CS 628 : Spring 2002 End-term Examination

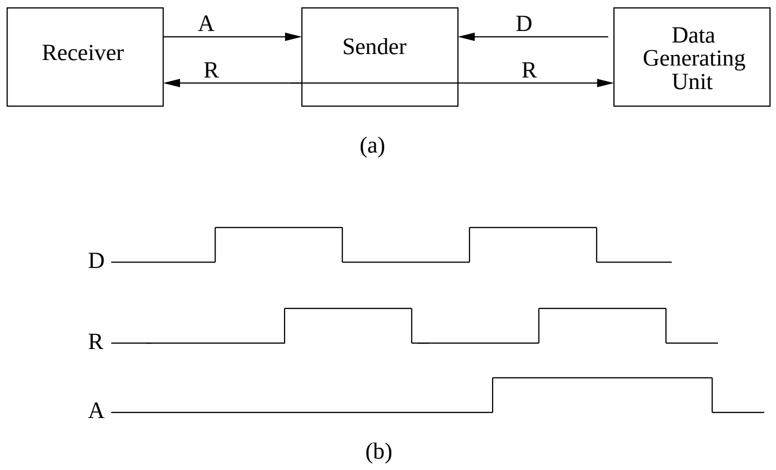

Figure 1:

1. *[10+10 Marks]* Consider a sustem of a data generating unit, a sender and a receiver, as shown in Fig. 1(a). The communication protocol between the sender and receiver is a "3-phase" handshake, described as follows. In the following description, an event on a signal signifies a change of value of the signal (from 1 to 0 or vice versa). The system starts with all signals at value 0. The sender first receives an event on input D signifying that data is ready to be sent to the receiver. It then generates an event on output R, which is seen by both the receiver and the data generating unit. The data generating unit next produces another event on D signifying that the next data is ready to be sent to the receiver. The sender generates the next event on R in response. The receiver now acknowledges by generating an event on A, and the handshake between the sender and the receiver completes. Even if the data generating unit generates the next event on D before the receiver generates the acknowledgement event on A, the sender must not generate the next event on R before it sees the acknowledgement event on A. A timing diagram showing the behavior of the signals is shown in Fig. 1(b).

   (a) Describe the interface behavior of the sender controller using a STG, and derive a K-map implementation of all output and (if needed) state signals.

   (b) Describe the interface behavior of the receiver controller using a Process Algebraic notation, and derive Production Rules (symmetrization not needed) for A and (if needed) state signals.

2. *[10+10 Marks]* Consider the process P with inputs a and c, and output b and d. The behaviour of P is as follows:

$$P = a{+}? \rightarrow (b{+}!||c{+}?) \rightarrow (d{+}!||a{-}?) \rightarrow (b{-}!||c{-}?) \rightarrow d{-}! \rightarrow P \tag{1}$$

   Assume that the environment obeys the protocol: thus, it first generates a+, and then generates c+. It waits for both b+ and c+ to happen before generating c- and so on.

You are required to come up with an extended burst mode (XBM) implementation of this process. You can come up with one or more communicating XBM machines that together implement this process.

(a) Give an XBM state transition diagram of each of the one or more machines you plan to use to implement the process.

(b) Give K-map implementation of each of the output and state variables of the machine(s) you described in the above part.

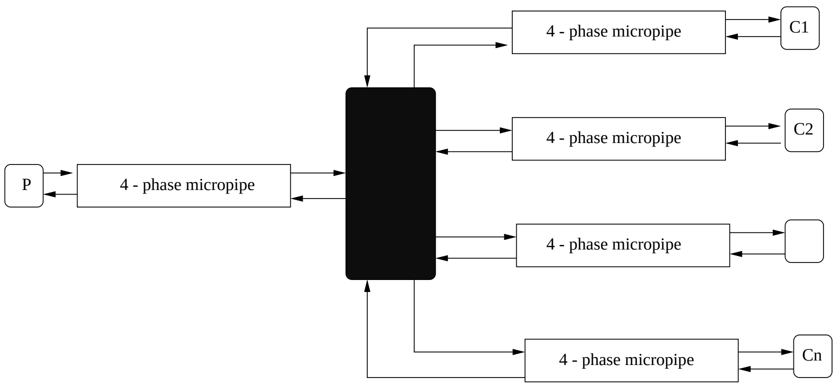

Figure 2:

3. *[10+10+10 Marks]* Recall the 4-phase micropipelines we looked at in Quiz 2. Suppose we have a producer process P connected at one end of such a 4-phase micropipeline. We have a bunch of consumer processes $C_1, ...C_n$ each connected at one end of 4-phase micropipelines. The situation is as shown in Fig. 2. We want to devise a black-box to connect the producer pipeline to the consumer pipelines.

   The black-box should be such that it connects one producer pipeline to more than one consumer pipelines. When a request comes from the producer pipeline, the black-box forwards the request to one of the consumer pipelines connected to it that can accept a request. If there are multiple consumer pipelines that can accept the request, it chooses one arbitrarily. However, if none of the consumer pipelines can accept the request, the black-box holds back the acknowledgement to the producer pipeline so that no further requests are generated from the producer after the producer pipeline is filled up.

   We would like to devise a black-box that allows scalability in connecting an arbitrary number of consumer pipelines. Thus, we want a black box design such that possibly multiple copies of the black box can be connected together to form a large black box to enable us to connect one producer process to as many consumer proceses as we wish. We would not want to design a new black box each time we want to add another consumer pipeline. Instead, we would possibly like to use more copies of the same black box, conect them up together in the right manner and hook them up with the producer and consumer pipelines, such that the requests from the producer pipeline are now distributed among all the consumer pipelines, depending on which is able to accept a request.

   (a) Describe the interface behavior of the black-box that you would choose to connect producer pipelines to multiple consumer pipelines, using a STG.

   (b) How many states does the corresponding state transition diagram have ?

   (c) Describe how you would connect multiple of these black-boxes to connect one producer pipeline to an arbitrary number of consumer pipelines.

4. *[10+10+5+5 Marks]* Consider the process P described in Question 2.

(a) Draw an automaton, clearly indicating the final states and failure states, such that it accepts all valid traces of process P and nothing more.

(b) Suppose a designer came up with an implementation where input a is directly connected to output b by means of a wire, and input c is directly connected to output d by means of another wire. Each of these wires have unbounded delays (you can't make any assumptions about their delays).

   i. Draw an STG describing the behavior of the designer's implementation. You must assume that the environment obeys the protocol described in Question 2. Thus, it waits for both b+ and c+ to happen before generating a-, and so on.

   ii. Draw an automaton that accepts the traces generated by the above STG and nothing more.

   iii. Show that this implementation is incorrect by giving a trace that is accepted by the implementation's automaton, but not by the specification's automaton. You need not mirror and compose the automata to do this. You should be able to find such a trace by simple inspection of the automata.