opcode	name	arg0	arg1	selector array	description
100	goto_op	optable slot of destination			unconditional jump
101	jump_if_zero_op	optable slot of destination			jump if top of sip control stack is zero
102	stop_op				immediately abort sial program. Useful during debugging but should not be used in production code-
103	call_op	optable slot of procedure			call a sial procedure, first push slot of next instruction on control stack
104	return_op				return from procedure. optable slot of caller is on the control stack
105	execute_op	super instruction table slot	number of arguments		execute indicated user provided super instruction. The argumnents are on the block selector stack
106	do_op	optable slot of enddo	number of indices = 1	first element is slot of loop index variable	serial do loop
107	enddo_op		number of indices	first element is slot of loop index variable	marks end of serial do loop
108	dosubindex_op	optable slot of enddo-subindex	parent index	first element is slot of loop index variable	serial loop over subindex
109	enddosubindex_op		parent	first element is slot of loop index variable	marks end of loop over subindex variable
110	exit_op				exit current do loop
111	where_op				
112	pardo_op	optable slot of enddo	number of indices	indices indicated in loop	beginning of pardo loop
113	endpardo_op		number of indices	indices indicated in loop	end of pardo loop
114	begin_pardo_section_op				start of a pardo section
115	end_pardo_section_-op				end of a pardo section
116	sip_barrier_op				
117	broadcast_static_op	index of static array			broadcast static array from rank whose value is on top of control stack

118	push_block_selector_op	rank	array table slot	selector slots	push the block selector onto the sip block selector stack. If the rank is 0, this is either a scalar or a static or contig array given without a selector
119	allocate_op	rank	array_table_slot	block selector indices(may contain wild cards)	allocate block(s) of local array.
120	deallocate_op	rank	array_table_slot	block selector indices	deallocate block(s) of local array
121	allocate_contiguous_op	rank	array_table_slot		allocates memory for a region of a contiguous local array. The boundaries are obtained from the control_stack where they have been pushed in the order they appear in the program, e.g. lower[0], upper[0]..lower[rank-1], upper[rank-1]
122	deallocate_contiguous_op	rank	array_table_slot		deallocates memory for a region of a contiguous local array. The boundaries are obtained from the control_stack where they have been pushed in the order they appear in the program, e.g. lower[0], upper[0]..lower[rank-1], upper[rank-1]
123	get_op	array table slot of desired block		selector slots	get block selector from selector stack and send get request to appropriate server (args are redundant)
124	put_accumulate_op	array table slot of right hand side	array table slot of left hand side		get right and left side blocks (left pushed firsts) from selector stack and send rhs block to appropriate server to accumulate into its copy of lhs block
125	put_replace_op	array table slot of right hand side	array table slot of left hand side		get right and left side blocks (left pushed firsts) from selector stack and send rhs block to appropriate server to replace its copy of lhs block
126	create_op	array table slot			create distributed array. In aces4, blocks are created lazily
127	delete_op	array table slot			delete distributed array
128	int_load_value_op	IntTable slot			loads current value of indicated int onto sip control stack

129	int_load_literal_op	value			loads value encoded in arg0 of instruction onto sip control stack
130	int_store_op	IntTable slot	opcode of operator, or int_store_op if plain assignment		removes value from top of sip control stack, performs indicated op with value of given int, and stores in given int
131	index_load_value_op	IndexTable slot			load current value of index and stores it on the control stack
132	int_add_op				removes the top two values from the control stack, adds them together, and pushes the result on the control stack.
133	int_subtract_op				removes the top two values from the control stack, subtracts the first popped from the second, and pushes the result onto the control stack
134	int_multiply_op				removes the top two values from the control stack, multiplies them together, and pushes the result on the control stack.
135	int_divide_op				removes the top two values from the control stack, divides the second popped by the first, and pushes the result onto the control stack
136	int_equal_op				==, args are popped from sip control stack, result is placed on control stack
137	int_nequal_op				!=, args are popped from sip control stack, result is placed on control stack
138	int_ge_op				¿=, args are popped from sip control stack, result is placed on control stack
139	int_le_op				¡=, args are popped from sip control stack, result is placed on control stack
140	int_gt_op				¿, args are popped from sip control stack, result is placed on control stack
141	int_lt_op				¡, args are popped from sip control stack, result is placed on control stack
142	int_neg_op				unary negation, arg is popped from sip control stack, result is placed on control stack

143	cast_to_int_op				removes scalar value from expression stack, converts to int, and puts it on the control stack
144	scalar_load_value_op	array table slot			loads value of scalar in given slot onto sip expression stack
145	scalar_store_op	array table slot of scalar	opcode of operator or scalar_store_op if plain assignment		removes value from top of sip expression stack, performs indicated op with value of given scalar, and stores in given scalar
146	scalar_add_op				removes top two elements from expression stack, adds together, pushes result on expression stack
147	scalar_subtract_op				removes top two elements from expression stack, subtracts top from next-to-top (i.e. args pushed left to right), pushes result on expressio stack
148	scalar_multiply_op				removes top two elements from expression stack, multiplies together, pushes result on expression stack
149	scalar_divide_op				removes top two elements from expression stack, divides next-to-top by top (i.e. args pushed left to right), pushes result on expression stack
150	scalar_exp_op				removes top two elements s,t from expression stack, computes s**t (c++ pow(s,t)), args pushed from left to right, pushes result onto expression stack
151	scalar_eq_op				==, args are popped from sip expression stack, result is placed on control stack
152	scalar_ne_op				!=, args are popped from sip expression stack, result is placed on control stack
153	scalar_ge_op				¿=, args are popped from sip expression stack, result is placed on control stack
154	scalar_le_op				¡=, args are popped from sip expression stack, result is placed on control stack
155	scalar_gt_op				¿, args are popped from sip expression stack, result is placed on control stack

156	scalar_lt_op				¡, args are popped from sip expression stack, result is placed on control stack
157	scalar_neg_op				unary negation, arg is popped from sip expression stack, result is placed on expression stack
158	scalar_sqrt_op				computes square root of value on top of sip expression stack, leaves result on top of expression stack
159	cast_to_scalar_op				removes top element from control stack, converts to double and leaves on top of expression stack
160	collective_sum_op	array table slot of lhs scalar			allreduce of rhs value which is on expression stack into lhs scalar. This operation synchronizes the workers
161	assert_same_op	array table slot of scalan			checks that value of scalar is within epsilon on all workers, and resets all to master's value
162	tensor_op	lhs rank	lhs array table slot	lhs selector	outer product, uses same routine as tensor contraction
163	block_copy_op	lhs rank	lhs array table slot	lhs selector	copies block from top of block selector stack to block in instruction. If one array is larger than the other, the extra indices are simple
164	block_permute_op			permutation	permute the block on the right side using the given permutation . RHS and LHS block selectors have been pushed onto block selector stack, first rhs then lhs
165	block_fill_op	lhs rank	lhs array table slot	lhs indices	gets value from expression stack and block from instruction. Sets each element of the block to the given value
166	block_scale_op	lhs rank	lhs array table slot	lhs indices	gets value from expression stack and block from instruction. Multiplies all of the elements of the block by the value

167	block_scale_assign_op	lhs rank	lhs array table slot	lhs indices	gets value from expression stack and block from block selector stack. Destinatin is in instruction. Multiplies all of the elements of the block by the value and leaves the result in the lhs block
168	block_scale_accumulate_op	lhs rank	lhs array table slot	lhs indices	gets value from expression stack and block from block selector stack. Destinatin is in instruction. Multiplies all of the elements of the block by the value add to the lhs block
169	block_accumulate_scalar_op	lhs rank	lhs slot	lhs selector indices	gets scalar value from expression stack and from instruction. Adds the scalar to each value in the block
170	block_add_op	lhs rank	lhs array slot	lhs selector	adds two blocks together element-wise and puts the result in the lhs array
171	block_subtract_op	lhs rank	lhs array slot	lhs selector	subtracts two blocks elementwise and puts the result in the lhs array
172	block_contract_op	lhs rank	lhs array slot	lhs selector	contracts two blocks and puts the result in the lhs array
173	block_contract_accumulate_op	lhs rank	lhs array slot	lhs selector	contracts two blocks and accumulates the result in the lhs array
174	block_contract_to_scalar_op				contracts two blocks where the result of contraction is a scalar. Leaves the result on the sip expression stack
175	block_load_scalar_op				all indices of block on top of selector stack are simple, "block" is a single scalar value, load it onto the sip expression stack
176	slice_op	lhs rank	lhs array table slot	lhs selector indices	copies subblock on lhs from rhs superblock
177	insert_op	lhs rank	lhs array table slot	lhs selector indices	inserts sublock on rhs into superblock on lhs
178	string_load_literal_op	slot in string literal table			loads slot in string table onto control stack.
179	print_string_op	append NL if 1			print the string whose slot in string table is on the sip control stack
180	println_op				print NL)

181	print_index_op	append NL if 1	index table slot		print current value of given index; the value is on the sip control stack
182	print_scalar_op	append NL if 1	array table slot, or unused if literal		print scalar whose value is on the sip epression stack
183	print_int_op	append NL if 1	array table slot or unused if literal		print int; value is on the sip control stack
184	print_block_op	append NL if 1			print the block whose selector is on the selector stack
185	gpu_on_op				
186	gpu_off_op				
187	gpu_allocate_op				
188	gpu_free_op				
189	gpu_put_op				
190	gpu_get_op				
191	gpu_get_int_op				
192	gpu_put_int_op				
193	set_persistent_op	string table slot	array table slot		Marks array with the given label as persistent. At the end of the current SIAL program, it will be saved for restoration in a future program in same run
194	restore_persistent_op	string_table_slot	array_table_slot		restores contents of persistent array with given label as indicated array
195	idup_op				duplicates the value on top of the control (integer) stack
196	iswap_op				swaps the top two values on the control (integer) stack
197	sswap_op				swaps the top two values on the expression (scalar) stack
198	invalid_op				