

# Lab2: Vending Machine Lab

EE312 Computer Architecture

Professor: Minsoo Rhu

Student names: Maro Han, Sanghyun Kim

Student Numbers: 20150912, 20150146

## Introduction:

지난번 Lab1: ALU Lab 에서는 Verilog 언어의 기본적인 사용법에 대해 익혀보았다면, 이번 Lab2: Vending Machine Lab 에서는 Register-Transfer Level(RTL), synchronous circuit 그리고 finite state machine 같은 좀 더 응용된 개념들을 바탕으로 자판기를 구현하도록 한 것 같다. 이번 Lab2 에서, 우리 조는 입력 신호를 바탕으로 하드웨어 레지스터 사이에서 논리적 연산을 통해 실시간으로 동기화된 디지털 회로를 모델화하는 추상화 방식: Register-Transfer Level(RTL)을 통해 각각의 역할에 따라 3 가지 상태로 나누어 자판기의 각 기능을 구현하였다. 각각의 상태마다 하는 일은 아래에서 추가로 설명할 것이다.

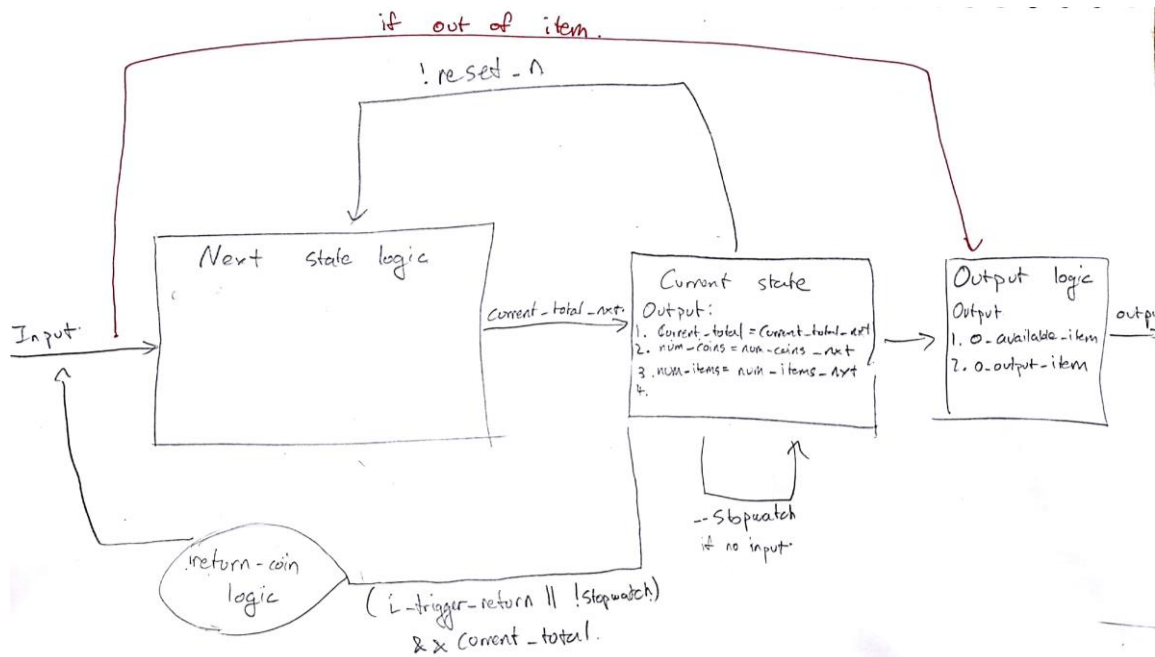
## Design:

자판기를 만들기 위한 각각의 상태에서 하는 일은 다음과 같다.

- State1 다음 상태를 위한 조합 논리 연산  
(Combinational logic for the next state)
- State2 어떤 상태를 초기화하거나 업데이트 하기 위한 순차 논리 회로  
(Sequential circuit to reset or update the states)
- State3 출력을 위한 조합 논리 연산  
(Combinational logic for the outputs)

또한, 지난 Lab1: ALU Lab 에서 배웠던 3 가지 Verilog Modeling Concepts 중에서 Behavioral Modeling 의 방식을 사용했다. State1, State3 은 입력 신호에 따른 조합 논리를 위한 것이고, State2 는 디지털 회로의 순차 논리를 위한 것이다.

우리 조의 자판기 디자인 figure 는 아래를 참조하면 나와있다.



Design Figure

### Implementation:

위의 디자인에 대한 설명을 살펴보면, always 구문을 이용하여 2 개의 조합 논리 1 개의 순차 논리, 총 3 개의 로직을 구현했다

각 상태에서 i\_input\_coin, i\_select\_item 입력을 통해 자판기 내 금액의 총합을 계산하거나(State1), State1 에서 다음 상태를 위해 레지스터에 임시로 저장해 놓았던 값들을 현재 값으로 업데이트 해주거나(State2), 각 TestBench 파일 내부의 함수에서 요구하는 출력들을(o\_output\_item, o\_available\_item) 계산한다(State3).

또한, stopwatch 라는 num\_items 라는 변수들이 3 가지 상태를 모두 넘나들며 10 초이상 입력이 없을 시 동전을 반환하면서 처음 상태로 돌아가는 역할이나, 자판기 내에 저장된 Item 의 갯수(이번 Vending Machine Lab 의 경우 10 개) 보다 많은 갯수를 출력하는 것을 탐지하는 역할을 한다.

마지막으로 i\_trigger\_return 의 입력, 반복적으로 on, off 되는 clk, 임시 저장 레지스터들을 잘 응용하여 주어진 기능을 모두 만족하는 자판기를 완성했다.

### Evaluation:

13 개의 모든 테스트 케이스를 통과했다.

```
# 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
```

Modelism 툴을 이용해 컴파일, 시뮬레이션을 돌렸다.

### Discussion:

도큐먼트에 자판기에 필요한 입력, 출력 값만 존재하고 그에 따른 기능에 대한 설명이 없어서 조금 아쉬웠다. 변수 이름을 보고 유추하거나, 코드에 적힌 주석들을 통해 자판기 내 아이템 갯수 제한(10 개)와 같은 내용들을 알 수 있었다.

### Conclusion:

이번 Lab2: Vending Machine Lab 을 통해 조합 논리(Combinational logic), 순차 논리(Sequential logic)를 응용해보는 경험을 가질 수 있었던 것 같다.

시뮬레이션을 직접 돌려보니 TestBench 내의 변수들의 변화를 쉽게 관찰할 수 있어 디버깅 하기 편리했던 것 같다.

