전자전기컴퓨터설계실험2

예비보고서

Week #2

전자전기컴퓨터공학부

2017440050

박종범

**6.1 2-bit AND, OR, XOR, NOR, NAND의 진리표를 각각 작성하시오.**

|  |  |
| --- | --- |
| AND | OR |
| |  |  |  | | --- | --- | --- | | A | B | A AND B | | 0 | 0 | 0 | | 0 | 1 | 0 | | 1 | 0 | 0 | | 1 | 1 | 1 | | |  |  |  | | --- | --- | --- | | A | B | A OR B | | 0 | 0 | 0 | | 0 | 1 | 1 | | 1 | 0 | 1 | | 1 | 1 | 1 | |
| XOR | NOR |
| |  |  |  | | --- | --- | --- | | A | B | A XOR B | | 0 | 0 | 0 | | 0 | 1 | 1 | | 1 | 0 | 0 | | 1 | 1 | 1 | | |  |  |  | | --- | --- | --- | | A | B | A NOR B | | 0 | 0 | 1 | | 0 | 1 | 0 | | 1 | 0 | 0 | | 1 | 1 | 0 | |
| NAND |  |
| |  |  |  | | --- | --- | --- | | A | B | A NAND B | | 0 | 0 | 0 | | 0 | 1 | 1 | | 1 | 0 | 1 | | 1 | 1 | 1 | |

**6.2 AND, OR, XOR, NOR, NAND의 Verilog 연산자 표현 방법을 조사하여 쓰시오.**

|  |  |
| --- | --- |
| 연산자 | Verilog 표현방법 |
| AND | Logic: &&, Bit wise: & |
| OR | Logic: ||, Bit wise: | |
| XOR | Bit wise: ^ |
| NOR | 축약: ~| |
| NAND | 축약: ~& |

**6.3 2개의 input을 갖고 5개의 output이 각각 AND, OR, XOR, NOR, NAND 게이트인 logic의 Verilog 코드를 작성하시오. 이 때 input과 output은 아래표를 참조하여 DIP switch와 LED에 연결하시오.**

**6.3.1 위 logic은 FPGA상에서 몇 개의 slice와 몇 개의 LUT를 사용할 것으로 예상되는가? 예상하는 수량과 이유를 기재하시오.**

위의 logic은 slice 1개와 3개의 LUT를 사용할 것으로 예상한다.

위의 logic은 1bit input 2를 받아 1bit output 5개를 출력하므로 LUT2 5개를 사용할 것이다. LUT6는 5비트 이하의 LUT 로 사용할 경우에는 나눠서 두 개의 LUT 2로 사용할 수 있으므로 2개의 output을 출력하는 LUT 2개와 1개의 output을 출력하는 LUT 1개로 총 3개의 LUT를 사용할 것이다. 또한 1 slice에는 LUT가 4개 있으므로 슬라이스 1개만으로 위의 logic을 구현할 수 있을 것이다.

**6.3.2 Vivado에서 implementation까지 수행 후 결과 레포트에서 slice 및 LUT 사용 개수가 적힌 부분을 캡쳐하여 레포트에 그림을 첨부하시오.**

|  |
| --- |
| 테이블이(가) 표시된 사진  자동 생성된 설명 |
| Slice 2개와 LUT 3개가 사용되었다.  이는 6.31에서 슬라이스 1개와 LUT 3개로 동작할 것이라 예측한 것과 결과가 다른데 이는 FPGA가 최적화를 진행할 때 사용하는 slice 개수와 chip의 면적보다는 output pin과의 연결관계를 고려하여 최적의 동작속도로 동작하게끔 slice를 할당하여 그런 것으로 생각된다. |

**6.3.3 시뮬레이션으로 input을 2’b00부터 2’b11까지 변화시켰을 때, output 파형을 5 개 각각을 캡쳐하여 레포트에 첨부하시오.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 입력 | a, b |  | | | |
| a = 0, b = 0 | a = 0, b = 1 | a = 1, b = 0 | a = 1, b = 1 |
| 출력 | v |  | | | |
| 0 | 0 | 0 | 1 |
| w |  | | | |
| 0 | 1 | 1 | 1 |
| x |  | | | |
| 0 | 1 | 1 | 0 |
| y |  | | | |
| 1 | 0 | 0 | 0 |
| z |  | | | |
| 1 | 1 | 1 | 0 |

**6.3.4 해당 시뮬레이션 결과 중 AND 게이트 결과만 시뮬레이션 결과를 토대로 진리 표를 작성 후 5.1에서 작성한 진리표와 비교하시오.**

|  |  |
| --- | --- |
| AND 게이트 시물레이션 결과 | 5.1에서 작성한 진리표 |
| |  |  |  | | --- | --- | --- | | a | a | v | | 0 | 0 | 0 | | 0 | 1 | 0 | | 1 | 0 | 0 | | 1 | 1 | 1 | | |  |  |  | | --- | --- | --- | | A | B | A AND B | | 0 | 0 | 0 | | 0 | 1 | 0 | | 1 | 0 | 0 | | 1 | 1 | 1 | |

AND 게이트 시물레이션 결과는 5.1에서 작성한 진리표와 일치한다.

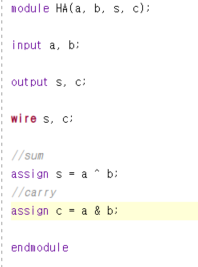
**6.4 Half adder의 진리표를 구한 후 카르노맵을 이용하여 C와 S에 대한 논리식을 각각 구 하시오.**

텍스트, 화이트보드이(가) 표시된 사진

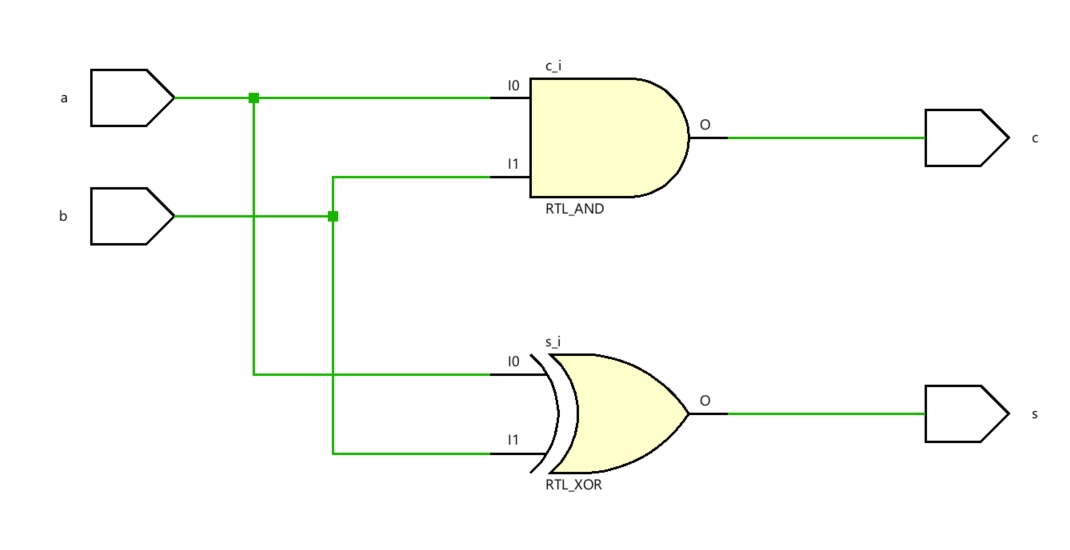
자동 생성된 설명

* C = A xor B
* S = AB

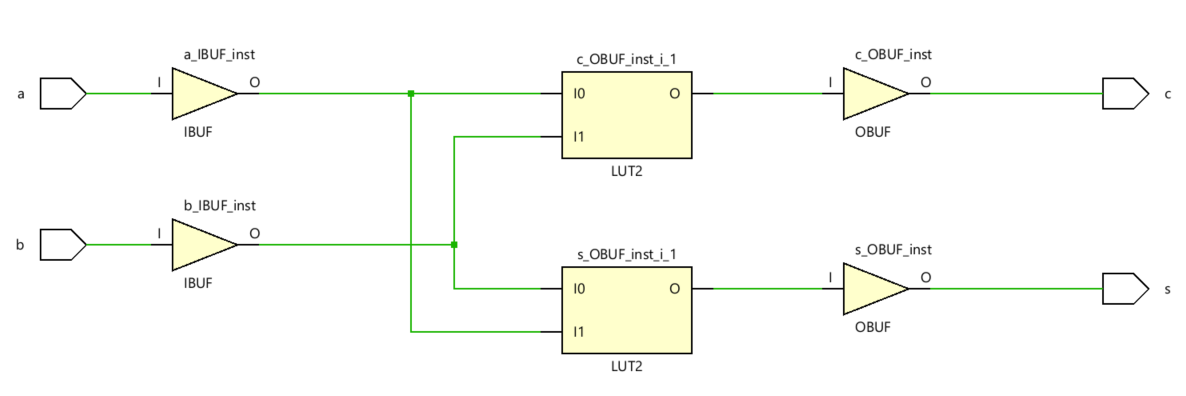
**6.5 Half adder를 6.4에서 구한 논리식을 이용하여 Verilog 코드를 작성하여 구현하시오.**



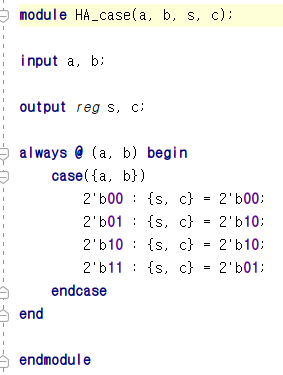
**6.5.1 synthesis 하기 전 schematic을 캡쳐하여 레포트에 첨부하시오.**



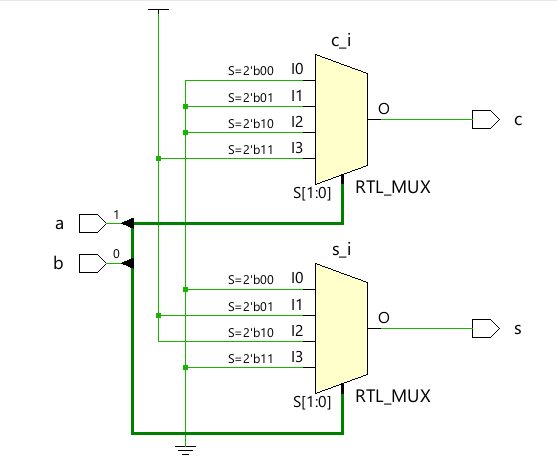
**6.5.2 Implementation 한 후 shcematic을 캡쳐하여 레포트에 첨부하시오.**



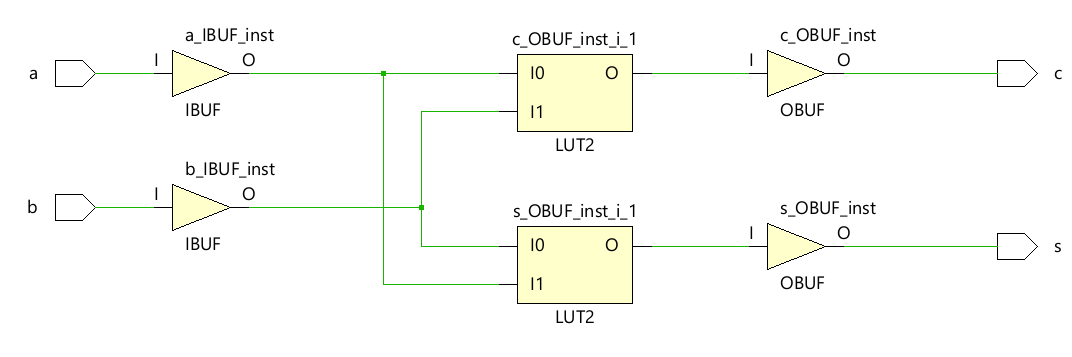
**6.6 Half adder를 case문을 이용하여 Verilog 코드를 작성하여 구현하시오.**

****

**6.6.1 synthesis 하기 전 schematic을 캡쳐하여 레포트에 첨부하시오.**

****

**6.6.2 Implementation 한 후 shcematic을 캡쳐하여 레포트에 첨부하시오.**

****

**6.7 Synthesis 하기 전 6.5.1과 6.6.1의 schematic은 차이가 있는가? (yes or no만 기재)**

Yes

**6.8 implementation까지 마친 후 5.5.2과 5.6.2의 schematic은 차이가 있는가? (yes or no만 기재)**

NO

**6.9 Full adder를 구현 후 input을 3’b000부터 3’b111까지 바꿨을 때 input/output 시뮬레 이션 파형을 캡쳐하여 첨부하시오. (input은 DIP 스위치, output은 LED)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
| x | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| y | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| z | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| s | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| c | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |