VLSI System Design

ELE301P

LAB - 8 - Report

Praveen B R

COE19B007

Submission Date: 03/11/2021

INDEX

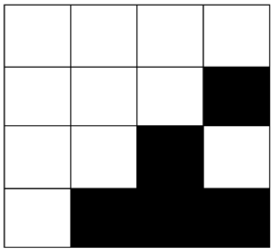
Q1) Run Length Encoding

Objective:

To perform run length encoding (Row wise) for following 2-D data in Verilog behavioural model and represent the output in 48-bit size.

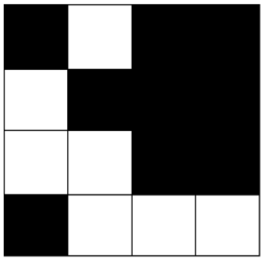
Theory:

- □ Run length Encoding is a form of lossless Data Compression in which runs of data is stored as a single value and its count.
- □ Inorder to perform the RLE for each row and print the output in 48-bit Output we need consider given images as a 2-D matrix
- □ In the given figures consider **Black as 0** and **White as 1** to insert in the 2-D array.



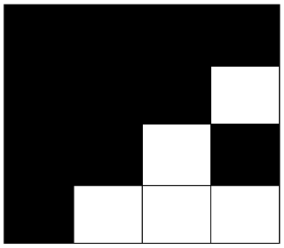
1111
1110
1101
1000

A



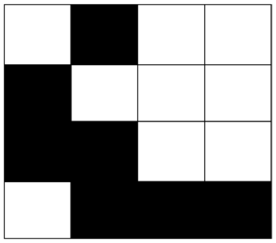
0100
1000
1100
0111

B



0000
0001
0010
0111

C



1011
0111
0011
1000

D

- □ After the Insertion we performed the RLE for each row and inserted the output in the 48-bit binary number(encoded the whole data into 48-bit number).

The Encoding Rules are as follows :

- □ Inorder to be able to encode completely, the worst case where 16 boxes converting to individual (alternating black white) in 48 bits we can maximum allot 3 bits for each.
- □ So, there can be a maximum of 16 triplets each with the first bit denoting the color, and other 2 bits denoting the number of appearances of the color in that row in a continuous manner.
- □ (00 - 1 ; 01 - 2 ; 10 - 3 ; 11 - 4 ;)

Eg: Row with data 1110 will be encoded as 110000
Row with data 1010 will be encoded as 100000100000

Code:

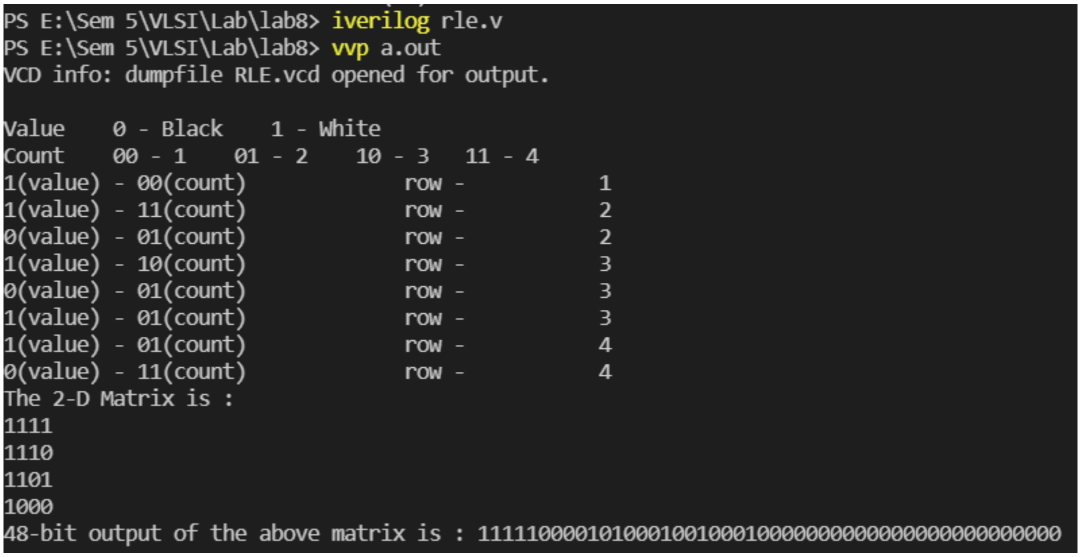
```
module Run_Length_Encoding(arr,op);
input [3:0][3:0] arr;
output reg [47:0] op;
reg previous,present,insert=0;
integer i,j,index;
reg size;
reg [1:0] count;
reg [3:0] Temp;
always@(*)
begin
op = 0;
size =4;
index = 47;
$display("\nValue     0 - Black     1 - White");
$display("Count     00 - 1     01 - 2     10 - 3    11 - 4");
for(i = 0;i<4;i = i + 1)
begin
Temp = arr[i];
for(j = 3;j>=0; j = j - 1)
begin
count = 0;
while((j>0) && (Temp[j] == Temp[j - 1]))
begin
j = j - 1;
count = count + 1;
end
```

```
$display("%b(value) - %b(count) \t\t row - %d",Temp[j],count+1'b1,i + 1);
op[index]= Temp[j];
index = index - 1;
{op[index],op[index - 1]} = count;
index = index - 2;
end
end
end
endmodule

module RLE_test_bench;
reg [3:0][3:0] Input;
wire [47:0] Output;
Run_Length_Encoding R1(Input,Output);
initial
begin
#0  Input[0] = 4'b1111;Input[1] = 4'b1110;Input[2] = 4'b1101;Input[3]
=4'b1000;
#10 Input[0] = 4'b0100;Input[1] = 4'b1000;Input[2] = 4'b1100;Input[3]
=4'b0111;
#10 Input[0] = 4'b0000;Input[1] = 4'b0001;Input[2] = 4'b0010;Input[3]
=4'b0111;
#10 Input[0] = 4'b1011;Input[1] = 4'b0111;Input[2] = 4'b0011;Input[3]
=4'b1000;
#10 $finish;
end
initial
begin
$monitor("The 2-D Matrix is
:\n%b\n%b\n%b\n%b\n",Input[0],Input[1],Input[2],Input[3],"48-bit output of
the above matrix is : %b",Output);
end
initial
begin
$dumpfile("RLE.vcd");
$dumpvars(0,RLE_test_bench);
end
endmodule
```

Output/Waveform:

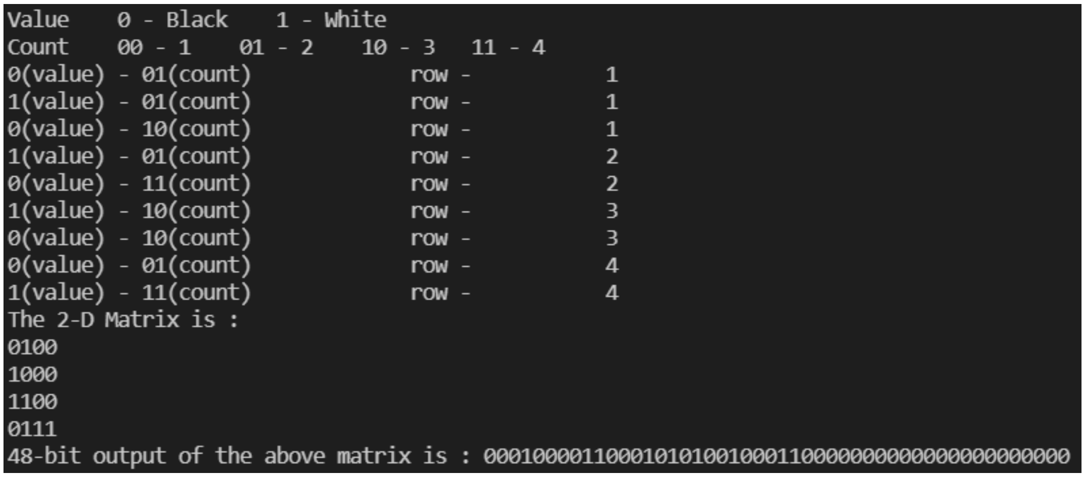
Terminal

A) 111110000101000100100010000000000000000000000000



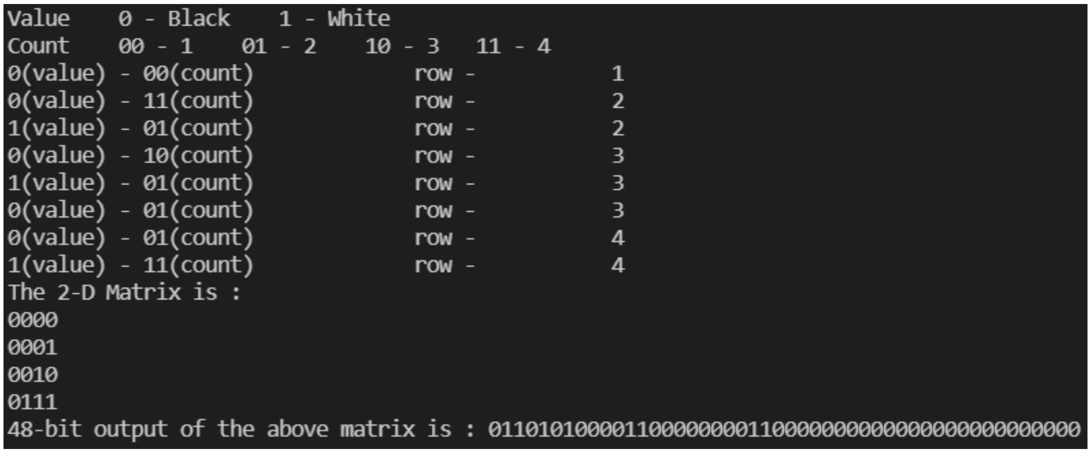
```
PS E:\Sem 5\VLSI\Lab\lab8> iverilog rle.v
PS E:\Sem 5\VLSI\Lab\lab8> vvp a.out
VCD info: dumpfile RLE.vcd opened for output.

Value    0 - Black    1 - White
Count    00 - 1    01 - 2    10 - 3   11 - 4
1(value) - 00(count)            row -           1
1(value) - 11(count)            row -           2
0(value) - 01(count)            row -           2
1(value) - 10(count)            row -           3
0(value) - 01(count)            row -           3
1(value) - 01(count)            row -           3
1(value) - 01(count)            row -           4
0(value) - 11(count)            row -           4
The 2-D Matrix is :
1111
1110
1101
1000
48-bit output of the above matrix is : 111110000101000100100010000000000000000000000000
```

B) 000100001100010101001000110000000000000000000000



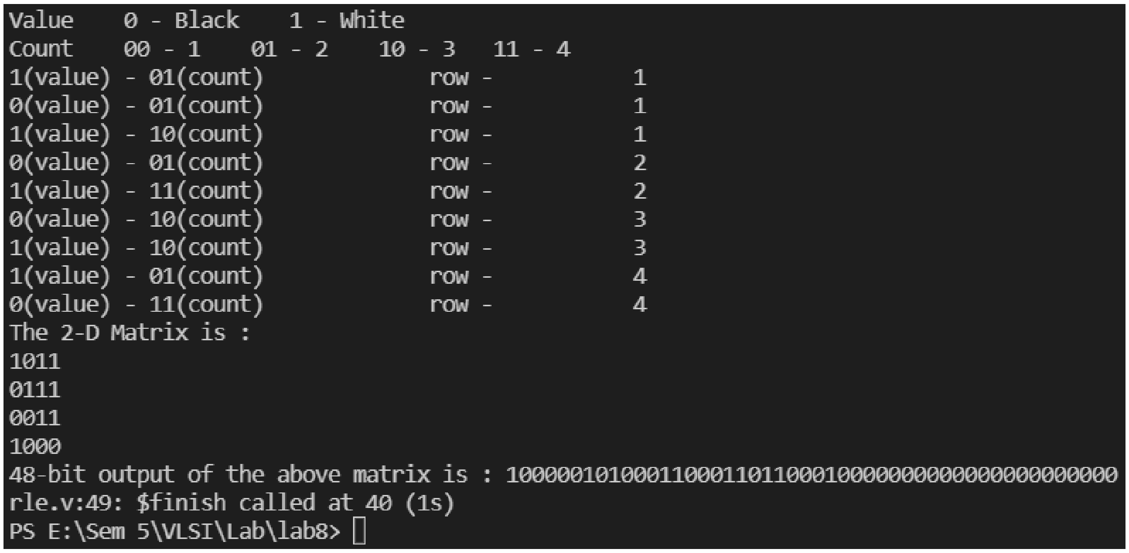
```
Value    0 - Black    1 - White
Count    00 - 1    01 - 2    10 - 3   11 - 4
0(value) - 01(count)            row -           1
1(value) - 01(count)            row -           1
0(value) - 10(count)            row -           1
1(value) - 01(count)            row -           2
0(value) - 11(count)            row -           2
1(value) - 10(count)            row -           3
0(value) - 10(count)            row -           3
0(value) - 01(count)            row -           4
1(value) - 11(count)            row -           4
The 2-D Matrix is :
0100
1000
1100
0111
48-bit output of the above matrix is : 000100001100010101001000110000000000000000000000
```

C) 011010100001100000000110000000000000000000000000



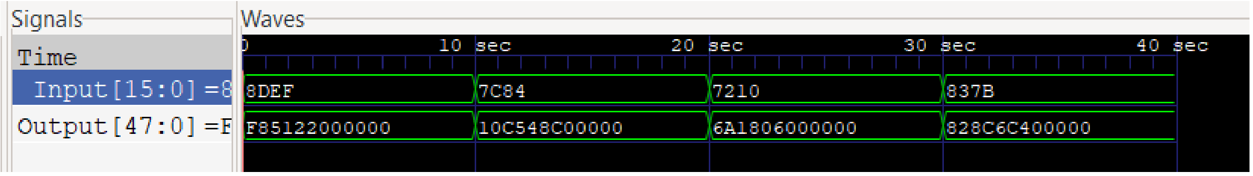
```
Value    0 - Black    1 - White
Count    00 - 1    01 - 2    10 - 3   11 - 4
0(value) - 00(count)            row -           1
0(value) - 11(count)            row -           2
1(value) - 01(count)            row -           2
0(value) - 10(count)            row -           3
1(value) - 01(count)            row -           3
0(value) - 01(count)            row -           3
0(value) - 01(count)            row -           4
1(value) - 11(count)            row -           4
The 2-D Matrix is :
0000
0001
0010
0111
48-bit output of the above matrix is : 011010100001100000000110000000000000000000000000
```

D) 100000101000110001101100010000000000000000000000



```
Value    0 - Black    1 - White
Count    00 - 1    01 - 2    10 - 3    11 - 4
1(value) - 01(count)             row -          1
0(value) - 01(count)             row -          1
1(value) - 10(count)             row -          1
0(value) - 01(count)             row -          2
1(value) - 11(count)             row -          2
0(value) - 10(count)             row -          3
1(value) - 10(count)             row -          3
1(value) - 01(count)             row -          4
0(value) - 11(count)             row -          4
The 2-D Matrix is :
1011
0111
0011
1000
48-bit output of the above matrix is : 100000101000110001101100010000000000000000000000
rle.v:49: $finish called at 40 (1s)
PS E:\Sem 5\VLSI\Lab\lab8> 
```

Waveform



Data is represented as hexadecimal.

Conclusion:

Thus, run length encoding (Row wise) has been successfully performed for given 2-D data in Verilog behavioural model and represented the output in 48-bit size.

Miscellaneous Questions

Question 1 Find the applications of Run length encoding?

- It was employed in Transmission of analog television signals in olden days.

- It is used for Image Compression during Chatting and in which we can increase the space.

- Run length Encoding is the Standard Approach in facsimile (FAX) coding.

- They are used in Video and Audio Compression and Text Files too.

Question 2 Mention the disadvantages of Run length encoding.
The Drawbacks are as Follows :

- If there are no repeatings in the Data to Compress then the Encoded Data will be large.
Example : 1010 (4 bits) will be encoded as 100000100000 (12 bits)

- Result needs to be decoded