

Goal

- Synthesize the timing characteristics of several embedded memory components, taking into account the electrical propagation delays through logical gates and along wires.

The SPSMALL Memory

A typical case study of the project VALMEM is the SPSMALL memory.

Description

- The SPSMALL memory is a small memory with a maximum total capacity of 64 kbits (3 to 512 words of 2 to 256 bits). We first chose to model the smallest memory consisting in three words of two bits, which leads to a netlist of 305 transistors.

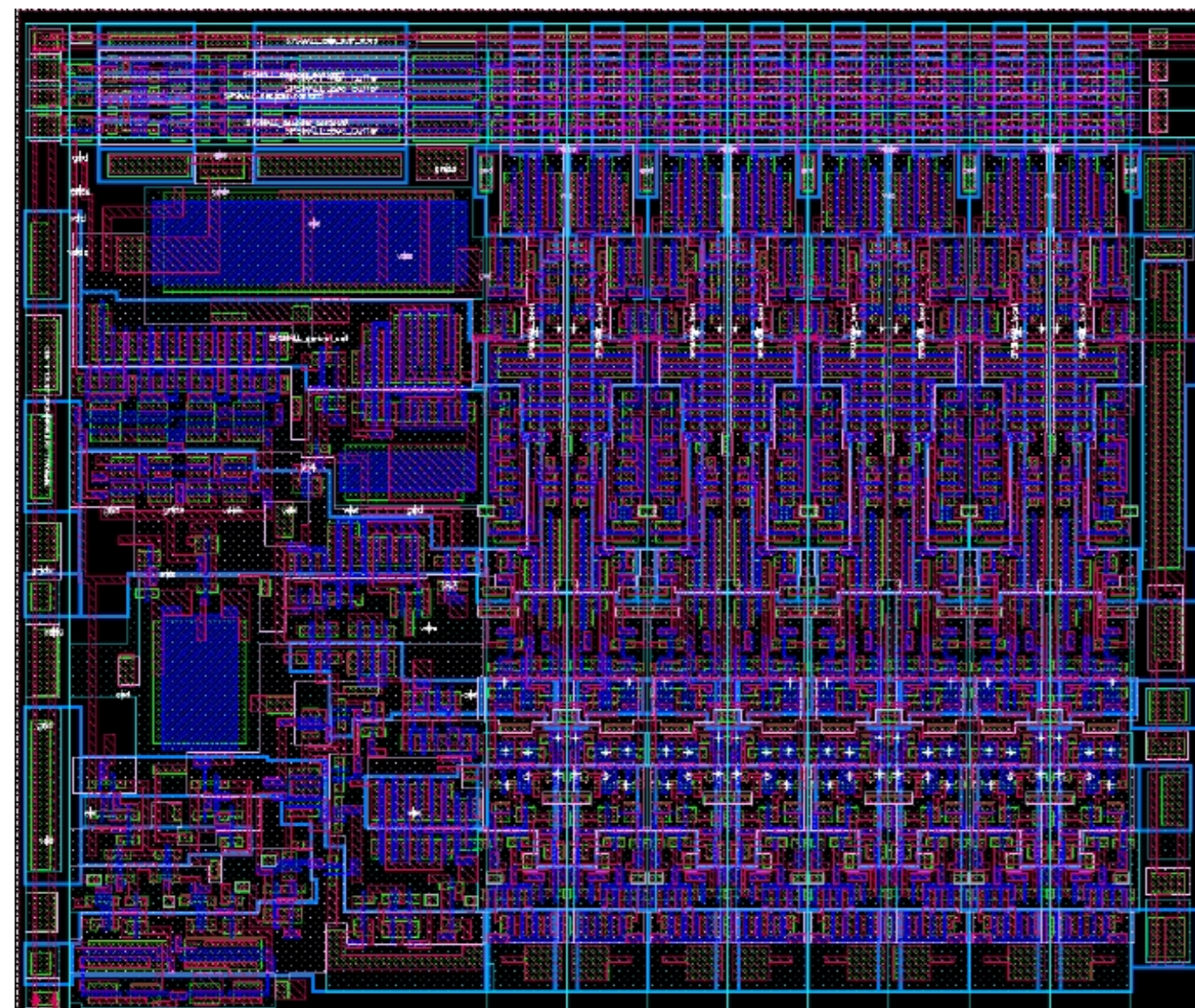


Figure: Masks of SPSMALL

Description of the environment

- The memory is embedded into a synchronous environment: all operations (*read*, *write*) are synchronized with the rising edge of a unique clock (CK). Address (A), input data (D), write enable not (WEN) signals are latched in at the rising edge of the clock (CK). WEN precises the direction of a memory operation (read or write). Output data are internally latched at the completion of each access, and are valid until the next positive edge of CK. During *write* operations, data written into the memory are copied back to the output data pins (write-through).

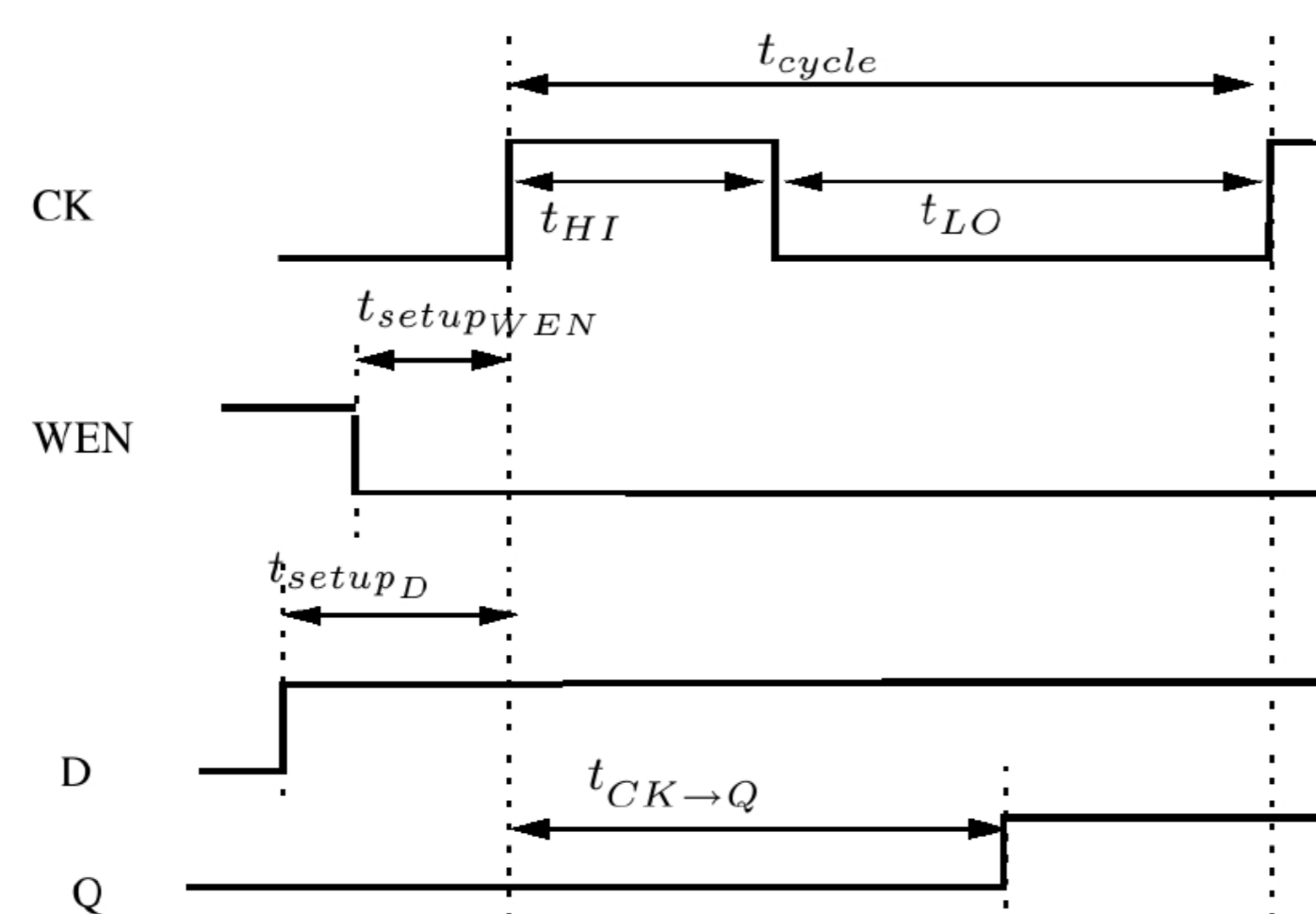


Figure: Environment of SPSMALL

Timing description given by the datasheet

- The datasheet given by the constructor defines the set of timing values corresponding to a nominal use of the memory. For instance, it defines the minimal clock cycle t_{cycle} . Moreover, two timings are defined for each input signals: the setup timing, *setup* and the hold timing, *hold*. These timings define an interval around the rising edge of the clock in which the input signals have to be kept stable to be correctly interpreted by the system.

Property to be Verified

- The response timings of the memory correspond to the delay between the rising edge of the global clock, sampling an operation to be performed, and the results of the memory produced on the output pins.
- In particular, we aim at verifying the response time $t_{CK \rightarrow Q}$, corresponding to a *write* operation. We are interested in showing that this response time $t_{CK \rightarrow Q}$ is smaller than the the time given by the datasheet (*taaw*), i.e. $t_{CK \rightarrow Q} \leq taaw$.

Methodology

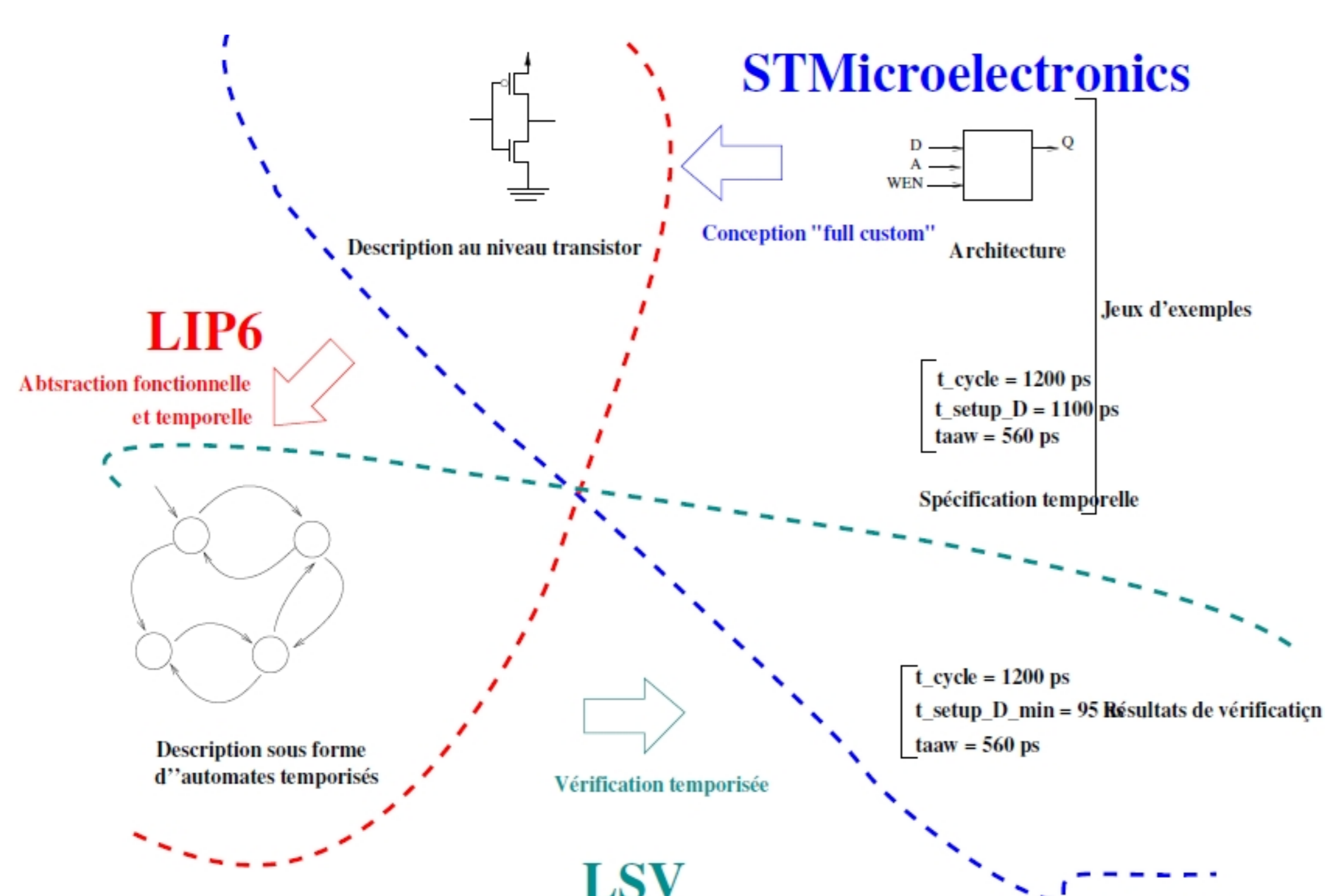


Figure: The organization of the project

From Transistors to VHDL

Transistors

- ST-Microelectronics provides the project with the description of the transistor netlist, with timed delays.

Functional abstraction

- The description in transistors is automatically translated into a description in VHDL, using the tool *Mygal* developed in the framework of this project. The VHDL code describes the memory under the form of a graph of functional components.

Timed abstraction

- The description of the delays for the transistors is automatically translated into timed intervals of delays for each gate.

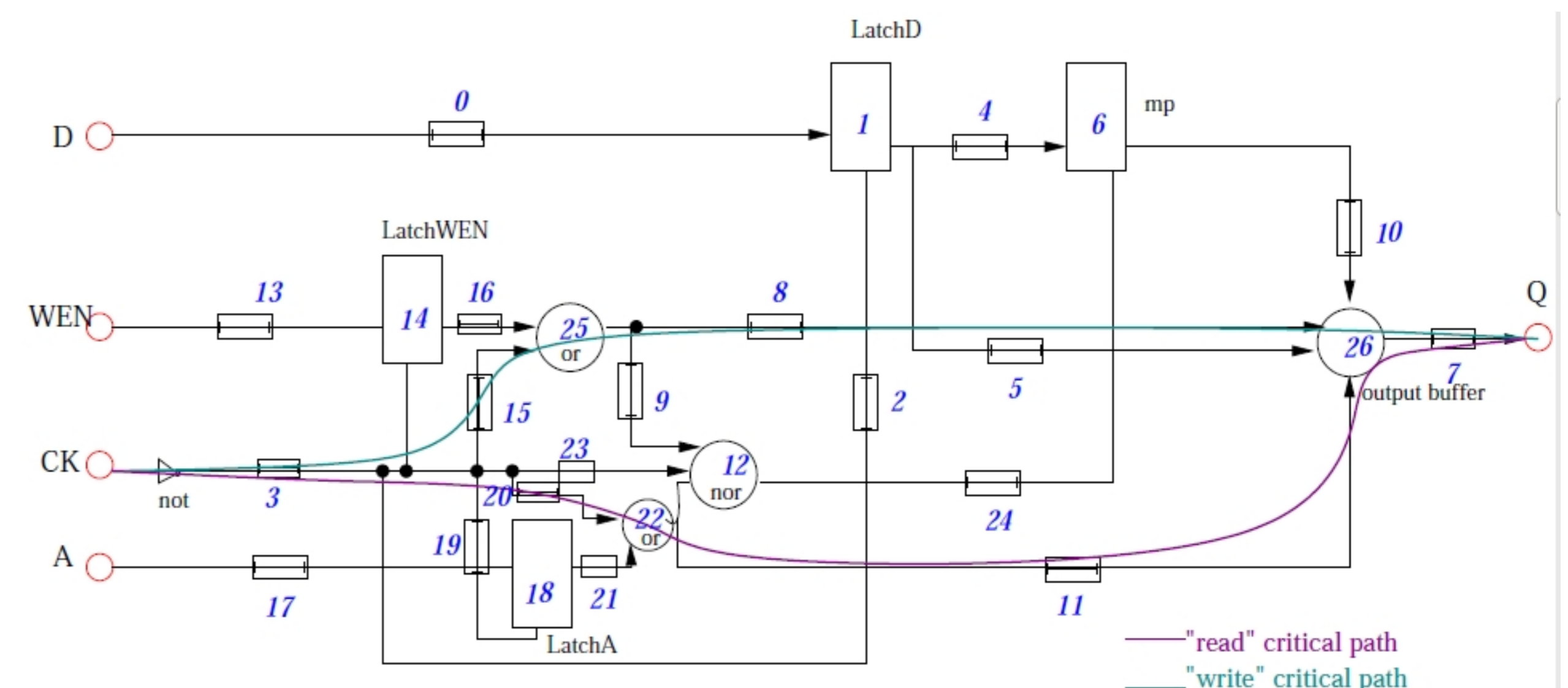


Figure: Abstract model of the SPSMALL memory

From VHDL to Timed Automata

Modeling with timed automata

- This description in gates and the delays are then automatically translated into a description in *timed automata*, an extension of finite state machines, using the tool *VHDL2TA* developed in the framework of this project.

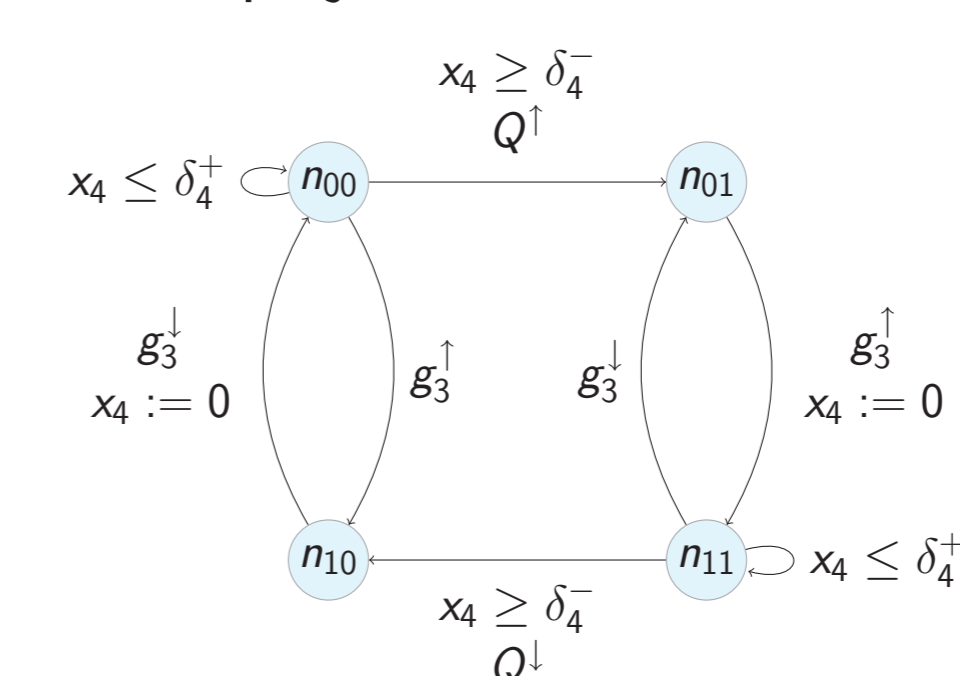


Figure: Example of timed automaton modeling a gate

From Timed Automata to Timing Constraints

Tool IMITATOR

- Using the tool *IMITATOR*, developed in the framework of this project, we synthesize automatically a set of symbolic linear constraints on the timings seen as parameters. This will allow us to *optimize the crucial timings* of the memory, such as setup and hold timing of input signals.

References

- Remy Chevallier, Emmanuelle Encrenaz, Laurent Fribourg and Weiwen Xu. *Timed Verification of a Generic Architecture of a Memory Circuit using Parametric Timed Automata*. International Journal of Formal Methods in System Design, vol 34(1), pp 59-81, 2009.
- Étienne André, Thomas Chatain, Emmanuelle Encrenaz and Laurent Fribourg. *An Inverse Method for Parametric Timed Automata*. International Journal of Foundations of Computer Science, vol 20(5), pp 819-836, 2009.
- Étienne André. *IMITATOR: A Tool for Synthesizing Constraints on Timing Bounds of Timed Automata*. In Martin Leucker and Carroll Morgan (eds.), ICTAC'09, LNCS 5684, pages 336-342. Springer, 2009.

The Team

