Report for Lab2: RTL

20140843 Taekang Eom

20150384 Eunyoung Hyung

1. Introduction

In this lab, we need to implement a simple vending machine while understanding Finite State Machine(FSM) and Register-Transfer-Level(RTL) which is the design method to implement a synchronous circuit. To do this, we need to understand from what registers do and what combinational logic is to FSM and RTL.

2. Design & Implementation

Basically we have two combinational logics and one sequential logic. One of combinational logics is for the next states(current_total_nxt, num_items_nxt, num_coins_nx) and another one is for the outputs(o_available_item, o_output_item, o_return_coin). Sequential logic is for resetting and updating the states.

1) Sequential logic

When reset signal is on(precisely since 'reset_n' is active low, when reset_n is off) we just initialize as we do in initial block.

If it's off, then we need to update all states. In this part, we should care about the timer in the vending machine. We introduce the variable 'waiting'. When both i_input_coin and i_select_item are off, the vending machine is waiting for the signal within the time, so waiting is set to 1 and also the timer, 'waitTime' decreases. Otherwise, when one of two inputs is on, waiting is set to 0, so the timer is reset to kWaitTime.

2) Combinational logic for the next states

To calculate the current_total_nxt, we just need values current_total, input_total, and output_total while considering the returned coins. Calculating num_items_nxt and num_coins_nxt are just basic calculation.

As 2-1) says, we should set waiting depending on the input signals. When both i_input_coin and i_select_item are off, the vending machine is waiting for the signal within the time, so waiting is set to 1, otherwise, when one of two inputs is on, waiting is set to 0. In addition to them, we consider the case of i_trigger_return. When it is on, to make timer ignore and keep decreasing the count so end up with timeout, we set waiting 1.

3) Combinational logic for the outputs

We can easily show the o_available_item by using condition with current_total and the prices of the products.

And also we can decide the o_output_item by using o_available_item which we calculate above and i_select_item.

o_return_coin should be set when timer is out or the i_trigger_return is on. As I said in 2-2), when that signal is on, timer just ignores it and keeps decreasing the count so it will end up with timeout. Therefore, we just care the condition here only with the 'waitTime' to return changes.

3. Discussion

First, we implement machine that return change immediately after i_trigger_return is on, but the test was failed. However, we found the way which is to make timer ignore the input signal requesting changes from the user. Then it cause a timeout and return change successfully in the test.

4. Conclusion

We understand RTL is composed of the two big parts; one is about combinational logic and one is about registers which are memorizing values. Of course we see that how signals flow between each other and how it works.