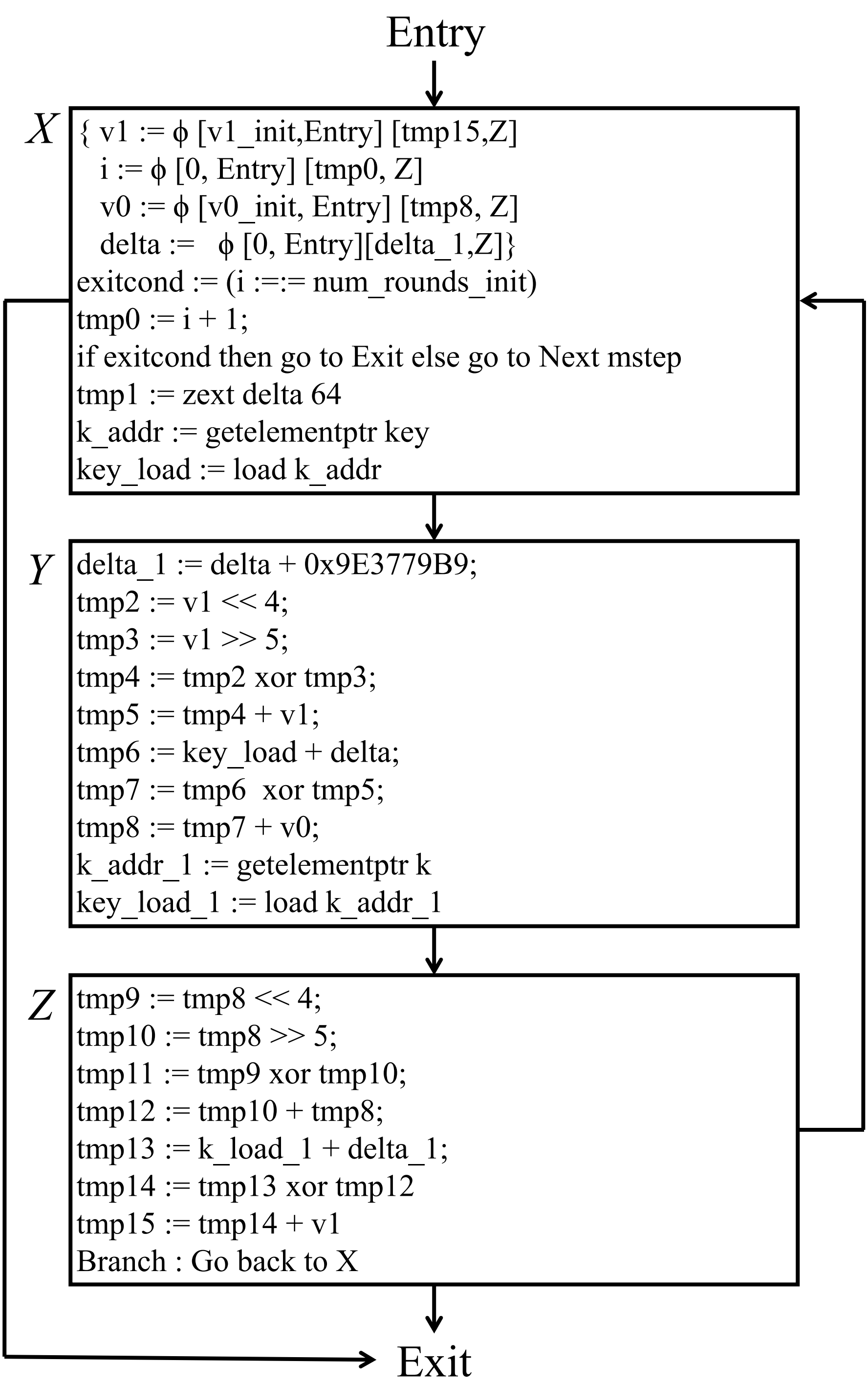
Entry
X
{ v1 := ϕ [v1_init,Entry] [tmp15,Z]
i := ϕ [0, Entry] [tmp0, Z]
v0 := ϕ [v0_init, Entry] [tmp8, Z]
delta := ϕ [0, Entry][delta_1,Z]}
exitcond := (i :=:= num_rounds_init)
tmp0 := i + 1;
if exitcond then go to Exit else go to Next mstep
tmp1 := zext delta 64
k_addr := getelementptr key
key_load := load k_addr
Y
delta_1 := delta + 0x9E3779B9;
tmp2 := v1 << 4;
tmp3 := v1 >> 5;
tmp4 := tmp2 xor tmp3;
tmp5 := tmp4 + v1;
tmp6 := key_load + delta;
tmp7 := tmp6 xor tmp5;
tmp8 := tmp7 + v0;
k_addr_1 := getelementptr k
key_load_1 := load k_addr_1
Z
tmp9 := tmp8 << 4;
tmp10 := tmp8 >> 5;
tmp11 := tmp9 xor tmp10;
tmp12 := tmp10 + tmp8;
tmp13 := k_load_1 + delta_1;
tmp14 := tmp13 xor tmp12
tmp15 := tmp14 + v1
Branch : Go back to X
Exit