

LAB 6. Finite State Machine Design – LED Bouncing Ball

I. Pre-lab

- Review the handout *7 Diagrams* especially the timing diagram section
- [5 pts] Design a first-pass state graph of the controller FSM
- Draw timing diagrams of the following scenarios showing **all** relevant input and output signals
 - [5 pts] $force_fast = 1$. $reset \rightarrow \{mode1, mode0\} = 00$ for 13 clock cycles $\rightarrow \{mode1, mode0\} = 01 \rightarrow 10$ more cycles
 - [5 pts] $force_fast = 0$. Show *count* 5 cycles before and 5 cycles after *enable* pulses one cycle high

II. Main System

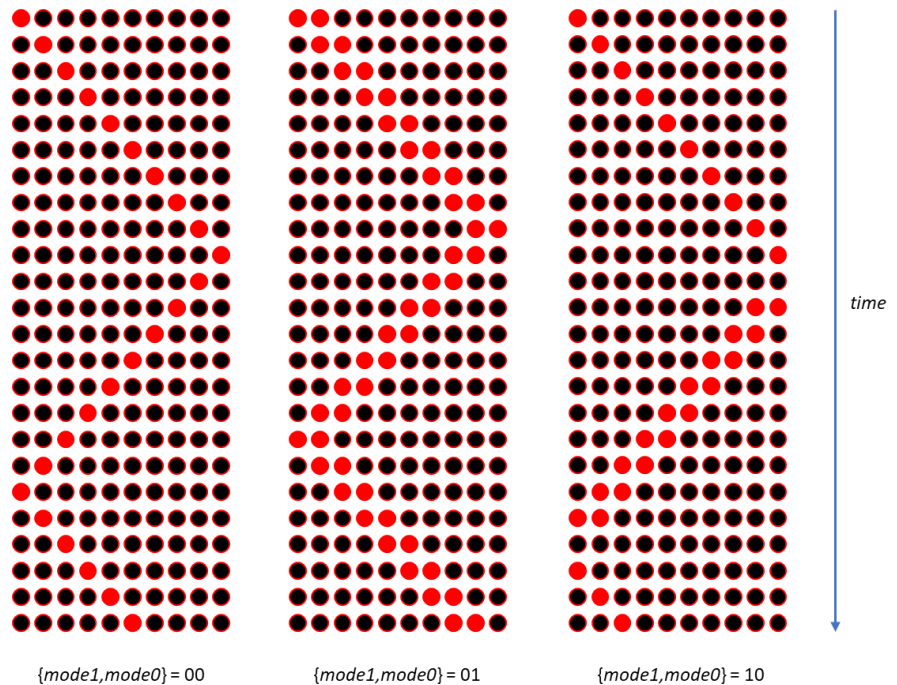
A finite state machine controls a 10-bit bidirectional shift register. The shift register simulates a "bouncing digital ball" moving across an array of 10 LEDs. The three modes of operation are illustrated in the diagram.

The user's inputs are as follows:

- *reset* (either KEY button). Pressing *reset* resets the system, places the ball at the left "wall", and begins the ball bouncing when *reset* is released.
- *mode1, mode0* (any two SW switches)
- *force_fast* (forces system to operate at maximum clock rate)

The 10-bit bidirectional shift register is implemented using D-type flip-flops. Design one slice of the shift register and replicate it ten times. The shift register's outputs are connected to the 10 LEDR LEDs, and it has the following input signals:

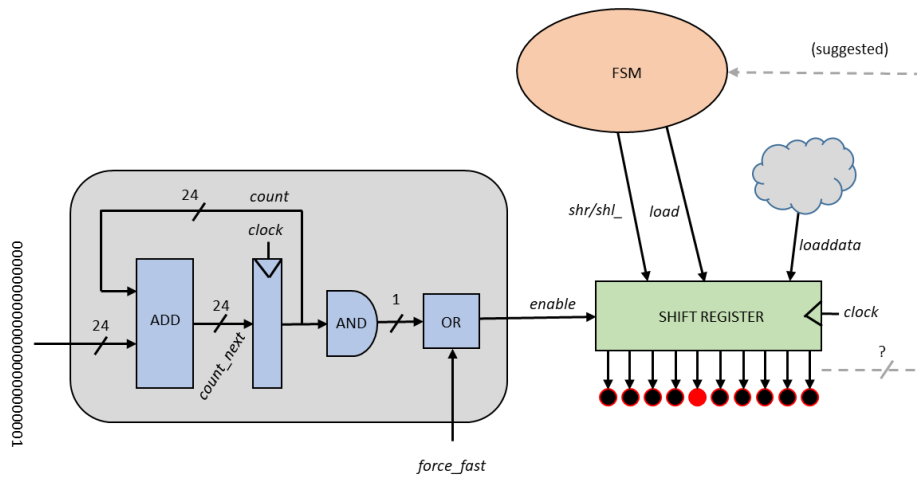
- *shr/shl_* ("0" = shift left, "1" = shift right),
- *load* (parallel 10-bit load),
- *loaddata* (10 bits of data whose source is to be designed by you—give this some thought), and
- *enable*.



As shown in the cases illustrated, the system operates in three modes controlled by the SW switches *mode1* and *mode0*. When *mode1* = 0 and *mode0* = 0, the lit LEDs simulate a moving ball made of one LED. When they equal "01", the moving ball is made of two LEDs. When they equal "10", the pattern has four components: a) single LED left to right, b) blank one cycle, c) double ball right to left, and d) blank one cycle. The patterns repeat indefinitely until the *mode1* and *mode0* inputs change however changes take effect only when the ball touches the left "wall."

III. Clock Divider

The DE10 Lite's 50 MHz clock is far too fast to observe the LED ball moving. Thus, the clock must be effectively slowed. Rather than changing the clock frequency, a robust way to accomplish the same effect is to clock the entire system using the 50 MHz clock but create an *enable* signal which pulses high for exactly one clock period at the desired frequency—in this case 2.98 Hz ($50,000,000 / 2^{24} = 2.98$) which is generated by a 24-bit adder (build a ripple-carry adder using full adders you designed in an earlier lab), a 24-bit register (build with D flip-flops), a 24-input AND gate (built with small AND gates as you wish), and a single 2-input OR gate to implement the input signal *force_fast* which forces the shift register to shift every clock cycle and will be useful in simulation.



System block diagram. For simplicity, many input and output signals are not shown.

IV. Controller

You may build your controller however you like including parameters such as:

- Moore or Mealy
- minimal number of flip-flops or one-hot
- any encoding between states and binary hardware states

V. Other Requirements

Connect each state bit inside your FSM to any LED segment within any of the 7-segment HEX displays for possible debugging help. Verify your design by simulating in ModelSim. Print a copy of your simulation waveforms for one complete cycle for your lab report. Verify your design on your FPGA board.

VI. Recommendations

- To test your clock divider in simulation, temporarily use a large number such as 1000...0 or 0100...0 rather than 0000...1 as the constant value connected to the adder's input.
- Add one LED to each of your state bits for easier debugging.
- Plan on significant debugging time. Plan on more if you check your design quickly.
- Read the web page, "Tips for building and debugging your circuits"

VII. Extra Credit [10 pts max]

Add a total of four different speeds for your bouncing LED(s) by modifying your enable counter and using two SW switches.

VIII. Grading

[15 pts] Prelab

[40 pts] Lab verification (simulation)

[100 pts] Lab verification (FPGA hardware)

[45 pts] Lab Report

- [20 pts] Complete documentation of your final functional design including: State diagram, State assignments, Next state equations, and circuit schematics.
- [10 pts] Tables, K-maps or other work showing how you derived your equations.
- [10 pts] Complete Quartus II schematic and simulation printouts
- [5 pts] Very briefly explain your design choices you made for your FSM design.