

2018년 2학기 컴퓨터공학실험Ⅱ
CSE3016-05반 4주차 결과 보고서

학번: 20171665

이름: 이 선 호

2018. 10. 05

목 차

I | AOI Gate Simulation 결과 및 과정

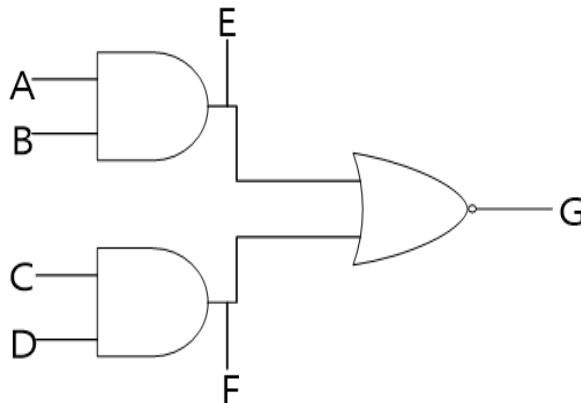
1. AOI Gate	3
-------------	-------	---

II | 논의

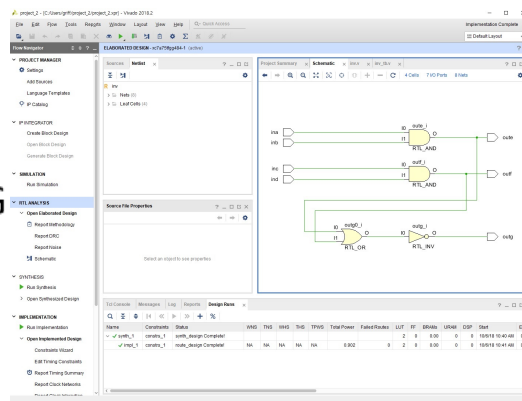
1. 결과 검토 및 논의사항	6
2. 추가 이론 조사 및 작성	7

I AOI Gate Simulation 결과 및 과정

1. AOI Gate



논리 게이트 회로도



Elaborated Design

1) Verilog 코딩

inv.v

```
`timescale 1ns / 1ps

module inv(ina, inb, inc, ind, oute, outf, outg);

input ina, inb, inc, ind;
output oute, outf, outg;

and(oute, ina, inb);
and(outf, inc, ind);
not(outg, oute || outf);

endmodule
```

[표 II -1-(1)] Verilog Design Source 코드

inv_tb.v

```
`timescale 1ns / 1ps

module inv_tb;
```

```

wire eout, fout, gout;
reg ain, bin, cin, din;

inv and_sim(.ina(ain),
.inb(bin),
.inc(cin),
.ind(din),
.oute(eout),
.outf(fout),
.outg(gout));

initial
begin
ain = 1'b0;
bin = 1'b0;
cin = 1'b0;
din = 1'b0;
end

always@(ain or bin or cin) begin
ain <= #50 ~ain;
bin <= #100 ~bin;
cin <= #200 ~cin;
din <= #400 ~din;
end

initial begin
    #800
    $finish;
end

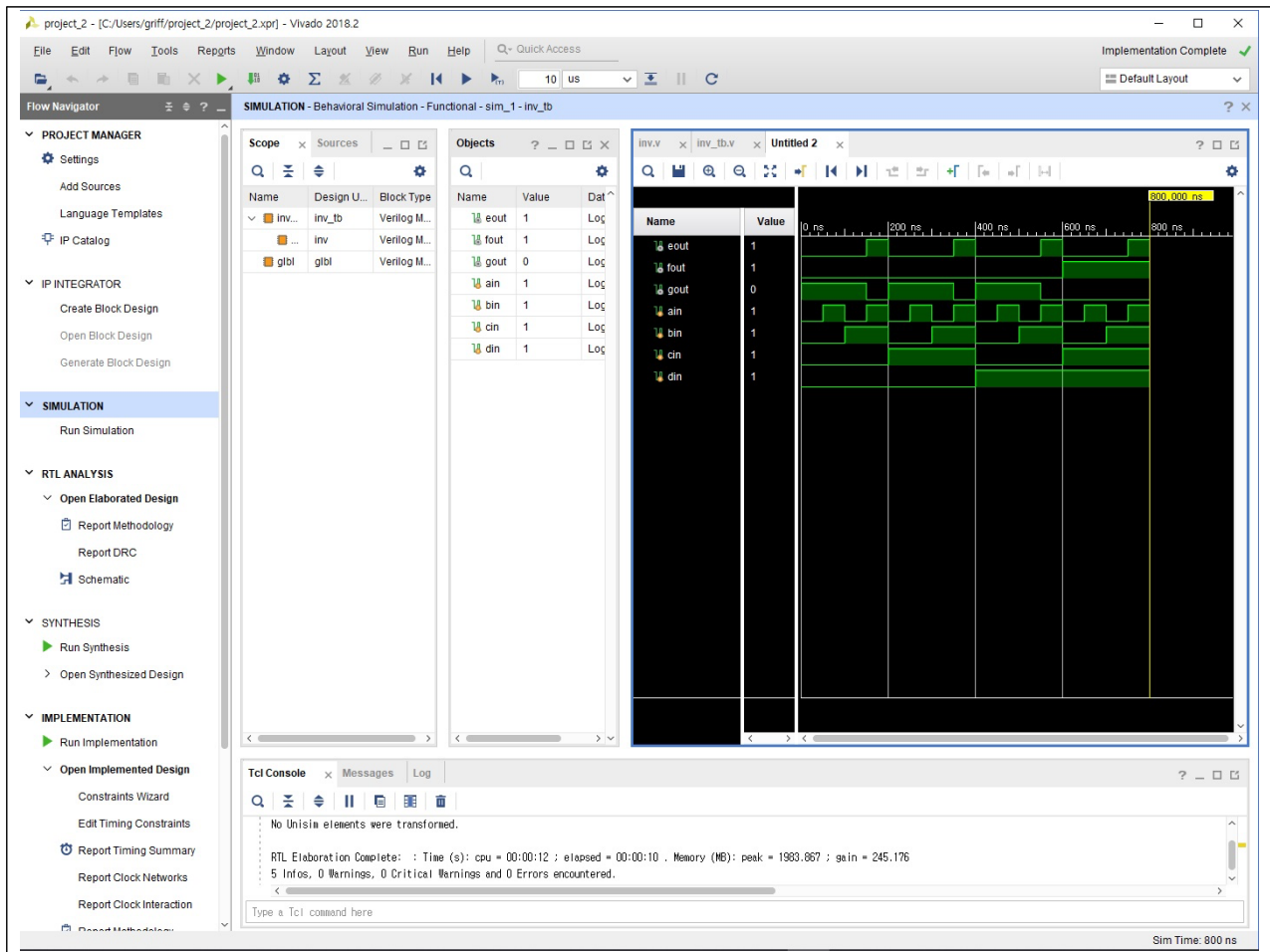
endmodule

```

[표 II-1-(2)] Verilog Simulation 코드

3) Simulation 결과

Simulation



[표 II -1-(3)] Simulation 결과

4) 진리표

Truth Table						
A	B	C	D	E	F	G
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	0	1
0	1	1	0	0	0	1

0	1	1	1	0	1	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	1	0

[표 II-1-(4)] (A)와 (B)의 진리표

5) 설명

Output E는 AND 게이트를 사용했으므로 A와 B의 input 값이 모두 1일 때 1을 출력하고, 그 외의 나머지 경우에는 0을 출력한다. Output F도 마찬가지로 C와 D의 input 값이 모두 1일 때 1을 출력하고, 그 외의 나머지 경우에는 0을 출력한다. Output G는 E와 F를 OR 논리 연산한 값을 NOT 연산하여 출력한 값이므로 OR 게이트 결과의 반대이다. 따라서 E와 F의 값 중에서 하나라도 1이 존재하면 G는 OR 게이트에서와는 다르게 0을 출력한다. 반면에 E와 F의 값이 모두 0이면 G는 1을 출력한다.

III 논의

1. 결과 검토 및 논의사항

수업 시간에 스스로 실습했던 코드에서는 'outtemp'라는 임시의 wire를 새로 선언해서 E와 F의 출력값을 OR 연산한 값을 outtemp를 통해 출력하고 나서 NOT 연산을 한 후 G로 AOI의 최종값을 출력했다. 그러나 불필요한 wire 선언을 줄이고자 결과 보고서에서는 '||' 기호를 통해 E와 F를 OR 연산한 값을 바로 NOT 연산하여 G를 통해 최종 output 값을 출력했다. Verilog에서 기본적으로 제공하는 AND, OR, NOT 연산도

사용할 수 있지만, Verilog의 문법에서 정의한 기본적인 논리 연산 기호를 사용해도 된다는 것을 알 수 있었다.

A와 B를 묶어서 첫 번째 쌍이라고 하고 C와 D를 묶어서 두 번째 쌍이라고 가정한다. 첫 번째 쌍에서 둘 중의 하나의 값이 0이고, 두 번째 쌍에서 둘 중의 하나의 값이 0이면 다른 값들과는 무관하게 AOI의 최종 출력 값은 1임을 알 수 있다. 예를 들어, A와 C의 값이 0이고, 나머지 input 값들을 모른다고 하더라도 G의 값은 1이다. 반면에, 첫 번째 쌍의 모든 값이 1이거나 두 번째 쌍의 모든 값이 1이면 다른 값들과는 무관하게 AOI의 최종 출력 값은 0임을 알 수 있다. 예를 들어, A와 B의 값이 모두 1이면 나머지 input 값들을 모른다고 하더라도 G의 값은 0이다. 이를 통해 AOI에서는 모든 input 값들을 일일이 알지 못하더라도 최소한 두 개의 값을 알고 있으면 최종 출력 값을 예측할 수 있는 경우가 존재한다는 것을 알 수 있다.

2. 추가 이론 조사 및 작성

AND, OR, 그리고 NOT 논리 연산을 응용한 Implication 연산도 존재한다. 논리식은 $x \subset y$ 또는 $x \supset y$ 이며, $x \subset y$ 는 $x + y'$ 와 동치이고 $x \supset y$ 는 $x' + y$ 와 동치이다.

AND와 OR 논리 연산을 사용한 Canonical Forms가 존재하는데, AND 연산을 사용하여 input들로 간단히 나타낸 최소항(Midterm)이 존재하고 OR 연산을 사용하여 input들로 나타낸 최대항(Maxterm)이 존재한다.

AND와 OR 게이트는 NAND 게이트와 NOR 게이트를 사용하여 구현할 수도 있다. 전자보다 후자를 사용하는 것이 상대적으로 더 효율적이기 때문인데, 이에 관한 내용은 4주차 예비보고서에 서술하였다. 아래의 표는 AND와 OR 게이트를 NAND와 NOR 게이트로 구현한 것을 보여준다.

