Use Merkle-Damgard transform to obtain a provably secure collision resistant hash function

Collisions for the hash function H are distinct inputs x and y such that H(x) = H(y).
A function H is collision resistant if it is infeasible for any probabilistic polynomial-time algorithm to find a collision in H. A family of functions indexed by s is given by

$$H^s(x) = H(s,x)$$

A hash function is a pair of algorithms (Gen, H) where Gen(1^n) outputs the index s (for choosing H^s). If H^s is defined only for inputs x of a certain length, we say it is a fixed length hash function.
A hash function(Gen, H) is collision resistant if for all probabilistic polynomial time adversaries A:

$$\Pr[\text{Output of Hash-game} = 1] <= \text{negl}(n)$$

The Merkle-Damgard transform says that if (Gen, h) is a fixed length collision resistant hash function, then (Gen, H) is a collision resistant hash function. It constructs hash functions $H^s(x)$ from fixed length hash functions (h^s) with inputs of length 2n and output length n.

Let (Gen,h) be a fixed length collision-resistant hash function for inputs of length 2l(n) and with output length l(n).

1. Gen : remains unchanged
2. H : on input a key s and a string x belonging to {0,1}* of length $L<2^{l(n)}$, do the following
    a. Set B := [L/l], that is, the number of blocks in x. Pad x with zeroes so its length is a multiple of l. Parse the padded result as the sequence of l-bit blocks x_1,....x_B. Set x_{B+1} := L, where L is encoded using exactly l bits.
    b. Set $z_0 := 0^l$
    c. For i = 1,..., B+1, compute $z_i := h^s(z_{i-1}||x_i)$
    d. Output z_{B+1}