

# 논리회로 설계 및 실험

3주차

# 3주차 목표

## 목표

1. Encoder와 Decoder 및 가산기에 대한 이해
2. 반가산기와 전가산기를 이용하여 구조적 설계를 이해하고 이를 활용한 HDL 모듈의 확장 실습

## 문제점

1. 실습파일 전부 업로드(회로도 파일도 저장 필수)
2. 검사 다 받고 이메일 전송(검사 안받고 보내면 점수없음)
3. 같은 조라도 같은파일 제출 금지(둘다 점수없음)

# Encoder와 Decoder

## Encoder

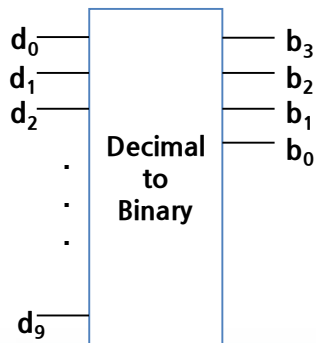
- 어떤 정보의 형태나 형식을 부호화(encoding)하여 다른 형태나 형식으로 변환하는 장치
- 처리속도 향상이나 데이터 압축 또는 데이터의 손실 방지를 위해서도 사용됨

## Decoder

- Encoder로 변환한 정보를 그에 대응하는 원래의 정보로 복호화(decoding)하여 주는 장치

## Encoder의 예

- 디지털 사진을 찍으면 실제로는 렌즈에 맺힌 상(analog)이 픽셀정보(digital)로 변환되어 저장됨
- 지난 2주차 실험에서는 십진수 정보를 이진수 형태(BCD code)로 변환하는 변환기를 구현함



Decimal-to-Binary  
변환기의 심벌

## Simple Encoder

- Simple Encoder는 one-hot code를 binary 정보로 변환함
- 이때  $2^n$ 개의 입력에 있어서 n개의 출력이 나옴

## Priority Encoder

- 입력 bits의 MSB부터 출발하여 0이 아닌 첫 번째 bit의 index가 출력값이 됨
- 이때 해당 bit가 아닌 다른 bit 값들은 무시되며 압축(손실)이 일어남

4 to 2 Simple Encoder

$I_3$	$I_2$	$I_1$	$I_0$	$O_1$	$O_0$	V
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	1	0	0	1	0	1
1	0	0	0	1	1	1

4 to 2 Priority Encoder

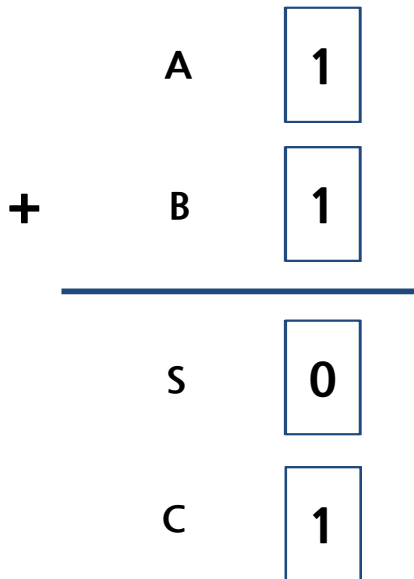
$I_3$	$I_2$	$I_1$	$I_0$	$O_1$	$O_0$	V
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

## 가산기 (Adder)

- 덧셈 연산을 수행하는 논리회로
- 한 자릿수 연산을 위해서는 Half adder, Full adder 등이 있음
- 멀티비트의 연산을 위해서는 Ripple carry adder, Carry look ahead adder 등이 있음

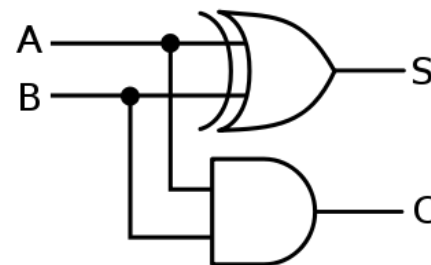
## 반가산기 (Half adder)

- 한 자릿수 덧셈을 수행하고 합(Sum)과 자리올림수(Carry)를 출력
- Carry는 AND gate, Sum은 XOR gate와 결과가 같음



반가산기의 진리표

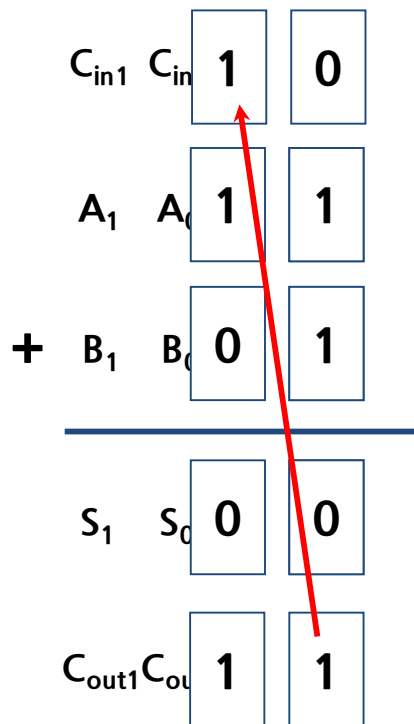
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



반가산기의 논리회로

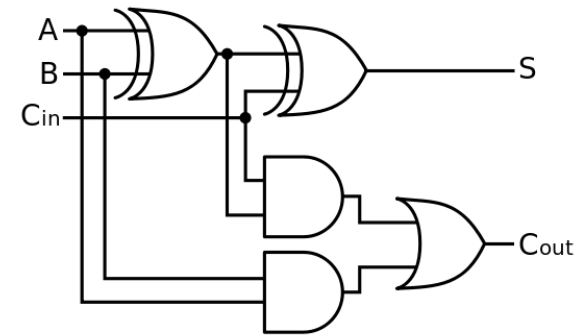
## 전가산기 (Full adder)

- 한 자릿수 덧셈을 수행할 때 이전 자리의 연산 결과로 받은 자리올림수(Carry)를 함께 연산하는 회로
- 두 개의 반가산기와 1개의 OR gate로 구성할 수 있음



전가산기의 진리표

A	B	$C_{in}$	$C_{out}$	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



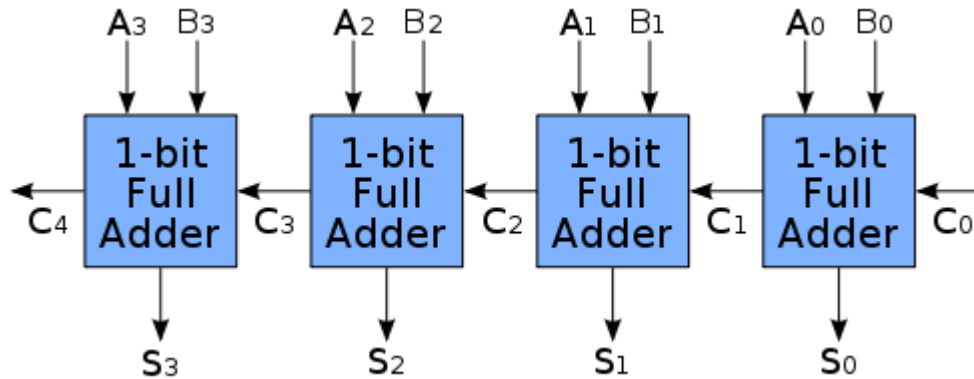
전가산기의 논리회로



# 리플 캐리 가산기

## 리플 캐리 가산기 (Ripple carry adder)

- 복수의 전가산기를 이용하여 복수 비트의 덧셈 연산을 할 수 있는 가산기
- 간단한 구조이지만 전가산기의 입력이 이전 전가산기의 출력이므로 전달 지연이 발생함



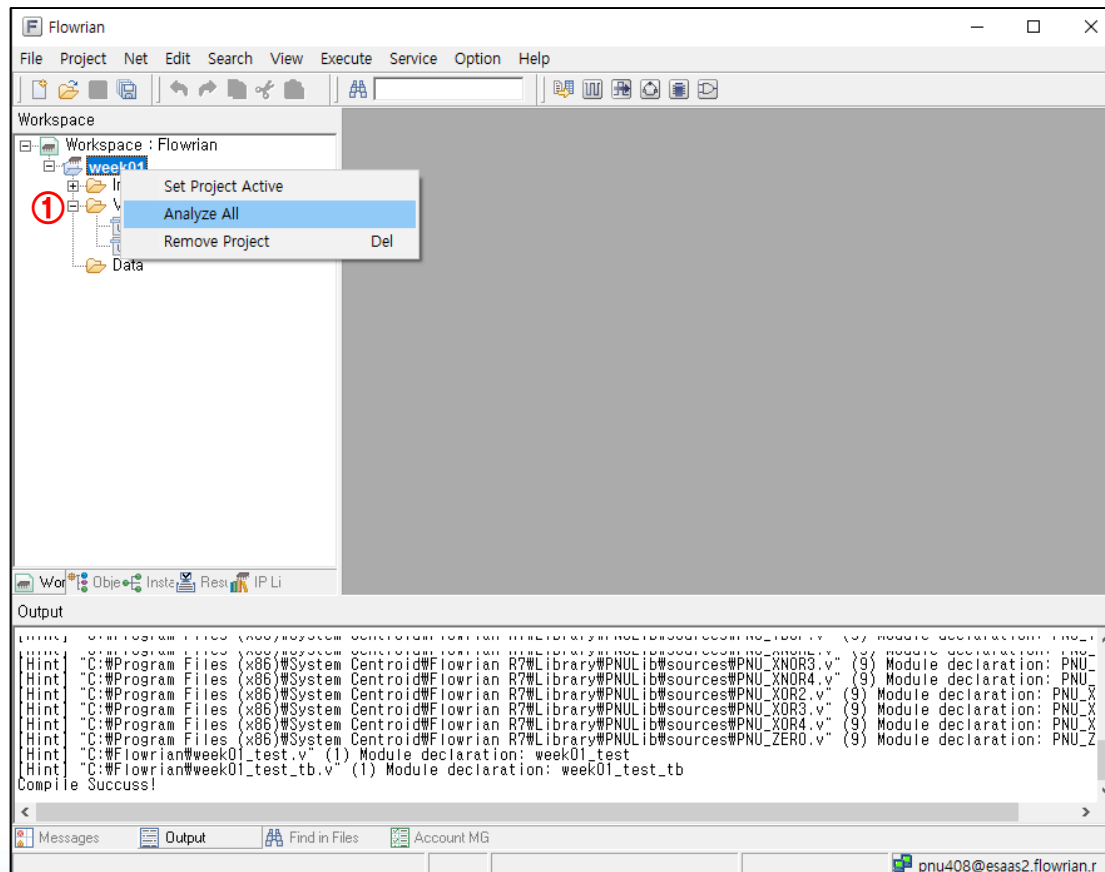
4bit Ripple carry adder의 구조

# 실습

Symbolian을 이용한 심벌 생성

## Symbolian 실행(1/2)

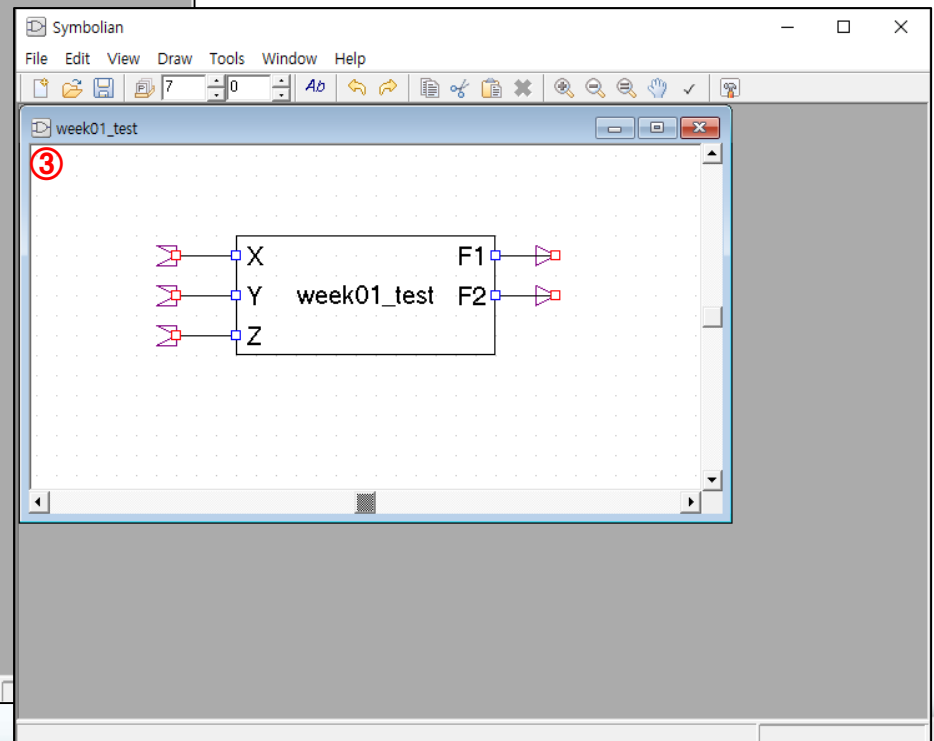
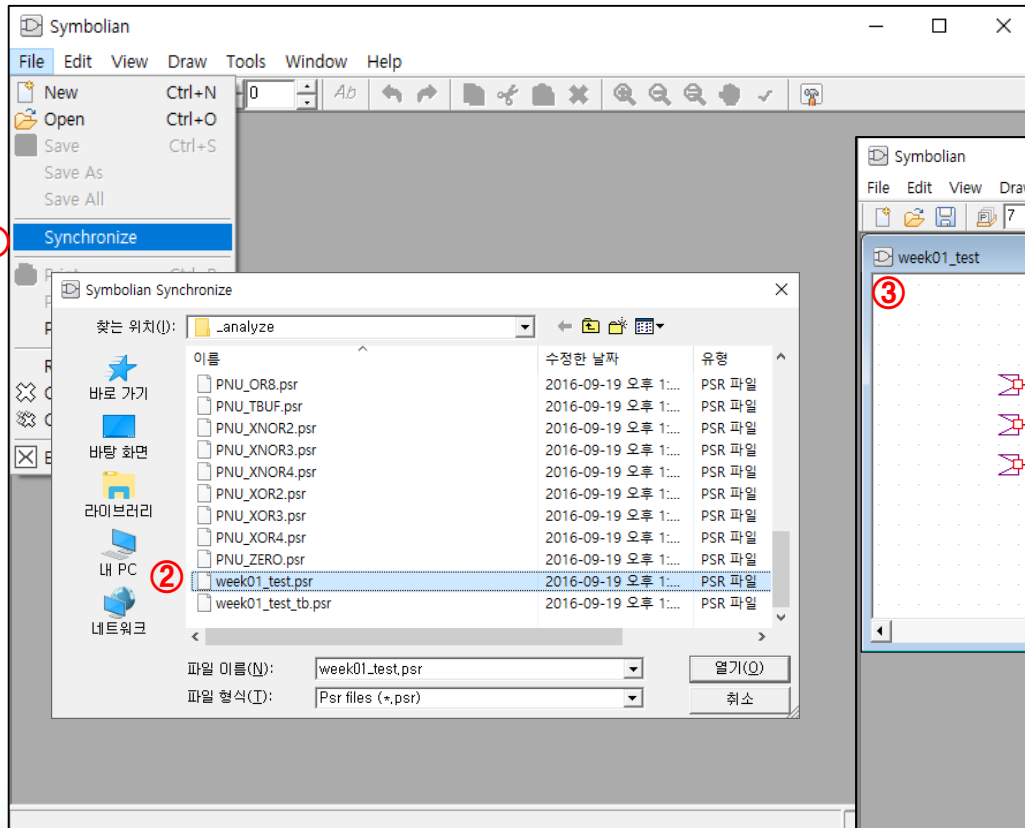
① [Analyze All]을 실행 (.v 파일을 추가한 상태에서)





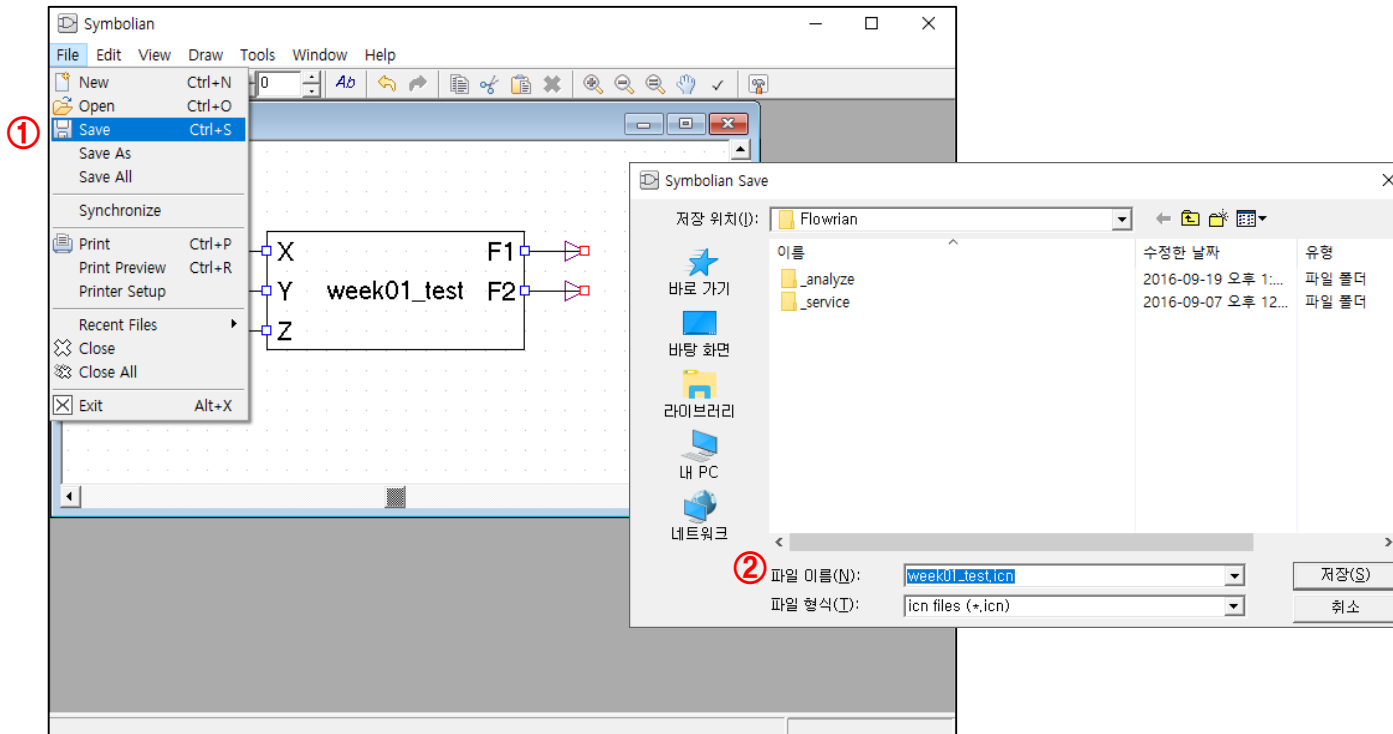
## Symbol 생성

- ① [File] -> [Synchronize]클릭
- ② Symbol로 만들 .psr파일 선택
- ③ Symbol 생성



## Symbol 저장

- ① [File] -> [Save]선택
- ② .icn파일 저장



## Symbol 사용

- ① Schman에 자동으로 .icn파일이 추가됨
  - ② 일반 게이트와 같은 방법으로 사용 가능
- ※참고 : include에 .v 파일을 추가해줘야 analyze 가능

