

# 어셈블리 프로그램 프로젝트 제안서

## Bilinear Interpolation

학 과: 컴퓨터공학과

담당교수: 이준환 교수님

학 번: 2017202088

성 명: 신해담

## 1. 제목과 목표

### A. 제목

Bilinear Interpolation

### B. 목표

임의의 부동소수점 데이터로 이루어진 20x20 정방행렬에 대해 Bilinear Interpolation을 수행하여 행과 열을 4배로 확장한 80x80 정방행렬을 계산하는 어셈블리 프로그램을 구현한다.

## 2. 일정

주차	12	13	14	15
제안서				
코드 작성				
코드 검증				
결과보고서				

## 3. Function별 알고리즘

- calcRow 함수

행의 중간값을 구하는 함수. 메모리에서 한 행에 존재하는 연속하는 수를 두개씩 읽어 사이값을 계산한다. 이 때 두 수의 중간값을 구하고, 구한 중간값과 두 점과의 중간값을 구해서 총 세 개의 값을 얻는다. 짝이 없는 마지막 수는 그 수를 나열하여 해결한다.

- calcCol 함수

열의 중간값을 구하는 함수. 메모리에서 두 행에 존재하는 연속하는 수를 한개씩 읽어 사이값을 계산한다. 이 때 두 수의 중간값을 구하고, 구한 중간값과 두 점과의 중간값을 구해서 총 세 개의 값을 얻는다. 짝이 없는 마지막 행의 경우 각 행의 값을 나열하여 해결한다.

- calcNewRow 함수

calcRow, calcCol 함수로 구한 새 값들 사이의 빈 공간을 채우는 함수. calcCol로 구한 연속하는 값 두 개를 이용하여 그 사이값을 계산한다. 이 때 두 수의 중간값을 구하고, 구한 중간값과 두 점과의 중간값을 구해서 총 세 개의 값을 얻는다. 짝이 없는 마지막 수는 그 수를 나열하여 해결한다.

- calcMedian 함수

두 소수값의 중간값을 구하는 함수. 두 수의 부호가 같으면 exponent를 동일하

게 맞춘 두 mentissa의 값을 ADD연산한 다음 오른쪽으로 한번 shift하여 사잇값을 구한다. 두 수의 부호가 다르면 mentissa를 SUB한 다음 오른쪽으로 shift한다.

#### 4. 예상되는 문제점

State를 줄이기 위해서 loop unrolling을 사용하려면 레지스터 사용을 최소한으로 해야 한다. calcMedian 함수는 소숫값을 다루므로 레지스터 사용량이 많으므로, 레지스터가 행렬의 값을 최대한 많이 가지도록 하기 위해서는 calcMedian을 최소한의 레지스터를 사용하도록 코드를 짜야한다.

행렬의 중간값을 계산하는 함수가 짝이 없는 외각수의 위치를 찾기 위해 loop 안에서 conditional instruction을 사용하면 그만큼 state를 소모하게 된다. 이로 인해 생기는 성능 저하를 최소화하기 위한 구조를 고려해야한다.

한번에 전체의 row/col을 구하는 방법과 매 줄마다 row/col을 구하는 방법, 2x2의 한 블록마다 row/col을 구하는 방법 중 가장 효율이 좋은 방법이 무엇인지 충분히 생각하고 진행해야 성능이 더 좋은 프로그램을 구현할 수 있다.

#### 5. 검증 전략

calcMedian에 임의의 소숫값을 입력하여 중간값이 제대로 구해지는지 확인한다.

calcRow와 calcCol 함수에서 중간값을 구하지 않고 메모리에 저장된 소수들을 제대로 읽어올 수 있는지 확인한다.

최종적으로 Bilinear Interpolation 프로그램을 구현하여 calcNewRow 함수 및 전체 함수가 제대로 연결되어 역할을 수행함을 검증한다.