ESE 570 Design for Manufacture
OVERVIEW

- Design Quality (Reminder)
- What is Design for Manufacturability?
- Modeling Process Variations
- Impact of Process Variations on Circuit Performance
- Parametric Yield Estimation
- Parametric Yield Optimization
DESIGN QUALITY

-> ACHIEVE SPECIFICATIONS (Static & Dynamic)
-> DIE SIZE
-> POWER DISSIPATION

-> TESTABLE
-> MANUFACTURABLE
-> RELIABLE
DESIGN QUALITY cont.

-> TESTABLE
  + generation of good test vectors
  + availability of reliable test fixture at speed
  + design of testable chip

-> MANUFACTURABLE
  + functional yield
  + parametric yield

-> RELIABLE
  + premature aging (Infant mortality)
  + ESD/EOS
  + latchup
  + on-chip noise and crosstalk
  + power and ground bouncing
WHAT IS DESIGN FOR MANUFACTURABILITY?

Design goal: All fabricated circuits meet all performance specs under all fab and operating conditions.

Impediments:
1. Random variations in fabrication process.
2. Random variations in operating conditions, e.g. $V_{DD}$, $T_{ambient}$.
3. Less than full chip testability (controllability and observability).
WHAT IS DESIGN FOR MANUFACTURABILITY?

DFM Practice:
1. Consider effects of random fabrication parameters and operating conditions early in the design process.
2. Design and layout done to reduce sensitivity to these variations.
   a. Design to performance specs with sufficient margins.
   b. Satisfy Design Rules with some margin.
3. Design for Testability.

DFM Metrics:
1. Functional yield.
2. Parametric yield.
4. Test fault coverage.
VLSI CAD Supports Design for Manufacturability

CATAGORIES OF CAD TOOLS

1. High Level Synthesis (HDLs)
2. Logic Synthesis
3. Circuit Optimization
   a. transistor sizing for min delays
   b. process variations
   c. statistical design
4. Layout
   a. floorplanning
   b. place & route
   c. module generation
   d. automatic cell placement and routing
5. Layout Extraction
6. Simulation (SPICE for circuit-level simulation)
7. Layout - Schematic Verification
8. Design Rule Check
RELATIONSHIP BETWEEN PROCESS/DEVICE PARAMETERS AND CIRCUIT/SYSTEM PERFORMANCE

- TIME, TEMP, IMPURITY DOSES
- LITHOGRAPHY
- ENVIRONMENT, OP CONDITIONS
- PROCESS PARAMETERS
- ENVIRONMENT, OP CONDITIONS
- CIRCUIT CHARACTERISTICS
- YIELD, COST
- ENVIRONMENT, OP CONDITIONS
- CIRCUIT CHARACTERISTICS
- SYSTEM (BOARD)
MODELING PROCESS VARIATIONS

\[ I_d = \mu C_{ox} \left( \frac{W}{L} \right) f(V_{DS}, V_{GS}, V_T) \]

\[ I_{d-actual} \neq I_{d-design} \text{ due to variations in } \mu, t_{ox}, W/L \text{ and } V_T \]

\[ I_{d-actual} \neq I_{d-design} \Rightarrow \text{variations in logic levels, delay times, power dissipation.} \]

TO MAKE CIRCUIT LESS SENSITIVE TO CIRCUIT/PROCESS PARAMETERS

1. Make critical devices larger, i.e. increase L and W, keeping W/L constant.

2. Careful orientation of key transistors in layout, e.g. to take advantage of oxide gradients to keep t_{ox} constant.

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  . \\
  . \\
  x_n
\end{bmatrix} = \begin{bmatrix}
  d_1 \\
  d_2 \\
  . \\
  . \\
  d_n
\end{bmatrix} + \begin{bmatrix}
  s_1 \\
  s_2 \\
  . \\
  . \\
  s_n
\end{bmatrix}
\]

\[ d_i = \text{designable or controllable parameter set} \]

\[ s_i = \text{random variation or noise parameter set} \]

\[ \bar{x} = \bar{d} + \bar{s} \]

actual = design + random \( \Delta \)
MODELING PROCESS VARIATIONS cont.

Internal rv’s -> Gaussian, correlated pdf
e.g. W, L, \( V_{T0} \), \( t_{ox} \), \( \mu \)

\[
f(s_i) = \frac{1}{\sigma_{si} \sqrt{2 \pi}} \exp\left[ -\frac{1}{2} \frac{s_i^2}{\sigma_{si}^2} \right]
\]

\( f(s_i) \)

For \( s_i^0 = 0 \)

\[
E\{s_i, s_j\} = \rho_{s_i s_j} \sigma_{s_i} \sigma_{s_j} \quad \neq 0 \quad \text{correlated} \quad s_i, s_j
\]

\[
\text{independent} \quad s_i, s_j
\]

\[
E\{\mathbf{s} \times s^t\} = Q = \begin{bmatrix}
\sigma_{s_1}^2 & \rho_{s_1 s_2} \sigma_{s_1} \sigma_{s_2} & \cdots & \rho_{s_1 s_n} \sigma_{s_1} \sigma_{s_n} \\
\rho_{s_2 s_1} \sigma_{s_2} \sigma_{s_1} & \sigma_{s_2}^2 & \cdots & \rho_{s_2 s_n} \sigma_{s_2} \sigma_{s_n} \\
\vdots & \vdots & \ddots & \vdots \\
\rho_{s_n s_1} \sigma_{s_n} \sigma_{s_1} & \rho_{s_n s_2} \sigma_{s_n} \sigma_{s_2} & \cdots & \sigma_{s_n}^2
\end{bmatrix}
\]

where \( -1 \leq \rho_{s_i s_j} \leq 1 \)

External rv’s -> Uniform, independent pdf
e.g. \( V_{DD} \), \( T \)

\[
f(s_i) = \begin{cases} 
\frac{1}{b - a} & \text{for } a \leq s_i \leq b \\
0 & \text{otherwise}
\end{cases}
\]

Invaluable ally to IC designers

Kenneth R. Laker, University of Pennsylvania, updated 27Apr15
IMPACT OF PROCESS VARIATIONS ON CIRCUIT PERFORMANCE MEASURES

Performance Measures

\[ \bar{r}(\bar{x}) = \bar{r}(\bar{d} + \bar{s}) \] actual performance

\[ \bar{r}^0(\bar{d}) = \bar{r}(\bar{d} + \bar{s}^0) \] nominal performance

\[ \bar{s}^0 \] mean values of \( \bar{s} \) usually \( \bar{s}^0 = 0 \)

\[ \bar{r}^0(\bar{d}) \] evaluated using Circuit Simulation

\[ \bar{r}(\bar{x}) \] evaluated using Monte Carlo Simulation

\[ r_1(x) : \quad V_{th} = \frac{V_{T0n} + \sqrt{\frac{1}{k_R}(V_{DD} + V_{T0p})}}{1 + \sqrt{\frac{1}{k_R}}} \quad k_R = \frac{k_n'(W/L)_n}{k_p(W/L)_p} \]

\[ r_2(x) : \quad \tau_p = \frac{\tau_{PHL} + \tau_{PLH}}{2} \]

or

\[ \tau_{PHL} = C_{load} \frac{2 V_{T0n}}{k_n (V_{DD} - V_{T0n})} \left[ \frac{2 V_{T0n}}{V_{DD} - V_{T0n}} + \ln \left( \frac{4(V_{DD} - V_{T0n})}{V_{DD}} - 1 \right) \right] \]

or

\[ r_2(x) : \quad \tau_{PLH} = C_{load} \frac{2 V_{T0p}}{k_p (V_{DD} + V_{T0p})} \left[ \frac{-2 V_{T0p}}{V_{DD} + V_{T0p}} + \ln \left( \frac{4(V_{DD} + V_{T0p})}{V_{DD}} - 1 \right) \right] \]
IMPACT OF PROCESS VARIATIONS ON CIRCUIT PERFORMANCE MEASURES cont.

\[ r = r(\bar{x}) = r(\bar{d} + \bar{s}) \]  
actual performance - random variable

\[ r^0(\bar{d}) = r(\bar{d} + \bar{s}^0) \]  
nominal or designed performance - deterministic variable

Each statistical circuit performance measure is illustrated using a histogram:

![Histogram of Propagation Delay](image)

\[ \text{parametric yield } (Y) = \frac{\text{total number of acceptable circuits}}{\text{total number of manufactured circuits}} \times 100\% \]
IMPACT OF PROCESS VARIATIONS ON CIRCUIT PERFORMANCE MEASURES cont.

\[ m_{\tau_p} = 0.184 \text{ ns} \]
\[ \sigma_{\tau_p} = 0.023 \text{ ns} \]

If spec for \( \tau_p \leq 0.19 \text{ ns} \): \( Y \approx 60\% \)
Some Real Defects in Chips

- Processing defects
  - Missing contact windows
  - Parasitic transistors
  - Oxide breakdown

- Material defects
  - Bulk defects (cracks, crystal imperfections)
  - Surface impurities (ion migration)

- Time-dependent failures (Age defects)
  - Dielectric breakdown
  - Electromigration

- Packaging failures
  - Contact degradation
  - Seal leaks
PARAMETRIC YIELD ESTIMATION

Acceptable Region - Performance Space
Consider a circuit where the performance measures were \( r_1 = \tau_p \) and \( r_2 = P_d \) and their specs

\[
\begin{align*}
\tau_p & \leq 0.16 \text{ nsec} \\
P_d & \leq 0.5 \text{ mW}
\end{align*}
\]

Acceptable Region
In Performance Space

\[ A_r = \{ r \mid a_k \leq r_k \leq b_k, \ k = 1, 2, \ldots, p \} \]

\( Y(\vec{r}) = \text{Probability}(\vec{r} \in A_r) \)

\( Y(\vec{r}) \) is a scalar, deterministic quantity that is difficult to evaluate. pdf's for \( r_k \) are usually not known specifically.
PARAMETRIC YIELD ESTIMATION cont.

Acceptable Region - Parameter Space

Actual values $(x_1, x_2)$ due to statistical variations about $\bar{d}$

$\bar{d} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$ = design point

Acceptable circuit parameters for the design point $\bar{d}$

$A_x = \{ \bar{x} \mid a_k \leq r_k(\bar{x}) \leq b_k, \; k = 1, 2, \ldots, p \ \text{and} \ \bar{x} \in \bar{X} \}$

where $A_x \subset \bar{X}$ where $\bar{X}$ - allowed circuit parameter space

Parametric yield $= Y(\bar{d}) = \text{Probability}(\bar{x} \in A_x) = \text{Probability}(\bar{d} + \bar{s} \in A_x)$

$Y$ is a scalar, deterministic quantity.
Monte Carlo sampling is the most often used method to estimate $Y$. 

Allowed circuit parameter values restricted to subset of circuit parameter space due to physical considerations.
PARAMETRIC YIELD MAXIMIZATION

1. Monte Carlo Based Methods
   \[ Y(\vec{d}) = \Pr(\vec{x} \in A_x) = \Pr(\vec{d} + \vec{s} \in A_x) \]
   
   a. Compute \( Y(\vec{d}) \) using Monte Carlo Simulation
   
   b. Determine \( \vec{d} \) to maximize \( Y(\vec{d}) \)

2. Design Centering Method
VLSI DFM and Manufacturing Process

1. Customer's need
2. Determine requirements
3. Write specifications
   - Design synthesis and Verification
4. Test development
5. Fabrication
   - Manufacturing test
6. Chips to customer

Test dies on wafer
Test packaged parts
Types of Manufacturing Tests

- **Characterization Testing**
  - Used to characterize devices and performed through production life to improve the process, hence yield

- **Production testing**
  - Factory testing of all manufactured chips for parametric faults and for random defects.
  
  - The test patterns may not cover all possible functions and data patterns but must have a high fault coverage of modeled faults.

  - The main driver is cost, since every device must be tested. Test time must be absolutely minimized.

  - Only a go/no-go decision is made.

- **Burn-In testing**
  - Ensure reliability of tested devices by testing.

  - Detect the devices with potential failures.
Testing Principle

Test patterns

---11
---01
......
---00
---10

Device Under Test (DUT)

Test responses

10---
00---
......
01---
10---

Stored
Correct
Responses

Comparator

Test result
EXAMPLE ADVANTEST Model T6682 ATE

Consists of

- Powerful computer
- Powerful DSP for analog testing
- Probe head: actually touches the bare dies or packaged chips to perform fault detection experiments
- Probe card: contains electronics to measure chip pin or pad
ATE Test Operation

**Diagram:**
- **Input Stimulus** connected to **Input Drivers**.
- **Pattern Memory** feeding **Expected Response**.
- **Compare Output** compares **Expected Response** with **Actual Response**.
- **DUT** (Device Under Test) receives **Input Stimulus** and produces the **Actual Response**.
- The **Pass/Fail** result is determined by comparing the **Expected Response** with the **Actual Response**.
- **Local Per-Pin Memory** stores the test results.
Two key factors are changing the way of VLSI ICs testing

- The manufacturing test cost has been not scaling
- The effort to generate tests has been growing geometrically along with product complexity

Cost: $/transistor

Si cost/transistor

Test cost/transistor

Source: SIA
Defect, Fault, and Error

- **Defect (imperfection in hardware):**
  - A defect in an electronic system is the unintended difference between the implemented hardware and its intended design.

- **Error:**
  - A wrong output signal produced by a defective system is called an error. An error is an “effect” whose cause is some “defect”.

- **Fault (imperfection in function):**
  - A representation of a “defect” at the abstracted function level is called a fault.
Observability & Controllability

- **Observability**: ease of observing a node by watching external output pins of the chip

- **Controllability**: ease of forcing a node to 0 or 1 by driving input pins of the chip

Combinational logic is usually easy to observe and control

Sequential logic or finite state machines can be very difficult, requiring many cycles to enter desired state

Good observability and controllability reduces number of test vectors required for manufacturing test.
  - Reduces the cost of testing
  - Motivates design-for-test
Common Fault Models

**Single stuck-at faults**

Transistor open and short faults

Memory faults

PLA faults (stuck-at, cross-point, bridging)

Functional faults (processors)

Delay faults (transition, path)
Test Process

- What faults to test (*fault modeling*)?
- How are test patterns obtained (*test pattern generation*)?
- How is test quality (fault coverage) measured (*fault simulation*)?
- How are test vectors applied and results evaluated (*ATE/BIST*)?
Three properties define a single stuck-at fault

Only one line is faulty

The faulty line is permanently set to 0 or 1

The fault can be at an input or output of a gate

Example: XOR circuit has 12 fault sites (●) and 24 single stuck-at faults
The Test Problem

Given a set of faults in the DUT, how do we obtain a certain (small) number of test patterns which guarantees a certain (high) fault coverage?

Fault coverage (FC)

- The measure of the ability of a test (a collection of test patterns) to detect faults that may occur on the DUT

\[
FC = \frac{\text{Number of detected faults}}{\text{Number of possible faults}}
\]
Determine the input pattern that exposes an s-a-0 fault occurring at node U at the output Z.

a. Choose an input pattern that sets $U = 1$, i.e. $A = B = 1$ - controllability.
b. The fault at $U$ needs to propagate to $Z$ to be observed, i.e. $Y = U$ iff $X = 1$ and $Z = U$ if $E = 0$.
c. $X = 1 \Rightarrow C = D = 1$.
d. The (unique) test vector can now be assembled: $A = B = C = D = 1$, $E = 0$. 
Design for Test

★ Design the chip to increase observability and controllability.

★ If each register could be observed and controlled, test problem reduces to testing combinational logic between registers.

★ Better yet, logic blocks could enter test mode where they generate test patterns and report the results automatically.
Scan Based Testing

Scan test is to obtain control and observability for FFs.
- It reduces *sequential Test Pattern Generation circuits* (TPG) to *combinational* TPG circuits.

- With Scan, a synchronous sequential circuit works in two modes:
  - Normal mode and Test mode:

In test mode, all FFs are configured as a shift register, with Scan-in and Scan-out routed to a (possibly dedicated) Pin-IN (PI) and Pin-OUT (PO).
Scan Based Testing

- Convert each flip-flop to a scan register
  - Costs one extra multiplexer

- Normal mode: flip-flops behave as usual

- Scan mode: flip-flops behave as shift register

- Contents of flops can be scanned out and new values scanned in
Boundary Scan for Board Test

Boundary-scan cell

Boundary-scan chain

Serial Data in

Serial Data out

Internal Logic

Internal Logic

Internal Logic

Internal Logic

System interconnect
Some Important Boundary Standards

- Digital Boundary Scan (IEEE 1149.1)
- Analog Boundary Scan (IEEE 1149.4)
- Boundary Scan for Advanced Networks (IEEE 1149.6)
- Embedded Core Test Standard (IEEE 1500)