
VHDL 및 실습

Quartus II, VHDL

과목	VHDL 및 실습
학과	전자공학과
학번	2011144024
이름	유대성 (오전)
제출일	
담당교수	최종성 교수님

1. 주제 및 배경이론



VHDL이라는 언어에 대해 이해하고 개발 방법에 대해 살펴보자.

VHDL이란?

VHSIC Hardware Description Language의 약자로 VHSIC은 다시 Very High Speed Integrated Circuits의 약자이다. 디지털 회로를 표현하는 **하드웨어 기술 언어**이다.

탄생배경

VHDL은 미국 국방부에서 주문형 집적회로(ASIC = Application Specific Integrated Circuit)를 문서화하기 위해 제안되었다. 처음엔 동작을 기술하는 데에서 시작하였지만 반대로 기술된 문서로 회로를 디자인할 순 없을까? 생각하게 되었고 이를 통해 동작할 수 있는 회로를 구성할 수 있게 되었다. 오늘날에는 검증, 시연 등의 다양한 기능을 내포하게 되었다.

특징

특정 기술이나 공정에 대해 독립적

- 공통된 기술 언어로 특정 기술이나 공정에 관계없이 설계가 가능

표준화 및 문서화

- **IEEE에 의해 표준화**되어 있으며 기술적 특징을 문서화할 수 있음

VHDL로 완전한 회로 시뮬레이션은 가능하지만 모든 구성이 합성되진 않음

일반적인 컴퓨터 프로그램에 비해 **병렬적으로 동작함**.

- 순차적 실행이 아닌 동시에 동작하는 방식이다. 전류가 모든 회로에 동시에 흐르듯 VHDL로 구술한 프로그램 또한 같은 특성을 지닌다.

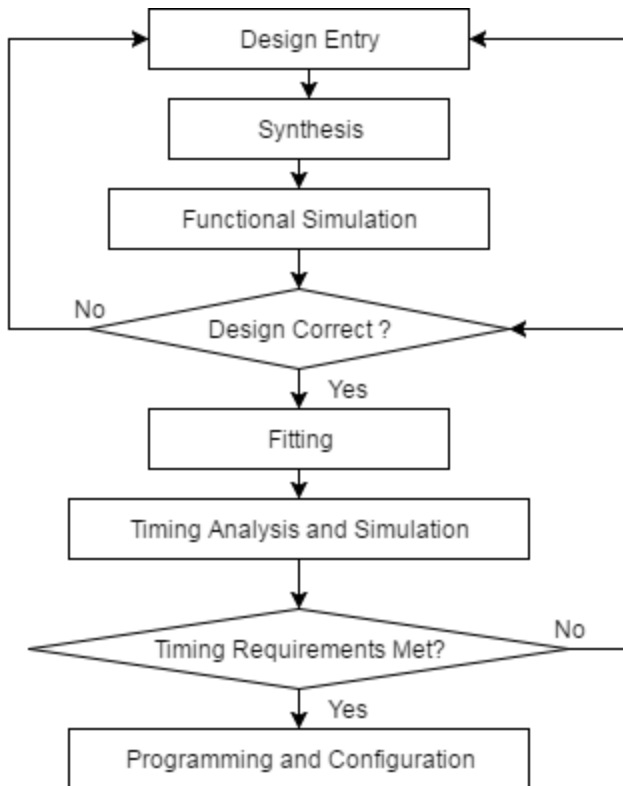
많은 시간과 노력이 필요

- VHDL 자체의 하드웨어적인 특징으로 Flow대로 구현하다 보면 시간이 많이 소요되고 복잡하다.

합성 룰에 따라 회로 성능이 좌우됨

- 합성 룰에 따라 VHDL 구문의 지원여부가 달라져 성능상의 문제가 발생

Design Flow



Design Entry : Schematic 또는 VHDL 로 회로를 구성하는 단계

Synthesis : 파형으로 입력레벨을 설정하는 단계

Functional Simulation : 입력레벨에 따른 출력레벨을 시뮬레이션 하는 단계

Design Correct? : 출력레벨을 확인

Fitting : Quartus 2 프로그램 내 Full Compilation, Programmer

Timing Analysis and Simulation : 회로의 물리적 특징에 의한 time delay 를 시뮬레이션

Timing Requirements Met? : 적정 수준의 time delay 인지 확인

Programming and Configuration : 물리적인 동작을 확인한다.

VHDL 의 구조

```
1 library ieee;
2 -- VHDL 은 IEEE 국제 표준을 사용한다.
3   use ieee.std_logic_1164.all;
4   use ieee.std_logic_arith.all;
5   use ieee.std_logic_unsigned.all;
6
7 entity and_gate is
8 -- Entity 선언부 는 설계하고자 하는 시스템의 외적 연결을 담당 한다.
9 -- 회로의 내부적인 구조나 연결등을 고려할 필요가 없으며,
10 -- 여기서 정의한 것을 통해 다음 Architecture Body 에서 내부적 동작을 표현한다.
11 -- 즉, 입출력 선을 정의 하는 부분이다.
12   port(
13     A : in std_logic;
14     B : in std_logic;
15     Y : out std_logic
16     -- 선언부 마지막 문장에 세미콜론이 없음에 유의하자.
17   );
18 end and_gate;
19
20 architecture BEH of and_gate is
21 -- 시스템의 동작을 세부적으로 정의한다.
22 begin
23   Y <= A and B;
24 end BEH;
```

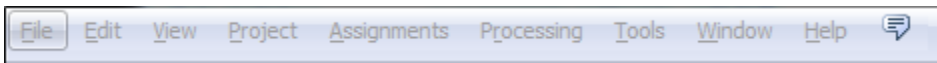
Colored by Color ScripterCS

2. Schematic (소스코드 및 코드 설명 & 시뮬레이션 결과 및 설명)

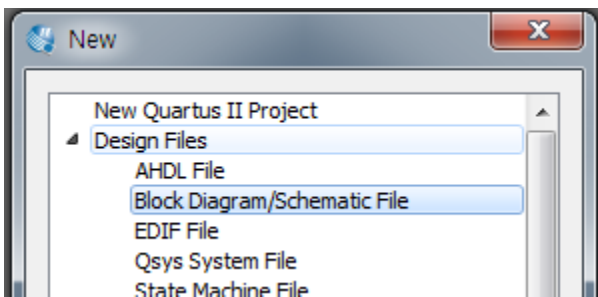
i Schematic 으로 설계한 후 테스트를 진행해보자.

프로젝트 생성

File > New Project Wizard



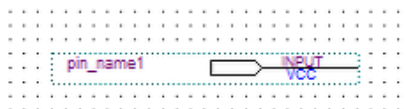
Schematic 파일 생성



- 1) 화면 위에서 빈공간 더블클릭 하여 소자 검색
 - Name 에 소자 이름으로 불러오기 가능 예제는 2 pin 의 AND 게이트 검색



- 2) 이제 입출력 핀을 생성해보자.
 - 도구모음의 아이콘을 이용하거나 소자 검색을 통해 불러올 수 있다.



- 3) 소자끼리의 연결은 아래 두 버튼이 활성화 된 상태에서 드래그를 통해서 연결이 가능하다.

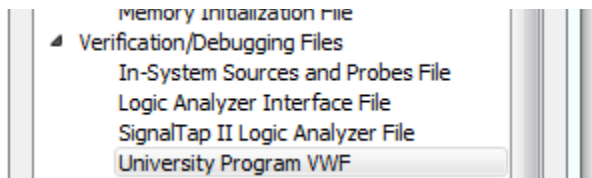


VWF 파일 생성

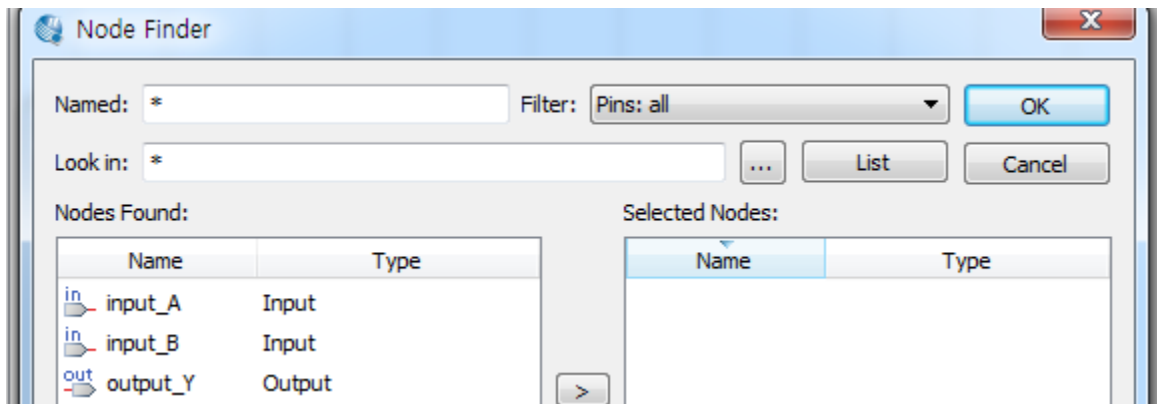
1) 아래 버튼을 눌러 VWF 파일에 사용될 입출력 정보를 저장하자.



2. University Program VWF 를 선택



좌측 박스의 빈공간을 더블클릭 또는 우클릭하여 입출력 정보를 추가하자.

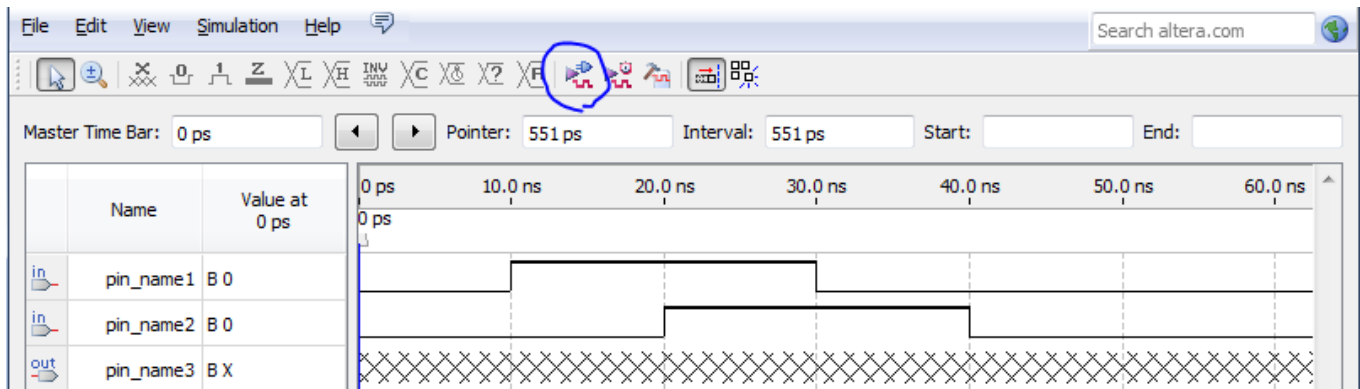


Function Simulation

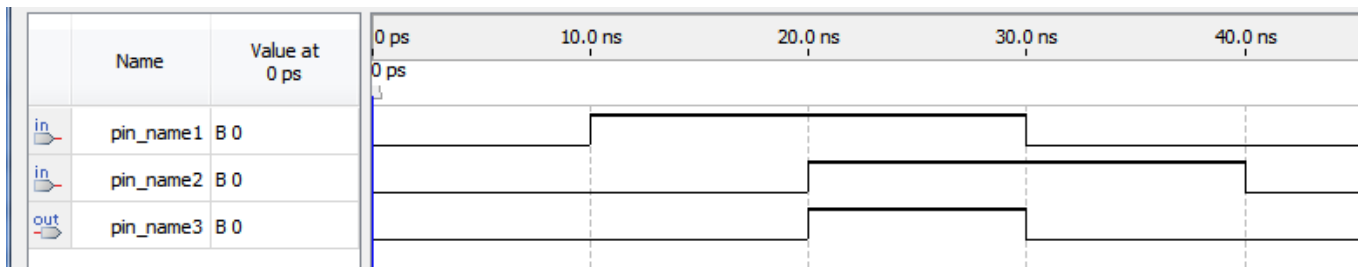
- 1) Simulation > function simulation



- 2) 입력 레벨을 설정한 후 아래 버튼을 누른다.



- 3) 아래와 같은 화면이 뜨면 성공

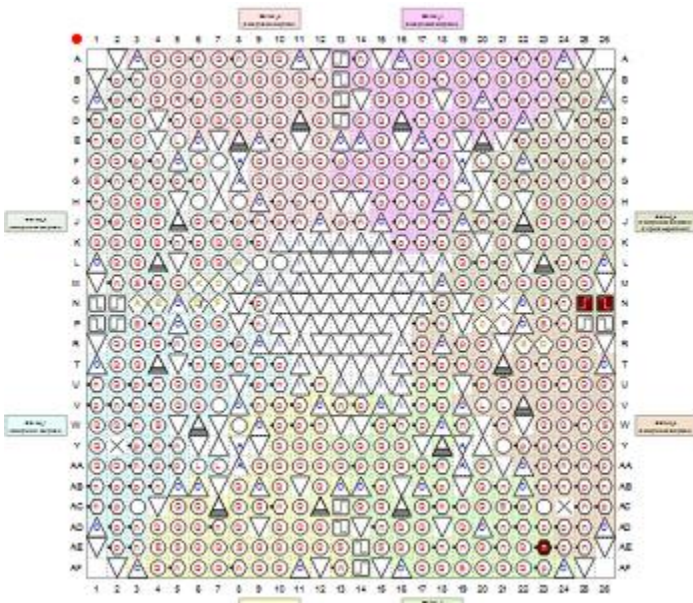


PIN PLANNER



- 1) Assignments > Pin Planner

Assignments

- 2) 아래 화면이 보인다면 성공



아래 Location 에 해당 핀번호를 입력하여 입출력 버스를 설정한다.

Named: *		Edit:  			
Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Local
pin_name1	Input	PIN_N25	5	B5_N1	PIN_C13
pin_name2	Input	PIN_N26	5	B5_N1	PIN_D13
pin_name3	Output	PIN_AE23	7	B7_N0	PIN_B11
<<new node>>					

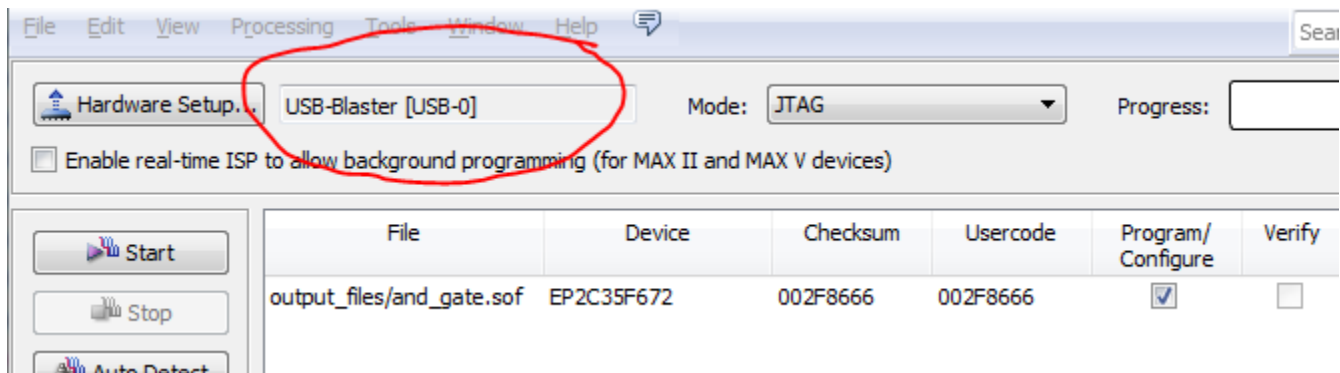
PROGRAMMER

- 1) tools > programmer



- 2) 해당 화면이 보인다면 성공

화면의 빨간 위치에 기기가 인식되지 않은 경우가 발생하면 맨 아래의 드라이버 설치 항목을 따라하면 된다.

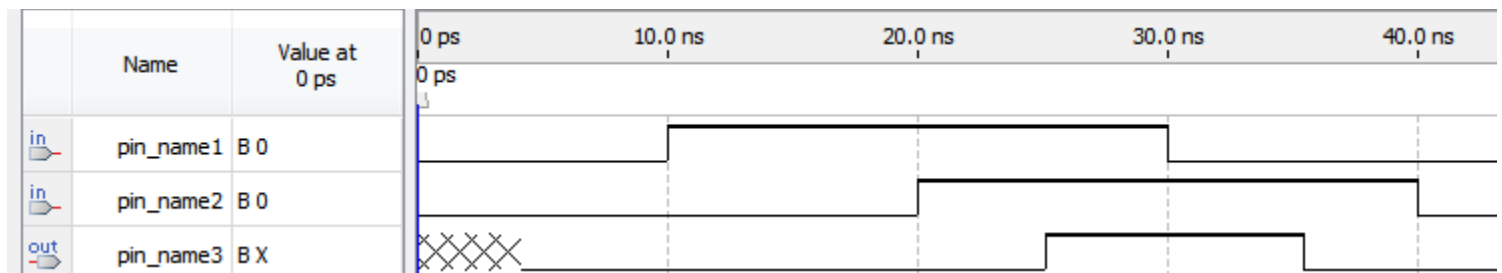


pof 는 영구 저장 시 사용하고 sof 는 플래시 데이터에 임시저장된다. sof 를 선택하여 설치하자.

여기까지 진행했다면 보드를 테스트 해볼 수 있다. 보드를 이용해 테스트 해보자.

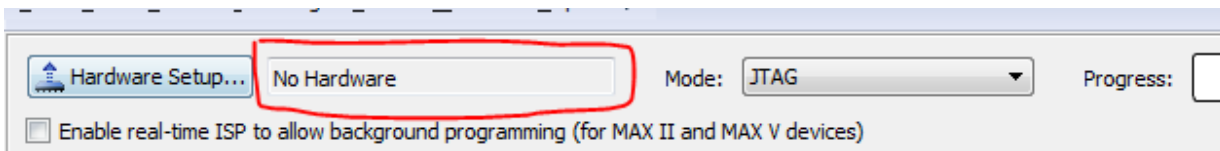
TIMING SIMULATION

아래와 같은 시간지연을 확인할 수 있다.



기기 인식 불가

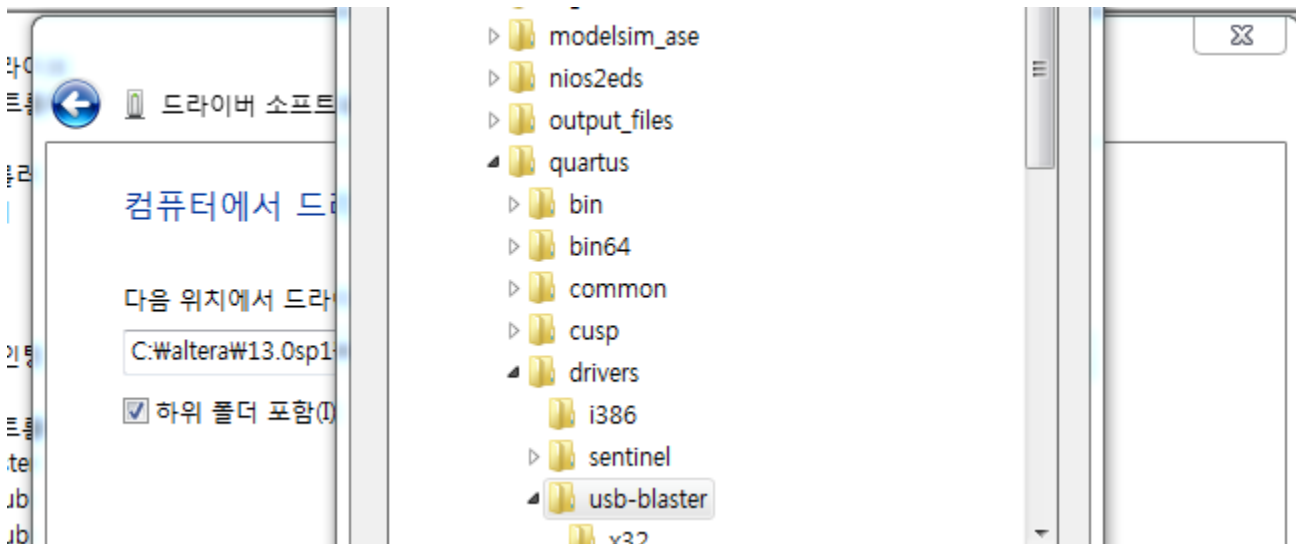
- 1) 아래와 같이 연결된 하드웨어를 인식하지 못하는 경우가 발생할 경우



- 2) 장치관리자로 들어가 범용 직렬 버스 컨트롤러를 확인한다.



여기에도 없다면 아래 경로를 찾아 해당 드라이버를 설치한다.



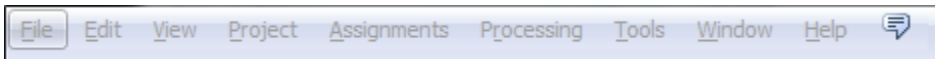
3. VHDL(소스코드 및 코드 설명 & 시뮬레이션 결과 및 설명)

i VHDL 로 설계한 후 테스트를 진행해보자.

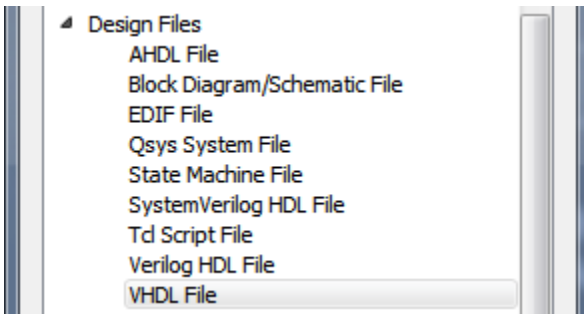
프로젝트 생성 단계는 Schematic 설계와 같으므로 생략한다.

VHDL 파일 생성

File > New



VHDL File



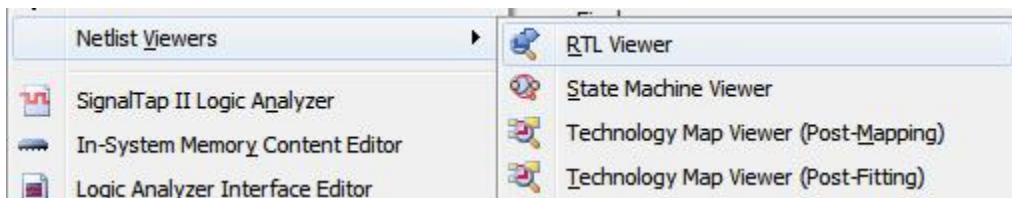
다음과 같은 화면에서 HDL 을 기술할 수 있다.

```
1  library ieee;
2      use ieee.std_logic_1164.all;
3      use ieee.std_logic_arith.all;
4      use ieee.std_logic_unsigned.all;
5
6  entity and_gate is
7  -- Entity 선언부 는 설계하고자 하는 시스템의 외적 연결을 담당 한다.
8  -- 회로의 내부적인 구조나 연결등을 고려할 필요가 없으며,
9  -- 여기서 정의한 것을 통해 다음 Architecture Body 에서 내부적 동작을 표현한다.
10 -- 즉, 입출력 선을 정의 하는 부분이다.
11      port(
12          A : in std_logic;
13          B : in std_logic;
14          Y : out std_logic
15          -- 선언부 마지막 문장에 세미콜론이 없음에 유의하자.
16      );
17  end and_gate;
18
19  architecture BEH of and_gate is
20  -- 시스템의 동작을 세부적으로 정의한다.
21  begin
22      Y <= A and B;
23  end BEH;
```

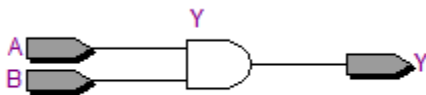
이후 시뮬레이팅 및 핀배치 과정은 Schematic 과정과 동일하므로 생략한다.

RTL Viewer

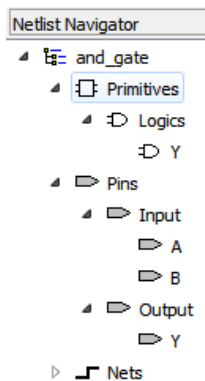
Toos > Netist Viewers > RTL Viewer 선택



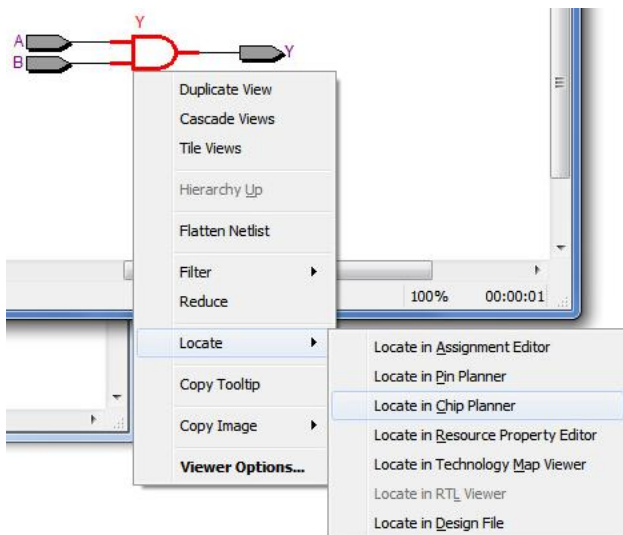
다음과 같은 화면이 나온다.



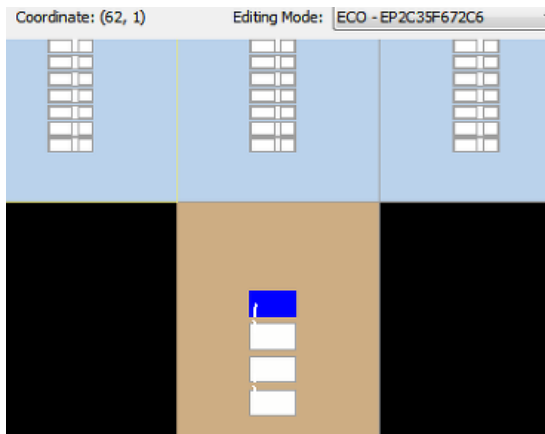
좌측 창 탭 중에서 Netist Navigator 탭을 선택하면 핀 정보 등을 볼 수 있다.



Logics 의 Y 선택 후 다음의 Locate in Chip Planner 를 선택



선택 후에 다음과 같은 화면을 볼 수 있다.



4. 실습보드 적용 결과

i 소스를 보드에 다운로드하여 예상한 결과에 맞게 동작하는 지 테스트하시오

미처 사진을 찍지 못했습니다. 이후 실습에서는 틈틈히 촬영하여 제출할 수 있도록 하겠습니다.

5. 실습소감

i 실습을 통해 경험한 것을 자유롭게 서술하시오.

유튜브 영상처럼 빠르게 진행할 순 없었으나 반복된 실습을 통해 각 툴의 위치를 찾는데 어려움이 없어졌고 각 기능이 어떤 역할을 하는 지 이해할 수 있었다. VHDL 과 Schematic 을 통해 디자인을 하고 각 툴을 사용해 Flow 대로 진행해보았는데 기존의 pspice 처럼 회로를 구성하는데 그치는 것이 아닌 눈에 보이는 결과를 얻어낼 수 있어서 재미있었다. 어렵지 않은 과제라 즐겁게 따라할 수 있었다.