# **Personal Feelings**

이번 과제는 Pipelined MIPS Simulator를 작성하기 위한 나만의 Micro architecture를 작성하고, 이를 C언어로 구현하며, 구현한 결과물을 평가하는 것이었다. 이는 이전 과제에서 구현한 Single-Cycle MIPS Simulator 결과물에 Pipeline이라는 개념을 적용하기 위해서 여러 하드웨어 구조적인 함수들을 추가로 구현 및 적용하고, 이 과정 중에서 발생하는 여러 문제점을 해결하여 기존의 Simulator의 성능을 소프트웨어적으로 향상하는 과제였다.

처음에 과제를 맞닥트렸을 때에는 매우 당황하였다. 기존에 구현한 Single-Cycle MIPS Simulator를 잘 구현하였다고 생각하였는데, 막상 Latch를 구현하니 이전에 돌아갔던 코드가 정상적으로 실행되지 않았으며, 이를 해결하기 위해 Data Forwarding 기법을 적용하였을 때도 코드가 정상적으로 돌아갔으나, 이를 확인하기 위한 디버깅 과정이 까다로워 많은 시행착오를 겪었다. Branch Predictor 구현은 초기에 2-bit Counter-Based Predictor를 완성하는 데 있어 개념적으로 정립이 안 된 상태에서 작업을 시작하였기 때문에 왜 이런 방법을 통해 Branch Prediction을 하면 효율적인지 깨닫지 못했다. 하지만 나머지 Branch Predictor들을 다 구현한 다음에 여러 방법론이 이전의 방법들의 단점들을 보완해 가면서 생겨난 것이라는 것을 배웠고, 이를 수치화 하여 비교해 봤을 때, 항상 봉안된 Branch Predictor가 좋은 성능을 나타내는 것은 아니라는 것을 깨달었다.

이번 과제를 끝마치고 만족스러운 결과물을 얻었으며, 동료와 서로 과제를 진행하면서 맞닥트린 문제들을 공유하며 이를 해결하는 경험을 하여 매우 의미 있던 시간이라고 느꼈다. 해당 과제 구현에 있어서 참고할 코드는 없었다고 생각한다. 과제에서 원하는 프로그램들을 수행하려면 기존에 우리가 알고 있는 Pipelined Micro architecture에 여러 Hardware 및 Value들을 추가하여 하는데, 인터넷에 공유되거나 검색하여 찾을 수 있는 여러 다른 코드들 및 과제들은 이와 다른 구조를 띄고 있다. 따라서 저번과제와 마찬가지로 그냥 처음부터 구현하는 게 합리적 이였다. 특히 이번 과제의 경우, 구현도 구현대로 시간이 오래 걸리며 구현 난이도가 높았다고 생각하지만, 과제를 진행하는 데 있어서 느낀 가장 큰 어려움은 바로 보고서를 작성하는 것이었다. Pipelined Micro architecture는 이전의 여러 Micro architecture의 단점들을 보완하며 나온 개념이기에 이전 개념 들로부터의 발전 과정을 보여주어야 했으며, 발전 과정에서 발생하는 문제점들을 해결하는 방안들 또한 설명해야 했고, 보고서에 작성되어야 할 내용과 이를 뒷받침하는 정보들 및 실례들을 작성하는 데 있어서 많은 시간과 노력이 들어갔다.

과제를 수행하는 데 있어 가장 흥미로웠던 점은 Branch Prediction이었다. 수업 중 예시 자료로서 우리에게 보여준 선배들의 보고서에는 Branch Prediction 확률에 대한 수치화와 이에 대한 성능 비교 항목이 존재하였는데, 이를 참고하여 보고서를 작성하니 해당 과목에 대한 이해도가 올라간 느낌을 받았다. 반대로 아쉬운 점은 해당 과목의 학업 진도와 과제 진행 정도가 서로 안 맞는다는 것이었다. 수업 진도에 맞춰 과제를 수행한다면 큰 문제점이 없겠지만, 이 경우에는 먼저 구현을 하고, 구현에 대한 피드백을 수업을 통해 받는 구조가 되어버려 시간이 많이 들어갔다. 반대로 해당 과목에 대한 개념들을 먼저 수업을 통해 습득하고 과제를 수행하려고 한다면, 과제를 특정 수준 이상으로 하기 위한 절대적인 시간이 부족하였다. 이는 다음에 해결되어야 할 문제라고 느꼈다. 개인적으로는 Left Days를 한 10일 정도로 늘려는 방안이 좋아 보이지만, 이 또한 형평성 문제 때문에 많은 단점을 불러일으킬 것 같다는 생각이 든다.

이제 마지막 과제인 Cache 구현이 남아있다. 현재로서 제일 궁금한 것은 해당 개념이 어떤 방식으로 지금까지 구현한 Pipelined Micro architecture의 성능을 향상시킬 지이다. 이를 통해 차후 Cache에 대한 개념 및 실제 작동 과정을 조금 더 자세히 배우고 싶다는 생각이 들었다.