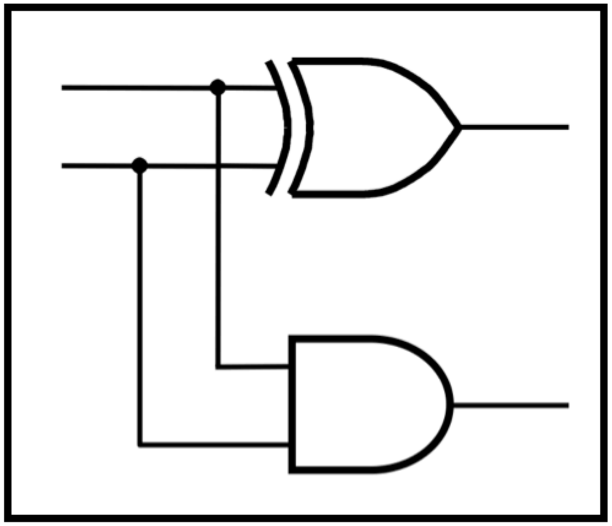


CprE 281: Digital Logic

Instructor: Alexander Stoytchev

http://www.ece.iastate.edu/~alexs/classes/

State Assignment Problem

CprE 281: Digital Logic
Iowa State University, Ames, IA
Copyright © Alexander Stoytchev

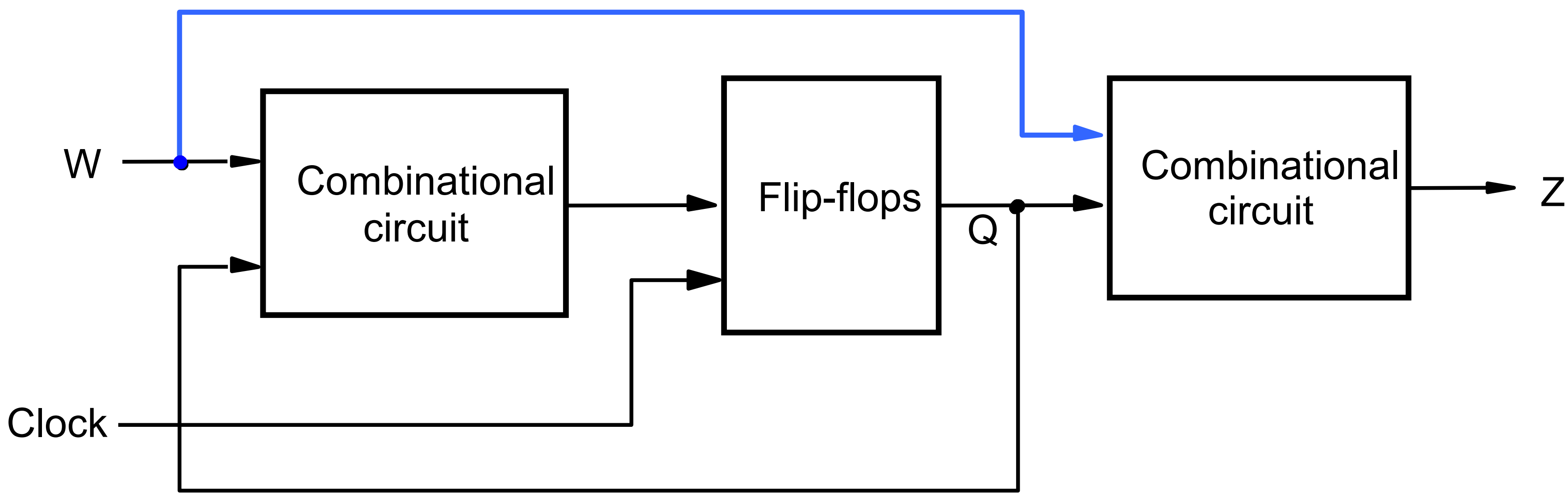
Administrative Stuff

- Homework 10 is due on Nov 8 @ 10pm

- Homework 11 is due on Nov 11 @ 10pm

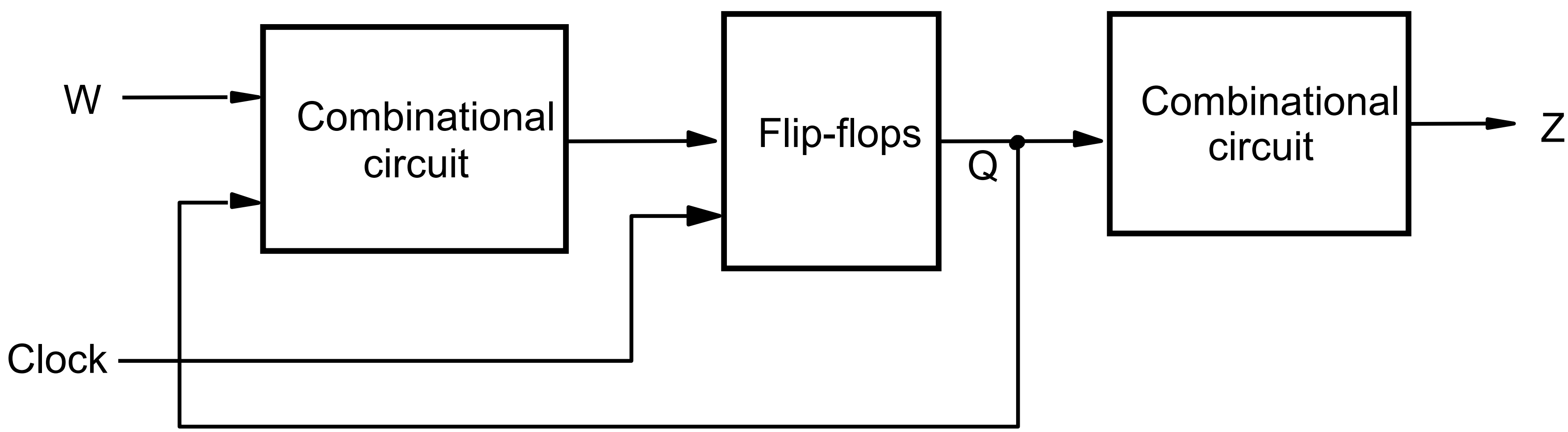
Quick Review

The general form of a synchronous sequential circuit



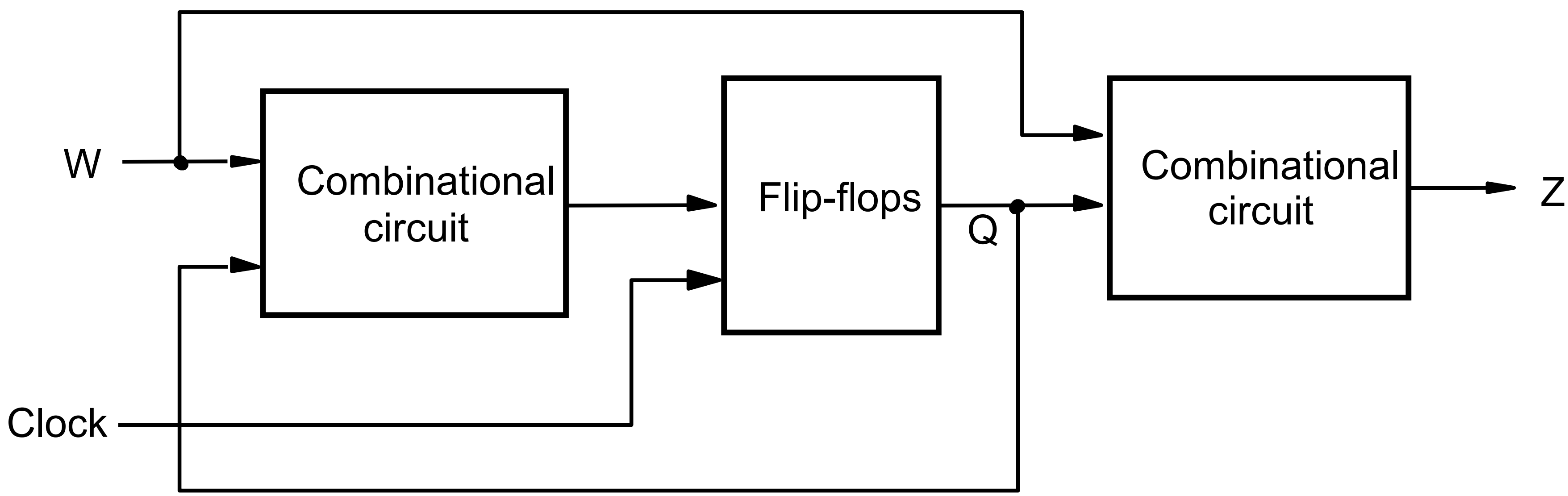
[Figure 6.1 from the textbook]

Moore Type



W
Combinational
circuit
Flip-flops
Q
Combinational
circuit
Z
Clock

Mealy Type


W
Combinational
circuit
Flip-flops
Q
Combinational
circuit
Z
Clock

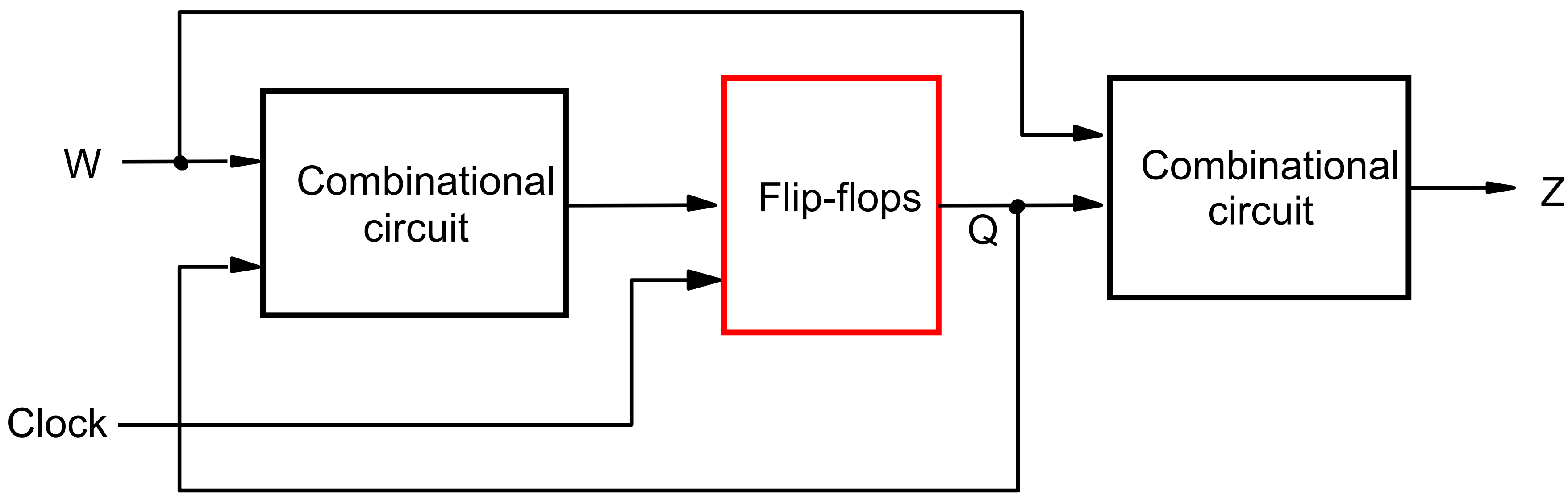
Moore Machine

- The machine’s current state and current inputs are used to decide which next state to transition into.
- The machine’s current state decides the current output.
- Named after Edward Moore (1925-2003) who published the idea in 1956.
- Moore, E. (1956). "Gedanken-experiments on Sequential Machines". Automata Studies, Annals of Math. Studies. Princeton Univ. Press (34): 129–153.

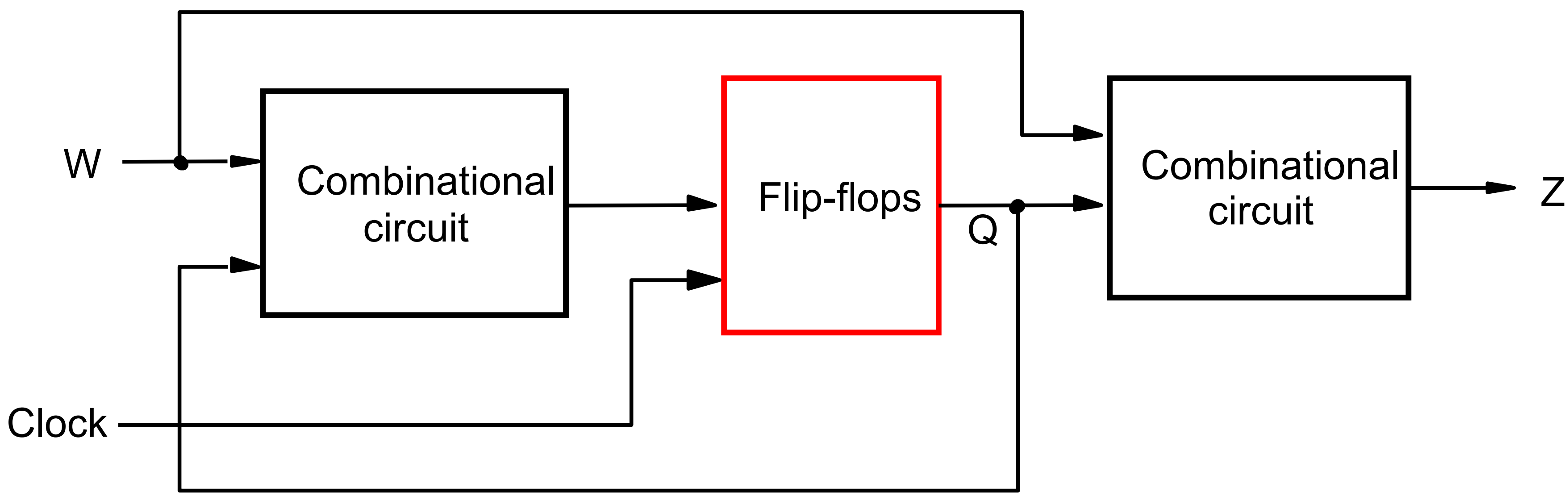
Mealy Machine

- The machine's current state and current inputs are used to decide which next state to transition into.
- The machine's current state and current input values decide the current output.
- Named after George Mealy (1927 – 2010) who published the idea in 1955.
- Mealy, G. (1955). "A method for synthesizing sequential circuits" The Bell System Technical Journal, Volume: 34, Issue: 5, Sept. 1955.

Today's lecture is about the mapping from states to flip-flop outputs

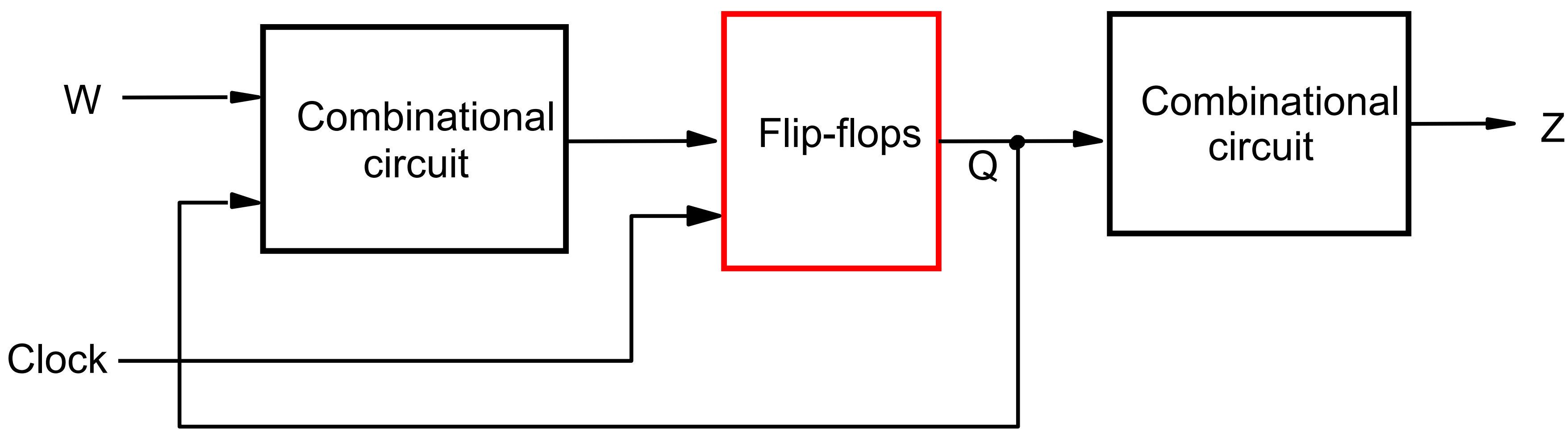


Today's lecture is about the mapping from states to flip-flop outputs



It applies to both Mealy and Moore machines…

Today's lecture is about the mapping from states to flip-flop outputs

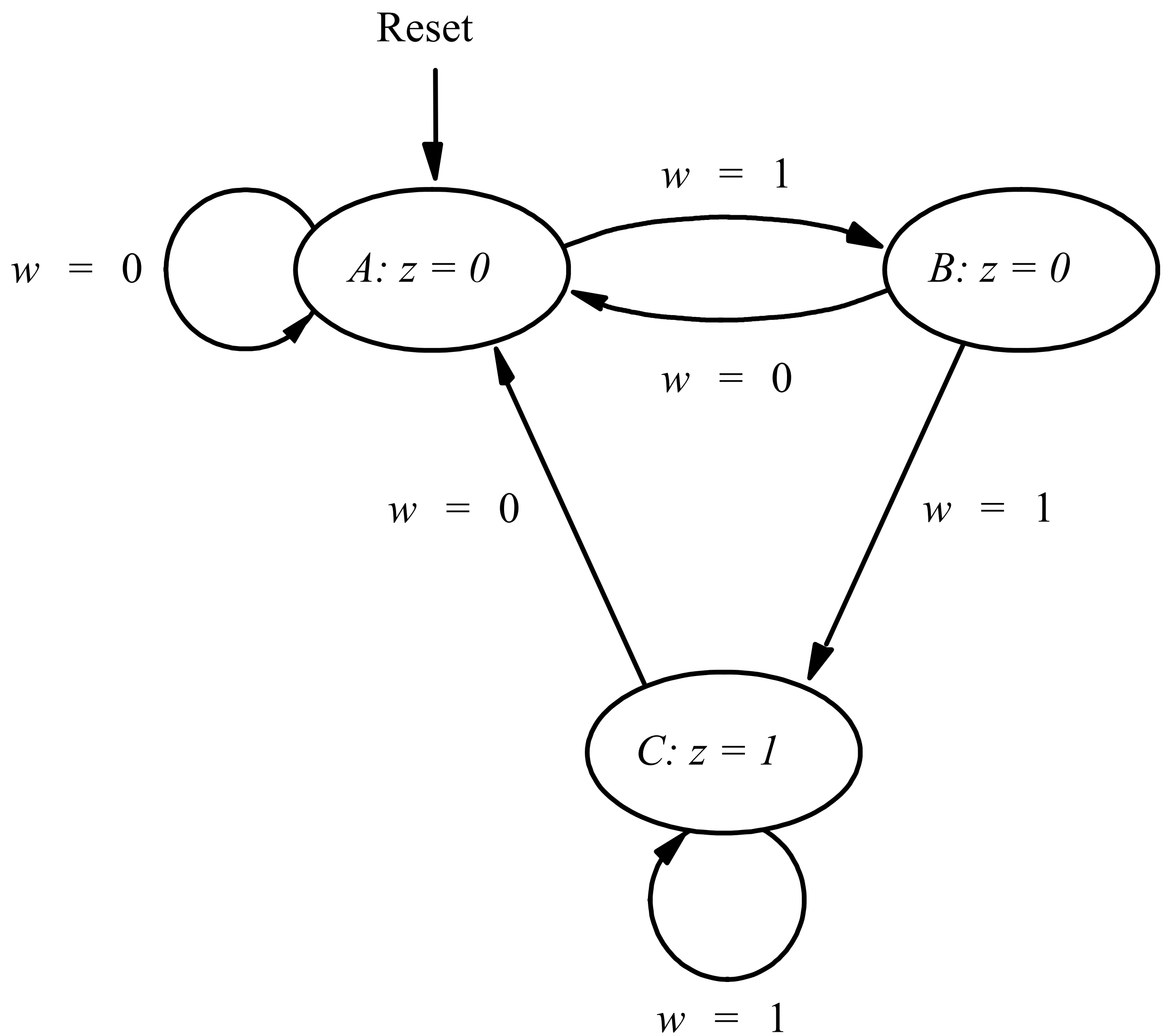


… but all examples will be for Moore machines.

Designing a Moore Machine

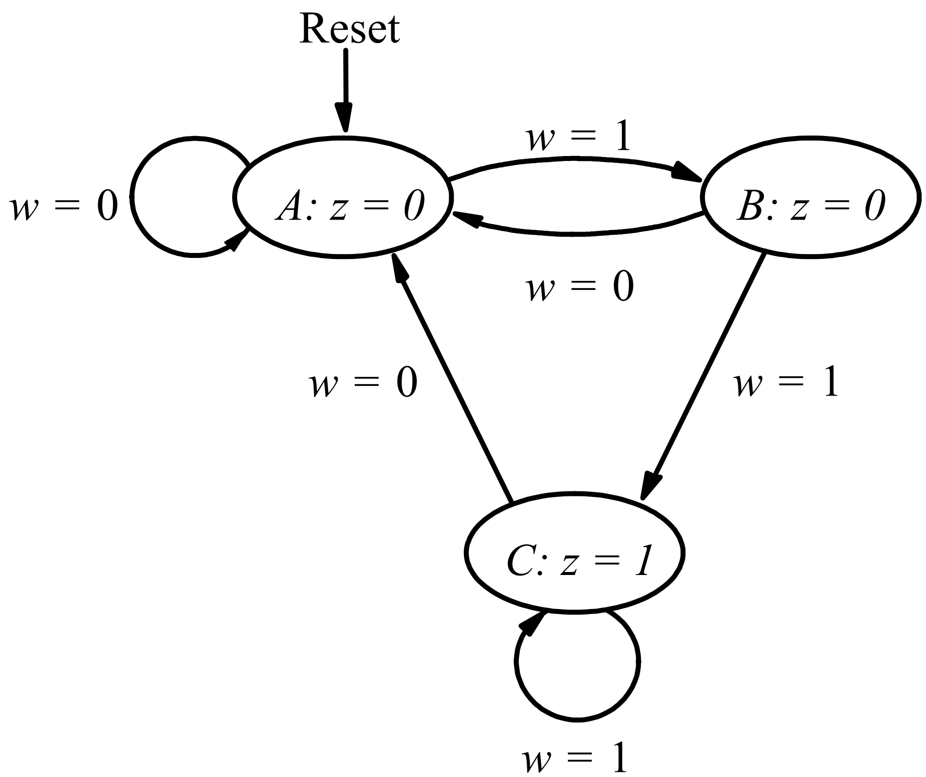
- Obtain the circuit specification.
- Derive a state diagram.
- Derive the state table.
- Decide on a state encoding.
- Encode the state table.
- Derive the output logic and next-state logic.
- Draw the circuit diagram
- Add a reset signal.

Example #1

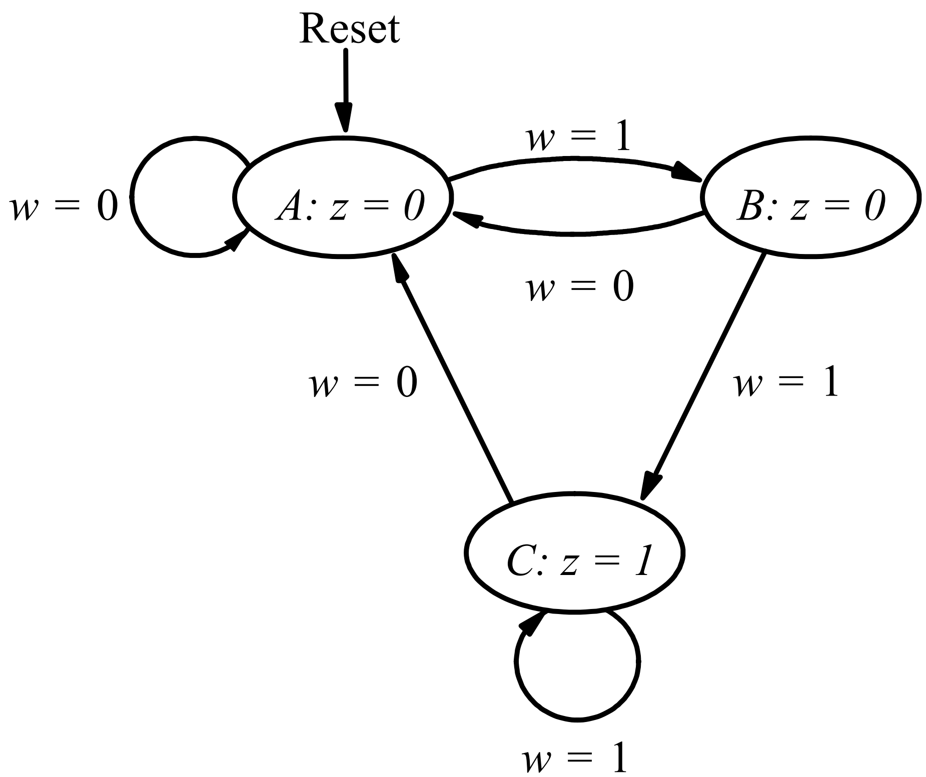


We need to find both the *next state logic* and the *output logic* implied by this machine.

[Figure 6.3 from the textbook]



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



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

[Figure 6.4 from the textbook]

State Encoding for Example #1: A=00, B=01, C=10

(Uses Two Flip-Flops)

How to represent the states?

One way is to encode each state with a 2-bit binary number

A = 00

B = 01

C = 10

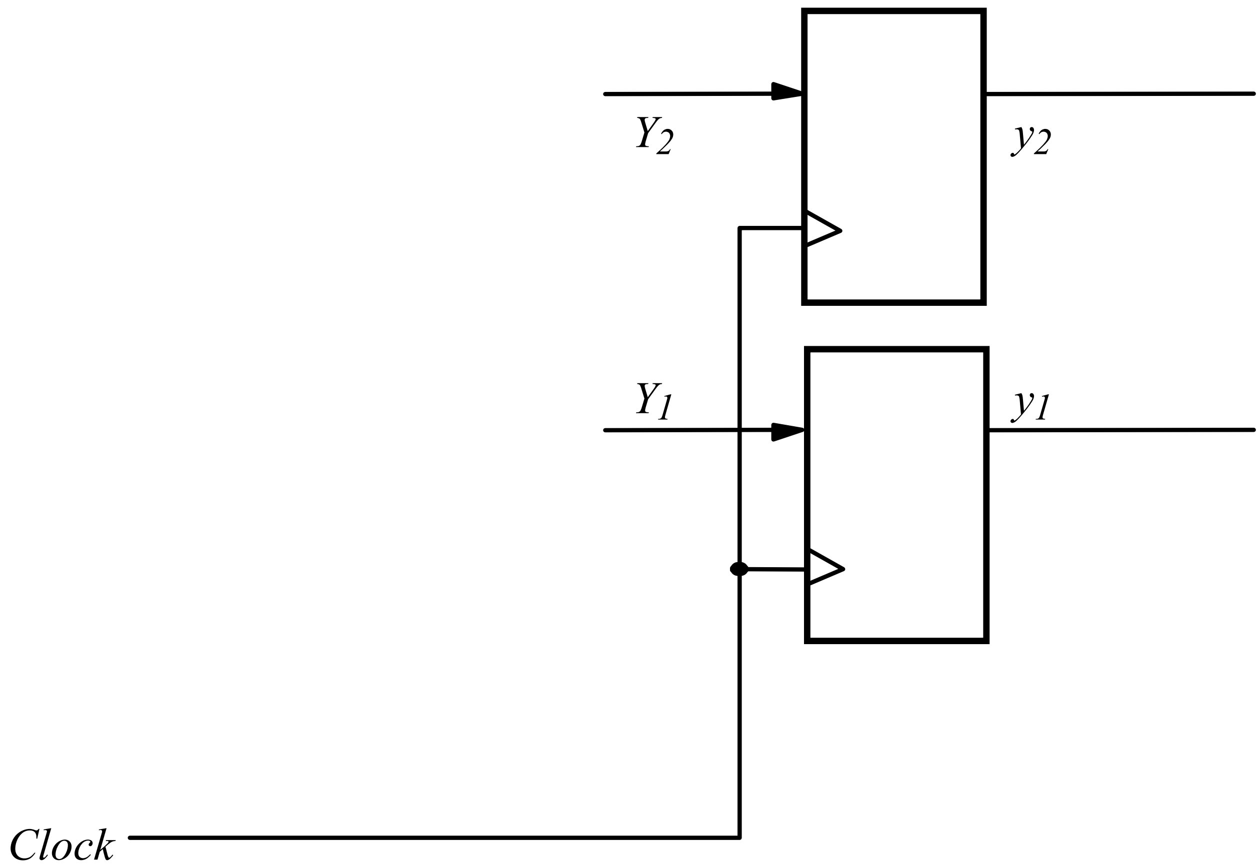
How to represent the states?

One way is to encode each state with a 2-bit binary number

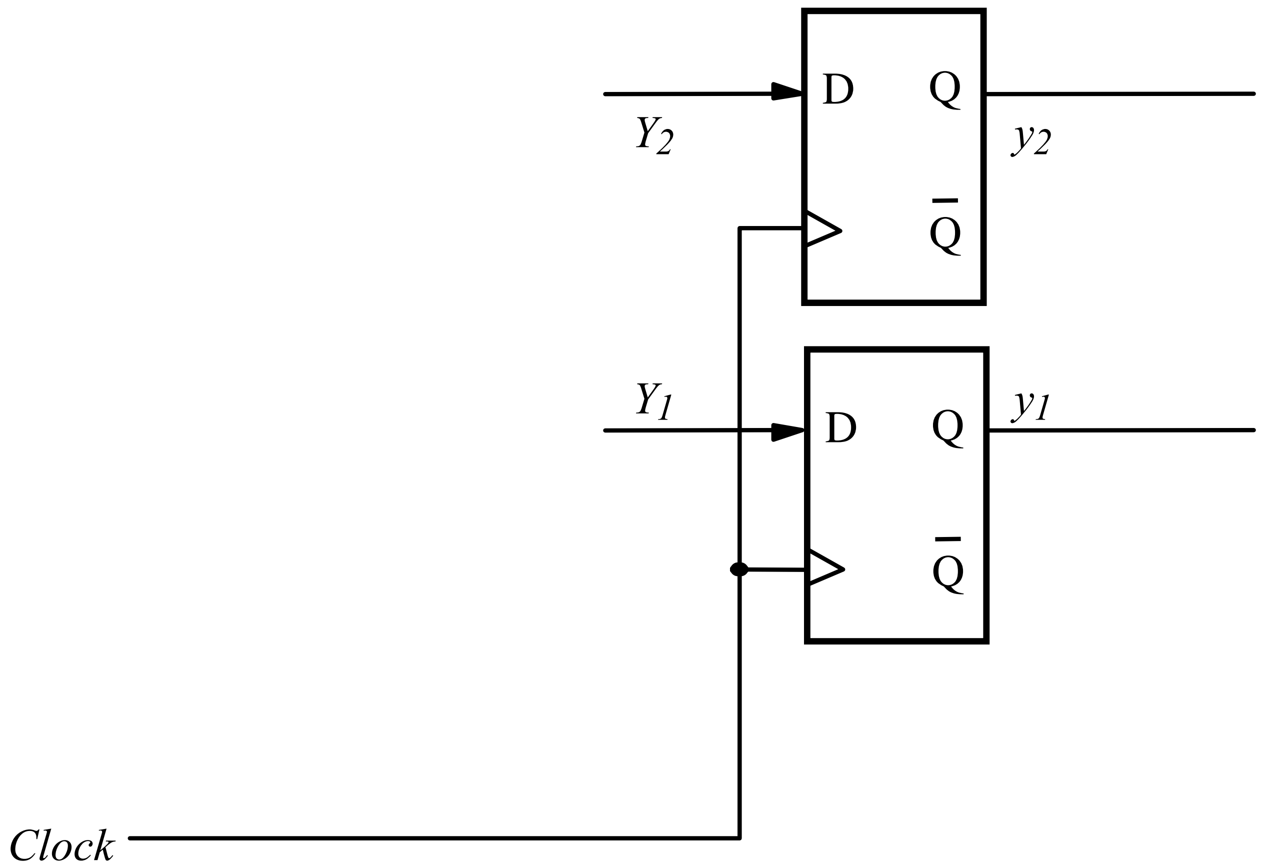
A = 00
B = 01
C = 10

How many flip-flops do we need?

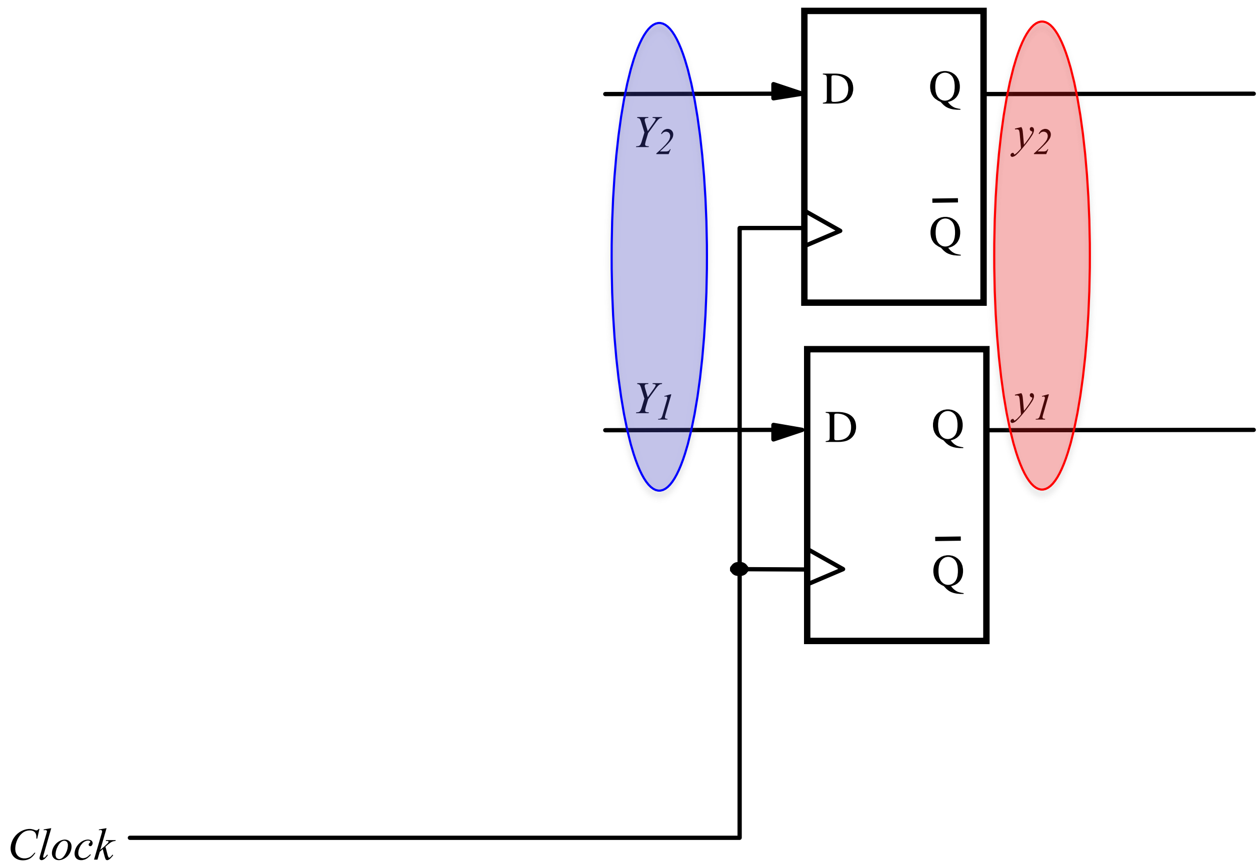
Let’s use two flip flops to hold the state of this machine



Y_2
y_2
Y_1
y_1
Clock



Let’s pick D Flip-Flops.

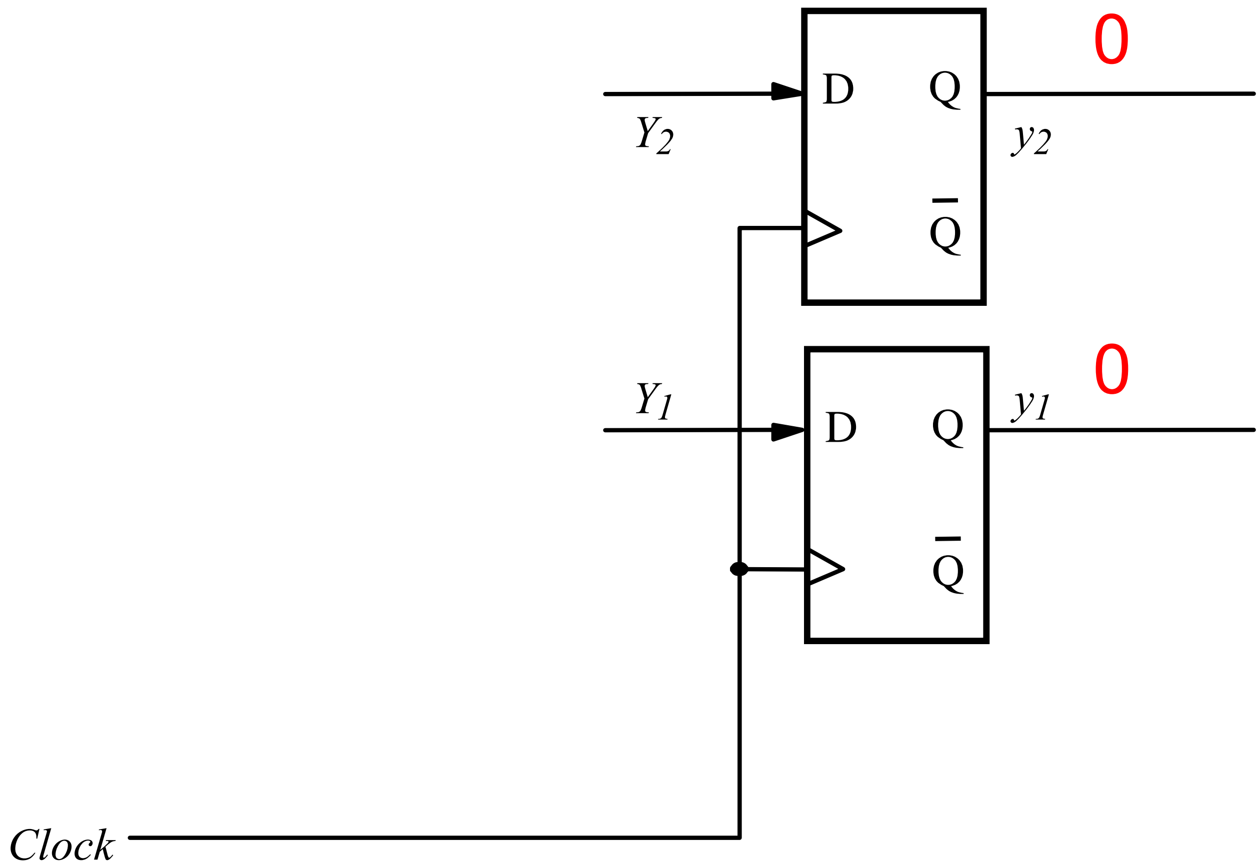


We will call y_1 and y_2 the *present state variables*.

We will call Y_1 and Y_2 the *next state variables*.

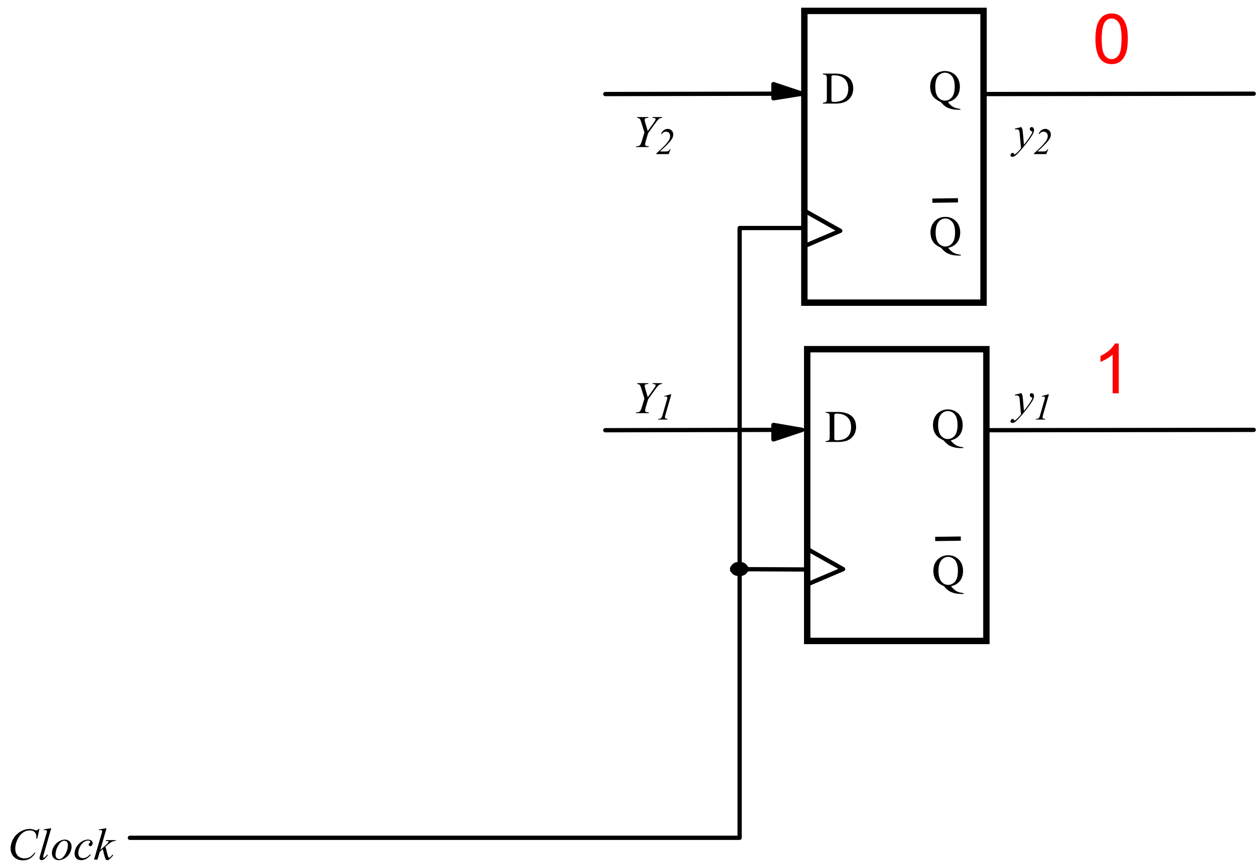
[Figure 6.5 from the textbook]

State = y_2 y_1



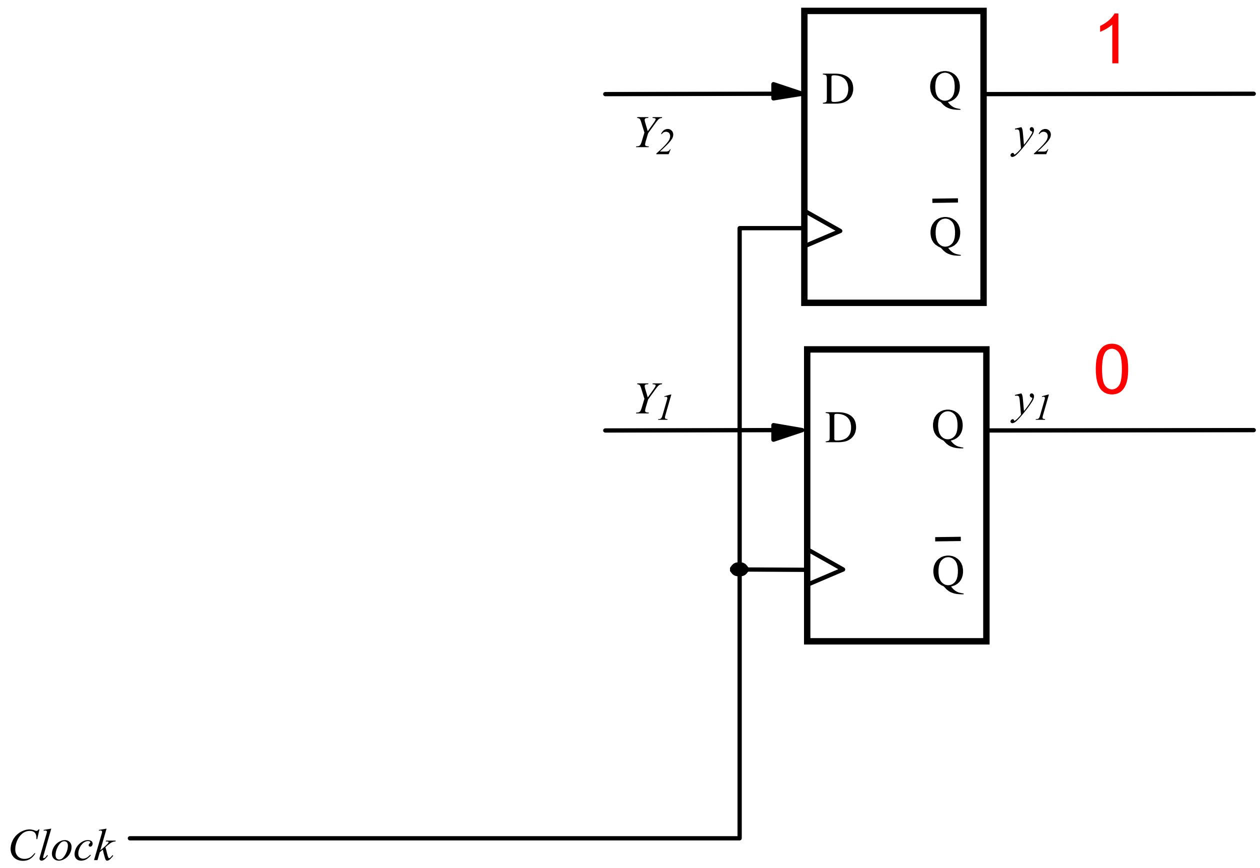
Two zeros on the output JOINTLY represent state A.

State = $y_2\ y_1$



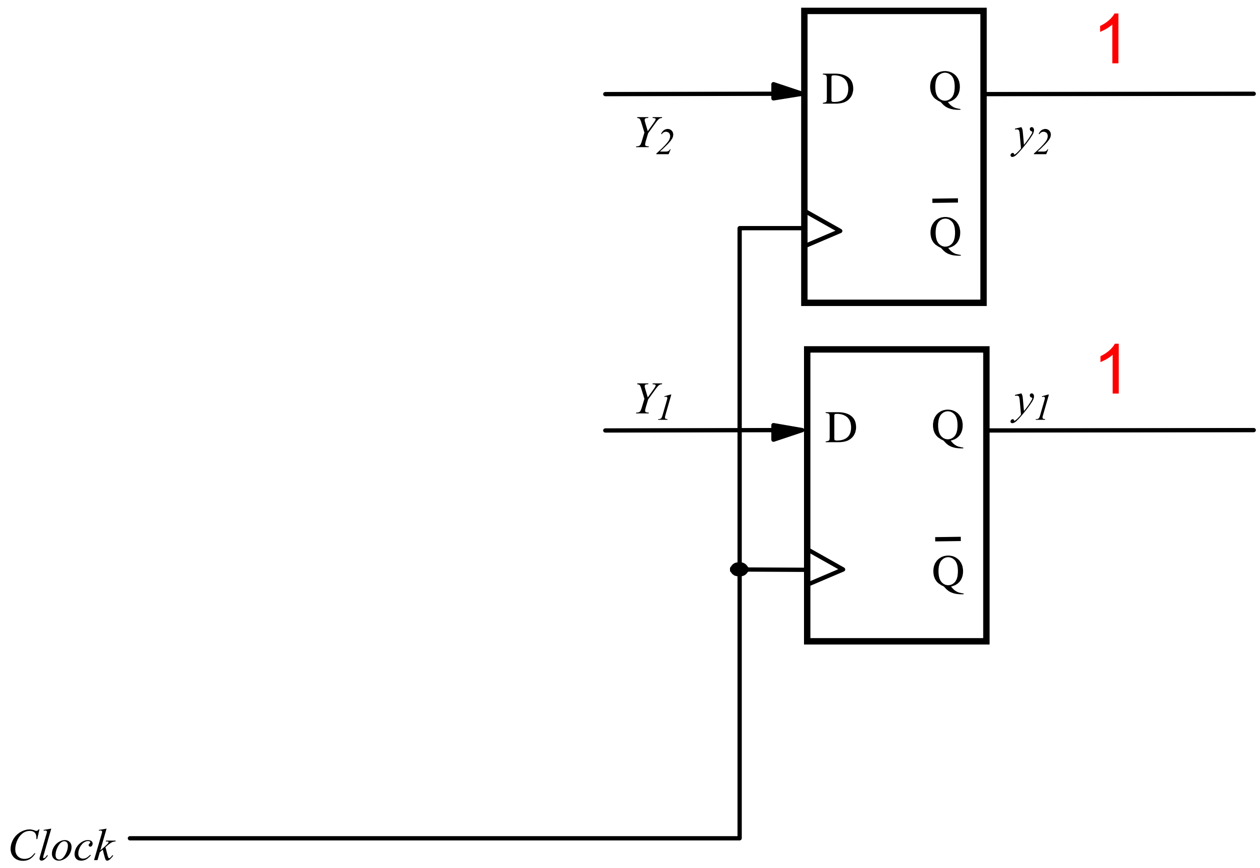
This flip-flop output pattern represents state B.

State = y_2 y_1



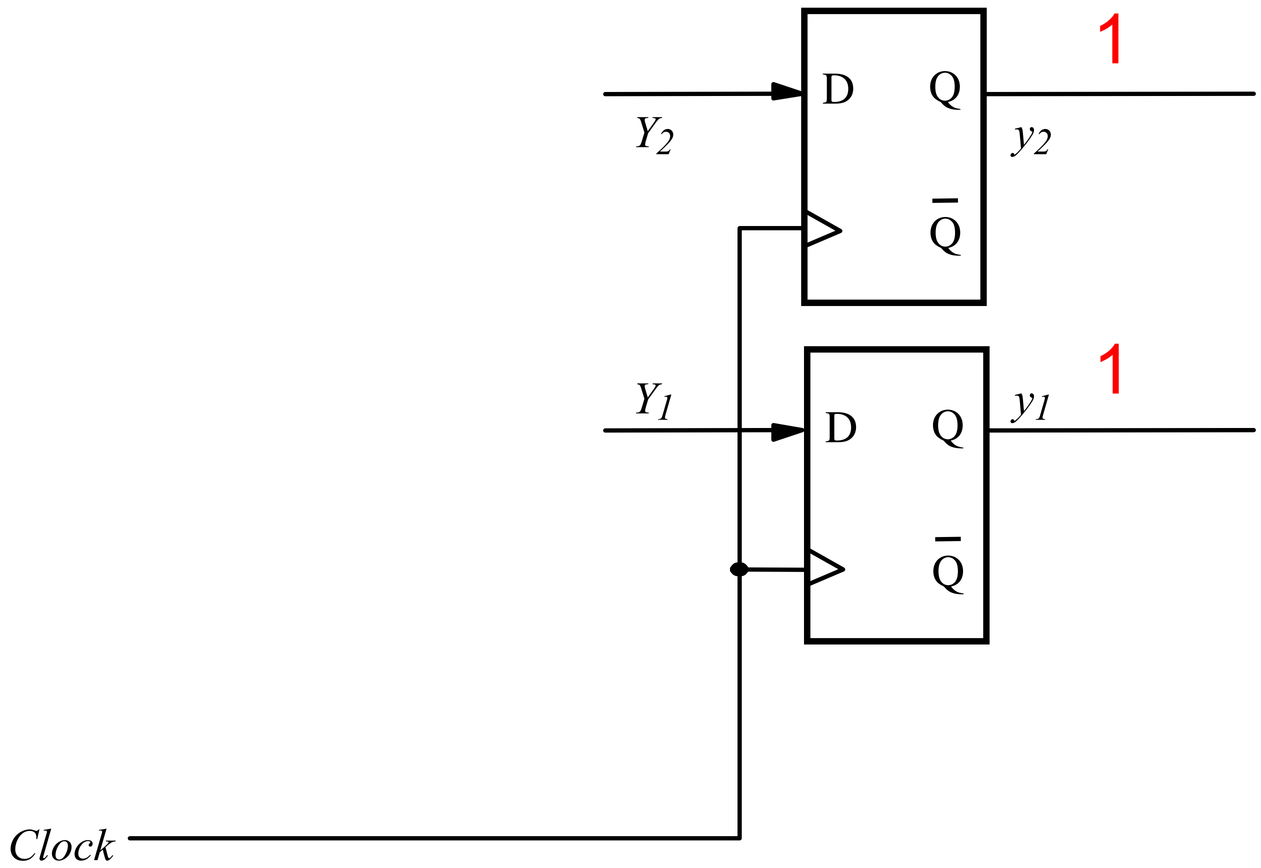
This flip-flop output pattern represents state C.

State = $y_2\ y_1$

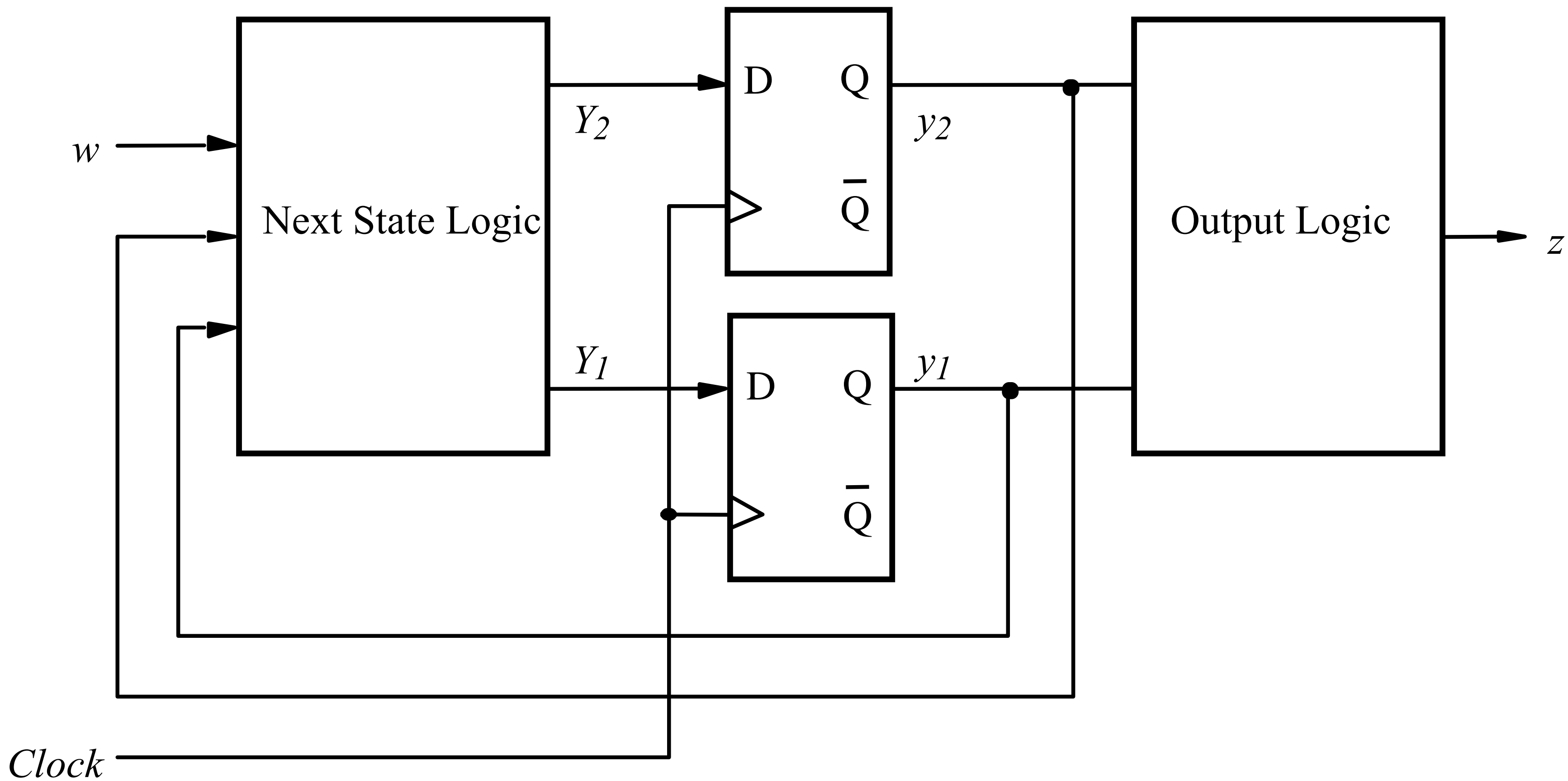


What does this flip-flop output pattern represent?

State = $y_2\ y_1$



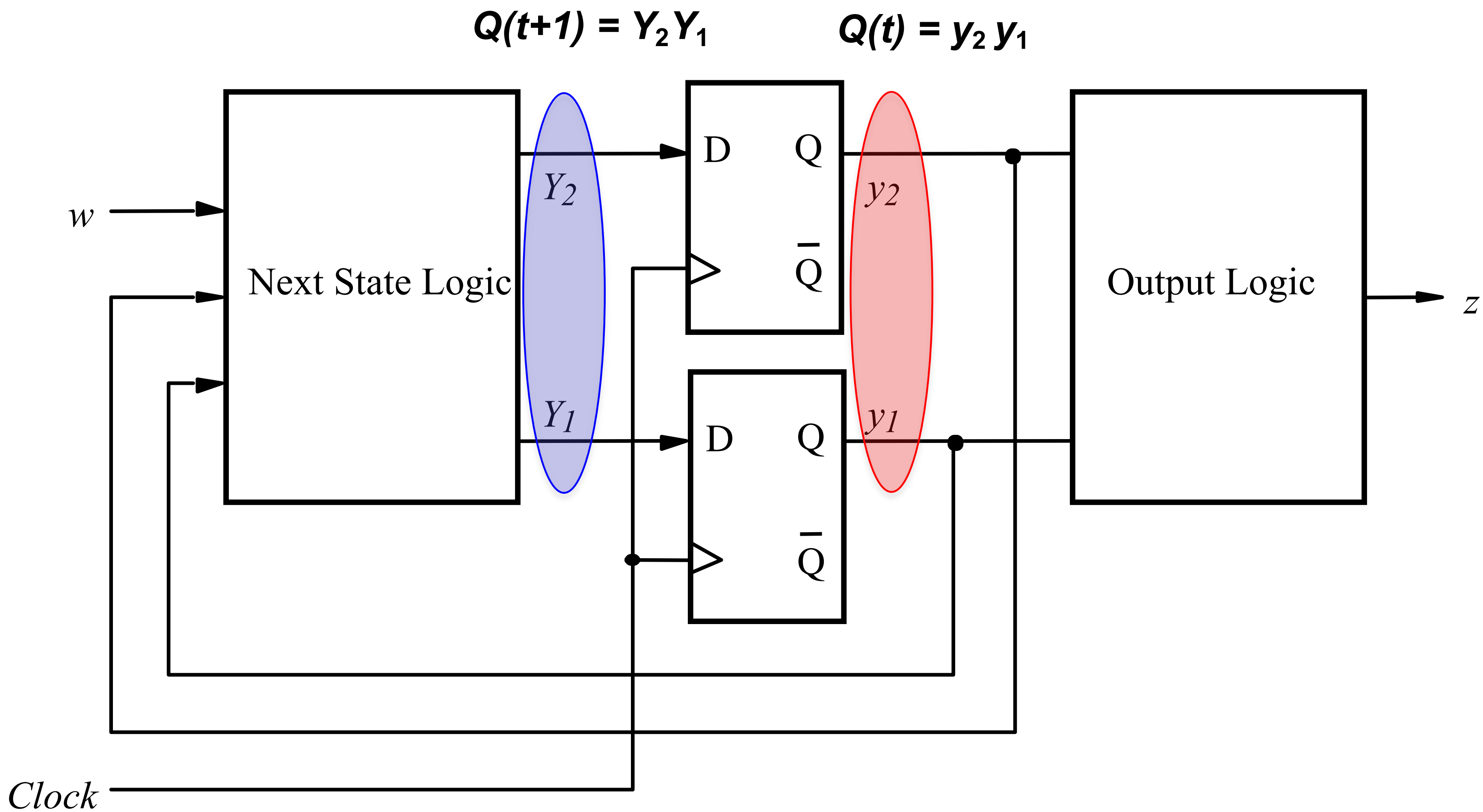
This would be state D, but we don't have one in this example. So this is an impossible state.



We will call y_1 and y_2 the *present state variables*.

We will call Y_1 and Y_2 the *next state variables*.

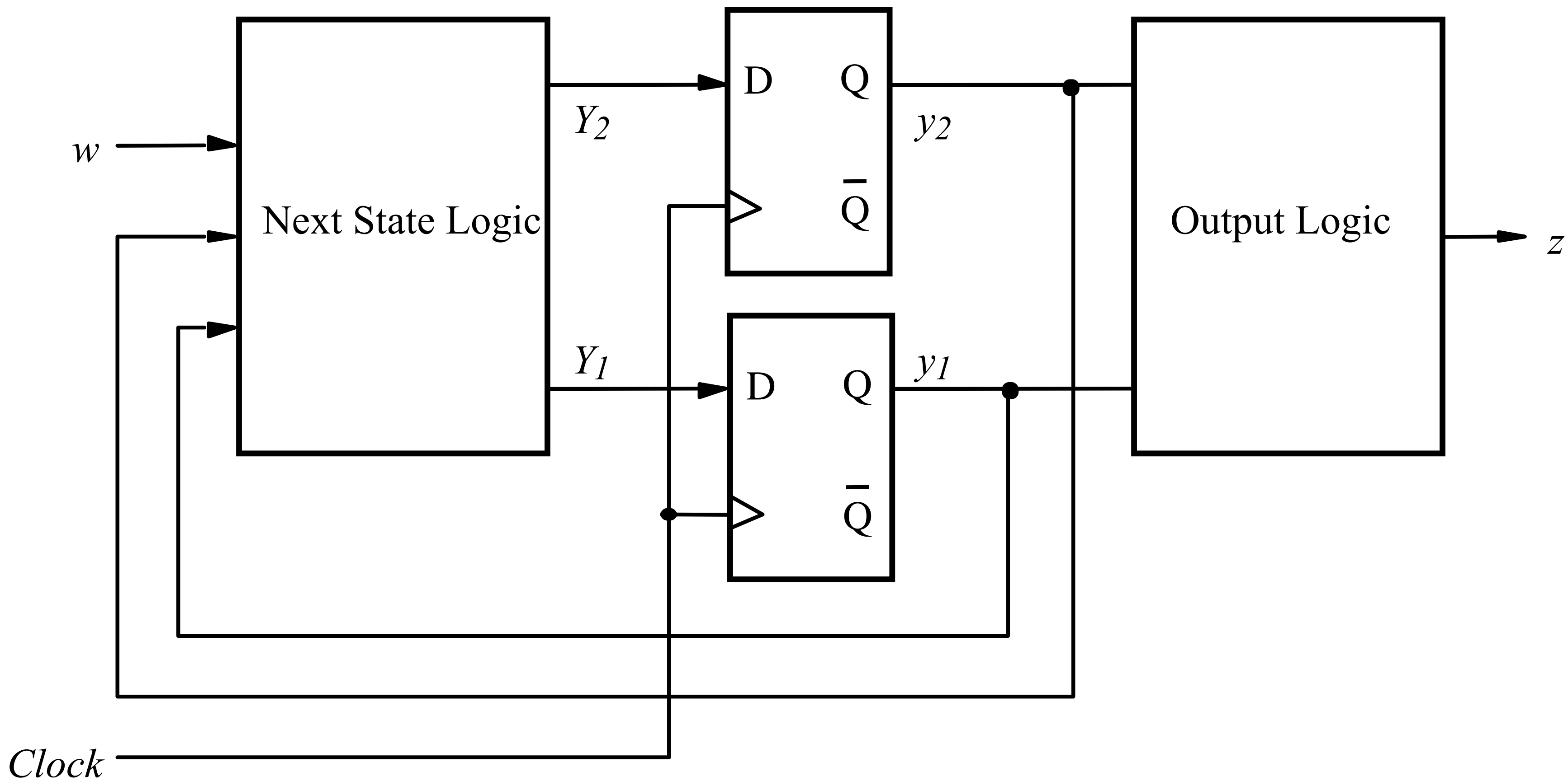
[Figure 6.5 from the textbook]



We will call y_1 and y_2 the *present state variables*.

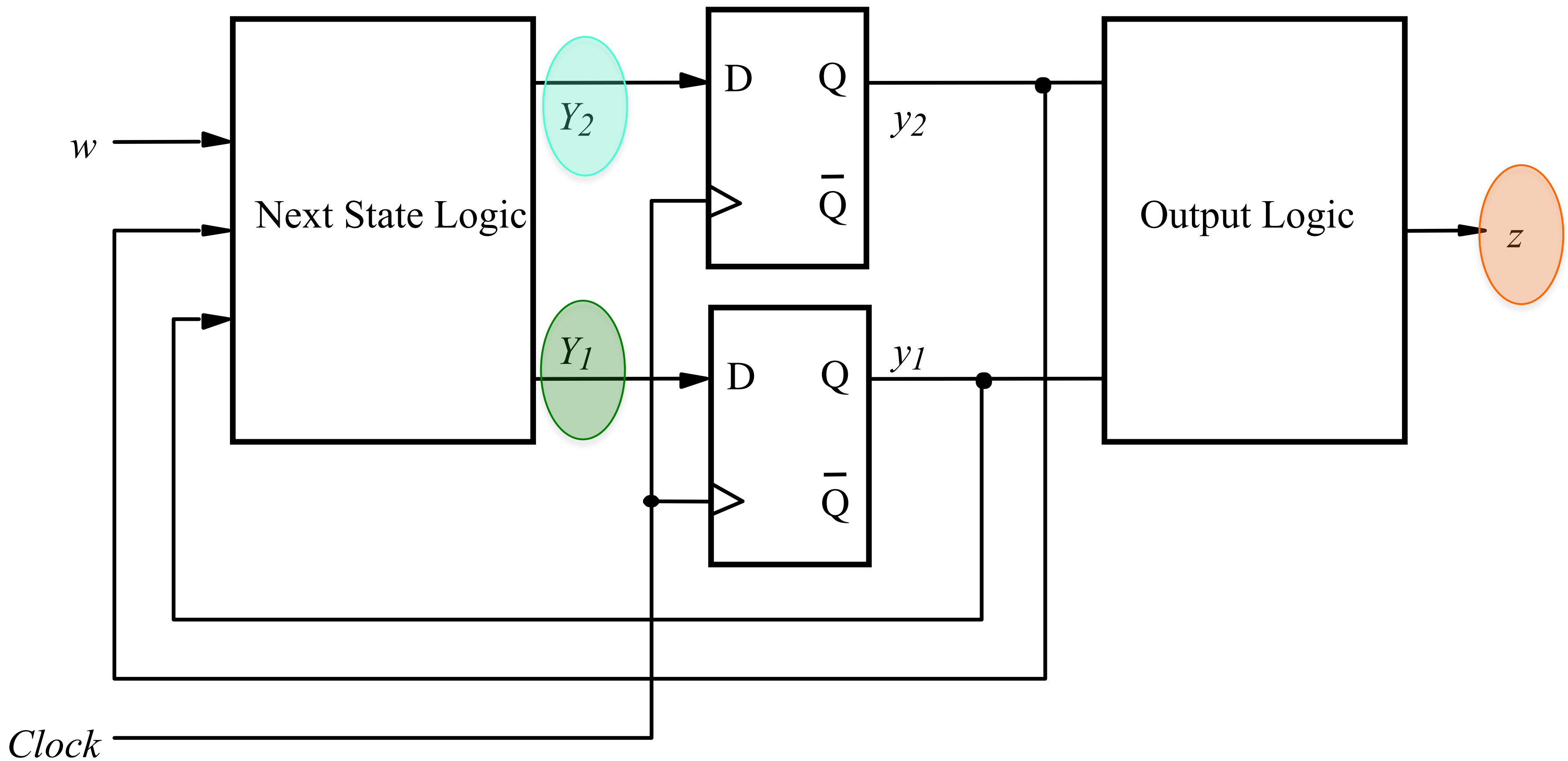
We will call Y_1 and Y_2 the *next state variables*.

[Figure 6.5 from the textbook]



We need to find logic expressions for
$Y_1(w, y_1, y_2)$, $Y_2(w, y_1, y_2)$, and $z(y_1, y_2)$.

[Figure 6.5 from the textbook]



We need to find logic expressions for $Y_1(w, y_1, y_2)$, $Y_2(w, y_1, y_2)$, and $z(y_1, y_2)$.

[Figure 6.5 from the textbook]

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

Suppose we encoded our states in the same order in which they were labeled:

A ~ 00
B ~ 01
C ~ 10

[Figure 6.4 from the textbook]

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	Next state $w = 0$	Next state $w = 1$	Output z
A	00			
B	01			
C	10			
	11			

The finite state machine will never reach a state encoded as 11.

[Figure 6.6 from the textbook]

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

We arbitrarily chose these as our state encodings. We could have used others.

[Figure 6.6 from the textbook]

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

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

w	y_2	y_1	Y_2	Y_1
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

y_2	y_1	z
0	0	
0	1	
1	0	
1	1	

[Figure 6.6 from the textbook]

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

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

w	y_2	y_1	Y_2	Y_1
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

y_2	y_1	z
0	0	0
0	1	0
1	0	1
1	1	d

[Figure 6.6 from the textbook]

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

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

y_2	y_1	z
0	0	0
0	1	0
1	0	1
1	1	d

[Figure 6.6 from the textbook]

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

[Figure 6.6 from the textbook]

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	d	
1	0	0		
1	0	1		
1	1	0		
1	1	1		

y_2	y_1	z
0	0	0
0	1	0
1	0	1
1	1	d

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

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

w	y_2	y_1	Y_2	Y_1
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	d	
1	0	0	0	
1	0	1	1	
1	1	0	1	
1	1	1	d	

y_2	y_1	z
0	0	0
0	1	0
1	0	1
1	1	d

[Figure 6.6 from the textbook]

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

[Figure 6.6 from the textbook]

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	d	
1	0	0	0	
1	0	1	1	
1	1	0	1	
1	1	1	d	

y_2	y_1	z
0	0	0
0	1	0
1	0	1
1	1	d

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

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

w	y_2	y_1	Y_2	Y_1
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	d	d
1	0	0	0	
1	0	1	1	
1	1	0	1	
1	1	1	d	

y_2	y_1	z
0	0	0
0	1	0
1	0	1
1	1	d

[Figure 6.6 from the textbook]

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

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	d	d
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	d	d

y_2	y_1	z
0	0	0
0	1	0
1	0	1
1	1	d

[Figure 6.6 from the textbook]

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

$Q(t) = y_2 y_1$ and $Q(t+1) = Y_2 Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	d	d
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	d	d

y_2	y_1	z
0	0	0
0	1	0
1	0	1
1	1	d

[Figure 6.6 from the textbook]

Note that the textbook draws these K-Maps differently from all previous K-maps (the most significant bit indexes the rows).

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

Y_1

w \ y_2y_1	00	01	11	10
0	0	0	d	0
1	1	0	d	0

Y_2

w \ y_2y_1	00	01	11	10
0	0	0	d	0
1	0	1	d	1

z

y_2 \ y_1	0	1
0	0	0
1	1	d

w	y_2	y_1	Y_2	Y_1
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	d	d
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	d	d

y_2	y_1	z
0	0	0
0	1	0
1	0	1
1	1	d

Don’t care conditions simplify the combinatorial logic

Y_1

w \ y_2y_1	00	01	11	10
0	0	0	d	0
1	1	0	d	0

Y_2

w \ y_2y_1	00	01	11	10
0	0	0	d	0
1	0	1	d	1

z

y_2 \ y_1	0	1
0	0	0
1	1	d

Ignoring don't cares

$$Y_1 = w\bar{y}_1\bar{y}_2$$

$$Y_2 = wy_1\bar{y}_2 + \bar{w}y_1y_2$$

$$z = \bar{y}_1y_2$$

Using don't cares

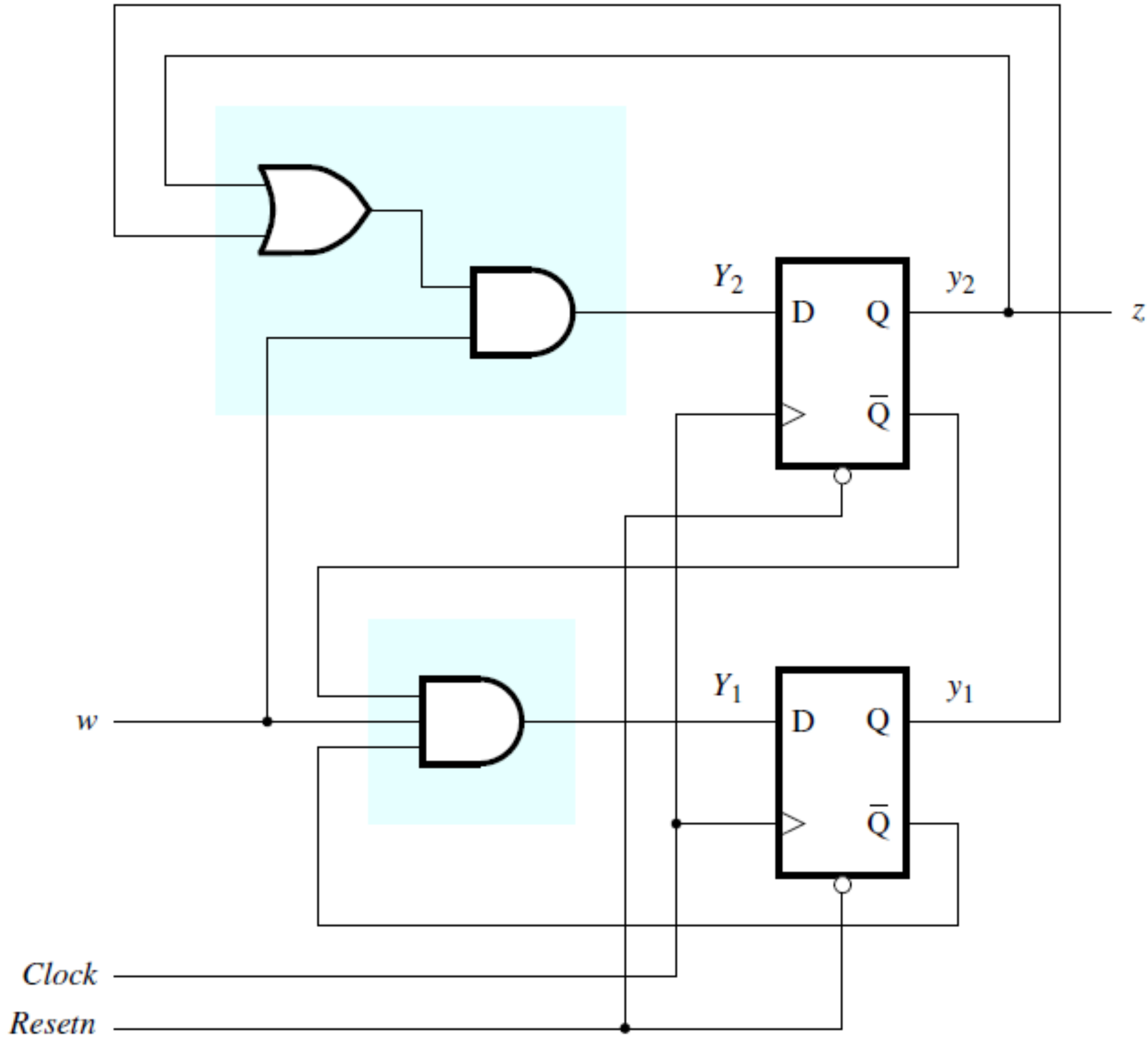
$$Y_1 = w\bar{y}_1\bar{y}_2$$

$$Y_2 = wy_1 + wy_2$$
$$= w(y_1 + y_2)$$

$$z = y_2$$

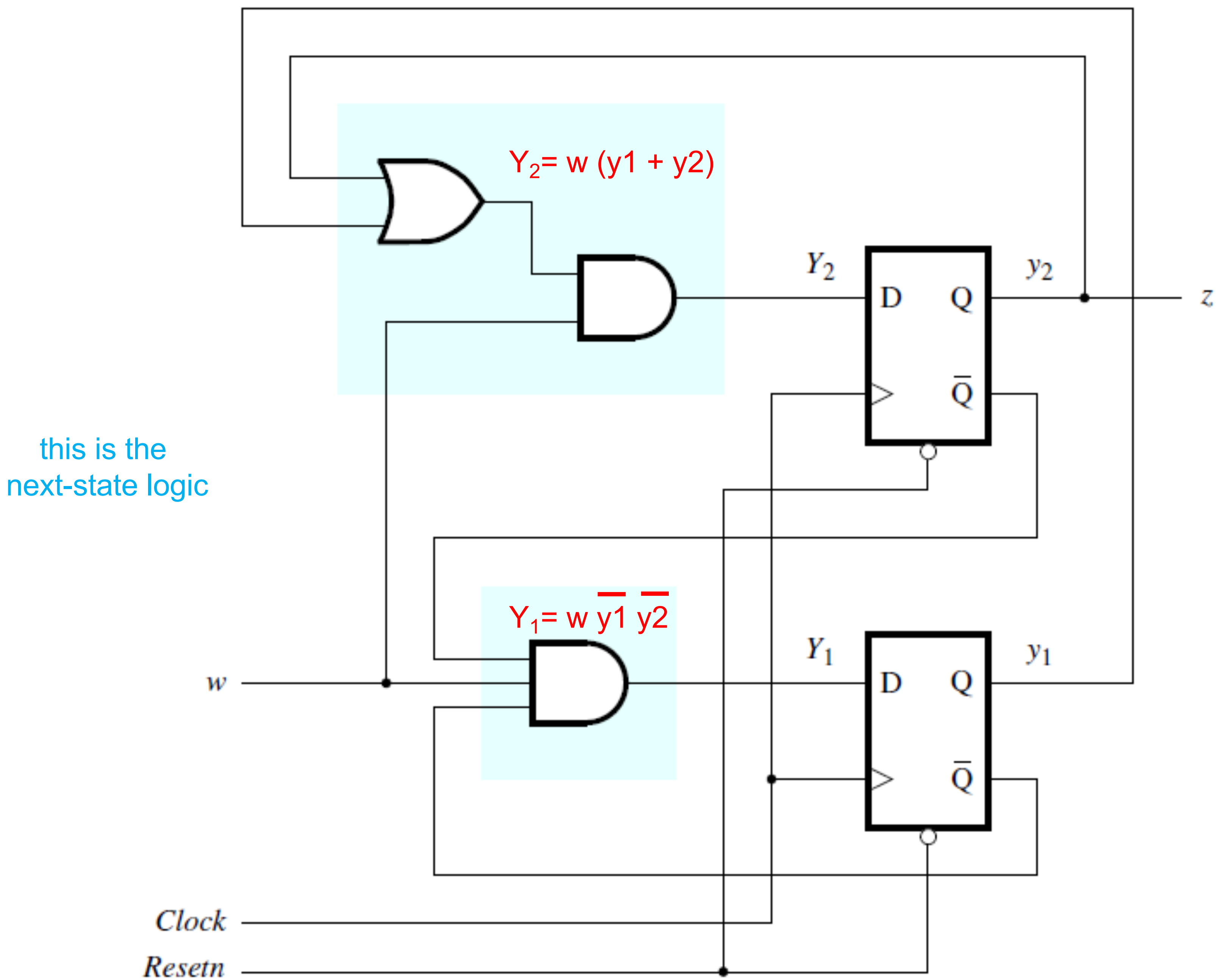
[Figure 6.7 from the textbook]

Draw the Circuit Diagram



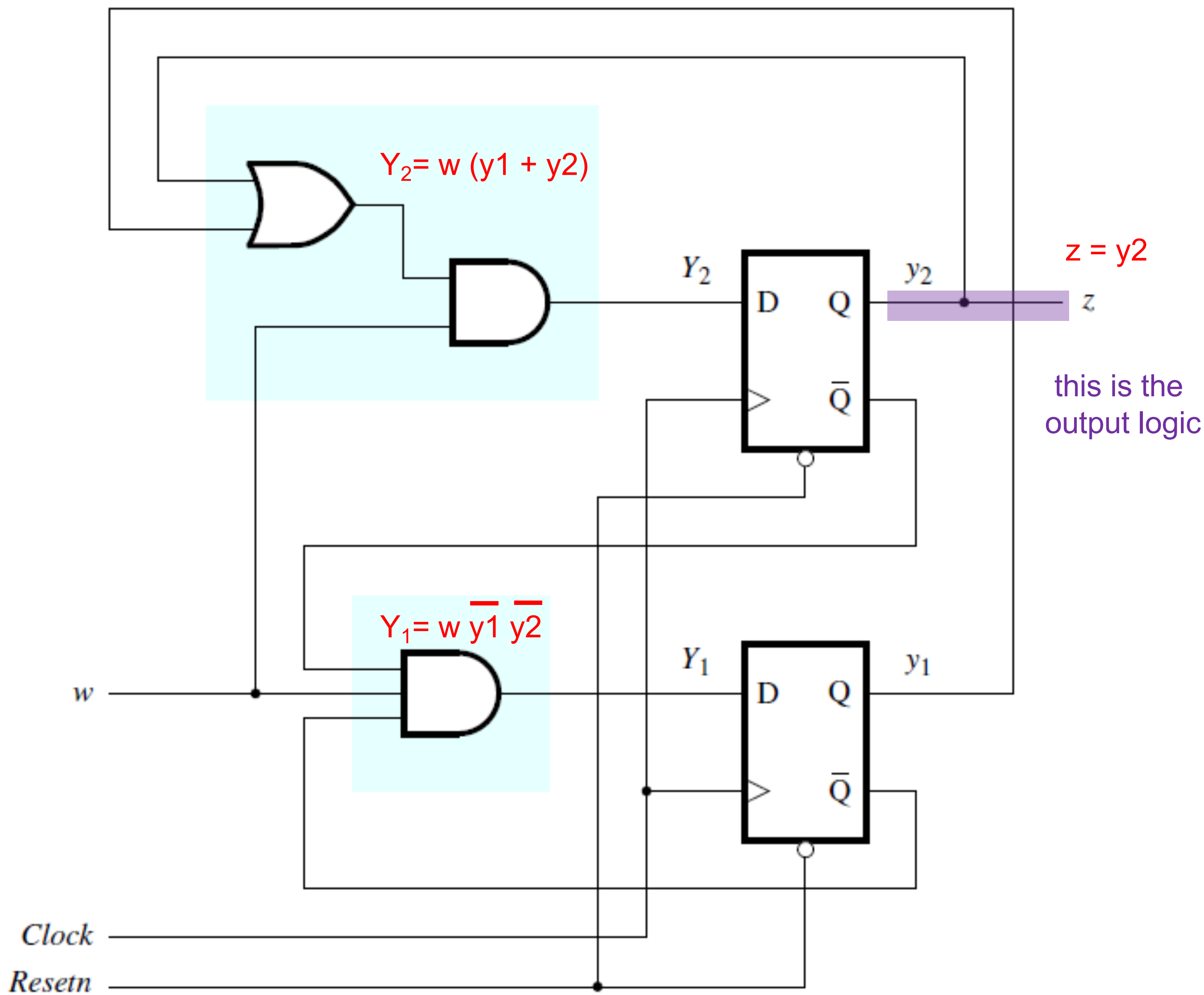
[Figure 6.8 from the textbook]

Draw the Circuit Diagram



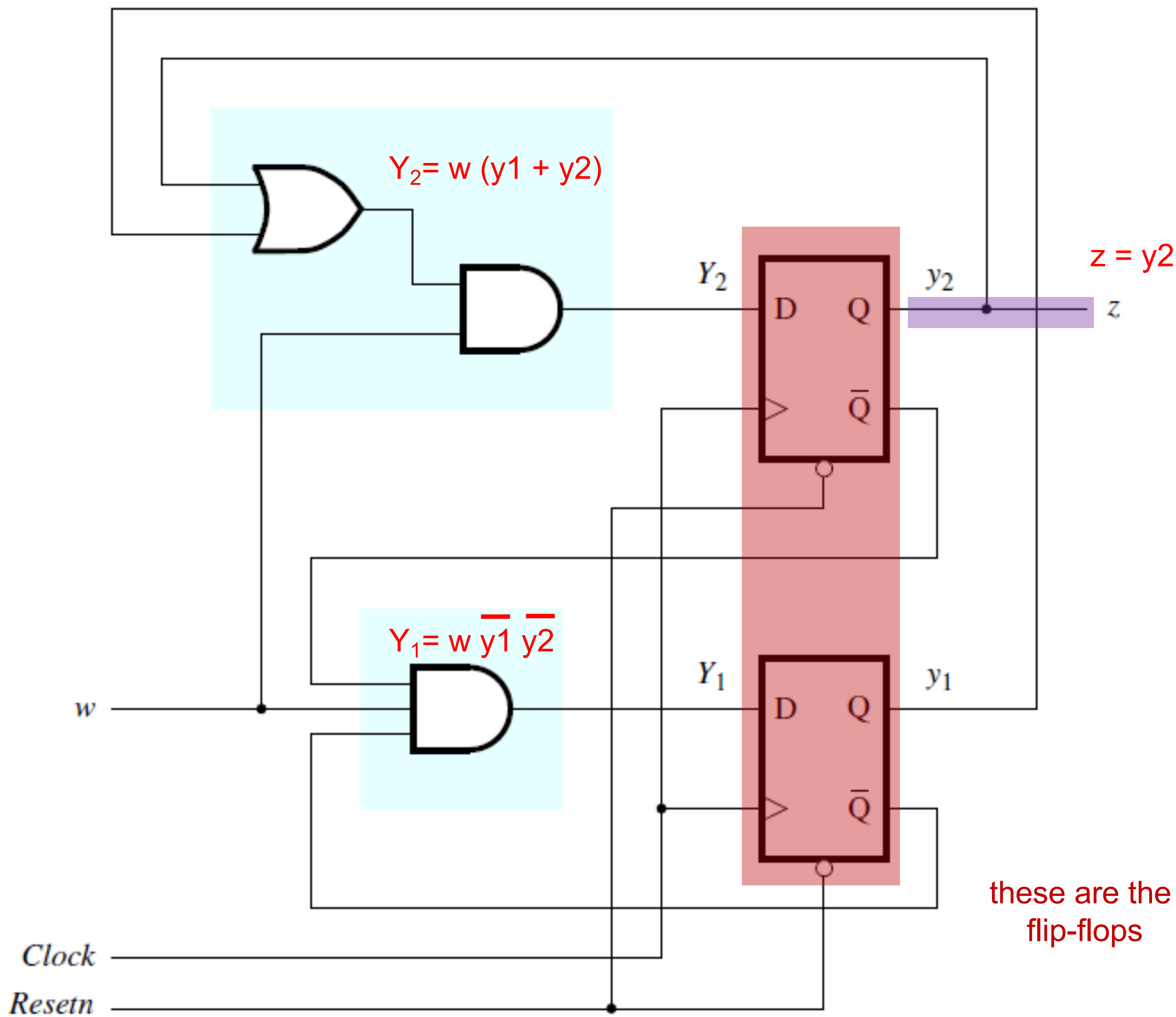
[Figure 6.8 from the textbook]

Draw the Circuit Diagram



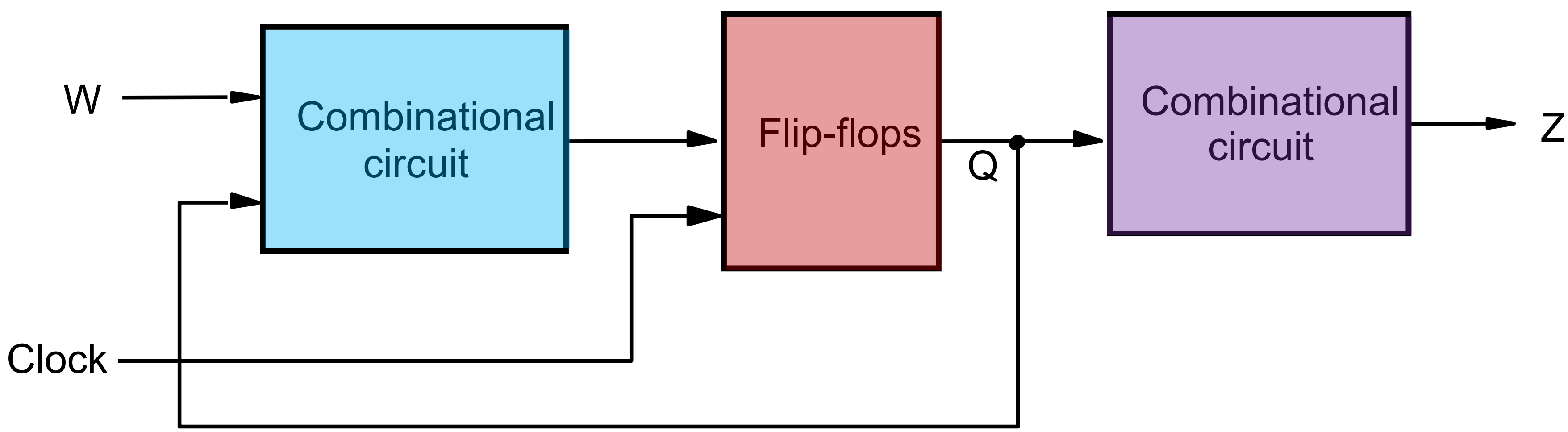
[Figure 6.8 from the textbook]

Draw the Circuit Diagram

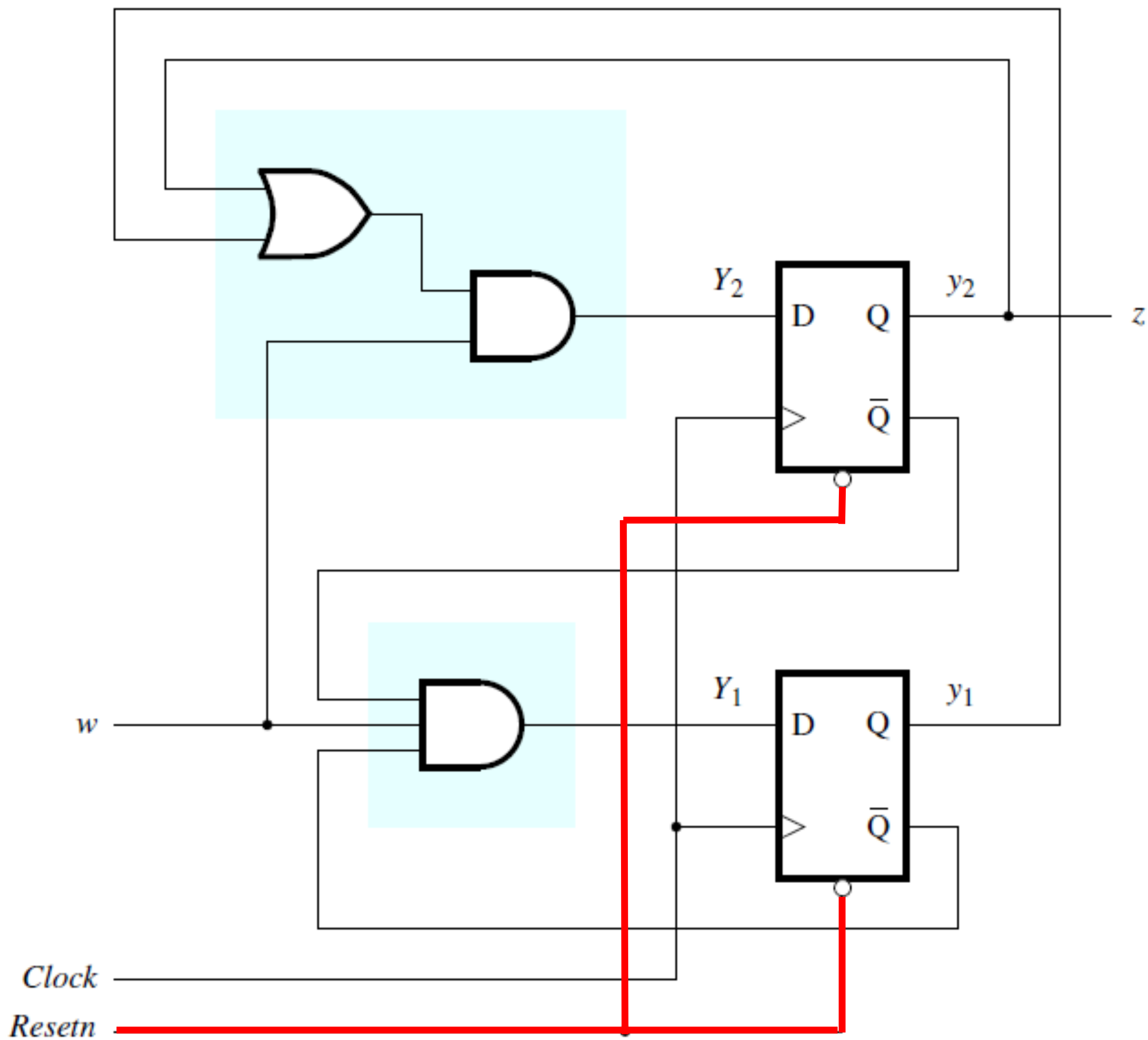


[Figure 6.8 from the textbook]

Moore Type

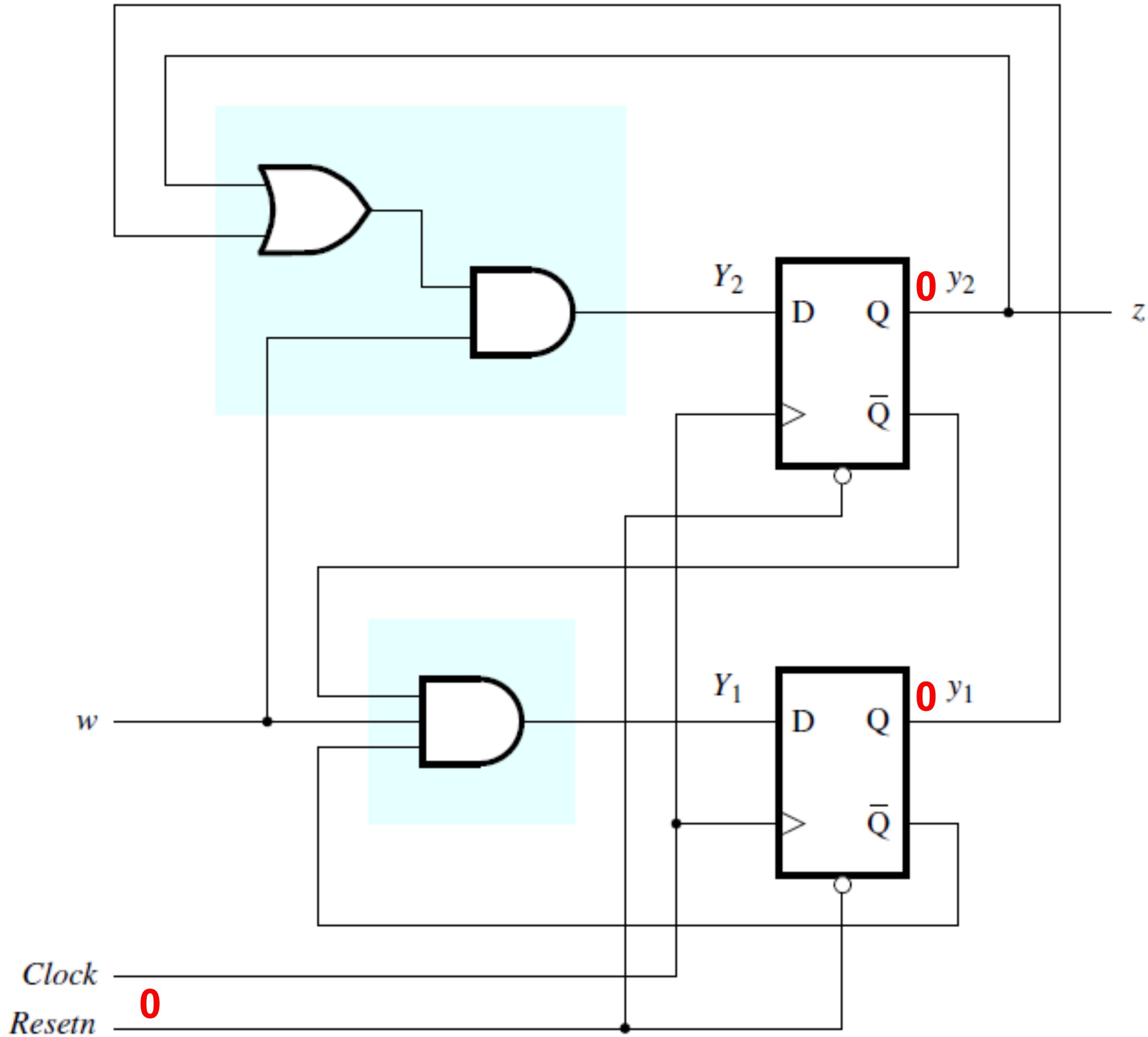

W
Combinational
circuit
Flip-flops
Q
Combinational
circuit
Z
Clock

Don’t Forget to Add the Reset Line



[Figure 6.8 from the textbook]

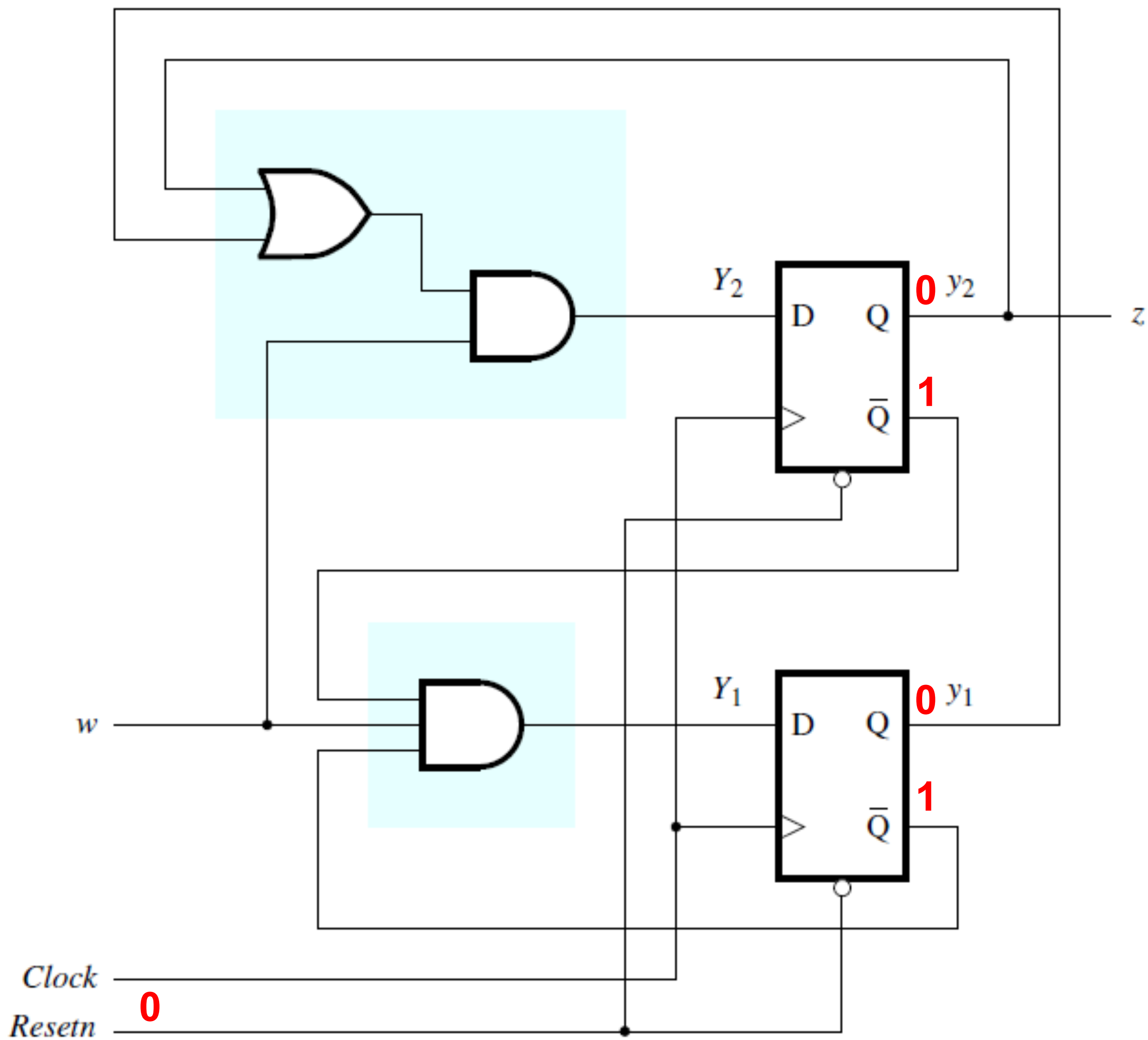
State A=00



When the reset signal is equal to zero it puts the machine back to its start state, which is state *00* (or A) in this case.

[Figure 6.8 from the textbook]

State A=00

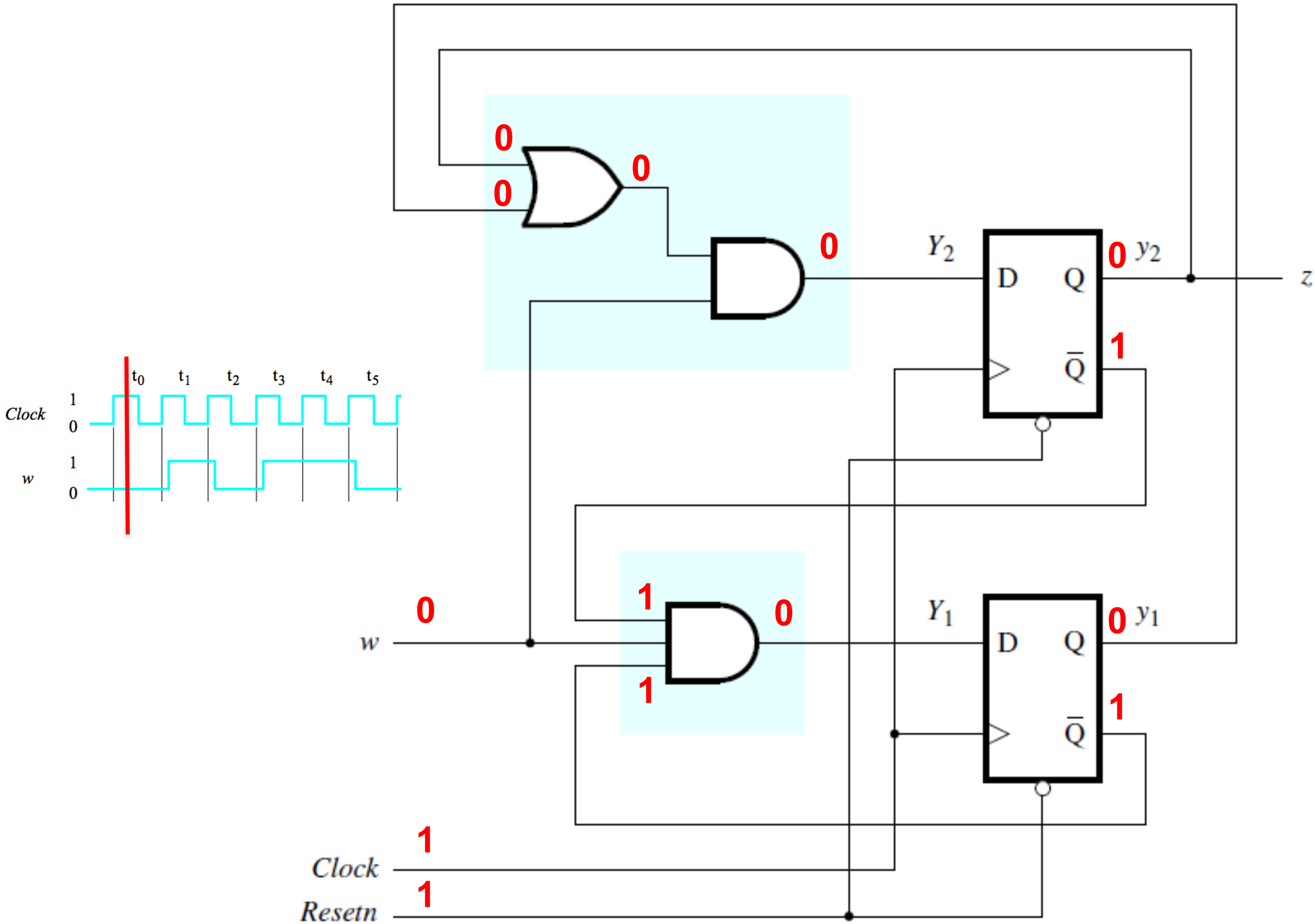


When the reset signal is equal to zero it puts the machine back to its start state, which is state *00* (or A) in this case.

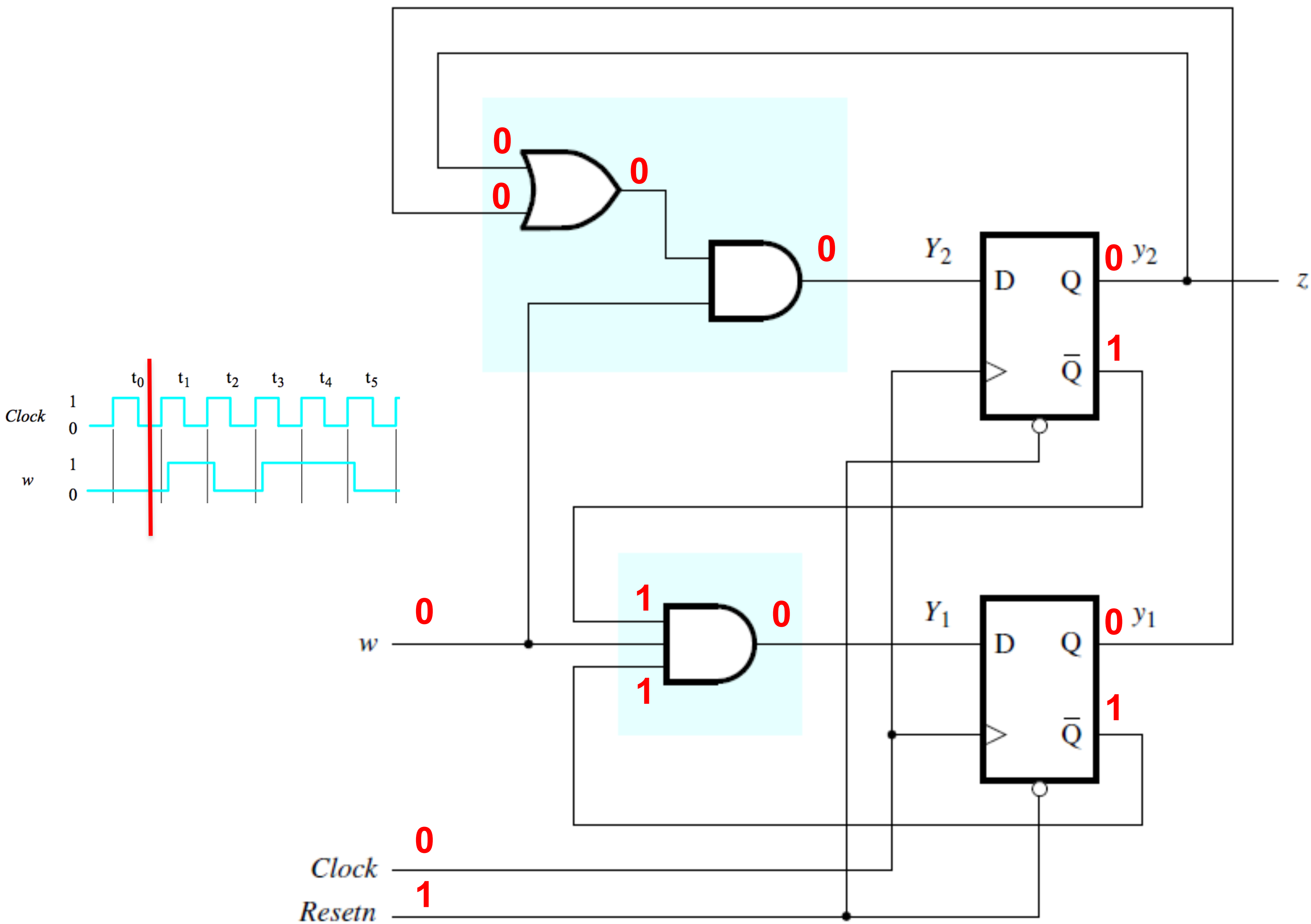
[Figure 6.8 from the textbook]

State = $y_2\ y_1$

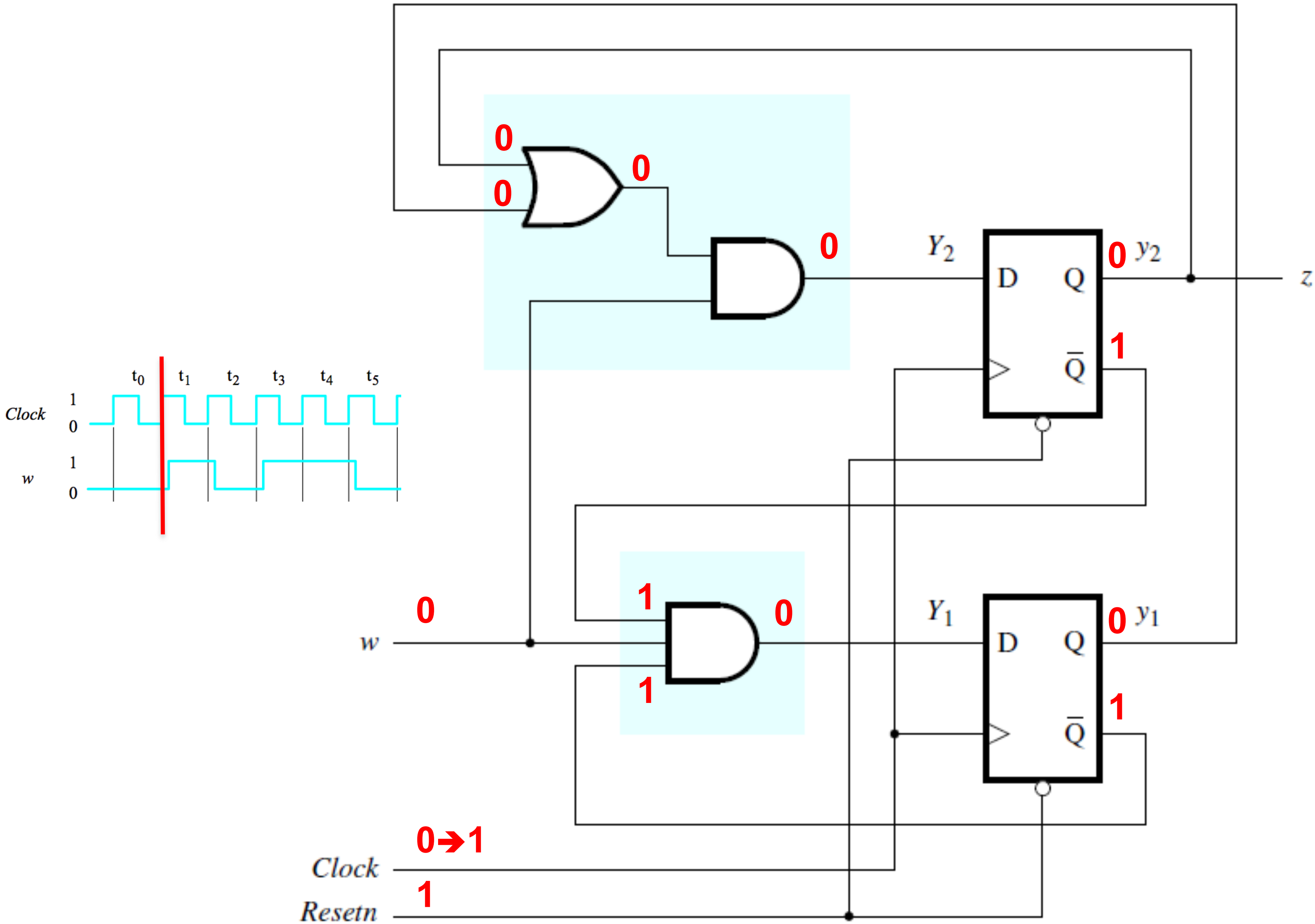
State A=00



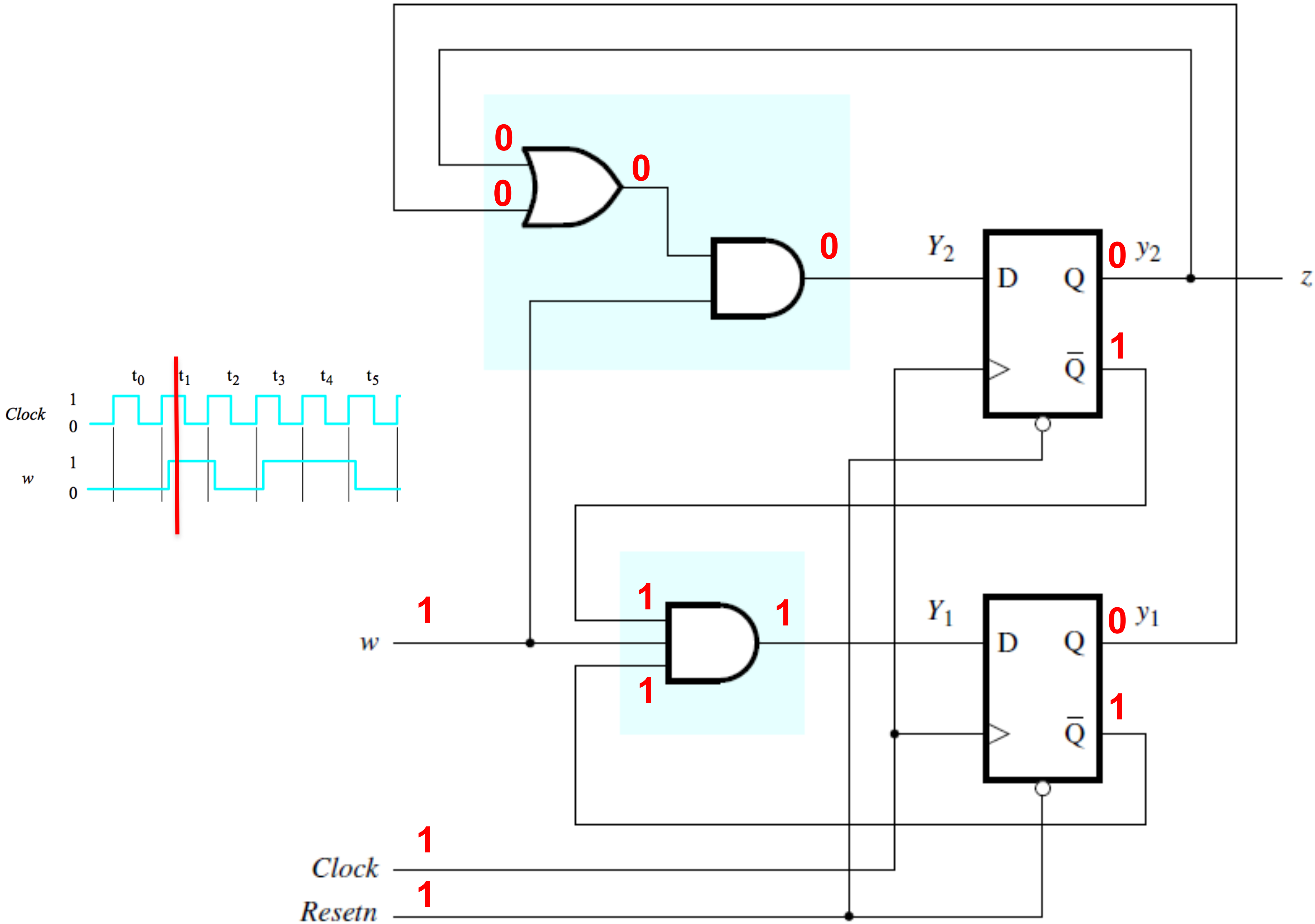
State A=00



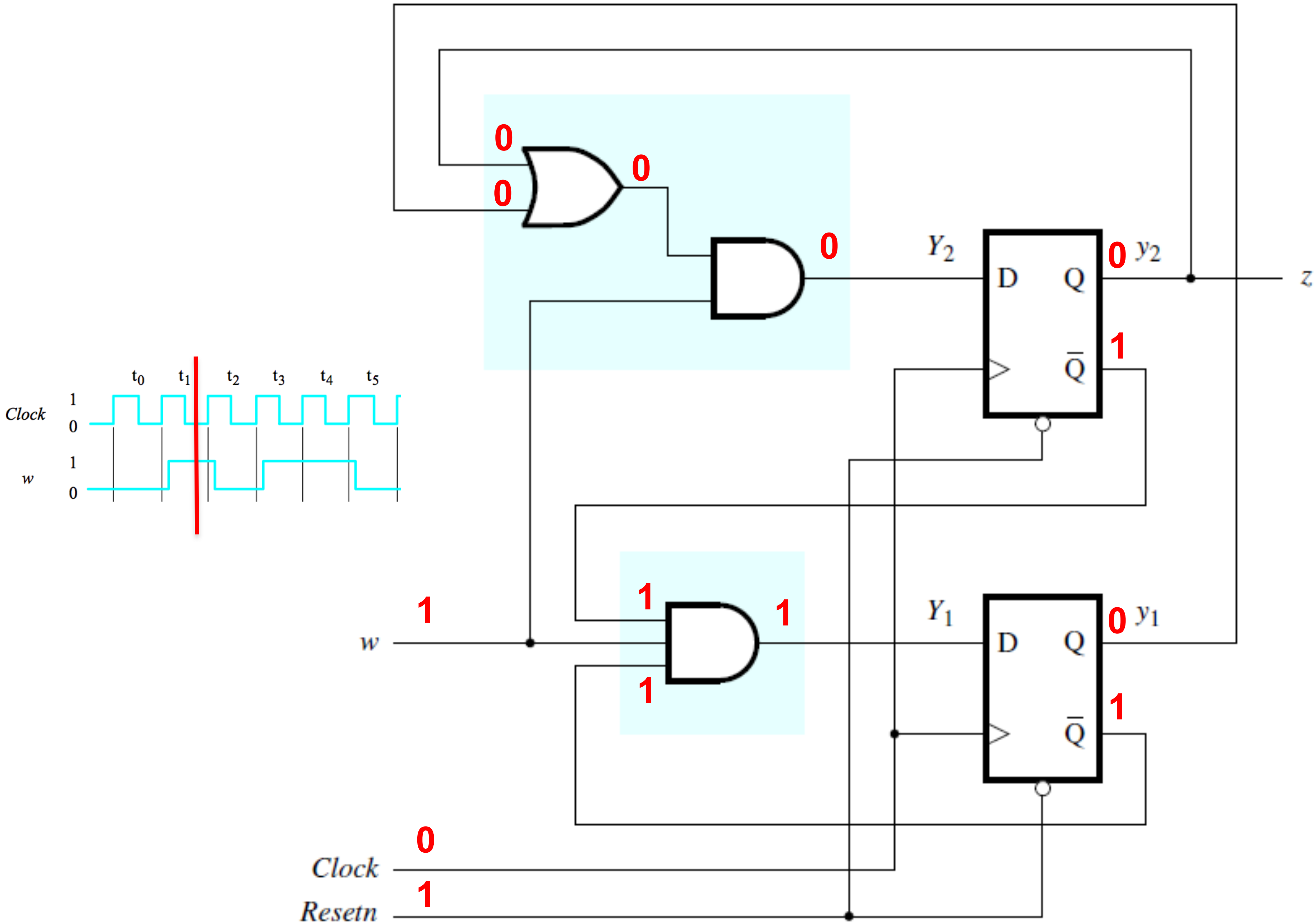
State A=00


t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0
0
0
0
0
Y2
D
Q
0
y2
Q̄
1
z
w
0
1
1
0
Y1
D
Q
0
y1
Q̄
1
Clock
0➔1
Resetn
1

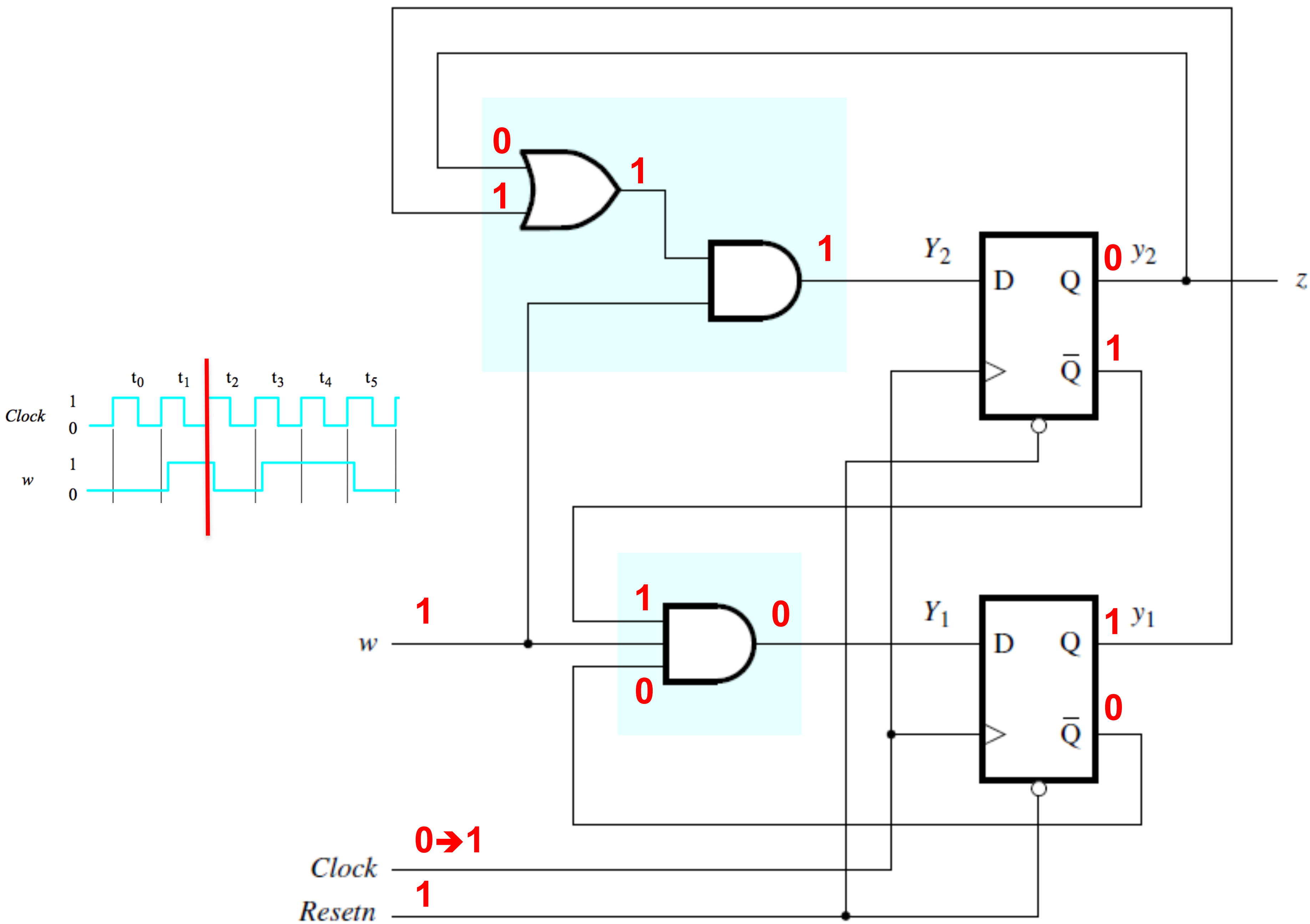
State A=00



State A=00

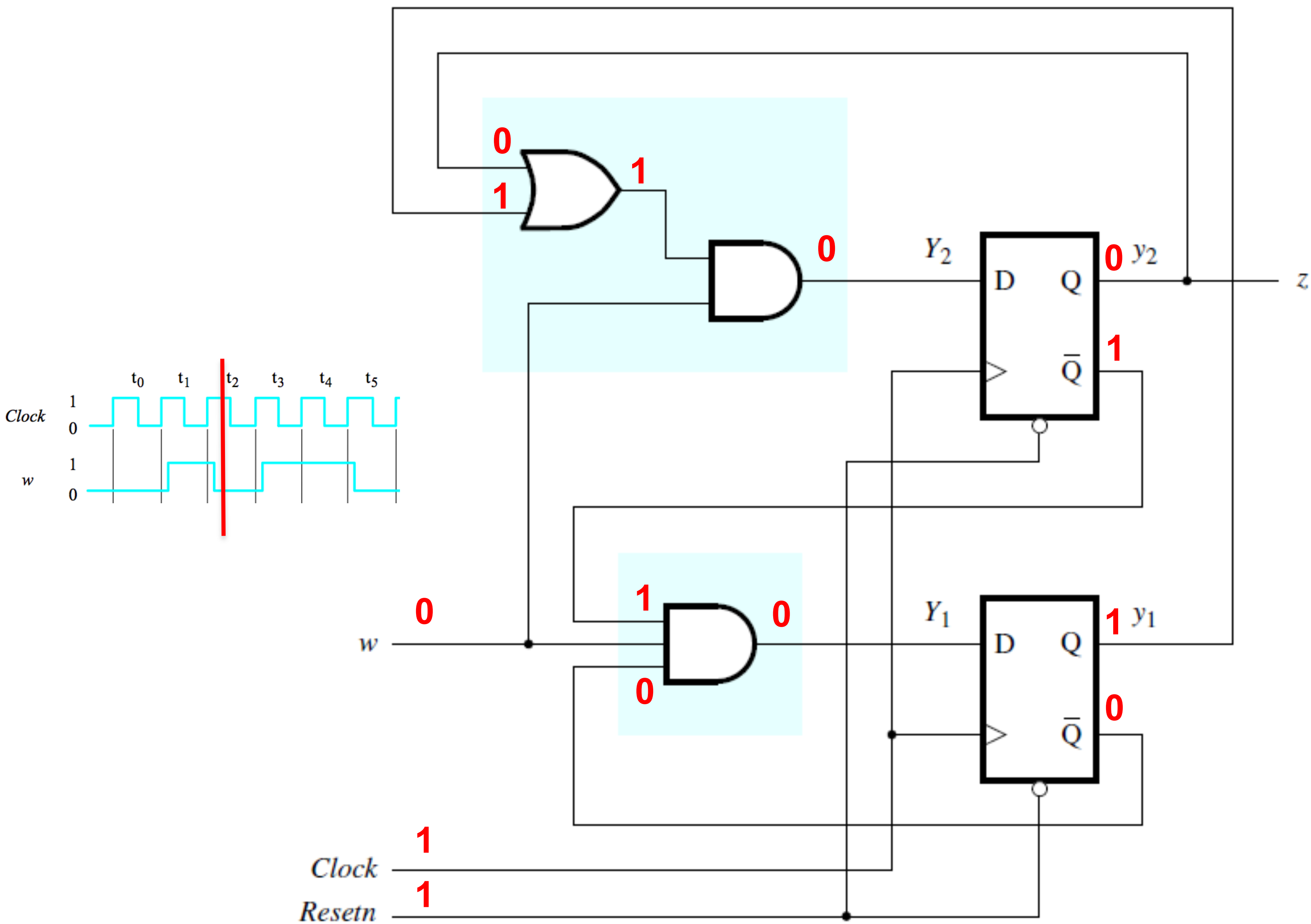

t0 t1 t2 t3 t4 t5
Clock
1
0
w
1
0
0
0
0
0
Y2
D Q
0 y2
z
1
Q̄
w
1
1
1
1
Y1
D Q
0 y1
1
Q̄
0
Clock
1
Resetn

State B=01



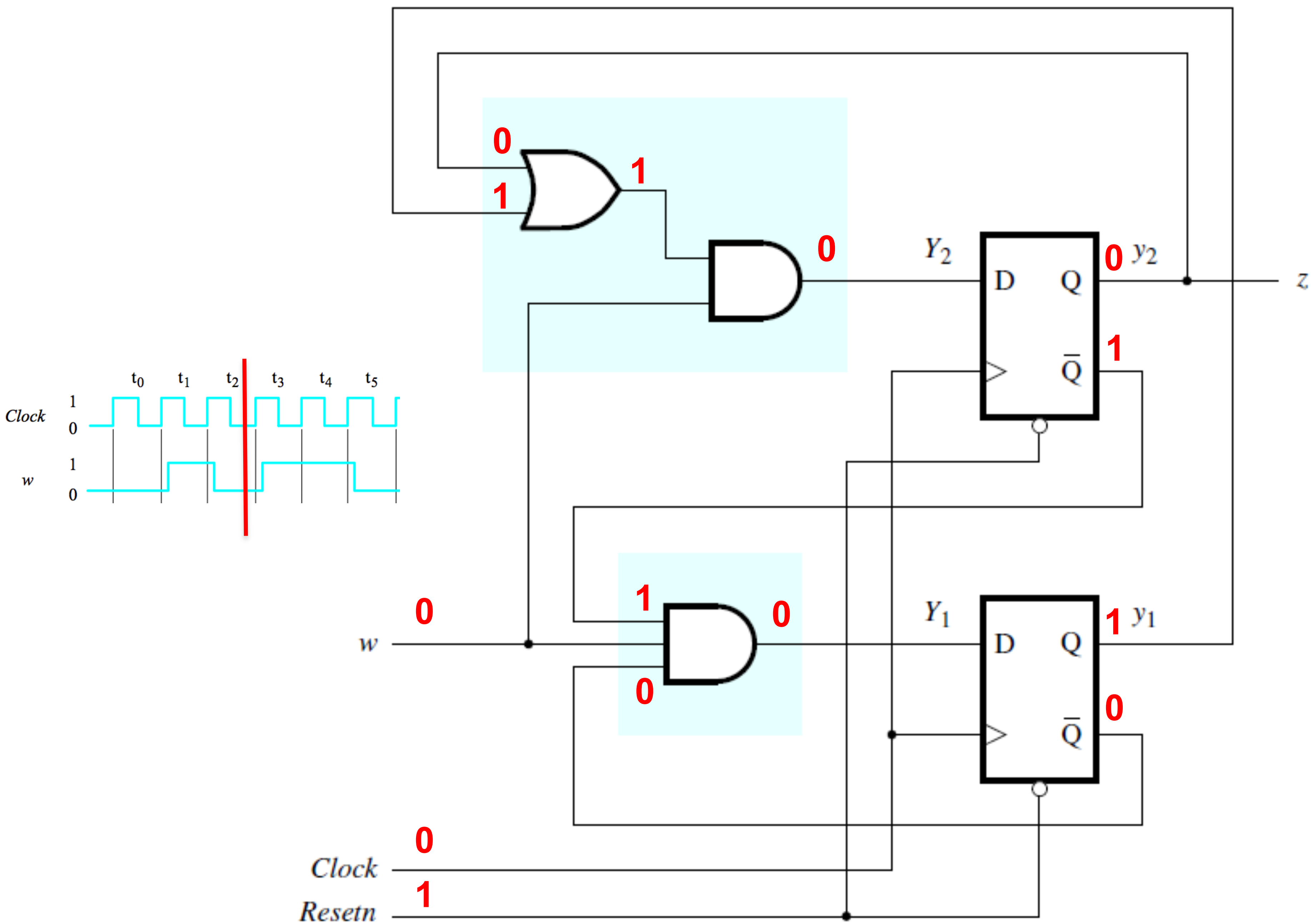
t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0
0
1
1
1
Y2
D
Q
0
y2
z
1
Q̄
1
1
0
w
1
0
Y1
D
Q
1
y1
0
Q̄
0→1
Clock
1
Resetn

State B=01



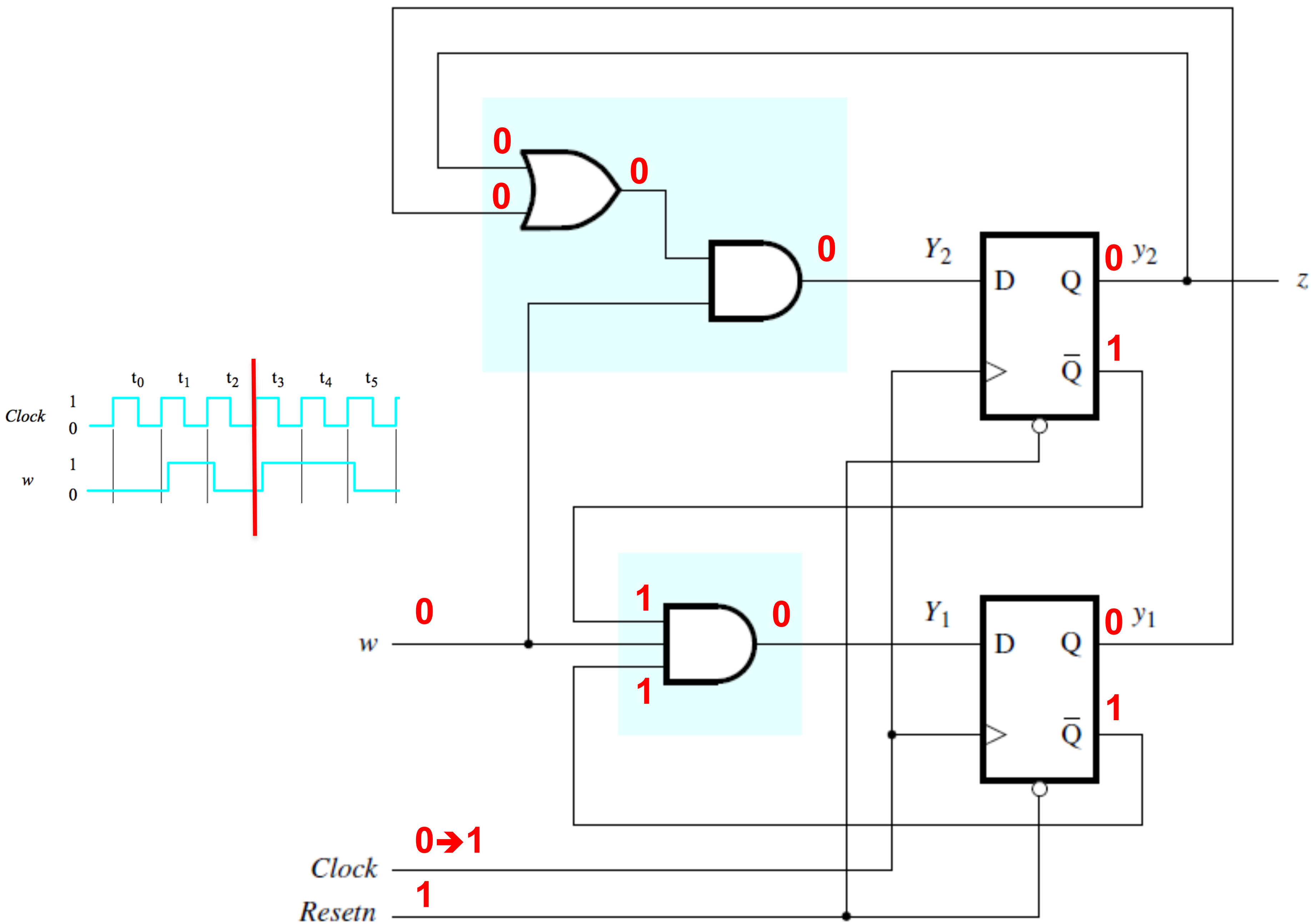
t_0 t_1 t_2 t_3 t_4 t_5
Clock
w
0
1
1
0
0
Y_2
D
Q
$\bar{Q}$
0 y_2
1
z
1
0
0
Y_1
D
Q
$\bar{Q}$
1 y_1
0
w
0
Clock
1
Resetn
1

State B=01

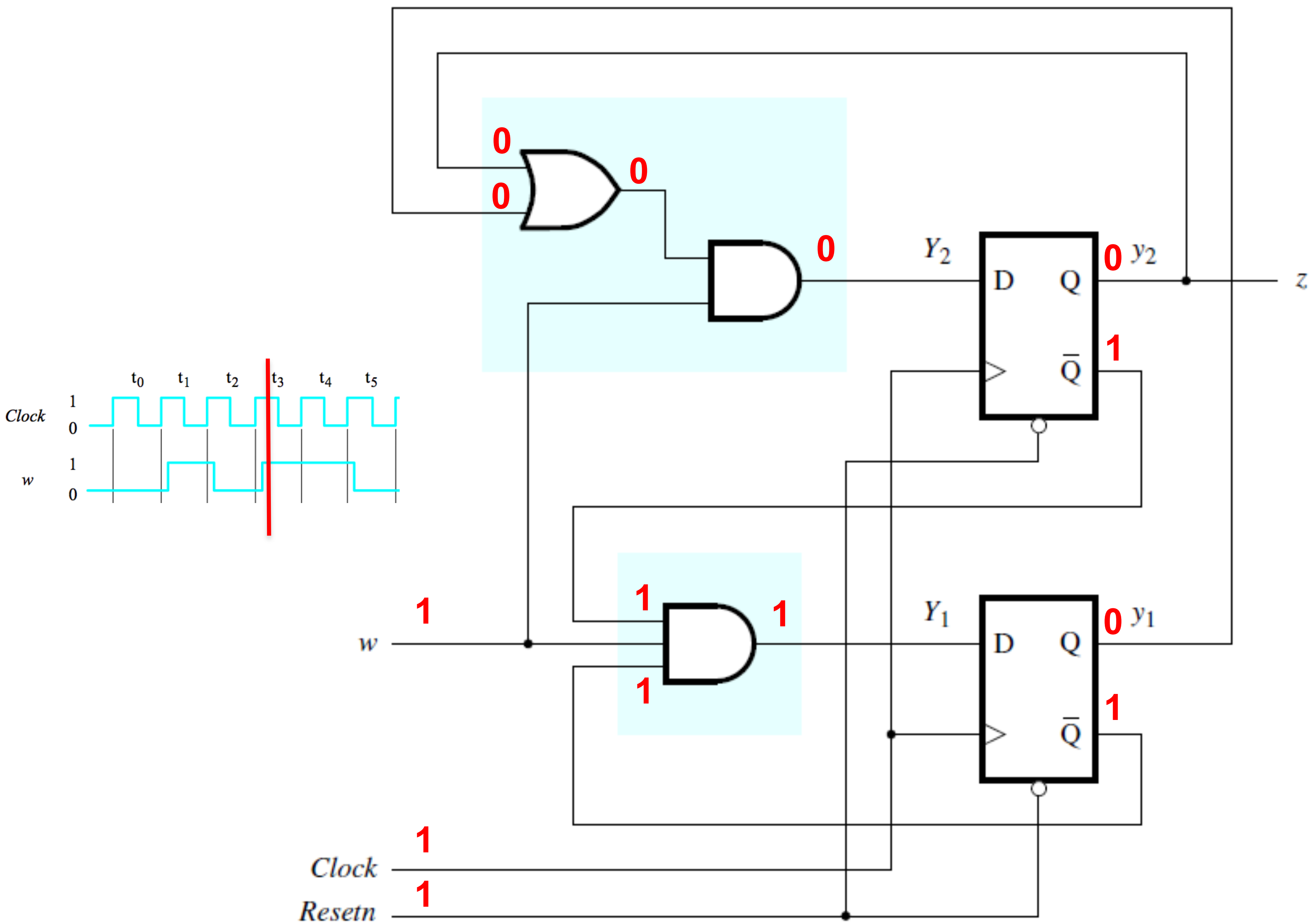


t0 t1 t2 t3 t4 t5
Clock
1
0
w
1
0
0
1
1
0
Y2
D Q
0 y2
z
1
1
0
1
0
0
Y1
D Q
1 y1
w
0
0
Clock
Resetn
1

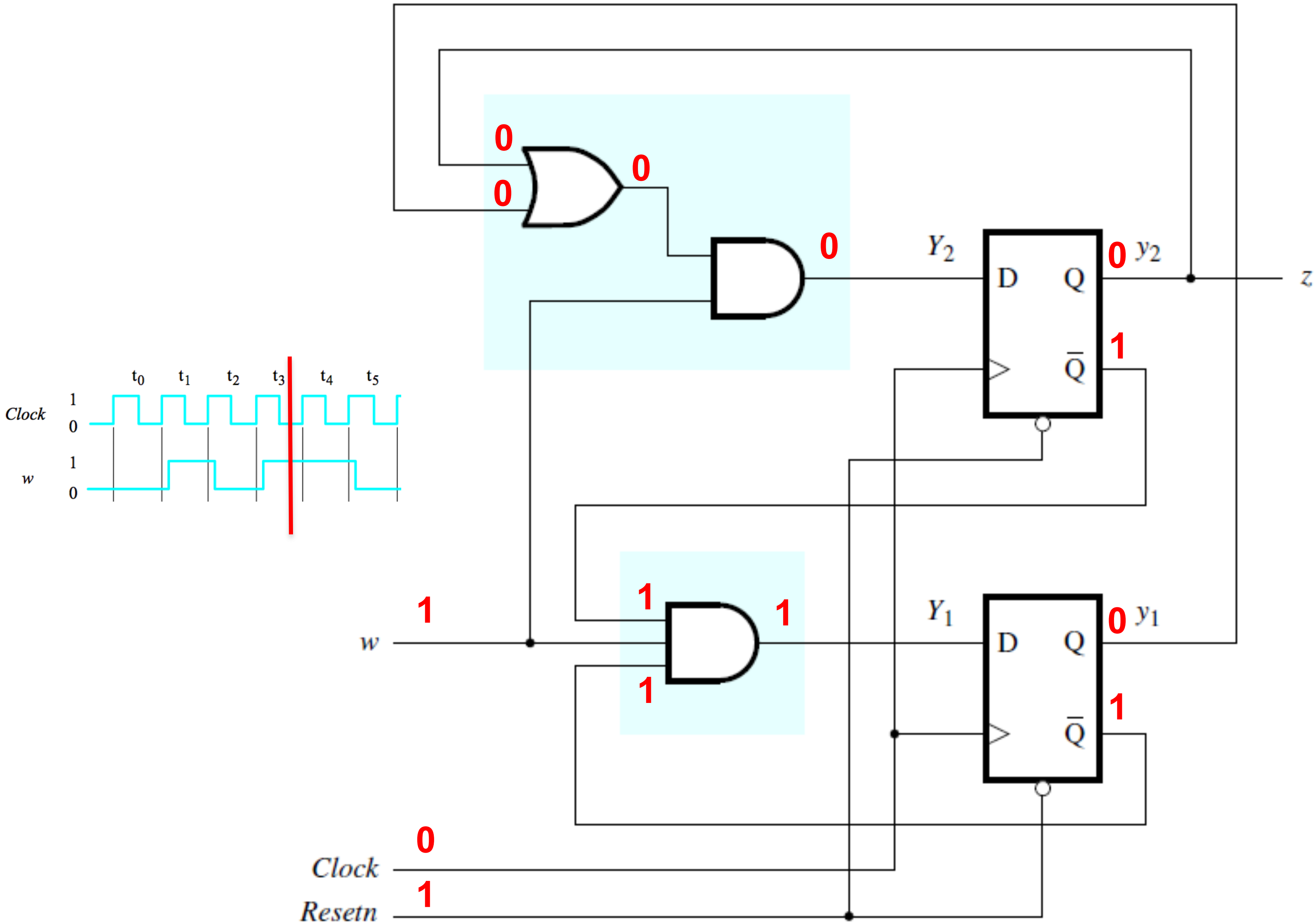
State A=00


t0 t1 t2 t3 t4 t5
Clock
1
0
w
1
0
0
0
0
0
Y2
D
Q
0 y2
z
1
Q̄
1
0
1
w
0
Y1
D
Q
0 y1
1
Q̄
0➔1
Clock
1
Resetn

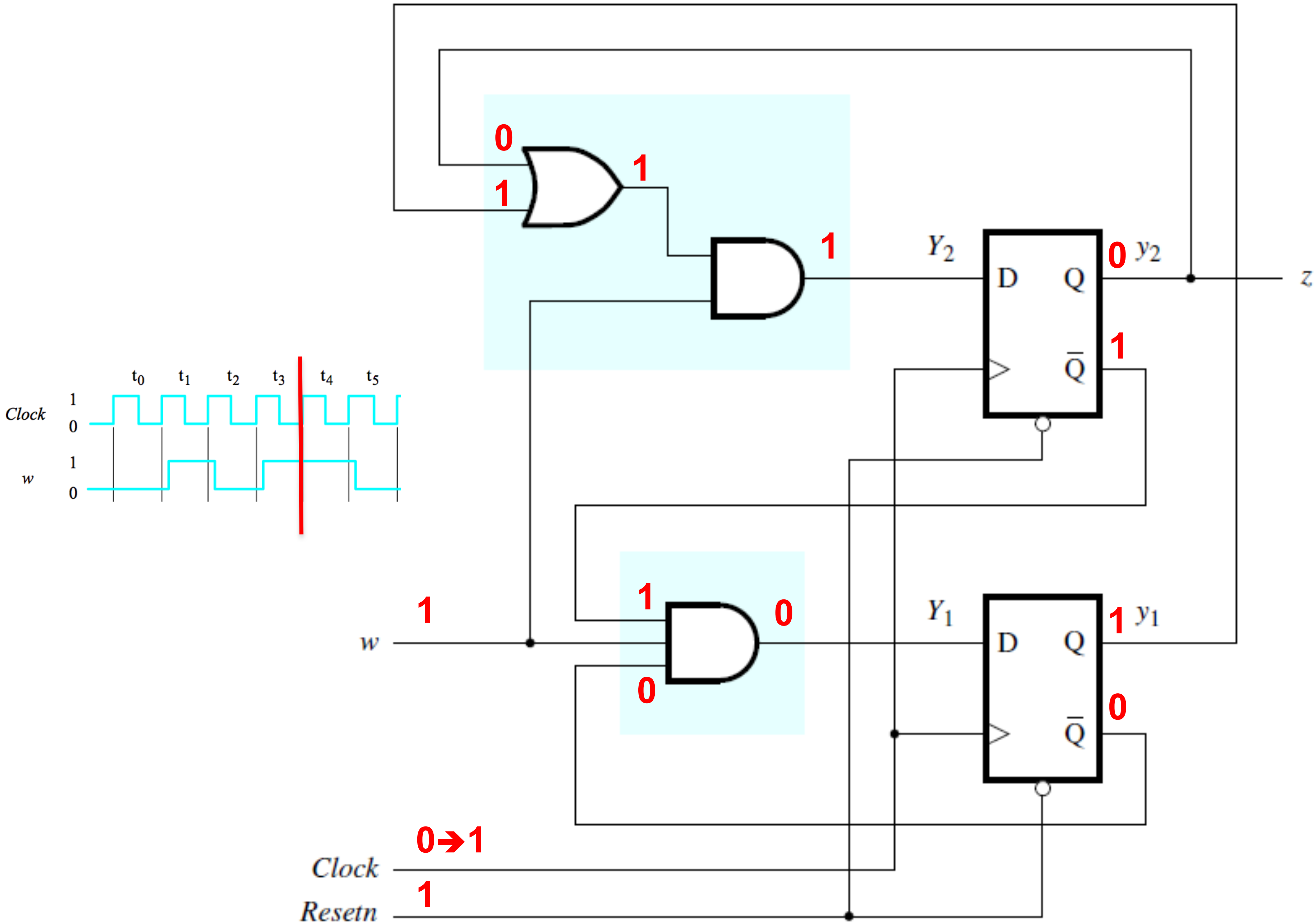
State A=00



State A=00

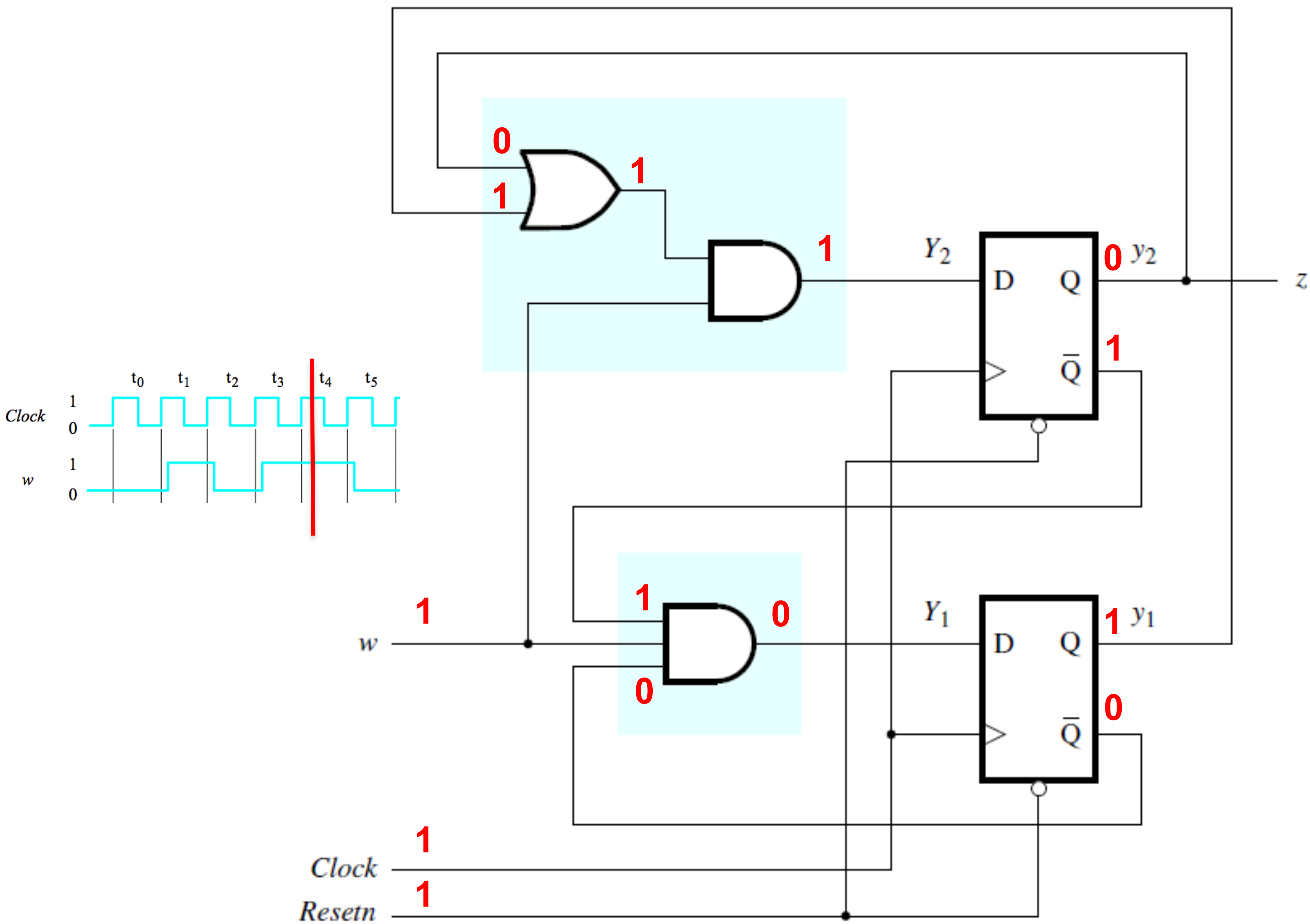


State B=01



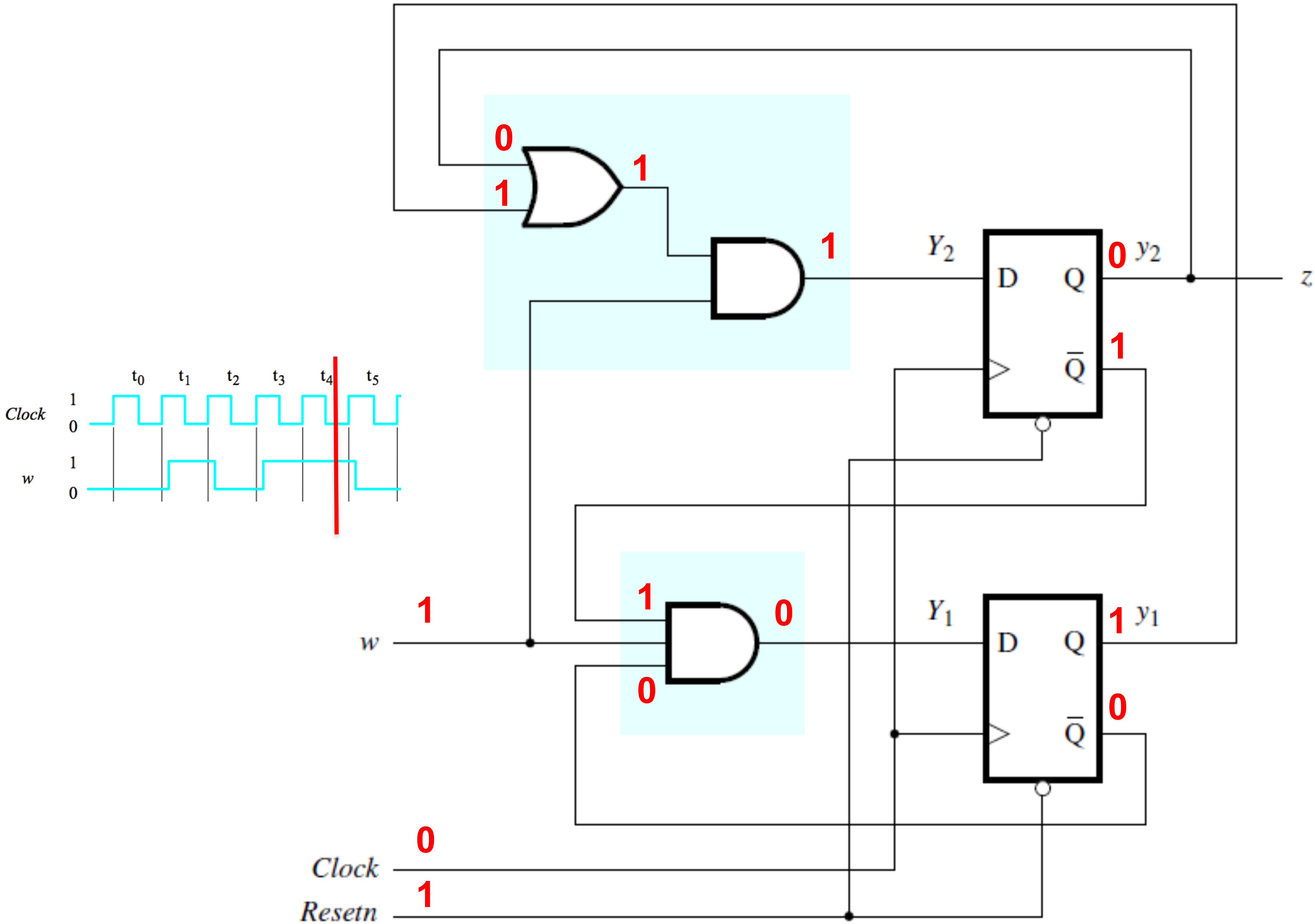
0
1
1
1
Y_2
D
Q
0 y_2
z
1
$\bar{Q}$
t0 t1 t2 t3 t4 t5
Clock
1
0
w
1
0
w
1
1
0
0
Y_1
D
Q
1 y_1
0
$\bar{Q}$
0→1
Clock
1
Resetn

State B=01



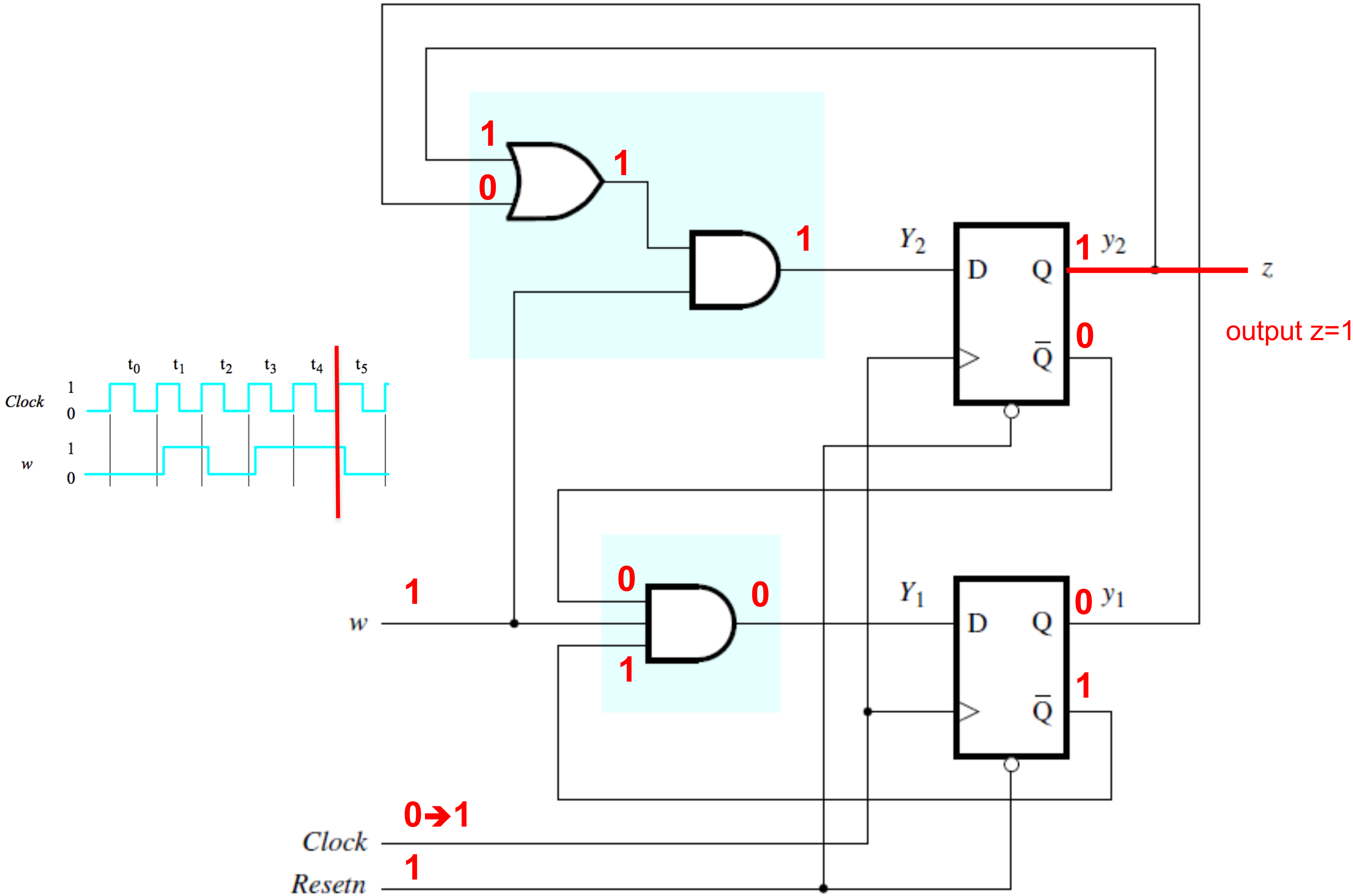
Clock
w
1
0
1
0
t0
t1
t2
t3
t4
t5
0
1
1
1
Y2
0 y2
D
Q
z
1
Q̄
1
0
w
1
0
Y1
1 y1
D
Q
0
Q̄
1
Clock
1
Resetn

State B=01



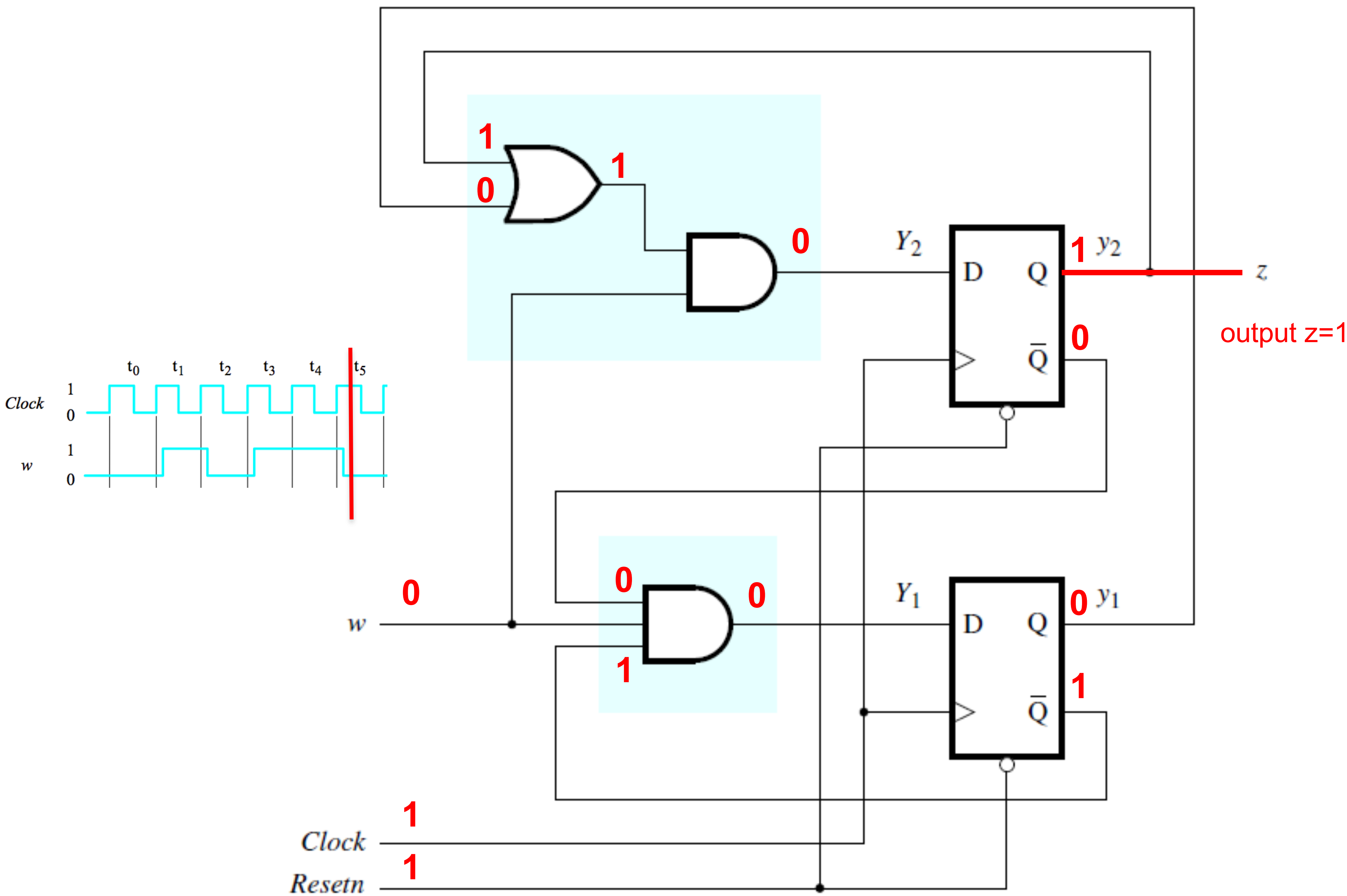
t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0
0
1
1
1
Y2
D
Q
0 y2
z
1
w
1
1
0
0
Y1
D
Q
1 y1
0
Clock
0
Resetn
1

State C=10

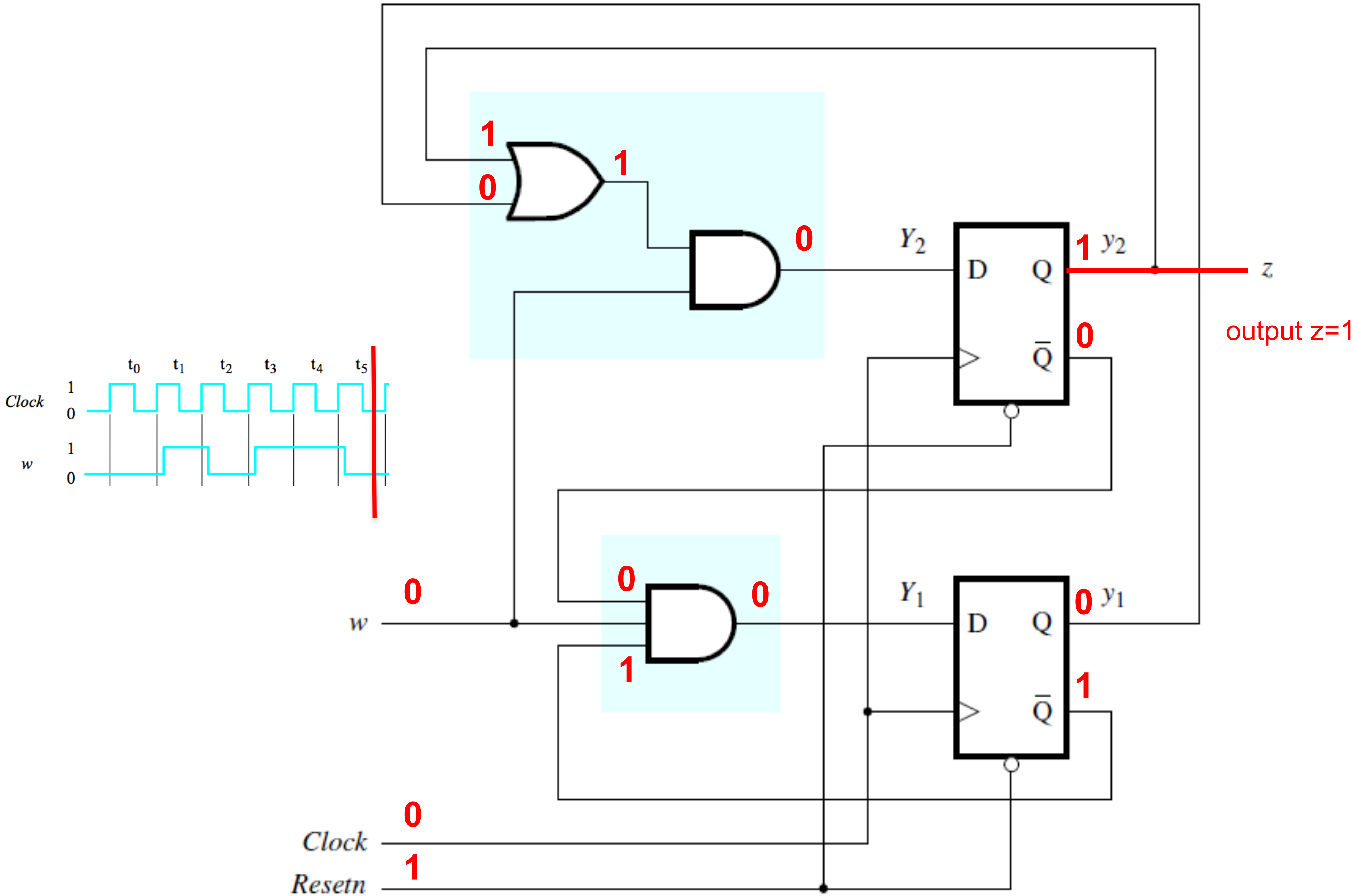


t0 t1 t2 t3 t4 t5
Clock
1
0
w
1
0
1
0
1
1
Y2
D
Q
1 y2
z
0
Q
output z=1
0
0
Y1
D
Q
0 y1
w
1
1
1
Q
Clock
0→1
Resetn
1

State C=10

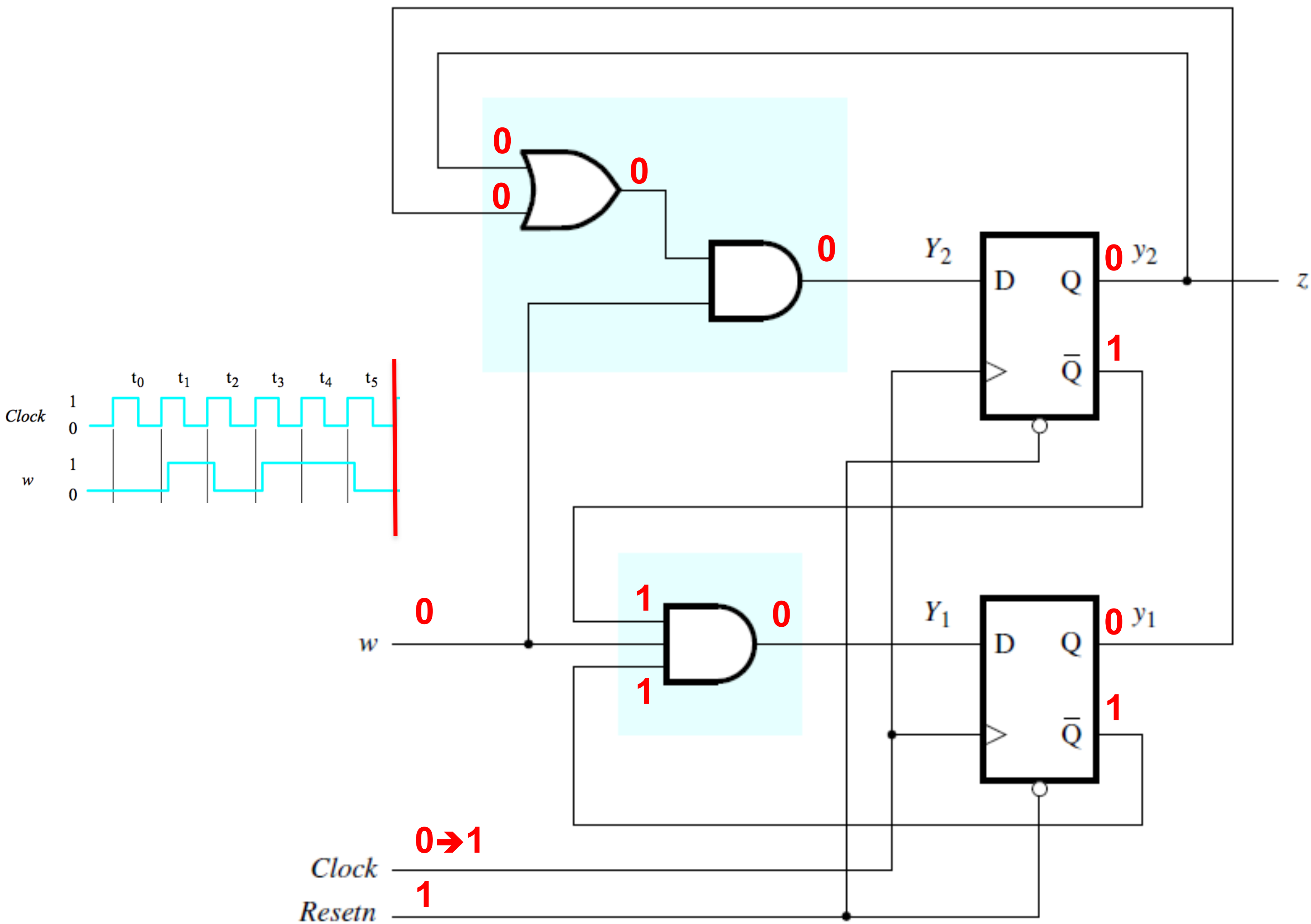

Clock
w
t0 t1 t2 t3 t4 t5
1
0
1
0
Y2
y2
D
Q
z
output z=1
Y1
y1
w
Clock
Resetn
1
0
1
0
1
0
0
0
0
1
0
1
1
1

State C=10



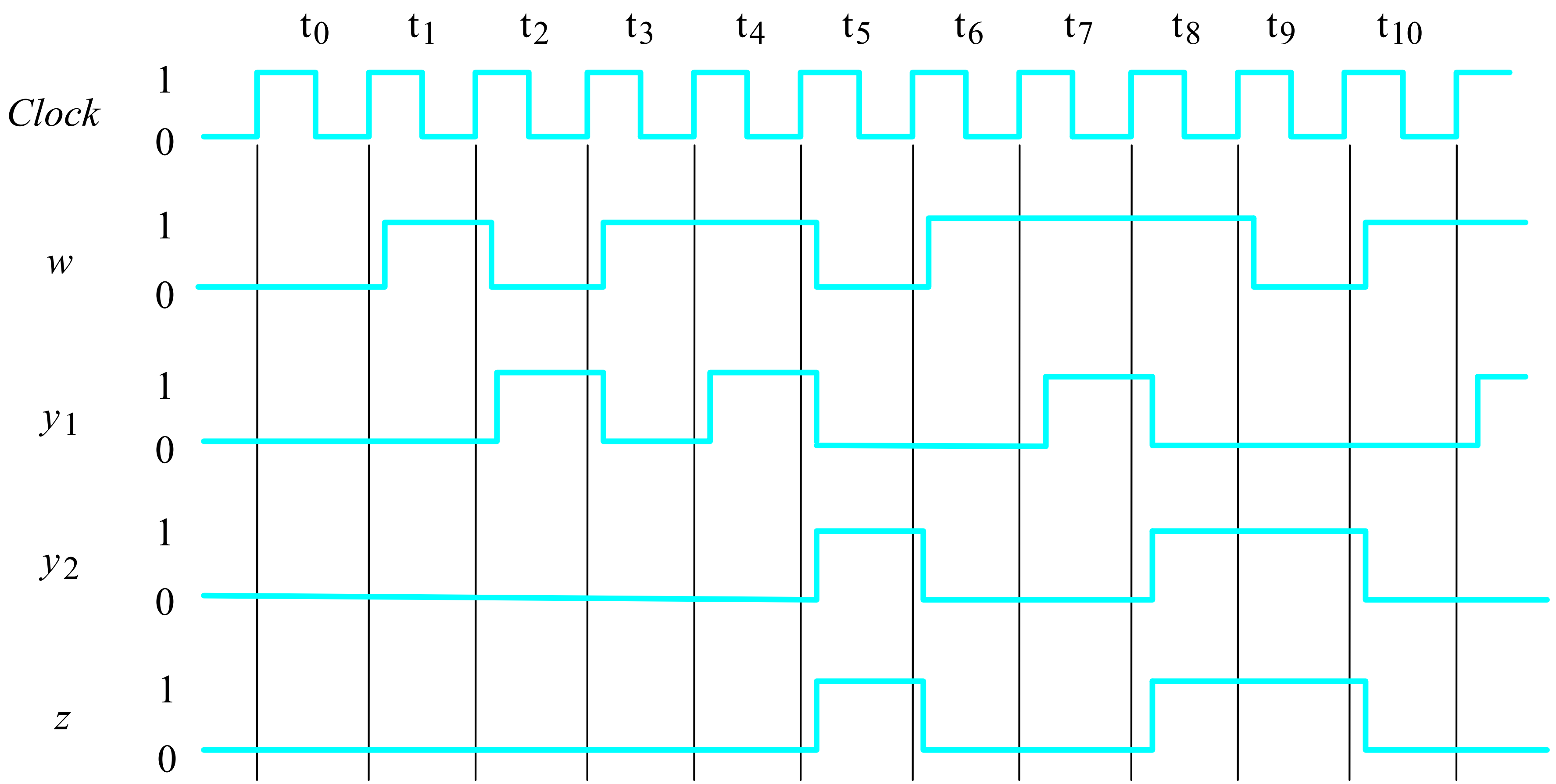
t0 t1 t2 t3 t4 t5
Clock
1
0
w
1
0
1
0
1
1
0
Y2
1 y2
D Q
z
0
output z=1
Q̄
w
0
0
0
1
Y1
0 y1
D Q
1
Q̄
0
Clock
1
Resetn

State A=00



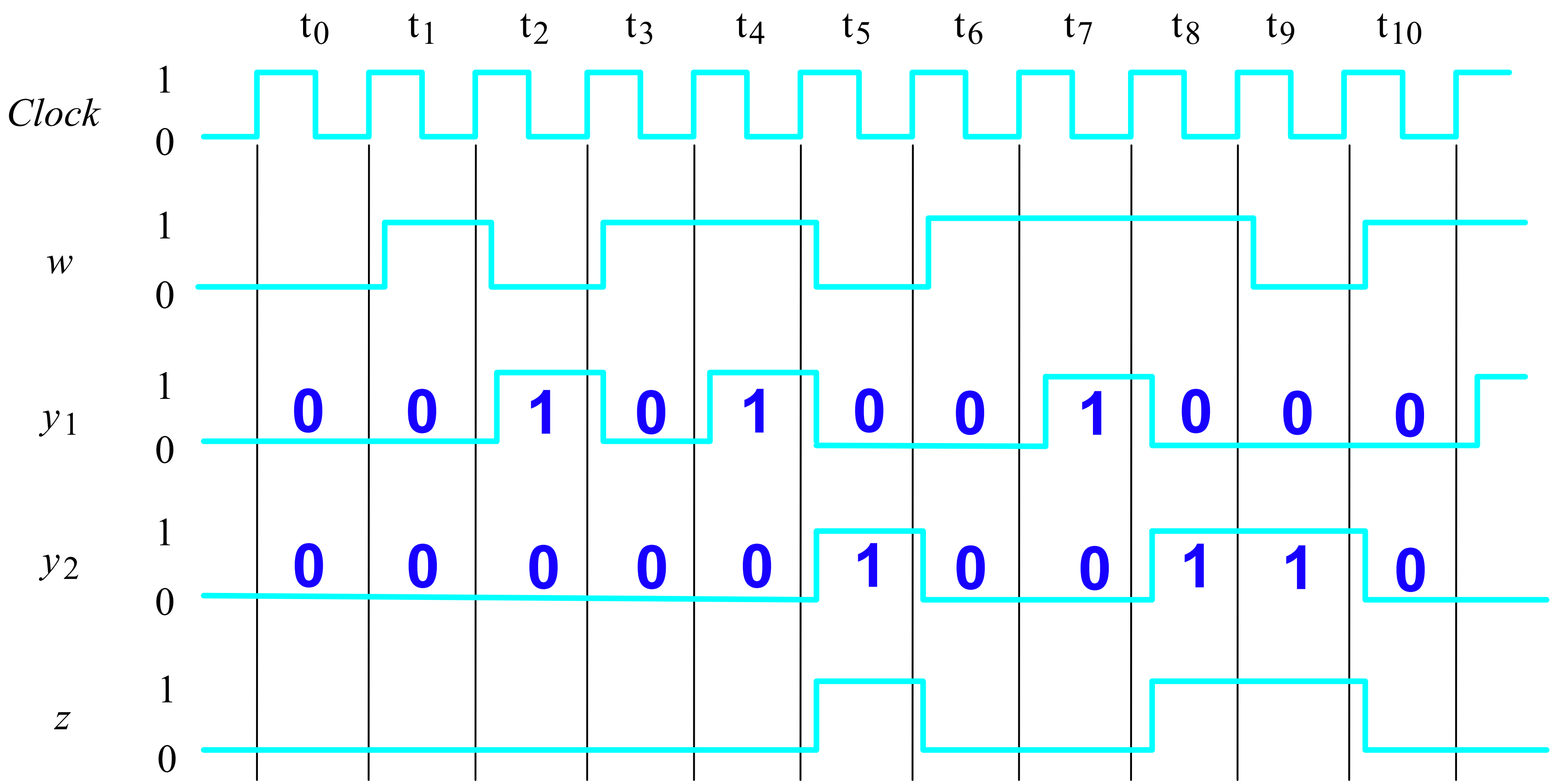
0
0
0
0
Y_2
D
Q
0 y_2
z
$\overline{Q}$
1
t_0 t_1 t_2 t_3 t_4 t_5
Clock
1
0
w
1
0
w
0
1
1
0
Y_1
D
Q
0 y_1
$\overline{Q}$
1
0→1
Clock
1
Resetn

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

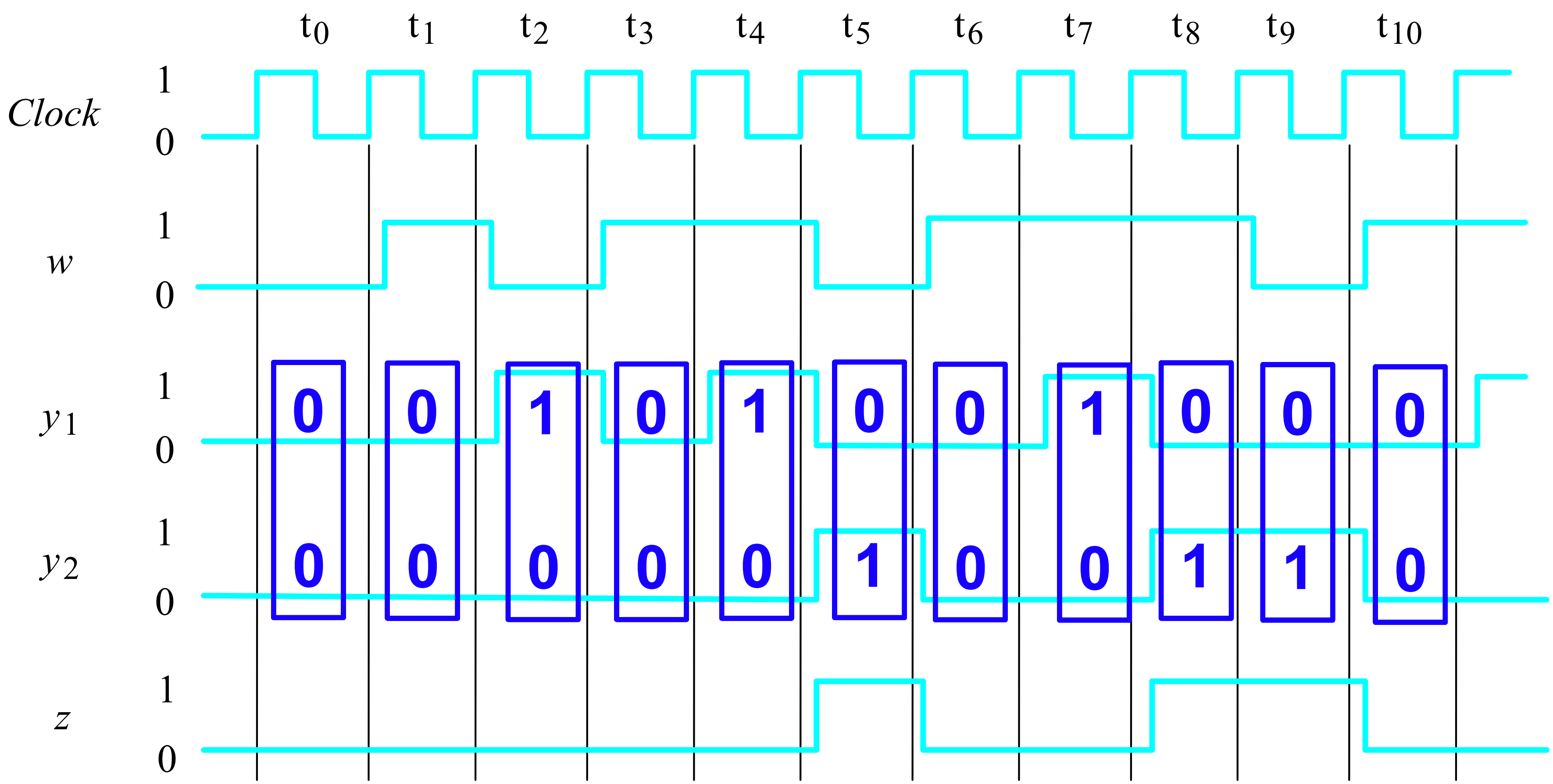


[Figure 6.9 from the textbook]

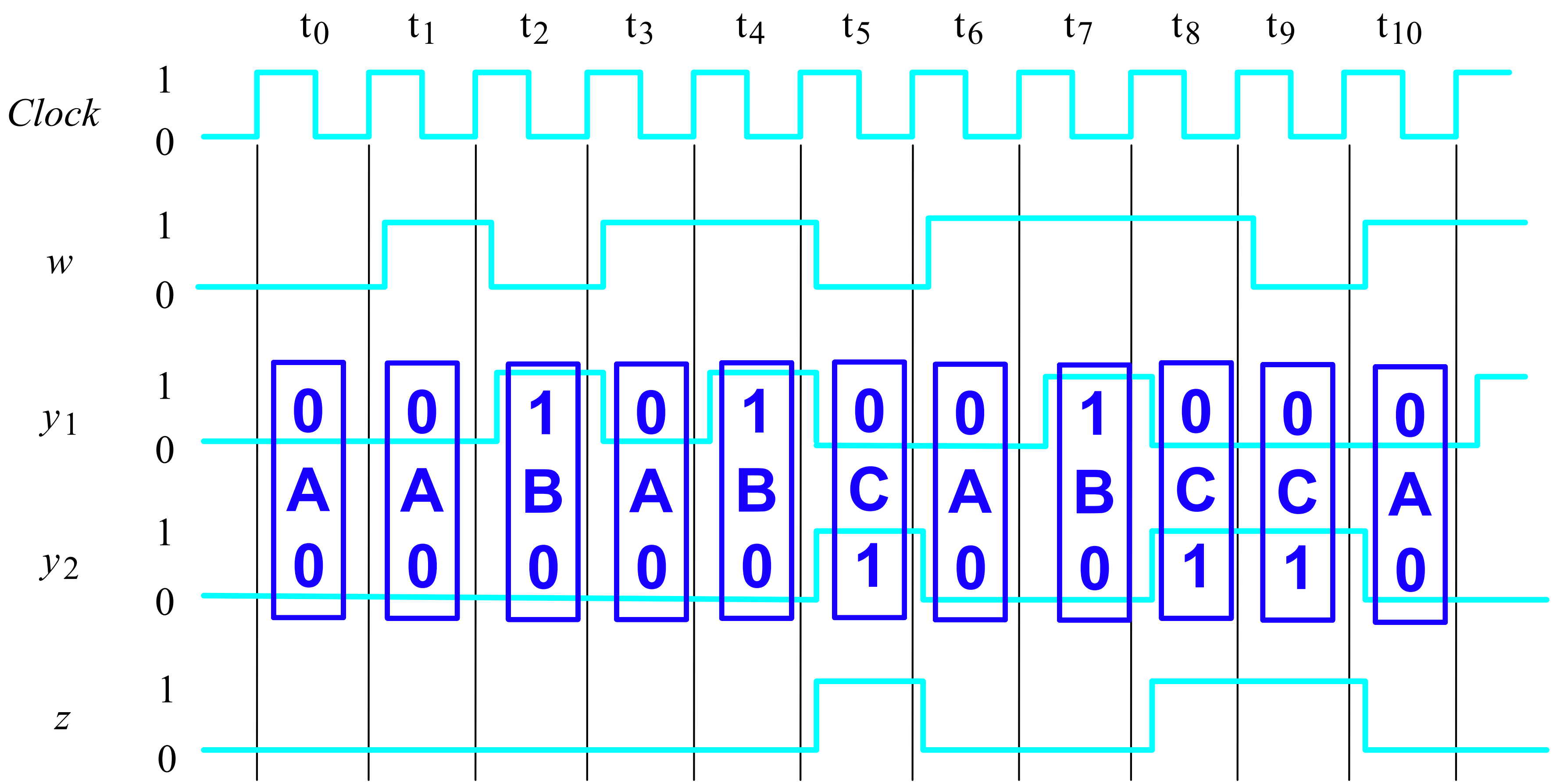
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0



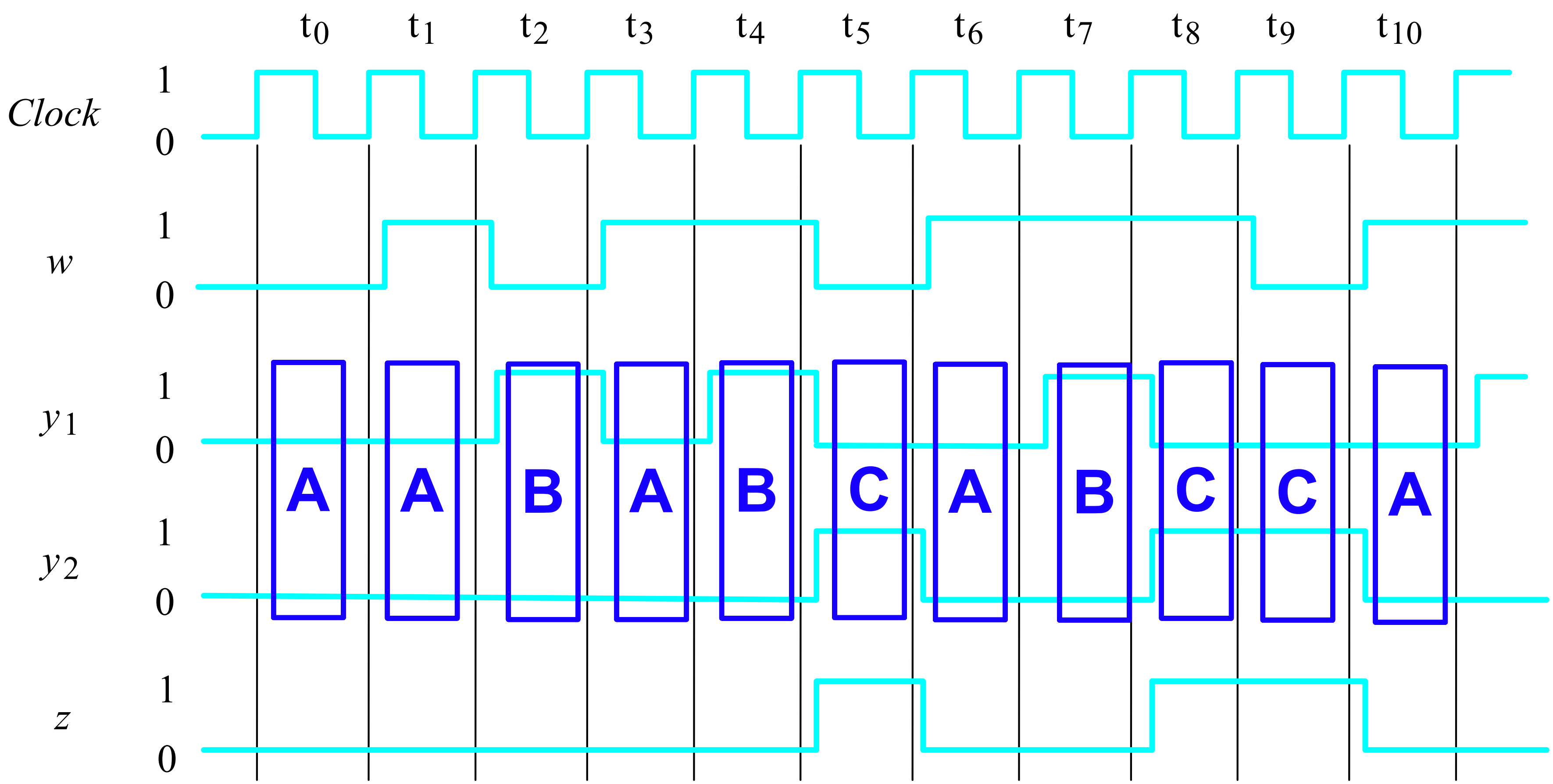
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0



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



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



Summary: Designing a Moore Machine

- Obtain the circuit specification.
- Derive a state diagram.
- Derive the state table.
- Decide on a state encoding.
- Encode the state table.
- Derive the output logic and next-state logic.
- Draw the circuit diagram
- Add a reset signal.

Alternative State Encoding for Example #1:
A=00, B=01, C=11

(Also Uses Two Flip-Flops)

A Better State Encoding

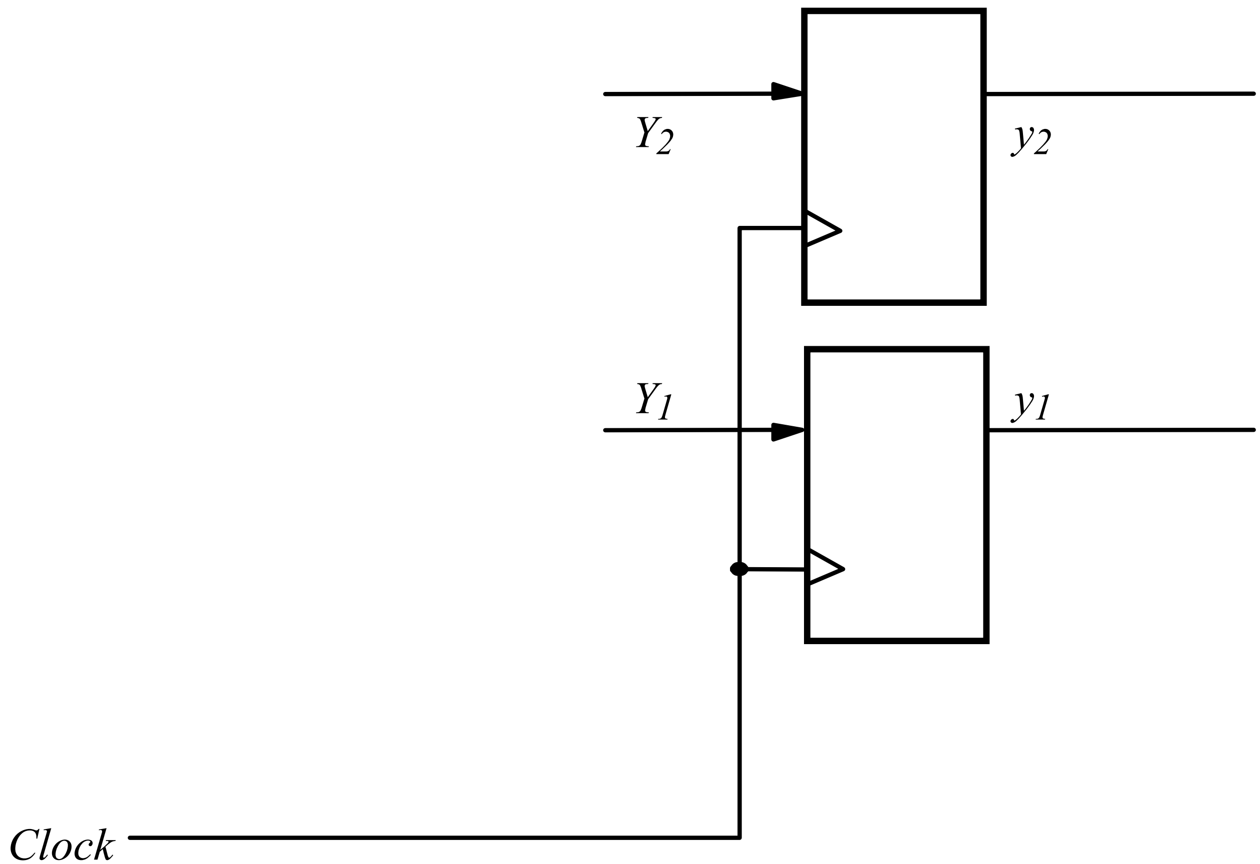
Present state	Next state		Output z
	$w = 0$	$w = 1$	
A	A	B	0
B	A	C	0
C	A	C	1

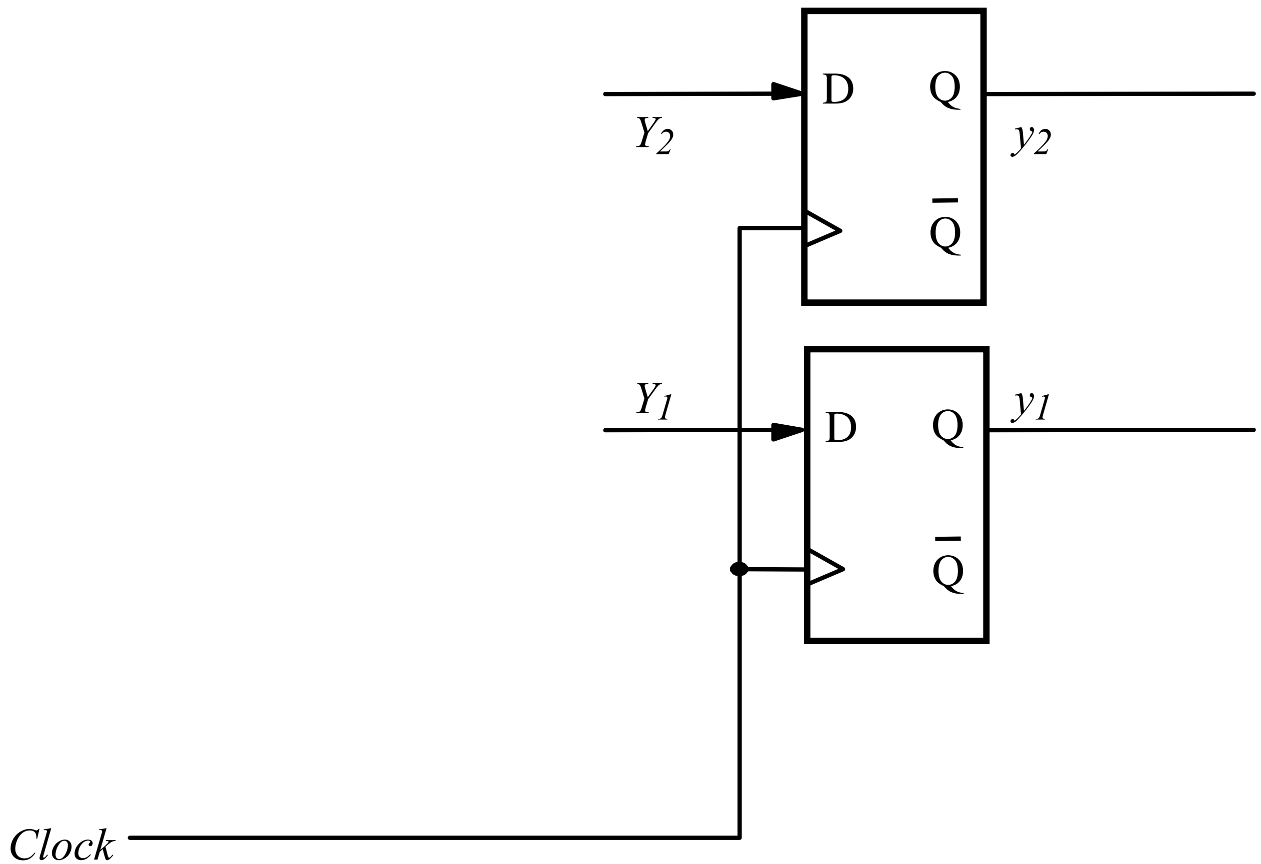
Suppose we encoded our states another way:

A = 00

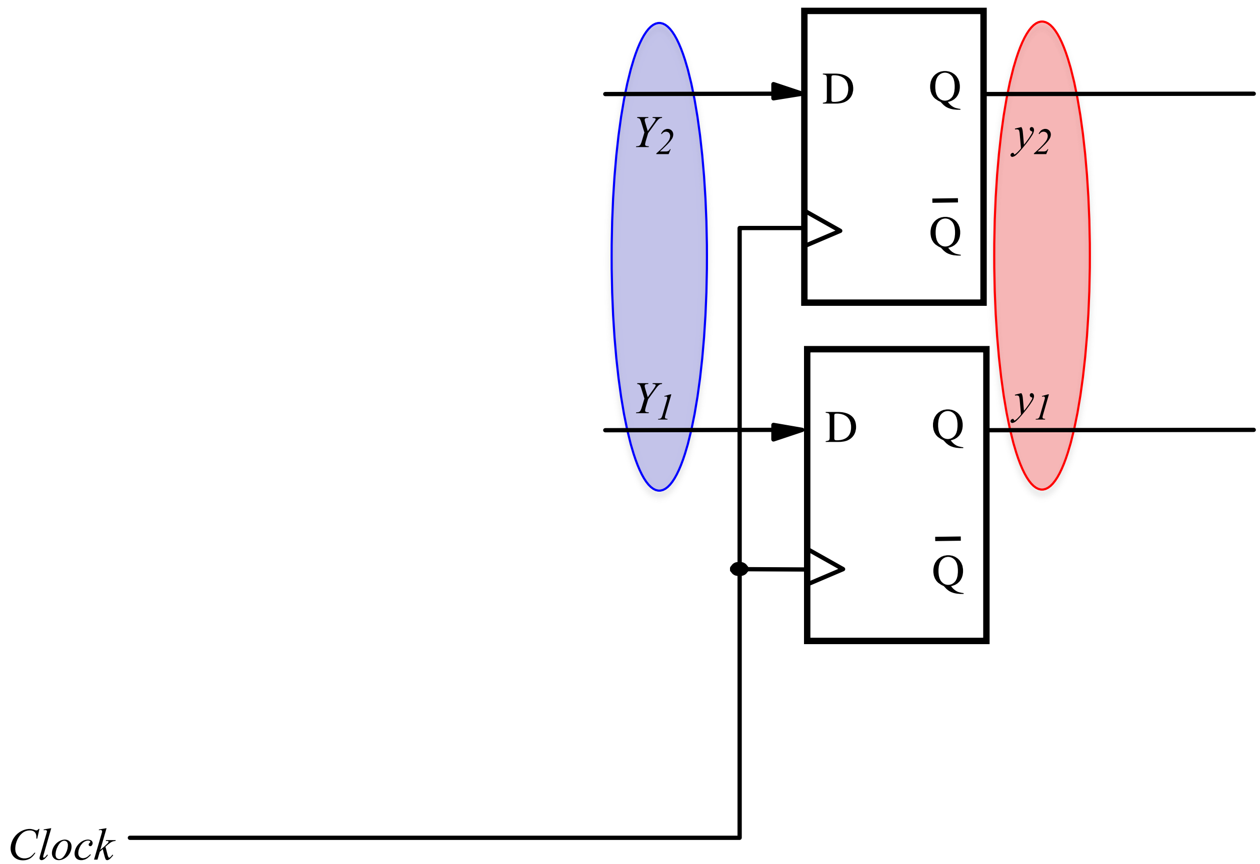
B = 01

C = 11


Y2
y2
Y1
y1
Clock



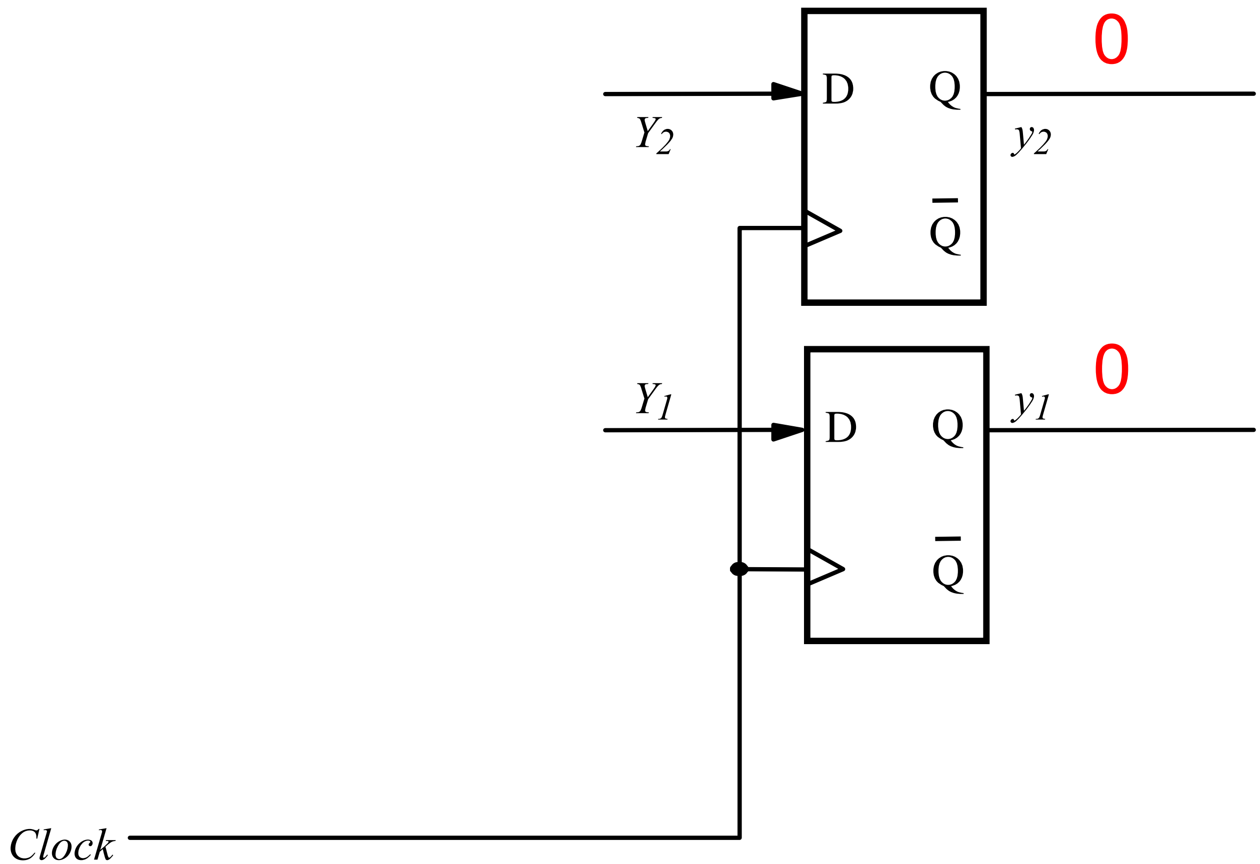
Let’s pick D Flip-Flops.



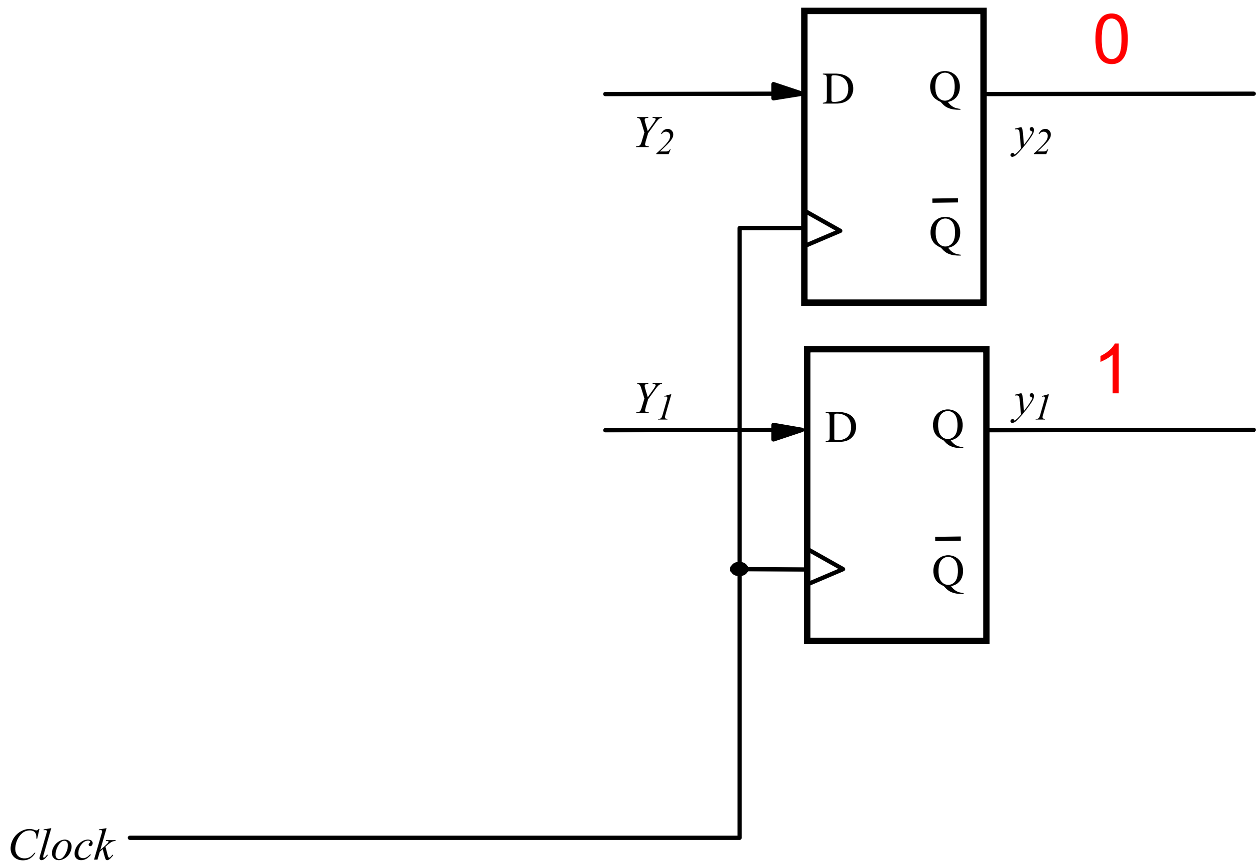
We will call y_1 and y_2 the *present state variables*.

We will call Y_1 and Y_2 the *next state variables*.

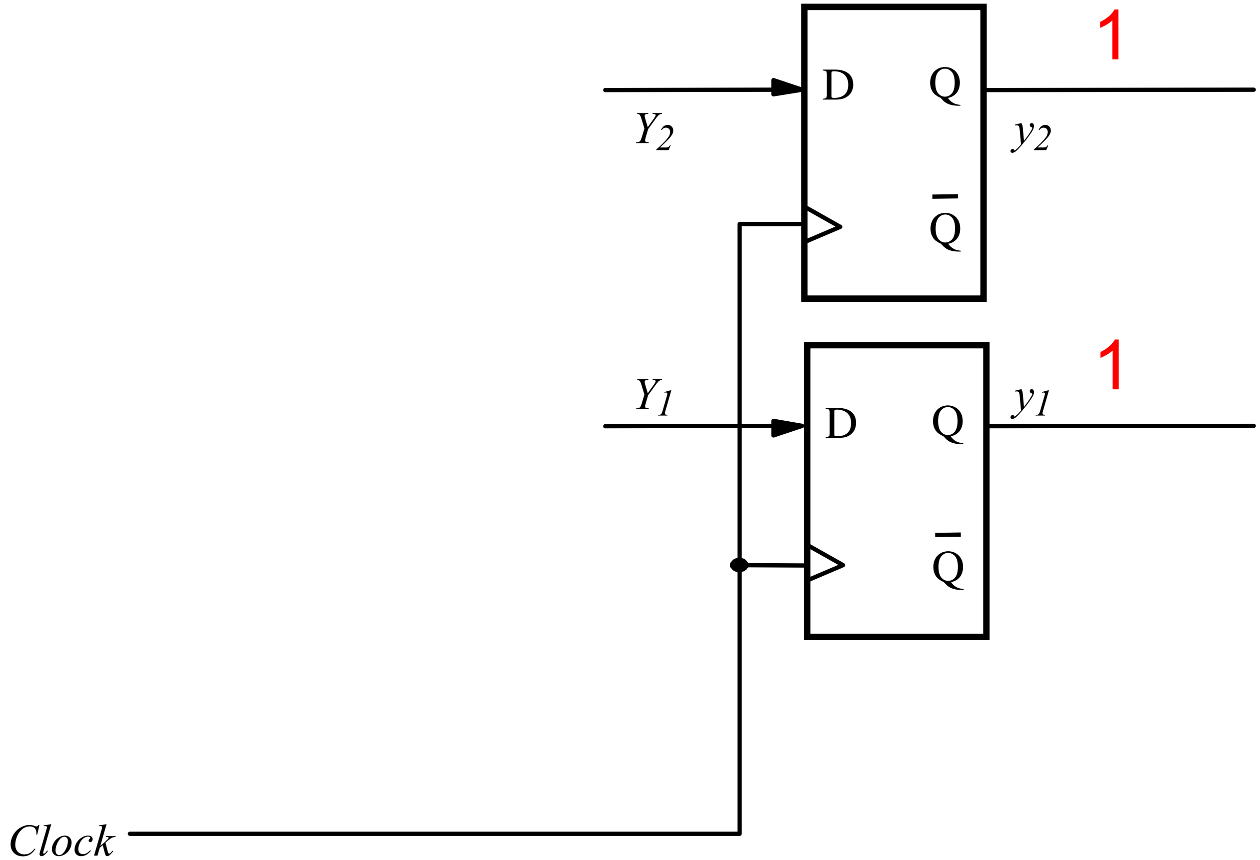
[Figure 6.5 from the textbook]



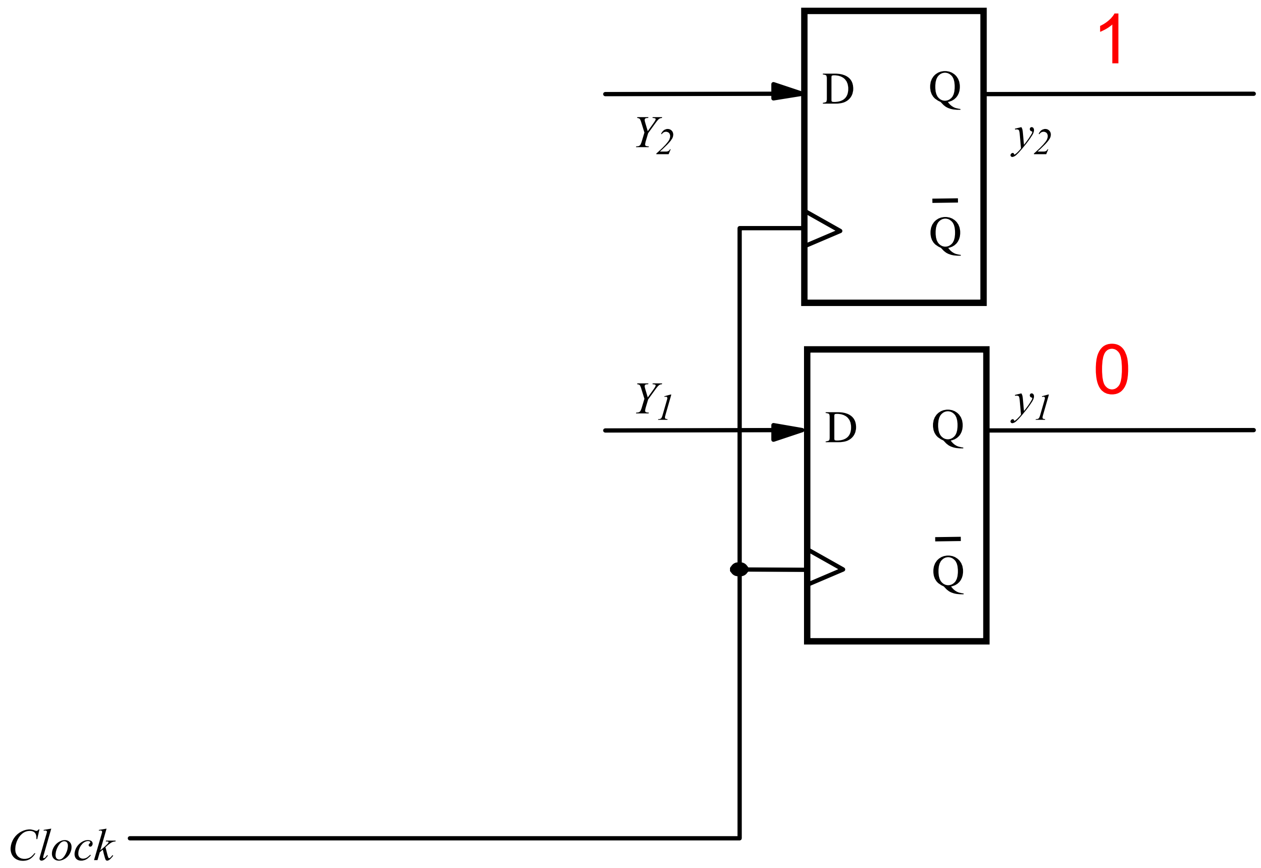
Two zeros on the output JOINTLY represent state A.



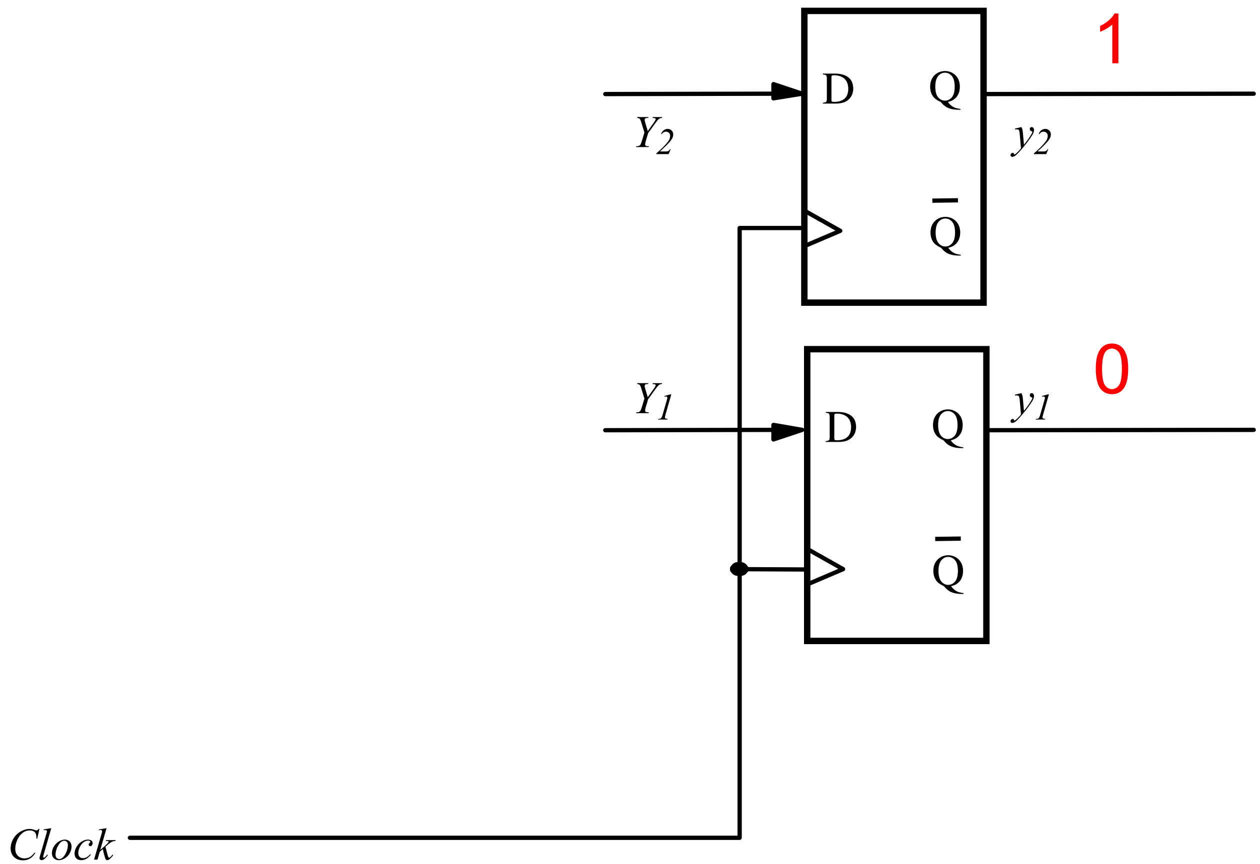
This flip-flop output pattern represents state B.



This flip-flop output pattern represents state C.



What does this flip-flop output pattern represent?



This would be state D, but we don't have one in this example. So this is an impossible state.

A Better State Encoding

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

Suppose we encoded our states another way:

A = 00

B = 01

C = 11

A Better State Encoding

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	Next state $w = 0$	Next state $w = 1$	Output z

A = 00

B = 01

C = 11

A Better State Encoding

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

A = 00

B = 01

C = 11

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

Let's Derive the Logic Expressions

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

[Figure 6.16 from the textbook]

	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*

[Figure 6.16 from the textbook]

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

y_2	y_1	z
0	0	
0	1	
1	0	
1	1	

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

$Q(t) = y_2 y_1$ and $Q(t+1) = Y_2 Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	1

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

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

w	y_2	y_1	Y_2	Y_1
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	1

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

$Q(t) = y_2 y_1$ and $Q(t+1) = Y_2 Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	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*

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	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*

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	
0	0	1	0	
0	1	0	d	
0	1	1	0	
1	0	0		
1	0	1		
1	1	0		
1	1	1		

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	1

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

	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*

w	y_2	y_1	Y_2	Y_1
0	0	0	0	
0	0	1	0	
0	1	0	d	
0	1	1	0	
1	0	0		
1	0	1		
1	1	0		
1	1	1		

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	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

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	
0	0	1	0	
0	1	0	d	
0	1	1	0	
1	0	0		
1	0	1		
1	1	0		
1	1	1		

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	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

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	
0	0	1	0	
0	1	0	d	
0	1	1	0	
1	0	0	0	
1	0	1	1	
1	1	0	d	
1	1	1	1	

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	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*

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	
0	0	1	0	
0	1	0	d	
0	1	1	0	
1	0	0	0	
1	0	1	1	
1	1	0	d	
1	1	1	1	

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	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*

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	
0	0	1	0	
0	1	0	d	
0	1	1	0	
1	0	0	0	
1	0	1	1	
1	1	0	d	
1	1	1	1	

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	1

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

$Q(t) = y_2 y_1$ and $Q(t+1) = Y_2 Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	
0	0	1	0	
0	1	0	d	
0	1	1	0	
1	0	0	0	
1	0	1	1	
1	1	0	d	
1	1	1	1	

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	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*

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	0
0	0	1	0	0
0	1	0	d	d
0	1	1	0	0
1	0	0	0	
1	0	1	1	
1	1	0	d	
1	1	1	1	

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	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

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	0
0	0	1	0	0
0	1	0	d	d
0	1	1	0	0
1	0	0	0	
1	0	1	1	
1	1	0	d	
1	1	1	1	

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	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*

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	0
0	0	1	0	0
0	1	0	d	d
0	1	1	0	0
1	0	0	0	
1	0	1	1	
1	1	0	d	
1	1	1	1	

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	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*

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	0
0	0	1	0	0
0	1	0	d	d
0	1	1	0	0
1	0	0	0	1
1	0	1	1	1
1	1	0	d	d
1	1	1	1	1

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	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*

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	0
0	0	1	0	0
0	1	0	d	d
0	1	1	0	0
1	0	0	0	1
1	0	1	1	1
1	1	0	d	d
1	1	1	1	1

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	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

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	0
0	0	1	0	0
0	1	0	d	d
0	1	1	0	0
1	0	0	0	1
1	0	1	1	1
1	1	0	d	d
1	1	1	1	1

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	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*

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	0
0	0	1	0	0
0	1	0	d	d
0	1	1	0	0
1	0	0	0	1
1	0	1	1	1
1	1	0	d	d
1	1	1	1	1

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	1

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	0
0	0	1	0	0
0	1	0	d	d
0	1	1	0	0
1	0	0	0	1
1	0	1	1	1
1	1	0	d	d
1	1	1	1	1

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	1

Deriving the Logic Expressions

Y_2: w \ y_2y_1	00	01	11	10
0	0	0	0	d
1	0	1	1	d

$$Y_2(w, y_2, y_1) = wy_1$$

Y_1: w \ y_2y_1	00	01	11	10
0	0	0	0	d
1	1	1	1	d

$$Y_1(w, y_2, y_1) = w$$

z: y_2 \ y_1	0	1
0	0	0
1	d	1

$$z(y_2, y_1) = y_2$$

$Q(t) = y_2y_1$ and $Q(t+1) = Y_2Y_1$

w	y_2	y_1	Y_2	Y_1
0	0	0	0	0
0	0	1	0	0
0	1	0	d	d
0	1	1	0	0
1	0	0	0	1
1	0	1	1	1
1	1	0	d	d
1	1	1	1	1

y_2	y_1	z
0	0	0
0	1	0
1	0	d
1	1	1

Original State Encodings

Y_1

w \ y_2y_1	00	01	11	10
0	0	0	d	0
1	1	0	d	0

Y_2

w \ y_2y_1	00	01	11	10
0	0	0	d	0
1	0	1	d	1

z

y_2 \ y_1	0	1
0	0	0
1	1	d

New State Encodings

Y_1

w \ y_2y_1	00	01	11	10
0	0	0	0	d
1	1	1	1	d

Y_2

w \ y_2y_1	00	01	11	10
0	0	0	0	d
1	0	1	1	d

z

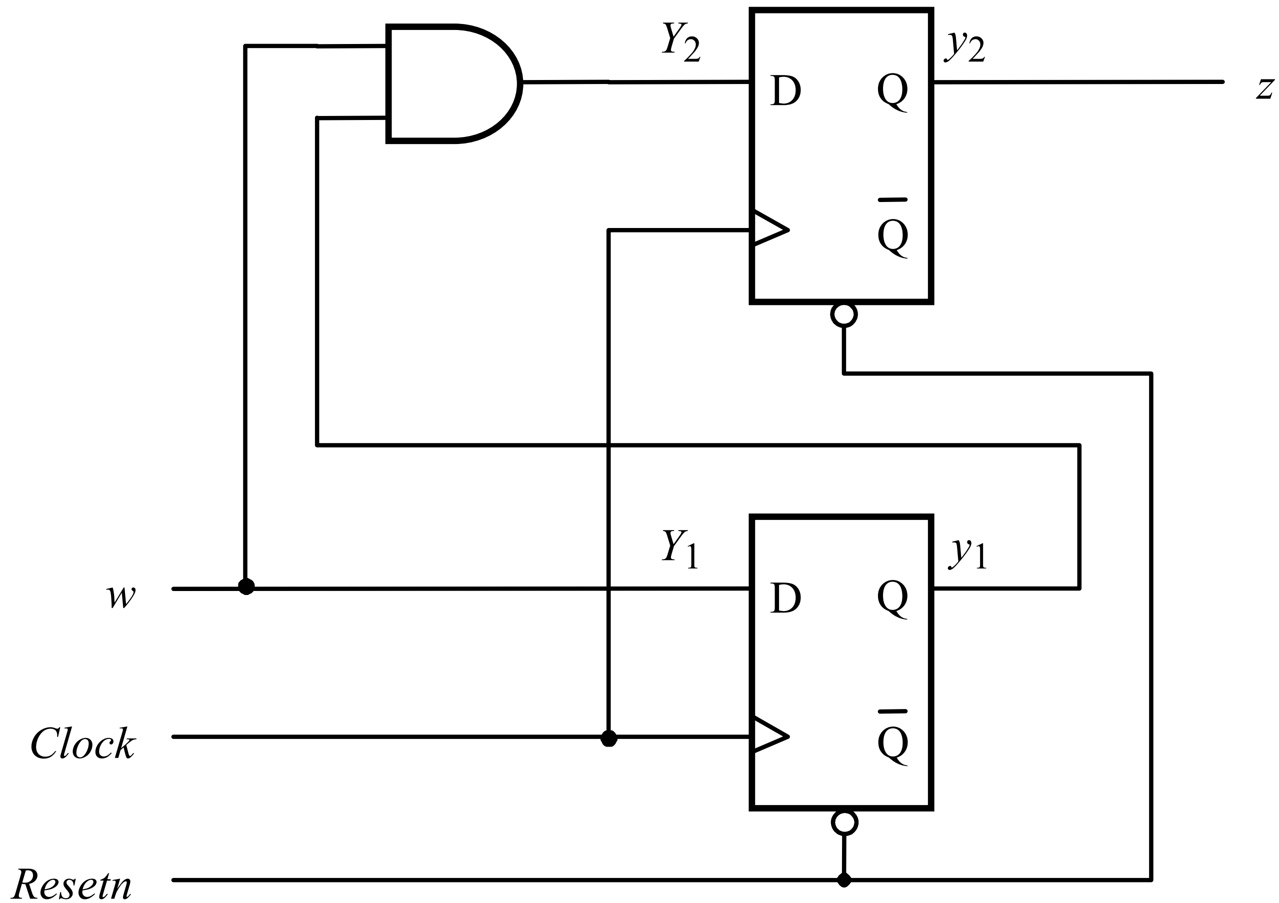
y_2 \ y_1	0	1
0	0	0
1	d	1

The New and Improved Circuit Diagram

$Y_1(\mathrm{w}, \mathrm{y}_2, \mathrm{y}_1) = w$

$Y_2(\mathrm{w}, \mathrm{y}_2, \mathrm{y}_1) = wy_1$

$z(\mathrm{y}_2, \mathrm{y}_1) = y_2$



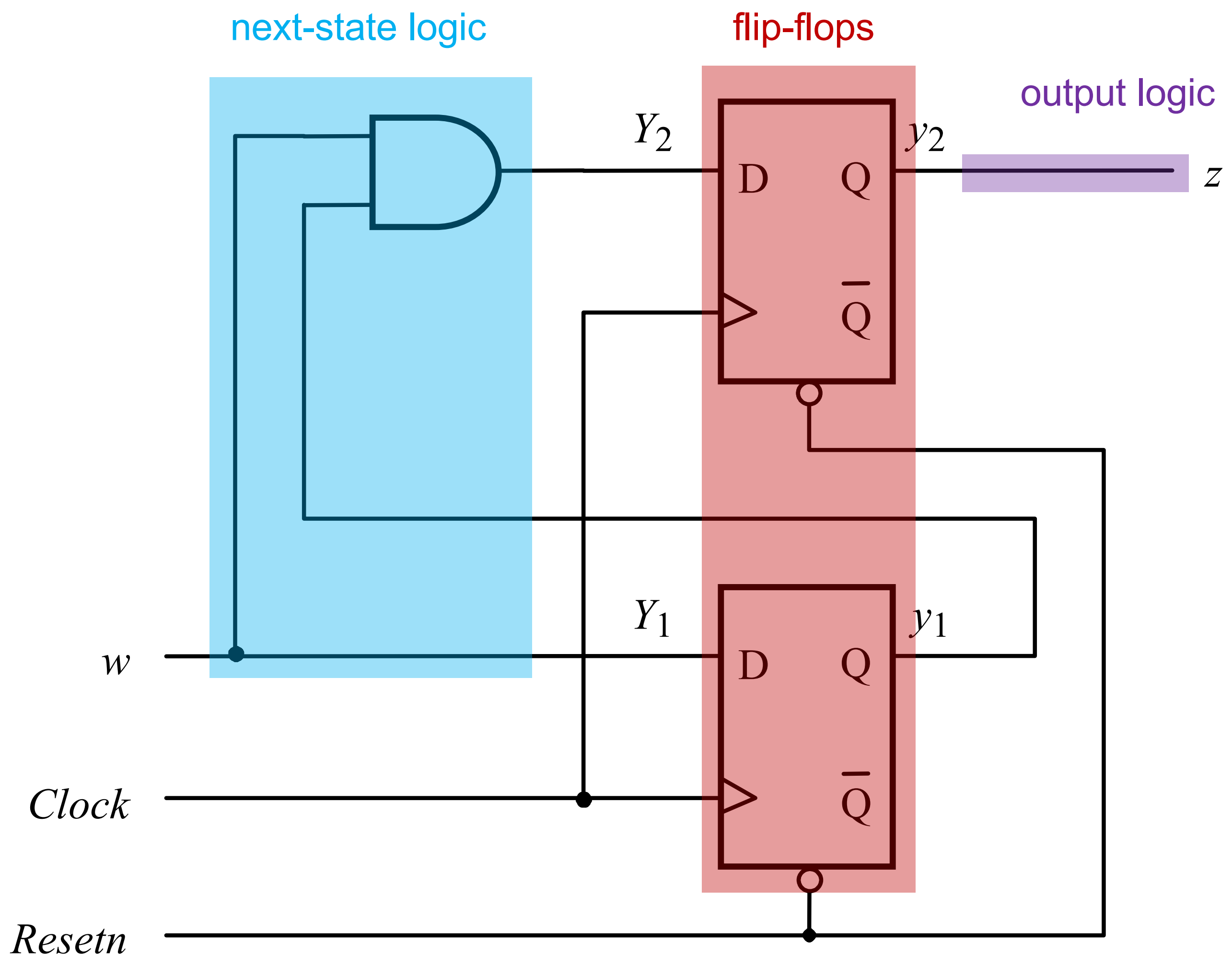
[Figure 6.17 from the textbook]

The New and Improved Circuit Diagram

$$Y_1(\mathrm{w}, \mathrm{y}_2, \mathrm{y}_1) = w$$

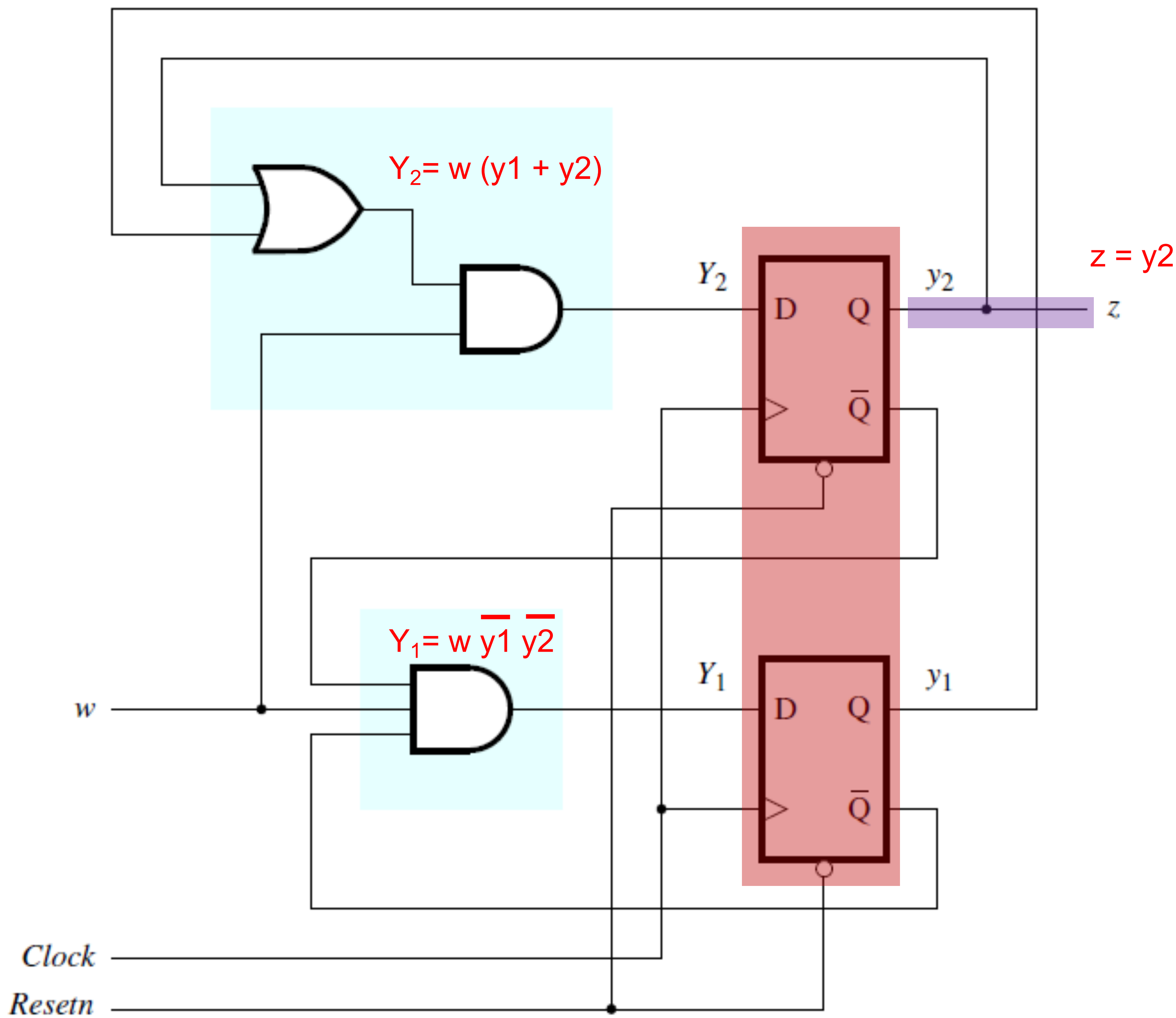
$$Y_2(\mathrm{w}, \mathrm{y}_2, \mathrm{y}_1) = wy_1$$

$$z(\mathrm{y}_2, \mathrm{y}_1) = y_2$$



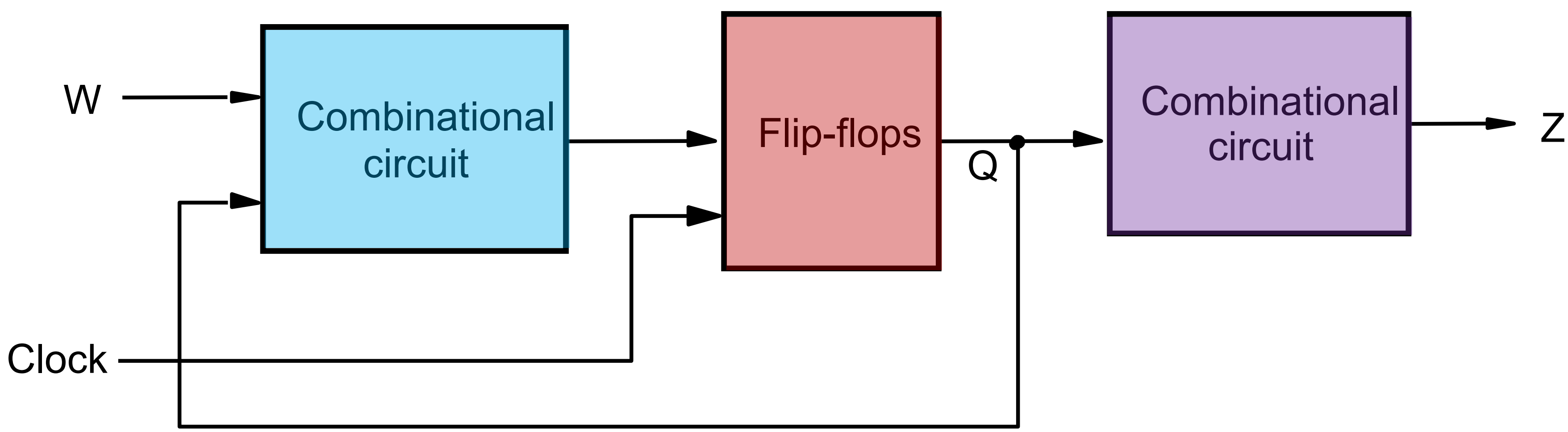
[Figure 6.17 from the textbook]

The Previous Circuit Diagram



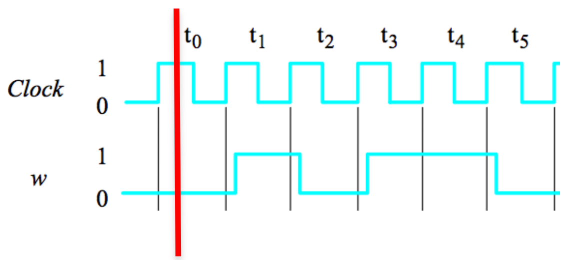
[Figure 6.8 from the textbook]

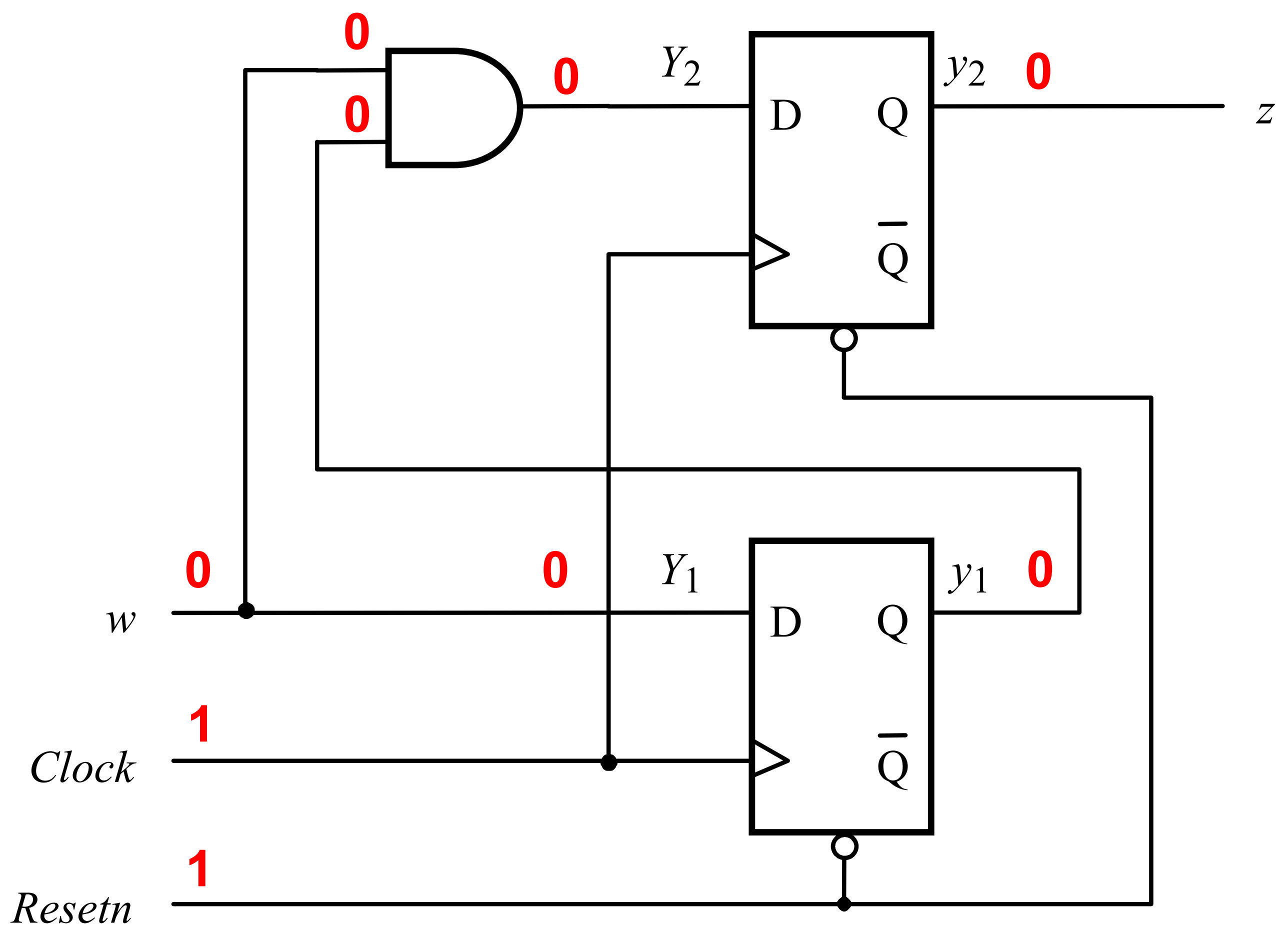
Moore Type



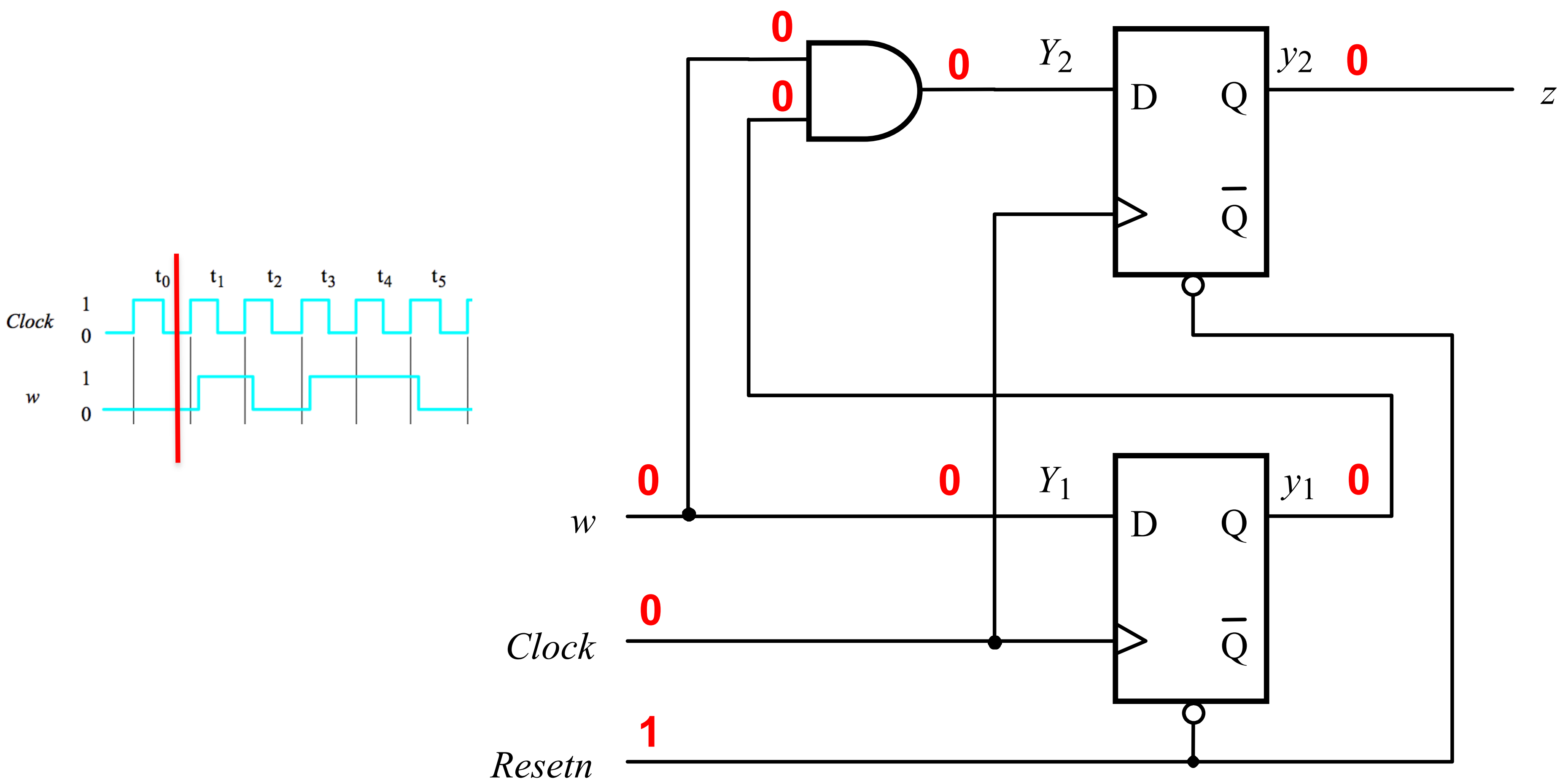
W
Combinational
circuit
Flip-flops
Q
Combinational
circuit
Z
Clock

State A=00


t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0

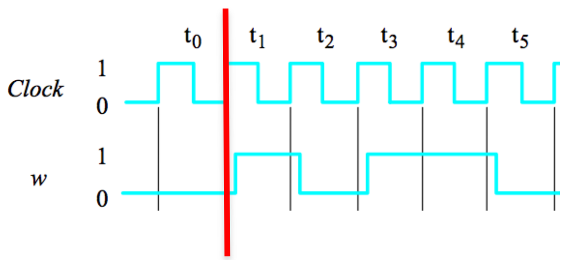

0
0
0
Y2
y2
0
D
Q
Q
z
0
0
Y1
y1
0
w
D
Q
1
Clock
Q
1
Resetn

State A=00

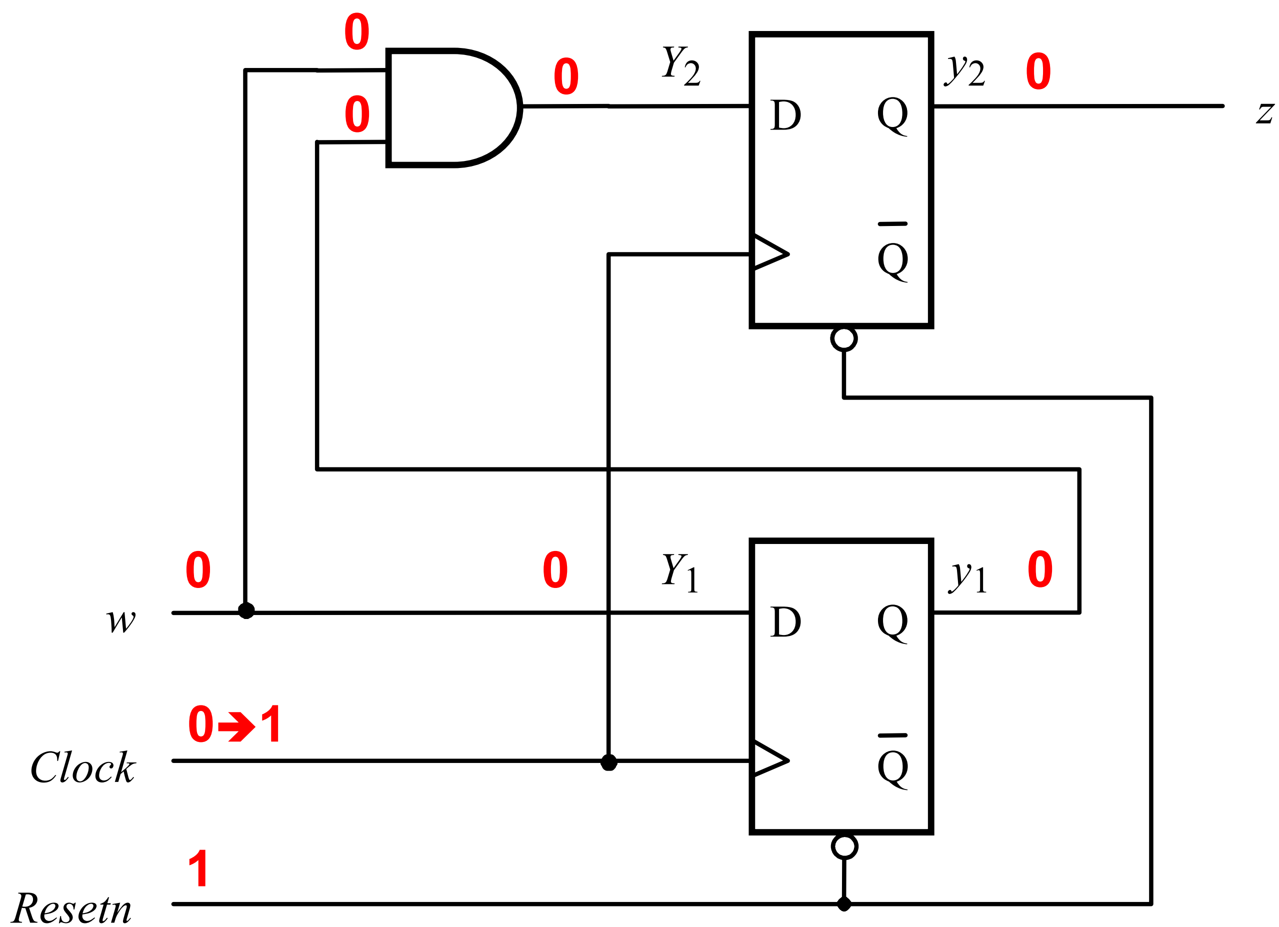


t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0
0
0
0
Y2
y2
0
D
Q
Q
z
0
0
Y1
y1
0
w
D
Q
0
Clock
Q
1
Resetn

State A=00

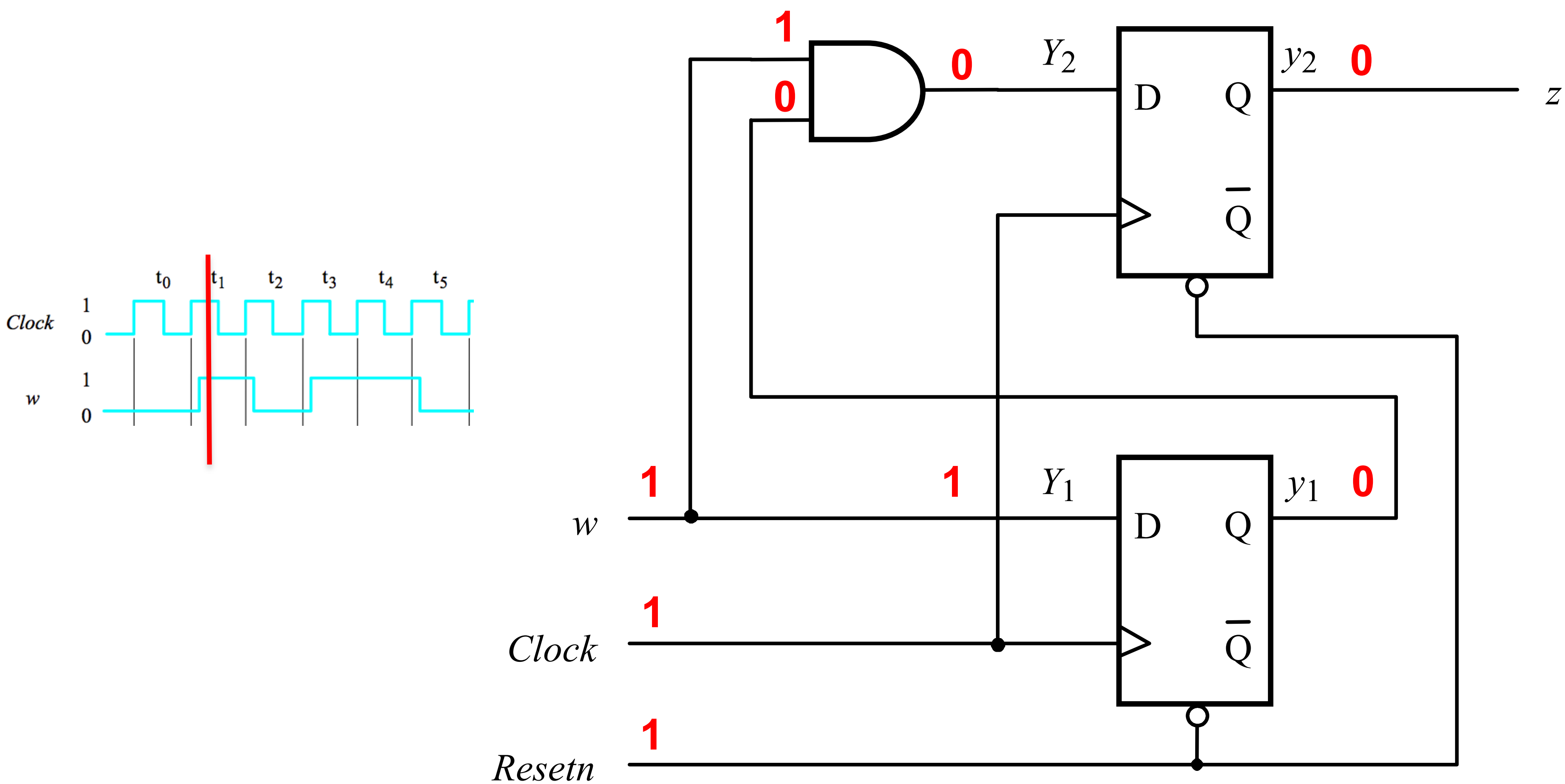


t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0



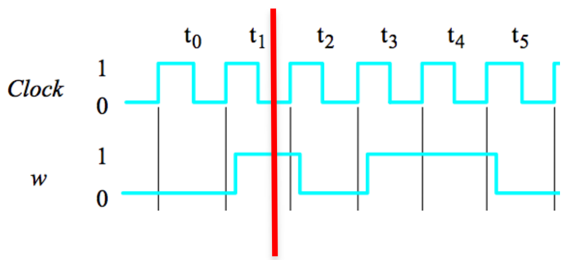
0
0
0
Y2
D
Q
y2
0
z
Q
0
0
Y1
D
Q
y1
0
w
0→1
Clock
Q
1
Resetn

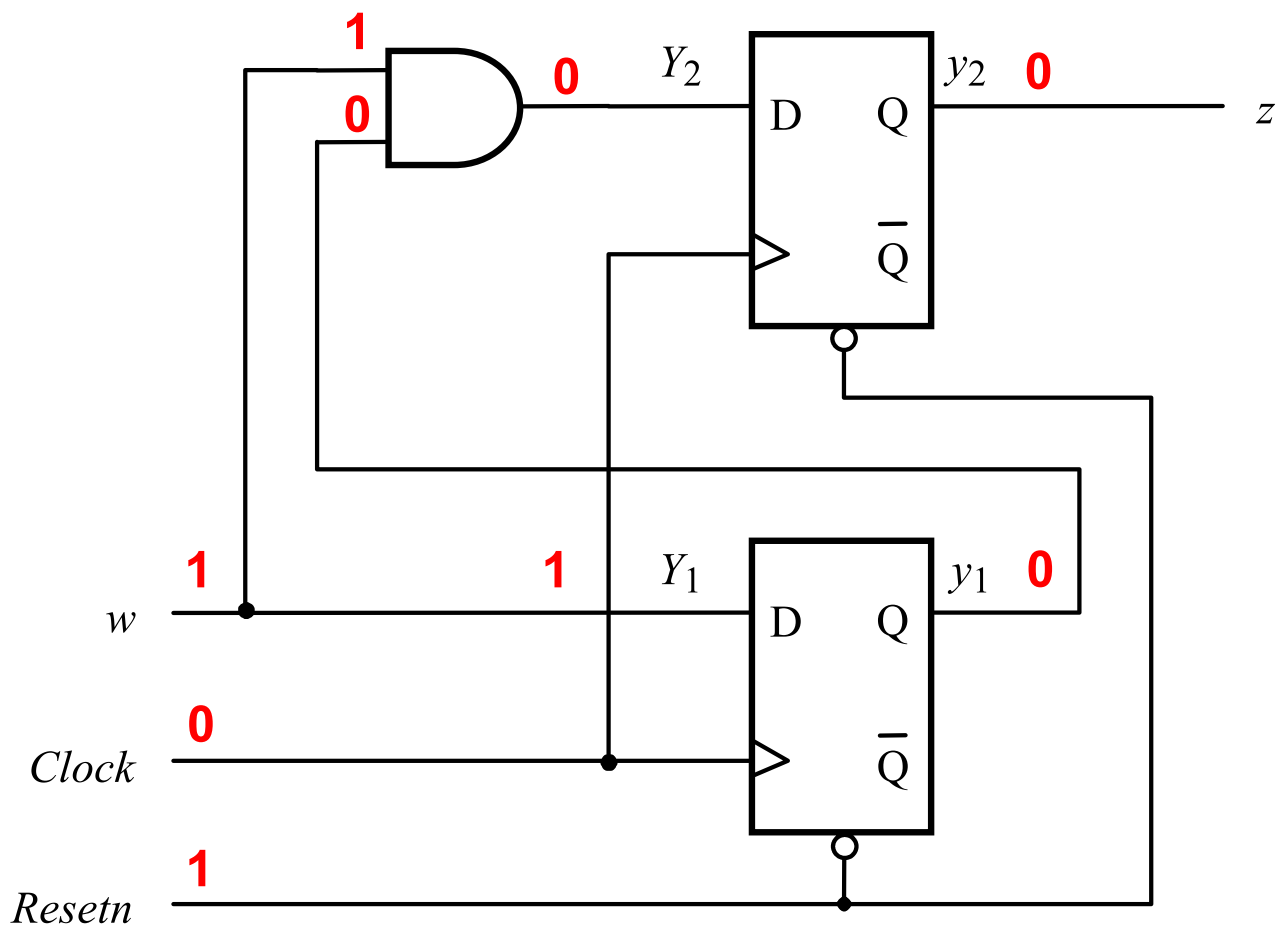
State A=00



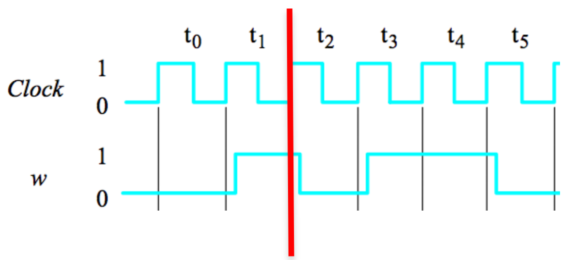
t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0
1
0
0
Y2
y2
0
D
Q
Q
z
1
1
Y1
y1
0
D
Q
Q
w
1
Clock
1
Resetn

State A=00

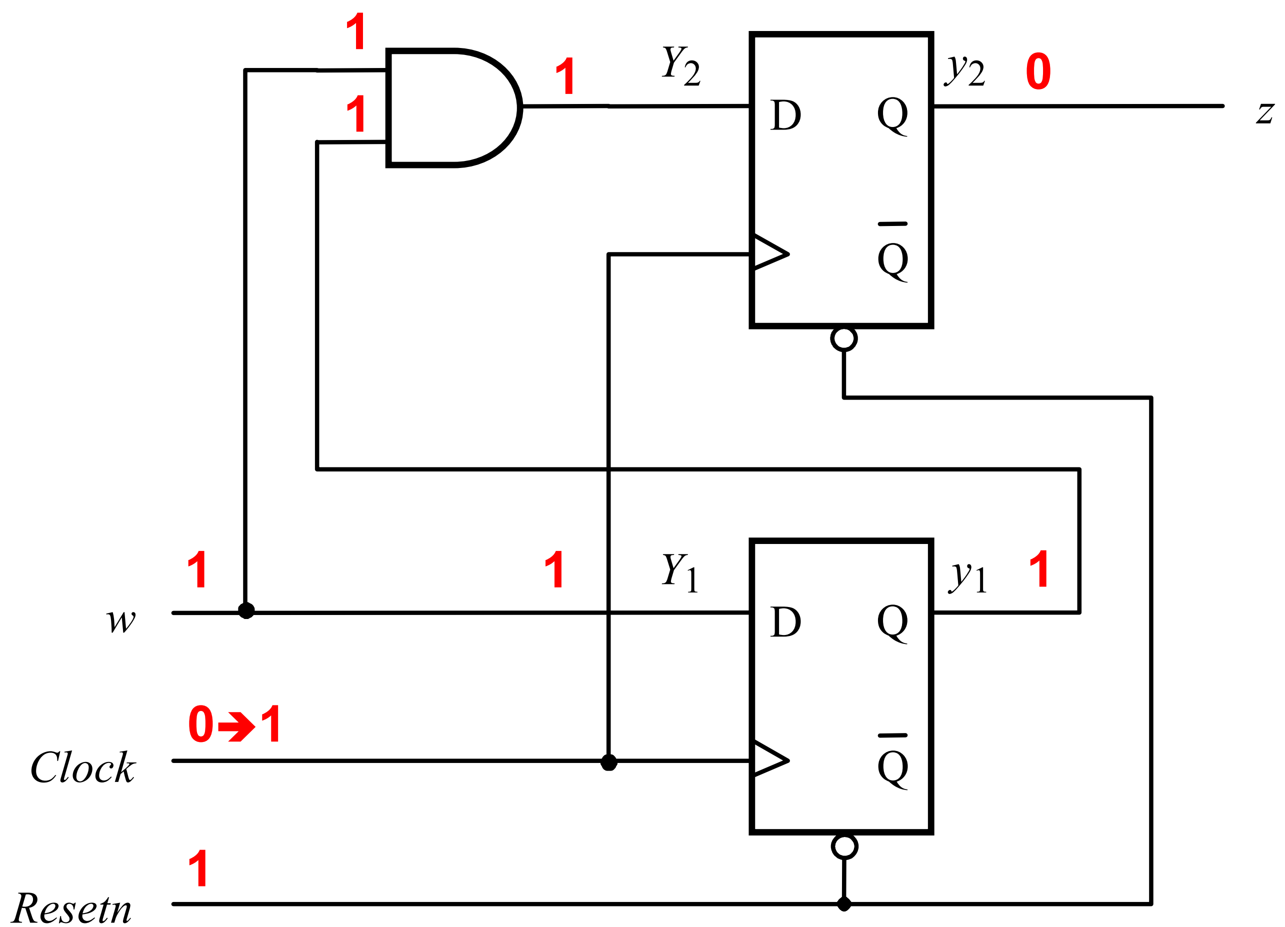

t0 t1 t2 t3 t4 t5
Clock
1
0
w
1
0


1
0
0
Y2
D
Q
Q̄
y2
0
z
1
1
Y1
D
Q
Q̄
y1
0
w
0
Clock
1
Resetn

State B=01

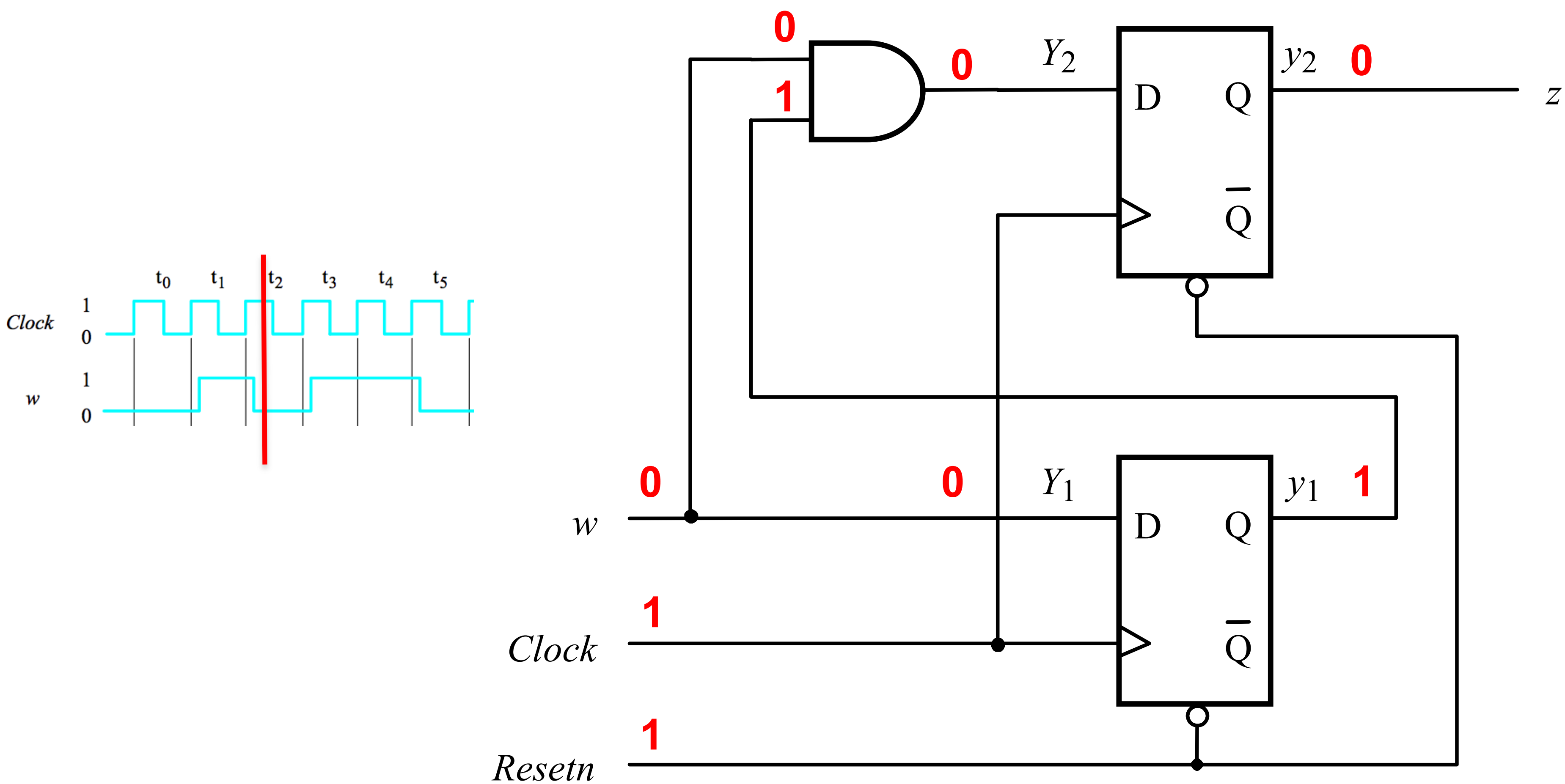


t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0



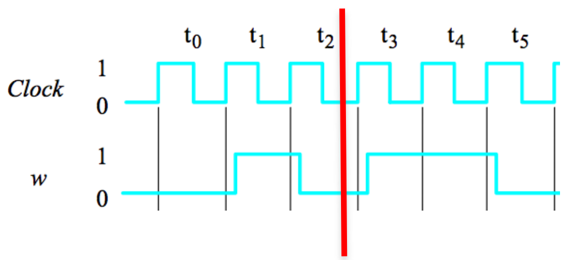
1
1
1
Y2
y2
0
D
Q
Q
z
1
1
Y1
y1
1
w
D
Q
0➔1
Clock
Q
1
Resetn

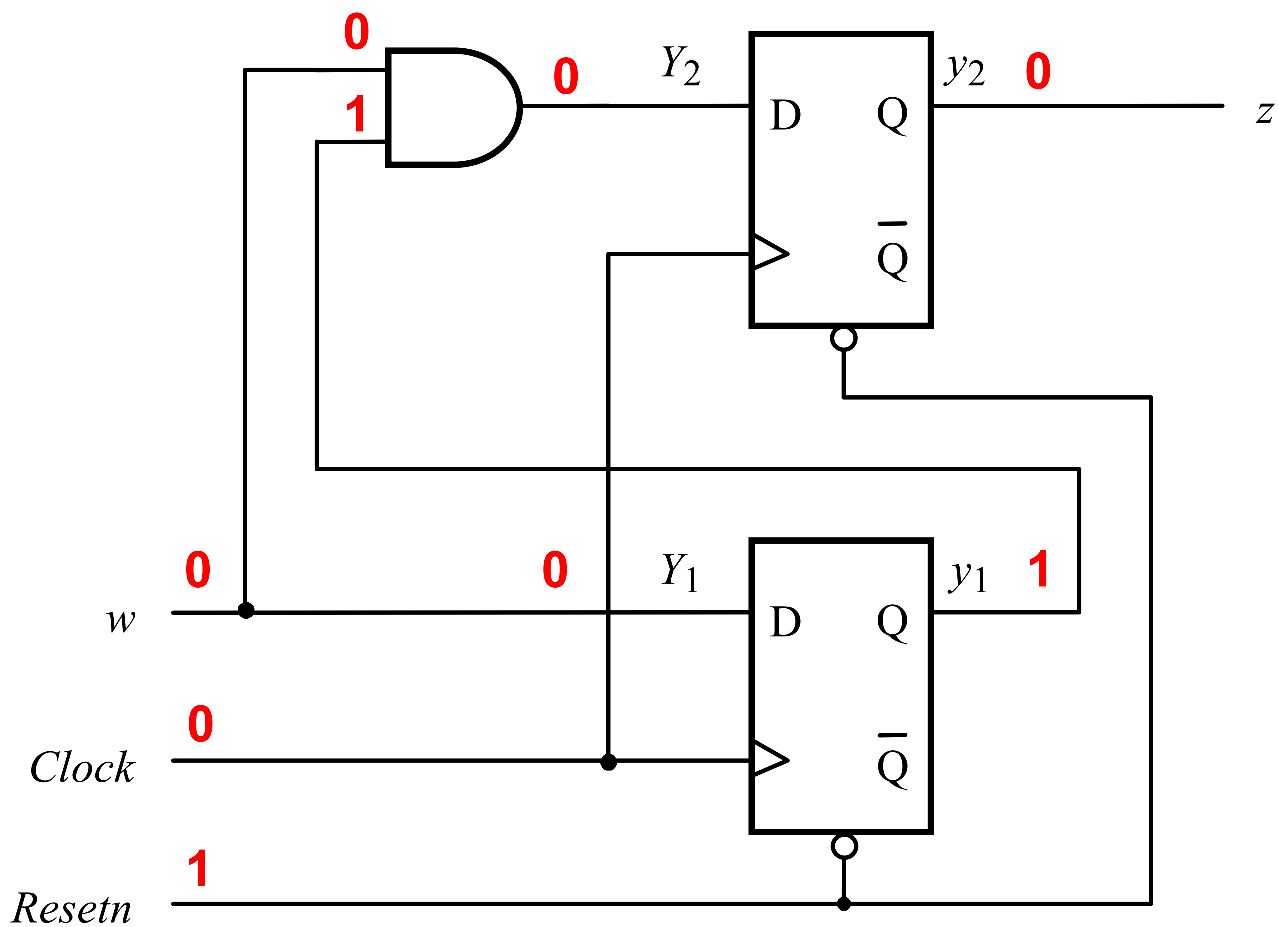
State B=01



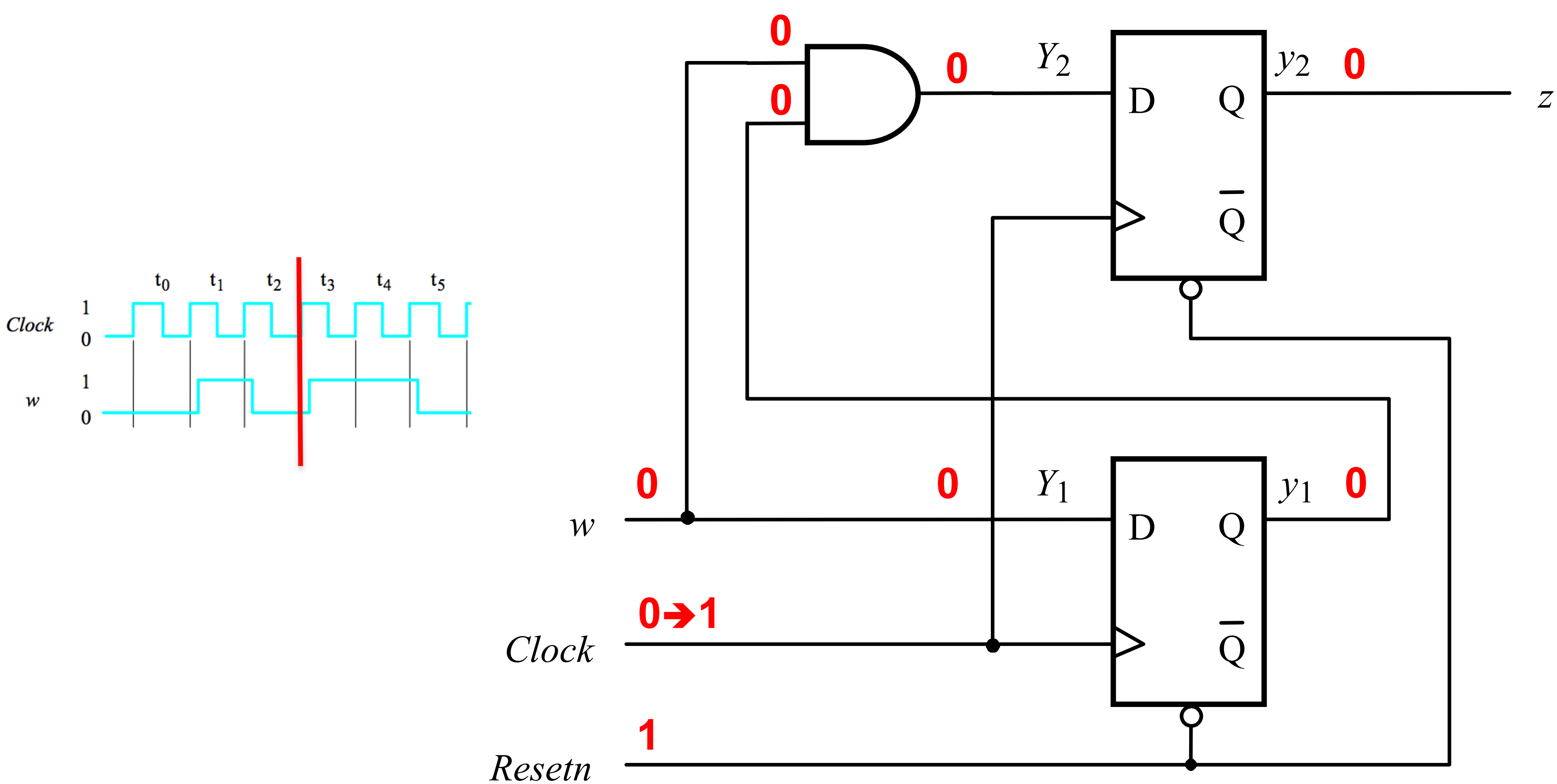
t0 t1 t2 t3 t4 t5
Clock
1
0
w
1
0
0
1
0
Y2
y2
0
D
Q
Q
z
0
0
Y1
y1
1
w
D
Q
1
Clock
Q
1
Resetn

State B=01


t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0

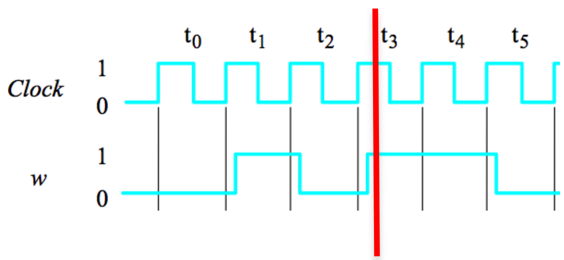

0
1
0
Y2
D
Q
Q
y2
0
z
0
0
Y1
D
Q
Q
y1
1
w
0
Clock
1
Resetn

State A=00

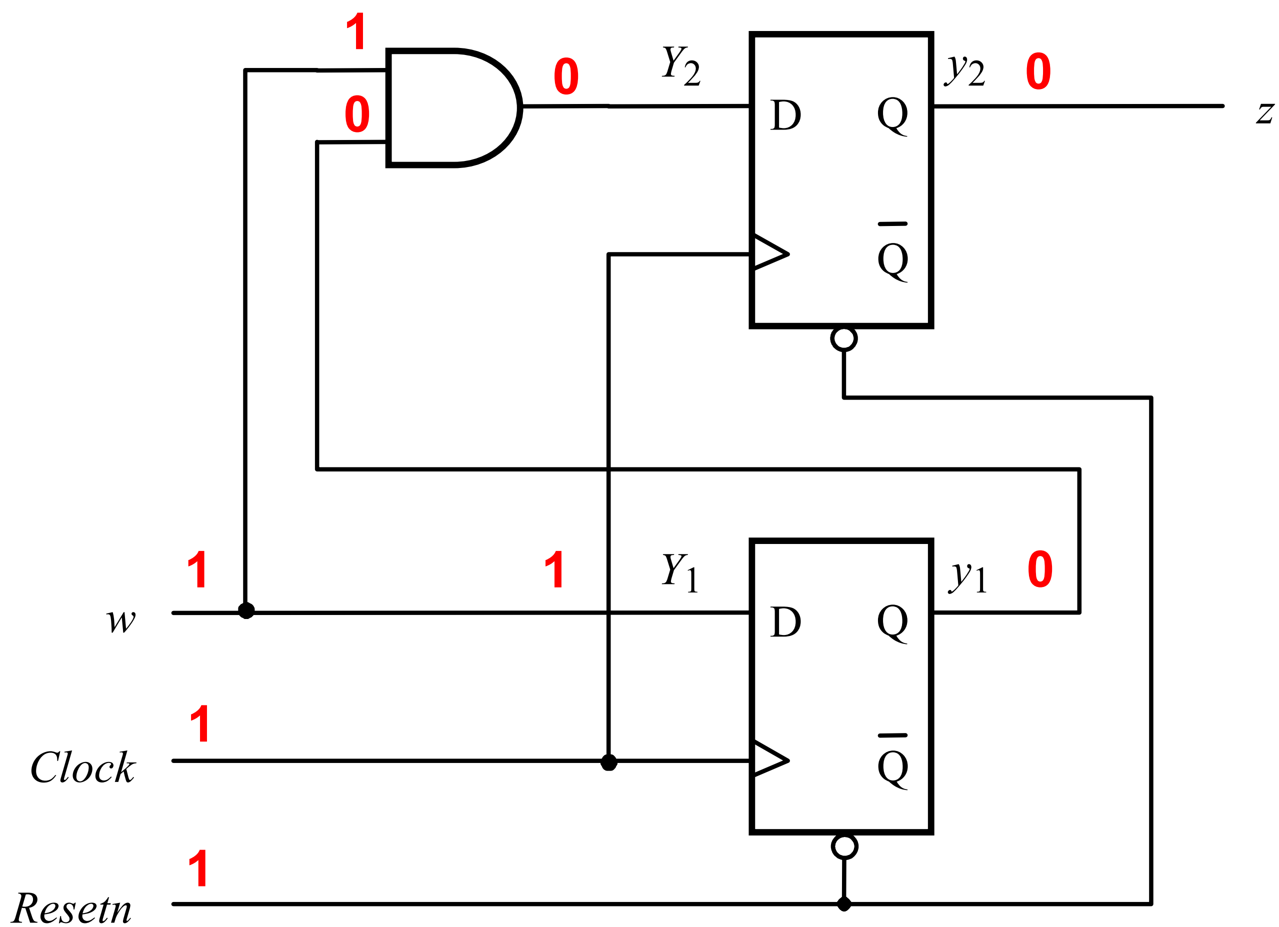


t0 t1 t2 t3 t4 t5
Clock
w
1
0
0
0
0
Y2
y2
z
D
Q
Q
0
0
Y1
y1
0→1
Clock
1
Resetn

State A=00

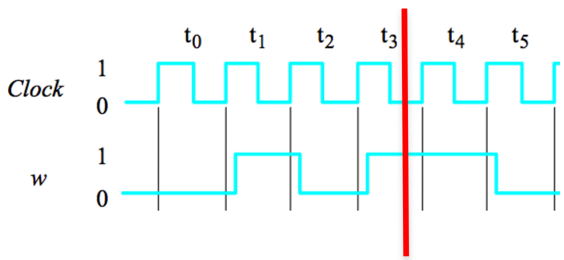


t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0

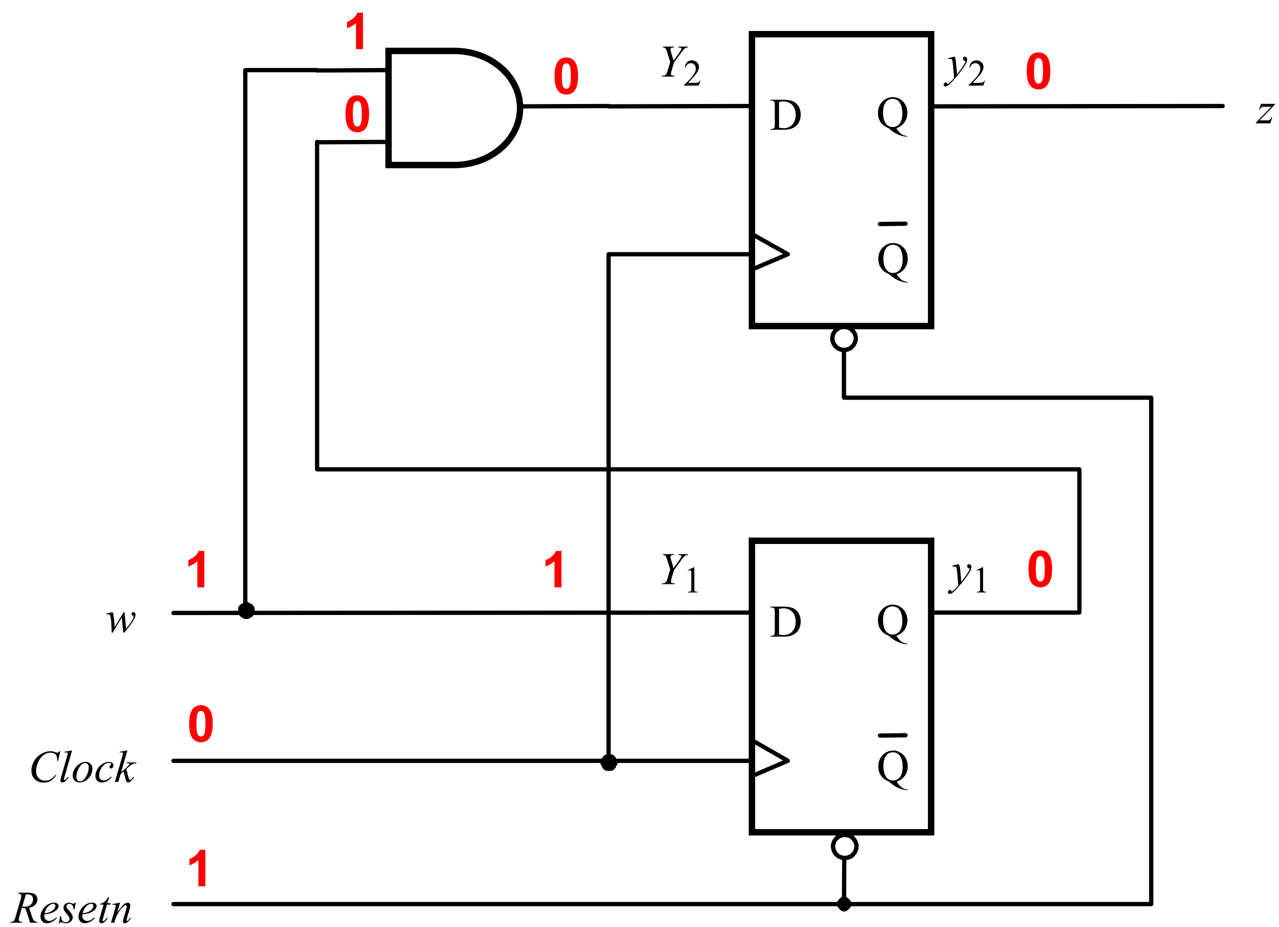


1
0
0
Y2
y2
0
D
Q
Q
z
1
1
Y1
y1
0
D
Q
Q
w
1
Clock
1
Resetn

State A=00

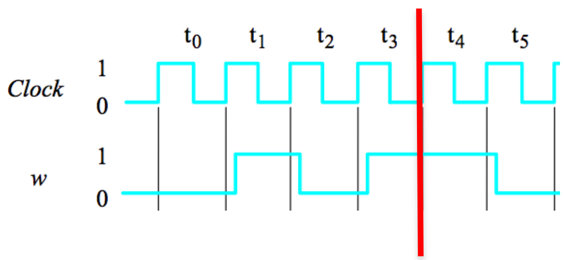


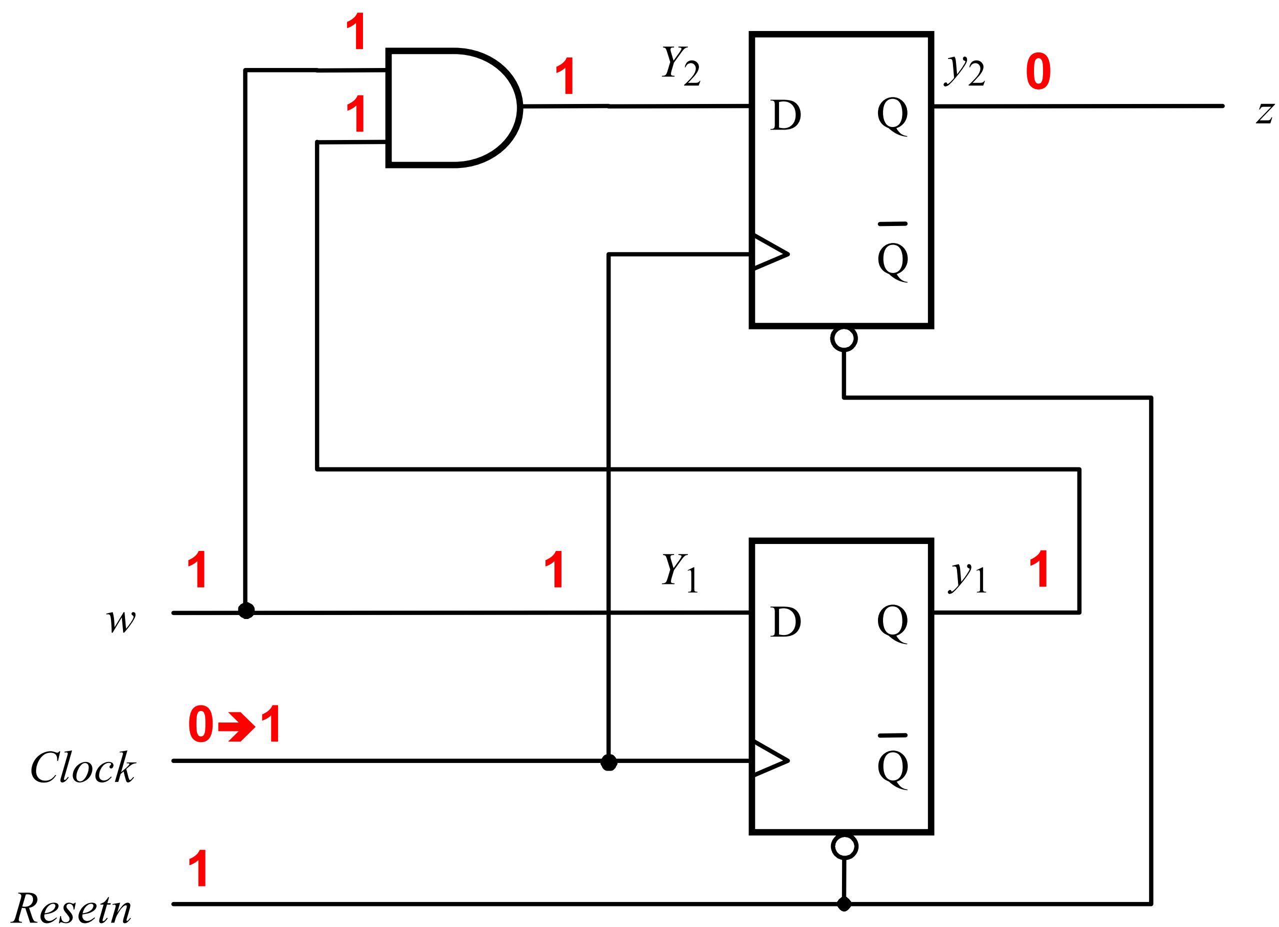
t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0



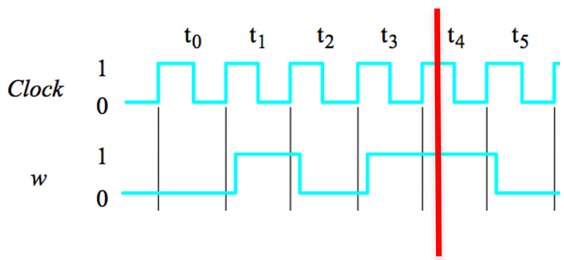
1
0
0
Y2
y2
0
D
Q
Q
z
w
1
1
Y1
y1
0
D
Q
Q
Clock
0
Resetn
1

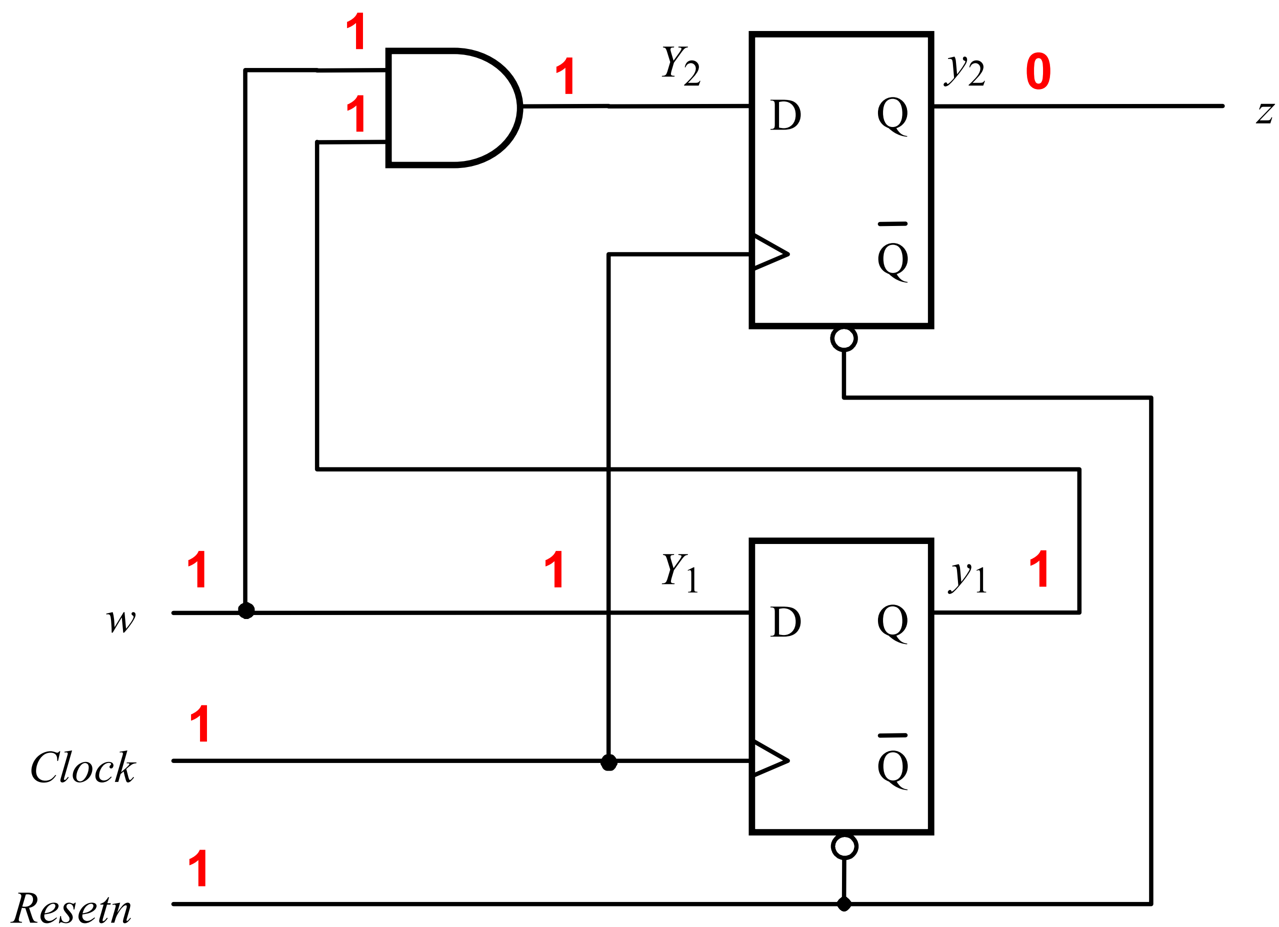
State B=01


t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0

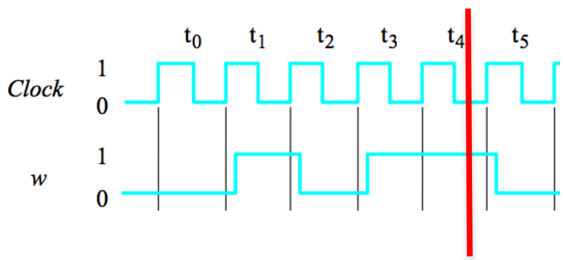

1
1
1
Y2
y2
0
z
D
Q
Q
w
1
1
Y1
y1
1
D
Q
Q
Clock
0➔1
Resetn
1

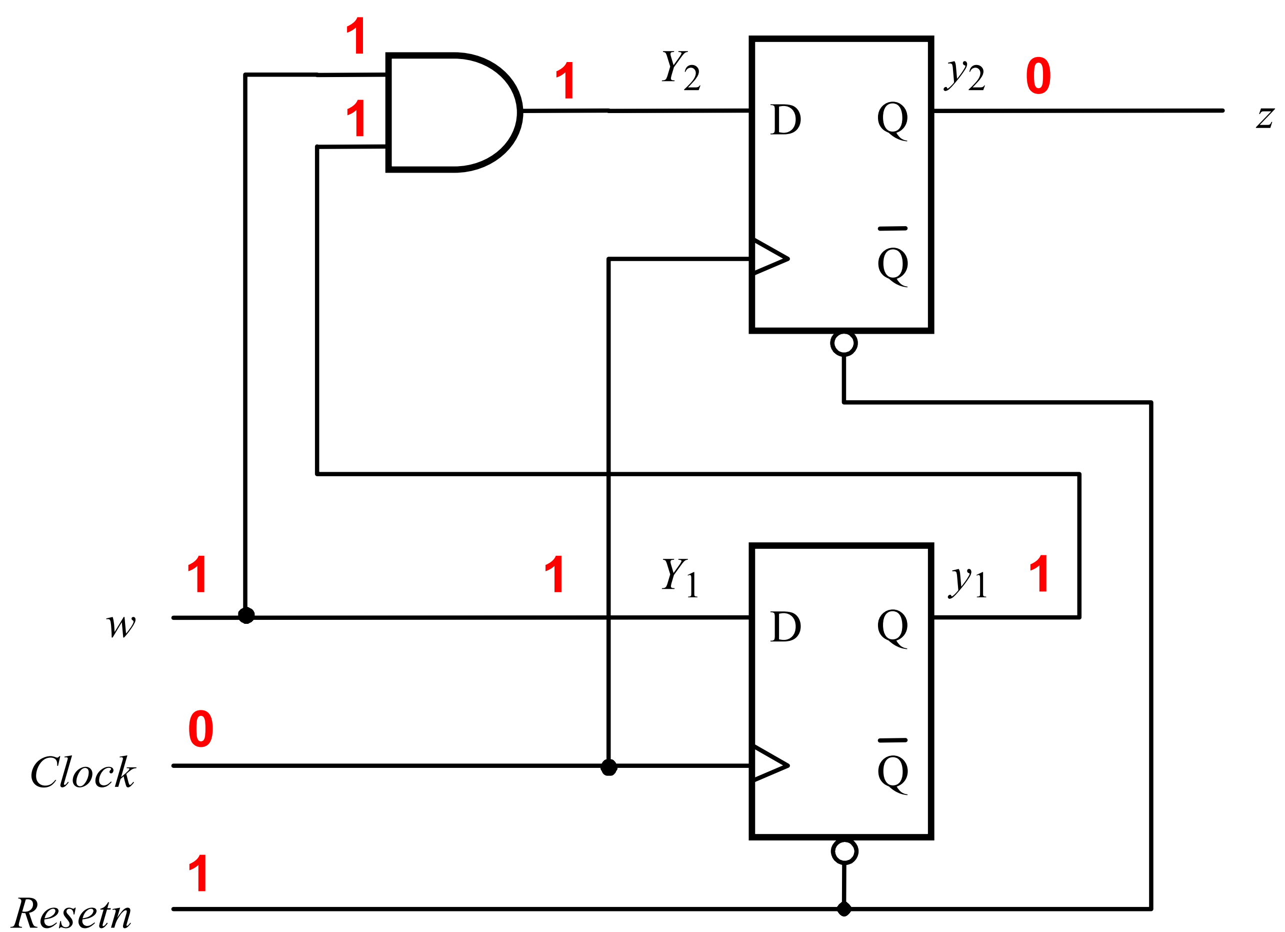
State B=01


t0 t1 t2 t3 t4 t5
Clock
1
0
w
1
0

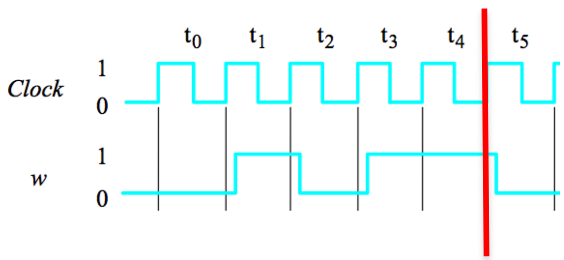

1
1
1
Y2
y2
0
D
Q
Q
z
1
1
Y1
y1
1
w
D
Q
1
Clock
Q
1
Resetn

State B=01

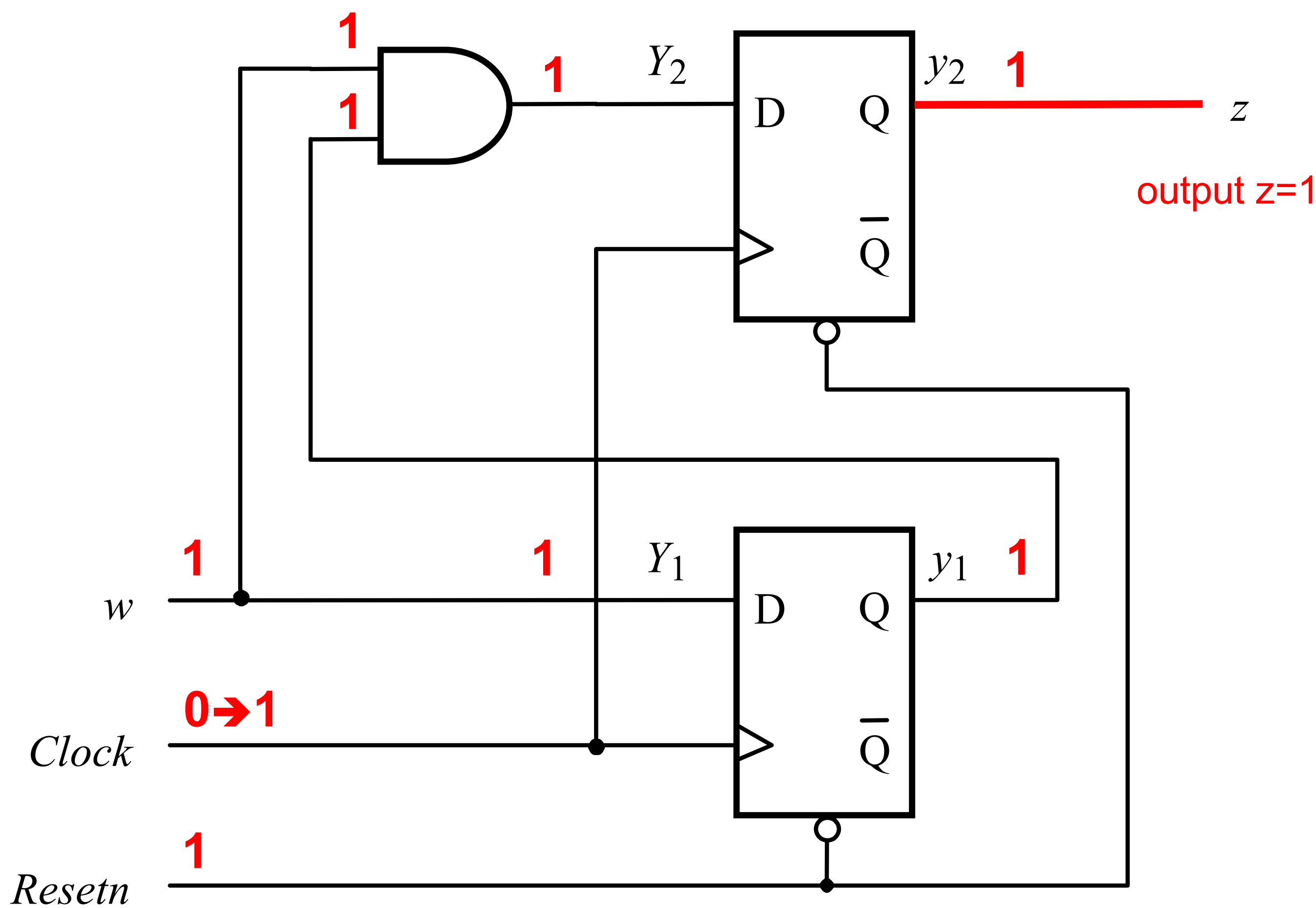

t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0


1
1
1
Y2
D
Q
y2
0
z
Q
1
1
Y1
D
Q
y1
1
w
0
Clock
Q
1
Resetn

State C=11

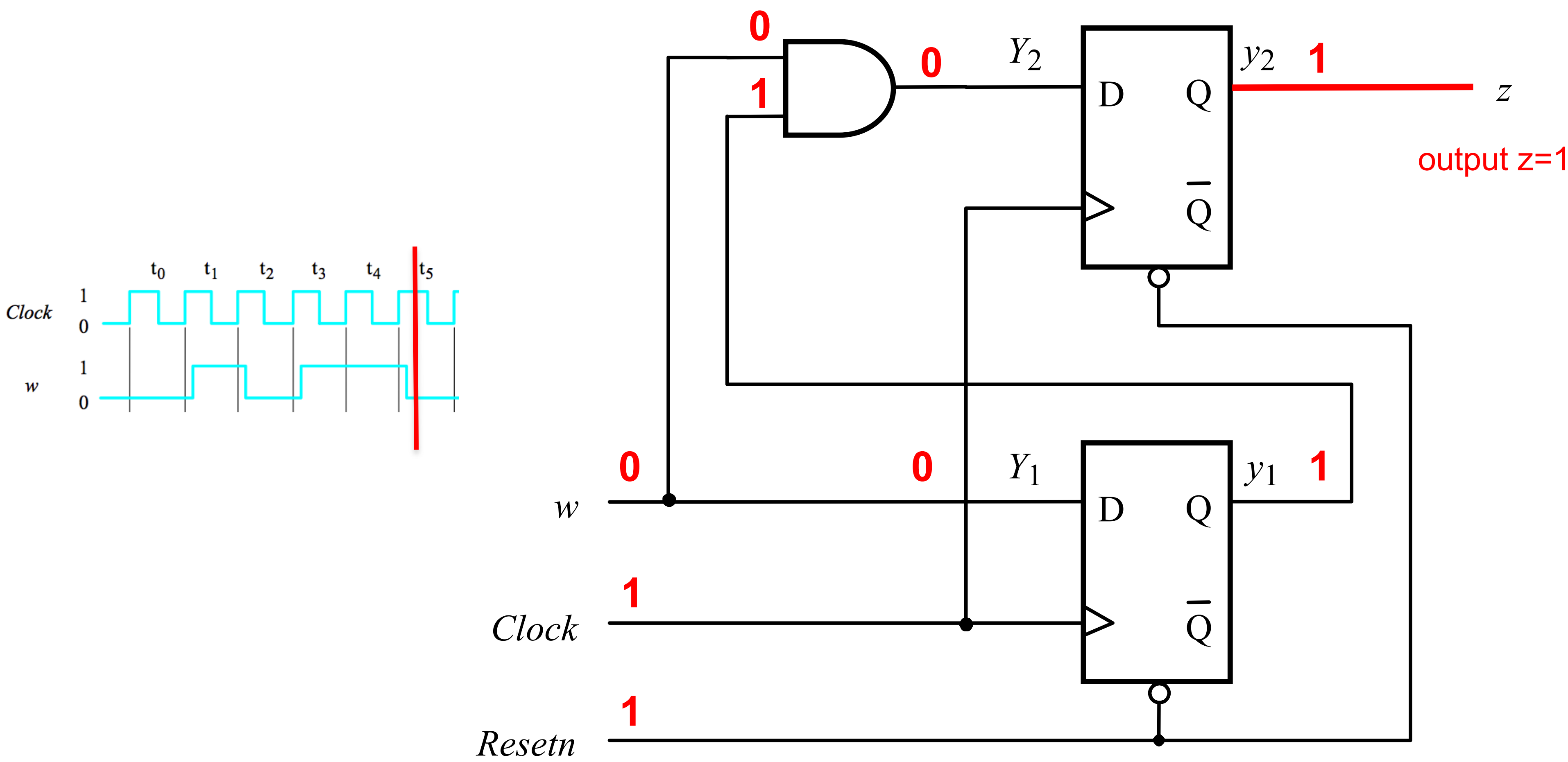


t0 t1 t2 t3 t4 t5
Clock
1
0
w
1
0



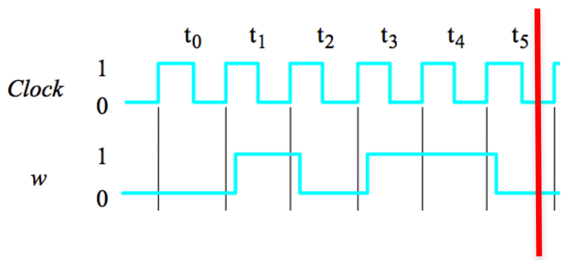
1
1
1
Y2
y2
1
D
Q
Q
z
output z=1
1
1
Y1
y1
1
w
D
Q
Q
Clock
0➔1
Resetn
1

State C=11

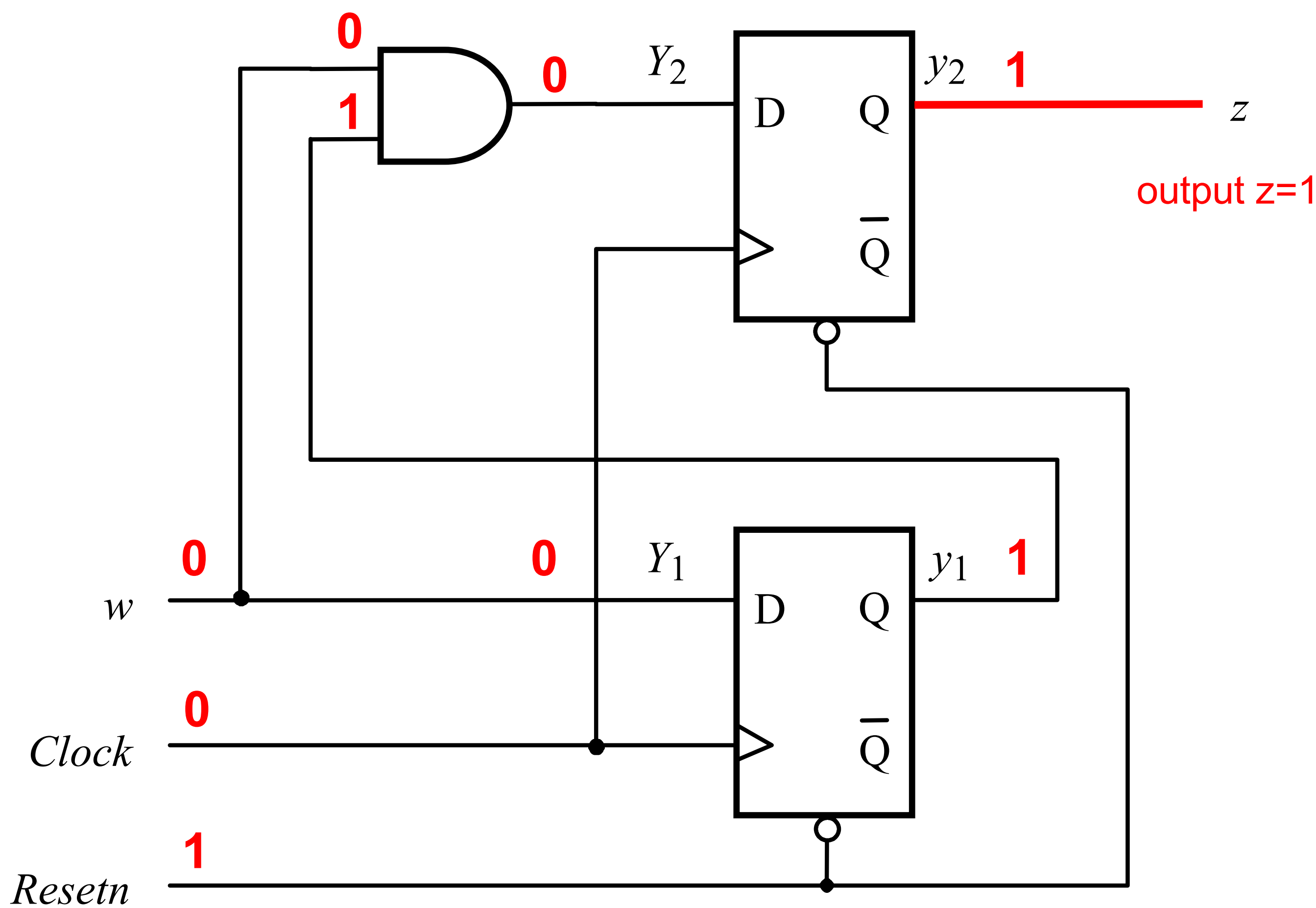


t0 t1 t2 t3 t4 t5
Clock
1
0
w
1
0
0
1
0
Y2
D
Q
y2
1
z
output z=1
Q
0
0
Y1
D
Q
y1
1
w
1
Clock
Q
1
Resetn

State C=11

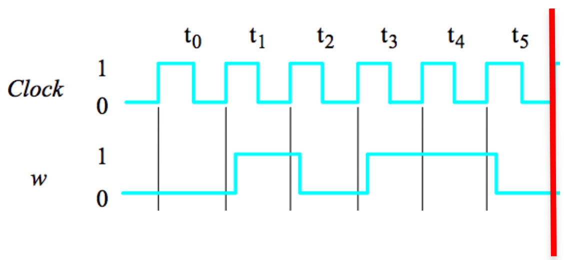


t0
t1
t2
t3
t4
t5
Clock
1
0
w
1
0

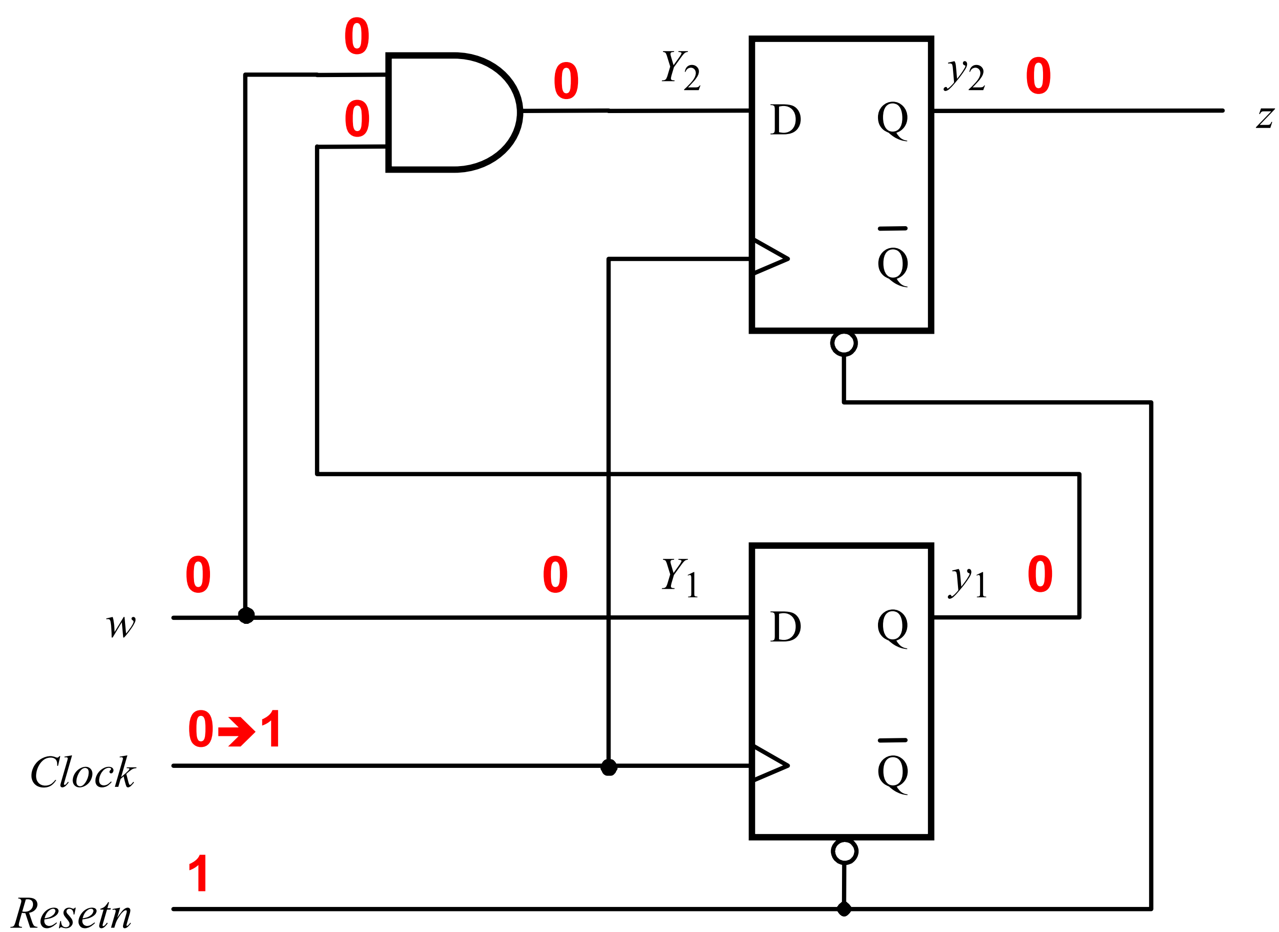


0
1
0
Y_2
D
Q
$\overline{Q}$
y_2
1
z
output z=1
0
0
Y_1
D
Q
$\overline{Q}$
y_1
1
w
0
Clock
1
Resetn

State A=00

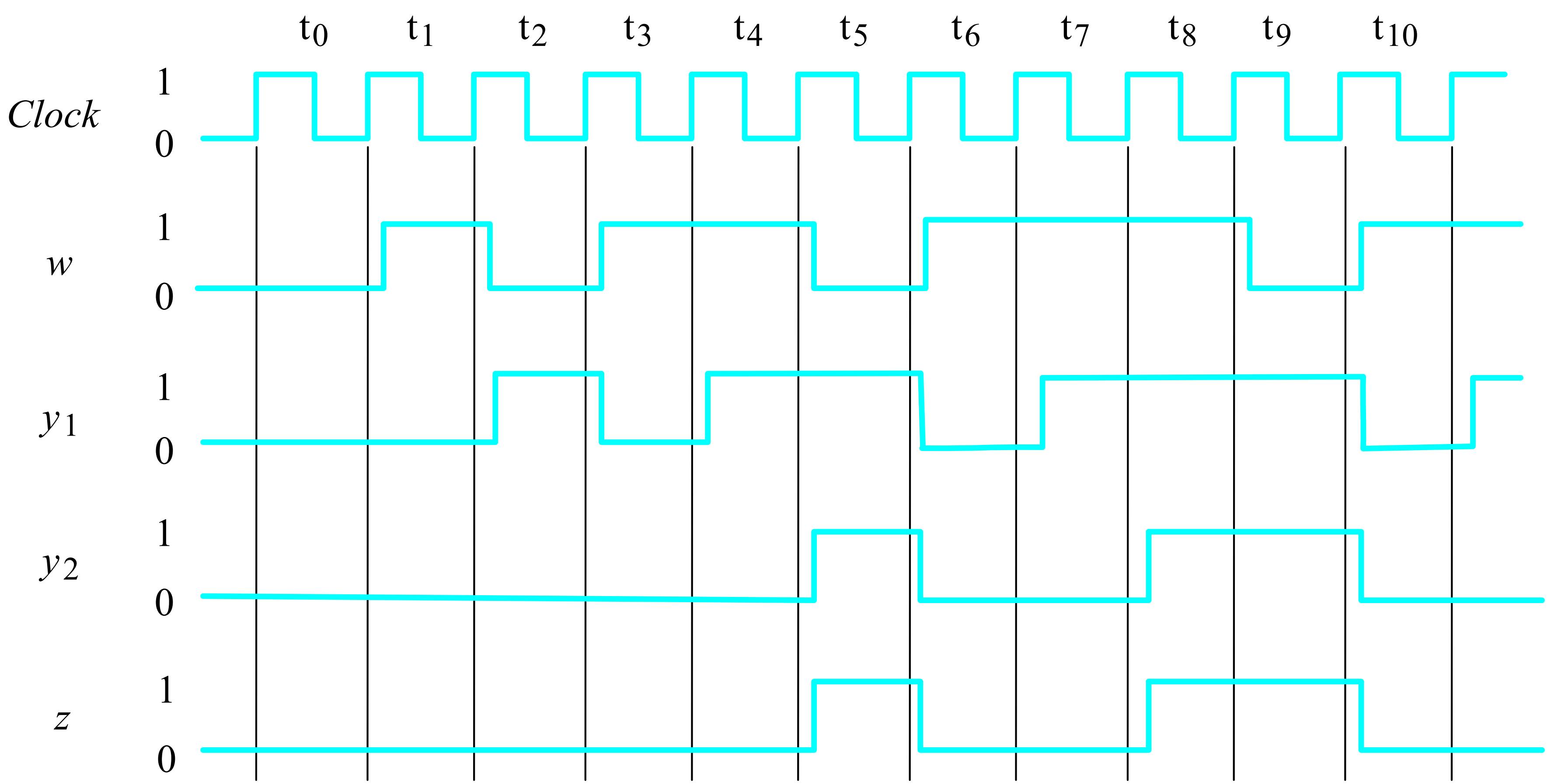


t0 t1 t2 t3 t4 t5
Clock
1
0
w
1
0

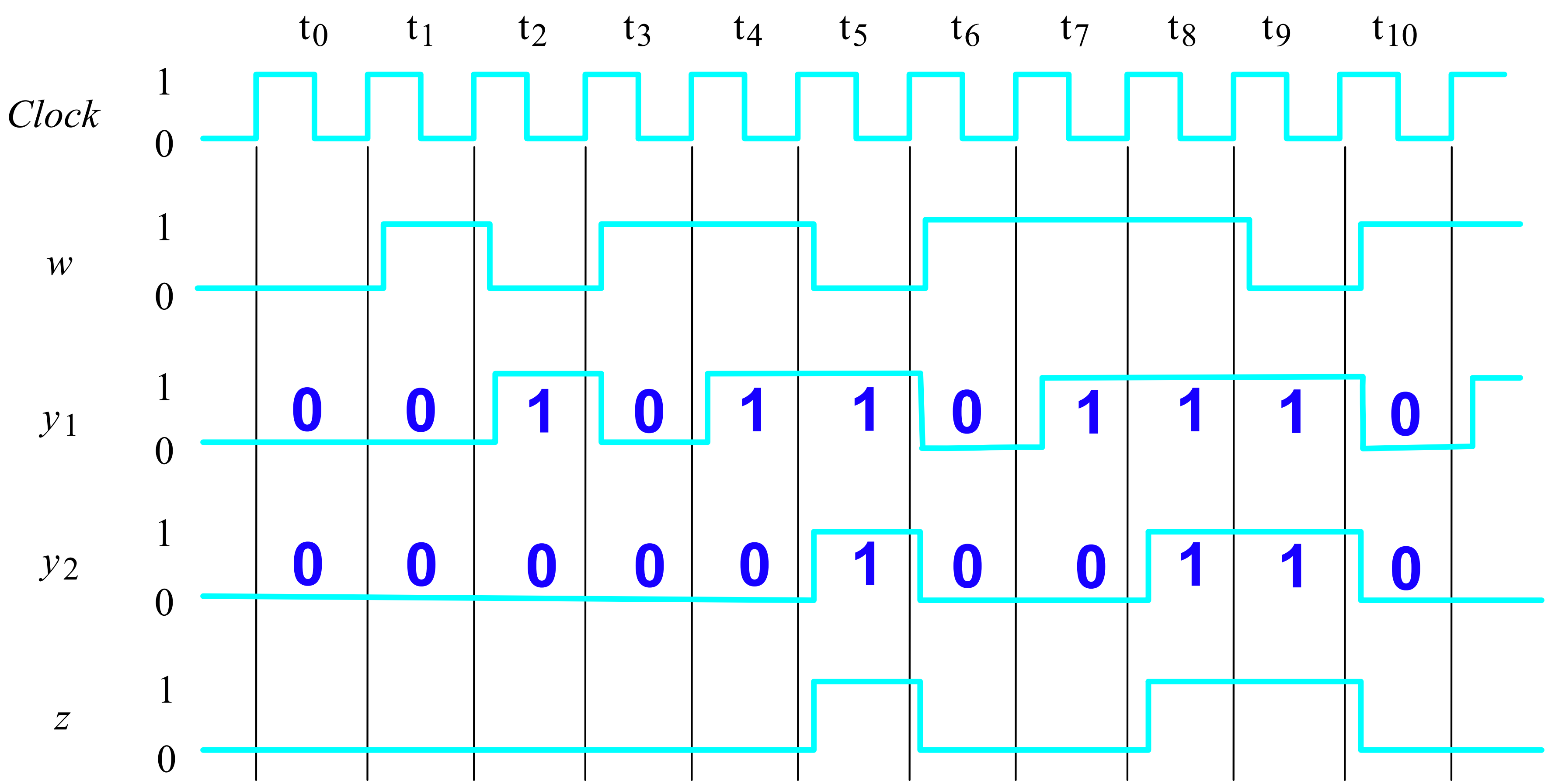


0
0
0
Y2
D
Q
Q
y2
0
z
0
0
Y1
D
Q
Q
y1
0
w
0→1
Clock
1
Resetn

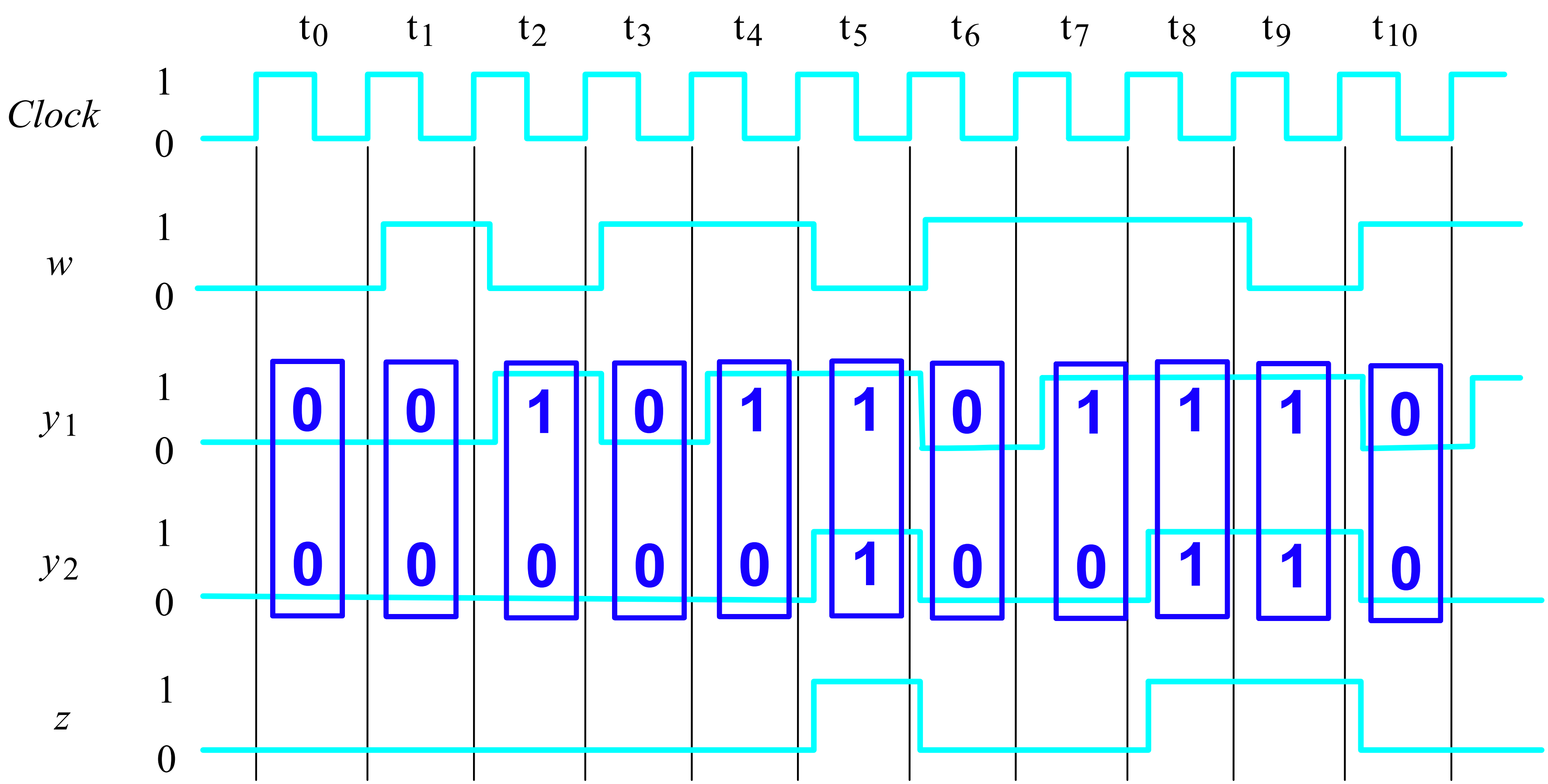
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0



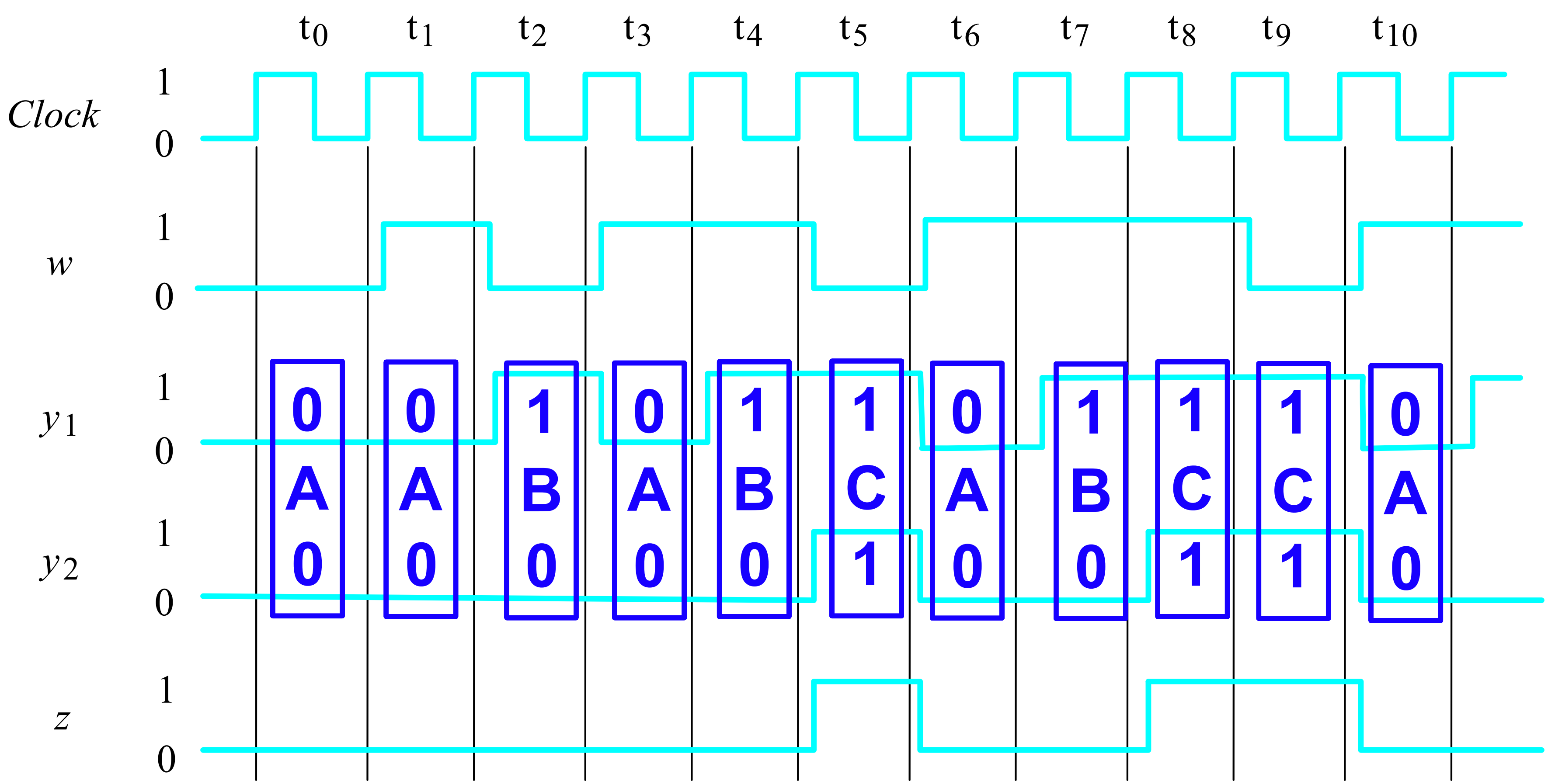
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0



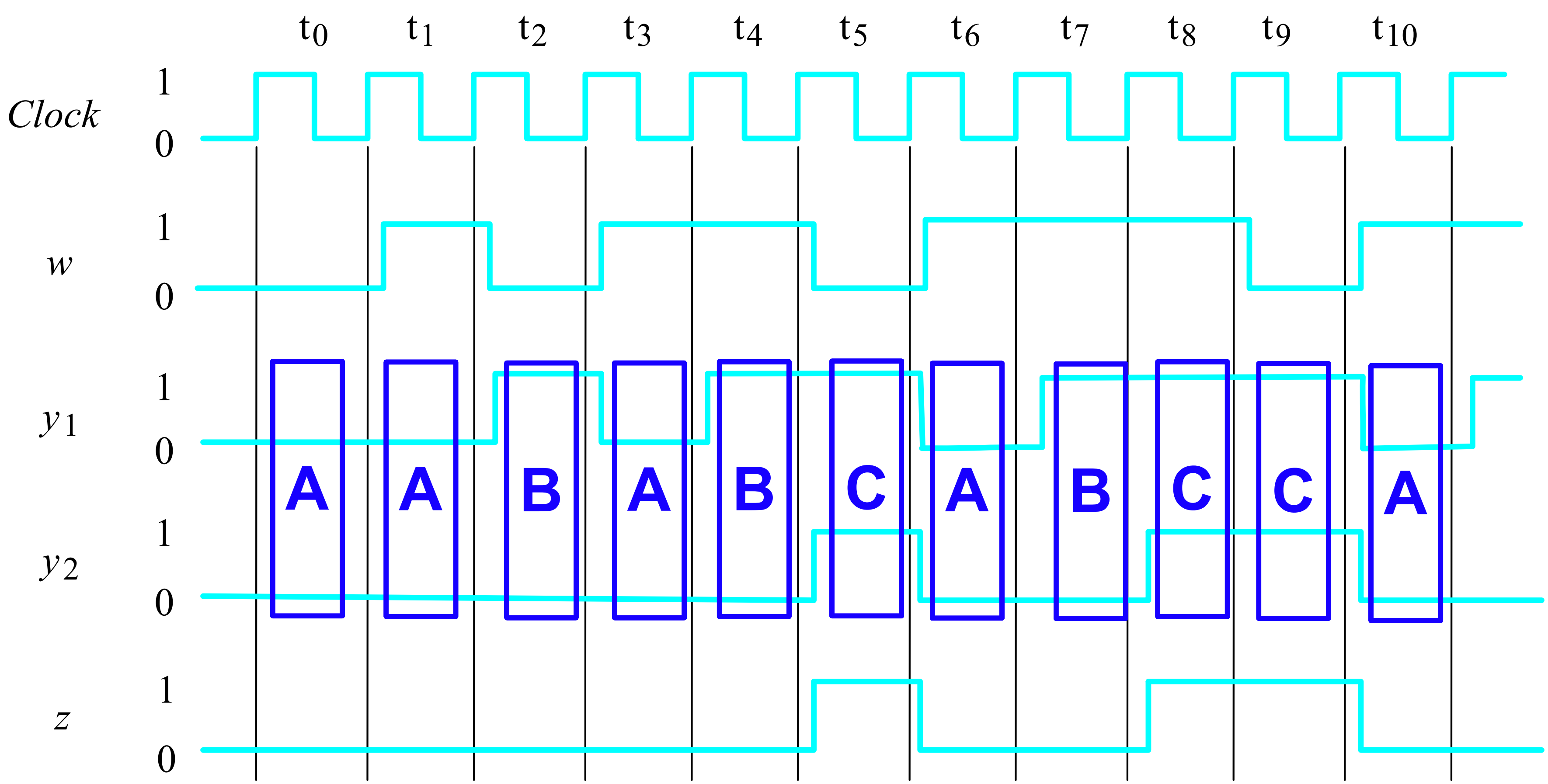
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w:	0	1	0	1	1	0	1	1	1	0	1
z:	0	0	0	0	0	1	0	0	1	1	0



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



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



Main Idea

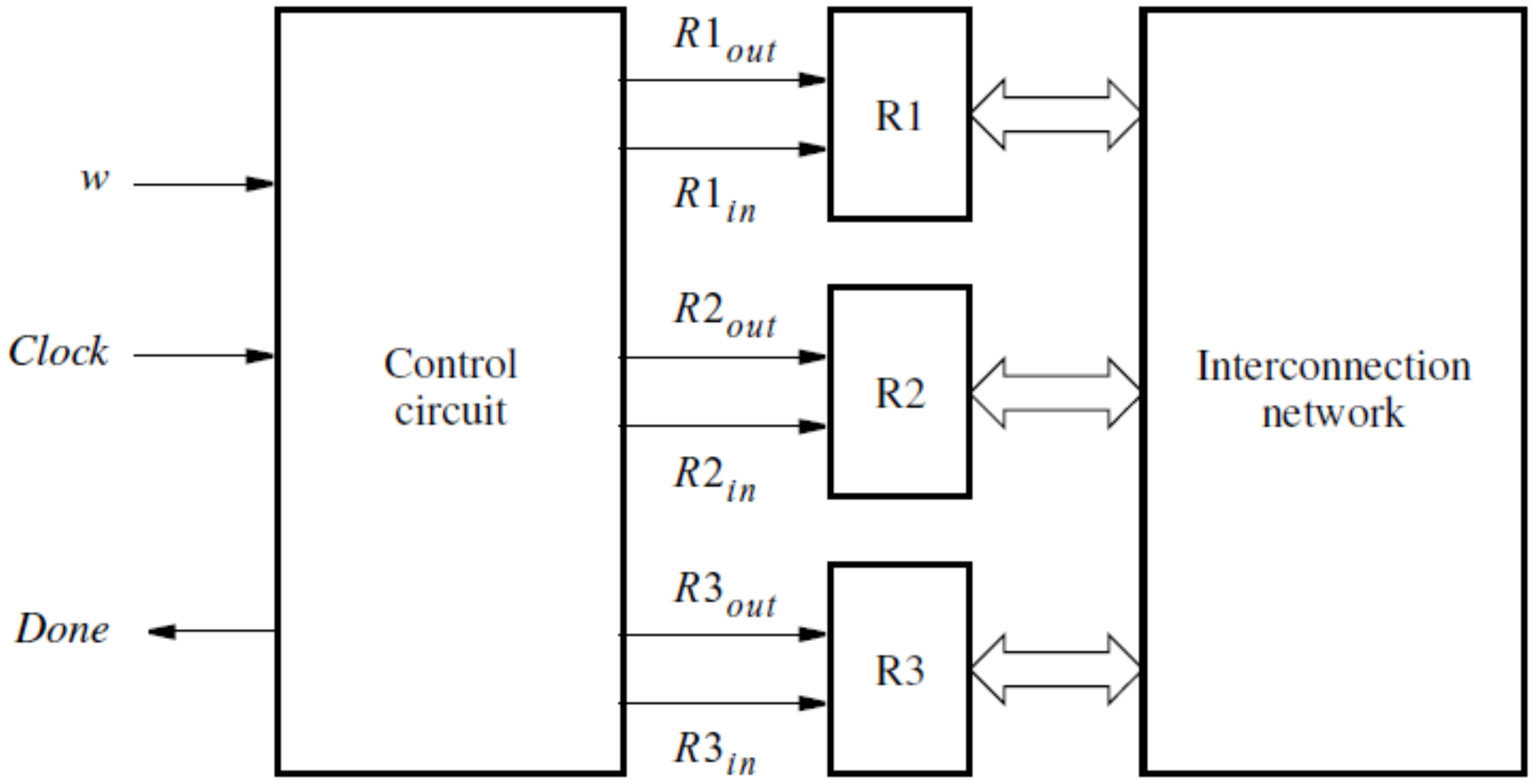
Different state assignments of the same Moore machine generally lead to different circuits.

Some may use fewer logic elements than others.

But they all do the same thing
(i.e., implement the same Moore machine).

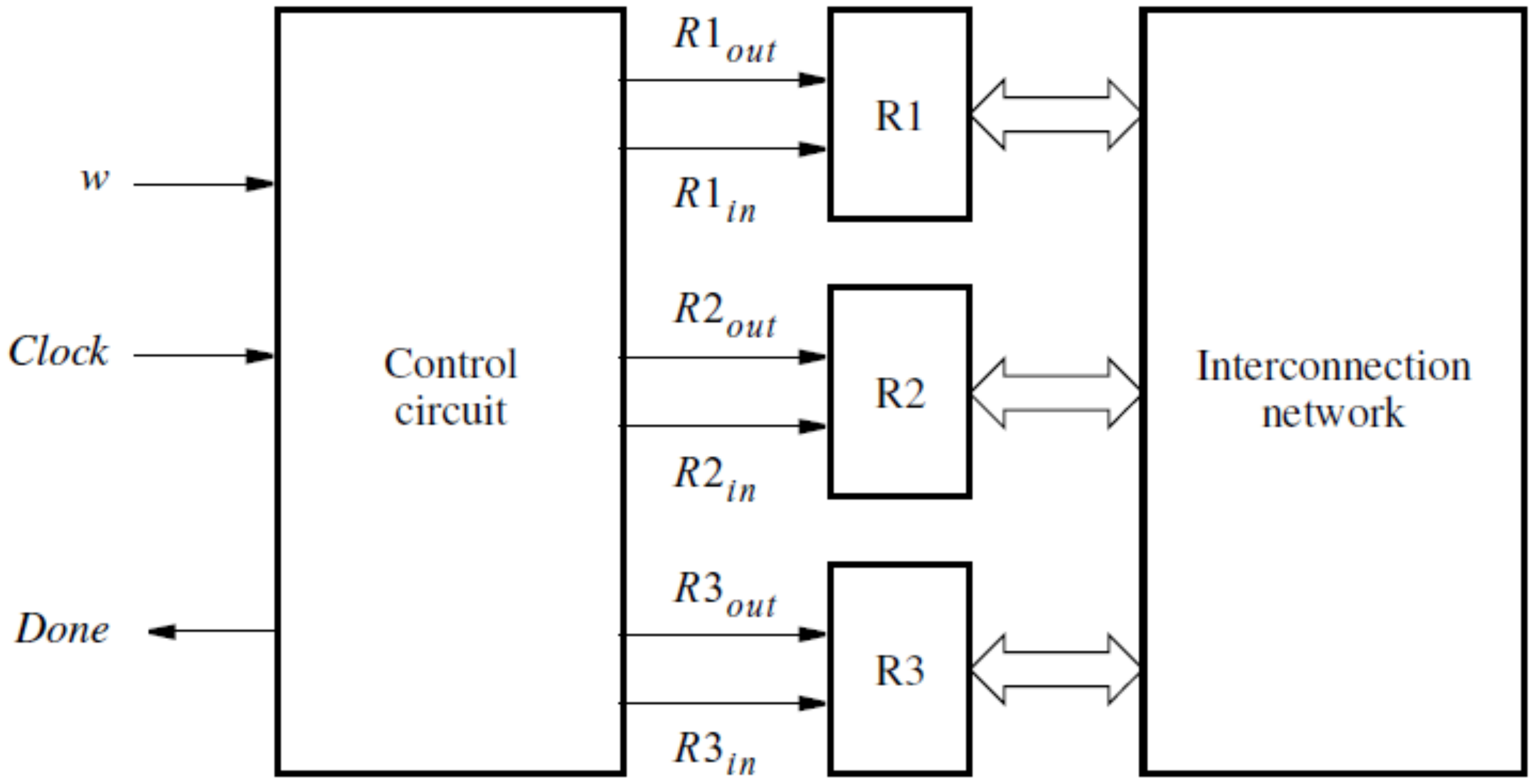
Example #2

Register Swap Controller



[Figure 6.10 from the textbook]

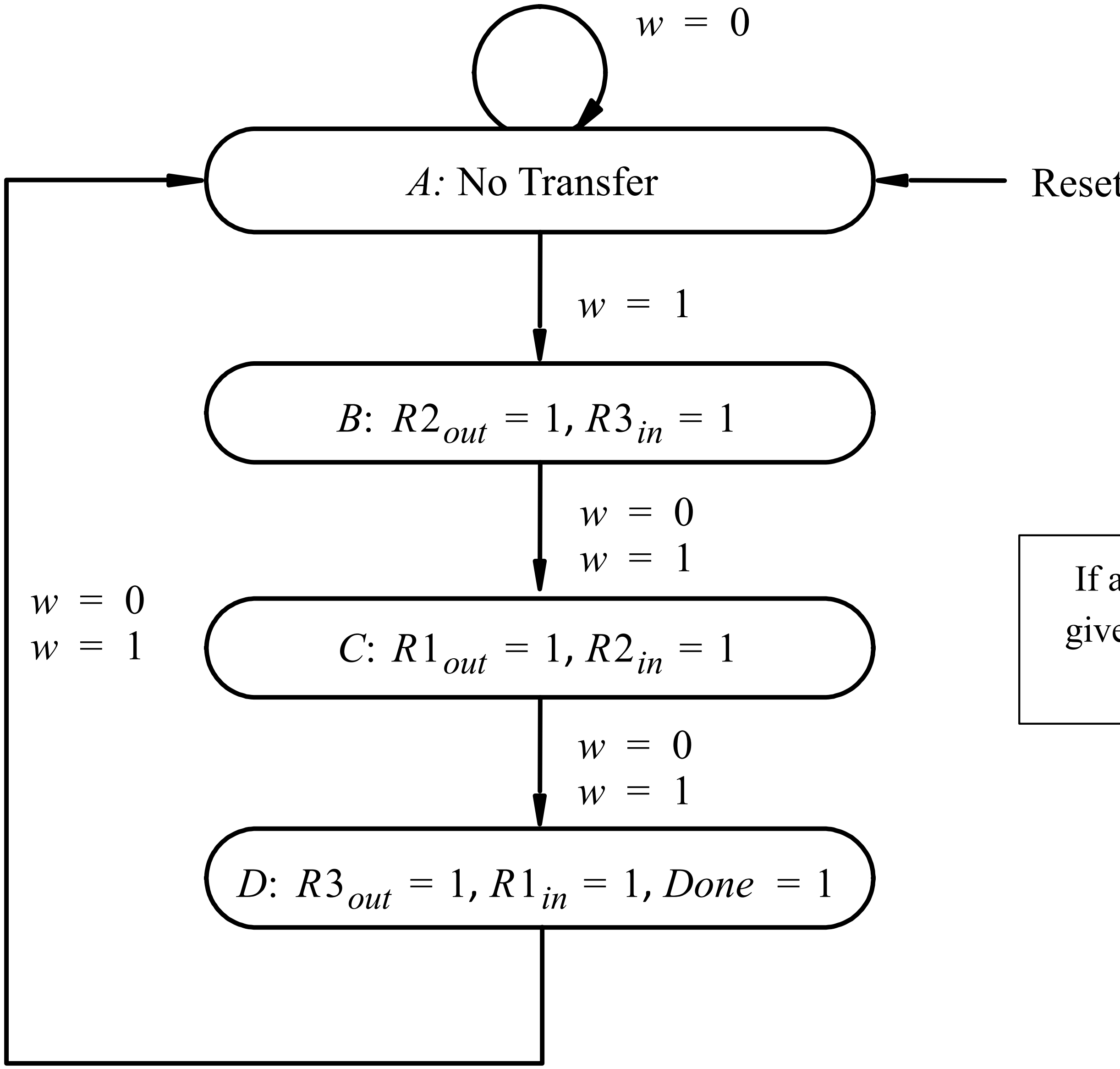
Register Swap Controller



Design a Moore machine control circuit for swapping the contents of registers R1 and R2 by using R3 as a temporary.

[Figure 6.10 from the textbook]

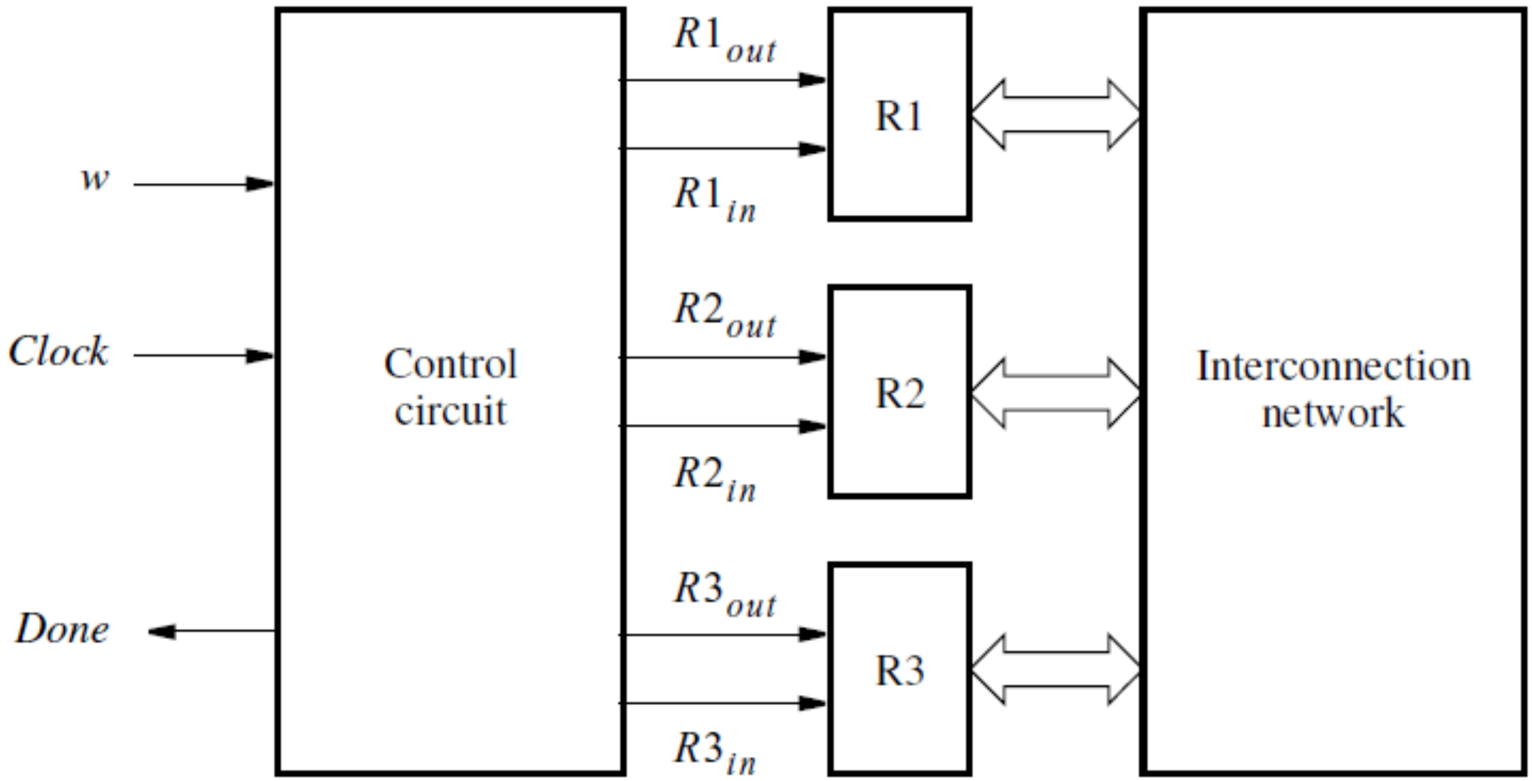
State Diagram



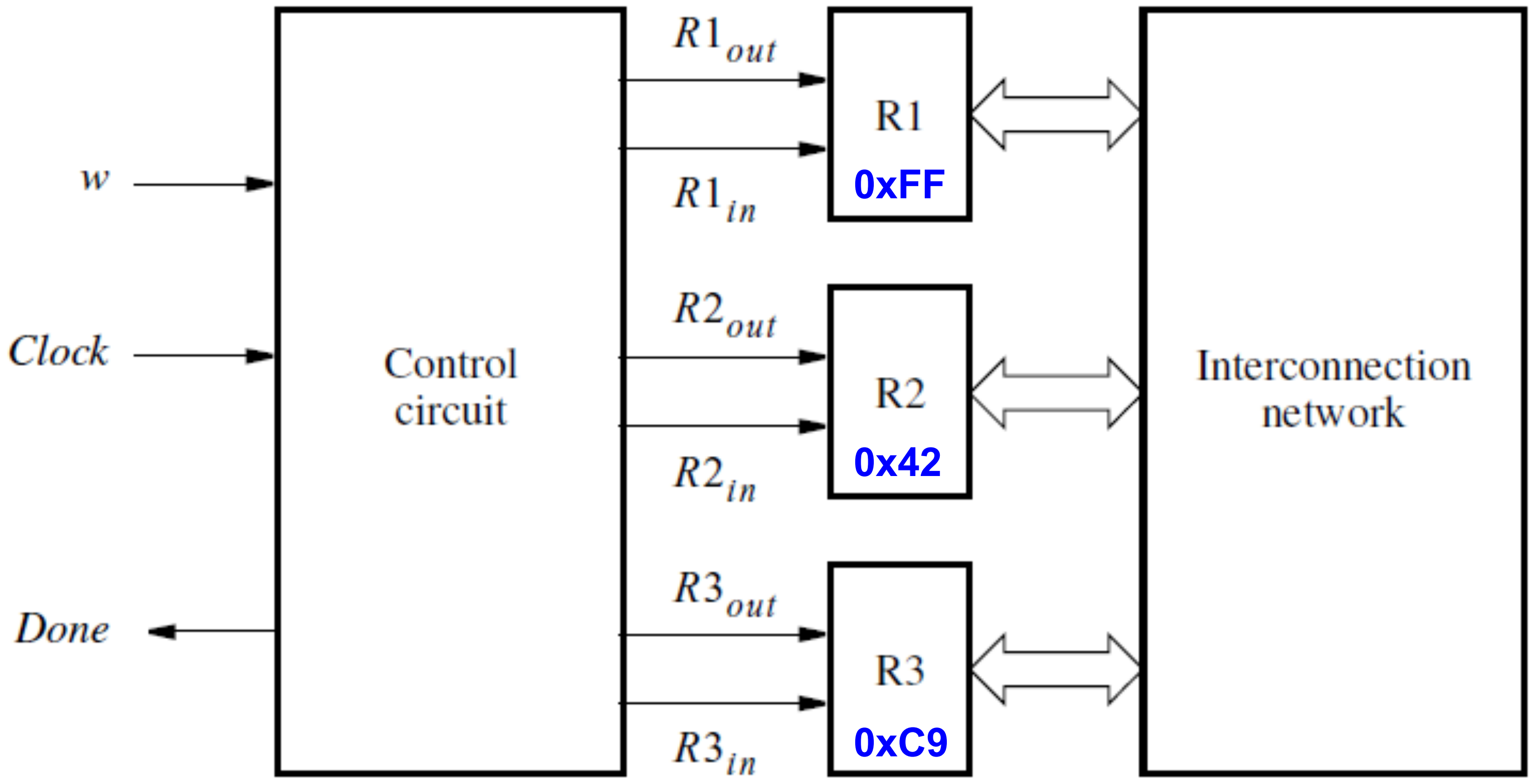
If an output is not given, then assume that it is 0.

[Figure 6.11 from the textbook]

Animated Register Swap

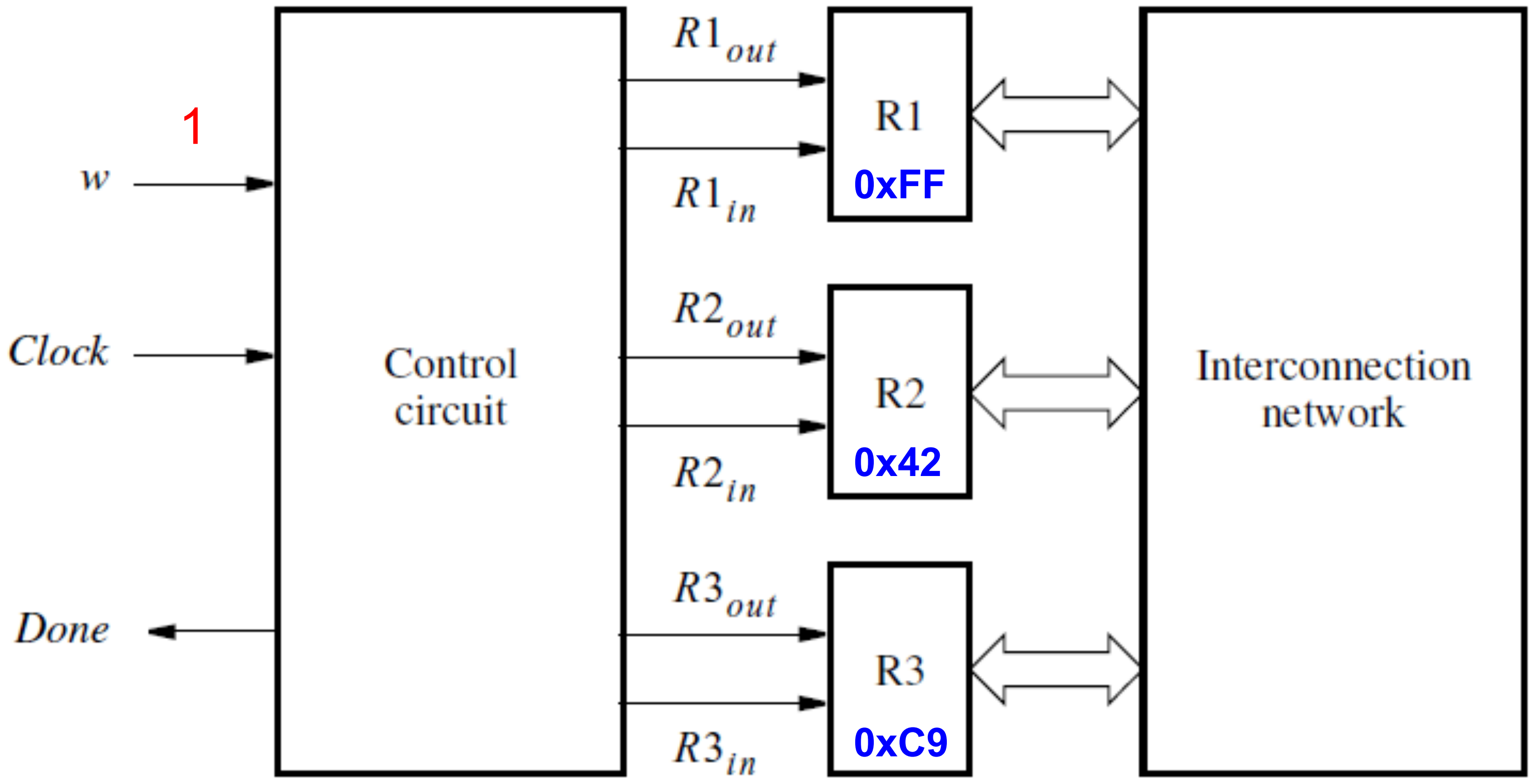


Animated Register Swap



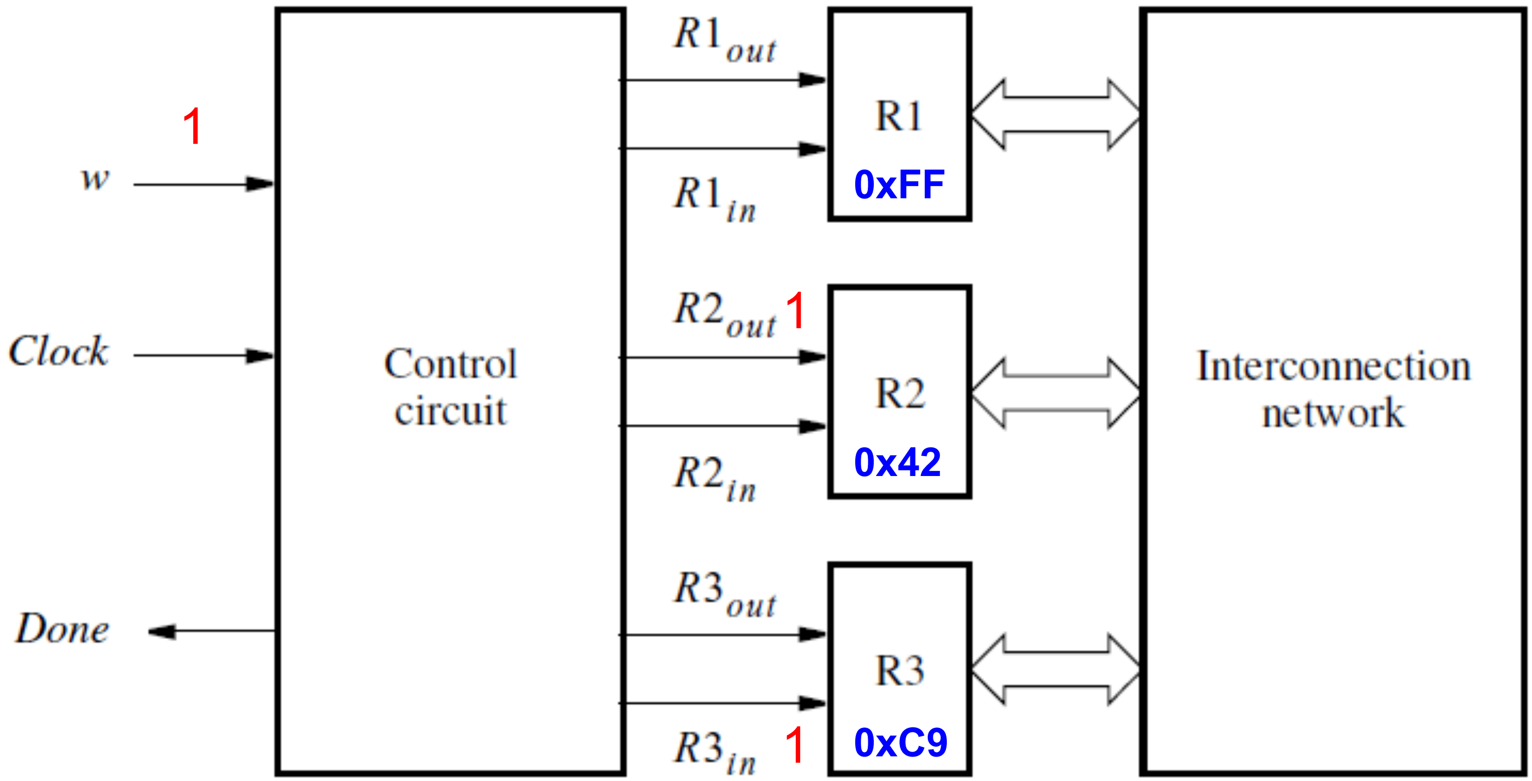
These are the original values of the 8-bit registers

Animated Register Swap

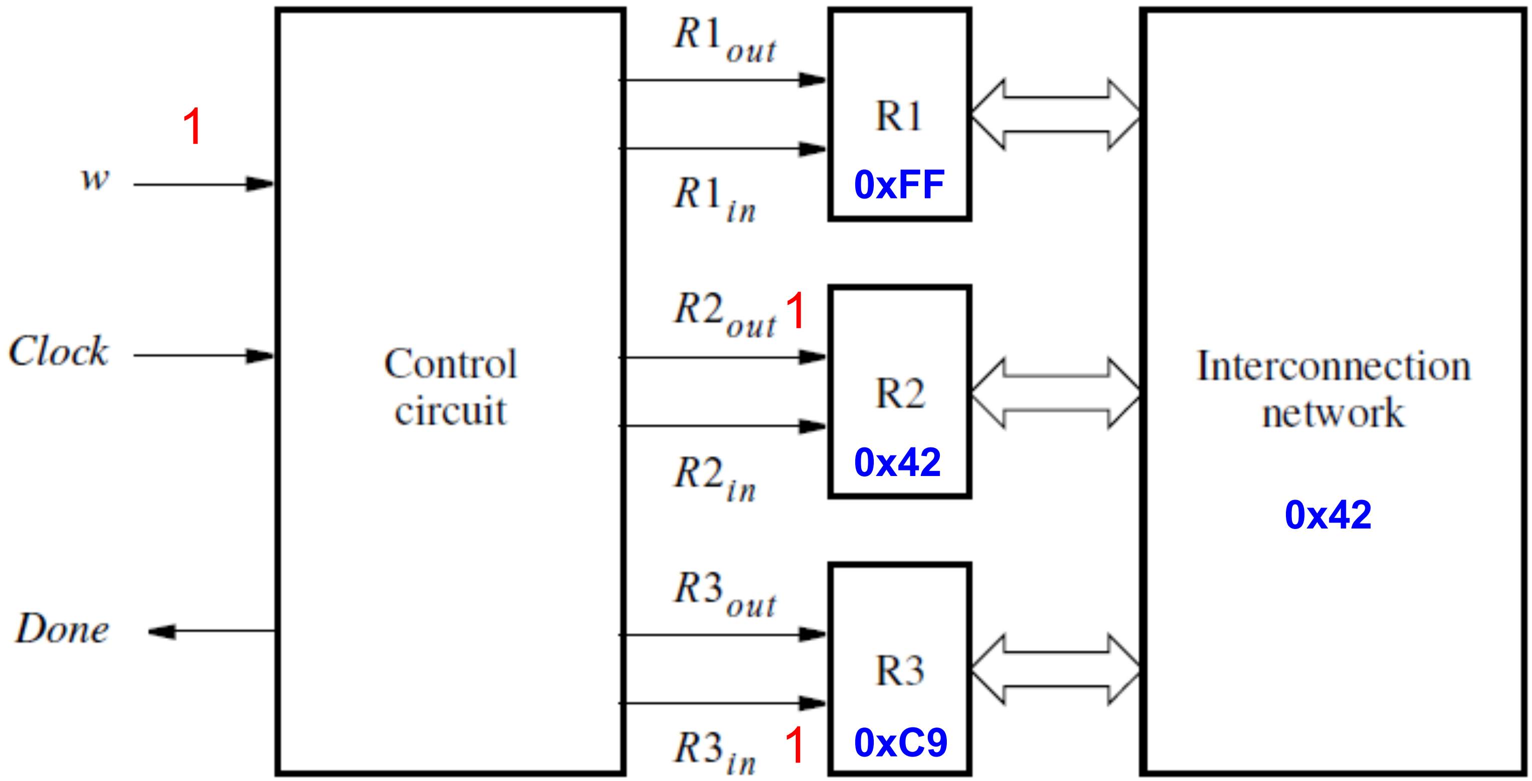


For clarity, only inputs that are equal to 1 will be shown.

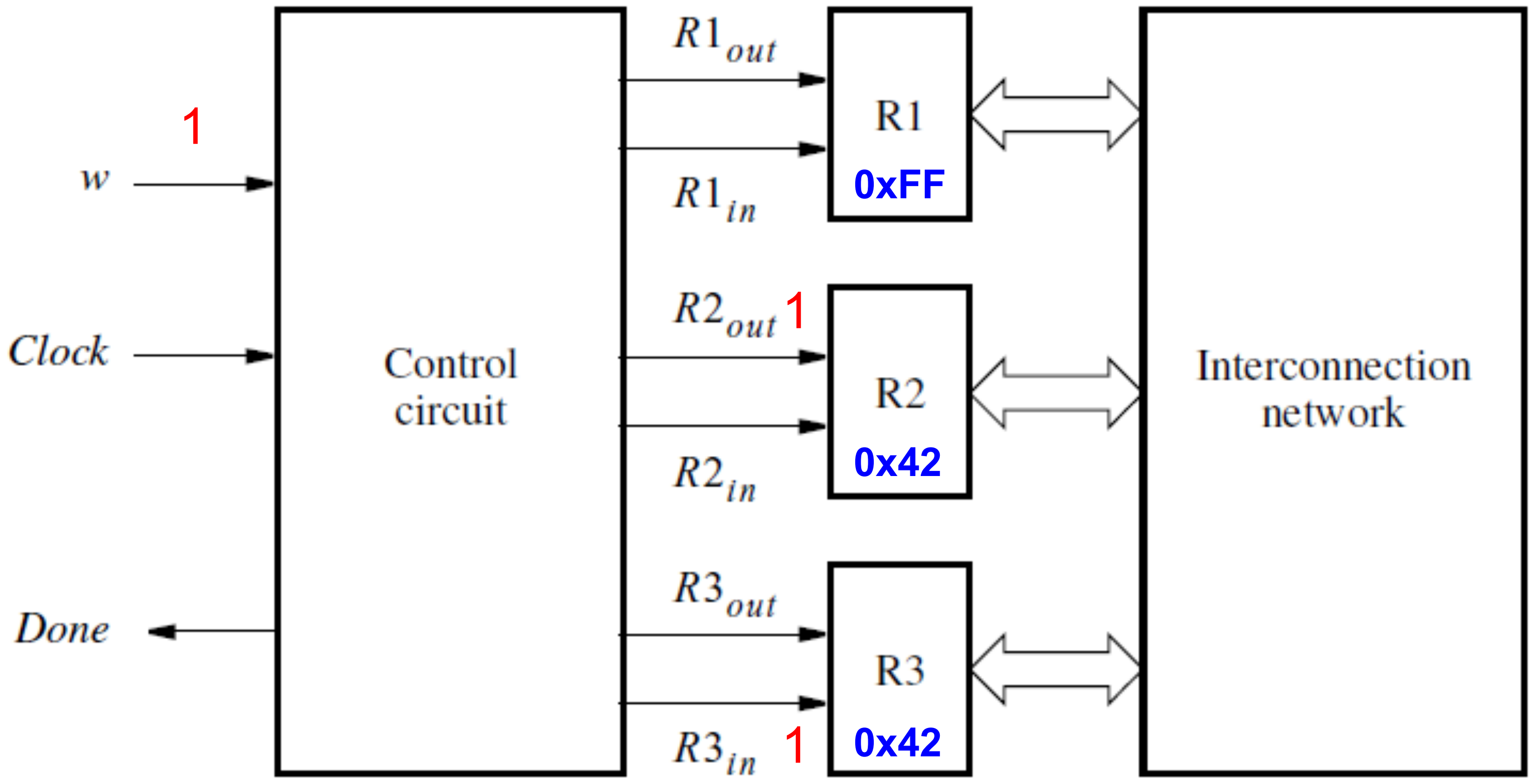
Animated Register Swap



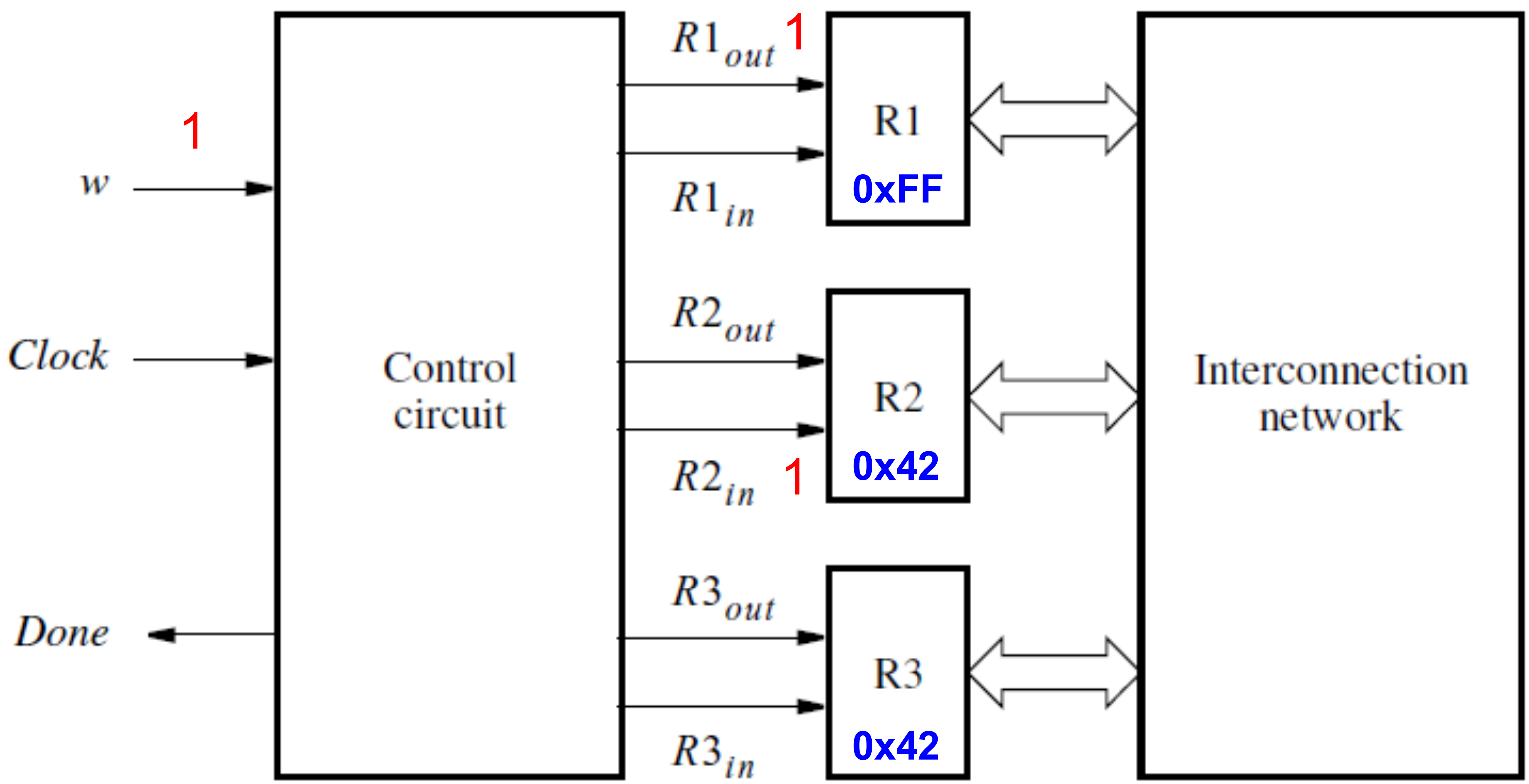
Animated Register Swap



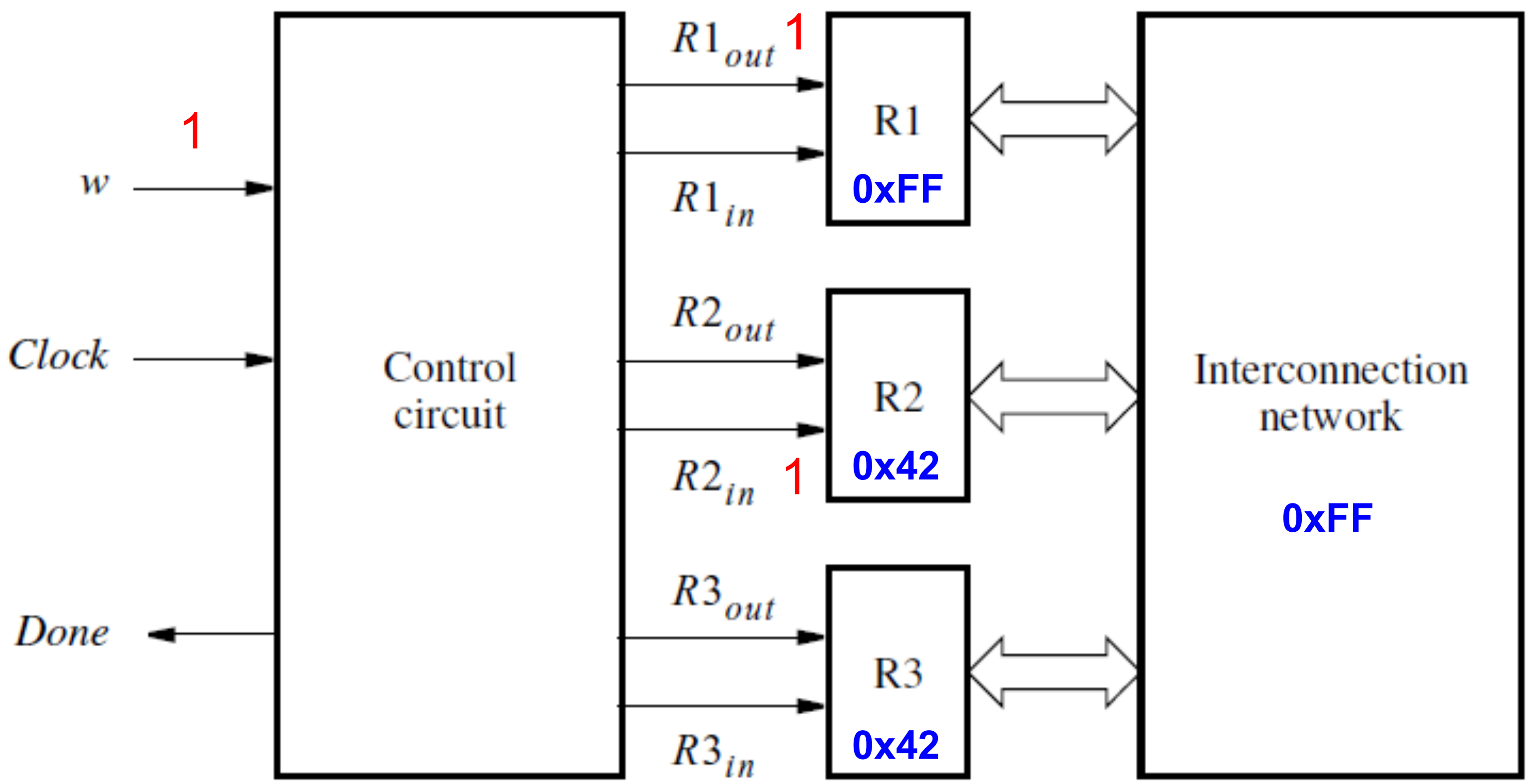
Animated Register Swap



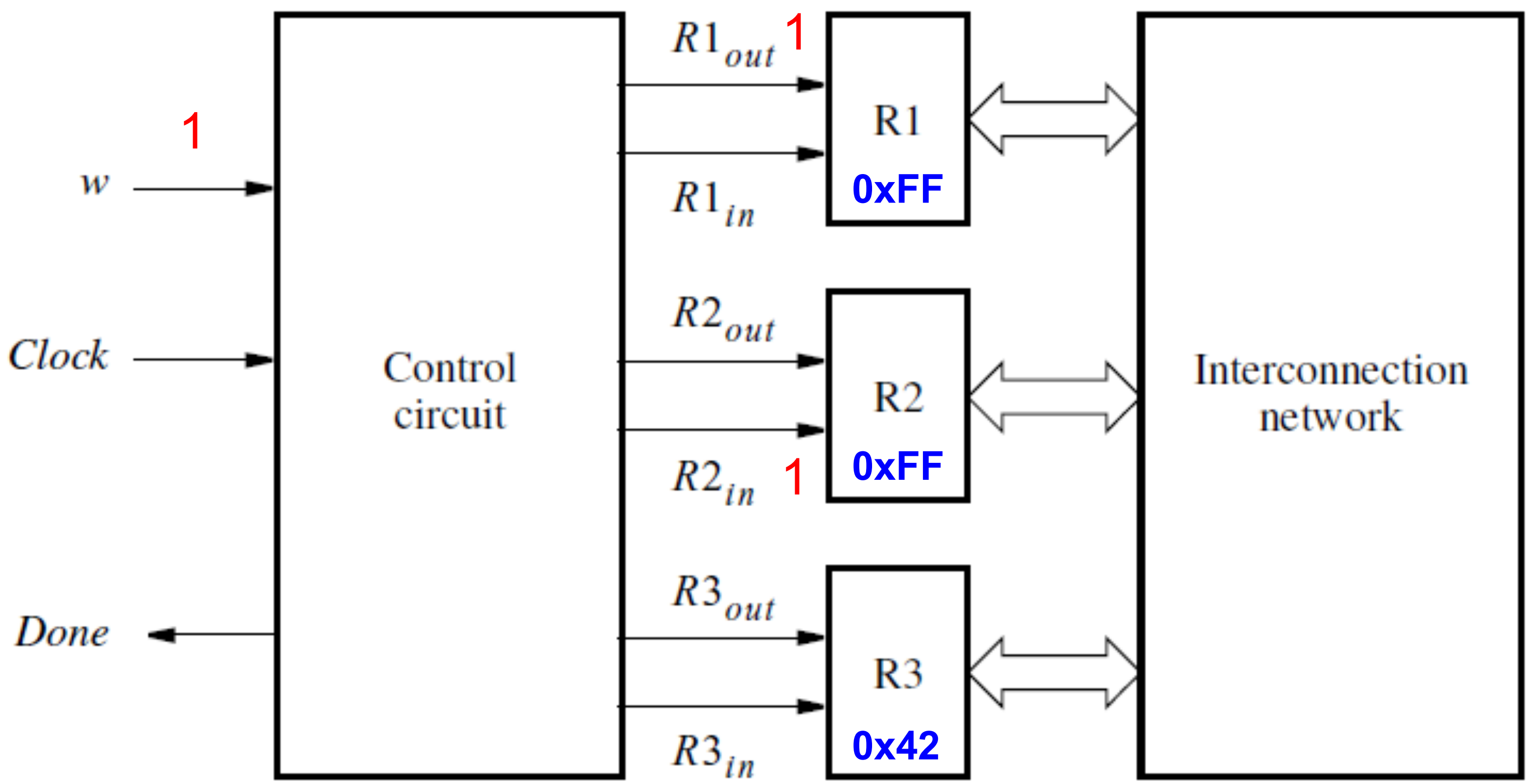
Animated Register Swap



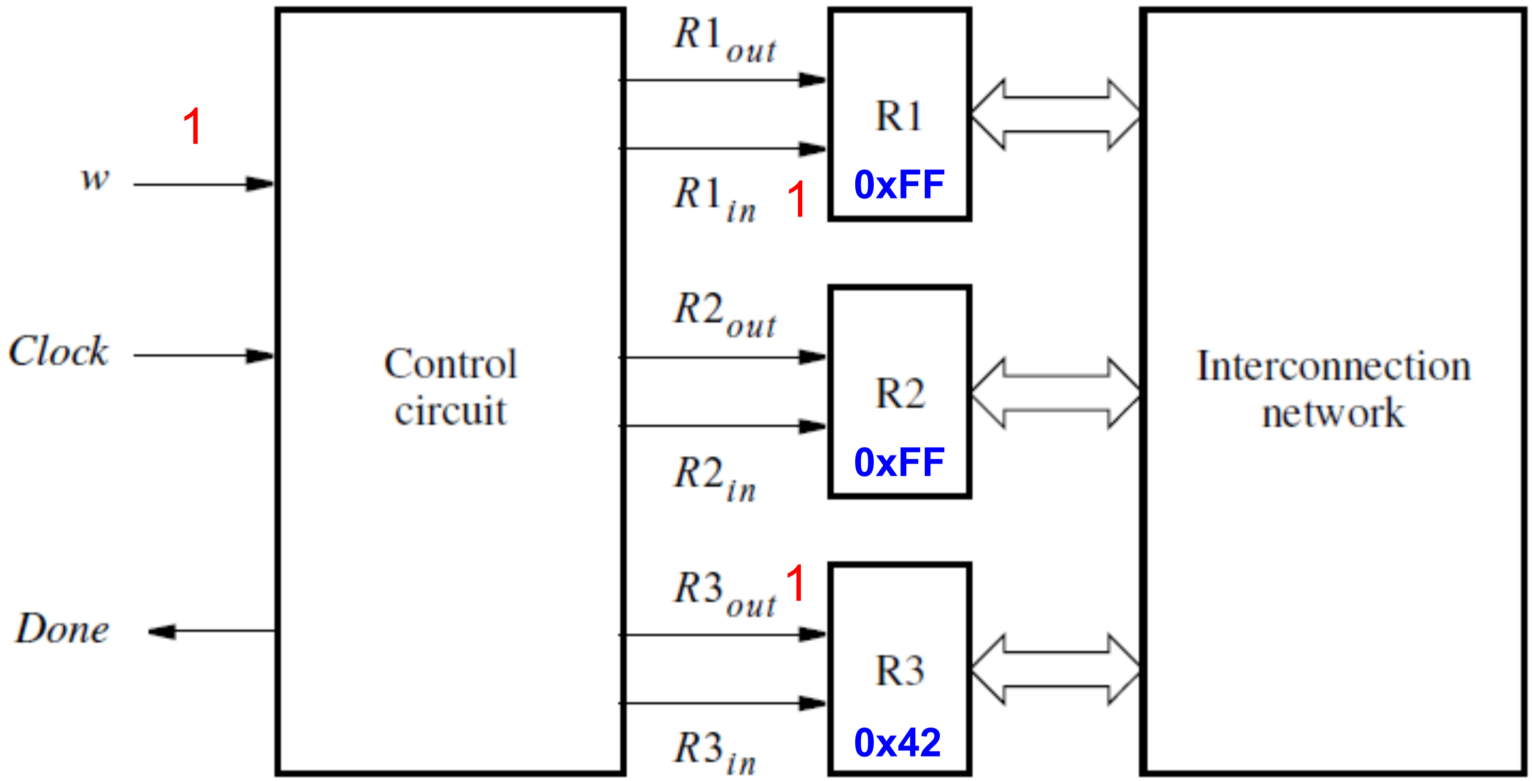
Animated Register Swap



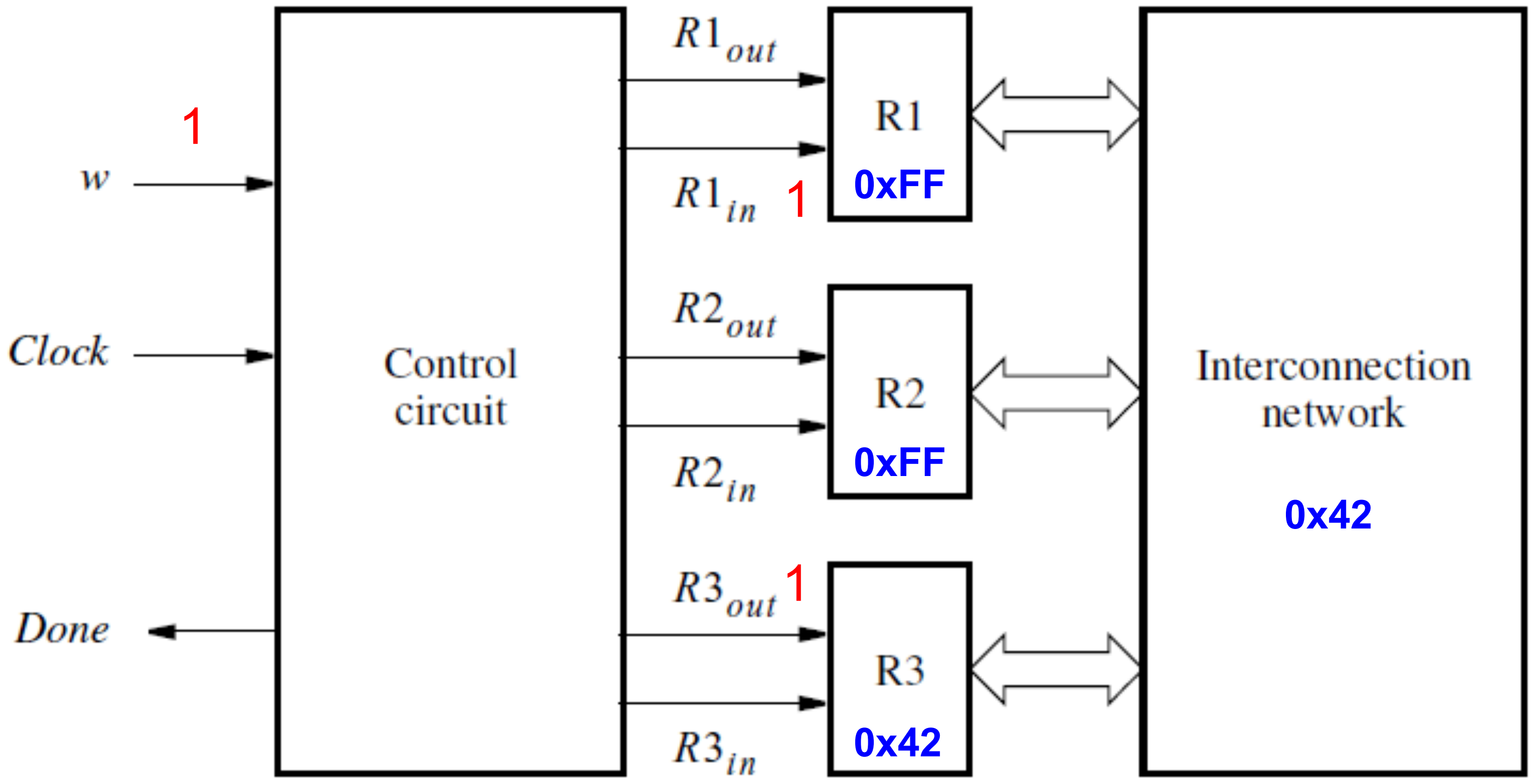
Animated Register Swap



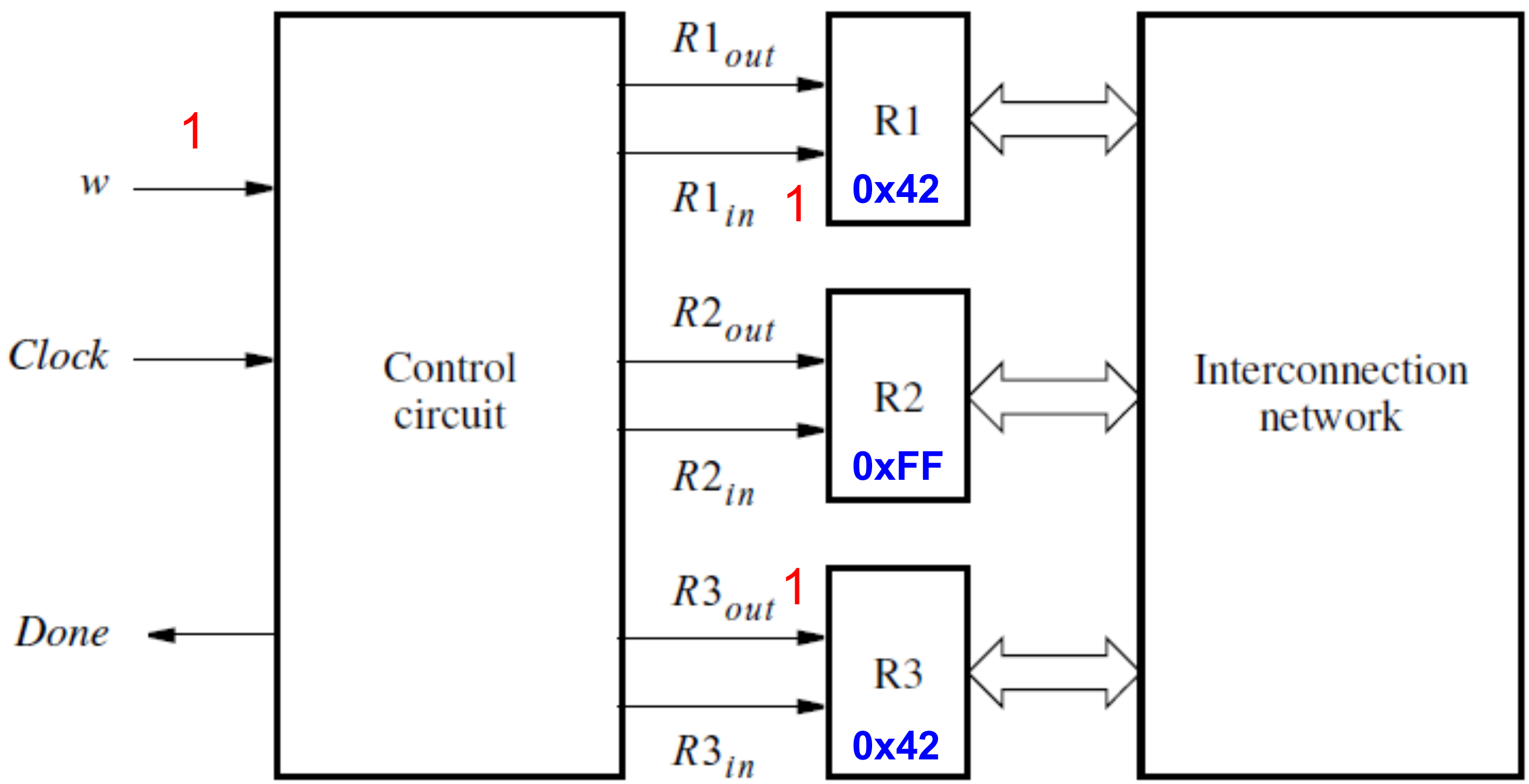
Animated Register Swap



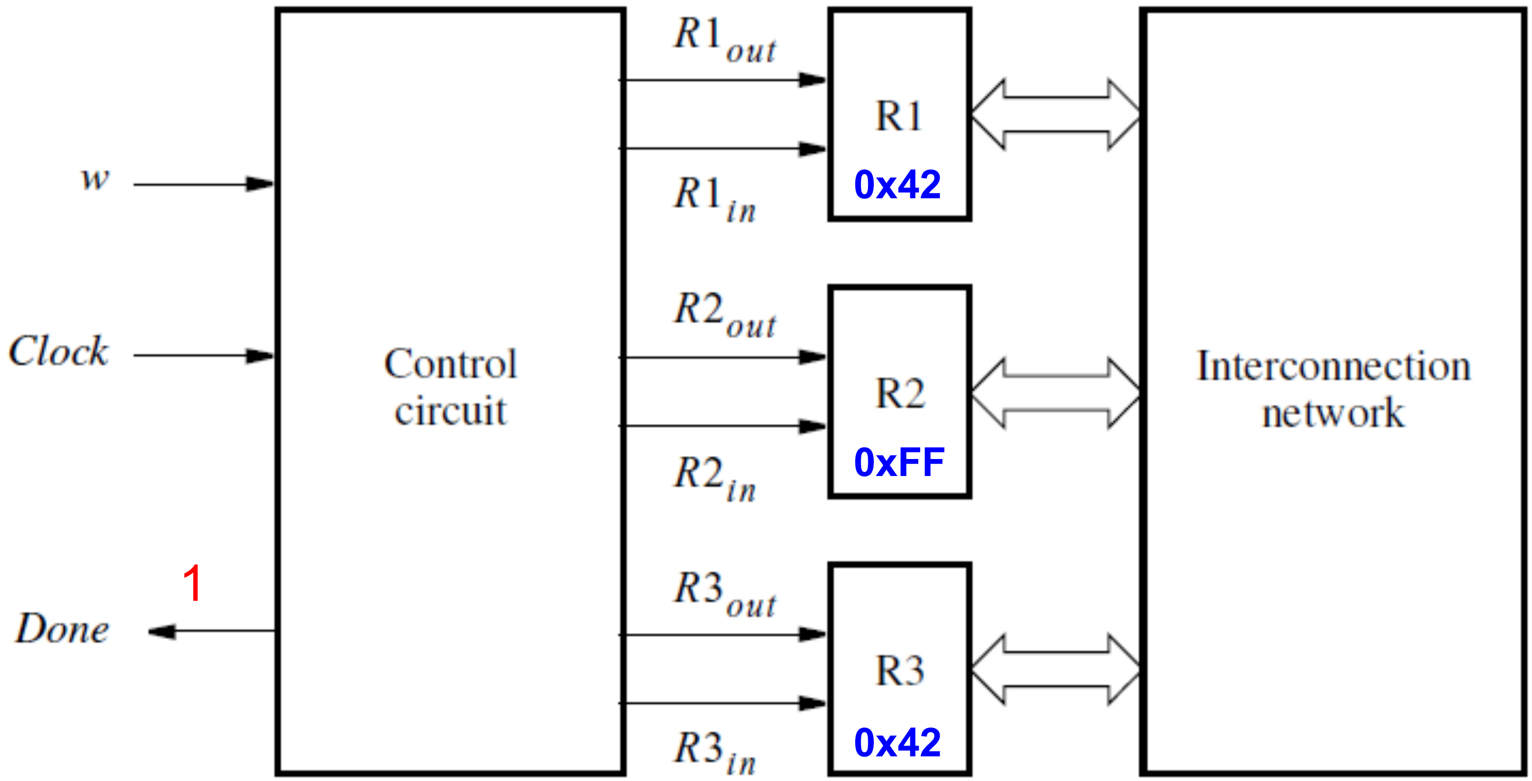
Animated Register Swap



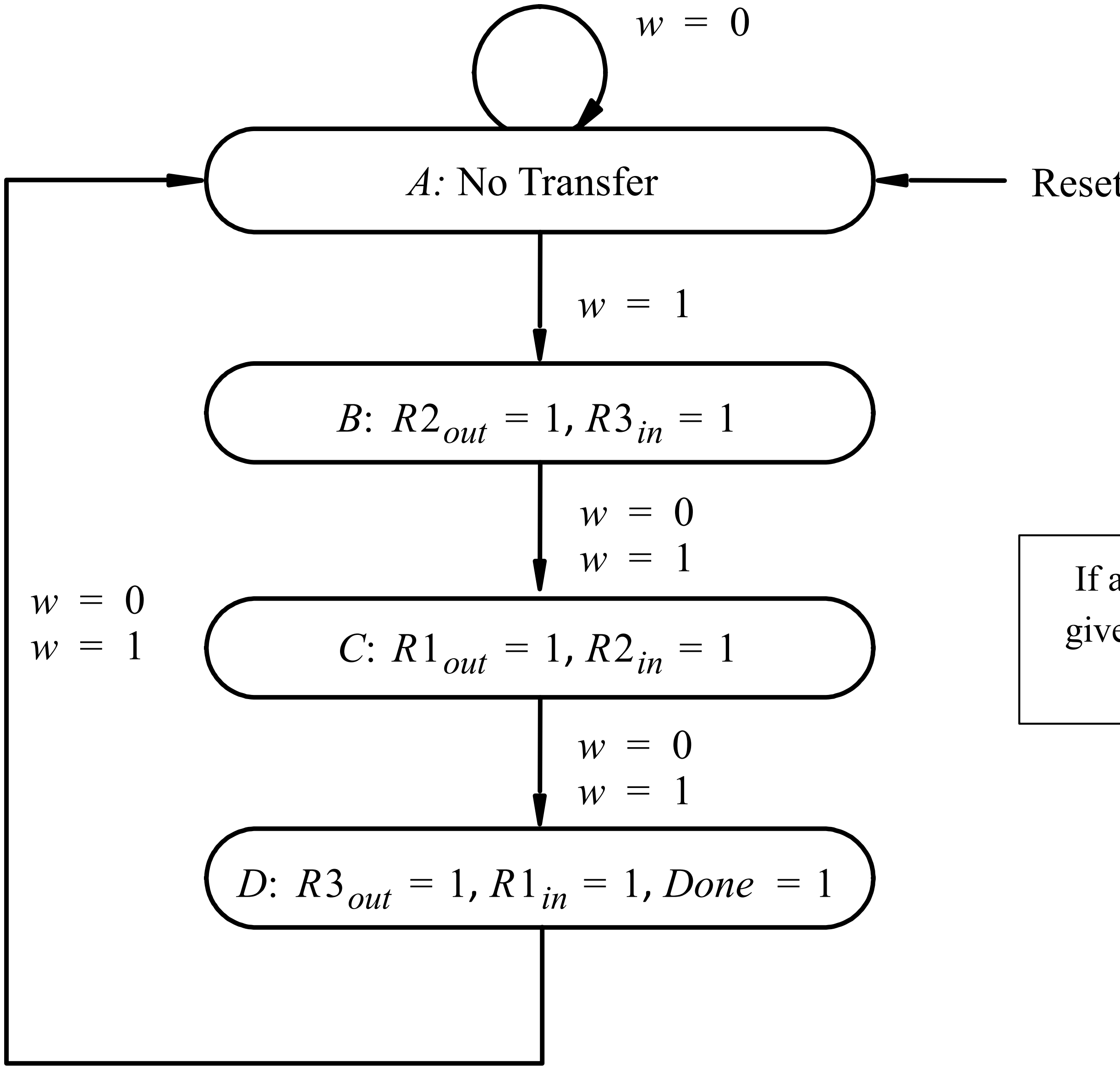
Animated Register Swap



Animated Register Swap



State Diagram



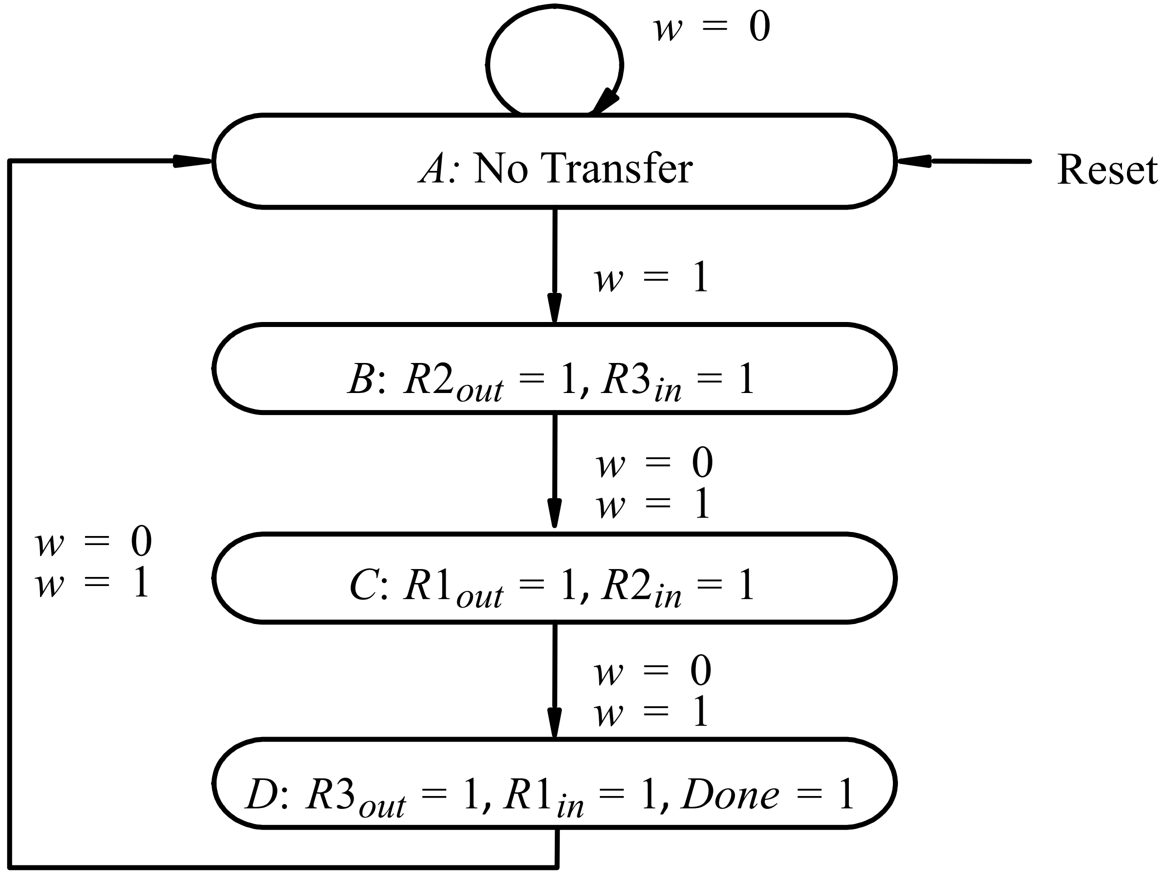
If an output is not given, then assume that it is 0.

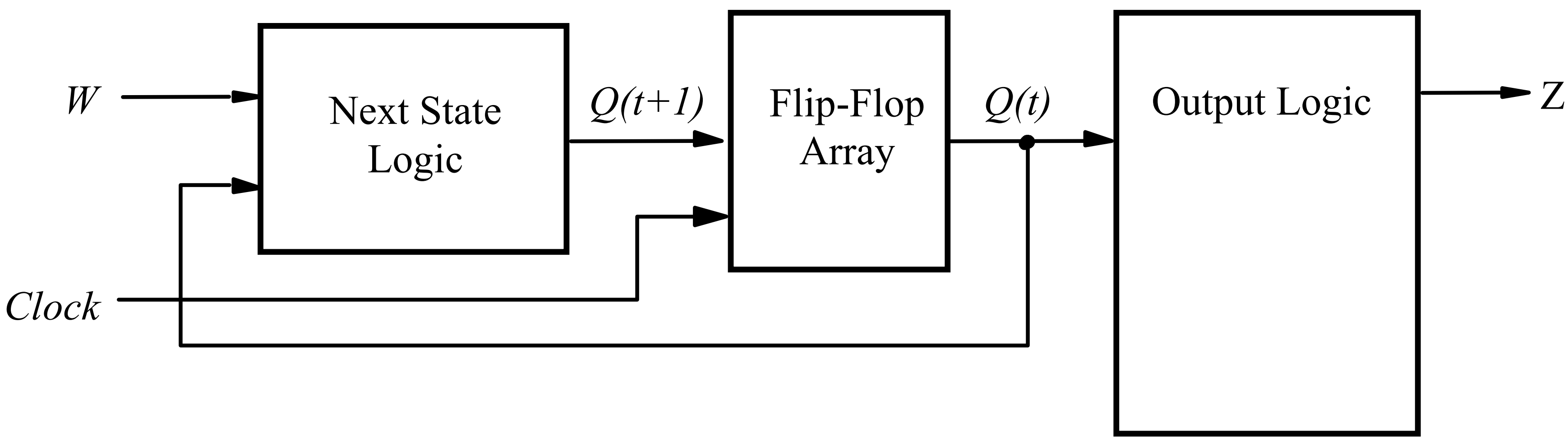
[Figure 6.11 from the textbook]

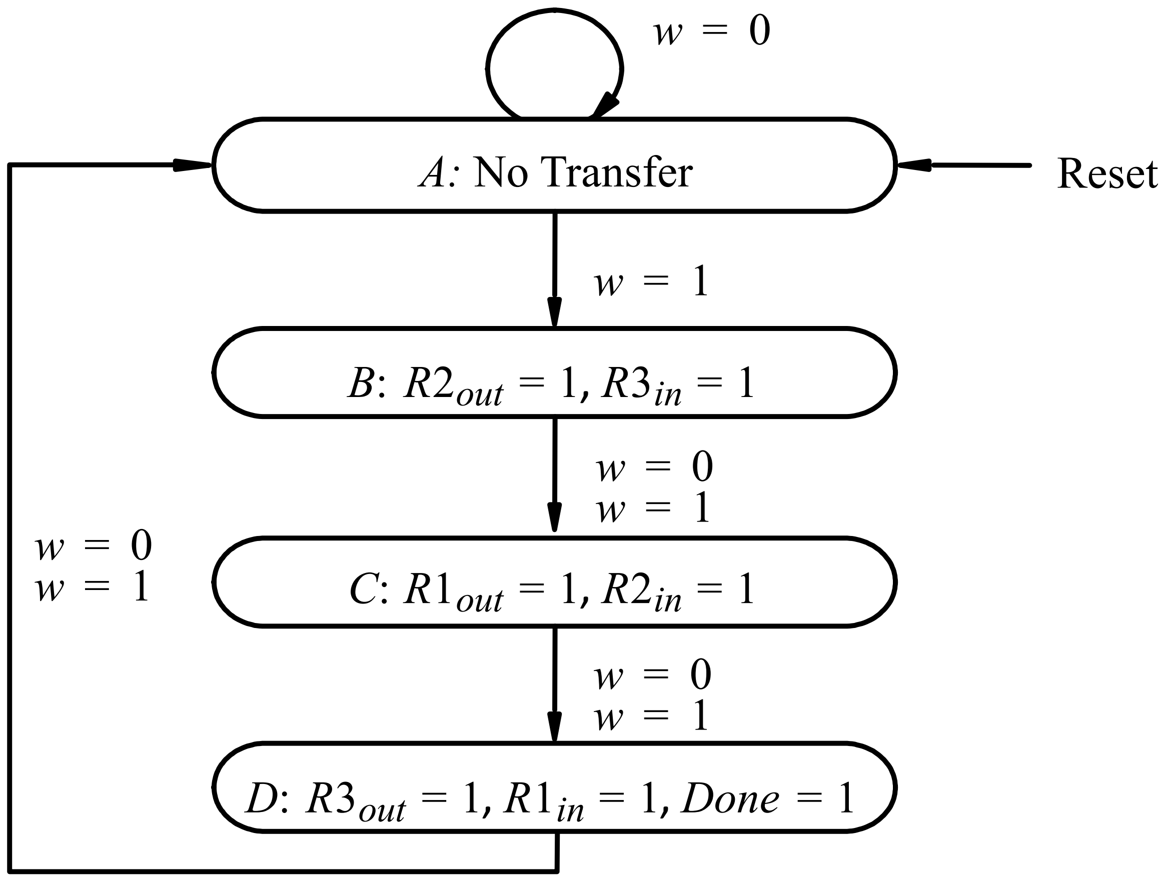
Some Questions

- How many flip-flops are we going to use?

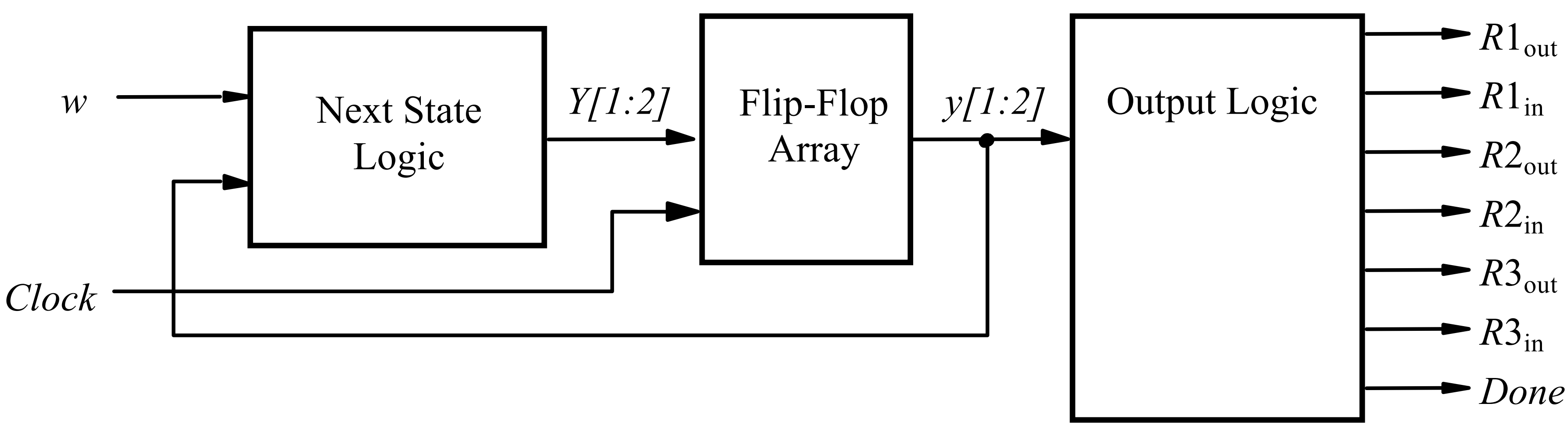
- How many logic expressions do we need to find?


w = 0
A: No Transfer
Reset
w = 1
B: R2out = 1, R3in = 1
w = 0
w = 1
C: R1out = 1, R2in = 1
w = 0
w = 1
D: R3out = 1, R1in = 1, Done = 1
w = 0
w = 1

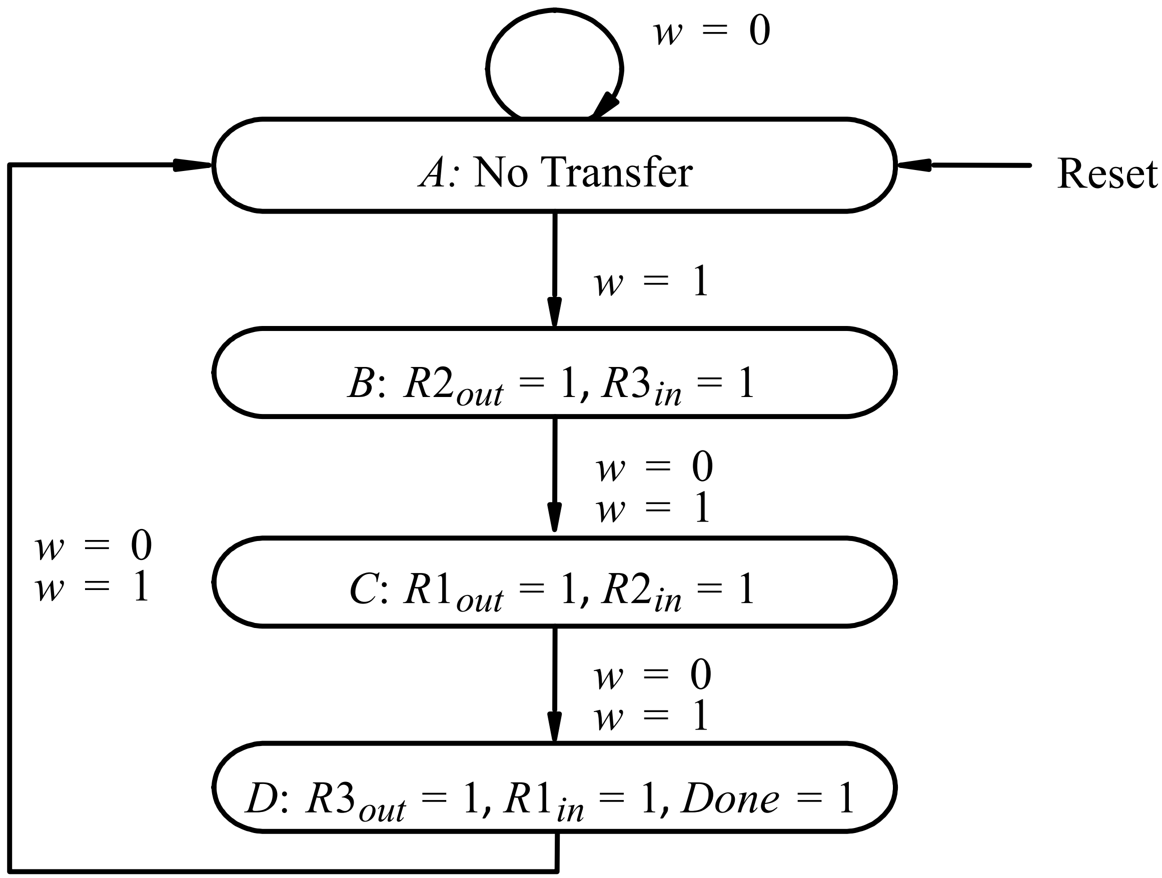

W
Next State Logic
Q(t+1)
Flip-Flop Array
Q(t)
Output Logic
Z
Clock



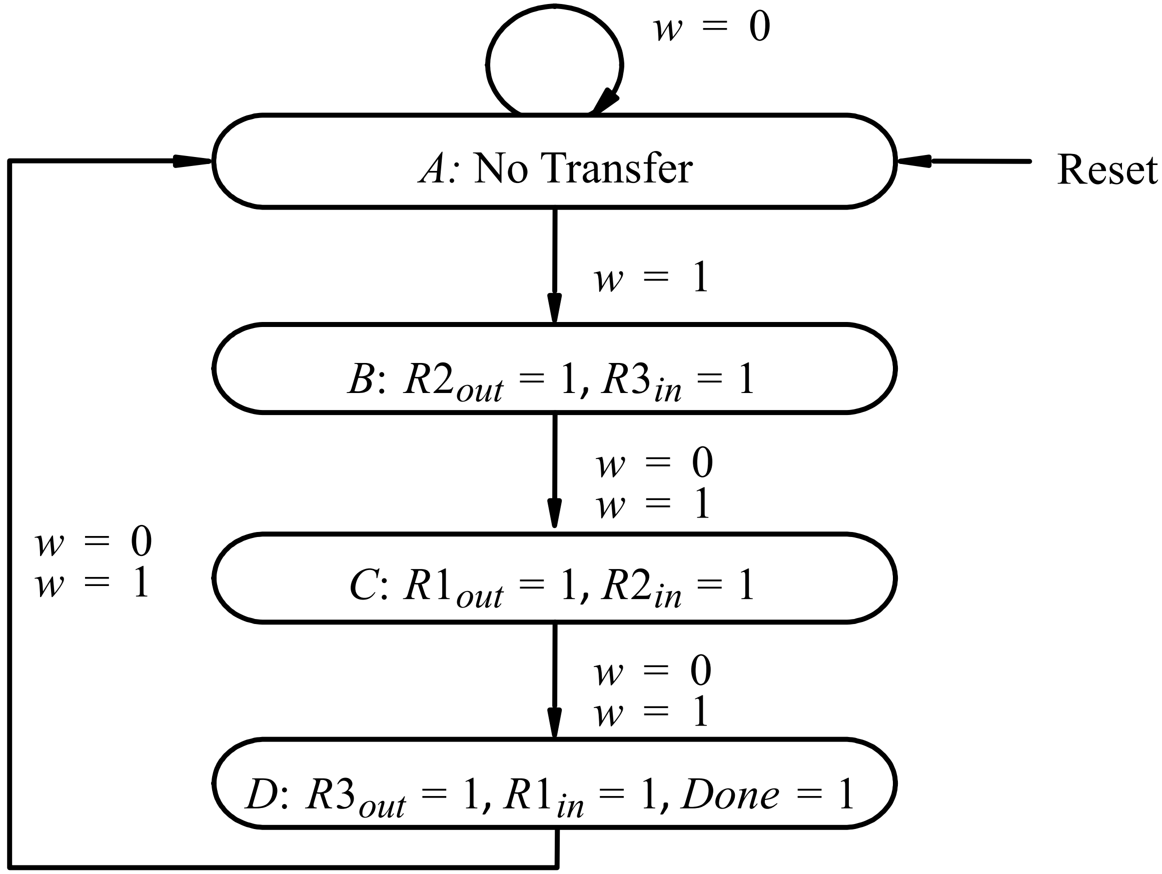
w = 0
A: No Transfer
Reset
w = 1
B: R2out = 1, R3in = 1
w = 0
w = 1
C: R1out = 1, R2in = 1
w = 0
w = 1
D: R3out = 1, R1in = 1, Done = 1
w = 0
w = 1



w
Next State Logic
Y[1:2]
Flip-Flop Array
y[1:2]
Output Logic
Clock
R1out
R1in
R2out
R2in
R3out
R3in
Done



Present state	Next state		Outputs						
	$w = 0$	$w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A									
B									
C									
D									



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

As we saw before, we can expect that some state encodings will be better than others.

We will consider three encoding schemes.

Encoding #1:
A=00, B=01, C=10, D=11

(Uses Two Flip-Flops)

State Table

Present state	Next state $w = 0$	Next state $w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*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

State-Assigned Table

	Present state y_2y_1	Next state $w = 0$ Y_2Y_1	Next state $w = 1$ Y_2Y_1	Outputs $R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A										
B										
C										
D										

[Figure 6.12 & 6.13 from the textbook]

State Table

Present state	Next state $w=0$	Next state $w=1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*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

State Assigned Table

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

[Figure 6.12 & 6.13 from the textbook]

State Table

Present state	Next state $w=0$	Next state $w=1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*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

State Assigned Table

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

[Figure 6.12 & 6.13 from the textbook]

State Table

Present state	Next state $w=0$	Next state $w=1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*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

State Assigned Table

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

[Figure 6.12 & 6.13 from the textbook]

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

y_2	y_1	w	Y_2	Y_1
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Let's derive the next-state expressions.

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

y_2	y_1	w	Y_2	Y_1
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

Pay attention to the way the columns of the truth table are labeled.

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

y_2	y_1	w	Y_2	Y_1
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

Y_1

$w \backslash y_2y_1$	00	01	11	10
0				
1				

Y_2

$w \backslash y_2y_1$	00	01	11	10
0				
1				

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

y_2	y_1	w	Y_2	Y_1
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

Y_1

w \ y_2y_1	00	01	11	10
0	0	0	0	1
1	1	0	0	1

Y_2

w \ y_2y_1	00	01	11	10
0	0	1	0	1
1	0	1	0	1

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

y_2	y_1	w	Y_2	Y_1
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

Y_1

w \ y_2y_1	00	01	11	10
0	0	0	0	1
1	1	0	0	1

Y_2

w \ y_2y_1	00	01	11	10
0	0	1	0	1
1	0	1	0	1

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

y_2	y_1	w	Y_2	Y_1
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	1	1
1	1	0	0	0
1	1	1	0	0

Y_1

w \ y_2y_1	00	01	11	10
0	0	0	0	1
1	1	0	0	1

$$Y_1 = w\bar{y}_1 + \bar{y}_1 y_2$$

Y_2

w \ y_2y_1	00	01	11	10
0	0	1	0	1
1	0	1	0	1

$$Y_2 = y_1\bar{y}_2 + \bar{y}_1 y_2$$

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

y_2	y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$
0	0			
0	1			
1	0			
1	1			

Let's derive the output expressions

	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	$y_2 y_1$	$Y_2 Y_1$	$Y_2 Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	00	00	0 1	0	0	0	0	0	0	0
B	01	10	1 0	0	0	1	0	0	1	0
C	10	11	1 1	1	0	0	1	0	0	0
D	11	00	0 0	0	1	0	0	1	0	1

y_2	y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$
0	0			
0	1			
1	0			
1	1			

Let's derive the output expressions.

We need to derive only these 3 unique ones.

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

y_2	y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$
0	0	0	0	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

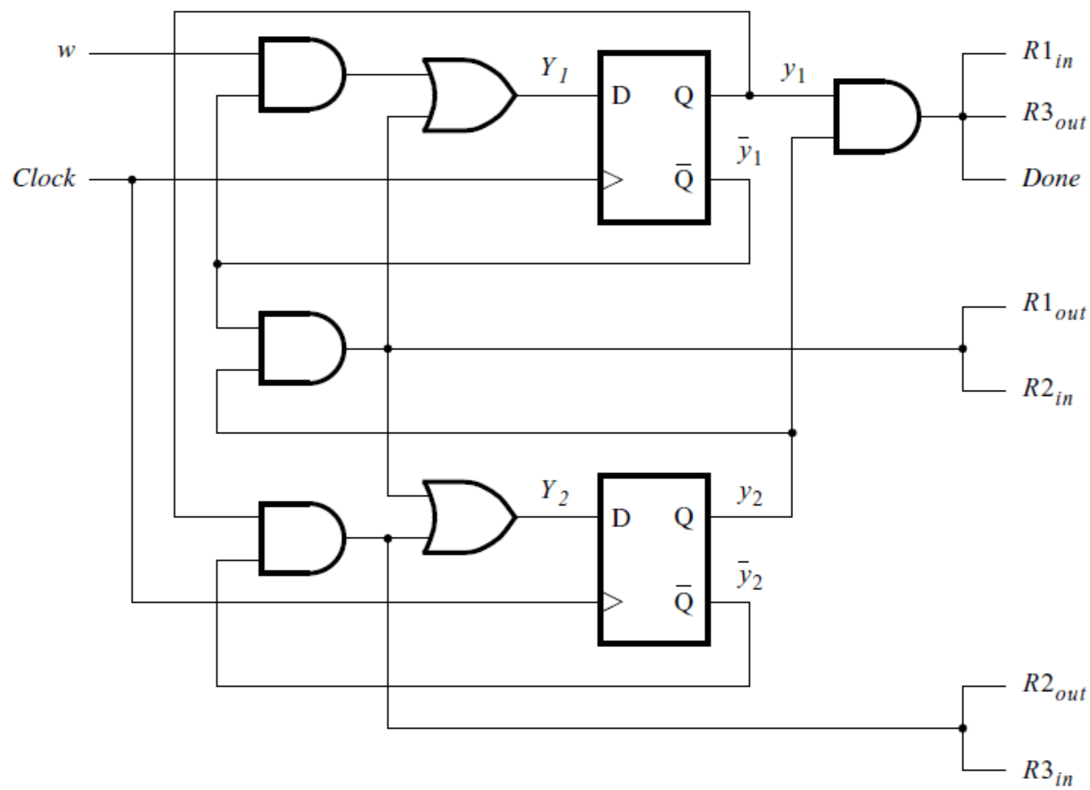
	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	$y_2 y_1$	$Y_2 Y_1$	$Y_2 Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	00	00	0 1	0	0	0	0	0	0	0
B	01	10	1 0	0	0	1	0	0	1	0
C	10	11	1 1	1	0	0	1	0	0	0
D	11	00	0 0	0	1	0	0	1	0	1

y_2	y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$
0	0	0	0	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

$$R1_{out} = R2_{in} = \overline{y_1}\, y_2$$

$$R1_{in} = R3_{out} = Done = y_1\, y_2$$

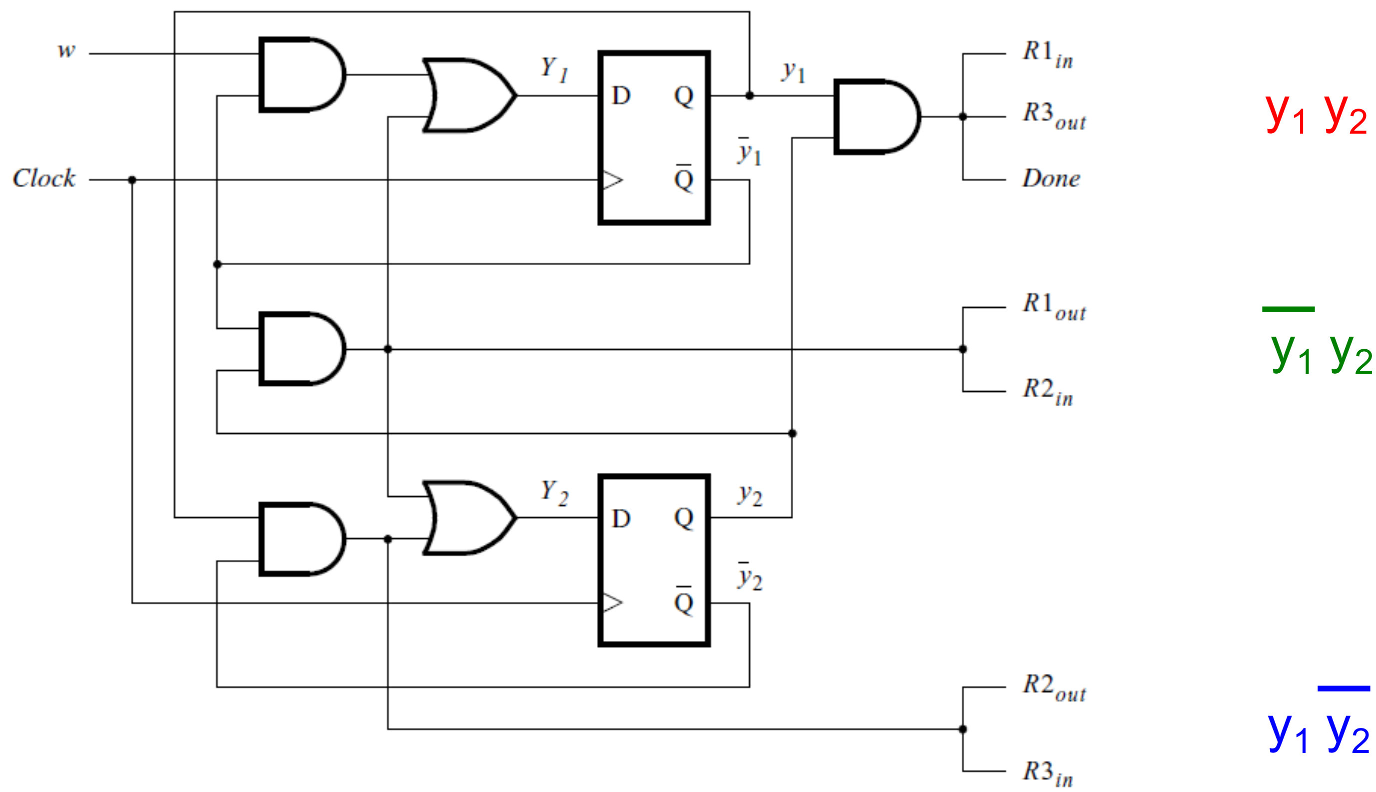
$$R2_{out} = R3_{in} = y_1\, \overline{y_2}$$



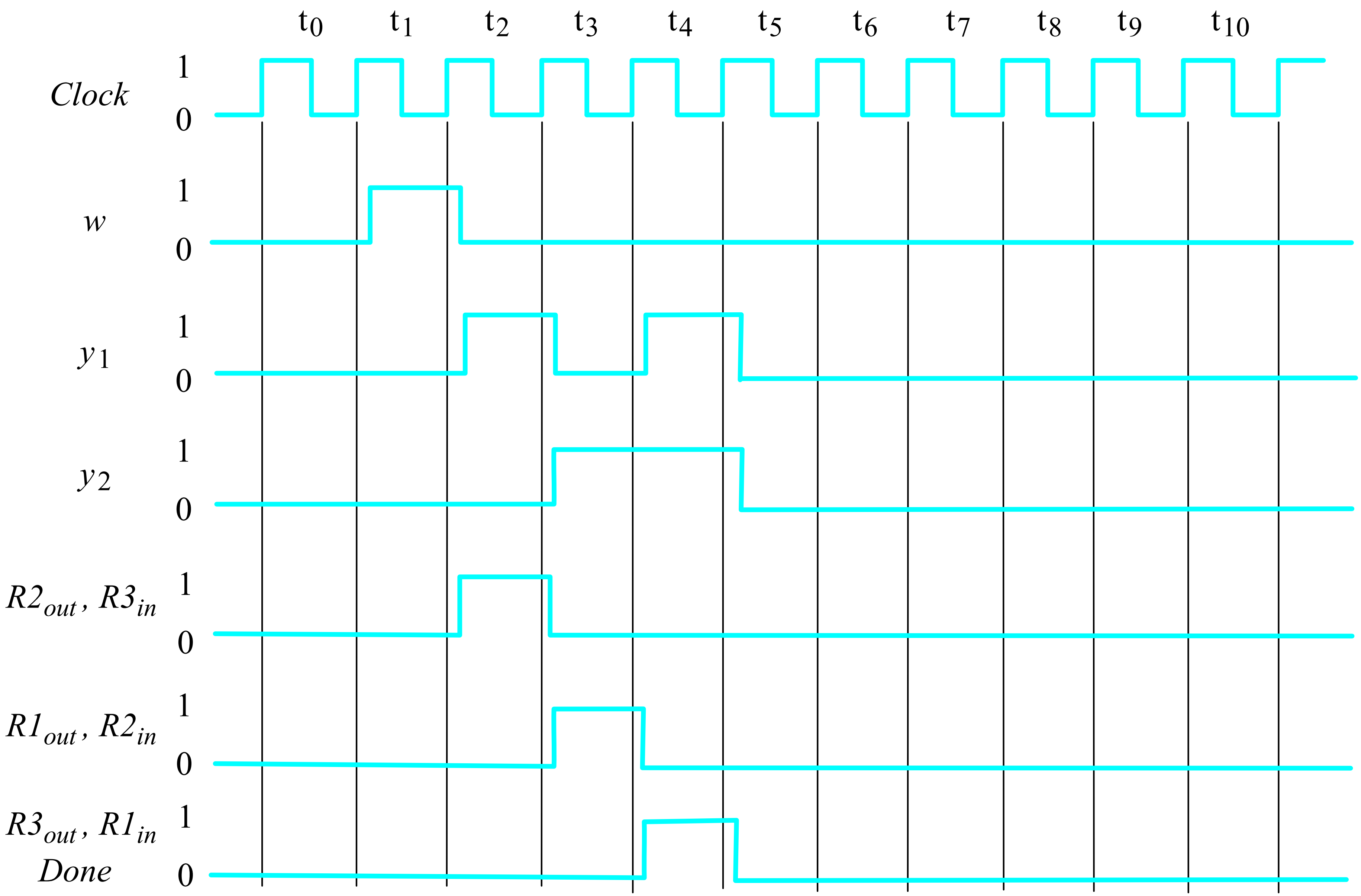
	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	00	00	0 1	0	0	0	0	0	0	0
B	01	10	1 0	0	0	1	0	0	1	0
C	10	11	1 1	1	0	0	1	0	0	0
D	11	00	0 0	0	1	0	0	1	0	1

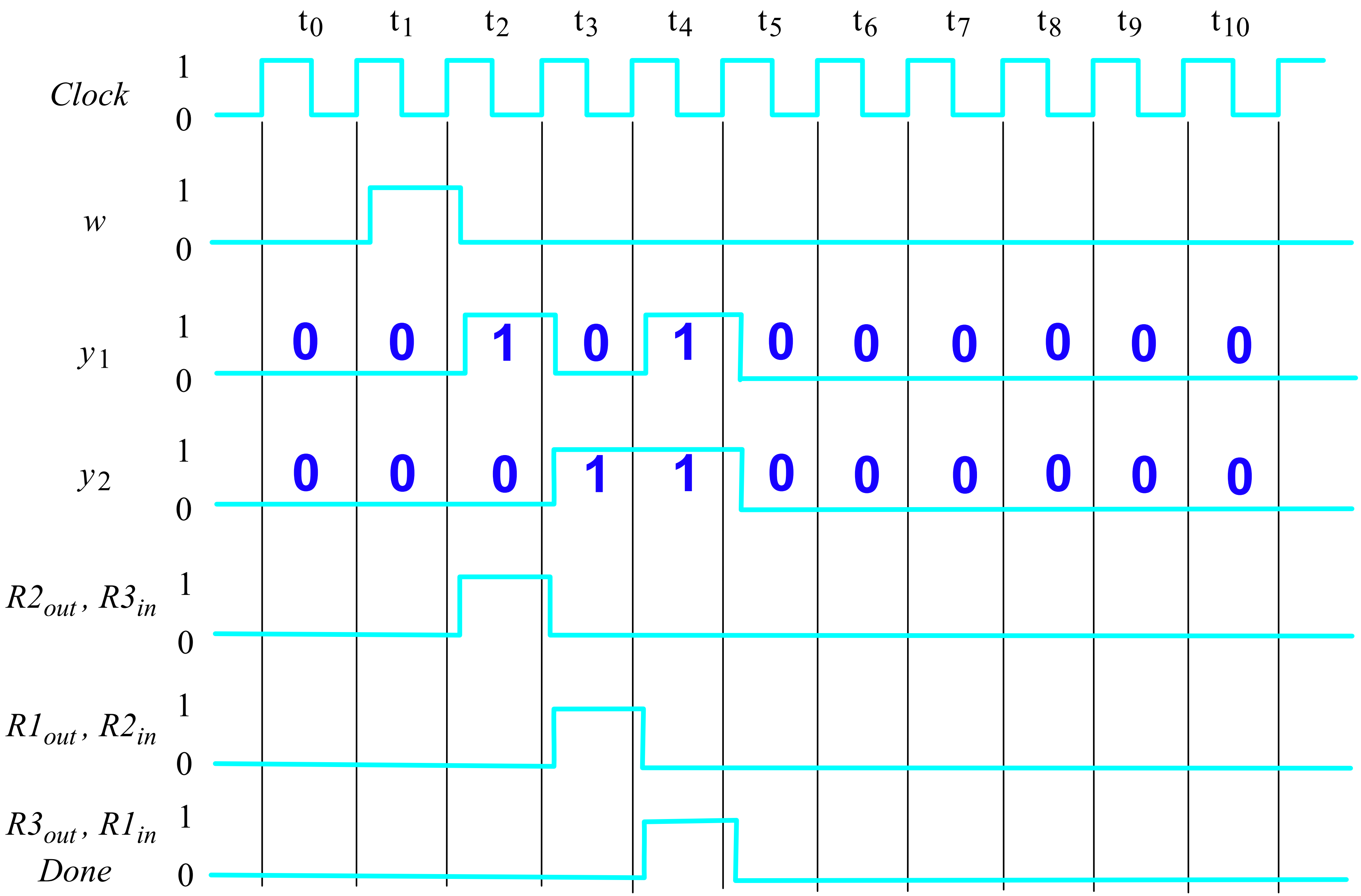
$$Y_1 = w\bar{y}_1 + \bar{y}_1 y_2$$

$$Y_2 = y_1\bar{y}_2 + \bar{y}_1 y_2$$

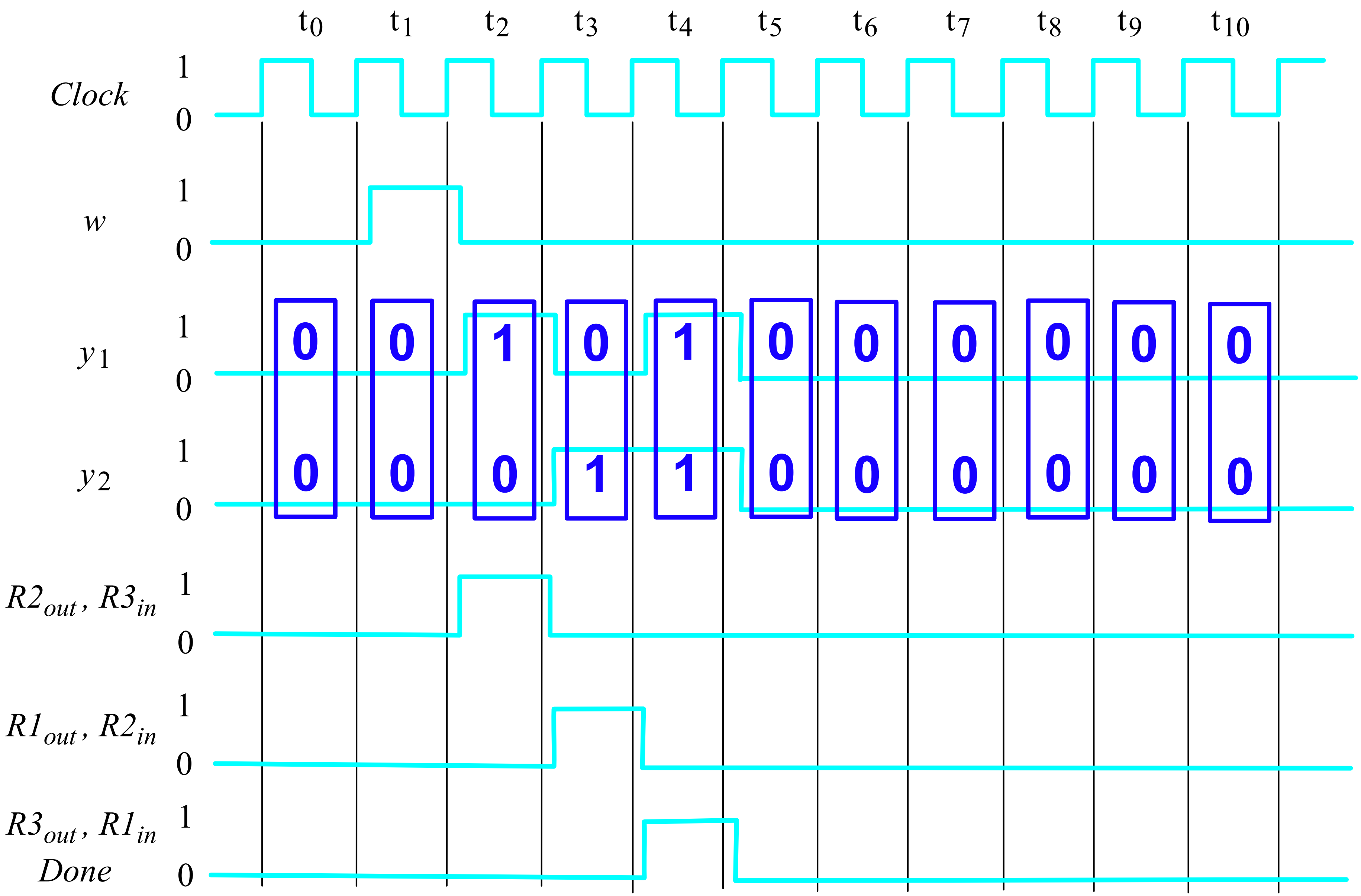


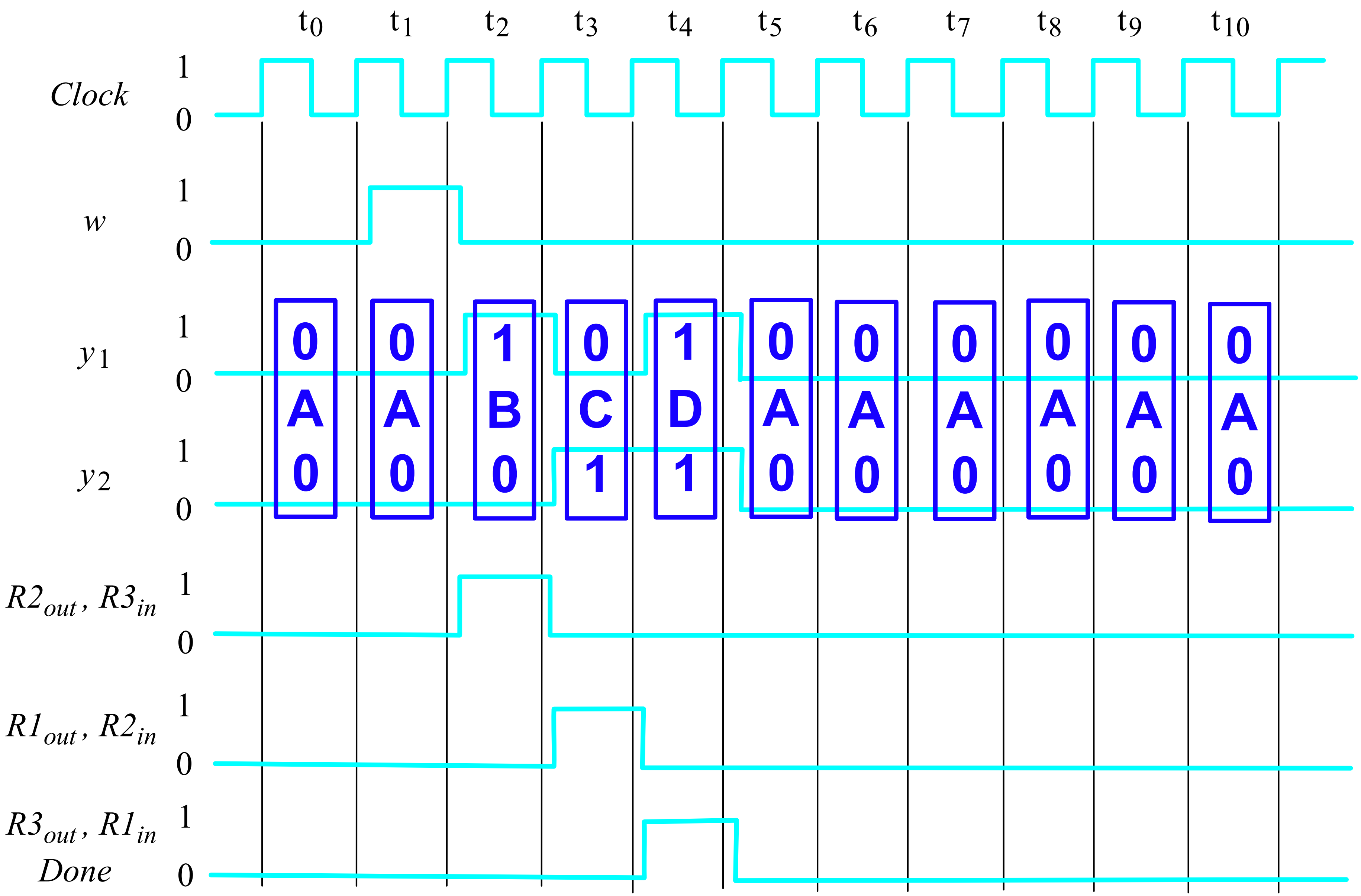
	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	00	00	0 1	0	0	0	0	0	0	0
B	01	10	1 0	0	0	1	0	0	1	0
C	10	11	1 1	1	0	0	1	0	0	0
D	11	00	0 0	0	1	0	0	1	0	1


t0 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10
Clock
w
y1
y2
R2out , R3in
R1out , R2in
R3out , R1in
Done
1
0

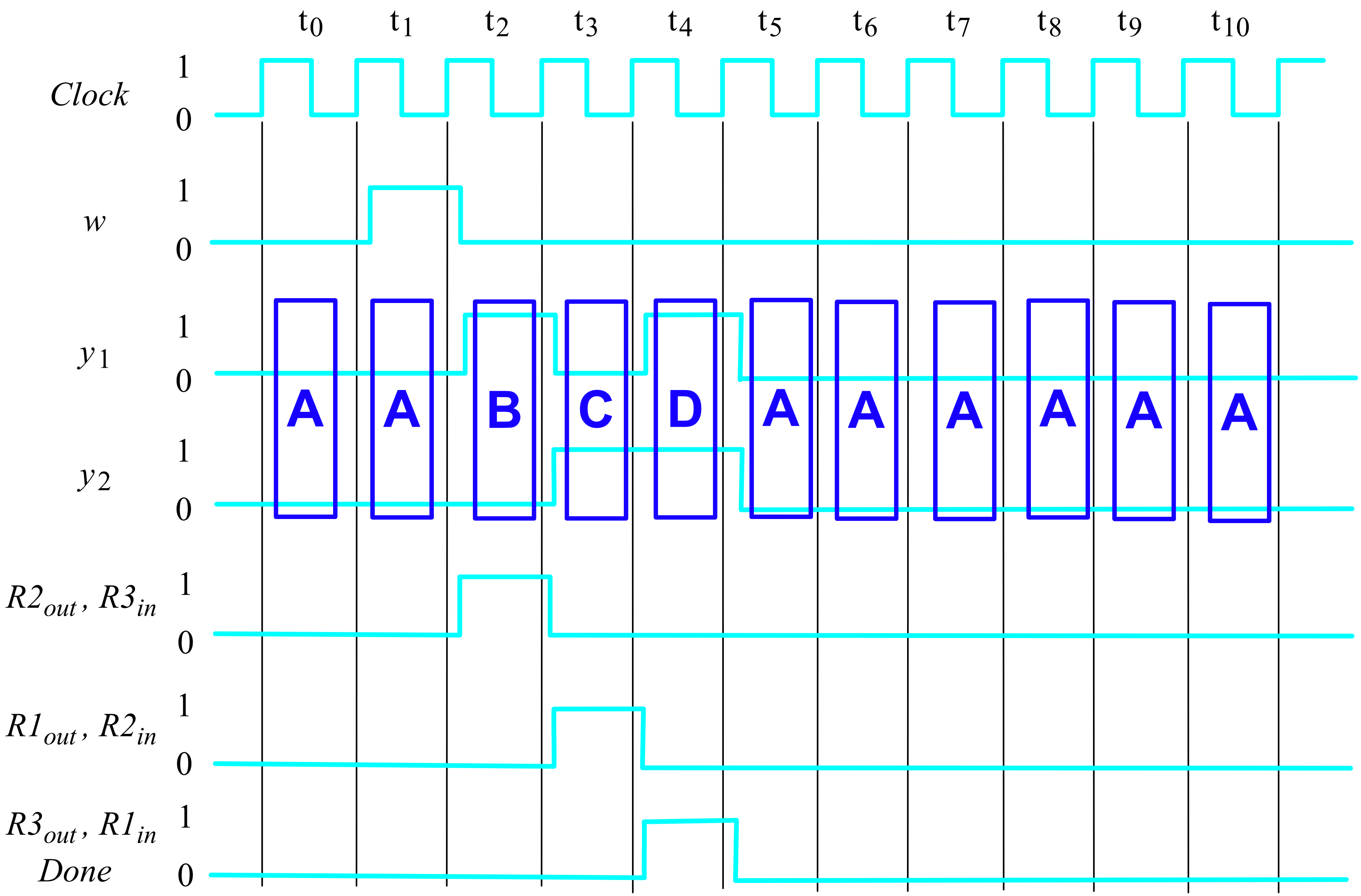


t0 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10
Clock
1
0
w
1
0
y1
1
0
0 0 1 0 1 0 0 0 0 0 0
y2
1
0
0 0 0 1 1 0 0 0 0 0 0
R2out, R3in
1
0
R1out, R2in
1
0
R3out, R1in
Done
1
0

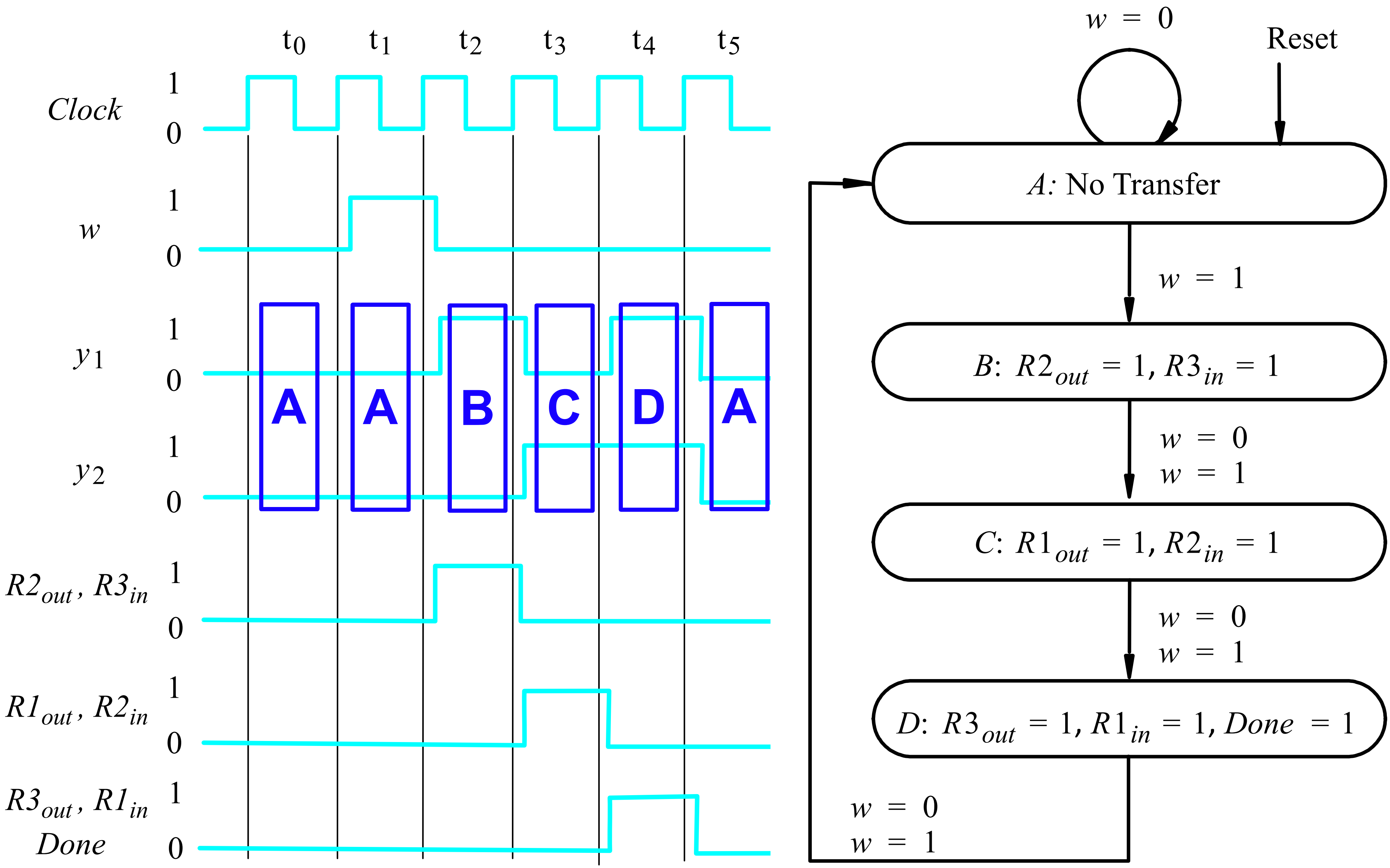

t0 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10
Clock
w
y1
y2
R2out, R3in
R1out, R2in
R3out, R1in
Done
1
0
0 0 1 0 1 0 0 0 0 0 0
0 0 0 1 1 0 0 0 0 0 0



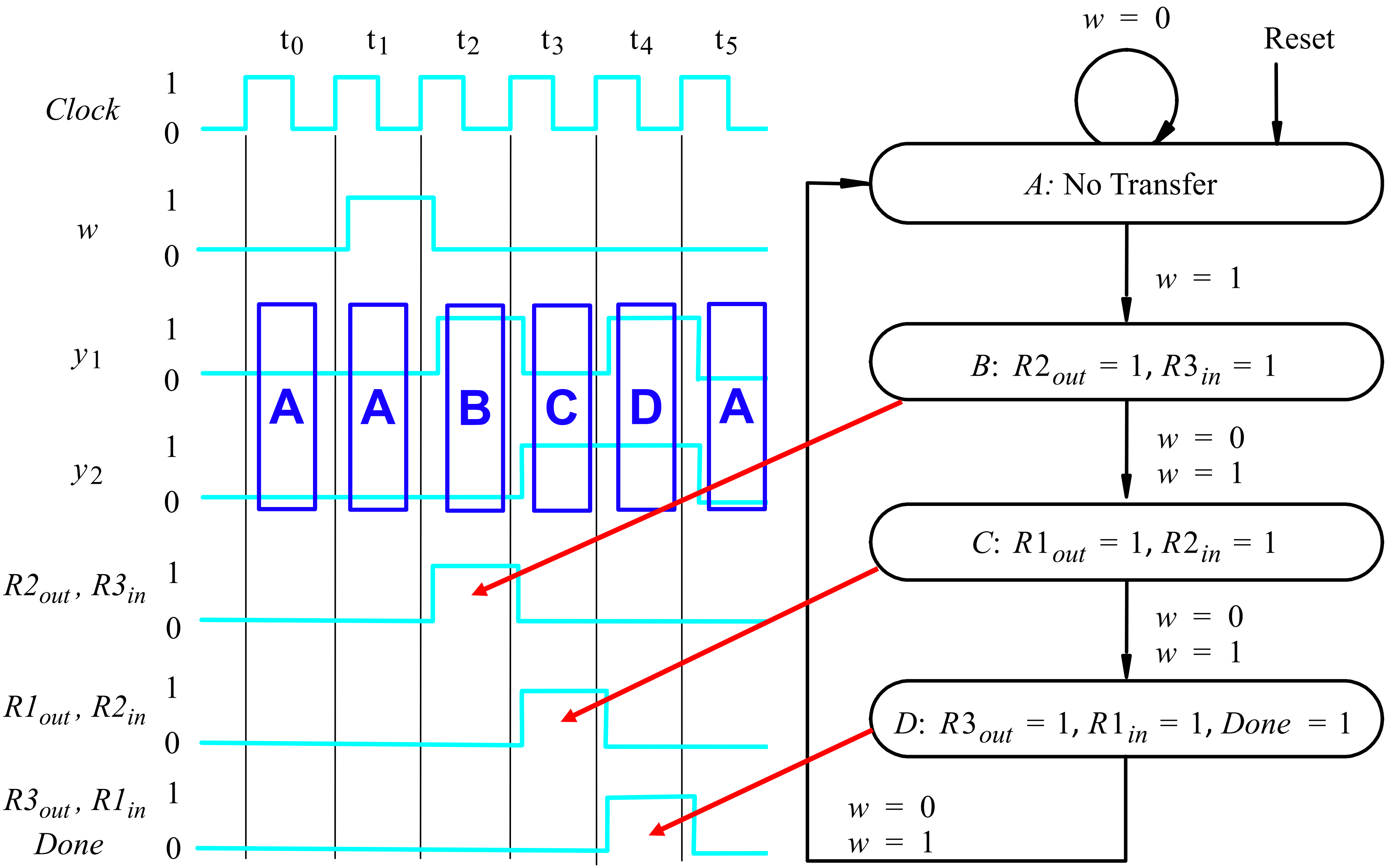
t0 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10
Clock
w
y1
y2
R2out, R3in
R1out, R2in
R3out, R1in
Done
1
0
0 A 0
0 A 0
1 B 0
0 C 1
1 D 1
0 A 0
0 A 0
0 A 0
0 A 0
0 A 0
0 A 0



t0 t1 t2 t3 t4 t5 t6 t7 t8 t9 t10
Clock
w
y1
y2
R2out, R3in
R1out, R2in
R3out, R1in
Done
1
0
A A B C D A A A A A A



t0
t1
t2
t3
t4
t5
Clock
w
y1
y2
R2out, R3in
R1out, R2in
R3out, R1in
Done
1
0
A
A
B
C
D
A
w = 0
Reset
A: No Transfer
w = 1
B: R2out = 1, R3in = 1
w = 0
w = 1
C: R1out = 1, R2in = 1
w = 0
w = 1
D: R3out = 1, R1in = 1, Done = 1
w = 0
w = 1



t0 t1 t2 t3 t4 t5
Clock
w
y1
y2
R2out, R3in
R1out, R2in
R3out, R1in
Done
A A B C D A
w = 0
Reset
A: No Transfer
w = 1
B: R2out = 1, R3in = 1
w = 0
w = 1
C: R1out = 1, R2in = 1
w = 0
w = 1
D: R3out = 1, R1in = 1, Done = 1
w = 0
w = 1

Encoding #2:
A=00, B=01, C=11, D=10

(Also Uses Two Flip-Flops)

State Table (same as before)

Present state	Next state $w = 0$	Next state $w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*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

State-Assigned Table

	Present state y_2y_1	Next state $w = 0$ Y_2Y_1	Next state $w = 1$ Y_2Y_1	Outputs $R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A										
B										
C										
D										

[Figure 6.12 & 6.18 from the textbook]

State Table (same as before)

Present state	Next state $w = 0$	Next state $w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*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

State-Assigned Table

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

[Figure 6.12 & 6.18 from the textbook]

State Table (same as before)

Present state	Next state $w=0$	Next state $w=1$	Outputs $R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*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

State-Assigned Table

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

[Figure 6.12 & 6.18 from the textbook]

State Table (same as before)

Present state	Next state $w=0$	Next state $w=1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*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

State-Assigned Table

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

[Figure 6.12 & 6.18 from the textbook]

	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	$y_2 y_1$	$Y_2 Y_1$	$Y_2 Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	00	00	0 1	0	0	0	0	0	0	0
B	01	11	1 1	0	0	1	0	0	1	0
C	11	10	1 0	1	0	0	1	0	0	0
D	10	00	0 0	0	1	0	0	1	0	1

y_2	y_1	w	Y_2	Y_1
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

Let's derive the next-state expressions

	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	$y_2 y_1$	$Y_2 Y_1$	$Y_2 Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	00	00	0 1	0	0	0	0	0	0	0
B	01	11	1 1	0	0	1	0	0	1	0
C	11	10	1 0	1	0	0	1	0	0	0
D	10	00	0 0	0	1	0	0	1	0	1

y_2	y_1	w	Y_2	Y_1
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	0

	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	$y_2 y_1$	$Y_2 Y_1$	$Y_2 Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	00	00	0 1	0	0	0	0	0	0	0
B	01	11	1 1	0	0	1	0	0	1	0
C	11	10	1 0	1	0	0	1	0	0	0
D	10	00	0 0	0	1	0	0	1	0	1

y_2	y_1	w	Y_2	Y_1
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	0

Y_1

w \ $y_2 y_1$	00	01	11	10
0				
1				

Y_2

w \ $y_2 y_1$	00	01	11	10
0				
1				

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

y_2	y_1	w	Y_2	Y_1
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	0

Y_1

w \ y_2y_1	00	01	11	10
0	0	1	0	0
1	1	1	0	0

Y_2

w \ y_2y_1	00	01	11	10
0	0	1	1	0
1	0	1	1	0

	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	$y_2 y_1$	$Y_2 Y_1$	$Y_2 Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	00	00	0 1	0	0	0	0	0	0	0
B	01	11	1 1	0	0	1	0	0	1	0
C	11	10	1 0	1	0	0	1	0	0	0
D	10	00	0 0	0	1	0	0	1	0	1

y_2	y_1	w	Y_2	Y_1
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	0

Y_1

w \ $y_2 y_1$	00	01	11	10
0	0	1	0	0
1	1	1	0	0

Y_2

w \ $y_2 y_1$	00	01	11	10
0	0	1	1	0
1	0	1	1	0

	Present state	Next state		Outputs						
		$w=0$	$w=1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$\mathrm{R1}_{out}$	$\mathrm{R1}_{in}$	$\mathrm{R2}_{out}$	$\mathrm{R2}_{in}$	$\mathrm{R3}_{out}$	$\mathrm{R3}_{in}$	*Done*
A	00	00	0 1	0	0	0	0	0	0	0
B	01	11	1 1	0	0	1	0	0	1	0
C	11	10	1 0	1	0	0	1	0	0	0
D	10	00	0 0	0	1	0	0	1	0	1

y_2	y_1	w	Y_2	Y_1
0	0	0	0	0
0	0	1	0	1
0	1	0	1	1
0	1	1	1	1
1	0	0	0	0
1	0	1	0	0
1	1	0	1	0
1	1	1	1	0

Y_1 (w \ y_2y_1)	00	01	11	10
0	0	1	0	0
1	1	1	0	0

$$Y_1 = w\bar{y}_2 + y_1\bar{y}_2$$

Y_2 (w \ y_2y_1)	00	01	11	10
0	0	1	1	0
1	0	1	1	0

$$Y_2 = y_1$$

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

y_2	y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$
0	0			
0	1			
1	0			
1	1			

Let's derive the output expressions

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

y_2	y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$
0	0			
0	1			
1	0			
1	1			

Let's derive the output expressions

Once again, we only need to derive these three unique ones.

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

	y_2	y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$
A	0	0	0		
B	0	1	0		
D	1	0	0		
C	1	1	**1**		

Note that C and D are swapped in the truth table due to the new state encoding that was chosen.

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

	y_2	y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$
A	0	0	0	0	0
B	0	1	0	0	1
D	1	0	0	1	0
C	1	1	1	0	0

	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	$y_2 y_1$	$Y_2 Y_1$	$Y_2 Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	00	00	0 1	0	0	0	0	0	0	0
B	01	11	1 1	0	0	1	0	0	1	0
C	11	10	1 0	1	0	0	1	0	0	0
D	10	00	0 0	0	1	0	0	1	0	1

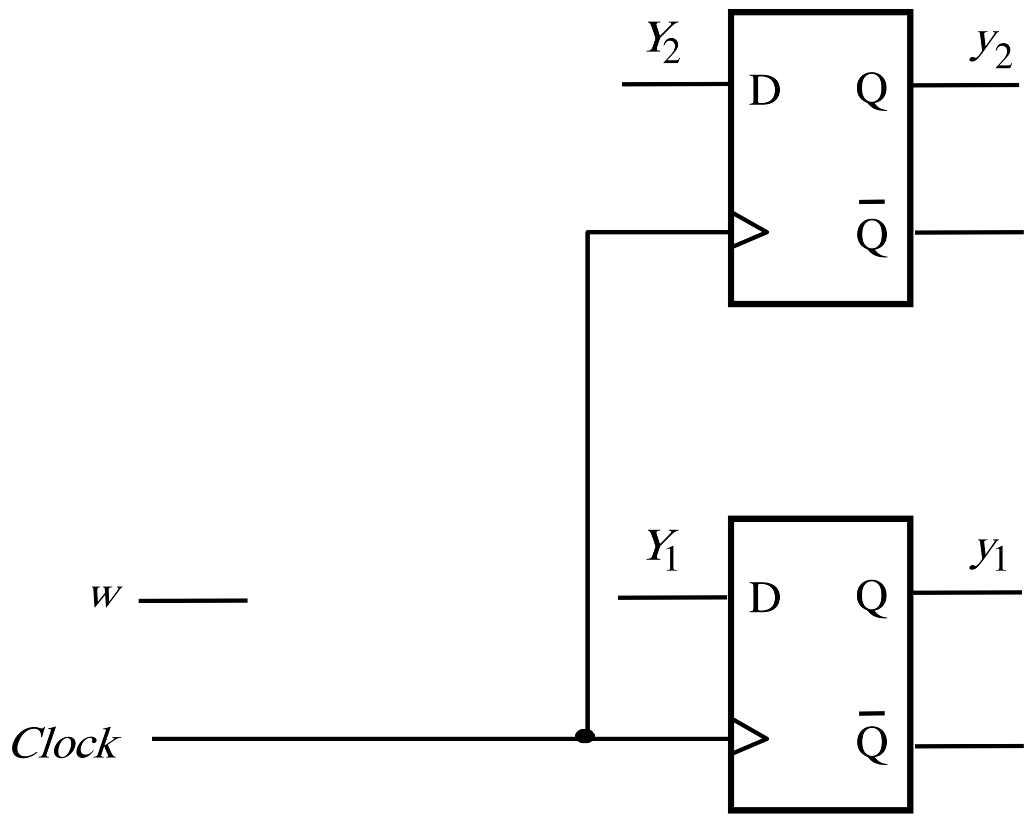
	y_2	y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$
A	0	0	0	0	0
B	0	1	0	0	1
D	1	0	0	1	0
C	1	1	1	0	0

$$R1_{out} = R2_{in} = y_1\, y_2$$

$$R1_{in} = R3_{out} = Done = \overline{y_1}\, y_2$$

$$R2_{out} = R3_{in} = y_1\, \overline{y_2}$$

Let's Complete the Circuit Diagram



$R1_{out} = R2_{in} = y_1\, y_2$

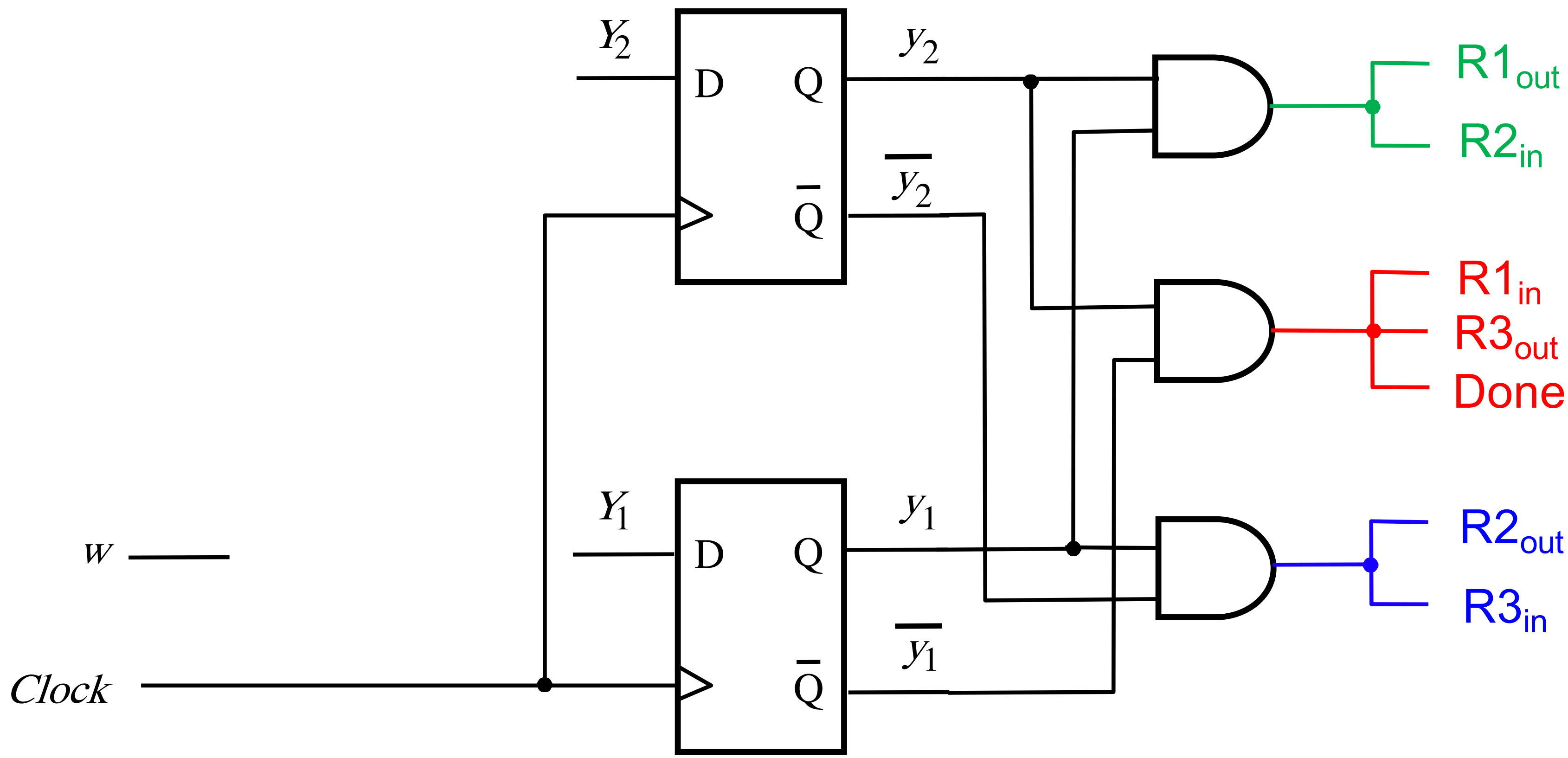
$R1_{in} = R3_{out} = Done = \overline{y_1}\, y_2$

$R2_{out} = R3_{in} = y_1\, \overline{y_2}$

$Y_1 = w\,\overline{y_2} + y_1\,\overline{y_2}$

$Y_2 = y_1$

Let's Complete the Circuit Diagram



$R1_{out} = R2_{in} = y_1\, y_2$

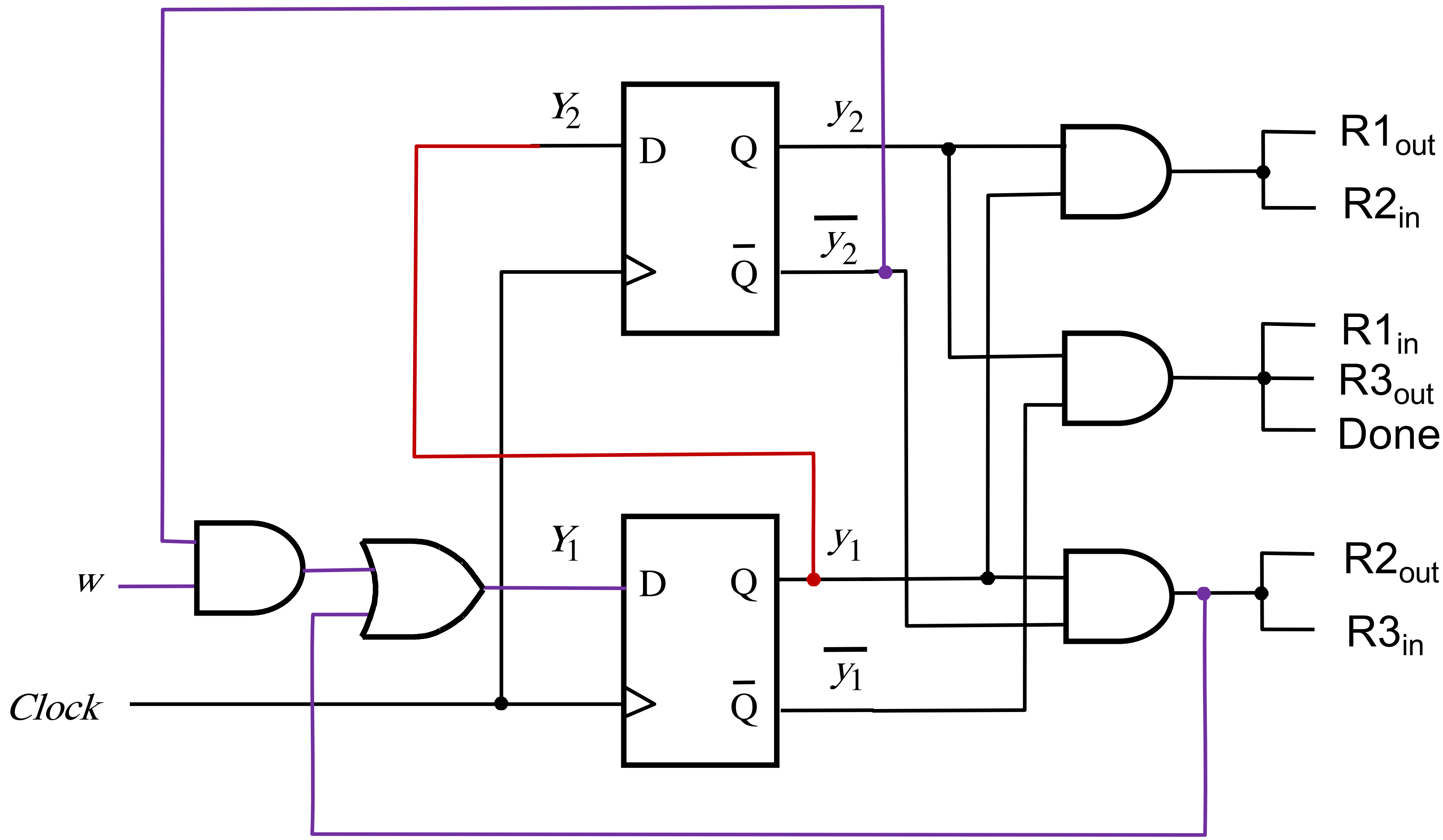
$R1_{in} = R3_{out} = Done = \overline{y_1}\, y_2$

$R2_{out} = R3_{in} = y_1\, \overline{y_2}$

$Y_1 = w\,\overline{y_2} + y_1\,\overline{y_2}$

$Y_2 = y_1$

Let's Complete the Circuit Diagram



$R1_{out} = R2_{in} = y_1\, y_2$

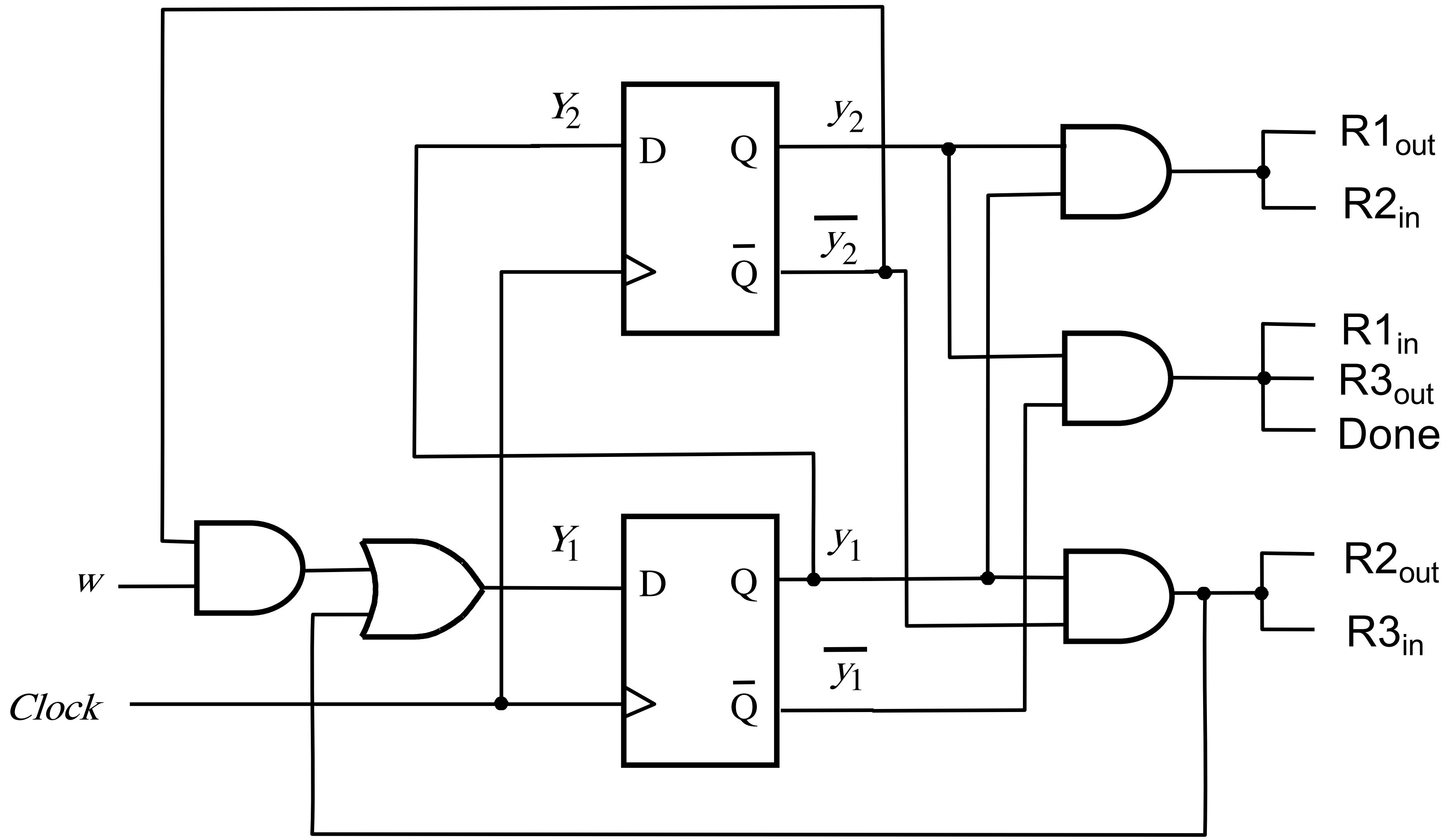
$R1_{in} = R3_{out} = \text{Done} = \overline{y_1}\, y_2$

$R2_{out} = R3_{in} = y_1\, \overline{y_2}$

$Y_1 = w\, \overline{y_2} + y_1\, \overline{y_2}$

$Y_2 = y_1$

Let's Complete the Circuit Diagram



$R1_{out} = R2_{in} = y_1\, y_2$

$R1_{in} = R3_{out} = \text{Done} = \overline{y_1}\, y_2$

$R2_{out} = R3_{in} = y_1\, \overline{y_2}$

$Y_1 = w\,\overline{y_2} + y_1\,\overline{y_2}$

$Y_2 = y_1$

Encoding #3:
A=0001, B=0010, C=0100, D=1000

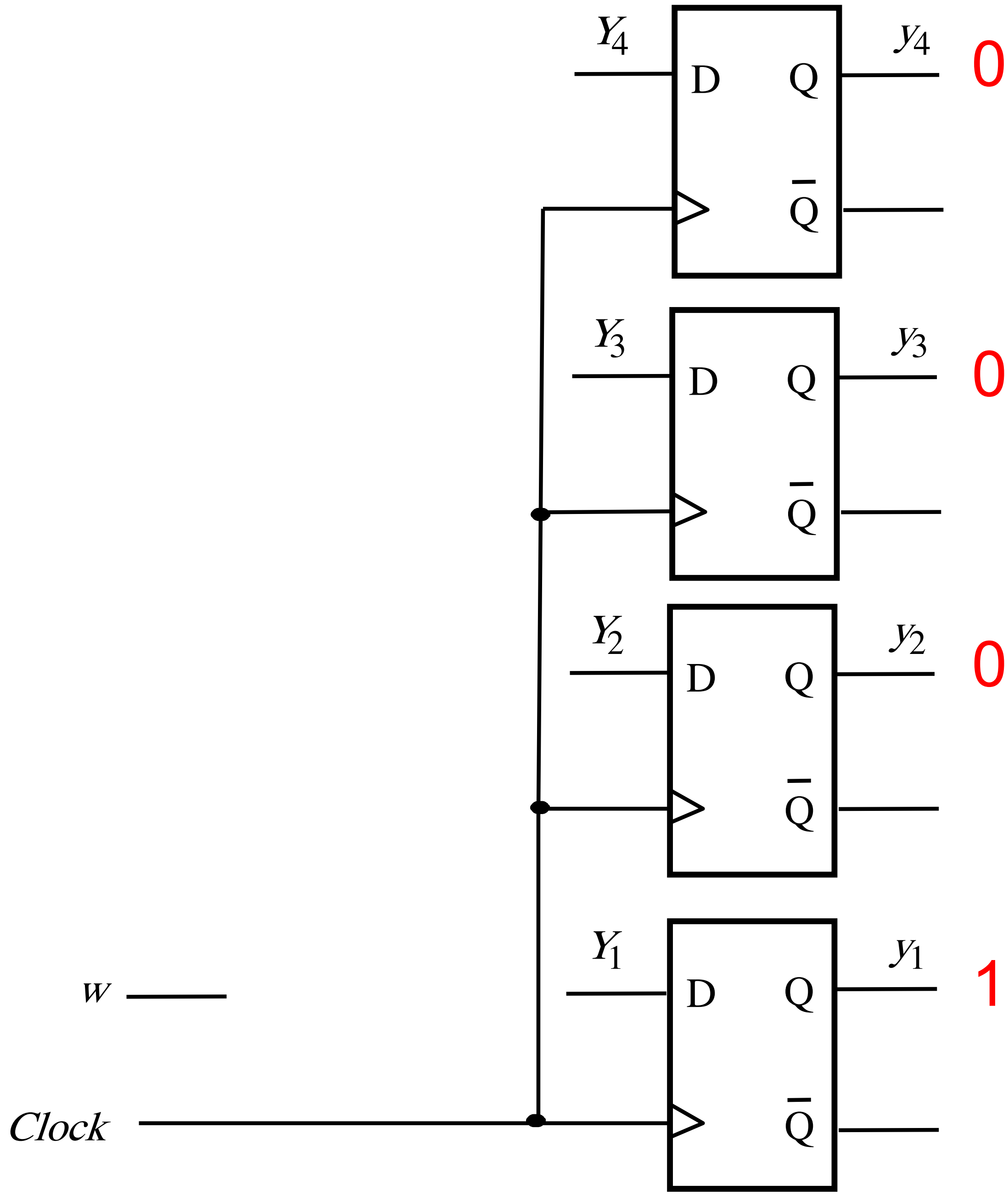
(One-Hot Encoding – Uses Four Flip-Flops)

One-Hot State Encoding

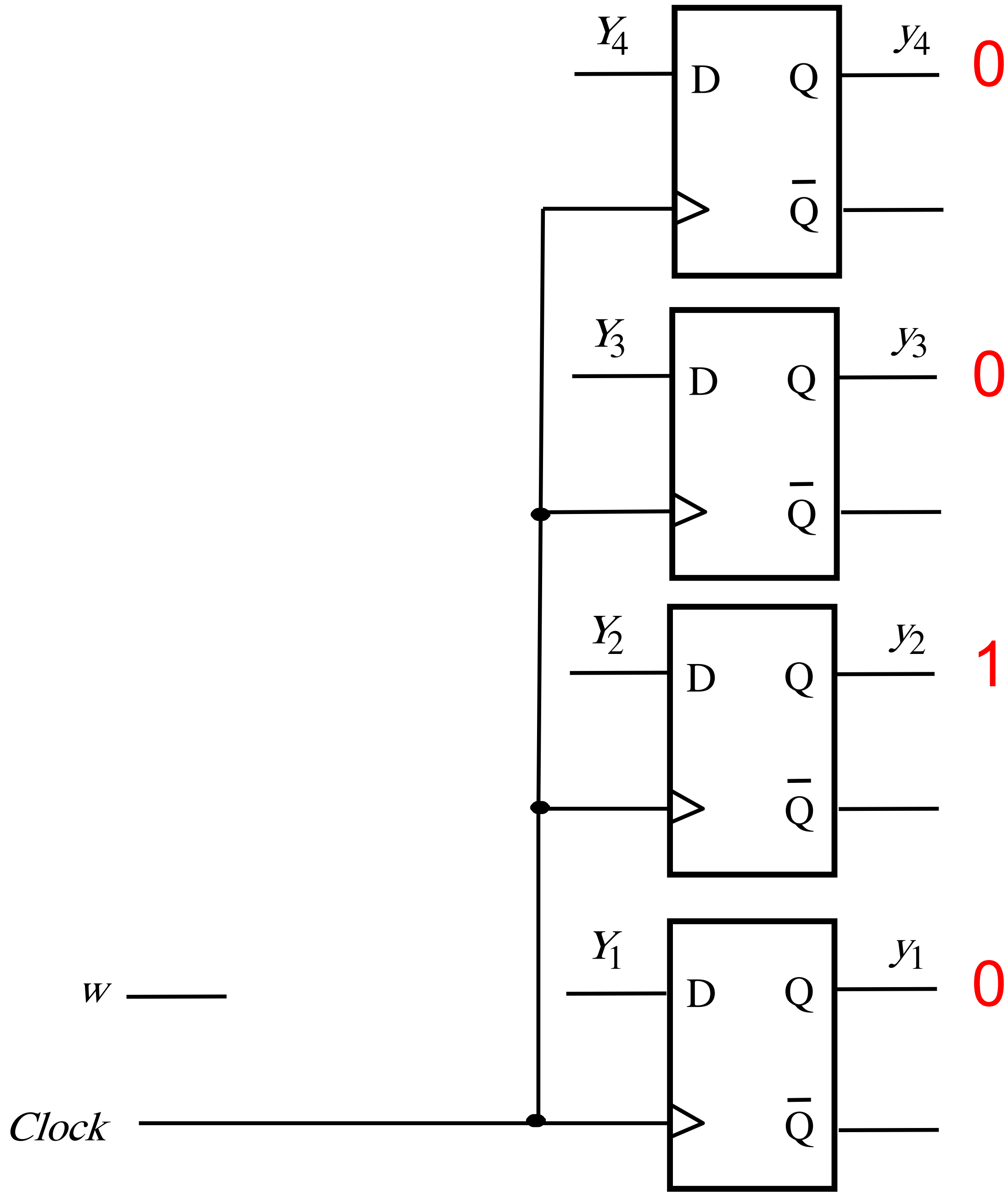
- So far, we have been encoding states in a way that minimizes the number of flip-flops.

- But sometimes we can decrease the complexity of our logic if we encode states more sparsely.

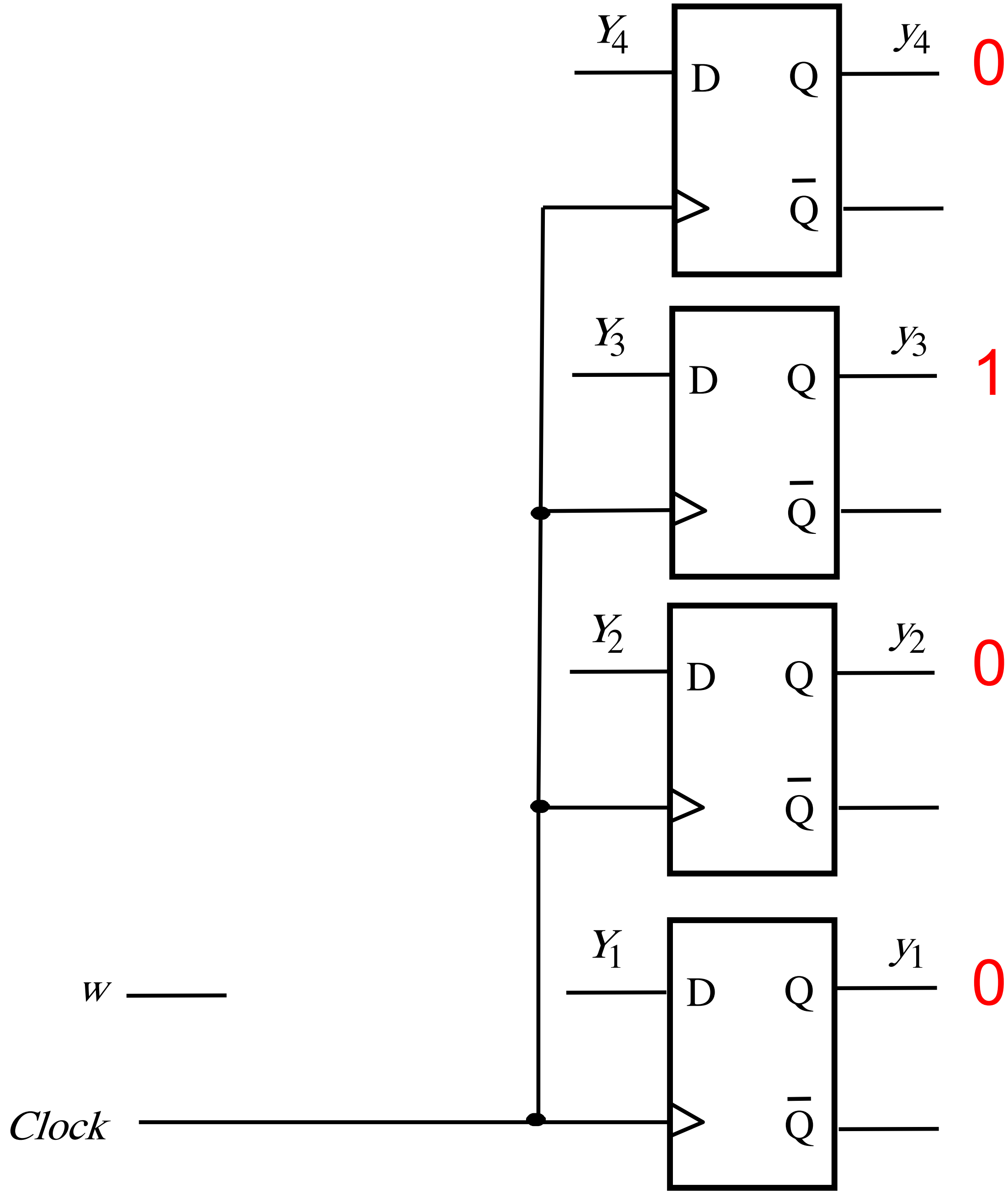
Encoding for State A



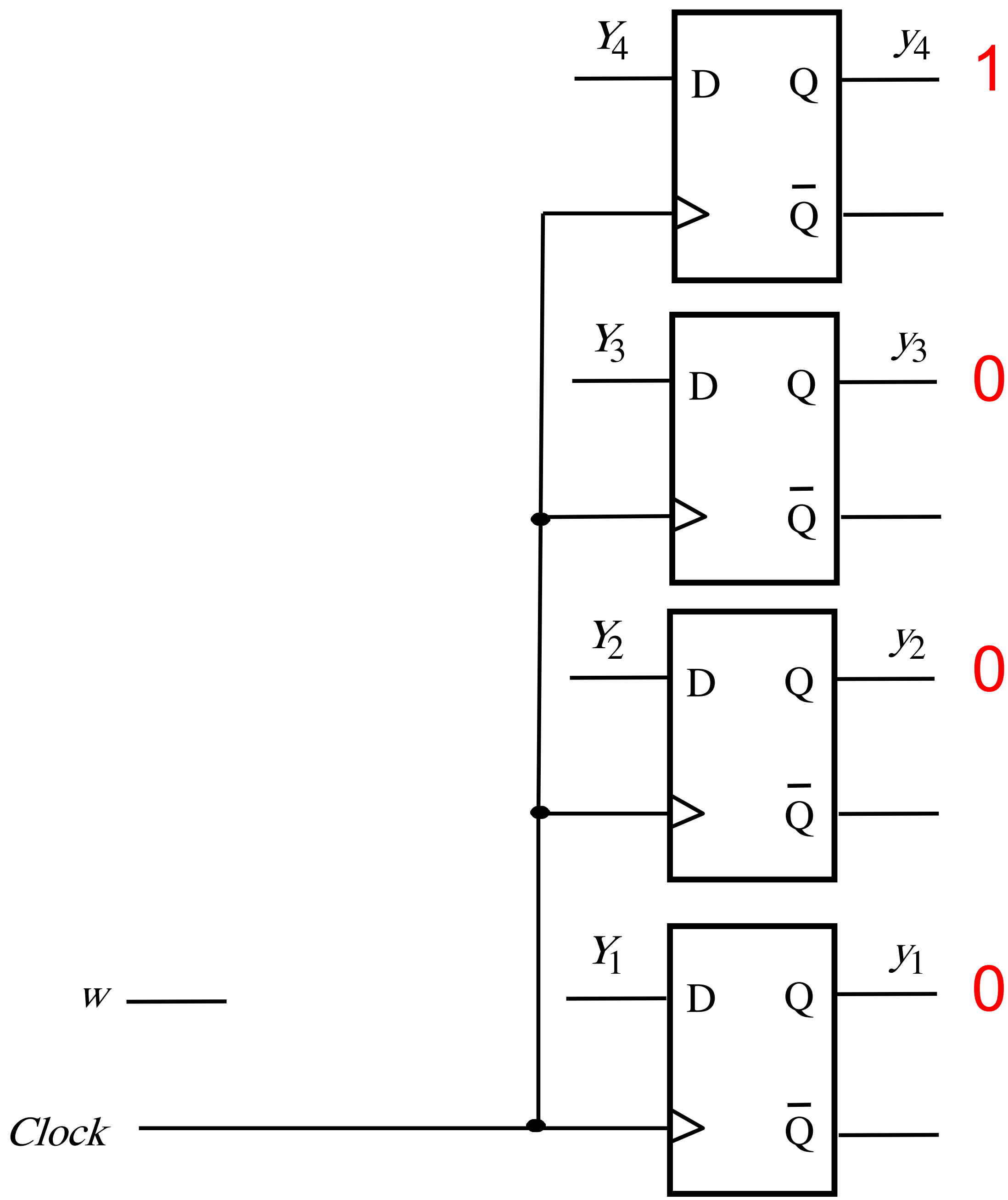
Encoding for State B



Encoding for State C



Encoding for State D



Register Swap Controller

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

Register Swap Controller

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

Let's use four flip-flops and the following one-hot state encoding scheme:

A = 0001

B = 0010

C = 0100

D = 1000

State Table (same as before)

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

State-Assigned Table

	Present State	Next State		Outputs						
		$w = 0$	$w = 1$							
	$y_4y_3y_2y_1$	$Y_4Y_3Y_2Y_1$	$Y_4Y_3Y_2Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A										
B										
C										
D										

[Figure 6.12 & 6.21 from the textbook]

State Table (same as before)

Present state	Next state $w = 0$	Next state $w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*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

State-Assigned Table

	Present State $y_4y_3y_2y_1$	Next State $w = 0$ $Y_4Y_3Y_2Y_1$	Next State $w = 1$ $Y_4Y_3Y_2Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	0 001									
B	0 010									
C	0 100									
D	1 000									

[Figure 6.12 & 6.21 from the textbook]

State Table (same as before)

Present state	Next state $w=0$	Next state $w=1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*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

State-Assigned Table

	Present State $y_4y_3y_2y_1$	Next State $w=0$ $Y_4Y_3Y_2Y_1$	Next State $w=1$ $Y_4Y_3Y_2Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	0 001	0001	0010							
B	0 010	0100	0100							
C	0 100	1000	1000							
D	1 000	0001	0001							

[Figure 6.12 & 6.21 from the textbook]

State Table (same as before)

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

State-Assigned Table

	Present State	Next State		Outputs						
		$w=0$	$w=1$							
	$y_4y_3y_2y_1$	$Y_4Y_3Y_2Y_1$	$Y_4Y_3Y_2Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	0 001	0001	0010	0	0	0	0	0	0	0
B	0 010	0100	0100	0	0	1	0	0	1	0
C	0 100	1000	1000	1	0	0	1	0	0	0
D	1 000	0001	0001	0	1	0	0	1	0	1

[Figure 6.12 & 6.21 from the textbook]

Let's Derive the Next-State Expressions

	Present State	Next State		Outputs						
		$w = 0$	$w = 1$							
	$y_4y_3y_2y_1$	$Y_4Y_3Y_2Y_1$	$Y_4Y_3Y_2Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	0 001	0001	0010	0	0	0	0	0	0	0
B	0 010	0100	0100	0	0	1	0	0	1	0
C	0 100	1000	1000	1	0	0	1	0	0	0
D	1 000	0001	0001	0	1	0	0	1	0	1

Let's Derive the Next-State Expressions

$$Y_1(w, y_4, y_3, y_2, y_1)$$
$$Y_2(w, y_4, y_3, y_2, y_1)$$
$$Y_3(w, y_4, y_3, y_2, y_1)$$
$$Y_4(w, y_4, y_3, y_2, y_1)$$

We need to do four 5-variable K-maps!

	Present State	Next State		Outputs						
		$w = 0$	$w = 1$							
	$y_4y_3y_2y_1$	$Y_4Y_3Y_2Y_1$	$Y_4Y_3Y_2Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	0 001	0001	0010	0	0	0	0	0	0	0
B	0 010	0100	0100	0	0	1	0	0	1	0
C	0 100	1000	1000	1	0	0	1	0	0	0
D	1 000	0001	0001	0	1	0	0	1	0	1

Let's Derive the Next-State Expressions

$$Y_1(w, y_4, y_3, y_2, y_1) = \overline{w}y_1 + y_4$$
$$Y_2(w, y_4, y_3, y_2, y_1) = wy_1$$
$$Y_3(w, y_4, y_3, y_2, y_1) = y_2$$
$$Y_4(w, y_4, y_3, y_2, y_1) = y_3$$

Or we can be smarter than that ☺

	Present State	Next State		Outputs						
		$w = 0$	$w = 1$							
	$y_4y_3y_2y_1$	$Y_4Y_3Y_2Y_1$	$Y_4Y_3Y_2Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	0 001	0001	0010	0	0	0	0	0	0	0
B	0 010	0100	0100	0	0	1	0	0	1	0
C	0 100	1000	1000	1	0	0	1	0	0	0
D	1 000	0001	0001	0	1	0	0	1	0	1

Let's Derive the Next-State Expressions

$$Y_1(w, y_4, y_3, y_2, y_1) = \overline{w}y_1 + y_4$$ (why?)

$$Y_2(w, y_4, y_3, y_2, y_1) = wy_1$$ (why?)

$$Y_3(w, y_4, y_3, y_2, y_1) = y_2$$ =1 only in B

$$Y_4(w, y_4, y_3, y_2, y_1) = y_3$$ =1 only in C

Or we can be smarter than that ☺

	Present State	Next State		Outputs						
		$w = 0$	$w = 1$							
	$y_4y_3y_2y_1$	$Y_4Y_3Y_2Y_1$	$Y_4Y_3Y_2Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	0 001	0001	0010	0	0	0	0	0	0	0
B	0 010	0100	0100	0	0	1	0	0	1	0
C	0 100	1000	1000	1	0	0	1	0	0	0
D	1 000	0001	0001	0	1	0	0	1	0	1

Let's Derive the Output Expressions

	Present State	Next State		Outputs						
		$w = 0$	$w = 1$							
	$y_4 y_3 y_2 y_1$	$Y_4 Y_3 Y_2 Y_1$	$Y_4 Y_3 Y_2 Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	0 001	0001	0010	0	0	0	0	0	0	0
B	0 010	0100	0100	0	0	1	0	0	1	0
C	0 100	1000	1000	1	0	0	1	0	0	0
D	1 000	0001	0001	0	1	0	0	1	0	1

Let's Derive the Output Expressions

$R1_{out}(y_4, y_3, y_2, y_1)$
$R1_{in}(y_4, y_3, y_2, y_1)$
$R2_{out}(y_4, y_3, y_2, y_1)$
$R2_{in}(y_4, y_3, y_2, y_1)$
$R3_{out}(y_4, y_3, y_2, y_1)$
$R3_{in}(y_4, y_3, y_2, y_1)$
$Done(y_4, y_3, y_2, y_1)$

We need to do seven 4-variable K-maps!

	Present State	Next State		Outputs						
		$w = 0$	$w = 1$							
	$y_4y_3y_2y_1$	$Y_4Y_3Y_2Y_1$	$Y_4Y_3Y_2Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	0 001	0001	0010	0	0	0	0	0	0	0
B	0 010	0100	0100	0	0	1	0	0	1	0
C	0 100	1000	1000	1	0	0	1	0	0	0
D	1 000	0001	0001	0	1	0	0	1	0	1

Let's Derive the Output Expressions

$R1_{out}(y_4, y_3, y_2, y_1) = y_3$ *equal to 1 only in State C*
$R1_{in}(y_4, y_3, y_2, y_1) = y_4$ *equal to 1 only in State D*
$R2_{out}(y_4, y_3, y_2, y_1) = y_2$ *equal to 1 only in State B*
$R2_{in}(y_4, y_3, y_2, y_1) = y_3$ *equal to 1 only in State C*
$R3_{out}(y_4, y_3, y_2, y_1) = y_4$ *equal to 1 only in State D*
$R3_{in}(y_4, y_3, y_2, y_1) = y_2$ *equal to 1 only in State B*
$Done(y_4, y_3, y_2, y_1) = y_4$ *equal to 1 only in State D*

Or we can be smarter than that by exploiting the one-hot encoded property

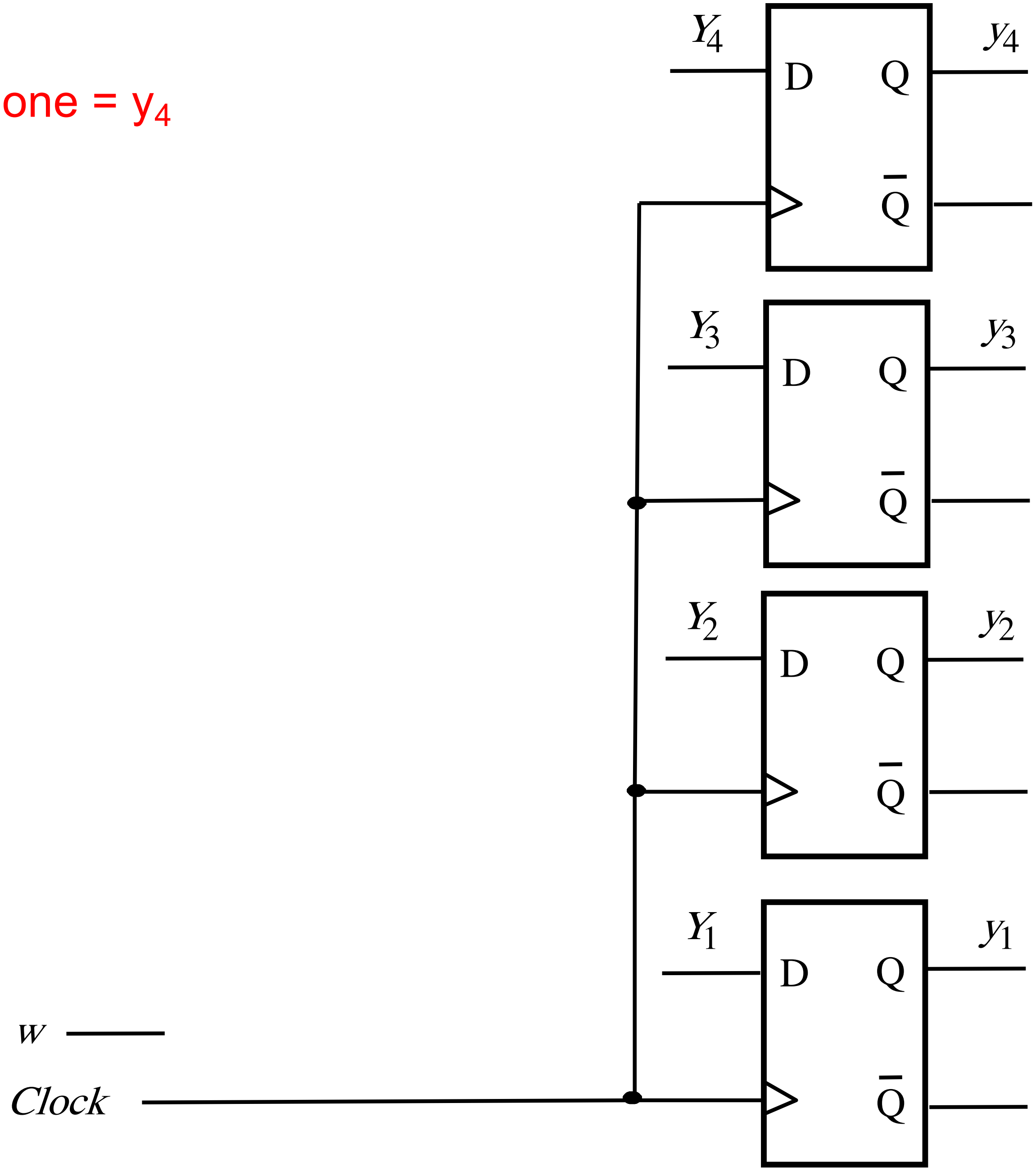
	Present State	Next State		Outputs						
		$w = 0$	$w = 1$							
	$y_4y_3y_2y_1$	$Y_4Y_3Y_2Y_1$	$Y_4Y_3Y_2Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	*Done*
A	0 001	0001	0010	0	0	0	0	0	0	0
B	0 010	0100	0100	0	0	1	0	0	1	0
C	0 100	1000	1000	1	0	0	1	0	0	0
D	1 000	0001	0001	0	1	0	0	1	0	1

Let's Complete the Circuit Diagram

$R1_{out} = R2_{in} = y_3$

$R1_{in} = R3_{out} = Done = y_4$

$R2_{out} = R3_{in} = y_2$



$Y_1 = \overline{w}\, y_1 + y_4$

$Y_2 = w\, y_1$

$Y_3 = y_2$

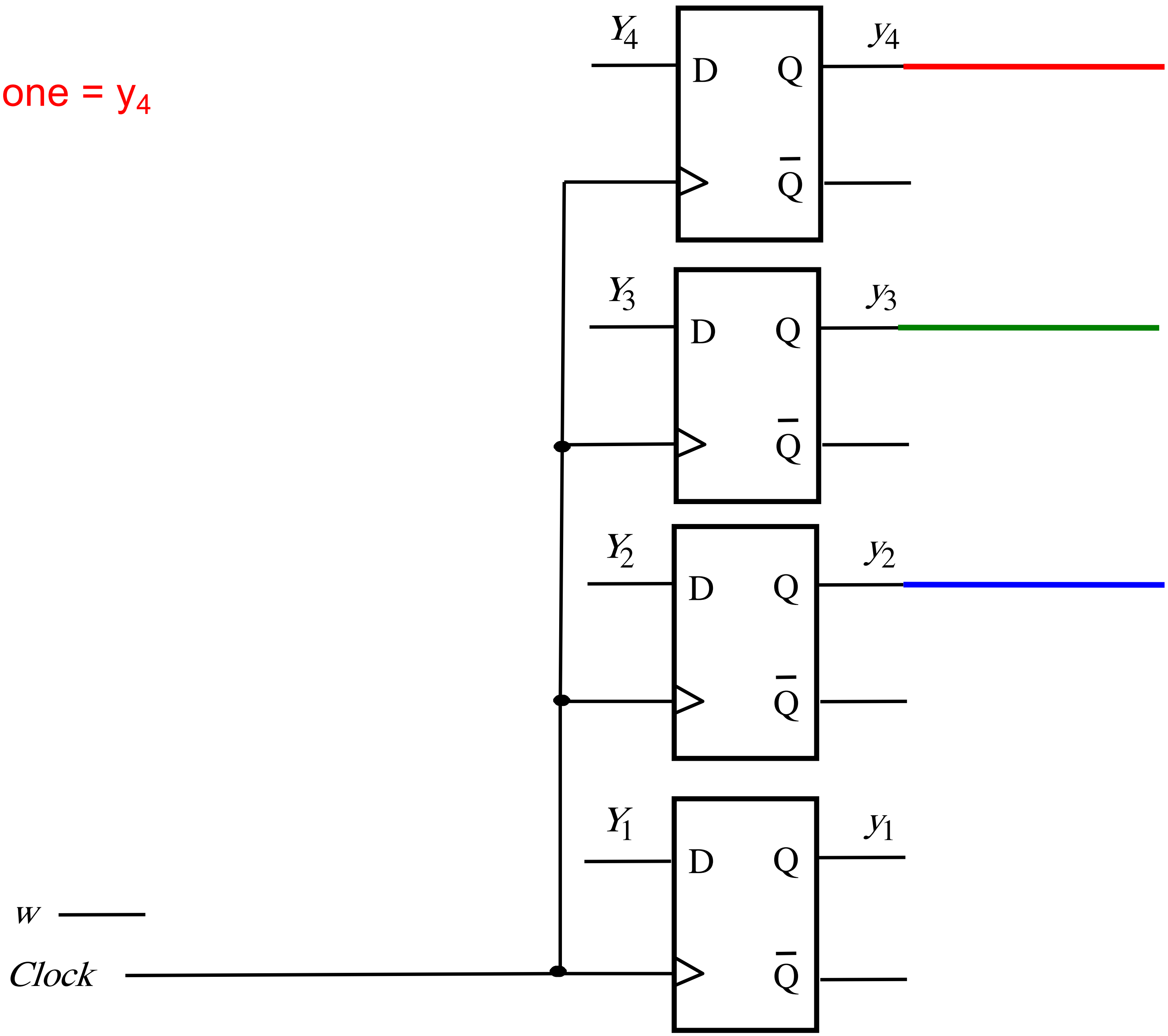
$Y_4 = y_3$

Let's Complete the Circuit Diagram

$R1_{out} = R2_{in} = y_3$

$R1_{in} = R3_{out} = Done = y_4$

$R2_{out} = R3_{in} = y_2$



$Y_1 = \overline{w}\, y_1 + y_4$

$Y_2 = w\, y_1$

$Y_3 = y_2$

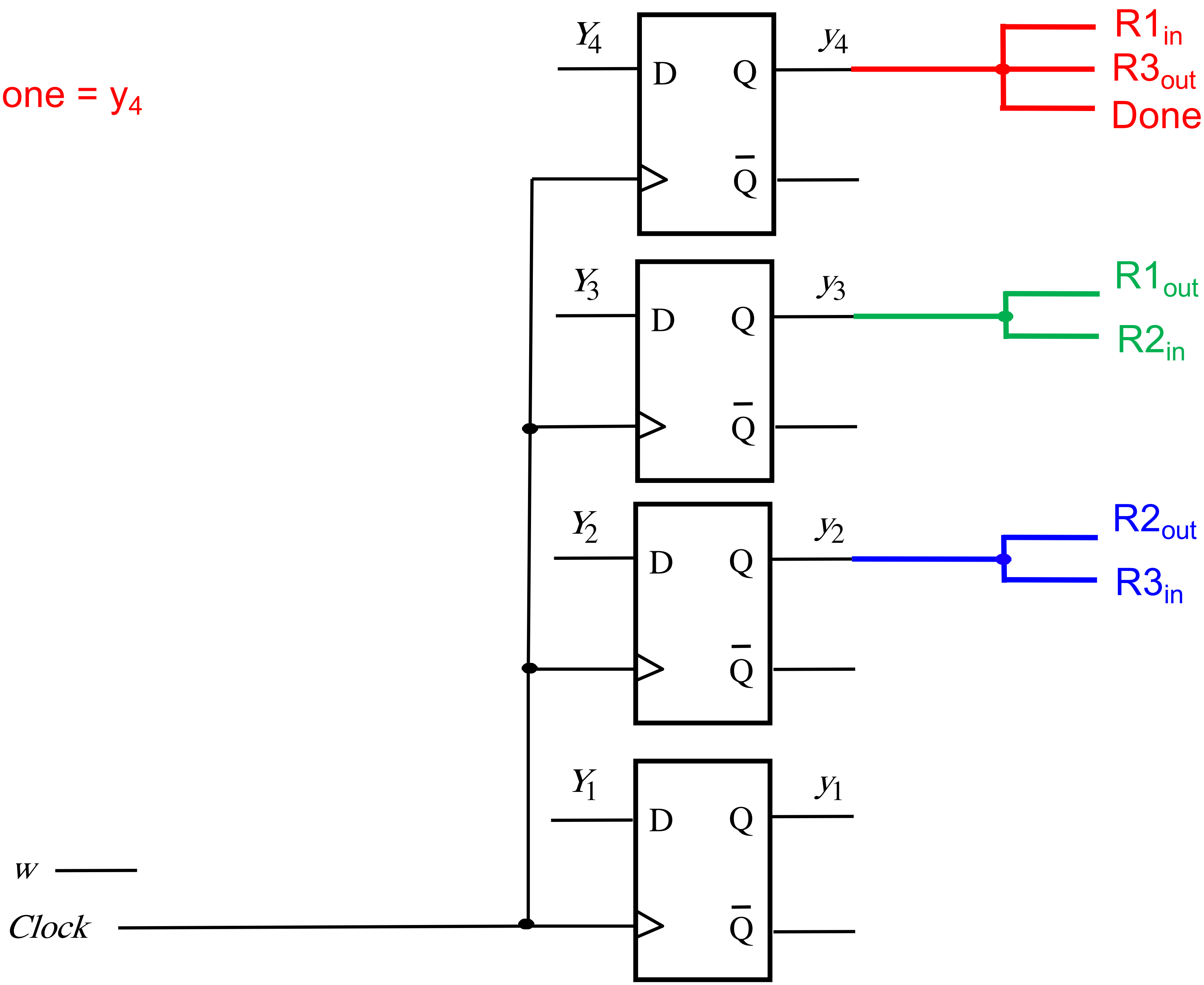
$Y_4 = y_3$

Let's Complete the Circuit Diagram

$R1_{out} = R2_{in} = y_3$

$R1_{in} = R3_{out} = Done = y_4$

$R2_{out} = R3_{in} = y_2$



$Y_1 = \overline{w}\, y_1 + y_4$

$Y_2 = w\, y_1$

$Y_3 = y_2$

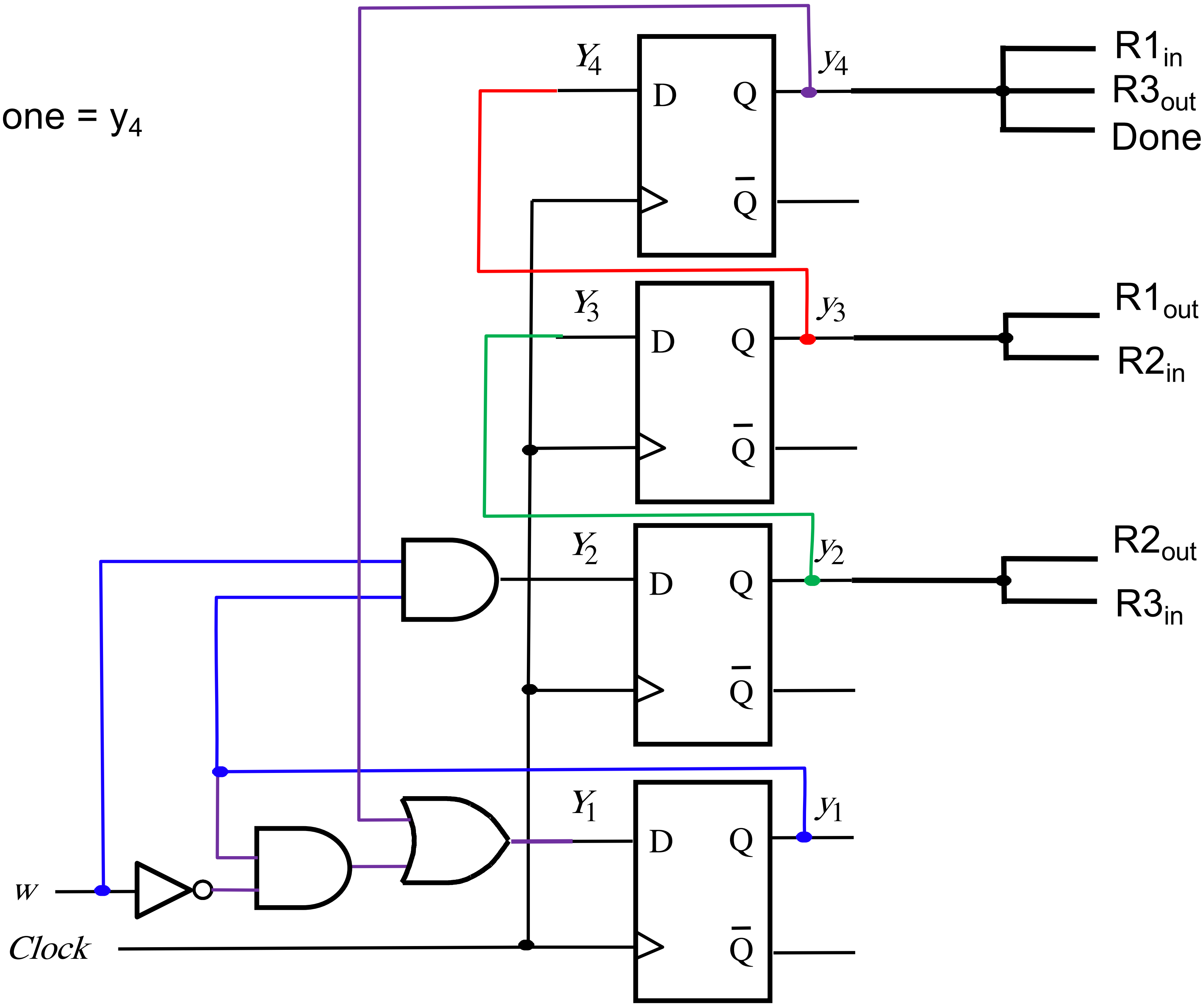
$Y_4 = y_3$

Let's Complete the Circuit Diagram

$R1_{out} = R2_{in} = y_3$

$R1_{in} = R3_{out} = Done = y_4$

$R2_{out} = R3_{in} = y_2$



$Y_1 = \overline{w}\, y_1 + y_4$

$Y_2 = w\, y_1$

$Y_3 = y_2$

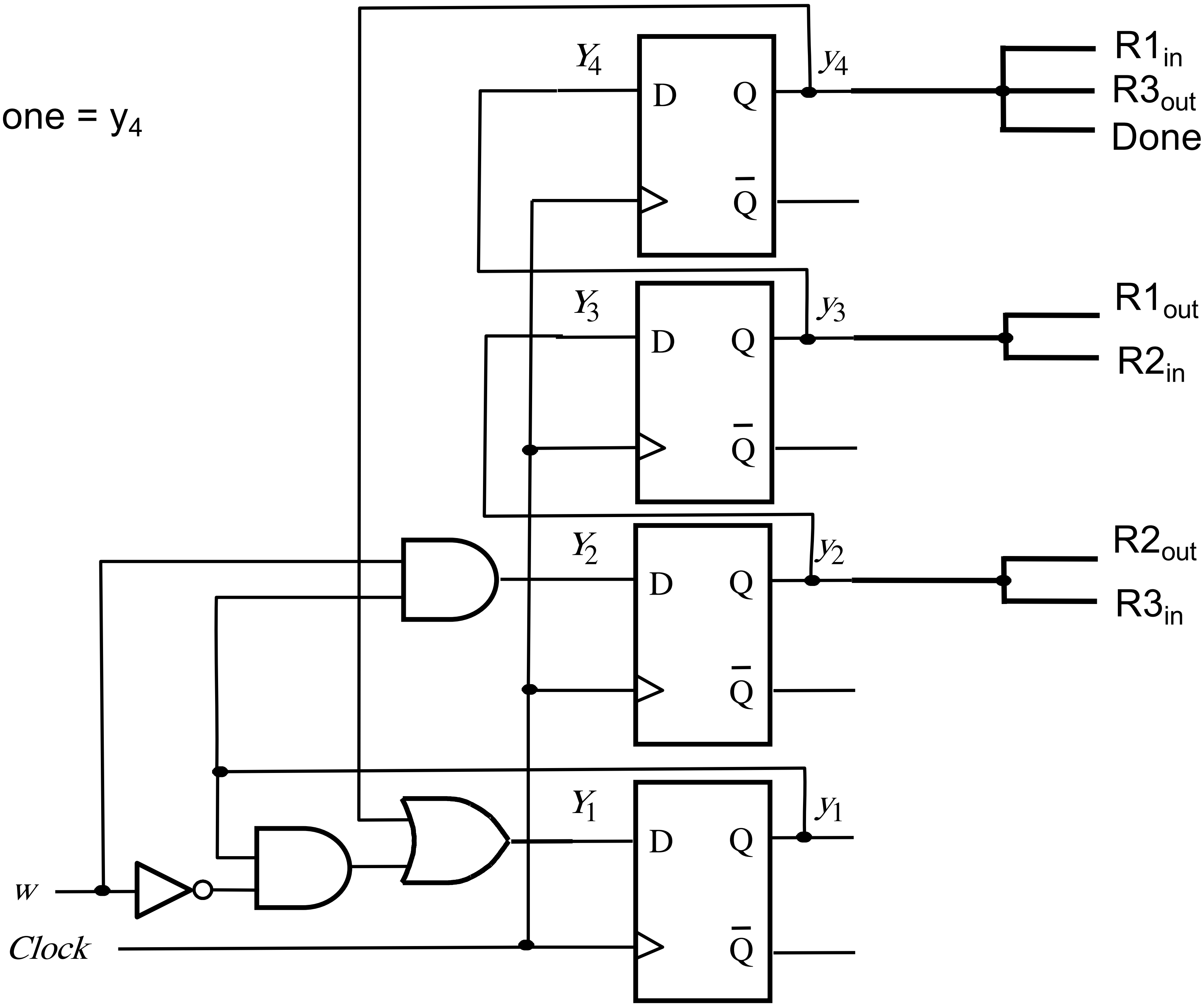
$Y_4 = y_3$

Let's Complete the Circuit Diagram

$R1_{out} = R2_{in} = y_3$

$R1_{in} = R3_{out} = Done = y_4$

$R2_{out} = R3_{in} = y_2$



$Y_1 = \overline{w}\, y_1 + y_4$

$Y_2 = w\, y_1$

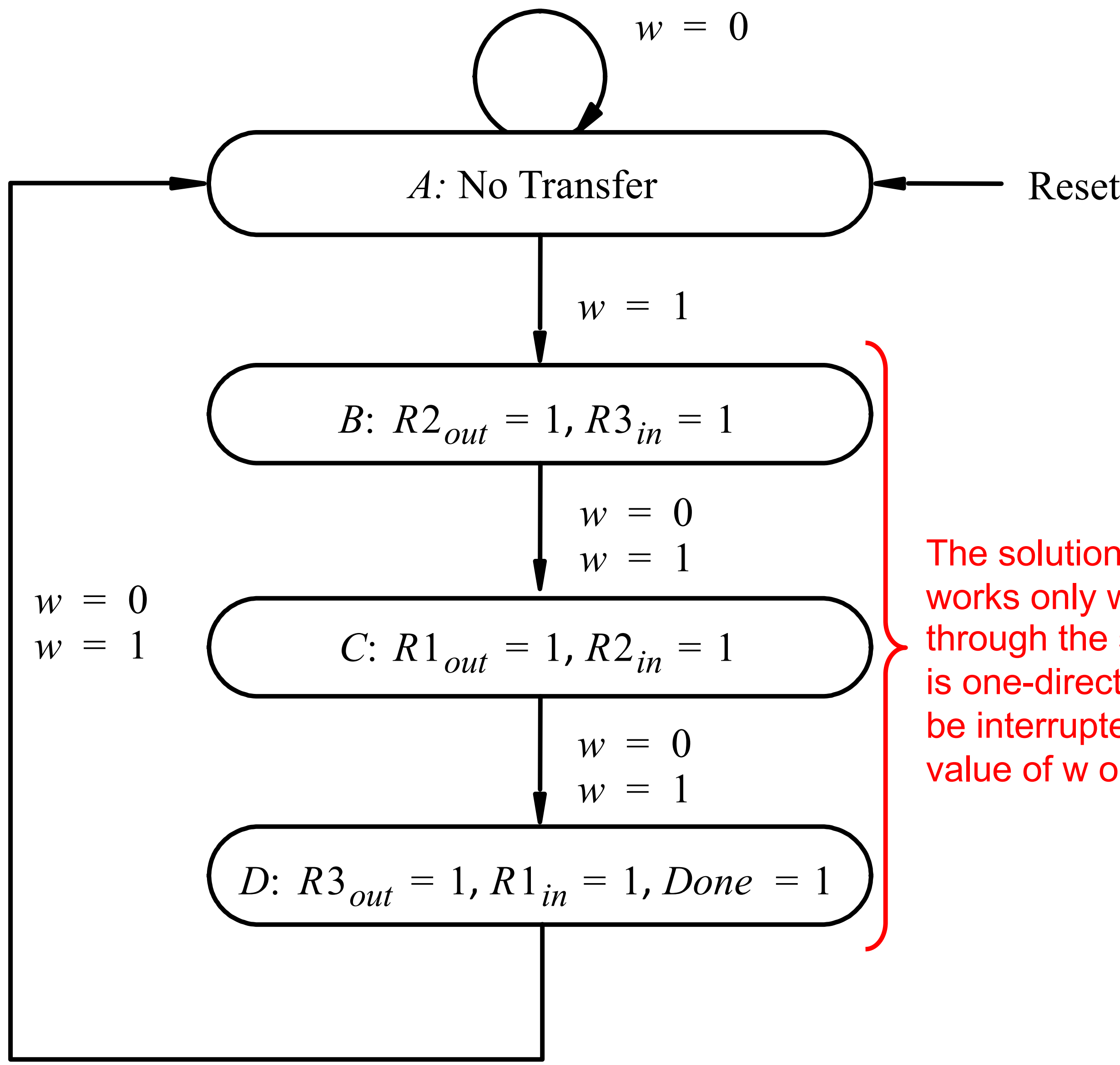
$Y_3 = y_2$

$Y_4 = y_3$

Encoding #4:
A=0001, B=0010, C=0100, D=1000

(same as before, but shows an alternative implementation with a 4-bit ring counter)

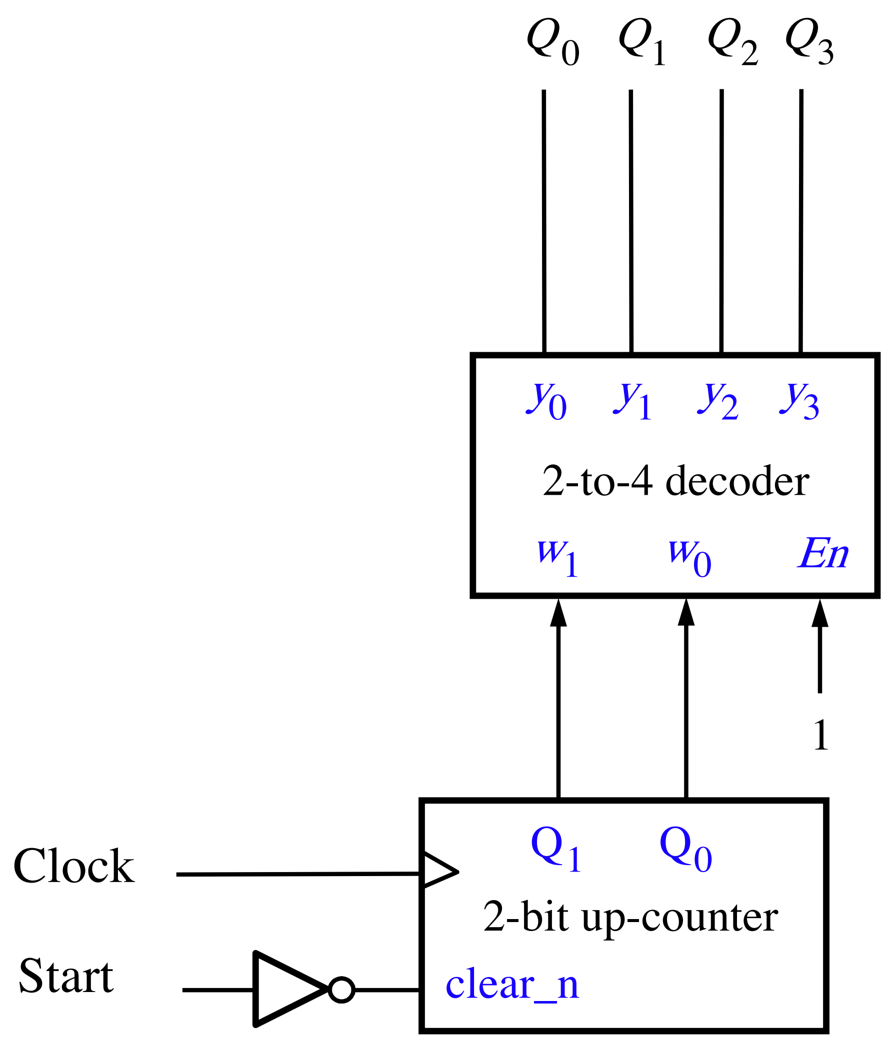
Exploit the Structure of the FSM



The solution with the ring counter works only when the traversal through the states of the FSM is one-directional and cannot be interrupted by the value of w once it has started.

[Figure 6.11 from the textbook]

Alternative version of a 4-bit ring counter



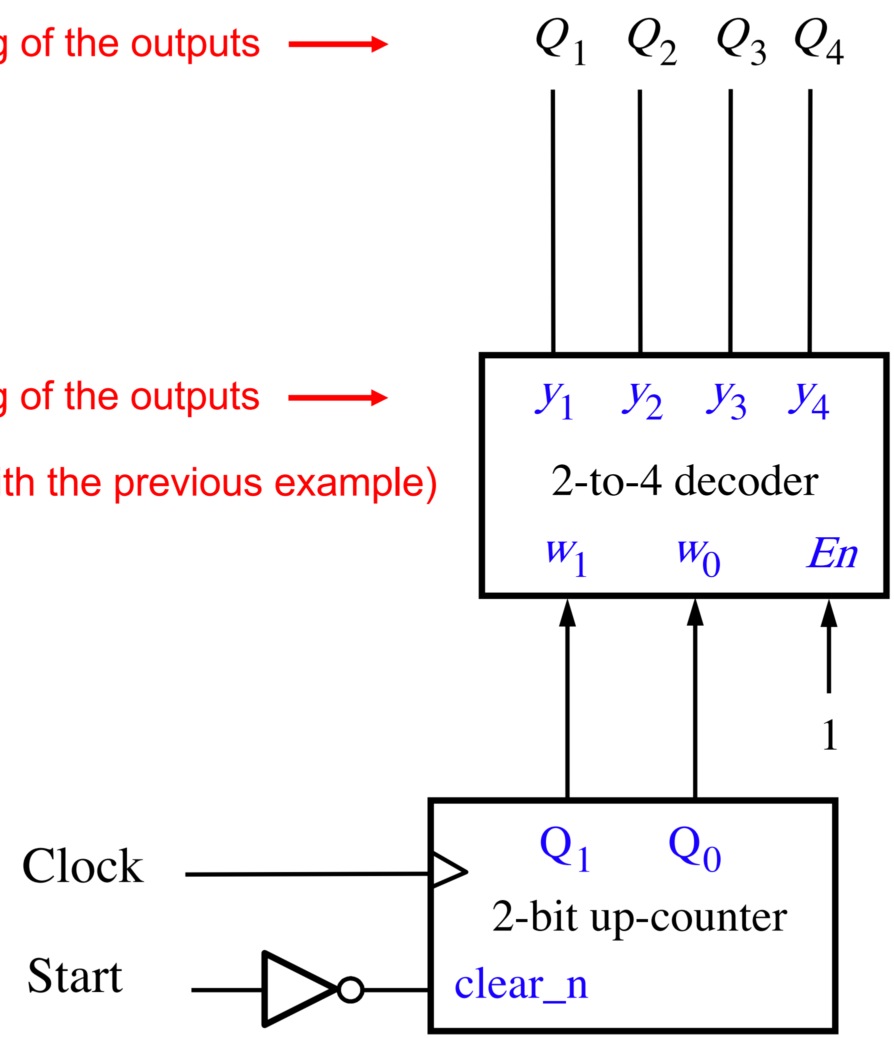
[Figure 5.28b from the textbook]

Alternative version of a 4-bit ring counter

Switch to 1-based indexing of the outputs

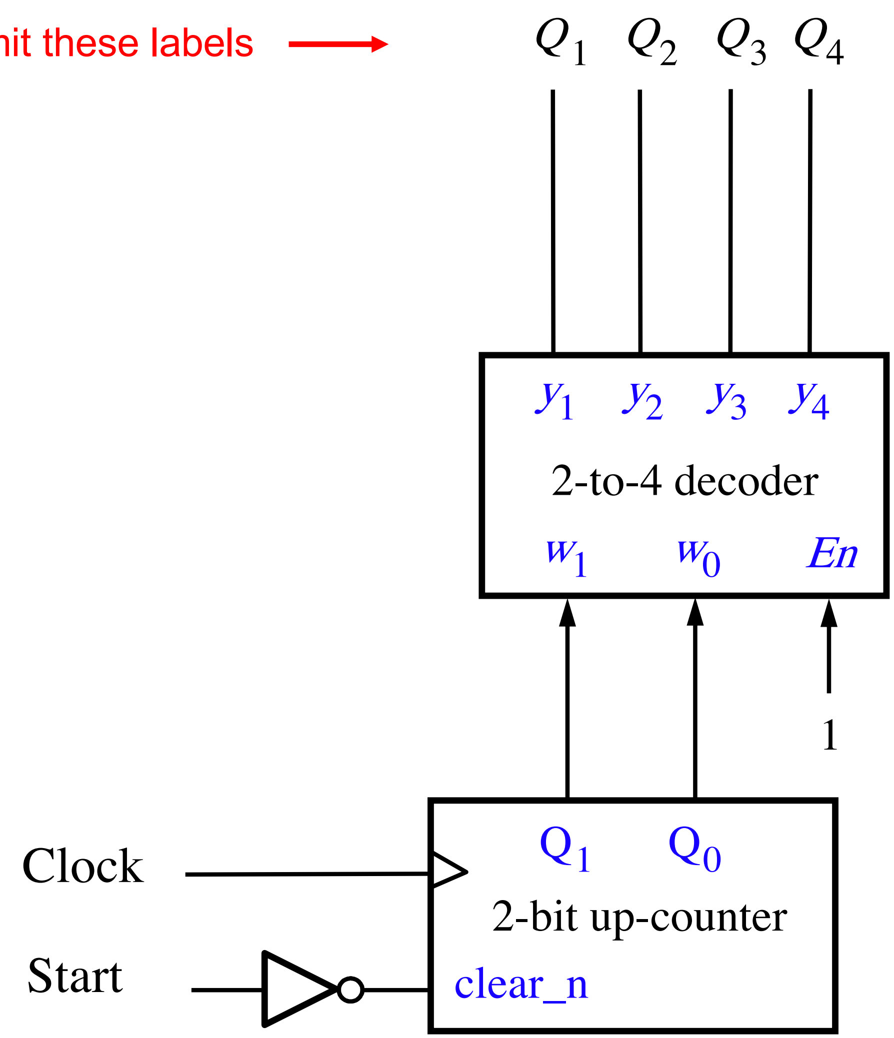
Switch to 1-based indexing of the outputs

(this is done to be consistent with the previous example)

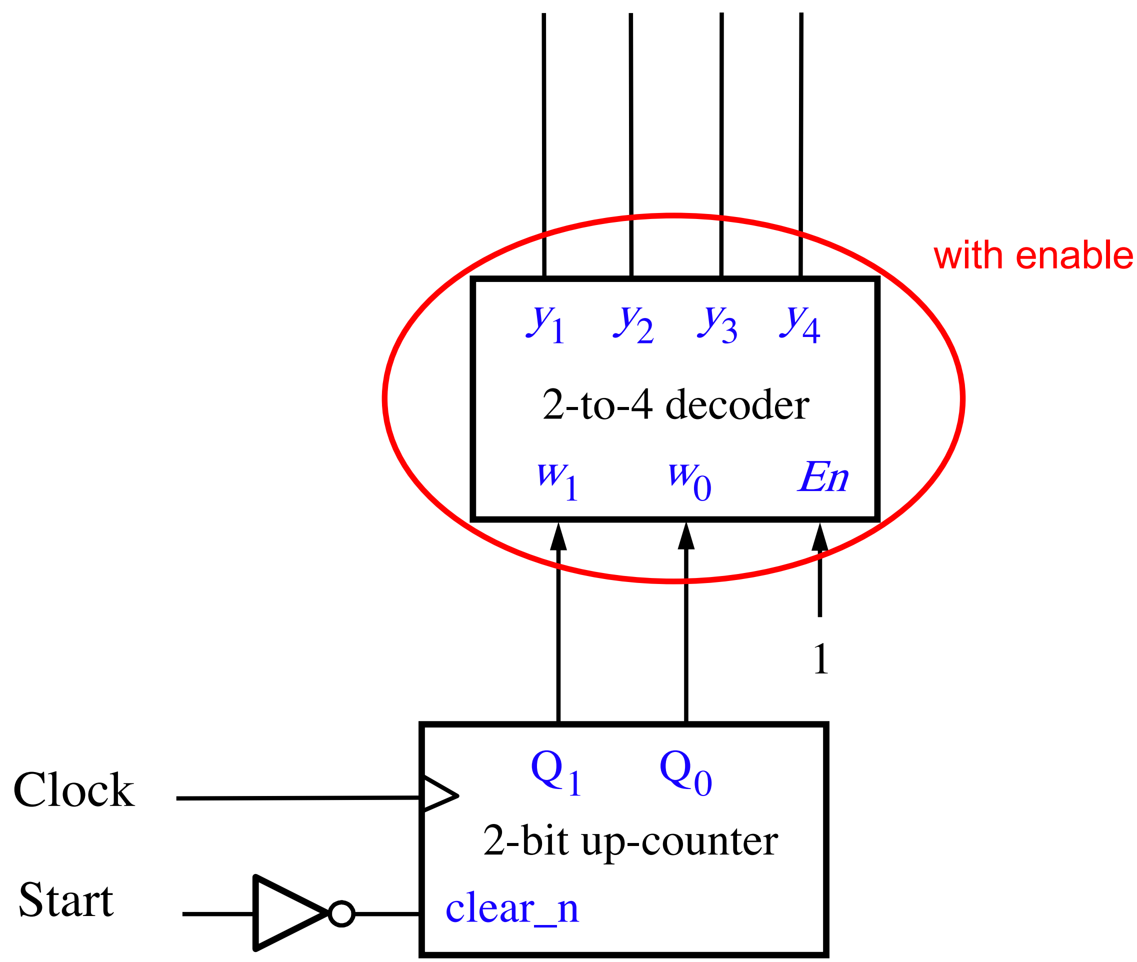


Alternative version of a 4-bit ring counter

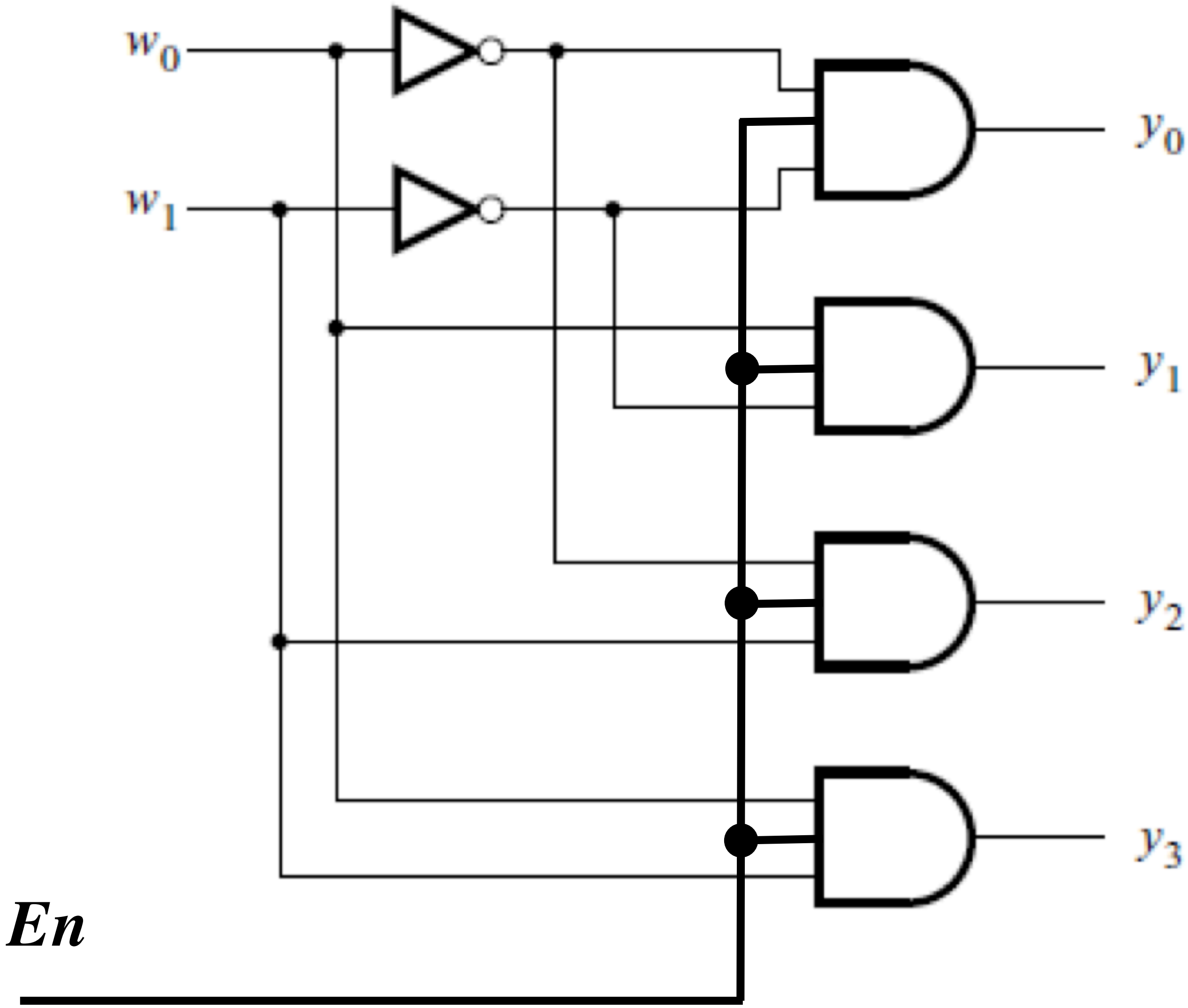
Also, omit these labels



Alternative version of a 4-bit ring counter

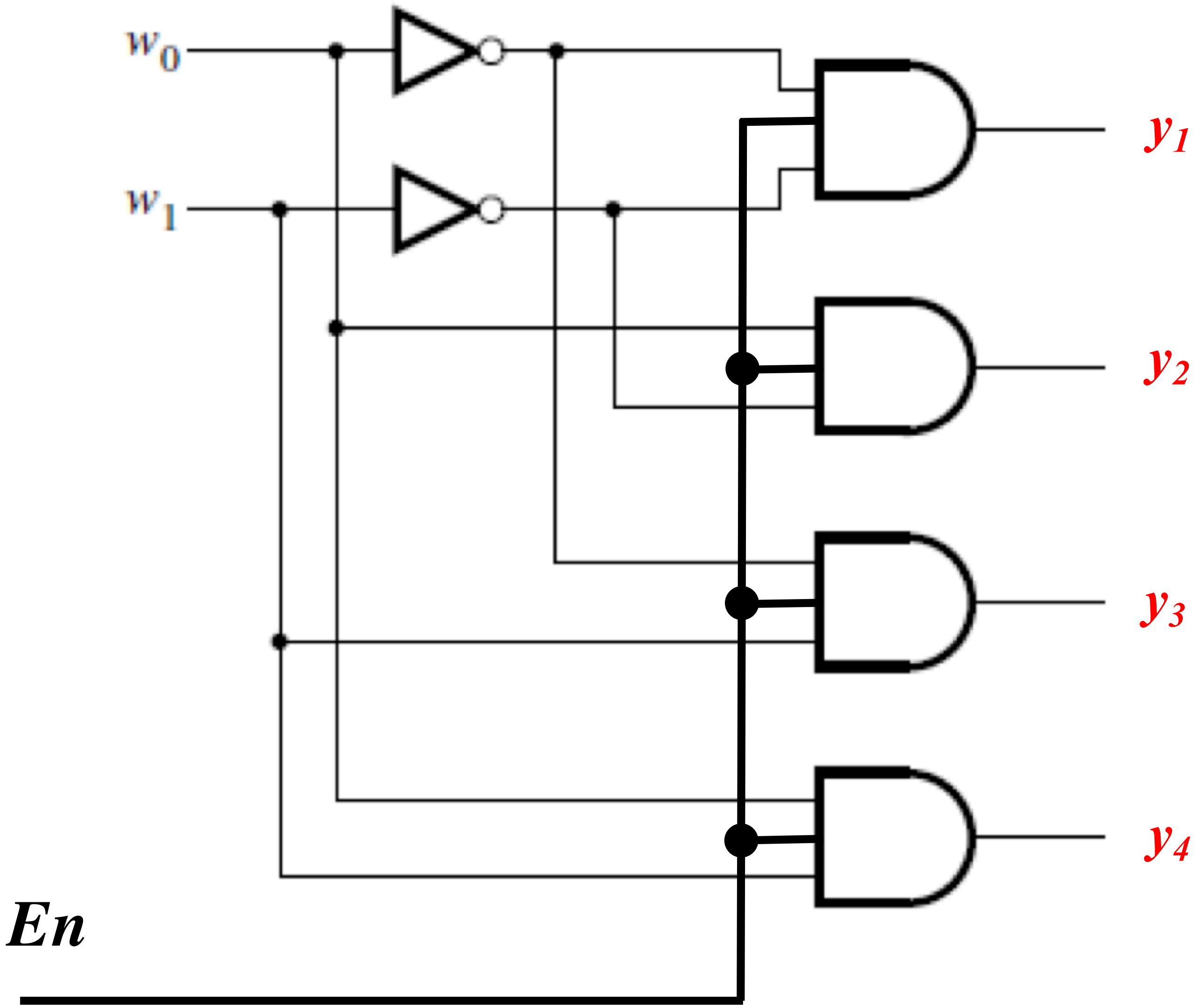


2-to-4 Decoder with Enable Input



[Figure 4.14c from the textbook]

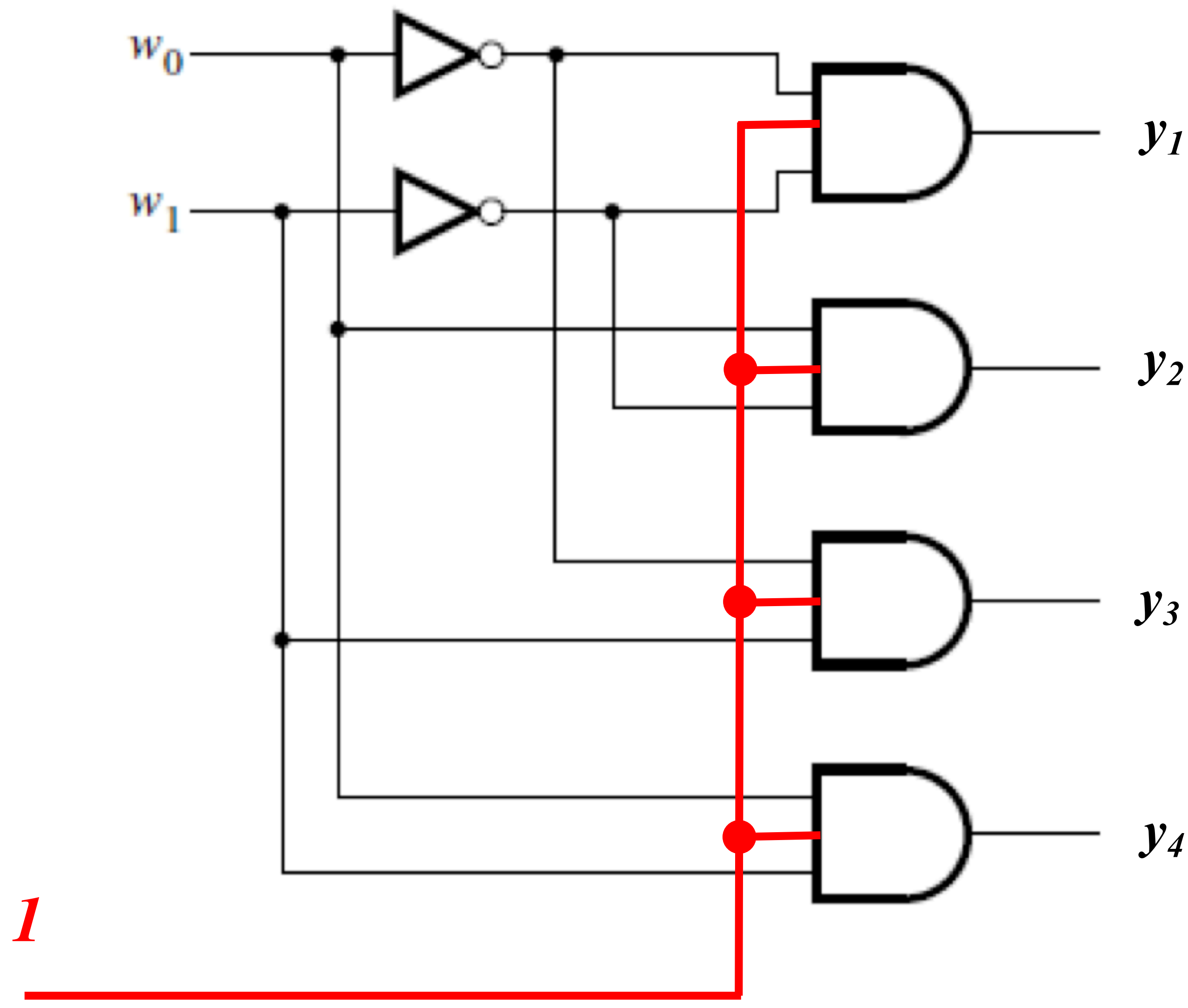
2-to-4 Decoder with Enable Input



Switch to 1-based indexing of the outputs

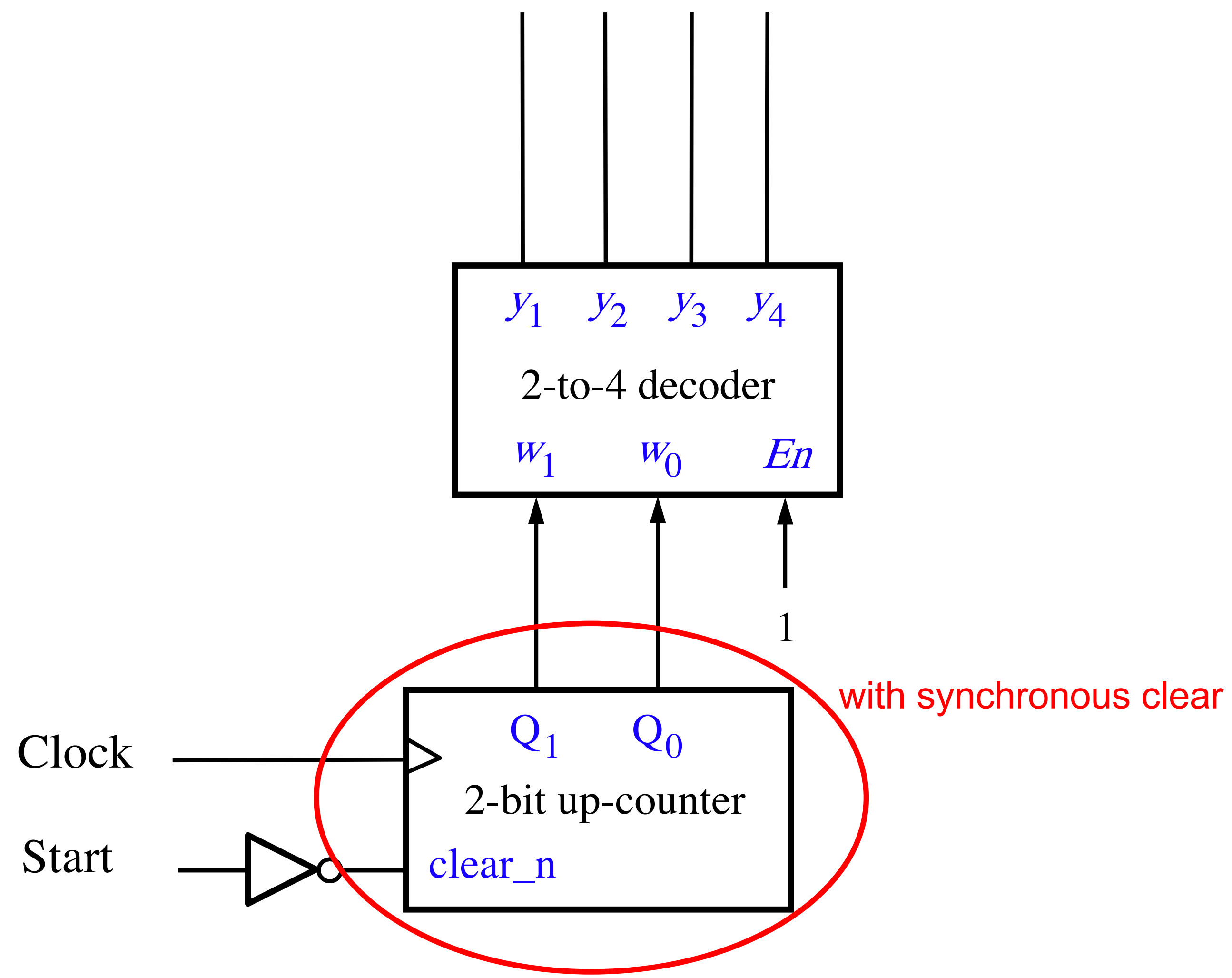
(this is done to be consistent with the previous example)

2-to-4 Decoder with Enable Input

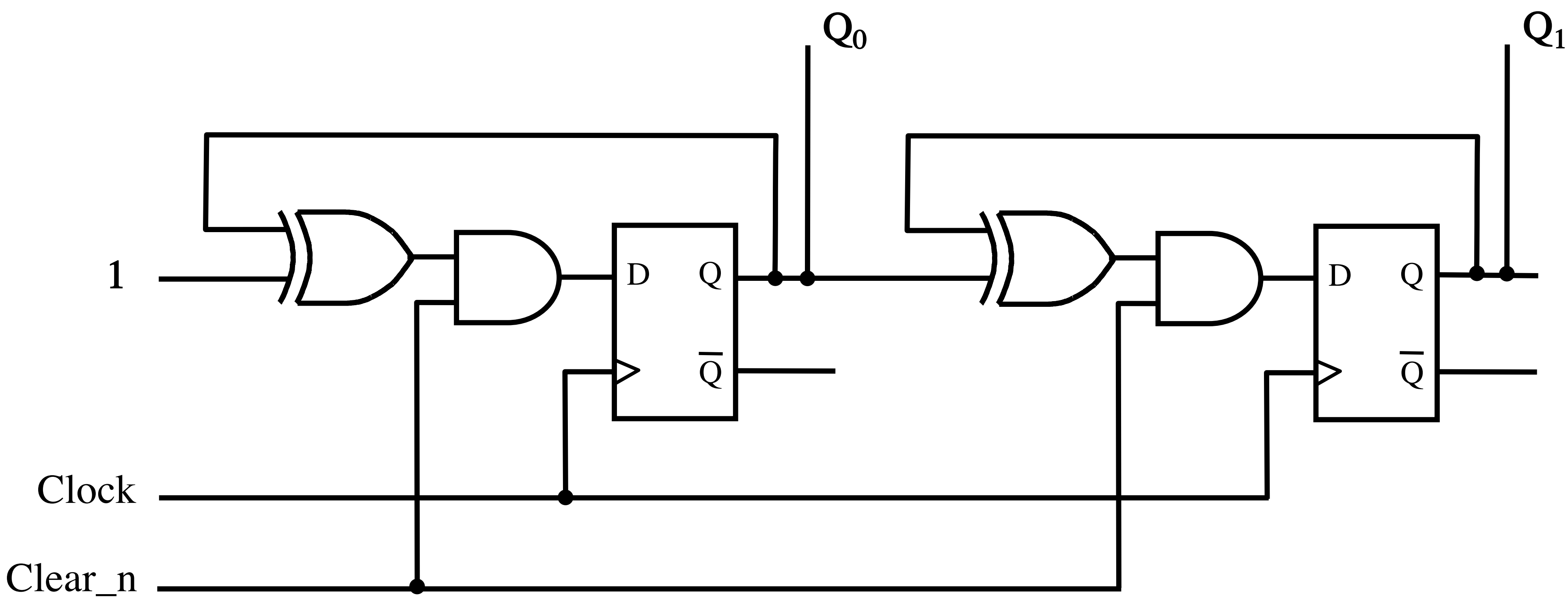


(always enabled in this example)

Alternative version of a 4-bit ring counter

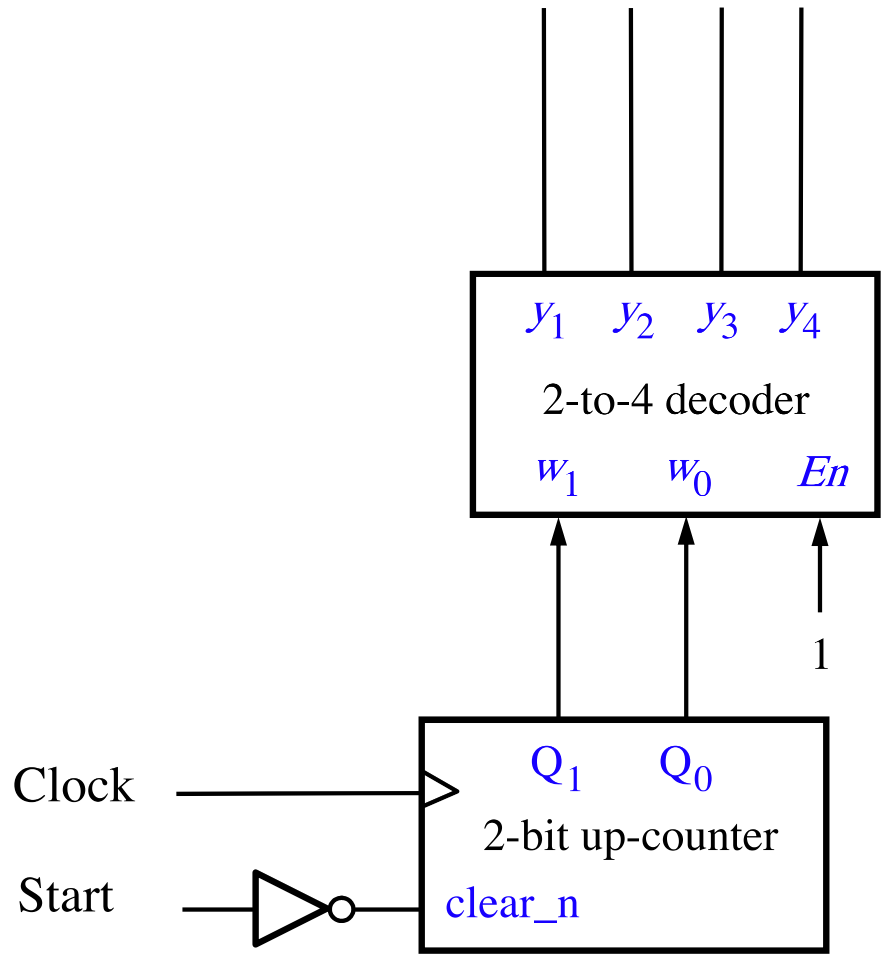


2-Bit Synchronous Up-Counter (with synchronous clear)

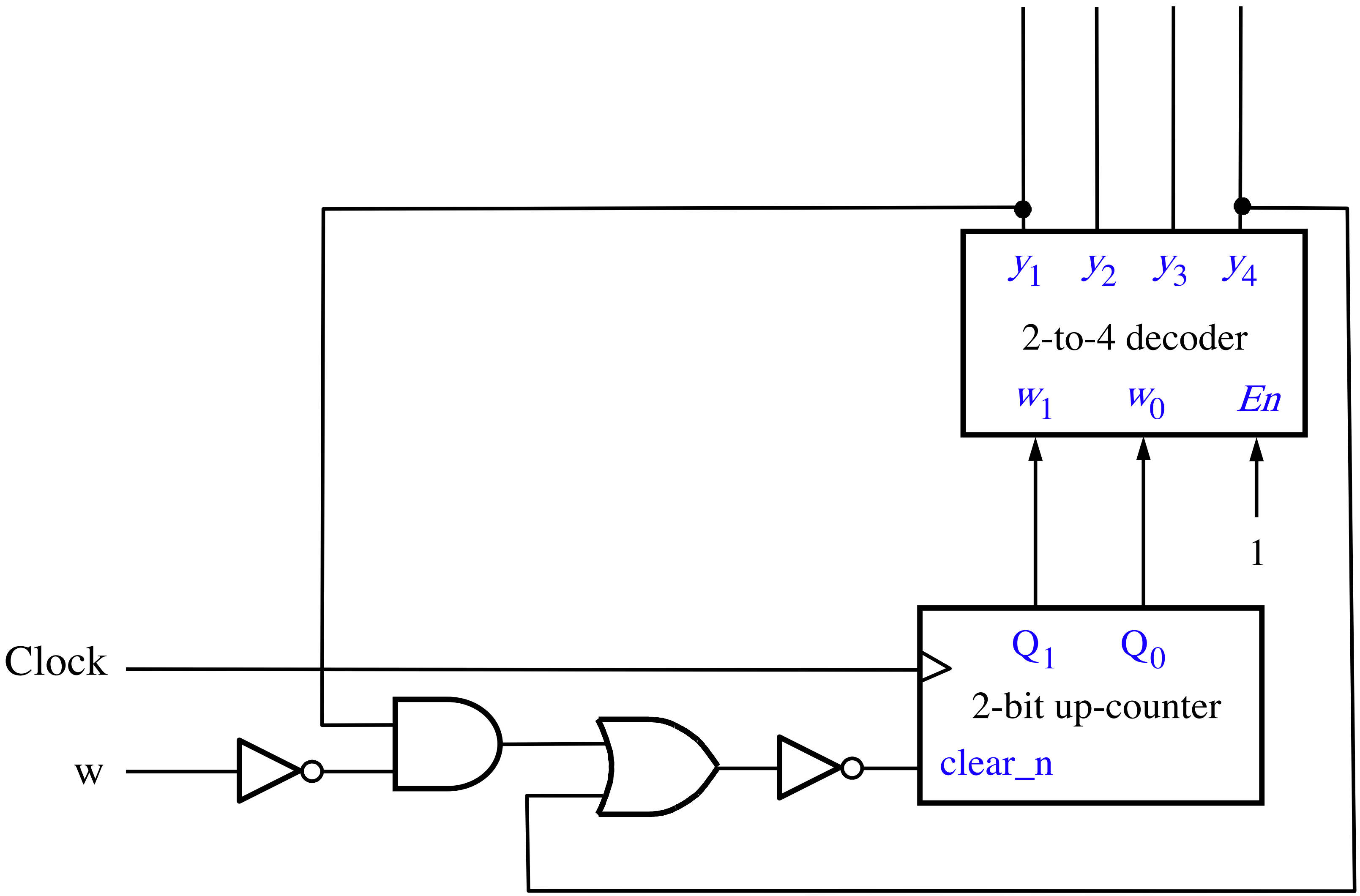


This counter can be cleared only on the positive clock edge.

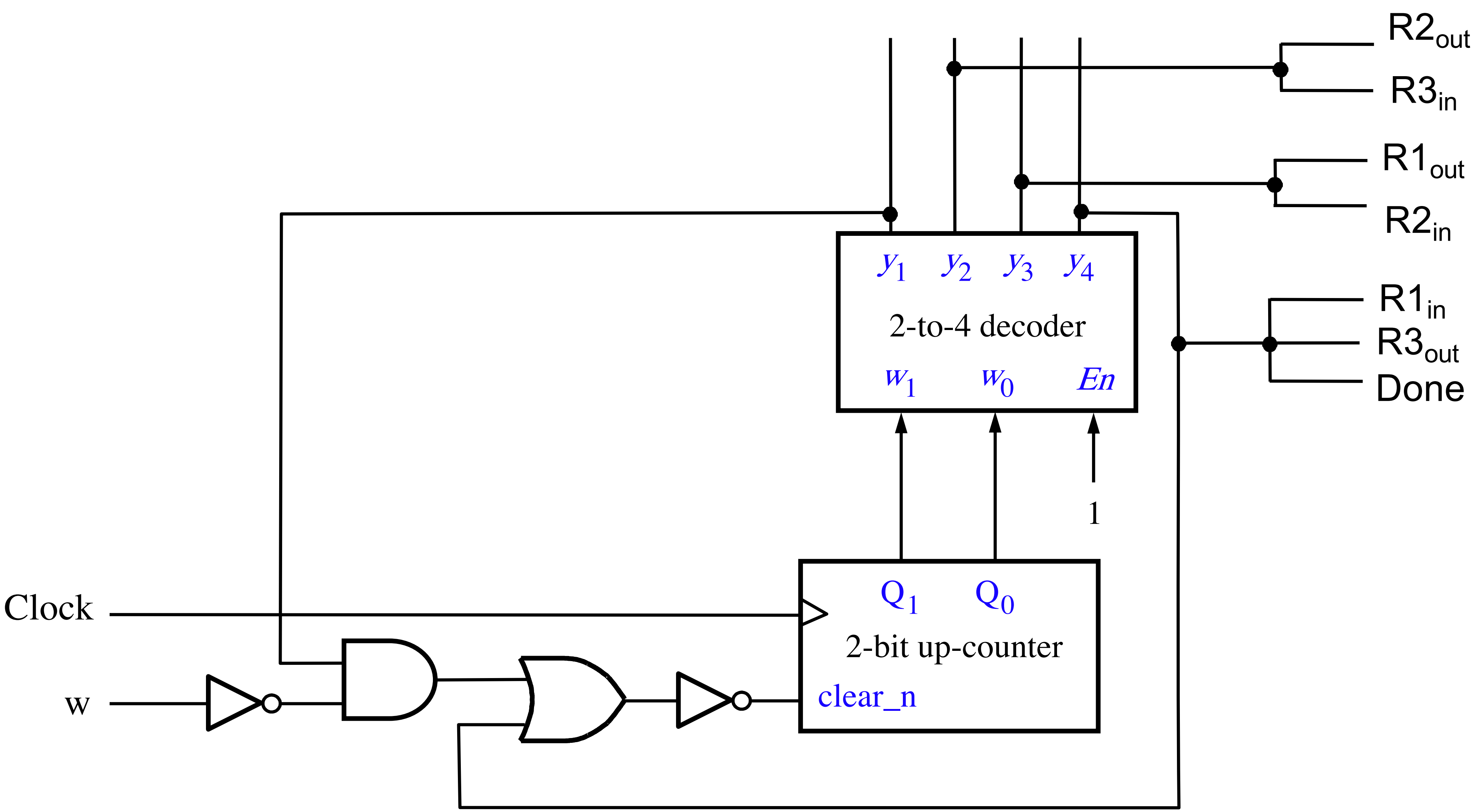
Let's Complete the Circuit Diagram



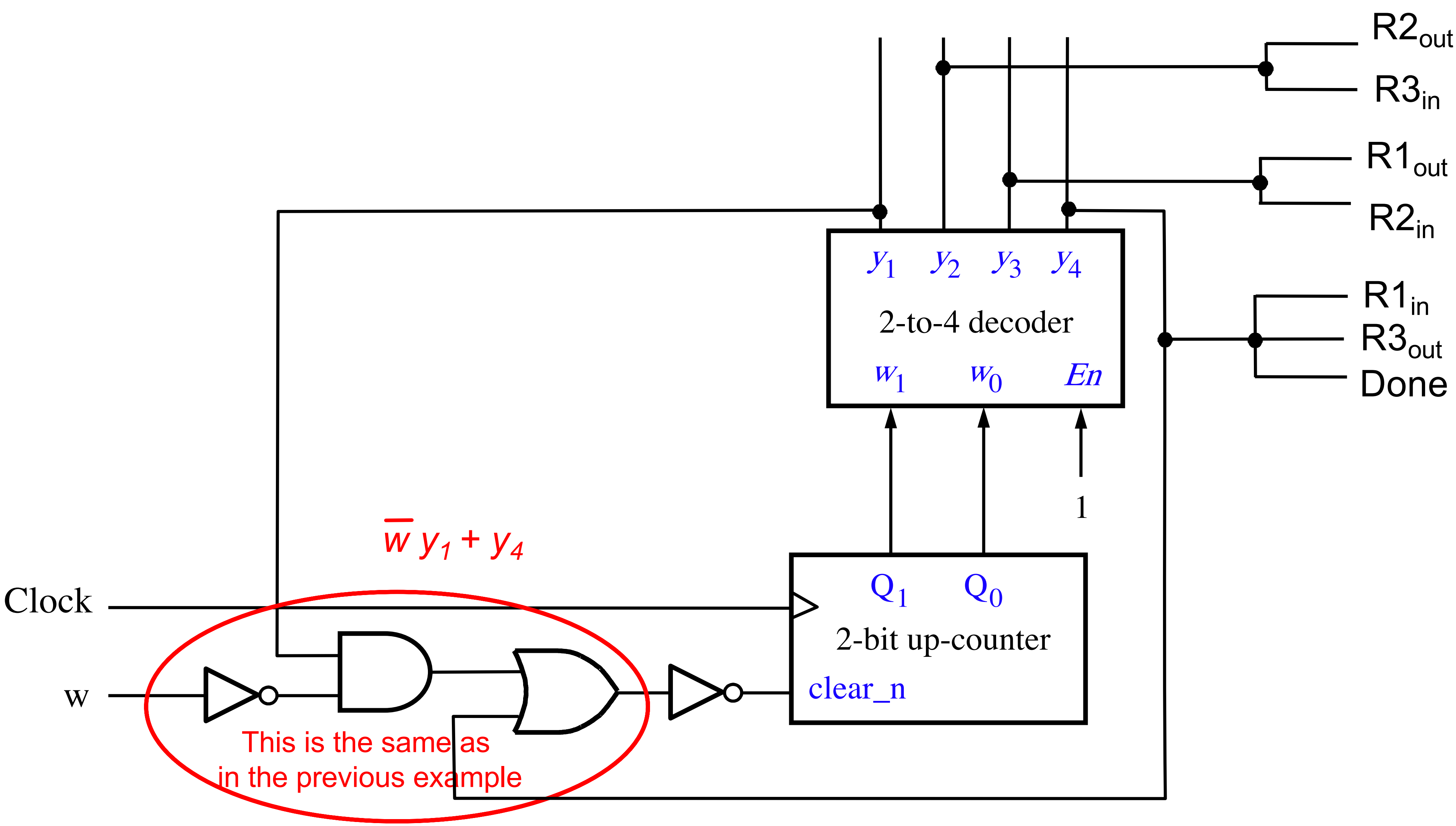
Let's Complete the Circuit Diagram



Let's Complete the Circuit Diagram



Let's Complete the Circuit Diagram



The Solution for Encoding #3

$R1_{out} = R2_{in} = y_3$

$R1_{in} = R3_{out} = Done = y_4$

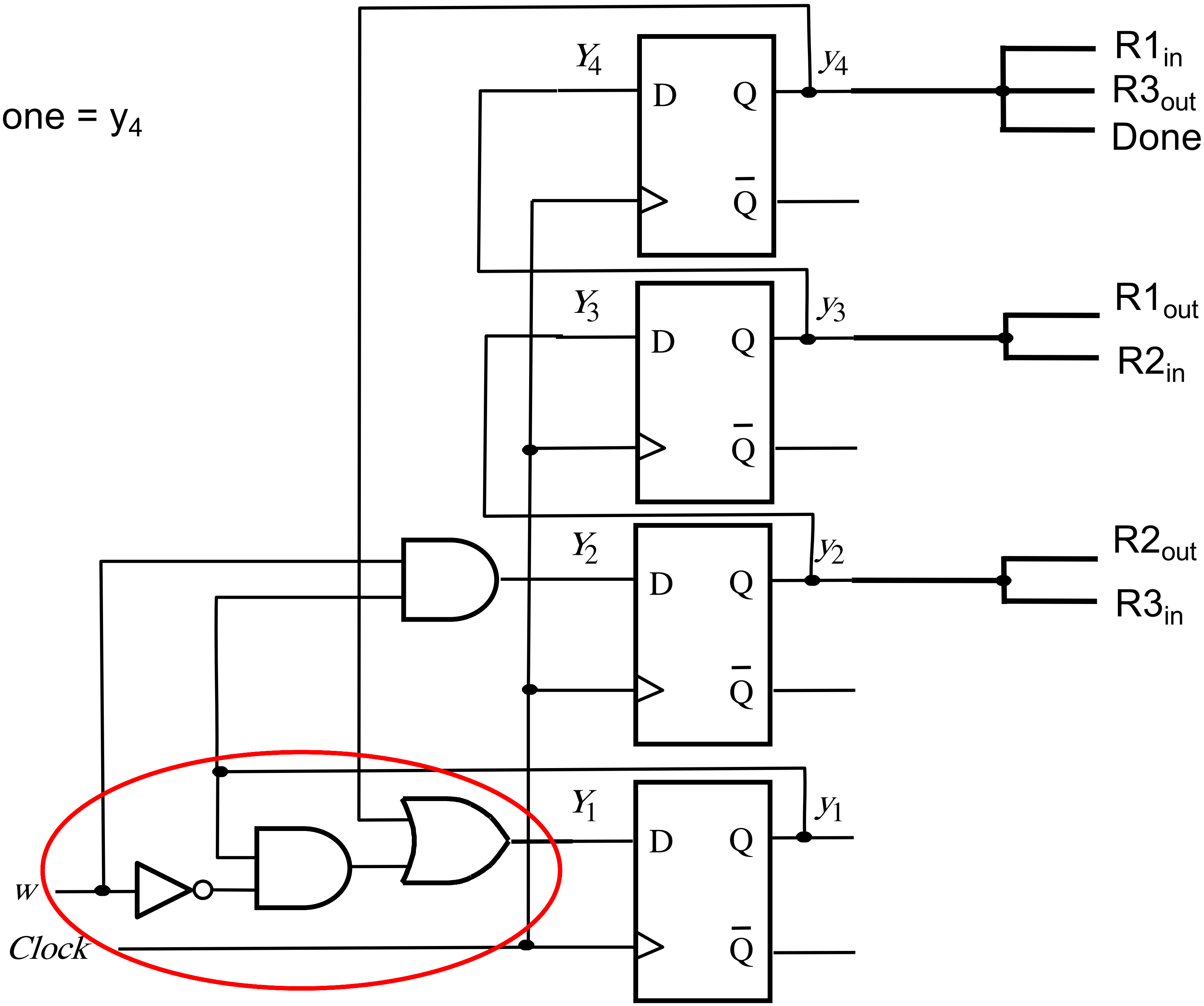
$R2_{out} = R3_{in} = y_2$

$Y_1 = \overline{w}\, y_1 + y_4$

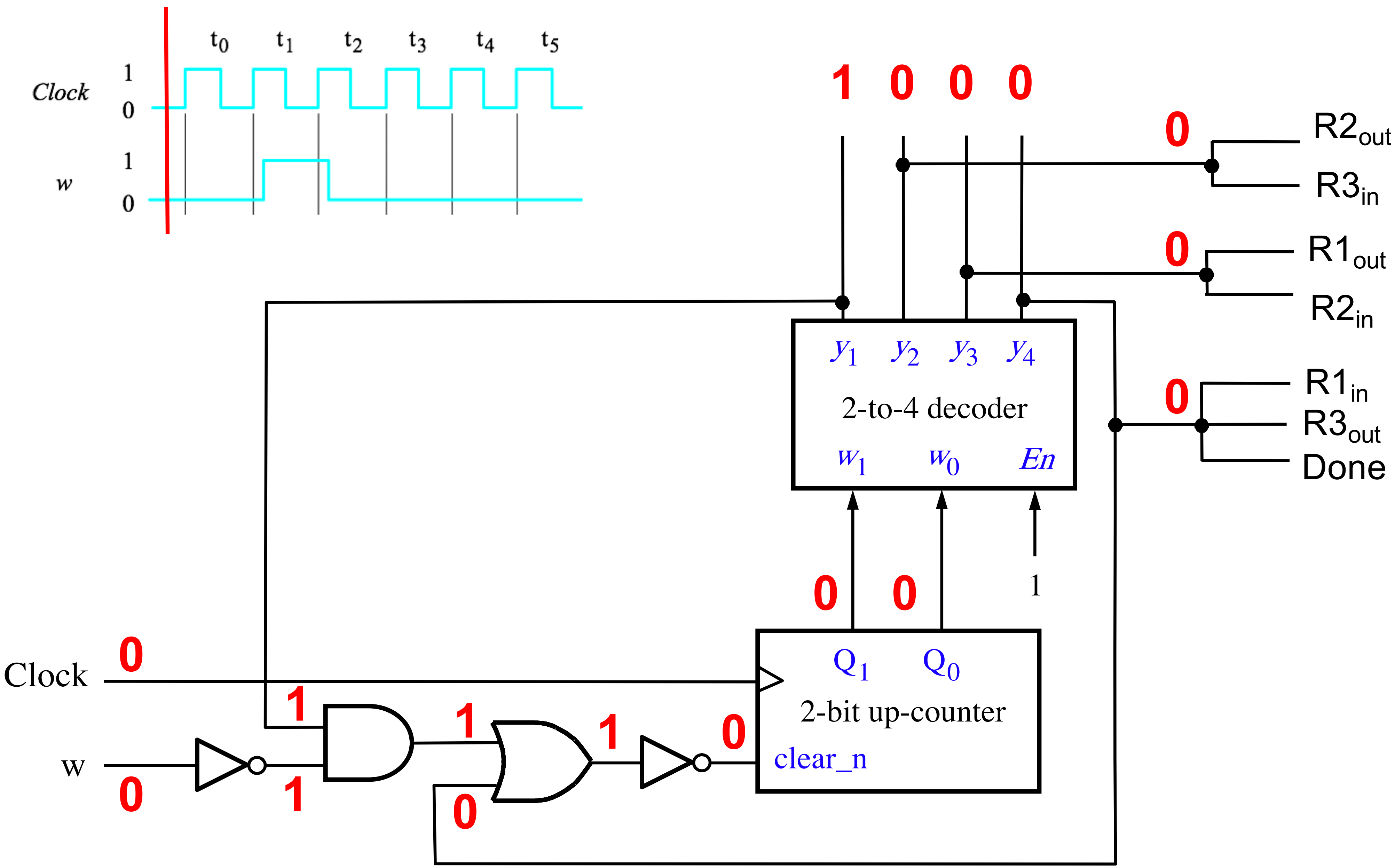
$Y_2 = w\, y_1$

$Y_3 = y_2$

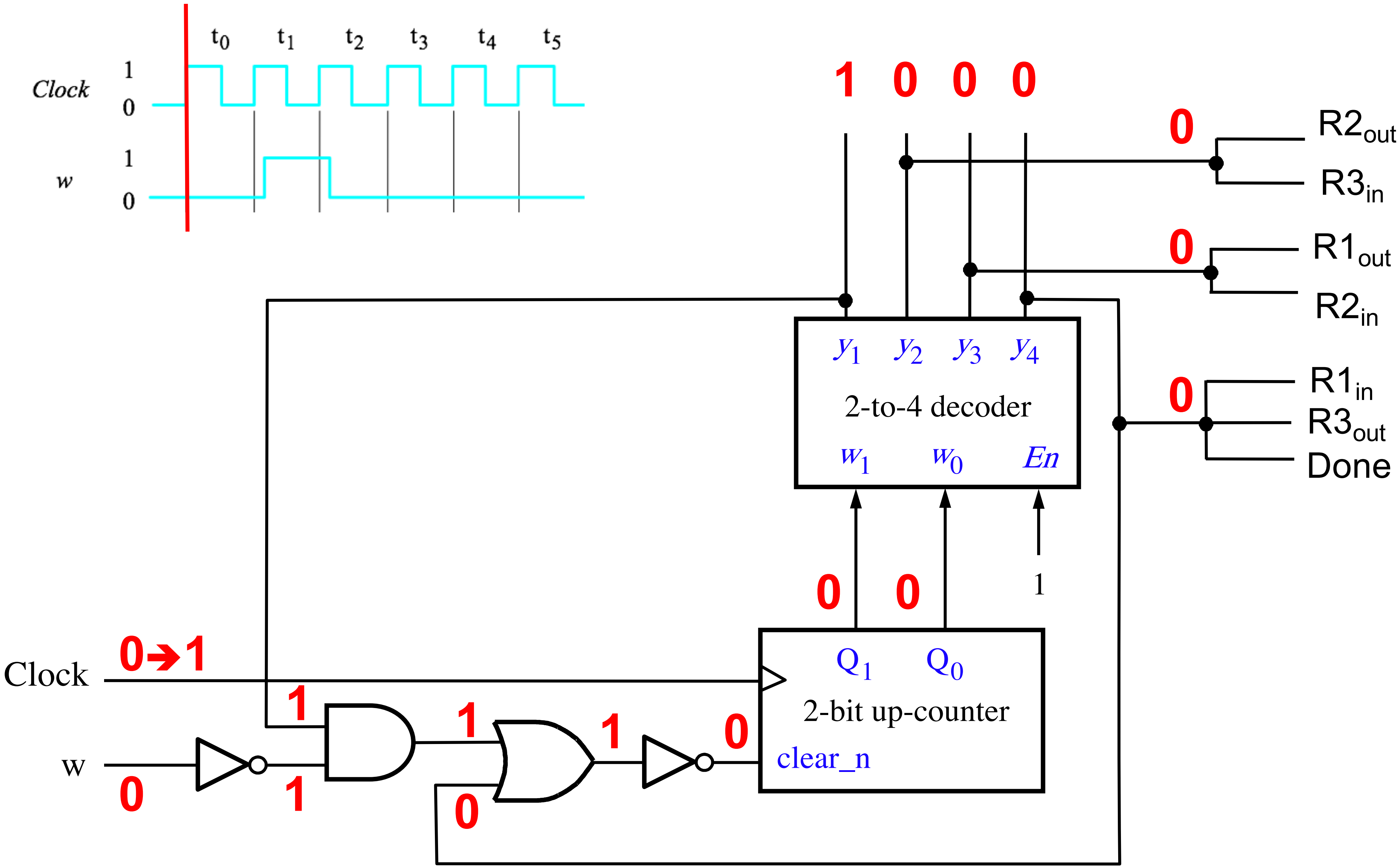
$Y_4 = y_3$



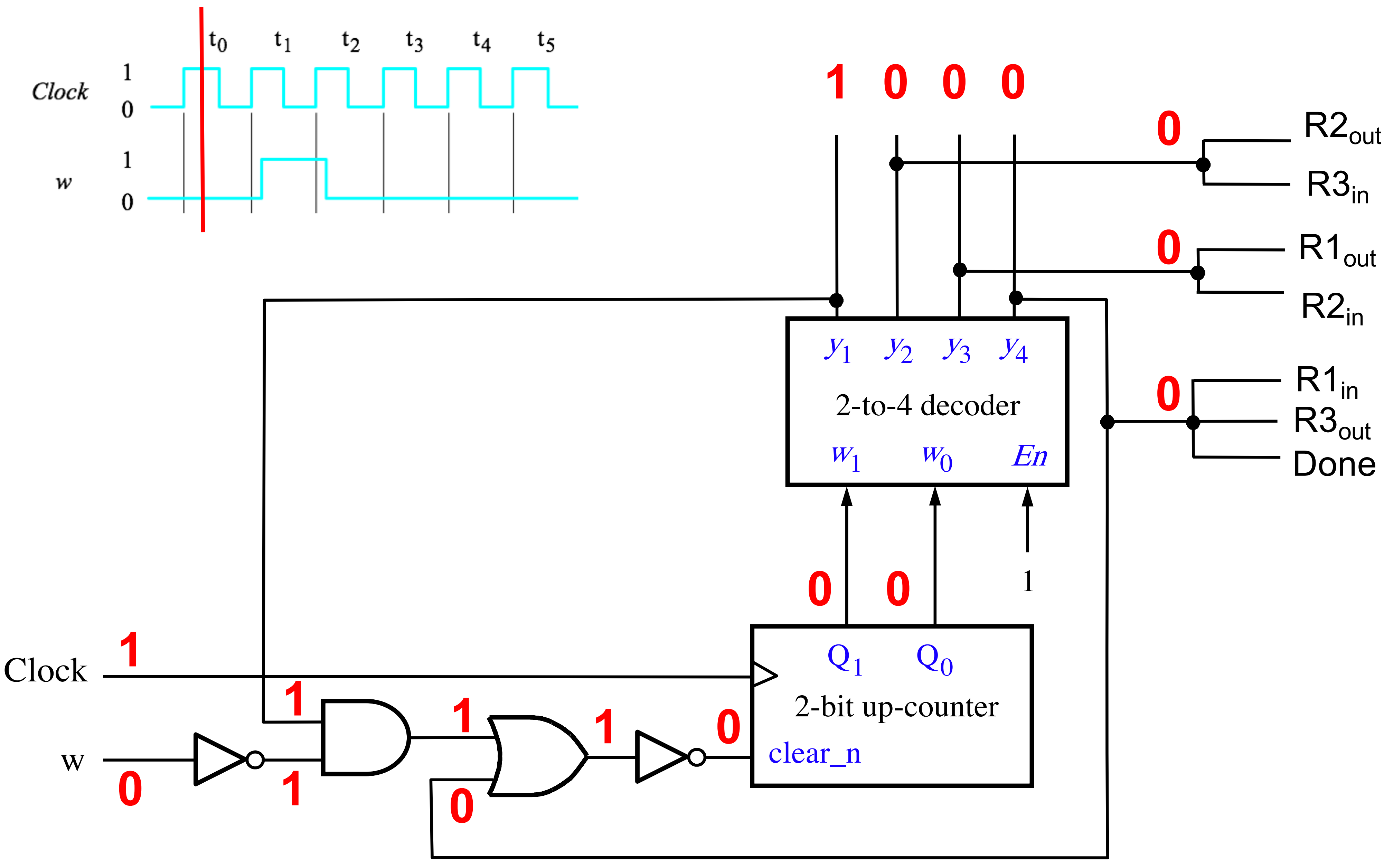
How Does It Work?



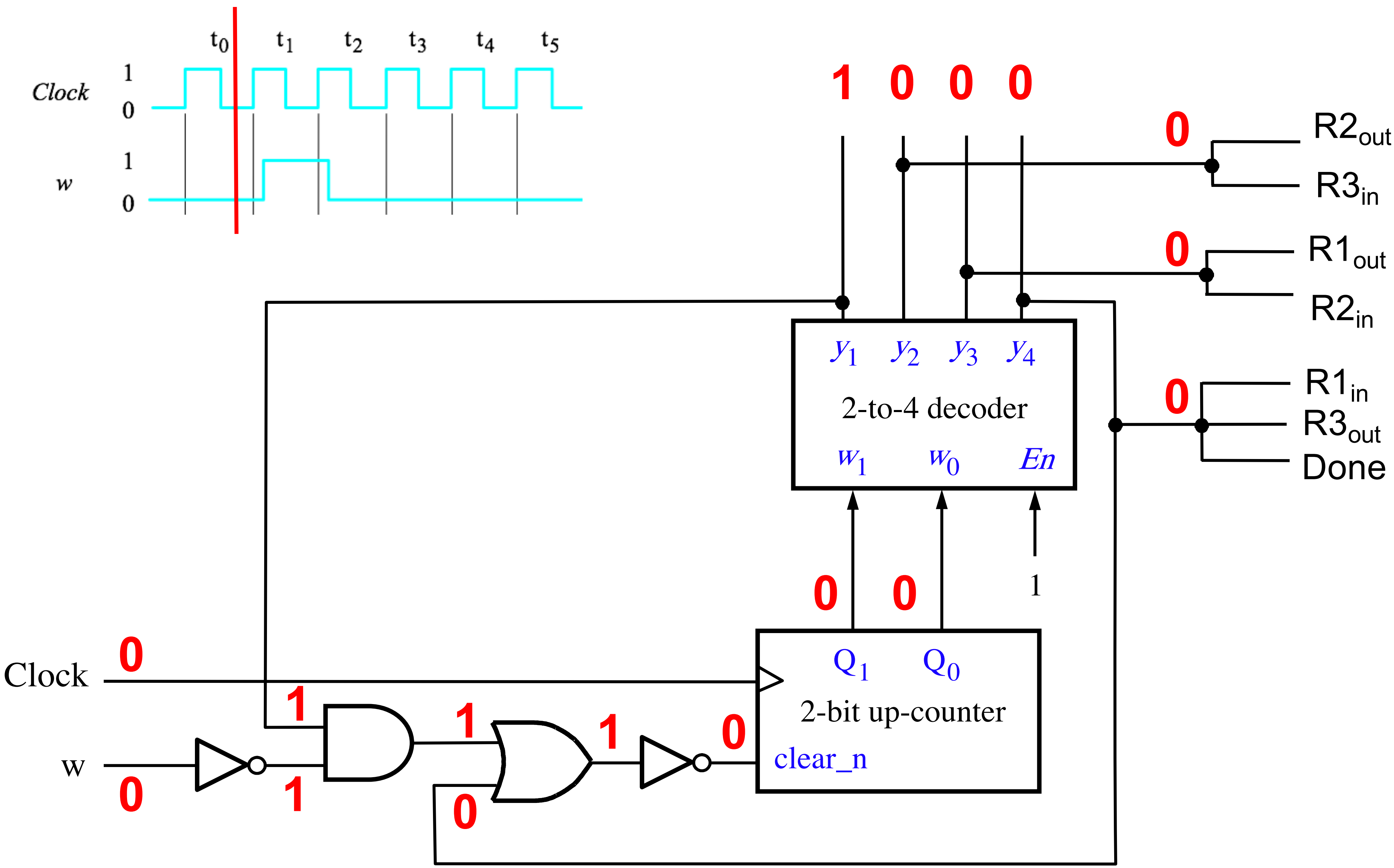
How Does It Work?



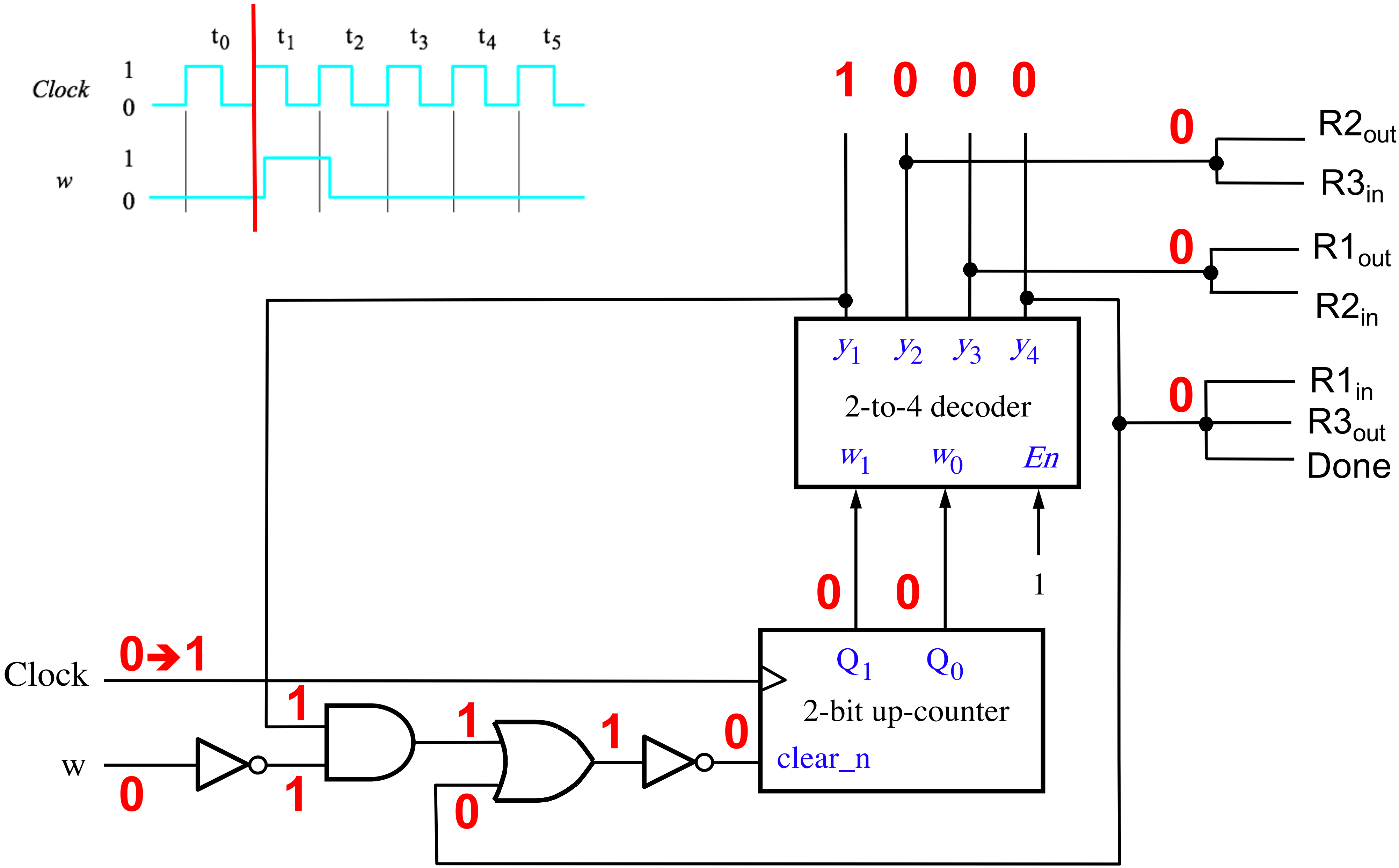
How Does It Work?



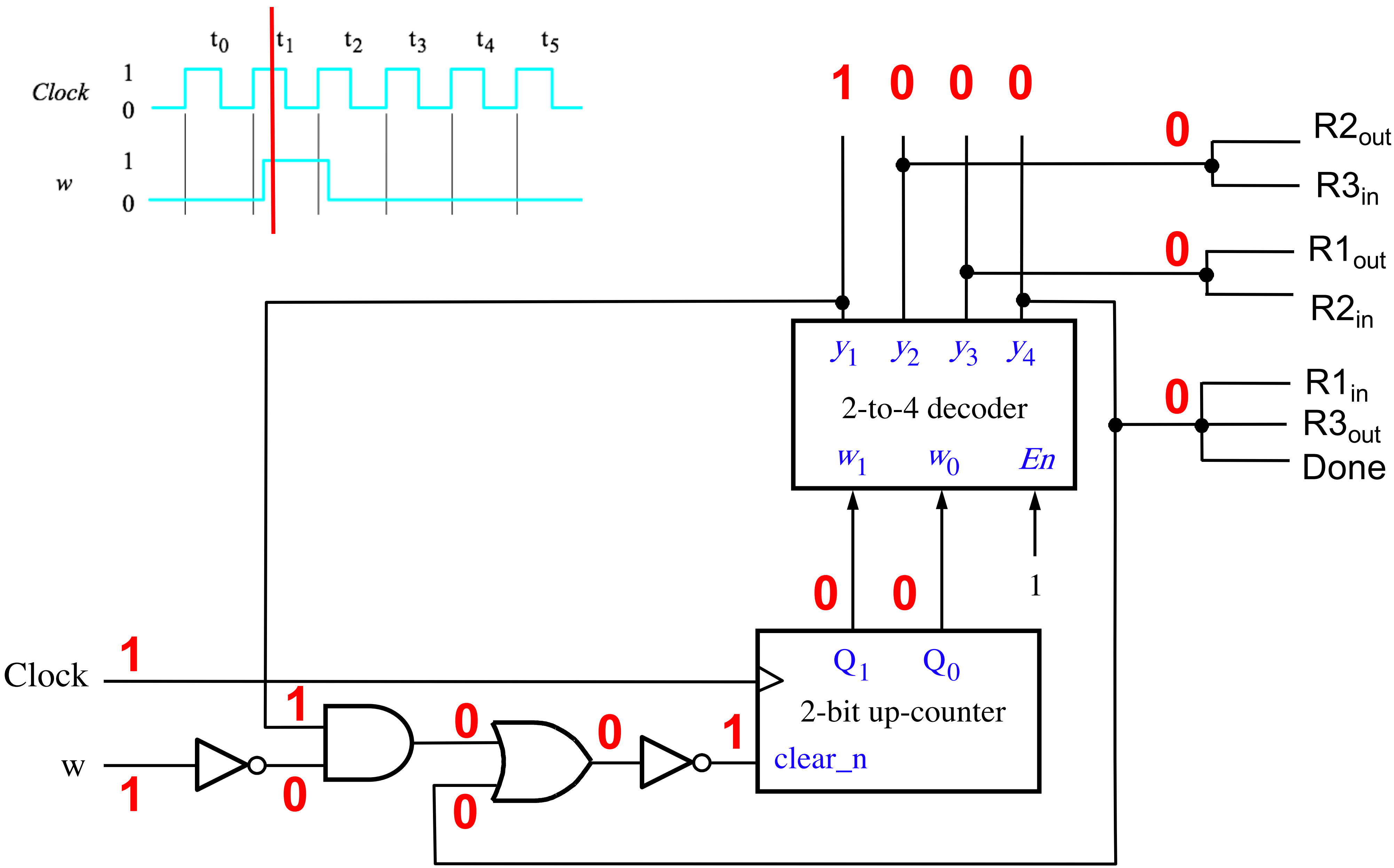
How Does It Work?



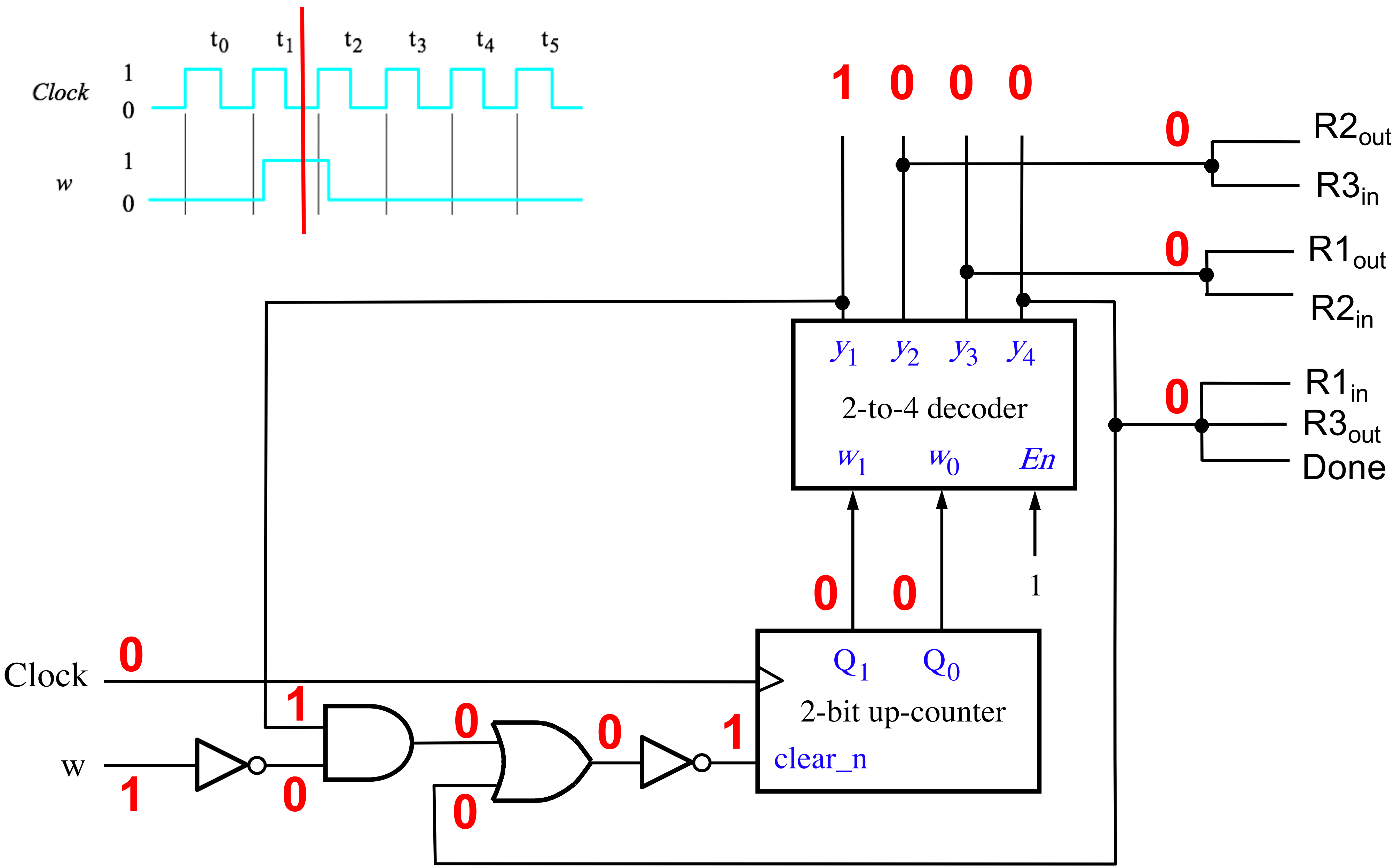
How Does It Work?



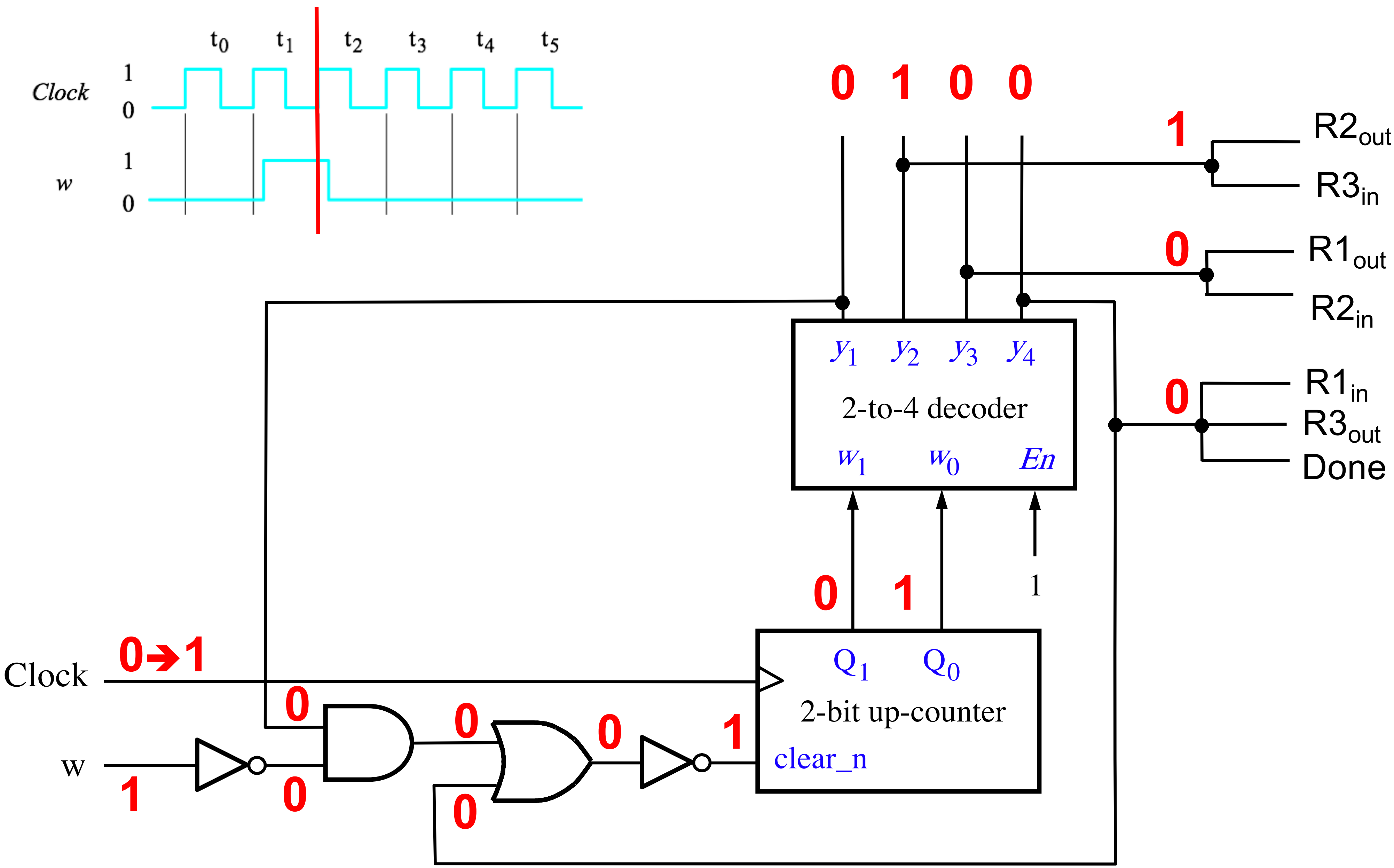
How Does It Work?



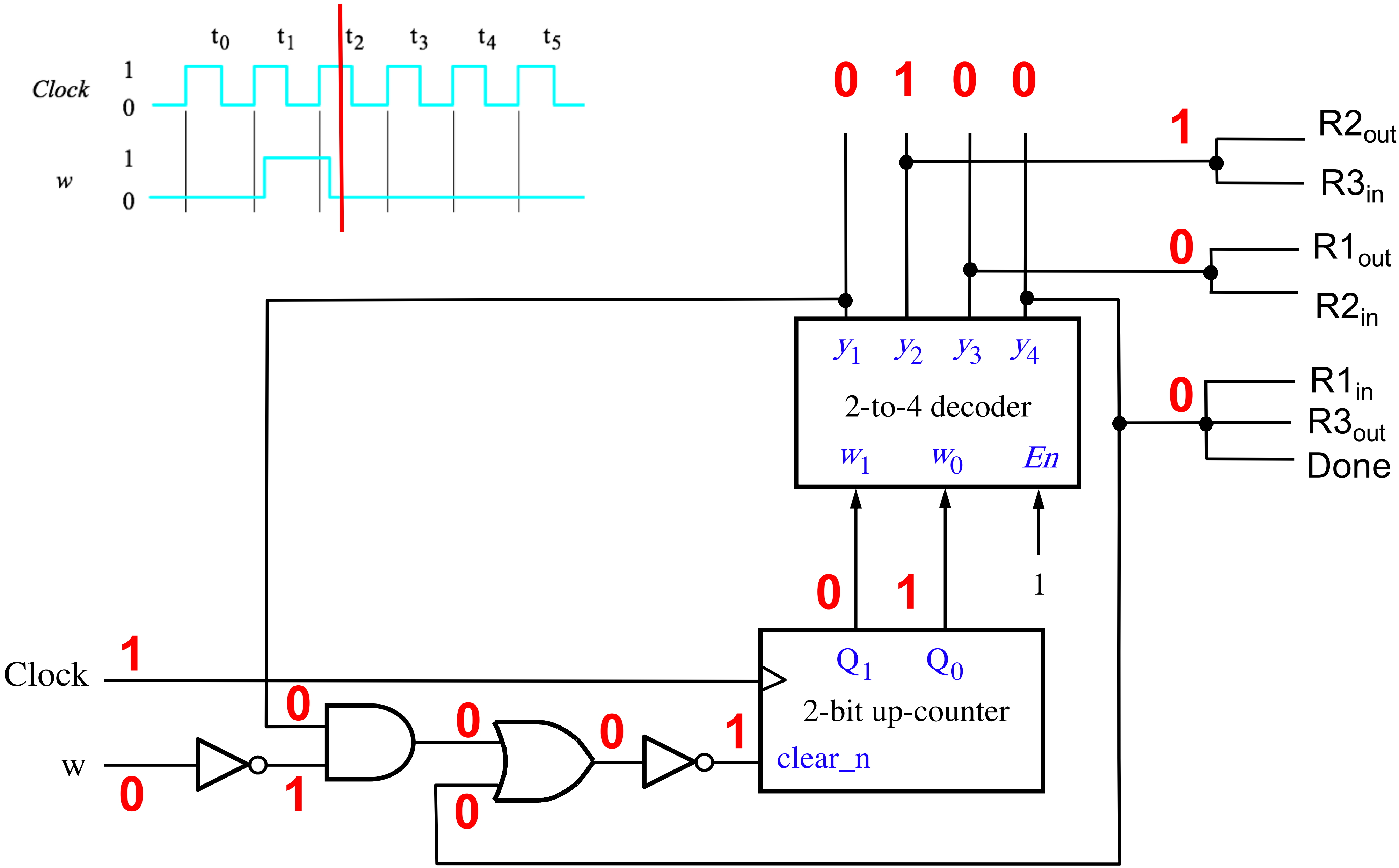
How Does It Work?



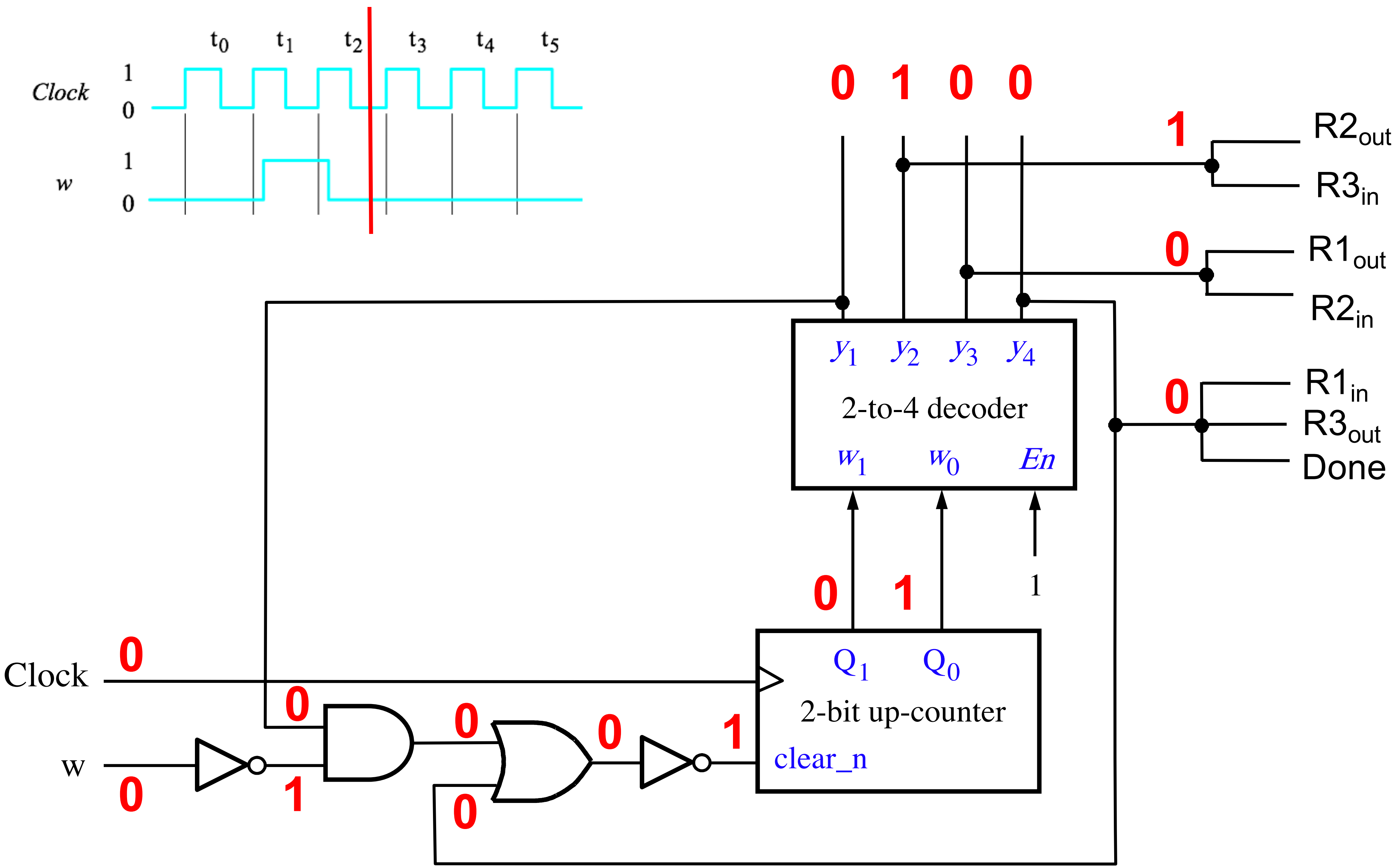
How Does It Work?



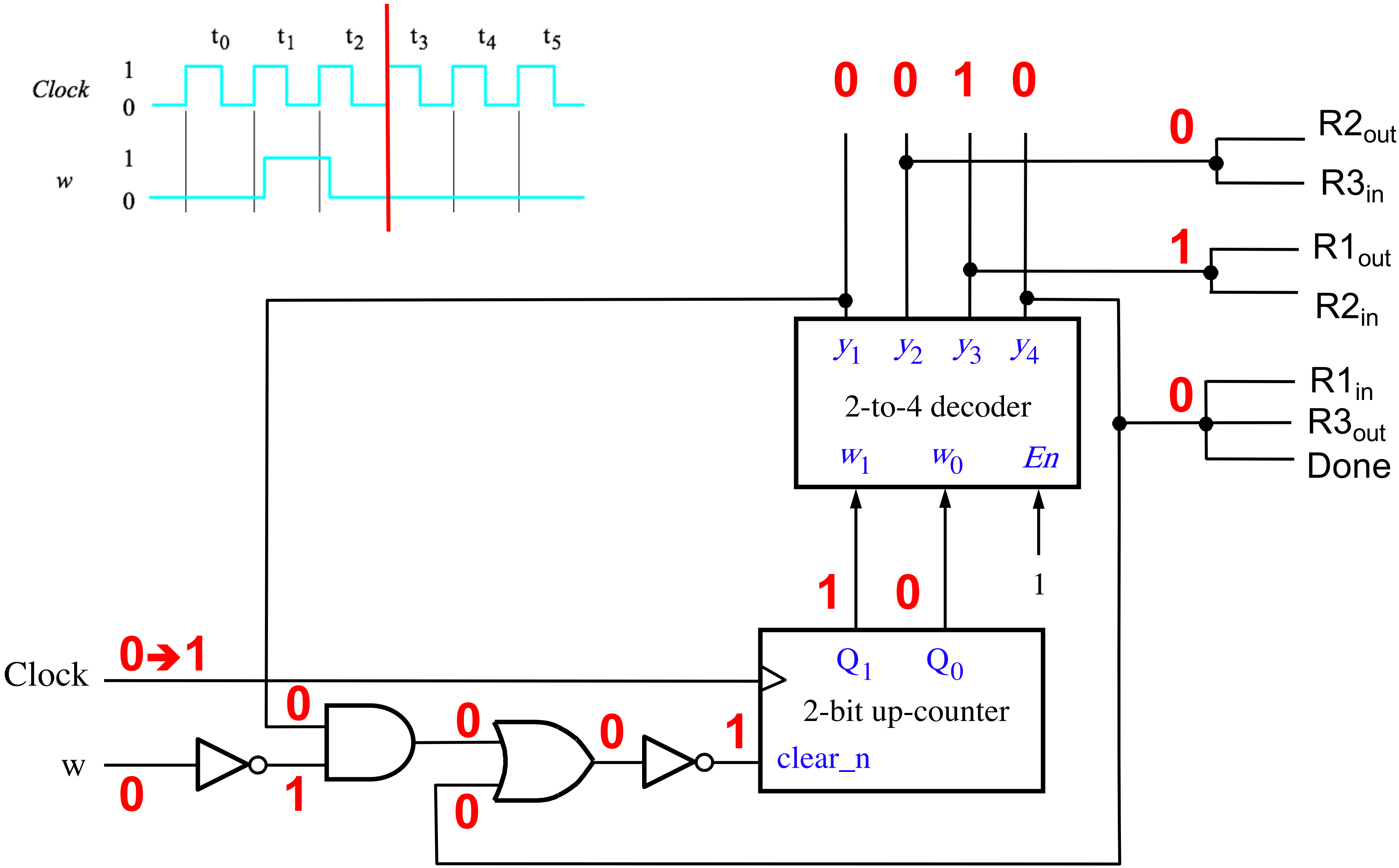
How Does It Work?



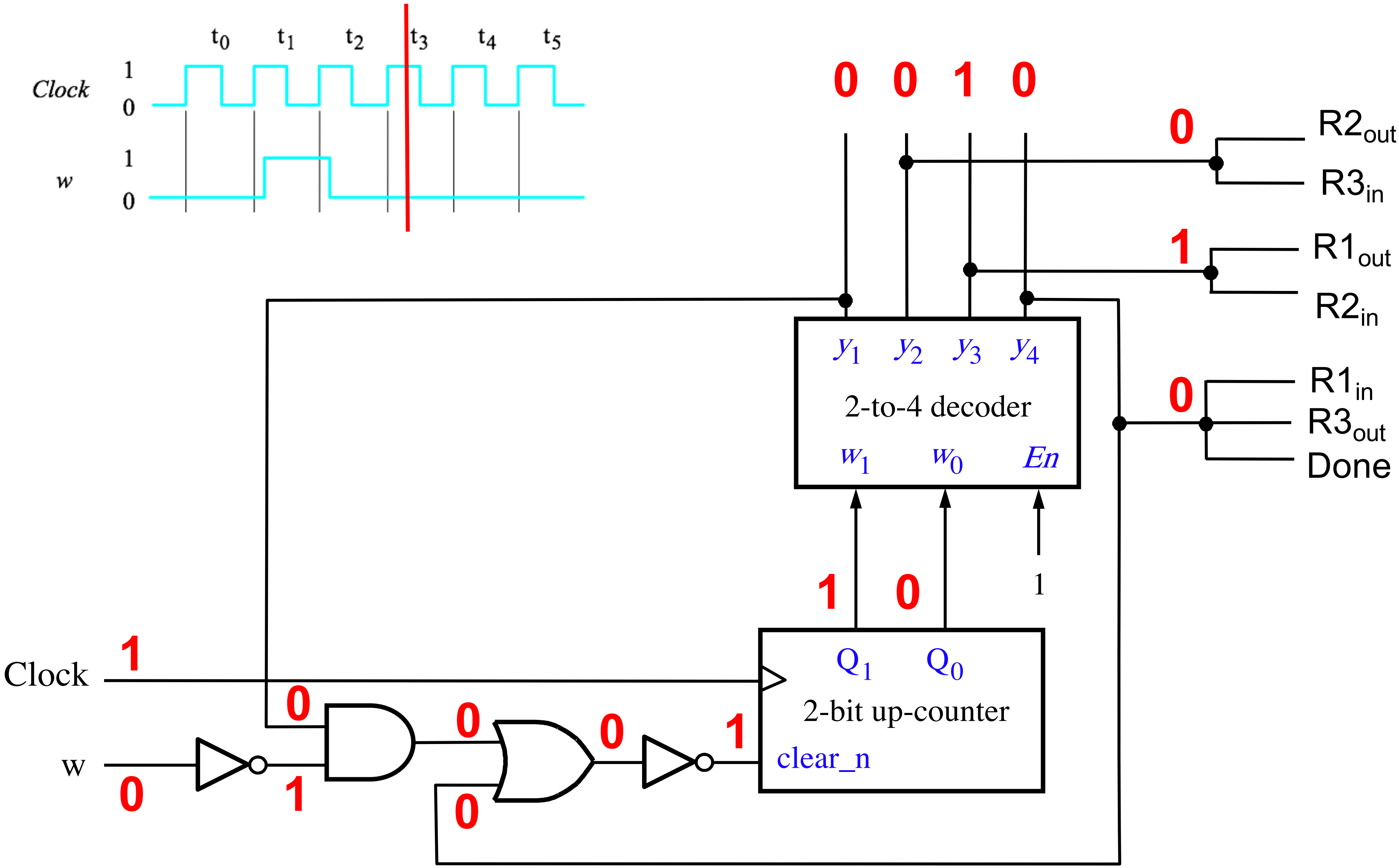
How Does It Work?



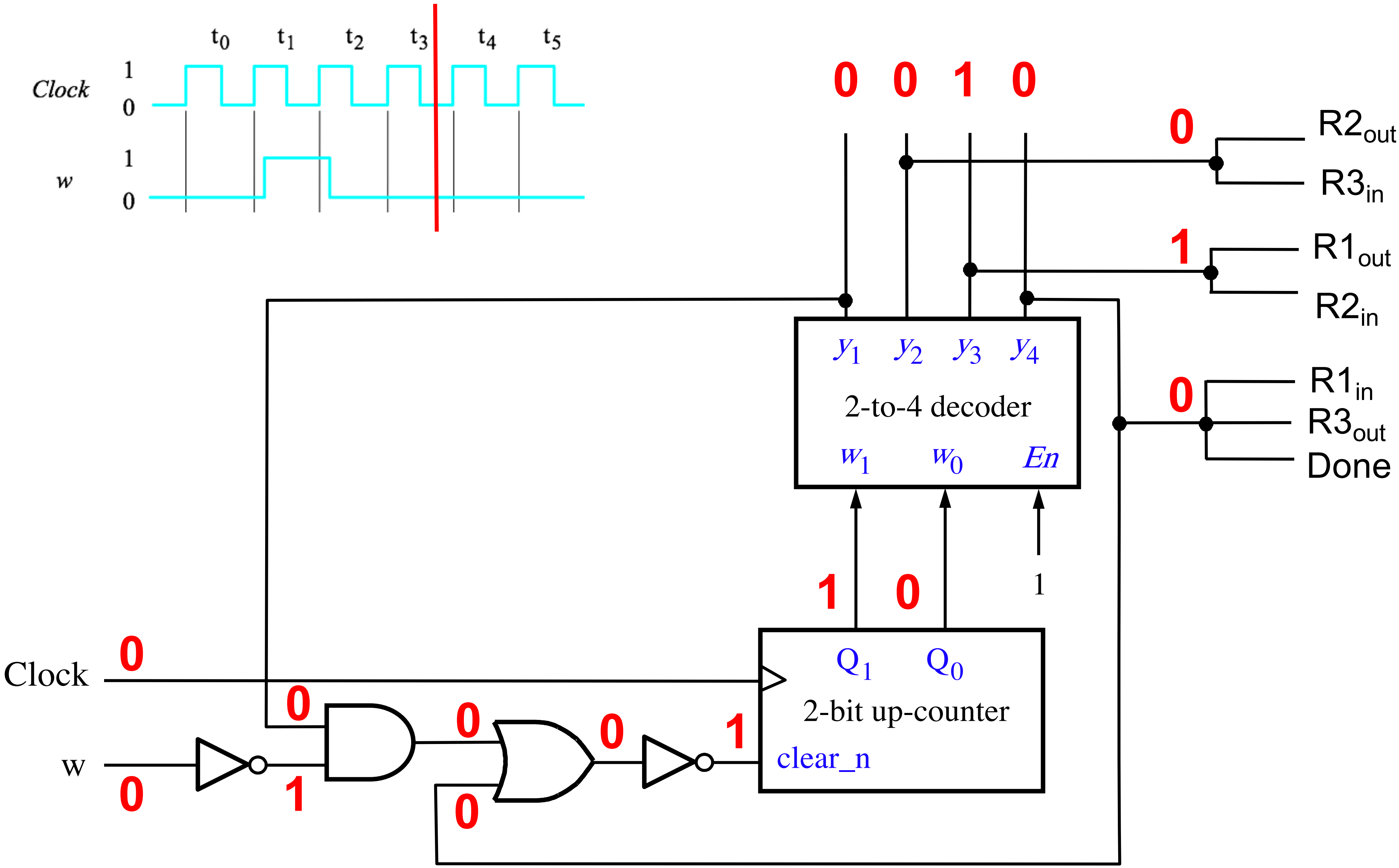
How Does It Work?



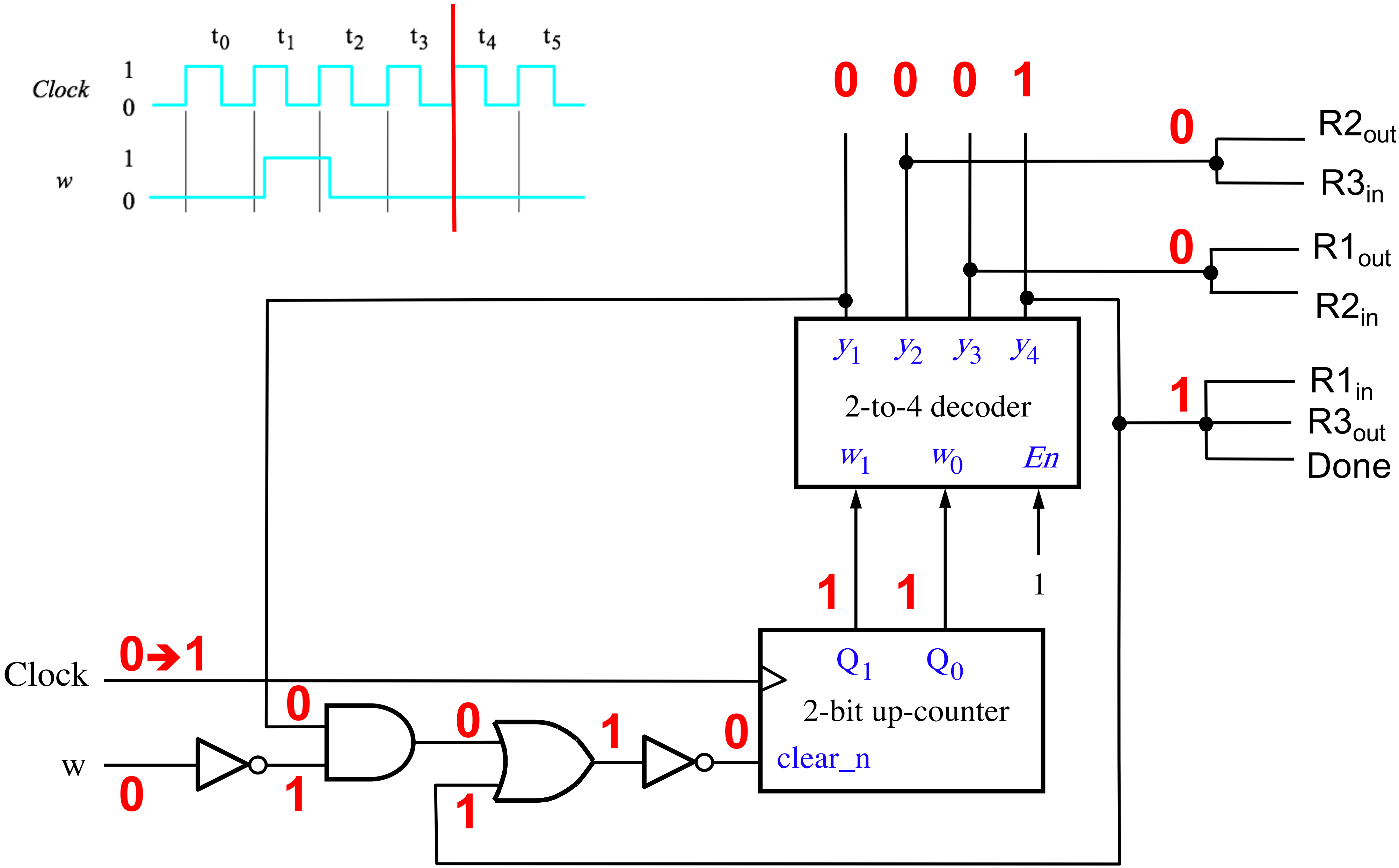
How Does It Work?



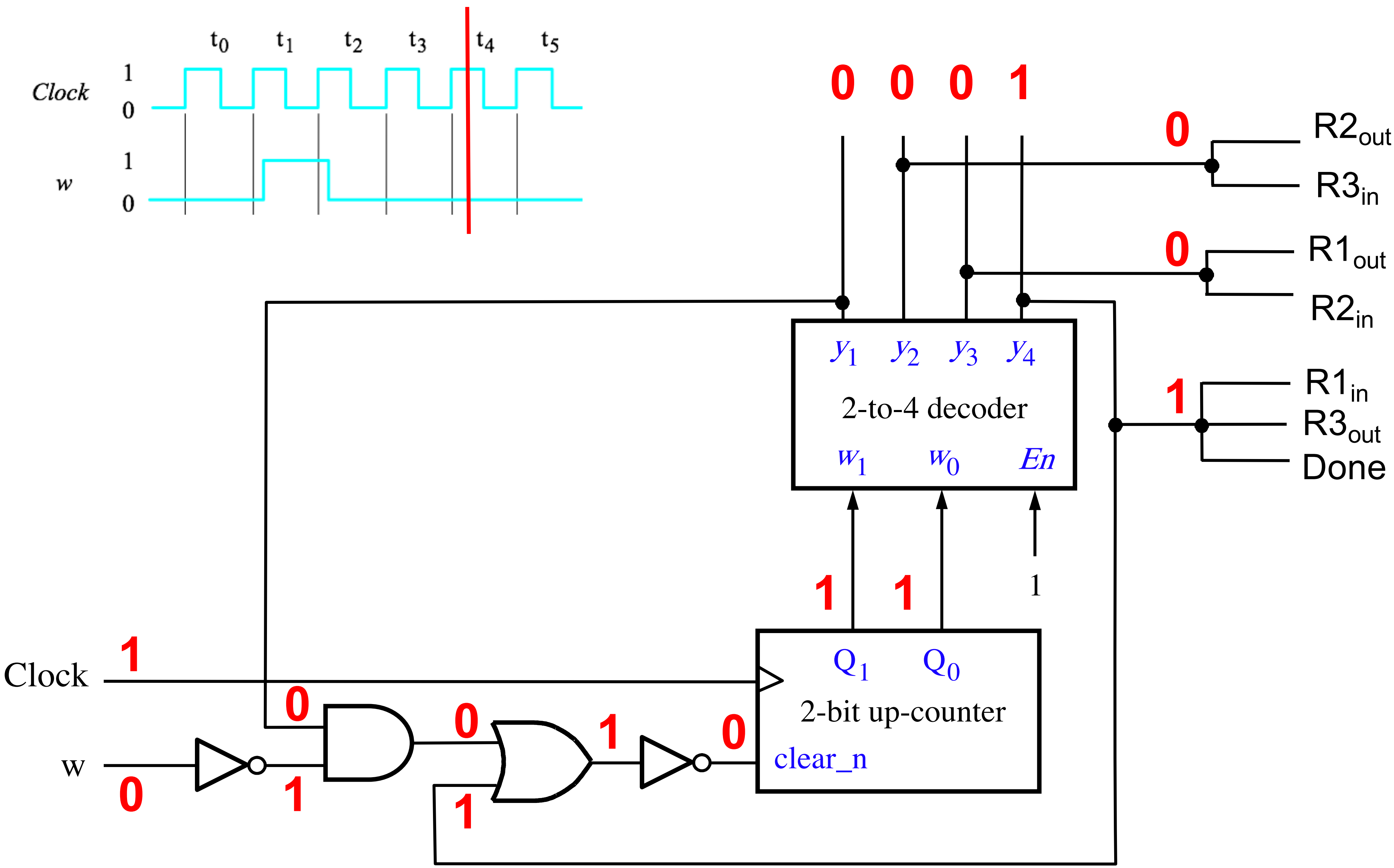
How Does It Work?



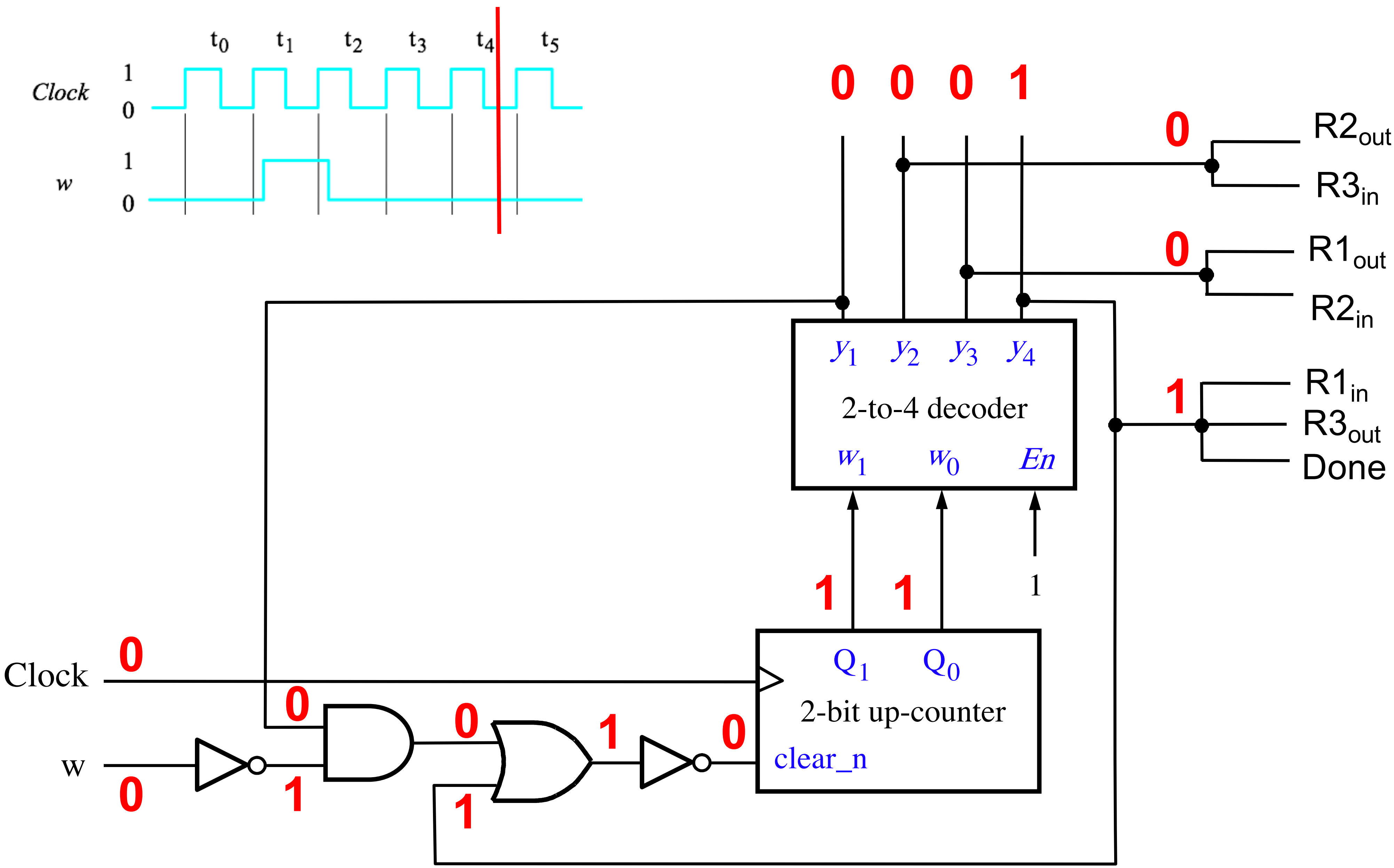
How Does It Work?



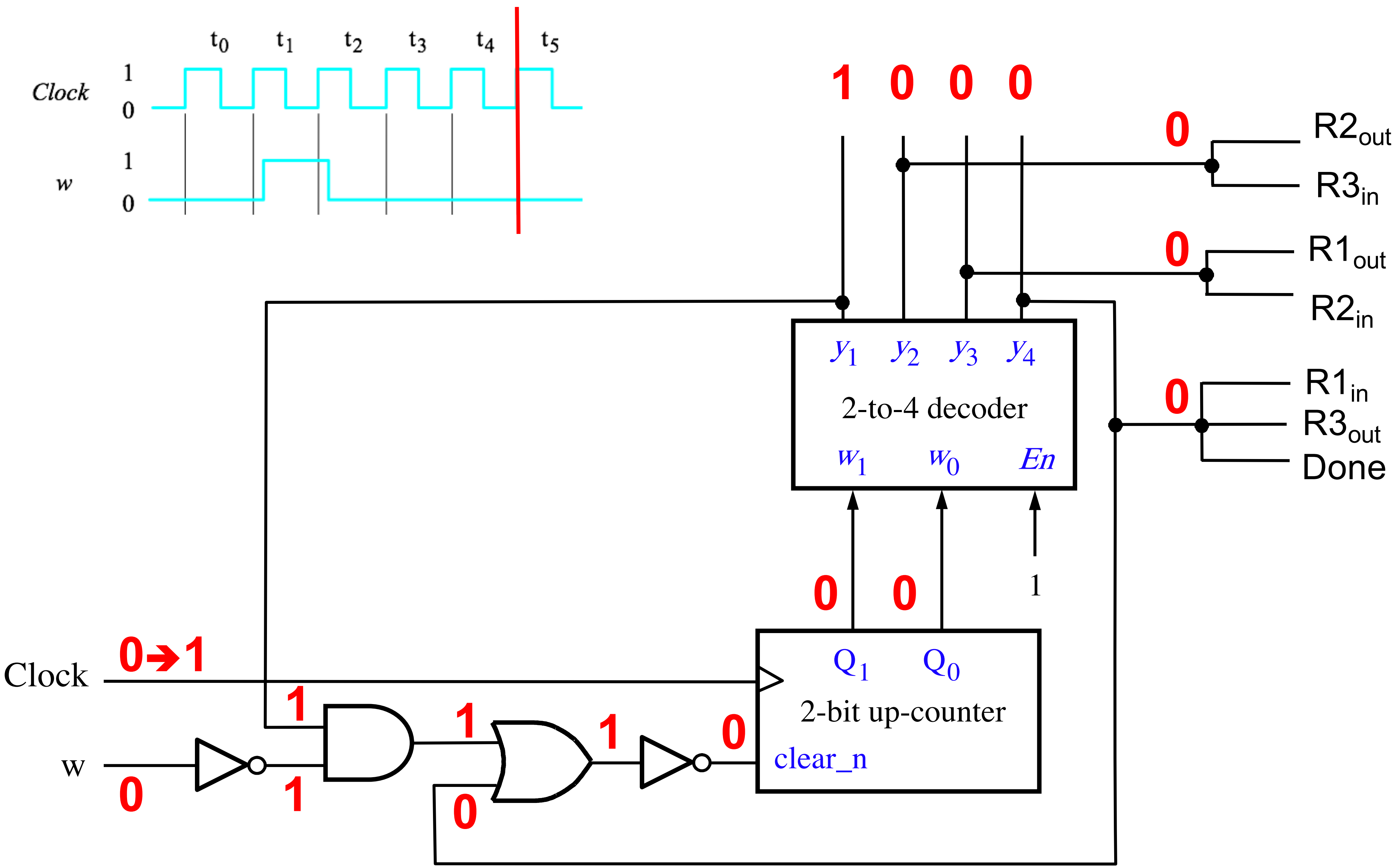
How Does It Work?



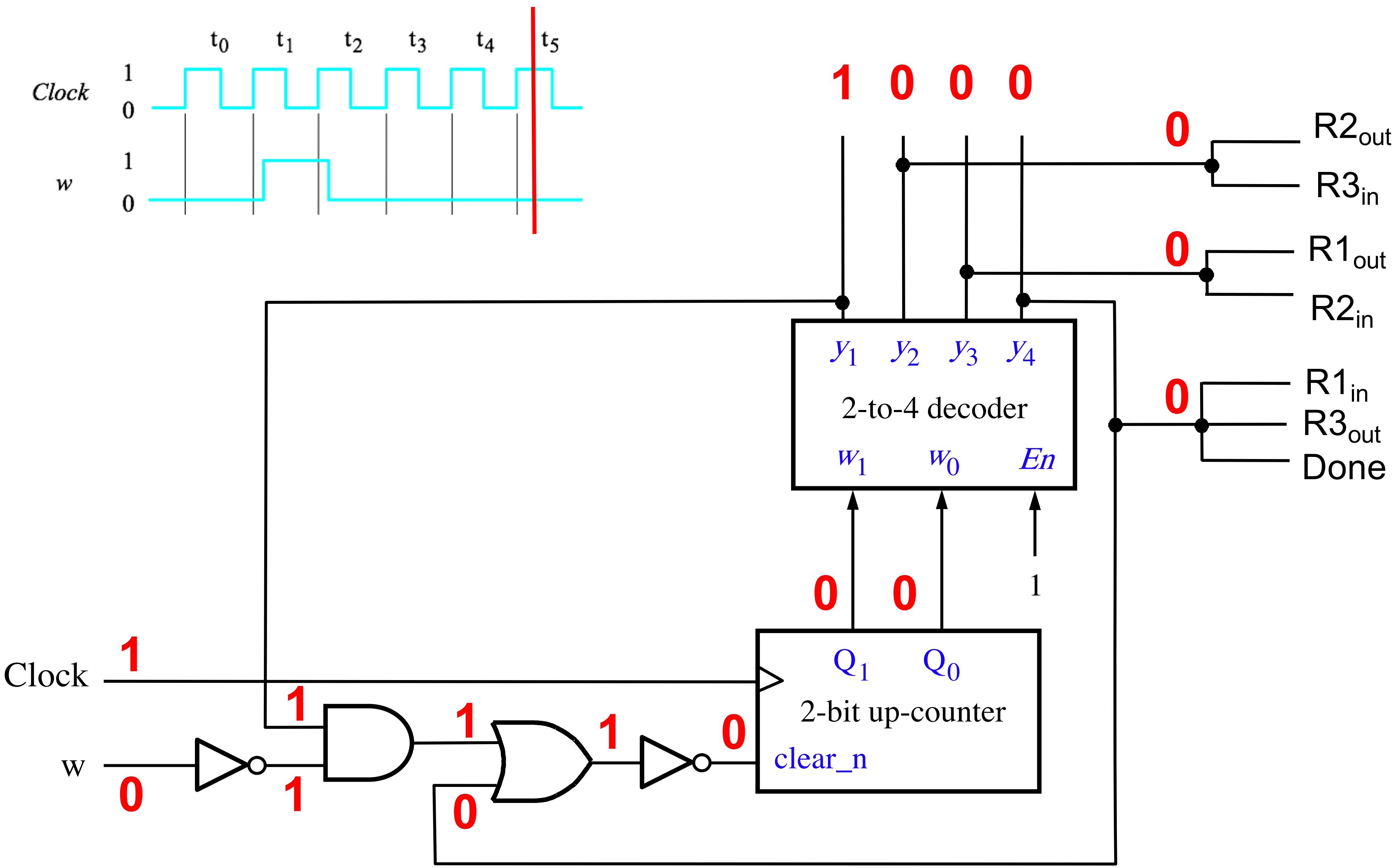
How Does It Work?



How Does It Work?



How Does It Work?



Questions?

THE END