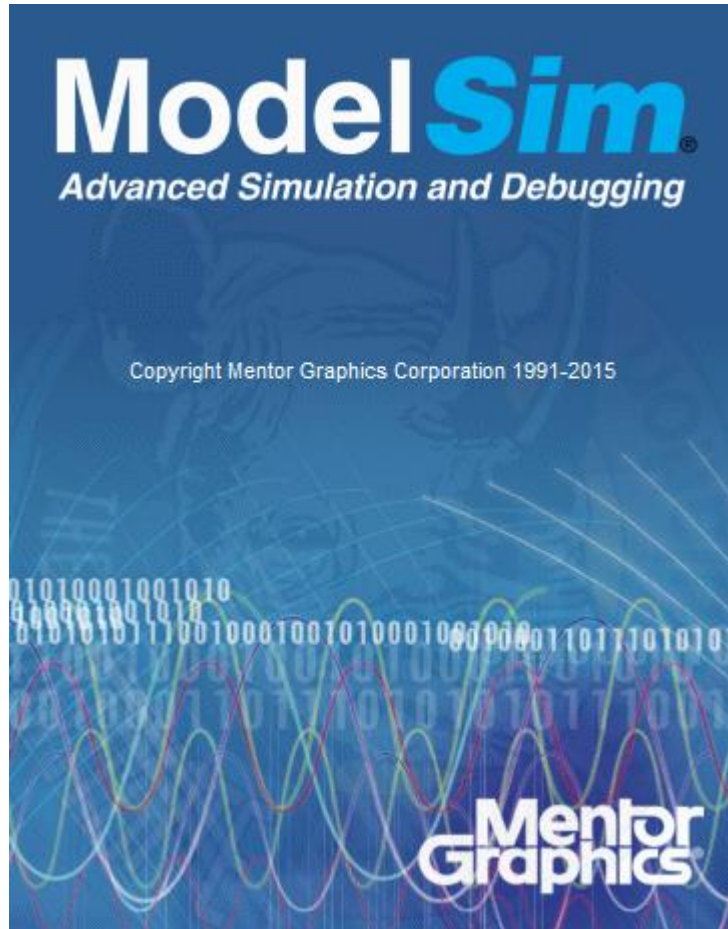


ModelSim-0



**2018 Spring
Media System Lab, Yonsei Univ.**



ModelSim

Beginner's Guide

0. ModelSim 설치

- ❑ MentorGraphics 홈페이지에서 ModelSim PE Student Edition 다운로드
 - www.mentor.com/company/higher_ed/modelsim-student-edition
 - Student Edition이지만 학교 도메인 (@yonsei.ac.kr) 이메일 주소가 필요하지는 않음

- ❑ ModelSim 설치 시 주의사항
 - 반드시 Windows 관리자 계정에서 설치할 것 (Mac/Linux 지원 X)
 - 일반 관리자 계정에서 정상 동작하지 않을 경우 Administrator 계정 사용 (Net user administrator /active:yes 커맨드로 활성화)
 - 불가피한 이유가 없다면 기본 경로에 설치 -> 관리자 권한으로 설치할 것 (C:\WModeltech_pe_edu_10.4a)
 - ModelSim 설치 이후 뜨는 팝업 인터넷 창에서 이메일을 등록해야 student_license.dat 파일을 전달 받을 수 있음
URL 주소가 토큰 기반이기 때문에 실수로 인터넷 창을 껐다면 ModelSim 삭제 후 재설치 해야함! (닫은 탭 다시 열기 X)
 - 라이선스 파일은 ModelSim을 설치한 Window 계정에 의존적임!
 - 설치/실행은 되도 Simulate가 되지 않으면 Administrator에서 재설치할 것 (라이선스 파일 mismatch거나 access 오류임)

Contents

1. ModelSim 실행

2. New Project 만들기

3. Verilog 코드 파일 추가

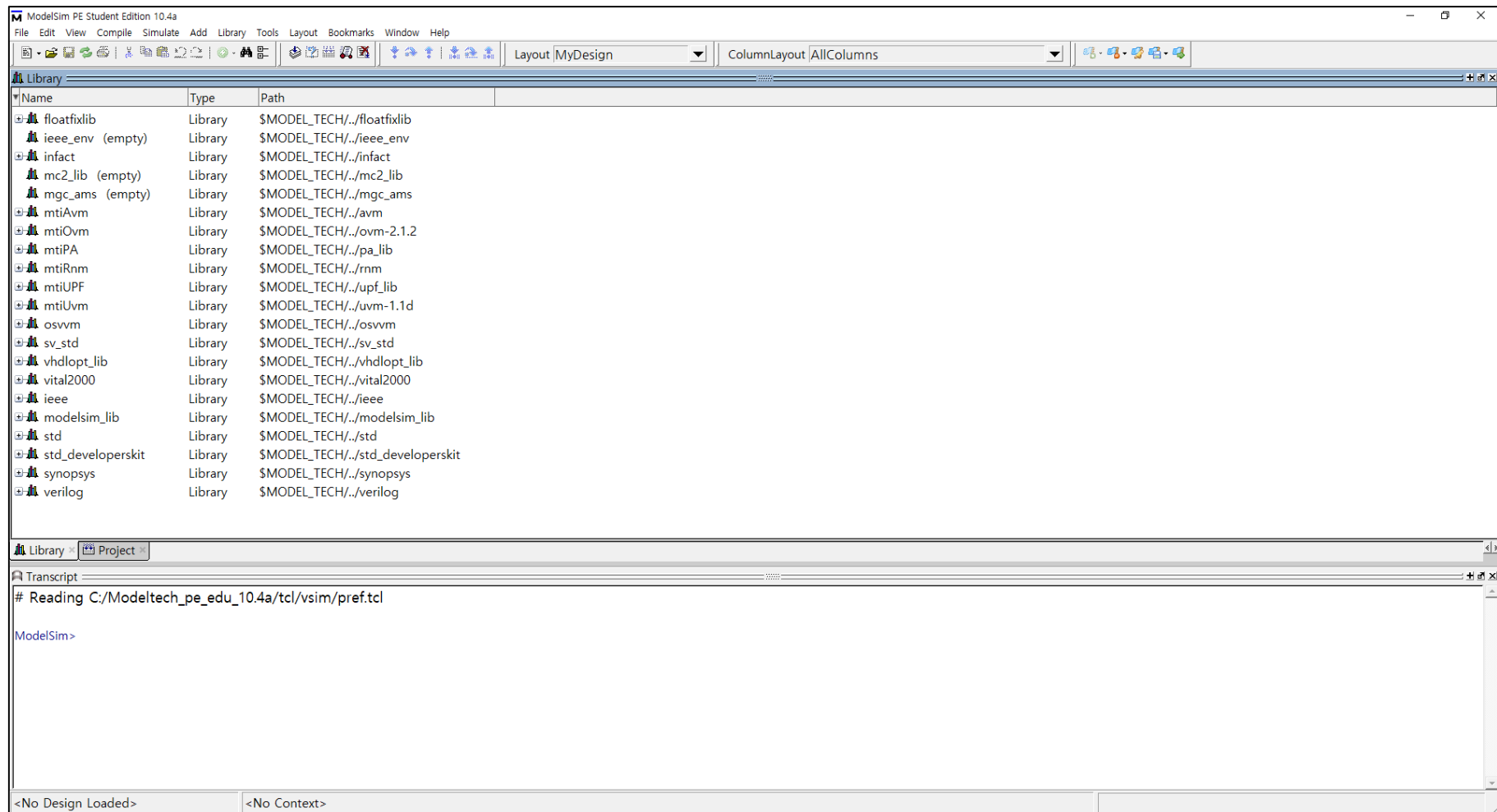
4. 코드 Compile

5. Simulate 환경 설정

6. Simulation 실행 결과

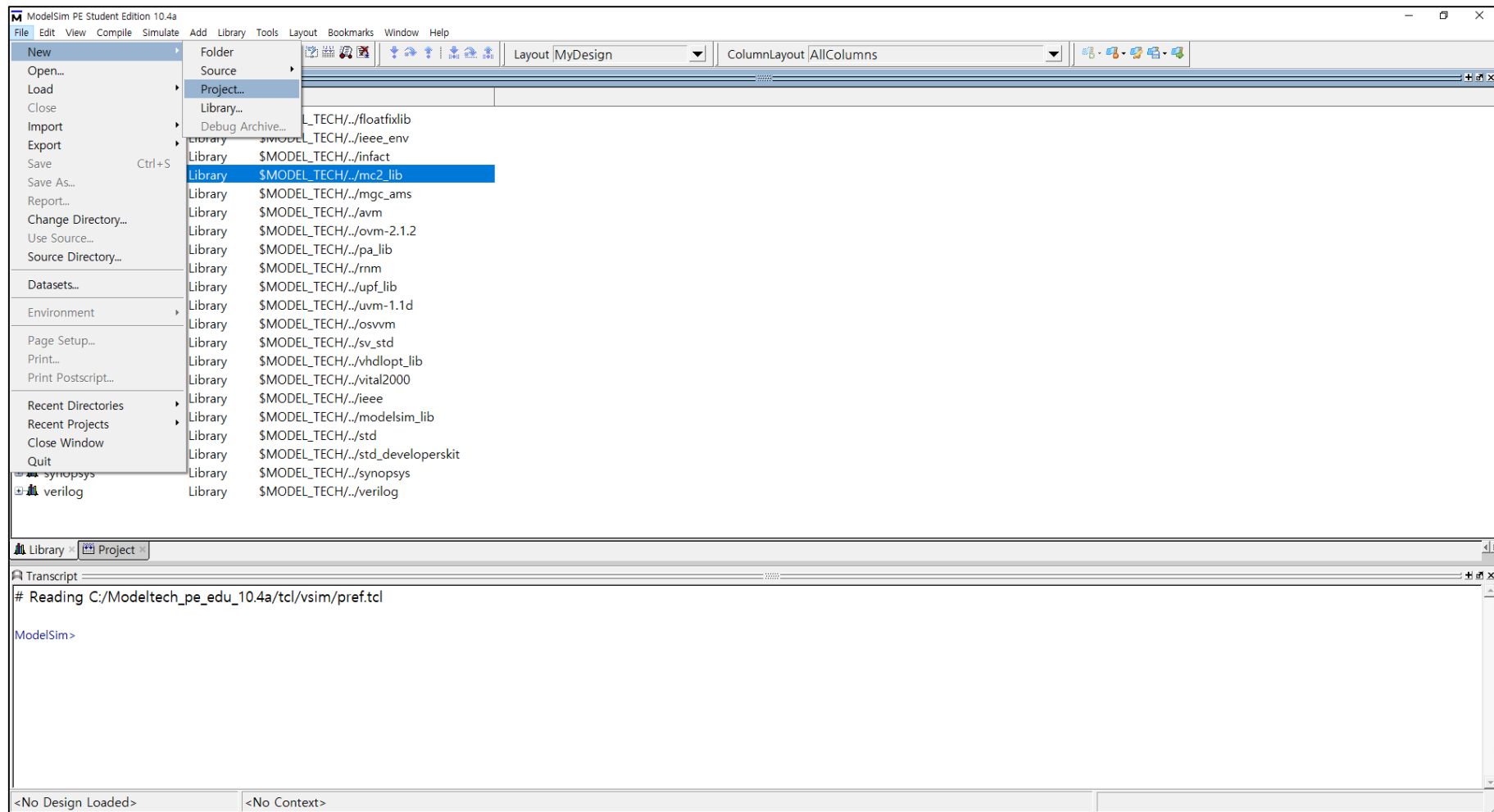
1. ModelSim 실행

■ ModelSim 첫 실행 화면



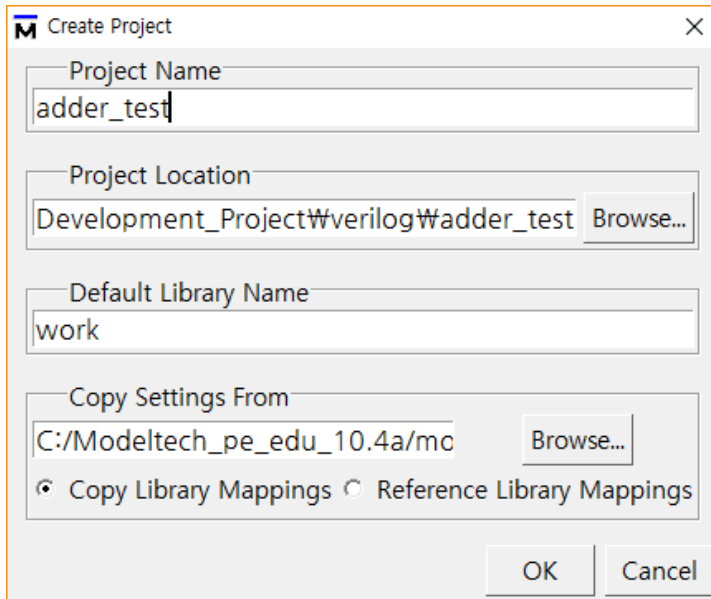
2. New Project 만들기

■ *File > New > Project* 메뉴 선택



2. New Project 만들기 (cont.)

■ Project Name 입력 및 Project Location, Default Library Name 설정



프로젝트 명 자유롭게 작성
(단순 .mpf 프로젝트 파일만 생성)

프로젝트 디렉토리 자유롭게 작성
(.mpf 파일이 생성되는 경로)

라이브러리 이름은 변경하지 않아도 됨
(이후 Simulate에서 사용할 이름)

■ Project 생성 후 디렉토리 모습



이름	수정한 날짜	유형	크기
work	2018-03-13 오후...	파일 폴더	
adder_test.mpf	2018-03-13 오후...	MPF 파일	91KB

3. Verilog 소스 작성 및 추가

□ Adder Simulation Example

- 한글 주석은 ModelSim 컴파일하면서 깨지므로 참고 목적으로만 활용

fulladd.v

```
// 1-비트 전가산기의 정의
module fulladd(sum, c_out, a, b, c_in);

// I/O 포트 선언
output sum, c_out;
input a, b, c_in;

// 내부 넷
wire s1, c1, c2;

// 프리미티브 논리 게이트의 파생
xor (s1, a, b);
and (c1, a, b);

xor (sum, s1, c_in);
and (c2, s1, c_in);

or (c_out, c2, c1);

endmodule
```

fulladd4.v

```
// 4-비트 전가산기의 정의
module fulladd4(sum, c_out, a, b, c_in);

// I/O 포트 선언
output [3:0] sum;
output c_out;
input [3:0] a,b;
input c_in;

// 내부 넷
wire c1, c2, c3;

// 4개의 1-비트 전가산기의 파생
fulladd fa0(sum[0], c1, a[0], b[0], c_in);
fulladd fa1(sum[1], c2, a[1], b[1], c1);
fulladd fa2(sum[2], c3, a[2], b[2], c2);
fulladd fa3(sum[3], c_out, a[3], b[3], c3);

endmodule
```


3. Verilog 소스 작성 및 추가 (cont.)

□ Adder Simulation Example

stimulus.v

```
// 스티뮬러스 모듈
module stimulus;

// 변수를 선언
reg [3:0] A, B;
reg C_IN;
wire [3:0] SUM;
wire C_OUT;
reg [31:0] STD_ID;

// FA1_4라는 4-비트 전가산기를 파생
fulladd4 FA1_4(SUM, C_OUT, A, B, C_IN);

// 신호값을 출력
initial
begin
    $monitor($time, " A = %b, B = %b, C_IN = %b, --- C_OUT = %b, SUM = %b\n", A, B, C_IN, C_OUT, SUM);
end

// 시뮬레이션 입력
initial
begin
    STD_ID = 32'd2018147486;
    A = 4'd0; B = 4'd0; C_IN = 1'b0;
    #5 A = 4'd3; B = 4'd4;
    #5 A = 4'd2; B = 4'd5;
    #5 A = 4'd9; B = 4'd9;
    #5 A = 4'd10; B = 4'd15;
    #5 A = 4'd10; B = 4'd5; C_IN = 1'b1;

end

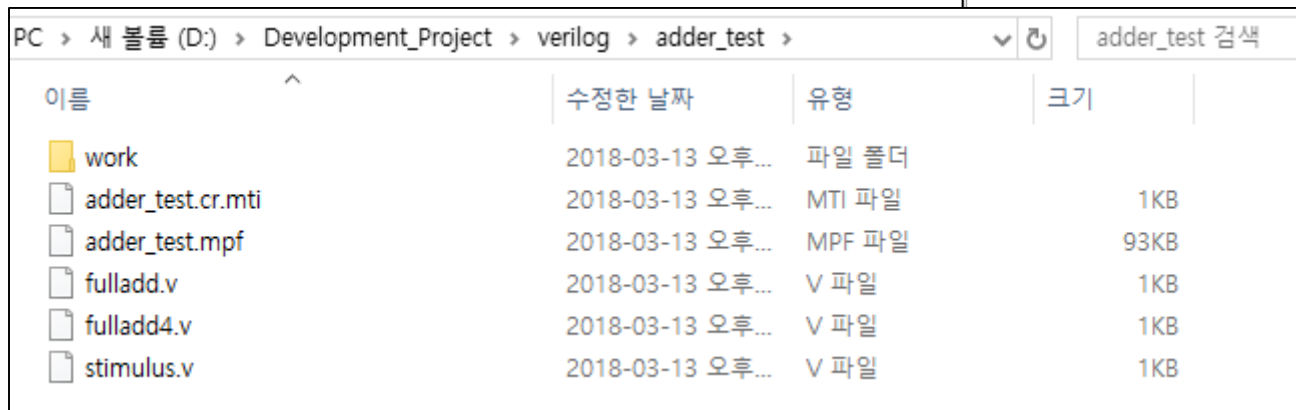
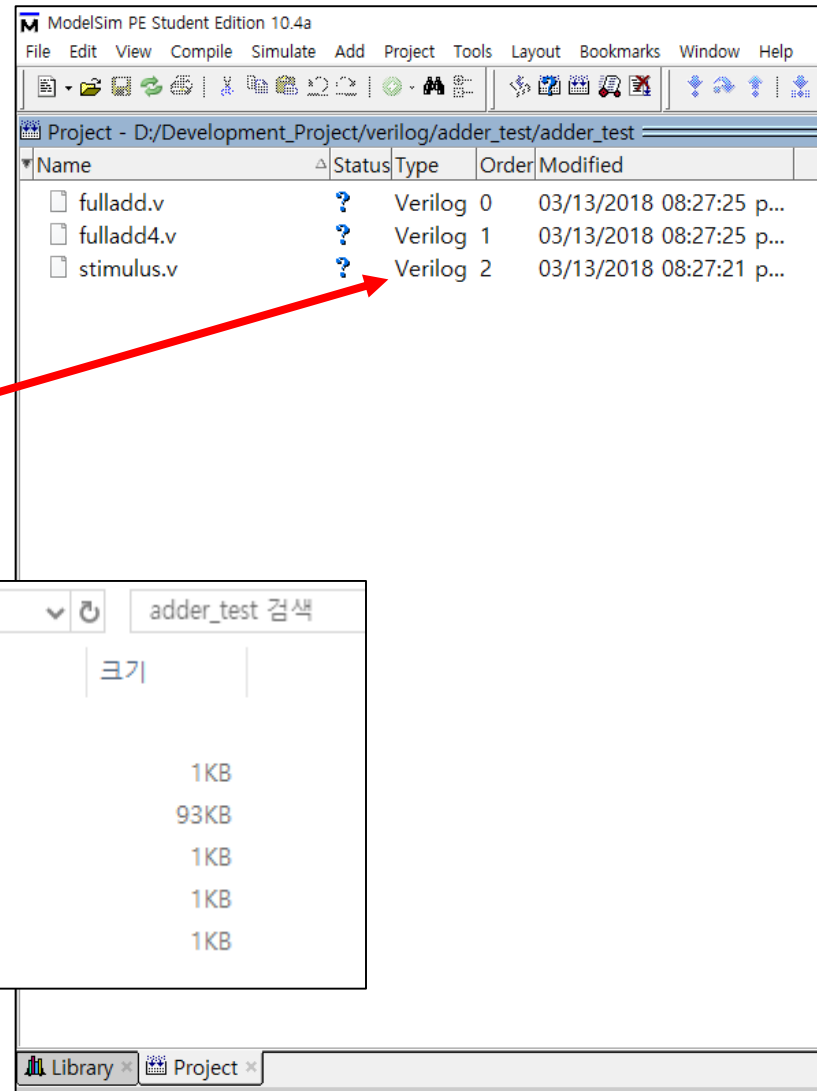
endmodule
```

2018147486 대신 자기 학번 입력

3. Verilog 소스 작성 및 추가 (cont.)

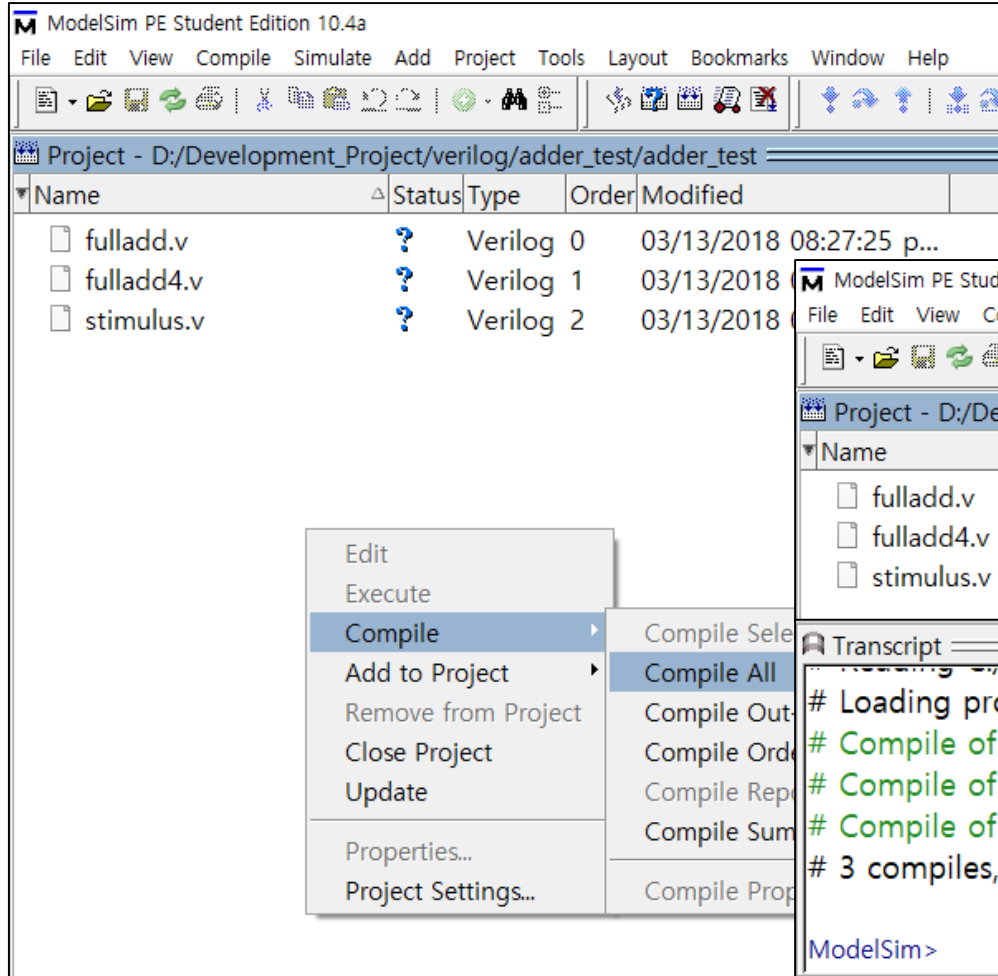
- Verilog 코드 작성 후 Project에 file 추가
 - ◆ Module files (*fulladd.v*, *fulladd4.v*)
 - ◆ Stimulus file (*stimulus.v*)

Type이 Verilog(VHDL 아님!)로
맞게 설정되었는지 확인

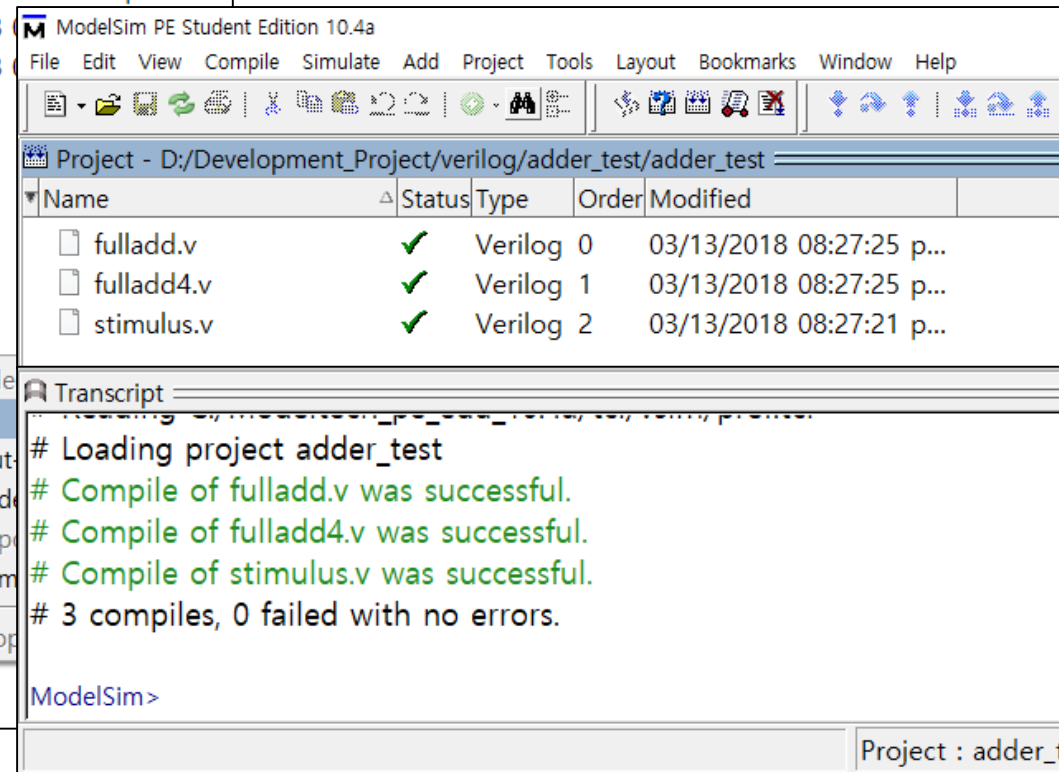


4. Compile

- 마우스 우클릭 > Compile > Compile All 선택



- Compile 완료



5. Simulate

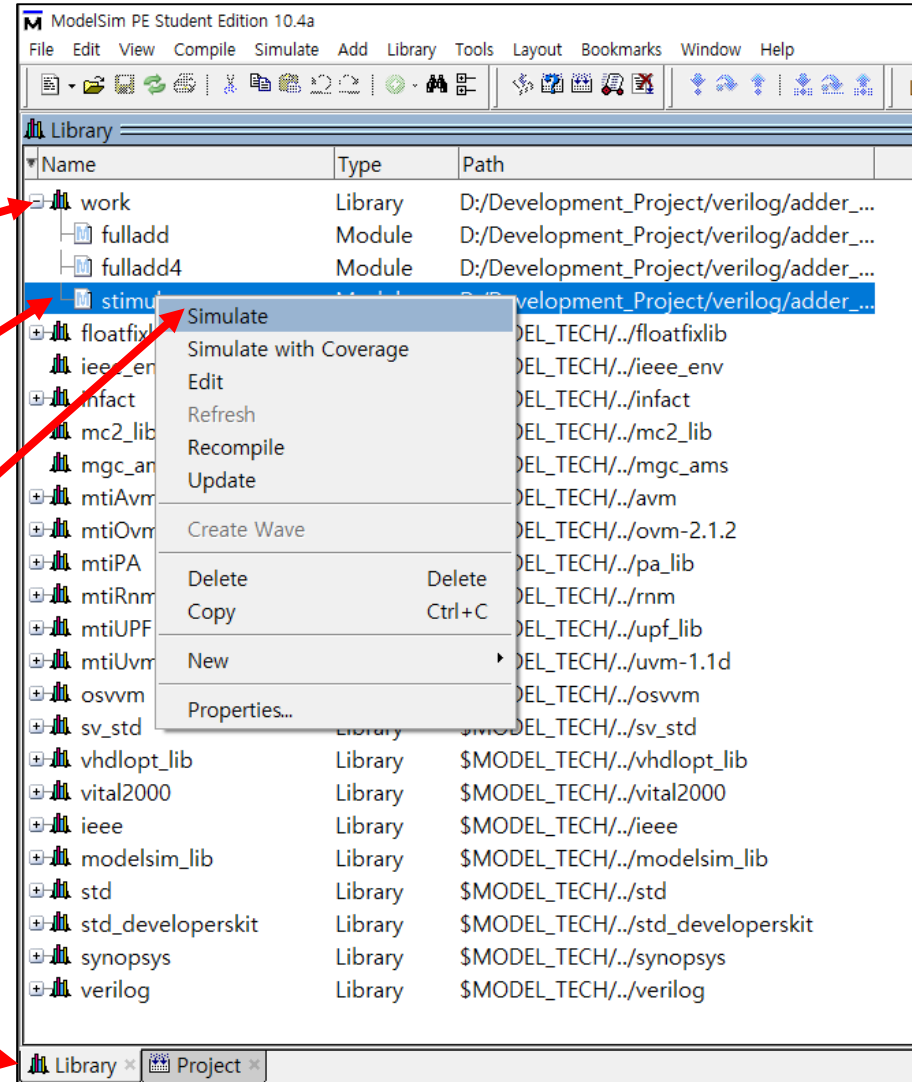
- *Library > work (default)*
 - ◆ *stimulus module 에서 Simulate 선택*

2. +work 클릭
(이전에 정의한 default library 이름)

3. stimulus 우클릭

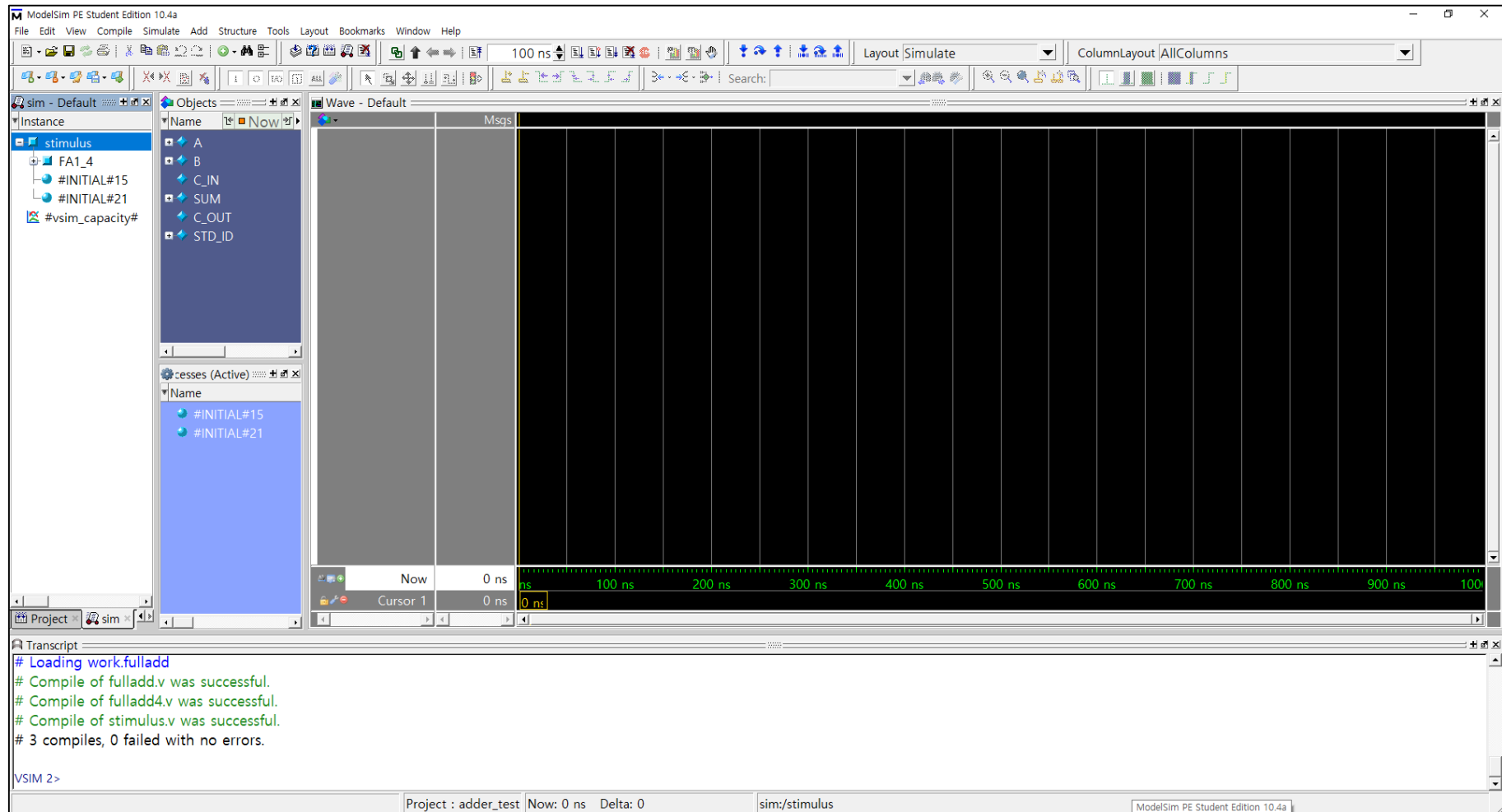
4. Simulate 클릭

1. Library 클릭



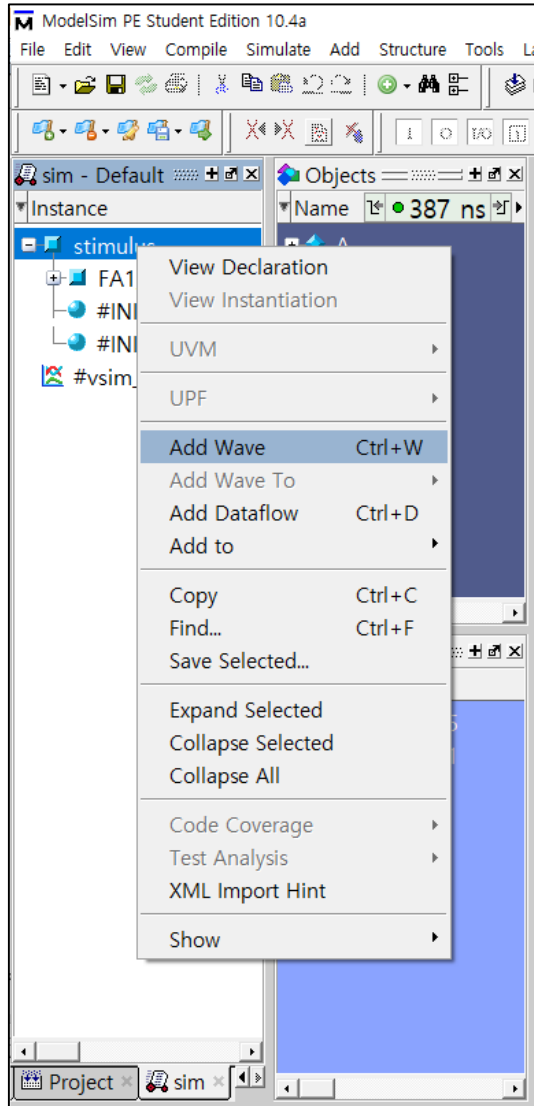
5. Simulate (cont.)

■ Simulation 초기 화면

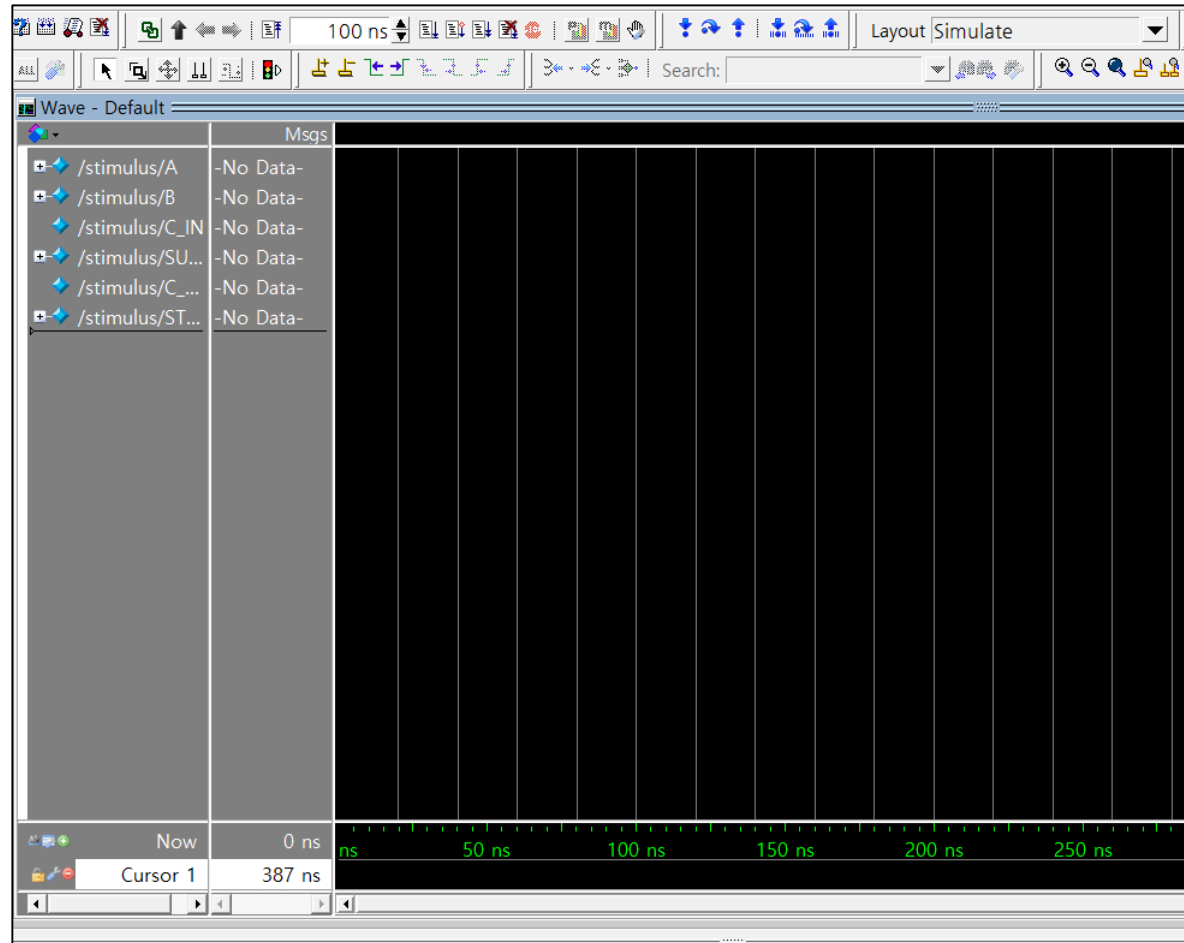


6. Simulate wave 추가

- *stimulus* 우클릭 > Add Wave 클릭

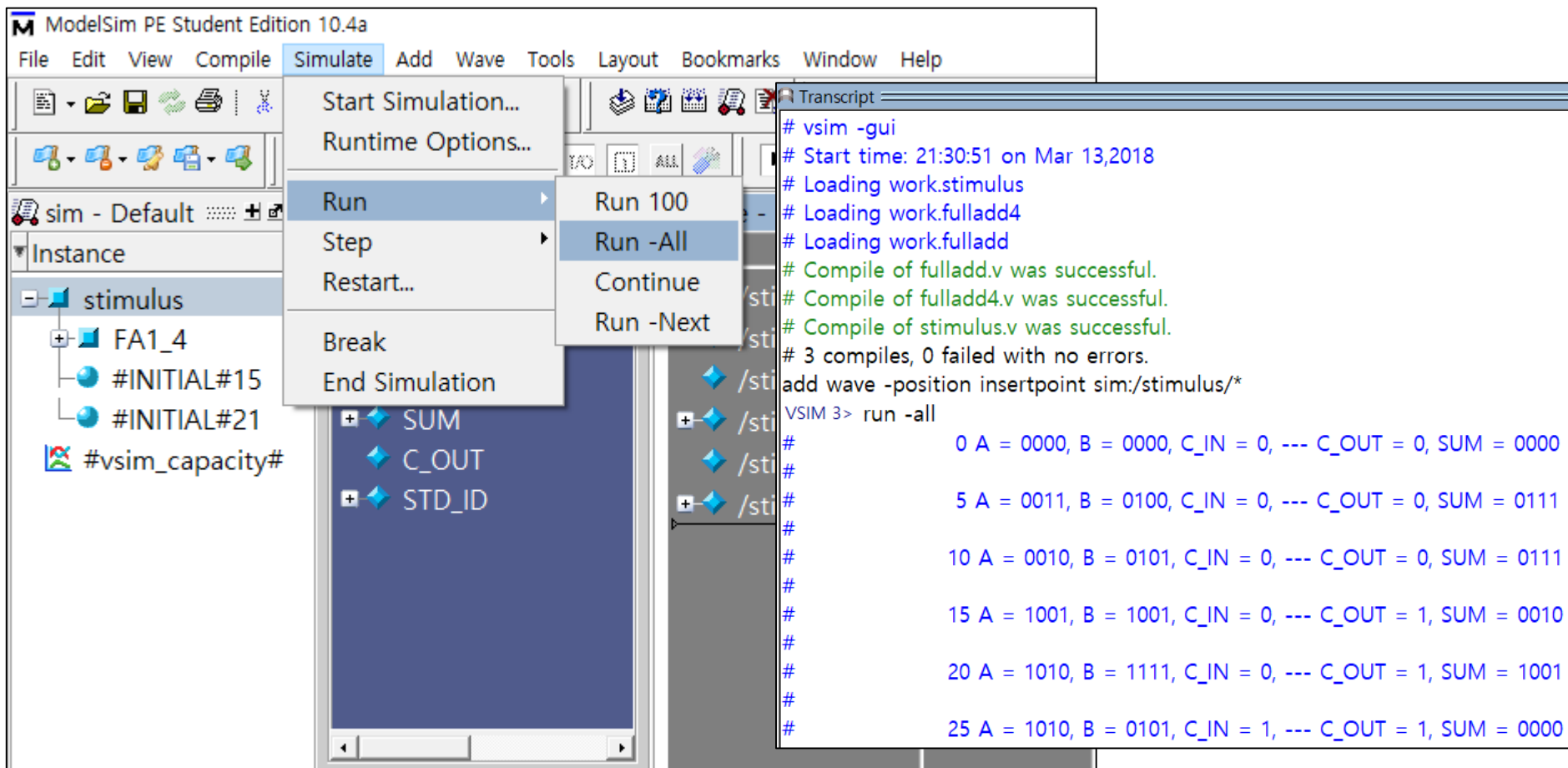


- *Wave form* 추가



7. 실행

- *Simulate > Run > Run -All* 클릭 또는 *Transcript* 창에서 *run -all* 입력 후 실행
 - ◆ *stimulus module* 내 *\$monitor* 실행 결과 출력
 - ◆ *Error* 발생 시 *error message* 출력



8. Simulation 실행 결과

■ Simulation 종료 후 화면

ModelSim PE Student Edition 10.4a

File Edit View Compile Simulate Add Wave Tools Layout Bookmarks Window Help

100 ns

Layout Simulate

ColumnLayout AllColumns

sim - Default

Objects

Instance

stimulus

FA1_4

#sim_capacity#

STD_ID

Processes (Active)

Name

Wave - Default

Msgs

/stimulus/A 4'h'a

/stimulus/B 4'h'5

/stimulus/C_IN 1'h'1

/stimulus/SU... 4'h'0

/stimulus/C_... 1'h'1

/stimulus/ST... 32'h784a7...

Object Declaration

Add

Edit

View

UPF

Radix

Global Signal Radix...

Symbolic

Binary

Octal

Decimal

Unsigned

Hexadecimal

ASCII

Time

Sfixed

Ufixed

Use Global Setting

Show Base

Numeric Enums

Symbolic Enums

1 float32

2 float64

Wave 아무 곳 클릭 후
Ctrl + (+키) 다섯 번 확대

STD_ID wave 우클릭
> Radix > Decimal 선택

VSIM 8>

0 ns to 1 us

Project : adder_test

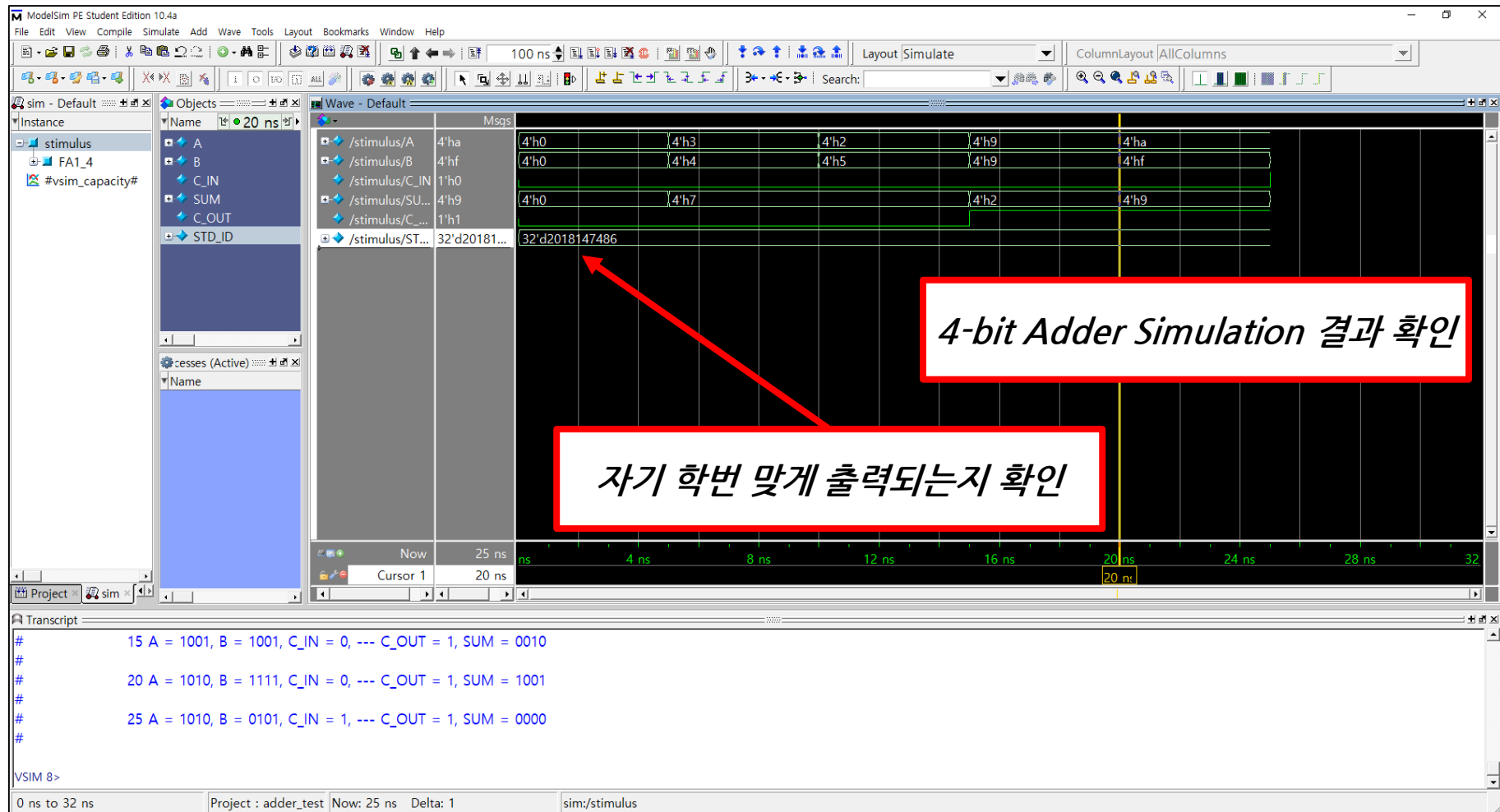
Now: 25 ns

Delta: 1

sim:/stimulus

8. Simulation 실행 결과 (cont.)

- Clock 경과에 따른 계산 결과 Wave form 출력



9. Simulation 보고서 제출

- ❑ 실행 결과 스크린 샷을 포함하여 ModelSim 시뮬레이션 보고서 제출
- ❑ OS(VM) 표기 / 자유 형식 / 1 페이지 이내 / 기한은 YSCEC 공지

