**디시설 프로젝트 제안서 초안**

4조 박민혁(20210010), 김지민(20210084), 양준영(20210216)

1. **구현하고자 하는 주제와 그 목적**

작은 디스플레이 속 가상의 애완동물을 키우는 추억의 장난감 ‘다마고치’에서 영감을 받아, ‘포닉스 키우기’를 구현하고자 한다. 사용자가 원하는 행동을 입력 받아 그에 따른 포닉스 상태 변화를 디스플레이로 출력하는 과정을 통해 FSM 설계 능력을 함양할 수 있을 것으로 예상한다. 더불어 플립플롭과 같은, 순차회로의 기본 요소들을 익히는 것이 목적이다.

1. **구현과 관련된 수업 내용 및 배경 지식**

FSM에 대해 알아보기 전에 우선 순차 회로의 기본 요소들을 알아보자. 래치(Latch), 플립플롭(Flip-flop)은 1 비트의 신호를 저장할 수 있다. Latch는 Level을 이용한 방식으로, 입력 변경에 따라 출력도 바로 바뀐다. 하지만 이런 경우 정확한 출력 타이밍을 알기 어렵다는 단점이 있는데, 이를 보완하기 위해 Rising Edge (Positive Edge)와 Falling Edge (Negative Edge)를 활용하기도 한다. 이렇게 Edge를 활용한 방식을 Filp Flop이라고 한다. 즉, 플립플롭은 정보를 저장할 시점을 정하기 위해 클록 신호를 추가로 받는 것이라고 할 수 있다.

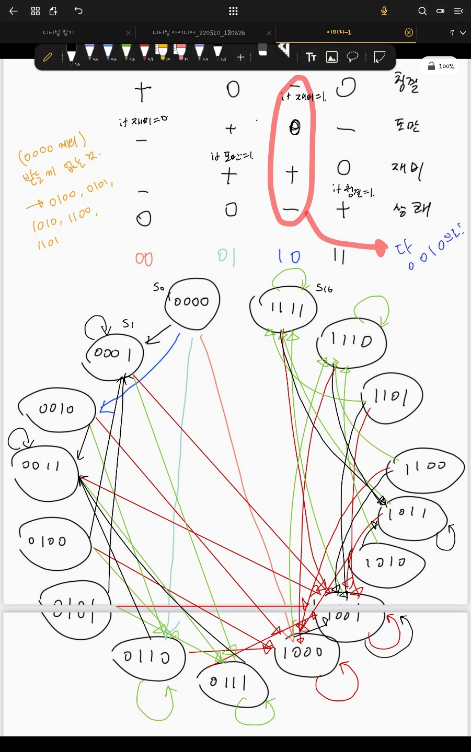
이번 프로젝트에서 사용할 소자는 JK 플립플롭이며, 기존 S-R 플립플롭에서 제한되었던 11 입력에 토글(toggle) 기능을 추가하였다. 기본적으로 J는 S, K는 R의 역할을 수행하며, 11이 입력되면 토글 기능 즉, 상태를 반전시키는 역할을 수행한다.

FSM (Finite State Machine)은 플립플롭이 가지는 데이터 즉, 상태(state)의 개수를 유한개로 정해 놓고, 사용자의 입력에 따라 상태의 전이를 제어하는 기계이다. FSM에는 크게 두 가지가 있는데, 첫 번째 Moore Machine은 현재 state에 의해서만 출력이 결정된다. 두 번째, Mealy Machine은 현재의 입력과 state 둘 다에 의해 출력이 결정된다. 예를 들어 S0, S1이라는 상태가 있다고 해보자. Moore Machine은 상태가 어떻게 전이되었는가에 영향을 받지 않고, 현재 상태가 S0, S1냐에 따라 출력이 달라진다. 반면 Mealy machine은 이전 상태에서 어떤 입력을 받아 현재 상태가 되었는지를 판단하고, 그때마다 출력이 달라진다. 이번 프로젝트에서는 출력이 현재 상태에만 영향을 받는 Moore Machine을 따른다.

유한 상태 기계를 설계하는 과정은 다음과 같이 정의할 수 있다.

1. 문제를 인지하고 정의한다.
2. 각 상태에 이진 값을 정의한다. (S0(00): 0 cent, S1(01): 5 cent, S2(10): 10 cent 등등)
3. State diagram을 그린다.
4. State Table을 만든다.
5. 3.을 토대로 카르노 맵 등을 이용하여 각 식을 정리한다.
6. 각 식을 적용한 플립플롭들을 적절히 배치해 회로도를 완성한다.

이와 같은 과정을 거쳐 FSM을 설계하고 직접 FPGA에 구현해본다.

** ㄷ. State transition diagram 및 State machine**

해당 State transition diagram은 목욕, 간식, 산책, 잠자기 등의 활동에 따른 state의 변화를 나타낸 diagram이다 (청결,포만,재미,상쾌) 밑의 표는 해당 state 변화에 대한 표이다

목욕은 청결을 증가시키고 포만과 재미를 감소시킨다

간식은 포만과 재미를 증가시킨다

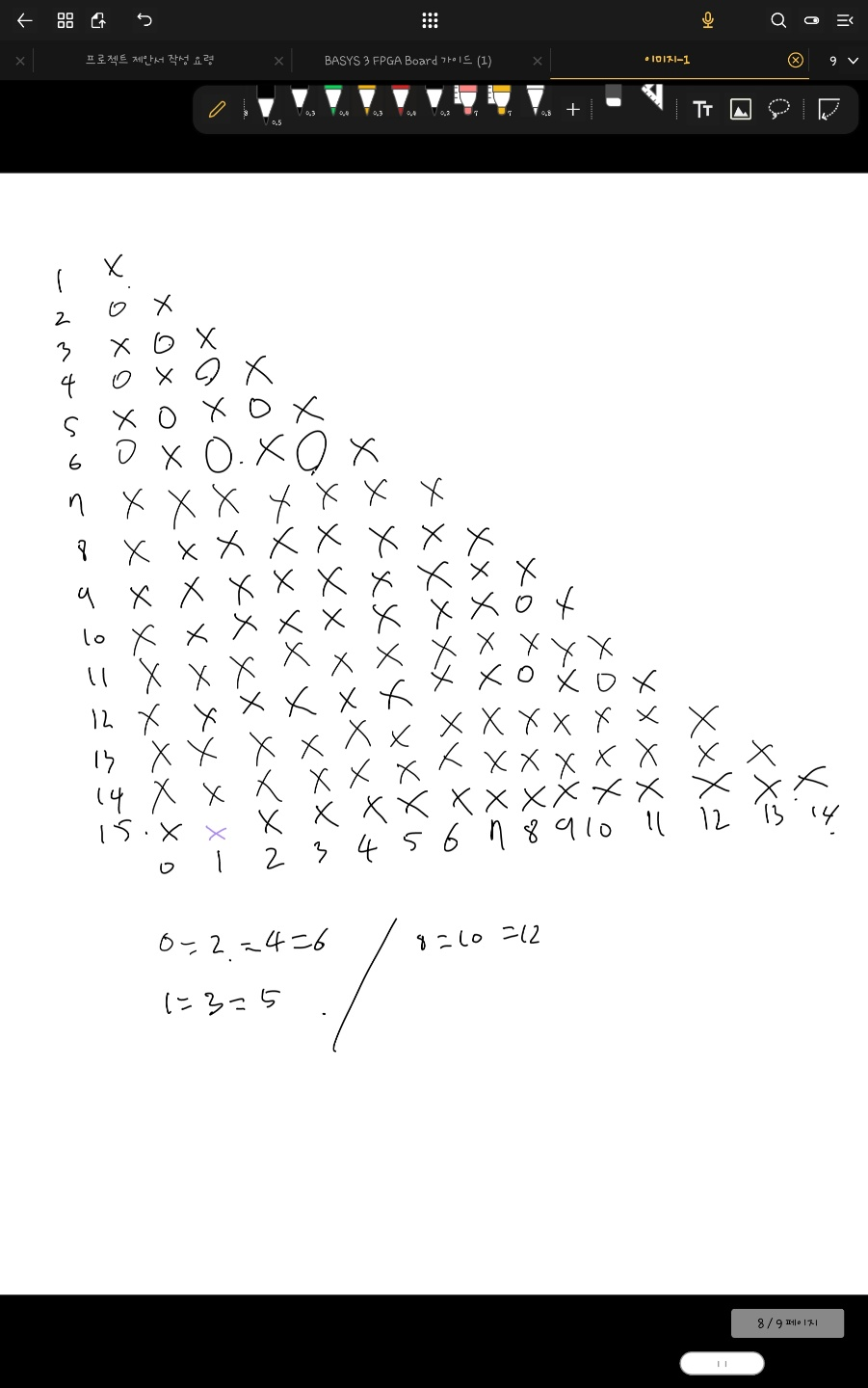
산책은 청결과 상쾌를 감소시키지만 재미를 증가시킨다

잠자기는 포만을 감소시키지만 상쾌를 증가시킨다.

테이블이(가) 표시된 사진

자동 생성된 설명

4개의 스탯 LED 중 3개가 켜졌을 때 결과 LED가 1개 켜지고, 4개다 켜지면 결과 LED가 2개 모두 켜진다.



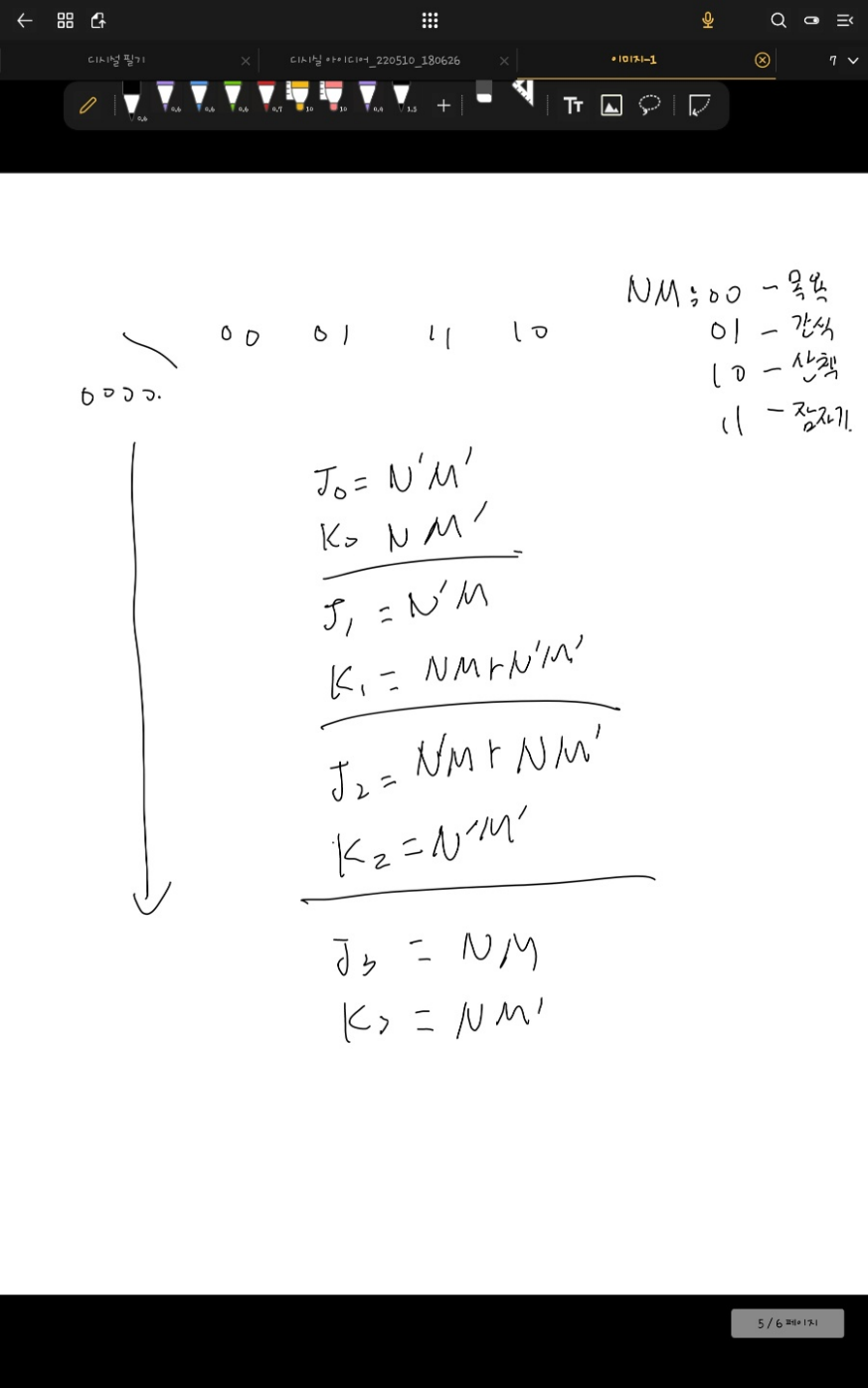
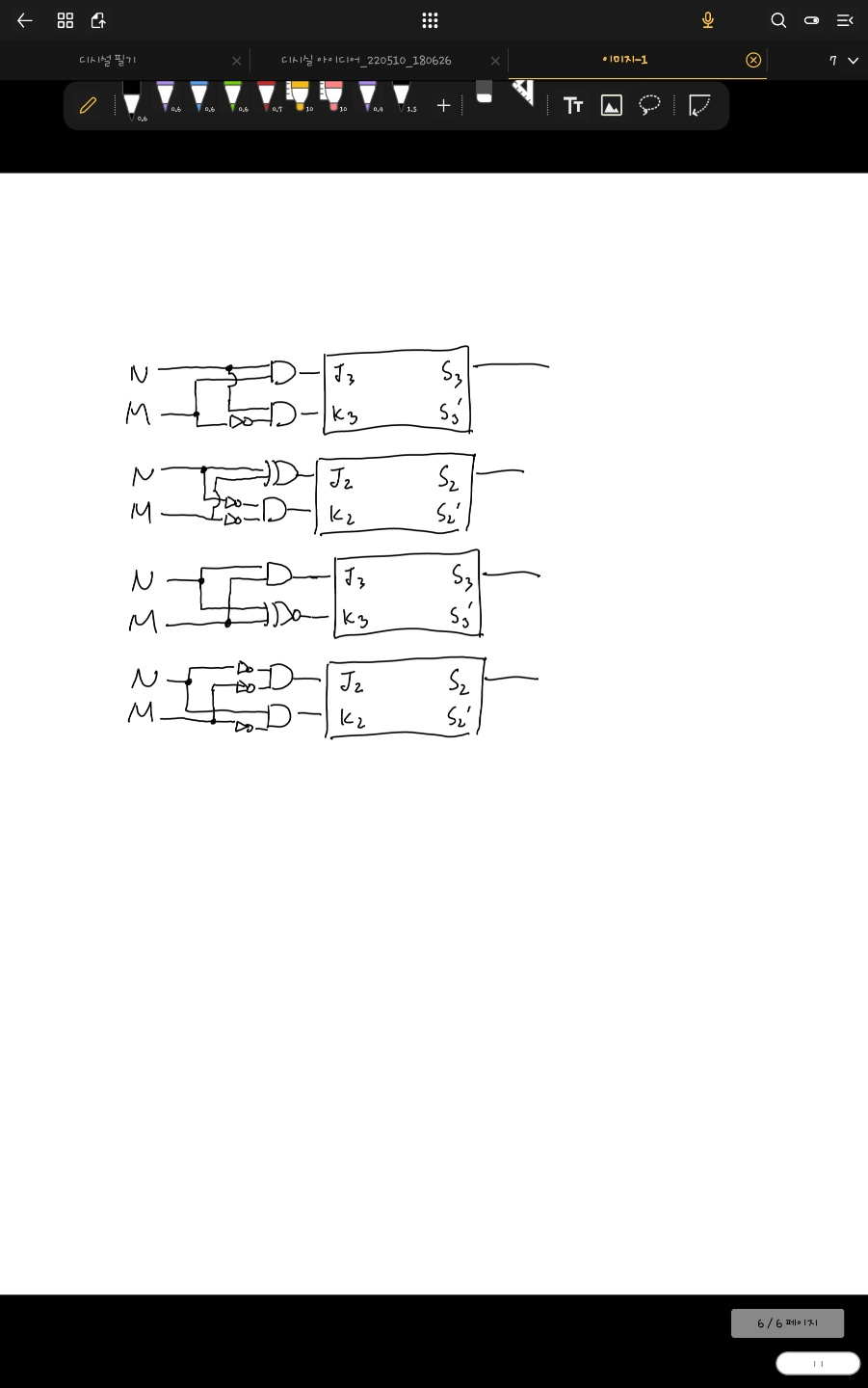
위 사진은 I-chart를 이용하여 상태를 줄인 것이다.

테이블이(가) 표시된 사진

자동 생성된 설명

줄인 상태에 대한 state table이다. 총 9개로 상태가 줄어든 것을 확인할 수 있다.

각 state에 대한 식을 quine-mccluskey 방식을 통해 정리하면 식은 왼쪽의 식처럼 정리되며 해당 식에 따른 State-machine은 밑의 그림과 같다.



**ㄹ. 입출력 및 동작 설명**

회로에는 4개의 누를 수 있는 버튼이 있다. 버튼들은 각각 목욕, 간식, 산책, 잠자기 행동 명령을 내릴 수 있게 해주는 입력장치이다. 또한 회로에는 4개의 LED가 존재하는데, 각각 청결, 포만, 재미, 상쾌 4가지 욕구의 현재 상태를 나타낸다. LED가 점등되어 있으면 해당하는 욕구가 충족된 상태를 의미한다.

처음 전원이 들어오면 네 LED는 전부 꺼진 상태로 시작한다. 사용자의 목표는 주어진 4개의 행동 명령을 통해서 포닉스의 4가지 욕구를 전부 충족시킨 상태, 즉 4개의 LED가 전부 점등되도록 만드는 것이다. 포닉스의 욕구 충족 상태에 따라 독립적으로 표시되는 효과가 디스플레이에 나타난다. 예를 들어 청결 상태가 0이라면, 포닉스의 몸 곳곳에 때가 묻어있고, 청결이 1이 되면 때가 전부 사라지게 된다. 특정 욕구에 의한 포닉스의 상태변화는 다른 욕구에 의한 상태변화에 영향을 주지 못한다. 예를 들면 청결 욕구에 의해 변화하는 때묻음은 포만 욕구에 의한 상태변화인 튀어나온 배에 영향을 주지 않고 독립적으로 표시된다.

게임을 시작하면 사용자는 4가지 버튼을 눌러 행동명령을 내릴 수 있다. 각 행동명령은 4가지 상태를 정해진대로 변화시키며, 각 욕구상태는 0이 되거나, 변화가 없거나, 1이 될 수 있다. 해당 수치는 사용자에게는 공개하지 않는다. 사용자가 원하는 행동명령을 입력하면 누른 버튼에 따라서 FF에 입력이 전달되어 state가 변한다. 사용자는 포닉스에게 필요한 행동 명령을 여러 차례 내리면서 포닉스의 모든 욕구를 충족시켜주면, 즉 state를 1111로 만들어주면 게임에서 승리하게 된다.

**ㅁ. 필요한 부품 및 사용처**

