Light Hash algorithm v4

Project

Design a hash module which generates a 32-bit digest for each 32-bit input data block performing the following operations:

$$
\begin{aligned}
&for\ (r = 0;\ r < 24;\ r++)\ \{\\
&\quad if(\ r == 0)\\
&\qquad H = SA(m);\\
&\quad else\\
&\qquad H = H \oplus IV;\\
&\quad H = \Theta(H);\\
&\quad H = \rho(H);\\
&\}\\
&d = FPX(H);
\end{aligned}
$$

Where:

r	is the number of rounds;
m	is the message;
d	is the digest of the message;
SA	is the State Array function, which transforms the 32-bit input data block *m* in an array of 4 elements, $m[i]$, each of 1 byte. The SA function works as follows:

$$
\begin{aligned}
&for\ (i = 0;\ i < 4;\ i++)\\
&\quad H[i] = m[i] \oplus IV[i];
\end{aligned}
$$

	$\boldsymbol{IV[0]}$	$\boldsymbol{IV[1]}$	$\boldsymbol{IV[2]}$	$\boldsymbol{IV[3]}$
Init. value	0x34	0x55	0x0F	0x14

Θ	is a permutation function which works as it follows:

$$
\begin{aligned}
&for\ (i = 0;\ i < 4;\ i++)\\
&\quad H[i] = H[3 - i];
\end{aligned}
$$

ρ	is a function which works as it follows:

$$
\begin{aligned}
&for\ (i = 0;\ i < 4;\ i++)\\
&\quad H[i] = (H[i] + 0\text{x}85) \bmod 0\text{xFD};
\end{aligned}
$$

FPX	is the Final Permutation and Xoring function, which works as it follows:

$$
\begin{aligned}
&for\ (i = 0;\ i < 4;\ i++)\\
&\quad d[i] = (H[3 - i]) \oplus IV[i];
\end{aligned}
$$

Additional design specifications

- The module shall have an asynchronous active-low reset port.
- The module interface should include appropriate flags to indicate the beginning and/or the end of a message, and, accordingly, the module should integrate appropriate logic for counting the bytes of the message.
- The module interface shall include an input flag to be driven as it follows: 1’b1, when input message byte on the corresponding input port is valid and stable (i.e. it can be used by the internal module logic), 1’b0, otherwise.
- The module interface shall include an output flag to be driven as it follows: 1’b1, when output digest on the corresponding output port is ready and stable (i.e., external modules can read and use it), 1’b0, otherwise.