# MPSL2019

Lab2

# Lab 2.1 Postfix arithmetic

- Setup stack pointer
- Calculate postfix expression with stack
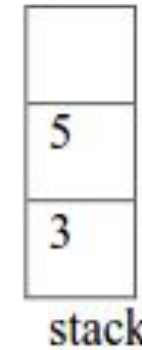
| 0x20000078 | 3 | PUSH |
| 0x20000074 | 5 | ← SP |
| 0x20000070 | | |
| 0x2000006C | | |
| 0x20000068 | | |
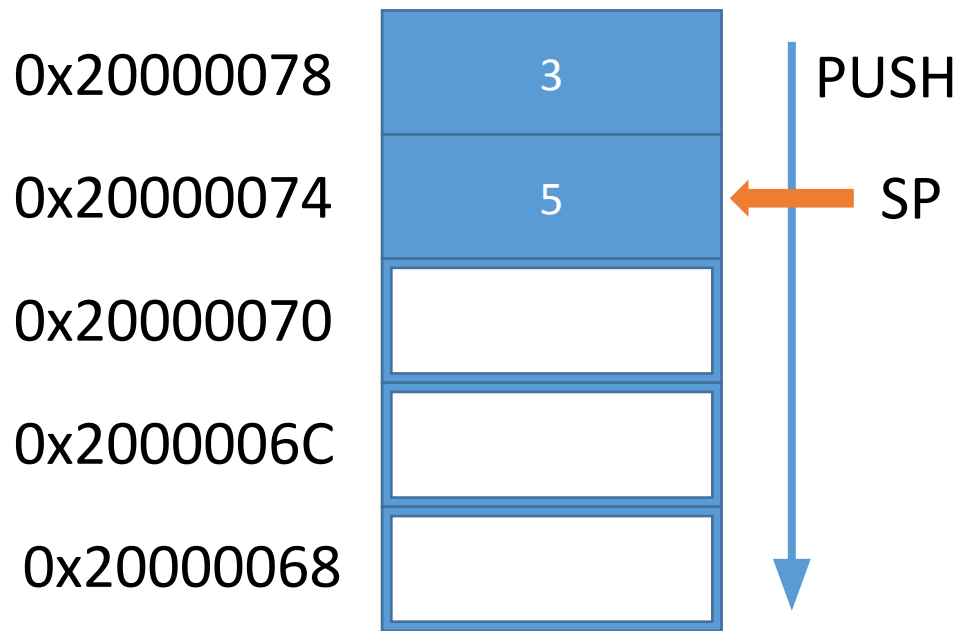
The processor uses a full descending stack.

(1) 3, 5, 4, -, +

| |
| 3 |

stack

(2) 3, 5, 4, -, +

| |
| 5 |
| 3 |

stack

(3) 3, 5, 4, -, +

| 4 |
| 5 |
| 3 |

stack

(4) 3, 5, 4, -, +

| |
| 1 |
| 3 |

stack

(5) 3, 5, 4, -, +

| |
| |
| 4 |
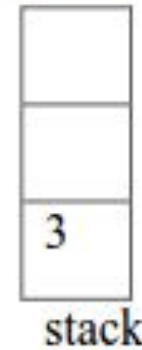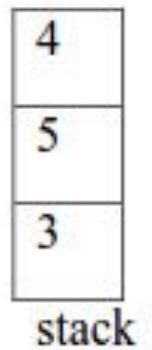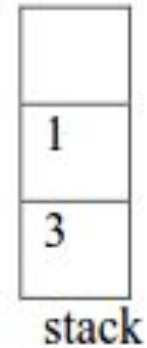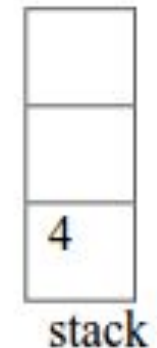
stack

# Lab 2.2 Calculating Greatest Common Divisor

- Use Stein's algorithm to find GCD
- Calculate the maximum stack size used in program



Function/subroutine call
PC changed to "FunctionA", and LR changed to address of instruction after BL

...

...

BL FunctionA

...

...

FunctionA

PUSH {R4-R6, LR}

...

BL FunctionB

...

POP {R4-R6, PC}

Save return address for FunctionA and registers to stack.

Function/subroutine call
PC changed to "FunctionB", and LR change to address of instruction after "BL FunctionB". If LR was not saved the return address for FunctionA would be lost.

FunctionB

...

BX LR

Return
PC changed to value stored in LR to resume execution of instructions after BL FunctionB

Return
Registers are restored and PC changed to value stored in stack to resume execution of instructions after BL FunctionA
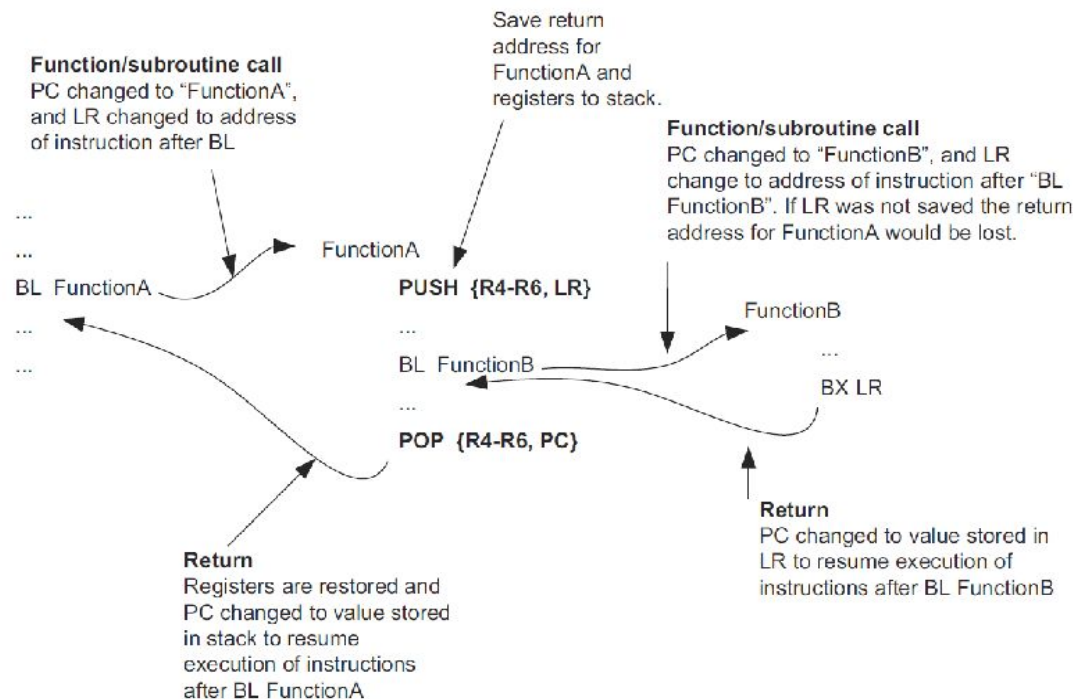
**Figure 6.3:**
Using push and pop of multiple registers in functions.

| $u_0$ | $v_0$ | $u_1=$ $u_0/2$ | $v_1=$ $v_0/2$ | $u_2=$ $u_1/2$ | $v_2=$ $v_1/2$ | $u_3=$ $u_2$ | $v_3=$ $v_2/2$ | $u_4=$ $(u_3v_3)/2$ | $v_4=$ $v_3$ |
|---|---|---|---|---|---|---|---|---|---|
| =36 | =24 | =18 | =12 | =9 | =6 | =9 | =3 | =3 | =3 |
| (even) | (even) | (even) | (even) | (odd) | (even) | (odd) | (odd) | | |

$k?$is now 1     $k?$is now 2

$u=v=3$
STOP