Digital Logic Design
ENEE 244-010x

Lecture 24

Announcements

- Homework 9 due today
- Thursday Office Hours (12/10) from 2:30-4pm
- Course Evaluations at the end of class today.
  - https://www.courseevalum.umd.edu/
  - Log in with directory id and password
- Final exam info:
  - Wednesday, Dec. 16 1:30-3:30pm in EGR 1108 (our regular classroom).
  - Review session next class and during Thursday's recitation
  - Information about final exam and review problems for the review sessions will be up on course webpage by tonight.

Agenda

- Last Time:
  - The State Assignment Problem (7.5)
  - Completing the Design of Clocked Synchronous Sequential Networks (7.6)
- This Time:
  - Synchronous Sequential network design using PLDs (7.6)
  - Algorithmic State Machines (8.1-8.2)

Realizations of Synchronous Sequential Networks using PLDs

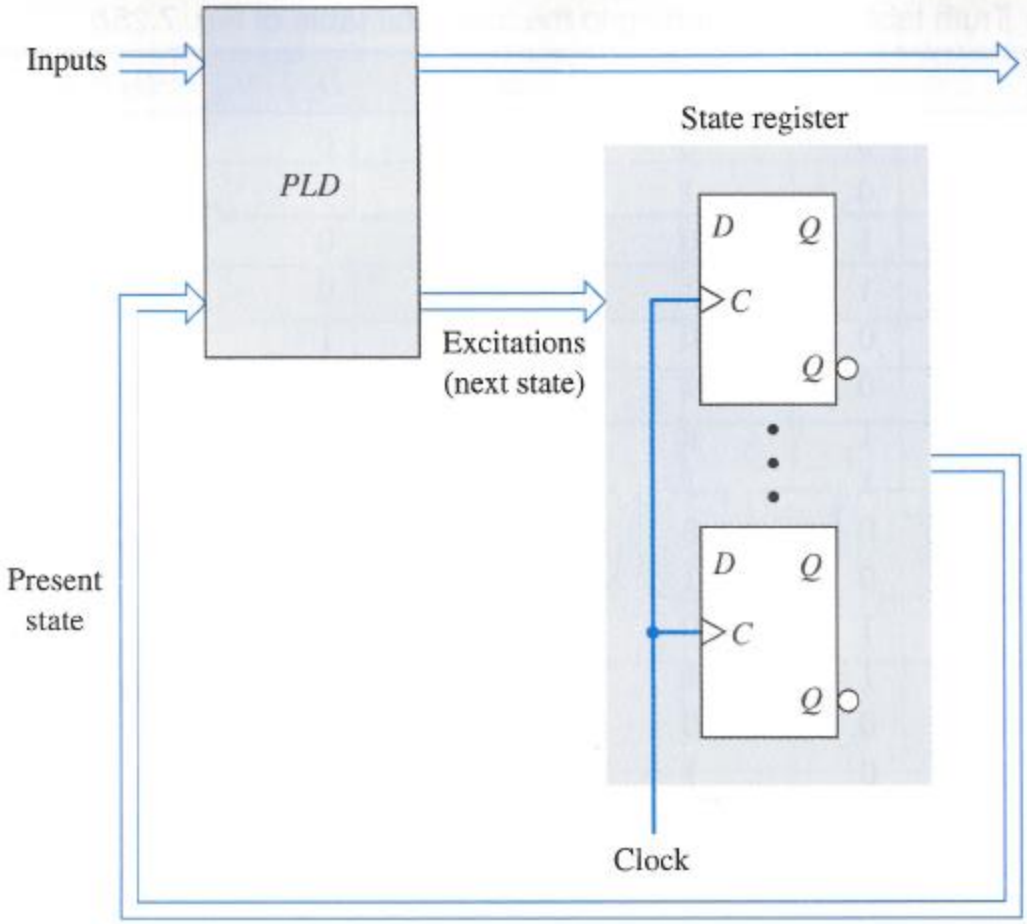
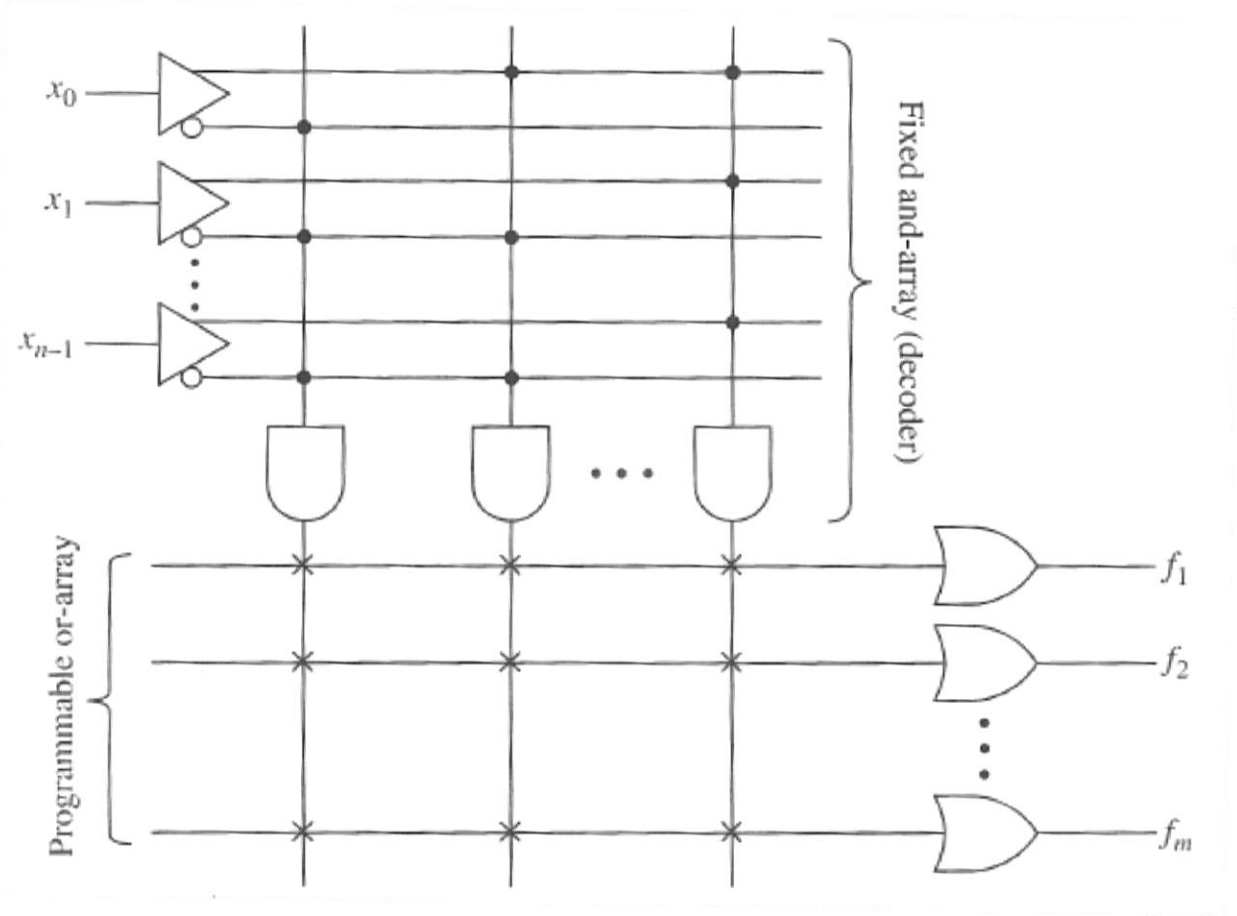


Figure 7.31 General structure of a clocked sequential network realization using a PLD and clocked *D* flip-flops.

PROM Realization



PROM Realization

Table 7.19 Excitation table involving *D* flip-flops corresponding to the transition table of Fig. 7.25*b*

Present state $(Q_1Q_2Q_3)$	Excitation $(D_1D_2D_3)$		Output (z)	
	Input (x)		Input (x)	
	0	1	0	1
000	000	001	0	0
001	010	011	1	0
010	000	011	0	1
011	100	000	1	1
100	010	001	0	0

Table 7.22 Truth table corresponding to the transition table of Fig. 7.25*b*

x	Q_1	Q_2	Q_3	D_1	D_2	D_3	z
0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	1
0	0	1	0	0	0	0	0
0	0	1	1	1	0	0	1
0	1	0	0	0	1	0	0
0	1	0	1	–	–	–	–
0	1	1	0	–	–	–	–
0	1	1	1	–	–	–	–
1	0	0	0	0	0	1	0
1	0	0	1	0	1	1	0
1	0	1	0	0	1	1	1
1	0	1	1	0	0	0	1
1	1	0	0	0	0	1	0
1	1	0	1	–	–	–	–
1	1	1	0	–	–	–	–
1	1	1	1	–	–	–	–

PLA Realization

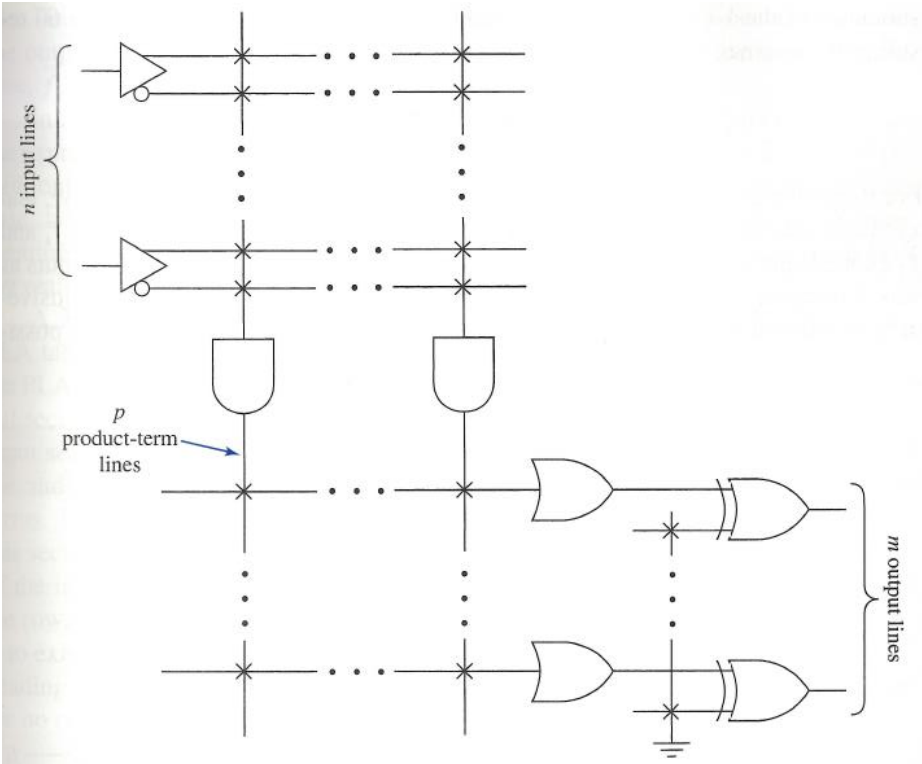


Figure 5.59 General structure of a PLA having true and complemented output capability.

Product term	Inputs x	y	z	Outputs f_1	f_2
$x\bar{y}$	1	0	–	1	1
$x\bar{z}$	1	–	0	1	–
$\bar{y}\bar{z}$	–	0	0	1	–
yz	–	1	1	–	1
			T/C	C	C

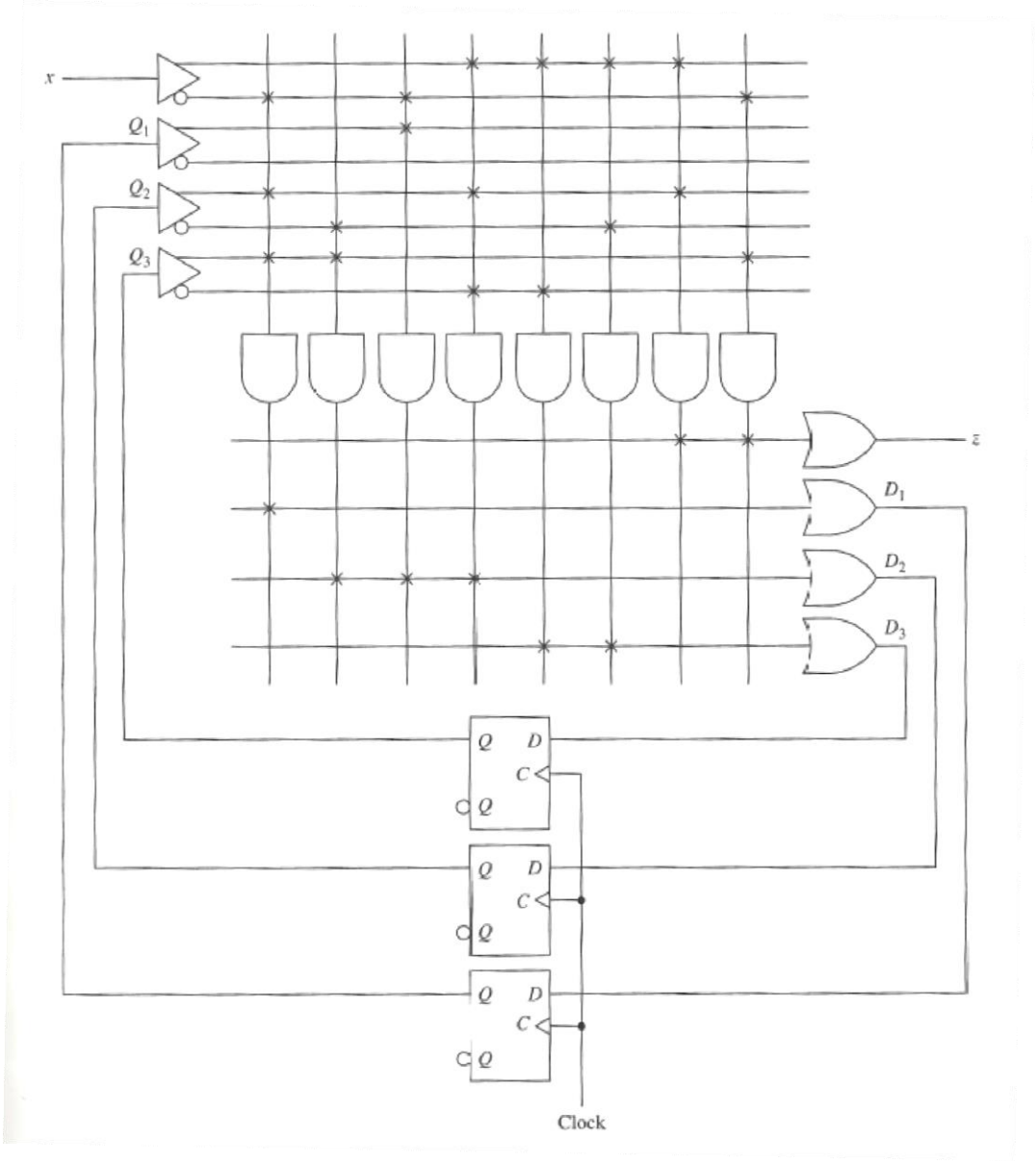
PLA Realization

$$D_1 = Q_2 Q_3 \overline{x}$$
$$D_2 = \overline{Q}_2 Q_3 + Q_1 \overline{x} + Q_2 \overline{Q}_3 x$$
$$D_3 = \overline{Q}_3 x + \overline{Q}_2 x$$
$$z = Q_2 x + Q_3 \overline{x}$$

Table 7.23 PLA table for Eqs. (7.24) to (7.27)

Product term	Inputs				Outputs			
	Q_1	Q_2	Q_3	x	D_1	D_2	D_3	z
$Q_2 Q_3 \overline{x}$	–	1	1	0	1	–	–	–
$\overline{Q}_2 Q_3$	–	0	1	–	–	1	–	–
$Q_1 \overline{x}$	1	–	–	0	–	1	–	–
$Q_2 \overline{Q}_3 x$	–	1	0	1	–	1	–	–
$\overline{Q}_3 x$	–	–	0	1	–	–	1	–
$\overline{Q}_2 x$	–	0	–	1	–	–	1	–
$Q_2 x$	–	1	–	1	–	–	–	1
$Q_3 \overline{x}$	–	–	1	0	–	–	–	1

PLA Realization



Final Topic: Algorithmic State Machines

- Previous chapter dealt with the classical approach to clocked synchronous sequential network design
  - Used models of Mealy and Moore
  - Also called Finite State Machines
- Algorithmic State Machines (ASM)
  - Different approach to clocked synchronous network design
  - Approach is higher-level
    - Uses flow-charts as in high-level programming
    - We explicitly name and use variables!
  - Capable of handling more complex systems

Algorithmic State Machine

- Partitions the system into two entities:
  - Controller
  - Controlled architecture (data processor)
- Data processor includes:
  - Flip-flops, shift registers, counters, adders/subtractors, comparators, etc.
- Controller supplies a time sequence of commands to the devices of the data processer
  - E.g. shift left, shift right, add, subtract, increment, reset.
  - Data processor supplies information about the status of its various devices.
- The controller is a hardware algorithm (thus is called an algorithmic state machine (ASM).

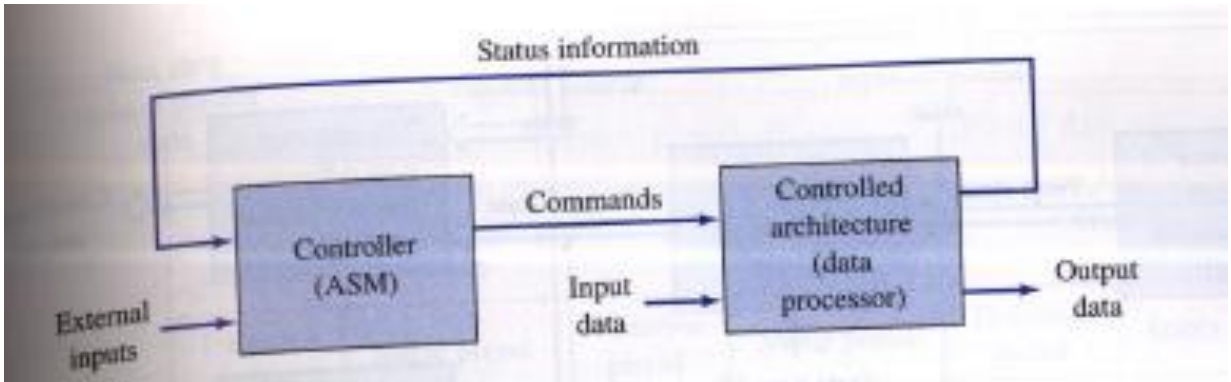


Figure 8.1 Partitioning of a digital system.

Model of an ASM

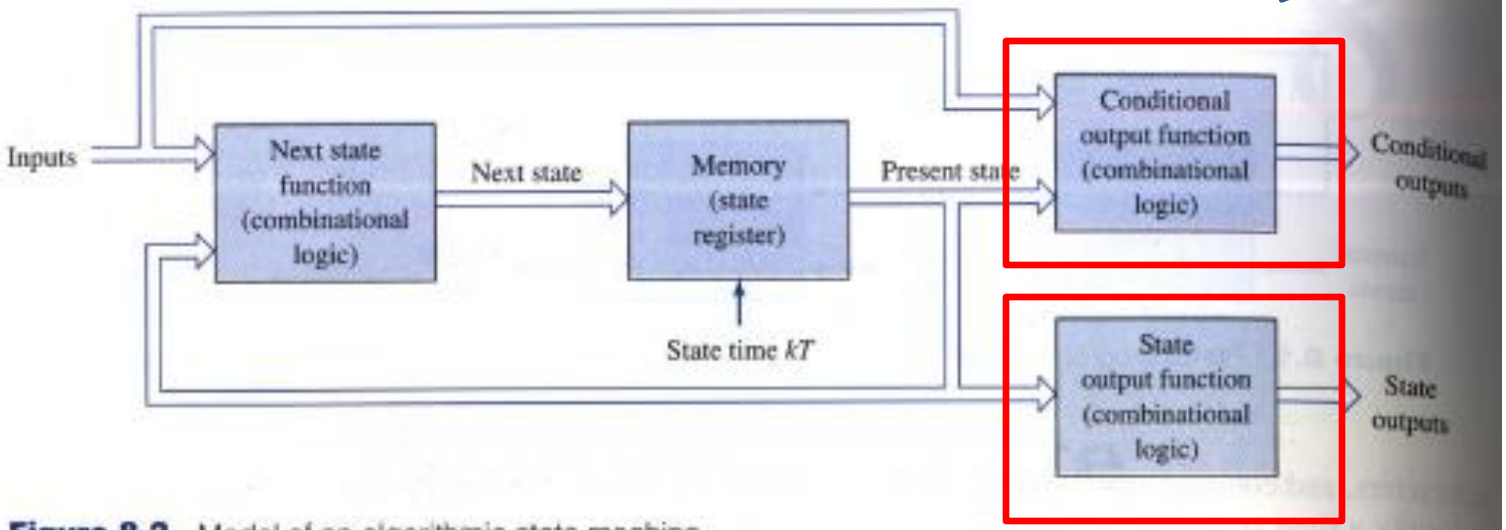


Figure 8.2 Model of an algorithmic state machine.

Mealy

Moore

ASM Charts Components

- State Box:

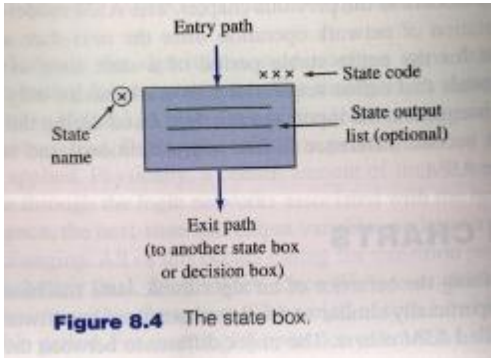
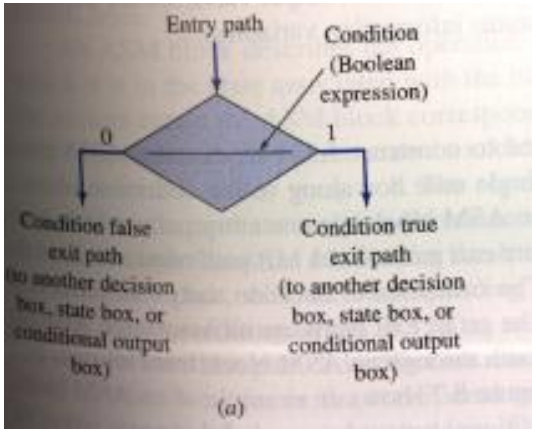


Figure 8.4 The state box.

The state output list contains output variables that are only a function of the state. Such variables that have logic-1 value are placed in this box.

- Decision Box



ASM Charts Components

- Conditional Output Box:

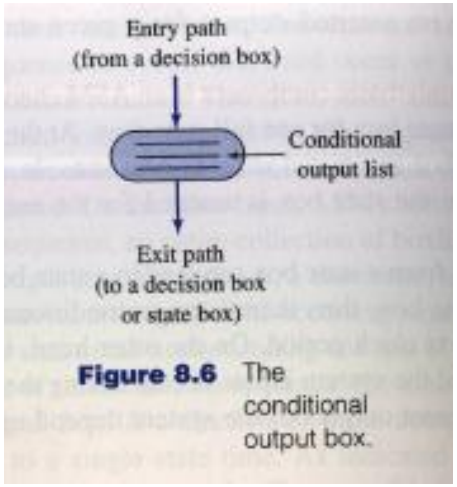


Figure 8.6 The conditional output box.

The conditional output list contains output variables that are a function of the state and external inputs. Such variables that have logic-1 value are placed in this box.

ASM Blocks

- Consists of the interconnection of a single state box along with a collection of decision and conditional output boxes.
  - One entry path, one or more exit paths leading to another state box.
  - A path through an ASM block from its state box to an exit path is called a link path.

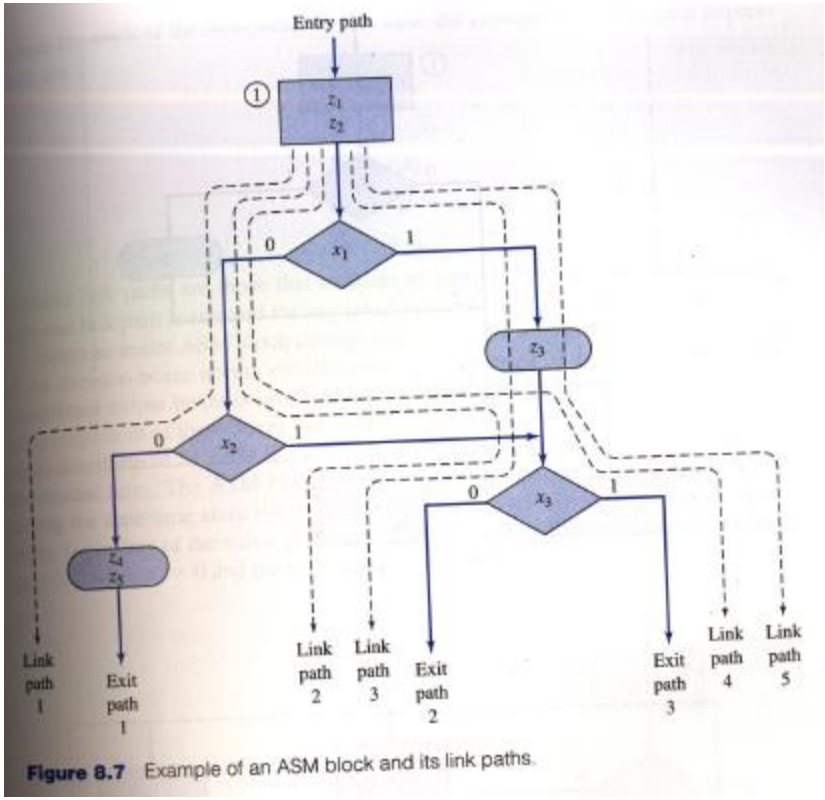


Figure 8.7 Example of an ASM block and its link paths.

Rules for ASM blocks

- For any valid combination of values to the decision-box variables, all simultaneously selected link paths must lead to the same exit path.
  - i.e. next state is uniquely determined
- There can be no closed loops that do not contain at least one state box
  - State box is the only component that is time dependent.

Simple ASM Charts

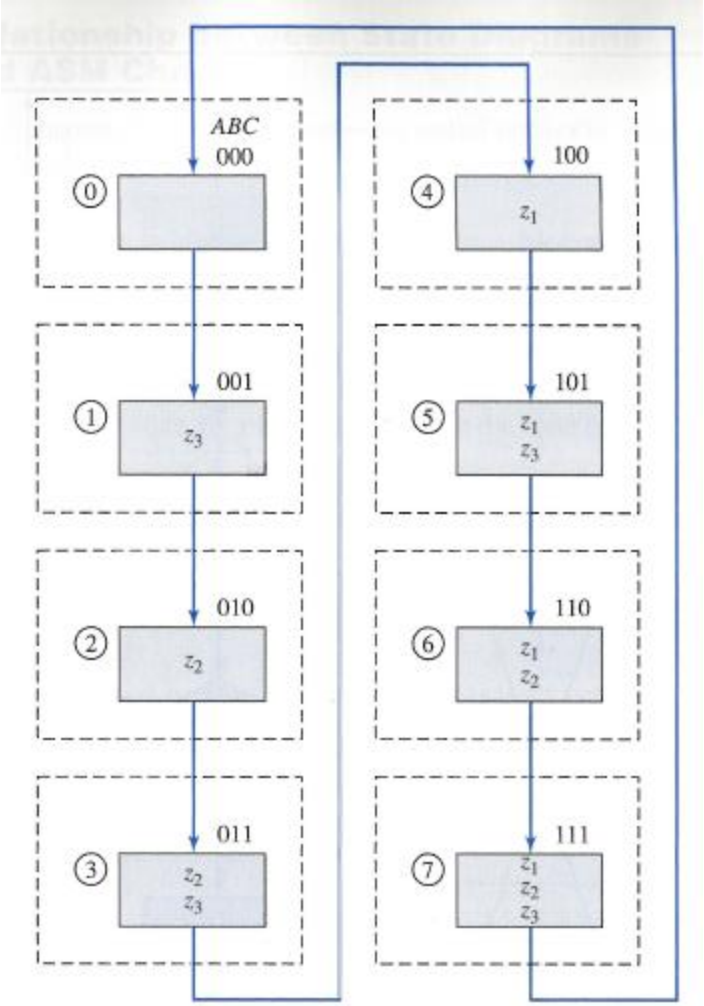


Figure 8.13 ASM chart for a mod-8 binary counter.

From State Diagram to ASM Chart

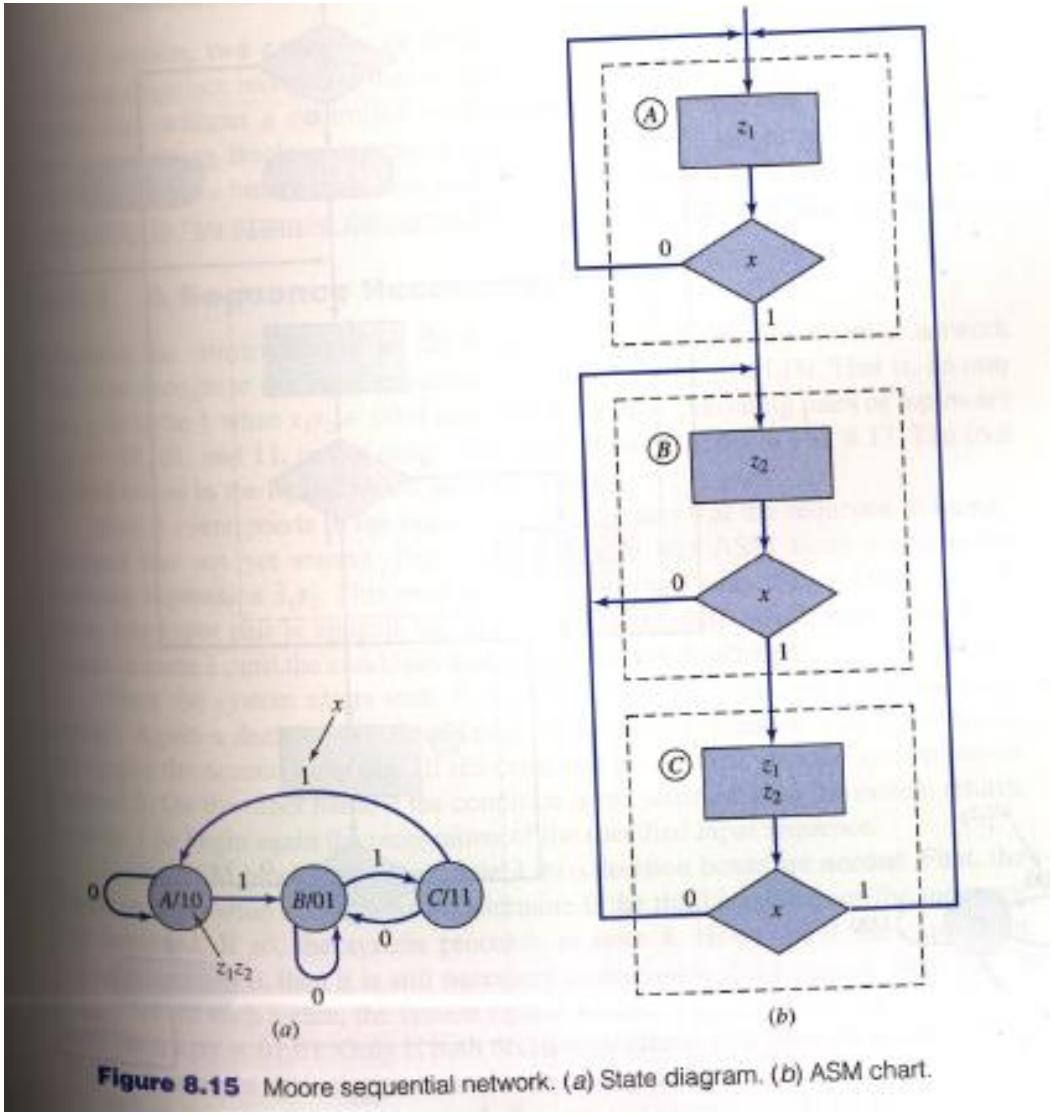


Figure 8.15 Moore sequential network. (*a*) State diagram. (*b*) ASM chart.

From State Diagram to ASM Chart

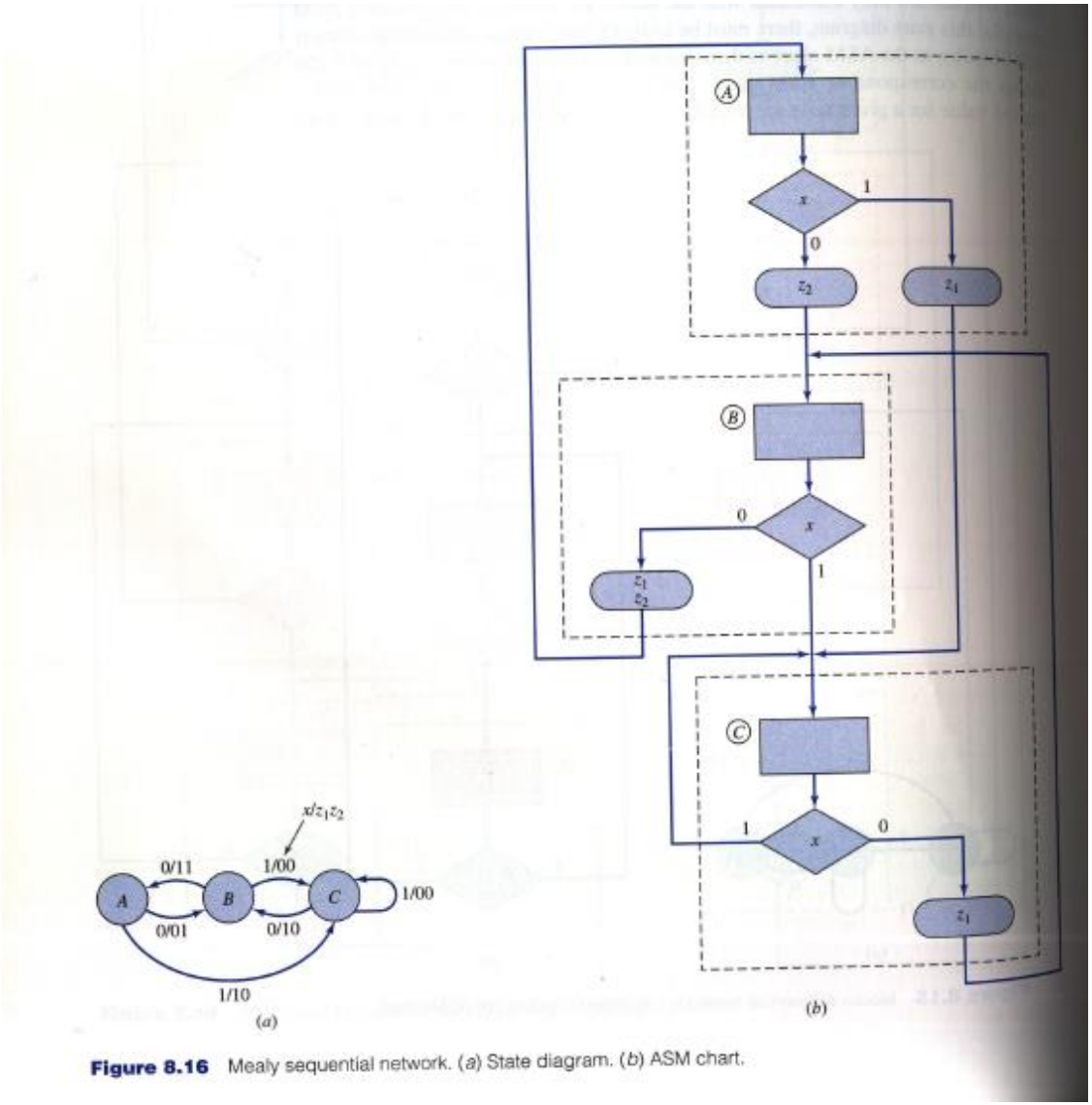


Figure 8.16 Mealy sequential network. (*a*) State diagram. (*b*) ASM chart.

Material after this slide is NOT on the Final Exam

Examples of ASM Charts

A Sequence Recognizer

- Recognize input sequence of pairs
$$x_1x_2 = 01,01,11,00$$
- An output z is to be 1 when $x_1x_2 = 00$ if and only if the three preceding pairs of inputs are
$$x_1x_2 = 01,01,11$$

in that order.

A Sequence Recognizer

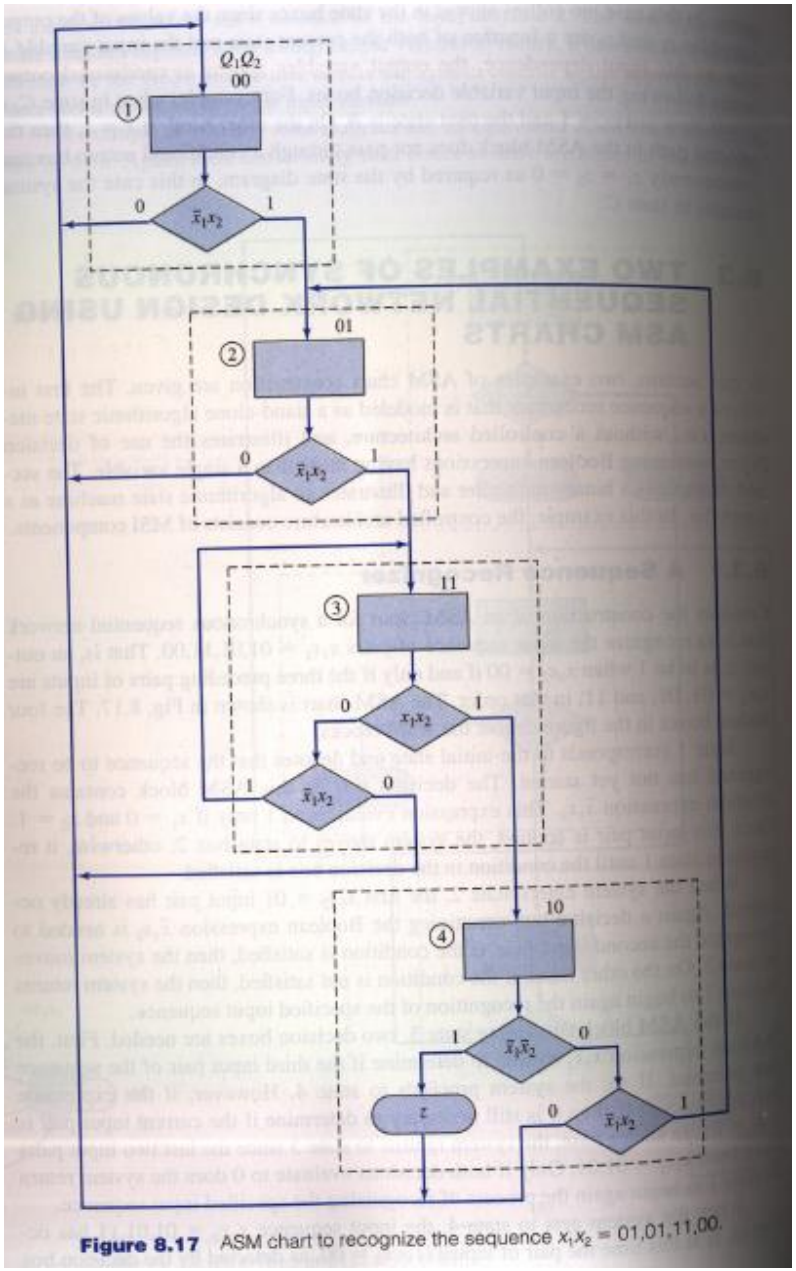


Figure 8.17 ASM chart to recognize the sequence x_1x_2 = 01,01,11,00.

A Parallel Binary Multiplier

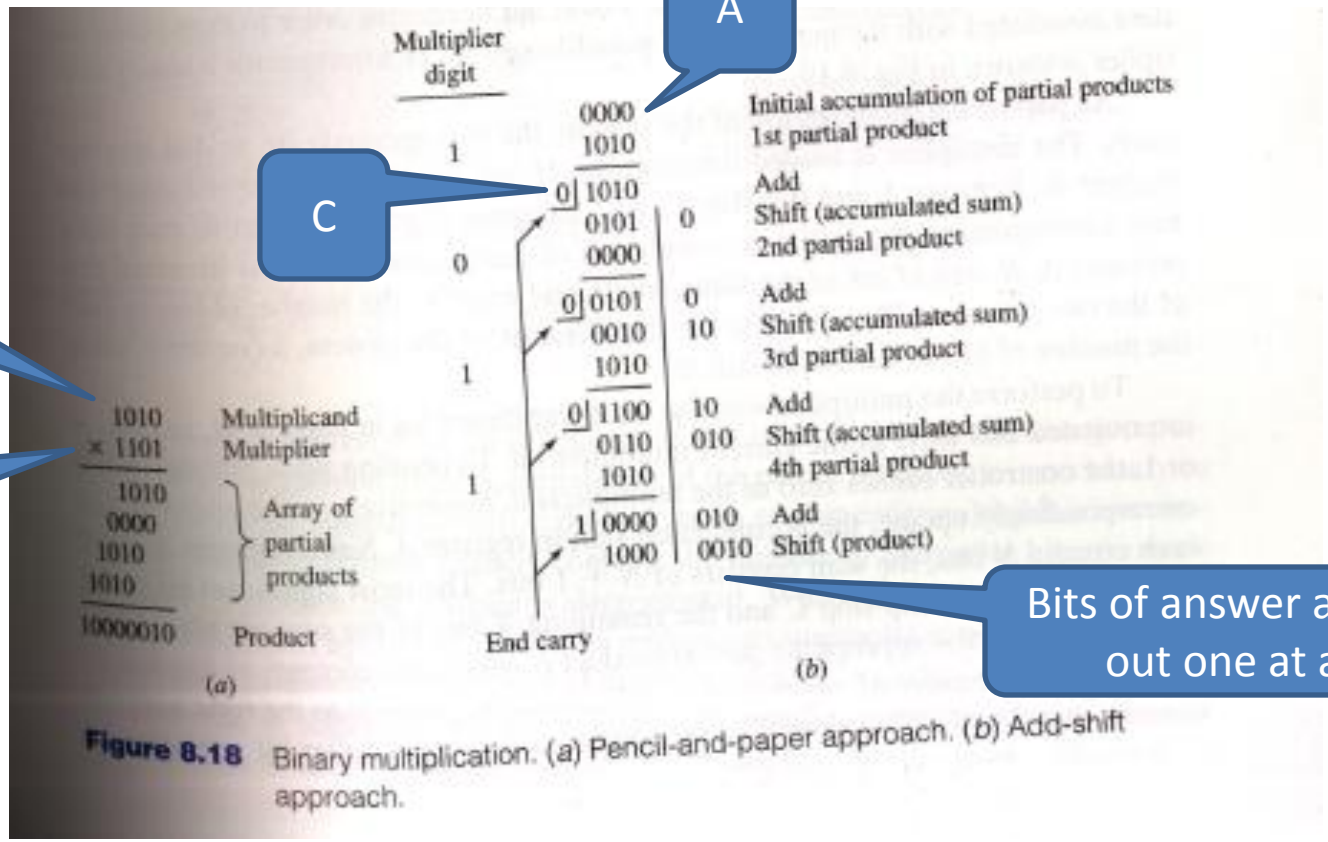


Figure 8.18 Binary multiplication. (a) Pencil-and-paper approach. (b) Add-shift approach.

A Parallel Binary Multiplier

Variables:

$S = 1$ indicates the multiplication is to start

M_1 is the multiplier bit appearing at the rightmost end of register M.

$Z = 1$ indicates the content of the counter is 0.

INIT = 1 indicates initialization should be performed:

1. Setting flip-flop C and register A to 0
2. Setting counter to the number of bits in the multiplier, N
3. Parallel loading the multiplier and multiplicand into registers M, B.

DECREM = 1 enables the counter for decrementing

ADD = 1 indicates A and B should be added and resulting N+1 bits entered into register A and flip-flop C

SR = 1 indicates the contents of flip-flop C, register A and register M should be shifted one bit position to the right while entering a 0 into flip-flop C.

COMPLETE = 1 indicates the multiplication process is complete.

A Parallel Binary Multiplier

Variables:

$S = 1$ indicates the multiplication is to start

M_1 is the multiplier bit appearing at the rightmost end of register M.

$Z = 1$ indicates the content of the counter is 0.

INIT = 1 indicates initialization should be performed:

1. Setting flip-flop C and register A to 0
2. Setting counter to the number of bits in the multiplier, N
3. Parallel loading the multiplier and multiplicand into registers M, B.

DECREM = 1 enables the counter for decrementing

ADD = 1 indicates A and B should be added and resulting N+1 bits entered into register A and flip-flop C

SR = 1 indicates the contents of flip-flop C, register A and register M should be shifted one bit position to the right while entering a 0 into flip-flop C.

COMPLETE = 1 indicates the multiplication process is complete.

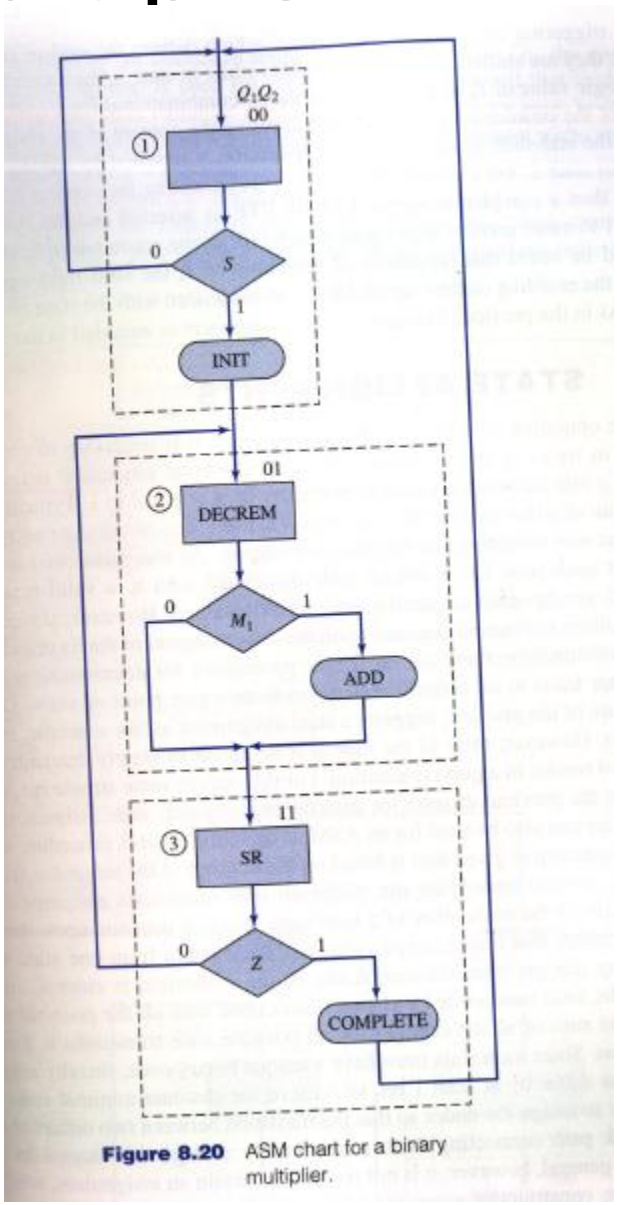


Figure 8.20 ASM chart for a binary multiplier.