

TABLE IX: The program semantics represented by PHP opcodes.

#	Type	OPCODE	RES	OP1	OP2	Ext	Brief
1	J	ZEND_JMP		ADR			Unconditional jump
2	J	ZEND_JMPZ			ADR		Conditional jump
3	J	ZEND_JMPNZ			ADR		Conditional jump
4	J	ZEND_JMPZ_EX			ADR		Conditional jump
5	J	ZEND_JMPNZ_EX			ADR		Conditional jump
6	J	ZEND_JMP_SET			ADR		Conditional jump
7	J	ZEND_COALESCE			ADR		Conditional jump
8	J	ZEND_ASSERT_CHECK			ADR		Conditional jump
9	J	ZEND_JMP_NULL			ADR		Conditional jump
10	J	ZEND_SWITCH_STRING				ADR	Switch,conditional jump
11	J	ZEND_EXIT		Variable			Exit and output
12	E	ZEND_FAST_CALL		ADR			
13	E	ZEND_FAST_RET					ADR is the next of ZEND_FAST_CALL
14	E	ZEND_THROW					Find ZEND_CATCH
15	E	ZEND_CATCH		Exception class name			New Exception
16	JA	ZEND_FE_RESET_R	To	Array variable	ADR		Foreach,conditional jump
17	JA	ZEND_FE_RESET_RW	To	Array variable	ADR		Foreach,conditional jump
18	JA	ZEND_FE_FETCH_R	Key	Array variable	Value	ADR	Foreach,conditional jump
19	JA	ZEND_FE_FETCH_RW		Array variable	To	ADR	Foreach,conditional jump
20	JV	ZEND_CASE	Boolean	Value1	Value2		Whether value1 equals value2
21	F	ZEND_INCLUDE_OR_EVAL	To	Value	Type		Include or dynamic PHP code execution
22	F	ZEND_INIT_FCALL			Function name		Initialization
23	F	ZEND_INIT_FCALL_BY_NAME			Function name		Initialization, funtion not defined in this file
24	F	ZEND_INIT_NS_FCALL_BY_NAME			Function name		Initialization, namespaces
25	FO	ZEND_INIT_METHOD_CALL		Object variable	Method name		Initialization, method
26	FO	ZEND_INIT_STATIC_METHOD_CALL		Class name	Method name		Initialization, static method
27	F	ZEND_INIT_USER_CALL		call_user_func(_array)	function name		Initialization, call_user_func
28	F	ZEND_INIT_DYNAMIC_CALL			function name		Initialization, dynamic
29	F	ZEND_SEND_VAL		Variable			Parameter Passing, constant
30	F	ZEND_SEND_FUNC_ARG		Variable			Parameter Passing, dynamic call
31	F	ZEND_SEND_VAR		Variable			Parameter Passing, variable
32	F	ZEND_SEND_VAL_EX		Variable			Parameter Passing, external
33	F	ZEND_SEND_VAR_EX		Variable			Parameter Passing, external
34	F	ZEND_SEND_REF		Variable			Parameter Passing, ref
35	F	ZEND_SEND_USER		Variable			Parameter Passing, USER_CALL
36	FA	ZEND_SEND_ARRAY		Variable			Parameter Passing, array
37	FA	ZEND_SEND_UNPACK		Variable			Parameter Passing, VARIADIC
38	F	ZEND_DO_FCALL	To				Function Execution
39	FO	ZEND_VERIFY_RETURN_TYPE		Variable			Magic methods, __toString
40	F	ZEND_RECV_INIT	To		Init value		Parameter Receiving
41	F	ZEND_RECV	To				Parameter Receiving
42	FA	ZEND_RECV_VARIADIC	To				variadic functions
43	F	ZEND_FUNC_GET_ARGS	To				func_get_args()
44	F	ZEND_FUNC_NUM_ARGS	To				func_num_args()
45	F	ZEND_RETURN_BY_REF		Variable			Function Return, ref
46	F	ZEND_RETURN		Variable			Function Return
47	F	ZEND_DECLARE_LAMBDA_FUNCTION	To	Fuction name			Closure function
48	F	ZEND_DECLARE_FUNCTION		Fuction name			Nested function
49	V	ZEND_ADD	To	Value1	Value2		OP1 + OP2 to RET
50	V	ZEND_SUB	To	Value1	Value2		OP1 - OP2 to RET
51	V	ZEND_MUL	To	Value1	Value2		OP1 * OP2 to RET
52	V	ZEND_DIV	To	Value1	Value2		OP1 / OP2 to RET
53	V	ZEND_MOD	To	Value1	Value2		OP1 % OP2 to RET
54	V	ZEND_SL	To	Value1	Value2		OP1 << OP2 to RET
55	V	ZEND_SR	To	Value1	Value2		OP1 >> OP2 to RET
56	V	ZEND_BW_OR	To	Value1	Value2		OP1 — OP2 to RET
57	V	ZEND_BW_AND	To	Value1	Value2		OP1 & OP2 to RET
58	V	ZEND_BW_XOR	To	Value1	Value2		OP1 ^ OP2 to RET
59	V	ZEND_POW	To	Value1	Value2		OP1 ** OP2 to RET
60	V	ZEND_BOOL_XOR	To	Value1	Value2		OP1 xor OP2 to RET
61	V	ZEND_FETCH_FUNC_ARG	To	From			Fetch dynamic func arg
62	V	ZEND_FETCH_RW	To	From			Fetch Value
63	V	ZEND_FETCH_W	To	From			Fetch Value
64	V	ZEND_FETCH_R	To	From			Fetch Value
65	V	ZEND_IS_SMALLER_OR_EQUAL	Boolean	Value1	Value2		Comparison Operators
66	V	ZEND_IS_SMALLER	Boolean	Value1	Value2		Comparison Operators
67	V	ZEND_IS_EQUAL	Boolean	Value1	Value2		Comparison Operators
68	V	ZEND_IS_NOT_EQUAL	Boolean	Value1	Value2		Comparison Operators
69	V	ZEND_ISSET_ISEMPY_DIM_OBJ	Boolean	Array variable	Key		Isset/isempty
70	V	ZEND_BOOL	To	Value1			Boolean value1
71	V	ZEND_BOOL_NOT	To	Value1			Boolean not value1
72	V	ZEND_PRE_INC		Value1			++Value1
73	V	ZEND_PRE_DEC		Value1			--Value1
74	V	ZEND_POST_INC		Value1			Value1++
75	V	ZEND_POST_DEC		Value1			Value1--
76	V	ZEND_ASSIGN_OP		Variable1	Variable2		Select operator based on fetch_flag
77	V	ZEND_ROPE_INIT	To	Variable1	Variable2		Concat Var1 and Var2 to RES
78	V	ZEND_ROPE_ADD	To	Variable1	Variable2		Concat Var1 and Var2 to RES
79	V	ZEND_ROPE_END	To	Variable1	Variable2		Concat Var1 and Var2 to RES

Continued on next page

Table IX continued from previous page

#	Type	OPCODE	RES	OP1	OP2	Ext	Brief
80	V	ZEND_CONCAT	To	Variable1	Variable2		Concat Var1 and Var2 to RES
81	V	ZEND_FAST_CONCAT	To	Variable1	Variable2		Concat Var1 and Var2 to RES
82	V	ZEND_DECLARE_CONST		Const name	Const value		Declare const
83	V	ZEND_FETCH_CONSTANT	To		Const name		Fetch const
84	V	ZEND_ECHO		Variable			Output
85	A	ZEND_INIT_ARRAY	To	Key	Value		Array Init
86	A	ZEND_ADD_ARRAY_ELEMENT	To	Key	Value		Array element added
87	A	ZEND_ASSIGN_DIM_OP		Array variable	Key		Select operator based on fetch_flag
88	A	ZEND_ARRAY_KEY_EXISTS	Boolean	Key	Array variable		Is key exists
89	A	ZEND_FETCH_DIM_IS	Boolean	Array variable	Key		Is key exists
90	AO	ZEND_OP_DATA		Variable			Determined by previous opcode
91	O	ZEND_NEW		Class name			New object
92	O	ZEND_DECLARE_ANON_CLASS		Class name			Anonymous class
93	O	ZEND_QM_ASSIGN	To	From			Question mark assign
94	O	ZEND_CLONE	To	From			Clone
95	O	ZEND_ASSIGN_STATIC_PROP		Properties name	Class name		Value from ZEND_OP_DATA
96	O	ZEND_ASSIGN_STATIC_PROP_REF		Properties name	Class name		Value(ref) from ZEND_OP_DATA
97	O	ZEND_FETCH_STATIC_PROP_R	To	Properties name	Class name		Fetch static Properties
98	O	ZEND_ASSIGN_OBJ_REF		Object variable	Properties name		Value(ref) from ZEND_OP_DATA
99	O	ZEND_INSTANCEOF	Boolean	Object variable	Class name		Instanceof Keyword
100	O	ZEND_GET_CALLED_CLASS	To				get_called_class()
101	O	ZEND_FETCH_CLASS	To		Object variable		Fetch class name
102	O	ZEND_FETCH_CLASS_CONSTANT	To	Class name	Constant		Fetch class constants
103	O	ZEND_FETCH_OBJ_R	To	Object variable	Properties name		Fetch object properties
104	O	ZEND_ISSET_ISEMPY_PROP_OBJ	Boolean	Object variable	Properties name		Isset/isempty
105	O	ZEND_UNSET_OBJ		Object variable	Properties name		Unset
106	O	ZEND_FETCH_OBJ_W	To	Object variable	Properties name		Fetch object properties
107	O	ZEND_ASSIGN_OBJ		Object variable	Properties name		Value from ZEND_OP_DATA
108	VAO	ZEND_ASSIGN		To	From		ASSIGN
109	VAO	ZEND_FETCH_DIM_W	To	Array variable	element		There are various value types in array
110	VAO	ZEND_ASSIGN_DIM		Array variable	element		Value from ZEND_OP_DATA
111	VAO	ZEND_FETCH_DIM_FUNC_ARG	To	Array variable	element		There are various value types in array
112	VAO	ZEND_FETCH_DIM_R	To	Array variable	element		There are various value types in array
113	VAO	ZEND_ISSET_ISEMPY_CV	Boolean	Variable			isset()
114	VAO	ZEND_MAKE_REF	Variable1	Variable2			Variable1 = &Variable2
115	VAO	ZEND_ASSIGN_REF		Variable1	Variable2		Variable1 = &Variable2
116	VAO	ZEND_YIELD		Variable			Yield
117	VAO	ZEND_GENERATOR_RETURN					Yield return
118	VAO	ZEND_BIND_LEXICAL		Variable	Compiled vars		Closure function parameter bind
119	VAO	ZEND_BIND_STATIC		Compiled vars			Closure function parameter bind
120	VAO	ZEND_BIND_GLOBAL		Compiled vars	GlobalVar name		Global variable declaration
121	VAO	ZEND_CAST	To	Variable1			Forced type convert based on fetch_flag
122	G	ZEND_FETCH_IS	To	Superglobals name			Fetch superglobals

In *Type*, **J**, **E**, **F**, **V**, **A**, **O**, and **G** represent jump instructions, exception handling, function calls, scalars, arrays, objects, and global variables, respectively. **G** specifically indicates that the opcode is used only in the context of global variables, while the variables involved in **V**, **A**, and **O** can also have global attributes depending on the context; **To** represent that the calculation result is saved here; The full name of *Ext* is **Extended_Value**, which is mentioned in the paper.