VLSI Architecture of a Cellular Automata based One-Way Function

D. Mukhopadhyay, P. Joshi* and D. RoyChowdhury*
Department of Computer Science, Dept of Computer Science and Engg, Madras, India
Email: debdeep@cse.iitm.ernet.in
*Department of Computer Science, Dept of Computer Science and Engg, Kharagpur, India
Email: pallavi.joshi@gmail.com, drc@cse.iitkgp.ernet.in

Abstract— **In this paper, a technique to generate expander graphs using Cellular Automata (CA) has been presented. The special class of CA, known as the Two Predecessor Single Attractor Cellular Automata (TPSA CA) has been characterized. It has been shown that the expander graphs built using the TPSA-CA are promising and lead to the development of strong one-way functions. The paper proposes an efficient architecture for the one-way function and implements it on an FPGA platform. Most of the widely used key establishment algorithms employ one-way functions based on modular exponentiation that are computationally very expensive. However, the implementation of the proposed architecture shows that the TPSA based one-way algorithm can be implemented very efficiently with very less consumption of computational resources like area and clock cycles. Such a one-way function can be an ideal replacement of modular exponentiation based one-way functions and thus help to develop fast and secure key establishment protocols.**

Index Terms— **One Way Function, Cellular Automata, Expander Graphs, FPGA Implementations, Area-Delay Product**

I. Introduction

Expander Graphs have been a significant tool both in theory and practice. It has been used in solving problems in communication and construction of error correcting codes as well as a tool for proving results in number theory and computational complexity. The combinatorial properties of the expander graphs can also lead to the construction of one-way functions [1]. Informally, the one-way functions are a class of functions in which the forward computation is easy, but the inverse is hard to find. The one-way functions form an important core of all key agreement algorithms which are an important step in secure electronic communication. The well known Diffie-Hellman key exchange algorithm [2] provides a ground-breaking solution to the problem of secured key distribution. However the security of the algorithm depends on the one-wayness of the modular exponentiation, which is a costly process in terms of computational resources. Since the seminal paper of Diffie-Helmann, there has been efforts in developing key exchange protocols whose security lies on one-way functions which are computationally efficient. However designing a strong one-way function which is computationally strong and yet hardware efficient is a challenging task.

The present work characterizes a special class of Cellular Automata (CA) [3], known as the *Two Predecessor Single Attractor Cellular Automata* (TPSA-CA) to generate expander graphs on the fly. The elegance of the scheme is that it uses regular, cascadable and modular structures of CA to generate random d regular graphs of good expansion property with very less storage. The state transitions of each TPSA is captured in a single state, which is known as the graveyard of the CA. Finally, the expander graphs have been used to construct the one-way function according to the proposal of [1]. The entire algorithm has been prototyped on a Vertex Xilinx XCV-1000 platform. Results show that the scheme is very elegant in the fact that very high throughputs are achievable with a minimal cost of hardware and power. The performance has been compared with that of efficient implementations of modular exponentiation in $GF(2^m)$ and results show that the proposed one-way function has a much superior VLSI implementation than a conventional one-way function.

The outline of the paper is as follows: *Section II* describes some of the preliminaries of expander graphs. The TPSA-CA is characterized in *section III* and is used to generate expander graphs. Subsequently, *section IV* presents the construction of the CA based one-way function which is implemented on Xilinx XCV-1000 platform in *section V*. The chapter is concluded in *section VI*.

II. Preliminaries on Expander Graphs

Informally *expander graphs* are a class of graphs $G = (V, E)$ in which every subset S of vertices expands quickly, in the sense that it is connected to many vertices in the set $\overline{S}$ of complementary vertices. It may be noted that the graph may have self loops and multiple edges. The following definition states formally the *expansion property* of these class of graphs [11].

Definition 1: The edge boundary of a set $S \in G$, denoted $\delta(S)$ is $\delta(S) = E(S, \overline{S})$ is the set of outgoing edges from S. The *expansion parameter* of G is defined

This paper is based on "An Efficient Design of Cellular Automata based Cryptographically Robust One-Way Function" by D. Mukhopadhyay, P. Joshi and D. RoyChowdhury, which appeared in the Proceedings of the 20th IEEE Conference on VLSI Design (VLSID), Bangalore, India, January 2007. © 2007 IEEE.

as:

$$h(G) \quad = \quad \min_{S:|S|\leq n/2}\frac{|\delta(S)|}{|S|}$$

where $|S|$ denotes the size of a set S.
There are other notions of expansion, the most popular being counting the number of neighbouring vertices of any small set, rather than the number of outgoing edges.

Following are some examples of expander graphs [12].

Example 1: Let G be a complete graph on n vertices i.e, the graph in which every vertex is connected to every other vertex. Then for any vertex in S, each vertex is connected to all the vertices in $\overline{S}$, and thus $|\delta S| = |S| \times |\overline{S}| = |S|(n-|S|)$. Thus the expansion factor is given by:

$$h(G) \quad = \quad min_{S:|S|\leq n/2}(n-|S|) = \lceil \frac{n}{2} \rceil$$

Example 2: Let G be a random d-regular graph, in which each of n vertices is connected to d other vertices chosen at random. Let S be a subset of atmost $n/2$ vertices. Then a typical vertex in S will be connected to roughly $d \times |\overline{S}|/n$ vertices in $\overline{S}$, and thus $|\delta S| \approx d \times |S||\overline{S}|/n$, and so

$$\frac{|\delta(S)|}{|S|} \quad \approx \quad d\frac{|\overline{S}|}{n}$$

Since, $|\overline{S}|$ has its minimum at approximately $n/2$ it follows that $h(G) \approx d/2$, independently of the size n.

Definition 2: A **family of expander graphs** (G_i where $i \in n$) is a collection of graphs with the following properties:

- The graph G_i is a d-regular graph of size n_i (d is the same constant for the whole family). $\{n_i\}$ is a monotone growing series that does not grow too fast (e.g $n_{i+1} \leq n_i^2$)).
- For all i, $h(G_i) \geq \epsilon \geq 0$.

Although d-regular graph random graphs on n vertices define an expander, for real life applications it is necessary to have more explicit constructions on $O(2^n)$ vertices, where n is the parameter defining the problem size. This is because to store a description of a random graph on so many vertices requires exponentially much time and space.

The properties of the eigenvalue spectrum of the adjacency matrix $A(G)$ can also be used to understand properties of the graph G.

The **adjacency matrix** of a graph G, denoted by $A(G)$ is an $n \times n$ matrix such that each element (u, v) denotes the number of edges in G between vertex u and vertex v [11]. For a d-regular graph, the sum of each row and column in $A(G)$ is d. By definition the matrix $A(G)$ is symmetric and therefore has an orthonormal base $v_0, v_1, \ldots, v_{n-1}$, with eigenvalues $\mu_0, \mu_1, \ldots, \mu_{n-1}$ such that for all i we have $Av_i = \mu_i v_i$. Without loss of generality we assume the eigenvalues sorted in descending order $\mu_0 \geq \mu_1 \geq \ldots \geq \mu_{n-1}$. The eigenvalues of $A(G)$ are called the spectrum of G. The following two results are important in estimating the expansion properties of the graph.

1) $\mu_0 = d$
2) $\frac{d-\mu_1}{2} \leq h(G) \leq \sqrt{2d(d-\mu_1)}$

Thus, the parameter $d-\mu_1$, also known as the *Spectral Gap* gives a good estimate on the expansion of the graph G. The graph is an expander if the spectral gap has a lower bound ϵ' such that $d - \mu_1 > \epsilon'$.

In the next section we present the construction of a family of random d regular graph using the properties of a special class of CA, known as the Two Predecessor Singe Attractor Cellular Automaton (TPSA CA). It has been shown that the graph has good expansion properties. The merit of the construction lies in the fact that the generation is extremely simple and leads to an efficient design as developed in subsequent sections.

III. Expander Graphs using TPSA CA

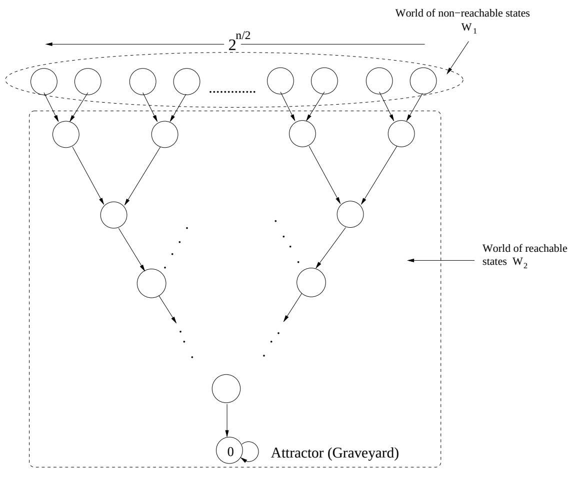
TPSA CA are a special class of non-group CA in which the state transition graph forms a single inverted binary routed tree at all zero state (**Fig. 1**). Every reachable state in the state transition graph has exactly two predecessors. The only cyclic state is the all zero state (for a non-complemented TPSA CA), which is an attractor (or graveyard). If T_n is the characteristic matrix of an n cell automaton then the necessary and sufficient conditions to be satisfied by the Transition matrix for the CA to be TPSA CA is [3]:

1) Rank(T_n)=$n-1$
2) Rank($T_n + I_n$)=n, I_n being an $n \times n$ identity matrix
3) Characteristic Polynomial = x^n
4) Minimal Polynomial = x^n

The following results [3] characterize the state transition of the non-complemented TPSA CA.

Lemma 1: [3] For an n cell TPSA CA with characteristic polynomial x^n and minimal polynomial x^n, (i) the number of attractors is 1, the all zero state, (ii) the number of states in the tree is 2^n.

Lemma 2: For an n cell TPSA CA having $m(x) = x^n$ the depth of the tree is n.



World of non−reachable states W1
2^{n/2}
World of reachable states W2
0
Attractor (Graveyard)

Figure 1. **The state transition graph of a non-complemented TPSA CA**

Following is an example of a 4 cell non-complemented TPSA CA.

Example 3: The state transition matrix for a 4 cell CA is denoted by:

$$T_4 = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}$$

The state transition diagram of the TPSA CA is shown in **Fig. 2(a)**. As an example let us compute the next state of 14, which in binary form is $X = (1110)^T$. Thus the next state is obtained as $Y = T_4 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = 1$

Thus the next state of 14 is 1, which may be observed in **Fig. 2(a)**. Here $\{0\}$ is the *attractor* or *graveyard* state. The states $\{5, 6, 4, 7, 8, 11, 9, 10\}$ make the *non-reachable world*, while the states $\{13, 14, 12, 15, 1, 2, 3, 0\}$ make the *reachable* world. The corresponding interconnection is given in **Fig. 2(b)**. As may be observed that the structure comprises of local interconnections leading to efficient designs.

Next, we present an method to recursively synthesize an n cell TPSA. The state transition matrix of the n cell TPSA is denoted by T_n and is generated from an $n-1$ cell TPSA CA characterized by the matrix T_{n-1}. The following theorem describes the property exploited in the construction.

Theorem 1: Given that T_{n-1} is the characteristic matrix of an $(n-1)$ cell TPSA, the matrix T_n denoted by:

$$T_n = \begin{pmatrix} & & & & | & 0 \\ & & T_{n-1} & & | & \vdots \\ & & & & | & 0 \\ - & - & - & - & - & - \\ 0 & \dots & 0 & 1 & | & 0 \end{pmatrix}$$

represents the characteristic matrix of an n cell TPSA.

Proof: It is evident that since the element at the n^{th} row and $(n-1)^{th}$ column is 1 and by the construction methodology all the rows have 0 in the $(n-1)^{th}$ columns the row added is linearly independent from the other rows of T_n. Hence it adds by 1 to the rank of T_{n-1}. Thus, $rank(T_n) = rank(T_{n-1}) + 1 = n - 1 + 1 = n$

Similarly, using the fact that $rank(T_{n-1} \oplus I_{n-1}) = n-1$ (where I_{n-1} is the identity matrix of order $n-1$), we have $rank(T_n \oplus I_n) = n$. The characteristic polynomial of the matrix T_n, denoted by $\phi_n(x)$ is evaluated as $det(T_n \oplus xI_n)$, where det denotes the determinant. Thus we have,

$$\phi_n(x) = \det \begin{pmatrix} & & & & | & 0 \\ & & & & | & 0 \\ & & & & | & \vdots \\ & T_{n-1} & \oplus & xI_{n-1} & | & 0 \\ & & & & | & \vdots \\ & & & & | & 0 \\ & & & & | & 0 \\ - & - & - & - & - & - \\ 0 & \dots & 0 & 1 & | & x \end{pmatrix}$$

$$= x\phi_{n-1}(x) = x.x^{n-1} = x^n$$

($\phi_{n-1}(x)$ denotes the characteristic polynomial of T_{n-1})

In order to evaluate the minimal polynomial we make use of the following proposition.

Lemma 3: Let $\phi_n(x)$ and $\psi_n(x)$ be the characteristic polynomial and the minimal polynomial of the matrix T_n, respectively. Let the greatest common divisor (gcd) of the matrix $(T_n \oplus I_n x)^\vee$ that is the matrix of algebraic complements of the elements of the matrix $(T_n \oplus I_n x)$ be $d(x)$. Then, $\phi_n(x) = d(x)\psi_n(x)$.

From the matrix $(T_n \oplus I_n x)^\vee$ it may be observed that the element at the position $(0, n)$ is 1 and thus the gcd $d(x)$ is also 1. Thus the minimal polynomial is equal to the characteristic polynomial which is x^n. Thus, we observe that the construction follows all the four necessary and sufficient requirements of a TPSA CA. This completes the proof. ■

Example 4: Given the fact that

$$T_2 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

is the characteristic matrix of a 2 cell TPSA CA. Thus, using the above theorem it is evident that:

$$T_3 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

is the characteristic matrix of a 3 cell TPSA CA.

We have seen above that the state transition in the above class of TPSA CA is governed solely by the characteristic matrix. This class of CA is known as the non-complemented TPSA CA. On the contrary when the next state is obtained by the application of the characteristic matrix and then xoring with a vector F, the CA is known as the complemented TPSA-CA.

The following results show how complementing the state transition function of the non-complemented CA generates a class of automaton with the same properties as the original TPSA CA.

Lemma 4: Corresponding to a non-complemented TPSA CA M_1 and a state Z, there exists a complemented CA M_2 with state Z as an attractor. If the characteristic matrix M_1 be indicated by T_n and it is required to build a complemented TPSA CA such that Z is the graveyard (attractor) then the characteristic matrix of the complemented CA, $\overline{T}_n$ is related to T_n by

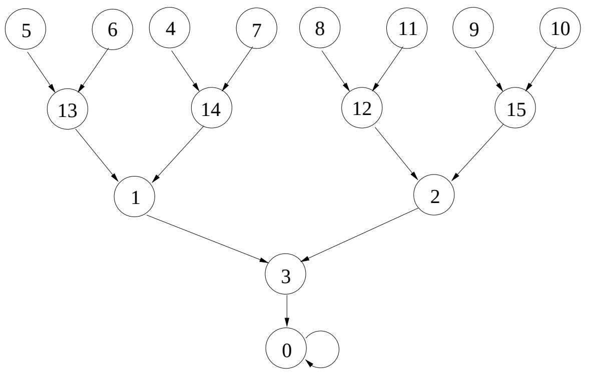
$$\overline{T}_n(X) = T_n(X) \oplus (I_n \oplus T_n)Z$$

where X is the seed to the CA and I_n is the identity matrix of order n.

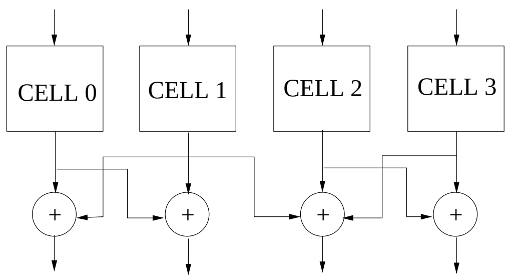
Lemma 5: A complemented TPSA CA has the same structure as a non-complemented TPSA CA. To emphasize

- Number of attractors in the complemented CA is the same as that in the original non-complemented CA.
- Number of reachable states and non-reachable states are same as that in the original non-complemented CA.

Lemma 6: If any state Z in the non-reachable world of a non-complemented CA is made the graveyard in a complemented TPSA, then the non-reachable elements



(a) The state transition graph of a 4 cell non-complemented TPSA CA



(b) The interconnection of a 4 cell non-complemented TPSA CA

Figure 2. **A** 4 **cell non-complemented TPSA CA**

become elements of the reachable world in the complemented CA and viceversa. Thus the reachable world (W_1) and the non-reachable world (W_2) exchange themselves (**Fig. 3**).

Proof: Let X and Z be two non-reachable elements in the n cell non-complemented CA with characteristic matrix T_n. Let X be the l^{th} level sister of Z. In all cases $l < n$. Thus, we have:

$$T_n^l(X) \;=\; T_n^l(Z)$$

Let us consider the state transition diagram of the complemented CA with Z as the graveyard. The state transition of the complemented CA is indicated by $\overline{T}_n$. We shall prove that in this state transition graph X is a reachable state. Let, the depth of X in the graph of the complemented CA be t. If t is less than n then X is a reachable state. Since, Z is the graveyard of this graph we have:

$$\begin{aligned} \overline{T}_n^t(X) &= Z \\ T_n^t(X) \oplus (I_n \oplus T_n^t)Z &= Z \\ T_n^t(X) &= T_n^t(Z) \end{aligned}$$

Thus, X and Z are t^{th} level sisters in the state transition graph of the non-complemented CA. But we know that they are l^{th} level sisters. Thus $t = l < n$. Thus, the depth of X is lesser than n and hence X is a reachable state in the state transition graph of the complemented CA. ■

A. Construction of expander graph using the TPSA CA

The TPSA CA can be effectively used to generate a random d regular graph on the fly. It may be noted that the entire nature of the graph is stored in the graveyard state, thus leading to a very compact storage of the graph. This is because given the graveyard state, the entire transition graph can be obtained.

In order to construct the d regular graph we proceed as follows: Let $Z_1 \in W_1$ (non-reachable world in the non-complemented TPSA CA) and $Z_2 \in W_2$ (reachable world in the non-complemented TPSA CA). Let, G_1 and G_2 be the state transition graphs with Z_1 and Z_2 as the graveyards respectively.

Clearly, in G_1 if $X \in W_1$, $degree(X) = 3$ and if $X \in W_2$, $degree(X) = 1$. Similarly, in G_2 if $X \in W_1$, $degree(X) = 1$ and if $X \in W_2$, $degree(X) = 3$. Here $degree$ is defined as the sum of the $indegree$ and the $outdegree$ in the corresponding graph.

Thus, in the graph G obtained by a union operation in the graphs G_1 and G_2, allowing multiple edges and self loops, we have for $X \in G$, $degree(X) = 4$. If we continue the union operation in the above method we have $degree(X) = 2(t + 1)$, where t is the number of union operations. **Table I** shows the result of an experimentation performed with the TPSA based regular graph. It measures the value of the two largest eigen values for random TPSA based graphs for degree $4, 8, 12$ and 16. The difference between the largest two eigen values is known as the spectral gap and should be large for good expansion of the graph. Results show that the spectral gap and hence the expansion increases proportionately with the number of union operations (t).

B. Setting parameters of the d regular TPSA based graph for good Expansion Properties

In the present section we compute the expansion obtained in the d regular TPSA based graph in terms of the parameters of the graph G. Let the number of nodes in the graph be n and the degree of each node is d. Let us consider a random d regular graph (**Fig. 4**) the subset A with αn vertices ($0 < \alpha < 1$).

For the graph G to have good expansion properties the set A should have more than βn ($0 < \beta < 1$) neighbours outside A. The probability of such an event should be high.

Equivalently, we may state that the probability that the number of neighbours of the vertices of A outside A is less than βn is negligible. Let us fix A and B such that $|A| = \alpha n$ and $|B| = \beta n$.

It is required that the vertices of A be matched to $N(A)$ (neighbours of A outside A) s.t $N(A) \subset B$. If

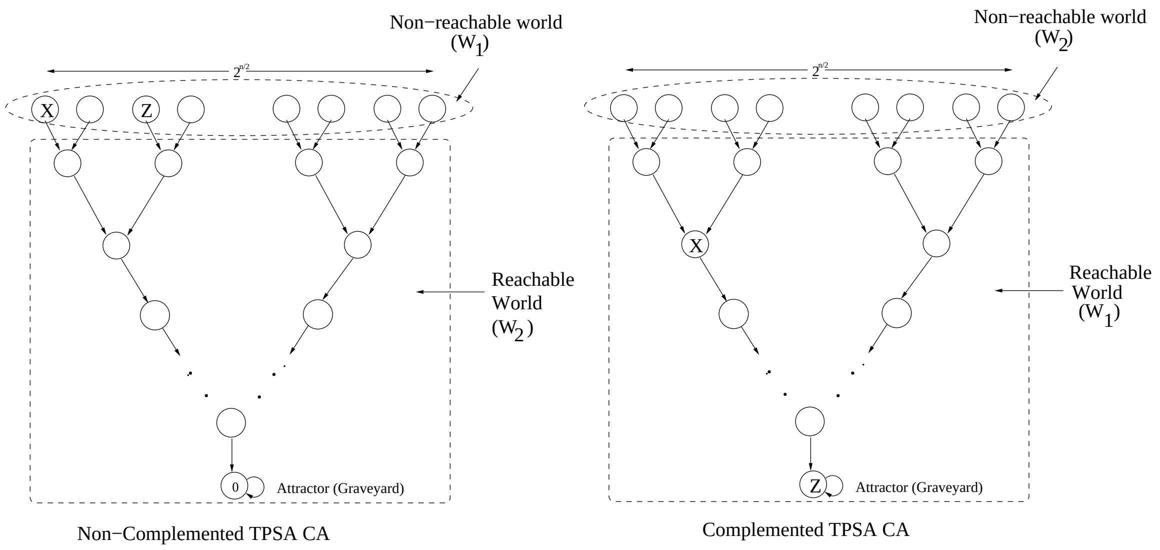


Figure 3. **The exchange of the worlds in a complemented TPSA CA**

TABLE I.
Spectrum of a 4 cell TPSA based regular graph

No. of Union (t)	Graveyards	Degree	First Eigen Value	Second Eigen Value	Spectral Gap (g)	g/t
1	{0},{4}	4	4	3.2361	0.76	0.76
3	{0,15},{4,8}	8	8	4.899	3.10	1.03
5	{0,15,3},{4,8,10}	12	12	6.3440	5.66	1.14
7	{0,15,3,2},{4,8,10,9}	16	16	5.2263	10.77	1.54

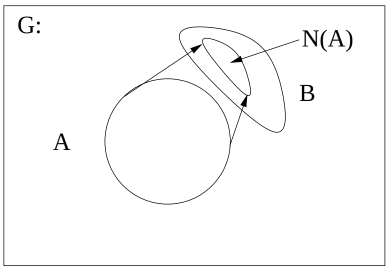


Figure 4. **Computing the expansion of the random d regular graph**

we first consider a single matching, αn vertices can have maximum αn neighbours in $N(A)$. The probability that the neighbours of A are in B is :

$$\begin{aligned} p &= \frac{\text{no. of ways in which N(A) can lie inside B}}{\text{no. of ways in which N(A) may be chosen outside A}} \\ &= \frac{\binom{\beta n}{\alpha n}}{\binom{n-\alpha n}{\alpha n}} \end{aligned}$$

Hence, if we consider a d-matching, assuming all the edges to be independent we have

$$\begin{aligned} \Pr[N(A) \subset B] &= \left(\frac{\binom{\beta n}{\alpha n}}{\binom{n-\alpha n}{\alpha n}} \right)^d \\ &\approx \left(\frac{\beta}{1-\alpha}\right)^{\alpha n d} \end{aligned}$$

Thus, we have the following probability:

$$\begin{aligned} &\Pr[\exists A \in G \text{ s.t } |A| = \alpha n, |N(A)| \le \beta n] \\ &\le \sum_{|A|=\alpha n} \sum_{|B|=\beta n} \left(\frac{\beta}{1-\alpha}\right)^{\alpha n d} \\ &\le \binom{n}{\alpha n}\binom{n}{\beta n}\left(\frac{\beta}{1-\alpha}\right)^{\alpha n d} \end{aligned}$$

Next, we use the following approximations:
$\binom{n}{\alpha n} = 2^{nH(\alpha)}$, $\binom{n}{\beta n} = 2^{nH(\beta)}$.

Here, $H(\alpha) = -\alpha \log_2 \alpha - (1-\alpha)\log_2(1-\alpha)$. Thus, setting $\alpha = \frac{1}{m}$ and $\beta = \frac{1}{2}$ and some simplifications we have:

$$\begin{aligned} &\Pr[\exists A \in G \text{ s.t } |A| = \alpha n, |N(A)| \le \beta n] \\ &\le 2^{n[\log_2 m-(1-\frac{1}{m})\log_2(m-1)+1+\frac{d}{m}\log\frac{m}{2(m-1)}]} \approx 2^{-cn} \end{aligned}$$

Hence in order to make the probability negligible we may choose parameters m and d such that c becomes positive.

The value of d gives us an estimate of the number of union operations t to be performed.

C. Mathematical Formulation of adjacency in the TPSA based Graph

The graph $G(V,E)$ can thus be described as a union operation between the graphs G_1 and G_2 where $Z_1 \in W_1$ and $Z_2 \in W_2$ are the respective graveyard states. The following algorithm computes the four neighbours of a given state in the graph G.

Algorithm 1: **Computing neighbourhood of a vertex X in G**

Input: Z_1, Z_2, a state $X \in G$. Any state X is an n-tuple, $(x_1, x_2, \ldots, x_n)$

Output: The four neighbours of X (U, V, W, Y)

```
Step 1: Computation of neighbours $U$ and $V$,
    $U = T_n(X) \oplus (I_n \oplus T_n)Z_1$
    $V = T_n(X) \oplus (I_n \oplus T_n)Z_2$
Computation of neighbours $W$ and $Y$
Step 2:
Compute, $X \oplus (I_n \oplus T_n)Z_1 = (\, x_1, x_2, \ldots, x_n \,)^T$
        If$(x_1 = x_2)\{/ * X \in G_1 * /$
        from the matrix defined in theorem 1
         set $w_n = 0$, Compute $w_{n-1} = x_n$,
        $w_{n-2} = x_{n-1}, \ldots, w_2 = x_3, w_1 = w_2 \oplus x_1$
Step 3:  set $y_n = 1$, Compute $y_{n-1} = x_n$,
        $y_{n-2} = x_{n-1}, \ldots, y_2 = x_3, y_1 = y_2 \oplus x_1\}$

        else$\{/ * X \in G_2 * /$
        repeat Step 2 with $Z_2$ replacing $Z_1$
        $\}$
```

It is evident that the distribution of the neighbourhood is identical to that of Z_1 or Z_2 which are randomly chosen. Thus, we have a four regular pseudo-random graph when we have only one union operation. The degree may be increased with the number of union operations. The advantage lies in the fact that in general the description of a random graph grows exponentially with the number of vertices. However, using the TPSA based construction one may generate graphs which exhibit randomness and also may be described using polynomial space, as we require to store only the graveyard state. For an N cell TPSA, in order to compute the neighbourhood of any state, Algorithm 1 is applied in constant time, as each of the steps $1, 2$ and 3 may be applied in constant time parallely on the N bit input vector. Herein lies the efficacy of the TPSA based expander graphs.

IV. Application of TPSA based Expander Graphs in Constructing One-way functions

The one-way function using the d-regular graph generated by the TPSA is based on the construction proposed in [1]. The one-way function maps a string in $\{0,1\}^n$ to another in $\{0,1\}^n$. The algorithm is based on a d regular graph generated by the TPSA. As already mentioned the graveyard states are from W_1 and W_2. Let Z_1 denotes the graveyard states from W_1 and Z_2 denotes the graveyard states from W_2. If there are i elements in Z_1 and Z_2 then the number of union operations required to generate the d regular graph is $2i-1$ and so we have $d = 2(2i-1+1)$. Thus we arrive at the number of graveyard states required as $i = \frac{d}{4}$.

The evaluation of the one-way function is as follows:

Algorithm 2: **One-way function (f) using the state transitions of a TPSA**

Input: $Z_1 \in W_1$, $Z_2 \in W_2$, a state $X \in \{0,1\}^n$. Thus the TPSA based regular graph has degree 4.

Output: The one-way output $Y \in \{0,1\}^n$ such that $Y = f(X)$

Step 1 Consider an N cell TPSA, where $N = \log_2(n)$. Generate a collection C, which constitutes of the neighbours of each node in the regular graph generated by the state transitions of the TPSA using **Algorithm 1**. Mathematically, $C = \{S_i, \text{ if } v \in S_i, E(v,i) = 1, i \in \{1, \ldots, n\}\}$.

Step 2 For $i = 1$ to n, project X onto each of the subsets S_i. If $S_i = \{i_1, i_2, \ldots, i_d\}$, then the projection of X on S_i denoted by X_{S_i}, is a string of length d indicated by $\{X_{i_1}, X_{i_2}, \ldots, X_{i_d}\}$.

Step 3 Evaluates a non-linear boolean function Π on each of the n projections thus giving an n bit output, $\{\Pi(X_{S_1}), \Pi(X_{S_2}), \ldots, \Pi(X_{S_n})\}$. The non-linear function Π is $\Pi(z_1, z_2, \ldots, z_d) = (\sum_{i=1}^{d/2} z_i z_{i+d/2}) \bmod 2$.

The forward transformation is very efficient as the total time required is $O(n)$. This may be observed as the time required to perform **Step 1** is due to that required to apply Algorithm 1 at all the 2^N vertices. Hence the total time is also proportional to $2^N = n$. The time required to apply **Step 2** and **Step 3** is also proportional with n and hence the total time required. However computing the inverse seems to be intractable even when the collection C is known. As proved in [1] the complexity of a proposed inverting algorithm is atleast exponential in $min_\pi\{max_i\{| \cup_{j=1}^{i} S_{\pi(j)}| - i\}\}$.

We have shown in **section III-B** that the probability that the size of the neighbourhood of any subset S is proportional to n is very high. Thus for all cases the value of $min_\pi\{max_i\{| \cup_{j=1}^{i} S_{\pi(j)}| - i\}\}$ is $O(n)$ and hence the complexity of a possible inverting algorithm proposed in is atleast exponential in $O(n)$ [1]. Thus the problem of inverting the one-way function seems to be intractable.

In the following section we present the implementation of the proposed architecture on a Xilinx XCV-1000 platform.

V. Implementation of the TPSA based One-way Function

In this section, we propose the architecture for n = 128 bit one-way function. The size of the TPSA CA is thus $N = log_2 n = log_2 128 = 7$. Using the 7-bit TPSA CA, we construct a 16-regular random expander graph. It has been computed that with these parameters, the expander graph has good expansion properties which in turn make the one-way function strong [1].

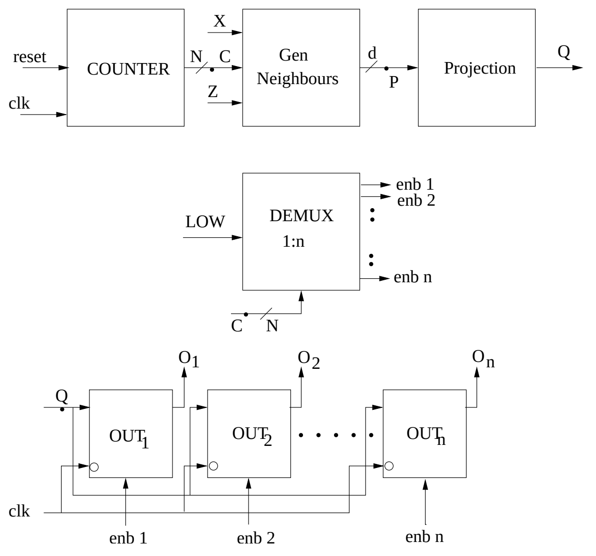


Figure 5. **Top-level view of the architecture**

Fig. 5 shows the top-level view of the architecture. The $N = 7$ bit up-counter counts from 0 to 127 ($n = 128$) and for each of these values the block Gen-Neighbours calculates the $d = 16$ neighbours in the TPSA based d-regular graph. Each neighbour selects a particular bit from the input X. The 16 bit output of the block Gen-Neighbours is thus the 16 bits selected from the input X. These 16 bits are taken by the block Projection which calculates the Boolean function Π on those bits. Finally the output bit of the Projection block (Q) is clocked into the proper flip-flop of the output register (OUT_1 to OUT_n). The output of the counter (c) determines which of the flip-flops (OUT_1 to OUT_n) in the output register is to be enabled (enb_1 to enb_n) and consequently the output of the Projection block (Q) gets clocked into that enabled flip-flop. The Z input to Gen-Neighbours is the set of the N-bit graveyards of all those TPSA CA whose union is to be carried out.

A. Details of the proposed architecture

The block Gen-Neighbours (**Fig 6**) is based on *Algorithm-1*. It can be seen that it has $i = 16/4 = 4$ pairs of the two types of graveyard states (Z_1 and Z_2). Each of these pairs generates 4 neighbours, each of which is then used to select a bit from the input X (P_j), where j varies from 1 to $4i = 16$. Each GN_j block (j varies from 1 to i) is elaborated in **Fig 7**. It has a $MUL - T_n$ unit which computes the multiplication of the matrix T_n with its input Y. The $MUL\_XOR\_I_n\_T_n$ carries out the multiplication of $I_n \oplus T_n$ with its input Z. The first two neighbours (N_1 and N_2) are determined using the first step of Algorithm 1. Thus $N_1 = T_n(X)(I_n \oplus T_n)Z_1$ and $N_2 = T_n(X)(I_n \oplus T_n)Z_2$. The other two neighbours are computed using the rest two steps of the algorithm.

Here $A[N:1] = X \oplus (I_n \oplus T_n)Z_1$ and $B[N:1] = X \oplus (I_n \oplus T_n)Z_2$. Thus if $A[1] = A[2]$ the bits of N_3

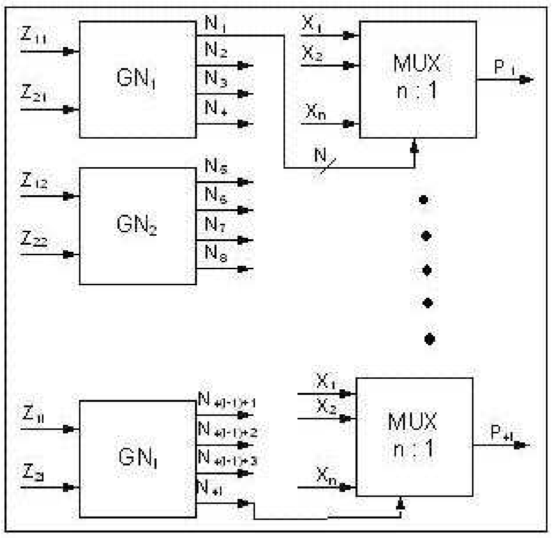


Figure 6. **Architecture of Gen-neighbours**

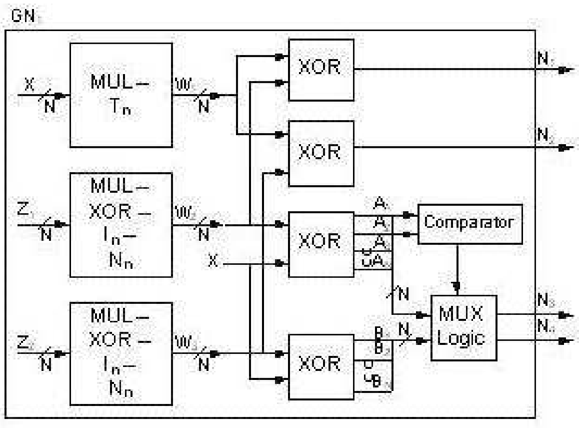


Figure 7. **Architecture of the GN Block**

and N_4 are determined using the A_j values, else they are decided using the B_j values (j varies from 1 to N). The block MUX logic depicts this process in **Fig 7**.

The Projection block computes the non-linear transformation Π on the 16 chosen bits of X. It calculates the XOR of the $d/2 = 8$ products to obtain the result of the transformation Π.

The proposed architecture has been implemented in Verilog HDL and has been verified using RTL simulations using Mentor Graphics ModelSim SE. The Verilog RTL has been prototyped on a Xilinx Virtex XCV1000 FPGA (pkg bg560). The characteristics of the design have been summarized in **table II**.

The resources of the design of the one-way function operating on n bits are as follows:

1) Number of Flip-Flops: n
2) Number of Muxes: d (n : 1 mux) + 1 (N : 1 mux) (Here d is the number of neighbours and

TABLE II.
Performance of the Design on Xilinx XCV-1000 platform

No of Slice Flip Flops	223, 1% of total
No of 4 Input LUTs	1343, 5% of total
No of Slices	6% of total
Maximum delay	0.842 ns
Power Consumption	32mW at $V_{cc} = 3.3V$
Throughput	3.5 Gbps @28.6 MHz

TABLE III.
Comparison of the proposed One-way function with respect to Implementation

	Proposed One-Way Function	**Efficient Implementation of exponentiation in $GF(2^n)$ [16]**
Order of resource usage	$8n$ LUT+ n FF	$15.7n$ LUT + $3n$ FF
Order of delay	n	n^2
Order of area -delay product	n^2	n^3

$N = log_2 n$)
3) Number of And gates: $nd/2$
4) Number of XOR gates: $d/2$ (1 bit xors) + d (N bit xors) + n ($d/2$ bit xors)
5) Counter: 1 N bit counter
6) Demux: 1 ($1 : n$ demux)

The above analysis shows that the resource utilization increases linearly with the input size.

Table III gives a comparison of the resources used, the delay and the area-delay product of the proposed one-way function with those of an efficient implementation of the exponentiation operation in $GF(2^n)$ [16]. In the table, n denotes the number of bits in the input and output.

VI. CONCLUSION

The paper proposes a method to generate expander graphs with good expansion properties based on the state transitions of a special class of Cellular Automata, known as the Two Predecessor Single Attractor CA (TPSA-CA). The expander graph has been finally used to compose a one-way function whose security lies on the combinatorial properties of the expander graphs. The design has been implemented on a Xilinx XCV-1000 platform and the performance of the design has been compared with that of efficient designs of the popular modular exponentiation based one-way function. Results have been furnished to prove that the design is superior in terms of area delay product and power consumption. The one-way function can be used to design key agreement protocols which provide authentication

REFERENCES

[1] Oded Goldreich, "Candidate One-Way Functions Based on Expander Graphs," Cryptology ePrint Archive, Report 2000/063, 2000. [Online]. Available: http://eprint.iacr.org/

[2] W.Diffie and M.Hellman, "New Directions in Cryptography," in *IEEE Transactions on Information Theory (22)*. IEEE, 1976, pp. 644–654.

[3] P. P. Chaudhuri, D. Chowdhury, S. Nandi, and S. Chattopadhyay, *Additive Cellular Automata Theory and its Application*. IEEE Computer Society Press, 1997, vol. 1.

[4] A.Menezes, P. Van. Oorschot and S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996, ch. 12, pp. 489–541.

[5] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*. Springer, 2003, ch. 5, pp. 138–199.

[6] Y. Yacobi and Z. Shmuely, "On key distribution systems," in *In the Proceedings of Advances in Cryptology-Crypto '89, LNCS 435*, December 1989, pp. 344–355.

[7] D. S. Wong and A. H. Chan, "Efficient and mutually authenticated key exchange for low power computing devices," in *In the Proceedings of Advances in Cryptology-Asiacrypt '2001, LNCS 2248*, 2001, pp. 272–289.

[8] C.-S. Park, "On certificate-based security protocols for wireless mobile communication systems," *IEEE Network*, vol. 11, no. 5, pp. 50–55, September/ October 1997.

[9] M. Jakobssen and D. Pointcheval, "Mutually authentication for low power mobile devices," in *Financial Cryptography, LNCS 2339*, 2001, pp. 178–195.

[10] K. M. Gunther Horn and C. J. Mitchell, "Authentication protocols for mobile network environment value-added services," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 2, pp. 383–392, March 2002.

[11] Nati Linial and Avi Wigderson, "Expander graphs and their applications," http://www.math.ias.edu/ boaz/ExpanderCourse/, 2003. [Online]. Available: http://www.math.ias.edu/ boaz/ExpanderCourse/

[12] Michael A. Nielsen, "Introduction to expander graphs," http://www.qinfo.org/people/nielsen/, 2005. [Online]. Available: http://www.qinfo.org/people/nielsen/

[13] A. Lubotzky, R. Phillips, and P. Sarnak, "Ramanujan graphs," *Combinatorica*, vol. 8, no. 3, pp. 261–277, 1988.

[14] G. A. Margulis, "Explicit constructions of expanders," *Problemy Peredači Informacii*, vol. 9, no. 4, pp. 71–80, 1973.

[15] ——, "Explicit group-theoretic constructions of combinatorial schemes and their applications in the construction of expanders and concentrators," *Problemy Peredachi Informatsii*, vol. 24, no. 1, pp. 51–60, 1988.

[16] Guido Bertoni, Jorge Guajardo, Sandeep Kumar, Gerardo Orlando, Christof Paar and Thomas Wollinger, "Efficient $GF(p^m)$ Arithmetic Architectures for Cryptographic Applications," in *CT-RSA 2003, LNCS 2612*, 2003, pp. 158–175.