12주차 예비보고서

전공: 컴퓨터공학과/생명과학과 학년: 9학기 학번: 20181435 이름: 박다희

**1.**

Counter는 두 개 이상의 flip-flop로 이루어져 있는 clock 펄스마다 정해진 순서대로 상태 변화가 나타나는 논리 회로이다. Counter는 계수기라고도 하는데 숫자가 반복 증가하기 때문인데 decoder을 통해 7-segment 발광 다이오드와 같은 곳에 숫자로 표현된다. 속도, 출력 또는 잡음 특성 등에 따라 여러 종류의 counter을 사용한다. Counter의 예시로는 먼저 binary ripple counter을 들 수 있다. 이진 리플 카운터는 JK flip-flop이나 T flip-flop과 같이 보수로 만드는 역할의 flip-flop이 직렬 연결되어 있는데 이러한 구성으로 인해 각각의 flip-flop 출력 값들이 다음 flip-flop의 입력 값으로 들어가게 된다.

테이블이(가) 표시된 사진

자동 생성된 설명

위의 그림에서 볼 수 있듯이 clock 펄스는 가장 낮은 자리의 flip-flop의 입력 값으로 들어간다. 보수를 취하는 조건은 다음과 같다.

테이블이(가) 표시된 사진

자동 생성된 설명

위의 표에 따르면 가장 낮은 자리의 A1은 clock 펄스마다 보수가 되고, A1의 값이 1의 값에서 0으로 변하면 A2의 값이 보수로 바뀌고 A2의 값이 1에서 0으로 바뀌면 A3의 값이 보수로 바뀐다. 예를 들어서 0111의 값이 1000으로 바뀌는 과정을 살펴보자. A1의 값이 clock 펄스에 의해 1에서 0으로 보수 값을 취하게 되면 A2가 보수를 갖게 되도 이로 인해 차례대로 A3, A4의 값도 보수 값을 갖게 한다. 이때 A4의 경우에는 출력 값이 마지막이기 때문에 0에서 1로 바뀌게 된다. 따라서 최종적으로 1000의 출력 값을 갖게 된다.

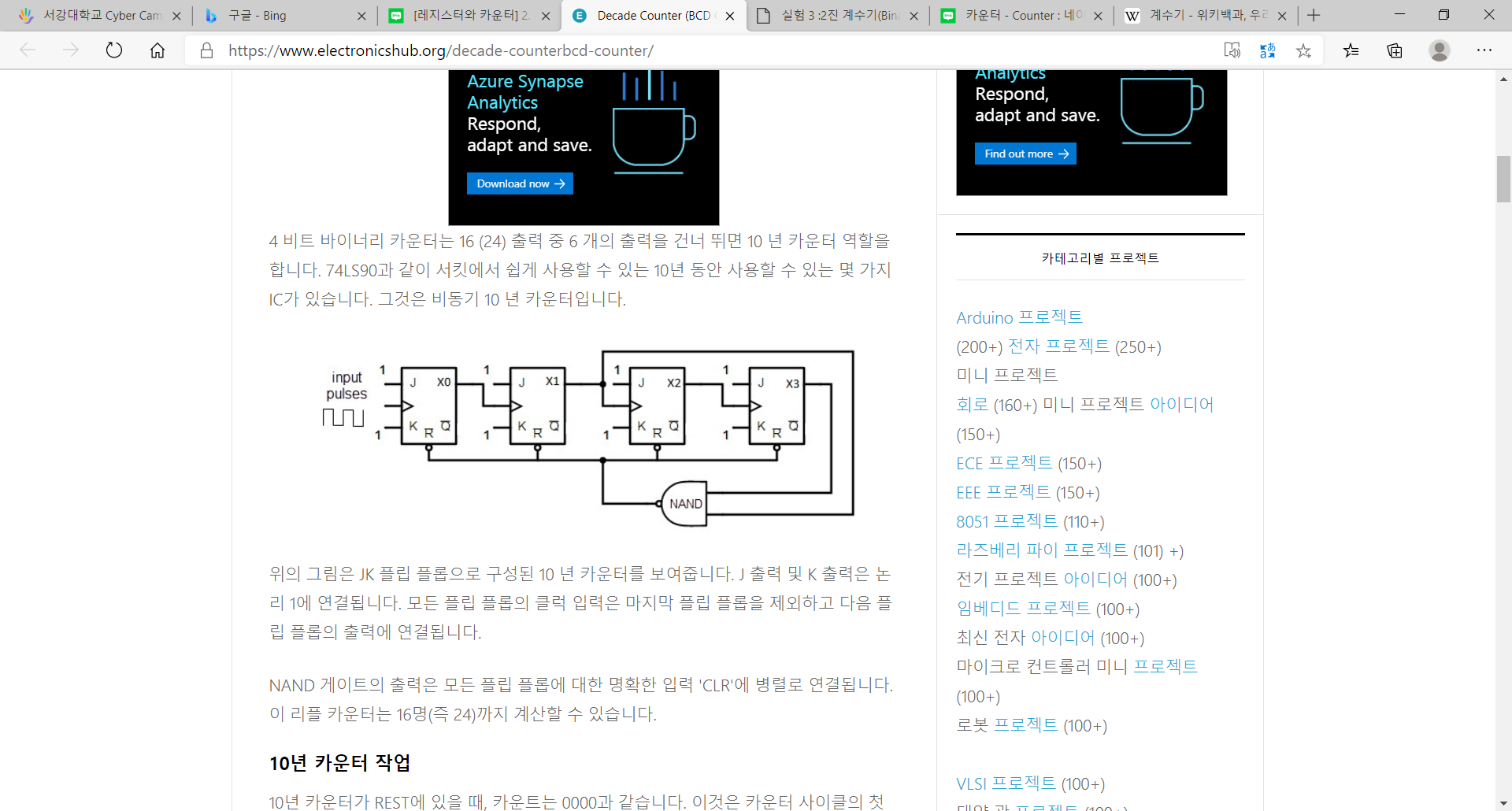
**2.**

Decade counter란 앞서 설명한 counter에서 2진수 대신 10진수로 계산하는 counter로 BCD counter고 불리기도 한다. 상태의 가짓수는 총 10개이며 0부터 9까지의 상태를 가지기 때문에 0000부터 1001까지의 값을 갖는다.

텍스트, 스크린샷, 소프트웨어, 웹 페이지이(가) 표시된 사진

자동 생성된 설명

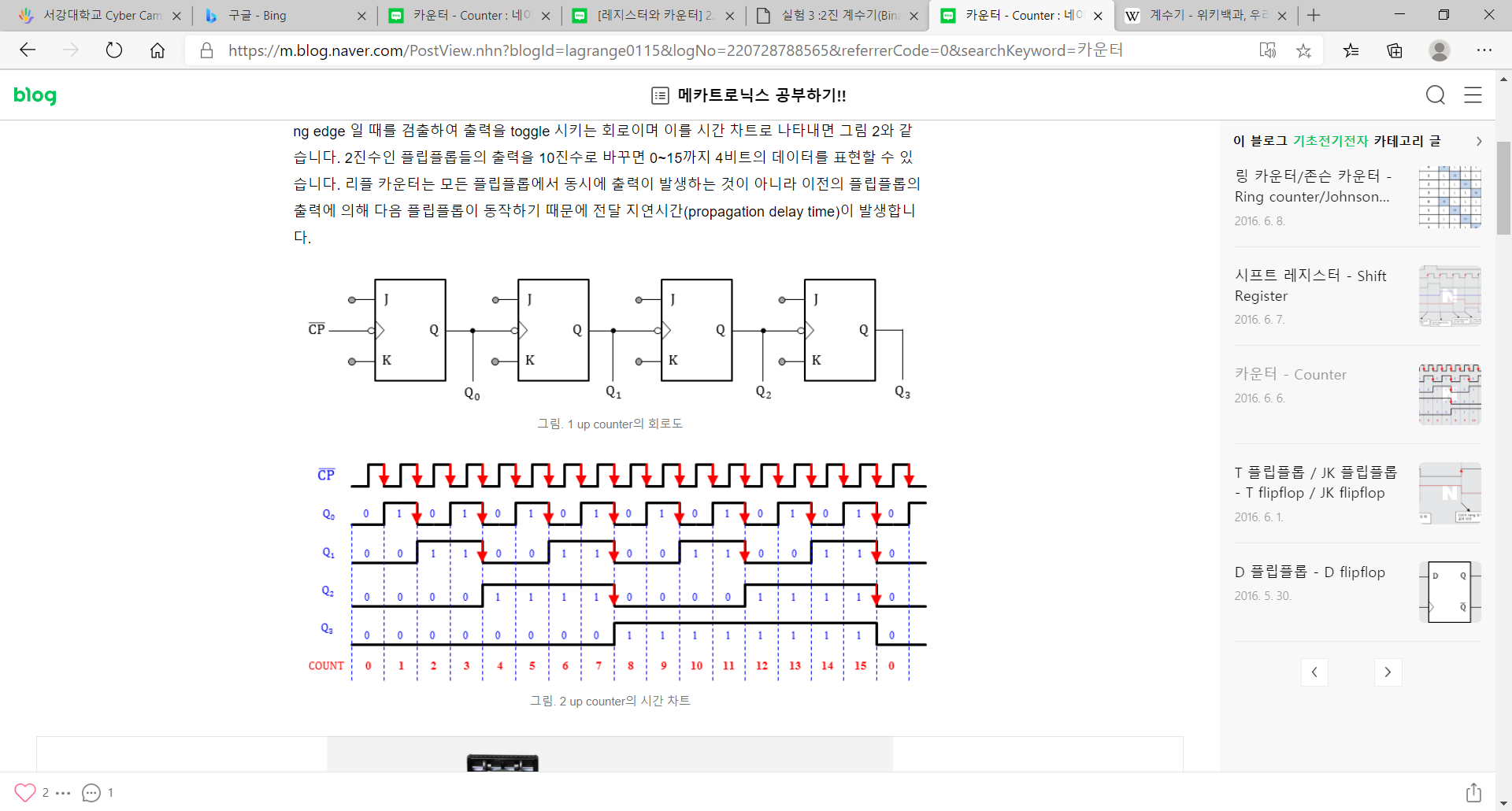
Decade counter을 회로로 나타내면 다음과 같다.



위의 그림을 보면 알 수 있다시피 decade counter는 JK flip-flop로 구성되어 있으며 4개의 JK flip-flop이 직렬 연결되어 각각의 flip-flop의 출력 값은 다음 flip-flop의 CLK값으로 들어가게 된다. 0000부터 1001까지의 값은 잘 가지지만 그 다음 숫자인 1010의 값을 갖게 될 경우에는 NAND gate에서 최종적으로 0을 갖게 되면서 4개의 flip-flop을 0로 초기화하면서 다시 0000으로 돌아가게 된다.

**3.**

Counter는 크게 동기식 카운터와 비동기식 카운터로 나눌 수 있는데 비동기식 카운터는 낮은 자릿수의 flip-flop의 출력 값에 의해 높은 자릿수의 flip-flop의 출력 값이 영향을 받기에 ripple counter라고도 불린다. 작동방식이 Ripple carry adder와 비슷하기에 비동기식 카운터도 전달 지연이 발생하고 이러한 단점으로 인해 카운터를 구성하는 모든 flip-flop이 같은 clock 펄스에 의해 활성화되는 동기식 카운터가 도입되었다. 앞서 설명한 binary ripple counter와 decade counter는 비동기식 카운터의 예시이다. 비동기식 카운터는 상향 카운터와 하향 카운터로 나눌 수 있는데, 상향 카운터는 하강 edge에서 활성화되고, 하향 카운터는 상승 edge에서 활성화되는 카운터이다. 상향 카운터를 회로와 그림으로 나타내면 다음과 같다.



동기식 카운터는 동일한 clock 펄스에 모든 flip-flop이 연결되어 있어서 전달 지연이 생기지 않는다는 장점이 있지만 그렇기 때문에 회로가 좀 더 복잡해 진다는 단점이 있다. 동기식 카운터의 회로는 다음과 같다.

텍스트, 소프트웨어, 멀티미디어 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진

자동 생성된 설명

**4.**

FSM은 Finite State Machine의 줄인 말로 state(상태)을 기반으로 하여 동작을 제어하는 것을 구현하기 위한 추상적 모형을 의미한다. 가질 수 있는 유한개의 상태들 중에서 반드시 하나의 상태를 취해야 하고, 현재 상태에서 특정한 조건을 만족할 시에 다른 상태로 바뀔 수 있다. 하나의 상태에서 다른 상태로 바뀌는 것을 transition(전이)고 하고 FSM의 일반적인 도식도는 다음과 같다.

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

위의 그림에서는 가질 수 있는 상태가 S1, S2, S3 이렇게 총 3개이며 각각의 상태에서 1의 입력 값을 가지면 S1은 S2로, S2는 S3으로, S3은 S1로 전이되며, 0의 입력 값을 가지면 현재의 상태를 유지하는 것을 나타낸다. FSM은 Moore Machine과 Mealy Machine로 나눌 수 있다.

**5.**

FSM을 크게 2가지로 나누면 Moore Machine과 Mealy Machine로 나눌 수 있는데, Moore Machine은 출력 값이 입력 값과는 관계 없이 현재 상태의 영향을 받아서 결정되고 Mealy Machine은 현재 상태와 입력 값에 의해 출력 값이 결정된다.

스크린샷, 텍스트, 소프트웨어, 도표이(가) 표시된 사진

자동 생성된 설명

위의 그림은 각각 Mealy Machine과 Moore Machine인데, 그림에서 볼 수 있듯이 Mealy Machine에서는 입력 값인 X(t)가 출력 값에 직접적으로 영향을 주는 반면 Moore Machine에서는 X(t)가 출력 값에 직접적으로 영향 주지 않고 회로를 한 번 이상 거치는 것을 확인할 수 있다.