7주차 결과보고서

전공 : 국제한국학과 학년 : 4학년 학번 : 20181202 이름 : 김수미

**1. Even Parity bit generator 및 checker 의 simulation 결과 및 과정에 대해서 설명하시오. (Truth table 작성 및 k-map 포함)**

Even Parity Bit Generator

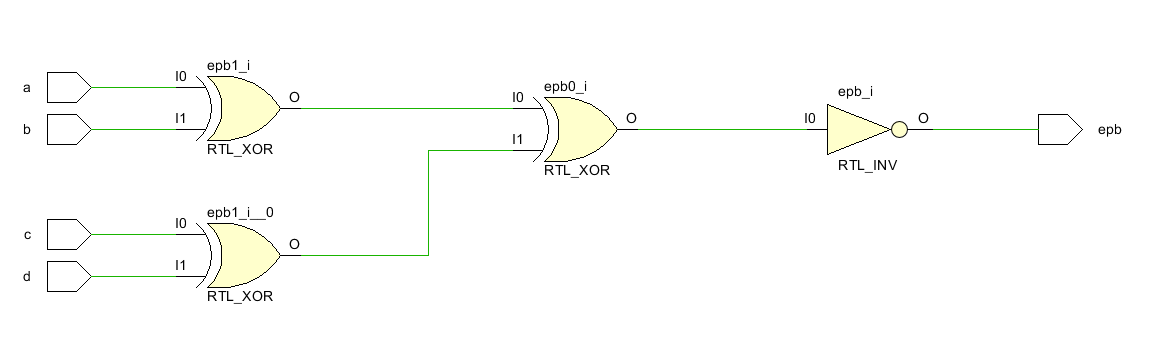
1) Truth Table & Karnaugh Map

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Even Parity Bit Generator | | | | | |
| A | B | C | D | P | Karnaugh Map |
| 0 | 0 | 0 | 0 | 1 | C:\Users\Loyola_work\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1.PNG |
| 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 | SOP Form |
| 1 | 1 | 1 | 1 | 1 | P = A’B’C’D’ + A’B’CD + A’BC’D + A’BCD’  + ABC’D’ + ABCD + AB’C’D + AB’CD’  = ~(A^B^C^D) |

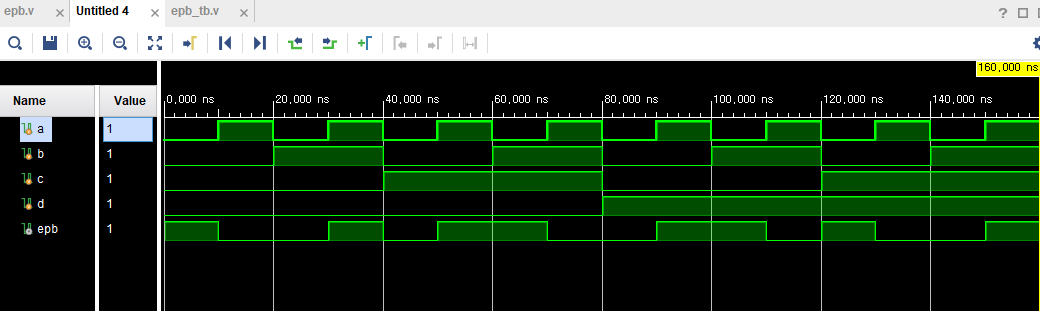
3) Source Code

|  |  |
| --- | --- |
| Design Source | Test Bench |
|  |  |

4) Schematic



5) Simulation



6) 결과 및 과정

Even Parity bit generator는 binary 데이터에서 ‘1’ 의 bit 수가 0개 또는 짝수 개이면 parity bit는 ‘1’ 로, 홀수 개이면 ‘0’ 으로 놓게 된다. 따라서 전체적으로는 항상 홀수 개의 ‘1’ 의 bit 수를 갖는 전송 데이터 형태로 목적지에 전송된다. 따라서 A,B,C,D 총 4 bit가 있으므로 4개의 bit에 1의 개수가 짝수개이면 Parity bit인 epb를 1로, 홀수개이면 0으로 두면 된다. 즉 4개의 bit를 XOR로 검사해주기만 하면 된다.

Even Parity Bit Checker

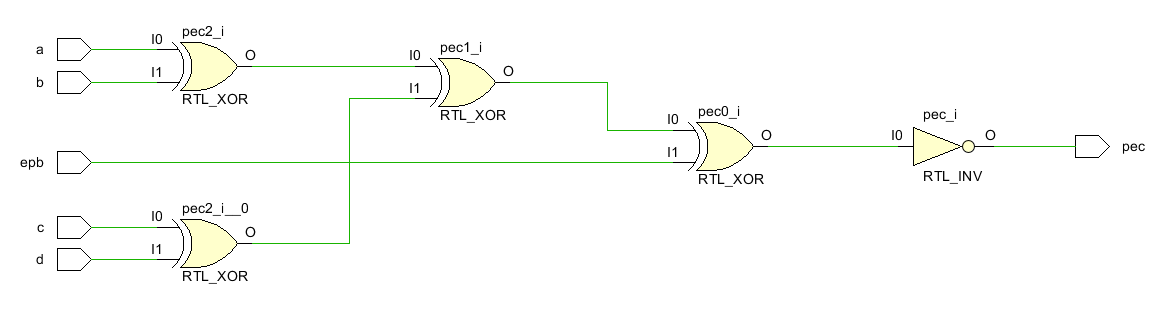
1) Truth Table & Karnaugh Map

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Even Parity Bit Checker | | | | | | |
| A | B | C | D | P | PEC | Karnaugh Map |
| 0 | 0 | 0 | 0 | 1 | 0 | < P = 0 >    < P = 1 > |
| 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 | 0 |
| 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 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 |

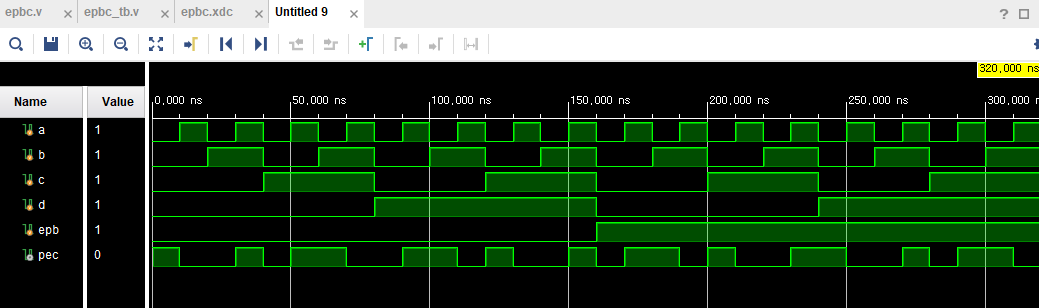
3) Source Code

|  |  |
| --- | --- |
| Design Source | Test Bench |
|  |  |

4) Schematic



5) Simulation



6) 결과 및 과정

Even Parity bit Checker는 입력 bit (4bit)와 parity bit 까지를 포함하여 에러가 발생했는지, 아닌지를 검사하는 역할을 수행한다. 입력 bit 와 parity bit를 포함하여 1이 0개 또는 짝수 개일 경우 이를 오류로 간주하여 check bit는 1을 표시하고, 1이 홀수 개 인 경우 check bit는 0을 표시한다.

위 simulation 결과에서 epb는 parity bit, pec는 check bit 이다. 나머지 bit는 입력 bit 이다.

**2. Odd Parity bit generator 및 checker 의 simulation 결과 및 과정에 대해서 설명하시오. (Truth table 작성 및 k-map 포함)**

Odd Parity Bit Generator

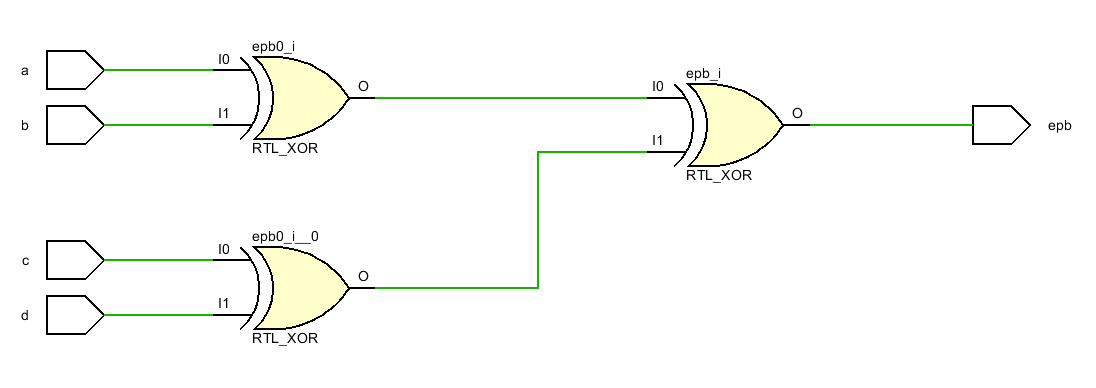
1) Truth Table & Karnaugh Map

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Odd Parity Bit Generator | | | | | |
| A | B | C | D | P | Karnaugh Map |
| 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 | SOP Form |
| 1 | 1 | 1 | 1 | 0 | P = A’B’C’D + A’B’CD’ + A’BC’D’ + A’BCD + ABC’D + ABCD’ + AB’C’D’ + AB’CD  = (A^B^C^D) |

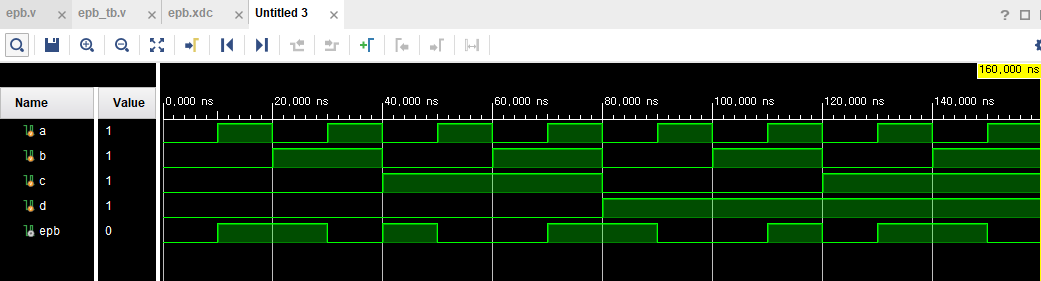
3) Source Code

|  |  |
| --- | --- |
| Design Source | Test Bench |
|  |  |

4) Schematic



5) Simulation



6) 결과 및 과정

Odd Parity Bit Generator는 binary 데이터에서 ‘1’ 의 bit 수가 홀수 개이면 parity bit는 ‘1’ 로,  
0개 또는 짝수 개이면 ‘0’ 으로 놓게 된다. 따라서 전체적으로는 항상 짝수 개의 ‘1’ 의 bit 수를 갖는 전송 데이터 형태로 목적지에 전송된다. 따라서 A,B,C,D 총 4 bit가 있으므로 4개의 bit에 1의 개수가 홀수 개이면 Parity bit인 epb를 1로, 짝수개이면 0으로 두면 된다.

Odd Parity Bit Checker

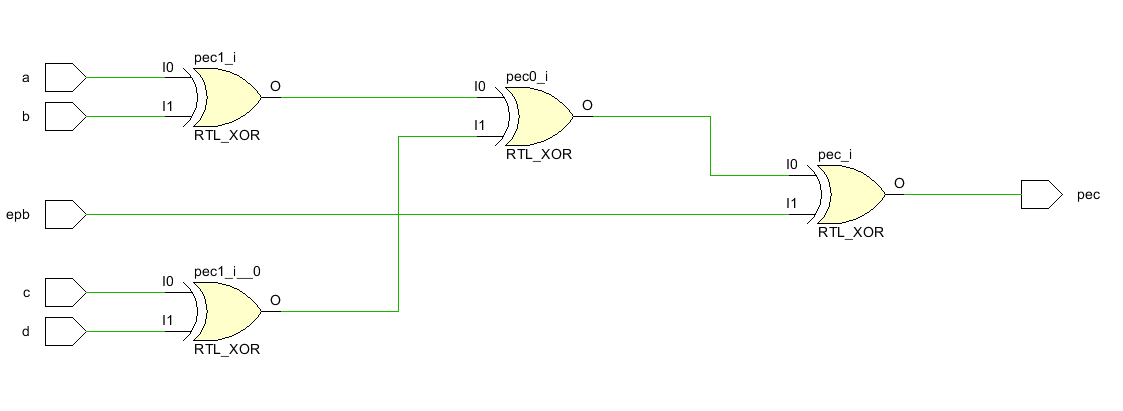
1) Truth Table & Karnaugh Map

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Odd Parity Bit Checker | | | | | | |
| A | B | C | D | P | PEC | Karnaugh Map |
| 0 | 0 | 0 | 0 | 1 | 1 | < P = 0 >    < P = 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 |

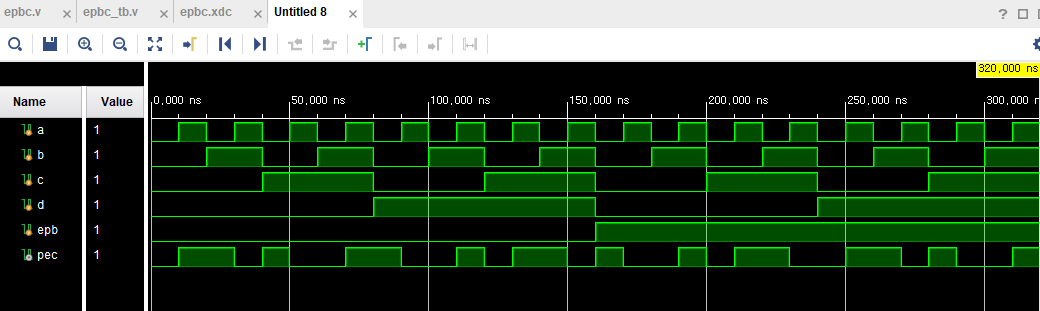
3) Source Code

|  |  |
| --- | --- |
| Design Source | Test Bench |
|  |  |

4) Schematic



5) Simulation



6) 결과 및 과정

Odd Parity bit Checker는 입력 bit (4bit)와 parity bit 까지를 포함하여 에러가 발생했는지, 아닌지를 검사하는 역할을 수행한다. 입력 bit 와 parity bit를 포함하여 1이 홀수 개일 경우 이를 오류로 간주하여 check bit는 1을 표시하고, 1이 0개 또는 짝수 개 인 경우 check bit는 0을 표시한다.

위 simulation 결과에서 epb는 parity bit, pec는 check bit 이다. 나머지 bit는 입력 bit 이다.

**3. 2-bit binary comparator simulation 결과 및 과정에 대해서 설명하시오. (Truth table 작성 및 k-map 포함)**

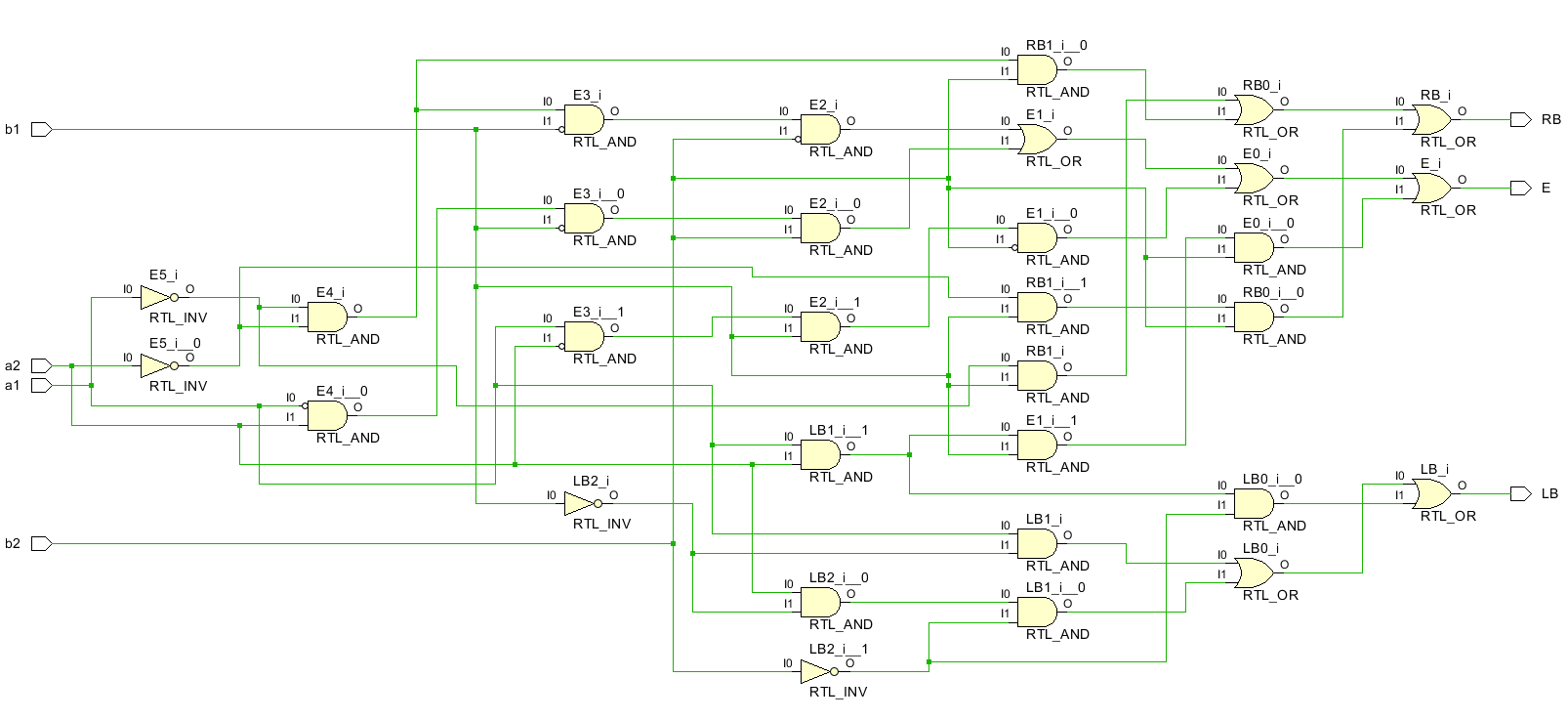
1) Truth Table & Karnaugh Map

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 2-bit Binary Comparator | | | | | | | |
| A1 | A2 | B1 | B2 | LB | E | RB | Karnaugh Map |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | LB |
| 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 | E |
| 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 | RB |
| 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 | SOP Form |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 | LB = AC’ + BC’D’ + ABD’  E = A’B’C’D’ + A’BC’D + ABCD + AB’CD’  RB = A’C + A’B’D + B’ |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

3) Source Code

|  |  |
| --- | --- |
| Design Source | Test Bench |
|  |  |

4) Schematic



5) Simulation



6) 결과 및 과정

2bit 짜리 수 두개의 대소비교를 하는 회로이다. A1,A2 가 하나의 2bit 짜리 수를 이루고, B1,B2가 다른 하나의 2bit 짜리 수를 이룬다. 이 때 A1과 B1이 MSB이다.

A1A2 그리고 B1B2를 나란히 둔 채로 대소비교를 한다고 가정하며, LB는 왼쪽이 큰 경우, 즉 A1A2의 값이 B1B2보다 큰 경우에 1의 값을 가지며 E 는 두 값이 같은 경우 1의 값을, RB는 오른쪽이 더 큰 경우, 즉 B1B2의 값이 A1A2보다 큰 경우에 1의 값을 가지게 된다.

**4. 결과 검토 및 논의 사항**

5개의 변수가 존재하는 경우 카르노맵을 그리는 것은 매우 번거롭다. 주어진 변수 및 Truth Table이 주어졌을 때 더 빠르게 논리식을 구할 수 있는 방법을 찾아 사용할 수 있다면 실습 시간을 줄일 수 있을 것 같다.

**5. 추가 이론 조사 및 작성**

1) Checksum

Parity Bit Checker 에서는 단순히 1의 개수가 짝수인지, 홀수인지 개수를 세어 오류가 발생했는지를 확인했다. Checksum이란 이와 거의 유사한 방법으로, 1의 발생 개수를 세는 대신 모든 Bit의 숫자를 더해 그 연산값이 홀수인지 짝수인지를 확인하는 방식이다.

단 XOR gate를 이용하여 Parity check를 하는 경우 전체 Bit 중 두개의 Bit에 오류가 발생하면 홀, 짝은 그대로이기 떄문에 오류를 검출할 수 없지만, checksum을 이용하면 전체 합을 알고 있기 때문에 2개 이상의 bit 오류가 발생하더라도 이를 검출할 수 있다는 장점이 있다.