



CprE 281: Digital Logic

Instructor: Alexander Stoytchev

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

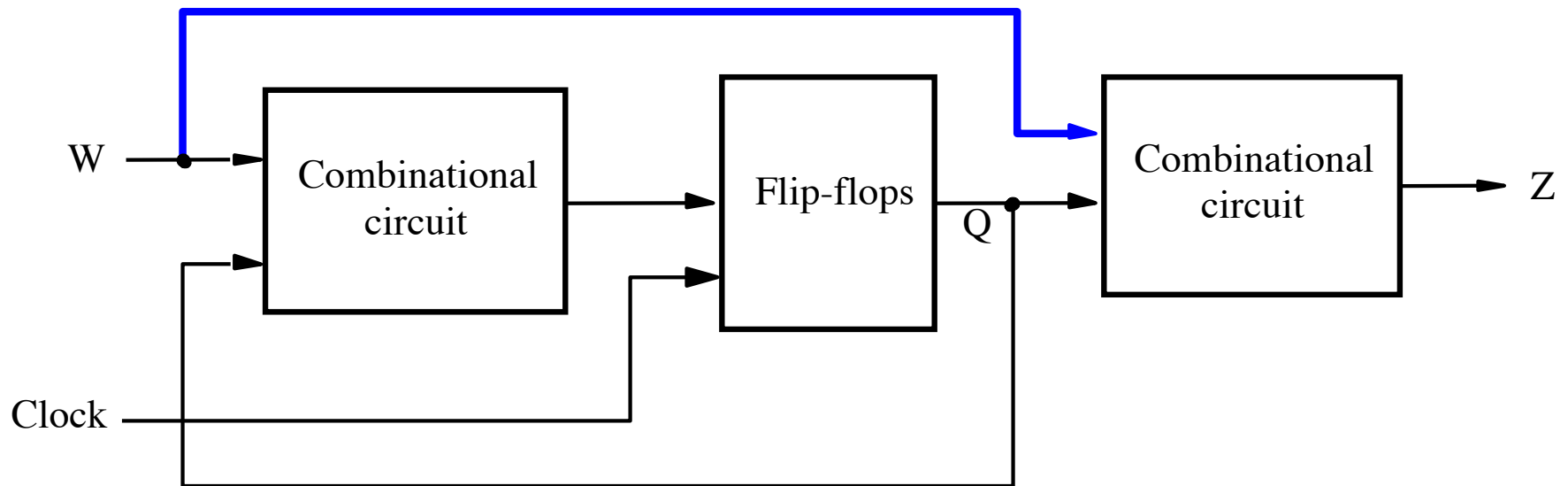
Mealy State Model

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

Administrative Stuff

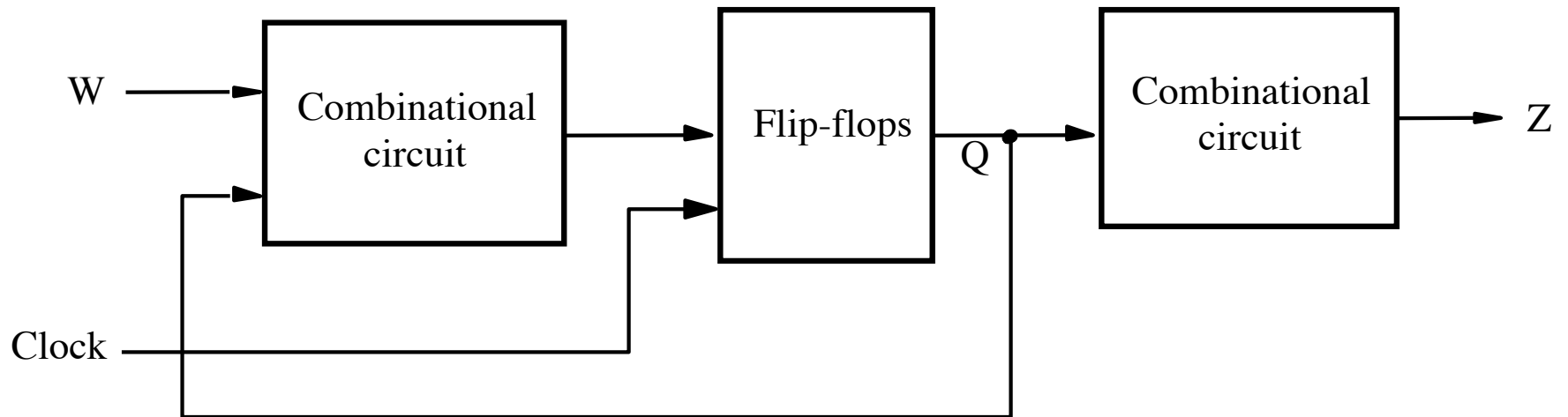
- **Homework 10 is out**
- **It is due on Monday Nov 2 @ 4pm**

The general form of a synchronous sequential circuit

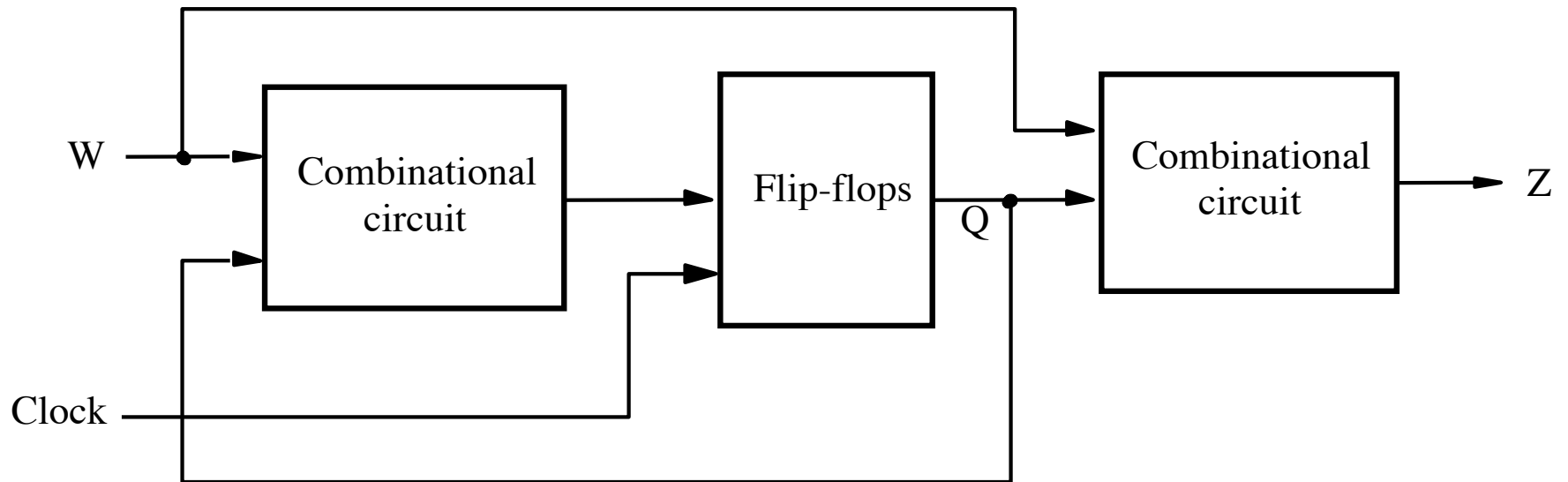


[Figure 6.1 from the textbook]

Moore Type



Mealy Type

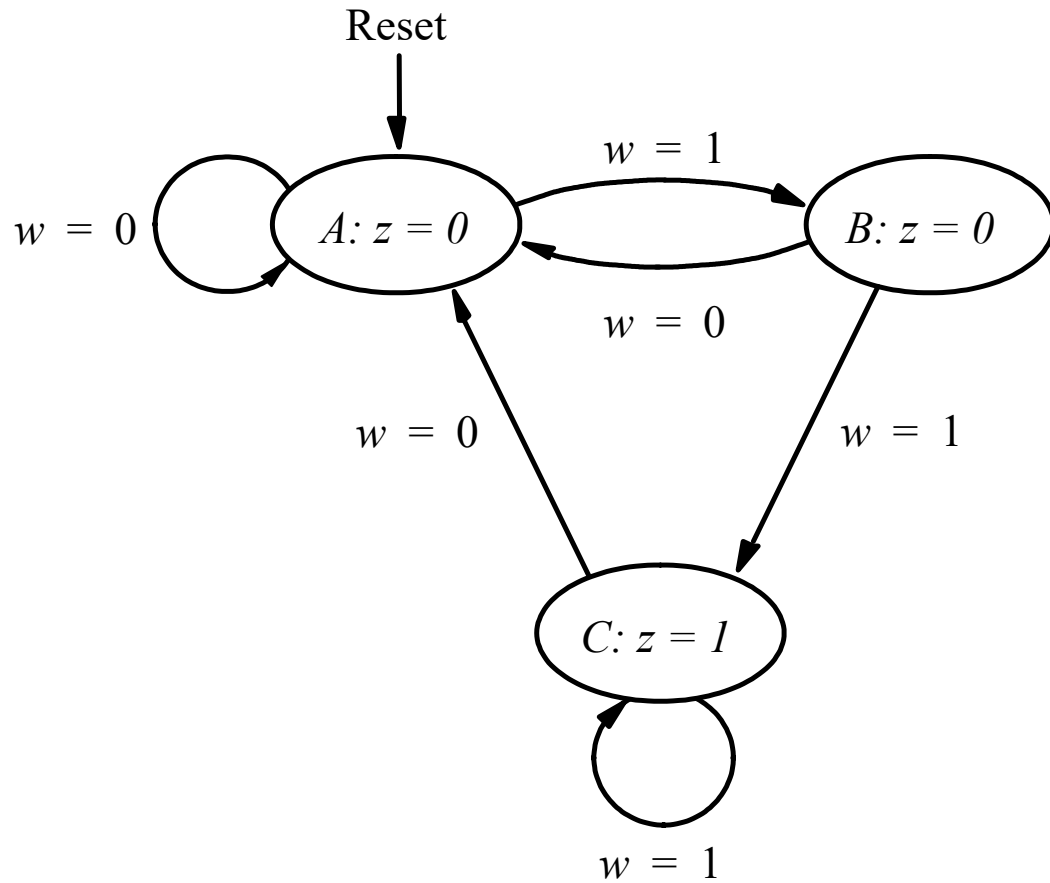


Sample Problem

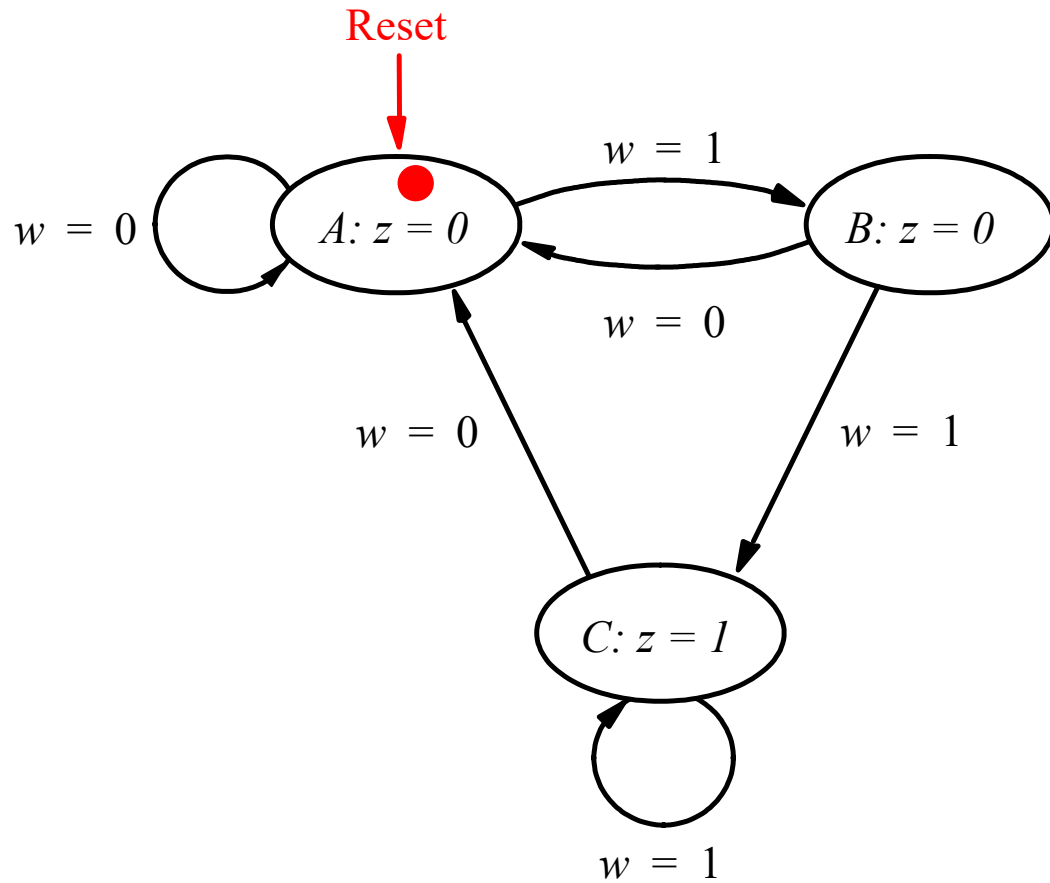
Implement a 11 detector. In other words, the output should be equal to 1 if two consecutive 1's have been detected on the input line.

The output should become 1 as soon as the second 1 is detected in the input.

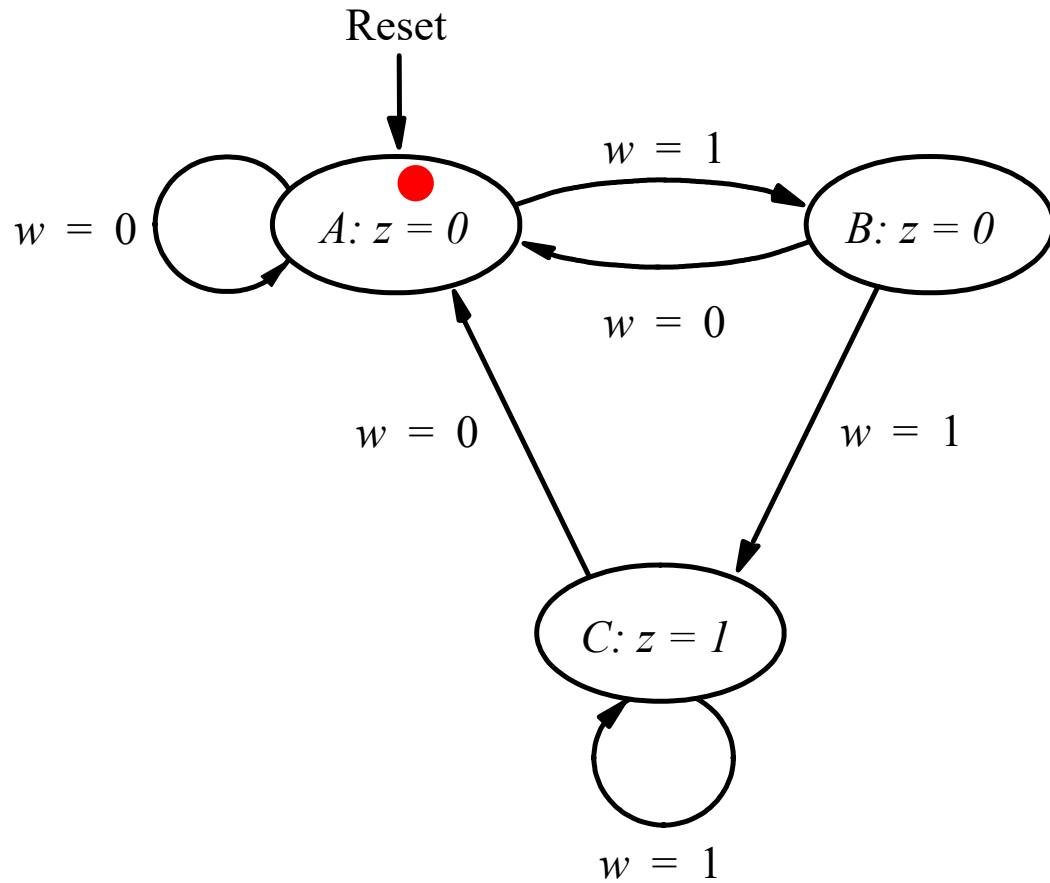
Moore Machine Implementation



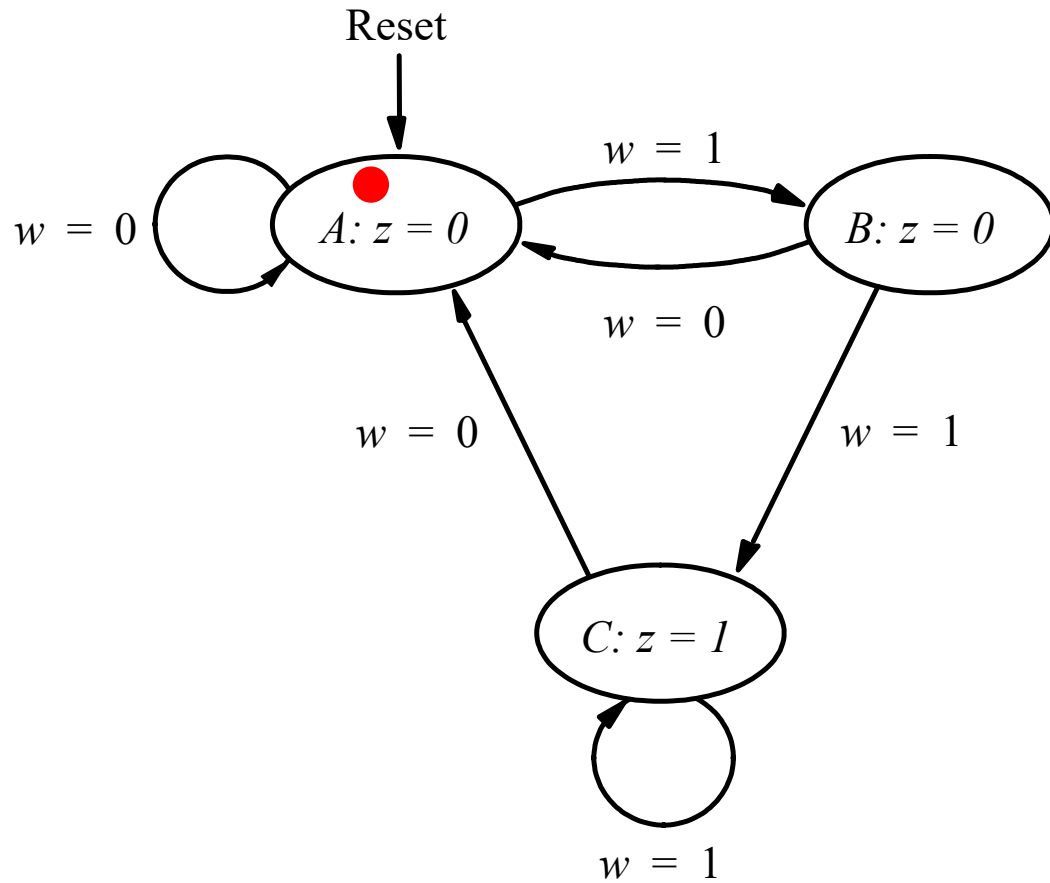
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	0	1	0	0	1	1	0



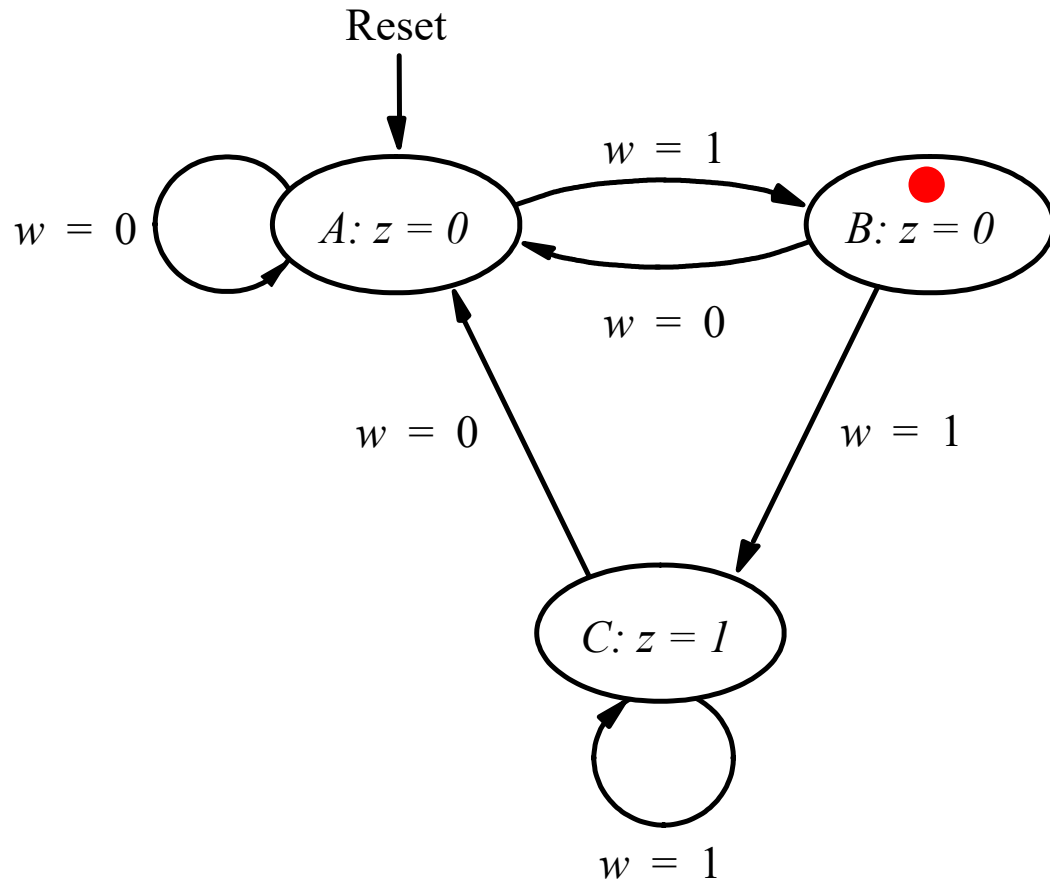
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	0	1	0	0	1	1	0



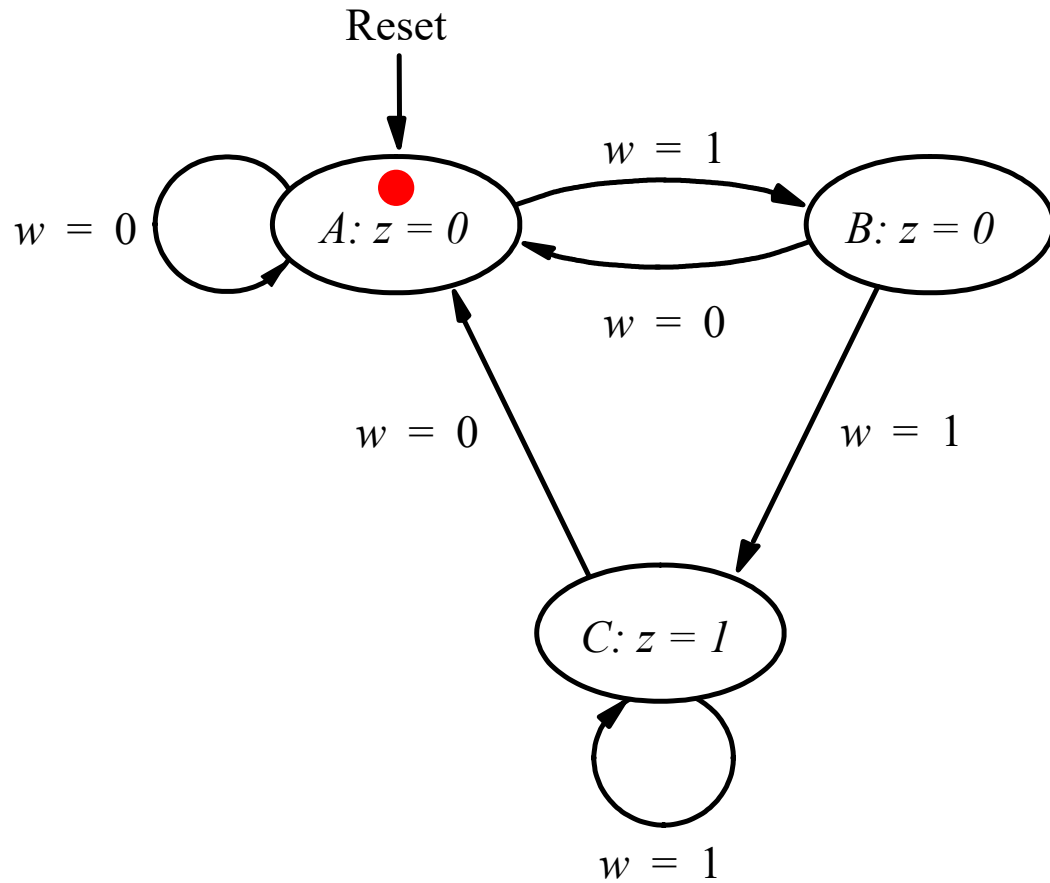
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	0	1	0	0	1	1	0



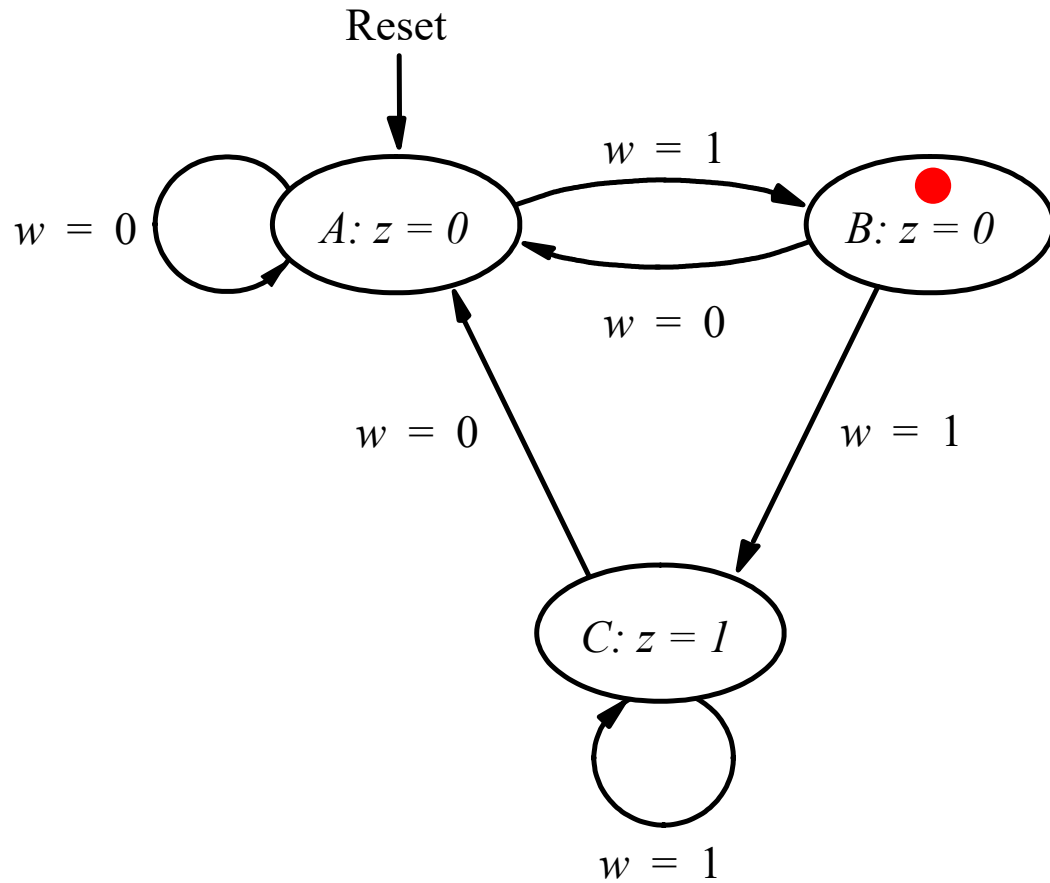
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	0	1	0	0	1	1	0



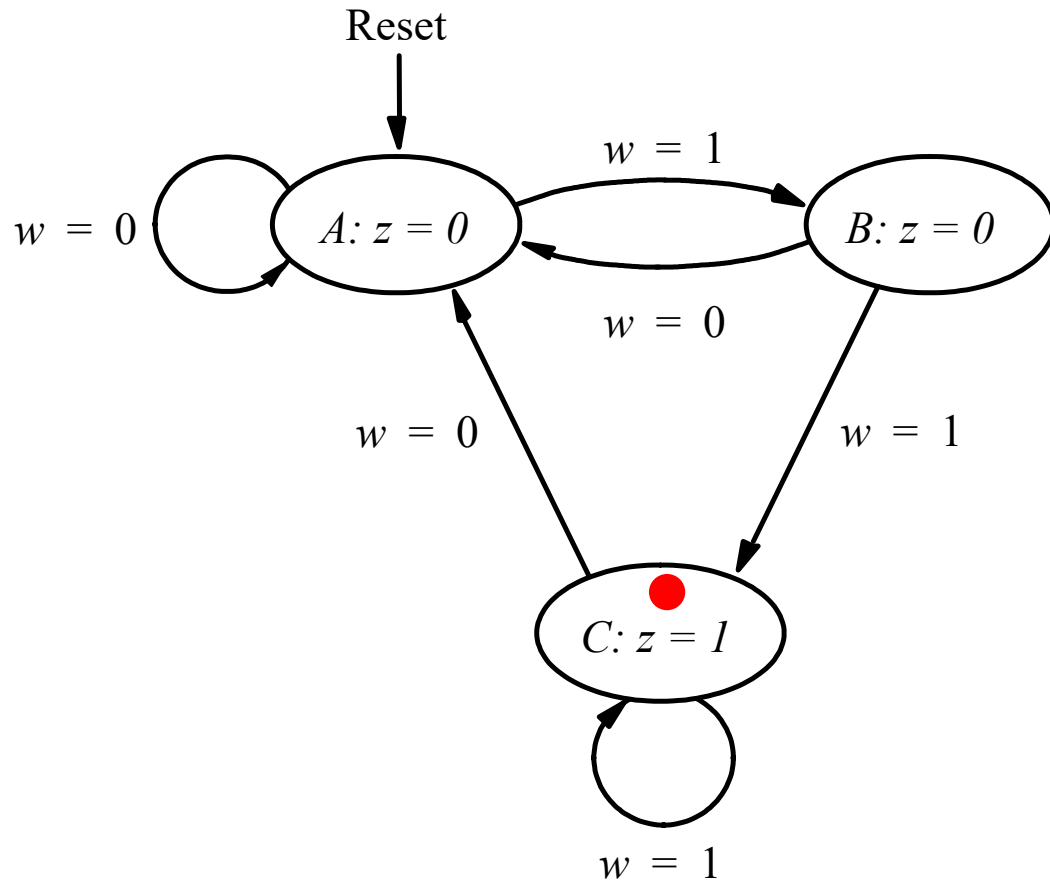
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	0	1	0	0	1	1	0



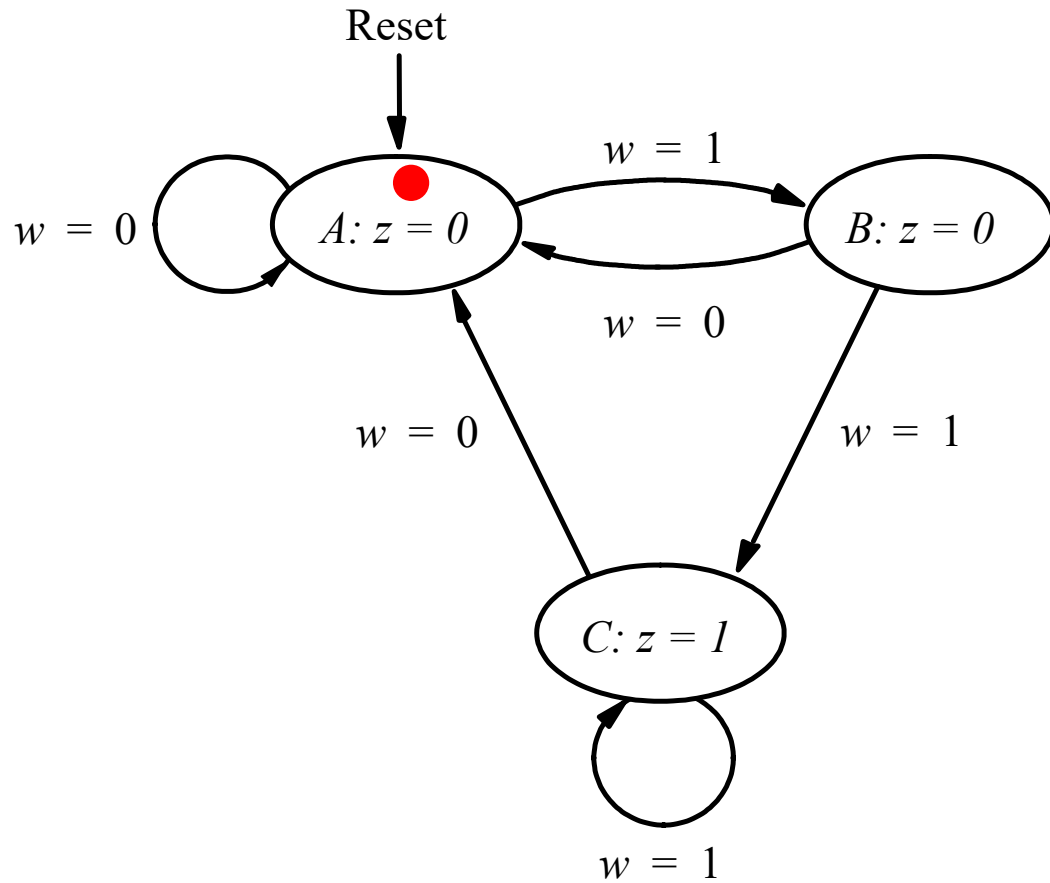
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	0	1	0	0	1	1	0



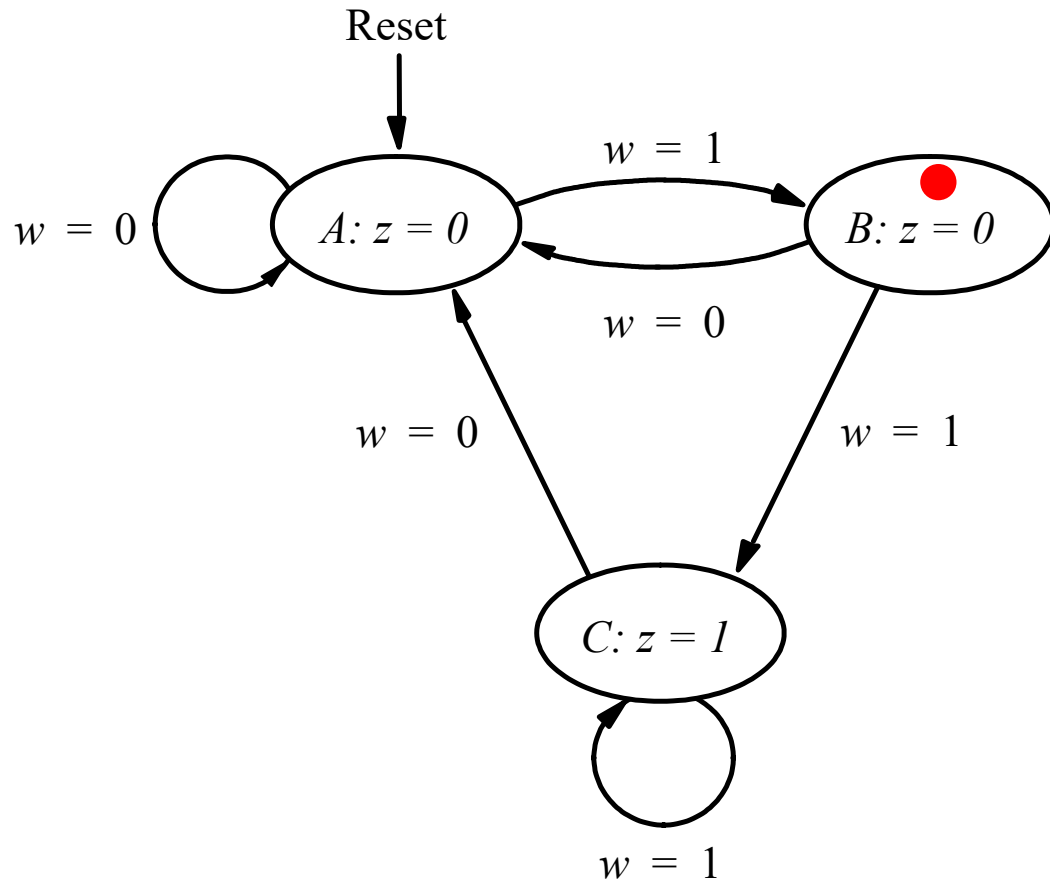
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	0	1	0	0	1	1	0



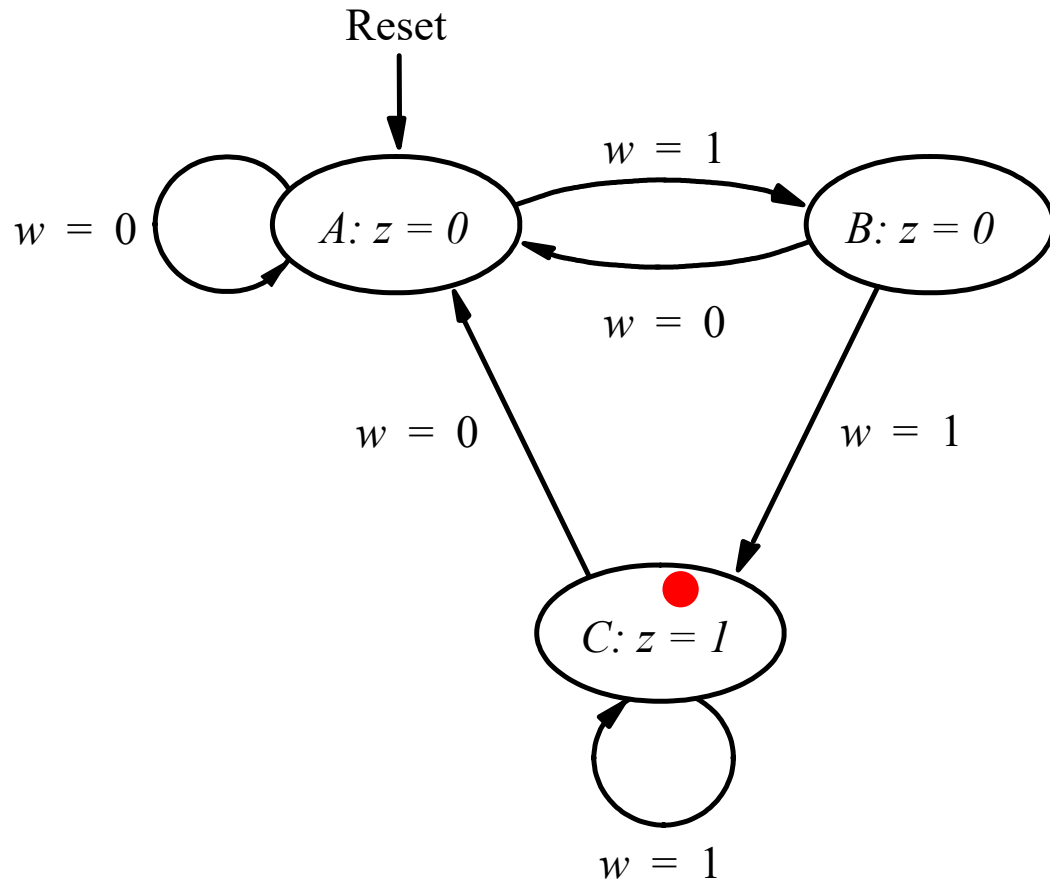
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	0	1	0	0	1	1	0



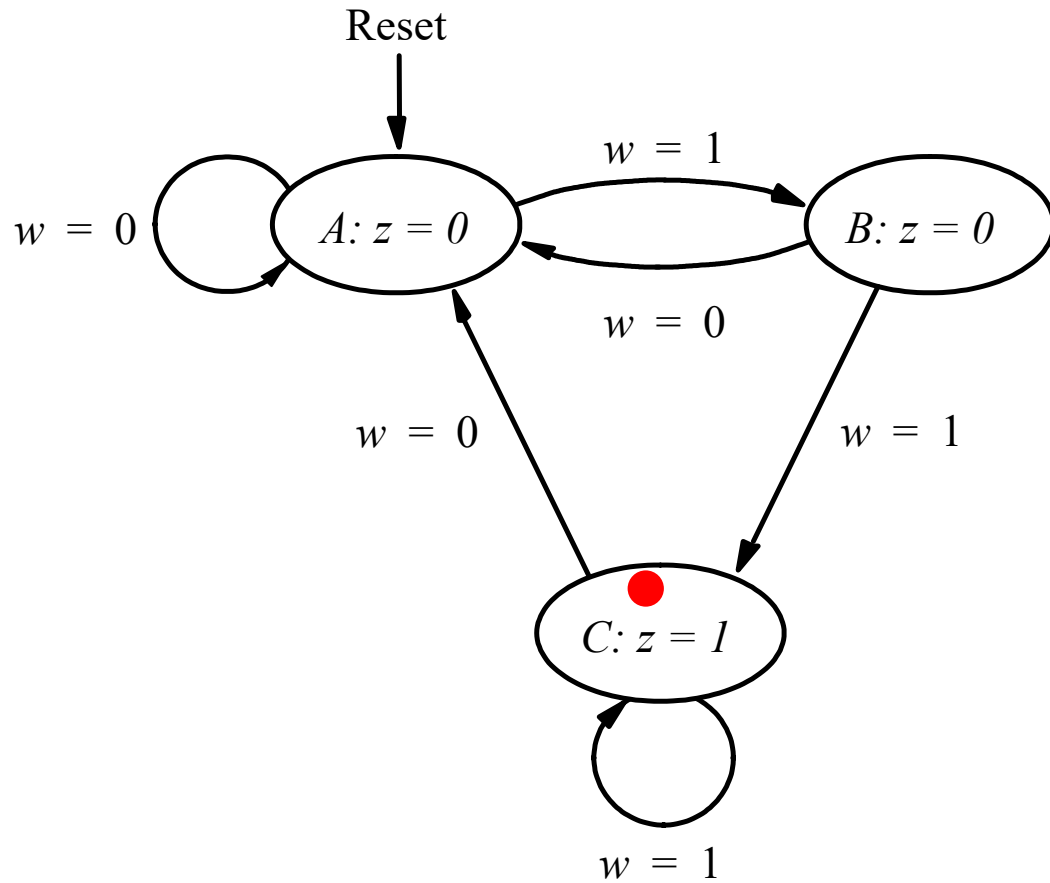
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	0	1	0	0	1	1	0



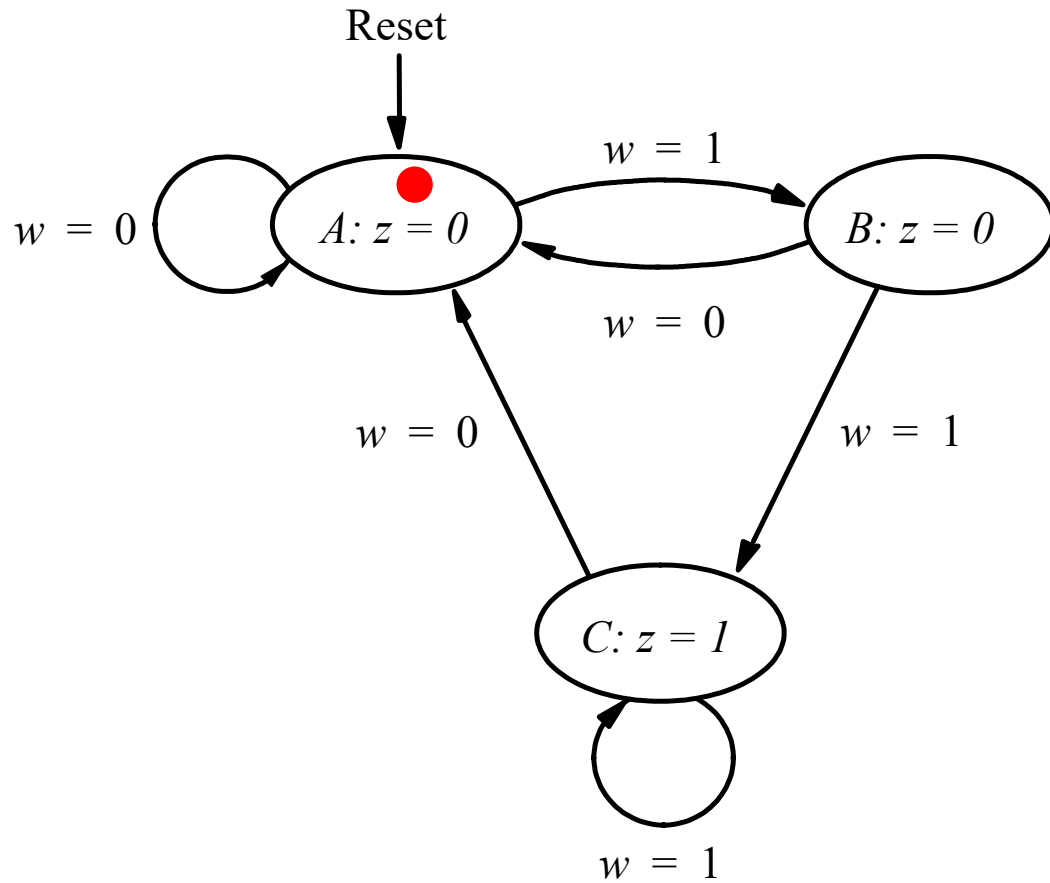
Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	0	1	0	0	1	1	0



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

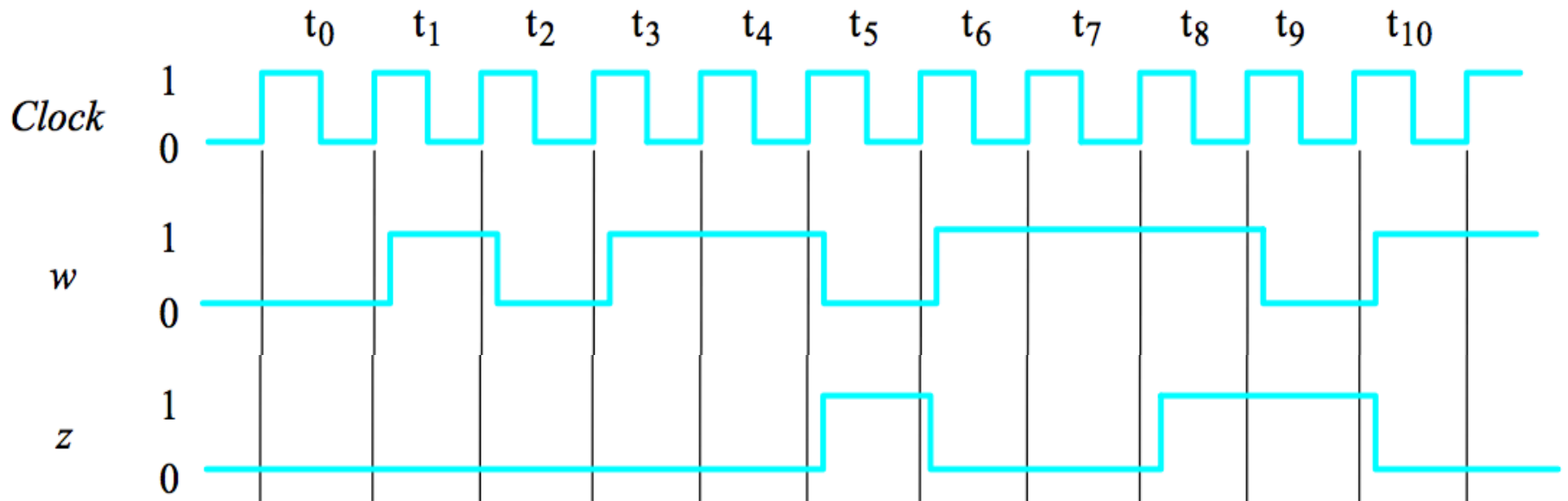


Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	0	1	0	0	1	1	0

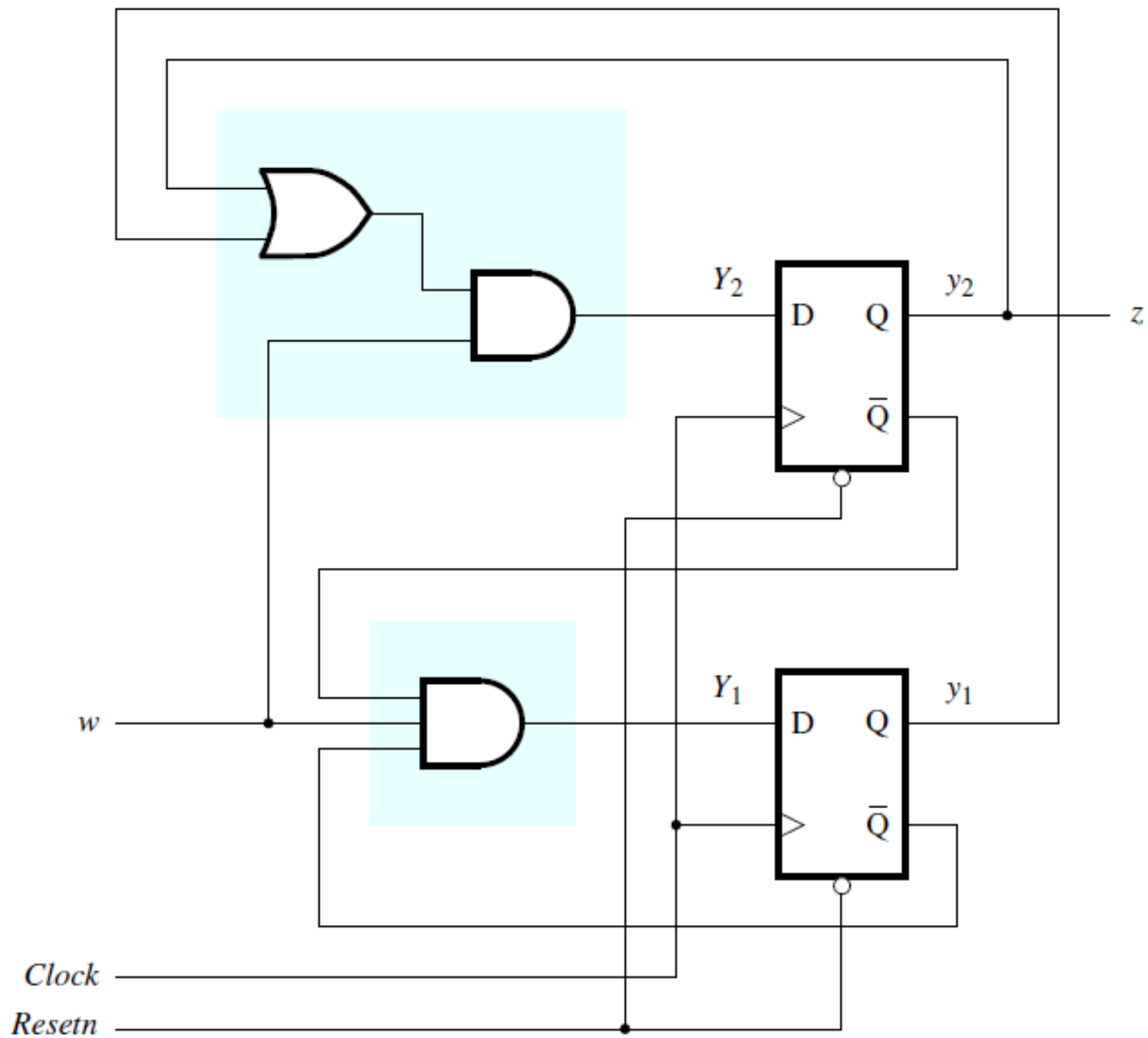


Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	0	1	0	0	1	1	0

Inferring the States

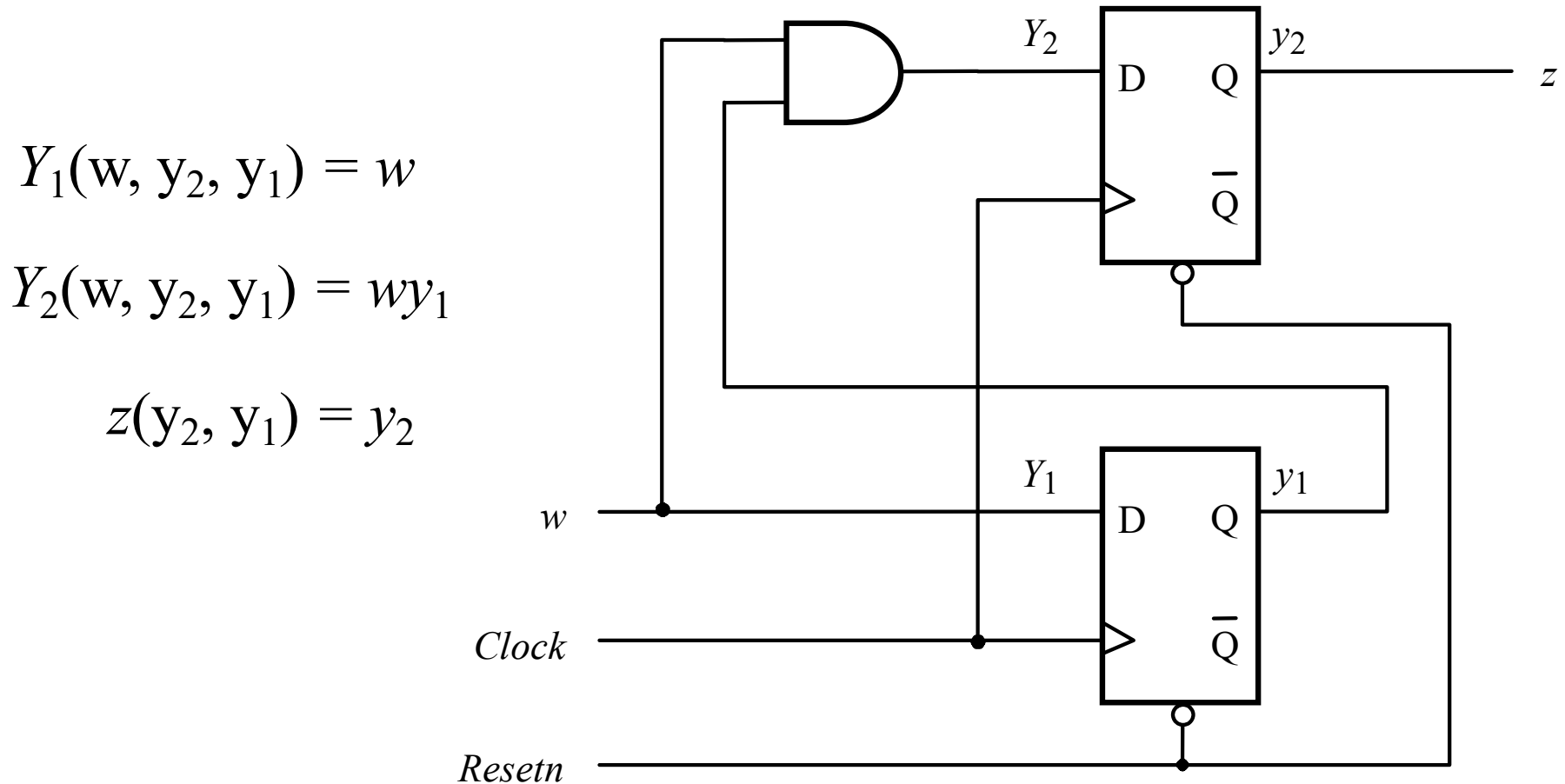


Clockcycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
<i>w</i> :	0	1	0	1	1	0	1	1	1	0	1
<i>z</i> :	0	0	0	0	0	1	0	0	1	1	0



[Figure 6.8 from the textbook]

The New and Improved Circuit Diagram



Moore Machine Implementation

Sequences of input and output signals

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

Sequences of input and output signals

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

Sequences of input and output signals

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

Sequences of input and output signals

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

Sequences of input and output signals

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

Sequences of input and output signals

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

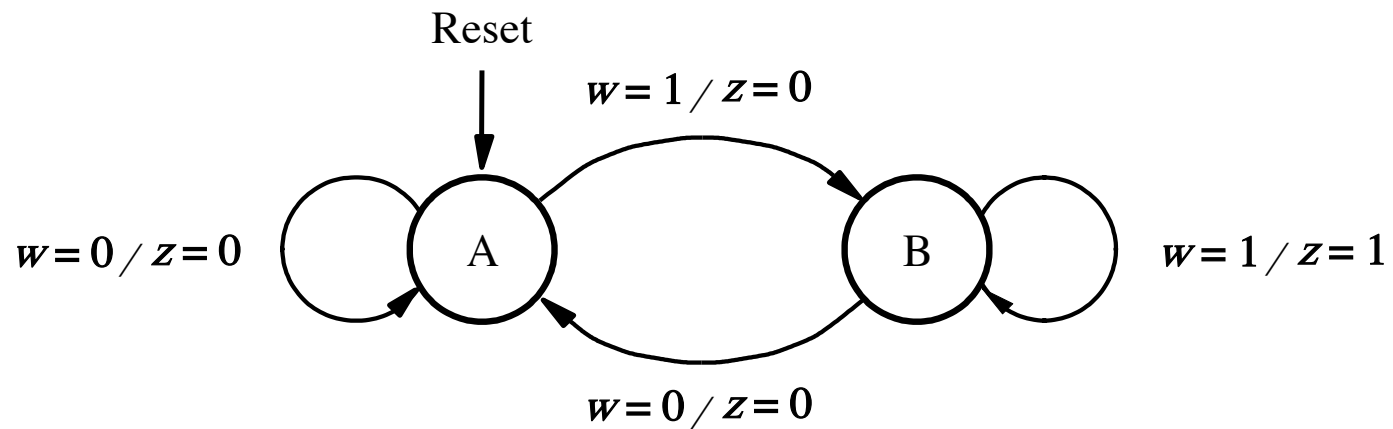
Sequences of input and output signals

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

Sequences of input and output signals

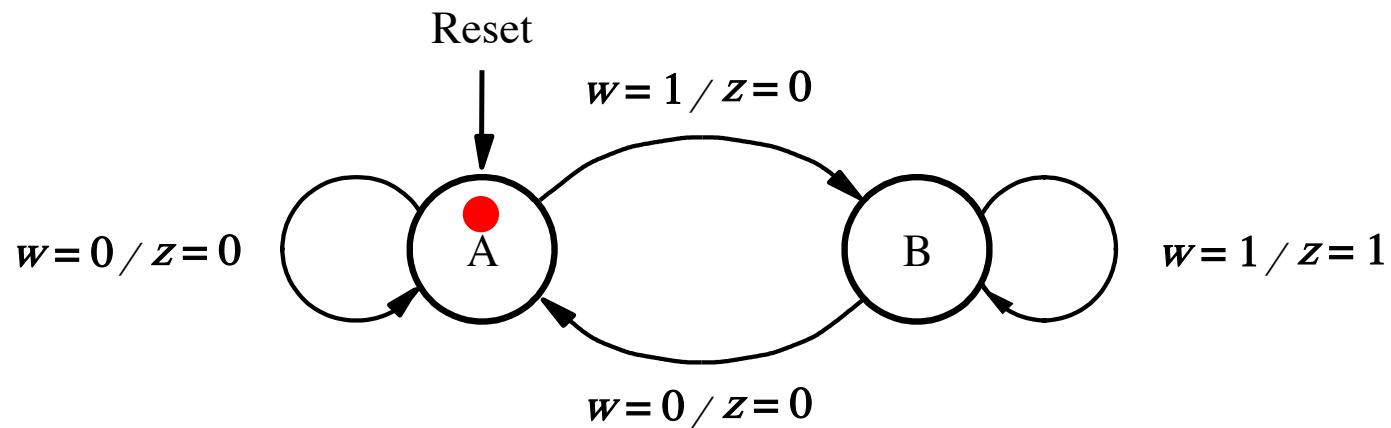
Clock cycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
input w :	0	1	0	1	1	0	1	1	1	0	1
output z :	0	0	0	0	1	0	0	1	1	0	0

State diagram of an FSM that realizes the task



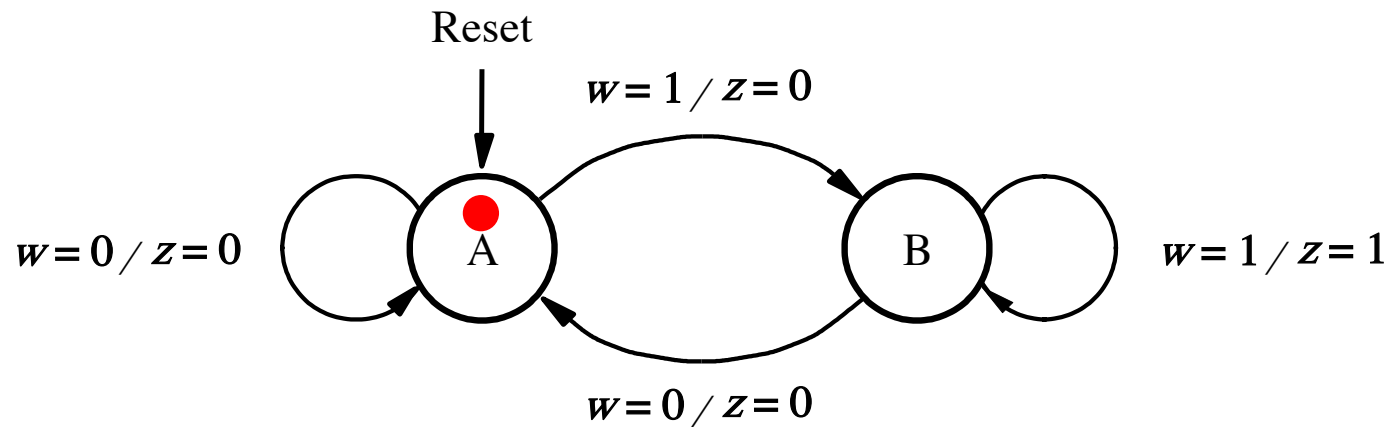
Let's Do a Simulation

Clock cycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
input w :	<input type="checkbox"/> 0	1	0	1	1	0	1	1	1	0	1
output z :	0	0	0	0	1	0	0	1	1	0	0



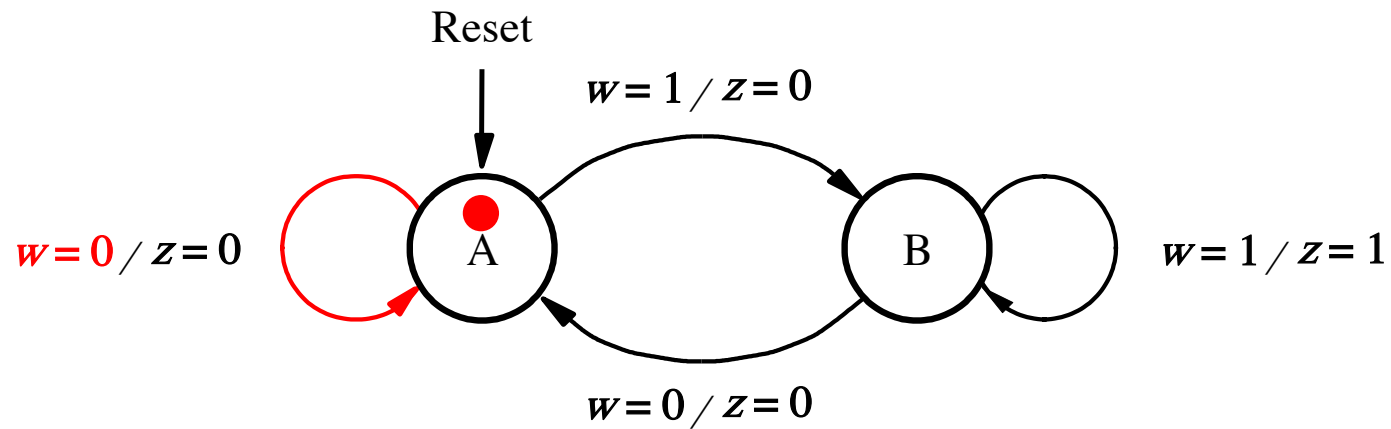
Let's Do a Simulation

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



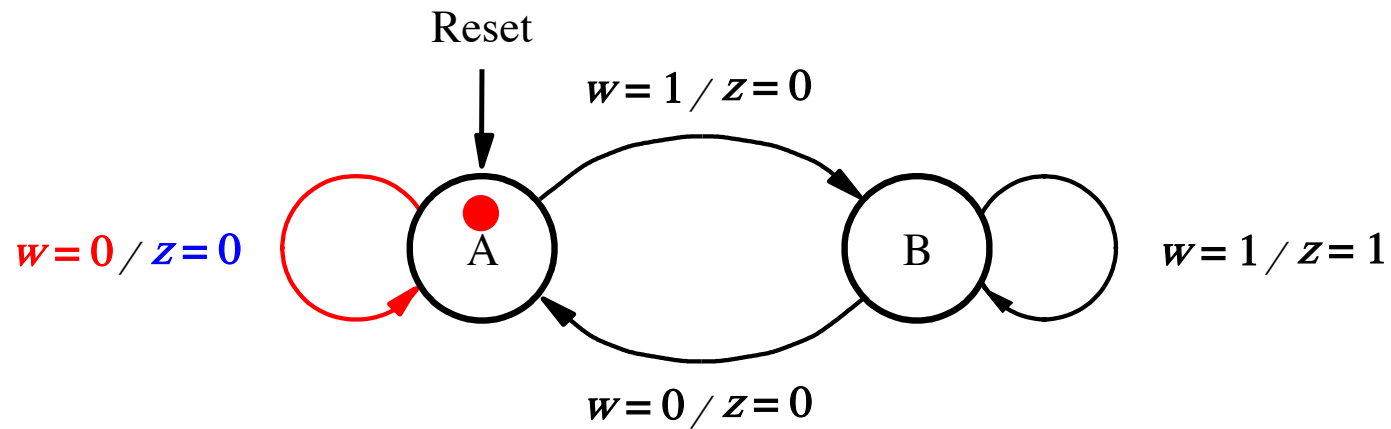
Let's Do a Simulation

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



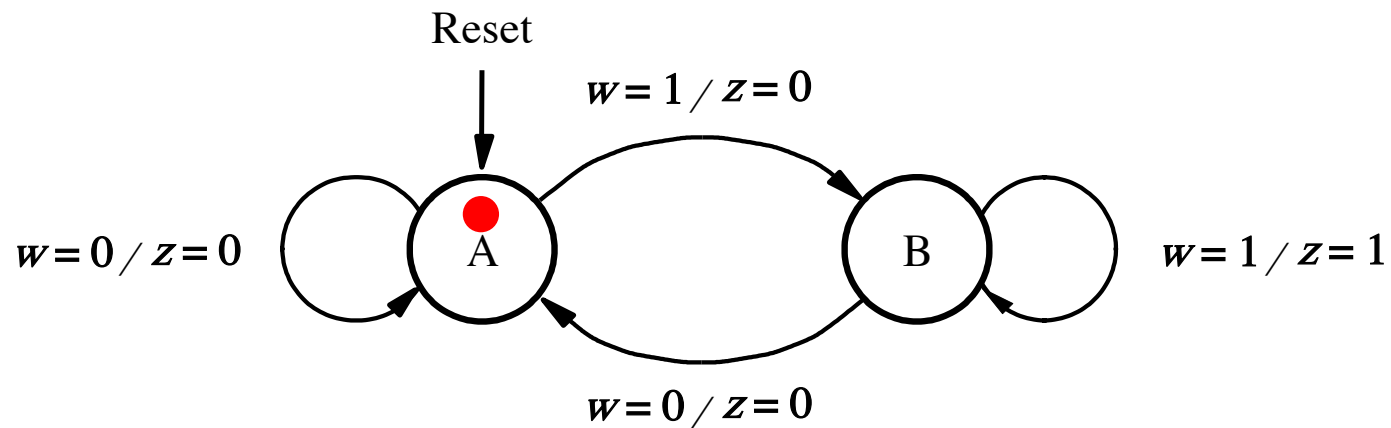
Let's Do a Simulation

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



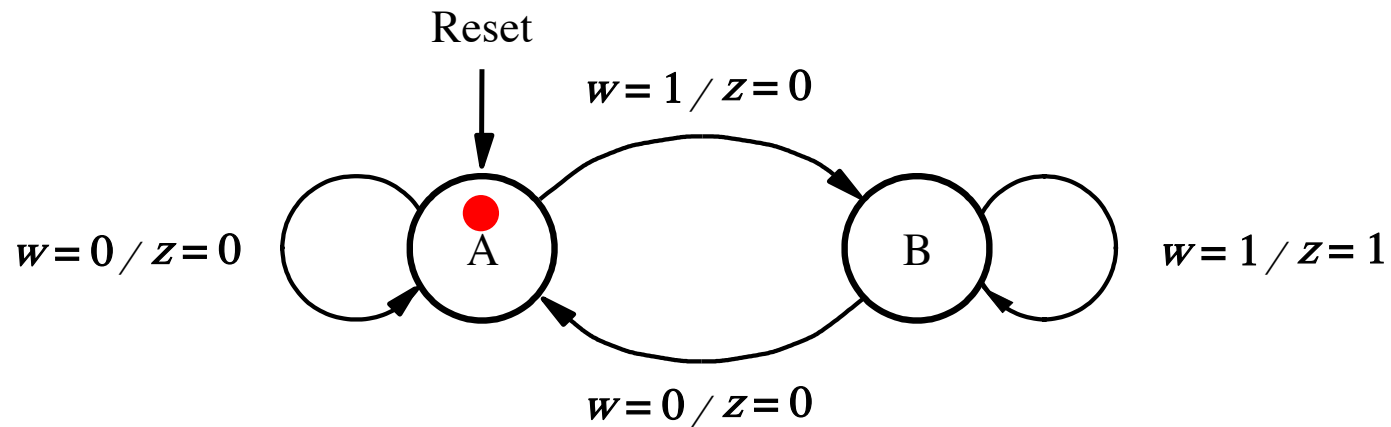
Let's Do a Simulation

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



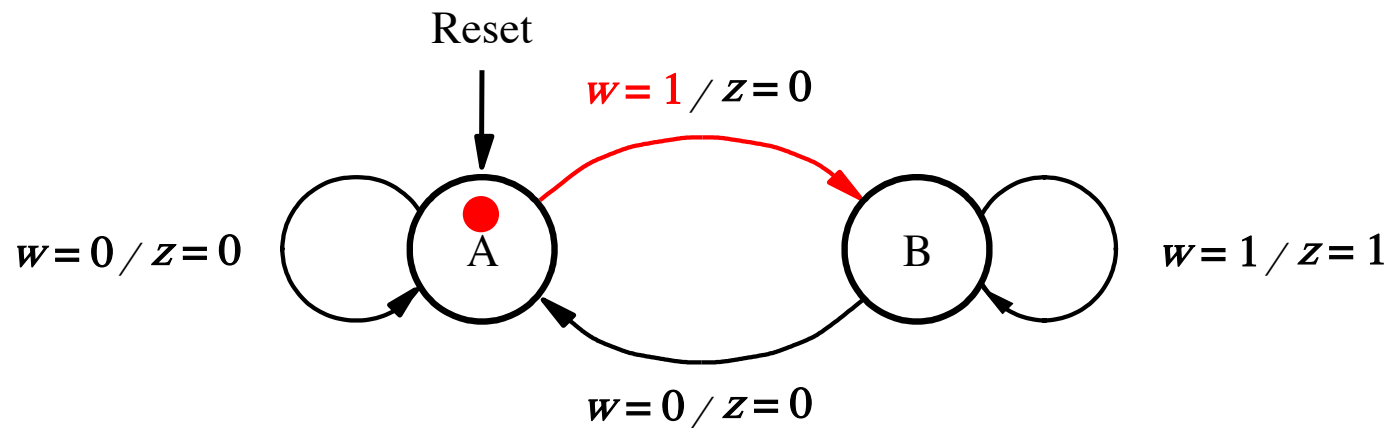
Let's Do a Simulation

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



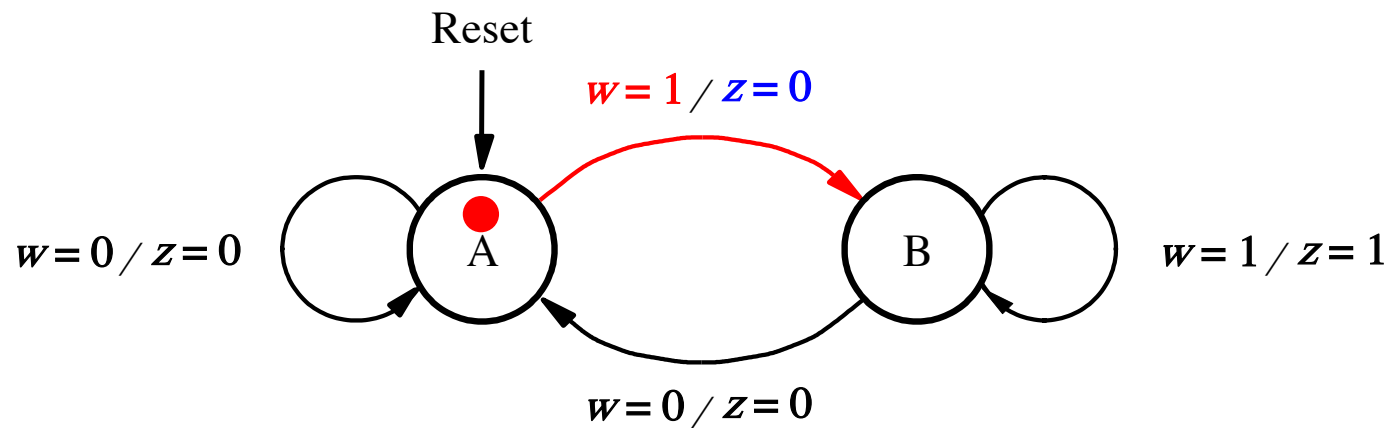
Let's Do a Simulation

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



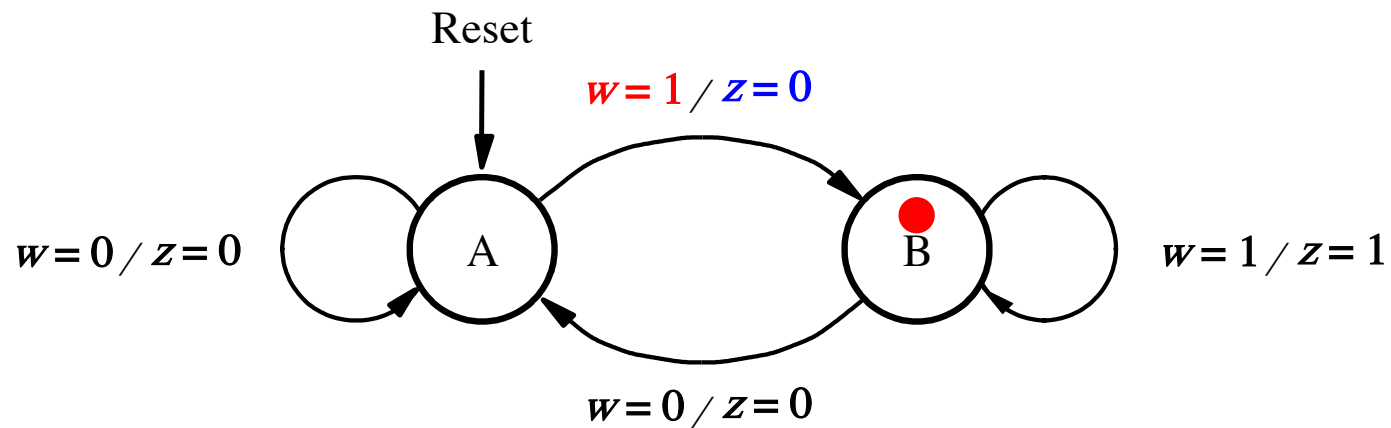
Let's Do a Simulation

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



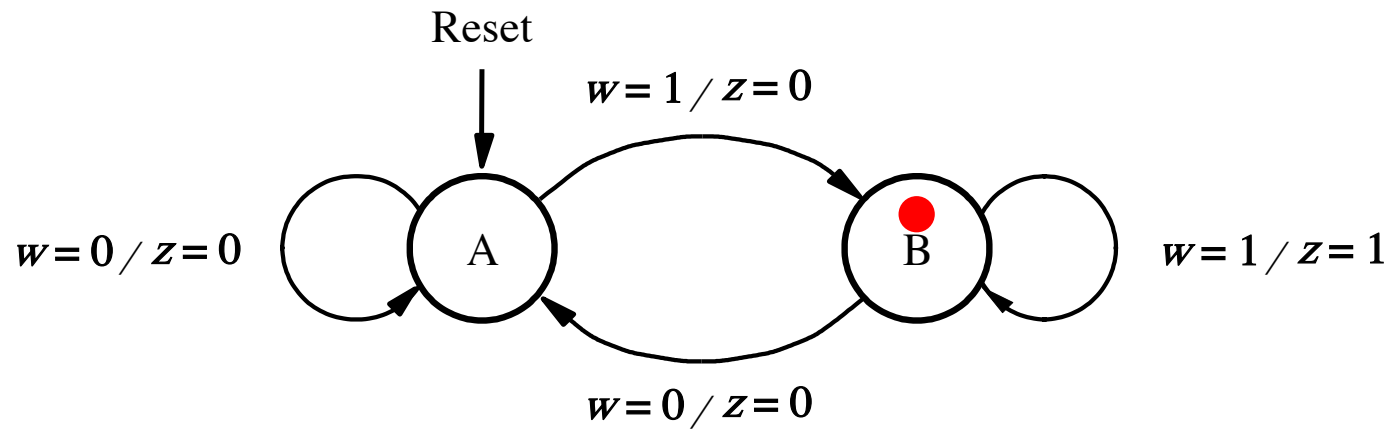
Let's Do a Simulation

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



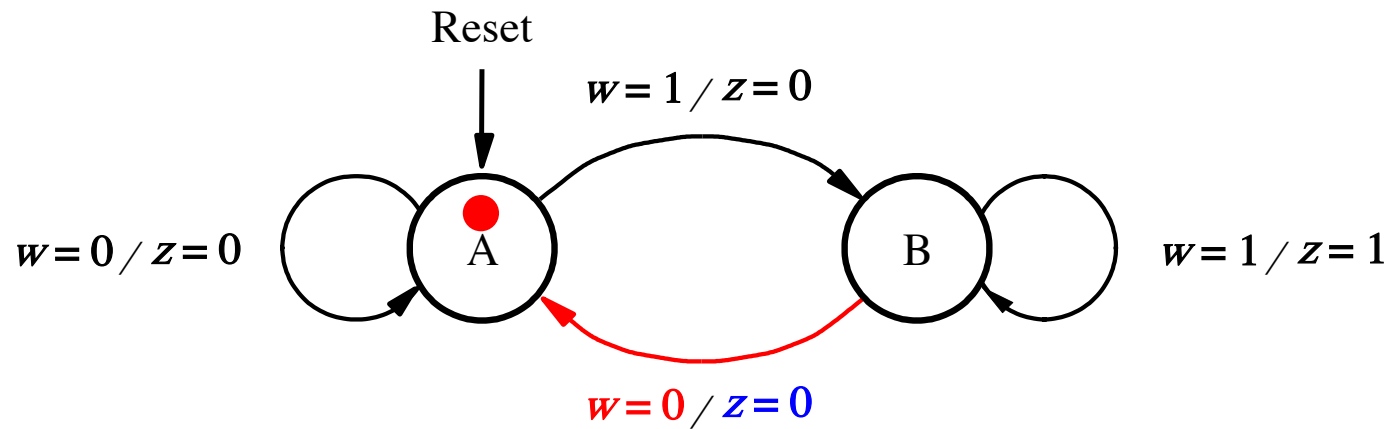
Let's Do a Simulation

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



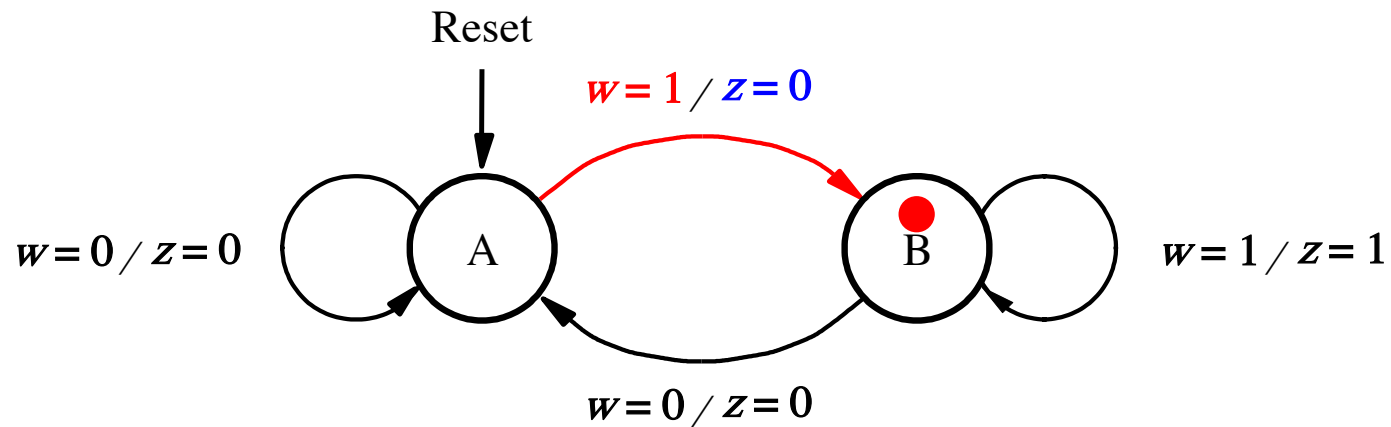
Let's Do a Simulation

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



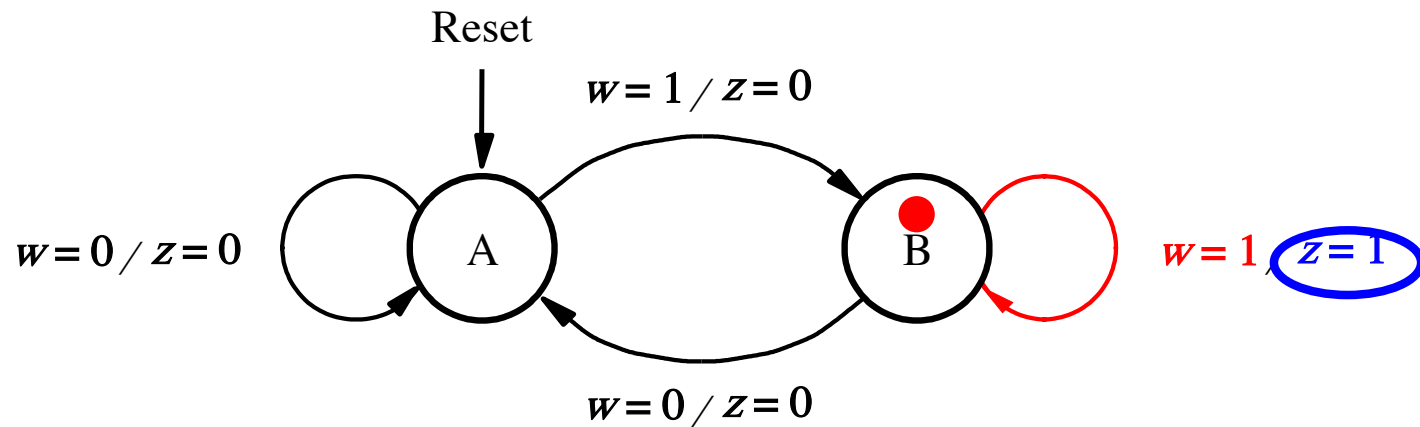
Let's Do a Simulation

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



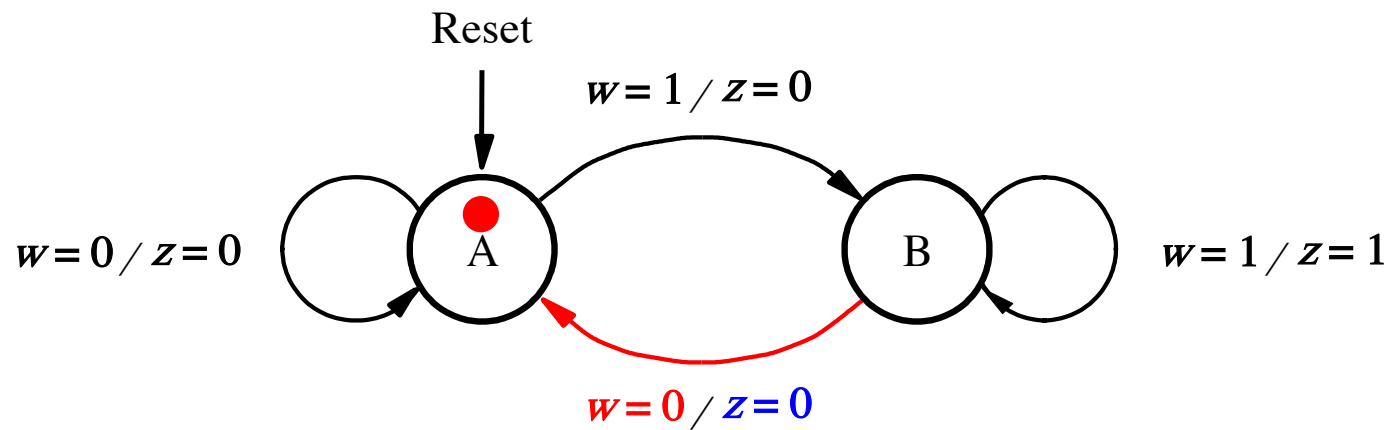
Let's Do a Simulation

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



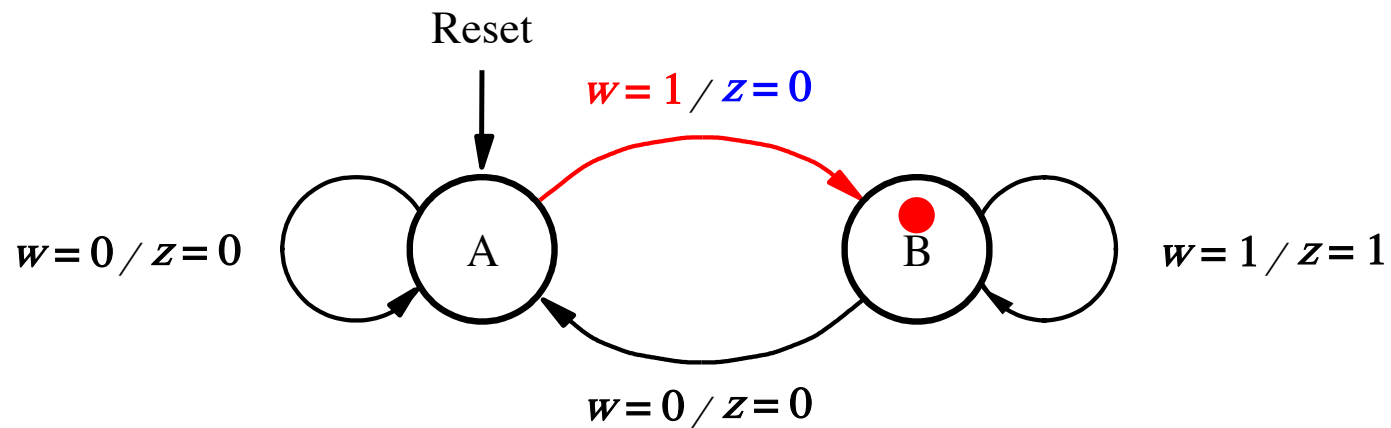
Let's Do a Simulation

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



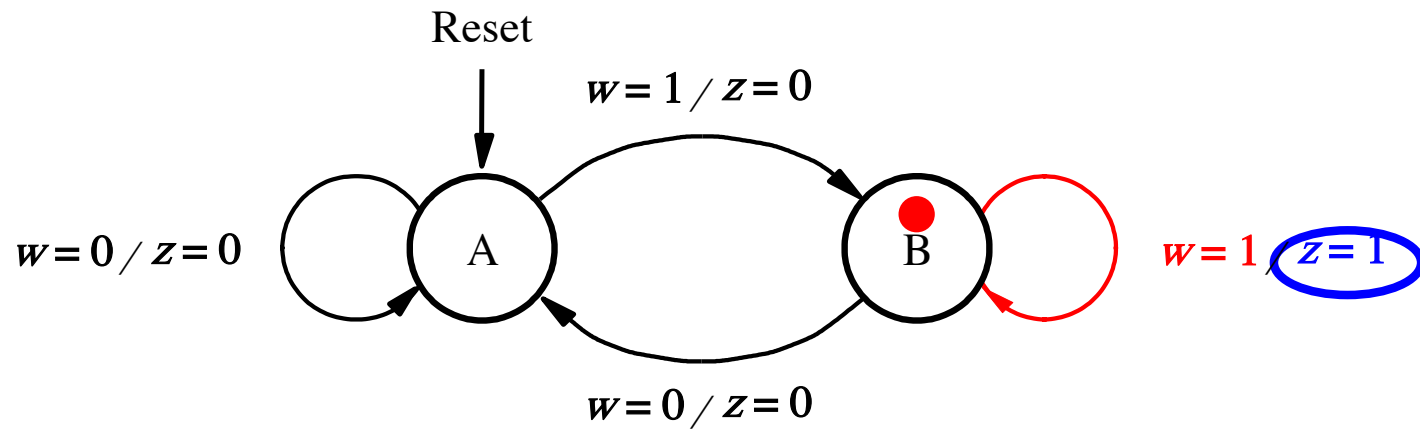
Let's Do a Simulation

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



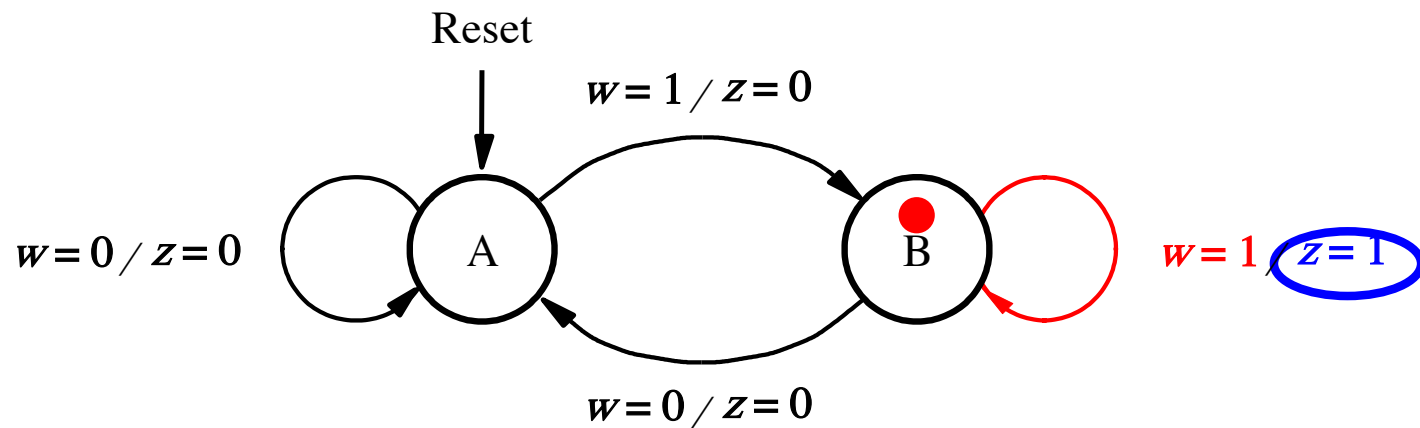
Let's Do a Simulation

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



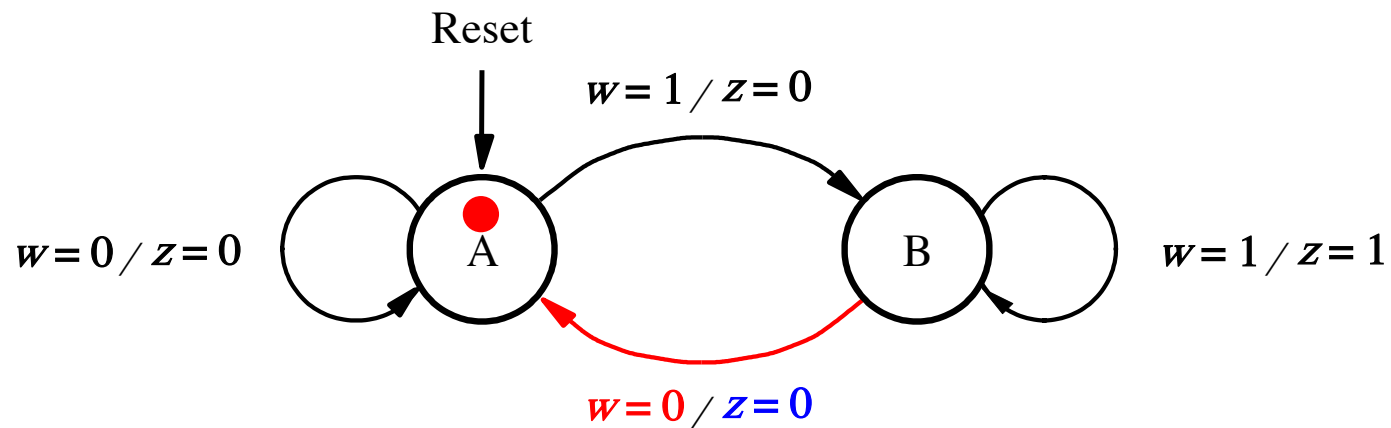
Let's Do a Simulation

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



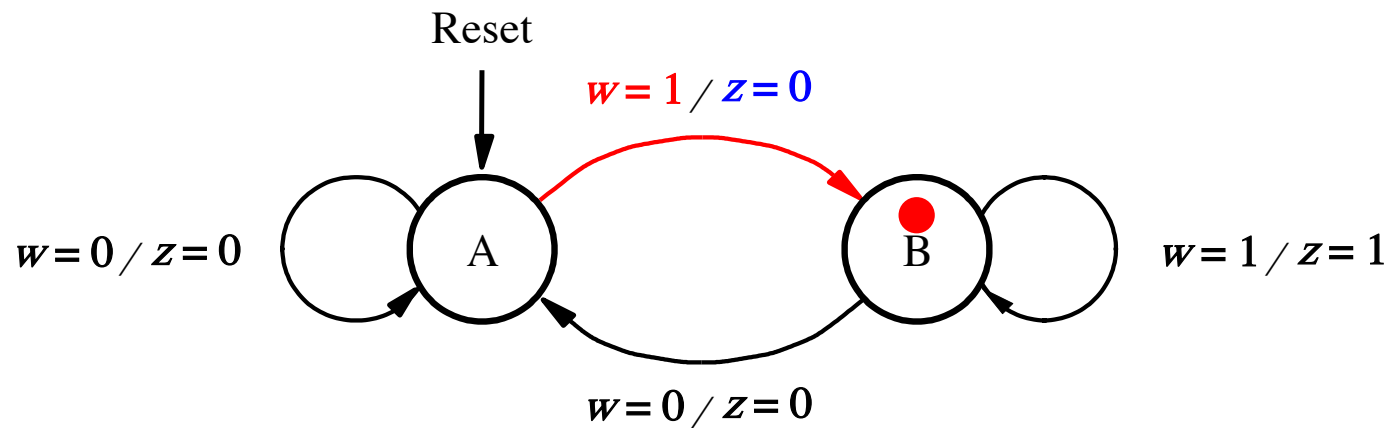
Let's Do a Simulation

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

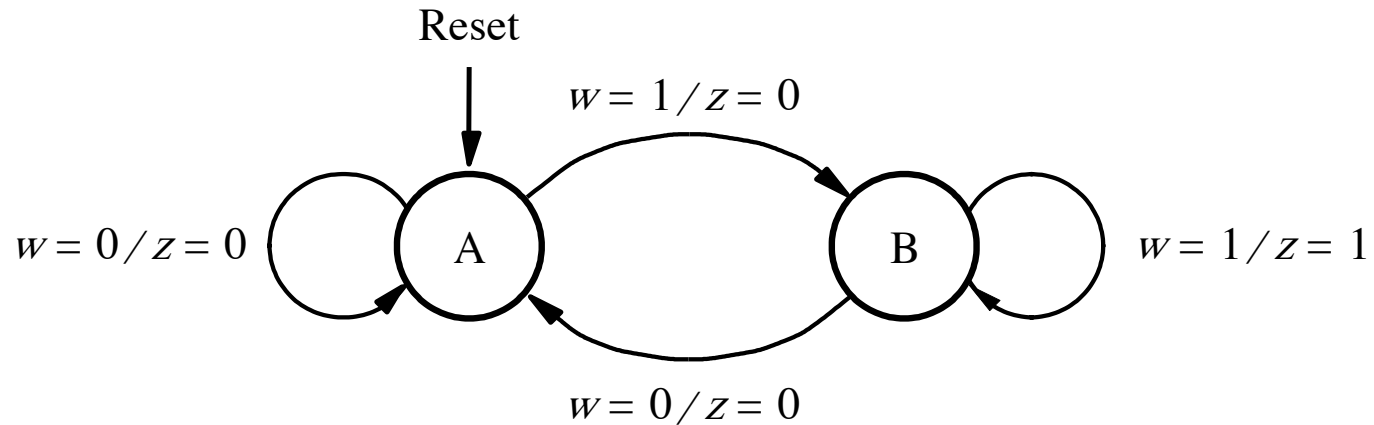


Let's Do a Simulation

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

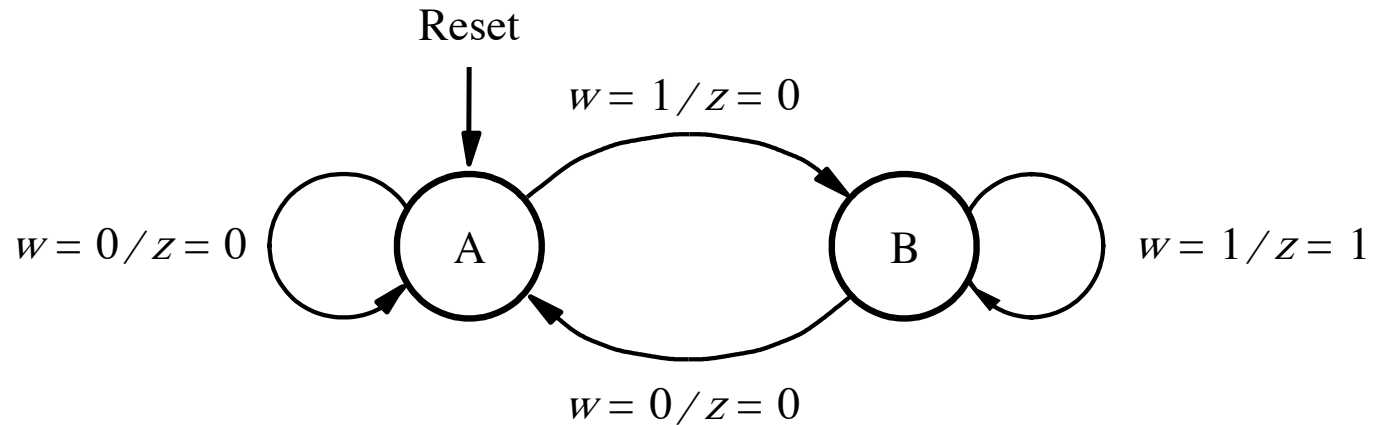


Now Let's Do the State Table for this FSM



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

Now Let's Do the State Table for this FSM



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

The State Table for this FSM

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

Let's Do the State-assigned Table

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

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

Let's Do the State-assigned Table

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

	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

The State-assigned 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

The State-assigned Table

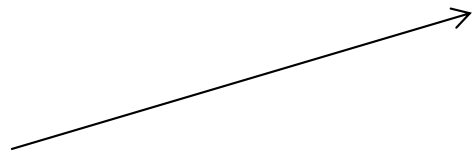
	Present state	Next state		Output	
		$w = 0$	$w = 1$	$w = 0$	$w = 1$
	y	Y	Y	z	z
A	0	0	1	0	0
B	1	0	1	0	1

$$Y = D = w \quad z = wy$$

The State-assigned Table

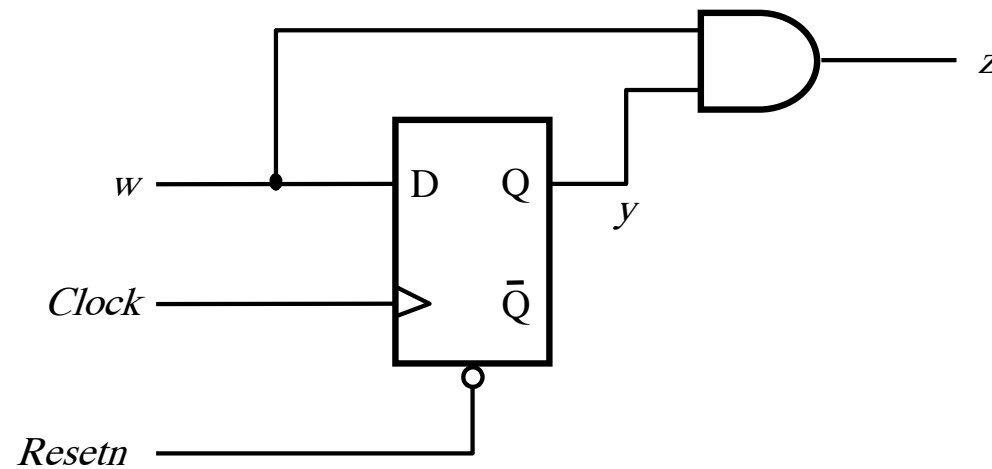
	Present state	Next state		Output	
		$w = 0$	$w = 1$	$w = 0$	$w = 1$
	y	Y	Y	z	z
A	0	0	1	0	0
B	1	0	1	0	1

$$Y = D = w \quad z = wy$$



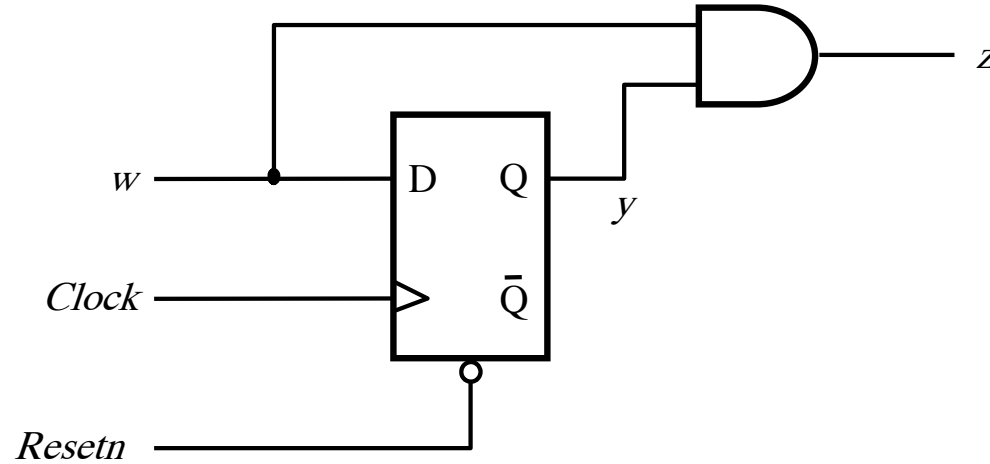
This assumes D flip-flop

Circuit Implementation of the FSM

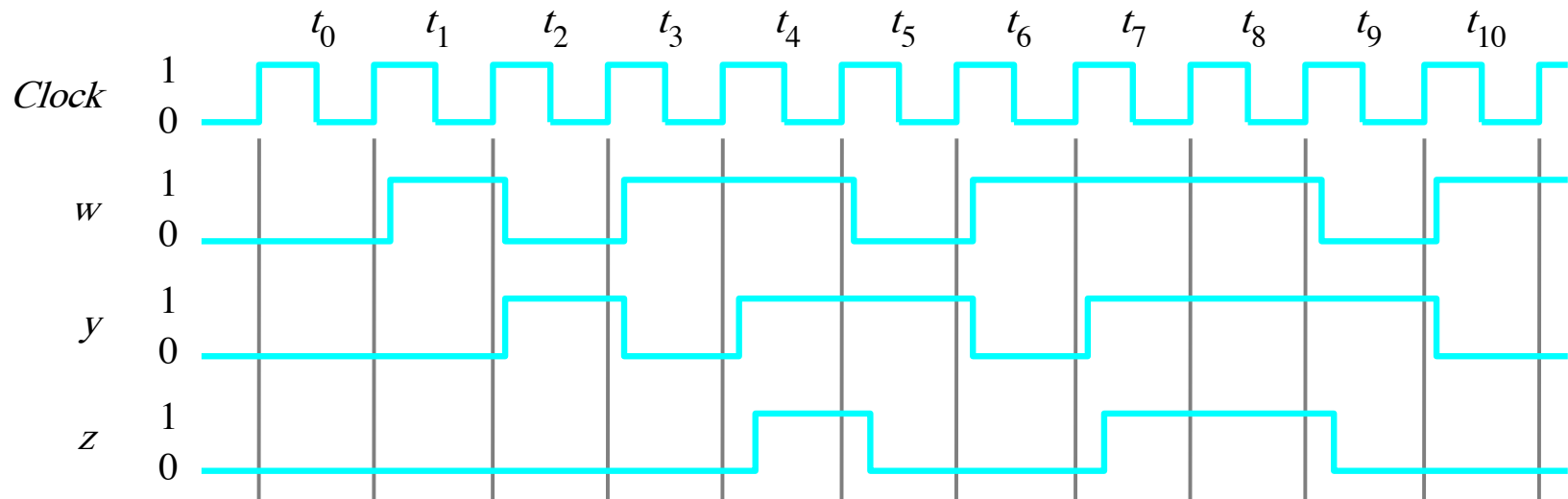


$$Y = D = w \quad z = wy$$

Circuit & Timing Diagram



(a) Circuit

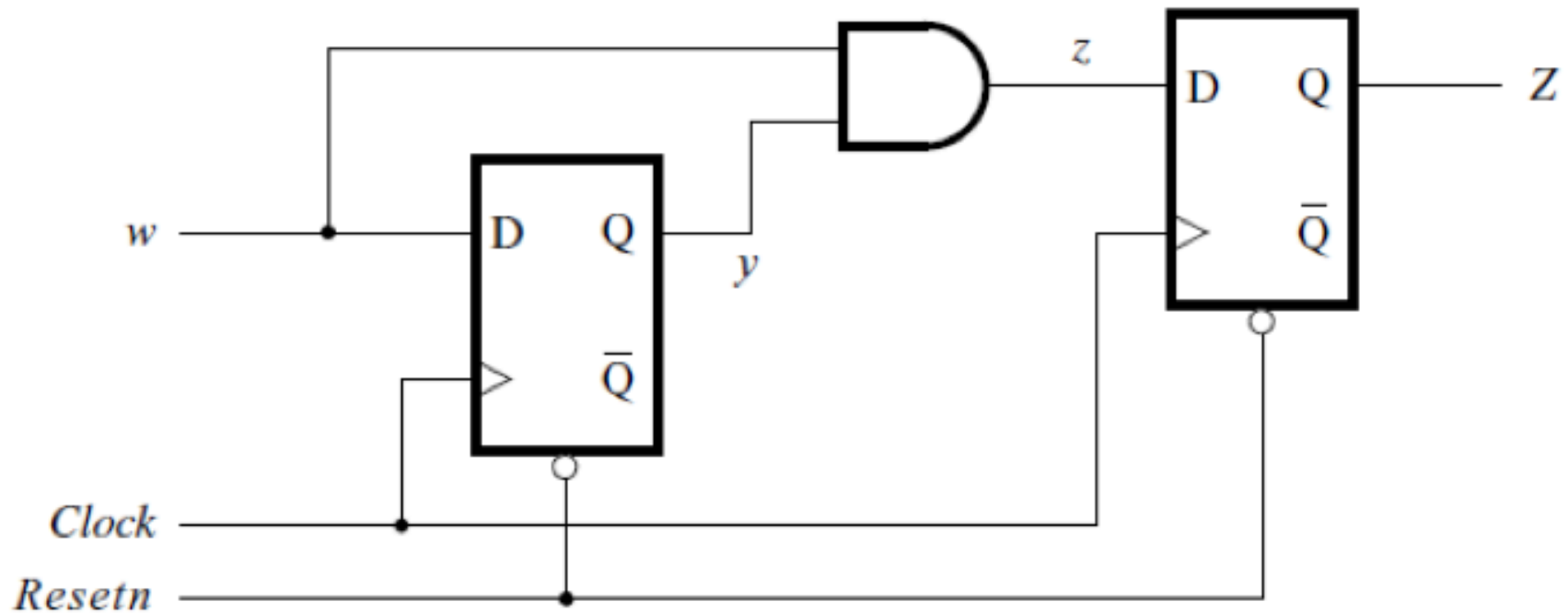


(b) Timing diagram

[Figure 6.26 from the textbook]

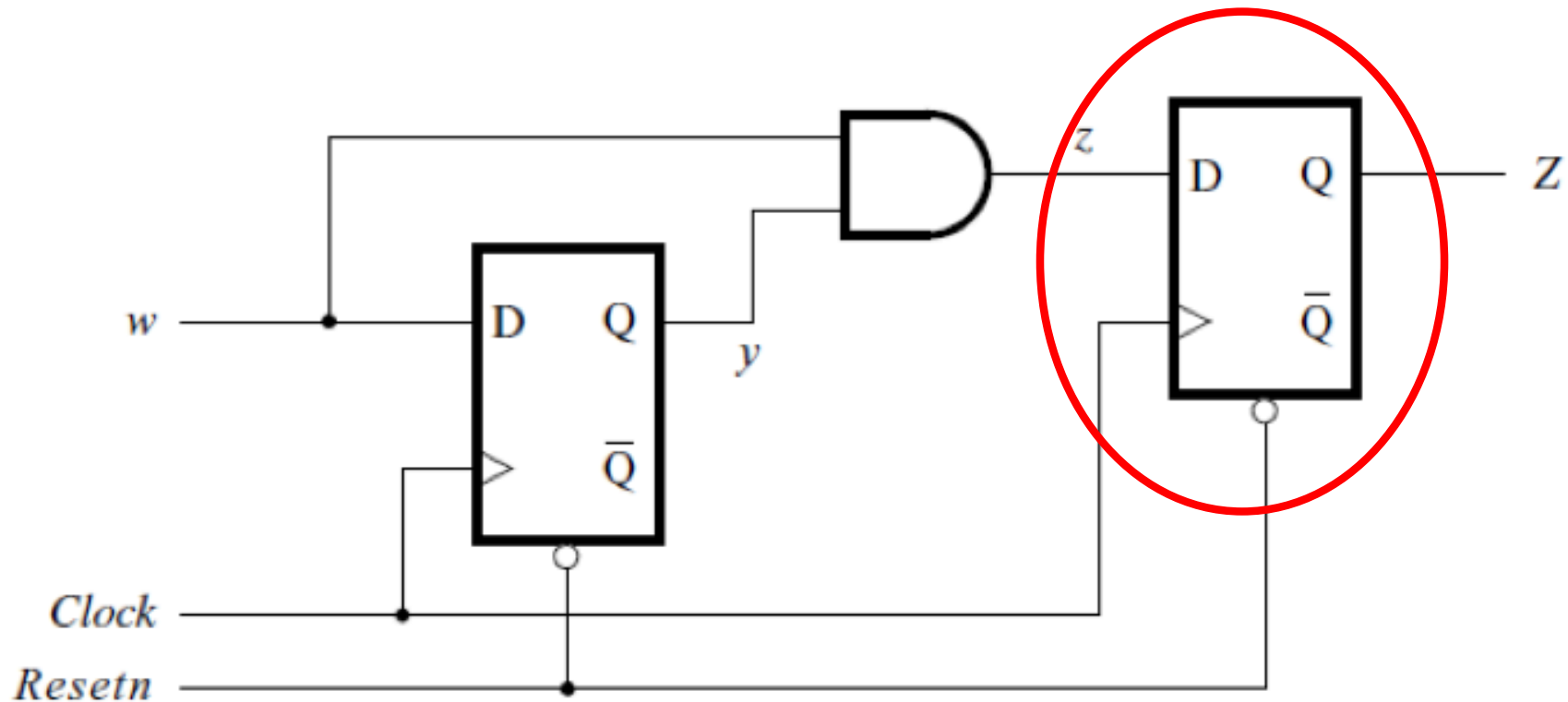
What if we wanted the output signal to be delayed by 1 clock cycle?

Circuit Implementation of the Modified FSM



[Figure 6.27a from the textbook]

Circuit Implementation of the Modified FSM



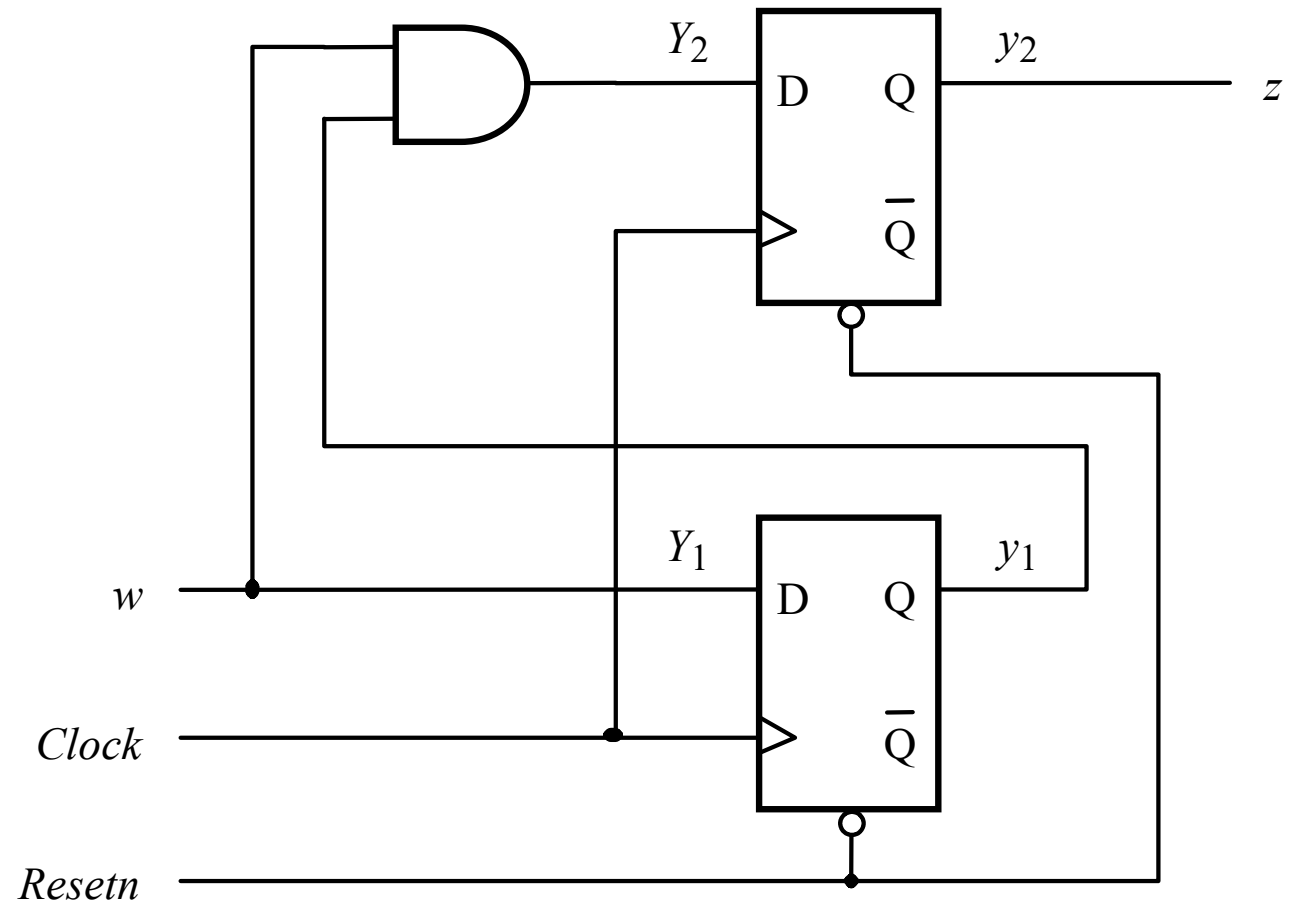
This flip-flop delays the output signal by one clock cycle

We Have Seen This Diagram Before

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

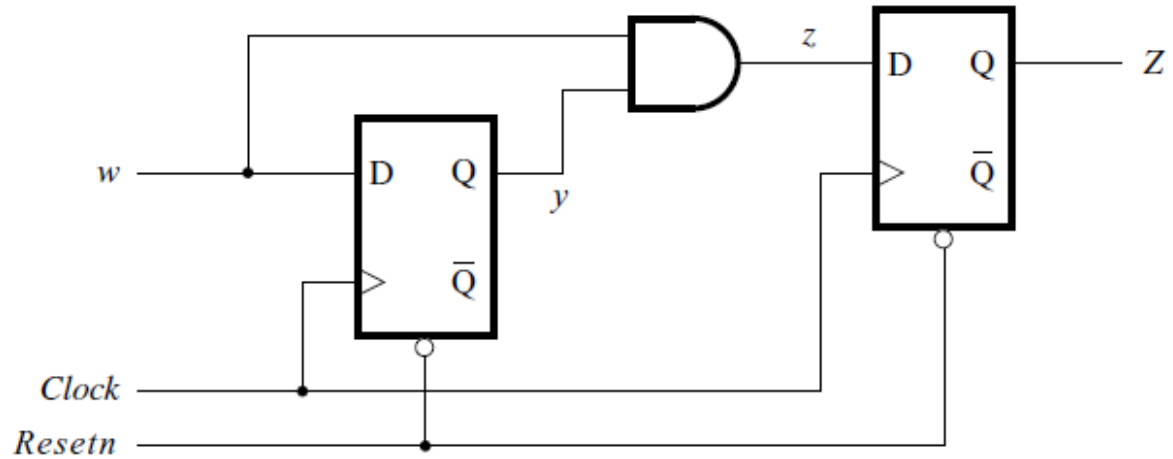
$$Y_2(w, y_2, y_1) = wy_1$$

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

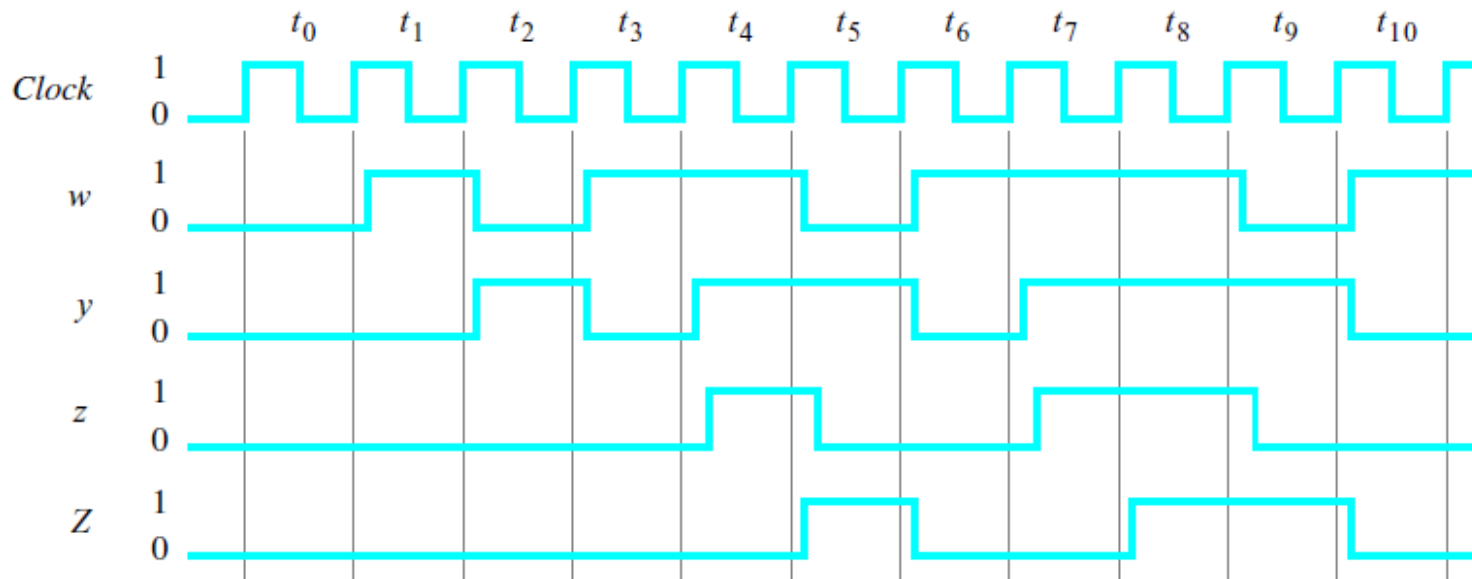


[Figure 6.17 from the textbook]

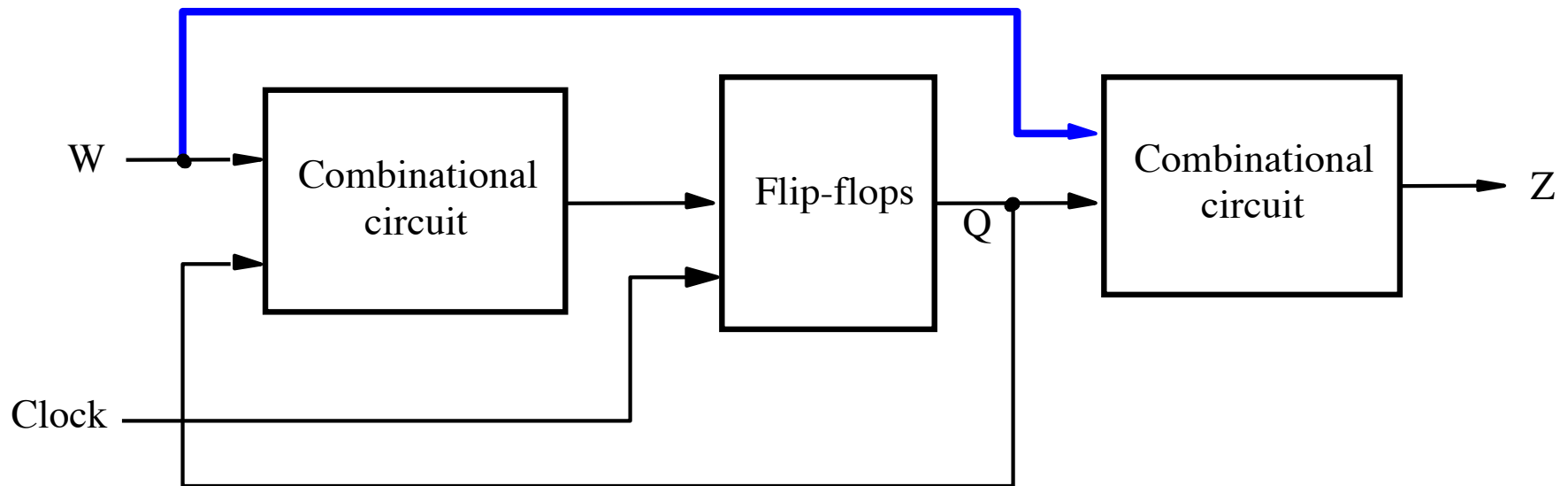
Circuit & Timing Diagram



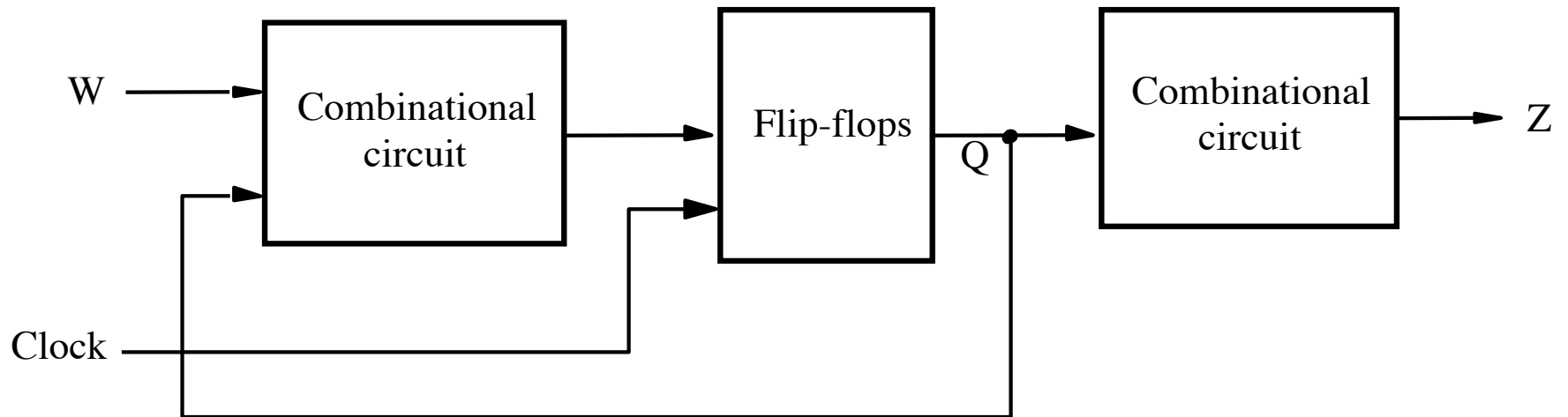
(a) Circuit



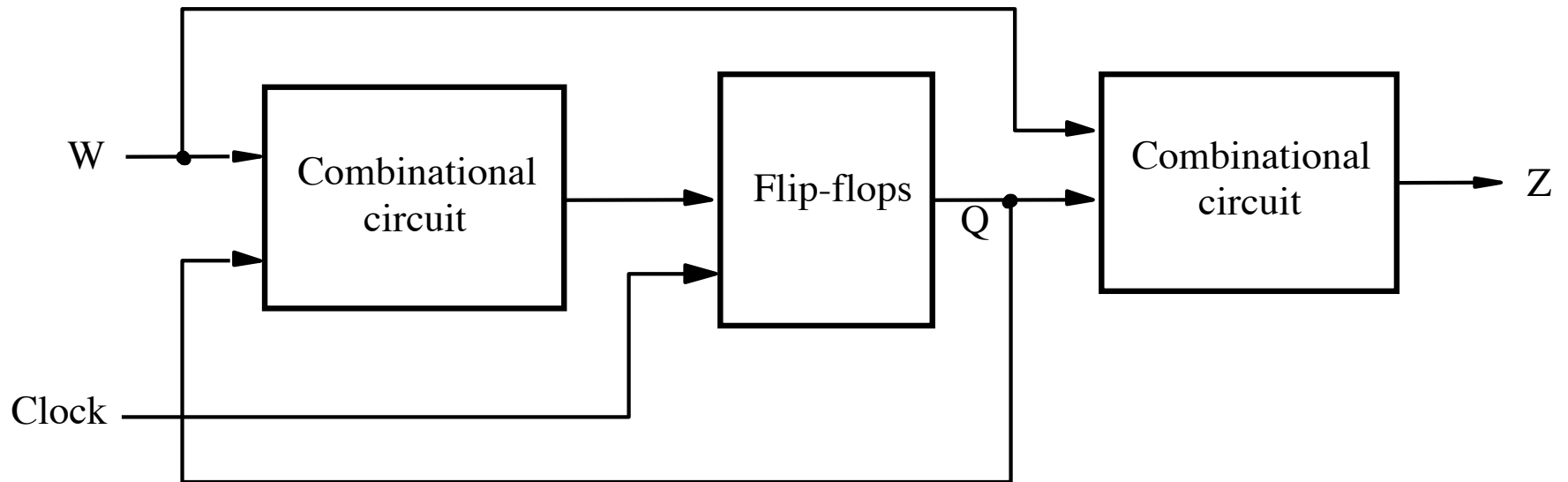
The general form of a synchronous sequential circuit



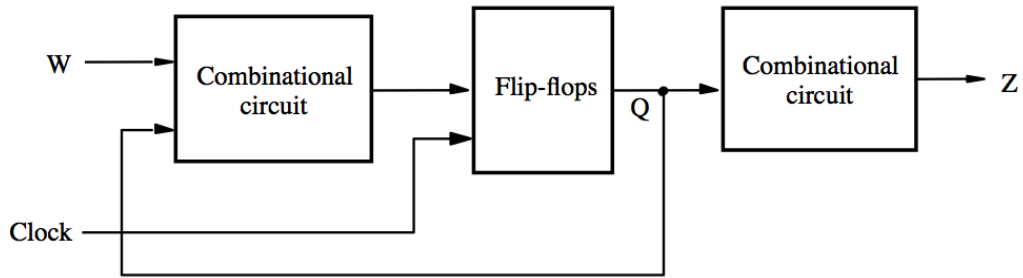
Moore Type



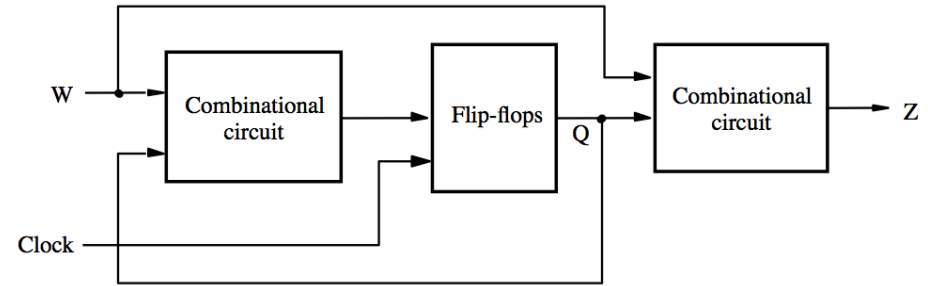
Mealy Type



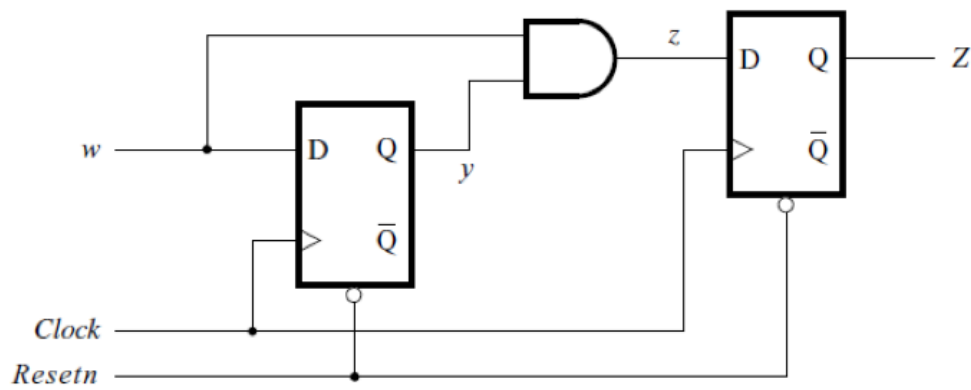
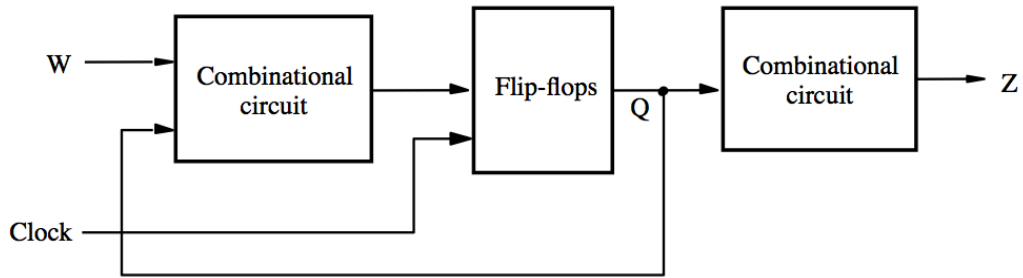
Moore



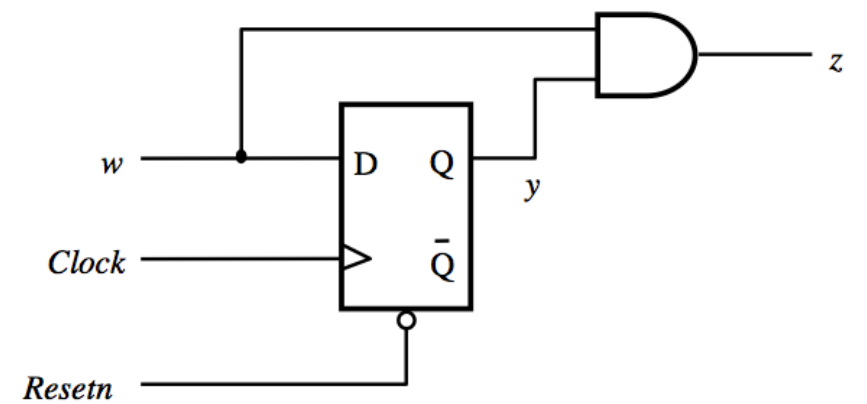
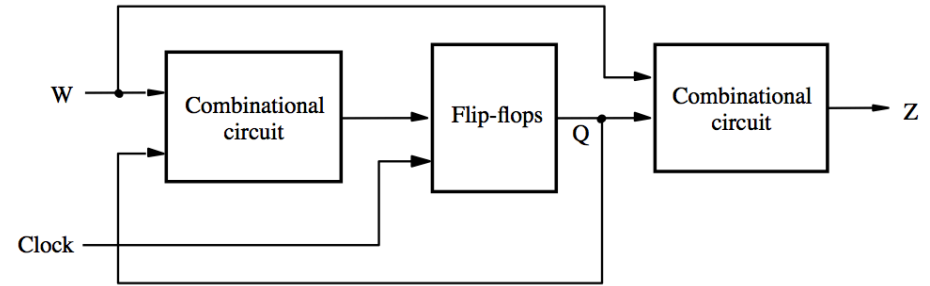
Mealy



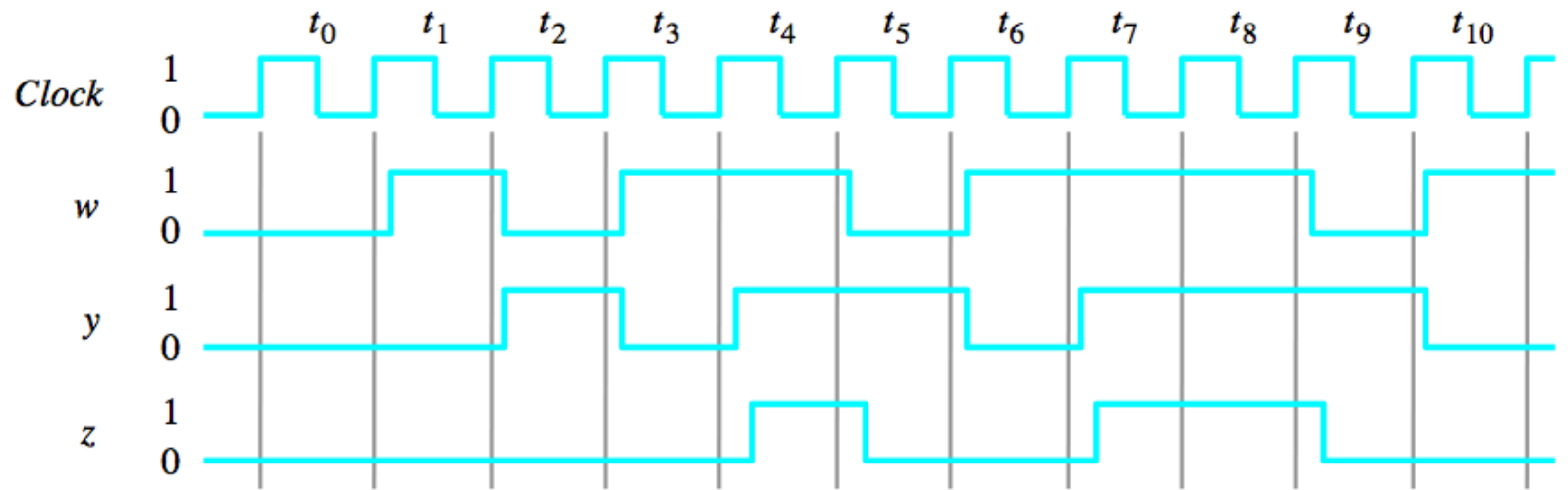
Moore



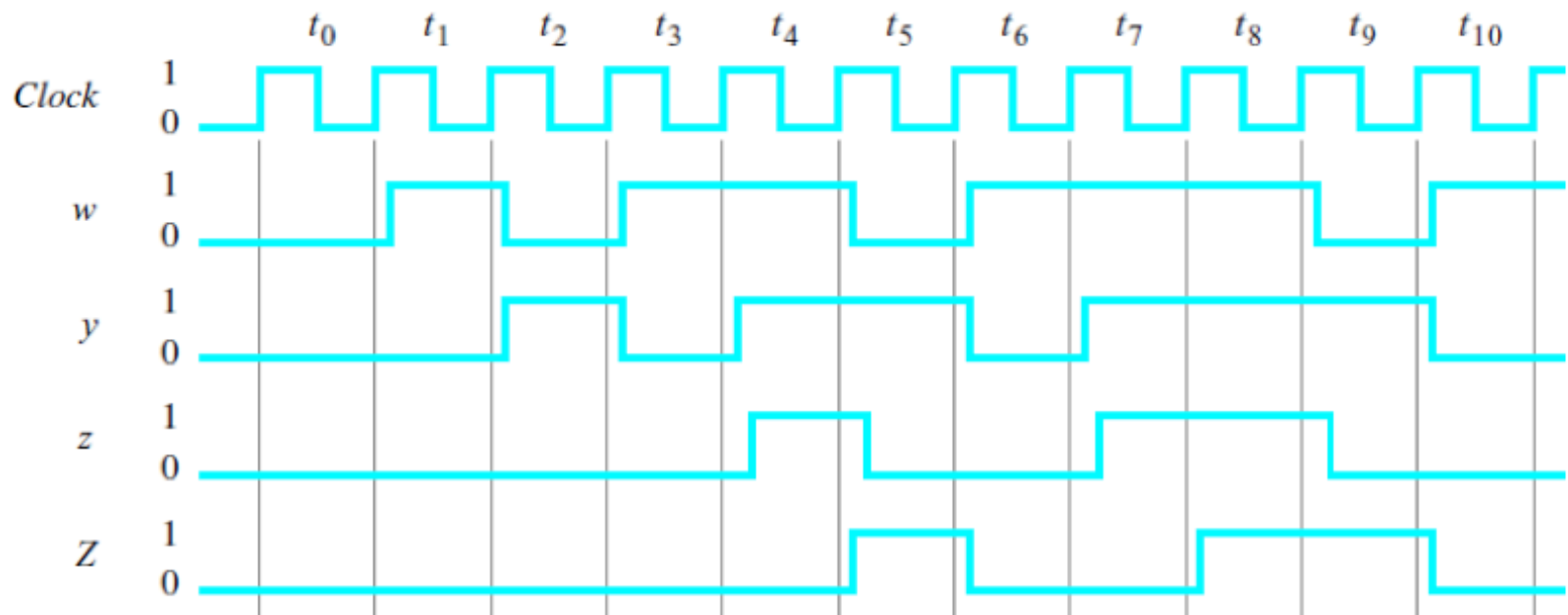
Mealy



Mealy

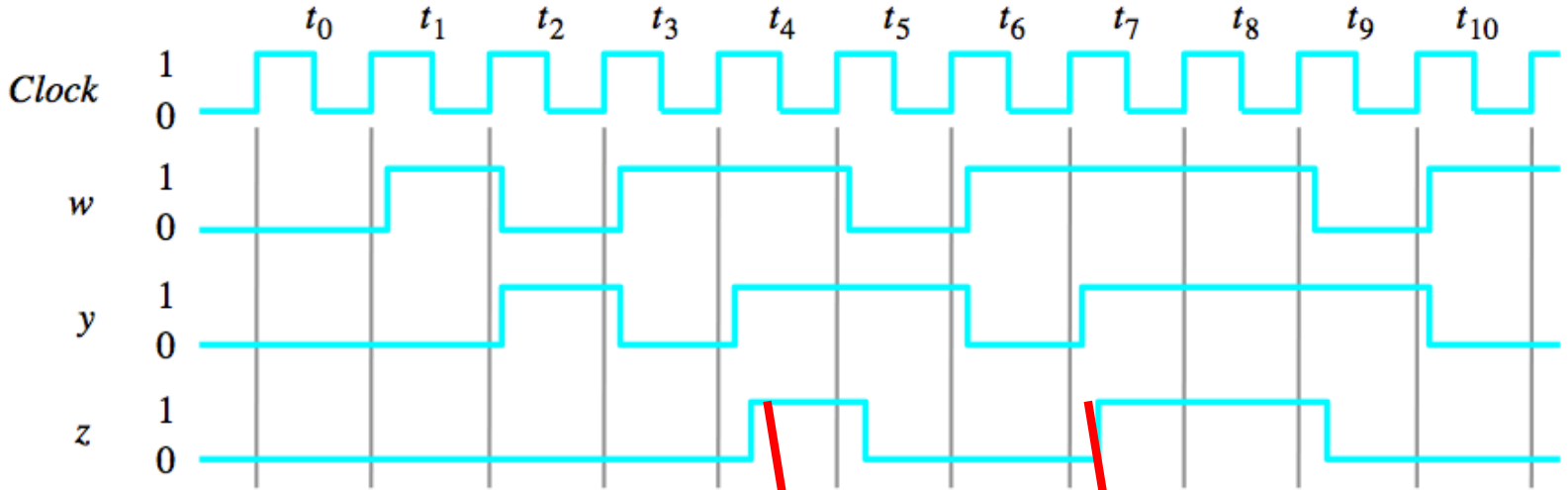


Moore

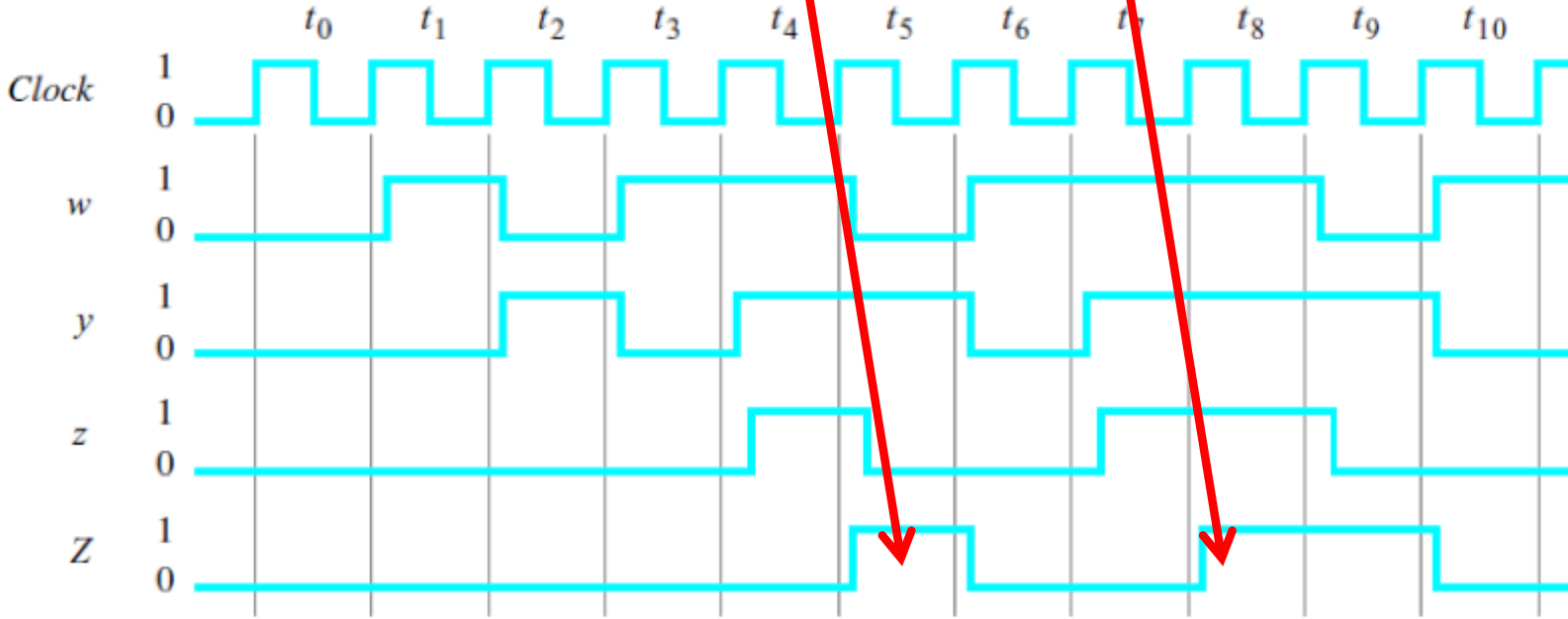


Notice that the output of the Moore machine is delayed by one clock cycle

Mealy



Moore

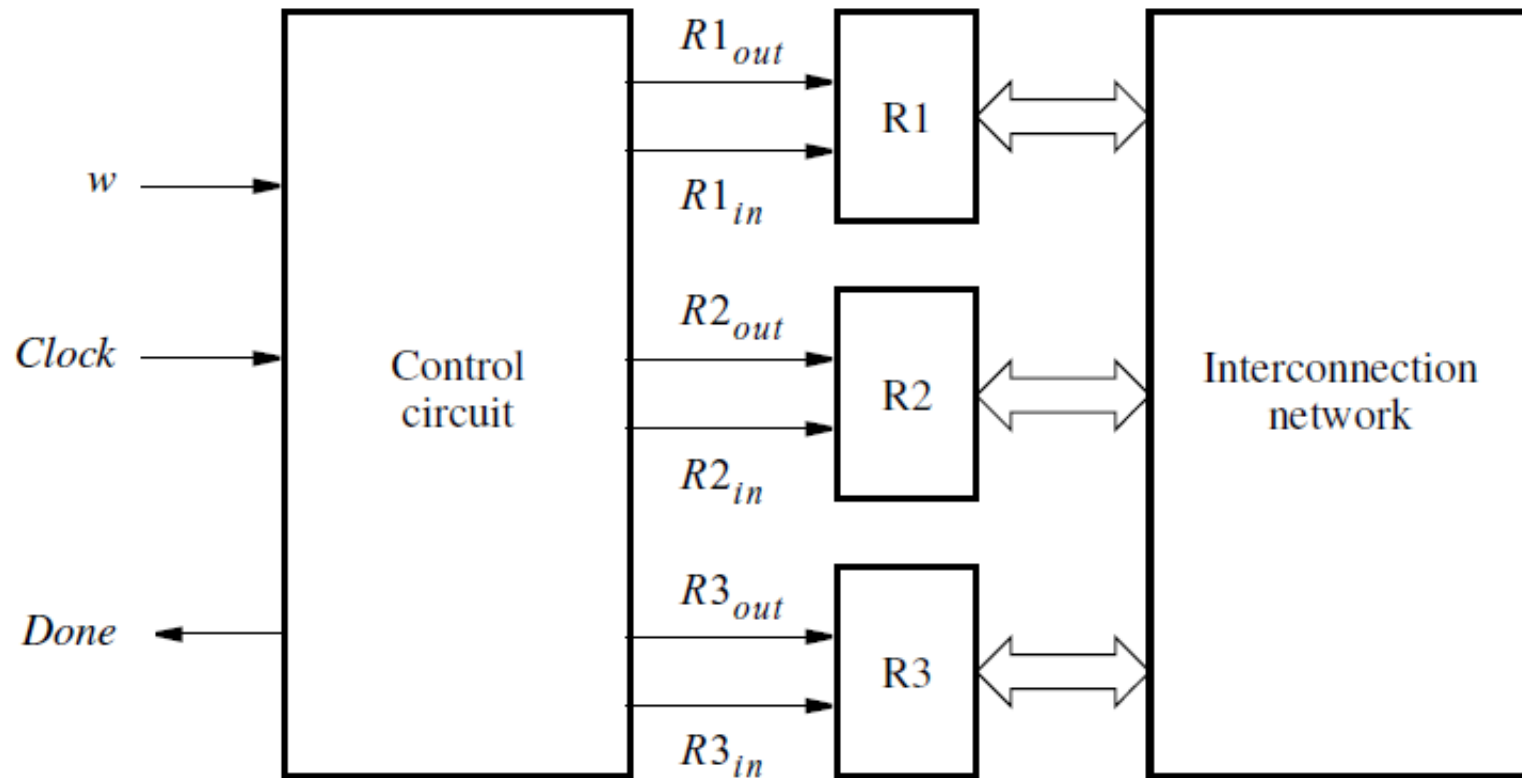


Questions?

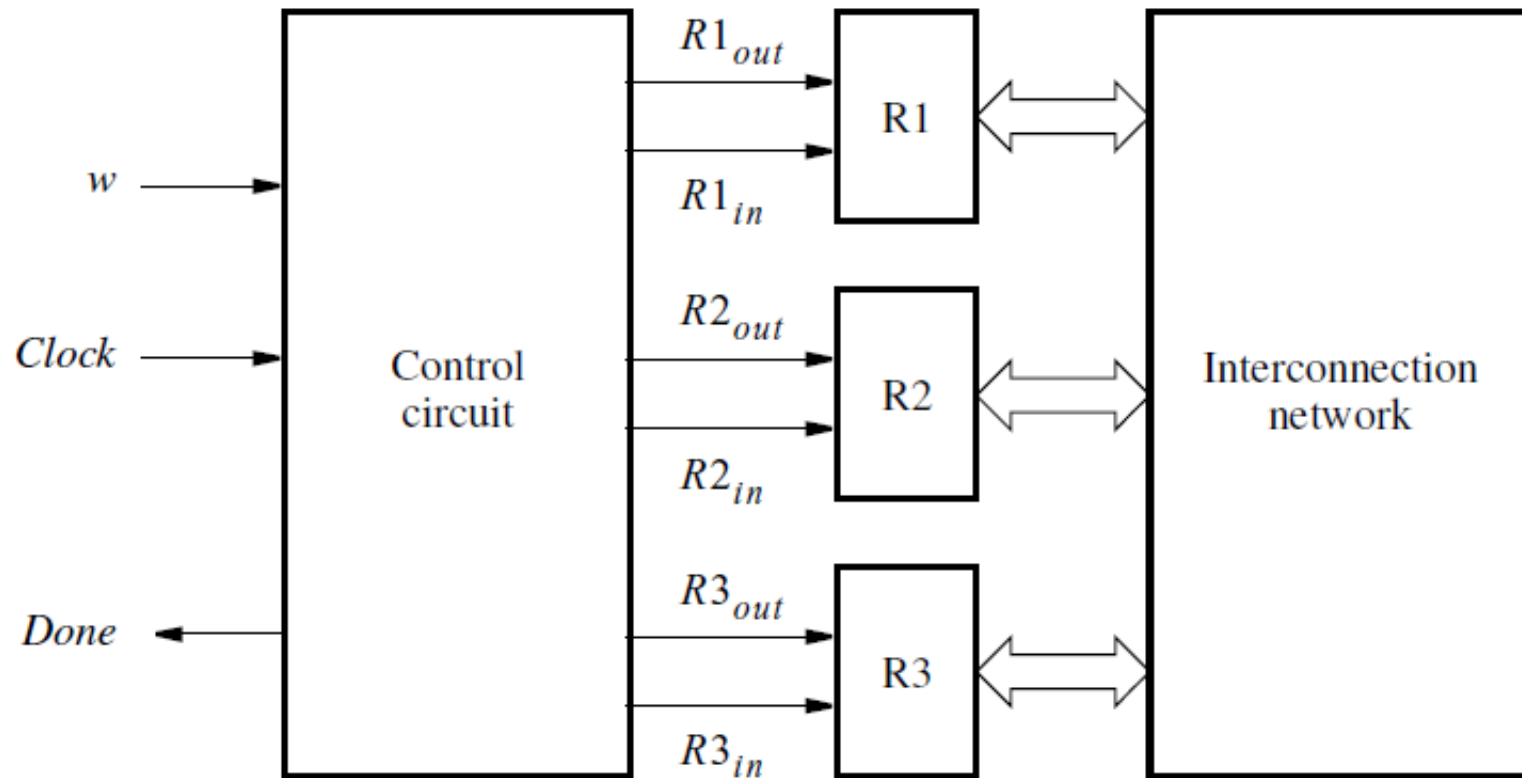
**More slides for
the State Assignment Problem**

Example #2

Register Swap Controller

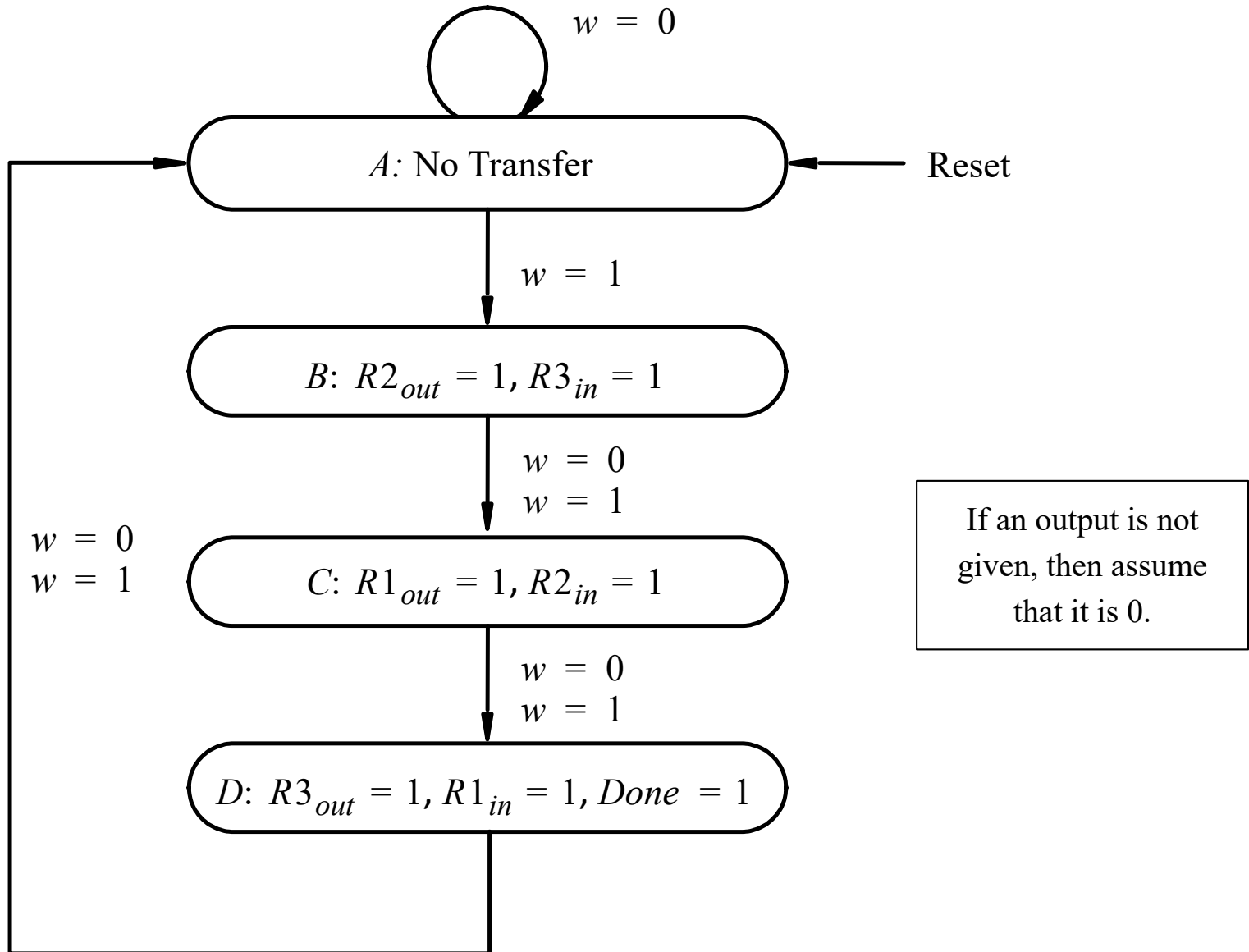


Register Swap Controller



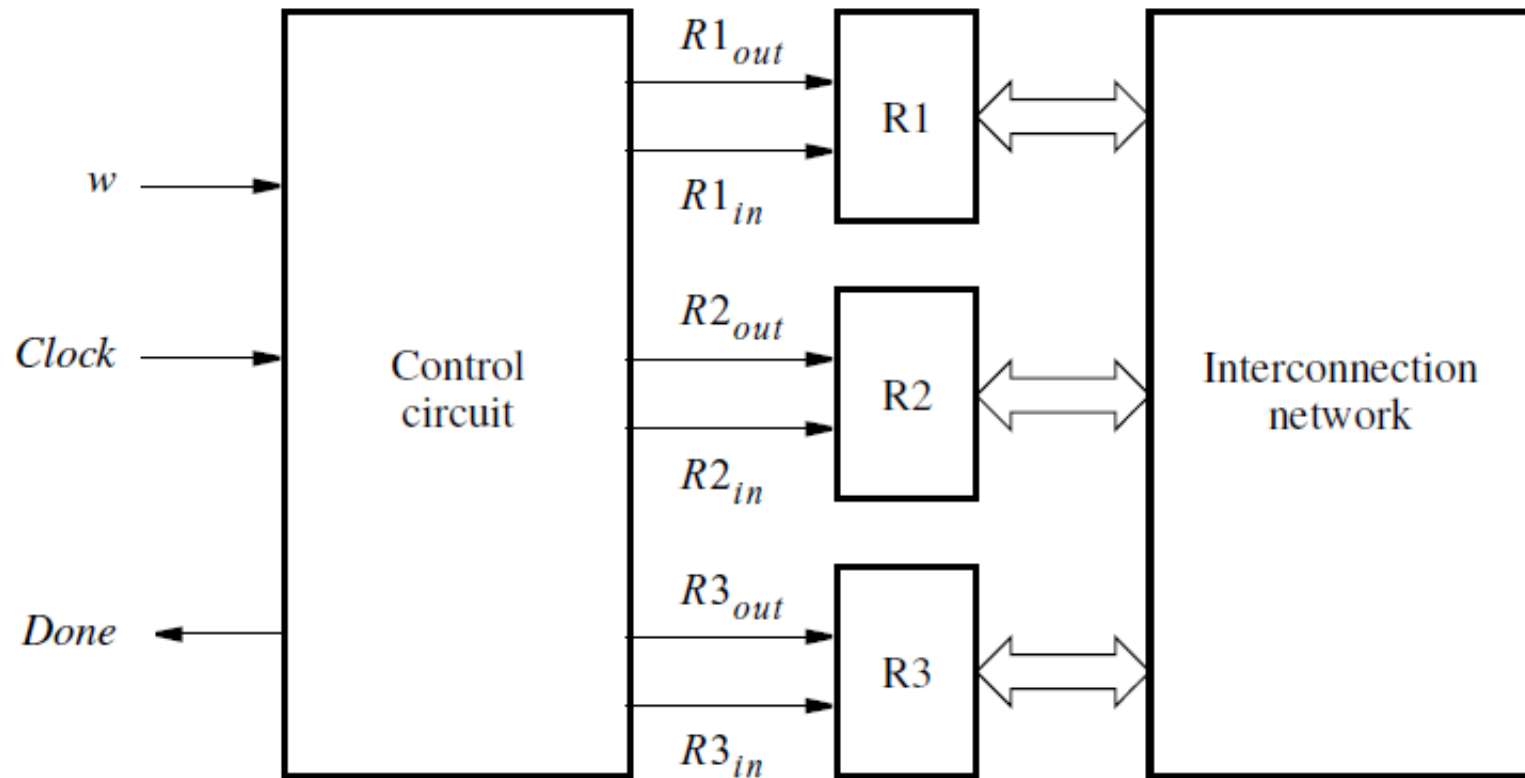
Design a Moore machine control circuit for swapping the contents of registers R1 and R2 by using R3 as a temporary.

State Diagram

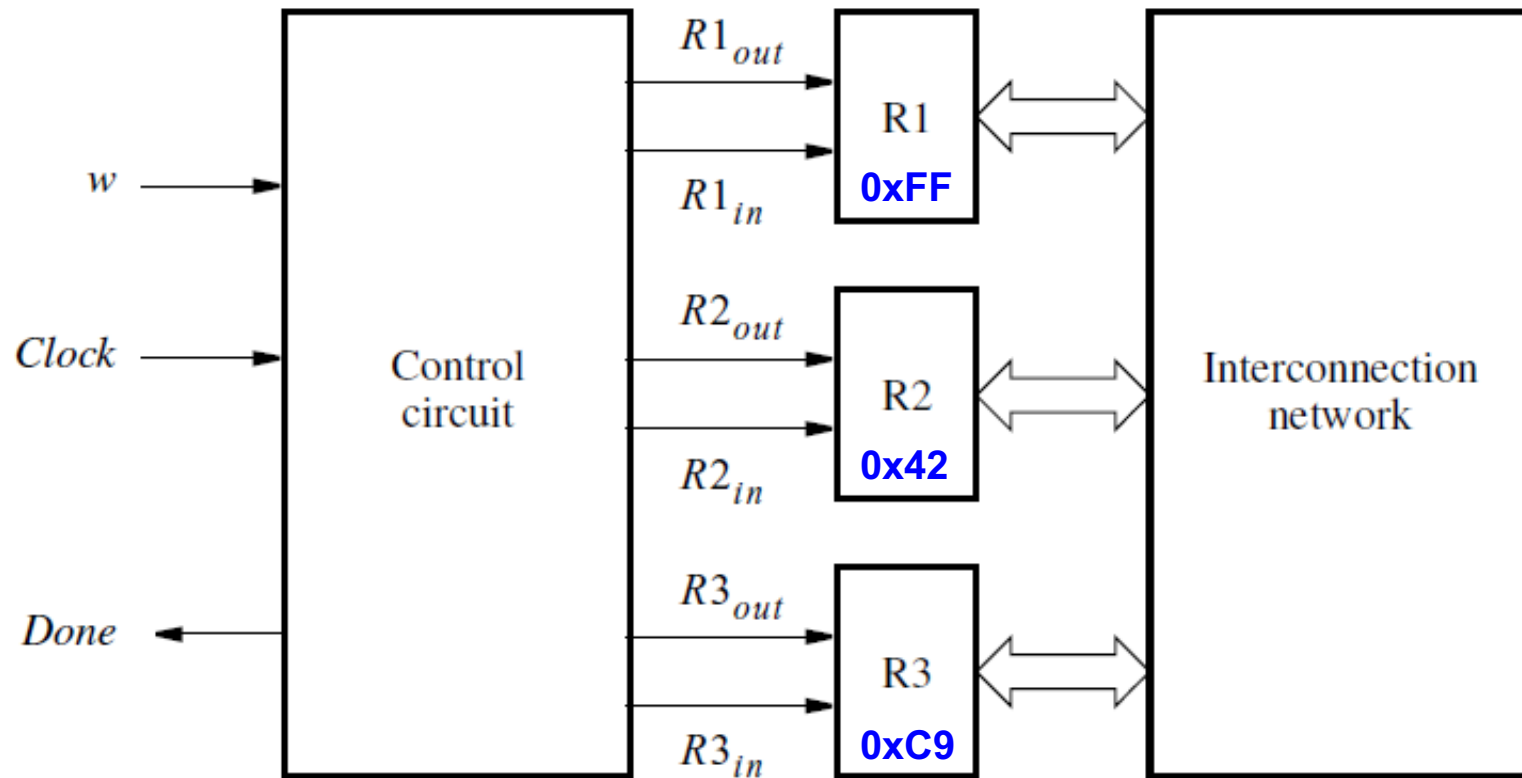


[Figure 6.11 from the textbook]

Animated Register Swap

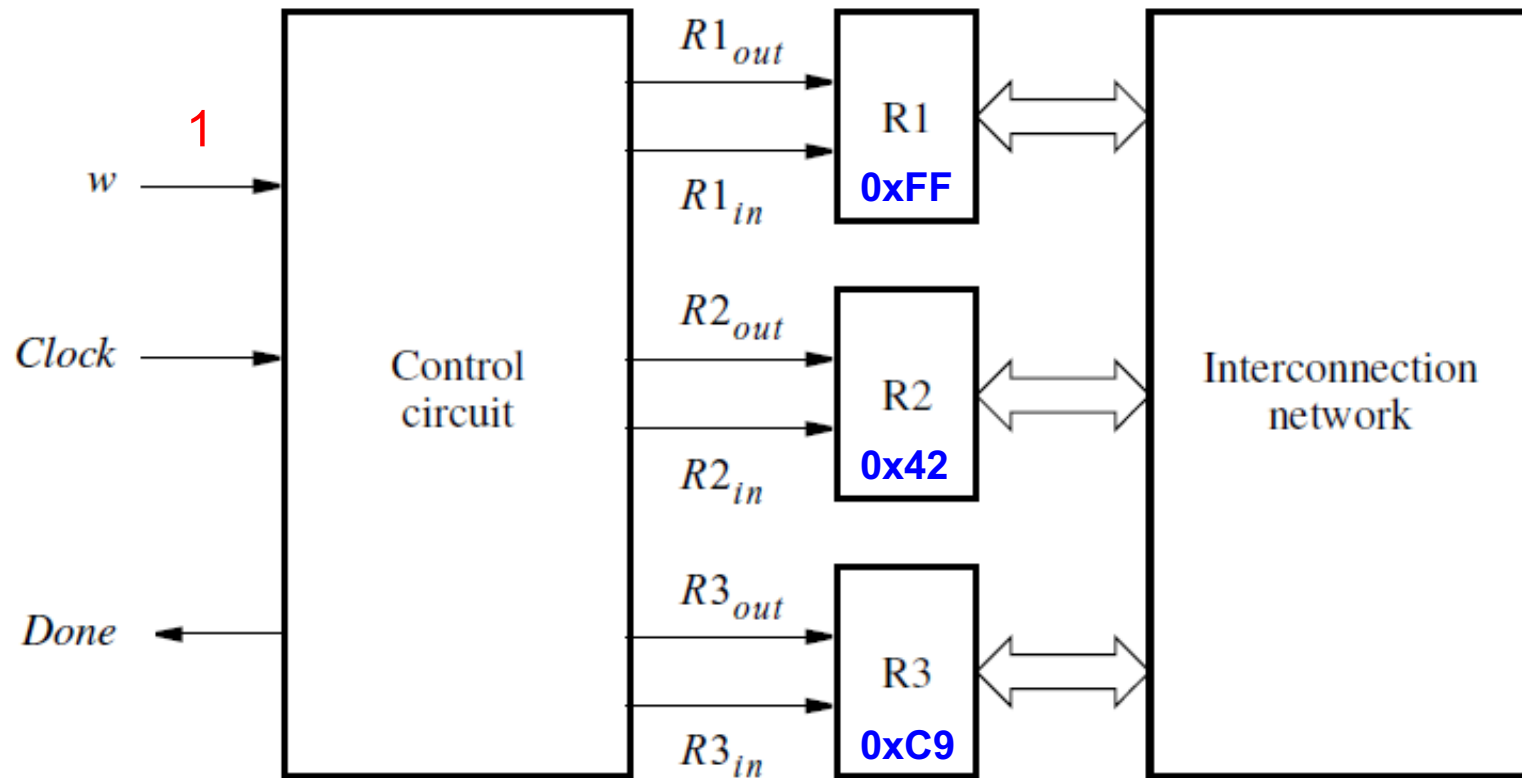


Animated Register Swap



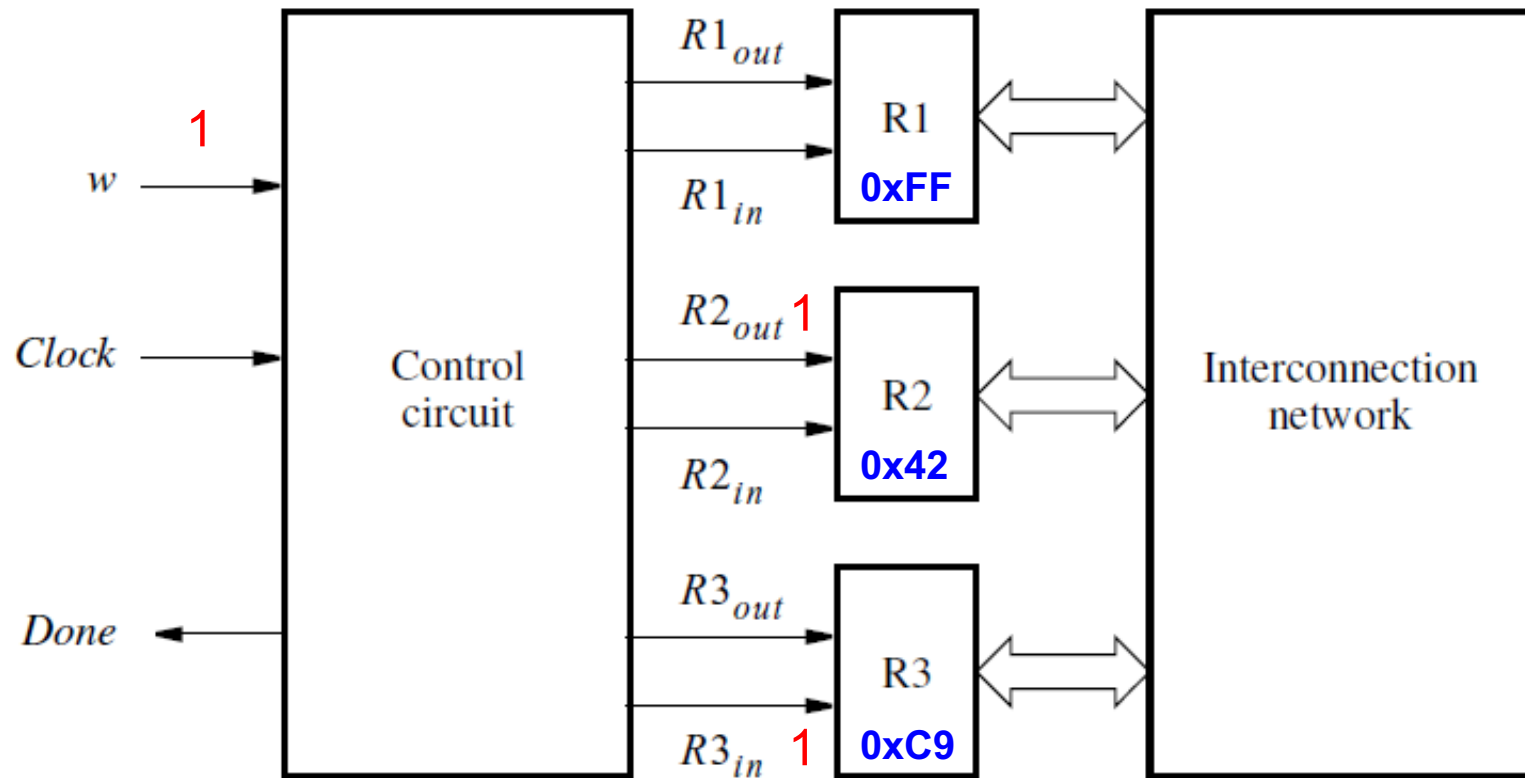
These are the original values of the 8-bit registers

Animated Register Swap

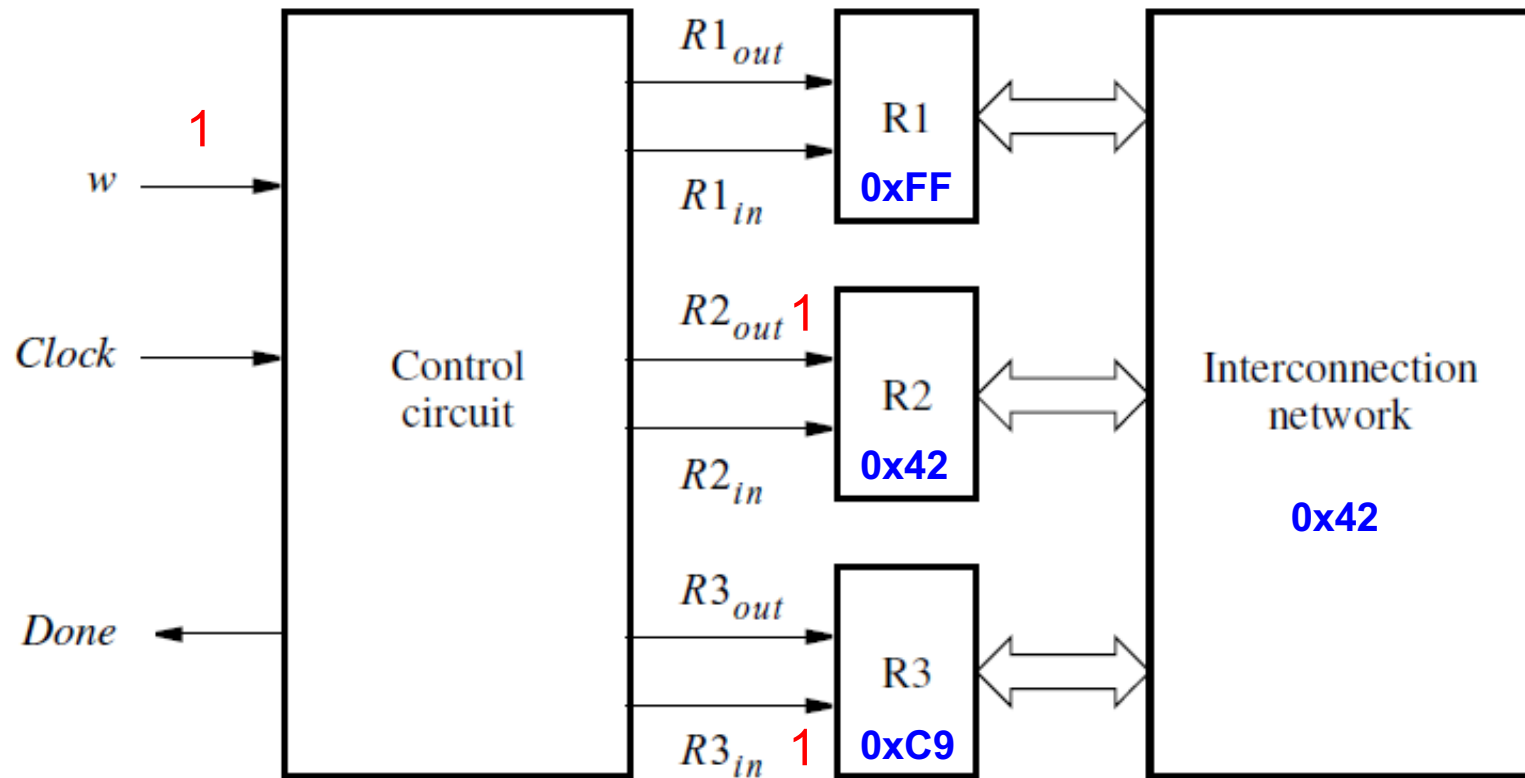


For clarity, only inputs that are equal to 1 will be shown.

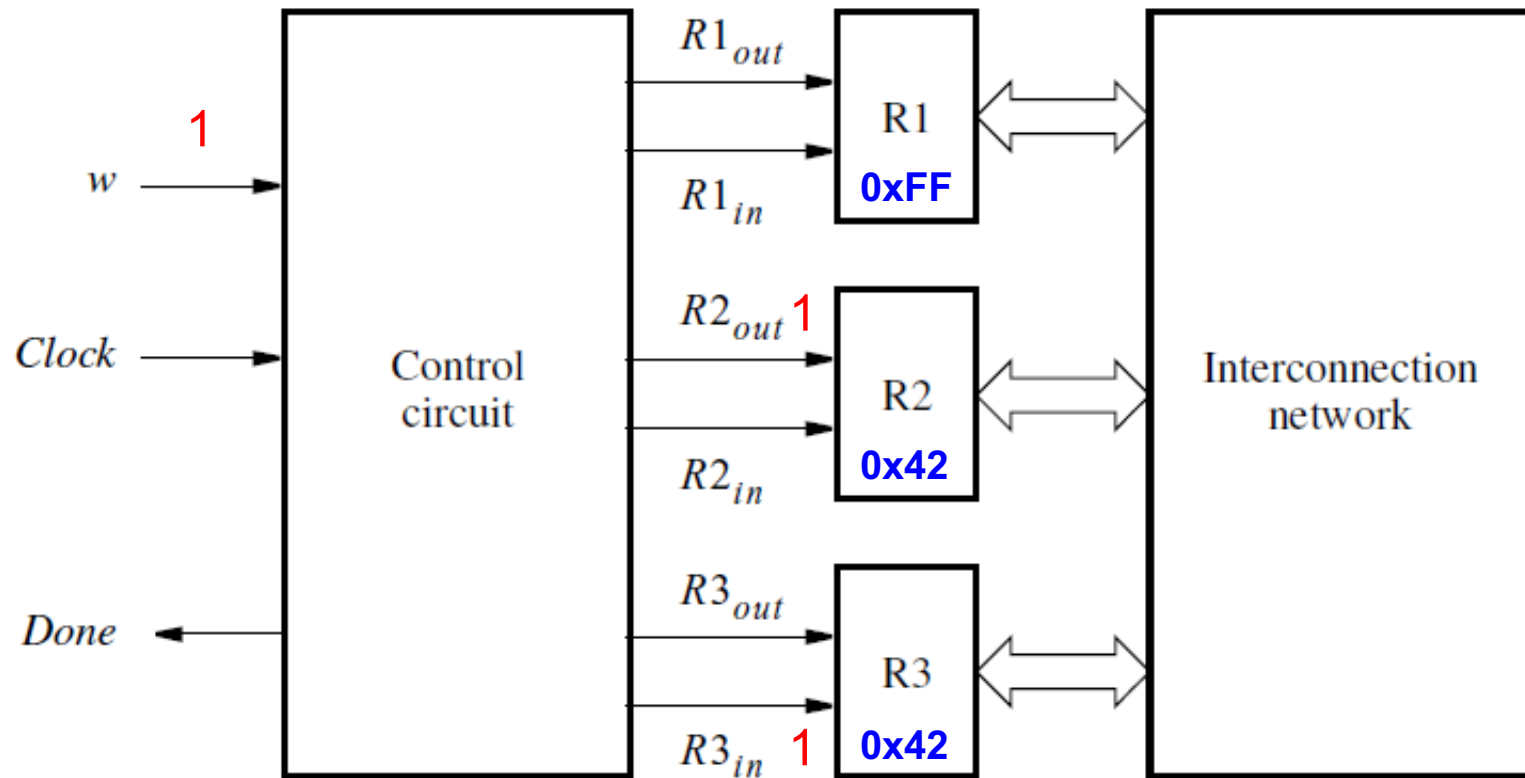
Animated Register Swap



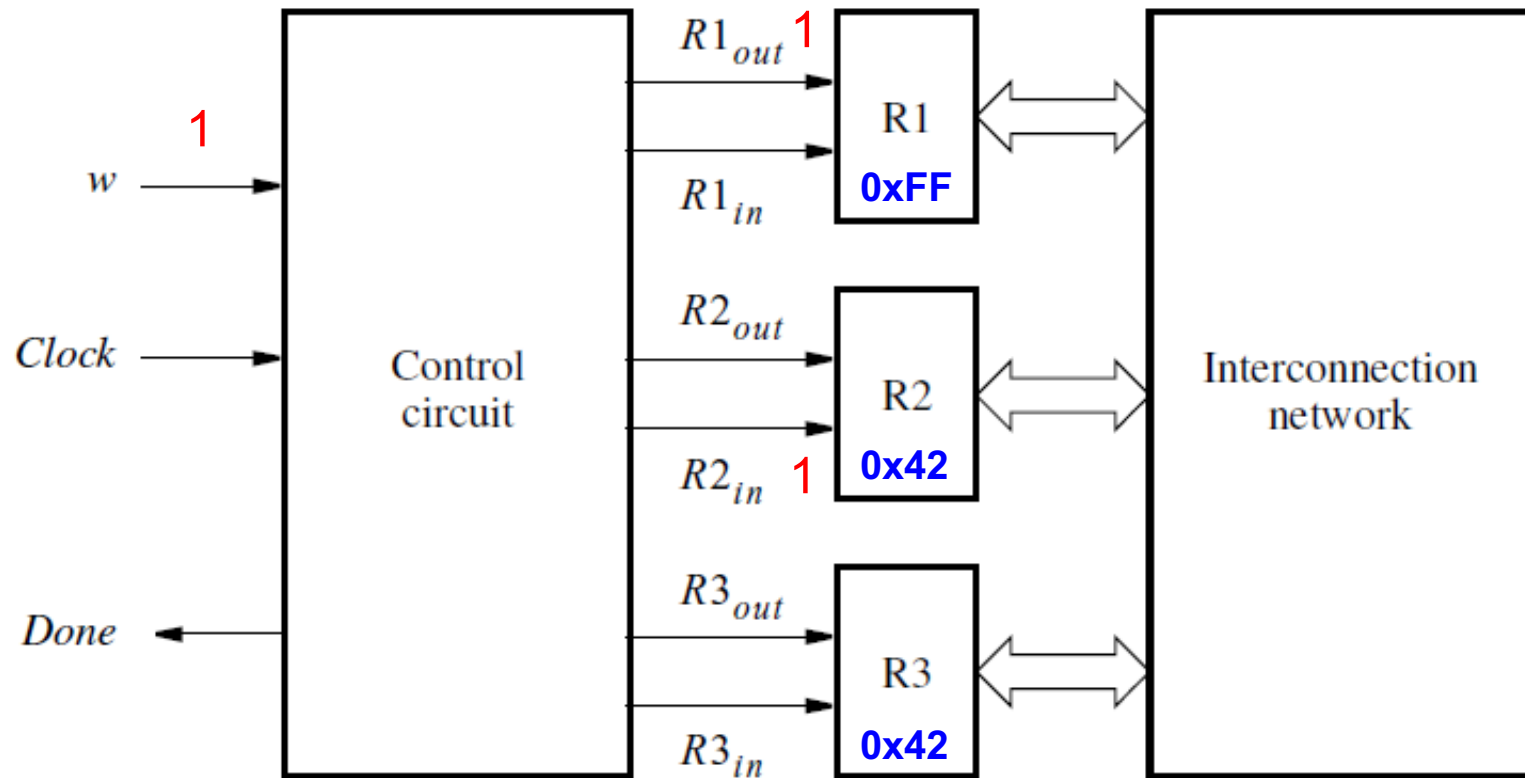
Animated Register Swap



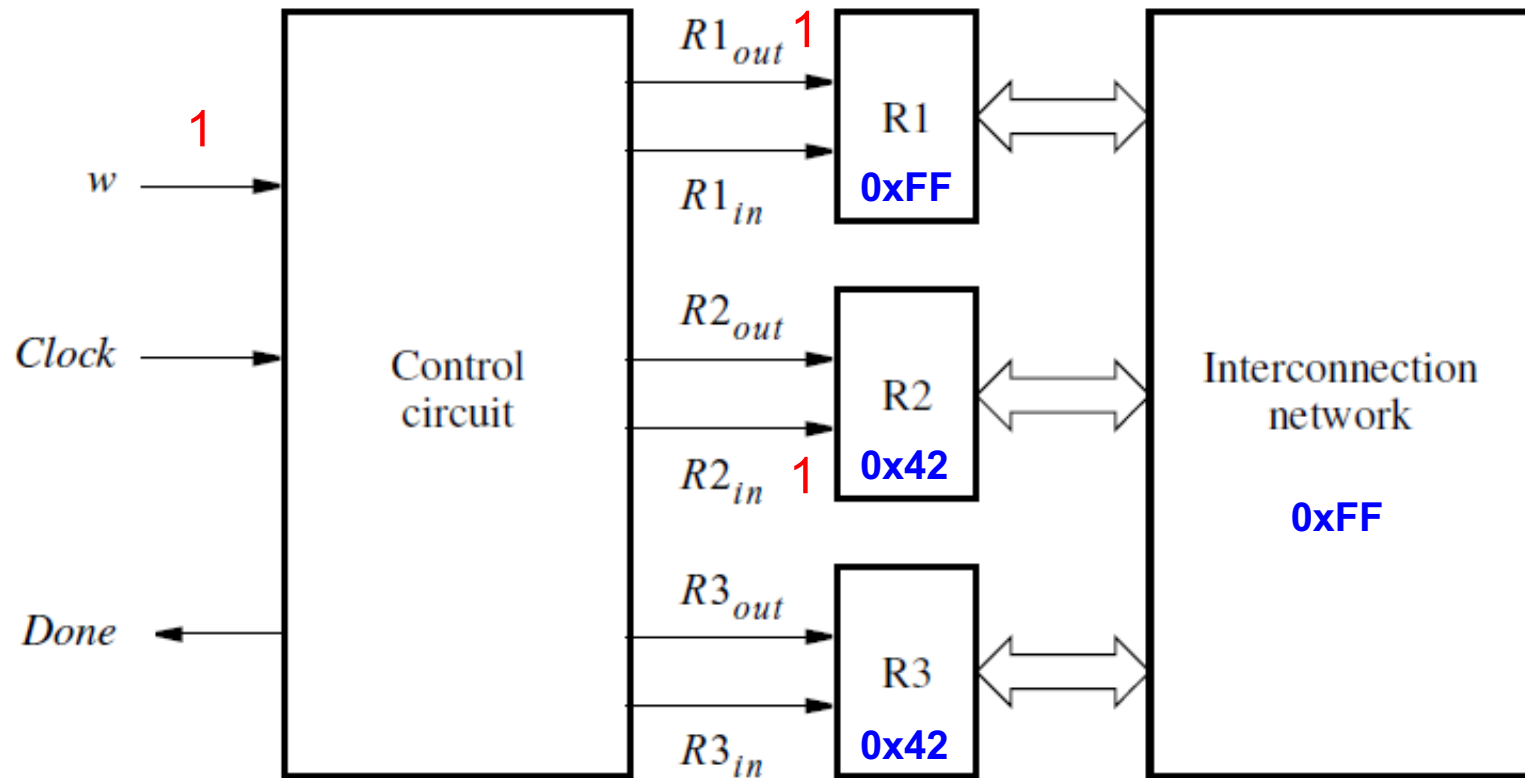
Animated Register Swap



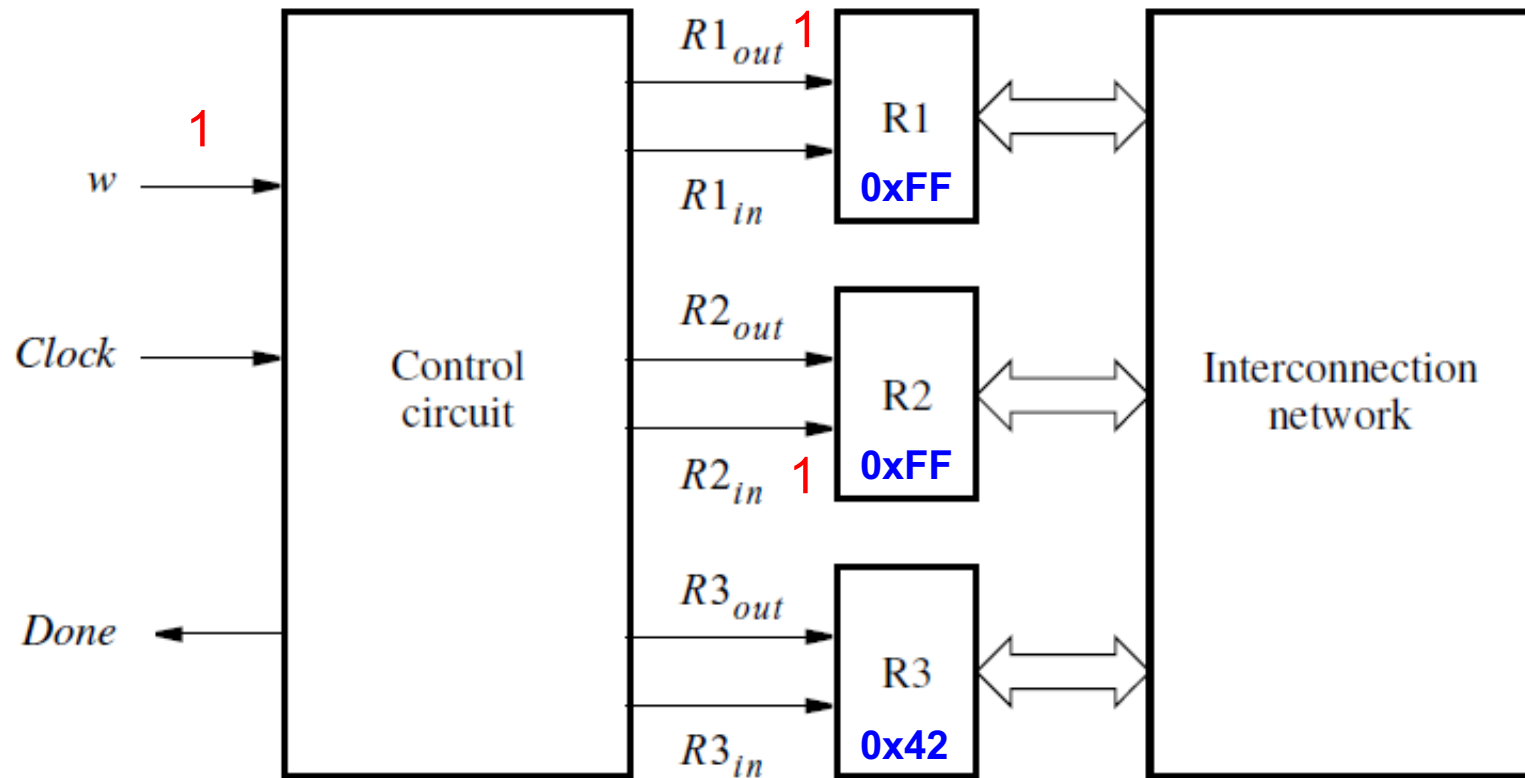
Animated Register Swap



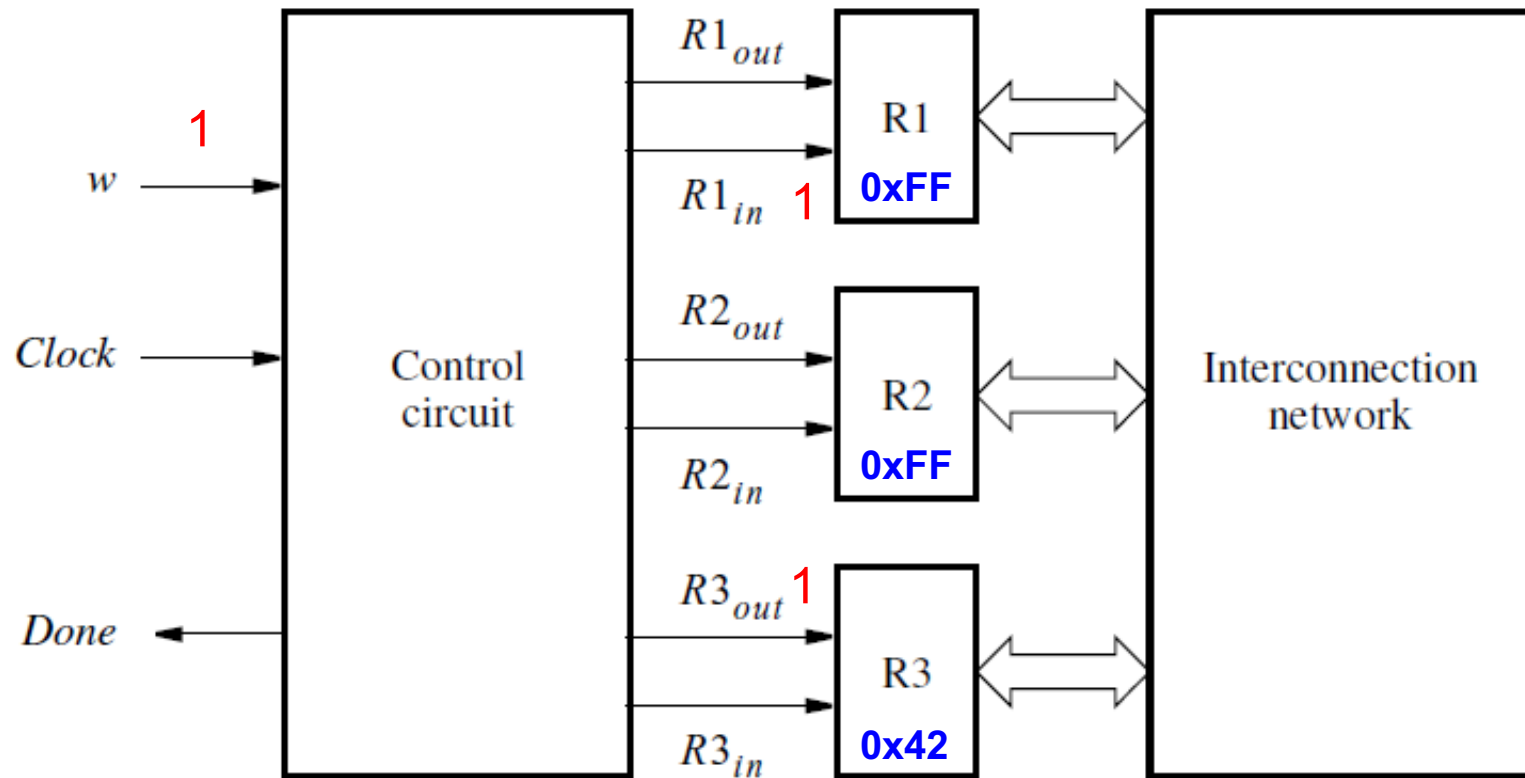
Animated Register Swap



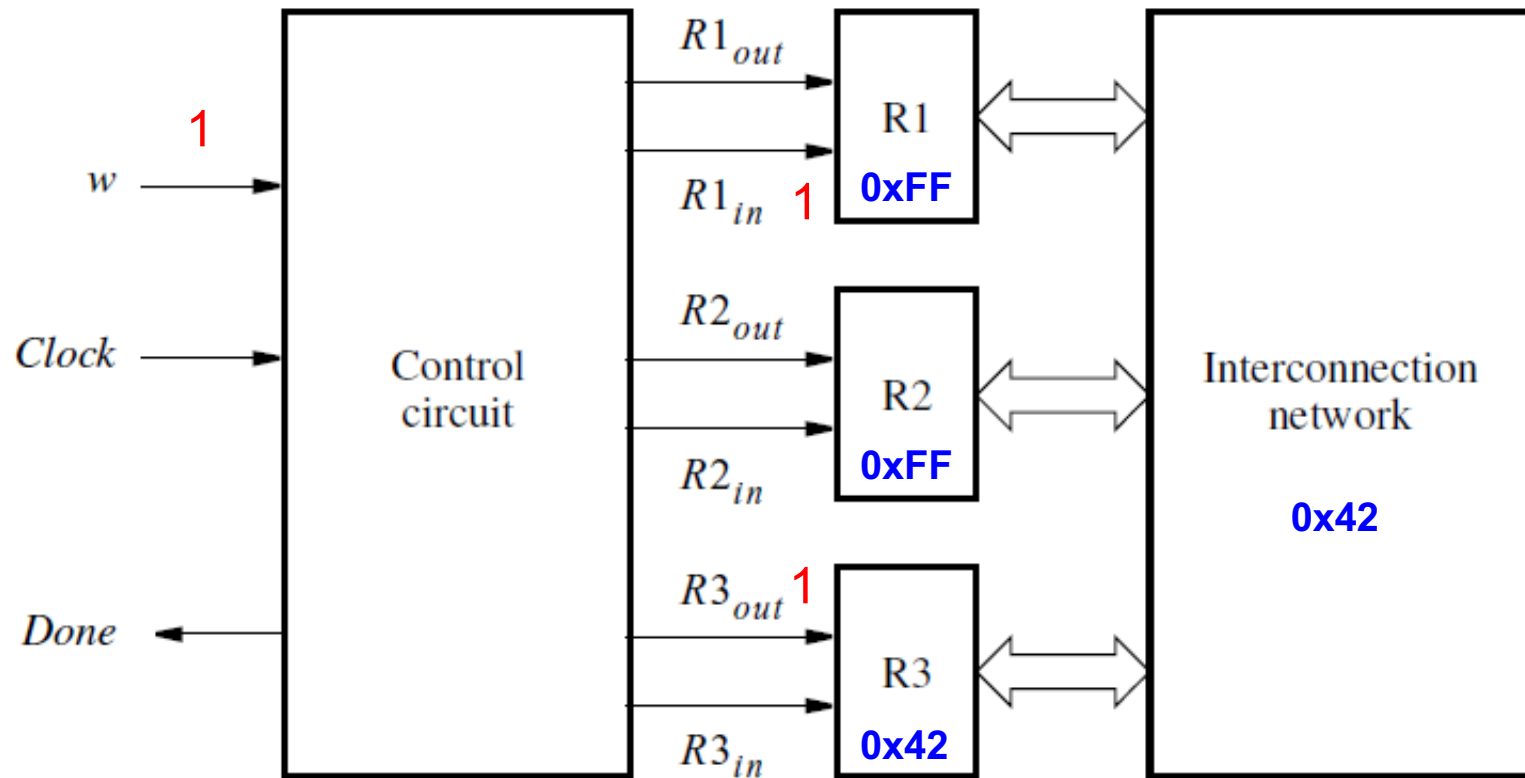
Animated Register Swap



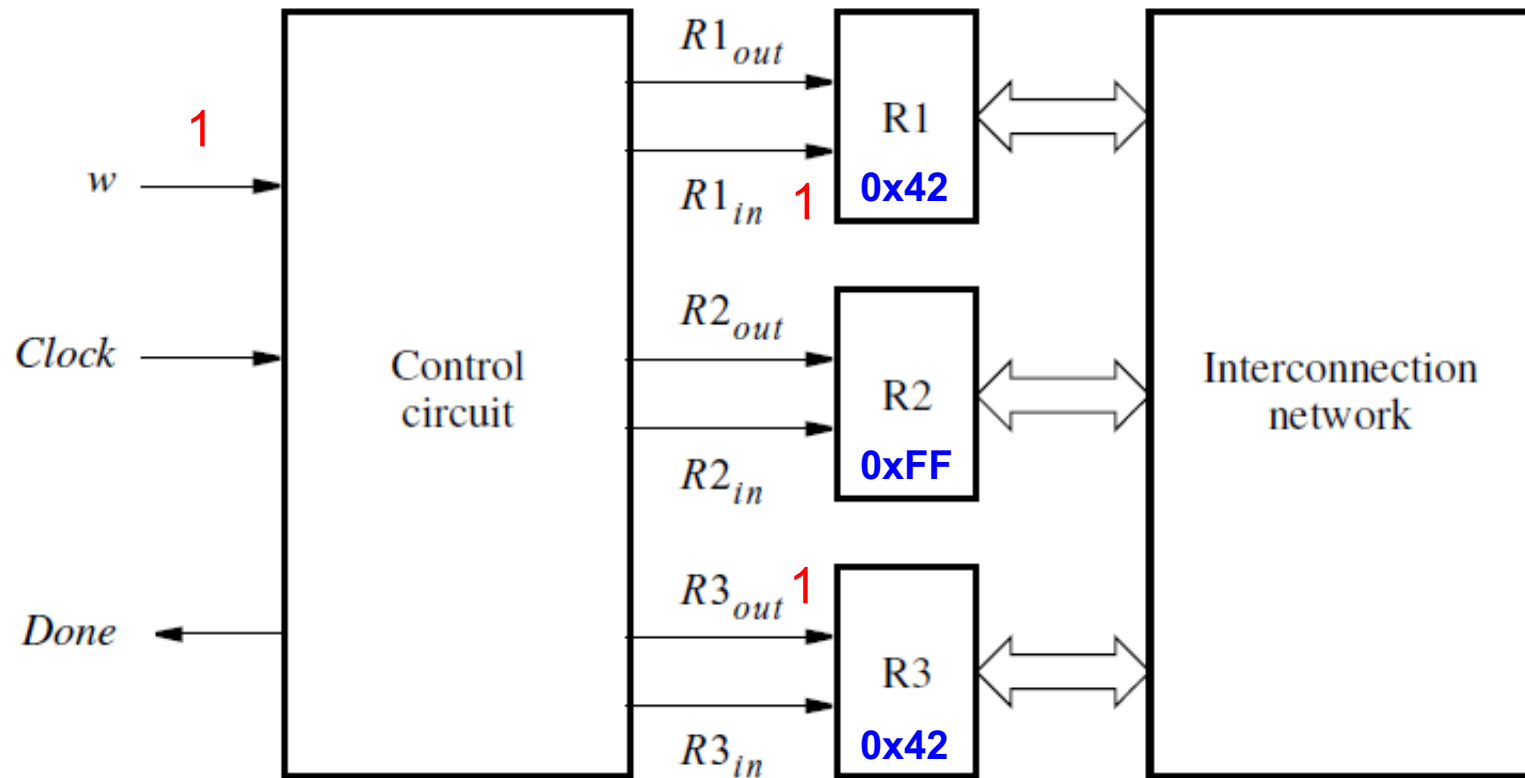
Animated Register Swap



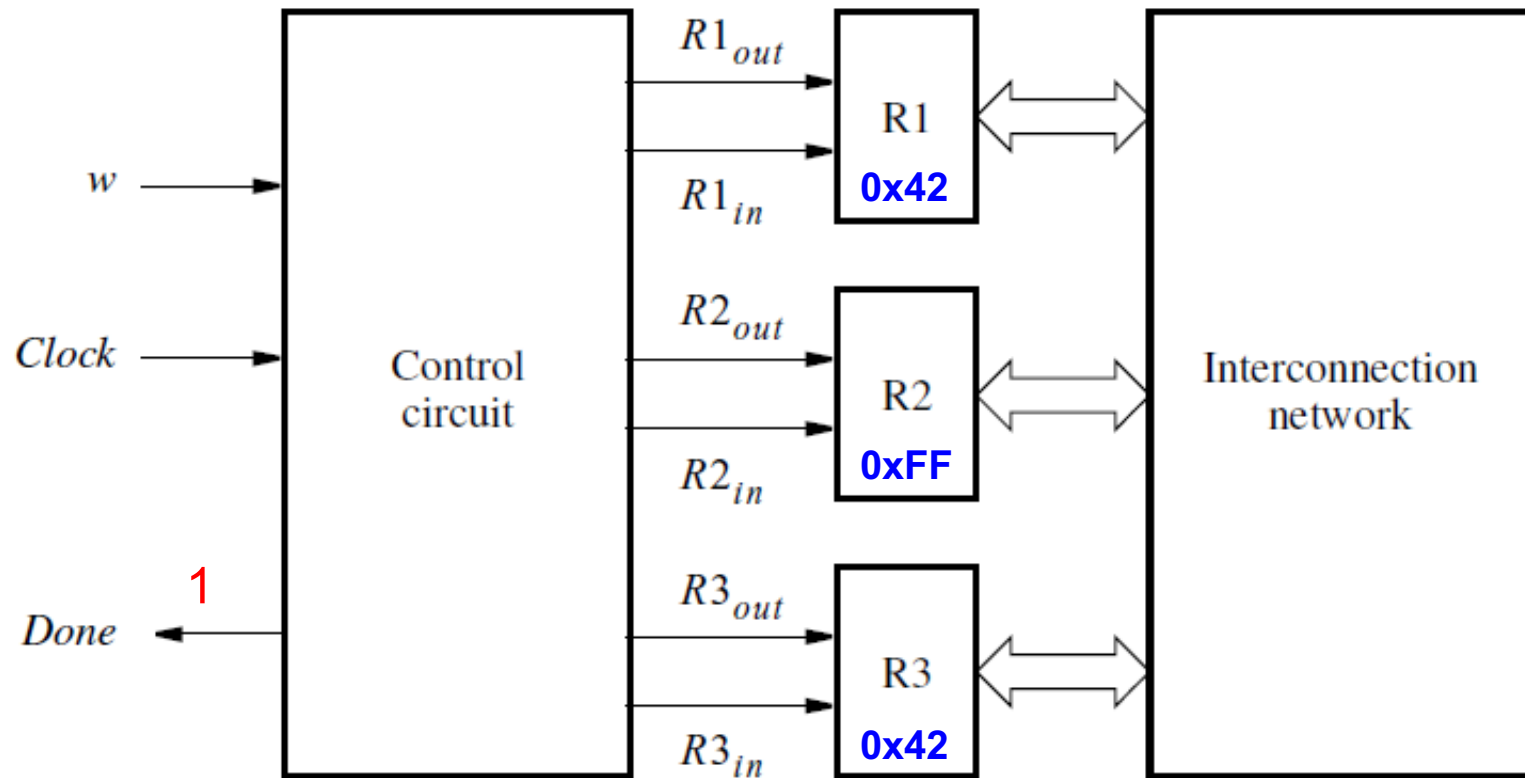
Animated Register Swap



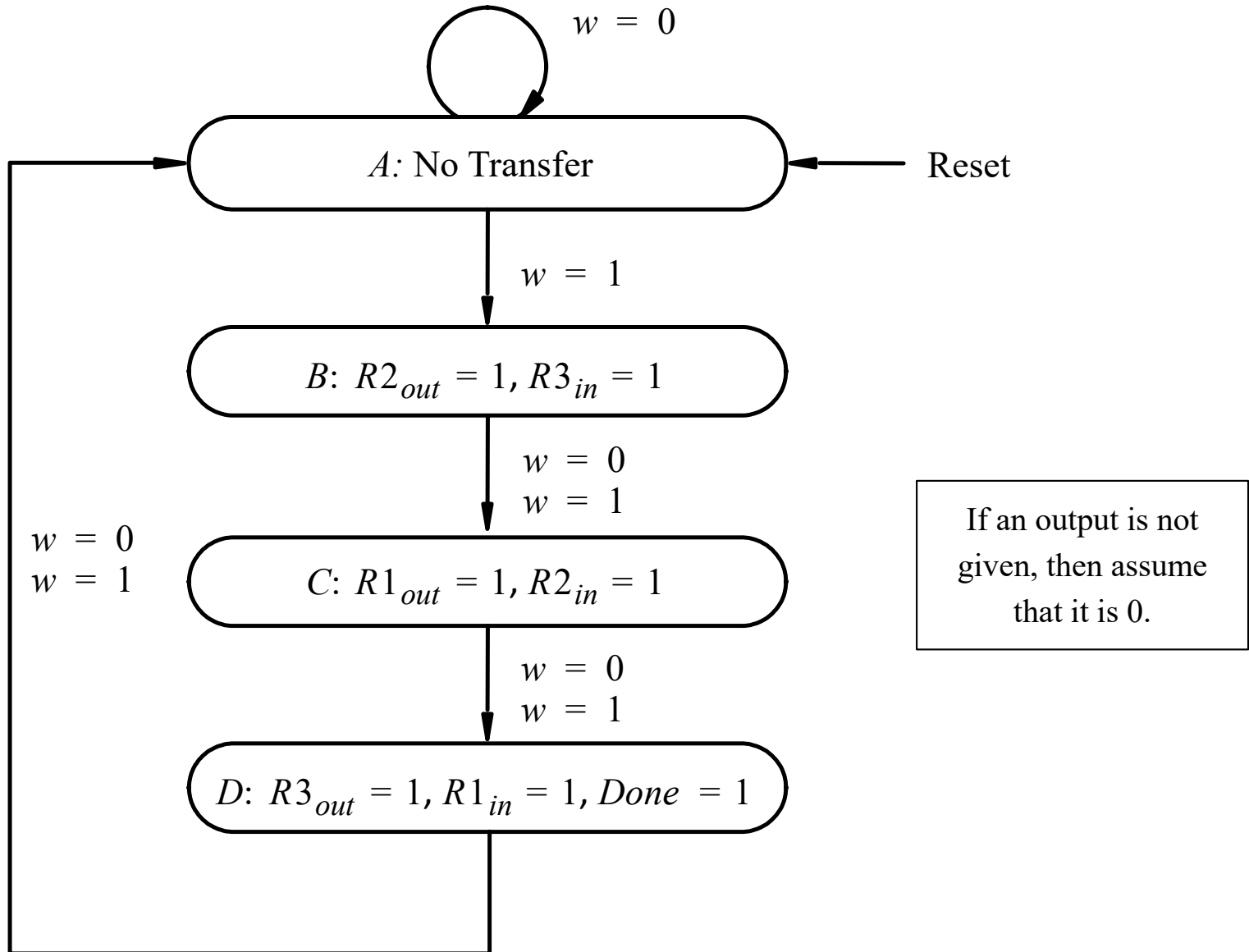
Animated Register Swap



Animated Register Swap



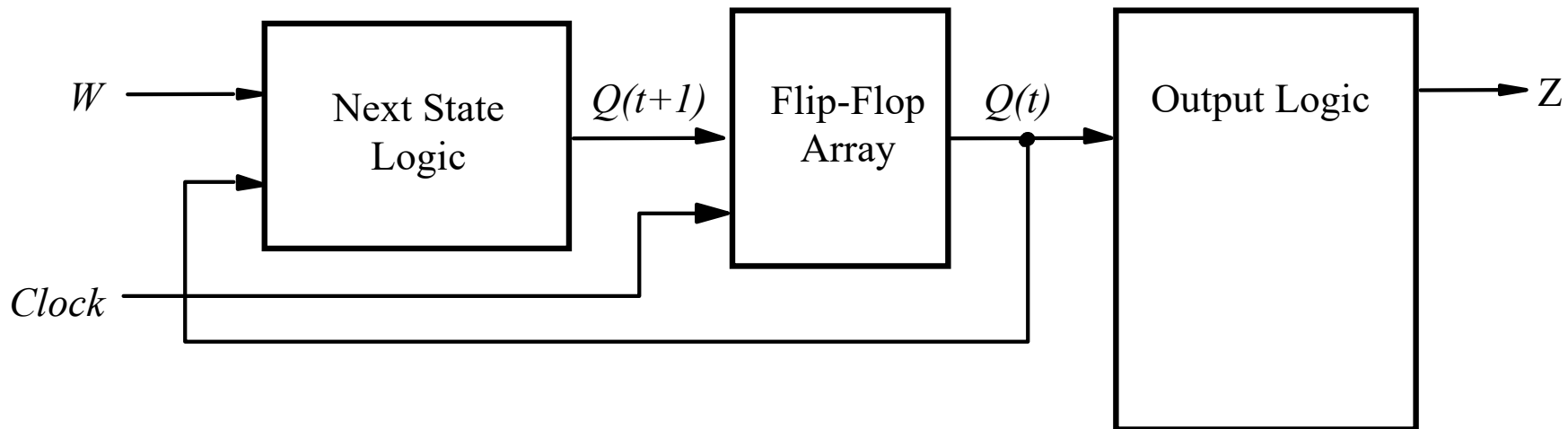
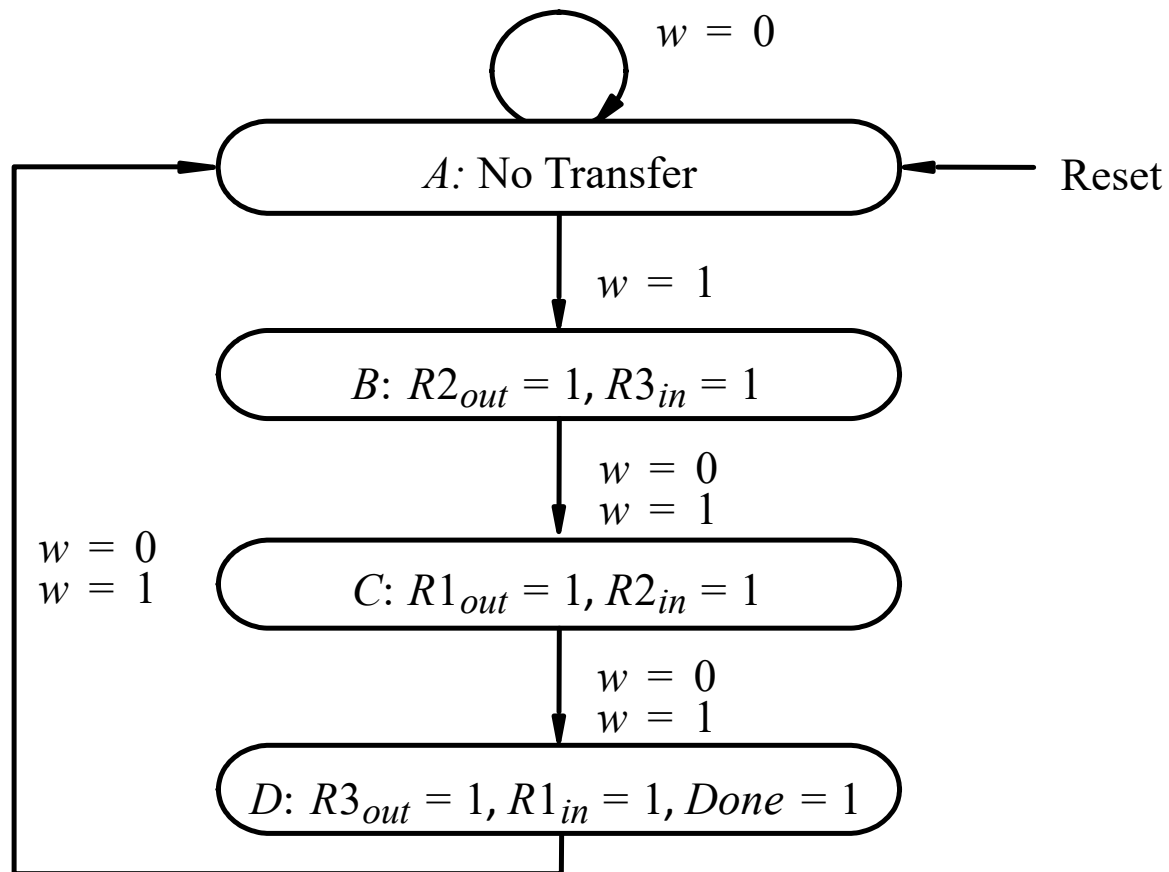
State Diagram

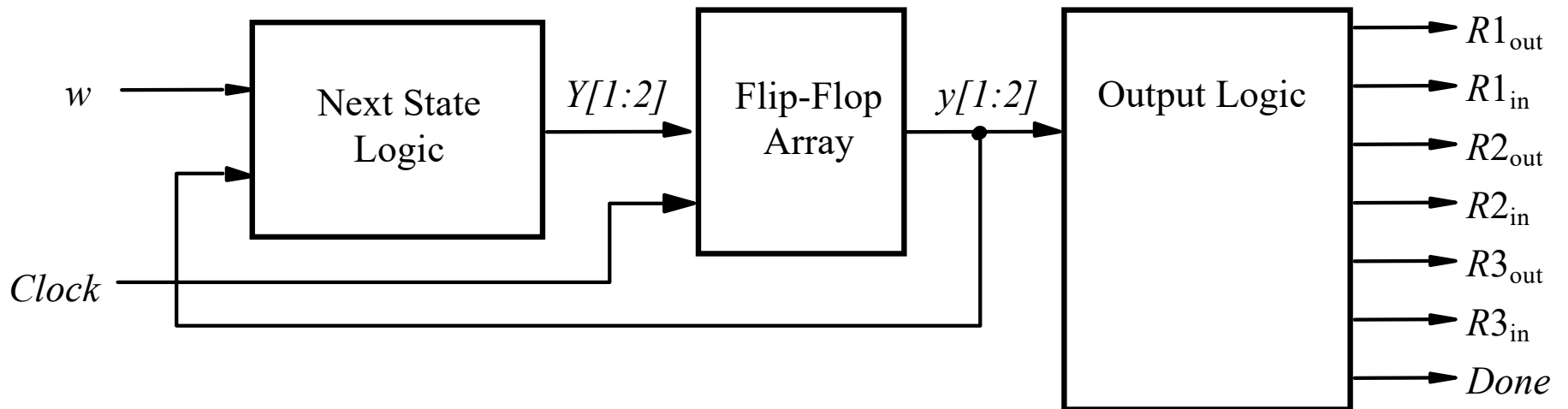
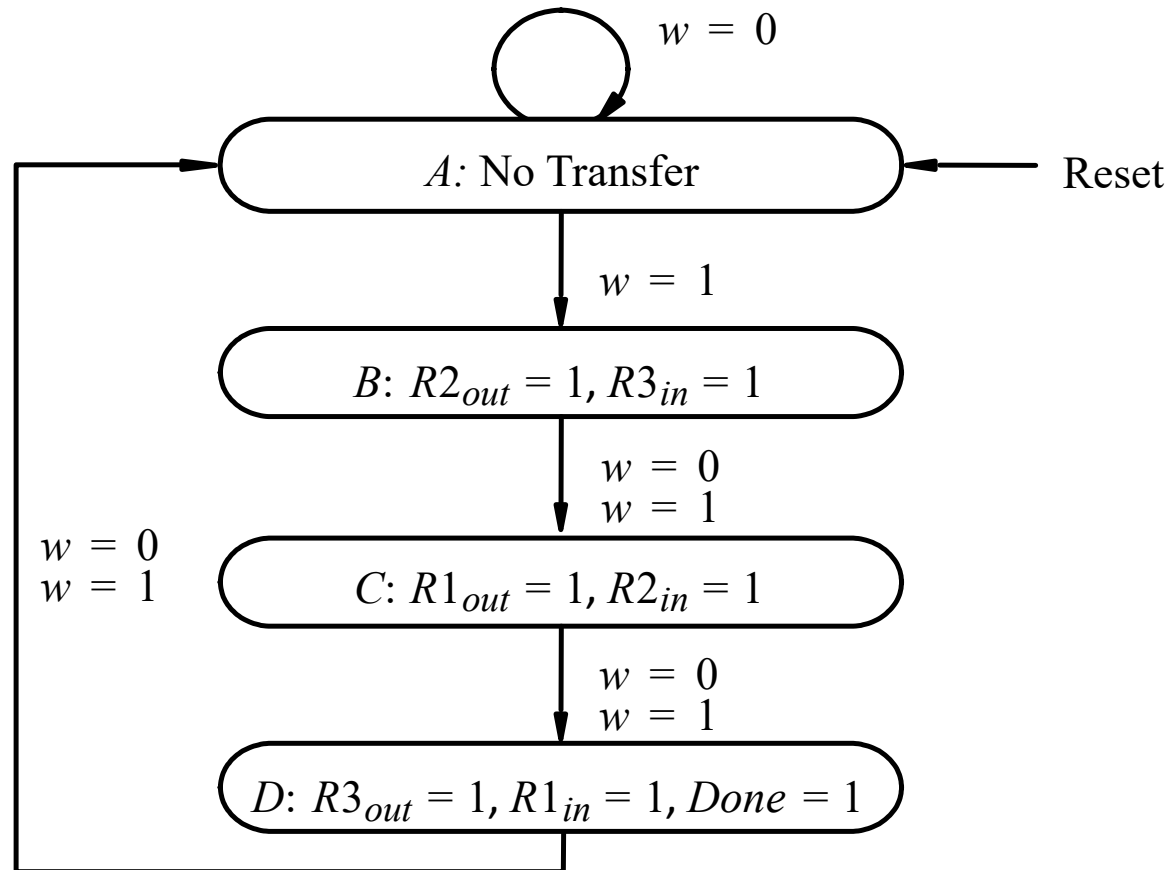


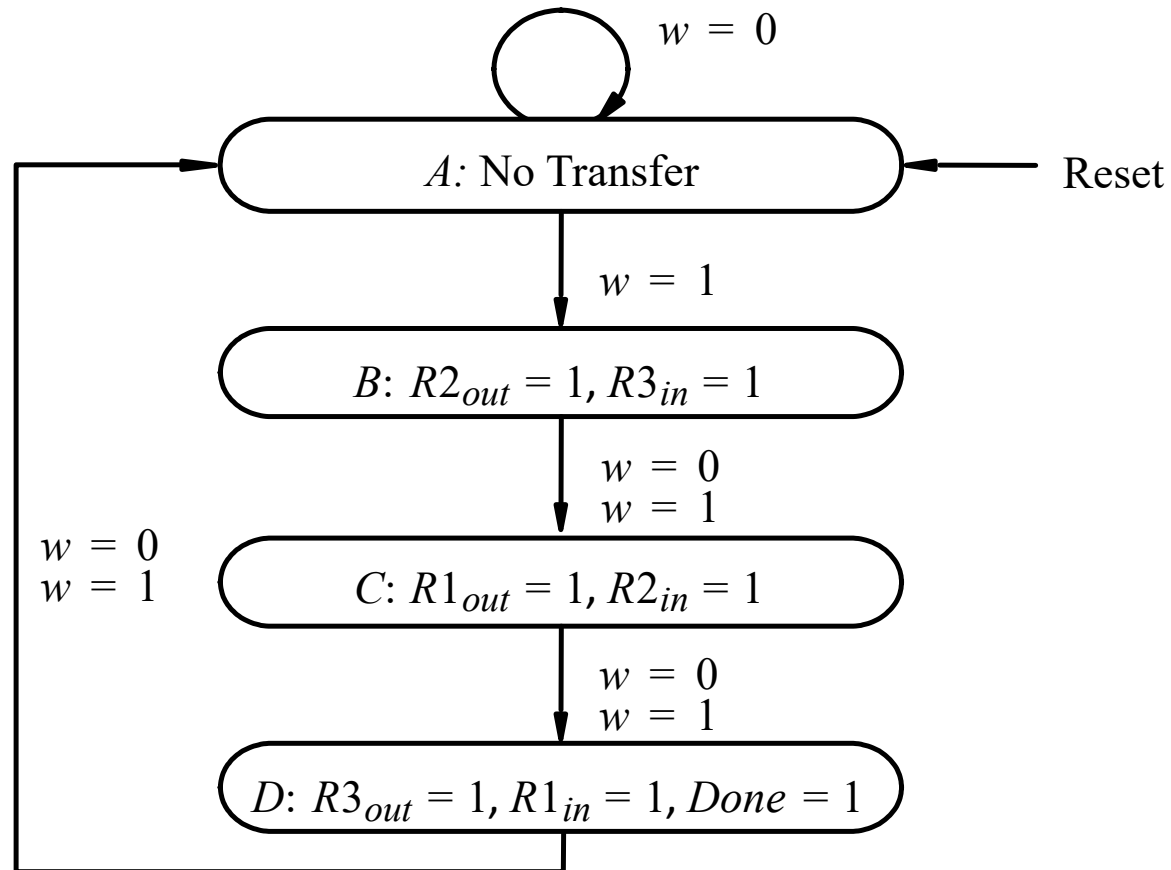
[Figure 6.11 from the textbook]

Some Questions

- **How many flip-flops are we going to use?**
- **How many logic expressions do we need to find?**







Present state	Next state		Outputs						
	$w = 0$	$w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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		Outputs						
	$w = 0$	$w = 1$	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
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_2y_1	Y_2Y_1	Y_2Y_1	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
A										
B										
C										
D										

State Table

Present state	Next state		Outputs						
	$w = 0$	$w = 1$	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
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_2y_1	Y_2Y_1	Y_2Y_1	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
A	00									
B	01									
C	10									
D	11									

State Table

Present state	Next state		Outputs						
	$w = 0$	$w = 1$	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
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_2y_1	Y_2Y_1	Y_2Y_1	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
A	00	00	01							
B	01	10	10							
C	10	11	11							
D	11	00	00							

State Table

Present state	Next state		Outputs						
	$w = 0$	$w = 1$	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
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						
		$w = 0$	$w = 1$	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
	Y_2Y_1	Y_2Y_1								
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

	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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.

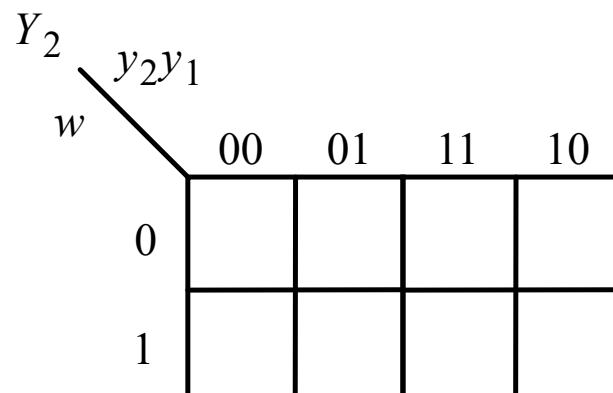
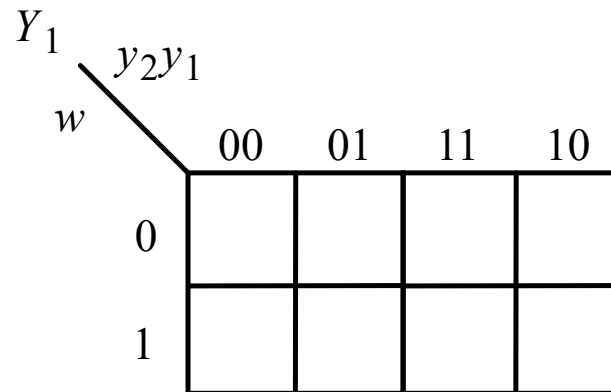
	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
A	00	00	0 1	0	0	0	0	0	0	0
B	01	10	1 0	0	0	1	0	0	1	0
C	10	11	1 1	1	0	0	1	0	0	0
D	11	00	0 0	0	1	0	0	1	0	1

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	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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



	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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

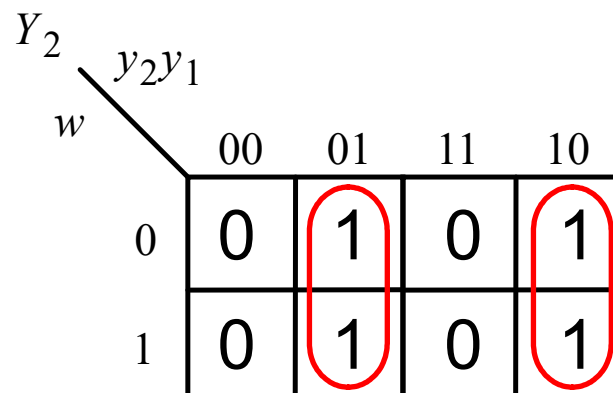
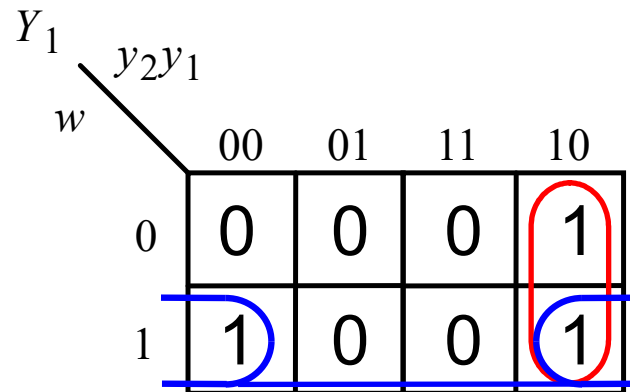
w	y_2y_1	00	01	11	10
0		0	0	0	1
1		1	0	0	1

Y_2

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

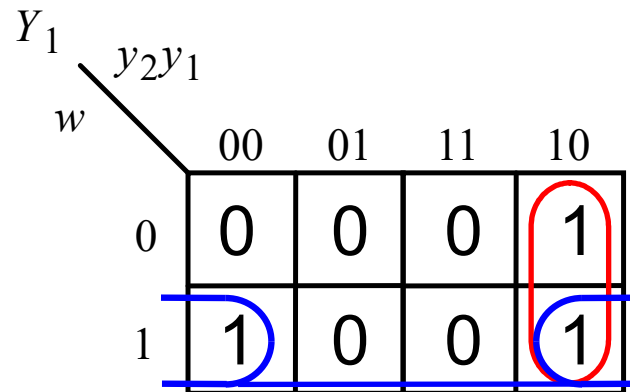
	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
A	00	00	0 1	0	0	0	0	0	0	0
B	01	10	1 0	0	0	1	0	0	1	0
C	10	11	1 1	1	0	0	1	0	0	0
D	11	00	0 0	0	1	0	0	1	0	1

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

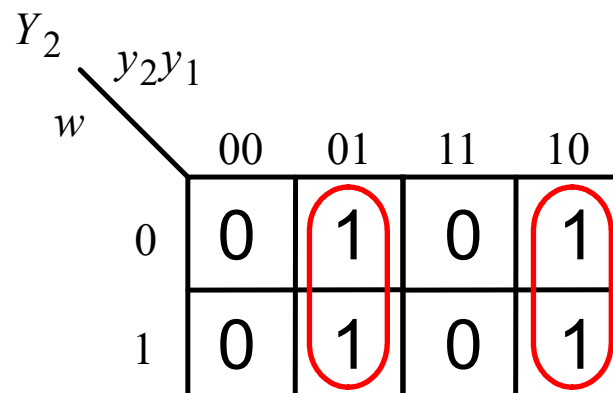


	Present state	Next state		Outputs						
		$w = 0$	$w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
	y_2y_1	Y_2Y_1	Y_2Y_1							
A	00	00	01	0	0	0	0	0	0	0
B	01	10	10	0	0	1	0	0	1	0
C	10	11	11	1	0	0	1	0	0	0
D	11	00	00	0	1	0	0	1	0	1

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



$$Y_1 = w\bar{y}_1 + \bar{y}_1y_2$$



$$Y_2 = y_1\bar{y}_2 + \bar{y}_1y_2$$

	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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

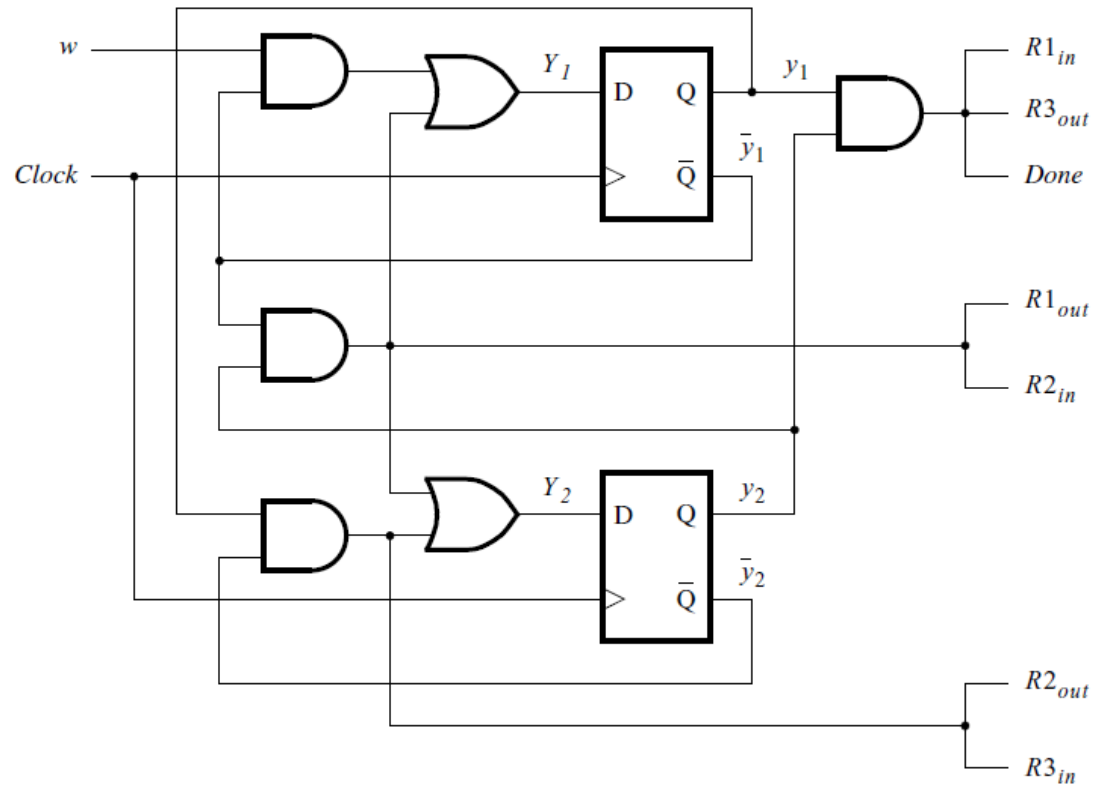
	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	$y_2 y_1$	$Y_2 Y_1$	$Y_2 Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
A	00	00	0 1	0	0	0	0	0	0	0
B	01	10	1 0	0	0	1	0	0	1	0
C	10	11	1 1	1	0	0	1	0	0	0
D	11	00	0 0	0	1	0	0	1	0	1

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

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

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

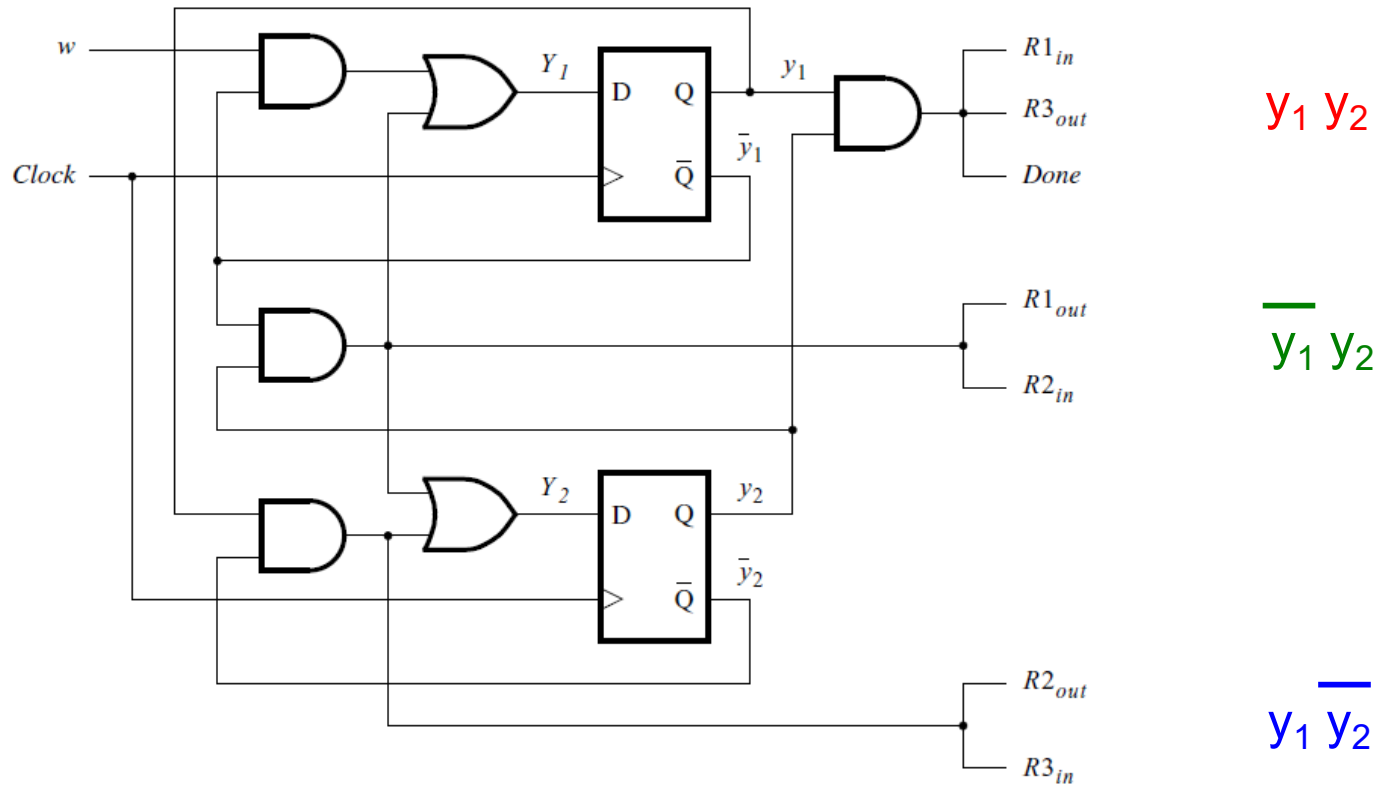
$$R2_{out} = R3_{in} = y_1 \overline{y_2}$$



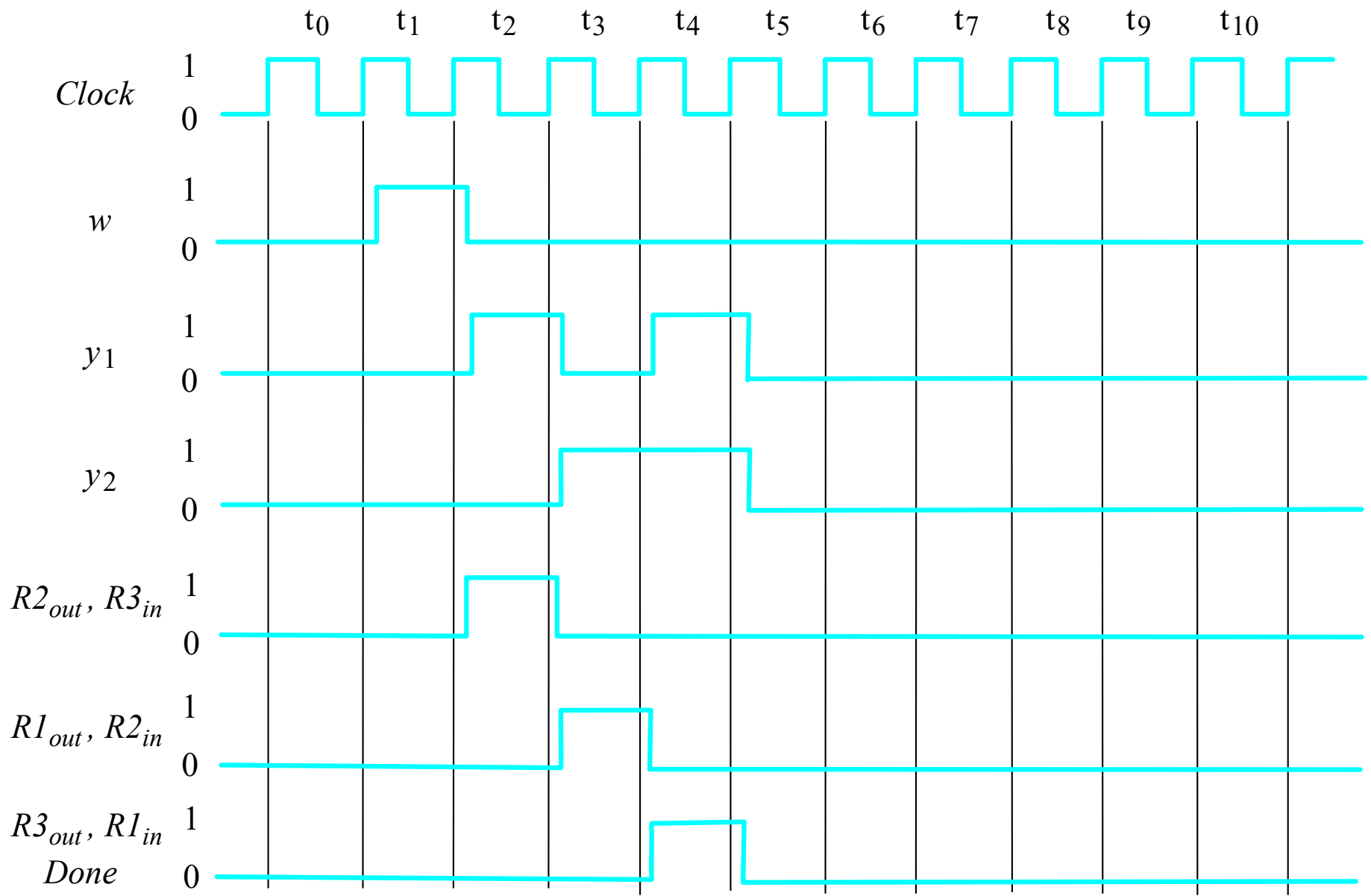
	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
A	00	00	0 1	0	0	0	0	0	0	0
B	01	10	1 0	0	0	1	0	0	1	0
C	10	11	1 1	1	0	0	1	0	0	0
D	11	00	0 0	0	1	0	0	1	0	1

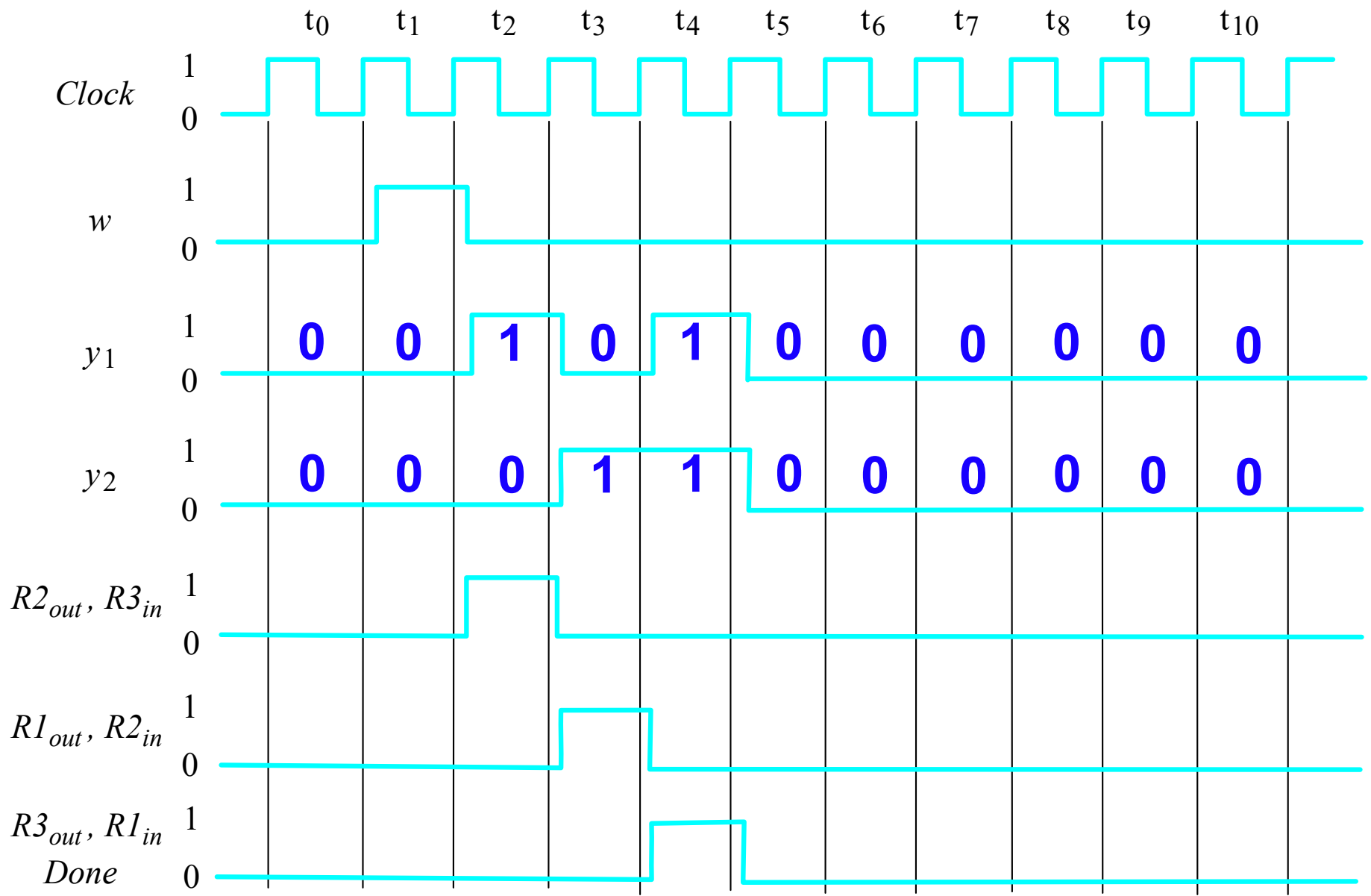
$$Y_1 = w\bar{y}_1 + \bar{y}_1y_2$$

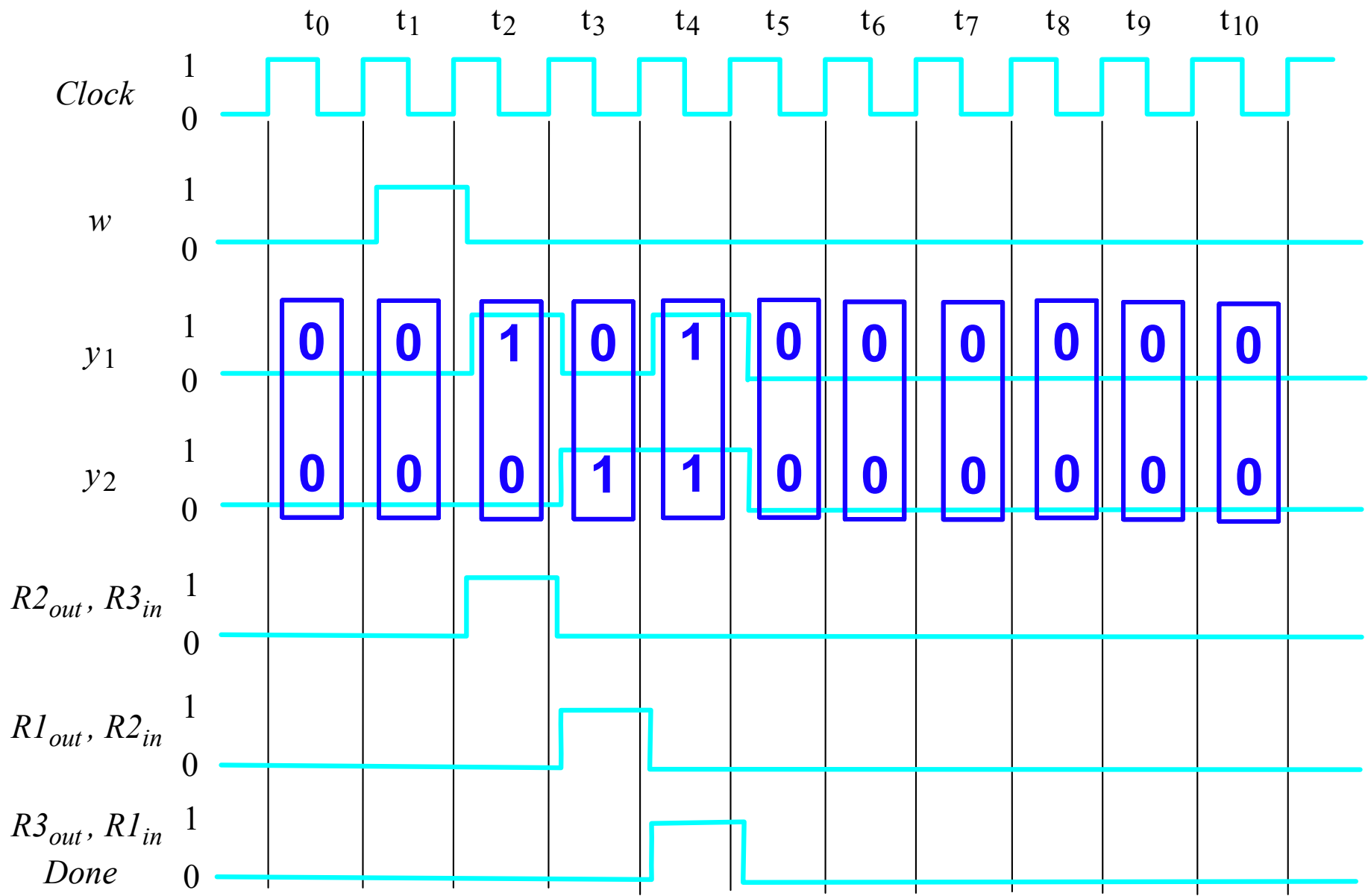
$$Y_2 = y_1\bar{y}_2 + \bar{y}_1y_2$$

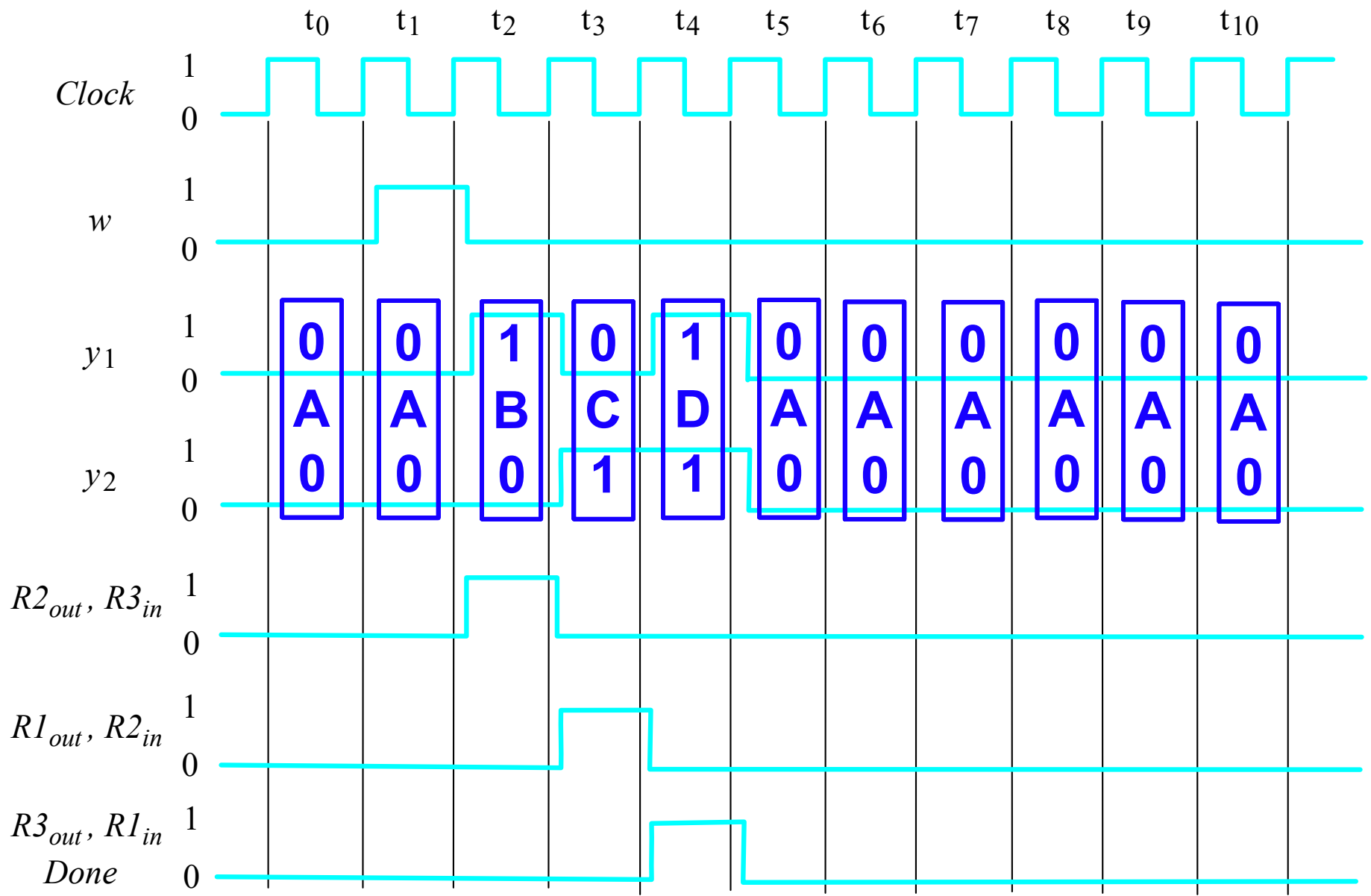


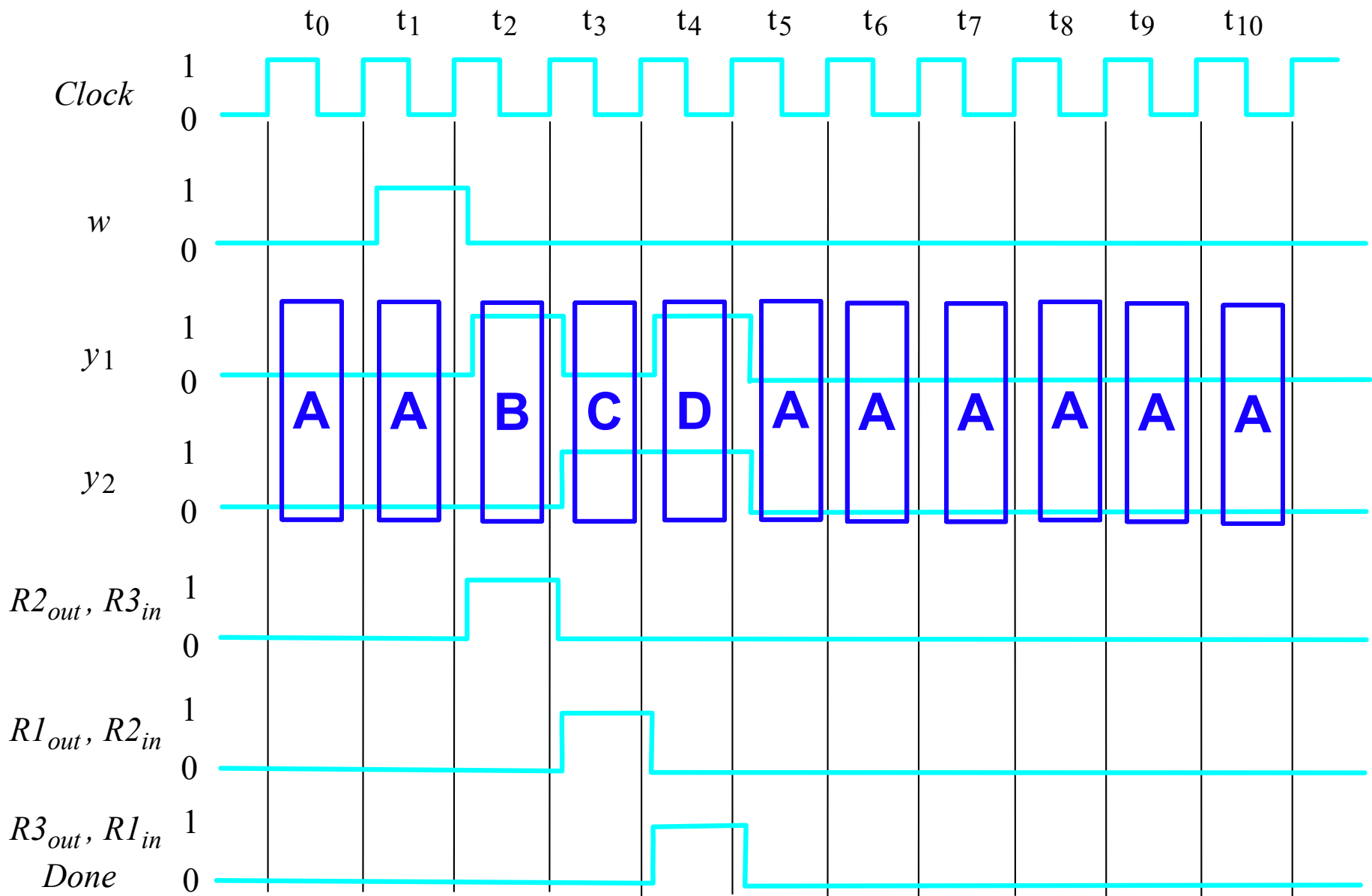
	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	$Done$
A	00	00	0 1	0	0	0	0	0	0	0
B	01	10	1 0	0	0	1	0	0	1	0
C	10	11	1 1	1	0	0	1	0	0	0
D	11	00	0 0	0	1	0	0	1	0	1

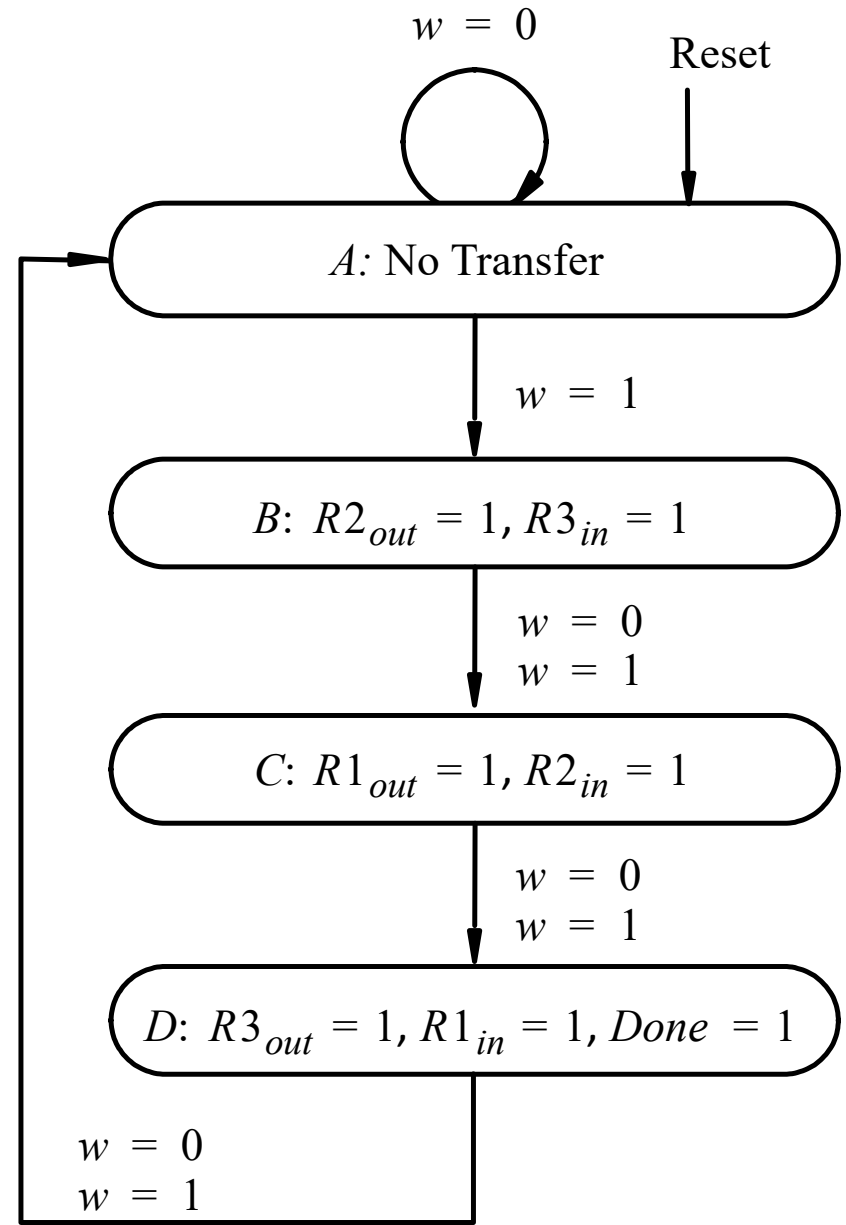
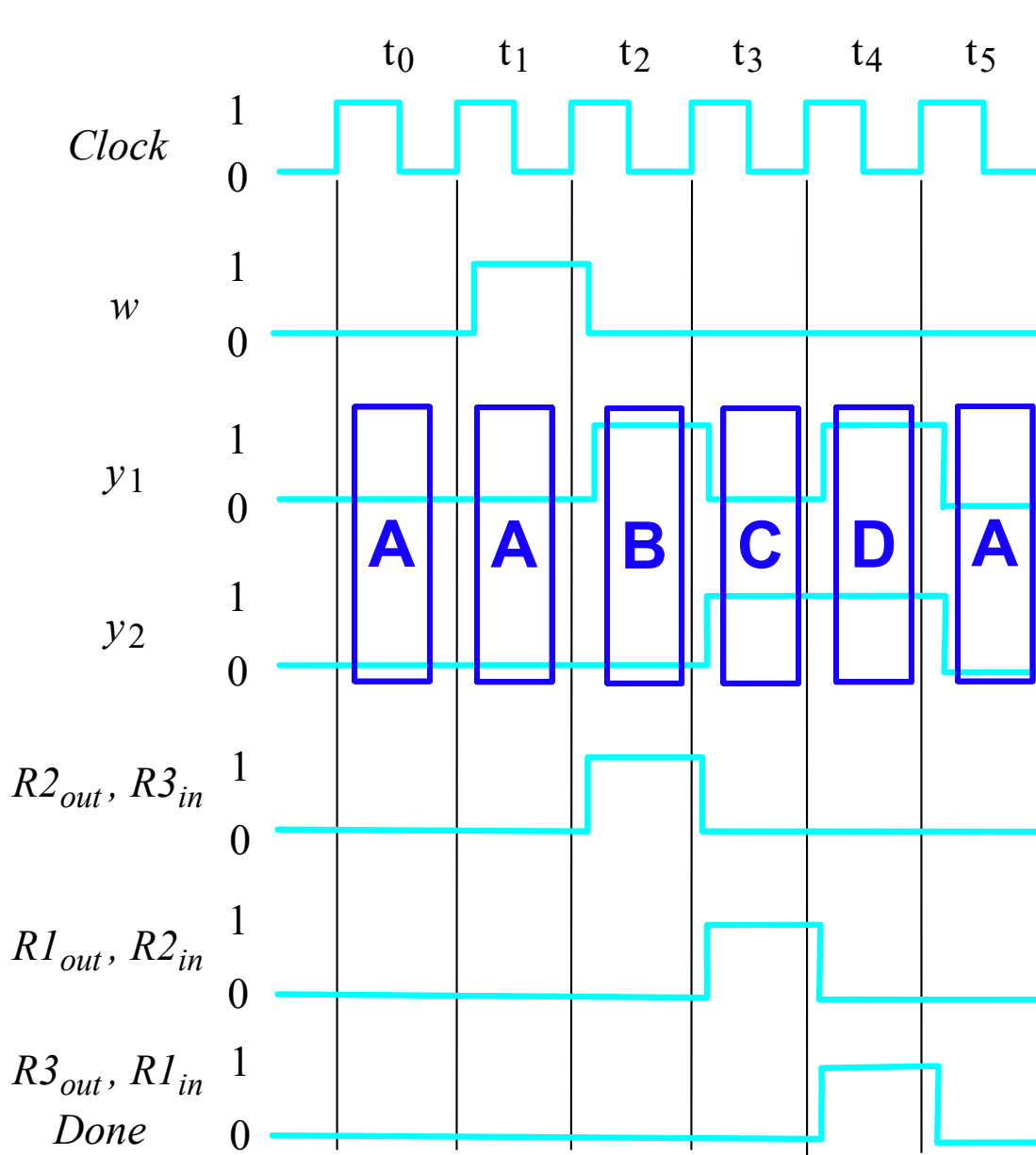


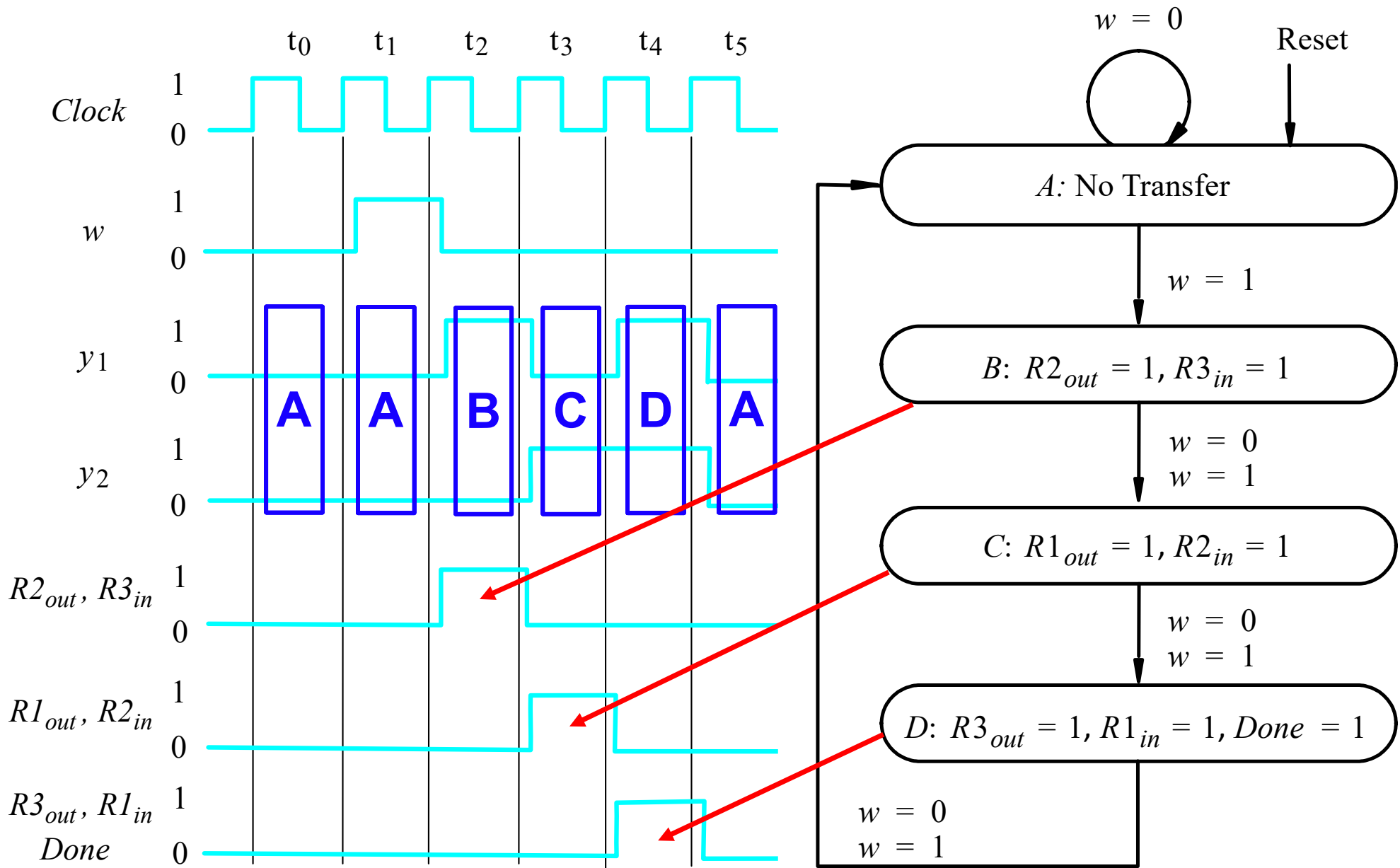












Encoding #2:

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

(Also Uses Two Flip-Flops)

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}	<i>Done</i>
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_2y_1	Y_2Y_1	Y_2Y_1	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
A										
B										
C										
D										

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}	<i>Done</i>
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_2y_1	Y_2Y_1	Y_2Y_1	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
A	00									
B	01									
C	11									
D	10									

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}	<i>Done</i>
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_2y_1	Y_2Y_1	Y_2Y_1	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
A	00	00	01							
B	01	11	11							
C	11	10	10							
D	10	00	00							

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}	<i>Done</i>
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_2y_1	Y_2Y_1	Y_2Y_1	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
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

	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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		

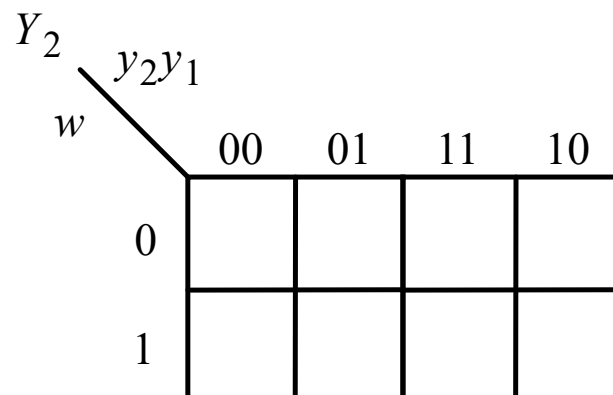
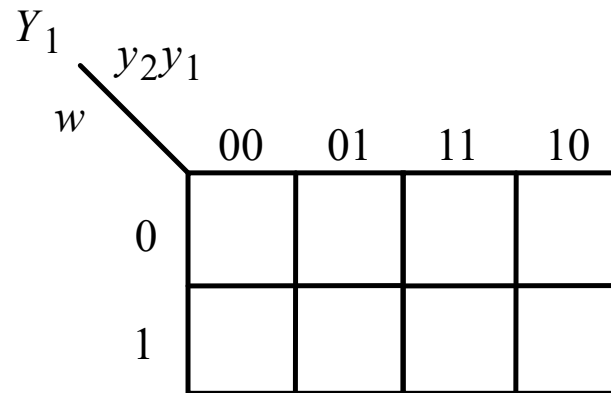
Let's derive the next-state expressions

	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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

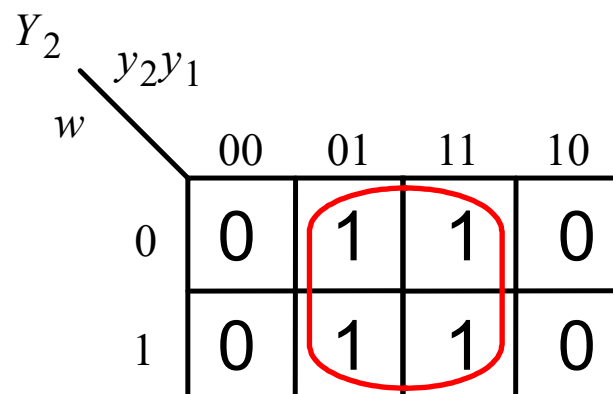
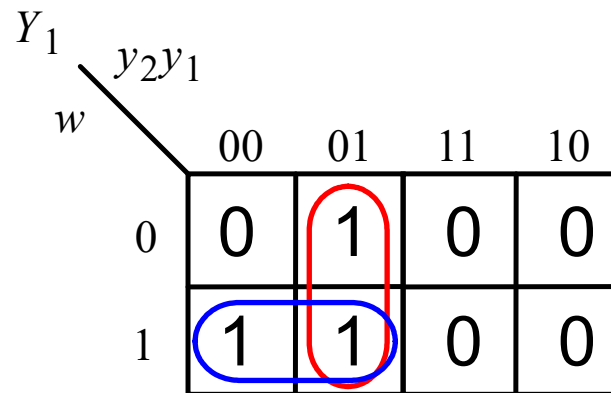
	y_2y_1	00	01	11	10
w					
0		0	1	0	0
1		1	1	0	0

Y_2

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

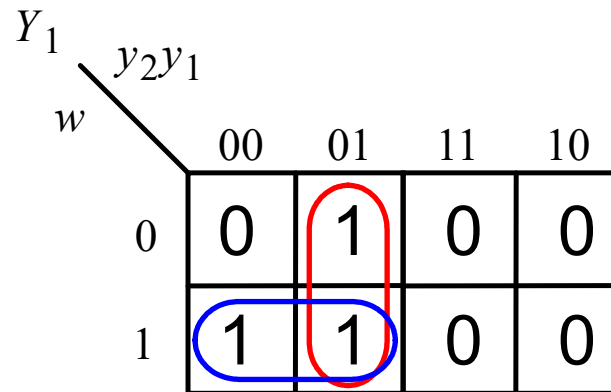
	Present state	Next state		Outputs						
		$w = 0$	$w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
	y_2y_1	Y_2Y_1	Y_2Y_1							
A	00	00	01	0	0	0	0	0	0	0
B	01	11	11	0	0	1	0	0	1	0
C	11	10	10	1	0	0	1	0	0	0
D	10	00	00	0	1	0	0	1	0	1

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

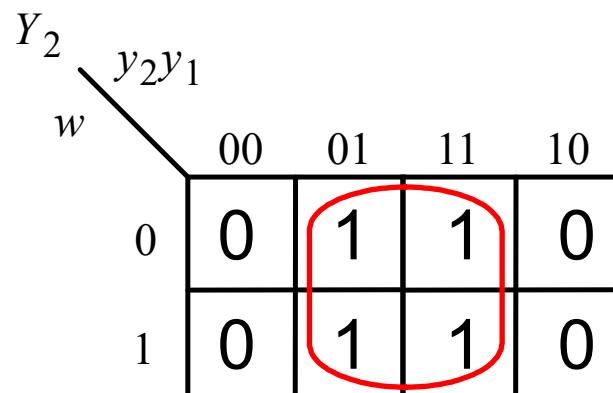


	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
A	00	00	0 1	0	0	0	0	0	0	0
B	01	11	1 1	0	0	1	0	0	1	0
C	11	10	1 0	1	0	0	1	0	0	0
D	10	00	0 0	0	1	0	0	1	0	1

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



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



$$Y_2 = y_1$$

	Present state	Next state		Outputs						
		$w = 0$	$w = 1$							
	y_2y_1	Y_2Y_1	Y_2Y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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						
		$w = 0$	$w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
	y_2y_1	Y_2Y_1	Y_2Y_1							
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						
		$w = 0$	$w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
	y_2y_1	Y_2Y_1	Y_2Y_1							
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						
		$w = 0$	$w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
	y_2y_1	Y_2Y_1	Y_2Y_1							
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						
		$w = 0$	$w = 1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
	$y_2 y_1$	$Y_2 Y_1$	$Y_2 Y_1$							
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

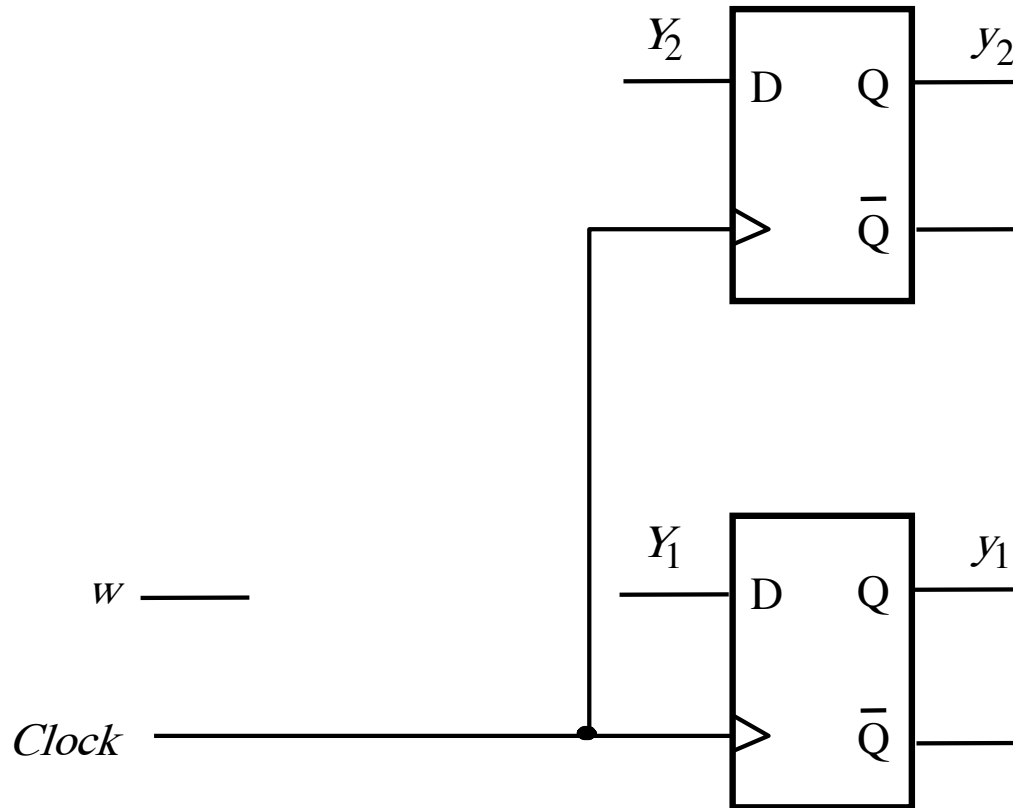
	y_2	y_1	$R1_{out}$	$R1_{in}$	$R2_{out}$
A	0	0	0	0	0
B	0	1	0	0	1
D	1	0	0	1	0
C	1	1	1	0	0

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

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

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

Let's Complete the Circuit Diagram



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

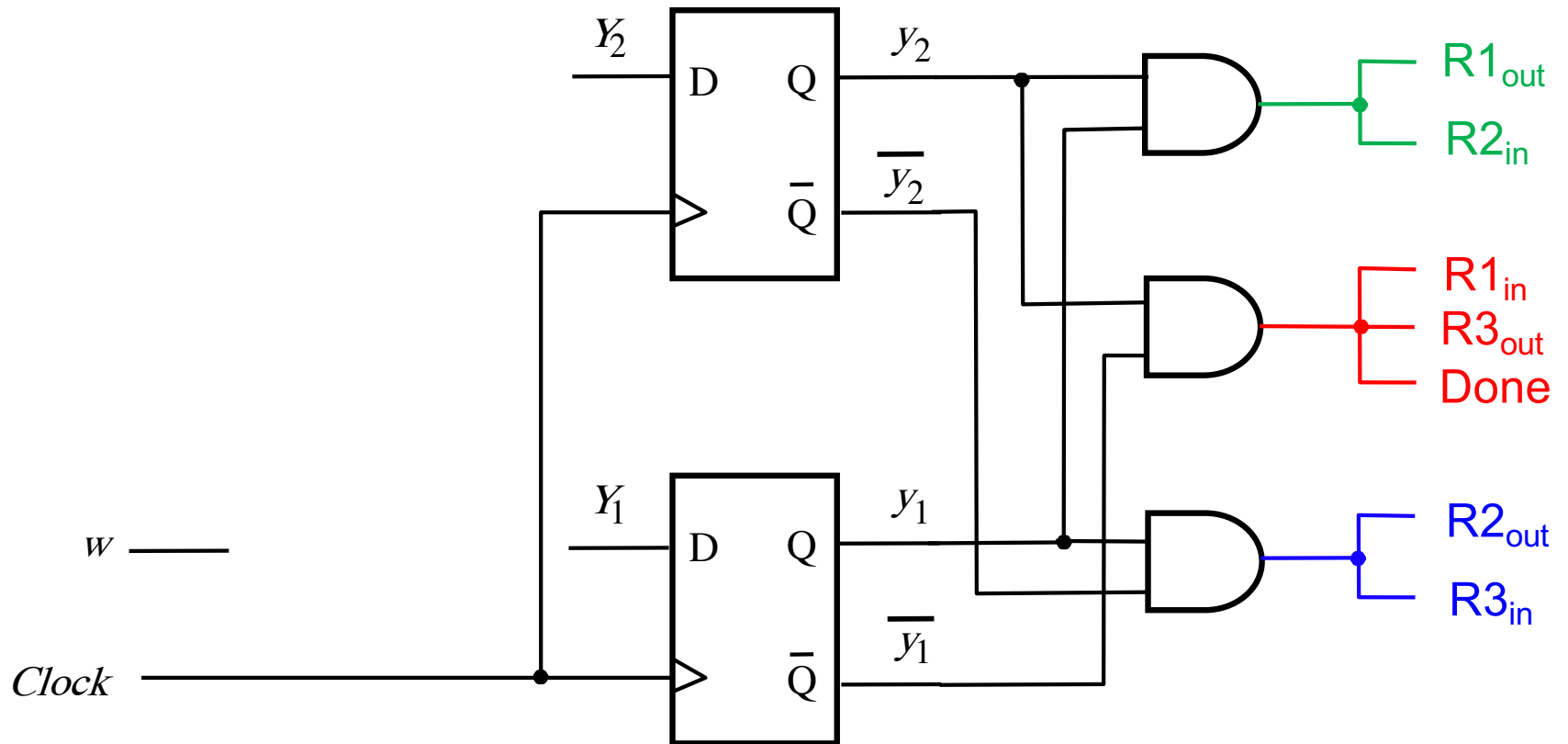
$$Y_2 = y_1$$

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

$$R1_{in} = R3_{out} = Done = \bar{y}_1 y_2$$

$$R2_{out} = R3_{in} = y_1 \bar{y}_2$$

Let's Complete the Circuit Diagram



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

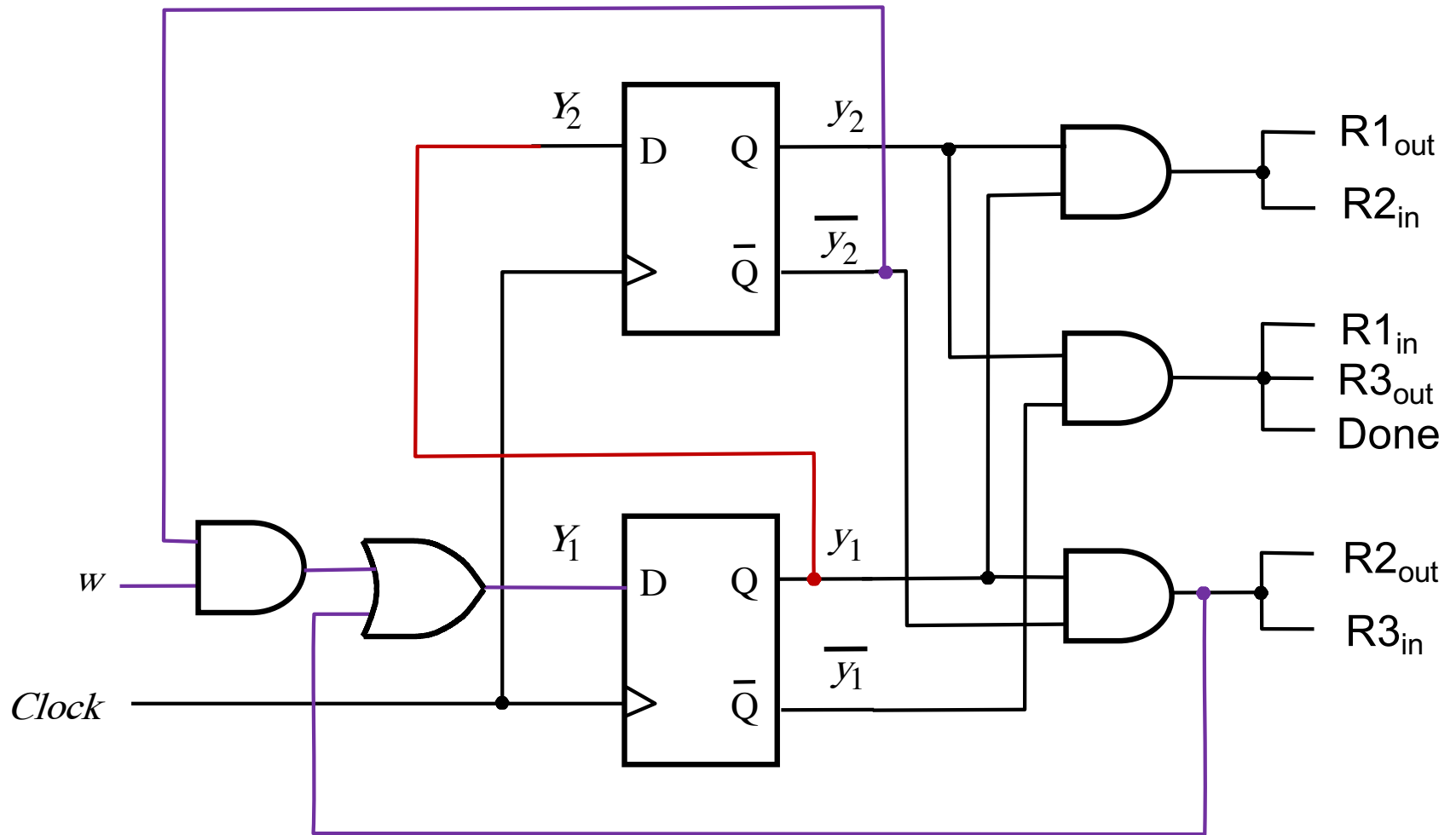
$$Y_2 = y_1$$

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

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

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

Let's Complete the Circuit Diagram



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

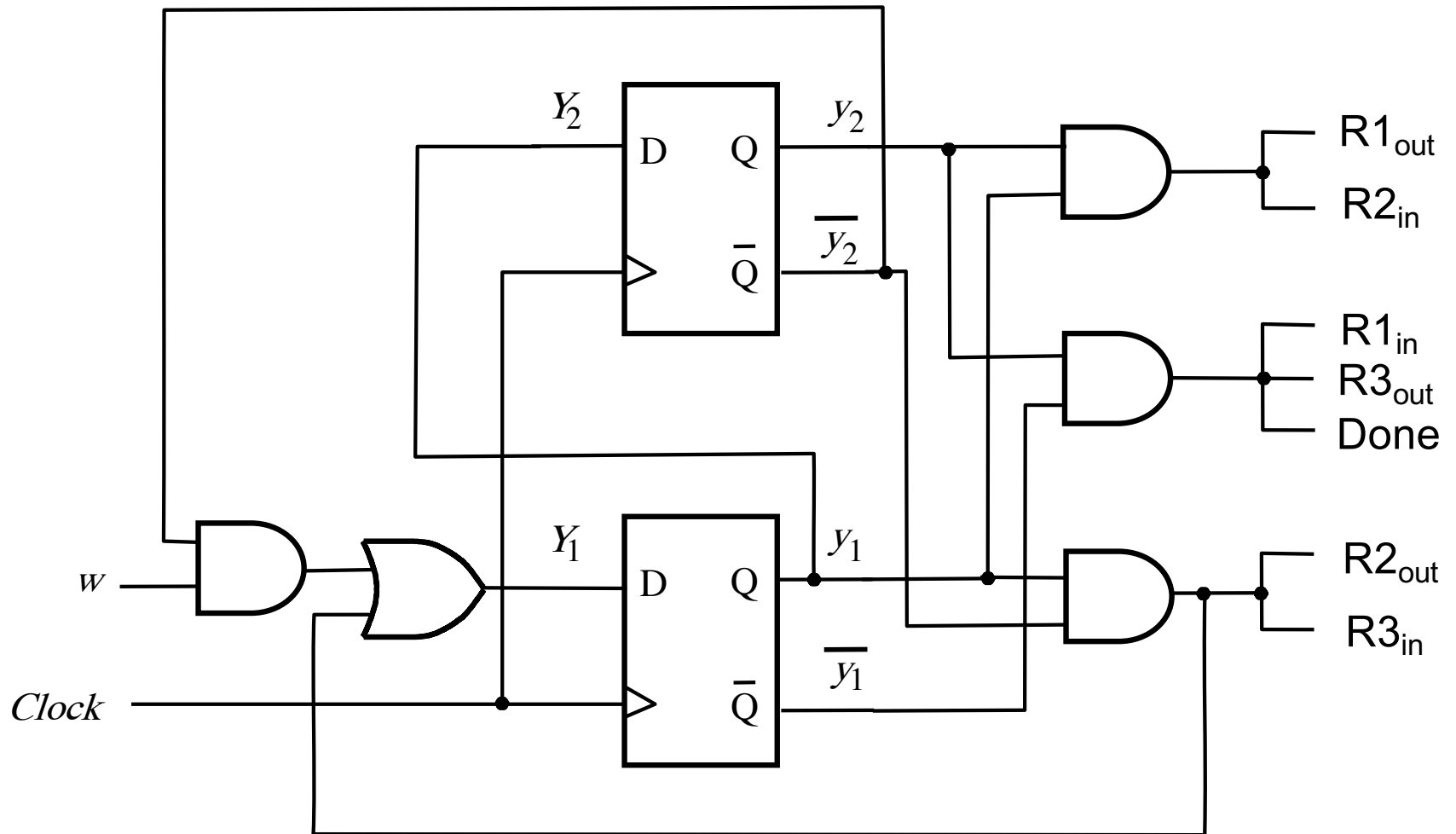
$$Y_2 = y_1$$

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

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

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

Let's Complete the Circuit Diagram



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

$$Y_2 = y_1$$

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

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

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

Encoding #3:

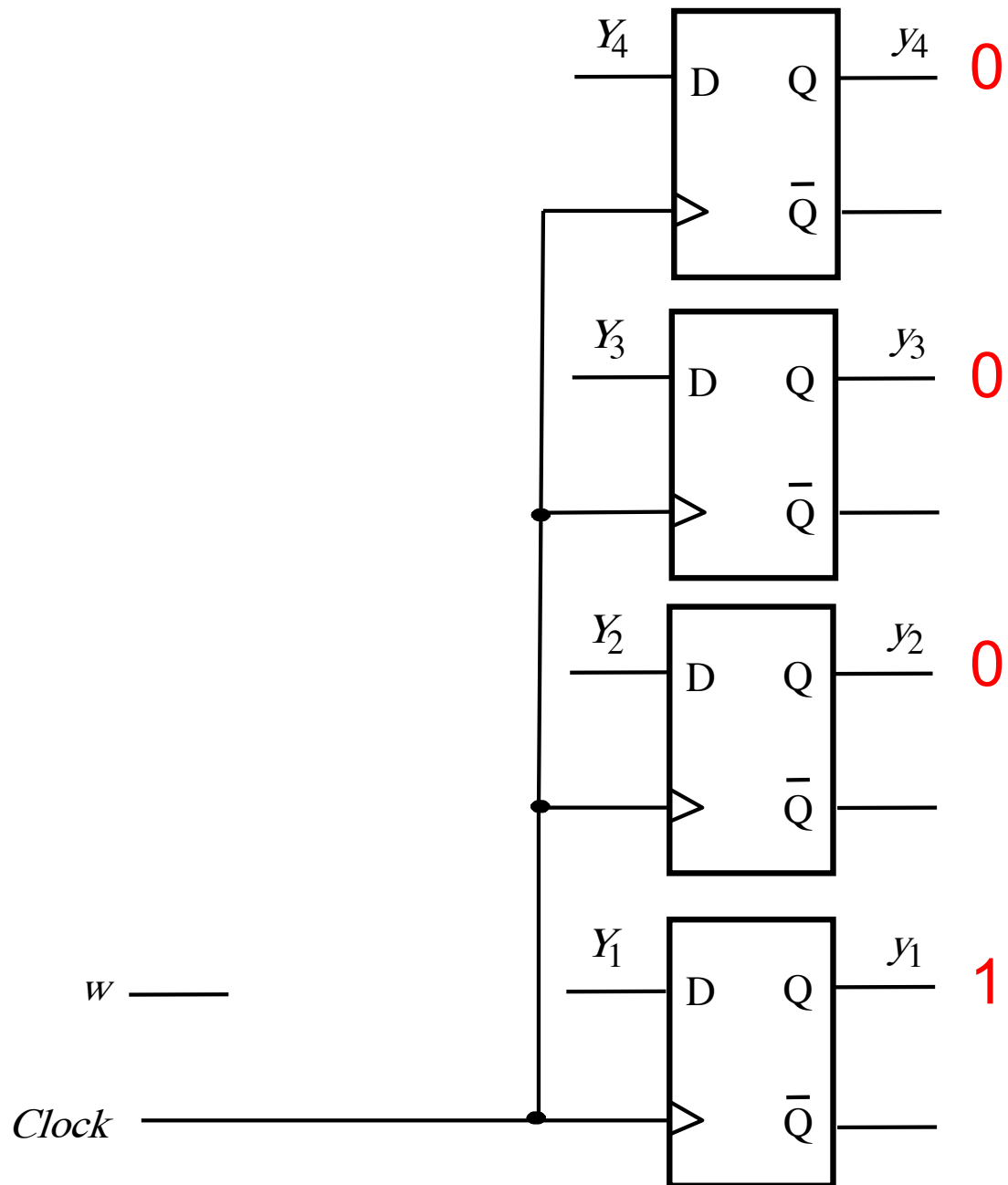
A=0001, B=0010, C=0100, D=1000

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

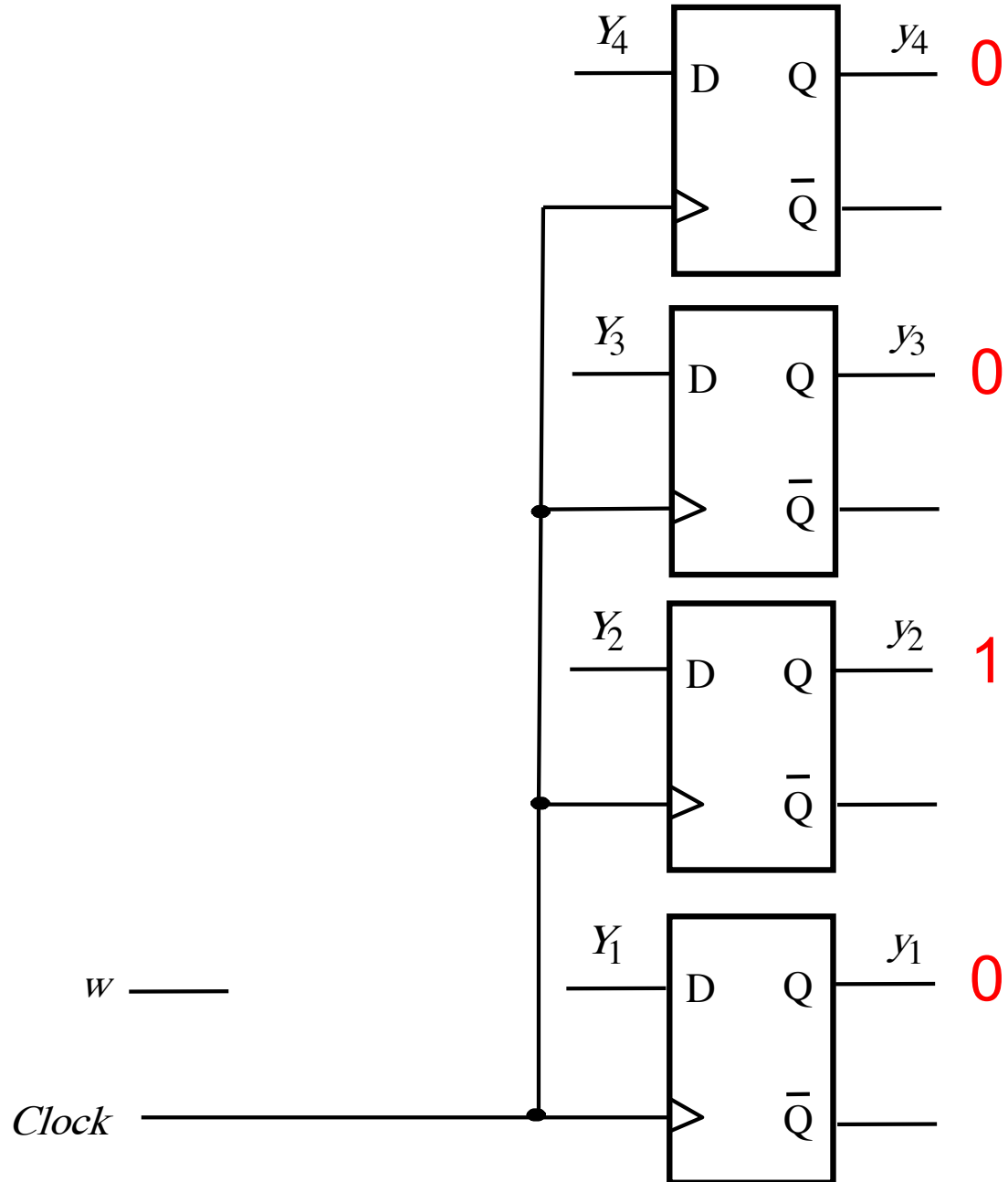
One-Hot State Encoding

- **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.**

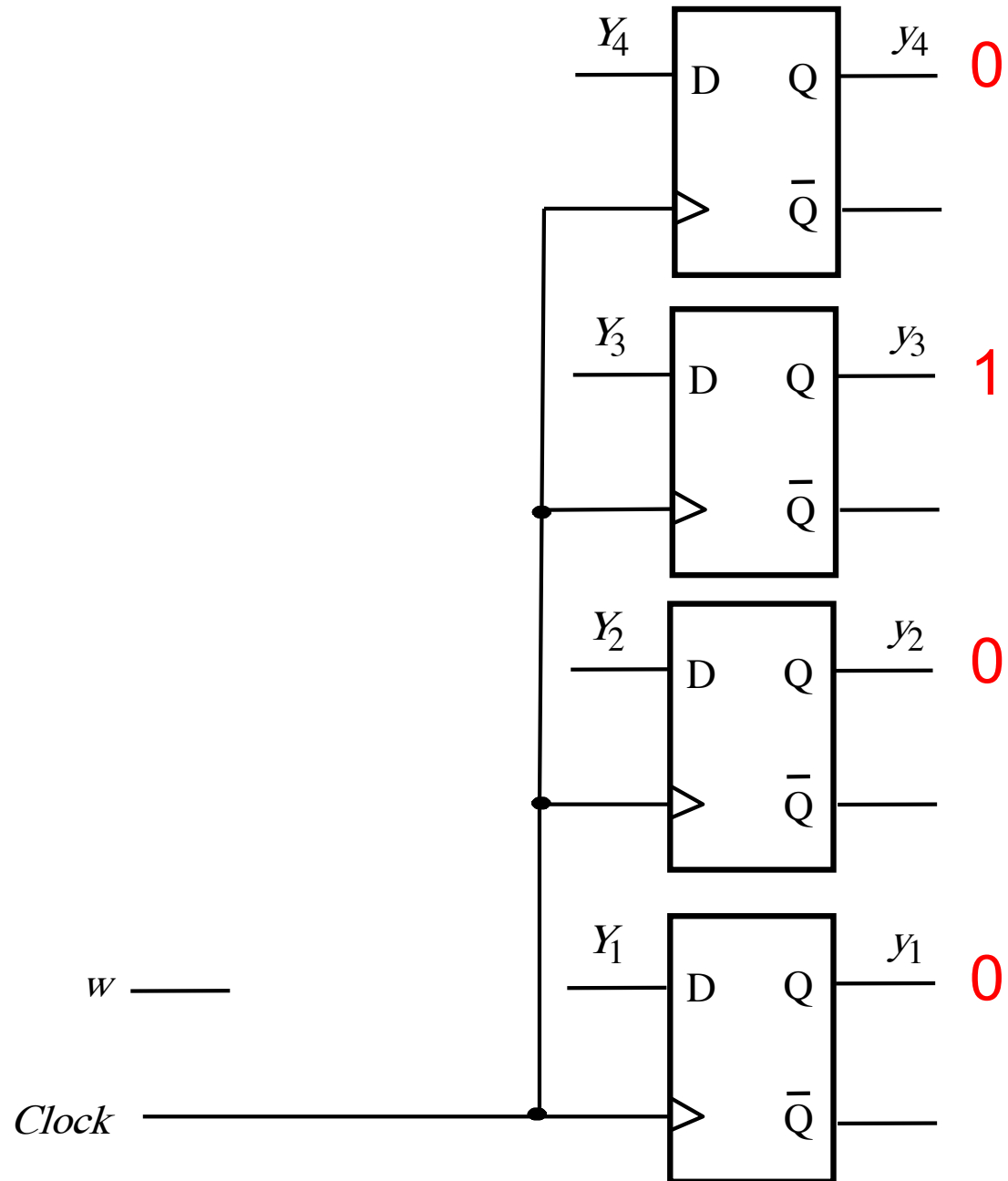
Encoding for State A



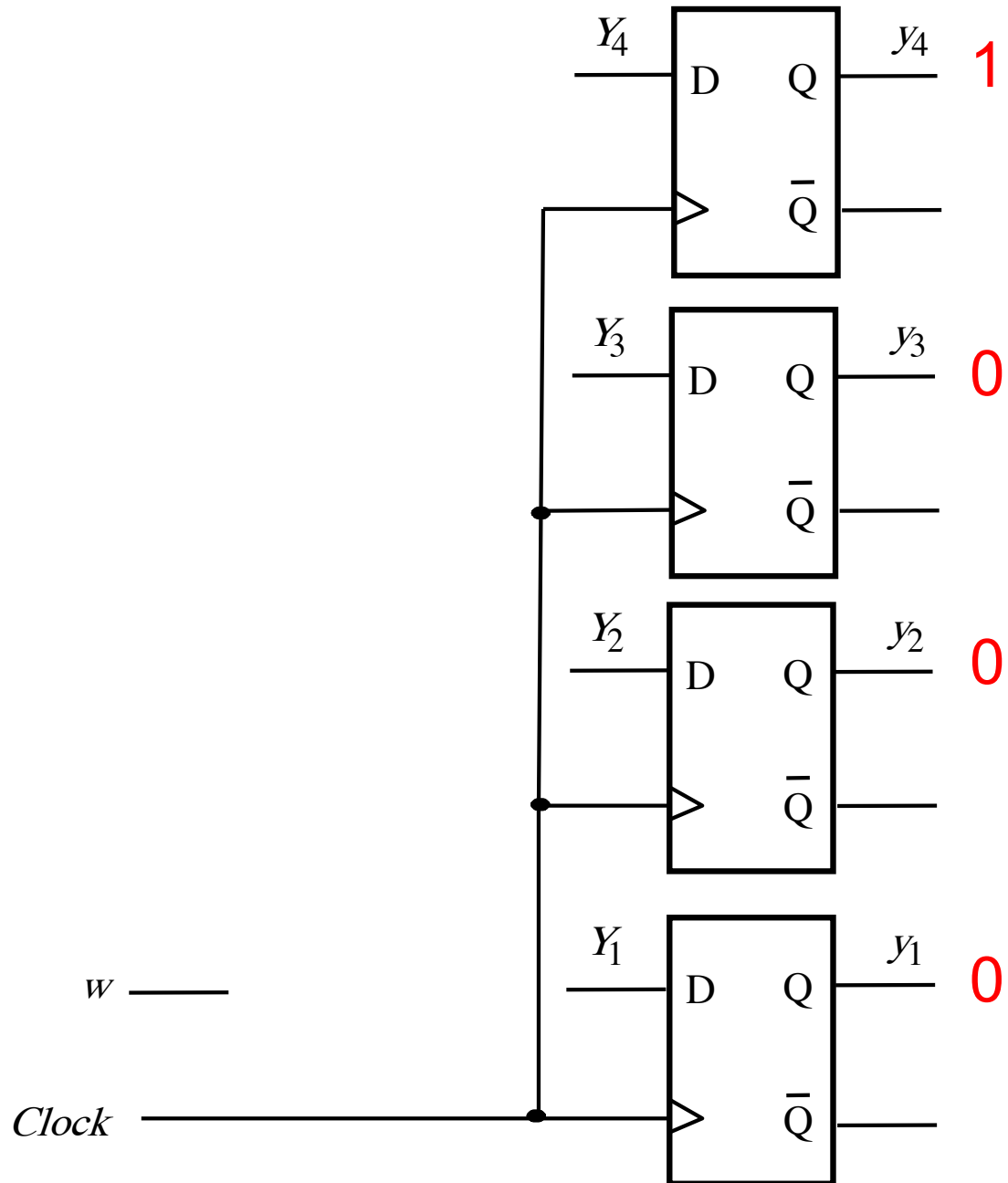
Encoding for State B



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}	<i>Done</i>
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}	<i>Done</i>
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}	<i>Done</i>
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		Outputs						
	$w = 0$	$w = 1$							
	$Y_4Y_3Y_2Y_1$	$Y_4Y_3Y_2Y_1$	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
A									
B									
C									
D									

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}	<i>Done</i>
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}	<i>Done</i>
A	0 001									
B	0 010									
C	0 100									
D	1 000									

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}	<i>Done</i>
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}	<i>Done</i>
A	0 001	0001	0010							
B	0 010	0100	0100							
C	0 100	1000	1000							
D	1 000	0001	0001							

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}	<i>Done</i>
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}	<i>Done</i>
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

	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}	<i>Done</i>
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						
		$w = 0$	$w = 1$							
	$y_4 y_3 y_2 y_1$	$Y_4 Y_3 Y_2 Y_1$	$Y_4 Y_3 Y_2 Y_1$	R1 _{out}	R1 _{in}	R2 _{out}	R2 _{in}	R3 _{out}	R3 _{in}	<i>Done</i>
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

- $Y_1(w, y_4, y_3, y_2, y_1) = \neg w y_1 + y_4$

- $Y_2(w, y_4, y_3, y_2, y_1) = w y_1$

- $Y_3(w, y_4, y_3, y_2, y_1) = y_2$

- $Y_4(w, y_4, y_3, y_2, y_1) = y_3$

Or we can be smarter than that 😊

	Present State	Next State		Outputs						
		$w = 0$	$w = 1$							
	$y_4 y_3 y_2 y_1$	$Y_4 Y_3 Y_2 Y_1$	$Y_4 Y_3 Y_2 Y_1$	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 Next-State Expressions

• $Y_1(w, y_4, y_3, y_2, y_1) = \neg w y_1 + y_4$ (why?)

• $Y_2(w, y_4, y_3, y_2, y_1) = w y_1$ (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						
		$w = 0$	$w = 1$							
	$y_4 y_3 y_2 y_1$	$Y_4 Y_3 Y_2 Y_1$	$Y_4 Y_3 Y_2 Y_1$	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

	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}	<i>Done</i>
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						
		$w = 0$	$w = 1$							
	$y_4 y_3 y_2 y_1$	$Y_4 Y_3 Y_2 Y_1$	$Y_4 Y_3 Y_2 Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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						
		$w = 0$	$w = 1$							
	$y_4y_3y_2y_1$	$Y_4Y_3Y_2Y_1$	$Y_4Y_3Y_2Y_1$	$R1_{out}$	$R1_{in}$	$R2_{out}$	$R2_{in}$	$R3_{out}$	$R3_{in}$	<i>Done</i>
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

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

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

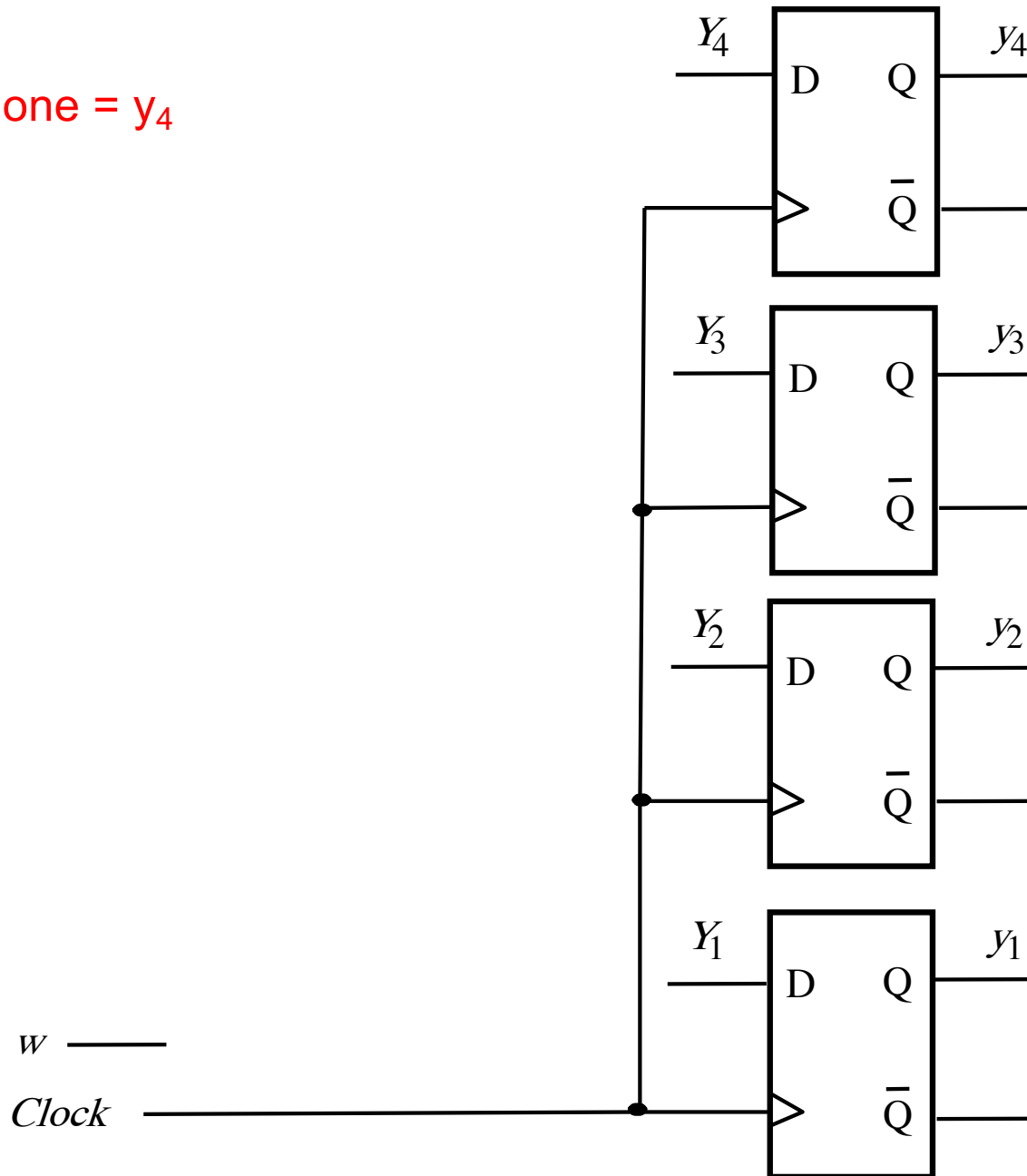
$$R2_{out} = R3_{in} = y_2$$

$$Y_1 = \bar{w} y_1 + y_4$$

$$Y_2 = w y_1$$

$$Y_3 = y_2$$

$$Y_4 = y_3$$



Let's Complete the Circuit Diagram

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

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

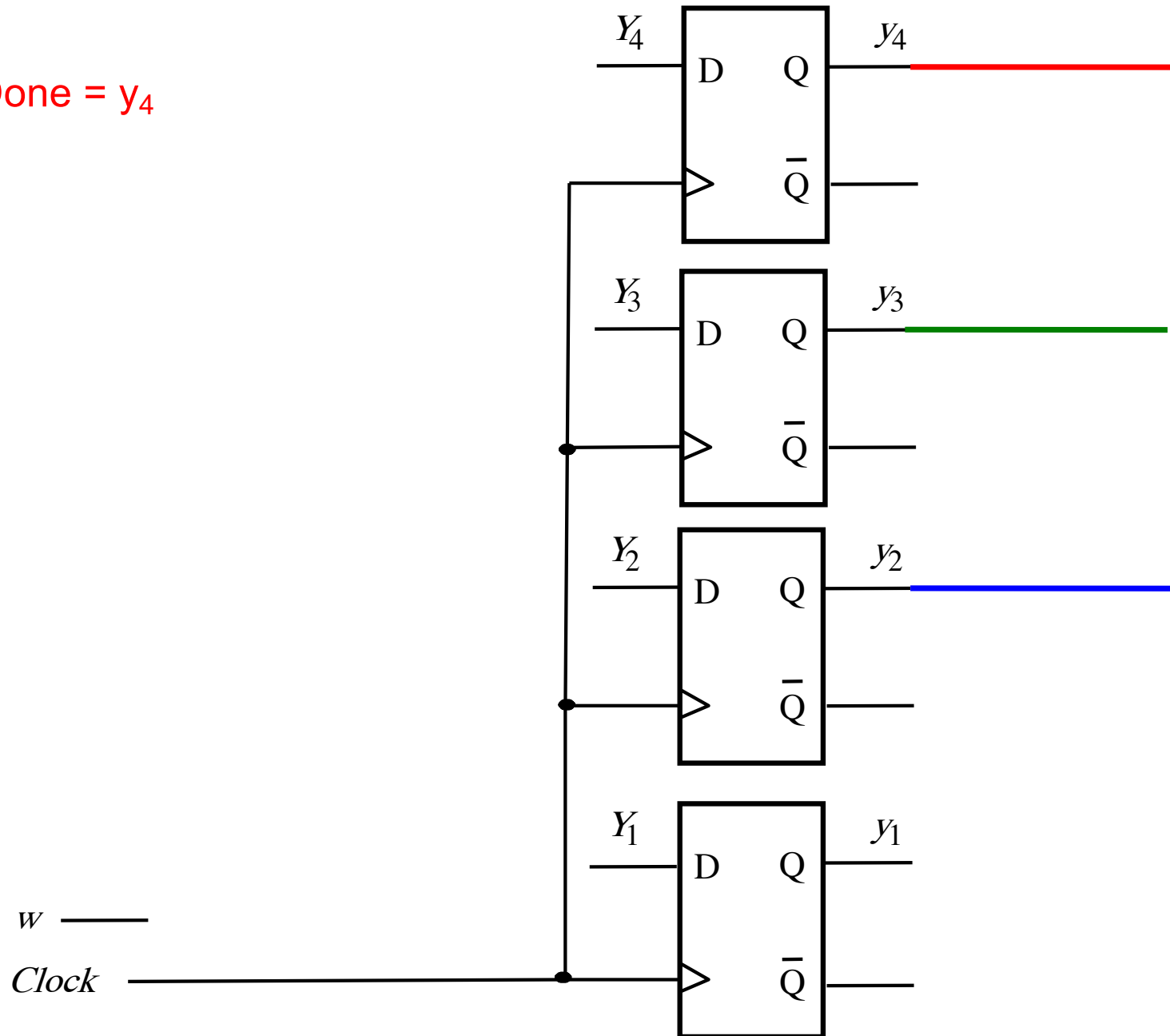
$$R2_{out} = R3_{in} = y_2$$

$$Y_1 = \bar{w} y_1 + y_4$$

$$Y_2 = w y_1$$

$$Y_3 = y_2$$

$$Y_4 = y_3$$

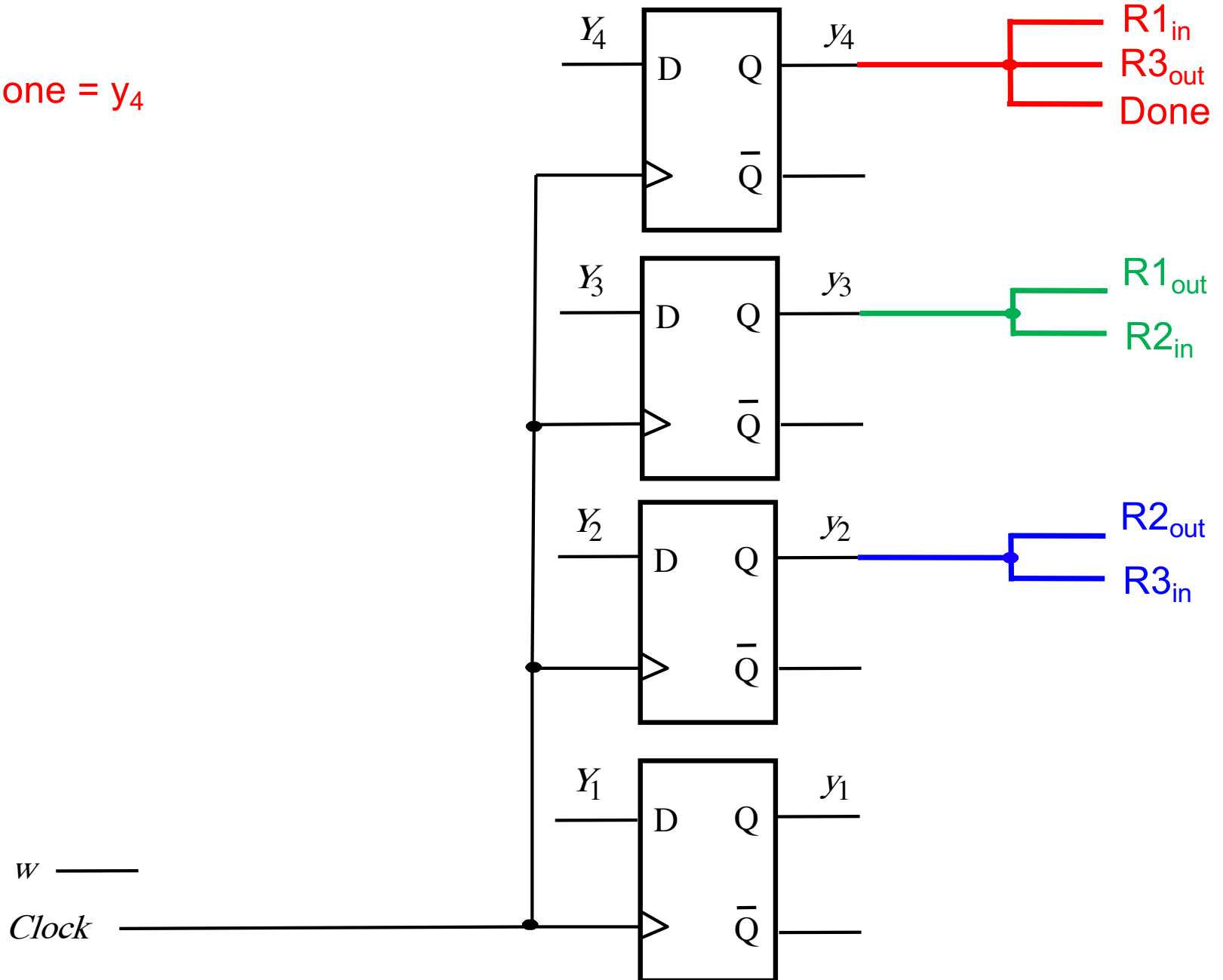


Let's Complete the Circuit Diagram

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

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

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



$$Y_1 = \bar{w} y_1 + y_4$$

$$Y_2 = w y_1$$

$$Y_3 = y_2$$

$$Y_4 = y_3$$

w _____

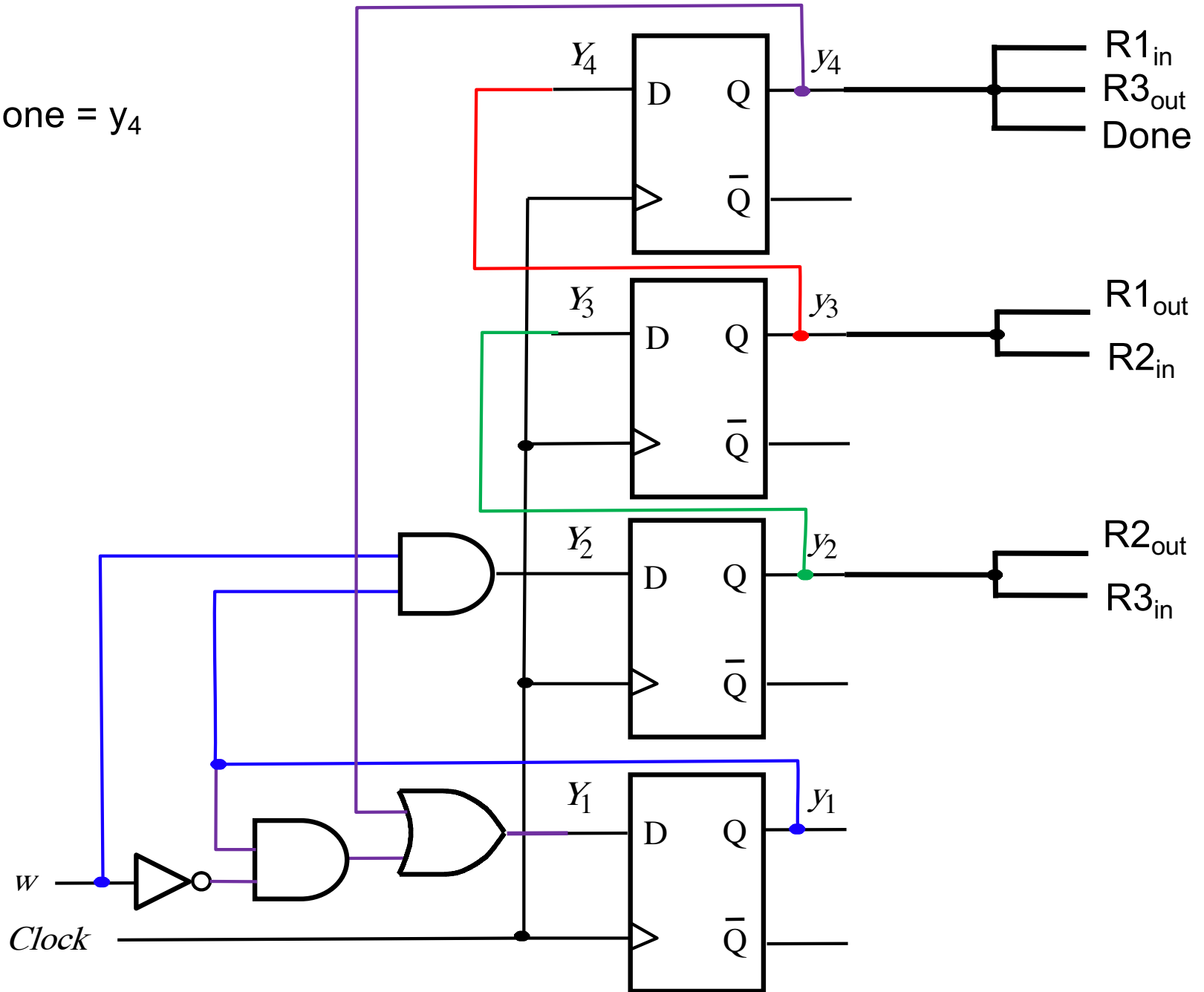
Clock _____

Let's Complete the Circuit Diagram

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

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

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

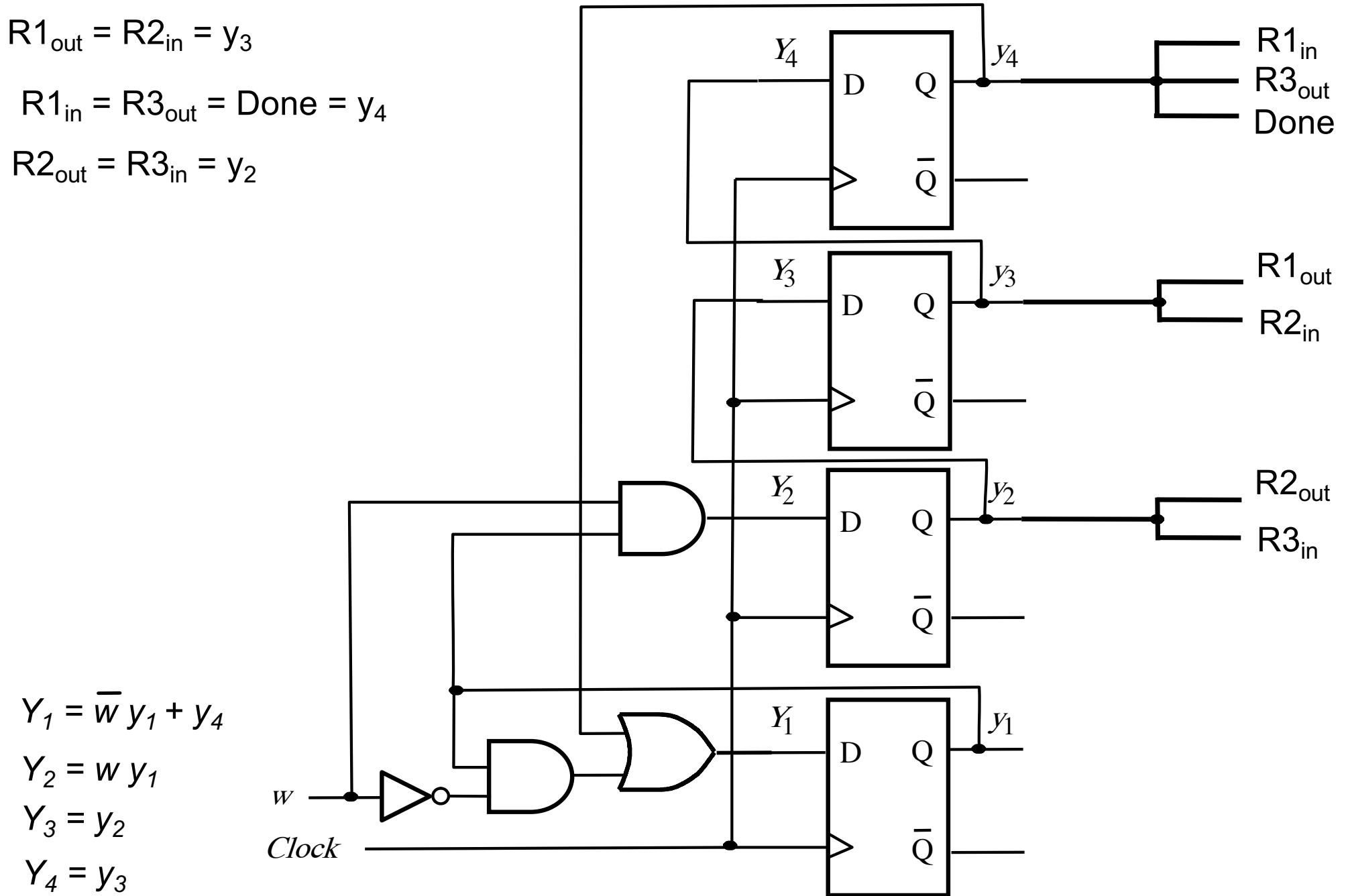


Let's Complete the Circuit Diagram

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

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

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

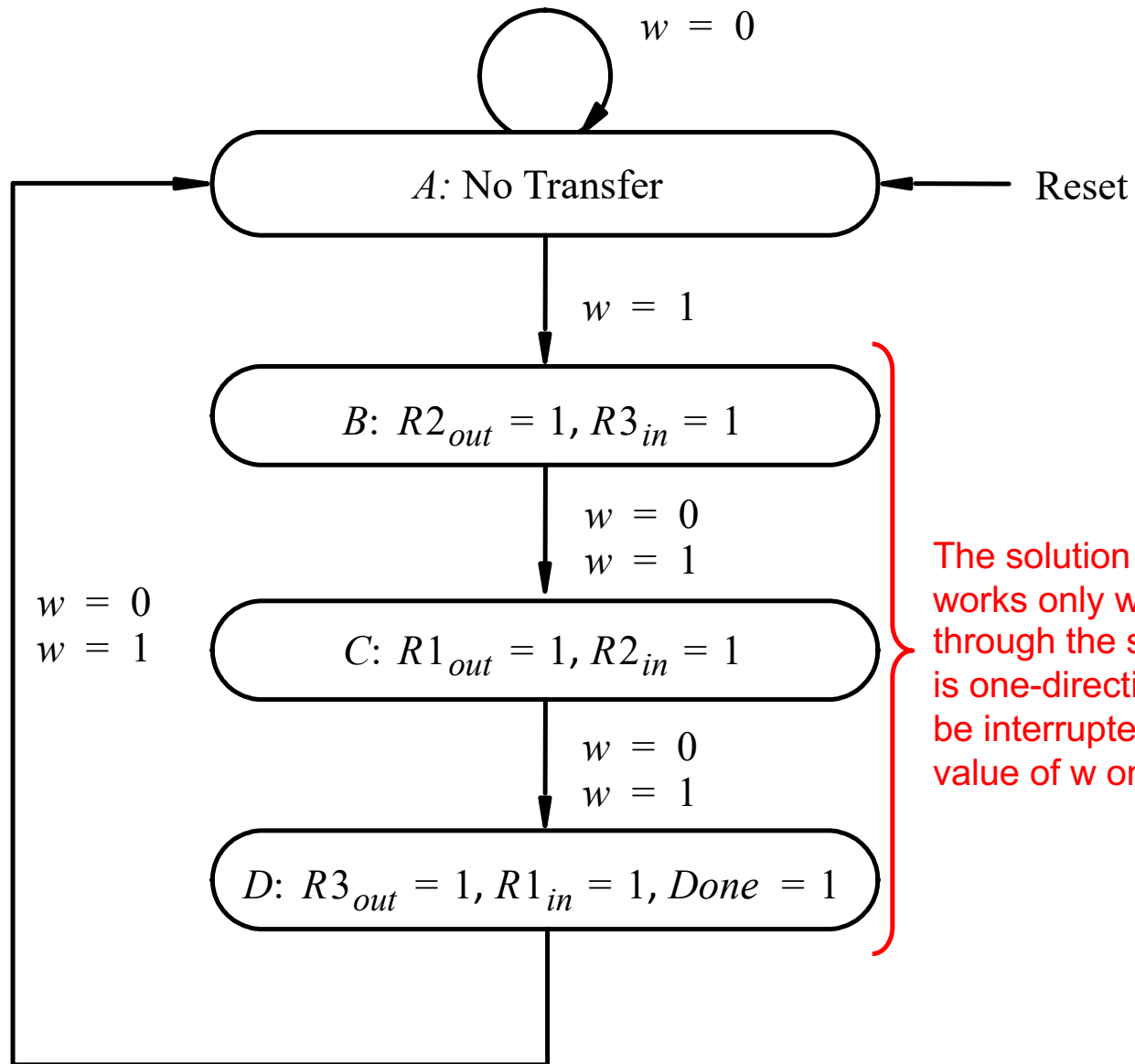


Encoding #4:

A=0001, B=0010, C=0100, D=1000

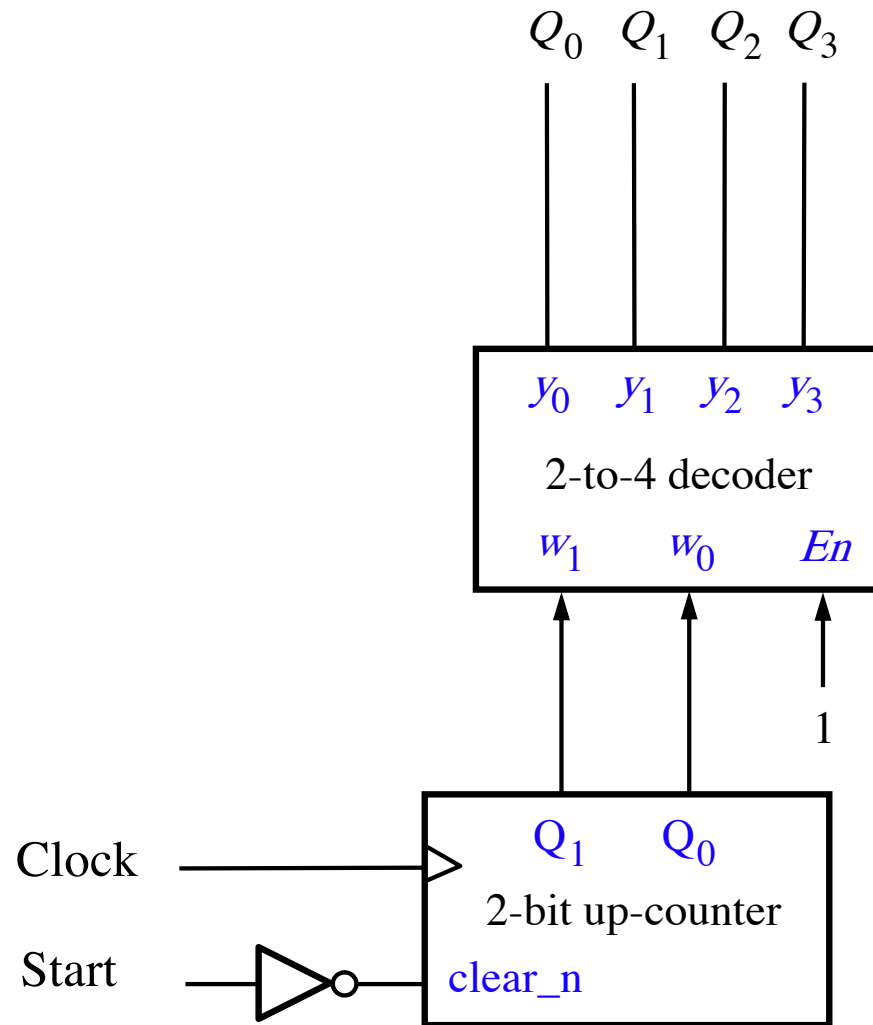
(same as before, but shows an alternative implementation with a 4-bit ring counter)

Exploit the Structure of the FSM



[Figure 6.11 from the textbook]

Alternative version of a 4-bit ring counter



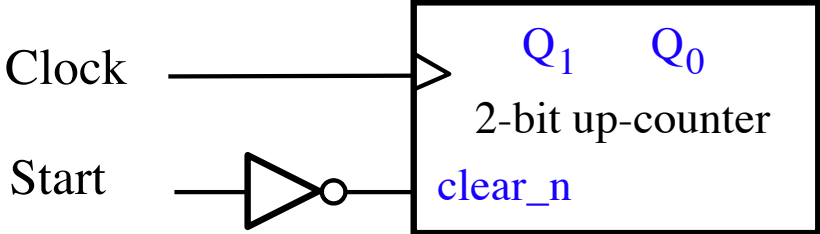
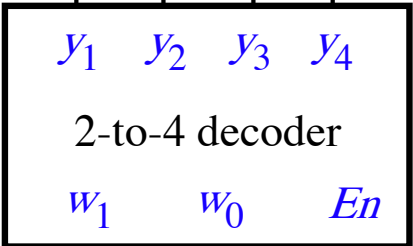
Alternative version of a 4-bit ring counter

Switch to 1-based indexing of the outputs →

Q_1 Q_2 Q_3 Q_4

Switch to 1-based indexing of the outputs →

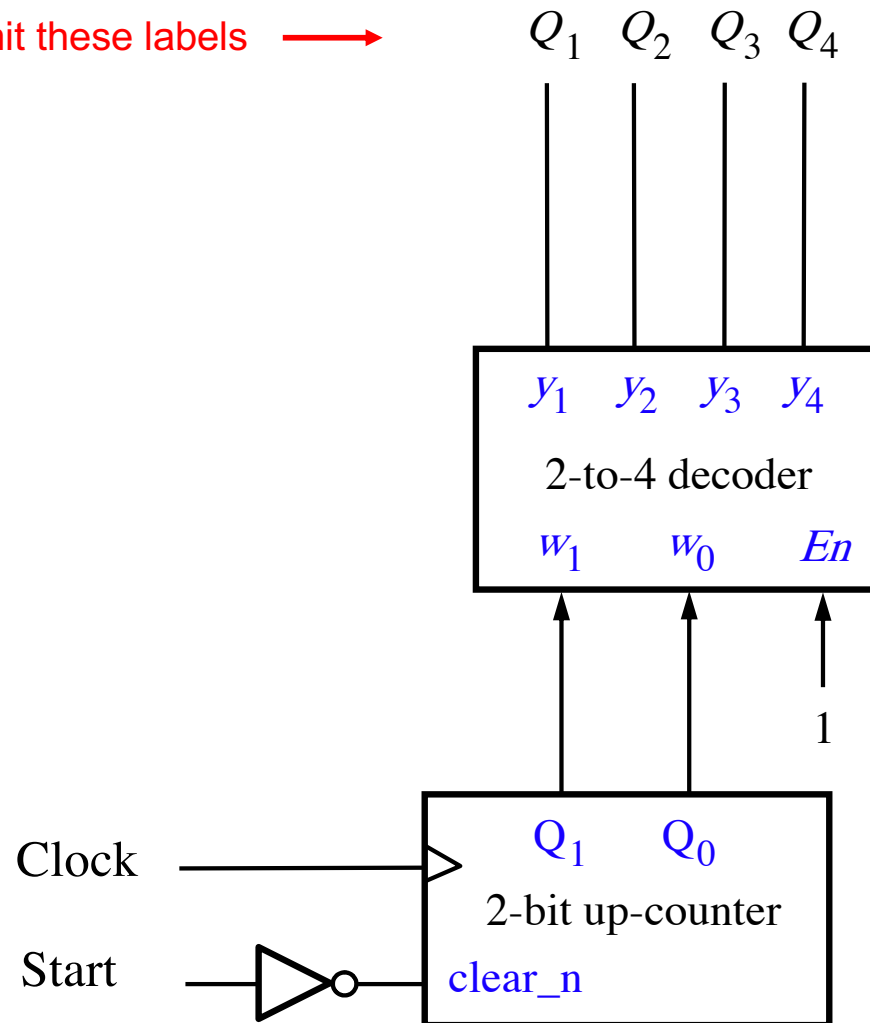
(this is done to be consistent with the previous example)



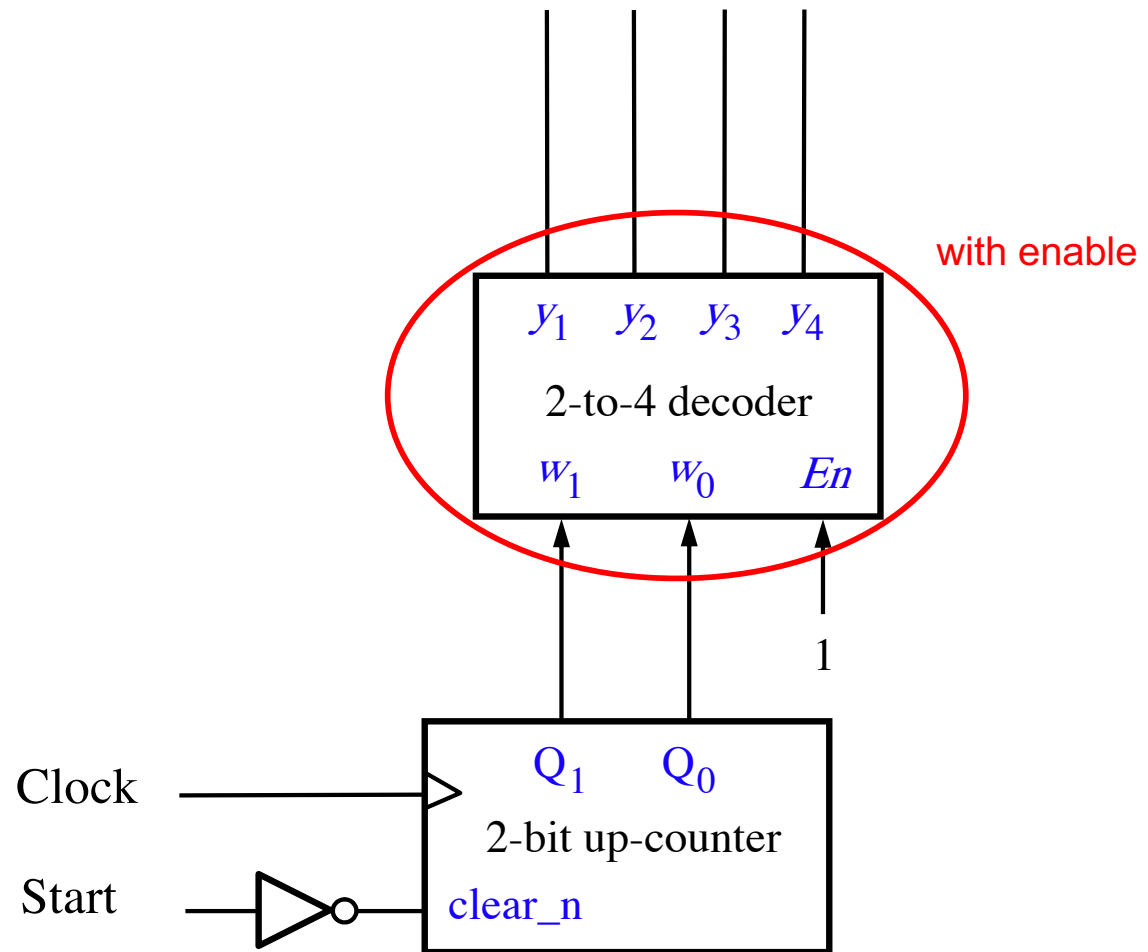
1

Alternative version of a 4-bit ring counter

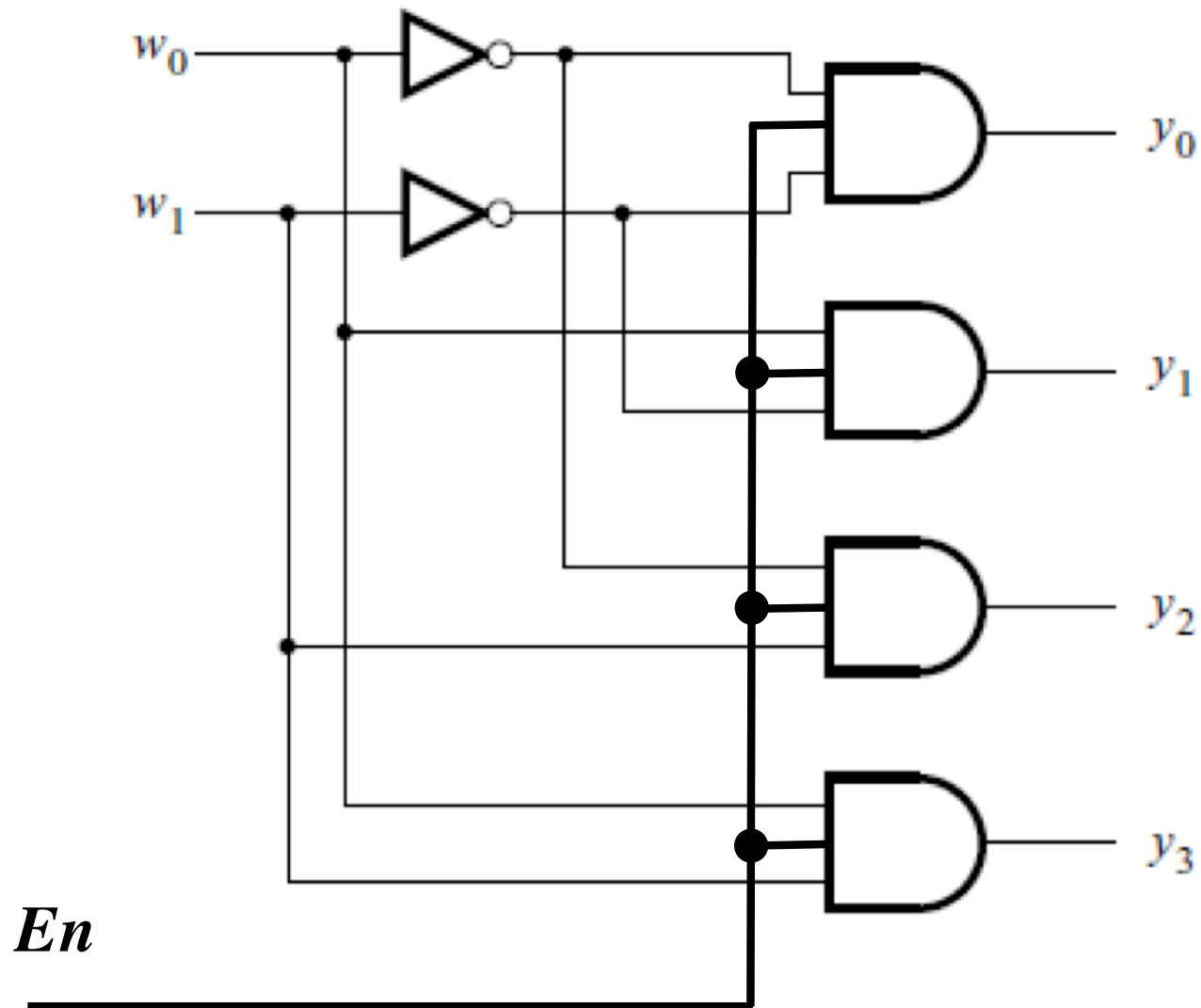
Also, omit these labels →



Alternative version of a 4-bit ring counter

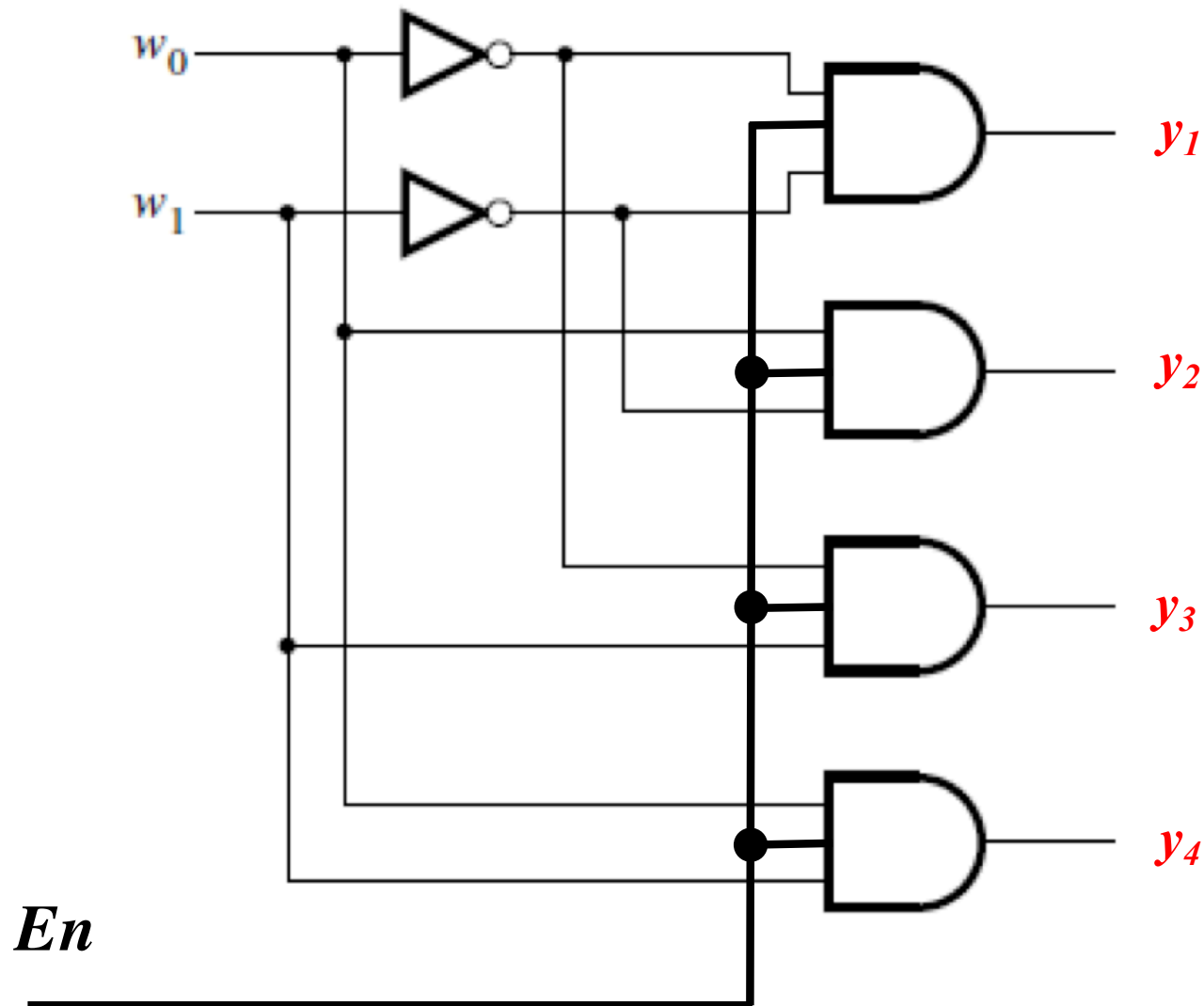


2-to-4 Decoder with Enable Input



[Figure 4.14c from the textbook]

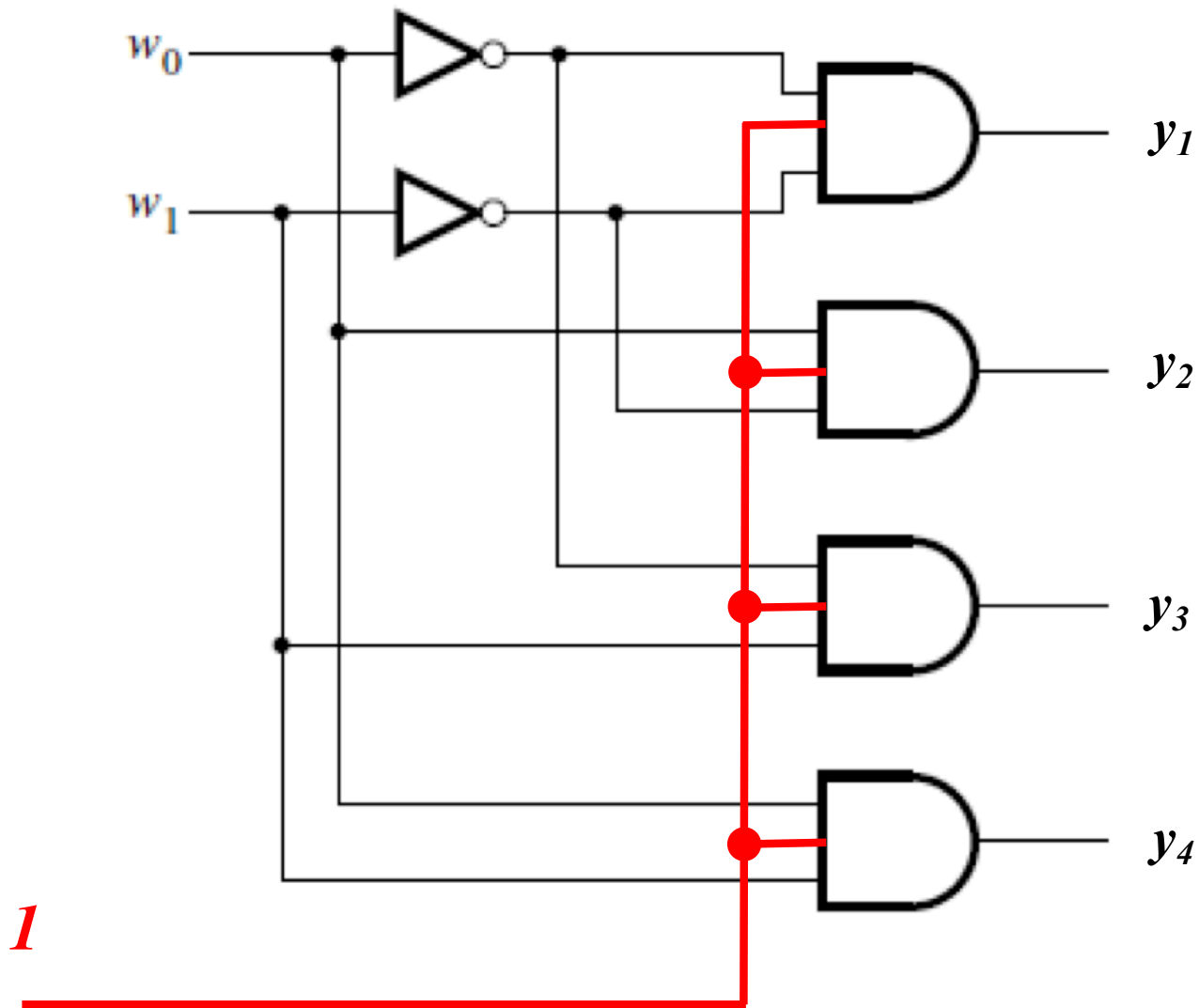
2-to-4 Decoder with Enable Input



Switch to 1-based indexing of the outputs

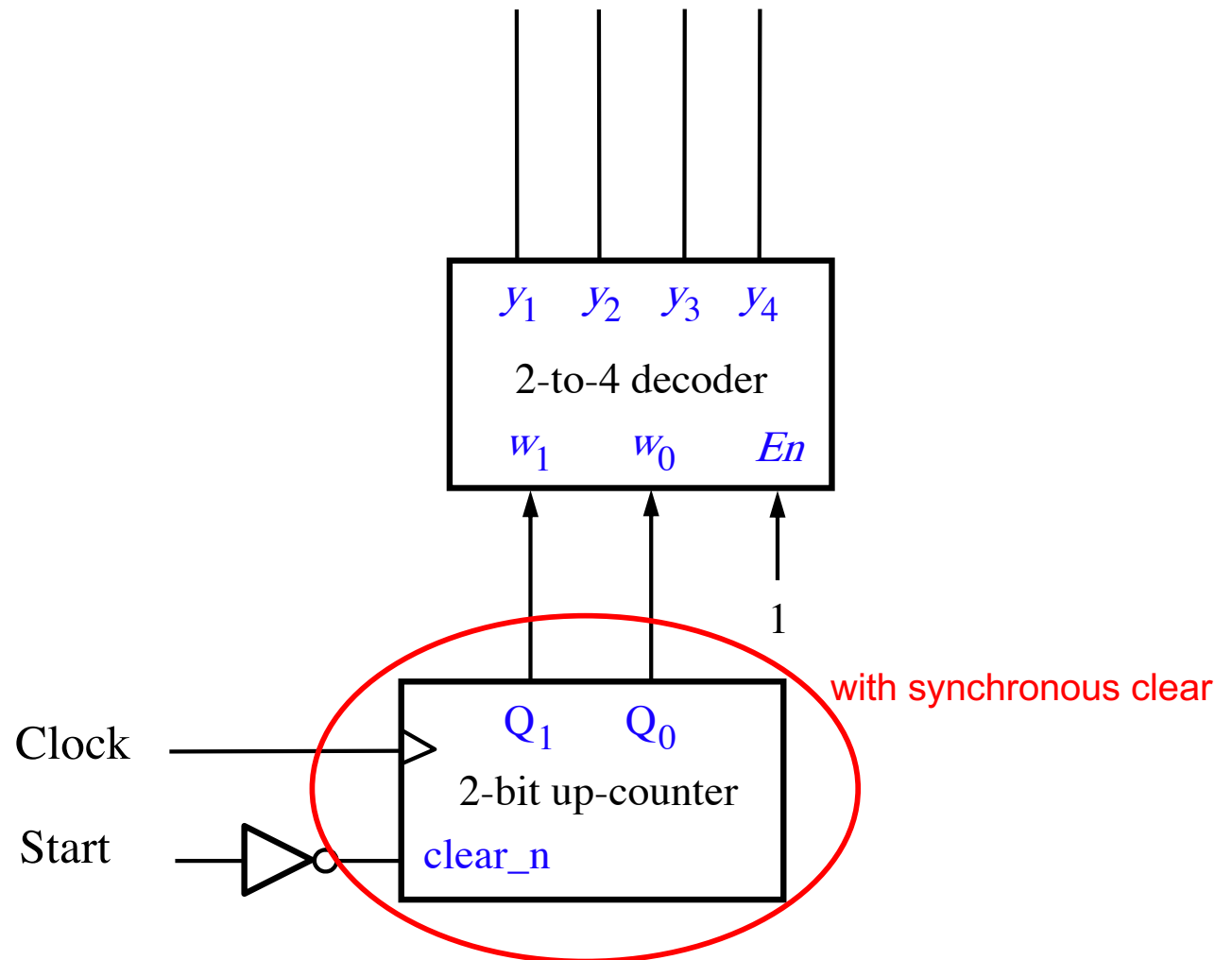
(this is done to be consistent with the previous example)

2-to-4 Decoder with Enable Input

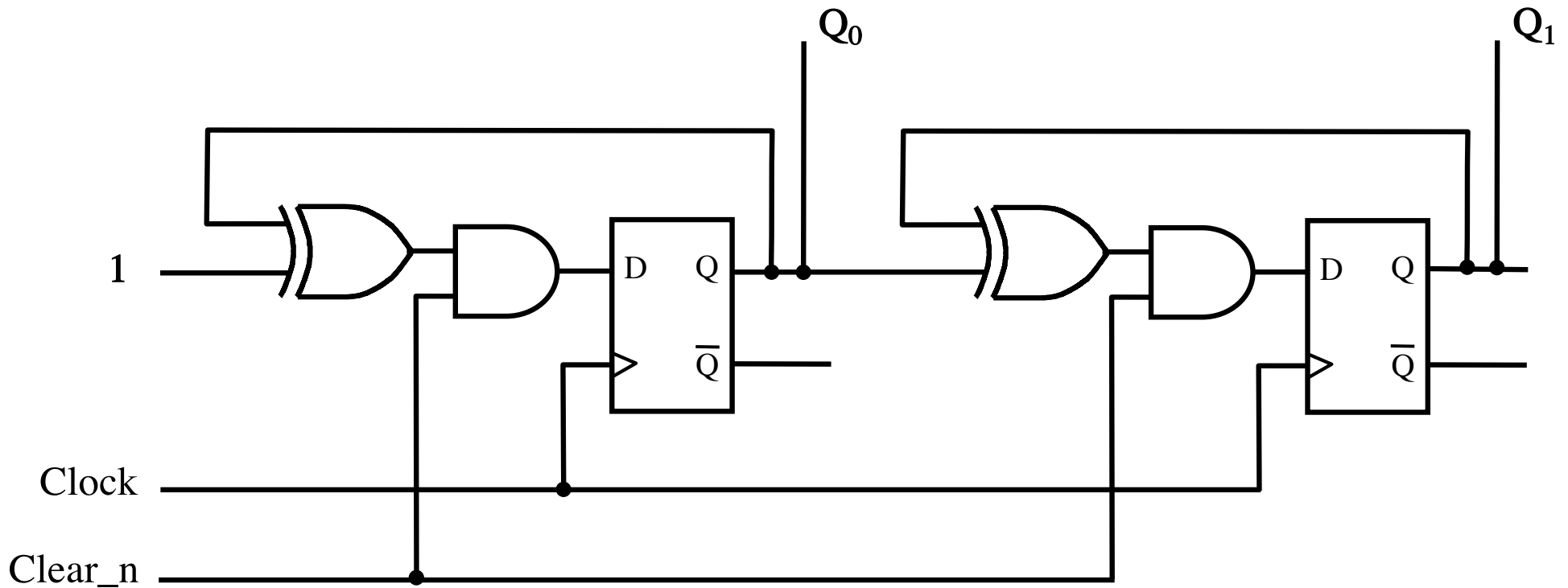


(always enabled in this example)

Alternative version of a 4-bit ring counter

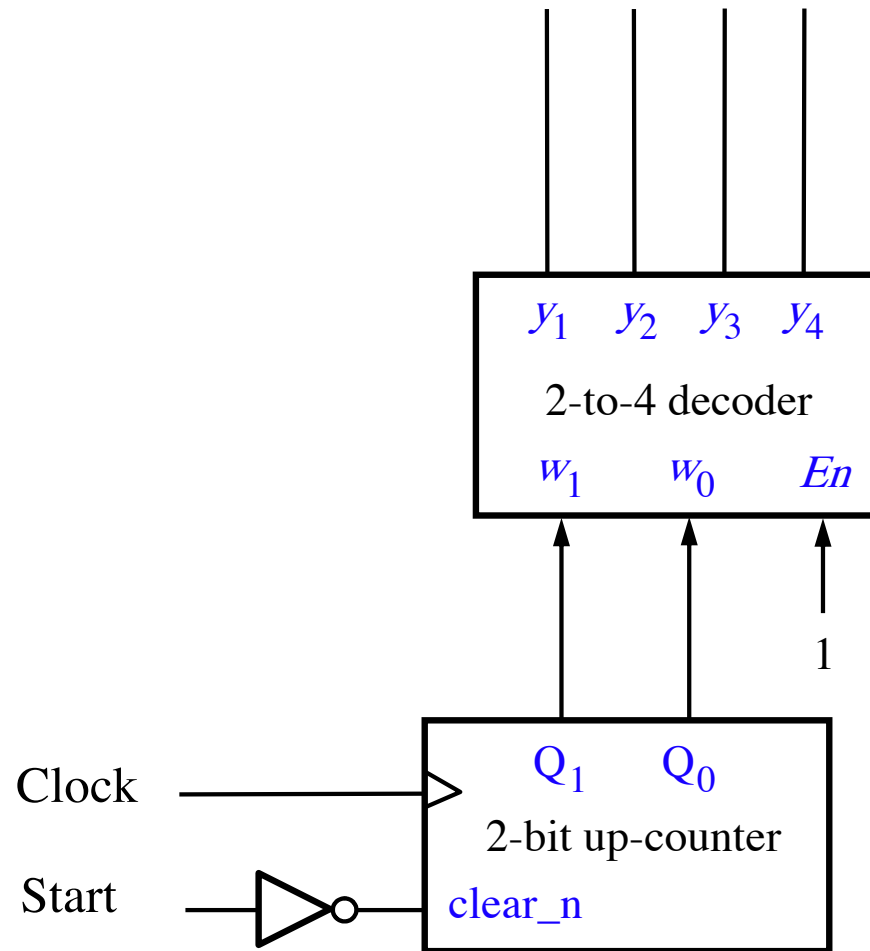


2-Bit Synchronous Up-Counter (with synchronous clear)

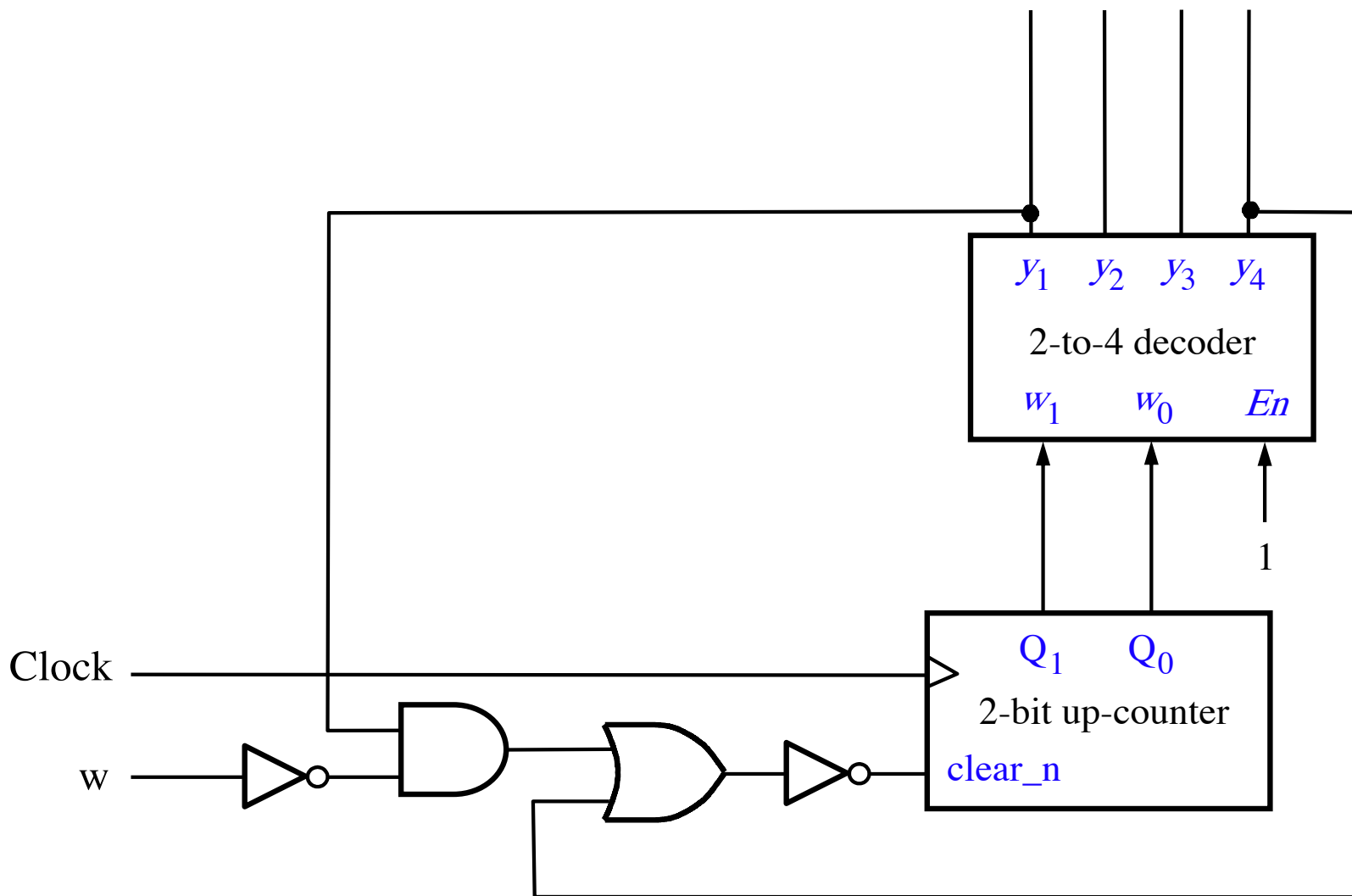


This counter can be cleared only on the positive clock edge.

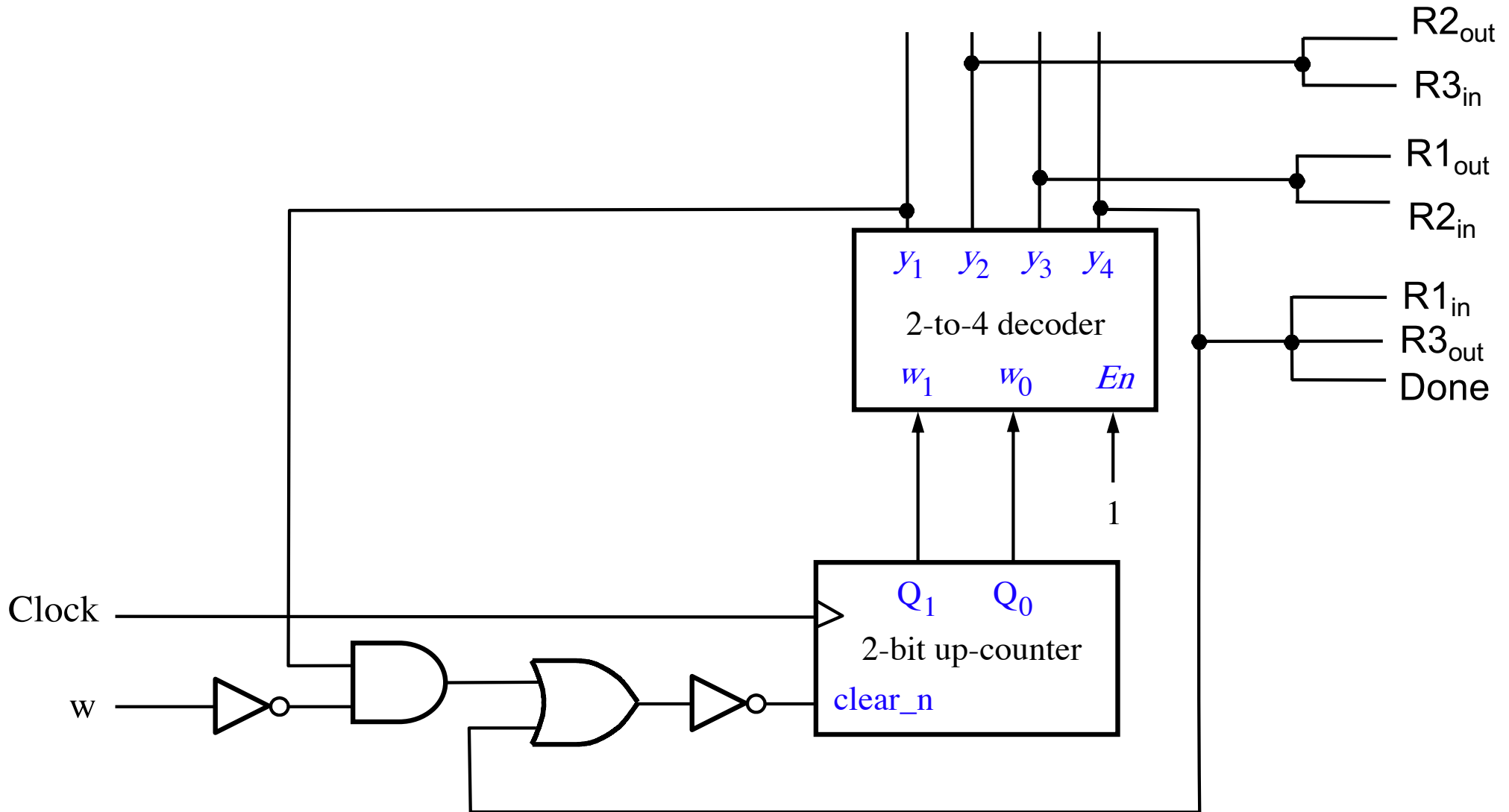
Let's Complete the Circuit Diagram



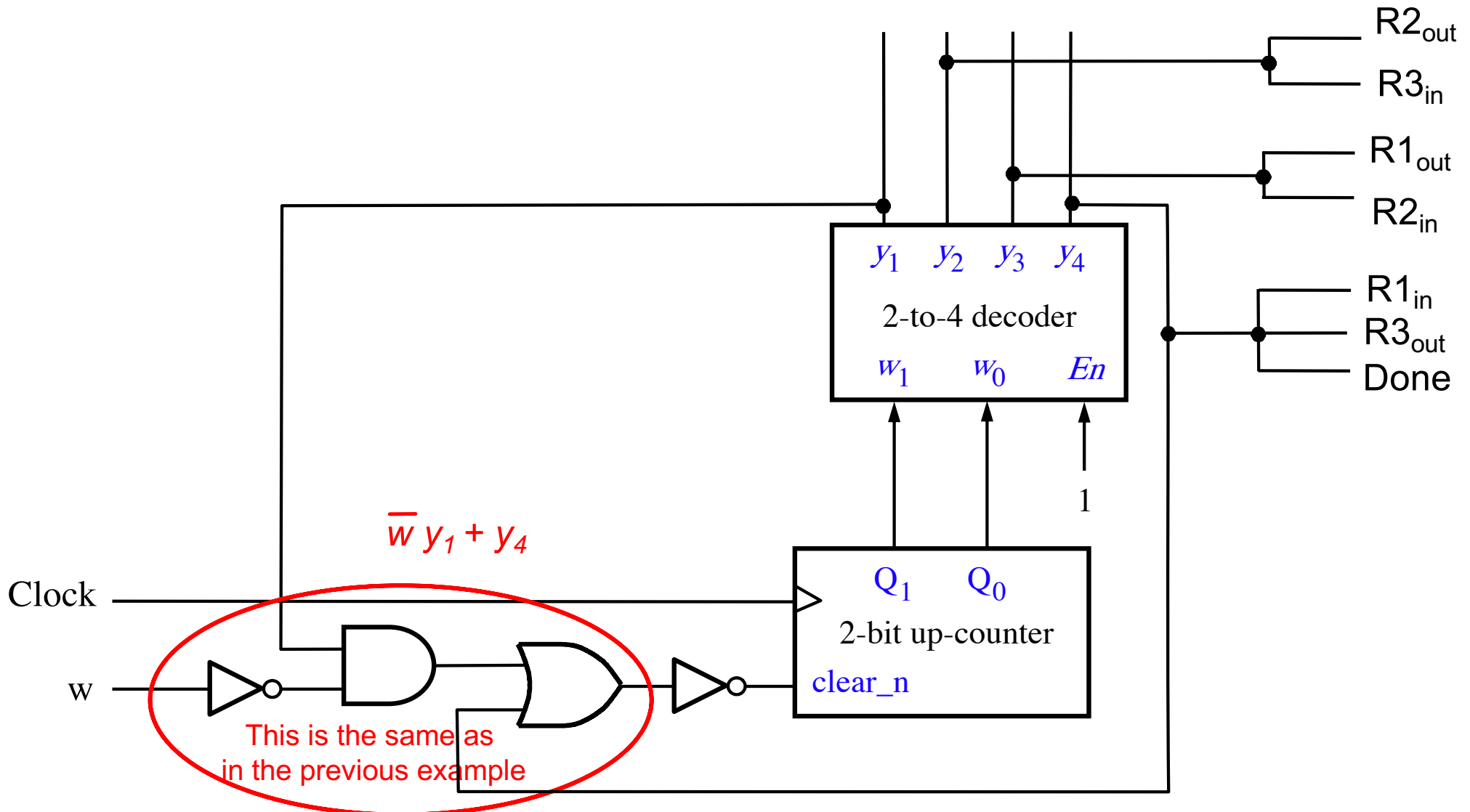
Let's Complete the Circuit Diagram



Let's Complete the Circuit Diagram



Let's Complete the Circuit Diagram

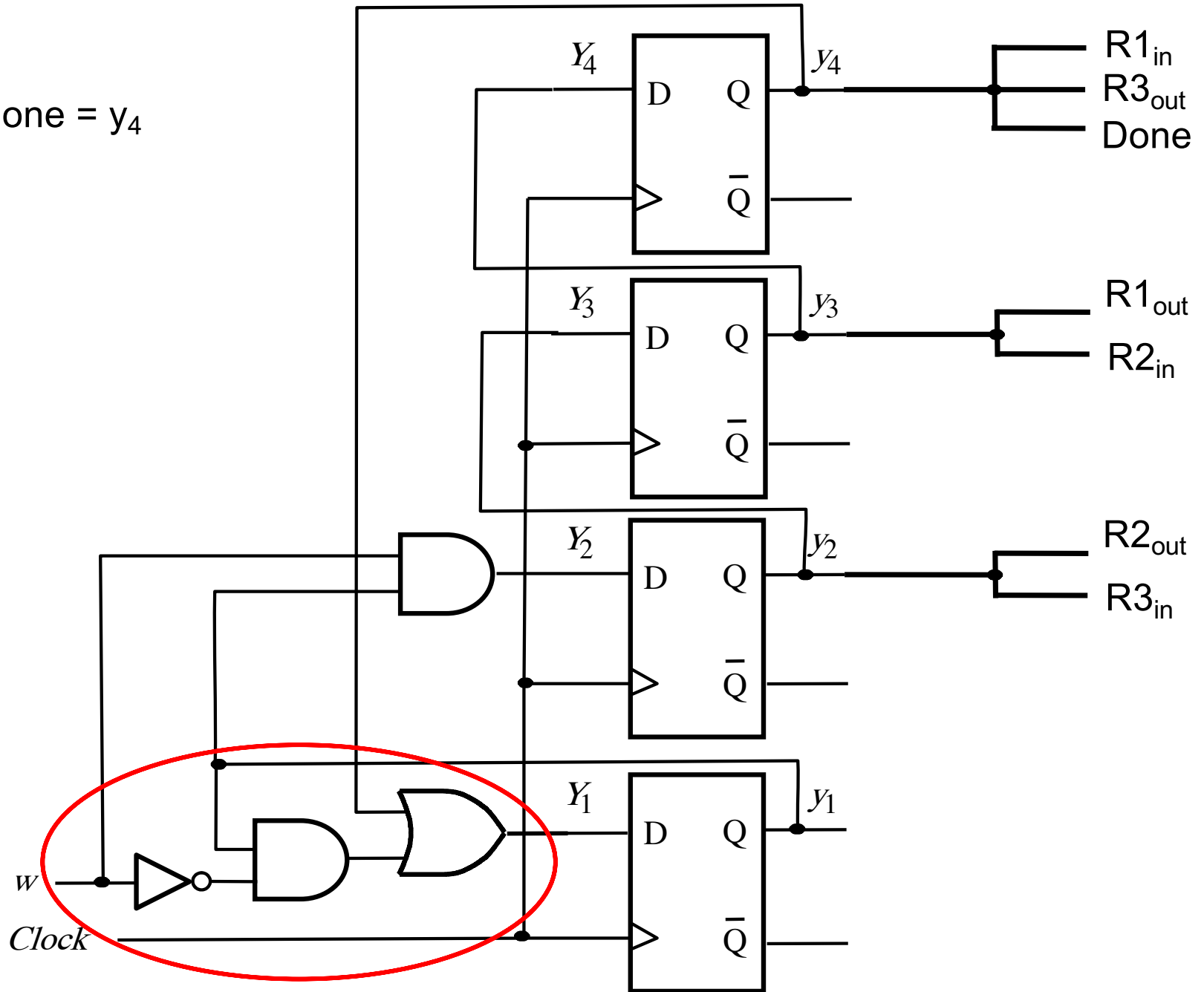


The Solution for Encoding #3

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

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

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



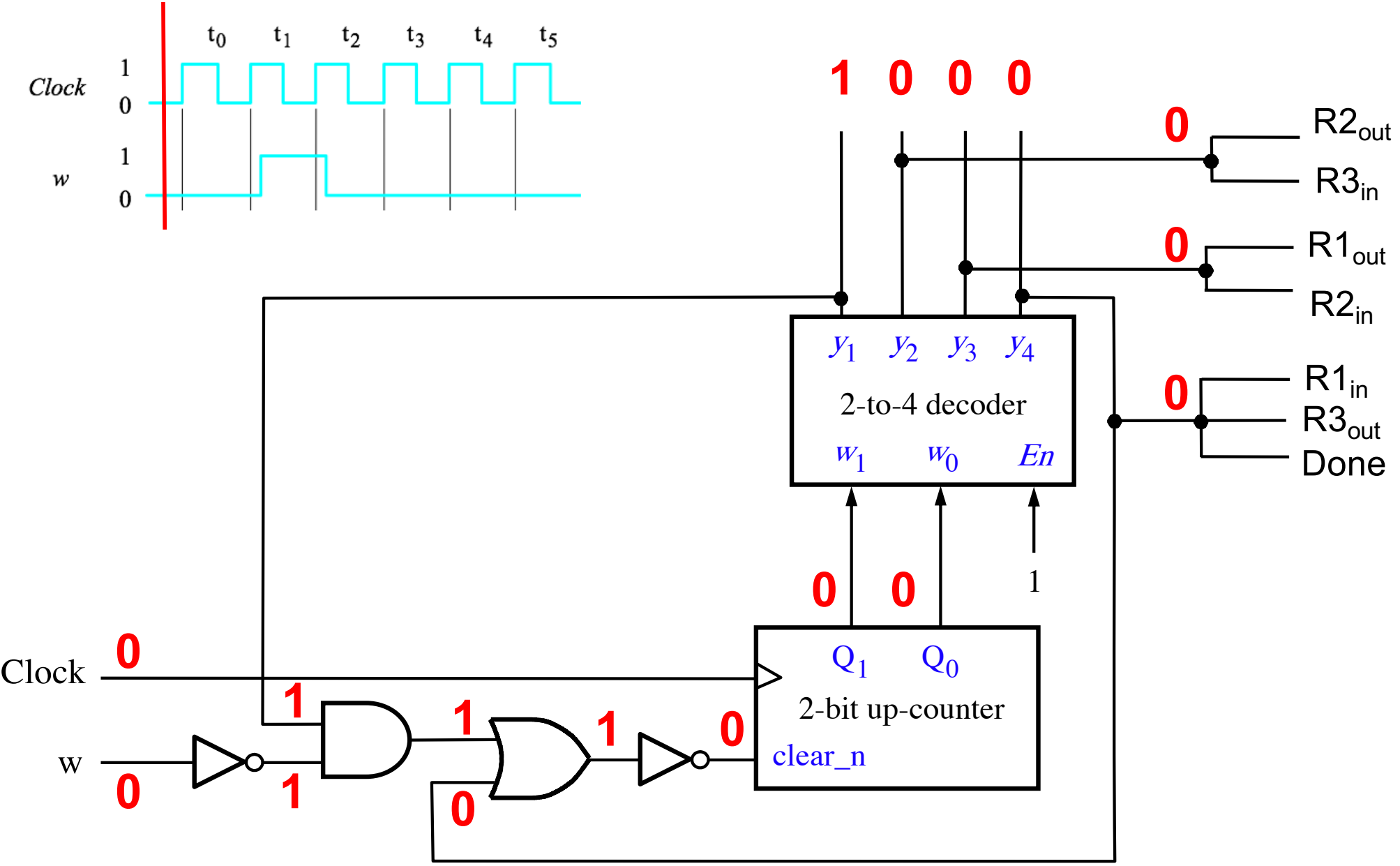
$$Y_1 = \bar{w} y_1 + y_4$$

$$Y_2 = w y_1$$

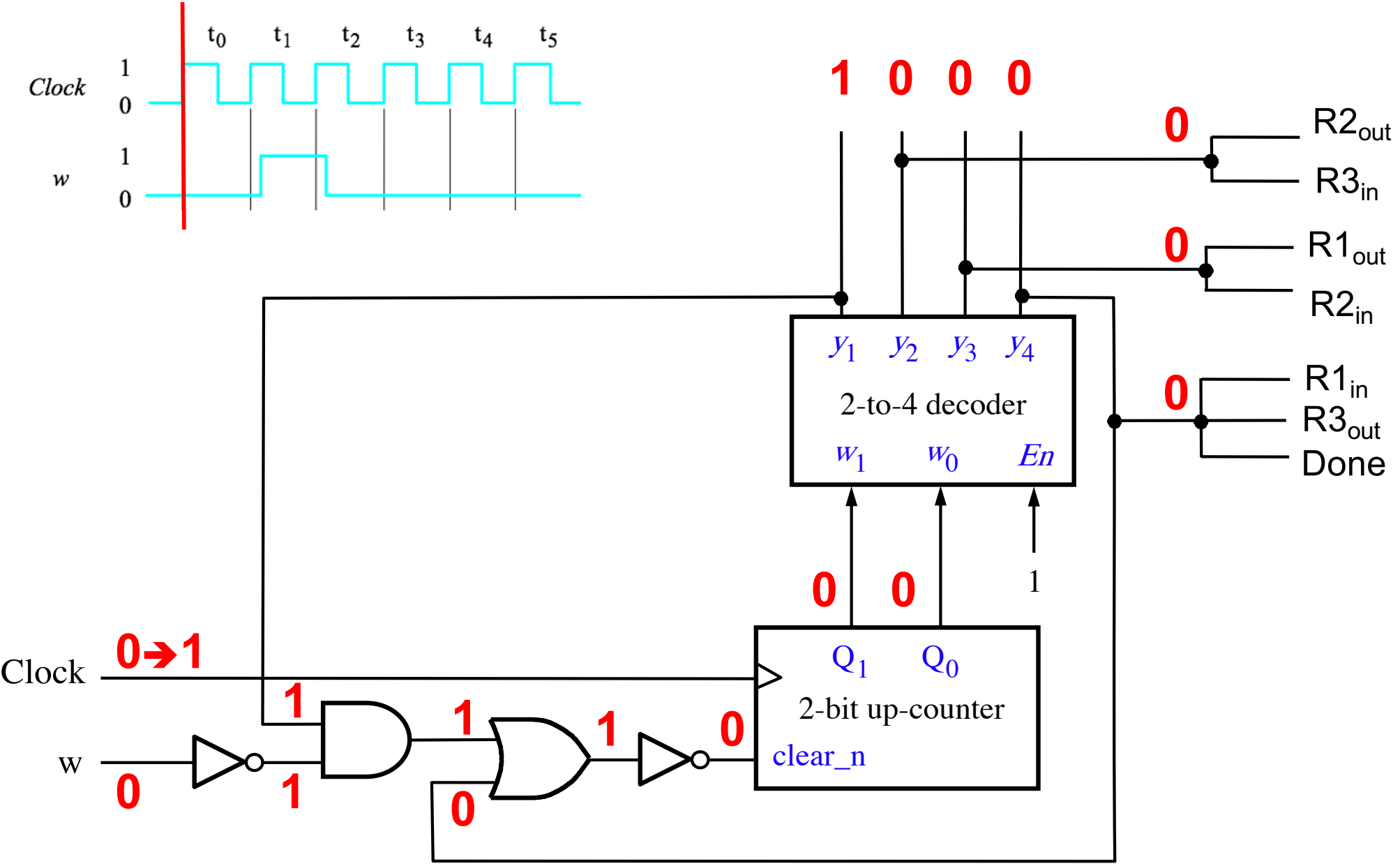
$$Y_3 = y_2$$

$$Y_4 = y_3$$

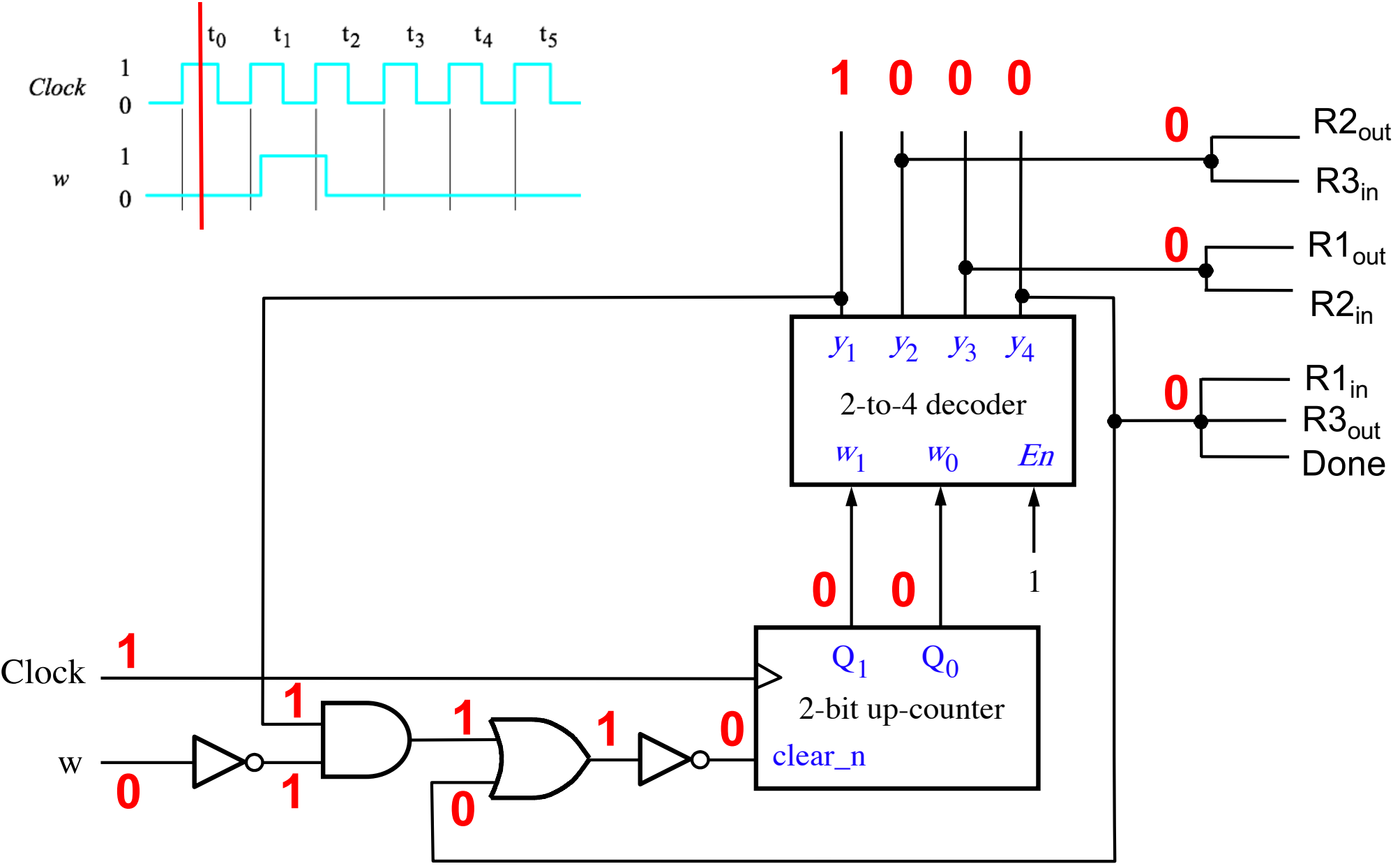
How Does It Work?



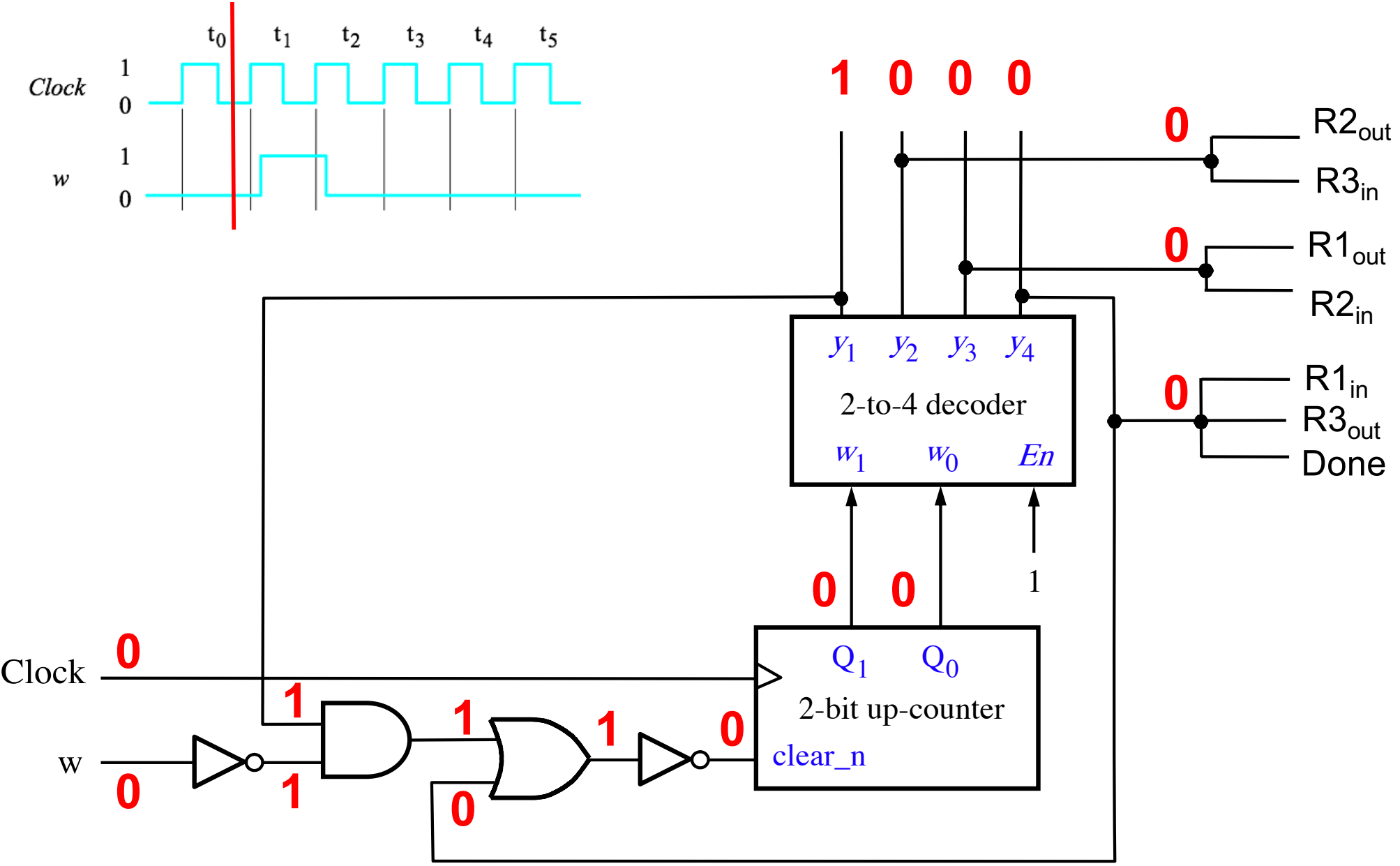
How Does It Work?



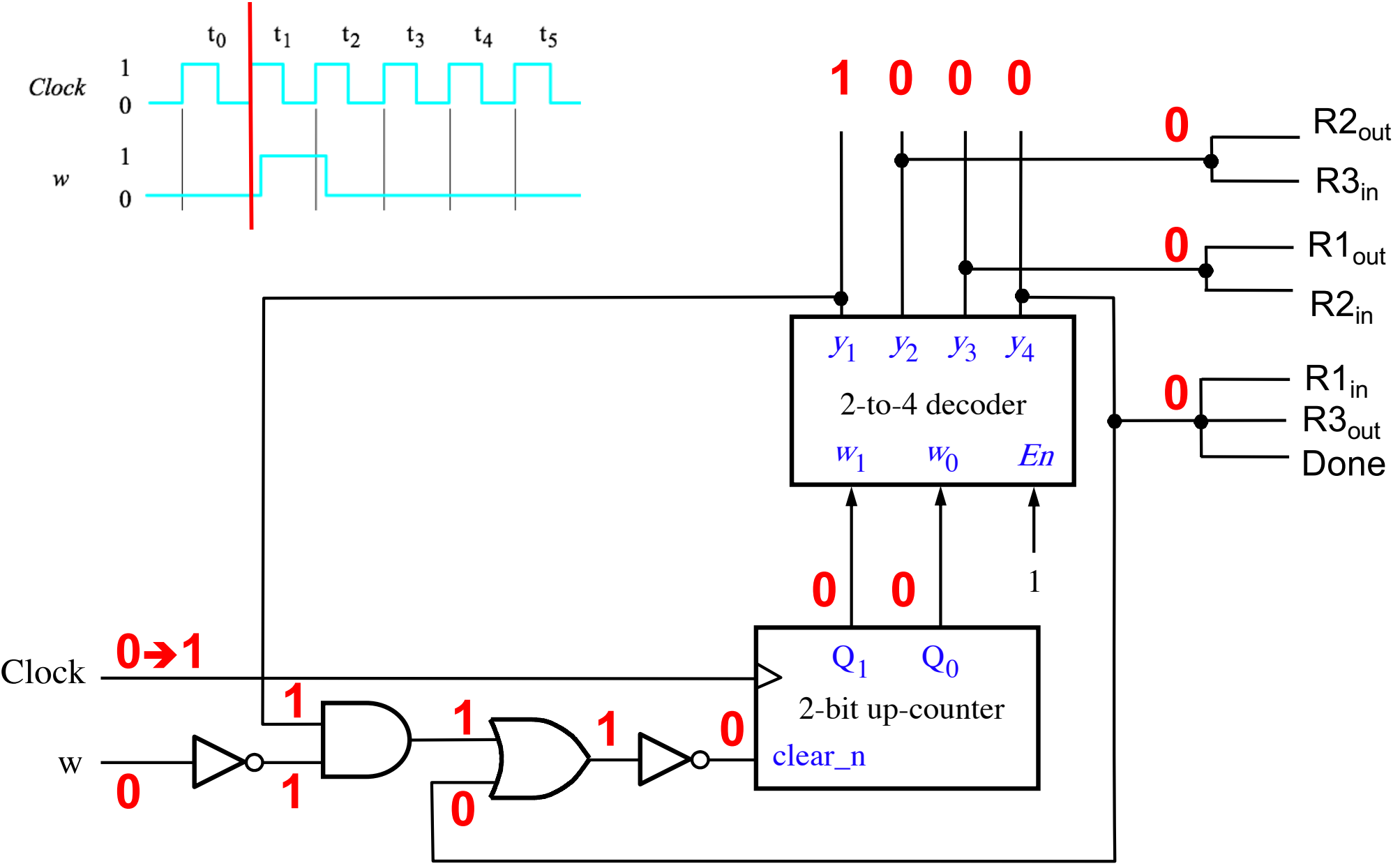
How Does It Work?



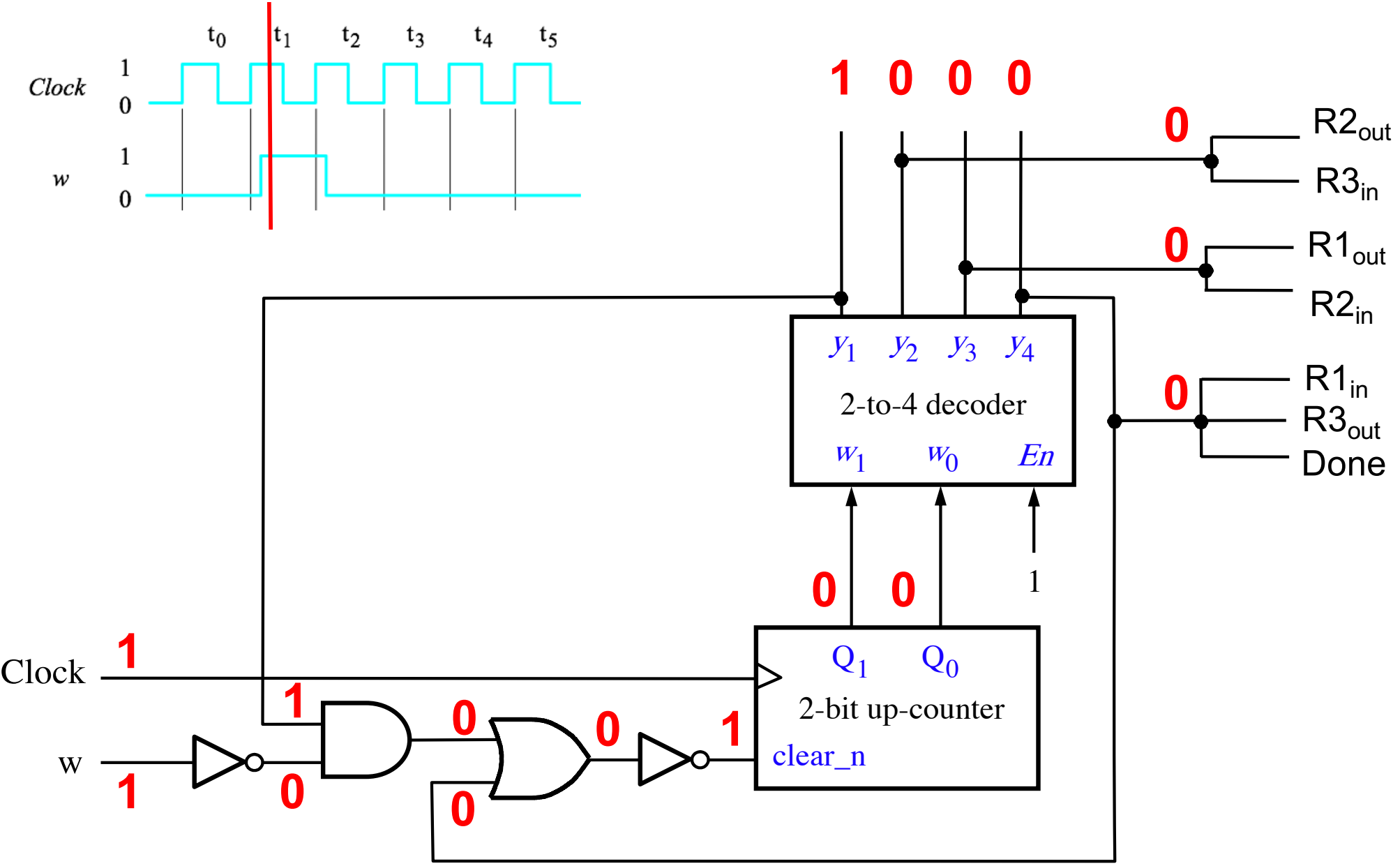
How Does It Work?



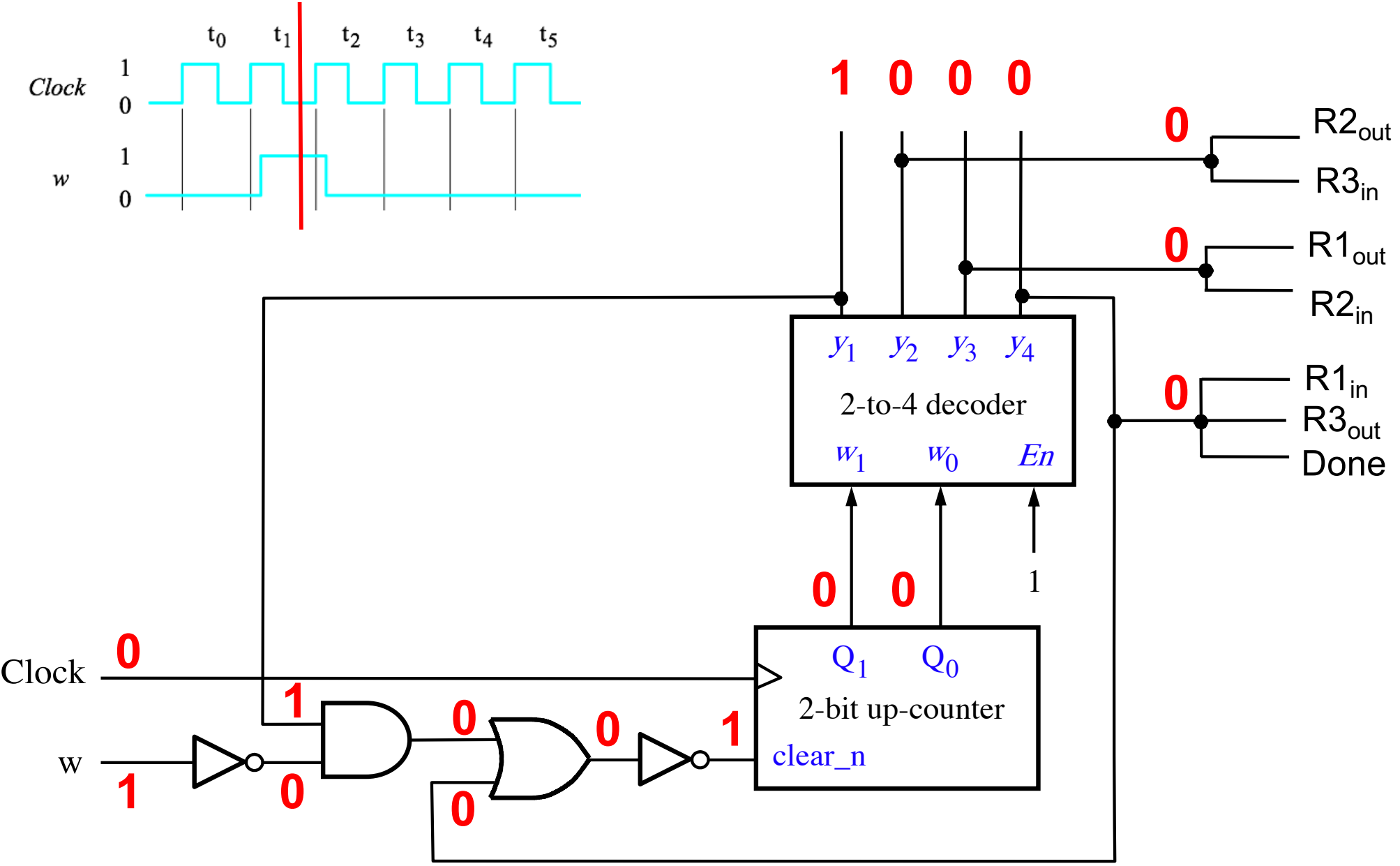
How Does It Work?



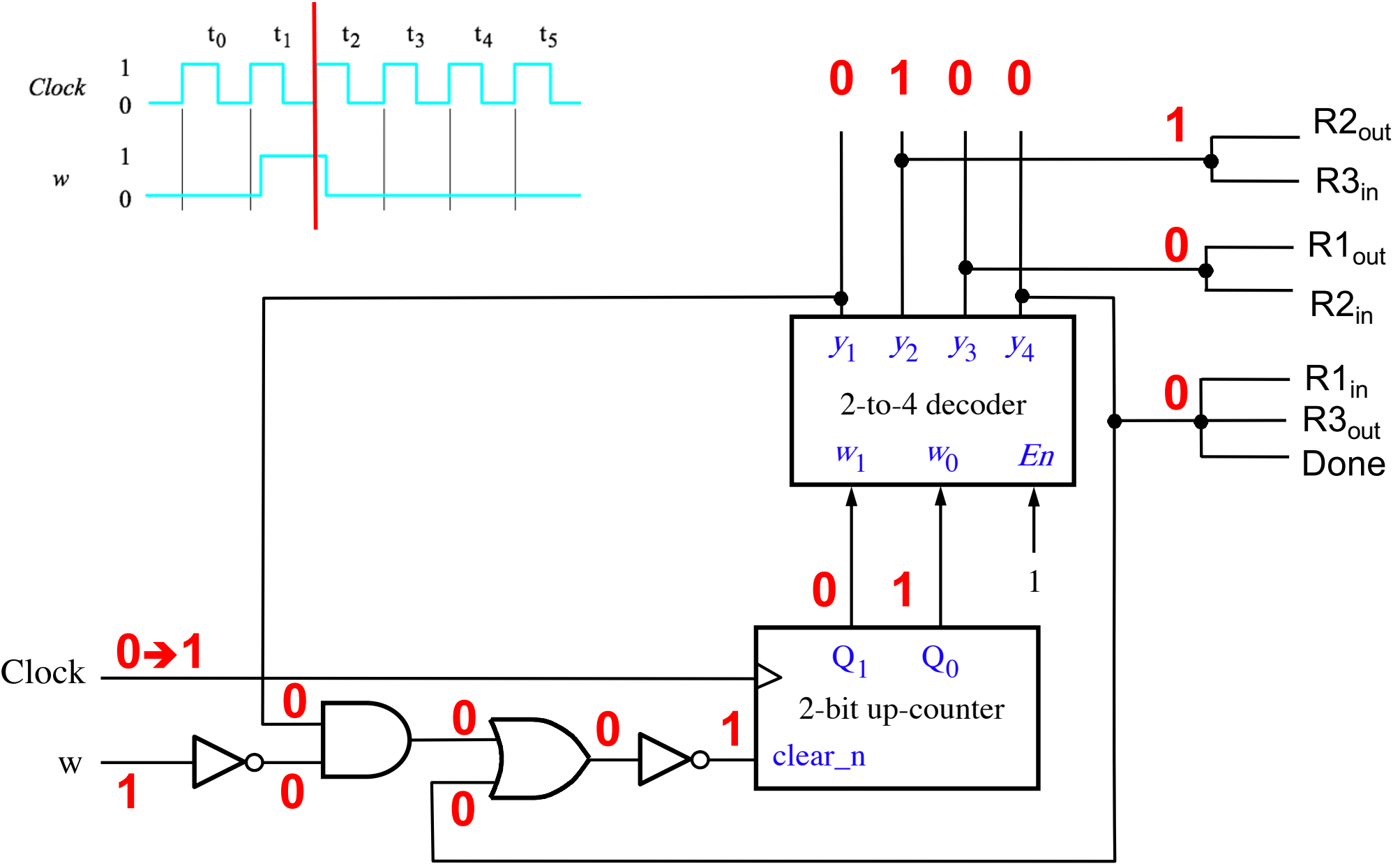
How Does It Work?



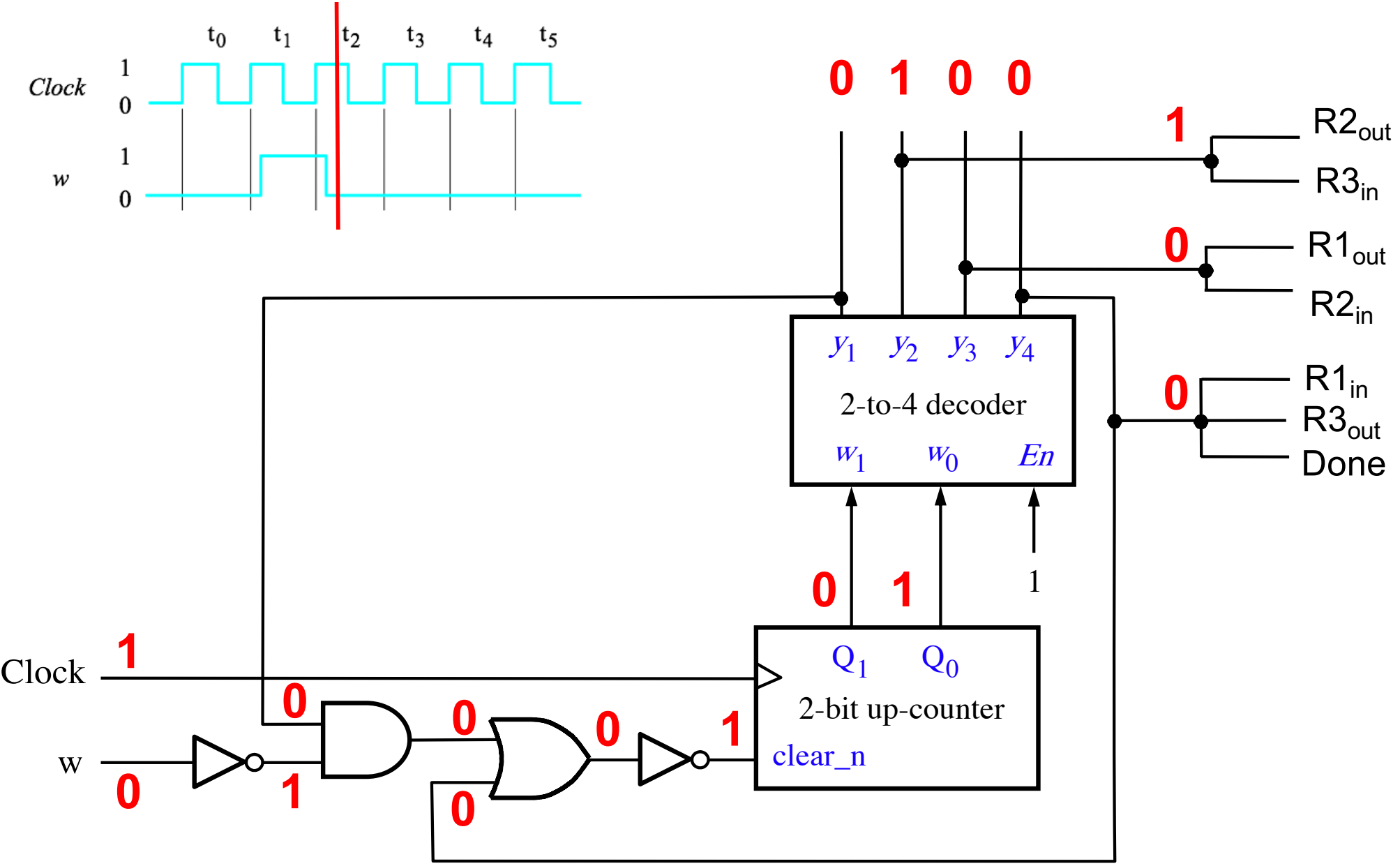
How Does It Work?



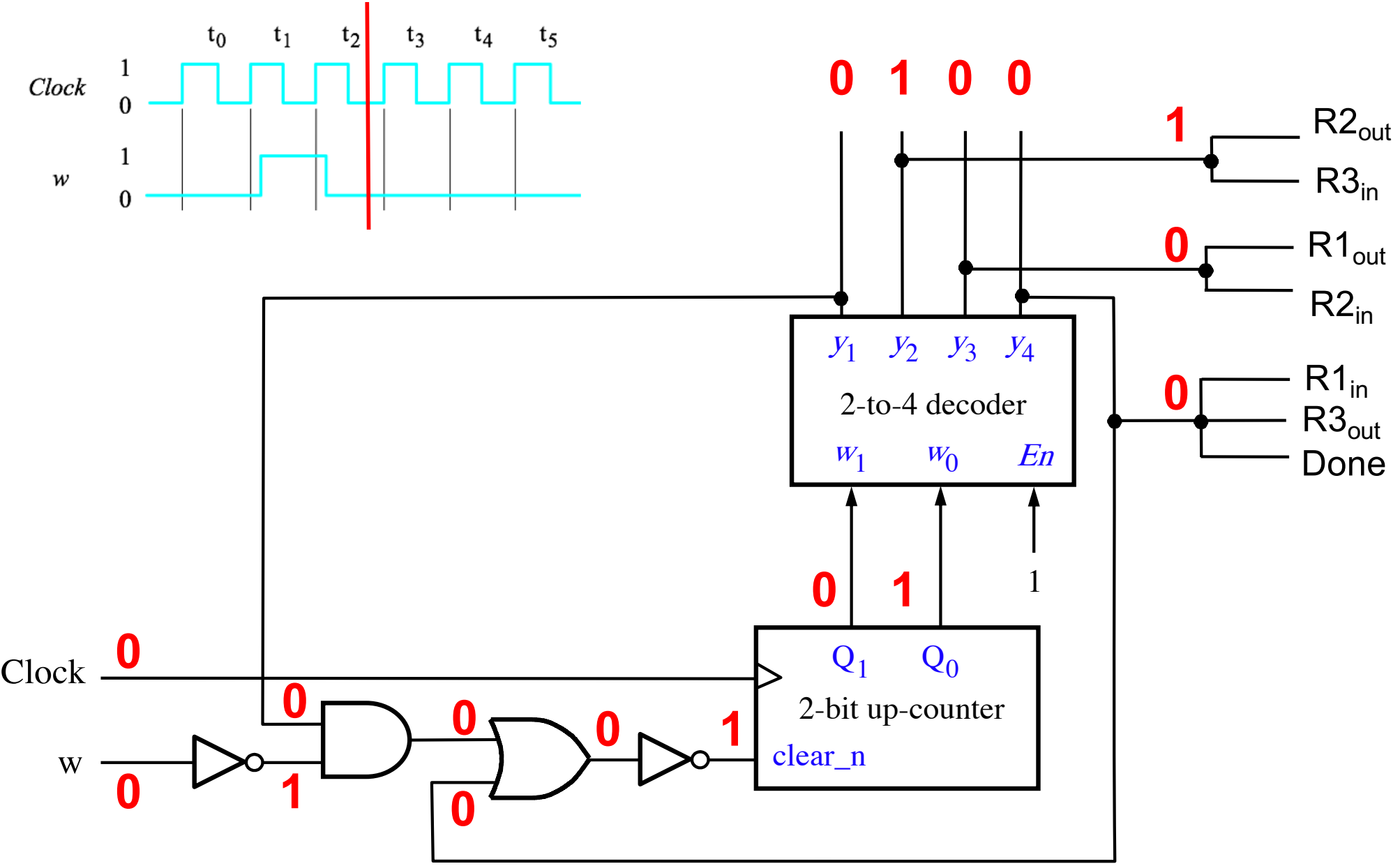
How Does It Work?



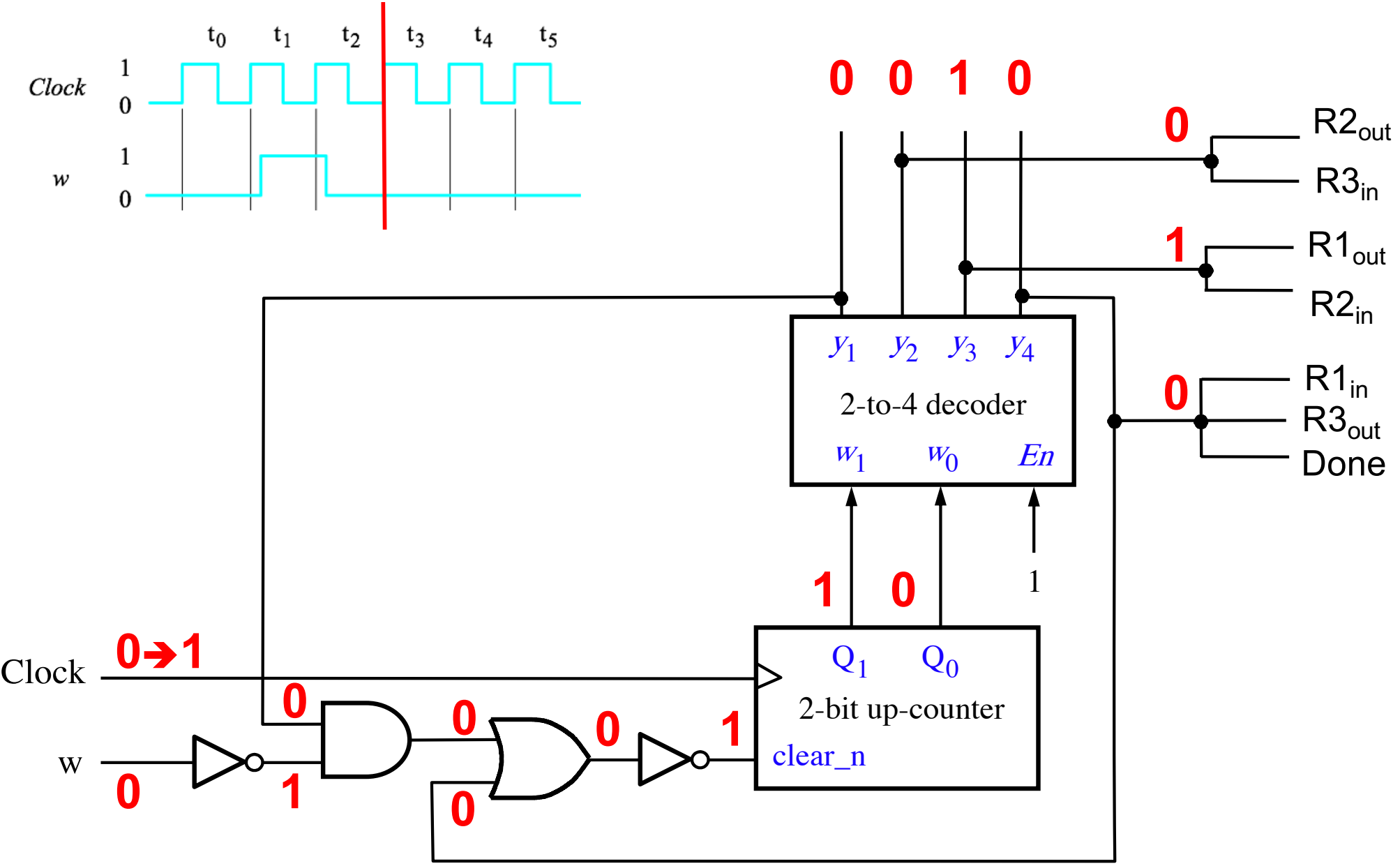
How Does It Work?



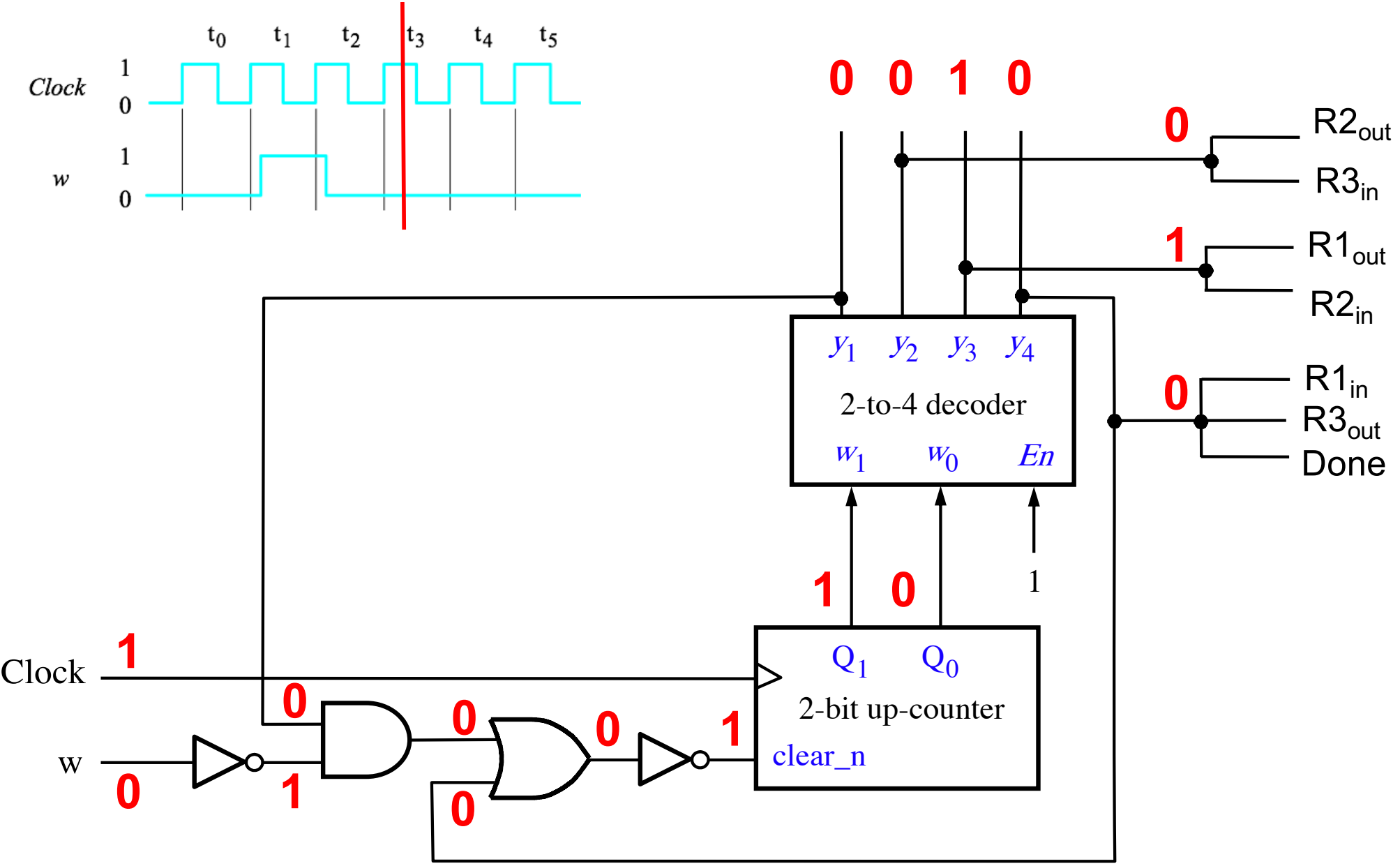
How Does It Work?



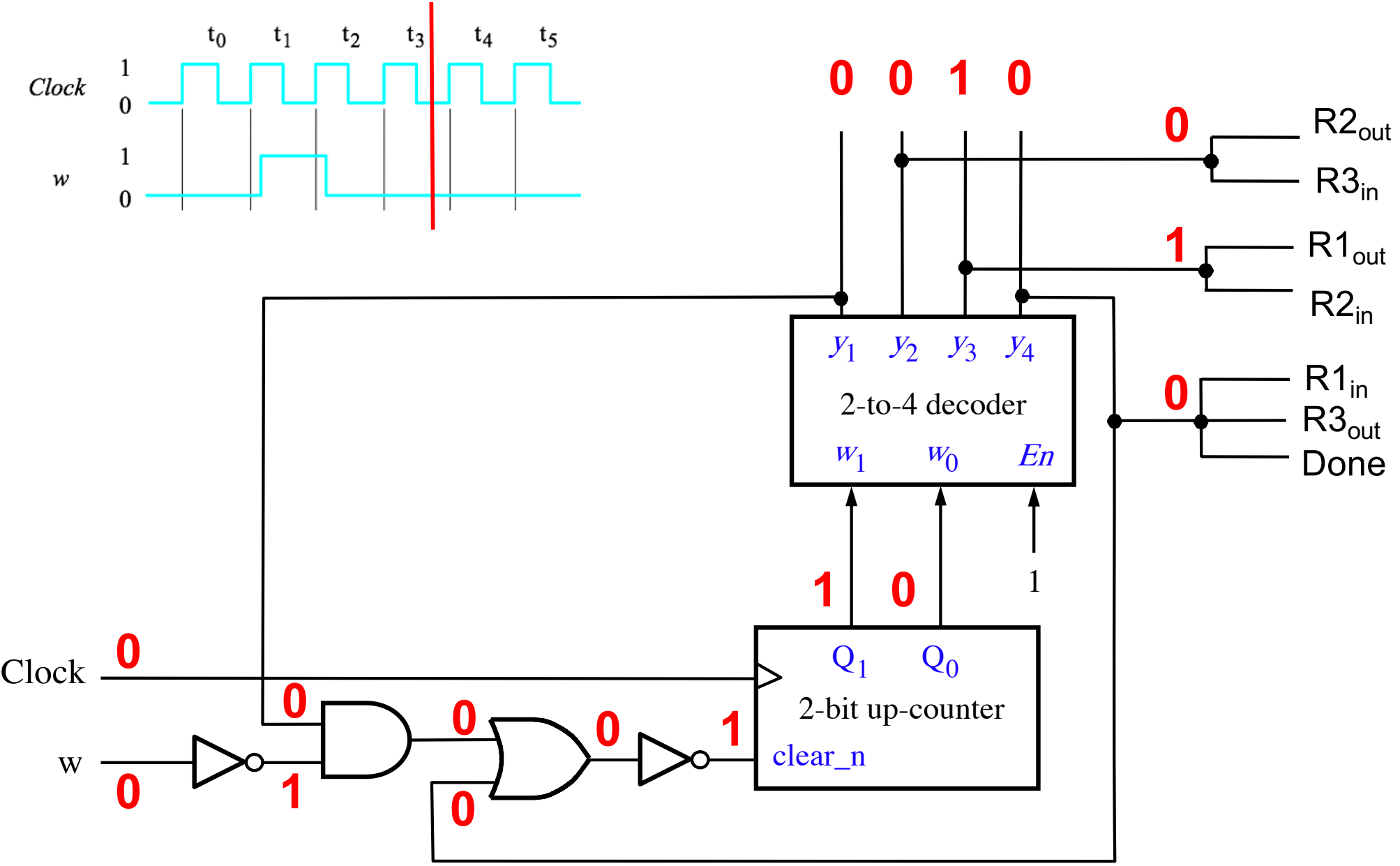
How Does It Work?



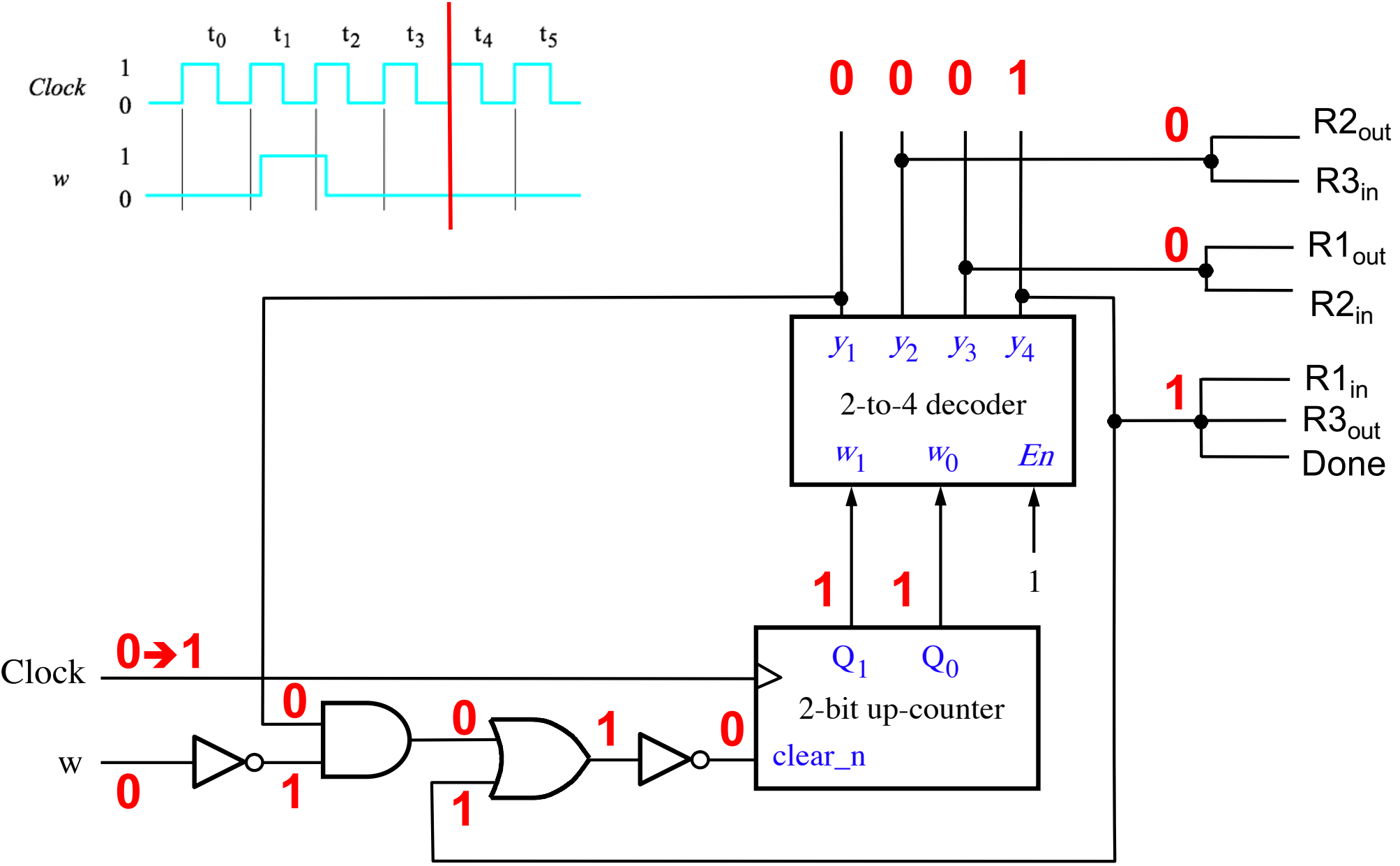
How Does It Work?



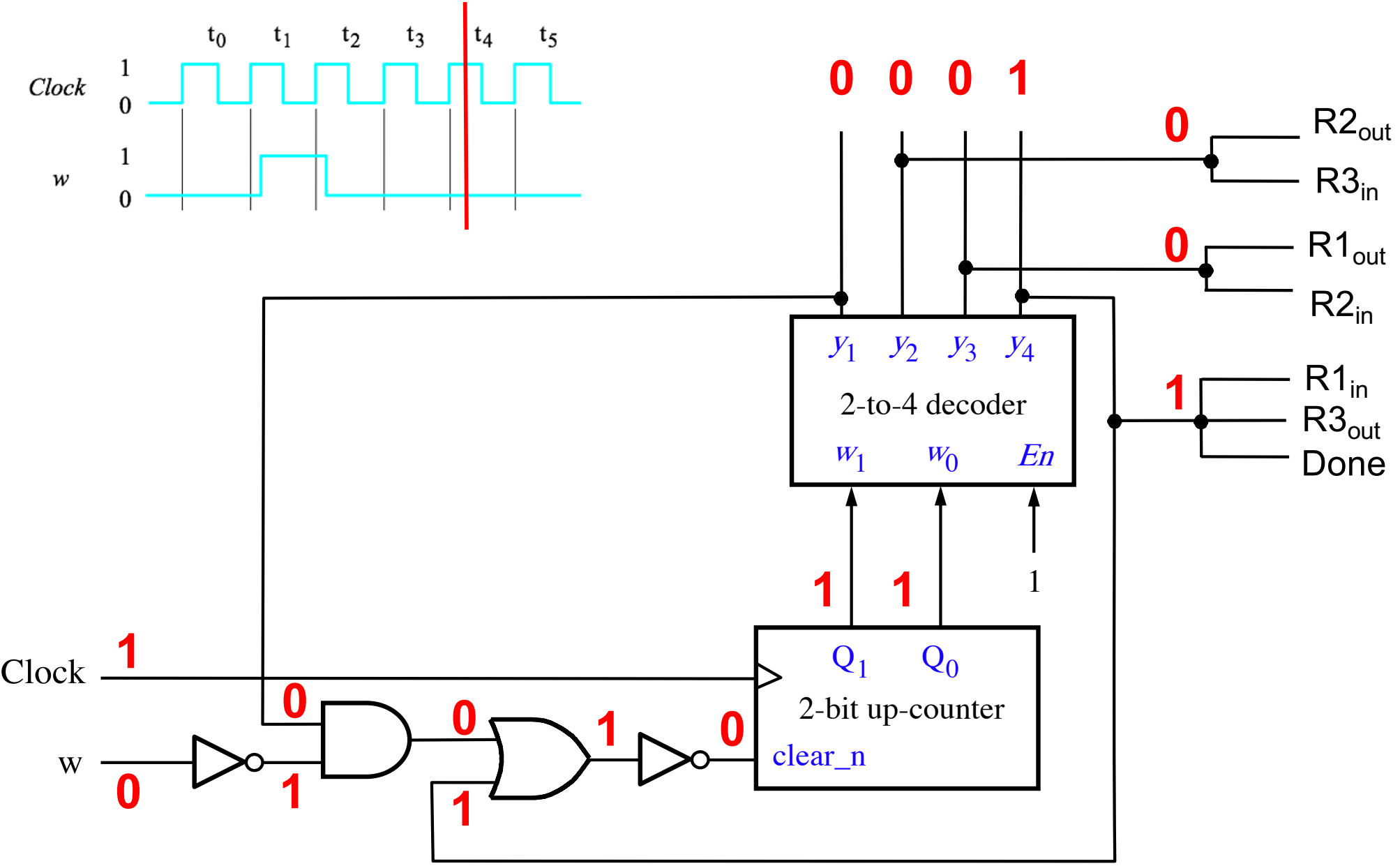
How Does It Work?



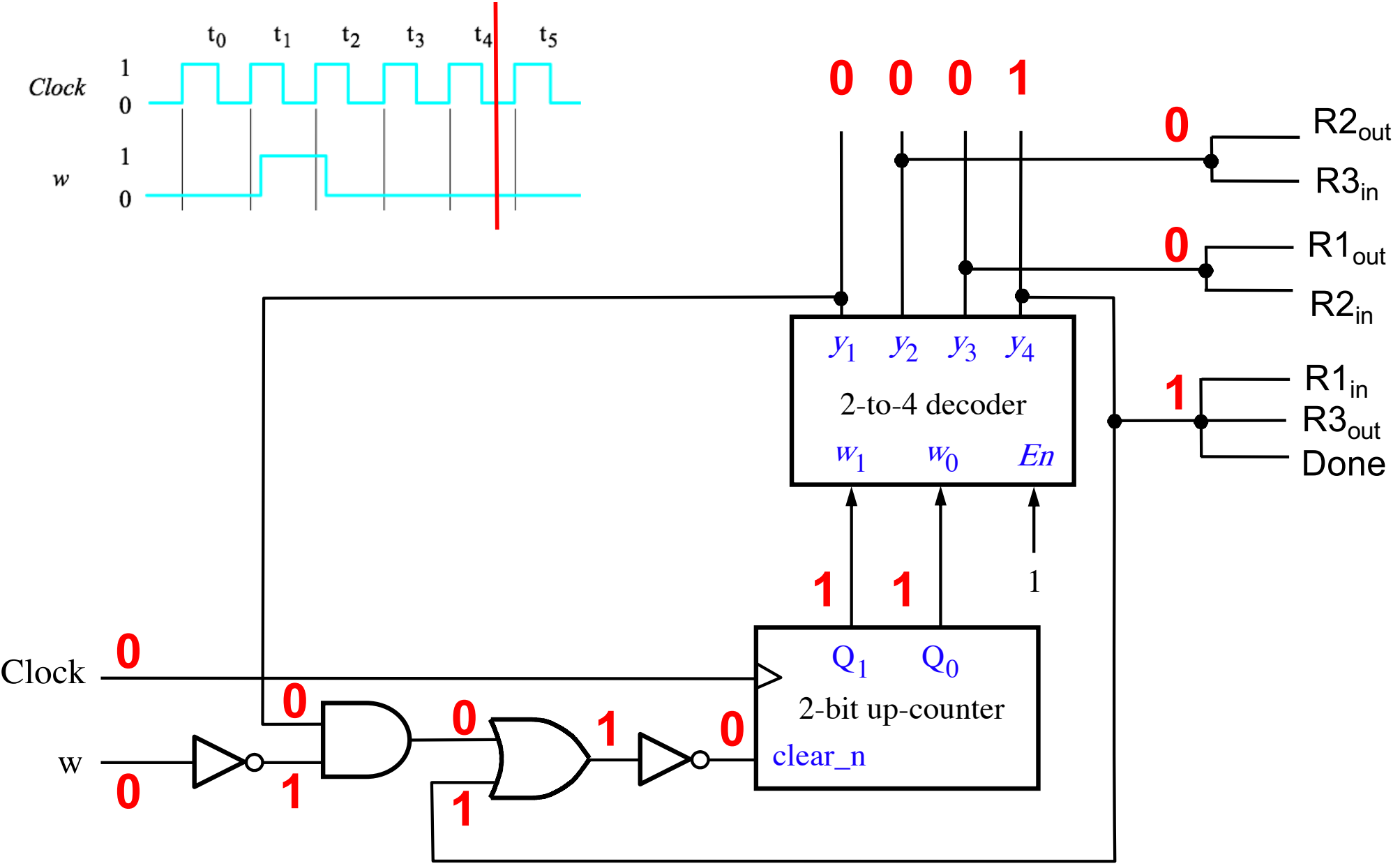
How Does It Work?



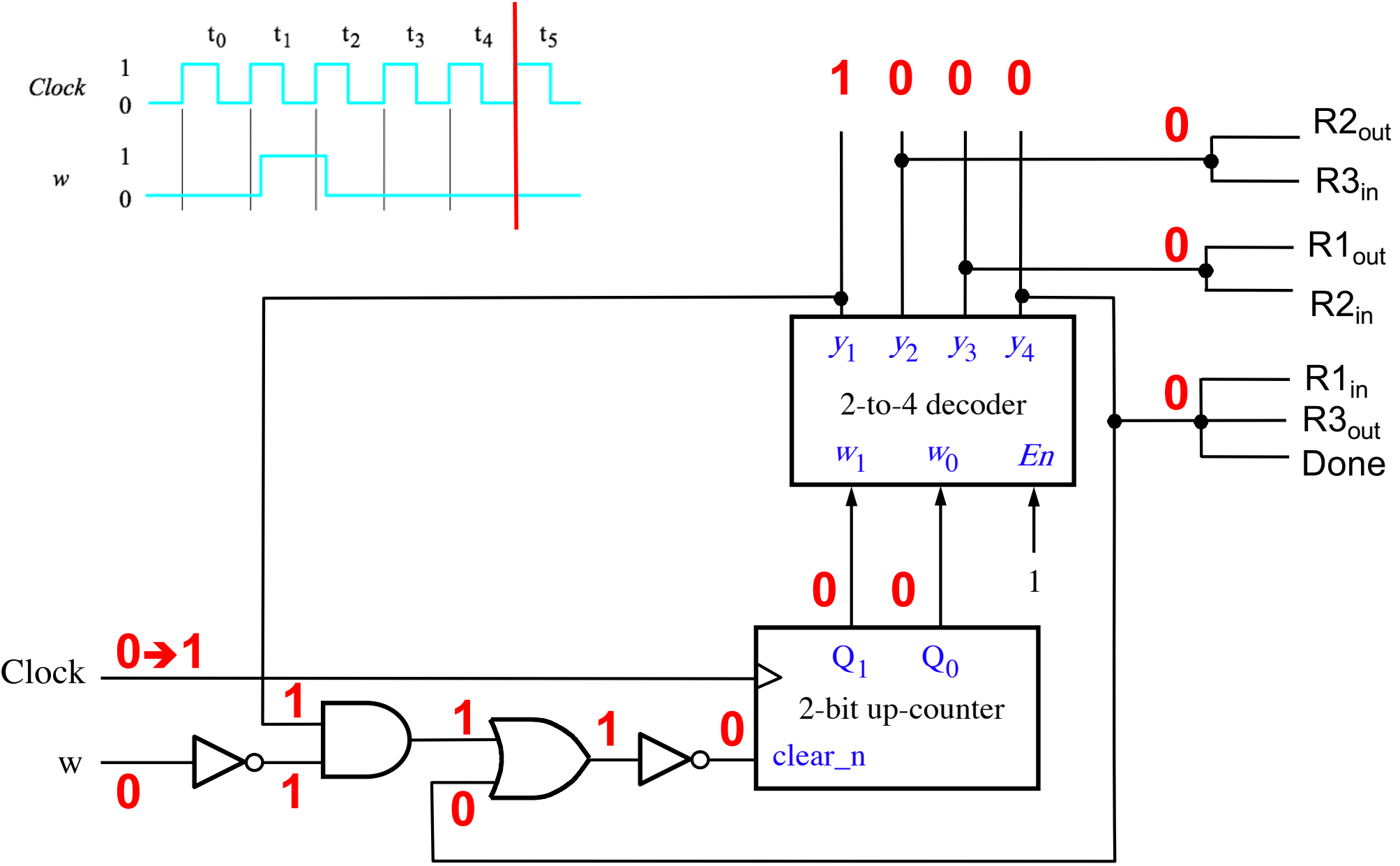
How Does It Work?



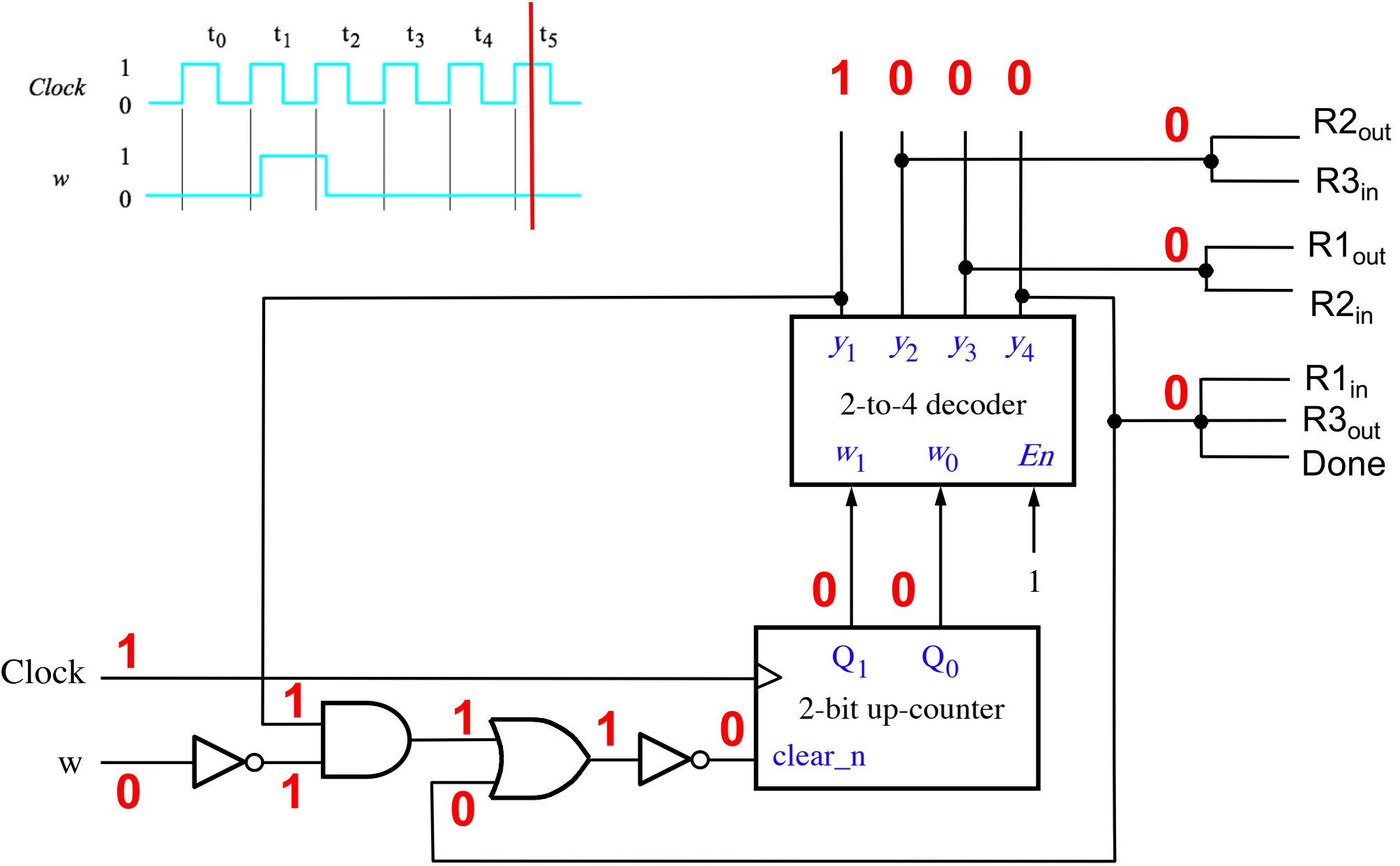
How Does It Work?



How Does It Work?



How Does It Work?



Questions?

THE END