Lab_3: Vending Machine

Introduction

We need to design a finite state machine for a vending machine. The vending machine can accept coins of 1 Dollar, 50 cents, 10 cents, and 5 cents. The price for any item sold in the vending machine will be set at startup. In cents, the selling price is the full multiples of 5 and never great than 315. When the inserted coins are more than the price, the vending machine will release the item and change.

Declare of the top-level

Please use the entity declaration as below：

```vhdl
entity VENDING_MACHINE is
    Port(Enable : in STD_LOGIC;
        RST : in STD_LOGIC;
        CLK : in STD_LOGIC;
        OneDollar : in STD_LOGIC;
        FiftyCents : in STD_LOGIC;
        TenCents : in STD_LOGIC;
        FiveCents : in STD_LOGIC;
        Deliver : out STD_LOGIC;
        Money : out STD_LOGIC_VECTOR (5 downto 0));
end VENDING_MACHINE;
```

Requirements

1. Behavior description

a) Each coin input will generate a square-wave pulse at the corresponding port. Due to the special construction of the mechanism, only one coin is thrown at a time. That is, there are no two square-wave pulses at the same time.

b) In the initial state, the input signal “`Enable`” is set to 0, and the vending machine is in **price-setting mode.** Set the selling price by putting in a coin. For example, if you want to set the selling price to $1.25, you can put in a dollar coin, two ten-cent coins, and a five-cent coin. The total amount of coins already invested will be exported via the `Money` port. You can output this amount in any encoding you like. If you don't set a sale price, it defaults to $1.25.

c) After setting the selling price, set the enable signal to 1 and enter the selling mode.

The total amount of coins already invested will be exported via the `Money` port. When you enter sales mode, you will no longer care about the value of the input port `Enable`

d) When the inserted coins are equal to or more than the price, the output port `Deliver` will be set to 1 and the amount of change (if exist) will be exported through the `Money` port.

e) When the `Deliver` port be set to 1, wait for 5 seconds and go back to sales mode. It's ready for next purchase.

f) Whenever the RST is set to 1, the vending machine will return to the initial state at the nearest clock rising edge.

2. Switch Debouncing

In the Behavior description we mentioned that for every coin we put in, we would produce a square wave pulse, but in reality, we were unable to do this due to the limitations of the device. In this experiment we used a button switch to simulate a coin insert operation. Pressing the switch once may produce the signal shown in the following figure.

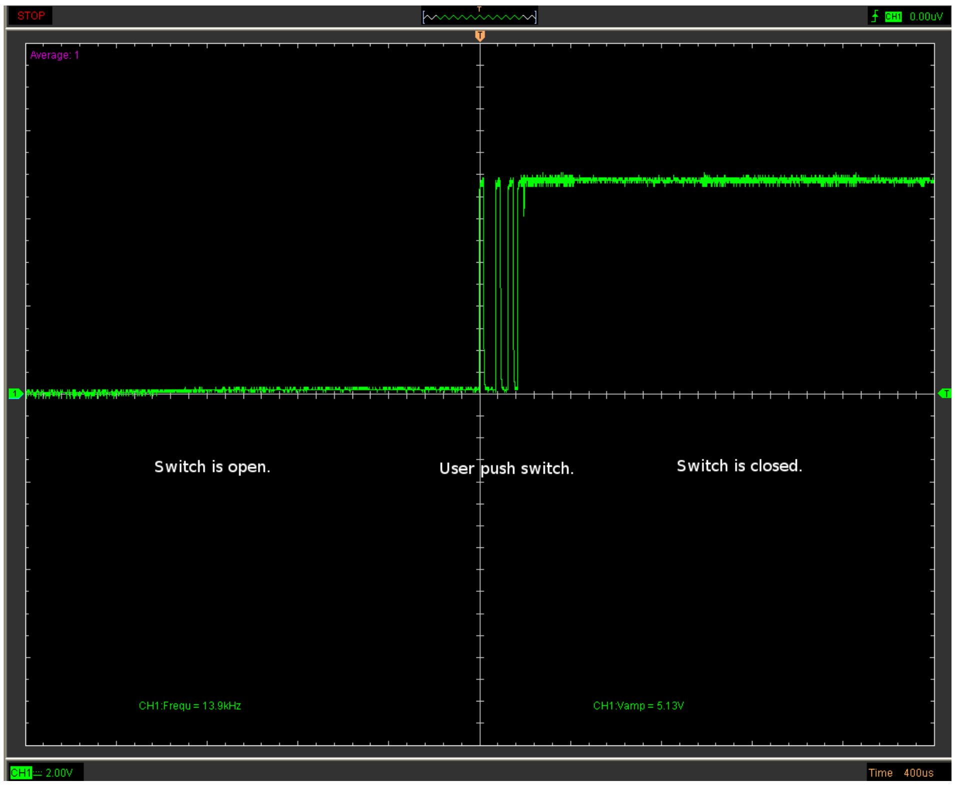


Figure 1 Switch Bounce

Note that there are multiple ups and downs in the waveform. Because the clock frequency is very fast, a single coin toss may be treated as multiple tosses without additional processing. This can take a toll on our customers, so we need to address this issue.

The intuitive idea is that when the signal changes, we should wait and see, and if it doesn't change again after a while, we think of it as a stable signal rather than a noise. On the contrary, if it changes while we wait, we dismiss it as a wobble and ignore the change

3. Detail

a) The algorithm can be divided into **Three** parts: **operator**, **controller** and **Debouncer**. This experiment requires you to use **more than three different modules** to implement this machine. **No data processing is allowed in the controller.**

b) The debouncer should be an additional component that can work independently. It will remove the jitter of the input signal without any other operations. In the absence of a debouncer, the vending machine will still work normally if the input signals are smooth and burr-free.

c) You need to use finite state machines (**FSM**) to implement controller. For finite state machines, you can find examples in Professor Ha's courseware. You can refer to the **provided code** to write the finite state machine as **two or three processes**.

d) The experiments in this course forces you to **separates the timing logic from the combinatorial logic**. **It is not acceptable to use clocks in the combinatorial logic.** **All timing logic can only be standard registers as provided.**

e) Draw state transition diagram and system block diagram before you start writing code. The debouncer does not need to appear on the system block diagram.

Code format

1. Keywords (except data type) are **all lowercase**.
2. The library names and data types are **all capitalized**.
3. Naming must be **meaningful** (except for the prescribed name). There are no restrictions on naming, and you can reasonably use uppercase letters and underscores to make your name easier to read and understand.
4. Use **four Spaces** or **a Tab** to indent your code. Reasonable indentation makes your code easier to read and understand.

First Submit

1. Compress the diagrams and **name the compressed package as follow the format**:

   Lab3_[YourNameInEnglish]_[YourStudentNumber].zip

2. You should submit:

   ■ State transition diagram

    - ◆ Draw it by computer rather than your hand

- ■ System block diagram/schematic
    - ◆ Draw it by computer rather than your hand

3. **File Organization Schema in Package**:

```
Lab3_[YourName]_[YourStudentNumber].zip
 ├──State_Transition_Diagram.pdf
 └──System_Block_Diagram.pdf
```

4. Upload the .zip file to related lab folder on BB.

Deadline of First Submit: 2020-10-21 23:00

Second Submit

1. Compress the HDL codes and **name the compressed package as follow the format**:

   Lab3_[YourNameInEnglish]_[YourStudentNumber].zip

2. You should submit:

- ■ Source codes
    - ◆ All of the source codes
- ■ Test bench codes
    - ◆ All of the test bench codes

3. **File Organization Schema in Package**:

```
Lab3_[YourName]_[YourStudentNumber].zip
    ├──VENDING_MACHINE.bit
    └──*.vhd
```

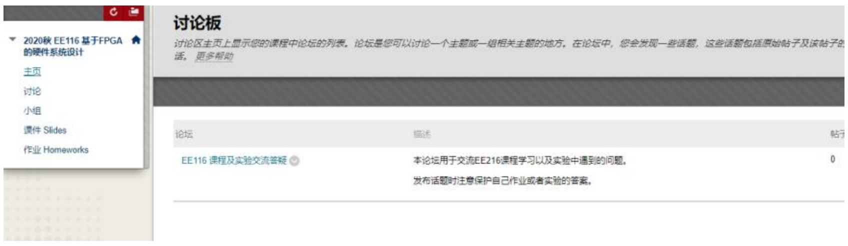
4. Upload the .zip file to related lab folder on BB.

Deadline of Second Submit: 2020-10-28 15:00

- Submit on time, get all scores.
- Submission time does not exceed 24 hours of the deadline, get half of the score.
- Submitted more than 24 hours from the deadline, get no score.

Any Question?

Any questions on course or labs can be proposed in the Discussing Forum on BB.



讨论板

讨论区主页上显示您的课程中论坛的列表。论坛是您可以讨论一个主题或一组相关主题的地方。在论坛中，您会发现一些话题，这些话题包括原始帖子及该帖子的
话。 更多帮助

论坛	描述	帖子
EE116 课程及实验交流答疑	本论坛用于交流EE216课程学习以及实验中遇到的问题。 发布话题时注意保护自己作业或者实验的答案。	0

We recommend you subscribe the forum to receive the newest topics on time.