**Lattice FPGA 개발 안내서**

목차

[관리 규칙 2](#_Toc108707961)

[파일 경로 규칙 2](#_Toc108707962)

[파일 경로 예시 3](#_Toc108707963)

[프로젝트 관리 규칙 4](#_Toc108707964)

[IDE 내부 소개 5](#_Toc108707965)

[프로젝트 파일 분류 5](#_Toc108707966)

[GPIO 핀 설정하기 6](#_Toc108707967)

[개발환경에서 GUI 설정 6](#_Toc108707968)

[그 외 방법 6](#_Toc108707969)

[프로그램 빌드 하기(디자인 검사) 7](#_Toc108707970)

[프로그램 넣기 7](#_Toc108707971)

# 관리 규칙

## 파일 경로 규칙

1. 한글 쓰지 마라
2. 파일의 용도별로 정확하게 분류하라
3. 시뮬레이션 파일을 따로 경로를 지정하라
4. 그 외 쌍욕 나오게 관리하지 마라 >> 당연한 것인데

## 파일 경로 예시

텍스트이(가) 표시된 사진

자동 생성된 설명

경로 설명(프로젝트 폴더 기준)

* DB : Data Base
* FPGA 전체 설계 관련 파일 경로
* Design
* 일반적인 Verilog 코드가 있는 경로(사용자가 지정함)
* IP
* Intellectual property(지적 재산권) : 레퍼런스로 제공되는 모듈 저장경로
* Top
* Verilog 코드가 있는 경로 / 프로젝트의 Main 함수 C 언어의 int main() 이다. 시작점
* Work
* 모델심 / 시뮬레이션 합성을 하기 위해 생성되는 파일경로. 일반적으로 Verilog 코드를 생성하면 자동으로 잡히는 경로이다. 경로를 분리해 디자인 관련 파일을 따로 관리하는게 프로젝트 진행에 있어 편하다.
* Impl1
* 프로젝트명으로 생성되는 경로 / 코드 생성시 자동으로 경로가 잡힘
* Doc
* FPGA 관련 자료를 넣어두는 문서 파일
* Sim
* 시뮬레이션을 돌릴 때 사용자가 생성하는 경로 / 시뮬레이션 파일을 여기에 저장

## 프로젝트 관리 규칙

1. GitHub로 제대로 관리하라
2. 업데이트 내용은 정확히 표현하라
3. 프로젝트 버전을 유의적 버전관리 기준에 따라라   
   >> [유의적 버전 2.0.0 | Semantic Versioning (semver.org)](https://semver.org/lang/ko/)

# IDE 내부 소개

## 프로젝트 파일 분류

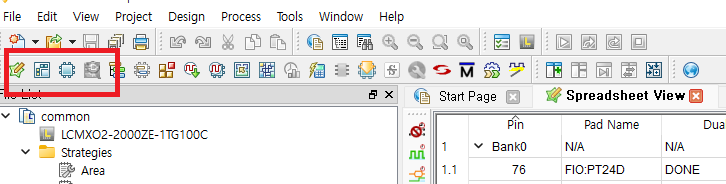
텍스트이(가) 표시된 사진

자동 생성된 설명

* Strategies
* 컴파일 관련 설정 / 자세한건 나중에 알아보자
* Impl1
* 프로젝트 명 / 하위에 기타 파일 내포함
* Input Files
* 베릴로그 코드가 있는 곳
* Synthesis Constraint Files
* 시놉시스 디자인 컴파일러에서 설정할 파일이 위치함
* LPF constraint Files
* Lattice Project Files

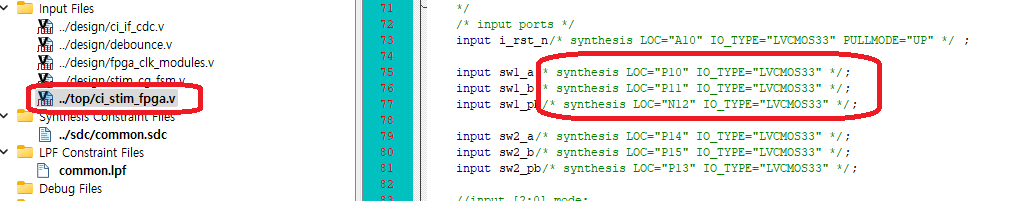
## GPIO 핀 설정하기

### 개발환경에서 GUI 설정



Spreadsheet View에서 GPIO를 설정한다.

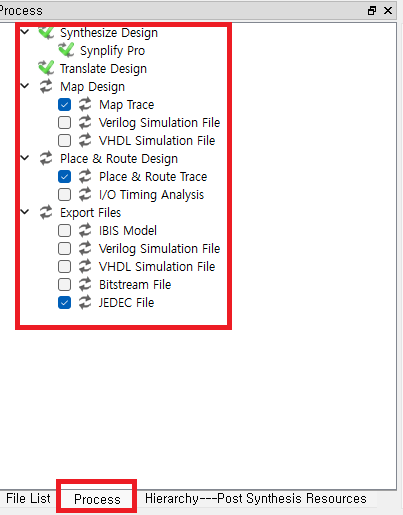
### 그 외 방법



코드에서 /\* synthesis LOC="P10" IO\_TYPE="LVCMOS33" \*/; 등을 넣어 설정한다.

이건 취향대로 해라. 솔직히 코드로 일일이 하는 것 보다 GUI로 설정하는게 더 빠를 듯 하다.

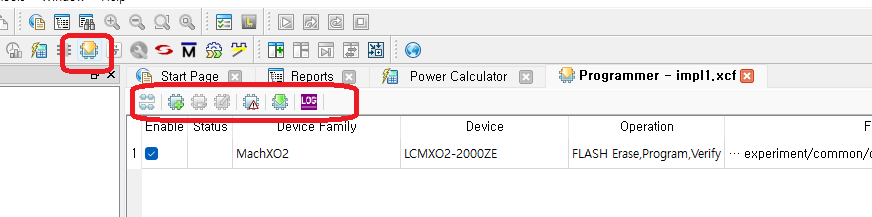
## 프로그램 빌드 하기(디자인 검사)



화면 좌측의 Process 탭을 연 다음 위의 절차를 실행한다.

발생하는 에러, 경고 메시지 관련 로그는 전부 필독한다. 거기에서 상태에 대한 정보가 나오니까.

## 프로그램 넣기



해당 아이콘을 눌러 진입 후 진행한다. 에러 발생시 확인 후 해결해라. 이상.