UNIT III COMBINATIONAL AND SEQUENTIAL CIRCUIT DESIGN

Part –A (2 Marks)

1. What is a BiCMOS?
   BiCMOS is a type of integrated circuit that uses both bipolar and CMOS technologies.

2. What are the problems of BiCMOS?
   - Fabrication is difficult.
   - Difficult to tune both bipolar and MOS components.

3. What is pull down device? [AUC NOV 2013]
   A device connected so as to pull the output voltage to the lower supply voltage usually 0V is called pull down device.

   A device connected so as to pull the output voltage to the upper supply voltage usually VDD is called pull up device.

5. What are the types of gate arrays in ASIC? [AUC NOV 2013]
   Channeled gate arrays, channel less gate arrays, Structured gate arrays.

6. What is LEF mean?
   LEF is an ASCII data format used to describe a standard cell library.
   LEF file contains technology and site extension.

7. What is DEF mean?
   DEF is an ASCII data format to describe design related information.

8. What are the various design changes you do to meet design power targets?
   - Level shifters for performance improvement.
   - Reducing leakage power by designing multi threshold voltage areas.
   - By employing clock gating cells, power saving can be achieved.

9. List the drawbacks of ratioed circuits. [AUC MAY 2011]
   - Slow rising transitions
   - High static power dissipation.

10. Define max-delay failure and min-delay failure in sequential circuits. [AUC MAY 2011]
    If the combinational logic delay is too large, the receiving element will miss the setup time and sample improper values. This is called max-delay failure.
When the hold time is large and contamination delay is small, the data incorrectly propagates through two successive elements on one clock edge, corrupting the state of the system. This is called minimum delay failure.

11. Draw the CMOS bistable element. [AUC MAY 2011]

![CMOS bistable element diagram]

12. Write a note on CMOS transmission gate logic [AUC APR 2011]

![CMOS transmission gate logic diagram]


Bubble pushing is a technique which is applied for static CMOS structure that is dual in nature that can be obtained using duality principle of De Morgan’s theorem.

14. What is dynamic CMOS logic?

Dynamic circuits rely on the temporary storage of signal values on the capacitance of high impedance nodes.

15. What is precharge phase?

The input to the next stage is charged up through the PMOS transistor when the clock is low, this phase of the clock is known as precharge phase.

16. What is evaluation phase?

When the clock is high, the PMOS is cutoff and the NMOS is turned on thereby disconnecting the output node from Vdd and providing a possible pull down path to ground through the NMOS transistor. This is called as evaluation phase.
17. What is called sequencing overhead?
   Sequencing elements delay tokens that arrive too early, preventing them from catching up with previous token. This creates additional delay that reduces performance of the system. This extra delay is called sequencing overhead.

18. Define contamination delay.
   It is the required interval between invalid input and invalid output is called contamination delay.

   It is the required interval to change the output and after applying the input signal is called propagation delay.

20. Define setup time. [AUC APR 2008]
   The time before the clock edge that the D input has to be stable is called setup time.

21. Define hold time .
   The time after the clock edge that the D input has to be stable is called hold time.

22. What is clock skew ?[AUC NOV 2013]
   Delay in arrival of clock signals that cuts the time available for useful compilation is called clock skew.

23. What is pulsed latch?
   Pulsed latch consists of a set of pass gates that keeps the value stored in PL output.

24. What is klass semidynamic flip flop ?
   It is domino style flip flop used to reduce load on the data network.

25. What is synchronizer ?
   A synchronizer is a circuit that accepts an input that can change at arbitrary times and produces an output aligned to the synchronizers clock.

26. What is arbiter?
   An arbiter is a circuit designed to determine which of several signals arrive first.
PART -B(16 MARKS)

1. Explain briefly about static CMOS Design [AUC NOV 2010, APR 2012]

CMOS circuit falls under the category of

- Static CMOS
- Dynamic CMOS

Static CMOS design is a combination of two networks

- Pull-up network
- Pull-Down network

At every point of time, each gate output is connected to either VDD or VSS via a low resistance path.

Static CMOS logic gate

Important properties

- At any instant of time, the output is connected to either VDD or VSS.
- All functions are composed of either AND ‘ed or OR ‘ed sub functions.
- AND function composed of PMOS transistor in parallel (Pull –Down network).
- OR function composed of PMOS transistor in series (Pull –UP network).
- The function of PUN – provide a connection between the output and VDD , when the output of logic gate is ‘1’.
- The function of PDN – provide a connection between the output and Vss , when the output of logic gate is ‘0’.

Bubble Pushing
• Static CMOS structures are dual networks which can be obtained by using duality theorem.

\[ AB = A + B \]

These expressions can also be written as

\[ A + B = \overline{AB} \]
\[ AB = \overline{A + B} \]

2. **Discuss in detail about the ratioed circuit and dynamic circuit CMOS logic configurations. [AUC MAY 2011]**

Ratioed logic attempts to reduce the number of transistors required to implement a given logic function, often at the cost of reduced robustness and extra power dissipation. In ratioed logic, a gate consists of an NMOS pull-down network that realizes the logic function and a simple load device, which replaces the entire pull-up network (Figure 8.6(a)). A ratioed logic which uses a grounded PMOS load is referred to as a pseudo-NMOS gate (Figure 8.6(b)).
The advantage of pseudo-NMOS gate is the reduced number of transistors \((N + 1)\) versus \(2N\) for complementary CMOS. The nominal high output voltage \((V_{OH})\) for this gate is \(V_{DD}\), but the nominal low output voltage \((V_{OL})\) is not \(0\) V. This results in reduced noise margins and more static power dissipation. Since the voltage swing on the output and the overall functionality of the gate depend on the NMOS and PMOS sizes, the circuit is called ratioed. This is in contrast to the ratioless logic such as complementary CMOS, where the low and high levels do not depend on transistor sizes.

The static power dissipation of pseudo-NMOS limits its use. When area is most important, however, its reduced transistor count compared with complementary CMOS is quite attractive. Pseudo-NMOS thus still finds occasional use in large fan-in circuits.

**Differential Cascade Voltage Switch Logic (DCVSL).** This logic family combines two concepts: differential logic and positive feedback. A differential gate requires that each input is provided in complementary format, and it produces complementary outputs in turn. The feedback mechanism ensures that the logic device is turned off when not needed. This logic completely eliminates static currents and provides rail-to-rail swing (Figure 8.7).

![Diagram of DCVSL](image)

The pull-down networks PDN1 and PDN2 use NMOS devices and are mutually exclusive, i.e., when PDN1 conducts, PDN2 is off, and when PDN1 is off, PDN2 conducts—such that the required logic function and its inverse are simultaneously implemented. The resulting circuit exhibits a rail-to-rail swing, and the static power dissipation is eliminated. The circuit is still ratioed since the sizing of the PMOS devices relative to the pull-down devices is critical to functionality, not just performance. In addition to the problem of increased design complexity, this circuit style has a power-dissipation problem due to crossover currents. During the transition, there is a period of time when PMOS and PDN are turned ON simultaneously, producing a short-circuit path.
3. Discuss in detail the characteristic of CMOS Pass – Transistor Logic.

A popular and widely used alternative to complementary CMOS is pass-transistor logic, which attempts to reduce the number of transistors required to implement logic by allowing the primary inputs to drive gate terminals as well as source-drain terminals. Figure 8.8 shows an implementation of the AND function constructed that way, using only NMOS transistors. In this gate, if the B input is high, the top transistor is turned ON and copies the input A to the output F. When B is low, the bottom pass-transistor is turned ON and passes a 0. The promise of this approach is that fewer transistors are required to implement a given function. The reduced number of devices has the additional advantage of lower capacitance.

![Diagram of a pass-transistor implementation of an AND gate.]

The pass-transistor gates cannot be cascaded by connecting the output of a pass gate to the gate input of another pass-transistor. Also, it is observed that a pure pass-transistor is not regenerative. A gradual signal degradation will be observed after passing through a number of subsequent stages.

While the circuit exhibits lower switching power, it may also consume static power when the output is high—the reduced voltage level may be insufficient to turn OFF the PMOS transistor of the subsequent CMOS inverter.

**Differential pass-transistor logic.** For high performance design, a differential pass-transistor logic family, called complementary pass-transistor logic (CPL) or DPL is commonly used. It accepts true and complementary inputs and produces true and complementary outputs. Several CPL gates (AND/NAND, OR/NOR, and XOR/XNOR) are shown in Figure 8.9.

![Diagram of CPL gates: AND/NAND, OR/NOR, XOR/XNOR.]

The CPL gates possess some interesting properties:

(i) Since the circuits are differential, complementary data inputs and outputs are always available. Thus, some complex gates such as XORs and adders can be realized efficiently with smaller number of transistors. Furthermore, the availability of both polarities of every signal eliminates the need for extra inverters.
(ii) CPL belongs to the class of static gates, because the output-defining nodes are always connected to either $V_{DD}$ or GND through a low-resistance path.

(iii) The design is very simple. More complex gates can be built by cascading the standard pass-transistor modules.

Unfortunately, differential pass-transistor logic suffers from static power dissipation and reduced noise margins, since the high input to the signal-restoring inverter only charges up to $V_{DD} - V_{Tn}$. This problem can be dealt with several solutions by the use of level restorer, multiple-threshold transistors and/or by the use of transmission gates.

4. **Explain briefly about Dynamic CMOS Design [AUC May 2011]**

Dynamic logic clearly can result in high-performance solutions compared to static. However, there are several important considerations that must be taken into account for dynamic circuits to function properly. These include charge leakage, charge sharing, capacitive coupling, and clock feedthrough.

*Charge leakage*. The operation of a dynamic gate depends on the dynamic storage of the output value on a capacitor. If the pull-down network is OFF, ideally, the output should remain at the precharged state of $V_{DD}$, during the *evaluation* phase. However, this charge gradually leaks away due to leakage currents finally resulting in a malfunctioning of the gate. Figure 8.11(a) shows the sources of leakage for the basic dynamic inverter circuit.
The basic construction of an \((n\text{-type})\) dynamic logic gate is shown in Figure 8.10. The PDN is constructed exactly as in complementary CMOS. The operation of this circuit is divided into two phases—precharge and evaluation—with the mode of operation determined by the clock signal, Clk.

![Dynamic Logic Gate Diagram](image)

The main advantages of dynamic logic are increased speed and reduced implementation area. Fewer devices to implement a given logic function implies that the overall load capacitance is much smaller. After the precharge phase, the output is high. For a low input signal, no additional switching occurs. As a result, \(t_{pLH} = 0\). On the other hand, the high-to-low transition requires the discharging of the output capacitance through the pull-down network. Therefore, \(t_{pHL}\) is proportional to \(C_L\) and the current-sinking capabilities of the pull-down network. The presence of evaluation transistor slows the gate somewhat, as it presents an extra series resistance. Omitting this transistor, while functionally not forbidden, may result in static power dissipation and potentially a performance loss.

The dynamic logic presents a significant advantage from a power perspective, due to the following reasons:

- First, the physical capacitance is lower since dynamic logic uses fewer transistor to implement a given function. Also, the load seen for each fan-out is one transistor instead of two.
- Second, by construction, the dynamic logic gates can have at most one transistor per clock cycle. Glitching (or dynamic hazards) does not occur in dynamic logic.
- Finally, dynamic gates do not exhibit short-circuit power since the pull-up path is not turned ON when the gate is evaluating.
5. Describe briefly about Signal Integrity Issues in Dynamic Design.

Sources 1 and 2 are the reverse-biased diode and subthreshold leakage of the NMOS pull-down device $M_1$, respectively. The charge stored on $C_L$ will slowly leak away through these leakage channels, causing a degradation in the high level [Figure 8.11(b)]. Therefore, dynamic circuits require a minimal clock rate, which is typically of the order of a few kHz. This makes the usage of dynamic techniques unattractive for low-performance products such as watches or processors that use conditional clocks. The PMOS precharge device also contributes some leakage current due to the reverse bias diode (source 3) and the subthreshold conduction (source 4). To some extent, the leakage current of the PMOS counteracts the leakage of the pull-down path. As a result, the output voltage is going to be set by the resistive divider composed of the pull-down and pull-up path.

Leakage is caused by the high-impedance state of the output node during the evaluate mode, when the pull-down path is turned OFF. The leakage problem may be counteracted by reducing the output impedance on the output node during evaluation. This often is done by adding a bleeder transistor as shown in Figure 8.12(a). The function of the bleeder is to compensate for the charge lost due to the pull-down leakage paths. To avoid the ratio problems associated with this style of circuit and the associated static power consumption, the bleeder resistance is made high. This allows the (strong) pull-down devices to lower the $Out$ node substantially below the switching threshold of the next gate. Often, the bleeder is implemented in a feedback configuration to eliminate the static power dissipation altogether [Figure 8.12(b)].

*Charge sharing.* Another important concern in dynamic logic is the impact of charge sharing. Consider the circuit in Figure 8.13. During the precharge phase, the output node is precharged to $V_{DD}$. Assume that all inputs are set to 0 during precharge, and that the capacitance $C_a$ is
discharged. Assume further that input B remains at 0 during evaluation, while input A makes a 0 to 1 transition, turning transistor $M_a$ ON. The charge stored originally on capacitor $C_L$ is redistributed over $C_L$ and $C_a$. This causes a drop in the output voltage, which cannot be recovered due to the dynamic nature of the circuit.

The most common and effective approach to deal with the charge redistribution is to also precharge critical internal nodes, as shown in Figure 8.14. Since the internal nodes are charged to $V_{DD}$ during precharge, charge sharing does not occur. This solution comes at the cost of increased area and capacitance.

![Circuit Diagram](image)

**Capacitive coupling.** The relatively high impedance of the output node makes the circuit very sensitive to crosstalk effects. A wire routed over or next to a dynamic node may couple capacitively and destroy the state of the floating node. Another form of capacitive coupling is backgate (or output-to-input) coupling. When designing and laying out dynamic circuits, special care is needed to minimize capacitive coupling.

**Clock feedthrough.** A special case of capacitive coupling is clock feedthrough, an effect caused by the capacitive coupling between the clock input of the precharge device and the dynamic output node. This coupling capacitance consists of the gate-to-drain capacitance of the precharge device, and includes both the overlap and channel capacitances. This capacitive coupling causes the output of the dynamic node to rise above $V_{DD}$ on the low-to-high transition of the clock, assuming that the pull-down network is turned OFF. Subsequently, the fast rising and falling edges of the clock couple onto the signal node.

The danger of clock feedthrough is that it may cause the normally reverse biased junction diodes of the precharge transistor to become forward biased. This causes electron injection into the substrate, which can be collected by a nearby high-impedance node in the 1 state, finally resulting in faulty operation. CMOS latch up may be another result of this injection. For all purposes, high-speed dynamic circuits should be carefully simulated to ensure that clock feedthrough effects stay within bounds.
Cascading Dynamic Gates

There is one major problem that complicates the design of dynamic circuits during straightforward cascading of dynamic gates to create multilevel logic structures. The problem is best illustrated with two cascaded n-type dynamic inverters as in Figure 8.15(a).

During the precharge phase (Clk = 0), the outputs of both inverters are precharged to $V_{DD}$. Assume that the primary input, In, makes a 0 to 1 transition [Figure 8.14(b)]. On the rising edge of the clock, output Out$_1$ starts to discharge. The second output should remain in the precharged state of $V_{DD}$ as its expected value is 1 (Out$_1$ transitions to 0 during evaluation). However, there is a finite propagation delay for the input to discharge Out$_1$ to GND. Therefore, the second output also starts to discharge. The charge loss leads to reduced noise margins and potential malfunctioning.

The correct operation is guaranteed as long as the inputs can only make a single $0 \to 1$ transition during the evaluation period.

Domino logic. A domino logic module consists of an n-type dynamic logic block followed by a static inverter. Domino CMOS has the following properties:

- Since each dynamic gate has a static inverter, only non-inverting logic can be implemented.
- Very high speeds can be achieved.
NP-CMOS. An alternative approach to cascading dynamic logic is provided by NP-CMOS, which uses two flavours (n-tree and p-tree) of domino logic, and avoids an extra static inverter in the critical path that comes with domino logic. In a p-tree logic gate, PMOS devices are used to build a pull-up logic network, including a PMOS evaluation transistor (Figure 8.16).

The NMOS precharge transistor drives the output low during precharge. The output conditionally makes a $0 \rightarrow 1$ transition during evaluation depending on its inputs. The NP-CMOS logic exploits the duality between n-tree and p-tree logic gates to eliminate the cascading problem. If the n-tree gates are controlled by Clk, and p-tree gates are controlled using Clk, n-tree gates can directly drive p-tree gates, and vice versa.

Problems

**AOI Example:** Implement the following complex logic function in CMOS,

$$G(A, B, C) = \overline{A \cdot B + C^*}$$

**CMOS implementation procedure**

*Step 1:* Connect two NFETs in series and apply inputs $A$ and $B$ to form the AND function $A \cdot B$.

*Step 2:* Connect another NFET in parallel to this combination from step 1 and apply the input.

*Step 3:* Now, to draw the PMOS logic, since the term $(A \cdot B)$ is ORed (in parallel) with $C$ in NMOS logic, in the corresponding PMOS network, both of them must be connected in series.

*Step 4:* Since $AB$ in NMOS is two transistors in series, the corresponding p-transistors must be connected in parallel.

*Step 5:* Apply inputs $A$, $B$ and $C$ to the respective PMOS transistors.

*Step 6:* Apply $V_{DD}$ to the top of PMOS logic and GND to the bottom of NMOS logic.

*Step 7:* Obtain the output expression from the point at which PMOS and NMOS logics are connected together (Figure 8.17).
6. Explain briefly about the working principle of LATCHES and REGISTERS.

**Bistability Principle**

The advantage of pseudo-NMOS gate is the reduced number of transistors \((N + 1)\) versus \(2N\) for complementary CMOS. The nominal high output voltage \((V_{OH})\) for this gate is \(V_{DD}\), but the nominal low output voltage \((V_{OL})\) is not 0 V. This results in reduced noise margins and more static power dissipation. Since the voltage swing on the output and the overall functionality of the gate depend on the NMOS and PMOS sizes, the circuit is called ratioed. This is in contrast to the ratioless logic such as complementary CMOS, where the low and high levels do not depend on transistor sizes.

Static memories use positive feedback to create a bistable circuit—a circuit having two stable states that represent 0 and 1. Figure shows two inverters connected in cascade, i.e., cross coupling of two inverters results in a bistable circuit. The circuit serves as a memory, storing either a 1 or a 0. In the absence of any triggering, the circuit remains in a single state (assuming that the power supply remains applied to the circuit) and thus remembers a value. A bistable circuit is also called flip-flop. A flip-flop is useful only if there also exists a means to bring it from one state to the other one. In general, two different approaches may be used to accomplish the following:

![Bistable Circuit Diagram](image)

- **Cutting the feedback**
  Once the feedback loop is open, a new value can easily be written into Out (or \(Q\)). Such a latch is called multiplexer based, as it realizes that the logic expression for a synchronous latch is identical to the multiplexer equation:

\[
Q = \overline{CLK} \cdot Q + CLK \cdot \overline{Q}
\]

This approach is the most popular in today’s latches.

- **Over powering the feedback loop**
  By applying a trigger signal at the input of the flip-flop, a new value is forced into the cell by overpowering the stored value. This needs a careful signing of the transistors in the feedback loop and the input circuitry.
Dynamic Transmission Edge Triggered Flip Flops

A fully dynamic positive edge-triggered register based on the master–slave concept is shown in Figure 9.7. When CLK = 0, the input data is sampled on storage node 1, which has an equivalent capacitance of \( C_1 \), consisting of the gate capacitance of \( I_1 \), the junction capacitance of \( T_1 \) and the overlap gate capacitance of \( T_2 \). During this period, the slave stage is in the hold mode, with node 2 in a high-impedance (floating) state. On the rising edge of the clock, the transmission gate \( T_2 \) turns ON, and the value sampled on Node 1 right before the rising edge propagates to the output \( Q \). Node 2 now stores the inverted version of Node 1. This implementation of an edge-triggered register is very efficient because it requires only eight transistors. The sampling switches can be implemented using NMOS-only pass transistors, resulting in an even simple six-transistor implementation. The reduced transistor count is attractive for high-performance and low-power systems.

The setup time of this circuit is simply the delay of the transmission gate, and it corresponds to the time it takes node 1 to sample the \( D \) input. The hold time is approximately zero, since the transmission gate is turned OFF on the clock edge and further inputs changes are ignored. The propagation delay is equal to the two-inverter delays plus the delay of the transmission gate \( T_2 \).
7. Explain in detail about CLOCK SKEW in CMOS design

*The C²MOS register.* Figure 9.16 shows a positive edge-triggered register that is based on a master–slave concept insensitive to clock overlap. This circuit is called the C²MOS (Clocked CMOS) register, and operates in two phases:

1. CLK = 0 (CLK̅ = 1): The first tri-state driver is turned ON, and the master stage acts as an inverter sampling the inverted version of D on the internal node X. The master stage is in the evaluation mode. Meanwhile, the slave section is a high impedance mode, or in a hold mode. Both transistors M₇ and M₈ are OFF, decoupling the output from the input. The output Q retains its previous value stored on the output capacitor C_{L₂}.

2. The roles are reversed when CLK = 1: The master stage is in hold mode (M₃-M₄ OFF) while the second section evaluates (M₇-M₈ ON). The value stored on C_{L₁} propagates to the output node through the slave stage, which acts as an inverter.

The advantage of this circuit is that the C²MOS register with CLK̅-CLK clocking is insensitive to overlap, as long as the rise and fall times of the clock edges are sufficiently small.

![Diagram of C²MOS register](image)

*Dual-edge registers.* In this, the sequential circuits are designed to sample the input on both edges (rising or falling) of the clock. The advantage of this scheme is that a lower clock frequency half the original rate is distributed for the same functional throughput, resulting in power savings in the clock distribution network.

The true single-phase clocked register (TSPCR) uses a single clock, CLK. The basic single-phase positive and negative latches are shown in [Figures 9.17(a) and (b)]. For the positive latch, when CLK is high, the latch is in the transparent mode and corresponds to two cascaded inverters; the latch is non-inverting, and propagates the input to the output [Figure 9.17(a)]. On the other hand, when CLK = 0, both inverters are disabled, and the latch is in the hold mode. Only the pull-up networks are still active, while the pull-down circuits are deactivated. As a result of the dual-stage approach, no signal can ever propagate from the input of the latch to the output in this mode. A register can be constructed by cascading positive and negative latches.
The advantage of this circuit is the use of single clock phase. The TSPC offers an additional advantage of the possibility of embedding logic functionality into the latches. This reduces the delay overhead associated with the latches. Figure 9.18 shows an example of positive latch that implements the AND of \( I_1 \) and \( I_2 \) in addition to performing the latching function. This approach of embedding logic into latches has been used extensively in the design of many high-performance processors.

![Positive latch diagram](image1)

**Figure 9.18** Adding logic to the TSPC approach (AND latch).

The disadvantage of TSPC is the slight increase in the number of transistors. Also, when the clock is low (for the positive latch), the output node may be floating, and it is exposed to coupling from other signals. Charge sharing can also occur if the output node drives transmission gates. Dynamic nodes should be isolated with the aid of static inverters, or made pseudo-static for improved noise immunity.

The TSPC circuit can be reduced in complexity as shown in Figure 9.19, where the first inverter is controlled by the clock. These circuits have the additional advantage that clock load is reduced by half. On the other hand, not all the node voltages in the latch experience the full logic swing.

![Negative latch diagram](image2)
8. Explain briefly about KLASS SEMIDYNAMIC FLIP-FLOP (SDFF)

The *Klass semidynamic flip-flop* (SDFF) is a cross between a pulsed latch and a flip-flop. Like the Partovi pulsed latch, it operates on the principle of intersecting pulses. However, it uses a dynamic NAND gate in place of the static NAND.

While the clock is low, \( X \) precharges high and \( Q \) holds its old state. When the clock rises, the dynamic NAND evaluates. If \( D = 0 \), \( X \) remains high and the top NMOS transistor turns OFF. If \( D = 1 \) and \( X \) starts to fall low, the transistor remains ON to finish the transition. This allows for a short pulse and hold time.

The dynamic front end serves as the master latch, while the second stage serves as the slave. The weak cross-coupled inverters staticize the flip-flop and the final inverter buffers the output node.

Like a pulsed latch, the SDFF accepts rising inputs slightly after the rising clock edge. Like a flip-flop, falling inputs must set up before the rising clock edge. It is called *semidynamic* because it combines the dynamic input stage with static operation.

![Diagram of KLASS SEMIDYNAMIC FLIP-FLOP (SDFF)](image)

**Differential Flip-Flops**

*Differential flip-flops* accept true and complementary inputs and produce true and complementary outputs. They are built from a clocked sense amplifier so that they can rapidly respond to small differential input voltages. While they are larger than an ordinary single-ended flip-flop—having an extra inverter to produce the complementary output—they work well with low-swing inputs such as register file bitlines and low-swing busses.

![Diagram of Differential Flip-Flops](image)

Figure 10.29(a) shows a differential *sense-amplifier flip-flop* (SA-F/F) receiving differential inputs and producing a differential output [Matsui94]. When the clock is low, the internal nodes \( X \) and \( Q \) precharge. When the clock rises, one of the two nodes is pulled down, while the cross-coupled pMOS transistors act as a keeper for the other node. The SR latch formed by the cross-coupled NAND gates behaves as a slave stage, capturing the output and holding it through precharge.
The flip-flop can amplify and respond to small differential input voltages, or it can use an inverter to derive the complementary input from D.

Dual Edge-Triggered Flip-Flops

Two conceptual designs for DET flipflops are shown in Figure along with circuit realizations. In the master-slave design of two separate master latches operate on opposite phases of the clock. The multiplexer, serving in place of the slave latch, selects the result of the opaque master. Transistor-level implementation of this design is shown in figure 9.

9. Write a brief note on Radiation-Hardened Flip-Flops

Soft errors caused by alpha particles or cosmic rays were once of primary concern in memories because RAM cells have the smallest node capacitance and weakest feedback, so they are easily disturbed, as discussed in Section 7.3.4. As transistors have scaled, soft error rates for flip-flops have increased to the point that they are important for high-reliability systems. Radiation-hardened flip-flops are designed to resist such errors. They are also critically important for space applications where the cosmic ray flux is much greater. The simplest way to minimize soft errors is to use a storage node holding enough charge that a particle strike is unlikely to flip the state. This has become difficult in nanometer processes because scaling reduces both the capacitance and voltage, greatly decreasing the charge. An unusually large storage node can still reduce the probability of disturbance, but it comes at a cost in performance, energy, and area.
Pulsed Latches

Pulsed latches are faster than flip-flops and offer some time borrowing capability at the expense of greater hold times. They have fewer clocked transistors and hence lower power consumption. If intentional time borrowing is not necessary,

![Pulsed Latch Diagram]

Transparent Latches

Transparent latches also have lower sequencing overhead than flip-flops and are attractive because they permit nearly half a cycle of time borrowing. One latch must be placed in each half-cycle. Data can arrive at the latch any time the latch is transparent. A convenient design approach is to nominally place the latch at the beginning of each half-cycle. Then time borrowing occurs when the logic in one half-cycle is longer than nominal and data does not arrive at the next latch until some time into the next half-cycle.

Figure illustrates pipeline timing for short and long logic paths between latches.

![Pipeline Timing Diagram]

When the path is short (a), the data arrives at the second latch early and is delayed until the rising edge. Therefore, it is natural to consider latches residing at the beginning of their half-cycle because short paths automatically adjust to operate this way.

When the path is longer (b), it borrows time from the first half-cycle into the second.

10. Explain briefly about i) Synchronizers ii) Arbiters

A synchronizer is a circuit that accepts an input that can change at arbitrary times and produces an output aligned to the synchronizer’s clock. Because the input can change during the synchronizer’s aperture, the synchronizer has a nonzero probability of producing a metastable output.

This section first examines the response of a latch to an analog voltage that can change near the sampling clock edge. The latch can enter a metastable state for some amount of time that is unbounded, although the probability of remaining metastable drops off exponentially with time.
i) A Simple Synchronizer

A synchronizer accepts an input $D$ and a clock. It produces an output $Q$ that ought to be valid some bounded delay after the clock. The synchronizer has an aperture defined by a setup and hold time around the rising edge of the clock. If the data is stable during the aperture, $Q$ should equal $D$. If the data changes during the aperture, $Q$ can be chosen arbitrarily.

Figure 10.46 shows a simple synchronizer built from a pair of flip-flops. $F1$ samples the asynchronous input $D$. The output $X$ may be metastable for some time, but will settle to a good level with high probability. $F2$ samples $X$ and produces an output $Q$ that should be a valid logic level and be aligned with the clock. The synchronizer has a latency of one clock cycle, $T_c$. It can fail if $X$ has not settled to a valid level by a setup time before the second clock edge.

![Synchronizer Diagram]

Each flip-flop samples on the rising clock edge when the master latch becomes opaque. The slave latch merely passes along the contents of the master and does not significantly affect the probability of metastability. If the synchronizer receives an average of $N$ asynchronous input changes at $D$ each second, the probability of synchronizer failure in any given second is

$$P(\text{failure}) = N \frac{T_0}{T_c} e^{-\frac{(T_c - t_{\text{setup}})}{\tau_s}}$$  \hspace{1cm} (10.27)

and the mean time between failures increases exponentially with cycle time

$$MTBF = \frac{1}{P(\text{failure})} = \frac{T_c}{N T_0} e^{\frac{T_c}{\tau_s}}$$  \hspace{1cm} (10.28)
ii) Arbiters

The arbiter of is closely related to the synchronizer. It determines which of two inputs arrived first. If the spacing between the inputs exceeds some aperture time, the first input should be acknowledged. If the spacing is smaller, exactly one of the two inputs should be acknowledged, but the choice is arbitrary.

For example, in a television game show, two contestants may pound buttons to answer a question. If one presses the button first, she should be acknowledged. If both press the button at times too close to distinguish, the host may choose one of the two contestants arbitrarily.

Figure shows an arbiter built from an SR latch and a four-transistor metastability filter. If one of the request inputs arrives well before the other, the latch will respond appropriately. However, if they arrive at nearly the same time, the latch may be driven into metastability. The filter keeps both acknowledge signals low until the voltage difference between the internal nodes n1 and n2 exceeds Vt, indicating that a decision has been made.

Such an asynchronous arbiter will never produce metastable outputs. However, the time required to make the decision can be unbounded,