Exercise 5: Memory Block

During the last exercise, you developed the register file. The register file features two outputs (registers to be read from) and one input (register to be written to). Now, you will implement the next component of the CPU, the memory blocks. There are multiple uses of memory entities: Instruction Memory and Data Memory.

Task 1. Implement a description of a memory entity. Use the log2-function implemented in the last exercise. Your implementation should adhear to the following requirements.

- One output for read access (and address input)
- One input for write access (and address input)
- The memory stores `memory_size` values
- Each value consists of `word_size` bits
- Values from write input are written on rising edge to the memory if writing is enabled.

For this implementation, there are no additional files required.

Task 2. Verify your design via simulation. Use one of the following provided testbenches:

- `memory_tb_lin.vhd` (Simulation Source)
- `memory_tb_fib.vhd` (Simulation Source)

Analyse the testbenches and explain, which values are written to memory. Run Behavioral Simulation for functional verification of your design. Tip: Set the simulation runtime to 4000 ns.

- Simulation > Simulation > `xsim.simulate.runtime`

Now, you will explore the mapping of your memory entity to FPGA resources.

Task 3. Run Synthesis and Implementation. For this, you can use the provided constraint file.

- `constraints.xcd` (Constraint)

Explore the utilized resources. Now, report utilization, power and timing. Utilization and Timing can be explored directly. For Power Analysis, export the saif-file from Post-Implementation-Simulation. Look at the following settings:

- Simulation > Simulation > `xsim.simulate.saif`

Task 4. In the previous task, you explored a mapping of the entity to the FPGA resources. What other components are available for this mapping? Modify the synthesis settings and rerun synthesis and implementation and the evaluation of your design. The following settings might help your endavour:

- Synthesis > `max_bram`

Bonus Task 1. Summarize your findings. Which synthesis setting yields the best mapping for a given memory size? Which metrics determine this decision?