AES Processor Design Report

디지털설계 및 실험 001분반

2023014973 노성민

목차

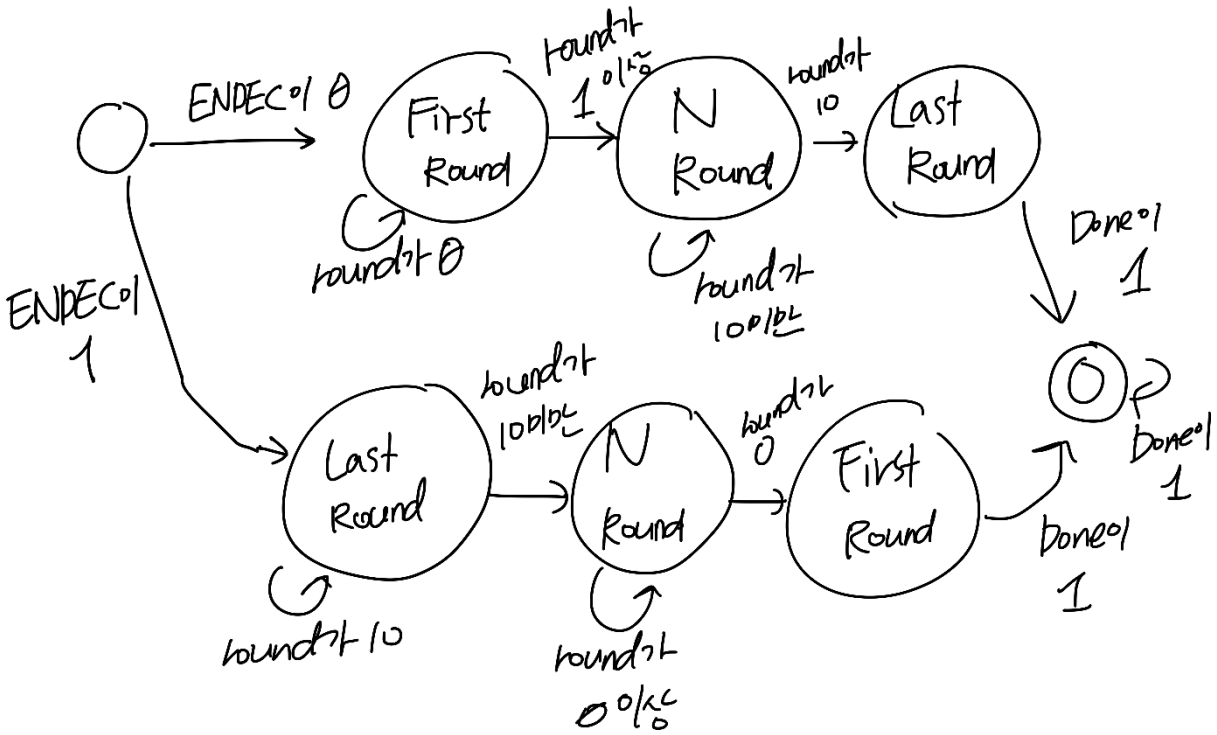
Design Report

※ C/C++, Verilog source code는 별도 첨부하였습니다.

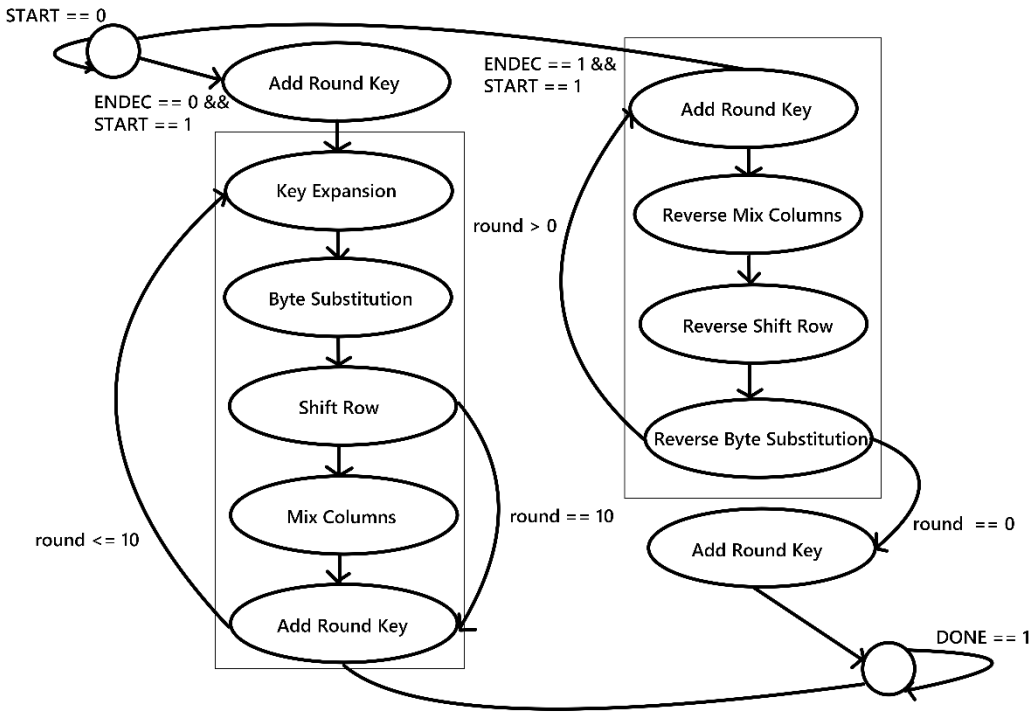
※ C/C++ source code는 Encryption.c, Decryption.c를 참고해주십시오.

※ Verilog source code에서는 AES.v가 design, TOP.v stimulus code입니다.

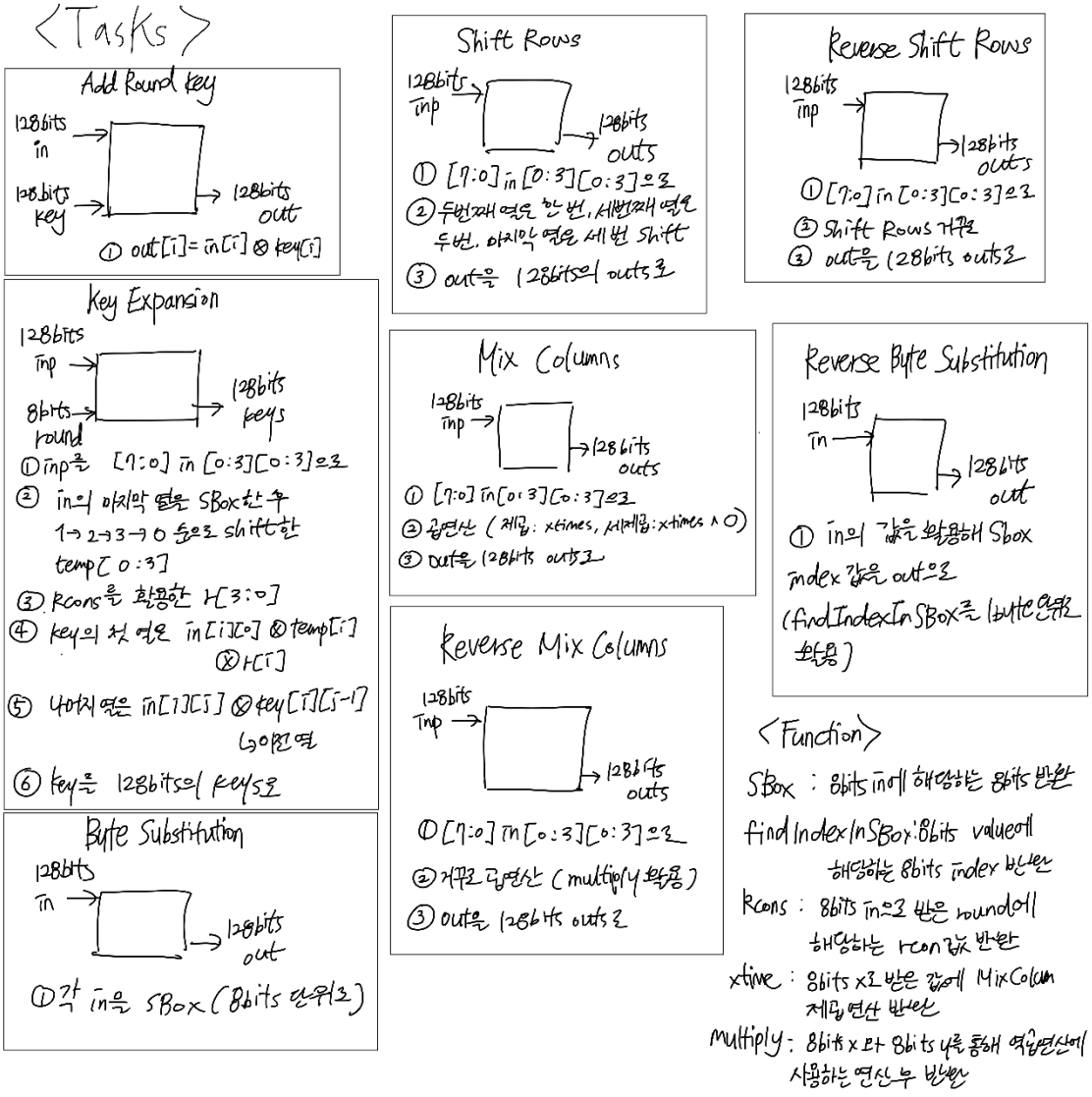
1. Finite State Machine



위 그림에서 볼 수 있듯이 ENDEC의 input에 따라 0일 경우 Encryption을 수행하는 state로, 1일 경우 Decryption을 수행하는 state로 가는 것을 볼 수 있습니다. 각 state에는 한 clock이 소모되며 모든 round를 수행했을 시 Done이 1이 되며 Final state로 가게 됩니다. Round별 자세한 기능은 아래 diagram에서 확인할 수 있습니다.



2. Datapath Design

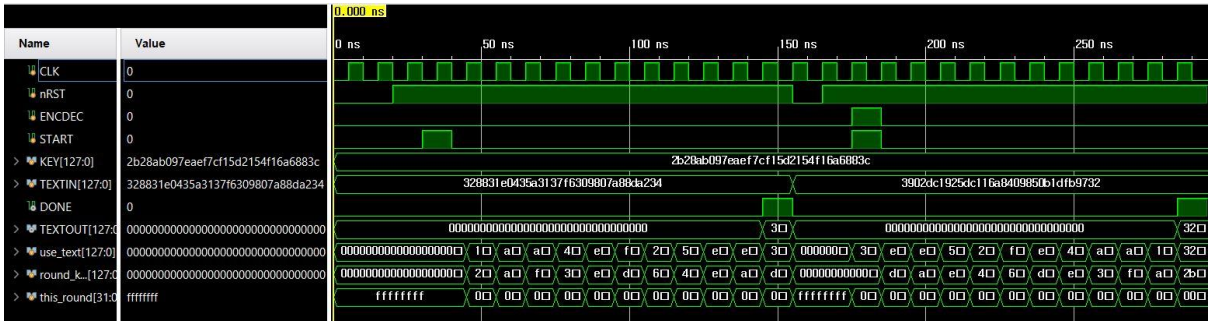


Finite State Machine의 두번째 그림에서 이어서 Datapath의 흐름을 확인할 수 있게 각 Task와 Function별 port와 설명을 명시하였습니다. 각 네모는 Task를 의미하여 우측 아래 Function에 대한 간략한 개요를 설명하였습니다.

3. Source Code Description

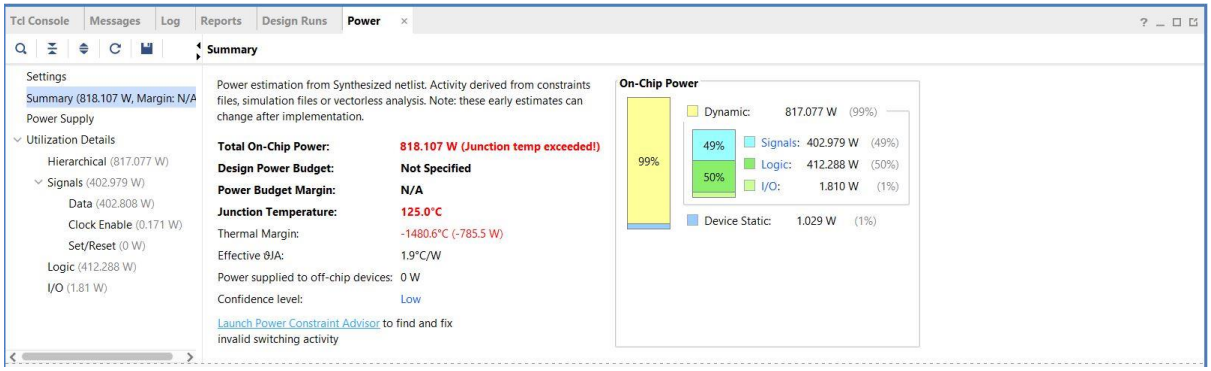
베릴로그 소스코드의 각 설명은 AES.v에 주석으로 자세하게 추가하였습니다. 해당 주석을 참고해주십시오.

4. Waveform



AES Rijndael Cipher explained as a Flash animation - YouTube에서 제공하는 예제를 통해 stimulation을 진행하였습니다. Decryption단계에서 각 Key Expansion을 한 클럭에 모두 시행하였기 때문에 추가적인 클럭 사용 없이 한 round당 한 클럭을 사용하여 구현하였습니다. Waveform에서는 각 라운드별 현황을 확인하기 위하여 this_round 변수를 사용하여 명시하였습니다.

5. Synthesis



소스코드를 성공적으로 합성하였으며 Schematic은 총 32장으로 첨부는 생략하였습니다. 그러나 현재 합성에서 Power부분에 문제점이 있었습니다. 설계에서 소비되는 전력이 너무 커 칩이 심각하게 과열될 것을 예상할 수 있습니다. Thermal Margin 또한 -1480.6 ℃로 현실적으로 사용할 수 없는 설계입니다. 분석해본 결과, 반복되는 연산이 많아 과도하게 복잡하기 때문이거나 설계 자체가 잘못됐을 가능성이 있었습니다. 전자는 cache와 같은 다양한 기법을 통해 해결할 수 있으나 후자일 경우 전력을 고려해 재설계가 필요합니다. 또 다른 가설로는 Task/Function 대신 Module로 분리하여 설계를 했더라면 전력소모와 발열 감소를 기대할 수 있지 않았을까 생각하였습니다.

이번 프로젝트를 통해 Verilog를 통한 HW설계에서는 단순히 작동하는 것과 stimulus가 의도한대로 수행하게 만드는 것뿐만 아니라 합성이 가능한지, 그리고 합성된 결과가 정상적으로 작동하며 전력소모가 현실적인지까지 고려해야 할 사항이 수도 없이 많다는 것을 몸소 느끼게 되었습

니다. Verilog를 통한 설계에서는 합성은 피해갈 수 없는 가장 중요한 요소 중 하나이며 저는 이번 프로젝트에서 적절한 합성 결과를 얻지 못하였습니다. 코드를 수행 가능하게 하고 stimulus를 의도한대로 조정하여도 태초의 설계부터 잘못되어 합성이 적절하게 되지 않으면 이 모든 활동이 물거품이 될 수 있다는 것을 알게 되었습니다. 이를 바탕으로 설계의 중요성을 되새기며 추가적으로 발열을 줄일 수 있는 여러 방법들에 대해서 후속적으로 공부해 나가고자 합니다.