CS628 Spring 2003 End-term Examination

Time: 3 hours Total marks: 85

- ***The exam is open book and notes.***
- ***Results/proofs covered in class/problem sessions/assignments may simply be cited, unless specifically asked for.***
- ***Unnecessarily lengthy solutions will be penalized.***
- ***If you need to make any assumptions, state them clearly.***
- ***Do not copy solutions from others or indulge in unfair means.***

1. *[10+10+5+10 marks]* We wish to design the three-input, two-output system shown in Fig. 1a. The behavior

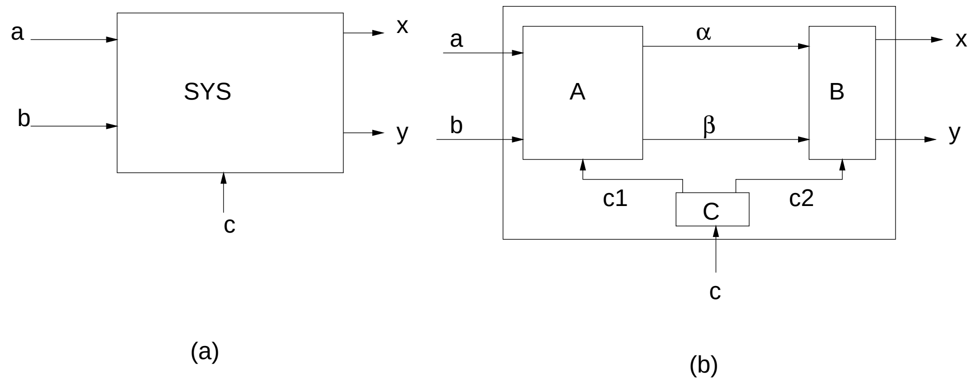

Figure 1:

of this system is given by the following process algebraic description:

$$\begin{aligned}\mathbf{SYS} \;&=\; ((a? \rightarrow x! \rightarrow c? \rightarrow x! \rightarrow c? \rightarrow y! \rightarrow c? \rightarrow y!)\\ &\quad [\,](b? \rightarrow y! \rightarrow c? \rightarrow y! \rightarrow c? \rightarrow x! \rightarrow c? \rightarrow x!)) \rightarrow \mathbf{SYS}\end{aligned}$$

Here, a? denotes an input event (change of value) on a, x! denotes an output event (change of value) on x, and similarly for others.

After several hours of deliberation, it has been decided to implement the system using three blocks as shown in Fig. 1b. Of these, block A has already been designed and its interface behavior is given by the following description:

$$\begin{aligned}\mathbf{A} \;&=\; ((a? \rightarrow \alpha! \rightarrow c? \rightarrow \alpha!)\\ &\quad [\,](b? \rightarrow \beta! \rightarrow c? \rightarrow \beta!)) \rightarrow \mathbf{A}\end{aligned}$$

You are required to come up with specifications and implementations of blocks **B** and **C**.

(a) Give process algebraic descriptions of **B** and **C**, which when composed with **A** as shown in Fig. 1b gives **SYS**.

(b) Derive a set of Production Rules for process **C**.
You must start with the weakest guards for each output transition, and then strengthen them as necessary. Starting with the strongest guards will incur marks penalty.

(c) Draw an STD for process **C**, labeling state transitions with rising and falling transitions on signals.

(d) In the above STD of the above STG, are there any CSC conflict(s)? If not, explain why not. If present, indicate where you would insert state signal transitions in the STG to remove the CSC conflict(s), without imposing any additional timing constraints on the environment.

2. *[10 + 10 + 10 marks]* The asynchronous system shown in Fig. 2 has the following behaviour:

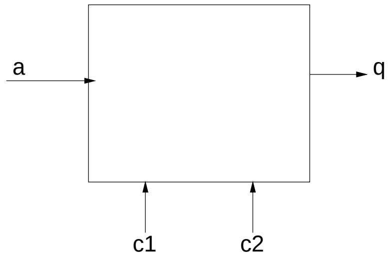

Figure 2:

- Input a is sampled by every alternate rising edge of input c_1.
- Input a is also sampled by every rising edge of input c_2.
- Output q is always the logical OR of the values sampled by c_1 and c_2. Thus, q potentially changes on every alternate rising edge of c_1 and on every rising edge of c_2.
- The environment ensures that c_1 and c_2 are never simulatenously high.

(a) Give an XBM state transition diagram of the system.

(b) Suppose we implement this machine using an interconnection of logic gates, and then derive a Kripke structure out of it. Using the set of atomic propositions $AP = \{c_1, c_2, a, q\}$, express the following requirement in LTL:
If a is sampled high by c_2, then q must become high before c_2 falls.
You may assume that once a value is sampled by a rising edge of a clock, the environment does not lower the clock until the corresponding output change (if any) has occurred.

(c) Suppose we feed the output q of the above system to a toggle flip-flop. A toggle flip-flop has input i and outputs o_1 and o_2, and its behaviour is given by:

$$\mathbf{TOG} = i? \ \rightarrow \ o_1! \ \rightarrow \ i? \ \rightarrow \ o_2 \ \rightarrow \ \mathbf{TOG}$$

Describe the behavior of the resulting system (inputs c_1, c_2, a and outputs o_1, o_2) using an XBM state transition diagram.

3. *[10 + 10 marks]* Consider a two-stage (two C-elements) micropipeline, as defined by Sutherland. Let the inputs to the micropipeline be i_1 and i_2, and let its outputs be o_1 and o_2.

(a) Describe the interface behaviour of the micropipeline using an XBM state transition diagram.

(b) Let the set of atomic propositions be $\{i_1, i_2, o_1, o_2\}$. Describe in LTL the following property:
If two consecutive requests are not acknowledged by the receiver, the sender cannot send a third request.