

OSI Handout Week03.a: RISC-V assembly I

1. Background I: Registers

Register	ABI Name	Description	Saver
x0	zero	Zero constant	—
x1	ra	Return address	Caller
x2	sp	Stack pointer	Callee
x3	gp	Global pointer	—
x4	tp	Thread pointer	—
x5-x7	t0-t2	Temporaries	Caller
x8	s0 / fp	Saved / frame pointer	Callee
x9	s1	Saved register	Callee
x10-x11	a0-a1	Function args / return values	Caller
x12-x17	a2-a7	Function args	Caller
x18-x27	s2-s11	Saved registers	Callee
x28-x31	t3-t6	Temporaries	Caller

[see also "RISC-V registers" in Reference page]

2. Background II: RISC-V assembly I

a) **addi** rd, rs1, immediate

rd = rs1 + immediate

b) **sw** rs2, offset(rs1)

Memory[rs1 + offset] = rs2

c) **mv** rd, rs1

rd = rs1

d) **call** rd, symbol

rd = pc+4
pc = &symbol

If rd is omitted, ra is implied.

e) **ret**

pc = ra

f) See others in "RISC-V instruction listings"
(on Reference page of OSI homepage)

3. Hellworld's C code (headers omitted)

```
1 void print_hello() {
2     printf("Hello world!\n");
3 }
4
5 int main() {
6     int a = 1;
7     print_hello();
8 }
```

4. Helloworld's asm code (simplified)

```
1 print_hello():
2 // void print_hello() {
3     addi    sp,sp,-16
4     sw      ra,12(sp)
5     sw      s0,8(sp)
6     addi    s0,sp,16
7     //     printf("Hello world!\n");
8     lui     a5,0x80208
9     mv      a0,a5
10    jal     80200534 <my_printf>
11 // }
12    nop
13    lw      ra,12(sp)
14    lw      s0,8(sp)
15    addi    sp,sp,16
16    ret
17
18
19 main():
20 // int main() {
21     addi    sp,sp,-32
22     sw      ra,28(sp)
23     sw      s0,24(sp)
24     addi    s0,sp,32
25 //     int a = 1;
26     li      a5,1
27     sw      a5,-20(s0)
28 //     print_hello();
29     jal     80200010 <print_hello>
30     li      a5,0
31 // }
32     mv      a0,a5
33     lw      ra,28(sp)
34     lw      s0,24(sp)
35     addi    sp,sp,32
36     ret
```