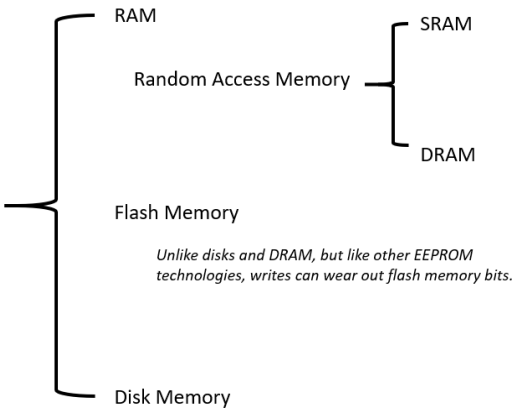
Memory Hierarchy

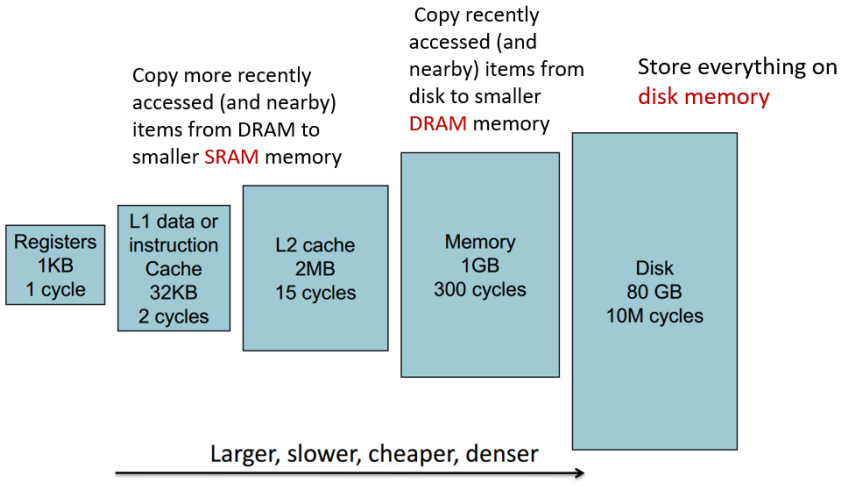
1. Memory Technologies

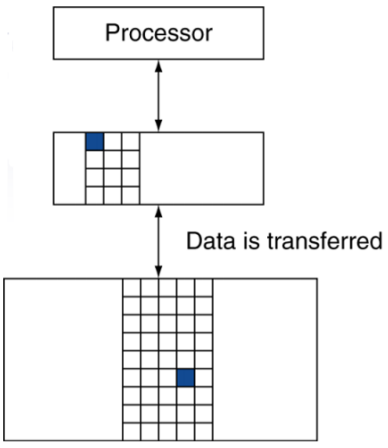


SRAMs don't need to refresh and so the access time is very close to the cycle time. SRAMs typically use six to eight transistors per bit to prevent the information from being disturbed when read. SRAM needs only minimal power to retain the charge in standby mode.

In a dynamic RAM (DRAM), the value kept in a cell is stored as a charge in a capacitor. A single transistor is then used to access this stored charge, either to read the value or to overwrite the charge stored there. Because DRAMs use only a single transistor per bit of storage, they are much denser and cheaper per bit than SRAM. As DRAMs store the charge on a capacitor, it cannot be kept indefinitely and must periodically be refreshed. That is why this memory structure is called dynamic, as opposed to the static storage in an SRAM cell.

2. Memory Hierarchy





Block (also called line): unit of copying

May be multiple words

The memory in upper level is originally empty

If accessed data is absent
- Miss: block copied from lower level
  - Time taken: miss penalty
  - Miss ratio: misses/accesses

If accessed data is present in upper level
- Hit: access satisfied by upper level
  - Hit ratio: hits/accesses = 1 – miss ratio
- Then accessed data supplied from upper level

3. Locality and source of misses

Locality

Temporal locality (locality in time): **if an item is referenced, it will tend to be referenced again soon.**

If you recently brought a book to your desk to look at, you will probably need to look at it again soon.

Spatial locality (locality in space): **if an item is referenced, items whose addresses are close by will tend to be referenced soon.**

For example, when you brought out the book on early English computers to fnd out about the EDSAC, you also noticed that there was another book shelved next to it about early mechanical computers, so you also brought back that book and, later on, found something useful in that book. Libraries put books on the same topic together on the same shelves to increase spatial locality.

Source of misses

Compulsory misses (aka cold start misses)

These are cache misses caused by the first access to a block that has never been in the cache.

Capacity misses

These are cache misses caused when the cache cannot contain all the blocks needed during execution of a program. Capacity misses occur when blocks are replaced and then later retrieved.

Conflict misses (aka collision misses)

These are cache misses that occur in set-associative or direct-mapped caches when multiple blocks compete for the same set. Confict misses are those misses in a direct-mapped or set-associative cache that are eliminated in a fully associative cache of the same size.

4. Calculate a cache size

32-bit address, direct mapped cache, n bit for index and block size is 2^m words, so m+2 bit for offset.

Size of tag field: 32-(n+m+2)

Size of cache: $2^n \times (block\ size + tag\ size + valid\ field\ size)$

$= 2^n \times (2^m \times 32 + (32 - n - m - 2) + 1)$

5. How to calculate block number and tag?

(1) Method 1

Transfer memory address to binary and compare with the format to get tag, index...

For example:

Tag	Index	Offset
31–10	9–5	4–0

$$132_{dec} = 00000000000000000000000\ 00100\ 00100_{binary}$$

Then we can get tag = 0, index = 4

$$3100_{dec} = 00000000000000000000011\ 00000\ 11100_{binary}$$

Then we can get tag=3, index = 0

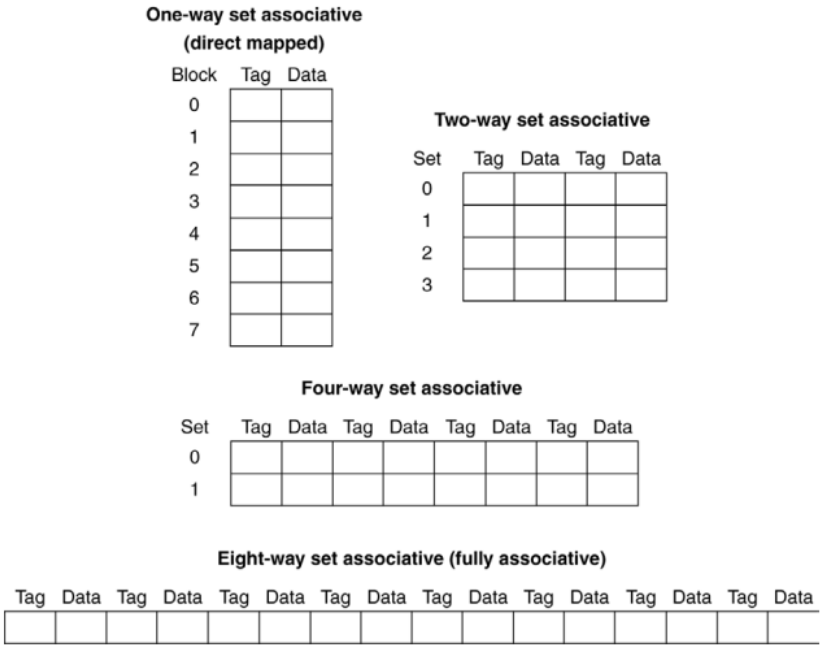
(2) Method 2

Suppose n bit for index, $number\ of\ entries = 2^n$

$$Tag = \left\lfloor \frac{byte\ address}{block\ size} \right\rfloor / (number\ of\ entries)$$

$$Index = \left\lfloor \frac{byte\ address}{block\ size} \right\rfloor mod\ (number\ of\ entries)$$

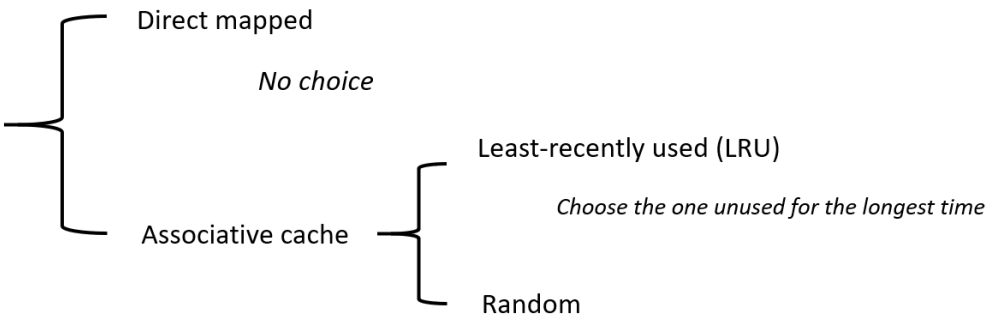
6. Set associative cache



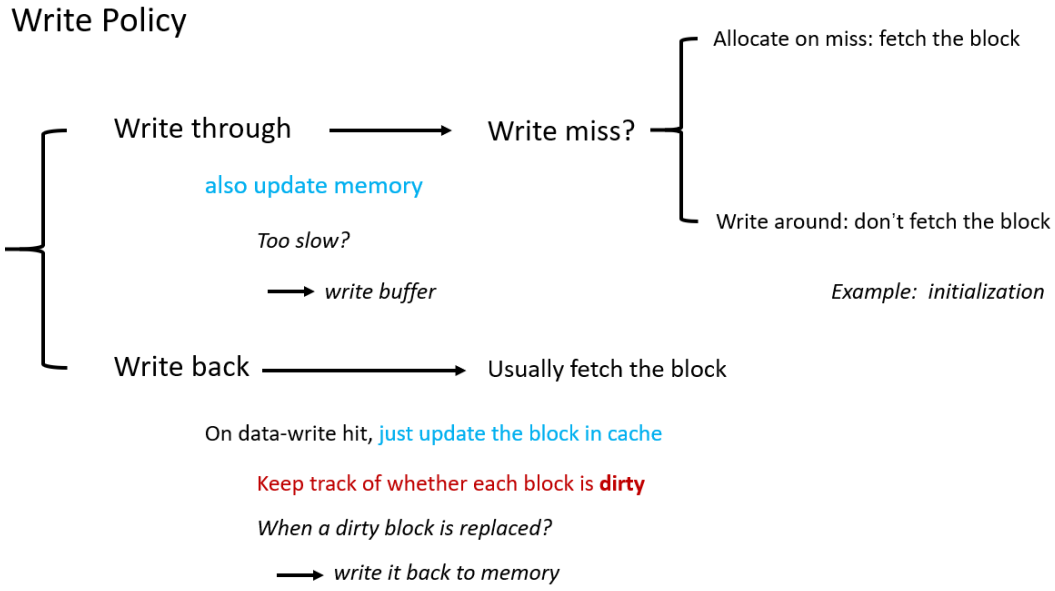
Example:

Increasing associativity requires more comparators and more tag bits per cache block. Assuming a cache of 4096 blocks, a 4-word block size, and a 32-bit address, find the total number of sets and the total number of tag bits for caches that are direct mapped, two-way and four-way set associative, and fully associative.

7. Replacement Policy



8. Write Policy



9. Cache Performance

$$Average\ Access\ Time\ (AMAT) = \boldsymbol{Hit\ time} + Miss\ rate \times Miss\ penalty$$

$$Memory\ Stall\ Cycles = \frac{Memory\ accesses}{Program} \times Miss\ rate \times Miss\ penalty$$

$$= \frac{Instrctions}{Program} \times \frac{Misses}{Instrction} \times Miss\ penalty$$

$$CPI_{stall} = CPI_{ideal} + average\ memory\ stall\ cycles$$

10. Dependable Memory Hierarchy

mean time to failure (MTTF) is a reliability measure.

Service interruption is measured as mean time to repair (MTTR).

Mean time between failures (MTBF) is simply the sum of ***MTTF + MTTR***.
Availability is then a measure of service accomplishment with respect to the alternation between the two states of accomplishment and interruption.

$$Availability = \frac{MTTF}{MTTF + MTTR}$$

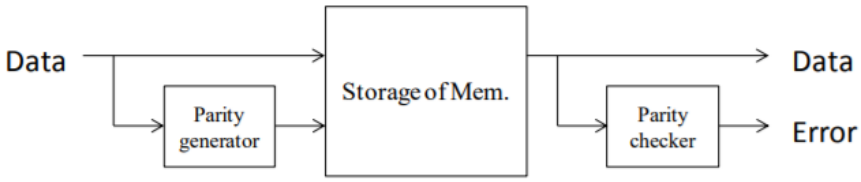
Three ways to improve MTTF:

1. *Fault avoidance*: Preventing fault occurrence by construction.
2. *Fault tolerance*: Using redundancy to allow the service to comply with the service specifcation despite faults occurring.
3. *Fault forecasting*: Predicting the presence and creation of faults, allowing the component to be replaced before it fails.

Three ways to reduce MTTR: *fault detection*, *fault diagnosis* and *fault repair*

11. Hamming code

Hamming distance: *Number of bits that are different between two bit patterns*.
For example, the distance between 011011 and 001111 is two.
In a **parity code**, the number of 1s in a word is counted; the word has odd parity if the number of 1s is odd and even otherwise. When a word is written into memory, the parity bit is also written. That is, the parity of the N+1 bit word should always be even.



Let p be total number of parity bits and d number of data bits in $p+d$ bit word.
p *parity bits can represent* 2^p states. Using 1 state to indicate "no error" and $2^p - 1$ states to indicate error in which bit.

$$2^p \geq p + d + 1$$

Add an additional parity bit for the whole word (p_n)
Decoding:
Let H = SEC parity bits
H even, p_n even, no error
H odd, p_n odd, correctable single bit error
H even, p_n odd, error in p_n bit
H odd, p_n even, double error occurred

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

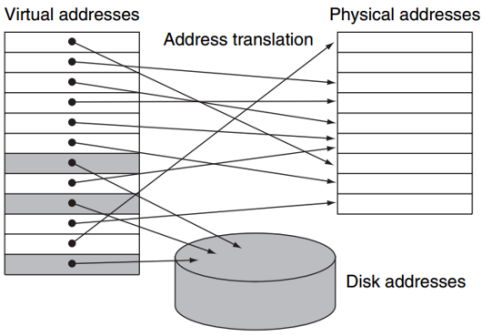
Example:
Assume one byte data value is 10011010. First show the Hamming ECC code for that byte, and then invert bit 10 and show that the ECC code finds and corrects the single bit error.

12. Virtual Memory

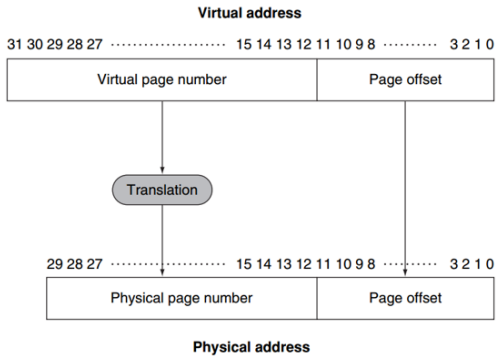
The main memory can act as a “cache” for the secondary storage, usually implemented with magnetic disks. This technique is called virtual memory.
Motivation:

1. To allow effcient and safe sharing of memory among multiple programs
2. To allow a single user program to exceed the size of primary memory.

A virtual memory block is called a **page**, and a virtual memory miss is called a **page fault**.



In virtual memory, blocks of memory (called pages) are mapped from one set of addresses (called virtual addresses) to another set (called physical addresses).



On page fault, the page must be fetched from disk.

Try to minimize page fault rate
1. Fully associative placement
2. Smart replacement algorithms

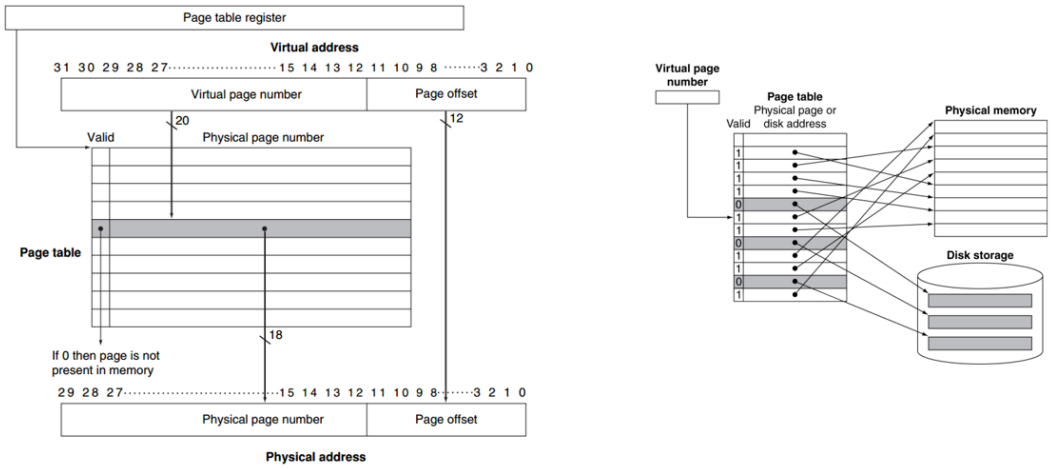
13.Page Tables

In virtual memory systems, we locate pages by using a table that indexes the memory; this structure is called a **page table**, and **it resides in memory**.

*Array of **page table entries (PTE)**, indexed by **virtual page number***
*To indicate the **location of the page table in memory**, the hardware includes a register that points to the start of the page table. → **page table register***

If page is present in memory
1. PTE stores the physical page number
2. Plus other status bits (referenced, dirty, …)

If page is not present
PTE can refer to location in swap space on disk



To reduce page fault rate, prefer ***least-recently used (LRU)*** replacement
- Reference bit (aka use bit) in PTE set to 1 on access to page
- Periodically cleared to 0 by OS
- A page with reference bit = 0 has not been used recently

Disk writes take millions of cycles
- Block at once, not individual locations
- Use **write-back**, because write through is impractical
- Dirty bit in PTE set when page is written

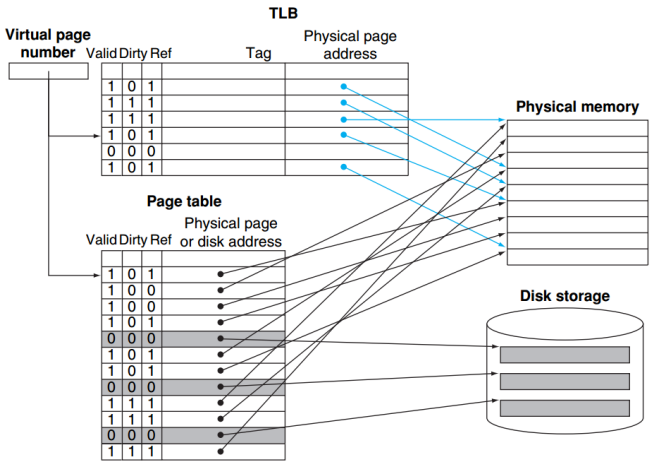
Example:
With a 32-bit virtual address, 4 KiB pages, and 4 bytes per page table entry, we can compute the total page table size:

14. Translation-lookaside buffer (TLB)

Since the page tables are stored in main memory, every memory access by a program can take at least twice as long: *one memory access to obtain the physical address* and *a second access to get the data*.

Modern processors include a special cache that keeps track of recently used translations. This special address translation cache is traditionally referred to as a translation-lookaside bufer (TLB)

- If page is in memory
  - Load the PTE from memory and retry
  - Could be handled in hardware
    - Can get complex for more complicated page table structures
  - Or in software
    - Raise a special exception, with optimized handler
- If page is not in memory (page fault)
  - OS handles fetching the page and updating the page table
  - Then restart the faulting instruction



The TLB acts as a cache of the page table for the entries that map to physical pages only.

15. Cache design Trade-offs

Design change	Effect on miss rate	Negative performance effect
Increase cache size	Decrease capacity misses	May increase access time
Increase associativity	Decrease conflict misses	May increase access time
Increase block size	Decrease compulsory misses	Increases miss penalty. For very large block size, may increase miss rate due to pollution.