SMART DIGITAL CIRCUIT SIMULATOR

[1]Dr.Mahesh, [2]Jayashree M S, [3]Abhilasha H
[1]Assistant professor, [2, 3]PG Student
[1, 2, 3] Department of ECE
[1, 2, 3]P.E.S College of Engineering,
Mandya-571401, INDIA

Abstract: Electronic circuit diagrams are foundation of studies in electronic science. A circuit diagram tells much information, behind any circuit diagram there are plenty of electronic ingredients which perform their specific tasks, today the electronic software tools failed to effectively convert the information automatically from a circuit image diagram to waveform. Hence in the technical institutions, teacher has to write a waveform for every parameter changes in circuit diagram and this process takes time consuming task of analysis. The traditional teaching learning process with the limitation of visual experience makes it challenging for the teachers to convey the ideas effectively. This made to analyze the problem interms of image processing and computer design automation, which can provide better solution. A tool that can consequently generate the waveform out of the captured circuit diagram would be incredibly valuable for them in their teaching. We present in this paper, a prototype for CAD (Computer Aided Design) tool that offers this functionality. The tool called smart digital circuit's simulator is an attempt to design and develop for on-the dias dynamic simulation of the electronic circuits specifically. Applications of image processing in pattern recognition helps in recognizing the parameter of the digital circuit diagram.

***IndexTerms* - dynamic simulation, netlist generation, Segmentation, Template matching.**

I. INTRODUCTION

Every day the world is changing and things are done differently. Advances in computerized innovation/digital technology have opened up many avenues of learning. Technology has made more information accessible to all groups of people. Technology has also influenced the way students are taught and in which they learn. To make the learning a superior encounter, technical institutions needs to adopt and refresh their teaching learning process with the inclusion of smart integrated technologies [1]. In technical institutions, while handling engineering courses like circuits, machines, and hydraulics etc [2], for the sake of understanding the ideas teacher has to write the circuits on the board to analyze its outputs and related parameters with waveforms. Here, for each change in parameter changes the output/waveform and it needs to be redrawn, which leads to difficulty for the teacher in terms of rewriting the waveform for every parameter changes. This eventually results in tedious time consuming task of analysis. To make learning of engineering or science courses powerful with emphasis on analysis and visualization, sophisticated and smart technology/innovations needs to be developed. The proposed work, smart circuit's simulator is an attempt to address the above discussed challenges while concentrating on electronic circuits specifically. It brings in prototype of smart circuits simulator for on the dias dynamic simulation of the circuit diagram captured. Applications of image processing in pattern recognition helps in recognizing the parameter of the digital circuit diagram.

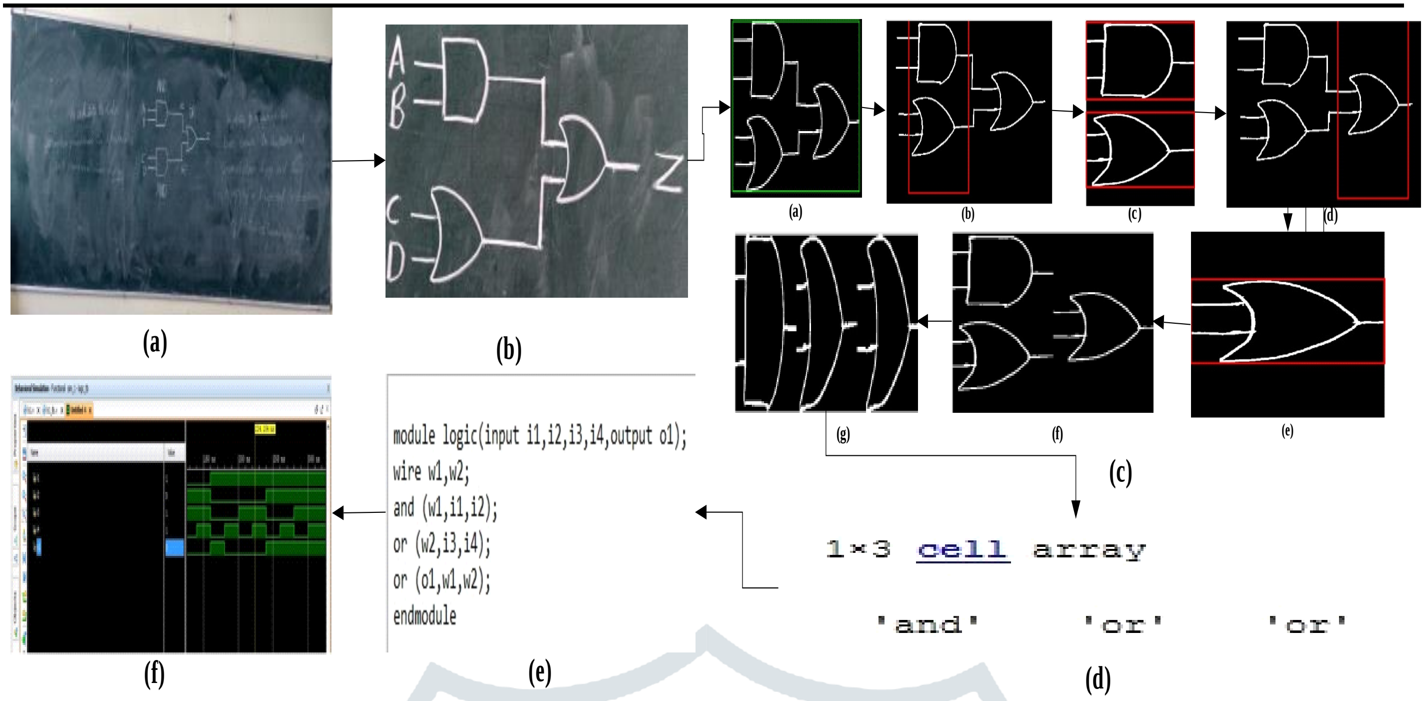
1.1. CAD TOOLS AND TEACHING AID IN ENGINEERING

Currently CAD tools are limited to lab exercises, due to their complex flow/process. Integration of image processing techniques to the CAD tools can makes them suitable for regular teaching aid [3]. CAD tools are used in classrooms to encourage students learning process and make it easier and interesting. There are the best tool for making teaching effective and the best dissemination of knowledge.

II. PROBLEM STATEMENT

Electrical circuit analysis needs identification of circuit elements and relation between parameters and expressions. The relation is obtaining in terms of netlist relating input and output. The same problem can be considered as image process. The circuit image contains information about the elements and their connectivity. The circuit image matrix can be processed to find out a set of line or equation relating the observable and controllable i.e., input and output.

Image preprocessing is the first step in the processing of input image .The input image is a combinational circuit diagram as shown in the fig 1(b),it specifically needs processing on the shapes and links rather than color information so it is necessary to convert rgb image to gray image and then into binary image. The binary image is converted in to noise free image is passed to the segmentation phase to obtain individual circuit elements as shown in the fig 1(c). Once elements are extracted, we have to recognize that element using template matching method, its results are as shown in the fig 1(d) and fig 1(e) shows identify the connection among the elements to generate the netlist and facilitate to find the output response for all possible conditions as shown in the figure 1(f). Figure 11 shows flow chart of proposed work.



```
module logic(input i1,i2,i3,i4,output o1);
wire w1,w2;
and (w1,i1,i2);
or (w2,i3,i4);
or (o1,w1,w2);
endmodule
```

```
1×3 cell array

    'and'    'or'    'or'
```

Figure 1: Illustration of the proposed work for digital circuits. (a) Circuit image on the board. (b) Captured image. (c) Segmentation (d) Circuit elements and their parameter recognition. (e) Netlist generation. (f) Output response.

The current problem can also be defined in the mathematical model

$$\{NE, NL, p_1, p_2 \ldots\ldots p_n, n\} = f(I) \quad (1)$$

$$\{netlist, values\} = NLG\{NE, NL, p_1, p_2 \ldots\ldots p_n, , n, I\} \quad (2)$$

$$Response = netlist\}_{Pn} \quad (3)$$

Where, I = captured image, NE= Number of elements, NL= Number of lines. p1,p2......pn = parameters of the captured circuit diagram, n=nodes of interest

It is being applied over input image I to recognize the parameters and extract the connectivity information as illustrated in the equation 1.Using these information to generate the netlist and find its output response as illustrated in the equation 2 and 3 respectively.

III. WORK FLOW

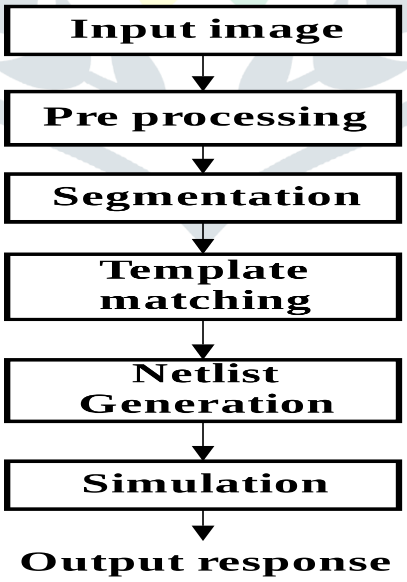


Figure 2: Block diagram of proposed work flow.

Problem can be solved by developing set of matrix processing algorithms. In essence identification of circuit elements is conceived as shapes recognition and template matching. The task of image processing for circuit analysis is implemented as a chain of image processing algorithms.

As it is discuss in the last Paragraph, to generate the netlist from given circuit diagram, the circuit diagram has to be split up in to different parts and elements must be identified out of that and also relation between the element must be identified, so these whole process is split up in to number of different stages as shown in the block diagram of proposed work in the figure 2.

1.1. PRE-PROCESSING

Preprocess the input image is in the equation (4) as shown in the figure 4(a), that it can eliminate the distortions or enhances some image object. In order to achieve this, the image is subjected to various preprocessing techniques [4] as shown in the fig 3. It is specifically needs processing on the shapes rather than color information is illustrated in the equation (6), so it is necessary to convert rgb image to gray image as shown in the figure 4(b) and then converted into binary image as shown in the figure 4(c), which applies thresholding on to the gray scale image with that the pixels having the value greater than the thresholding value is marked as zero and values greater than the thresholding value is marked as one as shown in the equation (7). The binary image is converted in to noise free image by removing small components below 30 pixels is illustrated in equation (8), which is ultimately affect on element wise segmentation as shown in the figure 4(d).Hence we get image which is suitable for further processing.

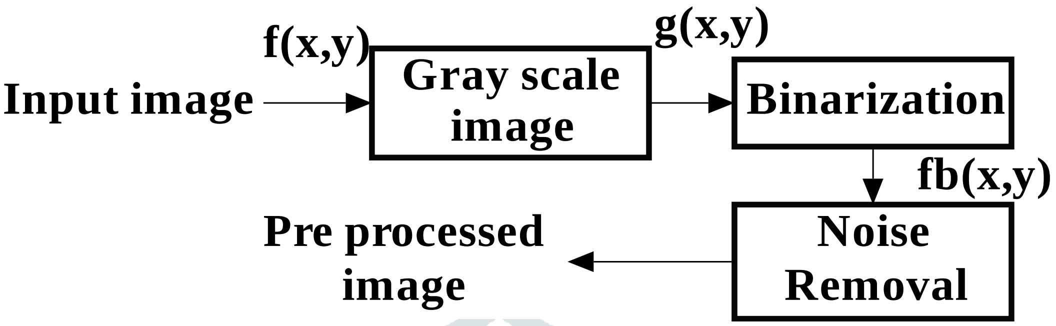
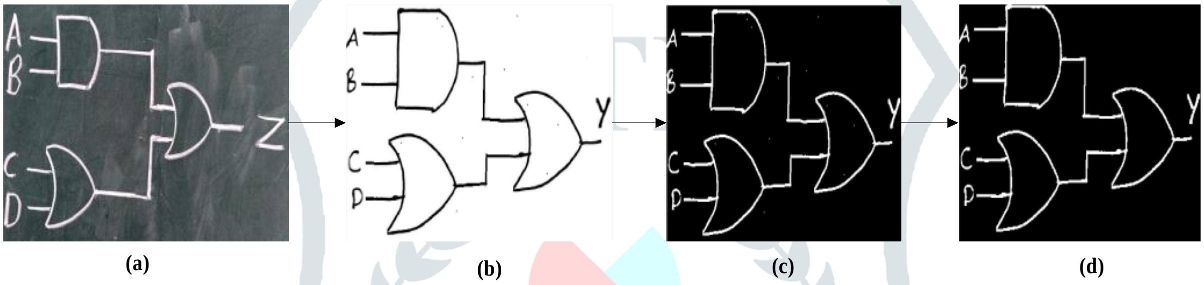


Figure 3: Preprocessing steps.



(a) (b) (c) (d)

Figure 4: Preprocessed image (a) Input image (b) Gray scale image (c) Binarize image. (d) Removal of small components.

$$O(x, y)=T[I(x, y)] \qquad (4)$$

$$(r, g, b)=O\ (x, y) \qquad (5)$$

$$g(x, y)= f(r, g, b) \qquad (6)$$

$$Ib = \mathrm{fb}(x,y) = \left\{\begin{matrix} 0 & if\ g(x,y) \geq T \\ 1 & otherwise \end{matrix}\right\} \qquad (7)$$

$$Inr = \left\{\begin{matrix} 1 & if\ small\ component\ in\ the\ fb(x,y) < 30\ pixel \\ 0 & otherwise \end{matrix}\right\} \qquad (8)$$

Where, I(x, y) is the original(input) image, O (x, y) is the enhanced image and T describes the transformation between the two images. g (x, y)=represents gray scale image pixel at (x, y), fb(x, y)=represents threshold image pixel at (x, y), Ib=represents binarized image. T=Threshold Value, Inr=represents removal of small components image.

1.2. SEGMENTATION

The preprocessed image is ready to be processed further for identifying the elements. To identifying the elements, we have to identify the stages. It is like a tree network, in order to trace the tree network, we have to identify the branches. For that we have to segmenting the elements of given image, this is need because template matching is applied for the isolated object [5]. This is done by, connected circuit are given a label and a rectangular box is plotted around circuit as shown in the figure 5(a) and getting the four coordinates of the circuit as illustrated in the equation (9). Varying this coordinates to crop the elements stage by stage as shown in the figure 5(b) & (d) is illustrated in the equation (9) and stored in an array as illustrated in the equation (10). In each stage as shown in figure 5(c) & 5(e), connected elements are given a label to identify the number of elements as shown in the figure 5(g). Connected elements in an array as shown in the figure 5(f) are given a label and a rectangular box is plotted around each connected element. Each rectangular box is cropped to get the isolated element as shown in the figure 6 and equation (11).Resized the cropped image to 420 x 240 pixels is illustrated in the equation (12) to better facilitate template matching as the templates stored in the library are also in the size of 420 x 240 pixels as shown in the figure 5(h).

$$[x\ y\ w\ l]=\text{BoundindBox (Inr)} \qquad (9)$$

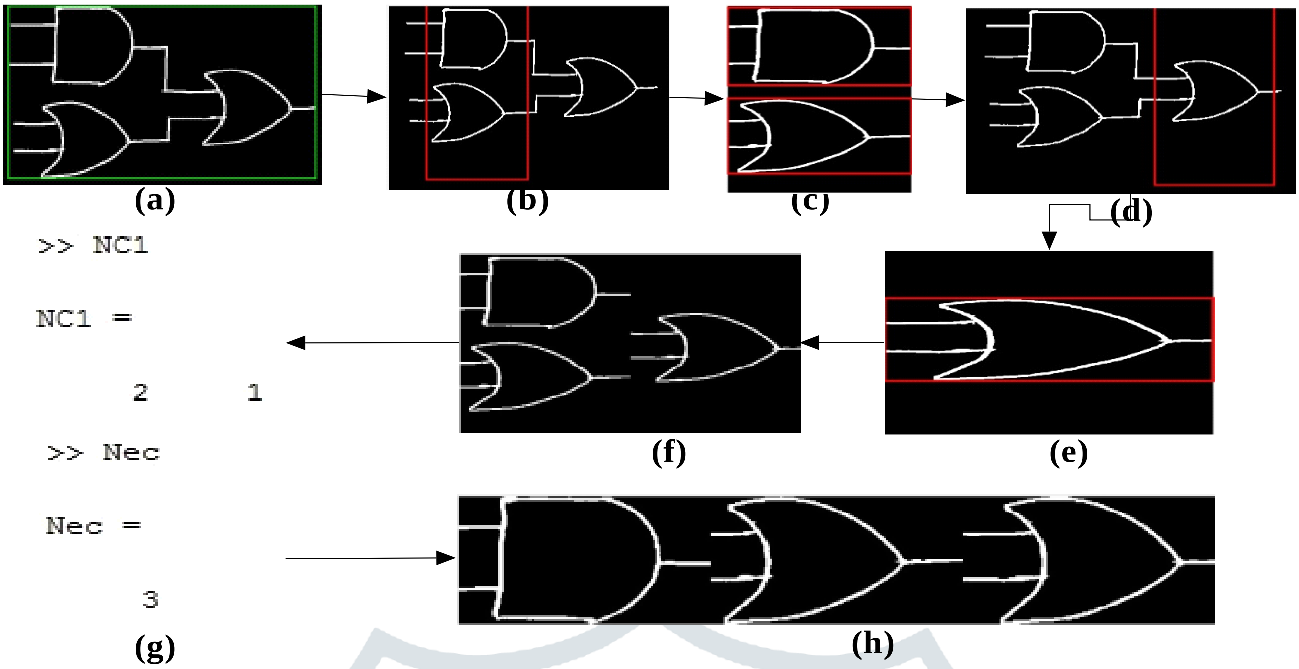
$$\text{Img\_array= crop(Inr,BoundindBox } (x+\Delta\ y\ w+\Delta\ \ l)) \qquad (10)$$

$$\text{Is=segmentation (img\_array)} \qquad (11)$$

$$\text{Ir=resize(Is,[240 420])} \qquad (12)$$

$$A_{(m,n)}=[Ir] \qquad (13)$$

Where, [x y w l]=x coordinate, y coordinate, width and length, Is=Segmented image, Ir=Resized image. Δ=variable, $A_{(m,n)}$= Input image matrix.



```
>> NC1

NC1 =

     2     1

>> Nec

Nec =

     3
```

Figure 5: Elements Segmentation (a) Apply bounding box around the circuit. (b) and (d) Stage by stage element extraction .(c) and (e)Label the element in each stage.(f)Stage wise elements are stored in an array.(g) Identified the number of elements in each stage.(h) Resized element.

1.3. TEMPLATE MATCHING

Once elements are extracted, we have to identify the element to find out the Boolean expression that needs to be simplified to get the reduction between the input and output, for that we have to do template matching. Template matching is finding the maximum possible correlation values of a symbol that best fits to the input image as shown in the figure 6. Before recognize the elements, first we need to create a library, contain samples of 54 handwritten elements in [6] is illustrated in equation (14). Resize each element to 420 x 240 pixels as shown in the figure 7. Here, the input image is stored as matrix and computes the correlation coefficient [7] is as shown in the figure 7(b), compare the resultant maximum correlated value with the stored templates then find the values of the recognized element [8] is as shown in the figure 7 (c) and equation (16). Through this method, handwritten elements are recognized with good accuracy as shown in the table 1.

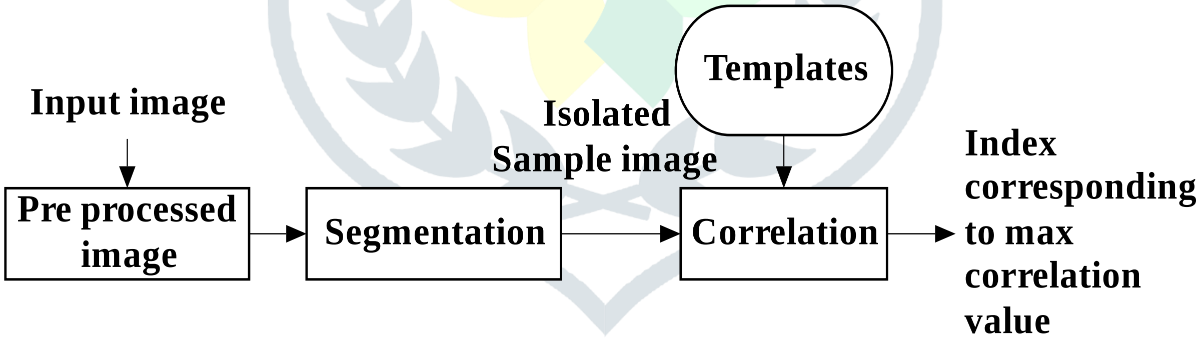


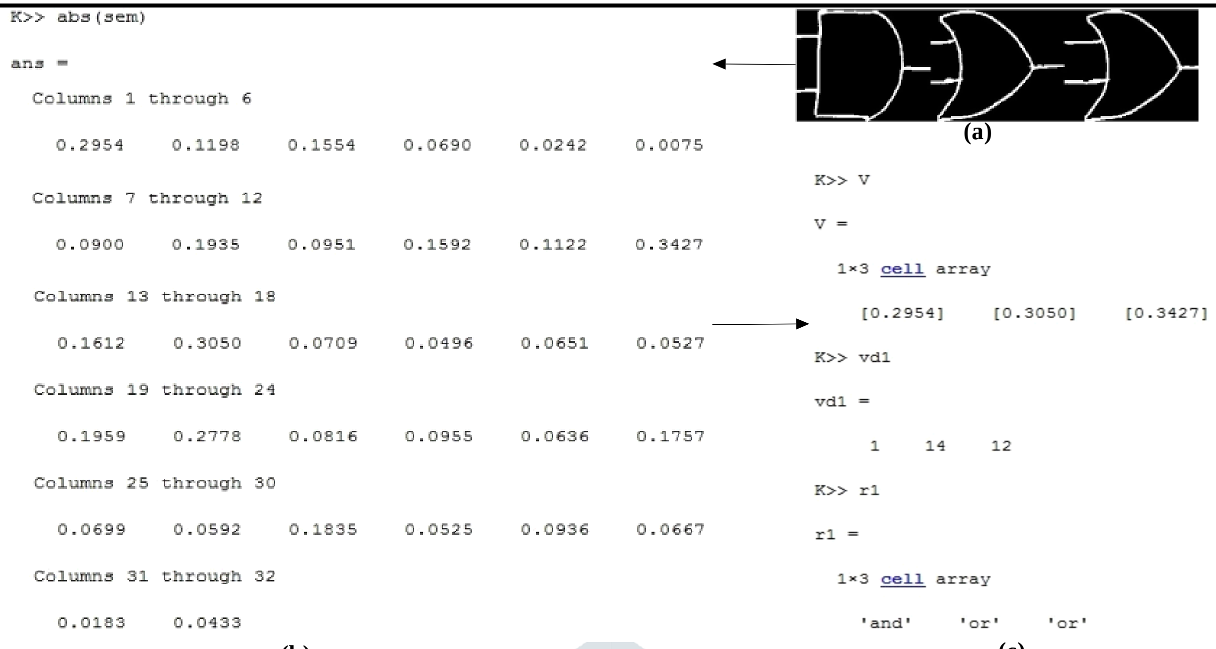
Figure 6: Template Matching

$$\text{Library} = \{\text{templates}\} \tag{14}$$

$$B_{(m,n)} = [\text{templates}] \tag{15}$$

$$r = \text{corr}\ (\text{grouparray}\ A_{(m,n)},\ \text{grouparray}\ B_{(m,n)}) \tag{16}$$

It returns the correlation coefficient r between A and B, where $A_{(m,n)}$= Input image matrix $B_{(m,n)}$= template matrix, A and B are matrices of the same size.



(a)

```
K>> abs(sem)

ans =

  Columns 1 through 6

    0.2954    0.1198    0.1554    0.0690    0.0242    0.0075

  Columns 7 through 12

    0.0900    0.1935    0.0951    0.1592    0.1122    0.3427

  Columns 13 through 18

    0.1612    0.3050    0.0709    0.0496    0.0651    0.0527

  Columns 19 through 24

    0.1959    0.2778    0.0816    0.0955    0.0636    0.1757

  Columns 25 through 30

    0.0699    0.0592    0.1835    0.0525    0.0936    0.0667

  Columns 31 through 32

    0.0183    0.0433
```

(b)

```
K>> V

V =

  1×3 cell array

    [0.2954]    [0.3050]    [0.3427]

K>> vd1

vd1 =

     1    14    12

K>> r1

r1 =

  1×3 cell array

    'and'    'or'    'or'
```

(c)

Figure 7:Recognition process (a) Resized elements. (b) Correlation values (c) Recognized elements using Template matching.

TABLE 1: Result Accuracy for Template Matching

CIRCUIT	NUMBER OF STAGES IN THE CIRCUIT	NUMBER OF ELEMENTS IN EACH CIRCUIT	ELEMENTS IN THE CIRCUIT	RECOGNIZED ELEMENTS	NUMBER OF RECOGNIZED ELEMENTS	%RECOGNIZATION	%FAILURE
CKT 1	2	3	AND,OR,OR	AND,OR,OR	3	100	0
CKT2	3	7	NAND,OR,NAND,AND,AND,OR,OR	NAND,OR,NAND,AND,AND,OR,OR	7	75	25
CKT3	2	4	NOR,NAND,AND,OR	NOR,NAND,AND,NOR	3	80	20
CKT4	3	5	NOR,NAND,AND,OR,AND	NOR,NAND,AND,NOR,AND	4	100	0
CKT5	2	3	AND,AND OR	AND,AND OR	3	100	0

1.4. NETLIST GENERATION

After the recognized the circuit elements, next is to identify the connection among the elements. In order to do that, we have to extract lines and find out nearest neighbours to those lines and there relation with respect to the element in the right side or left side by using element masking. This is done by moving the shifting box stage by stage in vertical direction and fix to exact position of the elements as shown in the figure 8(a). Mask the shifting box position by making this region black (zero) as shown in the figure 8(b) and equation (19). This image is used to identify the number of lines in each stage by applying bounding box to line stages as same procedure as the segmentation as shown in the figure 9.Apply label on the element masked image to find the location of each line in the circuit image to find the nearest neighbour of the each line at four corners, gives information about the continuous connectivity of the circuit as shown in the figure 8(e) and using this information to generate the matrix , first two values in each column in a matrix is inputs and another one is output as shown in the figure 8(f). Further we subtracting the element masked image from the input image as shown in the fig 8(c) is illustrated in equation (21) to find the centroid of each element and centroid of the each line from the element masked image as shown in the figure 8(h) is illustrated in equation (25),(29) by applying bounding box as shown in the figure 8(d) & 8(g) , it used to extract the connectivity between the inputs and the output of the particular element by comparing the centroid of elements with all centroid of the lines in the circuit image as shown in the figure 8(h), if centroid of the particular element is greater than the centroid of the line, that line is identified as input otherwise line is output for that element as shown in the equation (30).

$$\text{Ime=Inr} \tag{17}$$

$$\text{[x y w l]=BoundindBox(Ime)} \tag{18}$$

$$\text{Ime(BoundindBox(x+}\Delta\text{ y w+}\Delta\text{ l))=0} \tag{19}$$

$$\text{Nl=label(Ime)} \tag{20}$$

$$\text{Iml= Inr- Ime} \tag{21}$$

$$\text{Ime[x y l w]=BoundingBox(Ime)} \tag{22}$$

$$\text{Xc=((x+(x+l))/2)} \tag{23}$$

$$\text{Yc=((y+(y+w))/2)} \tag{24}$$

$$\text{Centroid Ce=[Xc ,Yc]} \tag{25}$$

$$\text{Iml[x y l w]=BoundingBox(Iml)} \tag{26}$$

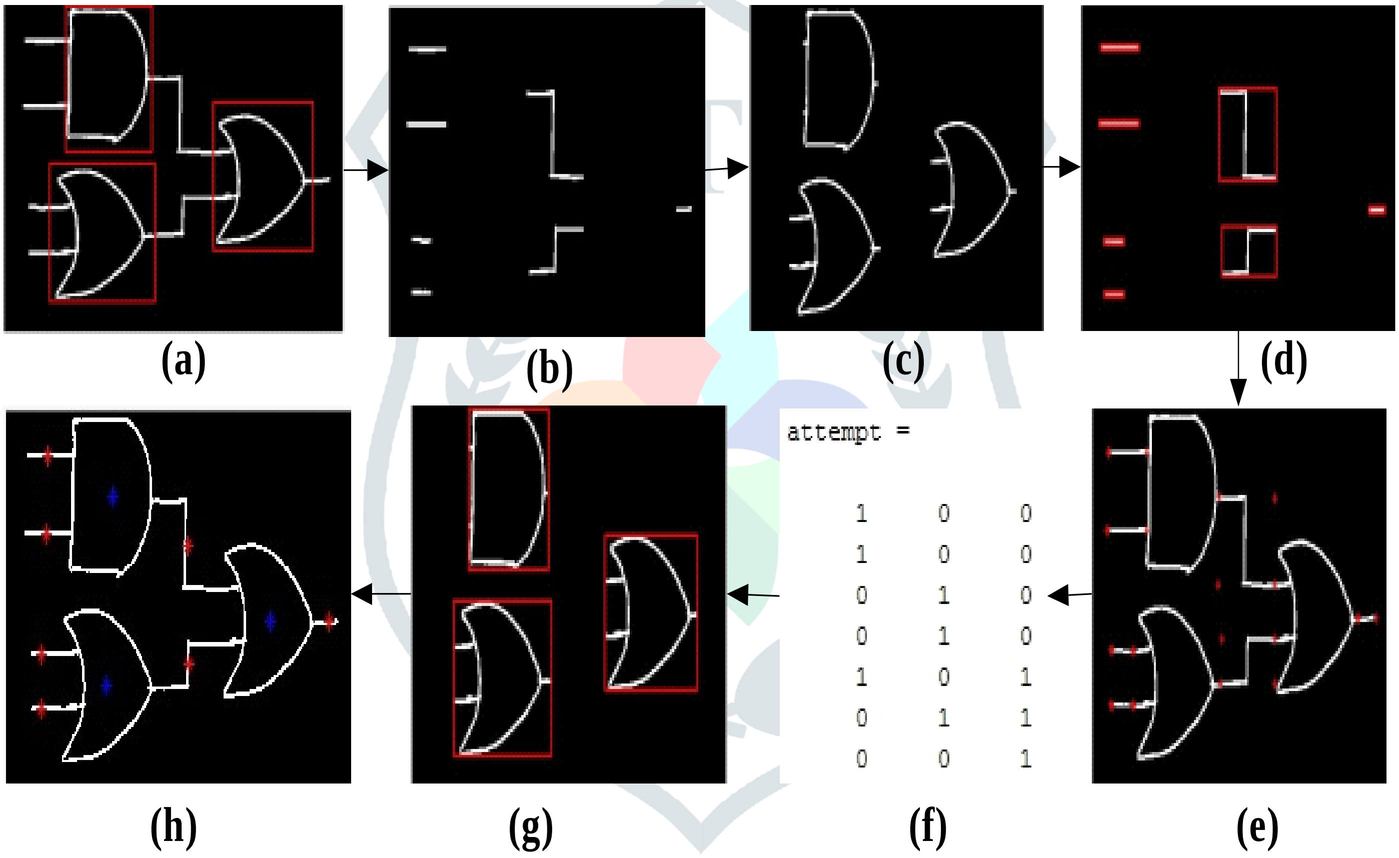
$$\text{Xc=((x+(x+l))/2)} \tag{27}$$

$$\text{Yc=((y+(y+w))/2)} \tag{28}$$

$$\text{Centroid Cl=[Xc ,Yc]} \tag{29}$$

$$\text{Connectivity information of particular } Element = \begin{Bmatrix} input & if\ Ce > Cl \\ output & if\ Ce < Cl \end{Bmatrix} \tag{30}$$

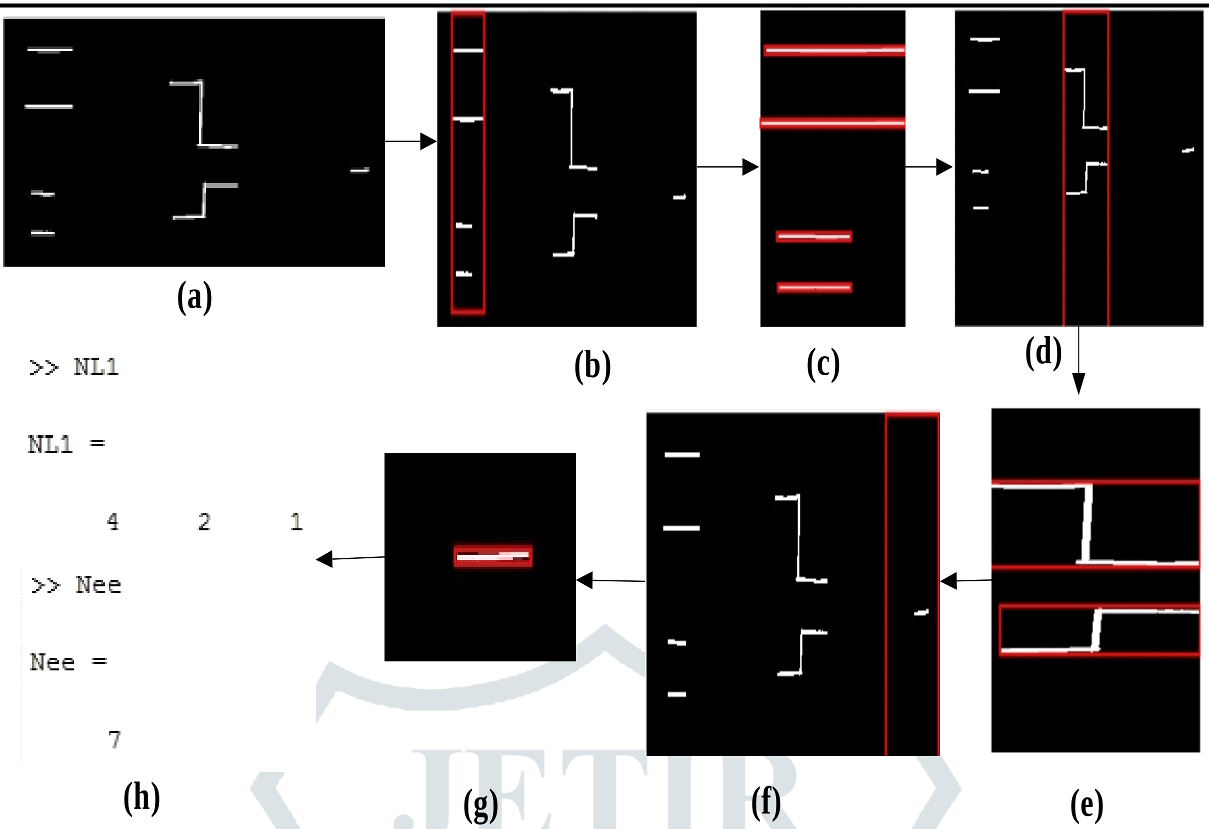
Where Ime=element masked image, Nl=Number of elements in each stage. , Iml=line masked image, Ce=centroid at each element, Cl=centroid at each line.



```
attempt =

     1     0     0
     1     0     0
     0     1     0
     0     1     0
     1     0     1
     0     1     1
     0     0     1
```

Figure 8: Extracting the connectivity information (a) Bounding box on the elements in the circuit. (b) Gate masked image. (c) Line masking. (d) Apply label on the gate masked image. (e) Finding nearest neighbors of the each line. (f) Input and output connectivity matrix. (g) Apply label on the line masked image. (h) Centroid.



```
>> NL1

NL1 =

     4     2     1

>> Nee

Nee =

     7
```

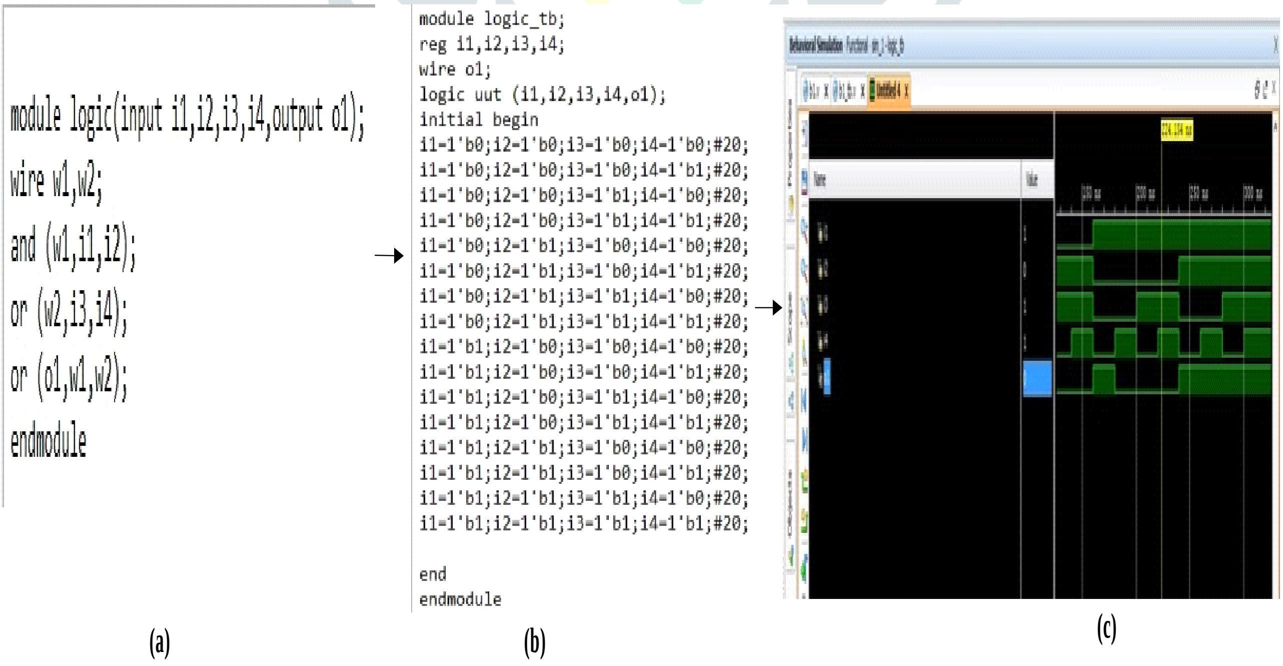
Figure 9: Steps of identifying the number of lines in each stage (a) Gate masked image (b),(d),(f) Applying bounding box to extract the line stage wise. (c),(e),(g) Label the lines in each stage. (h) Identified the number of lines in each stage.

Connectivity information extracted from the circuit is used to generate the verilog code and its test bench for all possible conditions as a netlist as shown in the fig 10(a) and fig 10(b) is illustrated in the equation (31).

$$\text{Netlist generation}=\text{NLG}\{\text{Nc, Nl, recognized elements, n, I}\} \qquad (31)$$

1.5. SIMULATION

A stand alone software that can process schematic of Digital circuit for mentioned parameters and conditions to produce output in the form of tabulations and waveforms. Netlist generated in the MATLAB that is fed to the VIVADO software for the simulation.Simulation results of the circuit as shown in the figure 10(c).



```
module logic(input i1,i2,i3,i4,output o1);
wire w1,w2;
and (w1,i1,i2);
or (w2,i3,i4);
or (o1,w1,w2);
endmodule
```

```
module logic_tb;
reg i1,i2,i3,i4;
wire o1;
logic uut (i1,i2,i3,i4,o1);
initial begin
i1=1'b0;i2=1'b0;i3=1'b0;i4=1'b0;#20;
i1=1'b0;i2=1'b0;i3=1'b0;i4=1'b1;#20;
i1=1'b0;i2=1'b0;i3=1'b1;i4=1'b0;#20;
i1=1'b0;i2=1'b0;i3=1'b1;i4=1'b1;#20;
i1=1'b0;i2=1'b1;i3=1'b0;i4=1'b0;#20;
i1=1'b0;i2=1'b1;i3=1'b0;i4=1'b1;#20;
i1=1'b0;i2=1'b1;i3=1'b1;i4=1'b0;#20;
i1=1'b0;i2=1'b1;i3=1'b1;i4=1'b1;#20;
i1=1'b1;i2=1'b0;i3=1'b0;i4=1'b0;#20;
i1=1'b1;i2=1'b0;i3=1'b0;i4=1'b1;#20;
i1=1'b1;i2=1'b0;i3=1'b1;i4=1'b0;#20;
i1=1'b1;i2=1'b0;i3=1'b1;i4=1'b1;#20;
i1=1'b1;i2=1'b1;i3=1'b0;i4=1'b0;#20;
i1=1'b1;i2=1'b1;i3=1'b0;i4=1'b1;#20;
i1=1'b1;i2=1'b1;i3=1'b1;i4=1'b0;#20;
i1=1'b1;i2=1'b1;i3=1'b1;i4=1'b1;#20;

end
endmodule
```

Figure 10: (a) Verilog code obtained in text output (b). Test bench obtained in text output. (c) Simulation results of the circuit.

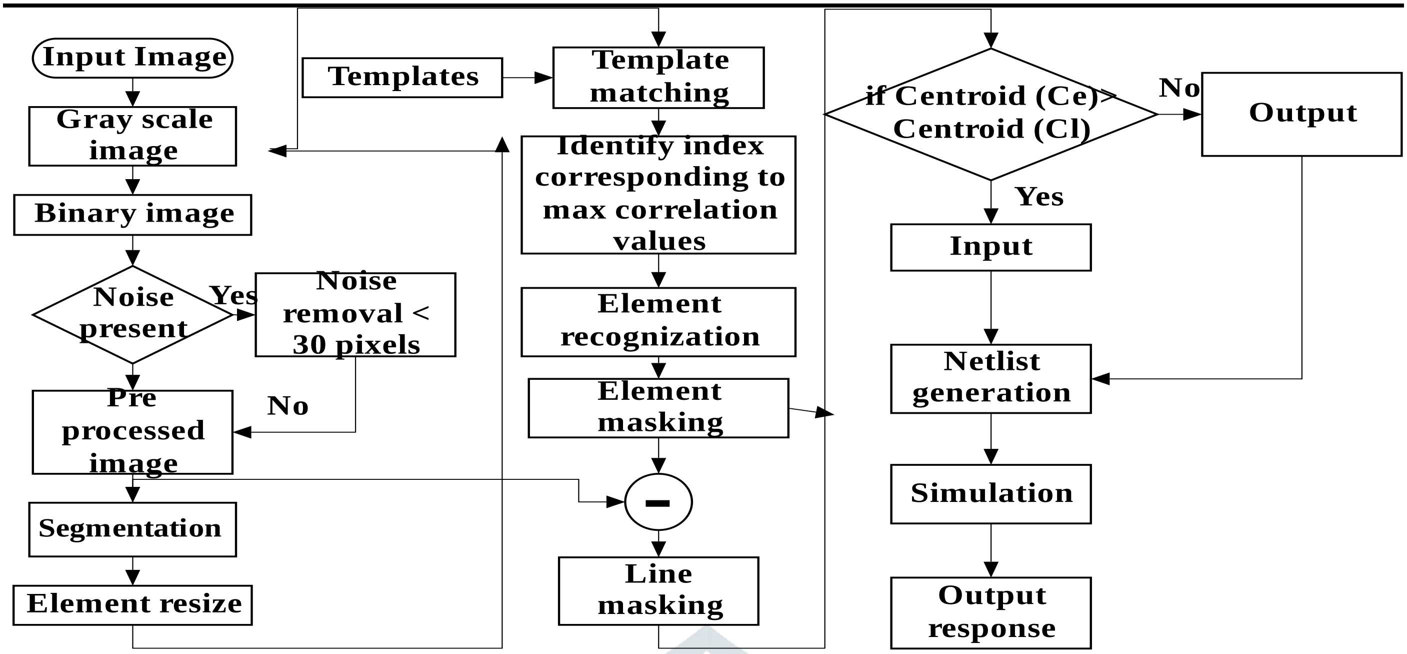


Figure 11: Flow chart of proposed work.

1.6. CONCLUSION AND FUTURE ENHANCEMENT

A series of algorithms for element recognition, identification and connectivity extraction have been developed. The algorithms have been validated with regressive simulation. The results of tests have shown with good qualitative accuracy. Current algorithms have been tested for test cases with fixed orientation of the elements, these algorithms can be further improved to account for the orientation of element.

System prototype are still facing a small number of demerits with extracting circuit elements for multi stages circuit ,element masking and detection of neighbor pixels of each line of the element to generating a netlist is still a challenging problem. Therefore, further research could be done to improve the system prototype into a better system by taking multi stage circuits and sequential circuits.

1.7. ACKNOWLEDGMENTS

First of all, I would like to express my sincere appreciation to the guide, Dr Mahesh sir, assistant professor of ECE Department. I am very grateful for suggesting the topic and providing guidance, invaluable advice and consistent support throughout the project. I have learned a lot from him. Without these, this project would not have been possible. I would like to thank the HOD and professor of the ECE department, Dr.K.A.Radhakrishna Rao. I would really like to express my thanks to M.Subramanyam sir, assistant professor of ECE Department and my friend Abhilasha H for some of their valuable inputs and discussions. I would like to express my gratefulness to all the non-teaching staff members in ECE department for image processing laboratory support with encouragement.

REFERENCES

[1]. Mohamed T. Boraie and Adil S. Balghonaim ," Optical Recognition of Electrical Circuit Drawings ",King Fahd University of Petroleum and Minerals Dhahran 3 126 1, Saudi Arabia IEEE @ 1997.
[2]. Yuhong Yu, Ashok Samal, “A System for Recognizing a Large Class of Engineering Drawings”, IEEE Transactions, August 1997.
[3]. Patricia Ruiz, Bernab´e Dorronsoro, Manuel Bueno Mora, “A New CAD Tool to Assist Industrial Design Engineering Students in the Implementation of Electrical Diagrams – CADDI”, IEEE ©2018.
[4]. G V S S K R Naganjaneyulu, A.V.Narasimhadhan and K Venkatesh, ‘‘Performance evaluation of OCR on poor resolution text document images using different pre processing steps ’’, IEEE ©2014.
[5]. Dr. A. Srisaila1, S. Kranthi2, K. Pranathi3, P. Madhavi Latha4, “Tag identification for vehicles by using Connected Components based Segmentation and template matching based recognition” , IEEE ©2018.
[6]. Wei Li, Stefan Neullens, Matthias Breier, Marcel Bosling, Thomas Pretz and Dorit Merhof,"Text Recognition for Information Retrieval in Images of Printed Circuit Boards",RWTH Aachen University, 52056 Aachen, Germany IEEE @2014.
[7]. Aravinda C.V, Dr.H.N.Prakash, “Template Matching Method For Kannada Handwriten Recognition Based On Correlation Analysis”, IEEE ©2014.
[8]. Smruti Rekha Panda, ”Odia Offline Typewritten Character Recognition using Template Matchingwith Unicode Mapping” IEEE ©2015.