Semiconductor Memory

1. The two tables below show a sample of commands available when interfacing to NAND flash (Table 1) and NOR flash (Table 2). What can you deduce from the nature of the Read/Program commands supported by the NAND and NOR Flash?

Table 1: NAND Flash Commands

Operation	Cycle 1	Cycle 2	Valid During Busy
PAGE READ	00h	30h	No
PAGE READ CACHE MODE START[1]	31h	–	No
PAGE READ CACHE MODE START LAST[1]	3Fh	–	No
READ for INTERNAL DATA MOVE[2]	00h	35h	No
RANDOM DATA READ[3]	05h	E0h	No
READ ID	90h	–	No
READ STATUS	70h	–	Yes
PROGRAM PAGE	80h	10h	No
PROGRAM PAGE CACHE[1]	80h	15h	No
PROGRAM for INTERNAL DATA MOVE[2]	85h	10h	No
RANDOM DATA INPUT for PROGRAM [4]	85h	–	No
BLOCK ERASE	60h	D0h	No
RESET	FFh	–	Yes

Table 2: NOR Flash Commands

		WRITE Operations											
		1st		2nd		3rd		4th		5th		6th	
Command	Length	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data	Addr	Data
READ/RESET	1	X	F0										
	3	555	AA	2AA	55	X	F0						
AUTO SELECT	3	555	AA	2AA	55	555	90						
PROGRAM	4	555	AA	2AA	55	555	A0	PA	PD				
UNLOCK BYPASS	3	555	AA	2AA	55	555	20						
UNLOCK BYPASS PROGRAM	2	X	A0	PA	PD								
UNLOCK BYPASS RESET	2	X	90	X	00								
CHIP ERASE	6	555	AA	2AA	55	555	80	555	AA	2AA	55	555	10
BLOCK ERASE	6+	555	AA	2AA	55	555	80	555	AA	2AA	55	BA	30
ERASE SUSPEND	1	X	B0										
ERASE RESUME	1	X	30										
READ CFI QUERY	1	55	98										

- NAND Flash supports PAGE READ and WRITE while NOR flash support Word Read/Write.
- The 'Program' command writes a word into a random address supplied. NOR flash architecture has better support for random read as it has sufficient address lines to map the entire memory.

- NAND flash requires multiplexed address and data sequence to enable a read, resulting in slower initial read. To compensate for this, read access is typically done in bulk at Page level.
- NAND flash typically has a cached version of the Page Read/Write, this helps to improve its performance. Important feature for storage. The cached Page Read also help to mitigate the slower Read operation of NAND flash memories.
- In general, NOR has a faster Random Read while NAND tend to do more bulk (Page level) read/write. This coupled with lower cost per bit makes NAND more suitable to be used as bulk storage memory while NOR is more suitable for system memory which requires random access.

2. The SSD controller manage the translation between Logical Block Address (LBA) from the Application processor to the Physical Block Address (PBA) of the actual storage location of the data in the SSD.
   - Table 3.1 shows the mapping of LBA to PBA.
   - For simplicity's sake, we can assume that these addresses at block level address, e.g. LBA block 0 (LBA0) is mapped to PBA block 9 (PBA9) in the table.
   - Table 3.2 shows the erase count of each PBA block.
   - LBA, PBA block correspond to the flash block in the flash structure, i.e. erasure is at block level.
   - A 'Y' in the 'Empty' column indicates that the Flash Block is empty.

(a) Complete Table 3.1 for Dynamic wear leveling algorithm. You can further assume that
   - Access sequence from the processor is LBA 7, 4, 3, A.
   - All access will go through the wear levelling algorithm for physical block assignment.
   - If the LBA was originally assigned to a different PBA, the PBA containing the old data will be erased.

   What is the consideration to take for the erasure process of the old data?

Table 3.1

LBA	PBA
0	9
1	X
2	X
3	5
4	X
5	4
6	6
7	7

Table 3.2
Erase Count Table

PBA	Erase Count	Empty
0	5	N
1	33	Y
2	15	Y
3	98	Y
4	45	N
5	11	N
6	69	N
7	78	N

8	X
9	X
A	0

8	A3	Y
9	F1	N
A	56	Y

- Evaluation of which PBA block to assign is done before any PBA block erasure. New data must be re-written to the new target PBA block before erasing the old data.

Dynamic

Table 3.1	
LBA	PBA
0	9
1	X
2	X
3	5 → A(3)
4	X → 1(2)
5	4
6	6
7	7 → 2(1),
8	X
9	X
A	0 → 5(4)

Table 3.2 (Dynamic)		
Erase Count Table		
PBA	Erase Count	Empty
0	5 → 6(4)	N → Y(4)
1	33	Y → N(2)
2	15	Y → N(1)
3	98	Y
4	45	N
5	11 → 12(3)	N → Y(3) → N(4)
6	3	N
7	78 → 79(1)	N → Y(1)
8	A3	Y
9	F1	N
A	56	Y → N(3)

(b) In table 3.2, you can see that PBA block 6 has a very low erase count but would not be evaluated under Dynamic wear levelling algorithm. Static wear levelling attempts to relocate such flash blocks that are seldom re-written to achieve more even erase count distribution. There are many ways to implement Static wear levelling, below is one example. Go through the algorithm described and suggest enhancement to the algorithm.

- On every access, all blocks (occupied and empty) will be evaluated for block assignment.
- PBA block with the lowest erase count will be selected to store the data.

Enhancement Suggestions

- In the current method, if the chosen block to write into is already occupied, the target block will have to be erase and relocated, this will result in more erasure operation, increasing the total erase counts. To avoid this issue, separate the evaluation of occupied and empty flash blocks
  a. Evaluate empty blocks to decide which flash blocks to use for active re-writing of blocks.
  b. Evaluate occupied blocks periodically to re-locate occupied blocks with low erase counts, freeing these blocks for usage for subsequent access.
- Occupied blocks in part (b) above are only relocated to empty blocks to avoid unnecessary erasure process.

- Empty blocks with high erase counts can be chosen to store data which are seldom re-written into. May or may not be possible as this will need info on LBA access statistics, and provided the files associated with the LBA stays constant.

3. Figure 5.2a and 5.2b shows the Read and Write timing diagram of a processor's external memory interface. The processor can configure the access timing via three parameters: setup, strobe and hold. The description of these parameters can be found in Figure 5.1. The example in the figure is configured for setup = 2, strobe = 5 and hold =1. Figure 5.3a and 5.3b shows the read and write access timing diagram of a SRAM. Configure the processor's external memory interface to match SRAM timing. Following are some information that you'll need for the computation
   - CPU Clock cycle = 5 ns
   - The unit of the setup-strobe-hold parameters are in CPU clock cycles.
   - Processor ARE_ pin is connected to SRAM OE_ pin
   - Processor latch in the data on rising edge of ARE_
   - Maximum value of SRAM tOE = 40 ns
     - tOE is the Output Enable to Output Valid timing.
   - Minimum value of SRAM tWP = 40ns
     - tWP is the Write Pulse Width timing
   - Minimum tDH = 5 ns
     - tDH is the Data Hold from WE signal inactive
   - Minimum tAS = 0 ns.
     - tAS is the Address setup time before WE signal is active

   - Max tOE_ = 40ns and min tWP = 40ns. Strobe = 40/5 = 8.
   - Min tDH = 5ns => hold = 5/5 = 1
   - Setup is less critical. Can be setup = 1 since minimum tAS = 0.

Parameters	Control Bits	Definition
Setup periods	RDSETUP WRSETUP	A setup period is the time in CPU clock cycles given to setup the address, chip enable ($\overline{CE}$), and byte enable ($\overline{BE}$) signals before the read strobe signal ($\overline{ARE}$) or write strobe signal ($\overline{AWE}$) falls. For an asynchronous read operation, this is also the setup period for the output enable signal ($\overline{AOE}$) before $\overline{ARE}$ falls.
Strobe periods	RDSTROBE WRSTROBE	A strobe period is the time in CPU clock cycles between the falling (activation) and rising (deactivation) of the read or write strobe signal.
Hold periods	RDHOLD WRHOLD	A hold period is the time in CPU clock cycles during which the address and byte enable lines are held active after the read or write strobe signal rises. For an asynchronous read operation, this is also the hold period for the output enable signal after $\overline{ARE}$ rises.

Figure 5.1: External Memory Interface Configuration Parameters

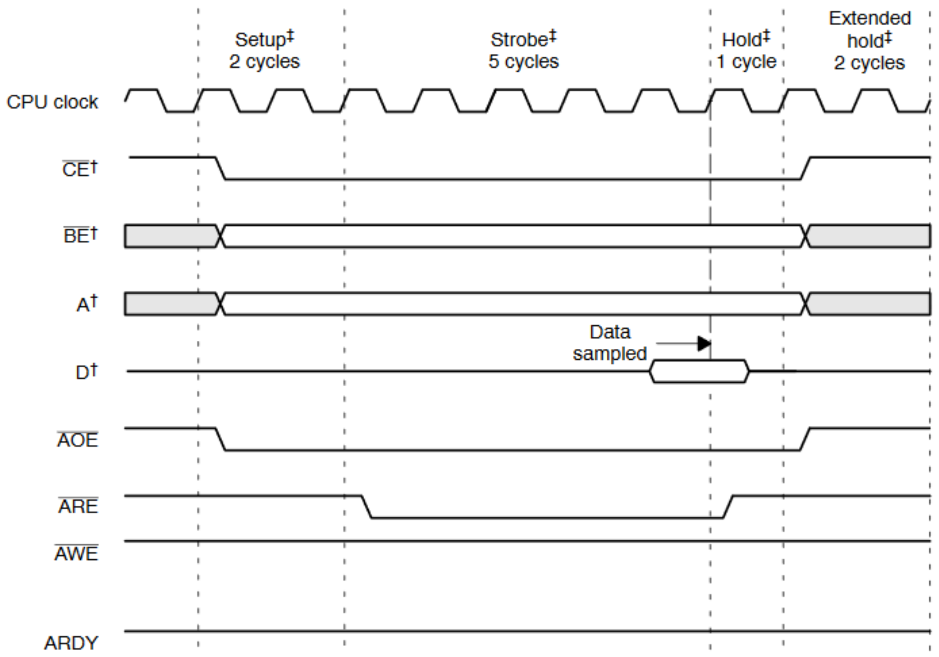


Figure 5.2: Processor External Read Timing Diagram

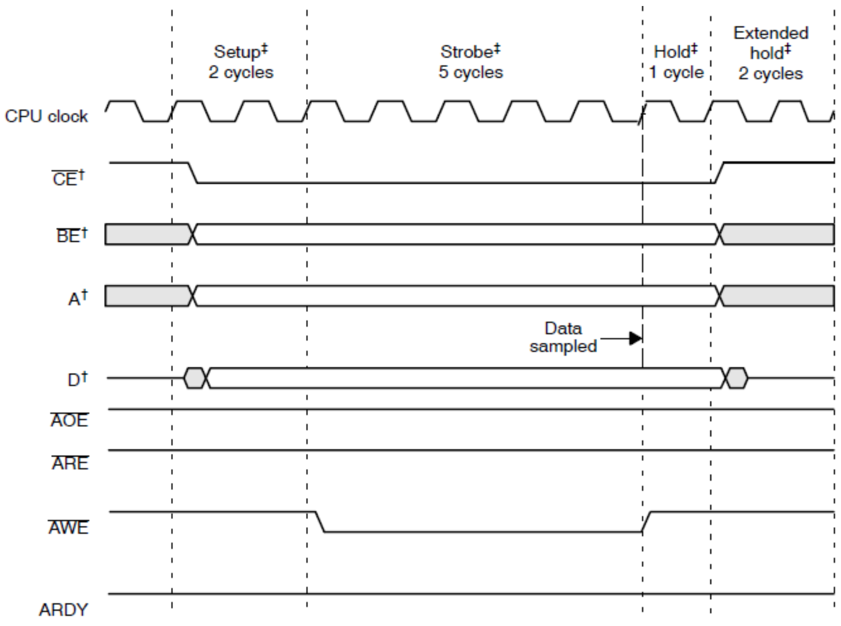


Figure 5.2b: Processor External Write Timing Diagram

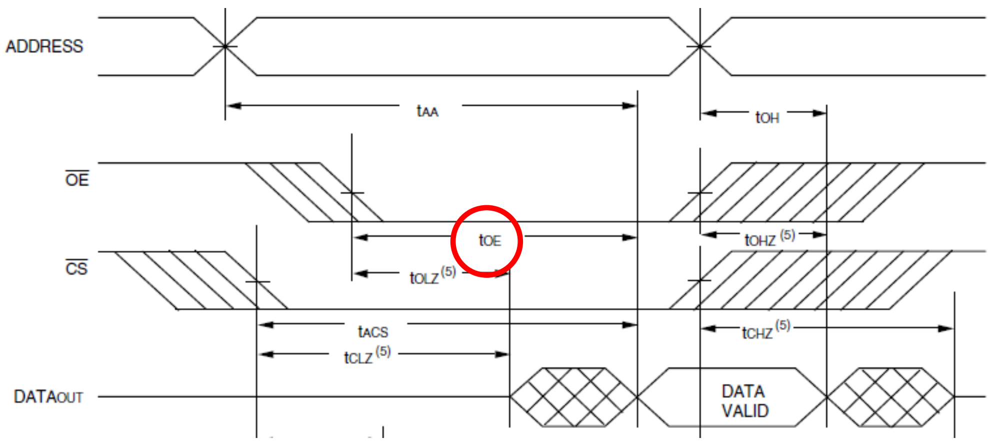


Figure 5.3: SRAM Read access timing diagram

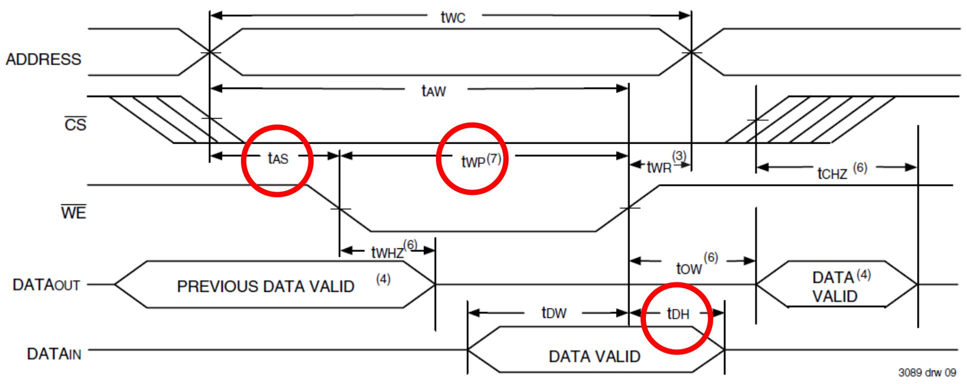
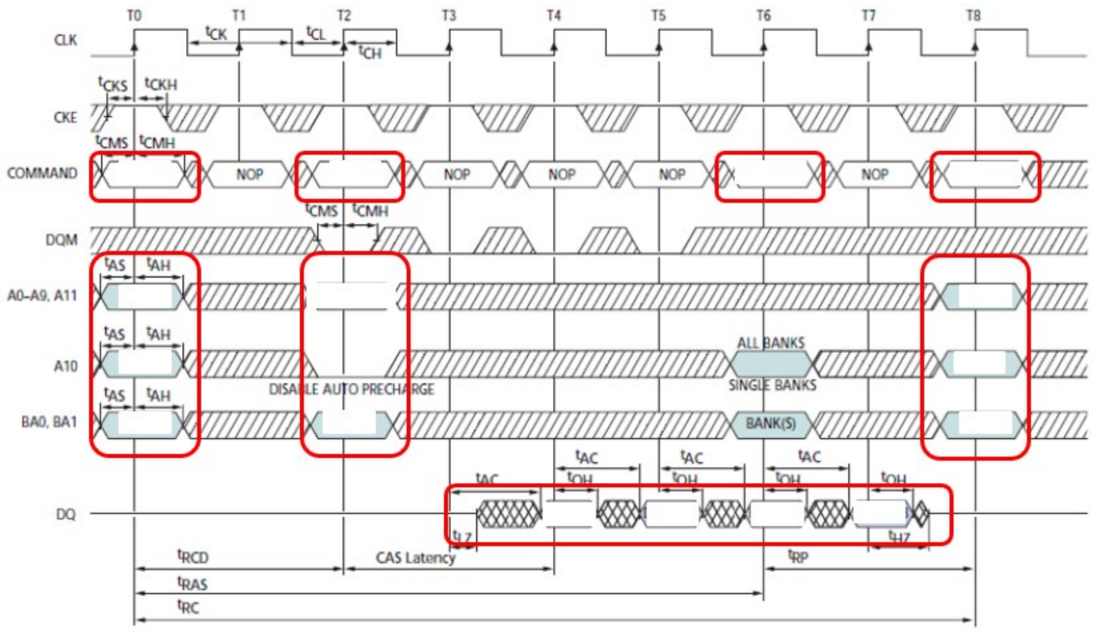


Figure 5.3b: SRAM Write access timing diagram

4. Complete the timing diagrams in Table 6.1 and 6.2 below, fill in the missing information on commands, data, addresses etc.
   - What is the CAS latency of the SDRAM for Table 6.1?

   - Table 6.1: CAS Latency =2.

Table 6.1: SDRAM Read without Auto-Precharge



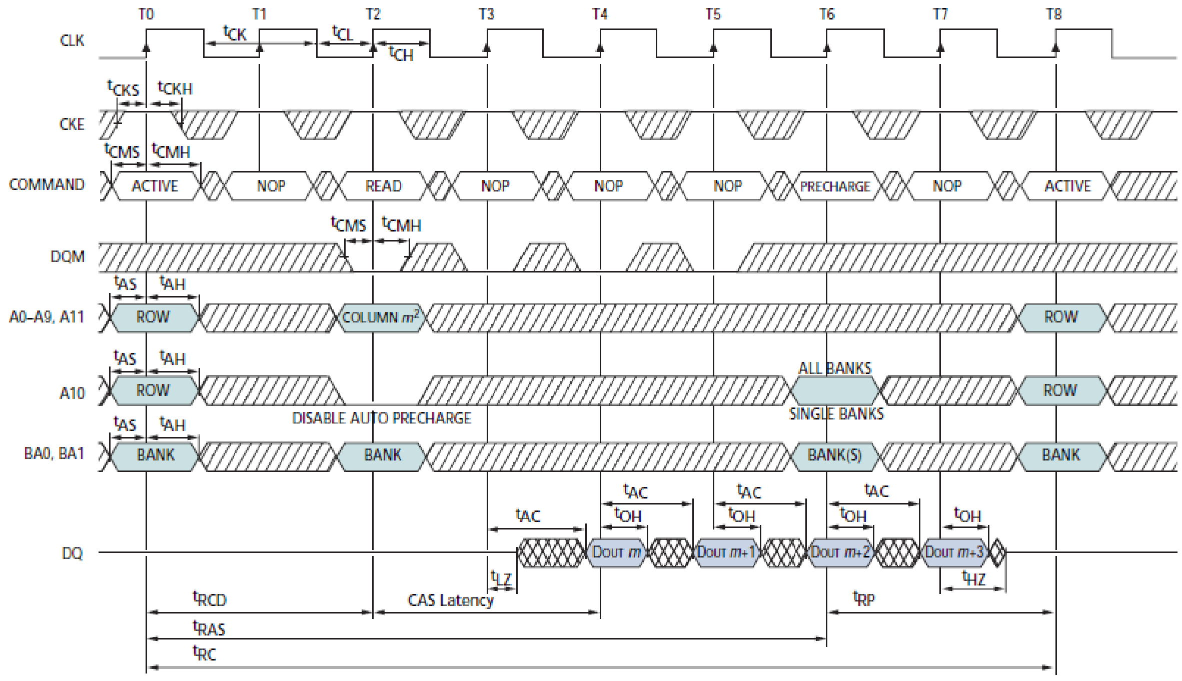
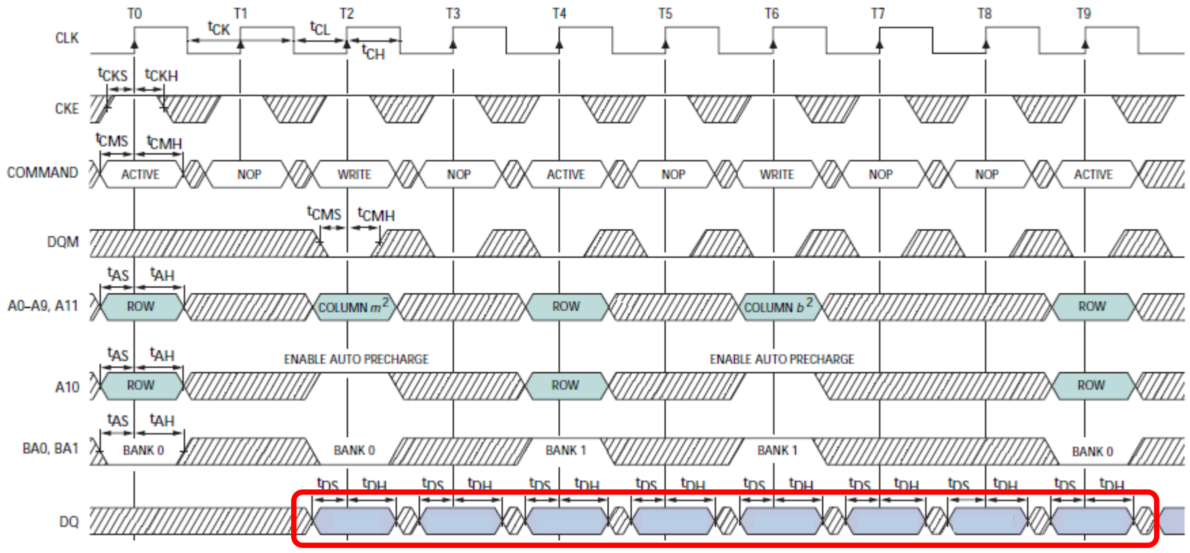
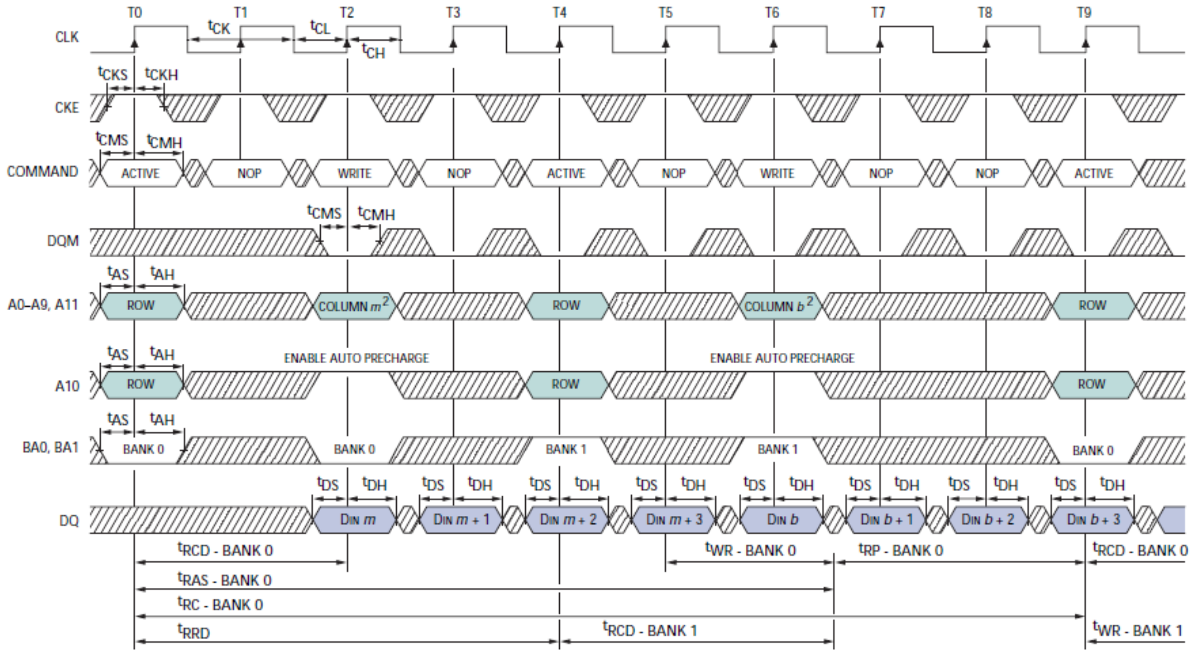


Table 6.2: SDRAM Write – Alternating Bank Access (with Auto-Precharge)

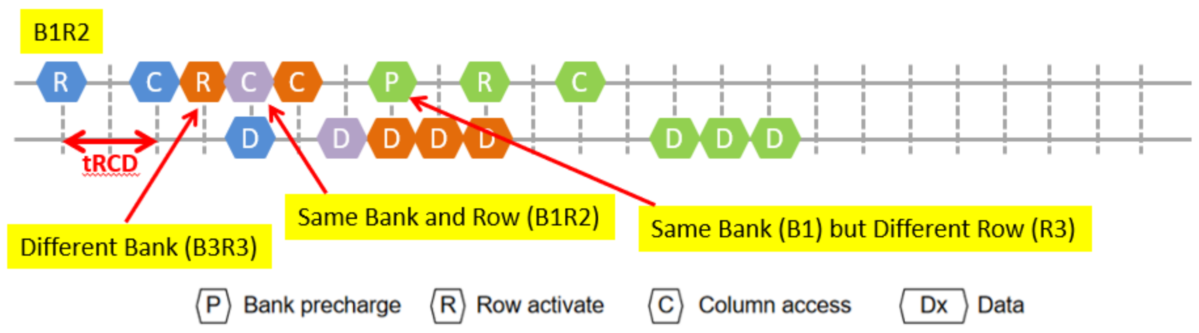




5. Arrange the access sequence to optimise the access time for the following data. Given that the tRCD(min) = 2 SDRAM clocks. tRCD is the minimum delay needed after an ACTIVE command before user can issue the READ/WRITE command to the SDRAM.

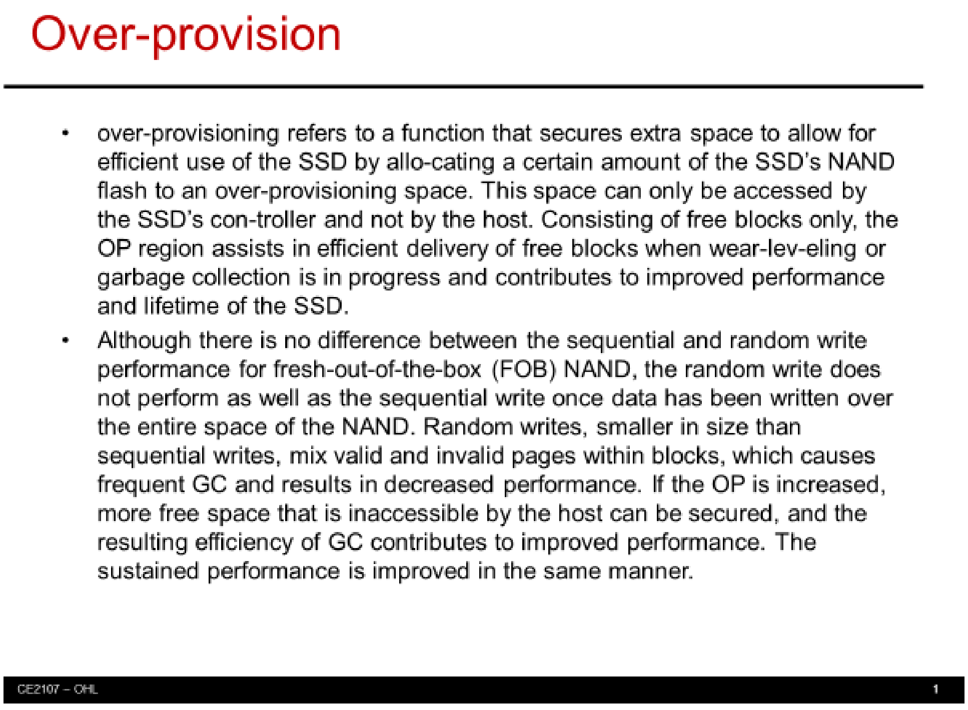
• Bank1-Row2-Byte0	(1 byte)
• Bank3-Row3-Byte2	(3 bytes)
• Bank1-Row3-Byte2	(3 bytes)
• Bank1-Row2-Byte3	(1 byte)

- Second access (B3R3B2) was brought forward as it is from a different bank, so access can be overlapped with the first access (B1R2B0).
- Fourth access (B1R2B3) is brought forward as well to save on one pre-charge cycle.



(Optional)

6. SSD Design considerations: homework given during lecture



Over-provision

- over-provisioning refers to a function that secures extra space to allow for efficient use of the SSD by allo-cating a certain amount of the SSD's NAND flash to an over-provisioning space. This space can only be accessed by the SSD's con-troller and not by the host. Consisting of free blocks only, the OP region assists in efficient delivery of free blocks when wear-lev-eling or garbage collection is in progress and contributes to improved performance and lifetime of the SSD.
- Although there is no difference between the sequential and random write performance for fresh-out-of-the-box (FOB) NAND, the random write does not perform as well as the sequential write once data has been written over the entire space of the NAND. Random writes, smaller in size than sequential writes, mix valid and invalid pages within blocks, which causes frequent GC and results in decreased performance. If the OP is increased, more free space that is inaccessible by the host can be secured, and the resulting efficiency of GC contributes to improved performance. The sustained performance is improved in the same manner.

CE2107 – OHL 1

SSD Design Consideration

- **Garbage Collection**
- Technique used to free up previously written blocks
- Consolidates pages by moving and rewriting pages from multiple blocks to fill up fewer new ones.
- Old blocks are then erased to provide storage space for new incoming data.

Figure 2: Garbage Collection

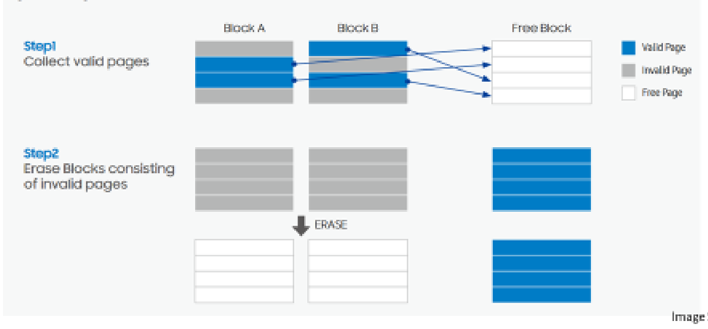
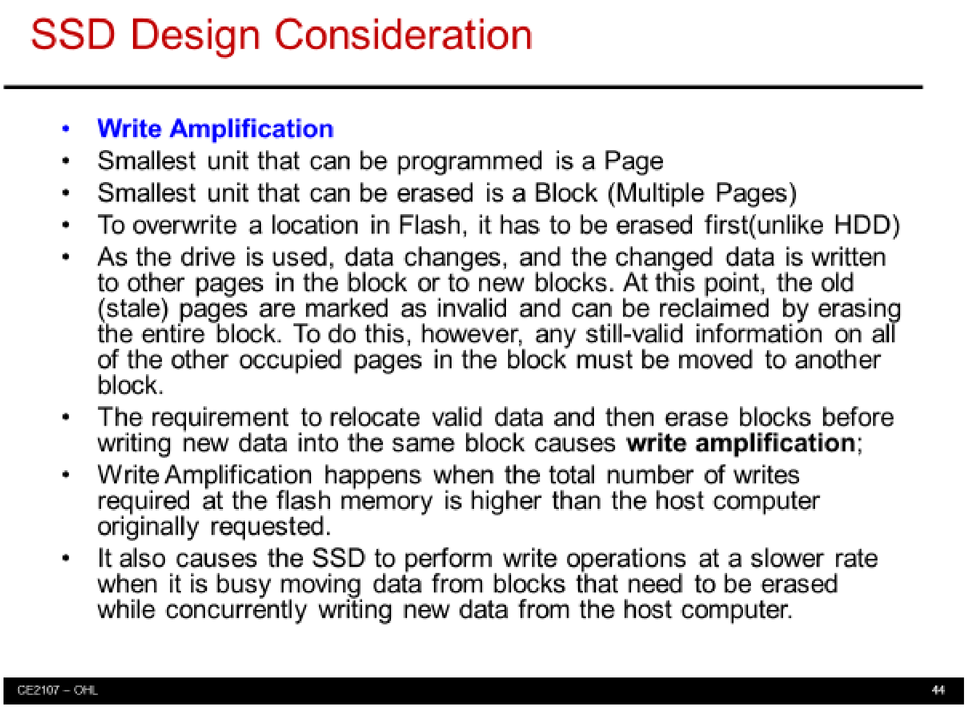


Image Source: Samsung

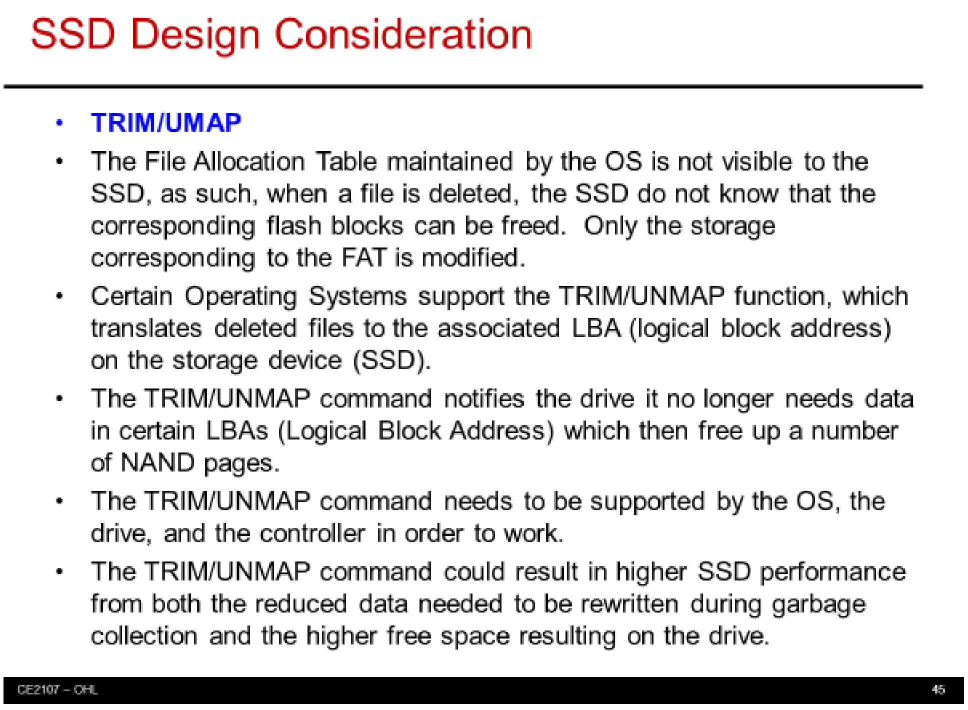
CE2107 – OHL 43



SSD Design Consideration

- **Write Amplification**
- Smallest unit that can be programmed is a Page
- Smallest unit that can be erased is a Block (Multiple Pages)
- To overwrite a location in Flash, it has to be erased first(unlike HDD)
- As the drive is used, data changes, and the changed data is written to other pages in the block or to new blocks. At this point, the old (stale) pages are marked as invalid and can be reclaimed by erasing the entire block. To do this, however, any still-valid information on all of the other occupied pages in the block must be moved to another block.
- The requirement to relocate valid data and then erase blocks before writing new data into the same block causes **write amplification**;
- Write Amplification happens when the total number of writes required at the flash memory is higher than the host computer originally requested.
- It also causes the SSD to perform write operations at a slower rate when it is busy moving data from blocks that need to be erased while concurrently writing new data from the host computer.

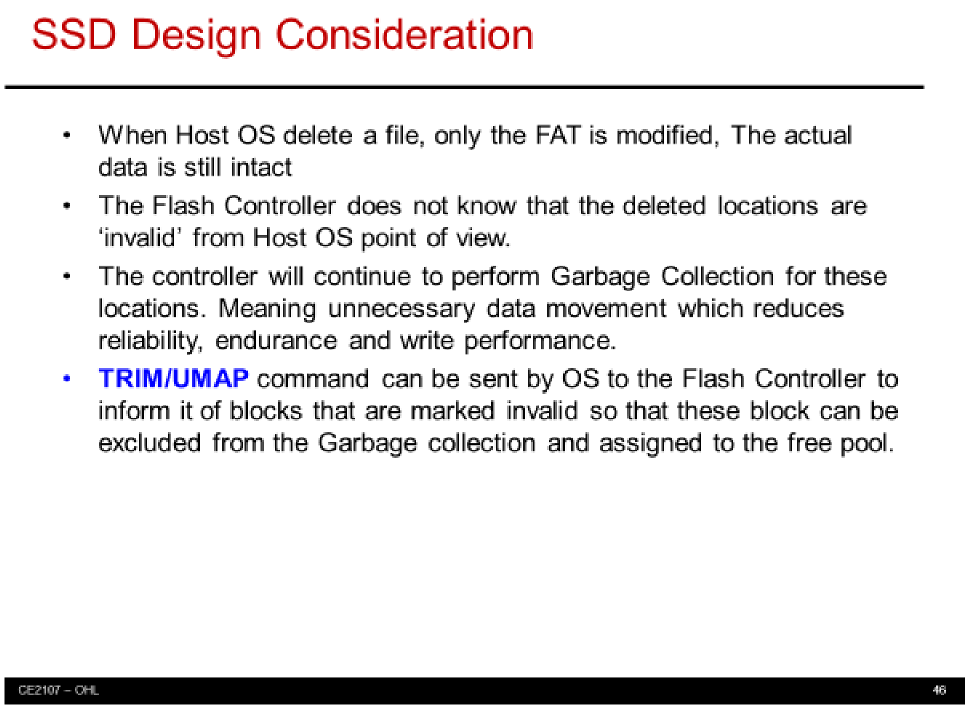
CE2107 – OHL 44



SSD Design Consideration

- **TRIM/UMAP**
- The File Allocation Table maintained by the OS is not visible to the SSD, as such, when a file is deleted, the SSD do not know that the corresponding flash blocks can be freed. Only the storage corresponding to the FAT is modified.
- Certain Operating Systems support the TRIM/UNMAP function, which translates deleted files to the associated LBA (logical block address) on the storage device (SSD).
- The TRIM/UNMAP command notifies the drive it no longer needs data in certain LBAs (Logical Block Address) which then free up a number of NAND pages.
- The TRIM/UNMAP command needs to be supported by the OS, the drive, and the controller in order to work.
- The TRIM/UNMAP command could result in higher SSD performance from both the reduced data needed to be rewritten during garbage collection and the higher free space resulting on the drive.

CE2107 – OHL 45



SSD Design Consideration

- When Host OS delete a file, only the FAT is modified, The actual data is still intact
- The Flash Controller does not know that the deleted locations are 'invalid' from Host OS point of view.
- The controller will continue to perform Garbage Collection for these locations. Meaning unnecessary data movement which reduces reliability, endurance and write performance.
- **TRIM/UMAP** command can be sent by OS to the Flash Controller to inform it of blocks that are marked invalid so that these block can be excluded from the Garbage collection and assigned to the free pool.

CE2107 – OHL 46