Writing parameterized modules

Proposed problems

Oprițoiu Flavius
flavius.opritoiu@cs.upt.ro

October 25, 2024

Problem 1

Construct a parameterized 4-to-1 multiplexer, named *mux2s*. The module should be parameterizable by its width, having the following interface:

```
1  module mux2s #(
2          parameter w = 4                  //width parameter
3  )(
4          input [w-1:0] d0,d1,d2,d3,  //4 data inputs
5          input [1:0] s,              //selection input
6          output [w-1:0] o            //data output
7  );
```

Build a testbench for verifying the multiplexer. In the testbench, the parameter *w* will have a value of 8 and the multiplexer should be tested with at least 8 distinct input configuration.

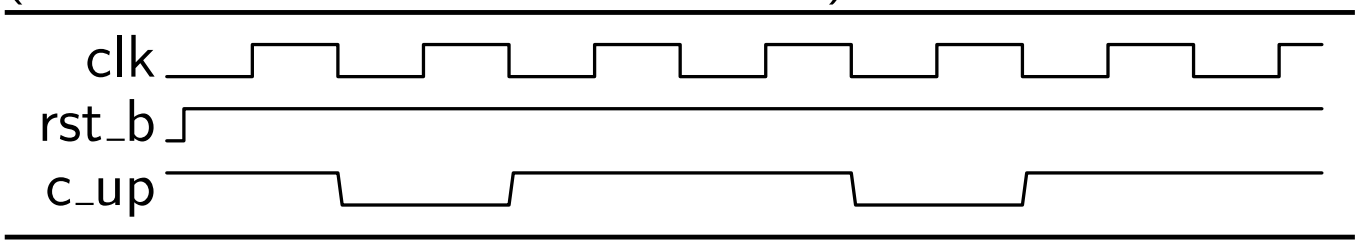
Note: Use the `$urandom()` Verilog system call for generating 32 bits of random data.

Problem 2

Construct a synchronous counter, named *count11*. The module should be parameterizable with its width and initial value. The counter's interface, besides *clk*, includes the following signals:

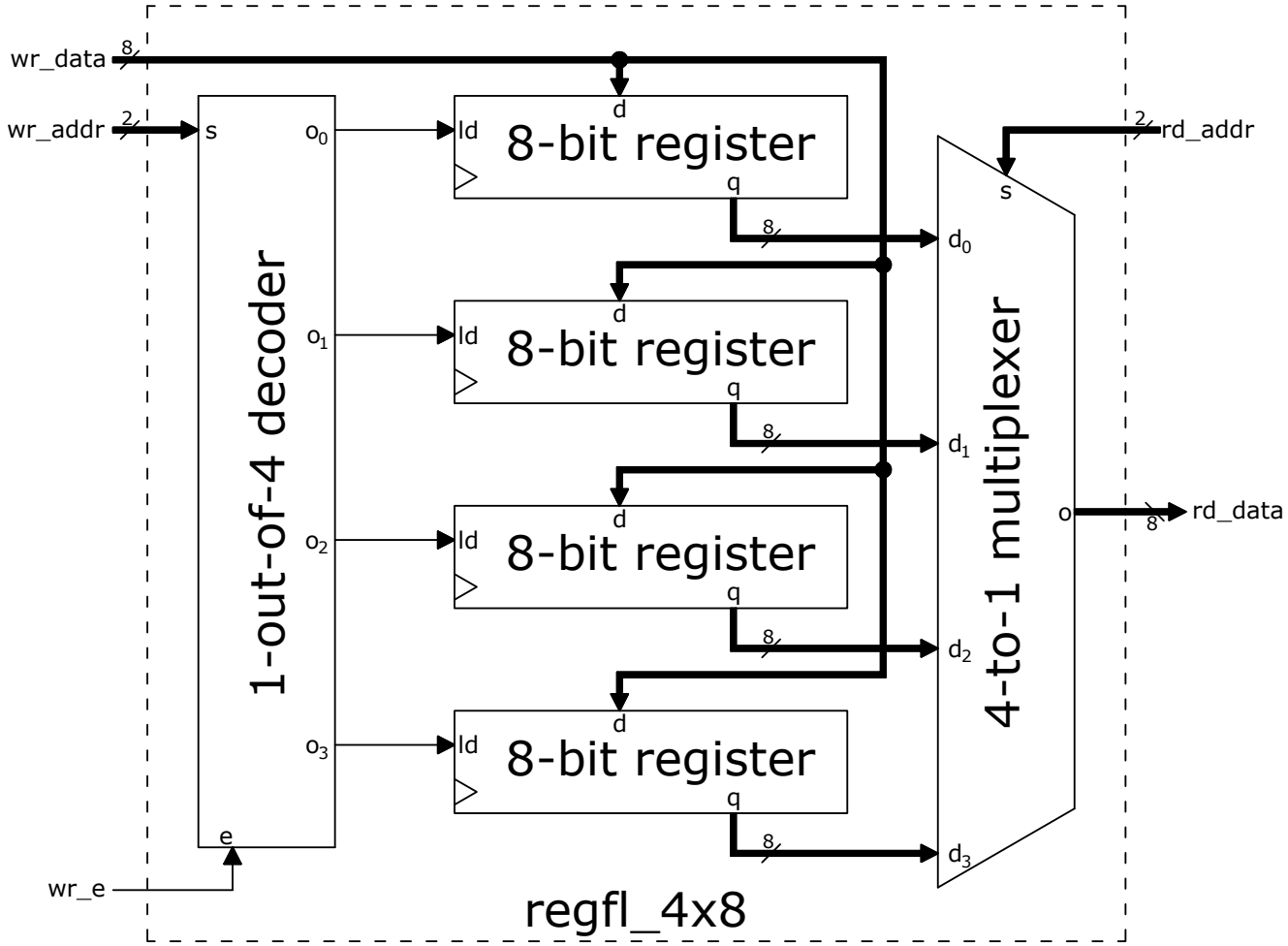
- *rst_b*, asynch., active low, sets content to *initial value*
- *c_up*, synch., active high, increments counter's value by 11
- *q*, output, counter's content

Parameterize the counter on 10 bits with an initial value of `'h3fb` and test it with the inputs given in the diagram bellow ($T_{clk} = 200ns$, $Pulse_{rst\_b} = 10ns$):



Problem 3

Construct the *4x8* register file architecture bellow



The clock and reset lines were omitted for clarity.

Note: The Verilog code of a width-parameterizable, parallel load register is available here

Problem 3 (contd.)

The *4x8* register file will have the following interface

```
1  module regfl_4x8 (
2    input clk,
3    input rst_b,//asynch
4    input [7:0] wr_data,
5    input [1:0] wr_addr,
6    input wr_e,
7    output [7:0] rd_data,
8    input [1:0] rd_addr
9  );
```

The register file has no read enable input, *rd_e*, meaning that at any moment, one of the 4 internal registers' content will be delivered to *rd_data*.

Problem 3 (contd.)

Test the *4x8* register file with a testbench that generates the inputs as in the diagram bellow ($T_{clk} = 100ns$, $Pulse_{rst\_b} = 5ns$)