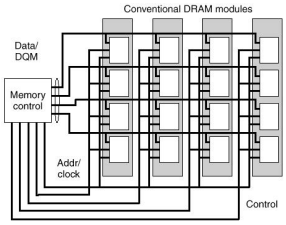
# Memory Hierarchy and Cache Design (3)

## Main Memory Background

- Conventional DRAM system




## Main Memory Background

- Performance Metrics of Main Memory:
  - Latency: Affects cache miss penalty
    - » *Access Time*: time between the request and when the desired word arrives
    - » *Cycle Time*: minimum time between requests
  - Bandwidth: Affects I/O performance & cache miss penalty (especially when a large block is used in the L2 cache)

## Main Memory Background

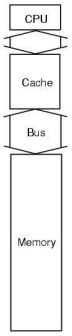
- Main Memory uses *DRAM* (dynamic RAM)
  - Dynamic because of the need to be refreshed periodically (but requires only 1 transistor/bit)
  - Addresses are divided into 2 parts:
    - » *RAS* or *Row Access Strobe*
    - » *CAS* or *Column Access Strobe*
- Cache uses *SRAM* (static RAM)
  - No refresh (but requires 6 transistors/bit)
  - Address not divided for fast access
- Comparison
  - Capacity: DRAM is 4-8 times that of SRAM
  - Cycle time: SRAM is 8-16 times faster than DRAM
  - Cost: SRAM is 8-16 times more expensive than DRAM

## Trends in DRAM

| Year of introduction | Chip size | Row access strobe (RAS) Slowest DRAM | Row access strobe (RAS) Fastest DRAM | Column access strobe (CAS) | Cycle time |
|---|---|---|---|---|---|
| 1980 | 64 Kbit | 180 ns | 150 ns | 75 ns | 250 ns |
| 1983 | 256 Kbit | 150 ns | 120 ns | 50 ns | 220 ns |
| 1986 | 1 Mbit | 120 ns | 100 ns | 25 ns | 190 ns |
| 1989 | 4 Mbit | 100 ns | 80 ns | 20 ns | 165 ns |
| 1992 | 16 Mbit | 80 ns | 60 ns | 15 ns | 120 ns |
| 1995 | 64 Mbit | 65 ns | 50 ns | 10 ns | 90 ns |

Capacity improves by 60% per year
Row access time improves by 7% per year

## Main Memory Organizations

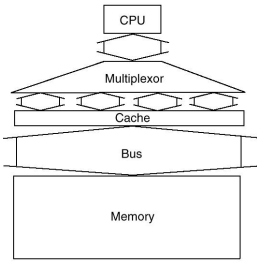



- **Basic Memory Organization**
  - **one-word wide bus**
  - **4 clock cycles to send the address**
  - **24 clock cycles for the access time per word**
  - **4 clock cycles to send a word of data**
- **Example**
  - **cache block of 4 words**
  - **miss penalty = 4 x (4 + 24 + 4) = 128 cycles**

## Faster Memory System

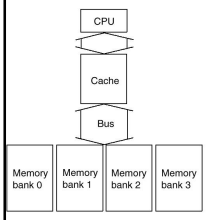
1. **Wider Main Memory**
2. **Simple Interleaved Memory**
3. **Independent Memory Banks**
4. **Avoiding Memory Bank Conflicts**
5. **DRAM-specific Interleaving**

## First Technique: Wider Main Memory



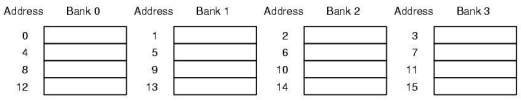

- **Cache miss penalty**
  - **two-word wide bus**
    - » **2 x (4 + 24 + 4) = 64 cycles**
  - **four-word wide bus**
    - » **1 x (4 + 24 + 4) = 32 cycles**
- **Drawbacks**
  - **Higher bus costs**
  - **Multiplexor**
  - **Reduced expandability**
  - **More frequent "read-modify-write"s in memories with error correction**

## Second Technique: Simple Interleaved Memory




- Memory consists of several DRAM Chips
  - Each chip is capable of autonomous operation
- Organize memory chips in banks and issue memory requests to all banks at the same time
- Banks are one word wide

## Four way interleaving




## Memory Interleaving

- Mapping addresses to banks affects the behavior of the memory system
  - Optimized for sequential access
- May spread consecutive addresses to several banks
  - Interleaving Factor
  - Normally word interleaved
    - » Can also be byte interleaved
    - » Depends on the organization of the bank
- Goal: deliver information from new bank on each cycle
  - Need more banks than the number of cycles to access a bank
- As memory chip size increases - use fewer chips
  - Constructing multiple banks becomes difficult
- Restricted Expandability
  - Can increase memory only by doubling it
- Cache Miss Penalty
  - 4 + 24 + 4 x 4 = 44 cycles

## Third Technique: Independent Memory Banks

- Multiple independent banks
  - Multiple memory controllers
  - Each bank uses separate address and data lines

## Fourth Technique: Avoiding Memory Bank Conflicts

- **Problem**

```
int x[256][512];
     for (j = 0; j < 512; j = j+1)
           for (i = 0; i < 256; i = i+1)
                 x[i][j] = 2 * x[i][j];
```

  - Even with 128 banks there are conflicts, since 512 is a multiple of 128
- **Software solutions**
  - loop interchange
  - Resizing the array
- **Hardware solutions**
  - Based on the Chinese Remainder Theorem
  - Prime number of banks
  - bank number = address mod number of banks
  - address within bank = address mod number of words in bank

## Avoiding Memory Bank Conflicts

- **Example**

| Address within bank | Memory bank | | | | | |
|---|---|---|---|---|---|---|
| | Sequentially interleaved | | | Modulo interleaved | | |
| | 0 | 1 | 2 | 0 | 1 | 2 |
| 0 | 0 | 1 | 2 | 0 | 16 | 8 |
| 1 | 3 | 4 | 5 | 9 | 1 | 17 |
| 2 | 6 | 7 | 8 | 18 | 10 | 2 |
| 3 | 9 | 10 | 11 | 3 | 19 | 11 |
| 4 | 12 | 13 | 14 | 12 | 4 | 20 |
| 5 | 15 | 16 | 17 | 21 | 13 | 5 |
| 6 | 18 | 19 | 20 | 6 | 22 | 14 |
| 7 | 21 | 22 | 23 | 15 | 7 | 23 |

## Fifth Technique: DRAM-Specific Interleaving

- **Multiple column accesses: page mode**
  - DRAM buffers a row of bits inside the DRAM for column access (e.g., 16 Kbits row for 64 Mbits DRAM)
  - Allow repeated column access without another row access
  - 64 Mbit DRAM: cycle time = 90 ns, optimized access = 25 ns
- **New DRAMs**
  - Example: RAMBUS
    - » each chip acts like a memory system
    - » uses a packet-switched bus
    - » is synchronous to the CPU clock
    - » returns a variable amount of data
    - » even performs its own refresh

## Summary of Faster Memory Systems

1. Wider Main Memory
2. Simple Interleaved Memory
3. Independent Memory Banks
4. Avoiding Memory Bank Conflicts
5. DRAM-specific Interleaving