
CS628 Homework #1

Max marks: 60 **Due Feb 13, 2004**

- *You must take a printout of this homework, and fill in your answers in the spaces provided. You must turn in the sheets filled with your answers.*
- *You may work out the answers separately and then fill in* ***all*** *the steps and the final solution in the spaces provided.*
- *Be brief, complete and stick to what has been asked.*
- *If needed, you may cite results/proofs covered in class without reproducing them.*
- ***Do not copy solutions from others. Penalty for suspected copying: FR grade***

1. Consider an $m \times n$ grid of aynchronous processes as shown in Fig. 1(a).

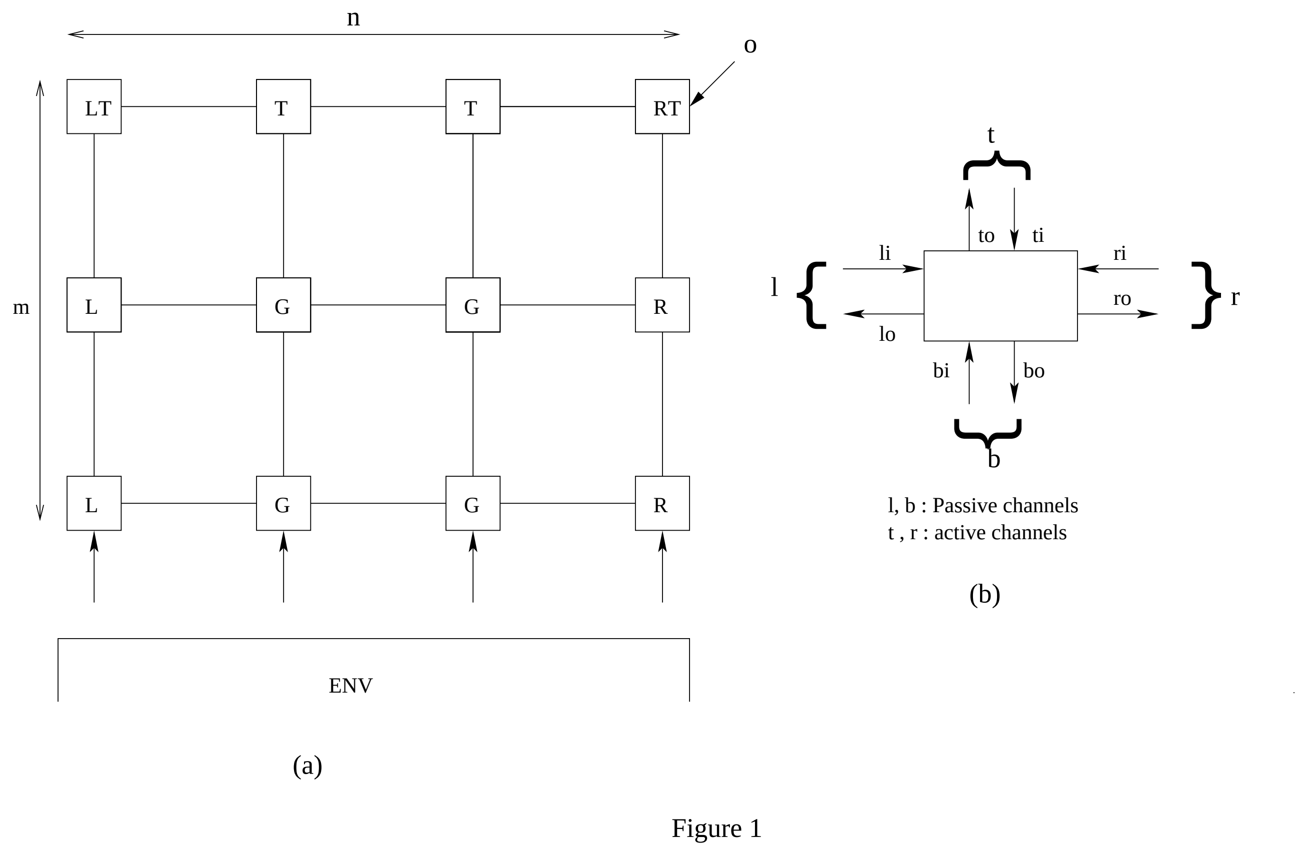

Figure 1: A grid of asynchronous processes

The processes in the grid can be classified into 6 types as follows:

- LT : Process at the top left of the grid.
- RT : Process at the top right of the grid.

- T : Processes in the top row other than LT and RT.
- L : Processes in the left column other than LT.
- R : Processes in the right column other than RT.
- G : All other processes.

Each process has upto 4 neighbours with which it communicates by means of handshakes. As examples, a G process has 4 neighbours, a T process has 3 neighbours, and the RT process has only 2 neighbours.

All handshakes for all processes are of the same type (2-phase or 4-phase). Each process has upto 4 communication channels. The channels are named l, r, t, b, as shown in Fig. 1(b). Each channel uses a binary valued signal/variable for receiving and sending handshake events. These are labeled as in Fig. 1(b). For each process, channels l and b, when present, are passive channels. Similarly, channels t and r, when present, are active channels.

A process receives requests from its bottom and/or left neighbours through the passive channels b and/or l, respectively. Once a process receives such a request, it forwards the request to its top or right neighbour through the active channels t or r, respectively. For a process with both top and right neighbours (e.g. a G process), requests are forwarded alternately to the top neigbour and to the right neighbour, with the top neighbour receiving the first forwarded request. For a process with no top neighbour (e.g. a T process) or no right neighbour (e.g. a R process), each incoming request is forwarded to its right or top neighbour respectively. A process can receive requests from its left or bottom neighbours in any order. Note that the RT process has no active channels to forward requests. In this sense it is a sink process that only receives requests.

The grid communicates with the environment through the bottom row of the grid. The environment pumps requests randomly to any of the processes in the bottom row. It pumps in a new request immediately after receiving an acknowledgment for its previous request from the grid. The request can go to any process in the bottom row randomly regardless of where the acknowledgement was received from the grid.

Our goal is to design processes T, L, R, LT and G such that all requests from the environment are eventually forwarded to the RT process. We would like the handshaking protocol between neighbouring processes to be such that no requests are lost or corrupted en route from the environment to the RT process. We would also like the handshaking protocol to be such that it becomes trivial to extend the grid row-wise or column-wise by connecting another row or column of appropriate processes.

Assuming a two-phase handshaking protocol for all processes, RT can be described by: $RT = (b_i? \rightarrow b_o![]l_i? \rightarrow l_o!) \rightarrow R$. Using a four-phase handshake, the corresponding description would be: $RT = (b_i? \rightarrow b_o! \rightarrow b_i? \rightarrow b_o![]l_i? \rightarrow l_o! \rightarrow l_i? \rightarrow l_o!) \rightarrow RT$.

Processes R and T are similar in the sense that active channel t in process R plays the role of active channel r in process T, and vice versa. Therefore, designing one of R and T suffices for our purposes.

Questions

(a) *[5+5+5 marks]* Choose whether you want to use a four-phase or two-phase handshake for all communication. You cannot choose different handshaking protocols for different processes. Based on your choice, describe the processes LT, L, and R using *Petri Nets*.

(b) *[7 marks]* Based on your choice of handshaking in the previous question, describe the process G using the *process algebraic* notation covered in class.

(c) *[20 marks]* Derive *production rules (PRs)* for process G. Your PRs should explicitly indicate the rising and falling of the signals/variables used. You must also use as weak guards as possible in your PRs.

.

(d) *[5 marks]* The minimum number of hops for a request to reach the *RT* process from the environment is $m-1$ (along the rightmost-column). The maximum number of hops is similarly $m+n-2$ (when the environment sends the request to the bottom-left process). Assuming that the environment sends requests to each of the m processes in the bottom row uniformly randomly, what is the *average number of hops* for a request to reach from the environment to the *RT* process? You must indicate all steps in your answer.

(e) *[10 marks]* Suppose the channel communication delay (time taken for a handshake event to reach a neighbouring process) in each direction is 1 time-unit, and the processing delay of each process is 0 time-units. What are the *fastest* and *slowest* rates at which the environment can pump new requests in the grid? Give complete justification for your answer.

(f) *[3 marks]* If we reshuffle the handshake events in the description of the processes, does it affect the rates in the previous question? You must provide a clear justification for your answer.