Ensimag
3rd year – option ISI
February 2020

Exam

Modeling and Verification of Concurrent and Real-time Systems

Duration: 2 hours
All documents allowed

Warning: You are encouraged to be as careful as possible in writing. You will be judged more on the QUALITY than the QUANTITY of your answers.

The score given for each question is purely informative.

Part I Modeling in LNT (10 points)

We consider the modeling of asynchronous circuits, which are built by interconnecting basic logical elements that are not governed by a global clock (we use the term "element" instead of "gate" to avoid the confusion with LNT gates). The basic elements considered here, shown on Figure 1, are the Boolean operations OR, AND, and NAND.

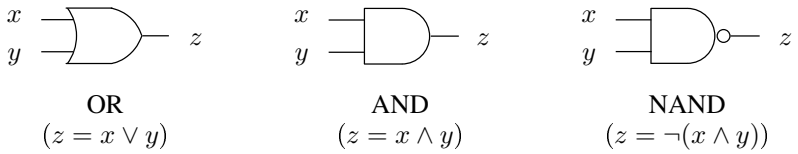


Figure 1: Basic logical elements

Each element is represented by an LNT process whose inputs and output are represented as gates carrying Boolean values. Thus, each LNT gate has the channel type LINK, defined below:

```
channel LINK is
    (Bool)
end channel
```

Elements are assumed to function asynchronously, in an event-driven way. The LNT process modeling an element starts by initializing its input values (given by the data parameters of the process), and then continues with the following cyclic behaviour: (i) it accepts a Boolean value V on one of its input gates; (ii) if V does not change the current value of the output, no output is produced; (iii) otherwise, the new value of the output is emitted on the output gate.

For example, an LNT process modeling the AND element whose input values are both initialized to false, would accept an input action "X (true)", which does not produce any output since it does not change the current value of the output, and then an input action "Y (true)", which will trigger an output action "Z (true)".

Question I.1 Basic logical elements

Following the informal description above, complete the definition of the process below modeling the NAND element. The data parameters X1 and X2 serve respectively to initialize the values received on the two inputs. The local variable RESULT represents the current output value, which must be updated and emitted on the OUTPUT gate every time this value is changed by the last input received. Note that parameters X1, X2 are declared as "**in var**", and therefore they can be assigned new values in the body of the process.

```
process NAND [INPUT1, INPUT2, OUTPUT:LINK] (in var X1, X2:Bool) is
    var RESULT:Bool in
        RESULT := not (X1 and X2);
        -- portion to complete (about 12 lines)
    end var
end process
```

Question I.2 Muller's C element

We consider Muller's C element, an asynchronous circuit having two inputs A, B and an output C. Every time the last values received on inputs A and B are the same, the circuit emits this value on output C; otherwise, it remains silent. An implementation of the C element using basic logical elements is shown on Figure 2, where only gates A, B, C are observable and gates P1, ..., P4 are used for internal connections.

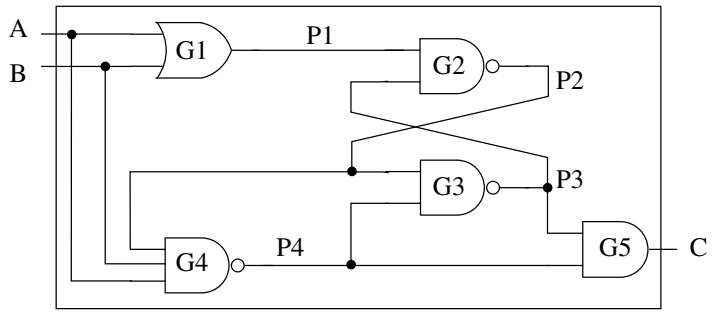


Figure 2: Mayevski's implementation of Muller's C element

We assume that the basic elements G1, ..., G5 operating in parallel are already available as LNT processes parameterized by their initial input values, for example as process G1 below (note that G4 is a NAND element with three inputs):

```
process G1 [INPUT1, INPUT2, OUTPUT:LINK] (in var X1, X2:Bool) is ... end process
```

Following the scheme given in Figure 2, complete the definition of process MULLER below.

```
process MULLER [A, B, C:LINK] (in XA, XB, XP1, XP2, XP3, XP4:Bool) is
    -- portion to complete (about 12 lines)
end process
```

Subsidiary question: knowing that inputs A, B are initially false (i.e., the parameters XA, XB must be set to false when invoking process MULLER), infer the values of the other parameters XP1, ..., XP4 representing the initial values of the inputs P1, ..., P4.

Question I.3 Synchronous product

We consider here the two NAND elements G2 and G3 used in the implementation of Muller's C element. Fragments of the corresponding Labeled Transition Systems (LTSs) are shown on Figure 3.

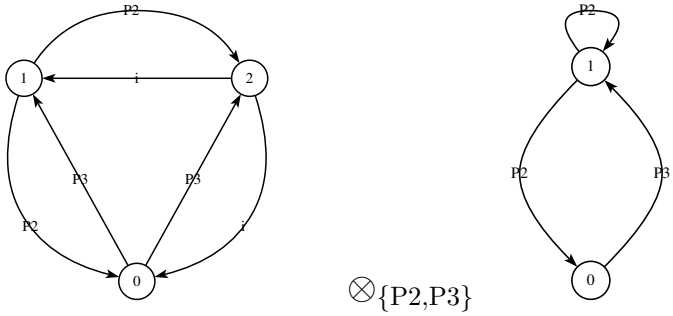


Figure 3: Simplified excerpts of the LTSs of the G2 (left) and G3 (right) elements

Draw the LTS of the synchronous product of the two LTSs in Figure 3, synchronized on gates P2 and P3.

Question I.4 Environment of the circuit

To ensure a proper functioning of Muller's C element in an asynchronous setting, the inputs A and B are allowed to change their value at most once between two consecutive outputs C. This constraint is ensured in LNT by an environment process ENV interacting with the MULLER process on gates A, B, C, and having the same value parameters as MULLER (initial values false for both inputs A and B).

The ENV process has a cyclic behaviour consisting of two steps: (i) it accepts, in parallel, inputs on gates A and B until both of them change their current value; (ii) then, it accepts an output C carrying any value, and restarts its cycle.

Complete the definition of process ENV below.

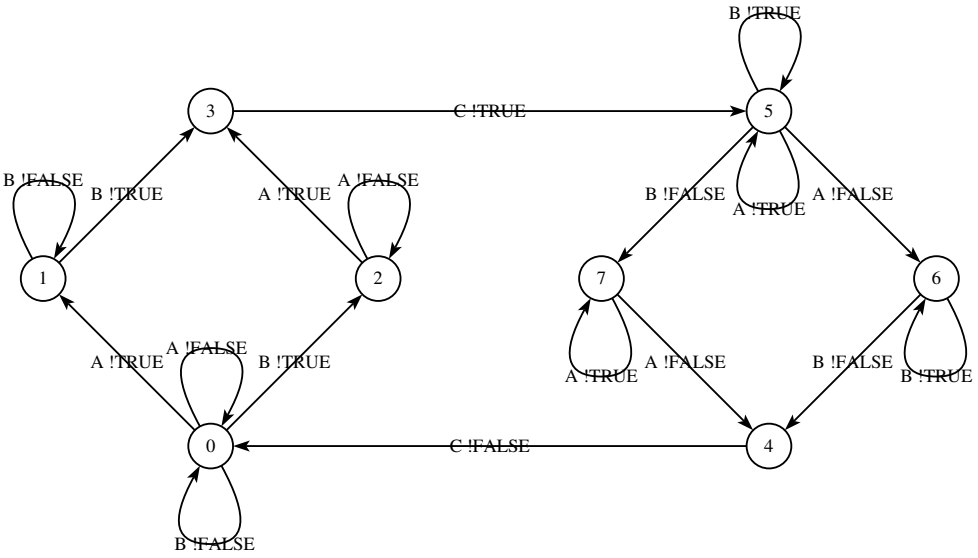
```
process ENV [A, B, C:LINK] (in var XA, XB:Bool) is
    loop
        -- portion to complete (about 20-22 lines)
    end loop
end process
```

Part II Timed automata (5 points)

TODO

Part III Temporal logic (5 points)

Consider the LTS illustrated on the figure below (the initial state has number 0), which represents the behaviour of the C element placed in its environment.



We define the following action predicates:

$$A = \text{“A !TRUE”} \vee \text{“A !FALSE”}$$
$$B = \text{“B !TRUE”} \vee \text{“B !FALSE”}$$

For each of the temporal logic formulas below, indicate the numbers of the states satisfying the formula.

1. $\langle\, \text{true} \,\rangle\, \text{true}$
2. $[\,\text{“C !TRUE”}\,]\, \text{false}$
3. $\langle\, \text{“A !TRUE”} \vee \text{“B !TRUE”} \,\rangle\, \text{false}$
4. $[\,\text{“A !TRUE” . “B !TRUE” . “C !TRUE”}\,]\, \text{false}$
5. $\langle\, (A \vee B)^* \text{ . “C !TRUE”} \,\rangle\, \text{true}$
6. $\mu X.\langle\, \text{true} \,\rangle\, \text{true} \wedge [\,\neg \text{“C !TRUE”}\,]\, X$
7. $[\,(\neg \text{“C !TRUE”})^*\,]\, \langle\, \text{true}^* \text{ . “C !TRUE”} \,\rangle\, \text{true}$
8. $\nu X.\langle\, A \,\rangle\, X$
9. $\mu X.\langle\, \text{“B !TRUE”} \,\rangle\, \text{false} \vee \langle\, \text{true} \,\rangle\, X$
10. $\nu X.\langle\, \text{“A !TRUE” . “B !TRUE”} \,\rangle\, X$

Subsidiary question: which of the above formulas have the same interpretation on any LTS?