5주차 결과보고서

전공: 컴퓨터공학과 학년: 3학년 학번: 20211547 이름: 신지원

**1.**

드모르간 제 1법칙과 2법칙으로 코드를 작성하고 그에 따른 simulation 결과를 통해 드모르간 법칙이 성립됨을 알아본다. 또한 실습에서 진행하였던 1Bit 비교기도 simulation 을 통해 input 에 따라 어떻게 결과가 나타나는지 알아본다.

**2.**

2-1. De-Morgan 제 1법칙 “ “

De-Morgan 의 제 1법칙은 논리합을 논리곱으로 변형하여 나타낸다. 다시 말해 A+B의 보수 취한 것이 A의 보수와 B의 보수와 곱한 것과 같다는 것이다. 따라서 가 과 같다는 의미를 지닌다. 코드를 작성할 때 위 식에서 A 는 aa로, B 는 bb 로 하여 input 2개를 선언하였고, 에 대한 출력값을 c 로, 에 대한 출력값을 d 로 선언하였다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

위 시뮬레이션을 보았을 때, c 와 d 의 신호가 모든 순간에 같음을 알 수 있다. 테스트벤치를 작성할 때 입력 신호의 경우의 수를 모두 보기 위하여 aa의 주기는 20에서 시작, bb 는 50에서 시작하였지만 한 순간에도 겹치지 않는 결과가 나타났기 때문에 임이 선언되었다.

그렇다면 처럼 input 자체에 not 일 경우에는 결과가 어떻게 될까? 위에서는 A 와 B 가 notA, notB 가 되었기 때문에 input 값이 notA, notB 라면 A, B 로 출력될 것이다.

2-2. De-Morgan 제 2법칙 “ “

De-Morgan 의 제 2법칙은 논리곱을 논리합으로 변형하여 나타낸다. 다시 말해 A•B의 보수 취한 것이 A의 보수와 B의 보수와 합한 것과 같다는 것이다. 따라서 가 와 같다는 의미를 지닌다. 코드를 작성할 때 위 식에서 A 는 aa로, B 는 bb 로 하여 input 2개를 선언하였고, 에 대한 출력값을 c 로, 에 대한 출력값을 d 로 선언하였다.

스크린샷이(가) 표시된 사진

자동 생성된 설명

위 시뮬레이션을 보았을 때, c 와 d 의 신호가 모든 순간에 같음을 알 수 있다. 테스트벤치를 작성할 때 입력 신호의 경우의 수를 모두 보기 위하여 aa의 주기는 20에서 시작, bb 는 50에서 시작하였지만 한 순간에도 겹치지 않는 결과가 나타났기 때문에 임이 선언되었다.

그렇다면 처럼 input 자체에 not 일 경우에는 결과가 어떻게 될까? 위에서는 A 와 B 가 notA, notB 가 되었기 때문에 input 값이 notA, notB 라면 A, B 로 출력될 것이다.

**3.**

3-1. (A'+B')\*C' = ((A\*B)+C)'

A의 보수와 B의 보수의 합집합과 C의 보수를 곱집합한 것과 보수와 부호를 반대로 한 결과는 같다는 것을 증명하고자 한다.

스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

.위는 (A'+B')\*C' 의 simulation 결과다. A’+B’ 는 A와 B 가 둘 다 1인 경우에만 0을, 나머지 경우에는 다 1을 출력할 것이다. (A'+B')\*C' 의 결과는 A와 B가 둘 다 1인 경우를 제외하고 C가 0일 때 1을 출력할 것이다.

스크린샷, 텍스트, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

위는 ((A\*B)+C)' 의 simulation 결과다. A\*B 는 A와 B 가 둘 다 1인 경우에만 1을, 나머지 경우에는 모두 0을 출력할 것이다. ((A\*B)+C)' 의 결과는 A와 B가 둘 다 1인 경우를 제외하고 C가 0일 때만 1을 출력할 것이다. 따라서 위의 결과와 1이 되는 경우가 동일하다. 따라서 (A'+B')\*C' = ((A\*B)+C)' 은 성립한다.

3-2. (A'\*B')+C' = ((A+B)\*C)'

A의 보수와 B의 보수의 곱집합과 C의 보수를 합집합한 것과 보수와 부호를 반대로 한 결과는 같다는 것을 증명하고자 한다.

스크린샷, 텍스트, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

.위는 (A'\*B')+C' 의 simulation 결과다. A' \* B'는 A와 B 가 둘 다 0인 경우에만 1을, 나머지 경우에는 다 0을 출력할 것이다. (A'\*B')+C' 의 결과는 A와 B가 둘 다 0인 경우를 제외하고 C가 1일 때만 0을 출력할 것이다.

스크린샷, 디스플레이, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

위는 ((A+B)\*C)' 의 simulation 결과다. A+B 는 A와 B 가 둘 다 0인 경우에만 0을, 나머지 경우에는 모두 1을 출력할 것이다. ((A+B)\*C)' 의 결과는 A와 B가 둘 다 0인 경우를 제외하고 C가 1일 때만 0을 출력할 것이다. 따라서 위의 결과와0이 되는 경우가 동일하다. 따라서 (A'\*B')+C' = ((A+B)\*C)' 은 성립한다.

**4.**

1Bit 비교기는 두 개의 input 을 비교한 값을 출력하는 것이다. 총 4개의 비교를 한다. 두 개의 출력이 같은가, 다른가, 전자가 더 큰가, 후자가 더 큰가 에 대한 비교를 수식으로 나타내자면, A=B, A ≠ B, A>B, A<B 로 나타낼 수 있다. 하지만 Verilog 에서 이러한 비교를 하기 위해선 bit 라는 점을 활용하여 코드를 작성해야 한다.

i ) A=B

a와 b가 같다면, 1과 1또는 0과 0 일 것이다. 이는 2개의 input 이기 때문에 1이 홀수개가 아니라면 두 input 은 같을 것이다. XOR gate 의 결과에 NOT 을 취하여 1이 하나라도 있으면 XOR gate 에 의하여 1, 그리고 NOT 에 의하여 0 을 취할 것이다. 따라서 출력값 c 는 ~(a^b) 의 값으로 선언되었다.

ii ) A ≠ B

i 에서 두 input 이 같은 값인가를 확인하기 위하여 XOR gate 결과에 NOT 을 붙였기 때문에, 다른 값인 지 확인하기 위해선 i 에서 도출한 식에 NOT 을 붙이면 될 것이다. 따라서 이는 XOR gate 의 값과 같다. 출력값 d 는 (a^b) 의 값으로 선언되었다.

iii ) A > B

A 가 더 큰 경우의 수는 (1,0) 뿐이다. 따라서 input B에 보수를 취한 다음 두 input 이 모두 1인지를 확인한다면 (1,0) 의 상황인지 확인할 수 있을 것이다. 따라서 출력값 e 는 a & ~b 의 값으로 선언되었다.

iiii ) A < B

iii 와 반대로 A 의 값에 보수를 취하여 AND gate 로 출력하여 확인할 수 있다. 따라서 출력값 f 는 ~a & b 로 선언되었다.

스크린샷, 텍스트, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

작성한 코드의 simulation 은 위와 같다. 이를 바탕으로 아래 진리표를 작성하였다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input aa | Input bb | Output c | Output d | Output e | Output f |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

진리표에서 확인할 수 있듯이 c 는 XOR gate의 출력값 에 NOT gate를 출력한 값, d 는 XOR gate 의 출력값, e 는 notB 와 A 의 AND gate 출력값, f는 notA 와 B 의 AND gate 출력값으로 말할 수 있다.

**5.**

5주차의 실습을 통하여 드모르간을 활용하여 더 많은 논리식이 같을 수 있음을 직접 코드를 짜 시뮬레이션을 통해 확인할 수 있었다. 2~4번 까지의 실험으로 드모르간과 1Bit 의 시뮬레이션과 진리표를 직접 생각하고 확인하며 드모르간에 대해서 더 잘 알 수 있게 되었다. 다만, 시뮬레이션 화면에서 모든 경우의 수를 보고자 하였지만 주기를 조절하지 못하여 00,01,10,11의 경우의 수를 모두 볼 수 없는 실습이 있었다. 다음 실험부터는 이를 주의하여 모든 경우의 수를 한 눈에 볼 수 있도록 하겠다.

**6.**

부울대수의 법칙 중에서 흡수 법칙에 대하여 알아보았다. 한 쪽이 다른 쪽에 흡수된다는 의미를 가지고 있다. 구체적으로는 합집합이나 교집합의 연산을 할 때, 양쪽에 있는 두 변 중에 하나만 남겨지는 현상을 의미한다. 수식으로 나타낸다면 A + A \*B = A, A \* A + B = A 로 나타난다. 이처럼 A + A'B은 = A(1+B) + A'B, = A + AB + A'B, = A + B(A+A'), = A + B 이렇게 4가지로 표현 가능하다.