Creation of a protocol monitor

Practical lesson 1, Advanced Programmable Logic Systems.

Hipólito Guzmán Miranda
Departamento de Ingeniería Electrónica
Universidad de Sevilla
hguzman@us.es

Creation of a protocol monitor

The objective of this lesson is to create a VHDL entity, not necessarily synthesizable, to facilitate the verification of a digital design. This entity monitors pin movement on a bus and, if a valid pin movement is detected, generates an output transaction containing the received 32-bit data word, along with a field indicating that the data is valid.

The protocol corresponds to the same one used in the previous exercise, customized according to the student's ID number. It is recommended to use the `protocol_common` package, which was used in the previous exercise.

Monitor operation:

The monitor must receive as inputs a clock signal, `clk`, and the protocol signals: `data`, `ena`, `startp` and `endp`. It will have a single output, `output_tran`, of the `record` type `protocol_type`. The entity's ports are described in the following table:

Nombre	Tipo de dato	Direction	Function
`clk`	`std_ulogic`	`in`	Clock input. If creating a non-synthesizable entity, it is recommended to sample the received data on the falling edge of the clock (`falling_edge(clk)`). If creating a synthesizable entity, the same clock edge used for driver development can be used.
`data`	`std_ulogic_vector` (width depends on the student's ID number)	`in`	Input data
`ena`	`std_ulogic`	`in`	Protocol enable signal
`startp`	`std_ulogic`	`in`	Transmission start signal
`endp`	`std_ulogic`	`in`	Transmission end signal
`output_tran`	`protocol_type`	`out`	Output transaction

Once the correct transmission of a complete 32-bit data has been detected, the monitor should generate an output transaction, simply by indicating the received data in the `data` field of `output_tran`, and driving a '1', for one clock cycle, in the `valid` field of `output_tran`.

Realization and evaluation of the practical lesson:

The protocol monitor adapted to the student's ID number (DNI/NIE/etc) must be developed. This protocol is exactly the same as the one used in the previous practical session.

The basic functionality of the protocol monitor is to generate a transaction if the `ena` signal has been activated and enough data blocks have been sent through the interface to reconstruct a 32-bit word (i.e., if the data protocol is followed approximately, without trying to detect errors).

Additions to the basic functionality will be considered when grading, such as:

- Ensuring that the monitor does not generate any transaction if there has been an incorrect pin movement (i.e., one that violates the protocol).
- Reporting when a transaction on the bus starts and ends using the `report` statement.
- Reporting that an error has been detected in the pin movement on the bus.
- Reporting that an error has been detected in the pin movement on the bus, indicating the cause (i.e., which part of the protocol was violated).

A report on the exercise must be prepared, describing the work performed and demonstrating that the protocol monitor has been implemented correctly. The demonstration of functionality must be based on simulations and can be done by reusing the driver block developed in the previous exercise. In addition to the report, all developed code must be submitted, including the testbenches and the `protocol_common` package.