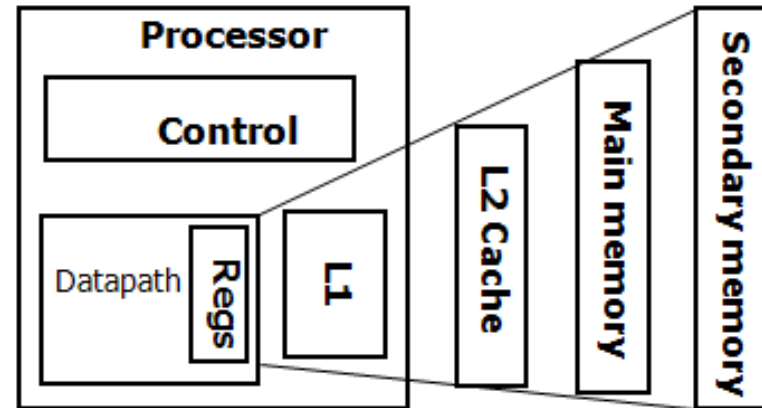


Lecture 6: Memory hierarchy and Computer performance

Topics

- Computer performance
 - Response time
 - Throughput
- Memory hierarchy
 - ... Caching, and virtual memory
- Amdahl's law (lab 8)
- Benchmarks – why we need them
- Comparing different CPU brands
 - Intel's Processor Number
 - AMD's Model Number



Technology Trends – Early Days

	Capacity	Speed (latency)
Processor	--	4 x in 3 yrs
DRAM	4 x in 3 yrs	2 x in 10 yrs
Disk	4 x in 3 yrs	2 x in 10 yrs

DRAM

Year	Size		Cycle Time	
1980	64Kb	1000:1	250 ns	2:1
1983	256Kb		220 ns	
1986	1 Mb		190 ns	
1989	4 Mb		165 ns	
1993	16 Mb		145 ns	
1997	64 Mb		120 ns	
... ..				

Technology Trends – Recent

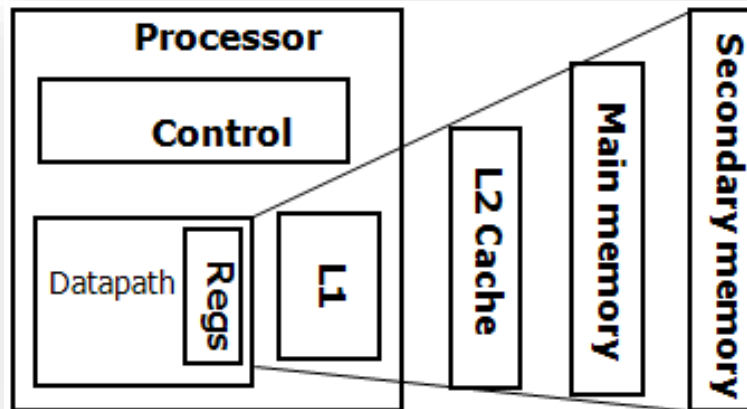
- What changes can we expect in the next 3-6-10 years cycle?

DRAM	Year	Size		Cycle Time	
	1997	64Mb	20:1	120 ns	20:1
	2003+	Over 1000 Mb		below 6 ns	
				
	2010+	Over 4096 Mb		below 2.5ns	
	2015+	>8 Gb	⋮	<1 ns	⋮
2020	500-750 Gb?			

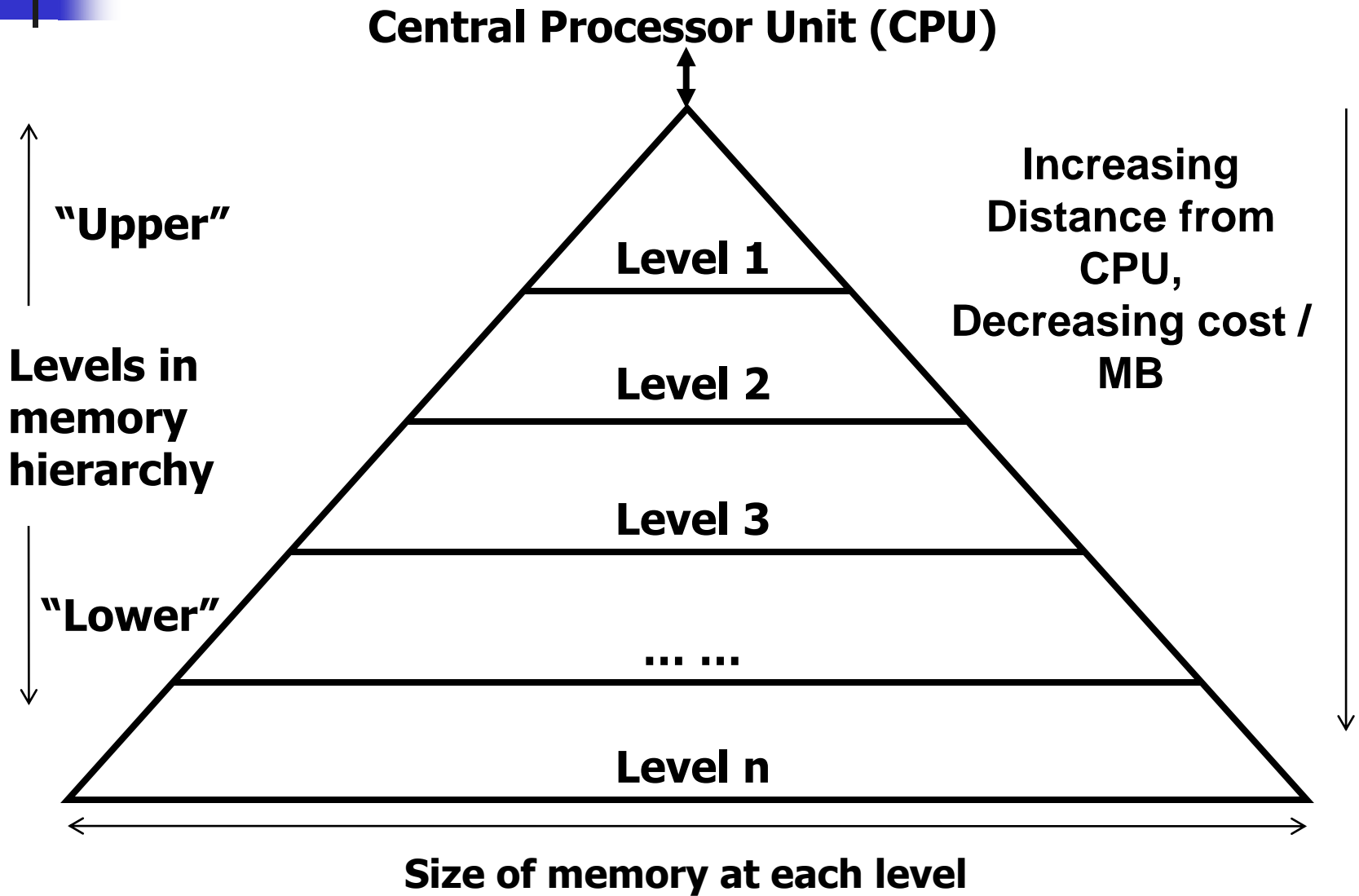
- DDR3** introduced way back in 2007 and are common now
- Current **DDR4**; PC industry is preparing transition to **DDR5**
 - DDR4 promises to run half as hot and twice as fast
- New memory technology: The new **3D XPoint** memory technology will change the way we think about storage (>10x denser than DRAM; >1,000x faster than SSD; search “Intel 3D XPoint” on the Internet).

Illusion of Large, Fast, Cheap Memory

- Fact remains: fast memories are small, large memories are slow
- Expectation: create a memory that is large, cheap and fast (most of the time)
- Strategy: Hierarchy of memory levels



Memory Hierarchy: The Pyramid





Memory Hierarchy: Library Analogy

- Working on paper in library at a desk

Option 1: Every time need a book	Option 2: Every time need a book
<ul style="list-style-type: none">▪ Leave desk to go to shelves (or stacks)▪ Find the book▪ Bring one book back to desk▪ Read section interested in▪ When done with section, leave desk and go to shelves carrying book▪ Put the book back on shelf▪ Return to desk to work▪ Next time need a book, go to first step	<ul style="list-style-type: none">▪ Leave some books on desk after fetching them (Recently read ones)▪ Only go to shelves when need a new book▪ When go to shelves, bring back nearby related books in case you need them; sometimes you'll need to return books not used recently to make space for new books on desk▪ Return to desk to work▪ When done, return books to shelves, carrying as many as you can per trip



Memory Hierarchy: Working Principles

- **The Principle of Locality:**
 - Program access a relatively small portion of the address space at any instant of time.
- **Temporal Locality (Locality in Time):**
 - Items accessed recently are likely to be accessed again soon.
 - Keep most recently accessed data items closer to the processor
- **Spatial Locality (Locality in Space):**
 - Items near those accessed recently are likely to be accessed soon.
 - Move blocks consisting of contiguous words to the upper levels
- **Programming constructs lead to Principle of Locality**
 - data: loop counters (temporal), arrays, stacks (spatial)
 - code: instructions in a loop (temporal), sequential instructions (spatial)



Memory Hierarchy: Big Idea

- Temporal locality
 - keep recently accessed data items closer to processor
- Spatial locality
 - moving contiguous words in memory to upper levels of hierarchy
- Smaller and faster memory technologies close to the processor
 - Cheap, slow memory furthest from processor
 - Fast hit time in highest level of hierarchy

If hit rate is high enough, hierarchy strategy offers access time close to the highest (and fastest) level and size equal to the largest level with lowest cost.

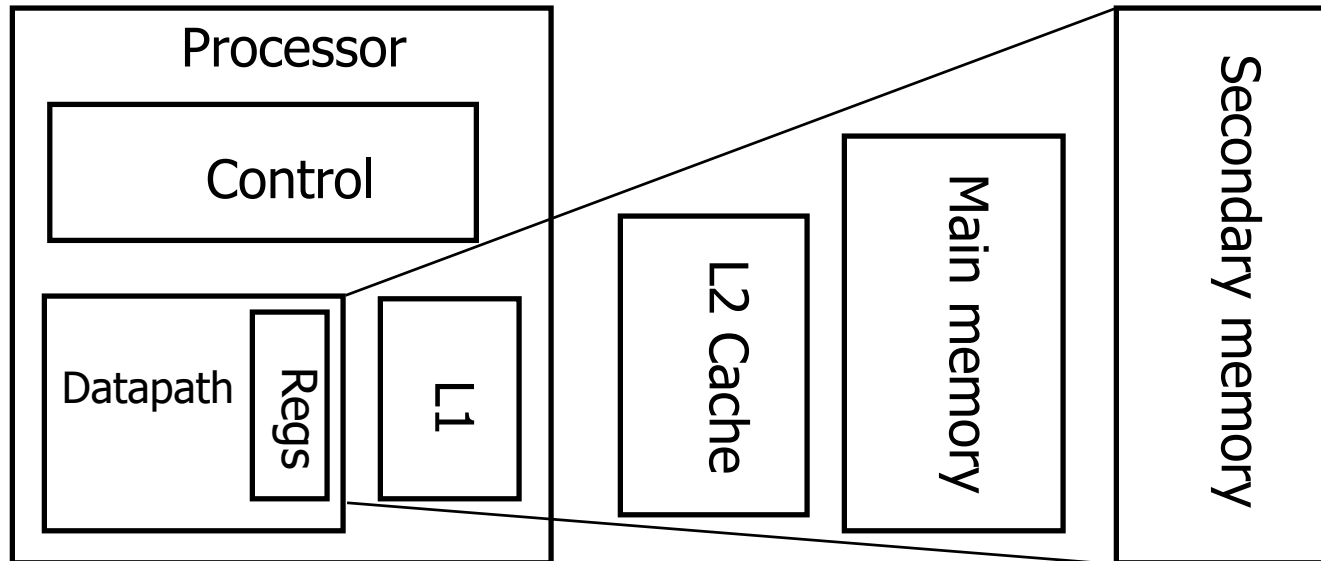
However, details of multiple levels were hidden from users.



Memory Hierarchy: Terminology

- Hit: data appears in some block in the upper level
 - Hit Rate: the fraction of memory access found in the upper level
 - Hit Time: Time to access the upper level which consists of
 - Time to determine hit/miss + Memory access time
 - Analogy: time to find, pick up book from desk
- Miss: data needs to be retrieved from a block in the lower level
 - Miss Rate = $1 - (\text{Hit Rate})$
 - Miss Penalty: Time to replace a block in the upper level + Time to deliver the block the processor
 - Analogy: time to go to shelves, find needed book, and return it to your desk, pick up
- Note: Hit Time \ll Miss Penalty

Memory Hierarchy: Sample



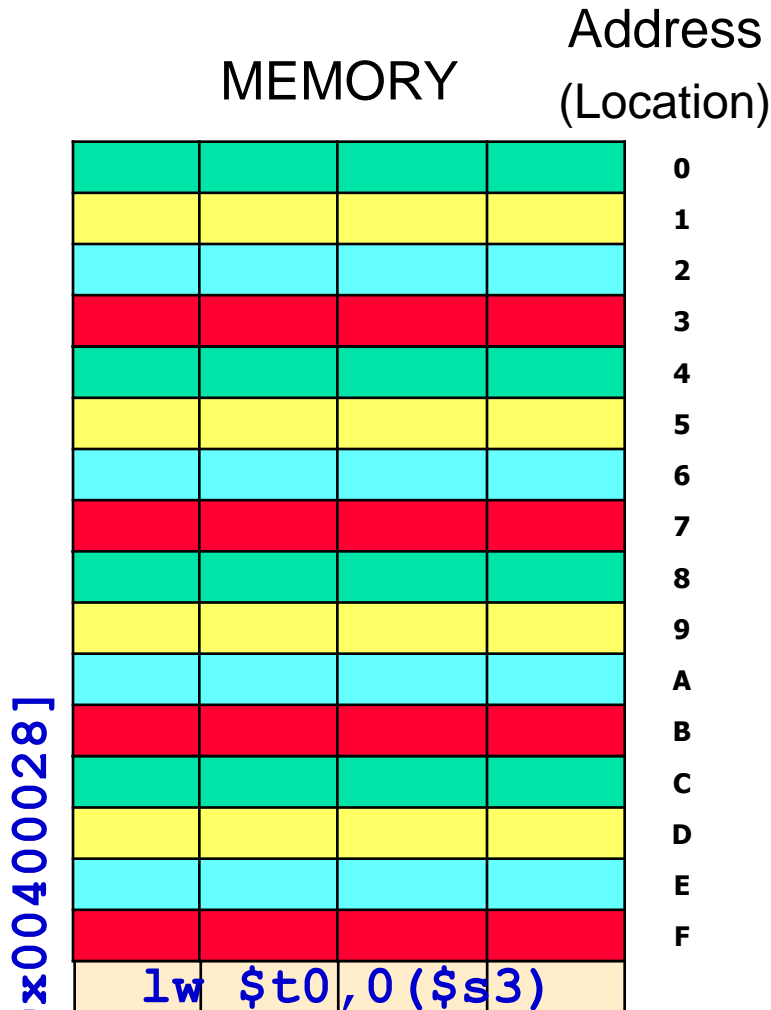
Speed(ns):	0.5ns	2ns	6ns	100ns	10,000,000ns
Size (MB):	0.0005	0.05	1-4	100-1000	100,000
Cost (\$/MB):	--	\$100	\$30	\$1	\$0.05
Technology:	Regs	SRAM	SRAM	DRAM	Disk



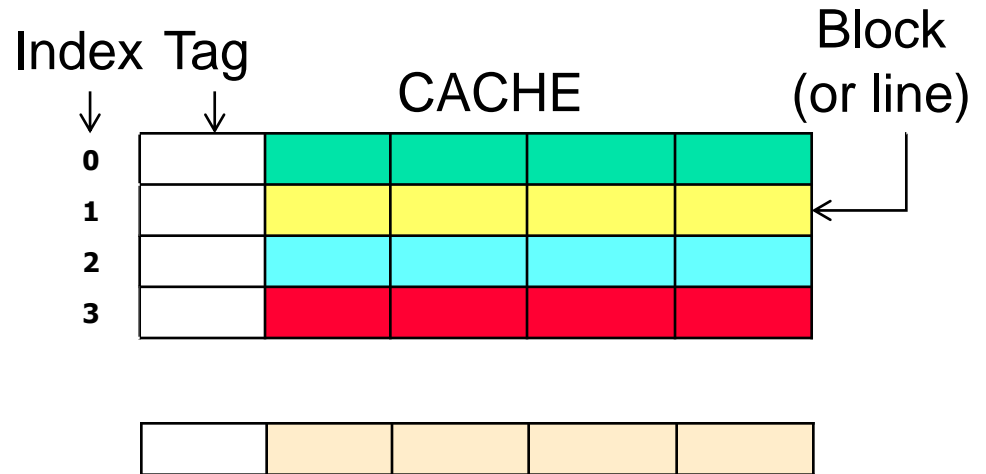
Memory Hierarchy: Management

- Registers \leftrightarrow Memory
 - By compiler (or Assembly Language Programmer)
- Cache \leftrightarrow Main Memory
 - By the hardware (invisible to compiler/programmer)
- Main Memory \leftrightarrow Disks
 - By the hardware and operating system (virtual memory)
 - By the programmer (Files)
- Cache: 1st Level of Memory Hierarchy
 - Cache consists of blocks, which have: a validity bit, tag, data
 - Memory address split into: Tag, Index and Offset (more later)
 - Direct Mapped Cache: each memory location has one only location in cache.
 - How do you know if something is in the cache?

Simplest Cache: Direct Mapped – p. 1 of 4



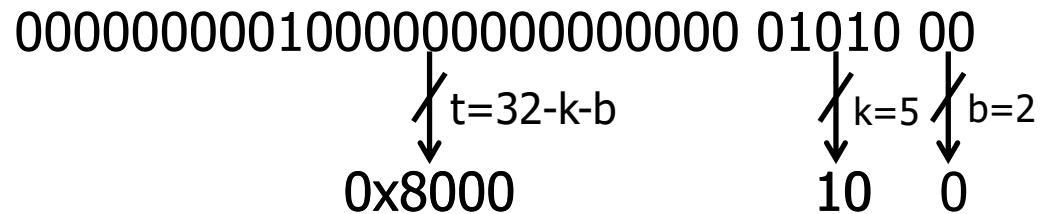
[0x00400028]



This CACHE has 4 blocks, each has 4 bytes

Typically CACHE has 2^k blocks, each has 2^b bytes/words. (32 blocks = 2^5 ; 4 bytes = 2^2)

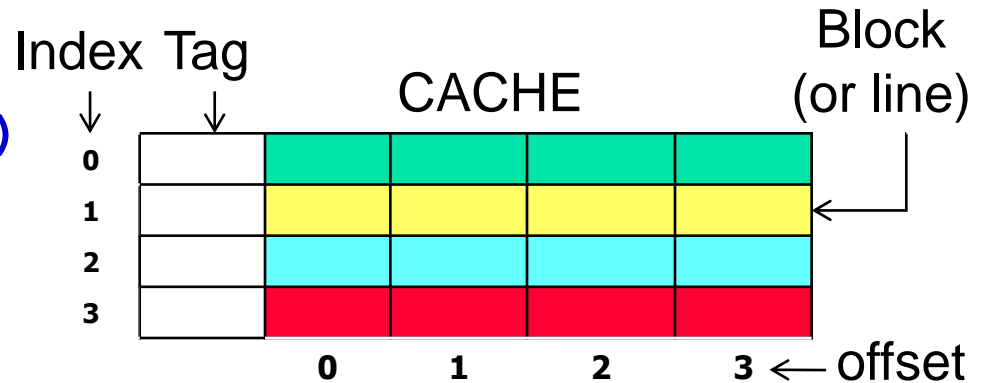
MEMORY Location -> CACHE location?



Direct Mapped Cache: Structure – p. 2 of 4

`[0x00400028] lw $t0, 0($s3)`

256 blocks = 2^8
 4 bytes = 2^2
 1 word = 2^0

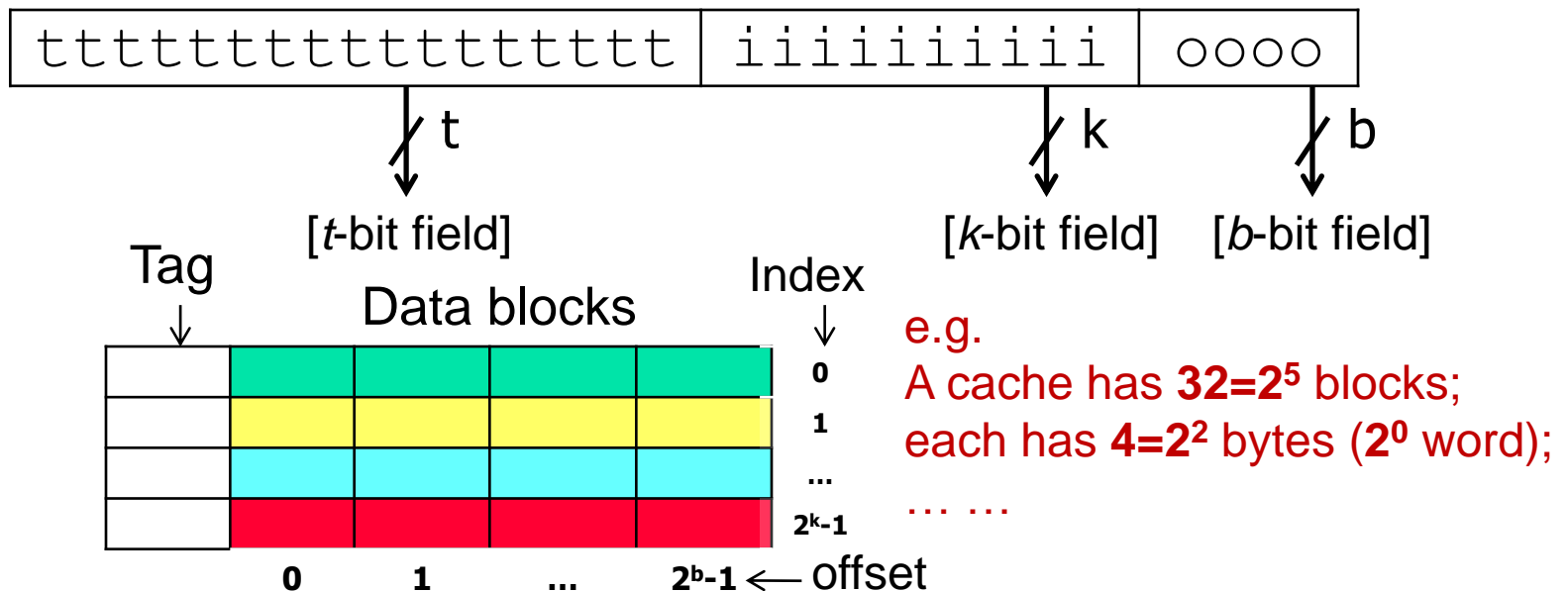


$k = \log_2(\text{No. of Blocks})$; $b = \log_2(\text{No. of Bytes or Words per Block})$

- Assume: CACHE has 2^k blocks, each has 2^b bytes or words
 - 2^k is the size of CACHE in blocks (indexed from 0 to 2^k-1)
 - 2^b is the size of block in bytes (a particular byte has an offset from 0 to 2^b-1) or words
- MEMORY Location (address) -> CACHE location (index)
 - **Question:** Given a memory address (32 bits), work out if the corresponding contents have been in CACHE?
 - **Method:** Decode MEMORY address; extract and parse certain fields to learn index, offset, and tag information etc.

Direct Mapped Cache: Addressing – p. 3 of 4

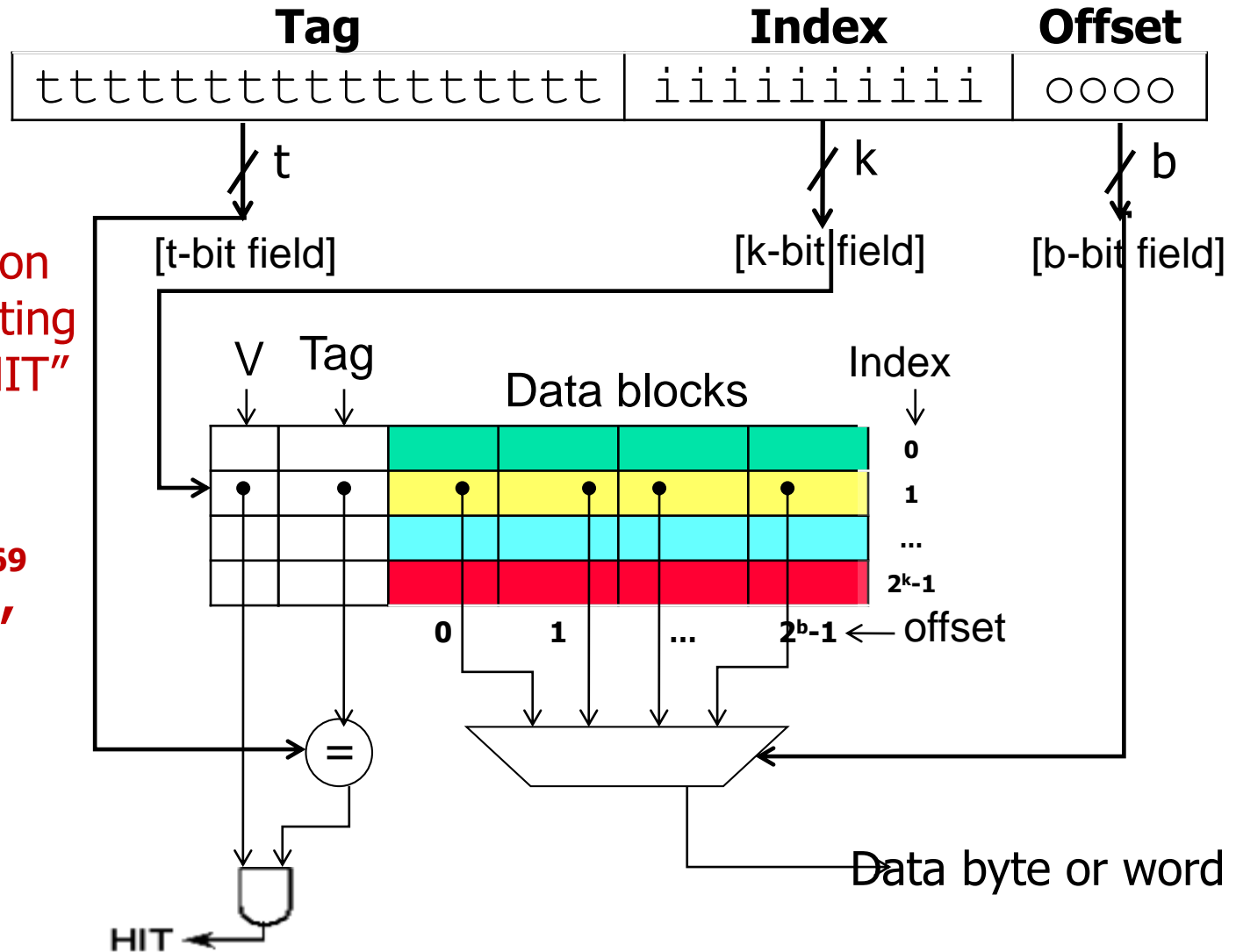
- Divide Memory Address into 3 portions (fields), and parse for the following information:
 - index: where in the cache to look (which block);
 - offset: which byte (or word) in block is start of the desired data;
 - tag: if the data in the cache corresponds to what in the memory address being looked for.



Direct Mapped Cache: Mapping – p. 4 of 4

What condition is for generating the signal "HIT" ?

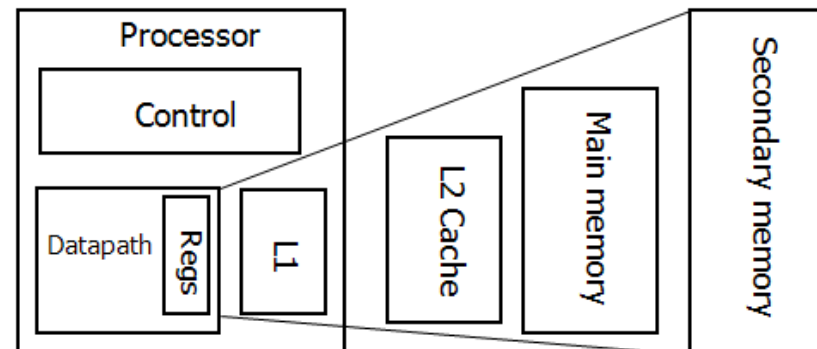
PH4, P₄₆₂₋₄₆₉
PH5, Fig5.10,
P₃₈₉
PH6, Fig5.10



Virtual Memory

OPTIONAL

- If Principle of Locality allows caches to offer (usually) speed of cache memory with size of DRAM memory, then recursively why not use at next level to give speed of DRAM memory, size of Disk memory?
- Called “Virtual Memory” (historically, it predates caches)
 - Also allows OS to share memory, protect programs from each other
 - Today, more important for protection as opposed to just another level of memory hierarchy



Protection Leads to New Terms

OPTIONAL

- Each computer has “**physical address space**” (e.g., 4 GigaBytes DRAM); also called “real memory”
- Each program is said to have “**virtual address space**” (e.g., 32 MegaBytes)
- Address translation: map page from virtual address to physical address
 - Allows multiple programs to use (different pages of physical) memory at same time
 - Can only access what OS permits
 - Also allows some pages of virtual memory to be represented on disk, not in main memory (memory hierarchy)



Comparing the 2 Levels of Hierarchy

OPTIONAL

- Cache Version
 - Block or Line
 - Miss
 - Block Size: 32-64B
 - Placement:
Direct Mapped, N-way Set Associative
- Virtual Memory version
 - Page
 - Page Fault
 - Page Size: 4K-8KB
 - Fully Associative
(a virtual page can be placed in any page of real memory)

Different terminology, but the same principle.

Performance: Airplane Analogy

Plane	Range (km)	Top Speed (km/h)	Passengers	Throughput (pass. x km/h)
Boeing 747	6640	976	470	458,720
BAD/Sud Concorde	6400	2160	132	285,120

- Which has higher “performance”? What’s performance:
 - Time to deliver 100 passengers, or:
 - Time to deliver 400 passengers?
- Option 1: **things-per-second**
 - bigger is better
- Option 2: **time-per-job**
 - smaller is better



Performance: Common metrics

- Two common performance metrics:
 - **Response time** (Execution Time, Latency): time for 1 job
 - shorter time = better performance
 - Flying Time: Concorde versus Boeing 747?
Concorde is $2160 / 976 = 2.2$ times “better performing”
 - **Throughput** (I/O bandwidth): jobs per unit time
 - more jobs completed = better performance
 - Throughput: Boeing versus Concorde?
Boeing 747 is: $458,720 / 285,120 = 1.6$ times “better performing”

Response time and throughput depend on each other: if one is affected, the other one is likely to be affected as well

- We will focus primarily on **execution time** for a single job
 - “improve execution time” = improve performance
 - We will stick to “**x times faster**” for comparison

Amdahl's Law (lab 8)

- Observation: The overall **Speedup** depends on the fraction of the time **affected** by the enhancement

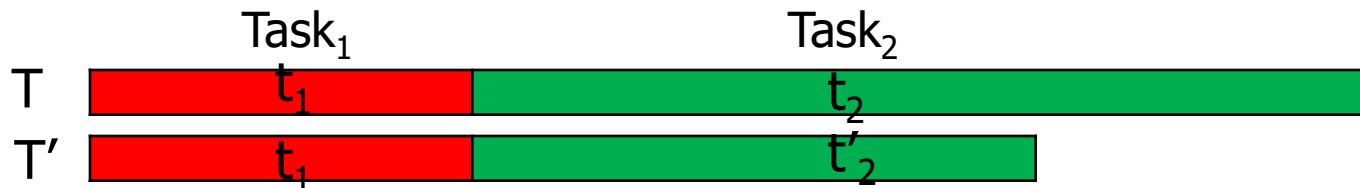
- **Speedup** = $\text{Time}_{\text{before enhancement}} \mathbf{T} / \text{Time}_{\text{after enhancement}} \mathbf{T}'$ (1)

Indicating how many times faster overall

- $T = t_1 + t_2$ (2)

- $T' = t_1 + t'_2$ (3)

- For easy calculation, let $\mathbf{a} = t_2/T$; $\mathbf{N} = t_2/t'_2$ (**N times faster on Task₂**)



- Consider the following scenario

- A program 'DEMO' runs for 100 sec, 80% being multiply operations
- The aim is to make the program 'DEMO' 4 times faster (overall Speedup)
- Question: how to achieve the above Speedup by improving only the speed of multiplication (calculate N)?

Benchmarking methods



- Run typical **programs** with typical **input** to estimate the performance. E.g.
 - Engineers: uses compiler, database, spreadsheet
 - Writers: uses word processor, drawing program, compression software
 - Game players: ...

Next we will look at old examples, followed by some benchmarking methods.

All slides are used for illustrative purpose only, for current data you need to check the appropriate Web sites!



Standardised Workload Benchmarks

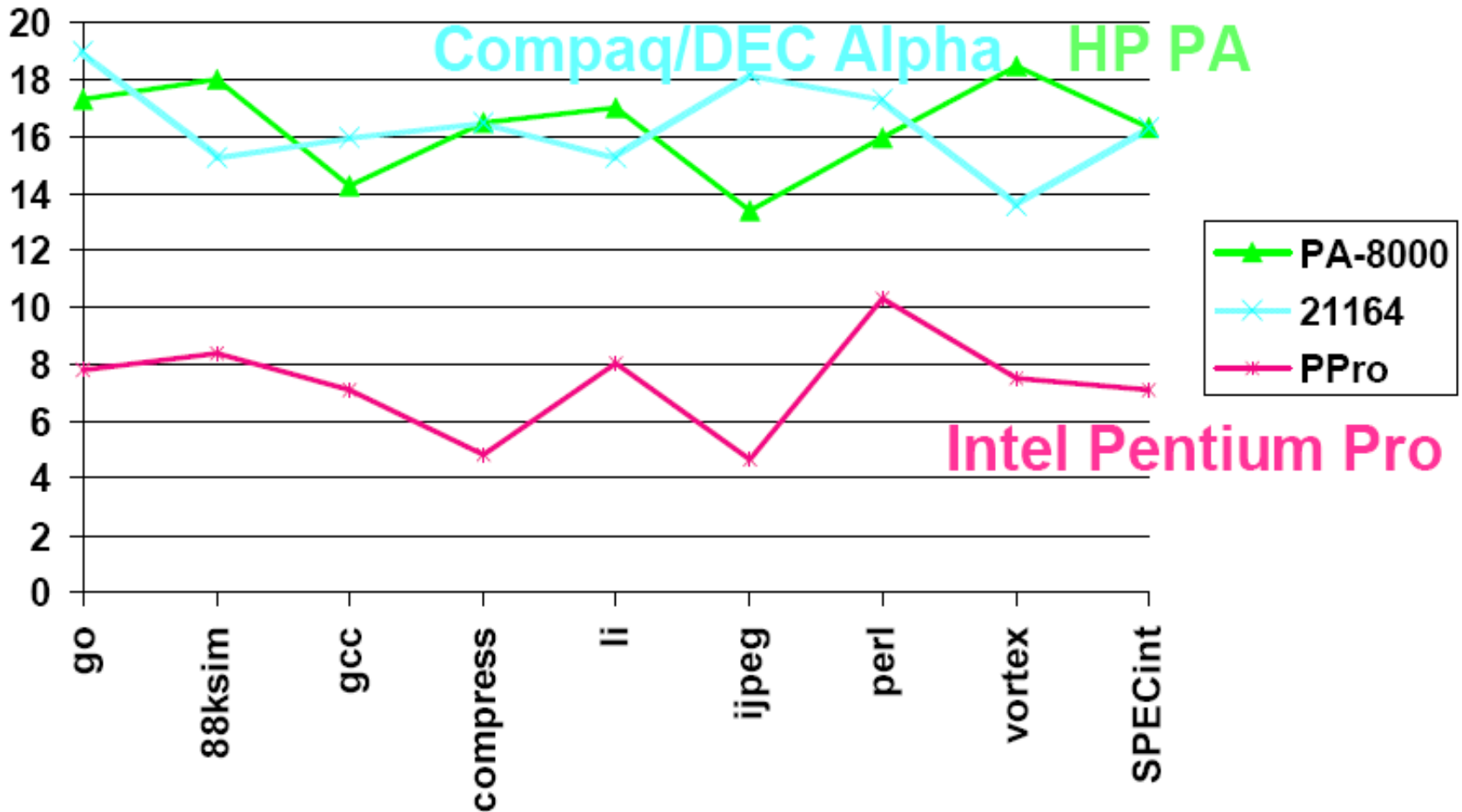
- SPEC: Standard Performance Evaluation Corporation
see: www.spec.org
 - SPEC is a non-profit corporation formed to "establish, maintain and endorse a standardized set of relevant benchmarks that can be applied to the newest generation of high-performance computers"
 - does not make recommendations on computer purchases
 - offers an objective series of applications-oriented tests, which can serve as common reference points and be considered during the evaluation process.
- While no one benchmark can fully characterize overall system performance, the results of a variety of realistic benchmarks can give valuable insight into expected real performance

SPEC CPU200x | CPU2017

- Better benchmarks are based on applications, and these applications can come from any areas of work. For example SPEC CPU200x (2004, 2005, 2006) suite includes applications from the following areas:
 - AI game theory
 - Compilers
 - Interpreters
 - data compression
 - Databases
 - weather prediction
 - fluid dynamics
 - Physics
 - chemistry
 - image processing
- When new SPEC developed, decisions: which applications include and in what proportion?

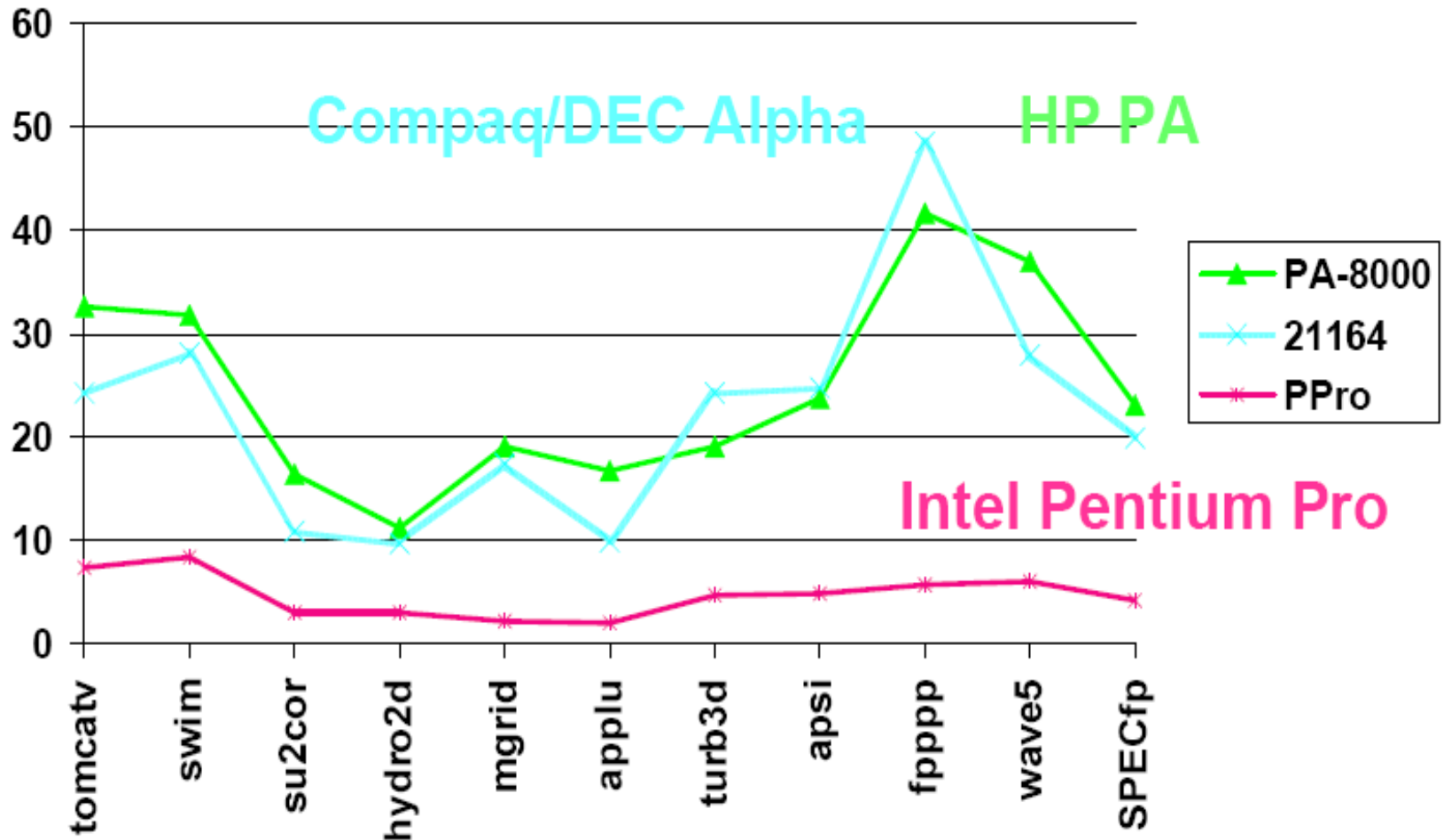
SPEC_{int95base} Performance (1997)

Which CPU has higher “performance” (integer applications benchmark)?



SPEC_{fp95base} Performance (1997)

Which CPU has higher “performance” (floating point applications benchmark)?

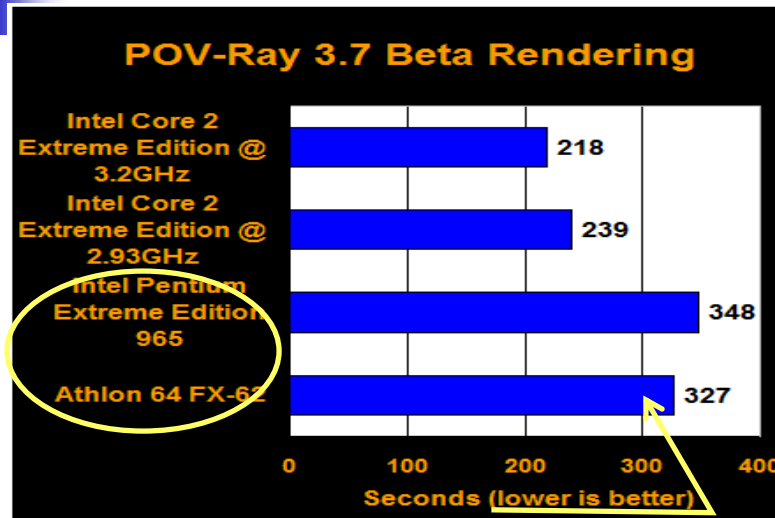


Business Winstone 2001 Benchmark Results

- Which CPU has higher “performance” – compare clock speed?

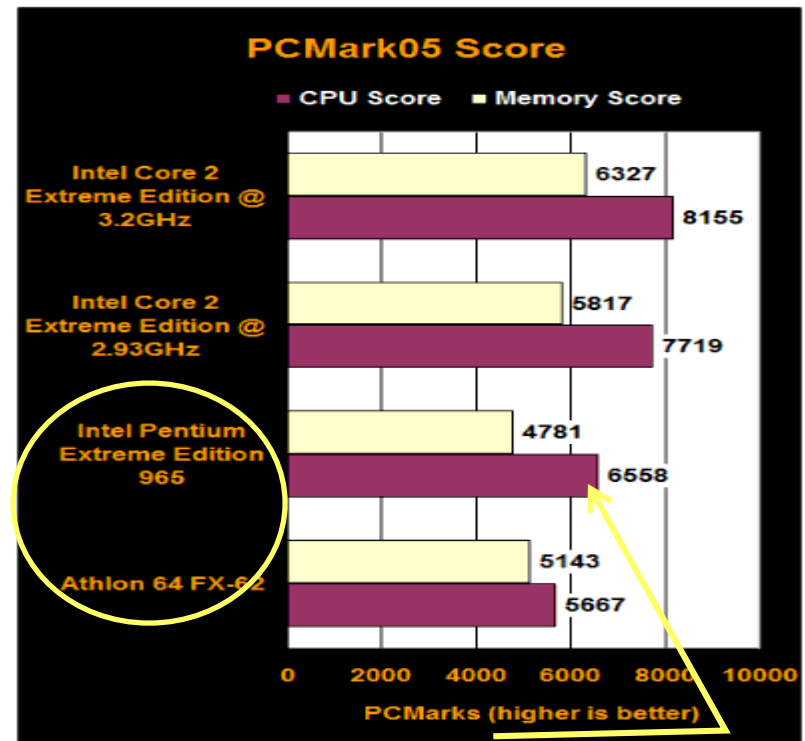
Processor	Clock [Ghz]	Score
AMD Athlon 1.33 DDR	1.33	49.2
Intel Pentium 4 - 1.5GHz	1.5	43.5
Intel Pentium 4 - 1.7GHz	1.7	45.2
Athlon XP 1800+	1.53	53.0
Intel Pentium 4 - 2.0GHz	2	50.0
Intel Pentium 4 - 2.0GHz N (Northwood)	2	53.1
Intel Pentium 4 - 2.2GHz N (Northwood)	2.2	54.2

Different Benchmarks, Different Results



- POV-Ray 3.6 is part of SPEC CPU 2006 benchmark suite.
 - widely used industry standard CPU benchmark suite, stressing a system's processor and memory subsystem.

Which CPU is "faster"?
still no clear answer



- PCMark05 consists of a series of synthetic benchmark suites.
 - each designed to test an individual subsystem, such as memory, processor, and hard drive.

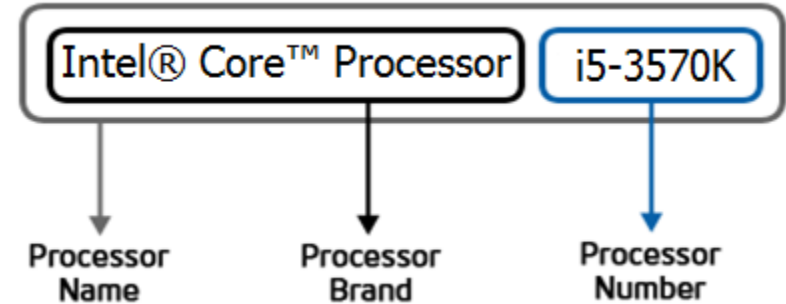
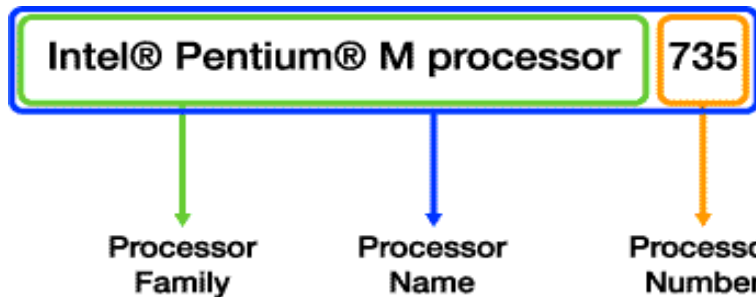


Comparing Performance

- In an ideal world
 - a single number would reliably define performance of a system
- In the real world
 - Firstly: decide what we measure: response time, throughput, CPU time
 - Secondly: run a series of benchmark programs
 - Thirdly: summarise the results
 - only then we can compare!
- Which benchmarking tool to use?
 - VeriTest and PC Magazine have discontinued benchmarking products, incl. Business Winstone
 - SPEC is still being developed
 - SPEC? PC World's WorldBench 5?
 - Easy to compare within the same brand
 - Difficult to compare between different manufacturers (Intel – AMD – Transmeta – IBM – Cyrix/VIA, etc.)

Intel's "Processor Number"

- Intel dropped "clock speed" metric and introduced a "Processor Number" (since 2004)
 - How will consumer react to this new numbering systems? More confusion or more clarity?



- A single number helps consumers to compare different processors. First digit allocated to family, next two digits based on the following characteristics:
 - Architecture. May include various architectural enhancements.
 - Cache (MB/KB).
 - Speed of the processor's internal clock (GHz/MHz),
 - Front Side Bus (GHz/MHz) - the connecting path between the processor and other key components such as the memory controller.



Intel's "Processor Number"

- For instance:
 - "535" CPU is faster than "525" ("5xx" and "5xx" - same family), but not necessarily faster than "330" ("5xx" and "3xx" are different families)
 - "520", "525", "530" are NOT equal performance enhancement steps!
- NO clearly visible reference to CPU clock speed anymore!
- Intel: the "Processor Number" is:
 - NOT the only factor in selecting a processor,
 - NOT a way to compare numbers across processor families,
 - NOT way to compare Intel and AMD CPUs
- So: we still do not have a single, universal, magic number (and we will probably never have one!).

AMD's "Model Number"

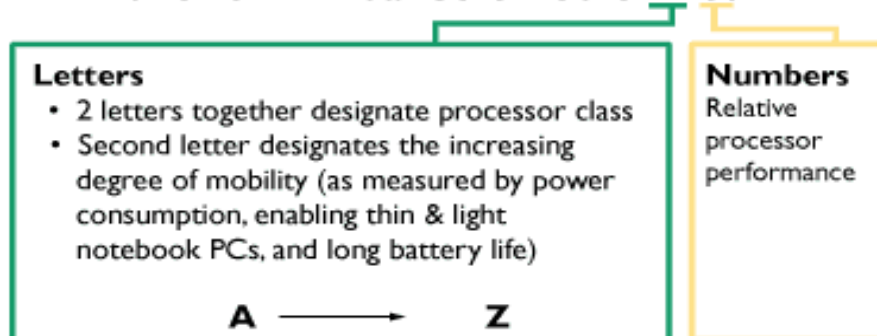
The rest is for
self-study

- Similar philosophy to AMD naming as in: "Athlon XP 3200+". Indicates relative software performance within same family only
- AMD Athlon™ 64 processors: 4 digit model number.
 - based on industry-standard benchmarks on a wide range of popular software.
- The model number methodology is designed to help end users simplify their PC purchase decision.
 - Higher the model numbers are indicative of relative performance of AMD processors in a processor family.
 - "+" indicates the "added performance benefits delivered by AMD's innovative processor designs".

AMD: Wait, There is Another "Model Number"!

- AMD Turion™ 64 X2 dual-core mobile technology: alphanumeric processor model numbers
- An example of AMD Turion 64 X2 dual-core mobile CPU:
 - Two letters followed by numbers.
 - The letters indicate the processors class
 - the second letter: increasing degree of mobility within the processor class.
 - The numbers indicate the relative processor performance: TL-60 represents better overall performance than TL-56.

AMD Turion 64 X2 Dual-Core Mobile TL-50



AMD "Model Number" Examples

Model Number	Frequency	L2 Cache	Packaging	Thermal Design Power
4000+	2.4GHz	1 MB	939-pin	89W
3800+	2.4GHz	512KB	939-pin	89W
3800+	2.4GHz	512KB	socket AM2	62W
3700+	2.2GHz	1 MB	939-pin	89W
3500+	2.2GHz	512KB	939-pin	67W
3500+	2.2GHz	512KB	socket AM2	62W

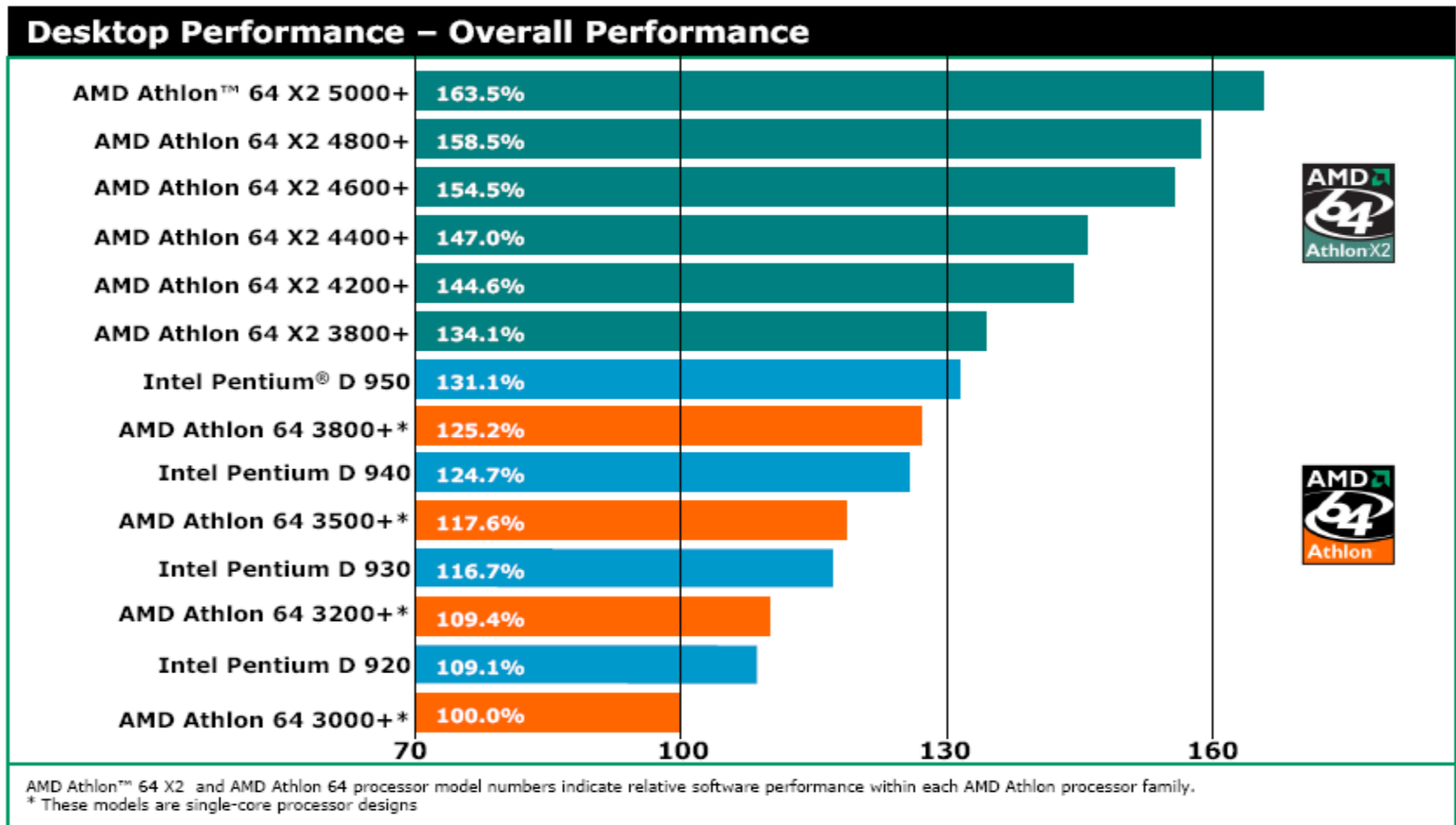
AMD Turion™ 64 Mobile Technology

Model Number	Thermal Design Power	Frequency	L2 Cache
MK-36	31W	2.0 GHz	512 KB
ML-44	35W	2.4 GHz	1 MB
ML-42	35W	2.4 GHz	512 KB
ML-40	35W	2.2 GHz	1 MB
ML-37	35W	2.0 GHz	1 MB
ML-34	35W	1.8 GHz	1 MB

Comparing AMD to Intel (different "numbers"!)

Overall Performance

(Geometric Mean of: Office Productivity, Digital Media and Games)





Revision and quiz

- When measuring CPU performance, different benchmarks may give different results:

1) True 2) False

- What is Temporal Locality about? What is Spatial Locality about?

- Handling FP immediate numbers

```
li.d $f0, 1.732
```


- FP comparison and branch

```
c.X.d $f0,$f2 # where X is: eq, lt, le
```

```
bc1t # branch if true
```

```
bc1f # branch if false
```

Recommended readings

General Data	UnitOutline LearningGuide Teaching Schedule Aligning Assessments 
Extra Materials	ascii_chart.pdf bias_representation.pdf HP_AppA.pdf instruction_decoding.pdf masking_help.pdf PCSpim.pdf PCSpim Portable Version Library materials

PH6: §5.1-§5.5: Memory hierarchy
PH5: §5.1-§5.5: Memory hierarchy
PH4: §5.1-§5.3: Memory hierarchy

Text readings are listed in Teaching Schedule and Learning Guide

PH6 (PH5 & PH4 also suitable): check whether eBook available on library site

PH6: companion materials (e.g. online sections for further readings)

<https://www.elsevier.com/books-and-journals/book-companion/9780128201091>

PH5: companion materials (e.g. online sections for further readings)
<http://booksite.elsevier.com/9780124077263/?ISBN=9780124077263>