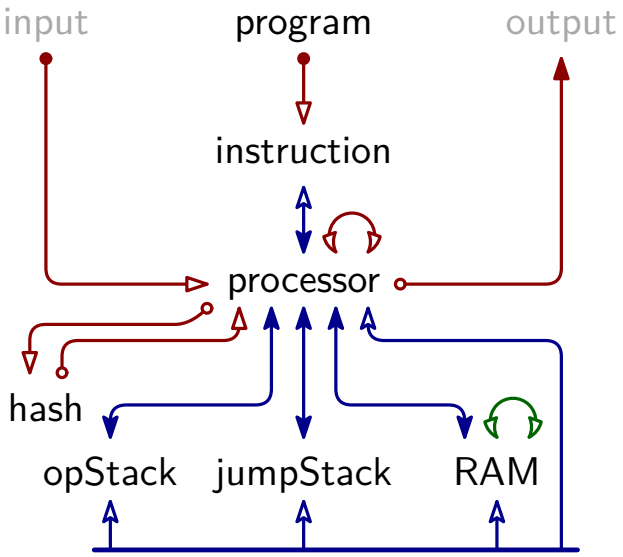
02	⊖	pop	
01	⊕	push + a	
04	⊕	divine	
05	⊕	dup + i	
09	○[16]	swap + i	
08	○	nop	
06	⊖	skiz	
13	○	call + d	
12	○	return	
16	○	recurse	
10	⊖	assert	
00	○	halt	
24	○[1]	read_mem	
28	○	write_mem	
32	○[10]	hash	
36	○[11]	divine_sibling	st12 % 2 = 0 ⇒ left node
40	○	assert_vector	
14	⊖[1]	add	
18	⊖[1]	mul	
44	○[1]	invert	
48	⊕[2]	split	hi → st0'
22	⊖[1]	eq	
52	⊕[2]	lsb	
56	○[3]	xxadd	
60	○[3]	xxmul	
64	○[3]	xinvert	
26	⊖[3]	xbmul	st0 · (st1, st2, st3)
68	⊕	read_io	
30	⊖	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	42	11	53
OpStack	5	2	7
RAM	7	6	13
JumpStack	6	2	8
Hash	49	2	51
$\sum$	116	26	142

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	RAMP	RAMV						
OpStack	CLK	clk_di					IB1 (≙ shrink stack)								OSP	OSV											
RAM	CLK	clk_di			bcpc0	bcpc1														RAMP	RAMV	IORD					
JumpStack	CLK	clk_di		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