Int. SPIE Conf. on Electronic Imaging, Feb. 1995, S. Jose, CA

SCAN IMAGE COMPRESSION/ENCRYPTION HARDWARE SYSTEM

N. G. Bourbakis+#, R.Brause^ and C.Alexopoulos*
+SUNY Binghamton Dept. of EE, AAAI Lab Binghamton, NY 13902
#University of Crete, Dept. ECE, Chania 73100, Crete, Greece
*Univ. of Patras Dept. of Computer Engr 26500 Patras,Greece
^Geothe University, Dept. of Informatics, Frankfurt, Germany

Abstract

This paper deals with the hardware design of an image compression/encryption scheme called SCAN. The scheme is based on the principles and ideas reflected by the specification of the SCAN language. SCAN is a fractal based context-free language which accesses sequentially the data of a 2-D array, by describing and generating a wide range (near (nxn)!) of space filling curves (or SCAN patterns) from a short set of simple ones. The SCAN method uses the algorithmic description of each 2-D image as SCAN patterns combinations for the compression and encryption of the image data. Note that each SCAN letter or word accesses the image data with a different order (or sequence), thus the application of a variety of SCAN words associated with the compression scheme will produce various compressed version of the same image. The compressed versions are compared in memory size and the best of them with the smallest size in bits could be used for the image compression/encryption. Note that the encryption of the image data is a result of the great number of possible space filling curves which could be generated by SCAN. Since the software implementation of the SCAN compression/encryption scheme requires some time, the hardware design and implementation of the SCAN scheme is necessary in order to reduce the image compression/encryption time to a real-time one.
The development of such an image compression encryption system will have a significant impact on the transmission and storage of images. It will be applicable in multimedia and transmission of images through communication lines.

KEYWORDS: SCAN Language; Image Compression; Image Decomposition; Scanning Algorithms; Image Encryption

1. Introduction

The use of data compression on huge amount of data, which has to be transmitted or stored, becomes extremely important for a variety of applications, such as multimedia, facsimile systems, communication, etc, [1-9]. Several techniques have been developed to compress
sets of data either with or without loss of the information contained in these sets of data. Some of the techniques are used for image compression and are either coding, or interpixel, or phychovisual etc. [1,4,7,9]. More specifically, one particular area in which the researchers have to deal with huge amount of data is the digital image processing area. Note that, only a single

grey level image of 1024x1024 pixels contains 1M bytes. Moreover, in many cases most of the images contain a great amounts of useless information, such as large homogeneous areas (sea, sky,etc), thus a reduction of these type of information will not affect the information included in a particular image. In addition, this compression scheme will speed up the transmission of the image data, or it will occupy less storage area.
Encryption is also another type of data process which is used to protect valuable data from undesirable readers [6,10,11,12]. Encryption becomes a very important process when data have to be stored or transmitted through a communication channel, where there a dangerous of undesirable accessibility is significant large. Digital images are frequently transmitted through communication channels, because of the increased demands from applications, such as geographic information data (images transmitted from satellites), telephone communication through channels (voice and image), distributed multimedia (communication through computer networks, where images and voice are transmitted), etc. The 2-D image data is usually scanned by a raster scan pattern and the generated 1-D sequence of data can be encrypted in blocks by using a block cipher, such as DES, or a stream cipher, namely a key generator, or a product cipher [11]. Thus, what about to encrypt the image data directly from the 2-D form.

In this paper an image compression-encryption hardware scheme is proposed by using the words (patterns, or orders) produced by an image processing language called SCAN [5,7]. SCAN is a context-free language which accesses sequentially the elements of a 2-D array (image) by describing and generating a wide range of accessing algorithms (permutations) from a short set of simple ones. The proposed methodology can compress and encrypt both binary and grey level images. More specifically, in binary images the SCAN compression scheme scans the 2-D array of image data by using the appropriate combination of algorithms (SCAN patterns) in order to generate the most closed "permutation" which will scan the image pixels with the same values, as continuous sequence, or as the smallest number of continuous sequences. No lost of the image information is happened by using this SCAN compression scheme. In case of grey level images, each 2-D image is segmented into k 2-D binary images, where k represents the number of bit/pixel. Thus, the SCAN method is applied to each binary image, like the binary case, and the final compression is the accumulation of the binary compressed schemes.
The encryption of both binary or grey level images occurs automatically, since the selected permutations, for the scanning of the particular binary image, defines the way that the image data will be encrypted. Of course the selected encryption pattern (permutation) probably will not be the most complex one generated by SCAN. However, with the SCAN compression effort, an encryption is obtained automatically.

2. The SCAN Language[5]

In order all the elements of an image to be accessed sequentially only once (no double accessing of any element) an appropriate algorithm, or scanning or space filling curve method is needed.

Definition: A scanning (or accessing, or space filling curve) method Sn, $n \in Z+$ is defined as a sequential collection of the 2^k elements of Pnn, $0 \leq k \leq \log(n)$, by following a defined or formulated order Sn = {[P(i,j) ... P(r,q)] : $1 \leq i,j,r,q \leq n$, without a repetition of the same element twice or more in the pattern, and the scan order is defined by an algorithm}

Notation: The total number of all scanning patterns (or permutations) of an image of nxn elements is **(nxn)!**

SCAN is a context-free language which accesses sequentially the elements of a 2-D array (image) by describing and generating a wide range of accessing algorithms (permutations) from a short set of simple ones.
There are two important concepts which characterize the SCAN methodology:
(i) The concept of hierarchical decomposition of an image into square subregions
(ii)The concept of recursive composition of two or more basic accessing techniques (algorithms) in such a way where the composed pattern can expand uniformly all over the image representation levels.
Thus, SCAN words convey information about both the decomposition to be applied to and the type of scanning techniques to be composed with.

Definition: A SCAN word wt ϵ L(G) generated by the grammar G is defined as

wt = La(1)2^I1 # La(2)2^I2 # ... # La(t)2^It

where t represents the length of wt with t ϵ Z+, La(i) ϵ Σ, Ii ϵ Z+, and (2^I1)(2^I2)...(2^It)=n

2.1. A Specific SCAN Language

The development of a specific SCAN language provides an efficient approach to the problem of modeling and generating all the 2-D accessing algorithmic patterns of an image of nxn elements.

A specific application oriented alphabet has been developed [5], which consists of 15 SCAN "letters":

Σ = {R,C,D,E,A,I,O,L,H,S,Y,W,Z,B,X}

Note that each "letter" (or symbol) of the alphabet represents a scan order (or algorithm) which can scan the elements of a 2-D image. Figure 1 shows the SCAN alphabet and the generation of the SCAN words.

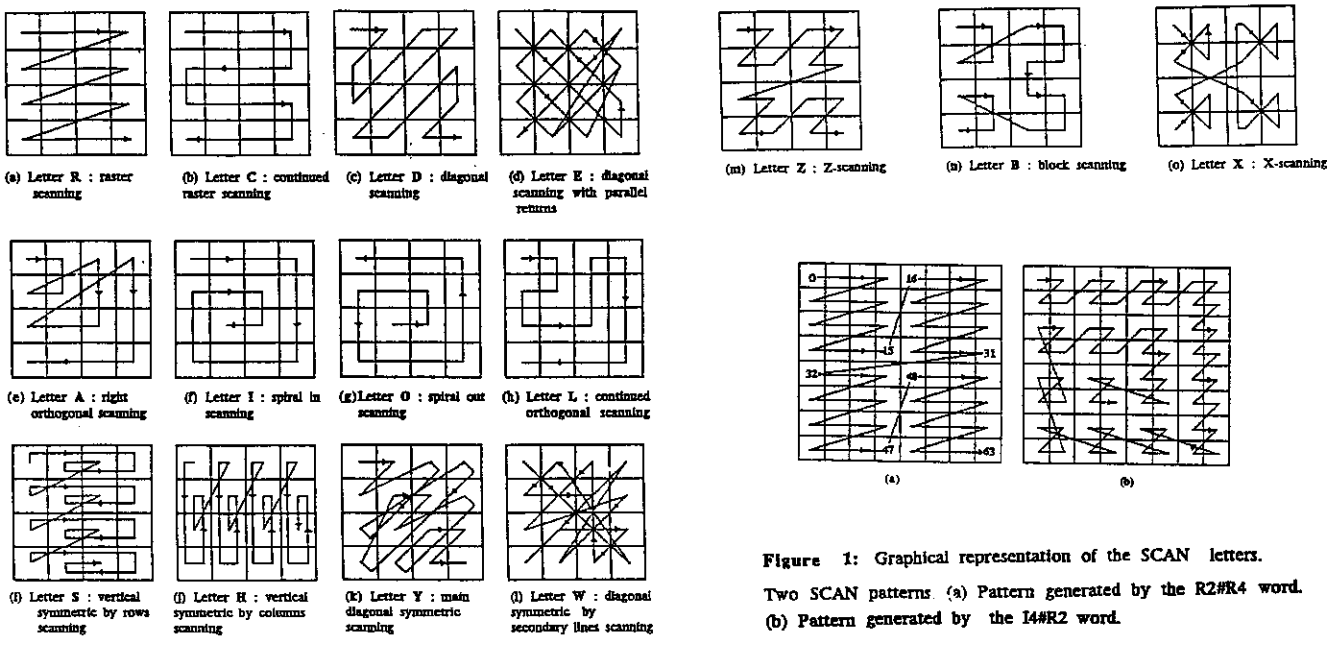

Figure 1: Graphical representation of the SCAN letters.
Two SCAN patterns (a) Pattern generated by the R2#R4 word.
(b) Pattern generated by the I4#R2 word.

3. The SCAN Compression/Encryption Scheme

3.1. Compression

. Binary Images

For binary images the SCAN compression scheme is based on the optimum selection of the SCAN pattern, which is able to produce the minimum amount of bits by representing the original image in the most compressed form without loss of information.

Definition: The optimum compression scheme (CS(wt)) is defined as the SCAN word wt, which "scans" the examined image Pnn and produces the minimum number (NM) of pairs {(P'(i),P'(j)}, where P'(i) represents the sequential numbering of the scan pattern (wt) at the first pixel with a specific value (v) and P'(j) represents the sequential numbering of the pattern (wt) at the last pixel with the same value (v).

$$CS(wt\ optimum) = wt + NM * \{P(i),P(j)\}$$

For the successful selection of the optimum compression scheme, a set of SCAN words applied on the original image Pnn by producing their own compression schemes (CS(wt)). The compression schemes are compared and the optimum is that one which produces the minimum number of bits for the safe compression of the image.

An illustrative example is given below to explain the way that the SCAN compression scheme works. Let consider a binary image of 8x8 pixels.

```
11110000
11110000
11110110
11110000
00001111
00001111
00001111
00001111
```

Figure 2: A binary image of 8x8 pixels

If the SCAN word wt = R is applied, then the produced compression scheme is CS(R) = R+[(1,4),(9,12),(17,20),(22,23),(25,28),(37,40),(45,48),(53,56),(61,64)] = 152 bits

If another SCAN word wt = Z2#O4 is applied, then the produced compression scheme is CS(Z2#O4) = Z2#O4 + [(1,18),(49,64)] = 48 bits. If however the SCAN word wt = X2#O4 is applied, then the produced compression scheme CS(X2#O4) = X2#O4 + [(1,34)] = 32 bits

Thus, the selected compression scheme is CS(X2#O4).

The Compression Algorithm

Begin: Input the image data and the number of SCAN letters M=15;
For i=1, to M, with step = 1 do
Select the SCAN letter w(i) e Σ
Start the SCAN length-run compression process of the 2-D image data by using the letter w(i);
Save the image compression scheme CS[w(i)];
End;
Compare the schemes CS[w(i)] and select that one, which produces the minimum number of bits (CS[w(k)op]),
1≤k≤M;

Use the SCAN letter w(k), which produces (CS[w(k)op]), as a "drive" letter for the generation of the SCAN word
wt(j) = w(k)#w(i), with w(i) ϵ Σ;
Keep w(k) constant and start a new SCAN length-run compression on the original image by using all the SCAN letters w(i);
For i=1, to M, with step = 1 do
Select w(i);
SCAN Length-run compression using wt(j);
Save CS[wt(j)];
End;
Compare the compression schemes CS[w(i)], and select that one that produces the minimum number of bits CS[wt(j)op];
If the new CS[wt(j)op] ≤ CS[w(k)op] then
use the SCAN word wt(j) as a "drive" word (t=2)
for the generation of a SCAN word (t=3) and repeat the same length-run process as previously described;
else, select the second "best" compression scheme
CS[w(k')] produced by the letter w(k') and
repeat the length-run process;
The whole process ends when no "better" optimum compression scheme is generated;
Stop:

. Grey Level Images

. Lossy

The SCAN scheme compresses grey level image data after the application of a fuzzy smoothing and segmentation on these data. In particular, the scheme for grey level image is slightly different from the one used for binary images. It uses some image processing techniques for the appropriate preparation of the image before the actual compression, which similar to binary one. More specifically, a histogram process takes place first, then an image segmentation is used to define the regions with different greyness. At that point, a new histogram process takes place to define the different level of greyness. Then the grey level value is used for the image is analyzed into a number of k bits, k = b1b2b3...bk. Each bi, ie[1,k] belongs to a 2-D binary matrix. Thus, the original image is converted into k binary ones, see figure 3, thus the binary compression scheme is used on each binary matrix and the accumulated compression results represent the grey image compression output:

$$CS(\text{wt optimum}) = k * [wt + NM * \{P(i),P(j)\}]$$

. Lossless

The SCAN compression scheme is also a lossless one. In this particular case, each image is segmented into a set of L binary images, according to the number of L bits per pixel, see figure 4. Then the binary compression scheme is applied on each image, and the accumulated results represent the final compressed image output.

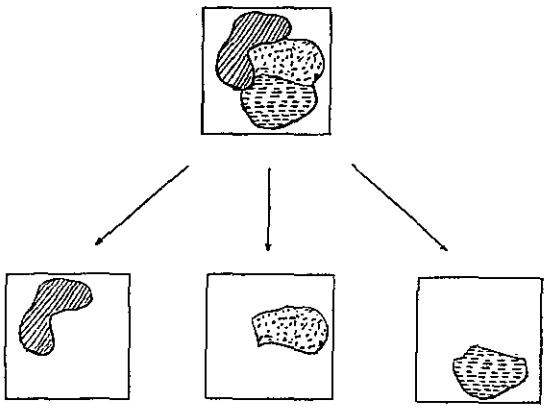

Figure 3

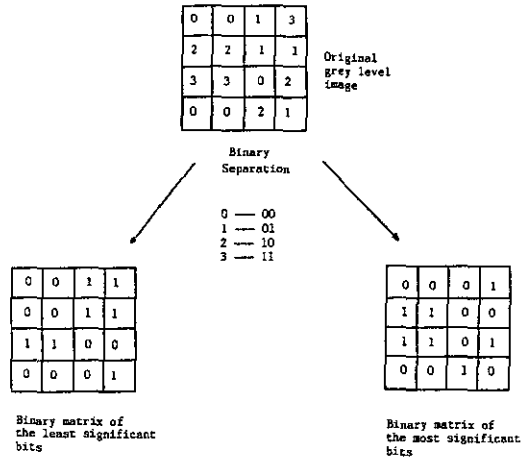

Figure 4

3.2. Encryption

In this section the description of the SCAN encryption methodology is presented by using scan patterns.

3.2.1. SCAN Fractals as Transposition Cipher[6]

The SCAN language provides a family of transformations from 2-D to 1-D representations. Each SCAN pattern defines a transposition of the image data into a 1-D representation. Thus, the family of the SCAN patterns could be considered as a transposition cipher, figure 5. The scan word (or permutation) wt defines the encryption and decryption key. The set of the regular patterns (or permutations) S' generated by the current version of the SCAN language is:

$$S'(n) = \sum_{t=1}^{\log n} \left[\frac{(\log n - 1)!}{(\log n - t)! * (t-1)!} + (4097)^{t-1} - 2^{t-1} \right] st$$

where, st is the number of distinct patterns generated by the t-letters words.

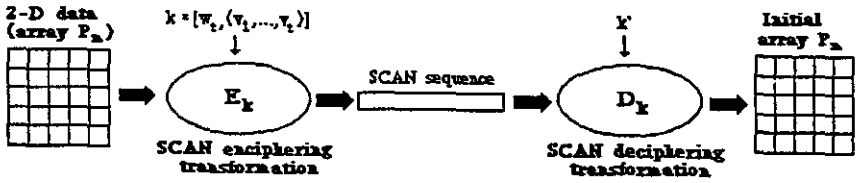

Figure 5: The SCAN cryptographic scheme.

3.2.2. Illustrative Example

Figure 6 shows an illustrative example obtained by the SCAN cryptographic system by using an 16x16 binary image.

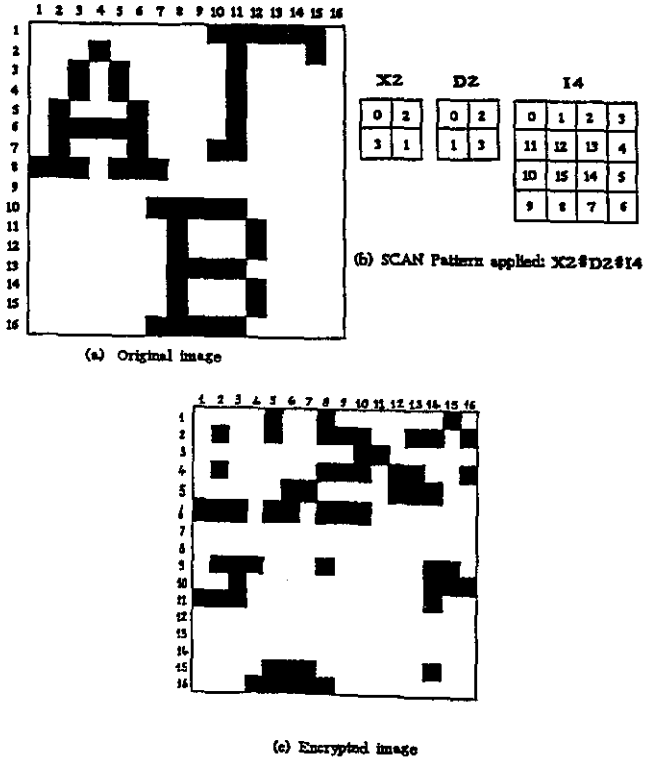

Figure 6 : The SCAN transposition cipher applied on a binary image.

4. Hardware Design of the SCAN Compressor/Encryptor

The following figure 7 presents the global configuration of the SCAN compressor/encryptor. More specifically, the original image is stored in the grey level memory. Then the ASIC preprocesses it and generates either the segments of the regions with the same grey level value, or the binary matrices based on the number of bits/pixel. The generated binary matrices are stored into the appropriate binary memories. From there, the processor defines the scan patterns which are implemented by the k decoders and which access (scan) the binary memories by extracting the scan patterns. The scan patterns fit into the SCAN compressor, which generates the compressed form of the regions with the same grey level values. The compressed forms are saved into the buffers R (recent) and B (best). When new scan compressed forms are generated, then the recent and the best are compared in the comparator, and the best one (less bits) replaces the previous best in the buffers. Finally, the composition of all the best compressed forms produces the compressed form of the input image.

4.1. SCAN Decoders

In this subsection we present the hardware design of the system's decoders, especially, the raster and z ones, see figures 8,9.

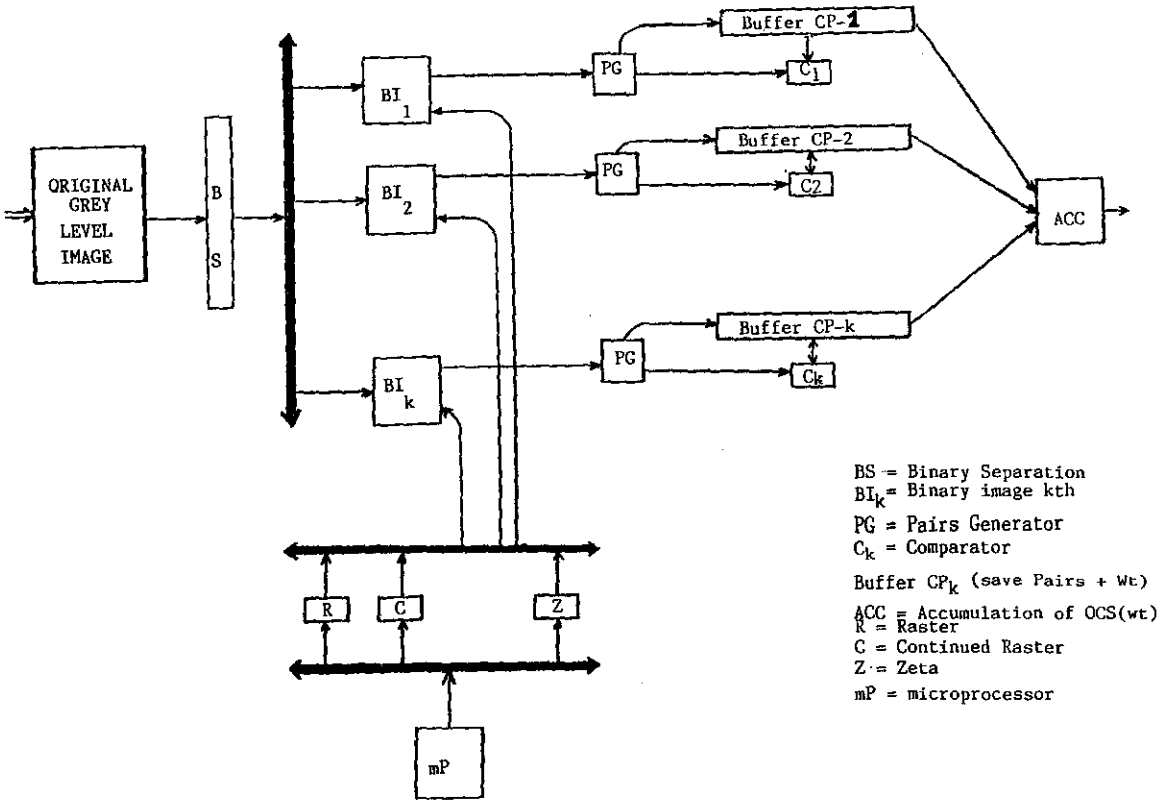

Figure 7

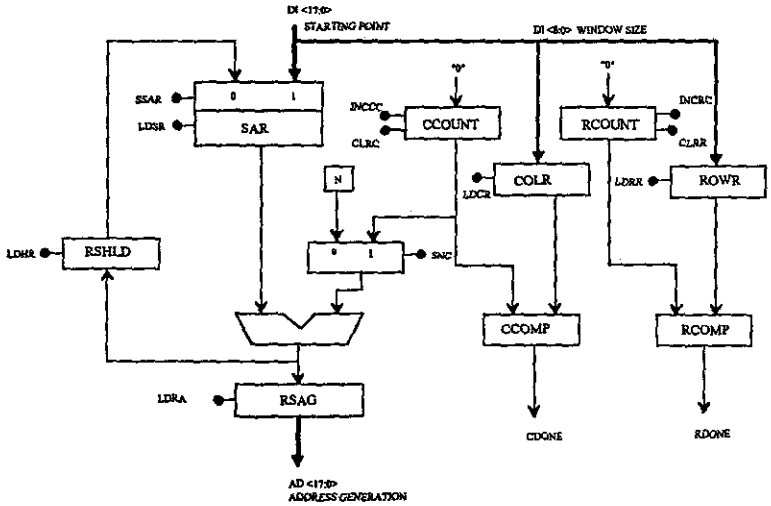

FIG. 8 : BLOCK DIAGRAM OF RASTER SCAN DECODER (DATA SECTION).

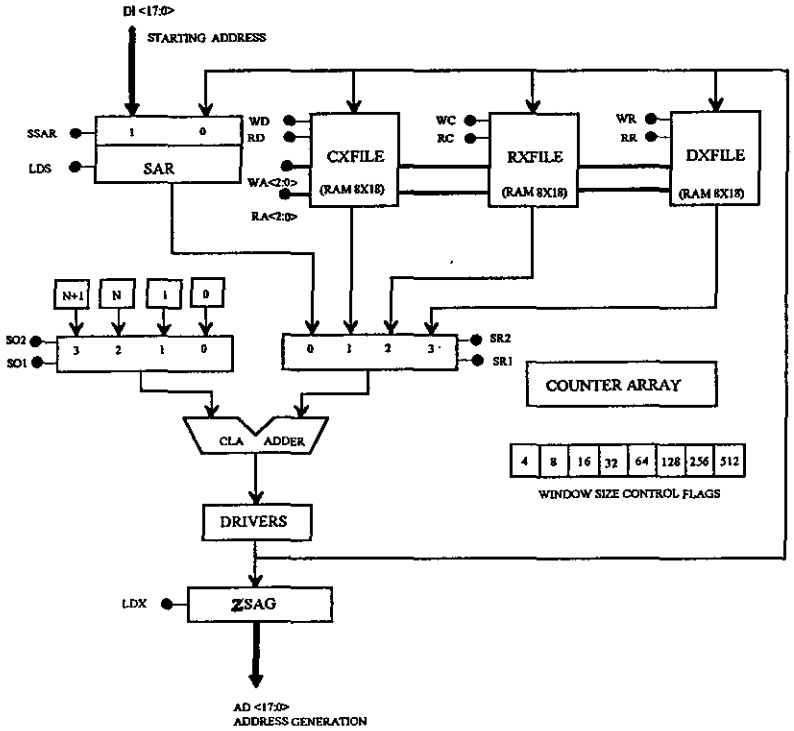

FIG. 9 BLOCK DIAGRAM FOR Z SCAN DECODER (DATA SECTION)..

5. CONCLUSIONS

In this paper the hardware system which implements the SCAN image compression encryption scheme was presented. The compression encryption scheme was based on an image context-free language called SCAN. Especially it was based on the generated patterns which were applied on the original image by reducing its size and encrypting its content. The compression methodology proposed here presents a good compression rate for binary images, since it provides a lossless result. It also gives a reasonable good compression ratio for grey level images with the option lossy or lossless (X-ray images). Another important feature of this SCAN methodology is that it can encrypt the compressed image. The combination of this two useful features make SCAN methodology attractive to some applications areas, such multimedia and medical imaging. The main disadvantage of this approach is the high computation time needed for the generation of the appropriate SCAN patterns to compress an image. The solution of this problem was the design of a hardware system in order to reduce significantly the computation time. The next step of this work is the actual evaluation and implementation of the hardware system and a comparison of it with other compression systems.

References

[1] R.Gonzalez and R.Woods "Digital Image Processing",Addison Wesley 1992
[2] A.N.Netravali and J.O.Limb,"Picture coding:A review",Proc.IEEE vol.68,No.3,1980, 366-406
[3] J.A.Storer and J.H.Reif (eds),"Proceedings of DDC'91, IEEE Computer Society Prees,Los Alamitos, CA, 1991
[4] M.Rabbani and P.W.Jones,"Digital Image Compression Techniques SPIE Press,vol.TT7,1991
[5] N.Bourbakis, C.Alexopoulos, A.Klinger, "A parallel implementation of the SCAN language", Int.Journal on Computer Languages, 14,4,1989

[6] N.Bourbakis and C.Alexopoulos, "Picture data encryption using SCAN patterns", Int.Journal on Pattern Recognition, vol.25,1992,pp.567-581
[7] M.Rabbani, M.Sezan and A.Tekalp, Image and Video Processing, SPIE Proceedings, Feb.3-4,1993, S.Jose, CA.
[8] N.Sorek and Y.Zeevi, On-line visual data compression along a 1-D scan",SPIE,vol.1001, pp.764-770,1988
[9] N.Bourbakis,"Image data compression techniques: a survey of generations", Technical Report, 1993,144 pages
[10] Federal information processing standards publication 46,"Specifications for the data encryption standard",Jan.1977
[11] D.E.Denning, "Cryptography and data security",Reading, Addison Wesley, Inc. 1982.
[12] L.Blum, M.Blum and M.Shub,"A simple unpredictable pseudorandom generator",SIAM,J. Comput., 15,2,364-383,May 1986
[13] J.Bartholdi,"Heuristics based on spacefiling curves for combinatorial problems in the plane" 1985
[14] Fractals in the classroom, vol.I&II, 1992, Verlang