Department of electronic systems

Examination paper for TFE4171 Design of digital systems 2

Examination date: May 23rd, 2020

Examination time (from-to): 9:00 - 13:00

Permitted examination support material: All support material is allowed. Still, please note the information about cheating / plagiarism, below.

Academic contact during examination: Snorre Aunet
Phone: +47 90013264

Technical support during examination: Orakel support services
Phone: 73 59 16 00

OTHER INFORMATION

If a question is unclear/vague – make your own assumptions and specify in your answer the premises you have made. Only contact academic contact in case of errors or insufficiencies in the question set.

Maximum number of points per task and sub-task are given in the text.
Maximum number of points totally: 100.
The final grade is calculated by summing your points from this exam scaled to 60% with the exercises and the term project both scaled to 20% each.
NB: This exam must be passed to pass in total. It is not sufficient that the total grade is a pass grade (E or better), the grade on the exam itself must also be E or better.

Saving: Answers written in Inspera are automatically saved every 15 seconds. If you are working in another program remember to save your answer regularly.

Cheating/Plagiarism: The exam is an individual, independent work. Examination aids are permitted. All submitted answers will be subject to plagiarism control. *Read more about cheating and plagiarism here: https://innsida.ntnu.no/wiki/-/wiki/English/Cheating+on+exams*

Notifications: If there is a need to send a message to the candidates during the exam (e.g. if there is an error in the question set), this will be done by sending a notification in Inspera. A dialogue box will appear. You can re-read the notification by clicking the bell icon in the top right-hand corner of the screen. All candidates will also receive an SMS to ensure that nobody misses out on important information. Please keep your phone available during the exam.

Weighting: How the questions are weigthed is indicated in the exam set.

IN CASE OF FILE UPLOAD USING SKETCHES/CALCULATIONS AND/ORD THIRD-PARTY SOFTWARE

- All files must be uploaded before the examination time expires. 15 minutes are added to the examination time to manage the sketches/calculations/files. (The additional time is included in the remaining examination time shown in the top left-hand corner.)
    - *How to digitize your sketches/calculations*
    - *How to create PDF documents*
    - *Remove personal information from the file(s) you want to upload*

ABOUT SUBMISSION

Your answer will be submitted automatically when the examination time expires and the test closes, if you have answered at least one question. This will happen even if you do not click “Submit and return to dashboard” on the last page of the question set. You can reopen and edit your answer as long as the test is open. If no questions are answered by the time the examination time expires, your answer will not be submitted.

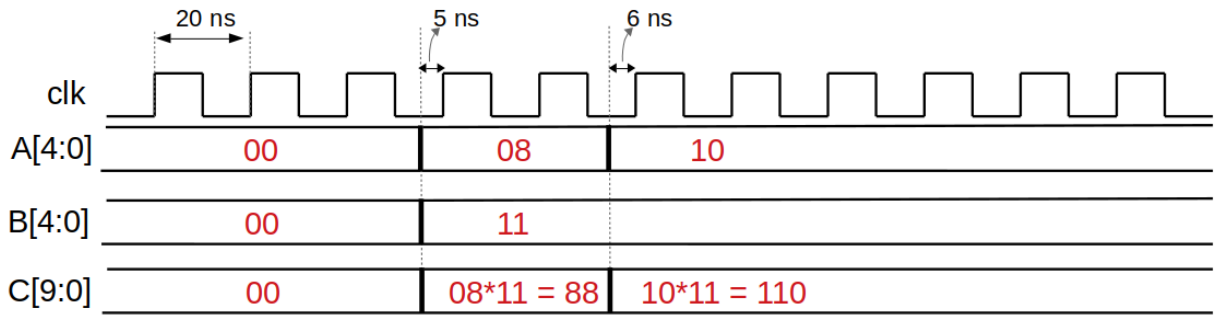
Withdrawing from the exam: If you wish to submit a blank test/withdraw from the exam, go to the menu in the top right-hand corner and click “Submit blank”. This can not be undone, even if the test is still open.

Accessing your answer post-submission: You will find your answer in Archive when the examination time has expired.

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

1- SystemVerilog (15 p)

As a verification engineer you are asked to generate the following waveform using SystemVerilog:



1a) Generate `A`, `B`, and `C` signals using a SystemVerilog task (`clk` is an input port). (12p)

1b) Can you implement it using a SystemVerilog function? Explain your **Yes** or **No** answer. (3p)

2- SystemVerilog Assertions (25 p)

2a) Write a SystemVerilog assertion that checks the following design property in an arbiter with `request`, `ack`, and `grant` signals: (5p)

"When the `request` signal becomes active (high), the `ack` signal becomes active after two or more clock cycles, remains active until it becomes inactive (low) at the same cycle as the `grant` signal gets active".

2b) Assume we have a finite state machine with the following states:

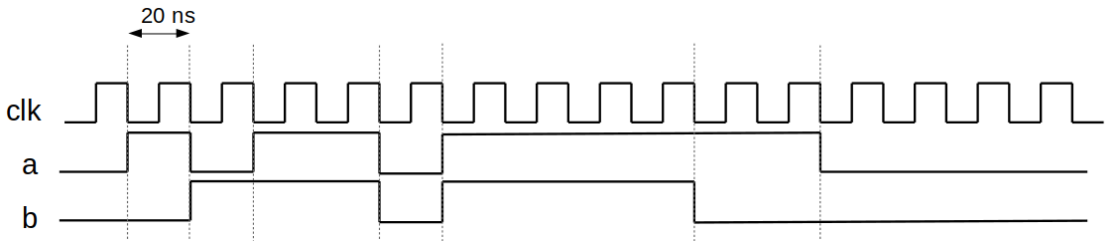
```
Typedef enum {IDLE, WAIT1, WAIT2, EXEC1, EXEC2, EXEC3, ACK}
```

Write SystemVerilog assertions for checking the following design properties:

Property P1: "The state machine eventually goes to `IDLE` state from `EXEC1`, `EXEC2`, `EXEC3`, and `ACK` states". (5p)

Property P2: "Within a specific but parameterized number of clock cycles, transition from states `EXEC1`, `EXEC2`, and `EXEC3` must happen to either `ACK` or `IDLE` state". (5p)

2c) Considering the following waveform, specify on which time points the assertions on the following properties pass or fail (ignore vacuous passes, and assume the first rising edge of `clk` happens at time 0 ns and it has a duty cycle of 50%). (10p)



- property p1; @(posedge clk) a |-> ##[1:4] b; end property;
- property p2; @(posedge clk) a[-> 2] ##1 b[-> 2]; end property;
- property p3; @(posedge clk) a[-> 2] within b[-> 3]; end property;

3- SystemVerilog Coverage (10 p)**:**

Assume we have a finite state machine with the following states:

`{S0, S1, S2, S3, S4, S5, S6, S7, S8}`

This state machine drives two output ports (`CMD` and `ADDRESS[8:0]`) towards a memory block and can perform *read* and *write* operations in the memory. The `CMD` port can have two values `Read` and `Write`.

Considering these, define *coverage points*/*coverage bins* to check if:

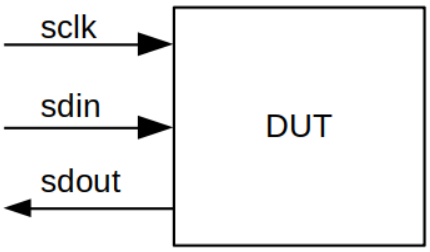
3a) All states are covered. (2p)

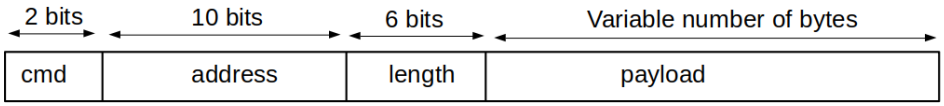
3b) Both `Read` and `Write` commands are tested in states `S2, S3, S4, S5, S6`. (4p)

3c) Both `Read` and `Write` operations are tested on all 512 memory entries. (4p)

4-UVM (10 p)**:**

A Design Under Test (DUT) with its three single-bit ports (`sclk, sdin,` and `sdout`), its serial communication packet format, as well as some constraints are shown in the following figures:





Packet Format

Constraints:

- Only `address[7:0]` bits are valid for the DUT
- `cmd[0]` is always '1'

You as the verification engineer are assigned to verify the DUT using UVM.

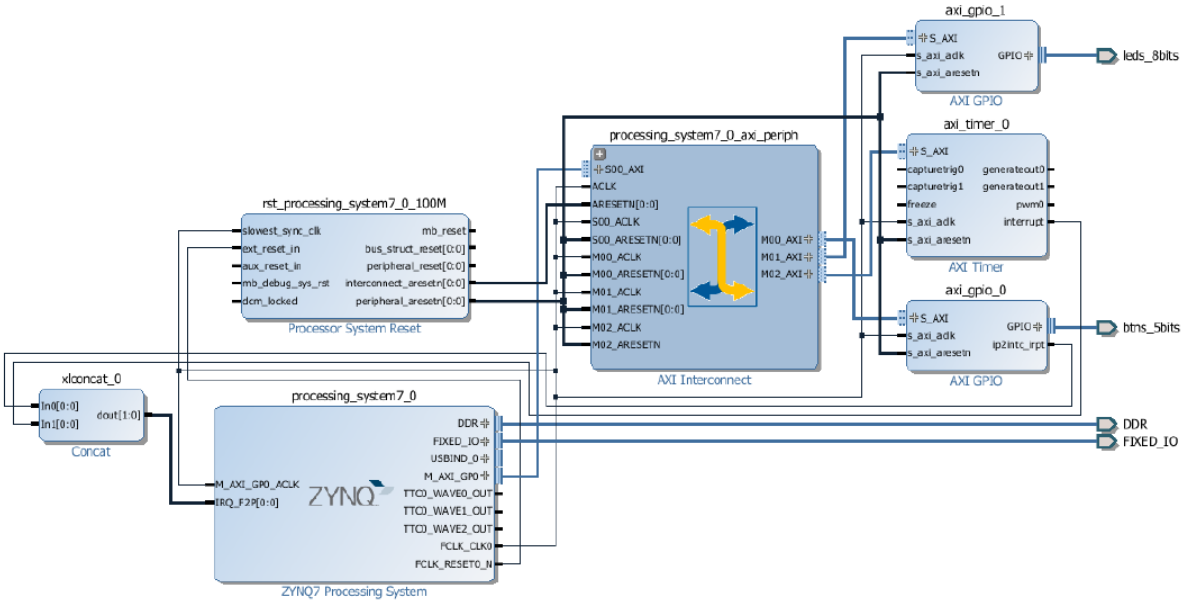
4a) Define a transaction class (sequence item class) that represents/encapsulates the packet, considering the provided constraints. The objects of this class can be used to generate random packets (with random `cmd`, `address`, `length`, and `payload`). (5p)

4b) Developing an *agent* for the communication protocol increases the reusability of the verification logic. Without showing/describing the details of implementation, by using a block-diagram show how the components inside an ***active*** agent for this protocol, test sequences, scoreboard, test environment, virtual interfaces, testbench, and the DUT are connected. (5p)

5 – SoC (20 p)

5a) What is SoC? Describe SoC components and present a block diagram of a common Soc structure with functional components and interfaces. (2p)

5b) A simple Xilinx SoC system is presented in the figure below. Describe functional components and communication interfaces in this system. (3p)



5c) Describe classification of hardware IP cores. Elaborate flexibility-performance trade-off for each type of the IP core. (3p)

5d) Describe each phase of FPGA-based IP core life cycle process. (6p)

5e) Describe the phases of a SoC codesign flow. (4p)

5f) What are the trade-offs to consider when performing functional partition on a SoC system? (2p)

6 – Formal verification (20 p)

6a) (6 p) Describe the role of invariants in Interval Property Checking (IPC). What characterizes a "good" invariant and what a "bad" invariant when proving a property? Will the property always fail if the chosen invariant covers unreachable states?

6b) (8 p) A property checker can produce a *counterexample* or a *witness* for a given property. Please, define these terms precisely.

Suppose you are running a property on a formal property checker.

I) Is it possible that for the same property, both a counterexample and a witness exist? Explain your answer.

II) Is it possible that, for a given property, there exists no counterexample? Explain your answer.

III) Is it possible that a property holds on a design but the property checker returns a counterexample?

6c) (6 p) Without repeating one of the examples from the class, describe in general words the nature of hardware security vulnerabilities by timing-based microarchitectural side channels. What is new in Spectre and Meltdown style attacks?