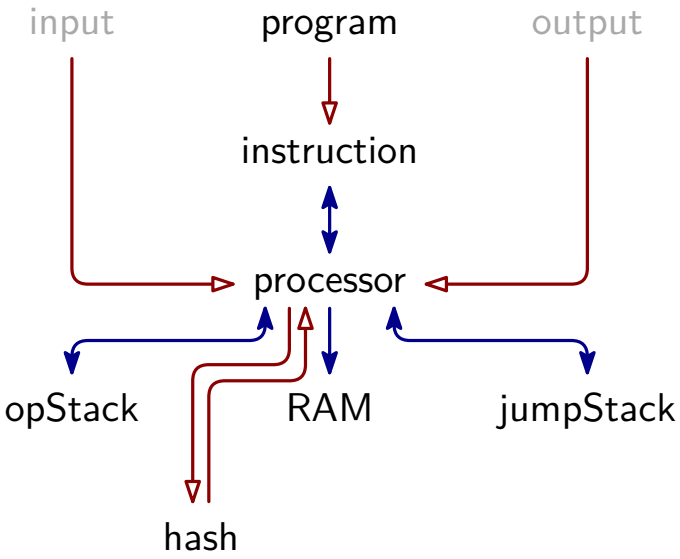
02	⊖	pop	
01	⊕	push + a	
04	⊕	divine	
05	⊕	dup + i	
09		swap + i	
08	○	nop	
17	○	if_then_call + d	
13	○	call + d	
12	○	return	
16	○	recurse	
10	⊖	assert	
00	○	halt	
20	⊙	read_mem	
24	○	write_mem	
28		hash	
32		divine_sibling	st12 % 2 = 0 ⇒ left node
36	○	assert_vector	
14	⁝	add	
18	⁝	mul	
40	⊙	invert	
44		split	hi → st0’
22	⁝	eq	
72		lsb	
56		xxadd	
60		xxmul	
64		xinv	
38	●	xbmul	st0 · (st1, st2, st3)
68	⊕	read_io	
42	⊖	write_io	



$p = 18446744069414584321$

i	$1/i$	$-1/i$
2	092...161	922...160
3	122...881	614...440
4	138...241	461...080
5	147...457	368...864
6	153...601	307...720

	base	ext	$\sum$
Program	3	1	4
Instruction	4	2	6
Processor	38	8	46
OpStack	4	1	5
RAM	4	1	5
JumpStack	5	1	6
Hash	49	2	51
$\sum$	107	16	123

Table	Base Columns																							
Program	Address			Instruction		IsPadding																		
Instruction	Address			CI	NIA	IsPadding																		
Processor	CLK	IsPadding	IP	CI	NIA	IB0	...	IB6	JSP	JSO	JSD	ST0	...	ST15	OSP	OSV	HV0	...	HV3	RAMV				
OpStack	CLK						IB1 (≙ shrink stack)								OSP	OSV								
RAM	CLK												RAMP (≙ ST1)							RAMV	IORD		
JumpStack	CLK			CI					JSP	JSO	JSD													
Hash	RoundNumber											ST0	...	ST15	CONSTANT0A			...	CONSTANT15B					

#clk	instruction
2	neg
4	sub
68	is_u32
139	split_assert
146	lte
148	lt
295	and
301	xor
195	reverse
164	div