Final Assessment | Fall - 2023

Md. Shafayet Hossain

CSE - 21st Batch | **Course Code:** CSE - 223

Course Title: Digital Electronics & Pulse Technique

ID: 2121210071

Answer to the Question no- 1

(a)

Master-Slave Flip-Flop Explained:

A master-slave flip-flop is a type of digital circuit designed to overcome the limitations of a single flip-flop, particularly the issue of "race-around condition." It consists of two interconnected flip-flops, one acting as the "master" and the other as the "slave."

Working Principle:

- **Data Capture:** During the active edge of the clock, the master flip-flop captures the data present at its input.
- **Data Transfer:** When the clock transitions to the inactive edge, the master flip-flop becomes isolated, and the slave flip-flop captures the data stored in the master.
- **Output Stabilization:** The slave flip-flop holds the captured data until the next clock cycle, providing a stable output regardless of changes in the input signal during the capture window.

.

Example:

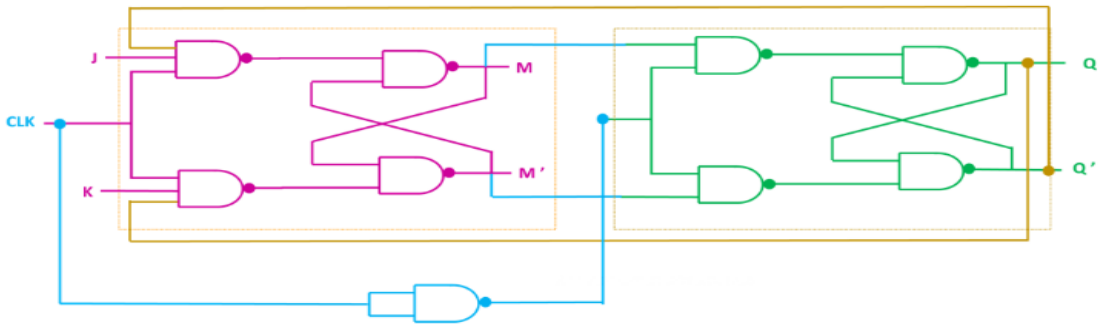
Imagine a system where you want to register a data signal that might change quickly. A single flip-flop might struggle to capture the correct value due to the race-around condition. Using a master-slave flip-flop, the master captures the data during the clock pulse, and the slave latches it at the end, ensuring a stable and reliable output.

Real-World Applications:

- Data synchronizers
- Frequency dividers
- Edge detectors
- Shift registers
- Counters

1(b)

Master-Slave Flip-Flop using JK Flip-Flops



Circuit Diagram of Master Slave JK Flip-Flop

Description:

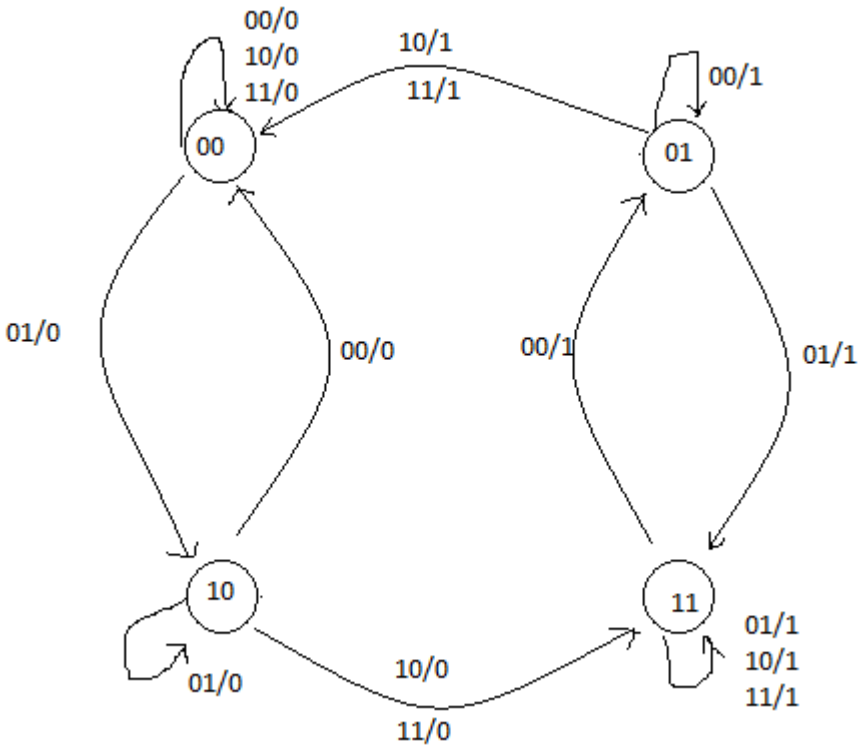
- The master flip-flop receives the clock signal directly and samples its inputs (J and K) on the rising edge.
- The slave flip-flop receives the inverted clock signal (delayed by one clock cycle) and samples the master's output (Q and Q') on the falling edge.
- This ensures that the slave only changes its state based on the stable output of the master, eliminating timing hazards.
- The feedback loop maintains the state information until the next clock cycle.

Functionality:

- Imagine the clock signal rising (positive edge).
    - The master flip-flop samples its inputs (J and K) and stores the new state based on those inputs.
    - Since the slave clock is low (due to the inverter), the slave flip-flop remains unaffected.
- When the clock signal falls (negative edge).
    - The master flip-flop holds its state, not affected by the falling edge.
    - The slave flip-flop samples the master's output (stored value) and sets its own state accordingly.
    - Now, the slave's output reflects the previous master's state.

Answer to the Question no- 2

State Diagram Drawing



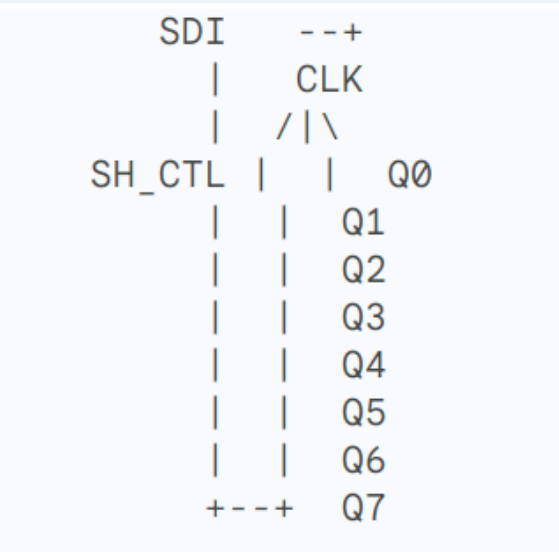
Answer to the Question no- 3

(a)

8-Bit Serial Register

An 8-bit serial register is a digital circuit that stores and manipulates 8 bits of data one bit at a time. Here's a breakdown of its structure and function-

Diagram:



Components:

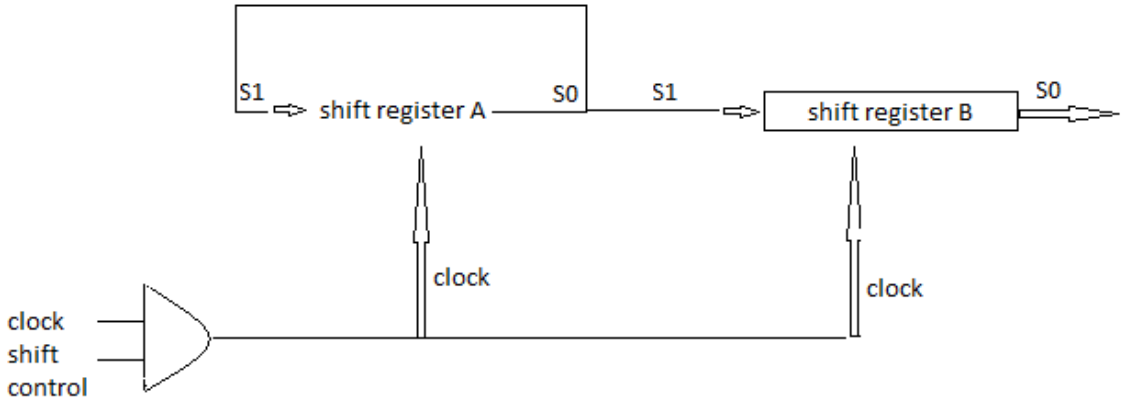
- **Flip-flops:** The core element, 8 flip-flops (usually D flip-flops) store the individual bits of data. Think of them as tiny boxes that can hold either a 0 or a 1.
- **Shift Register Logic:** Controls the movement of data within the register. This includes:
    - Serial input (SDI): Receives the incoming data, one bit at a time.
    - Clock input (CLK): Synchronizes the movement of data. On each rising edge (positive transition), data shifts between flip-flops.
    - Parallel outputs (Q0-Q7): Provide access to all 8 bits simultaneously.
    - Shift control (SH_CTL): (Optional) In some registers, this input determines the shift direction (left or right) or enables/disables shifting.

Functionality:

- Loading data:
    - Data is presented at the SDI input.
    - On a rising CLK edge, the data enters the first flip-flop (D0).
    - Subsequent rising edges shift the data one position to the right, pushing out the previous data bit from the last flip-flop (Q7).
- Shifting data:
    - With each rising CLK edge, data is shifted within the register, regardless of new data at SDI.
    - This allows serial-to-parallel conversion (serial input, parallel output) or vice versa.

3(b)

Here's how serial transfer from register A (Reg A) to register B (Reg B) works –

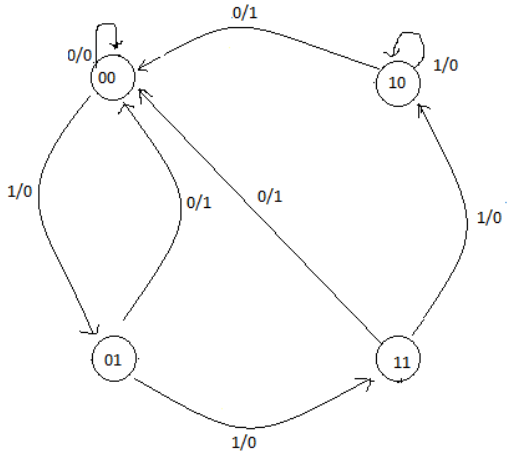


Example:

Timing Pulse	Shift Register A	Shift Register B
Initial Value	1 0 1 1	0 0 1 0
After T1	1 1 0 1	1 0 0 1
After T2	1 1 1 0	1 1 0 0
After T3	0 1 1 1	0 1 1 0
After T4	1 0 1 1	1 0 1 1

Answer to the Question no- 4

State Diagram



Present State		Next State				Output	
		X = 0		X = 1		X = 0	X = 1
A	B	A	B	A	B	Y	Y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

Here X = 0 and X = 1

So,

A(++1) = AX+BX

B(++1) = A'X

Y(+) = (A+B) X'