**5주차 결과 보고서**

남민혁

**1. De-Morgan 의 제 1,2 법칙의 simulation 결과 및 과정에 대해서 설명하시오. (NAND,NOR 과 비교 포함)**

**1) De-Morgan의 제 1법칙**

**-Schematic 비교**

|  |  |
| --- | --- |
| A | A picture containing icon  Description automatically generated |
| B | Shape  Description automatically generated |

**-Verilog 코딩**

|  |  |  |
| --- | --- | --- |
|  | (A) | (B) |
| Verilog | `timescale 1ns / 1ps  module two\_input\_demorgan\_type\_1\_a(  input a,  input b,  output c  );    assign c = ~ (a | b);    endmodule | `timescale 1ns / 1ps  module two\_input\_demorgan\_type\_1\_b(  input a,  input b,  output c  );  assign c = (~a) & (~b);  endmodule |
| Test Bench | `timescale 1ns / 1ps  module two\_input\_demorgan\_type\_1\_a\_tb;  reg aa;  reg bb;  wire cc;  two\_input\_demorgan\_type\_1\_a u\_two\_input\_demorgan\_type\_1\_a(  .a (aa),  .b (bb),  .c (cc)  );  initial aa = 1'b0;  initial bb = 1'b0;  always aa = #200 ~ aa;  always bb = #100 ~bb;  initial begin  #400  $finish;  end  endmodule | `timescale 1ns / 1ps  module two\_input\_demorgan\_type\_1\_b\_tb;  reg aa;  reg bb;  wire cc;  two\_input\_demorgan\_type\_1\_b u\_two\_input\_demorgan\_type\_1\_b(  .a (aa),  .b (bb),  .c (cc)  );  initial aa = 1'b0;  initial bb = 1'b0;  always aa = #200 ~ aa;  always bb = #100 ~bb;  initial begin  #400  $finish;  end  endmodule |

**-Simulation 출력 결과 비교**

|  |  |
| --- | --- |
| A | A picture containing timeline  Description automatically generated |
| B | A picture containing timeline  Description automatically generated |

불리안 식에서 두 식이 동일함을 보장하는 만큼 실제로 Verilog로 시뮬레이션 한 결과도 동일하게 나타났다.

**-NOR gate와의 비교**

(a)는 사실상 지난 시간에 구현했던 NOR게이트와 동일하다. 진리표 상으로도 두 input이 0일때만 1을 출력한며, 그것의 Schematic을 보더라도 OR게이트로 만들어낸 결과에 NOT 게이트를 활용함으로써 그 결과를 반전시킨다는 점에서 NOR 게이트와 동일하다. 이때 드모르간의 제1법칙은 지난 시간에 구현한 NOR게이트를 AND 와 NOT 게이트로도 구현할 수 있음을 보여준다.

**2) De-Morgan의 제 2법칙**

**-Schematic 비교**

|  |  |
| --- | --- |
| A | Icon  Description automatically generated |
| B | A picture containing drawing  Description automatically generated |

**-Verilog 코딩**

|  |  |  |
| --- | --- | --- |
|  | (A) | (B) |
| Verilog | `timescale 1ns / 1ps  module two\_input\_demorgan\_type\_2\_a(  input a,  input b,  output c  );  assign c = ~ (a & b);  endmodule | `timescale 1ns / 1ps  module two\_input\_demorgan\_type\_2\_b(  input a,  input b,  output c  );  assign c = (~a) | (~b);  endmodule |
| Test Bench | `timescale 1ns / 1ps  module two\_input\_demorgan\_type\_2\_a\_tb;  reg aa;  reg bb;  wire cc;  two\_input\_demorgan\_type\_2\_a u\_two\_input\_demorgan\_type\_2\_a(  .a (aa),  .b (bb),  .c (cc)  );  initial aa = 1'b0;  initial bb = 1'b0;  always aa = #200 ~ aa;  always bb = #100 ~bb;  initial begin  #400  $finish;  end  endmodule | `timescale 1ns / 1ps  module two\_input\_demorgan\_type\_2\_b\_tb;  reg aa;  reg bb;  wire cc;  two\_input\_demorgan\_type\_2\_b u\_two\_input\_demorgan\_type\_2\_b(  .a (aa),  .b (bb),  .c (cc)  );  initial aa = 1'b0;  initial bb = 1'b0;  always aa = #200 ~ aa;  always bb = #100 ~bb;  initial begin  #400  $finish;  end  endmodule |

**-Simulation 출력 결과 비교**

|  |  |
| --- | --- |
| A | A picture containing chart  Description automatically generated |
| B | A picture containing green, sitting, monitor, player  Description automatically generated |

불리안 식에서 두 식이 동일함을 보장하는 만큼 실제로 Verilog로 시뮬레이션 한 결과도 동일하게 나타났다.

**-NAND gate와의 비교**

(a)는 사실상 지난 시간에 구현했던 NAND게이트와 동일하다. 진리표 상으로도 두 input이 1일때만 0을 출력한며, 그것의 Schematic을 보더라도 AND게이트로 만들어낸 결과에 NOT 게이트를 활용함으로써 그 결과를 반전시킨다는 점에서 NAND 게이트와 동일하다. 이때 드모르간의 제2법칙은 지난 시간에 구현한 NAND게이트를 OR 와 NOT 게이트로도 구현할 수 있음을 보여준다.

**(3) 진리표**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Input A | Input B | A+B | A \* B | A’ | B’ | (A+B)’ | A’ \* B’ | (A \* B)’ | A’ + B’ |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

**2. (A'+B')\*C' = ((A\*B)+C)' 의 simulation 결과 및 과정에 대해서 설명하시오. [+ 및 \* 위치 바꾼 모양도 수행]**

**1) (A'+B')\*C' = ((A\*B)+C)'**

**-Schematic 비교**

|  |  |
| --- | --- |
| A | A picture containing diagram  Description automatically generated |
| B | Diagram  Description automatically generated |

**-Verilog 코딩**

|  |  |  |
| --- | --- | --- |
|  | (A) | (B) |
| Verilog | `timescale 1ns / 1ps  module three\_input\_demorgan\_type\_1\_a(  input a,  input b,  input c,  output d  );  assign d = ((~a) | (~b)) & (~c);  endmodule | `timescale 1ns / 1ps  module three\_input\_demorgan\_type\_1\_b(  input a,  input b,  input c,  output d  );  assign d = ~((a & b) | c );  endmodule |
| Test Bench | `timescale 1ns / 1ps  module three\_input\_demorgan\_type\_1\_a\_tb;  reg aa;  reg bb;  reg cc;  wire dd;  three\_input\_demorgan\_type\_1\_a u\_three\_input\_demorgan\_type\_1\_a(  .a (aa),  .b (bb),  .c (cc),  .d (dd)  );  initial aa = 1'b0;  initial bb = 1'b0;  initial cc = 1'b0;  always aa = #400 ~ aa;  always bb = #200 ~bb;  always cc = #100 ~cc;  initial begin  #800  $finish;  end  endmodule | `timescale 1ns / 1ps  module three\_input\_demorgan\_type\_1\_b\_tb;  reg aa;  reg bb;  reg cc;  wire dd;  three\_input\_demorgan\_type\_1\_b u\_three\_input\_demorgan\_type\_1\_b(  .a (aa),  .b (bb),  .c (cc),  .d (dd)  );  initial aa = 1'b0;  initial bb = 1'b0;  initial cc = 1'b0;  always aa = #400 ~ aa;  always bb = #200 ~bb;  always cc = #100 ~cc;  initial begin  #800  $finish;  end  endmodule |

**-Simulation 출력 결과 비교**

|  |  |
| --- | --- |
| A | A picture containing green, monitor, sitting, screen  Description automatically generated |
| B | A screen shot of a computer  Description automatically generated |

불리안 식에서 두 식이 동일함을 보장하는 만큼 실제로 Verilog로 시뮬레이션 한 결과도 동일하게 나타났다.

**-진리표**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input A | Input B | Input C | (A'+B')\*C' | ((A\*B)+C)' |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 |

**2) (A'\*B')+C' = ((A+B)\*C)'**

**-Schematic 비교**

|  |  |
| --- | --- |
| A | A picture containing diagram  Description automatically generated |
| B | A picture containing diagram  Description automatically generated |

**-Verilog 코딩**

|  |  |  |
| --- | --- | --- |
|  | (A) | (B) |
| Verilog | `timescale 1ns / 1ps  module three\_input\_demorgan\_type\_2\_a(  input a,  input b,  input c,  output d  );  assign d = ((~a) & (~b)) | (~c);  endmodule | `timescale 1ns / 1ps  module three\_input\_demorgan\_type\_2\_b(  input a,  input b,  input c,  output d  );  assign d = ~((a | b) & c);  endmodule |
| Test Bench | `timescale 1ns / 1ps  module three\_input\_demorgan\_type\_2\_a\_tb;  reg aa;  reg bb;  reg cc;  wire dd;  three\_input\_demorgan\_type\_2\_a u\_three\_input\_demorgan\_type\_2\_a(  .a (aa),  .b (bb),  .c (cc),  .d (dd)  );  initial aa = 1'b0;  initial bb = 1'b0;  initial cc = 1'b0;  always aa = #400 ~ aa;  always bb = #200 ~bb;  always cc = #100 ~cc;  initial begin  #800  $finish;  end  endmodule | `timescale 1ns / 1ps  module three\_input\_demorgan\_type\_2\_b\_tb;  reg aa;  reg bb;  reg cc;  wire dd;  three\_input\_demorgan\_type\_2\_b u\_three\_input\_demorgan\_type\_2\_b(  .a (aa),  .b (bb),  .c (cc),  .d (dd)  );  initial aa = 1'b0;  initial bb = 1'b0;  initial cc = 1'b0;  always aa = #400 ~ aa;  always bb = #200 ~bb;  always cc = #100 ~cc;  initial begin  #800  $finish;  end  endmodule |

**-Simulation 출력 결과 비교**

|  |  |
| --- | --- |
| A | Graphical user interface  Description automatically generated |
| B | Graphical user interface  Description automatically generated |

불리안 식에서 두 식이 동일함을 보장하는 만큼 실제로 Verilog로 시뮬레이션 한 결과도 동일하게 나타났다.

**-진리표**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Input A | Input B | Input C | (A'\*B')+C' | ((A+B)\*C)' |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |

**3. 1Bit 비교기의 simulation 결과 및 과정에 대해서 설명하시오. (2 input, 4 output)[진리표 작성]**

**-Schematic**

Diagram, schematic

Description automatically generated

**-Verilog 코딩**

|  |  |
| --- | --- |
| Verilog | `timescale 1ns / 1ps  module one\_bit\_comparator(  input a,  input b,  output eq,  output neq,  output l,  output g  );    assign l = (~a) & b;  assign g = a & (~b);  assign eq = ~(l | g);  assign neq = l | g;  endmodule |
| Test Bench | `timescale 1ns / 1ps  module one\_bit\_comparator\_tb;  reg aa;  reg bb;  wire ee;  wire nn;  wire ll;  wire gg;  one\_bit\_comparator u\_one\_bit\_comparator(  .a (aa),  .b (bb),  .eq (ee),  .neq (nn),  .l (ll),  .g (gg)  );  initial aa = 1'b0;  initial bb = 1'b0;  always aa = #200 ~ aa;  always bb = #100 ~bb;  initial begin  #400  $finish;  end  endmodule |

**-Simulation 출력 결과**

A picture containing screenshot, green, sitting, holding

Description automatically generated

**-진리표**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Input A | Input B | A = B (eq) | A B (neq) | A > B (g) | A < B (l) |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 |

**4. 결과 검토 및 논의사항**

본 실험을 통해서 입력이 2개와 3개일 때 드모르간의 법칙이 성립함을 확인하였다. 구체적으로 드모르간 법칙에 따라 동일한 식으로 판명된 각각의 식에 대해 논리 회로로 구성하여 모든 가능한 입력 값에 대하여 출력 값이 동일함을 simulation을 통해 확인하였다.

더 나아가 마지막 실습으로서 1bit 비교기를 구현하였는데, 지난 3, 4차시에 구현했던 AND, OR, NOT, NAND, NOR, XOR 게이트를 활용하여 실생활에 의미를 가진 진리표를 구현해볼 수 있었다.

**5. 추가 이론 조사 및 작성**

입력이 2개인 드모르간 법칙이 성립함을 확인하는 실험을 통하여 지난 주차에 살펴보았던 NAND와 NOR 게이트가 각각 NOT과 OR, NOT과 AND 게이트로 구현할 수 있음을 알 수 있었다. 이를 통하여 부분적으로 NAND와 NOR게이트가 모든 논리게이트를 표현할 수 있음을 부분적으로 확인할 수 있었다.

참고로 AND와 OR게이트는 NAND와 NOR 게이트로 아래와 같이 표현할 수 있다.

**-AND게이트**

A picture containing application

Description automatically generated

**-OR 게이트**

A picture containing diagram

Description automatically generated

**6. 참고 문헌**

강석태, “Verilog HDL Summary”, http://vlsi.hongik.ac.kr/lecture/%EC%8B%A4%ED%97%98/Verilog\_Summary.pdf.

한국기술교육대학교 전기전자통신공학부, “디지털시스템 설계 및 실습” Verilog HDL 문법 강의자료, https://cms3.koreatech.ac.kr/sites/yjjang/down/dsys11/M01\_VerilogHDL01.pdf.

장영조, “디지털공학 및 실습 6장 강의자료: 부울함수의 간소화”, https://cms3.koreatech.ac.kr/sites/yjjang/down/digi09/ch06.pdf.