Technische Universiteit Eindhoven University of Technology

Student name:

Student number:

Examination cover sheet

(to be completed by the examiner)

Course name: Hardware Verification — Course code: 2IMF20

Date: 31-10-2017

Start time: 09:00 — End time : 12:00

Number of pages: 4

Number of questions: 6

Maximum number of points/distribution of points over questions:100

Method of determining final grade: divide total of points by 10

Answering style: open questions

Exam inspection: With your instructor

Other remarks: It is not allowed to use study materials or a computer during the exam.

Instructions for students and invigilators

Permitted examination aids (to be supplied by students):

- ☐ Notebook
- ☐ Calculator
- ☐ Graphic calculator
- ☐ Lecture notes/book
- ☐ One A4 sheet of annotations
- ☐ Dictionar(y)(ies). If yes, please specify:
- ☐ Other:

Important:

- examinees are only permitted to visit the toilets under supervision
- it is not permitted to leave the examination room within 15 minutes of the start and within the final 15 minutes of the examination, unless stated otherwise
- examination scripts (fully completed examination paper, stating name, student number, etc.) must always be handed in
- the house rules must be observed during the examination
- the instructions of examiners and invigilators must be followed
- no pencil cases are permitted on desks
- examinees are not permitted to share examination aids or lend them to each other

During written examinations, the following actions will in any case be deemed to constitute fraud or attempted fraud:

- using another person's proof of identity/campus card (student identity card)
- having a mobile telephone or any other type of media-carrying device on your desk or in your clothes
- using, or attempting to use, unauthorized resources and aids, such as the internet, a mobile telephone, etc.
- using a clicker that does not belong to you
- having any paper at hand other than that provided by TU/e, unless stated otherwise
- visiting the toilet (or going outside) without permission or supervision

TECHNISCHE UNIVERSITEIT EINDHOVEN
Department of Mathematics and Computer Science

Examination Hardware Verification (2IMF20)
Tuesday, October 31st, 2017, 09h00 – 12h00.

Your answers should be formulated and written down clearly. Answers must be written in English. First read **ALL** questions once! Good luck.

Context: A simple power control protocol

In this examination, we will consider a simple controller that is handling requests to turn the power on or off.

Let name `cst` the state of the controller, which is composed of three bits:

- `cst`[2] (request): a transition of low to high represents a request to turn the power on, a transition from high to low represents a request to turn the power off;
- `cst`[1] (accept): a transition from low to high means that the controller accepts a request to turn the power on, a transition from high to low means that the controller accepts a request to turn the power off;
- `cst`[0] (deny): a transition from low to high means that the controller denies a request to turn the power off.

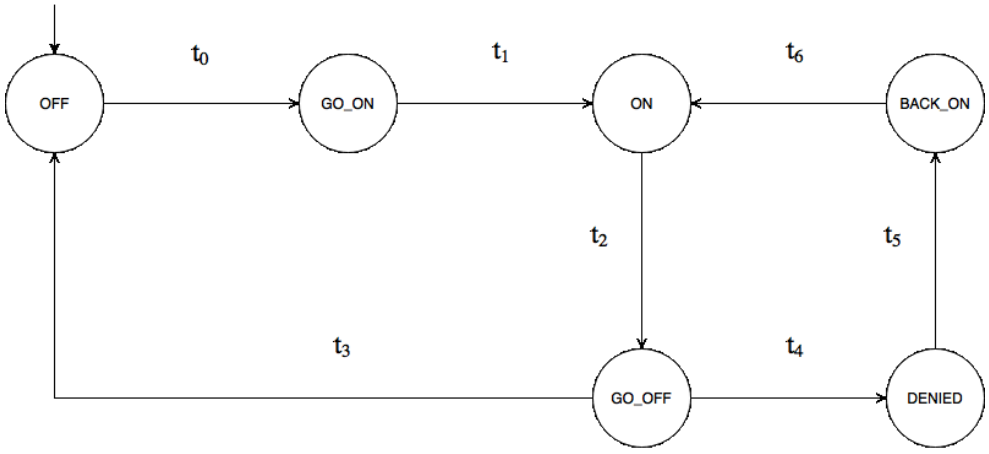

Figure 1: A simple controller for power management.

Figure 1 shows the possible state transitions. Each transition is numbered from t_0 to t_6. The encoding of each state – `cst[2], cst[1],cst[0]` – is as follows:

- `000`: OFF;

- `100`: GO_ON;
- `110`: ON;
- `010`: GO_OFF;
- `011`: DENIED;
- `111`: BACK_ON.

Note that the protocol is such that when the controller denies a request to turn the power off, it must receive a request to turn the power on to move back to state "ON".

Linear Time Logic (LTL)

1. (**15 points**) Consider the protocol described in Figure 1. Write one or more **LTL** formulas expressing the requirements below. For each requirement, state whether it is satisfied or not satisfied by the transition system pictured in Figure 1. When a requirement is not satisfied, give a counter-example.

   (a) The controller shall always accept a request to turn the power on.

   (b) The controller shall never deny a request without having accepted at least one request before. (hint: a request has been accepted before when bit `cst[1]` is high.)

   (c) The controller shall reach state "ON" (`110`) infinitely often.

   (d) When in the initial state, the controller shall always go through the following sequence of states: 000 ("OFF"), 100 ("GO_ON"), 110 ("ON") with some possible delay between the transitions.

System Verilog Assertions (SVA) and Property Specification Language (PSL)

2. (**10 points**) Write assertions written in either System Verilog or PSL for the properties of the previous question.

Computation Tree Logic (CTL)

3. (**15 points**) Consider the protocol described in Figure 1. Write one or more **CTL** formulas expressing the requirements below. For each requirement, state whether it is satisfied or not satisfied by the transition system pictured in Figure 1. When a requirement is not satisfied, give a counter-example.

   (a) The controller shall possibly deny a request to turn the power on.

(b) The controller shall always possibly reach state "ON" (`110`).

(c) The controller shall always reach state "ON" (`110`) after two steps from the initial state.

(d) When a request is denied, the controller shall always eventually return to state "ON" (`110`).

Binary Decision Diagrams (BDD)

4. (**15 points**) Consider the protocol described in Figure 1. Define the required variables of your ROBDD's and choose (carefully) an ordering for these variables. Draw for your chosen ordering the ROBDD's representing the following:

   (a) the controller is in state "GO_OFF" (`010`);

   (b) transitions t_3 and t_4 that are possible from state "GO_OFF" (`010`);

   (c) the set of the states reachable from state "GO_OFF" (`010`).

5. In the previous question, the third ROBDD is the result of doing forward reachability using the first two ROBDD's. Mention the two operations performed to compute the third ROBDD from the first two. Explain briefly for each operation the derivation of the result from the inputs.

Bounded Model Checking (BMC)

6. (**40 points**) Consider the protocol pictured in Figure 1. Consider the following properties:

   (a) $P = \mathbf{G}\neg cst[3]$

   (b) $Q = \mathbf{F}(cst == 110)$

   Property P expresses the fact that a request is never denied. Property Q expresses that eventually state "ON" is reached. In the following questions you will use Bounded Model Checking to prove or disprove these two properties.

   (a) Write down the transition relation as a Boolean function $T(c_2, c_1, c_0, c'_2, c'_1, c'_0)$, where c_i denotes the value of `cst[i]` before a transition and c'_i denotes the value of `cst[i]` after a transition. This is the usual "prime" notation.

   (b) Define predicate $I(c_2, c_1, c_0)$ that returns true if and only the values of c_i represent an initial state.

   (c) We will attempt BMC at a depth of 4, that is, the length k of paths is $k = 4$. We define variables representing each state of such paths. That is, we define $s_0 = c_2^0 c_1^0 c_0^0$, $s_1 = c_2^1 c_1^1 c_0^1$, $s_2 = c_2^2 c_1^2 c_0^2$, $s_3 = c_2^3 c_1^3 c_0^3$, and $s_4 = c_2^4 c_1^4 c_0^4$. Write the Boolean formula encoding the validity of paths.

   (d) Some paths have a loop. Some paths do not have a loop. Let $\mathbf{L}_4$ be the Boolean formula encoding the existence of a loop in paths of length 4. Write down this formula.

(e) Now that you have defined the encoding for valid paths and the existence of a loop, what is the property that paths must satisfy? Write down this property in LTL for property P and for property Q.

(f) Write down the encoding of the property for P for paths with a loop.

(g) Write down the encoding of the property for P for paths without a loop.

(h) Write down the encoding of the property for Q for paths with a loop.

(i) Write down the encoding of the property for Q for paths without a loop.

(j) Now you have all the pieces to build the global formula encoding the satisfiability of the LTL formulas P and Q. Write down the resulting property for P. Write down this property for Q.

(k) After simplifying as much as possible, write down the two CNF formulas that would be submitted to a SAT solver.

(l) What can you conclude about the validity of formula P? What about the validity of formula Q?