

# Lecture 28

## EE 421 / CS 425

# Digital System Design

Fall 2025

**Shahid Masud**

# Topics

- Barrel Shifter
- Asynchronous Sequential Circuits
- Types of Asynchronous Circuits:
  - Pulse mode Asynchronous Circuits
- Analysis of Pulse Mode Circuits
- Design of Pulse Mode Circuits

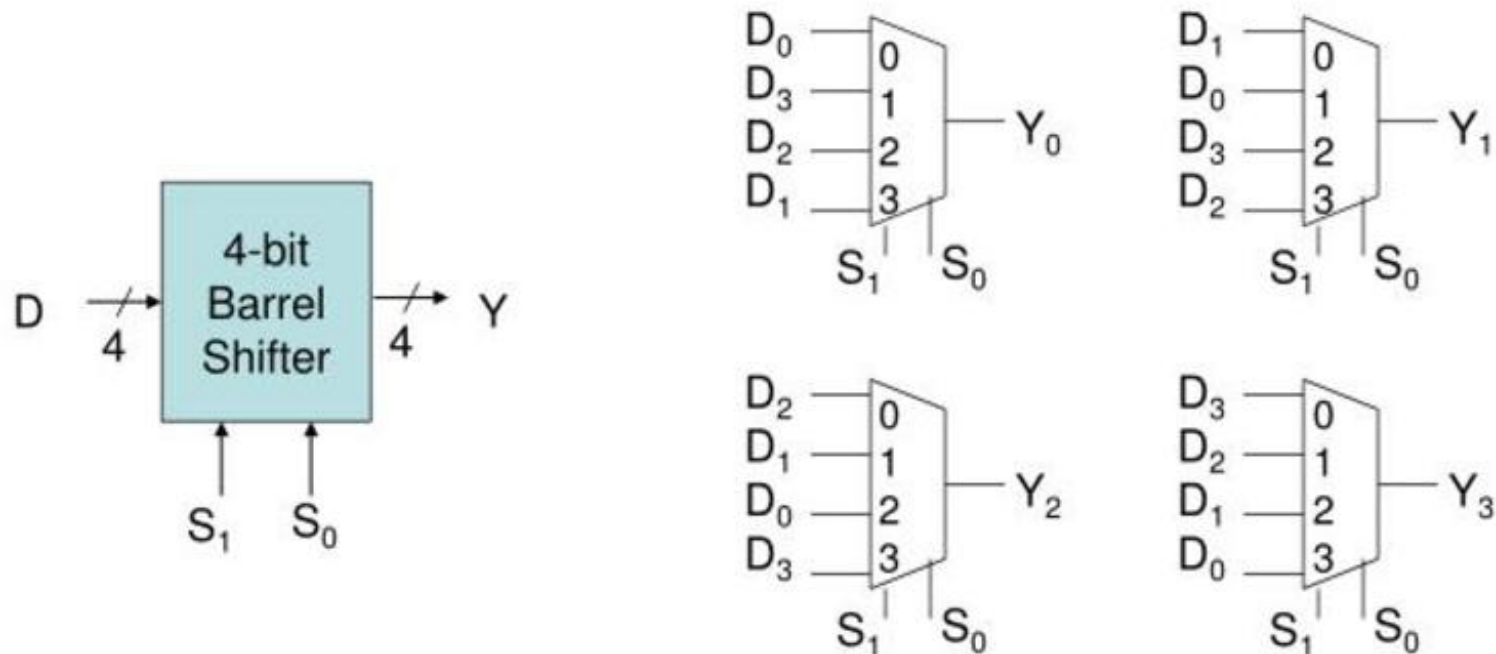
Quiz 6 Today

# Barrel Shifter

# What is a Barrel Shifter?

## Barrel Shifter

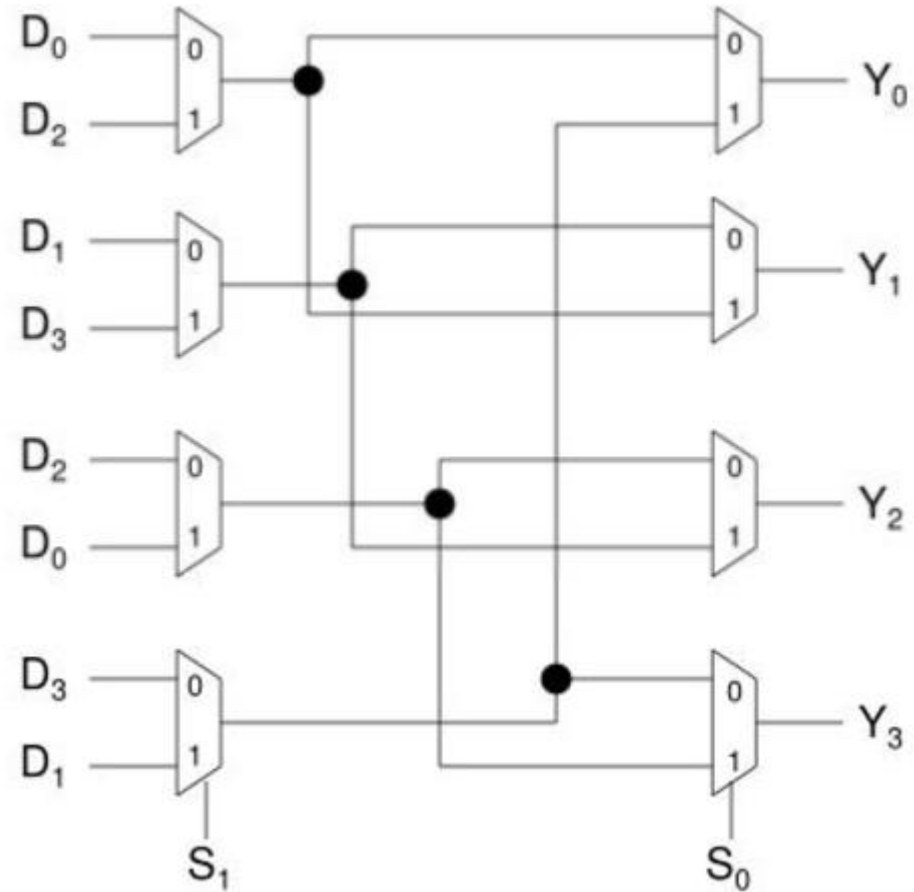
- Example: 4-bit barrel shifter, can shift input by 0, 1, 2 or 3 bits.
- $S_0S_1$  determines how many bits to shift (left).
- Use 4 4-to-1 multiplexers.



# An Architecture of a Barrel Shifter

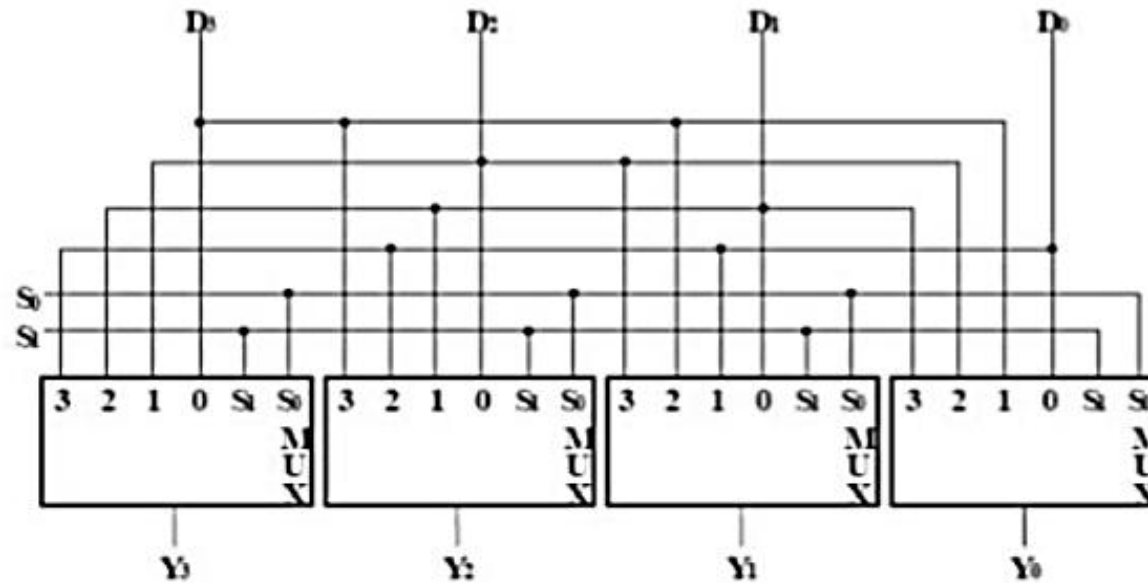
- 32-bit barrel shifter: can use 32 32-to-1 multiplexers
- However, large fan-in undesirable. So, use layers of multiplexers

Example:  
Use 2 layers of 4  
2-to-1 multiplexers  
for 4-bit barrel  
shifter



# A 4-bit Barrel Shifter Design

## 4-Bit Barrel Shifter



- A rotate is a shift in which the bits shifted out are inserted into the positions vacated
- The circuit rotates its contents left from 0 to 3 positions depending on Selector S.

Function Table for 4-Bit Barrel Shifter

Select		Output				Operation
S <sub>1</sub>	S <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	
0	0	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	No rotation
0	1	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	D <sub>3</sub>	Rotate one position
1	0	D <sub>1</sub>	D <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	Rotate two positions
1	1	D <sub>0</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	Rotate three positions

Note that a left rotation by three (3) positions is the same as a right rotation by one position in this 4 bit barrel shifter

# Asynchronous Digital Design

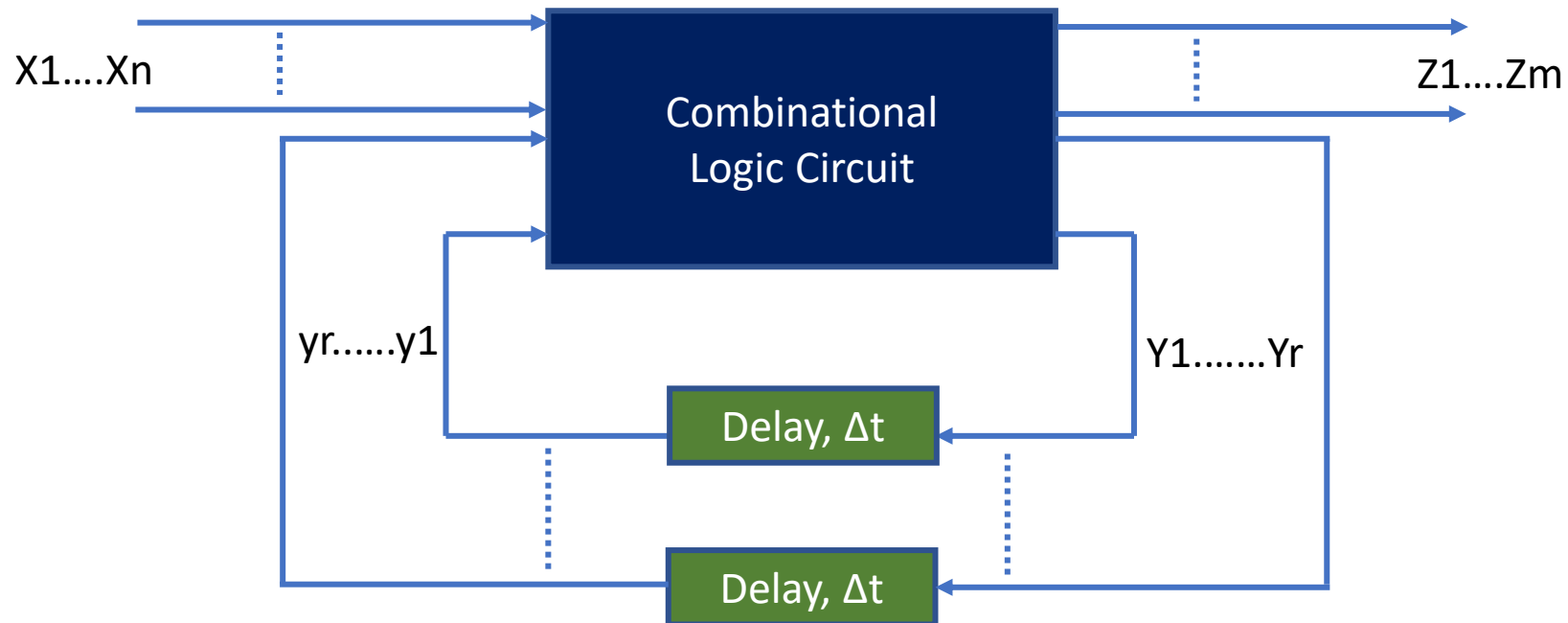
# Asynchronous Sequential Circuits

- A clock is **not** utilized to synchronize state changes
- Only external inputs and current states are available to cause state changes



# Fundamental Mode Asynchronous Circuits

- These circuits have **level inputs** and **unclocked memory elements**
- Uses inherent delays present in logic circuit elements
- Assume that **each delay element has same delay  $\Delta t$**

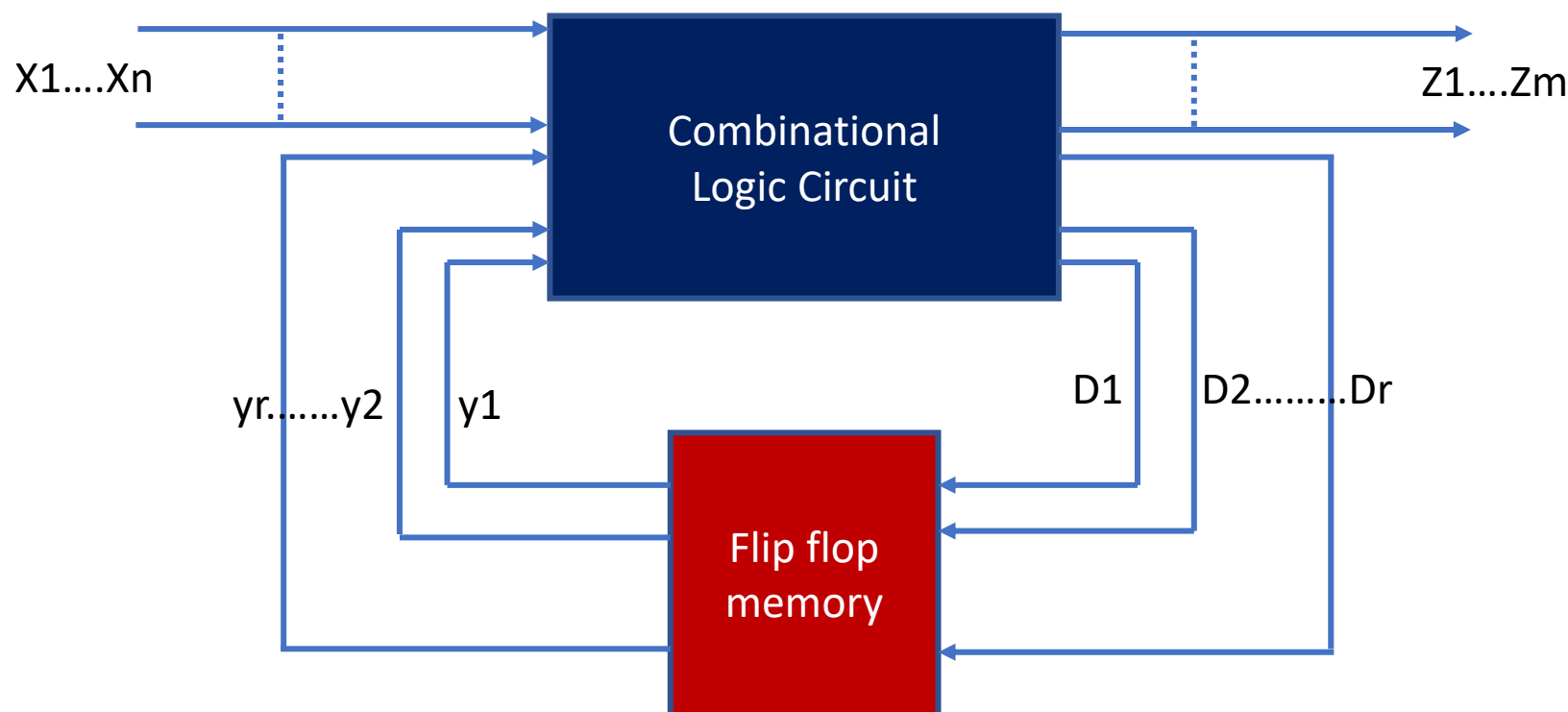


# Other conditions for Fundamental Mode cct

- All delays in a circuit branch are lumped to a single  $\Delta t$
- The delay is always  $\Delta t$
- Inputs are restricted so that only one input variable is allowed to change value at a given instant to ensure correct behavior
- For predictable operations, input changes should be spaced in time by at least  $\Delta t$ ; the time needed for the circuit to settle down to a stable state following an input change

# Pulse Mode Asynchronous Circuits

- Pulses will not occur simultaneously on two or more input lines
- Memory element transitions are limited only by input pulses
- Input variables are used either only in uncomplemented or in the complemented forms, but not both together



# Pulse Mode Asynchronous Sequential

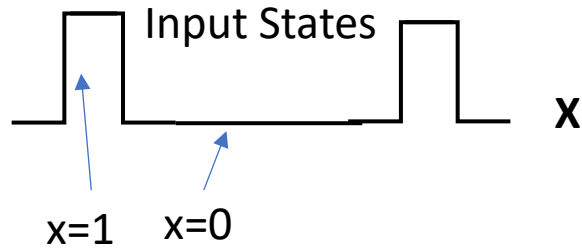
## Pulse Mode Asynchronous Circuits :

- In the pulsed mode, the input variables are allowed to be applied in the form of pulses, rather than in the form of levels.
- But the width of input pulses is a critical parameter. There are two restrictions on the width of the input pulses.
  - The first requirement is that the pulses should be long enough so that the circuit can respond to them.
  - The second requirement is that the pulses should not be too long so that they are still present after the new secondary state is reached.
- The base of calculating the minimum pulse width is the propagation delay of the excitation logic.
- The maximum pulse width is calculated based on the total propagation delay through the excitation logic and the memory.

# Conditions for Pulse mode circuits

- Input pulses should be spaced in time by at least the response time of the slowest memory element
- No memory element will be in process of changing state when a new input pulse arrives

# Analysis of a Pulse Mode Asynchronous Sequential Circuit

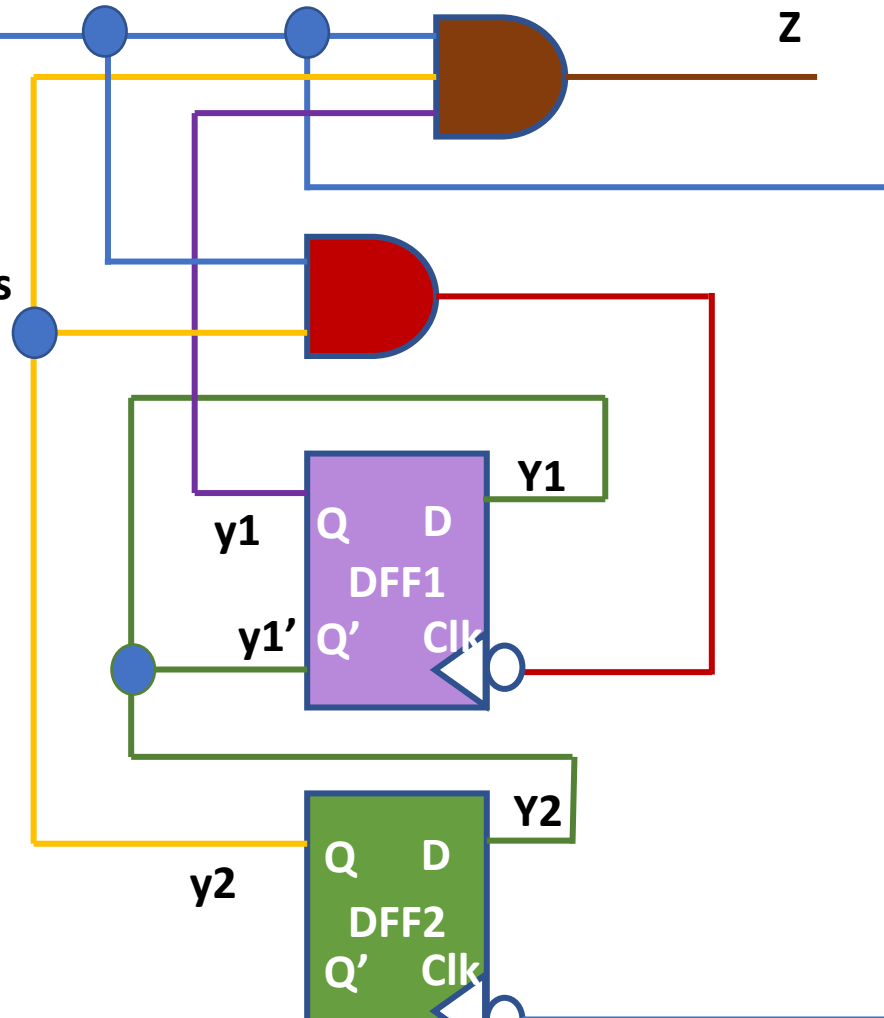


Input X is made up of asynchronous pulses

Initial state  $y_1=y_2=0$

Present states =  $y_1, y_2$  (small y)

Next states =  $Y_1, Y_2$  (capital Y)



Logic Equations:

$$D1=y1'$$

$$D2=y1'$$

$$Z=X.y1.y2$$

$$C1=X.y2$$

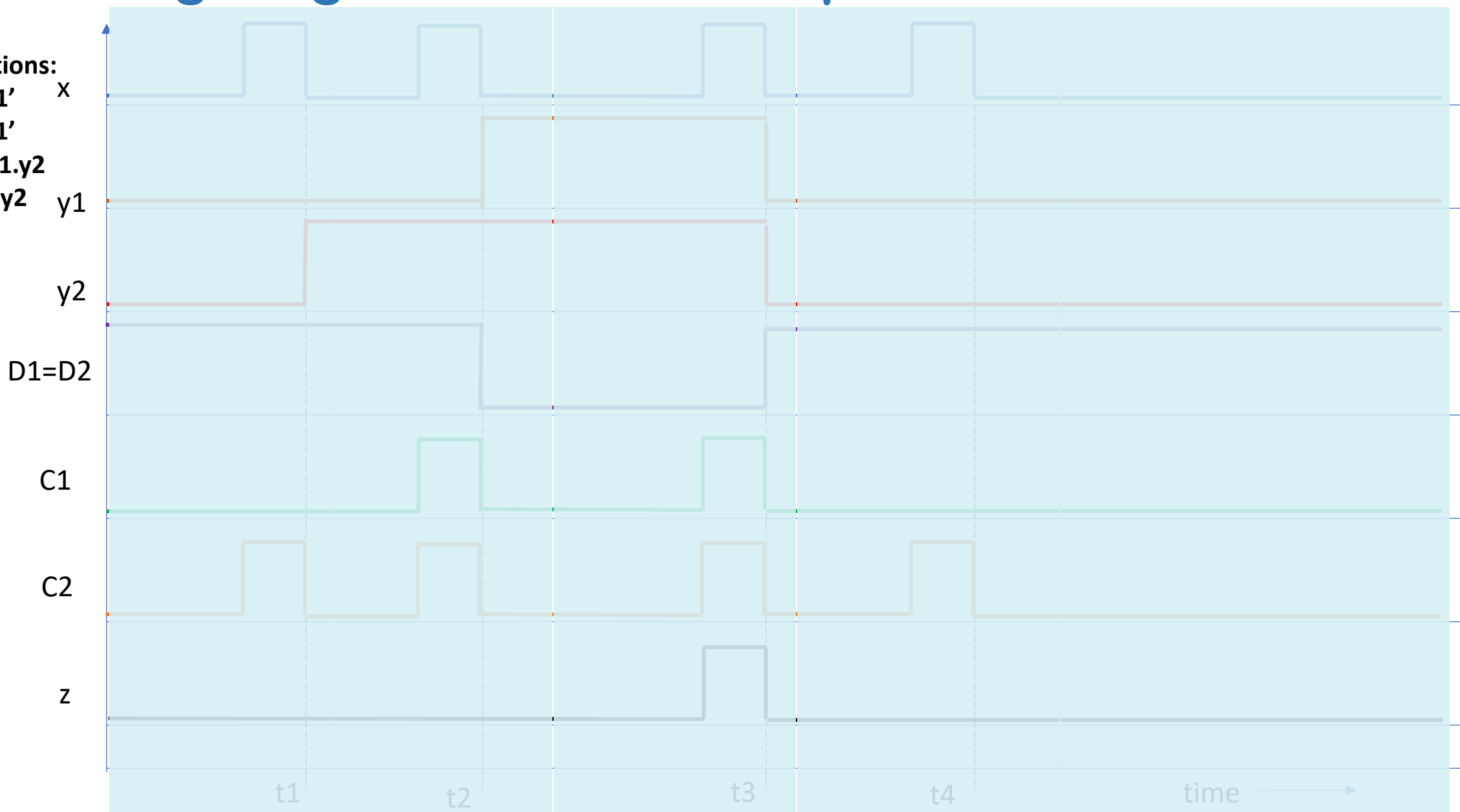
$$C2=X$$

# Contd: Analysis Pulse Mode Asynch Seq Cct

- Define:
- Inputs:
  - $I_0$  = no pulse on X
  - $I_1$  = pulse on X
- States:
  - A = 00 (y1y2)
  - B = 01
  - C = 10
  - D = 11
- Outputs:
  - Z = 0
  - Z = 1

# Timing Diagram of Pulsed Sequential Circuit

Logic Equations:  
 $D1 = y1'$   
 $D2 = y1'$   
 $Z = X.y1.y2$   
 $C1 = X.y2$   
 $C2 = X$



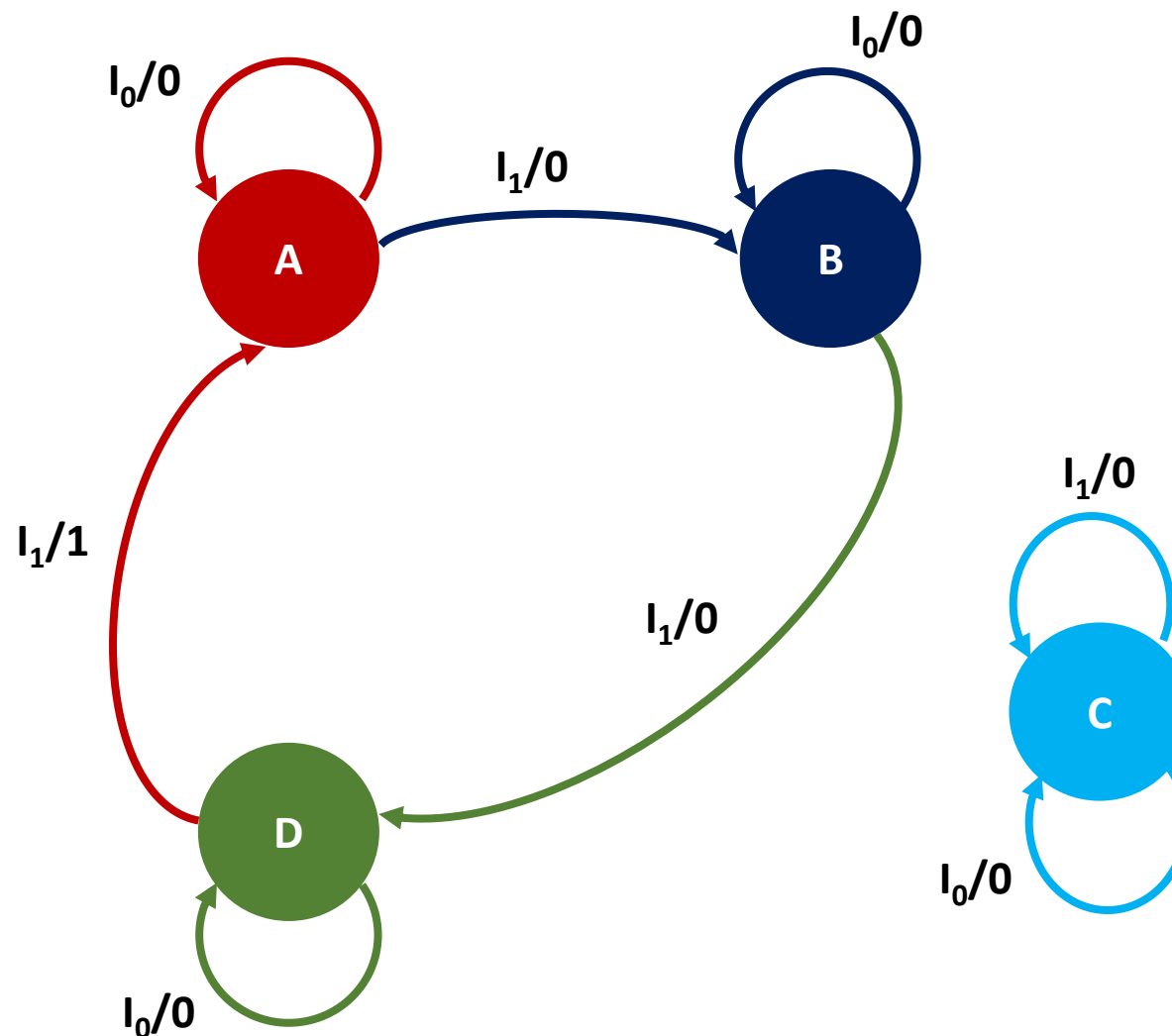


# State Table and STG for Pulsed Sequential Circuit

Present state	$I_0$	$I_1$
A	A/0	B/0
B	B/0	D/0
C	C/0	C/0
D	D/0	A/1

Next State / Output

$I_0$  = no pulse on X  
 $I_1$  = pulse on X



# K-maps

Present state	I <sub>0</sub>	I <sub>1</sub>
A	A/0	B/0
B	B/0	D/0
C	C/0	C/0
D	D/0	A/1

For D1

y1y2	X=0	X=1
00	1	1
01	1	1
11	0	0
10	0	0

- Inputs:
  - I<sub>0</sub> = no pulse on X
  - I<sub>1</sub> = pulse on X
- States:
  - A = 00 (y1y2)
  - B = 01
  - C = 10
  - D = 11
- Outputs:
  - Z = 0
  - Z = 1

For C1

y1y2	X=0	X=1
00	0	0
01	0	1
11	0	1
10	0	0

For D2

y1y2	X=0	X=1
00	1	1
01	1	1
11	0	0
10	0	0

For C2

y1y2	X=0	X=1
00	0	1
01	0	1
11	0	1
10	0	1

For Y1

y1y2	X=0	X=1
00	0	0
01	0	1
11	1	0
10	1	1

For Y2

y1y2	X=0	X=1
00	0	1
01	1	1
11	1	0
10	0	0

For Z

y1y2	X=0	X=1
00	0	0
01	0	0
11	0	1
10	0	0

# Design of Pulse Mode Asynchronous Circuits

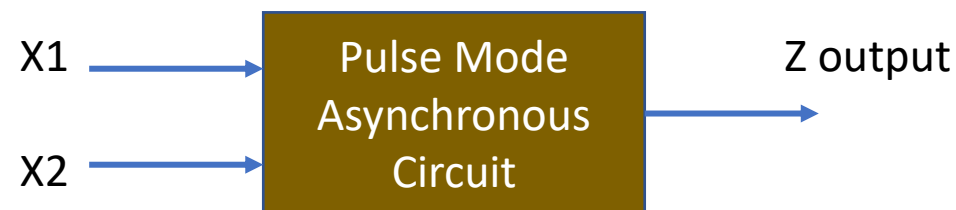
- To remember:
  - No clock is present
  - Inputs occur as pulses on any one input line at one time
  - Only un-complemented or complemented forms are used as inputs throughout
  - Triggering of Latches or flip flops is accomplished by utilizing information in input pulses

# Design Steps in Pulse Mode Asynchronous

1. Derive a state diagram or a state table
2. Minimize the state table through State Reduction
3. Choose a state assignment and generate the transition/output table
4. Select the type of flipflop and determine the excitation equations
5. Determine the output equations
6. Choose appropriate circuit elements and draw the circuit diagram

# Example – Design of Pulse Mode circuit

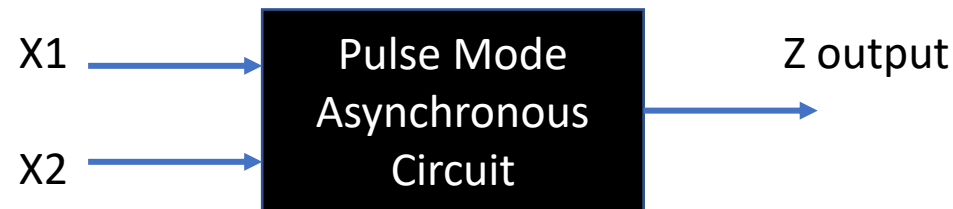
- Problem Statement: Design a pulse mode circuit that has two input lines, X1 and X2. There is one output line Z.
- The circuit should produce an output pulse to coincide with the last input pulse in the sequence X1 – X2 – X2 (Sequence Detector)
- No other input sequence should produce an output pulse



Block diagram of the desired system

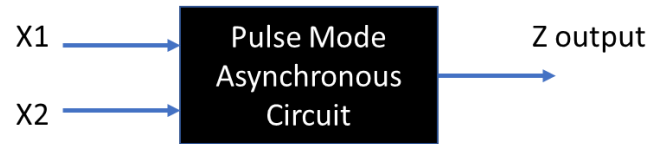
# Example – Pulse mode – Step 1

- Define the following three states of the circuit:
- A: indicates that the last input was X1
- B: indicates that the sequence X1 – X2 has occurred
- C: indicates that the sequence X1 – X2 – X2 has occurred



Block diagram of the desired system

# Example – Pulse Mode – STG and State Table

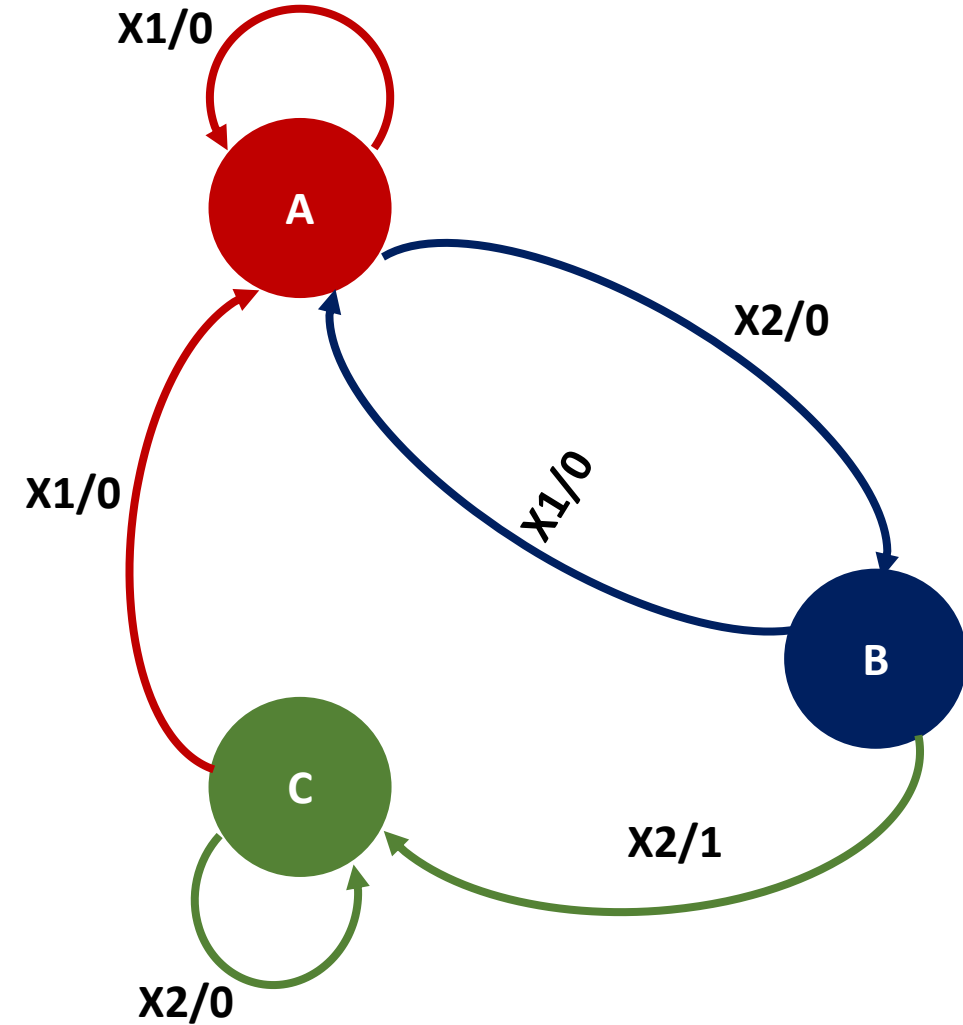


**A** indicates Last Input was X1

**B** indicates sequence X1 – X2 has appeared

**C** indicates sequence X1 – X2 – X3 has appeared

Present State	Next State/Output	
	X1 appears	X2 appears
A	A/0	B/0
B	A/0	C/1
C	A/0	C/0

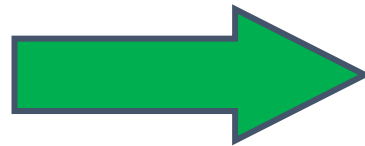


Mealy FSM

# Example Pulse Mode – State Assignments

Assign States: A=00, B=01, C=10 and generate the transition/output table

Present State	Next State/Output	
	When X1 appears	When X2 appears
A	A/0	B/0
B	A/0	C/1
C	A/0	C/0



Present State	Next State/Output Y1Y2/Z	
	y1y2	When X1 appears When X2 appears
00	00/0	01/0
01	00/0	10/1
10	00/0	10/0



# Example – Pulse mode implement using T FF

State Table for Y1

Present State	Next State/Output Y1Y2/Z	
y1y2	When X1 appears	When X2 appears
00	00/0	01/0
01	00/0	10/1
10	00/0	10/0

Present State	Next State	
y1y2	X1	X2
00	0	0
01	0	1
11	d	d
10	0	1

State Table for T1

Present State	Next State Y1, Y2	
y1y2	When X1	When X2
00	0	0
01	0	1
11	d	d
10	1	0

$$T1 = X1.y1 + X2.y2$$

# Example – Pulse Mode Implement using TFF (contd)

State Table for Y2

Present State	Next State	
y1y2	When X1	When X2
00	0	1
01	0	0
11	d	d
10	0	0

State Table for T2

Present State	Next State	
y1y2	When X1	When X2
00	0	1
01	1	1
11	d	d
10	0	0

Present State	Next State/Output Y1Y2/Z	
y1y2	When X1 appears	When X2 appears
00	00/0	01/0
01	00/0	10/1
10	00/0	10/0

State Table for Z output

Present State	Next State	
y1y2	When X1	When X2
00	0	0
01	0	1
11	d	d
10	0	0

$$T2 = X1.y2 + X2.y1'$$

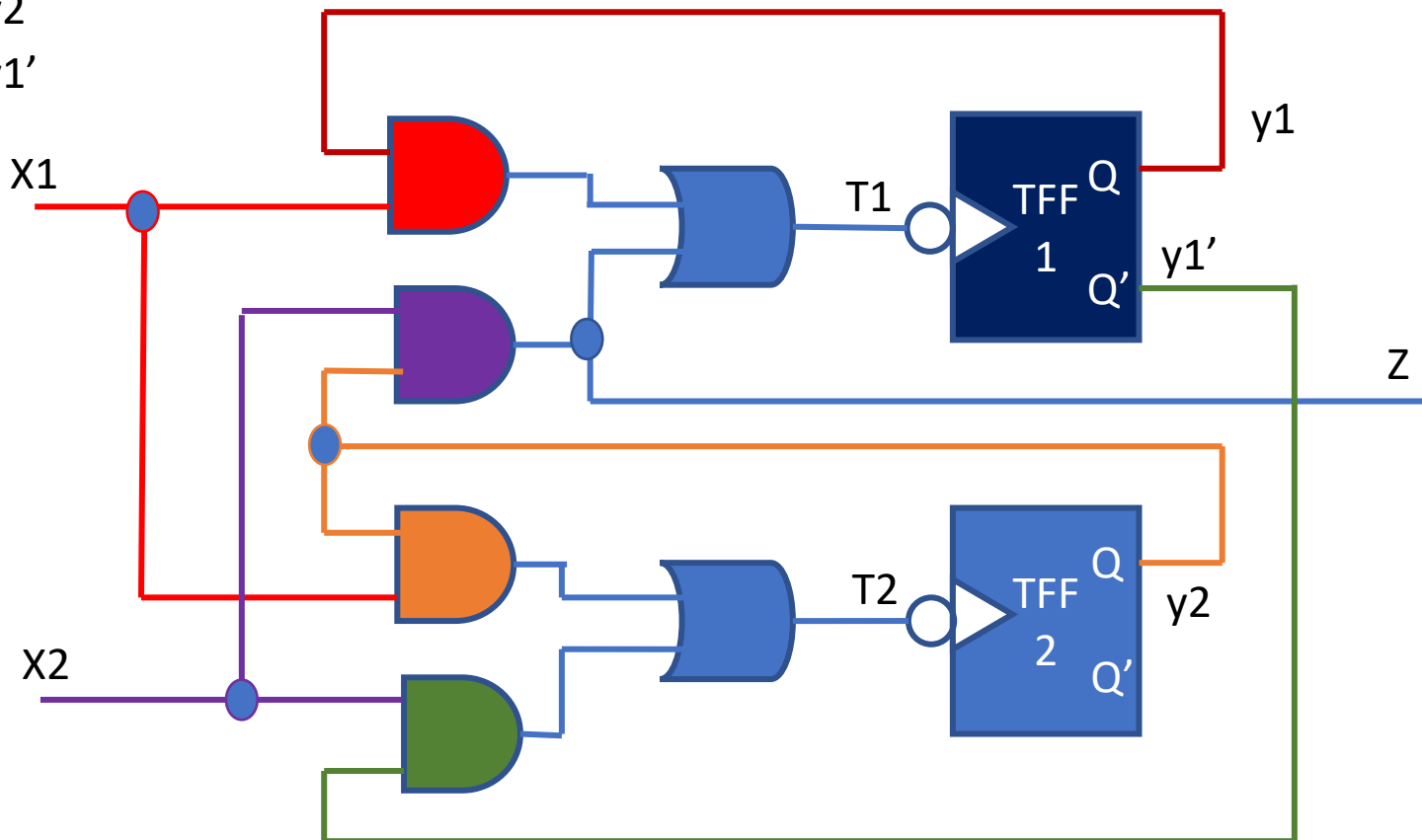
$$Z = X2.y2$$

# Example – Pulse mode Circuit diagram

$$T1 = X1.y1 + X2.y2$$

$$T2 = X1.y2 + X2.y1'$$

$$Z = X2.y2$$



T1, T2 are in toggle mode  
Toggle Input is enabled to '1'