Task 3: Issues at minimal cost implementation

Given a truth table, it was requested to see what happens when the truth table is implemented with the smallest quantity of logic gates. To perform this task, ICs that have NAND logic gates only will be used. This is because in most of ICs there are many logic gates so it's a waste of logic gates if we use for example just one of them. Note that this can also be implemented with NOR only ICs.

Low cost approach

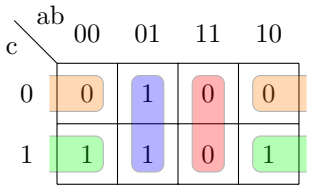

Figure 1: Karnaugh Map of the given truth table

Let Z be the output label, its reduced minterms expression is:

$$Z = (\bar{A}.B) + (C.\bar{B}) \tag{1}$$

Also, its reduced maxterms expression is:

$$Z = (\bar{A} + \bar{B}).(C + B) \tag{2}$$

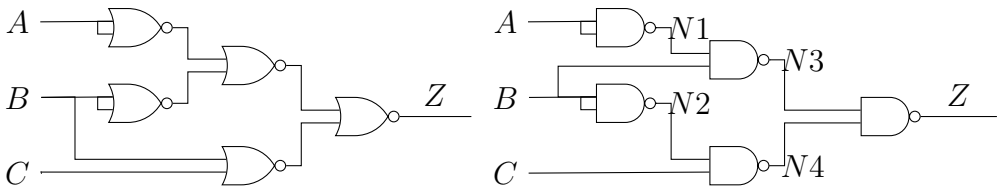

Figure 2: Implementation with NOR gates and NAND gates respectively

Note that the quantity of logic gates in both cases are the same.

Hazards

Sometimes propagation delays cause unexpected and unwanted transitions in the ouput. Depending on what we see in the output is how this issue is called. If there is a transition from 1 to 0 and the output was supposed to stay at 1 it is said that the circuit has a static 1-hazard. If the output must go from 0 to 1(or 1 to 0) and before establishing at 1 (or 0), it changes its value it is said that the circuit has a dynamic hazard.

Hazards can always be found at the Karnaugh map. An input that follows multiple paths to an output can create a glitch (due to the hazards in the circuit) if, for example, one path has an inverter (in this case implemented with a NAND logic gate) and one does not. This issue is called "asymmetric path delay" and it can be seen in Figure 1 with the input B.

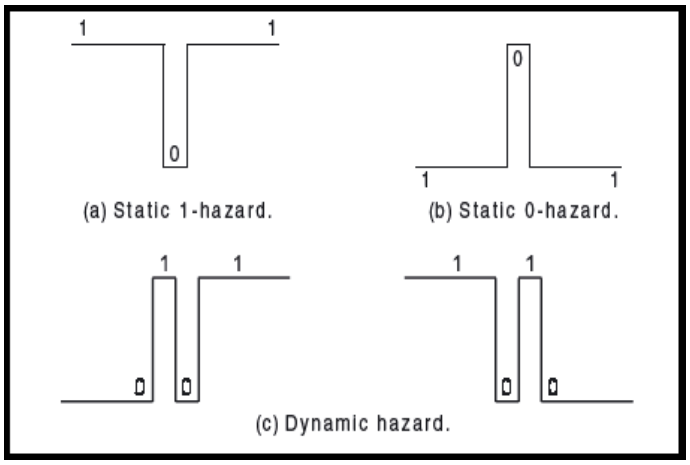

Figure 3: Different types of hazards

Static 1-Hazard example

Take the NAND implementation into account, assume all the propagation times equal and then observe transition of the states A,B,C from 011 to 001 respectively.

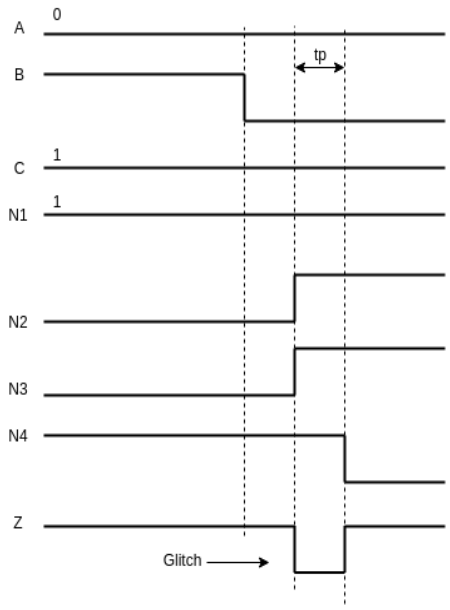

Figure 4: Propagation time analysis

Note that tp in Figure 4 denotes the propagation time.

It can be seen that there should be a glitch at the output.The different combination of delays that produce a glitch may or may not be likely to occur in the implementation of the circuit. In some instances it is very unlikely that such delays would occur.

Measures obtained from NAND implementation

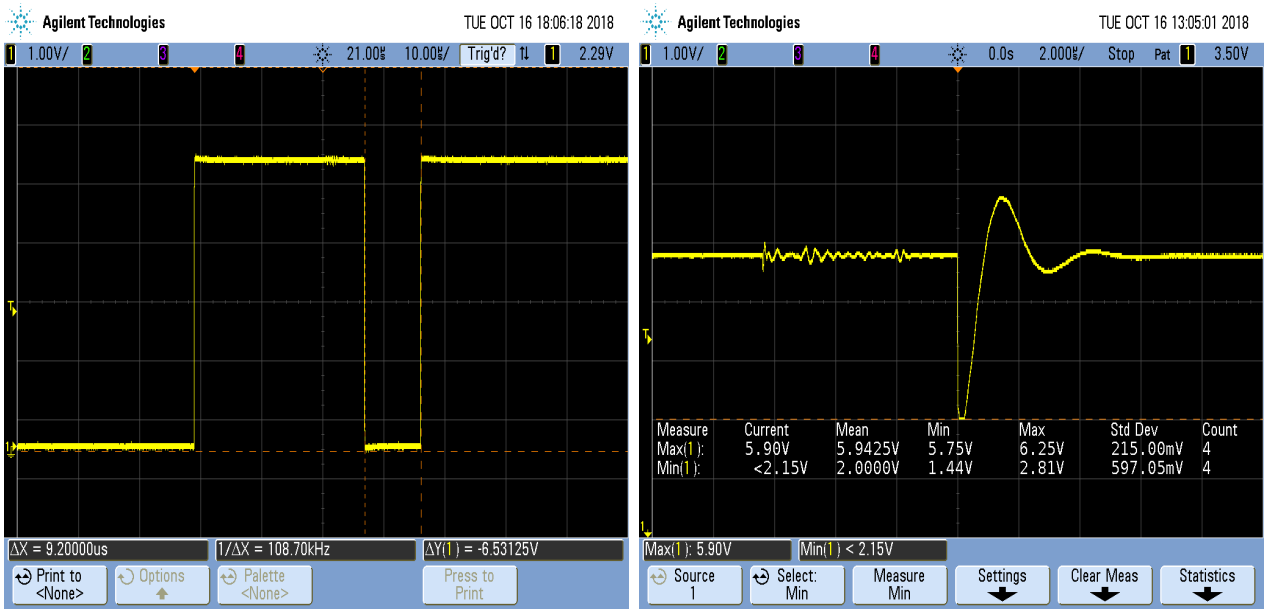

Figure 5: Dynamic hazard and static 1-hazard respectively

Conclusion

In order to remove glitches it is needed to add redundant groups in the Karnaugh Map so that disjoint groups overlap.

c \ ab	00	01	11	10
0	0	1	0	0
1	1	1	0	1

Figure 6: Karnaugh with redundant logic

In this case the output is given by:

$$Z = (\bar{A} + \bar{B}).(C + B) + (\bar{A}.C)$$

Note that the new term term of the expression is not dependent of B so regardless of the variation of B there won't be glitches in the output.