Engr354: Digital Logic Circuits

Chapters 8
Sequential Logic Design Principles

Dr. Curtis Nelson

Sequential Logic Design Principles

In this chapter you will learn about:

- Design techniques for circuits that use flip-flops;
- State diagrams;
- Clocked synchronous state-machines;
- Clocked synchronous state-machine design.

Circuit Type Review

- ***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 as a result of changes in inputs.
- ***Synchronous*** – controlled by a clock.

Summary of Terminology

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

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

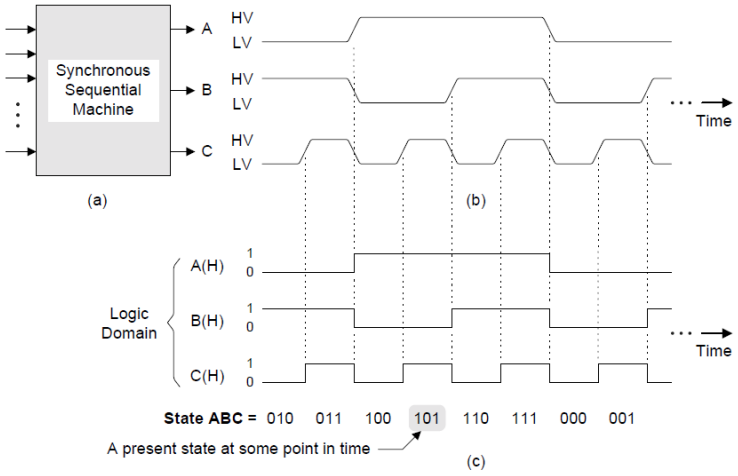
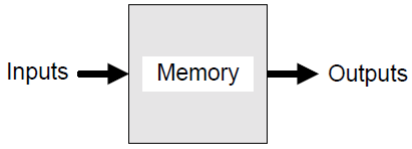


Figure 10.1 A sequence of logic events from a synchronous state machine. (a) Block diagram symbol and (b) output voltage waveforms. (c) Timing diagram representing the positive logic interpretation of the voltage waveforms and showing a sequence of logic states.

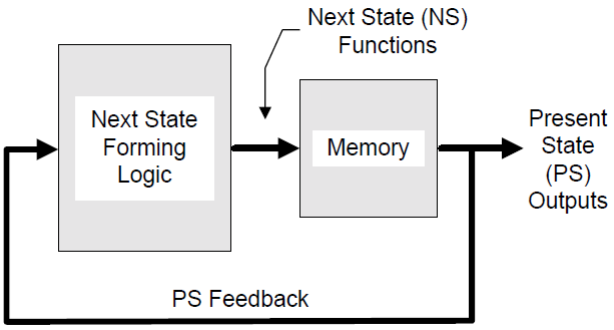
Basic Models for Sequential Machines

- The *most basic* model contains memory, where the inputs act as a "look-up" address;
- Example – LUT's, PLA's, ROM's, RAM's, etc.



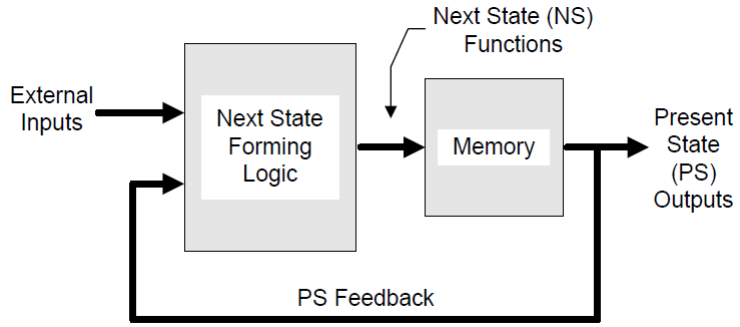
Basic Models for Sequential Machines

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



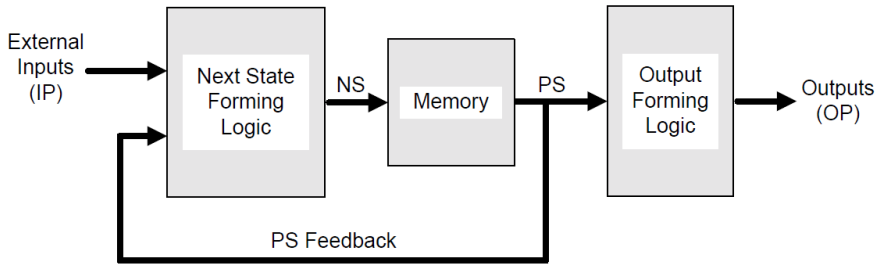
Basic Models for Sequential Machines

- Next, we add *External Input* capability.



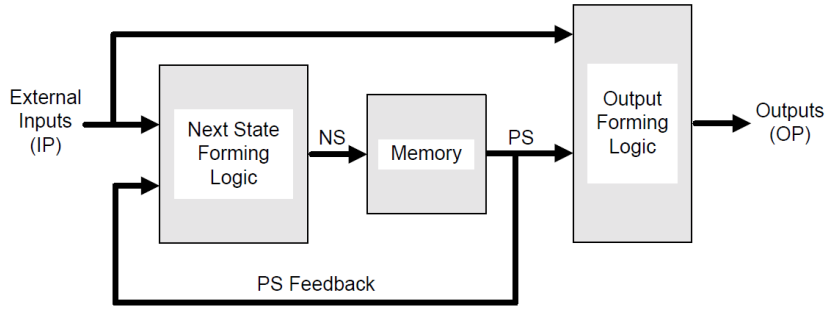
Moore's Model of a Sequential Machine

- Next, we add an *Output Forming Logic block.*



Mealy's Model of a Sequential Machine

- Next, we route inputs to the output forming logic block.



Fully Documented State Diagram

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

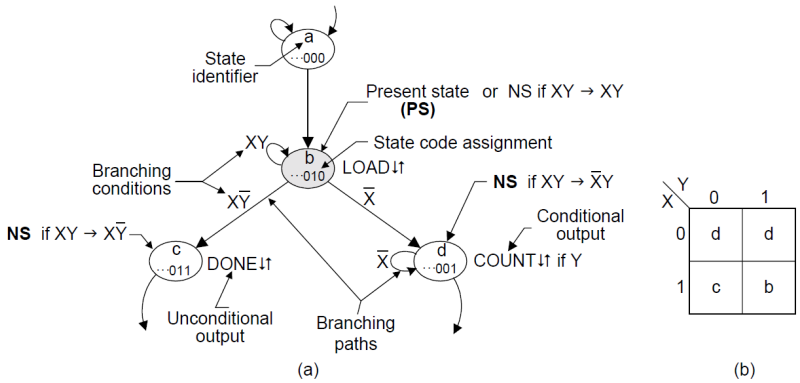


Figure 10.6 (a) Features of the fully documented state diagram section. (b) The input/state map for state b.

Fully Documented State Diagram

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

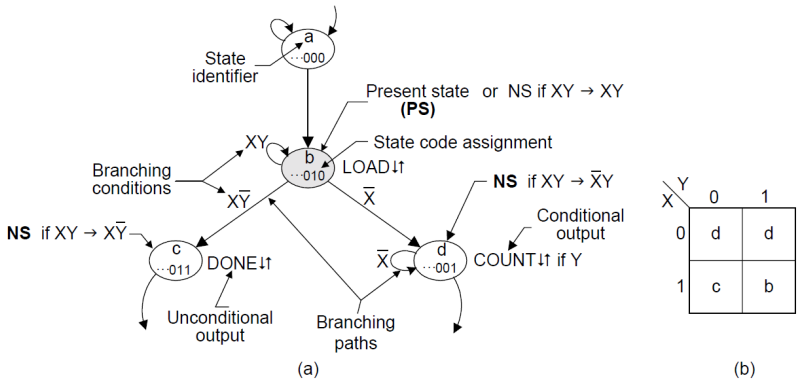


Figure 10.6 (a) Features of the fully documented state diagram section. (b) The input/state map for state b.

Fully Documented State Diagram

- Unconditional outputs;
- Conditional outputs.

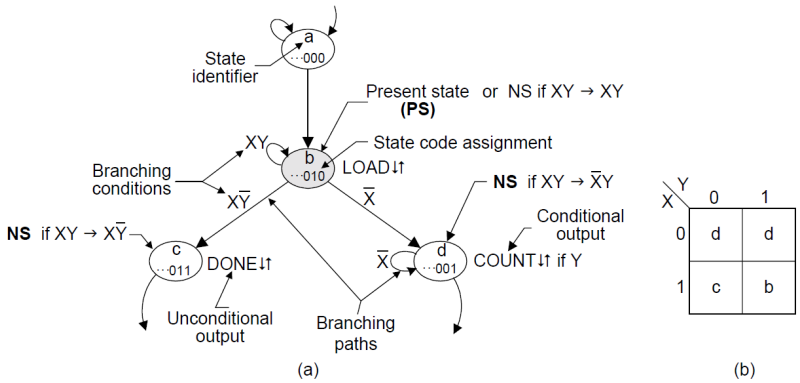


Figure 10.6 (a) Features of the fully documented state diagram section. (b) The input/state map for state b.

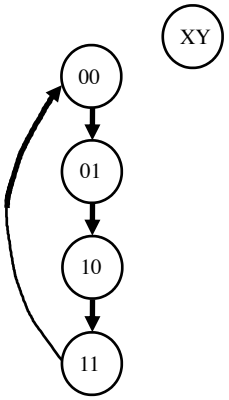
State Machine Design Process

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

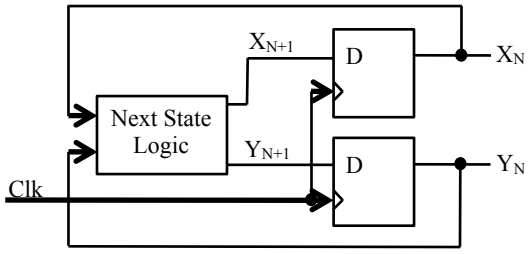
- Step 1 – Definition (given in title).
- Step 2 – Block diagram:
  - No inputs (other than Clk);
  - 2-bit output (X, Y).

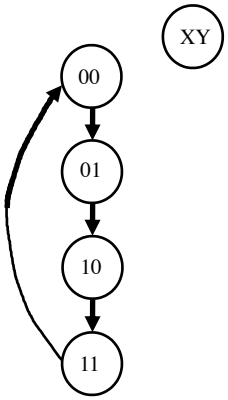
- Step 3 – State diagram.



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 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.

Sequential Logic Design Principles Summary

In this chapter you learned about:

- Design techniques for circuits that use flip-flops;
- State diagrams;
- Clocked synchronous state-machines;
- Clocked synchronous state-machine design.