컴퓨터 공학 기초 설계 및 실험1

결과 보고서

실험제목 : Adder& Subtractor using 2’s complement

실험일자: 2018년 05월 31일 (목)

제출일자: 2018년 06월 07일 (목)

학 과: 컴퓨터정보공학부

담당교수: 이준환

실습분반: 목요일(0,1,2)

학 번: 2015722025

성 명: 정용훈

결과보고서

1. 제목 및 목적
   1. 제목

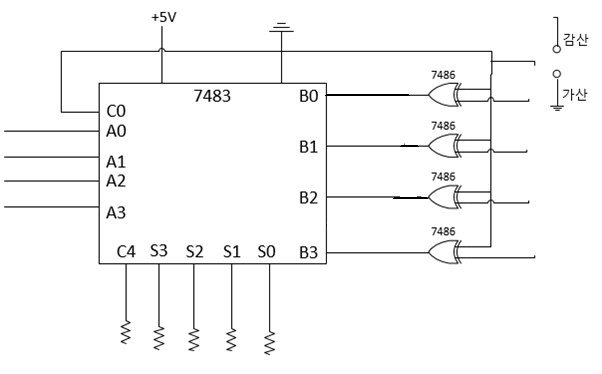
Adder& Subtractor using 2’s complement

* 1. 목적

보수에 대한 이해와 보수를 이용한 병렬 가감산기의 회로를 설계할 수 있다. 설계한 가감산기가 올바르게 동작함을 확인하고 원리를 이해할 수 있다. 가감산기의 동작 확인을 목적으로 한다.

1. 실험 결과

**-실험 8-1**



-전가산기

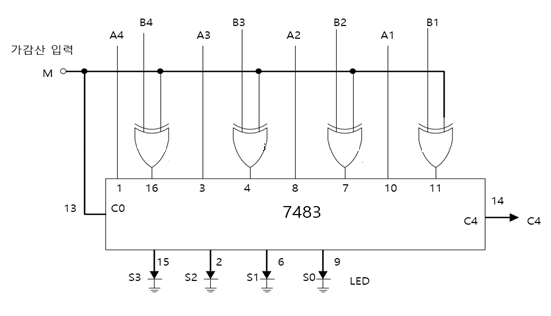
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 입력 A | | | | 입력 B | | | | 출력 | | | | |
| A3 | A2 | A1 | A0 | B3 | B2 | B1 | B0 | C4 | S3 | S2 | S1 | S0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

-전감산기

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 입력 A | | | | 입력 B | | | | 출력 | | | | |
| A3 | A2 | A1 | A0 | B3 | B2 | B1 | B0 | b4 | d3 | d2 | d1 | d0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

위의 회로는 2의 보수를 이용한 4비트 2진 전감산기와 전가산기의 실험회로이다. 입력 스위치가 감산일때는 전감산기 역할을하고, 가산일때는 전가산기 역할을 한다. 전감산기에서 2의 보수가 사용되는데, XOR 게이트를 이용한다. XOR 게이트는 입력 값 둘 중 하나가 1이면 다른 입력 값을 반전시키는 특징이 있다. 이를 이용하여 각 입력 값을 반전시킨 후 마지막에 감산 스위치의 입력 값을 더해주면 2의 보수를 구할 수 있다.

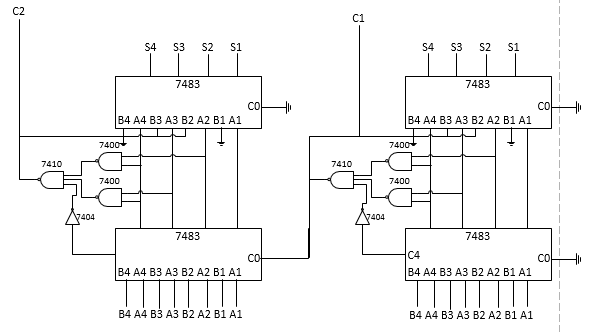
**-실험 8-2**



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 입력 A | | | | | 입력 B | | | | 출력 | | | | |
| M | A4 | A3 | A2 | A1 | B4 | B3 | B2 | B1 | C4 | S3 | S2 | S1 | S0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |

실험의 원리는 실험 8-1과 동일하다 M이 0일 때 XOR게이트는 다른 입력 값을 출력으로 가지는 특징을 가지고 있기 때문에 게이트가 없는 것과 같이 동작을 한다. 하지만 M이 1일 경우 B입력에 대해 보수가 출력 값을 가지게 되며 C는 감산일 때 1이 더해지게 된다. 그러므로 감산은 A의 값과 B의 보수의 값 그리고 1이 더해지게 되므로 결과 적으로 2의 보수 감산이 되는 것이다.

**-실험 8-3**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 10의 자리 | 1의 자리 | 자리 올림 | |
| 번  호 | A4 A3 A2 A1  + B4 B3 B2 B1  S4 S3 S2 S1 | A4 A3 A2 A1  + B4 B3 B2 B1  S4 S3 S2 S1 | C2 | C1 |
| (1) | 0 0 0 1  + 0 0 1 0  0 1 0 0 | 0 1 1 0  + 0 1 0 0  0 0 0 0 | 0 | 1 |
| (2) | 0 0 1 1  + 0 0 1 0  0 1 0 1 | 0 0 0 1  + 0 0 1 1  0 1 0 0 | 0 | 0 |
| (3) | 0 1 0 0  + 0 0 1 1  1 0 0 0 | 0 1 1 1  + 0 1 1 0  0 0 1 1 | 0 | 1 |

위의 회로는 두 자리 BCD병렬 가산기 실험회로 이다. 일의자리와 10의자리를 나누어 계산하고 1의자리에서 carry out이 발생하면 10의자리로 넘겨준다. 1의자리의 합이 10이상일 경우 2진수 0110을 더해주어 표현한 후 carry out을 발생시킨다. 결과표에 표시하지 않았지만 (1)번 (3)번의 경우에는 1의자리합이 9이상이므로 0110을 더해주어 표현하였다. 0110을 더해주는 이유는 4자리 2진수는 15까지 표현이 가능한데, 이 가산기는 BCD가산기 이므로 9까지만 표현되기 때문이다.

1. 고찰

실험8-1은 감산과 가산을 선택하여 연산을 한다. 가산 연산시에는 XOR 게이트에 입력 값 중 하나가 항상 0이되므로 B의 각 비트가 반전되지 않아서 보수가 되지않는다. 하지만 감산 연산 시 XOR 게이트를 입력 값 중 하나가 항상 1이되므로 B의 각 비트는 반전되고, 1을 추가로 더해주어 보수를 만든 후, B의 보수를 adder에 입력 값으로 하여 감산연산을 수행하였다.

실험 8-3 을 연결하는데 꽤 어려움이 있었다 특히 입력 값을 16개를 받게 되므로 입력 값을 따로 구분해서 기억하고 있어야 했고, 입력 값을 직접 VDD와 그라운드에 연결 하여 결과 값을 구하려면 회로를 계속 바꾸어 주어야하는 문제점이 있었다.