TED ÜNİVERSİTESİ

CMPE361 Computer Organization

Department of Computer Engineering
TED University- Fall 2023

Memory Systems 1

These Slides are mainly based on slides of the text book (downloadable from the book’s website).

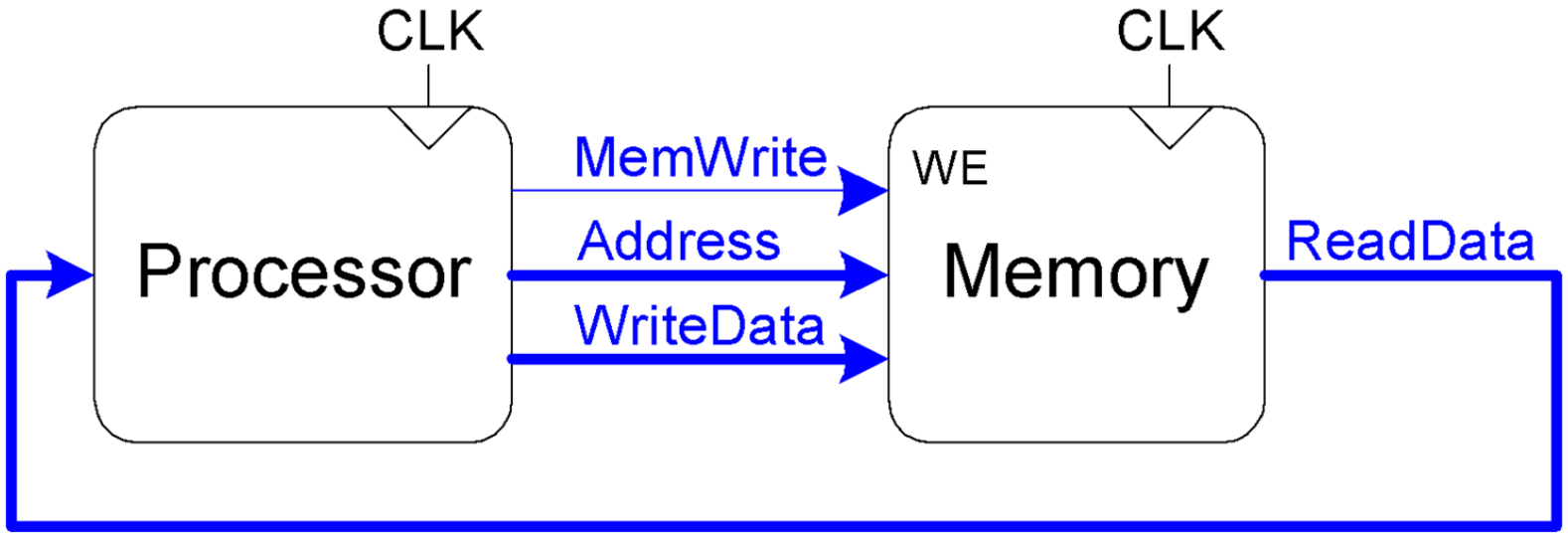
Memory

- Computer system performance depends on the memory system as well as the processor microarchitecture.
- So far, an ideal memory system for a single clock cycle is assumed.
- This is very small memory for a very slow processor!
- However, processor speed has increased at a faster rate than memory speeds. DRAM memories are currently 10 to 100 times slower than processors.
- Demand for increasing DRAM memory speeds continues.

Introduction

- Computer performance depends on:
  - Processor performance
  - Memory system performance

Memory Interface



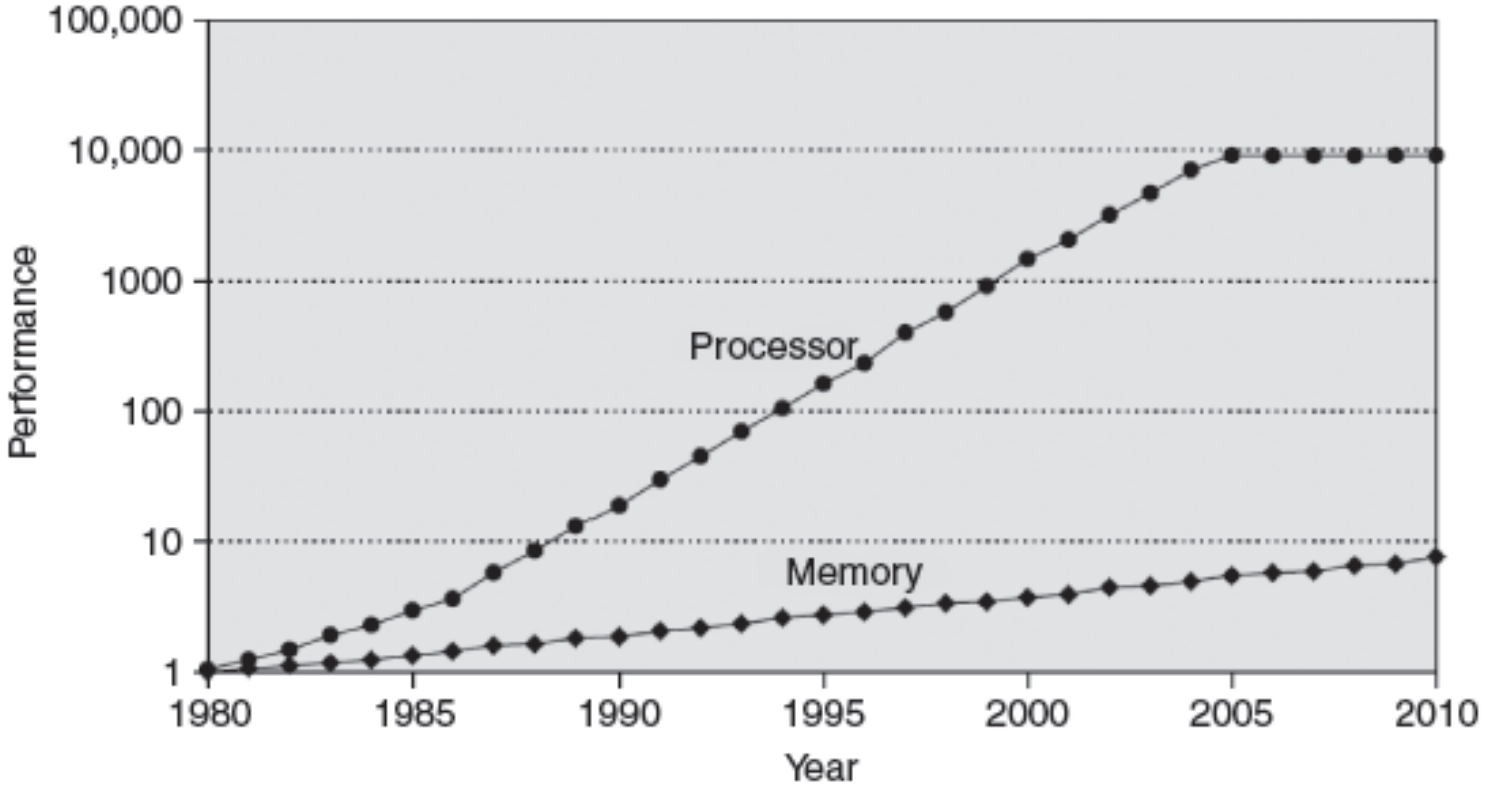
Memory processor interface

- The processor sends an address over the Address bus to the memory system.
- For a read, MemWrite is 0 and the memory returns the data on the ReadData bus.
- For a write, MemWrite is 1 and the processor sends data to memory on the WriteData bus.
- Memory usage principle: read the text on “a metaphor of books in a library” to understand the memory usage principles…

Note for instructor: Explain the principle of locality through library example.

Processor-Memory Gap

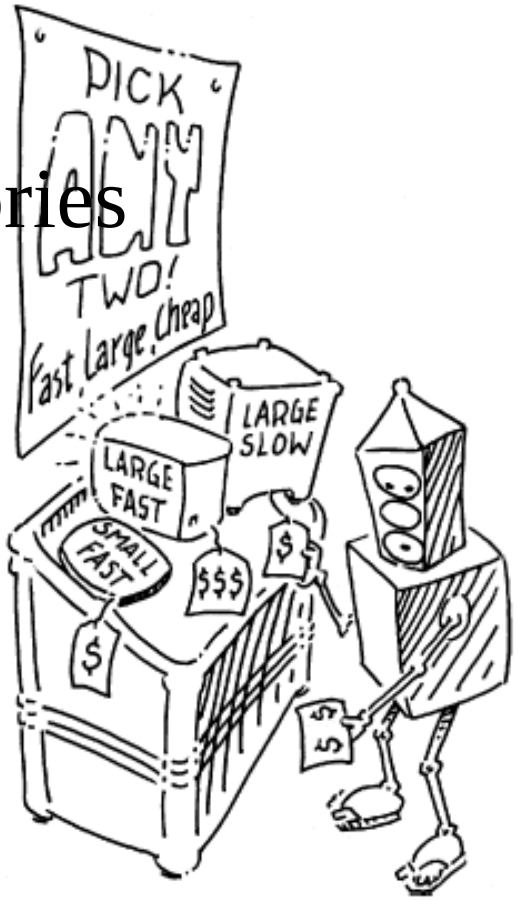
In about 1980, processor and memory speeds were the same. They have diverged since then, with memories badly falling behind.



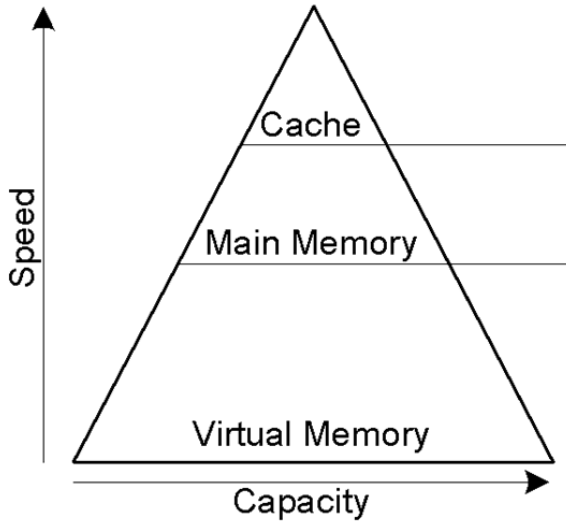
Memory System Challenge

- Make memory system appear as fast as processor as far as possible!
- How?
  - Make use of hierarchy of memories
- Ideal memory:
  - Fast
  - Cheap (inexpensive)
  - Large (capacity)

But can only choose two!



Memory Hierarchy



Technology	Price / GB	Access Time (ns)	Bandwidth (GB/s)
SRAM	$10,000	1	25+
DRAM	$10	10 - 50	10
SSD	$1	100,000	0.5
HDD	$0.1	10,000,000	0.1

memory hierarchy with typical costs, access times, and bandwidth in 2012 technology. (Figure 8.3)

Using Locality for performance

Exploit locality to make memory accesses fast

- **Temporal Locality:** Locality in time
  - If data used recently, likely to use it again soon
  - **How to make use of it:** keep recently accessed data in higher levels of memory hierarchy
- **Spatial Locality:** Locality in space
  - If data used recently, likely to use data soon
  - **How to make use of it:** when access data, bring into higher levels of memory hierarchy

Memory hierarchy performance

- **Hit at a memory level:** data found in that level of memory hierarchy
- **Miss at a memory level:** data not found (must go to next level)

Hit Rate = # hits / # memory accesses
= 1 – Miss Rate

Miss Rate = # misses / # memory accesses
= 1 – Hit Rate

- **Average memory access time (AMAT):** average time for processor to access data

$$\textbf{AMAT} = t_{\text{cache}} + MR_{\text{cache}}[t_{MM} + MR_{MM}(t_{VM})]$$

=hitRate for cache x Access time of cache

=hitRate for memory x Access time of memory

=hitRate for storage x Access time of storage

Memory Performance Example 1

- A program has 2,000 loads and stores 1,250 of these data values in cache
- Rest supplied by other levels of memory hierarchy
- **What are the hit and miss rates for the cache?**

Memory Performance Example 1

- A program has 2,000 loads and stores 1,250 of these data values in cache
- Rest supplied by other levels of memory hierarchy
- **What are the hit and miss rates for the cache?**

Hit Rate = 1250/2000 **= 0.625**

Miss Rate = 750/2000 = **0.375** = 1 – Hit Rate

Memory Performance Example 2

- Suppose processor has 2 levels of hierarchy: cache and main memory
- t_{cache} = 1 cycle, t_{MM} = 100 cycles
- **What is the AMAT of the program from Example 1?**

Memory Performance Example 2

- Suppose processor has 2 levels of hierarchy: cache and main memory
- $t_{\text{cache}} = 1$ cycle, $t_{MM} = 100$ cycles
- **What is the AMAT of the program from Example 1?**

$$\begin{aligned} \textbf{AMAT} &= t_{\text{cache}} + MR_{\text{cache}}(t_{MM}) \\ &= [1 + 0.375(100)] \text{ cycles} \\ &= \textbf{38.5 cycles} \end{aligned}$$

Gene Amdahl, 1922-

- **Amdahl's Law:** the effort spent increasing the performance of a subsystem is wasted unless the subsystem affects a large percentage of overall performance
- Co-founded 3 companies, including one called Amdahl Corporation in 1970