Course outline

How does an NPTEL online course work?

NPTEL » C-Based VLSI Design

Prerequisite: Week 0

Week1: Introduction to Cbased VLSI Design

Week2: C-Based VLSI Design: Basic Scheduling

Week3: C-Based VLSI Design: List Based

Scheduling Week 4: C-Based VLSI

Design: Advanced Scheduling

Week 5: C-Based VLSI Design: Allocation and

Binding

Week 6: C-Based VLSI Design: Allocation, Binding, Data-path and Controller

Generation

Week 7: C-Based VLSI Design: Efficient Synthesis of C Code

Week 8: C-Based VLSI Design: Hardware Efficient C Coding

Week 9: C-Based VLSI Design: Impact of Compiler

Optimizations in Hardware Week 10: Verification of High-

level Synthesis

- Lec 1: Simulation Based Verification
- Lecture Note for Lec 1
- Lec 2: RTL to C Reverse Engineering

Lecture Note for Lec 2

 Lec 3: Phase-wise Verification of HLS

Lec 4: Equivalence between

- Lecture Note for Lec 3
- C and RTL Lecture Note for Lec 4
- Quiz: Week 10: Assignment 10
- Week 10: Feedback Form Solution: Assignment 10
- Week 11: Securing Design

with High-level Synthesis

Week 12: Introduction to EDA

and Recent Advances in C-Based VLSI Design

Download

Live Sessions

## Week 10: Assignment 10

The due date for submitting this assignment has passed.

Due on 2021-10-06, 23:59 IST. As per our records you have not submitted this assignment.

1) Assume R<sub>c</sub> represents the condition of execution and r<sub>c</sub> represents the data transformation for a computation c in a finite state machines with 1 point datapaths (FSMD). Then, two computations c1 and c2 are equivalent if  $\bigcirc$  R<sub>c1</sub> = R<sub>c2</sub> and r<sub>c1</sub>!= r<sub>c2</sub>

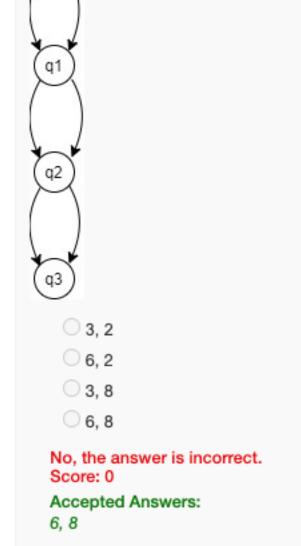
 $\bigcirc$  R<sub>c1</sub> != R<sub>c2</sub> and r<sub>c1</sub> = r<sub>c2</sub>  $\bigcirc$  R<sub>c1</sub> = R<sub>c2</sub> and r<sub>c1</sub> = r<sub>c2</sub> R<sub>c1</sub> != R<sub>c2</sub> and r<sub>c1</sub> != r<sub>c2</sub> No, the answer is incorrect. Score: 0

Accepted Answers:  $R_{c1} = R_{c2}$  and  $r_{c1} = r_{c2}$ 

2) What would be the total number of computations for the FSMD generated from the code snippet given below? 1 point if(c1){ if(c2) t1 = a + b;

t1 = c x d; else{  $t1 = c \times d;$ 01 02 3 04 No, the answer is incorrect. Score: 0

3) What would be the total number of paths and computations for the FSMD M1, respectively? Consider q0 as the reset state 2 points or start state and nodes with more than one outgoing edge as cut-points. (Note: A computation is an execution of the behaviour and a path is between two cut-points without having any intermediate cut-point)



Accepted Answers:

3

always @ (posedge ap\_clk) begin

4) What would be the correct conversion of c code from the Verilog code given below. Consider a, c, x, y and m are registers. 2 points

if (1'b1 == ap\_CS\_fsm\_state2) begin a <= b; c <= d; end end assign b = x + y; assign d = a + m; ap\_CS\_fsm\_state2;  $x_old = x;$  $y_old = y;$ 

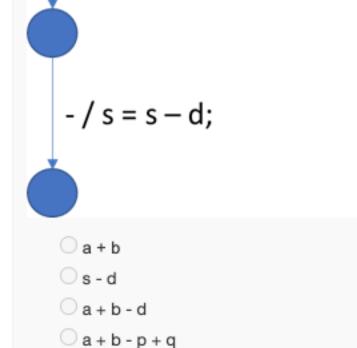
if (1 == ap\_CS\_fsm\_state2)

a = x + y; if (1 == ap\_CS\_fsm\_state2) c = a + m;ap\_CS\_fsm\_state2;  $b_old = b;$  $d_old = d$ ; if (1 == ap\_CS\_fsm\_state2)  $a = b_old;$  $c = d_old;$ if (1 == ap\_CS\_fsm\_state2) a = x + y; c = a + m;ap\_CS\_fsm\_state2;  $a_old = a;$  $x_old = x;$  $y_old = y$ ;  $m_old = m;$  $c_old = c;$ if (1 == ap\_CS\_fsm\_state2)  $a = x_old + y_old;$ if (1 == ap\_CS\_fsm\_state2)  $c = a_old + m_old;$  None of the above No, the answer is incorrect. Score: 0 Accepted Answers: ap\_CS\_fsm\_state2;  $a\_old = a;$  $x_old = x$ ;  $y\_old = y;$  $m_old = m;$  $c\_old = c$ ; if (1 == ap\_CS\_fsm\_state2)  $a = x_old + y_old;$ 

-/s = a + b; d = p - q;

5) Consider the following path. What would be the final symbolic value of s at the end of the path?

1 point



No, the answer is incorrect.

Accepted Answers:

Score: 0

a+b-p+q

 $if (1 == ap\_CS\_fsm\_state2)$ 

 $c = a\_old + m\_old;$ 

S1: Formal verification checks the equivalence of inputs and outputs of HLS symbolically; it does not need any test case.

S2: Simulation based verification of guarantees 100% correctness of the High-level Synthesis.

6) Consider the following two statements. Which one is TRUE about verification of High-level Synthesis?

1 point

1 point

Only S1 is True Only S2 is True

Both S1 and S2 are True Both S1 and S2 are False No, the answer is incorrect.

Score: 0 Accepted Answers: Only S1 is True

7) Reverse engineering an equivalent C code from the HLS generated RTL helps in faster simulation based verification of HLS. 1 point True

False No, the answer is incorrect. Score: 0

Accepted Answers: True

8) Which one of the following statements is correct?

S1. RTL simulation is faster than C simulation. S2. Phase wise verification of HLS needs intermediate information from the HLS tool.

S3. It is not possible to extract a C like behaviour from the RTL generated by the HLS tool.

Only S2 Only S3

None of them No, the answer is incorrect.

Only S1

Only S2

Score: 0 Accepted Answers: