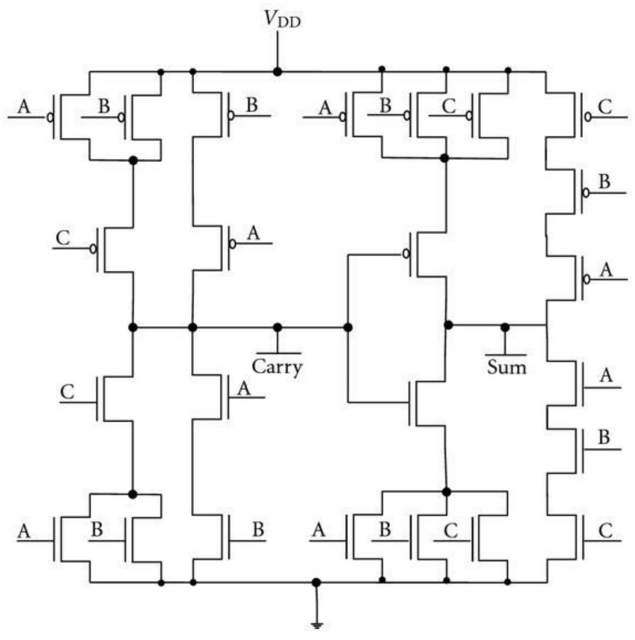
Name: Abhishek Sen **Roll. No. -**18700316089

Microelectronics and VLSI Design (EC-702)

1. Draw the circuit of CMOS Full Adder and explain its operation.



Explanation:

First we have to write the Boolean expression for the Full Adder. Then take the complement as CMOS always perform automatic inversion. Firstly design a Half Adder circuit. As the output carry of one half adder is taken as input of the fresh input of the second half adder, therefore for designing the final carry of the full adder, the 2 half adder carry needs to be added (OR operation).

When input logic is low(0), the nMOS is OFF and pMOS is ON. Hence the output is connected to VDD through pMOS. When input logic is high(1), the pMOS is OFF and nMOS is ON. Hence the output is connected to GND through nMOS.

2. Write down the difference between Twin-tub process and n-well process.

In n-well process the substrate is P-type. P channel device is formed into the n-well. The n-channel device is directly constructed on the substrate itself. In twin tub or twin well process both n-well and p-well are fabricated on single N-type substrate. It is possible to tune independently threshold voltage, body effect and the channel trans-conductance of both P and N type transistors using this process. With n+ or p+ as starting material , lightly doped epitaxial layer is formed on this layer n-well and p-well are formed. Unbalanced drain parasitics are observed in p and n well CMOS process. Twin tub process avoids this problem.

3. Show that for Ideal CMOS inverter (W/L) p =2.5 (W/L) n

For ideal CMOS inverter

$V_{OH} = V_{DD}$

$V_{OL} = 0$

$V_{inv} = [\ V_{thn} + (1/\sqrt{\beta})(V_{DD} + V_{thp})] / (1 + 1/\sqrt{\beta})$

$\beta = \eta n/\eta p = [\mu n(\varepsilon ox/tox)n\ (W/L)n\ ] / [\mu p(\varepsilon ox/tox)p(W/L)p]$

for $\beta = 1$,

$(W/L)n / (W/L)p = \mu p/\mu n \approx 1/2.5$

$(W/L)p \approx 2.5\ (W/L)n$

4. Explain the design flow of an ASIC.

1. The ASIC design process begins from writing a functional description containing detailed requirements for the chip. We can start design on basis of a functional description prepared by the customer. Alternatively, we can create the functional description document based on the customer's demands expressed in any form. At no time will we share your information with anyone without your explicit permission.
2. Based on your demands, our team estimates the amount of resources needed and produces a Statement of Work. After reaching an agreement, the actual work is started.
3. The first step is similar to FPGA design. The following tasks are run in parallel:

- Writing a synthesizable *RTL (register transfer level)* description (either on Verilog or VHDL) of the device.
- Writing a *behavioral* model, which is used to verify that the design meets its requirements.
- Writing a *verification plan* and a corresponding *verification environment* which describes and implements the method of proving the design correctness.

4. The RTL description is verified against the behavioral model by out dedicated Validation and Verification Department. This approach reduces the probability of the design error since no RTL designer tests his own code.
5. Most modern ASIC designs are complex enough to the stage when it's impossible to tell apart valid chips from faulty at the production stage without special preparations during the design stages. These preparations are called DFT (*design for test*). DFT techniques include:

- *Scan path* insertion – a methodology of linking all registers into one long shift register (scan path). This can be used to check small parts of design instead of the whole design (the latter being almost always impossible).
- *BIST (built-in self test)* – a device used to check RAMs. After being triggered it feeds specific test patterns to the RAM module, reads back and compares results.
- *ATPG (automatic test pattern generation)* – a method of creating test vectors for scan paths and BIST automatically. Most modern EDA tool chains incorporate such a feature.

6. The synthesizable and verified RTL undergo *logic synthesis*. The synthesized reads RTL input, user-specified *constraints* and a *cell library* from the foundry. The output of the synthesis process is a gate-level *netlist*.
7. The netlist must undergo *formal verification* to prove that RTL and netlist are equivalent.
8. Preliminary timing results after synthesis are analyzed, critical paths are checked against the project performance requirements. If needed, the RTL description, constraints or synthesis options are modified, and the synthesis is repeated.

9. When timing constraints are finally met, the design proceeds to the *layout*, which consists of floor planning, placement and routing. Some other important tasks are performed at this step, including *clock tree* insertion.
10. Final (post-layout) timing results are again compared with performance requirements. If it doesn’t fit, the floor plan can be changed or placement run with other parameters.
11. The last stage before tape out includes the following checks:

- *DRC (design rule check)* is a check that the layout conforms to the foundry-specific rules.
- *LVS (layout versus schematic)* is a formal equivalence check between the post-synthesis netlist and the final layout.

12. At last the resulting layout in GDSII format is handed to the semiconductor fabrication plant (foundry). This process is called *tape out*.

5. Discuss the concepts of regularity, modularity and locality in VLSI design.

Regularity: Decomposition of a large system in simple and similar blocks as□ much as possible. Example:□ Design of array structures consisting of identical cells - such as a parallel multiplication array.

Modularity: Modularity in design means that the various functional blocks□ which make up the larger system must have well-defined functions and interfaces. Modularity allows that each block or module can be designed□ relatively independently from each other. All of the blocks can be combined with ease at the end of the□ design process, to form the large system. The concept of modularity enables the parallelisation of the□ design process.

Locality: The concept of locality also ensures that connections are mostly□ between neighbouring modules, avoiding long-distance connections as much as possible.