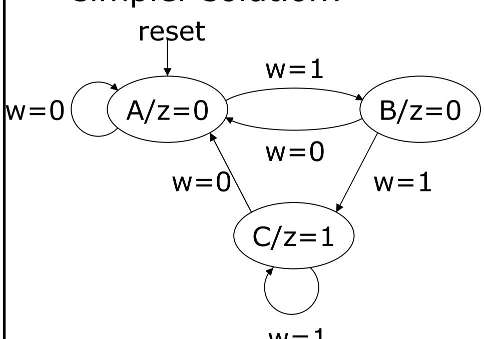
ECE380 Digital Logic

Synchronous Sequential Circuits:

State Assignment Problem,
Mealy State Machines

State assignment problem

- For the sequential circuit examples shown thus far, we have considered only a simple, straightforward state assignment
- Could another, different state assignment lead to a simpler solution?



	Present state y_2y_1	Next state w=0 Y_2Y_1	Next state w=1 Y_2Y_1	Output z
A	00	00	01	0
B	01	00	10	0
C	10	00	10	1
	11	dd	dd	d

Alternate state assignment

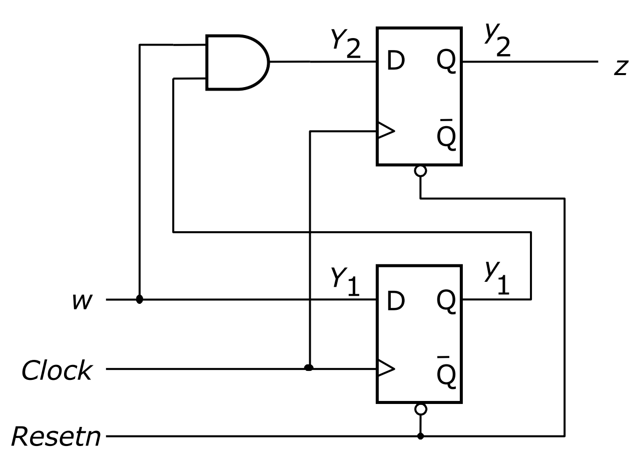
- If we change the state assignment for the previous problem such that A=00, B=01, C=11 and 10 is the unused state, the state assigned state table becomes the following

Present state	Next state w=0	Next state w=1	Output z
A	A	B	0
B	A	C	0
C	A	C	1

	Present state y_2y_1	Next state w=0 Y_2Y_1	Next state w=1 Y_2Y_1	Output z
A	00	00	01	0
B	01	00	11	0
C	11	00	11	1
	10	dd	dd	d

Simplified circuit implementation

$Y_1=D_1=w$
$Y_2=D_2=wy_1$
$z=y_2$



State assignment problem

- In general, circuits are much larger than the example given, and different state assignments can have a significant impact on the cost of the final implementation
- It is often impossible (impractical) to find the best assignment for a large circuit because the number of available states is large
- CAD tools usually perform state assignment using heuristic techniques

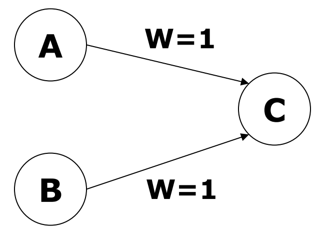
State assignment guidelines

- Does <u>not</u> guarantee a minimum solution
- Assignments for states are adjacent if they differ in only one state variable

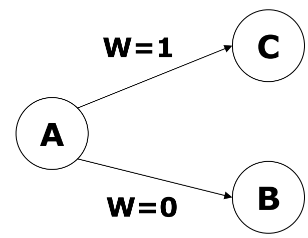
1. States that have the same next state for a given input should be given adjacent assignments
2. States that are the next state of the same state should be given adjacent assignments
3. States that have the same output for a given input should be given adjacent assignments (groups ‘1’s on the output K-map)

State assignment guidelines

1. States that have the same next state for a given input should be given adjacent assignments



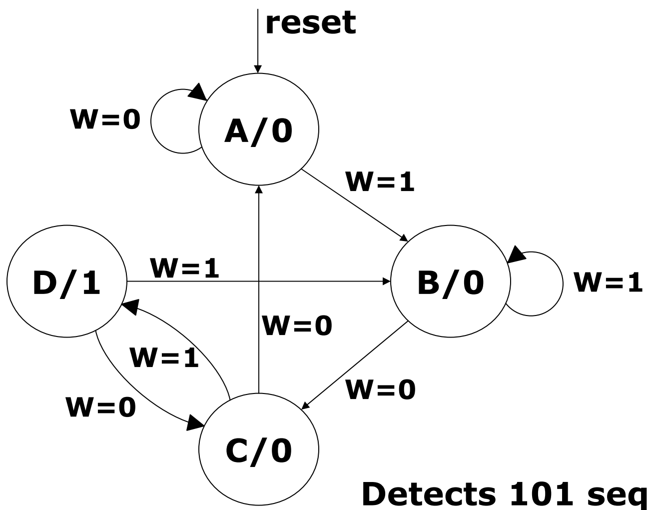
2. States that are the next state of the same state should be given adjacent assignments



State assignment guidelines

- Keep the following in mind:
  1. Assign the starting state to the '0' cell on the map (i.e. the starting state has all flip-flop outputs=0)
  2. Satisfy guideline 1 and multiple occurrences of guideline 2 first
  3. If the guidelines require that 3 or 4 states be adjacent, place these states within a group of 4 adjacent squares on the map
  4. Guideline 3 is less important than 1 or 2 unless the circuit is to have multiple outputs

Example Moore state diagram



Detects 101 sequence

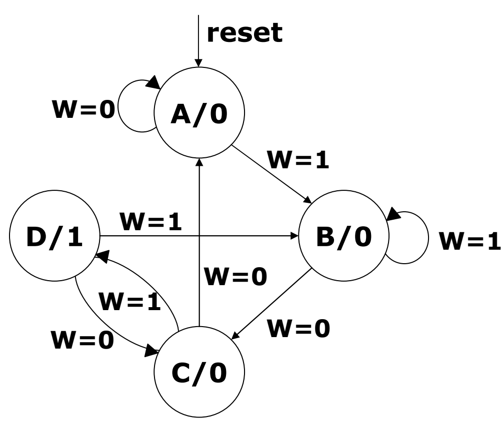
Example state assignment

- Two state variables $y_1 y_0$
- According to guidelines
- A=00

1. {A,D}, {B,D}, {A,C}
2. {A,D}, {B,C}
3. {A,B,C}

$y_1 \backslash y_0$	0	1
0	A	C
1	D	B

A=00
B=11
C=01
D=10

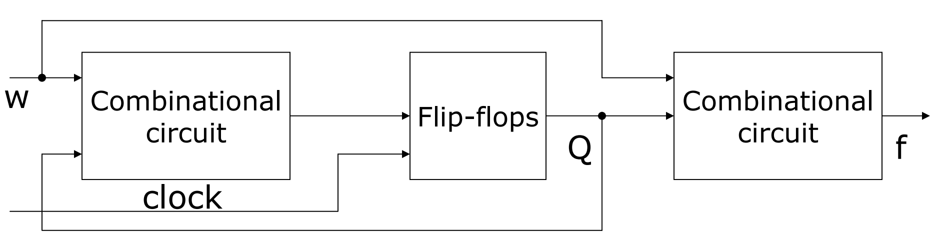


One hot encoding

- One possibility for state assignment is to use as many state variables as there are states in a sequential circuit
- For each state, all but one of the state variables are 0
- The variable whose value is 1 is deemed to be hot
  - ***One-hot encoding*** method
- Increases the number of flip-flops needed for implementation, but tends to lead to simpler output expressions
  - Simpler output expressions may lead to a faster circuit since there may be less propagation delay from the flip-flop outputs to the final outputs of the sequential circuit

Mealy state model

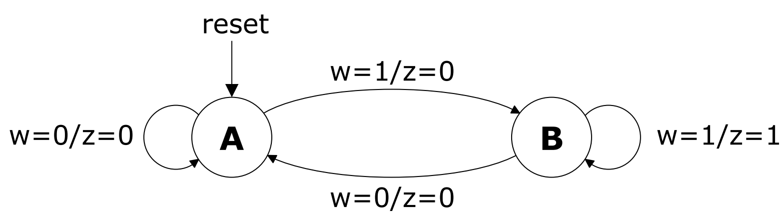
- **Mealy model**: circuit outputs depend on the present state of the circuit and the primary inputs, giving additional flexibility in designing sequential circuits
- Greater flexibility often leads to simpler circuits



Mealy state machine

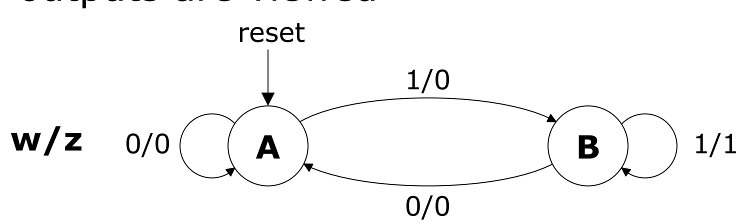
Mealy state diagram

- For the Mealy model, outputs are no longer associated with a particular state
  - Outputs are associated with transitions between states
- Typical Mealy model state diagram
  - Detects w=11 sequence



Mealy model state table

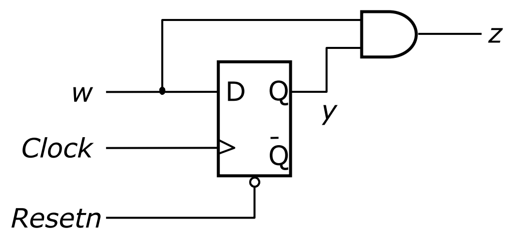
- The state table for a Mealy model FSM differs from the Moore model FSM only in how the outputs are viewed



Present state	Next state		Output z	
	$w = 0$	$w = 1$	$w = 0$	$w = 1$
A	A	B	0	0
B	A	B	0	1

State-assigned state table

Present state	Next state		Output	
	$w = 0$	$w = 1$	$w = 0$	$w = 1$
y	Y	Y	z	z
A 0	0	1	0	0
B 1	0	1	0	1



Design example

- Construct a Mealy state diagram for a sequence detector that detects the input sequence w=101