Solution of Model Q Paper 1

Q1 (a) Explain the basic steps of CMOS fabrication.

Introduction

CMOS (Complementary MOS) technology is used to fabricate most modern digital ICs because of its low power and high packing density. Fabrication is done on a silicon wafer by repeatedly applying steps like **oxidation, photolithography, etching, diffusion/implantation, deposition, and metallization**.

1. Starting Material – Wafer Preparation

- Begin with a **single-crystal silicon wafer**, usually **p-type** for an n-well CMOS process.
- Wafers are grown using the **Czochralski process** and sliced into thin circular discs.
- They are lapped, polished, and **chemically cleaned** to remove dust and contaminants.

---

2. Oxidation

- Purpose: to grow a thin insulating layer of **silicon dioxide (SiO_2)** on the wafer.
- Process: place wafer in a **high-temperature furnace** (900–1100°C) with:
    - **Dry oxygen (O_2)** → high-quality, thin oxide (gate oxide).
    - **Steam (H_2O)** → thicker oxide (field oxide).
- Reaction:

  $Si\ (solid) + O_2 \rightarrow SiO_2$
- Oxide acts as:
    - Insulator
    - **Mask** for dopant diffusion/implantation
    - Gate dielectric in MOSFETs

---

3. Photolithography

- Used to **transfer patterns** from a mask to the wafer surface.
- Steps:

1. **Photoresist coating:** spin-coat a thin layer of light-sensitive resist on oxide.
2. **Soft bake:** remove solvents.
3. **Mask alignment:** place a photomask (with desired pattern) over wafer.
4. **Exposure:** shine UV light; it changes solubility of exposed regions.
5. **Development:** developer solution removes either exposed or unexposed resist (depending on positive/negative resist).
6. **Hard bake:** strengthen the remaining resist.

Result: patterned photoresist defining where later steps occur.

Diagram: oxide + resist, then after exposure and development, showing patterned resist windows.

---

4. Etching

- Remove material (usually oxide) in the **opened windows**.
- **Wet etching:** chemical solution (e.g., HF for SiO_2), isotropic.
- **Dry etching (plasma/ion):** anisotropic, used in submicron technology for vertical sidewalls.
- After etching, the **underlying silicon** is exposed through windows, while other areas are still protected by oxide + resist.

---

5. Diffusion / Ion Implantation

- Purpose: to introduce **dopant atoms** (boron for p-type, phosphorus/arsenic for n-type) into selected regions.
- **Diffusion:**
    - Wafer exposed to dopant gas at high temperature.
    - Dopants diffuse into silicon.
- **Ion implantation (modern process):**
    - Dopant ions are accelerated in an electric field and **shot into silicon**.
    - Very precise control of dose and depth.
- Used to form:

    - **n-well** or p-well
    - **Source and drain** regions (n^+ for NMOS, p^+ for PMOS)
    - Channel stop regions, etc.

---

6. Gate Oxide and Polysilicon Gate Formation

- Grow a **thin gate oxide** (~nm range) where MOS transistors will be.
- Deposit **polysilicon** (polycrystalline silicon) on top of gate oxide.
- Pattern polysilicon by lithography + etching → this defines the **gate**.
- MOSFET source and drain implants are typically **self-aligned** to this gate (very important for scaling).

---

7. Metallization (Interconnect Formation)

- Deposit metal layer (Al or Cu).
- Pattern using lithography and etching to form **interconnect lines**:
    - Connect source, drain, gates, power rails, etc.
- Modern processes use **multiple metal layers** separated by inter-layer dielectrics and connected by **vias**.

---

8. Passivation and Packaging

- Deposit a **passivation layer** (e.g., Si_3N_4) to protect the chip from moisture and contamination.
- Open **bond pad windows** through passivation.
- **Dicing**: wafer is cut into individual dies.
- Each die is **bonded** into a package; fine bond wires connect chip pads to package leads.

Summary for Q1(a)

In summary, CMOS fabrication is a **multi-step repeating cycle** of oxidation → lithography → etching → doping → deposition → patterning. The goal is to form millions of MOSFETs with precise dimensions and interconnections on a single wafer.

Q1 (b) Define: (i) Lithography (ii) Packaging (iii) Interconnects.

(i) Lithography

- **Definition:** Process of transferring geometric patterns from a **mask** onto the wafer using **photoresist and UV light**.
- Steps: coat, expose, develop, then use the pattern to guide etching or implantation.
- It controls **device size, gate length, and layout**, so it directly affects scaling and chip density.

(ii) Packaging

- After fabrication, the bare die is **very fragile** and cannot be used directly.
- **Packaging** is the process of:
  - Mounting the die on a substrate/lead frame.
  - Connecting die pads to external leads (wire bonding / flip-chip).
  - Encasing in plastic or ceramic.
- Functions:
  - Protects chip from **physical damage and environment**.
  - Provides **mechanical form factor** and **electrical interface** for PCB mounting.

(iii) Interconnects

- **Metal lines** inside the IC that connect various transistors, gates, blocks, and power rails.
- Modern VLSI uses:
  - Several **metal layers** (M1, M2, …)
  - **Vias** to connect between layers
- Interconnects are critical for:
  - Delay (RC of wires)
  - Power distribution
  - Signal integrity (crosstalk, noise)

Q2 (a) Explain the twin-tub CMOS process and its advantages.

Concept of Twin-Tub

- In basic CMOS, you may have:
  - **N-well process**: PMOS in n-well, NMOS in p-substrate.
- In **twin-tub (twin-well)**:
  - You create **both n-well and p-well** in a lightly doped epitaxial layer.
  - **NMOS** devices sit in a **p-well**.
  - **PMOS** devices sit in an **n-well**.

Diagram to draw: cross-section showing:

- Epi layer on substrate
- P-well with NMOS (n^+ S/D)
- N-well with PMOS (p^+ S/D)
- Separate well contacts.

Process Steps (Conceptual)

1. Start with **heavily doped substrate** (e.g., p^+). Grow a **lightly doped epitaxial layer** above it (n-type).
2. Apply oxidation and lithography to define **n-well regions**; implant p-type dopant to form p-wells and drive in.
3. Similarly, define and form **p-wells** with n-type dopant where NMOS will reside.
4. After wells are formed, proceed with:
   - Gate oxide growth
   - Polysilicon gate formation
   - Source/drain implantation:
     - n^+ S/D in p-well → NMOS
     - p^+ S/D in n-well → PMOS
5. Add contacts, interconnect metals, and passivation as usual.

Advantages of Twin-Tub Process

1. **Independent Optimization** of NMOS and PMOS:
    - Can choose doping concentrations and profiles separately for n-well and p-well.
    - Achieve desired **threshold voltages (V_{TH})** for both types.
    - Balance drive strengths (since electron and hole mobilities differ).
2. **Better Performance Symmetry:**
    - More symmetrical rise/fall times because both NMOS and PMOS can be tuned.
3. **Improved Latch-Up Immunity:**
    - Use of **epi layer on heavily doped substrate** reduces substrate resistance.
    - Separate wells and careful design can lower gain of parasitic pnp/npn.
4. **Flexibility in Scaling:**
    - Easier to adapt to newer technologies and different supply voltages.
5. **Reduced Body Effect Issues:**
    - Proper well biasing and doping can minimize body effect for each device type.

Conclusion for Q2(a)

Twin-tub CMOS is more complex than simple n-well, but it is the **industry standard** for high-performance CMOS processes because it provides better control over device parameters and reliability.

Q2 (b) Explain thermal oxidation of silicon with a neat diagram.

Aim of Oxidation

- To grow **SiO_2** on the silicon surface for:
    - Field isolation
    - Gate oxide
    - Mask for diffusion/implantation

Basic Principle

- At high temperature, silicon reacts with oxygen or steam to form **SiO_2**.
- Two common processes:
  1. **Dry oxidation:**

     $Si+O_2 \rightarrow SiO_2$
  2. **Wet oxidation:**

     $Si+2H_2O \rightarrow SiO2+2H_2$

Process Steps

1. Wafer is **cleaned** and placed in a **quartz boat**.
2. Inserted into a **horizontal/vertical furnace tube**.
3. Temperature raised to ~900–1100°C.
4. Pass **O_2** for dry oxidation or **steam (H_2O vapour)** for wet oxidation.
5. Oxide layer grows over time; thickness controlled by time and temperature.

Dry vs Wet Oxidation

- **Dry:**
  - Slower growth rate.
  - Very high-quality oxide (few defects).
  - Used for **thin gate oxide**.
- **Wet:**
  - Faster growth (thick oxide in less time).
  - Slightly lower quality.
  - Used for **field oxide** and isolation.

Importance

- Oxide thickness affects:
  - Threshold voltage
  - Gate capacitance
  - Breakdown voltage
- Precise control is crucial for VLSI.

Q3 (a) Explain the operation and I–V characteristics of an n-channel enhancement MOSFET.

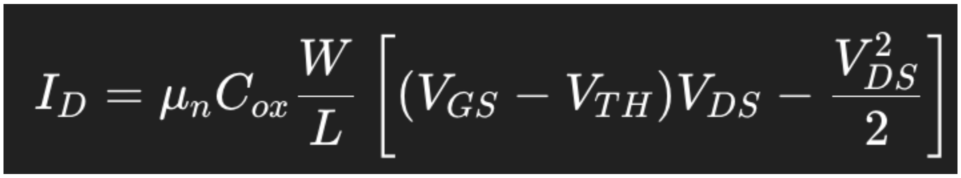
Structure and Basics

- Built on **p-type substrate** with **n^+ source and drain** diffusion regions.
- Gate is polysilicon over thin gate oxide.
- **Enhancement type**: normally OFF at $V_{GS} = 0$.

Diagram: cross-section showing p-substrate, n^+ source/drain, gate oxide, gate.

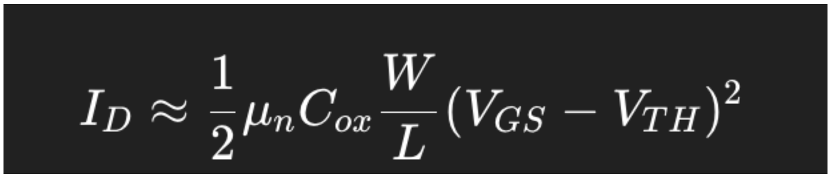
Operation Regions

1. **Cutoff Region ($V_{GS} < V_{TH}$)**
    - No inversion channel under the gate.
    - Drain current $I_D \approx 0$ (except leakage).
    - Device behaves like an open switch.
2. **Linear (Triode) Region**
    - Condition: $V_{GS} > V_{TH}$, and VDS $< (V_{GS} - V_{TH})$.
    - An n-channel forms between source and drain; behaves as a **resistor** controlled by gate voltage.
    - Approximate current equation:



$$I_D = \mu_n C_{ox} \frac{W}{L} \left[(V_{GS} - V_{TH}) V_{DS} - \frac{V_{DS}^2}{2} \right]$$

    - I_D increases with both V_{DS} and V_{GS}.
3. **Saturation Region**
    - Condition: $V_{GS} > V_{TH}$ and $V_{DS} \geq (V_{GS} - V_{TH})$.
    - Channel **pinches off** near the drain.
    - Drain current becomes (approximately) independent of V_{DS}:



$$I_D \approx \frac{1}{2}\mu_n C_{ox}\frac{W}{L}(V_{GS} - V_{TH})^2$$

  - MOSFET behaves like a **current source** controlled by V_GS.

4. **Channel-Length Modulation (Practical Effect)**
   - In reality, I_D slightly increases with V_{DS} in saturation due to effective channel length reduction.
   - Modeled by factor $(1 + \lambda V_{DS})$.

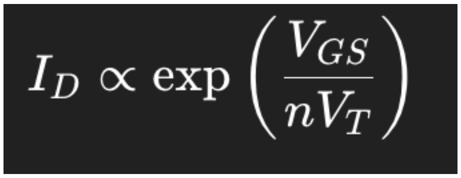
Q3 (b) What is subthreshold operation?

Definition

- Region where $\mathbf{V_{GS} < V_{TH}}$, but **a weak inversion layer** still exists under gate.
- Drain current is small but non-zero – called **subthreshold current** or **leakage current**.

Behavior

- Current approximately follows an **exponential law**:



$$I_D \propto \exp\left(\frac{V_{GS}}{nV_T}\right)$$

  where
  - VTV_TVT is thermal voltage (~26 mV at room temperature)
  - nnn is slope factor (~1–2)
- Device behaves somewhat like a **BJT in forward active region** in this zone.

Importance

- In digital circuits:
  - Contributes to **static power dissipation** (leakage when transistor is “off”).
- In ultra-low-power analog/digital design:

  - Designers **intentionally use subthreshold region** to get nA–µA currents at low supply voltages.

Q4 (a) Draw and explain the flow of circuit simulation using SPICE.

Aim of SPICE

- To simulate **voltages and currents** in circuits mathematically **before** fabricating or building them.

Flow Steps

1. **Netlist Creation**
   - Describe the circuit in text form:
     - Component type (R, C, L, M for MOSFET, Q for BJT etc.)
     - Node connections
     - Values and models
   - Example line: `M1 drain gate source bulk NMOS_MODEL W=... L=...`
2. **Model Definition**
   - Specify transistor models (level 1, 2, BSIM etc.) with parameters like µ, V_TH, λ.
3. **Reading Netlist & Forming Equations**
   - SPICE parses netlist and sets up **nodal equations** using **modified nodal analysis (MNA)**.
   - For linear components (R, C, L), coefficients are fixed.
   - For non-linear devices (diodes, MOSFETs), they are updated iteratively.
4. **DC Operating Point Analysis**
   - Solve for steady-state node voltages with DC sources.
   - Use **Newton–Raphson iteration**:
     - Guess node voltages
     - Evaluate device currents and conductances
     - Solve linearized equations

    - Repeat until convergence.

5. **Requested Analyses**
    - **DC Sweep:** vary a DC source, record outputs.
    - **AC Analysis:** small-signal frequency response around DC operating point.
    - **Transient Analysis:** simulate circuit behavior vs time (e.g. switching waveforms).
6. **Convergence Check**
    - After each iteration/time step, check if difference in voltages/currents < tolerance.
    - If not, reduce time step or damping and repeat.
7. **Results and Post-Processing**
    - SPICE writes results to output files.
    - User plots node voltages, currents, etc., using waveform viewer.

Q4 (b) Explain power dissipation in CMOS circuits.

Total Power in CMOS

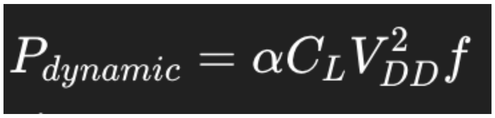
$$P_{total} = P_{dynamic} + P_{short\text{-}circuit} + P_{static}$$

1. Dynamic Power

- Due to **charging and discharging** of capacitors when signals switch.
- Each time a node with capacitance C_L charges from 0 to V_{DD}, energy drawn:

$$E = \frac{1}{2} C_L V_{DD}^2$$

- If a node switches at frequency f with activity factor α:



$$P_{dynamic} = \alpha C_L V_{DD}^2 f$$

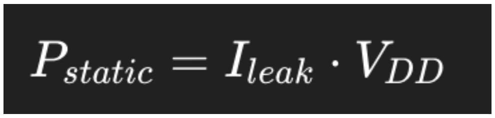
- Dominant component in most CMOS logic.

2. Short-Circuit Power

- During transitions, for a short time both NMOS and PMOS can be **ON simultaneously**, creating a direct path from V_DD to GND.
- This causes **short-circuit current**.
- Happens when input rise/fall is slow.
- Reduced by having steeper transitions and proper transistor sizing.

3. Static (Leakage) Power

- Ideally, static CMOS has **no DC path** from V_DD to GND in steady states.
- In practice, leakage sources:
  - **Subthreshold leakage** (V_GS < V_TH but small current flows).
  - **Gate oxide tunnelling** (very thin gate oxides).
  - **Junction leakage** of reverse-biased diodes.
- Static power:



$$P_{static} = I_{leak} \cdot V_{DD}$$

Techniques to Reduce Power

- Lower V_{DD}
- Reduce C_L (shorter wires, smaller transistors).
- Lower switching activity (clock gating, operand isolation).
- Use high-V_{TH} devices for non-critical paths to reduce leakage.

Q5 (a) Differentiate random logic and structured logic.

Random Logic

- Logic implemented using **arbitrary combinations** of gates.
- Layout has **irregular** placement and routing.

- Typically used for:
    - Small control blocks
    - Glue logic
    - Custom controllers
- Advantages:
    - Flexible, optimized for specific function.
- Disadvantages:
    - Harder to automate for very large designs.
    - Testing and layout more complex.

Structured Logic

- Uses **regular, repeating patterns** like:
    - PLAs / ROMs
    - Standard-cell arrays
    - Systolic arrays
    - FPGAs
- Characteristics:
    - **Regular layout** → easier automated placement and routing.
    - Better **predictable timing and testing**.
- Used in:
    - Datapath units (adders, multipliers).
    - Programmable logic (FPGAs, CPLDs).

Q5 (b) Explain quasi-static register cells.

Register Basics

- A **register** stores one or more bits using latches or flip-flops.
- **Static** storage uses cross-coupled inverters (like SRAM cell).

Quasi-Static Register Cell

- Uses **static CMOS inverters** for storage (like a latch), but controlled by **clocked transmission gates**.
- Operation:
  - **Write phase:** clock enables a transmission gate; input D passes through to internal node of latch.
  - **Hold phase:** clock disables gate; cross-coupled inverters hold the value.

Why "Quasi-Static"?

- Data is statically stored as long as power exists.
- But internal refresh or clocking may be needed in some designs (e.g. when node leakage is non-negligible in very small technologies).
- Simpler than full dynamic storage (like DRAM), but may be optimized for area or speed, hence called "quasi-static".

Q6 (a) What are races and hazards in sequential circuits?

Hazards

- Occur in **combinational logic** due to **different propagation delays** in paths.
- Types:
  1. **Static-1 Hazard:**
     - Output is supposed to stay at 1.
     - Due to delay mismatch, it momentarily becomes 0 (glitch).
  2. **Static-0 Hazard:**
     - Output is supposed to stay at 0.
     - Momentarily goes to 1.
  3. **Dynamic Hazard:**
     - Output is supposed to change once, but changes multiple times (e.g., 0 → 1 → 0 → 1).

Example diagram: two-path network implementing Y = A + B using different gates, and timing waveforms showing glitch.

Avoiding Hazards:

- Add **redundant logic terms**.
- Use **K-map hazard removal** techniques.
- Synchronize outputs using flip-flops.

Races (in Asynchronous Sequential Circuits)

- In asynchronous circuits, the **state is encoded** in multiple bits (e.g., Q1, Q2).
- **Race**: when input causes multiple state bits to change and **their order of change matters**.
- **Non-Critical Race:**
    - Order of bit changes does not affect final state.
    - Safe.
- **Critical Race:**
    - Different transition orders lead to **different final states**.
    - Highly undesirable → can cause unpredictable behavior.

Avoiding Races:

- Use **proper state assignment** (Gray coding; only one bit changes at a time).
- Use **synchronous design** as far as possible.
- Ensure only one flip-flop changes for a given transition.

Q6 (b) Briefly explain state minimization.

Aim

- Reduce number of states in FSM while **preserving input–output behavior**.
- Benefits:
    - Fewer flip-flops
    - Less combinational logic
    - Lower area and power

Method (for Completely Specified Machines)

1. **Write State Table:**
    - List present states, inputs, next states, and outputs.
2. **Find Equivalent States:**
    - Two states are equivalent if, for every possible input sequence, they produce the same outputs and go to equivalent states.
3. **Partitioning Method:**
    - Start with groups of states having same output.
    - Recursively split groups if next-state patterns differ.
    - When no further split is possible, states left in same group are equivalent.
4. **Merge States:**
    - Replace each group of equivalent states with a **single new state**.
5. **Draw Reduced State Table and Diagram**
    - Implement new FSM with fewer states.

For **incompletely specified machines**, successive-approximation or implication tables can be used.

Q7 (a) Explain latch-up in CMOS circuits.

Parasitic Structure

- In a CMOS process (e.g., n-well), you inherently form **parasitic BJTs**:
    - PNP: p^+ (PMOS source) – n-well – p-substrate
    - NPN: n^+ (NMOS source) – p-substrate – n-well
- These are **cross-coupled** forming a p-n-p-n structure like an SCR.

Latch-Up Phenomenon

- Under abnormal conditions (voltage spikes, ESD, input driven beyond rails):
    - Junctions between these regions can become **forward biased**.
    - One parasitic BJT turns ON and injects current into the base of the other.
    - This forms a **positive feedback loop**:

    - PNP drives NPN
    - NPN drives PNP

- Once triggered, the p-n-p-n structure conducts heavily between V_DD and GND → **large current**.
- Even after the triggering event is removed, the conduction **stays on** until supply is cut or device is damaged.

Consequences

- Overheating
- Permanent damage of device
- Wrong logic levels / malfunction

Q7 (b) How can latch-up be prevented?

Layout and Process Techniques

1. **Guard Rings:**
    - Surround NMOS regions with **p^+ guard rings** tied to GND.
    - Surround PMOS regions in n-well with **n^+ guard rings** tied to V_DD.
    - These rings collect carriers and prevent them from turning on parasitic BJTs.
2. **Well and Substrate Taps:**
    - Use **frequent contacts** to substrate (p^+ to GND) and well (n^+ to V_DD).
    - Reduce substrate and well resistances so voltage rise is small.
3. **Increased Spacing:**
    - Increase distance between n^+ and p^+ diffusions partly forming parasitic BJTs.
    - Reduces current gain and coupling between devices.
4. **Epi Substrate and Low-Resistivity Bulk:**
    - Use lightly doped epi on heavily doped substrate to reduce latch-up susceptibility.
5. **ESD and I/O Protection:**

    - Ensure no I/O pin goes much beyond supply rails.
    - Use **clamping diodes** and proper I/O structures.

Q8 Short notes (any two)

I’ll give all four so students can pick any two in exam.

(i) Algotronix

- Example of **early FPGA-based computing platform**.
- Shows concept of mapping algorithms directly into **programmable hardware** instead of writing software on CPU.
- Uses configurable logic blocks and interconnects.
- Demonstrates advantages of FPGAs for:
    - High-speed DSP
    - Cryptography
    - Rapid prototyping of digital systems

(ii) MOS Capacitance Model

- MOS transistor has multiple capacitances:
    - Gate–source (C_{gs})
    - Gate–drain (C_{gd})
    - Gate–bulk
    - Source/drain junction capacitances (C_{js}, C_{jd})
- Values depend on:
    - Operating region (cutoff, linear, saturation)
    - Bias voltages
- These capacitances determine:
    - **Propagation delay** of gates
    - **Dynamic power consumption**
- Accurate timing analysis uses these capacitances in RC delay models (e.g., Elmore delay).

(iii) Microcoded Controller

- A control unit implemented using **microprogramming**.
- Control signals are stored as **microinstructions** in a ROM/PLA.
- Each microinstruction word contains bits for:
  - ALU function
  - Register enables
  - MUX select lines, etc.
- A **microprogram counter** steps through microinstructions to implement each machine instruction.
- Advantages:
  - Easier to modify/extend instruction set (change microcode).
  - Good for complex CISC designs.

(iv) Systolic Array

- A regular 2D array of processing elements (PEs) that **compute and pass data rhythmically** like a heartbeat.
- Each PE:
  - Performs simple operations (e.g., multiply–accumulate).
  - Passes partial results to neighbors.
- Very efficient for:
  - Matrix multiplication
  - Convolution
  - Linear algebra in AI/ML
- Used in many modern AI accelerators and DSP chips.