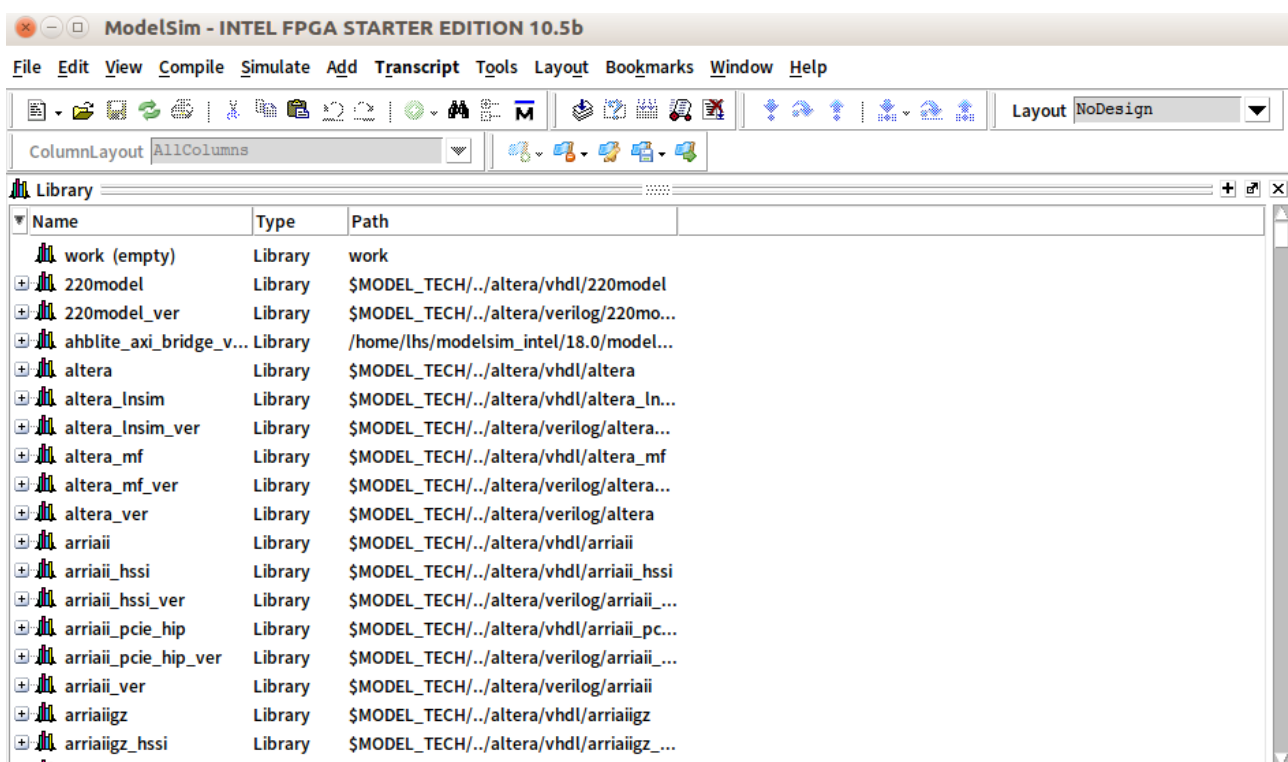


pwm_step

multisim사용

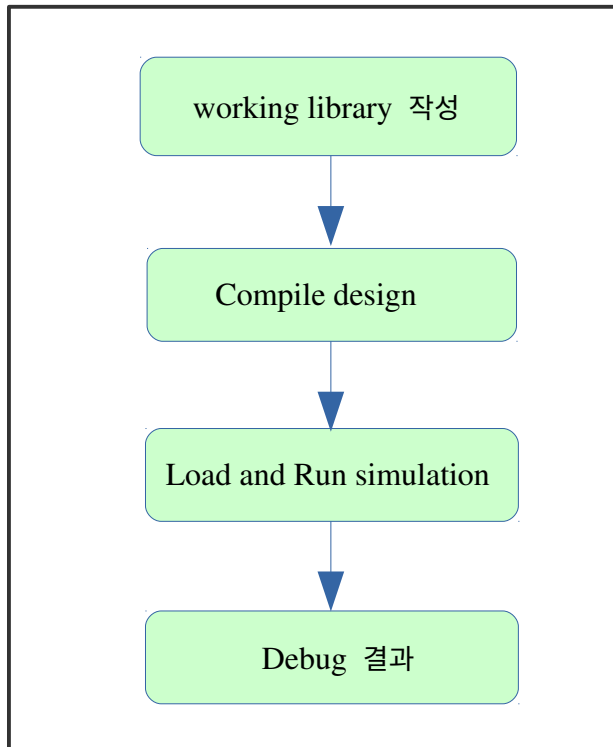
1. Simulation 사용법 - 1

모델심에서 직접 verilog 코드를 짜서 라이브러리에 추가한 후 시뮬레이션 하는 방법이다. 프로젝트를 이용해 수행하는 방법도 있는데, 뒤에서 설명하도록 한다. 지금은 코드만 라이브러리에 추가해서 수행해 본다. \$ vsim & 으로 실행하면 다음과 같은 실행 창이 뜬다.

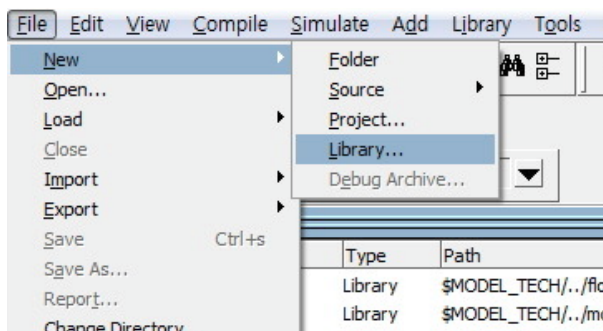


우선 시뮬레이션을 생성할 폴더를 선택해야한다. File >> Change Directory를 선택해 사용할 폴더를 선택한다. 필자는 /home/lhs/test_m 폴더를 선택했다. 폴더를 선택하면 가장 아래쪽의 Transcript 창에 폴더가 이동 되었다는 메시지가 뜬다. (이는 이동 명령어이다. 터미널 창처럼 사용 할 수 있다.)

이제 라이브러리를 살펴 보자. 모델심은 다음과 같은 과정을 거쳐 시뮬레이션을 만들 수 있다고 한다.



1) Working library를 만들어 본다.



work 라는 이름으로 라이브러리를 만든다. 그러면 자동으로 아까 설정했던 디렉터리 폴더로 저장된다. 라이브러리 창을 살펴보면 work 라이브러리가 추가된 것을 확인할 수 있다. 그리고 home/lhs/test/work 폴더가 생성된 것을 알 수 있고, 그 안에는 _info 파일이 존재한다. 이 파일은 라이브러리의 정보를 기록한 파일이다. 그리고 modelsim.ini 파일이 새로 생긴 것을 확인할 수 있다. Transcript 창에서 잘 만들어졌는지 확인한다.

```
Transcript :
# Cannot change working directory to /home/lhs/test_m : no such file or directory
cd /home/lhs/test_m
# reading /home/lhs/modelsim_intel/18.0/modelsim_ase/linuxaloem/./modelsim.ini
vlib work
vmap work work
# Model Technology ModelSim - Intel FPGA Edition vmap 10.5b Lib Mapping Utility 2016.10 Oct 5 2016
# vmap work work
# Copying /home/lhs/modelsim_intel/18.0/modelsim_ase/linuxaloem/./modelsim.ini to modelsim.ini
# Modifying modelsim.ini
ModelSim> ls
# modelsim.ini
# work
```

- ini 파일을 복사해 library mapping을 수행했다.
- 설계할 회로(Entity)를 저장할 라이브러리를 만들었다.

2) 코드 작성

인터넷 예제를 따라해보려 한다. vhd로 되어 있어 vhd를 선택한다. 선택하면 Edit 창이 생기는데 회로를 복붙하면된다.

```
library IEEE;

use IEEE.std_logic_1164.all;

entity or_gate is

    port( X, Y : in std_logic;

        A : out std_logic);

end;

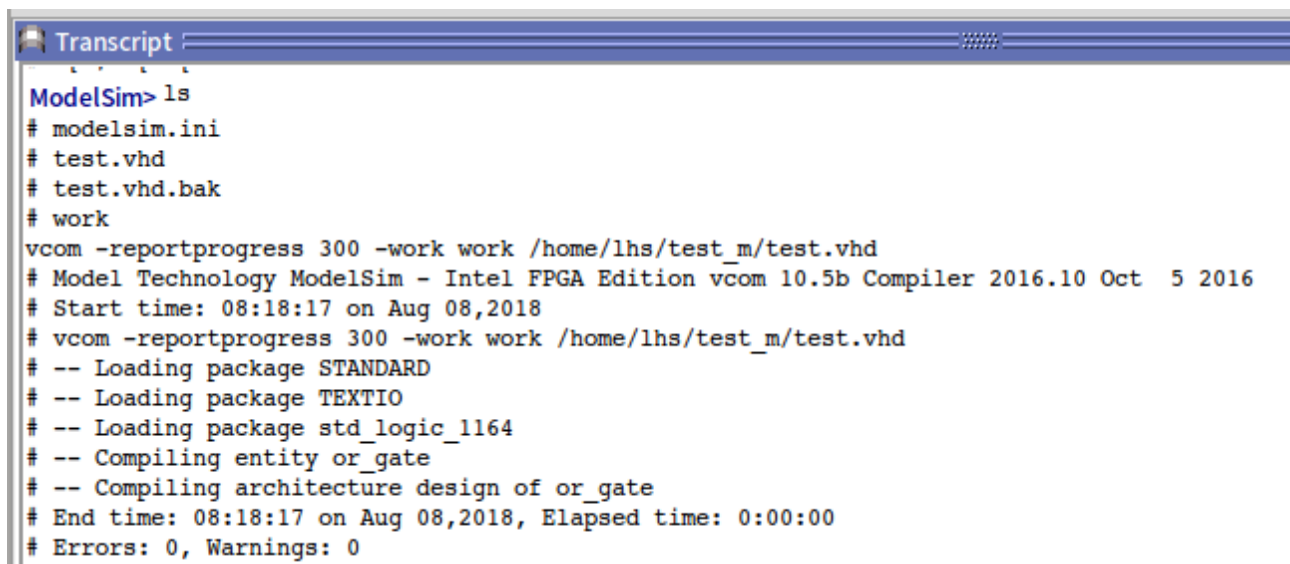
architecture design of or_gate is

begin

    A <= X or Y;

end design;
```

File >> save로 파일 이름을 정하고 저장한다. test.vhd 란 이름으로 저장하겠다. 이제 compile을 클릭한다. 그럼 Transcript 창에 다음과 같은 메시지가 뜬다.

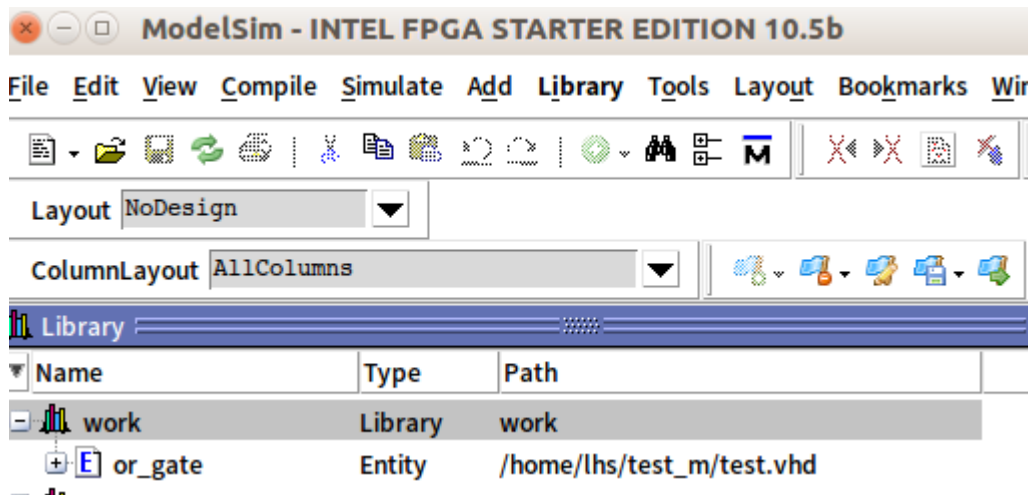


```
Transcript

ModelSim> ls
# modelsim.ini
# test.vhd
# test.vhd.bak
# work
vcom -reportprogress 300 -work work /home/lhs/test_m/test.vhd
# Model Technology ModelSim - Intel FPGA Edition vcom 10.5b Compiler 2016.10 Oct  5 2016
# Start time: 08:18:17 on Aug 08,2018
# vcom -reportprogress 300 -work work /home/lhs/test_m/test.vhd
# -- Loading package STANDARD
# -- Loading package TEXTIO
# -- Loading package std_logic_1164
# -- Compiling entity or_gate
# -- Compiling architecture design of or_gate
# End time: 08:18:17 on Aug 08,2018, Elapsed time: 0:00:00
# Errors: 0, Warnings: 0
```

컴파일이 완료 되었음을 확인할 수 있다. 이렇게 설계한 회로는 모두 위에서 만들었던 work 라이브러리 에 보관된다.

library창을 보면 or_gate회로가 추가 된 것을 볼 수 있다.



3) 이제 회로를 완성했고, 이를 작동시킬 testbench라는 파일을 작성해야 한다. testbench를 작성하는 이유는 만든 회로에 입력을 정해주는 것이다. 입력을 설정해주는 부가 회로를 만들어준다고 생각하면 된다. vhdl파일을 하나 더 만들고 test_tb.vhd로 정해준다. 그 뒤 저장을 하고 컴파일을 한다. 코드는 다음과 같다.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity tb_or_gate is
end tb_or_gate;

architecture design of tb_or_gate is
    component or_gate is
        port( X, Y : in std_logic;
              A : out std_logic);
    end component;

    signal X, Y : std_logic := '0';
    signal A : std_logic := '0';

begin
    U0 : or_gate port map(X=>X, Y=>Y, A=>A);

    process
    begin
        X <= '0';
        Y <= '0';
        wait for 10ns;
```

```

X <= '0';
Y <= '1';

wait for 10ns;

X <= '1';
Y <= '0';

wait for 10ns;

X <= '1';
Y <= '1';

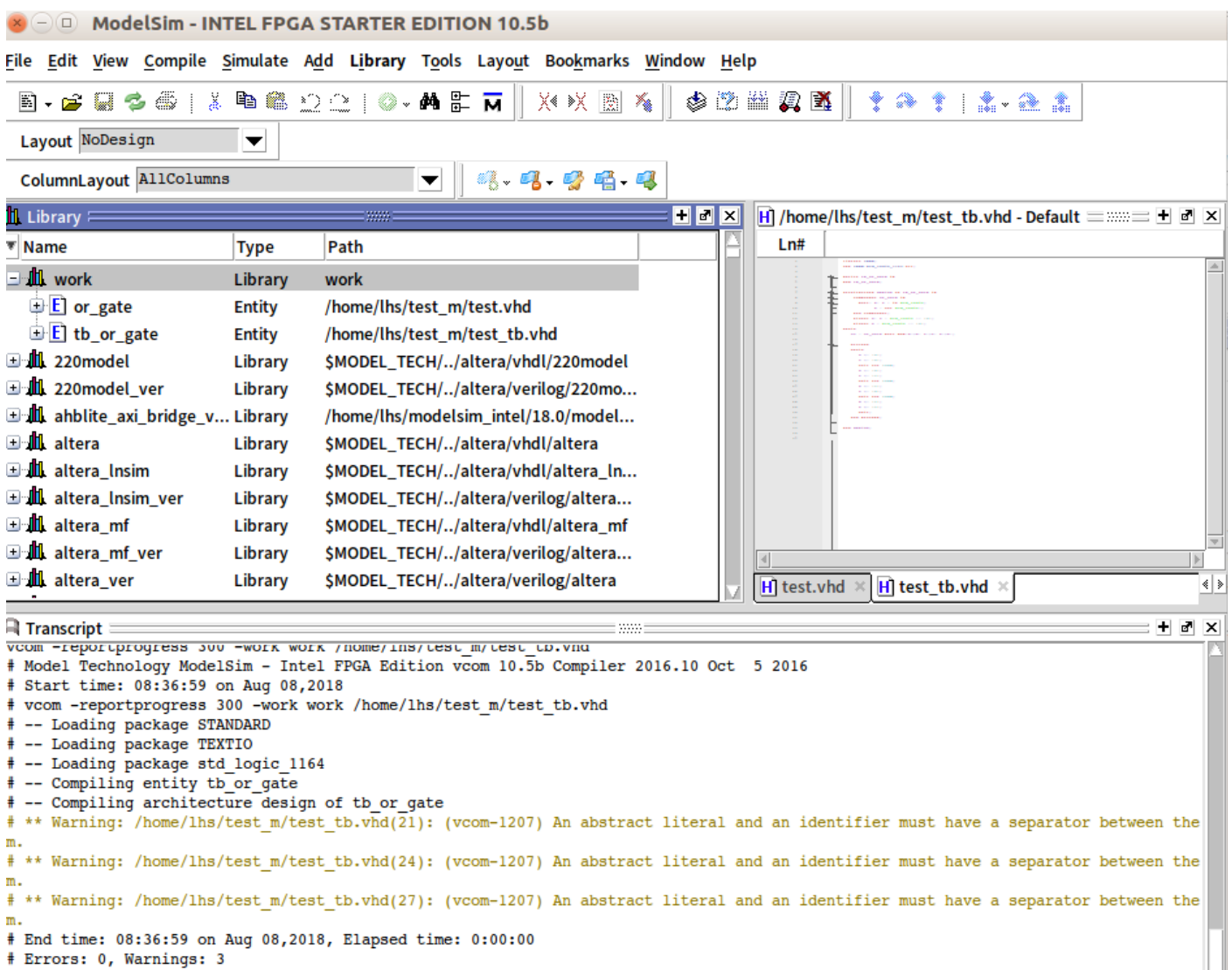
wait;

end process;

end design;

```

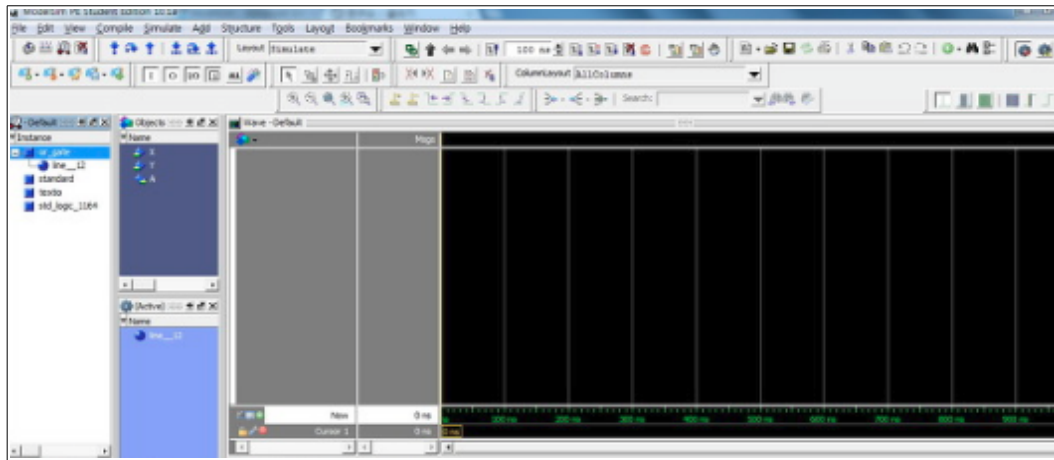
컴파일까지 진행하면 다음과 같은 화면을 볼 수 있다.



4) 라이브러리 창을 살펴보면 tb_or_gete 가 work 라이브러리에 추가되어 있다. 이제 모든 회로 구성이 되었다. 시뮬레이션은 testbench 회로를 이용하여 start simulation 을 진행한다.

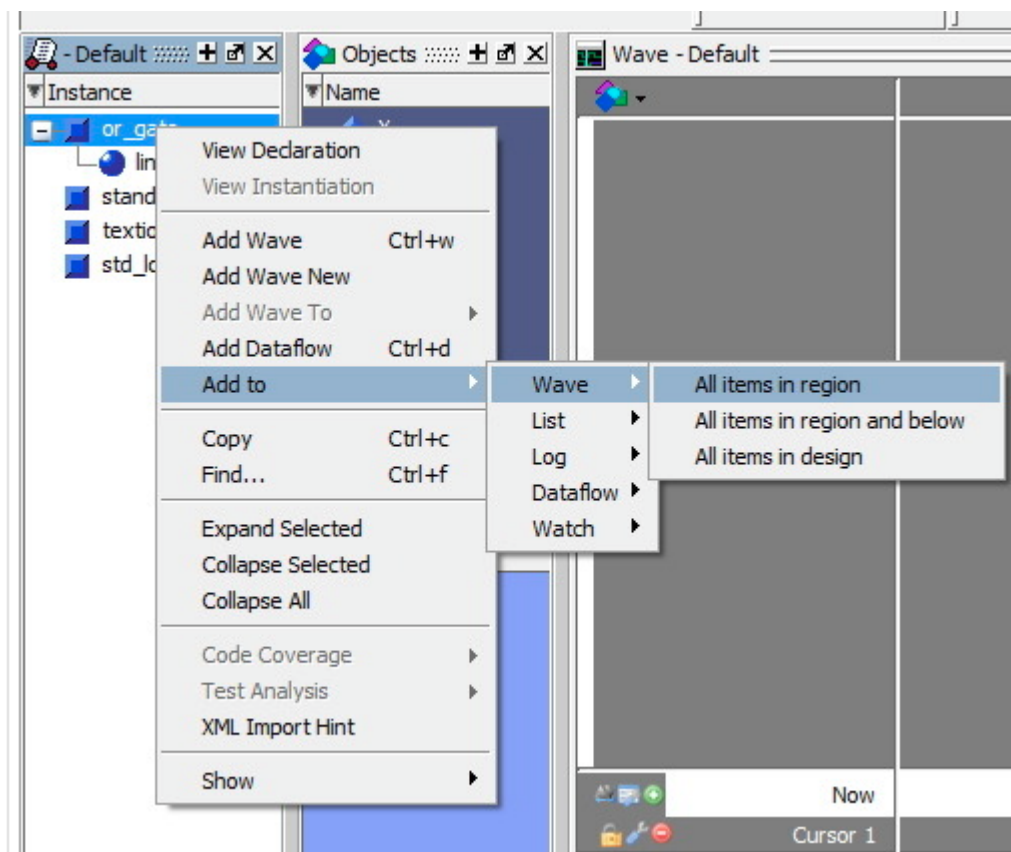
Simulation >> Start simulation.

이제 wave 창을 보면된다. 없다면 View 메뉴에서 추가하도록 하자.



이제 wave창에 신호를 추가해야한다.

or_gate 의 입출력신호를 모두 추가한다.



추가가 됐다면 Transcript 창에 run 명령을 이용해서 시뮬레이션을 수행하면 된다.

\$ run 시간 을 입력하면된다.

\$ run 20ns 를 입력하면 다음과 같은 파형을 확인할 수 있다.

ModelSim - INTEL FPGA STARTER EDITION 10.5b

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

100 fs

Layout Simulate

ColumnLayout AllColumns

Instance

- tb_or_gate
 - U0
 - line_18
 - standard
 - textio
 - std_logic_1164

ts

- X
- Y
- A

Wave - Default

	Msgs
/tb_or_gate/X	0
/tb_or_gate/Y	0
/tb_or_gate/A	0

Now 00400 fs

Cursor 1 11 fs

20000000 fs

Transcript

```
run
VSIM 19> run
```