Computer Science 605.411

Module 11 Example Set 1

1. a) On average what fraction of a rotation must a disk make to get to the beginning of a random sector on a track?
On average it takes ½ revolution to get to the beginning of a random sector on a track.

b) If a disk rotates 500 RPM (revolutions per minute), how long on average is the rotational delay to get to the start of a random sector?

The rotation rate of 500 RPM = 500/60 = 8.333 revolutions per second
So one revolution takes 60/500 = 1/8.333 = 0.12 seconds
The rotational delay = 0.5 * 0.12 seconds = 0.06 seconds

2. Give a concise answer to each of the following questions.
a) What is memory mapped I/O?
This is an I/O scheme in which portions of the address space are assigned to I/O devices so that the I/O device registers (data and control) can be accessed using the normal memory load and store instructions.

b) Why is DMA an improvement over direct programmed controlled I/O?
DMA is a mechanism that provides a device controller the ability to transfer data directly to or from the memory without involving the processor. This allows the CPU to perform arithmetic and other instructions while the DMA is going on in parallel.

c) When would DMA be a poor choice for I/O?
DMA is not useful when the amount of data that needs to be transferred between memory and the I/O device is relatively small (as for a mouse or keyboard). This is because the overhead of setting up the DMA transfer could actually take longer than the actual transfer. So DMA is used for block tranfers.

d) How can DMA cause the CPU to potentially use stale memory data?
Since DMA transfers go directly to memory, they bypass the CPU as well as the cache. This makes it possible for the contents of cache to become inconsistent with the content of memory.

3. Suppose we have a magnetic disk drive with the following parameters:

Average seek time	12 ms
Rotation rate	3600 RPM
Transfer rate	3.5 MB/second
# sectors per track	64
Sector size	512 bytes
Controller overhead	5.5 ms

a) What is the average time to read a single sector?
Disk access time = seek time + rotational delay + transfer time + controller overhead
= 12 + (0.5* 60/3600)*1000 + (512/(3.5*2^{20}))*1000 + 5.5 = 25.97 ms

b) What is the average time to read 16 consecutive sectors in the same track?
The seek time and rotational delay would be the same; the transfer time would correspond to 16*512 = 8192 bytes of data.
Disk access time = seek time + rotational delay + transfer time + controller overhead
= 12 + (0.5* 60/3600)*1000 + (8192/(3.5*2^{20}))*1000 + 5.5 = 28.07 ms

4. The daily computer system processing load for a certain company consists of 60% CPU activity and 40% disk activity. Customers are complaining that the system is too slow. After doing some research, you have learned that you can upgrade the disks for $8000 to make them 2.5 times as fast as they are currently. You have also learned that you can upgrade the CPU to make it 1.4 times as fast for $5000.

a) Which option would you choose to yield the best performance improvement for the least amount of money?
Changing the CPU would provide a speedup = 1/((1-0.60) + (0.60/1.4)) = 1.2069 which is a 20.69% speedup.
Changing the disks would provide a speedup = 1((1-0.40) + (0.40/2.5)) = 1.3158 or a 31.58% speedup.
The cost per 1% speedup for the CPU = $5000/20.69% = $241.66
The cost per 1% speedup for the disks = $8000/31.58% = $253.32
So upgrading the CPU is the better option based on cost.

b) Which option would yield the bigger improvement if cost is no concern?
Upgrading the disks provides the greater speedup but costs more. So this is the better option if cost is of no concern.

5. Answer the following questions:

a) What is the average time to read or write a 512-byte sector for a typical disk rotating at 7200 RPM? The average seek time is 8 ms, the transfer rate is 20 MB/sec, and the controller overhead is 2 ms. Assume that the disk is idle so that there is no need to wait for any previous operation to complete.

Average disk access time =
seek time + rotational delay + transfer time + controller overhead =
8 ((0.5*60*1000/7200) + (512/20*2^{20})*1000 + 2 = 14.17 ms

b) A program repeatedly performs a three-step process: it reads in a 4-KB block of data from disk, does some processing on that data, and then writes out the result as another 4-KB block elsewhere on the disk. Each block is contiguous and randomly located on a single track on the disk. The disk drive rotates at 7200 RPM, has an average seek time of 8 ms, and has a transfer rate of 20 MB/sec. The controller overhead is 2 ms. No other program is using the disk or processor, and there is no overlapping of disk I/O with data processing. The processing step takes 20 million clock cycles and the clock rate is 400 MHz. What is the overall speed of the system in blocks processed per second assuming no other overhead?

Disk read or write time for a 4 KB block
= seek time + rotational delay + transfer time + controller overhead
= 8 + (0.5*60*1000/7200) + (4*1024/20*2^{20})*1000 + 2 = 14.17 ms

Processing time for a 4 KB block = number of clock cycles * cycle time
= 20*10^6 * (1/(400*10^6)) = 50 ms

Total time to process one 4 KB = read time + processing time + write time
= 14.17 + 50 + 14.17 = 78.34 ms

Number of blocks processed per second = 1/(total time to process 1 block)
= 1/78.34 ms = 0.01276 per ms or 12.76 per second

Module 11 Example Set 2

1. A function has been written to service I/O requests from a certain device.
a) What determines the maximum rate at which the function will be called if interrupts are used in servicing the device?
The frequency at which the device generates interrupts would dictate the maximum rate at which the function must be called. Since interrupts occur at unpredictable time, the context of the running program must be saved and restored as well.

b) What determines the maximum rate at which the function will be called if polling is used in servicing the device?
The maximum rate would be determined by the duration of the polling loop.

c) If it takes 50 cycles to poll the device, could an I/O rate of fifty million requests per second be handled if the CPU clockrate is 2.5 GHz?
A clock rate of 2.5 GHz corresponds to a cycle time of 0.4 ns. Fifty cycles would be 50*0.4 = 20 ns. This corresponds to 50 million operations per second, however each request requires that the device be polled as well as serviced. Since the time to service the device will be greater than 0, 50 million requests could not be handled. The maximum rate of requests that can be handled would depend on the sum of the polling time plus the service time for the device.

2. The read access delay for an I/O system is defined as the time required for the device to acquire the requested data and prepare to start transmitting the data. The data transfer rate for an I/O device is defined as the number of bytes per second that it can transmit. Suppose that an I/O system has an access delay of 10 seconds and transfers data at the rate of 4096 bytes per second.
How long will the system require to complete a 3145728-byte I/O request?
The time required on the system is 10 + 3145728/4096 = 778 seconds.

3. The combined controller overhead and seek time for a certain disk system is 14 milli-seconds on average. It also takes an average of one half revolution of the disk to get to the beginning of a requested sector. The disk rotates at a rate of 7200 revolutions per minute. Each track contains 128 sectors, each of which is 512 bytes in size. The disk can transfer data as fast as it can be read. For this problem assume that there are no gaps between sectors.
How long will it take to read the contents of a file that is 50 kB in size if the file resides on a single track?
It takes on average 14 milli-seconds to move the access head to the track that contains the file. On average 0.5 revolutions will be required to get to the beginning of the file. One revolution takes 8.33 milli-seconds for one revolution, so one half would be 4.167 milli-seconds. The file contains 50*1024 = 51200 bytes of data which occupies 51200/512 = 100 sectors. It takes 100*8.33/128 = 6.51 milli-seconds to transfer the file contents. Therefore the total time = 14 + 4.167 + 6.51 = 24.677 milli-seconds to read the file contents.

4. Forty percent of the workload on a certain computer system corresponds to the execution of code not related to I/O. The remaining 60% is due to I/O operations. The workload requires 600 milli-seconds to complete.
a) What speedup would be achieved by making the I/O system 80% faster?
The original I/O systems consumes 0.6*600 = 360 milli-seconds and the code not related to I/O consumes 240 milli-seconds.
The improved I/O systems would take 360/1.8 = 200 seconds, so it provides a speedup of 600/(200+240) = 1.36

b) What speedup would be achieved by making the code not related to I/O 80% faster?
The improved code not related to I/O would consume 0.4*600/1.8 = 133.33 milli-seconds, so it provides a speedup of 600/(133.33+360) = 1.22

Computer Science 605.411

Module 11 Example Set 3

1. A processor has eight interrupt lines (numbered 0 – 7) from highest (0) to lowest (7) priority. Initially, no interrupts are pending and the following sequence of interrupts then occurs: 4, 7, 1, 3, 0, 5, 6, 4, 2, 1. Once an interrupt occurs, no further interrupts can get in until the processing of the current interrupt has been completed. That is, interrupts simply pend until the interrupt handler returns. Upon return from the interrupt handler, the highest priority pending interrupt would be serviced next. If the time required to handle an interrupt is such that two additional interrupts occur while the current interrupt is being serviced, in what order would the ten interrupts be serviced? Write down the interrupt numbers in the order in which the interrupts would be serviced.

Interrupt 4 is handled first. By the time processing for interrupt 4 is complete, interrupts 7 and 1 will have arrived, so interrupt 1 gets handled. By the time processing is complete for interrupt 1, interrupts 7, 3 and 0, will be pending; so interrupt 0 is handled next. Upon completing interrupt 0, interrupts 7, 3, 5 and 6 will be pending and interrupt 3 is handled next. At the end of interrupt 3, interrupts 7, 5, 6, 4 and 2 will be pending and interrupt 2 is handled next. While processing interrupt 2, interrupt 1, the final interrupt arrives so interrupts 7, 5, 6, 4 and 1 will be pending. These five interrupts will be handled in priority order. Hence the ten interrupts would be serviced in the order: 4, 1, 0, 3, 2, 1, 4, 5, 6, 7.

2. An I/O device transfers input data at the rate of 10^7 bytes per second over an I/O bus with a total bandwidth of $100 * 2^{20}$ bytes per second. The input data consists of 2500 independent pages each of which is 4096 bytes in size. The CPU for this system has a clock rate of 200 MHz (i.e. 200 million cycles per second) and the memory system is fast enough to sustain these transfer rates.

a) During each I/O transfer, the processor performs no other activity than that required to receive the input data. If the arrival of each data byte causes an interrupt, what is the maximum number of clock cycles that the CPU can consume in responding to and handling the interrupts from the device if it is to avoid the loss of data?

To avoid the loss of data, the time required to respond the each interrupt and input the data byte must be less than the time interval between consecutive bytes. The limiting factor is the speed at which the device transfers data since the bus bandwidth is 10 times the device's data rate.

The I/O device transfers data at a rate of 10^7 bytes per second , so the interval between bytes is 10^{-7} seconds which corresponds to $10^{-7} / (5*10^{-9}) = 20$ CPU clock cycles. So no more than 20 cycles must be consumed between the arrival of adjacent bytes.

b) When a DMA controller is used to perform the transfer, 1000 CPU cycles are required to set up and initiate the DMA transaction and an additional 1500 CPU cycles are required to respond to the interrupt when the DMA transfer completes. What fraction of the available CPU cycles would be consumed to perform a DMA transfer of a single page?

With the DMA controller, the processor only consumes cycles for the purpose of I/O when it is setting up the DMA transfer and when it is responding to the interrupt at the conclusion of the DMA transfer.
The time required for the device to transfer a page is
$4096 * 10^{-7}$ seconds $= 4096 * 10^{-7} / 5 * 10^{-9} = 81920$ cycles.
The CPU consumes 1000 cycles to setup the DMA transaction + 1500 cycles to service the interrupt at the conclusion of the transfer for a total of 2500 cycles.
The total time required to complete the all activity required to transfer a single page corresponds to 1000 + 81920 + 1500 = 84420 cycles. However, the CPU is only active during 2500 of these cycles for the purpose of the transfer.
This corresponds to 2500/84420 = 2.96 per cent of the available CPU cycles.

3. The read access delay for an I/O system is defined as the time required for the device to acquire the requested data and prepare to start transmitting the data. The data transfer rate for an I/O device is defined as the number of bytes per second that it can transmit. Suppose that there are two different I/O systems A and B. System A has a data transfer rate of 5120 bytes per second and has an access delay of 5 seconds. System B has a data transfer rate of 3072 bytes per second and has an access delay of 4 seconds.

a) How long will each system require to complete a 3145728-byte I/O request?
System A: 5 + (3145728 / 5120) = 619.4 seconds
System B: 4 + (3145728 / 3072) = 1028 seconds

b) It is observed that for an I/O request of size N bytes, the two systems require the same amount of time to complete the request. What value is N?
$5 + (N / 5120) = 4 + (N / 3072) \Rightarrow 5 - 4 = (N / 3072) - (N / 5120) = (5N - 3N) / 15360, \Rightarrow 1 = 2N / 15360 = N / 7680 \Rightarrow N =$ 7680 bytes.

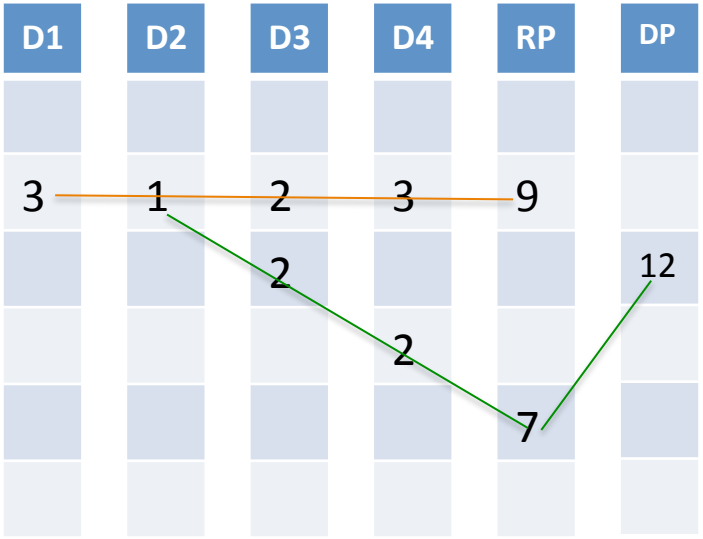
4. "Cycle stealing" is one mode of operation used by a DMA controller in performing I/O transfers. What are two other modes that could be used by the DMA controller?

The two other modes are transparent and burst (i.e. block) transfer mode.

RAID 6 Example

Since RAID6 makes use of a two-parity scheme, it is also referred to as RAID-DP (for dual parity). It works by storing data in horizontal rows, calculating parity for data in the row and storing the parity on a separate disk. However, it also includes a second parity disk which stores diagonal parity across the disks in a RAID-DP group.

To simplify the illustration of its operation, we will compute the parity as the simple arithmetic sum of the data blocks in each row or along each diagonal. The following example is based on a system with 4 data disks, a horizontal or row parity disk and a diagonal parity disk.



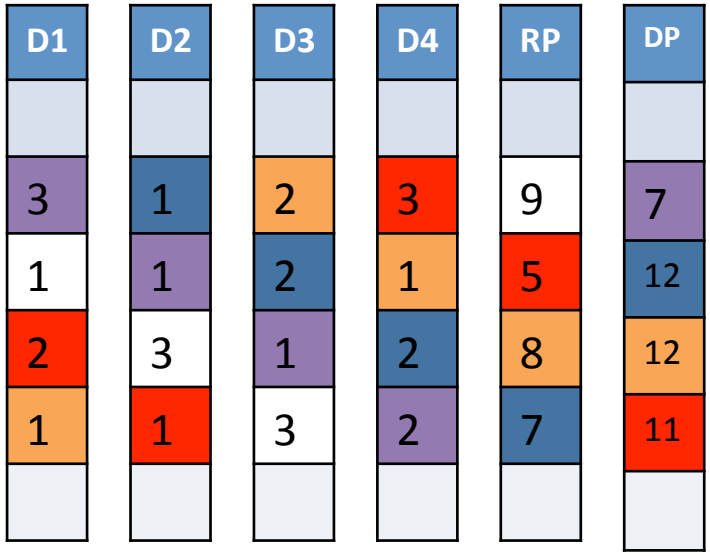
Row parity RP = 3+1+2+3 = 9
If D2 is lost, its data block can be computed as 9 - 3 - 2 - 3 = 1 the parity minus the sum of the remaining data blocks in the row.

DP = 1+2+2+7 = 12
(diagonal parity)

RAID 6 Example

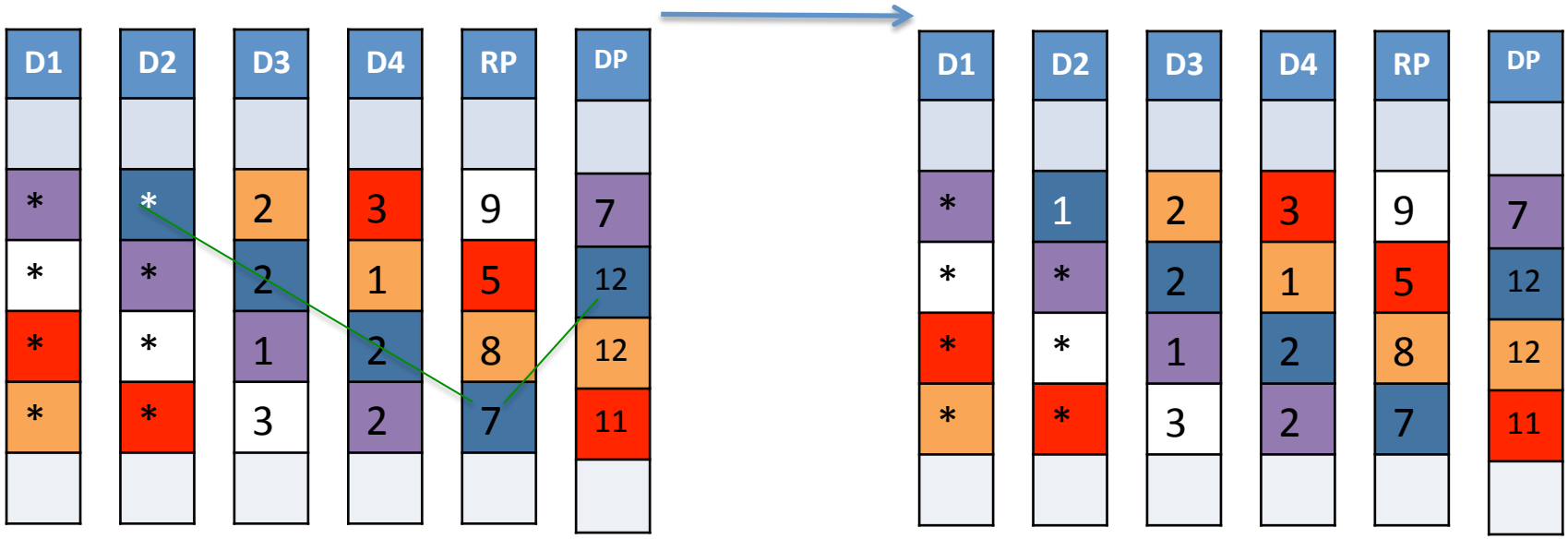
The diagram below shows the contents of all blocks on the disks and the color scheme indicates which blocks are within each diagonal.

Each diagonal parity stripe misses exactly one of the disks, but each diagonal stripe misses a different disk. This results in one diagonal stripe that doesn't get parity generated for it and is not stored on the diagonal parity disk (the white or noncolored stripe in this example).



RAID 6 Example

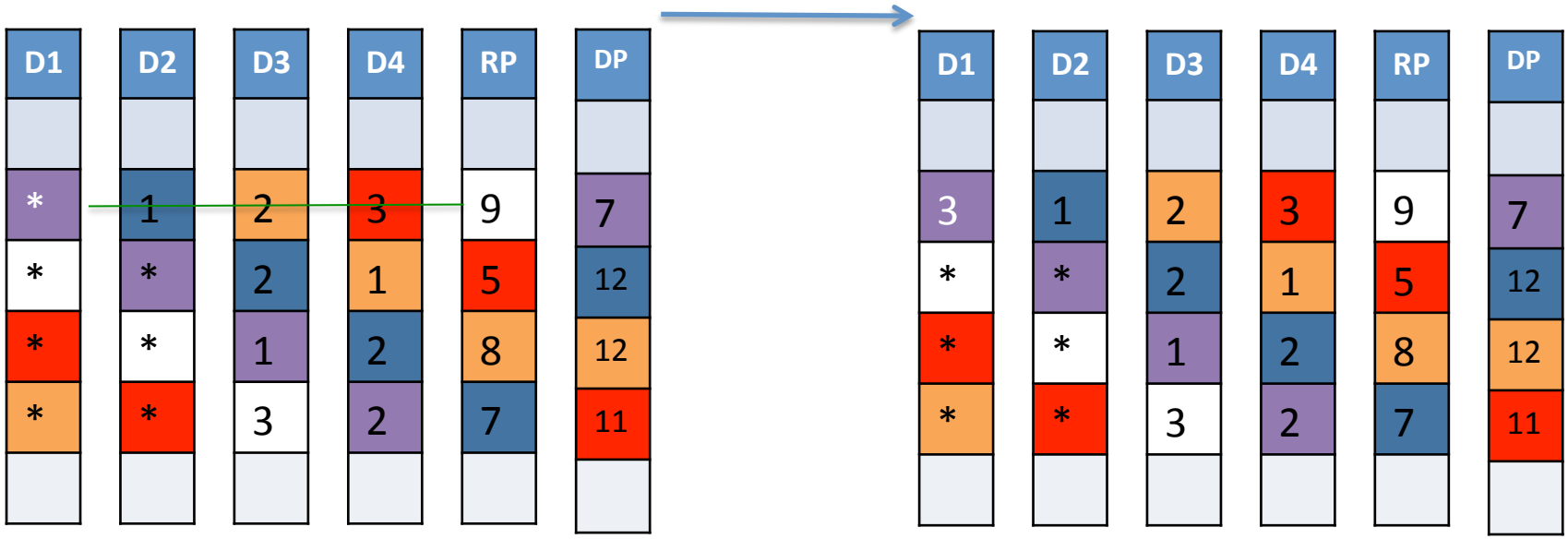
To see how missing blocks on two disks can be reconstructed, assume that both disk1 and disk2 are lost due to a head crash or some other event:



The first missing block on D2 is given by $12 - 7 - 2 - 2 = 1$.

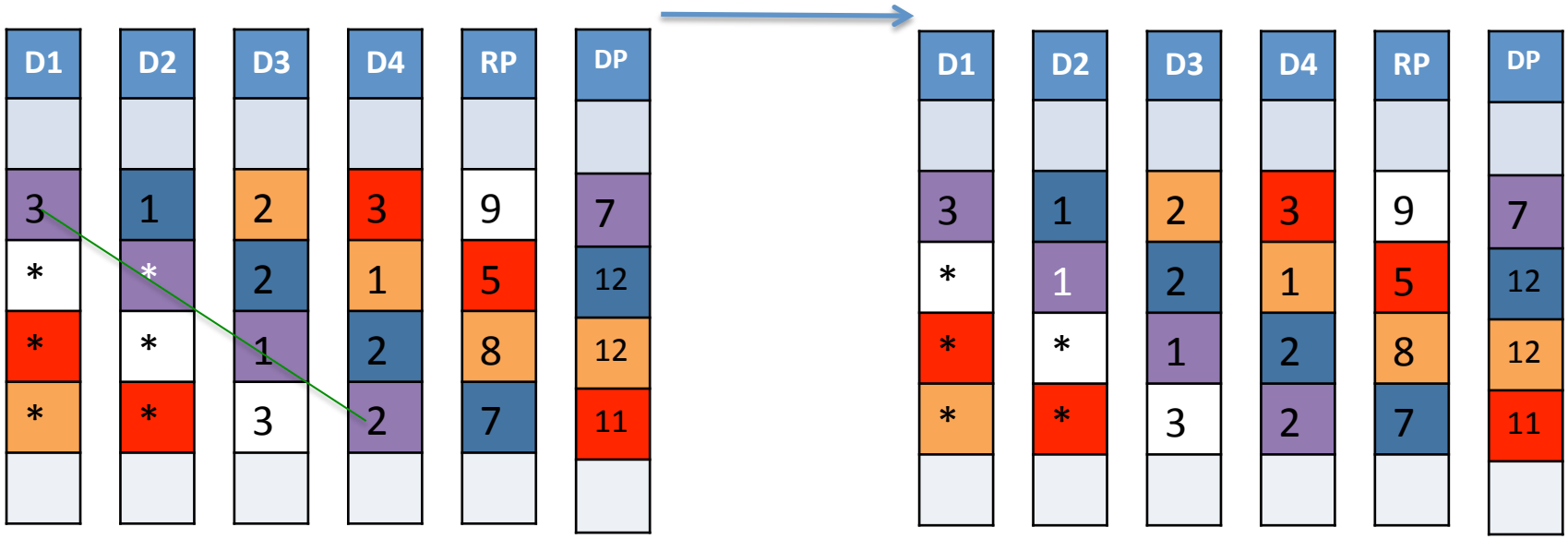
RAID 6 Example

Once the missing blue diagonal information has be reconstructed, the first missing block on D1 can be computed using the row parity as 9 – 3 – 2 – 1 = 3



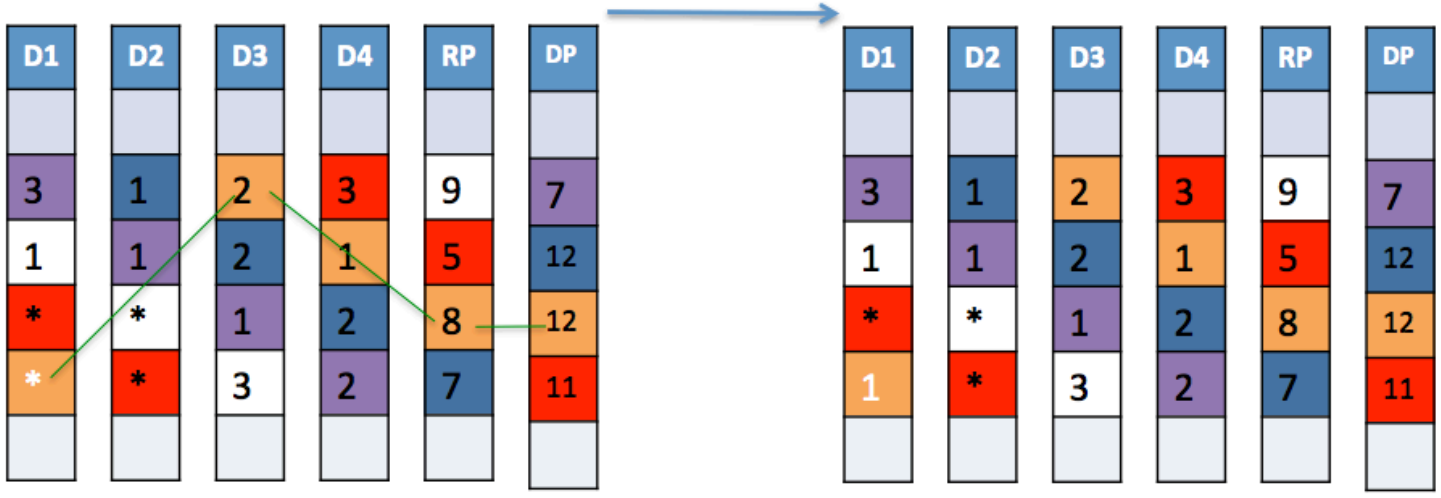
RAID 6 Example

Now using the purple diagonal stripe we get 7 - 3 -1 - 2 = 1 for the second missing block on disk2.

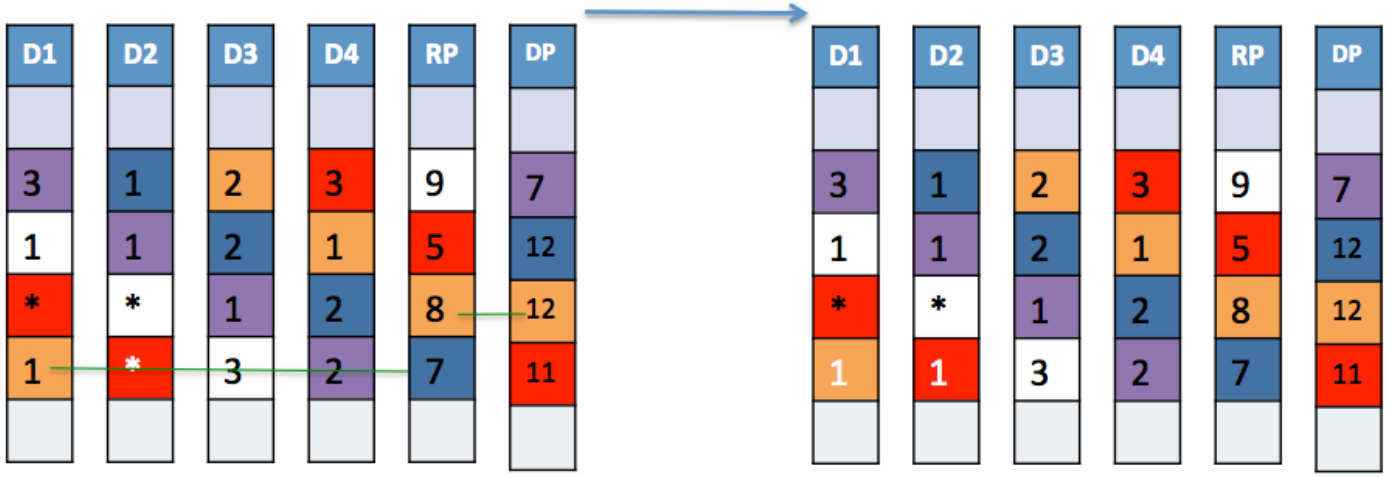


Next the second missing block on disk1 is given by 5 – 1 – 2 – 1 = 1 using row parity

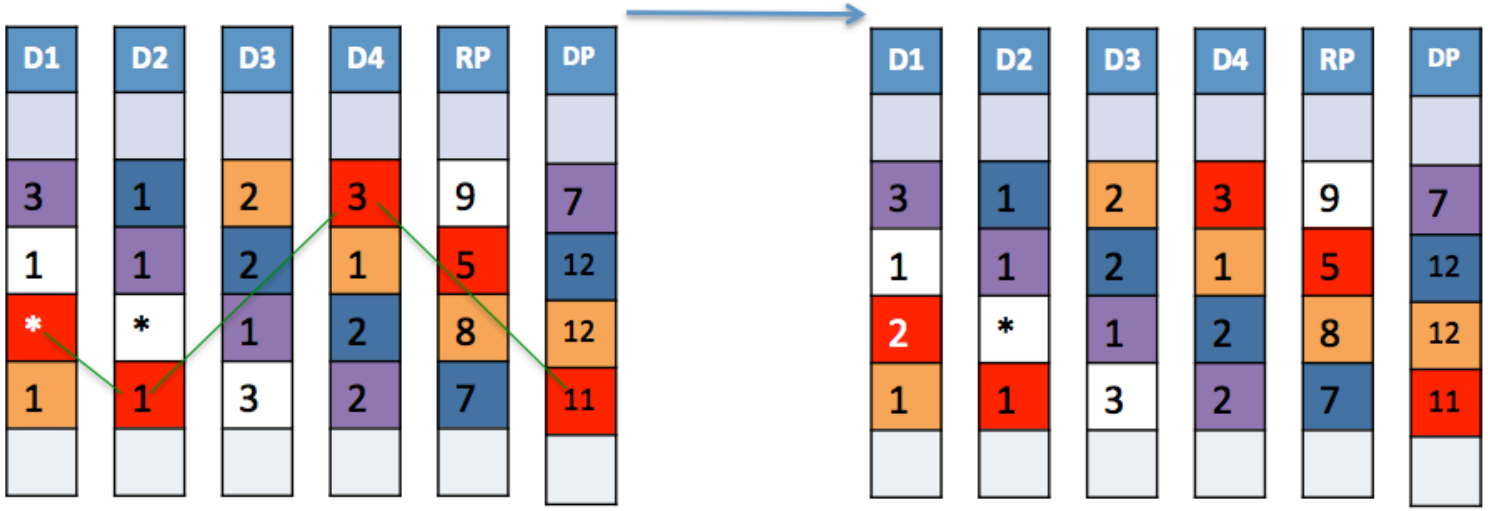
Using the orange diagonal strips gives 12 – 8 – 1 – 2 = 1 for the fourth missing block on D1.



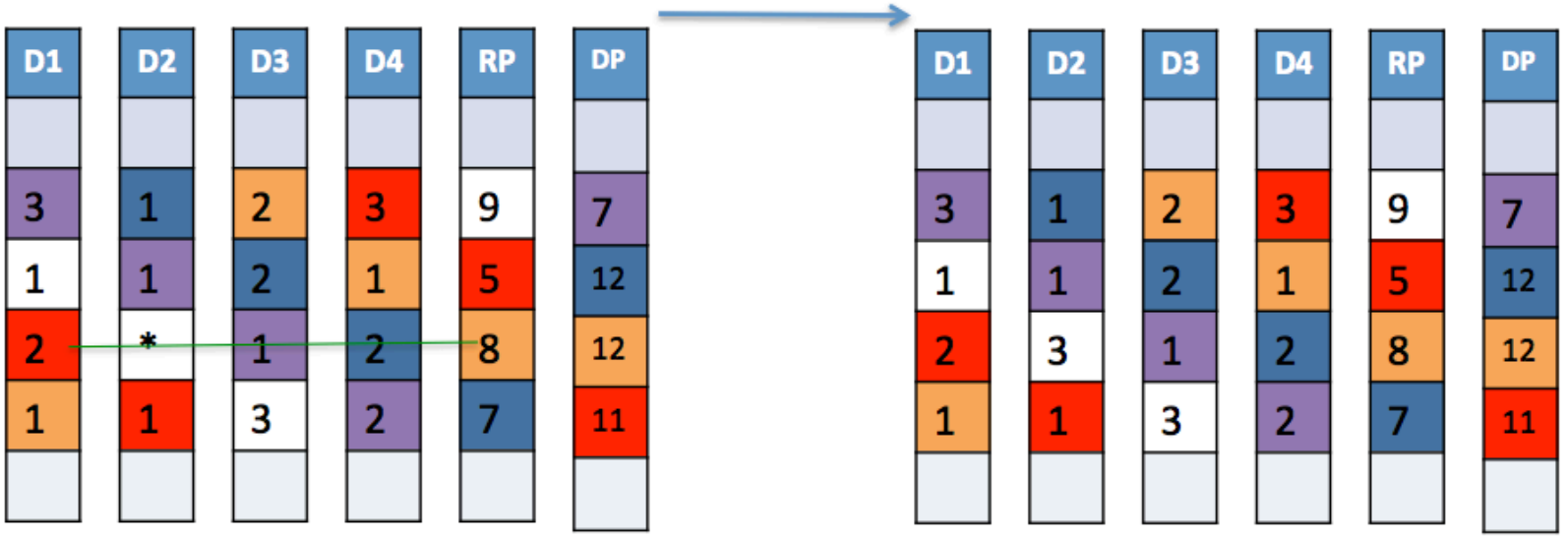
Next using row parity gives 7 – 2 – 3 – 1 = 1 for the missing block 4 on D2.



The red diagonal stripe gives 11 – 5 – 3 – 1 = 2 for the missing block 3 on D1.

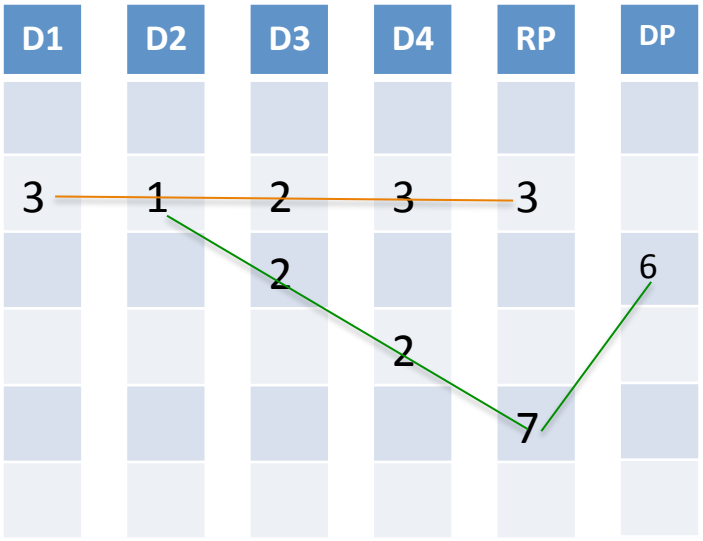


The final missing block on D2 is obtained using row parity as 8 – 2 – 1 – 2 = 3 to complete restoration of data blocks on the two missing disks.



Keep in mind that the XOR function would normally be used instead of addition. Addition was only used in this example to make it easier to follow.

The following example shows how the XOR (rather than addition and subtraction) is used to compute the horizontal or row parity and diagonal parity for the system. The symbol ^ is used to denote the XOR operator. XOR is used in computing parity and reconstructing missing blocks.



Row parity RP = 3^1^2^3 = 3
If D2 is lost, its data block can be computed as the XOR of the parity with the remaining data blocks in the row.
D2 = 3 ^ 2 ^ 3 ^ 3 = 1

DP = 1^2^2^7 = 6 (diagonal parity)

Module 11 Example Set 4

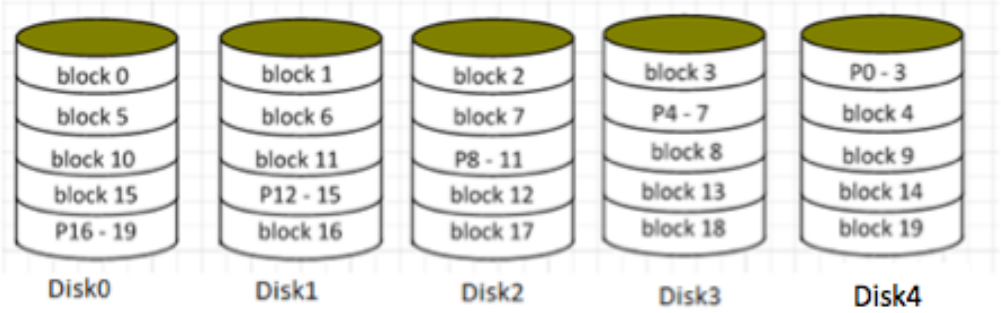
1. An Intel 80386 processor employs a bus cycle consisting of 2 clock cycles to access memory (i.e. to read or write to memory) over a 32-bit non-multiplexed data bus. If this system is driven by a 33-MHz clock, what is the bus bandwidth for:

a) a zero-wait state memory b) a one-wait state memory

a) A 33MHz clock rate corresponds to a cycle time of 30ns. With zero wait states the memory cycle time is 2*30ns=60ns. This corresponds to a bus bandwidth of 4bytes/60ns or 66.67 million bytes per second.

b) With one wait state the memory cycle time is 90ns which corresponds to a bus bandwidth of 4bytes/90ns = 44.44 million bytes per second.

2. Consider the following RAID5 system that uses a simple bit-wise XOR parity scheme:



Suppose that disk2 fails and is replaced by a new blank disk which is to be written with the information that had been on the original disk2.

a) Write down an expression for the parity block and for each of the four data blocks that should be written to the new disk2.

P8-11 = _____block10 XOR block11 XOR block8 XOR block9_____
Block2 = ____block0 XOR block1 XOR block3 XOR P0-3___
Block7 = ____ block5 XOR block6 XOR block4 XOR P4-7_____
Block12 = ___ block15 XOR block14 XOR block13 XOR P12-15_____
Block17 = ___ block16 XOR block18 XOR block19 XOR P16-19___

b) Assume that a disk read or write operation takes T time units and that the time to compute the XOR of two or more strips is also T time units. What is the minimum total number of time units required to reconstruct the complete new disk2 image? That is, if the minimum total time required to reconstruct disk2 is N*T, what value is N?

Blocks on separate disks can be read in parallel (T units since each block is read from a different disk), the XOR takes T units (they are done in parallel), then the reconstructed block is written to disk2 (T units). Therefore a minimum of 3T units are required to reconstruct and write each of the missing blocks. Hence a total of 5*3T = 15T is required. So N=15.

3. Explain which type of bus (synchronous or asynchronous) would be most appropriate for handling communications between the CPU and:
a) a mouse – **an asynchronous bus would work best in this case since the input from the mouse occurs at unpredictable times and only a small amount of data is transmitted by the mouse.**

b) a memory controller – **information must be transmitted between memory and the CPU at a relatively high rate to keep the CPU from stalling. Synchronous buses avoid the overhead of the handshake signals used with an asynchronous bus and so would be better in this case.**

4. Explain whether bus skew is more of a problem for
a) serial bus or for a parallel bus – **this would be more of a problem for parallel buses, since all of the bits within a byte or word would be sent at the same time over separate pathways. The difference in the travel time for the various bits would have to be below a specified value if the information content is to be correctly interpreted. With a serial bus there would only be one pathway over which all of the bits travel one after the other.**

b) a synchronous bus or for an asynchronous bus – **with a synchronous bus, each step in the transaction is expected to occur after at a specific time, but for the asynchronous bus handshake signals are exchanged to insure that each step has completed before the next step begins. Hence bus skew would be more of a problem for the synchronous bus since differences in the travel time between the bits that are transmitted could violate the fixed bus schedule. With the asynchronous bus, there is no preset schedule- the handshake signals control the progress of the transaction.**

Module 11 Example Set 5

1. Assume that each stripe within a certain RAID5 system contains 6 strips or blocks B0, B1, B2, B3, B4 and the corresponding parity strip P0-4.
a) Write down an expression for P0-4 as a function of the other strips.
P0-4 = B0 ^ B1 ^ B2 ^ B3 ^ B4 where ^ denotes the exclusive OR operation.

b) Complete the following equation:
B2 ^ B3 ^ P0-4 = _____
The answer is B0 ^ B1 ^ B4 since
B2 ^ B3 ^ P0-4 = B2 ^ B3 ^ B0 ^ B1 ^ B2 ^ B3 ^ B4
= B2 ^ B2 ^ B3 ^ B3 ^ B0 ^ B1 ^ B4 = 0 ^ 0 ^ B0 ^ B1 ^ B4 = B0 ^ B1 ^ B4

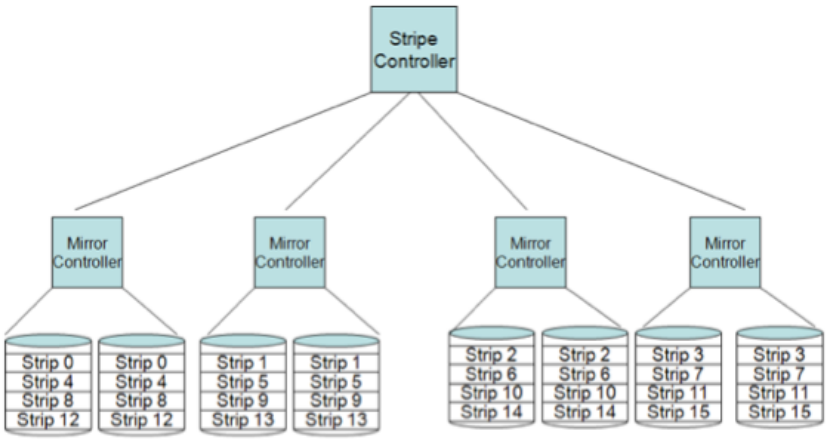
2. Two blocks are to be written in parallel on a RAID system. What restrictions apply if the system is:

a) RAID 4
The two blocks must reside within the same stripe and the parity block for that stripe must be updated at the same time. If the two blocks were in different stripes, the two parity blocks could not be written in parallel since they are on the same disk. Each disk can only perform one read or one write at a time.

b) RAID 5
The two blocks as well as the parity block or blocks to which they correspond must all map to separate disks.

3. A RAID6 system contains 12 stripes each of which contains 6 data strips or blocks. How many separate disks are required for this system.
Each data strip within a stripe would reside on a separate disk and two disks are needed to accommodate the two independent parity functions.
Hence a total of 6+2 = 8 disks would be required.

4. Consider the RAID10 system shown below:



RAID 10

a) What is the minimum number of disks that have to fail to make the system inoperable?

The minimum number is 2. If the two disks in any of the mirrored pairs fail, then the system will be inoperable.

b) What is the maximum number of disks that can fail without making the system inoperable?

The maximum number is 4. One disk in each of the four mirrored pairs can fail and the system can continue to operate by using the mirror images.

Computer Science 605.411

Module 11 Example Set 6

1. On what concepts are RAID systems based?
   RAID systems employ multiple disk drives each of which can perform a separate access in parallel. By acting in parallel the array of smaller less expensive independent disks can provide improved performance over a single large more expensive disk. The interface presented by the RAID controller appears to the programmer or operating system to be the same as that for a single disk system. One or more of the disk drives in the array can be used to hold parity information that allows data lost, due to some event such as a head crash, to be reconstructed.

2. Which of the RAID levels or categories is not capable of providing fault tolerance?
   RAID 0 systems include no parity or error correcting information and therefore become inoperable due to lost data. RAID 0 improves performance by striping or distributing data across multiple disk drives operating in parallel, but provide no data recovery capability.

3. Which of the RAID levels is the most expensive for a given amount of data storage?
   RAID 1 is most expensive since it duplicates each of the data disk to produce mirror images or shadow disks. The mirror images serve as backups in the event of data loss. So RAID 1 doubles the cost for a given amount of stored data.

4. Why do reads tend to be faster than writes on RAID 1 systems?
   Since the disks are not synchronized (i.e., the access heads are not at the same position on a data disk and on its mirror image), data to be read is taken from the first of the two copies that is encountered. Writes require that the data be written twice (to the desired block on both images) which takes longer because the two writes are not totally overlapped.

5. Why is RAID 2 not used commercially?
   RAID 2 employs striping at the bit level (as few as one bit from each record may be on separate disks) and employs error correcting codes that have relatively high overhead. For example, four data disks may require as many as 3 parity disks for a 75% overhead. While this is less than the 100% overhead for RAID 1, it is still unacceptably high. Current inexpensive disk drives include their own internal error-correcting capability at the bit level, so RAID 2 is not used.

6. For what type of I/O operations is RAID 3 most attractive?
   In a RAID3 system the access heads on all disk drives are ganged together and synchronized. Each transfer involves all of the disks. Hence RAID 3 works best when transferring large data blocks that span all or several of the disk drives. This means that only blocks or strips within the same stripe or row can be accessed at the same time in parallel. Accessing strips within different stripes requires separate I/O operations.

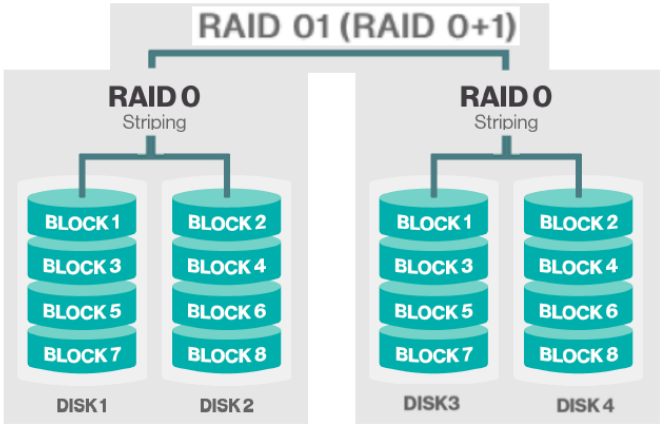
7. What is the main differences between RAID 4 and RAID 3?
   RAID 4 uses striping at the block level while RAID 3 uses bit-level striping. Hence the blocks or strips on a RAID 4 system tend to be larger than the RAID 3 counterparts. Both systems employ a single parity disk that must be accessed any time a stripe is modified. The access head on the disks within a RAID 4 system can move independently of each other, while those on the RAID 3 system move in unison.

8. What is one of the main advantages of RAID 4 over RAID 3?
   RAID 4 can support both large data transfers that span multiple blocks or multiple small transfers that map to separate blocks. However, the single parity disk can be a bottleneck and limit which writes can be done in parallel. Writing blocks in different stripes would require updating the parity block for each stripe; but the parity disk, as well as the other disks, can only perform one read or write at a time.

9. How is the parity disk bottleneck problem for RAID 4 eliminated in RAID 5?
   The parity blocks in RAID 5 systems are distributed among all of the disks rather than being concentrated on a single disk. This allows some combinations of writes to blocks within different stripes to be done in parallel along with updating the corresponding parity blocks.

10. When updating a stripe within a RAID 4 or RAID 5 system, the corresponding parity block for the stripe must also be updated. How can the new parity block be computed?
    The new parity block can be computed as the cumulative XOR of the new data block with all of the other data blocks within the stripe (this requires reading all of the other data blocks). Alternatively, the old data block and the old parity block for the stripe can be read and XORed. This has the effect of removing or subtracting out the old data block. The result can then be XORed with the new data block to obtain the new parity block.

11. How does a RAID01 (0+1) system differ from a RAID10 (1 + 0) system?
A RAID 01 system can be viewed as a RAID1 system in which each of the disks is a RAID0 system.



A RAID10 system can be viewed as a RAID0 system in which each of the disks has a mirror image.