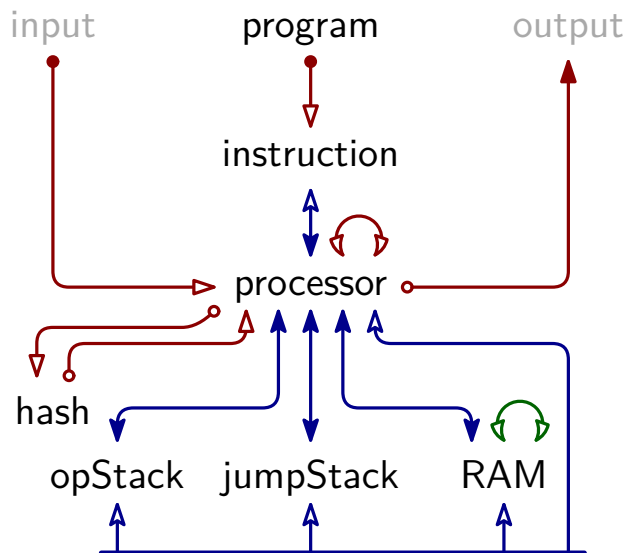


```

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
20 ○1 read_mem
24 ○ write_mem
28 ○10 hash
32 ○11 divine_sibling  st12 % 2 = 0 ⇒ left node
36 ○ assert_vector
14 ⊖1 add
18 ⊖1 mul
40 ○1 invert
44 ⊕2 split          hi → st0'
22 ⊖1 eq
48 ⊕2 lsb
52 ○3 xxadd
56 ○3 xxmulo
60 ○3 xinverso
26 ⊖3 xbmulo          st0 · (st1, st2, st3)
64 ⊕ 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	Σ
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
Σ	116	26	142

Table	Base Columns																							
Program	Address			Instruction		IsPadding																		
Instruction	Address			CI	NIA	IsPadding																		
Processor	CLK	IsPadding	IP	CI	NIA	IB0	...	IB6	JSP	JS0	JSD	ST0	...	ST15	OSP	OSV	HV0	...	HV3	RAMP	RAMV			
OpStack	CLK	clk.di			IB1 ($\hat{=}$ shrink stack)										OSP	OSV								
RAM	CLK	clk.di			bcpc0		bcpc1												RAMP	RAMV	IORD			
JumpStack	CLK	clk.di			CI				JSP				JS0	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