논리회로 설계 및 실험

- 2주차 : 기본적인 논리식 -

담당 교수: 권동현 교수
조교: 송수현 박사과정

컴퓨터보안연구실 : http://sites.google.com/csl-pnu

공지사항

1. 조편성 및 자리배치 고정
2. 과제(실습)는 수업전에 자료를 보고 미리 진행해도 됨. 또한, 강의시간 내에 완성 못할 시, 해당 주의 금요일까지 조교에게 제출

2주차 목표

목표

1. 문제에서 논리식을 유도해내고 논리회로로 구현
2. Encoder와 Decoder 개념이해 및 논리회로 구현

논리회로

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

논리회로 설계

1. 설계 방법 및 설계 툴이 다양함
2. 논리회로 설계 및 실험 과목에서는 Quartus 툴을 사용하여 회로를 설계함

Quartus 소개

- 하드웨어 설계 툴
- 논리회로의 설계와 시뮬레이션 기능을 가진 소프트웨어

Quartus 기능

Quartus

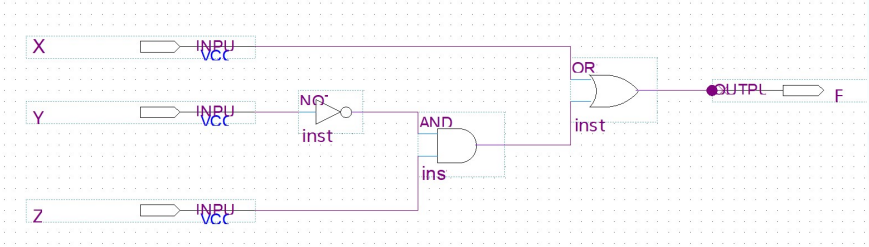
: 전체 설계 프로젝트 관리

- **디자인** : 심벌들을 사용해 회로를 작성
- **시뮬레이션** : Model sim을 이용한 입력파형 편집 및 파형 분석
- **Verilog 파일 변환** : .V 파일로 변환

논리식과 논리회로

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

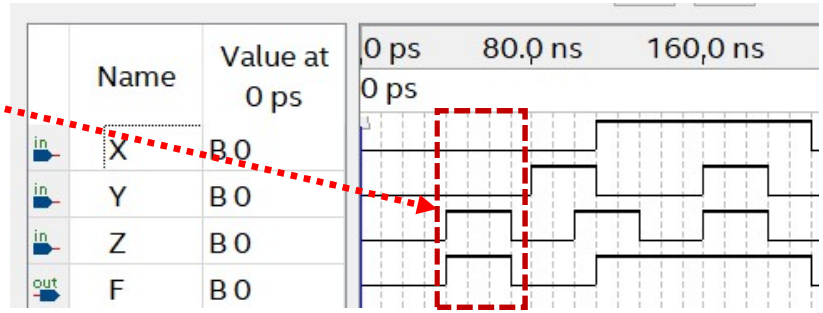
“ 합항은 OR 게이트 “
곱항은 AND 게이트
보수는 NOT 게이트



Quaturs로 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



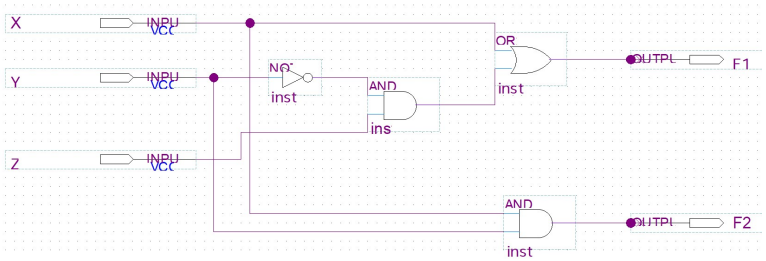
Quaturs 툴로 시뮬레이션한 $F(X,Y,Z) = X + \bar{Y}Z$ 의 파형

논리식과 논리회로

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

$$F_2(X,Y) = XY$$

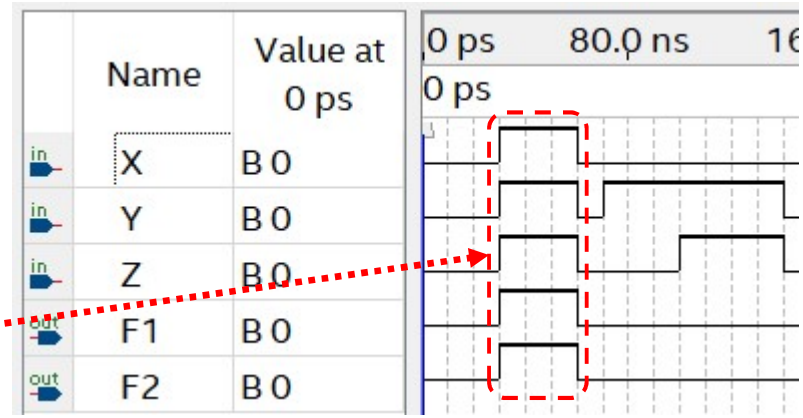
“ 두 개 이상의 논리식도 “
동시에 표현



Quartus로 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

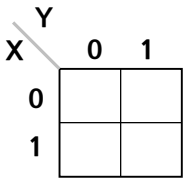
F_1, F_2 두 논리식의 진리표



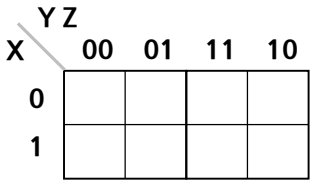
Quartus로 시뮬레이션한 F_1, F_2 두 논리식의파형

카르노맵(Karnaugh map, K-map)

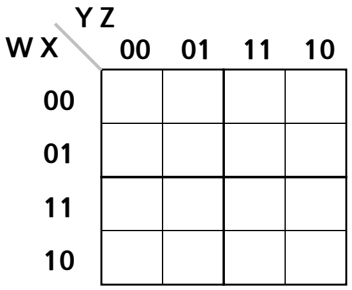
적은 개수의 변수에 한해 논리식의 함수를 유도해내는 방법 중 하나



2변수 카르노맵



3변수 카르노맵



4변수 카르노맵

카르노맵 응용

예시 문제)

1bit 입력 A와 B를 비교하여,
A>B이면 LED1, A=B이면 LED2가 켜지는 1bit 비교기 설계

예시 문제 접근

입력 : A, B
출력 : LED1, LED2, LED3

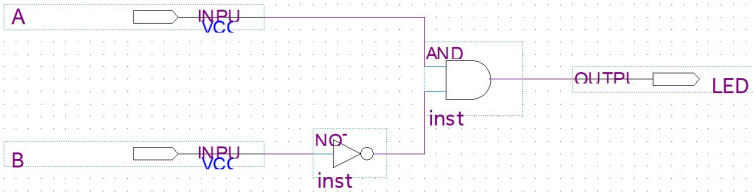
① A>B 인 경우는 A가 1이고 B는 0
② A=B 인 경우는 A, B가 0 또는 1

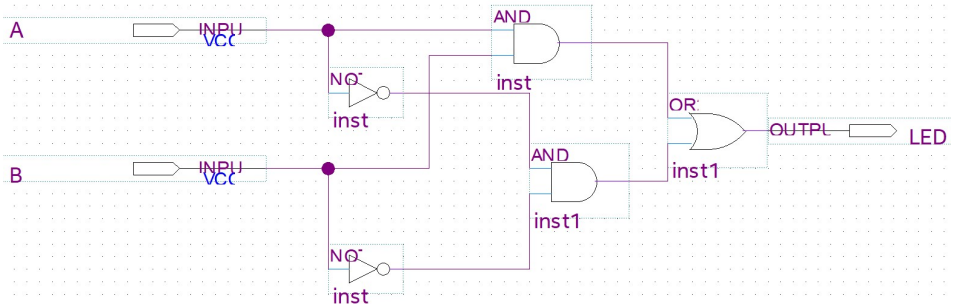
①

X \ Y	0	1
0		
1	1	

②

X \ Y	0	1
0	1	
1		1





Encoder와 Decoder

Encoder

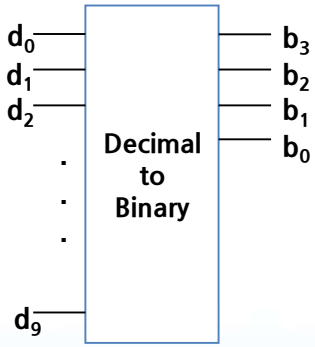
- 어떤 정보의 형태나 형식을 부호화(encoding)하여 다른 형태나 형식으로 변환하는 장치
- 처리속도 향상이나 데이터 압축 또는 데이터의 손실 방지를 위해서도 사용됨

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
- 이번 2주차 실험에서는 십진수 정보를 이진수 형태(BCD code)로 변환하는 변환기를 구현함



Decimal-to-Binary
변환기의 심벌

Encoder

Simple Encoder

- Simple Encoder는 one-hot code를 binary 정보로 변환함
- 이때 2^n개의 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

잘못된 회로 설계의 예

예시)
입력 : A, B
출력 : LED1

A버튼을 누르거나 B버튼을 누르면 LED1이 켜지는 회로

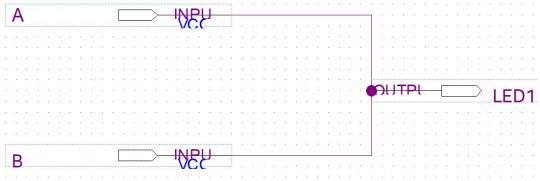


Fig1. 두 개의 입력이 하나의 출력에 연결

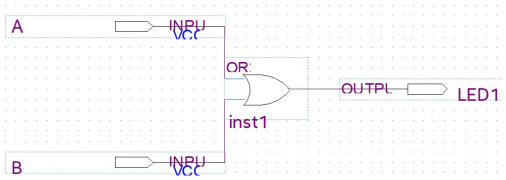


Fig2. 두 개의 입력을 논리 게이트로 연결

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

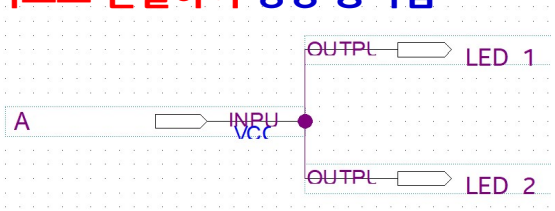


Fig3. 하나의 입력에 두 개 이상의 출력

Fig3과 같이 하나의 입력에 둘 이상의 출력은 정상 동작함

Logic Array 형식의 설계

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

Slide11과 같이 회로도가 복잡해질수록 상단의 설계 방식은 디버깅이 어려움
Logic Array와 유사한 하단의 설계 방식을 권함

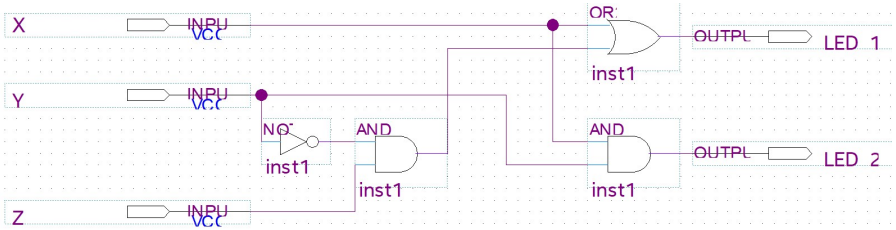


Fig1. 기본적인 설계 드로잉

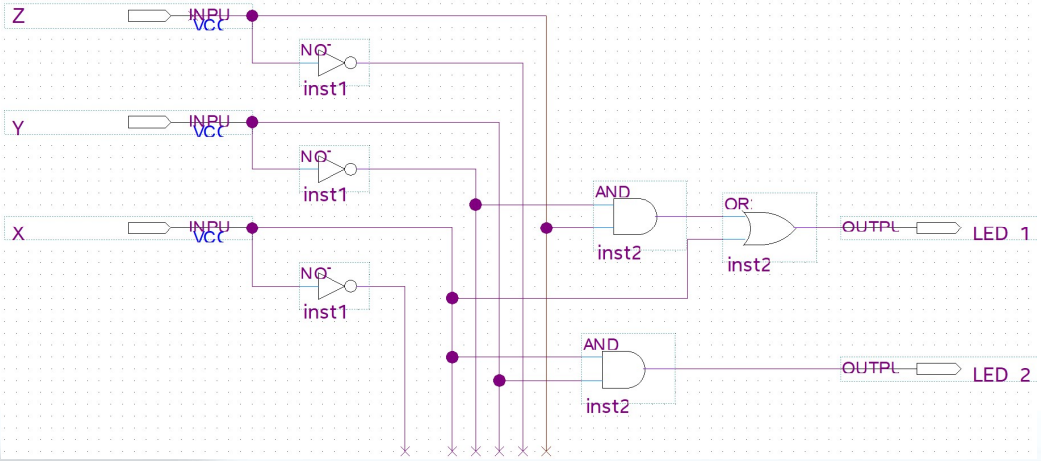
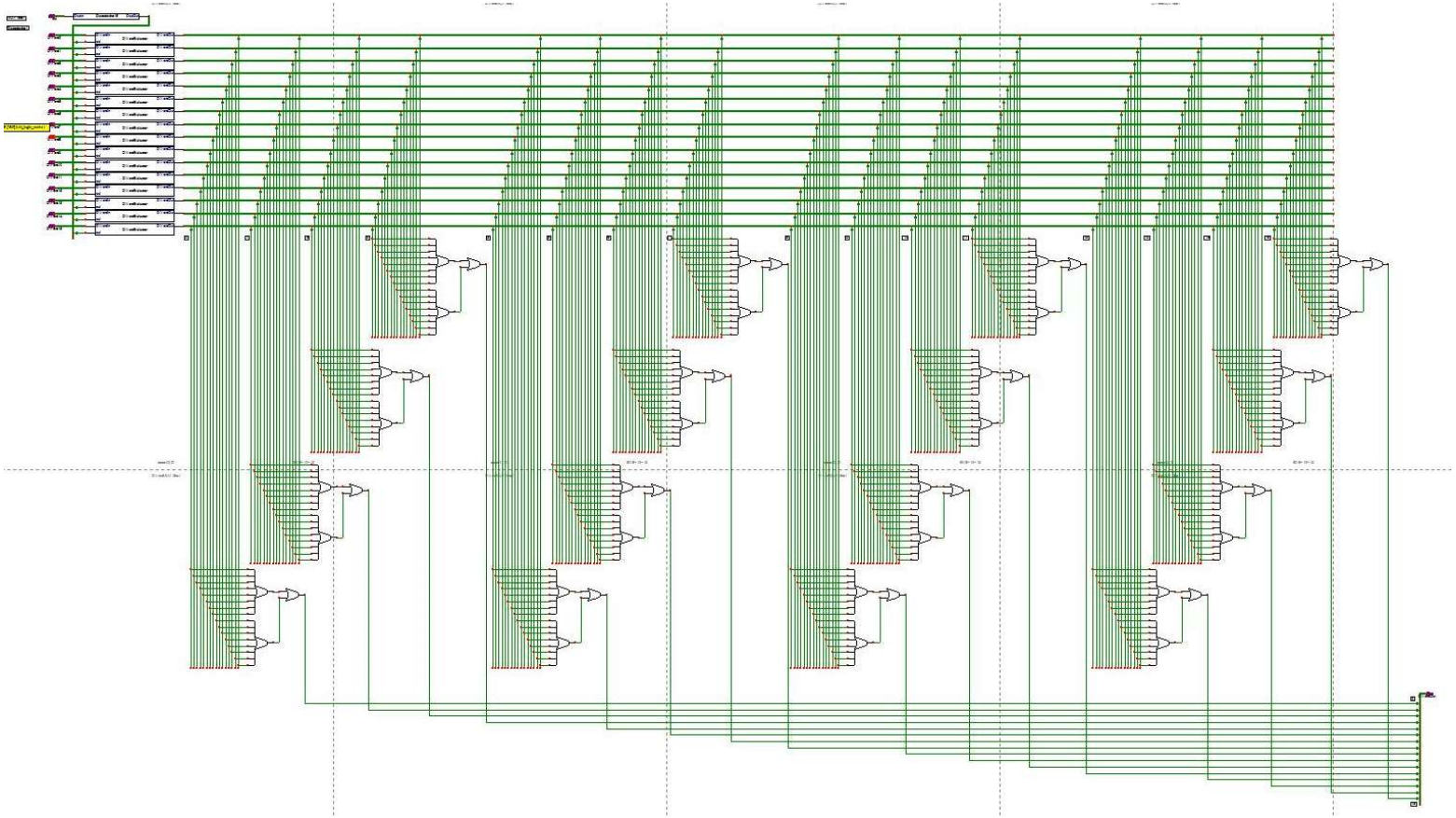


Fig2. 입력 포트의 배선을 순서대로 정리한 드로잉

참고

예시) 256bit 메모리의 일부



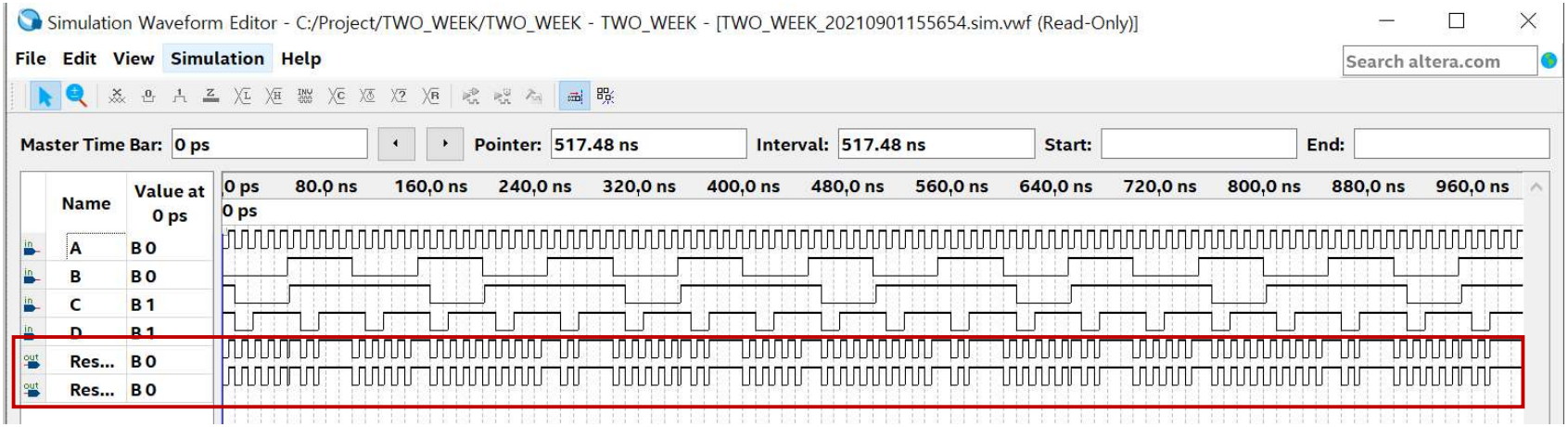
실습

- 논리식 및 항등식을 각각 구현 및 검증
- Decimal-to-Binary 회로 구현

실습 1

다음 논리식들이 항등식을 만족하는지 시뮬레이션으로 확인하시오

1. $(A + B)(A + CD) = A + BCD$



실습 2

아래 기능을 만족하는 Decimal-to-Binary 회로를 구현하시오

입력 : d_0, d_1, d_2 ~ d_{11}
출력 : b_3, b_2, b_1, b_0

입력 d_2 가 1이면 출력 {b_3, b_2, b_1, b_0} = 0010
입력 d_{11} 가 1이면 출력 {b_3, b_2, b_1, b_0} = 1011
(테스트벤치의 입력은 여러 bit 중 하나의 bit만 1로 넣을 것)

즉, Decimal에 해당하는 입력을 받으면 이를 Binary로 변환하는 회로