논리회로 설계 및 실험

2주차

2주차 목표

목표

1. 문제에서 논리식을 유도해내고 논리회로로 구현
2. 논리회로 설계를 위한 Flowrian 사용법 학습

부산대학교
PUSAN NATIONAL UNIVERSITY

논리회로

1. 주어진 입력에 대해 논리 연산을 수행하여 원하는 결과를 출력하는 회로
2. Boolean 식을 논리 게이트를 사용하여 물리적으로 구현한 전자회로

논리회로 설계

- 설계 방법 및 설계 툴이 다양함
- 논리회로 설계 및 실험 과목에서는 Flowrian이라는 툴을 사용하여 회로를 설계함

Flowrian 소개

- 인터넷 기반의 하드웨어 설계 툴
- HDL(Hardware Description Language) 텍스트 에디터 뿐만 아니라 그래피컬한 환경의 설계 툴을 지원

Flowrian 구성

Flowrian

: 전체 설계 프로젝트 관리

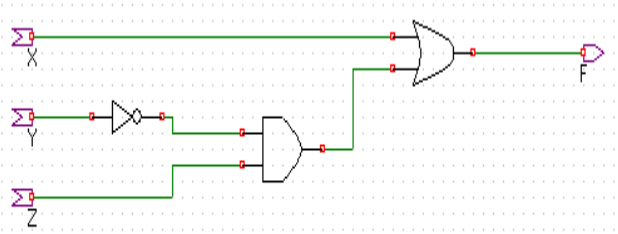
- **Schmian** : 논리회로도 편집기, 심벌들을 사용해 회로를 작성
- **Stadian** : 상태도 편집기, 컨트롤 유닛을 작성
- **Symbolian** : 심벌 편집기, 작성한 모듈을 심벌로 만들어 재사용
- **Stimulian** : 테스트벤치 입력파형 편집기
- **Wavian** : 테스트벤치 파형 분석기

부산대학교 PUSAN NATIONAL UNIVERSITY

논리식과 논리회로

$$F(X,Y,Z) = X + \bar{Y}Z$$

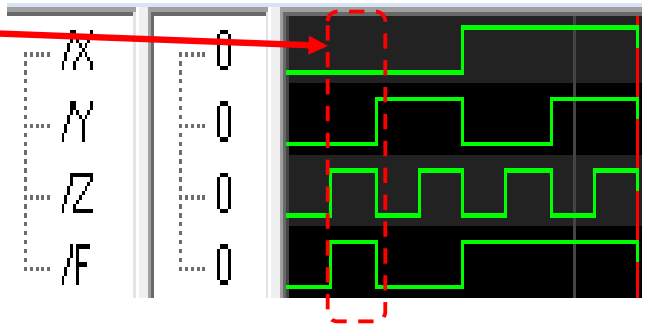
“합항은 OR 게이트”
곱항은 AND 게이트
보수는 NOT 게이트



Flowrian 툴로 설계한 $F(X,Y,Z) = X + \bar{Y}Z$ 의 회로도

X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$F(X,Y,Z) = X + \bar{Y}Z$ 의 진리표



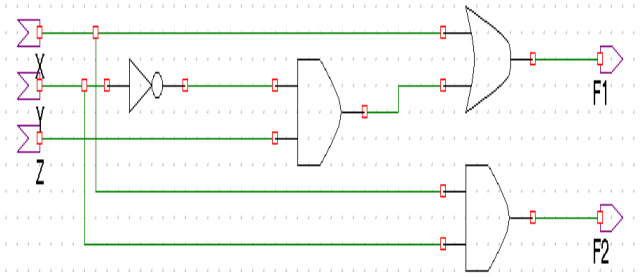
Flowrian 툴로 시뮬레이션한 $F(X,Y,Z) = X + \bar{Y}Z$ 의 파형

부산대학교 PUSAN NATIONAL UNIVERSITY

논리식과 논리회로

$$F_1(X,Y,Z) = X + \bar{Y}Z$$
$$F_2(X,Y) = XY$$

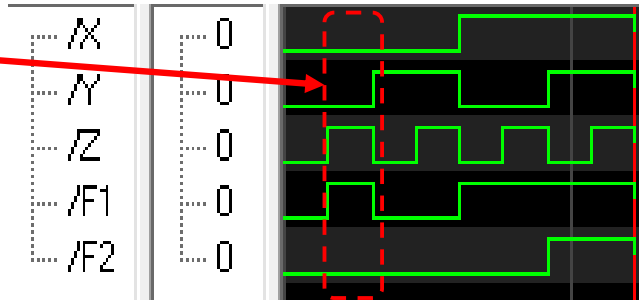
“ 두 개 이상의 논리식도 동시에 표현 “



Flowrian 툴로 설계한 F_1, F_2 두 논리식의회로도

X	Y	Z	F_1	F_2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	1
1	1	1	1	1

F_1, F_2 두 논리식의 진리표

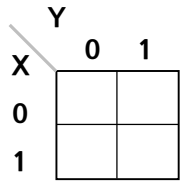


Flowrian 툴로 시뮬레이션한 F_1, F_2 두 논리식의파형

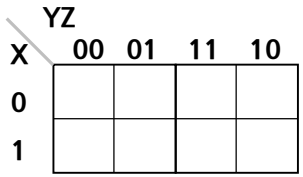
부산대학교 PUSAN NATIONAL UNIVERSITY

카르노맵(Karnaugh map, K-map)

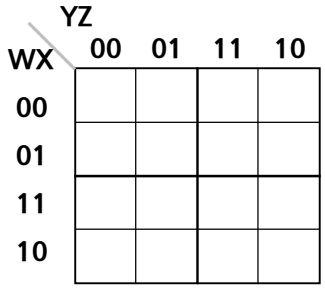
적은 개수의 변수에 한해 논리식의 함수를 유도해내는 방법 중 하나



2변수 카르노맵



3변수 카르노맵



4변수 카르노맵

카르노맵 응용

예시 문제

1bit 입력 A와 B를 비교하여,
A>B이면 LED1, A=B이면 LED2, A<B이면 LED3이 켜지는 1bit 비교기 설계

부산대학교 PUSAN NATIONAL UNIVERSITY

예시 문제 접근

입력 : A, B
출력 : LED1, LED2, LED3

① A>B 인 경우는 A가 1이고 B는 0
② A=B 인 경우는 A, B가 0 또는 1
③ A<B 인 경우는 A가 0이고 B는 1

①

A \ B	0	1
0		
1	1	

②

A \ B	0	1
0	1	
1		1

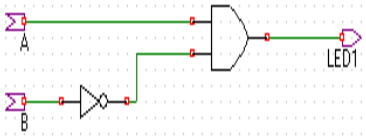
③

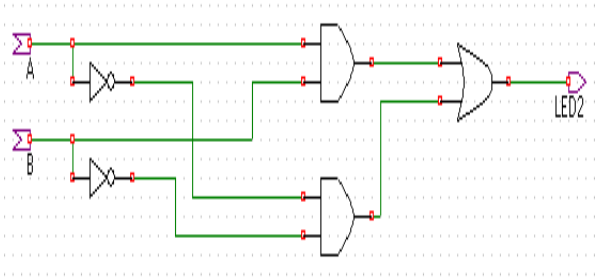
A \ B	0	1
0		1
1		

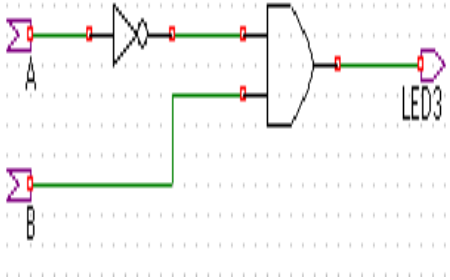
① $LED1 = A\bar{B}$

② $LED2 = AB + \bar{A}\bar{B}$

③ $LED3 = \bar{A}B$







부산대학교 PUSAN NATIONAL UNIVERSITY

잘못된 회로 설계의 예

예시
입력 : A, B
출력 : LED1

A버튼을 누르거나 B버튼을 누르면 LED1이 켜지는 회로

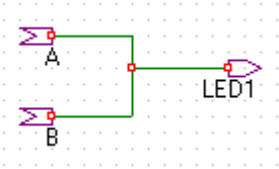


Fig1. 두 개의 입력이 하나의 출력에 연결

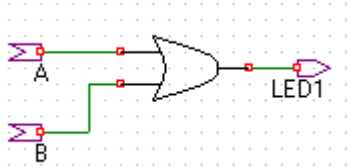


Fig2. 두 개의 입력을 논리 게이트로 연결

Fig1은 LED1에서의 출력이 0인지 1인지 알 수 없으므로 x (don't care)를 출력하여 설계 의도와 다름
Fig2는 두 개의 입력을 논리 게이트로 연결하여 정상 동작함

참고로 아래 Fig3과 같이 하나의 입력에 둘 이상의 출력은 정상 동작함

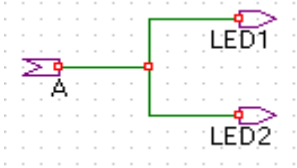


Fig3. 하나의 입력에 두 개 이상의 출력

부산대학교 PUSAN NATIONAL UNIVERSITY

Logic Array 형식의 설계

$$F_1(X, Y, Z) = X + \bar{Y}Z$$
$$F_2(X, Y) = XY$$

Slide11과 같이 회로도가 복잡해질수록 좌측의 설계 방식은 디버깅이 힘듦
Logic Array와 유사한 우측의 설계 방식을 권함

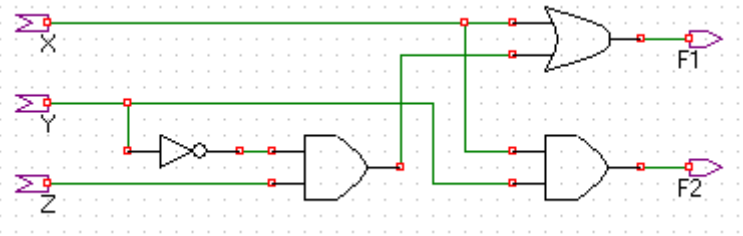


Fig1. 기본적인 설계 드로잉

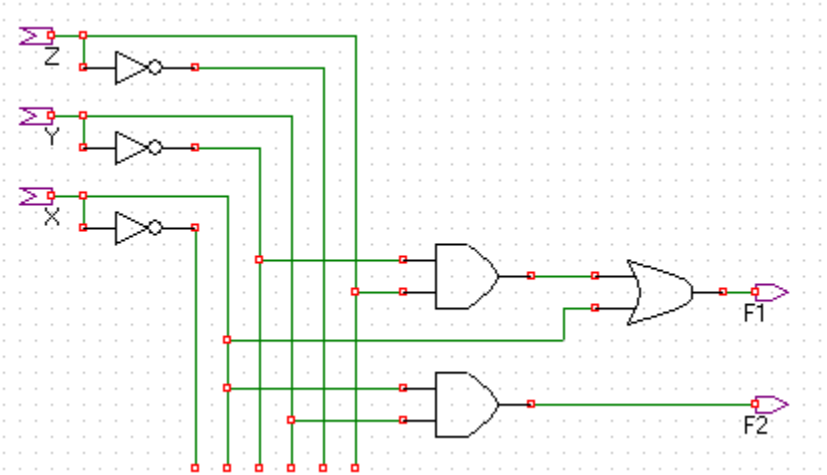
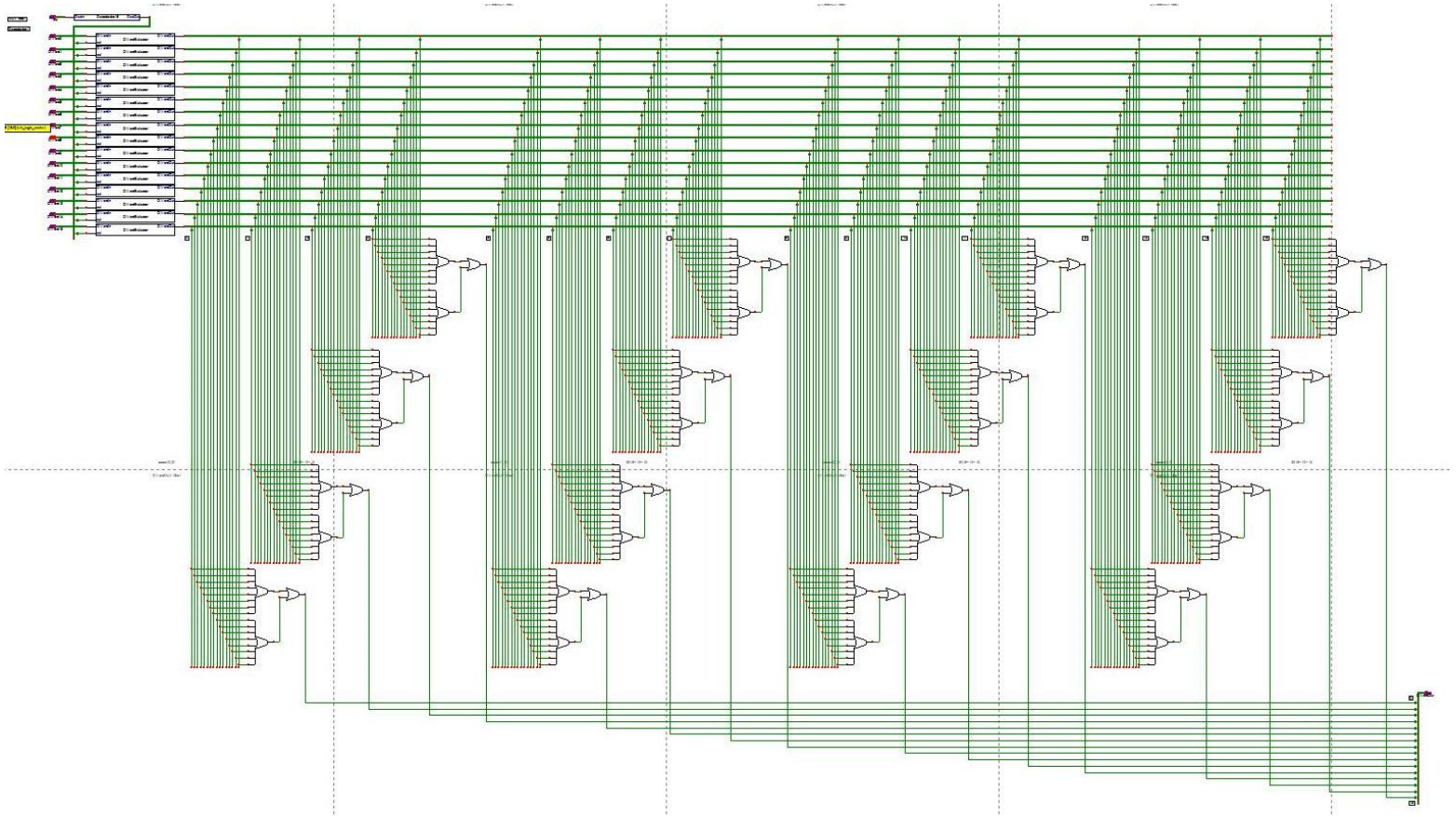


Fig2. 입력 포트의 배선을 순서대로 정리한 드로잉

부산대학교 PUSAN NATIONAL UNIVERSITY

작년 수업에 사용한 256bit 메모리의 일부

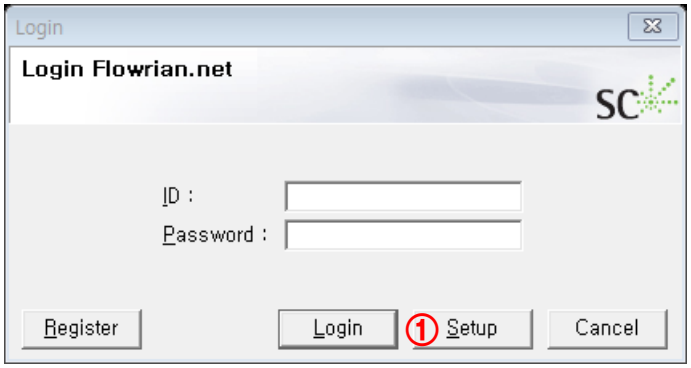


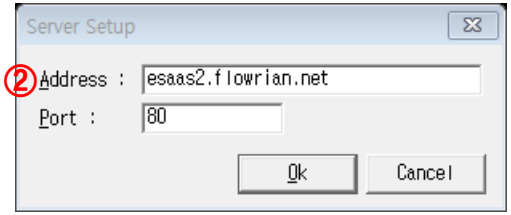
실습

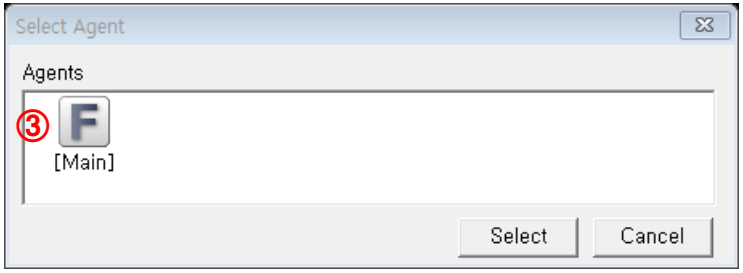
Flowrian을 이용한 회로 설계

Flowrian 로그인

① Setup을 선택
② Address를 “esaas2.flowrian.net”으로 수정 후 로그인
③ [Main] 을 선택하여 Flowrian 실행

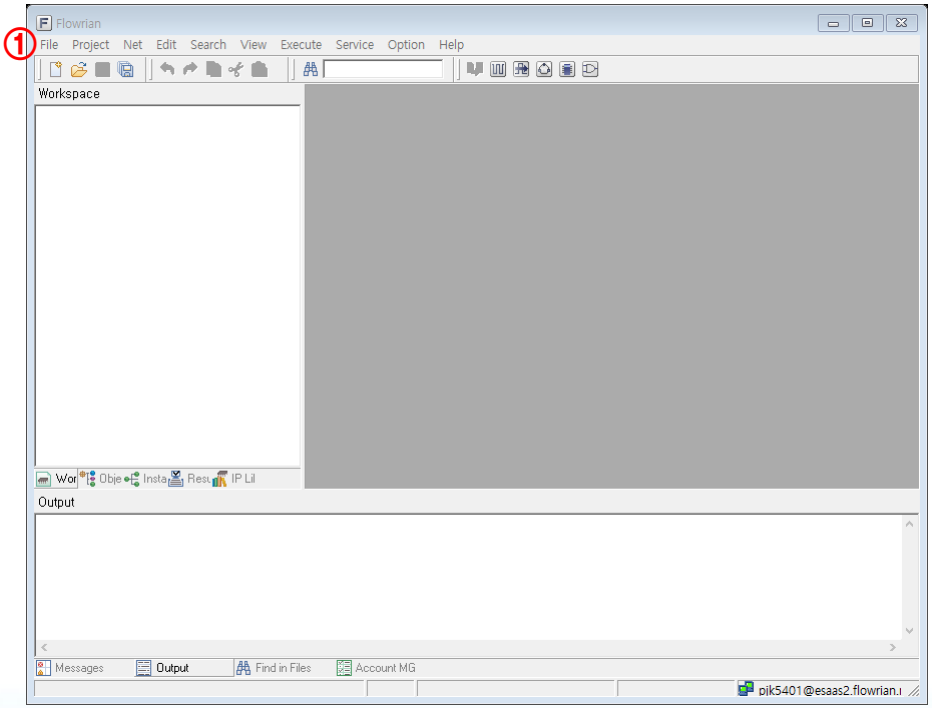


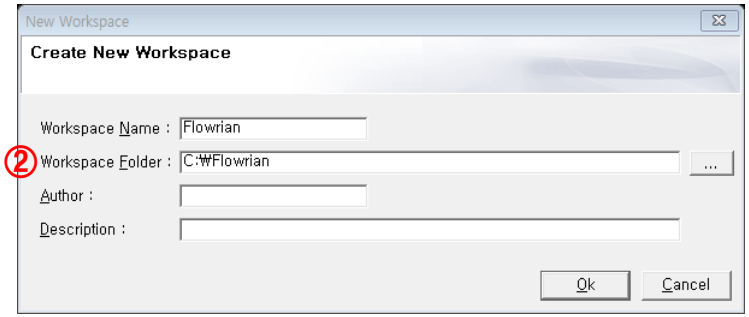




Flowrian Workspace 생성 (1/2)

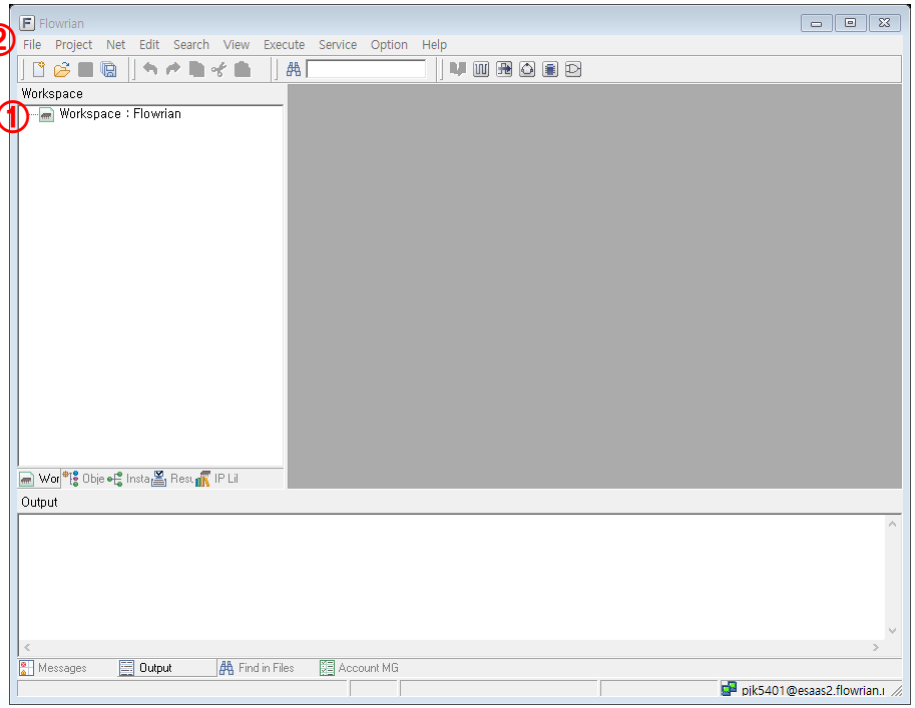
Workspace는 최상위 작업 공간
① [File]에서 [New workspace] 선택
② Workspace의 경로와 이름을 지정

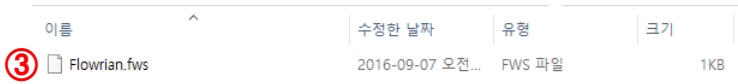




Flowrian Workspace 생성 (2/2)

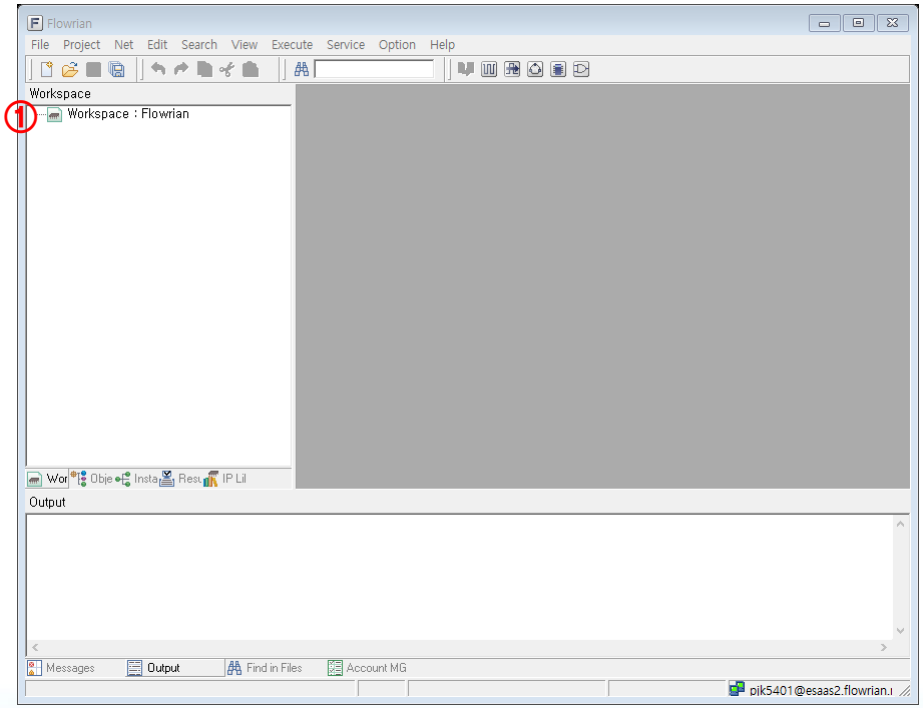
① Workspace가 생성됨
② [File] -> [Save Workspace]를 실행하면
③ 해당 경로에 .fws 파일이 생성됨

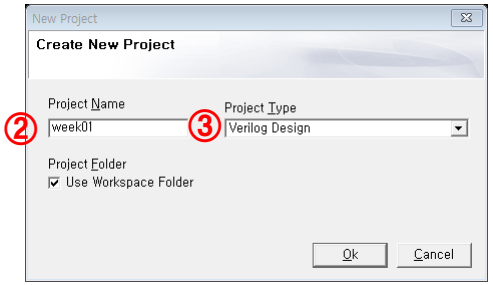




Flowrian Project 생성 (1/2)

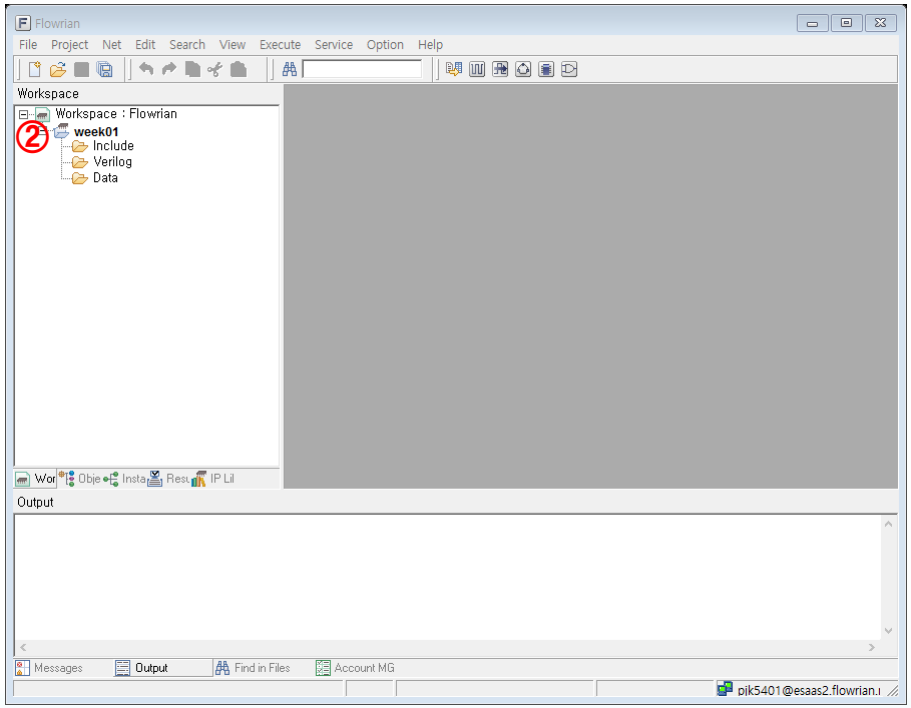
하나의 Workspace가 여러 개의 프로젝트를 가질 수 있음
① Workspace에서 우클릭 후 [Add new project] 실행
② 프로젝트 명을 지정
③ 프로젝트 타입은 Verilog Design

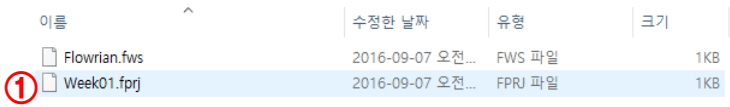




Flowrian Project 생성 (2/2)

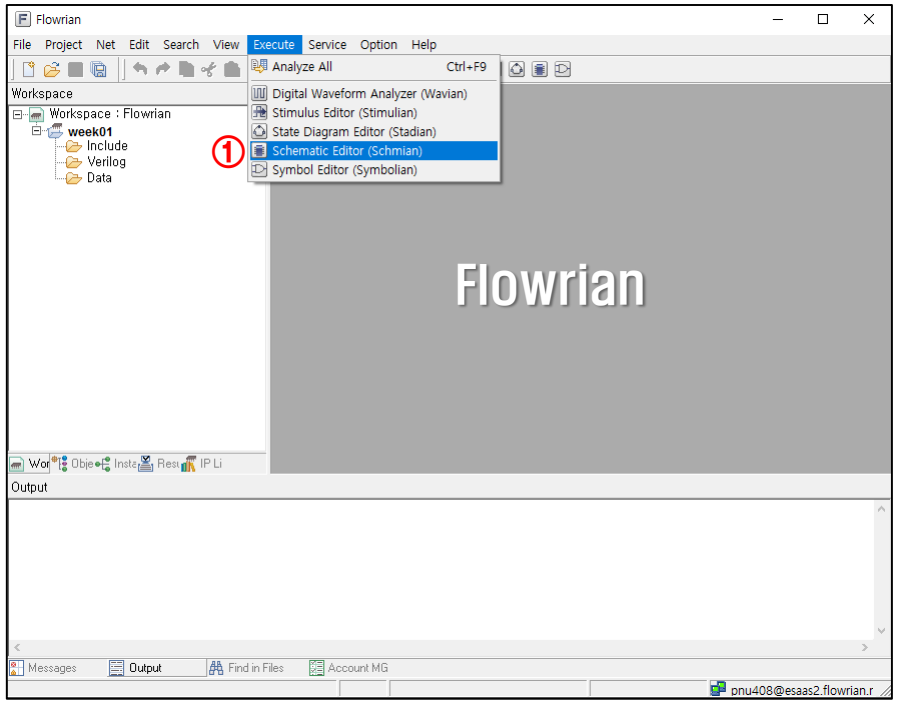
① 프로젝트를 생성하면 workspace 폴더에 .fprj 파일이 생성됨
② 하나 이상의 프로젝트가 있을 때, 프로젝트에서 우클릭 후
[Set Project Active]를 통해 해당 프로젝트 활성화

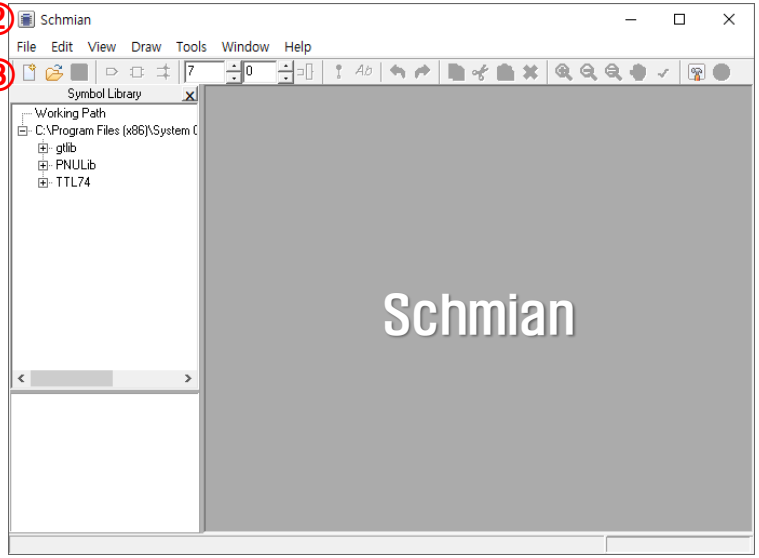




회로도 생성 (1/2)

① Flowrian에서 [Execute] -> [Schematic Editor] 실행
② Schmian 툴이 실행 됨
③ Schmian에서 [File] -> [New] 또는 [New Document] 아이콘 실행





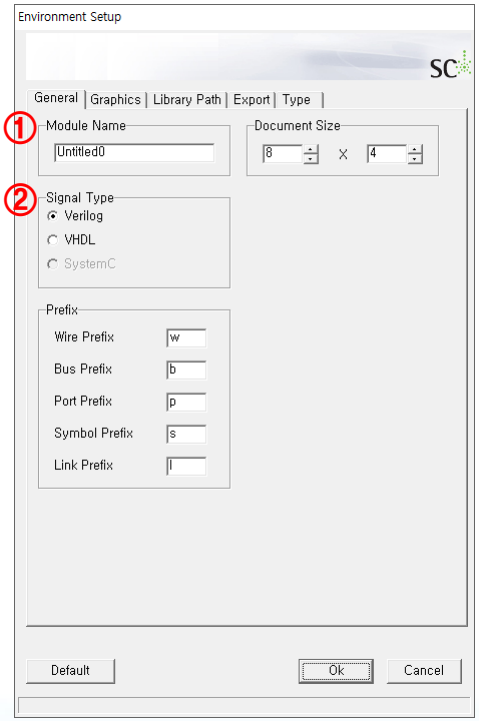
부산대학교 PUSAN NATIONAL UNIVERSITY

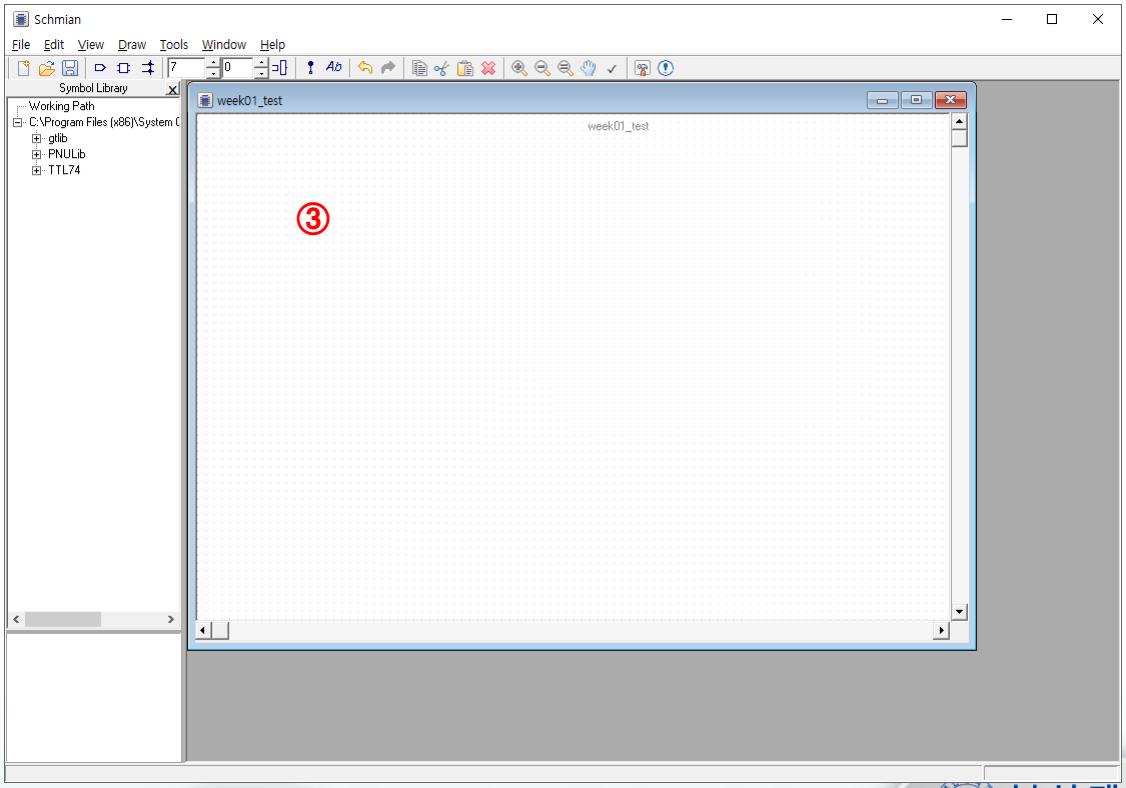
회로도 생성 (2/2)

① Module Name 지정
- 모듈명은 영어로 시작해야 하며 숫자, 또는 특수기호로 시작할 수 없음
- 올바른 모듈명이 아니면 정상 동작을 하지 않고 Compile error가 발생

② Signal Type은 Verilog 선택

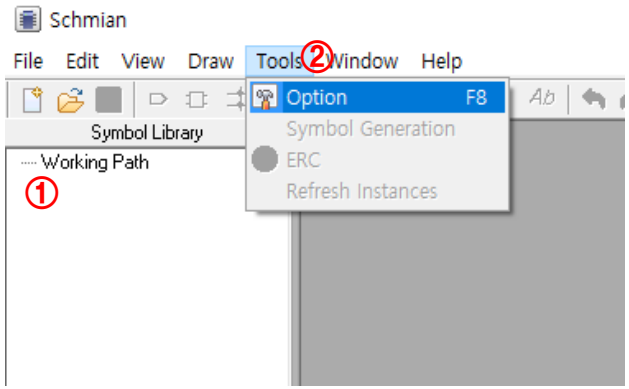
③ 신규 회로도 생성

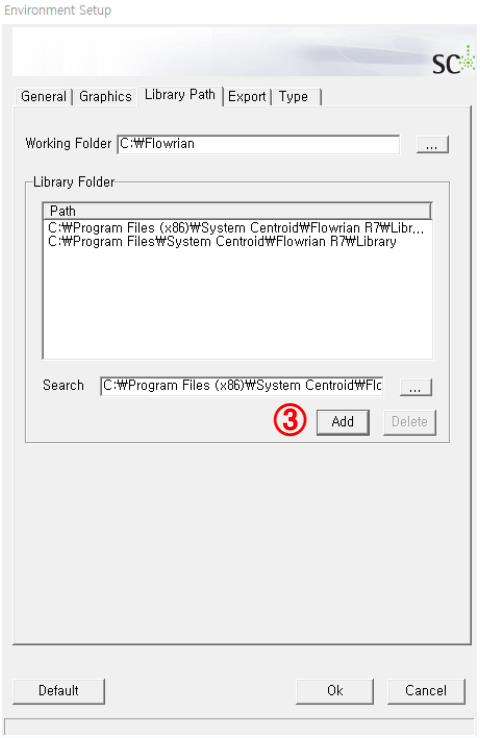




참고

① Working Path에 라이브러리가 보이지 않으면
② Schmian에서 [Tools] -> [Option] 선택
③ 라이브러리 경로 확인 후 [Add] -> [Ok] 클릭
- C:₩Program Files (x86)₩System Centroid₩Flowrian R7₩Library
- 위 경로에 플로리안이 설치되어 있을 수도 있음

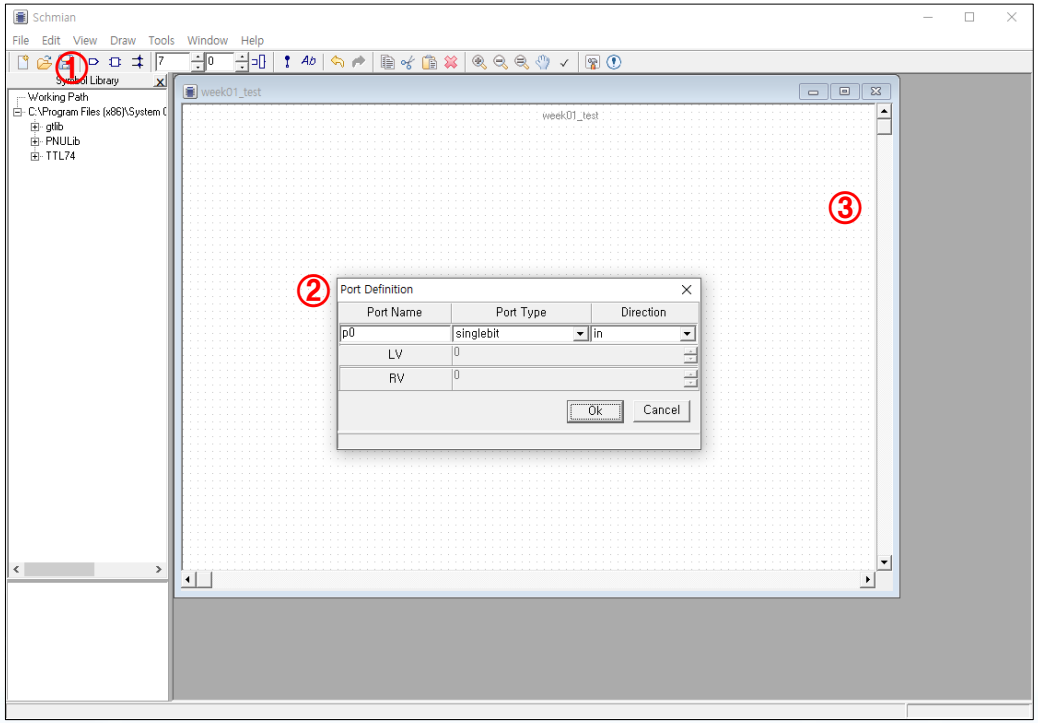




부산대학교 PUSAN NATIONAL UNIVERSITY

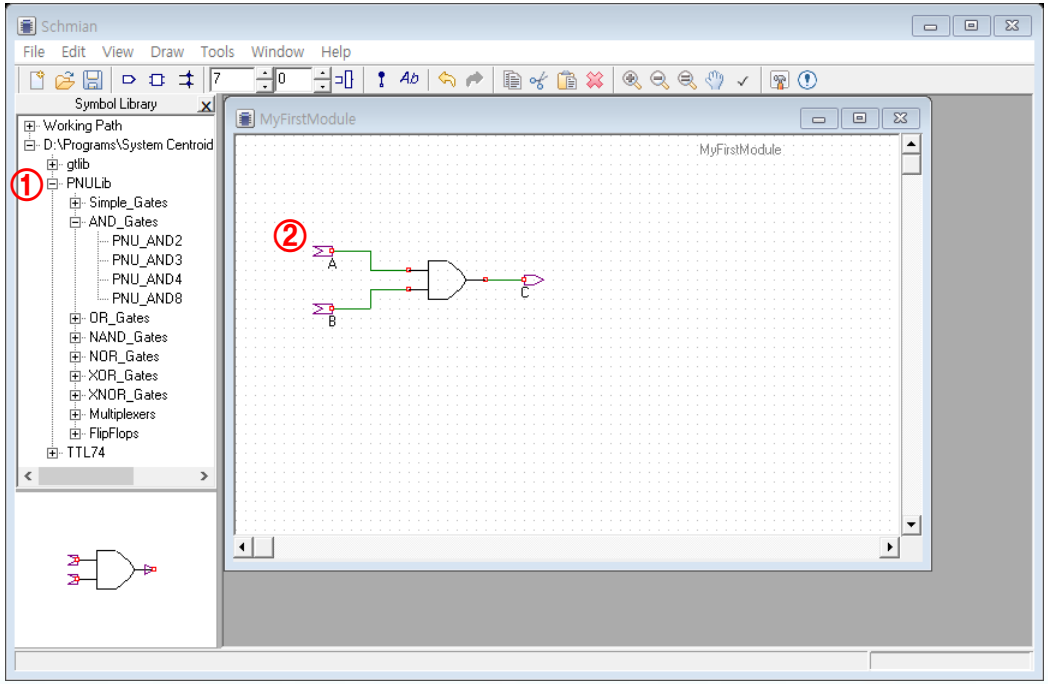
회로도 작성 (1/2)

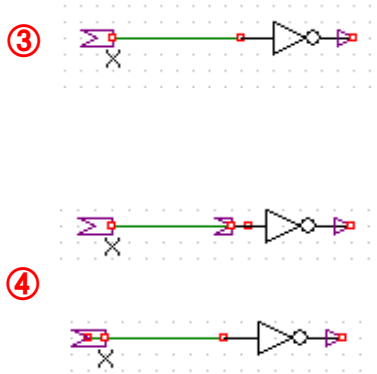
① [Draw] -> [Port] 또는 Port 아이콘 실행
② Port Definition
- Port Name : 모듈명 작성 규칙과 같음
- Port Type : singlebit 또는 multibit 선택
- Direction : 입력 또는 출력 선택



회로도 작성 (2/2)

① 논리 게이트는 PNULib에서 선택하여 Drag&Drop
② 입출력 포트와 논리 게이트간에 선을 이어 회로 작성
③ 정상적인 연결
④ 연결되지 않음



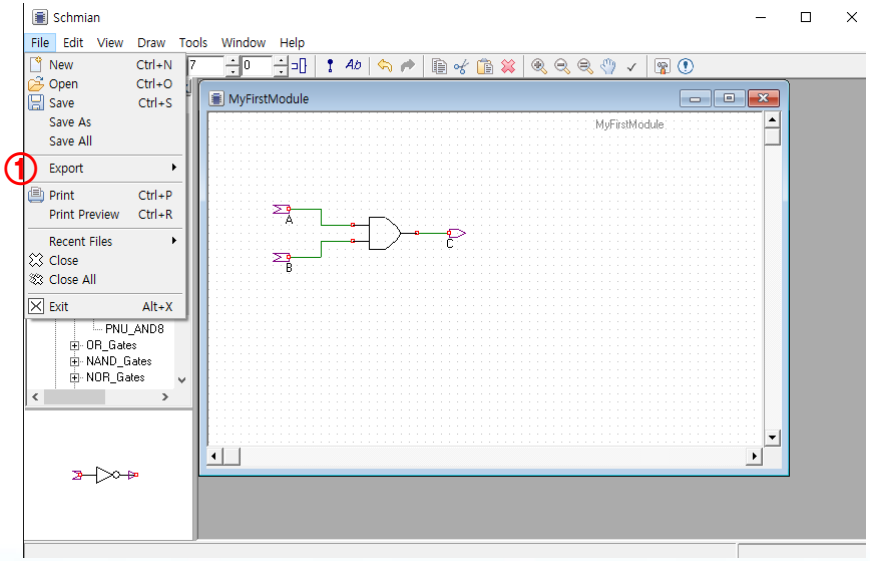


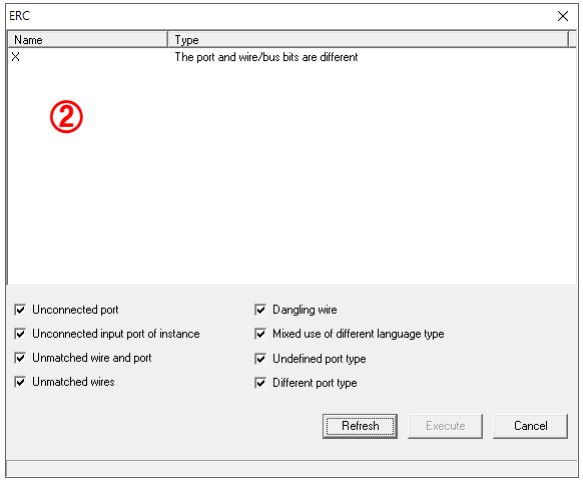
회로도 저장 및 변환

① [Export] -> [Verilog] -> [ISim] 실행
- 파일명 지정 후 저장 하면 .v 파일 생성
- 이 때, 파일명은 회로도의 모듈명과 일치시킬 것

② 연결이 끊어졌거나 포트 등의 문제가 있을 시 ERC 창에서 에러 출력

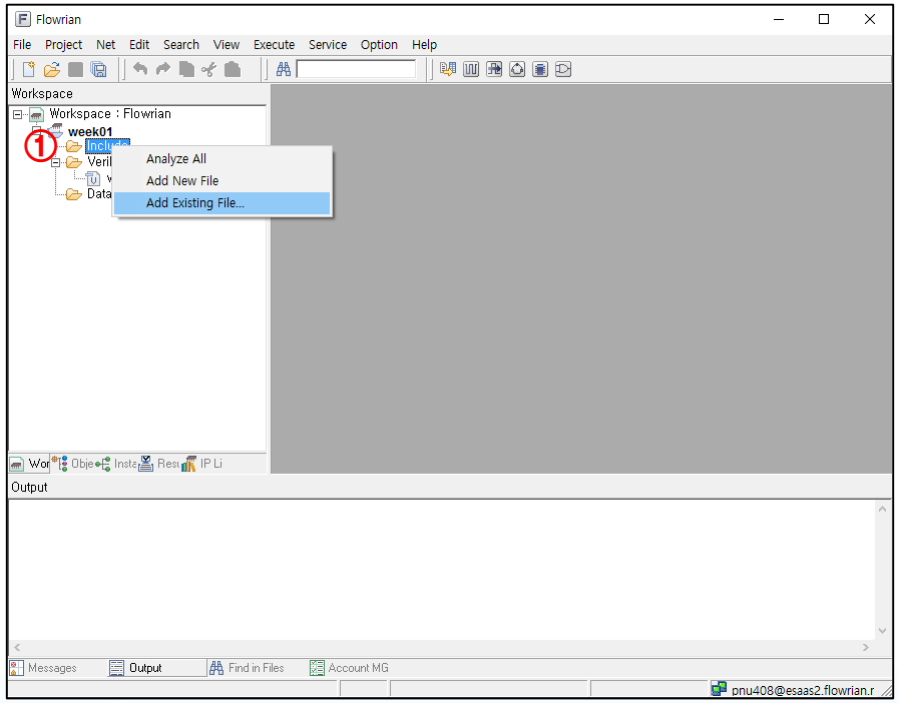
※ Save와 Export의 차이
- [Save] 는 회로도(.sch)를 저장
- [Export] 는 회로도를 HDL 파일(.v)로 변환 후 저장



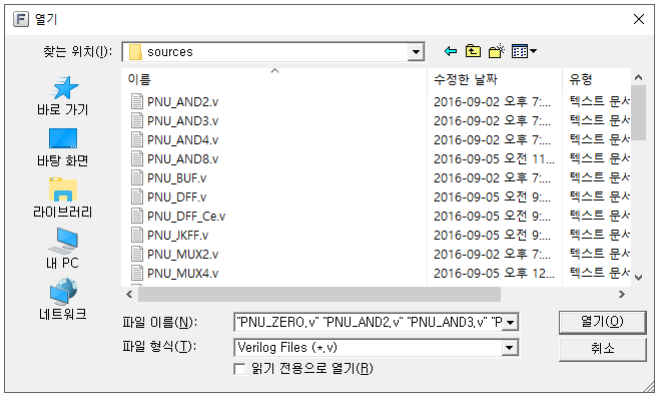


프로젝트에 라이브러리 추가

① 프로젝트의 Include 폴더에서 우클릭 후 [Add Existing File] 실행
② C:₩Program Files₩System Centroid₩Flowrian R7₩Library₩PNULib₩sources 경로의 모든 .v 파일 선택

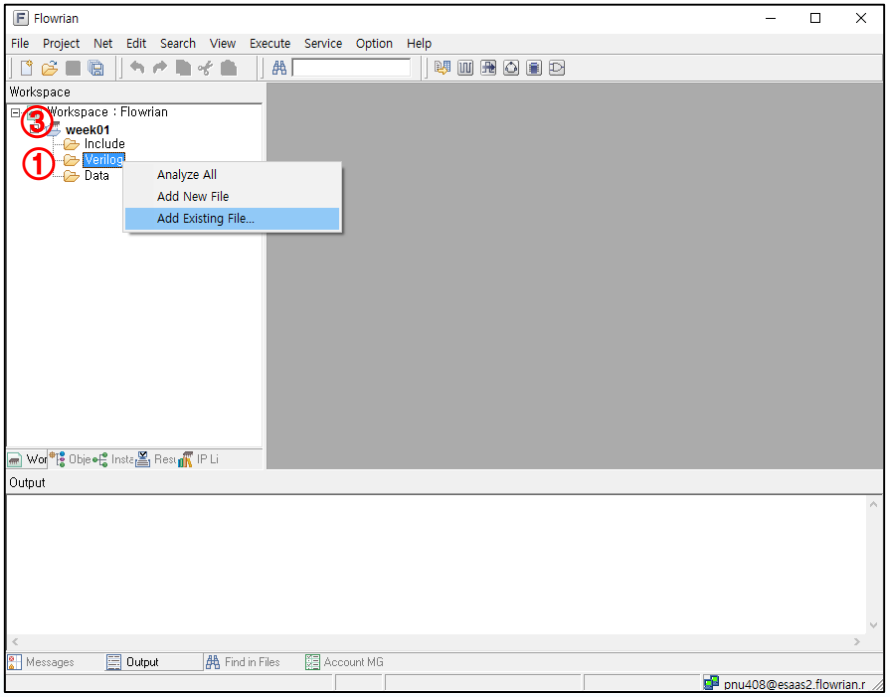


②

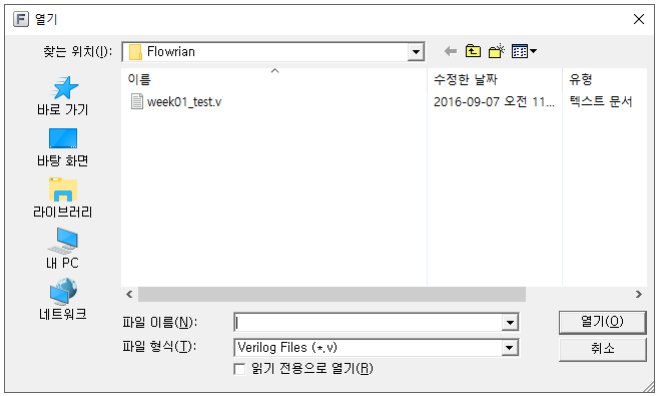


프로젝트에 작성한 모듈 추가

① 프로젝트의 Verilog 폴더에서 우클릭 후 [Add Existing File] 실행
② Schmian에서 작성 후 HDL(.v)로 변환한 파일을 선택
③ 프로젝트에서 우클릭 후 [Analyze All] 명령 실행
- Analyze를 하고 나면 Workspace에 _analyze 라는 폴더가 생김
- _analyze 폴더 안에는 프로젝트에 추가한 모듈을 해석한 내용이 .psr 파일로 저장됨

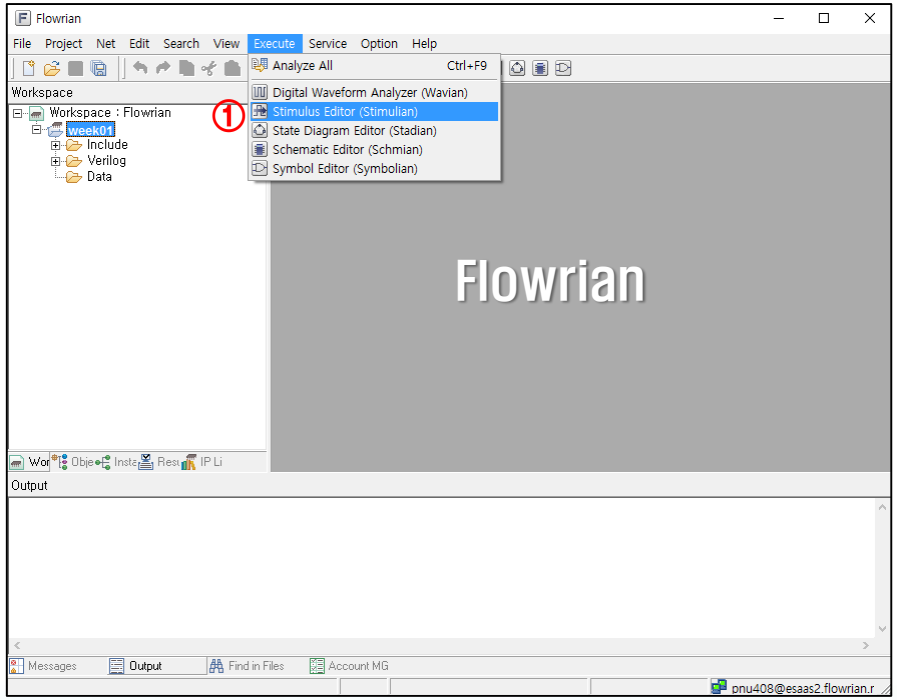


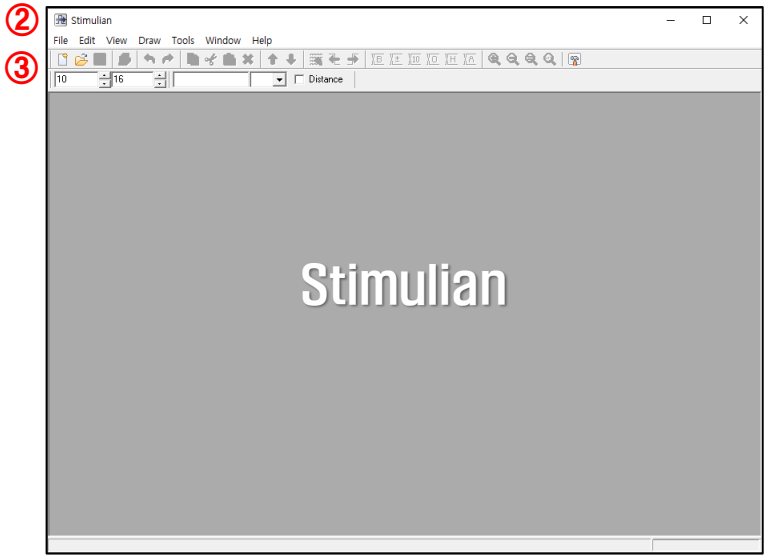
②



모듈 테스트벤치 작성 (1/2)

① Flowrian에서 [Execute] -> [Stimulus Editor] 실행
② Stimulian 실행
③ Stimulian에서 [File] -> [Synchronize] 실행
④ Flowrian에서 Analyze 후 생성된 .psr 파일을 선택

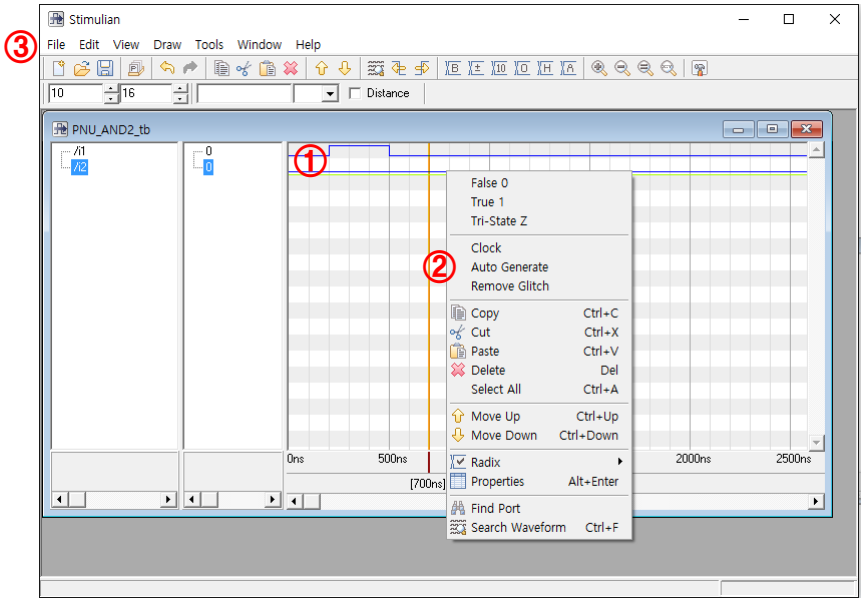




부산대학교 PUSAN NATIONAL UNIVERSITY

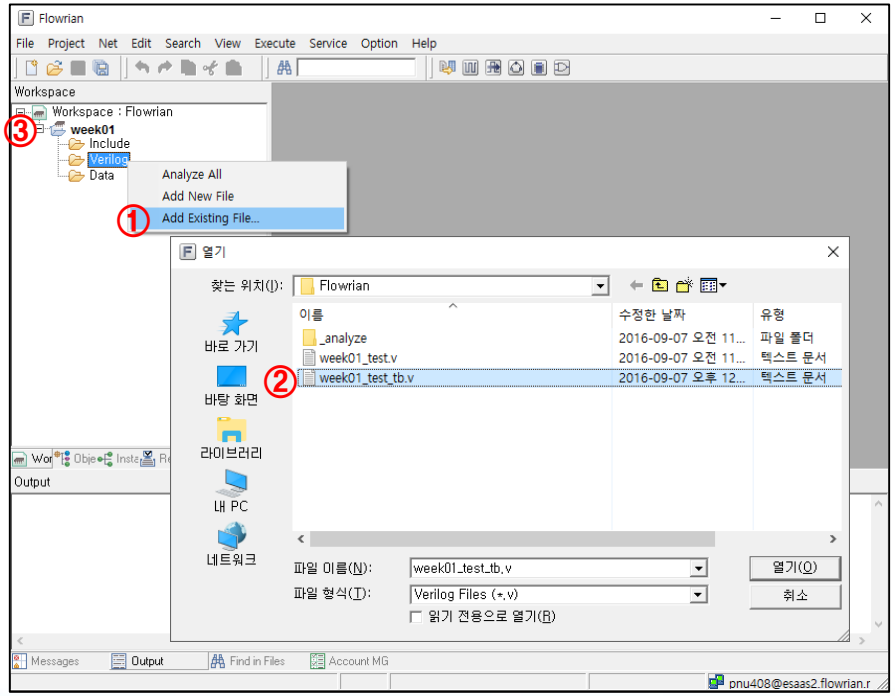
모듈 테스트벤치 작성 (2/2)

Stimulian에서는 모듈의 입력 포트로 보낼 신호를 생성함
① 파형을 클릭하여 high 또는 low 신호를 지정
② 우클릭하여 Clock이나 Auto Generate 등의 메뉴에서 반복적인 신호 생성 가능
③ [File] -> [Export] -> [Verilog] 를 실행하여 HDL(.v) 파일로 변환하여 저장
이때, 파일명은 “모듈명_tb.v” 파일로 정해짐



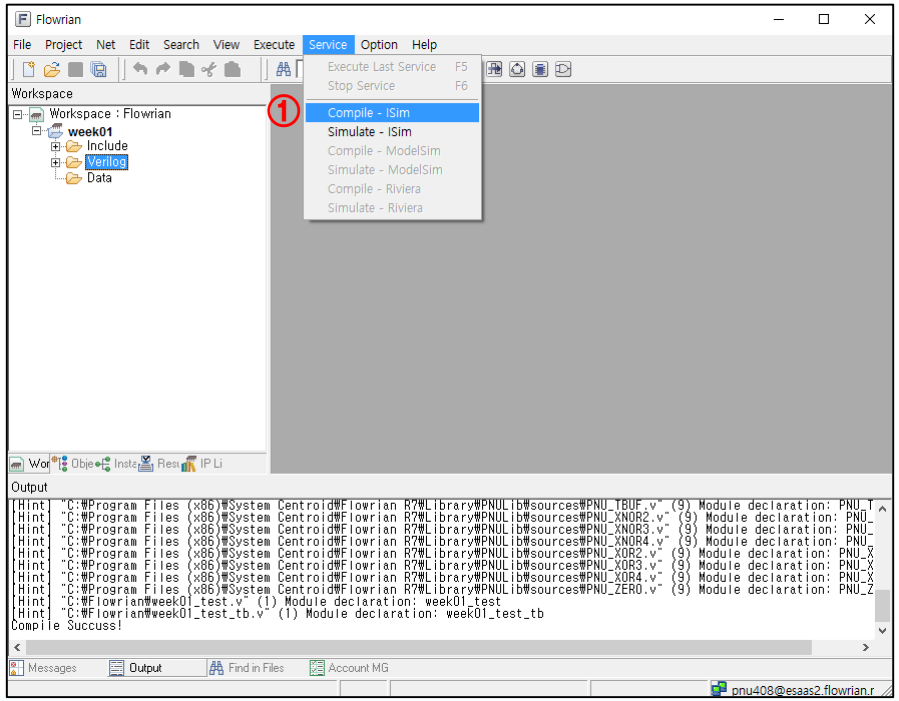
프로젝트에 작성한 테스트벤치 파일 추가

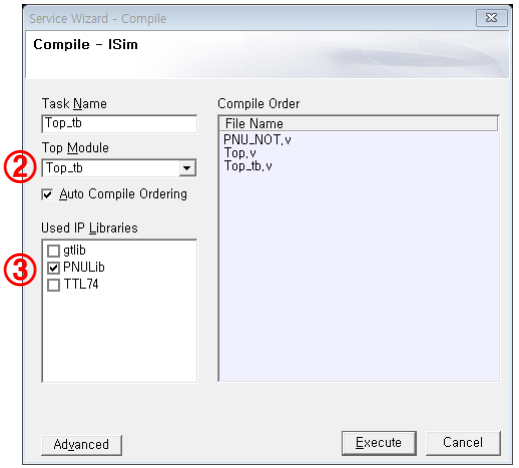
① 프로젝트의 Verilog 폴더에서 우클릭 후 [Add Existing File] 선택
② Stimulian에서 작성한 HDL(~_tb.v)로 변환한 파일을 선택
③ 프로젝트에서 다시 한번 [Analyze All] 실행



테스트벤치 컴파일

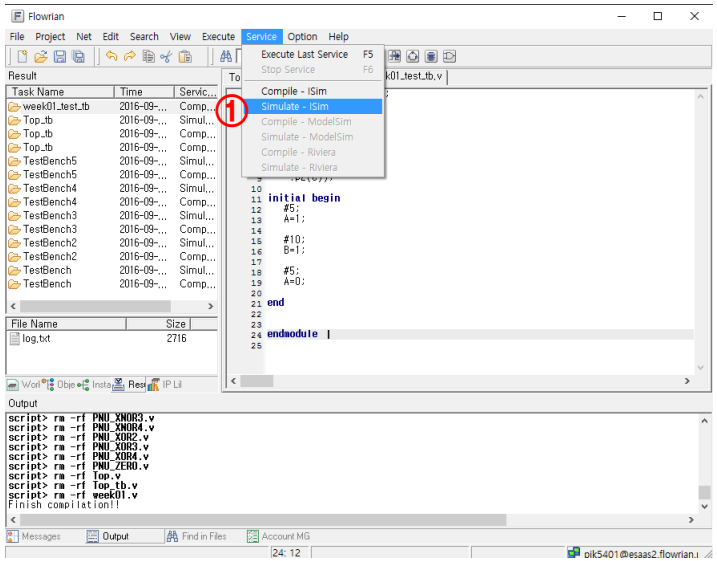
① Flowrian에서 [Service] -> [Compile - ISim] 실행
② Top Module에 컴파일 할 테스트벤치 선택
③ 라이브러리는 PNULib 선택

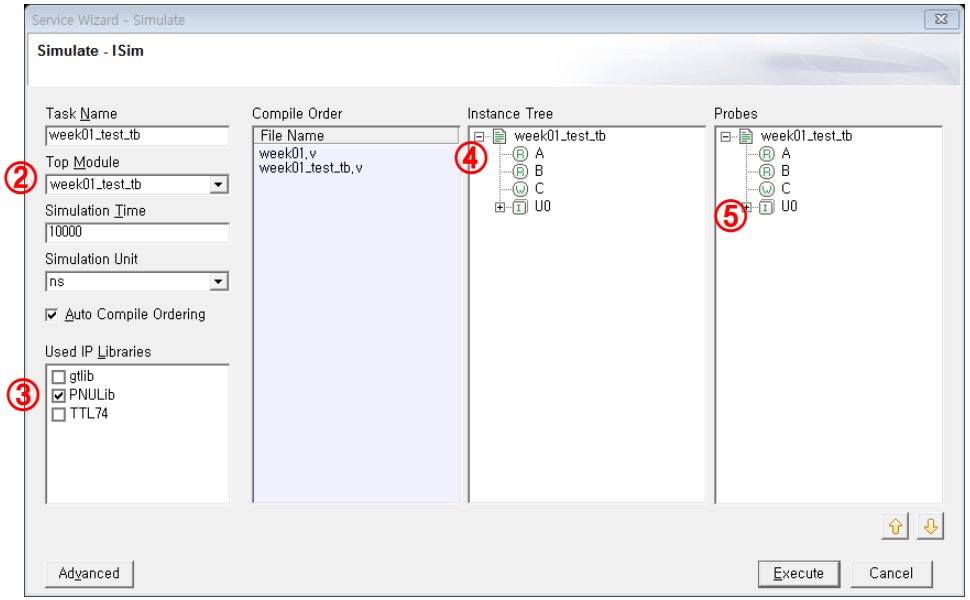




테스트벤치 시뮬레이션 (1/3)

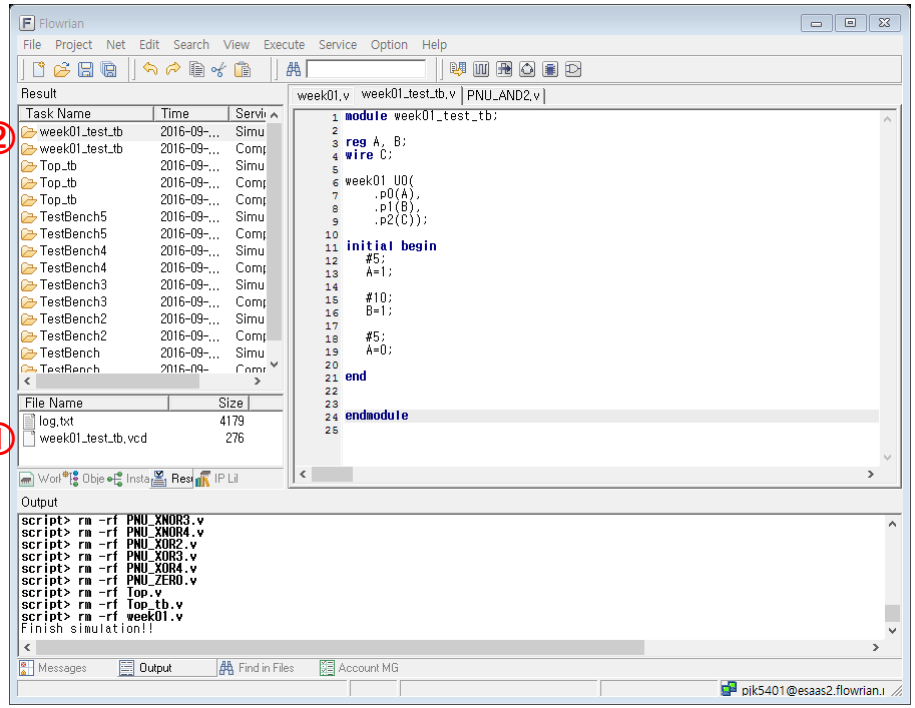
① Flowrian에서 [Service] -> [Simulate - ISim] 실행
② Top Module에 시뮬레이션을 실행할 테스트벤치 선택
③ 라이브러리는 PNULib 선택
④ 파형을 확인할 포트를 더블클릭
⑤ 포트만 남기고 나머지는 삭제해도 됨





테스트벤치 시뮬레이션 (2/3)

① 시뮬레이션 해석이 성공하면 .vcd 파일을 생성
.vcd 파일을 더블 클릭하면 Wavian이 실행되며 파형을 확인
② .vcd 파일이 생성 안되면 컴파일과 시뮬레이션 로그를 확인하여 에러를 확인



테스트벤치 시뮬레이션 (3/3)

① 적당하게 척도를 조절하여 파형을 확인
② 포트명을 우클릭하여 Radix(16진법, 10진법 등)를 변경하거나 파형의 색을 변경 가능