IDARA INIUBONG ETUK

17/ENG02/025

FLIP FLOP APPLICATIONS

Edge triggered (clocked) flip flops are versatile devices that can be used in a wide variety of applications including counting, storing of binary data, transferring binary data from one location to another, and many more. Almost all applications utilize the FF's clocked operation. Many of them fall under the category of sequential circuits. A sequential circuit is one in which the outputs follow a predetermined sequence of states, with a new state occurring each time a clock pulse occurs.

FLIP-FLOP SYNCHRONIZATION

Most digital systems are principally synchronous in their operation because most of the signals will change states in synchronism with the clock transitions. In many cases, however, there will be an external signal that is not synchronized to the clock; in other words, it is asynchronous. Asynchronous signals often occur as a result of a human operator's actuating an input switch at some random time relative to the clock signal. This randomness can produce unpredictable and undesirable results.

DATA STORAGE

A flip flop stores 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. A register is simply a data storage device for a number of bits in which each flip flop store one bit of information (0 or 1). Thus a 4 bit register consists of 4 individual flip flops, each able to store one bit of information at a time.

DATA TRANSFER

Flip flops can also be used extensively to transfer the data. For this purpose shift register is used. A shift register which is able to shift or transfer its content within itself without changing the order of the bits. It may be designed to shift or transferred one bit at a time, when a clock pulse is applied. The shift register can be built using RS, JK OR D flip flops.

There are two ways to transfer data between computers: Serial Transmission and Parallel Transmission.

Serial Transmission

Data is sent bit by bit from one computer to another in two directions. Each bit has a clock pulse rate. Eight bits are transmitted at a time, with a start and stop bit known as a parity bit, which is 0 and 1, respectively. Data cables are used when transmitting data over a longer distance. The data cable has D-shaped 9 pin cable that connects the data in series.

Categories of Serial Transmission

Asynchronous Transmission – an extra bit is added to each byte to alert the receiver to the arrival of new data. 0 is used as a start bit, while 1 used as a stop bit.
Synchronous transmission – no extra bit is added to each byte. Data is transferred in batches, each of which contains multiple bytes.

Parallel Transmission

Several bits are transmitted simultaneously with one clock pulse rate. It transmits quickly as it utilises several input and output lines for sending the data.

It uses a 25-pin port with 17 signal lines and 8 ground lines. The 17 signal lines are divided as

- **4 lines – initiate handshaking**
- **5 lines – communicate and notify errors**
- **8 lines – transfer data**

Applications

Serial transmission occurs between two computers, or from a computer to an external device located far away.

Parallel transmission can take place within a computer system, through a computer bus, or to an external device located nearby.
Examples:

One example of serial mode transmission is a connection established between a computer and a modem using the RS-232 protocol. An RS-232 cable can accommodate 25 wires, but only two of these wires are for data transmission; the rest are designated for overhead control signaling. The two data wires run using simple serial transmission in either direction.

In this example, a computer may be far from the modem, making parallel transmission very expensive. With this in mind, speed of transmission is considered less important when compared to the economic advantage of serial transmission.

An example of parallel mode transmission is a connection established between a computer and a printer. Most printers are within 6 meters (about 20 feet) from the transmitting computer, and the slight cost for extra wires is offset by the added speed gained through parallel transmission of data.

Comparison between Serial and Parallel Transmission

Basis for Comparison	Serial Transmission	Parallel Transmission
Definition	Data flows in 2 directions, bit by bit	Data flows in multiple directions, 8 bits (1 byte) at a time
Cost	Economical	Expensive
Number of bits transferred per	1 bit	8 bits or 1 byte

clock pulse		
Speed	**Slow**	**Fast**
Applications	**Used for long distance communication**	**Used for short distance communication**
Example	**Computer to computer**	**Computer to printer**

Differences between Serial and Parallel Transmission

- **Serial transmission requires a single line to send data. Parallel transmission requires multiple lines to send data.**
- **There are fewer errors and less noise in serial transmission, since the transmission is done one bit at a time. There are more errors and noise in parallel transmission, since the transmission is done multiple bits at a time.**
- **Serial transmission is slower since data flows through a single line; conversely, parallel transmission is faster since data flows through multiple lines.**
- **Serial transmission is 'full duplex' since the sender can send and receive data at the same time. Parallel transmission is 'half duplex' since the data can be sent or received at any given time.**
- **The cables used in serial transmission are thinner, longer, and more economical compared to the cables used in parallel transmission.**
- **Serial transmission is reliable and straightforward. Parallel transmission is unreliable and complicated.**

Both serial and parallel transmissions have advantages and disadvantages. Parallel transmission is used for shorter distances and provides greater speed, while serial transmission is reliable for transferring data over longer distances. Both serial and parallel transmissions are individually essential for transferring data.

FREQUENCY DIVISION

Flip flops can divide the frequency of periodic waveform. When a pulse wave is used to toggle a flip flop, the output frequency becomes one half the input frequency.

The output of each flip flop is half the frequency of an input. If the input frequency is 160 KHz then output of each flip flop would be so after first flip flop, 40 after second flip flop and 20 after third flip flop.

Input frequency 160 KHz

Frequency of first flip flop 80 KHz

Frequency of second flip flop 40KHz

Frequency of third flip flop 20KHz

STATE TRANSITION DIAGRAM

A state transition diagram is a digraph whose nodes are states and whose directed arcs are transitions labelled by event names.

It is used to represent a finite state machine.

These are used to model objects which have a finite number of possible states and whose interaction with the outside world can be described by its state changes in response to a finite number of events. A state transition diagram is a digraph whose nodes are states and whose directed arcs are transitions labelled by event names. A state is drawn as a rounded box containing an optional name. A transition is drawn as an arc with the arrow from the receiving state to the target state. The label on the arrow is the name of the event causing the transition. State transition diagrams have a number of applications. They are used in object-oriented modelling techniques to represent the life cycle of an object. They can be used as a

finite state recognizer for a regular language, for instance, to describe regular expressions used as variables in computer languages.

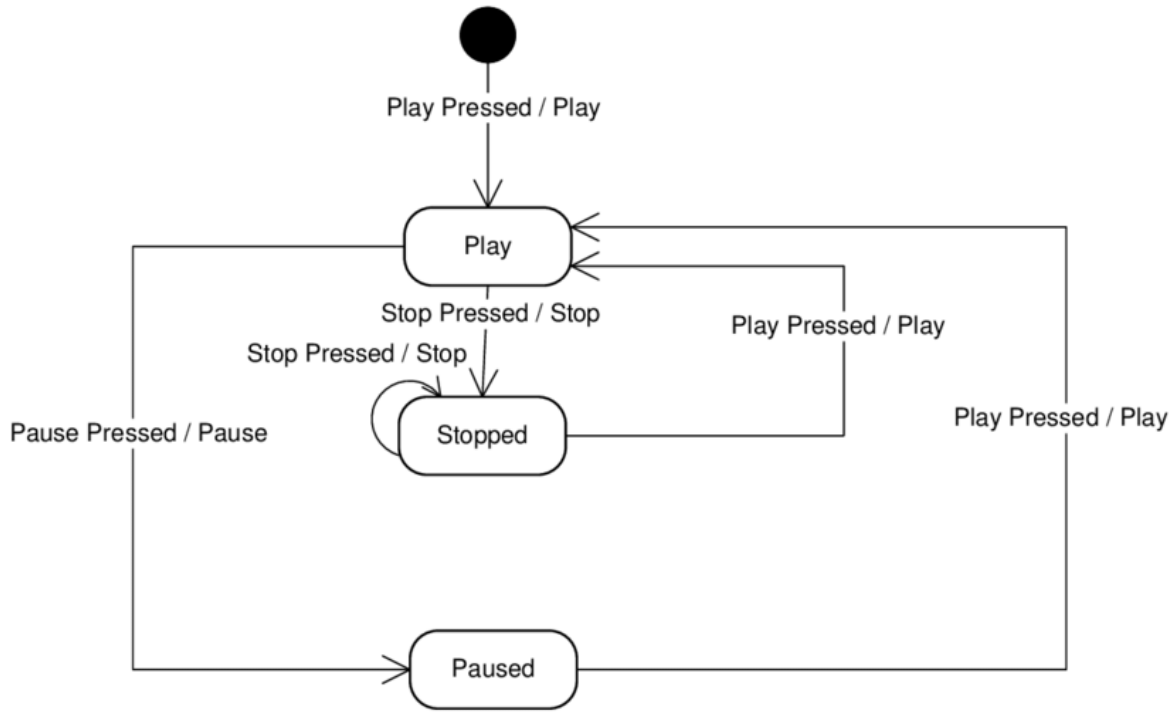


FIG: STATE DIAGRAM FOR THE INTERACTION DESIGN OF A CD PLAYER

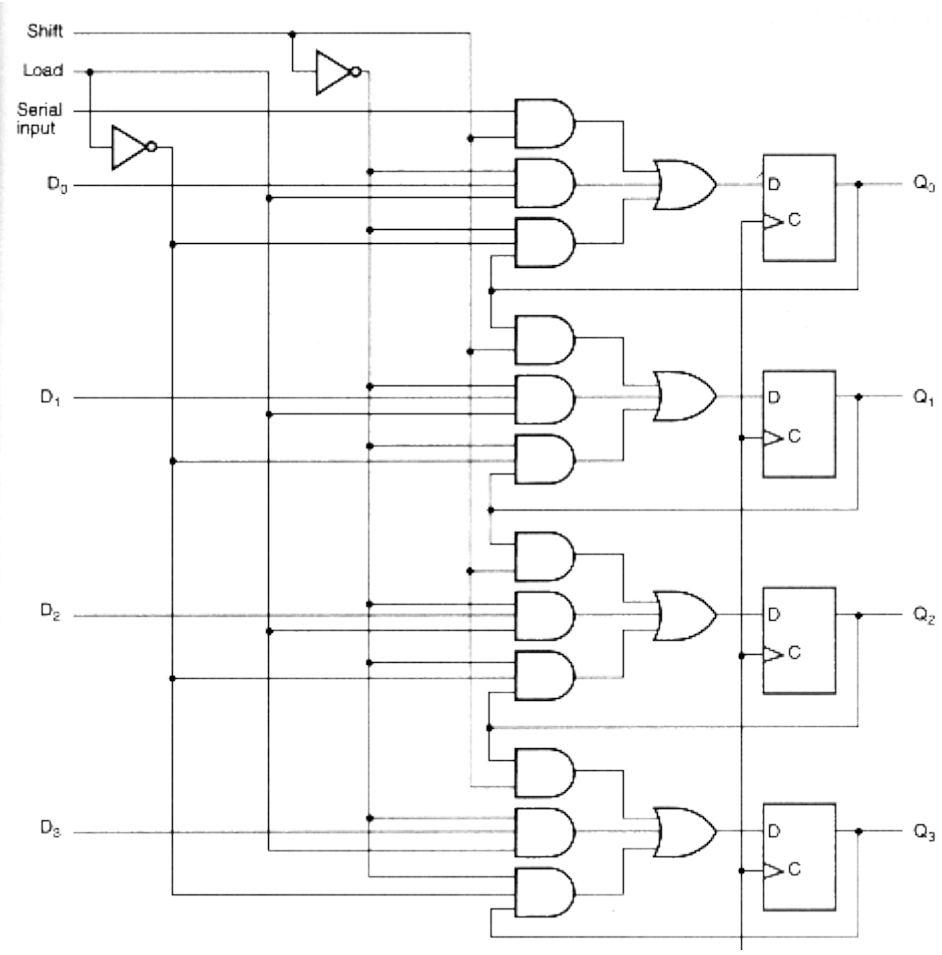
REGISTERS AND COUNTERS

A register is a set of flip-flops with combinational logic to implement state transitions that allow information to be *stored* and *retrieved* from them. In the simplest form, a flip-flop is a one-bit register.

A counter is simply a register with combinational logic to implement counting, that is it is possible to retrieve the contents, add or subtract one to the contents, and then store it back into the register in one operation.

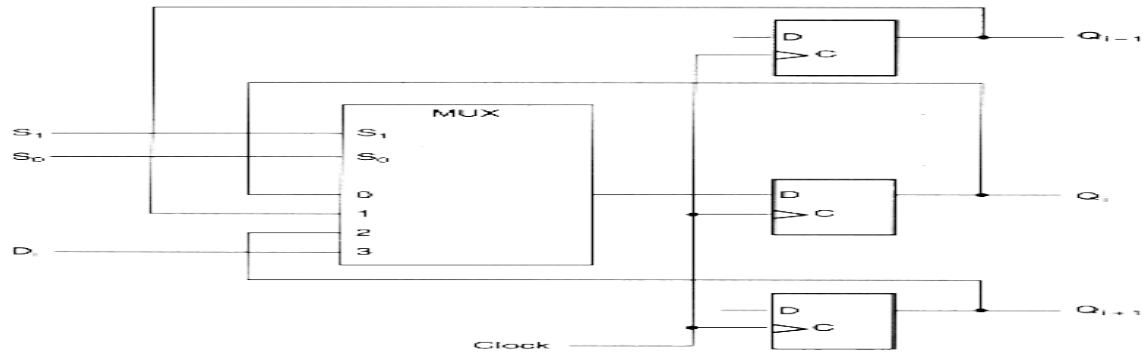
UNIVERSAL SHIFT REGISTER

It is useful sequential building block allows serial shifts and parallel load functions.



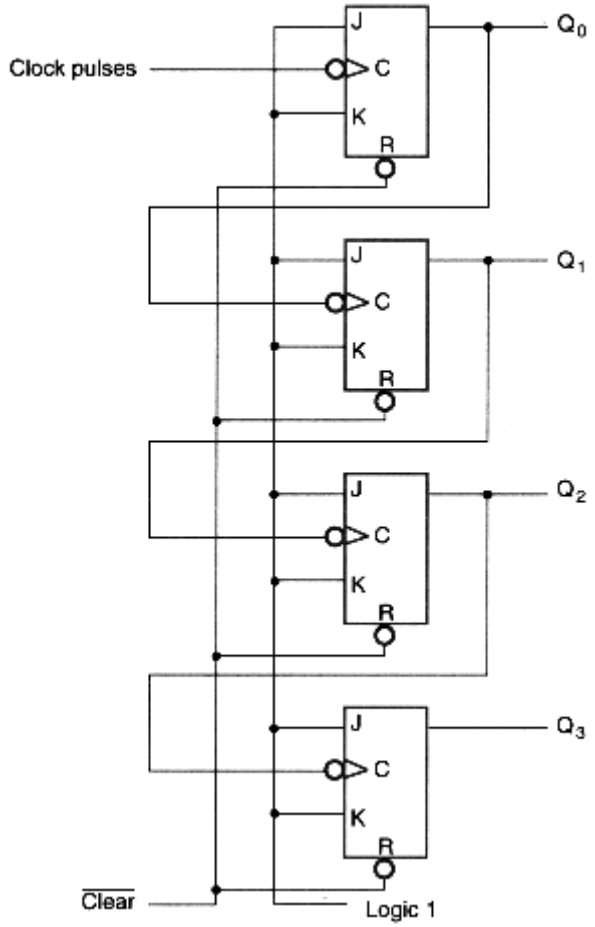
We can simply the design using combination building blocks, and the example below, called a *universal shift register* allows serial shift in either

direction.



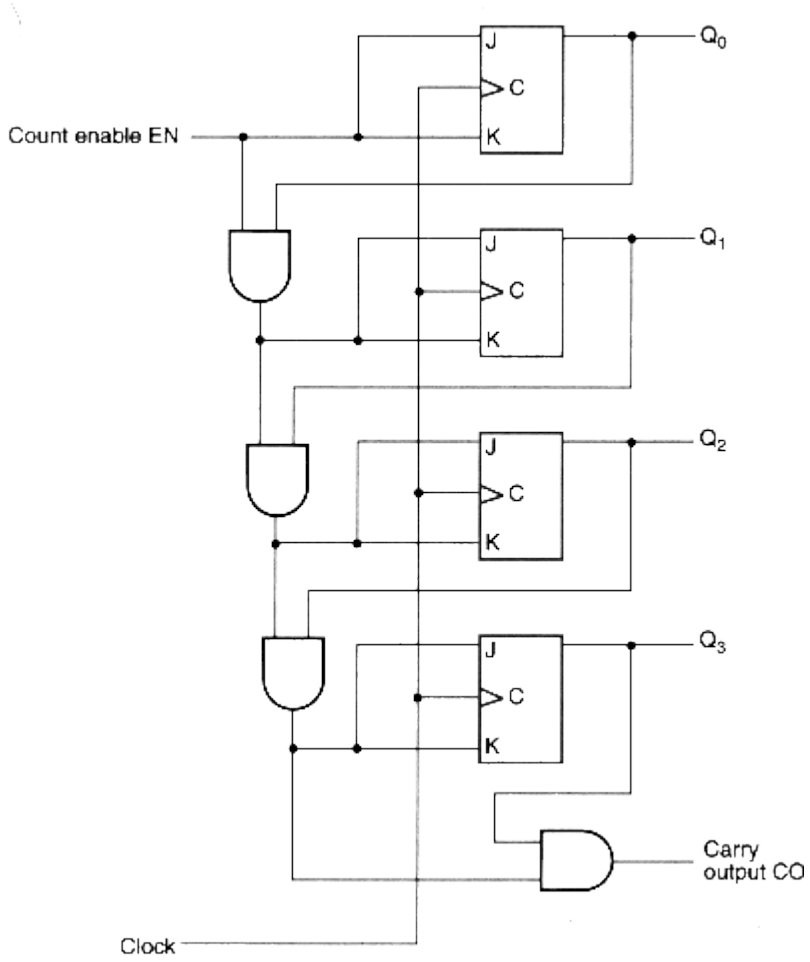
COUNTERS

A four-bit up-counter using *J-K* flip-flops is shown below. This will always count on each clock pulse.

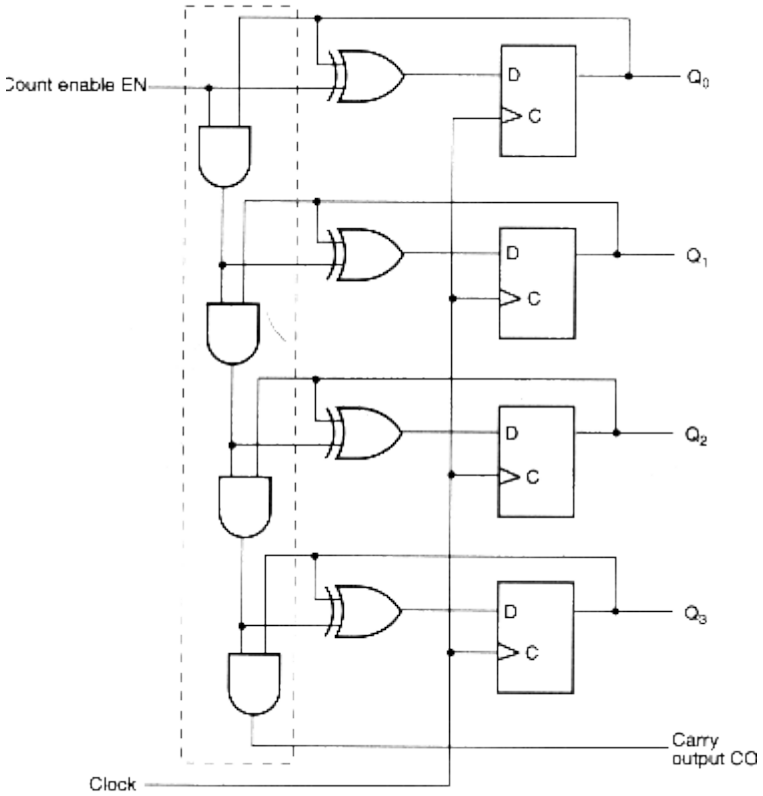


When we need to count based on a given *enable* signal, the ripple carry

problem emerges.

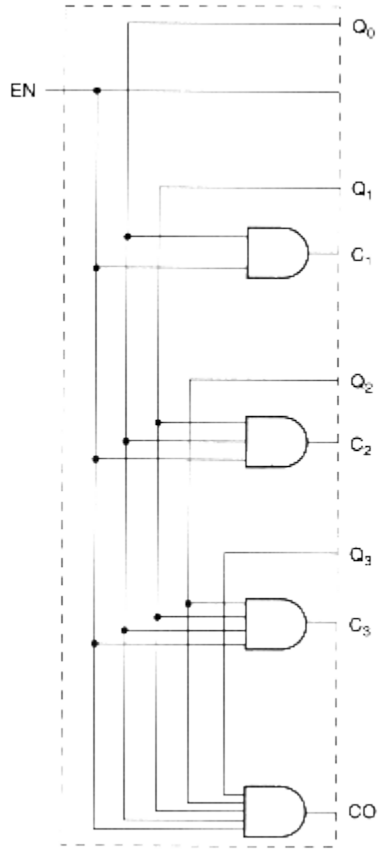


In the case of an implementation using D-type flip-flops, we can use *serial gating* to propagate the enable signal as shown below.



(a) Serial gating

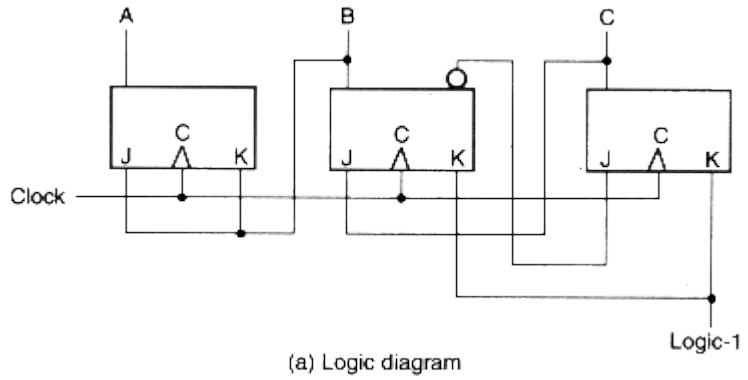
Alternatively a faster *parallel* gating is shown but this involves gating the clock signal itself.



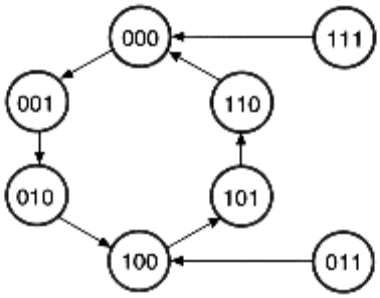
(b) Parallel gating

ARBITRARY COUNTER

The following is an example of a *counter* which counts through an arbitrary sequence.



(a) Logic diagram



(b) State diagram