|  |
| --- |
| **손성우** |

|  |
| --- |
|  |
| **논리회로의 수학적 특징과 Java를 이용한 시뮬레이터 제작** |
|  |

목차

**서론**

**주제 선정 이유**

**논리회로**

논리회로

논리회로의 수학적 특징과 논리식

**논리회로 시뮬레이터**

기초 설계

논리회로 시뮬레이터

실행

머리말

컴퓨터는 최초의 컴퓨터인 에니악이 등장한 이후, 3차 산업혁명과 4차 산업혁명을 거치며 컴퓨터는 인류의 동반자가 되었다. 이러한 컴퓨터는 생각보다 단순한 구조를 가지고 있다. 컴퓨터는 기본적으론 단순한 계산기이다. 곱셈, 나눗셈도 하지 못하고, 심지어는 뺄셈조차 하지 못한다. 컴퓨터는 덧셈만이 가능하다. 그렇다면 어떻게 컴퓨터는 탄도 궤적 계산, 적분, 물리학 시뮬레이션 같은 매우 복잡한 계산들을 할 수 있을까? 이 질문에 대한 답은 논리회로에서 찾을 수 있다. 논리회로는 계산을 수행하는 컴퓨터에서 가장 중요한 부품이다. 논리회로는 논리게이트의 집합이다. 논리게이트는 기본적으로 4종류가 있다. 4종류의 논리게이트를 다양한 방식으로 조합하면 덧셈을 뺄셈으로, 곱셈으로, 그리고 나눗셈으로 변환시킬 수 있고, 복잡한 계산을 수행할 수 있다. 논리회로는 컴퓨터에서 가장 중요한 부품이자, 컴퓨터 그 자체인 것이다. 그렇기에, 컴퓨터를 이해하기 위해선 논리회로에 대한 이해가 필수적이다. 논리회로를 이해하기 위해 나는 이 보고서에서 논리회로의 기본적 개념, 논리회로의 수학적 특징에 대해 탐구하고 Java 프로그래밍 언어로 실제 작동하는 논리회로 시뮬레이터를 제작할 것이다.

주제 선정 이유

어렸을 때부터 컴퓨터에 흥미를 보이고, 기계를 조립하는 것을 좋아했던 나는 초등학생때부터 프로그래머의 꿈을 키워 나갔다. 프로그래밍을 독학하고, 컴퓨터를 많이 사용하게 되며, 자연스럽게 컴퓨터의 구조에도 관심을 가지게 되어 컴퓨터 구조에 대해 독학했다. 컴퓨터 구조를 공부하며, 컴퓨터의 작동 원리와, 다양한 부품들을 알게 되었고, 컴퓨터의 뇌인 CPU의 단위체인 논리회로에 대해 알게 되었다. 그렇게 논리회로에 대해 공부하기 시작했다. 논리회로와 수학은 큰 상관관계를 갖는다. 논리회로는 논리식으로 정리할 수 있다. 논리식은 명제로 이루어진 식이다. 명제로 이루어졌기에 집합과도 큰 연관이 있고, 명제와 집합은 모두 수(하)의 단원 중 하나이다. 이렇듯 다양한 학문이 융합한 논리회로는 나의 흥미를 끌었고, 논리회로에 대해 탐구하는 것이 학교자율과정에 적합한 주제라 생각되어 이 주제를 선정했다.

논리회로

**논리회로**

“논리회로는 하드웨어를 구성하는 기본 요소인 논리 게이트로 구성된다. 논리 게이트는 한 개 이상의 입력 단자와 하나의 출력 단자로 구성되는 전자회로다.”

임석구, 홍경호 - [디지털 논리회로]

논리회로는 0과 1의 입력에 따라 0과 1 둘 중 하나의 결과를 출력하는 논리게이트의 집합체인 회로이다. 논리회로를 구성하는 논리 게이트에는 여러가지 종류가 있다. 논리회로의 종류로는 기본적인 논리게이트인 AND, OR, NOT, BUFFER 게이트와 기본적 논리 게이트를 조합해 만든 응용 게이트인 NAND, NOR, XOR, XNOR게이트가 있다.

AND게이트

AND게이트는 입력 값이 모두 1일때만 1을 출력하는 논리 게이트로, 논리곱을 나타낸다. AND게이트는 입력 값 중 하나라도 0이 있으면 출력 값은 0이 된다. AND게이트에 대한 논리식은 [ F = A \* B ]로 나타낸다

OR게이트

OR게이트는 입력이 모두 0인 경우에만 0을 출력하고, 입력 중에 1이 하나라도 있으면 1을 출력하는 논리 게이트로, 논리합을 나타낸다. OR게이트에 대한 논리식은 [ F = A + B ] 로 나타낸다.

NOT게이트

NOT게이트는 입력이 0이라면 1을 출력하고, 1이라면 0을 출력하는 입력 값을 반전시켜 출력하는 논리 게이트로, 논리부정을 나타낸다. NOT게이트에 대한 논리식은 [ F = A’ ] 로 나타낸다.

BUFFER게이트

BUFFER게이트는 입력 값을 그대로 출력하는 단순한 전송을 나타내는 논리게이트이다. BUFFER게이트는 신호 전달 지연의 기능을 한다. BUFFER게이트에 대한 논리식은 [ F = A ] 로 나타낸다.

NAND게이트

NAND게이트는 AND게이트와 NOT게이트를 결합한 게이트로, 입력이 모두 1인 경우에만 0을 출력하고, 그렇지 않으면 1을 출력한다. NAND게이트에 대한 논리식은 [ F = (A\*B)’ ] 로 나타낸다.

NOR게이트

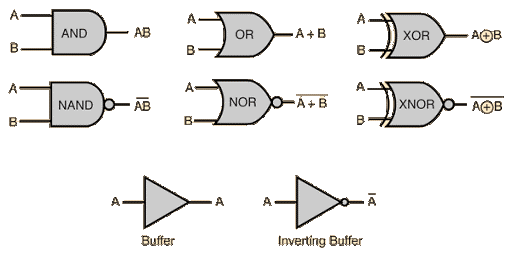
NOR게이트는 OR게이트와 NOT게이트를 결합한 게이트로, 입력이 모두 0인 경우에만 1을 출력하고, 입력 중에 하나라도 1이 있으면 0을 출력한다. NOR게이트에 대한 논리식은 [ F = (A+B)’ ] 로 나타낸다.

XOR게이트

XOR게이트는 AND, OR, NOT 게이트를 결합한 게이트로, 입력의 1이 홀수개인 경우에 1을 출력하고, 그렇지 않으면 0을 출력하는 논리 게이트로, 배타적 논리합을 나타낸다. XOR게이트에 대한 논리식은 [ F = A ⊕ B = A’B + AB’ ] 로 나타낸다.

XNOR게이트

XNOR게이트는 XOR게이트에 NOT 게이트를 결합한 게이트로, 입력의 1이 짝수개인 경우에 1을 출력하고, 그렇지 않으면 0을 출력하는 논리 게이트이다. XNOR게이트에 대한 논리식은 [ F = (A ⊕ B)’ = A’B’ + AB ] 로 나타낸다.



(각 논리회로들의 기호와 논리식)

**논리회로의 수학적 특징과 논리식**

논리회로의 수학적 특징을 알아보기 위한 첫걸음은 불 대수이다. 불 대수는 1854년 영국의 수학자 조지 불이 논리 계산을 형식화하여 도입한 대수 체계이다. 이진법을 사용하는 것이 측징이다. 기본적으로 불 대수식은 \*, +, ‘를 이용해 표현한다. 이는 각각 논리곱, 논리합, 논리부정으로써 AND, OR, NOT 게이트가 나타내는 논리식이다.

논리식은 일반적인 다항식처럼 이중 부정, 교환, 분배, 결합법칙이 성립하며, 집합법칙인 드모르간 법칙 또한 성립한다. 드모르간 법칙이란 논리합(곱)의 부정은 각각 부정의 논리곱(합)과 같다는 법칙이다. 이 부분에 대해선 아래에서 다루도록 하겠다. 드모르간 법칙은 논리식에서 특별히 중요한 법칙인데, 드모르간 법칙을 통해 NAND게이트로 모든 논리회로를 구현할 수 있다는 것을 증명할 수 있다.

논리 회로를 논리식으로 변환하는 방법은 어렵지 않다. 단순히 입력단으로부터 연결된 회로 들로부터 하나씩 나아가며 계산해가면 된다. 논리식을 논리 회로로 변환하는 것은 단순히 위의 과정을 거꾸로 하면 된다.

**항등의 법칙**

[ A + 0 = 0 + A = A ], [ A\*1 = 1\*A = A ], [ A + 1 = 1 + A = A ], [ A \* 0 = 0 \* A = 0 ], [ A + A = A ], [ A \* A = A ] [ A + A’ = 1 ], [ A \* A’ = 0 ]

항등의 법칙은 A가 0과 1중 어떤 값이더라도 항상 동일한 결과를 내는 법칙이다.

**이중 부정 법칙**

[ A’’ = A ]

이중 부정의 법칙은 A를 두 번 부정하면 A가 된다는 법칙이다.

**교환 법칙**

[ A + B = B + A ] , [ A\*B = B\*A ]

교환 법칙은 A와 B의 위치를 교환해도 성립하는 법칙이다.

**결합 법칙**

[ ( A + B ) + C = A + ( B + C ) ], [ ( A\*B )\*C = A\*( B\*C ) ]

결합 법칙은 한 식에서 연산이 두 번 이상 연속될 때, 앞쪽의 연산을 먼저 계산한 값과 뒤쪽의 연산을 먼저 계산한 결과가 항상 같을 경우 성립하는 법칙이다.

**분배 법칙**

[ A\*( B + C ) = A\*B + A\*C ], [ A + B\*C = ( A + B )\*( A + C) ]

분배법칙은 식 안의 문자를 다른 식에 분배해도 성립하는 법칙이다.