Let me report the progress for the period to 03-06/03. During the period the following actions are performed:

1) the literature in the field of cryptanalysis and polynomial decomposition/approximation is initiated, the compact representation of the system in terms of polynomial coefficients, wavelet coefficients and sha2 parameters is analysed;

2) the work in the Deep Neural Network implementation for the SHA-2 (complete or partial) approximation is considered, the scope of Deep Neural Network literature is analysed, the references are attached to the report);

3) the model for the SHA-2 approximation is proposed, the considered model is aimed to approximate the SHA-2 in the R space, as the SHA-2 is interpreted as a continuous function in R. The model is aimed to deal with the infeasibility problem and gradient instability (caused by multiple layers of the SHA-2 structure). The model has the benefits of the LSTM structure with the encapsulation of the specific activations. At the same time minimalism in the parametrization in tandem with use of cross-entropy loss function, L2 regularization, almost-complete weight initialization give the ground to believe the directional search is possible;

4) the directional search strategies are applied in order to:
   - find the solution (for the system defined by SHA-2 with fixed W and mask b parameters);
   - search the nearest solution with the minimal changes in W;
   - make transition from one to another systems defined in SHA-2 (same as systems are defined by polynomials);

Let me describe the proposed model in details.

LSTM Model for the SHA-2 Approximation

The SHA-2 system is interpreted as the continuous function which the arguments and dependent variable defined in the R space. The threshold functions are overloaded using the squashing functions. (Overloading is performed by Neural Networks defined subsequently).

The model is using the minimalism in the parametrization, some terms/calculations irrelevant to optimized parameters are considered constant, thus some terms/parts of the calculation are not overloaded. Overloaded are only that particular SHA-2 terms which are directly related with optimized parameters or which potentially might be causing infeasibility problems).

Let me remind you the SHA-2 system (make some methodological formalis) in order to clear which terms/parts are overloaded by the model. The common (high-level) block-scheme of the SHA-2 is shown in the figures below.

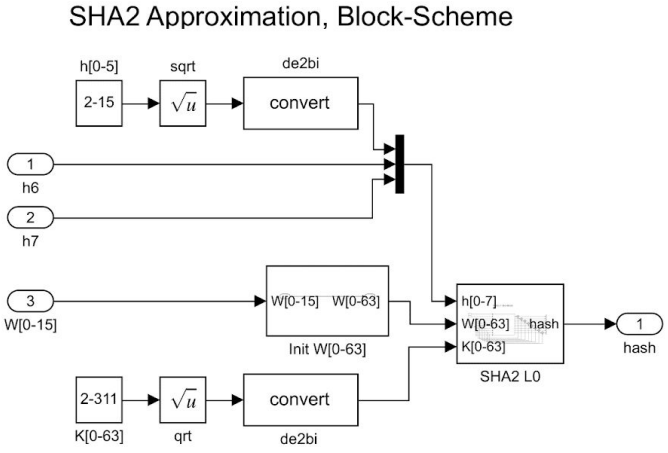

Fig. 1.1 SHA-2 Block-Scheme, high level description

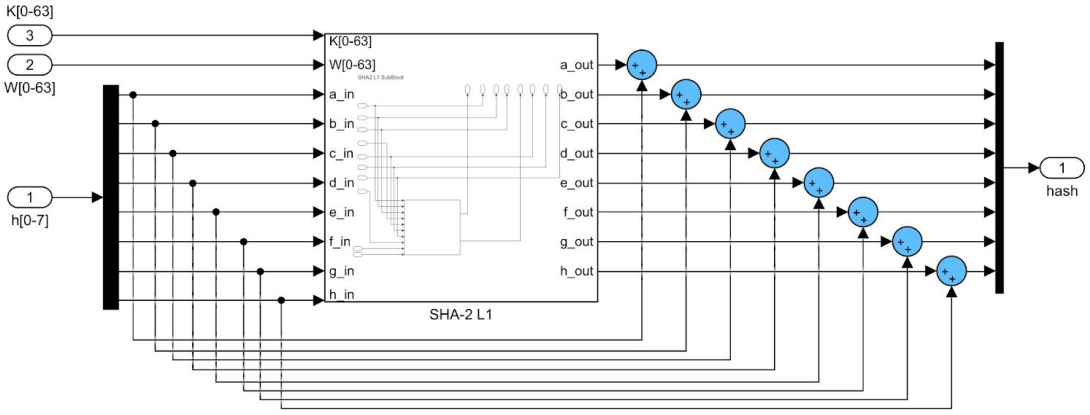

Fig. 1.2 SHA-2 Block-Scheme, high level (L0) description

Accordingly model the SUM, XOR and AND operators are overloaded. Overloaded operators are displayed by the light blue signs. The subsequent/detailed methodological description of SHA-2 is shown in the **Fig.1.3-1.4**.

From the **Fig.1.3-1.4**. observe the only operators overloaded are SUM, XOR and AND. (The operator NOT is not overloaded, for simplicity it is replaced by $1-x$ term. Also the **circshift** "circular shift" operator (abbreviated by z^{-n}) is not overloaded. (The operator **lshift** "left shift" is not displayed since it used only in the Init W block, see the **Fig. 1.1**). Also some terms related to $W[i]$ and $K[j]$ are constant (irrelevant to $h[0-7]$ and thus are not overloaded, for this reason some SUM terms related only to $W[i]$ and $K[j]$ are not overloaded in the default version of the model (also during this reason the methodological description of **Init W** sub-block is not extended).

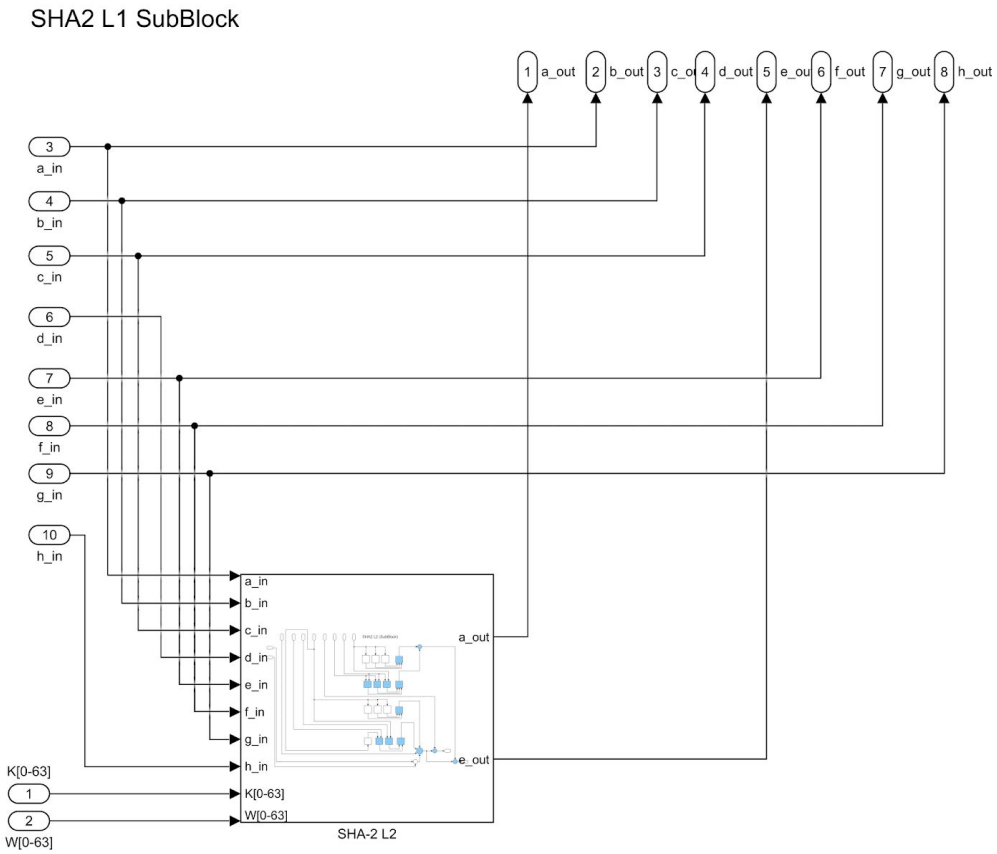

Fig. 1.3 SHA-2 Block-Scheme, mid-level (L1) description (single-round displayed)

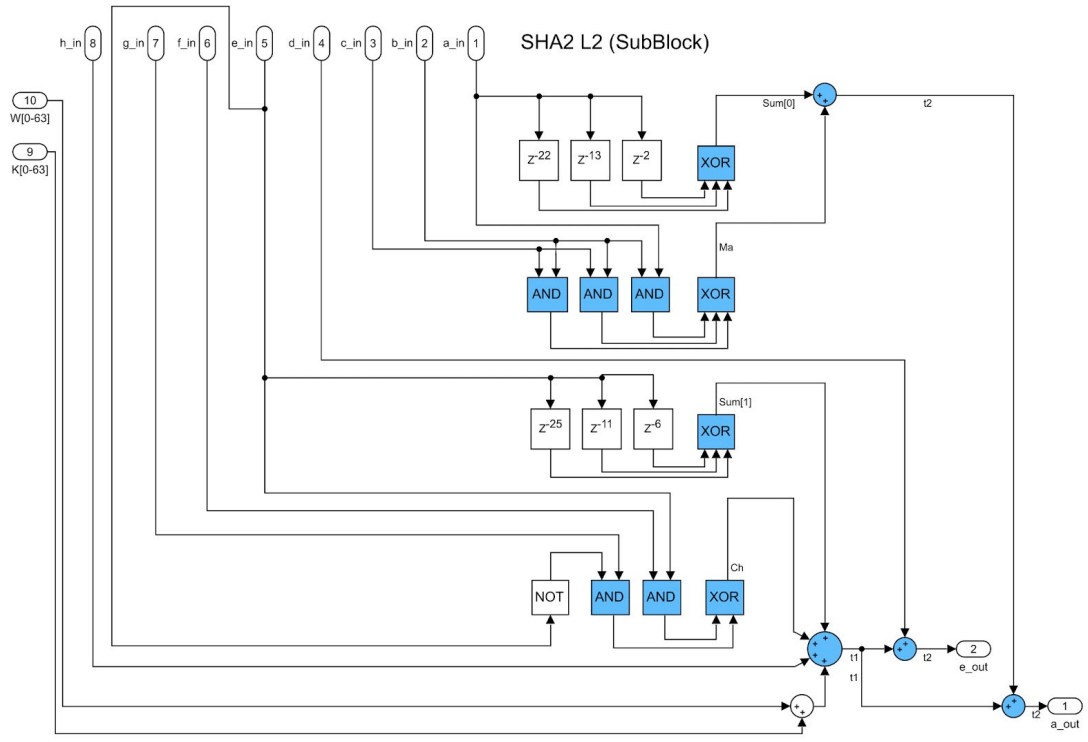

Fig. 1.4 SHA-2 Block-Scheme, lower-level (L2) description

Let me summarize the scope of operators overloaded:

Table 1.1. The Summary for Operators Overloaded

Operator	Overloaded by	Type of NN	#In ~ #Out	# In Round	# At all
SUM	SUM NN	LSTM	32 ~ 32	[8] + [10]*r	648
XOR	XOR NN	Pattern Net	1 ~ 1	[0] + [4]*r	256
AND	AND NN	Pattern Net	1 ~ 1	[0] + [5]*r	320

The total number of operators overloaded is 8 + 18 *r, which is from 768 to 1224 depending on number of rounds. (Note we use the 32 cell 1-to-1 LSTM Network for the SUM operator and Pattern Net for AND, XOR NN - these are shallow 2 layer networks with tanh-sigmoidal and log-sigmoidal activations).

Let me describe briefly how exactly particular networks are applied/encapsulated in the model approximating the SHA-2 structure. There is level-by-level explanation of the proposed model, from the lower level to the higher.

XOR and AND Approximation NN, Nano-Level Structure

At the lower level the structure of the Neural Networks overloading the XOR and AND operators is shown in the figure below. On the Nano Level these two described neural networks has same architecture - shallow 2 layer networks with tanh-sigmoidal and log-sigmoidal activations. We use two inputs $X1$ and $X2$ in R, and one output Y in R space. (Note the weight of the network for AND and XOR operators are different).

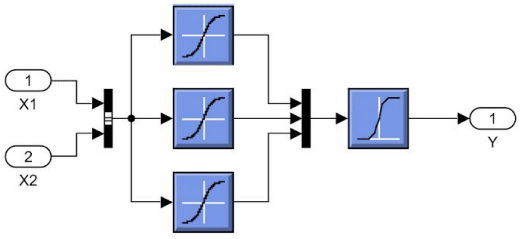

Fig. 1.5 XOR and AND Approximation Neural Networks, nano-level

SUM Approximation NN, Nano-Level Structure

Since the SUM Approximation NN is the LSTM Network, the Network on -the nano level is interpreted as the LSTM cell see the figure below. The defined uses as inputs the two contemporaneous bits of $X1$ and $X2$ and output k-th bit of Y in the $Y = X1 + X2$. The cell inputs and outputs are the variables in R. The detailed description of the LSTM cell shown in the figure below.

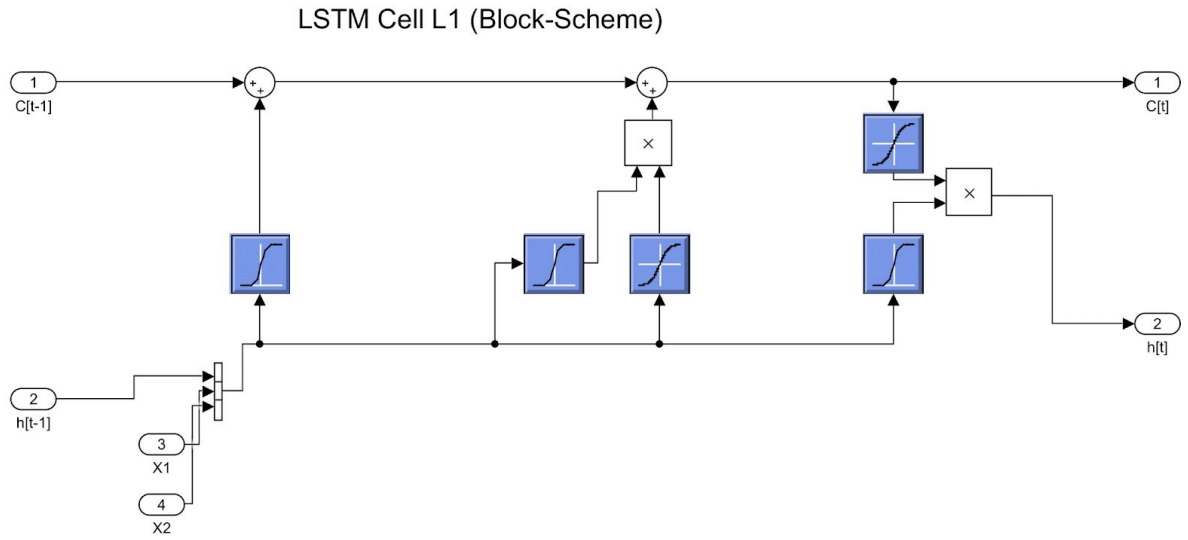

Fig. 1.6 XOR and AND Neural Networks, nano-level

Let us get focused how the higher level is methodologically described.

XOR and AND Approximation NN, Micro-Level Structure

The structure of the XOR and AND approximation Neural networks (in tandem with shortcuts used for the oveloads) are displayed as per block-scheme below.

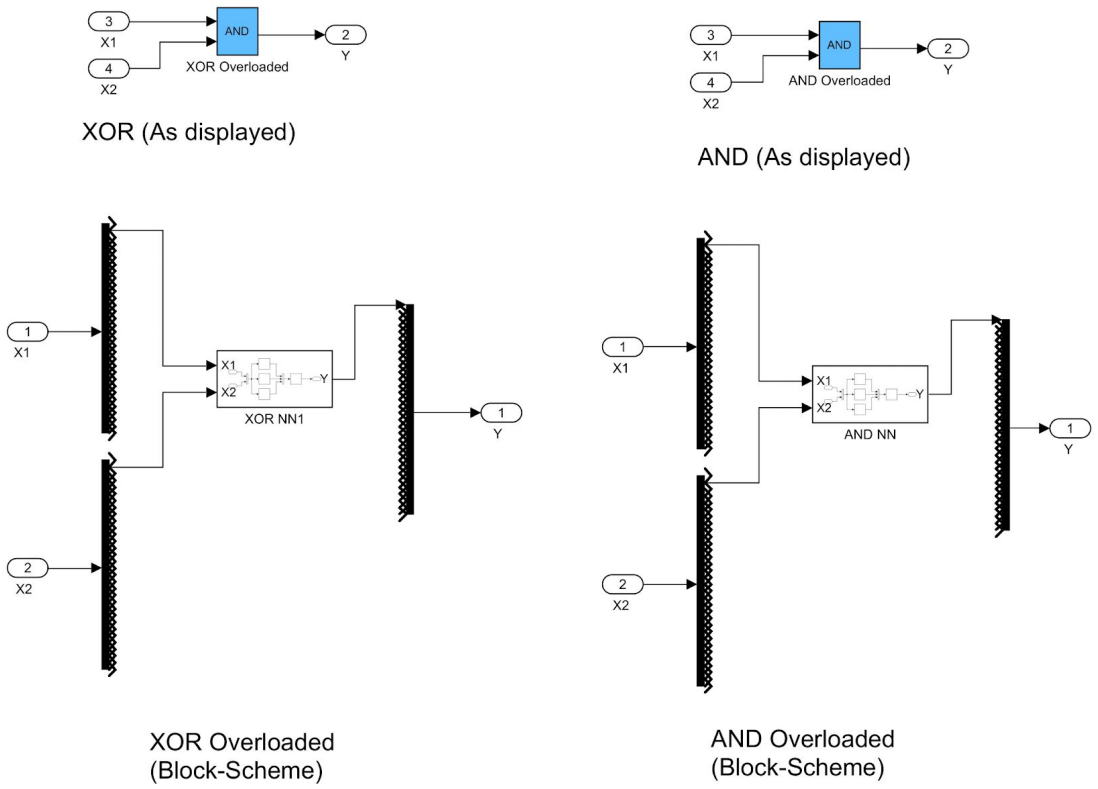

Fig. 1.7 a) XOR Neural

Fig. 1.7. b) AND Neural Network

From the Fig. 1.7. observe the both networks uses the k-th (contemporaneous) bit of inputs $X1$ and $X2$ and provide k-th bit of Y in the output.(With bit undersoon the equivalent in R).

SUM Approximation NN, Micro-Level Structure

The structure of the LSTM Network is displayed in the block-schemes below. See the L0 and L1). From the two block-schemes shown below, the 32-bit input of the $X1$ and $X2$ are transferred to 32-bit output Y (in this case the bit equivalents in R are considered).

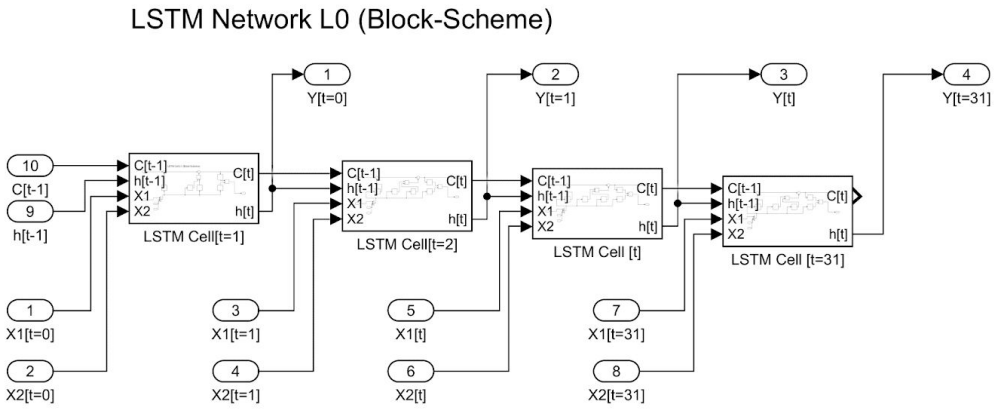

Fig. 1.8 SUM Neural Networks, LSTM Cells aggregated (micro-level)

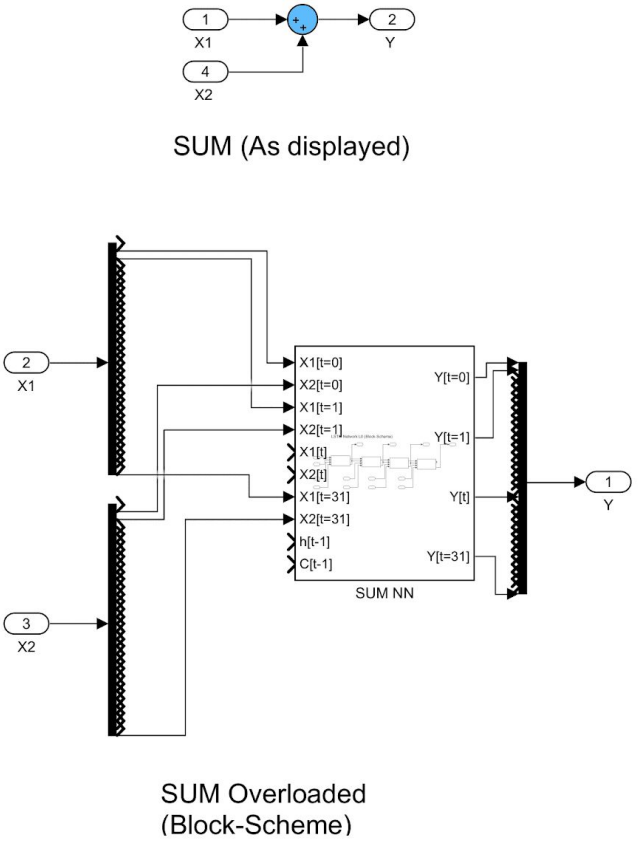

Fig. 1.9 SUM Neural Networks, LSTM Cells aggregated (view outside the block)

From the bloch-schemes in Fig. 1.8-1.9 the submission is performed from lower to higher bits using the 32 cell LSTM Neural Network with 1-to-1 relation, same as it is done by using bitwise addition by column method.

SUM, XOR, AND Approximation NN, Higher-Level Structure

In current section the higher level aggregation is methodologically described. The key idea of the model is to prevent infeasibility by encapsulating the SUM, XOR and AND Neural Network in the LSTM Cell with subsequent aggregation. The encapsulation scheme is displayed in the block-scheme below.

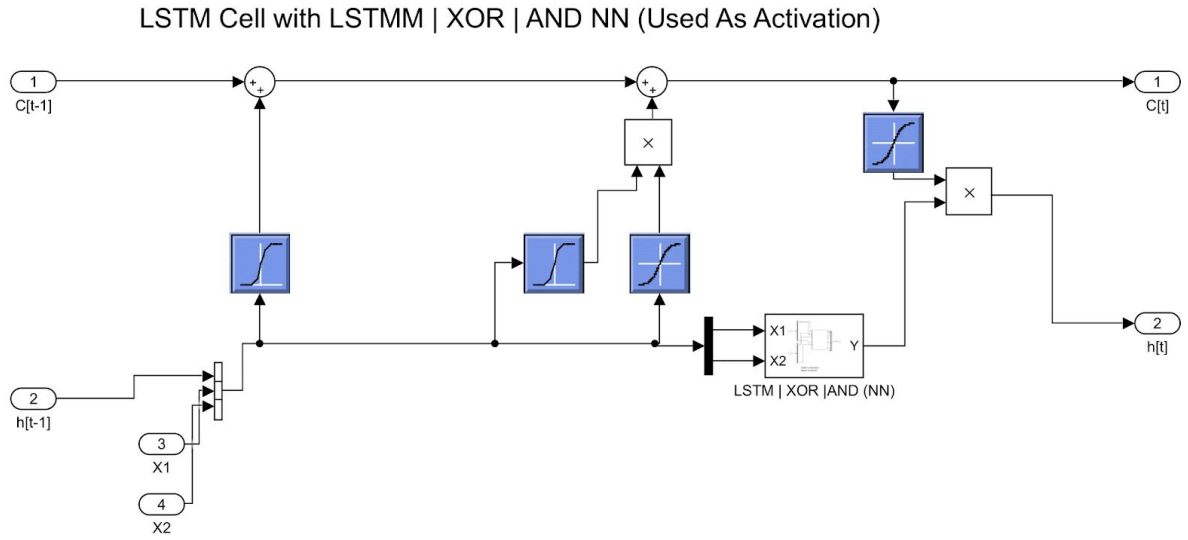

Fig. 1.10 XOR, ANDor SUM Approximation NN encapsulated in the LSTM cell

The idea of the model is to alter the sigmoidal (output gate) of the LSTM cell to another activation function, here another NN Structure is interpreted as an activation function. The subsequent (higher order) aggregation of LSTM is shown in the block-scheme below.

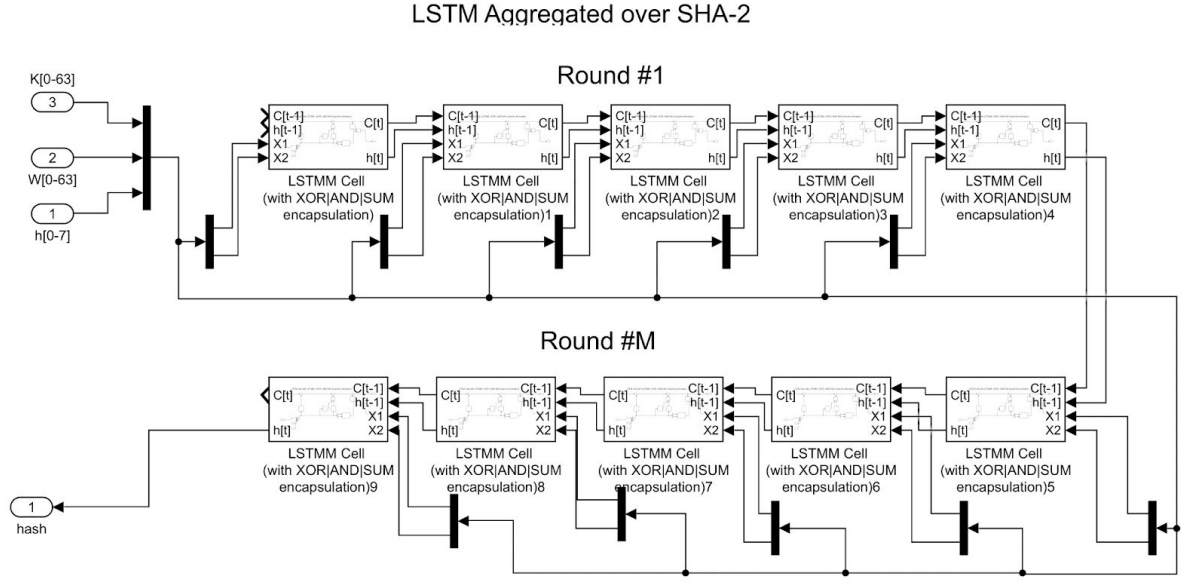

Fig. 1.11 Aggregation of the LSTM Cells (with subsequent XOR|AND|SUM NN encapsulations)

As observed from the Fig.1.11 the calculations of the SHA-2 round are made consequently, as per methodological description. The only change is that “+”, “V” and “&” operators are overloaded in R, (the terms “~” are replaced by $1-x$). The rest calculations performed without changes.

The changes of the activation function are made in order to prevent infeasibility:

- infeasibility in the prev models were caused by multiple sigmoidal activation, the results in-between sigmoids remain constant)
- in current model multiple sigmoidal activations are splitted, the term from “sigmoid” is altered to “sigmoid * hyperbolic tangent (hidden state)”, the LSTM “channel” is established over the whole SHA-2 sheme.