

컴퓨터 구조 Lab1

20190782 컴퓨터공학과 최서영

20190439 컴퓨터공학과 오승훈

Introduction

이번 lab에서는 combinational logic과 sequential logic의 차이를 이해하고, 그를 이용한 간단한 FSM 예제인 vending machine을 design 해보고, ModelSim을 이용하여 RTL design을 직접 해보는 것을 목표로 하였다.

Design

이번 lab의 경우 조교님이 처음 제공해주신 vending machine skeleton에서는 총 vending_machine.v인 top module과 3개의 sub module로 구성이 되어 있었지만 state_change라는 모듈이 필요 없을 것이라고 판단을 하여, calculate_current_state 와 check_time_and_coin 으로만 구성을 하였다. 또한 state를 vending machine이 가지고 있는 금액으로 나누되 item의 가격과 반환될 수 있는 돈의 경우의 수인 111 ~ 000을 즉 1600 ~ 0원을 기준으로 8개의 state를, select_item을 기반으로 4개의 state를 설정하여 vending machine의 state를 나누는 것으로 하였다.

Top module에서는 skeleton code를 sub module로의 input, output wire들의 개수 변경을 제외하고는 변경 상황을 없도록 하였고, calculate_current_state 에서는 현재 금액이 얼마인지, 그 금액에 따라 available한 item은 무엇인지, item을 고르게 되면 그 item을 살 수 있다면 output 되는 item은 무엇인지, wait_time이 0이 되면 돈을 반환하고, i_trigger_return이 1이되면 돈을 반환하는 과정을 수행하는 module이다. 또한 check_time_and_coin 에서는 현재 wait_time을 계산을 해주는 과정을 수행하게 된다. clk마다 wait_time은 1씩 감소가 되며, input으로 받아오는 현재 돈을 기반으로 coin이 insert 되었거나, select한 아이템을 살 수 있는 금액 이상일 때 wait_time을 초기화시키는 동작을 수행하도록 design을 하였다.

Implementation

이번 lab에서는 vending machine을 구현해보는 것이 목표였다. 이를 위해 총 2가지 모듈로 나누어 계산을 하였다. 먼저, wait_time 값을 정의하는 check_time_and_coin 모듈과 실제로 out_total의 금액을 계산하고, out_total과 wait_time과 trigger에 따라 o_available_item, o_output_item, o_return_coin의 값을 정의하는 calculate_current_state 모듈이 존재한다.

- check_time_and_coin

input 값과 현재 가지고 있는 금액(out_total)에 따라 wait_time 값을 변환시켜준다.
input으로 coin이 들어왔으면 wait_time을 100초로 초기화 시켜준다. input으로

select_item이 들어왔다면 select한 item 가격과 out_total 가격을 비교하여 out_total이 더 많으면 wait_time을 100초로 초기화 시킨다.

또한, clk이 positive edge가 되면 wait_time을 1씩 뺀다. 만약 reset_n 값이 1로 들어왔다면 wait_time을 100초로 초기화 시킨다.

- calculate_current_state

input 값과 현재 가지고 있는 금액(out_total)에 따라 out_total, o_output_item, o_available_item, o_return_coin의 값을 정한다. input으로 coin이 들어왔으면 out_total에 해당 coin 값을 더해준다. input으로 select_item이 들어왔다면 select한 item 가격과 out_total 가격을 비교하여 out_total이 더 많으면 out_total에 해당 item 값을 빼주고, o_output_item값을 해당 select한 item 값으로 바꾸어준다.

Out_total 값을 이용하여 현재 고를 수 있는 item, o_available_item 값을 지정해 준다.

만약, wait_time이 0초가 되면 돈을 반환해준다. Out_total에 따라 o_return_coin 값을 정해주는데, o_return_coin 값 만큼 out_total에서 해당 값을 빼서 0원이 될 때까지 반환이 되도록 한다. i_trigger_return 값이 들어와도 같은 동작을 한다.

만약 reset_n 값이 들어온다면 o_available_item, o_output_item, out_total 모두 0으로 초기화 시켜준다.

Discussion

이번 lab이 처음으로 스켈레톤 코드를 받아서 하는 lab이어서 처음 스켈레톤 코드를 받고 해석을 할 때 어려움이 있었다. 또한 스켈레톤 코드에 맞추어 코드를 짜야 한다는 생각에 더 잘 안되는 어려움이 존재하였다. 그래서 팀원과 함께 코드를 덮어놓고 디자인을 하여 스켈레톤 코드를 변형시키는 등의 노력으로 lab을 진행할 수 있었다.

wait_time 값을 0보다 작아질 수 있도록 구현을 해줬는데 이때 wait_time이 0보다 작아지면 현재 reg는 unsigned로 해석을 하기 때문에 if(wait_time <= 0) 이라는 조건을 넣었을 때 unsigned로 해석을 하므로 0보다 작아지는 것이 아닌 MAX값이 되는 것을 고려하지 않아 구현에 차질이 있기도 했다.

Conclusion

결과적으로 vending machine 기능을 모두 구현할 수 있었다. 이를 테스트 코드를 통해 확인하였다.