```
Entry
X | \{ v0_1 := \phi [v0, Entry] [v0_2, Z] \}
    v1_1 := \phi [v1, Entry] [v1_2, Z]
               := \phi [0, Entry] [i_1, Z]
    phi_mul := \phi [0, Entry] [next_mul, Z] }
    exitcond := ( i == 32)
               := i + 1
    i 1
    if (exitcond) then goto Exit else goto Next step
    next mul := phi mul + 0x9e3779b9
Y
               := v1 1 << 4
     tmp
     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
            := tmp7 + k2 read
     tmp8
     tmp9 := v0 2 >> 5
     tmp10 := tmp9 + k3 read
           := v0 2 + next mul
     tmp11
     tmp12
           := tmp11 xor tmp8
     tmp13 := tmp12 + tmp10
     v1 2 := tmp13 + v1 1
     Go to X
  { v0_1
X
               := \phi [v0, Entry] [v0_2, Z]
    v1 1
               := \phi [v1, Entry] [v1_2, Z]
               := \phi [0, Entry] [i_1, Z]
               := \phi [0, Entry] [next_mul, Z] 
    phi_mul
    exitcond := ( i == 32)
               := i + 1
    i 1
    if (exitcond) then goto Exit else goto Next step
               := phi_mul + 0x9e3779b9
    next mul
Y
               := v1 1 << 4
    tmp
    tmp1
               := tmp + k0 read
                                                       S_{loop}
               := v1 1 >> 5
    tmp2
    tmp3
               := tmp2 + k1 read
    tmp4
               := v1 1 + next mul
Z
    tmp5
               := tmp3 xor tmp4
    tmp6
               := tmp5 xor tmp1
               := tmp6 + v0 1
    v0 2
    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
    Go to X
  { v0 1
                     := \phi [v0, Entry] [v0_2, Z]
                    := \phi [v1, Entry] [v1_2, Z]
              := \phi [0, Entry] [i 1, Z]
              := \phi [0, Entry] [next_mul, Z] 
   phi_mul
   exitcond := (i == 32)
              := i + 1
                                                      S_{preExit}
```

Exit