4주차 결과보고서

전공: 기계공학과 학년: 3학년 학번: 20191820 이름: 김형준

**1.**

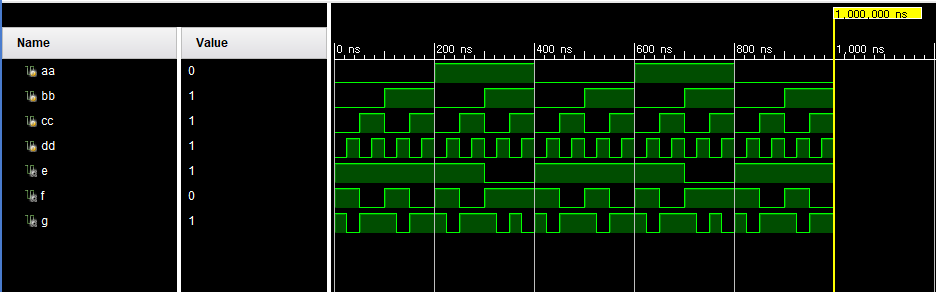
▪ NAND, NOR, XOR, AOI Gate의 원리와 동작을 이해한다.

▪ Verilog를 사용해 다중입력 NAND, NOR, XOR, AOI Gate를 구현한다.

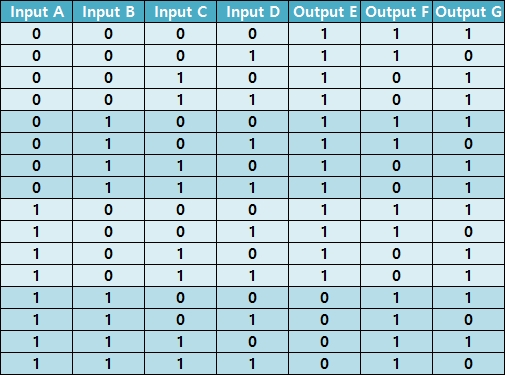
▪ 입력 신호 생성 후 Simulation을 통해 Verilog로 구현된 각 Gate의 동작을 확인한다.

▪ FPGA를 통해서 Verilog로 구현된 각 회로의 동작을 확인한다.

**2.**



[4-input NAND gate의 simulation 결과]

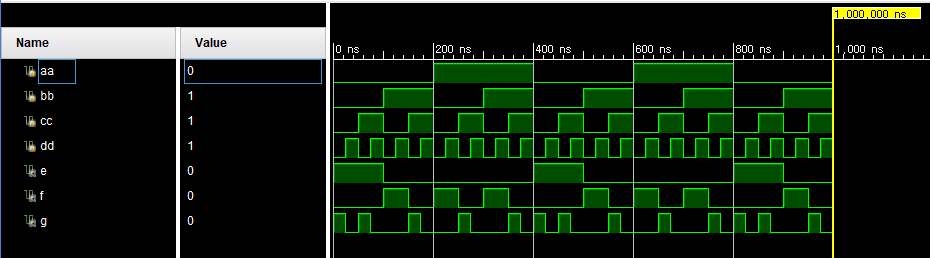


[4-input NAND gate의 진리표]

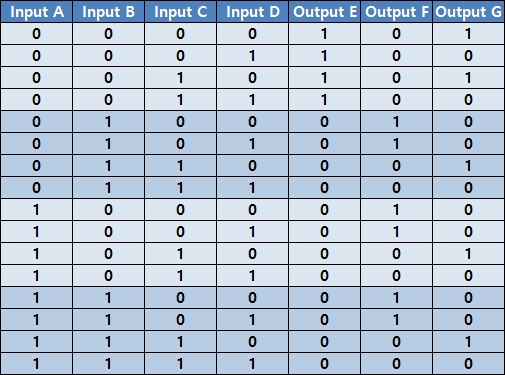
4-input NAND gate (B)에서 입력으로 A, B, C, D를, 출력으로 E, F, G를 갖는다.

E는 A와 B의 NAND 연산의 결과이다. (E=~(A&B)) 따라서 E는 A, B가 모두 1일때만 0을 갖고, 나머지는 1을 갖게 된다. E와 마찬가지로 F는 E와 C의 NAND 연산의 결과이고 (F=~(E&C)), G는 F와 D의 NAND 연산의 결과이다. (G=~(F&D)) 따라서 F는 E, C가, G는 F, D가 모두 1일때만 0을 갖고, 나머지는 1을 갖게 된다. 위와 같이 (B)에서는 4-input NAND gate를 NAND 게이트 3개로 구현 했고, (A)에서는 NAND 게이트 1개에 모든 입력을 한번에 받는 방식으로 구현했는데, 각각의 Boolean 식은 (A), (B)로 일치하지 않으므로 최종 Output 값은 서로 달라지게 된다.

**3.**



[4-input NOR gate의 simulation 결과]

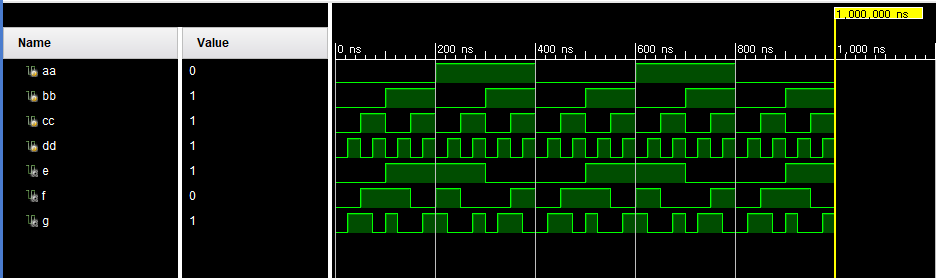


[4-input NOR gate의 진리표]

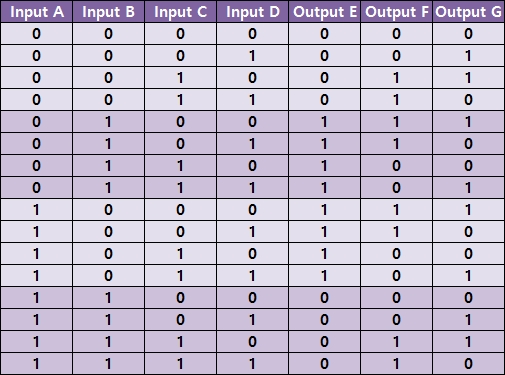
4-input NOR gate (B)에서 입력으로 A, B, C, D를, 출력으로 E, F, G를 갖는다.

E는 A와 B의 NOR 연산의 결과이다. (E=~(A|B)) 따라서 E는 A, B가 모두 0일때만 1을 갖고, 나머지는 0을 갖게 된다. E와 마찬가지로 F는 E와 C의 NOR 연산의 결과이고 (F=~(E|C)), G는 F와 D의 NOR 연산의 결과이다. (G=~(F|D)) 따라서 F는 E, C가, G는 F, D가 모두 0일때만 1을 갖고, 나머지는 0을 갖게 된다. 위와 같이 (B)에서는 4-input NOR gate를 NOR 게이트 3개로 구현 했고, (A)에서는 NOR 게이트 1개에 모든 입력을 한번에 받는 방식으로 구현했는데, 각각의 Boolean 식은 (A), (B)로 일치하지 않으므로 최종 Output 값은 서로 달라지게 된다.

**4.**



[4-input XOR gate의 simulation 결과]



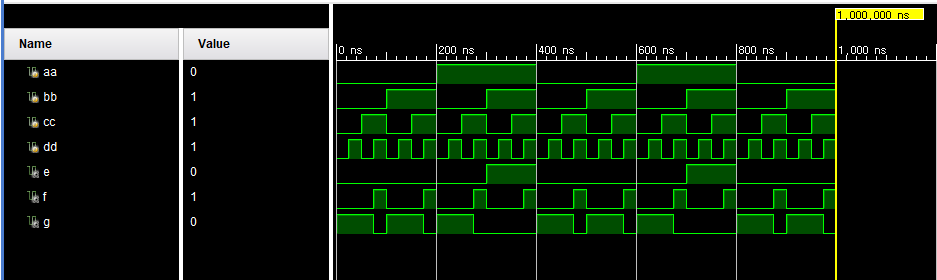
[4-input XOR gate의 진리표]

4-input XOR gate (B)에서 입력으로 A, B, C, D를, 출력으로 E, F, G를 갖는다.

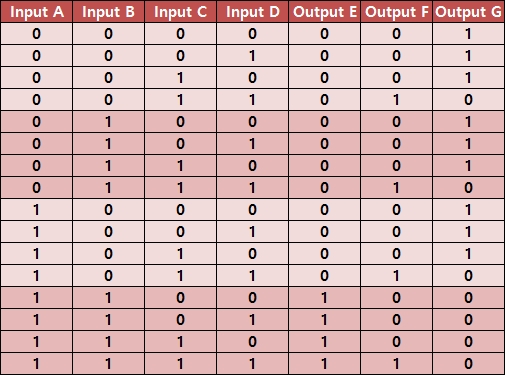
E는 A와 B의 XOR 연산의 결과이다. (E=~(A^B)) 따라서 E는 A, B가 같을 때 0을 갖고, 다를 때 1을 갖게 된다. E와 마찬가지로 F는 E와 C의 XOR 연산의 결과이고 (F=~(E^C)), G는 F와 D의 XOR 연산의 결과이다. (G=~(F^D)) 따라서 F는 E, C가, G는 F, D가 같을 때 0을 갖고, 다를 때 1을 갖게 된다. 결과적으로 G는 A, B, C, D 중 1이 홀수 개 일 때 1을 갖고, 짝수 개 일 때 0을 갖게 된다.

위와 같이 (B)에서는 4-input XOR gate를 XOR 게이트 3개로 구현 했고, (A)에서는 XOR 게이트 1개에 모든 입력을 한번에 받는 방식으로 구현했는데, 각각의 Boolean 식은 (A), (B)로 일치하므로 (XOR 연산은 결합법칙이 성립함) 최종 Output 값은 서로 같게된다.

**5.**

****

[4-input AOI gate의 simulation 결과]



[4-input AOI gate의 진리표]

4-input AOI gate 에서 입력으로 A, B, C, D를, 출력으로 E, F, G를 갖는다.

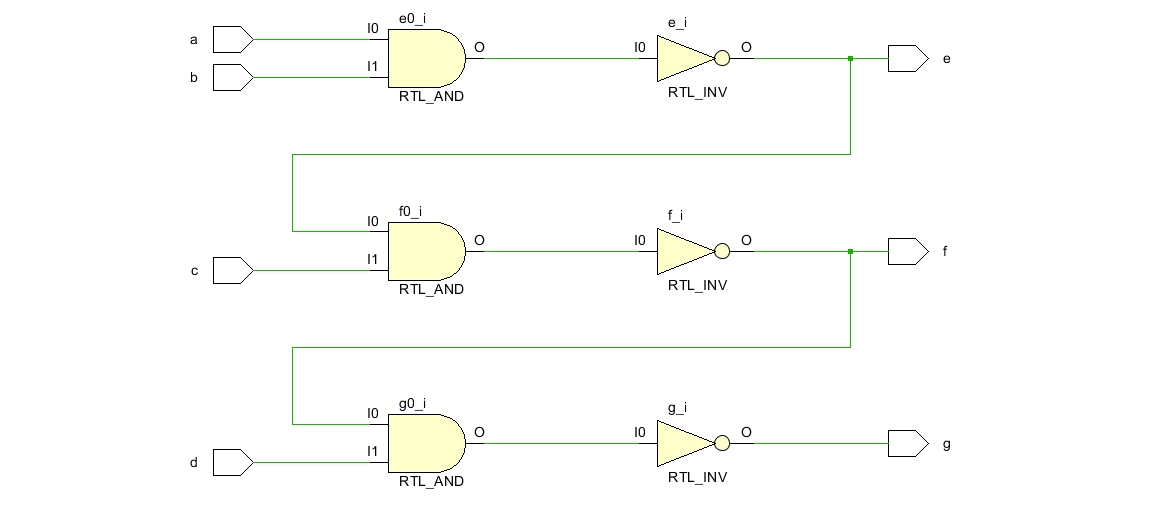
E는 A와 B의 AND 연산의 결과이다. (E=A&B) 따라서 E는 A, B가 모두 1일때만 1을 갖고, 나머지는 0을 갖게 된다. F는 C와 D의 AND 연산의 결과이다. (F=C&D) 따라서 F는 C, D가 모두 1일때만 1을 갖고, 나머지는 0을 갖게 된다. G는 E와 F의 NOR 연산의 결과이다. (G=~(E|F)) 따라서 G는 E, F가 모두 0일때만 1을 갖고, 나머지는 0을 갖게 된다. 결과적으로 G는 E와 F가 0일 때만 1을 가지므로 (A,B), (C,D)가 모두 0일 때만 1을 갖고, 나머지의 경우는 0을 갖는 것을 알 수 있다.

**6.**

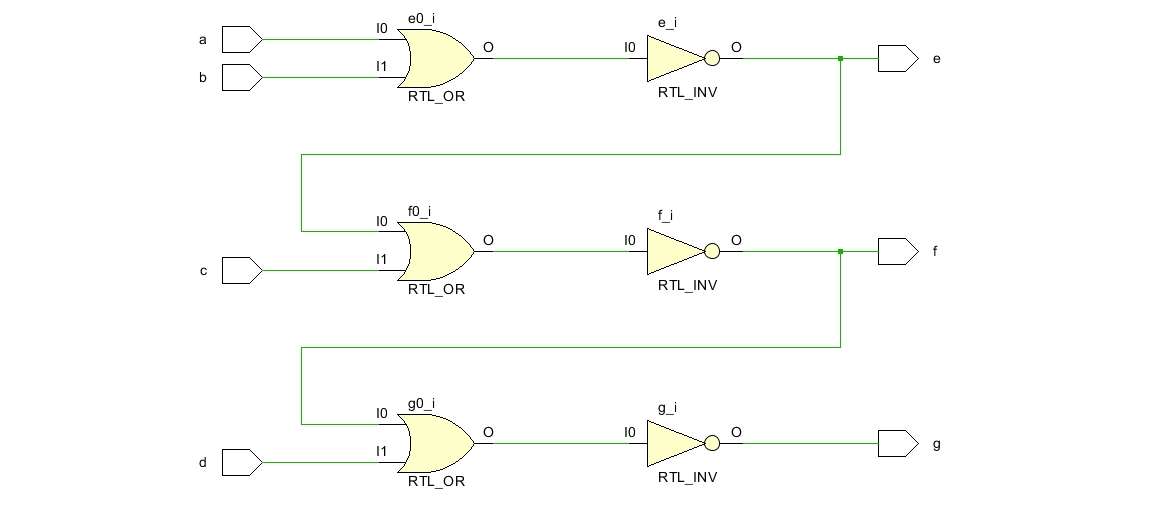
2. ~ 5.까지 각각의 시뮬레이션 결과와 진리표를 비교했을 때 차이가 없으므로 의도한 대로 코드가 작성되었음을 확인 할 수 있다. 이 실험을 통해 다중입력 AOI 게이트의 구조와 동작을 이해할 수 있었고, 다중입력 XOR 게이트는 입력 중 1의 개수가 홀수일 때 1을 반환하고 아닌 경우에는 0을 반환하는 것을 알 수 있었다.

또한, 다중입력 NAND 게이트와 NOR 게이트는 결합법칙이 성립하지 않아, 다중입력 게이트를 구성할 때 하나의 게이트로 구성할지, 또는 여러 개의 게이트로 구성할지에 따라 결과값이 달라지는 것을 알 수 있었다.

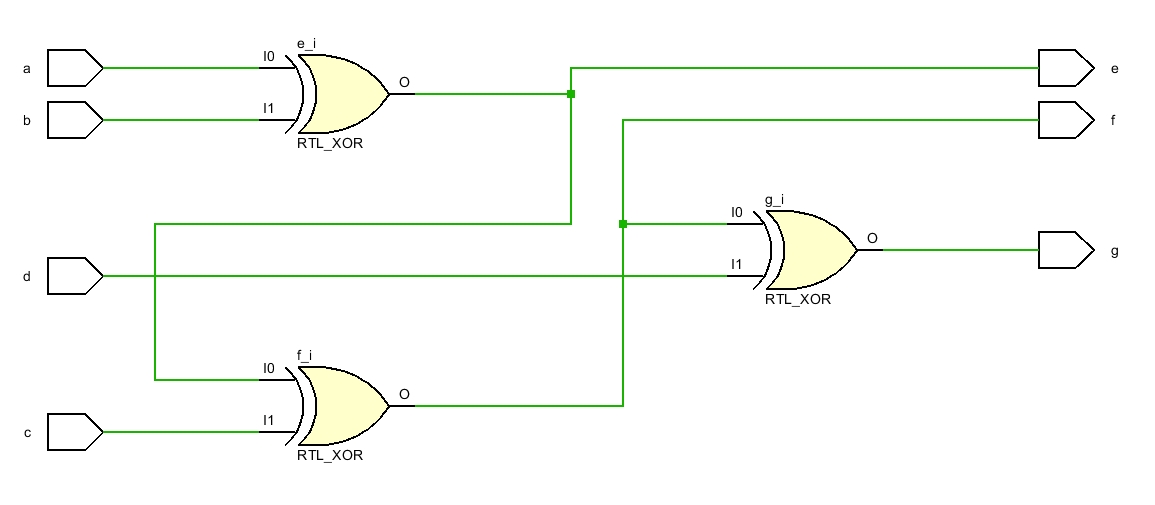
아래의 사진들은 2. ~ 5. 각각의 Schematic이다.



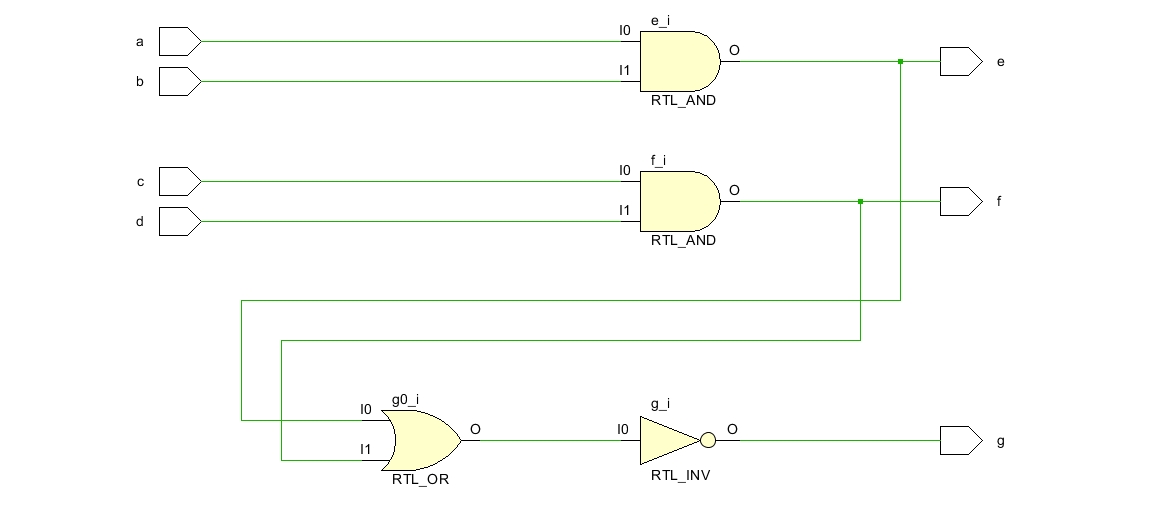
[4-input NAND gate의 schematic]



[4-input NOR gate의 schematic]



[4-input XOR gate의 schematic]



[4-input AOI gate의 schematic]

**7.**

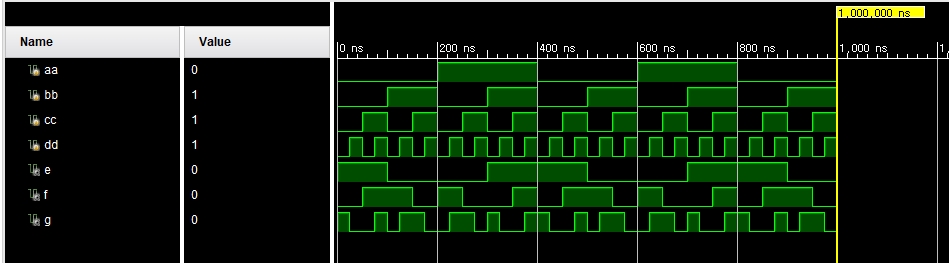
AND, OR게이트의 보수(Complement)인 NAND, NOR 게이트가 존재하듯이, XOR 게이트의 보수인 XNOR 게이트도 존재한다. Boolean 식에서는 또는 로 표기한다.

XNOR 게이트는 두 개의 입력이 같은 값이면 1을, 다른 값이면 0을 반환한다.

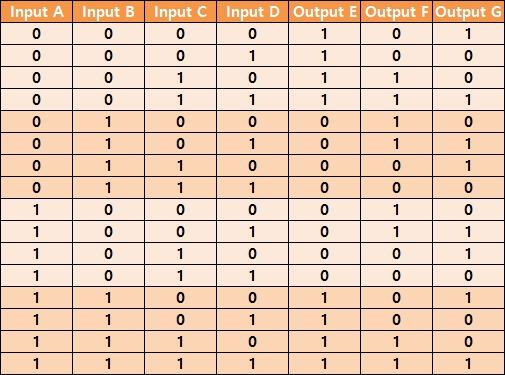
XNOR 게이트는 XOR 게이트와 마찬가지로 결합법칙이 성립하므로, 다중입력 XNOR 게게이트를 구성할 때 한 개의 XNOR 게이트로 구성하는 것과 여러 개의 XNOR 게이트로 구성하는 것 사이에 결과값의 차이는 없다. 따라서, 다중입력 XNOR 게이트의 결과값은 입력 중 1의 개수가 짝수일 때 1을, 홀수일 때 0을 가지게 된다.

위 실험과 같이 4-input XNOR gate를 구현해 simulation과 진리표, schematic을

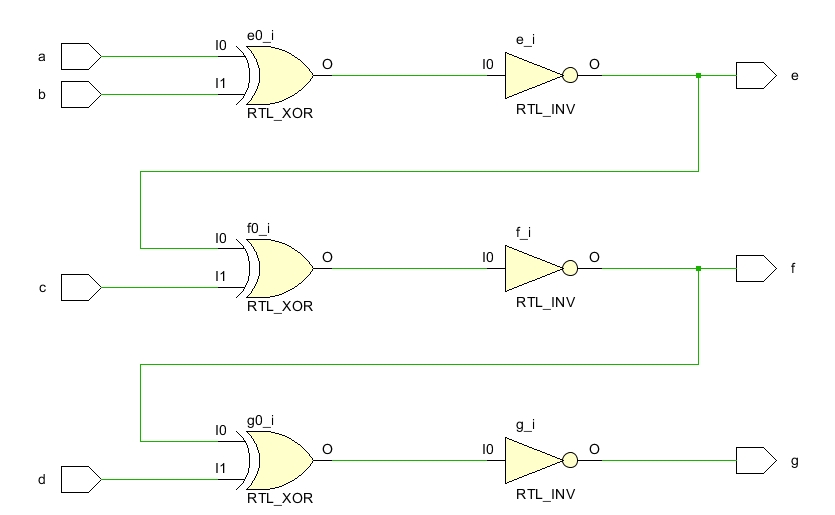
구하면 다음과 같다.



[4-input XNOR gate의 simulation 결과]



[4-input XNOR gate의 진리표]



[4-input XNOR gate의 schematic]