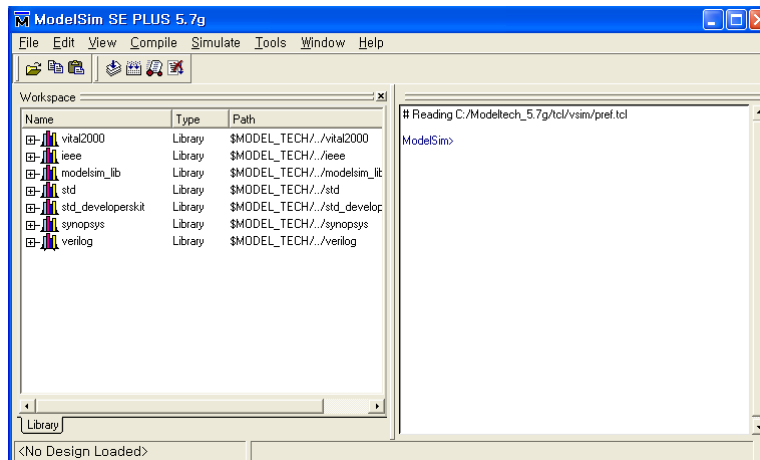


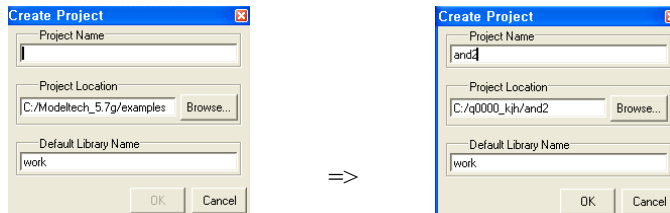
# ModelSim 기초 사용법

- 간단한 and gate 예제를 통해 ModelSim의 기초적인 사용법을 익힌다.

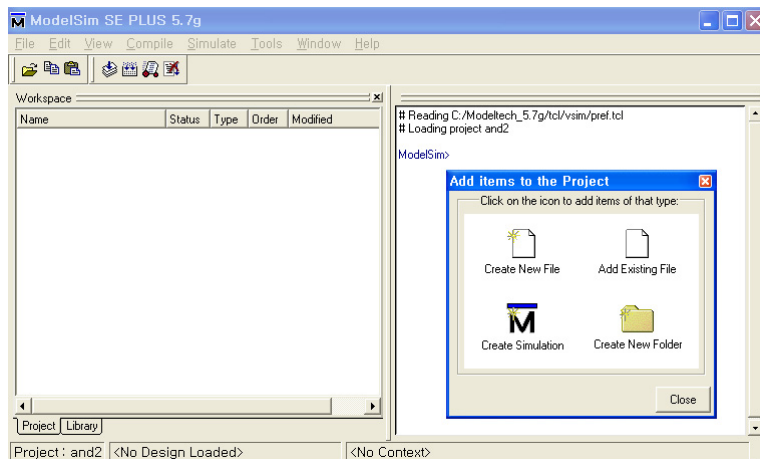
## 1. ModelSim을 실행.



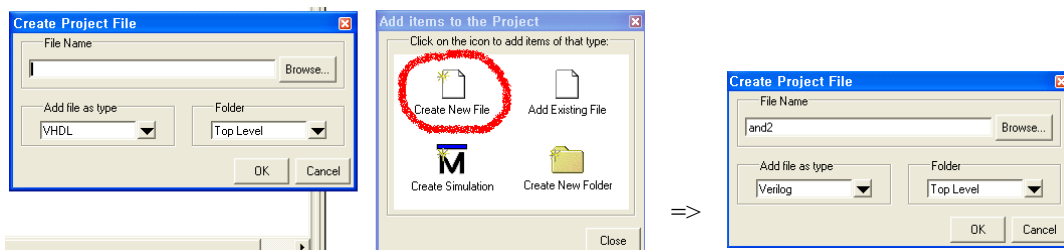
## 2. 프로젝트를 생성 : [File] => [New] => [Project]



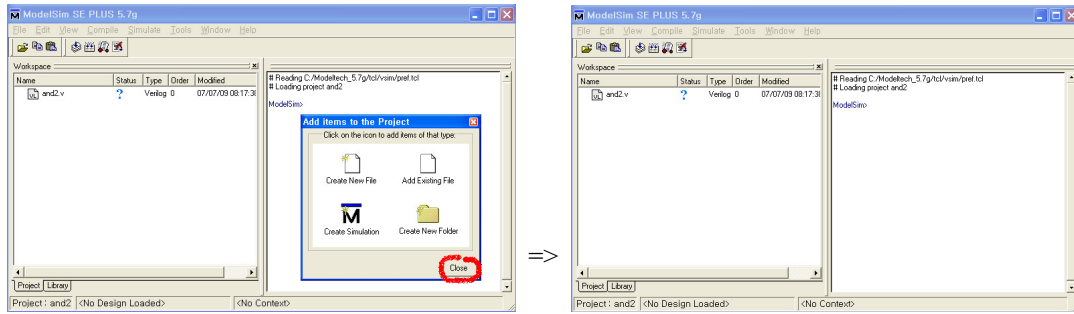
- [Project Name]에 적당한 이름을 입력. 여기선 "and2".
  - [Browse]를 눌러 “C:\q0000\_kjh\and2” 폴더 지정.
  - work는 ModelSim에서의 작업을 수행하기 위한 기본 폴더.
- => 위의 오른쪽 그림처럼 지정한 다음 [OK]를 누름.



## 3. “Create New File”을 누르면, “Create Project File” 창이 뜬.

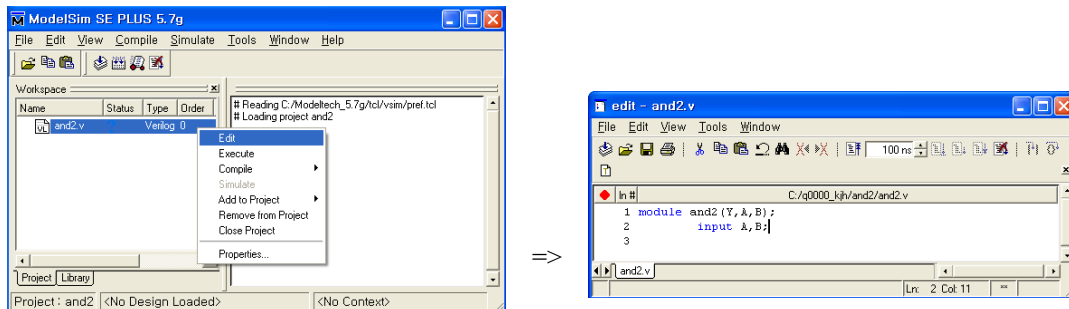


- [File Name]에 적당한 이름을 입력. 여기서는 "and2".
- [Add file as type]에는 Verilog를 이용할 것이므로 “Verilog”를 선택한 후, [OK] 누름.



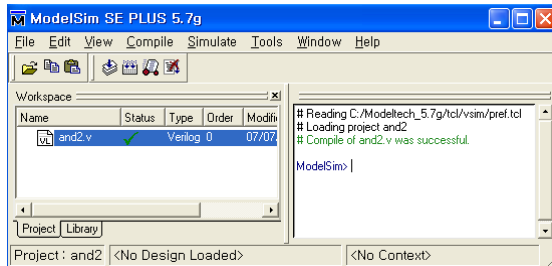
- “Close”를 누르면, 위의 오른쪽 그림과 같이 됨.
- 여기서 Project 탭 안에 “and2.v”라는 파일이 생성된 것을 볼 수 있음.

4. “and2.v”라는 파일을 선택한 뒤, 오른쪽 마우스 클릭을 하여 “Edit”를 선택.



- Edit 창이 나오면, 거기에 and gate에 대해 코딩함.

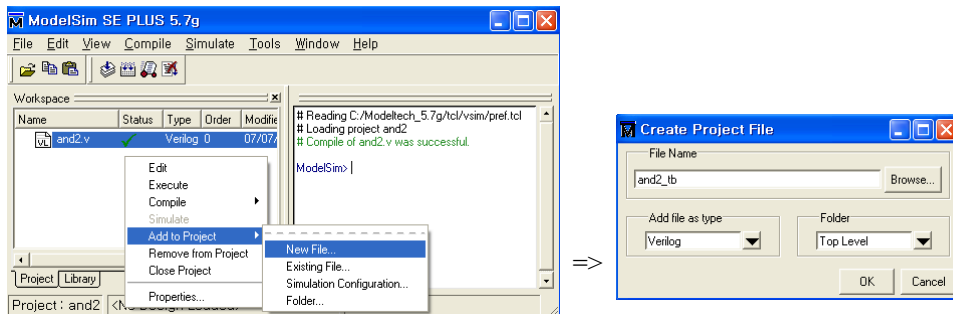
5. 코딩을 다하였다면, 저장을 하고 “compile”을 수행함.



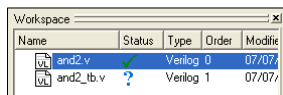
- 컴파일은 4번 그림에서 “compile”을 선택하거나, 위의 “workspace” 상단의 를 클릭.
- compile이 성공적으로 수행되었다면, 위와 같이 Status의 “?”가 “√”로 바뀜.

6. Compile이 성공적으로 끝나면, test bench를 작성.

- "workspace"에서 오른쪽 마우스 클릭을 하여 [Add to Project] => [New File...] 선택.



- 위 오른쪽 창에서 그림과 같이 선택하고 [OK].



- 4번의 방법대로 다음을 코딩.

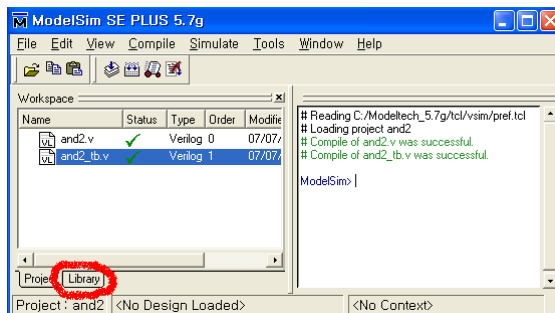
```

module and2_tb; // test bench
    wire y_out;
    reg a_in;
    reg b_in;
    and2 U0( .Y (y_out), .A (a_in), .B (b_in) );
    initial begin //test vector
        a_in = 'b0; b_in = 'b0;
        #10
        a_in = 'b0; b_in = 'b1;
        #10
        a_in = 'b1; b_in = 'b0;
        #10
        a_in = 'b1; b_in = 'b1;
    end endmodule

module and2(Y, A, B);
    input A, B;
    output Y;
    and (Y, A, B);
endmodule

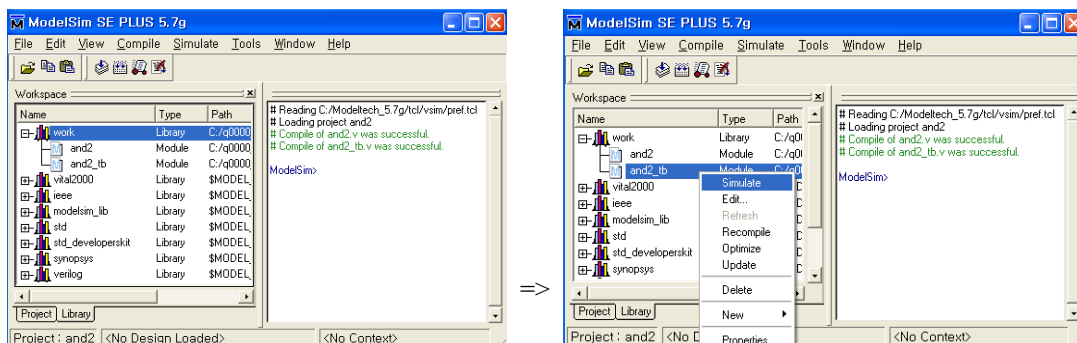
```

- 코딩을 마쳤으면, 위와 동일한 방법으로 compile함. 제대로 compile되면 다음과 같이 됨.

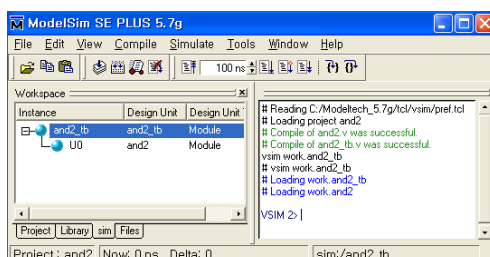


## 7. Test bench까지 compile 되었으며, 시뮬레이션을 해보자.

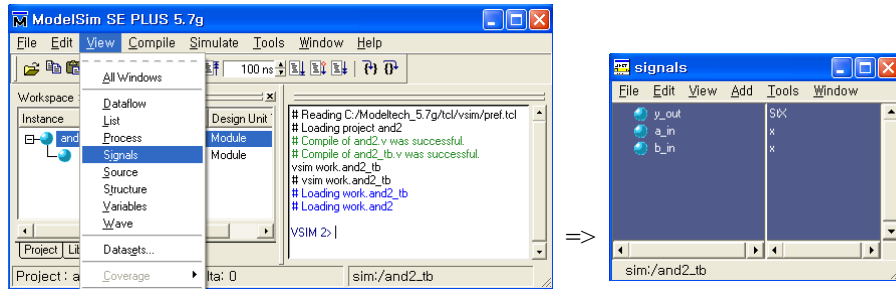
- 위 그림에서 “Library” 탭을 클릭하면,



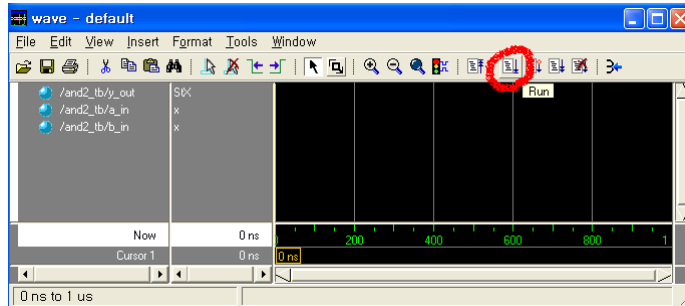
- “work”라는 디렉토리에서 우리가 작성한 파일을 확인 할 수 있음.
- 위 오른쪽과 같이 “and2\_tb.v”파일을 클릭하고 오른쪽 마우스 클릭하여 “Simulate” 선택.



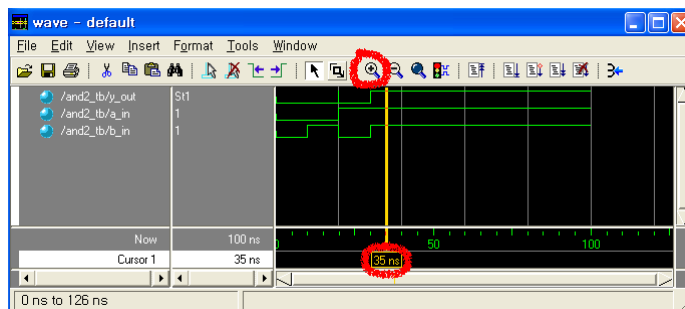
- 그러면 위와 같은 화면을 볼 수 있고, 이제는 wave를 보면 된다.
- [View] => [Signals]을 선택하거나, 터미널에 "view signals"라고 치면 됨.



- [Add] => [Wave] => [Signals in Region]을 선택하여 signal들을 wave에 추가.
- 그러면 아래와 같은 wave 창을 볼 수 있음.



- 이제 "Run"버튼을 누르면 됨.



- 위 그림은 Run 후에 확대한 그림.

- The End -

- \* **test bench** : 설계한 모듈(칩)이 실제 환경에서 어떻게 동작할 것인지를 test하기 위한 가상 환경. 즉, 설계된 Verilog HDL 코드에 테스트 입력을 적용하기 위해서 정의 됨. 이를 통하여 설계가 맞는지 검사.  
테스트벤치는 외부와 입출력을 하지 않고 자체 내에서 입력 파형을 제공.