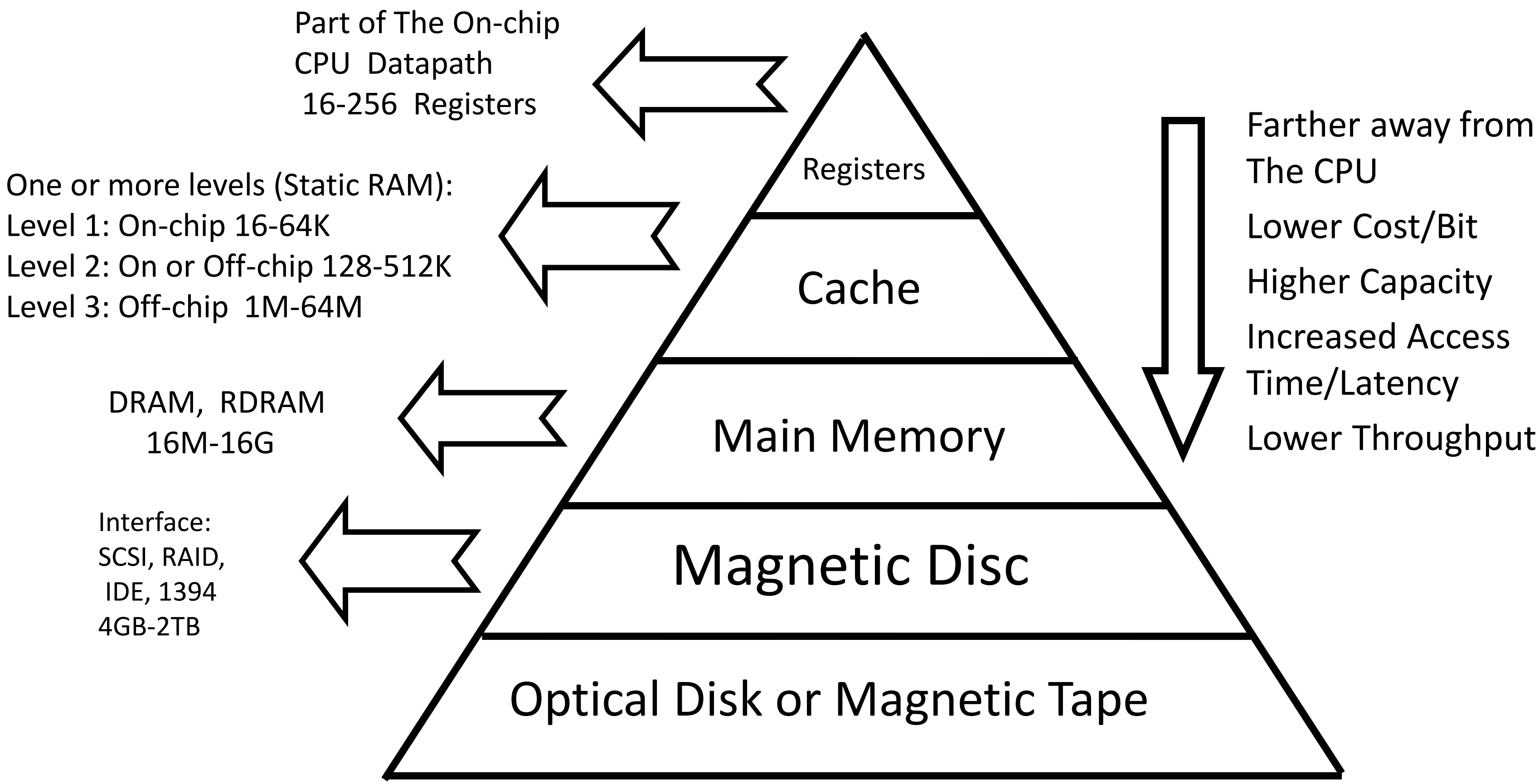
Memory Hierarchy Design

- Memory hierarchy design becomes more crucial with recent multi-core processors:
  - Aggregate peak bandwidth grows with # cores:
    - Intel Core i7 can generate two references per core per clock
    - Four cores and 3.2 GHz clock
      - 25.6 billion 64-bit data references/second +
      - 12.8 billion 128-bit instruction references
      - = 409.6 GB/s!
    - DRAM bandwidth is only 6% of this (25 GB/s)
    - Requires:
      - Multi-port, pipelined caches
      - Two levels of cache per core
      - Shared third-level cache on chip
- High-end microprocessors have >10 MB on-chip cache
  - Consumes large amount of area and power budget

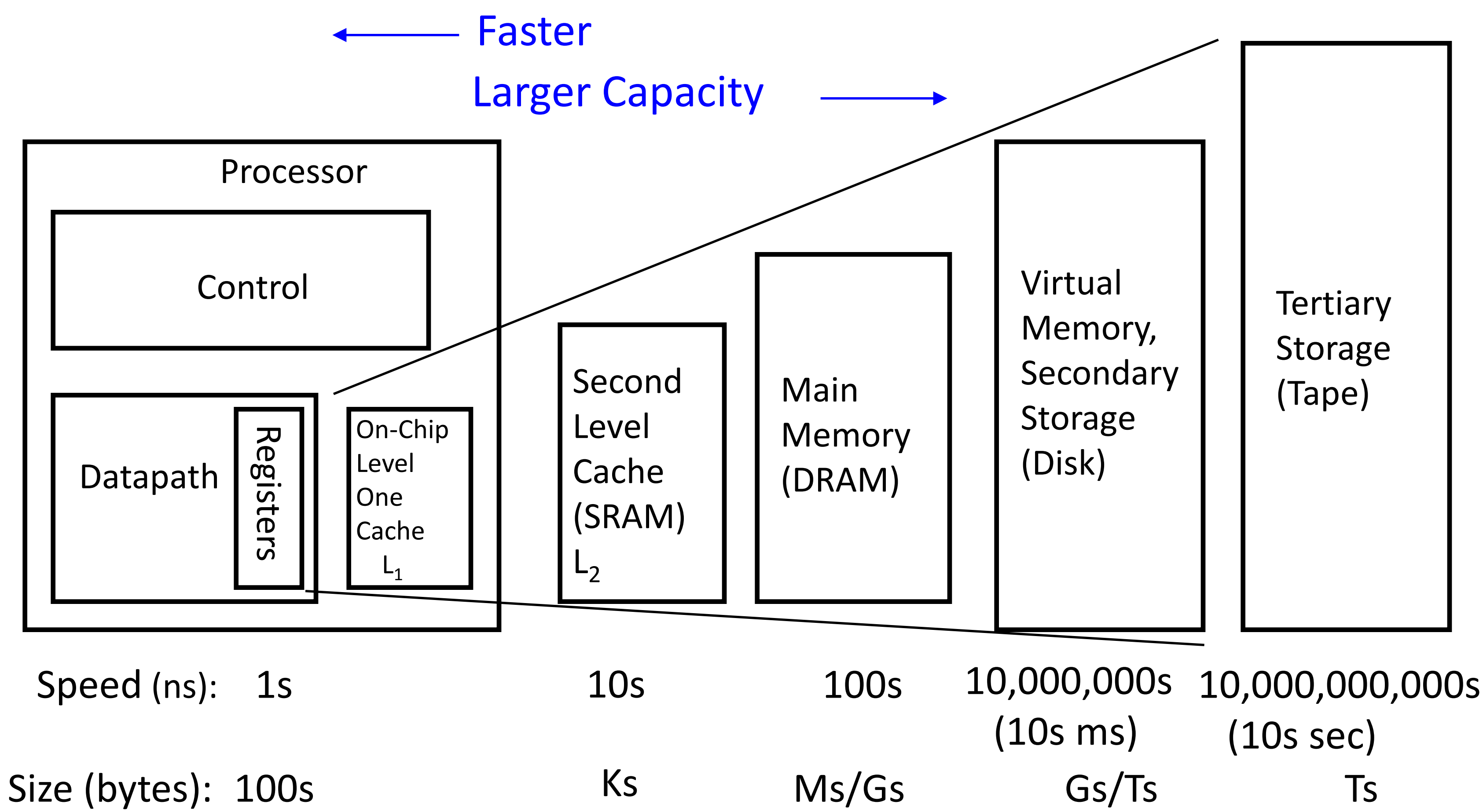
Memory Hierarchy

- The idea is to build a memory subsystem that consists of:
  - Very small, very fast, very expensive memory “close” to the processor.
  - Larger, slower, but more affordable memory “further away” from the processor.
  - Hence, provide the appearance of virtually unlimited memory while minimizing delays to the processor.
- The memory hierarchy is organized into levels of memory with the smaller, more expensive, and faster memory levels closer to the CPU: registers, then primary Cache Level (L_1), then additional secondary cache levels (L_2, L_3…), then main memory, then mass storage (virtual memory).

Levels of The Memory Hierarchy



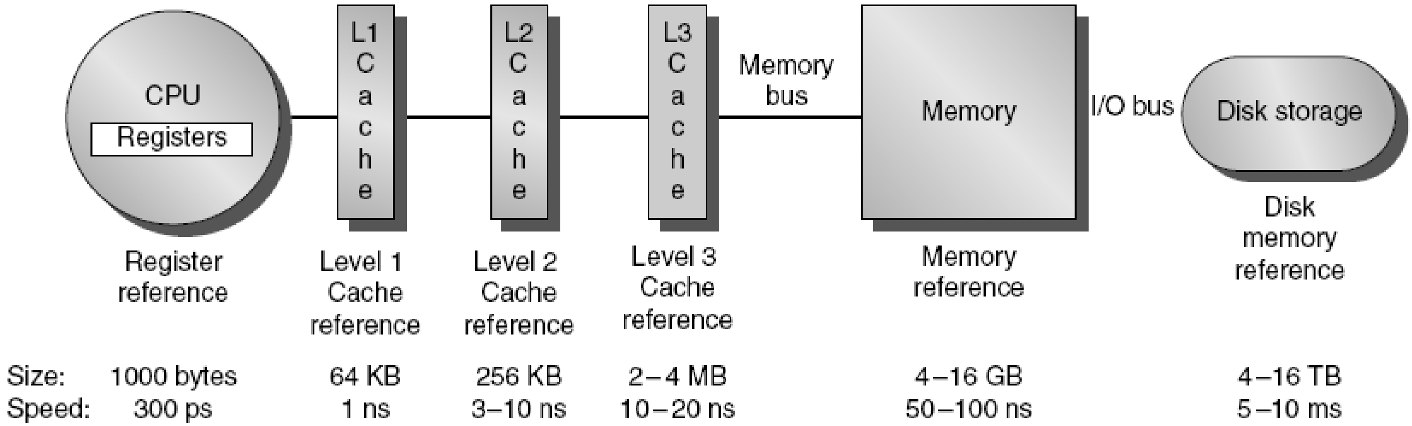
A Typical Memory Hierarchy
(With Two Levels of Cache)



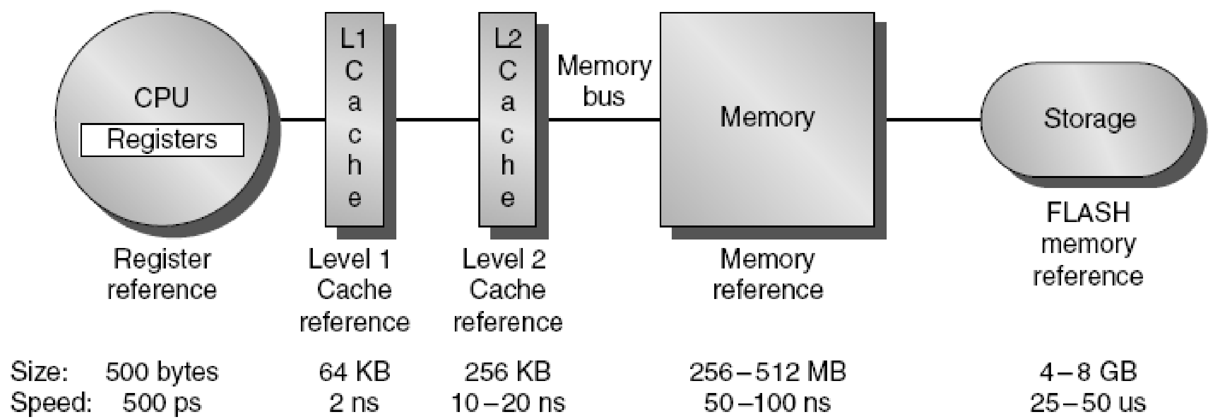
Levels of The Memory Hierarchy

Level	**1**	**2**	**3**	**4**
Called	Registers	Cache	Main memory	Disk storage
Typical size	< 1 KB	< 4 MB	<4 GB	> 1 GB
Implementation technology	Custom memory with multiple ports, CMOS or BiCMOS	On-chip or off-chip CMOS SRAM	CMOS DRAM	Magnetic disk
Access time (in ns)	2–5	3–10	80–400	5,000,000
Bandwidth (in MB/sec)	4000–32,000	800–5000	400–2000	4–32
Managed by	Compiler	Hardware	Operating system	Operating system/user
Backed by	Cache	Main memory	Disk	Tape

Recent Typical Configurations



(a) Memory hierarchy for server



(b) Memory hierarchy for a personal mobile device

Memory Hierarchy: Apple iMac G5

Managed by compiler

Managed by hardware

Managed by OS, hardware, application

07	Reg	L1 Inst	L1 Data	L2	DRAM	Disk
Size	**1K**	**64K**	**32K**	**512K**	**256M**	**80G**
Latency Cycles, Time	**1, 0.6 ns**	**3, 1.9 ns**	**3, 1.9 ns**	**11, 6.9 ns**	**88, 55 ns**	**10^7, 12 ms**

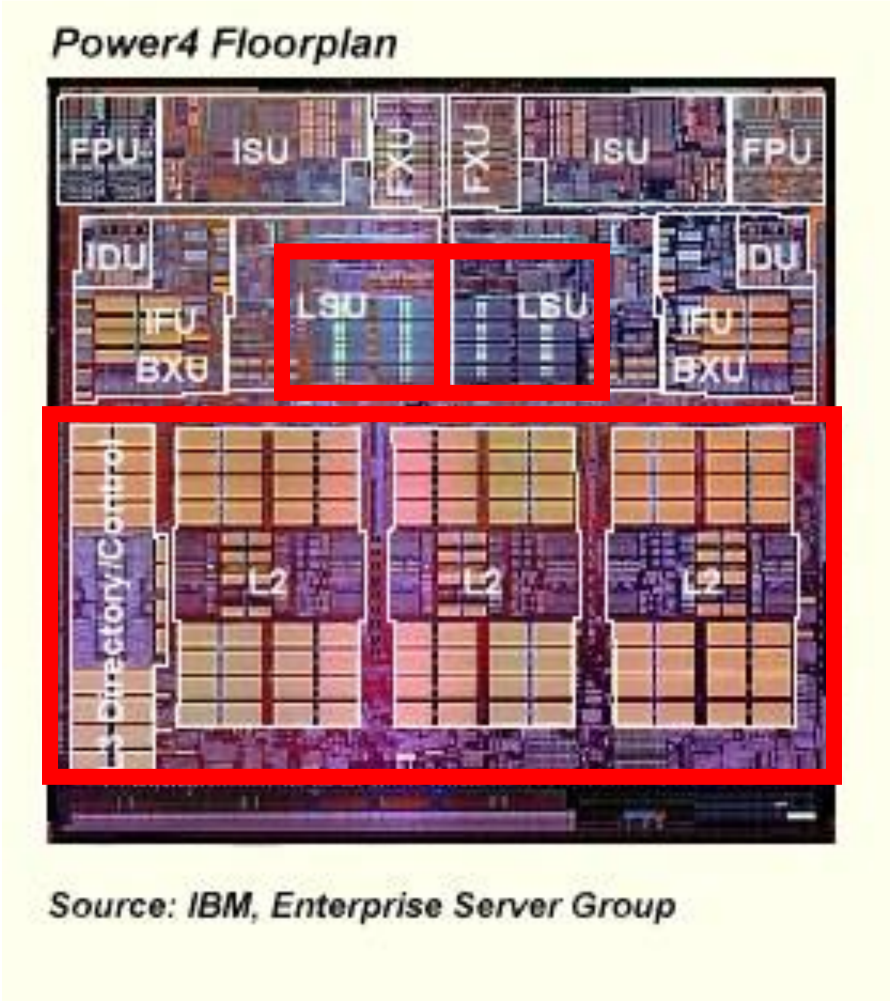
Goal: Illusion of large, fast, cheap memory

Let programs address a memory space that scales to the disk size, at a speed that is usually as fast as register access

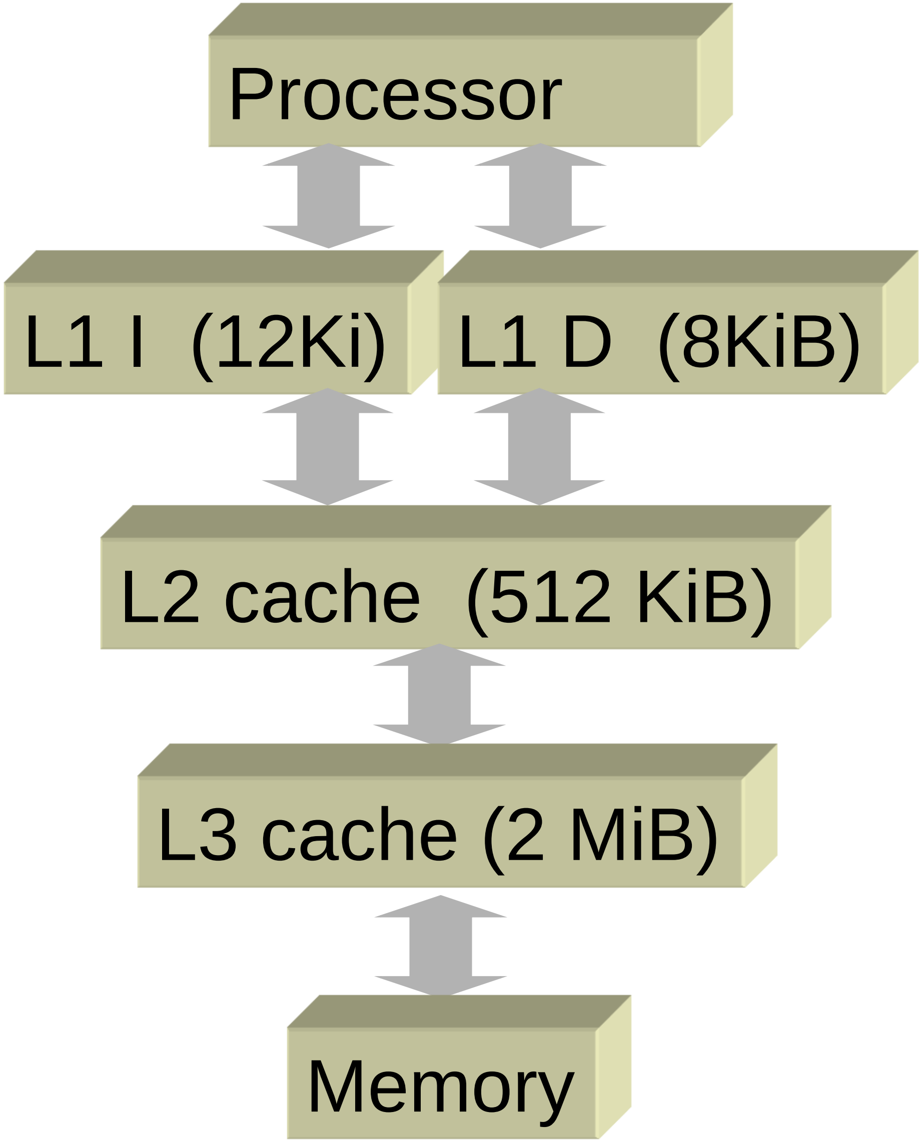
Memory on the CPU Chip

Architects Use Transistors to Tolerate Slow Memory

- Cache
  - Small, Fast Memory
  - Holds information (expected) to be used soon
  - Mostly Successful
- Apply Recursively
  - Level-one cache(s)
  - Level-two cache
- Most of microprocessor die area is cache!



Pentium 4 Cache hierarchy (Gallatin)



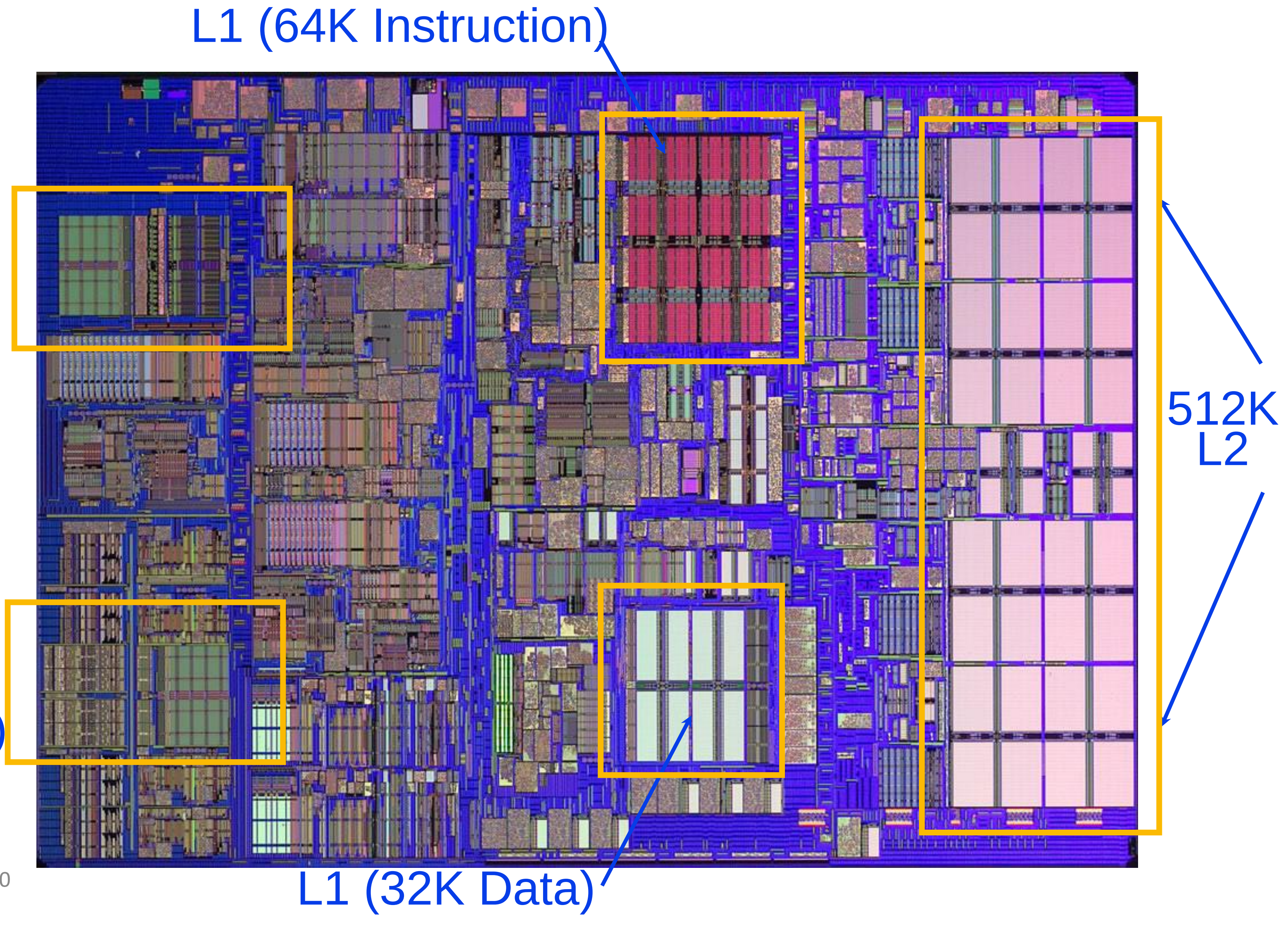
Cycles: 2

Cycles: 19

Cycles: 43

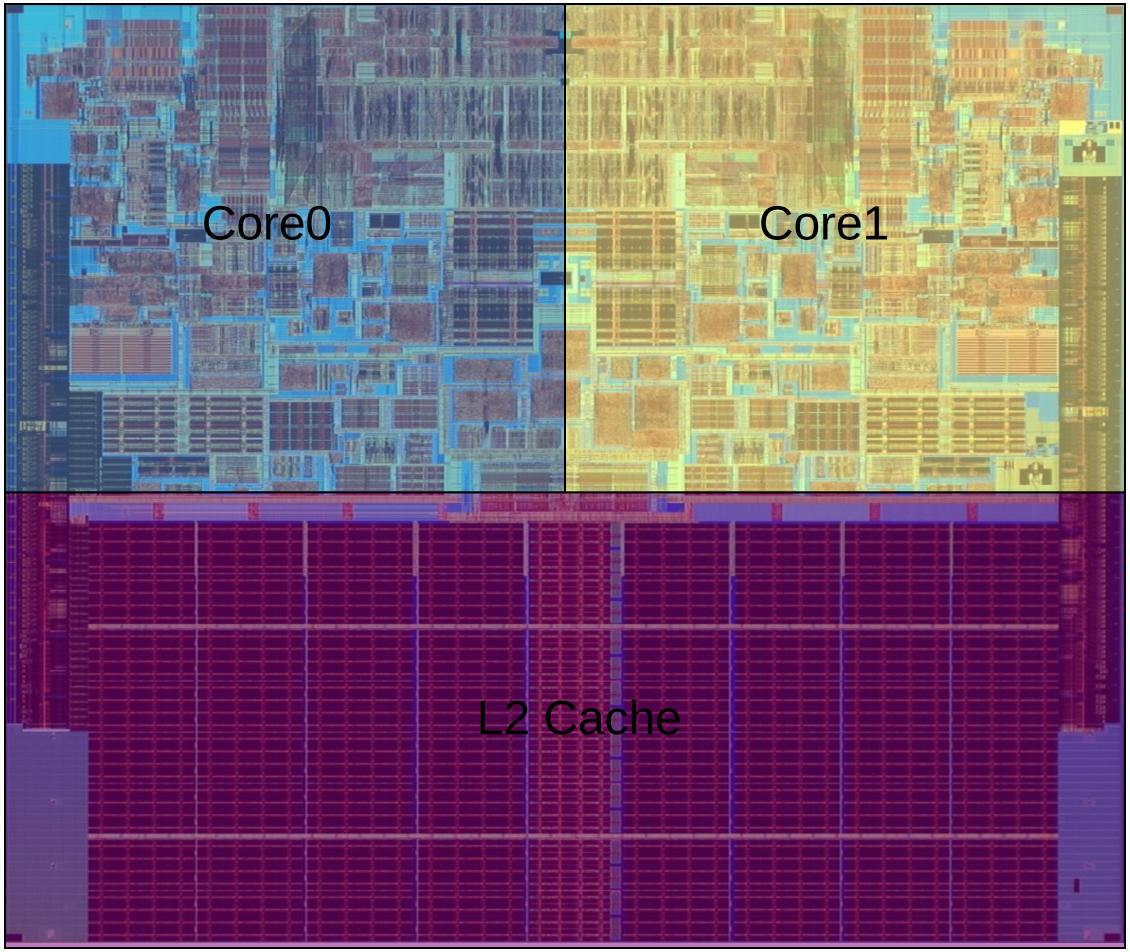
Cycles: 206

iMac's PowerPC 970: All caches on-chip



Case study: Intel Core2 Duo

L1	32 KB, 8-Way, 64 Byte/Line, LRU, WB 3 Cycle Latency
L2	4.0 MB, 16-Way, 64 Byte/Line, LRU, WB 14 Cycle Latency



Source: http://www.sandpile.org

Memory Hierarchy Basics

- When a word is not found in the higher level, a *miss* occurs:
  - Fetch word from lower level in hierarchy, requiring a higher latency reference
  - Also fetch the other words contained within the *block*
    - Takes advantage of spatial locality
  - Place block into cache in any location within its *set*, determined by address
    - block address MOD number of sets

Memory Hierarchy Operation

If an instruction or operand is required by the CPU, the levels of the memory hierarchy are searched for the item starting with the level closest to the CPU (Level 1 cache):

- **If the item is found, it's delivered to the CPU resulting in a cache hit.**
- **If the item is missing from an upper level, resulting in a miss, the level just below is searched.**
- **For systems with several levels of cache, the search continues with cache level 2, 3 etc.**
- **If all levels of cache report a miss then main memory is accessed.**
  - CPU ↔ cache ↔ memory: Managed by hardware.
- **If the item is not found in main memory resulting in a page fault, then disk (virtual memory), is accessed for the item.**
  - Memory ↔ disk: Managed by hardware and the operating system.

Memory Hierarchy Basics

- *n* sets => *n-way set associative*
  - *Direct-mapped cache =>* one block per set
  - *Fully associative* => one set

- Writing to cache: two strategies
  - *Write-through*
    - Immediately update lower levels of hierarchy
  - *Write-back*
    - Only update lower levels of hierarchy when an updated block is replaced
  - Both strategies use *write buffer* to make writes asynchronous

Memory Hierarchy Basics

- Miss rate
  - Fraction of cache access that result in a miss

- Causes of misses
  - Compulsory
    - First reference to a block
  - Capacity
    - Blocks discarded and later retrieved
  - Conflict
    - Program makes repeated references to multiple addresses from different blocks that map to the same location in the cache

Memory Hierarchy Basics

$$\frac{\text{Misses}}{\text{Instruction}} = \frac{\text{Miss rate} \times \text{Memory accesses}}{\text{Instruction count}} = \text{Miss rate} \times \frac{\text{Memory accesses}}{\text{Instruction}}$$

$$\text{Average memory access time} = \text{Hit time} + \text{Miss rate} \times \text{Miss penalty}$$

- Note that speculative and multithreaded processors may execute other instructions during a miss
  - Reduces performance impact of misses

Impact on Performance

- Suppose a processor executes at
  - Clock Rate = 200 MHz (5 ns per cycle)
  - CPI = 1.1
  - 50% arith/logic, 30% ld/st, 20% control
- Suppose that 10% of memory operations get 50 cycle miss penalty
- CPI = ideal CPI + average stalls per instruction

  = 1.1(cyc) +(0.30 (datamops/ins)

  x 0.10 (miss/datamop) x 50 (cycle/miss))

  = 1.1 cycle + 1.5 cycle = 2.6
- 58 % of the time the processor is stalled waiting for memory!
- a 1% instruction miss rate would add an additional 0.5 cycles to the CPI!