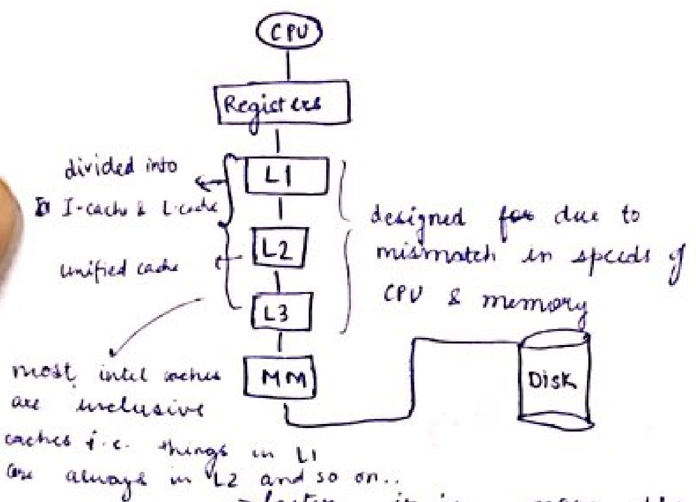
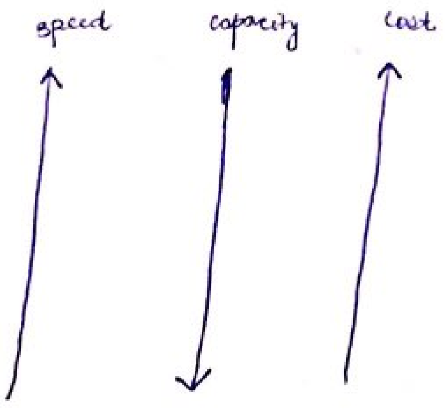
THE MEMORY HIERARCHY :





→ faster it is, more the cost per bit

Name	Regs	cache	MM	Disk
Size	< 1 KB	64 K/256 K/2-4 MB	4-64 GB	> 1 TB
Tech	CMOS	SRAM	DRAM	Magnetic
Access time	.15 - .3 nsec	.5/3-10/10-20 ns	30-200 ns	5 ms
manage by	compiler	Hardware	OS	OS
Backed by	Cache	MM	Disk	Tape

2^{31}

32 Regs.

4 Gigs

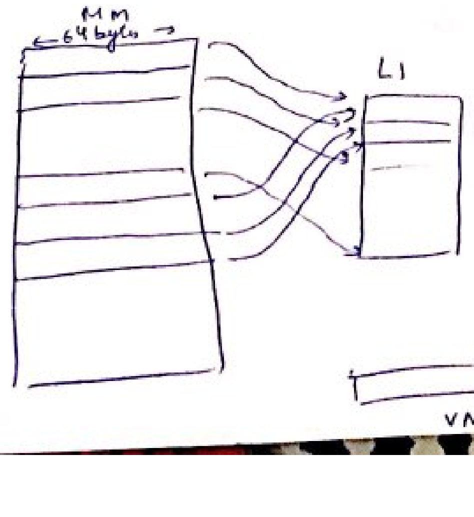
$2^{31} \times 32$

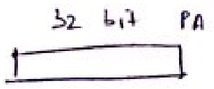
$10^9 \times 32$

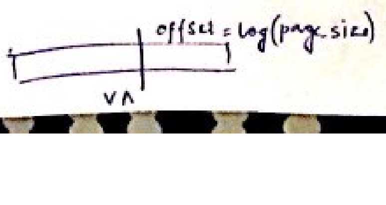
b = block or line (64 bytes on most Intel M/cs)

spatial locality → accessed something then high chance of accessing in the vicinity

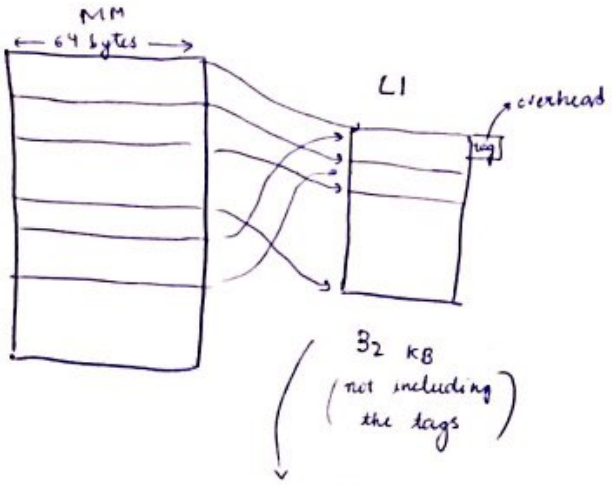
temporal locality → " " in near past " " in future

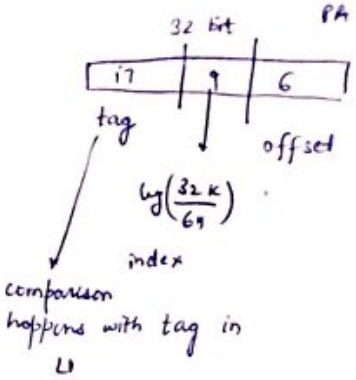


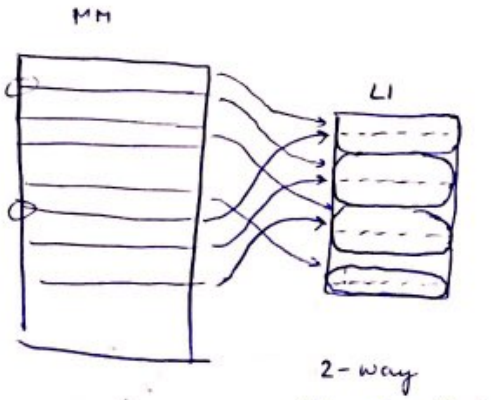


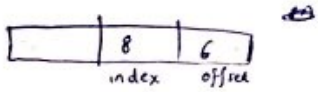


→ L1 is virtually addressed, L2 & L3 are physically addressed









- Assume CPI = 1 w/o mem. stalls
- 50% of all instr. are L/S
  Miss penalty = 25 cycles
  miss rate = 2%

So CPI after factoring mem stalls :-

Avg # of stalls per inst :

= 1.5 * 0.02 * 25 = 0.75

So CPI = 1.75

$\frac{1}{2} \times \frac{2}{100} \times 25$

$\frac{1}{2} \times 1 \rightarrow \frac{1}{2} \times \frac{2}{100} \times 25$

$\frac{1}{2} \times$

$\frac{2}{100} \rightarrow$ / mem

1.5 = average mem acc / Ins.

$\frac{1}{2} \cdot 1$

Design process :

1) C. size (cache)
2) Line Size
3) Associativity (every line can map to a set & the cardinality of the set is associativity)

4) # levels
5) Replacement policy
6) Write update policy
7) Write Alloc vs Not alloc
8) Split vs unified

$\log_2 512 \times 10^9$

~~3+3~~

9 + ~~3~~ (30)

9 × 3

64-( )	log(c/ba)	log b
tag	index	unified

c → cache size

b → block/line size

a → associativity

if 512 GB

↓

39 → 40

64 KB l → 4 KB

a - 8

$|tag| = 40 - 6 - \log \left| \frac{2^{12}}{2^3 \cdot 2^6} \right|$

= 27

Miss Rate

Avg mem access time

Exec. time.

IC × CPI × Cycle time → associativity will affect this since you will have to parallel search ~~f~~ in the set.

⇒ Avg mem access time = Hit time + MR (Miss rate) × Miss Penalty

Higher assoc → ~~High~~ ↓ MR (but at a=8 diminishing increase)

1-2 thumb rule → if cache size is 32K then decrease in MR if we
2-way
Inc cache size to 64K keeping a=1 or equivalently Inc associativity to 2 keeping cache size 32K.

→ Hit time = 1 cycle, Miss Penalty = 25 cycles

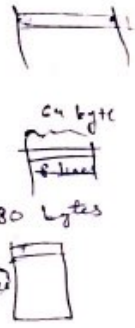
Assume 64 KB cache

Assoc	clock cycle time	MR	Mem Access Time
1	1	0.37	$1 + 0.37 \times 25 = 1.925$
2	1.36	0.31	$1.36 + 0.31 \times 25 = 2.135$
4	1.44	0.30	$1.44 + 0.30 \times 25 = 2.19$
8	1.52	0.29	$1.52 + 0.29 \times 25 = 2.245$

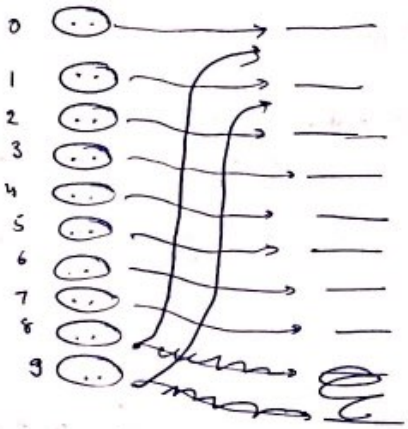
⇒ Eg:- toy cache = 64 bytes
line size = 8 bytes

What is the miss rate after executing 100 iterations of the loop, & each iteration has 20 instructions?

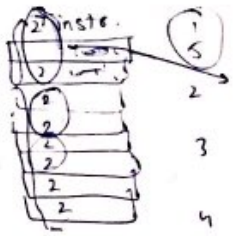
a = 1, LRU replacement policy

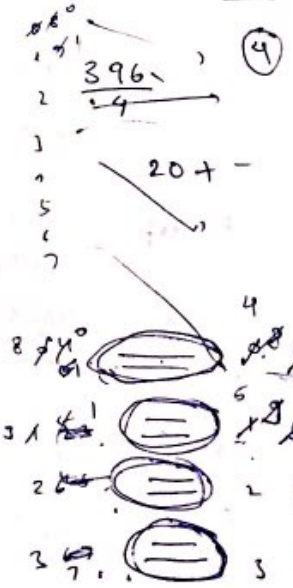


20 instr



406 = 4×99+10



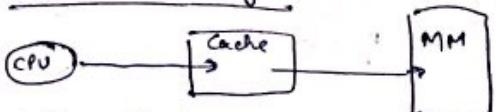


In 2-way (a=2) →

6×99+10

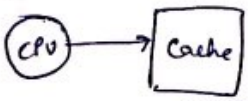
MM update policy:

Write through



WRITE BUFFER

Write Back



Strategy on a ~~Block~~ Write Miss

- Write Allocate
  Block is allocated on a write miss
- Write Non Allocate
  Block is not allocated

→

	W Alloc	No Alloc
W M[100]	M	M
W M[100]	H	M
R M[200]	M	M
W M[200]	H	H
R M[100]	H	M
W M[100]	H	H

Miss on a read always bring it.

Effectiveness of seperate i-cache & d-cache

16 KB I-cache, MR = $\frac{3.82}{1000 \text{ instr.}}$

16 KB D-cache, MR = $\frac{40.9}{1000 \text{ instr.}}$

VS

32 KB unified, MR = $\frac{43.3}{1000}$ inst

Assume 36% instr are L/S. Instr. Hit takes 1 cycle

Data " " 1 cycle with seperate D-cache

& 2 cycles with unified

MP = 200 cycles

What is the AMAT?

$1 \times \left(2 + \frac{43.3}{1000} \times 200 + \frac{36}{100} \times 1 \times 2\right)$

$\Rightarrow MR_{sep} = \frac{44.72}{1360} = 0.0329$

$MR_u = \frac{43.3}{1360} = 0.0318$

AMAT

Sep → $\frac{100}{136} \times \left(1 + \frac{3.82}{1000} \times 200\right) + \frac{36}{136} \times \left(1 + \frac{40.9}{0.36} \times \frac{200}{1000}\right) = 7.57$ cycles

$\Rightarrow$ Total cycles = $1000 + \frac{3.82}{1000} \times 200$

Unified →

$1 + 2 \times 0.36 + \frac{43.3}{1360} \times 200 = 8.08$ cycles

1.72

$1.36 + \frac{86.6}{10}$

$+ \frac{43.3}{1360} \times 200$

→

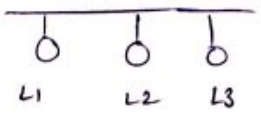
→

Loads	MR	MP	Access time
16	2.54	80 + 2	2.673
32	1.35	+ 4	2.139
64	1.06	+ 8	1.933
128	1.02	+ 16	1.98
256	1.15	+ 32	2.288

Projects

→ **Multiprocessor cache coherence:**

→



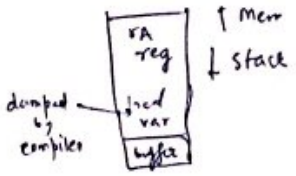
Matrix Multiplication

WB (Write Back)
WT (Write Through)

Domain Specific Archi.

→ Pattinson paper (in ML)
→ Jouppi (TPUs performance analyses)

Buffer overflow



Microarchitectural attacks

RSA , AES

↓
10 rounds - 16 table lookups & 16 xors

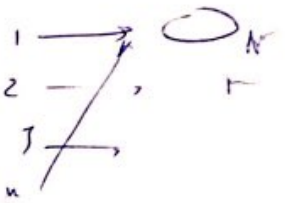
4K tables (4 tables of size 1K)

x — 1 class — x

AMD opteron case study :

TLB entry is 8 bytes
L1 hit time = 1 cycle
L2 hit time = 7 cycle
L1 TLB size → 40 PTEs
L2 " " → 512 "

→ Block repl ⇒ LRU (approx)
L1 → fully assoc
L2 → 4 way st.



L1-split I-cache & D-cache 64 KB
2-way SA, Hit Time = 3 cycles
LRU;

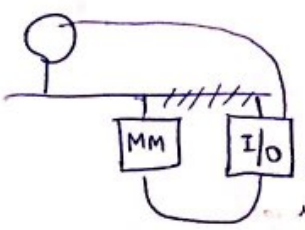
L2 - unified 512 KB
16-way SA
Hit time = 9 cycles

L3 - unified, 2 MB
32-way SA
Hit time = 38 cycles

I/O - related terms

- mem mapped I/O vs I/O instr
- control register / status reg.
- polling vs interrupt driven I/o

→ DMA (Direct mem Access)



when this guy is ready with disk info, it will place it in main memory

Reliability, Availability, MTTF, MTBF, MTTR

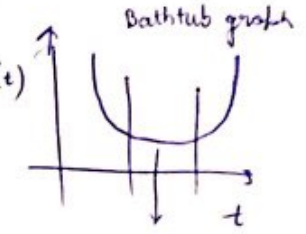
$R(t)$ = prob $\{$ S is functioning in $[0,t]$ $\}$

$N_0(t)$ of N components are operational at t

$N_f(t)$ of N components have failed in $[0,t]$

$$R(t) = \frac{N_0}{N}$$

$$\frac{dR}{dt} = \frac{-1}{N}\frac{dN_f}{dt}$$



λ remains more or less constant in this area.

→ $z(t)$ (failure rate or hazard rate) $= \frac{1}{N_0}\frac{dN_f}{dt} = \left(\frac{-\frac{dR}{dt}}{R}\right)$

If $z(t) = \lambda z$ (in start)

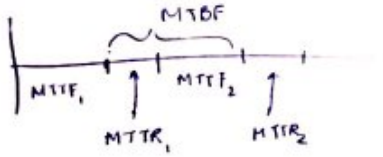
then $R(t) = e^{-\lambda t}$

→ Let x be a r.v. representing lifetime of a component & let F be cdf of x

Then $R(t) = Pr[x > t] = 1 - F(t) = \int_t^\infty f(x)dx = e^{-\lambda t}$ ($f(x) = \lambda e^{-\lambda x}$)

MTTF

$$MTTF = \int x f(x)dx = \int_0^\infty x\,\lambda e^{-\lambda x}dx = \frac{1}{\lambda}$$



Availability (uptime) $= \frac{MTTF}{MTTF + MTTR}$

AFR (Annual failure rate):

- % of devices expected to fail in a year for a given MTTF.
- 50K servers, each with 2 disks

MTTF = 1000000 hours

= 14 years

AFR. Annually how many disks will fail: $\frac{8760}{10^6} \times 10^5 = 876$ disks

2 disks failing everyday