

## Lab 2: Vending Machine Lab

20170616 정희진 20140727 이동현

### 1. Introduction

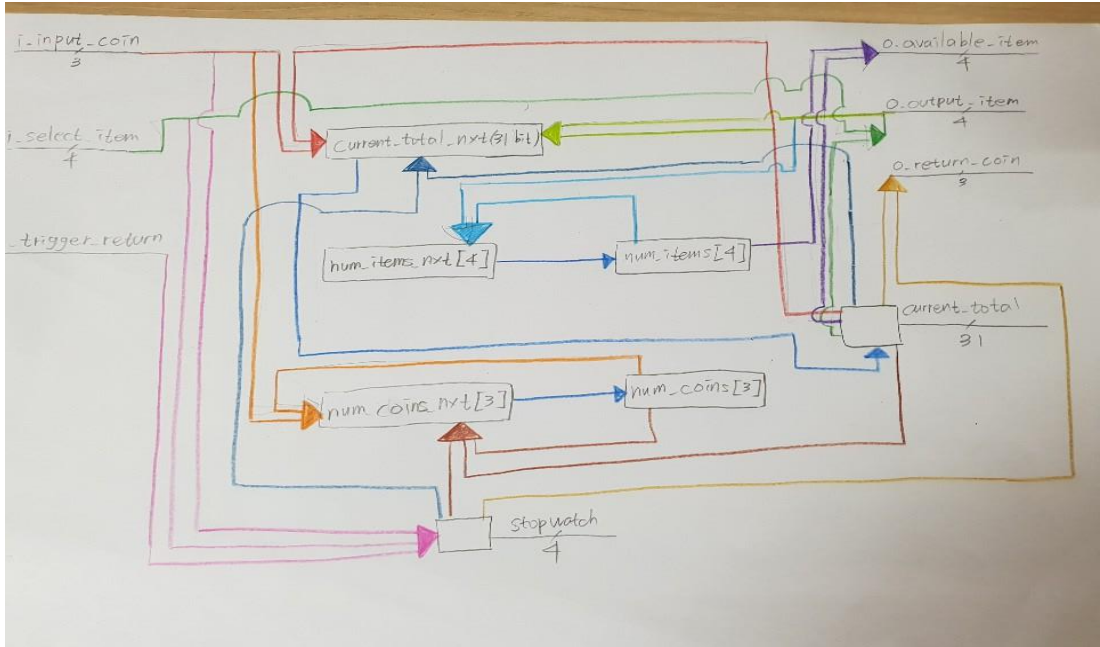
이번 lab2에서는 Verilog language를 이용해 vending machine을 만들어보았다. vending machine의 input에는 clk, reset\_n, i\_input\_coin, i\_trigger\_return으로 하였고 output은 o\_available\_item, o\_output\_item, o\_return\_coin, stopwatch, current\_total, return\_temp로 하였다. 각각의 설명은 다음과 같다.

input	
clk	testcase를 진행하면서 vending machine 안의 변수들이 synchronized 되도록 사용한다.
reset_n	vending machine(test)을 새로 시작할 때 사용한다; 0일 때 reset이 된다
i_input_coin (3 bit)	vending machine안에 coin을 넣을 때 1이 된다; i_input_coin[2]는 1000 coin, i_input_coin[1]은 500 coin, i_input_coin[0]은 100 coin을 나타낸다.
i_select_item (4 bit)	vending machine에 있는 item을 고를 때 1이 된다; i_select_item[3]은 2000 valued item, i_select_item[2]은 1000 valued item, i_select_item[1]은 500 valued item, i_select_item[0]은 400 valued item을 나타낸다.
i_trigger_return	거스름돈을 반환하고자 할 때 1이 된다.
output	
o_available_item (4 bit)	vending machine에 있는 item을 고를 수 있을 때 1이 된다; o_available_item[3]은 2000 valued item, o_available_item[2]은 1000 valued item, o_available_item[1]은 500 valued item, o_available_item[0]은 400 valued item을 나타낸다.
o_output_item (4 bit)	vending machine에 있는 item을 나올 때 1이 된다; o_output_item [3]은 2000 valued item, o_output_item[2]은 1000 valued item, o_output_item [1]은 500 valued item, o_output_item [0]은 400 valued item을 나타낸다.
o_return_coin (3 bit)	넣어준 돈을 반환하면 1이 된다. o_return_coin[2]은 1000, o_return_coin [1]은 500, o_return_coin[0]은 100을 나타낸다.
stopwatch	item을 뽑는 시간을 기다릴 때 사용된다..
current_total	현재 넣어준 coin의 총합을 나타낸다.

위의 input과 output을 이용하여 coin을 넣고 item을 선택하면 선택된 item이 나오고 거스름돈이 나오는 알고리즘을 구현해보았다.

## 2. Design

이번 랩에서는 input과 output, 그리고 state를 이용하여 코드를 작성해보았다. 각각의 변수들끼리의 연관관계를 아래와 같이 그림으로 그리면서 코드를 작성하였다.



(register은 네모로 표시하였고, reset=0인 경우는 그림에 넣지 않았다.)

연관관계를 코드로 작성하면서 always문을 많이 사용하였고, 특히 clock이 0에서 1로 올라갈 때마다 값을 바꿔주어야 하는 경우(ex. state, 그림에서 stopwatch와 연결된 모든 선들)도 따져보았다.

## 3. Implementation

vending machine에서 state는 총 3개로 만들었다. input된 동전의 총합을 세는 state(current\_total), 각각의 item개수(num\_items)와 coin의 개수(num\_coins)를 세는 state를 만들었다. 이 state들은 clock이 0에서 1로 올라갈 때마다 next state가 state가 되도록 설계하였다.

available item과 output된 item은 state에 따라(output item은 select item input도 고려해야한다) 다르게 결과값이 나오도록 설계하였다. 또한 state들도 위 그림에서처럼 다양하게 input과 output에 따라 결과값에 영향을 받는다.

stopwatch의 경우 coin을 넣거나 item을 고를 때마다 초기상태(7)로 set이 되도록 하였다. 하지만 거스름돈을 원하게 된다면 stopwatch가 0으로 된다. 그 외의 대기 상태는 clock이 한번 0에서 1로 올라갈 때마다 1씩 줄어들도록 코드를 만들었다. 그리고 stopwatch가 0이거나 그 이하일 때(vending\_machine.v에서는 stopwatch가 unsigned integer이기 때문에 최상위 bit가 1인 경우로 코드를 작성했다) 거스름돈을 반환하도록 작성하였다.

#### 4. Evaluation

아래와 같이 모든 testcase를 통과하였으며 속도도 빨랐다.

```
VSIM 2> run -all
# TEST                               InitialTest :
# PASSED
# TEST                               Insert100CoinTest :
# PASSED
# TEST                               Insert500CoinTest :
# PASSED
# TEST                               Insert1000CoinTest :
# PASSED
# TEST                               Select1stItemTest :
# PASSED
# TEST                               Select2ndItemTest :
# PASSED
# TEST                               Select3rdItemTest :
# PASSED
# TEST                               Select4thItemTest :
# PASSED
# TEST                               WaitReturnTest :
# PASSED
# TEST                               TriggerReturnTest :
# PASSED
# TEST                               ItemTestTest :
# PASSED
# TEST                               ItemTestTest :
# PASSED
# TEST                               ItemTestTest :
# PASSED
# Passed = 13, Failed = 0
- -
```

#### 5. Discussion

vending machine안에 coin의 개수가 제한되어 있음에도 불구하고 cointest가 없어 확인할 수 없었습니다. vending\_machine\_TB.v 파일 안에 cointest라고 명시가 되어있음에도 coin의 개수를 확인하지 않고 결과창에는 itemtest라고 나타났습니다.

input과 output, 그 외의 변수들의 설명이 자세하게 나타나있지 않아서 처음에 많이 어려웠습니다. classum을 통해 알 수 있었지만 pdf에 설명을 추가해주시면 어려움이 많이 없어질 것 같습니다.

#### 6. Conclusion

Verilog language의 문법을 익히고, clock과 state 개념을 이용해 코드를 작성하는 연습을 할 수 있었다.