Engr433: Digital Design

Review of Sequential Circuits

Dr. Curtis Nelson

Sequential Logic Review

In this presentation you will learn about:

- Logic circuits that can store information;
- Basic cells, latches, and flip-flops;
- State diagrams;
- Synchronous circuit design.

Circuit Types

- ***Combinational*** – output depends only on the input.
- ***Sequential*** – output depends on input and past behavior
  - Requires use of storage elements;
  - Contents of the storage elements are called *state;*
  - Circuit goes through a sequence of states.
- ***Synchronous*** – controlled by a clock.

Clock Signals

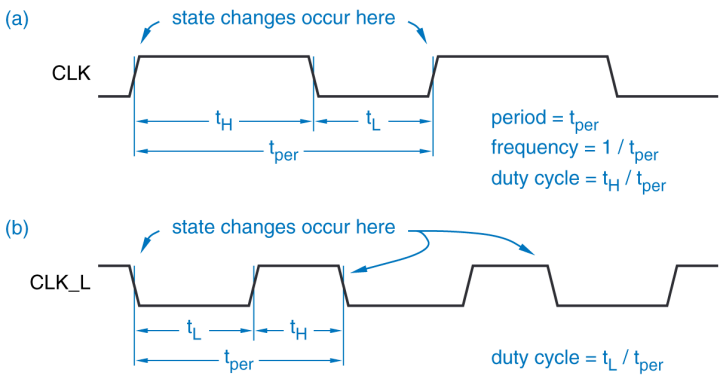
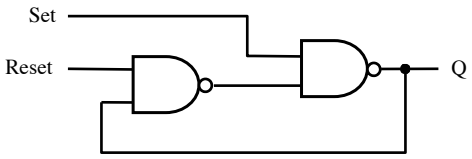


Figure 7-1

Clock signals: (a) active high; (b) active low.

NAND-Centered (Set Dominant) Basic Cell

- Inputs are active low (when they are ***asserted***);
- Operation table indicates assertion, not voltage, levels.



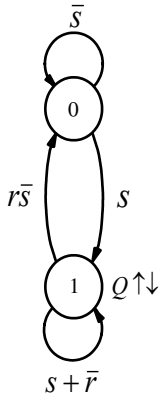
Circuit

S	R	Action	Q_{N+1}
0	0	hold	Q_N
0	1	reset	0
1	0	set	1
1	1	set	1

Operation Table

$Q_n \rightarrow Q_{n+1}$	Inputs S	R
$0 \rightarrow 0$	0	ϕ
$0 \rightarrow 1$	1	ϕ
$1 \rightarrow 0$	0	1
$1 \rightarrow 1$	1	ϕ
	ϕ	0

Excitation Table



State Diagram

Combined Form of the Basic Cell

$Q_n \rightarrow Q_{n+1}$	Inputs S	R
$0 \rightarrow 0$	0	ϕ
$0 \rightarrow 1$	1	ϕ
$1 \rightarrow 0$	0	1
$1 \rightarrow 1$	1	ϕ
	ϕ	0

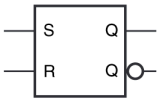
Excitation Table NAND-centered

$Q_n \rightarrow Q_{n+1}$	Inputs S	R
$0 \rightarrow 0$	0	ϕ
	ϕ	1
$0 \rightarrow 1$	1	0
$1 \rightarrow 0$	ϕ	1
$1 \rightarrow 1$	ϕ	0

Excitation Table NOR-centered

$Q_n \rightarrow Q_{n+1}$	Inputs S	R
$0 \rightarrow 0$	0	ϕ
$0 \rightarrow 1$	1	0
$1 \rightarrow 0$	0	1
$1 \rightarrow 1$	ϕ	0

Excitation Table Combined Form



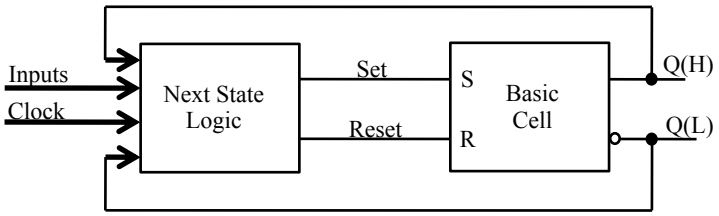
Basic Cell

Designing Latches - A Model

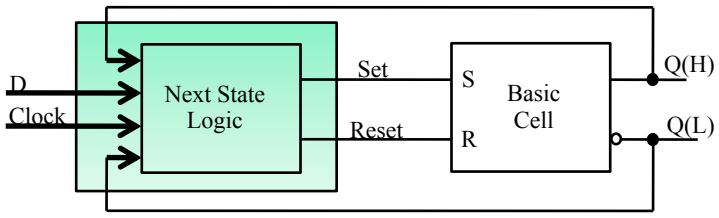
- Latch - a logic circuit that transfers the input state to the output state when the *clock* signal is high and latches and holds the input when the *clock* signal goes low.

$Q_n \rightarrow Q_{n+1}$	Inputs S	R
$0 \rightarrow 0$	0	ϕ
$0 \rightarrow 1$	1	0
$1 \rightarrow 0$	0	1
$1 \rightarrow 1$	ϕ	0

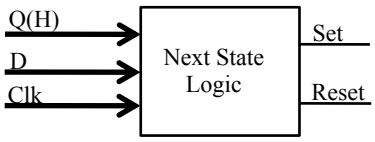
Excitation Table Combined Form



Design of a Clocked D (Data) Latch



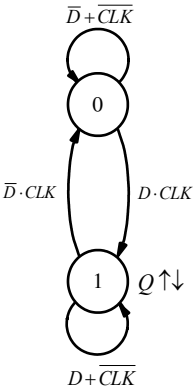
Design of a Clocked D (Data) Latch



Block Diagram

Clk	D	Qn	Qn+1	Set	Reset
0	0	0	0	0	ϕ
0	0	1	1	ϕ	0
0	1	0	0	0	ϕ
0	1	1	1	ϕ	0
1	0	0	0	0	ϕ
1	0	1	0	0	1
1	1	0	1	1	0
1	1	1	1	ϕ	0

Truth Table



State Diagram

Clk	D	Action	Q_{N+1}
0	x	hold	Q_N
1	0	reset	0
1	1	set	1

Operation Table

$Q_n \rightarrow Q_{n+1}$	Inputs S	R
$0 \rightarrow 0$	0	ϕ
$0 \rightarrow 1$	1	0
$1 \rightarrow 0$	0	1
$1 \rightarrow 1$	ϕ	0

Excitation Table
Basic Cell

$$Set = clk \cdot D$$
$$Reset = clk \cdot \overline{D}$$

Equations

Clocked D Latch

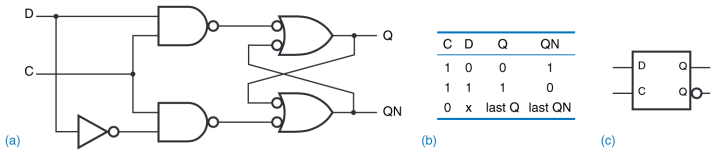


Figure 7-12
D latch: (a) circuit design using NAND gates; (b) function table; (c) logic symbol.

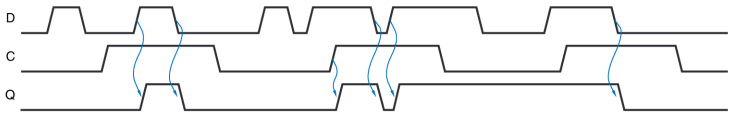
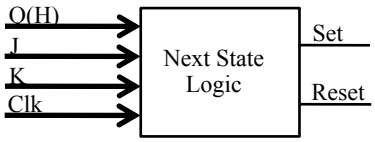


Figure 7-13
Functional behavior of a D latch for various inputs.

Design of a Clocked JK Latch



Block Diagram

Clk	J	K	Action	Q_{N+1}
0	x	x	Hold	Q_N
1	0	0	Hold	Q_N
1	0	1	Reset	0
1	1	0	Set	1
1	1	1	Toggle	$\overline{Q}_N$

Function Table

$Q_n \rightarrow Q_{n+1}$	Inputs S	R
$0 \rightarrow 0$	0	ϕ
$0 \rightarrow 1$	1	0
$1 \rightarrow 0$	0	1
$1 \rightarrow 1$	ϕ	0

Excitation Table
Basic Cell

Clk	J	K	Qn	Qn+1	Set	Reset
0	0	0	0	0	0	ϕ
0	0	0	1	1	ϕ	0
0	0	1	0	0	0	ϕ
0	0	1	1	1	ϕ	0
0	1	0	0	0	0	ϕ
0	1	0	1	1	ϕ	0
0	1	1	0	0	0	ϕ
0	1	1	1	1	ϕ	0
1	0	0	0	0	0	ϕ
1	0	0	1	1	ϕ	0
1	0	1	0	0	0	ϕ
1	0	1	1	0	0	1
1	1	0	0	1	1	0
1	1	0	1	1	ϕ	0
1	1	1	0	1	1	0
1	1	1	1	0	0	1

Truth Table

$$Set = clk \cdot J \cdot \overline{Q}$$

$$Reset = clk \cdot K \cdot Q$$

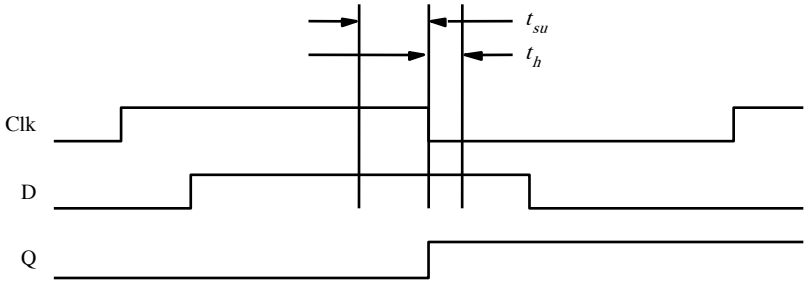
Equations

Terminology

- Latches are often called ***transparent*** because the output will follow the input as long as the clock signal is high.
- Flip-flops are ***edge-triggered***
  - Positive-edge triggered (PET) is when action occurs on the rising edge of the clock signal;
  - Negative-edge triggered (NET) is when action occurs on the falling edge of the clock signal.
- Types of Flip-flops
  - D (very, very common, 74HC74);
  - Toggle (occasionally used by CAD programs);
  - JK (hardly ever used, 74HC109);
  - SR (rarely used).

Setup and Hold Times

- Setup time (t_{SU}) is the time interval preceding the active transition point of the CLK during which all data inputs must remain stable.
- Hold time (t_H) is the time interval following the active transition point of the CLK during which all data inputs must remain stable.
- See data sheet for 74HC74



Positive-Edge-Triggered D Flip-Flop

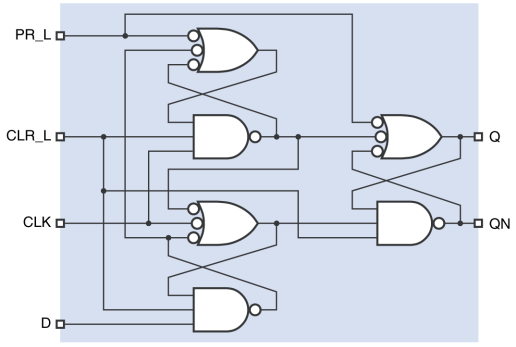


Figure 7-20
Commercial circuit for a positive-edge-triggered D flip-flop such as 74LS74.

PET D Flip-Flop with *Clear* and *Preset*

- Synchronous – transitions or actions occur in relation to the CLK signal;
- Asynchronous – transitions or actions are not dependent on the CLK signal;
- 74HC74 for example:

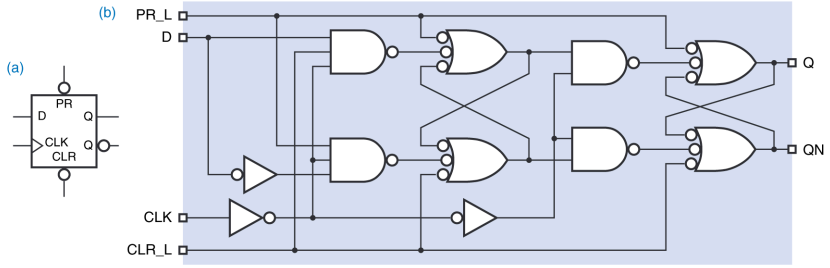
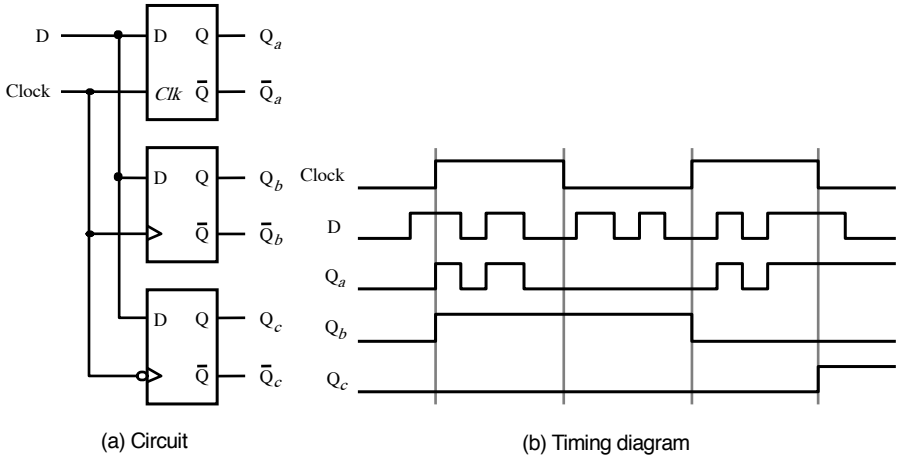


Figure 7-19

Positive-edge-triggered D flip-flop with preset and clear:
(a) logic symbol; (b) circuit design using NAND gates.

Level-Sensitive vs. Edge-Triggered

- Level-sensitive = latch
- Edge-triggered = flip-flop



Summary of Terminology

- Basic cell – cross-coupled NAND/NOR.
- Gated latch – output changes only when *Clk* is asserted:
  - Gated SR latch;
  - Gated D latch;
  - Gated JK latch.
- Flip-flop – output changes only on *Clk* edge:
  - Edge-triggered;
  - Three main types:
    - D (very, very common, 74HC74);
    - Toggle (occasionally used by CAD programs);
    - JK (hardly ever used, 74HC109).

Sequential Machines

- Logic circuits that transition through a sequence of states are called *synchronous sequential machines*, or more simply, *state machines.*
- A *logic state* is a unique set of binary values that characterize the logic status of a sequential machine at some point in time.

A Sequence of Logic Events

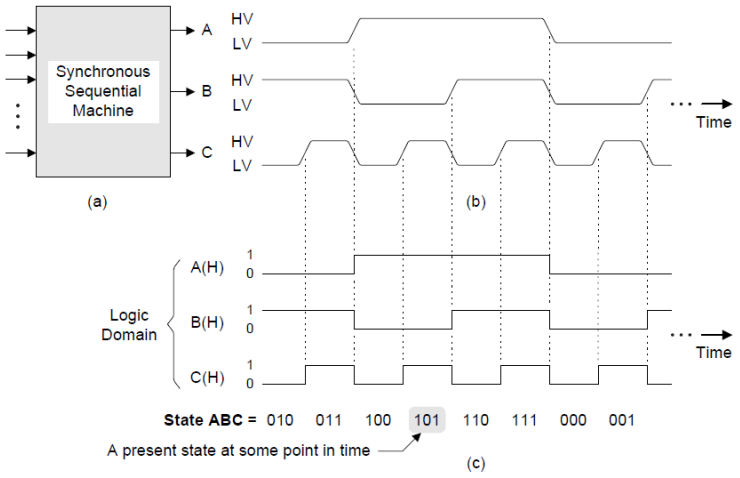
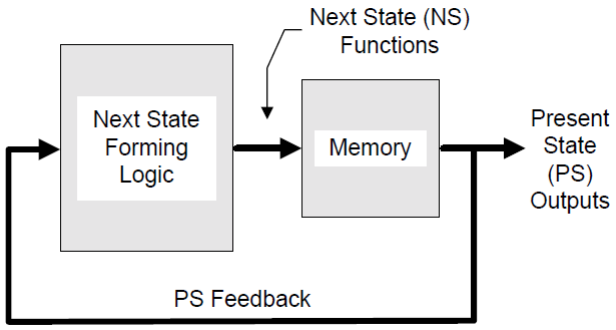


Figure 10.1 A sequence of logic events from a synchronous state machine. (a) Block diagram symbol and (b) output voltage waveforms. (c) Timing diagram representing the positive logic interpretation of the voltage waveforms and showing a sequence of logic states.

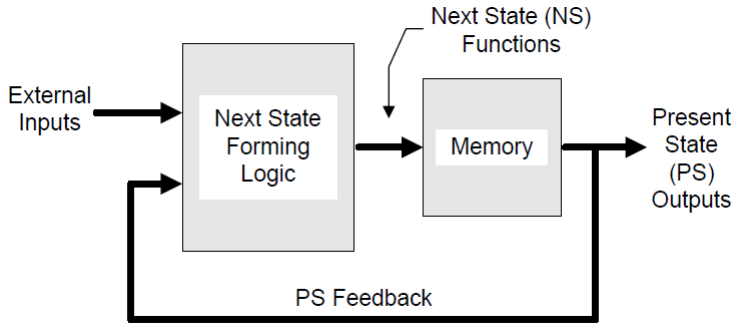
Basic Models for Sequential Machines

- A model with no external inputs – the output sequence is controlled by the memory and clock only
  - A counter is an example.



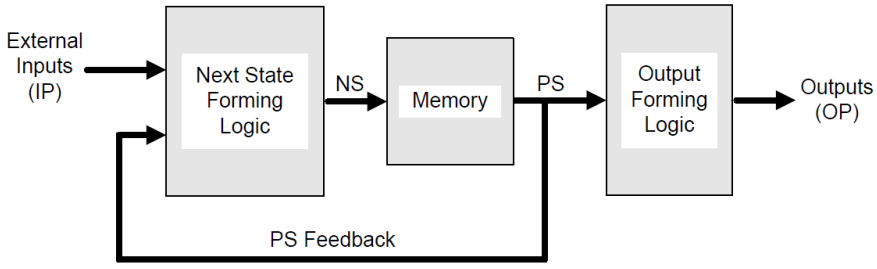
Basic Models for Sequential Machines

- Next, we add *External Input* capability
  - An up/down counter is an example.



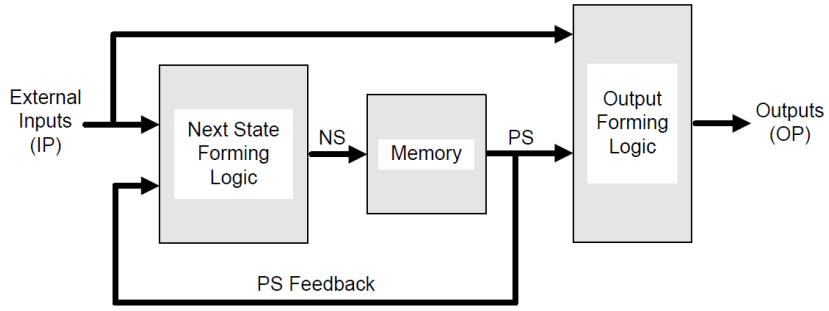
Moore's Model of a Sequential Machine

- Next, we add an *Output Forming Logic block*
  - This is known as Moore's model.



Mealy's Model of a Sequential Machine

- Next, we route inputs to the output forming logic block
  - This is known as Mealy's model.



Fully Documented State Diagram

- Present states;
- Next states;
- Previous states.

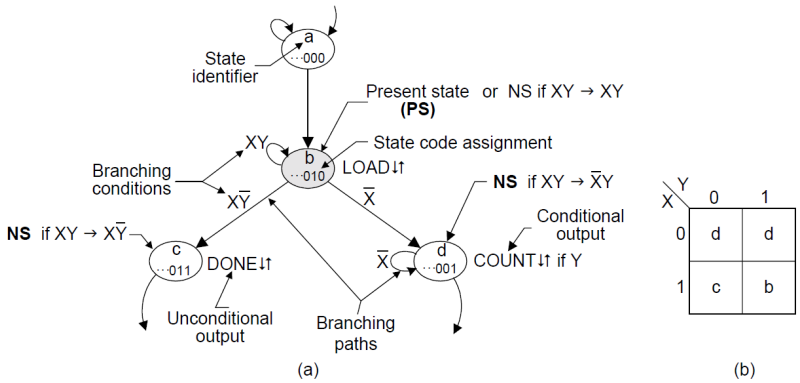


Figure 10.6 (a) Features of the fully documented state diagram section. (b) The input/state map for state b.

Fully Documented State Diagram

- State identifier;
- State code assignment;
- Branching conditions (sum of = 1).

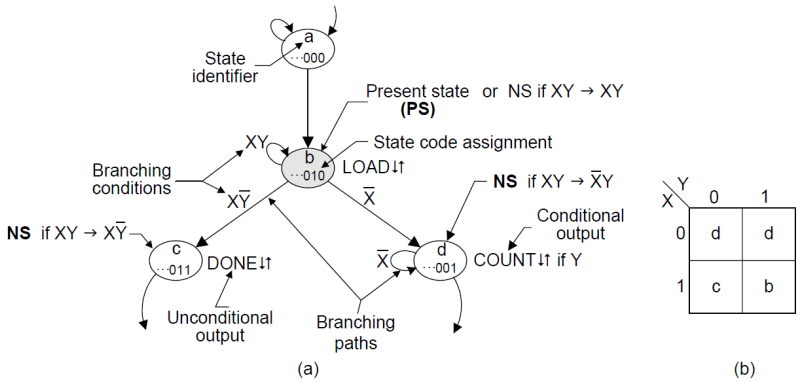


Figure 10.6 (a) Features of the fully documented state diagram section. (b) The input/state map for state b.

Fully Documented State Diagram

- Unconditional outputs;
- Conditional outputs.

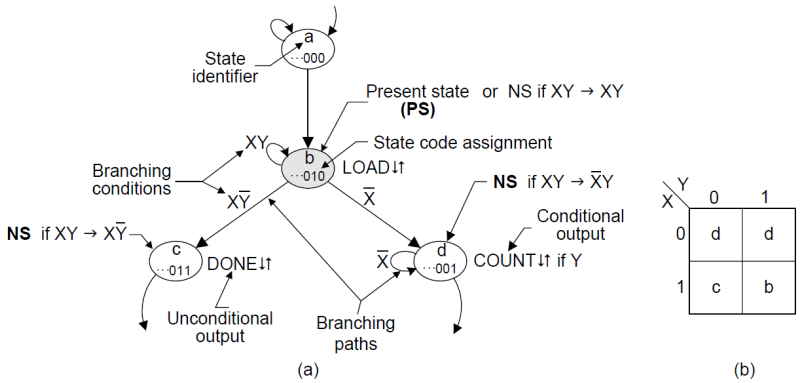


Figure 10.6 (a) Features of the fully documented state diagram section. (b) The input/state map for state b.

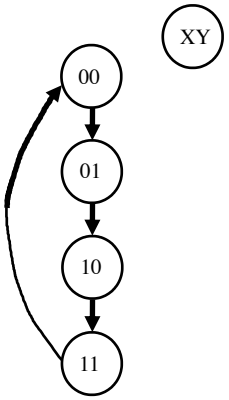
State Machine Design Process

1. Get design specifications;
2. Create a block diagram specifying all inputs and outputs;
3. Design a state diagram using as few states as possible;
4. Make binary state assignments, maximizing logical adjacencies in the K-maps;
5. Plot entered variable K-maps;
6. Read minimum next state decoder logic;
7. Develop output decoder logic by plotting the output K-maps;
8. Draw schematic;
9. Do a logic simulation before wiring your circuit.

Design of a 2-bit Binary Up-Counter

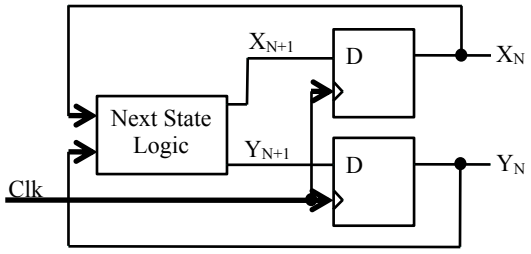
- Step 1 – Definition (given in title).
- Step 2 – Block diagram
  - No inputs (other than Clk);
  - 2-bit output (X, Y).

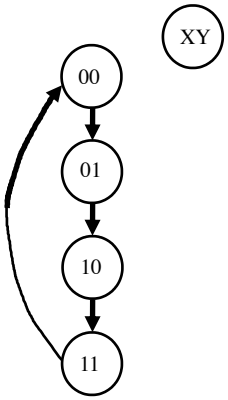
- Step 3 – State diagram.



State Diagram

Design of a 2-bit Binary Up-Counter





State Diagram

- Step 4 – Binary state assignments
  - Same as output counting code in this example.
- Step 5 – Truth table.

X_N	Y_N	X_{N+1}	Y_{N+1}
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

$Q_n \rightarrow Q_{n+1}$	D
$0 \rightarrow 0$	0
$0 \rightarrow 1$	1
$1 \rightarrow 0$	0
$1 \rightarrow 1$	1

Excitation Table

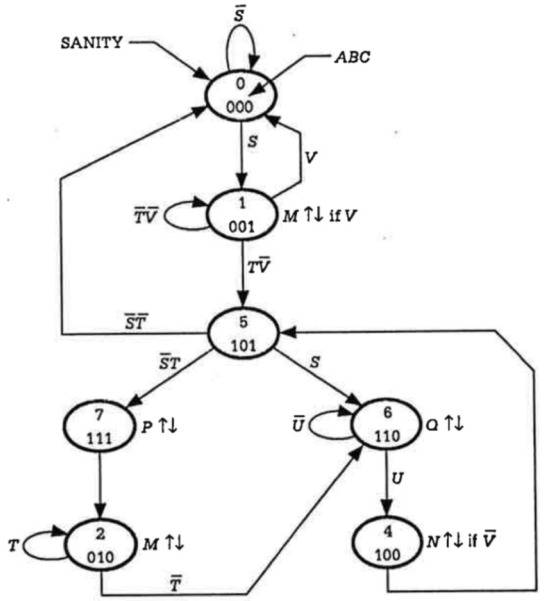
$$X_{N+1} = X_N \oplus Y_N$$
$$Y_{N+1} = \overline{Y_N}$$

Equations

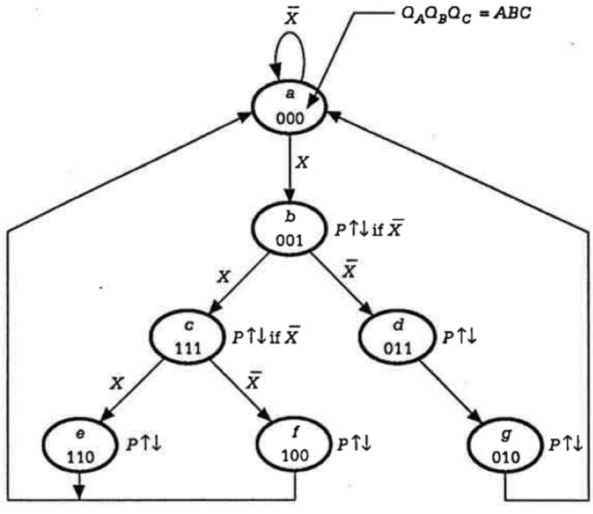
Other State Machine Design Examples

- Design of a 2-bit grey-code up counter;
- Design of a 2-bit grey-code up-down counter;
- Use an output decoder
  - Design a $00 \rightarrow 11 \rightarrow 10 \rightarrow 11$ and repeat counter.
- Alternative state machine architectures:
  - State decoders;
  - State decoders and fully addressed multiplexers for NS logic;
  - State decoders and reduced order mux's for NS logic.

Alternative SM Example #1



Alternative SM Example #2



Other Synchronous System Issues

- Asynchronous inputs
  - Input synchronizing
- Catching short inputs
- Clock skew
- Output glitch analysis
- State code assignments
  - Into rule
  - Out of rule
- Static and dynamic hazards
- Switch debouncing