GUIDED BY
PREETAM SUMAN
CSE2003

COMPUTER ARCHITECTUTRE AND ORGANIZATION

PRESENTATION

GROUP 13

TEAM MEMBERS

ADITYA BHATE

21BCE11228

NIKHIL SINGH

21BCE11237

AYAN GORAIN

21BCE11287

HARSH SEIWAL

21BCE11291

MAMTA KUMARI

21BCE11303

TOPICS

1. RAID ARCHITECTURE
2. PIPELINING
3. DATA HAZARDS
4. INSTRUCTIONAL HAZARDS

RAID

ARCHITECTURE

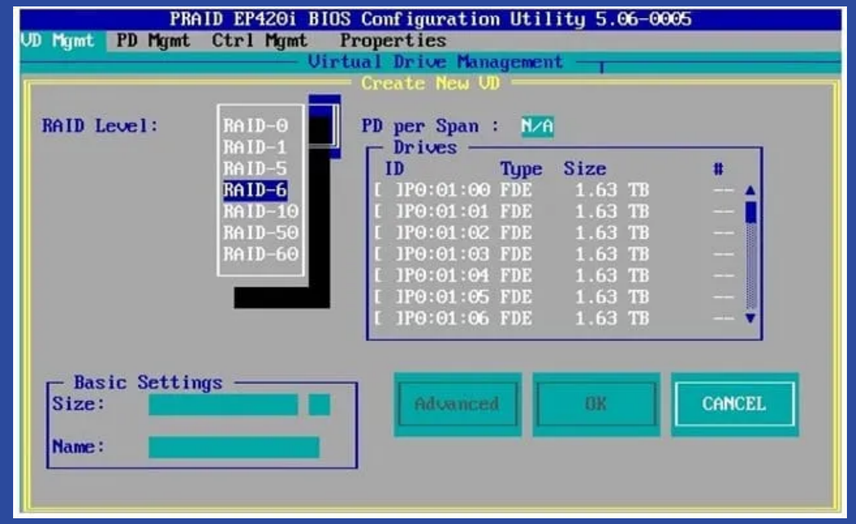
RAID

RAID is an acronym for Redundant Arrays of Independent/ Inexpensive Disks. RAID combines multiple, small, inexpensive disk drives into an Array of disk drives which is treated as a single storage unit (drive) by the computer

RAID Architecture Technology provides Data protection and Data consistency.

RAID plays a vital role in storing large amounts of data while preserving Data Integrity. It helps in real-time data recovery when a hard drive fails. In other words, RAID either divides or duplicates the task of one hard disk between multiple disks.

This is done to create data redundancy in case of a drive failure. RAID Mode also called Raid level that is set for different application.
E.g:
when mode “RAID 0” is set, then the system splits the data evenly between two or more disks.



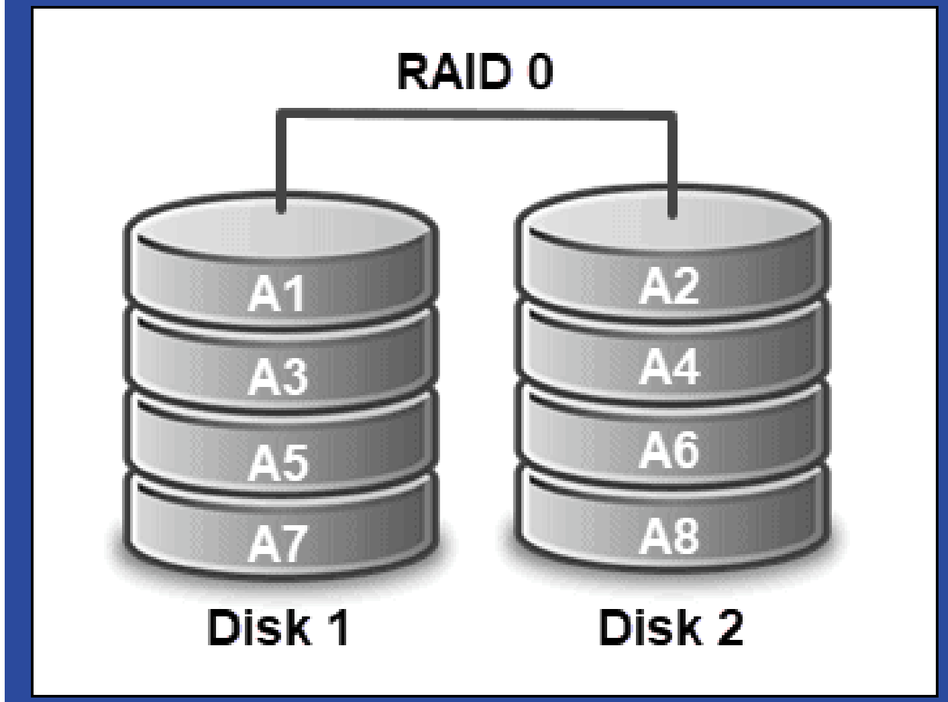
RAID 0

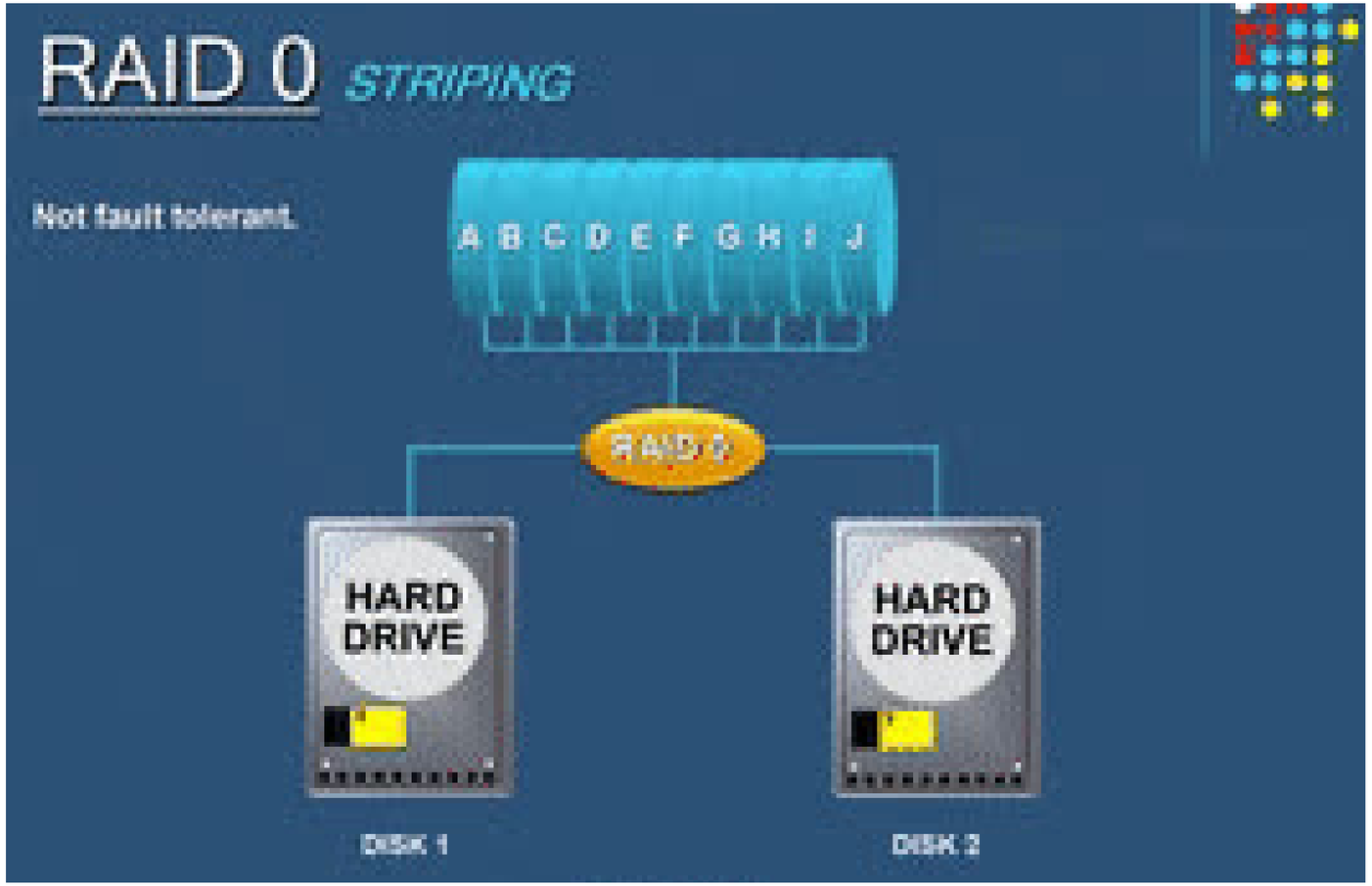
RAID 0 (also known as a stripe set or striped volume) splits ("stripes") data evenly across two or more disks, without parity information, redundancy, or fault tolerance. Since RAID 0 provides no fault tolerance or redundancy, the failure of one drive will cause the entire array to fail; as a result of having data striped across all disks, the failure will result in total data loss. This configuration is typically implemented having speed as the intended goal. RAID 0 is normally used to increase performance, although it can also be used as a way to create a large logical volume out of two or more physical disks.

A RAID 0 setup can be created with disks of differing sizes, but the storage space added to the array by each disk is limited to the size of the smallest disk. For example, if a 120 GB disk is striped together with a 320 GB disk, the size of the array will be 120 GB × 2 = 240 GB. However, some RAID implementations allow the remaining 200 GB to be used for other purposes.

The diagram in this section shows how the data is distributed into stripes on two disks, with A1:A2 as the first stripe, A3:A4 as the second one, etc. Once the stripe size is defined during the creation of a RAID 0 array, it needs to be maintained at all times. Since the stripes are accessed in parallel, an n-drive RAID 0 array appears as a single large disk with a data rate n times higher than the single-disk rate.

STRIPING




RAID 0 STRIPING
Not fault tolerant.
A B C D E F G H I J
RAID 0
HARD DRIVE
HARD DRIVE
DISK 1
DISK 2

PERFORMANCE

A RAID 0 array of n drives provides data read and write transfer rates up to n times as high as the individual drive rates, but with no data redundancy. As a result, RAID 0 is primarily used in applications that require high performance and are able to tolerate lower reliability, such as in scientific computing or computer gaming.

Some benchmarks of desktop applications show RAID 0 performance to be marginally better than a single drive.
Striping does not always increase performance (in certain situations it will actually be slower than a non-RAID setup), but in most situations it will yield a significant improvement in performance". Synthetic benchmarks show different levels of performance improvements when multiple HDDs or SSDs are used in a RAID 0 setup, compared with single-drive performance. However, some synthetic benchmarks also show a drop in performance for the same comparison.

RAID 0

ADVANTAGES OF RAID 0:

- COST-EFFICIENT AND STRAIGHTFORWARD TO IMPLEMENT.
- INCREASED READ AND WRITE PERFORMANCE.
- NO OVERHEAD (TOTAL CAPACITY USE).

DISADVANTAGES OF RAID 0:

- DOESN'T PROVIDE FAULT TOLERANCE OR REDUNDANCY.

WHEN RAID 0 SHOULD BE USED??

RAID 0 is used when performance is a priority and reliability is not. If you want to utilize your drives to the fullest and don't mind losing data, opt for RAID 0.

On the other hand, such a configuration does not necessarily have to be unreliable. You can set up disk striping on your system along with another RAID array that ensures data protection and redundancy.

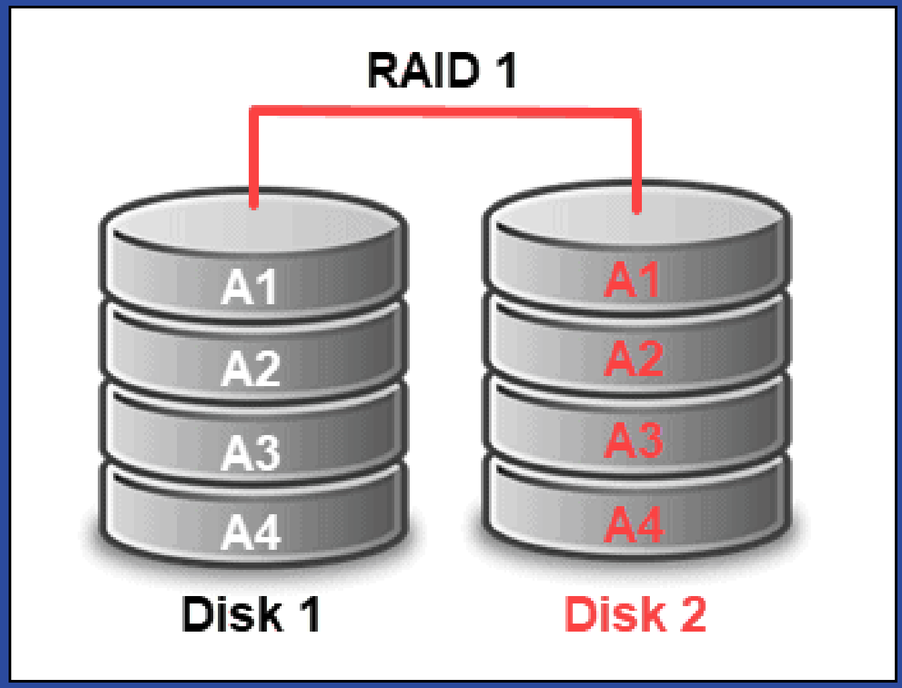
RAID 1

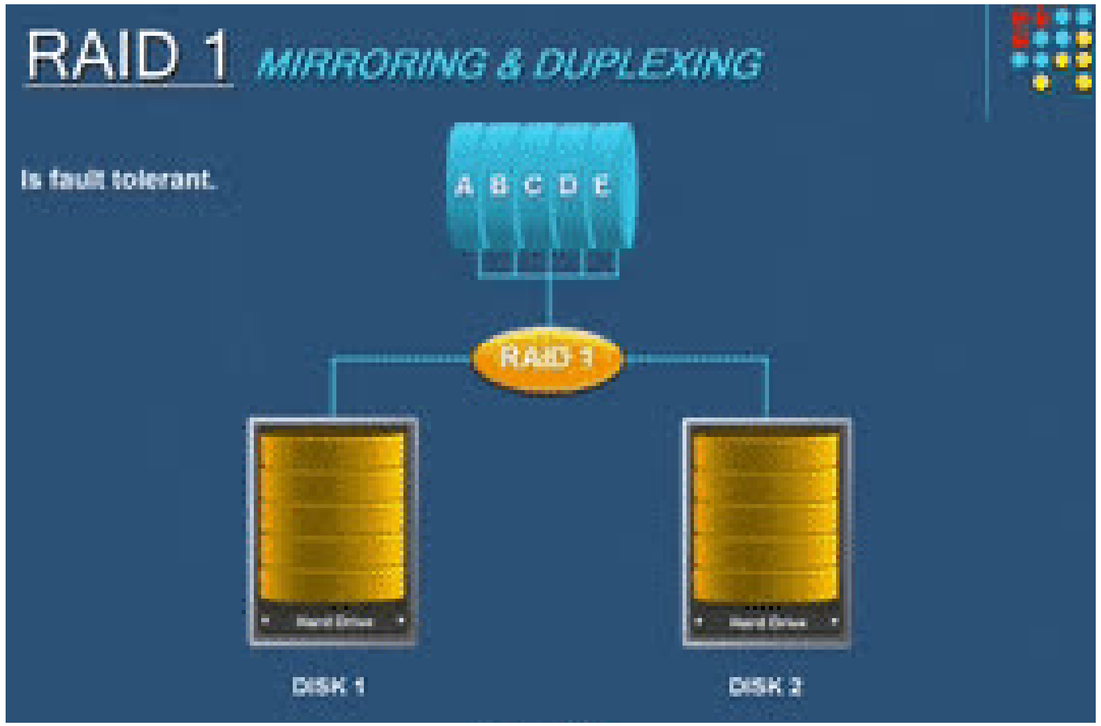
RAID 1 is an array consisting of at least two disks where the same data is stored on each to ensure redundancy. The most common use of RAID 1 is setting up a mirrored pair consisting of two disks in which the contents of the first disk is mirrored in the second. This is why such a configuration is also called mirroring.

Unlike with RAID 0, where the focus is solely on speed and performance, the primary goal of RAID 1 is to provide redundancy. It eliminates the possibility of data loss and downtime by replacing a failed drive with its replica.

in such a setup, the array volume is as big as the smallest disk and operates as long as one drive is operational. Apart from reliability, mirroring enhances read performance as a request can be handled by any of the drives in the array. On the other hand, the write performance remains the same as with one disk and is equal to the slowest disk in the configuration.

MIRRORING




RAID 1
MIRRORING & DUPLEXING
Is fault tolerant.
A B C D E
RAID 1
DISK 1
DISK 2

PERFORMANCE

ADVANTAGES OF RAID 1:

- INCREASED READ PERFORMANCE.
- PROVIDES REDUNDANCY AND FAULT TOLERANCE.
- SIMPLE TO CONFIGURE AND EASY TO USE.

DISADVANTAGES OF RAID 1:

- USES ONLY HALF OF THE STORAGE CAPACITY.
- MORE EXPENSIVE (NEEDS TWICE AS MANY DRIVERS).
- REQUIRES POWERING DOWN YOUR COMPUTER TO REPLACE FAILED DRIVE.

WHEN RAID 1 SHOULD BE USED??

RAID 1 is used for mission-critical storage that requires a minimal risk of data loss. Accounting systems often opt for RAID 1 as they deal with critical data and require high reliability.

It is also suitable for smaller servers with only two disks, as well as if you are searching for a simple configuration you can easily set up (even at home).

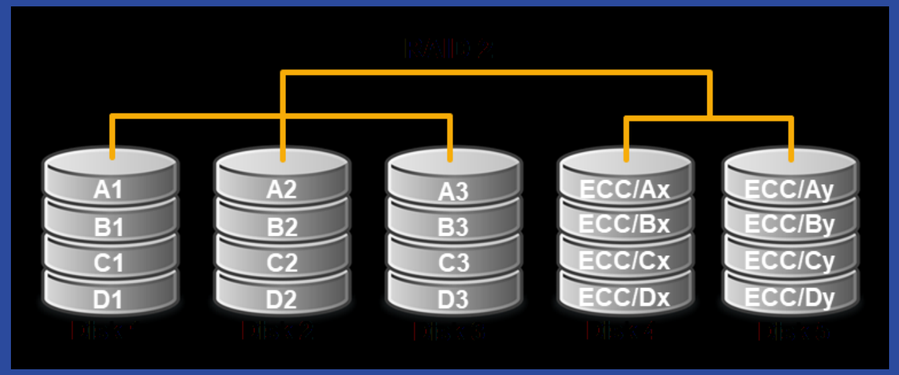
RAID 2

BIT-LEVEL STRIPING WITH DEDICATED HAMMING-CODE PARITY

RAID 2 is rarely used in practice today. It combines bit-level striping with error checking and information correction. This RAID implementation requires two groups of disks – one for writing the data and another for writing error correction codes. RAID 2 also requires a special controller for the synchronized spinning of all disks.

Instead of data blocks, RAID 2 stripes data at the bit level across multiple disks. Additionally, it uses the Humming error Code correction (ECC) and stores this information on the redundancy disk.

The array calculates the error code correction on the fly. While writing the data, it strips it to the data disk and writes the code to the redundancy disk. On the other hand, while reading data from the disk, it also reads from the redundancy disk to verify the data and make corrections if needed.



RAID 2

ADVANTAGES OF RAID 2:

- RELIABILITY.
- THE ABILITY TO CORRECT STORED INFORMATION.

DISADVANTAGES OF RAID 2:

- EXPENSIVE.
- DIFFICULT TO IMPLEMENT.
- REQUIRE ENTIRE DISKS FOR ECC (ERROR CODE CORRECTION)

WHEN RAID 2 SHOULD BE USED??

RAID 2 is not a common practice today as most of its features are now available on modern hard disks.

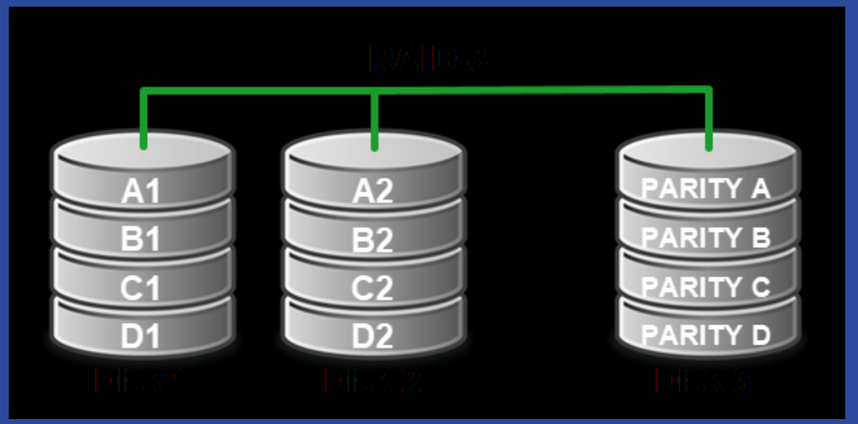
Due to its cost and implementation requirements, this RAID level never became popular among developers.

RAID 3

BIT-LEVEL STRIPING WITH DEDICATED PARITY

Like RAID 2, RAID 3 is rarely used in practice. This RAID implementation utilizes bit-level striping and a dedicated parity disk. Because of this, it requires at least three drives, where two are used for storing data strips, and one is used for parity.

To allow synchronized spinning, RAID 3 also needs a special controller. Due to its configuration and synchronized disk spinning, it achieves better performance rates with sequential operations than random read/write operations.



RAID 3

ADVANTAGES OF RAID 3:

- GOOD THROUGHPUT WHEN TRANSFERRING LARGE AMOUNTS OF DATA.
- HIGH EFFICIENCY WITH SEQUENTIAL OPERATIONS.
- DISK FAILURE RESILIENCY.

DISADVANTAGES OF RAID 3:

- NOT SUITABLE FOR TRANSFERRING SMALL FILES.
- COMPLEX TO IMPLEMENT.
- DIFFICULT TO SET UP AS SOFTWARE RAID.

WHEN RAID 10 SHOULD BE USED??

RAID 10 is often used in use cases that require storing high volumes of data, fast read and write times, and high fault tolerance.

Accordingly, this RAID level is often implemented for email servers, web hosting servers, and databases.

RAID 4

BLOCK-LEVEL STRIPING WITH DEDICATED PARITY

RAID 4 is another unpopular standard RAID level. It consists of block-level data striping across two or more independent disks and a dedicated parity disk.

The implementation requires at least three disks – two for storing data strips and one dedicated for storing parity and providing redundancy. As each disk is independent and there is no synchronized spinning, there is no need for a controller.

RAID 4 configuration is prone to bottlenecks when storing parity bits for each data block on a single drive. Such system bottlenecks have a large impact on system performance.

RAID 4

ADVANTAGES OF RAID 4:

- FAST READ OPERATIONS.
- LOW STORAGE OVERHEAD.
- SIMULTANEOUS I/O REQUESTS.

DISADVANTAGES OF RAID 4:

- BOTTLENECKS THAT HAVE BIG EFFECT ON OVERALL PERFORMANCE.
- SLOW WRITE OPERATIONS.
- REDUNDANCY IS LOST IF THE PARITY DISK FAILS.

WHEN RAID 4 SHOULD BE USED??

RAID 4 works best with use cases requiring sequential reading and writing data processes of huge files.

Still, just like with RAID 3, in most solutions, RAID 4 has been replaced with RAID 5.

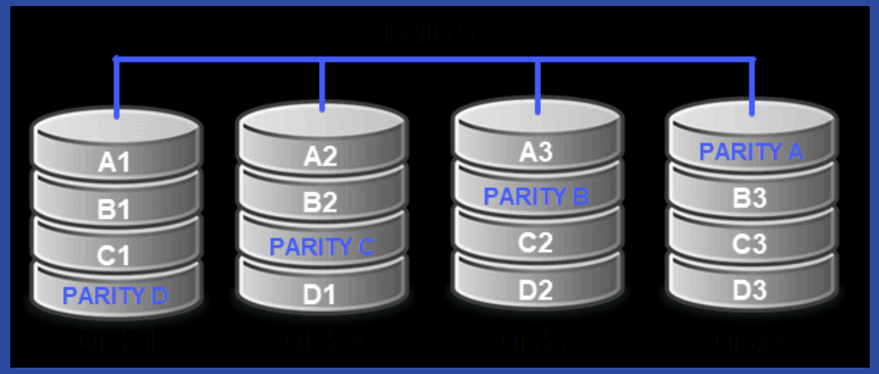
RAID 5

RAID 5 is considered the most secure and most common RAID implementation. It combines striping and parity to provide a fast and reliable setup. Such a configuration gives the user storage usability as with RAID 1 and the performance efficiency of RAID 0.

This RAID level consists of at least three hard drives (and at most, 16). Data is divided into data strips and distributed across different disks in the array. This allows for high performance rates due to fast read data transactions which can be done simultaneously by different drives in the array.

Parity bits are distributed evenly on all disks after each sequence of data has been saved. This feature ensures that you still have access to the data from parity bits in case of a failed drive. Therefore, RAID 5 provides redundancy through parity bits instead of mirroring.

STRIPING WITH PARITY




RAID 5
STRIPING WITH PARITY
Requires 3 or more disks.
A B C D E F G H I J K L
RAID 5
Hard Drive
Hard Drive
Hard Drive
Hard Drive
DISK 1
DISK 2
DISK 3
DISK 4

RAID 5

ADVANTAGES OF RAID 5:

- HIGH PERFORMANCE AND CAPACITY.
- FAST AND RELIABLE READ SPEED.
- TOLERATES SINGLE DRIVE FAILURE.

DISADVANTAGES OF RAID 5:

- LONGER REBUILD TIME.
- USES HALF OF THE STORAGE CAPACITY (DUE TO PARITY).
- IF MORE THAN ONE DISK FAILS, DATA IS LOST.
- MORE COMPLEX TO IMPLEMENT.

WHEN RAID 5 SHOULD BE USED??

RAID 5 is often used for file and application servers because of its high efficiency and optimized storage.

**Additionally,
it is the best, cost-effective solution if continuous data access is a priority and/or you require installing an operating system on the array.**

RAID 6

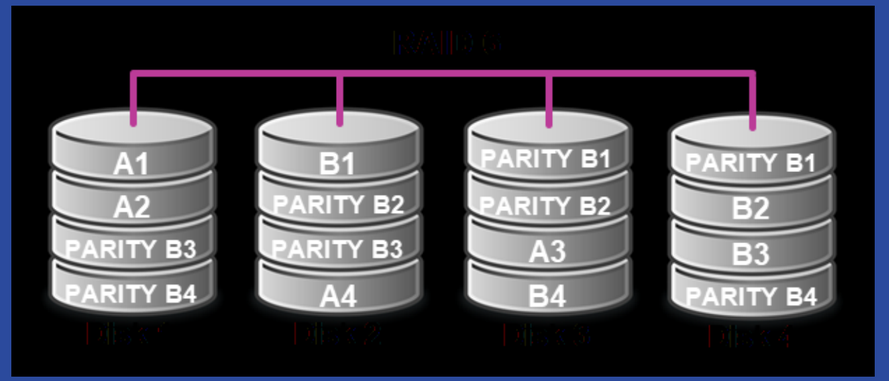
STRIPING WITH DOUBLE PARITY

RAID 6 is an array similar to RAID 5 with an addition of its double parity feature. For this reason, it is also referred to as the double-parity RAID.

This setup requires a minimum of four drives. The setup resembles RAID 5 but includes two additional parity blocks distributed across the disk. Therefore, it uses block-level striping to distribute the data across the array and stores two parity blocks for each data block.

Block-level striping with two parity blocks allows two disk failures before any data is lost. This means that in an event where two disks fail, RAID can still reconstruct the required data.

Its performance depends on how the array is implemented, as well as the total number of drives. Write operations are slower compared to other configurations due to its double parity feature.



RAID 6

ADVANTAGES OF RAID 6:

- HIGH FAULT AND DRIVE-FAILURE TOLERANCE.
- STORAGE EFFICIENCY (WHEN MORE THAN FOUR DRIVES ARE USED).
- FAST READ OPERATIONS.

DISADVANTAGES OF RAID 6:

- REBUILD TIME CAN TAKE UP TO 24 HOURS.
- SLOW WRITE PERFORMANCE.
- COMPLEX TO IMPLEMENT.
- MORE EXPENSIVE.

WHEN RAID 6 SHOULD BE USED??

RAID 6 is a good solution for mission-critical applications where data loss cannot be tolerated.

Therefore, it is often used for data management in defense sectors, healthcare, and banking.

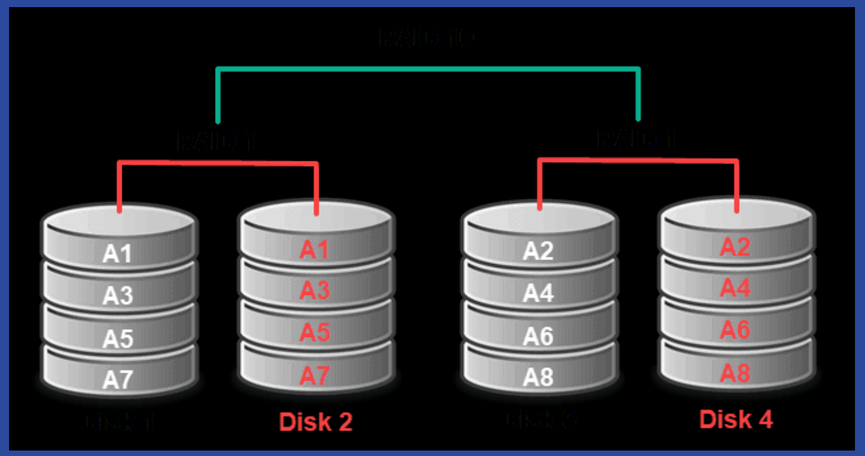
RAID 10

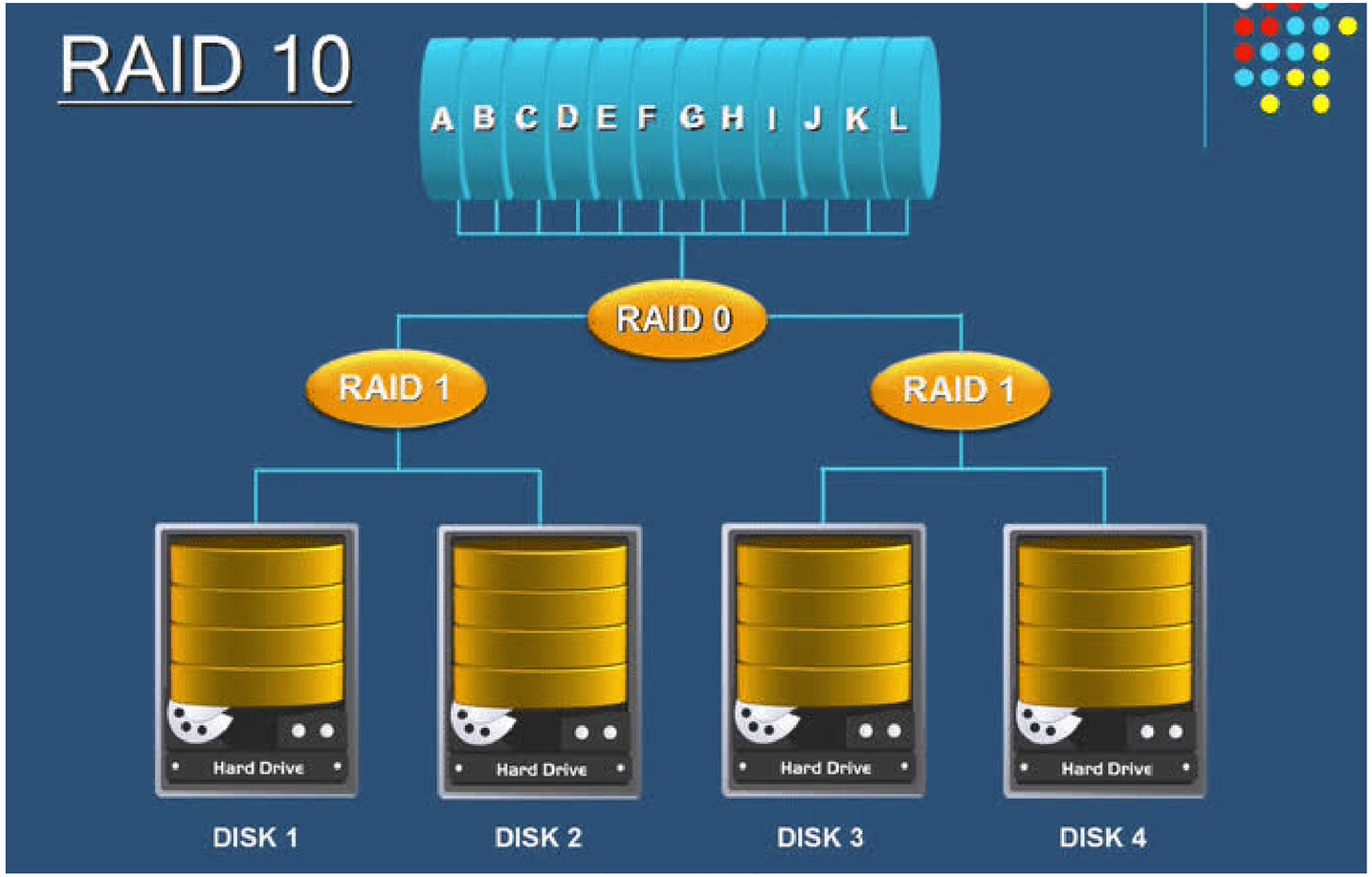
RAID 10 is part of a group called nested or hybrid RAID, which means it is a combination of two different RAID levels. In the case of RAID 10, the array combines level 1 mirroring and level 0 striping. This RAID array is also known as RAID 1+0.

RAID 10 uses logical mirroring to write the same data on two or more drives to provide redundancy. If one disk fails, there is a mirrored image of the data stored on another disk. Additionally, the array uses block-level striping to distribute chunks of data across different drives. This improves performance and read and write speed as the data is simultaneously accessed from multiple disks.

To implement such a configuration, the array requires at least four drives, as well as a disk controller.

MIRRORING WITH STRIPING




RAID 10
A B C D E F G H I J K L
RAID 0
RAID 1
RAID 1
Hard Drive
Hard Drive
Hard Drive
Hard Drive
DISK 1
DISK 2
DISK 3
DISK 4

NESTED (HYBRID) RAID

SOME POPULAR HYBRID RAID LEVELS INCLUDE:

- RAID 01 (STRIPING AND MIRRORING; ALSO KNOWN AS “MIRROR OF STRIPES”)
- RAID 03 (BYTE-LEVEL STRIPING AND DEDICATED PARITY)
- RAID 10 (DISK MIRRORING AND STRAIGHT BLOCK-LEVEL STRIPING)
- RAID 50 (DISTRIBUTED PARITY AND STRAIGHT BLOCK-LEVEL STRIPING)
- RAID 60 (DUAL PARITY AND STRAIGHT BLOCK-LEVEL STRIPING)
- RAID 100 (A STRIPE OF RAID 10S)

HOW HYBRID RAID ARE NAMED??

Hybrid RAID implementations are named after the RAID levels they incorporate.

In most cases, they include two numbers where their order represents the layering scheme.

RAID 10

ADVANTAGES OF RAID 10:

- HIGH PERFORMANCE.
- HIGH FAULT-TOLERANCE.
- FAST READ AND WRITE OPERATIONS.
- FAST REBUILD TIME.

DISADVANTAGES OF RAID 10:

- LIMITED SCALABILITY.
- COSTLY (COMPARED TO OTHER RAID LEVELS).
- USES HALF (50%) OF THE DISK SPACE CAPACITY.
- MORE COMPLICATED TO SET UP.

WHEN RAID 10 SHOULD BE USED??

RAID 10 is often used in use cases that require storing high volumes of data, fast read and write times, and high fault tolerance.

Accordingly, this RAID level is often implemented for email servers, web hosting servers, and databases.

2

PIPELINING

BY ADITYA BHATE

DATA PIPELINING

Pipelining is a process of arrangement of hardware elements of the CPU such that its overall performance is increased. Simultaneous execution of more than one instruction takes place in a pipelined processor.

WHY PIPELINING ?

To improve the performance of a CPU we have two options:

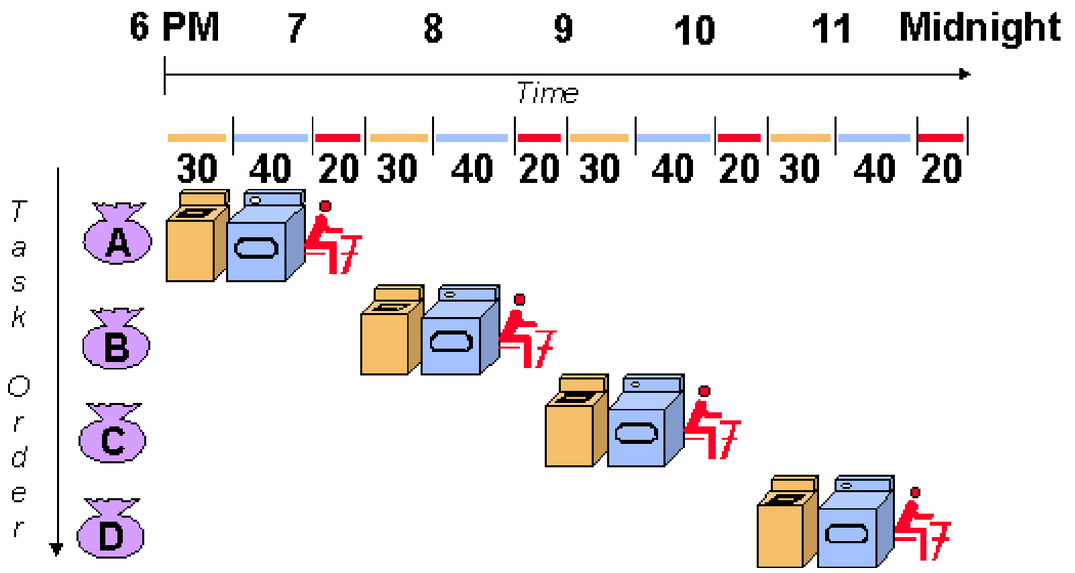
1) Improve the hardware by introducing faster circuits.

2) Arrange the hardware such that more than one operation can be performed at the same time.

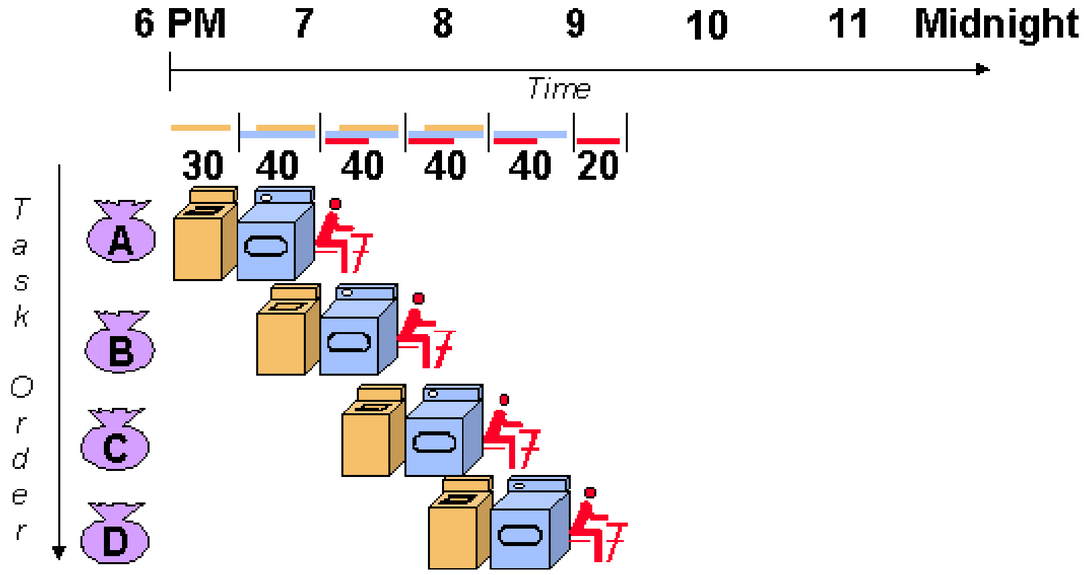
(Since there is a limit on the speed of hardware and the cost of faster circuits is high).

EX:

BEFORE PIPELINING



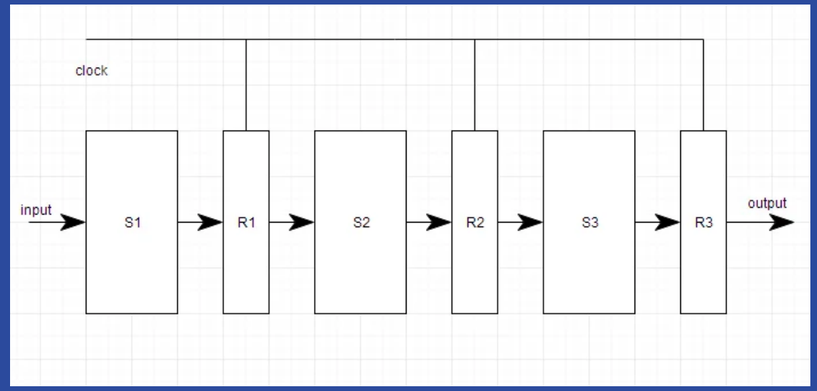
AFTER PIPELINING



Implementing Pipelining

To implement/design a basic pipeline:

- In a pipelined processor, a pipeline has two ends, the input end, and the output end. Between these ends, there are multiple stages/segments such that the output of one stage is connected to the input of the next stage and each stage performs a specific operation.
- Interface registers are used to hold the intermediate output between two stages. These interface registers are also called latch or buffer.
- All the stages in the pipeline along with the interface registers are controlled by a common clock.



Execution in a Pipelined Processor

The execution sequence of instructions in a pipelined processor can be visualized using a space-time diagram. For example, consider a processor having 4 stages and let there be 2 instructions to be executed. We can visualize the execution sequence through the following space-time diagrams:

Non-overlapped execution:

Stage / Cycle	1	2	3	4	5	6	7	8
S1	I_1				I_2			
S2		I_1				I_2		
S3			I_1				I_2	
S4				I_1				I_2

Total time = 8 Cycle

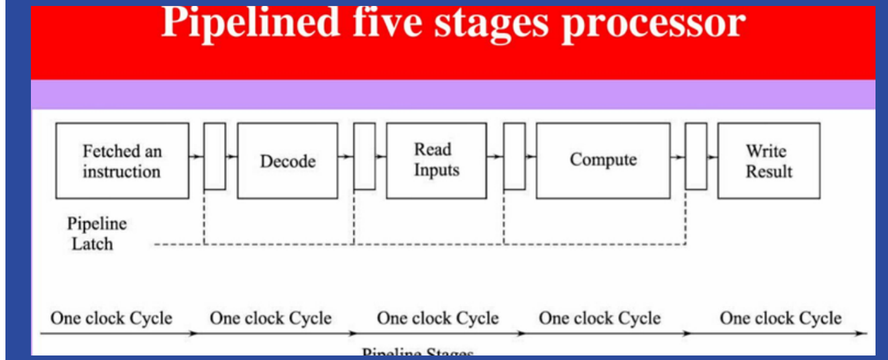
Overlapped execution:

Stage / Cycle	1	2	3	4	5
S1	I_1	I_2			
S2		I_1	I_2		
S3			I_1	I_2	
S4				I_1	I_2

Execution in a Pipelined Processor

Pipeline Stages RISC processor has 5 stage instruction pipeline to execute all the instructions in the RISC instruction set.

- Stage 1 (Instruction Fetch) the CPU reads instructions from the address in the memory whose value is present in the program counter.
- Stage 2 (Instruction Decode), Instruction is decoded and the register file is accessed to get the values from the registers used in the instruction.
- Stage 3 (Instruction Execute) Operations are performed.
- Stage 4 (Memory Access) memory operands are read and written from/to the memory that is present in the instruction.
- Stage 5 (Write Back) computed/fetched value is written back to the register present in the instructions.



Performance of a Pipelined Processor

Consider a ‘k’ segment pipeline with clock cycle time as ‘Tp’. Let there be ‘n’ tasks to be completed in the pipelined processor. Now, the first instruction is going to take ‘k’ cycles to come out of the pipeline but the other ‘n – 1’ instructions will take only ‘1’ cycle each, i.e, a total of ‘n – 1’ cycles. So, time taken to execute ‘n’ instructions in a pipelined processor:

$$ET_{pipeline} = k + n - 1 \text{ cycles}$$
$$= (k + n - 1)\ Tp$$

So, speedup (S) of the pipelined processor over the non-pipelined processor, when ‘n’ tasks are executed on the same processor is:

$$ET_{non\text{-}pipeline} = n * k * Tp$$

$$S = \text{Performance of pipelined processor} / \text{Performance of non-pipelined processor}$$

Performance of a Pipelined Processor

Some other formulae in pipelining

Clock cycle = (no. of stages) + (no. of instructions - 1)

Total Time = stages * Time period + (no. of instruction - 1) * Time period

Clock per instruction (CPI) = Clock / instruction

3

DATA HAZARDS

DATA HAZARDS

Data Hazards occur when an instruction depends on the result of previous instruction and that result of instruction has not yet been computed. whenever two different instructions use the same storage. the location must appear as if it is executed in sequential order.

DATA HAZARDS

1. READ AFTER WRITE (RAW)
2. WRITE AFTER READ (WAR)
3. WRITE AFTER WRITE (WAW)

Read after Write (RAW)

- It is also known as True dependency or Flow dependency. It occurs when the value produced by an instruction is required by a subsequent instruction.
- Stalls are required to handle these hazards.

Example:

ADD R1, --, --;

SUB --, R1, --;

Write after Read (WAR)

It is also known as anti dependency. These hazards occur when the output register of an instruction is used right after read by a previous instruction.

Example:

ADD --, R1, --;

SUB R1, --, --;

Write after Write (WAW)

It is also known as output dependency. These hazards occur when the output register of an instruction is used for write after written by previous instruction.

Example:

ADD R1, --, --;

SUB R1, --, --;

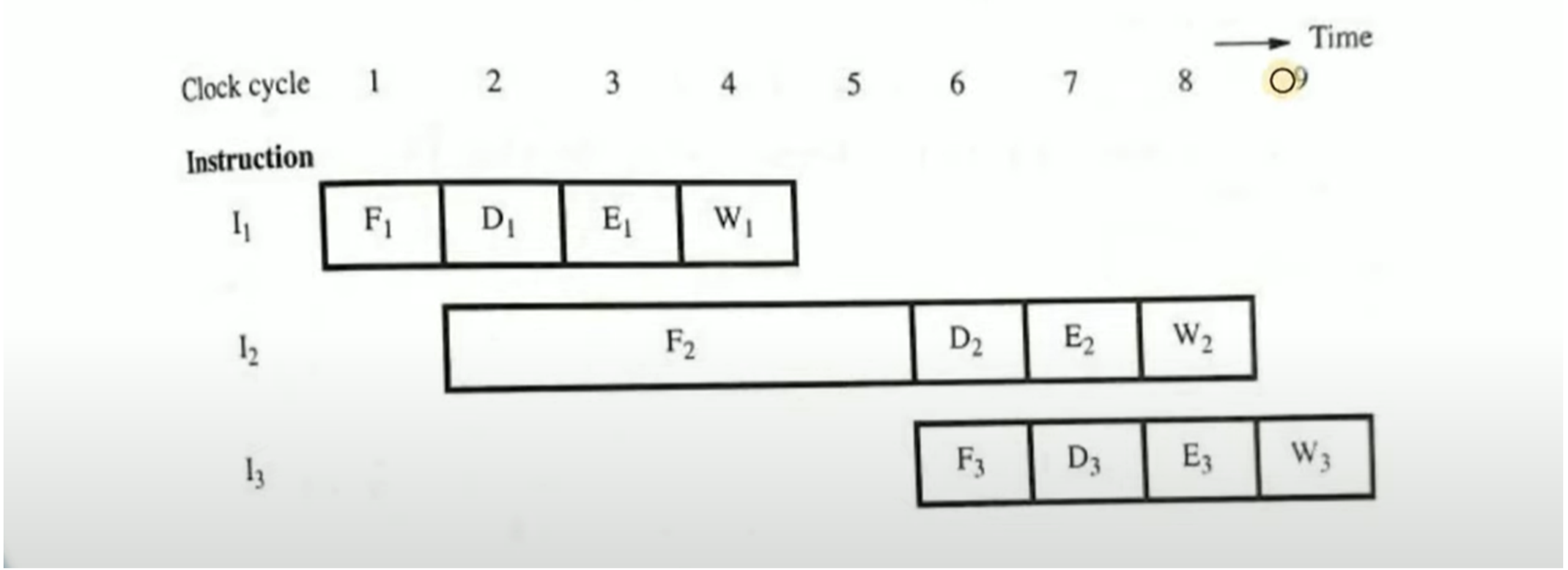
4

INSTRUCTION HAZARDS

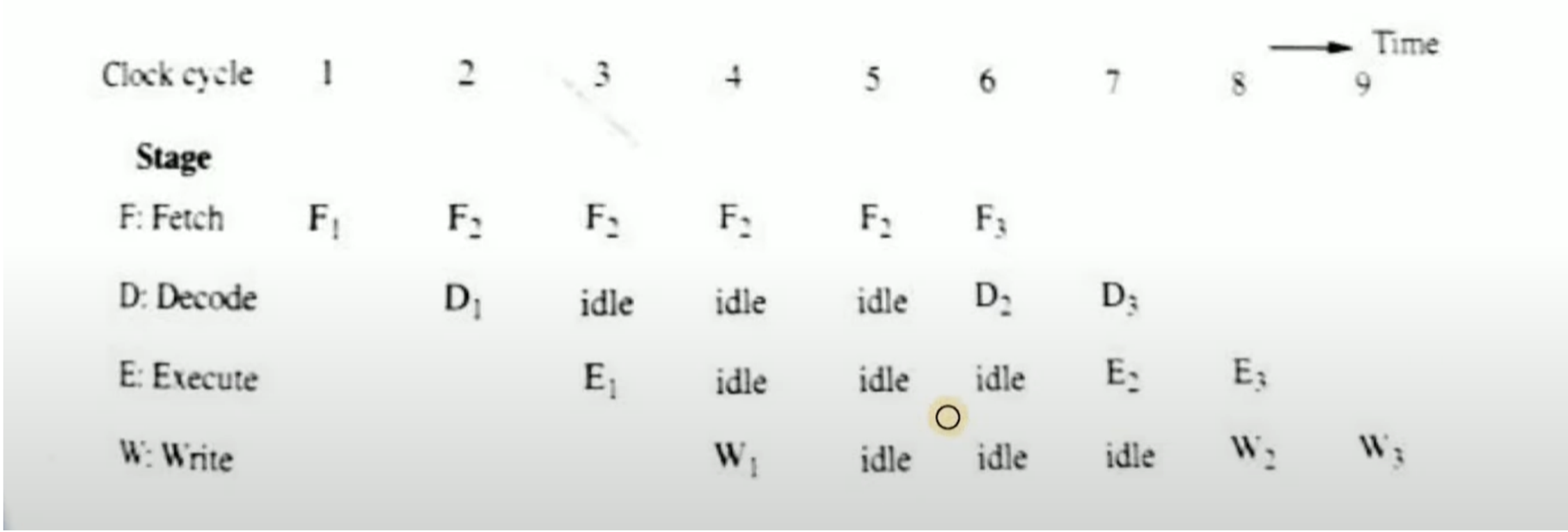
INSTRUCTION HAZARDS

- In, instruction hazard the pipeline will get stop or it will get stalled because of delay in the availability of an instruction so, either the delay in instruction will happened or it may not be available. So during this circumstances instruction hazard can occur.
- For example,this may be a result of a miss in the cache,requiring the instruction to be fetched from the main memory.
- Such hazards are often called control hazards or instruction hazards.

The effect of a cache miss on pipelined operation is shown below :



Let the branch target be instruction Ik.



Clock cycle	1	2	3	4	5	6	7	8	9 → Time
Stage									
F: Fetch	F_1	F_2	F_2	F_2	F_2	F_3			
D: Decode		D_1	idle	idle	idle	D_2	D_3		
E: Execute			E_1	idle	idle	idle	E_2	E_3	
W: Write				W_1	idle	idle	idle	W_2	W_3

THANK YOU