CS-1216 – Monsoon 2023 – Assignment 4

Rutam Kathale,

November 20, 2023

1. The logical address space in a computer system consists of 128 segments of capacity 32 pages of 4K words. The physical memory consists of 4K page frames, each of 4K word capacity. What are the logical and physical address formats? Represent in binary the logical address for segment 35 and word number 1999 in page 16.

We first calculate all the # of bits required for components in the system.
For segment number we need log2(128) = 7bits to represent the segment number.
For page number, we need log2(32) = 5 bits.
Having 4K words, we need log2(4096) = 12 bits to represent the word number. The calculation for page frame number will be same as bits for word number since we have 4K page frames.
Bits for logical address would be sum of page number, word number, and segment number => 5 + 12 + 7 = 24 bits.
We similarly would have physical address format as sum of bits for page frame number and word number => 12 + 12 = 24 bits.

Now we need representation of Segment 35, Page 16, Word 1999. We can simply convert these into binary based on total bits available for each component and join them to get our logical representation.
35 in 7 bit binary = 0100011
16 in 5 bits binary = 10000
1999 in 12 bit binary = 011111001111
APpending these, we get the address as 010001110000011111001111 in 24 bit binary.
NOTE: I have considered leading zeroes considering bits, so if we were to not use that, the solution would be slightly different (10001110000111110011111).

2. A disk subsystem employed for the virtual memory organization has the following specifications: 32 sectors per track, 512 bytes per sector, 3600 RPM, average seek time of 30 msec, average instruction execution time is 1.5 microsec and page size of 4 Kbyte. In the event of page fault, compute the wastage of instruction execution capability of CPU before execution can continue the program. What will be the impact if the average seek time is reduced to 25 msec by employing an improved disk drive?

Total time the CPU spends waiting for the page to be loaded from the disk to memory that is the waiting time, will be a reliable metric to check wastage of instructions and if it would change due to seek time. We can consider seek time, rotational delay, and data transfer time.
We have average seek time currently as 30ms.
We have 3600 revolutions per minute that is, 6/360 (seconds) to complete a revolution = 16.7 ms approx. Since we do not have an idea of the sector when the seek completes, we can assume it to be halfway around from r-w head.
So, our rotational delay would be about 8.3 ms.
And since we have page size of 4Kb and 512 bytes per sector, a page will have about 4000/512 = 7.8 that is approximately 8 sectors.
So since we have 32 sectors per track, time to read 1 sector would be time for 1 rotation (8.3 ∗ 2 approx 16.6 msec), divided by our sectors per track => 16.7/32 approx 0.520 ms.
That and we have 8 sectors in total => 8∗0.520 = 4.16ms.
So our total waiting time would be: 30 + 8.3 + 4.16 = approx 42.46 that is, 42.50 ms.
Given this, since we have average instruction execution time, number of instructions executed in our waiting time would be 42500/1.5 that is aprox 28,333 instructions.
With improved seek time, total waiting time 25 + 8.3 + 4.16 = about 37.46 ms.
Total instructions for this time would be 37460/1.5 = about 25000 instructions.

So, improving seek time definitely reduces wastage of instructions.
NOTE: Since I have rounded off values at every point, my solution might not be exact, but it would follow the trend of saving wastage of instructions on improving seek time.

3. A disk pack has 19 surfaces and the storage area on each of them has an inner diameter of 22 cm and an outer diameter of 33 cm. The maximum storage density on any track is 2000 bit/cm and the minimum spacing between tracks is 0.25 mm. (a) What is the storage capacity of the pack? (b) What is the data transfer rate in bytes per sec at a rotational speed of 3600 RPM. (c) Using two 16-bit words, suggest a suitable scheme for specifying disk address

(a) Lets assume that each track has a uniform width = 0.25 mm. Let's also assume CAV i.e. storage density is constant in all tracks.
To calculate storage capacity of the pack, we need stoage capacity of each surface which will require bits per track (which will require storage density and track length(which requires width of each track)) and number of tracks.
First, width of track is 0.25mm (Given).
Total width would be outer radius - inner radius = 11cm/2 = 5.5cm.
Then the number of tracks would be 5.5 / 0.025 (spacing) = 220.
So we have 220 tracks.
Length of rach track would be average circumference which by fomula would be $\pi *$ average radius = π ((33+22)/2) = 86.3 cm.
Based on this, storage capacity in bits for each track would be length x max storage density bits = 86.3 $*$ 2000 $\approx$ 172600.
Then we have storage capacity of a surface which would be product of storage capacity in bits for each tracks and total tracks we have = 220 x 172600 = 37972000 bits.
So, now going one level up, we have 19 surfaces and we'll multiply with capacity for each surface => 19 * 37972000 bits = 721468000 bits or 90183500 bytes that is, about 90 MB.

(b) Now we have rotational speed as 3600 RPM, that is 60 revolutions per second.
We can get the tangential speed by multiplying these revolutions by circumference of outermost track that is,
2 x π x (0.33) x 60 $\approx$ 6220.35
Now that we have the tangential speed, we can get the transfer speed by taking product of tangential speed with storage density that is 2000.
We get 6200.35 x 2000 = 1240070 bytes per second (1.27 MB/s).
(c) To specify disc address, we should be able to identify surface number, track number, and sector number. Since we have 19 surfaces, we need to have bits more than 19 to represent surface number. So we can take $2^5 = 32$, that is 5 bits. We have 220 tracks per surface, so we can use 8 bits to represent tracks similarly. So out of two 16-bit words, first word can be used for these (leaving 3 bits.).
In second bit number, we can use all of the bits (apart from 5 surface bits) to address sectors per track. This way, we can specify any address within disk.

4. Consider again the parameters of problem 3 above. Suppose that the main memory of the computer has 32-bit word and 500 nanosec cycle time. Assuming that the disk transfers data to/from the main memory on a cycle stealing basis, evaluate the percentage of memory cycles stolen during data transfer period.

We have data transfer rate about 1.27 MB/s. So number of memory cycles per second will be inverse of 500 ns which is 0.0000005.
We can get data transferred per cycle by dividing our data rate by these memory cycles and get the stolen cycles by multiplying it with our cycle time and take percentage. We get data transferred per memory cycles to be 0.7 bytes and so, the stolen cycles comes out to be about 0.0031%.

5. Assume the following specifications for a system – 500 nanosec memory cycle time for read/write, average of 2 microsec for execution of an instruction. Determine the peak and average data transfer rate for each of the following

(a) For interrupt-driven I/O, we execute 6 instructions per transfer. Since we have average execution time as 2 microseconds, per transfer we need 12 microseconds by executing 6 instructions. So, since transfer is of one byte, data transfer rate = data transferred/time per transfer = 1/(12 x 1/1024) = 83.3 KB/s.

(b) For programmed I/O, each transfer executes the 4 CPU instruction sequence. So, we take 8 microsec per transfer. Data transfer rate is 1 byte / 8 microseconds = 125 KB/sec.

(c) For DMA, in block transfer mode, time to transfer one byte is 500 nanosec since there's no involvement of instructions/CPU. So, the data transfer rate is 1 byte / 500 nanosec = 2 MB/sec.
In cycle stealing mode, the time to transfer one byte is 2 x 500 nanosec (since half of memory cycles use memory bus) = 1 microsec. Therefore, the data transfer rate is 1 byte / 1 microsec = 1 MB/sec.