

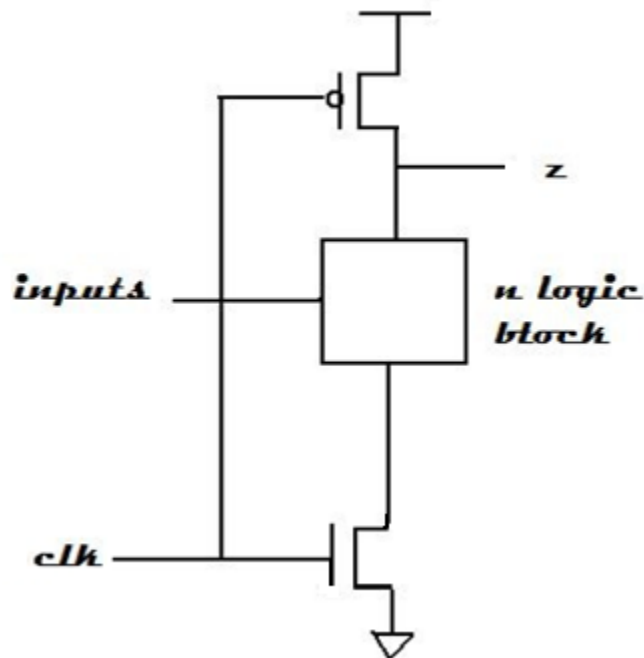
DYNAMIC CMOS LOGIC

Figure 5: Dynamic CMOS logic

This logic looks into enhancing the speed of the pull up device by precharging the output node to vdd. Hence we need to split the working of the device into precharge and evaluate stage for which we need a clock. Hence it is called as dynamic logic. The output node is precharged to vdd by the pmos and is discharged conditionally through the nmos. Alternatively you can also have a p block and precharge the n transistor to vss. When the clock is low the precharge phase occurs. The path to Vss is closed by the nmos i.e. the ground switch. The pull up time is improved because of the active pmos which is already precharged. But the pull down time increases because of the ground switch.

There are a few problems associated with the design, like

1. Inputs have to change during the precharge stage and must be stable during the evaluate. If this condition cannot occur then charge redistribution corrupts the output node.
2. A simple single dynamic logic cannot be cascaded. During the evaluate phase the first gate will conditionally discharge but by the time the second gate evaluates, there is going to be a finite delay. By then the first gate may precharge.

3.6 CLOCKED CMOS LOGIC (C2MOS)

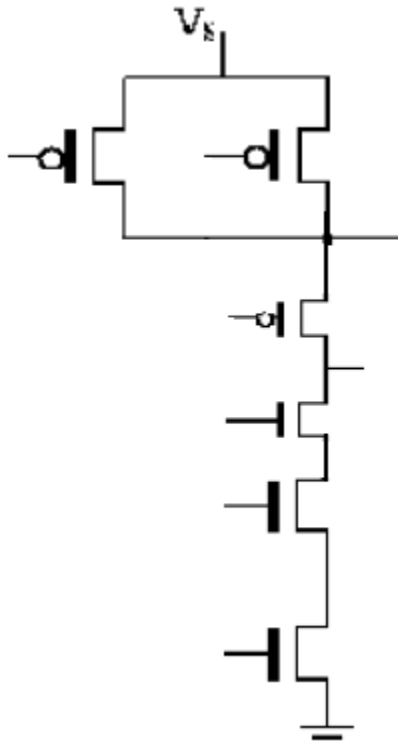


Figure 6: C2mos logic.

3.7 CMOS DOMINO LOGIC

The disadvantage associated with the dynamic CMOS is overcome in this logic. In this we are able to cascade logic blocks with the help of a single clock. The precharge and the evaluate phases retained as they were. The change required is to add a buffer at the end of each stage. This logic works in the following manner. When the $\text{clk}=0$, i.e. during the precharge stage the output of the dynamic logic is high and the output of the buffer is low. Since the subsequent stages are fed from the buffer they are all off in the precharge stage. When the gate is evaluated in the next phase, the output conditionally goes low and the output of the buffer goes high. The subsequent gates make a transition from high to low.

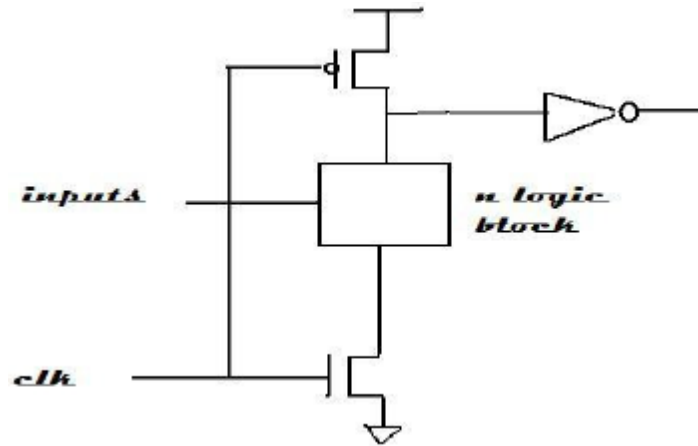


Figure 7: Cmos domino logic.

Hence in one clock cycle the cascaded logic makes only one transition from 1 to 0 and buffer makes a transition from 0 to 1. In effect we can say that the cascaded logic falls like a line of dominos, and hence the name. The advantage is that any number of logic blocks can be cascaded provided the sequence can be evaluated in a single clock cycle. Single clock can be used to precharge and evaluate all the logic in a block. The limitation is that each stage must be buffered and only non-inverted structures are possible.

A further fine tuning to the domino logic can also be done. Cascaded logic can now consist of alternate p and n blocks and avoid the domino buffer. When $clk=0$, i.e. during the precharge stage, the first stage (with n logic) is precharged high and the second a p logic is precharged low and the third stage is high. Since the second stage is low, the n transistor is off. Hence domino connections can be made.

The advantages are we can use smaller gates, achieve higher speed and get a smooth operation. Care must be taken to ensure design is correct.

3.7.1 NP DOMINO LOGIC (ZIPPER CMOS)

