5주차 결과보고서

전공 : 컴퓨터공학과 학년 : 2학년 학번 : 20201597 이름 : 신동준

1. **실험 목적**

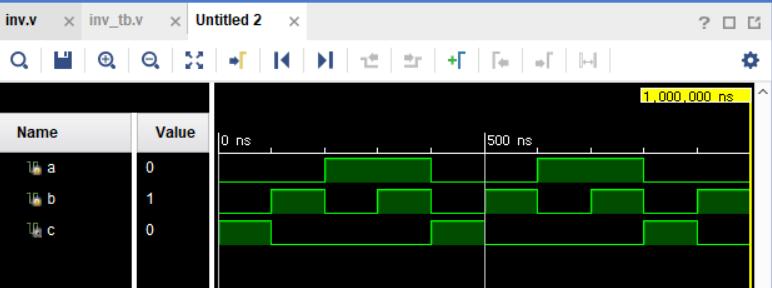
드 모르간 정리에 대해서 이해하고, 불리언 함수 동작을 확인하기 위해 1비트 비교기를 만들어본다. Verilog로 각각을 코딩하고 simultation을 통해서 결과를 확인하는 방법을 이해한다.

1. **De-Morgan 의 제 1,2 법칙의 simulation 결과 및 과정에 대해서 설명하시오.**
2. 1법칙

A :

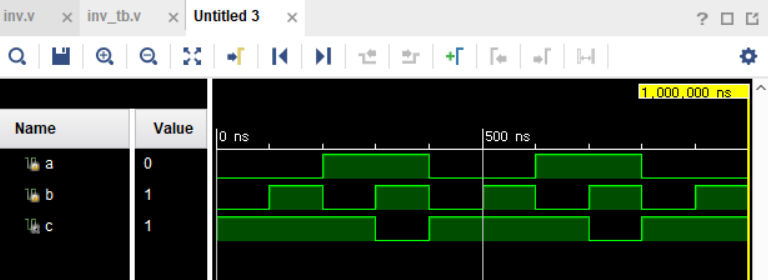


B:



1. 2법칙

A :



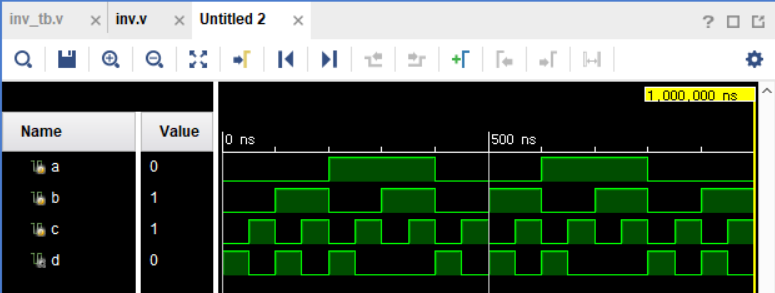
B :



각각을 비교해본 결과 NOR인 경우(1법칙), NAND인 경우(2법칙) 모두 각각 simulation이 동일한 wave form인 것을 알 수 있었다.

1. **(A'+B')\*C' = ((A\*B)+C)' 의 simulation 결과 및 과정에 대해서 설명하시오.**

(1-a) (~a | ~b) & ~c;

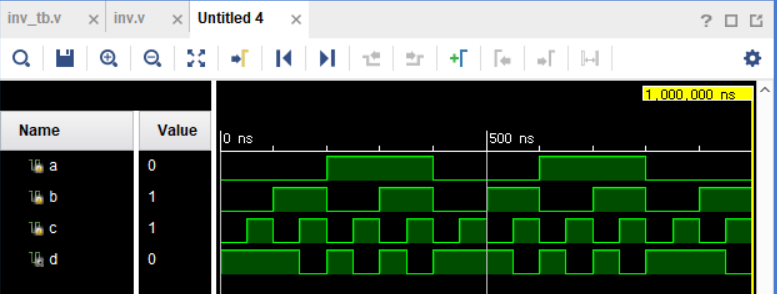


(1-b) ~((a & b) | c);

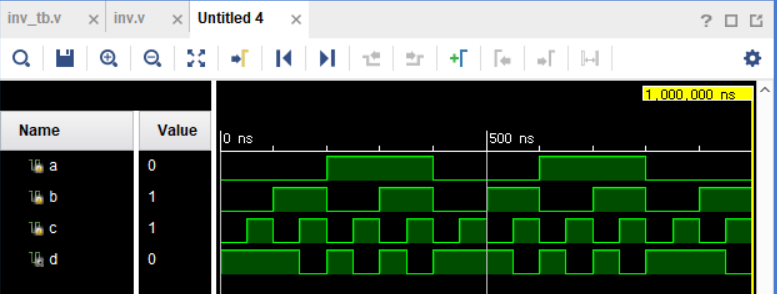
텍스트, 실내, 스크린샷이(가) 표시된 사진

자동 생성된 설명

(2-a) (~a & ~b) | ~c;



(2-b) ~((a | b) & c);



1의 경우 2의 경우에서 simulation을 돌린 wave form을 확인해본 결과 다음과 같은 결과를 얻을 수 있었다. 드 모르간 법칙을 기반으로 예상한 결과 처럼 각각 a,b의 경우가 동일한 wave form으로 나왔으며 NOT이 괄호 안으로 들어갈 때 and는 or로 or는 and로 바뀐 효과를 내는 것을 알 수 있었다.

1. **1Bit 비교기의 simulation 결과 및 과정에 대해서 설명하시오.**



시뮬레이션 결과는 다음과 같다. a,b를 각각 입력받고, 두 수가 같으면 XOR 연산으로 같은 경우에서만 true, 두 수가 다를 때는 왼쪽을 기준으로 작으면 less가 true, 크면 greater이 true이게 AND연산과 NOT연산으로 구현했다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | A=B | A!=B | A>B | A<B |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

1. **결과 검토 및 논의사항.**

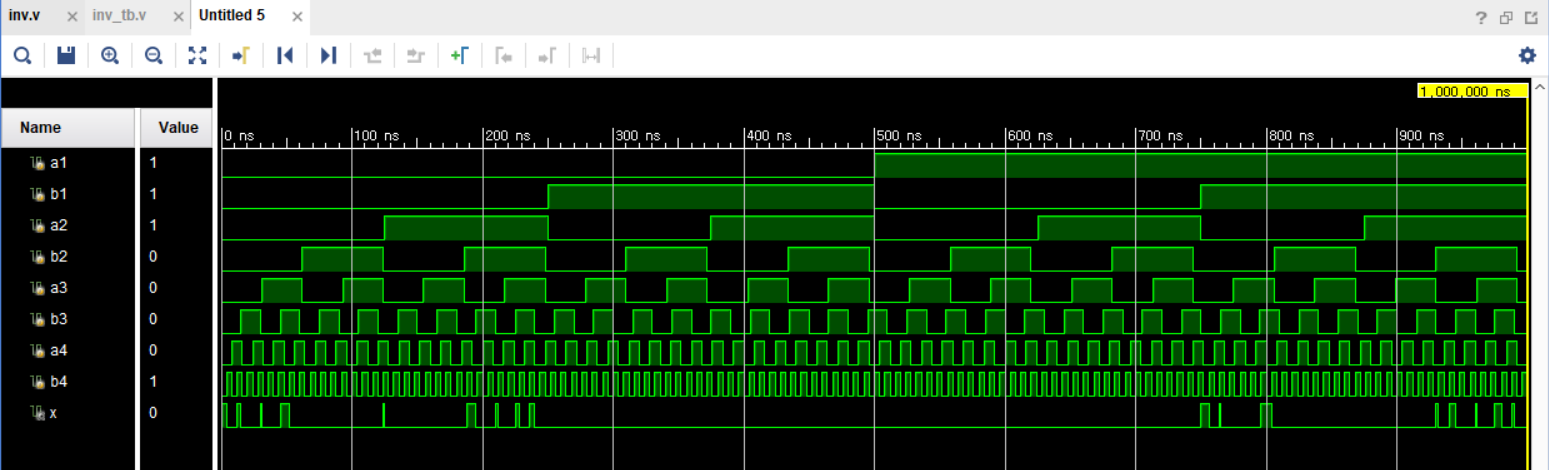
진리표로 예상할 수 있었던 결과와 wave form이 동일한 형태로 나왔다. 드 모르간 법칙은 AND와 OR 연산을 서로 바꾼 뒤, 보수를 취했을 때 동일한 input에서 동일한 output이 나오게 됨을 의미한다. 이때 1법칙은 논리곱을 합으로, 2법칙은 논리합을 곱으로 바꾸는 과정이 되겠는데 위의 3번 결과에서도 볼 수 있듯, 괄호가 중첩되어 있어도 동일하게 바깥쪽 괄호부터 처리해주면 된다.

2input 4output 1bit 비교기에서는 숫자의 equal을 XOR gate로 먼저 비교해주고, comp는 따로 AND와 NOT연산을 이용해 작은 걸 표현할 쪽에 NOT을 붙여줘서 비교한다.

1. **추가 이론 조사 및 작성.**

우리는 1bit 비교기만 확인해보았지만 조금 더 확장해서 4bit 비교기를 만들어보았다.

XOR 연산과 AND 연산으로 a 4개의 bit와 b 4개의 bit가 모두 동일할 때만 x가 true가 되도록 설정하고 돌려봤다.(이하 주요 연산 코드와 simulation 결과)



|  |
| --- |
| `timescale 1ns / 1ps  module inv(  …  assign t1 = ~(a1 ^ b1);  assign t2 = ~(a2 ^ b2);  assign t3 = ~(a3 ^ b3);  assign t4 = ~(a4 ^ b4);  assign x = t1 & t2 & t3 & t4;  endmodule |