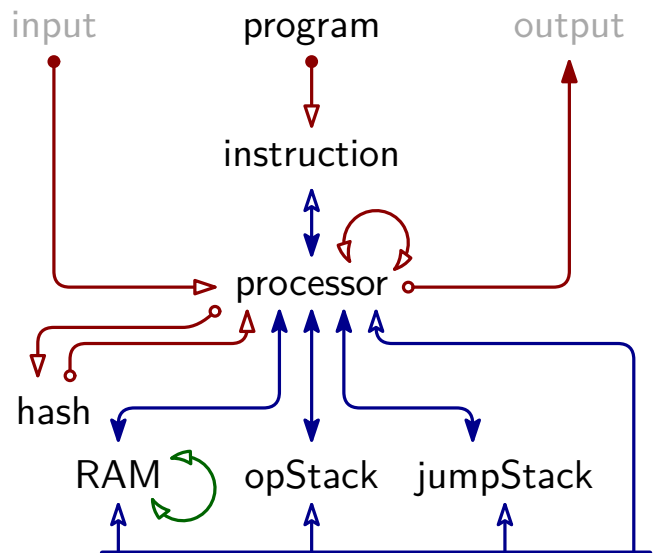


```

006  ⊖  pop
005  ⊕  push + a
004  ⊕  divine
013  ⊕  dup + i
021  swap + i
012  ○  nop
014  ⊖  skiz
029  ○  call + d
020  ○  return
028  ○  recurse
022  ⊖  assert
000  ○  halt
008  ⊙  read_mem
016  ○  write_mem
036  hash
044  divine_sibling  st12 % 2 = 0 ⇒ left node
052  ○  assert_vector
060  ☹  add
068  ☹  mul
076  ⊙  invert
084  split          hi → st0'
092  ☹  eq
100  lsb
108  xxadd
116  xxmulo
124  xinvert
132  ☼  xbmulo      st0 · (st1, st2, st3)
140  ⊕  read_io
030  ⊖  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	43	11	54
OpStack	5	2	7
RAM	7	6	13
JumpStack	6	2	8
Hash	49	2	51
Σ	117	26	143

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	RAMV	RAMP	
OpStack	CLK						IB1 ($\hat{=}$ shrink stack)							OSP		OSV						
RAM	CLK	RAMP	RAMV		IORD	bcpc0	bcpc1	clk_di														
JumpStack	CLK			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