

All these cant access @ once

Bus arbitration \rightarrow Technique to share the bus amongst different components

When I/O is done (mainly to main memory through I/O controller & mem controller)

DMA (not through some CPU)

Assumption till now

memory access time is negligible:

- \rightarrow Operands are in regs
- \rightarrow Operands are immediate

memory stall is zero

Not true, you can have delays in bus sw due to CPU relinquishing main memory

\Rightarrow CPU needs to wait.

Time required: Time to acquire system bus (α)

+ memory access time (locate memory cell) (β)

To ^{either} W/R one byte of data $\rightarrow \alpha + \beta$

$\Rightarrow k$ bytes of data $= k(\alpha + \beta)$ [which is quite a bit]

if $\alpha = 0.1 \text{ ns}$

$\beta = 3 \text{ ns}$

500 bytes $\Rightarrow 500 \times 3.1 \text{ ns}$ which isn't

[$\alpha \ll \beta$ CPU decides when to relinquish the bus, I/O must request CPU]

if \Rightarrow CPU normally has access, only in the worst case (as we're considering), α has some value.]

\Rightarrow Write/Read k bytes of data @ a time

$\Rightarrow T_{\text{time}} = \alpha + k\beta$

\hookrightarrow access system bus just once

A few questions

① Why do I read k bytes of data?

- (i) To reduce the system bus latency
- * (ii) Locality (??)

② Where do I keep ' k ' bytes of data?

Ans: Local memory in CPU, a.k.a cache.

Why is cache faster?

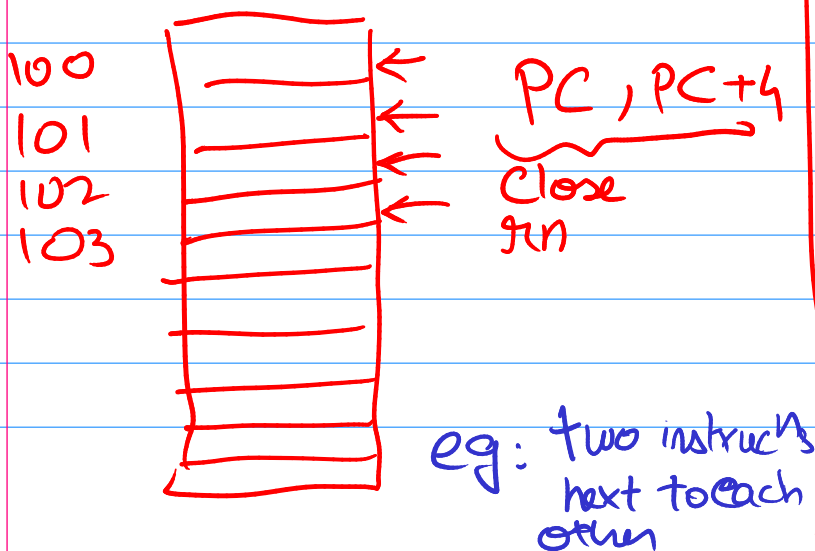
- (i) Technology it uses
- (ii) Closer to CPU

Locality of Reference: Spatial & Temporal

Spatial LDR

Access to memory locations occurring close together in space

Temporal LDR: Access to the memory location occurring close together in time



eg: jmp label

$x \rightarrow t$
 $y \rightarrow t+1$
 $\rightarrow x$ may not be $y-4$
but they are temporally close

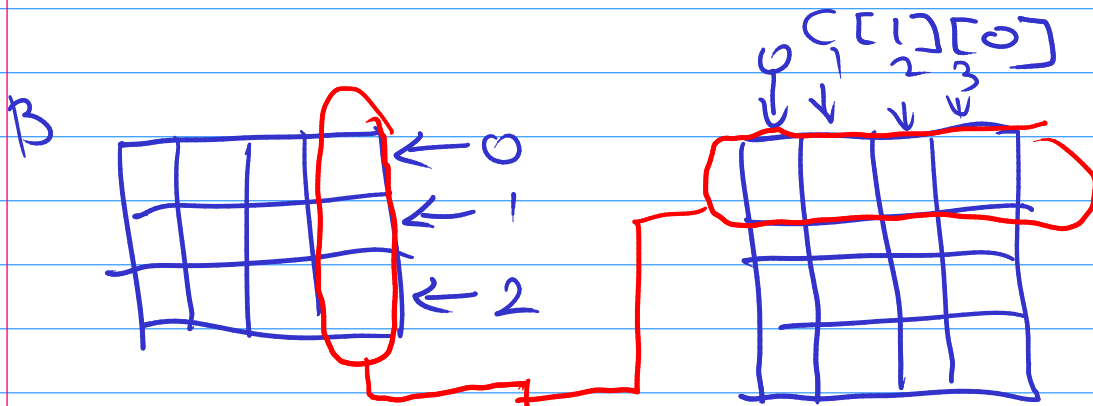
```
for (i=0; i < l; i++)
  for (j=0; j < m; j++)
```

```
    for (k=0; k < n; k++)
```

```
      A[i][j] = B[i][k] * C[k][j]
```

Spatial locality
C[0][0] then

Row major



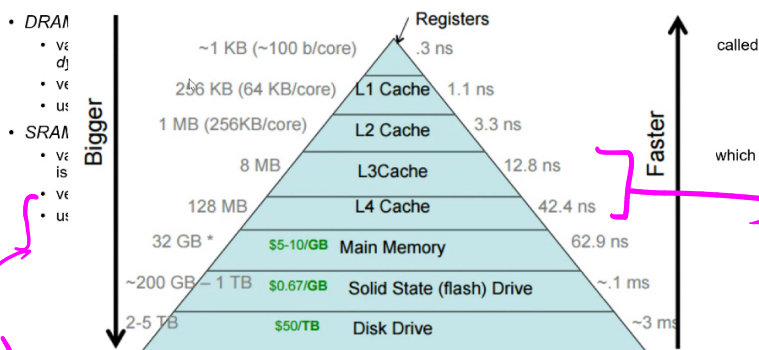
done @ RUNTIME
by OS & HW

Bring the entire row & column @ once

* \Rightarrow This is why we should read k bytes of data

$\alpha + k\beta$, on caching we replace β with γ

Memory Hierarchy



some arch's only

Animation lol, will try to take ss of this

Memory Wall Performance Gap

The Memory Wall

Performance Gap

