# 9. The Memory Hierarchy (3) Main Memory

Main memory is the name given to the level below the cache(s) in the memory hierarchy. There is a large variety of dimensions, but a smaller one in speed due to the fact that vendors use the same chips to build memory arrays. A main memory may have a few MBytes for a typical Personal Computer, tens to hundreds of MBytes for a workstation, hundreds of MBytes to GBytes for supercomputers. The capacity of main memory has continuously increased over the years, as prices have dramatically dropped. The main memory must satisfy the cache requests as quickly as possible (the main memory should have a low latency), and must provide sufficient bandwidth for I/O devices and for vector units (if it is the case).

The **access time**, is defined as the time between the moment the read command is issued and the moment the requested data is at outputs.
The **cycle time** is defined as the minimum time between successive accesses to memory. The cycle time is usually greater than the access time.

## 9.1 DRAM/SRAM

To access data in a memory chip with a capacity of NxM bits, one must provide a number of addresses equal to:

$\log_2 N$

N is the number of “words” each chip has; each “word” is M bits wide. As the technology improved, the packaging costs become a real concern, as the number of address lines got greater and greater.

**Example 9.1** ADDRESS LINES:

Which is the number of address lines needed for a 4 Mbit memory chip:
a) organized as 4Mx1;
b) organized as 1Mx4?

**Answer:**
a) 4 M = $2^{22}$ hence the number of address lines is
$\log_2 2^{22} = 22$

b) 1 M = $2^{20}$therefore the number of address lines is
$\log_2 2^{20} = 20$

To keep memory chips cheap, the solution adopted for Dynamic RAM (DRAM) integrated circuits was to multiplex the address, thus reducing the number of pins for addresses to half, and adding two new control lines: **RAS** (Row Address Strobe), which loads into an internal buffer half of the address supplied by the control on the address lines, and **CAS** (Column Address Strobe) which handles the second half of the address.

**Example 9.2** NUMBER OF PINS:

How many pins has a 1Mx1 memory chip:
a) in DRAM technology;
b) in SRAM technology?

**Answer:**
Organization is important because it says how many pins are needed for data input and data output lines; it still does not say everything about the chip, more precisely it does not say if the input and output lines are the same or are separate.

Let's suppose that the chip in this example has one input line and one output line.

a) 1 M = $2^{20}$
$n_A = \log_2 2^{20} = 20$ addresses

a)For DRAM the number of address lines (pins) is half of the address size:

| | | |
|---|---|---|
| 10 | address lines | |
| 1 | RAS | |
| 1 | CAS | |
| 1 | WE | (Write Enable) |
| 1 | Din | (the data input line) |
| 1 | Dout | (the data output line) |
| 2 | for power supply | |

Total 17 pins needed for 1Mx1 DRAM. A real circuit has 18 pins, with one pin unused but devoted to use as an address line in the 4Mx1 chips.

b)For SRAM the number of address lines is the same as the address size:

| | |
|---|---|
| 20 | address lines |
| 1 | WE |
| 1 | Din |
| 1 | Dout |
| 2 | for power supply |

Total 25 pins needed for a 1Mx1 SRAM which fits in a 26 pin chip.

The reason for which SRAM address lines are not multiplexed is **speed;** the package is however more expensive than the package for a DRAM with the same capacity.

Another problem the designer faces, when using DRAM circuits is the refresh: each row in a DRAM circuit has to be accessed within some time interval (say 2 milliseconds), to prevent data from getting lost. This is a consequence of the dynamic technology, where data is stored as electric charge in a small capacitor: due to unavoidable losses in the dielectric of the capacitor, the charge decreases in time; the purpose of refresh is to periodically restore the charge on the capacitors (a charged capacitor stores a 1) thus preserving data.

The refresh requirement implies that the memory will be sometimes unavailable, being busy to do refresh: usually it takes a memory cycle to do a refresh to one row in DRAM. The number of refresh cycles in the critical time interval is a circuit specification, and can be found in the circuit's data sheet. Multiplexed address also mean an involved timing for DRAM memories; the cycle time is always larger than the access time.

SRAM use more transistors per memory-bit to prevent the loss of data. The cycle time for a static memory is very close to the access time, and is in the range of nanoseconds to tens of nanoseconds.

With the today's technology the maximum capacity of DRAMs is 16 times larger than that of SRAMs; the cycle time for a SRAM is, on the other hand, 8 to 16 times shorter than that of a DRAM.

As we saw in chapter 8, the faster a CPU is, the higher the miss penalty is. DRAM capacity has increased, over the last decade, quadrupling every three years. Unfortunately this is not the case with the DRAM performance: the 64 Kbit circuit, introduced in 1980, had a cycle time of 250 ns, while the 4 Mbit circuit introduced in 1989 has a cycle time of 165 ns; the capacity is 64 times larger but the performance is only 51% better (figures for the access time are quite similar).

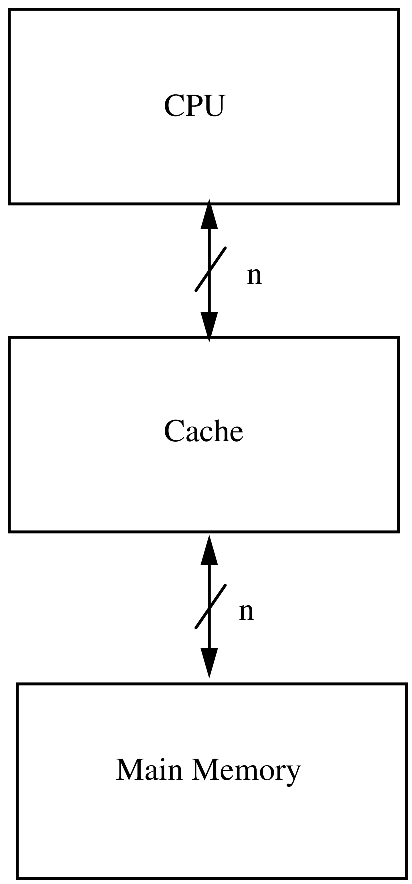


**FIGURE 9.1** Connecting the Main memory to the CPU and cache; all buses have the same width.

## 9.2 Possible Organizations for Main Memory

We shall compare different memory organizations based on the following assumptions:

- all transfers are multiple of word (1 word = 4 bytes);
- 1 clock cycle to send the address;
- 10 clock cycles for the access time;
- 1 clock cycle for a bus transfer of the accessed item.

The simple and cheap approach for memory organization is to have transfers, between all levels of the memory hierarchy, the same size, as depicted in Figure 9.1.

**Example 9.3** MEMORY ORGANIZATION:

Compute the miss penalty and the memory bandwidth for a word organized memory system. The cache block size is 8 words (32 bytes).

**Answer:**
For each word in the block the address must be transmitted (1 clock cycle), a fixed amount of time has to be spent waiting (10 clock cycles), and each word has to be transferred into the cache (1 clock cycle); therefore the miss penalty is:

miss_penalty = 8*(1 + 10 + 1) = 96 clock cycles

The memory bandwidth is:

$$\text{memory\_bandwidth} = \frac{\text{bytes\_transferred}}{\text{clock\_cycles}}$$

$$\text{memory\_bandwidth} = \frac{32}{96} = 0.33 \text{ bytes/clock cycle}$$

There are two parameters that can be modified to obtain a larger memory bandwidth: to increase the number of bytes transferred in the same amount of time, and to decrease the number of clock cycles necessary to complete a block transfer. The two possibilities correspond to two different memory organizations: a wider memory and interleaved memory respectively.

### Wider Main Memory

The basic organization of a wider memory is presented in Figure 9.2. The data bus between the cache and main memory is wider than the bus between the cache and the CPU (the size of this one is the size of the

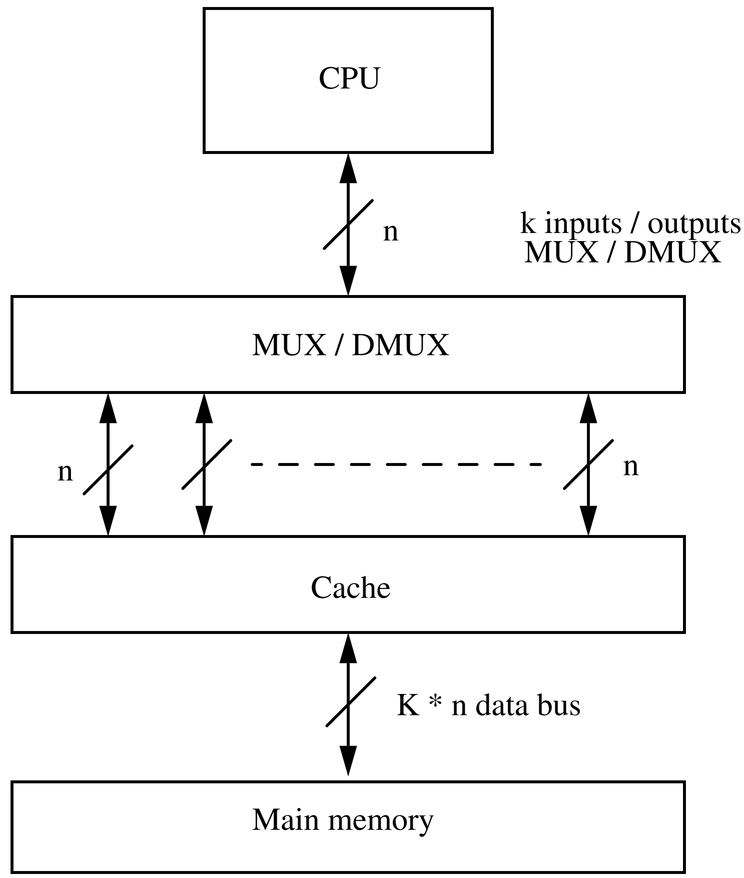


**FIGURE 9.2** Wide memory organization. k * n bits are transferred between main memory and the cache, but only n bits between CPU and the cache.

datapath). In this case the address no longer indicates a word in main memory but a block; it is, in other words, the block-frame address that comes to main memory.

---

**Example 9.4** MISS PENALTY AND MEMORY BANDWIDTH:

Calculate the miss penalty and the bus bandwidth for a wide memory organization. The cache line is 8 words wide, and the data bus is also 8 words wide.

**Answer:**
It takes one clock cycle to transmit the address, 10 clock cycles to access a line in the memory (8 words are accessed at once), and one more clock cycle to transfer the line into the cache; hence, the miss penalty is:

miss_penalty = 1 + 10 + 1 = 12 clock cycles

and the memory bandwidth is:

$$\text{memory\_bandwidth} = \frac{\text{bytes\_transferred}}{\text{clock\_cycles}}$$

$$\text{memory\_bandwidth} = \frac{32}{12} = 2.67 \text{ bytes/clock cycle}$$

The memory bandwidth is 8 times larger than for the word organized memory.

There is a price to be paid for better performance: because the CPU reads only n bits at a time and the cache is n*k bits organized, there must be a multiplexer between the cache and the CPU. Incidentally this is the case with cache represented in Figure 8.5 where we have a multiplexer that selects the proper word from the block (note also that, if the memory is byte addressable the scheme gets more involved, in that the type of item being addressed must be stated).

Another problem with the wide memories is related to the price paid by the customer: if the chips available for expansion have a NxM capacity, then the number of circuits the user must add into the system is a multiple of:

n*k / M

because the user has to add complete lines to the memory for expansion.

**Interleaved Memory**

The memory can be organized in banks, each bank one word wide, such that transfers between the cache and main memory are word wide, but several words can be read at once and then transferred one after the other to the cache. Figure 9.3 presents a interleaved memory organization.

Having an interleaved memory there is only one address that the CPU has to supply to main memory: this address will access a word in the bank number:

(address) modulo (number of banks)

while the other banks will access words at addresses successive to the address issued by the CPU.

**Example 9.5** MISS PENALTY AND MEMORY BANDWIDTH:

Compute the miss penalty and the memory bandwidth for a 4 bank main memory; each bank is one word wide. The cache is 8 words wide.

**Answer:**
Because the memory has 4 banks, there will be 4 words read in a single burst; this takes one clock cycle to send the address, 10 clock cycles waiting time for the memories to access date, and 4 clock cycles to read the four word coming from the four banks. Since the cache block is 8 words wide this process has to be repeated; therefore, the miss penalty is:

miss_penalty = 2*(1 + 10 + 4*1) = 30 clock cycles

and the memory bandwidth is:

$$\text{memory\_bandwidth} = \frac{\text{bytes\_transferred}}{\text{clock\_cycles}}$$

$$\text{memory\_bandwidth} = \frac{32}{30} = 1.1 \text{ bytes/clock cycle (roughly)}$$

Interleaving gives fairly good performance as compared with a word organized main memory, still having the advantage that it does not require a wider data bus. The manner in which addresses are mapped to banks affect the memory's behavior; mapping word addresses to banks, with banks word wide, is a natural solution for the today's 32 bit machines that access the most frequent words in the memory.

An interleaved memory behaves fine in the case of cache misses due to the fact that words are read sequentially, and transferred one at a time to the cache; it is also attractive for write-back caches, as the words in a block are written sequentially into the memory, with the price of a single access time.

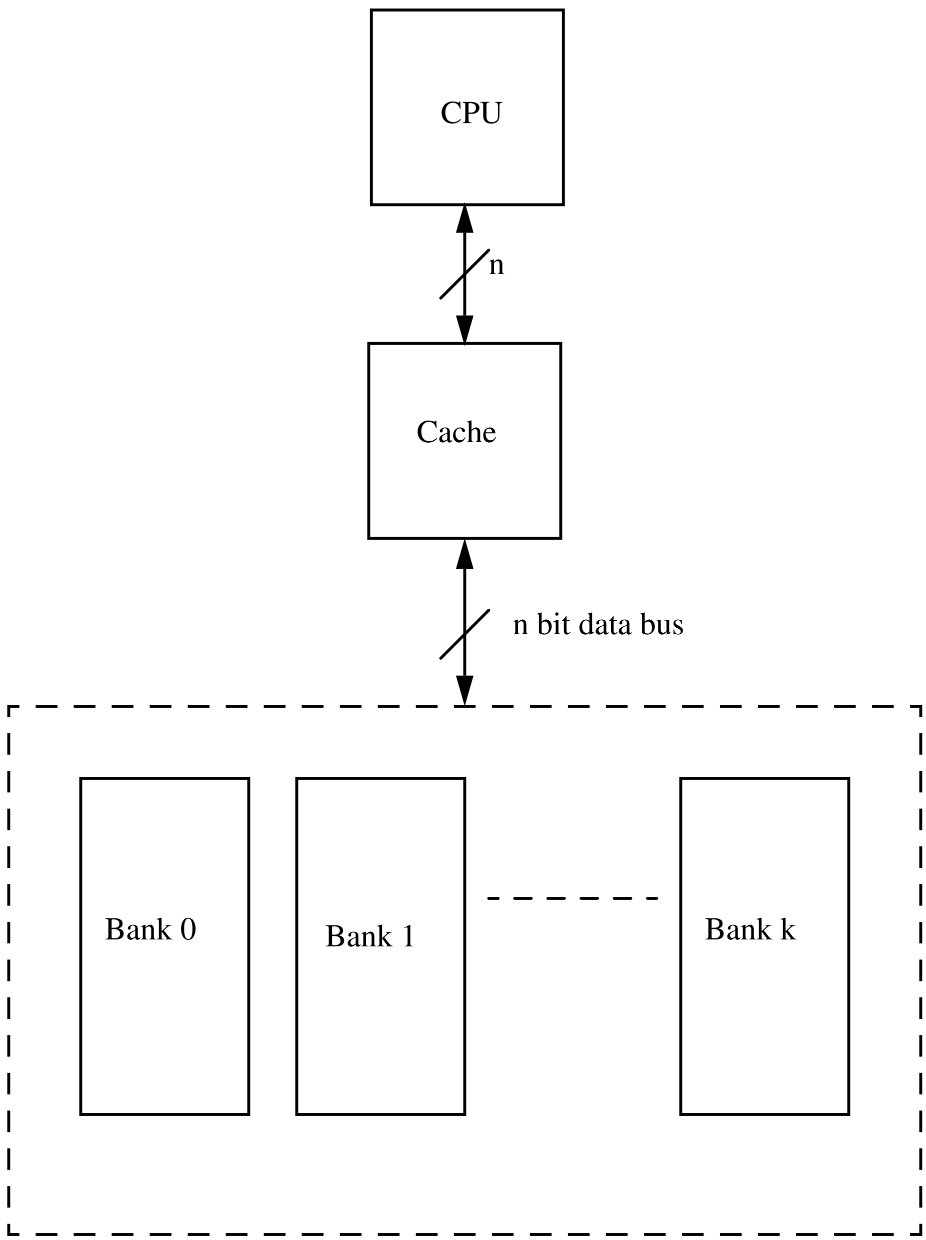


**FIGURE 9.3** Interlaced memory organization. CPU, cache and each memory bank have the same width.

There is however the same drawback as for the wide memory organization, the price that the user must pay to increase the system's memory capacity. In this case the user must add the same number of chips in each bank, for a memory upgrade. Moreover, as the memory chips get larger capacities, it is more difficult to organize the chips in banks, as the following example suggests.

**Example 9.6** MEMORY CAPACITY AND NUMBER OF CHIPS:

The maximum memory a PC can address is 16 MBytes; you have to design a four bank interleaved memory for this system. Each bank is byte organized. How many chips are necessary, using:
a) 1Mx1 chips;
b) 4Mx1 chips;
c) 16Mx1 chips?

**Answer:**
The capacity of a bank is:

$$\text{bank\_capacity} = \frac{\text{memory\_capacity}}{\text{number\_of\_banks}}$$

$$\text{bank\_capacity} = \frac{16\text{MB}}{4} = 4\text{MB/bank}$$

a)

$$n_a = (4 \text{ banks}) * \left(\frac{4\text{Mx8}}{1\text{Mx1}}\text{circuits per bank}\right) = 4 * 32 = 128 \text{ circuits}$$

b)

$$n_b = (4 \text{ banks}) * \left(\frac{4\text{Mx8}}{4\text{Mx1}}\text{circuits per bank}\right) = 4 * 8 = 32 \text{ circuits}$$

c)

$$n_a = (4 \text{ banks}) * \left(\frac{4\text{Mx8}}{16\text{Mx1}}\text{circuits per bank}\right) = 4 * 2 = 8 \text{ circuits ??}$$

We are in trouble because we need at least 8 circuits per bank to ensure the proper number of inputs and outputs (using 16Mx1 circuits); using 8 such circuits per bank means a capacity of:

8 * 16Mx1 = 16 MByte/bank

If the system can access only 16 MB, it results that 3/4 of the main memory is inaccessible.

Given the actual conditions in which the CPU performance increases at a faster pace than the memory performance, memory organizations that reduce the cache penalty tend to become common place.

## Exercises

**9.1** A memory hierarchy is being designed for a system. The following possibilities have to be investigated:
a) cache block size is 1 word, miss rate is 20%
b) cache block size is 4 words, miss rate is 10%
c) cache block size is 8 words, miss rate is 2%
In every case there are 1.4 memory accesses per instruction. Which is the best choice for the main memory? State your assumptions.

**9.2** Explore the possibility of using some of the features the new DRAM circuits offer, to improve the memory performance: here are some of the standard DRAM improvements you might want to consider:
a) nibble mode;
b) page mode;
c) static column.

**9.3** A new idea being studied is to move the cache closer to the memory, more precisely on the same memory chip die; this is tempting because, in the case of read, a whole row is accessed: for a 1Mx1 DRAM memory a row is 1024 bits wide (supposing the die is square). How do you think could this improve the memory performance? For a good introduction in this, you could consider a series of articles in the IEEE Spectrum, October 1992.