**2024 Spring CSED311 Lab 1  
Vending Machine Report**

Team ID: 67735

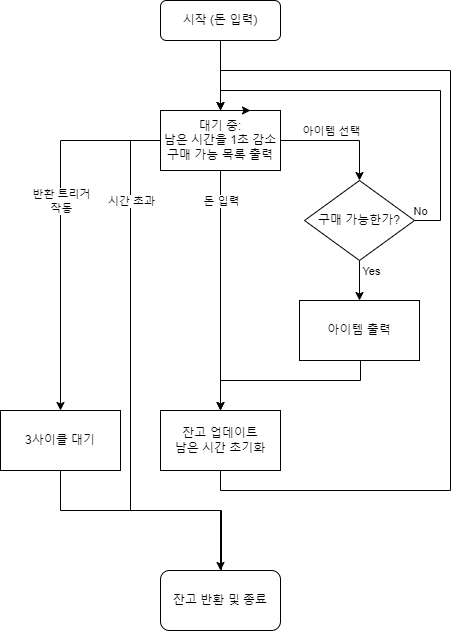
20220312 박준혁, 20220871 홍지우

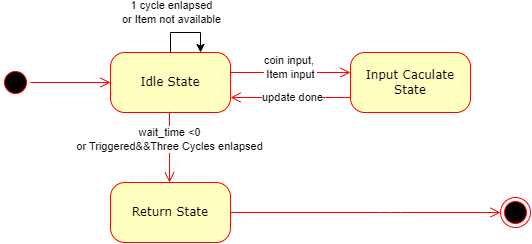
**명예서약 (Honor Code)**

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

1. **Introduction**
   1. 이번 lab에서는 간단한 finite state machine인 자판기를 구현하였다. 돈 입력, 아이템 고르기, 타이머, 동전 반환을 Moore Machine으로 구현하였다.
2. **Design**
   1. 순서도



* 1. State Transition Diagram  
     
  2. FSM Type

Moore Machine: Output depends on state only  
전부 계산하고 나서 clk에 맞추어서 업데이트하는 식으로 디자인 하였다.

* 1. Module Design

vending\_machine 안에 check\_time\_and\_coin, calculate\_current\_state, change\_state 3개의 모듈이 들어가 있다.

* + 1. check\_time\_and\_coin

시간을 동기적으로 1씩 줄인다.  
시간이 다 되거나 반환 트리거가 작동하면 잔돈을 반환한다.  
돈 입력이 있거나, 아이템 입력이 있고 구매 가능할 시 시간을 초기화한다.

* + 1. calculate\_current\_state

돈 입력, 아이템 입력, 반환 값들을 계산한다.  
또한 구매 가능 목록, 구매한 아이템을 출력한다.

* + 1. change\_state

바뀐 값들을 토대로 잔고를 업데이트한다.  
리셋된 경우 잔고를 0으로 만든다.

1. **Implementation**
   1. check\_time\_and\_coin
      1. Combinational Logic  
         wait\_time 이 0이 되거나 트리거되고 나서 3사이클 지나면  
         반환할 코인 값을 계산한다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* + 1. Sequential Logic

기본적으로 clk에 맞추어 시간을 1씩 감소시킨다.  
i\_input\_coin, i\_select\_item, reset\_n이 있으면 시간을 초기화한다.  
트리거가 되었을 때 시간을 3 흘려보내는 역할도 한다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

* 1. calculate\_current\_state
     1. Combinational Logic
        1. Caculate next values: input값에 따라 값을 계산한다.

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

* + - 1. Calculate Outputs: 구매한 아이템, 구매가능 아이템 출력

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

* 1. change\_state
     1. Sequential Logic: 리셋시 돈 초기화, 바뀐 잔고 업데이트

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

1. **Discussion**
   1. wait\_time 관련

check\_time\_and\_coin에서 wait\_time을 서로 다른 로직에서 접근하니까 에러가 발생하였다. 따라서 하나의 sequential logic 안에서 전부 처리해 주는 방식으로 구현하였다.

* 1. input\_total, output\_total, return\_total 관련  
     이 세가지 값은 원래 calculate\_current\_state의 output으로만 사용되게 skeleton code에 짜여 있었다. 그러나 이 값들을 사용하지 않자 warning이 계속 뜨는 문제가 있었다.  
     따라서 change\_state의 입력값으로 넣어준 뒤, change\_state에서 계산하는 방식으로 구현하였다.

1. **Conclusion**

이번 lab을 통해서 sequential logic, combinational logic과 <=, =의 차이(non-blocking, blocking)를 이해할 수 있게 되었다. 또한 FSM을 직접 구현하므로써 주어진 문제를 어떻게 설계하는 지에 대해서도 알아볼 수 있었다.