Introduction to VLSI Design, June 2004 – Answers to Questions 1..4

1. a. *Explain why manufacturing test is so important for ICs and what is meant by design for test*

Answer should include:

Manufacturing process is error-prone and complex and manufacturers will routinely work at the edge of what is possible – trading yield off for marketing advantage

Systems are complex (50M+ transistors) and individual faults may give rise to failure

Systems have very low observability – that is it is difficult to decide what is going on in the IC merely by looking at the pins

Design for test is the methodology that makes ICs testable. It involves:

considering the needs of test at all points in the design flow.

Designing a test architecture in as part of the overall design's methodology *that is compatible with the company's testing infrastructure*.

Adhering to a well-defined synchronous methodology that will allow the automatic generation of test structures (scan chains, BIST) and test vectors. **(3)**

b. *Identify the different types of fault that might be encountered in a manufactured IC and identify the type of fault that is predominantly assumed to occur in testing ICs.*

Stuck-at faults – where a logic level does not change

Open faults – where a wire is disconnected

Short faults – where two unconnected wires are shorted together

Iddq faults – where the leakage current of the transistor is higher than expected

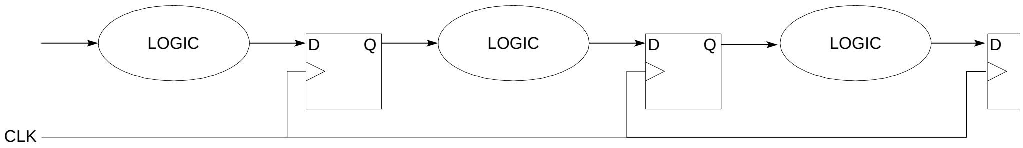
delay faults – where the propagation delay along a particular path is different (usually slower) than expected.

Predominantly, stuck-at faults are used because they cover ~95% of the cases and other faults can be modelled by combinations of stuck-at faults. **(3)**

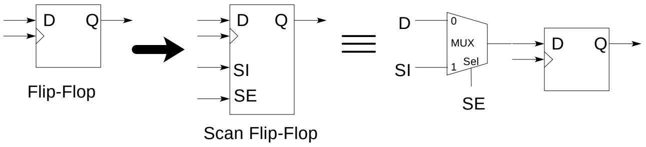
c. *Describe the organisation and operation of the following:*

i) *scan testing*

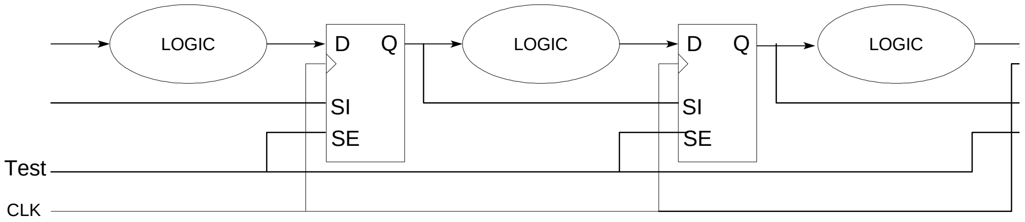
Looking at the structure of *normal*, sequential circuits:



The circuit appears to be stages of logic separated by stages of data storage in flip-flops. All of the flip-flops are driven by the same clock (in practice, the clock is arranged as a tree of buffers so that each flip-flop is clocked at nearly the same instant) and, in this way, the behaviour of the circuit is reliable and deterministic. Normally, given the sequential nature of circuits, it would be very time consuming (or impossible) to get the registers in the IC into a state to allow a particular test of the combinatorial logic to be undertaken. However, if the flip-flops could be loaded independently during testing then they could be set to any arbitrary state before each test allowing, in effect, the problem of testing a sequential circuit to be changed to testing a combinatorial circuit with prior initialisation. To do this, the flip-flops are replaced by scan flip-flops:



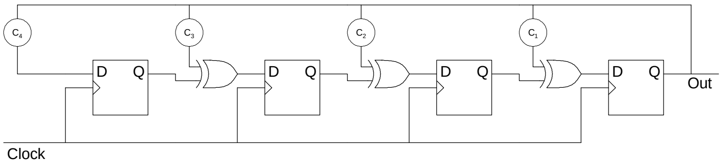
If SE (Scan Enable) is 0, then the input to the flip-flop is derived from the normal D input. However, if SE is 1 (as it would be during test), then the input to the flip-flop comes from the Serial Input (SI) and this is used to set up the state of the flip-flops before each test.



The output of a flip-flop is connected to the SI of another flip-flop to form a long shift-register - a scan chain. Data can be serially loaded into the chain of flip-flops to set their state to the input values needed to test each block of combinatorial logic and then, with Test disabled, the circuit is clocked once which stores the output of the combinatorial logic into the flip-flops. Test is enabled again and the output values can be unloaded (for comparison with their expected values) using the scan chain whilst the next test inputs are loaded in. This sort of scheme is a trade-off: it still takes a number of clock cycles to set up a test (equal to the number of flip-flops in a scan chain) but the external overhead is very low: only 3 pins are required - Test, Scan-In, and Scan-Out. **(3)**

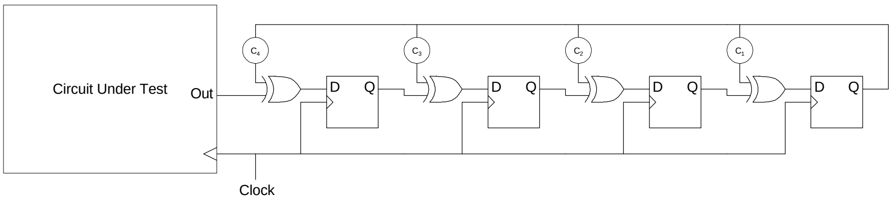
ii) *Built-in Self-Test*

BIST is based upon a linear-feedback shift register (a 4 bit example is shown here):



The values C_1 to C_4 (which are either 1 or 0) determine the pattern of values in the registers (and at the output) and there are certain combinations C_1 to C_4 that of give rise to *maximal-length sequences*. That is, every combination of value appears in the register (except all 0s) in a well-defined but seemingly random sequence that repeats. Indeed, Linear-Feedback Shift Registers (LFSR) are often used to generate pseudo-random sequences of bits or numbers - hence the name Pseudo-Random Binary Sequence (PRBS) generator. The properties of PRBSs are well-known and have been used, amongst other things, for cryptography.

In BIST, the LFSR is augmented as follows:



An output from the circuit being tested is combined with one of the feedback paths in the LFSR (the inputs to the circuit being tested are provided by other LFSRs). During BIST, the circuit and the LFSRs are initialised to defined states and a defined number of clocks are applied. As the circuit and LFSRs are clocked, the state of the inputs and, hence, output changes in a way determined by the function of the circuit and the state of the LFSR connected to the output changes *influenced by the output from the circuit being tested.* At the end of the test, the state of this LFSR, the signature, should be well-defined but seemingly random value. If the sequence of outputs were different - even in a single value - the signature in this LFSR would be completely different. Comparing the value in the LFSR at the end of the test gives confidence that the circuit is performing properly. This is the same as the cyclic redundancy check (CRC) that is used to error-check communication channels. The LFSRs can be very long to ensure that the probability of the same value occurring for correct and incorrect operation is very low. These LFSRs can be augmented to deal with multiple inputs (Multiple Input Shift Registers – MISR).

(3)

In design terms, the MISRs can be merged with existing register with a normal function and a function that is invoked when testing is enabled.

d. *The logic shown in **Figure 1** is to be tested for a fault at node X. Describe how this would be done. Ensure that you identify sensitisation and propagation in your explanation and any changes to the logic that you might need to make to allow node X to be tested.*

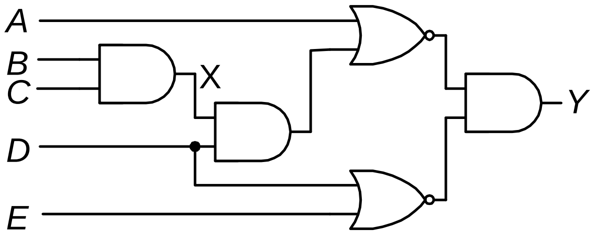


Figure 1

The fault-modelling approach would postulate a particular stuck-at fault for X. That is, stuck-at-1 for example. Sensitisation would be where a set of inputs would be applied to force node X to the opposite state, 0. Propagation would be where the inputs applied would allow the state of X to be propagated to the output allowing the observer to distinguish between the 1 and 0 state at X.

In the case in Figure 1, to force X to a 0, then either B or C must be 0. To propagate X to the output, D must be 1 (allow X to propagate through the AND gate) and A must be 0. The other input to the last AND gate (derived from D and E) must be 1 (in which case Y = Xbar). However, to do this both D and E must be 0 and this incompatible with the other requirement for D. This problem would be alleviated by adding some test logic that would be interposed between D and the NOR gate so that this situation could be avoided.

(8)

2. **a.** *Describe the attributes of CMOS logic that make it the dominant logic style used in digital ICs.*

Answer should include:

good performance and density

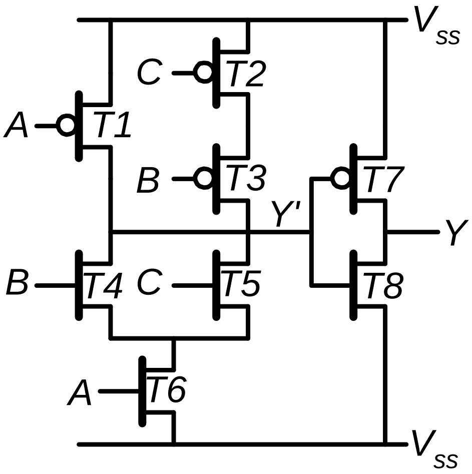
power performance scales with frequency and is (relatively) low in quiescent mode

scales (reasonably) well as the technology shrinks **(4)**

b. *For the function,* $Y = \overline{A.\overline{B+C} + \overline{A}}$ *:*

i) *Draw a CMOS circuit that minimises the number of transistors used;*

The function can be transformed into $Y = A.(B+C)$ and it should be noted that this is not a naturally inverting function. Consequently, let us work out $Y' = \overline{Y} = \overline{A.(B+C)}$. This translates into the following circuit:



(6)

ii) *Size the transistors to form a minimum-sized gate (you may assume that the mobility of holes is 0.5 that of electrons);*

Making the assumption that we want the resistance between an output node and a supply voltage to be the same as a minimum-sized n-type transistor (which has a width of 1).

T1 = 2, T2 = 4, T3 = 4, T4 = 2, T5 = 2, T6 = 2, T7 = 2, T8 = 1 **(2)**

iii) *Estimate, simply, the delay of the circuit relative to a minimum-sized inverter.*

Looking at the delay of the inverter, assuming that the capacitance of a minimum sized n-type transistor is *C* and that the capacitance at the drain of a transistor is ½ of the gate capacitance, the capacitance at *Y* will be ½(*C*+2*C*) and, to a first order the delay is ∝1.5*CR* where *R* is the effective on-state resistance of a minimum sized n-type transistor (any path from a node to the supply voltage). The capacitance at *Y'* is 0.5(2*C*+4*C*+2*C*+2*C)*+*C*+2*C* = 8*C*. That is the delay is ∝*8CR*. Consequently, the a simple estimate of the delay is that it 5.3x a simple inverter. **(4)**

c. *The circuit in part* ***b.*** *is to be used to drive a load whose capacitance is equivalent to 44 times the capacitance of a minimum-sized n-type transistor. How would you alter the circuit to drive this load effectively?* **(4)**

Driving a large load is done with tapered buffers where each buffer in series increases exponentially in size. The number of buffers is ln(C_{load}/C_{in}). Looking at each of the inputs, *A* is looking into 4*C*, whilst *B* and *C* look into 6*C*. Taking this latter figure as C_{in}, the number of buffers is ln(44/6)≈2. With this gate, this can be done by re-sizing the inverter on the end – increasing the width of the transistors by e^1.

3. a. *The drain (V_d) of the n-type pass transistor in **Figure 3** is driven from a perfect voltage source. Estimate, the small-signal, quiescent, on-state resistance of the FET for:*

i) $V_d = V_{dd}$

V_g is high and so the transistor conducts and, because V_d is the driven node, V_s should follow V_d. Unfortunately, this is not the case.

When V_d is driven to V_{DD}, the voltage on the drain and the gate of the transistor are the same (V_g=V_{DD}) and the transistor must be saturated. Therefore,

$$I_{DS} = \frac{b_N}{2}\cdot(V_{GS} - V_T)^2$$

It appears that I_{DS} is not dependent on V_{DS} but, clearly, V_{GS}=V_{DS} and so:

$$I_{DS} = \frac{b_N}{2}\cdot(V_{DS} - V_T)^2$$

$$\frac{d}{dV_{DS}} I_{DS} = b_N \cdot (V_{DS} - V_T) = \frac{1}{r}$$

In this case, as the output voltage at V_s rises, the effective resistance of the FET increases until, V_s=V_{DD}-V_T. At this point:

$$r = \frac{1}{b_N \cdot (V_{DS} - V_T)} = \frac{1}{b_N \cdot (V_T - V_T)} = \infty$$

r is infinite and I_{DS}=0.

(3)

ii) $V_d = 0$

In this case, V_{GS}-V_T>V_{DS} and so the transistor is in ohmic mode. Consider the drain current for the transistor:

$$I_{DS} = b_N \cdot \left(V_{GS} - V_T - \frac{V_{DS}}{2}\right)\cdot V_{DS}$$

By differentiating the current, the small signal resistance between V_d and V_s, *r*, can be found.

$$\frac{d}{dV_{DS}} I_{DS} = \frac{1}{r} = b_N \cdot (V_{GS} - V_T - V_{DS})$$

$$r = \frac{1}{b_N \cdot (V_{GS} - V_T - V_{DS})}$$

V_{GS} = V_{DD}. and V_{DS}=0. In this case the drain current is 0 but *r* is:

$$r = \frac{1}{b_N \cdot (V_{DD} - V_T)}$$

(3)

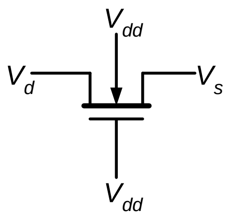


Figure 3

b. *The point labelled V_S in **Figure 3** is loaded with a capacitance of C_L that, initially, has 0V on it. At t=0, the voltage V_d is raised from 0V to V_{dd}. Determine the voltage, V_S, as a function of t.*

When V_d is pulled high, assuming that it is 0 initially, *A* acts as the drain connection and V_{DS}=V_{DD}-V_s (where V_s is the voltage on the capacitor) and V_g=V_{DD}-V_s. That is, they are equal and, whilst the transistor conducts, it is saturated. in this case:

$$I_{DS} = \frac{b_N}{2} \cdot (V_{DD} - V_T - V_s)^2 = C_L \frac{d}{dt} V_s$$

Integrating both sides:

$$\int \frac{b_N}{2C_L} dt = \int \frac{dV_Y}{(V_{DD} - V_T - V_s)^2}$$

Yielding:

$$\frac{b_N}{2C_L} t + K = \frac{1}{(V_{DD} - V_T - V_s)}$$

The initial condition, at t=0, V_s=0, can be used to solve for K:

$$K = \frac{1}{(V_{DD} - V_T)}$$

And, again the voltage at V_s can be found:

$$\frac{b_N}{2C_L} t + \frac{1}{V_{DD} - V_T} = \frac{1}{(V_{DD} - V_T - V_s)}$$

$$V_s = V_{DD} - V_T - \frac{2C_L (V_{DD} - V_T)}{b_N (V_{DD} - V_T)t + 2C_L}$$

$$V_s = \frac{b_N (V_{DD} - V_T)^2 t + 2C_L (V_{DD} - V_T) - 2C_L (V_{DD} - V_T)}{b_N (V_{DD} - V_T)t + 2C_L} = \frac{b_N (V_{DD} - V_T)^2 t}{b_N (V_{DD} - V_T)t + 2C_L}$$

$$V_s = \frac{\frac{b_N}{2C_L} (V_{DD} - V_T)^2 t}{1 + \frac{b_N}{2C_L} (V_{DD} - V_T)t}$$

(8)

c. *You are told that, especially for small values of t, the equation:*

$$\frac{XY^2 t}{1 + XYt} \approx Y\left(1 - e^{-XYt/1.5}\right)$$

*is satisfied. From this fact, and the answer in part **b.**, estimate, approximately, the effective resistance of the FET in this situation.*

This is clearly not an exponential but referring to the above expression, X=b_N/2C_L, and Y=(V_{DD}-V_T) and based on this:

$$V_s = (V_{DD} - V_T)\left(1 - e^{-\frac{b_N(V_{DD}-V_T)}{3C_L}t}\right)$$

That is the effective on-state resistance of the FET in this case is approximately

$$R_O = \frac{3}{b_N(V_{DD} - V_T)}$$

(6)

4. **a.** *Show that the dynamic power consumption of a CMOS IC, due to switched capacitance, can be modelled by:*

$$P_{sw} = af_{clk}V_{DD}^2\sum_{i=1}^{n}C_i$$

where the terms have their usual meaning.

Consider charging a capacitor, C to V_{DD}. The charge on the capacitor will be CV_{DD}. As the capacitor is discharged to 0V, this charge will flow down to earth. The net charge moved through V_{DD} is, therefore, CV_{DD}. If this operation is being done f times a second then the charge moved per second is fCV_{DD} and this is current, axiomatically. This current flows across V_{DD} and so the power dissipated by this switching activity, P_{sw}, is fCV_{DD}^2. Where f is the frequency at which the input is being driven.

If we are to extend this expression from a single gate to an entire circuit we must perform a summation across all of the gates and interconnect in the circuit. So if we assume that there are n wires in the design and the total capacitance associated with wire$_i$ and the load that it is driving is C_i then the total switching power dissipated by the circuit should be:

$$P_{sw} = \sum_{i=1}^{n} f_iC_iV_{DD}^2$$

This expression assumes that each wire is switching at its own frequency, f_i. However, in practice, the switching of all the wires will be controlled by a single frequency f_{clk} and each wire will change state of the rising clock edge with a defined probability a_i. In this case, the expression becomes:

$$P_{sw} = f_{clk}V_{DD}^2\sum_{i=1}^{n}a_iC_i$$

In many cases, it is possible to simplify the expression further by assuming a value for a that is representative for the whole circuit rather than an individual wire.

$$P_{sw} = af_{clk}V_{DD}^2\sum_{i=1}^{n}C_i$$

What gives rise to C_i? **(4)**

In terms of a circuit, C_i=C_{in}+C_{wire} (the sum of the gate's input capacitance and the capacitance of the wire driving the gate input). **(2)**

b. *How does the leakage power consumption of a CMOS circuit arise and, in the future, what will the factors be that will affect this leakage power consumption.*

Leakage current arises from a number of causes: sub-threshold conduction of the transistors between the drain and the source, between the drain/source and the substrate. Additionally, current that tunnels through the gate is a source of leakage. In the future, as technology scales, V_T will reduce, resulting in increased sub-threshold conduction and, importantly, the oxide will thin, resulting in a greatly increased tunnelling current. **(2)**

c. *Give an example of a technique that can be used to reduce power consumption at each of the following levels of circuit/system design – identifying how the reduction in power consumption comes about.*

i) *Architecture*

Let us consider a regular algorithm that can be parallelised (at the expense of area, of course). The specification for the algorithm might demand a certain throughput of data. An implementation based around a single processing block will be required to meet this performance specification in full, an implementation based around *n* processing blocks in parallel still needs to meet the performance specification but each block needs to run *n* times more slowly. Given that each block does not need to run as quickly the power supply can be reduced. If we assume that the speed of the logic is related to effective on-state resistance of the FET and that the delay can be increased by a factor of *n*.

$$\frac{n}{V_{DD}-V_T}=\frac{1}{V_{DDR}-V_T}$$

$$\frac{V_{DD}+V_T(n-1)}{n}=V_{DDR}\approx\frac{V_{DD}}{n}+V_T$$

The power consumption of the system with the reduced power supply voltage, V_{DDR}, will be as a result of *n* blocks operating at this reduced supply voltage:

If the power consumption of the non-parallel design is kV_{DD}^2 then the consumption of the parallel design would be nkV_{DDR}^2 and if we neglect V_T then the power consumption will be reduced by a factor of *n*. Obviously, the area has gone up by a factor of *n* and the area/time product (a commonly used figure of merit) will be unchanged. **(4)**

ii) *Logic*

At the logical level there is a variety of things that can be done. For example, consider the way in which data is coded in an arithmetic circuit. If a data word is coded as 2’s complement then what happens when a small +ve numbers on a data bus alternate with small –ve numbers. For example:

0000000000010011	19
1111111111111010	-6

Due to the coding, the upper bits alternative between 0s and 1s (consuming power) even though they do not carry any real information. Data can be coded in other ways – sign/magnitude, for example. In this scheme, the most significant bit carries the sign information whilst the remaining bits are unsigned. The example above would be encoded as:

0000000000010011	19
1000000000000110	-6

In this example the number of state changes when the data is coded in 2’s complement is 13 whilst for the sign/magnitude encoding there are only 4 transitions. **(4)**

d. *A 2-input, minimum-sized NAND gate is driven by two random, unrelated input signals. These signals are derived from a circuit that is clocked at 400MHz. It is estimated that the probability of each input signal changing state is 0.25. The NAND gate is driving a load capacitance of 10fF. Estimate the power consumption associated with the NAND gate and the load that it drives (you may* **(4)**

assume that other interconnect capacitance can be neglected).

(For the CMOS process: m_E*=0.08*m^2*/Vs,* m_H*=0.04*m^2*/Vs,* t_{OX}*=10nm,* W_{min}*=1mm,* L_{min}*=0.25mm,* e_0e_r*=3.45x*10^{-11}*F/m,* V_{DD}*=3.3V,* V_{TN}*=-*V_{TP}*=0.6V)*

A minimum-sized n-type transistor has a gate capacitance of $3.45\text{x}10^{-11}\text{x } 1\text{x}10^{-6}\text{x } 0.25\text{x}10^{-6}/\ 10\text{x}10^{-9}$=0.863fF. The NAND gate has two stacked n-type transistors that will be $2W_{min}$ wide and two parallel p-type transistors of the same width. The gate capacitance for each of these will be 1.726fF. Consequently, the total gate capacitance driven by each input will be 3.45fF. If the inputs are random then the probability of a transition of an input is 0.5. This means that the effective rate at which a signal would transition 0→1→0 would be 0.25*f*. Consequently, the power consumption associated with each of the inputs will be:

$0.25\text{x}400\text{x}10^{6}\text{x}3.45\text{x}10^{-15}\text{x}3.3^{2}=3.76\mu\text{W}$

Therefore, the power consumption associated with both inputs will be 7.52µW.

Assuming that the state of the output is, essentially, random, the probability of a transition from 1→0 is 0.25 and the probability of a transition from 0→1 is 0.75. Consequently, the probability of a 0→1→0 transition is 0.1875. The capacitance associated with the output node will be $0.5(2C_{Gmin}+2C_{Gmin}+2C_{Gmin})$+10fF =3x0.863fF+10fF =12.59fF. Therefore, the power consumption will be:

$0.1875\text{x}400\text{x}10^{6}\text{x}12.59\text{x}10^{-15}\text{x}3.3^{2}=10.26\mu\text{W}$

Consequently, the total dynamic power consumption due to switched capacitance will be 17.77µW.

NLS/MB