**CSE3016-03 컴퓨터공학실험II**

( 3 )주차 결과보고서

|  |  |
| --- | --- |
| 담당조교 | 석혜경 |
| 제출 날짜 | 2019년10월 2일 |
| 작성자 | 양희원 |
| 학번 | 20181652 |

1. **실험 목적**

디지털 시스템의 회로를 구성하는 가장 기본적인 요소이자, 0과 1로 대변되는 논리대수에 의한 연산을 직접회로 형태로 구현하는 gate에 대해 이해한다.

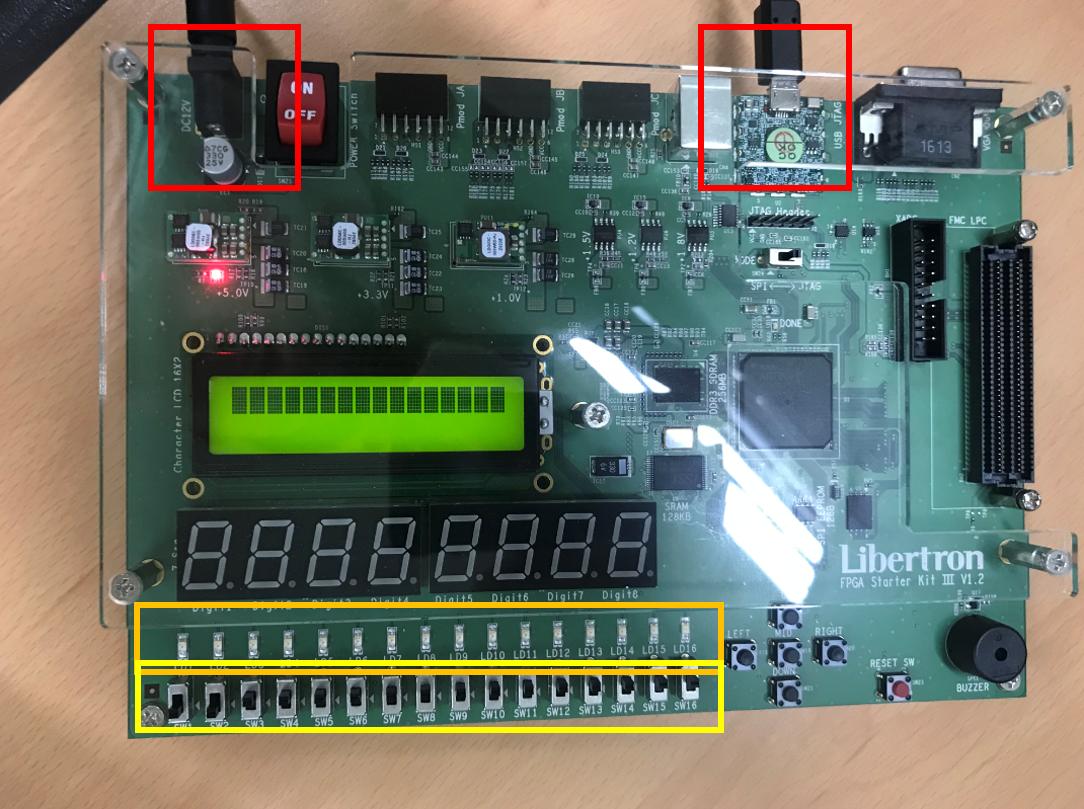
AND/OR/NOT gate의 동작을 확인한다.

Verilog를 사용하여 다중입력 AND/OR/NOT gate를 구현한다.

입력 신호를 생성하고, 구현한 각각의 gate를 simulation을 거쳐 그 동작을 확인한다.

FPGA를 통해서 Verilog로 구현된 회로의 동작을 확인하고 이해한다.

1. **FPGA 동작법을 설명하시오.**



가장 먼저 빨간색 사각형 부분을 각각 전원과 컴퓨터에 연결한다.

노란색 사각형의 스위치는 INPUT을 의미한다. 각 스위치를 내리면 INPUT=1이 된다. INPUT이 n개일 때, 스위치 n개를 사용한다. INPUT이 여러 개일 경우, 코드에서 INPUT으로 선언한 순서대로 왼쪽부터 스위치의 순서가 된다.

주황색 사각형은 OUTPUT의 결과를 나타낸다. OUTPUT=1일 때 LED불이 켜진다. OUTPUT이 여러 개일 경우, INPUT의 스위치와 마찬가지로 코드에서 선언한 순서대로 LED에 나타난다.

스위치를 조절하여 INPUT을 설정해 각각의 경우에 따른 OUTPUT 결과를 확인할 수 있다.

1. **3-input AND gate의 simulation 결과 및 과정에 대해서 설명하시오. (3 input, 2 output)**



1. Boolean식

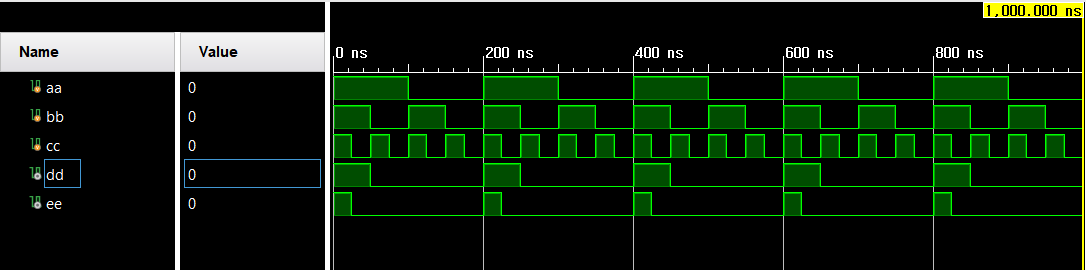
D = A\*B

E = D\*C

1. Verilog 코드

|  |  |
| --- | --- |
| Inv.v | Inv\_tb.v |
| `timescale 1ns / 1ps  module threeAnd(ina,inb,inc,outd,oute);  input ina,inb,inc;  output outd,oute;  assign outd = ina & inb;  assign oute = outd & inc;    endmodule | `timescale 1ns / 1ps  module inv\_tb;  reg aa;  reg bb;  reg cc;  wire dd;  wire ee;  threeAnd u\_threeAnd(  .ina(aa),  .inb(bb),  .inc(cc),  .outd(dd),  .oute(ee)  );  initial aa = 1'b1;  initial bb = 1'b1;  initial cc = 1'b1;  always aa = #100 ~aa;  always bb = #50 ~bb;  always cc = #25 ~cc;  initial begin  #1000  $finish;  end  endmodule |

1. Simulation



1. FPGA 동작 (a=SW1, b=SW2, c=SW3, d=LD1, e=LD2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SW1 | SW2 | SW3 | LD1 | LD2 |
| OFF | OFF | OFF | OFF | OFF |
| OFF | OFF | ON | OFF | OFF |
| OFF | ON | OFF | OFF | OFF |
| OFF | ON | ON | OFF | OFF |
| ON | OFF | ON | OFF | OFF |
| ON | OFF | OFF | OFF | OFF |
| ON | ON | OFF | ON | OFF |
| ON | ON | ON | ON | ON |

SW1과 SW2가 동시에 올라가 있을 때(SW1,SW2=ON : A,B=1), LD1에 불이 켜진다. (LD1=ON : D=1)

SW1, SW2, SW3 스위치 세 개가 동시에 올라가 있을 때(SW1,SW2,SW3=ON : A,B,C=1), LD2에 불이 켜진다. (LD2=ON : E=1)

1. Truth Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | E |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

1. **4-input AND gate의 simulation 결과 및 과정에 대해서 설명하시오. (4 input, 3 output)**



1. Boolean식

E = A\*B

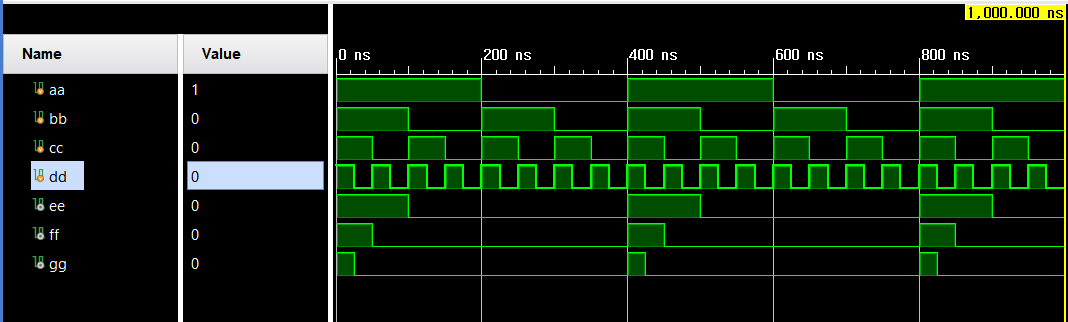
F = C\*E

G = F\*D

1. Verilog 코드

|  |  |
| --- | --- |
| Inv.v | Inv\_tb.v |
| `timescale 1ns / 1ps  module fourAnd(ina,inb,inc,ind,oute,outf,outg);  input ina,inb,inc,ind;  output oute,outf,outg;  assign oute = ina & inb;  assign outf = oute & inc;  assign outg = outf & ind;    endmodule | `timescale 1ns / 1ps  module inv\_tb;  reg aa;  reg bb;  reg cc;  reg dd;  wire ee;  wire ff;  wire gg;  fourAnd u\_fourAnd(  .ina(aa),  .inb(bb),  .inc(cc),  .ind(dd),  .oute(ee),  .outf(ff),  .outg(gg)  );  initial aa = 1'b1;  initial bb = 1'b1;  initial cc = 1'b1;  initial dd = 1'b1;  always aa = #200 ~aa;  always bb = #100 ~bb;  always cc = #50 ~cc;  always dd = #25 ~dd;  initial begin  #1000  $finish;  end  endmodule |

1. Simulation



1. FPGA 동작 (a=SW1, b=SW2, c=SW3, d=SW4, e=LD1, f=LD2, g=LD3)

SW1과 SW2가 동시에 올라가 있을 때(SW1,SW2=ON : A,B=1), LD1에 불이 켜진다. (LD1=ON : E=1)

SW1, SW2, SW3 가 동시에 올라가 있을 때(SW1,SW2,SW3=ON : A,B,C=1), LD2에 불이 켜진다. (LD2=ON : F=1)

SW1, SW2, SW3, SW4가 동시에 올라가 있을 때(SW1,SW2,SW3,SW4=ON : A,B,C,D=1), LD3에 불이 켜진다. (LD3=ON : G=1)

1. Truth Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |

1. **3-input OR gate의 simulation 결과 및 과정에 대해서 설명하시오. (3 input, 2 output)**



1. Boolean식

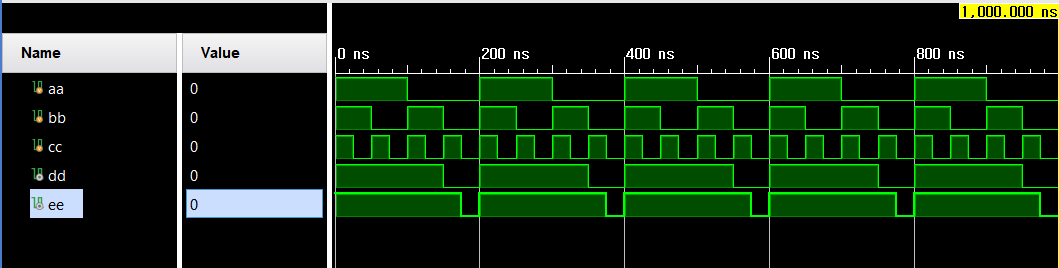
D = A|B

E = C|D

1. Verilog 코드

|  |  |
| --- | --- |
| Inv.v | Inv\_tb.v |
| `timescale 1ns / 1ps  module threeOr(ina,inb,inc,outd,oute);  input ina,inb,inc;  output outd,oute;  assign outd = ina | inb;  assign oute = inc | outd;  endmodule | `timescale 1ns / 1ps  module inv\_tb;  reg aa;  reg bb;  reg cc;  wire dd;  wire ee;  threeOr u\_threeOr(  .ina(aa),  .inb(bb),  .inc(cc),  .outd(dd),  .oute(ee)  );  initial aa = 1'b1;  initial bb = 1'b1;  initial cc = 1'b1;  always aa = #100 ~aa;  always bb = #50 ~bb;  always cc = #25 ~cc;  initial begin  #1000  $finish;  end  endmodule |

1. Simulation



1. FPGA 동작 (a=SW1, b=SW2, c=SW3, d=LD1, e=LD2)

SW1과 SW2 중 하나라도 동시에 올라가 있을 때(SW1=ON or SW2=ON : A=1 or B=1), LD1에 불이 켜진다. (LD1=ON : D=1)

SW1, SW2, SW3 스위치 세 개가 동시에 올라가 있을 때(SW1= ON or SW2=ON or SW3=ON : A=1 or B=1 or C=1), LD2에 불이 켜진다. (LD2=ON : E=1)

1. Truth Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | D | E |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

1. **4-input OR gate의 simulation 결과 및 과정에 대해서 설명하시오. (4 input, 3 output)**



1. Boolean식

E = A|B

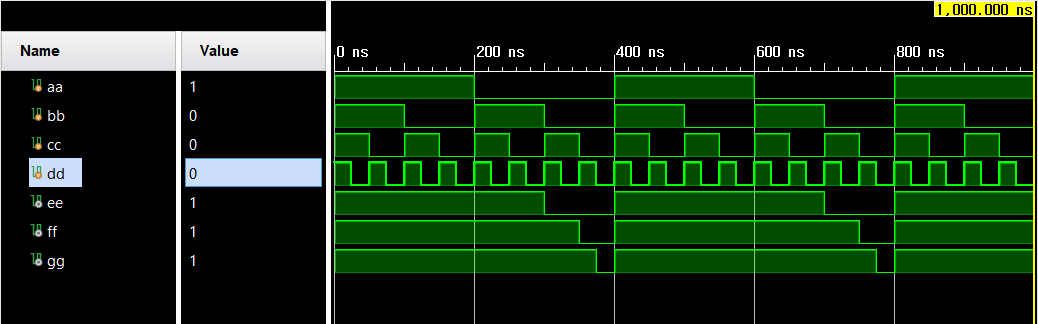
F = C|E

G = D|F

1. Verilog 코드

|  |  |
| --- | --- |
| Inv.v | Inv\_tb.v |
| `timescale 1ns / 1ps  module fourOr(ina,inb,inc,ind,oute,outf,outg);  input ina,inb,inc,ind;  output oute,outf,outg;  assign oute = ina | inb;  assign outf = oute | inc;  assign outg = outf | ind;  endmodule | `timescale 1ns / 1ps  module inv\_tb;  reg aa;  reg bb;  reg cc;  reg dd;  wire ee;  wire ff;  wire gg;  fourOr u\_fourOr(  .ina(aa),  .inb(bb),  .inc(cc),  .ind(dd),  .oute(ee),  .outf(ff),  .outg(gg)  );  initial aa = 1'b1;  initial bb = 1'b1;  initial cc = 1'b1;  initial dd = 1'b1;  always aa = #200 ~aa;  always bb = #100 ~bb;  always cc = #50 ~cc;  always dd = #25 ~dd;  initial begin  #1000  $finish;  end  endmodule |

1. Simulation



1. FPGA 동작 (a=SW1, b=SW2, c=SW3, d=SW4, e=LD1, f=LD2, g=LD3)

SW1과 SW2 중 하나라도 올라가 있을 때(SW1=ON or SW2=ON : A=1 or B=1), LD1에 불이 켜진다. (LD1=ON : E=1)

SW1, SW2, SW3 중 하나라도 올라가 있을 때(SW1=ON or SW2=ON or SW3=ON : A=1 or B=1 or C=1), LD2에 불이 켜진다. (LD2=ON : F=1)

SW1, SW2, SW3, SW4 중 하나라도 올라가 있을 때(SW1=ON or SW2=ON or SW3=ON or SW4=ON : A=1 or B=1 or C=1 or D=1), LD3에 불이 켜진다. (LD3=ON : G=1)

1. Truth Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F | G |
| 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 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 |

1. **결과 검토 및 논의사항**

Input의 개수와 gate의 종류에 따라 나누어 실험을 진행하였다. 이때 n-input에 대하여 1-output과 (n-1)-output의 중간 결과 값은 다르지만 최종 결과 값은 같다는 것을 확인할 수 있었다. 보고서에는 (n-1)-output의 결과에 대해서만 기술하였지만, boolean식이 달라도 최종 결과 값은 일치함을 확인할 수 있었다.

FPGA를 이용하여 얻은 결과는 이론적으로 truth table과 동일하다는 것을 확인할 수 있었다.

해당 실험에서는 한 종류의 논리 gate만을 이용해서 각 gate가 어떤 식으로 작동하는지 알아보았다. 일반적으로 논리 회로를 구성할 때에는 다양한 논리 gate를 이용하여 그에 따른 결과를 볼 수 있다. 따라서 단일 gate 회로가 아닌 복잡한 회로에 대해서도 탐구할 수 있을 것이다.

1. **추가 이론 조사 및 작성**
2. 논리 연산

논리 연산(logic operation, logic connective) 혹은 불 연산(Boolean operation)은 참, 거짓 {1,0} 두 가지 원소만 존재하는 집합에서의 연산이다.

논리합(OR), 논리곱(AND), 부정(NOT), 배타적 논리합(XOR), 명제, 동치 등이 있다.

1. 논리 회로

논리 회로는 불 대수를 물리적 장치에 구현한 것으로, 하나 이상의 논리적 입력 값에 대해 논리 연산을 수행하여 하나의 논리적 출력 값을 얻는 전자 회로를 말한다. AND, OR, NOT의 기본 불 대수를 수행하며, 이 기본 불 대수들의 결합으로 복합적인 논리 기능을 수행한다.

논리 회로의 설계는 논리식이나 진리표가 사용된다.

1. Boolean algebra

불 대수는 디지털 회로 설계에 응용된다. 디지털 회로는 전압의 High, Low만으로 정보를 연산하기 때문에, 기본적으로 조합 회로는 불 대수에 있는 논리식을 써서 나타낼 수 있다.

동일 진리표를 만족하는 불 대수는 여러 개가 될 수 있지만, 불 대수에 대한 진리표는 하나이다. 동일 진리표에 대한 논리 회로도는 여러 개가 될 수 있다.