**실험 3. 디코더와 멀티플렉서**

2022. 04. 19

20210054 정하우

1. **개요**

Multiple-output 회로를 대표하는 디코더와 Multiple-input 회로를 대표하는 멀티플렉서의 기능을 이해하고 이를 사용해 회로를 구성한다.

본 실험에서는 3가지 회로를 다룬다. 여러 개의 active low 2 to 4 decoder를 통해 4 to 16decoder만들기, 4bit 소수판별기와 배수검출기, 마지막으로 5bit Majority function을 구현한다. 세부적인 학습 목표는 다음과 같다.

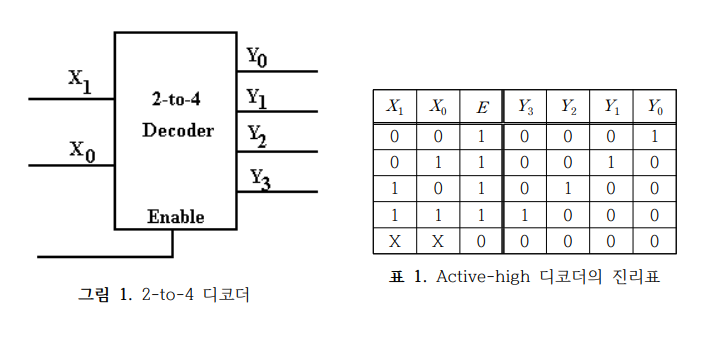
- Active-low 디코더 확장 이해 및 구현

- 특수 목적 디코더 구현

- 멀티플렉서의 데이터 선택 기능을 활용한 Majority function 구현

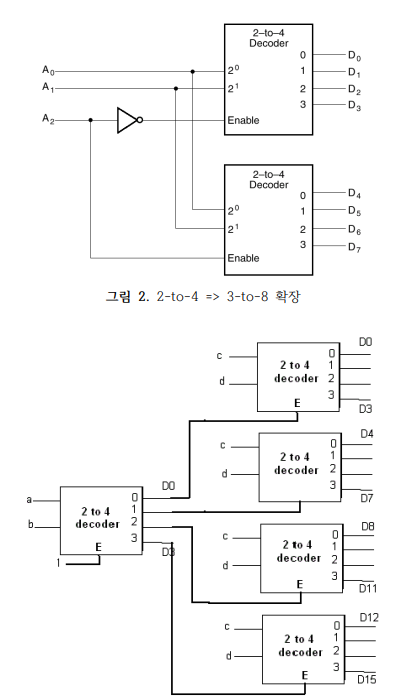
1. **이론적 배경**
2. 디코더 (Decoder / De-Multiplexer)

디코더는 n개의 이진 입력을 받아 최대 개의 고유 출력을 가지는 회로다. 출력값은 하나만 1이고 나머지는 다 0을 가진다. n개 의 이진 입력과 개의 서로 다른 출력을 가지는 경우, 각 출력이 곧 minterm이 기 때문에 minterm generator라고도 한다. 실제 디코더 소자에는 n개의 입력뿐 아니라 EN, 혹은 Enable 입력이 추가로 존재한다. EN은 말 그대로 디코더를 켜거 나 끄기 위해서 사용된다. 디코더의 입력과 출력 특성을 표현할 때는 n-to-과 k-of-라는 표현을 사용 한다. n-to-은 n개의 입력을 받아 개의 출력을 가진다는 입력과 출력의 관계 를 표현하고, k-of-은 개의 입력 중에서 k개의 입력이 동시에 참이 된다는 출 력의 특성을 나타낸다.

****

1. 디코더 확장

EN을 사용해 디코더를 켜거나 끌 수 있다는 점을 이용해 여러 개의 디코더를 연 결하여 더 많은 수의 입력을 처리할 수 있다. 이를 디코더 확장(Decoder expansion)이라고 한다. 예를 들어, 그림 2와 같이 2-to-4 디코더를 두 개 사용해 3-to-8 디코더를 구성 하거나, 그림 3과 같이 2-to-4 디코더를 다섯 개 사용해 4-to-16 디코더를 구성할 수 있다.



3) 특수 목적 디코더

- 소수 판별기

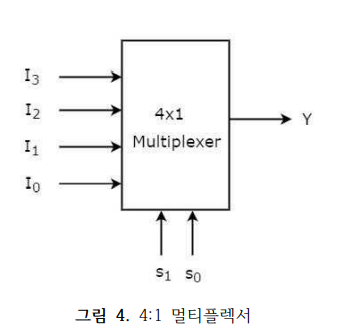
주어진 입력이 소수일 때 참을 출력하는 회로다.

- 배수 검출기

비슷한 원리로 배수일 때 참을 출력하는 회로이다.

1. 멀티플렉서 (Multiplexer; MUX)

멀티플렉서는 선택 신호에 따라 여러 입력 신호 중 하나를 골라 출력하는 회로다. 주로 개의 입력 신호를 ㅜ개의 선택 신호로 선택하고, 그 하나를 출력한다.



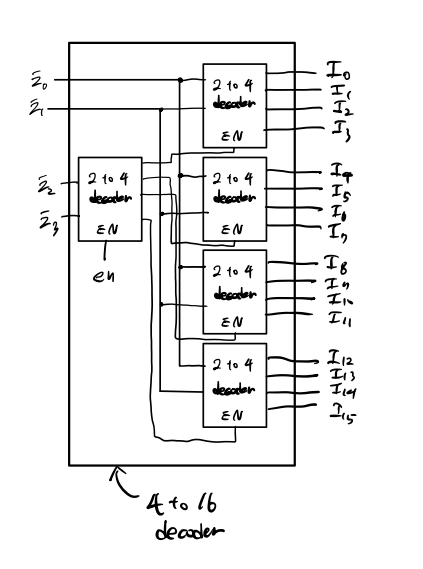
1. Majority / Minority function

홀수 개의 입력에 대해 다수 / 소수를 차지하는 입력을 나타내는 함수다. 예를 들어, 세 개의 입력이 1, 1, 0이라면 Majority function과 Minority function의 결과는 각각 1과 0이다.

1. 멀티플렉서를 사용한 함수 표현

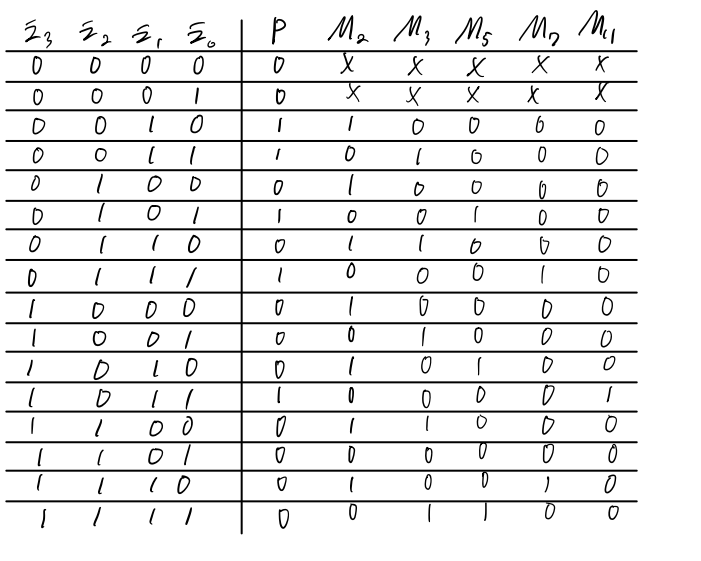
, ()

1. **실험 준비**
2. 2-to-4 Active-low enable, Active-low output, Active-high input 디코더로 4-to-16 Active-low enable, Active-low output, Active-high input 디코더를 구현한다.

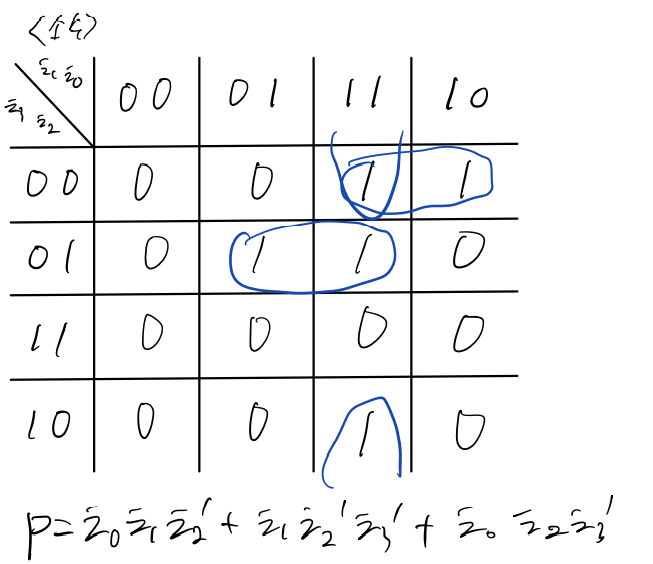


1. 4비트 소수 판별기와 배수 검출기(2, 3, 5, 7, 11)의 진리표를 구하고, 식을 단순화한다.

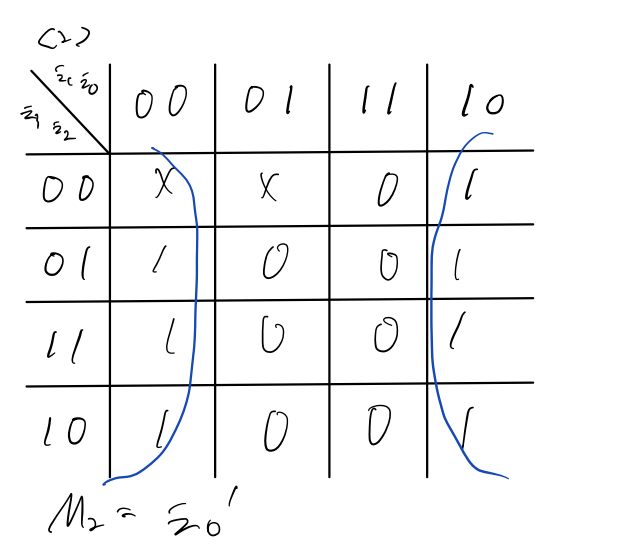
<truth table>



<소수>



<2의 배수>

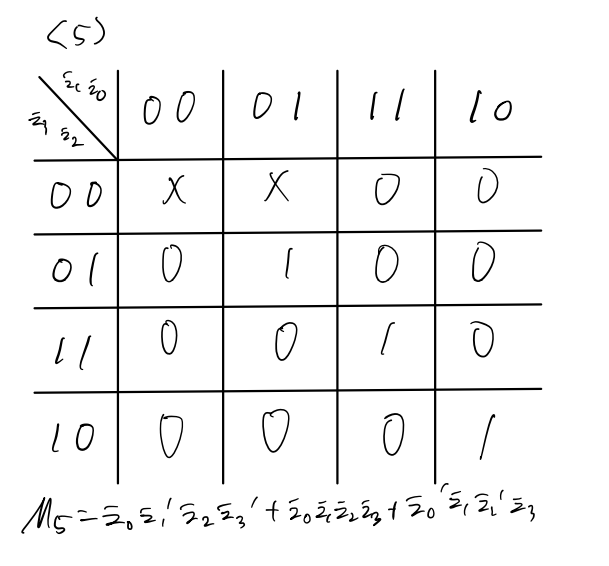


<3의 배수>

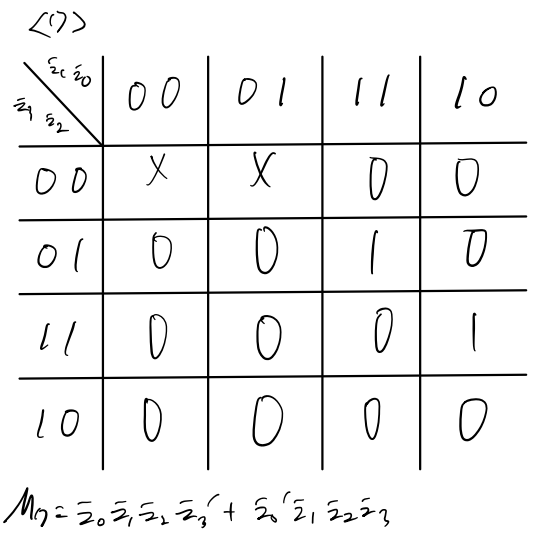
텍스트이(가) 표시된 사진

자동 생성된 설명

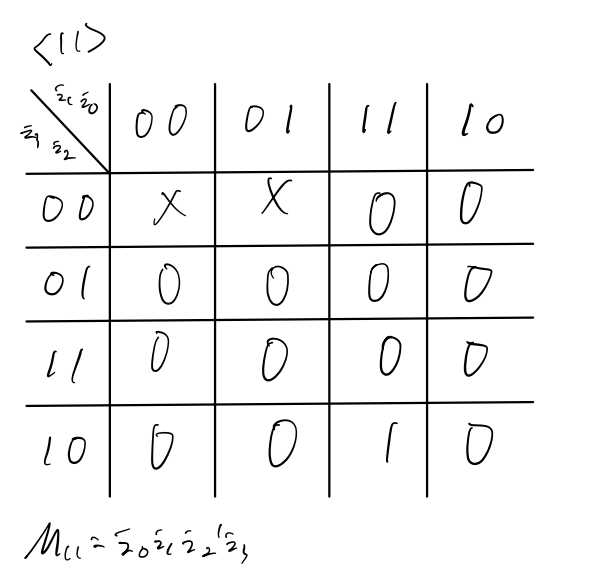
<5의 배수>



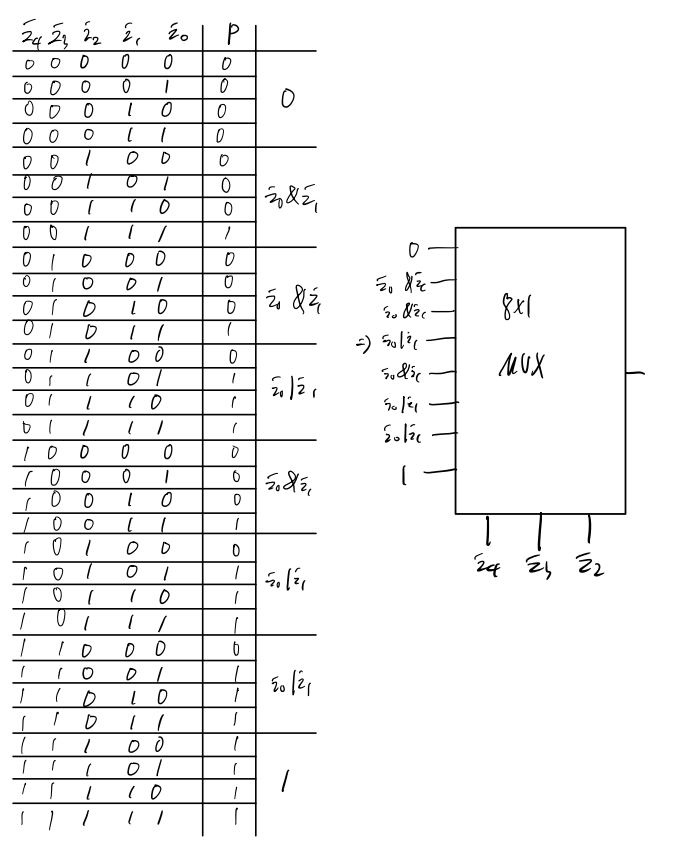
<7의 배수>



<11의 배수>

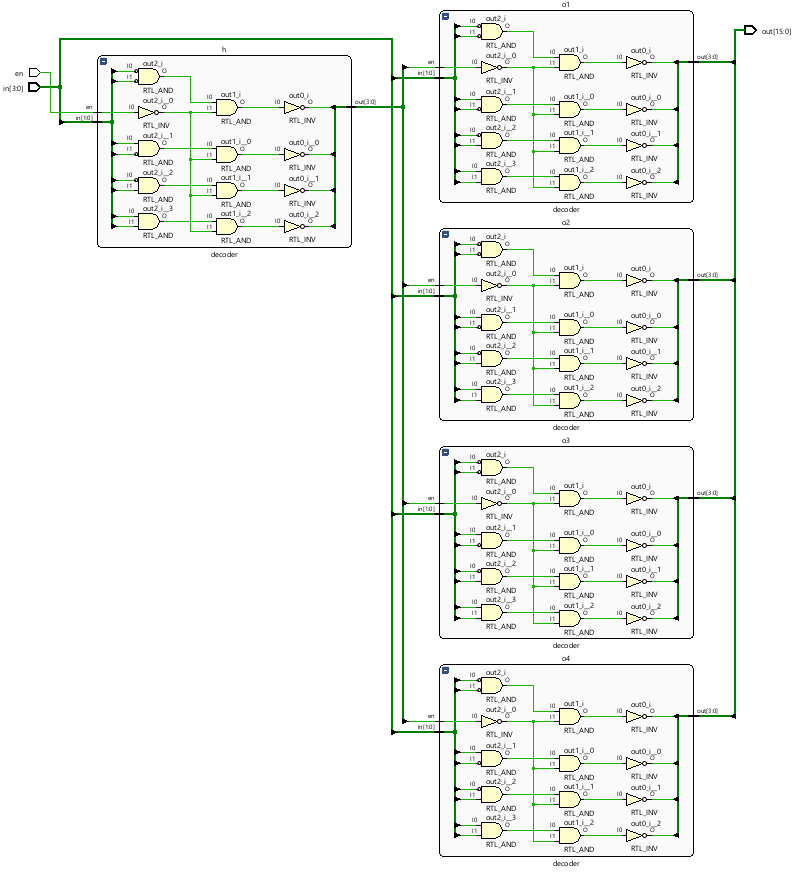


1. 5비트 Majority function의 진리표를 구하고, 식을 SOP꼴로 바꿔 8:1 멀티플렉서로 표현한다.

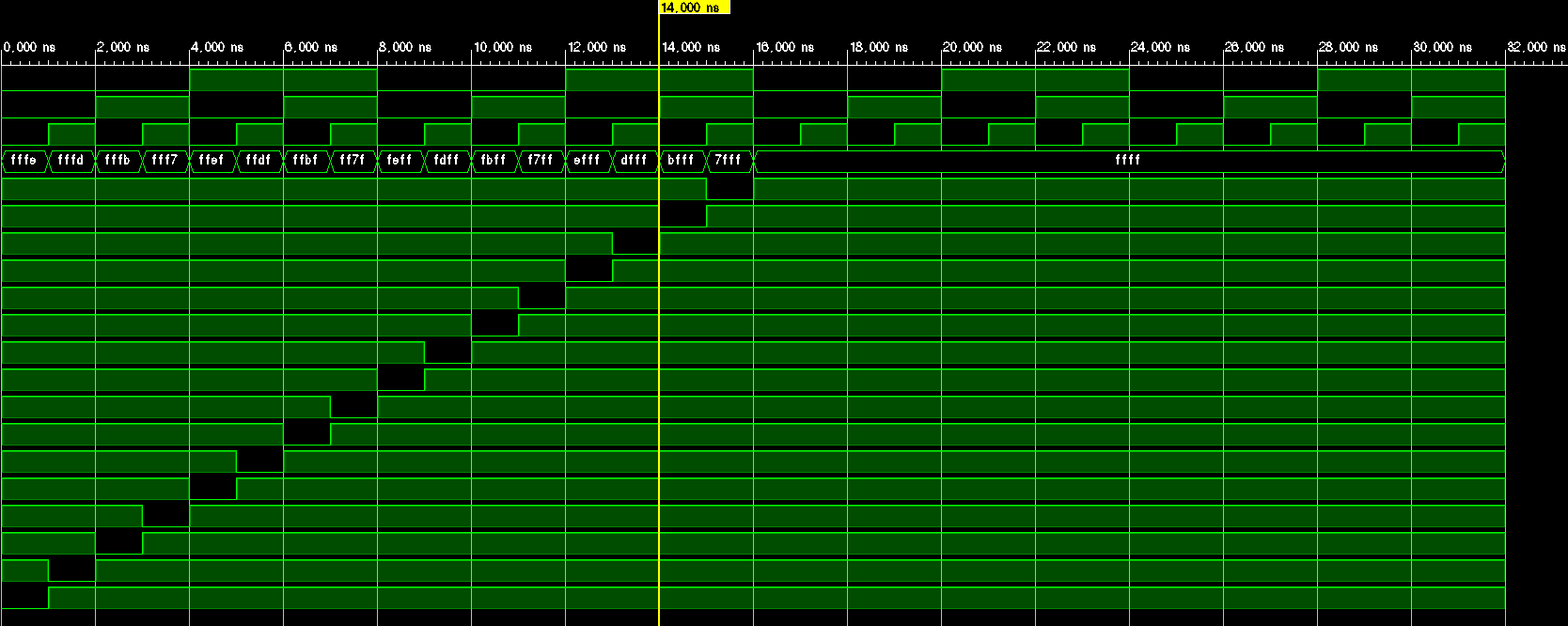


1. **결과**
2. lab3 1.v

회로도는 다음과 같다

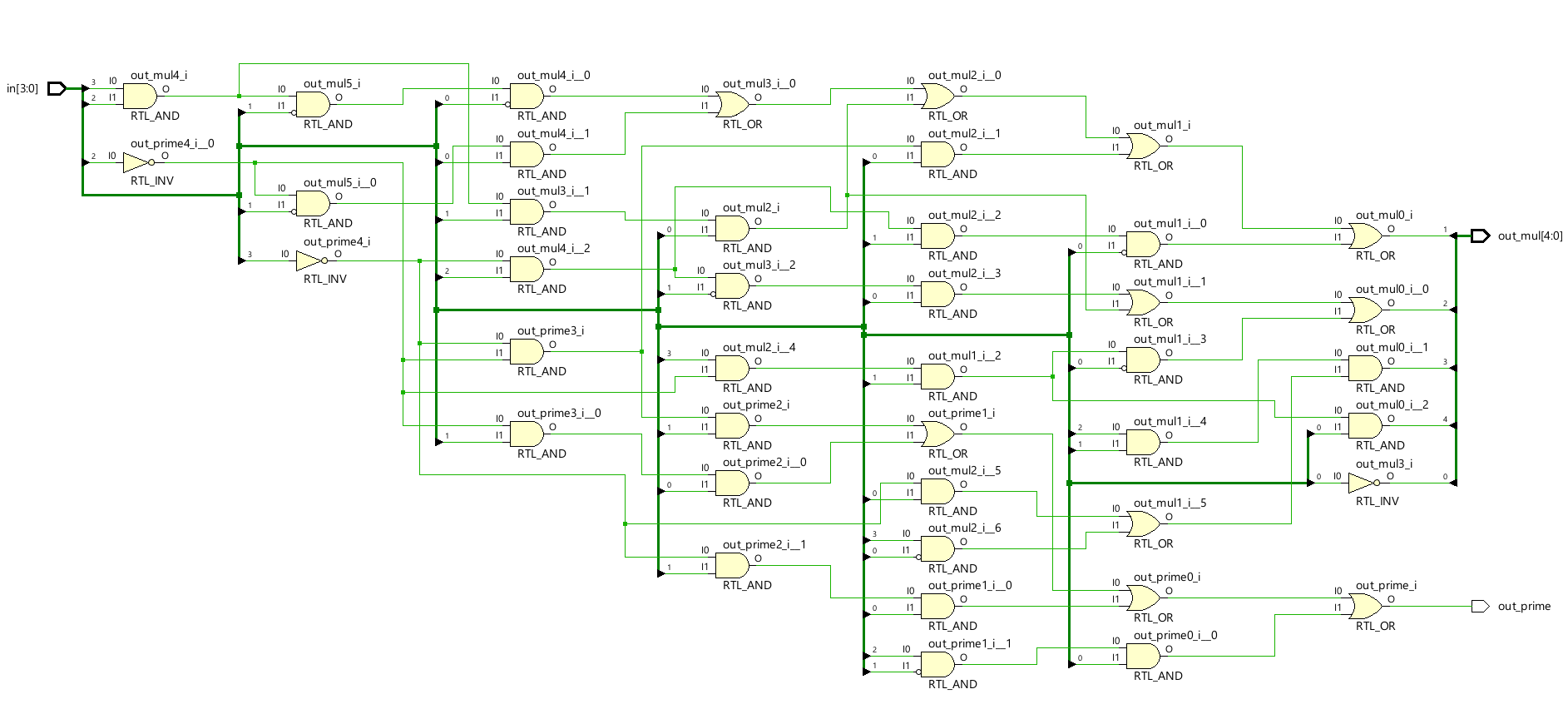


시뮬레이션 결과는 다음과 같다.

****

1. lab3\_2.v

회로는 다음과 같다

****

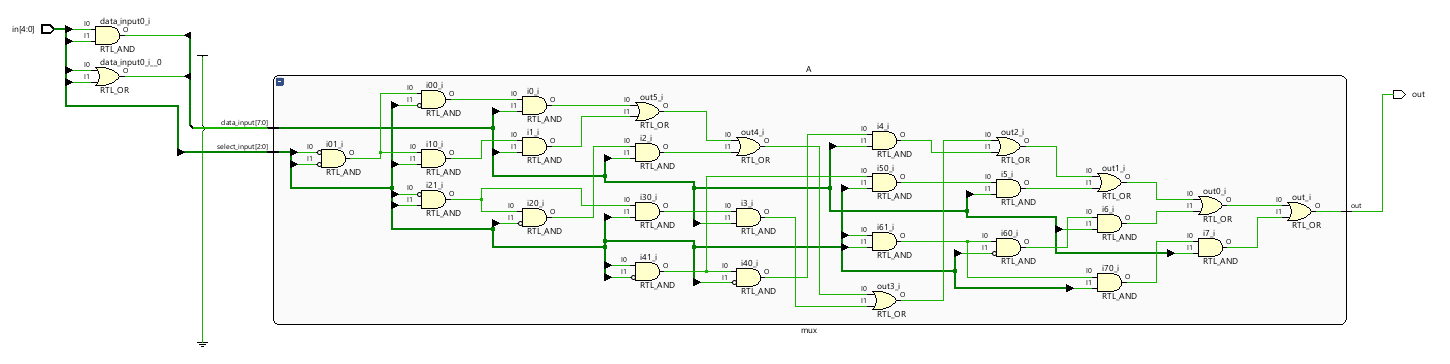
시뮬레이션 결과는 다음과 같다

**텍스트, 점수판이(가) 표시된 사진

자동 생성된 설명**

1. lab3\_3.v

회로는 다음과 같다

****

시뮬레이션 결과는 다음과 같다

**텍스트, 점수판이(가) 표시된 사진

자동 생성된 설명**

1. **논의 및 결론**

처음에는 복잡하다고 생각했었는데, truth table을 그리고 차근차근 해 보니 좋은 결과가 나왔던거 같다

처음에는 0은 모든 수의 배수라고 생각했는데, 그렇게 하지 말라고 해서 0과 1은 don’t care로 돌리고 하였다. 그런데 인터넷으로 찾아본 결과 0은 모든 정수의 배수라는 사실을 알게 되었다. 다음에는 0도 true라고 두고 실험을 진행해 보고 싶다.