

Figure 8.1. The general form of a sequential circuit.

Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	0	1	0	0	1	1	0

Figure 8.2. Sequences of input and output signals.

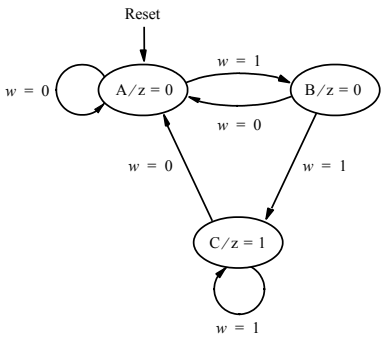


Figure 8.3. State diagram of a simple sequential circuit.

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

Figure 8.4. State table.

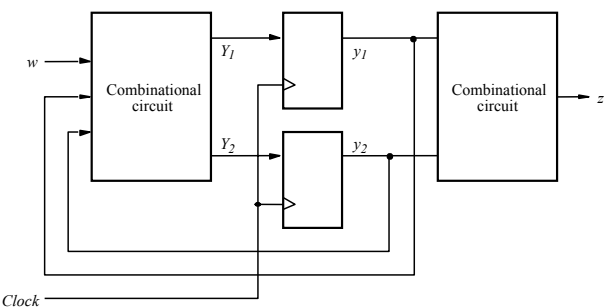


Figure 8.5. A general sequential circuit.

	Present state	Next state		Output z
		$w = 0$	$w = 1$	
	$y_2 y_1$	$Y_2 Y_1$	$Y_2 Y_1$	
A	00	00	01	0
B	01	00	10	0
C	10	00	10	1
	11	dd	dd	d

Figure 8.6. A state-assigned table.

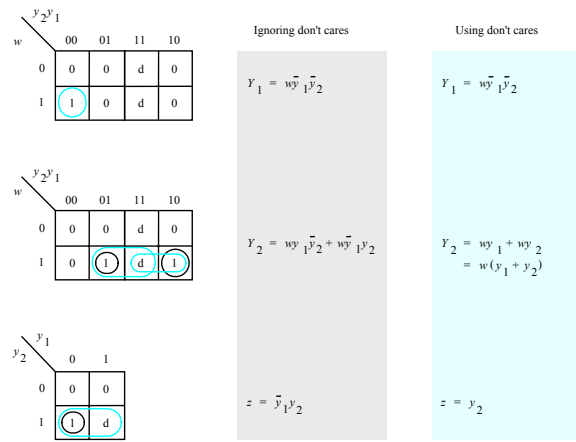


Figure 8.7. Derivation of logic expressions.

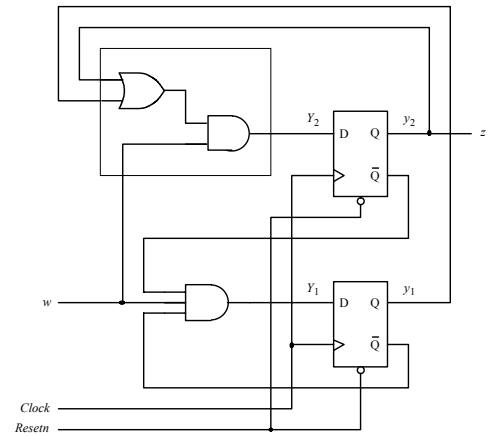


Figure 8.8. Sequential circuit derived in Figure 8.7.

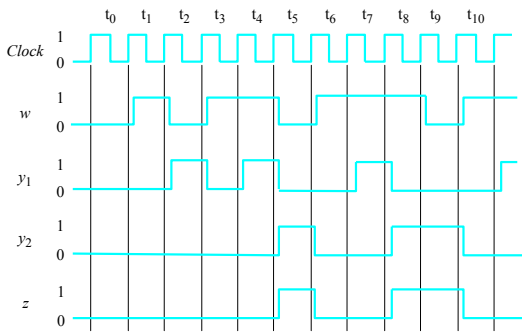


Figure 8.9. Timing diagram.

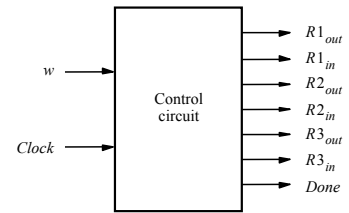


Figure 8.10. Signals needed in Example 8.1.

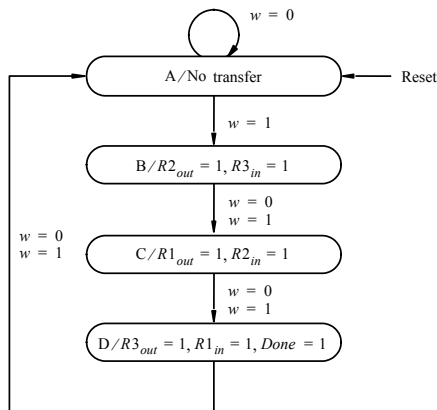


Figure 8.11. State diagram.

Present state	Next state		Outputs						
	w=0	w=1	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	Done
A	A	B	0	0	0	0	0	0	0
B	C	C	0	0	1	0	0	1	0
C	D	D	1	0	0	1	0	0	0
D	A	A	0	1	0	0	1	0	1

Figure 8.12. State table.

Present state y_2y_1	Nextstate		Outputs						
	$w = 0$	$w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	Done
	Y_2Y_1	Y_2Y_1							
A 00	00	01	0	0	0	0	0	0	0
B 01	10	10	0	0	1	0	0	1	0
C 10	11	11	1	0	0	1	0	0	0
D 11	00	00	0	1	0	0	1	0	1

Figure 8.13. State-assigned table.

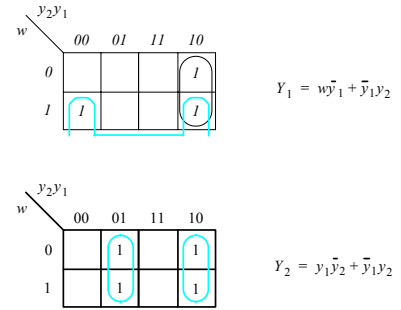


Figure 8.14. Derivation of next-state expressions.

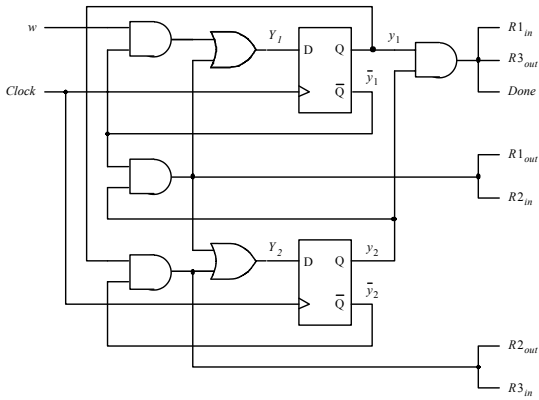


Figure 8.15. Sequential circuit derived in Figure 8.14.

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

Figure 8.16. Improved state assignment for the sequential circuit in Figure 8.4.

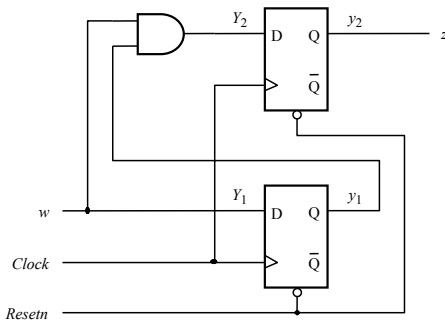


Figure 8.17. Final circuit for the improved state assignment.

	Present state y_2y_1	Nextstate		Outputs						
		$w = 0$	$w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	Done
		Y_2Y_1	Y_2Y_1							
A	00	00	01	0	0	0	0	0	0	0
B	01	11	11	0	0	1	0	0	1	0
C	11	10	10	1	0	0	1	0	0	0
D	10	00	00	0	1	0	0	1	0	1

Figure 8.18. Improved state assignment for the sequential circuit in Figure 8.12.

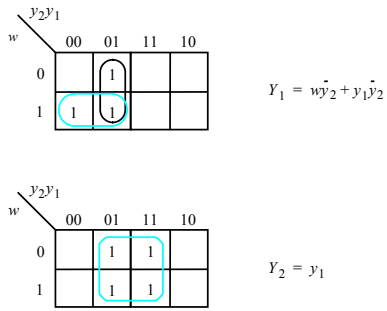


Figure 8.19. Derivation of next-state expressions.

Present state	Nextstate		Output z
	$w = 0$	$w = 1$	
$y_3y_2y_1$	$Y_3Y_2Y_1$	$Y_3Y_2Y_1$	
A	001	010	0
B	010	100	0
C	100	001	1

Figure 8.20. One-hot state assignment for the sequential circuit in Figure 8.4.

Present state	Nextstate		Outputs						
	$w = 0$	$w = 1$	$R1_{out}$	$R1_m$	$R2_{out}$	$R2_m$	$R3_{out}$	$R3_m$	$Done$
$y_4y_3y_2y_1$	$Y_4Y_3Y_2Y_1$	$Y_4Y_3Y_2Y_1$							
0 001	0001	0010	0	0	0	0	0	0	0
0 010	0100	0100	0	0	1	0	0	1	0
0 100	1000	1000	1	0	0	1	0	0	0
1 000	0001	0001	0	1	0	0	1	0	1

Clock cycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	1	0	0	1	1	0	0

Figure 8.21. One-hot state assignment for the sequential circuit in Figure 8.12.

Figure 8.22. Sequences of input and output signals.

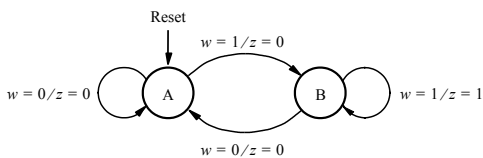


Figure 8.23. State diagram.

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

Figure 8.24. 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

Figure 8.25. State-assigned table.

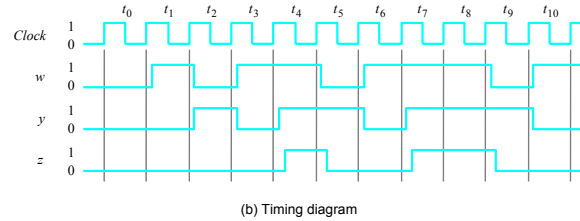
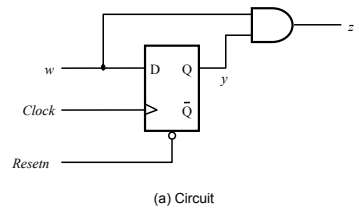


Figure 8.26. Implementation of FSM in Figure 8.25.

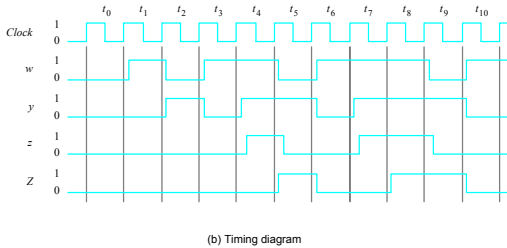
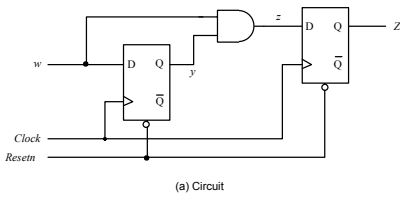


Figure 8.27. Circuit that implements the specification in Figure 8.2.

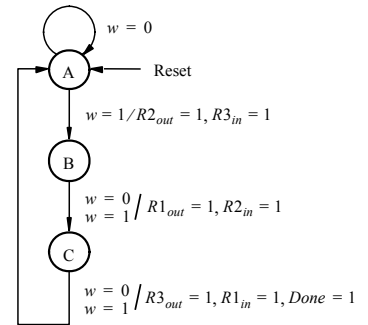


Figure 8.28. State diagram for Example 8.4.

```

module simple (Clock, Resetn, w, z);
input Clock, Resetn, w;
output z;
reg [2:1] y, Y;
parameter [2:1] A = 2'b00, B = 2'b01, C = 2'b10;

// Define the next state combinational circuit
always @(w or y)
  case (y)
    A: if (w) Y = B;
       else Y = A;
    B: if (w) Y = C;
       else Y = A;
    C: if (w) Y = C;
       else Y = A;
       default: Y = 2'bxx;
  endcase

// Define the sequential block
always @(negedge Resetn or posedge Clock)
  if (Resetn == 0) y <= A;
  else y <= Y;

// Define output
assign z = (y == C);
endmodule

```

Figure 8.29. Verilog code for the FSM in Figure 8.3.

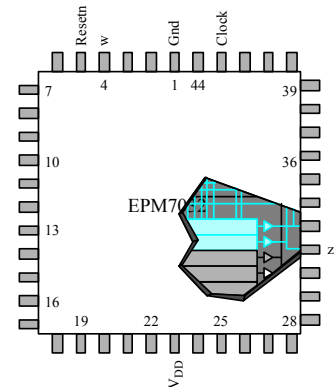


Figure 8.31. An FSM circuit in a small CPLD.

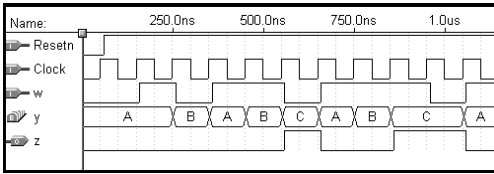


Figure 8.32. Simulation results.

```

module simple (Clock, Resetn, w, z);
input Clock, Resetn, w;
output z;
reg z;
reg [2:1] y, Y;
parameter [2:1] A = 2'b00, B = 2'b01, C = 2'b10;

// Define the next state combinational circuit
always @(w or y)
begin
    case (y)
        A: if (w) Y = B;
           else Y = A;
        B: if (w) Y = C;
           else Y = A;
        C: if (w) Y = C;
           else Y = A;
        default: Y = 2'bxxx;
    endcase
    z = (y == C); //Define output
end

// Define the sequential block
always @(negedge Resetn or posedge Clock)
if (Resetn == 0) y <= A;
else y <= Y;
endmodule

```

Figure 8.33. Second version of code for the FSM in Figure 8.3.

```

module simple (Clock, Resetn, w, z);
input Clock, Resetn, w;
output z;
reg [2:1] y;
parameter [2:1] A = 2'b00, B = 2'b01, C = 2'b10;

// Define the sequential block
always @(negedge Resetn or posedge Clock)
if (Resetn == 0) y <= A;
else
    case (y)
        A: if (w) y <= B;
           else y <= A;
        B: if (w) y <= C;
           else y <= A;
        C: if (w) y <= C;
           else y <= A;
        default: y <= 2'bxx;
    endcase

// Define output
assign z = (y == C);
endmodule

```

Figure 8.34. Third version of code for the FSM in Figure 8.3.

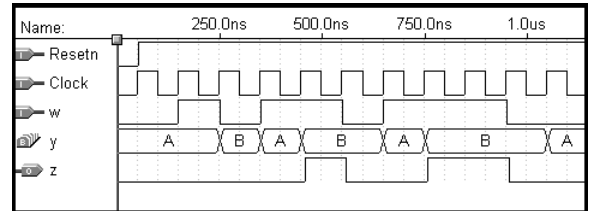


Figure 8.37. Simulation results for the Mealy machine.

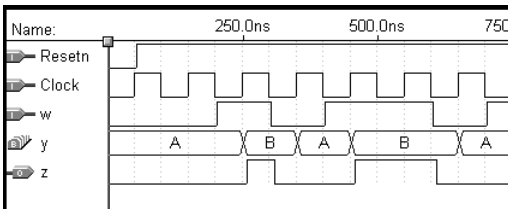


Figure 8.38. Potential problem with asynchronous inputs to a Mealy FSM.

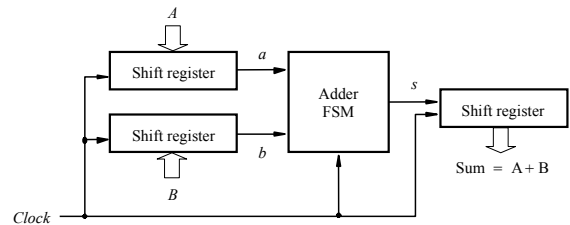


Figure 8.39. Block diagram of a serial adder.

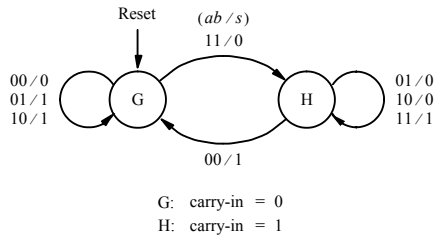


Figure 8.40. State diagram for the serial adder.

Present state	Next state				Output s			
	$ab=00$	01	10	11	00	01	10	11
G	G	G	G	H	0	1	1	0
H	G	H	H	H	1	0	0	1

Figure 8.41. State table for the serial adder.

Present state	Next state				Output			
	$ab=00$	01	10	11	00	01	10	11
y	Y				s			
0	0	0	0	1	0	1	1	0
1	0	1	1	1	1	0	0	1

Figure 8.42. State-assigned table for the serial adder.

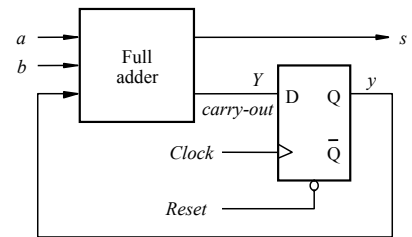


Figure 8.43. Circuit for the adder FSM.

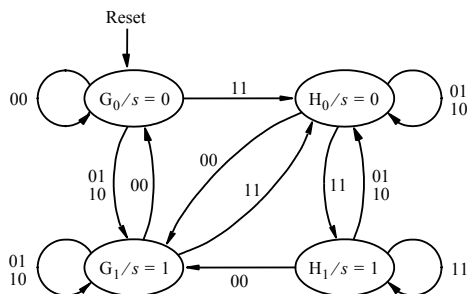


Figure 8.44. State diagram for the Moore-type serial adder FSM.

Present state	Nextstate				Output s
	$ab=00$	01	10	11	
G_0	G_0	G_1	G_1	H_0	0
G_1	G_0	G_1	G_1	H_0	1
H_0	G_1	H_0	H_0	H_1	0
H_1	G_1	H_0	H_0	H_1	1

Figure 8.45. State table for the Moore-type serial adder FSM.

Present state y_2y_1	Nextstate				Output s
	$ab=00$	01	10	11	
00	00	01	01	10	0
01	00	01	01	10	1
10	01	10	10	11	0
11	01	10	10	11	1

Figure 8.46. State-assigned table for the Moore-type serial adder FSM.

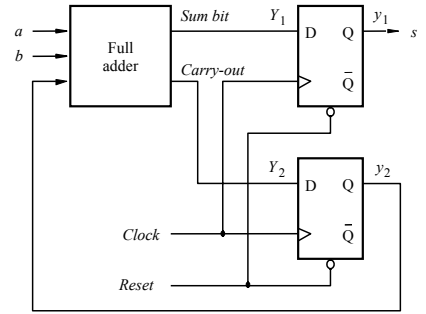


Figure 8.47. Circuit for the Moore-type serial adder FSM.

```

module shiftme (R, L, E, w, Clock, Q);
  parameter n = 8;
  input [n-1:0] R;
  input L, E, w, Clock;
  output [n-1:0] Q;
  reg [n-1:0] Q;
  integer k;

  always @(posedge Clock)
  if (L)
    Q <= R;
  else if (E)
  begin
    for (k = n-1; k > 0; k = k-1)
      Q[k-1] <= Q[k];
    Q[n-1] <= w;
  end
endmodule

```

Figure 8.48. Code for a left-to-right shift register with an enable input.

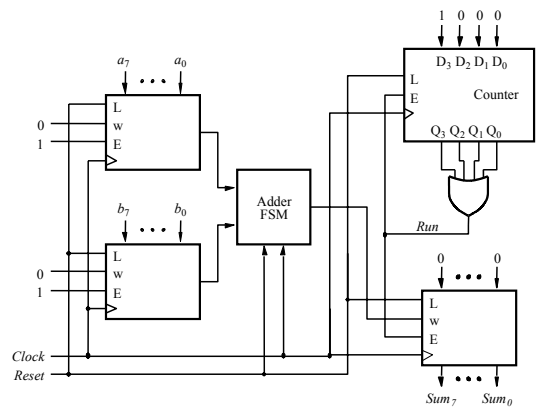


Figure 8.50a. Synthesized serial adder.

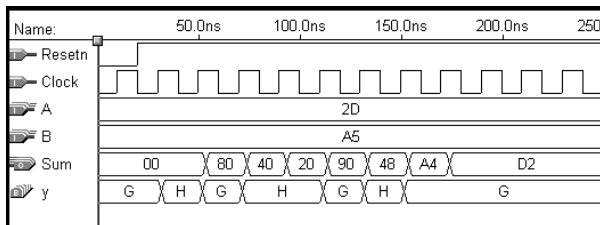


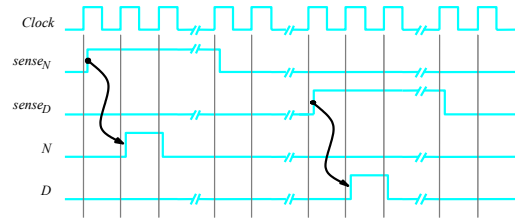
Figure 8.50b. Simulation results for the synthesized serial adder.

Present state	Next state		Output z
	$w = 0$	$w = 1$	
A	B	C	1
B	D	F	1
C	F	E	0
D	B	G	1
E	F	C	0
F	E	D	0
G	F	G	0

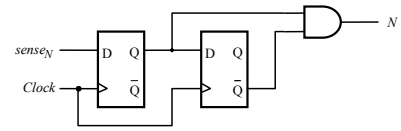
Figure 8.51. State table for Example 8.6.

Present state	Nextstate		Output z
	w = 0	w = 1	
A	B	C	1
B	A	F	1
C	F	C	0
F	C	A	0

Figure 8.52. Minimized state table for Example 8.6.



(a) Timing diagram



(b) Circuit that generates N

Figure 8.53. Signals for the vending machine.

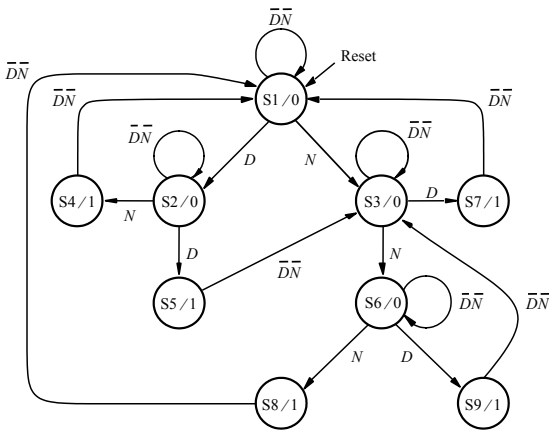


Figure 8.54. State diagram for Example 8.7.

Present state	Next state				Output z
	DN = 00	01	10	11	
S1	S1	S3	S2	-	0
S2	S2	S4	S5	-	0
S3	S3	S6	S7	-	0
S4	S1	-	-	-	1
S5	S3	-	-	-	1
S6	S6	S8	S9	-	0
S7	S1	-	-	-	1
S8	S1	-	-	-	1
S9	S3	-	-	-	1

Figure 8.55. State table for Example 8.7.

Present state	Next state				Output z
	DN = 00	01	10	11	
S1	S1	S3	S2	-	0
S2	S2	S4	S5	-	0
S3	S3	S2	S4	-	0
S4	S1	-	-	-	1
S5	S3	-	-	-	1

Figure 8.56. Minimized state table for Example 8.7.

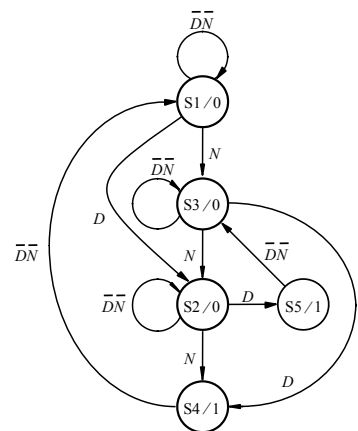


Figure 8.57. Minimized state diagram for Example 8.7.

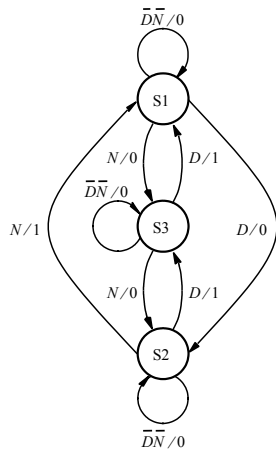


Figure 8.58. Mealy-type FSM for Example 8.7.

Present state	Next state		Output z	
	w = 0	w = 1	w = 0	w = 1
A	B	C	0	0
B	D	-	0	-
C	F	E	0	1
D	B	G	0	0
E	F	C	0	1
F	E	D	0	1
G	F	-	0	-

Figure 8.59. Incompletely specified state table for Example 8.8.

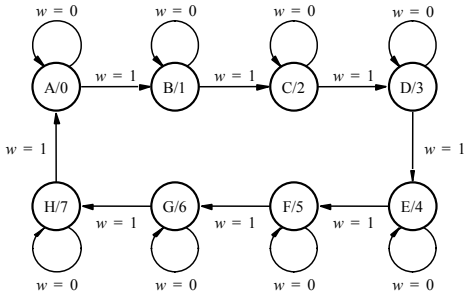


Figure 8.60. State diagram for a counter.

Present state	Next state		Output
	w = 0	w = 1	
A	A	B	0
B	B	C	1
C	C	D	2
D	D	E	3
E	E	F	4
F	F	G	5
G	G	H	6
H	H	A	7

Figure 8.61. State table for the counter.

	Present state $y_2y_1y_0$	Next state		Count $z_2z_1z_0$
		w = 0	w = 1	
		$Y_2Y_1Y_0$	$Y_2Y_1Y_0$	
A	000	000	001	000
B	001	001	010	001
C	010	010	011	010
D	011	011	100	011
E	100	100	101	100
F	101	101	110	101
G	110	110	111	110
H	111	111	000	111

Figure 8.62. State-assigned table for the counter.

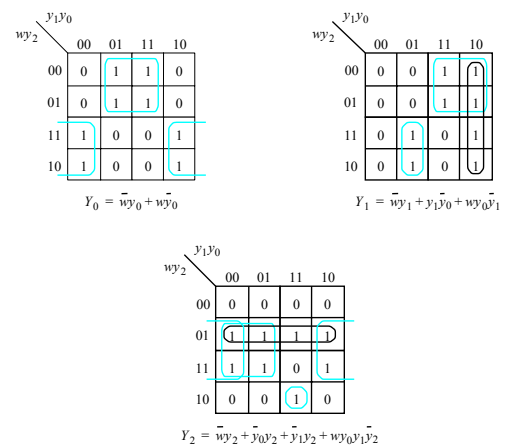


Figure 8.63. Karnaugh maps for D flip-flops for the counter.

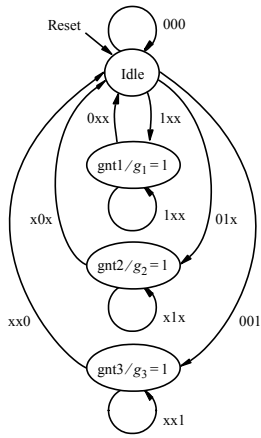


Figure 8.72. State diagram for the arbiter.

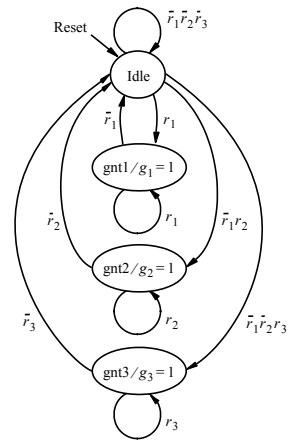


Figure 8.73. Alternative style of state diagram for the arbiter.

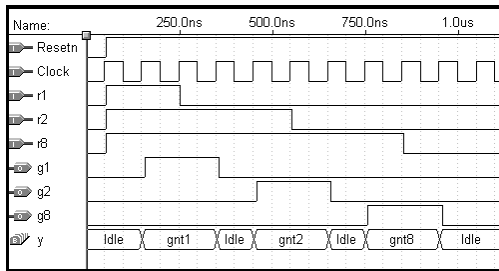


Figure 8.75. Simulation results for the arbiter circuit.

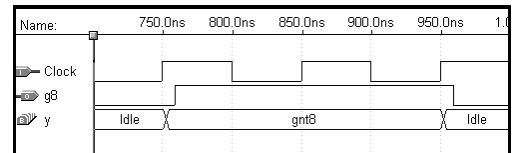


Figure 8.76. Output delays in the arbiter circuit.

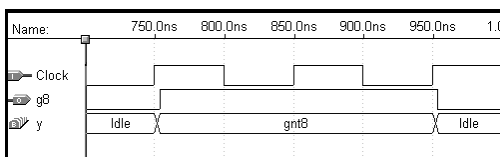


Figure 8.77. Output delay when using one-hot encoding.

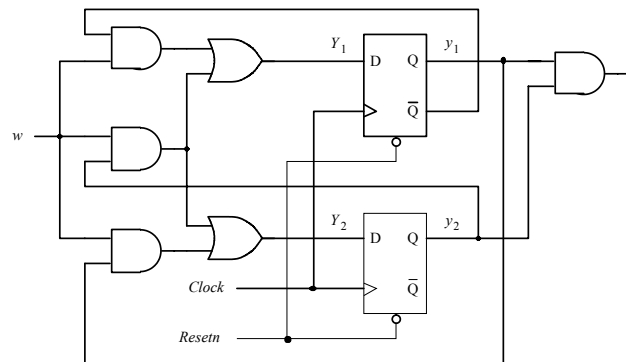


Figure 8.78. Circuit for Example 8.9.

Present state y_2y_1	Next State		Output z
	$w = 0$	$w = 1$	
	Y_2Y_1	Y_2Y_1	
00	00	01	0
01	00	10	0
10	00	11	0
11	00	11	1

(a) State-assigned table

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

(b) State table

Figure 8.79. Tables for the circuit in Example 8.9.

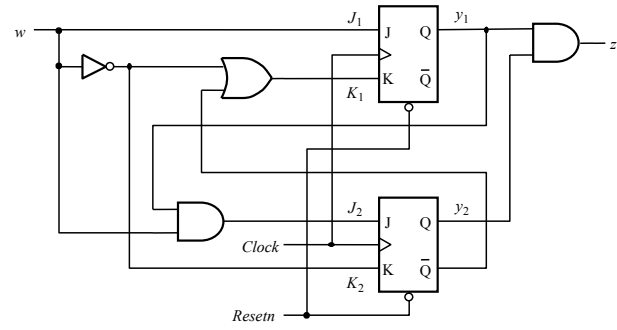


Figure 8.80. Circuit for Example 8.10.

Present state y_2y_1	Flip-flop inputs				Output z
	$w = 0$		$w = 1$		
	J_2K_2	J_1K_1	J_2K_2	J_1K_1	
00	01	01	00	11	0
01	01	01	10	11	0
10	01	01	00	10	0
11	01	01	10	10	1

Figure 8.81. Excitation table for the circuit in Figure 8.80.

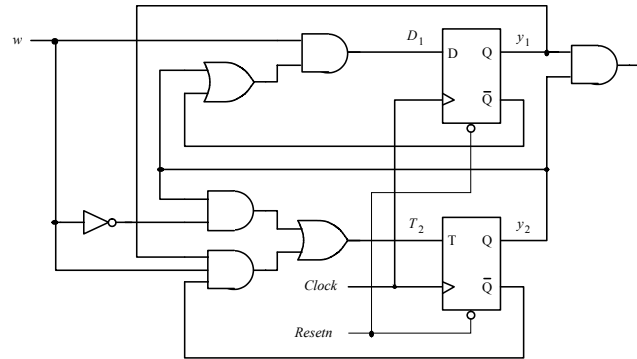


Figure 8.82. Circuit for Example 8.11.

Present state y_2y_1	Flip-flop inputs		Output z
	$w = 0$	$w = 1$	
	T_2D_1	T_2D_1	
00	00	01	0
01	00	10	0
10	10	01	0
11	10	01	1

Figure 8.83. Excitation table for the circuit in Figure 8.82.

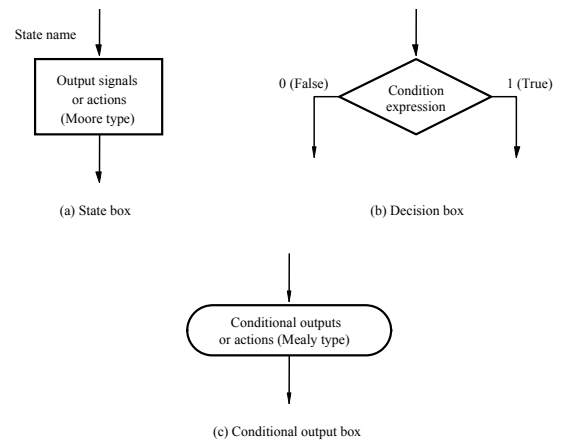


Figure 8.84. Elements used in ASM charts.

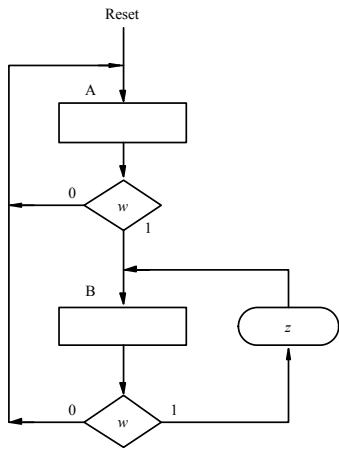


Figure 8.86. ASM chart for the FSM in Figure 8.23.

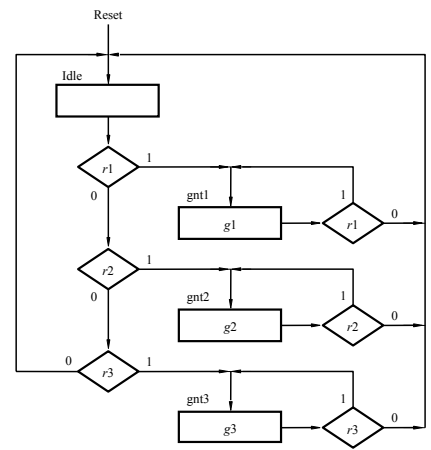


Figure 8.87. ASM chart for the arbiter.

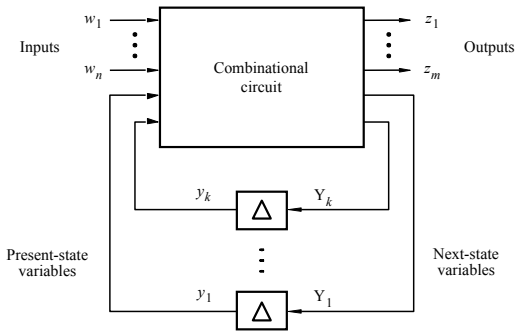


Figure 8.88. The general model for a sequential circuit.

Present state $y_2 y_1$	Next state		Output z
	$w = 0$ $Y_2 Y_1$	$w = 1$ $Y_2 Y_1$	
00	10	11	0
01	01	00	0
10	11	00	0
11	10	01	1

Figure P8.1. State-assigned table for problems 8.1 and 8.2.

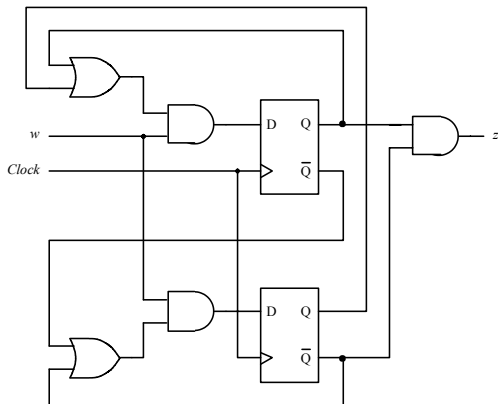


Figure P8.2. Circuit for problem 8.29.