

# BLITZ Instruction Set

Op Code (Decimal)	Op Code (Hex)	Instruction	Format	Privileged Instruction	Modifies Cond. Codes
96	60	add	$R_a, R_b, R_c$	D	X
128	80	add	$R_a, data16, R_c$	E	X
97	61	sub	$R_a, R_b, R_c$	D	X
129	81	sub	$R_a, data16, R_c$	E	X
98	62	mul	$R_a, R_b, R_c$	D	X
130	82	mul	$R_a, data16, R_c$	E	X
99	63	div	$R_a, R_b, R_c$	D	X
131	83	div	$R_a, data16, R_c$	E	X
100	64	sll	$R_a, R_b, R_c$	D	X
132	84	sll	$R_a, data16, R_c$	E	X
101	65	srl	$R_a, R_b, R_c$	D	X
133	85	srl	$R_a, data16, R_c$	E	X
102	66	sra	$R_a, R_b, R_c$	D	X
134	86	sra	$R_a, data16, R_c$	E	X
103	67	or	$R_a, R_b, R_c$	D	X
135	87	or	$R_a, data16, R_c$	E	X
104	68	and	$R_a, R_b, R_c$	D	X
136	88	and	$R_a, data16, R_c$	E	X
105	69	andn	$R_a, R_b, R_c$	D	X
137	89	andn	$R_a, data16, R_c$	E	X
106	6A	xor	$R_a, R_b, R_c$	D	X
138	8A	xor	$R_a, data16, R_c$	E	X
115	73	rem	$R_a, R_b, R_c$	D	X
149	95	rem	$R_a, data16, R_c$	E	X
107	6B	load	$[R_a+R_b], R_c$	D	
139	8B	load	$[R_a+data16], R_c$	E	
108	6C	loadb	$[R_a+R_b], R_c$	D	
140	8C	loadb	$[R_a+data16], R_c$	E	
109	6D	loadv	$[R_a+R_b], R_c$	D	X
141	8D	loadv	$[R_a+data16], R_c$	E	X
110	6E	loadbv	$[R_a+R_b], R_c$	D	X
142	8E	loadbv	$[R_a+data16], R_c$	E	X
111	6F	store	$R_c, [R_a+R_b]$	D	
143	8F	store	$R_c, [R_a+data16]$	E	
112	70	storeb	$R_c, [R_a+R_b]$	D	
144	90	storeb	$R_c, [R_a+data16]$	E	
113	71	storev	$R_c, [R_a+R_b]$	D	X
145	91	storev	$R_c, [R_a+data16]$	E	X
114	72	storebv	$R_c, [R_a+R_b]$	D	X
146	92	storebv	$R_c, [R_a+data16]$	E	X

## BLITZ Instruction Set

Op Code (Decimal)	Op Code (Hex)	Instruction	Format	Privileged Instruction	Modifies Cond. Codes
64	40	call	$R_a+R_c$	C	
160	A0	call	data24	F	
65	41	jmp	$R_a+R_c$	C	
161	A1	jmp	data24	F	
66	42	be	$R_a+R_c$	C	
162	A2	be	data24	F	
67	43	bne	$R_a+R_c$	C	
163	A3	bne	data24	F	
68	44	bl	$R_a+R_c$	C	
164	A4	bl	data24	F	
69	45	ble	$R_a+R_c$	C	
165	A5	ble	data24	F	
70	46	bg	$R_a+R_c$	C	
166	A6	bg	data24	F	
71	47	bge	$R_a+R_c$	C	
167	A7	bge	data24	F	
72	48				
168	A8				
73	49				
169	A9				
74	4A	bvs	$R_a+R_c$	C	
170	AA	bvs	data24	F	
75	4B	bvc	$R_a+R_c$	C	
171	AB	bvc	data24	F	
76	4C	bns	$R_a+R_c$	C	
172	AC	bns	data24	F	
77	4D	bnc	$R_a+R_c$	C	
173	AD	bnc	data24	F	
78	4E	bss	$R_a+R_c$	C	
174	AE	bss	data24	F	
79	4F	bsc	$R_a+R_c$	C	
175	AF	bsc	data24	F	
80	50	bis	$R_a+R_c$	C	
176	B0	bis	data24	F	
81	51	bic	$R_a+R_c$	C	
177	B1	bic	data24	F	
82	52	bps	$R_a+R_c$	C	
178	B2	bps	data24	F	
83	53	bpc	$R_a+R_c$	C	
179	B3	bpc	data24	F	

# BLITZ Instruction Set

Op Code (Decimal)	Op Code (Hex)	Instruction	Format	Privileged Instruction	Modifies Cond. Codes
84	54	push	$R_c, [--R_a]$	C	
85	55	pop	$[R_a++] , R_c$	C	
192	C0	sethi	$data16, R_c$	G	
193	C1	setlo	$data16, R_c$	G	
194	C2	ldaddr	$data16, R_c$	G	
195	C3	syscall	$R_c + data16$	G	
0	00	nop		A	
1	01	wait		A	X
2	02	debug		A	
3	03	cleari		A	X
4	04	seti		A	X
5	05	clearp		A	X
6	06	setp		A	X
7	07	clears		A	X
8	08	reti		A	X
9	09	ret		A	
10	0A	debug2		A	
88	58	tset	$[R_a], R_c$	C	
86	56	readu	$R_c, R_a$	C	X
147	93	readu	$R_c, [R_a + data16]$	E	X
87	57	writu	$R_a, R_c$	C	X
148	94	writu	$[R_a + data16], R_c$	E	X
32	20	ldptbr	$R_c$	B	X
33	21	ldptlr	$R_c$	B	X
89	59	ftoi	$F_a, R_c$	C	
90	5A	itof	$R_a, F_c$	C	
116	74	fadd	$F_a, F_b, F_c$	D	
117	75	fsub	$F_a, F_b, F_c$	D	
118	76	fmul	$F_a, F_b, F_c$	D	
119	77	fdiv	$F_a, F_b, F_c$	D	
91	5B	fcmp	$F_a, F_c$	C	X
92	5C	fsqrt	$F_a, F_c$	C	
93	5D	fneg	$F_a, F_c$	C	
94	5E	fabs	$F_a, F_c$	C	
120	78	fload	$[R_a + R_b], F_c$	D	
150	96	fload	$[R_a + data16], F_c$	E	
121	79	fstore	$F_c, [R_a + R_b]$	D	
151	97	fstore	$F_c, [R_a + data16]$	E	