

Arrayed instantiation and generate constructs

Proposed problems

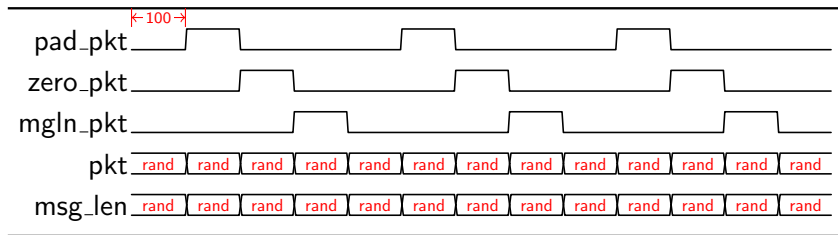
Oprîtoiu Flavius
flavius.opritoiu@cs.upt.ro

November 4, 2024

Problem 1

Build module *pktmux* (slide 16). Include in your solution:

1. the script file, "run_pktmux.txt"
2. a testbench generating inputs as in the timing diagram bellow



For generating 64-bit random values, use the Verilog task bellow:

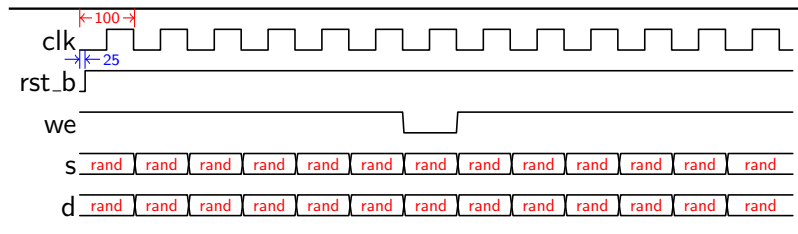
```
1 task urand64(output reg [63:0] r);
2     begin
3         r[63:32] = $urandom;
4         r[31:0] = $urandom;
5     end
6 endtask
```

To use the above task, call it by: `urand64(msg_len);`

Problem 2

Build module *regfl* (slides 14-15). Include in your solution:

1. the script file, "run_regfl.txt"
2. a testbench generating inputs as in the timing diagram below



For generating 64-bit random values, use the *urand64()*; Verilog task.

Problem 3

Build module *sha2indpath* (slides 12-13). Include in your solution:

1. the script file, "run_sha2indpath.txt"
2. a testbench generating inputs as in the timing diagram bellow

