BUS Arbitration in Computer Organization

S SUDIPTADANDAPAT

Read Discuss Courses

Introduction :

In a computer system, multiple devices, such as the CPU, memory, and I/O controllers, are connected to a common communication pathway, known as a bus. In order to transfer data between these devices, they need to have access to the bus. Bus arbitration is the process of resolving conflicts that arise when multiple devices attempt to access the bus at the same time.

When multiple devices try to use the bus simultaneously, it can lead to data corruption and system instability. To prevent this, a bus arbitration mechanism is used to ensure that only one device has access to the bus at any given time.

There are several types of bus arbitration methods, including centralized, decentralized, and distributed arbitration. In centralized arbitration, a single device, known as the bus controller, is responsible for managing access to the bus. In decentralized arbitration, each device has its own priority level, and the device with the highest priority is given access to the bus. In distributed arbitration, devices compete for access to the bus by sending a request signal and waiting for a grant signal.

Bus Arbitration refers to the process by which the current bus master accesses and then controller that has access to a bus at an instance is known as a **Bus master**.

A conflict may arise if the number of DMA controllers or other controllers or processors try to access the common bus at the same time, but access can be given to only one of those. Only one processor or controller can be Bus master at the same point in time. To resolve these conflicts, the Bus Arbitration procedure is implemented to coordinate the activities of all devices requesting memory transfers. The selection of the bus master must take into account the needs of various devices by establishing a priority system for gaining access to the bus. The **Bus Arbiter** decides who would become the current bus master.

Applications of bus arbitration in computer organization:

Shared Memory Systems: In shared memory systems, multiple devices need to access the memory to read or write data. Bus arbitration allows multiple devices to access the memory without interfering with each other.

Multi-Processor Systems: In multi-processor systems, multiple processors need to communicate with each other to share data and coordinate processing. Bus arbitration allows multiple processors to share access to the bus to communicate with each other and with shared memory.

Input/Output Devices: Input/Output devices such as keyboards, mice, and printers need to communicate with the processor to exchange data. Bus arbitration allows multiple input/output devices to share access to the bus to communicate with the processor and memory.

Real-time Systems: In real-time systems, data needs to be transferred between devices and memory within a specific time frame to ensure timely processing. Bus arbitration can help to ensure that data transfer occurs within a specific time frame by managing access to the bus.

Embedded Systems: In embedded systems, multiple devices such as sensors, actuators, and controllers need to communicate with the processor to control and monitor the system. Bus arbitration allows multiple devices to share access to the bus to communicate with the processor and memory.

There are two approaches to bus arbitration:

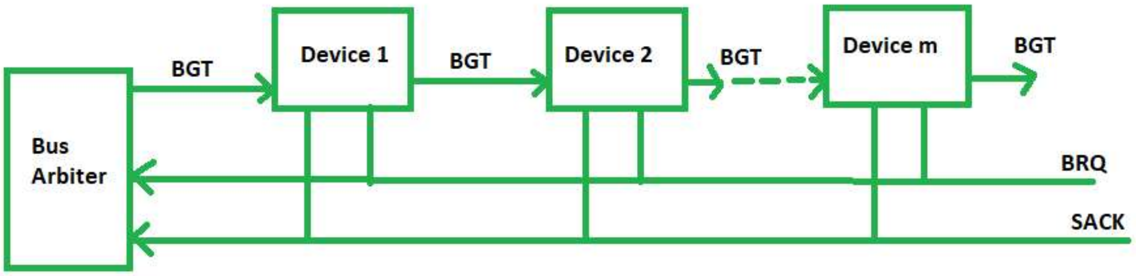
1. **Centralized bus arbitration –**
   A single bus arbiter performs the required arbitration.

2. **Distributed bus arbitration** –
   All devices participating in the selection of the next bus master.

Methods of Centralized BUS Arbitration:

There are three bus arbitration methods:

(i) Daisy Chaining method: It is a simple and cheaper method where all the bus masters use the same line for making bus requests. The bus grant signal serially propagates through each master until it encounters the first one that is requesting access to the bus. This master blocks the propagation of the bus grant signal, therefore any other requesting module will not receive the grant signal and hence cannot access the bus. During any bus cycle, the bus master may be any device – the processor or any DMA controller unit, connected to the bus.



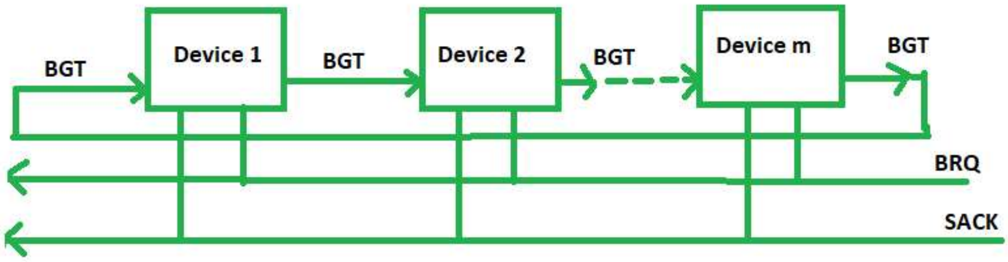
Daisy chained bus arbitration

Advantages:

- Simplicity and Scalability.
- The user can add more devices anywhere along the chain, up to a certain maximum value.

- The value of priority assigned to a device depends on the position of the master bus.
- Propagation delay arises in this method.
- If one device fails then the entire system will stop working.

(ii) Polling or Rotating Priority method: In this, the controller is used to generate the address for the master(unique priority), the number of address lines required depends on the number of masters connected in the system. The controller generates a sequence of master addresses. When the requesting master recognizes its address, it activates the busy line and begins to use the bus.



Rotating priority bus arbitration

Advantages –

- This method does not favor any particular device and processor.
- The method is also quite simple.

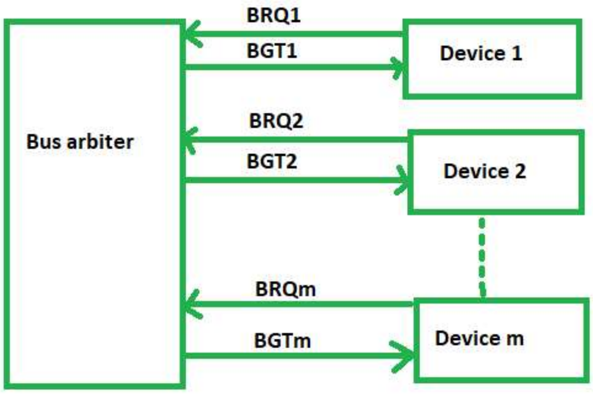
Disadvantages –

- Adding bus masters is difficult as increases the number of address lines of the circuit.
- If one device fails then the entire system will not stop working.

(iii) Fixed priority or Independent Request method –

In this, each master has a separate pair of bus request and bus grant lines and each pair has a priority assigned to it.

The built-in priority decoder within the controller selects the highest priority request and asserts the corresponding bus grant signal.



Fixed priority bus arbitration method

Advantages –

- This method generates a fast response.

Disadvantages –

- Hardware cost is high as a large no. of control lines is required.

Distributed BUS Arbitration :

In this, all devices participate in the selection of the next bus master. Each device on the bus is assigned a 4bit identification number. The priority of the device will be determined by the generated ID.

Uses of BUS Arbitration in Computer Organization :

Bus arbitration is a critical process in computer organization that has several uses and benefits, including:

1. Efficient use of system resources: By regulating access to the bus, bus arbitration ensures that each device has fair access to system resources, preventing any single device from monopolizing the bus and causing system slowdowns or crashes.
2. Minimizing data corruption: Bus arbitration helps prevent data corruption by ensuring that only one device has access to the bus at a time, which minimizes the risk of multiple devices writing to the same location in memory simultaneously.
3. Support for multiple devices: Bus arbitration enables multiple devices to share a common communication pathway, which is essential for modern computer systems with multiple peripherals, such as printers, scanners, and external storage devices.
4. Real-time system support: In real-time systems, bus arbitration is essential to ensure

the bus, bus arbitration can ensure that critical tasks are given the resources they need to execute in a timely manner.

5. Improved system stability: By preventing conflicts between devices, bus arbitration helps to improve system stability and reliability. This is especially important in mission-critical systems where downtime or data corruption could have severe consequences.

Issues of BUS Arbitration in Computer Organization :

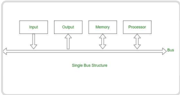
Bus arbitration is a critical process in computer organization that has several uses and benefits, including:

1. Efficient use of system resources: By regulating access to the bus, bus arbitration ensures that each device has fair access to system resources, preventing any single device from monopolizing the bus and causing system slowdowns or crashes.
2. Minimizing data corruption: Bus arbitration helps prevent data corruption by ensuring that only one device has access to the bus at a time, which minimizes the risk of multiple devices writing to the same location in memory simultaneously.
3. Support for multiple devices: Bus arbitration enables multiple devices to share a common communication pathway, which is essential for modern computer systems with multiple peripherals, such as printers, scanners, and external storage devices.
4. Real-time system support: In real-time systems, bus arbitration is essential to ensure that high-priority tasks are executed quickly and efficiently. By prioritizing access to the bus, bus arbitration can ensure that critical tasks are given the resources they need to execute in a timely manner.
5. Improved system stability: By preventing conflicts between devices, bus arbitration helps to improve system stability and reliability. This is especially important in mission-critical systems where downtime or data corruption could have severe consequences.

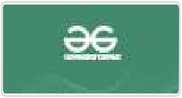
Last Updated : 06 May, 2023

59

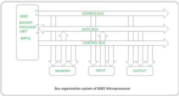
Similar Reads



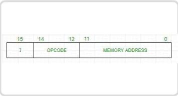
Difference between Single Bus Structure and Double Bus Structure



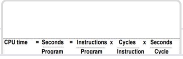
Difference Between System Bus and Address Bus



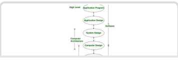
Bus organization of 8085 microprocessor



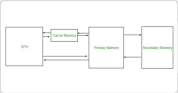
Computer Organization | Basic Computer Instructions



Computer Organization | Performance of Computer



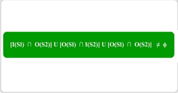
Differences between Computer Architecture and Computer



Cache Memory in Computer Organization

Computer Organization and Architecture | Pipelining | Set 1 (Execution, Stages and…

Computer Organization and Architecture | Pipelining | Set 3 (Types and Stalling)



Computer Organization and Architecture | Pipelining | Set 2 (Dependencies and Data Hazard)

Previous

Next

Article Contributed By :

S

SUDIPTADANDAPAT

SUDIPTADANDAPAT

Vote for difficulty

Current difficulty : Medium

Easy Normal Medium Hard Expert

Improved By : Stark17, shubhagrawal1280, yatindracr7, siddharthsingh0705, vicky978654, kreshanth_sv, nikhilbhoi9739, rushi5758

Article Tags : Computer Organization & Architecture

Improve Article

Report Issue

A-143, 9th Floor, Sovereign Corporate
Tower, Sector-136, Noida, Uttar Pradesh -
201305

feedback@geeksforgeeks.org

GET IT ON Google Play

Download on the App Store

Company

About Us

Legal

Careers

In Media

Contact Us

Advertise with us

Explore

Job-A-Thon Hiring Challenge

Hack-A-Thon

GfG Weekly Contest

Offline Classes (Delhi/NCR)

DSA in JAVA/C++

Master System Design

Master CP

Languages

Python

Java

C++

PHP

GoLang

SQL

R Language

Android Tutorial

DSA Concepts

Data Structures

Arrays

Strings

Linked List

Algorithms

Searching

Sorting

Mathematical

Dynamic Programming

DSA Roadmaps

DSA for Beginners

Basic DSA Coding Problems

Complete Roadmap To Learn DSA

DSA for FrontEnd Developers

DSA with JavaScript

Web Development

HTML

CSS

JavaScript

Bootstrap

ReactJS

DSA Roadmap by Sandeep Jain

Computer Science

GATE CS Notes

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning Tutorial

Maths For Machine Learning

Pandas Tutorial

NumPy Tutorial

NLP Tutorial

Deep Learning Tutorial

Competitive Programming

Top DSA for CP

Top 50 Tree Problems

Top 50 Graph Problems

Top 50 Array Problems

Top 50 String Problems

Top 50 DP Problems

Top 15 Websites for CP

Interview Corner

Company Wise Preparation

Preparation for SDE

Experienced Interviews

Internship Interviews

Express.js

Lodash

Python

Python Programming Examples

Django Tutorial

Python Projects

Python Tkinter

OpenCV Python Tutorial

Python Interview Question

DevOps

Git

AWS

Docker

Kubernetes

Azure

GCP

System Design

What is System Design

Monolithic and Distributed SD

Scalability in SD

Databases in SD

High Level Design or HLD

Low Level Design or LLD

Top SD Interview Questions

GfG School

CBSE Notes for Class 8

CBSE Notes for Class 9

CBSE Notes for Class 10

CBSE Notes for Class 11

Aptitude Preparation

Commerce

Accountancy

Business Studies

Economics

Management

Income Tax

Finance

Statistics for Economics

SSC/ BANKING

SSC CGL Syllabus

SBI PO Syllabus

SBI Clerk Syllabus

IBPS PO Syllabus

IBPS Clerk Syllabus

Aptitude Questions

SSC CGL Practice Papers

English Grammar

UPSC

Polity Notes

Geography Notes

History Notes

Science and Technology Notes

Economics Notes

Important Topics in Ethics

UPSC Previous Year Papers

Write & Earn

Write an Article

Improve an Article

Pick Topics to Write

Write Interview Experience

Internships