Design of Digital Systems: Lab Assignment 5

The objective of this assignment is to implement a vending machine system using a finite state machine.

Background

The actions taken by a computer as a result of inputs often depend on previous inputs. State machine is a good choice here as it simplifies the design process. Given their popularity, the conversion of the logic is performed by the synthesis tools, leaving the engineer to focus on the design of the state machine.

Program Specification

In this assignment, a vending machine system is prebuilt for you as shown in Figure 1. However, the vending machine subsystem is missing. Your task is to implement this subsystem. The vending machine system, excluding the subsystem, is provided to you in the file `vending_machine_system.vhd` on Canvas. Make sure to add it into your project.

The vending machine system prevents the subsystem from having multiple coin pushes or soda requests happening simultaneously. Therefore, when handling the subsystem, you will assume that only one action will happen at any single time and you will have enough time to complete the action before the next action happens. In addition, the locking system provides a lock signal that is enabled for around 2.5 seconds. You will use this signal to keep the LED signals (`soda_drop`, `coin_reject`, `error_amt`, `error_reserved`) on for 2.5 seconds so that you can see the LED instead of an instant blink.

Note that Figure 1 doesn't provide the complete picture of the system but instead provides the necessary information for you to understand what is connected to the subsystem.

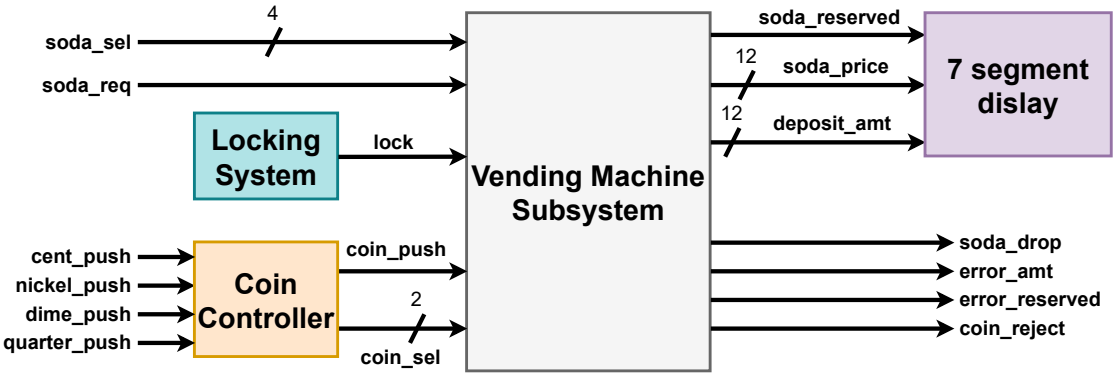

Figure 1: Vending machine

The design must meet the following specifications and use all components listed at least once. Some will be used multiple times.

Vending Machine Subsystem

The vending machine subsystem (name it `vending_machine_subsystem`) is show in Figure 2. The subsystem has the following entity:

- Inputs:
  - `clk` (1 bit): The clock signal. Flip-flops are triggered on the rising edge.
  - `rst` (1 bit): The reset signal. Synchronous, active low.
  - `soda_sel` (4 bits): Selects a specific soda
  - `soda_req` (1 bits): Requests the specified soda
  - `lock` (1 bit): Locks the LED signals
  - `coin_push` (1 bit): Pushes the coin
  - `coin_sel` (2 bits): Selects the coin to push
- Outputs:
  - `soda_reserved` (1 bit): Selected soda is reserved
  - `soda_price` (12 bits): Price of selected soda in cents
  - `soda_drop` (1 bit): Drop the requested soda
  - `error_amt` (1 bit): Requested soda price is greater than deposit amount
  - `error_reserved` (1 bit): Requested soda is reserved
  - `coin_reject` (1 bit): Reject coin
  - `deposit_amt` (12 bits): Deposit amount

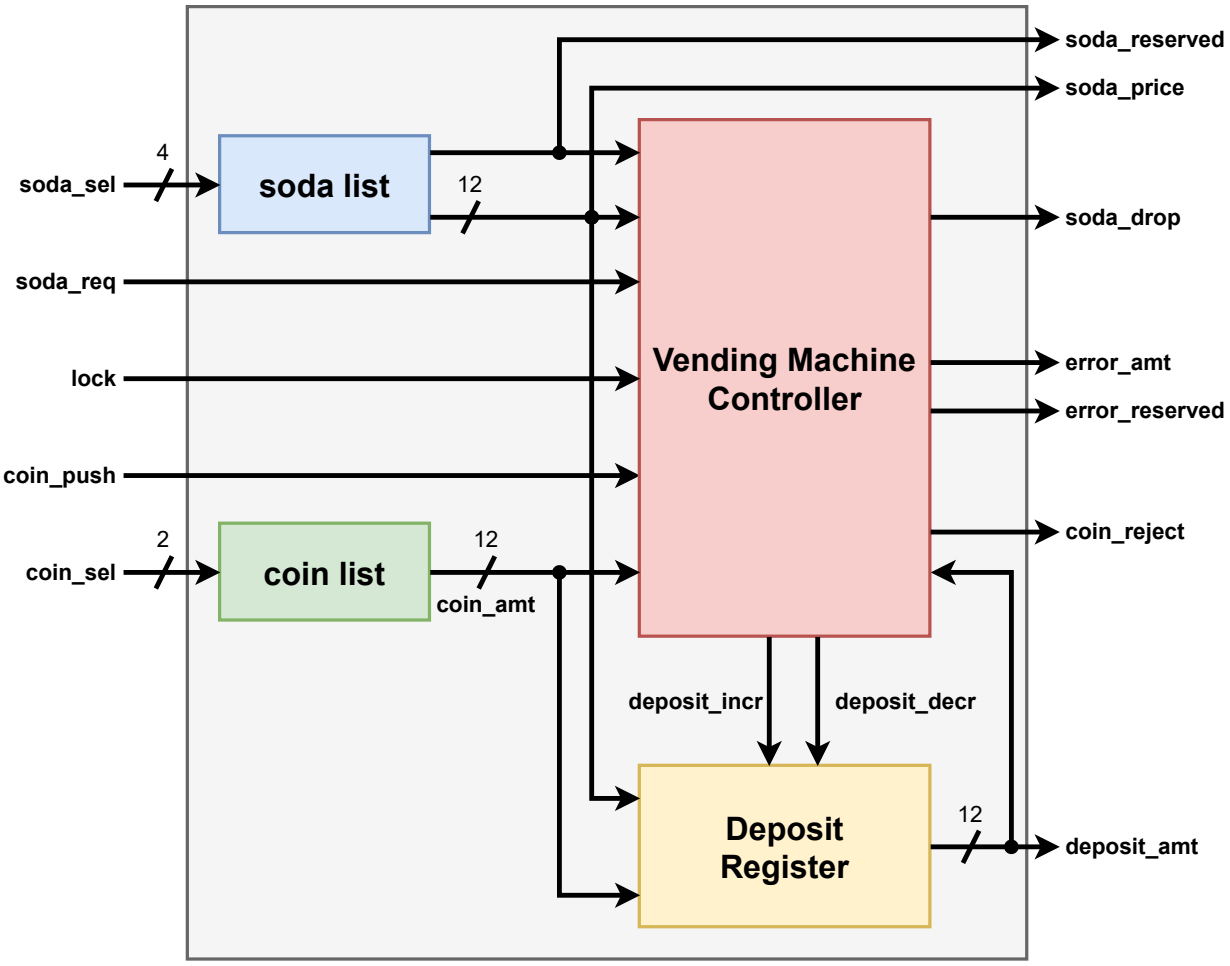

Figure 2: Vending machine subsystem

Soda List

The soda list (name it `soda_list`) gives the price of the selected soda in cents (12-bit integer) and tells whether or not that soda is reserved. Table 1 provides the list of sodas available in the vending machine and their prices.

Table 1: Soda list

Soda	Price
0000	$0.55
0001	$0.85
0010	$0.95
0011	$1.25
0100	$1.35
0101	$1.50
0110	$2.25
0111	$2.50
1000	$3.00
1001-1111	Reserved

Coin List

The coin list (name it `coin_list`) gives the value of the selected coin in cents (12-bit integer). Table 2 provides the value of each coin.

Table 2: Coin list

Coin	Price
00	$0.01
01	$0.05
10	$0.10
11	$0.25

Deposit Register

The deposit register (name it `deposit_register`) holds the amount deposited in cents thus far. The subsystem has the following entity:

- Inputs:
  - `clk` (1 bit): The clock signal. Flip-flops are triggered on the rising edge.
  - `rst` (1 bit): Resets the deposit amount to \$0.00. Synchronous, active low.
  - `incr` (1 bit): Increments the deposit amount by the `incr_amt`
  - `incr_amt` (12 bits): The amount to increment in cents
  - `decr` (1 bit): Decrements the deposit amount by the `decr_amt`
  - `decr_amt` (12 bits): The amount to decrement in cents
- Outputs:
  - `amt` (12 bits): Deposit amount in cents

Vending Machine Controller

The vending machine controller (name it `vending_machine_ctrl`) implements the finite state machine (FSM) provided in Figure 3. Note that when pushing a coin, the coin is rejected when it makes the deposit amount larger than \$10.00. The controller has the following entity:

- Inputs:
  - `clk` (1 bit): The clock signal. Flip-flops are triggered on the rising edge.
  - `rst` (1 bit): Resets the FSM to the IDLE state. Synchronous, active low.
  - `lock` (1 bit): Locks the LED signals
  - `soda_reserved` (1 bit): The soda is reserved
  - `soda_price` (12 bits): The price of the soda in cents
  - `soda_req` (1 bit): Request a soda
  - `deposit_amt` (12 bits): Deposit amount in cents
  - `coin_push` (1 bit): Pushes the coin in
  - `coin_amt` (12 bits): Coin amount in cents
- Outputs:
  - `soda_drop` (1 bit): Drop the requested soda
  - `deposit_incr` (1 bit): Request to increment the deposit amount

- `deposit_decr` (1 bit): Request to decrement the deposit amount
- `coin_reject` (1 bit): Request coin
- `error_amt` (1 bit): Requested soda price is greater than deposit amount
- `error_reserved` (1 bit): Requested soda is reserved

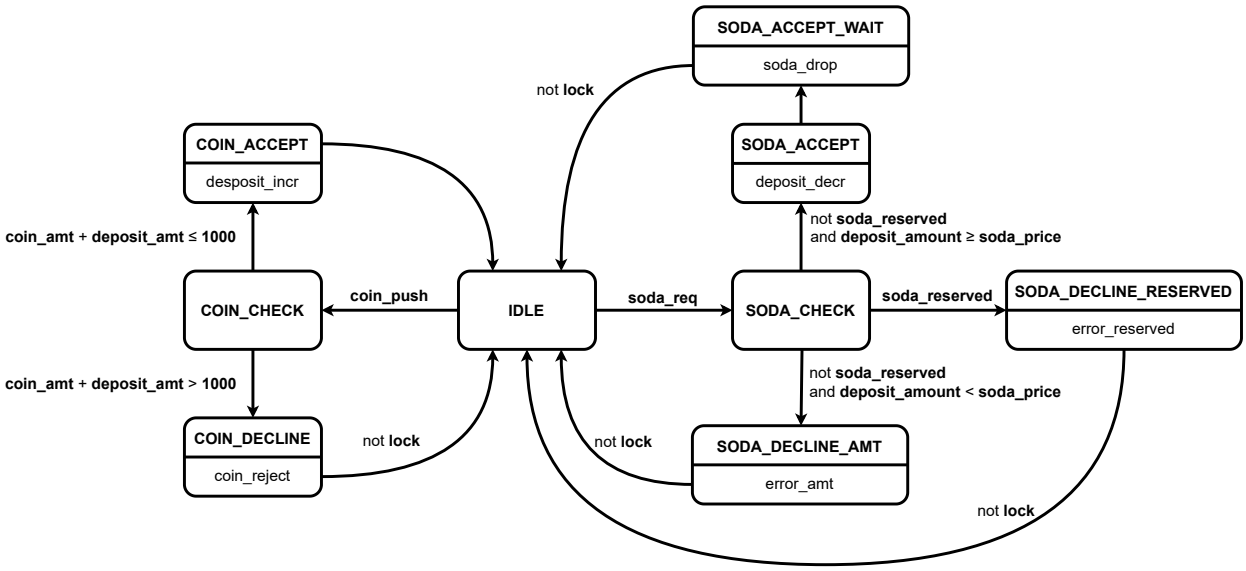

Figure 3: Vending machine finite state machine

Testbench

The testbench is provided to you in the file `vending_machine_subsystem_tb.vhd` on Canvas. The testbench will test every possible scenario and will give you an error in the TCL Console at the bottom window of Vivado for every scenario that fails. Use the waveform, as well as, the description of the error to figure out the issues and fix them. Once all scenarios are passing, you will get the output "`Simulation passed!`" in the TCL Console.

Submission

- For the source code, only submit the vending machine **SUB**system design sources. Furthermore, the simulation and constraints sources are provided to you and should not be submitted.
- For this lab, do not include waveforms. Instead report what you get in the TCL Console at the bottom window of Vivado.

- Collect the area and timing results.
- For the demo, test all possible scenarios.

FPGA port map

The constraint file is provided to you in the file `vending_machine_constraints.xdc` on Canvas. The soda price will show up on the left 4 digits of the 7-segment display, while the deposit amount will show up on the right 4 digits of the 7-segment display. The ports for the remaining IO signals are summarized in the following table.

IO	**FPGA port**
clk	E3
rst	Button CPU RESET
soda_sel	Switches 3-0
soda_req	Button BTNC
cent_push	Button BTNU
nickel_push	Button BTNR
dime_push	Button BTND
quarter_push	Button BTNL
soda_drop	LED 0
error_amt	LED 1
error_reserved	LED 2
coin_reject	LED 3