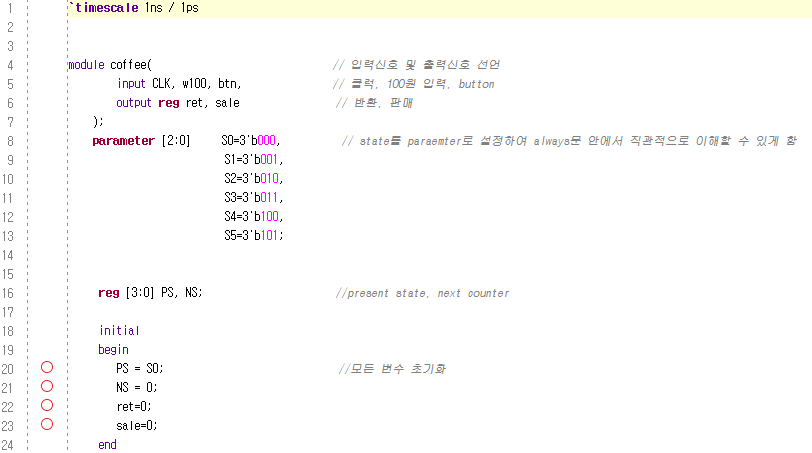
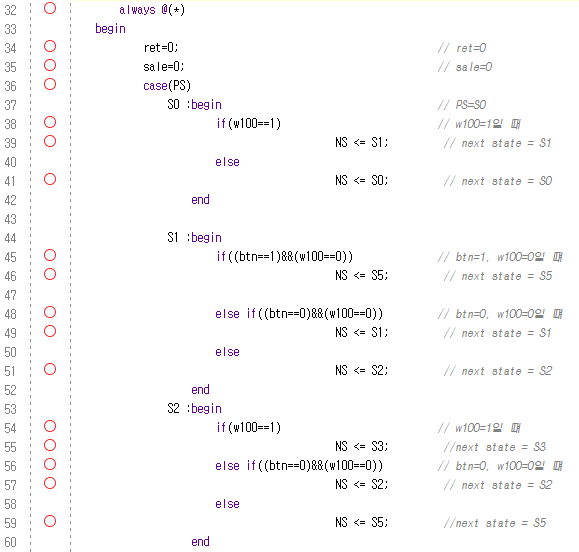
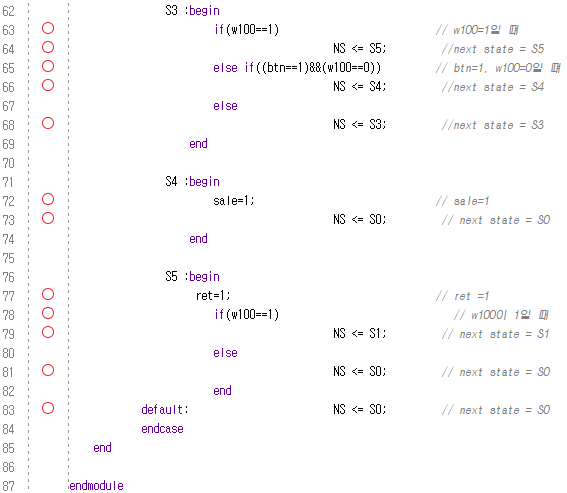
**Lab week7.**

# **Homework1. INVERTER1**

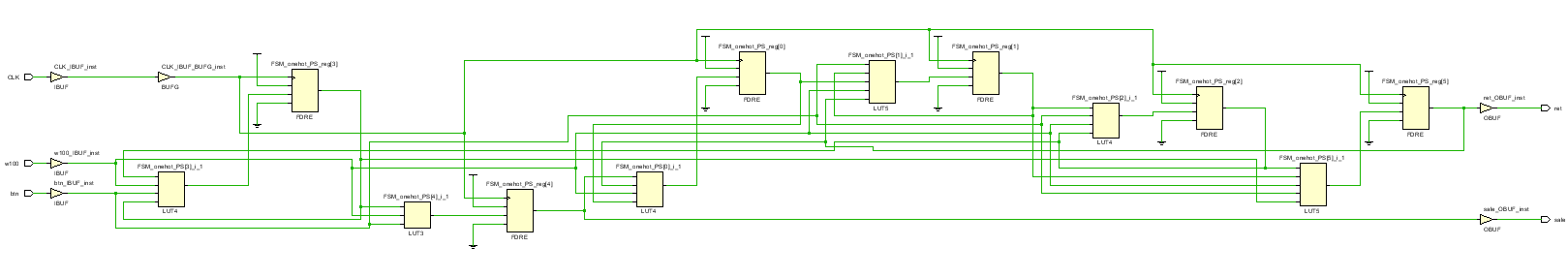
**Verilog Code / 주석**

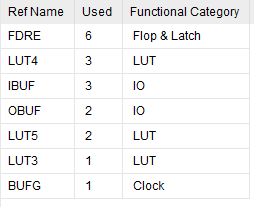
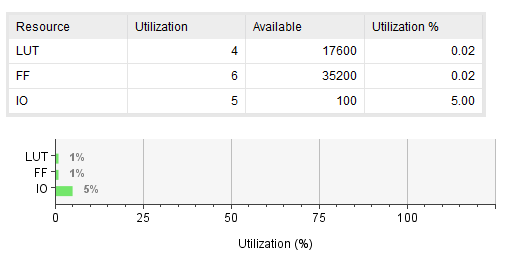






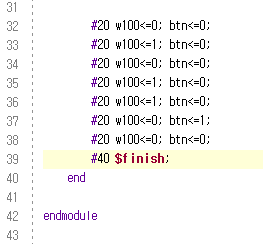
**Synthesis Report**

****

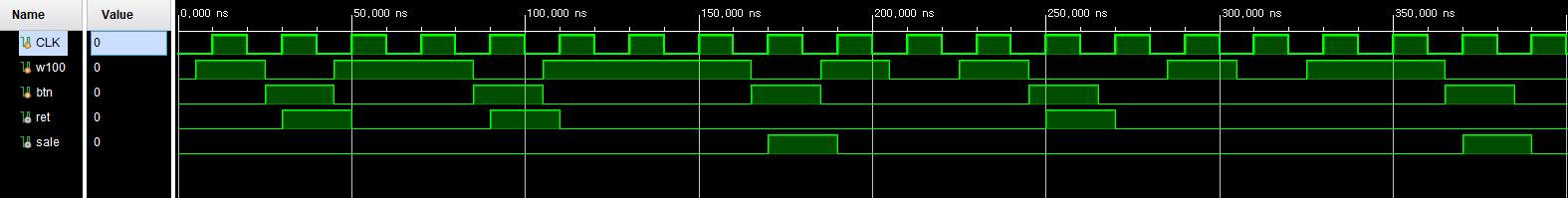
****

**Test Bench Code**





**Simulation Result**



|  |
| --- |
| **Discussion** |
| . - Verilog Coding을 시작하기 전 작성한 Block Diagram    - 작성한 Verilog Module에 대한 설명  위 module은 vending machine에의 동작을 나타낸 것이다.  Vending machine은 300원이 입력되었을 때 버튼을 누르면 sale 신호를 내보내고 나머지 100원, 200원에서 버튼을 누르면 동전을 반환하며 300원을 초과하는 입력에도 반환을 한다.  - 작성한 Code가 어떤 동작을 하는지? 왜 그런 동작을 하는지?  위 코드에서는2개의 always문을 사용하였다.  첫번째 always문은 state를 변경해준다.  두번째 always문은 입력에 따라 값을 출력한다.  S0에서는 w100의 입력이 들어오면 S1으로 이동한다.  S1에서는 W100의 입력이 들어오면 S2로 이동하고 btn이 1이고 w100이 눌리지 않았다면 S5로 이동하고 ret을 출력한다.  그 외의 경우에는 S2를 유지한다.  S2에서는 w100이 들어오면 S3로 이동하고 btn이 1이고 w100이 0이면 S5로 이동하고 ret를 출력한다.  S3에서는 w100이 들어오면 S5로 이동하게 되며 ret를 출력한다.  이때 btn이 1이고 w100이 0이면 S4로 이동하게 되며 Sale을 출력하게 된다.  S4에서는 S0로 이동한다.  S5에서는 w100이 0이면 S0로 이동하고 w100이 1이면 1으로 이동한다.  - 작성한 Test Bench Code가 어떤 동작을 하는지?  Test bench에서는 CLK, w100과 btn을 바꿔가며 동작을 확인한다.  - Simulation 파형이 왜 그렇게 나온것인지?  Simulation은 이론과 동일하게 나왔다.  Moore machine으로 구성하여 State가 정해지면 바로 출력이 발생할 수 있도록 coding하였고 동일하게 출력되는 것을 확인할 수 있었다.  - 작성한 Code가 잘 동작하지 않는다면 어디서 문제가 발생한 것인지?  문제가 발생하지 않았다.  - 오류를 해결했다면, 어떤 부분에서 해결을 했고, 왜 오류가 발생하였는지? |

문제가 발생하지 않았다.