The Synchronized State-Machine Puzzle

Tim Smith

April 17, 2004

Abstract

This is a problem that I *think* I got from a Martin Gardner book, where it was stated in terms of a firing squad of soldiers, each of which could only communicate with his immediate neighbor. The problem is to specify a procedure by which they can all fire simultaneously.

I have reformulated the problem in terms of block boxes and state machines, both to make it more precise, and because that is more geekly.

The Black Boxes There are N identical black boxes, arranged in a line. We will call the boxes B_1, B_2, ..., B_N.

Each box has an LED on top.

Each box has an internal clock signal. The boxes all run at the same clock speed, and their clocks are synchronized.

Each box has two inputs, IN_L and IN_R, and two outputs, OUT_L and OUT_R.

Each box may set the outputs to any of 32 voltage levels, V_1, ..., V_{32}.

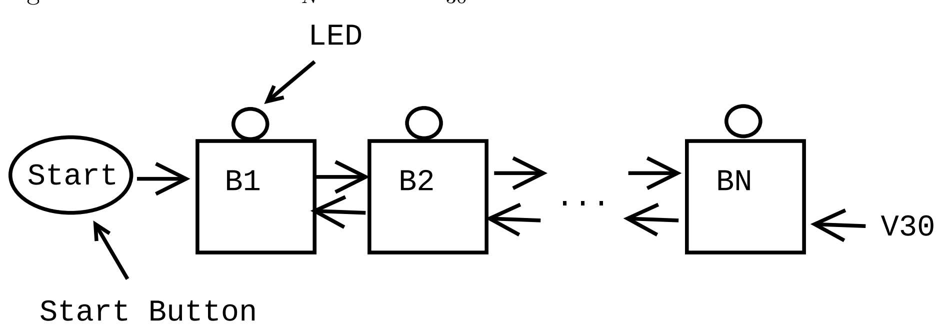
Each box can sense those 32 voltage levels on the inputs. The inputs are latched at the start of each clock cycle.

OUT_R of B_i $(1 \leq i < N)$ is connected to IN_L of B_{i+1}.

OUT_L of B_i $(1 < i \leq N)$ is connected to IN_R of B_{i-1}.

OUT_L of B_1 is not connected to anything. IN_L of B_1 is connected to a button, labeled "Start". When this button is not pressed, B_1 will sense V_{32} on its IN_L. When the "Start" button is pressed, B_1 will sense V_{31}.

OUT_R of B_N is not connected to anything. IN_R of B_N is connected to a voltage source that causes B_N to sense V_{30}.



Each box powers up with its LED off, and with both outputs outputting V_1.

Each box contains a finite state machine. The inputs to the state machine are IN_L and IN_R, and, or course, the current state number. The actions the state machine can take are to turn the LED on or off, and to set the voltage level on OUT_L and OUT_R, and, of course, to change states.

The Puzzle Design the state machine so that when the "Start" button is pressed, all the boxes will turn their LEDs on at the same time. A description of the algorithm the state machines implement will be sufficient to solve this problem. You do not have to produce an actual specification of the state machine, although doing so earns extra geek points.

You may specify constraints on the value of N, provided that there are arbitrary large N that meet those constraints. For example, you may specify that your solution requires that N be even, or that N be prime, or that N be a power of two. You could not require that N be a Mersenne prime, for example, because it is not known if there are arbitrarily large Mersenne primes.

Note N can be *arbitrarily* large. This means that solutions based on counting will NOT work. For example, an algorithm most people quickly come up with is to have the boxes figure out where they are in line, and then to pass the "Start" signal down the line of boxes, and B_i turns on its LED on $N+1-i$ cycles after receiving the signal (or something like that). This solution fails because N can be chosen so large that your state machine will not have enough states to count the boxes. You will have to be cleverer than this!