Bus Arbitration in Computer Organisation Architecture

Bus Arbitration in Computer Organisation Architecture

- A device that initiates data transfers on the bus at any given time is called a bus master.
- In a computer system, there may be more than one bus master such as a DMA controller or a processor etc.
- These devices share the system bus and when a current master bus relinquishes another bus can acquire the control of the processor.
- Trying to get access to a bus at the same time, but access can be given to only one of these
- System buses shared between the controller and processor
- It is the process by which the next device to become the bus master is selected and bus mastership is transferred to it.
- The selection of the bus master is usually done on the priority basis.
- The bus arbiter may be the processor or a separate controller connected to the bus.

Bus arbitration process

- Refers to a process by which the current bus master accesses and then leaves the control of the bus and passes it to another busrequesting processor unit

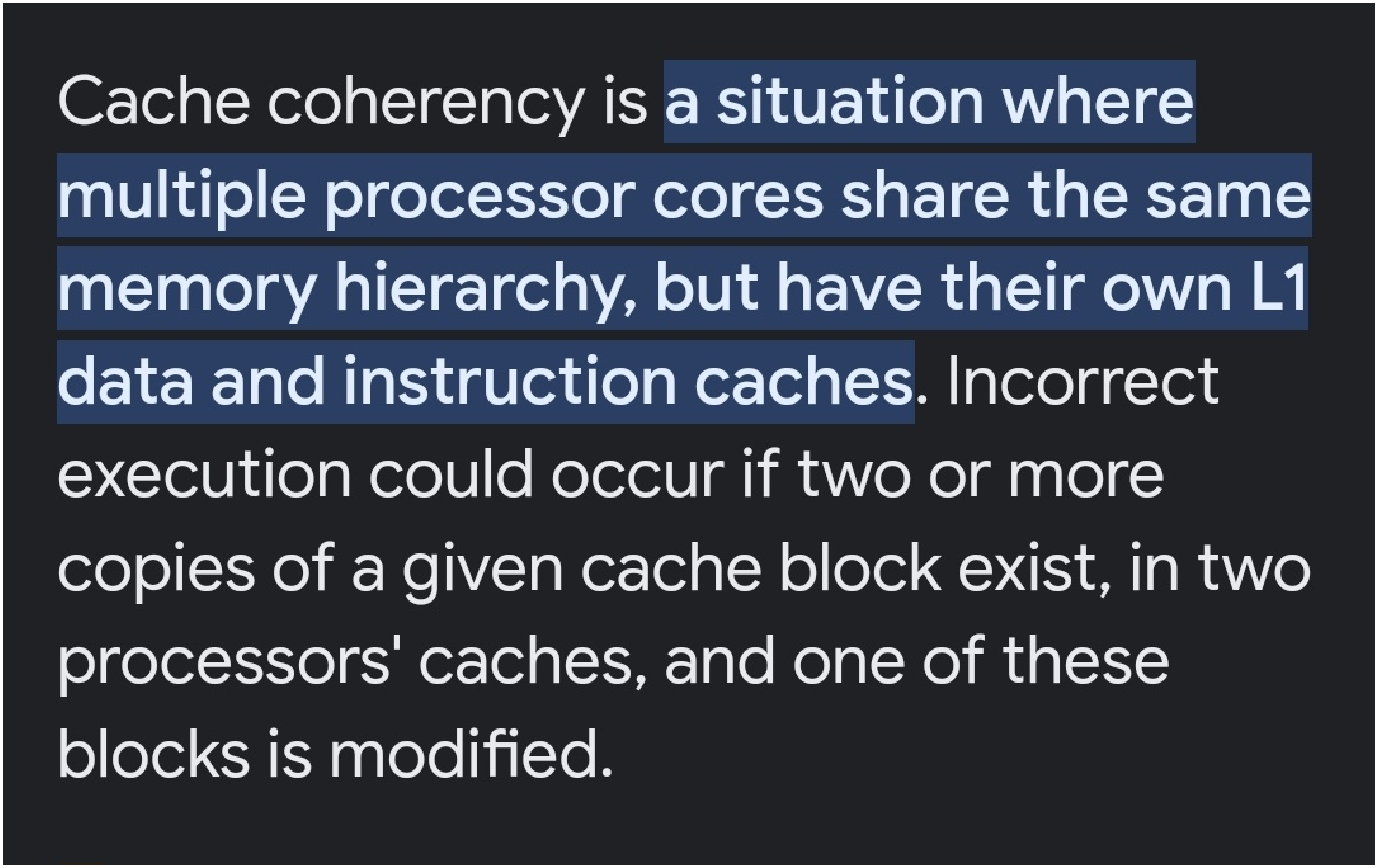
Types of Bus Arbitration

- There are two types of bus arbitration namely

1. Centralised

Arbitration. 2.

Distributed Arbitration.



Cache coherency is a situation where multiple processor cores share the same memory hierarchy, but have their own L1 data and instruction caches. Incorrect execution could occur if two or more copies of a given cache block exist, in two processors' caches, and one of these blocks is modified.

Instructions / Cycle	1	2	3	4	5	6
I_1	IF	ID	EX	MEM	WB	
I_2		IF	ID	EX	MEM	WB

When the instruction tries to enter into the write back stage of the pipeline, at that time, all the previous instructions contained by the program have already passed through the read stage of register and read their input values. Now without causing any type of problem, the write instruction can write its destination register. The WAR instructions contain less problems as compared to the WAW because in WAR, before the write back stage of a pipeline, the read stage of a register occur.

WAW

WAW can be referred to as **'Write after Write'.** It is also known as Output Data dependency. If the later instruction tries to write on operand before earlier instruction writes it, in this case, the WAW hazards will occur. The condition to detect the WAW hazard is when O_n and O_{n+1} both have a minimum one common operand.

For example:

The dependency is described as follows:

add **R1,** R2, R3

sub **R1,** R2, R4

Here addition instruction creates a WAW hazard because subtraction instruction writes on the same register. The hazard for instructions 'add **R1,** R2, R3' and 'sub **R1,** R2, R4' are described as follows:

Instructions / Cycle	1	2	3	4	5	6	7
I_1	IF	ID	EX	MEM	MEM2	MEM3	WB
I_2		IF	ID	EX	MEM	WB	

In the write back stage of a pipeline, the output register of instruction will be written. The order in which the instruction with WAW hazard appears in the program, in the same order these instructions will be entered the write back stage of a pipeline. The result of these instructions will be written into the register in the right order. The processor has improved performance as compared to the original program because it allows instructions to execute in different orders.

RAW:

RAW hazard can be referred to as **'Read after Write'.** It is also known as Flow/True data dependency. If the later instruction tries to read on operand before earlier instruction writes it, in this case, the RAW hazards will occur. The condition to detect the RAW hazard is when O_n and I_{n+1} both have a minimum one common operand.

For example:

I_1: add **R1**, R2, R3

I_2: sub R5, **R1**, R4

There is a RAW hazard because subtraction instruction reads output of the addition. The hazard for instructions 'add **R1,** R2, R3' and 'sub R5, **R1,** R4' is described as follows:

Instructions / Cycle	**1**	**2**	**3**	**4**	**5**	**6**
I_1	IF	ID	EX	MEM	WB	
I_2		IF	ID	EX	MEM	WB

The RAW hazard is very common.

WAR

WAR can be referred to as 'Write after Read'. It is also known as Anti-Data dependency. If the later instruction tries to write an operand before the earlier instruction reads it, in this case, the WAR hazards will occur. The condition to detect the WAR hazard is when I_n and O_{n+1} both have a minimum one common operand.

For example:

The dependency is described as follows:

add R1, **R2,** R3

sub **R2,** R5, R4

Here addition instruction creates a WAR hazard because subtraction instruction writes R2, which is read by addition. In a reasonable (in-order) pipeline, the WAR hazard is very uncommon or impossible. The hazard for instructions 'add R1, **R2,** R3' and 'sub **R2,** R5, R4' are described as follows:

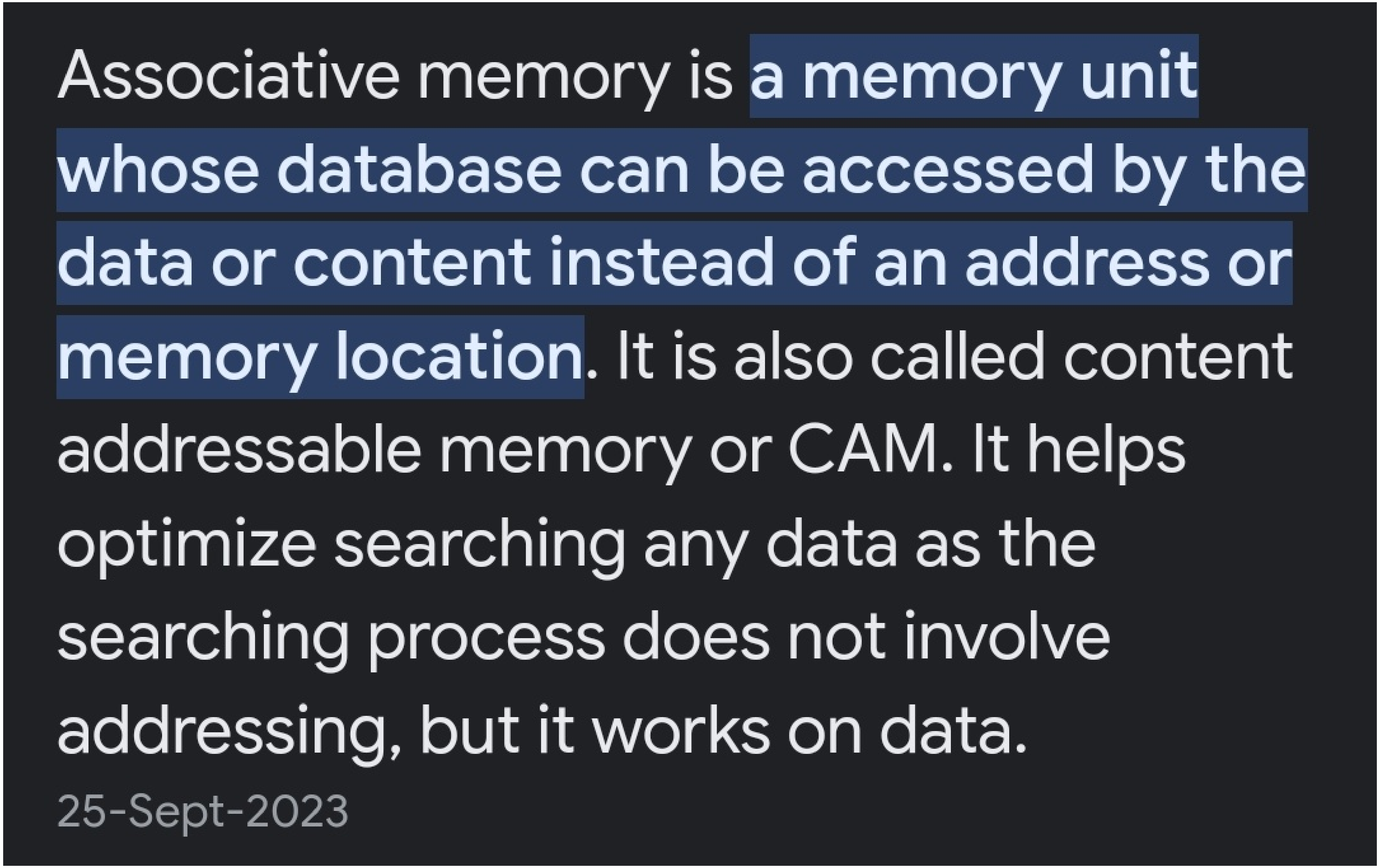
Data Hazards

Data hazards occur when instructions that exhibit data dependence, modify data in different stages of a pipeline. Hazard cause delays in the pipeline. There are mainly three types of data hazards:

1) RAW (Read after Writing) [Flow dependency]

2) WAR (Write after reading) [Anti-Data dependency]

3) WAW (Write after Writing) [Output dependency]



Associative memory is **a memory unit whose database can be accessed by the data or content instead of an address or memory location**. It is also called content addressable memory or CAM. It helps optimize searching any data as the searching process does not involve addressing, but it works on data.

25-Sept-2023

Memory Interleaving is less or More an Abstraction technique. Though it’s a bit different from Abstraction. It is a Technique that divides memory into a number of modules such that Successive words in the address space are placed in the Different modules.

to be generated are hard wired	done only at the instruction level
More costlier as everything has to be realized in terms of logic gates	Less costlier than hardwired control as only micro instructions are used for generating control signals
It cannot handle complex instructions as the circuit design for it becomes complex	It can handle complex instructions
Only limited number of instructions are used due to the hardware implementation	Control signals for many instructions can be generated
Used in computer that makes use of Reduced Instruction Set Computers(RISC)	Used in computer that makes use of Complex Instruction Set Computers(CISC)

Hardwired Control Unit	Microprogrammed Control Unit
Hardwired control unit generates the control signals needed for the processor using logic circuits	Microprogrammed control unit generates the control signals with the help of micro instructions stored in control memory
Hardwired control unit is faster when compared to microprogrammed control unit as the required control signals are generated with the help of hardwares	This is slower than the other as micro instructions are used for generating signals here
Difficult to modify as the control signals that need to be generated are hard wired	Easy to modify as the modification need to be done only at the instruction level
More costlier as everything has to be realized in terms of logic gates	Less costlier than hardwired control as only micro instructions are used for generating control signals
It cannot handle complex	

State table method

- Here the behavior of control unit is represented in the form of a table, which is known as the **state table**.
- Here, each row represents the T-states and the columns represent the instructions.
- Every intersection of the specific column to each row indicates which control signal will be produced in the corresponding T- state of an instruction.
- Here the hardware circuitry is designed for each column(i.e. for a specific instruction) for producing control signals in different T-states.

Advantage –

- It is the simplest method.
- This method is mainly used for small instruction set processors(i.e. in RISC processors).