Lab 10 - Memory design

Part 1

This laboratory work presents the design methodology of memory modules for microprocessor-based systems.

1. Theoretical considerations

1.1. Memory types

Memories can be classified based on several criteria:

i. by type of operation allowed:

- ROM memories – Read Only Memory – can only be read (the write operation is performed by the manufacturer)
- PROM memories – Programmable ROM – can be inscribed just one time by the user
- UV-EPROM memories – Ultra Violet Erasable PROM – can be erased by UV rays, then rewritten by the user (limited number of erases – approx. 10-100); both write and erase operations need special devices
- EEPROM memories – Electrically Erasable PROM – can be erased electrically (high number of erases – approx. 1000 - 10000); they don't need special devices for writing or erasing; delete/rewrite cycle take longer than a normal read cycle
- FLASH memories – similar to EEPROM, but designed in another technology
- RAM memories – Random Access Memory – can be read and written during normal system operation

ii. by design technology:

- bipolar memories – designed in TTL technology; high speeds, low capacity, high power consumption, high cost
- MOS static and dynamic memories – MOS transistors; medium to high speed, high capacity (especially the dynamic memories), low cost, easy integration, low power consumption, most used memories in microprocessor systems
- CMOS memories – using complementary MOS transistors; reduced power consumption (low voltage needed when storing information), medium capacity and speed, used for storing configuration settings
- ECL memories – uses coupled emitter; high speeds and power consumption, low capacity, difficult to use in TTL systems

iii. by data retention mode:

- ROM memories – keeps the data even after there is no supply voltage
- Static memories – keeps the information as long as there is a supply voltage present
- Dynamic memories – keeps the data for a limited time (approx. 2ms), needs periodic refresh of the stored data (usually, by simulating read cycles)

iv. by the position in relation with the CPU
   - — cache memories – high speed memory, low capacity, intimately related to the CPU (on the same chip as the CPU)
   - — main memory (operative) – medium-high capacity memory, medium speed, random access, occupies the physical addressing space of the CPU
   - — virtual memory – abstract concept in which the physical addressing space, which is accessible by a program, is extended over the external memory; implementation mechanisms: pagination and segmentation
   - — external memory – very high capacity memory (theoretically, infinite), low speed, semi-random access (uses blocks); implemented on magnetic or optic media (flexible disk, hard-disk, optical disc, etc.), is used for storing data on long term.

1.2. **Access to memory locations**

In microprocessor-based systems, the memory modules are connected to the system bus. The access to the memory locations is performed via transfer cycles, using the bus signals (address signals, data and commands). Each transfer cycle is represented through time diagrams, in which for each signals on the bus, the evolution in time is displayed. The time parameters of a transfer cycle depend on the CPU type, communication policies of the bus and on which types of memory circuits are used. The figures below illustrate the transfer diagrams for a memory read and write cycle.

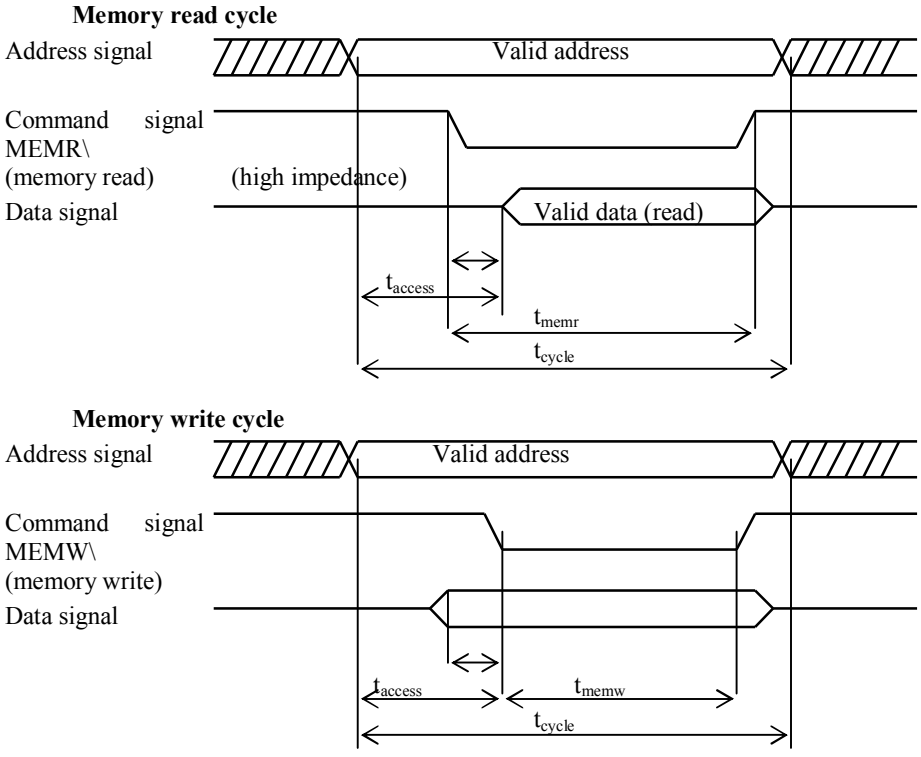


Figure 1 Diagrams of memory read/write

When designing a memory module, one must take into consideration the following:

- Read access time (t_{access}) – the time needed to read a word, beginning with the address validation until the data is present on the data bus; sometimes, the access time is measured in relation to the command signal (MEMR\); if the access time of the memory circuit is greater than the time allowed on the bus, then the memory module interface must extend the transfer cycle by disabling the READY signal to the CPU
- The time to maintain the command signal MEMR\ active (t_{memr})
- The time necessary for a complete read or write cycle (t_{cycle})
- Write access time (t_{access}) – the time needed to write a word in the memory, beginning with the address selection and up to the moment when the write command is generated; just like the memory read cycle, if the access time of the memory circuit is greater than the time allowed on the bus, the memory module interface must temporarily disable the READY signal, in order to extend the transfer cycle; the enable of the memory write signal is measured in comparison with the moment when the data is present on the bus
- The time to maintain the memory write signal (t_{memw}) is usually lower than t_{memr}

Remarks:

- When reading, the data is generated after the address and command signals are activated and is maintained a short time after disabling them
- When writing, the data must be present on the bus before the write signal is activated and must be valid for the entire duration of the command signal
- The access time covers the delays which appear in a memory circuit due to memory address decoding circuits and selection of input and output amplifiers
- The designer must choose correct parameters of the memory circuits, in order to cover any time restrictions of the bus; also, must take in consideration the delay produced by amplification circuits from the interface and bus

2. Designing static memories

In order to design a memory module, a designer must take into consideration some forced parameters:

- Memory capacity
- Memory structure (basic unit of memory access: byte, word, double word, etc.)
- Bus type (number of data signals, address signals, types of command signals, time parameters for transfer cycles, etc.)
- Location of the memory module in the CPU addressing space, expressed as start address (usually, it is multiple of the capacity of a module)
- Basic unit memory circuit which will be used
- Other specific requirements (ex: dual access on two buses, controlled refresh, mechanism to detect and correct errors, etc.)

The design stages of a memory module are the following:

- Main block diagram – will highlight: memory matrix, decoder block, amplification circuits, command circuit
- Memory sub-module design – must comply with the imposed requirements of the memory organization (byte, word, double word, etc.); more basic unit memory modules are connected in series, in order to obtain the desired size
- Memory matrix design – connecting the sub-modules designed earlier, in order to obtain the whole capacity of the memory (connected in parallel)
- Decoder block design – this circuit must generate the selection signals for each sub-module and for the whole memory module; decoding is based on the address and command signals
- Designing the amplification circuits – unidirectional amplification circuits for addresses and bidirectional amplification circuits for data
- Designing the command circuit – this circuit will generate signals for the amplifiers (selection), for the validation of the decoder circuit, for controlling the read and write sequence, for temporarily disabling the READY signal, etc.
- The time restrictions are checked, by adding up the delays for each separate signal; critical are the selection signals (generated by the decoder) and control signals for the amplifiers

Consider the following design example of a memory module which has the following design parameters:

- Capacity: 512k words (= 1 MB)
- Structure: word (16 bit), can address the first or last 8 bit of a word
- Bus characteristics (ISA type)
    - 24 address lines
    - 16 data lines
    - Command signals: memory read (RD\), memory write (WR\)
- Start address: C0.0000 (H)
- Basic memory circuit: 64k * 8 bit

Main block diagram

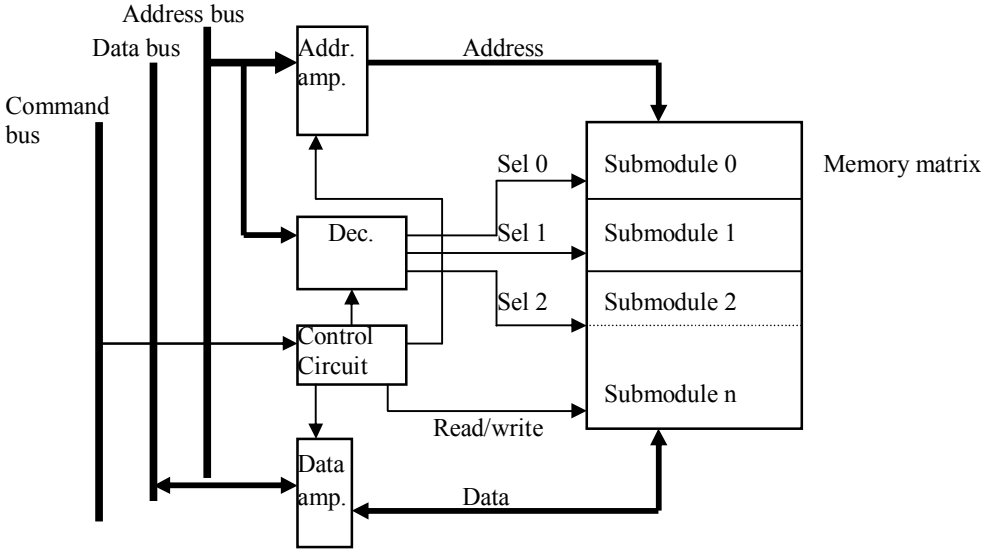
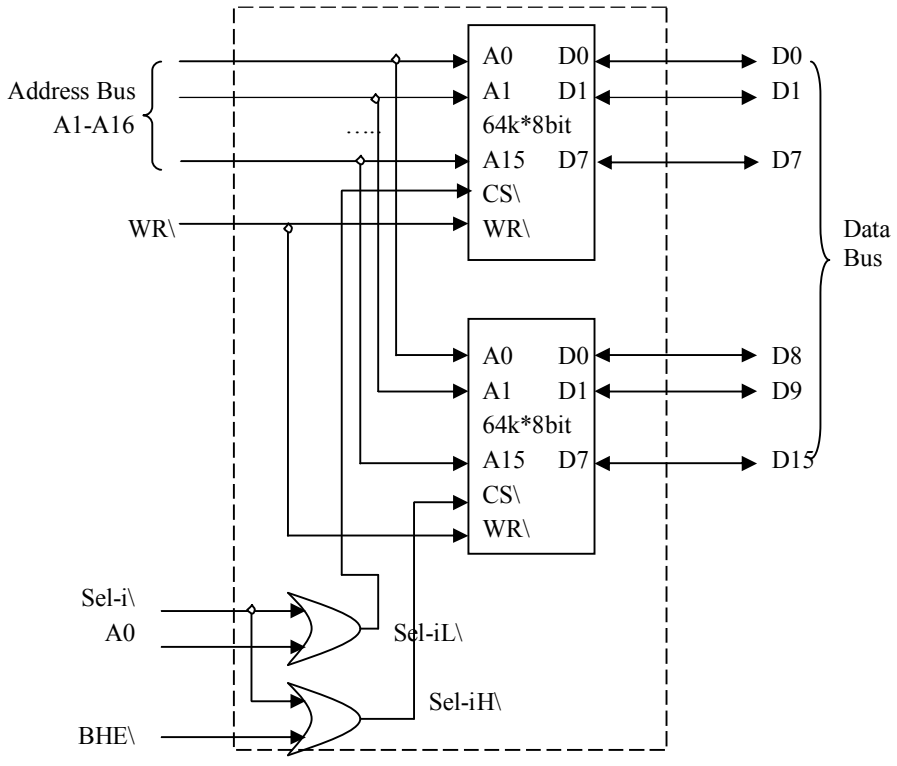


Figure 2. Main block diagram of the memory module

Sub-module block diagram



Sub-module i (64k * 16 bit=128KB)

Figure 3. Block design of sub-module i

Memory matrix

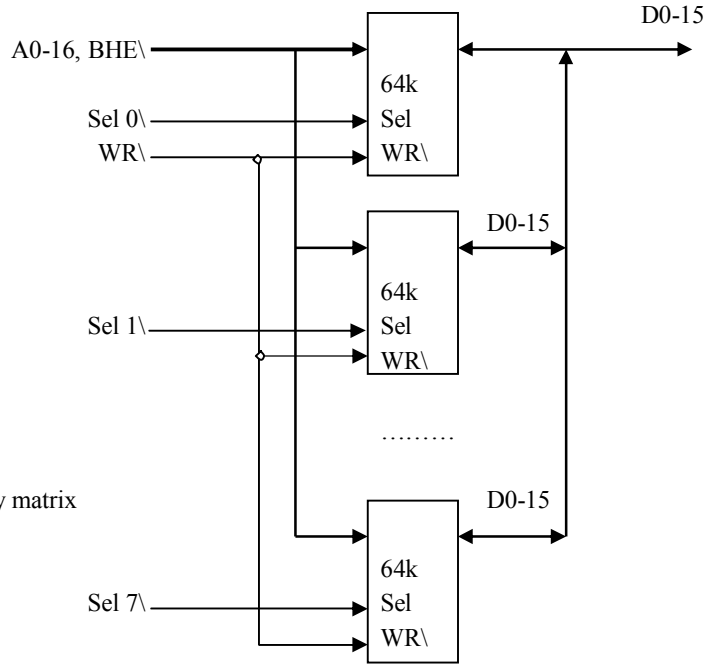


Figure 4. Memory matrix

Decoder circuit

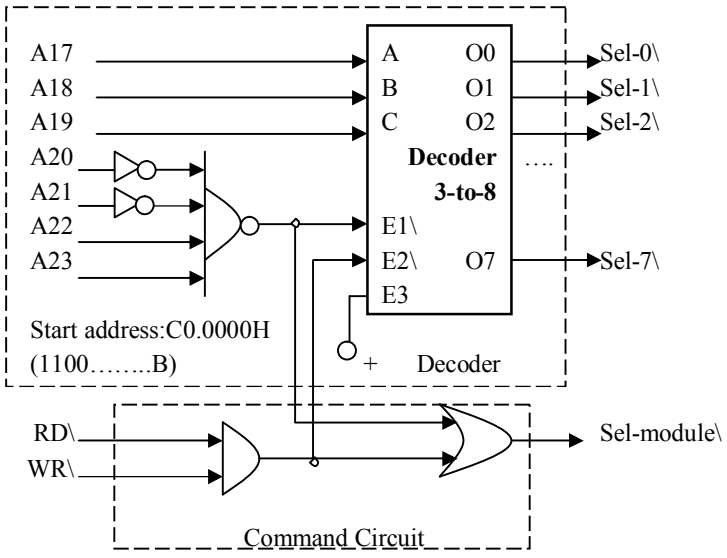


Figure 5. Decoder and control circuit

Amplification circuit

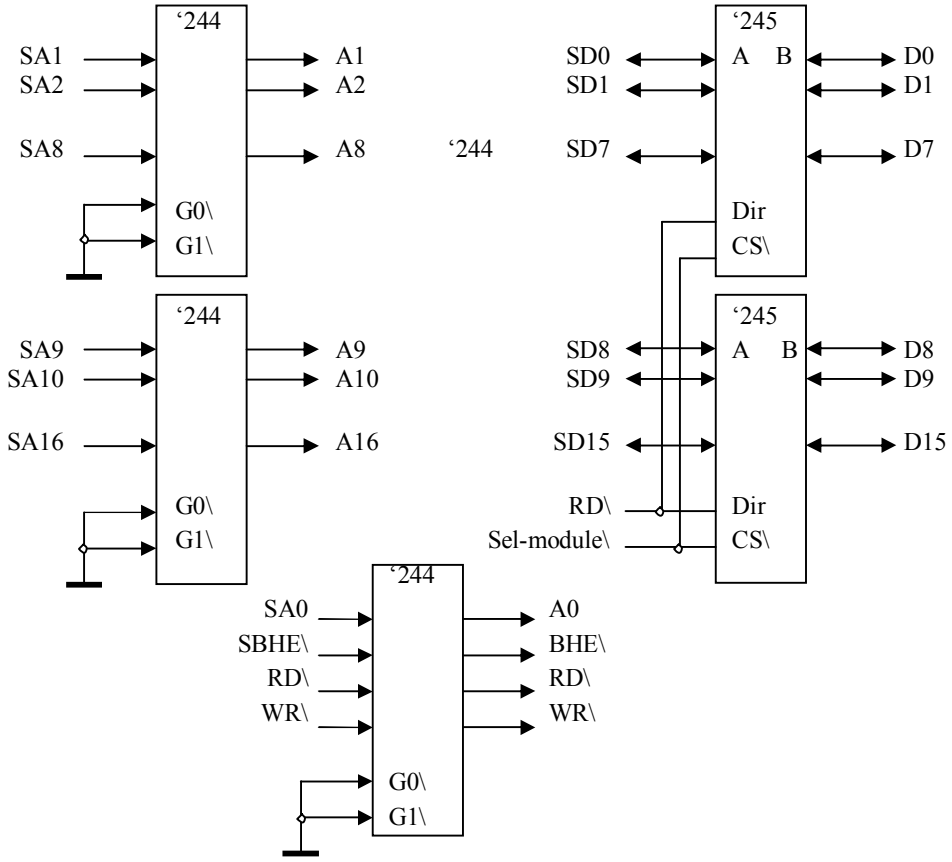


Figure 6. Amplification module

3. Applications

3.1 Design in VHDL a static memory using the structural model. Follow the design parameters mentioned earlier, as well as the block diagrams. Simulate the design and run it on a FPGA board.

3.2 Search online for other designs of static memories. Compare them to your VHDL implementation and point out which is more efficient in terms of structure and organization, hardware complexity, etc.