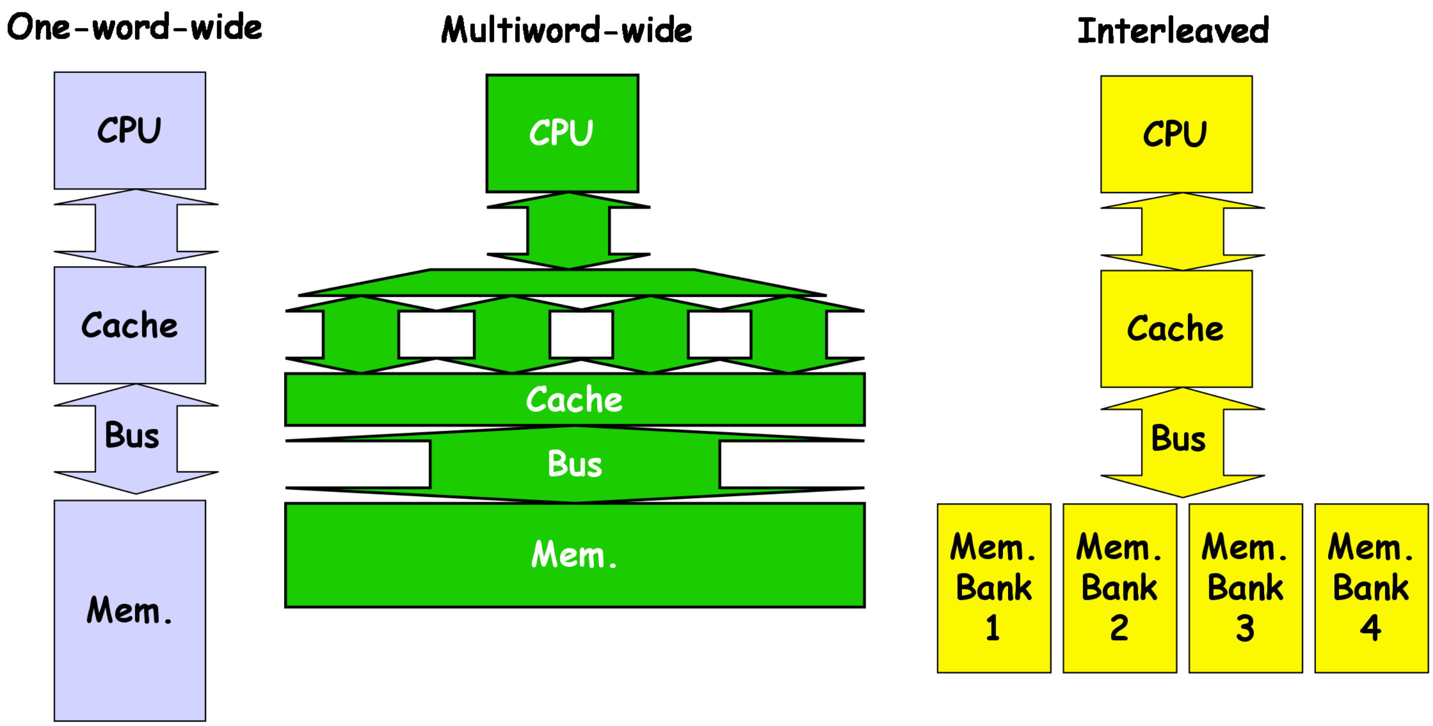
Board Notes on Memory Organization and Disk I/O

Part A: Memory Organization

Main memory can be organized in several different ways (see picture below)



Part A:

Given:

- A cache block size of 4 words
- 1 CC to send and address to memory (e.g. time on bus)
- 10 CCs for each DRAM access initiated
- 1 CC to send a word of data on the bus

For a one-word-wide configuration, what is the miss penalty? How many bytes are transferred per CC?

- Miss penalty:
  - 1 CC to send address + number of accesses x (access time + transfer time)
    - Transfer time is # of times bus is used
  - Therefore:
    - 1 + 4 x (10 + 1)
    - 1 + 4 x (11)
    - 45
- Bytes / CC
  - 4 words x (4 bytes / word) / 45 CCs = 0.36 Bytes / CC

Part B:

Now, if we use wider memory/cache, also need to increase the bandwidth between levels – otherwise, the wider memory/cache is not much good! For this example, let's add BW and assume the following:

- Main memory is 4 words wide
- We have the same cache/memory access times as before.

What is the miss penalty? How many bytes are transferred per clock cycle?

- Miss penalty:
  - 1 CC to send address + number of accesses x (access time + transfer time)
    - Transfer time is # of times bus is used
  - Therefore:
    - 1 + 1 x (10 + 1)
    - 1 + 1 x (11)
    - 12

- Bytes / CC
  - 4 words x (4 bytes / word) / 12 CCs = 1.33 Bytes / CC
  - 4x better, but 4x more HW too.

Part C:

With multiple DRAM chips, we could *interleave* them to get some parallelism and organize the data such that parallel reads are in fact practical. Further, assume the following:

- The bandwidth between the cache and main memory is the same as in Part A
- DRAM banks are one word wide
- By sending an address, all 4 banks can be addressed simultaneously, but data must be sent back serially.

What is the miss penalty? How many bytes are transferred per clock cycle?

- Miss penalty:
  - Send time + access time x transfer time
  - 1 + 10 x (4 x 1)
  - 15 CCs

- Bytes / CC
  - 4 words x (4 bytes / word) / 15 CCs = 1.1 Bytes / CC
  - 12 vs. 15 – for 75% less bandwidth!

Part B: Disk Seek Time

Part A:

A disk has the following parameters:

- 3600 RPM (= 60 Rotations per second – may help to think in terms of tracks per second)
- Average seek time = 9 ms
- 100 sectors per tack, 512 bytes per sector
- Controller + queuing delays = 1 ms

What is the average time to read 1 sector (512 Bytes)?

Rate_transfer = 100 sectors/track * 512 Bytes/sector * 60 RPS = 2.4 MB/s

T_transfer = 512 B / 2.4 MB/s = 0.2 ms

T_rotation = 0.5 / 60 RPS = 8.3 ms

(on average, disk must rotate ½ way around)

T_disk = 9 ms (seek) + 8.3 ms (rotation) + 0.2 ms (transfer) + 1 ms (controller) = 18.5 ns

Notes:

- T_transfer only small part! What does this suggest?
  - Bring in lots of information at a time so you don't have to pay mechanical overhead
- Also, t_queuing can get worse with more requests pending.

Part B:

A disk drive has 15 platters. Each platter has 2 surfaces. The drive has 250 cylinders. Each track has 256 sectors. Each sector has 64 bytes. The average seek time is 3 ms.

What is the average disk access time if we want to transfer 4,096 bytes (i.e. 4KB) of data, our disk rotates at 15,000 RPM, and there is a controller overhead of 2 ms? **Be exact. Do not round.**

Answer:

Transfer Rate =

0.25 rotations / ms * 256 sectors / track * 64 bytes / sector = 4096 bytes/ms

Average Disk Access Time =

3 ms (for average seek time) +

.5 / [15000 RPM * (1 min / 60s) * (1 s / 1000 ms)]$ +

4096 bytes / 4096 bytes / ms+

2 (controller overhead)

Average Disk Access Time =

3 + 0.125 ms + 1 ms + 2 ms

Average Disk Access Time =

6.125 ms

Part C: Disk Seek Time

The test sequence is: 1011, 57, 272, 40, 512, 717, 320
The disk head starts at 117.

Part A:
What is the total distance the disk head moves with the Shortest Seek Time First algorithm?

Answer: 117 → 57 → 40 → 272 → 320 → 512 → 717 → 1011
(117-57) + (57-40) + (272-40) + (320-272) + (512-320) + (717-512) + (1011-717)
= 1048

Part B:
There is a potential problem with using the Shortest Seek Time First (SSTF) algorithm. In 10 words or less, describe it.

Answer: Starvation