Digital Logic Design
ENEE 244-010x

Lecture 21

Announcements

- No homework over break
  - Homework 9 will be posted on Monday, 11/30.

Agenda

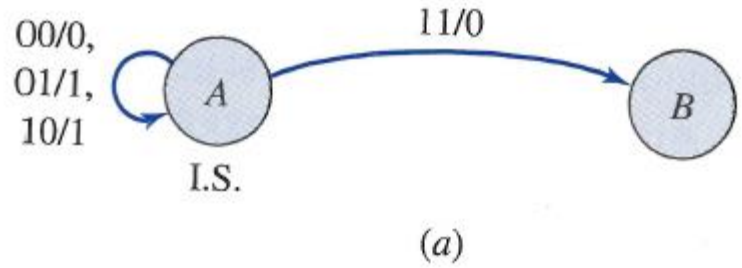
- Last time:
  - Structure and Operation of Clocked Synchronous Sequential Networks (7.1)
  - Analysis of Clocked Synchronous Sequential Networks (7.2)
- This time:
  - Modeling Clocked Synchronous Sequential Network Behavior (7.3)
  - State Table Reduction (7.4)

Modeling clocked synchronous sequential network behavior

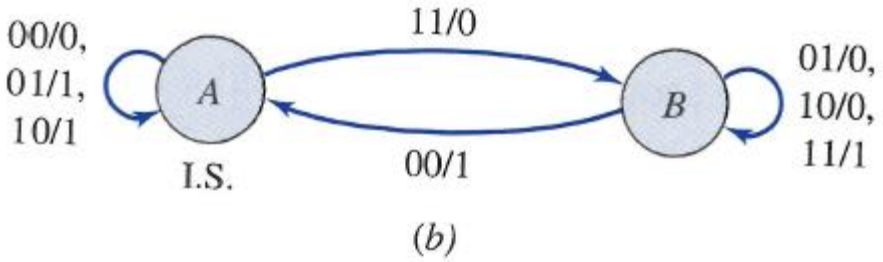
- Approach for the synthesis of clocked synchronous sequential networks:
  - State table/state diagram is constructed from word specifications.
  - State reduction technique to obtain a state table with minimum number of states.
  - Transition table is formed by coding the states of the state table.
  - Excitation table is constructed based on the flip-flop types to be used.
  - From the excitation table, the excitation and output expressions for the network are determined.
  - Finally, the logic diagram is drawn.

Examples of Modeling Step

State Diagram for Mealy serial binary adder



(a)



(b)

Figure 7.12 Obtaining the state diagram for a Mealy serial binary adder. (a) Partial state diagram. (b) Completed state diagram.

State Diagram for Moore serial binary adder

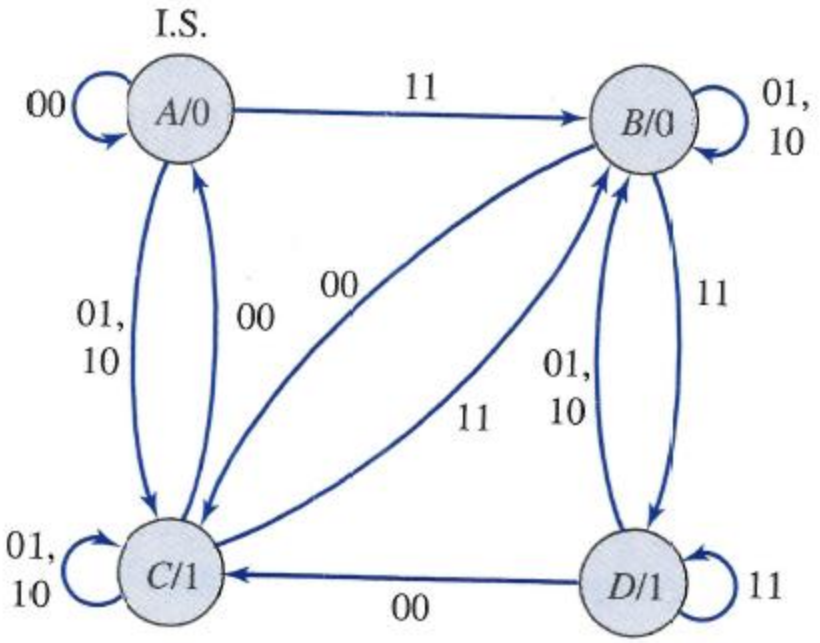
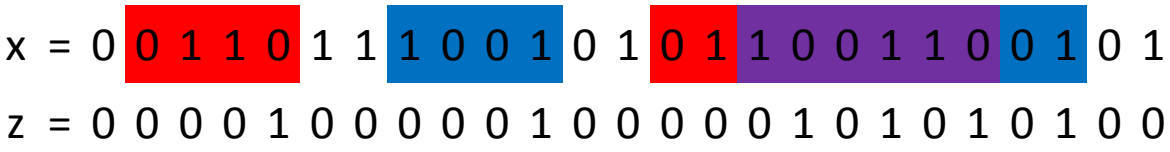


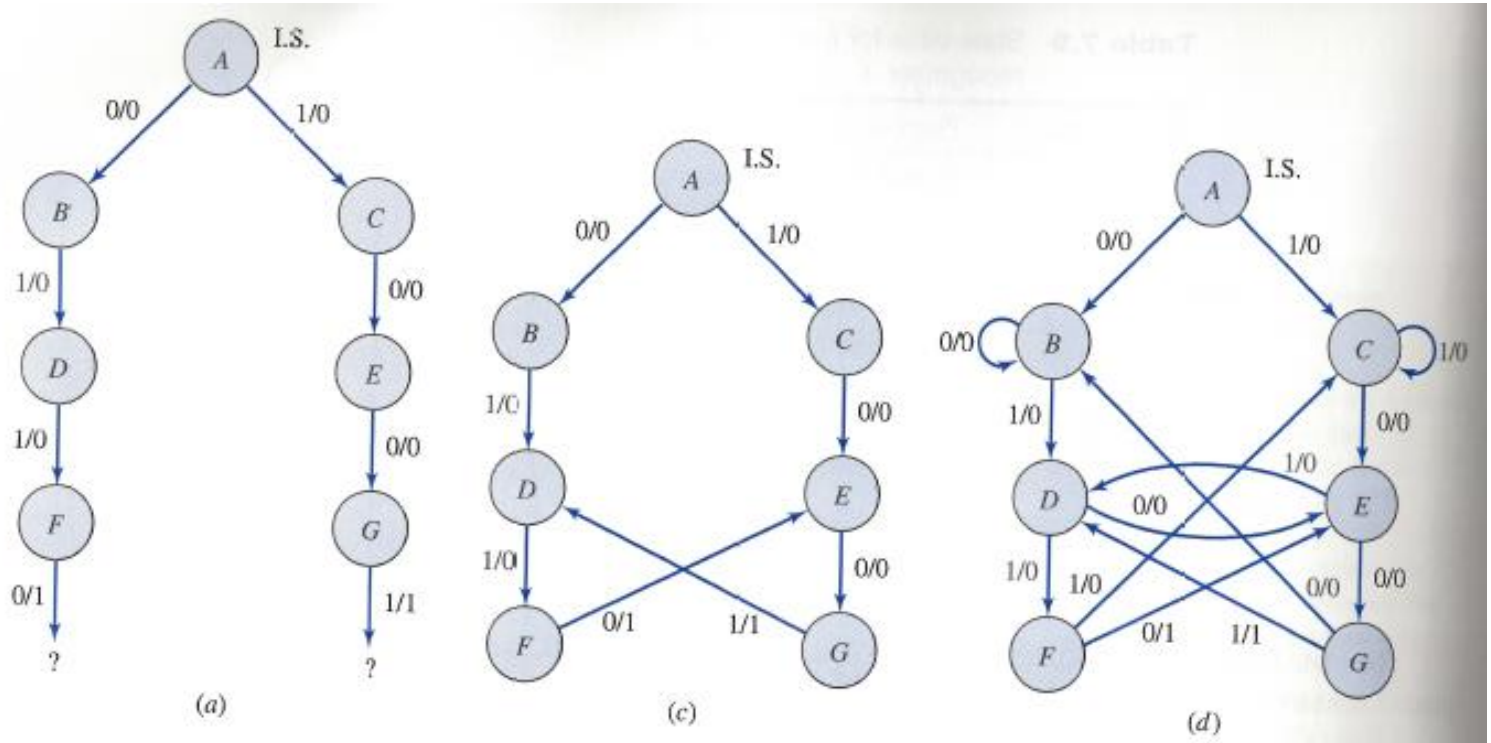
Figure 7.13 State diagram for a Moore serial binary adder.

A Sequence Recognizer

- Network produces a 1 output iff the current input and the previous three inputs correspond to either 0110 or 1001
- 0110/1001 Sequence Recognizer
- The 1 output is to occur at the time of the fourth input of the recognized sequence. Outputs of 0 are to be produced at all other times.
- A Mealy network model is developed since the output is a function of the current input x.
- Network is not required to reset upon the occurrence of the fourth input.
- Sequences may overlap.
- Example:



A Sequence Recognizer



A: No inputs received (initial state).
B: Last input received was 0 but not 10 nor 100.
C: Last input received was 1 but not 01 nor 011.
D: Last two inputs received were 01.
E: Last two inputs received were 10.
F: Last three inputs received were 011.
G: Last three inputs received were 100.

(b)

Figure 7.16 A 0110/1001 sequence recognizer. (*a*) Beginning the detection of the sequences 0110 or 1001. (*b*) Definition of states. (*c*) Completing the detection of the two sequences 0110 or 1001. (*d*) Completed state diagram.

Another Sequence Recognizer

- An output of 1 is to be produced iff the three input symbols following two consecutive input 0's include at least one 1.
- At all other times the output is to be 0.
- The output of 1 is to be coincident with the third input symbol of the three-input-symbol sequence.
- Upon completing the analysis of the three input symbols following the pair of 0 inputs, the network is to reset itself and await for another pair of 0's and then at least one 1 in the following seqence of three input symbols.
- Since the output is to be coincident with the third input symbol, a Mealy network is implied.
- Example:

x = 0 1 0 0 0 1 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 1 1

z = 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1

A Sequence Recognizer

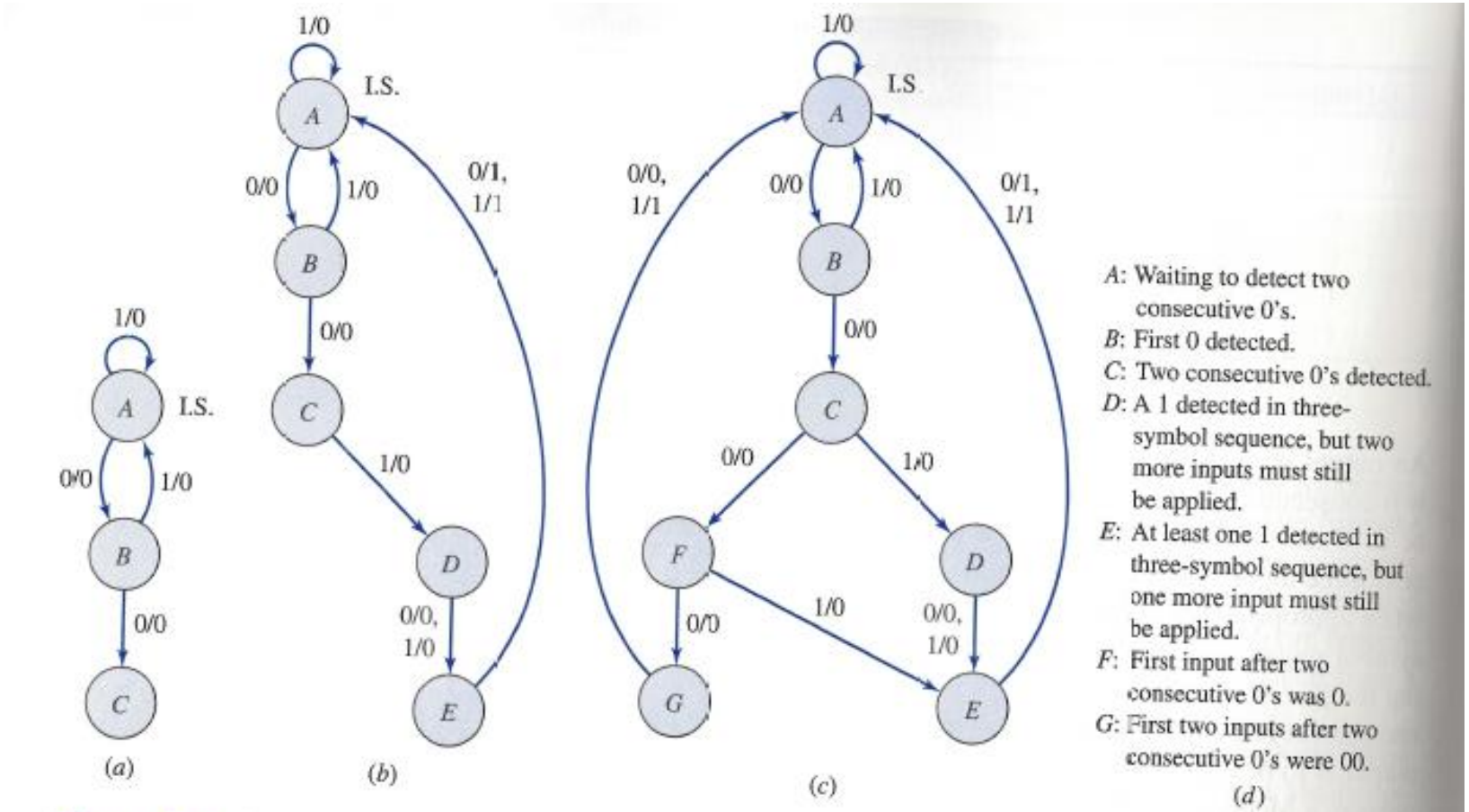


Figure 7.15 State diagram for a sequence recognizer. (*a*) Detection of two consecutive 0's. (*b*) Partial analysis of the three-symbol sequence. (*c*) Completed state diagram. (*d*) Definition of states.

Final Example

- Single input x and a single output z.
- The output of the network is initially 0.
- Changes on the next input immediately following each even occurrence of $x = 1$.
- The effect of the input is to be delayed by one clock period. The value of x during the next clock period does not affect the output of the network at that time.
- A Moore sequential network is to be realized
- Example:

x = 0 0 1 1 0 0 0 1 0 1 0 1 1 1 0

a b c

z = 0 0 0 0 1 1 1 1 1 1 0 0 0 1 1

Final Example

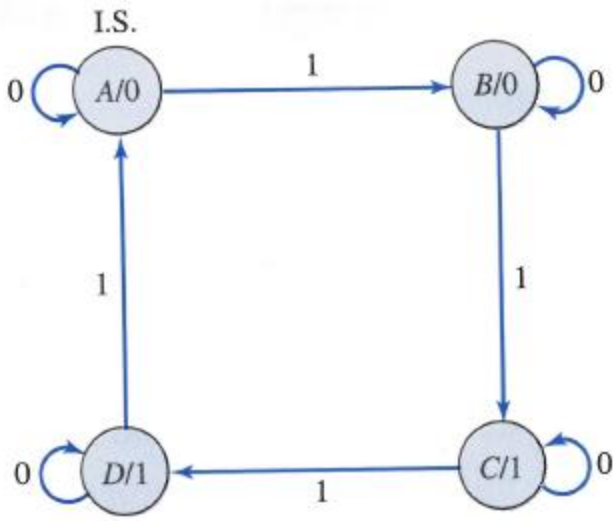


Figure 7.17 State diagram for the final example.

Table 7.12 State table for Fig. 7.17

Present state	Next state		Output (z)
	Input (x)		
	0	1	
*A	A	B	0
B	B	C	0
C	C	D	1
D	D	A	1

A: The output changes to 0 since the last occurrence of $x = 1$ was even
B: The output remains at 0 since the last occurrence of $x = 1$ was odd
C: The output changes to 1 since the last occurrence of $x = 1$ was even
D: The output remains at 1 since the last occurrence of $x = 1$ was odd

State Table Reduction

State Table Reduction

- The state table is a description of the terminal behavior of a clocked synchronous sequential network.
- During the process of creating the state table, more states may be defined then are really necessary.
- In analysis of synchronous sequential networks, bin codes for the states were replaced by arbitrary symbols when going from transition table to state table.
- For synthesis, the opposite process is performed: Arbitrary state symbols are replaced with binary codes.
- Smaller number of binary digits necessary -> smaller number of flip-flops.
- Reducing number of states may also simplify the combinational logic.

State Table Reduction Procedure

- 3-step process:
  - Equivalent pairs of states are determined
  - Sets of equivalent states are established
  - Reduced state table is constructed with one state for each of the sets of equivalent states

Determining Equivalent Pairs of States

- Recall that given a state table, an input sequence and some starting state, the resulting output sequence can be determined.
- Assume we run two experiments:
  - One starts in state p
  - The second starts in state q

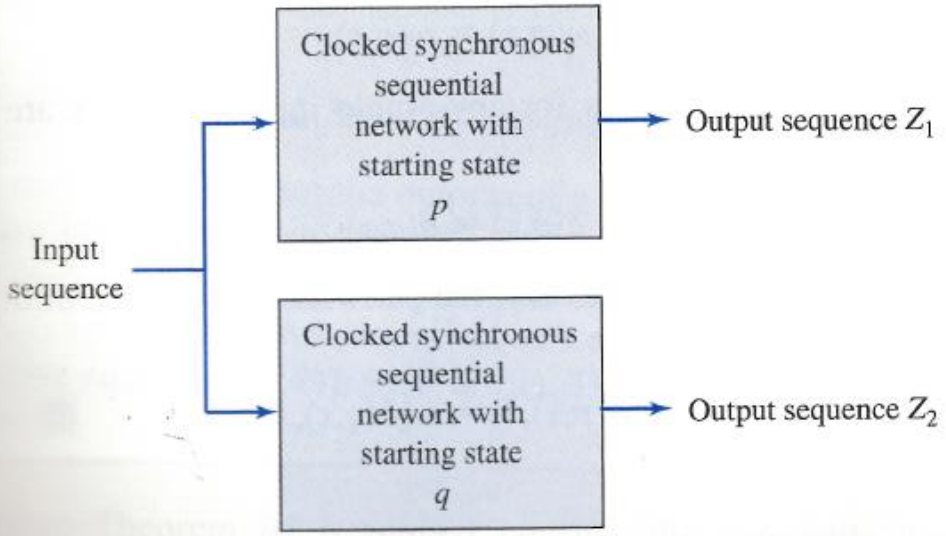


Figure 7.18 Experiment for determining equivalent pairs of states.

Determining Equivalent Pairs of States

- States p and q are said to be equivalent or indistinguishable $(p \equiv q)$ iff for all input sequences applied to the two starting states, identical output sequences from both networks result $(Z_1 \equiv Z_2)$.
- If states p and q are not equivalent, then they are said to be distinguishable $(p \neq q)$.

Determining Equivalent Pairs of States

Theorem:

Two states p and q of a clocked synchronous sequential network are equivalent iff for each combination of values of the input variables

1. Their outputs are identical
2. Their next states are equivalent