

```
.data
arg1: .word 5
.text
main:

addi sp,sp,16
sw ra,12(sp)
sw s0,8(sp)
addi s0,sp,1
addi a0,zero,1
sw a0,-12(s0)
lw a0, -12(s0)
call factorail(int)
mv a1,zero
sw a0,-16(s0)
add a0,zero,a1
lw s0,8(sp)
lw ra,12(sp)
addi sp,sp,16
ecall
ret
```

```
factorail(int):
addi sp,sp,-32
sw ra,28(sp)
sw s0,24(sp)
addi s0,sp,32
sw a0,-12(s0)
addi a0,zero,1
sw a0,-16(s0)
sw a0,-20(s0)
jal loop
```

```
loop:
```

```
lw a0,-20(s0)
lw a1,-12(s0)
jal loop
```

(SINGLE STEP)

The Ripes IDE interface shows the source code on the left and the disassembled assembly on the right. The source code is in C, defining a factorial function. The assembly view shows the corresponding RISC-V instructions. The instruction at address 00000044, `sw x8, 24(x2)`, is highlighted in red.

```
2 .data
3 argl: .word 5
4 .text
5 main:
6 addi sp,sp,16
7 sw ra,12(sp)
8 sw s0,8(sp)
9 addi s0,sp,1
10 addi a0,zero,1
11 sw a0,-12(s0)
12 lw a0,-12(s0)
13 call factorial(int)
14 mv a1,zero
15 sw a0,-16(s0)
16 add a0,zero,a1
17 lw s0,8(sp)
18 lw ra,12(sp)
19 addi sp,sp,16
20 ecall
21 ret
22 factorial(int):
23 addi sp,sp,-32
24 sw ra,28(sp)
25 sw s0,24(sp)
26 addi s0,sp,32
27 sw a0,-12(s0)
28 addi a0,zero,1
29 sw a0,-16(s0)
30 sw a0,-20(s0)
31 jal loop
32 loop:
33 lw a0,-20(s0)
34 lw a1,-12(s0)
35 jal loop
36
```

Disassembled assembly (addresses 00000000 to 00000068):

```
<main>:
0: 01010113      addi x2,x2,16
4: 00112623      sw x1,12(x2)
8: 00812423      sw x8,8(x2)
c: 00110413      addi x8,x2,1
10: 00100513      addi x10,x0,1
14: fea42a23      sw x10,-12(x8)
18: ff442503      lw x10,-12(x8)
1c: 00000317      auipc x6,0x0
20: 028300e7      jalr x1,x6,40
24: 00000593      addi x11,x0,0
28: fea42823      sw x10,-16(x8)
2c: 00b00533      add x10,x0,x11
30: 00812403      lw x8,8(x2)
34: 00c12083      lw x1,12(x2)
38: 01010113      addi x2,x2,16
3c: 00000073      ecall
40: 00008067      jalr x0,x1,0

<factorail(int)>:
44: fe010113      addi x2,x2,-32
48: 00112e23      sw x1,28(x2)
4c: 00812c23      sw x8,24(x2)
50: 02010413      addi x8,x2,32
54: fea42a23      sw x10,-12(x8)
58: 00100513      addi x10,x0,1
5c: fea42823      sw x10,-16(x8)
60: fea42623      sw x10,-20(x8)
64: 004000ef      jal x1,0x68,<loop>

<loop>:
68: fec42503      lw x10,-20(x8)
6c: ff442583      lw x11,-12(x8)
70: ff9fff0ef      jal x1,0x68,<loop>
```

(SINGLE STEP)

The Ripes IDE interface shows the Single Cycle RISC-V Processor architecture in the center. The processor components include PC, Instr. memory, Decode, Registers, ALU, Data memory, and Branch. The instruction `sw x8, 24(x2)` is being executed. The right panel shows the Registers and Instruction memory. The bottom panel shows the Execution info.

Registers

Name	Alias	Value
x0	zero	0x00000000
x1	ra	0x00000024
x2	sp	0x7fffffe0
x3	gp	0x10000000
x4	tp	0x00000000
x5	t0	0x00000000
x6	t1	0x0000001c
x7	t2	0x00000000

Instruction memory

BP	Addr	Stage	Instruction
<input type="checkbox"/>	0x34		lw x1,12(x2)
<input type="checkbox"/>	0x38		addi x2,x2,16
<input type="checkbox"/>	0x3c		ecall
<input type="checkbox"/>	0x40		jalr x0,x1,0
<input type="checkbox"/>	0x44		addi x2,x2,-32
<input type="checkbox"/>	0x48		sw x1,28(x2)
<input type="checkbox"/>	0x4c	*	sw x8,24(x2)

Execution info

Cycles: 11
Instrs. retired: 11
CPI: 1
IPC: 1
Clock rate: 1.03 Hz

(FIVE STAGE PROCEDURE)

The screenshot shows the Ripes IDE interface. The left sidebar contains icons for Activities, 1010 Editor, Processor, and Memory. The main window is titled 'Ripes' and has a menu bar (File, Edit, Help) and a toolbar. The 'Source code' pane on the left shows the following assembly code:

```

2: .data
3: arg1: .word 5
4: .text
5: main:
6: addi sp,sp,16
7: sw ra,12(sp)
8: sw s0,8(sp)
9: addi s0,sp,1
10: addi a0,zero,1
11: sw a0,-12(s0)
12: lw a0,-12(s0)
13: call factorial(int)
14: mv a1,zero
15: sw a0,-16(s0)
16: add a0,zero,a1
17: lw s0,8(sp)
18: lw ra,12(sp)
19: addi sp,sp,16
20: ecall
21: ret
22: factorial(int):
23: addi sp,sp,-32
24: sw ra,28(sp)
25: sw s0,24(sp)
26: addi s0,sp,32
27: sw a0,-12(s0)
28: addi a0,zero,1
29: sw a0,-16(s0)
30: sw a0,-20(s0)
31: jal loop
32: loop:
33: lw a0,-20(s0)
34: lw a1,-12(s0)
35: jal loop
36:

```

The 'Input type' is set to 'Assembly' and 'View mode' is set to 'Disassembled'. The right pane shows the disassembled assembly code:

```

00000000 <main>:
0: 01010113      addi x2,x2,16
4: 00112623      sw x1,12(x2)
8: 00812423      sw x8,8(x2)
c: 00110413      addi x8,x2,1
10: 00100513      addi x10,x0,1
14: fea42a23      sw x10,-12(x8)
18: ff442503      lw x10,-12(x8)
1c: 00000317      auipc x6,0x0
20: 028300e7      jalr x1,x6,40
24: 00000593      addi x11,x0,0
28: fea42823      sw x10,-16(x8)
2c: 00b00533      add x10,x0,x11
30: 00812403      lw x8,8(x2)
34: 00c12083      lw x1,12(x2)
38: 01010113      addi x2,x2,16
3c: 00000073      ecall
40: 00000067      jalr x0,x1,0

00000044 <factorial(int)>:
44: fe010113      addi x2,x2,-32
48: 00112e23      sw x1,28(x2)
4c: 00812c23      sw x8,24(x2)
50: 02010413      addi x8,x2,32
54: fea42a23      sw x10,-12(x8)
58: 00100513      addi x10,x0,1
5c: fea42823      sw x10,-16(x8)
60: fea42623      sw x10,-20(x8)
64: 004000ef      jal x1,0x68,<loop>

00000068 <loop>:
68: fec42503      lw x10,-20(x8)
6c: ff442583      lw x11,-12(x8)
70: ff9ff0ef      jal x1,0x68,<loop>

```

(FIVE STAGE PROCEDURE)

The screenshot shows the Ripes IDE interface with the 'Processor' view selected. The main window displays a detailed diagram of the five-stage processor architecture. The stages are labeled: PC, Instr. memory, Decode, ALU, and MEM/WB. The diagram shows the flow of data and control signals between these stages. The 'Registers' pane on the right shows the current state of the registers:

Name	Alias	Value
x4	tp	0x00000000
x5	t0	0x00000000
x6	t1	0x00000000
x7	t2	0x00000000
x8	s0	0x80000001
x9	s1	0x00000000
x10	a0	0x00000001
x11	a1	0x00000000

The 'Instruction memory' pane shows the current instruction being executed:

BP	Addr	Stage	Instruction
<input type="checkbox"/>	0x34		lw x1,12(x2)
<input type="checkbox"/>	0x38		addi x2,x2,16
<input type="checkbox"/>	0x3c		ecall
<input type="checkbox"/>	0x40		jalr x0,x1,0
<input type="checkbox"/>	0x44	IF	addi x2,x2,-32
<input type="checkbox"/>	0x48		sw x1,28(x2)
<input type="checkbox"/>	0x4c		sw x8,24(x2)

The 'Execution info' pane shows the following statistics:

- Cycles: 11
- Instrs. retired: 7
- CPI: 1.57
- IPC: 0.636
- Clock rate: 1.73 Hz

