|  |  |
| --- | --- |
| **실험 결과 보고서**  **(5주차)** | 학 번 : 122201856  이 름 : 김다영  제출일 : 2021.10.07  분 반 : 002  실험조 : 개인 |

1. **실험 제목 : Karnaugh Map**
2. **실험 목적**

* Karnaugh Map을 이용하여 Bool Algebra 함수를 단순화할 수 있다.
* Karnaugh Map을 이용해 단순화 한 함수를 OrCAD Pspice를 이용해 simulation한다.

1. **실험준비**
2. 장비 셋팅 : 개인 노트북에 OrCAD Pspice 실행
3. 사용기구 및 부품 : OrCAD Pspice Program
4. 팀원 역할 분담 내역 : 개인별로 실험을 진행하였다.
5. **실험결과**
   * + - 1. **실험1**

**- 식①**

**설계방법[1] (Truth Table→Boolean equation→Logic diagram) 활용**

F는 m­3,m5, m6, m7­에서 활성화(1)되므로 이를 Truth Table에 나타내면 다음과 같다.

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | F |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

[표 1] 실험1 설계방법[1]에서의 F의 진리표

Truth Table에 따라 F를 Boolean equation으로 다음과 같이 나타낼 수 있다.

**설계방법[2] (Truth Table→Karnaugh Map→Boolean equation→Logic diagram) 활용**

식①을 참고하여 활성화되는 minterm에 1을 표시해주면 다음과 같이 K-map을 그릴 수 있다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | BC | 00 | 01 | 11 | 10 |
| A |  |
| 0 | | 0 | 0 | 1 | 0 |
| 1 | | 0 | 1 | 1 | 1 |

[표 2] 실험1 F의 Karnaugh Map

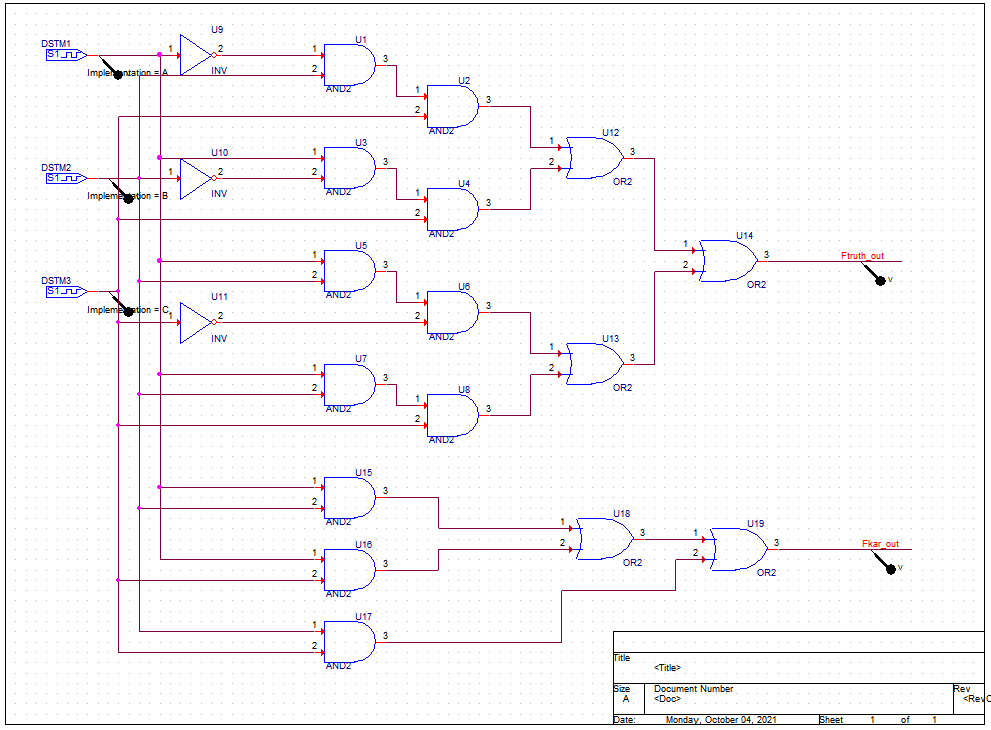
[표2]의 K-Map을 활용하여 Boolean equation을 작성하자.

* ①의 경우  
  빨간색 동그라미의 경우는 A와 C는 1로 고정되어 있고 B만 0과 1 두 가지 input을 가지고 있는 경우이다. 이때 Boolean equation을 간소화하면 CA로 나타낼 수 있다.
* ②의 경우  
  노란색 동그라미의 경우는 B와 C는 1로 고정되어 있고 A만 0과 1 두 가지 input을 가지고 있는 경우이다. 이때 Boolean equation을 간소화하면 BC로 나타낼 수 있다.
* ③의 경우  
  파란색 동그라미의 경우는 A와 B는 1로 고정되어 있고 C만 0과 1 두 가지 input을 가지고 있는 경우이다. 이때 Boolean equation을 간소화하면 AB로 나타낼 수 있다.

위 세 경우에서 구한 식을 모두 더하면 다음과 같은 Boolean equation을 구할 수 있다.

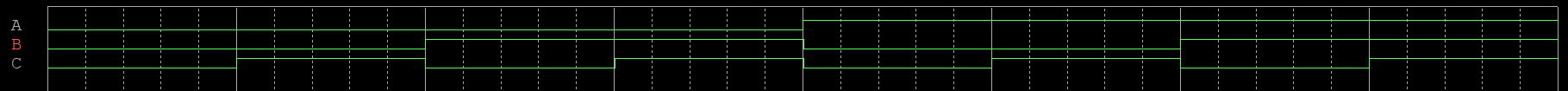
**OrCAD Pspice simulation**

설계방법[1]과 설계방법[2]에서의 Boolean equation을 OrCAD Pspice를 이용해 Logic diagram으로 구성하면 [그림1] 같이 회로를 그릴 수 있다. [그림1]의 Logic diagram에는 3개의 입력이 필요하므로 3개의 입력 A, B, C를 설정해야 한다.



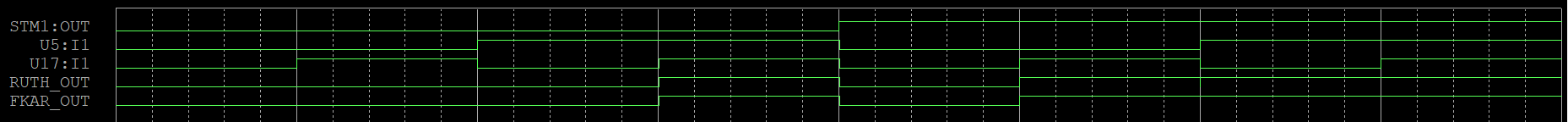
[그림 1] 실험1의 설계방법[1]/설계방법[2]에 따른 OrCAD 논리 회로도

입력이 A, B, C 3개이므로 23=8개의 입력을 설정해줘야 한다. X축 값의 범위는 0ms부터 80ms로 설정하여 10ms마다 다른 경우의 신호를 입력한다. 8개의 입력 신호를 모두 설정해준 결과 [그림2]와 같은 입력 파형을 확인할 수 있다. 이때 wire에서의 입출력 파형을 확인하기 위해 각각의 입력(A, B, C)과 출력(Ftruth\_out, Fkar\_out)에 Voltage/Level Marker를 설치한다.



[그림 2] 실험1 입력 파형

이후 simulation file을 생성하고 Run To Time을 입력 신호의 시간 범위와 동일한 80ms로 설정한 뒤 RUN하면 [그림 3]와 같은 입력 A, B, C의 파형에 따른 Ftruth\_out(설계방법[1]의 출력)과 Fkar\_out(설계방법[2]의 출력) 파형이 발생하는 것을 볼 수 있다.



**[그림 3] 입력 A,B,C에 따른 출력**

**Ftruth\_out(설계방법[1])과 Fkar\_out(설계방법[2])**

이때 Fkar\_out에 대한 Truth table을 작성하면 [표 3]과 같으며 이는 설계방법[1]에서의 F의 진리표인 [표 1]과 동일하다는 것을 알 수 있다. 또 [그림 3]에서의 Ftruth\_out과 Fkar\_out 출력 파형이 동일한 것을 통해 logic circuit을 구성하는 방식의 차이가 존재하더라도 출력 결과는 동일하다는 것을 알 수 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | F |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

[표 3] 실험1 설계방법[2]에서의 F의 진리표

* + - * 1. **실험2**

**– 식②**

**설계방법[2] (Truth Table→Karnaugh Map→Boolean equation→Logic diagram) 활용**

식②를 참고하여 활성화되는 minterm에 1을 표시하고 Don’t care value에 X를 표시해주면 다음과 같이 K-map을 그릴 수 있다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | CD | 00 | 01 | 11 | 10 |
| AB |  |
| 00 | | X | 1 | 1 | 1 |
| 01 | | 1 | 0 | 0 | 1 |
| 11 | | X | 0 | 0 | 1 |
| 10 | | 1 | 0 | 0 | 1 |

[표 4] 실험2 F의 Karnaugh Map

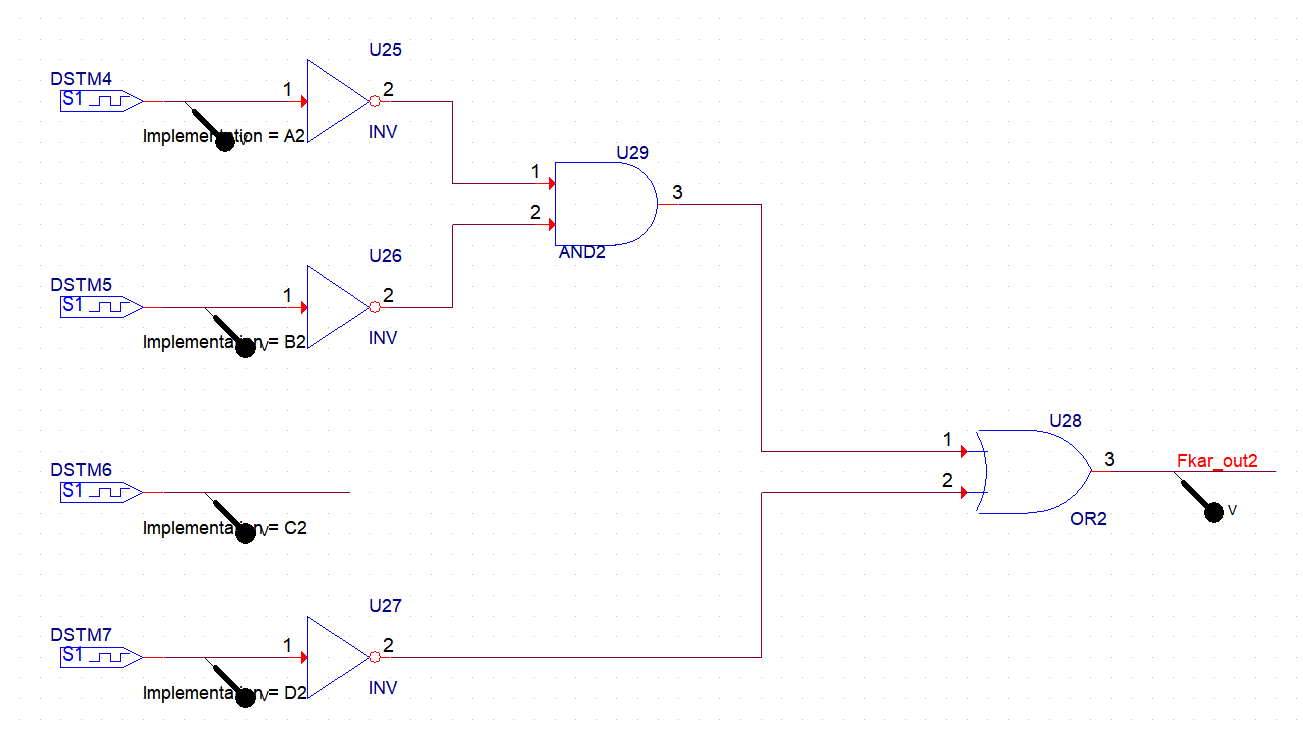
[표4]의 K-Map을 활용하여 Boolean equation을 작성하자.

* ①의 경우  
  빨간색 동그라미의 경우는 D만 0으로 고정되어 있고 A, B, C는 0과 1 두 가지 input을 가지고 있는 경우이다. 이때 Boolean equation을 간소화하면 D’으로 나타낼 수 있다.
* ②의 경우  
  초록색 동그라미의 경우는 A와 B는 0으로 고정되어 있고 C와 D는 0과 1 두 가지 input을 가지고 있는 경우이다. 이때 Boolean equation을 간소화하면 A’B’로 나타낼 수 있다.

위 두 경우에서 구한 식을 모두 더하면 다음과 같은 Boolean equation을 구할 수 있다.

**OrCAD Pspice simulation**

설계방법[2]에서의 Boolean equation을 OrCAD Pspice를 이용해 Logic diagram으로 구성하면 [그림4] 같이 회로를 그릴 수 있다. [그림4]의 Logic diagram에는 4개의 입력이 필요하므로 4개의 입력 A2, B2, C2, D2를 설정해야 한다.



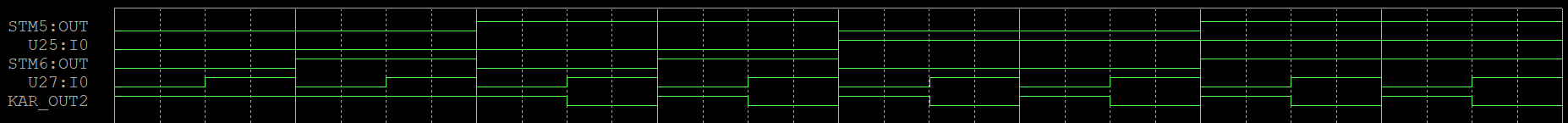
[그림 4] 실험2 설계방법[2]에 따른 OrCAD 논리 회로도

입력이 A2, B2, C2, D2 4개이므로 24=16개의 입력을 설정해줘야 한다. X축 값의 범위는 0ms부터 160ms로 설정하여 10ms마다 다른 경우의 신호를 입력한다. 16개의 입력 신호를 모두 설정해준 결과 [그림5]와 같은 입력 파형을 확인할 수 있다. 이때 wire에서의 입출력 파형을 확인하기 위해 각각의 입력(A2, B2, C2, D2)과 출력(Fkar\_out)에 Voltage/Level Marker를 설치한다.



[그림 5] 실험2 입력 파형

이후 simulation file을 생성하고 Run To Time을 입력 신호의 시간 범위와 동일한 160ms로 설정한 뒤 RUN하면 [그림 6]와 같은 입력 A, B, C, D의 파형에 따른 Fkar\_out(설계방법[2]의 출력) 파형이 발생하는 것을 볼 수 있다.



**[그림 6] 입력 A,B,C에 따른 출력 Fkar\_out(설계방법[2])**

이때 Fkar\_out에 대한 Truth table을 작성하면 [표 5]와 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **F** |
| **0** | **0** | **0** | **0** | **1** |
| **0** | **0** | **0** | **1** | **1** |
| **0** | **0** | **1** | **0** | **1** |
| **0** | **0** | **1** | **1** | **1** |
| **0** | **1** | **0** | **0** | **1** |
| **0** | **1** | **0** | **1** | **0** |
| **0** | **1** | **1** | **0** | **1** |
| **0** | **1** | **1** | **1** | **0** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **F** |
| **1** | **0** | **0** | **0** | **1** |
| **1** | **0** | **0** | **1** | **0** |
| **1** | **0** | **1** | **0** | **1** |
| **1** | **0** | **1** | **1** | **0** |
| **1** | **1** | **0** | **0** | **1** |
| **1** | **1** | **0** | **1** | **0** |
| **1** | **1** | **1** | **0** | **1** |
| **1** | **1** | **1** | **1** | **0** |

[표 5] 실험2 설계방법[2]에서의 F의 진리표

* + - * 1. **논리식 간소화를 통한 실험1 검토**

실험 1의 설계방법[1]에서 Truth Table에 따른 F의 Boolean equation을 다음과 같이 구할 수 있었다.

**– 식③**

Boolean Algebra에 따라 식③을 간소화하여 다음과 같은 결과를 얻을 수 있다.

식③을 간소화한 결과 **F = AB+BC+CA**라는 논리식을 구할 수 있으며 이는 실험1에서 설계방법[2]를 통해 구한 식②과 동일하다는 것을 알 수 있다.

1. **고찰**

이번 실험은 K-map을 활용하여 논리 회로를 구성하고, 기존에 사용하던 논리 회로 설계 방식과 결과가 동일한지를 알아보는 실험이었다. 실험1과 2를 진행한 결과 방법에 상관없이 출력이 동일하다는 것을 알 수 있었다.

실험 2에서 K-map을 이용하여 식을 간소화하기 위해 이웃한 1과 X를 묶을 때, 4개와 8개를 한 묶음으로 묶은 게 아니라 4개짜리 세 묶음으로 묶어 간소화했다. 이후 간소화한 식을 확인했는데 Boolean Algebra에 따라 식이 더 간소화된다는 것을 확인했고 K-map을 다시 검토하여 8개로 묶을 수 있다는 것을 확인할 수 있었다. 1을 작게 묶어도 식이 간소화되긴 하지만 크게 묶을수록 더욱 식이 간단해질 수 있다는 것을 실수를 통해 알아낼 수 있었다.

또 K-map을 통해 간소화한 식이 기존의 논리 설계 방식에서 Truth Table로 식 F를 작성한 후 F를 Boolean Algebra에 따라 간소화한 식과 동일하다는 것을 알 수 있었다. F의 논리식을 간소화하기 위해선 간소화 식을 외우고 있어야 한다는 불편함이 있지만, K-map을 이용하면 간소화 식을 외우고 있지 않아도 쉽게 논리식을 간소화시킬 수 있다는 점에서 K-map의 유용함을 다시 알 수 있었다.

|  |  |
| --- | --- |
| **예비실험 보고서**  **(6주차)** | 학 번 : 122201856  이 름 : 김다영  제출일 : 2021.10.03  분 반 : 002 |

1. **실험 제목 : OP-AMP를 이용한 기본증폭[1]**
2. 실험 목적

* OP-AMP의 특성을 알고, 이를 활용한 반전 증폭기 및 비반전 증폭기의 기능을 학습한다.
* OP-AMP를 이용한 회로를 설계하여 OrCAD Pspice를 이용해simulation한다.

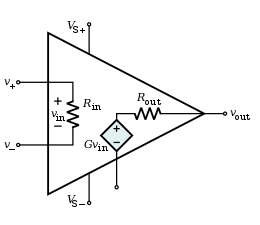
1. 실험 이론
   1. OP-AMP

텍스트, 안테나이(가) 표시된 사진

자동 생성된 설명

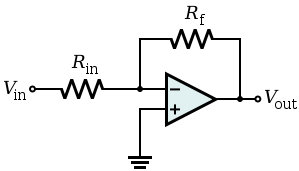
[그림 1] 연산 증폭기의 회로 기호

* + OP-AMP는 연산 증폭기로 두 개의 차등 입력과, 한 개의 단일 출력을 가지는 직류 연결형 고이득 전압 증폭기이다.
  + 하나의 연산 증폭기는 그 입력 단자 간의 전위차보다 대개 백배에서 수 천배 큰 출력 전압을 생성한다.
  1. OP-AMP 기능 및 특성



[그림 2] 연산 증폭기 등가모델

* + 연산 증폭기는 자신의 두 입력 단자에 인가된 전압 신호의 차(V2-V1)를 감지하고, 이 값에 이득 A를 곱한 후, 그 결과 전압 A(V2-V1)를 출력 단자 3에 나타내도록 고안되어 있다. 따라서 V3= A(V2-V1)이다.
  + 이상적인 연산 증폭기는 어떤 입력 전류도 끌어들이지 못하므로 단자 1과 단자2로 들어오는 신호 전류는 모두 0이다. 따라서 이상적인 연산 증폭기의 입력 임피던스는 무한대이다.
  + 단자 3과 접지 사이의 전압은 단자 3으로부터 나와 부하 임피던스로 들어가는 전류에 관계없이 항상 A(V2-V1)이다. 즉 이상적인 연산 증폭기의 출력 임피던스는 0이다.
  + 이상적인 연산 증폭기는 무한대의 대역폭을 가지며, 이득 A는 매우 크고 이상적으로는 무한대의 값을 가진다.
  1. 반전 증폭기

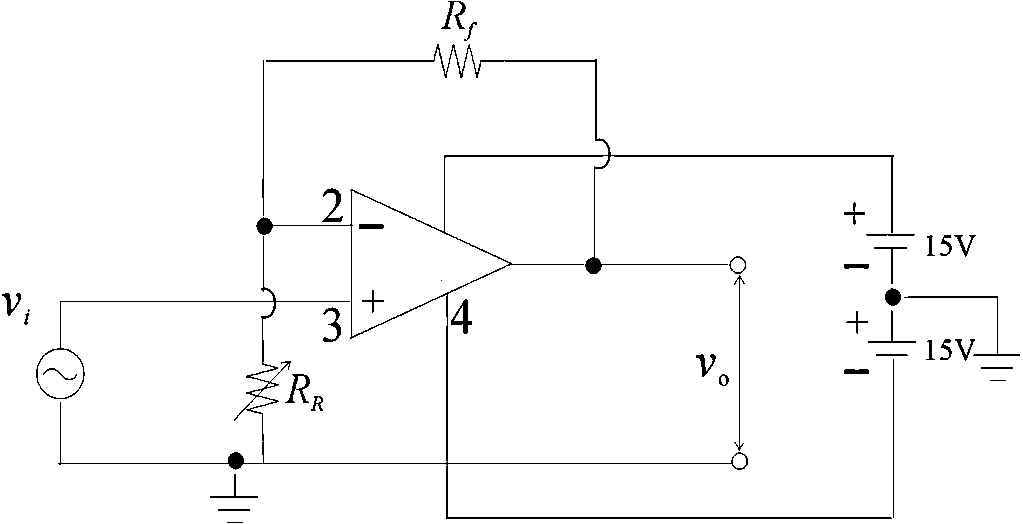


[그림 3] 반전 증폭기 회로

* + 반전 증폭기는 입력된 신호에 대해 정해진 증폭도로 신호가 반전되어 출력되는 증폭기이다.
  + 정의 V3= A(V2-V1)를 변형하여 이라 할 수 있다. 연산 증폭기의 이득 A는 매우 크기 때문에, 그 입력 단자 사이의 전압은 무시할 수 있을 정도로 작다. 따라서 이고 가 성립한다. 따라서 두 입력 단자가 같은 전위로 따라다니게 되는 현상이 발생한다.
  + 단자 2에 어떤 전압이 걸리더라도 무한대의 이득 A로 인해 단자 1에 그 전압이 자동적으로 나타난다. 단자 2는 종종 접지로 접속되어 V2=0이 되고 이러한 경우 V1=0이 된다. 이때 단자 1이 **가상 접지**되었다고 말한다.
  + 이상적인 연산 증폭기는 무한대의 입력 임피던스를 가지고 있으므로, 그것의 입력 단자로는 전류가 흐르지 않는다. 따라서 Rin에 흐르는 전류와 Rf에 흐르는 전류가 동일하다는 것을 알 수 있고, 이에 대해 키르히호프의 법칙을 이용하면 다음과 같은 식이 성립한다.

이때 Vs는 가상 접지된 지점이므로 0이라고 할 수 있다. 따라서 이를 적용하면 다음 식이 성립한다.

* 1. 비반전 증폭기



[그림 4] 비반전 증폭기 회로

- 연산 증폭기의 특징에 따라 입력단자 V+와 V-에 들어가는 입력의 전류는 0이며 이때 V+ = V-이다. 따라서 회로에서 Vi= V+ = V-가 성립한다.

- Vout과 Vs에 대해 전압 분배법칙을 이용하면 다음 식이 성립한다.

이때 Vs는 가상 접지 지점으로 Vi와 유사한 값을 가진다. 따라서

가 성립하며 위 식을 정리하면 다음 식을 구할 수 있다.

- 참고문헌 -

[1] MAXIM Application Note 1108: Understanding Single-Ended, Pseudo-Differential and Fully-Differential ADC Inputs — Retrieved November 10, 2007

[2] Adel S. Sedra, 마이크로전자회로, 한티미디어, 2016