

CSED273 Lab 2 보고서

2022.04.01

20210643 김현준

1. 개요

본 Lab 2에서는 불 대수식을 여러 방법으로 단순화해보고, 단순화 전 후를 비교하여 게이트 개수 감소 등의 효과가 있는지 실제로 확인한다. 특히 불 대수식을 단순화하는 Karnaugh-Map 방식의 알고리즘을 이해하고, 단순화 전 후의 논리 게이트와 와이어 수를 비교하여 효과를 확인하는 과정을 수행한다. 또한, 본 실험에서는 2-Bit Magnitude Comparator 회로의 불 대수식을 단순화하여 그 효과를 알아본다.

2. 이론적 배경

1) 불 대수식의 단순화

불 대수식의 복잡도는 해당 식을 구현하는데 필요한 와이어 및 논리 게이트의 개수로 평가될 수 있다. 복잡도를 낮추면 그 불 대수식에 들어가는 게이트와 와이어의 수를 줄일 수 있고, 소비 전력을 줄이며, 작동 속도를 높일 수 있다. 복잡도를 낮추기 위해서 불 대수식의 단순화 과정을 거쳐야 한다. 식을 직접 여러 연산 법칙을 활용하여 단순화하는 방법도 있지만, 보다 편리한 방법으로 Karnaugh-Map 방식과 Quine-McCluskey 알고리즘이 존재한다. 근본적으로는 하나의 변수를 소거하는 같은 방식을 사용하는데, 카노 맵 방식은 인간이 쉽게 다룰 수 있도록 지도에 표시하여 단순화하는 방식이고, 퀸 매클러스키 알고리즘은 이를 체계화한 알고리즘이다.

카노 맵 방식은 PI(Prime Implicant)를 찾고, 그 중에서 EPI(Essential Prime Implicant)를 찾아 포함되지 않는 부분은 PI로 보충하는 식으로 단순화 형태를 찾고, 매클러스키 알고리즘은 최대한 Literal들을 제거하고, min set of PI로 EPI를 찾는 것을 반복하여 이루어진다.

2) 2-Bit Magnitude Comparator

2-Bit Magnitude Comparator는 두 개의 2-Bit 수가 입력으로 주어졌을 때, 대소관계를 판별하여 출력하는 회로이다. 이때, 대소 관계 판별이 알맞은 경우 1을, 그렇지 않은 경우 0을 출력하게 된다. 예를 들어 A, B가 입력으로 주어졌을 때, A가 10, B가 01인 경우 $A > B$ 의 출력부에서는 1이, $A = B$ 의 출력부에서는 0이, $A < B$ 의 출력부에서는 0이 출력되게 된다.

3. 실험 준비

1) GT

먼저, $A > B$ 인 경우 1을, 그렇지 않은 경우에는 0을 입력하는 경우이다. 이 때, Karnaugh-Map을 그려보면 아래와 같다.

입력		A_0A_1			
		00	01	11	10
B_0B_1	00	0	1	1	1
	01	0	0	1	1
	11	0	0	0	0
	10	0	0	1	0

따라서 단순화하지 않은 불 대수식은 아래와 같다.

$$A_0'A_1B_0'B_1' + A_0A_1B_0'B_1' + A_0A_1'B_0'B_1' + A_0A_1B_0B_1 + A_0A_1'B_0B_1 + A_0A_1B_0B_1'$$

아래와 같이 단순화를 해 보면 결과적으로 아래의 불 대수 단순화 식을 얻을 수 있다.

입력		A_0A_1			
		00	01	11	10
B_0B_1	00	0	1	1	1
	01	0	0	1	1
	11	0	0	0	0
	10	0	0	1	0

단순화 식: $A_0B_0' + A_1B_0'B_1' + A_0A_1B_1'$

2) EQ

A=B일 경우 1을, 그렇지 않을 경우 0을 출력하는 경우이다. 카노 맵을 그리면 아래와 같다.

입력		A_0B_1			
		00	01	11	10
B_0B_1	00	1	0	0	0
	01	0	1	0	0
	11	0	0	1	0
	10	0	0	0	1

따라서 단순화하지 않은 불 대수식을 아래와 같다. 그런데, EQ의 경우 Literal을 소거할 수 있는 경우가 존재하지 않기 때문에, 이 불 대수식이 가장 단순한 형태로 나타낸 불 대수식이 될 것이다.

단순화 식: $A_0'A_1'B_0'B_1' + A_0'A_1B_0'B_1 + A_0A_1B_0B_1 + A_0A_1B_0B_1'$

3)LT

마지막으로 LT는 A<B일 경우 1을, 그렇지 않을 경우 0을 출력하는 회로가 될 것이다. 카노 맵을 표현하면 아래와 같다.

입력		A_0B_1			
		00	01	11	10
B_0B_1	00	0	0	0	0
	01	1	0	0	0
	11	1	1	0	1
	10	1	1	0	0

따라서 이를 볼 대수식으로 쓰면 아래와 같다.

$$A_0'A_1'B_0'B_1 + A_0'A_1'B_0B_1 + A_0'A_1B_0B_1' + A_0'A_1B_0B_1 + A_0A_1'B_0B_1' + A_0A_1B_0B_1$$

또한 카노 맵에서 단순화를 하면, 아래와 같이 최종 단순화 식을 얻을 수 있다.

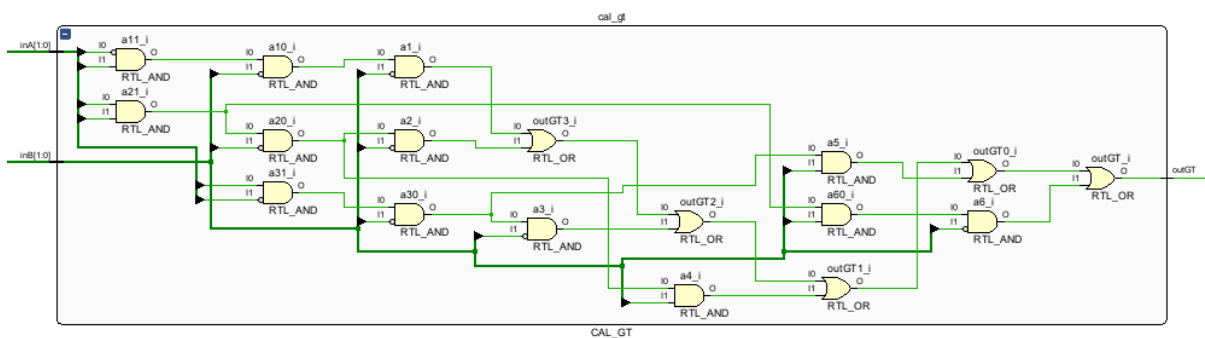
입력		A_0A_1			
		00	01	11	10
B_0B_1	00	0	0	0	0
	01	1	0	0	0
	11	1	1	0	1
	10	1	1	0	0

단순화 식: $A_0'B_0 + A_0'A_1B_1 + A_1'B_0B_1$

4. 결과

1) GT

우선 단순화 이전인, $A_0'A_1B_0'B_1' + A_0A_1B_0'B_1' + A_0A_1'B_0'B_1' + A_0A_1B_0'B_1 + A_0A_1'B_0'B_1 + A_0A_1B_0B_1'$ 식을 구현한 결과는 아래와 같았고, 와이어는 22개, 논리 게이트는 18개로 나타나는 것을 확인할 수 있었다.

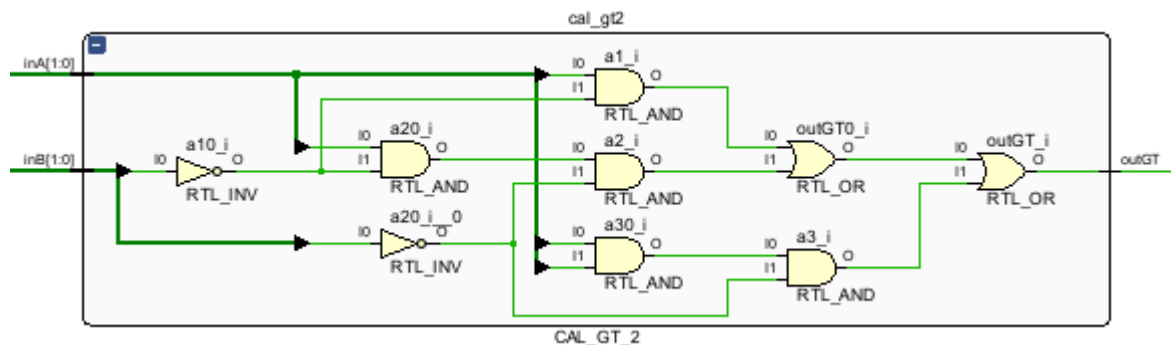


▼ cal_gt (CAL_GT)

> Nets (22)

> Leaf Cells (18)

다음으로 단순화 이후인 $A_0B_0' + A_1B_0'B_1' + A_0A_1B_1'$ 식을 구현한 결과는 아래와 같았고, 와이어는 13개, 게이트는 9개로 나타나는 것을 확인할 수 있었다.

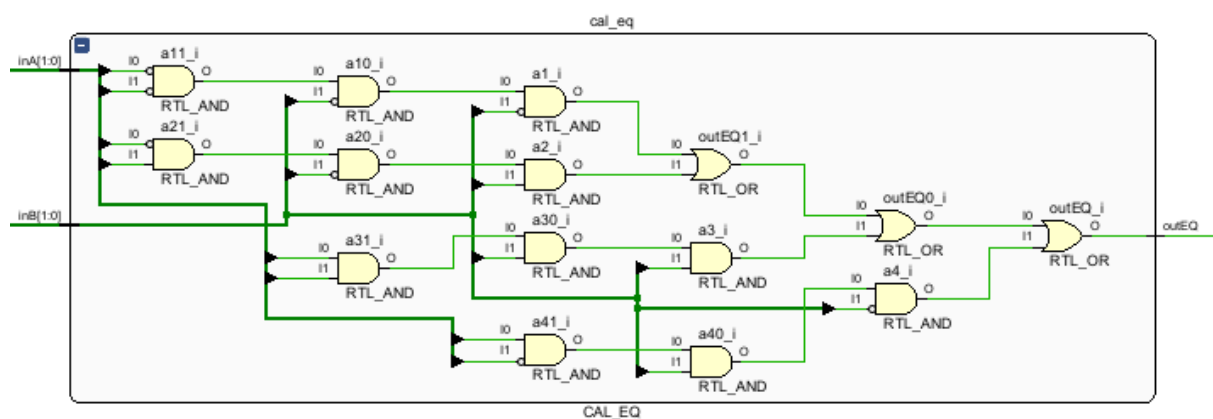


▼ cal_gt2 (CAL_GT_2)
 > Nets (13)
 > Leaf Cells (9)

단순화 진행 이후 와이어와 게이트 수 모두 약 절반 정도로 줄어든 것을 확인할 수 있었고, 이로써 단순화의 효과를 확인할 수 있었다. 게이트와 와이어 수가 줄어들고 함께 전류 소비량이나 소요 시간도 줄어들어 회로의 효율성이 증대될 것으로 예측된다.

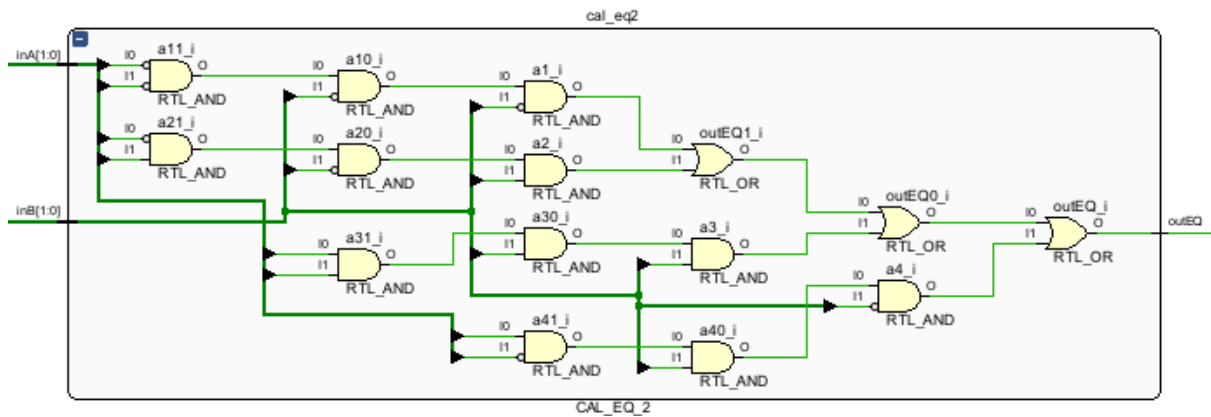
2) EQ

식 $A_0'A_1B_1'B_1' + A_0'A_1B_0'B_1 + A_0A_1B_0B_1 + A_0A_1B_0B_1'$ 를 구현한 결과 아래와 같이 구현되는 것을 확인할 수 있었고, 이때 와이어 수는 19개, 게이트 수는 15개임을 확인할 수 있었다.



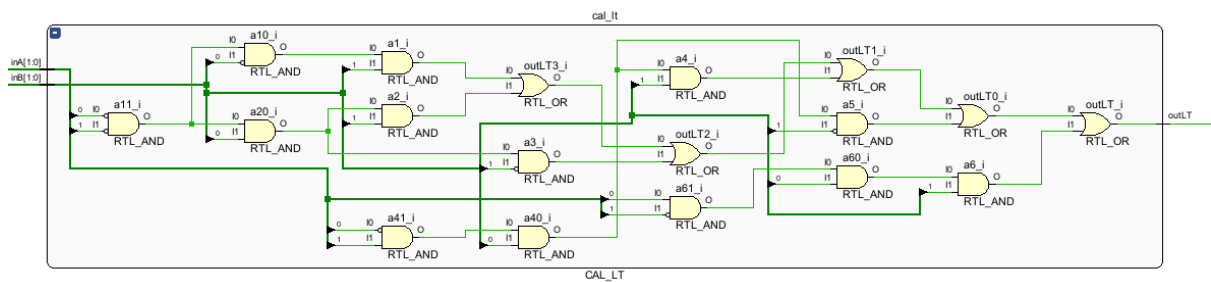
▼ cal_eq (CAL_EQ)
 > Nets (19)
 > Leaf Cells (15)

$A_0'A_1B_1'B_1' + A_0'A_1B_0'B_1 + A_0A_1B_0B_1 + A_0A_1B_0B_1'$ 식의 경우 더 단순화하는 것은 불가능하기 때문에, 식을 더욱 단순화하여 단순화 효과를 살펴보는 것은 할 수 없었다.



3) LT

우선 단순화 이전의 식인 $A_0'A_1'B_0'B_1 + A_0'A_1'B_0B_1 + A_0'A_1'B_0B_1' + A_0'A_1'B_0B_1 + A_0'A_1'B_0B_1' + A_0'A_1'B_0B_1$ 를 구현한 결과는 아래와 같았다. 이때 와이어 개수는 22개, 게이트 개수는 18개로 나타나는 것을 확인할 수 있었다.

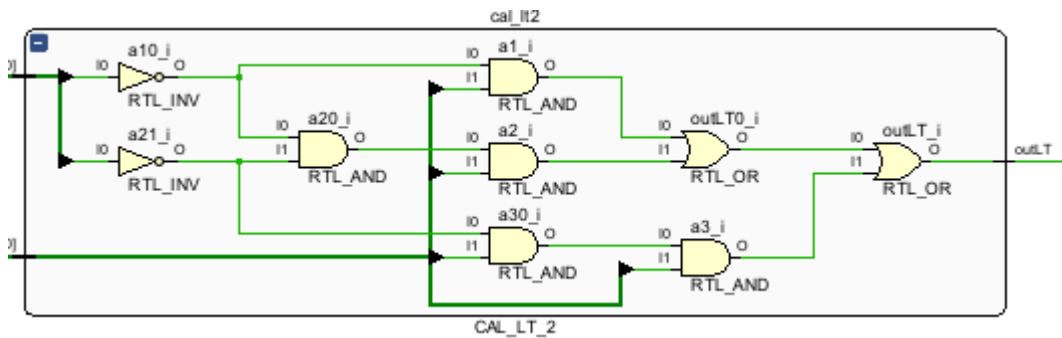





▼ cal_lt (CAL_LT)

> Nets (22)

> Leaf Cells (18)

다음으로 단순화한 형태의 식인 $A_0'B_0 + A_0'A_1'B_1 + A_1'B_0B_1$ 를 구현해본 결과 아래와 같이 나타났고, 와이어 개수는 13개, 게이트 개수는 9개로 단순화 이전보다 줄어든 것을 확인할 수 있었다.



▼  cal_lt2 (CAL_LT_2)
 >  Nets (13)
 >  Leaf Cells (9)

LT의 경우에도 GT와 마찬가지로 단순화 진행 이후 와이어와 게이트 수 모두 약 절반 정도로 줄어든 것을 확인할 수 있었고, 이로써 단순화의 효과를 확인할 수 있었다. 게이트와 와이어 수가 줄어들고 함께 전류 소비량이나 소요 시간도 줄어들어 회로의 효율성이 증대될 것으로 예측된다.

5. 논의

처음에 실험을 계획한 대로 결과를 잘 얻을 수 있었다. GT와 LT의 경우 와이어 수는 22개에서 13개로, 게이트 수는 18개에서 9개로 줄어드는 것을 Vivado 환경에서 확인할 수 없었다. EQ의 경우 더 단순한 형태로 단순화할 수 없으므로, 게이트 및 와이어 개수가 그대로인 것으로 파악되었다. 또한, 이번 Lab2에서는 assign keyword와 세 개의 연산자가 허용됨에 따라, 기본적으로는 Gate-Level Modeling을 하지만 Not 게이트 대신 ~ 연산자를 사용하여 코드를 작성하였다. 이번 Lab2를 통해 카노 맵을 이용한 불 대수식 단순화의 효과를 느낄 수 있었고, 불 대수식 단순화의 개념과 카노 맵 알고리즘의 개념을 확고히 할 수 있었다. 또한 Vivado 환경과 프로그래밍에 더 익숙해질 수 있었다.