Lab 6: Shuttle Bus

Submission Due Dates:

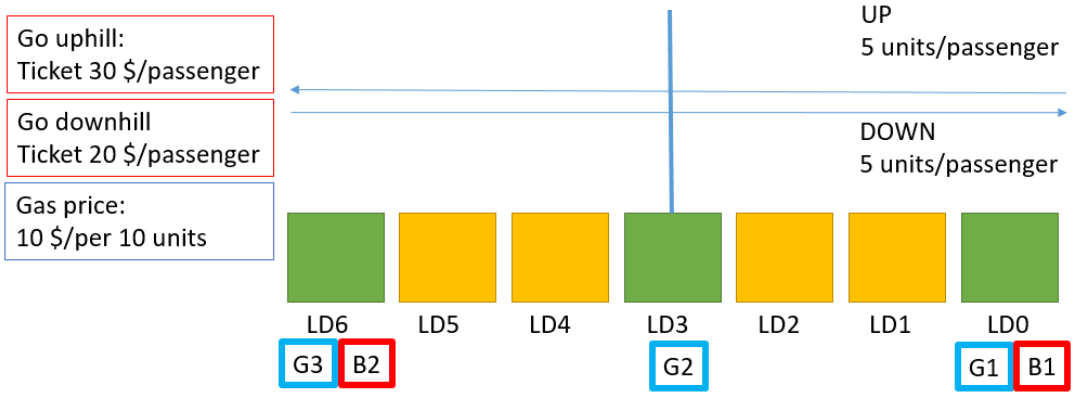
Demo:	2021/11/23 17:20
Source Code:	2021/11/23 18:30
Report:	2021/11/28 23:59

Objective

1 Getting familiar with modeling finite state machines with Verilog.
2 Getting familiar with the keyboard control on the FPGA board.

Description

The community bus company operates a popular Green Line route between the city center and a beautiful mountain top nearby. The shuttle bus goes uphill or downhill to transport passengers. Passengers need to buy one-way tickets to take the bus at Bus Stop B1 or B2. The shuttle bus requires a different amount of gas to run according to the number of passengers on the bus. There are three (3) gas stations, G1, G2, and G3, along the route. The driver should refuel the bus based on the company policy. In this lab assignment, your mission is to stimulate the operation of the Green Line bus route, including the number of passengers on the bus, its position, gas amount, and the revenue of the bus company.



Passenger

Two green bus stops, B1 and B2, are located at LED0 and LED6, respectively.

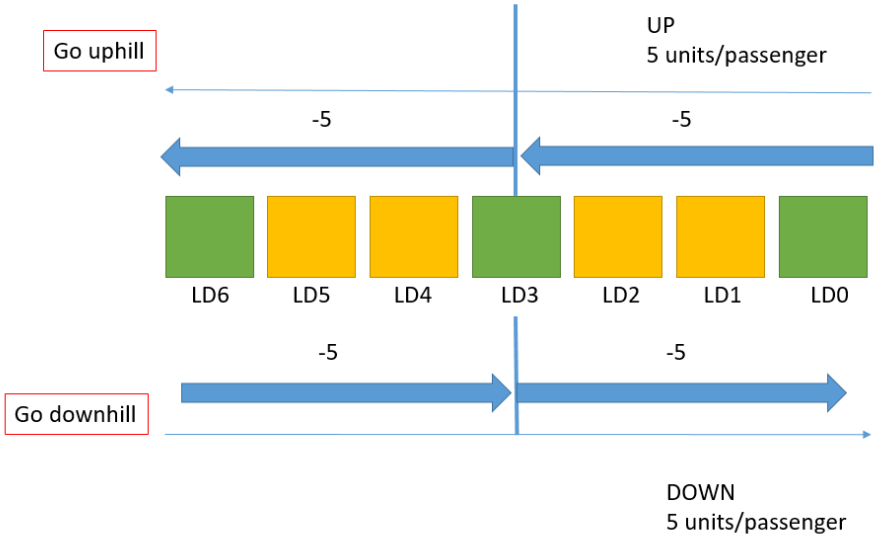
- LED[10:9] presents the number of passengers on the bus. When the bus arrives at a bus stop, passengers get off the bus first. Then the waiting passengers get on the bus. If no passenger is on the bus, passengers waiting at the bus stop can directly get on the bus. People will get on the bus at once but get off one by one.
- Use the keyboard number ‘1’ and ‘2’ to increase the number of people waiting at the Bus Stops B1 and B2, respectively. The maximum capacity of each bus stop is two (2) persons. LED[15:14] and LED[12:11] present the number of passengers at B1 and B2, respectively.
- The maximum capacity of the bus is also two (2).

Gas

Three gas stations, G1, G2, and G3, are located at LED0, LED3, and LED6, respectively.

- Gas is sold 10 units at a time and costs $10.
- DISPLAY[3:2] presents the remaining gas units. The maximum gas capacity of the bus is 20.
  - When the bus arrives at the gas station, the driver can refuel the bus.
  - Gas consumption:
    - The gas consumption between any two adjacent gas stations is 5 units per passenger.
    - You should update the current gas capacity when the bus arrives at LED0, LED3, and LED6.

    For example, if there is (1) passenger on the bus:



Refueling policy

- When the bus arrives at G2 and there is a passenger on the bus.
- When the bus arrives at G1 or G3 and picks up the passenger.
- When there is no passenger on the bus, the driver doesn't have to refuel.
- Fill up if it is possible (i.e., when the revenue ≥ the costs of gas).

Income

DISPLAY[1:0] presents the revenue. The maximum revenue is 90.

Increase:

- When passengers get on the bus, they pay $30 for going uphill or $20 for going downhill.

Decrease:

- Refueling costs $10 per 10 gas units.

Position of the bus

- Use LED[6:0] to show the current position of the bus. When there are no passengers on the bus and no passengers waiting for the bus, the bus stops and stands by at B1 or B2.
- When the bus arrives at one stop if the other stop has passengers waiting, the bus heads for the other stop to pick up passengers.

When the bus stands by at one stop and passengers pop up at both B1 and B2 at the same time, the bus serves the nearest passengers first.

I/O signal specification

- clk: the clock signal with the frequency of 100MHz (connected to pin W5)
- rst: the asynchronous active-high reset (connected to BTNC). The shuttle bus position should be reset to B1 with the LED0 on, while other LEDs are off.
- DISPLAY[6:0]: the signals to show the digits on the 7-segment displays.
- DIGIT[3:0]: the signals to enable one of the 7-segment displays.
- LED[15:0]: LED[6:0] indicates the shuttle bus position; LED[15:14] and LED[12:11] indicate the number of people waiting at B1 and B2; LED[10:9] indicates the number of people on the bus; other LEDs are not used and kept off.

Note

- For the keyboard control, you may refer to KeyboardController and KeyboardDecoder discussed in class.
- PS2_CLK and PS2_DATA are inout signals, that is, bidirectional signals. Don’t change them to input or output.
- LED13, LED8, LED7 are not used in this lab. Make sure to turn them off all the time.
- You must design at least one finite state machine (FSM).
- Use the following template for your design:

```
module lab06(
  input clk,
  input rst,
  inout PS2_CLK,
  inout PS2_DATA,
```

```
  output [3:0] DIGIT,
  output [6:0] DISPLAY,
  output [15:0] LED
  );
  // add your design here
endmodule
```

Demo video:
https://drive.google.com/file/d/1pfFa8BQpW5je9kQZ0vehhLgSbw0b5vmP/view?usp=sharing

Hint

- The recommended frequency for the 7-segment display controller is clk/(2^10), and clk/(2^26) ~ clk/(2^27) for the rest parts of the design.
  - You may refer to the demo video for the bus moving and passengers getting off.
- Designing your block diagram before coding is highly recommended.

Attention

- The XDC constraints file is ready to use without any modification.
- You should hand in one single file, i.e., lab6.v. If you have several modules for your design, merge them into lab6.v. Don't include KeyboardDecoder, Onepulse, Debounce into lab6.v.
- You should also hand in your report as lab6_report_StudentID.pdf (i.e., lab6_report_109000000.pdf).
- You should be able to answer questions of this lab from TA during the demo.
- Please do not hand in any compressed files, which will be considered as an incorrect format.
- You should be able to answer the questions of this lab from TA during the demo.
- You need to generate the bitstream before the demo.
- If you have any questions about the specification, feel free to ask on the EECLASS forum.