

# Hardware System Design

## Term Project v0, Interim

2013-11392 김지현

### 개요

이번 Term Project v0 중간제출은 Lab 6에서 작성했던 벡터\*벡터 내적기를, 행렬\*벡터 곱셈기로 만들고 테스트벤치를 실행하는 것이다. 행렬\*벡터 곱셈기를 처음부터 짜라고 했으면 아주 막막했겠지만, 벡터\*벡터 내적기를 이미 완성한 상태이기 때문에, PE를 여러 개로 늘리고 여러개의 벡터\*벡터 내적을 병렬로 수행하면 끝이기 때문에 크게 어렵지 않게 구현을 마칠 수 있었다.

### 구현체 설명

가장 먼저 Lab 6에서 작성한 코드를 리팩터링하는 과정부터 시작했다. Lab 6에선 state를 하나의 변수 안에 몰아넣었었는데, 이부분이 복잡한 프로그래밍을 하기에 적절하지 않아, state 변수를 세개로 분리하였다.

그 다음 Testbench를 벡터\*벡터 에서 행렬\*벡터로 고쳤다. 테스트벤치는 어차피 하는 일이 Testbench의 데이터를 my\_pectl과 연결한 뒤, start pin을 한번 켜주는 것 밖에 없어서, testbench 입력의 크기를 키우는것으로 쉽게 수정할 수 있었다. Rdaddr 입력의 비트 수 조정도 필요하다. 벡터\*벡터 곱을 행렬\*벡터 곱으로 고친 결과 테스트벤치 데이터의 크기가 훨씬 커졌기 때문이다.

그 다음 my\_pe를 16개로 늘렸다. 행렬\*벡터 곱의 좌변인 행렬의 각 행을 각각 16개의 my\_pe에 저장하고, S\_CALC state에서 16개의 행과 우변의 벡터 하나를 동시에 내적시키면 그것이 행렬곱이 되기 때문이다. 다만 my\_pectl이 테스트벤치로부터 한번에 하나의 데이터만 읽을 수 있기때문에, S\_LOAD state에서 보내는 시간이 16배로 늘었다. 그리고 행렬곱 결과를 한번에 출력할 수 있도록 하기위해, my\_pectl이 행렬\*벡터곱의 결과인 길이 16짜리 벡터를 16개의 출력 레지스터로 한번에 출력할 수 있도록 수정했다.

### 실험 결과 분석

Waveform이 너무 커서 고화질 이미지를 별도로 첨부하였으니, 아래의 이미지가 식별이 어려울 경우 첨부파일을 확인하길 바란다.

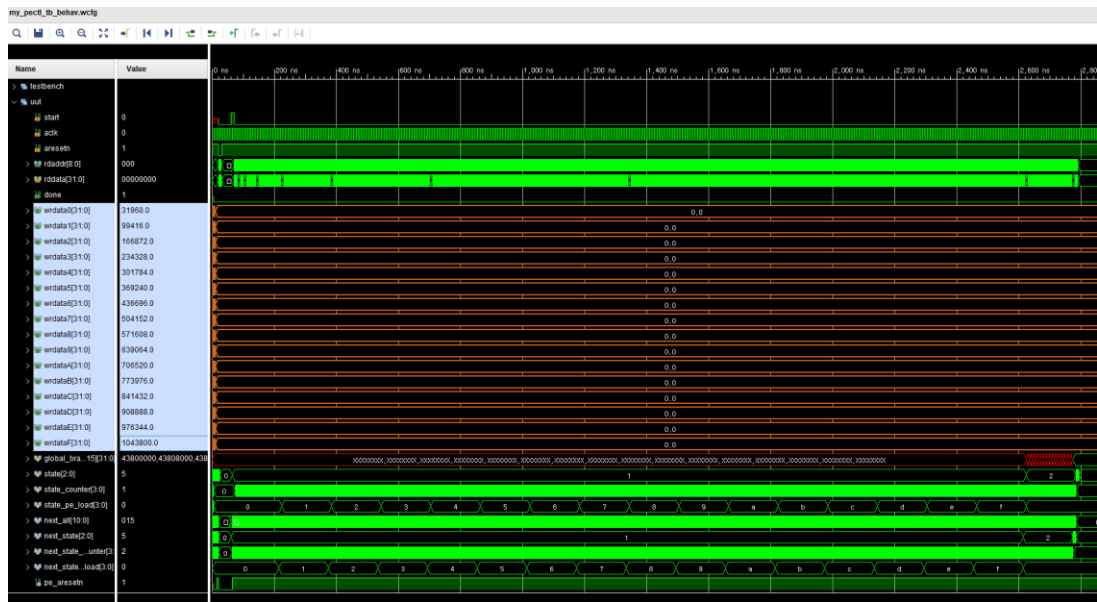


Figure 1. Waveform of S\_LOAD stage

S\_LOAD 상태에서의 waveform이다. 메모리를 순차적으로 읽으면서, 각 PE에 맞는 행렬의 각 행을 미리 입력시키는 과정이다. 행렬 데이터 크기가 크기 때문에, 데이터를 읽는데에도 많은 시간이 필요한 것을 볼 수 있다.

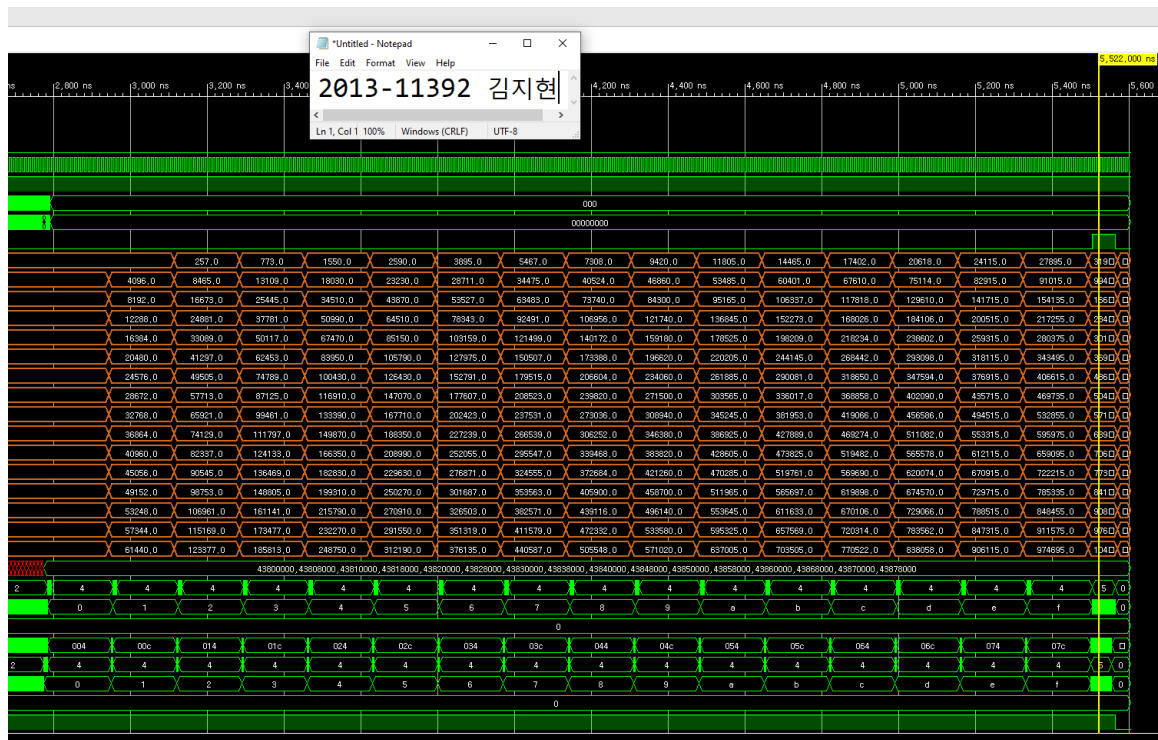


Figure 2. Waveform of S\_CALC stage

S\_CALC 상태에서의 waveform이다. 입력시켜두었던 행렬을 계산하는 모습이고, 각 PE가 자신의 행을 주어진 벡터에 하나씩 곱하며 더하는(Multiply-Accumulate) 모습을 볼 수 있다.



## 결론

Term Project의 최종목표인 행렬곱 가속기의 핵심부분을 완성하셔서 매우 기뻐다. 맨 처음 학기가 시작했을 때 뉴럴넷 가속기를 만든다고 해서 굉장히 막막했는데, 차근차근 수업과 실습을 따라가니 이런 과제도 어렵지 않게 해낼 수 있어 굉장히 재밌고 고무적이었다. 나머지 과정인 행렬곱 가속기를 FPGA에 탑재하고 quantize하는 과정 등도 기대된다.