컴퓨터 공학 기초 설계 및 실험1

결과 보고서

실험제목 : Half-adder & full-adder Multiplexer and de -multiplexer

실험일자: 2018년 05월 24일 (목)

제출일자: 2018년 05월 31일 (목)

학 과: 컴퓨터정보공학부

담당교수: 이준환

실습분반: 목요일(0,1,2)

학 번: 2015722025

성 명: 정용훈

결과보고서

1. 제목 및 목적
   1. 제목

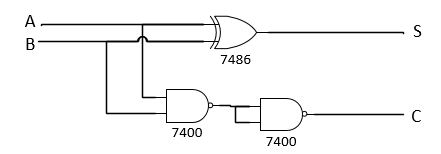
Half-adder & full-adder Multiplexer and de-multiplexer

* 1. 목적

반가산기, 반감산기, 전가산기와 전감산기의 구성과 동작 특성, 구성 원리를 이해 할 수 있고 가산기의 연산장치를 이해하는데 목적이 있다. 또한 먹스와 디먹스의 동작원리 및 특성을 이해 할 수 있다.

1. 실험 결과

**실험 6-2**

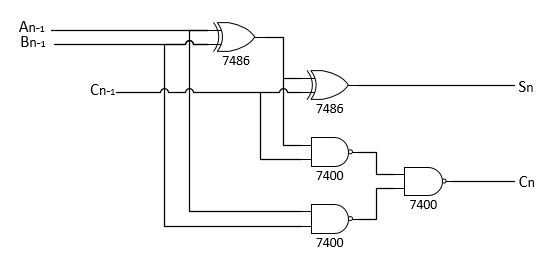


|  |  |  |  |
| --- | --- | --- | --- |
| 입력 | | 출력 | |
| A | B | S | C |
| 0 | 0 | 0 | 0 |
| 0 | +5 | 1 | 0 |
| +5 | 0 | 1 | 0 |
| +5 | +5 | 0 | 1 |

위 회로는 반가산기의 회로이다. 입력인 A,B를 합하여 1비트의 합을 S로 출력하며 A,B를 합산한 결과 올림수가 발생하는 경우 C의 값이 출력된다. 한계점으로는 그 전 자리의 올림수를 받을 수 없기에 추가적인 반가산기와 OR Gate가 있어야 여러비트의 가산이 가능하다. 위 표의 논리식은 다음과 같다.

S=AB, C=A∙B

**실험 7-1**

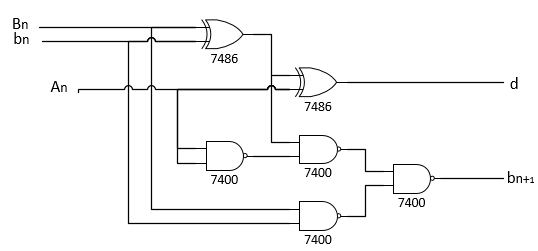


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 입력 | | | 출력 | |
| An | Bn | Cn-1 | Sn | Cn |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | +5 | 1 | 0 |
| 0 | +5 | 0 | 1 | 0 |
| 0 | +5 | +5 | 0 | 1 |
| +5 | 0 | 0 | 1 | 0 |
| +5 | 0 | +5 | 0 | 1 |
| +5 | +5 | 0 | 0 | 1 |
| +5 | +5 | +5 | 1 | 1 |

위 회로는 전가산기 회로이다. A,B는 마찬가지로 가산을 하는 입력 값이고, Cn-1은 그 전 자리에서의 올림수이다. 따라서 A,B,Cn-1을 모두 가산한 합을 Sn으로 출력하고 올림수가 발생하게 되면 Cn으로 값을 출력한다. 반 가산기와는 달리 그 전 자리수의 올림을 받을 수 있으므로 N개의 전가산기를 연결하여 N비트의 계산을 할 수 있다. 논리식은 아래와 같다.

Sn=AnBnCn-1, Cn=An∙Bn + (AnBn) ∙Cn-1

**실험 7-3**



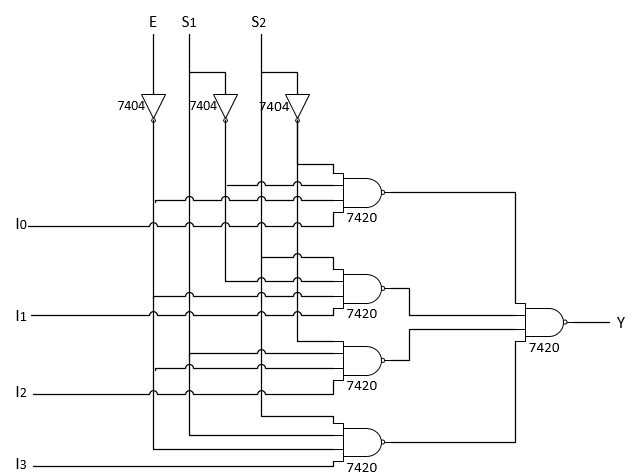
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 입력 | | | 출력 | |
| An | Bn | bn | bn+1 | d |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | +5 | 1 | 1 |
| 0 | +5 | 0 | 1 | 1 |
| 0 | +5 | +5 | 1 | 0 |
| +5 | 0 | 0 | 0 | 1 |
| +5 | 0 | +5 | 0 | 0 |
| +5 | +5 | 0 | 0 | 0 |
| +5 | +5 | +5 | 1 | 1 |

위 회로는 전감산기 회로이다. 입력값 An은 피감수, Bn은 감수이다. 계산을 할 때는 다음과 같은 방법으로 한다. “(피감수)-(감수)” , 아래 자리에서 발생한 내림수는 bn 이다. 입력 값을 감산한 결과인 차는 d로 출력되고 자리내림이 발생 할 경우 bn+1로 출력한다. 논리식은 다음과 같다.

bn+1=(Bnbn) ∙An’ bn

d=AnbnBn

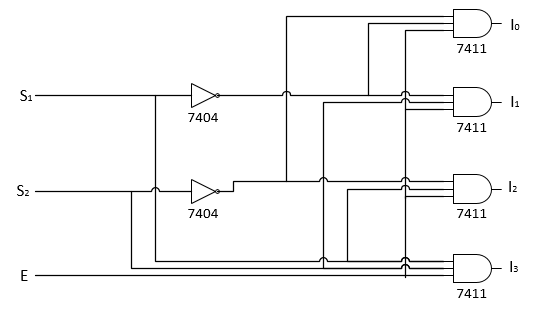
**실험 16-1**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 입력 | | | | | | | 출력 |
| F | S1 | S2 | I0 | I1 | I2 | I3 | Y |
| +5 | X | X | X | X | X | X | 0 |
| 0 | 0 | 0 | 0 | +5 | +5 | +5 | 0 |
| 0 | 0 | 0 | +5 | 0 | 0 | 0 | 1 |
| 0 | 0 | +5 | 0 | +5 | 0 | 0 | 1 |
| 0 | 0 | +5 | +5 | 0 | +5 | +5 | 0 |
| 0 | +5 | 0 | 0 | 0 | +5 | 0 | 1 |
| 0 | +5 | 0 | +5 | +5 | 0 | +5 | 0 |
| 0 | +5 | +5 | 0 | 0 | 0 | +5 | 1 |
| 0 | +5 | +5 | +5 | +5 | +5 | 0 | 0 |

4채널 멀티플렉서 회로이다. F값이 1일 때 작동하지 않고, 사용자가 입력한 S1,S2에 의하여 출력값이 I0, I1, I2, I3 중 결정된다. S1,S2=(0,0) 일때 I0가 출력되고, S1,S2=(0,1)일 때 I1이 출력되고, S1,S2=(1,0) 일때 I2가 출력되고, S1,S2=(1,1)일 때 I3이 출력된다.

**실험 16-3**



|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 입력 | | | 출력 | | | |
| E | S1 | S2 | I0 | I1 | I2 | I3 |
| 0 | X | X | 0 | 0 | 0 | 0 |
| +5 | 0 | 0 | 1 | 0 | 0 | 0 |
| +5 | 0 | +5 | 0 | 1 | 0 | 0 |
| +5 | +5 | 0 | 0 | 0 | 1 | 0 |
| +5 | +5 | +5 | 0 | 0 | 0 | 1 |

4채널 디멀티플렉서 회로이다. E값이 0일때는 작동을 하지 않는다. S1,S2=(0,0) 일때 I0만 1로 출력되고, S1,S2=(0,1)일 때 I1만 1로 출력되고, S1,S2=(1,0) 일때 I2만 1로 출력되고, S1,S2=(1,1)일 때 I3만 1로 출력되므로 2진수를 10진수로 바꾸어주는 디코더 역할을 한다고 볼 수 있다.

1. 고찰

그림의 회로도는 간단하게 연결 할 수 있을 것처럼 보였다. 하지만 실제로 회로를 연결 하는데 있어 상당히 복잡하게 선을 연결하게 되었다. 그럼으로써 다른 게이트에 들어가는 input을 잘못 연결하는 문제가 실험을 하는 중 종종 일어나게 되었다. Out put을 확인하며 디버깅을 통해 회로를 다시 연결하여 문제를 해결하게 되었다.

이론에서 배웠던 먹스와 디먹스의 차이점을 실험을 통해 좀더 확실히 알 수 있는 실험이 되었다 먹스는 사용자의 S입력값에 따라 출력 값이 결정되는 것을 알 수 있으며, 디먹스가 디코더의 역할을 할 수 있다는 것을 알게 되었다.