5주차 결과보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20231561 이름: 심소현

**1.**

.................

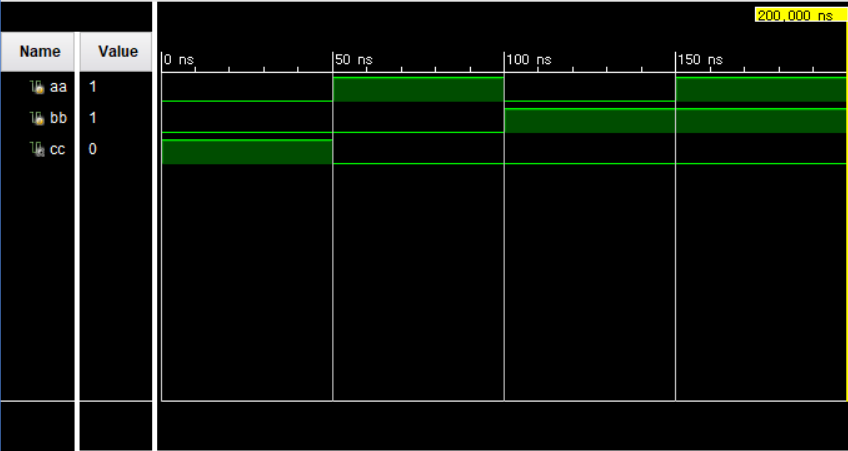
5주차 실험의 목적은 De-Morgan 정리와 Boolean 함수, 1-Bit 비교기를 Verilog로 구현하고 FPGA로 확인하기 위함이다. 실험을 통해서 각각의 경우에 대한 simulation을 확인하고 이론에 따른 결과와 일치하는지 비교할 수 있다.

................

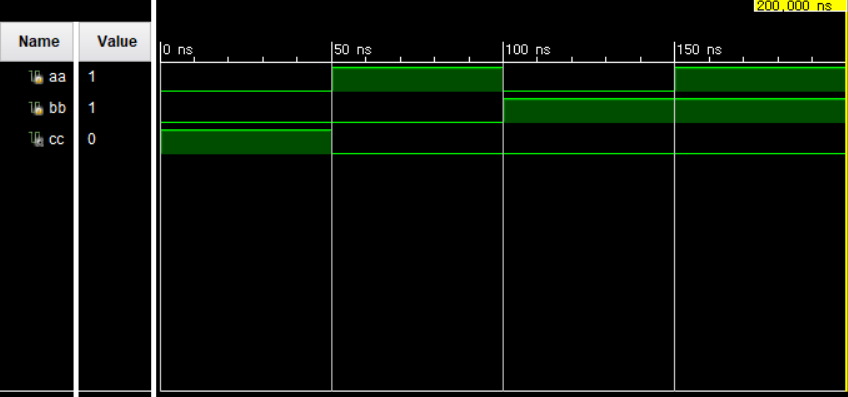
**2.**

.......................

De-Morgan의 제 1법칙에 대한 simulation과 설명이다.



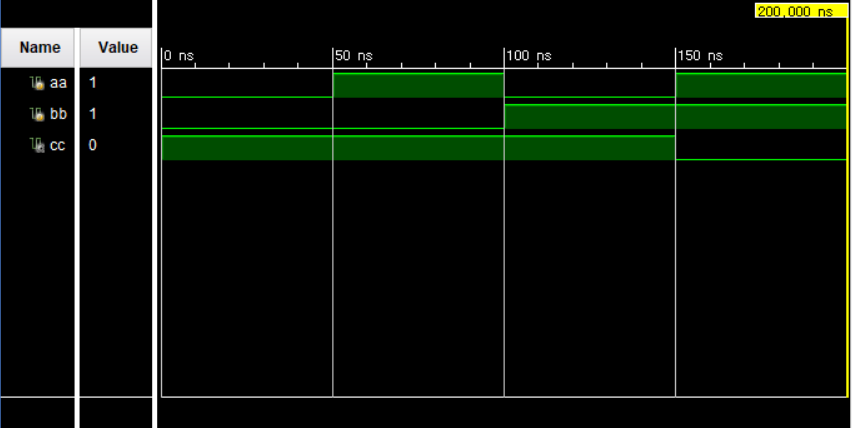
위의 그림은 (A+B)’의 simulation이다.



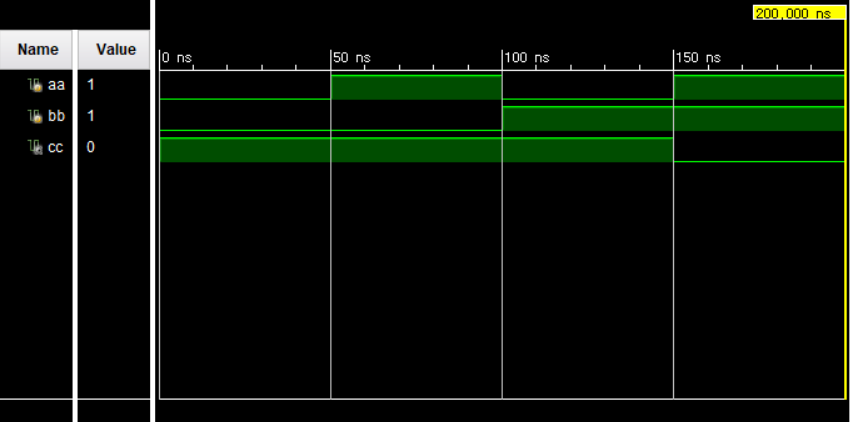
위의 그림은 A’\*B’의 simulation이다.

각 식의 simulation을 보았을 때 input에 대한 output이 같음을 알 수 있다. De-Morgan 제 1법칙은 전체 논리합에 대한 부정은 각각의 변수를 부정한 값의 논리곱과 같다는 것이다. 이는 NOR 연산과 동일하다고 볼 수 있는데, NOR 연산은 A와 B 중 하나라도 1이라면 0을 출력하며 두 값이 모두 0이어야 1을 반환하는 연산자이다. 즉 OR 연산자의 부정이라고 할 수 있기 때문에 A|B의 부정인 (A|B)’가 De-Morgan 제 1법칙에 의하여 A’&B’와 같다는 것을 알 수 있다.

De-Morgan의 제 2법칙에 대한 simulation과 설명이다.



위의 그림은 (A\*B)’의 simulation이다.



위의 그림은 A’+B’의 simulation이다.

각 식의 simulation을 보았을 때 input에 대한 output이 같음을 알 수 있다. De-Morgan 제 2법칙은 전체 논리곱에 대한 부정은 각각의 변수를 부정한 값의 논리합과 같다는 것이다. 이는 NAND 연산과 동일하다고 볼 수 있는데, NAND 연산은 A와 B가 모두 1이라면 0을 출력하며 두 값 중 하나라도 0이 있다면 1을 반환하는 연산자이다. 즉 AND 연산자의 부정이라고 할 수 있기 때문에 A&B의 부정인 (A&B)’가 De-Morgan 제 2법칙에 의하여 A’|B’와 같다는 것을 알 수 있다.

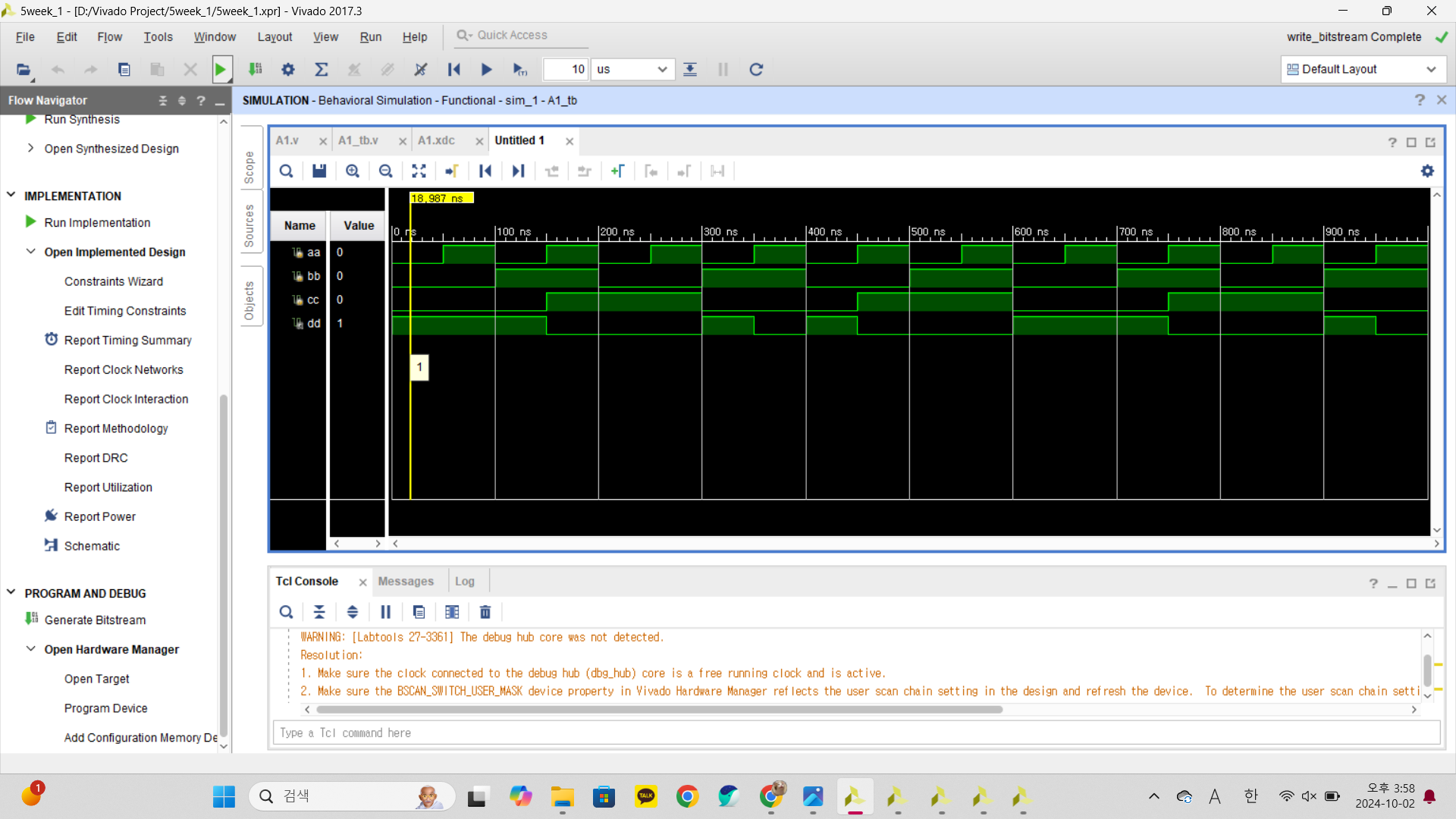
.........................

**3.**

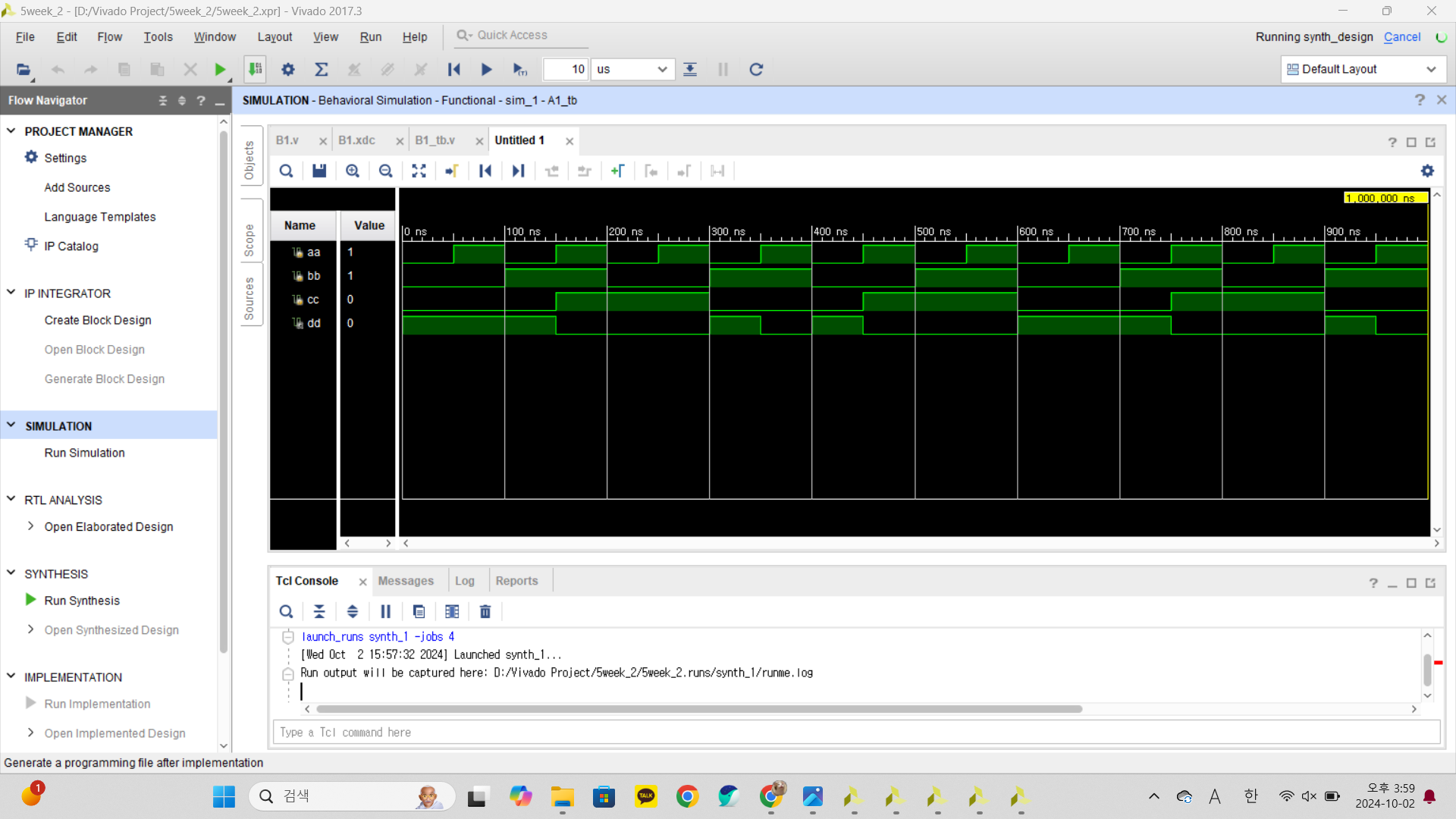
.......................

다음은 Boolean Function의 식에 대한 비교 설명이다.

우선 (A'+B')\*C' = ((A\*B)+C)'에 대하여 설명하겠다.



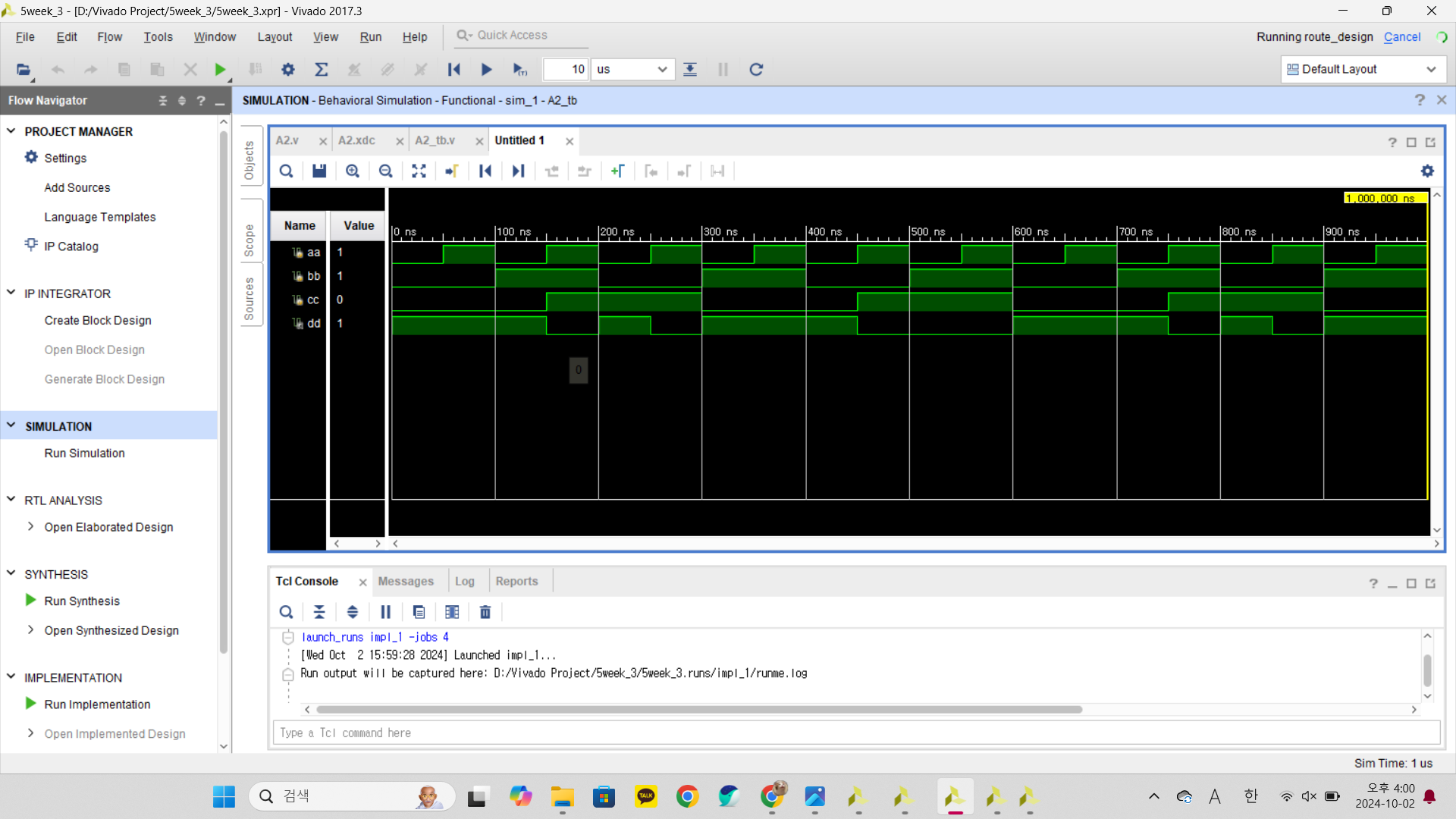
위의 그림은 (A’+B’)\*C’의 simulation 결과이다.



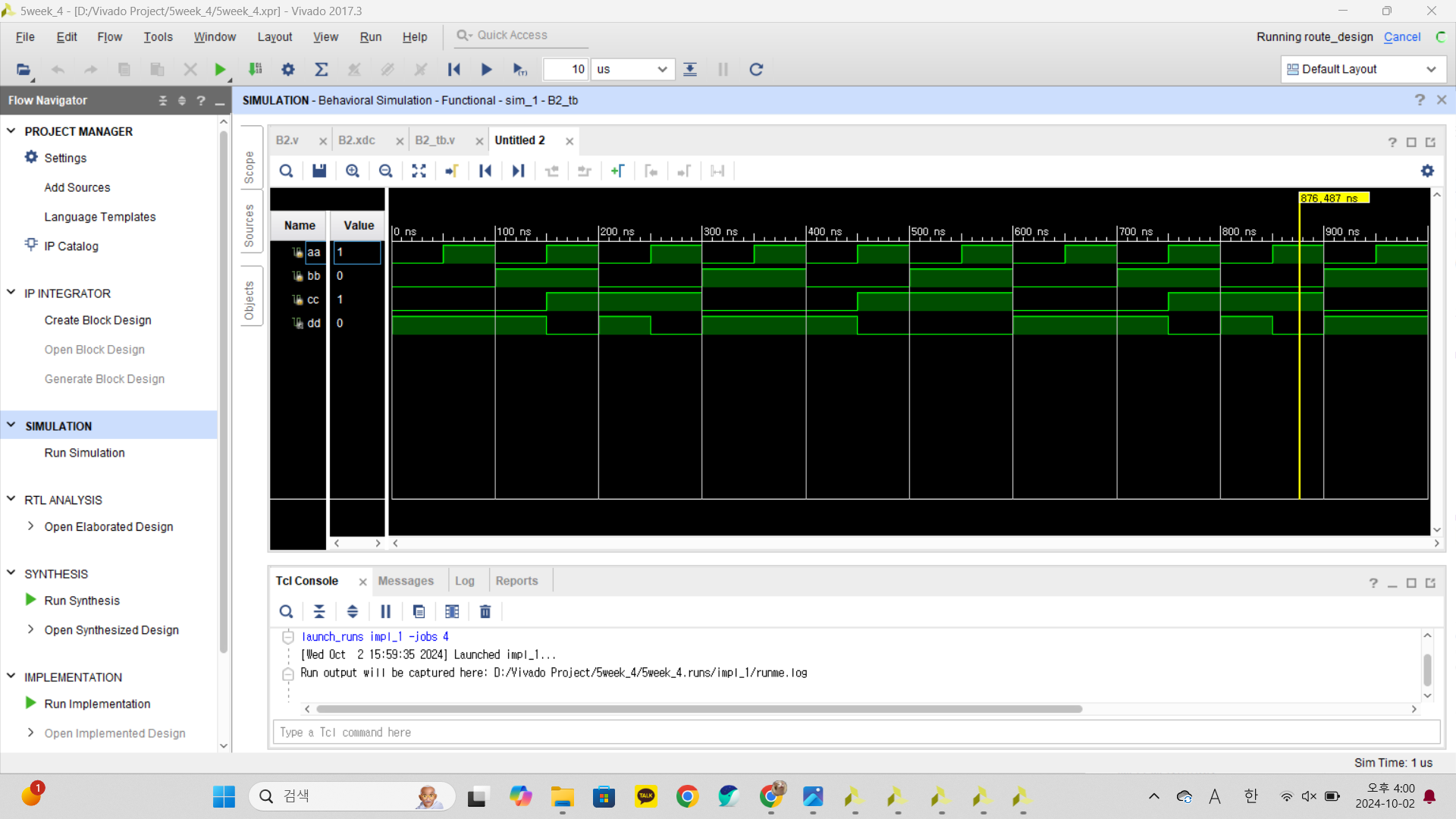
위의 그림은 ((A\*B)+C)’의 simulation 결과이다.

위 두 식의 simulation을 통해 input에 대한 output의 결과가 같음을 알 수 있다. 우선 첫 번째 식은 De-Morgan 제 2법칙에 따라 (A\*B)’와 같다. 이를 적용하면 (A\*B)’\*C’의 식이 된다. 그 후에 이어지는 식에서는 De-Morgan 제 1법칙에 따라 ((A\*B)+C)’와 같음을 알 수 있다. 따라서 De-Morgan 법칙을 응용한 비교식으로 이론에 의하면 좌변과 우변의 값이 같아야 한다. 이를 코드로 나타낸 simulation에서도 옳다는 것이 증명되었으므로 두 표현식이 동일하다는 것을 확인할 수 있었다.

다음으로는 (A'\*B')+C' = ((A+B)\*C)'에 대하여 설명하겠다.



위의 그림은 (A’\*B’)+C’의 simulation 결과이다.



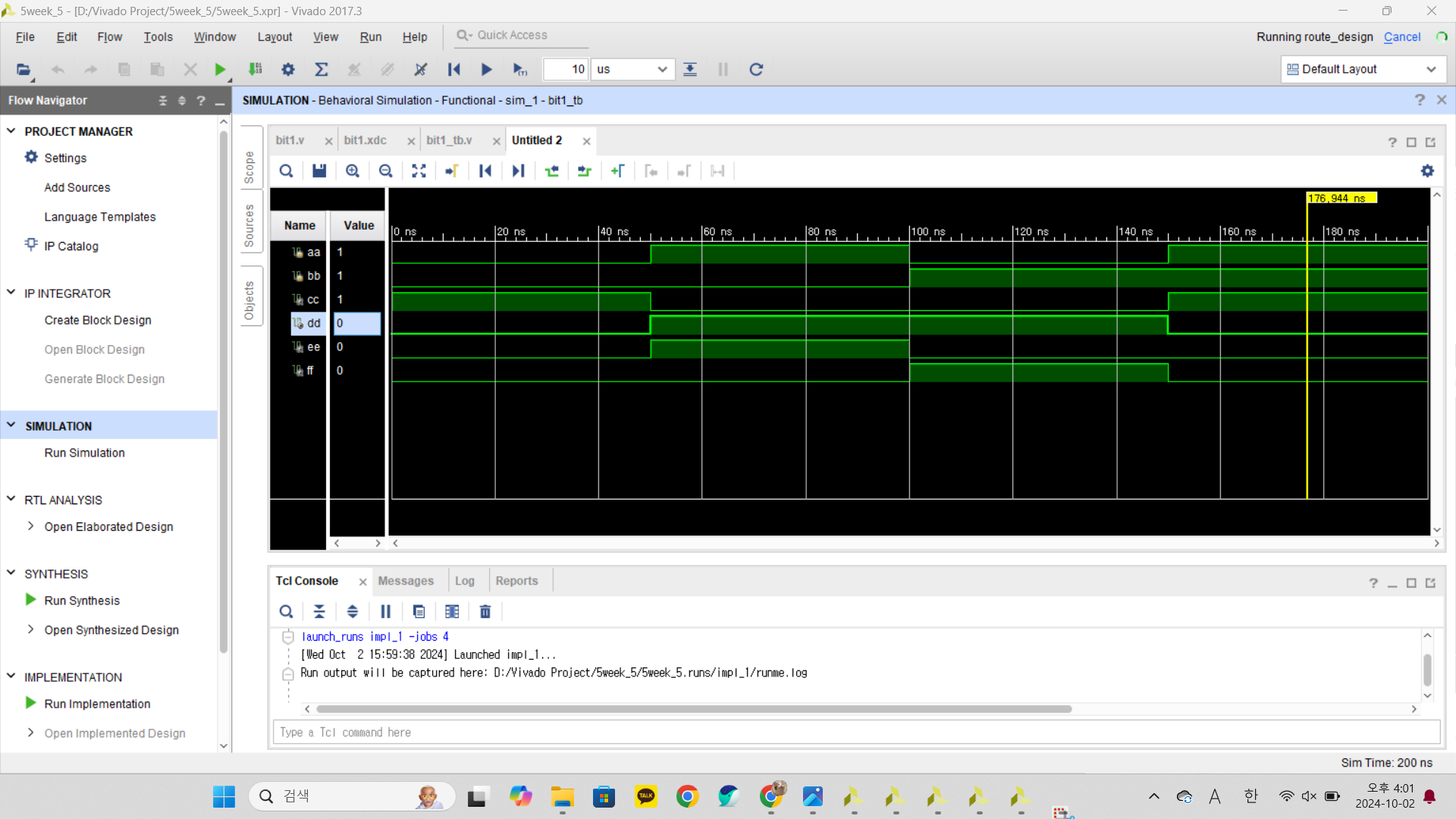
위의 그림은 ((A+B)\*C)’의 simulation 결과이다.

위 두 식의 simulation을 통해 input에 대한 output의 결과가 같음을 알 수 있다. 우선 첫 번째 식은 De-Morgan 제 1법칙에 따라 (A+B)’와 같다. 이를 적용하면 (A+B)’+C’의 식이 된다. 그 후에 이어지는 식에서는 De-Morgan 제 2법칙에 따라 ((A+B)\*C)’와 같음을 알 수 있다. 따라서 De-Morgan 법칙을 응용한 비교식으로 이론에 의하면 좌변과 우변의 값이 같아야 한다. 이를 코드로 나타낸 simulation에서도 옳다는 것이 증명되었으므로 두 표현식이 동일하다는 것을 확인할 수 있었다.

.........................

**4.**

.......................



1 Bit 비교기의 simulation 결과는 위의 그림과 같다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 Bit 비교기의 진리표는 위의 표와 같다.

1 Bit 비교기는 두 개의 입력 비트 값을 비교한 뒤 결과를 4가지 출력으로 나타낸다. 출력에는 A=B, A!=B, A>B, A<B가 있다. 따라서 A와 B의 값이 0 혹은 1인지에 따라서 각각의 결과를 계산하게 되며 입력에 대한 출력은 위의 진리표와 simulation으로 확인할 수 있다.

.........................

**5.**

.......................

이번 실험으로 각 연산의 simulation 결과는 De-Morgan 법칙과 일치하였고 이와 연관된 NAND와 NOR 논리 연산자 또한 검토할 수 있었다. 또한 3가지 입력을 가진 논리식도 De-Morgan 법칙을 응용하여 구성할 수 있으며 복잡한 논리식을 다른 방식으로 변환하는 것 또한 확인할 수 있었다. 1 Bit 비교기 실험에서는 0과 1의 입력에 따라 출력이 어떻게 이루어지는지 모두 확인할 수 있었으며 이와 같은 실험들을 통해 논리 연산의 동작을 simulation을 통하여 알 수 있었다.

.........................

**6.**

.......................

논리식의 변환 과정을 검토하는 실험을 했기 때문에 이를 응용하여 논리 회로를 최적화하는 방안에 대한 조사를 해도 좋을 듯 하다. 회로의 크기와 연산 시간을 줄일 수 있는 효율적인 방법에 대한 연구를 해보고 싶다. 또한 예비 보고서에서 조사했던 카르노 맵의 적용에 대한 조사를 추가적으로 하고 싶다. 최적화된 논리식을 도출하는 방안 중 하나로 어디에 적용해야 더 효율적일지 알아보려 한다.

.........................