**Full Hash Algorithm v2 (DES S-Box based)**

**Project Report**

**2021/2022**



**Project Report for the project of the course**

**‘Hardware and Embedded Security’ of the M.Sc. in Cybersecurity**

Professors: Students:

*Prof. Sergio Saponara Francesco Venturini*

*Ing. Luca Crocetti Pierfrancesco Bigliazzi*

# Specification Analysis

The given specification presents the requirements for the hardware design of an hash module based on the S-box of the DES algorithm. The input of the module is a message that can have an arbitrary length and that is divided in bytes to perform the computation, the output is a 32-bit digest formed by the concatenation of 8 nibbles.

The module is characterized by the following parameters:

module full\_hash\_des\_box(

input rst\_n,

input clk,

input M\_valid,

input [7:0] message,

input [63:0] counter,

output reg [31:0] digest\_out,

output reg hash\_ready

);

where:

* **rst\_n**: asynchronous active-low reset port
* **clk**: system clock
* **M\_valid**: input port that must be asserted when providing the input message byte message (1’b1 when input character is valid and stable, 1’b0 otherwise)
* **message**: input message byte, can be any 8-bit ASCII character
* **counter**: real byte length of message (if the message length is 1 byte, then C = 1)
* **digest\_out**: 32-bit output register for the computed hash value
* **hash\_ready**: output port that must be asserted when the generated output digest is available (1’b1 when output digest is valid and stable, 1’b0 otherwise)

For each byte of the input message the hash module performs the following operation:

for(r=0; r<4; r++)

for(i=0; i<8; i++)

H[i] = (H[(i+1) mod 8]) ⊕ S(M6)) ≪ ⌊𝑖/2⌋

We can see that the computation is composed by 4 rounds. In each round each byte of the message is compressed using the S-box of the DES algorithm (in which the first and last bits of the byte select the row of the table and the remaining 4 central bits select the column). The table outputs a specific 6-bit value than is then XORed with a value of the 32-bit array H according to the formula above (this variable is initialized with the values: 0x4B71DF03). The partial result of this operation is then left-shifted circularly by n bits, where n=⌊𝑖/2⌋ and ⌊ ⌋ is the floor function.

Once the last message byte has been processed, the hash module performs a final elaboration on the partial result given by the overlap of the previous computations:

for(i=0; i<8; i++)

H[i] = (H[(i+1) mod 8]) ⊕ S(C6[i])) ≪ ⌊𝑖/2⌋

In this case, the S-box is called with a parameter C6[i], a 6-bit vector that is obtained from the i-th byte of the input counter according to the following operation:

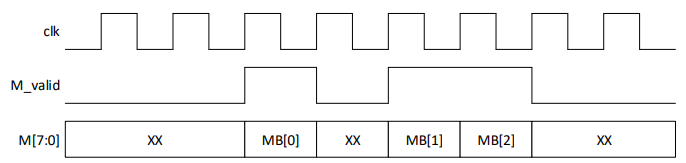
𝐶6 [𝑖] = {𝐶[𝑖][7] ⊕ 𝐶[𝑖][1], 𝐶[𝑖][3], 𝐶[𝑖][2], 𝐶[𝑖][5] ⊕ 𝐶[𝑖][0], 𝐶[𝑖][4], 𝐶[𝑖][6]}

The S-box is structured as the following:

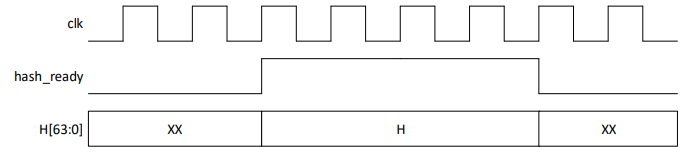
Immagine che contiene tavolo

Descrizione generata automaticamente

The expected waveforms are the following. For the input:



For the output:



# Block Diagram and Design Choices

# Expected Waveforms

# Testbench

# Implementation of RTL design on FPGA and results

# Static Timing Analysis