CS 315-01 Lab RISC-V Assembly 4
Functions continued

RISC-V Registers

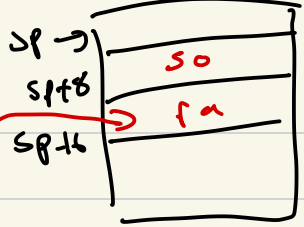
x0	zero	always zero
x1	ra	return address
x2	sp	stack pointer
x3	gp	not going to use
x4	tp	not going to use

a	a0 - a7	args and return value
t	t0 - t6	temp
s	s0 - s11	saved

Caller saved regs
a0 - a7, t0 - t6, ra

Callee saved regs
s0 - s11, sp

Review of function calling



multiple of 16

```
foo:
    addi sp,sp,-16
    sd ra,(sp)
    sd s0,8(sp)
    .
    .
    call bar
    add
    .
    .
    ld s0,8(sp)
    ld ra,(sp)
    addi sp,sp,16
    ret
```

caller

PC → call bar

RA = PC+4

PC = RA (main)

```
bar:
    add a0,a0,a1
    .
    .
    .
    ret
```

PC = RA (foo)

How to save RA?

preserve

Typical function call steps (complex)

1. Allocate stack space
   addi sp, sp, -X
   X is >= 16 and a multiple of 16
   X % 16 == 0
2. Save RA on stack
   sd ra, (sp)
2b. save any callee-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 callee-saved regs from step 4
9. computation
10. put return value into a0

(11) restore and callee saved regs

(12) restore ra

(13) de-allocate the stack

addi sp, sp, X

(17) ret