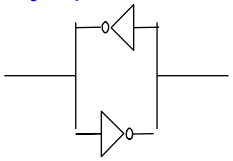
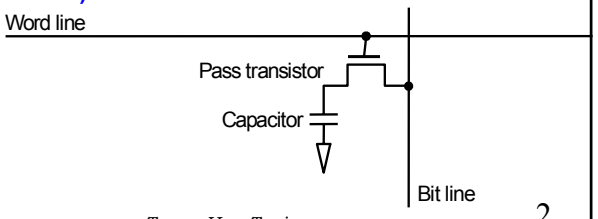
Chapter Seven

Memories: Review

- **SRAM:**
  - **value is stored on a pair of inverting gates**
  - **very fast : 3 ~20 ns; takes up more space than DRAM (4 to 6 transistors); The most expensive among the three ($1~5 per Mbyte)**
- **DRAM:**
  - **value is stored as a charge on capacitor**
  - **slower than SRAM, 45~90 ns; $0.1~0.5 per Mbyte (PC-133,128 MB, $19.95, 2001/12)**
- **Magnetic disk (Hard disk)**
  - **value is stored in the direction of magnetic field;**
  - **7~15 ms; The most cheap one among the three,$0.002~0.004 per Mbyte (40 GB, $110, 2001/12)**





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

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	145 ns
1995	64 Mb	120 ns

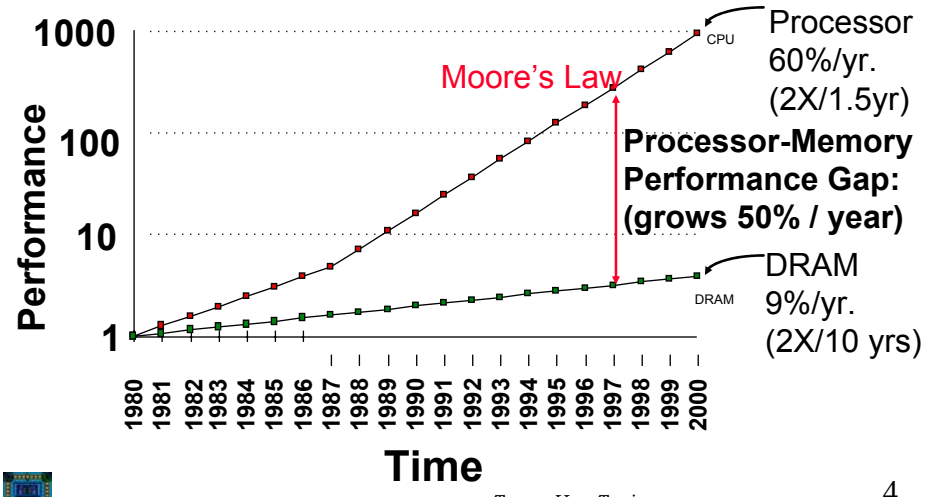
1000:1! 2:1!

•Access time: For RAM: The time it takes to perform a read or write operation. For non-RAM type, the time it takes to position the read-write mechanism at the desired location

•$T_N = T_A + N/R$, where T_N =Average time to read or write N bits, T_A =Average access time, N=Number of bits, R=transfer rate, in bits per second

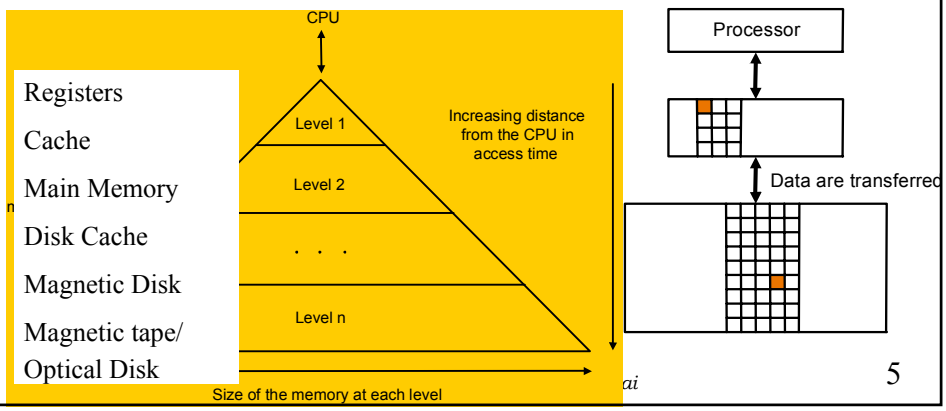
Why Memory Hierarchy?

Processor-DRAM Memory Gap (latency): Rely on Caches to bridge gap



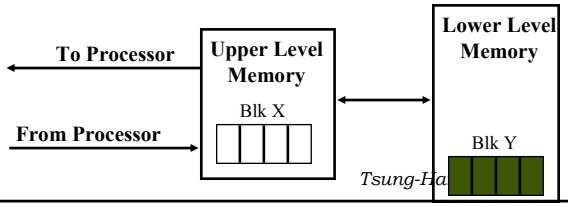
Exploiting Memory Hierarchy

- **Users want large and fast memories!**
  - **However, it is impossible to fulfill both the requirement of large and fast memory ; Example, make reference to a book in the library**
  - **Solution: build a memory hierarchy**



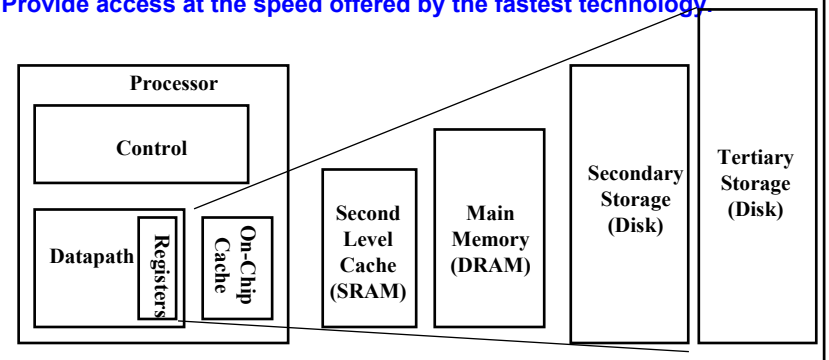
Locality: A concept to improve efficiency

- **If an item is referenced**
  - **temporal locality: it will tend to be referenced again soon**
  - **spatial locality: nearby items will tend to be referenced soon.**
- **Our initial focus: two levels (upper, lower)**
  - **block: minimum unit of data**
  - **hit: data requested is 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 RAM access time + Time to determine hit/miss**
  - **miss: data requested is not in the upper level; Miss Rate = 1 - (Hit Rate); Miss penalty: the time to replace a block in the upper level with the corresponding block from the lower level**



Memory Hierarchy of a Modern Computer System

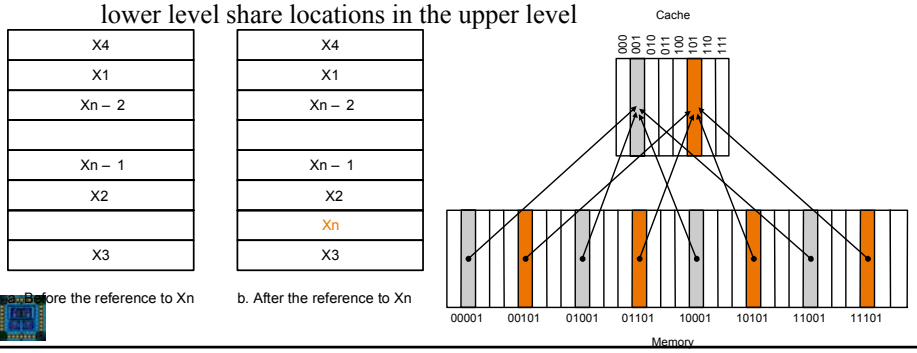
- By taking advantage of the principle of locality:
  - Present the user with as much memory as is available in the cheapest technology.
  - Provide access at the speed offered by the fastest technology.



Speed (ns):	1s	10s	100s	10,000,000s (10s ms)	10,000,000,000s (10s sec)
Size (bytes):	100s	Ks	Ms	Gs	Ts

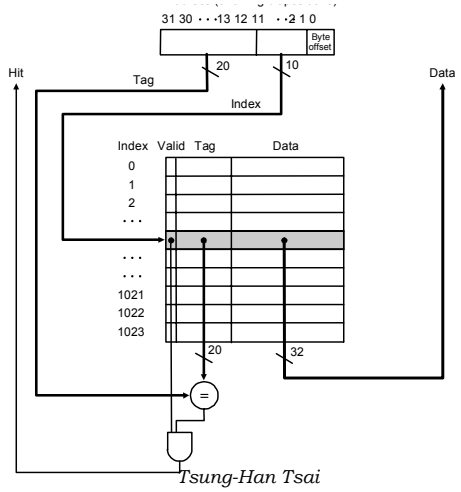
7.2 The Basics of Caches

- Two issues:
  - How do we know if a data item is in the cache?
  - If it is, how do we find it?
- Our first example:
  - block size is one word of data
  - "direct mapped": For each item of data at the lower level, there is exactly one location in the cache where it might be. (Block address) modulo (Number of cache blocks in the cache), lots of items at the lower level share locations in the upper level



Direct Mapped Cache:Single-Word block

- A referenced address is divided into
  - A cache index: Which is used to select the block
  - A tag field: compared with the tag field of the cache
- Example: P.550

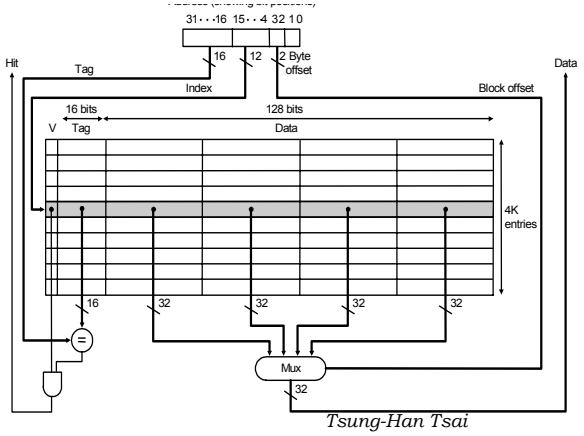


Hits vs. Misses

- Read operation
  - hits:this is what we want!
  - Misses:stall the CPU, fetch block from memory, deliver to cache, restart
- Write operation
  - method 1: write through used in DECStation 3100
    - hits:can replace data in cache and memory (write-through).It does not provide very good performance because every write causes the data to be written to main memory => A write buffer scheme to partially release this pain.
    - misses:also replace the data in the cache and memory
  - Method 2: write back, need more complex implementation
    - hits and misses: write the data only into the cache (write-back the cache later)
- Example: Fig.7.9

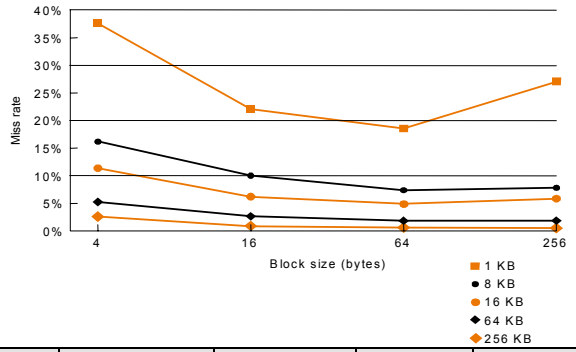
Direct Mapped Cache:Multiword Block

- **Taking advantage of spatial locality: to have a cache block that is larger than one word in length; Example: Fig.7.10 and P.556=>Read operation is the same except when miss occurred a whole block is replaced; write operation is different ! -> require read operation when miss occurs**



Performance vs. Block Size

- **Increasing the block size tends to decrease miss rate:**

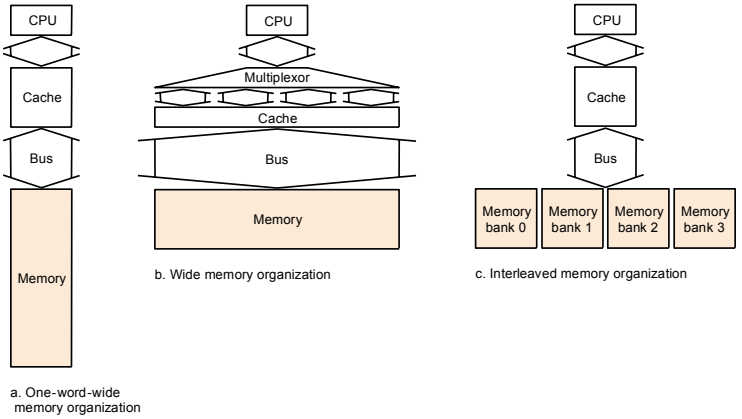


Program	Block size in words	Instruction miss rate	Data miss rate	Effective combined miss rate
gcc	1	6.1%	2.1%	5.4%
	4	2.0%	1.7%	1.9%
spice	1	1.2%	1.3%	1.2%
	4	0.3%	0.6%	0.4%

- **Use split caches because there is more spatial locality in code**

Hardware Issues

- **Make reading multiple words easier by using banks of memory**



Performance

- **Simplified model:**
  execution time = (execution cycles + stall cycles) ∗ cycle time
  stall cycles = # of instructions * miss ratio * miss penalty
- **Two ways of improving performance:**
  - **decreasing the miss ratio**
  - **decreasing the miss penalty**
- **Some Issues**
  - **Processor speeds continue to increase very fast; much faster than either DRAM or disk access times**
  - **Design challenge: dealing with this growing disparity**
- **Trends:**
  - **synchronous SRAMs (provide a burst of data)**
  - **redesign DRAM chips to provide higher bandwidth or processing**
  - **restructure code to increase locality**
  - **use prefetching**

Table 1.1 Microprocessor Comparisons

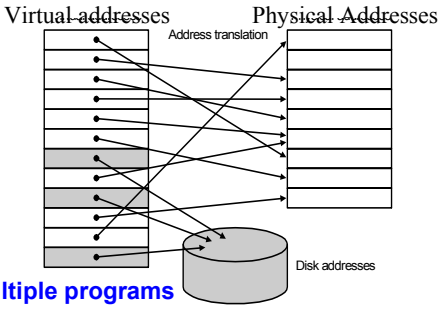
Microprocessors currently being shipped

Microprocessor	Alpha 21064	MIPS R4400SC	PA7100	Pentium	PowerPC 601	Super SPARC	68040	80486
Company	Digital Equipment Corp.	MIPS Technologies Inc.	Hewlett-Packard Co.	Intel Corp.	IBM Corp. and Motorola Inc.	Sun Microsystems Corp. and Texas Instruments Inc.	Motorola Inc.	Intel Corp.
Introduction date	2/92	11/92	2/92	3/93	4/93	5/92	1989	6/91
Architecture and organization								
Type	RISC	RISC	RISC	CISC	RISC	RISC	CISC	CISC
Width, bits (a)	64	64	32	32	32	32	32	32
On-chip cache, kB (instruction / data)	8 / 8	16 / 16	None	8 / 8	32 unified	20 / 16	4 / 4	8 unified
Off-chip cache, MB (instruction / data)	16	4	1 / 2	External controller	External controller	External controller	External controller	External controller
No. of registers (general-purpose / FP)	32 / 32	32 / 32	32 / 32	8 / 8	32 / 32	136 / 32	16 / 8	8 / 8
Instruction issues per cycle	2	1	2	2	3	3	1	1
No. of independent units	4	N.A.	3	3	3	5	N.A.	N.A.
No. of pipeline stages (integer / FP)	7 / 10	7 / 10	5 / 6	5 / 8	4 / 6	4 / 5	3 / 6	5 / N.S.
Endian (b)	Little	Big / little	Big	Little	Big / little	Big	Big	Little
Typical latency (integer / FP)	1 / 6	1 / 4	1 / 2	1 / 3	1 / 3	1 / 3	1 / 3	N.S.
Multiprocessing support?	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Technology and performance								
Technology	0.68 µm CMOS	0.6 µm CMOS	0.8 µm CMOS	0.8 µm BiCMOS	0.65 µm CMOS	0.7 µm BiCMOS	0.65 µm CMOS	0.8 µm CMOS
Die size, mm	15.3 by 12.7	12 by 15.5	14.2 by 14.2	17.2 by 17.2	11 by 11	16 by 16	10.8 by 11.7	N.S.
Transistors, millions	1.68	2.3	0.85	3.1	2.8	3.1	1.2	1.2
Metallization layers	3	2	3	3	4	3	2	3
Operating voltage, V	3.3	5 / 3.3	5	5	3.6	5.3	5	5
Clock, MHz	200	150	100	66	80	60	25	50
SPECint 92 (c)	130	88	81	67.4	85	80	21	27.9
SPECfp 92 (c)	184	97	150	63.6	105	100	15	13.1
Packaging and price								
Peak power, W	30	15	23	16	9.1	14.2	6	5
Cooling	Heat sink	Heat sink	Heat sink	Fan plus heat sink	Ambient plus heat sink	Forced air plus heat sink	Ambient plus heat sink	Fan or heat sink
Ceramic package (pins / style)	431 / PGA	447 / PGA	504 / PGA	273 / PGA	304 / QFP	293 / PGA	179 / PGA	168 / PGA
US $ price per 1000	$505	$1100	N.S.	$898	$545/$557 (d)	$999	$233	$432

CISC, RISC = reduced-, complex-instruction-set computer; FP = floating point; N.A. = not applicable; N.S. = not supplied; PGA = pin-grid array; QFP = quad flat pack.
(a) Width of integer registers.
(b) Endian refers to the order in which bytes are transferred; big = highest-order byte first, little = lowest-order byte first.
(c) Integer and floating-point performance benchmarks.
(d) IBM and Motorola's respective prices.

Virtual Memory

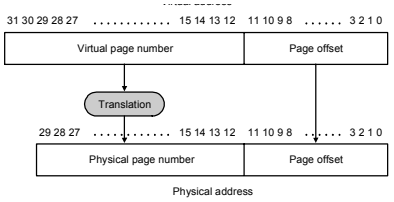
- **Main memory can act as a cache for the secondary storage (disk)**
  - **Virtual memory technique: automatically manages the two levels of the memory hierarchy represented by main memory (physical memory) and secondary storage**



- **Two major motivations**
  - **sharing of memory among multiple programs**
  - **to remove the burden of limited main memory**
- **Advantages:**
  - **illusion of having more physical memory**
  - **program relocation**
  - **protection**

Pages: virtual memory blocks

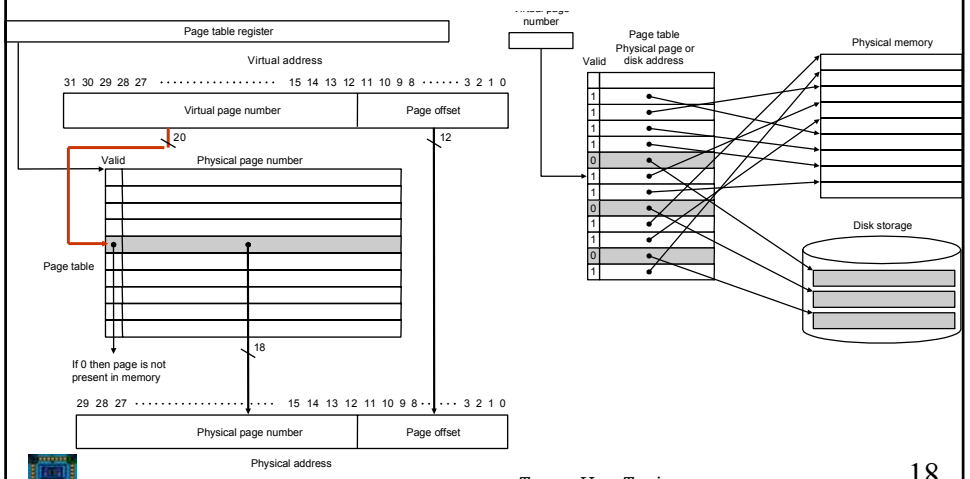
- The address is broken into a virtual page number and a page offset
- Page faults: the data is not in memory, retrieve it from disk
  - huge miss penalty, thus pages should be fairly large (e.g., 4KB)
  - reducing page faults is important
  - can handle the faults in software instead of hardware
  - using write-through is too expensive so we use writeback



  - In Fig.7.21, the main memory can have at most 1 GB, while the virtual address space is 4 GB

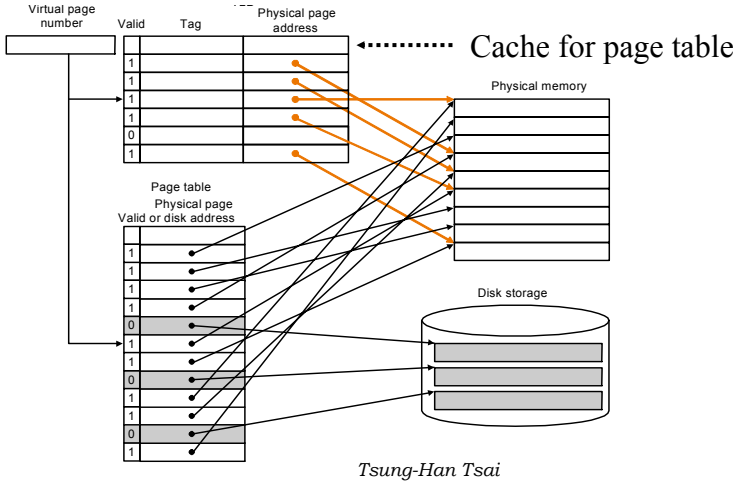
Page Tables

•Resides in memory is indexed with the page number from virtual address and contains the corresponding physical page number

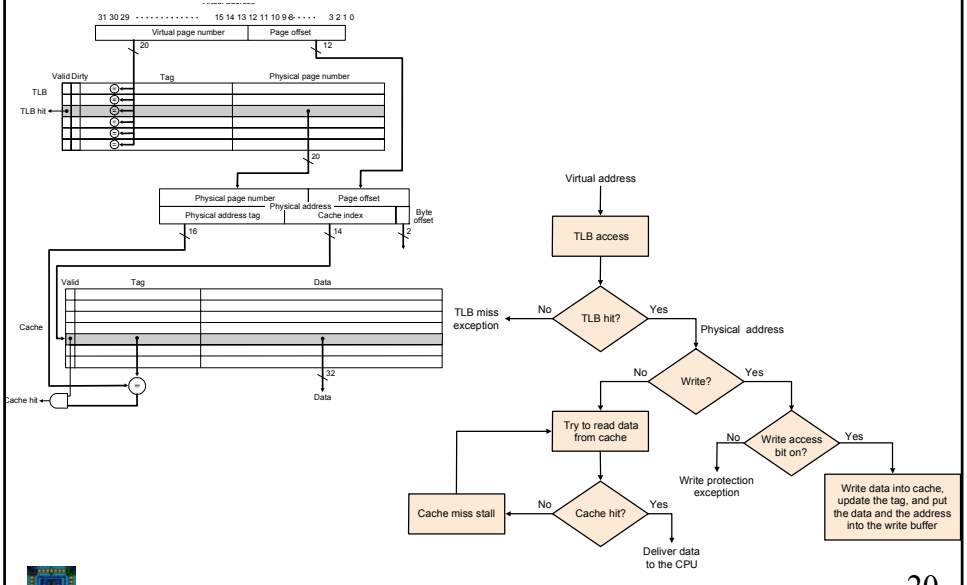


Making Address Translation Fast: The TLB

- **To improve "every memory access by a program can take at least two accee"(one page table access and one real data access in physical memory)**
  - **A cache for address translations: translation lookaside buffer**



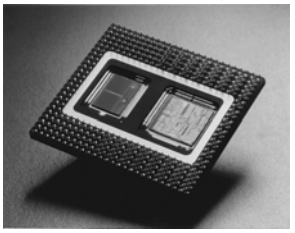
Integrating Virtual Memory TLBs and caches



Modern Systems

- **Very complicated memory systems:**

Characteristic	Intel Pentium Pro	PowerPC 604
Virtual address	32 bits	52 bits
Physical address	32 bits	32 bits
Page size	4 KB, 4 MB	4 KB, selectable, and 256 MB
TLB organization	A TLB for instructions and a TLB for data Both four-way set associative Pseudo-LRU replacement Instruction TLB: 32 entries Data TLB: 64 entries TLB misses handled in hardware	A TLB for instructions and a TLB for data Both two-way set associative LRU replacement Instruction TLB: 128 entries Data TLB: 128 entries TLB misses handled in hardware



Characteristic	Intel Pentium Pro	PowerPC 604
Cache organization	Split instruction and data caches	Split intruction and data caches
Cache size	8 KB each for instructions/data	16 KB each for instructions/data
Cache associativity	Four-way set associative	Four-way set associative
Replacement	Approximated LRU replacement	LRU replacement
Block size	32 bytes	32 bytes
Write policy	Write-back	Write-back or write-through

Some Issues

- **Processor speeds continue to increase very fast**
  — much faster than either DRAM or disk access times
- **Design challenge: dealing with this growing disparity**
- **Trends:**
  - **synchronous DRAMs (provide a burst of data), DDR SDRAM, RAMBUS**
  - **redesign DRAM chips to provide higher bandwidth or processing**
  - **restructure code to increase locality**
  - **use prefetching (make cache visible to ISA)**