CS254 (Spring 2018): Lab Assignment 4 Instructions

February 20, 2018

In this lab, you will use the FPGAlink package used in the previous lab to obtain information about railway tracks passing through a junction where your railway signal controller is located. This information will come from a backend computer (in this case, your laptop) through the interface (in software and hardware) provided by FPGALink. You must then process this information, and display appropriate signals in all directions using the LEDs on the Digilent Atlys board.

For the sake of simplicity, we will assume that all railway signaling junctions have co-ordinates in a grid as shown in Fig. **??**. Each junction has a co-ordinate, which is a pair of two 4-bit unsigned integers. Thus, a junction can have co-ordinates $(0010, 0011)$ if it is at position $(2, 3)$ on the grid. We will use this pair of 4-bit unsigned integers as the identity of a junction.

There are 8 possible directions in which railway tracks can cross a junction; let's call them `N, NE, E, SE, S, SW, S, NW`. For convenience, we will assign a 3-bit identifier to each direction, with 000 for `N`, 001 for `NE` and so on, until 111 for `NW`.

Not every junction may have tracks passing through it in all 8 possible directions. We wish to indicate this by a binary variable `TrackExists` for each direction and for each junction. Thus, if `TrackExists` is 1 for direction 001 for junction $(001, 010)$, it means there exists a track passing through the junction at co-ordinates $(1, 2)$ in the `NE` direction.

Even if a junction has a track passing through it in a particular direction, the track may be unusable for various reasons (like maintenance or accident). We wish to indicate this by a binary variable `TrackOK` for each direction and for each junction. Thus, continuing with the above example, if `TrackOK` is 0 for direction 001 for junction $(001, 010)$, then the track through the junction at $(1, 2)$ in `NE` direction is unusable.

When a junction has a track in a particular direction passing through it, the next railway signal (controller) on the grid in the same direction may be several hops away, or may not exist at all. The number of hops until the next railway signal (controller) in a given direction starting from a given junction is denoted by a 3-bit unsigned integer `NextSignal`. Specifically, if `NextSignal` is 0, it means there are no further functional railway signal controllers in the specified direction starting from the current junction. Continuing with our previous example, if `NextSignal` is 001 for direction 001 for junction $(001, 010)$, then the next signal in `NE` direction starting from junction $(1, 2)$ is at co-ordinates $(2, 3)$.

Your job is to design a hardware circuit (which should eventually be part of your railway signal controller) that should do the following:

- It has a dedicated channel (from 0 through 127) assigned to it for reading data from the host computer. Suppose this channel is i.
- It always sends its co-ordinates (two 4-bit unsigned integers) on channel i to the host computer.
- It periodically reads `TrackExists`, `TrackOK`, `direction` and `NextSignal` for each direction in the sequence `N, NE, E, SE, S, SW, S, NW` as an 8-bit quantity from the host computer on channel i. After it has read information for all 8 directions, it waits for 16 seconds to read information for the 8 directions again.

- Once it has read `TrackExists`, `TrackOK`, `direction` and `NextSignal` for tracks in each of the 8 directions passing (or not passing) through it, it must light up the LEDs on the Digilent Atlys board as follows:
  - LEDs L_0, L_1 and L_2 are to be used to indicate *red*, *amber* and *green* signals respectively.
  - LEDs L_5, L_6 and L_7 are to be used to indicate the direction (`N, NE, E, SE, S, SW, S, NW`) for which the signal is intended.
  - LEDs L_3 and L_4 should be switched off at all times for now.
  - If a track exists and is ok in a particular direction but the next signal in that direction is more than 1 hop away, the green signal should turn on for that direction.
  - If a track exists and is ok in a particular direction but the next signal in that direction is ≤ 1 hop away, the amber signal should turn on for that direction.
  - If a track does not exist or exists but is not ok in a particular direction, the red signal should turn on for that direction.
  - The signal for each direction (LEDs L_0, L_1, L_2), along with the code for that direction (LEDs L_5, L_6, L_7) should stay on for 1 second, before the signal for the next direction and the code for the next direction is displayed.
- After displaying signals in all directions for a junction, the controller should read `TrackExists`, `TrackOK` and `NextSignal` for each direction in the sequence `N, NE, E, SE, S, SW, S, NW` again (16 seconds after having read the previous set of 8 bytes) from the host computer on channel i and repeat the above signaling.