



CE1911 LABORATORY

SYSTEM DESIGN SPECIFICATIONS

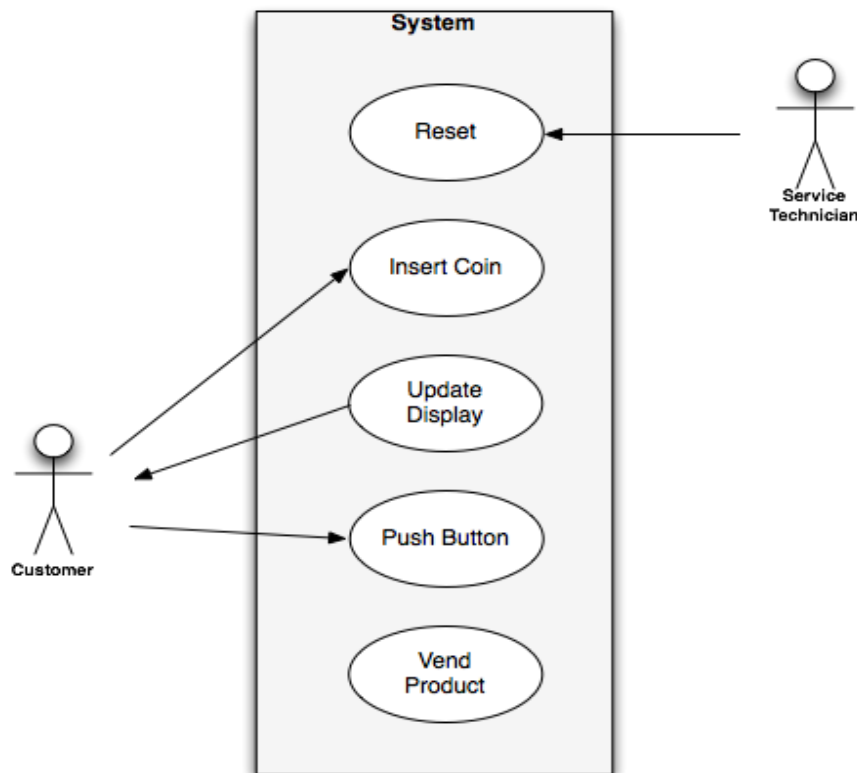
DESCRIPTION

This specification describes and defines the basic requirements of the CE1911 vending machine controller. The controller accepts coins, displays the current coin total, and dispenses Pepsi and water bottles when enough money has been collected and the user presses a choice button. The machine does not dispense change for coin overage – the coin mechanism drops coins through to the coin return box when enough money has already been inserted. All bottles cost \$1.50.

REQUIREMENTS

1. The system must accept quarters. No other coin is accepted.
2. The system must reset into a state with a zero-cent balance.
3. The system must provide a coin total on a user display.
4. The system must display the word Pepsi or the word Water when vending.
5. The system must spend one second in a state that displays the fixed price of \$1.50 when a vend button is pressed but not enough money has been inserted. This behavior reminds the user that they have not inserted enough money.
6. The system must operate from DC power or USB supplied power.

USE-CASE DIAGRAM





CE1911 LABORATORY

SYSTEM DESIGN SPECIFICATIONS

USE-CASE EVENTS

1. Reset Event
 - A. The service technician or power-on resets the system.
 - B. The system identifies the reset request.
 - C. The system responds by resetting into a zero cents balance state.

2. Insert Coin
 - A. The customer inserts a nickel, dime, or quarter.
 - B. Nickels and dimes are dropped to the coin return box by the coin mechanism.
 - C. Quarters are accepted and dropped to the cash box.
 - B. If a quarter is inserted then the system updates to a new coin balance state.
 - C. The system updates the display.

3. Update Display
 - A. An inserted quarter causes a display update.
 - B. The display updates to the new coin balance.

Alternate Path

 - A. The user pushes a vend button when insufficient funds are collected.
 - B. The display updates to 1.50 for one second.

Alternate Path

 - A. The user pushes a vend button when sufficient funds have been collected.
 - B. The display updates to Pepsi for one second if Pepsi is selected.
 - C. The display updates to Water for one second if Water is selected.

Alternate Path

 - A. The service technician pushes the reset button.
 - B. The display updates to reset for one second.

4. Push Button
 - A. The user pushes a vend button when sufficient funds are collected.
 - B. The system vends the selected product.

Alternate Path

 - A. The user pushes a vend button when insufficient funds are collected.
 - B. The system updates the display for one second with the product price.

5. Vend Product
 - A. The system vends the chosen product. Vending requires one second.
 - B. The system returns to the zero cent balance state.



CE1911 LABORATORY

SYSTEM DESIGN SPECIFICATIONS

SPECIFICATION OF SYSTEM INPUTS AND OUTPUTS

1. System Inputs
 - A. The system uses an **active-low** digital input to request synchronous system reset.
 - i. The pushbutton produces logic 0 or logic 1.
 - ii. The pushbutton request is **active when low** (active-low).
 - B. The system uses **active-high** digital inputs Q, P, and W.
 - i. The Q signal is active when a quarter has been inserted.
 - ii. The P signal is active when the Pepsi selection button has been pressed.
 - iii. The W signal is active when the Water selection button has been pressed.
2. System Outputs
 - A. The system outputs two active-high signals called VP, and VW.
 - i. Output VP drives the vend motor to dispense Pepsi.
 - ii. Output VW drives the vend motor to dispense Water.
 - B. The system produces forty digital output signals that drive the LED segments of a five 7-segment displays mounted on the front panel of the vending machine.
 - i. The system does not use a dollar sign (\$) symbol.
 - ii. The system uses the DSEG7 font alpha-numerical font for all numbers and letters displayed on the seven-segment display.
 - iii. **Reference:** <http://www.keshikan.net/fonts-e.html>
 - iv. The forty signals are created as five 8-bit vectors called HEX5, HEX4, HEX3, HEX2, HEX1, and HEX0.

USER INTERFACE SPECIFICATIONS

The following user interface values are generated.

000 025 050 075 100 125 150

RESET PEPSI WATER



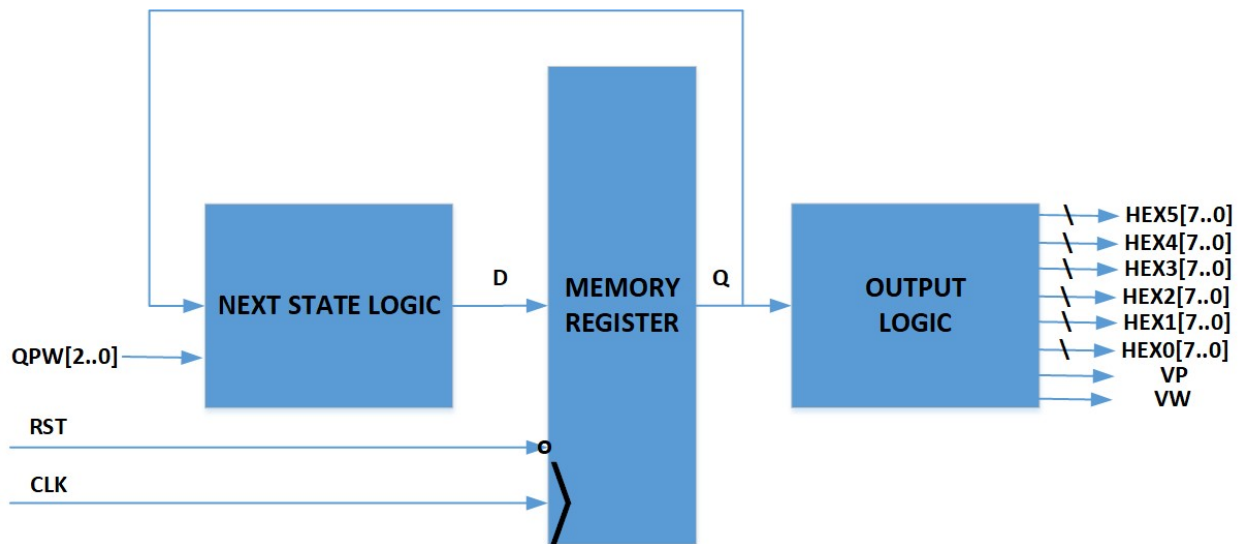
CE1911 LABORATORY SYSTEM DESIGN SPECIFICATIONS

FUNCTIONAL SPECIFICATION

The system is a Moore finite state machine.

1. The machine enters the reset state when the active-low pushbutton is pushed. The reset state always transitions to the zero cents balance state on the next clock tick. The machine displays rESEt on the 7-segment displays when the machine is in the reset state.
2. The machine responds to coin insertions by moving to appropriate balance states.
3. The maximum balance state is S150. No change is given for coin overage.
4. The machine responds to vend buttons by vending when \$1.50 is collected.
5. The machine responds to vend buttons with insufficient funds by displaying \$1.50 for one second.
6. The machine responds to simultaneous quarter and button signals by ignoring them. The coin mechanism drops the quarter through to the coin return box.

MOORE MACHINE DIAGRAM





CE1911 LABORATORY

SYSTEM DESIGN SPECIFICATIONS

IMPLEMENTATION TECHNOLOGY

1. A single behavioral VHDL file describes the complete system.
2. The system resides as a configured circuit in an off-the-shelf Altera DE10-Lite board.
3. Table 1 documents expected project files.
4. Tables 2 and 3 documents Altera DE10-Lite pin assignments.

Table 1: Project Files

PROJECT NAME: LW6			
FILENAME	FORMAT	NEW OR REUSED	CIRCUIT
lw6.vhd	behavioral VHDL	new	complete Moore machine implementation

Table 2: DE10-Lite Pin Assignments for System Inputs and Basic Outputs

SIGNAL	I/O	DE10 PIN	DE10-LITE I/O DEVICE	DE10-LITE USER MANUAL
CLK	input	P11	clock circuit	Page 24
RST	input	B8	pushbutton KEY0	Page 25
Q	input	D12	slide switch SW2	Page 26
P	input	C11	slide switch SW1	Page 26
W	input	C10	slide switch SW0	Page 26
VP	output	AB6	Arduino header Arduino_IO1	Pages 32 and 33
VW	output	AB5	Arduino header Arduino_IO0	Pages 32 and 33
HEX5	output bus	see manual	7-segment display HEX5	Pages 28 and 29
HEX4	output bus	see manual	7-segment display HEX4	Pages 28 and 29
HEX3	output bus	see manual	7-segment display HEX3	Pages 28 and 29
HEX2	output bus	see manual	7-segment display HEX2	Pages 28 and 29
HEX1	output bus	see manual	7-segment display HEX1	Pages 28 and 29
HEX0	output bus	see manual	7-segment display HEX0	Pages 28 and 29

SIMULATION TEST BENCH

A VHDL test bench verifies the reset and operational use cases.

1. Write a clock process that generates a 1s clock period.
2. Write a tester process that drives RST active low for two clock periods.
3. Ensure that RST does not rise at the same time as a rising clock edge.
4. Control the QPW input bus using the tester process.
5. Test a perfect quarter entry sequence with a vend request for Pepsi. Capture the simulation screenshot.
6. Rewrite the tester process to assert Q, Q, Q, P, Q, W, Q, W through time. This simulates a random user input pattern.
7. Rewrite the tester process to assert Q, Q, P and W, Q, Q, Q, Q, Q, Q, P and W through time. This simulates a random user input with the user pressing both buttons simultaneously.

LABORATORY TEST PLAN

Laboratory testing verifies the reset and operational use cases.

1. Add a VHDL clock divider to the **lw6.vhd** file. The divider slows the 50MHz clock to 1Hz.
2. Rebuild the design.
3. Use the Quartus Programmer Tool to configure the circuit in the DE10-Lite MAX 10M50 FPGA.
4. Operate the toggle and pushbutton switches. Verify correct operation.



CE1911 LABORATORY

SYSTEM DESIGN SPECIFICATIONS

HINTS

1. This machine can be solved with around 17 states.
2. The reset event enters the RESET state.
3. A three-bit input bus called QPW will make the next-state logic easier to write.
4. Each state has eight exit arrows because of the three-bit input bus.
5. Describe all eight combinations in each state.
6. Pay close attention to functional specification 6: Input patterns 101, 110, and 111 are ignored and the machine remains in the same state.
7. It is challenging to build a machine with only one error state because how would that error state know which balance state to return to. A simple implementation mirrors each insufficient balance state with a corresponding display-price state. For example: S0 and DP0 or S25 and DP25.
8. Consider using VHDL constants to define the 7-segment display patterns. Then, use the constant name in all with-selects for 7-segment busses.
9. Consider using VHDL to assign the pins rather than using the pin planner.

architecture behavioral of lw7 is

```
-- assign pins
attribute chip_pin: string;
attribute chip_pin of CLK: signal is "P11";
attribute chip_pin of RST: signal is "B8";
attribute chip_pin of QPW: signal is "D12,C11,C10";
...

-- hex display values
constant zero : std_logic_vector(7 downto 0) := B"11000000";
constant one  : std_logic_vector(7 downto 0) := B"11111001";
...
constant letterP: std_logic_vector(7 downto 0) := B"10001100";
...

begin

    -- next state logic
    ...

    -- register process
    ...

    -- output logic
    with Q select
    HEX0 <= zero when S0,
           zero when S50,
    ...
```