ΕΠΛ221:
Οργάνωση Υπολογιστών και Συμβολικός Προγραμματισμός

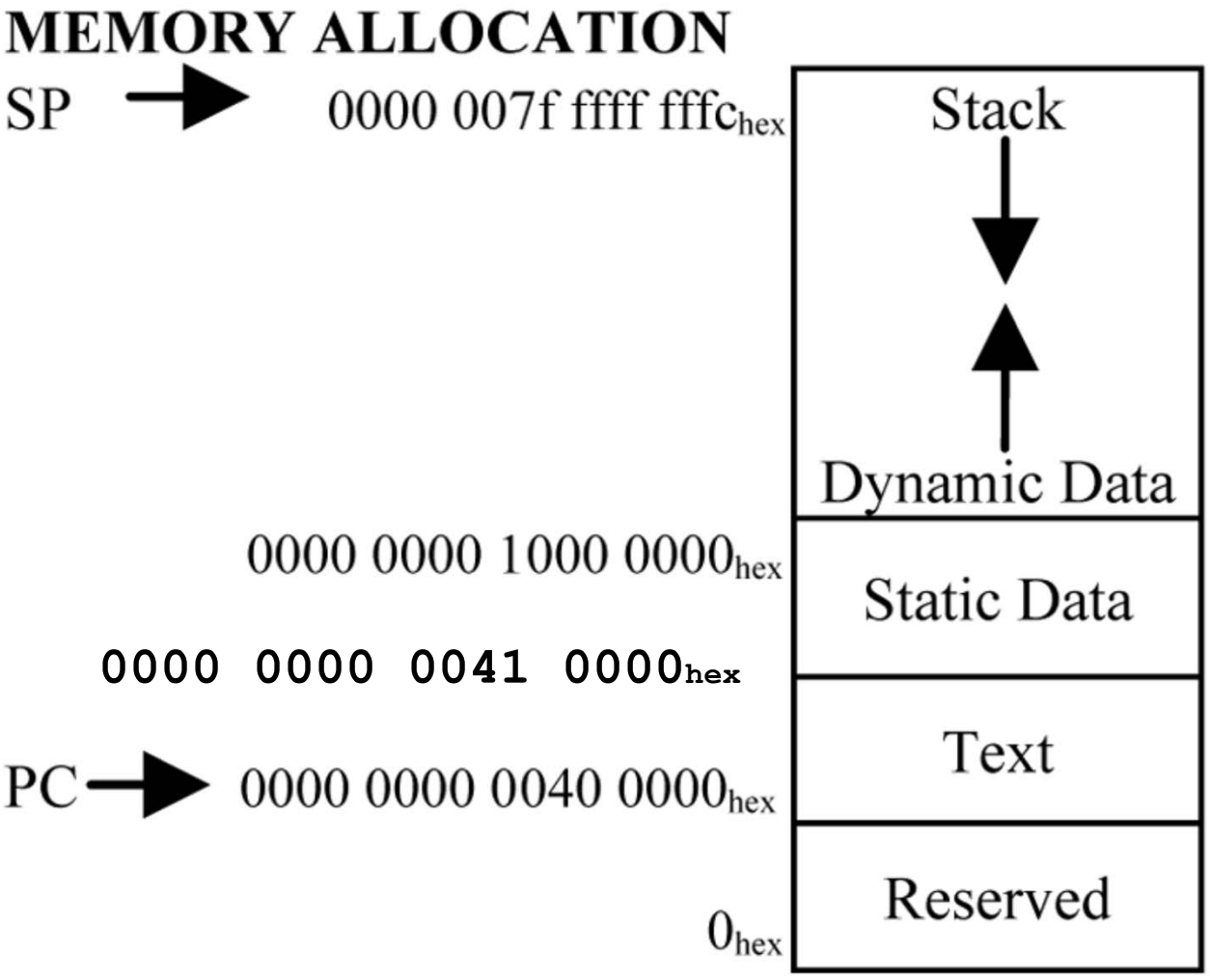
Εργαστήριο Αρ. 5

Εισαγωγή στην Αρχιτεκτονική
ARMv8-A

Στοίβα, Υπόλοιπο

Πέτρος Παναγή, PhD

Memory Allocation LEGv8



Stack Pointer

Stack Pointer: when used as a load/store base register, and in a small selection of arithmetic instructions, register number 31 provides access to the *current stack pointer*.

Size (bits)	32b	64b
Name	WSP	SP

When the base register is SP the stack pointer is required to be quadword (16 byte, 128 bit) aligned prior to the address calculation and write-backs – misalignment will cause a stack alignment fault. The stack pointer may not be used as an index register.

Load/store addressing modes in the A64 instruction set broadly follows T32 consisting of a 64-bit base register (Xn or SP) plus an immediate or register offset.

Type	Immediate Offset	Register Offset	Extended Register Offset
Simple register (exclusive)	[base{,#0}]	n/a	n/a
Offset	[base{,#imm}]	[base,Xm{,LSL #imm}]	[base,Wm,(S|U)XTW {#imm}]
Pre-indexed	[base,#imm]!	n/a	n/a
Post-indexed	[base],#imm	n/a	n/a
PC-relative (literal) load	label	n/a	n/a

A "register offset" means that the memory address is the base register value plus the value of 64-bit index register Xm optionally scaled by the access size (in bytes), i.e. shifted left by log2(size).

Άσκηση 1

Με την βοήθεια την στοίβας αντιστρέψτε μία συμβολοσειρά στη μνήμη και τυπώστε την στην οθόνη με το printf

```
./a.out
!!! yaD ecin a evaH dna 122LPE ot emocleW
```

Βασική Ιδέα Υλοποίησης

```
    adr     x0, message_str     // Load the address of message_str
    mov     x19, x0             // Keep one copy of X0 to read forward the string
    mov     X20, x0             // Keep a second copy of X0 for writing backward



    add     sp, sp, -256            // Move SP 256 bytes Down

    mov     x21, sp                 // The pointer to the writting place
    strb    wzr, [x21], 1           // Store a zero as the stack top character
loop_1:
    ldrb    w1, [x19], 1            // Load a character from the message_str, X19++
    cbz     w1, cont                // Until the character is zero '\0'
    strb    w1, [x21], 1            // Store the character into stack, X21++
    b loop_1
```

Άσκηση 2

Υπολογισμός Υπολοίπου

Βασική ιδέα:

```
Πηλίκο = διαιρετέος / διαιρέτη
Υπόλοιπο = διαιρετέος – (Πηλίκο * διαιρέτη)
```

Παράδειγμα:

103/25=2 → 2*25=100 → 103-100=3

```
./a.out
Enter the value of dividend: 103
Enter the value of divisor: 25
The Remainder of the division 103/25 is 3.
```