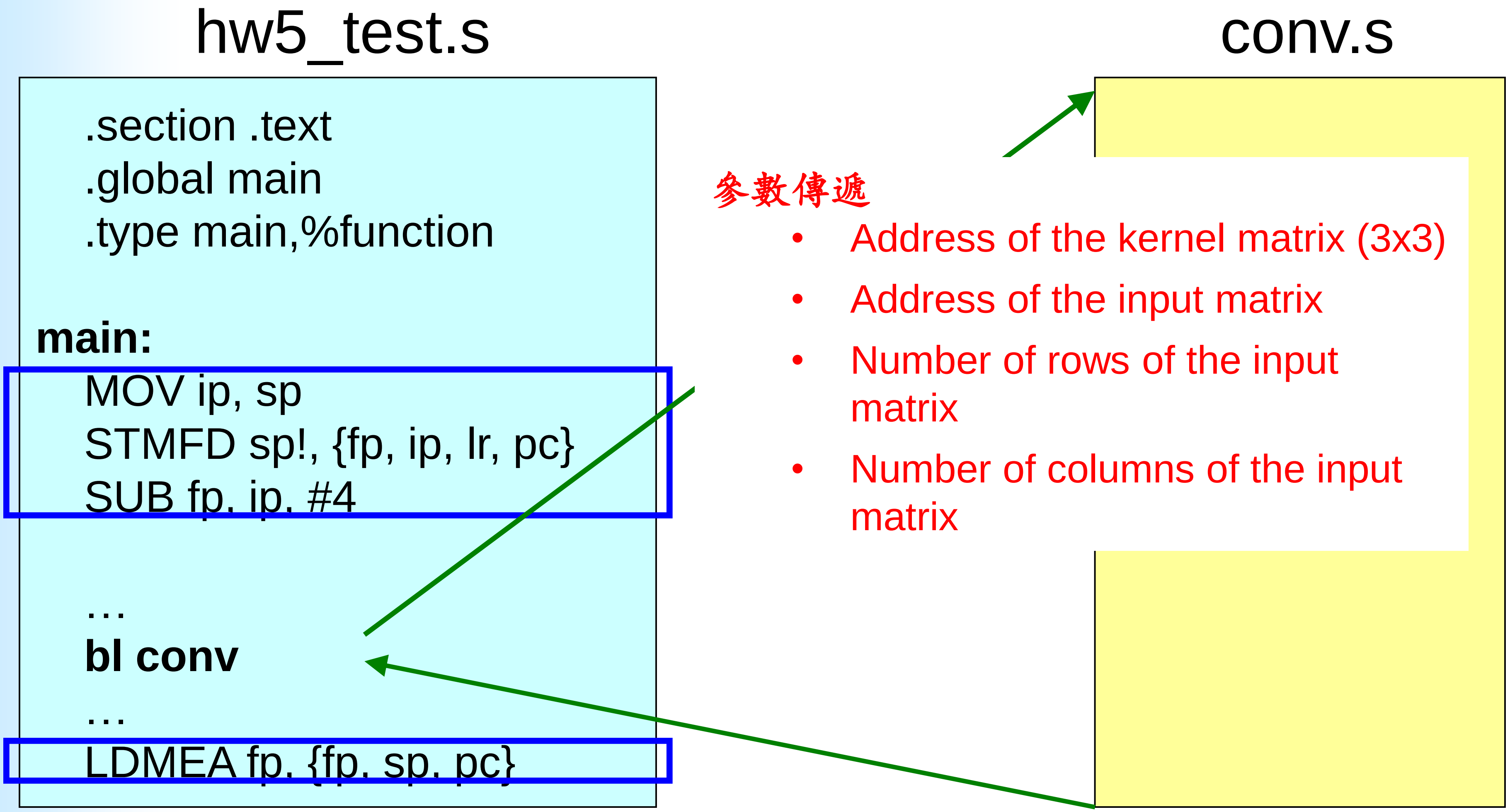
Homework #5 (1)

- Use ARM assembly to write a function called **conv** that does the convolution. **(請參閱作業4)**
- Function **conv**: 4 parameters (遵守APCS規則)
  - Address of the kernel matrix (3x3)
  - Address of the input matrix
  - Number of rows of the input matrix
  - Number of columns of the input matrix
- Function **conv**: return value (遵守APCS規則)
  - Address of the first element of the output matrix

Input matrix預設為4x5 matrix.



hw5_test.s

```
    .section .text
    .global main
    .type main,%function

main:
    MOV ip, sp
    STMFD sp!, {fp, ip, lr, pc}
    SUB fp, ip, #4

    …
    bl conv
    …
    LDMEA fp, {fp, sp, pc}
```

conv.s

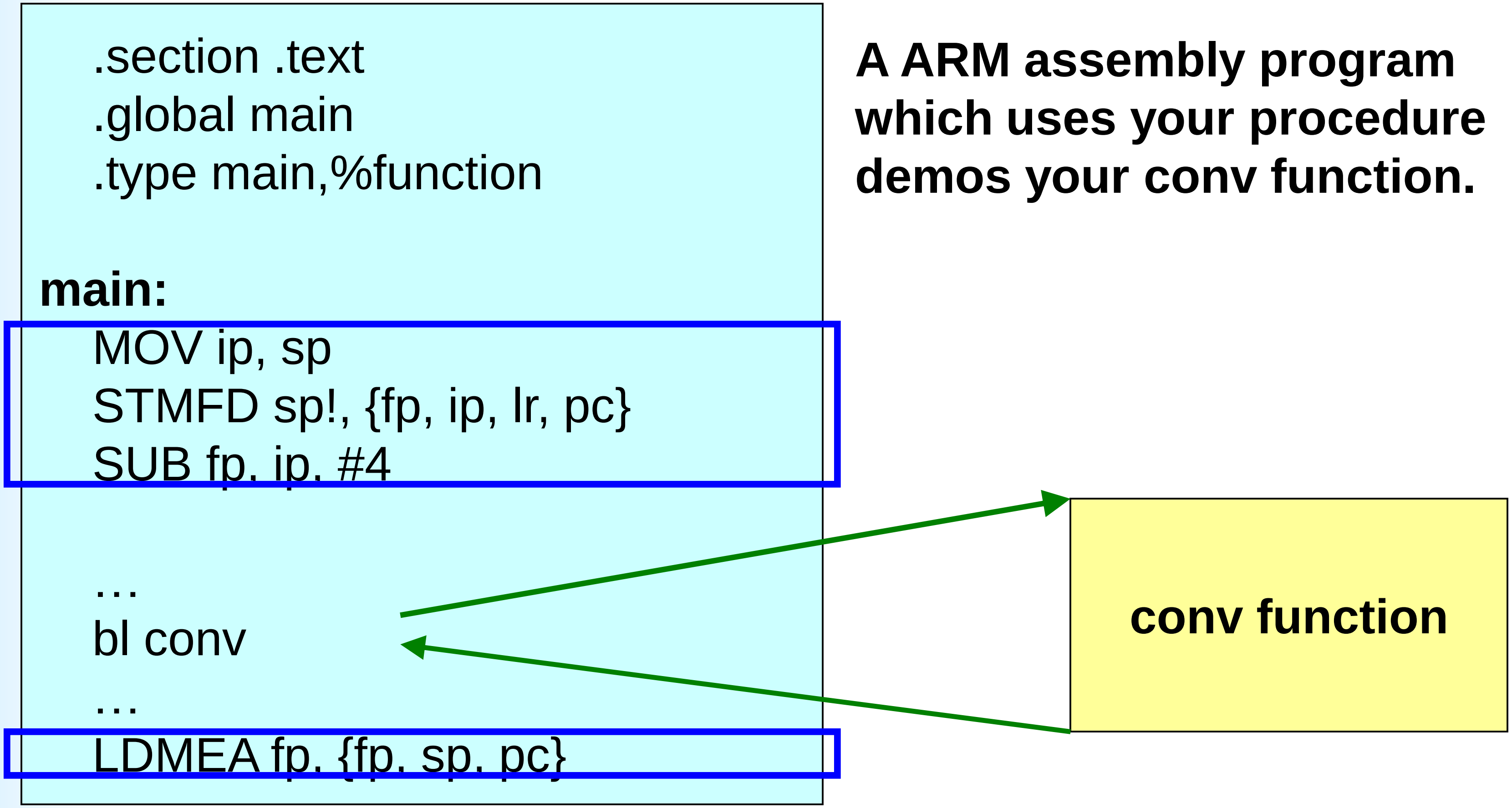
參數傳遞

- Address of the kernel matrix (3x3)
- Address of the input matrix
- Number of rows of the input matrix
- Number of columns of the input matrix

傳回值: Address of the first element of the output matrix

Homework #5 (2)

A ARM assembly program which uses your procedure demos your conv function.



```
    .section .text
    .global main
    .type main,%function

main:
    MOV ip, sp
    STMFD sp!, {fp, ip, lr, pc}
    SUB fp, ip, #4

    …
    bl conv
    …
    LDMEA fp, {fp, sp, pc}
```

hw5_test.s

```
 .section .text
        .global main
        .type main,%function
main:
        MOV ip, sp
        STMFD sp!, {fp, ip, lr, pc}
        SUB fp, ip, #4


        /* --- begin of your function ---
        r0 <= address of the kernel matrix
        r1 <= address of the input matrix
        r2 <= Number of rows of the input matrix
        r3 <= Number of columns of the input matrix */

        bl  conv
        nop
        /* --- end of your function --- */



        LDMEA fp, {fp, sp, pc}
        .end
```

傳回值output matrix 的位址存放在r0

Homework #5 (3)

conv.s

```
        .section .text
        .global conv
        .type conv,%function

conv:
        /* function start */
        MOV ip, sp
        STMFD sp!, {r4-r11, fp, ip, lr, pc}
        SUB fp, ip, #4

        /* --- begin your function --- */
        /* 傳入值會放在r0, r1, r2, r3 */

        /* call malloc() for memory space of the output matrix */

        /* DO convolution */

        /* 把傳回值 (output的位址) 放在r0 */
        /* --- end of your function --- */
        /* function exit */
        LDMEA fp, {r4-r11, fp, sp, pc}
        .end
```

請留意callee saved registers

參數傳遞

Do convolution

How to Compile Your Program?

```
$ arm-non-eabi-gcc -g -O0 hw5_test.s conv.s -o hw5.exe
```

Homework #5 (4)

- Program should be assembled and linked by gcc
  - 使用於作業一所安裝完成的cross compiler與cross binutils
- Program should be executed under **GDB ARM simulator**
- 程式中應有適當的說明（註解）
- 程式應遵守APCS規則
- You should turn in to **ECOURSE2**
  - “**README.txt**” file: 文字檔，描述你程式的內容、如何編譯程式、如何執行你的程式
  - Your ARM assembly procedure，**檔名為：conv.s**
  - An ARM assembly program which uses your gcd function，**檔名為：hw5_test.s**
  - Makefile
  - Any file needed in your work

Homework #5 (5)

- 請勿繳交【利用編譯器所自動產生的組合語言程式】
- 請勿抄襲
- 請將欲繳交的檔案壓縮成 <hw5_學號.tar.bz2>, 上傳壓縮檔
- Deadline: December 8 (Sunday), 2019