LAB#3

( Due: See course web page )

Instructor: Dr. Choon Kim

Objective

- Based on the experience from LAB#1&2, learn how to design, simulate, synthesize, program on FPGA and test FSM (Finite State Machine) digital system using Altera Quartus II CAD SW and DE1 FPGA board.

- Learn and become familiar with logic design using Verilog Hardware Description Language

Instructions

1. Your LAB#3 project name should be L3Cyyy, where yyy=your CID (e.g., L3C079 if your CID=079).

2. [Golden Solution] The golden solution .pof and .sof files are provided. In any LAB, student must play with golden solution as a reference whenever he/she has a question during design. Also check with golden solution before going to Demo to make sure your design follows it. Do not forget reading "Rules when you use a golden solution:" in LAB#1 document.

3. Use Verilog HDL design. Use the following Verilog top-level module interface code for your design. **No part of this code is allowed to be modified.** The top-level module name must be same as your LAB project name.

```
module L3Cyyy(  // where yyy=your CID. For example, L3C079 if your CID=079
    input [9:0] sw,  // ten up-down switches, SW9 - SW0
    input [3:0] key,  // four pushbutton switches, KEY3 - KEY0
    input clock,     // 24MHz clock source on Altera DE1 board
    output [9:0] ledr,  // ten Red LEDs, LEDR9 - LEDR0
    output [7:0] ledg,  // eight Green LEDs, LEDG8 - LEDG0
    output reg [6:0] hex0, hex1, hex2, hex3    // four 7-segment, HEX3 - HEX0
    );
```

3. Our acceptable timing margin for real-time clock operation is -30% and +30%.
A vending machine company requests you to design a Soda Vending Machine (VM) controller circuit with following specifications. The price of a soda was set to 35 cents.

LAB#3 Project Operation Overall Flow

**Warning**: Following operations are ***prerequisite*** conditions. You will get zero(0) point for LAB#3 if you fail these operations regardless of PARTS(#1-5).

1) **Initial state**
   When power is turned on, your DE1 board must be in the following **initial state**:
   - all SWs are in DOWN position
   - no key is PRESSED
   - all leds(ledg and ledr) are OFF
   - VM is in stop state(i.e., not started yet)
   - HEX[3:0] displays your CID. For example, HEX[3:0]=0097 if your CID=097
     (Reminder: Golden solution has HEX[3:0]=0353 since it's CID=353)

2) How to start VM from the **initial state**:
   Press EnterKey(key[3]) once. The hex[3:0] should display "0000" as an initial display.

3) How to stop VM and return to **initial state**:
   Turn off power first and turn on power with initial setting specified in above 1).
VM Operation Specifications
(Reminder: Check & play with golden solution whenever you have a question.)

1) How to deposit money to VM:
Money deposit to VM is made by setting up amount of money first followed by pressing EnterKey[key[3]].
Setting up amount of money (ONLY one sw is 1 at a time!)

- sw[8] // 1 for Reset input (=clearing current Deposit balance and Change balance to 0, but
  // NOT clearing the total number of dispensing - See Report mode case below sec.2)
- sw[4] // 1 for Credit-card input (= makes 35 cents immediately regardless of current balance)
- sw[3] // 1 for One-dollar bill input
- sw[2] // 1 for Quarter input
- sw[1] // 1 for Dime input
- sw[0] // 1 for Nickel input

EnterKey
key[3] // one pressing deposits above money into VM one time.

2) Operating modes (two modes: Normal mode and Report mode)

WARNING: DO NOT operate VM with more than one sw are UP ("Multiple SW input error").

IF sw[9] = 0 { // This is Normal mode.
  IF error cases (See sec. 4.) { // This is error cases displaying error message.
    hex[3:1] = "Err"; hex[0] = off
  }
  ELSE { // This is good (non-error) case ...
    hex[3:2] = Deposit balance. It displays value up to "35". (Note that soda price was set to 35 cents.)
    hex[1:0] = Change balance e.g., 15
    For example, a dispensing occurs and hex[3:0]=35 15 after two quarters (=50 cents) were deposited.
    For example, a dispensing occurs and hex[3:0]=35 20 after three dimes followed by a quarter (=55 cents)
  }
}

ELSE IF sw[9] = 1 (with all other sw are down) { // This is Report mode. This is a pause (not reset!) operation.
  - hex[0] displays the total number of dispensing made only by coin input (NOT by One-dollar bill or
    Credit-card-input!) since the board power was turned on.
  - Therefore, DO NOT make One-dollar-bill input or Credit-card input before testing sw[9]=1 function.
    Only coin input is allowed BEFORE testing this condition.
  - hex[0] is in hex with modulo-16 format. hex[3:1] = OFF.
  - The sw[8](=Reset input) should NOT reset the total number of dispensing value to 0.
  - WARNING: This sw[9]=1 operation pauses (not reset!) the current operation. Therefore it should NOT CHANGE
    the value of current variables in your design, such as Deposit balance or Change balance. So, when
    sw[9] returns back to 0(down, i.e., Normal mode), hex[3:0] must resume displaying the previous value
    that it was displaying before sw[9] went 1(up). For example, hex[3:0]=35 15 if it had been displaying 35 15.
}
3) Dispensing indication
Whenever the Deposit balance is 35 or more, a dispensing occurs with following action(s).
(For example, a dispensing occurs after three dimes followed by a quarter (=55 cents)

- When NOT dispensing, all LEDs(i.e., both green and red) are OFF.
- When dispensing, all green LEDs(led[7:0]) are blinking(half-second period with 50% duty cycle)

4) Three Error cases to be checked: ("Err " display cases)

4.1) Multiple inputs case:
When more than one sw are UP among sw[0,1,2,3,4,8] and EnterKey is pressed(e.g., sw[2]=sw[3]=1), hex[3:0] should display "Err " (note hex[0]=OFF).

4.2) Consecutive same inputs of one-dollar bill or credit-card case:
Two consecutive one-dollar bill inputs or two consecutive credit-card inputs should display "Err " on hex[3:0]. For example, one-dollar bill input followed by one-dollar bill input, or credit-card input followed by credit-card input.
(However, note that one-dollar bill input followed by credit-card input, or credit-card input followed by one-dollar bill input is O.K.)

4.3) Credit Card input when hex[3:0]=3500 and LEDG blinking(i.e., when Deposit=35 and Change=00)
This is another error case when hex[3:0] should display "Err ".

How to clear(=reset) the "Err " display and continue normal operation?
The "Err " display should be cleared to hex[3:0]=0000(i.e., value of Deposit and Change balance are reset to zero) by either pressing EnterKey or Reset input. Then normal operation can continue.

Operations to be checked during Demo

PART 1(3 pts) : Coin input (i.e., Quarter, Dime, Nickel) test
PART 2(3 pts) : Report mode test
PART 3(3 pts) : Credit-card and Reset inputs test
PART 4(3 pts) : One-dollar bill input test
PART 5(3 pts) : "Err " display test