NAME: Nwadike Reginald-Francis Chukwuma

MATRIC NUMBER: 17/ENG02/052

Applications of Flip Flops

Application of the flip flop circuit mainly involves in bounce elimination switch, data storage, data transfer, latch, registers, counters, frequency division, memory, etc.

Flip flop synchronization

What is synchronous flip flop?
The normal data inputs to a **flip flop** (D, S and R, or J and K) are referred to as **synchronous** inputs because they have effect on the outputs (Q and not-Q) only in step, or in sync, with the clock signal transitions.

Sequence Detectors

A **sequence detector** is a sequential circuit that outputs 1 when a particular pattern of bits sequentially arrives at its data input. The figure below shows a block diagram of a sequence detector. It has two inputs and one output. The inputs are the clock used to synchronize the functionality of the circuit and the data input.

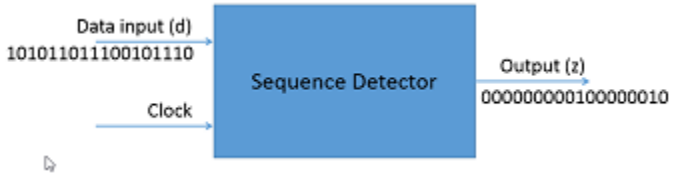


Figure 1: A block diagram of a sequence detector with sample input and output

The data input receives the input sequence. In our figure, the input sequence and the output sequence of the circuit are a sample of a 0111 sequence detector. If you follow the input and output sequences, you can see that only when the last four bits of the input sequence are 0111 does the output turn to 1 during one clock cycle. It then turns back to 0.

DATA STORAGE

Data Storage. A flip flop **store** one bit at a time in **digital circuit**. In order to **store** more than one bit flip flop can be connected in series and parallel called registers. ... Thus a 4 bit register consists of 4 individual flip flops, each able to **store** one bit of information at a time.

DATA TRANSFER

Data is **transferred** in the form of bits between two or more **digital** devices. There are two methods used to transmit **data** between **digital** devices: serial **transmission** and parallel **transmission**. Serial **data transmission** sends **data** bits one after another over a single channel.

PARALLEL DATA TRANSFER

In **data transmission**, **parallel communication** is a method of conveying multiple binary digits (bits) simultaneously. It contrasts with

serial **communication**, which conveys only a single bit at a time; this distinction is one way of characterizing a communications link.

SHIFT REGISTER

The Shift Register is another type of sequential logic circuit that can be used for the storage or the transfer of binary data.

This sequential device loads the data present on its inputs and then moves or “shifts” it to its output once every clock cycle, hence the name **Shift Register**. *Shift Registers* are used for data storage or for the movement of data and are therefore commonly used inside calculators or computers to store data such as two binary numbers before they are added together, or to convert the data from either a serial to parallel or parallel to serial format. The individual data latches that make up a single shift register are all driven by a common clock (Clk) signal making them synchronous devices.

Shift left operation

Featured snippet from the web

Left Shifts

The bit positions that have been vacated by the **shift operation** are zero-filled. A **left shift** is a logical **shift** (the bits that are **shifted** off the end are discarded, including the sign bit).

Shift Left is a practice intended to find and prevent defects early in the software delivery process. The idea is to improve quality by moving tasks to the **left** as early in the lifecycle as possible. **Shift Left** testing means testing earlier in the software development process

PARALLER VS SERIAL TRANSFER

In a digital communications system, there are 2 methods for data transfer: **parallel** and **serial**. Parallel connections have multiple wires running parallel to each other (hence the name), and can transmit data on all the wires simultaneously. Serial, on the other hand, uses a single wire to transfer the data bits one at a time.

FREQUENCY DIVISION AND COUNTING

For **frequency division**, toggle mode flip-flops are used in a chain as a divide by two counter.

The final output clock signal will have a frequency value equal to the input clock frequency divided by the MOD number of the counter. Such circuits are known as “divide-by-n” counters. Counters can be formed by connecting individual flip-flops together and are classified according to the way they are clocked.

The basic idea of making a frequency counter is to count the number of cycles of the input in one second. We can keep one counter that keeps counting the rising/falling edges of the input signal. Then we can read the value of this counter once in a second and then clear it. The value we read from the counter is nothing but the frequency of the input signal.

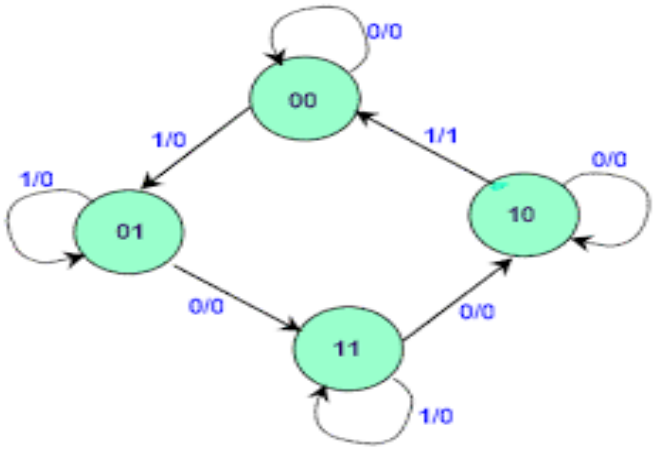
COUNTERS AND REGISTERS

Counters and registers belong to the category of MSI sequential logic circuits. They have similar architecture, as both counters and registers comprise a cascaded arrangement of more than one flip flop with or without combinational logic devices. Both constitute very important building blocks of sequential logic, and different types of counter and register available in integrated circuit (IC) form are used in a wide range of digital systems. While counters are mainly used in counting applications, where they either measure the time interval between two

unknown time instants or measure the frequency of a given signal, registers are primarily used for the temporary storage of data present at the output of a digital circuit before they are fed to another digital circuit.

STATE TRANSITION DIAGRAM

A **diagram** consisting of circles to represent **states** and directed line segments to represent **transitions** between the **states**. One or more actions (outputs) may be associated with each **transition**.
The **diagram** represents a finite **state** machine.



AN EXAMPLE OF A STATE TRANSITION DIAGRAM

MOD NUMBER

The **Modulus** (or **MOD-number**) of a **counter** is the total **number** of unique states it passes through in one complete counting cycle with a **mod**-n **counter** being described also as a divide-by-n **counter**.
The **modulus** of a **counter** is given as: 2^n where n = **number** of flip-flops.

ASYNCHRONOUS (RIPPLE) COUNTER

A **ripple counter** is an **asynchronous counter** where only the first flip-flop is clocked by an external clock. All subsequent flip-flops are clocked by the output of the preceding flip-flop. **Asynchronous** counters are also called **ripple**-counters because of the way the clock pulse **ripples** it way through the flip-flops.

SIGNAL FLOW

Signal flow is the order of operations a sound goes through. When you record an instrument, the sound goes through different stages before you hear it through the speakers. **Signal flow** is the list of steps that sound goes through in order for you to hear it. It's kind of like riding a train.

PROPAGATION DELAY IN RIPPLE COUNTERS

A major problem with ripple counters arises from the propagation delay of the flip-flops constituting the counter. The effective propagation delay in a ripple counter is equal to the sum of propagation delays due to different flip-flops. The situation becomes worse with increase in the number of flip-flops used to construct the counter, which is the case in larger bit counters. An increased propagation delay puts a limit on the maximum frequency used as clock input to the counter. We can appreciate that the clock signal time period must be equal to or greater than the total propagation delay. The maximum clock frequency therefore corresponds to a time period that equals the total propagation delay. If tpd is the propagation delay in each flip-flop, then, in a counter with N flip-flops having a modulus of less than or equal to 2N , the maximum usable clock frequency is given by $f_{max} = 1/(N \times t_{pd})$. Often the propagation delay times are specified in the case of flip-flops, one for

LOW-to-HIGH transition (tpLH) and the other for HIGH-to-LOW transition (tpHL) at the output. In such a case, the larger of the two should be considered for computing the maximum clock frequency

SYNCHRONOUS COUNTERS

The propagation delay becomes prohibitively large in a ripple counter with a large count. On the other hand, in a synchronous counter, all flip-flops in the counter are clocked simultaneously in synchronism with the clock, and as a consequence all flip-flops change state at the same time. The propagation delay in this case is independent of the number of flip-flops used. Since the different flip-flops in a synchronous counter are clocked at the same time, there needs to be additional logic circuitry to ensure that the various flip-flops toggle at the right time.

Advantages of synchronous over asynchronous counter

it can operate on higher frequency than **asynchronous counter** as it does not have cumulative delay because of same clock is given to each flip flop.

Less likely to end up in erroneous state. They are faster as the the propagation delay are small as compared to **asynchronous counters**. There are no counting errors as compared to **asynchronous counters**.

Performance is much better, liable and portable circuit**.**

What is MOD number in counter?
The **number** of states or counting sequences through which a particular **counter** advances before returning once again back to its original first state is called the **modulus** (**MOD**). In other words, the **modulus** (or just **modulo**) is the **number** of states the **counter** counts and is the dividing **number** of the **counter**.

SYNCHRONOUS DOWN AND UP/DOWN COUNTERS

Counters are used in many different applications. Some count up from zero and provide a change in state of output upon reaching a predetermined value; others count down from a preset value to zero to provide an output state change.

However, some counters can operate in both up and down count mode, depending on the state of an up/down count mode input pin. They can be reversed at any point within their count sequence. Dual purpose ICs such as the TTL 74LS190 and 75LS191 are available which implement both Up and Down count functions.

The TTL 74LS190 is a 4-bit device that can be switched between Up and Down modes, and provides a BCD decade output; the 74LS191 is a binary counter. The counters are synchronous, but they are asynchronously presettable. Four data inputs (A – D) allow the preset target to be loaded. The counter is decremented or incremented synchronously with the low to high transition of the clock. The counters can be cascaded in high-speed mode.

A simple three-bit Up/Down synchronous counter can be built using JK flip-flops configured to operate as toggle or T-type flip-flops giving a maximum count of zero (000), advancing through 001, 010 to seven (111) and back to zero again.

PRESETTABLE COUNTERS

What is a presettable counter?
Presettable: This means when the LD input is high, then whatever binary value is present on LOAD INPUTS, will be immediately copied to the outputs and stay that way until LD goes low. This enables the **counter** to begin from any value. , eg **count** from 6 to 15. Note this only works if the RESET input is low.

ACTIVE HIGH AND LOW DECODING

An active HIGH decoder produces high outputs to indicate detection While active LOW means that the function gets done when input is in low state.

BDC COUNTER DECODING

A BDC counter has 10 states that can be decoded using the techniques described previously. BCD decoders provide 10 outputs corresponding to the decimal digits 0 through 9 and represented by the states of the counter.

ANALYZING SYNCHRONOUS COUNTERS

Synchronous Counter Analysis

Synchronous counters can be analyzed by using a procedure similar to the analysis of asynchronous counters to classify fully or define the counter operation.
Analyze a synchronous counter circuit by proceeding through the following steps:
1. Verify that the counter system clock is common to all flip-flop stages and that the circuit is in fact a synchronous counter.
2. Determine the number of stages of the counter by counting the flip-flops or outputs.
3. Determine the type of flip-flops and the input logic function for each stage. For reference, recall the present state-present input-next state table for each flip-flop.
4. Construct a present input-present state-next state table for the counter to determine the inputs to the flip-flops and the resulting outputs.
5. Analyze the counter using the present input-present state-next state table to determine the complete count sequence. Continue the analysis until the count sequence begins to repeat.
6. Determine the modulus of the counter.
7. Construct a state transition diagram to describe the counter operation.
8. Graph the output waveforms produced by the counter. The key to the synchronous counter analysis is the present state-present input-next state table that serves as a truth table description of the counter operation as it progresses with each clock pulse. To complete this table properly, we need to define correctly the flip-flops of each stage by their own present input-present state-next state table

The steps to design a Synchronous Counter using JK flip flops are:

- Describe a general sequential circuit in terms of its basic parts and its input and outputs. ...
- Draw the state diagram for the given sequence.
- Develop a next-state table for the specific **counter** sequence.

J-K FLIP FLOP EXCITATION TABLE

Excitation Table-

The excitation table of any flip flop is drawn using its truth table.

What is excitation table?

For a given combination of present state Q_n and next state Q_{n+1}, excitation table tell the inputs required.

Q_n	Q_{n+1}	S	R
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

SYNCHRONOUS COUNTER DESIGN PROCEDURES

1. Determine the desired number of bits of FFs needed to support the counting sequence's
2. Build a state transition diagram. Be sure to include all states
3. Build a state/excitation truth table
4. Simplify expressions for J and K inputs for each F/F on K-maps
5. Implement the synchronous counter/state machine circuit
6. Draw the timing diagram (if needed).

STEPPER MOTOR CONTROL

Stepper motors are DC **motors** that move in discrete steps. They have multiple coils that are organized in groups called "phases". By energizing each phase in sequence, the **motor** will rotate, one step at a time. With a computer **controlled** stepping you can achieve very precise positioning and/or speed **control**

Synchronous counter design with D FF

Synchronous counter (**D** flipflops) A 4-bit **synchronous counter** built from **D**-flipflops with carry-input (count-enable) and carry-output. In this circuit, the single clock signal is directly connected to all flipflops, so that all flipflops change state at the same time.

SERIAL IN SERIAL OUT SHIFT REGISTERS

Serial In Serial Out (SISO) **shift** registers are a kind of **shift** registers where both data loading as well as data retrieval to/from the **shift register** occurs in **serial**-mode. ... Here the data word which is to be stored is fed bit-by-bit at the input of the first flip-flop.

PARALLEL IN SERIAL OUT

In **Parallel In Serial Out** (PISO) shift registers, the data is loaded onto the register in **parallel** format while it is retrieved from it serially. ... Thus the bits of the input data word (Data in) appearing as inputs to the gates A_2 are passed on as the outputs of OR gates at each individual combinational circuit.

Serial in parallel out

Serial in Parallel Out (SIPO) Shift Register. In **Serial In Parallel Out** (SIPO) shift registers, the data is stored into the register serially while it is retrieved from it in **parallel**-fashion. ... In general, the register contents are cleared by applying high on the clear pins of all the flip-flops at the initial stage.

RING COUNTER

A **ring counter** is a type of counter composed of flip-flops connected into a shift register, with the output of the last flip-flop fed to the input of the first, making a "circular" or "ring" structure.