TUTORIAL #8	Computer Memory Types and Usage, Reliability and Redundancy (**Solutions**) ***CONFIDENTIAL***	(CE/CZ1006)

8.1 Computer Memory Types and Usage

1. What memory type would you use to implement the cache in Processor Cx1006-200M16 in the case study notes? Explain your choice.

[Suggested Solution]

SRAM. Cache needs to be implemented with very fast memory so as to keep up with the processor operating speed. DRAM is too slow and its transfer procedure too complex to be used for cache operation. Flash is not suitable due to its finite erase cycles and slower speed.

2. Name two main types of Flash memory available in the market. What are the differences between them? Which application/product areas are they used in?

[Suggested Solution]
NAND and NOR Flash Memory.
NAND:

- Data is accessed a page at a time. Command used to open a particular page followed by the individual bytes read/write. Does not support Execute in Place.
- More compact due to layout of the cells => more cost effective.
- Used mainly as storage memory. E.g. SSD, USB Flash Drive.

NOR:

- Allows Random Data Read by providing address information only.
- Supports Execute in Place. i.e. Code stored in NOR Flash can be executed directly without having to transfer to RAM first.
- Less compact than NAND flash.
- Used mainly as system memory to store program code.

3. Reference the two HDDs listed in the case study notes (HDD001 and HDD002)

   a. What is the **capacity** of each drive?

   [Suggested Solution]

   HDD001:

   Capacity = 4*1024*128*512 = 268, 435, 456 Bytes = 256Mbytes.
   Note: 1Mbyte = 2^20 bytes

   HDD002:

   Capacity = 8*1024*256*512 = 1, 073, 741, 824 bytes = 1GByte.

   Note:

   1GByte = 2^30 bytes,

# of cylinders = # of tracks of each surface,

# of heads = # of valid surfaces

b. For HDD001,

i. What is its **access time**?

[Suggested Solution]

RPS = 5000/60

Average rotational delay = $T_{R,AV}$ = 0.5/RPS = 0.5/(5000/60) = 6ms

Seek time = T_S = 5ms

Access time, T_A = T_S + $T_{R,AV}$ = 5 + 6 = 11ms

ii. What is the time needed to transfer a 4Kbyte file stored in random non-consecutive sectors on different tracks? Assume that every sector is on a different track.

[Suggested solution]
RPS = 5000/60 per second
Access time T_A = 11 ms
# of sectors = 4*1024/512 = 8
Transfer time T_T for 1 sector = 512/(RPS*D_T*D_S) = 512/((5000/60)*128*512)
= 93.75 us
Total time, T_{TOTAL} = 8*(T_A + T_T) = 88.75 ms

Note: This illustrate the effect for HDD fragmentation. If a file is fragmented, i.e. stored in non-consecutive sectors/clusters, the effective performance of the HDD data transfer will drop, as shown by the total access time for the two examples above.

iii. After defragmenting HDD001, what would be the time needed to transfer a 280Kbyte file?

[Suggested solution]
Defragmentation => file is stored in consecutive sectors.

However, 280KByte is large than size of one track (128*512=64KByte).

It occupies 4 full tracks and 24 Kbyte on the fifth track.

RPS = 5000/60 per second
Access time T_A = 11 ms

T_T for one full track = 1/RPS = 12 ms

Total time for 1 track, T_{TOTAL} = 11+12 ms = 23 ms

4 tracks = 4*23 = 92ms

Last 24KByte:
Transfer time T_T = N/(RPS*DT*DS) = (24*1024)/((5000/60)*128*512) = 4.5 ms
Total time, T_{TOTAL} = 11+4.5ms = 15.5 ms

Total time needed to transfer 280KByte file = 92+15.5 = 107.5 ms

Note: T_A is required when moving from one track to the other since the R/W head need to be positioned at the starting sector of the next track.

c. If you are building a Network Access Storage for your home to act as a backup storage for your home's computers, which HDD would you choose? Justify your choice.

[Suggested Solution]
HDD002.
- Statistically more robust as MTTF = 1M hours, which is 2X that of HDD001. So less likely to fail.
- Higher storage
- Higher Performance (higher RPM, shorter Seek Time)
- HDD drives for NAS is usually more expensive (per bit wise) than typical consumer drives.

d. Would you use a SSD instead for Q3(c) above? Since SSD is more robust than HDD and robustness is very important for backup storage.

[Suggested Solution]
HDD.
- SSD is still has a higher cost per bit than HDD. Not ideal for application that requires huge storage, especially for consumer home use.
- Finite read/write cycles of SSD compared to HDD's almost infinite read/write cycles.
- SSD cannot replace HDD in backup storage area, including data centers and servers, yet. This primarily due to cost.

4. What would be the memory choices for the system and storage memory for each scenarios below? Justify your memory choice selection in terms of functionality, performance and cost.
   a. Entry level Microsoft Windows desktop computer for general office use but needs huge data storage capacity to store videos relating to the company product.
   b. Ultra-thin Android Tablet with long battery life.

[Suggested Solutions]

a. System Memory: DRAM, Storage Memory: Magnetic HDD
   - Entry level Desktop, General office use => low cost and only need average performance => DRAM and HDD as cost is lower then SSD
   - Windows OS typically requires a few GByte of system memory. At this size, SRAM will be too expensive. Higher end product uses higher speed DRAM such as DDR3, DDR4 etc
   - Large storage for videos => Magnetic HDD (lower cost per bit than SSD)

b. System Memory: DRAM, Storage Memory: NAND Flash
   - Tablet => thinner form factor and low power consumption
   - Use of NAND flash for both power consumption and small form factor
   - Similar to Windows OS, Android needs at least 0.5 GByte of System memory. So DRAM used for cost reason, preferably low power DRAM for lower power consumption

8.2 Reliability

5. Reference the case study notes,

   a. What is the probability that the HDD is still functioning after 1 year? Compute for HDD001and HDD002.

[Suggested solution]

HDD001:

MTTF = 500,000 hours

t = 24*365 = 8760 hours

$$R(t) = e^{-\frac{t}{MTTF}} = 0.983$$

HDD002:

MTTF = 1,000,000 hours

t = 24*365 = 8760 hours

$$R(t) = e^{-\frac{t}{MTTF}} = 0.991$$

*Note: Manufacturer typically derived MTTF value by testing many devices instead of one device, the result is then divided by the number of devices under test. So MTTF should strictly have a unit of device*hours instead of hours.*

   b. Would HDD001 definitely fail before HDD02?

[Suggested Solution]

No. MTTF is just a statistical estimation.

8.3 Redundancy

6. Various RAID configurations have been described in the lecture.
   a. Which RAID configuration(s) cannot tolerate any disk failure? Explain. Under what situation do we use this RAID configuration?

[Suggested solution]

RAID-0 → No redundancy
RAID-0 gives the best data transfer performance as data is accessed simultaneously across two or more HDDs.

   b. Which RAID configuration(s) can only tolerate one single disk failure? Explain.

[Suggested solution]

RAID-1 and RAID-5 → RAID-1 uses 100% redundancy (mirroring). RAID-5 uses parity information to recover from failed disk.

   c. Which RAID configuration(s) is able to tolerate more than one disk failure? Explain.

[Suggested solution]

RAID-6. Has double distributed parity, i.e. each data block has two sets of parity data which it could depend on to reconstruct the data. So data could be recovered even if two disks failed.

(Not necessary to be covered during tutorial)

7. Why does the Processor Cx1006-200M16 has two different types of non-volatile memory (Flash and EEPROM) on-chip?

[Suggested Solution]

Flash has a larger page size (order of Kbytes) so can only be erased at Kbytes level. This is not suitable for certain use case such as storing of user configuration parameters, which is typically done at order of bytes level. An EEPROM, which has a smaller page size (tens of bytes) is more suitable for such use case as it result in less page erases. Also, EEPROM has a larger erase cycle endurance i.e. it could tolerate larger number of erasures before failing.