SEQUENTIAL CIRCUITS

- What is a sequential circuit?
- Difference between combinatorial and sequential circuits…

The basic and fundamental element of sequential logic is the *bistable multivibrator*.

Biestáveis	Possuem dois estados estáveis e a capacidade de armazenar informação (1 biestável pode armazenar 1 bit).
Monoestáveis	Possuem apenas um estado estável. Normalmente são utilizados para temporização ou em linhas de atrasos em sistemas digitais.
Aestáveis	Não possuem nenhum estado estável. Este tipo de circuitos oscila livremente entre os seus dois estados possíveis. Uma aplicação deste tipo de multivibradores é como geradores de sinais de *clock*.

The class of bistable multivibrators can be divided into:

Biestáveis
- **Assíncronos**
- **Síncronos**
  - **Activados por Nível**
  - **Activados por Flanco**

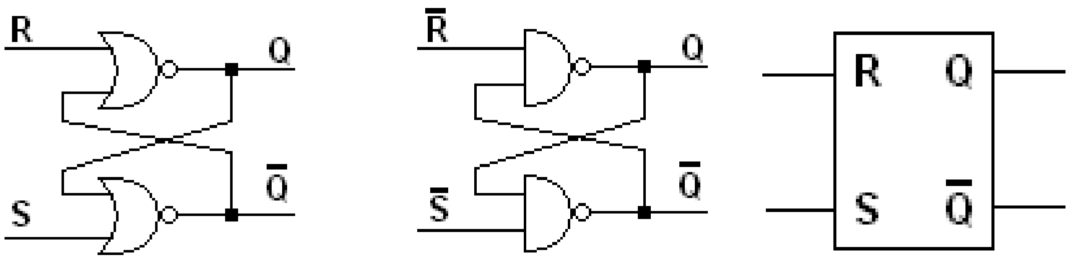
• To synchronous bistable activated by edge is given the name Flip-Flop's.

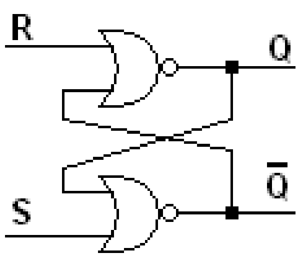
•All others will be called Latch's.

Asynchronous bistables

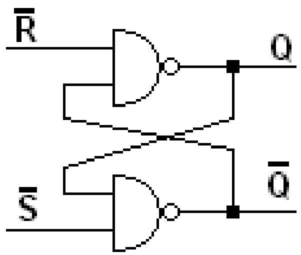
<u>Type **RS**</u>

NOTE: RESET and PRESET





NOR

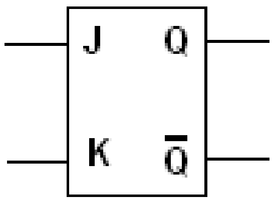


NAND

R	S	Q
0	0	$\mathbf{Q_{n+1} = Q_n}$ (1)
0	1	1
1	0	0
1	1	Proibida (2)

R	S	Q
0	0	Proibida (2)
0	1	0
1	0	1
1	1	$\mathbf{Q_{n+1} = Q_n}$ (1)

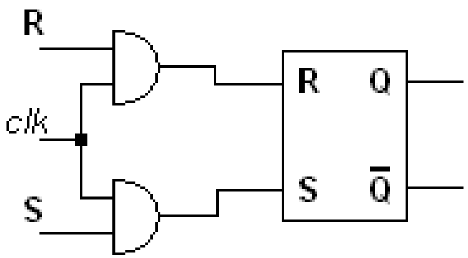
Type **JK**

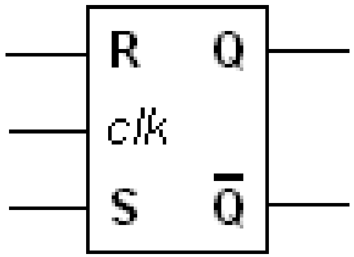


J	K	Q
0	0	$\mathbf{Q_{n+1} = Q_n}$
0	1	0
1	0	1
1	1	$\mathbf{Q_{n+1} = \overline{Q}_n}$ (1)

Bistable Synchronous Enabled by Level

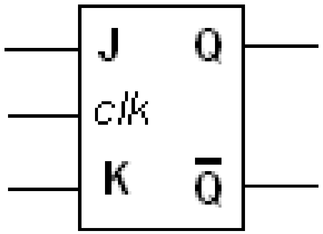
<u>Type **RS**</u>





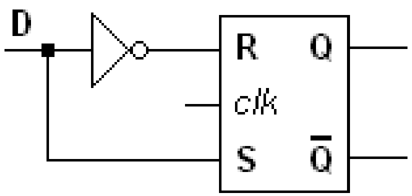
clk	**S**	**R**	**Q**
0	0	0	$\mathbf{Q_{n+1} = Q_n}$
0	0	1	$\mathbf{Q_{n+1} = Q_n}$
0	1	0	$\mathbf{Q_{n+1} = Q_n}$
0	1	1	$\mathbf{Q_{n+1} = Q_n}$
1	0	0	$\mathbf{Q_{n+1} = Q_n}$
1	0	1	1
1	1	0	0
1	1	1	Proibido

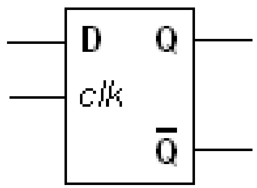
Type **JK**



clk	**J**	**K**	**Q**
0	0	0	$\mathbf{Q_{n+1} = Q_n}$
0	0	1	$\mathbf{Q_{n+1} = Q_n}$
0	1	0	$\mathbf{Q_{n+1} = Q_n}$
0	1	1	$\mathbf{Q_{n+1} = Q_n}$
1	0	0	$\mathbf{Q_{n+1} = Q_n}$
1	0	1	0
1	1	0	1
1	1	1	$\mathbf{Q_{n+1} = \overline{Q}_n}$

Type **D**





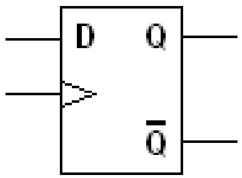
clk	**D**	**Q**
0	0	$\mathbf{Q_{n+1} = Q_n}$
0	1	$\mathbf{Q_{n+1} = Q_n}$
1	0	0
1	1	1

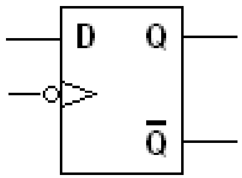
Synchronous bistable activated by level can cause problems when the frequencies involved are high.

Solution: bistable activated at the flank

Bistable Synchronous Enabled by Flank

Type **D**

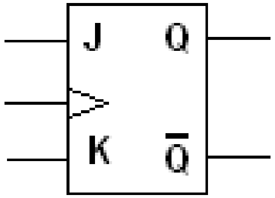


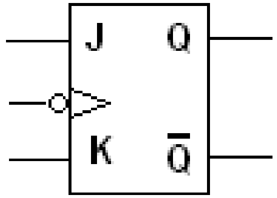


clk	**D**	**Q**
$\uparrow$	0	0
$\uparrow$	1	1
X	X	$Q_{n+1} = Q_n$

clk	**D**	**Q**
$\downarrow$	0	0
$\downarrow$	1	1
X	X	$Q_{n+1} = Q_n$

<u>Typo **JK**</u>





clk	J	K	Q
$\uparrow$	0	0	$Q_{n+1} = Q_n$
$\uparrow$	0	1	0
$\uparrow$	1	0	1
$\uparrow$	1	1	$Q_{n+1} = \bar{Q}_n$
X	X	X	$Q_{n+1} = Q_n$

clk	J	K	Q
$\downarrow$	0	0	$Q_{n+1} = Q_n$
$\downarrow$	0	1	0
$\downarrow$	1	0	1
$\downarrow$	1	1	$Q_{n+1} = \bar{Q}_n$
X	X	X	$Q_{n+1} = Q_n$

Counters

• The count digital systems are one of the main applications for bistable mutivibradores.

• Composed by a set of flip-flop's mounted in cascade that evolve their states in a designed predetermined sequence.

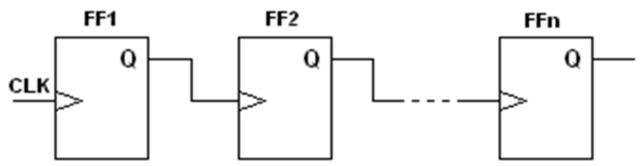
A counter module: the number of synchronism signal cycles after which the counter returns to its initial state.

The capacity of a counter is the highest number, expressed in any binary code, which can be represented at its outputs

Contadores

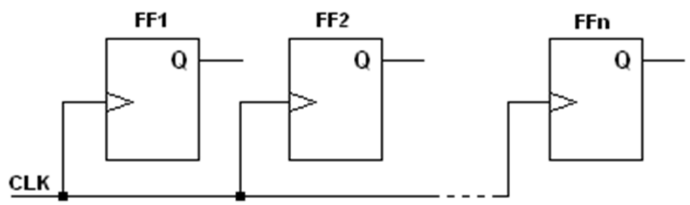
Assíncronos

Á excepção do primeiro flip-flop, cujo sinal de sincronismo é o sinal de clock, a saída de cada flip-flop será o sinal de relógio do flip-flop seguinte.



Síncronos

O sinal de relógio é aplicado simultaneamente a todos os flip-flop's, i.e. as saídas de todos os biestáveis são actualizadas simultaneamente.

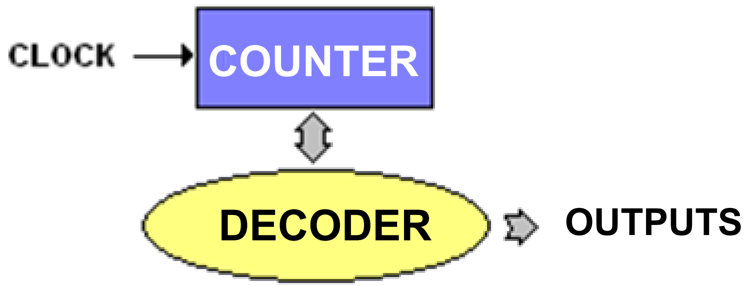


In asynchronous counters the propagation time is greater than the synchronous counters **(why?)**

A generic counter does the count from 0 to $2^n - 1$

where *n* denotes the number of involved bistables.

• Change the value of the module counter or its capacity.

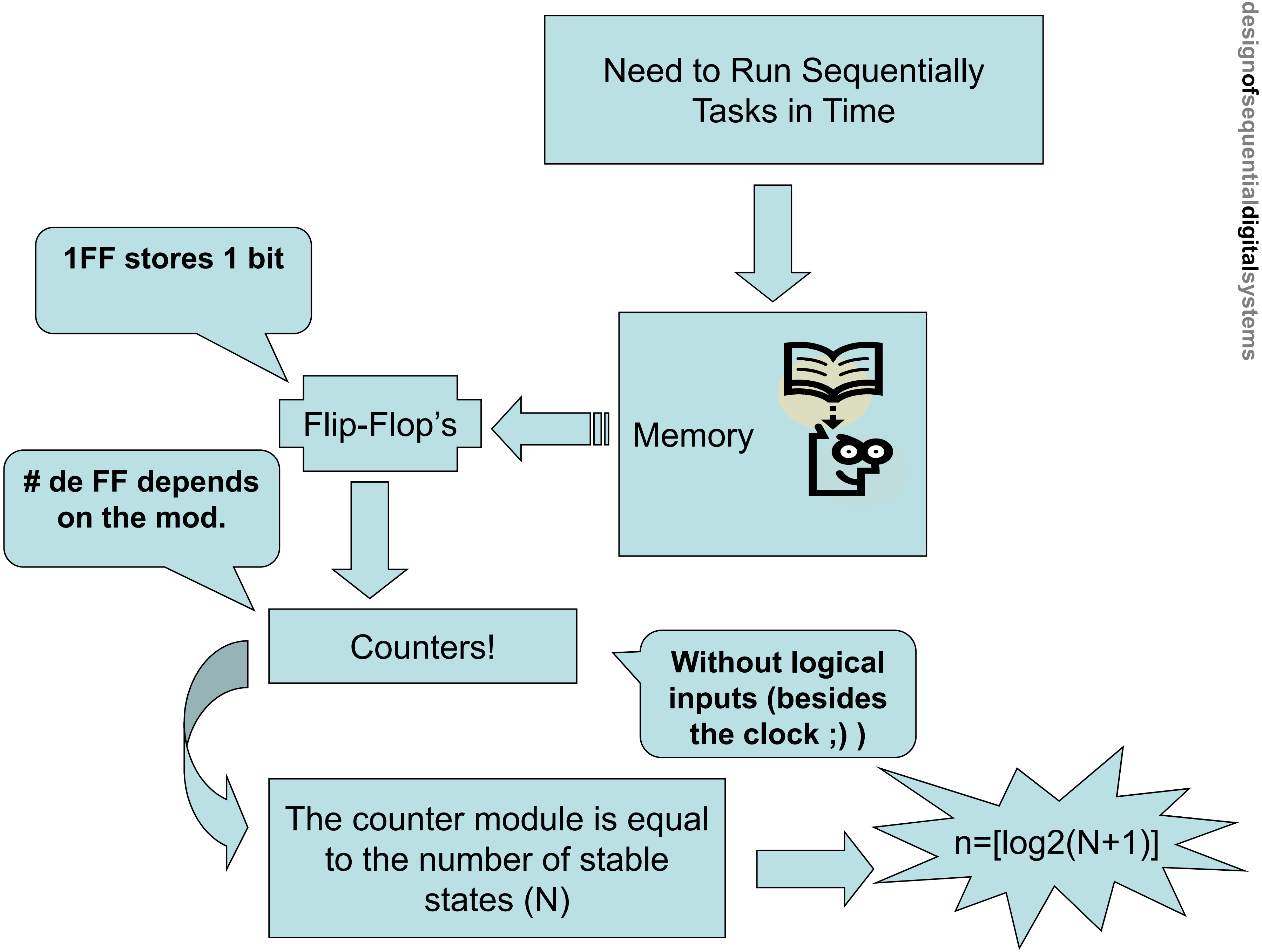


• The decoder is usually a combinatorial circuit developed from the present counter states and the states that really want as output.

COUNTERS DESIGN:

EXAMPLE #1 : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 0, 1, ... (asynchronous counter)

EXAMPLE #2 : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 0, 1, ... (synchronous counter JK)

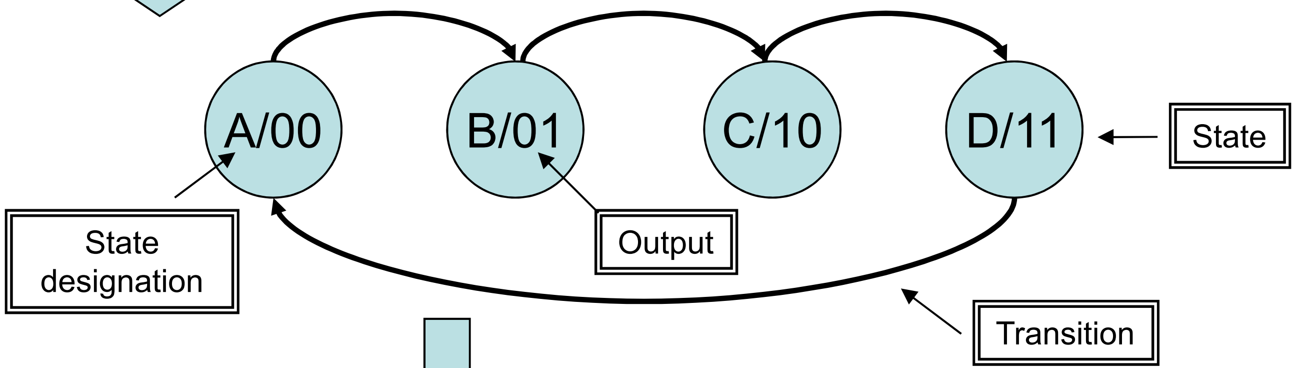

Need to Run Sequentially Tasks in Time
Memory
1FF stores 1 bit
Flip-Flop’s
de FF depends on the mod.
Counters!
Without logical inputs (besides the clock ;))
The counter module is equal to the number of stable states (N)
n=[log2(N+1)]

ex. synchronous counter mod-4 and capacity 3 : 0, 1, 2, 3, 0, 1, ...

How many Flip-Flop's??

States Diagram

Finite States Machines

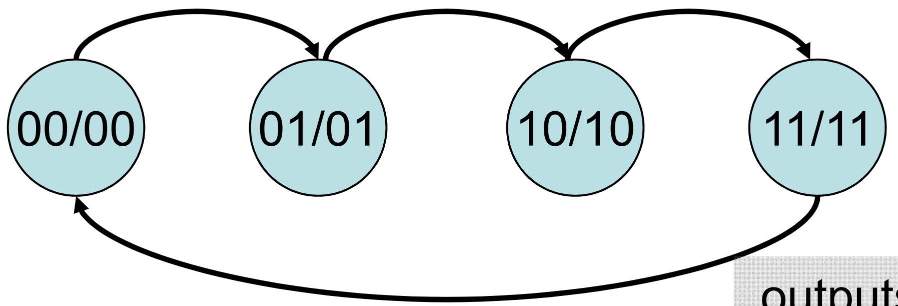


Associate with each of the states (A, B, C and D) a system state (FF's)

States Atribution

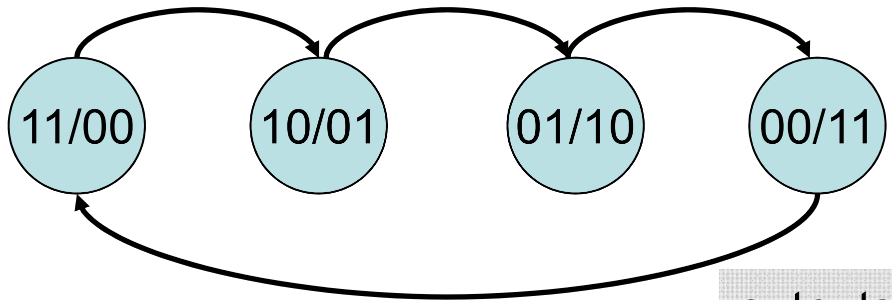
ex:

A	00
B	01
C	10
D	11



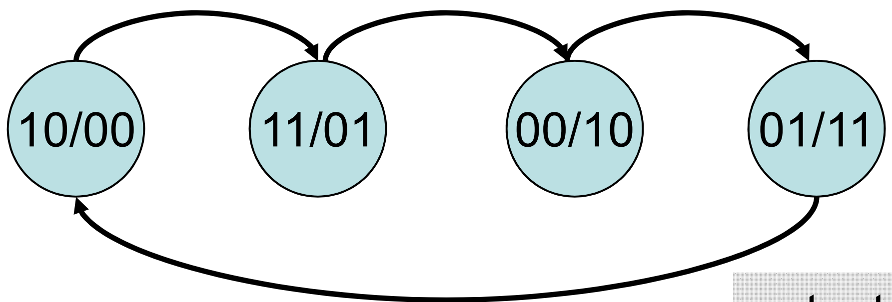
outputs=[Q1 : Q0]

A	11
B	10
C	01
D	11



outputs=[$\overline{Q1}$: $\overline{Q0}$]

A	10
B	11
C	00
D	01



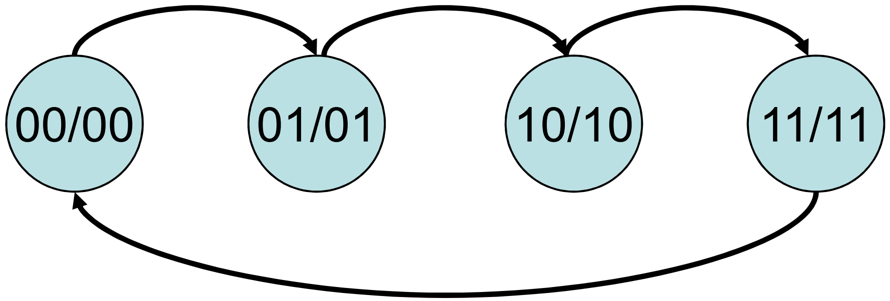
outputs=[$\overline{Q1}$: Q0]

In projects carried out it is always considered:
OUTPUTS = STATES

ex. counter that performs the counting -> 15, 16, 17, 15, 16, 17,...

Different atributions=> diferente logic circuits

From the states diagram...



outputs=[Q1 : Q0]

To the states transition table

Present states		Next states		Outputs	
Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}		
0	0	0	1	0	0
0	1	1	0	0	1
1	0	1	1	1	0
1	1	0	0	1	1

Memory elements:
JK, SR, D

Assuming Flip-flop's D...

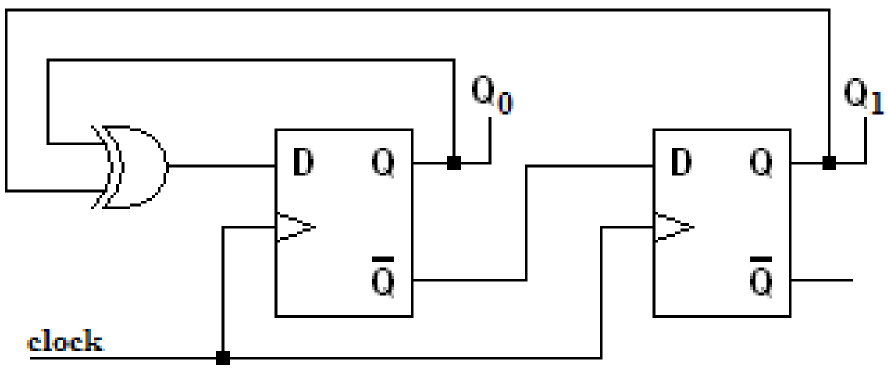
Present states		Next states		Output		FF's inputs	
Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}	(MSB)	(LSB)	D_1	D_0
0	0	0	1	0	0	0	1
0	1	1	0	0	1	1	0
1	0	1	1	1	0	1	1
1	1	0	0	1	1	0	0

From excitation equations ...

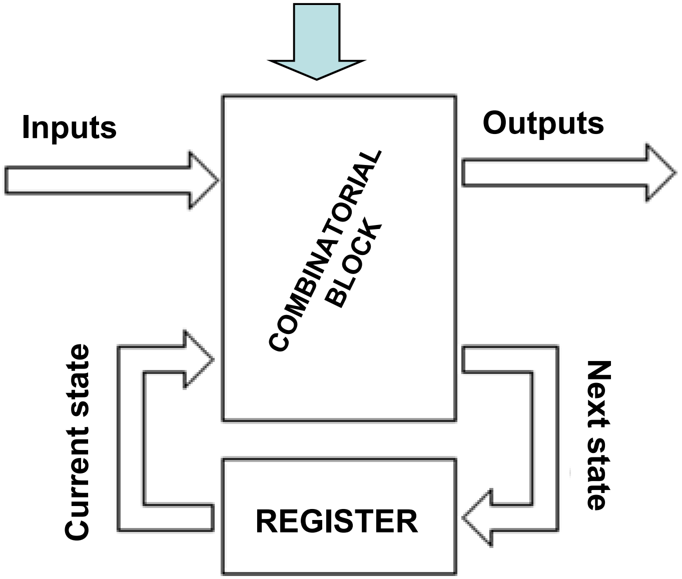
$$D_0 = Q_1 \oplus Q_0$$

$$D_1 = \bar{Q}_0$$

To the logic circuit!



A sequential system can have other degrees of freedom than the "clock"!!!!!!



Combinatorial block
- set of logic gates
- has input and output lines
  + responsable for information admission
  + change of any system physical state to control

A sequential process requires storing the system previous states - **register block**

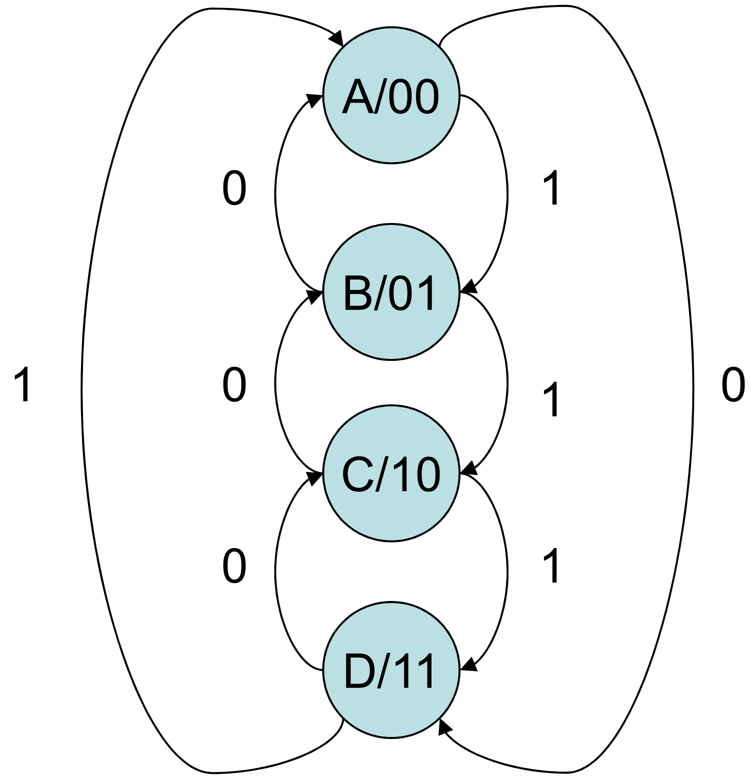
1. Convert verbal specifications for a state diagram
2. Build the States Table
3. Selection of Memory Elements
4. Simplification of Excitation Equations
5. Implementation of Sequential Circuit

EXAMPLE #1:
Up/Down counter mod-4

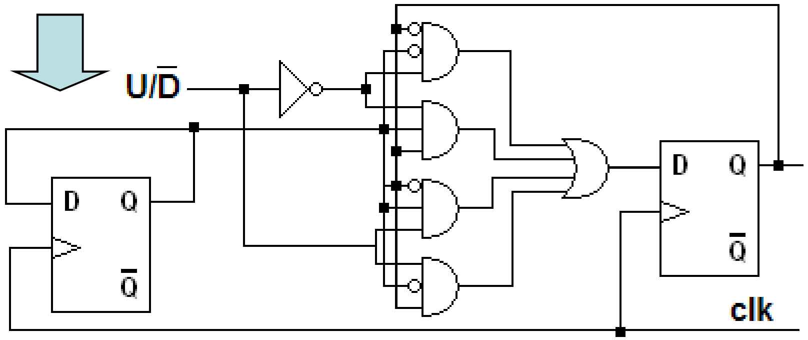
Input Variable:

- U/~D ('1' upward counting e '0' downward counting

Current state		Input	Next state	
Q_1^n	Q_0^n	$U/\overline{D}$	Q_1^{n+1}	Q_0^{n+1}
0	0	0	1	1
0	0	1	0	1
0	1	0	0	0
0	1	1	1	0
1	0	0	0	1
1	0	1	1	1
1	1	0	1	0
1	1	1	0	0

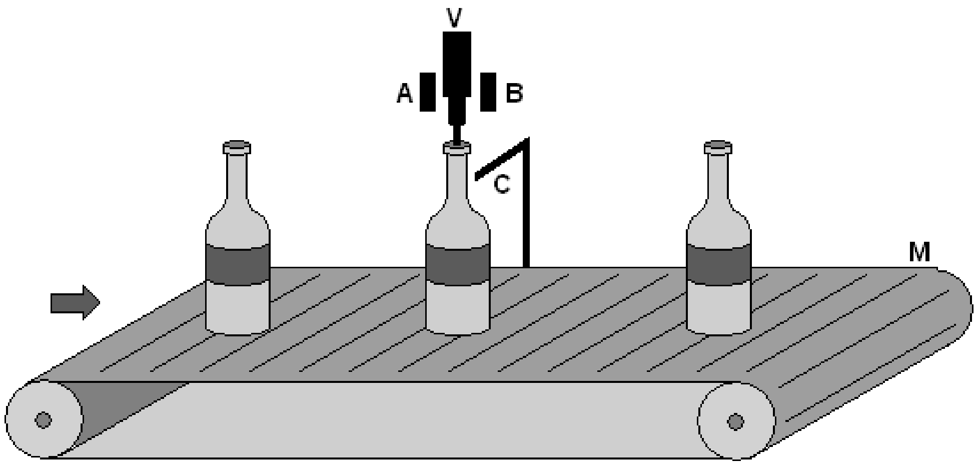


Flip-Flop's D

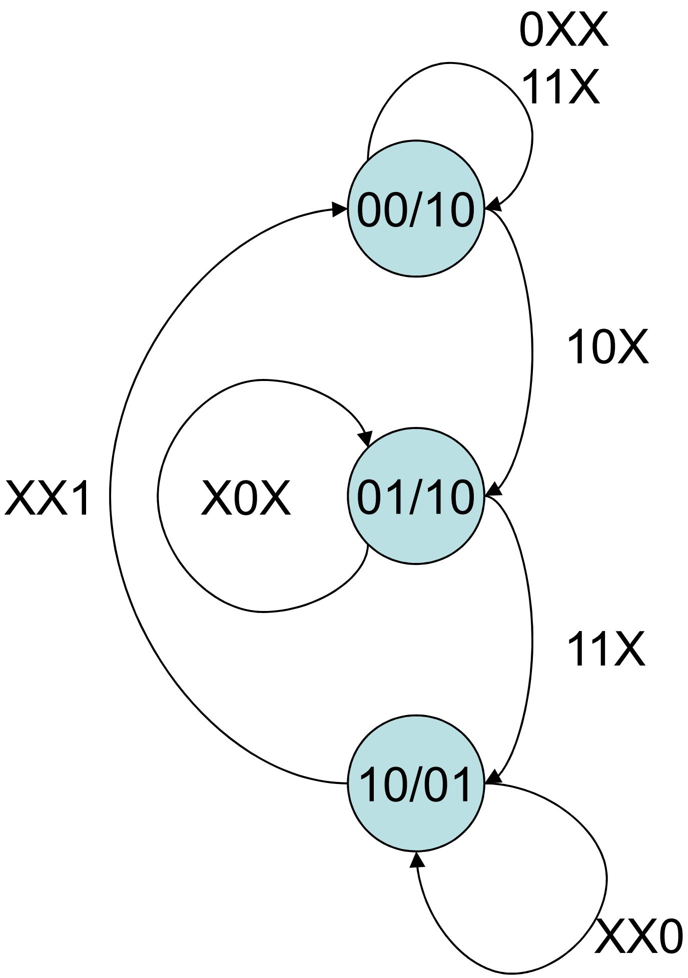


EXAMPLE #2
Production line automatization

Consider a production step of a water bottling plant. Is intended to develop a digital circuit capable of controlling in an automatic way, the filling process of the bottles. For this, the system has three sensors, two of position (**A** and **B**) and a level (**C**), and two actuators, an electro-valve **V** and the motor of the conveyor **M**. Initially the conveyor moves until a bottle assumes the filling position. It is considered that the bottle is well positioned when the **B** sensor becomes active after **A**. At that moment the engine stops and the valve opens starting the filling operation. This operation is terminated when the level sensor is activated.



States Diagram

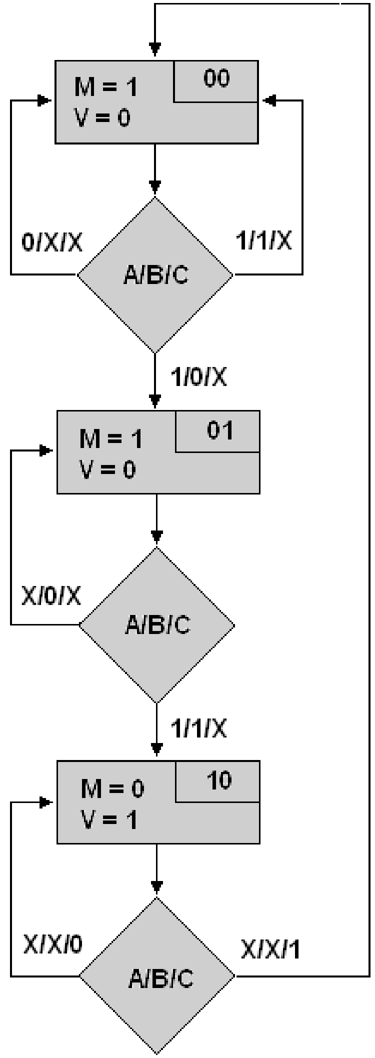


Flowchart

States : Rectangles

Transmission lines

Decision variables: Diamond



States Table

Current state		Inputs			Next state		Outputs	
Q_1^n	Q_0^n	A	B	C	Q_1^{n+1}	Q_0^{n+1}	M	V
0	0	0	X	X	0	0	1	0
0	0	1	1	X	0	0	1	0
0	0	1	0	X	0	1	1	0
0	1	X	0	X	0	1	1	0
0	1	0	X	X	0	1	1	0
0	1	1	1	X	1	0	1	0
1	0	X	X	0	1	0	0	1
1	0	X	X	1	0	0	0	1

Excitation Equations

Assuming Flip-Flop's type D...

$$\boxed{D = Q^{n+1}}$$

$Q_0^n = 0$

AB \ CQ_1^n	00	01	11	10
00		1		
01		1		
11		1		
10		1		

$Q_0^n = 1$

AB \ CQ_1^n	00	01	11	10
00				
01				
11	1			1
10				

$$\boxed{D_1 = \overline{Q_0^n} \cdot \overline{C} \cdot Q_1^n + A \cdot B \cdot \overline{Q_1^n} \cdot Q_0^n}$$

$Q_0^n = 0$

AB \ CQ_1^n	00	01	11	10
00				
01				
11				
10	1			1

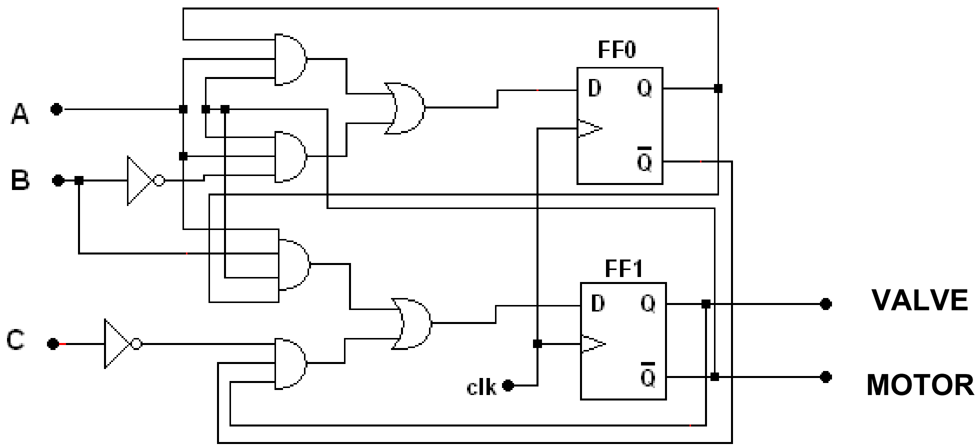
$Q_0^n = 1$

AB \ CQ_1^n	00	01	11	10
00	1			1
01	1			1
11				
10	1			1

$$\boxed{D_0 = \overline{Q_1^n} \cdot Q_0^n \cdot \overline{A} + A \cdot \overline{B} \cdot \overline{Q_1^n}}$$

$$\boxed{M = \overline{V} = \overline{Q}_1}$$

Digital Circuit

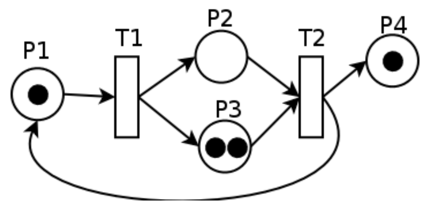


Note: The outputs depend only on the states!!!!!!

THE FINITE STATES MACHINE

Number of possible simultaneous states?

>1
PETRI nets



Used in computation theory

=1
Finite States Machine

a.k.a States machine or "Automata"

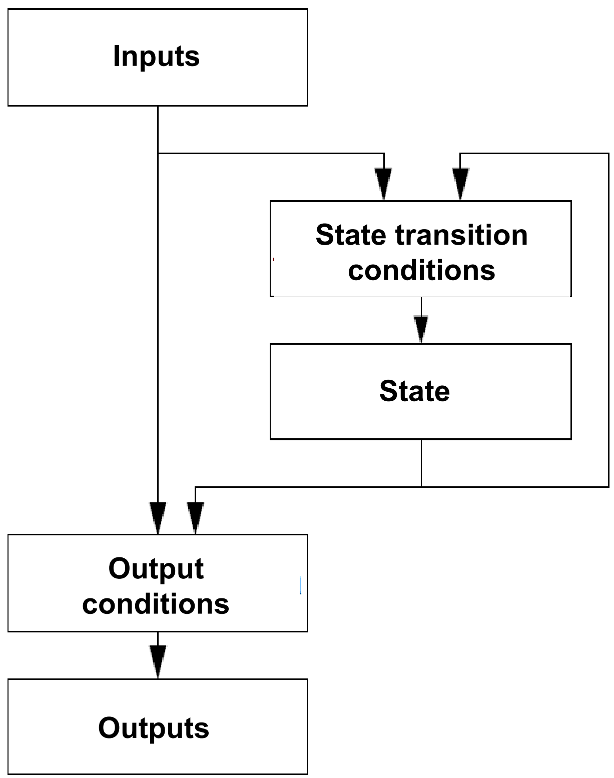
many states as distinct situations

state concept as information about the history

output=f(current states,inputs)

MEMORY

THE FINITE STATES MACHINE



States Machines representation

Transitions Matrix

States Diagram

Uses two symbols:
- circles – represents state
- arcs – represent transitions

Tipos de Acções:
- Acesso – quando entra num estado
- Saída – quando sai de um estado
- Entrada – quando uma condição de entrada é verdadeira

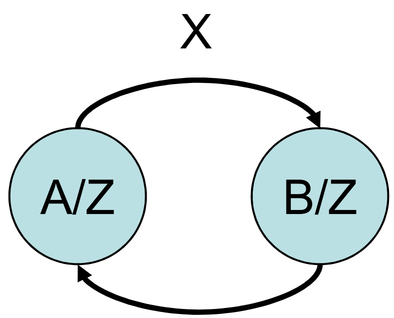
BASIC MODELS OF FINITE STATES MACHINE: MOORE AND MEALY MACHINE

Two Sequential Computation Paradigms

To the moment:
Outputs=f(States)

MOORE Machine:

Outputs only states function

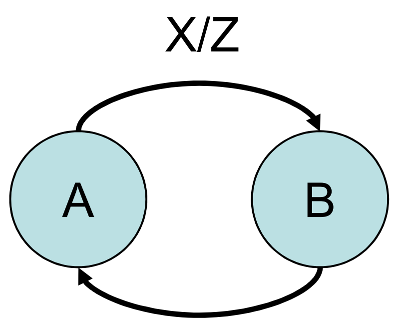


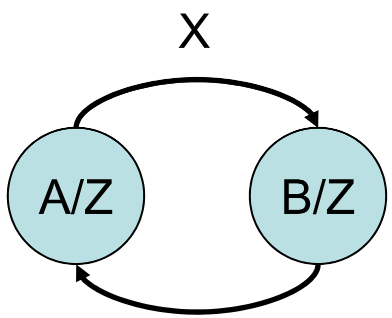
The input value in n+ only reflects in n+1 !!!

Outputs=f(States,Inputs)

MEALY Machine:

Outputs function of states and inputs





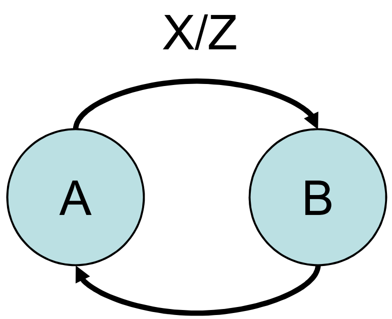
MOORE machine:

The transfer of current states to the following states depends only on X (and the clock!)

The output variable (Z) depends only on the present state

Distinct Symbolisms

Output defined in the state interior Vs.
Output defined in transition



MEALY machine:

In the arches are defined inputs and outputs: "1/0" means that the symbol "1" causes the "0“ symbol as output

The output variable (Z) depends on the present state and input

Advantages	Moore	Mealy
	- Fault detection + evident	- less states
	- Higher robustness	

EXAMPLE #1:
Door of a lift

Input variables:

- Open Door Sensor (PA)
- Closed Door Sensor (PF)
- Closing Switch (IF)
- Opening Switch (IA)

Output variables:

- Closes Door (FP)
- Opens Door (AP)

MOORE

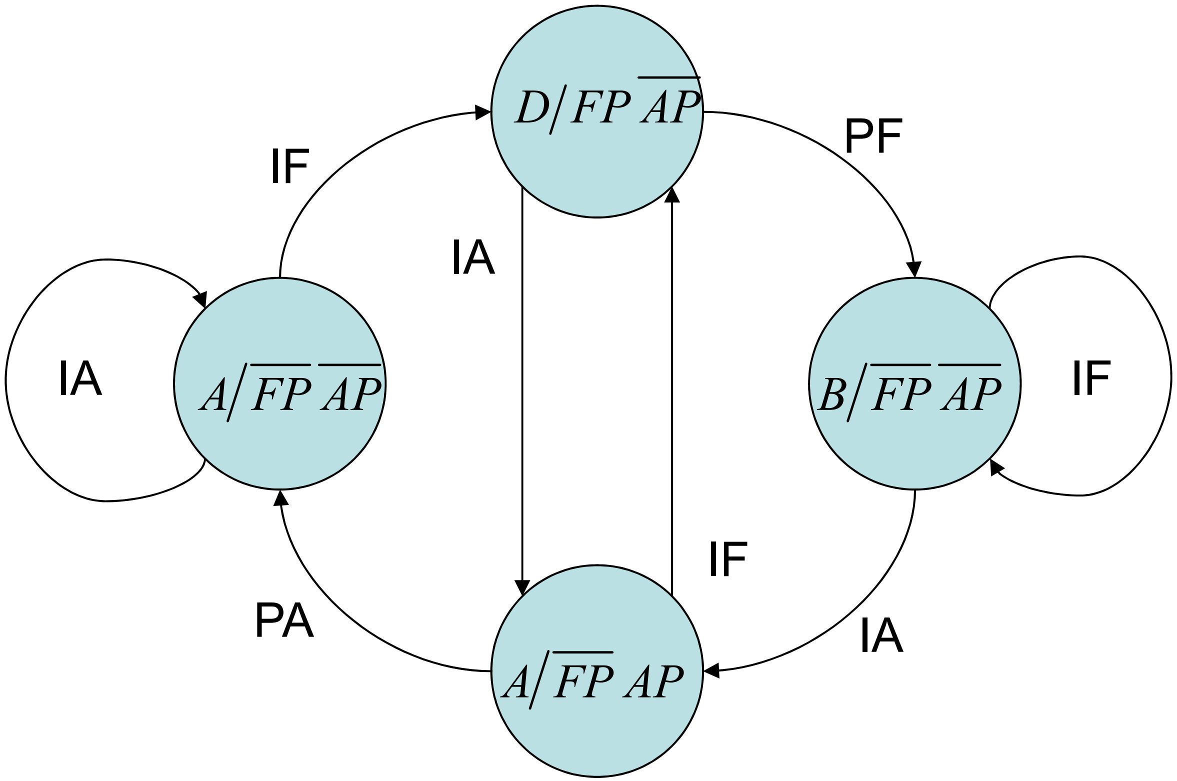
States:

A – Open Door

B – Closed Door

C – Opening Door

D – Closing Door



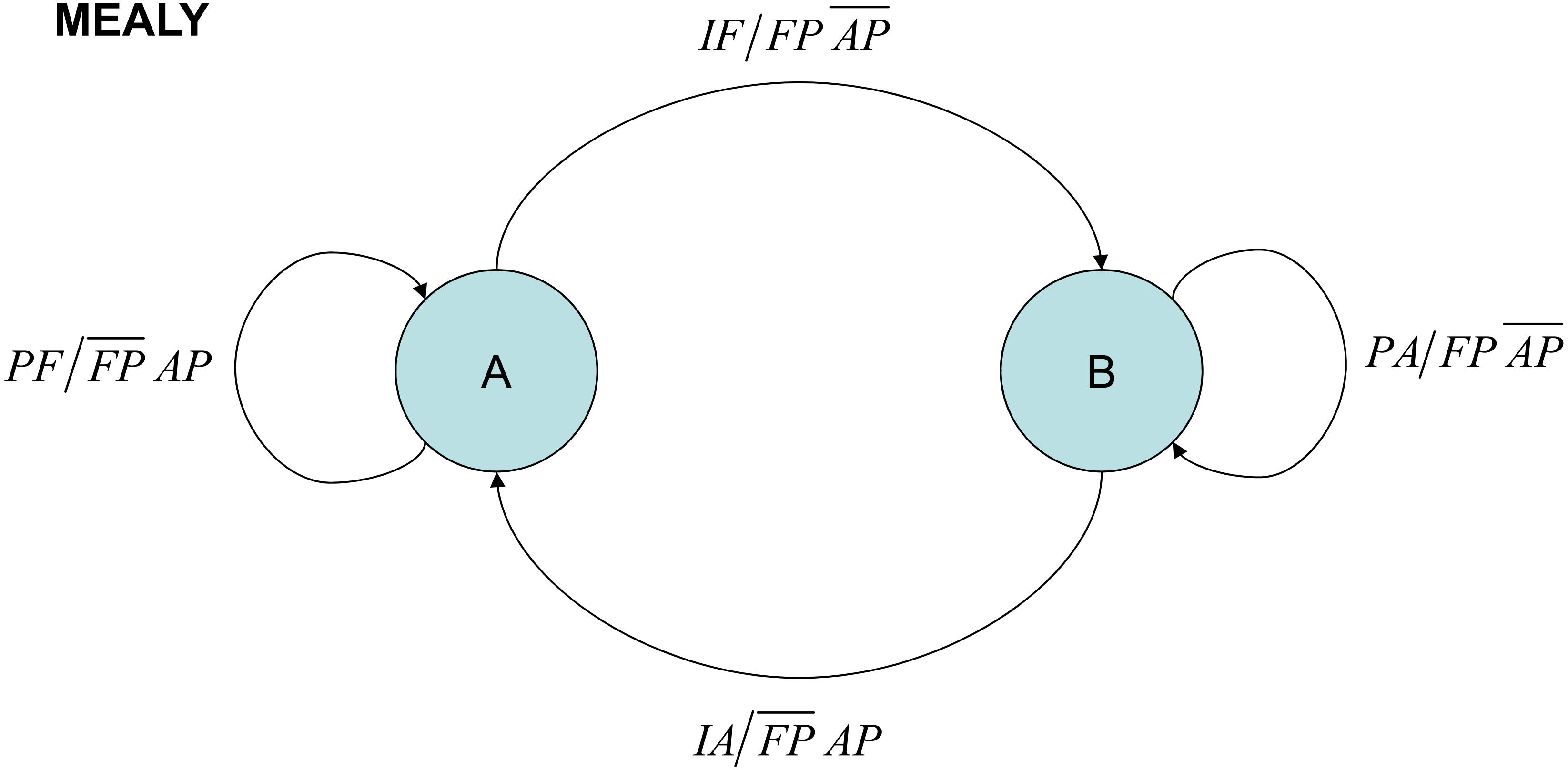
Input variables:

- Open Door Sensor (PA)
- Closed Door Sensor (PF)
- Closing Switch (IF)
- Opening Switch (IA)

Output variables:

- Closes Door (FP)
- Opens Door (AP)

MEALY



EXAMPLE #2: Sequence Detector

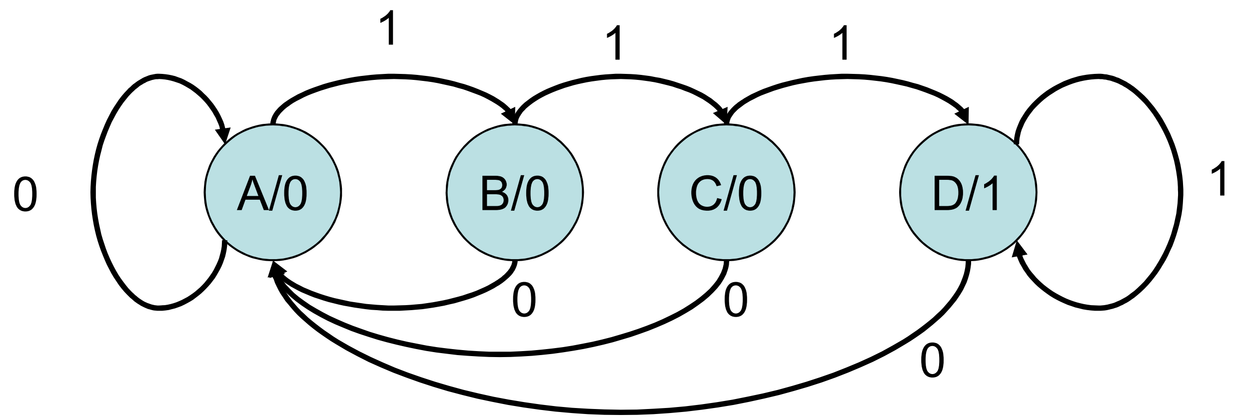
Develop a system capable of detecting that a particular sequence has been introduced. In this case, when the applied input has the sequence '111'

Variables:

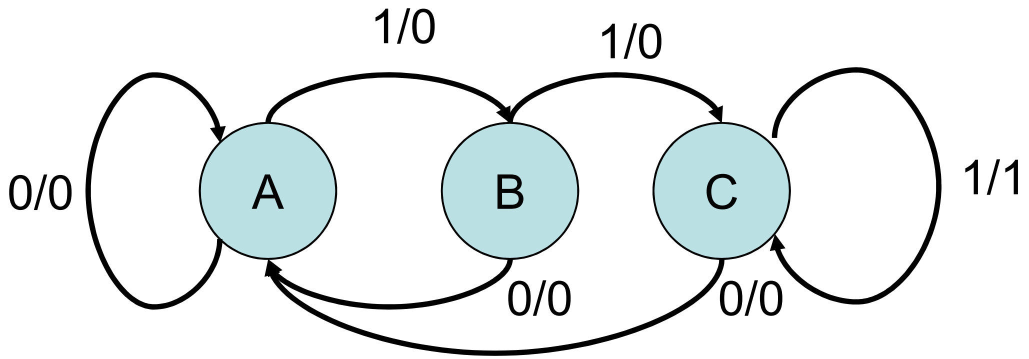
1 input (X), 1 output (Z)

If Z=1 the sequence was detected otherwise Z=0

MOORE



MEALY



Do:

Flowchart....

States Transition Table....

Logic Circuit