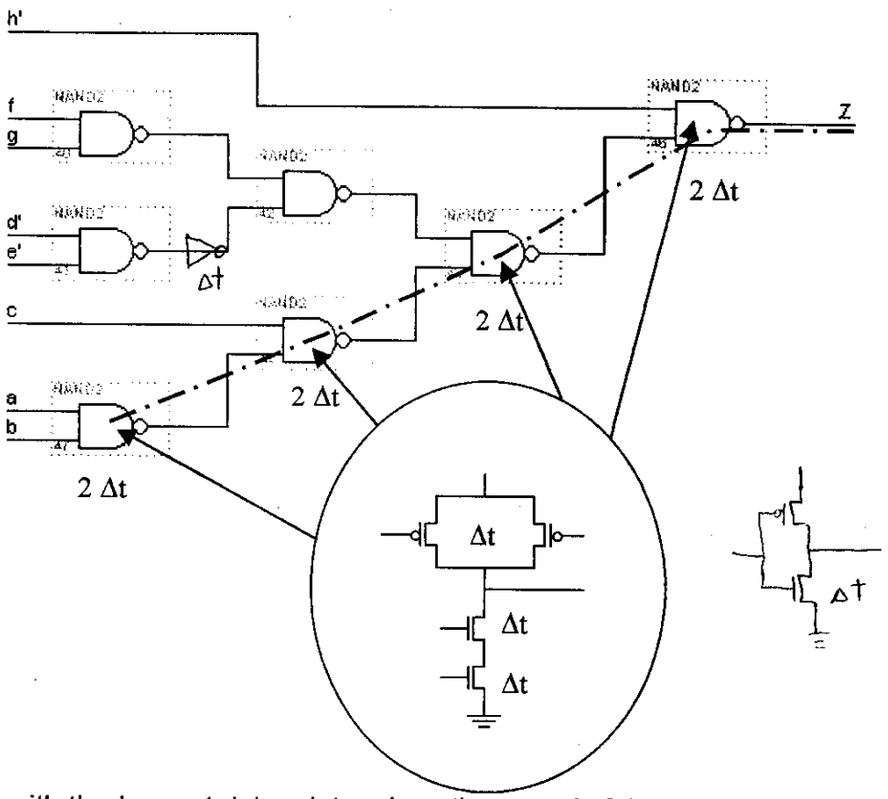
Jason Evans
EEC 180A

Consider the following logic function:

$Z = (ab + c)(d + e + fg) + h$

This function can be implemented using either two level or multi-level logic. Limiting the implementation to 2 Input NAND gates requires multi-level logic as shown below.

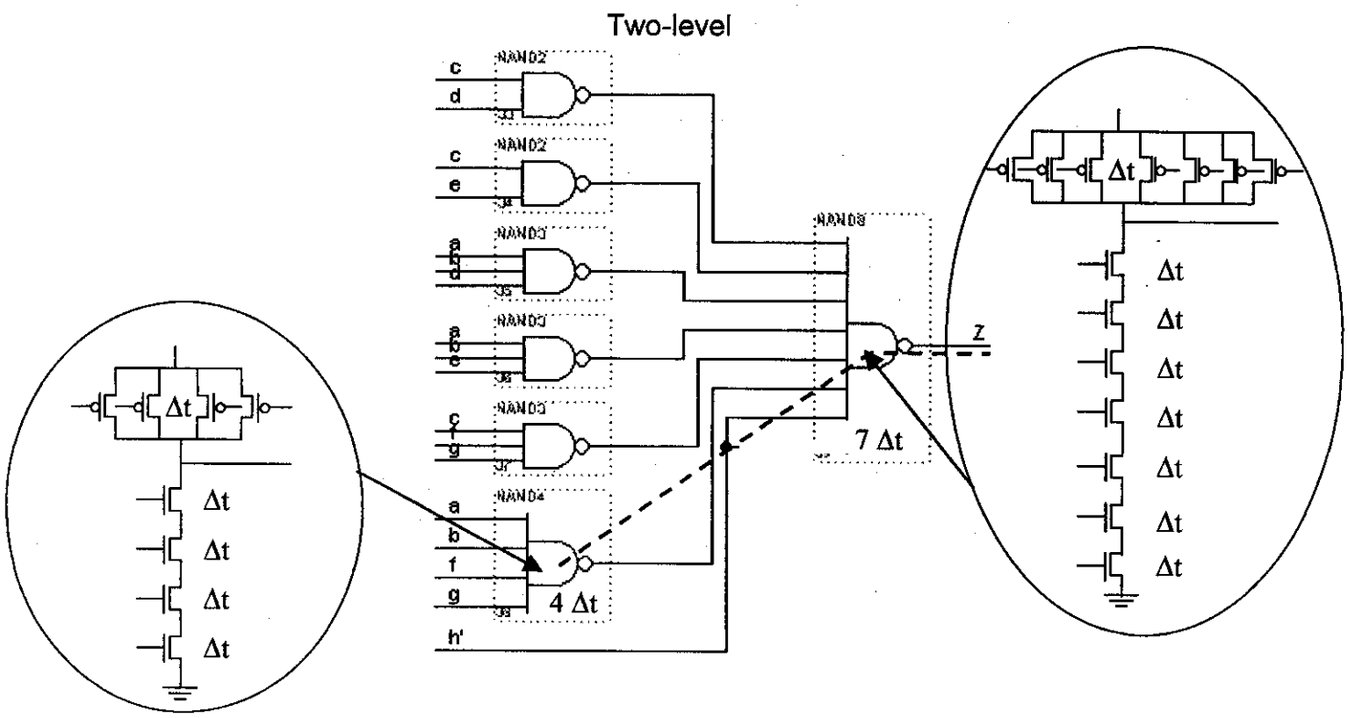
Multi-level



The path with the longest delay determines the speed of the network. In this case, the longest delay includes four 2 Input NAND gates, each with a maximum delay of 2 Δt through the n-tree making the total delay through the network a maximum of 4 X (2 Δt) = ~~8 Δt~~ 8 Δt + 1 Δt = 9 Δt

Two-level logic

To implement the function using two-level logic, NAND gates with large fan-ins are required. When expanded, the function has seven terms requiring a NAND gate with a fan-in of seven. The two-level logic schematic is shown below.



Again, the speed of the network depends on the slowest path. In this case, the gates with the highest fan-in are the slowest because adding more transistors to the gate adds more delay as shown above. The slowest path includes the 4 Input NAND gate which contributes a 4 Δt delay and the 7 Input NAND gate which contributes a 7 Δt delay making the total maximum delay through this network equal to **11 Δt**.

It seems that with this function the **multi-level logic design is faster** than the two-level logic design. However, this is only true when all of the transistors in the gates have equal delay.

The following approach uses transistors with unequal delay:

Another approach to this problem is to look at the datasheets for each of the NAND gates and find the delay specification. Using this method, the two-level logic design would be faster because the delay of a 2 Input and a 4 Input NAND gate is 10ns each and the delay of an 8 Input NAND gate is 15ns. This would make the total delay through the two-level network 25ns whereas the total delay through the multi-level network would be 40ns.

Since this exercise required that all transistors have the same delay, the multi-level logic design is fastest.