Universidade de Aveiro

Mestrado em Engenharia de Computadores e Telemática

Arquitecturas de Alto Desempenho

Memory

Academic year 2021/2022

1. Assume a memory subsystem with 23 address lines.

   1.1. How many bytes, half-words, words and double-words can be stored in it?

   With 23 address lines, the memory subsystem can address 2^{23}, or 8M, memory storage units. Typically, each memory unit represents a byte (8 bits) of storage. Thus, one can store 8M bytes, 4M half-words (16 bits), 2M words (32 bits) and 1M double-words (64 bits).

   1.2. Give examples of addresses (in hexadecimal form) for bytes, half-words, words and double-words stored in the last quarter of its addressing space.

   23 address lines generate addresses in hexadecimal ranging from `000000` to `7FFFFF`. The last quarter of this addressing space starts at `600000` and goes up to `7FFFFF`.

   Thus, as examples, one may consider

   bytes: `6789C8`, `6789C9`, `6789CA`, `6789CB` (any address is valid)
   half-words: `6789C8`, `6789CA`, `70F124`, `70F126` (address least significant bit must be zero)
   words: `6789C8`, `6789CC`, `701230`, `701234` (address 2 least significant bits must be zero)
   double-words: `6789C8`, `6789C0`, `701230`, `701238` (address 3 least significant bits must be zero).

   1.3. Which of the following addresses are classified as *misaligned* if one intends to reference a byte, a half-word, a word and a double-word: 14902_{10}, 3710_{10}, 5555_{10}, 764_{10} and 2760_{10}? Convert them first to hexadecimal form before giving your answer.

   $14902_{10} \equiv 3A36_{16}$ – misaligned address to reference a word or a double-word
   $3710_{10} \equiv E7E_{16}$ – misaligned address to reference a word or a double-word
   $5555_{10} \equiv 15B3_{16}$ – misaligned address to reference a half-word, a word or a double-word
   $764_{10} \equiv 2FC_{16}$ – misaligned address to reference a double-word
   $2760_{10} \equiv AC8_{16}$ – not a misaligned address to reference any of the above.

   1.4. Can the address 11899675_{10} express a valid byte reference for the memory subsystem?

   No, because 11899675_{10} is an address requiring at least 24 addressing lines to be generated and is therefore beyond an addressing space provided by 23 addressing lines.

2. Consider a 128Mbit SDRAM chip organized in 8 banks of 4096 rows x 1024 columns of 4 bits.

   2.1. How many address lines are there? Justify clearly your answer.

   There are 12 addressing lines to reference a specific row, which are multiplexed with 10 addressing lines to reference a specific column. In addition, there are 3 extra addressing lines to reference a specific bank. Thus, there are a total of 15 addressing lines.

   2.2. How many chips are needed to build a 64-bit data wide memory subsystem? Justify clearly your answer.

   Since each chip provides a 4 bit data wide storage unit, 16 chips ($16 \times 4 = 64$) are required to provide a 64-bit data wide memory subsystem.

2.3. What is the storage capacity of such subsystem? Justify clearly your answer.

It has a storage capacity of 32 M double-words, or of 256 M bytes. This can be readily seen by the fact that it is built by using 16 chips in parallel to provide a 64 bit wide data bus, each having a 32M half-byte (4 bits) storage capacity.

2.4. Sketch how the external address bus are mapped internally to make data access as efficient as possible. What is the main concern to be taken into account?

The main concern is to be able to take advantage of the burst transfer mode when transferring blocks of data between the main memory and the cache. So, the column address should be mapped into the least 10 significant bits of an aligned double-word address. Furthermore, one should also take advantage of the multibank organization to improve the transfer of a page between the main memory and the swapping area for a page-based virtual memory organization.