

컴퓨터구조 (CSED311)

Lab 1 Report

20200220 오상윤

(1) Introduction

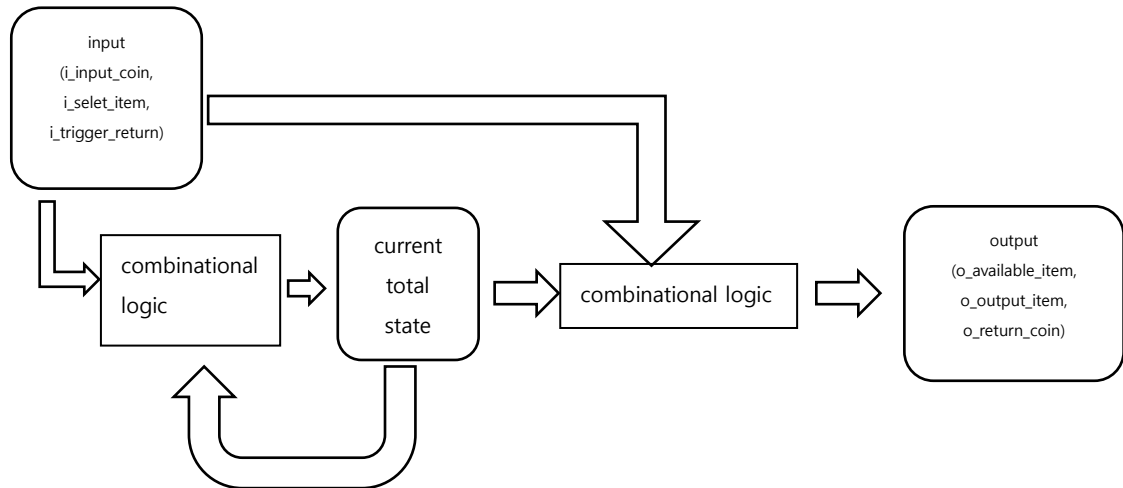
베릴로그를 이용하여 간단한 vending machine RTL을 구현해보았다. 주어진 Vending machine의 interface는 i_input_coin, i_select_item, i_return_trigger, clk, reset_n의 input들과 o_output_item, o_available_item, o_return_coin의 output들이 주어졌다. 즉, 동전의 종류와 개수, 선택된 아이템의 종류, return 버튼, 클럭 사이클이 입력으로 주어졌을 때, 구매가 가능한 item과 vending machine으로부터 dispensed된 item, 그리고 return 되는 동전이 출력되어야 한다.

전체적인 알고리즘은 다음과 같다.

1. Vending machine에 동전이 입력으로 들어오면, 동전의 type의 따라 i_input_coin의 값이 변화하고, waiting time이 100으로 초기화된다.
 2. Vending machine이 입력으로 들어온 동전을 포함하여, 현재까지 입력된 동전들의 합을 바탕으로 이용가능한 item들을 o_available_item을 통해 출력하며 waiting time을 1씩 감소시킨다.
 3. waiting time이 0이 되기 전에 돈이 들어오면 다시 1로 돌아간다.
 4. waiting time이 0이 되기 전에 이용가능한 item들 중 하나가 i_select_item을 통해 입력될 경우 item이 dispense 된다. waiting time을 100으로 초기화 하고 2로 돌아간다.
 - 4-1. i_select_item을 통해 입력된 item이 이용 가능하지 않을 경우 아무것도 하지 않는다.
 5. waiting time이 0이 될 때까지 input이 없는 경우 잔돈을 return 한다.
- i_return_trigger가 입력으로 들어올 경우 잔돈을 return하고 1로 돌아간다.

(2) Design

Output이 current state와 current inputs에 따라 결정되는 Mealy Machine을 이용하여 FSM을 디자인하였다.



*submodules

1. check_time_and_coin

coin 반환 시간을 update함과 동시에 현재까지 입력된 금액을 바탕으로 o_return_coin 값을 계산한다. input으로 clk, reset_n, i_input_coin, i_select_item, i_trigger_return을 받으며, i_input_coin이나 i_select_item 값이 바뀌면 입력값을 바탕으로 남은 돈을 계산하고 wait time을 초기화한다. i_trigger_return을 입력으로 받거나 wait time이 0인 경우 o_return_coin을 계산한다. wait time은 매 clk마다 1씩 줄어든다.

2. calculate_current_state

동전이 입력되었거나 item이 선택 되었을 때 current total state를 바탕으로 다음 current total state를 계산한다. input으로 i_input_coin, o_return_coin, i_select_item, item_price, coin_value, current_total, wait_time 을 받는다. total state를 0, 1, 2 세가지로 나누어 introduction의 알고리즘을 기준으로 1번을 0의 state, 2번을 1의 state, 4번을 2의 state로 나타내었으며, input과 state의 조합에 따라 적절하게 다음 state가 계산되도록 하였다. 또한 input 값이 변화할 때마다 현재 state에 따라 각각 input_total, return_total, 그리고 현재 입력된 돈을 계산하고 state에 맞는 output을 계산하도록 하였다.

3. change_state

clk가 positive edge일 때마다 input으로 들어온 reset_n 값을 확인하여 state를

current_total에서 current_total_nxt 로 변경할 지 결정하도록 하였다.

3) Implementation

1. check_time_and_coin (clk, reset_n, i_input_coin, i_select_item, i_trigger_return, wait_time, o_return_coin)

combinational logic을 이용하여 i_trigger return 값이 바뀔 때, wait time이 0 으로 설정되도록 하였다. 또한 moneySum이라는 reg를 생성하여 입력된 coin의 합을 계산하고, 특정 item이 선택되었을 때 moneySum과 해당 아이템의 가격을 비교하여 moneySum의 값을 변경하는 sequential logic 을 만들었다. 또한 o_return_coin을 계산하기 위해, wait time이 0이 되었을 때, moneySum 값을 확인하는 sequential logic 을 만들었다.

2. calculate_current_state (i_input_coin, i_select_item, item_price, coin_value, current_total, input_total, output_total, return_total, current_total_nxt, wait_time, o_return_coin, o_available_item, o_output_item)

next state를 계산하기 위해 combinational logic을 이용하여 i_input_coin과 current_total, i_seleted_item, wait_time의 값을 바탕으로 introduction의 알고리즘을 기준으로 1번을 0의 state, 2번을 1의 state, 4번을 2의 state라고 정하여 해당 state 값을 next state가 갖도록 하였다.

또한 현재의 current_total_state에 따라 1의 state인 경우 input_total과 return_total을 계산하고, o_available_item을 출력하도록 하고, 2의 state인 경우 선택한 item과 이용 가능한 item의 index가 겹치는 경우 해당 item을 o_output_item으로 출력한 다음 return_total을 계산하는 sequential logic을 구현하였다.

3. change_state

모든 state를 reset하는 경우 current_total을 0으로, 그렇지 않은 경우 current_total을 current_total_nxt로 바꿔주도록 하는 sequential circuit을 구현하였다.

4) Discussion

check_time_and_coin module을 구현하는 과정에서 wait_time을 지정할 때 바로

wait_time에 값을 넣어주었더니 일부 logic에서 의도치 않게 동작하는 경우가 많아 이를 방지하기 위해 wait_time에 시간을 지정하기 전 temp_wait_time이라는 임시 reg에 먼저 시간을 지정한 다음 posedge clk에서 temp_wait_time을 wait_time에 넣어주어 문제를 해결하였다.

5) Conclusion

Vending machine을 베릴로그를 이용하여 직접 설계하고 구현해보면서 module의 동작 방식, Moore Machine과 Mealy Machine의 차이, synchronous logic, asynchronous logic 등에 대한 이해를 높일 수 있었다. 또한 wire와 reg를 언제 사용하는지 그 차이점에 대해 알 수 있었다.