

컴퓨터공학실험2 5주차 결과보고서

전공: 컴퓨터공학과

학년: 2

학번: 20191559

이름: 강상원

1. De Morgan의 제 1,2 법칙의 simulation 결과 및 과정에 대해서 설명하시오.

(NAND,NOR과의 비교 포함)

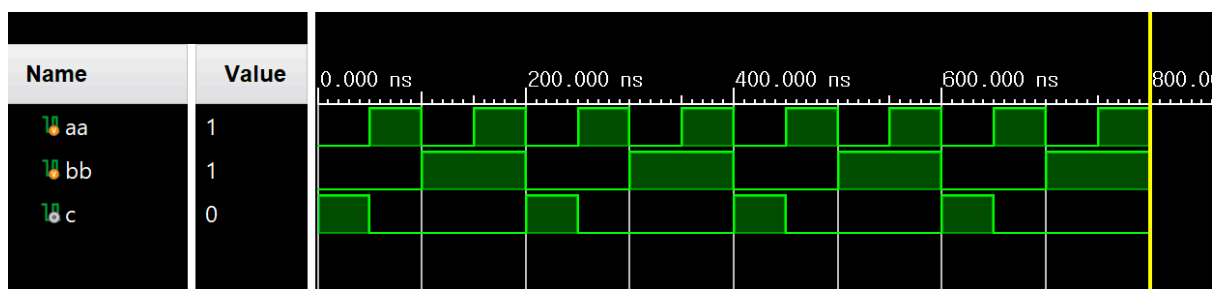
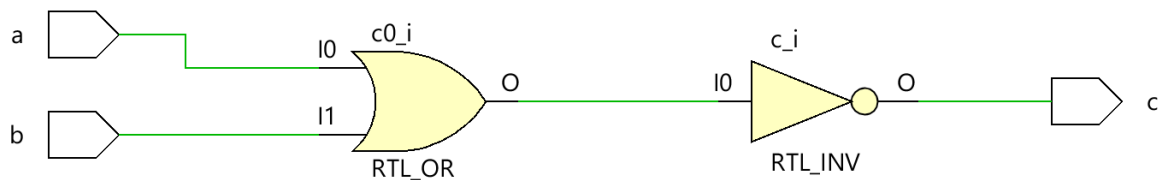
- 제1법칙

(A) 는 $assign\ c = \sim(a \mid b)$ 와 같이 나타낼 수 있다. Schematic과 Simulation 결과는 아래와 같다.

```

1 | `timescale 1ns / 1ps
2 |
3 | module De_Morgan_one_a(
4 |     input a, b,
5 |     output c
6 | );
7 |
8 |     assign c = ~(a | b);
9 |
10 | endmodule
11 |

```

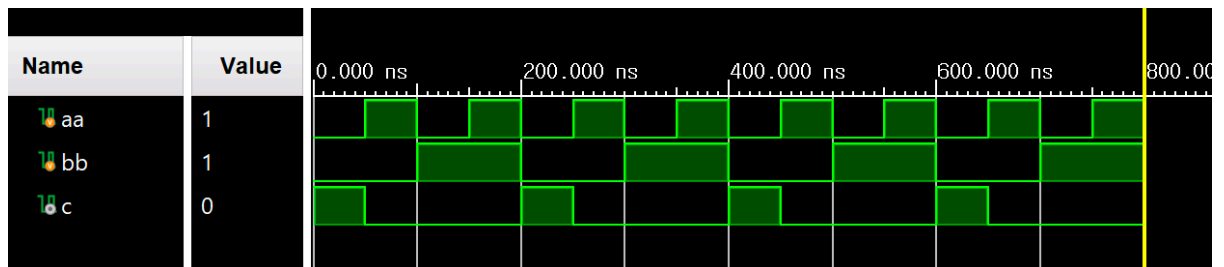
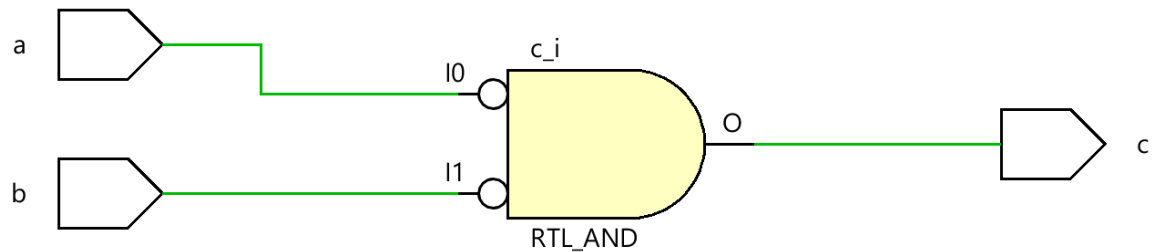


(B) 는 `assign c = (~a) & (~b)`와 같이 나타낼 수 있다. Schematic과 Simulation 결과는 아래와 같다.

```

1  `timescale 1ns / 1ps
2
3  module De_Morgan_one_b(
4      input a, b,
5      output c
6  );
7
8      assign c = (~a) & (~b);
9
10 endmodule
11

```



(A)와 (B)의 시뮬레이션 결과를 비교해 보면 최종 결과값이 같게 나타남을 알 수 있다. 드모르간의 제1법칙이 성립함을 Verilog 시뮬레이션을 통해 증명할 수 있었다.

※ 드모르간의 제1법칙 : $(A + B)' = A' \cdot B'$

위 (A)와 (B) 모두 NOR gate와 동일한 기능을 수행함을 알 수 있다. (OR gate의 부정형)

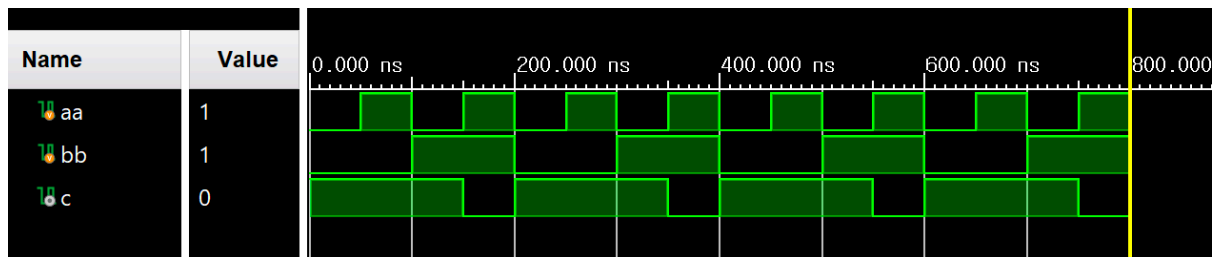
- 제2법칙

(A) 는 $\text{assign } c = \sim(a \& b)$ 와 같이 나타낼 수 있다. Schematic과 Simulation 결과는 아래와 같다.

```

1  `timescale 1ns / 1ps
2
3  module De_Morgan_two_a(
4      input a, b,
5      output c
6  );
7
8      assign c = ~(a & b);
9
10 endmodule
11

```

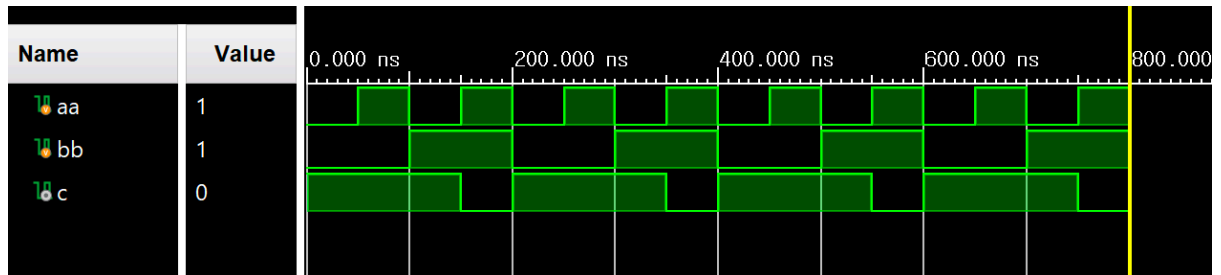
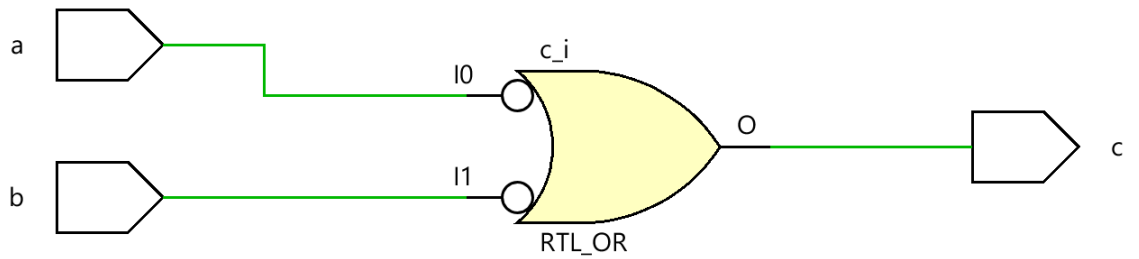


(B) 는 $\text{assign } c = (\sim a) \mid (\sim b)$ 와 같이 나타낼 수 있다. Schematic과 Simulation 결과는 아래와 같다.

```

1  `timescale 1ns / 1ps
2
3  module De_Morgan_two_b(
4      input a, b,
5      output c
6  );
7
8      assign c = (~a) | (~b);
9
10 endmodule
11

```



(A)와 (B)의 시뮬레이션 결과를 비교해 보면 최종 결과값이 같게 나타남을 알 수 있다. 드 모르간의 제2법칙이 성립함을 Verilog 시뮬레이션을 통해 증명할 수 있었다.

※ 드모르간의 제2법칙 : $(A \cdot B)' = A' + B'$

위 (A)와 (B) 모두 NAND gate와 동일한 기능을 수행함을 알 수 있다. (OR gate의 부정형)

2. $(A'+B') \cdot C' = ((A \cdot B) + C)'$ 의 simulation 결과 및 과정에 대해서 설명하시오.

[+ 및 * 위치 바꾼 모양도 수행]

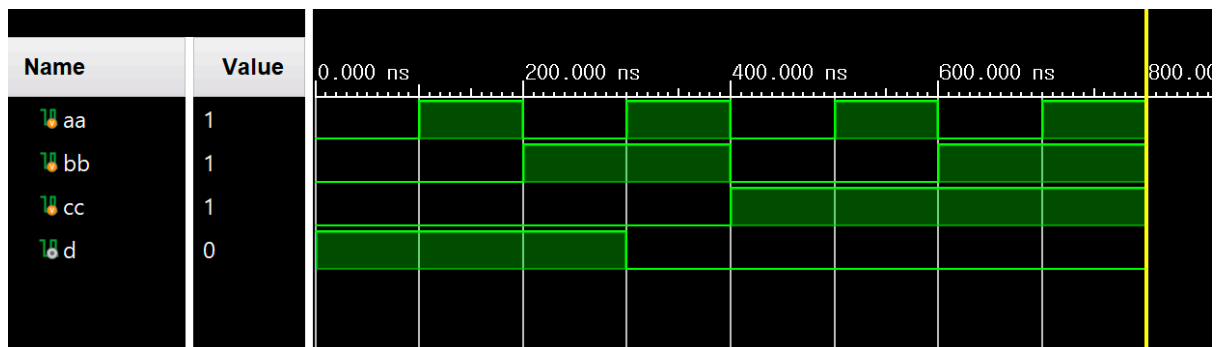
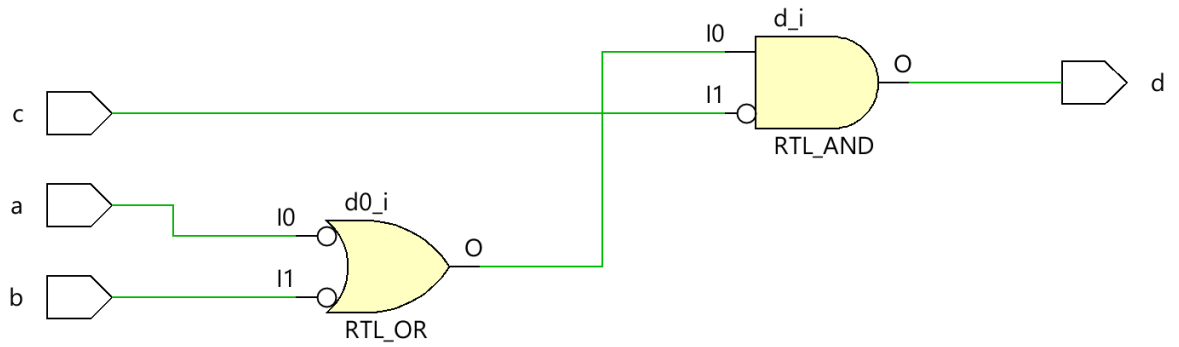
- $(A'+B') \cdot C'$

assign d = ((~a) | (~b)) & (~c)와 같이 나타낼 수 있다. Schematic과 Simulation 결과는 아래와 같다.

```

1 | `timescale 1ns / 1ps
2 |
3 | module Boolean_Func_one_a(
4 |     input a, b, c,
5 |     output d
6 | );
7 |
8 |     assign d = ((~a) | (~b)) & (~c);
9 |
10 | endmodule
11 |

```



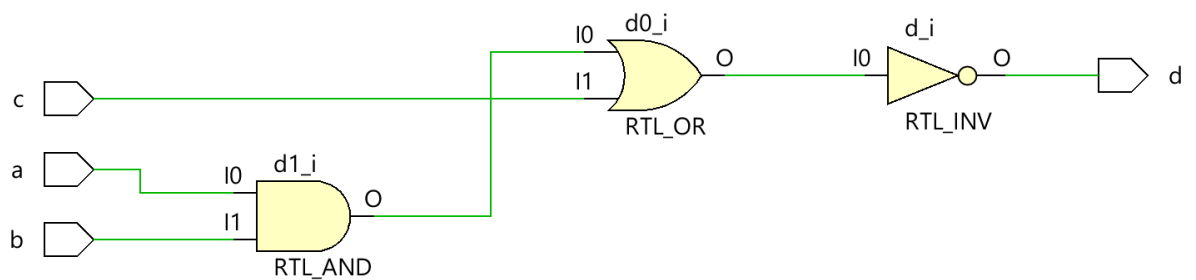
- $((A*B)+C)'$

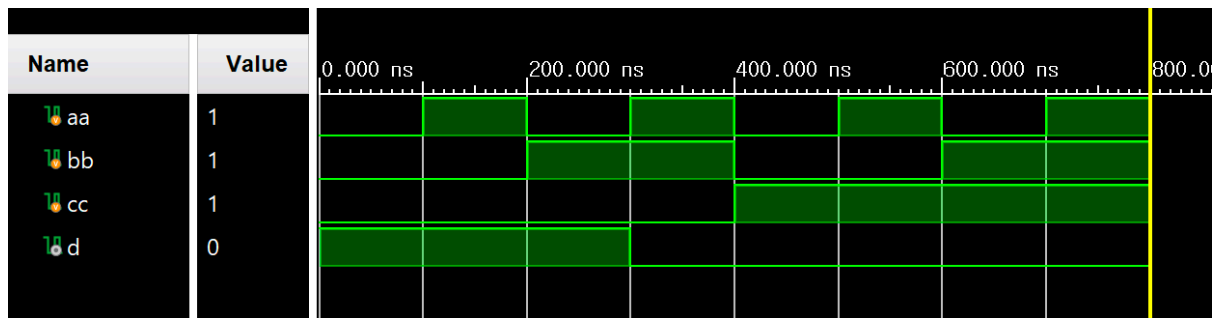
assign $d = \sim((a \& b) | c)$ 와 같이 나타낼 수 있다. Schematic과 Simulation 결과는 아래와 같다.

```

1      `timescale 1ns / 1ps
2
3      module Boolean_Func_one_b(
4          input a, b, c,
5          output d
6      );
7
8      assign d = ~((a & b) | c);
9
10     endmodule
11

```





$(A'+B')*C'$ 와 $((A*B)+C)'$ 의 시뮬레이션 결과를 비교해 보면 최종 결과값이 같게 나타남을 알 수 있다. 드모르간의 법칙을 이용하면 두 식이 같음을 증명할 수 있다.

$$\begin{aligned}
 \text{증)} \quad (A'+B')*C' &= (A*B)' * C' && (\text{드모르간 제1법칙}) \\
 &= ((A*B) + C)' && (\text{드모르간 제2법칙})
 \end{aligned}$$

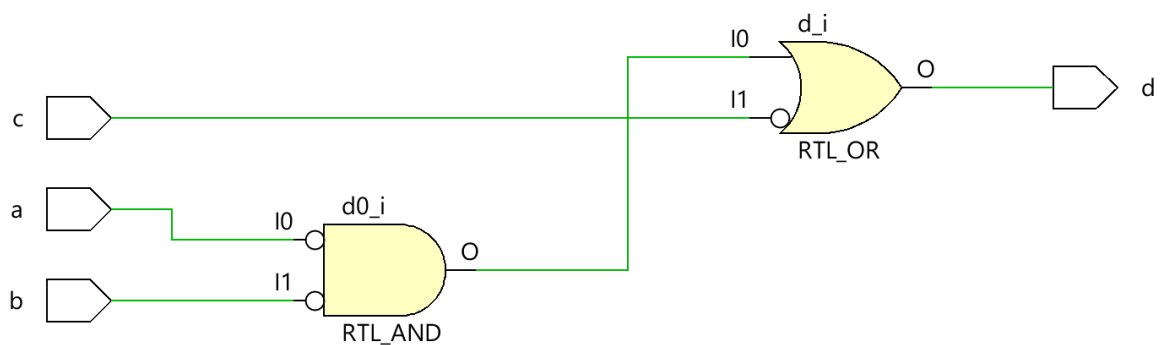
- $(A'*B')+C'$

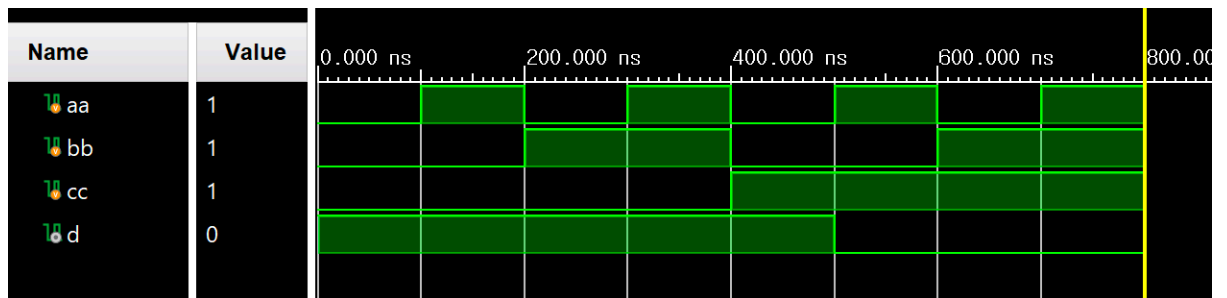
assign d = ((~a) & (~b)) | (~c)와 같이 나타낼 수 있다. Schematic과 Simulation 결과는 아래와 같다.

```

1      `timescale 1ns / 1ps
2
3      module Boolean_Func_two_a(
4          input a, b, c,
5          output d
6      );
7
8          assign d = ((~a) & (~b)) | (~c);
9
10     endmodule
11

```





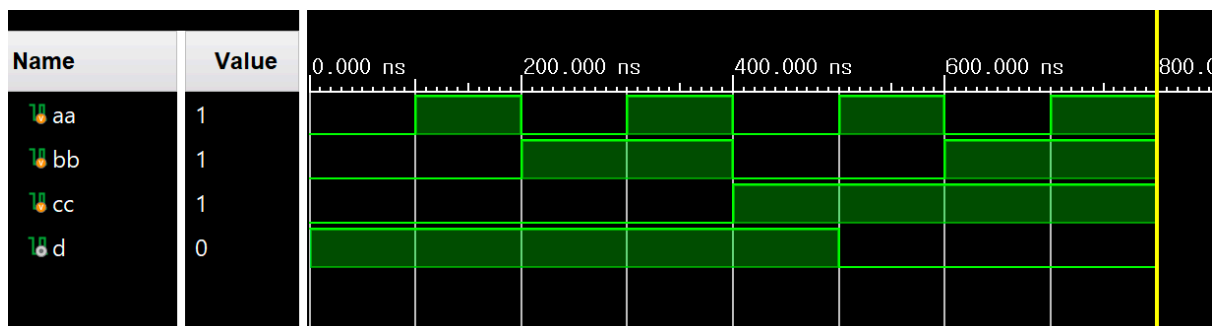
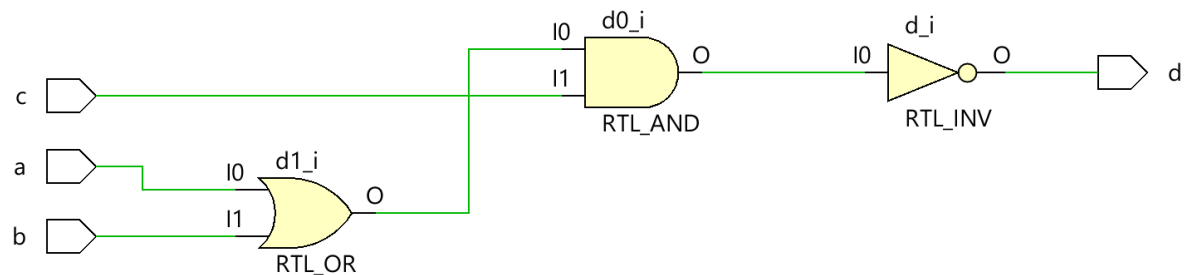
- $((A+B)*C)'$

assign d = ~((a | b) & c)와 같이 나타낼 수 있다. Schematic과 Simulation 결과는 아래와 같다.

```

1  `timescale 1ns / 1ps
2
3  module Boolean_Func_two_b(
4      input a, b, c,
5      output d
6  );
7
8      assign d = ~((a | b) & c);
9
10 endmodule
11

```



$(A'B') + C'$ 와 $((A+B) * C)'$ 의 시뮬레이션 결과를 비교해 보면 최종 결과값이 같게 나타남을 알 수 있다. 드모르간의 법칙을 이용하면 두 식이 같음을 증명할 수 있다.

$$\begin{aligned} \text{증) } (A'B') + C' &= (A+B)' + C' && (\text{드모르간 제2법칙}) \\ &= ((A+B) * C)' && (\text{드모르간 제1법칙}) \end{aligned}$$

3. 1Bit 비교기의 simulation 결과 및 과정에 대해서 설명하시오.

(2 input, 4 output)[진리표 작성]

$A=B$ 출력값은 두 입력값이 같을 때 1을 출력하므로 다음과 같이 표현할 수 있다. $(A \oplus B)'$

$A \neq B$ 출력값은 두 입력값이 다를 때 1을 출력하므로 다음과 같이 표현할 수 있다. $A \oplus B$

$A > B$ 출력값은 $A=1, B=0$ 일 때 1을 출력하므로 다음과 같이 표현할 수 있다. $A * B'$

$A < B$ 출력값은 $A=0, B=1$ 일 때 1을 출력하므로 다음과 같이 표현할 수 있다. $A' * B$

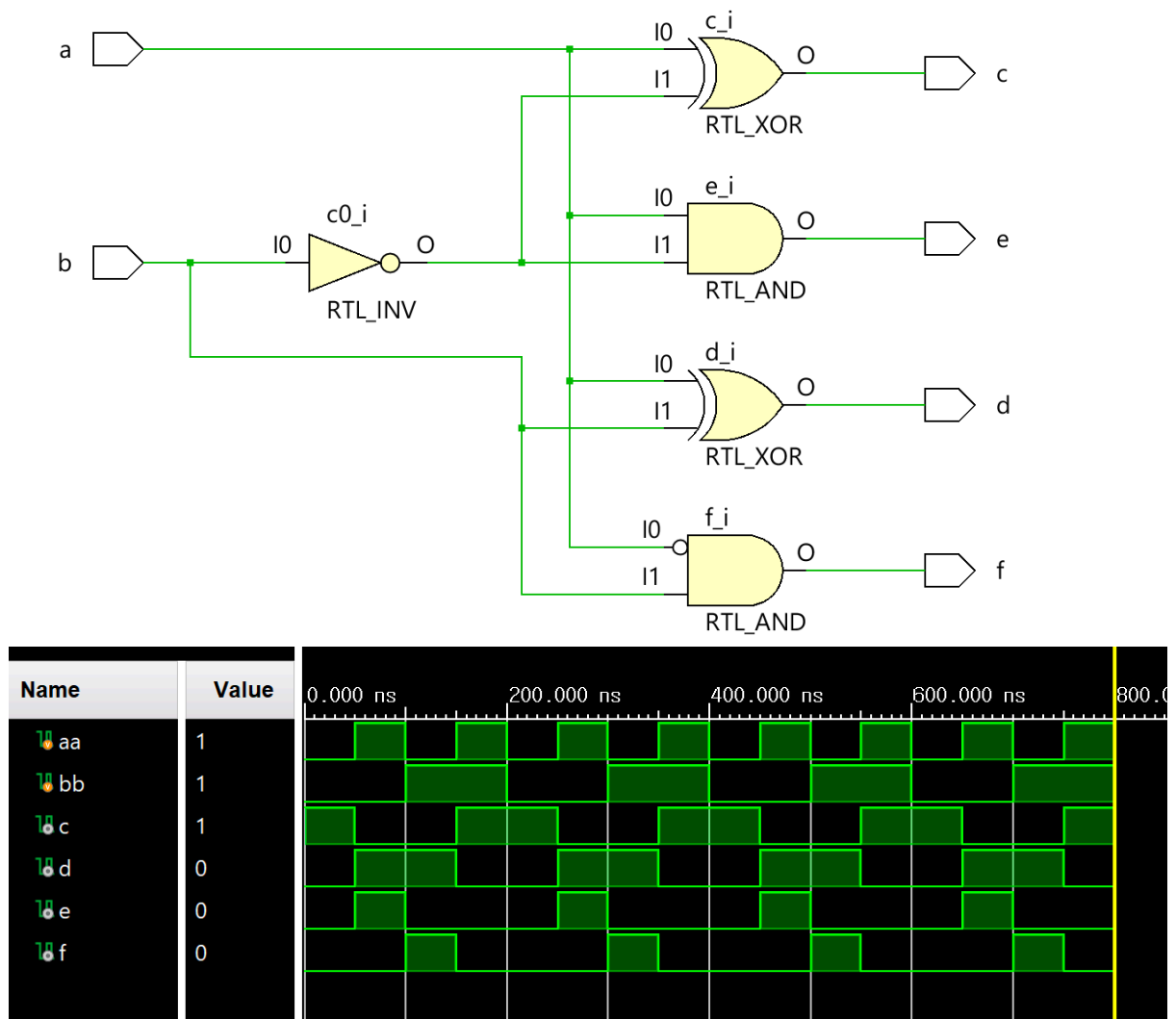
`assign c = a ^ (~b);`, `assign d = a ^ b;`, `assign e = a & (~b);`, `assign f = (~a) & b;`와 같이 나타낼 수 있다.

여기서 c, d, e, f는 각각 $A=B, A \neq B, A > B, A < B$ 의 결과값을 의미한다. Schematic과 Simulation 결과는 아래와 같다.

```

1 | `timescale 1ns / 1ps
2 |
3 | module One_Bit_Comp(
4 |     input a, b,
5 |     output c, d, e, f
6 | );
7 |
8 |     assign c = a ^ (~b);
9 |     assign d = a ^ b;
10 |    assign e = a & (~b);
11 |    assign f = (~a) & b;
12 |
13 | endmodule
14 |

```

$A=B$ 출력값은 두 입력값이 같을 때 1을 출력하므로 다음과 같이 표현할 수 있다. $(A \oplus B)'$

$A \neq B$ 출력값은 두 입력값이 다를 때 1을 출력하므로 다음과 같이 표현할 수 있다. $A \oplus B$

$A > B$ 출력값은 $A=1, B=0$ 일 때 1을 출력하므로 다음과 같이 표현할 수 있다. $A * B'$

$A < B$ 출력값은 $A=0, B=1$ 일 때 1을 출력하므로 다음과 같이 표현할 수 있다. $A' * B$

아래는 1Bit 비교기의 진리표이다.

Input a	Input 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