디지털 논리 실험 및 설계

DIGITAL LOGIC LAB & DESIGN

(학수번호: 013705)

Manual

v3.0

2021.02.15

전자전기공학부

홍익대학교

기본 사항 및 실험보고서 작성 요령

1. 실험 준비

디지털 논리 실험 및 설계는 2학년 1학기에 디지털 논리회로를 신청하여 수강하고 있는 학생(또는 이미 수강한 학생)을 대상으로 강의를 통해 이론적으로 배우는(또는 배운) 논리회로의 이론을 실험적으로 입증하고자 개설된 과목이다. 따라서 실험을 원활하게 수행하기 위해서는 해당 주제에 대한 논리회로의 이론을 미리 예습하여 내용을 잘 숙지한 상태로 실험에 임해야 한다. 일부 실험 결과가 이론 결과와 다르게 나올 수 있으므로 실험에 임하기 전에 이론적인 결과를 미리 파악하여 실험 결과가 이론결과와 다를 경우 그 원인에 대해 조원과 함께 토의하여 결과보고서를 준비하는 과정도 필요하다.

2. 조편성

[비대면 수업의 경우 조편성 없이 개인실험을 진행한다] 실험은 2인 1조를 원칙으로 한다. 원하는 경우 학생이 실험 조를 편성하여 조교의 승인을 받을 수 있다. 그러지 않은 경우 조교가 임의로 실험 조를 편성한다.

3. 보고서

보고서는 예비보고서만 작성한다.

3.1 예비보고서

예비보고서에는 '2. 실험 준비'에서 요구하는 내용이 포함되어야 한다. 또한 예비보고서에는 기본 및 응용실험에서 예상되는 이론적인 실험 결과 및 그 논리적인 이유가 포함되어야 한다. 이론적인 실험 결과는 복잡한 회로의 경우는 Logic Works를 이용하여 구하고 간단한 회로의 경우는 수업에서 배운 방법을 이용하여 간단하게 결과를 얻으면 된다. Logic Works 프로그램은 P동 PC실에 설치되어 있으며 실험에서 결선하는 회로들은 간단하여 동영상을 참조하여 배우면 사용 가능하다. (필요 시 조교에게 도움을 받을 것). 요약하면 예비보고서는 '실험 준비'에 대한 내용 및 예상되는 실험 결과를 간단하게 작성하여 제출한다. 예비보고서는 개인당 하나씩 작성하여 클래스 넷을 통하여 제출한다. 각 주차별 예비보고서의 만점은 10점이다. 변경된 매뉴얼을 확인하지 않고 이전 매뉴얼을 기준으로 작성한 보고서의 경우 부정행위로 간주하여 0점처리한다. 총 10주차 실험의 예비보고서 총합 100점을 20점으로 변환하여 (0.2를 곱하여) 최종성적에 포함한다.

보고서는 '조_학번_이름_주차_예비보고서'와 같이 파일제목을 작성한다. 클래스넷에 제출한 파일제목이 위 형식을 따르지 않은 경우 감점.

예1) 1조_B123456_노승문_1주차_예비보고서.hwp

예2) 12조_B123456_노승문_1주차_예비보고서.doc

비대면 수업이라 조편성을 하지 않는 경우 조번호 생략

예1) B123456_노승문_1주차_예비보고서.hwp
예2) B123456_노승문_1주차_예비보고서.doc

3.2 결과보고서

결과보고서는 작성하지 않는다.

4. 실험

[비대면 수업의 경우 실험 영상을 촬영후 youtube에 업로드하여 해당 링크를 제출하는 형태로 사전에 검사를 받을 수 있다.] 모든 기본 및 응용 실험은 수업시간 중 조교의 검사를 받는다. 검사를 받지 못한 실험에 대해서는 '실험점수'항목에서 감점이 있다. 예를 들어, 기본 및 응용 총 5개 실험 중 4개 성공하면 해당 주차 실험 점수의 80%를 부여한다. 해당 수업시간에 실험을 완료하지 못하였더라도 다음 시간에 실험에 성공할 경우 점수를 얻을 수 있다 (해당 실험의 다음 수업까지만 가능).

수강생들은 실험 후 사용한 칩을 원래 자리로 반납하여야한다. 망가져서 사용이 어려워진 부품은 버리도록 한다. 사용한 키트, 책상 위, 또는 서랍 안에 전선이나 칩을 넣어두지 않도록한다. 키트의 전원을 분리하고 사용한 멀티미터 등의 장비도 원래대로 정리해두도록 한다. 조교는 실험 후 각 조의 정리 상태에 따라 해당주차 실험점수에서 0%-30%를 감점한다.

5. 기타

5.1 강의동영상

본 실험은 이론수업과 같은 학기에 진행되어 이론수업보다 실험수업의 진도가 더 빠른 경우가 발생하기도 한다. 이에 매 실험의 내용에 해당하는 이론을 간략하게 강의하는 동영상을 업로드한다. 모든 수강생은 예비보고서 작성에 앞서 해당 동영상을 확인한다.

5.2 실험부품

실험 후 사용한 칩은 원래 자리로 반납한다. 망가져서 사용이 어려워진 부품은 버린다. 사용한 키트 안에 전선 혹은 칩을 넣어두지 않는다. 키트의 전원을 분리하고 사용한 멀티미터 등의 장비도 원래대로 정리해둔다. 조교는 실험 후 정리 상태에 따라 감점을 한다.

5.3 로직웍스

예비실험 및 프로젝트 진행 시 시뮬레이션 tool인 LogicWorks를 사용할 수 있다. P동 PC실에서 사용이 가능하며 '로직웍스'로 검색하면 다운로드 가능하다.

5.4 출석

실험수업은 출석하여 실제로 실험을 진행하는 것이 중요하므로 엄격한 기준을 적용한다. 3회 포함 3회 이상 결석의 경우 F학점을 부여한다. 10분 이상 1시간 이하 늦으면 지각처리 (0.5회

결석처리, 출석점수 -2점) 1시간 이상 늦으면 추가로 결석 0.75회 처리 (출석점수 -3점)한다. 결석의 경우 출석점수 4점을 감점한다. 출석점수 감점이 10점 이상이면 출석점수를 0점처리한다. 첫 주차 오리엔테이션 수업은 수강 정정기간이기 때문에 전원 출석 처리한다.

5.5 최종성적

학점은 출석 10점, 예비보고서 20점, 실험점수 30점, 프로젝트 40점으로 한다.

주별 수업계획서

주	실험 내용	비고
1	조 편성 및 실험 기본 요령 설명	
2	1주차: AND, OR, NOT Gate 실습 및 datasheet	실험 manual
3	2주차: NAND, NOR, XOR 및 응용	실험 manual
4	3주차: Decoder	실험 manual
5	4주차: Multiplexer	실험 manual
6	5주차: Adder	실험 manual
7	6주차: ALU	실험 manual
8	7주차: JK FlipFlop	실험 manual
9	8주차: D-FlipFlop	실험 manual
10	9주차: Shift Register	실험 manual
11	10주차: Counter	실험 manual
12	11주차: Project 설명 및 납땜	
13	12주차: Project 진행	
14	13주차: Project 진행	
15	14주차: Project 진행	

※ 작동 원리 및 이론은 디지털 논리회로 수업교재 및 Floyd의 Digital Fundamentals를 참조할 것

1주차:

1. 실험 목적

Datasheet를 보는 법을 익히고 AND, OR, NOT 등 기본 논리게이트에 대해 학습한다.

2. 실험 준비

- AND 게이트 7408의 datasheet를 읽는 법을 간단하게 서술하고 4.1 기본 실험 (1)의 회로를 어떻게 결선하여야 하는지 pin번호를 이용하여 설명하시오.
- VCC와 GND를 직접 결선(직접연결)하면 안되는 이유를 설명하시오.
- 기본 실험 (4)의 회로를 구현하시오.
- 응용 실험 (2)의 회로를 구현하시오.
- 응용 실험 (3)의 결과를 예상해보고 본인의 생각을 서술하시오.

3. 기기 및 부품

NOT 게이트 7404, OR 게이트 7432, AND 게이트 7408

4. 실험 과정

4.1 기본 실험

(1) [그림 1]과 같이 회로를 결선한 후 AND 게이트의 진리표를 그려보시오.

[그림 1]

A	B	X
0	0	
0	1	
1	0	
1	1	

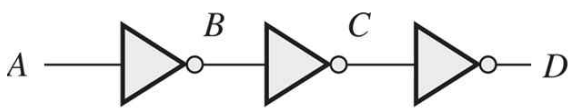
※ 결선 시 VCC와 GND를 꼭 연결하도록 한다.

(2) [그림 2]와 같이 회로를 결선한 후 OR 게이트의 진리표를 그려보시오.

[그림 2]

A	B	X
0	0	
0	1	
1	0	
1	1	

(3) [그림 3]과 같이 회로를 결선하시오. 아래 표에 그 결과를 기록하시오.



[그림 3]

A	B	C	D
0			
1			

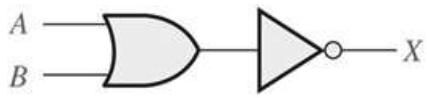
(4) AND 게이트를 두 개 사용하여 3입력 AND 게이트를 구현하시오. 즉, $X=ABC$ 의 회로를 구현하시오. 아래 표에 그 결과를 기록하시오.

A	B	C	X
0	0	0	
0	1	0	
1	0	1	
1	1	1	

※ 스위치를 이용하면 입력을 간편하게 변경할 수 있다.

4.2 응용 실험

(1) [그림 4]와 같이 회로를 결선하시오. 아래 표에 그 결과를 기록하시오.



[그림 4]

A	B	X
0	0	
0	1	
1	0	
1	1	

(2) $X=\overline{A}B+A\overline{B}$의 회로를 NOT, AND, OR 게이트를 이용하여 구현하고 아래 표에 그 결과를 기록하시오.

A	B	X
0	0	
0	1	
1	0	
1	1	

(3) 위 기본 실험 (1)의 AND 게이트를 VCC와 GND를 제외하고 결선하시오. 실험으로 나타난 결과를 아래 표에 기록하시오.

A	B	X
0	0	
0	1	
1	0	
1	1	

2주차:

1. 실험 목적

NAND, NOR, XOR 등 논리게이트의 특성 및 응용에 대해 학습한다.

2. 실험 준비

- NAND 게이트 7400, NOR 게이트 7402, XOR 게이트 7486의 datasheet를 확인하시오. **(1주차 이후에는 pin번호를 자세히 설명할 필요 없음)**
- 기본 실험 (4)의 회로를 구현하시오.
- 응용 실험 (1), (2), (3)의 회로를 구현하시오.
- 응용 실험 (3)의 결과를 예상하고 이유를 서술하시오. (힌트: 드모르간 법칙).

3. 기기 및 부품

NAND 게이트 7400, NOR 게이트 7402, XOR 게이트 7486

4. 실험 과정

4.1 기본 실험

(1) [그림 1]과 같이 회로를 결선한 후 NAND 게이트의 진리표를 그려보시오.

[그림 1]

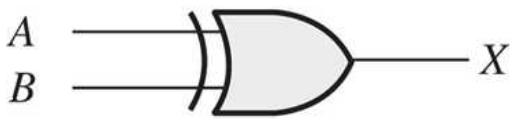
A	B	X
0	0	
0	1	
1	0	
1	1	

(2) [그림 2]와 같이 회로를 결선한 후 NOR 게이트의 진리표를 그려보시오.

[그림 2]

A	B	X
0	0	
0	1	
1	0	
1	1	

(3) [그림 3]과 같이 회로를 결선하시오. 아래 표에 그 결과를 기록하시오.



[그림 3]

A	B	X
0	0	
0	1	
1	0	
1	1	

(4) NAND 게이트를 두 개 사용하여 AND 게이트를 구현하시오. 즉, $X=AB$ 의 회로를 구현하시오. 실험을 통해 이를 확인하고 아래 표에 그 결과를 기록하시오.

A	B	X
0	0	
0	1	
1	0	
1	1	

4.2 응용 실험

(1) NAND게이트 4개만을 이용하여 XOR게이트를 구현하고 아래 표에 그 결과를 기록하시오.

A	B	X
0	0	
0	1	
1	0	
1	1	

※ 힌트: A XOR $B = A(\overline{A}+\overline{B})+B(\overline{A}+\overline{B})$

(2) A, B, C, D 네 개의 입력을 받는 패리티 확인(parity check)회로를 XOR게이트 3개로 구현하시오. 패리티 확인 회로는 네 개의 입력 중 HIGH(1)이 짝수개면 LOW(0)을, HIGH(1)이 홀수개면 HIGH(1)을 출력하는 회로이다. 실험을 통해 이를 확인하고 아래 표에 그 결과를 기록하시오.

A	B	C	D	X
0	0	0	0	
0	1	0	1	
1	0	1	1	
1	1	1	1	

(3) NOR 게이트를 세 개 사용하여 $X=\overline{\overline{A+B}+\overline{A+C}}$ 의 회로를 구현하시오. 실험을 통해 이 회로가 $X=A+BC$와 같은 회로임을 확인하고 아래 표에 그 결과를 기록하시오.

A	B	C	X
0	0	0	
0	1	0	
1	0	1	
1	1	1	

3주차:

1. 실험 목적

부호기(encoder)와 복호기(decoder)의 동작 원리 및 특성을 살펴본다.

2. 실험 준비

- 기본 실험 (1)의 회로가 2-bit 복호기인 이유를 설명하시오.
- 기본 실험 (2)의 회로가 2-bit 부호기인 이유를 설명하시오.
- 기본 실험 (3)에서 7을 표시하기 위한 입력 $ABCD$가 무엇인지 설명하시오.
- BCD to 7-segment decoder 7447과 7-segment 5161의 datasheet를 확인하시오.
- $\overline{LT}$ 기능에 대해 서술하시오.
- Active LOW와 Active HIGH에 대하여 설명하시오.

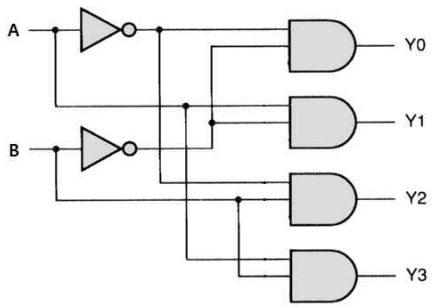
3. 기기 및 부품

BCD to 7-segment decoder 7447, 7-segment 5161, NOT 게이트 7404, OR 게이트 7432, AND 게이트 7408

4. 실험 과정

4.1 기본 실험

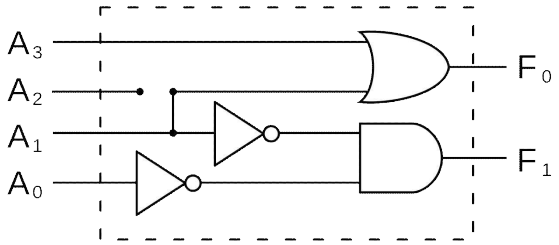
(1) [그림 1]과 같이 2-bit 복호기를 결선한 후 아래 진리표를 그려보시오.



[그림 1]

B	A	Y0	Y1	Y2	Y3
0	0				
0	1				
1	0				
1	1				

(2) [그림 2]과 같이 2-bit 부호기를 결선한 후 아래 진리표를 그려보시오.



A3	A2	A1	A0	F1	F0
0	0	0	1		
0	0	1	0		
0	1	0	0		
1	0	0	0		

[그림 2]

※ 응용실험 (1)와 함께 진행하시오.

(3) 주어진 기판의 7-segment에 숫자 7을 표시하시오.

※ 7-segment decoder 7447를 사용하지 말고 기판의 A, B, C, D에 직접 결선한다.

4.2 응용 실험

(1) [그림 2]의 부호기 회로에 두 개 이상의 입력이 1인 경우 중 한 가지를 골라 어떤 일이 일어나는지 실험해보시오.

(2) BCD to 7-segment decoder 7447을 이용하여 7-segment 5161에 숫자 7를 표시하시오.

※ 7-segment 5161을 오래 사용 시 과열될 수 있으니 주의하시오.

(3) BCD to 7-segment decoder 7447의 $\overline{LT}$기능을 테스트 해보시오.

(4) BCD to 7-segment decoder 7447에 $ABCD=1011$을 입력으로 넣었을 때 나오는 결과를 확인 해보시오. 기판의 7-segment에 같은 입력을 넣었을 때의 결과와 비교하시오. **[비대면 수업시 진행하지 않음]**

4주차:

1. 실험 목적

멀티플렉서(multiplexer)와 디멀티플렉서(demultiplexer)의 동작 원리 및 특성을 살펴본다.

2. 실험 준비

- 멀티플렉서와 부호기(encoder)의 차이를 설명하시오.
- 4-to-1 Multiplexer 74153, 2-to-1 Multiplexer 74157, 1-of-4 Decoder 74139, 3-INPUT AND 게이트 7411의 datasheet를 확인하시오.
- 4-to-1 Multiplexer 74153의 EN에 대해 설명하시오.
- 1-of-4 Decoder 74139가 Decoder와 Demultiplexer의 기능을 동시에 할 수 있음을 설명하시오. 이를 이용하여 기본 실험 (2)를 어떻게 결선할 수 있는지 설명하시오.
- 응용 실험 (1)이 8-to-1 멀티플렉서로 동작하는 원리를 자세히 설명하시오.
- 응용 실험 (2)가 4-to-1 멀티플렉서로 동작하는 원리를 자세히 서술하시오.

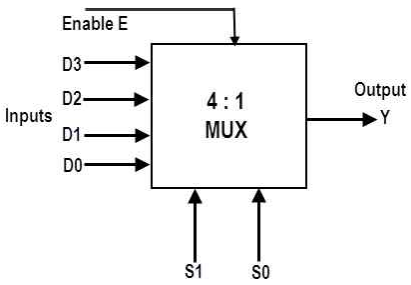
3. 기기 및 부품

4-to-1 Multiplexer 74153, 2-to-1 Multiplexer 74157, 1-of-4 Demultiplexer 74139, 3-INPUT AND 게이트 7411, NOT 게이트 7404, OR 게이트 7432

4. 실험 과정

4.1 기본 실험

(1) 4-to-1 Multiplexer 74153을 결선하여 아래 진리표를 작성하시오.

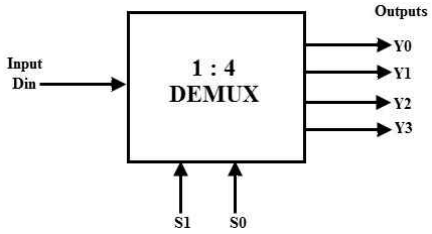


[그림 1]

$\overline{E}$	S1	S0	D3	D2	D1	D0	Y
1	1	1	1	1	0	0	
1	1	0	0	1	0	1	
0	0	1	0	0	1	1	
0	0	1	1	0	0	0	
0	1	0	0	1	1	1	
0	1	1	1	1	0	0	

※ 74153의 EN은 Active-LOW임을 유의하시오. [위 표역시 $\overline{E}$를 기준으로 작성되었음.]

(2) 1-to-4 Decoder 74139를 이용하여 Active-LOW 디멀티플렉서를 구현하고, 아래 진리표를 작성하시오. [힌트: EN을 INPUT처럼 사용하시오.]

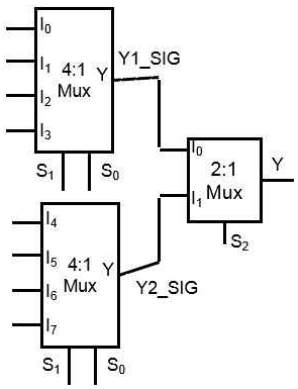


[그림 2]

I	S1	S0	Y0	Y1	Y2	Y3
0	0	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

4.2 응용 실험

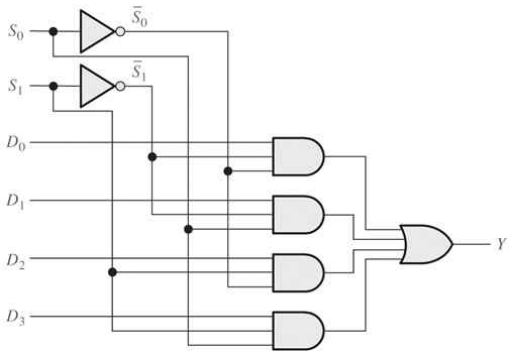
(1) [그림 3]과 같이 4-to-1 Multiplexer 74153와 2-to-1 Multiplexer 74157을 이용하여 8-to-1 멀티플렉서를 결선하시오. 실험을 통해 8-to-1 멀티플렉서가 잘 동작하는지 확인하시오.



[그림 3]

※ VCC, GND, EN을 모두 연결하였는지 확인하시오.

(2) [그림 4]와 같이 4-to-1 Multiplexer를 AND, OR, NOT 게이트를 이용하여 구현하시오. 그 결과가 기본 실험 (1)과 일치하는지 확인하시오. 3-INPUT AND 게이트 7411을 사용하시오.



[그림 4]

5주차:

1. 실험 목적

이진 덧셈의 원리를 이해하고 반가산기(half adder)와 전가산기(full adder)의 동작을 확인한다.

2. 실험 준비

- 4.1 기본 실험 (2)의 전가산기 [그림 2]는 반가산기 [그림 1] 두 개와 하나의 OR 게이트로 이루어져 있다. [그림 2]의 회로가 전가산기로 동작하는 원리를 설명하시오.
- 응용 실험 (1), (2)의 회로를 구현하시오.
- LSB와 MSB의 의미를 조사하시오.
- 응용 실험 (2)의 회로를 순차적으로 연결하는 방식에 대해 자세히 서술하시오.

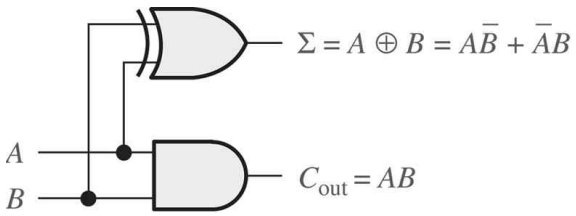
3. 기기 및 부품

AND 게이트 7408, OR 게이트 7432, XOR 게이트 7486

4. 실험 과정

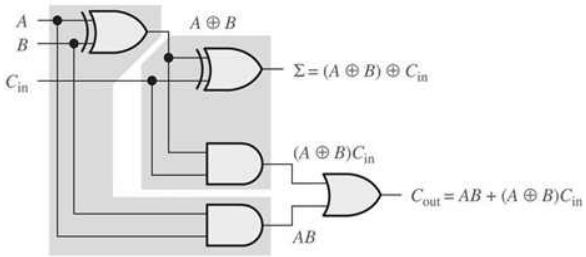
4.1 기본 실험

(1) [그림 1]과 같이 반가산기를 결선한 후 아래 진리표를 그려보시오.



A	B	C_{out}	Σ
0	0		
0	1		
1	0		
1	1		

(2) [그림 2]와 같이 전가산기를 결선한 후 아래 진리표를 그려보시오.

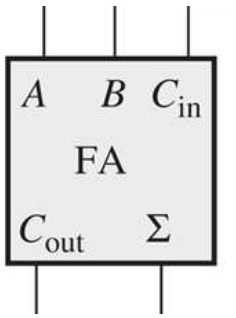


[그림 2]

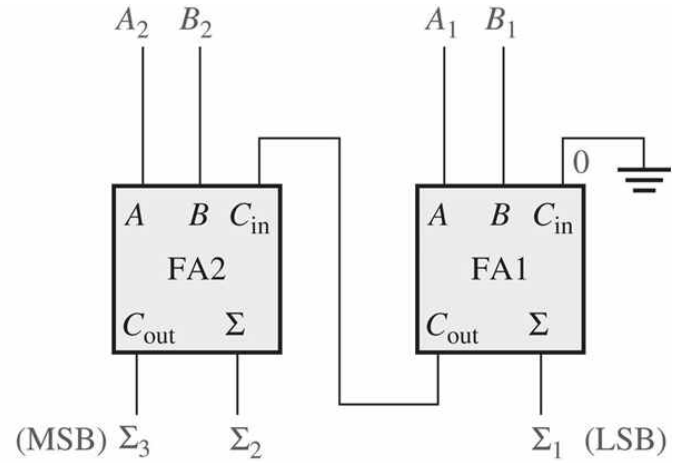
A	B	C_{in}	C_{out}	Σ
0	0	1		
0	1	0		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

4.2 응용 실험

(1) 전가산기 두 개를 직렬로 연결하여 두 자리 이진수 덧셈기를 구현하시오. 위 [그림 2]를 아래 [그림 3]과 같이 도식화 하였을 때, 두 자리 이진수 덧셈기는 [그림 4]와 같이 구현할 수 있다.



[그림 3]



[그림 4]

[그림 4]의 회로를 XOR, AND, OR 게이트로 구현한 후 아래 표에 실험 결과를 기록하시오.

A_2	A_1	B_2	B_1	Σ_3	Σ_2	Σ_1
0	1	1	0			
0	1	1	1			
1	1	1	1			
0	0	0	1			

※ 한 번에 모든 회로를 연결하지 않는다. 전가산기를 한 개 구현할 때마다 각각의 동작을 확인하고 모든 전가산기가 동작할 때 연결한다.

(2) 위 회로를 이용하여 (새로운 전가산기를 추가하지 말고) $1_{(2)}+11_{(2)}+10_{(2)}$을 계산하시오.

※ $11_{(2)}+11_{(2)}$을 직접 계산할 경우 감점.

6주차:

1. 실험 목적

4-bit 논리연산장치 (ALU: Arithmetic Logic Unit)에 대해 이해한다.

2. 실험 준비

- ALU 74181의 datasheet을 읽고 네 자리 이진수의 덧셈을 74181을 이용하여 어떻게 구현할 수 있는지 설명하시오.
- 74181을 이용하여 두 개의 네 자리 이진수가 같은지 판별하는 방법을 설명하시오.
- 이진수의 뺄셈을 어떻게 구현하는지 설명하시오.
- 응용실험 (1)의 연산을 ALU를 이용하여 어떻게 계산하는지 서술하시오.
- 응용실험 (2)에서 다루는 AB minus 1 기능에 대해 설명하시오.

3. 기기 및 부품

ALU 74181

4. 실험 과정

4.1 기본 실험

(1) ALU를 이용하여 $1011_{(2)}+1001_{(2)}$ (OR 가 아닌 실제 덧셈)을 계산하시오.

(2) ALU의 $A \oplus B$ (XOR)기능을 이용하여 $1011_{(2)}$과 $0101_{(2)}$이 같지 않음을 보이시오. 또, $1011_{(2)}$과 $1011_{(2)}$이 같음을 보이시오.

※ $A=B$기능을 직접 사용하지말고 XOR기능을 이용하시오.

4.2 응용 실험

(1) ALU를 이용하여 $1011_{(2)}-0101_{(2)}-1_{(2)}$을 계산하시오.

※ ALU의 주어진 기능들을 확인하시오.

(2) ALU를 이용하여 AB minus 1 기능을 테스트해보시오. 두 가지 이상의 입력으로 해당 기능이 잘 동작하는지 확인하시오.

7주차:

1. 실험 목적

S-R Latch와 J-K Flip-flop의 동작 원리를 살펴본다.

2. 실험 준비

- S-R Latch와 $\overline{S}-\overline{R}$ Latch의 동작에 대해 설명하시오.
- Pulse detector와 CLK에 대해 설명하고 응용 실험 (2) [그림 4]의 원리를 설명하시오.
- J-K Flip-flop의 동작에 대해 설명하시오.
- $\overline{PRE}$와 $\overline{CLR}$에 대해 설명하시오.
- J-K Flip-flop 7476, 3-INPUT NAND 7410의 datasheet를 확인하시오.
- 응용실험 (2)의 pulse transition detector에서 사용하는 NOT gate의 개수가 늘어나면서 발생하는 차이에 대하여 서술하시오.
- 실제 실험에서 스위치 입력을 한번만 주었는데 결과가 여러번 바뀌는 현상 (채터링)이 발생할 수 있다. 이에 대해 설명하시오.

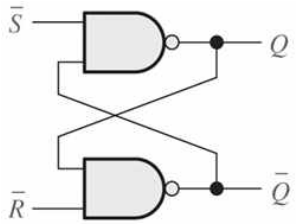
3. 기기 및 부품

J-K Flip-flop 7476, 3-INPUT NAND 7410, NOT 게이트 7404, NAND 게이트 7400, NOR 게이트 7402

4. 실험 과정

4.1 기본 실험

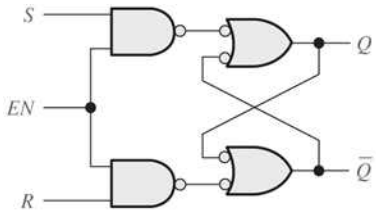
(1) [그림 1]과 같이 $\overline{S}-\overline{R}$ Latch를 결선하고, 아래 진리표를 완성하시오.



[그림 1]

$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		

(2) [그림 2]와 같이 Gated S-R Latch를 결선하고, 아래 진리표를 완성하시오. EN이 0인 상태에서 S와 R을 변화시켜본다.

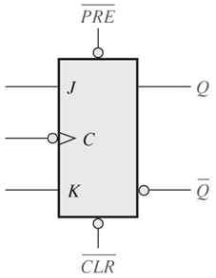


[그림 2]

EN	S	R	Q	$\overline{Q}$
0	0	0		
0	0	1		
0	1	0		
1	0	0		
1	1	0		
1	0	1		

※ 위 네 개의 게이트는 모두 NAND게이트이다.

(3) [그림 3]의 J-K Flip-flop 7476을 결선하고, 아래 진리표를 완성하시오. $\overline{PRE}$와 $\overline{CLR}$에 주의하시오. 아래의 여섯 가지 입력을 순서대로 실행하여 본다. 실험은 J, K를 먼저 설정한 후 CLK을 변화시키는 방법으로 수행한다.



[그림 3]

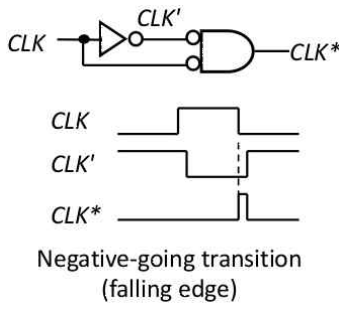
CLK	J	K	Q	$\overline{Q}$
↑ ↓	0	0		
↑ ↓	1	0		
↑ ↓	1	0		
↑ ↓	0	1		
↑ ↓	1	1		
↑ ↓	0	0		

※ Datasheet에서는 $\overline{PRE}$를 S(Set)로 $\overline{CLR}$를 C(Clear)로 CLK를 CP로 표기하였다. $\overline{PRE}$와 $\overline{CLR}$에 1을 입력으로 준다. VCC와 GND도 잊지 말고 결선한다. 초기 Q값은 0으로 가정한다.

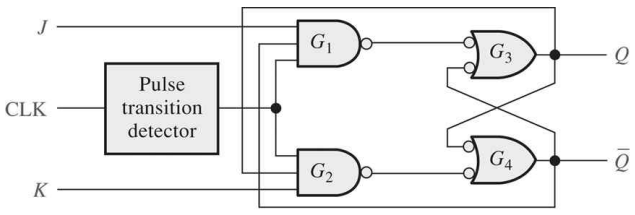
4.2 응용 실험

(1) [그림 3]의 J-K Flip-flop에 $\overline{PRE}$와 $\overline{CLR}$를 연결하지 말고 다시 실험하여 진리표를 작성하시오.

(2) [그림 4]의 Pulse transition detector를 이용하여 [그림 5]와 같이 J-K Flip-flop을 직접 구현하고 기본 실험 (3)의 결과와 비교하시오. Pulse transition detector가 제대로 작동하지 않는다면 NOT gate를 늘려가며 실험한다.



[그림 4]



[그림 5]

8주차:

1. 실험 목적

D Latch와 D Flip-flop의 동작 원리를 살펴본다.

2. 실험 준비

- Gated D Latch의 동작에 대해 설명하시오.
- D Flip-flop의 동작에 대해 설명하시오.
- D Flip-flop 7474의 datasheet를 확인하시오.
- T Flip-flop의 동작에 대해 설명하시오.
- 응용실험 (1)과 응용실험 (2)의 회로를 비교하시오.
- D Flip-flop이 아닌 J-K Flip-flop으로 응용실험 (1)의 회로 [그림 3]과 동일한 기능의 회로를 구현하시오.
- D Flip-flop이 아닌 J-K Flip-flop으로 응용실험 (2)의 회로 [그림 4]와 동일한 기능의 회로를 구현하시오.

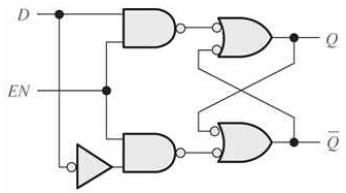
3. 기기 및 부품

D Flip-flop 7474, NOT 게이트 7404, NAND 게이트 7400, XOR 게이트 7486

4. 실험 과정

4.1 기본 실험

(1) [그림 1]과 Gated D Latch를 결선하고, 아래 진리표를 완성하시오.

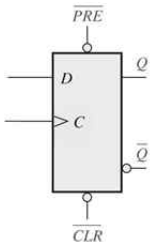


[그림 1]

EN	D	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		

※ 초기 Q값은 0으로 가정한다.

(2) [그림 2]의 D Flip-flop 7474를 결선하고, 아래 진리표를 완성하시오. $\overline{PRE}$와 $\overline{CLR}$에 주의하시오. 아래의 여섯 가지 입력을 순서대로 실행해보시오.



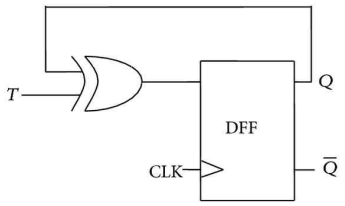
[그림 2]

CLK	D	Q	$\overline{Q}$
↑ ↓	0		
↑ ↓	0		
↑ ↓	1		
↑ ↓	0		
↑ ↓	1		
↑ ↓	1		

※ Datasheet에서는 $\overline{PRE}$를 S(Set)로 $\overline{CLR}$를 C(Clear)로 CLK를 CP로 표기하였다. VCC와 GND도 잊지 말고 결선한다. 실험은 D 값을 먼저 설정한 후 CLK을 변화시키는 방법으로 수행한다. 초기 Q값은 0으로 가정한다.

4.2 응용 실험

(1) [그림 3]과 같이 D Flip-flop을 이용하여 T Flip-flop을 구현하시오. 실험을 통해 아래 진리표를 작성하시오.

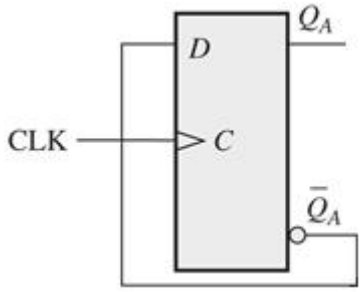


[그림 3]

CLK	T	Q	$\overline{Q}$
↑ ↓	0		
↑ ↓	0		
↑ ↓	1		
↑ ↓	0		
↑ ↓	1		
↑ ↓	1		

※ 위 그림에서 DFF는 D Flip-flop을 의미한다. $\overline{PRE}$, $\overline{CLR}$, VCC, GND도 잊지 말고 결선한다. 초기 Q값은 0으로 가정한다.

(2) [그림 4]와 같이 회로를 구현하고, 실험을 통해 아래 진리표를 작성하시오.



[그림 4]

CLK	Q	$\overline{Q}$
↑ ↓		
↑ ↓		
↑ ↓		
↑ ↓		

※ 위 회로는 CLK이외의 입력이 없다. $\overline{PRE}$, $\overline{CLR}$, VCC, GND도 잊지 말고 결선한다. 초기 Q값은 0으로 가정한다.

9주차:

1. 실험 목적

시프트 레지스터 (Shift register)의 동작 특성을 확인하고 기본적인 카운터를 구현한다.

2. 실험 준비

- 8-bit Serial-in Parallel-out Shift Register 74164의 datasheet를 확인하고 $\overline{MR}$의 역할에 대하여 설명하시오. 왜 입력이 A와 B로 나누어져 있는지 설명하시오.
- 존슨 카운터와 링 카운터에 대하여 설명하시오.
- 응용실험 (2)에서 초기화 하는 과정을 자세히 서술하시오.

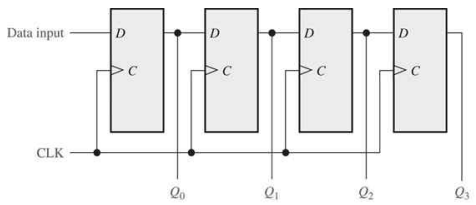
3. 기기 및 부품

8-bit Serial-in Parallel-out Shift Register 74164, D Flip-flop 7474, NOT 게이트 7404

4. 실험 과정

4.1 기본 실험

(1) [그림 1]과 같이 D Flip-flop을 이용하여 4-bit Serial-in Parallel-out 시프트 레지스터를 구현하시오. 초기 Q_0-Q_3값을 기록하고 이후 CLK를 변화시켜가며 나온 결과를 아래 표에 기록하시오.



[그림 1]

CLK	D	Q_0	Q_1	Q_2	Q_3
↑ ↓	1				
↑ ↓	0				
↑ ↓	1				
↑ ↓	0				
↑ ↓	1				
↑ ↓	1				

※ $\overline{PRE}$, $\overline{CLR}$, VCC, GND도 잊지 말고 결선한다. 응용실험 (1), (2)를 이어서 진행한다. 초기 Q값들은 모두 0으로 가정한다. 매 실험에서 새로운 D값을 설정후 CLK를 변화시켜야 한다.

(2) 8-bit Serial-in Parallel-out Shift Register 74164를 이용하여 8-bit 시프트 레지스터를 구현하고 동작을 확인하시오. CLK를 변화시켜가며 나온 결과를 아래 표에 기록하시오.

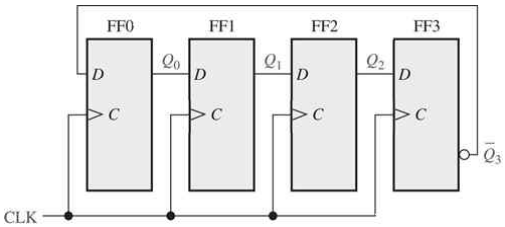
CLK	D	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
↑	1								
↑	0								
↑	1								
↑	0								
↑	1								
↑	1								

※ $\overline{MR}$도 잊지 말고 결선한다. 초기 Q값들은 모두 0으로 가정한다. 응용실험 (3)을 이어서 진

행한다.

4.2 응용 실험

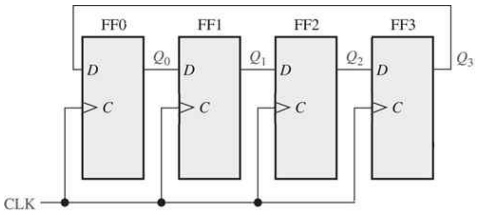
(1) [그림 2]와 같이 4-bit 존슨 카운터를 구현하시오. 실험 전 모든 Q_0-Q_3 값이 LOW(0)임을 확인 한 후 진행하시오.



[그림 2]

CLK	Q_0	Q_1	Q_2	Q_3
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				

(2) [그림 3]과 같이 링 카운터를 구현하시오.



[그림 3]

CLK	Q_0	Q_1	Q_2	Q_3
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				

※ 실험 전 Q_0는 HIGH(1), 그 외의 Q_1-Q_3은 LOW(0)임을 확인 후 진행한다. 이러한 초기값은 D Flip-flop의 $\overline{PRE}$기능을 이용하여 구현한다.

(3) 8-bit Serial-in Parallel-out Shift Register 74164를 이용하여 8-bit 존슨 카운터를 구현하고 아래 표를 완성하시오.

CLK	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
↑ ↓								
↑ ↓								
↑ ↓								
↑ ↓								
↑ ↓								
↑ ↓								
↑ ↓								
↑ ↓								

※ $\overline{MR}$을 이용하여 초기 Q값들을 모두 LOW(0)으로 만든 후 진행한다.

10주차:

1. 실험 목적

동기식(synchronous), 비동기식(asynchronous) 카운터(counter)에 대하여 공부한다.

2. 실험 준비

- 비동기식 카운터와 동기식 카운터의 작동원리와 차이점에 대하여 서술하시오.
- positive edge triggered D Flip-flop인 7474를 이용하여 [그림 1]의 회로를 어떻게 결선할지 설명하시오.
- 응용실험 (1)이 십진 카운터로 동작하는 원리에 대하여 서술하시오.
- 응용실험 (2)가 십진 카운터로 동작하는 원리에 대하여 서술하시오.

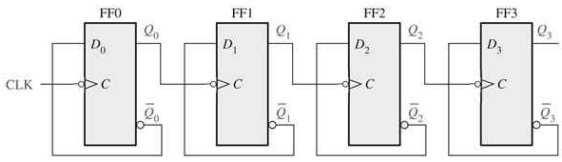
3. 기기 및 부품

D Flip-flop, J-K Flip-flop, OR 게이트 7432, AND 게이트 7408, NAND 게이트 7400

4. 실험 과정

4.1 기본 실험

(1) [그림 1]과 같이 4-bit 비동기식 카운터를 구현하시오.

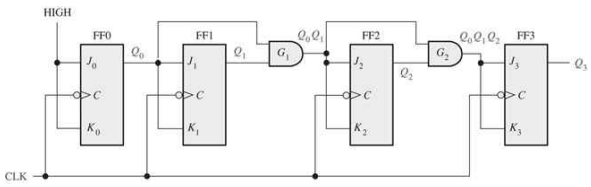


[그림 1]

CLK	Q_0	Q_1	Q_2	Q_3
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				

※ 실험 전 모든 Q_0-Q_3 값이 LOW(0)로 설정 후 진행한다. 응용실험(1)을 이어서 진행한다.

(2) [그림 2]와 같이 4-bit 동기식 카운터를 구현하시오.



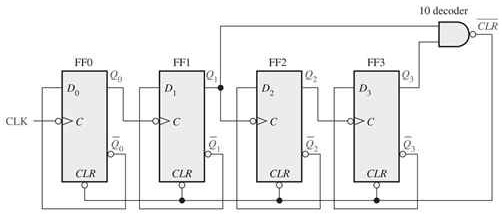
[그림 2]

CLK	Q_0	Q_1	Q_2	Q_3
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				

※ 실험 전 Q_0-Q_3 값을 LOW(0)로 설정 후 진행한다. 응용실험 (2)를 이어서 진행한다.

4.2 응용 실험

(1) [그림 3]과 같이 4-bit 비동기식 십진 카운터를 구현하시오.

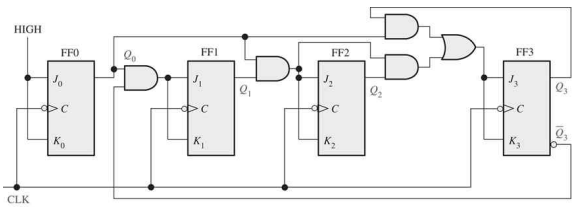


[그림 3]

CLK	Q_0	Q_1	Q_2	Q_3
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				

※ 실험 전 $Q_0 = 1, Q_1 = 1, Q_2 = 1, Q_3 = 0$으로 설정 후 진행한다.

(2) [그림 4]과 같이 4-bit 동기식 십진 카운터를 구현하시오.



[그림 4]

CLK	Q_0	Q_1	Q_2	Q_3
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				
↑ ↓				

※ 실험 전 $Q_0 = 1, Q_1 = 1, Q_2 = 1, Q_3 = 0$으로 설정 후 진행한다.