6주차 예비보고서

전공 : 컴퓨터공학과 학년 : 2학년 학번 : 20211558 이름 : 윤준서

**1. 전가산기와 반가산기**

반가산기는 두 개의 비트 입력(A, B)을 더하여 합(S)과 자리 올림 수(C)를 구하는 조합 논리 회로다. 즉 두 입력 값의 합이 자리 수를 올리면 C의 값은 1이 된다. 반가산기는 두 개의 비트만 더할 수 있고, 이전 단계의 자리 올림 수는 처리하지 않는다.

반가산기의 논리식과 진리표는 다음과 같다.

**S = A + B**

**C = A • B**

|  |  |  |  |
| --- | --- | --- | --- |
| **A** | **B** | **S** | **C** |
| **0** | **0** | **0** | **0** |
| **0** | **1** | **1** | **0** |
| **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **1** |

전가산기는 두 개의 비트 입력(A, B)과 자리 올림 수(Carry-in)를 더하여 합(S)과 자리 올림 수(Carry-out)을 구하는 조합 논리 회로다. 반가산기와 달리 이전 단계의 자리 올림 수를 처리하기에 여러 비트의 덧셈에 사용할 수 있다.

전가산기의 논리식과 진리표는 다음과 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** |  | **S** |  |
| **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **1** | **0** |
| **0** | **1** | **0** | **1** | **0** |
| **0** | **1** | **1** | **0** | **1** |
| **1** | **0** | **0** | **1** | **0** |
| **1** | **0** | **1** | **0** | **1** |
| **1** | **1** | **0** | **0** | **1** |
| **1** | **1** | **1** | **1** | **1** |

**2. 전감산기와 반감산기**

반감산기는 두 개의 비트 입력(X, Y)을 빼서 차(D)와 자리 내림 수(B)를 구하는 조합 논리 회로다. 즉 두 입력 값의 차가 자리 수를 내리면 B의 값은 1이 된다. 반감산기는 두 개의 비트만 뺄 수 있고, 이전 단계의 자리 내림 수는 처리하지 않는다.

반감산기의 논리식과 진리표는 다음과 같다.

**D = X ⊕ Y**

**B = X' • Y**

|  |  |  |  |
| --- | --- | --- | --- |
| **X** | **Y** | **D** | **B** |
| **0** | **0** | **0** | **0** |
| **0** | **1** | **1** | **1** |
| **1** | **0** | **1** | **0** |
| **1** | **1** | **0** | **0** |

전감산기는 두 개의 비트 입력(X, Y)과 자리 올림 수(Borrow-in)를 빼서 차(D)와 자리 내림 수(Borrow-out)을 구하는 조합 논리 회로다. 반감산기와 달리 이전 단계의 자리 내림 수를 처리하기에 여러 비트의 뺄셈에 사용할 수 있다.

전감산기의 논리식과 진리표는 다음과 같다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **X** | **Y** |  | **D** |  |
| **0** | **0** | **0** | **0** | **0** |
| **0** | **0** | **1** | **1** | **1** |
| **0** | **1** | **0** | **1** | **1** |
| **0** | **1** | **1** | **1** | **0** |
| **1** | **0** | **0** | **0** | **1** |
| **1** | **0** | **1** | **0** | **0** |
| **1** | **1** | **0** | **0** | **0** |
| **1** | **1** | **1** | **1** | **1** |

**3. BCD 가산기**

BCD(Binary-Coded Decimal)란 말 그대로 10진수 숫자(0 ~ 9)를 4비트 이진수로 표현하는 방식이다. BCD 가산기는 두 개의 4비트 BCD 숫자와 이전 단계의 자리 올림 수를 입력으로 받은 후, 두 BCD 숫자를 더해서 합과 자리 올림 수를 구하는 조합 논리 회로다.

두 숫자의 합이 9(1001)를 넘을 경우 10진수 범위가 초과된다. 이때 조정 작업을 실행하는데, 6(0110)을 더하여 8비트 숫자로 만들어서 십의 자리 수와 일의 자리 수를 구분하여 결과 값을 출력한다.

예시로 10진수 7(0111)과 6(0110)을 더하는 과정은 다음과 같다.

1. 4비트로 변환해 더하면 1101(10진수로 13)이다.

2. 9를 초과하므로 6(0110)을 더하면 0001 0011이다.

3. 최종 출력은 0001 0011으로, 이는 10진수 13을 나타내며 두 자리 숫자로 표현된다. 십의 자리와 일의 자리를 분리한 형태다.

**4. 병렬 가감산기**

병렬 가감산기는 뺄셈 연산을 2의 보수(two's complement) 덧셈으로 표현한 가산기다. 컴퓨터에선 뺄셈을 덧셈으로 변환하여 처리하는데, 이 때 2의 보수가 사용된다. 즉 **A - B**를 연산할 때 B의 2의 보수 형태를 구한 값을 C라 하면, **A + C**를 계산하여 출력하는 방식이다. 이 때 C를 구하는 방법은 다음과 같다.

**-B = C = B' + 1**

여러 비트를 동시에 계산할 수 있는 병렬 가감산기는 속도에 유리한 조합 논리 회로다.

**5. Carry Look-Ahead Adder / Ripple Carry Adder**

CLA로도 불리는 Carry Look-Ahead Adder와 RCA로도 불리는 Ripple Carry Adder는 모두 다수의 비트 덧셈을 수행하는 가산기다. 두 가산기는 자리 올림 수를 계산하는 방식에 차이가 있다.

**RCA**의 경우 두 개의 n비트 변수를 더할 때 각 자리의 덧셈을 전가산기를 통해 계산한다. 이후 첫 번째 자리에서부터 자리 올림이 발생하면 두 번째 자리에 전달한다. 이 과정을 가장 높은 자리까지 순차적으로 진행한다. 회로 설계가 간단하지만 순차적인 진행 방식에 의해 비트 수가 증가할 수록 연산 시간이 길어진다는 단점이 있다.

이러한 RCA의 문제점을 개선하기 위해 만들어진 가산기가 **CLA**이다. CLA도 마찬가지로 자리 올림 수를 순차적으로 전달하지만, 자리 올림을 미리 계산해 처리함으로써 연산 속도를 줄인다. 이 자리 올림 계산을 위해 사용하는 신호가 있는데, Generate와 Propagate라고 불린다. Generate 신호는 두 변수에 AND 연산을 통해 자리 올림이 발생하는지 확인한다. Propagate 신호는 두 변수에 OR 연산을 통해 자리 올림 수가 다음 자리로 전달되는지 확인한다. 다음은 자리 올림 연산의 과정이다.

위와 같이 순차적으로 비트 계산을 자리 올림을 포함하여 한번에 처리하기에 CLA는 연산 속도가 빠르다는 장점이 있지만, RCA와 반대로 회로가 복잡해지는 단점이 있다.

**6. 기타 이론**

감가산기, 반감산기 등의 조합 회로는 게이트의 조합으로 이루어지는 논리 회로로 입력이 주어지면 바로 출력이 나온다. 메모리 소자를 갖지 않기에 이전의 회로 상태가 출력에 영향을 미치지 않는다. 다음은 기타 여러 조합 회로의 간단한 특징이다.

**리플 캐리 가산기** : 2개 이상의 전가산기를 병렬로 연결하여 여러 비트의 덧셈을 계산한다.

**인코더** : 여러 개의 입력 신호 중 단 하나의 활성화(값이 1)된 입력을 표현하기 위해 암호화하여 출력하는 장치다. 10진수 등의 정보를 2진수와 같은 코드로 변환하는 부호기라고 부르기도 한다. 활성화되는 입력 신호가 바뀌면 암호화되는 출력도 바뀐다.

**디코더** : 암호화된 코드를 해독하는 회로로 복호기라는 말로도 불린다. 인코딩 된 출력 값이 담고 있는 하나의 활성화된 신호를 다시 풀어서 출력하는 장치로, 하나의 비트만 1의 값을 가진다. 입력선에 나타나는 n비트의 2진 코드를 최대 2^n 개의 서로 다른 정보로 바꿔주는 조합 논리 회로다. 인코더와 반대의 연산을 수행한다.

**멀티플렉서(MUX)** : 여러 개의 입력 선들 중 하나를 선택해 출력하는 회로다. 데이터 선택기라고도 불린다.

**디멀티플렉서(DEMUX)** : 한 개의 입력 선으로 가능한 여러 개의 출력 선들 중 하나를 선택해 출력하는 회로다. 데이터 분배기라고도 불리며, 멀티플렉서와 반대의 연산을 수행한다.