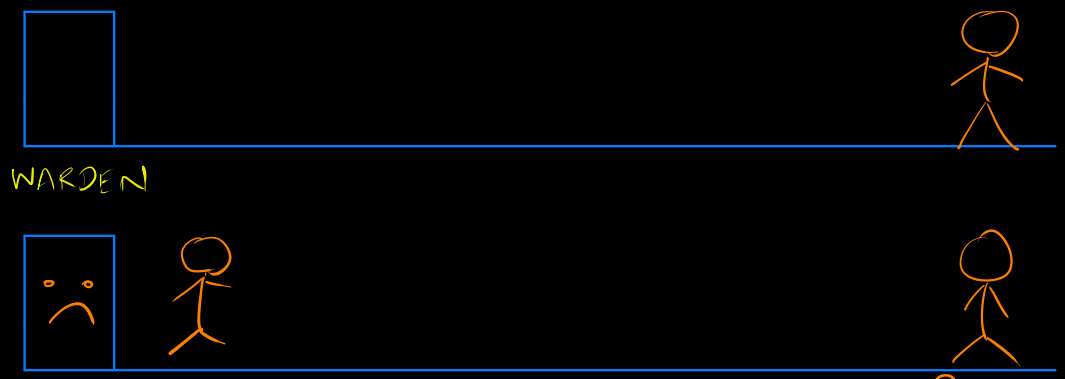
Memory System Performance

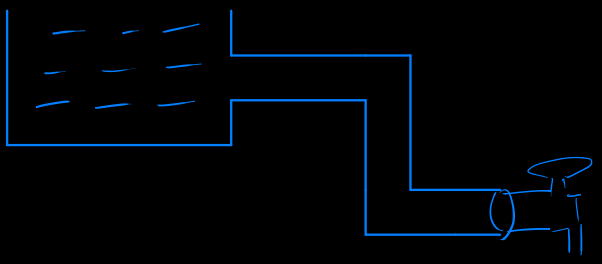


If Corridor is getting mopped ?

If Student is sleeping ?

If Corridor is only One-way ?

Water Tank to Tap



Higher the tank ?

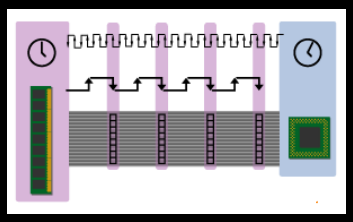
Wider the pipe ?

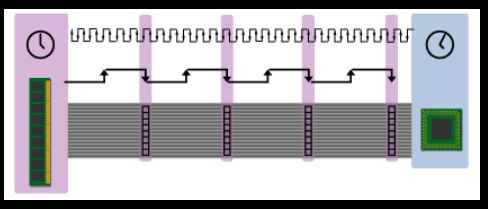
Pumping motor at top ?

Conveyor Belt

Faster the movement ?
(Quick Slots)

Larger slots ?





CPU: Request Control
Wait for grant
Send data
Relinquish Control

RAM: Receive Request
Prepare data
Request Control
Wait for grant
Send data
Relinquish Control

Time taken between CPU's request for data on the bus to its arrival on the bus to the CPU

LATENCY (in units of time)

```
struct Node{ int i;
             char c;
             } ; // 3 bytes
struct Node arr[10];
```

arr[0] + 6, 7, 8 → arr[1]

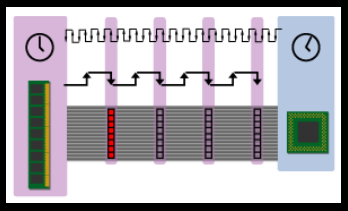
If latency = 3 bus cycles ⇒ 30 ns

CPU needs to minimum wait for 3 bus cycles
(Many CPU cycles)

0 - 7 - 1 Blk
8 - 15 -

1 Block = 8 Bytes

(ID) $\frac{100}{8} \times 8 = 96$



150th byte ⇒ RAM: 8 Bytes
Which 8 bytes?
96, 97, 98, 99, 100, 101, 102, 103

STRUCTURE PADDING

CPU: Ask a byte
→ (Should I ask for more?)

RAM: Sends the block containing the byte
→ (Should I predict and send more?)

Rate at which data can be moved between the CPU & RAM : BANDWIDTH (in units of bytes/sec)

100 MHz, 64-bit bus : At one cycle, 8 bytes sent

100×10^6 in a sec $\Rightarrow 8 \times 100 \times 10^6$ bytes/s

800 MB/s

133 MHz, 32-bit bus : At one cycle, 4 bytes

133×10^6 in a sec $\Rightarrow$ 532 MB/s

Narrow Corridor Vs Wide Corridor

One Corridor Vs Multiple Corridors

32-bit Vs 64-bit

Single Block Vs Multiple Blocks

Bus Width $\nearrow$ Bandwidth ✓ [TRAFFIC FLOW]

$\rightarrow$ Latency ✗

Memory blocks $\nearrow$ Bandwidth ✓

$\rightarrow$ Latency ✗

Technology $\nearrow$ Bandwidth ✓

$\rightarrow$ Latency ✓

If Latency of an interconnect is l seconds and its bandwidth is b bytes/sec, then time taken to transmit n bytes

= First 8 $\rightarrow$ l sec, Next 8 $\rightarrow$ l + one bus cycle

$= l + (n-8)/8 \times$ one bus cycle

$= l + n/b$

100 bytes : (0-7) 8 bytes $\xrightarrow{0^{th}\ bc}$ l sec

(8-15) $\xrightarrow{1^{st}\ bc}$ 1 bc $(l+1)$

$\frac{92}{8} = \frac{11}{}$ (16-23) $\xrightarrow{2^{nd}\ bc}$ 2 bc $(l+2)$

$\vdots$

(80-87) $\xrightarrow{10^{th}}$ $l+10$

Time taken (88-96) $\xrightarrow{11^{th}}$ $l+11$

$\approx$ $l + \frac{100-8}{8} \times \frac{1\ bc}{}$

$$\boxed{l + \frac{n}{b}}$$

Processor operating at 1 GHz ⇒ CPU cycle = 1 ns

DRAM latency 100 ns

No. of FMA units = 2

FMA operation completion = 1 cycle

4 FLOPs in 1 cycle

4×10^9 FLOPS $= 4$ GFLOPS

Peak performance = 4 GFLOPS (FLOPS Vs FLOPs)

(Peak Processor Rating)

Block = 1 word = 4 Bytes

CPU $\xrightarrow{\text{REQ 1}}$ RAM $\xrightarrow{\text{BLK 1}}$ CPU $\xrightarrow{\text{REQ 2}}$ RAM $\xrightarrow{\text{BLK 2}}$ ...

⇒ STALLS ⇒ STALLS →

```
float a[100], b[100], ans = 0.0f;
int i = 0;
// a & b are set here
for(i = 0; i < 100; i++)
    ans = ans + a[i] * b[i];
```

a[0] dot

How many FLOPS can this program achieve?
or what is the peak speed of this algorithm?

Fetch a[0] : 100 ns Fetch b[0] : 100 ns

= 200 ns

After 200 ns, 2 FLOPs can be done

400 ns, 4 " " " = 200 ns 2

$\frac{10^7}{4 \times 10^9} \times 100$ 1 SEC, ?. ? ⇒ $2 \times 5 \times 10^6$ $\times ? = 1$ s ?

$= 0.25$ s.

$= 10$ MFLOPS $= 10^7$ //

$\frac{10^9}{200} = \frac{1}{2} \times 10^7$

$= 5 \times 10^6$

1 Block = 4 words (each word = 4 bytes)

Does the peak speed change? 1 RAM block = 16 Bytes

↓ Bandwidth = 16 Bytes

Fetch a[0] : 100 ns ⇒ fetches a[1], a[2] & a[3]

Fetch b[0] : 100 ns ⇒ " b[1], b[2] & b[3]

In 200 ns : Compute using a[0] & b[0] : 2 FLOPs

a[1] & b[1] : 2

a[2] & b[2] : 2

200 ns : 8 FLOPs a[3] & b[3] : 2

8 FLOPs

1 s : ?

$8 \times 5 \times 10^6 = 40$ MFLOPS