Princeton University
COS 217: Introduction to Programming Systems
SPARC Architecture Summary

Registers			
			Global Registers:
%r0	%g0		Zero
%r1	%g1		Temporary value (destroyed)
%r2	%g2		Global variable (saved?)
%r3	%g3		Global variable (saved?)
%r4	%g4		Global variable (saved?)
%r5	%g5		Global variable (saved?)
%r6	%g6		Global variable (saved?)
%r7	%g7		Global variable (saved?)
			Output Registers:
%r8	%o0		Function actual parameter or return value from caller (destroyed)
%r9	%o1		Function actual parameter (destroyed)
%r10	%o2		Function actual parameter (destroyed)
%r11	%o3		Function actual parameter (destroyed)
%r12	%o4		Function actual parameter (destroyed)
%r13	%o5		Function actual parameter (destroyed)
%r14	%o6	%sp	Stack pointer (saved)
%r15	%o7		Addr of call instruction / temporary value (destroyed)
			Local Registers:
%r16	%l0		Local variable (saved) (used by interrupt handler)
%r17	%l1		Local variable (saved) (used by interrupt handler)
%r18	%l2		Local variable (saved)
%r19	%l3		Local variable (saved)
%r20	%l4		Local variable (saved)
%r21	%l5		Local variable (saved)
%r22	%l6		Local variable (saved)
%r23	%l7		Local variable (saved)
			Input Registers:
%r24	%i0		Function formal parameter or return value to caller (saved)
%r25	%i1		Function formal parameter (saved)
%r26	%i2		Function formal parameter (saved)
%r27	%i3		Function formal parameter (saved)
%r28	%i4		Function formal parameter (saved)
%r29	%i5		Function formal parameter (saved)
%r30	%i6	%fp	Frame pointer (saved)
%r31	%i7		Return addr - 8 (saved)
			Floating-Point Registers:
%f0			Floating-point value (saved)
...			...
%f31			Floating-point value (saved)

Stack Structure

%sp	register window	92 bytes	Current Function’s Stack Frame
%sp + 4	register window (cont.)		
%sp + 8	register window (cont.)		
%sp + 12	register window (cont.)		
%sp + 16	register window (cont.)		
%sp + 20	register window (cont.)		
%sp + 24	register window (cont.)		
%sp + 28	register window (cont.)		
%sp + 32	register window (cont.)		
%sp + 36	register window (cont.)		
%sp + 40	register window (cont.)		
%sp + 44	register window (cont.)		
%sp + 48	register window (cont.)		
%sp + 52	register window (cont.)		
%sp + 56	register window (cont.)		
%sp + 60	register window (cont.)		
%sp + 64	**structure pointer**		
%sp + 68	actual parameter 1 (often unused)		
%sp + 72	actual parameter 2 (often unused)		
%sp + 76	actual parameter 3 (often unused)		
%sp + 80	actual parameter 4 (often unused)		
%sp + 84	actual parameter 5 (often unused)		
%sp + 88	actual parameter 6 (often unused)		
%sp + 92	**additional actual parameters and saved floating-point registers**	??? bytes	
...	...		
%fp – 4	**local variables**		
%fp	register window	92 bytes	Calling Function’s Stack Frame
%fp + 4	register window (cont.)		
%fp + 8	register window (cont.)		
%fp + 12	register window (cont.)		
%fp + 16	register window (cont.)		
%fp + 20	register window (cont.)		
%fp + 24	register window (cont.)		
%fp + 28	register window (cont.)		
%fp + 32	register window (cont.)		
%fp + 36	register window (cont.)		
%fp + 40	register window (cont.)		
%fp + 44	register window (cont.)		
%fp + 48	register window (cont.)		
%fp + 52	register window (cont.)		
%fp + 56	register window (cont.)		
%fp + 60	register window (cont.)		
%fp + 64	**structure pointer**		
%fp + 68	formal parameter 1 (often unused)		
%fp + 72	formal parameter 2 (often unused)		
%fp + 76	formal parameter 3 (often unused)		
%fp + 80	formal parameter 4 (often unused)		
%fp + 84	formal parameter 5 (often unused)		
%fp + 88	formal parameter 6 (often unused)		
%fp + 92	**additional formal parameters and saved floating-point registers**		
...	...		

Instruction Format 1 (call)

31-30	Op code (01)
29-0	Displacement

Instruction Format 2 (sethi)

31-30	Op code (00)
29-25	Destination register
24-22	Op code continued (100)
21-0	Immediate operand

Instruction Format 2 (branches and nop)

31-30	Op code (00)
29	Annul bit
28-25	Condition
24-22	Op code continued
21-0	Displacement

Instruction Format 3
(all other instructions, two source registers)

31-30	Op code (10 or 11)
29-25	Destination register
24-19	Op code continued
18-14	Source register 1
13	0
12-5	(unused)
4-0	Source register 2

Instruction Format 3
(all other instructions, source register and immediate operand)

31-30	Op code (10 or 11)
29-25	Destination register
24-19	Op code continued
18-14	Source register 1
13	1
12-0	Source immediate constant