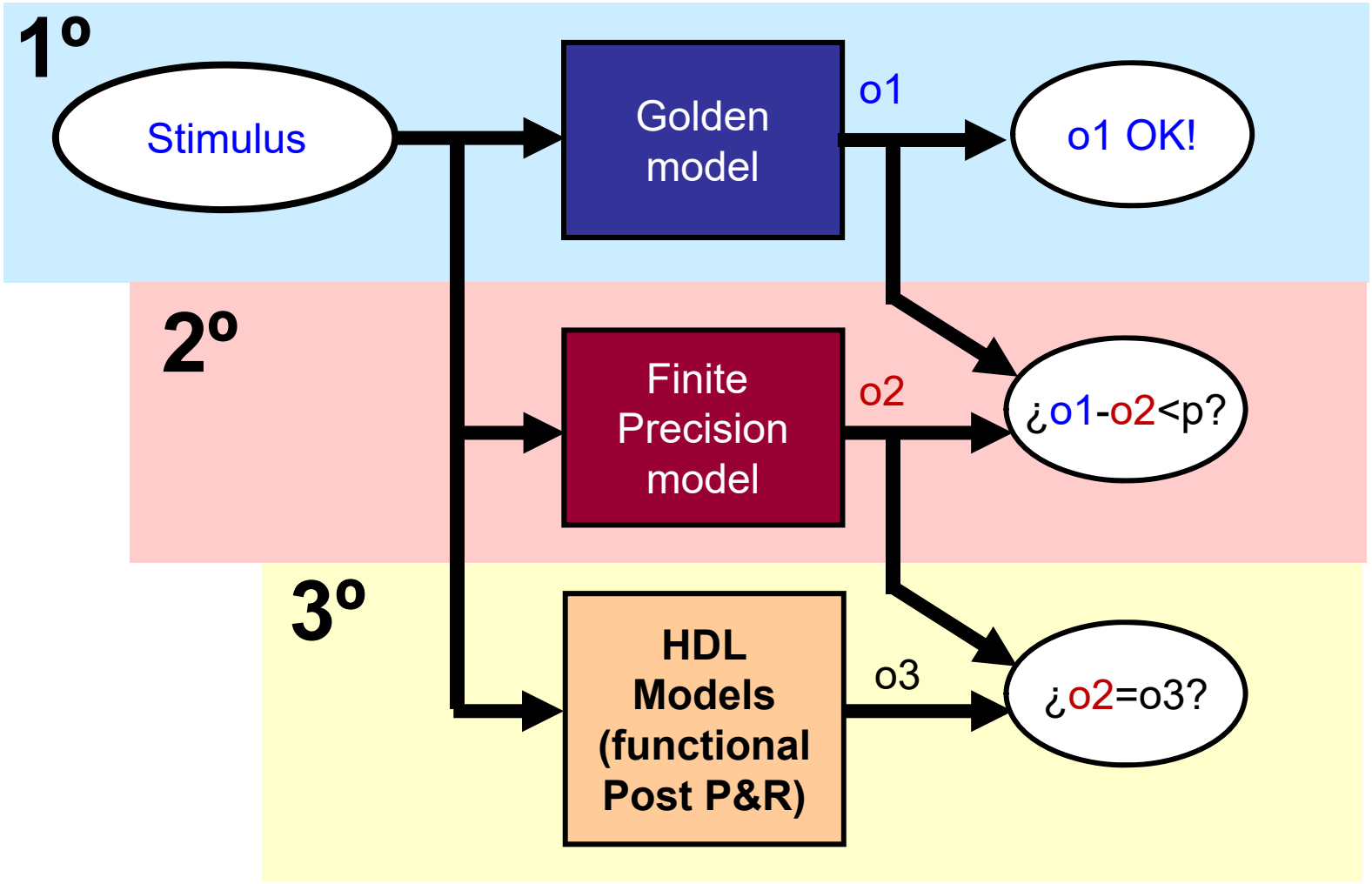
Verificación con I/O de texto

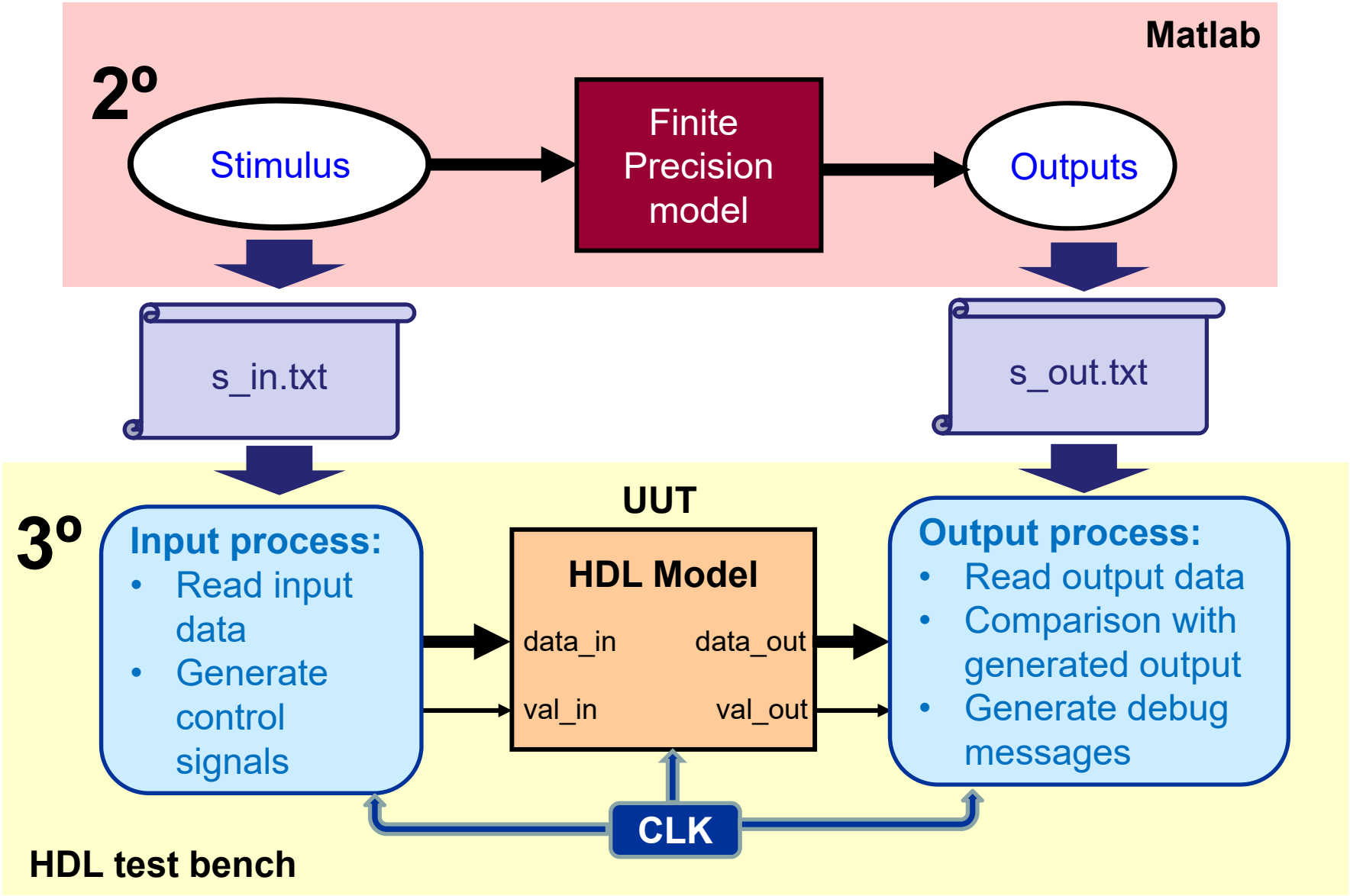
Procesado Digital de la Señal en FPGA

2021/2022

Flujo de verification



Flujo de verification



Verilog system functions for file I/O

Open a file

- $fopen opens an existing file for reading (“r”) or writing (“w”)
- it returns an integer containing the file number or 0 if there was an error

```
integer f;
f = $fopen(“filename”,”r”);
```

Read formatted text

- $fscanf reads formatted text from the file according to the format and writes the results to args
- it returns the number of successful assignments performed

```
integer read_data, scan_num;
scan_num = $fscanf(f, "%d\n", read_data);
```

%d → decimal
%b → binary
\n → newline

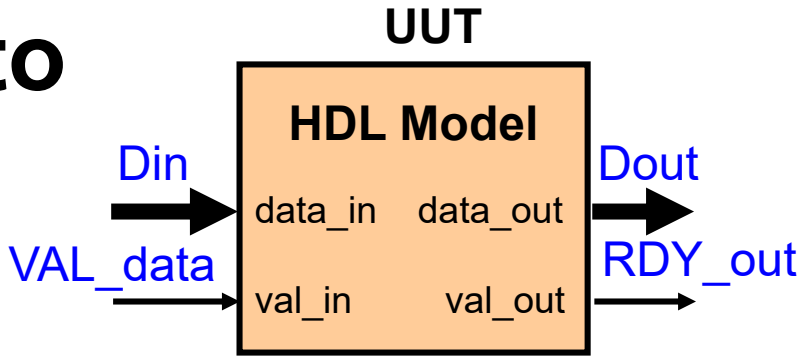
Test for end of file

- $feof tests for end of file
- If an end-of-file has been reached while reading from the file, a non-zero value is returned; otherwise, a 0 is returned

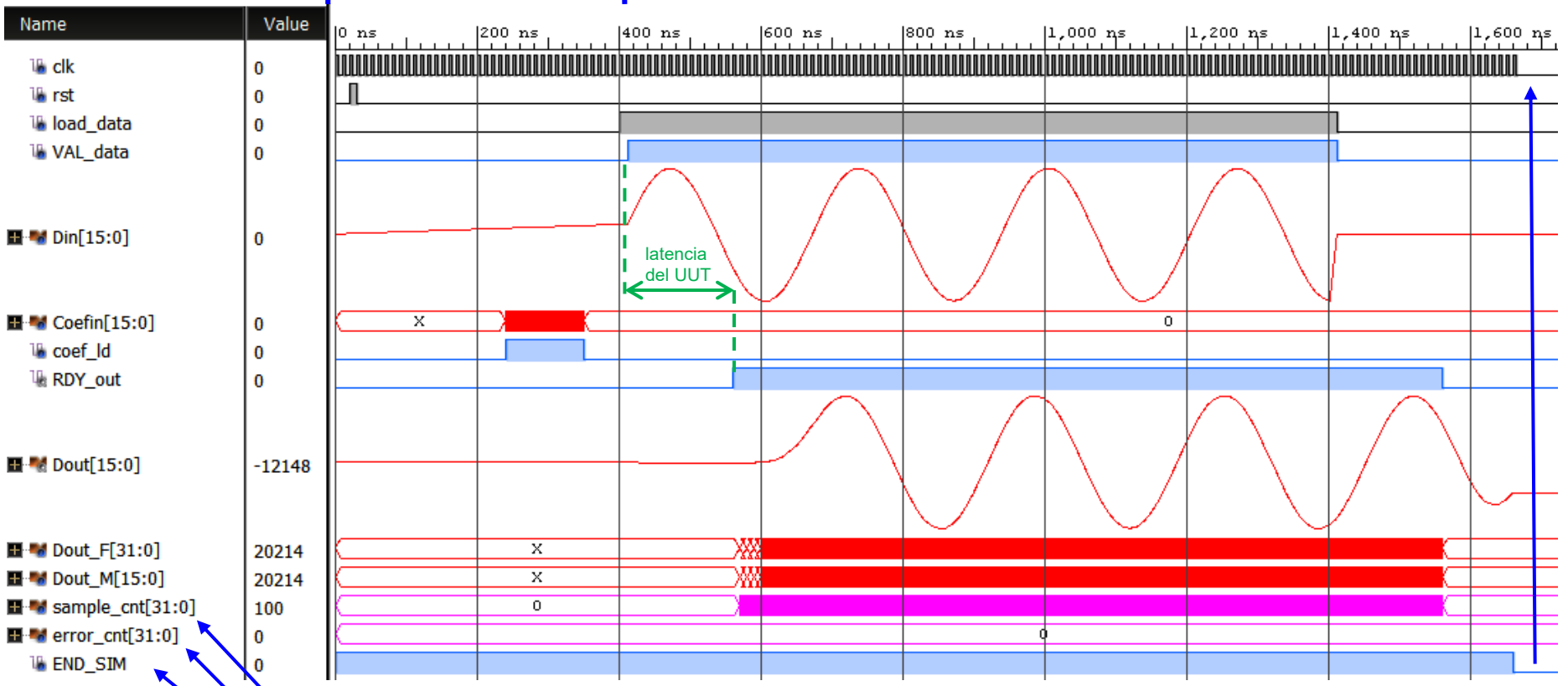
```
$eof(f);
```

More info: Reading files from Verilog models (http://www.angelfire.com/in/verilogfaq/pli.html)

Test Bench con I/O texto



proceso "**initial**": prepara el sistema e indica el comienzo de la lectura del fichero de datos



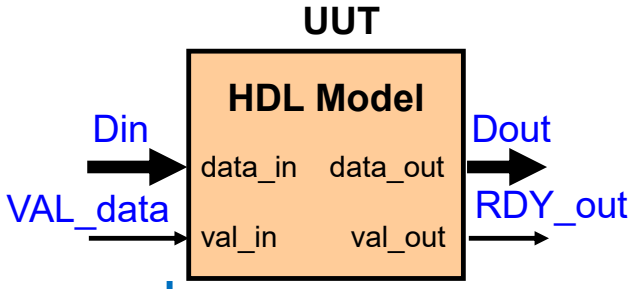
contador de muestras procesadas

contador de errores en la salida del UUT

Indicador de final de simulación

el final de simulación para e clk

Entrada de datos




```
always@(posedge clk)
    if (load_data)
        begin
            scan_data_in = $fscanf(data_in_file, "%d\n", Din_file);
            Din <= #(PER/10)  Din_file;
            VAL_data <= #(PER/10)  1'b1;
            if ($feof(data_in_file))
                begin
                    VAL_data <= #(PER+PER/10)  1'b0;
                    load_data <= #(PER/10) 1'b0;
                    END_SIM = #(25*PER) 1'b0;
                end
        end
```

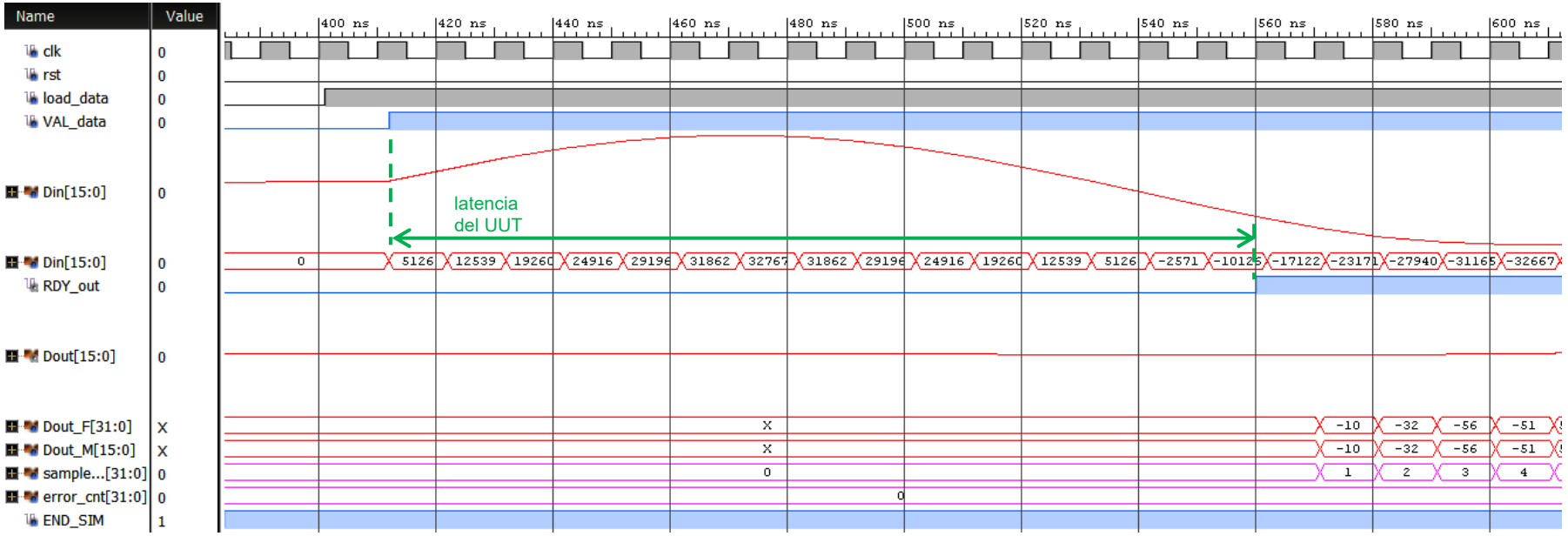
señal del proceso initial para iniciar la lectura de datos

data_in del UUT

val_in del UUT

para la lectura de datos

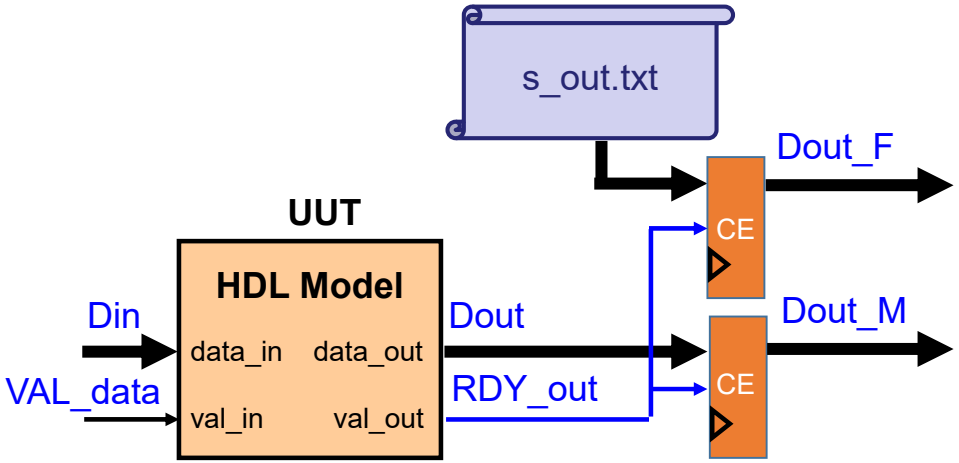
indica fin de simulación

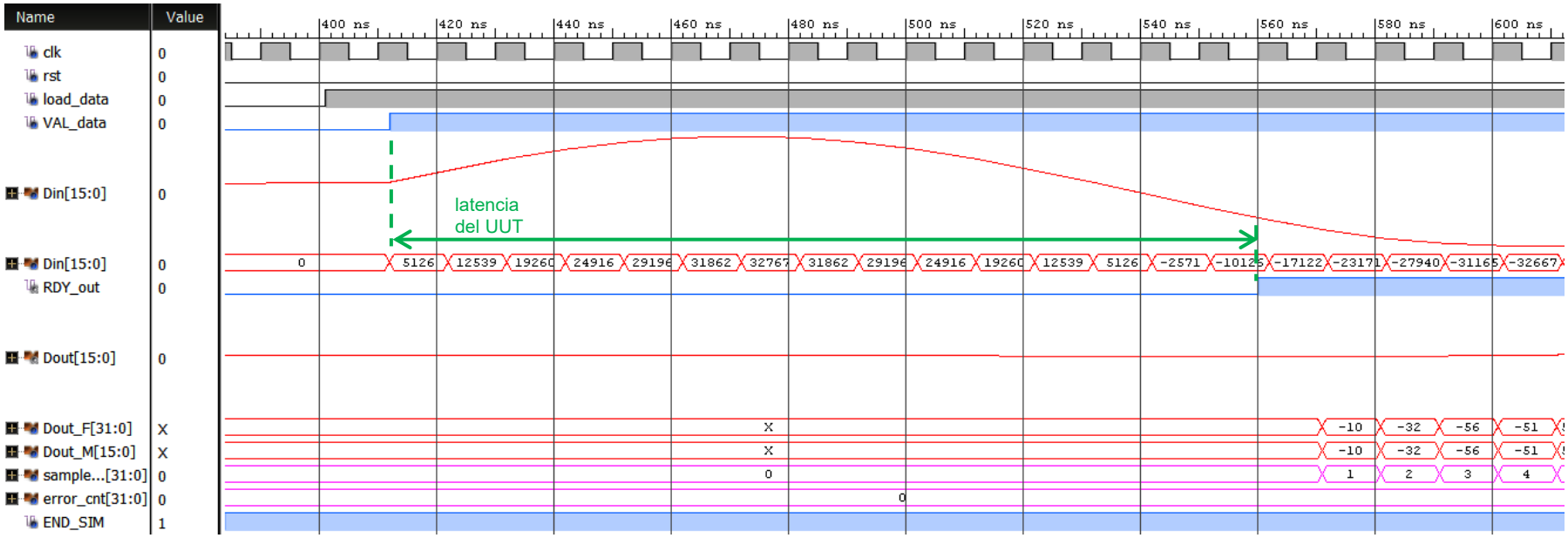


Salida de datos

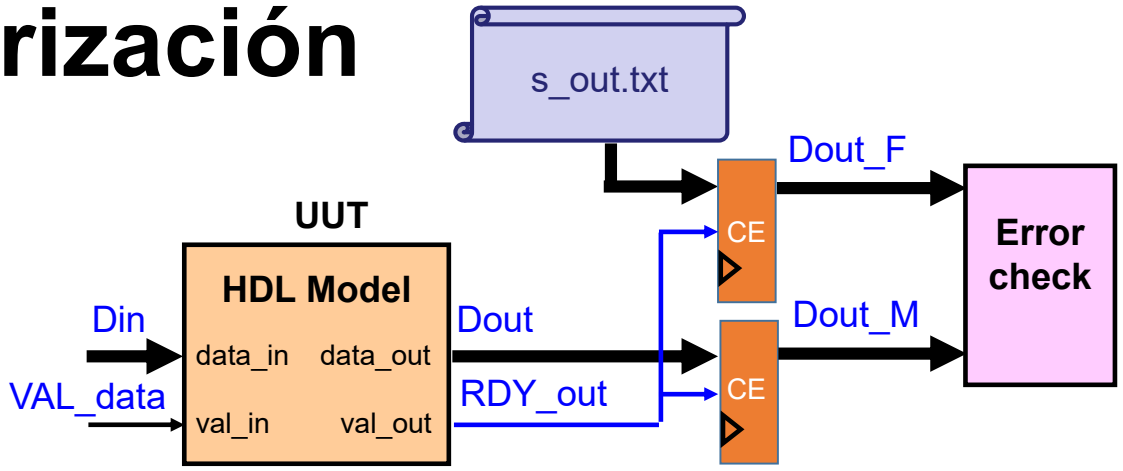
val_out del UUT

```
always@(posedge clk)
        if (RDY_out)
         begin
              sample_cnt = sample_cnt +1;
              if (!$feof(data_out_file))
                  begin
                       scan_data_out = $fscanf(data_out_file, "%d\n", Dout_file);
                       Dout_F <= #(PER/10) Dout_file;
                       Dout_M <= #(PER/10) Dout;
                  end
         end
```

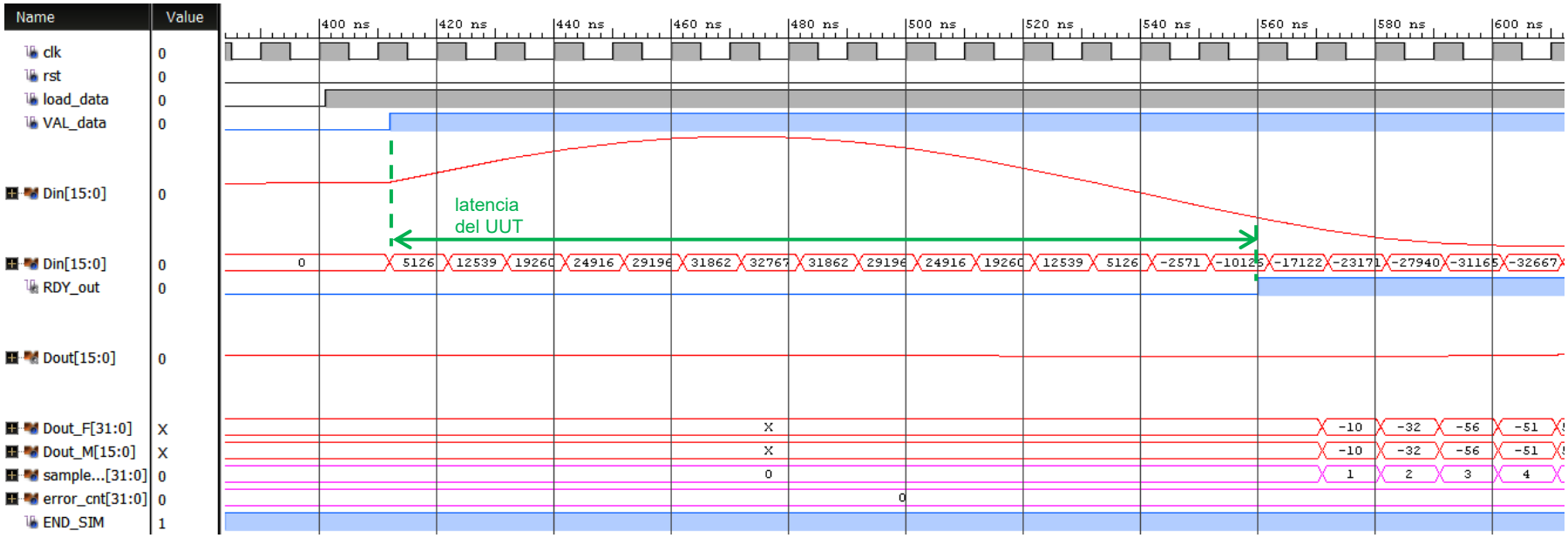




Cuenta y monitorización de errores




```
always@(Dout_F,Dout_M)
        if (Dout_F != Dout_M)
             begin
                  error_cnt = error_cnt + 1;
                  $display("Error in sample number ","%d", sample_cnt);
                  $display("    Dout_M = %d", Dout_M);
                  $display("    Dout_F = %d", Dout_F);
             end
```



Fin de la simulación

```
always@(END_SIM)
    if (!END_SIM)
        begin
            $display("Number of checked samples ","%d", sample_cnt);
            $display("Number of errors ","%d", error_cnt);
        end

endmodule
```

Mensaje en consola del simulador:

```
Number of checked samples          100
Number of errors                   0
```