

Assignment Lustre and SCADE programming

In each of the following exercise, please construct Lustre code and simulate it using Luciole tool. Please submit pictures of timing diagrams for enough input test cases to illustrate all aspects of the behaviour.

1. Program a controller for CAR Headlights.
There are two inputs, one called ON and another called MODE and two outputs called LIGHT and HI.

MODE can take values NORMAL, FLASHING or HIGH.

If ON is true headlight is lighted.

If MODE is NORMAL the beam is low.

If MODE is FLASHING, the beam is KEPT HIGH for next 10 cycles.

If MODE is HIGH then beam is HI.

2. Design and program a traffic light controller at a 4 way cross road of Main road and Side road. There are sensors on each road which detect whether car is waiting there. (Assume only one lane in each direction.) Use your own imagination to decide the behaviour.
3. Program in Lustre a master slave J-K Flip Flop (with clock) as node with boolean inputs J, K and clock. See http://www.electronics-tutorials.ws/sequential/seq_2.html

Use this JK Flip flop node to program a modulo 12 counter by modifying the circuit given in link below. (see synchronous 4 bit counter in

http://www.electronics-tutorials.ws/counter/count_3.html)

4. A Mealy machine is finite state automaton with output. Each transition also gives output produced when it is taken. Consider the Mealy Machine given at web page:
<http://www2.elo.utfsm.cl/~lsb/elo211/aplicaciones/Pennsylvania/abel.ex1.1.gif>

Encode its input-output behaviour as a Lustre V6 Node.

5. Consider the following properties. We want an output (say OK) to denote that the property holds for the current cycle. Program a Lustre node to output OK for every property below.
 - a) A is continuously true in past.
 - b) Between every A and subsequent B there is at least one C.
 - c) Between every A and subsequent B there is always C. (Interpret this as you see fit.)
 - d) If Req is continuously high for last 3 cycles (including present cycle), then ACK is true otherwise false.
 - e) (BUS ARBITER) The inputs are arrays size n called Req^n and Ack^n .
 - Check whether mutual exclusion of Act holds (use Lustre V6 array iterators).
 - Check whether There is any spurious output Act[i] without Req[i].