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: 01-11-2016

Start time: 13:30 — End time : 16:30

Number of pages: 5

Number of questions: 5

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, November 1, 2016, 13h30 – 16h30.

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

Linear Time Logic (LTL)

1. (**20 points**) Consider the arbiter block[1] in Figure 1.

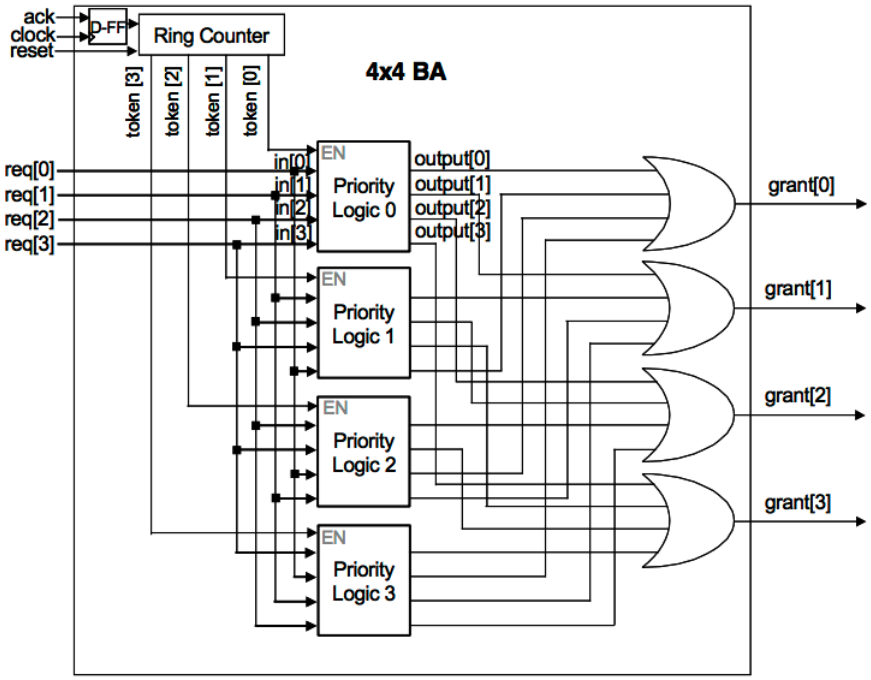

Figure 1: An arbiter block.

This arbiter consists in a D flip-flop, priority logic blocks, a 4-bit ring counter, and four 4-input OR gates. Each priority logic block is implemented in combinatorial logic according to the truth table shown in Figure 2. Inputs are given priority in ascending order, that is, `in[0]` has the highest priority, `in[1]` has the next priority, and so on. The Ring Counter is a 4-bit token ring. There are four tokens. Each token enables exactly one priority logic block. This ensures that at most one request is granted. The token is shifted to the left each cycle where `ack` is high. This ensures absence of starvation. Each input is eventually granted access to the resource controlled by the arbiter. The token

[1] Taken from "Round-robin Arbiter Design and Generation" by Shin et al. ISSS'02.

is initialised to `4'b0001`. The grant signals indicate which agent is granted access. For instance, if `grant[0]` is high, then agent 0 (with `req[0]`) is granted access.

EN	in [0]	in [1]	in [2]	in [3]	output [0]	output [1]	output [2]	output [3]
0	X	X	X	X	0	0	0	0
1	1	X	X	X	1	0	0	0
1	0	1	X	X	0	1	0	0
1	0	0	1	X	0	0	1	0
1	0	0	0	1	0	0	0	1

Figure 2: Truth table of priority logic blocks.

Write LTL formula's expressing the following requirements:

(a) Mutual exclusion of the grant signals. At any time, at most one grant signal is high.

(b) Absence of starvation. All requests are eventually granted.

(c) There is never a grant signal without a request from the corresponding agent, for instance, we never have `grant[1]` without having `req[1]`.

(d) Environment assumptions: consider signals `ack, clock, reset`. Write LTL formulas characterising the required assumptions on these signals to prove the two properties above.

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

2. (**15 points**) Write assertions written in either System Verilog or PSL for the properties and assumptions of the previous question.

Computation Tree Logic (CTL)

3. (**15 points**) Consider the T-junction shown in Figure 3. It consists of three traffic lights: two for the major road and one for the minor road. Each light goes the usual sequence of red, yellow, green, and back to red. You can assume the two lights of the major road to be fully synchronised and therefore abstract them away to a single light. The minor road has a sensor detecting the presence of a car. When a car is detected, the traffic light will go green to let the car enter the major road and then go back to red. Present a set of atomic propositions – try to minimise the number of propositions – that are needed to describe the following properties of the traffic light controller of the T-junction as CTL formulas and give the corresponding CTL formulas:

(a) The lights of the major and minor roads shall never be green at the same time.

(b) The major road traffic light shall eventually become green.

(c) When a car is detected, the minor road light shall eventually become green.

(d) Lights shall follow the sequence red, green, yellow, red.

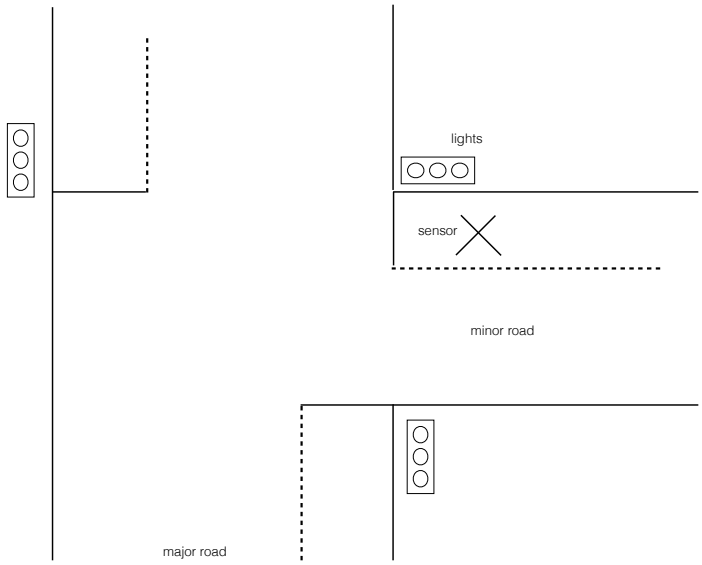

Figure 3: A T-junction.

Binary Decision Diagrams (BDD)

4. (**10 points**) Consider the truth table in Figure 2 again. We only consider signals `in[0]` to `in[3]` and `output[0]` to `output[3]`. Order these variables – think well about how to select this ordering – and draw a ROBDD representing the Boolean function described by the truth table in Figure 2.

Bounded Model Checking (BMC)

5. (**40 points**) Consider the Ring Counter in Figure 1. The state diagram in Figure 4 shows the state of this component where the state variable corresponds to the possible values of vector `token` in the figure. As it can be seen, this vector starts with `token[0]` high and then shift the '1' to the left. Consider the following properties:

   (a) $P = \mathbf{G}\neg(\neg\texttt{token}[3] \wedge \neg\texttt{token}[2] \wedge \neg\texttt{token}[1] \wedge \neg\texttt{token}[0])$

   (b) $Q = \mathbf{F}\ \texttt{token}[3]$

   Property P expresses the fact that there is always one token active. Property Q expresses the fact that eventually the fourth token is active. In the following questions you will use Bounded Model Checking to prove or disprove these two properties.

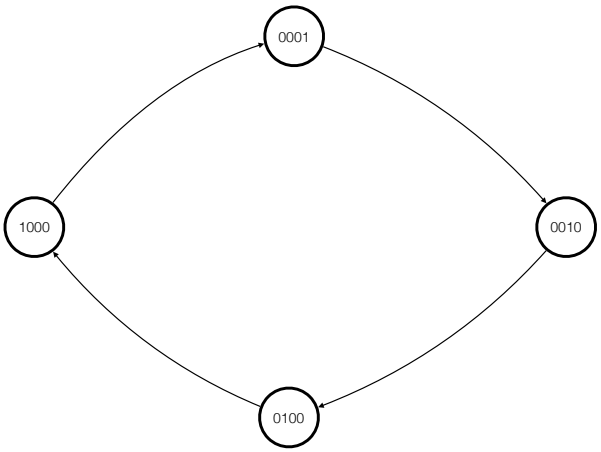

Figure 4: Transition diagram for Ring Counter.

(a) Write down the transition relation as a Boolean function $T(t_3, t_2, t_1, t_0, t'_3, t'_2, t'_1, t'_0)$, where t_i denotes the value of `token[i]` before a transition and t'_i denotes the value of `token[i]` after a transition. This is the usual "prime" notation.

(b) Define predicate $I(t_3, t_2, t_1, t_0)$ that returns true if and only the values of t_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 three variables representing each state of such paths. That is, we define $s_0 = t_3^0 t_2^0 t_1^0 t_0^0$, $s_1 = t_3^1 t_2^1 t_1^1 t_0^1$, $s_2 = t_3^2 t_2^2 t_1^2 t_0^2$, $s_3 = t_3^3 t_2^3 t_1^3 t_0^3$, and $s_4 = t_3^4 t_2^4 t_1^4 t_0^4$. 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?