CPE 431/531

Chapter 5 – Large and Fast: Exploiting Memory Hierarchy

Dr. Rhonda Kay Gaede

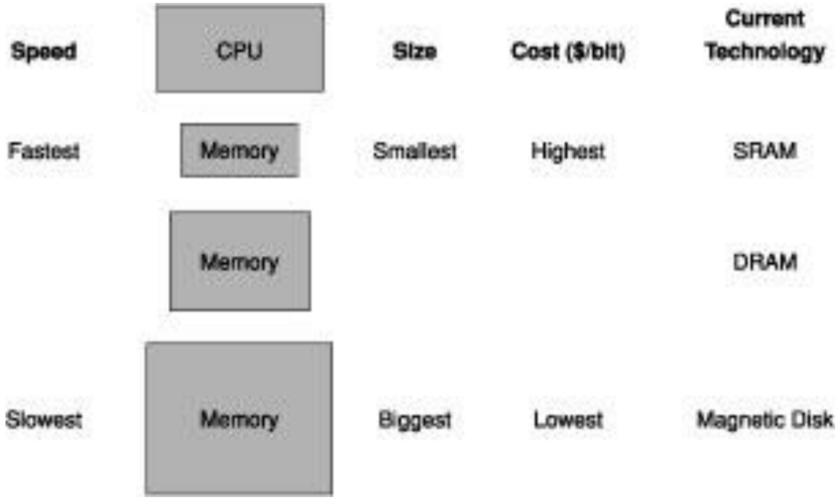
THE UNIVERSITY OF
ALABAMA IN HUNTSVILLE

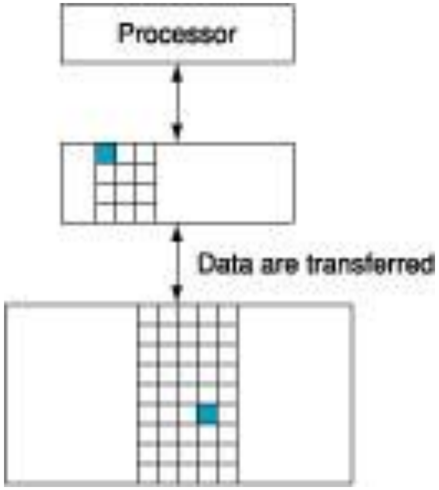
5.1 Introduction

- Programmers always want _____________ amounts of ______memory. Caches give that ___________
- Principle of Locality
  - Temporal Locality - ____ _______________
  - Spatial Locality - ____________________________ _____________________
- Build a memory _______________.

5.1 Cache Terminology

- Data is copied between only ___ levels at a time.
- The minimum data unit is a _________.
- If the data appears in the upper level, this situation is called a ___. The data not appearing in the upper level is called a _____.





5.1 More Terminology

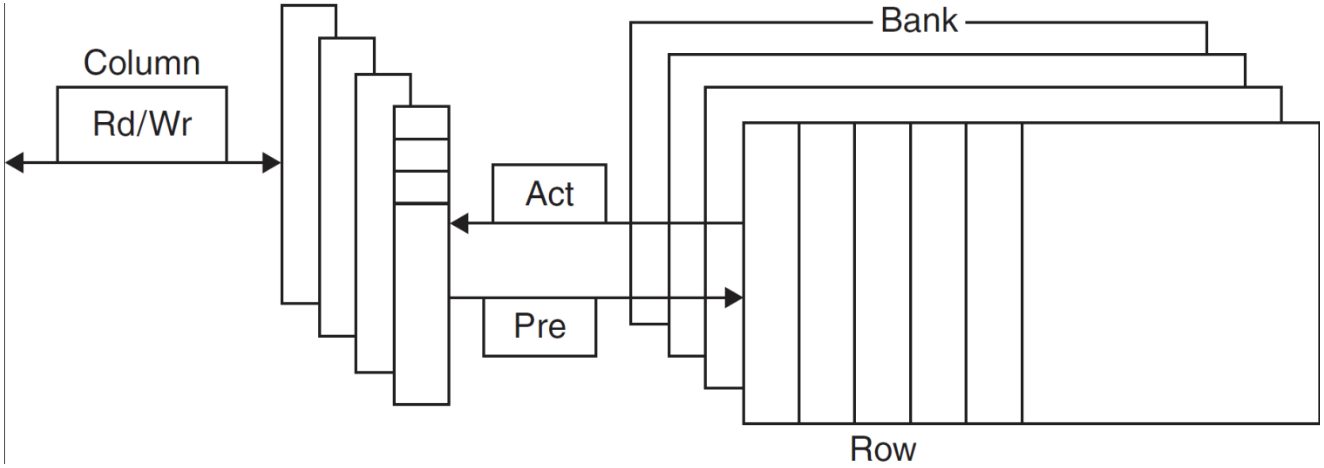
- The __________ is the fraction of memory accesses found in the upper level.
- The ___________ is the fraction of memory accesses not found in the upper level.
- The ______________ is the time to access the upper level of the memory hierarchy.
- The _______________ is the time to replace a block in the upper level with the corresponding block from the lower level, plus the time to deliver this block to the processor.

5.2 Memory Technologies

- _______ RAM (____)
  - 0.5ns – 2.5ns, $2000 – $5000 per GB
- ___________RAM (____)
  - 50ns – 70ns, $20 – $75 per GB
- __________ disk
  - 5ms – 20ms, $0.20 – $2 per GB
- Ideal memory
  - Access time of ______
  - Capacity and cost/GB of ____

5.2 DRAM Technology

- Data stored as a _________ in a _______________
  - Single _______________ used to access the _______
  - Must periodically be ________________
    - _____ contents and ______ back
    - Performed on a DRAM "____"

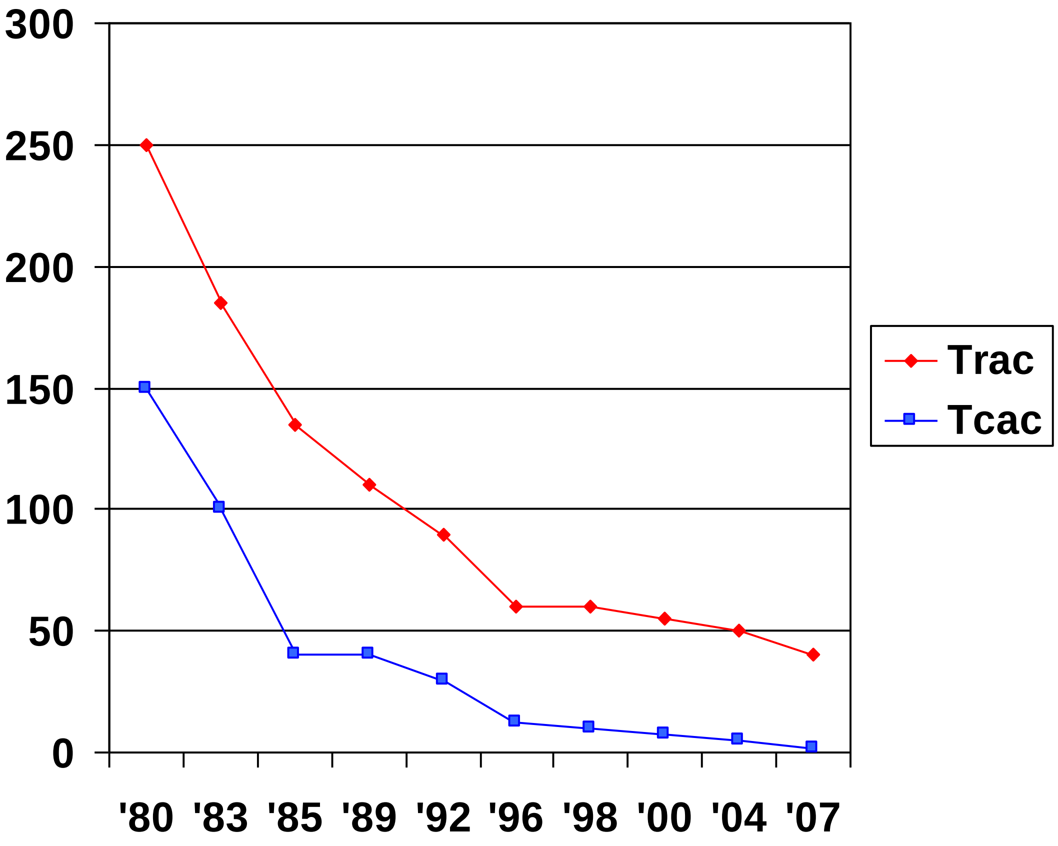


5.2 Advanced DRAM Organization

- Bits in a DRAM are organized as a ____________________ _______
  - DRAM accesses an ___________ ______
  - ________ mode: supply _________________ words from a ____ with ___________ __________
- ________ data rate (DDR) DRAM
  - Transfer on ________ and _________ clock edges
- _______ data rate (QDR) DRAM
  - Separate DDR ________ and ___________

5.2 DRAM Generations

Year	Capacity	$/GB
1980	64Kbit	$1500000
1983	256Kbit	$500000
1985	1Mbit	$200000
1989	4Mbit	$50000
1992	16Mbit	$15000
1996	64Mbit	$10000
1998	128Mbit	$4000
2000	256Mbit	$1000
2004	512Mbit	$250
2007	1Gbit	$50

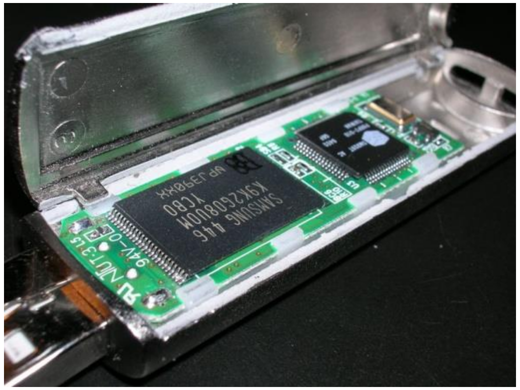


5.2 DRAM Performance Factors

- _____ __________
  - Allows ________ words to be read and refreshed in _________
- __________________ ________
  - Allows for _____________ accesses in bursts without needing to send _____ _________
  - Improves __________
- ________ ____________
  - Allows _______________ access to __________ DRAMs
  - Improves ___________

5.2 Flash Storage

- _______________ semiconductor storage
  - ___× – _______× faster than ______
  - __________, _______ power, more _________
  - But more $/GB (between ______ and _______)

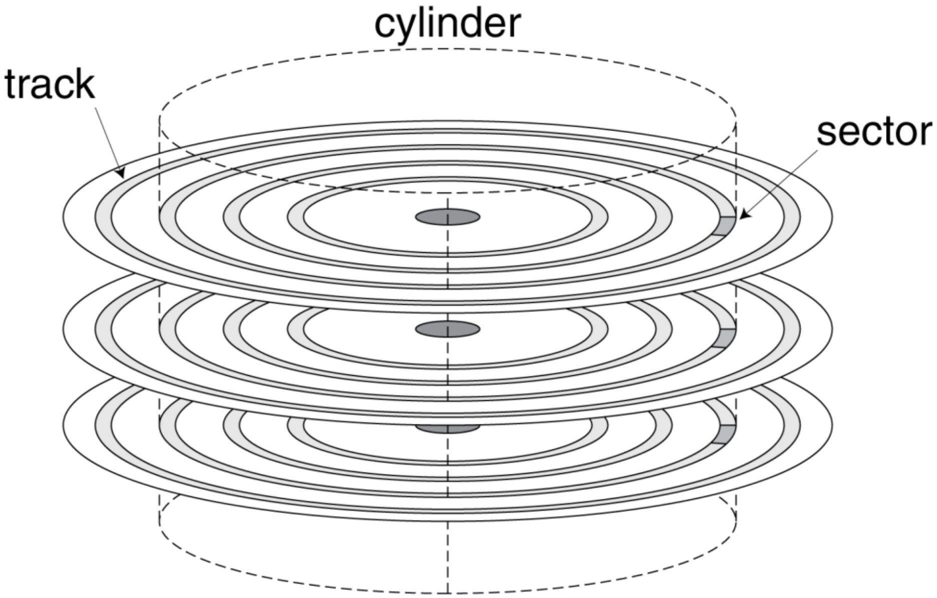


5.2 Flash Types

- _____ flash: bit cell like a _____ gate
  - _________ read/write access
  - Used for ____________ memory in ___________ systems
- ______ flash: bit cell like a ______ gate
  - _________ (bits/area), but _________________ access
  - ____________ per GB
  - Used for ____ _____, ______ _________, …
- Flash bits wears out after _________ of accesses
  - Not suitable for direct ____ or _____ replacement
  - _____ __________: ______ data to less used blocks

5.2 Disk Storage

- ______________, ___________, ___________ storage



5.2 Disk Sectors and Access

- Each _______ records
  - Sector ___
  - Data (____ bytes, ______ bytes proposed)
  - ______ correcting code (ECC)
    - Used to hide _______ and recording _____
  - ______________ fields and gaps
- Access to a _______ involves
  - _________ delay if other accesses are pending
  - _____: move the heads
  - __________ latency
  - Data ________
  - ___________ overhead

5.2 Disk Access Example

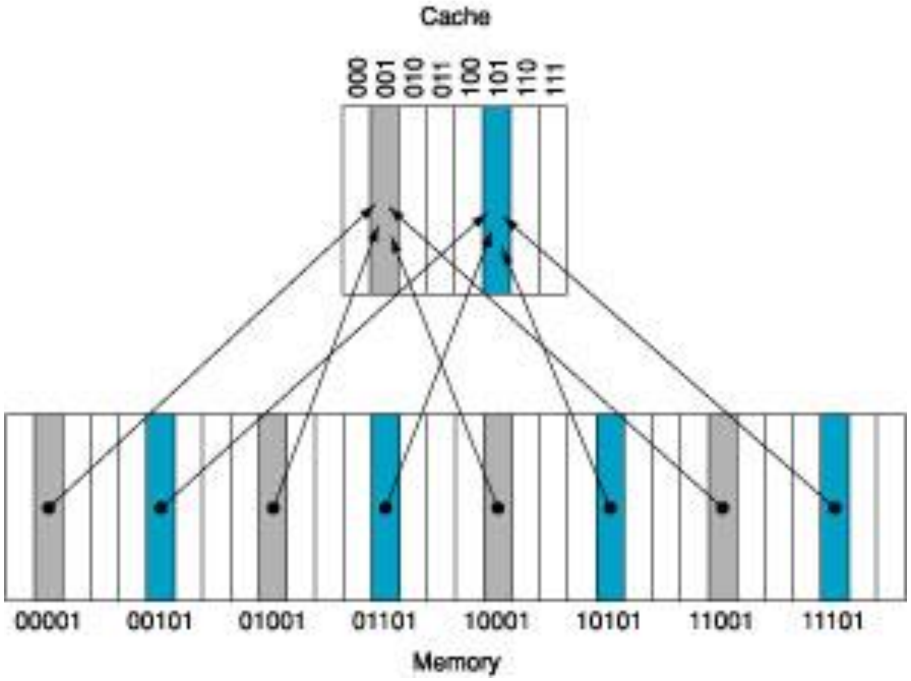
- Given
  - 512B sector, 15,000rpm, 4ms average seek time, 100MB/s transfer rate, 0.2ms controller overhead, idle disk
- Average read time
- If actual average seek time is 1ms

5.2 Disk Performance Issues

- Manufacturers quote __________ seek time
  - Based on ____ __________ seeks
  - ___________ and ___ _______________lead to smaller ________ average seek times
- Smart disk _____________ allocate _________ sectors on disk
  - Present _________ sector interface to host
  - SCSI, ATA, SATA
- Disk drives include ________
  - __________ sectors in anticipation of access
  - Avoid _____ and _____________ _____

5.3 Burning Question

- How do we know whether a data item is in the cache?
- If it is, how do we ____ it?
- The simplest scheme is that each item can be placed in __________ one place (________ mapping).
- Mapping

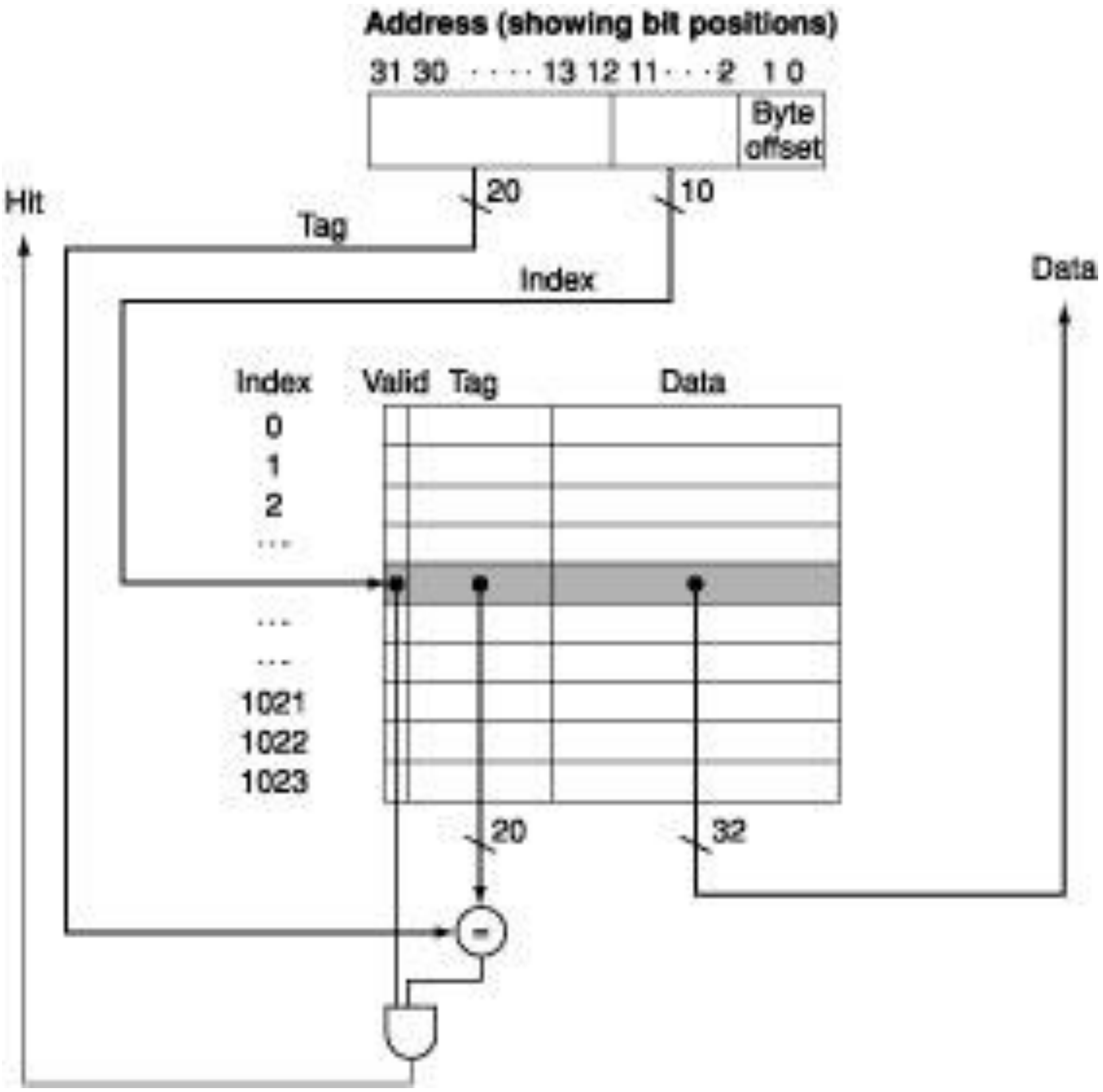


5.3 Accessing a Cache

Index	V	Tag	Data
000			
001			
010			
011			
100			
101			
110			
111			

22
26
22
26
16
3
16
18
16

5.3 Mapping Implemented in Hardware



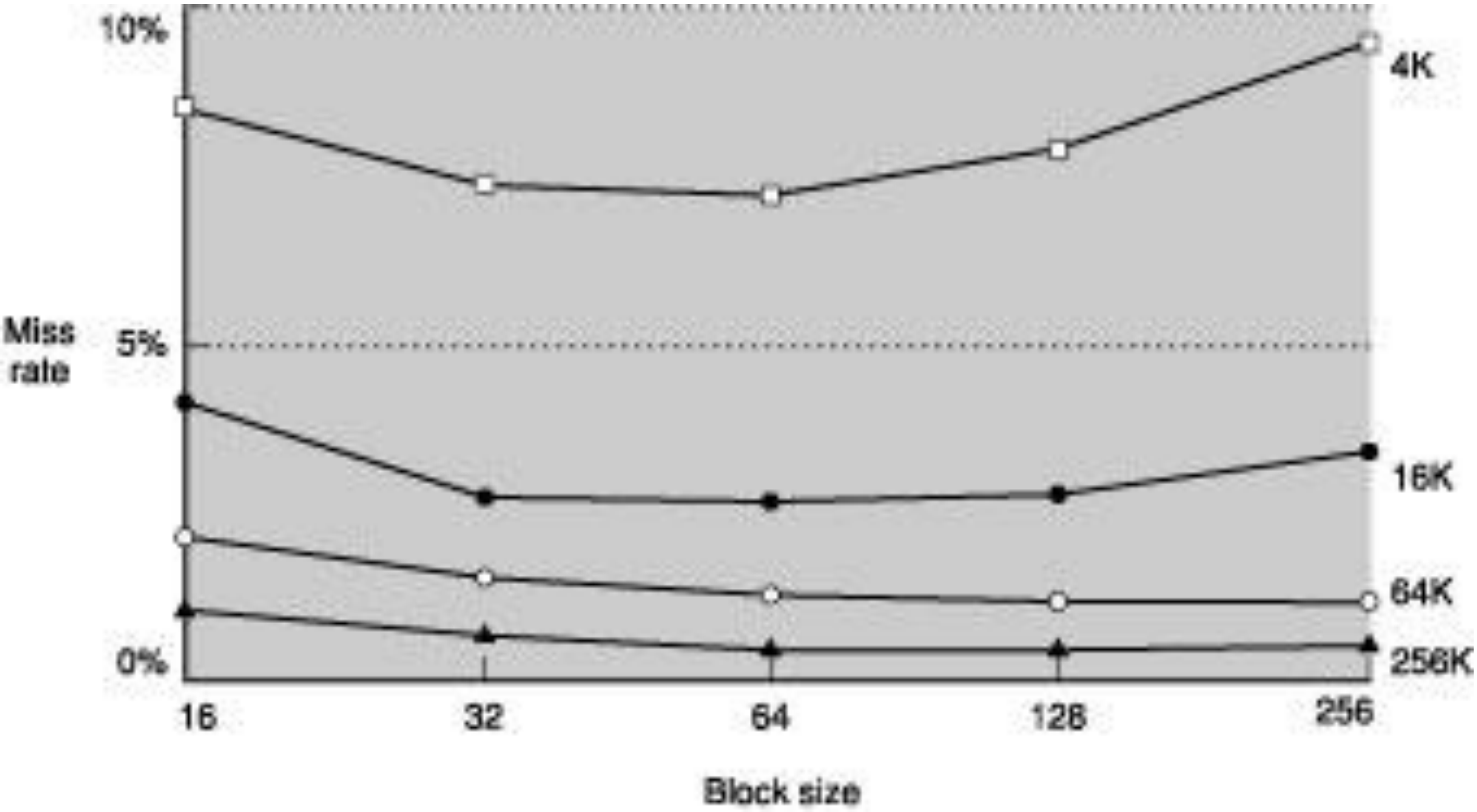
5.3 Total Storage Required

Example: How many total bits are required for a direct-mapped cache with 16 KB of data and four-word blocks, assuming a 32-bit address?

5.3 Mapping an Address to a Multiword Cache Block

Consider a cache with 64 blocks and a block size of 16 bytes. What block number does byte address 1200 map to?

5.3 Miss Rate versus Block Size



5.3 Handling Cache Misses

Instruction Cache Miss

1. Send the original PC value (current PC – 4) to the memory.

2. Instruct main memory to perform a read and wait for the memory to complete its access.

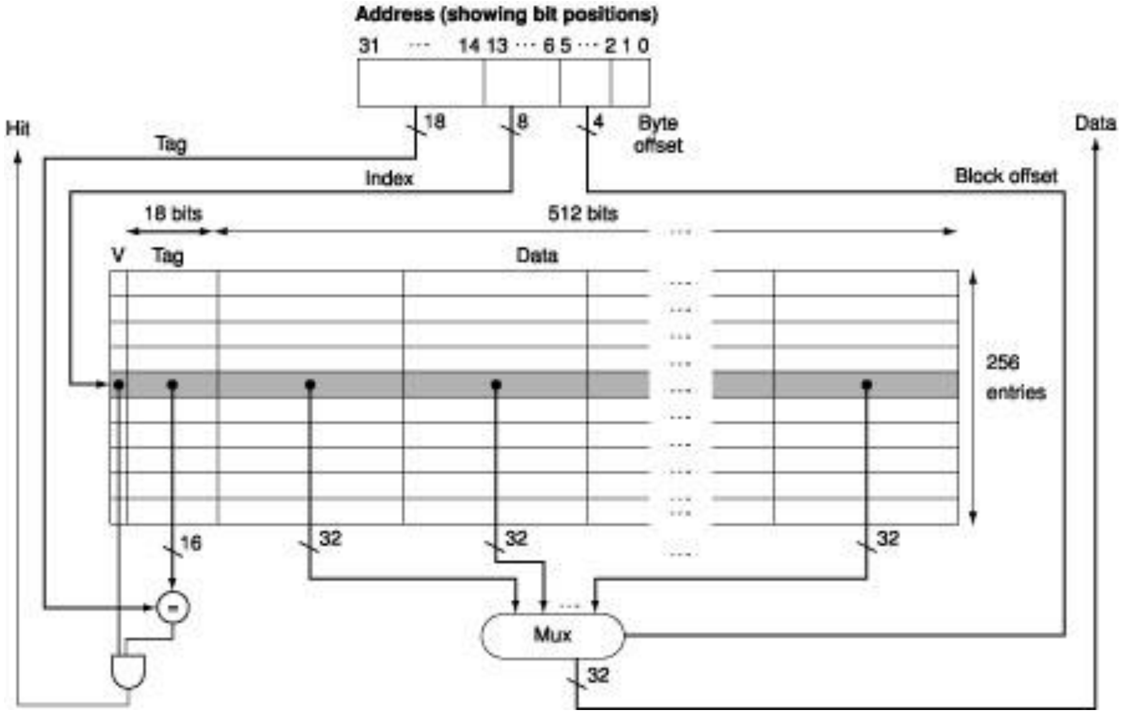
3. Write the cache entry, putting the data from memory in the data portion of the entry, writing the upper bits of the address into the tag field and turn the valid bit on.

4. Restart the instruction execution at the first step, which will refetch the instruction, this time finding it in the cache.

5.3 Handling Writes

- Suppose on a store instruction, we wrote the data into only the data cache (and not ________________ ________).
- Then the cache and main memory are said to be ____________
- Solution A: ____________________________ (______ _________)
  - Problem: ___________________________
  - Remediation: ____________________
- Solution B: ___________________________________ (______ _____)

5.3 An Example Cache



This processor has a __ stage pipeline.

When operating at peak speed, the processor can request both an _________ word and a _____ word on every clock cycle.

Separate __________ and ______ caches are used, each with __ words and __-word blocks.

For writes, the FastMATH offers both ___________ and ___________, letting the ___ decide.

The Intrinsity FastMATH Processor is a fast _____________ processor that uses the MIPS architecture and a __________ cache implementation.

5.4 Measuring and Improving Cache Performance

- CPU time = (___________________________ + _____________________) x ________________
- Memory-stall clock cycles = _________________ + ________________
- Read-stall cycles =
- Write-stall cycles =
- Memory-stall clock cycles =
- Calculating Cache Performance
  - i_{miss} = 2 %, d_{miss} = 4 %, $CPI_{perfect}$ = 2, miss penalty = 100 cycles, 36 % loads and stores

5.4 Impact of Increased Clock Rate

- Suppose the processor in the previous example ________ its clock rate, making the miss penalty _________.
- Total miss cycles per instruction = ________.
- Total CPI
- Performance with fast clock compared to performance with slow clock

5.4 Average Memory Access Time

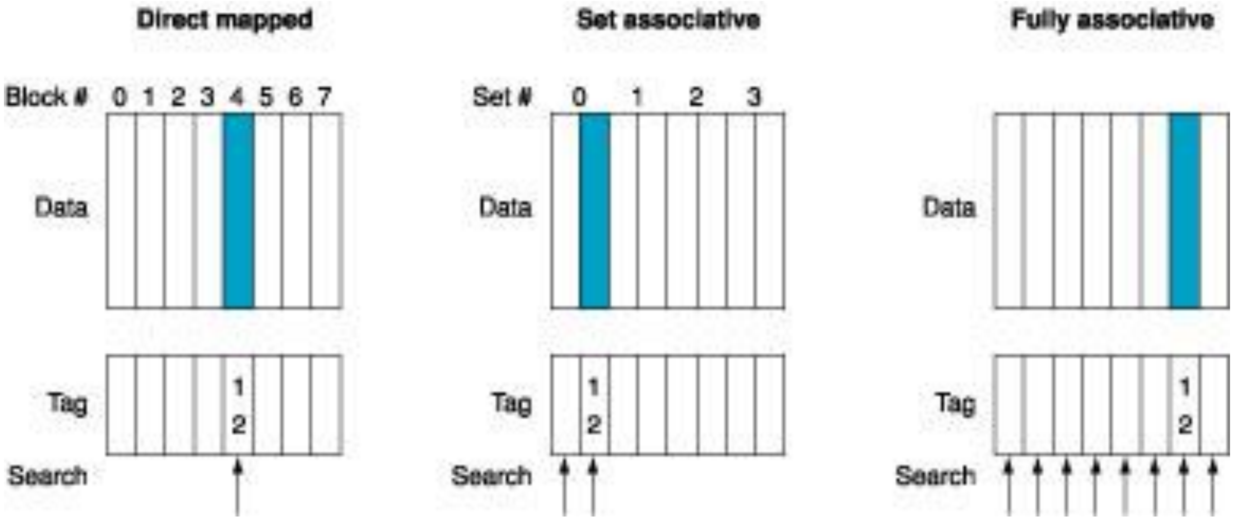
- To capture the fact that the time to _______ ____ for both ____ and _______ affects performance, designers sometimes use average memory access time (AMAT) as a way to examine ___________ ______ _______.
- AMAT = Time for a hit + Miss rate x Miss penalty
- Find the AMAT for a processor with CT = 1 ns, Miss penalty = 20 cycles, Miss rate = 0.05/instruction, Cache access time = 1 cycle. Assume that read and write miss penalties are the same and ignore other write stalls.

5.4 Flexible Placement Reduces Cache Misses

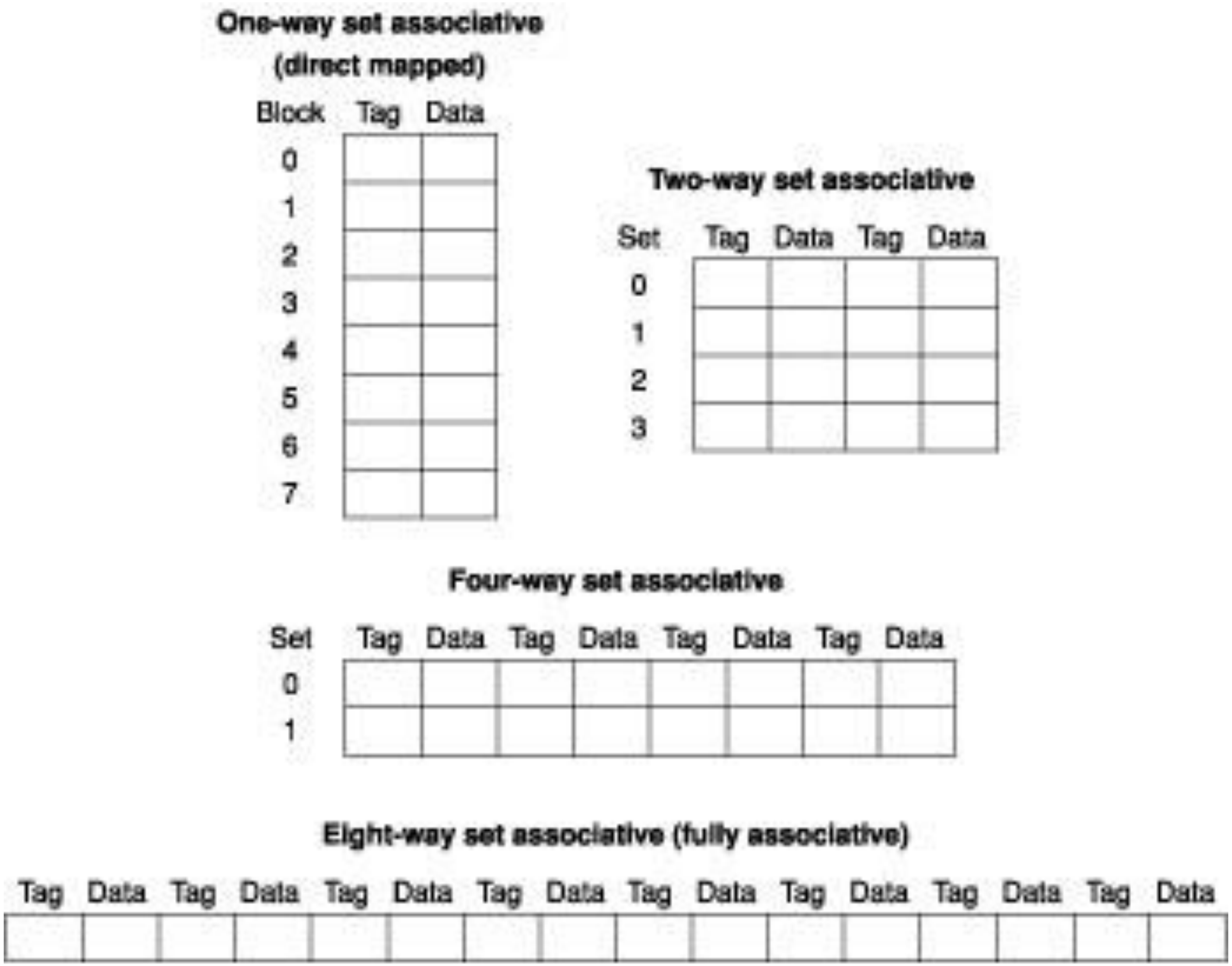
- One Extreme - direct mapped -
- Middle Range - set associative -
- Other Extreme - fully associative -
- Set associative mapping

5.4 Conceptual View of Set Associativity

- One Extreme - direct mapped -
- Middle Range - set associative -
- Other Extreme - fully associative -
- Set associative mapping



5.4 Pseudo-Implementation View of Set Associativity



One-way set associative
(direct mapped)

Block Tag Data
0
1
2
3
4
5
6
7

Two-way set associative

Set Tag Data Tag Data
0
1
2
3

Four-way set associative

Set Tag Data Tag Data Tag Data Tag Data
0
1

Eight-way set associative (fully associative)

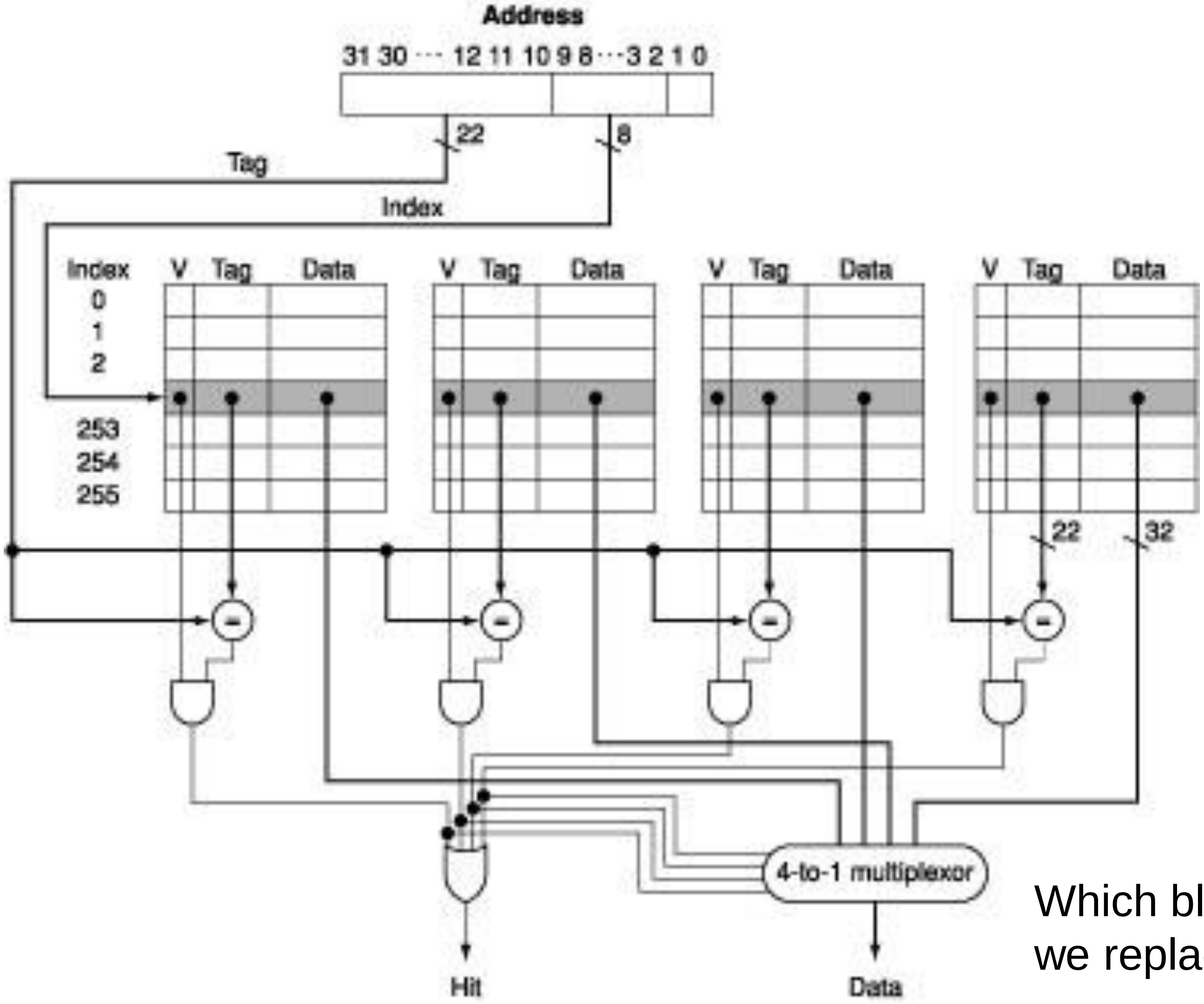
Tag Data Tag Data Tag Data Tag Data Tag Data Tag Data Tag Data Tag Data

5.4 Misses and Associativity

Look at three small caches (four one word blocks): Address sequence: 0, 8, 0, 6, 8

a. fully associative b. two-way set associative c. direct mapped

5.4 Locating a Block in the Cache



Which block do we replace?

5.4 Tag Size Considerations

- Size of Tags versus Set Associativity

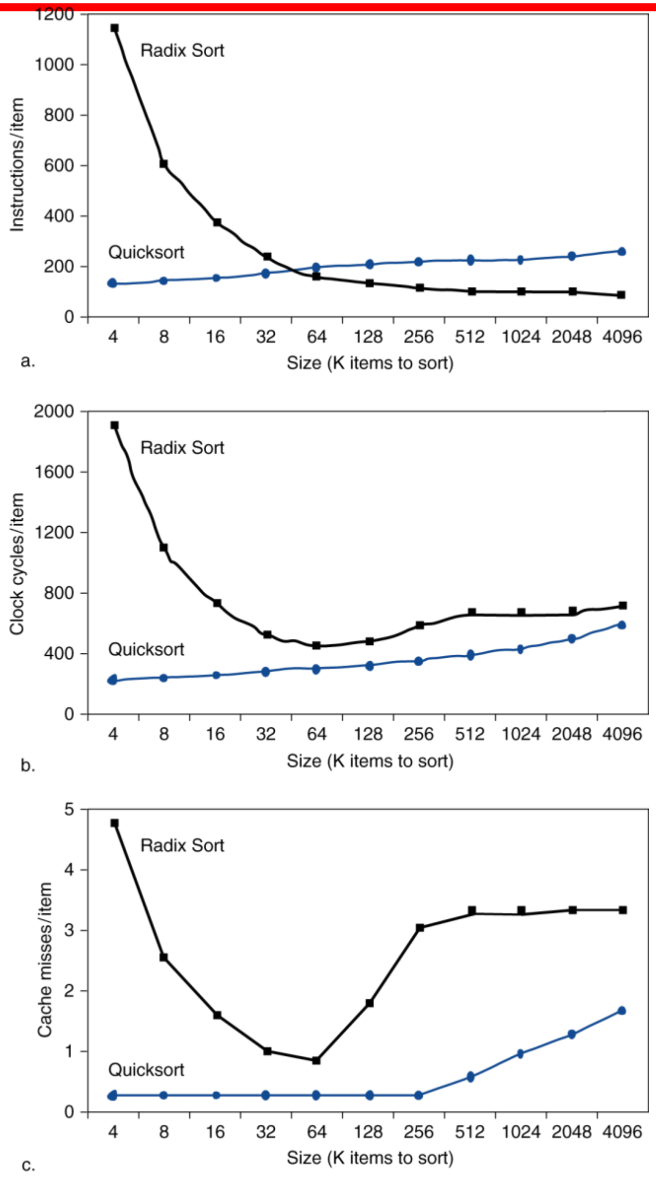
  For a cache with 4K blocks, a 32-bit address with 0 bits for block and byte offsets, find the #sets, #tag bits for 1, 2, 4 and fully associative organizations

5.4 Performance of Multilevel Caches

- Example:
  - $CPI_{base} = 1.0$, CR = 4 GHz
  - $Mem_{access} = 100$ ns, $L1inst_{miss} = 2$ %
  - $L2_{access} = 5$ ns, $L2_{miss}$ per instruction = 0.5 %

5.4 Interactions with Software

- Misses depend on memory access patterns
  - Algorithm behavior
  - Compiler optimization for memory access



5.4 Software Optimization via Blocking

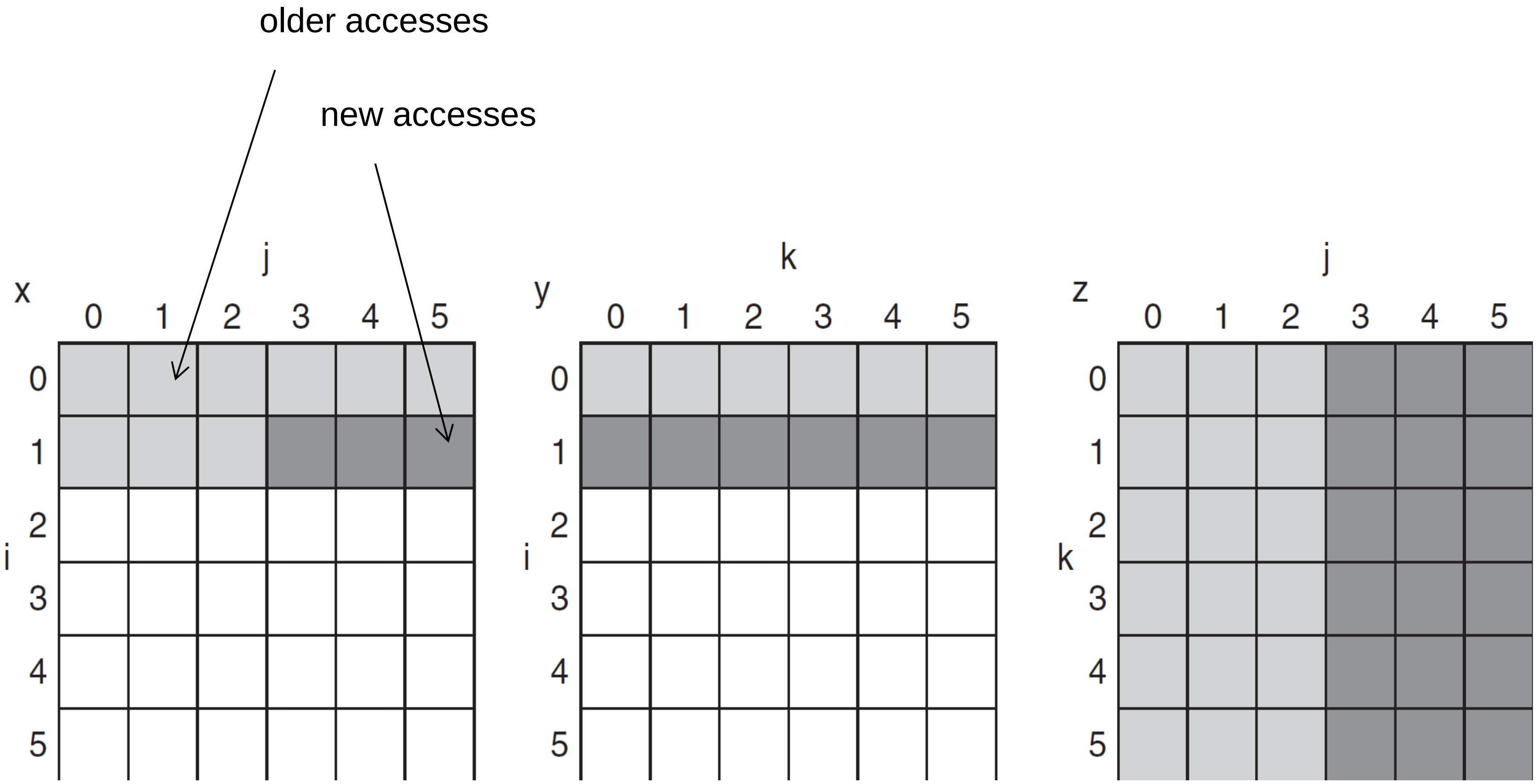
Goal: maximize accesses to data before it is replaced

Consider inner loops of DGEMM:

Blocked algorithms operate on submatrices or blocks rather than entire rows or columns of an array

```
for (int j = 0; j < n; ++j)
{
  double cij = C[i+j*n];         /* cij = C[i][j] */
  for( int k = 0; k < n; k++ )
    cij += A[i+k*n] * B[k+j*n]; /* cij += A[i][k]*B[k][j] */
    C[i+j*n] = cij;             /* C[i][j] = cij */
}
```

5.4 Array Access Patterns

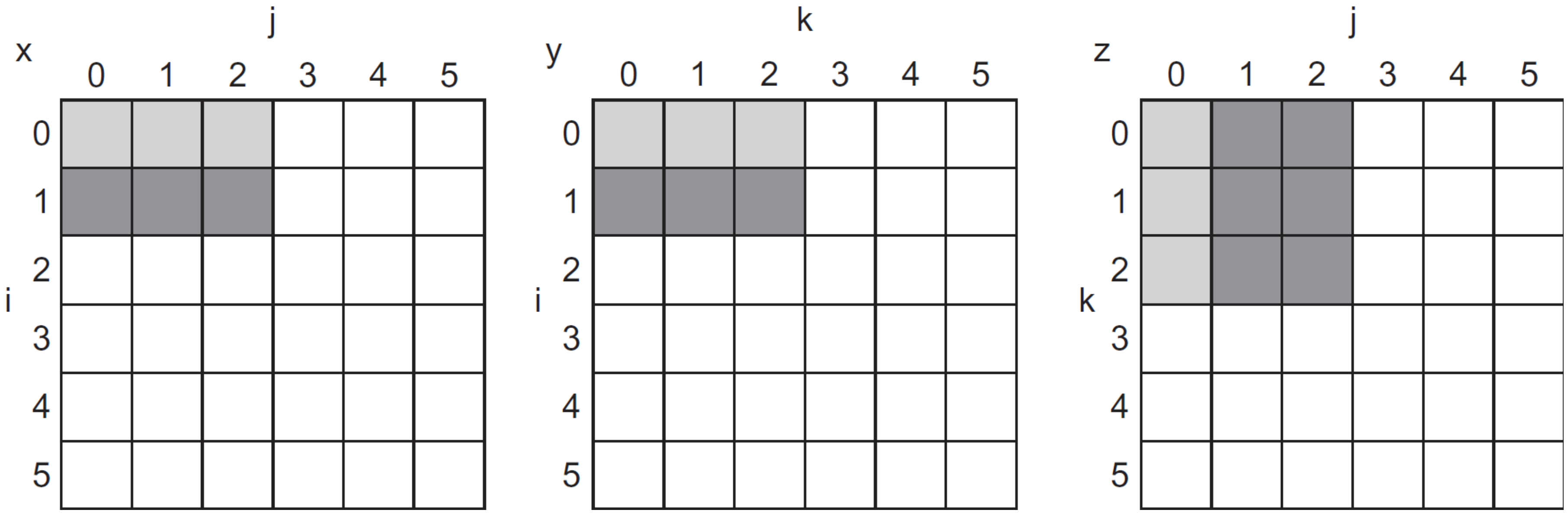

older accesses

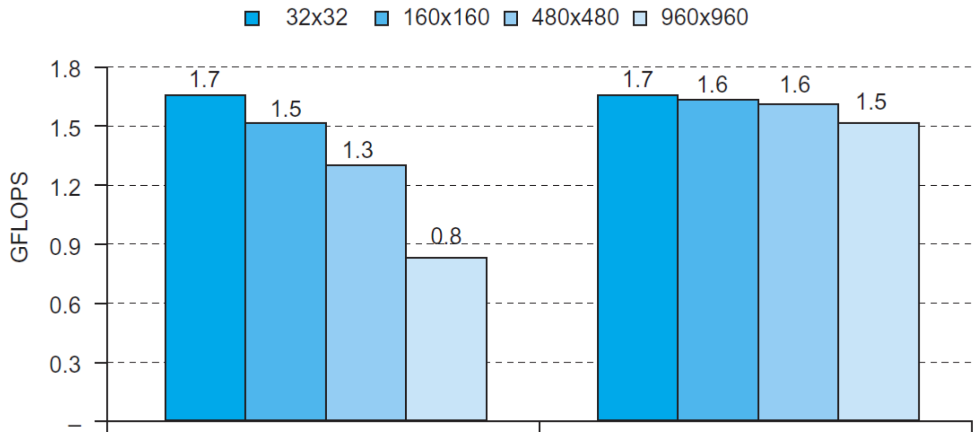
new accesses

5.4 Cache Blocked DGEMM

```
#define BLOCKSIZE 32
void do_block (int n, int si, int sj, int sk, double *A, double
*B, double *C)
{
 for (int i = si; i < si+BLOCKSIZE; ++i)
  for (int j = sj; j < sj+BLOCKSIZE; ++j)
  {
   double cij = C[i+j*n];/* cij = C[i][j] */
   for( int k = sk; k < sk+BLOCKSIZE; k++ )
   cij += A[i+k*n] * B[k+j*n];/* cij+=A[i][k]*B[k][j] */
  C[i+j*n] = cij;/* C[i][j] = cij */
 }
}
void dgemm (int n, double* A, double* B, double* C)
{
 for ( int sj = 0; sj < n; sj += BLOCKSIZE )
  for ( int si = 0; si < n; si += BLOCKSIZE )
   for ( int sk = 0; sk < n; sk += BLOCKSIZE )
    do_block(n, si, sj, sk, A, B, C);
}
```

5.4 Blocked DGEMM Access Pattern





5.5 Dependable Memory Hierarchy

- Two states of service

  1. Service accomplishment -

  2. Service interruption –

- Transitions from 1 to 2 are ____________, transitions from 2 to 1 are ______________.

- Failures can be ____________ or _______________.

- Reliability is a measure of the continuous service accomplishment, the metric is _______________________.

- Availability is a measure of the service accomplishment with respect to the alternation between the two states of accomplishment and interruption.

5.5 MTTF vs. AFR for Disks

- Some disks today are quoted to have a 1,000,000-hour MTTF. As 1,000,000 hours is 114 years, it would seem like they practically never fail. Warehouse scale computers that run Internet services such as Search might have 50,000 servers. Assume each server has 2 disks. Use AFR to calculate how many disks we would expect to fail per year.

5.5 Availability Considerations

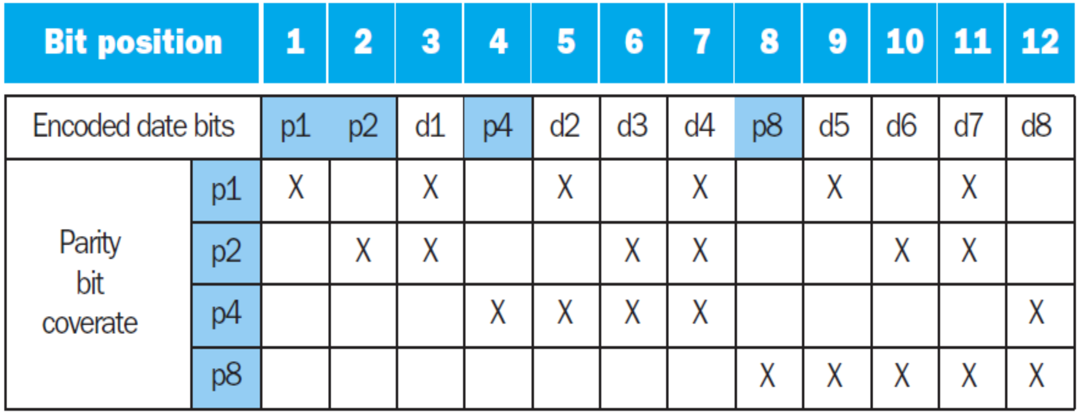
- To increase ______, you can improve the ________ of components or design systems to ________ operation in the presence of components that have _______.
- Three techniques
  - Fault avoidance
  - Fault tolerance
  - Fault forecasting
- We also need to work on decreasing ______

5.5 Single Error Detection - Parity

- Hamming distance
  - Number of ___ that are ___________ between two bit patterns
- Minimum distance = __ provides single bit error detection
  - e.g. ______ code
- Minimum distance = __ provides _______ error correction, ___ ____ error detection

5.5 Encoding Single Error Correcting Hamming Code

- To calculate Hamming code:
  - Number bits from 1 on the left
  - All bit positions that are a _______ __ __ are _______ bits
  - Each ________ bit checks certain ______ bits:



Bit position		1	2	3	4	5	6	7	8	9	10	11	12
Encoded date bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8
Parity bit coverate	p1	X		X		X		X		X		X	
	p2		X	X			X	X			X	X	
	p4				X	X	X	X					X
	p8								X	X	X	X	X

5.5 Decoding Single Error Correcting Hamming Code

- ______ of parity bits indicates which bits are __ _____
  - Use numbering from _________ procedure
  - E.g.
    - Parity bits = _____ indicates ___ _____
    - Parity bits = _____ indicates bit ___ was flipped

5.5 SEC/DEC Hamming Code

- Add an additional parity bit for the _______ _____ (p_n)
- Make Hamming distance = __
- Decoding:
  - Let H = SEC parity bits
    - H even, p_n even, __________
    - H odd, p_n odd, _____________________________________
    - H even, p_n odd, ________________
    - H odd, p_n even, ________________________________
- Note: ECC DRAM uses SEC/DEC with 8 bits protecting each 64 bits

5.6 Virtual Machines Redux

- First developed in the 1960s, they have remained an important part of ___________ computing and have recently gained popularity due to
  - Increasing importance of __________ and __________
  - The failures in ___________ and ___________ of standard operating systems
  - The _________ of a single computer among many related users
  - The dramatic increase in ____ _____ of processors
- Broadest Definition
  - Includes basically all ___________ methods that provide a standard software interface, like the __________________________________
- Our Definition
  - Provide a complete _______________ environment at the _____ level

5.6 Virtual Machine Basics

- System virtual machines present the illusion that users have an _____ _______ to themselves, including a copy of the _________ ______.
- With a VM, multiple OSes all ____ the ________ resources.
- The software that supports VMs is called a _____ _______ ______ (VMM) or _________.
- The underlying hardware is called the ____, sharing resources among the ______ VMs.

5.6 Virtual Machine Ancillary Benefits

- Our interest is primarily in improving ____________
- Other benefits include
  - _________ ________: a typical deployment might be some OSes running legacy OSes, many running the current stable OS release, and a few testing the next OS release.
  - _________ _________: Consolidate the number of servers. Some VMMs support migration of a running VM to a different computer, either to balance load or to evacuate from failing hardware.

5.6 Requirements of a Virtual Machine Monitor

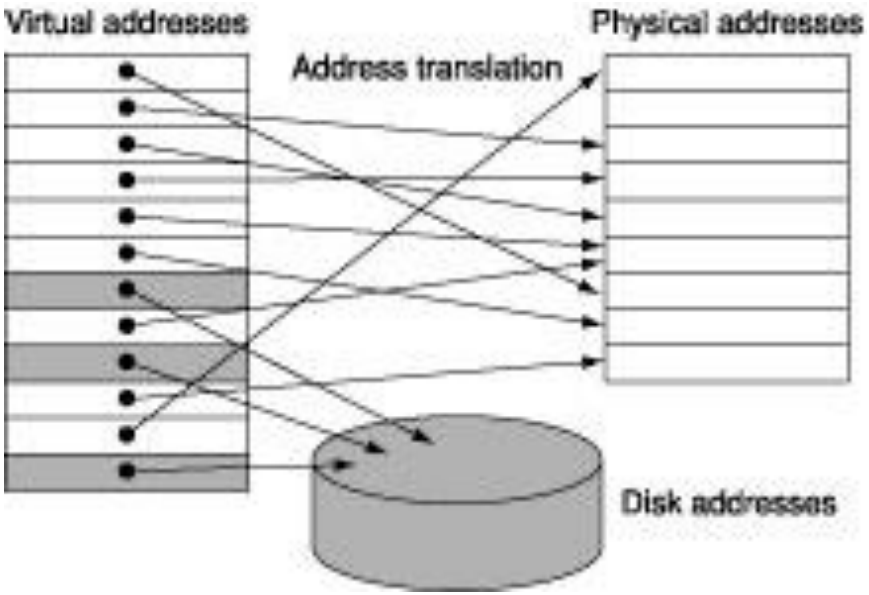
- Guest software should behave on a VM exactly as if it were running on the _____ _______, except for _________________ behavior or limitations of _____ _________ shared by multiple VMs.
- Guest software should not be able to ______ __________ of real system resources directly.
- At least ___ processor modes, ______ and ____.
- A _________ subset of instructions available only in ______ mode, all system ___________ must be controllable only via these instructions.

5.7 Virtual Memory

- The ___ _______ can act as a ______ for the ____________ storage.
- Historically, two motivations for virtual memory
  - ____
  - ____
- Virtual memory implements the _______________ of a program’s address space to ____________________.
- This translation process enforces _________________ of a program’s address space from other programs.

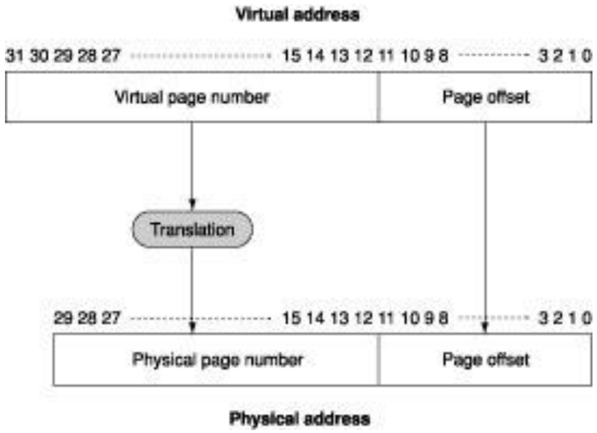
5.7 Virtual Memory Terminology

- A virtual memory _______________ is called a _______.
- A virtual memory _________ is called a ______ _________.
- Each _______ address is translated to a ________ address.
- This process is called _______________________.



5.7 Virtual Memory Facts

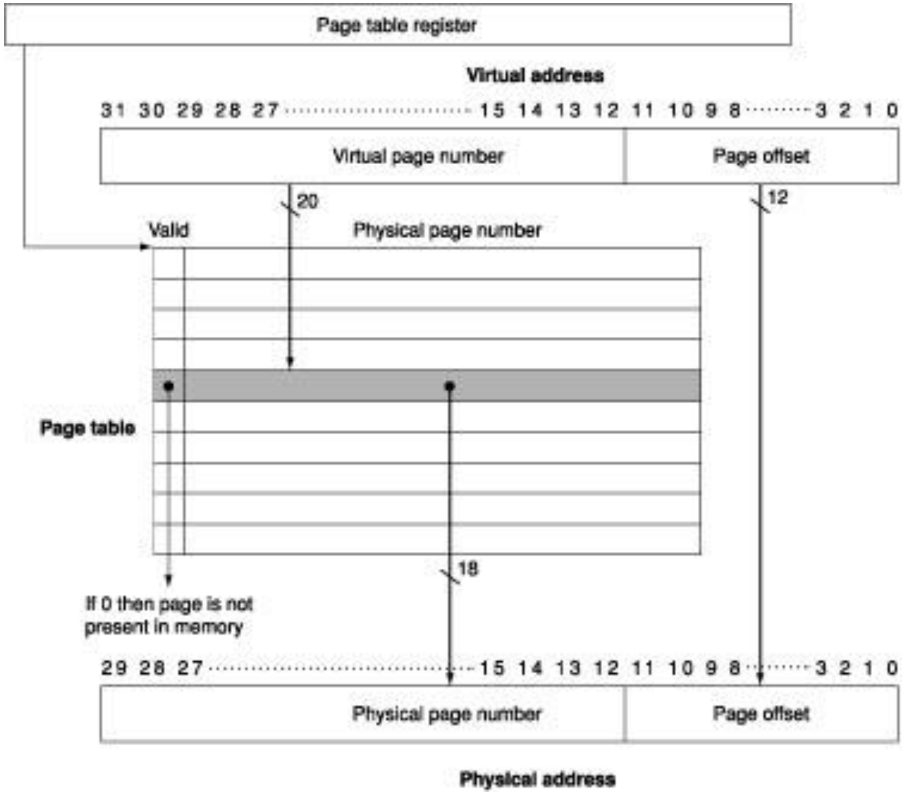
- A virtual _______ is broken into a virtual ____ _______ and a ____ ______.



- A page fault takes _______ of cycles to process
  - Pages should be _____ enough to ________ the high access time, though ________ systems are going smaller.
  - _____ __________ placement of pages is _______.
  - Page faults can be handled in ________.
  - Virtual memory uses ________.

5.7 Virtual Memory Mapping

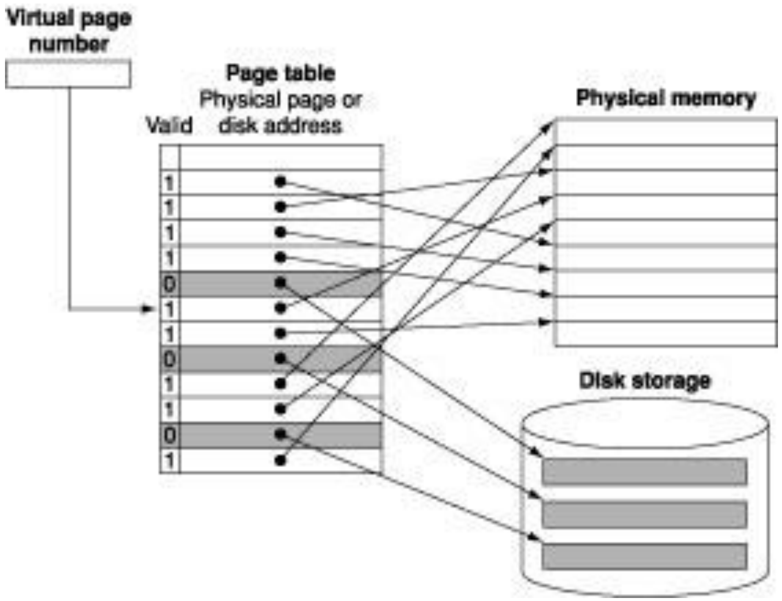
- Pages are ________ by using a _____ ______ that _______ memory.
- Each ________ has its own _____ _______.
- The _____ ______ register points to the __________ of the _____ ______.
- A _____ table is too __________, _____________ page tables are used.
- The _______ of a ________ consists of the _____ ______ _______, _______ _______ and _________.



5.7 Page Faults

- The __________ ________ manages page replacement.
- The __________ ________ usually creates the ______ __ ____ for all of the pages of a process when it creates the process, this space is called ______ ________

- A data structure records where _____ _____ is stored on disk. Another data structure tracks which ___________ and which _____ __________ use each ________ _____.
- On a page fault, the _______________
- _____ page is evicted.
- Consider 10, 12, 9, 7, 11, 10, then 8



5.7 Making Address Translation Fast: The TLB

- With virtual memory, you need ___ memory accesses, one extra for the _________.
- Add a ____ to keep track of _____ translations.
- It's called a __________ ________ _____ (TLB).
- A TLB ____ may or may not be a ____ _____

TLB characteristics

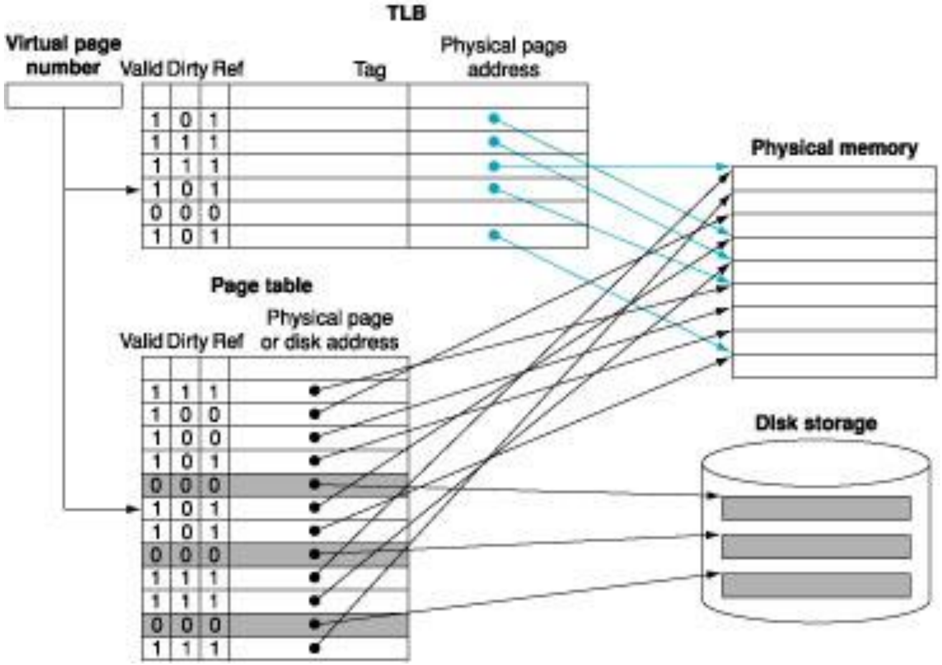
size: ________________

block size: ___________

hit time: ______________

miss penalty: ___________

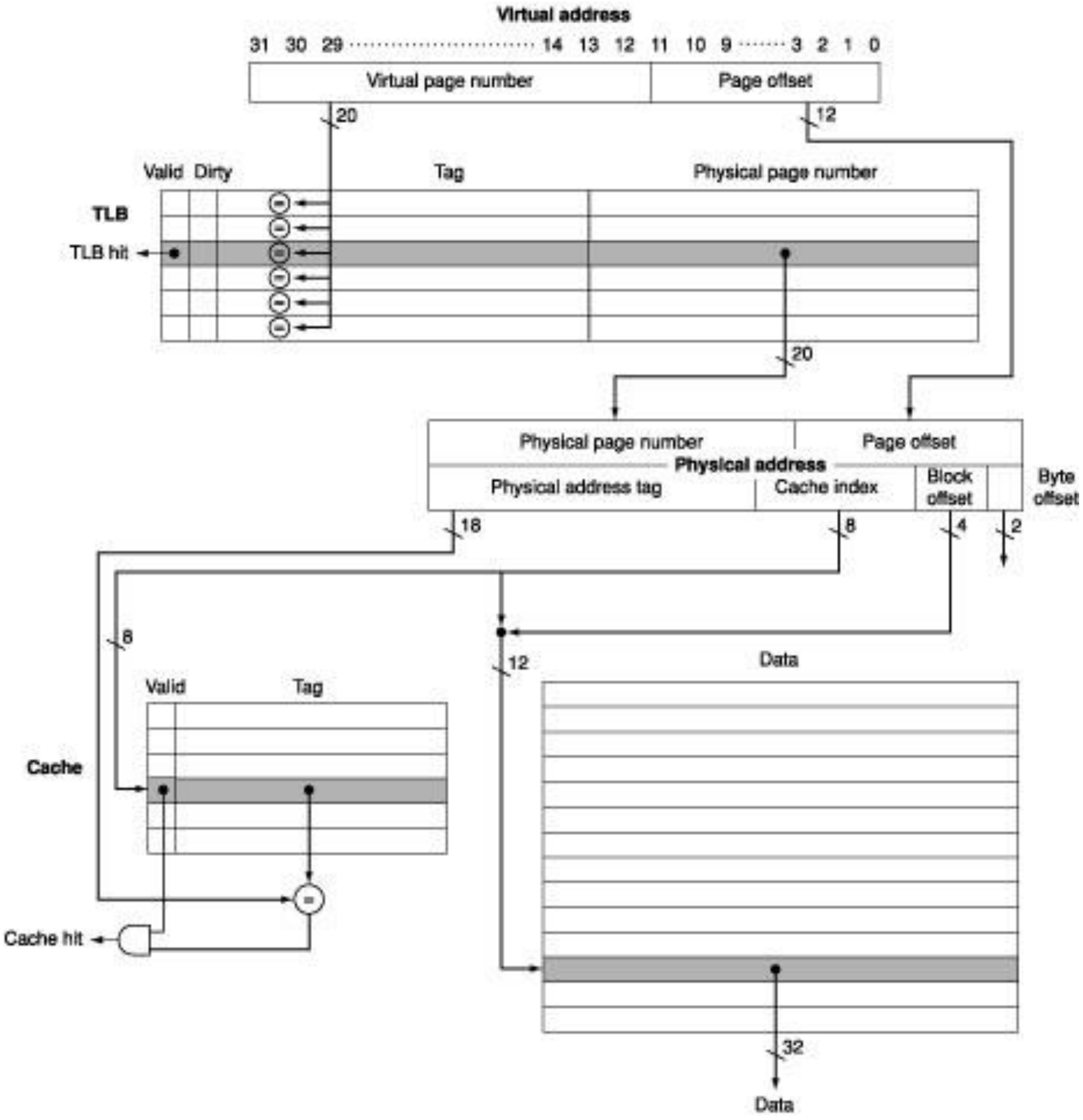
miss rate: ______________



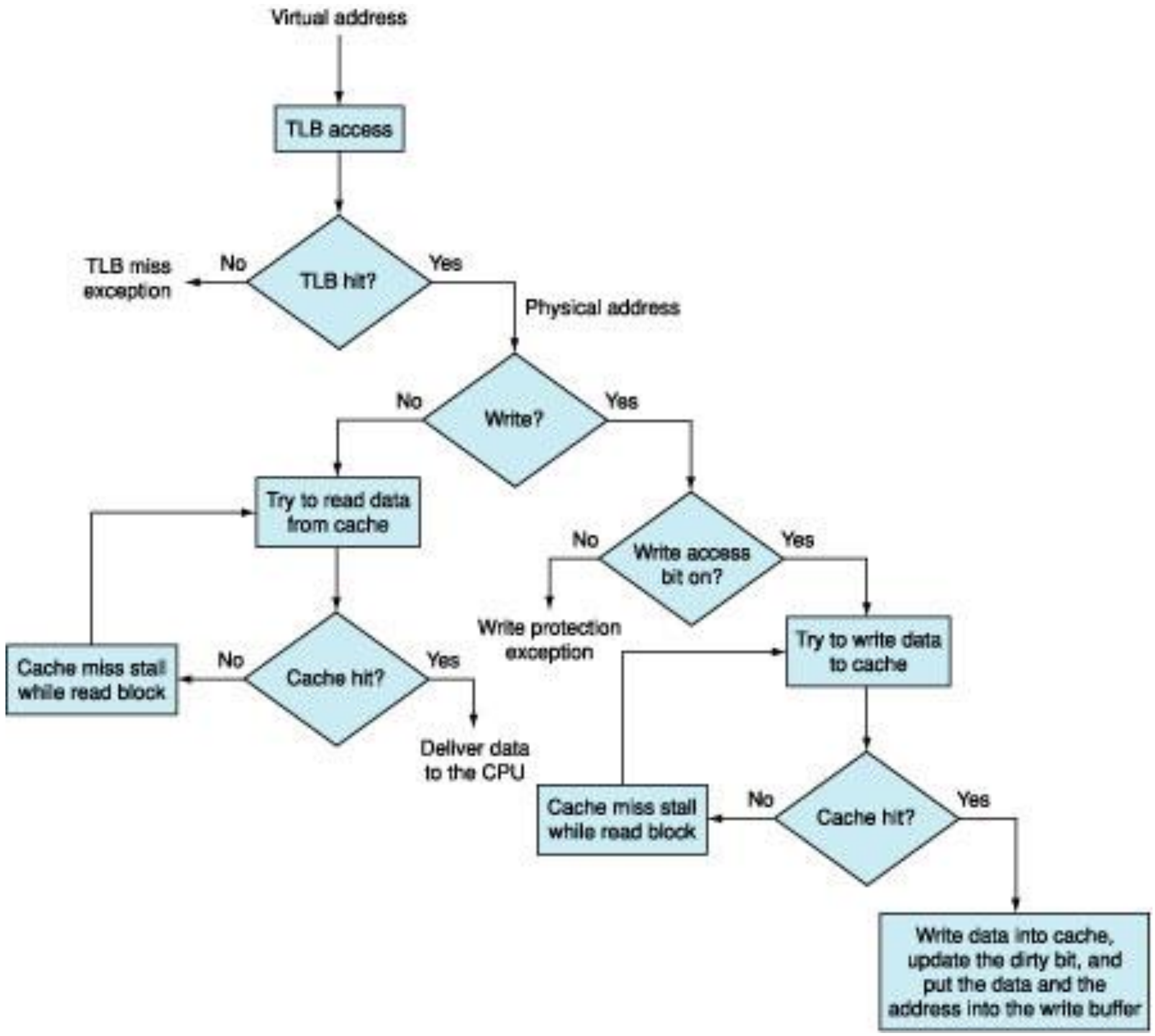
Valid	Dirty	Ref
1	0	1
1	1	1
1	1	1
1	0	1
0	0	0
1	0	1

Valid	Dirty	Ref
1	1	1
1	0	0
1	0	0
1	0	1
0	0	0
1	0	1
1	0	1
0	0	0
1	1	1
1	1	1
0	0	0
1	1	1

5.7 The Intrinsity FastMATH TLB



5.7 Processing a read or write-through


Virtual address
TLB access
TLB hit?
No
TLB miss exception
Yes
Physical address
Write?
No
Yes
Try to read data from cache
Write access bit on?
No
Write protection exception
Yes
Try to write data to cache
Cache miss stall while read block
Cache hit?
No
Yes
Deliver data to the CPU
Cache miss stall while read block
Cache hit?
No
Yes
Write data into cache, update the dirty bit, and put the data and the address into the write buffer

5.4 Overall Operation of a Memory Hierarchy

TLB	Page Table	Cache	Possible? If so, under what circumstance?
miss	miss	miss	TLB misses and is followed by a page fault; after retry, data must miss in cache
miss	miss	hit	Impossible: data cannot be allowed in cache if the page is not in memory
miss	hit	miss	TLB misses, but entry found in page table; after retry, data misses in cache
miss	hit	hit	TLB misses, but entry found in page table; after retry, data is found in cache
hit	miss	miss	Impossible: data cannot be allowed in cache if the page is not in memory
hit	miss	hit	Impossible: data cannot be allowed in cache if the page is not in memory
hit	hit	miss	Possible, though the page table is never really checked if TLB hits

5.7 Implementing Protection with Virtual Memory

- Hardware Must Provide
  - At least two modes
    - ___________________
    - ___________________
  - Provide a portion of the ________ ____ that a user process can ____ but not ____
    - ____
  - Provide mechanisms whereby the processor can move between modes
    - ____________________________
    - _____________________________________
- Software Can Help
  - Place the ____ ______ in the __________ address space of the

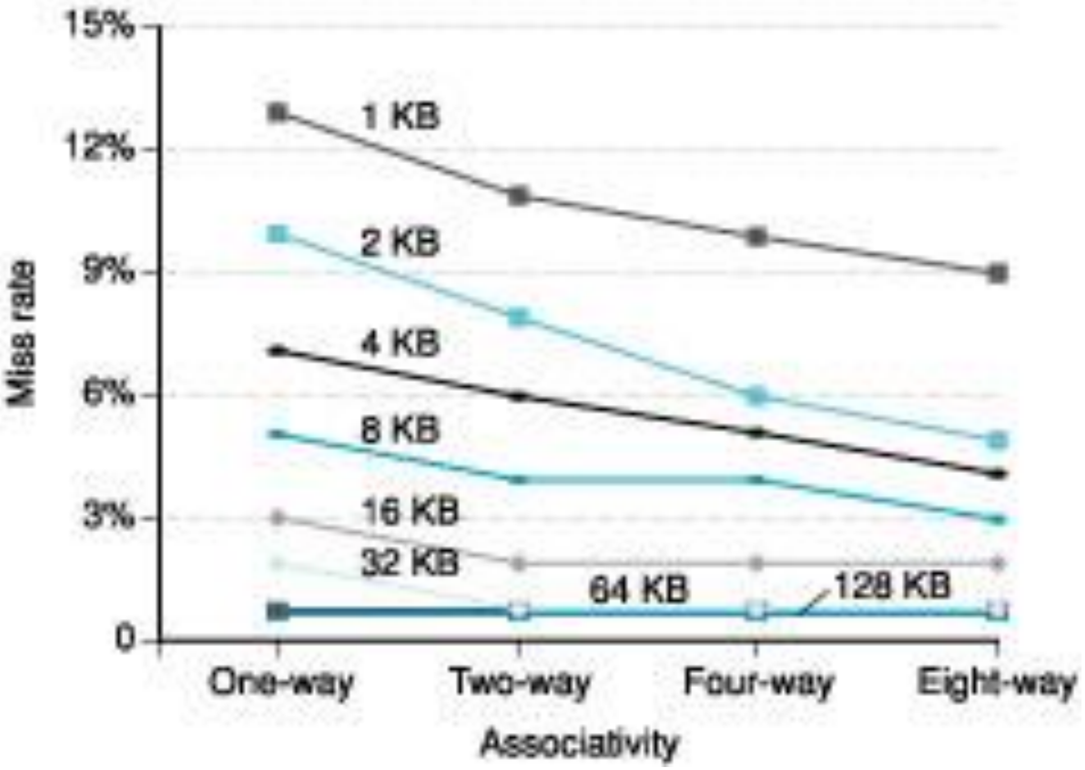
    ____

5.7 Summary

- Pages are made _________ to take advantage of ____________ locality and reduce the _______ rate.
- The mapping between virtual addresses and physical addresses, which is implemented in a _______ _________, is made ________ _________________so that a virtual page can be placed _____________ in main memory.
- The ______________ ________uses techniques, such as LRU and a reference bit, to choose which pages to __________.

5.8 Where can a Block be Placed?

- A _____ of Associativities is possible
- Advantage: _________ associativity _________ miss rates.
- Disadvantage: Increasing __________ increases ___ and _____ ____.



5.8 How is a Block Found?

- Cache
  - _Small_ degrees of associativity are used because _large_ degrees are _expensive_
- Virtual Memory
  - _Full_ _associativity_ makes sense because
    - Misses are _very_ _expensive_
    - _Software_ can implement _sophisticated_ replacement schemes
    - Full map can be easily _indexed_
    - _Large_ items means small number of _mappings_

5.8 Replace Which Block on a Cache Miss?

- Cache
  - __________
  - ____
- Virtual Memory
  - ____

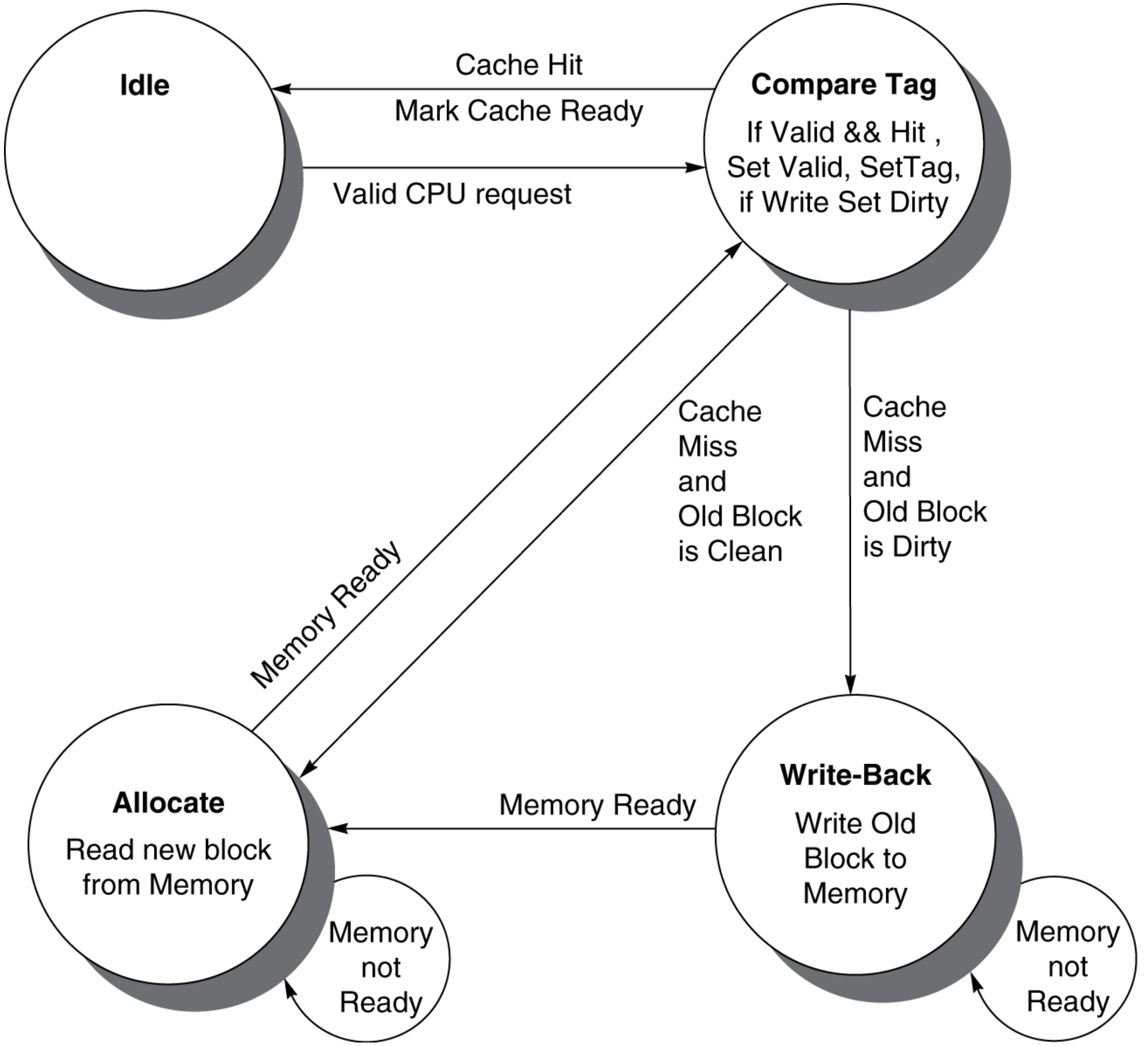
5.8 What Happens on a Write?

- Caches - write-back is the ________ of the ______
- Virtual memory – ______________ uses write-back

5.8 The Three Cs

- Cache misses occur in three categories:
  - Compulsory misses -
  - Capacity misses -
  - Conflict misses -
- The _________ in designing memory hierarchies is that every ________ that ___________ improves the ___ rate can also ___________ affect overall performance.

5.9 Finite State Cache Controller



5.10 Cache Coherence

- Multiple processors commonly _____ an _______ ______
- They may bring _____ into ______ and then write to their _______copies
- These local copies are likely
  - Different from ____ _____
  - Different from the _____ ______ value
- Example

Time step	Event	CPU A's cache	CPU B's cache	Memory
0				0
1	CPU A reads X	0		0
2	CPU B reads X	0	0	0
3	CPU A writes 1 to X	1	0	1

5.10 Snooping

- Every cache has the ________ _______ of the block along with the block
- The caches are all __________ via some ___________ mechanism (___ or ________).
- All cache controllers ________ or _______ on the medium to see whether or not they have a _____ of the ______ requested.
- The ________ CPU broadcasts an ___________ of the block.
- Example

CPU activity	Bus activity	CPU A's cache	CPU B's cache	Memory
				0
CPU A reads X	Cache miss for X	0		0
CPU B reads X	Cache miss for X	0	0	0
CPU A writes 1 to X	Invalidate for X	1		0
CPU B read X	Cache miss for X	1	1	1

5.16 Concluding Remarks

- The difficulty of building a memory system to keep pace with faster processors is underscored by the fact that the raw material for main memory, DRAMs, is essentially the same in all computers, fast or slow.
- Multilevel caches facilitate cache optimizations.
  - __________________
  - ____ _______________
- Other trends
  - ____
  - ____________