Propagation Delay in Ripple Counters

What is Propagation Delay?

Propagation delay is the *time it takes for a change at the input of a flip-flop to cause a change at its output*.

Each flip-flop in a counter takes a **small amount of time (Δt)** to respond to a clock pulse.

Example:

If a flip-flop takes **25 ns (nanoseconds)** to change its output after receiving a clock pulse, then **its propagation delay = 25 ns**.

What Happens in Ripple Counters?

A **ripple counter** is a type of **asynchronous counter**, meaning not all flip-flops are clocked at the same time.

The **first flip-flop (FF_o)** receives the external clock directly.

The **second flip-flop (FF_1)** receives the output of FF_o as its clock input.

The **third flip-flop (FF_2)** is clocked by FF_1, and so on.

Thus, the **clock signal "ripples"** through the flip-flops — one after another.

Email (noorabdalkareem@uomus.edu.iq)

How Propagation Delay Accumulates?

Total delay = n × tpd,

where **n** = number of flip-flops, **tpd** = propagation delay of one flip-flop.

Each flip-flop introduces a small delay. So, if you have multiple flip-flops connected in sequence:

Flip-Flop	Trigger Source	Delay Added	Cumulative Delay
FF_0	External Clock	tpd	tpd
FF_1	Output of FF_0	tpd	2 × tpd
FF_2	Output of FF_1	tpd	3 × tpd
FF_3	Output of FF_2	tpd	4 × tpd

Propagation Delay in Ripple Counters

How Propagation Delay Accumulates?

Example: Suppose a 4-bit ripple counter is built using flip-flops with: Propagation delay per flip-flop = 25 ns.

Then:

Total propagation delay = 4 × 25 ns = **100 ns**.

So, the output will be fully stable **100 ns after** the clock pulse.

Email (noorabdalkareem@uomus.edu.iq)

Propagation Delay in Ripple Counters:

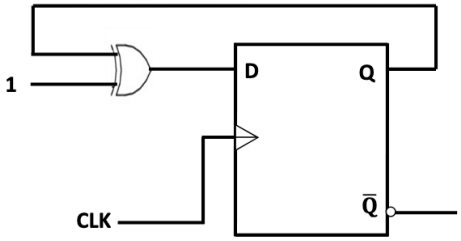


Fig 1 Toggle D flip-flop

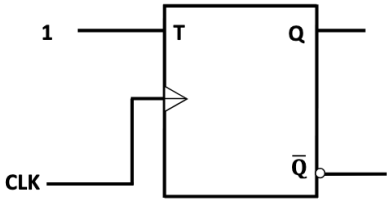


Fig 2 Toggle T flip-flop

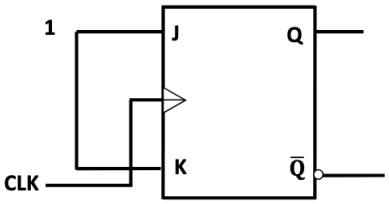
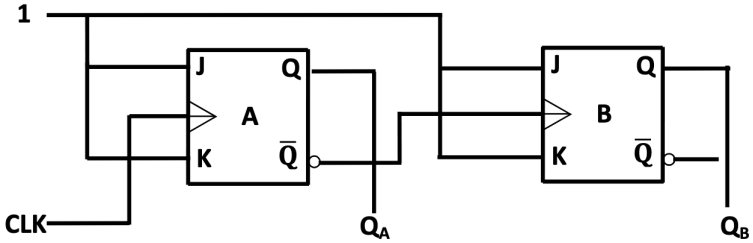


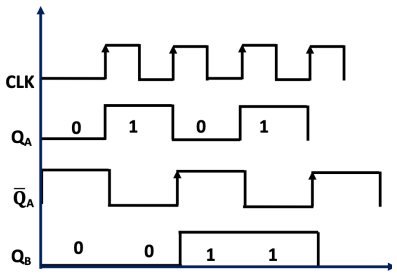
Fig 3 Toggle J-K flip-flop

Example: Design 2-bits up (**ripple counter)** asynchronous counter using J-K flip-flops with a positive edge clock pulse.

Answer:



B	A	Output
0	0	0
0	1	1
1	0	2
1	1	3



Email (noorabdalkareem@uomus.edu.iq)

Modulus (MOD):

Modulus (MOD) – the number of states it counts in a complete cycle before it goes back to the initial state. Thus, the number of flip-flops used depends on the MOD of the counter (ie; MOD-4 use 2 FF (2-bit), MOD-8 use 3 FF (3-bit), etc..)

Example: MOD-4 Ripple/Asynchronous Up-Counter.

If a counter has **n flip-flops**, each flip-flop can store 2 states (0 or 1).

$$\text{MOD} = 2^n$$

- **n = number of flip-flops (bits)**
- **MOD = total number of counts**

Modulus Counters

Maximum Count (N)

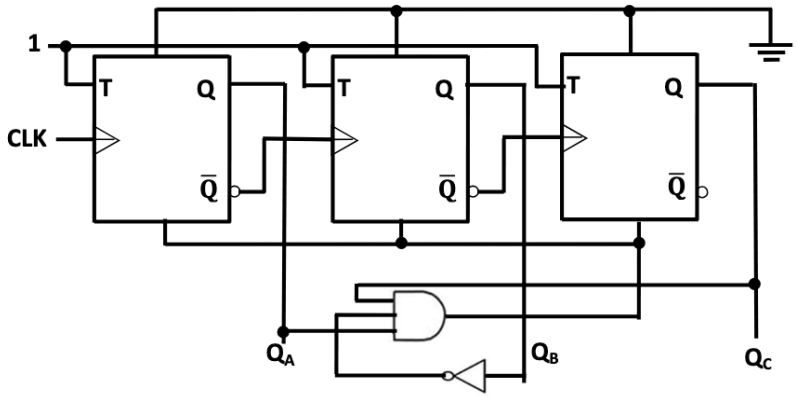
The maximum count of the counter is related to the number of (flip-flops) that build the counter which can be expressed as: $N = 2^n - 1$

For example, for four flip-flops, the maximum count is $N = 2^4 - 1 = 15$ which is equivalents to (1111) in the binary system.

Number of Flip-Flops (n)	Counter Type	Counting Range	MOD Value
1	1-bit counter	0 → 1	MOD-2
2	2-bit counter	0 → 3	MOD-4
3	3-bit counter	0 → 7	MOD-8
4	4-bit counter	0 → 15	MOD-16

This type of counter is used when the application needs certain count such as to (1001). These counters are built by controlling the clear element of the flip-flops thus when reach the required count clear all flip-flops of the counter.

Example: design (Mod 5) up counter using T flip-flops using AND gate as control element.



C	B	A	COUNT
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
0	0	0	0

Classwork during the Lab:

Design a (Mod 8) up counter using D flip-flops with a negative edge clock pulse. Draw the timing diagram and truth table for this counter.

Homework:

Design a (Mod 11) up counter using T flip-flops with negative edge clock pulse, use NAND gate as a control unit. Draw the timing diagram and truth table for this counter.

Email (noorabdalkareem@uomus.edu.iq)