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: 27-10-2015

Start time: 13:30 | End time : 16:30

Number of pages: 5

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)
Monday, October 27, 2015, 13h30 – 16h30.

Your answers should be formulated and written down clearly. First read **ALL** questions once!

Linear Time Logic (LTL)

1. (**15 points**) Suppose we have two users, *Peter* and *Betsy*, and a single printer device *Printer*. Both users perform several tasks, and every now and then they want to print their results on the *Printer*. Since there is only a single printer, only one user can print a job at a time. Suppose we have the following atomic propositions for *Peter* at our disposal:

   - *Peter.request*: indicates that *Peter* has requested usage of the printer;
   - *Peter.use:* indicates that *Peter* is using the printer;

   For *Betsy*, similar predicates are defined. Specify in LTL the following properties:

   (a) Mutual exclusion, that is, only one user at a time can use the printer.

   (b) Finite time of usage, that is, a user can print only for a finite amount of time.

   (c) Absence of individual starvation, that is, if a user requests to print something, he/she eventually is able to do so.

   (d) Alternating access, that is, users must strictly alternate in printing.

Computation Tree Logic (CTL)

2. (**20 points**) Consider an elevator system that services $N > 0$ floors numbered 0 through $N - 1$. There is an elevator door at each floor with a call button and an indicator light that signals whether or not the elevator has been called. In the elevator cabin there are N send buttons (one per floor) and N indicator lights that inform to which floor(s) is going to be sent. For simplicity consider $N = 4$. Present a set of atomic propositions – try to minimise the number of propositions – that are needed to describe the following properties of the elevator system as CTL formulas and give the corresponding CTL formulas:

   (a) The doors are "safe", that is, a floor door is never open if the cabin is not present at the given floor.

(b) The indicator lights correctly reflect the current requests. That is, each time a button is pressed, there is a corresponding request that needs to be memorised until completion (if ever).

(c) The elevator only services the requested floors and does not move when there is no request.

(d) All requests are eventually satisfied.

Binary Decision Diagrams and Combinatorial Equivalence

3. (**10 points**) Using the ordering $x_1 < x_2 < x_3 < x_4 < x_5$, construct the ROBDD for the majority function:

$$\mathbf{MAJ}(x_1, x_2, ..., x_6) \equiv (x_1 + x_2 + x_3 + x_4 + x_5) \geq 3$$

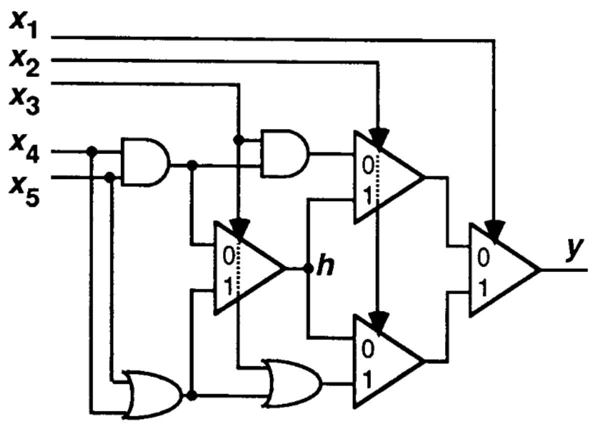

Figure 1: Realisation of a 5-bit majority function.

4. (**15 points**) Figure 1 shows the realisation of the majority function. Note that triangles are 2-input multiplexers. Prove using ROBDDs that this circuit is equivalent to the majority function defined in the previous question.

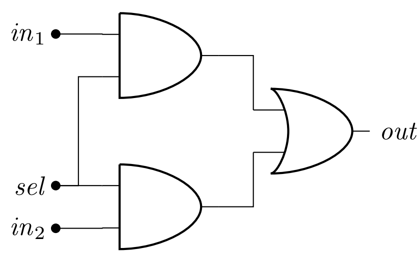

Figure 2: Realisation of a two input multiplexer

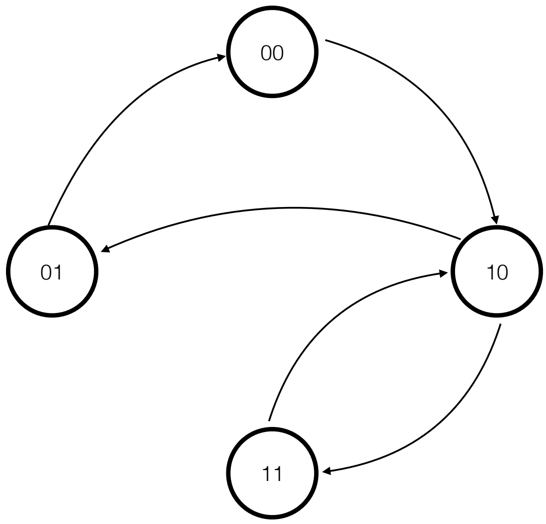

Figure 3: A transition system.

And-Inverter Graphs

5. (**10 points**)Draw an AIG (And-Inverter-Graph) representing the 2-input multiplexer shown in Figure 2.

Bounded Model Checking

6. (**30 points**)Assume a system composed of threads. A thread can be in four different states:

   - `idle`: the thread is inactive.
   - `running`: the thread is active.
   - `suspended`: the thread has been suspended.
   - `halted`: the thread has been stopped.

   Figure 3 shows the transitions between these states assuming the following encoding. Each state is encoded as a sequence of two bits, noted a and b. State `idle` is encoded by the sequence 00, state `running` as the sequence 10, state `suspended` as 11, and state `halted` as 01. So, a thread starts in the `idle` state and makes a transition to the `running` state. Then it can either go to the state `suspended` or the state `halted`. Once `suspended` a thread goes back to the `running` state. Once `halted` a thread goes to the `idle` state. Consider the property that a thread is eventually halted, that is, it cannot run forever. Formally, this is expressed in LTL as the formula **F**(`halted`) or using the state encoding $\mathbf{F}(\neg a \wedge b)$. In the following questions you will use Bounded Model Checking to prove or disprove this property.

(a) Write down the transition relation as a Boolean function $T(a, b, a', b')$, where a and b denotes the values of the variables in the current state and a' and b' denote the values of the variables after a transition. This is the usual "prime" notation.

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

(c) We will attempt BMC at a depth of 2, that is, the length k of paths is $k = 2$. We define three variables representing each state of such paths. That is, we define $s_0 = a_0b_0$, $s_1 = a_1b_1$, and $s_2 = a_2b_2$. Write the Boolean formula encoding the validity of paths.

(d) Some paths have a loop. Some paths do not have a loop. Let $\mathbf{L}_2$ be the Boolean formula encoding the existence of a loop in paths of length 2. 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.

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

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

(h) Now you have all the pieces to build the global formula encoding the satisfiability of the LTL formula $\mathbf{F}(\neg a \wedge b)$. Write down the resulting property.

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

(j) What can you conclude about the validity of the formula $\mathbf{F}(\neg a \wedge b)$?