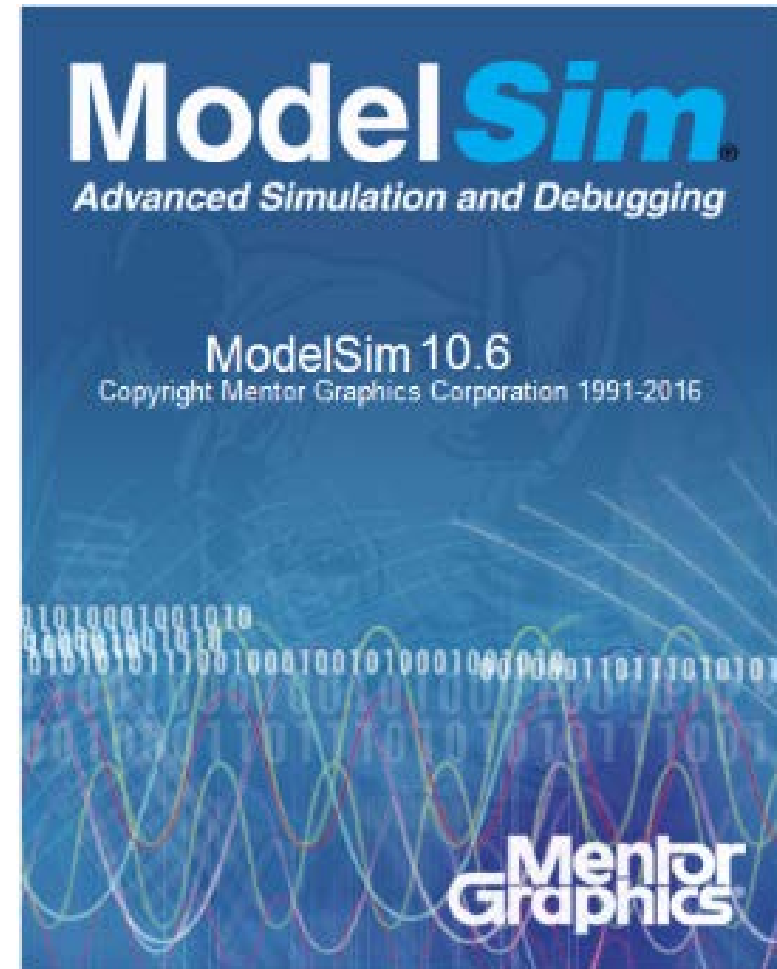


ModelSim 튜토리얼

ModelSim

- Mentor Graphics 의 HDL Simulator 인 ModelSim은 ASIC/FPGA 개발 및 디버깅 환경을 지원
- Verilog, VHDL, SystemC, SystemVerilog, Mixed Language 를 지원하는 통합 디버깅 환경 제공



ModelSim 설치(1/2)

- https://www.mentor.com/company/higher_ed/modelsim-student-edition

Download Student Edition

First Name •
Uijeong

Last Name •
Sang

Email •
syj5982@gmail.com
A valid email address is required.

Phone •
01051338947

Company •
Kyunghee univ

Primary job function •
Student

Address 1 •
ero, Giheung-gu, Yongin-si, Gyeonggi-do 17104

Address 2

Country •
KOREA REPUBLIC OF

Please Provide Your City •
suwon

Submit

I Accept This Agreement

- 설치파일 다운로드 후 실행

ModelSim 설치(2/2)

- 설치 중 아래 페이지에 정보를 입력하면 입력한 메일로 라이선스 파일이 전송

ModelSim PE Student Edition – License Request

Please complete the form below to have a license file emailed to you.

First Name *	Last Name *
<input type="text"/>	<input type="text"/>
Email *	Phone * (No Dashes or Spaces)
<input type="text"/>	<input type="text"/>
Email (Please Re-enter your email) *	Please verify your email is correct, as the ModelSim Student Edition license file will be emailed to you.
<input type="text"/>	
Address *	Address 2
<input type="text"/>	<input type="text"/>
City *	State/Province (US or Canada Only)
<input type="text"/>	<input type="text"/>
Country *	Zip/PostCode *
<input type="text" value="UNITED STATES"/>	<input type="text"/>

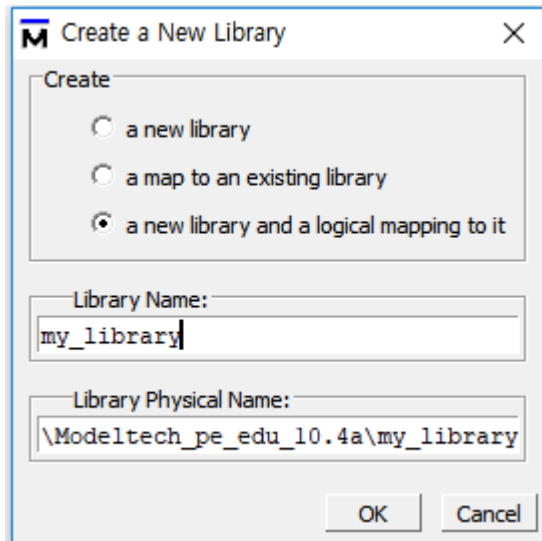
- 메일로 받은 라이선스 파일을 설치경로(C:\WModeltech_pe_edu_10.4a)에 복사

ModelSim 설치가 안되는 경우

- 다운로드받은 설치파일(.exe)을 압축해제 후 Disk1 폴더의 Setup.exe를 직접 실행
- 그래도 안되는 경우 아래 링크 참고
<http://vcodecs.com/display/DD/00.+ModelSim+on+Windows+10>

ModelSim 라이브러리 생성

- File->New->Library 선택



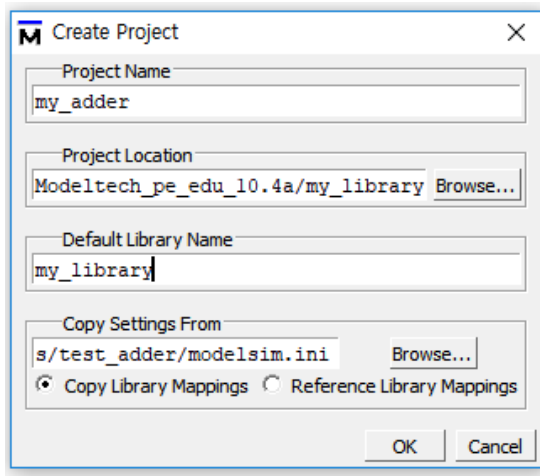
- 라이브러리 이름 입력 후 OK



- 라이브러리 생성

ModelSim 프로젝트 생성

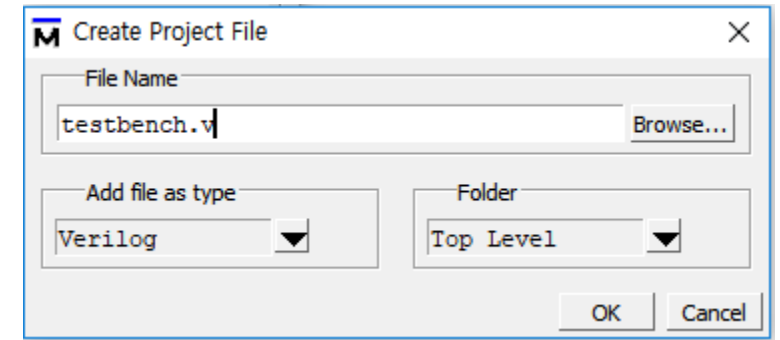
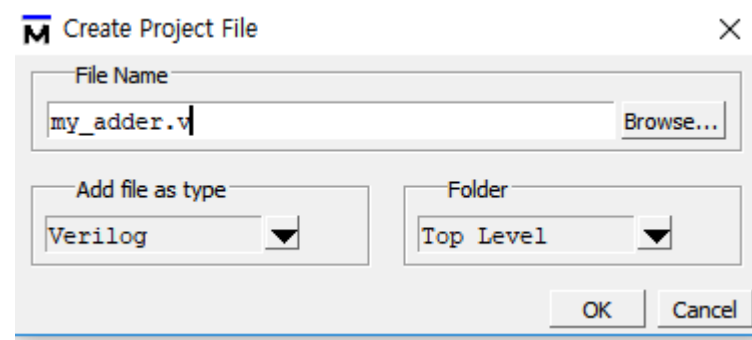
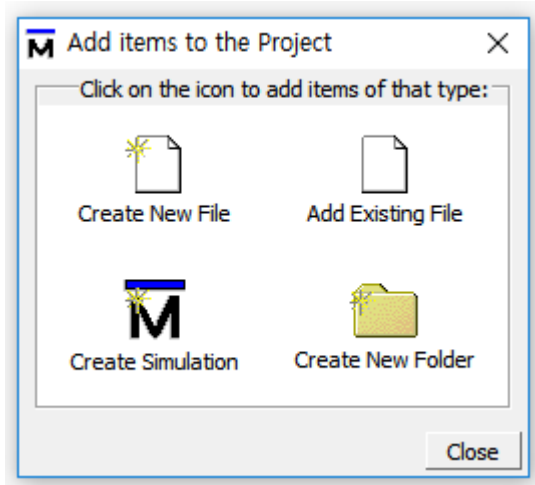
- File->New->Project 선택



- 프로젝트 이름, 경로, 라이브러리 이름 입력 후 OK
- 프로젝트를 다시 열 경우 File->Open에서 Project Files 선택 후
오픈

프로젝트에 verilog파일 생성

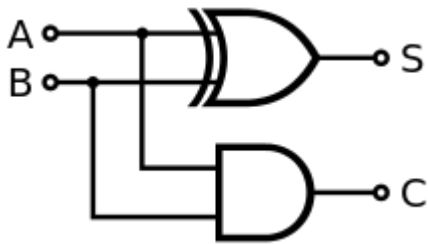
- Create New File 선택



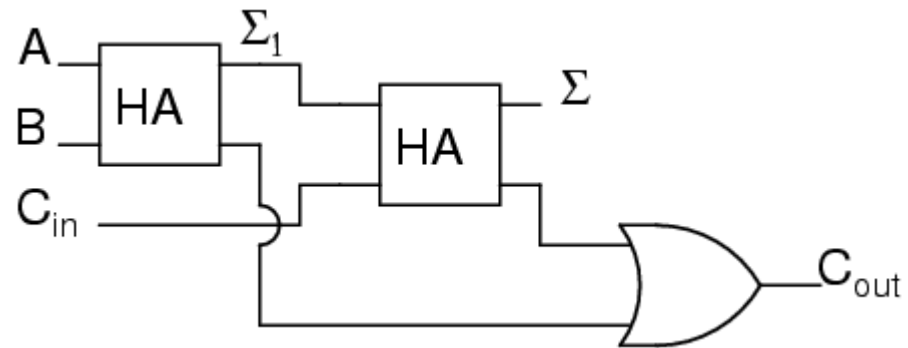
- 파일 이름 입력, Add file as type Verilog 선택 후 OK
- 예시에서는 my_adder.v, half_adder.v, testbench.v 세 파일 생성

Verilog를 이용한 Full Adder 구현

- Verilog를 사용하여 Half Adder를 구현
- Half Adder를 사용하여 Full Adder를 구현
- 구현된 Full Adder를 testbench를 통해 시뮬레이션



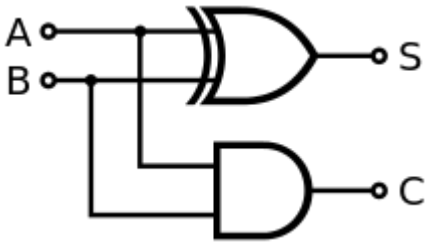
Half Adder



Full Adder

Half Adder 구현

- Half_adder.v 파일 오픈
- Half Adder의 회로를 아래와 같이 Verilog로 구현
모듈명은 파일명과 동일



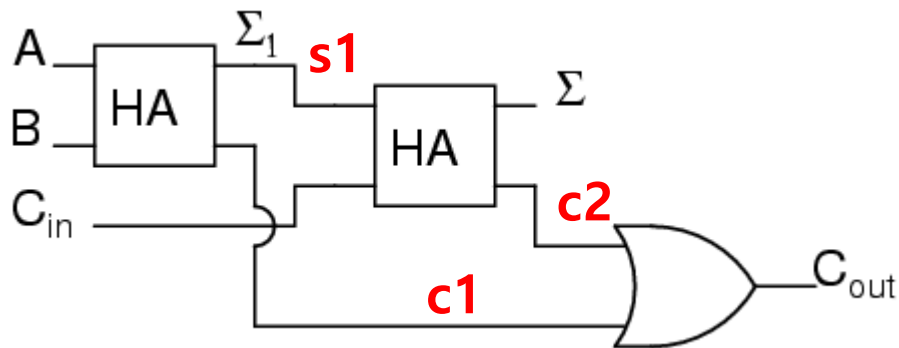
Half Adder

```
module half_adder(a,b,s,c);  
  input a,b;      입력 변수 선언  
  output s,c;     출력 변수 선언  
  xor(s,a,b);     XOR gate 생성  
  and(c,a,b);     AND gate 생성  
endmodule
```

Full Adder 구현

- my_adder.v 파일 오픈
- 앞서 구현한 Half Adder의 인스턴스를 활용하여 구현

모듈명은 파일명과 동일



Full Adder

```
module my_adder(a,b,cin,s,cout);  
  input a,b,cin;  
  output s,cout;  
  wire s1,c1,c2; 결과 값을 연결해주는 변수
```

```
  half_adder ha1(a,b,s1,c1);  
  half_adder ha2(cin,s1,s,c2);
```

```
  or(cout,c1,c2); 앞서 생성한 half_adder의 인스턴스  
endmodule 생성
```

Testbench 구현

- testbench.v 파일 오픈
- 구현한 Full adder의 시뮬레이션을 위한 코드 작성
- Input의 값을 시간에 따라 변경하는 코드를 작성

작성한 모듈의 시간 단위 정의
`timescale 시간단위/오차범위
`timescale 100ps/1ps

```
module tsetbench;
  wire t_s, t_cout;   wire : output port로 값을 넣을 시그널
  reg t_a, t_b, t_cin; reg : input port로 값을 넣을 시그널

  my_adder fa(t_a,t_b,t_cin,t_s,t_cout);
                                     my_adder의 인스턴스 생성

  initial
  begin
    t_a = 0;
    t_b = 0;
    t_cin = 0;

    #5                               #5 정의된 시간의 배수로 시간 흐름을 정의
    t_a = 1; 예시의 경우 단위가 100ps임으로 500ps의 흐름을
                                     나타냄

    #5
    t_b = 1;

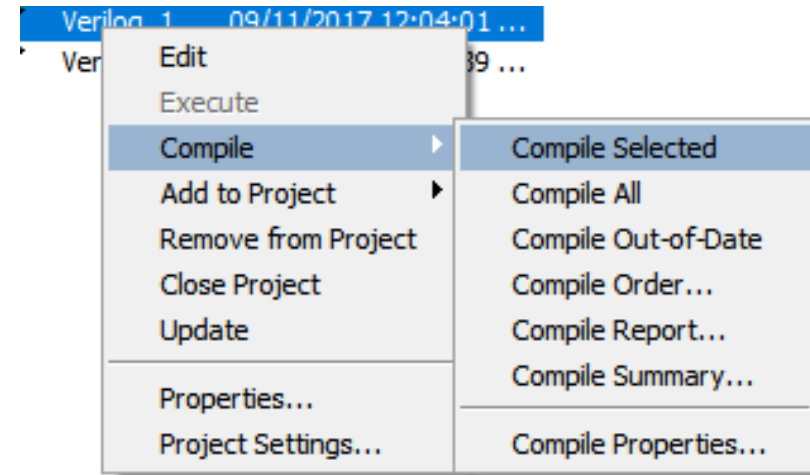
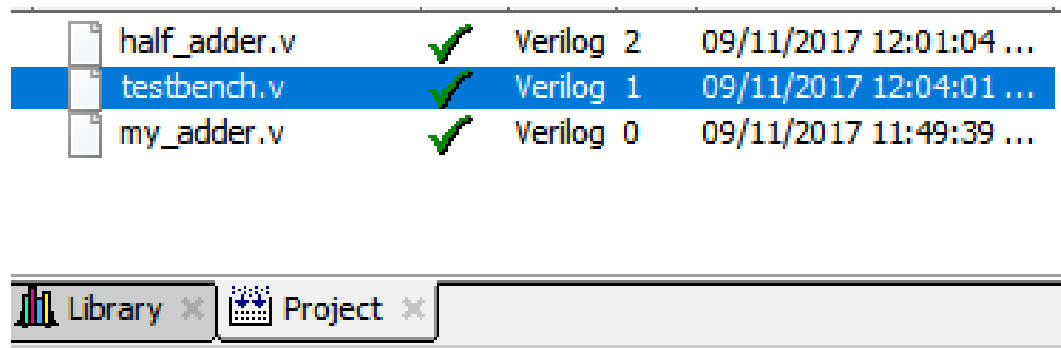
    #5
    t_cin = 1;

  end

endmodule
```

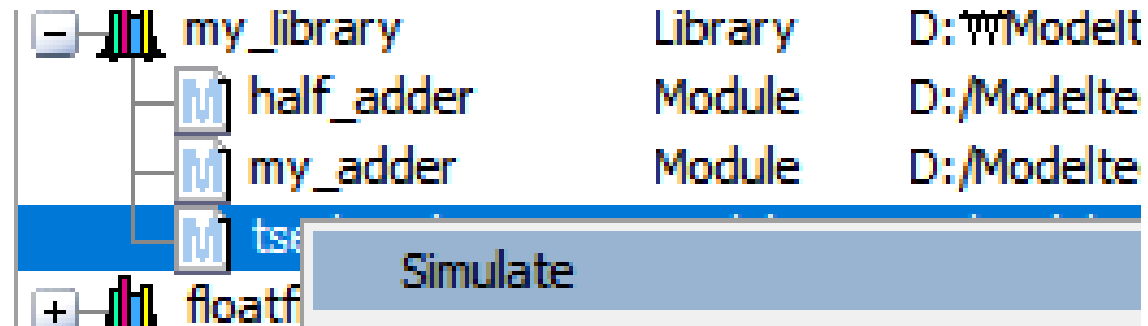
컴파일

- 프로젝트 탭의 Verilog파일들을 half_adder, my_adder, testbench의 순서로 오른쪽 클릭 후 컴파일



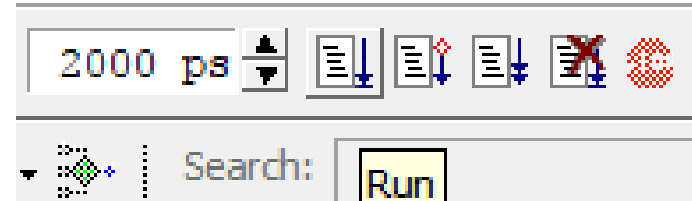
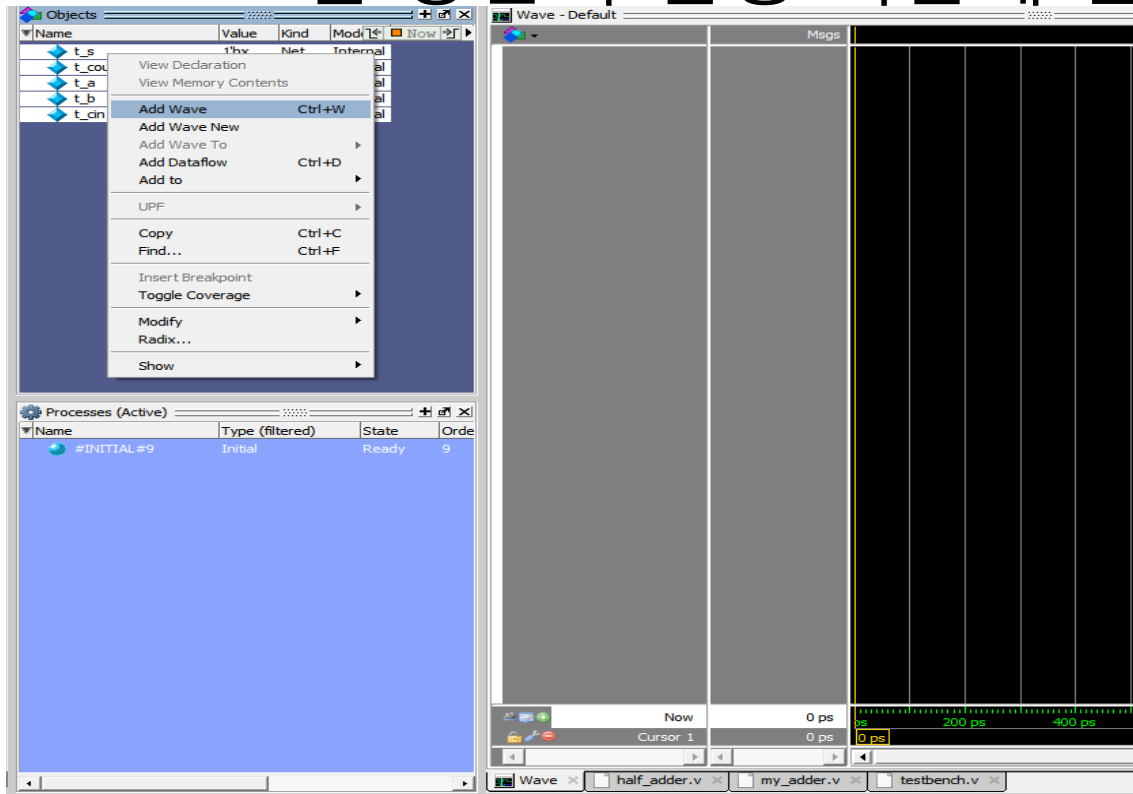
Full Adder 시뮬레이션(1/3)

- Library 탭에서 my_library 클릭 후 testbench를 오른쪽 클릭하여 Simulate 실행



Full Adder 시뮬레이션(2/3)

- Simulate 실행 후 아래 그림과 같이 값을 확인할 object에 대해 Add Wave 클릭
- 프로그램 상단의 실행 버튼에 원하는 수행시간 입력 후 Run



Full Adder 시뮬레이션(3/3)

- 시뮬레이션 실행 후 아래와 같이 작성한 Full Adder가 동작하는 것을 확인

