Proposal of Novel, 4 columned Netlist format

Aaryan Makwana, 22BEE002
dept of electrical engineering, Institute of technology.
Nirma University

***Abstract*—This electronic document is a "live" template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. **CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.* (*Abstract*)**

Keywords—component, formatting, style, styling, insert (*key words****)***

I. INTRODUCTION

Netlist is a file which describes the connections of all nets (wires) in a digital circuit, containing information of signal origin and termination, and possible fanning out. ISCAS has provided a netlist format (7 columns) which is accepted by the VLSI community and widely used in softwares and analysis. This paper aims on proposing a new netlist which contains the same information in 4 columns instead of 7. Which provide benefits of storage space and program memory.

The paper will talk about ISCAS netlist, explaining the netlist in section 2, draw backs of ISCAS netlist and modifications to eliminate them in section 3, quantification of benefits gained from the modification in section 4 and conclusion of use of new netlist format in section 5.

II. ISCAS NETLIST

ISCAS netlist is a 7-columned table which contains information of net no., connection, type, no. of fan-outs, no. of fan-ins, fan-in 1, fan-in 2. Considering an example to understand the format, look at the image 1 and table 1.

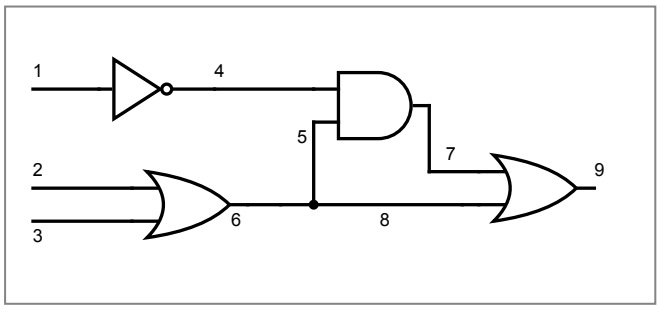

Image 1: logical circuit 1

Net no.	Conn	Type	#fanout	#fanin	Fanin 1	Fanin 2
1	gate	inpt	1	0	-	-
2	gate	inpt	1	0	-	-
3	gate	inpt	1	0	-	-
4	gate	not	1	1	1	-
5	fan	from	6	1	-	-
6	gate	or	2	2	2	3
7	gate	and	1	2	4	5
8	fan	from	6	1	-	-
9	gate	or	0	2	7	8

Table 1: ISCAS netlist for logical circuit 1

The first row of the table represents net1, which is an primary input. This is encoded by "type" which says "inpt". Since its an input, it has 1 fan-out, and no fan-ins. The same is true for rows 2 and 3.

The fourth row represents net4 and its an output from not gate. Its connection type is gate and type is not. Since its a not gate, it should have only one fan-in which is net1 in this case. Number of fan-outs for a net which is not an output should always be 1, which indicates flow of logic is not stuck at the current net. Similar to 4th row, 6th row also represents an output of logic gate. But net6 is output from OR gate and thus has 2 fan-ins. Since it also fanouts into net5 and net8, it has two fan-out. Two two input nets of the gate are net2 and net3, which are mentioned in fan-in1 and fan-in2.

The fifth row represents net5 which is a fan-out of net6. Thus its "conn", "type" and "#fanout" is "fan"-"from"-"6" respectively, which says its a fan-out of net6. Since logic flow does not end at this net, no. of fan-ins is not zero. This is same in case of net8 or row 8.

The 9th row, is a primary output, thus logic flow terminates at this net explaining 0 in "#fanout". Since it is also an output from OR gate, it has two inputs, net7 and net8.

This is how the complete information of how the gates are connected is encoded in the table.

III. CONVERSION FROM ICAS TO 4 COLUMN NETLIST

ISCAS gives a good way to encode circuit in a table, but it is also observed that some of its columns contain redundant information and thus can be optimised or discarded. The first optimisation merges the second and third column, as using two words (data points) to describe type of net is excessive and can be done in 1. Eg- for primary input, "type" will be "inpt" for a logic gate output, the "type" will be "AND", "OR" etc, depending on the type of gate. For a fan-out, the "type" will be "buff". In case of a primary output, one net needs to be expressed in two rows, the first row represents its logic gate connection (if any) and the other declares it as an output. After this optimisation, the transformed table is given in table 2.

Net no.	Type	#fanout	#fanin	Fanin 1	Fanin 2
1	inpt	1	0	-	-
2	inpt	1	0	-	-
3	inpt	1	0	-	-
4	not	1	1	1	-
5	buff	6	1	-	-
6	or	2	2	2	3
7	and	1	2	4	5
8	buff	6	1	-	-
9	or	0	2	7	8
10	Outpt	0	1	9	-

Table 2: first modification of ISCAS netlist for logical circuit 1

Looking at table 2, it is conclude that we have eliminated one column and added one (number of output) rows thus the total data points stored have reduced.

The second optimisation focuses on "#fanin" and "#fanout". Depending upon type of net, ie. "inpt", "AND", "NOT", we can infer the no. of fan-ins, meanwhile no. of fanouts can be ignored as moving through logic it can be easily calculated. In case of fan-outs, the net from which the fan-out occurs can simply be moved to "fanin 1". Thus removing these rows does not reduce the information stored in the table. The modified table is given in table 3.

Net no.	Type	Fanin 1	Fanin 2
1	inpt	-	-
2	inpt	-	-
3	inpt	-	-
4	not	1	-
5	Buff	6	-
6	or	2	3
7	and	4	5
8	Buff	6	-
9	or	7	8
10	Outpt	9	-

Table 3: final modification of ISCAS netlist for logical circuit 1

Observing table 3, it can be concluded that the information stored in table 1 is present in table 3 without loss of information. It is also observed that no. of datapoints required to store the same information have drastically reduced, from 63 in table 1 to just 40 in table 3.

IV. Memory Benefits

In order to quantify the memory usage in each case, certain assumptions are taken; which are, each data point in the table is a string of length 5 (as no data in the table is more than 5 letters). Therefore its size will be 4*5 = 20 bytes.

In ISCAS netlist, no. of rows is determined by no. of nets and no. of column always remains same (7). Therefore building an equation relating no. of nets and size of netlist will be:

$$S_{ISCAS} = 140 \times N_{nets}$$

In case of 4-column netlist the size depends on both, no. of nets and no. of outputs. Total no. of rows is the sum of no. of nets and no. of outputs. Meanwhile no. of columns remain constant, which is 4. Building and equation from this relation, we get:

$$S_{4col} = 80 \times (N_{nets} + N_{outputs})$$

Plotting these equations as a gradient in N_{nets} vs $N_{outputs}$ plane, we can solve this using linear programming. Where $S_{4col} < S_{ISCAS}$. This plot is represented in image 2.

Looking at image 2, the green area represents condition where moving to 4-column netlist gives size advantage. In red area, ISCAS is better. The purple area marks area which is not possible as no. of outputs cannot be greater than no. of nets itself. The dividing line between red and green is:

$$N_{nets} = \frac{11}{7} N_{outputs}$$

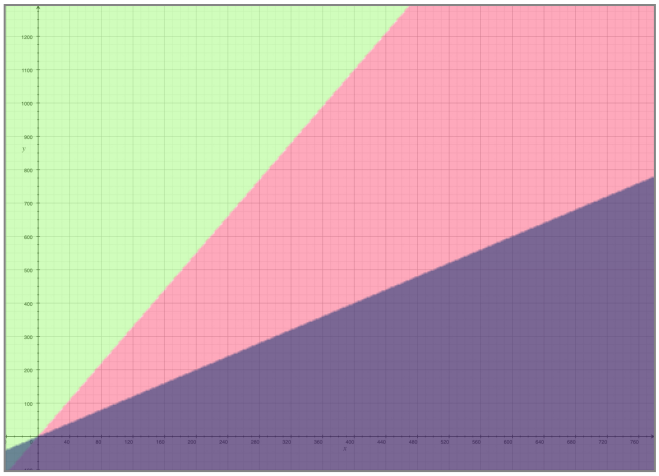

Image 2: gradient plot of size advantage of 4-column vs ISCAS (green) size of ISCAS > size of 4-column, (red) size of ISCAS < size of 4-column, (purple) such netlist is not possible.

Conclusion

This new 4-column netlist provides definite memory benefits when used in netlists where output nets are less than 63% of total nets. Since most netlist or logical circuits have even less than 30% of its total nets as outputs, it is concluded that use of the new net provides benefit in most cases.