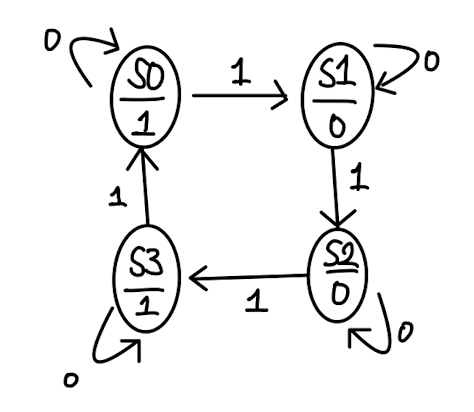
14주차 예비보고서

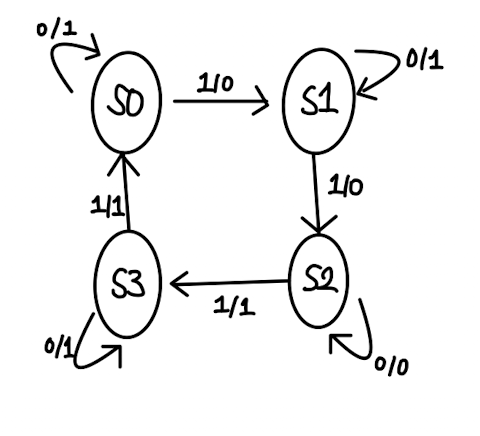
전공: 심리학과 학년: 3학년 학번: 20190345 이름: 김동현

1.

FSM은 Finite State machine의 약자로 컴퓨터 프로그램 혹은 전자 논리 회로를 설계하는 데에 사용되는 수학적 모델이다. FSM은 임의의 시간에 하나의 상태를 가지며, 하나의 상태에서 다른 상태로 바뀌는 전이를 통해 상태를 변화한다. FSM은 deterministic finite automata와 non-deterministic finite automata로 나뉜다. Deterministic finite automata는 임의의 시간에 하나의 상태만으로 변환될 수 있는 반면, non-deterministic finite automata는 임의의 시간에 하나 이상의 상태로 변환될 수 있다는 차이가 있다. FSM의 예시로, 유한 상태 변환기가 있다. 유한 상태 변환기는 현재의 상태가 입력에 따라 다른 상태로 변환되며 이에 맞는 출력값을 생성한다. 유한 상태 변환기는 moore model과 mealy model로 나뉜다. 아래와 같이 Moore model과 mealy model의 state diagram은 다음과 같이 표현할 수 있다.



Moore model의 경우 상태와 출력값이 동시에 표현되며, 입력값에 의해 상태가 변화한다.



Mealy model의 경우 상태만 표현되며, 입력값에 의해 상태 변화 및 출력값이 결정된다.

2.

mealy model은 출력값이 현재 상태와 입력값에 따라 결정되는 유한 상태 변환기이다. mealy model은 moore model에 비해 더 적은 state를 필요로 하며, 회로를 구현하기 상대적으로 어려우나, 입력값에 대한 출력이 즉각적으로 반영된다

예시로 1이 3회 연속으로 입력될 경우 1을 출력하는 시스템을 mealy model로 구현하면 다음과 같다.

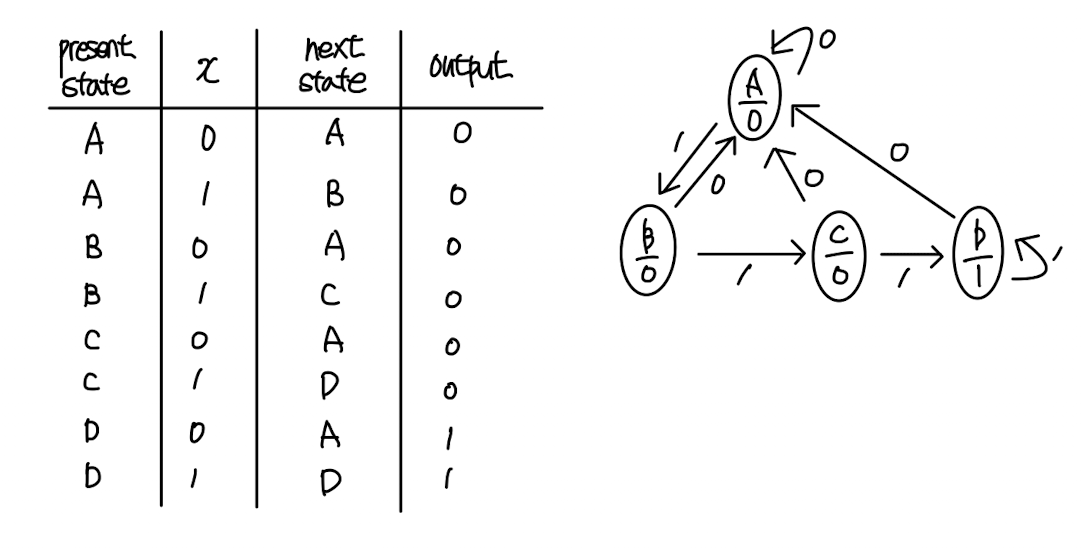
테이블이(가) 표시된 사진

자동 생성된 설명

3.

Moore model은 출력값이 현재 상태에 의해서만 결정되는 유한 상태 변환기이다. Moore model은 mealy model과 비교하여, 더 많은 state를 필요로 하며, 회로를 구현하기 상대적으로 쉬우며, 입력값에 대한 출력이 느리게 반영된다.

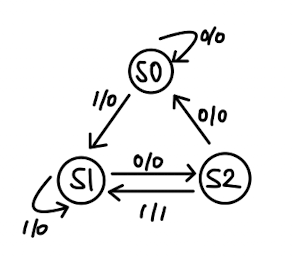
예시로 1이 3회 연속으로 입력될 경우 1을 출력하는 시스템을 moore model로 구현하면 다음과 같다.



상태를 표시할 때, 같은 예제에 대해 moore model은 4개의 상태, mealy model은 3개의 상태로표현되는 것으로 보아 moore model이 mealy model이 더 많은 상태를 가진다. 뿐만 아니라 출력값이 moore model은 상태와 함께 표시되지만, mealy model은 입력값과 함께 표시된다는 점이 차이를 보인다.

4.

Sequence detector는 데이터 sequence에서 특정 패턴이 발견될 때마다 1을 출력하는 detector이다. 다음은 0과 1로 구성된 데이터 sequence에서 101 패턴이 발견될 때마다 1을 출력하는 detector 예시이다. 0011011001010100의 데이터가 주어질 경우, 0000010000010100이 출력된다. 위 예시를 mealy model이며, JK flip-flop을 사용해서 회로를 구성하고자 한다. 우선 101패턴을 발견하기 위한 state diagram을 만든다.



S0은 101 패턴 중 첫번째 자리의 1이 입력으로 주어진 경우이다. S1은 101 패턴 중 10까지 입력으로 주어진 경우이다. S2는 101패턴이 발견된 경우로, S1에서 S2로 상태가 변화하는 순간이 101 패턴이 나온 순간이므로 1을 출력한다. 이러한 state diagram을 통해 state table을 구성한다.

테이블이(가) 표시된 사진

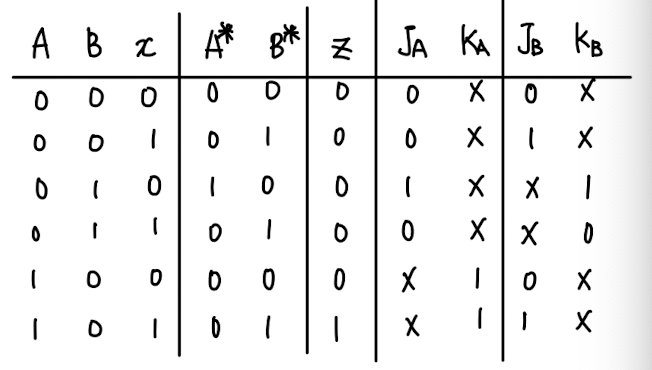
자동 생성된 설명

구성된 state table을 통해 logic diagram을 JK flip-flop을 사용하여 구성하고자 한다. 이를 위해 JK flip-flop의 excitation table을 구하면 다음과 같다.

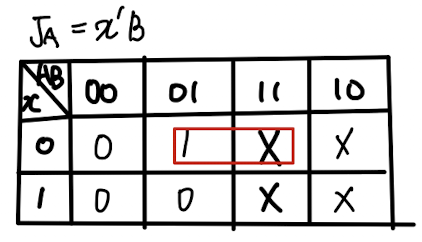
텍스트, 체척계이(가) 표시된 사진

자동 생성된 설명

뿐만 아니라 S0,S1,S2의 상태를 2비트의 수로 표현하면 각각 00,01,10으로 표현한다. A\*는 첫번째 비트 A의 다음 상태를, B\*는 두번째 비트 B의 다음 상태를, z는 출력값을 의미한다. 이를 통해 state table을 구성하면 다음과 같다.

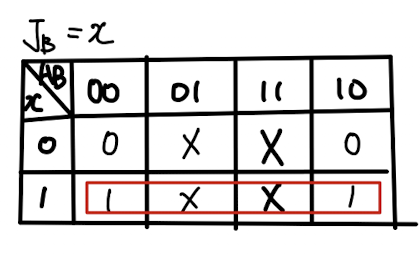


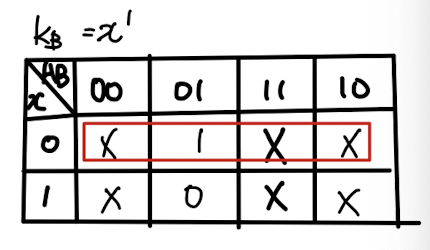
현재 상태와 다음 상태 (A,A\*), (B,B\*)를 JK flip-flop excitation table을 통해 각 비트를 나ㅏㅌ내는 JK flip-flop의 J,K을 결정한다. J,K값과 출력값을 결정하기 위해 카르노맵을 구성하면 다음과 같다.

\

테이블이(가) 표시된 사진

자동 생성된 설명

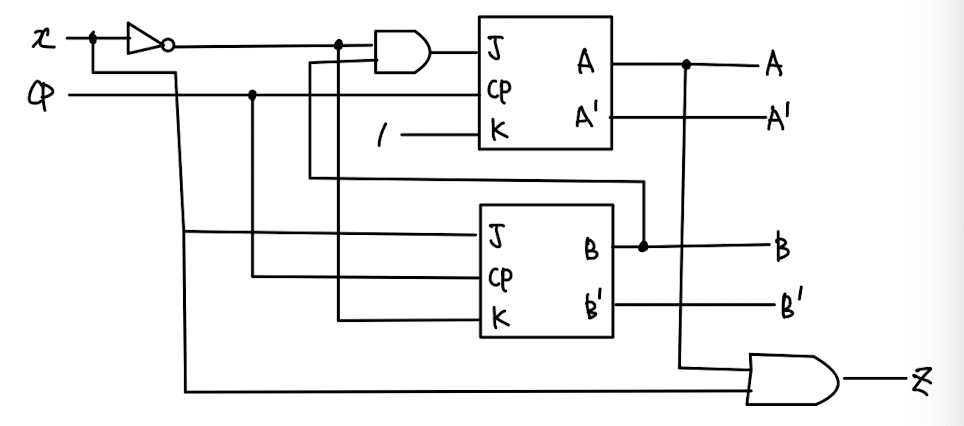




테이블이(가) 표시된 사진

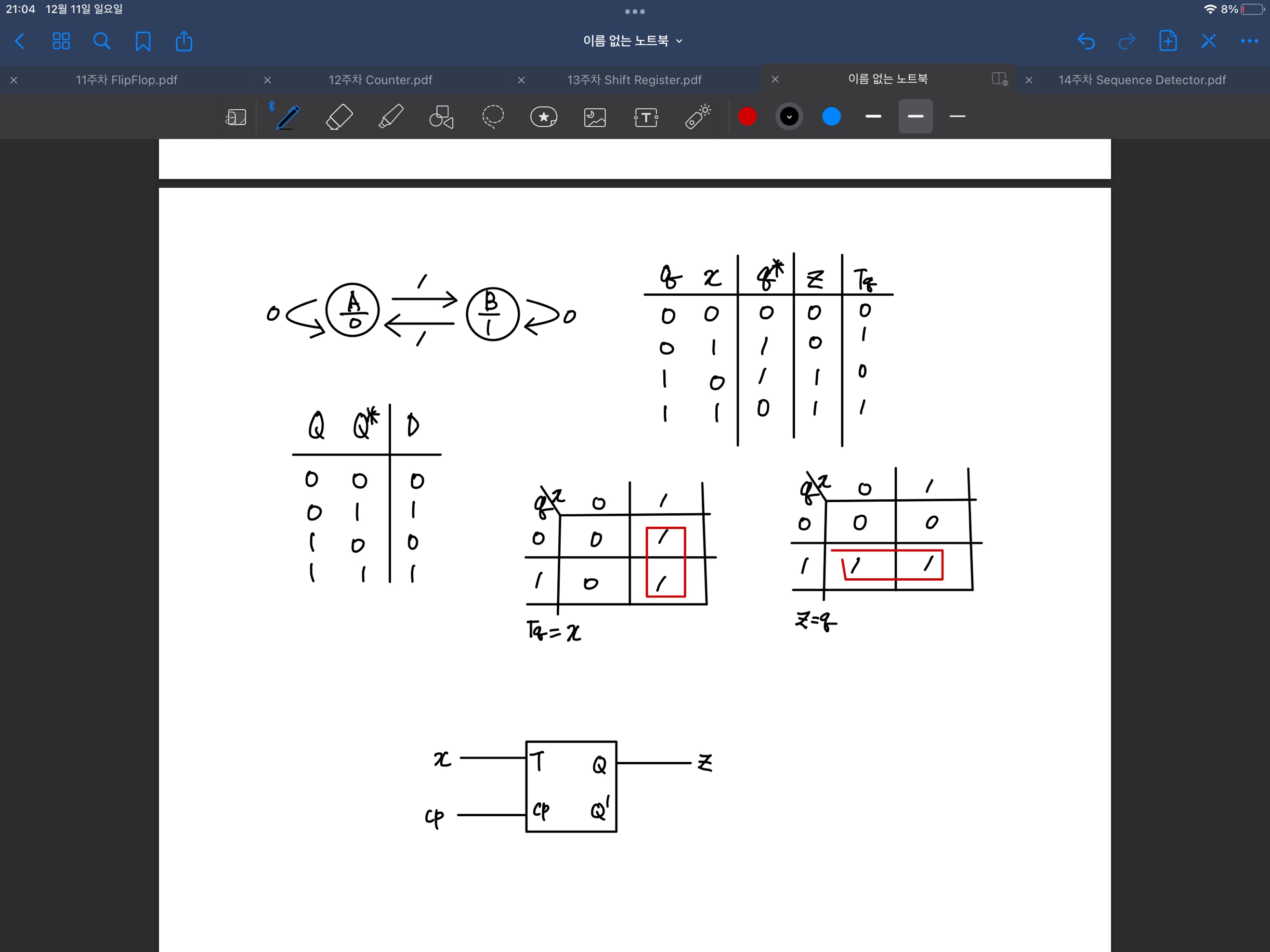
자동 생성된 설명

카르노맵을 통해 단순화 한 식을 통해 logic diagram을 나타낼 수 있다. 이는 101 sequence detector를 나타낸다.

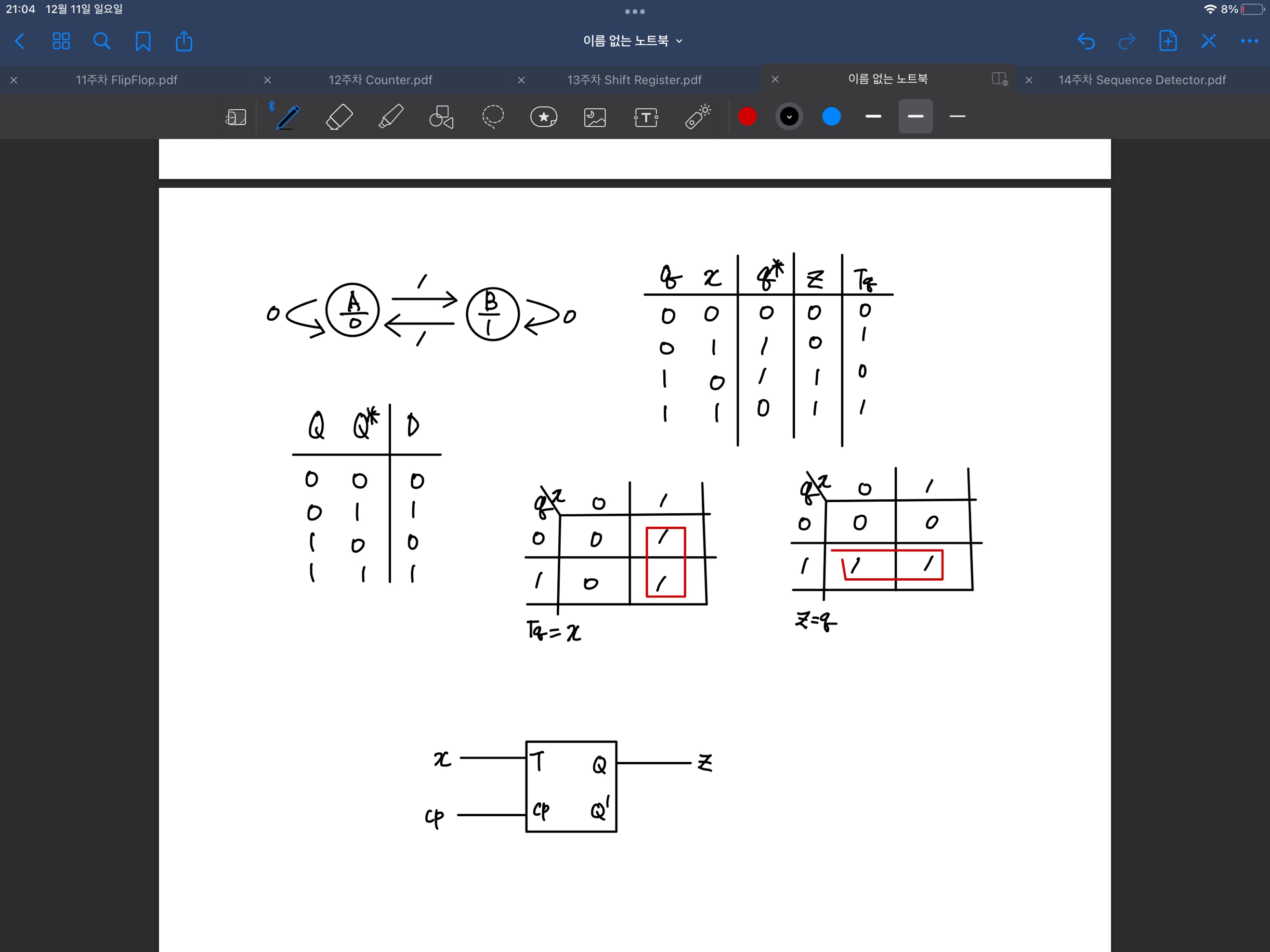


5.

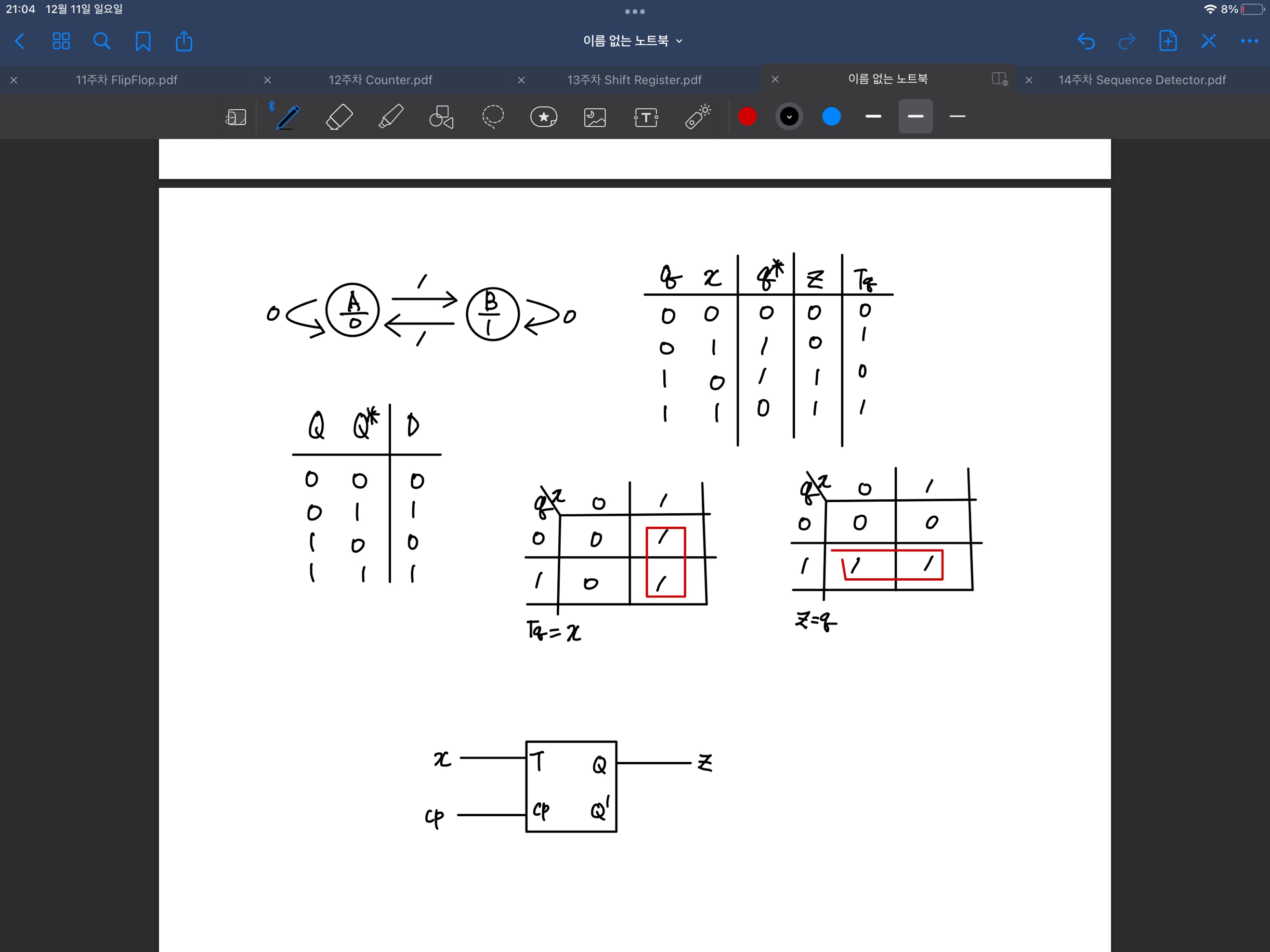
Parity checker example 또한 sequence detector를 응용하여 만들어낼 수 있다. 우선 parity checker는 1의 개수가 짝수이면 정상이므로 0을, 1의 개수가 홀수이면 오류이므로 1을 출력한다. 이를 state diagram으로 표현하면 다음과 같다. A는 1의 개수가 짝수개일 때, B는 1의 개수가 홀수개일 때를 나타낸다. 1이 입력으로 들어오면 1의 개수가 변화하므로 상태 또한 변화하고, 그렇지 않은 경우 1의 개수는 유지되므로 상태 또한 유지된다.



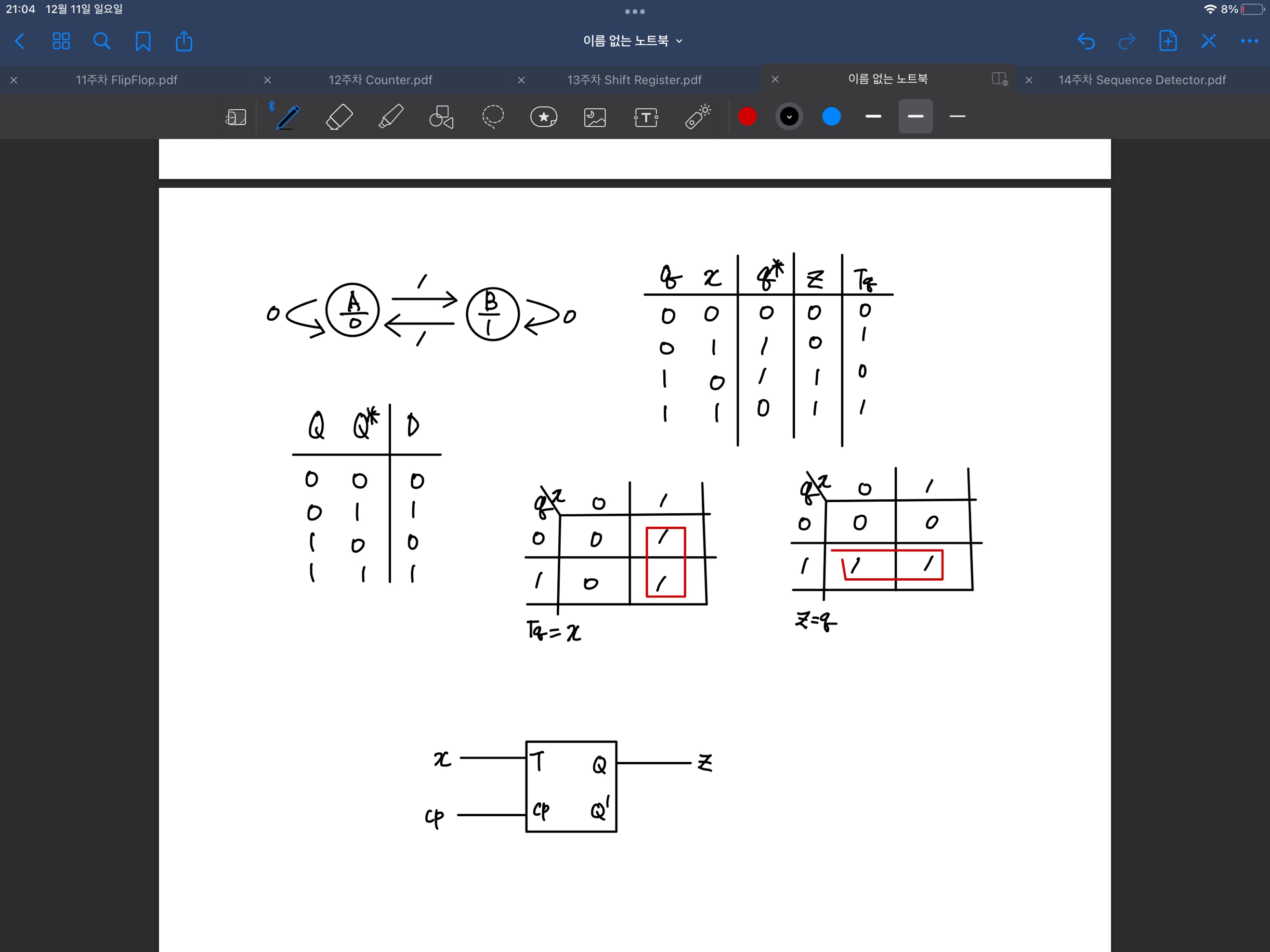
위의 state diagram을 기반으로 state table을 만들었다. T flip-flop을 사용하여 회로를 구성한다.



k-map을 통해 flip-flop과 출력값을 간소화하면 다음과 같다.



이를 통해 logic diagram을 구성하면 다음과 같은 결과가 나온다.



6. 참고문헌

Alan B. Marcvitz, Introduction to Logic Design, McGraw-Hill(2010)

Jonathan E. Steinhart, “한 권으로 읽는 컴퓨터 구조와 프로그래밍”