Lecture07Quiz

Due Oct 29 at 11:55pm **Points** 8 **Questions** 8
Available Oct 22 at 9:55am - Oct 29 at 11:59pm **Time Limit** 11 Minutes

This quiz was locked Oct 29 at 11:59pm.

Attempt History

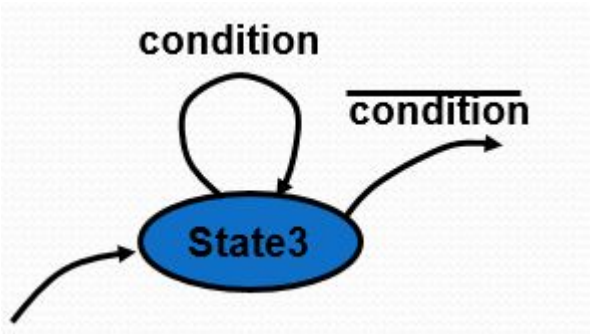
	Attempt	Time	Score
LATEST	Attempt 1	11 minutes	7.33 out of 8

ⓘ Correct answers are hidden.

Score for this quiz: **7.33** out of 8
Submitted Oct 29 at 11:51pm
This attempt took 11 minutes.

Question 1 — 1 / 1 pts

The following excerpt from a state diagram represents:



- ○ A "for" loop with a fixed number of iterations
- ◉ A while loop implemented in a synthesizeable manner

- A bad way of implementing a "while" loop
- An infinite loop

Question 2

1 / 1 pts

$**readmemh** is easier to use than $**readmemb** because with $**readmemb** the file is specified in binary, but with $**readmemh** the file is just an ASCII text file. false

With both $**readmemh** & $**readmemb** the address data in the file is optional true

Answer 1:

false

Answer 2:

true

They are both ASCII text files. One just represents the numbers in binary and the other in hex.

Question 3

1 / 1 pts

```
fork
  begin : timeout1          // This
    repeat(70000) @(posedge clk);
    $display("ERROR: timed out wai
    $stop();
  end
  begin                     // This
    @(posedge cmd_rdy);
    disable timeout1;       // Cance
  end
join
```

The code shown above either waits 70000 clocks, or until "cmd_rdy" goes high

The instructor was rather excited about this code. At least as excited as that mono-tone boring guy gets

Answer 1:

either waits 70000 clocks, or until "cmd_rdy" goes high

Answer 2:

was rather excited about this code. At least as excited as that mono-tone boring guy gets

Partial

Question 4

0.67 / 1 pts

We can write to more than 1 file with a single verilog statement false .

If we name both the **begin** and **end** in a **begin**/**end** pair and the compiler can inform us if we have a mismatch true

The free version of ModelSim on Windows supports concurrent assertions false

Answer 1:

false

Answer 2:

true

Answer 3:

false

Partial

Question 5

0.67 / 1 pts

One disadvantage of tasks declared within a package is they can't access global signals. true

One thing you can't do in packages is declare custom types. false

The primary reason our ALU example from back in Lec02 was clunky was the **op** signal could not be declared as an enumerated type. false

Answer 1:

true

Answer 2:

false

Answer 3:

false

Question 6

1 / 1 pts

Below is a portion of a verilog model of a ROM with parametizable depth (number of words in the ROM is passed in as a parameter). The word width is fixed at 16-bits.

```
module ROM(clk,addr,dout);
  parameter mem_depth = 8192;

  input clk;
  input [????:0] addr;
  output [15:0] dout;

  ...

endmodule
```

what should ???? should be replaced with so the address is the correct width

- mem_depth - 1
- 2^mem_depth - 1
- 2^mem_depth
- $clog2(mem_depth) - 1
- $clog2(mem_depth/2)
- 12

Question 7

1 / 1 pts

The **primary** issue with the following code is: The signal "sig" being monitored is passed as a value not a pointer

```
task wait4sig(input sig, input int clks2wait);
  fork
    begin: timeout
      repeat(clks2wait) @(posedge clk);
      $display("ERR: timed out waiting for sig in wait4sig");
      $stop();
    end
    begin
      @(posedge sig);        // signal of interest asserted
      disable timeout;
    end
  join
endtask
```

Answer 1:

The signal "sig" being monitored is passed as a value not a pointer

Question 8

1 / 1 pts

For each statement below indicate if it is a property of a **task** or a **function**.

Can be declared as a **void** type in system verilog Functions

Can having timing delays and controls Tasks

Have to have at least one input Functions

Can be recursive Both

Can return multiple outputs Both

Answer 1:

Functions

Answer 2:

Tasks

Answer 3:

Functions

Answer 4:

Both

Answer 5:

Both

Quiz Score: **7.33** out of 8