VENDING MACHINE

ECE261 Project Proposal Presentation

Members: ZHANG, Yulin
         CHEN, Zhe
         ZHANG, Yanni
         ZHANG, Yayuan
Abstract

This project will design and implement a coin operated vending machine controller.

The project involved:
• System-level architecture (top-level schematic)
• cell designs (register, 7-segment LCD)
• gate-level schematic → Verification of the functionality: do simulation (digital and analog) using ModelSim
• transistor-level schematic
• gate layout → pass DRC/LVS
• Cost estimation
Vending Machine Features

• Vend 4 products
• Accept 3 kinds of coins
• Change back in Nickels
• Displaying the amount of money inputted
• Displaying which product a user has chosen
• One coin can be entered at one time
• Self-starting with the initial coin
• The controller will display warning if insufficient money has been deposited
• First insert coins, then choose one product at a time!! (cannot vend more than one product at a time)
Design Specifications

- **Inputs**: Dimes, Nickels, Quarters
- **Outputs**: Product and/or coins (only one cent)
- **User Interface**
  - **Buttons**: A,B,C,D for product select. These buttons will be used as the action button; E for Cash back
  - **LCDs**: Two digital 7-seg LCDs showing amount of coins inputted; one digital 7-seg LCD showing which product the user chooses
Data flow Diagram

Vending Machine Controller of Logic

Input Vector X

Output Vector Y

4 electrical-to-mechanical drivers

7-Segment Display Decoder 1

7-Segment Display Decoder 2

To be implemented
Top-level Schematic (1)

encoder1:
3-bit input to 8-bit output

Adder
D[7:0] A1 Q[7:0]

ItemPrice [7:0]

Items Prices Logic

Encoder2:
4-bit input to 7-bit output

Money [7:0]

7-seg digital display

Clk1=20MHZ

reset
Top-level Schematic (2)

If difference > 0 | difference = 0 then localreset2 = 1;
If difference < 0 Then localreset = 1; Y display error.
Vending Machine Controller of Logic (1)

Input $X_1, X_2, X_3$

$X_1 = 1$: a nickle is deposited in the slot
$X_2 = 1$: a dime is deposited in the slot
$X_3 = 1$: a quarter is deposited in the slot

Reset: when reset = 1, the input ports are eliminated
usually reset keeps “0”.

Encoder 1:
3-bit input to 8-bit output

<table>
<thead>
<tr>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>B0</th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>B5</th>
<th>B6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Vending Machine Controller of Logic (2)

Adder

CLK↑

D[7:0]

Q[7:0]

B[7:0]

Q[7:0]

CLK1 = 20MHZ
Vending Machine Controller of Logic(4)

Encoder2:
4-bit input to 7-bit output

item#[6:0]

7-seg digital display

<table>
<thead>
<tr>
<th>m0</th>
<th>m1</th>
<th>m2</th>
<th>m3</th>
<th>item0</th>
<th>item1</th>
<th>item2</th>
<th>item3</th>
<th>item4</th>
<th>item5</th>
<th>item6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Vending Machine Controller of Logic(3)

Use shift register to implement “item” block

Prices of each item has been programmed before user uses the machine
Number of Transistors

8-bit+8-bit input Adder: 640 Transistors
Four 8-bit output shift registers: 4*8*16=512 Transistors
One 8 D flip-flop arrays: 8*16=144 Transistors
One 4 D flip-flop arrays 144/2=77 T
16-bit input subtracter: approximately 448+236=684 T
First encoder(3-bit input to 8-bit output): Two 2-input OR Gates=2*4=8 T
Second encoder(4-bit input to 7-bit output): Two 2-input OR Gates=2*4=8 T
Three 4-input AND Gates: 3*(4*2+2)=30 T
Six 2-input OR Gates: 6*(2*2+2)=36T
One 3-input OR Gate and One-4-input OR Gate: (6+2)+(8+2)=18T
One 2-input And Gate: 4T
One D flip-flop: 16T; One inverter: 2T
(8+4) And Gates: 4*12=28T
Third encoder(8-bit to 14-bit output): approximately 250T
Four ens and Three Ors: 4*8*(4+2)+3*8*(4+2)=256T

Total: approximately 2447 Transistors
System Clocks & Power

System contains two clocks: clk1-20MHz  clk2-10MHz

Clk1 is the main clock which controls the A1 D flip-flop array and A2 D flip-flop array. These two arrays are used to maintain the system status.

System status is defined by the money deposited and the item which the customer wants to buy. Since the “1” level voltage created by coins and by user’s pressing the button cannot maintain a long time, we use D flip-flop arrays to record the amount of money and the item.

Clk2 is used to control a D flip-flop which can delay the localreset2 signal to make sure that before the system reset it has finished enabling the mechanical component.

VDD=5v. Suppose average capacitance of each nMOS transistor is C then each pMOS transistor is 2C, then the average capacitance of each MOS transistor is 1.5C. Given the activity factor of static logic circuits is 0.1 while for register and flip-flop circuits the value is 1, then we can calculate the power by using $P=aC\times V\times V\times f$

$$P = (144+77)\times 1\times 1.5C\times 25\times 20MHz + 16\times 1\times 1.5C\times 25\times 10MHz + (2447-144-77-16)\times 0.1\times 1.5C\times 25\times 20MHZ = 1219500 \times 1.5C \times (\text{volt}\times\text{volt}) \times \text{MHz}$$

If $1.5C = 12\times 10^{-6}\text{nF}$ Then $P \sim 14.6\text{mW}$