Here's a **moderate-difficulty (6/10) activity** designed to help students understand and compare **Bus-Based, Crossbar, and Omega Interconnection Networks** in parallel computer architecture.

**🔍 Activity: Explore and Compare Interconnection Networks**

**Topic:** Parallel Architecture  
**Focus:** Bus-Based, Crossbar, and Omega Networks  
**Difficulty Level:** 6/10  
**Target Audience:** Undergraduate students in Computer Architecture or Parallel Computing

**🧠 Learning Objectives**

By completing this activity, students will be able to:

* Explain the structure and function of three interconnection networks.
* Evaluate trade-offs in terms of **cost**, **performance**, and **scalability**.
* Analyze communication paths for data transfer between processors and memory modules.

**📘 Activity Structure**

**Part 1: Quick Theory Recap (10 mins)**

Provide the following definitions (can be done through reading or short lecture):

* **Bus-Based Network**: A single shared communication line; all components communicate via this line.
* **Crossbar Network**: Every processor is connected to every memory module through a dedicated path.
* **Omega Network**: Multistage interconnection using 2×2 switches in log₂(N) stages.

**Part 2: Network Mapping Exercise (15 mins)**

**Materials Needed:** Pen & paper or digital drawing tool (e.g., Lucidchart, draw.io)

**Task:**  
Draw network diagrams for 4 processors and 4 memory modules:

* Bus-Based Network
* Crossbar Network
* Omega Network

✅ *Tip for instructor:* Provide a blank table with processor labels P0-P3 and memory M0-M3.

**Part 3: Routing Simulation (20 mins)**

**Task:** Use the drawn networks to simulate data transfer between processor and memory pairs.

For each network, simulate the following transfers:

* P0 → M2
* P1 → M3
* P2 → M1
* P3 → M0

**Questions:**

1. Can these transfers happen in parallel? If not, explain why.
2. Which network completes all transfers the fastest?
3. In which network is a **collision/conflict** most likely?

💡 *Optional Challenge:* Identify blocking and non-blocking network properties.

**Part 4: Cost & Performance Analysis (10 mins)**

Fill in this comparison table:

| **Feature** | **Bus-Based** | **Crossbar** | **Omega** |
| --- | --- | --- | --- |
| Switch Count (for 4x4) | ? | ? | ? |
| Parallelism | ? | ? | ? |
| Scalability | ? | ? | ? |
| Cost | Low/High | ? | ? |
| Blocking? | Yes/No | ? | ? |

🔎 Use this prompt to encourage critical thinking:  
*"How would each network scale to 8 processors and 8 memory modules? What design trade-offs emerge?"*

**✅ Submission**

Students submit:

* Their drawn diagrams
* Routing simulation outcomes
* Completed comparison table

**🧩 Bonus Activity (Optional)**

Implement the routing logic of a simple Omega network in code (Python or pseudocode), showing how addresses are routed through stages using bit-switch logic.

Would you like a printable PDF worksheet or interactive slide version of this activity?

Ask ChatGPT

**You said:**

answers

**ChatGPT said:**

Here are the **answers** for the activity sections based on **4 processors (P0–P3) and 4 memory modules (M0–M3)**:

**✅ Answers: Activity on Interconnection Networks**

**🔸 Part 2: Network Diagrams**

**1. Bus-Based Network:**

* A single bus connects all processors and memory modules.
* Only one communication can happen at a time.

**2. Crossbar Network:**

* A 4×4 grid of switches.
* Each processor has a direct line (via switch) to every memory module.
* All processors can potentially communicate simultaneously.

**3. Omega Network (4x4):**

* Consists of **log₂(4) = 2 stages**, each with **2×2 switches**.
* Total: **4 switches per stage × 2 stages = 8 switches**.
* Bit-level routing: input address bits used to determine the path.

**🔸 Part 3: Routing Simulation**

| **Transfer** | **Bus-Based** | **Crossbar** | **Omega** |
| --- | --- | --- | --- |
| P0 → M2 | ✔️ if bus is free | ✔️ Direct path | ✔️ Routed via stages |
| P1 → M3 | ❌ if P0 active | ✔️ Direct path | ❌ if P0 active (conflict) |
| P2 → M1 | ❌ if bus busy | ✔️ Direct path | ❌ if routing overlaps |
| P3 → M0 | ❌ if bus busy | ✔️ Direct path | ❌ if routing overlaps |

**🔍 Answers to Questions:**

1. **Can transfers happen in parallel?**
   * **Bus:** ❌ No – shared medium, only one at a time.
   * **Crossbar:** ✔️ Yes – all can happen simultaneously.
   * **Omega:** ❌ Not always – may have **blocking** due to shared switch paths.
2. **Which network completes all transfers fastest?**
   * **Crossbar** – supports **full parallel communication** with no blocking.
3. **Where is conflict most likely?**
   * **Bus-Based:** Shared bus = collision if multiple requests.
   * **Omega:** Switch-level conflict if paths overlap.
   * **Crossbar:** ❌ Least likely – all processors have independent paths.

**🔸 Part 4: Comparison Table**

| **Feature** | **Bus-Based** | **Crossbar** | **Omega** |
| --- | --- | --- | --- |
| Switch Count (4×4) | 0 (just bus) | 16 (4×4 matrix) | 8 (2×4 per stage × 2) |
| Parallelism | Very Low | High (non-blocking) | Moderate (blocking) |
| Scalability | Poor | Poor (cost↑ fast) | Good (log₂N stages) |
| Cost | Low | Very High | Moderate |
| Blocking? | Yes | No (non-blocking) | Yes (blocking) |

**🧠 Bonus Concept Recap**

* **Bus-Based**: Easy, cheap, but only one communication at a time.
* **Crossbar**: Best performance but very expensive and non-scalable.
* **Omega**: Balanced trade-off – cost-effective and scalable but blocking can occur.

Would you like this in PDF or PowerPoint format for teaching or classroom distribution?

Top of Form



Attach

Search

Voice

Bottom of Form