ΕΠΛ221:
Οργάνωση Υπολογιστών και Συμβολικός Προγραμματισμός

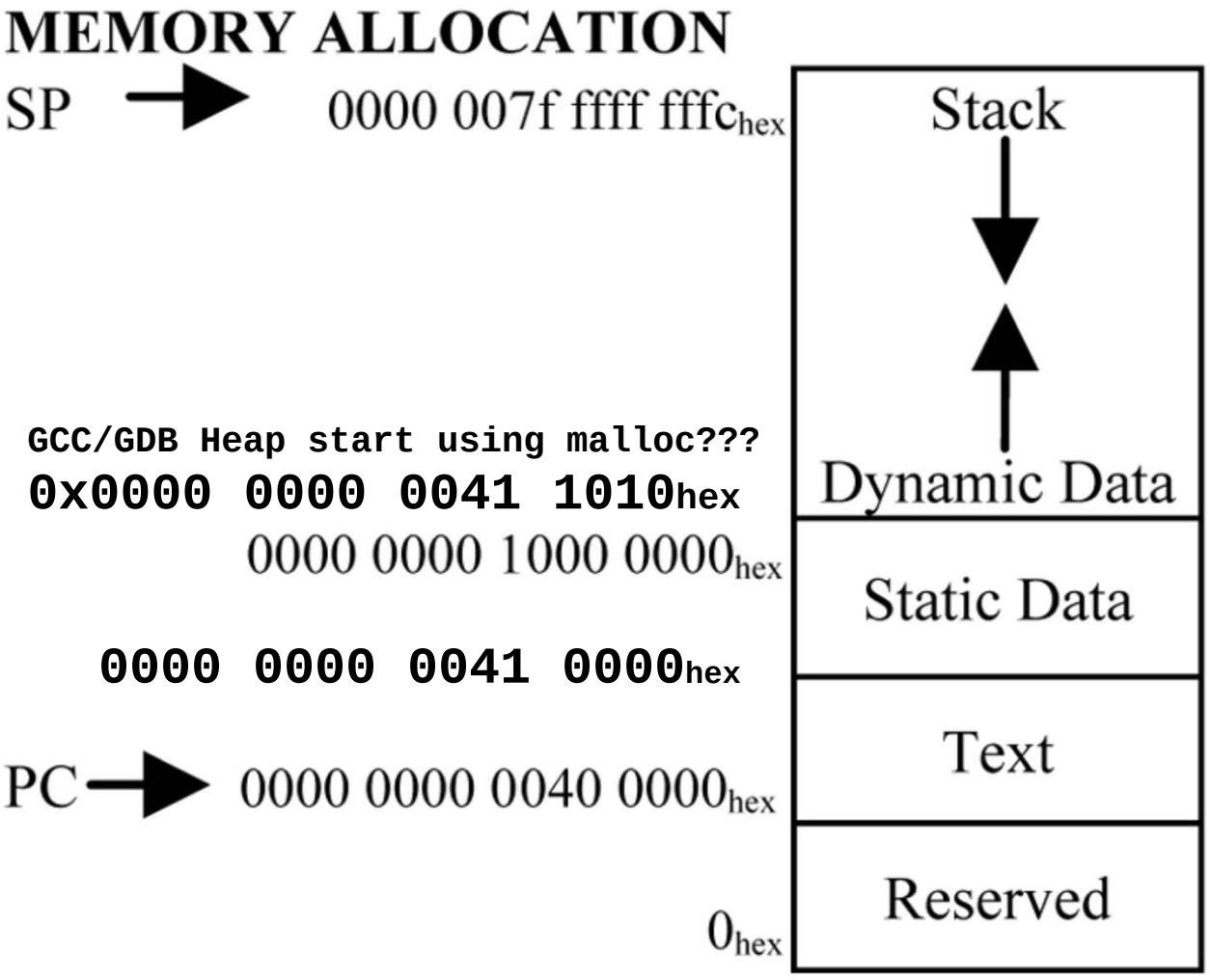
Εργαστήριο Αρ. 6

Εισαγωγή στην Αρχιτεκτονική
ARMv8

Leaf Functions and Memory Allocation

Πέτρος Παναγή, PhD

Memory Allocation LEGv8



Integer Registers used for Instructions

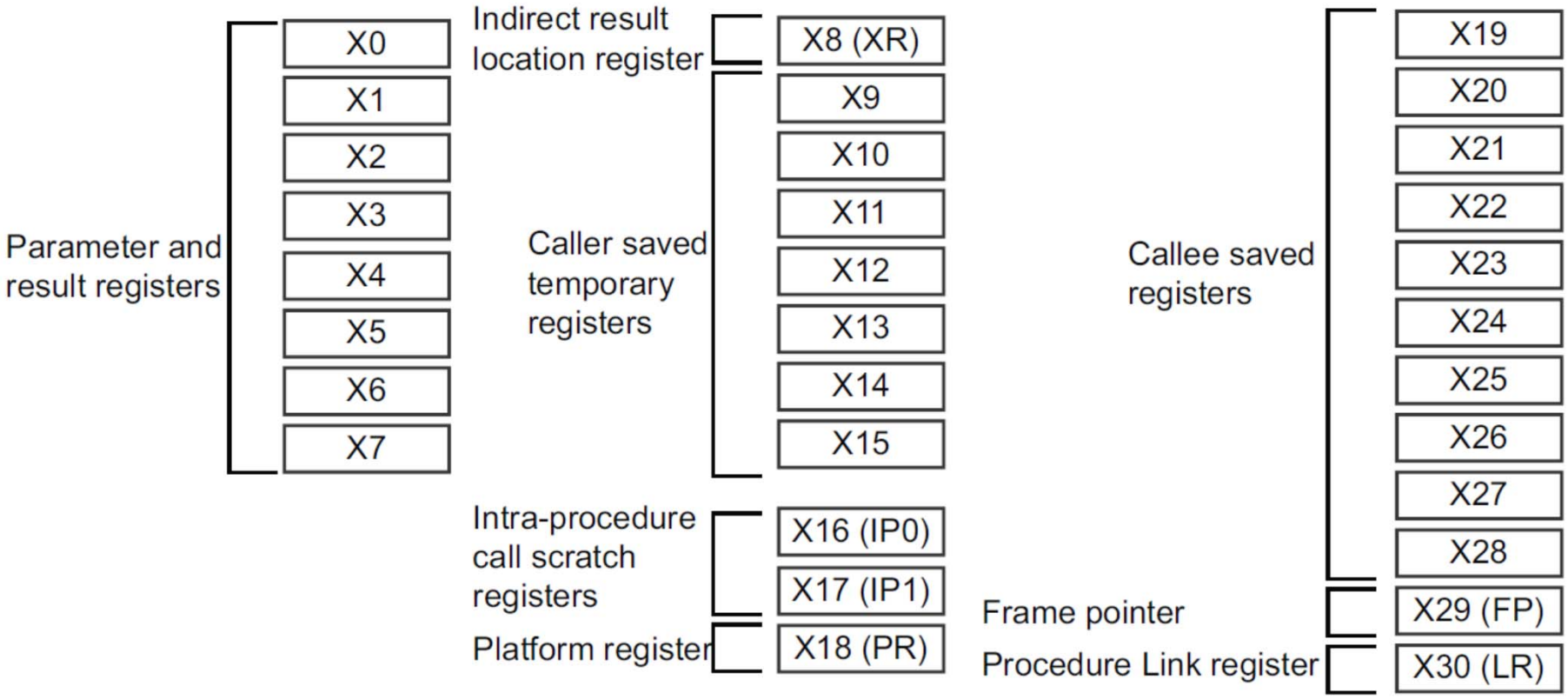


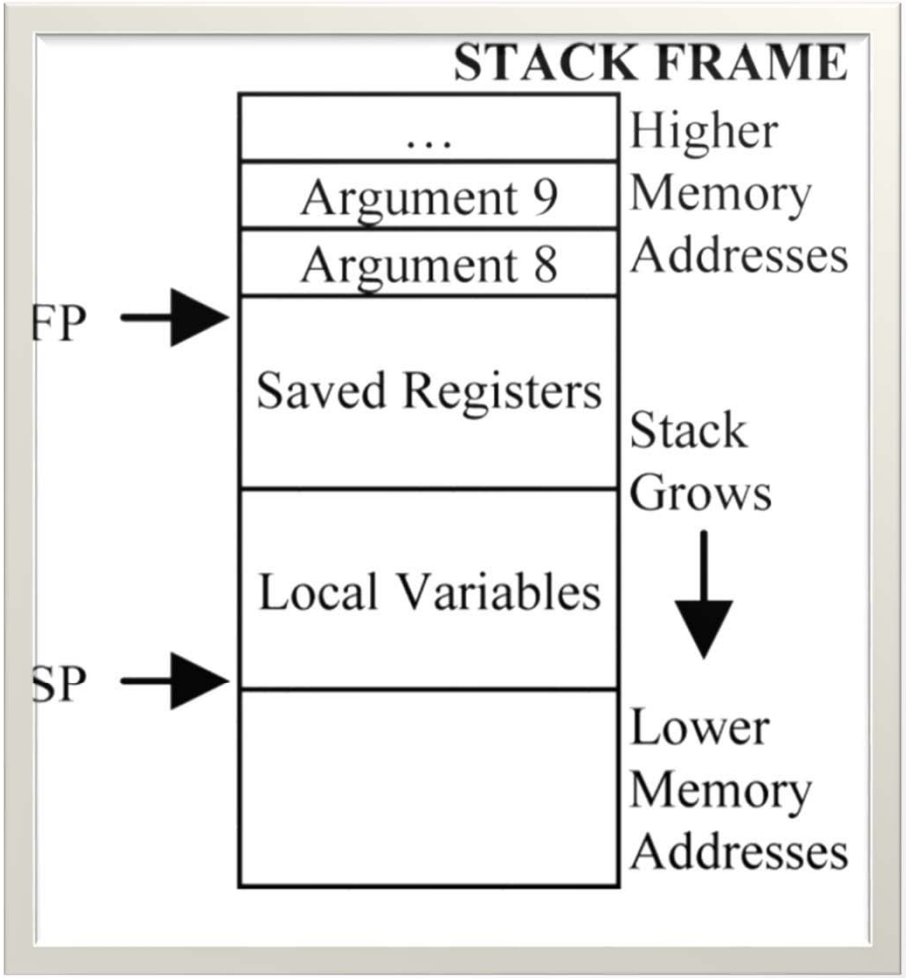
Figure 9-1 General-purpose register use in the ABI

Integer Registers used for Instructions

Register	Special	Role in the procedure call standard
SP		The Stack Pointer.
r30	LR	The Link Register.
r29	FP	The Frame Pointer
r19…r28		Callee-saved registers
r18		The Platform Register, if needed; otherwise a temporary register. See notes.
r17	IP1	The second intra-procedure-call temporary register (can be used by call veneers and PLT code); at other times may be used as a temporary register.
r16	IP0	The first intra-procedure-call scratch register (can be used by call veneers and PLT code); at other times may be used as a temporary register.
r9…r15		Temporary registers
r8		Indirect result location register
r0…r7		Parameter/result registers

Table 2. General purpose registers and AAPCS64 usage

Stack Frame Layout



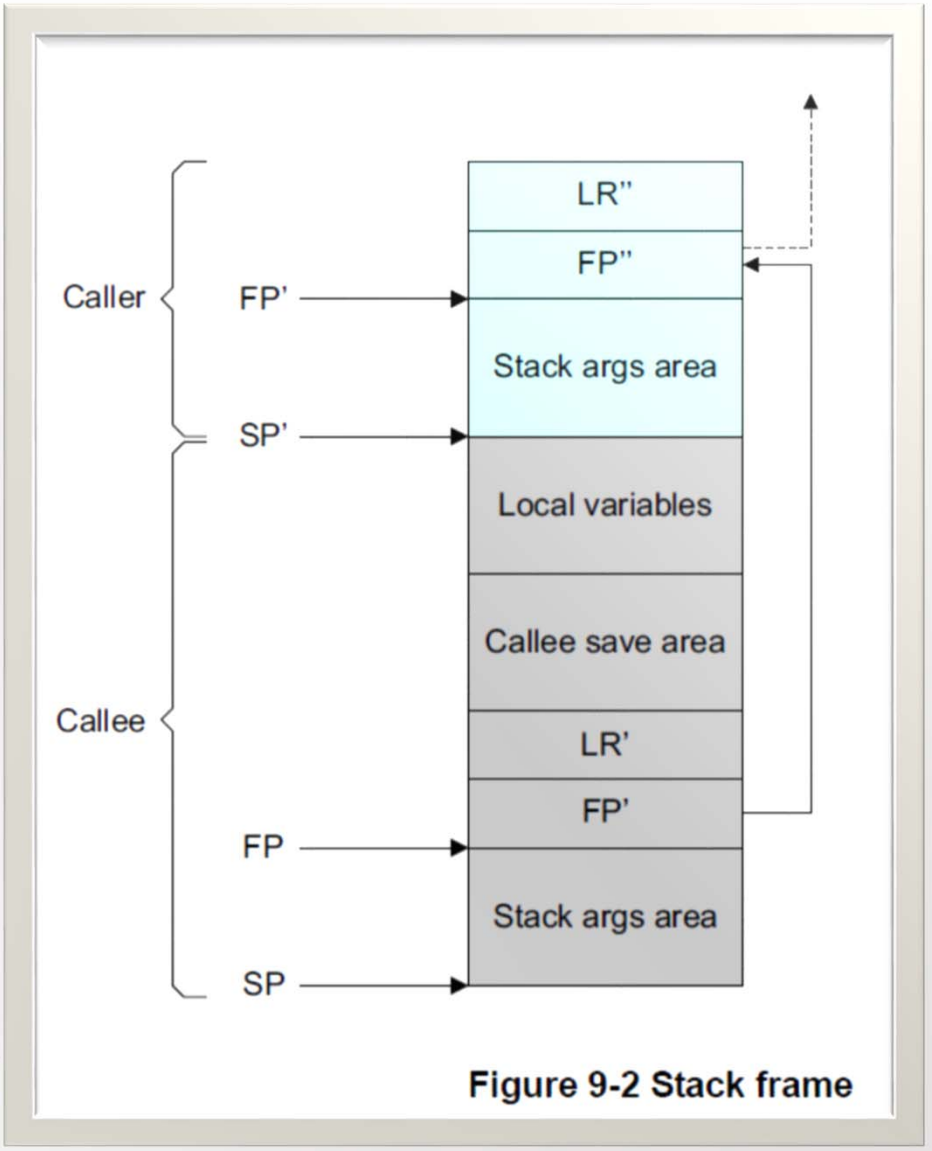


Figure 9-2 Stack frame

Stack Frame Layout

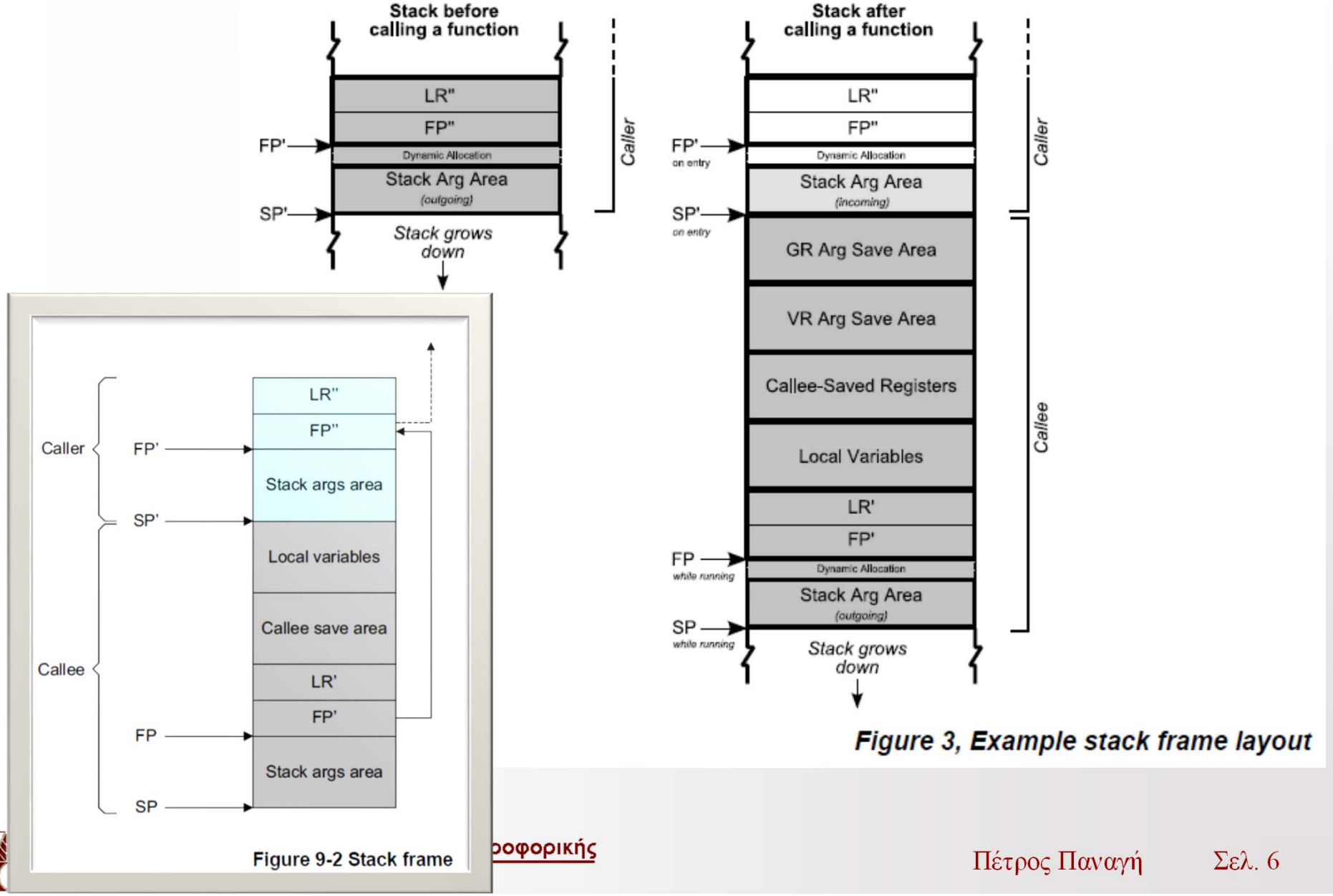


Figure 3, Example stack frame layout

Exercises

Lab6_exercise1.s (scanf and prinft as functions)

```
bl scanf_int
bl printf_int

bl scanf_lld
bl printf_lld
```

Exercises

Working with Functions and Tables

Lab6_exercise2.s (Static Global Allocation in .data)

```
int table[128];
void main(void){
```

Lab6_exercise3.s (Static Local allocation in main)

```
void main(void){
int table[128];
```

Lab6_exercise4.s (Dynamic Allocation with malloc)

```
void main(void){
int * table = (int *) malloc(sizeof(int)*128);
…
free(table)
```