**어셈블리프로그램 설계 및 실습 – term project**

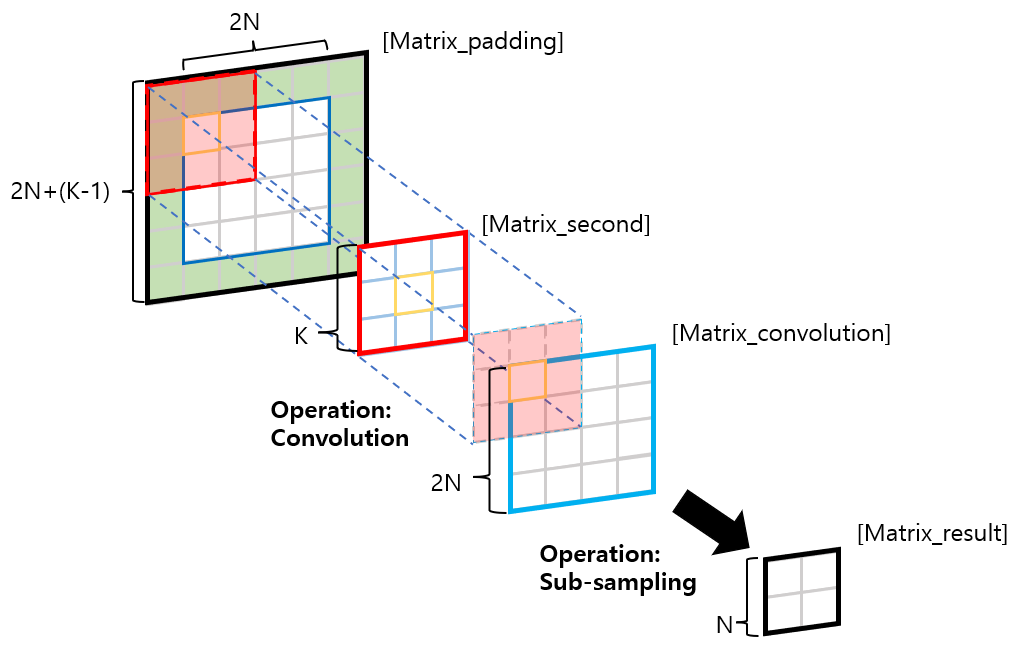
**Matrix convolution**

“꿈을 가지십시오. 그리고 정열적이고 명예롭게 이루십시오.”

이번 학기 설계 과제는 임의의 부동소수점 데이터로 이루어진 2N-by-2N 행렬에 **padding**을 수행한 뒤 **convolution** 및 **sub-sampling**을 수행한 N-by-N의 행렬을 출력하는 어셈블리 코드를 작성하는 것이다. 과제의 자세한 specification을 충분히 이해하여 과제를 수행하도록 한다.

1. **System Overview**

프로젝트의 전체 동작은 다음과 같다. 임의의 부동소수점 데이터로 이루어진 2N-by-2N 행렬**[Matrix\_first]**과 K-by-K 행렬**[Matrix\_second]**가 주어진다. **[Matrix\_first]**에 **[Matrix\_second]**을 convolution 하기위해 **[Matrix\_first]**를 padding하여 **[Matrix\_padding]**을 생성한다. **[Matrix\_padding]**에 **[Matrix\_second]**를 convolution한 2N-by-2N 행렬**[Matrix\_convolution]**에 sub-sampling을 수행하여 최종적으로 N-by-N의 행렬**[Matrix\_result]**를 구한다. 본 프로젝트의 목표는 Matrix\_result를 연산하는 과정에서, 성능이 가장 좋은 코드를 구현하는 것이다. 이를 위해 필요한 알고리즘의 사용에는 제한이 없다. 성능의 기준은 code size와 state2을 곱한 값으로, 결과 값이 작을수록 성능이 좋다고 정의한다.



**Fig1. System Overview**

* *구체적인 설계 사양은 업데이트 될 수 있습니다. 업데이트 시 KLAS의 해당 과목 공지사항에 공지할 테니 확인하시기 바랍니다.*

1. **Specification**

**주어진 Matrix\_first와 Matrix\_second의 연산을 통해 Matrix\_result를 생성하는 과정은 다음과 같다.**

* **Step 1. Padding (2.2.1절)**
* **Step 2. Convolution (2.2.2절)**
* **Step 3. Sub-sampling (2.2.3절)**

**각 step에 대한 상세 동작은 2.2절에 예시와 함께 설명되어 있다.**

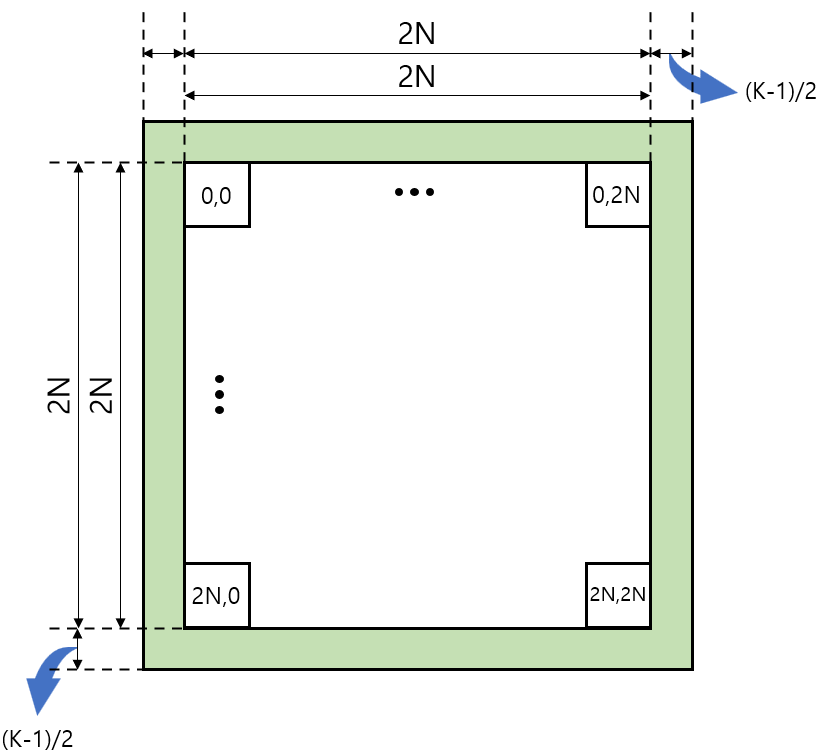
* 1. **Project specification**
* Project conditions
  + Sub-sampling scale
    - 행과 열 각 배
    - Sub-sampling 과정에서 선택되는 data는 Fig4와 동일
  + Boundary exception(padding method)
    - Matrix의 가장자리(boundary)의 경우 padding을 적용하여 convolution을 수행
    - Padding하여 생성될 데이터와 위치가 가장 가까운 데이터의 값으로 padding 수행
  1. **Functions**
     1. **Padding**

**- Input :** 2N-by-2N 행렬**[Matrix\_first]**

**- Output :** 2N+(K-1)-by-2N+(K-1) 행렬**[Matrix\_padding]**

**- Operation :** 입력된 2N x 2N의 크기를 가지는 행렬의 가장자리(boundary)의 경우 위치가 **가장 가까운 값을** 복사하여 2N+(K-1)-by-2N+(K-1)의 크기를 가지는 행렬을 Output으로 한다.

**Figure 2.**는 임의의 N과 K값에 대하여, Matrix\_first에 padding을 해야 하는 크기가 K크기에 의하여 (K-1)/2로 정해지는 것을 나타낸 것이다. Matrix\_first의 상하좌우에 각각 (K-1)/2만큼씩 padding을 수행한다.



**Fig2. 임의의 N, K 값에 대한 padding 예제**

**Figure 3.**는 N과 K의 값이 각각 2, 3인 경우, Matrix\_first에 padding을 수행하는 것을 나타낸 것이다. K가 3인경우 (K-1)/2가 1이므로 Matrix\_first의 상하좌우에 1의 크기만큼 padding을 수행하여야 한다.

테이블이(가) 표시된 사진

자동 생성된 설명

**Fig3. N=2, K=3인 경우에 대한 padding 예제**

* + 1. **Convolution**

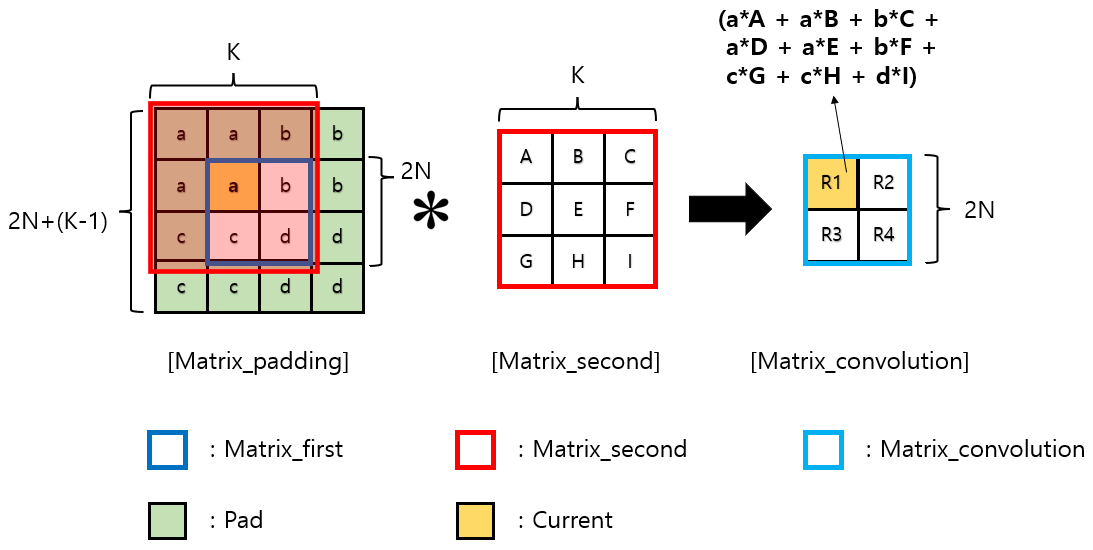
**- Input :** 2N+(K-1)-by-2N+(K-1) 행렬**[Matrix\_padding]**

**:** K-by-K 행렬**[Matrix\_second]**

**- Output :** 2N-by-2N 행렬**[Matrix\_convolution]**

**- Operation :** Padding의 출력인 2N+(K-1)/2-by-2N+(K-1)/2 크기 행렬의 a, b, c, d위치에서 K-by-K 크기의 행렬을 figure 4의 연산방식과 동일하게 연산하여 2N-by-2N 행렬을 Output으로 한다.

**Figure 4.**는 N과 K의 값이 각각 1, 3인 경우, Matrix\_padding에 Matrix\_second를 convolution하는 것을 나타낸 것이다. Matrix\_second의 중심(E)을 Matrix\_padding에서 pad 부분을 제외한 Matrix\_first영역에 위치하여 연산을 수행한 결과로 이루어진 Matrix\_convolution을 Output으로 한다. 이때, 연산을 수행한 결과가 Matrix\_convolution에서 위치하는 좌표는 Matrix\_second의 중심(E)이 위치한 Matrix\_first의 좌표와 동일하다. 예를 들어 **Figure 4.**와 같이 Matrix\_second의 중심(E)이 위치한 좌표가 Matrix\_first의 (0,0)인 경우, 해당 연산결과는 Matrix\_convolution의 (0,0)에 위치하게 된다.



**Fig4. N=1, K=3인 경우에 대한 convolution 예제**

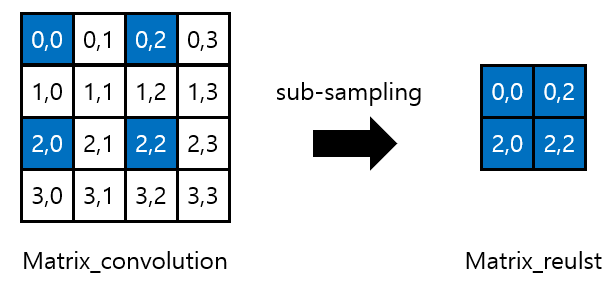
* + 1. **Sub-sampling**

**- Input**  **:** 2N-by-2N 행렬**[Matrix\_convolution]**

**- Output :** N-by-N 행렬**[Matrix\_result]**

**- Operation :** 입력 행렬을 1/2의 크기로 Sub-sampling 하기 위해서 짝수행과 짝수열의 데이터만을 선택하여, 선택된 데이터를 figure 5와 같이 Sub-sampled 행렬로 구성하여 출력한다.

Figure 5.는 N의 값이 2인 경우, Matrix\_convolution에 짝수행, 짝수열에 대한 sub-sampling을 수행하는 것을 나타낸 것이다. 이때, Matrix\_convolution에서 선택된 짝수행, 짝수열에 위치하는 값들이 Matrix\_result에서 위치하는 좌표는 Matrix\_convolution에서의 좌표(x, y)의 x와 y를 각각 2로 나눈 (x/2, y/2)가 된다. 예를 들어 **Figure 5.**와 같이 Matrix\_convolution의 짝수행, 짝수열들{(0, 0), (0, 2), (2, 0), (2, 2)}은 각각 Matrix\_result의 {(0, 0), (0, 1), (1, 0), (1, 1)}에 위치하게 된다.



**Fig5. 짝수행, 짝수열에 대한 Sub-sampling 예시**

* **본 프로젝트의 경우 N=32, K=3로 고정**
* **Label: Matrix\_first**
  + 임의의 데이터(2N-by-2N)는 0 ~ 255 사이의 정수 값을 원소로 가짐
* **Label: Matrix\_second**
  + Filter의 weight값(K-by-K)은 0~1사이의 부동소수점 값(실수, floating point number)을 원소로 가짐
* **Label: Matrix\_result**
  + Matrix\_first와 Matrix\_second의 원소들을 이용하여 연산을 수행한 결과(N-by-N)가 저장되는 데이터 공간의 시작 주소 값
  + 데이터를 저장할 때에는 4byte 단위로 저장하며, 0x60000000 번지로 통일
* **데이터는 별도의 첨부파일로 제공**
  + 제공하는 첨부파일을 코드에 복사
    - 모든 데이터의 index는 행,열 순서

텍스트, 전자기기이(가) 표시된 사진

자동 생성된 설명 테이블이(가) 표시된 사진

자동 생성된 설명

* **Floating point관련 instruction 사용 금지**
* **MUL instruction 사용금지, Booth 알고리즘을 활용하여 구현**
* **주석 필수 작성(미작성시 감점사유)**
* **프로그램은 주석을 포함하여 다음과 같이 종료**
  + MOV pc, #0 ;Program end

1. **Important dates**

* Issue date: 11월 16일
* 제안서 제출
  + Softcopy: 11월 23일 23시 59분까지 종합정보서비스 과제 제출에 제출
* 최종 결과 보고서 제출
  + Softcopy: 11월 30일 23시 59분까지 종합정보서비스 과제 제출에 제출
  + 늦은 제출은 12월 01일 23시 59분까지
* Project **제안서** 발표: 11월 25일 설계 수업시간
  + 발표 희망자는 김민섭 조교에게 메일 보낼 것(minsub20@kw.ac.kr). (선착순)
* Project **결과 보고서** 발표: 12월 02일 설계 수업시간
  + 발표 희망자는 김민섭 조교에게 메일 보낼 것(minsub20@kw.ac.kr). (선착순)

1. **Report outline**
   1. 제안서

* 보고서: 최소한 다음의 내용들이 포함되어야 하며 그 외의 것을 추가하는 것은 자유
  + 과제 제목, 과제 목표, 일정, 각 function 별 알고리즘, 예상되는 문제점, 검증 전략 등
  1. 결과 보고서
* 발표: PowerPoint로 발표 희망자에 한해 10분 분량으로 작성
* 보고서: 최소한 다음의 내용들이 포함되어야 하며 그 외의 것을 추가하는 것은 자유

1. Introduction

일정 및 계획을 포함할 것

1. Project Specification
2. Algorithm
3. Performance & Result
4. Conclusion

과제 완료 후 기대되는 학습 효과를 포함할 것

* **보고서의 가장 ideal한 양식은 논문의 형태입니다.**

1. **프로젝트 점수 분배**

* 제안서: 20%
* 코드: 30%
* Performance: 20%
* 결과보고서: 30%