4주차 결과보고서

|  |
| --- |
| - NAND/NOR/XOR Gate 동작의 이해  - 4-input NAND gate  - 4-input NOR gate  - 4-input XOR gate  - 4-input AOI gate |

20141196 김성희

**1. NAND/NOR/XOR Gate 동작의 이해**

**1-1. NAND**

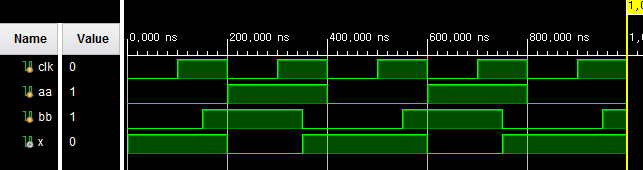


<truth table>

|  |  |  |
| --- | --- | --- |
| A | B | Output |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

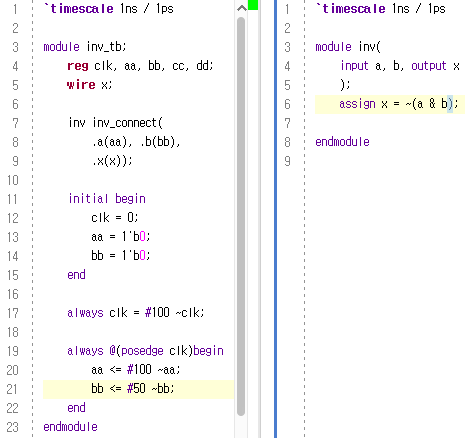
\* AND의 결과에 NOT을 취한 값과 동일하다.

<simulation>



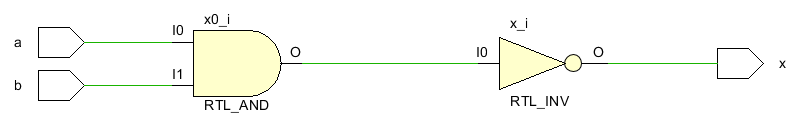
Aa와 bb는 입력, x는 출력값이다.

<code>



bb**;**

<schematic>



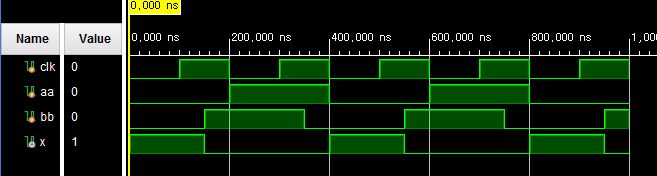
**1-2. NOR**



<truth table>

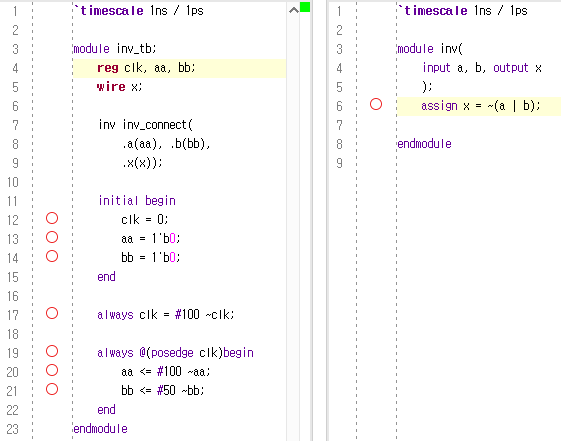
|  |  |  |
| --- | --- | --- |
| A | B | Output |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

<simulation>

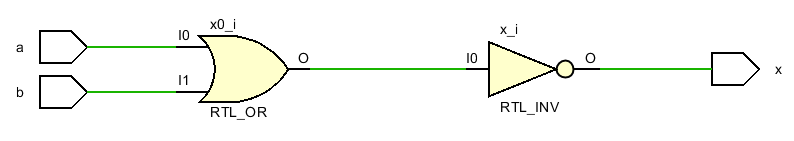
\* aa와 bb는 입력, x는 출력 값이다.

\* OR의 결과에 NOT을 취한 값과 동일하다.

<code>



<schematic>



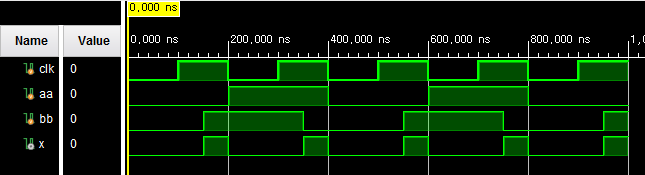
**1-3. XOR**

**** ****

<truth table>

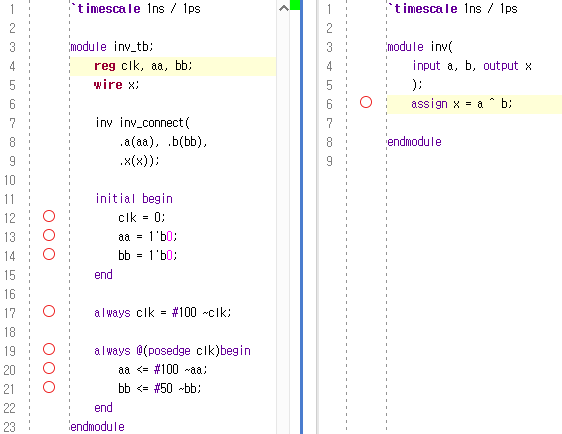
|  |  |  |
| --- | --- | --- |
| **A** | **B** | **Output** |
| **0** | **0** | **0** |
| **0** | **1** | **1** |
| **1** | **0** | **1** |
| **1** | **1** | **0** |

<simulation>

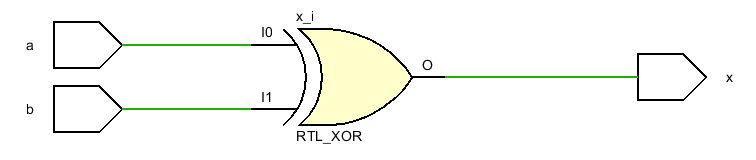


\* aa와 bb는 입력, x는 출력 값이다.

\* 입력 값이 서로 다른 경우에만 1의 값을 가진다.<code>



<schematic>



**2. 4-input NAND Gate 구현 (진리표, simulation, schematic)**



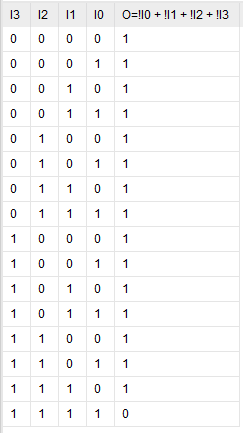
(A)



(B)

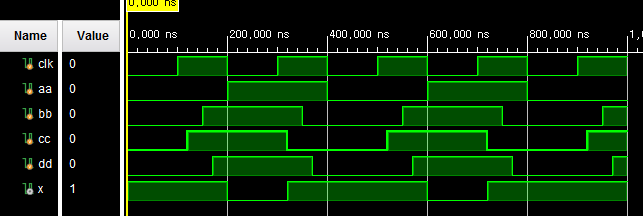
A에서 E = ~(A & B & C & D)이고, B에서 G = ~(~(~(A & B) & C) & D)이다. 따라서 최종 결과값이 다르다. (최종 결과가 같았던 다중입력 AND Gate와의 차이점)

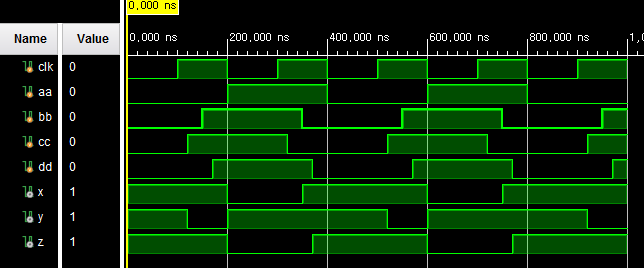
<truth table> 상/하 : A/B



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A (In)** | **B (In)** | **C (In)** | **D (In)** | **E (Out)** | **F (Out)** | **G (Out)** |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

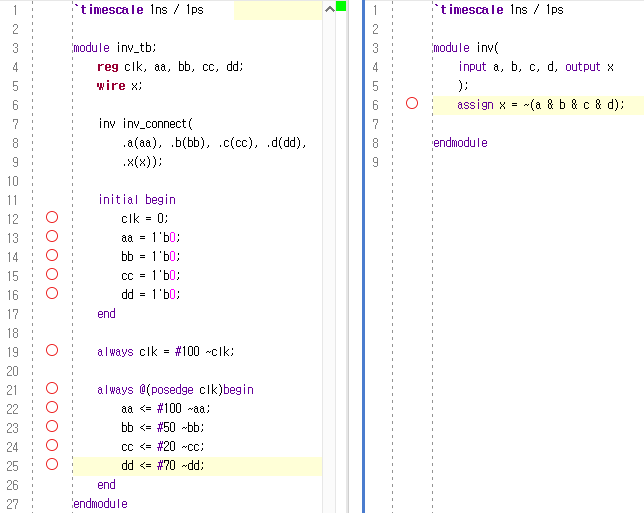
<simulation> 상/하 : A/B

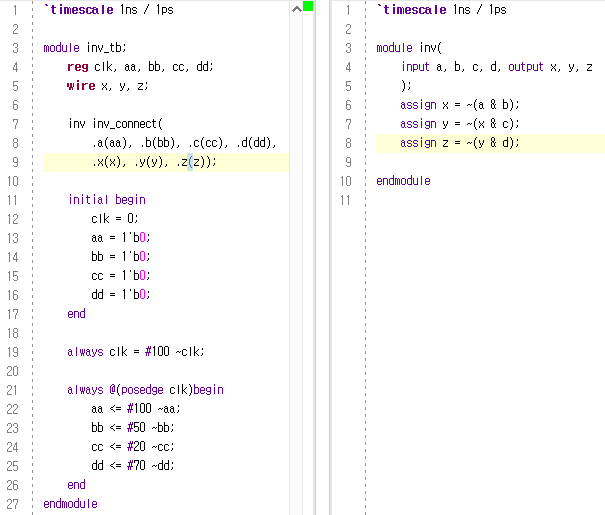




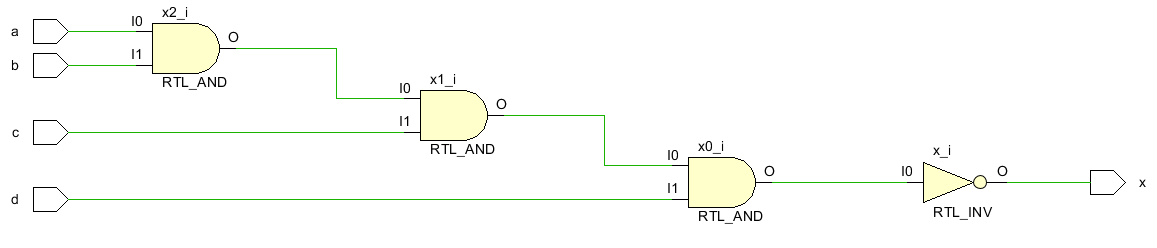
\* 앞에서 언급했듯 x(위 simulation)과 z 값이 다르다.

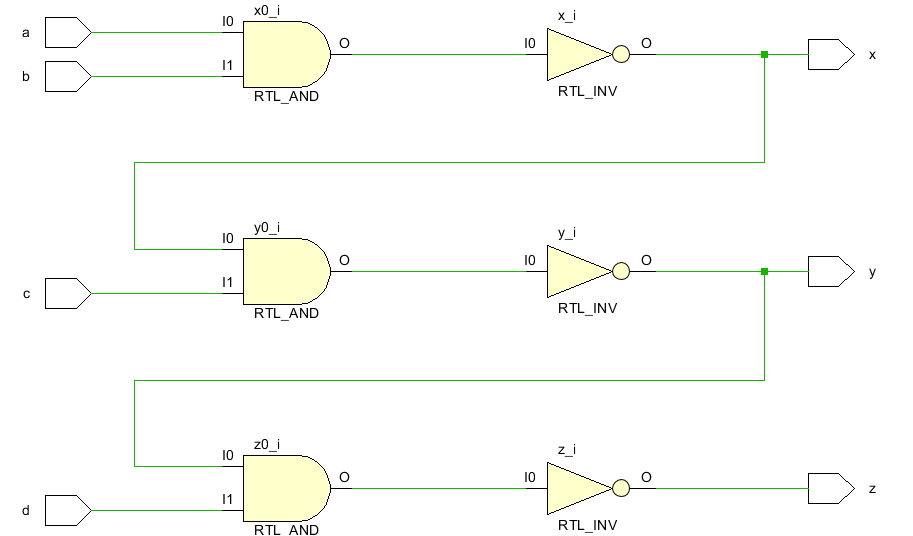
<code> 상/하 : A/B





<schematic> 상/하 : A/B





**3. 4-input NOR Gate 구현 (진리표, simulation, schematic)**



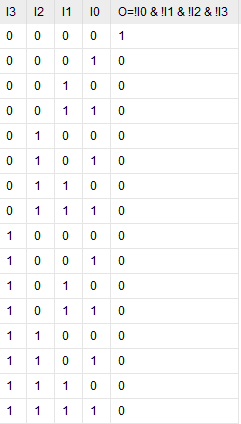
(A)



(B)

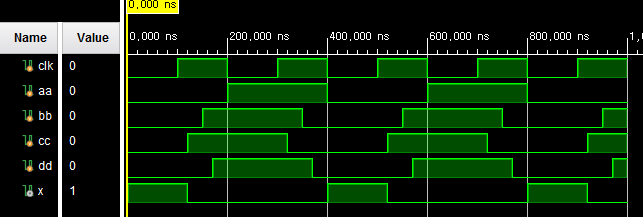
A에서 E=~(A | B | C | D), B에서 G=~(D | ~(C | ~( A | B)))이다. 따라서 최종 결과값이 다르다. (최종 결과가 같았던 다중입력 AND Gate와의 차이점)

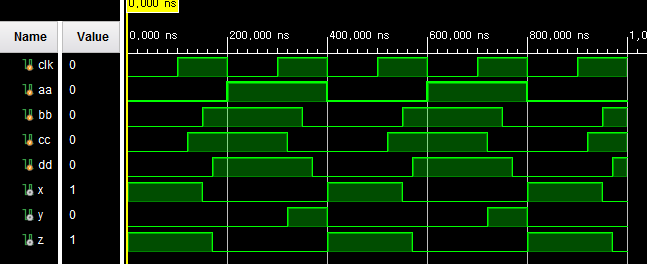
<truth table> 상/하 A/B



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A (In)** | **B (In)** | **C (In)** | **D (In)** | **E (Out)** | **F (Out)** | **G (Out)** |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 |

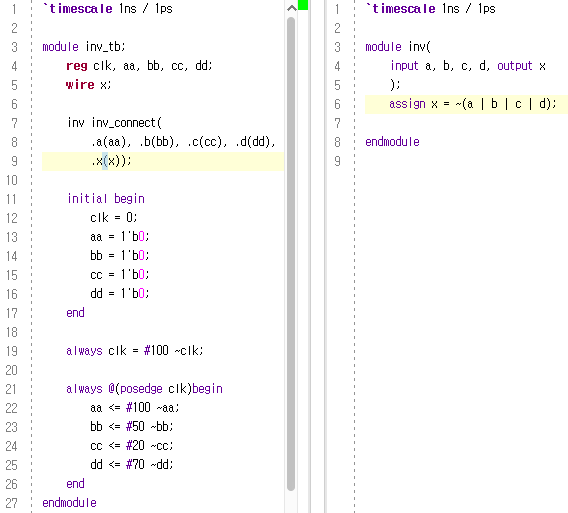
<simulation> 상/하 A/B

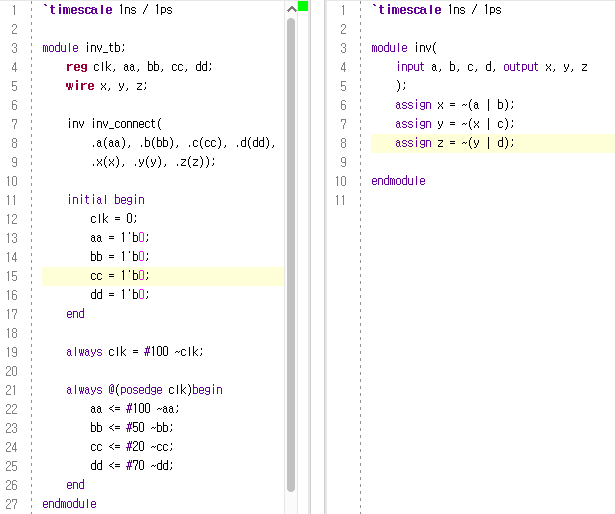




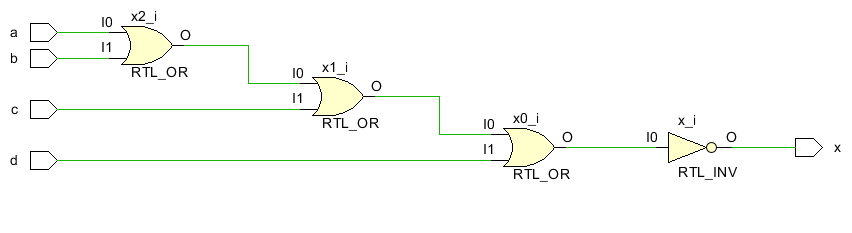
\* 앞에서 언급했듯 x(위 simulation)과 z 값이 다르다

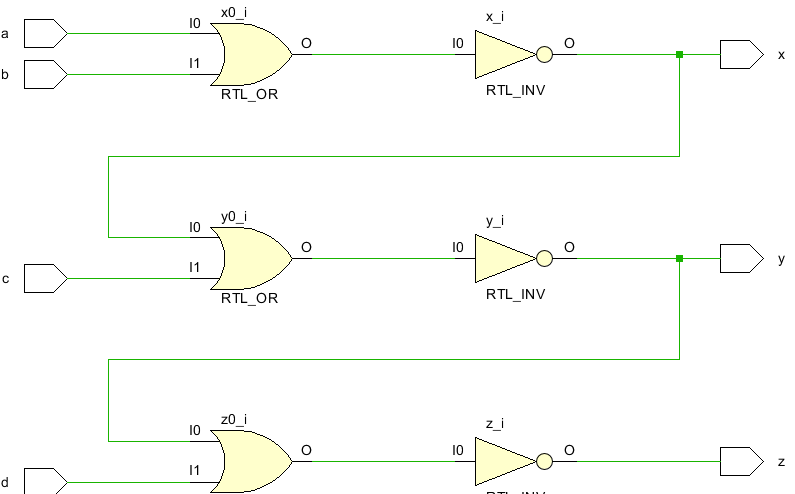
<code> 상/하 A/B





<schematic> 상/하 A/B





**4. 4-input XOR Gate 구현 (진리표, simulation, schematic)**

(A)





(B)

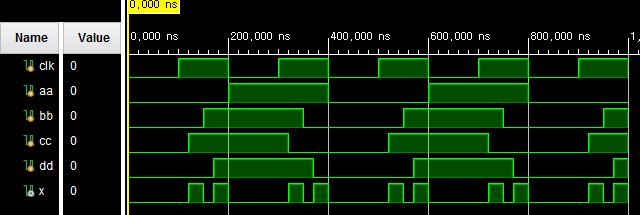
Xor는 교환법칙이 성립한다. 따라서 최종 결과값도 같다.

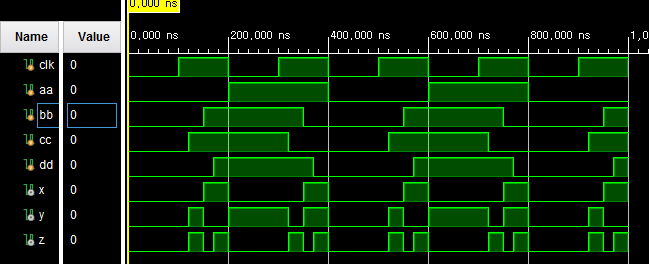
<truth table> 상/하 A/B

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A (In)** | **B (In)** | **C (In)** | **D (In)** | **E (Out)** |
| 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 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A (In)** | **B (In)** | **C (In)** | **D (In)** | **E (Out)** | **F (Out)** | **G (Out)** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 |

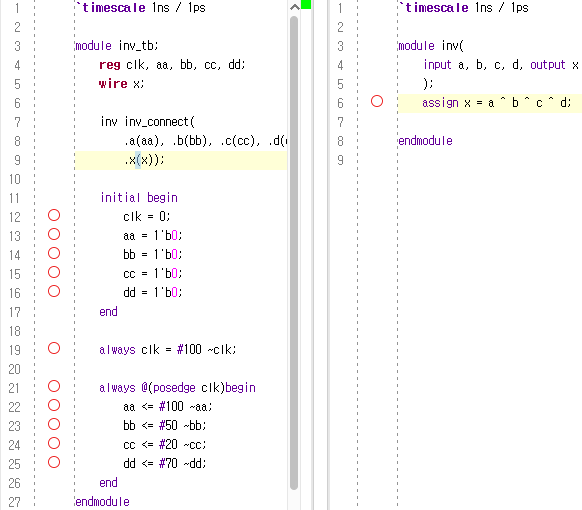
<simulation> 상/하 A/B

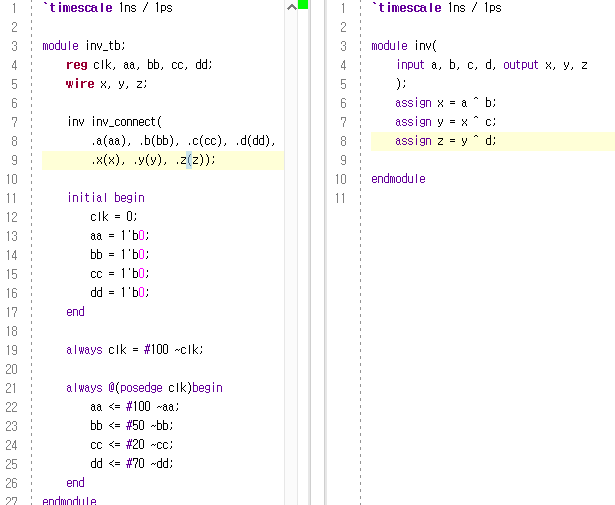




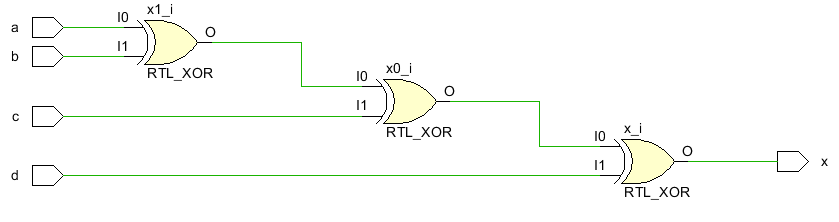
\* 최종 결과값이 같다. (교환법칙 성립)

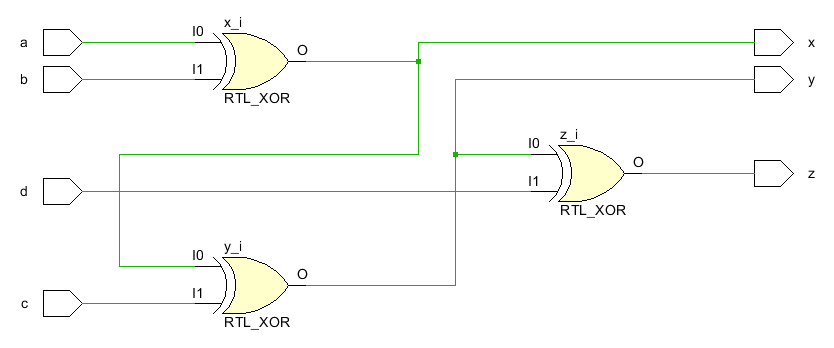
<code> 상/하 A/B





<schematic> 상/하 A/B



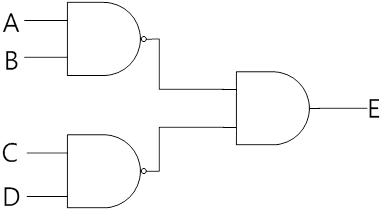


**5. 4-input AOI Gate 구현 (진리표, simulation, schematic)**



**(가)**

E = ~(AB + CD) = (~(AB) )(~(CD) ) (드모르간 법칙)



**(나)**

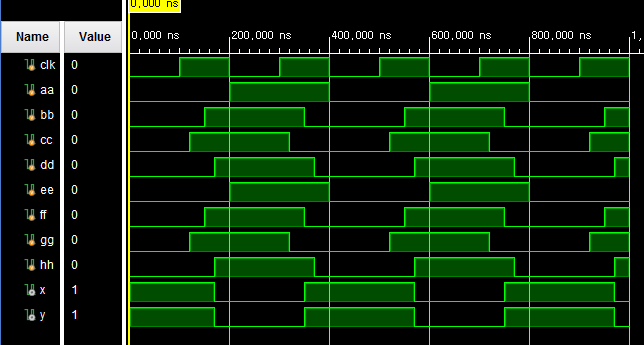
- AND와 OR의 조합은 NAND 게이트로 바꿔서 쓸 수 있다.

- CMOS 기술을 쓴다면 아래 방식이 트랜지스터가 2개 덜 들어간다.

<truth table>

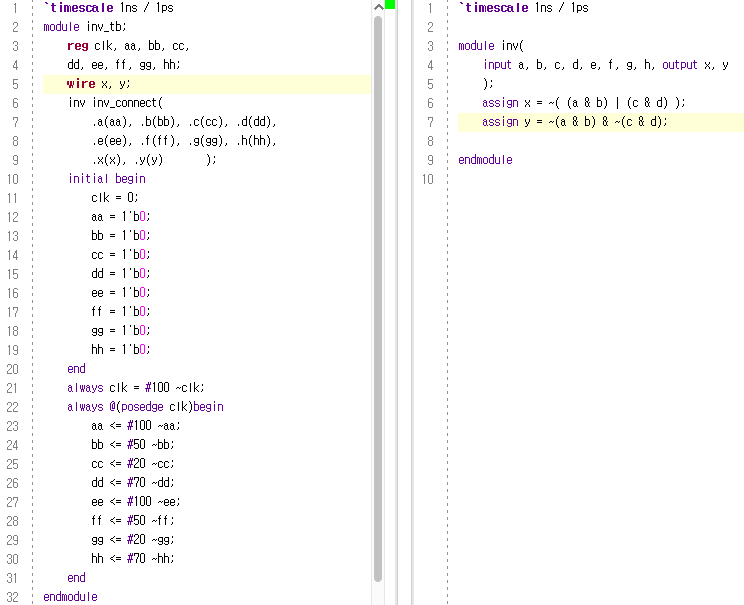
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A (In)** | **B (In)** | **C (In)** | **D (In)** | **E (Out)** |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

<simulation>



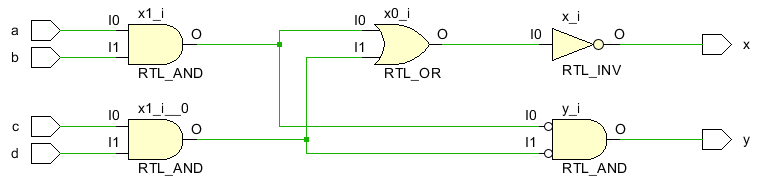
\* a~d, x는 각각 (가)의 입,출력이다. e~h, y는 각각 (나)의 입,출력이다.

<code>



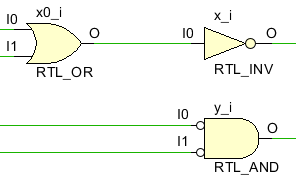
\* a~d, x는 각각 (가)의 입,출력이다. e~h, y는 각각 (나)의 입,출력이다.

<schematic>



X는 (가)의 결과, y는 (나)의 결과.

**6. 결과 검토 및 논의사항 + 추가 이론**



드모르간 법칙,

위 그림은 ~(A+B) = (~A)x(~B)의 회로 버전이다.

~(AxB) = (~A)+(~B)의 경우는 위 회로에서 AND, OR게이트를 서로 바꾼 것을 생각하면 된다.