1. DRAM Bank Conflicts

**Bank conflict!**

- Large latency

2. Timeline of DRAM Bank Conflicts

- Four requests to the **same** DRAM bank
  - WR, WR, RD, RD
  - Large latency due to 3 problems:
    1. Serialization of requests
    2. Write penalty after WR request
    3. "Thrashing" of row-buffer

- WR, RD → time (Short latency when four reqs are to different banks)
- WR, RD → time

3. Our Goal

- **Goal:** Cost-effectively mitigate the detrimental effects of bank conflicts
- **Naïve Solution:** Simply add more banks
  - Very expensive

4. Two Key Observations

1. A DRAM bank is divided into **subarrays**
   - Each subarray has a **local row-buffer**

2. **Subarrays are mostly independent**...
   - Except when sharing **global structures**

5. Key Idea

Reduce the sharing of...

1. **Global decoder:** enable parallel access to multiple subarrays
2. **Global row-buffer:** utilize multiple local row-buffers concurrently

6. Mechanism: **MASA**

*Multitude of Activated Subarrays*

*Add two latches to each subarray*

1. **Subarray Address Latch**
   - Stored per-subarray row-address
2. **Designated-Bit Latch**
   - Connects one subarray’s local row-buffer to global row-buffer

---

**IPC Increase**

- **Baseline** vs **MASA**
  - Baseline: 1
  - MASA: +13%

**Row-Buffer Hit-Rate**

- Baseline: 90%
  - MASA: +13%

**Die-Size**

- **Baseline** vs **MASA**
  - Die-Size < 0.15%
  - MASA: 0.15% 36.3%

**Normalized Dynamic Energy**

- Baseline: 1
  - MASA: -19%