

CS 315-02 Lab RISC-V Assembly 4

Functions Continued

RISC-V Registers 64 bits
 8 bytes

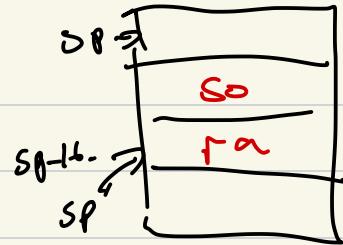
X0	zero	always zero (0)
X1	ra	return address
X2	sp	stack pointer
X3	gp	
X4	tp	} not going to use these

a	a0-a7	args and return value
t	ta-tb	temp
s	s0-s11	saved

Caller-saved regs
a0-a7, ta-tb, ra

callee-saved regs PC
s0-s11, sp program counter

Review of function Calling



foo:

addi sp,sp,-16

sd ra, 8(sp)

sd so, 8(sp)

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

;

Typical function call steps (complex)

① Allocate stack space

add: $sp, \delta p, -X$ $X \geq 16$

$\times 0\% 16 = 0$
multiple of 16

② Save RA on stack

sd ra, (sp)

Save any caller-saved regs

③ computation

④ Save any "used" caller-saved regs

⑤ Put values into args: a0, a1, ...

⑥ call func

⑦ use/store a0 (return value)

⑧ restore caller-saved regs from step (4)

⑨ computation

- (10) put return value into a0
 - (11) restore any callee-saved reg's
 - (12) restore r11
 - (13) deallocate the stack
addi \$f,\$p,X
 - (14) ret
-

