TEA Step + Inverse in ACL2

```
(defun tea-encrypt-step (v0 v1 k0 k1 k2 k3 sum)
  (declare (xargs :guard (and (natp v0) (< v0 #
                 x100000000)
                              (natp v1) (< v1 #
                                     x100000000)
                              (natp k0) (< k0 #
                                     x100000000)
                              (natp k1) (< k1 #
                                     x100000000)
                              (natp k2) (< k2 #
                                     x100000000)
                              (natp k3) (< k3 #
                                     x100000000)
                              (natp sum) (< sum #
                                     x100000000))))
  (let* ((sum (mod (+ sum #x9E3779B9) #x100000000
                 ))
         (v0 (mod (+ v0 (logxor (+ (ash v1 4) k0)
                                (+ v1 sum)
                                (+ (ash v1 -5) k1))
                                      )
                  #x100000000))
         (v1 (mod (+ v1 (logxor (+ (ash v0 4) k2)
                                (+ v0 sum)
                                (+ (ash v0 -5) k3))
                                      )
                  #x100000000)))
    (mv v0 v1 sum)))
```

```
(defun tea-decrypt-step (v0 v1 k0 k1 k2 k3 sum)
  (declare (xargs :guard (and (natp v0) (< v0 #
                 x100000000)
                              (natp v1) (< v1 #
                                     x100000000)
                              (natp k0) (< k0 #
                                     x100000000)
                              (natp k1) (< k1 #
                                     x100000000)
                              (natp k2) (< k2 #
                                     x100000000)
                              (natp k3) (< k3 #
                                     x100000000)
                              (natp sum) (< sum #
                                     x100000000))))
  (let* ((v1 (mod (- v1 (logxor (+ (ash v0 4) k2)
                                (+ v0 sum)
                                (+ (ash v0 -5) k3)))
                  #x100000000))
         (v0 (mod (- v0 (logxor (+ (ash v1 4) k0)
                                (+ v1 sum)
                                (+ (ash v1 -5) k1)))
                  #x100000000))
         (sum (mod (- sum #x9E3779B9) #x100000000)
                 ))
    (mv v0 v1 sum)))

(defthm tea-step-invertible
  (implies (and (natp v0) (< v0 #x100000000)
                (natp v1) (< v1 #x100000000)
                (natp k0) (< k0 #x100000000)
                (natp k1) (< k1 #x100000000)
                (natp k2) (< k2 #x100000000)
                (natp k3) (< k3 #x100000000)
```