Entry

X

```
{v0_1        := ϕ [v0, Entry] [v0_2, Z]
 v1_1        := ϕ [v1, Entry] [v1_2, Z]
 i           := ϕ [0, Entry]   [i_1, Z]
 phi_mul     := ϕ [0, Entry]   [next_mul, Z] }
 exitcond    := ( i == 32)
 i_1         := i + 1
 next_mul    := phi_mul + 0x9e3779b9
```

Y

```
 tmp         :=  v1_1 << 4
 tmp1        := tmp + k0_read
 tmp2        := v1_1 >> 5
 tmp3        := tmp2 + k1_read
 tmp4        := v1_1 + next_mul
```

Z

```
 tmp5        := tmp3 xor tmp4
 tmp6        := tmp5 xor tmp1
 v0_2        := tmp6 + v0_1
 tmp7        := v0_2 << 4
 tmp8        := tmp7 + k2_read
 tmp9        := v0_2 >> 5
 tmp10       := tmp9 + k3_read
 tmp11       := v0_2 + next_mul
 tmp12       := tmp11 xor tmp8
 tmp13       := tmp12 + tmp10
 v1_2        := tmp13 + v1_1
```

S_{pre}

X

```
{v0_1        := ϕ [v0, Entry] [v0_2, Z]
 v1_1        := ϕ [v1, Entry] [v1_2, Z]
 i           := ϕ [0, Entry]   [i_1, Z]
 phi_mul     := ϕ [0, Entry]   [next_mul, Z] }
 exitcond    := ( i == 32)
 i_1         := i + 1
 next_mul    := phi_mul + 0x9e3779b9
```

Y

```
 tmp         :=  v1_1 << 4
 tmp1        := tmp + k0_read
 tmp2        := v1_1 >> 5
 tmp3        := tmp2 + k1_read
 tmp4        := v1_1 + next_mul
```

Z

```
 tmp5        := tmp3 xor tmp4
 tmp6        := tmp5 xor tmp1
 v0_2        := tmp6 + v0_1
 tmp7        := v0_2 << 4
 tmp8        := tmp7 + k2_read
 tmp9        := v0_2 >> 5
 tmp10       := tmp9 + k3_read
 tmp11       := v0_2 + next_mul
 tmp12       := tmp11 xor tmp8
 tmp13       := tmp12 + tmp10
 v1_2        := tmp13 + v1_1
```

S_{loop}

```
{v0_1        := ϕ [v0, Entry] [v0_2, Z]
 v1_1        := ϕ [v1, Entry] [v1_2, Z]
 i           := ϕ [0, Entry]   [i_1, Z]
 phi_mul     := ϕ [0, Entry]   [next_mul, Z] }
 exitcond    := ( i == 32)
 i_1         := i + 1
```

$S_{preExit}$

Exit