7주차 결과보고서

전공: 아트엔테크놀로지 학년: 3학년 학번: 20191172 이름: 함승우

1.

우선 even parity generator는 1이 짝수 개이면 parity bit를 0로, 홀수 개이면 1으로 나온다. 이에 따라 진리표를 작성해보자면

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | E |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 |

이 진리표를 활용하여 카르노 맵을 그려보자면,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CD | 00 | 01 | 11 | 10 |
| AB |  |  |  |  |  |
| 00 |  | 0 | 1 | 0 | 1 |
| 01 |  | 1 | 0 | 1 | 0 |
| 11 |  | 0 | 1 | 0 | 1 |
| 10 |  | 1 | 0 | 1 | 0 |

이는 E = ABC’D + ABCD’ + AB’C’D’ + AB’CD +A’B’C’D + A’B’CD’ + A’BC’D’ + A’BCD이다. 이 수식을 정리하면 xor수식에 따라서 A^B^C^D가 나오게 된다.

따라서 Verilog 코딩은

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

tb파일의 경우에는 입력 값 a, b, c, d를 8, 4, 2, 1 주기로 맞추었다. Simulation 결과는 이렇다

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

자동 생성된 설명

다음은 even parity bit checker로 입력 값에서 에러가 발생했는지 아닌지를 검사하는 역할을 수행한다. 입력 bit들과 패리티 비트를 한꺼번에 연산하여 1이 홀수 개일 경우 이를 오류로 받아들인다. 또한 check bit을 1로 출력한다. 하지만 1이 짝수 개일 경우에는 이를 오류로 간주하지 않으며 check bit 또한 0으로 출력한다. Parity bit를 p로 check bit는 e로 출력을 할 것이고, 나머지 input은 a, b, c, d를 사용할 것이다.

이를 기반으로 진리표를 작성하면,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D | P | e |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 |

5개의 변수를 가지고 카르노 맵을 그려야 하는데, 5개 변수 이상인 경우에는 힘들기 때문에, parity bit을 0일 때와 1일 때를 구분 지어 카르노 맵을 그리겠다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CD | 00 | 01 | 11 | 10 |
| AB |  |  |  |  |  |
| 00 |  | 0 | 1 | 0 | 1 |
| 01 |  | 1 | 0 | 1 | 0 |
| 11 |  | 0 | 1 | 0 | 1 |
| 10 |  | 1 | 0 | 1 | 0 |

위는 p가 0일 때, 아래는 p가 1일 때이다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CD | 00 | 01 | 11 | 10 |
| AB |  |  |  |  |  |
| 00 |  | 1 | 0 | 1 | 0 |
| 01 |  | 0 | 1 | 0 | 1 |
| 11 |  | 1 | 0 | 1 | 0 |
| 10 |  | 0 | 1 | 0 | 1 |

따라서 e는 a^b^c^d^p라고 할 수 있다. 이에 따라 Verilog 코딩을 하면,

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

tb파일 이전과 동일하게 8421 주기로 하며, 마지막 입력값인 p 또한 1의 주기를 따랐다.

이에 대한 결과로 simulation은 아래와 같다.

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

자동 생성된 설명

2.

Odd parity bit generator부터 설명하자면, binary 데이터에서의 1이 홀수 개이면 0으로, 짝수 개이면 1으로 출력한다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | E(parity bit) |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

이 진리표를 바탕으로 카르노 맵을 작성하면,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CD | 00 | 01 | 11 | 10 |
| AB |  |  |  |  |  |
| 00 |  | 1 | 0 | 1 | 0 |
| 01 |  | 0 | 1 | 0 | 1 |
| 11 |  | 1 | 0 | 1 | 0 |
| 10 |  | 0 | 1 | 0 | 1 |

이고 이를 식으로 표현하면, ABC’D’ + ABCD + AB’C’D + AB’CD’+A’B’C’D’ + A’B’CD + A’BC’D + A’BCD’이다. 따라서 이를 정리하면 ~(A ^ B ^ C ^ D)이다. 이에 따라 verilog 코딩은

‘timescale 1ns / 1ps

Module bog(

Input a,

Input b,

Input c,

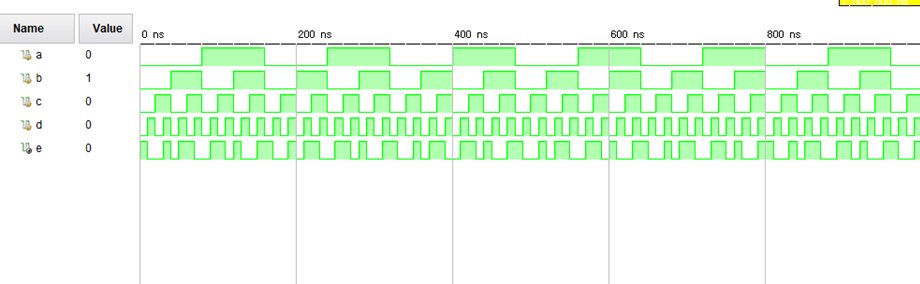
Input d,

Output e);

Assign e = ~(a^b^c^d)

Endmodule

이다.



Odd parity checker는 마찬가지로 오류를 검출하는 모듈이다. 앞서 even parity와 마찬가지지만, 5개의 입력비트 중에 1의 개수가 짝수이면 오류로 간주하고 check bit을 1로 출력한다. 1의 개수가 홀수라면 check bit을 0으로 출력한다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| a | b | c | d | p | e |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |

5개의 변수를 가지고 카르노 맵을 그려야 하는데, 5개 변수 이상인 경우에는 힘들기 때문에, parity bit을 0일 때와 1일 때를 구분지어 카르노 맵을 그리겠다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CD | 00 | 01 | 11 | 10 |
| AB |  |  |  |  |  |
| 00 |  | 1 | 0 | 1 | 0 |
| 01 |  | 0 | 1 | 0 | 1 |
| 11 |  | 1 | 0 | 1 | 0 |
| 10 |  | 0 | 1 | 0 | 1 |

위는 p가 0일 때, 아래는 p가 1일 때이다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CD | 00 | 01 | 11 | 10 |
| AB |  |  |  |  |  |
| 00 |  | 0 | 1 | 0 | 1 |
| 01 |  | 1 | 0 | 1 | 0 |
| 11 |  | 0 | 1 | 0 | 1 |
| 10 |  | 1 | 0 | 1 | 0 |

이 식을 정리하자면,

~(a^b^c^d^p)이다. 이 식을 토대로 Verilog 코딩을 하면,

‘timescale 1ns / 1ps

Module boc(

Input a,

Input b,

Input c,

Input d,

Input p

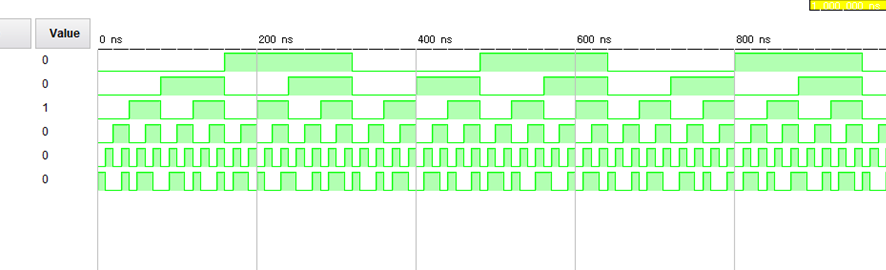
Output e);

Assign e = ~(a^b^c^d^p)

Endmodule

이다.

이에 따른 시뮬레이션 결과는 이렇다.



3.

2 bit의 수가 입력되니 input은 4개로 받는다. 또한 A1A2가 입력된다고 하고, B1B2가 입력된다고 할 때 이 A, B를 compare하는 경우의 수는 A가 B보다 작은 경우, 큰 경우, 같은 경우라고 할 수 있다. 입력을 a1, a2, b1, b2라고 하고, 출력을 r1(a>b)일 때, r2(a=b)일 때, r3(a=b)일때라고 가정해보자. 그렇다면 진리표를 구성해보자면

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A1 | A2 | B1 | B2 | R1 | R2 | R3 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

R1에 대해서 카르노 맵을 그려보자면

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | B1B2 | 00 | 01 | 11 | 10 |
| A1A2 |  |  |  |  |  |
| 00 |  | 0 | 0 | 0 | 0 |
| 01 |  | 1 | 0 | 0 | 0 |
| 11 |  | 1 | 1 | 0 | 1 |
| 10 |  | 1 | 1 | 0 | 0 |

A1A2부분의 주황색 사각형들은 한 묶음이다. 그러므로, 총 3가지의 묶음으로 나타낼 수 있다.

R2는

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | B1B2 | 00 | 01 | 11 | 10 |
| A1A2 |  |  |  |  |  |
| 00 |  | 1 | 0 | 0 | 0 |
| 01 |  | 0 | 1 | 0 | 0 |
| 11 |  | 0 | 0 | 1 | 0 |
| 10 |  | 0 | 0 | 0 | 1 |

총 4가지의 묶음으로 나타낼 수 있고,

마지막으로 r3는

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | B1B2 | 00 | 01 | 11 | 10 |
| A1A2 |  |  |  |  |  |
| 00 |  | 0 | 1 | 1 | 1 |
| 01 |  | 0 | 0 | 1 | 1 |
| 11 |  | 0 | 0 | 0 | 0 |
| 10 |  | 0 | 0 | 1 | 0 |

B1B2의 11부분에서의 주황색 부분은 한 묶음이다. 따라서 R3는 3가지 묶음으로 볼 수 있다.정리하자면

R1 = A1C’ + AC’B2’ + A1A2B2’

R2 = A1’A2’B1’B2’ + A1’A2B1’B2 + A1A2B1B2 + A1A2’B1B2’

R3 = A1’C + A1’A2’B2 + A2’이다.

실습시간에 minterm sop 형식으로 구현하지 않아 코드의 r3 부분과 보고서의 내용과 다르다.

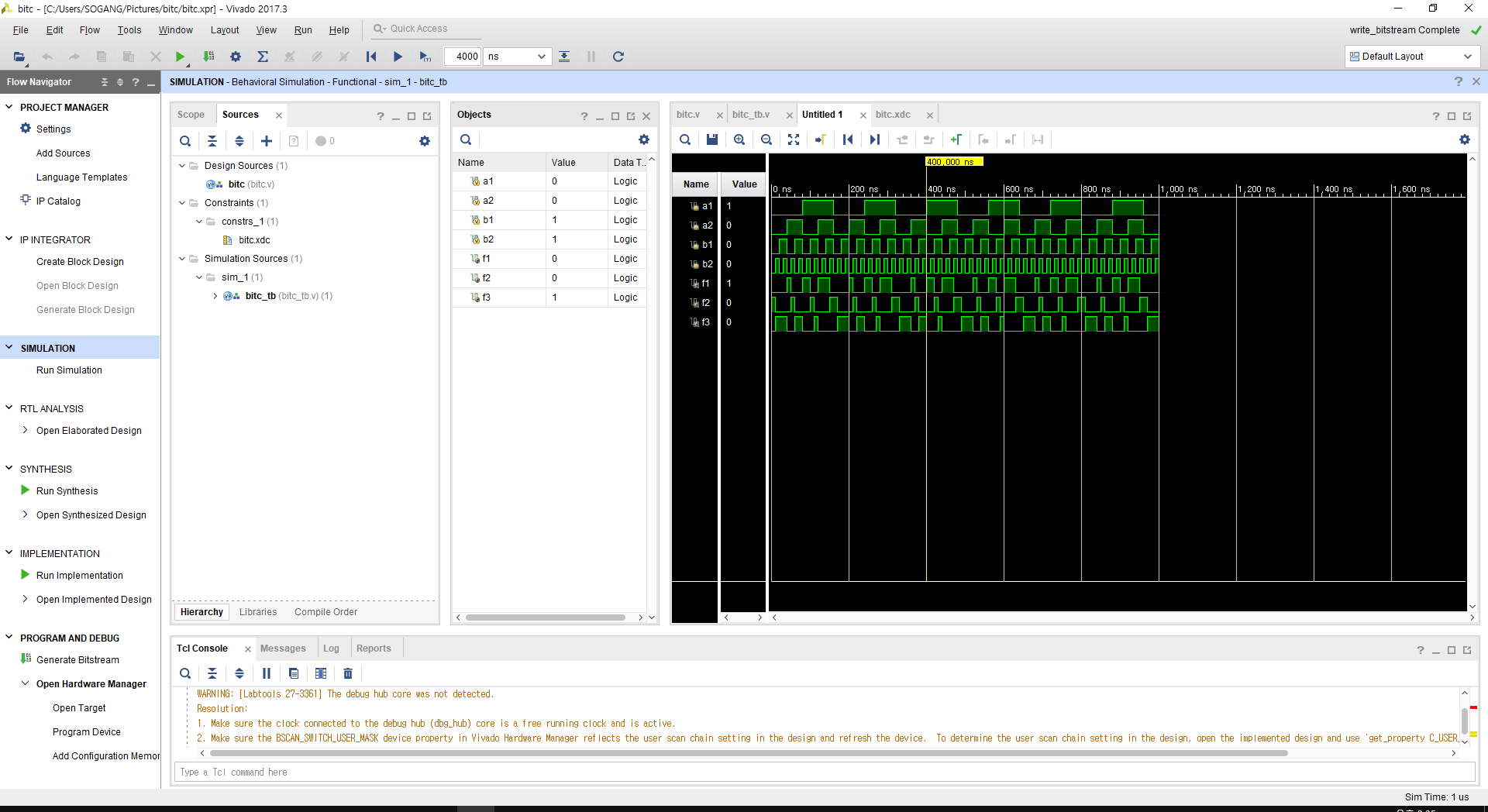
따라서 Verilog 코딩은 이렇게 할 수 있다.

텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 소프트웨어, 디스플레이이(가) 표시된 사진

자동 생성된 설명

tb파일은 a1, a2, b1, b2의 주기를 8, 4, 2, 1로 맞추어 출력하였다. 따라서 시뮬레이션 결과는 아래와 같다.



4.

Parity bit 조사기와 생성기를 살펴보았고, 이때 even과 odd에 따라서 달라진다는 점도 확인했다. 서로 반대되는 결과를 가지는 것도 확인할 수 있었다. 2 bit binary 비교기에서는 2 bit 숫자끼리의 대소비교를 하는 회로를 설계했고, 이 진리표에 따라서 verilog로 코딩하고, simulation 결과를 확인할 수 있었다.

5.

오류의 발생원인을 찾아본 결과 크게 4가지로 파악할 수 있었다. 첫번째로는 전송 신호가 전송 매체를 통과하는 과정에서 거리에 따라 점차 약해지기 때문에 생기는 현상인 감쇠, 두번째로는 주로 유선 전송 매체에서 발생하며, 하나의 전송 매체를 통해 여러 신호를 전달했을 경우 주파수에 따라서 그 속도가 달라져 생기는 오류인 지연 왜곡, 세번째로는 서로 다른 주파수들이 하나의 전송 매체를 공유할 때 주파수 간의 합이나 차로 인해서 새로운 주파수가 생성되는 이유인 상호 변조 잡음, 마지막으로 비연속적이고 외부적인 충격에 의한 결함 등이 포함되어 있는 충격 잡음이 있다. 이러한 이유 때문에, 수신 측에서 오류를 확인하기 위해서 parity bit를 사용한다는 것을 다시 생각해볼 수 있다.