5주차 결과보고서

전공: 생명과학과 학년: 4학년 학번: 20182186 이름: 김승원

**1. 실험목적**

드모르간의 정리를 확인하고 Boolean function을 확인한다. 이후 Verilog를 사용해서 De Morgan의 정리와 Boolean function의 동작을 구현해준다. 입력신호를 생성하여 simulation을 돌려서 구현한 결과를 확인해보고, 최종적으로 Verilog로 구현한 회로의 동작을 FPGA을 통하여 확인한다.

**2.**

**2-1) 드모르간의 제 1법칙**

**<design source code>**

`timescale 1ns / 1ps

module dm1(

input a,b,

output x,y

);

assign x=~(a|b), y=(~a)&(~b);

endmodule

**<simulation source code>**

`timescale 1ns / 1ps

module tb\_dm1;

reg A, B;

wire X, Y;

dm1 dm1gate(.a(A), .b(B), .x(X), .y(Y));

initial begin

A= 1'b0;

B= 1'b0;

end

always begin

A= #10 ~A;

B= #15 ~B;

end

initial begin

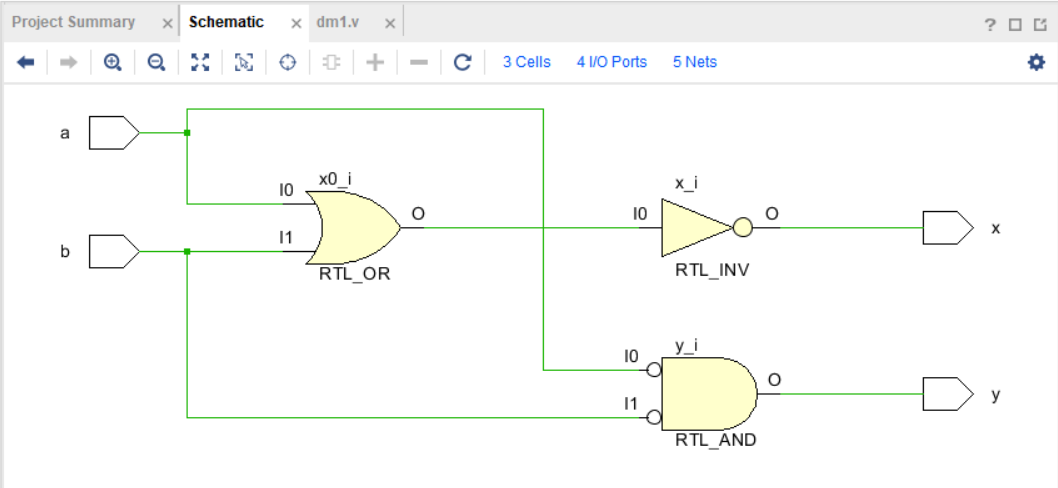
#1000

$finish;

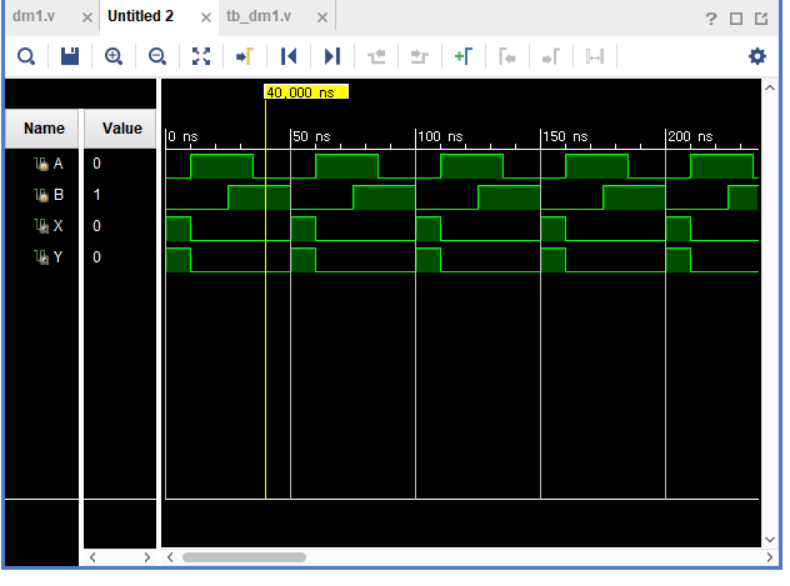
end

endmodule

**<드모르간 제 1법칙 schematic 결과>**



**<드 모르간 제 1법칙 simulation 결과>**



De Morgan의 제 1법칙을 Vivado에서 simulation한 결과이다. X를 ~(A+B)로 Y를 ~A&~B로 동작시켰다. 결과적으로 ~(A+B)와 ~A&~B는 같은 결과를 나타내는 것으로 확인됐다.

**2-2) 드모르간 제 2법칙**

**<design source code>**

`timescale 1ns / 1ps

module dm1(

input a,b,

output x,y

);

assign x=~(a&b), y=(~a)|(~b);

endmodule

**<Simulation source code>**

`timescale 1ns / 1ps

module tb\_dm2;

reg A, B;

wire X, Y;

dm2 dm2gate(.a(A), .b(B), .x(X), .y(Y));

initial begin

A= 1'b0;

B= 1'b0;

end

always begin

A= #10 ~A;

B= #15 ~B;

end

initial begin

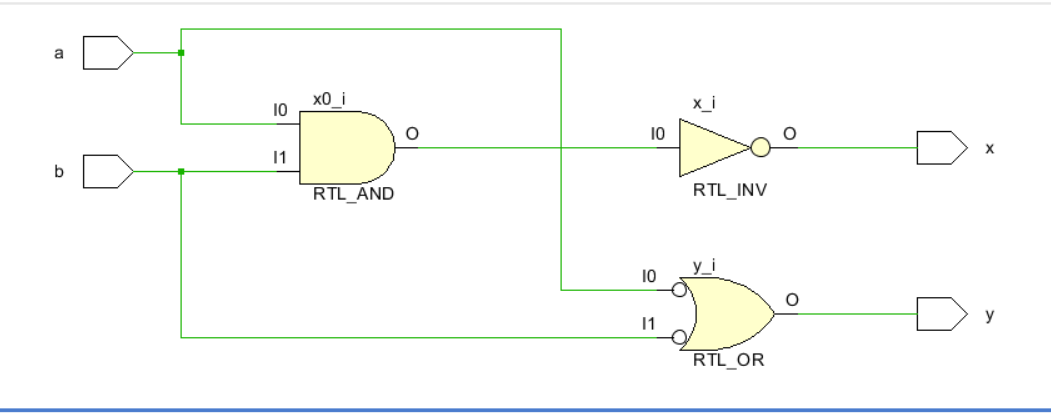
#1000

$finish;

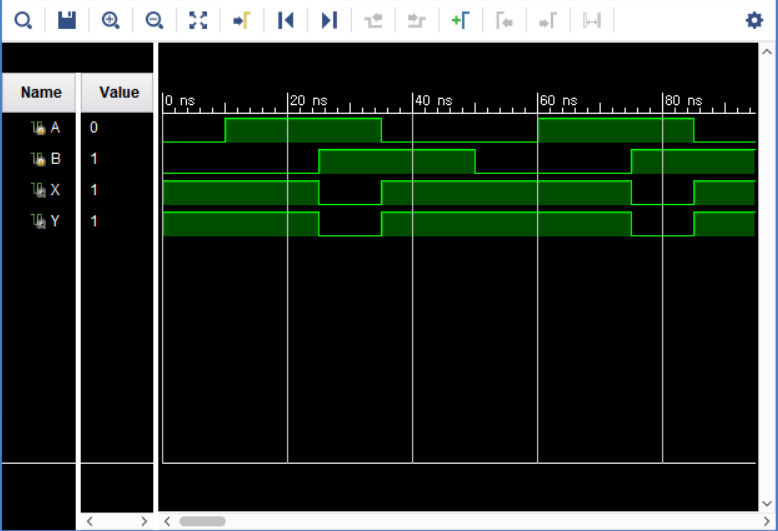
end

endmodule

**<드모르간 제 2법칙 schematic 결과>**



**<드모르간 제 2법칙 simulation 결과>**



De Morgan의 제 2법칙을 Vivado에서 simulation한 결과이다. X를 ~(A&B)로 Y를 (~A)|(~B)로 동작시켰다. 결과적으로 X의 출력과 Y출력을 확인해봤고, ~(A+B)와 ~A&~B는 완전히 같은 결과를 나타내는 것으로 확인됐다.

3.

**3-1) Boolean function 1 실습**

**<Design source code>**

`timescale 1ns / 1ps

module bf1(

input a,b,c,

output x,y

);

assign x=(~a|~b)&~c, y=~((a&b)|c);

endmodule

**<Simulation source code>**

`timescale 1ns / 1ps

module tb\_bf1;

reg A, B, C;

wire X, Y;

bf1 bf1gate(.a(A), .b(B), .c(C), .x(X), .y(Y));

initial begin

A= 1'b0;

B= 1'b0;

C= 1'b0;

end

always begin

A= #10 ~A;

B= #20 ~B;

C= #30 ~C;

end

initial begin

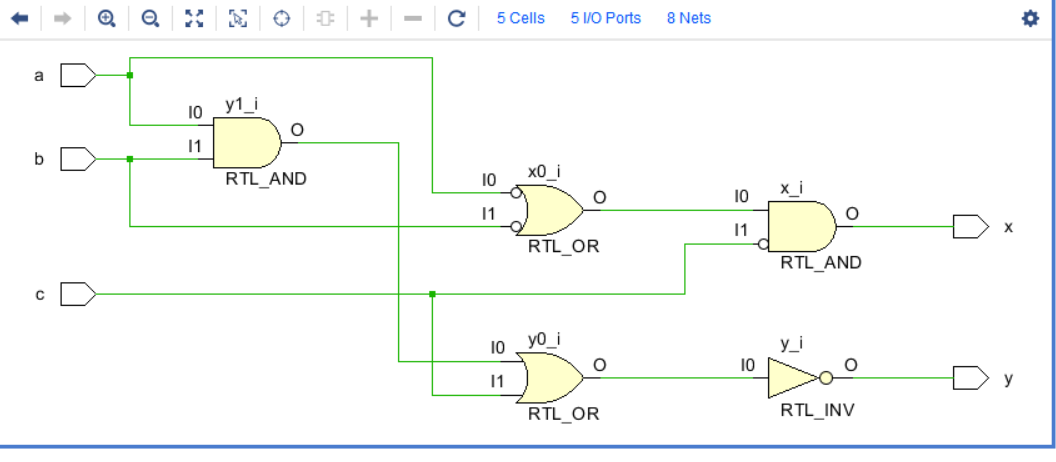
#1000

$finish;

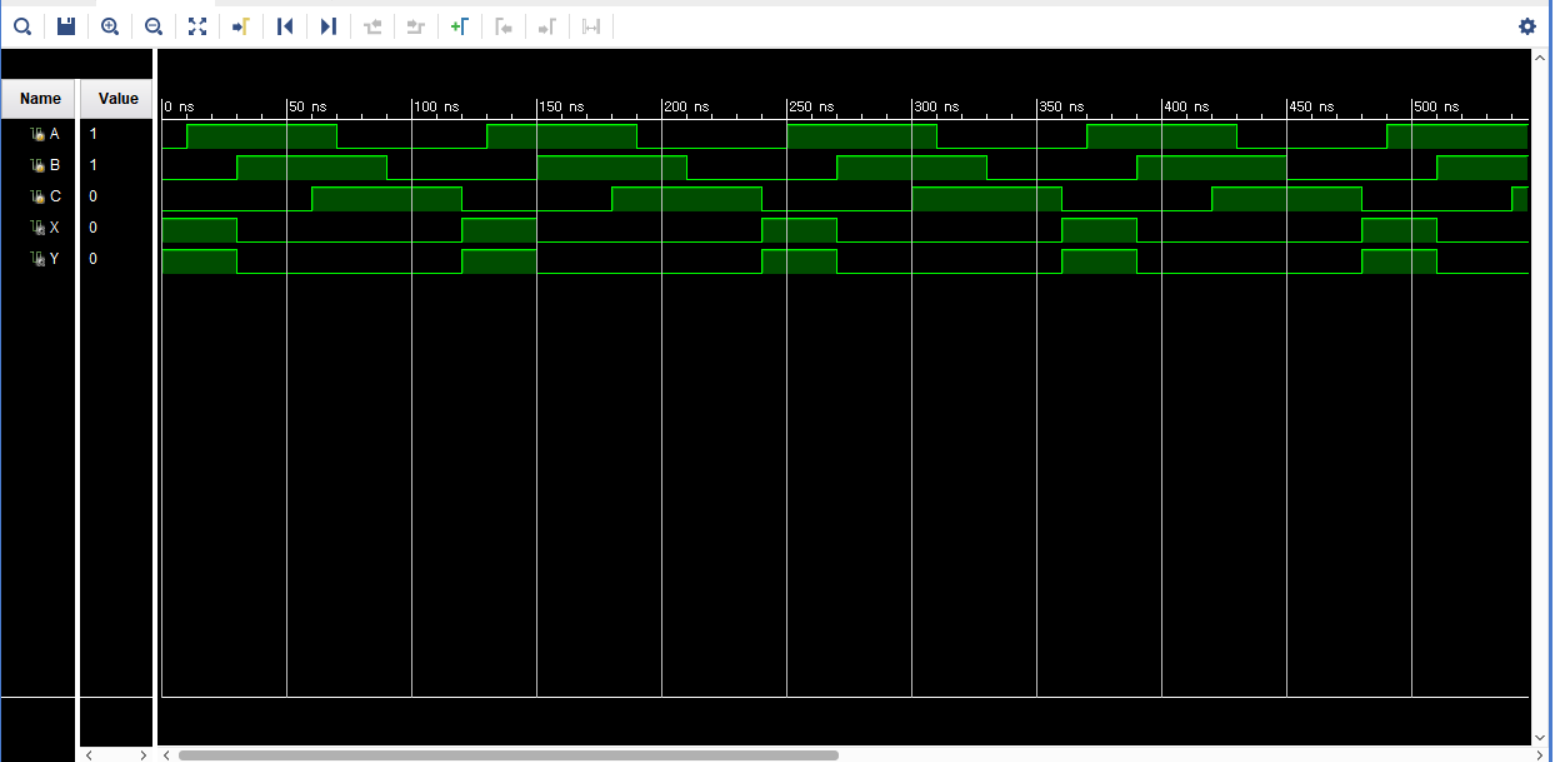
end

endmodule

**<Boolean function 1 schematic 결과>**



**<Boolean function 1 simulation 결과>**



Boolean function 1을 Vivado에서 simulation한 결과이다. X를 (~A|~B)&~C로 Y를 ~((A&B)|C)로 동작시켰다. 결과적으로 X의 출력과 Y출력을 확인해봤고, (~A|~B)&~C 와 ~((A&B)|C)는 완전히 같은 결과를 나타내는 것으로 확인됐다.

**3-2) Boolean function 2 실습**

**<Design source code>**

`timescale 1ns / 1ps

module bf2(

input a,b,c,

output x,y

);

assign x=(~a&~b)|~c, y=~((a|b)&c);

endmodule

**<Simulation source code>**

`timescale 1ns / 1ps

module tb\_bf2;

reg A, B, C;

wire X, Y;

bf2 bf2gate(.a(A), .b(B), .c(C), .x(X), .y(Y));

initial begin

A= 1'b0;

B= 1'b0;

C= 1'b0;

end

always begin

A= #10 ~A;

B= #20 ~B;

C= #30 ~C;

end

initial begin

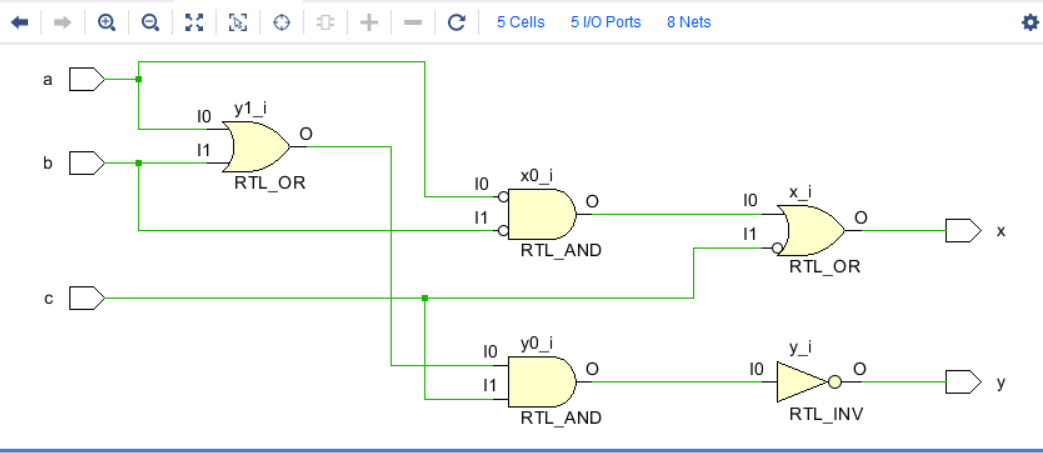
#1000

$finish;

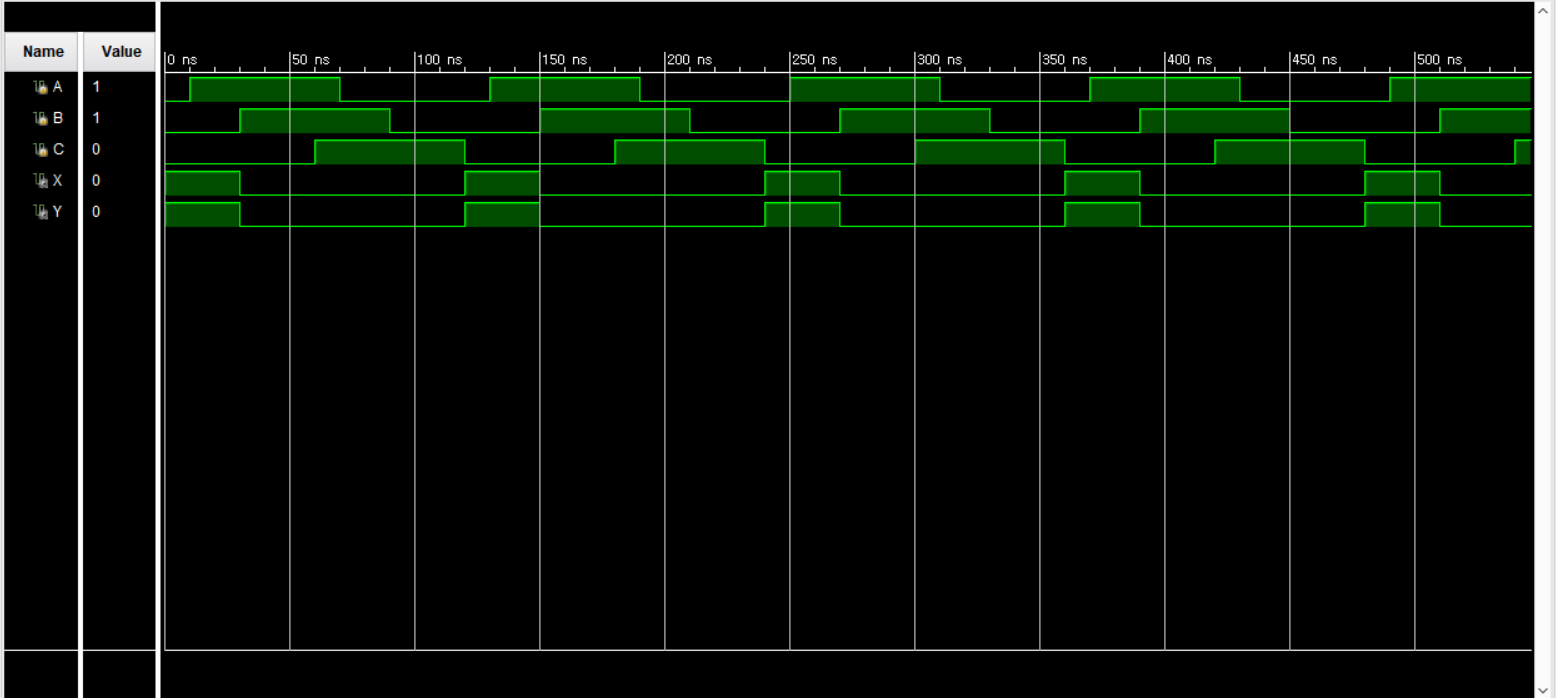
end

endmodule

**<Boolean function 2 schematic 결과>**



**<Boolean function 2 simulation 결과>**



Boolean function 2를 Vivado에서 simulation한 결과이다. X를 (~A&~B)|~C로 Y를 ~((A|B)&C)로 동작시켰다. 결과적으로 X의 출력과 Y출력을 확인해봤고, (~A&~B)|~C 와 ~((A|B)&C)는 완전히 같은 결과를 나타내는 것으로 확인됐다.

5. 1bit 비교기 구현

**<Design source code>**

`timescale 1ns / 1ps

module bit1(

input a,b,

output x,y,w,z

);

assign x=~(a^b),

y=a^b,

w=a&~b,

z=~a&b;

endmodule

**<Simulation source code>**

`timescale 1ns / 1ps

module tb\_bit1;

reg A, B;

wire X, Y, W, Z;

bit1 bit1gate(.a(A), .b(B), .x(X), .y(Y), .w(W), .z(Z));

initial begin

A= 1'b0;

B= 1'b0;

end

always begin

A= #10 ~A;

B= #20 ~B;

end

initial begin

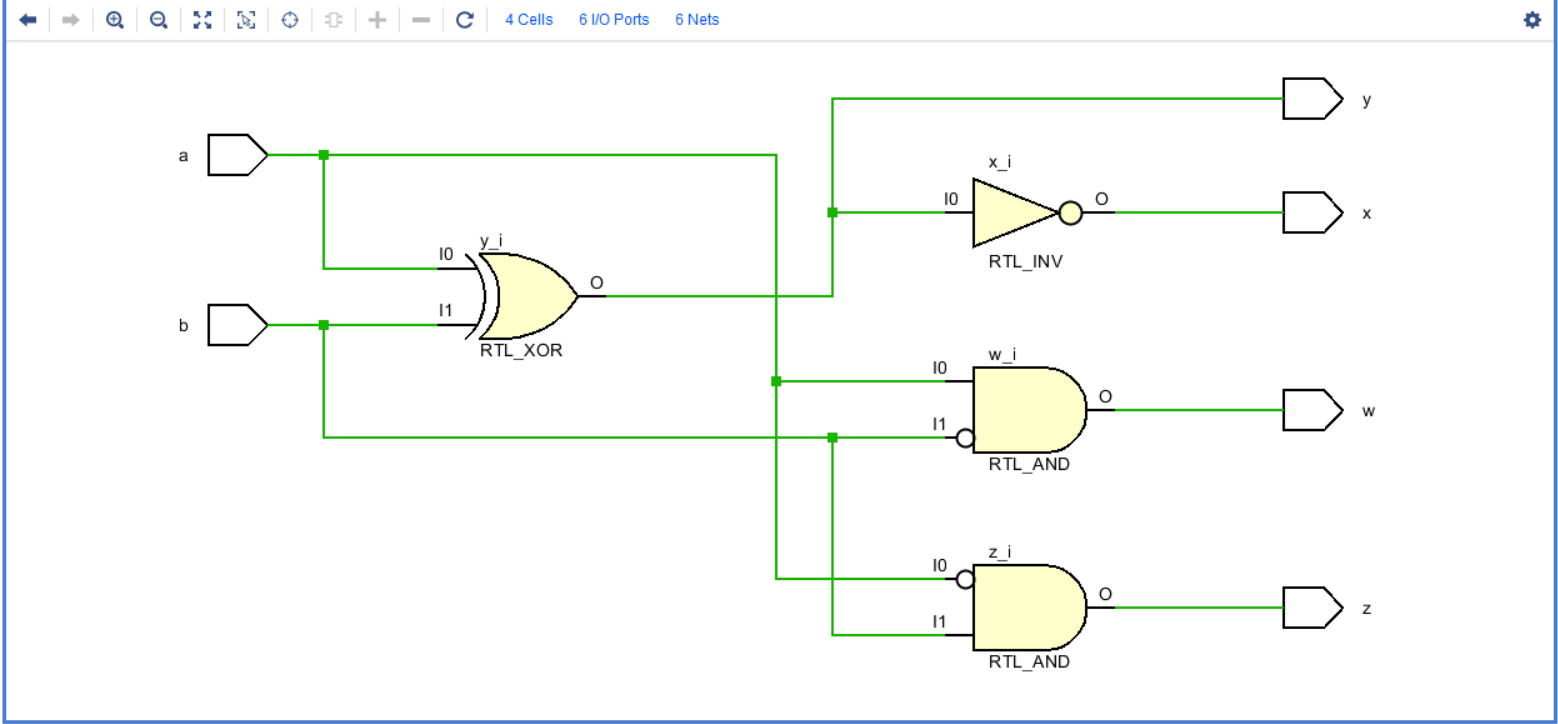
#1000

$finish;

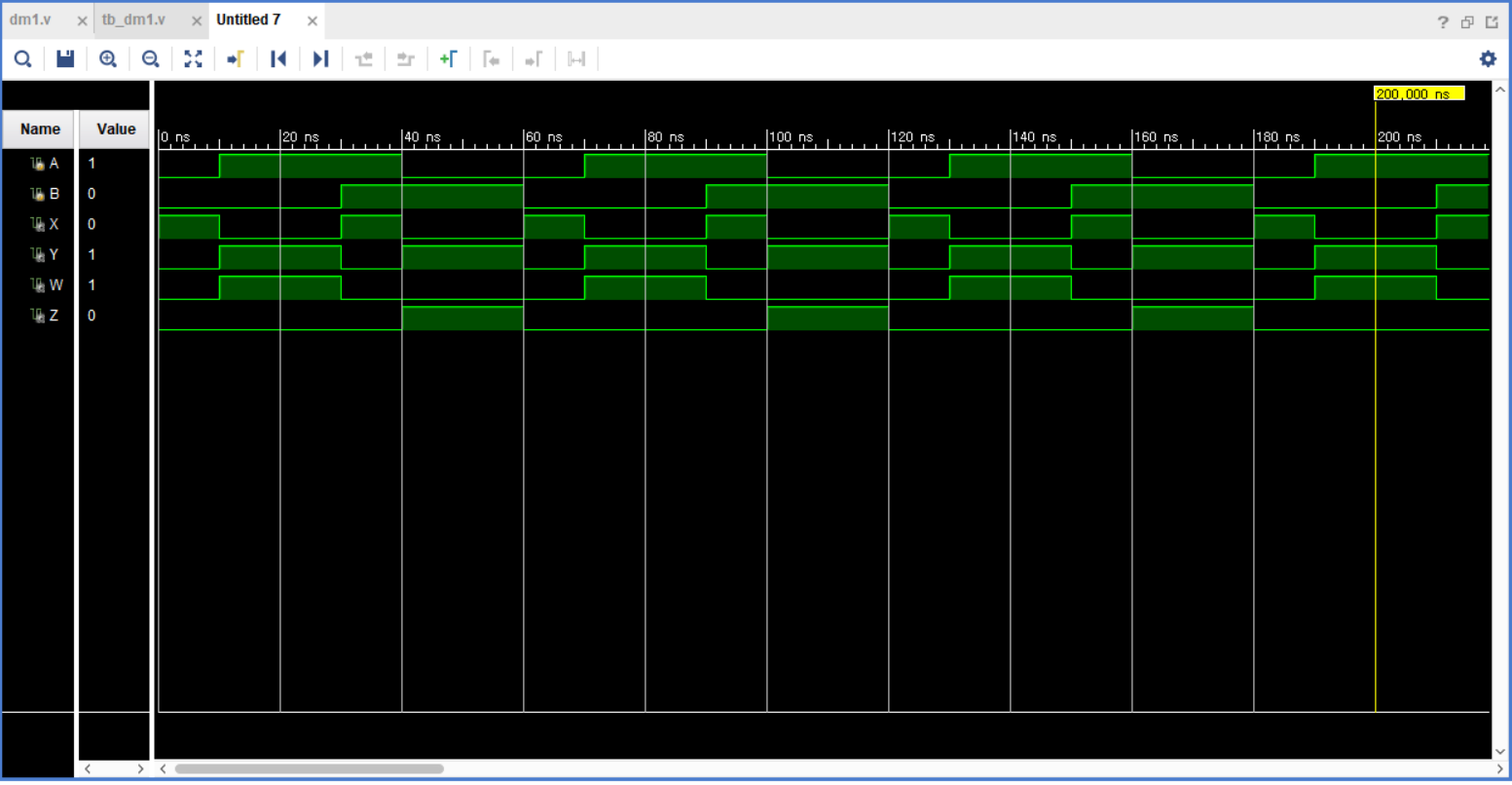
end

endmodule

**<1bit 연산기 schematic 결과>**



**<1bit 연산기 simulation 결과>**



**<1bit 연산기 진리표>**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | X(A=B) | Y(A!=B) | W(A>B) | Z(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 |

5. 결과 및 검토 사항

우선 드 모르간 제 1법칙과 제 2법칙은 각각 ~(A|B)와 ~A&~B가 서로 같은 결과가 나오는지 비교하고, ~(A&B)와 ~A|~B가 서로 같은 결과가 나오는지 비교했다. 결과적으로 제1법칙과 제 2법칙은 모두 성립한다는 결과를 확인했다. 이후 드모르간 법칙을 응용해 Boolean function에 적용하였다. (~A|~B)&~C 와 ~((A&B)|C)가 서로 같은 지 확인했다. 결과적으로 드모르간 법칙이 성립한 다는 것을 Boolean function1을 통해서도 성립함을 확인했다. 그리고 +와 \*의 위치가 바뀐 상태도 마찬가지로 드모르간 법칙이 성립함을 Boolean function2를 통해 확인했다. 이후 최종적으로 1bit 연산기를 코딩하고 simulation한 후 FPGA로 동작 시켰다. 진리표를 통해 결과값을 최종적으로 확인했다. 결과적으로

x=~(a^b),

y=a^b,

w=a&~b,

z=~a&b

위와 같이 할당해준 아웃풋들은 각각 A=B, A!=B, A>B, A<B의 뜻이 있다는 것을 확인했다.

어떠한 식이 같은 지 simulation을 확인할 수 있다. Input으로 몇 개를 받고, output을 서로 다르게 설정한 후 비교하고 싶은 식을 각각의 output에 assignment해서 simulation을 통해 비교할 수 있다.

6. 추가 이론 조사

Terminologies 정리

- Literal: 변수 하나이거나, 그 변수의 complement를 뜻한다.

- product term: literal들이 AND로 연결되어 있는 형태이다.

- minterm: 모든 변수가 포함되어 있는 product term인 경우 minterm이라 한다.

- Sum of products: OR로 연결된 product term이다.

- Canonical sum: minterm들을 합한 것이다.

- Minimum SOP: product terms를 최대한 적게 사용한 SOP expressions인 경우이다.