

An Introduction to the Logic  
of  
Silicon Chips

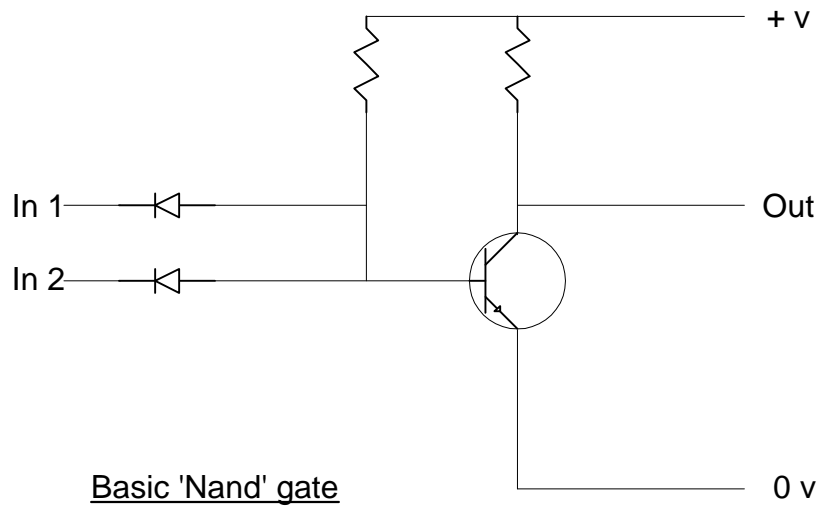
Here is a photo of a typical silicon chip, taken alongside the tip of my little finger. Modern chips can be made a good deal smaller than the one shown - just a few millimetres across. The chip comprises a (very small) piece of semiconductor material encapsulated in a plastic package, which has a number of metal terminal pins connected to the semiconductor and leading out of the package. The chip in the photo has 14 pins.

Most silicon chips have a set of 'input' pins, 'output' pins and power-supply pins, and the function of the chip is to control the voltages on the output terminals in response to the voltages measured on the input terminals. Chips are manufactured in thousands of different types, and a key difference between these is the relationship the chip effects between its inputs and outputs.

Digital chips are primarily associated with inputs and outputs that can only take two legitimate values - traditionally called 'nought' and 'one', though the actual voltages may well not be zero and one. But they will have two distinct values and you should not expect to see an output voltage which is not one of these two acceptable values. Analogue chips more commonly deal with voltages that can take an arbitrary (infinite) number of legitimate values between a lower and upper limit.

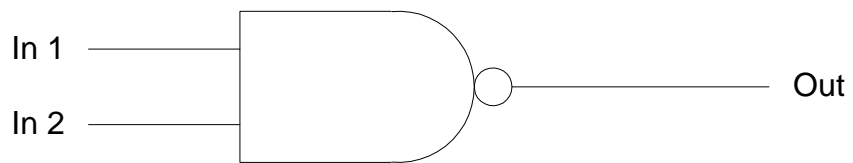


This presentation will cover some simple digital chips, explain some of their internal logic, and show how repeated use of the simplest components can build towards a system of astonishing complexity.

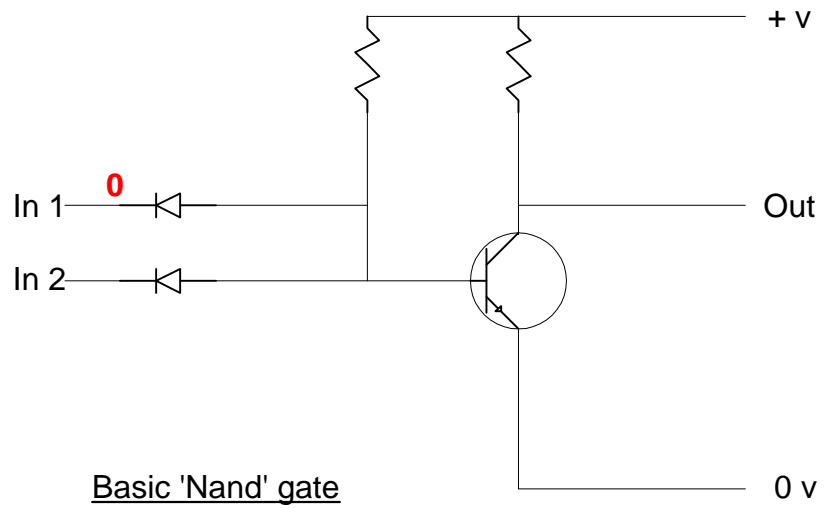


In1	In2	Out
0	0	1
1	0	1
0	1	1
1	1	0

Truth Table

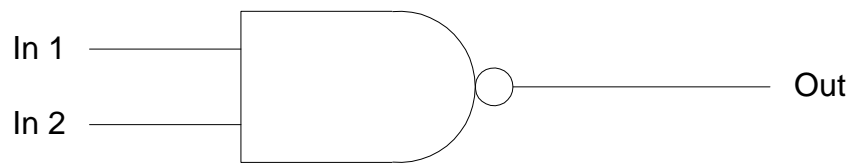


Standard Logic Symbol

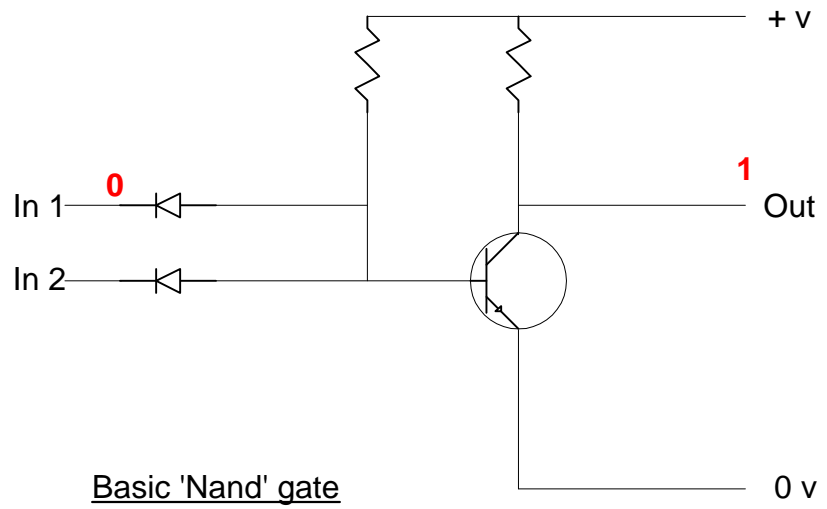


In1	In2	Out
0	0	1
1	0	1
0	1	1
1	1	0

Truth Table

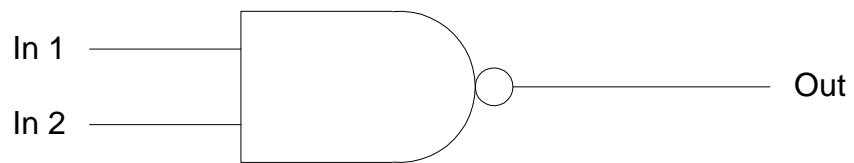


Standard Logic Symbol

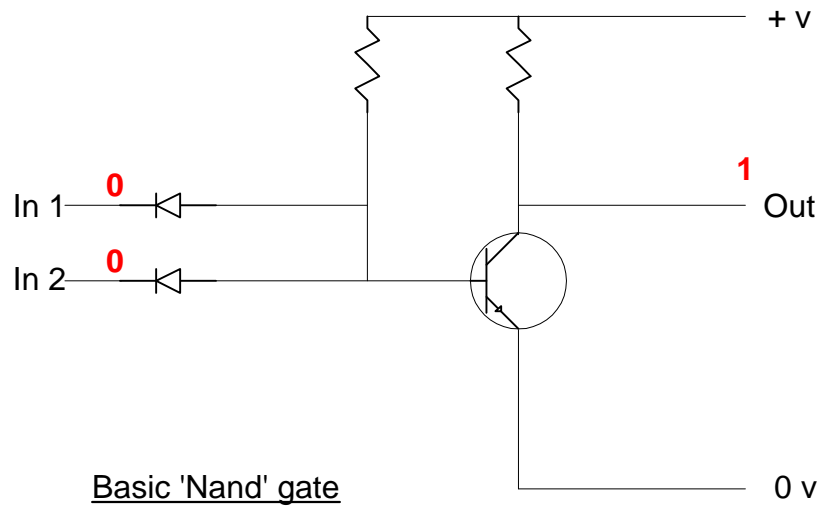


In1	In2	Out
0	0	1
1	0	1
0	1	1
1	1	0

Truth Table

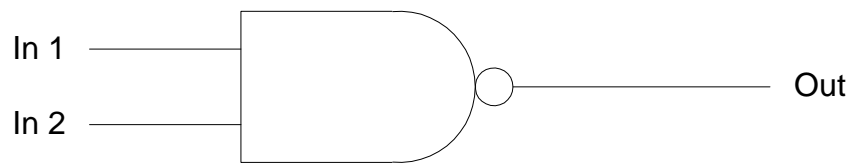


Standard Logic Symbol

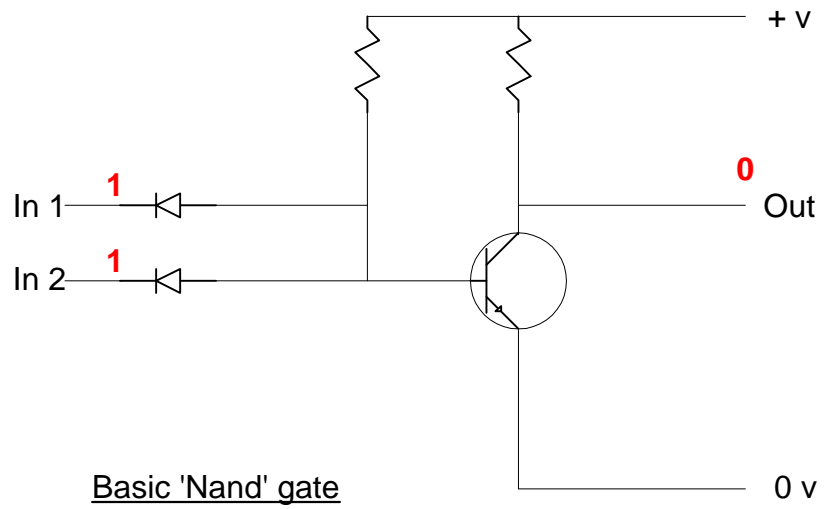


In1	In2	Out
0	0	1
1	0	1
0	1	1
1	1	0

Truth Table

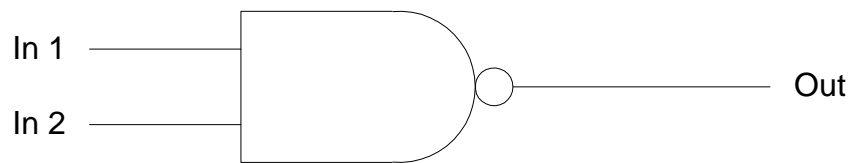


Standard Logic Symbol



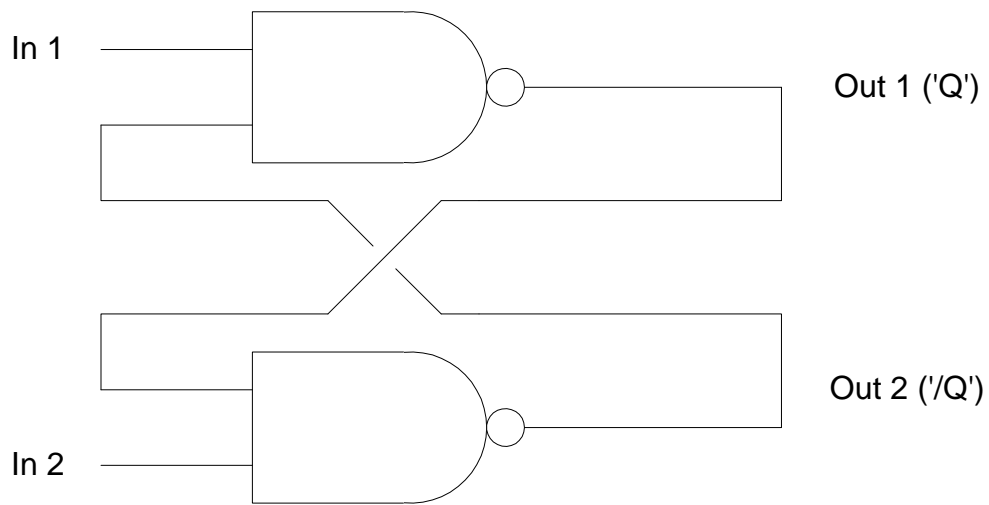
In1	In2	Out
0	0	1
1	0	1
0	1	1
1	1	0

Truth Table



Standard Logic Symbol

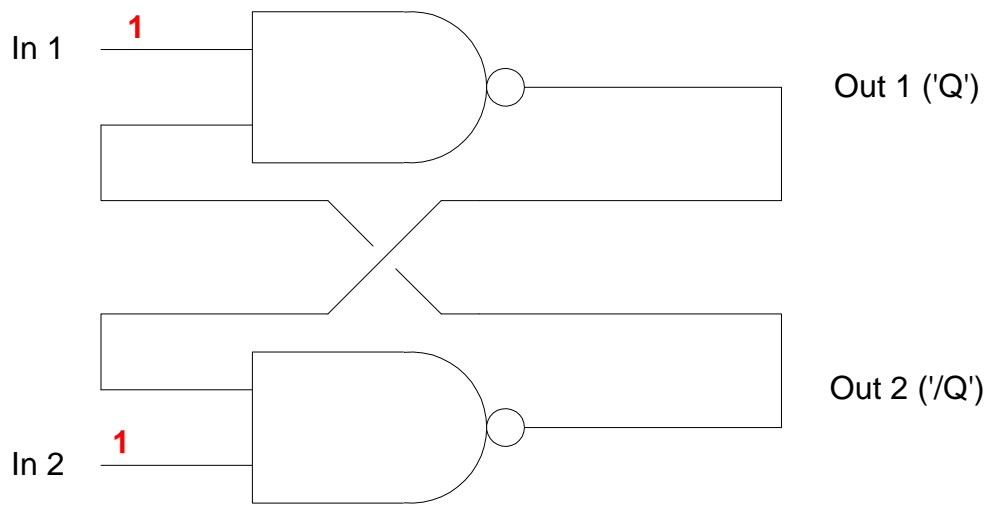




Basic Latch

In1	In2	Q	/Q
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	/Q

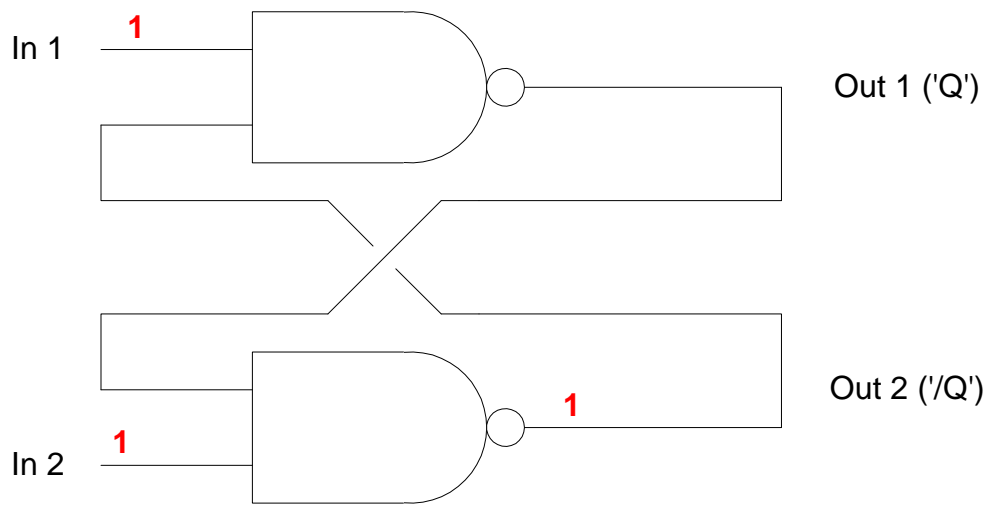
Truth Table



Basic Latch

In1	In2	Q	/Q
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	/Q

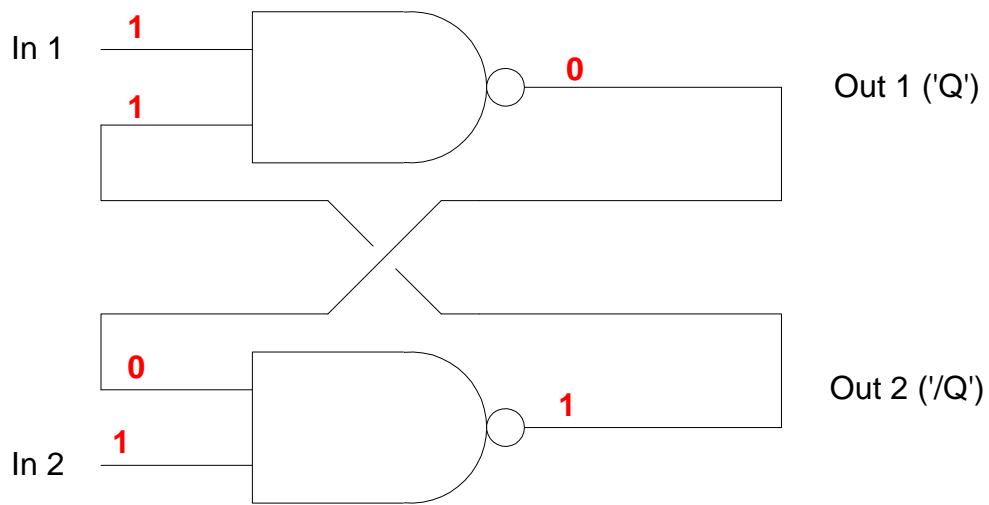
Truth Table



Basic Latch

In1	In2	Q	/Q
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	/Q

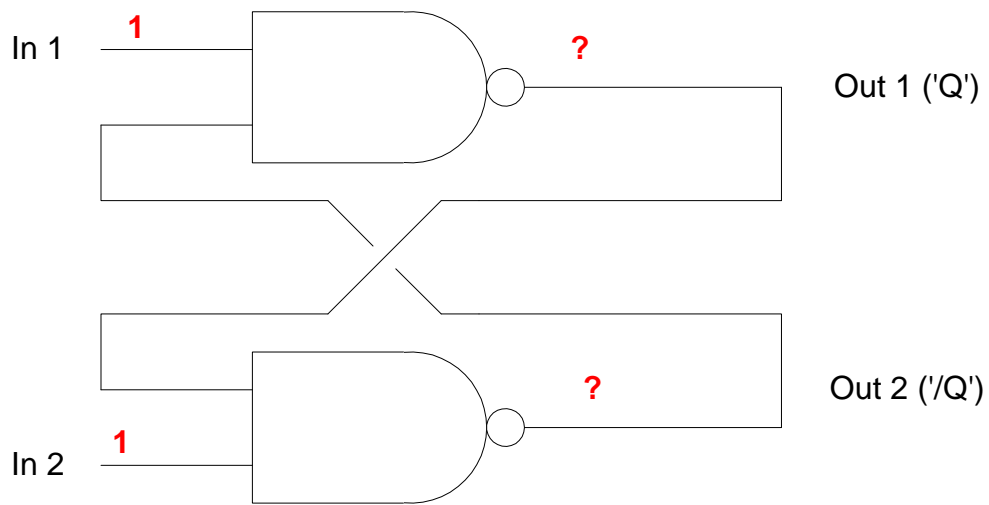
Truth Table



Basic Latch

In1	In2	Q	/Q
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	/Q

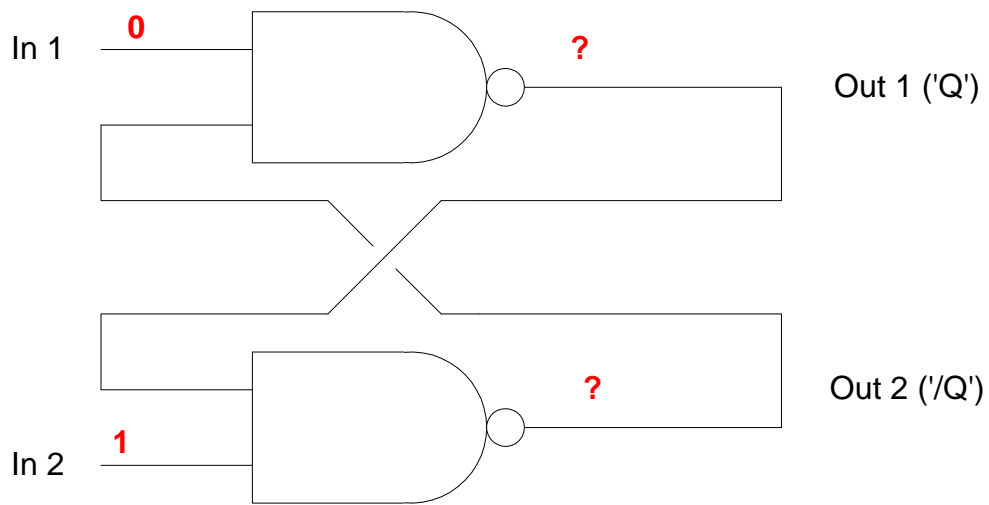
Truth Table



Basic Latch

In1	In2	Q	/Q
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	/Q

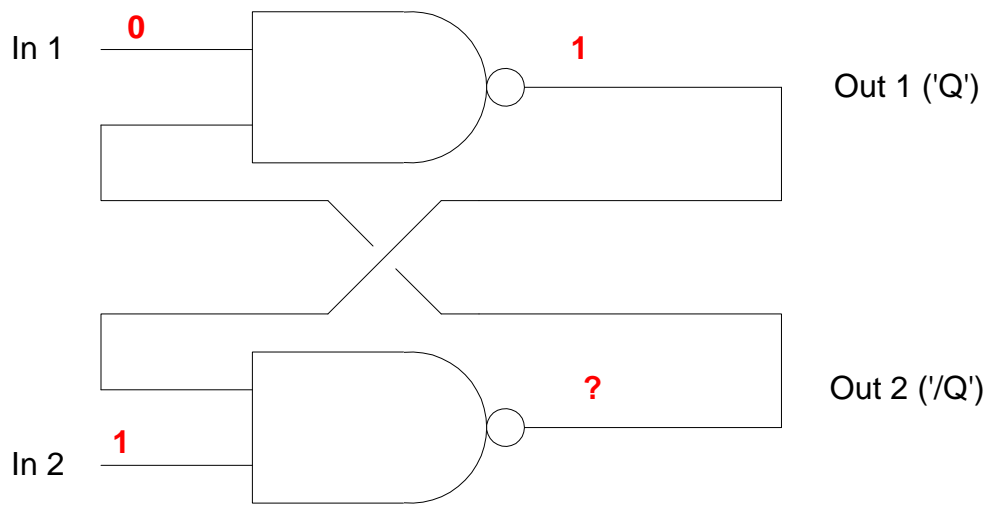
Truth Table



Basic Latch

In1	In2	Q	/Q
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	/Q

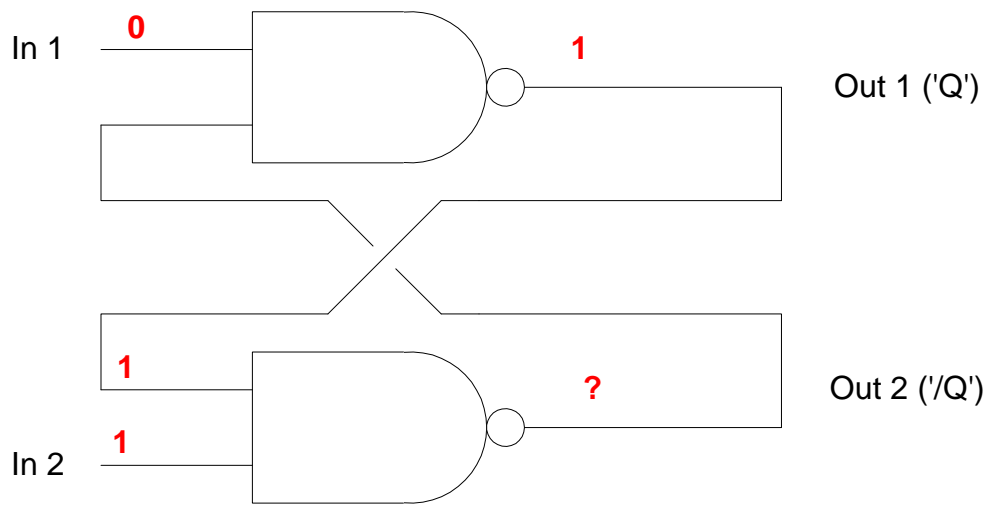
Truth Table



Basic Latch

In1	In2	Q	/Q
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	/Q

Truth Table

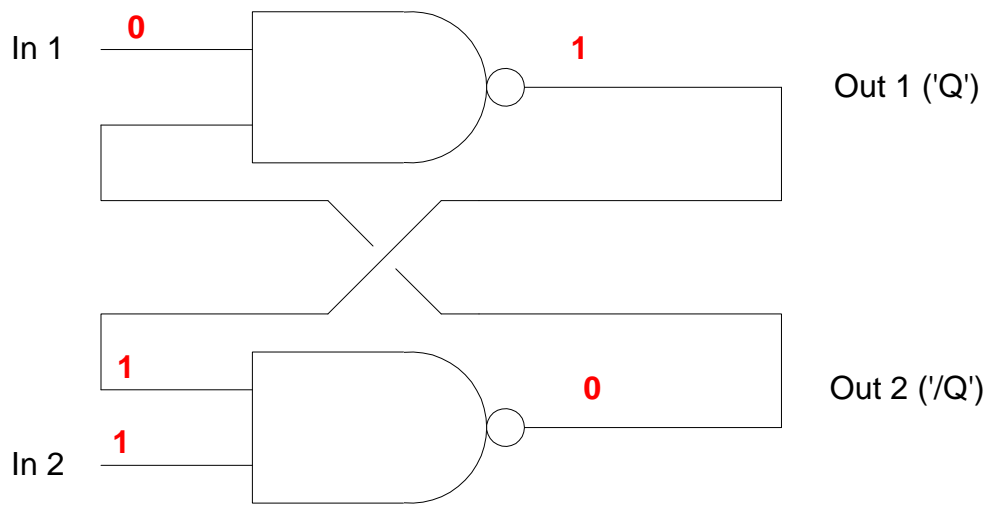


Basic Latch

In1	In2	Q	/Q
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	/Q

Truth Table

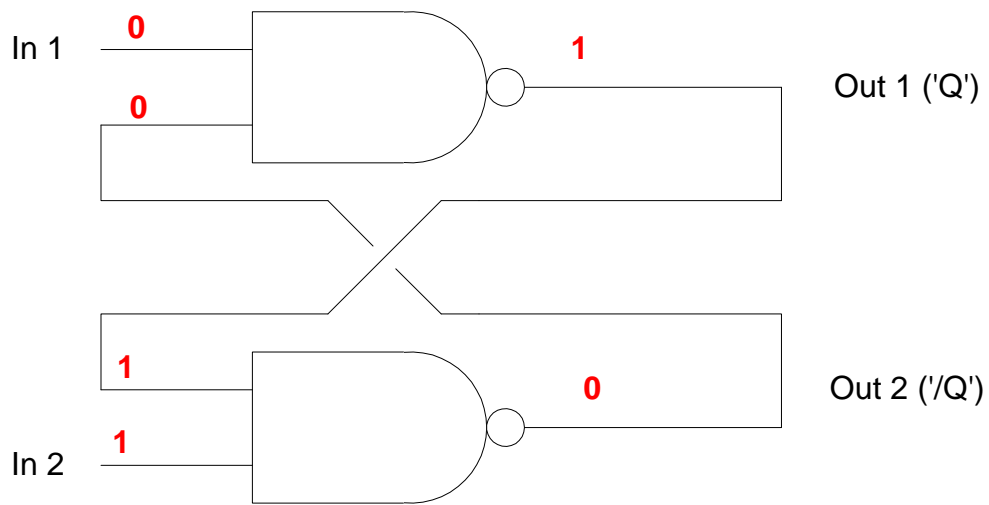




Basic Latch

In1	In2	Q	/Q
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	/Q

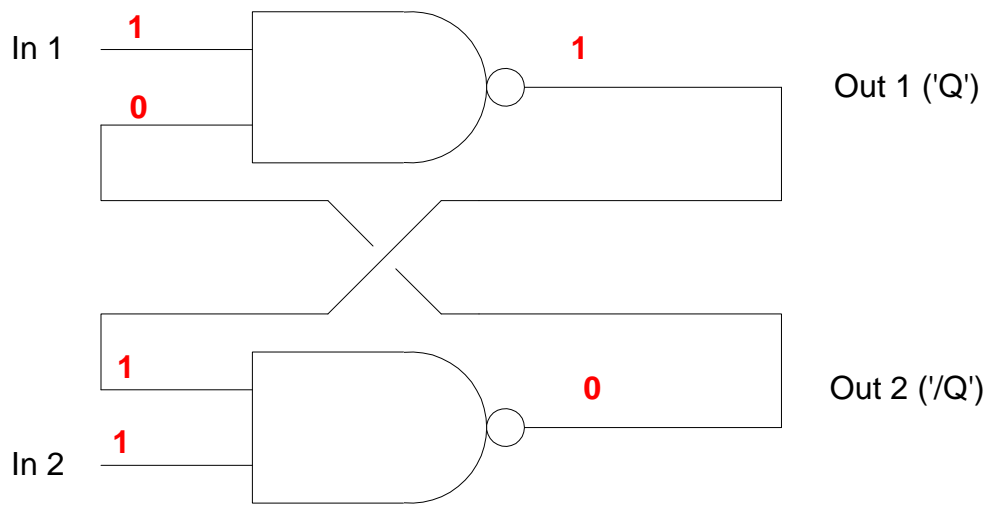
Truth Table



Basic Latch

In1	In2	Q	/Q
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	/Q

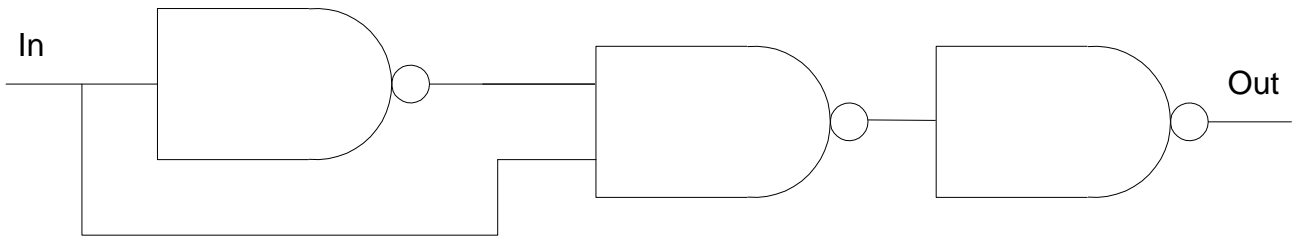
Truth Table



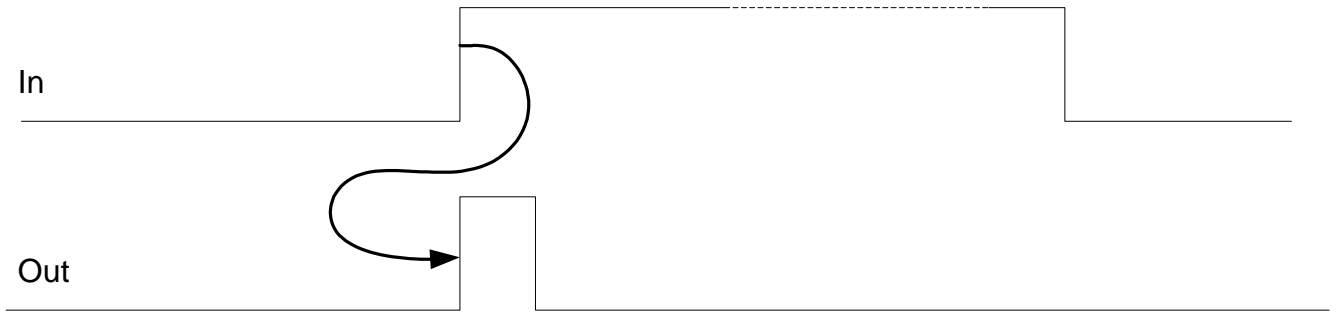
Basic Latch

In1	In2	Q	/Q
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Q	/Q

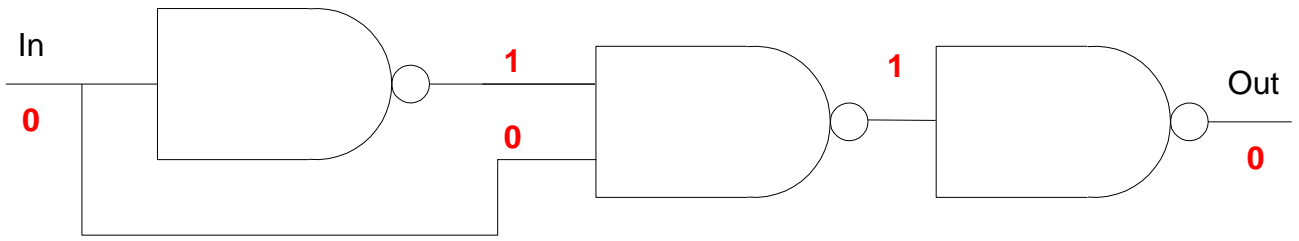
Truth Table



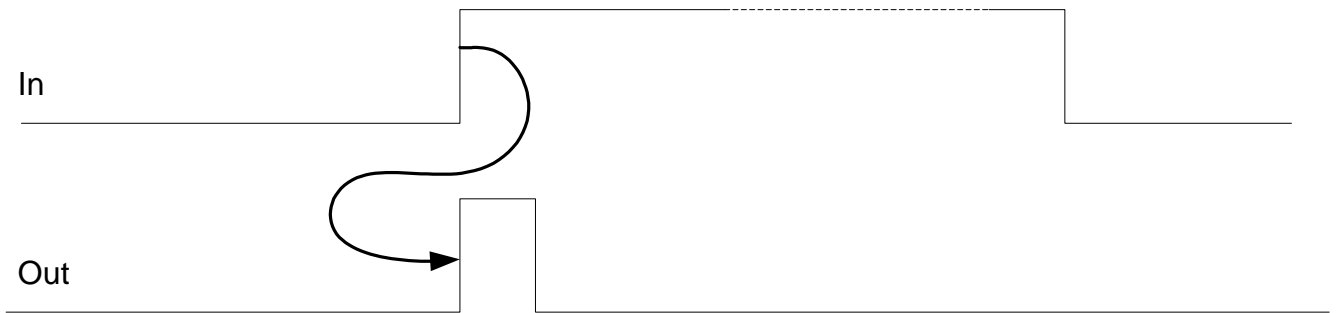
Edge Generator



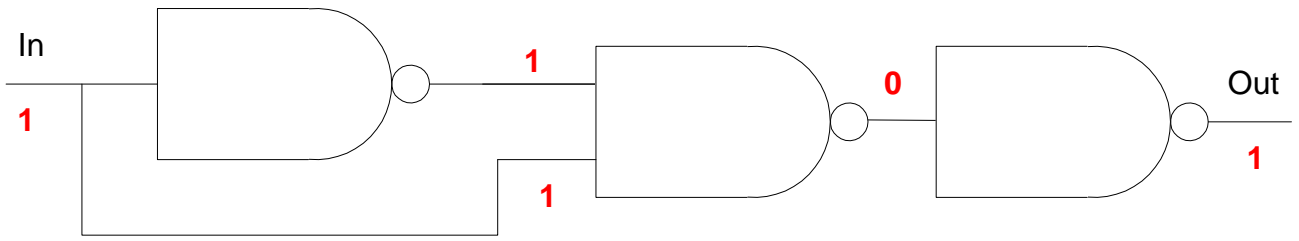
Timing Diagram



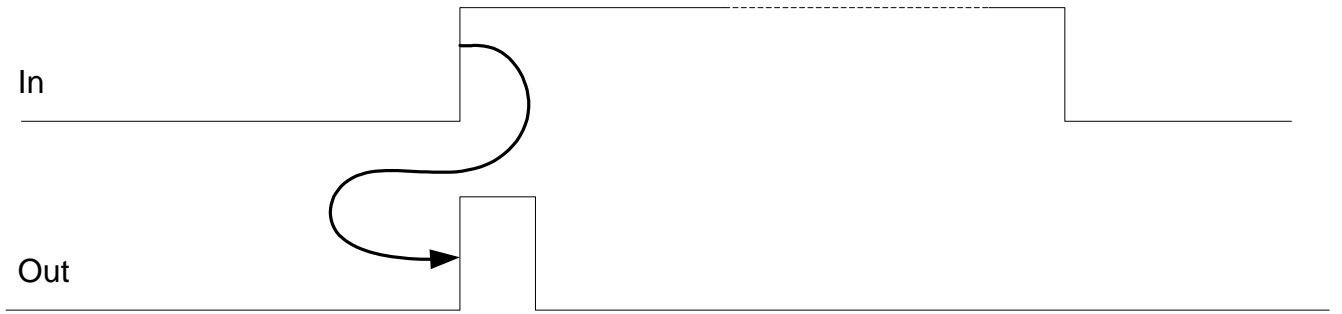
Edge Generator



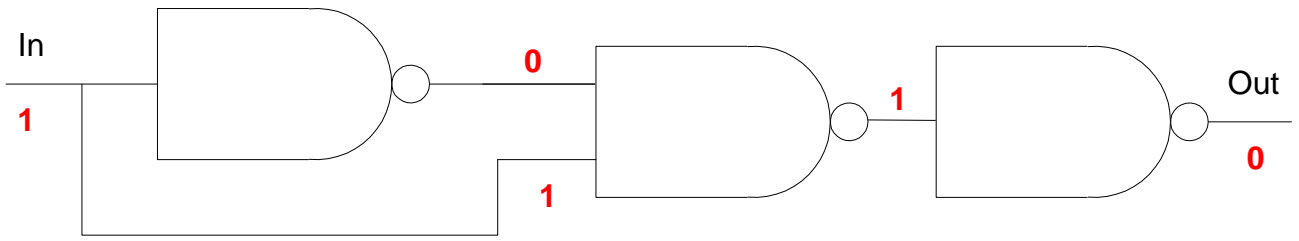
Timing Diagram



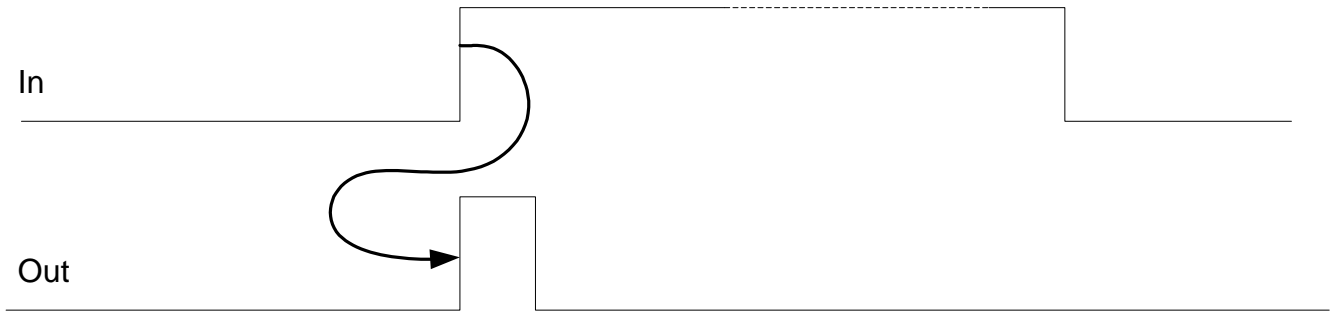
Edge Generator



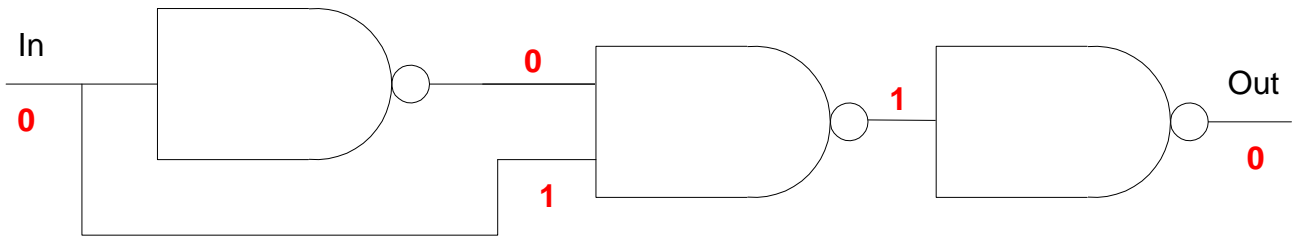
Timing Diagram



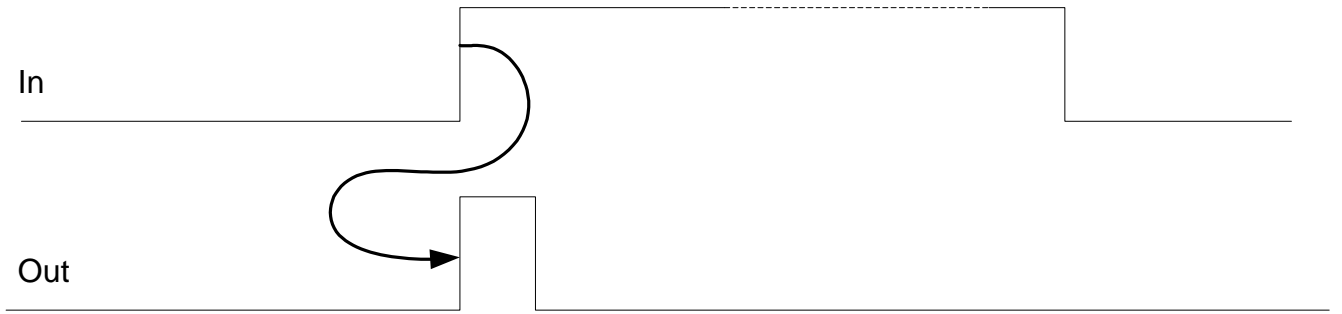
Edge Generator



Timing Diagram

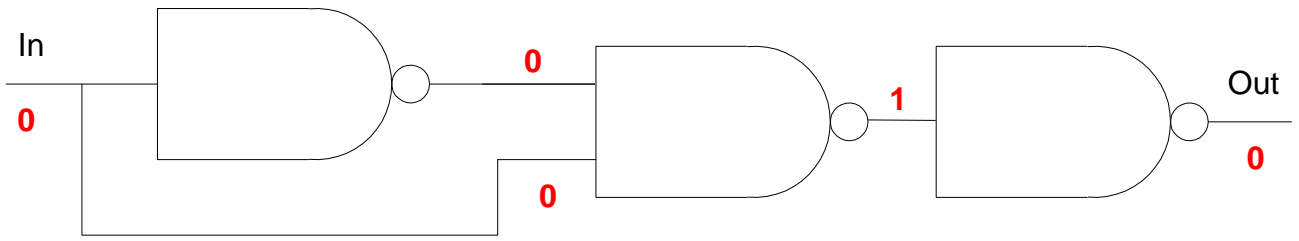


Edge Generator

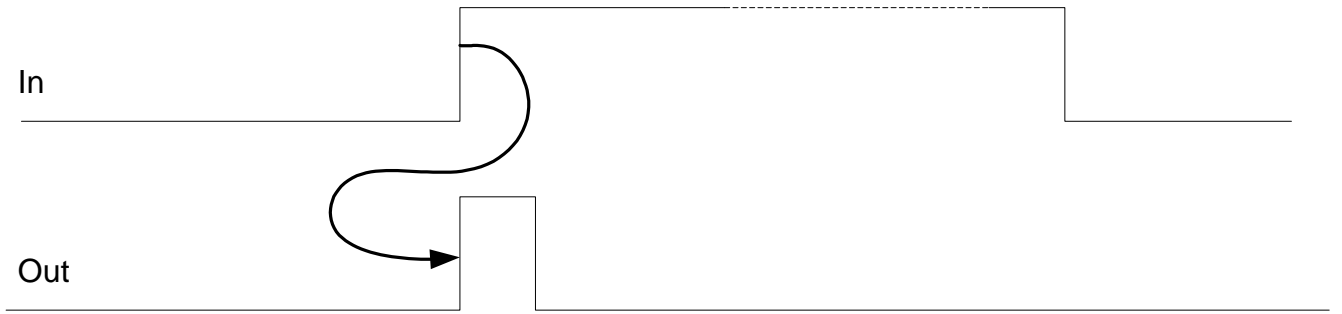


Timing Diagram

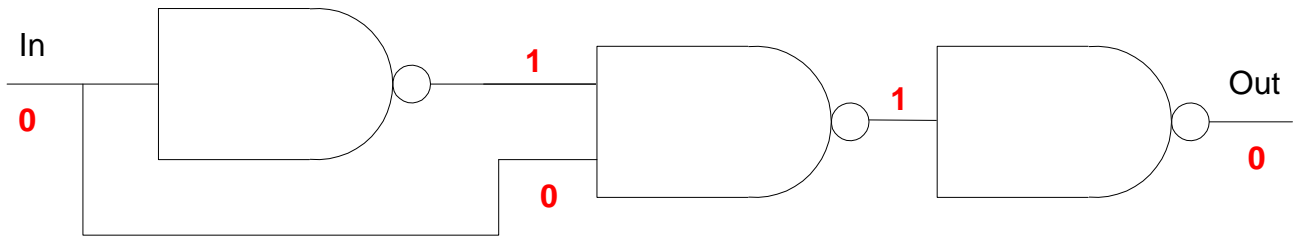




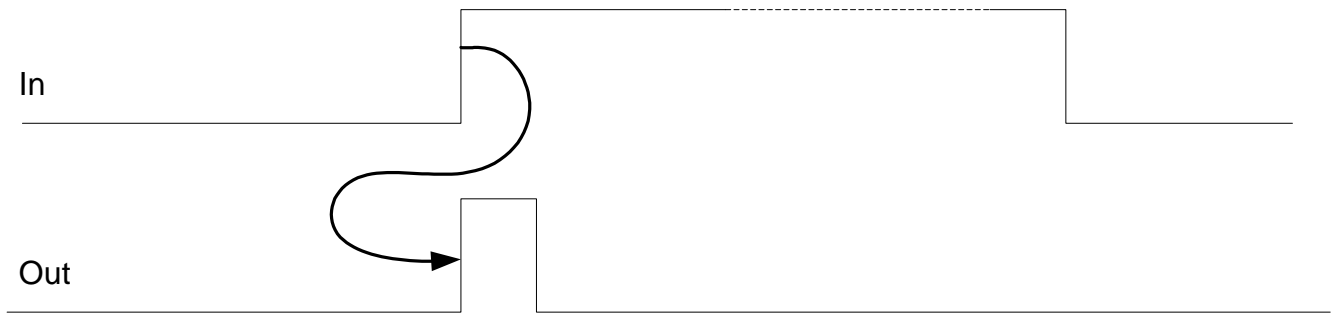
Edge Generator



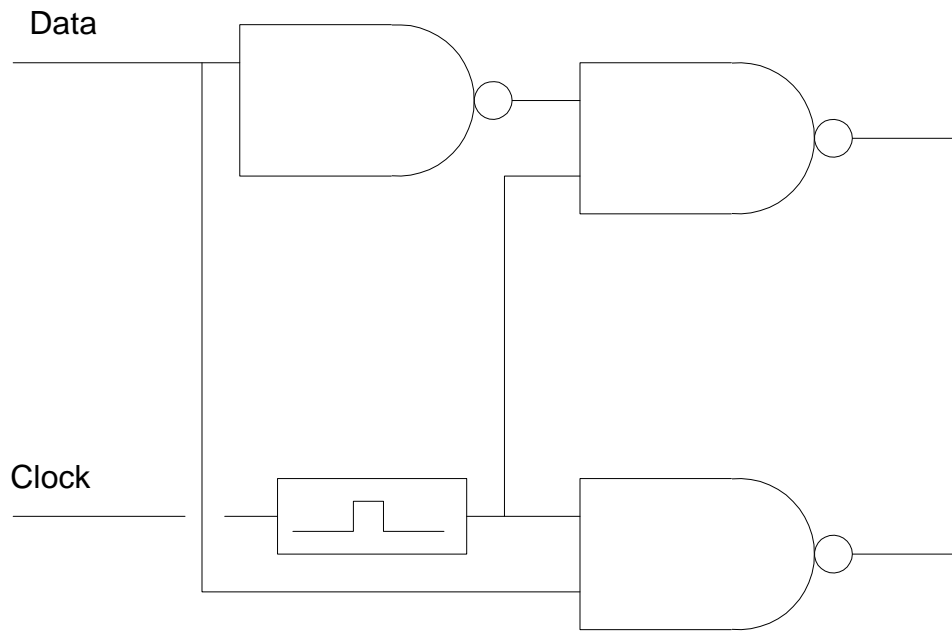
Timing Diagram



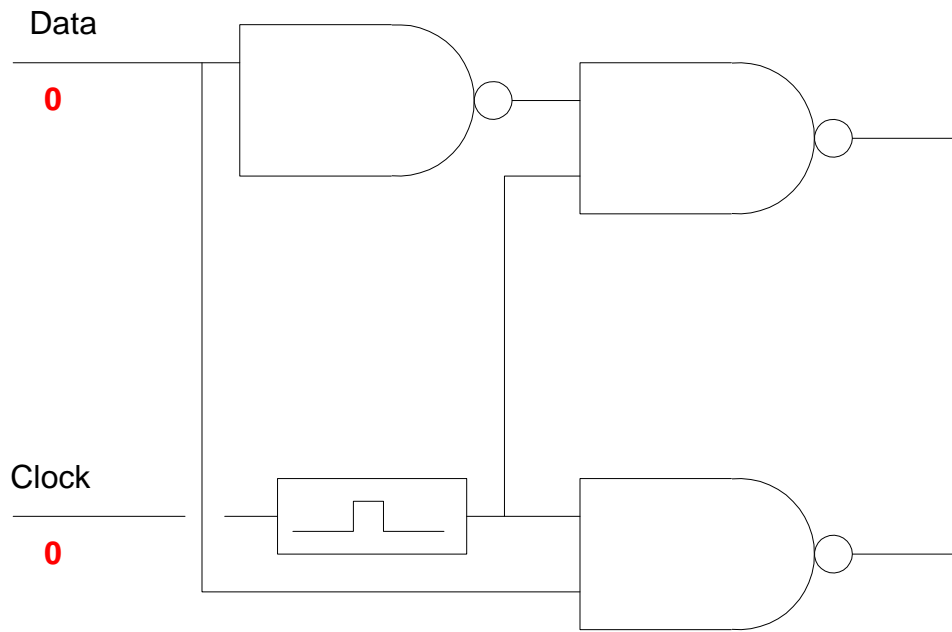
Edge Generator



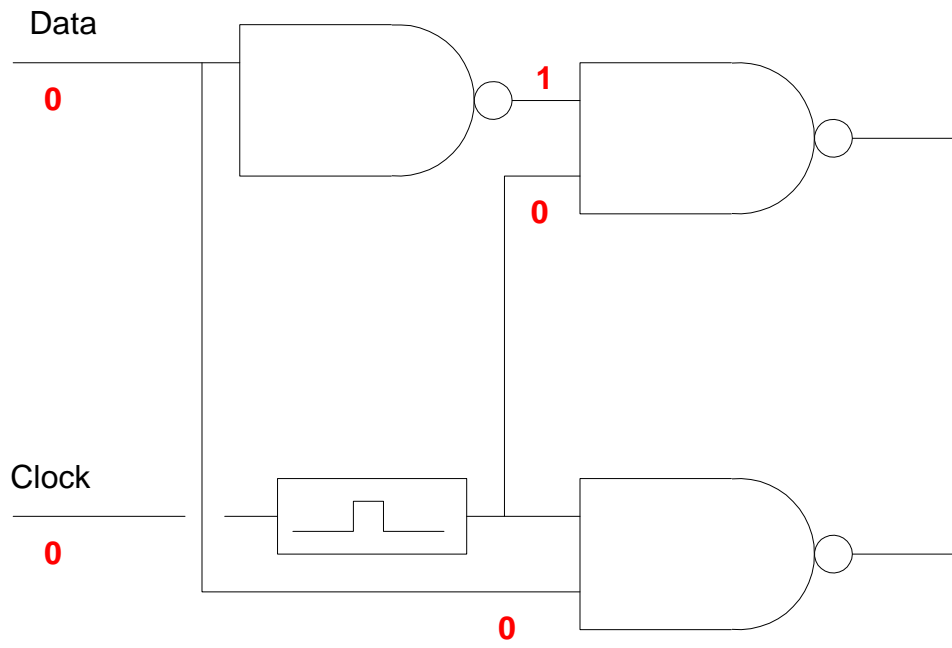
Timing Diagram



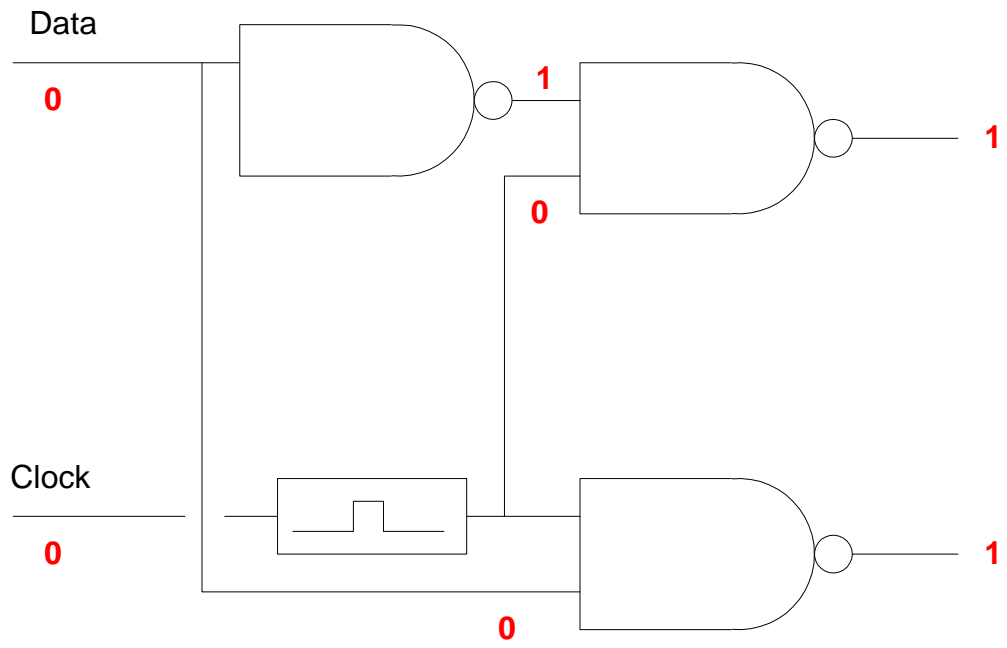
Pulse Switching Logic



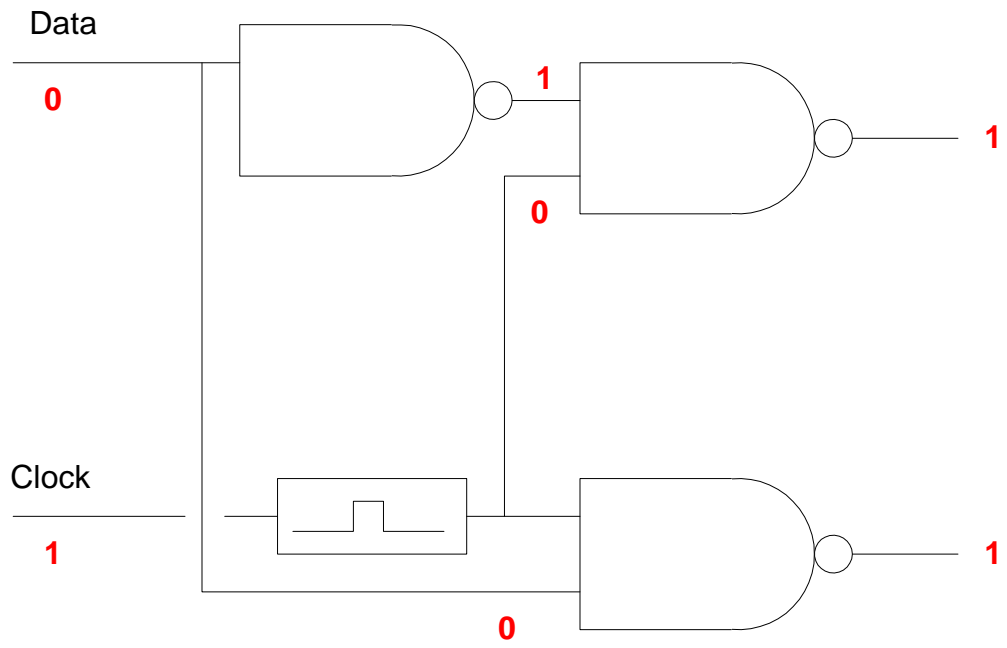
Pulse Switching Logic



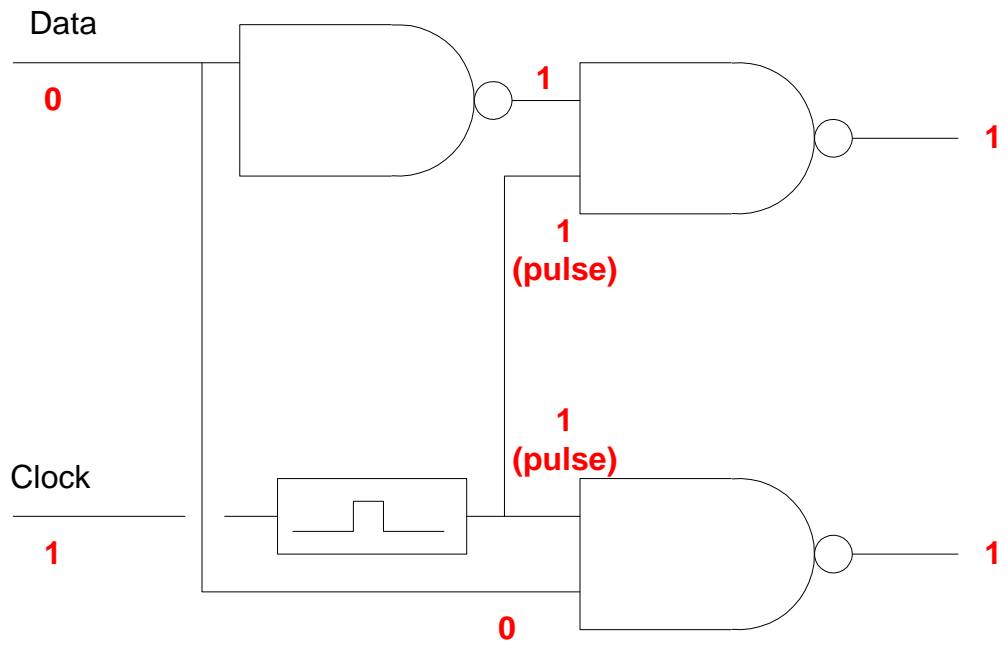
Pulse Switching Logic



Pulse Switching Logic

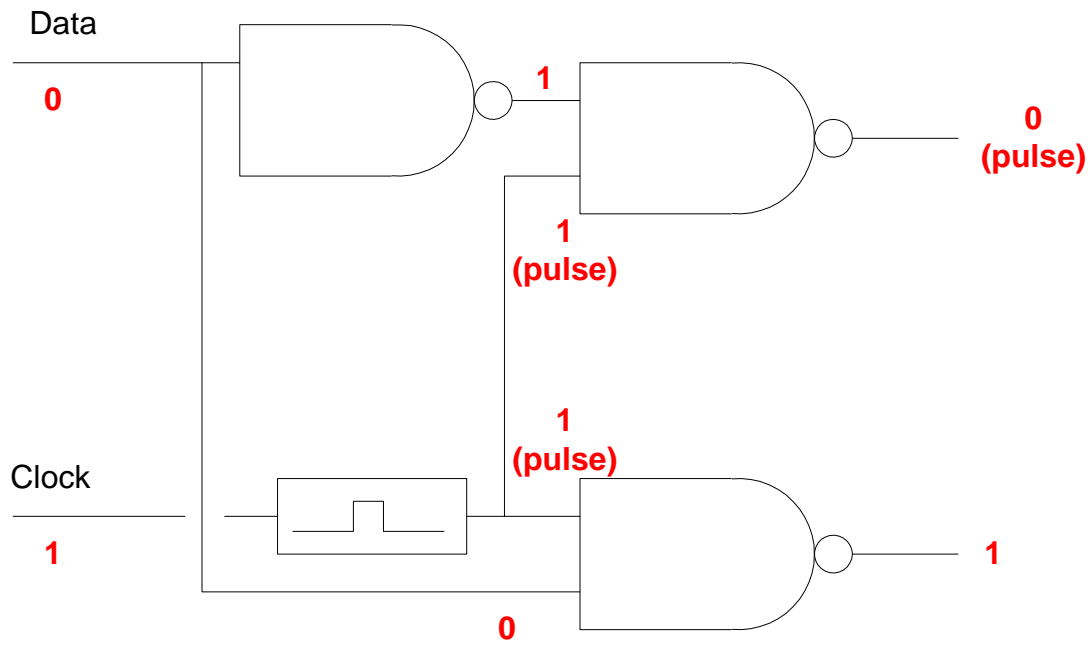


Pulse Switching Logic

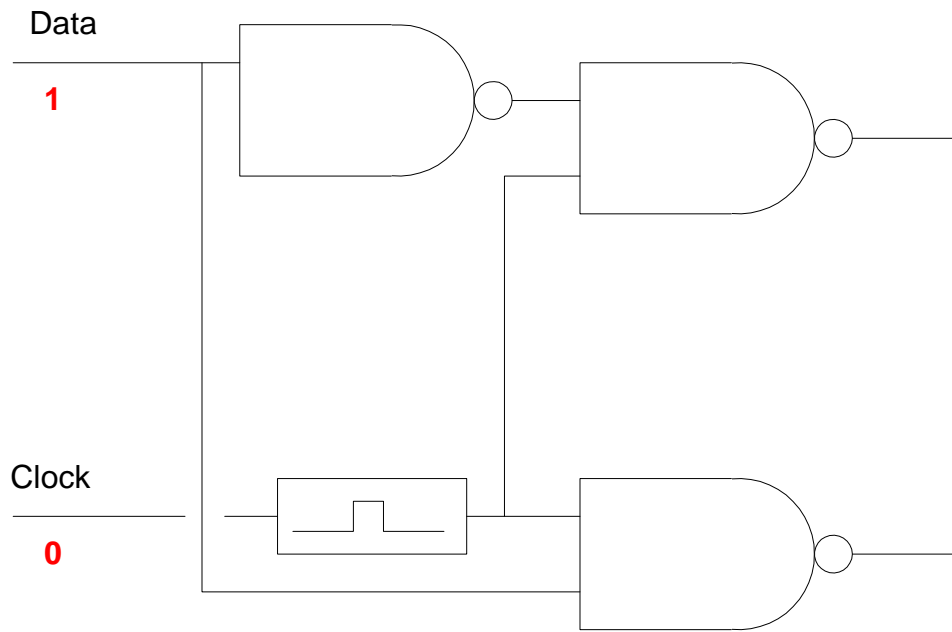


Pulse Switching Logic

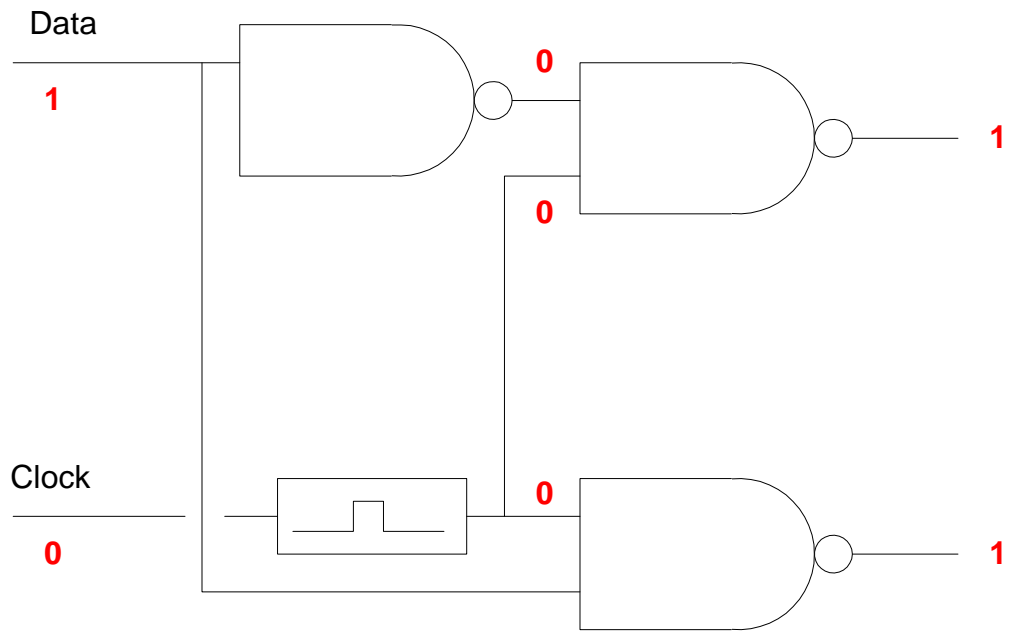




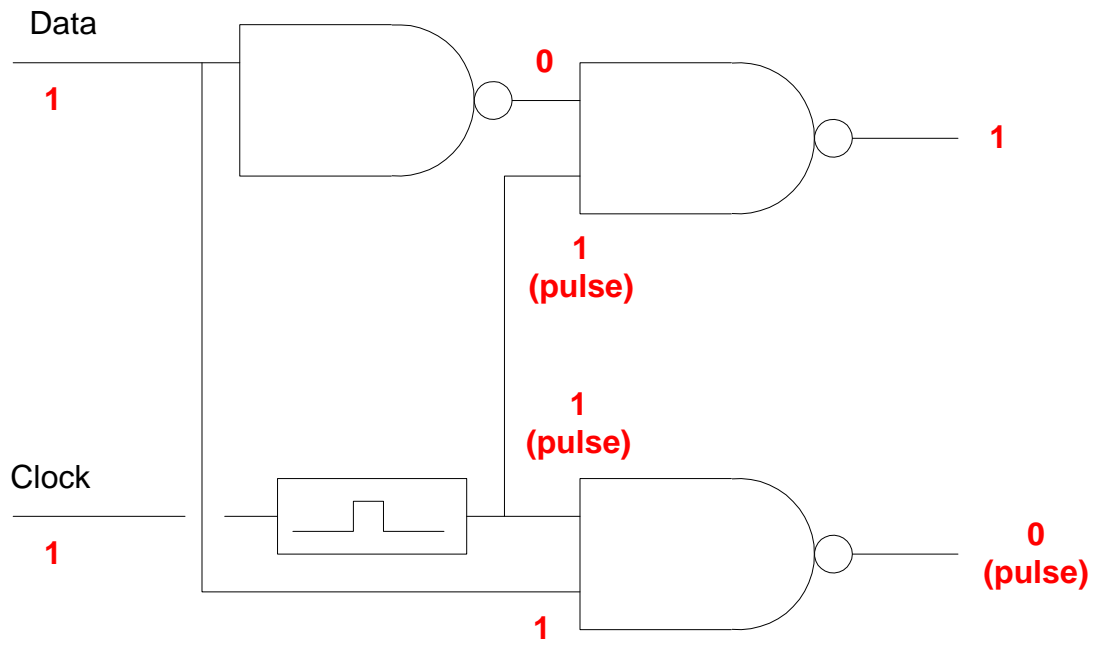
Pulse Switching Logic



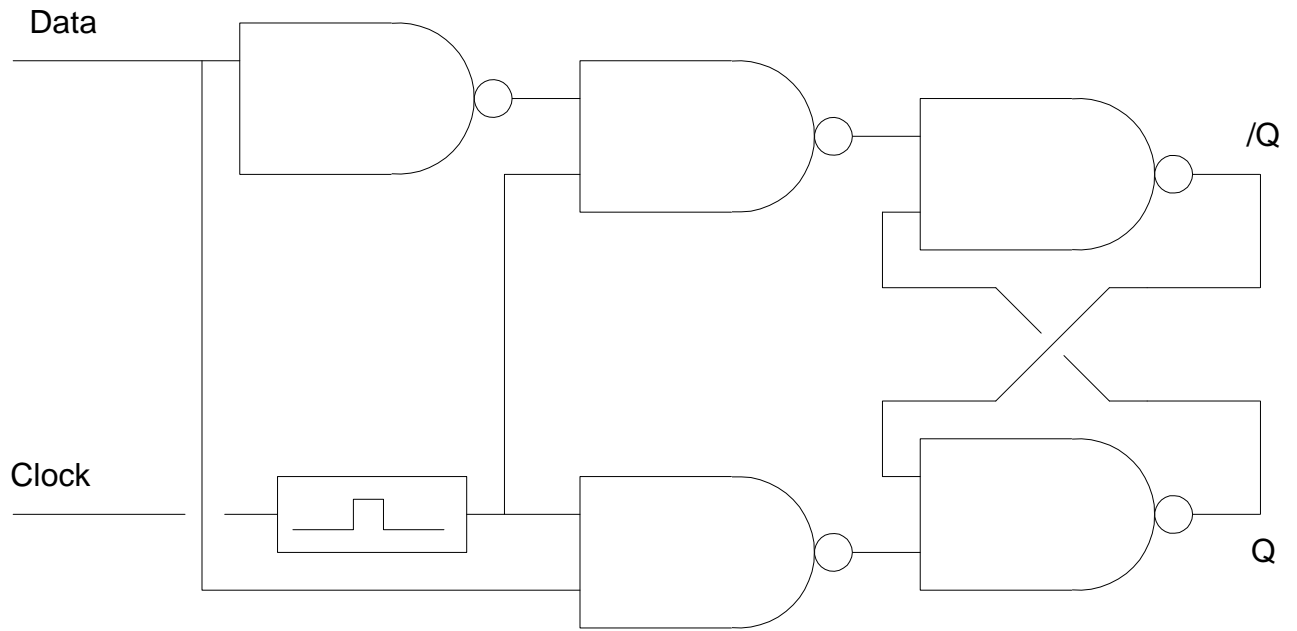
Pulse Switching Logic



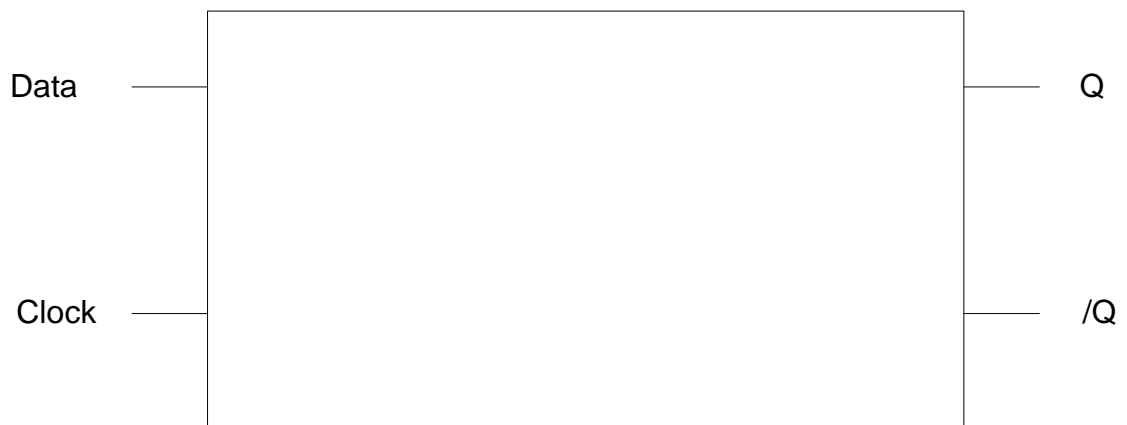
Pulse Switching Logic



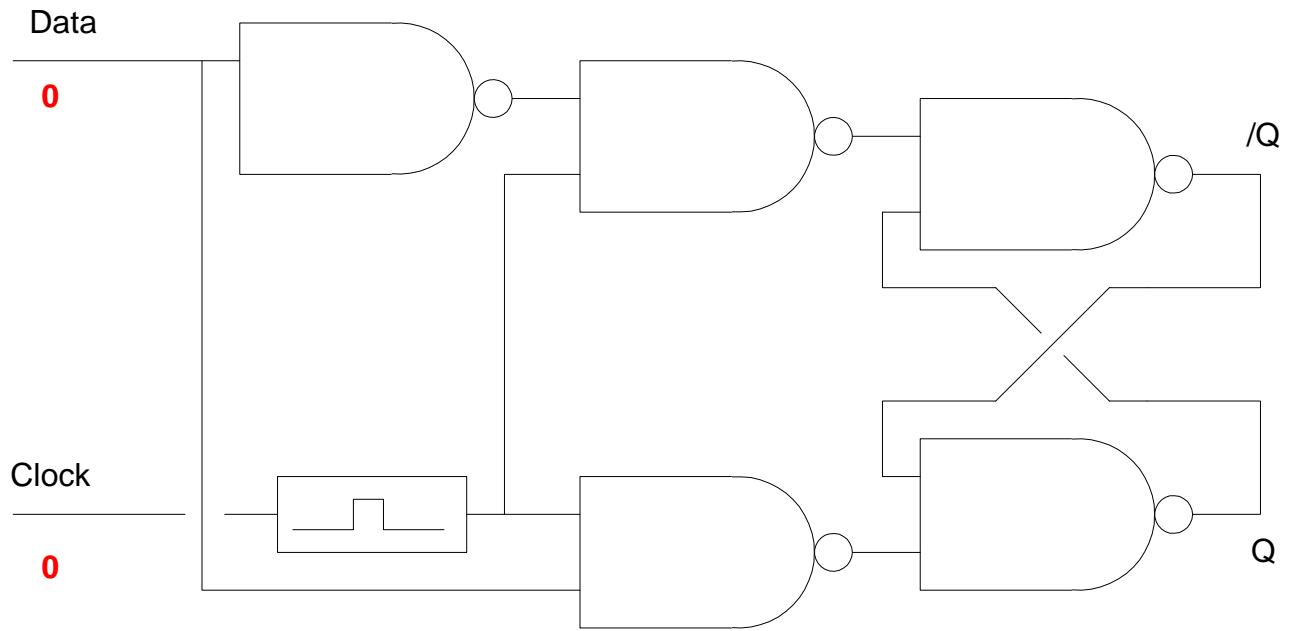
Pulse Switching Logic



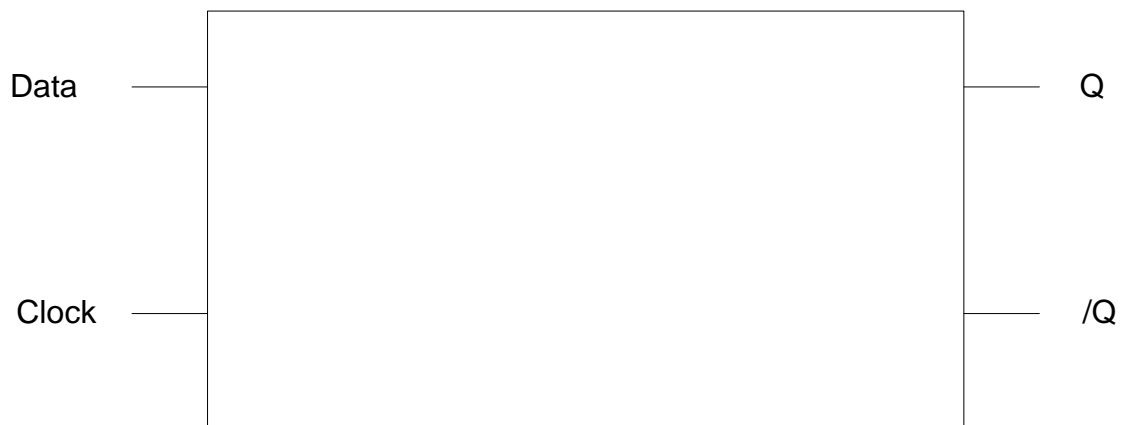
D Type Latch



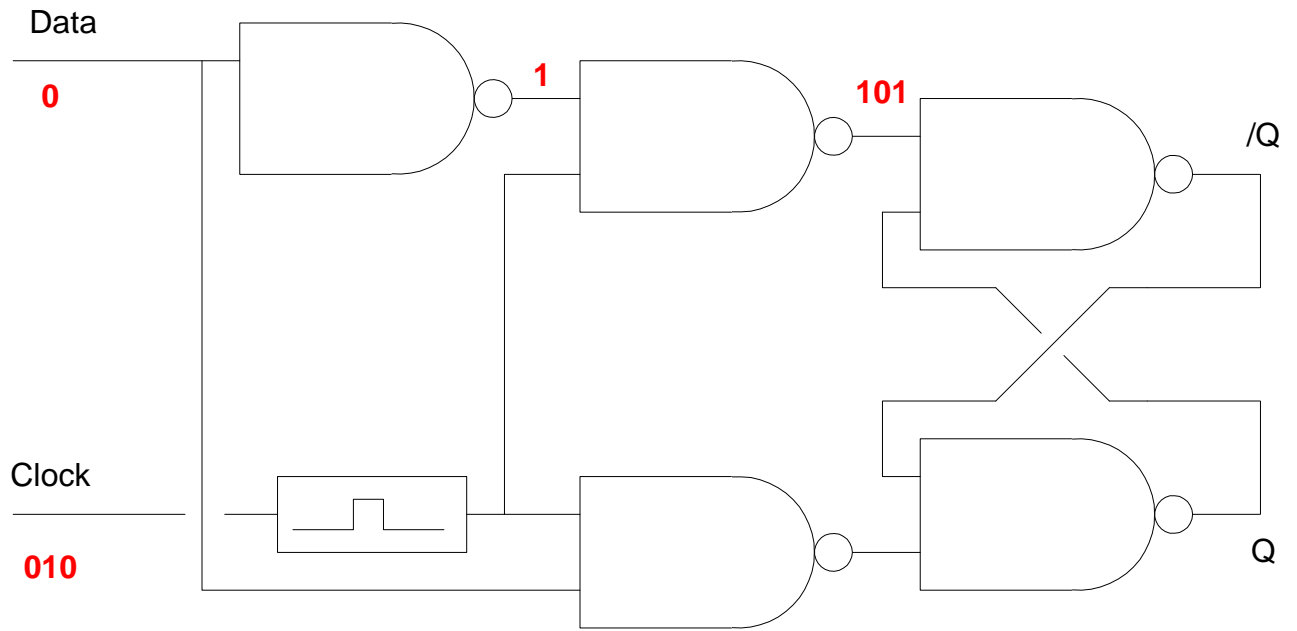
Standard Logic Symbol



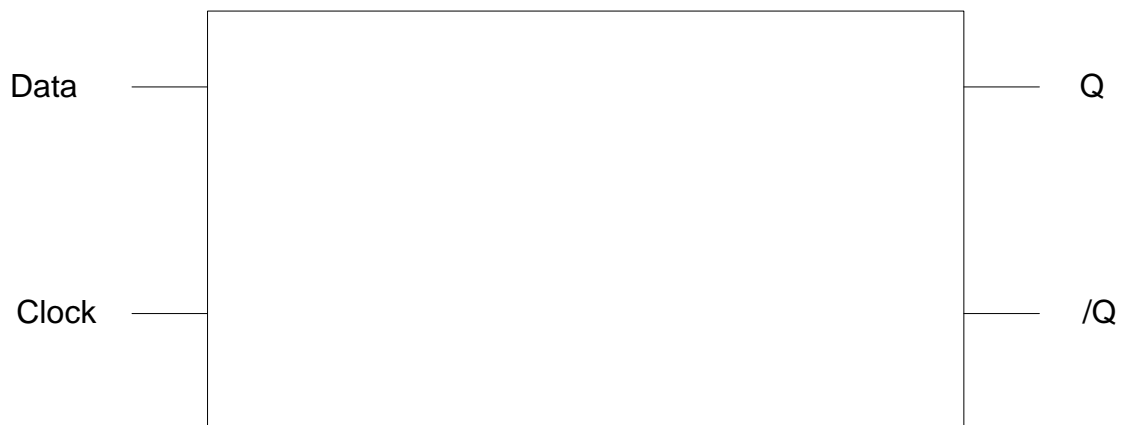
D Type Latch



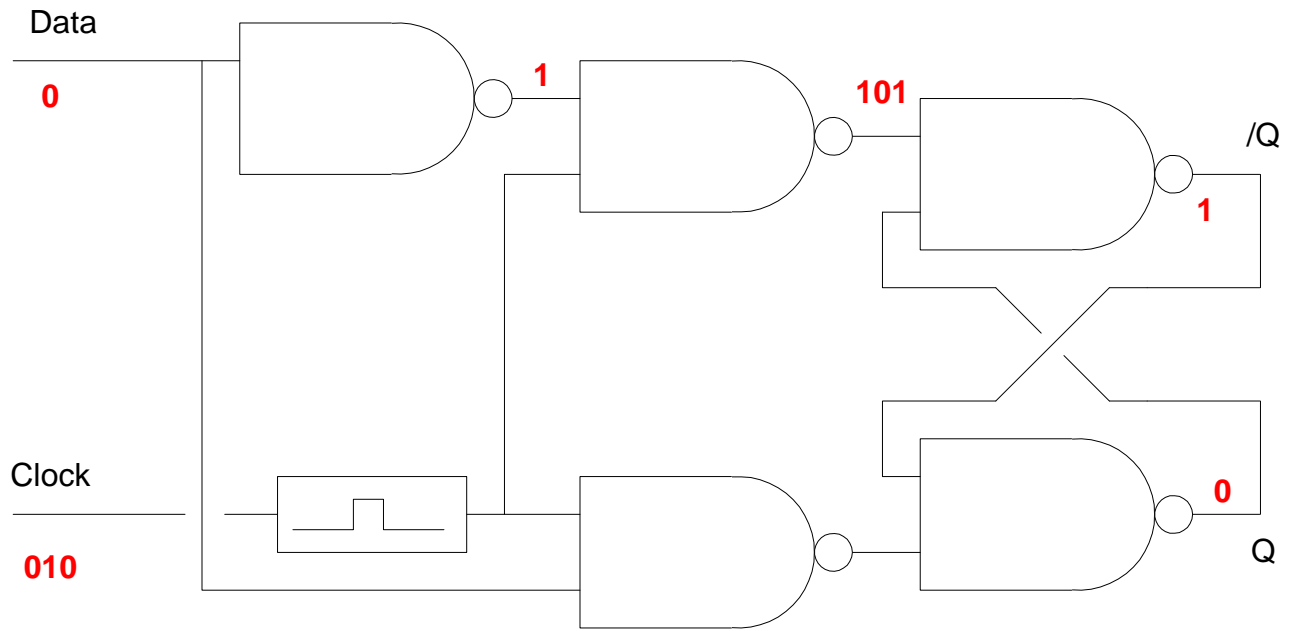
Standard Logic Symbol



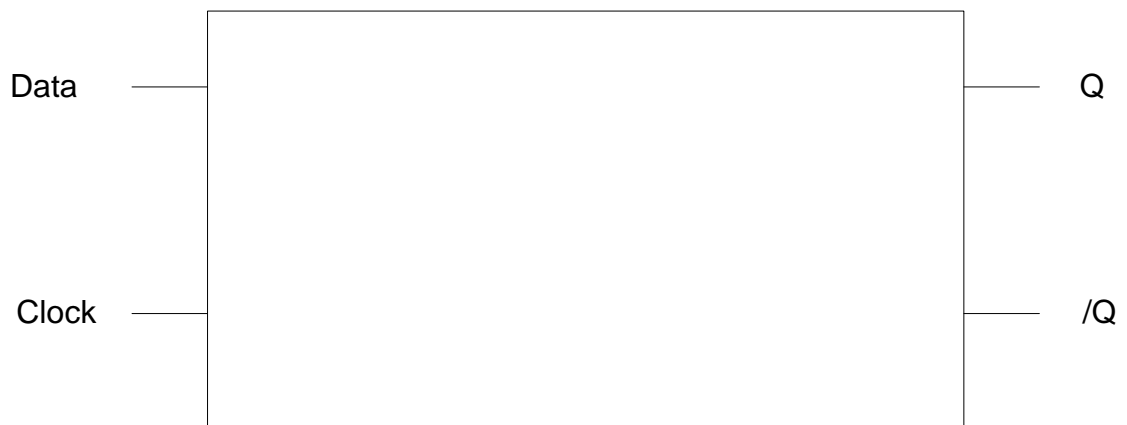
D Type Latch



Standard Logic Symbol

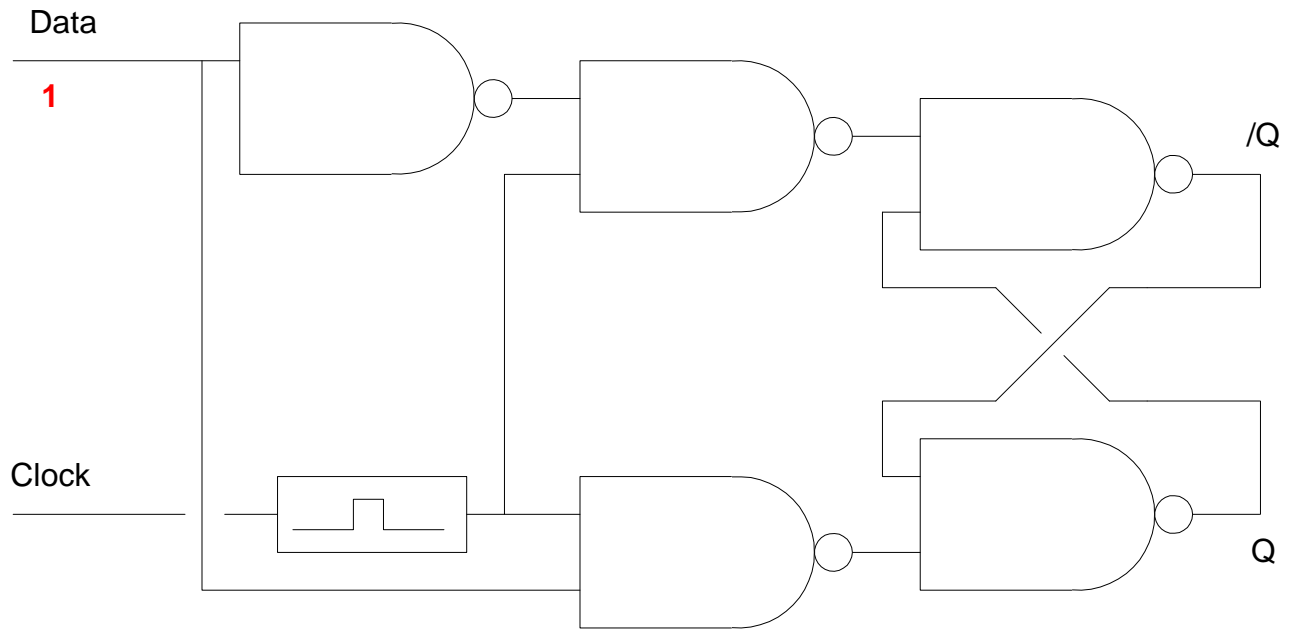


D Type Latch

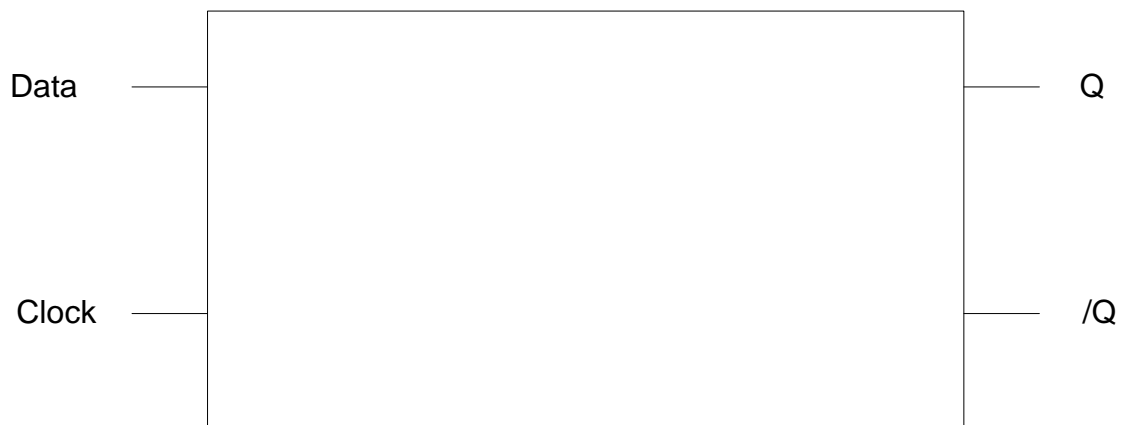


Standard Logic Symbol

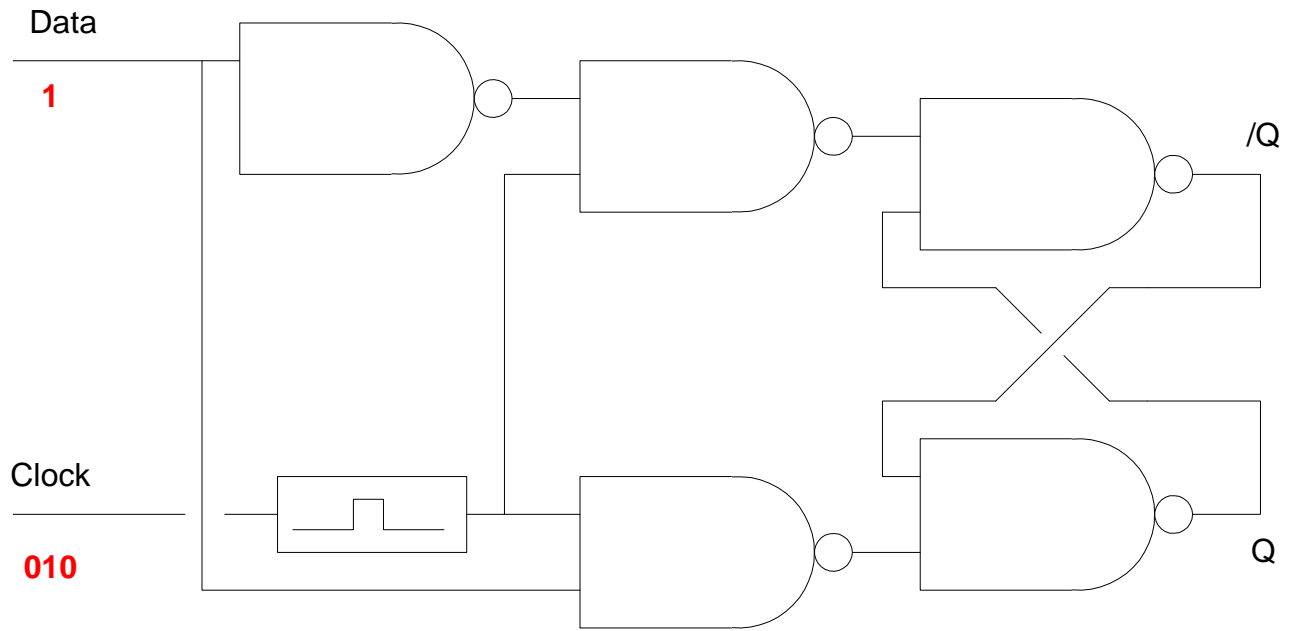




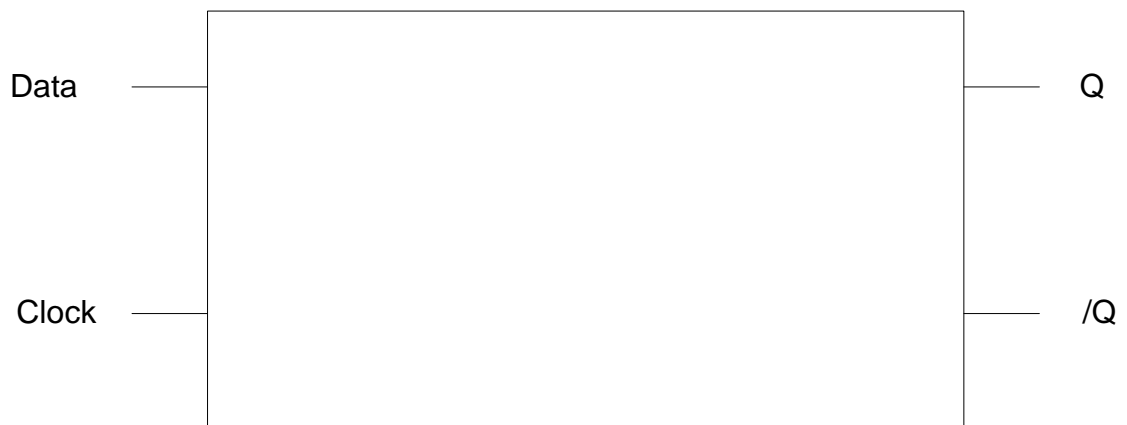
D Type Latch



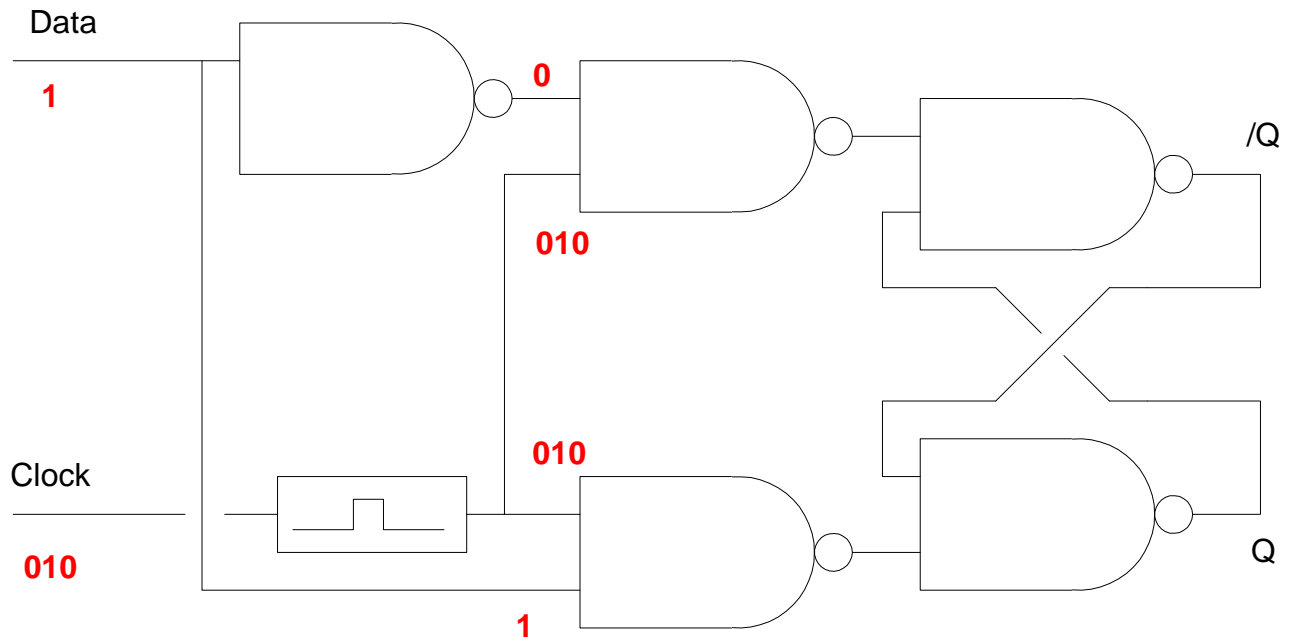
Standard Logic Symbol



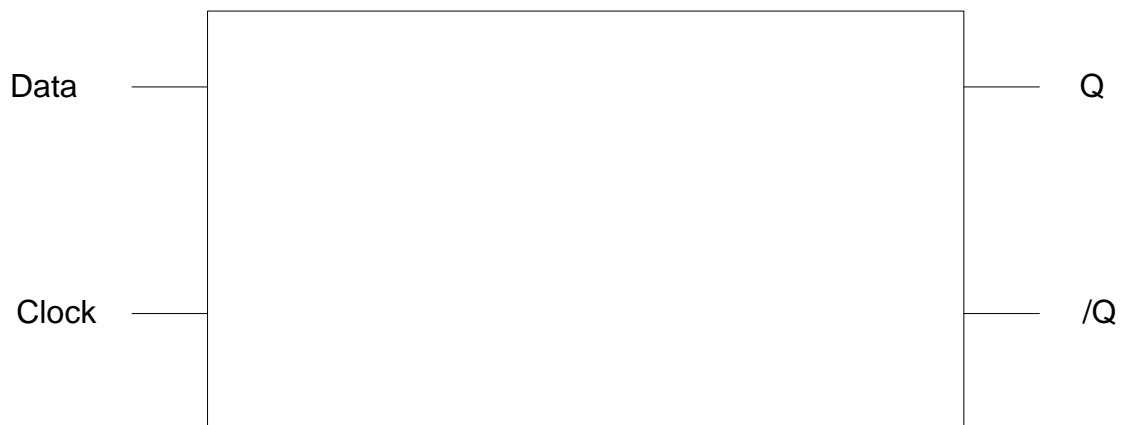
D Type Latch



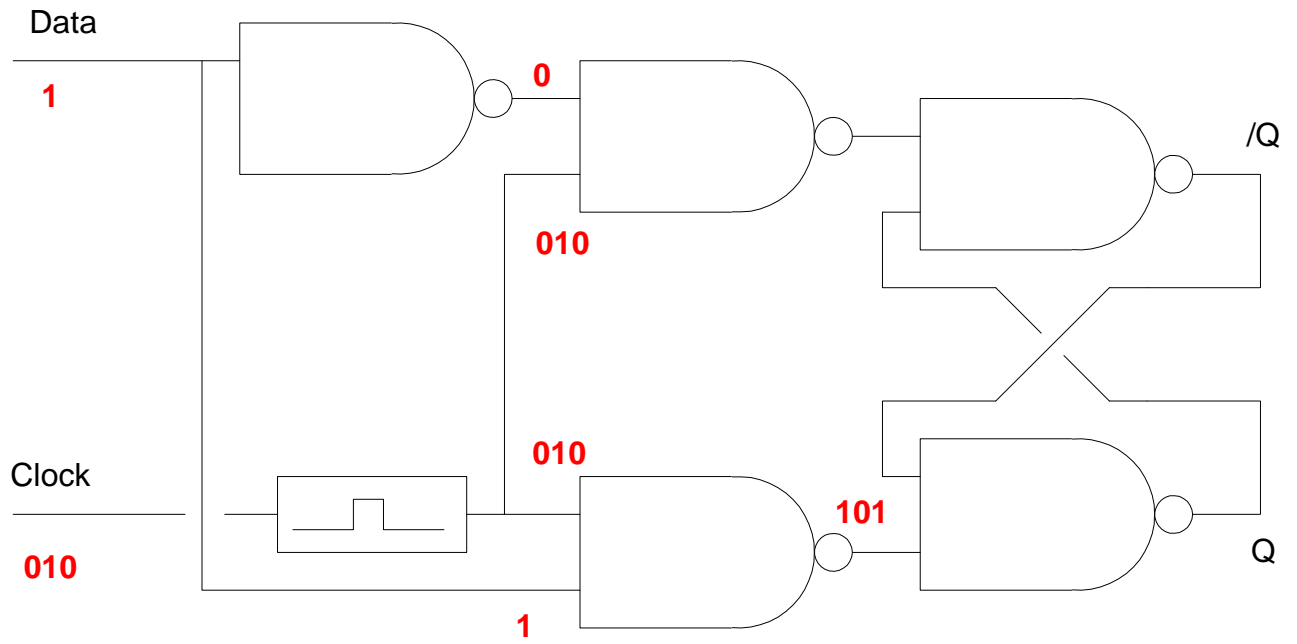
Standard Logic Symbol



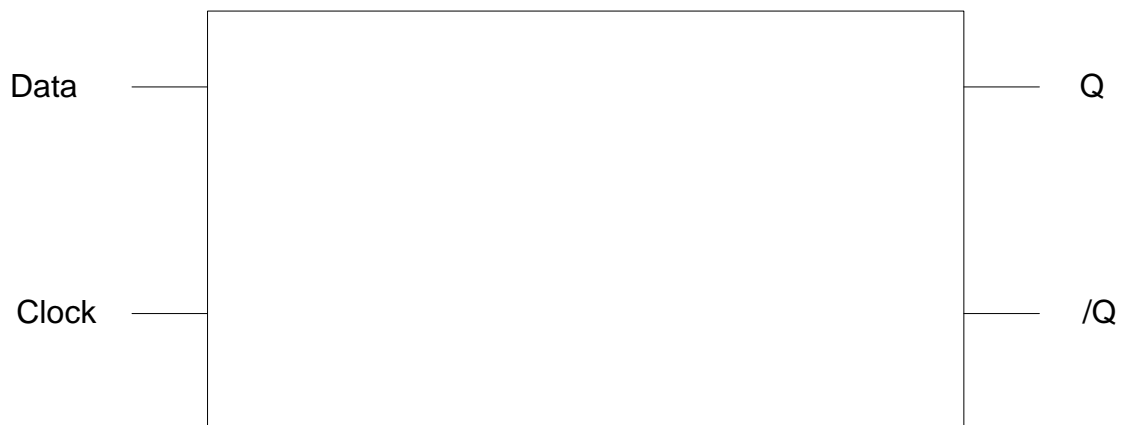
D Type Latch



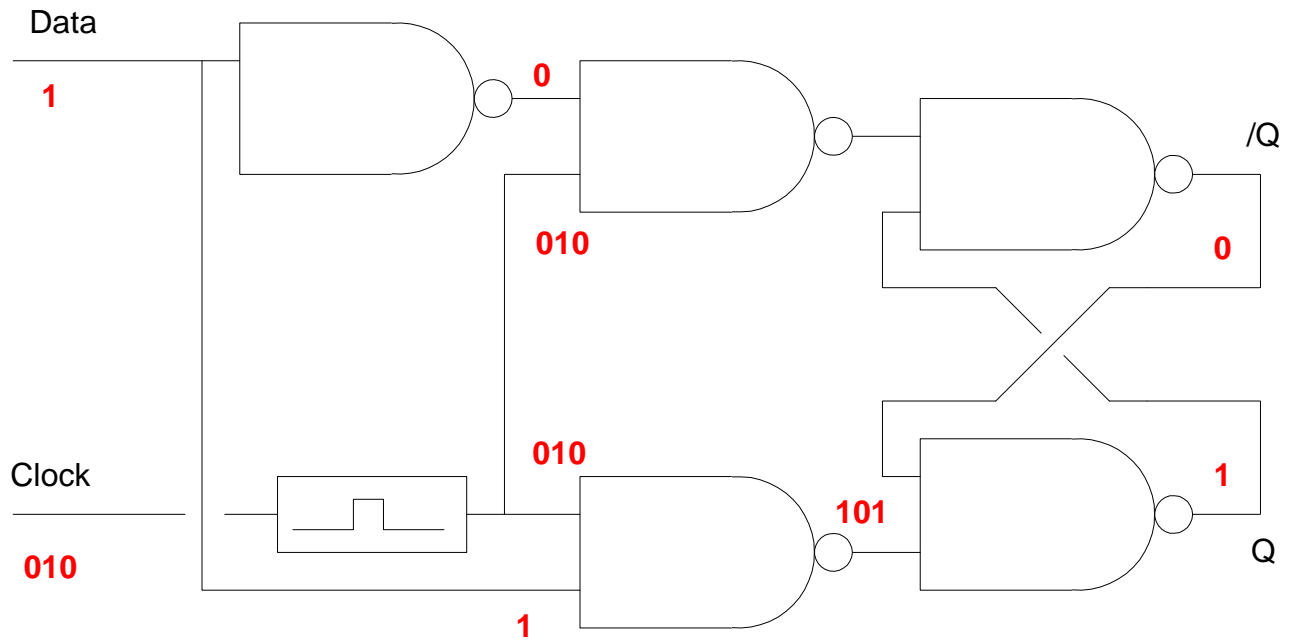
Standard Logic Symbol



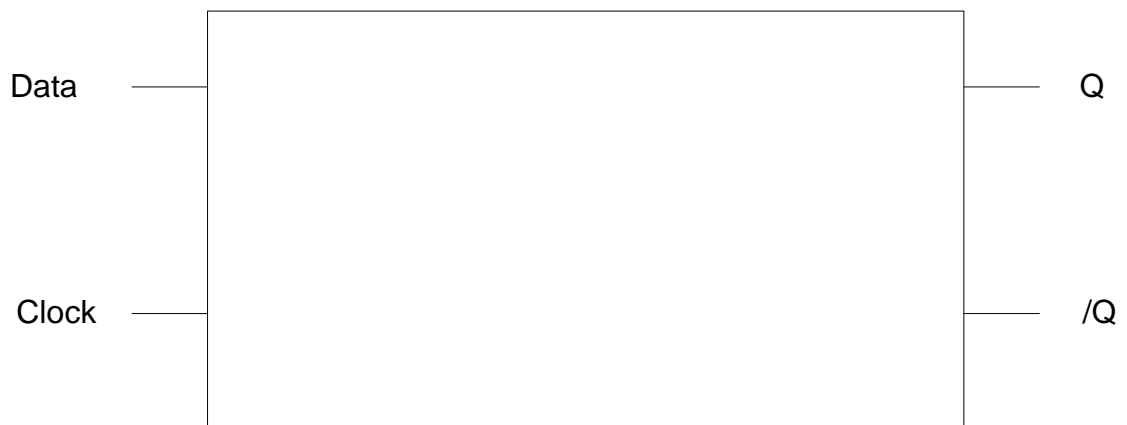
D Type Latch



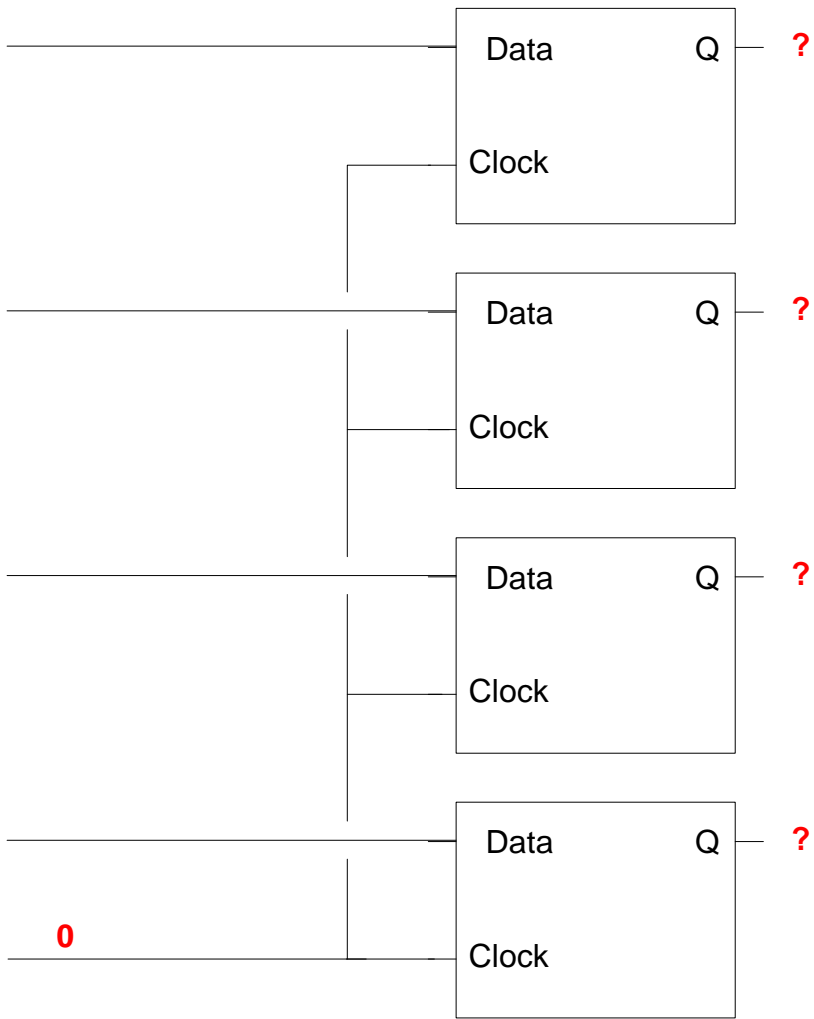
Standard Logic Symbol



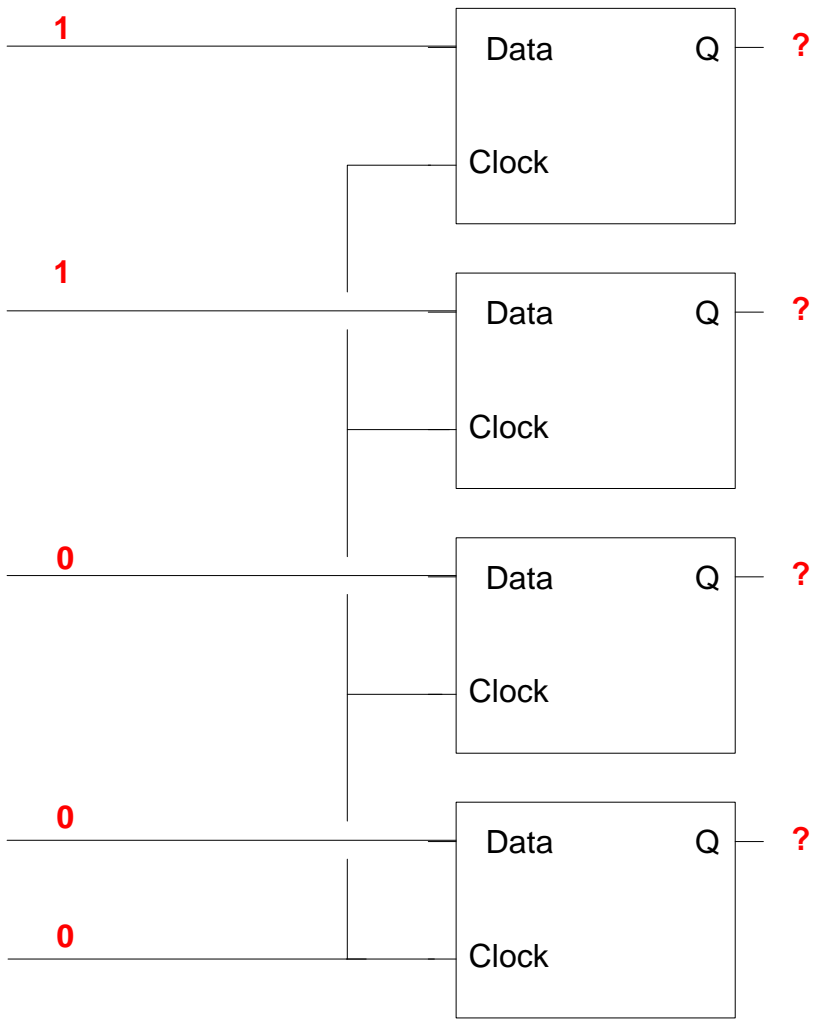
D Type Latch



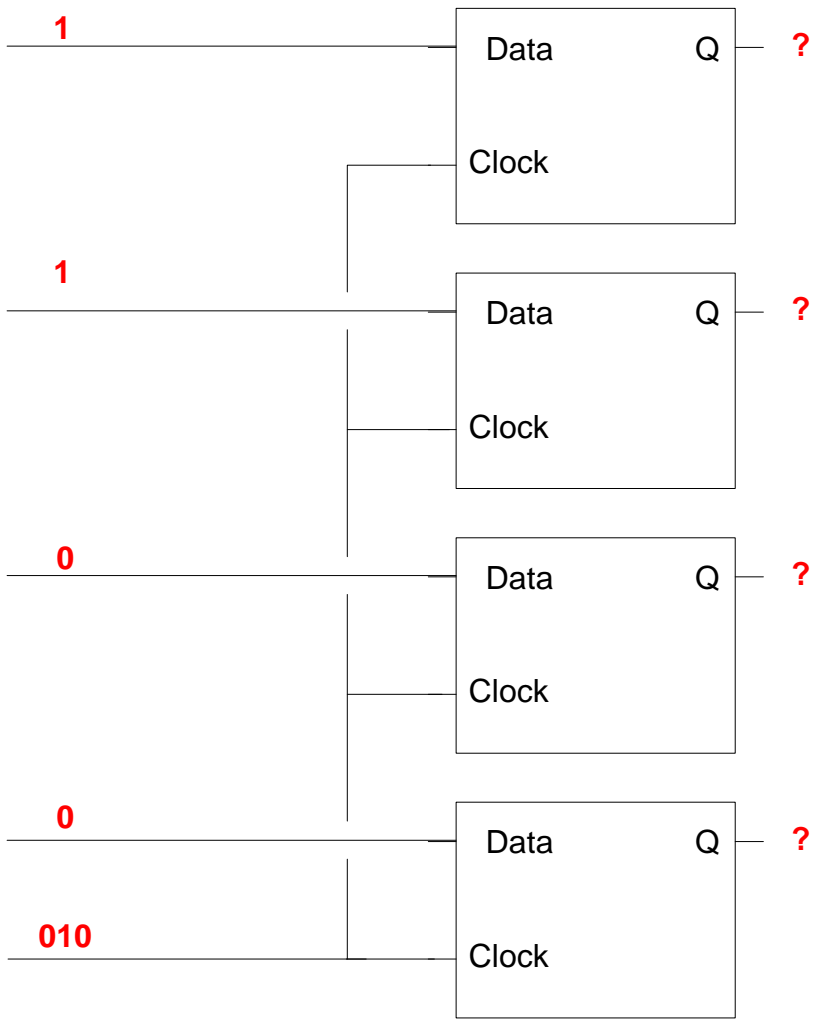
Standard Logic Symbol



4-bit data latch

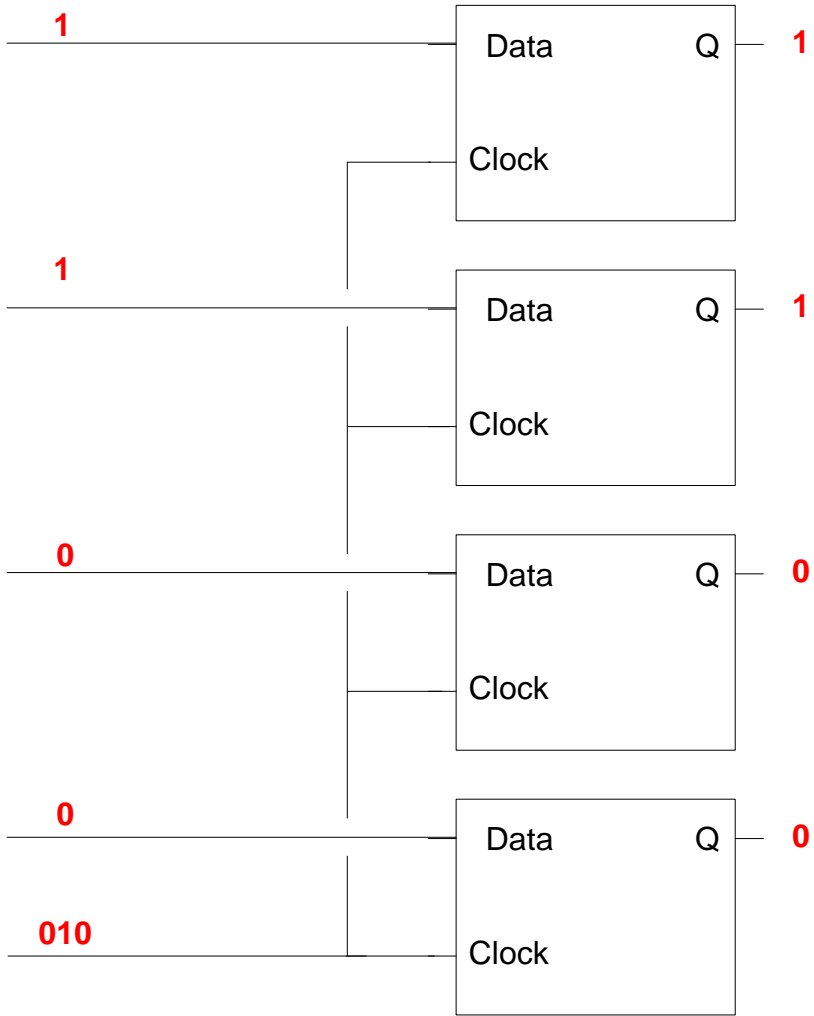


4-bit data latch

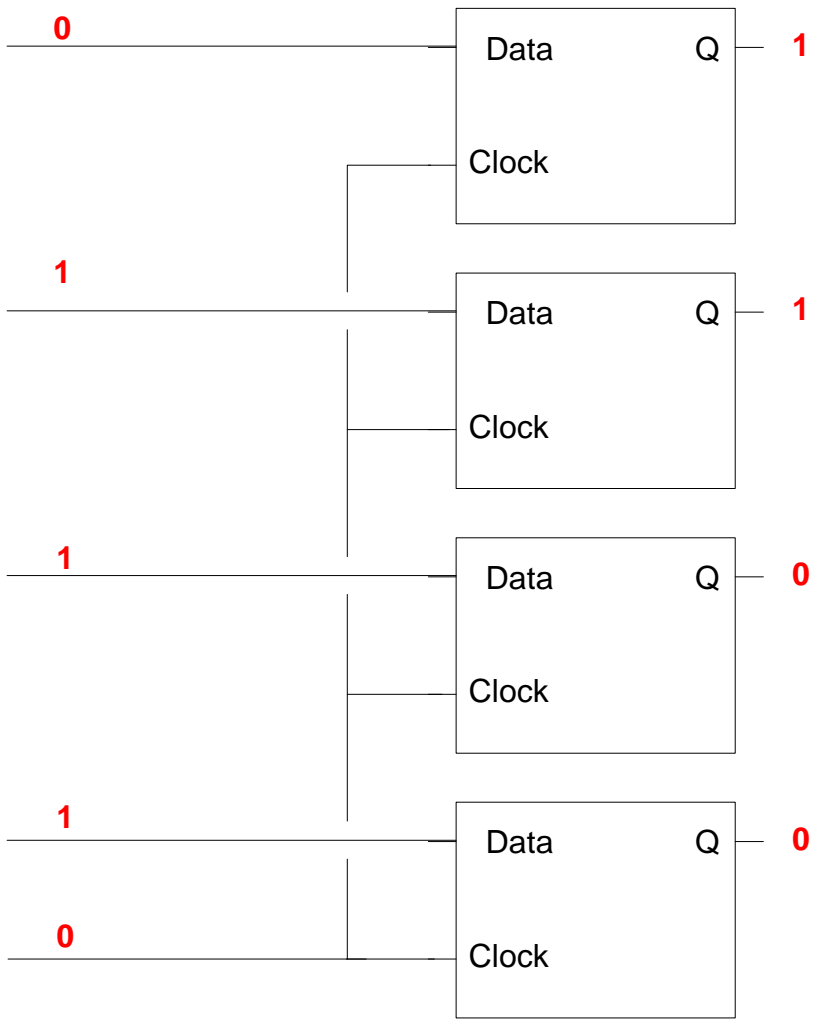


4-bit data latch

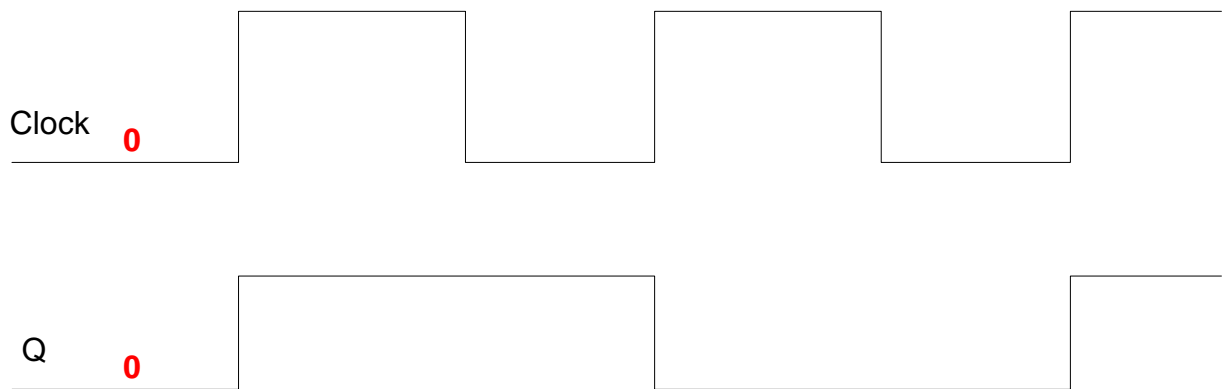




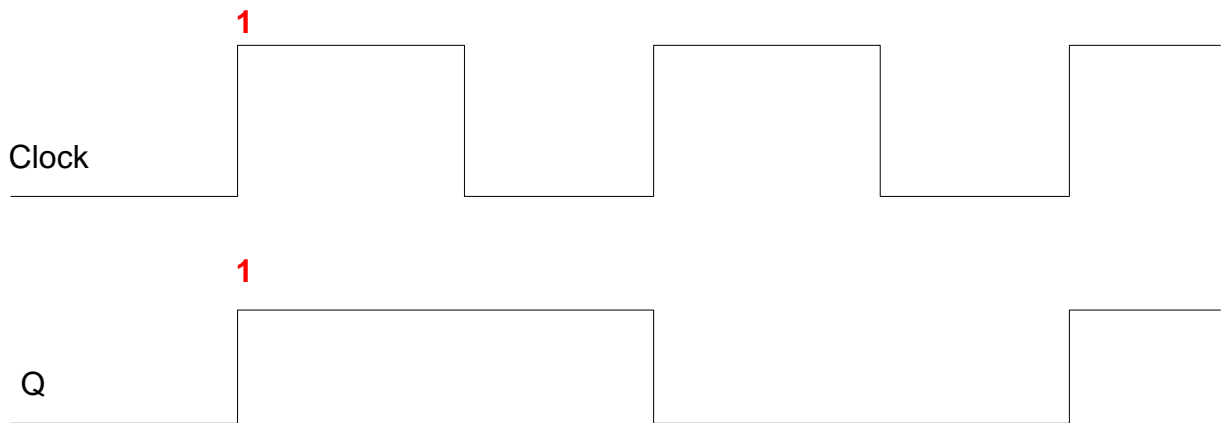
4-bit data latch



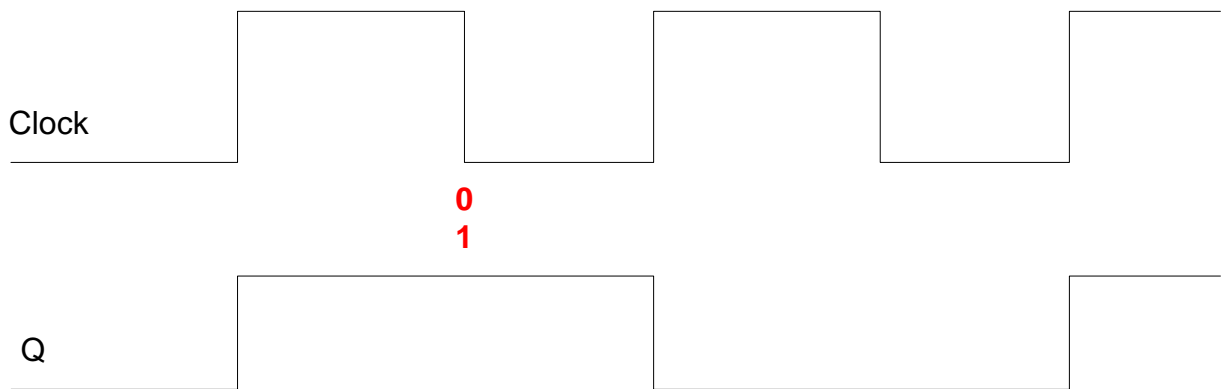
4-bit data latch



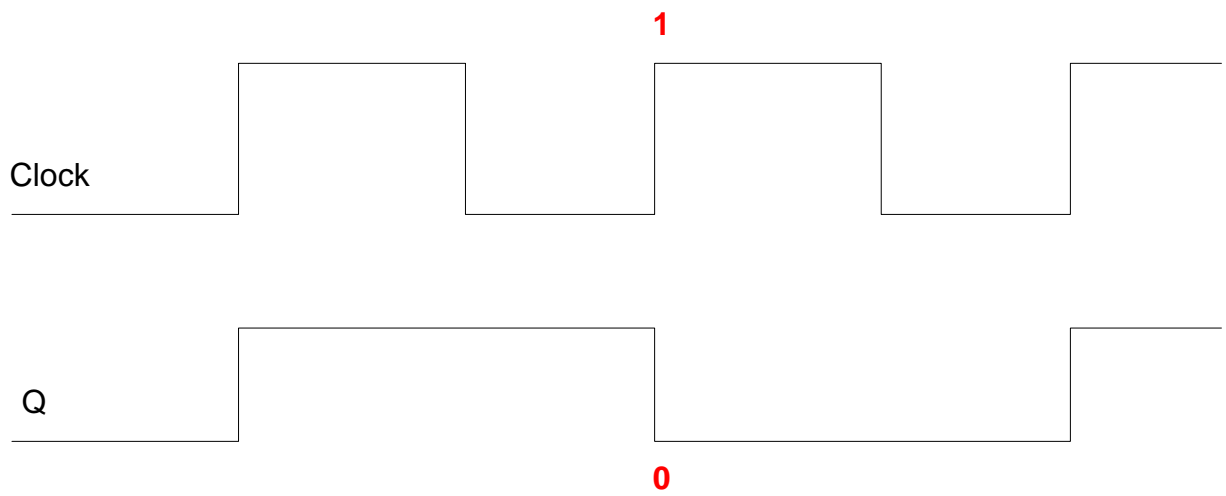
D Latch as Divide by 2



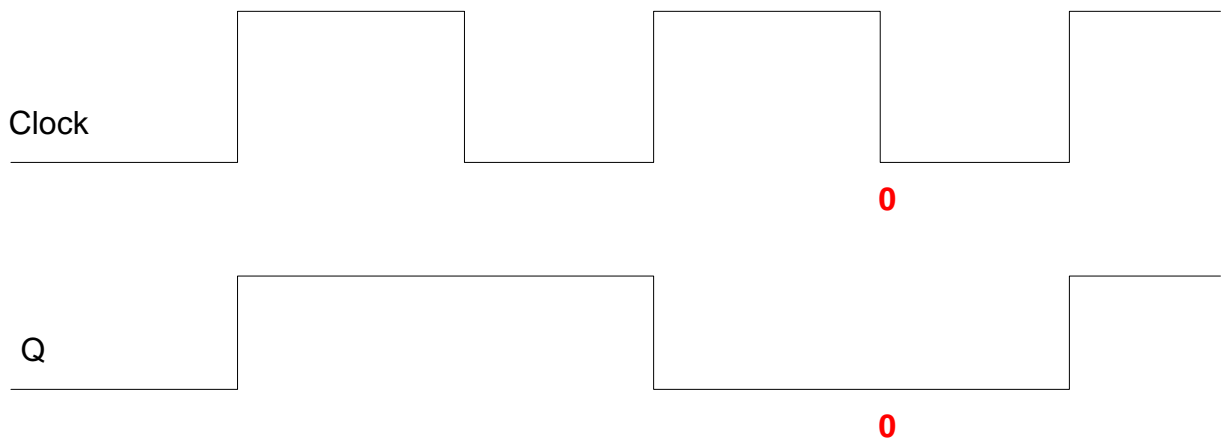
D Latch as Divide by 2



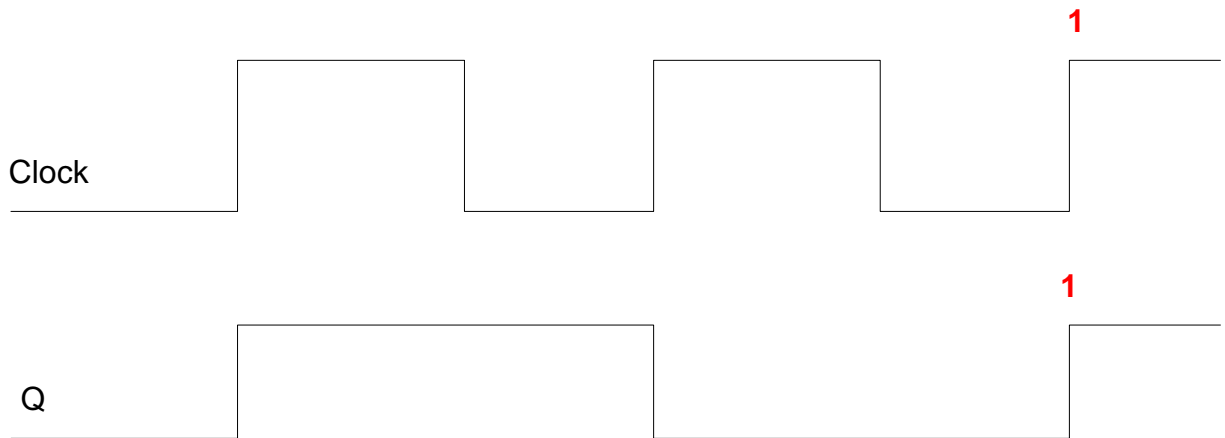
D Latch as Divide by 2



D Latch as Divide by 2

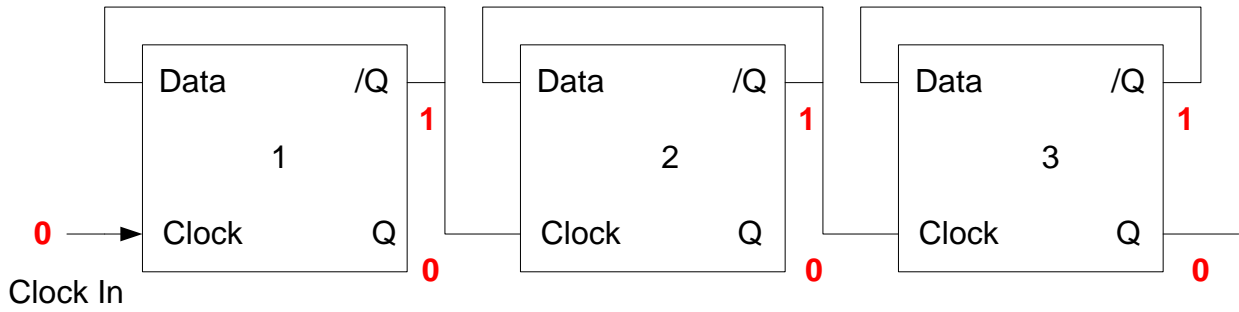


D Latch as Divide by 2

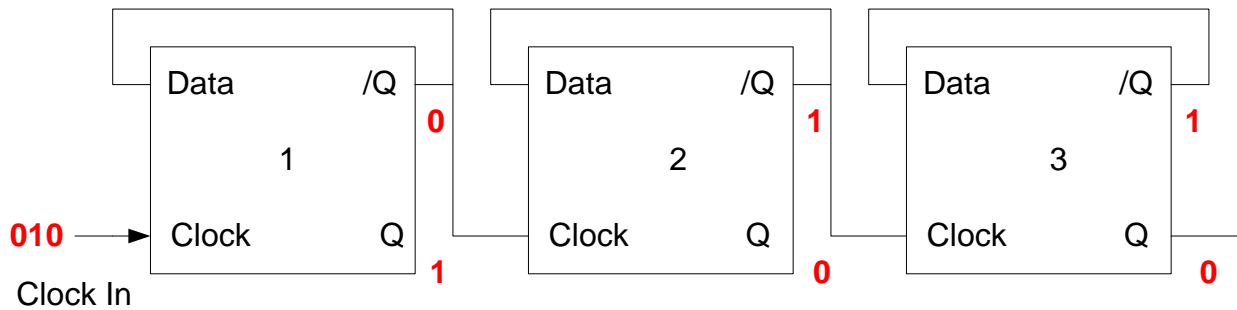


D Latch as Divide by 2

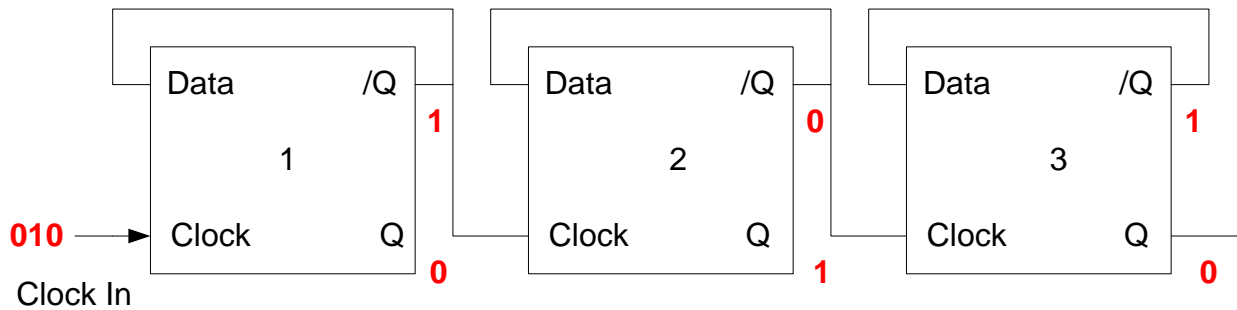




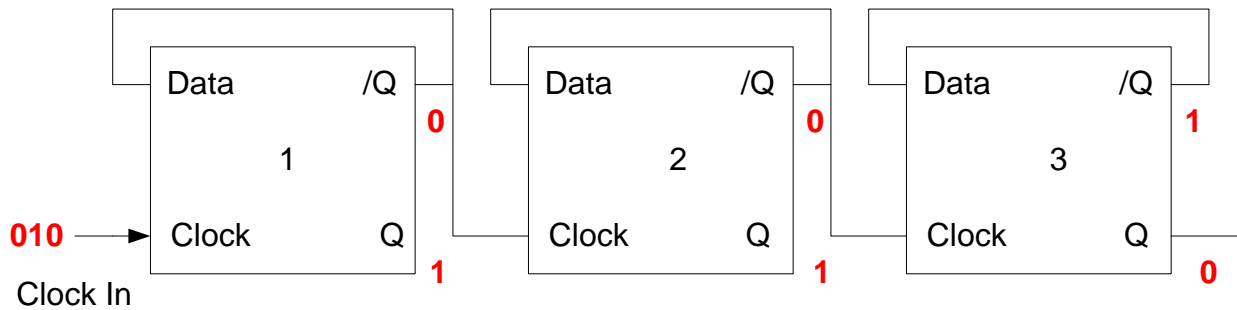
D latches as 3 stage binary counter



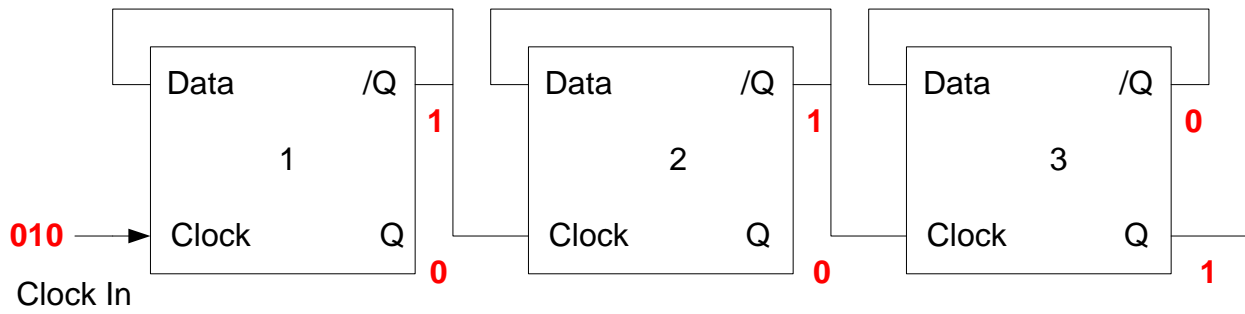
D latches as 3 stage binary counter



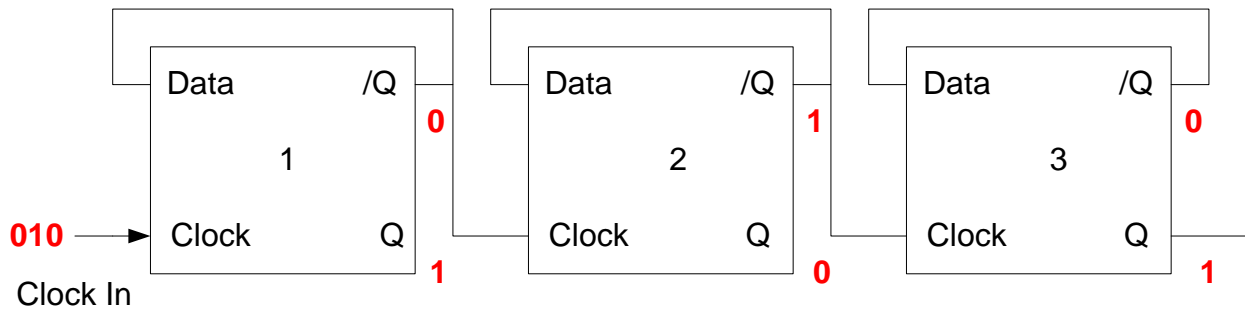
D latches as 3 stage binary counter



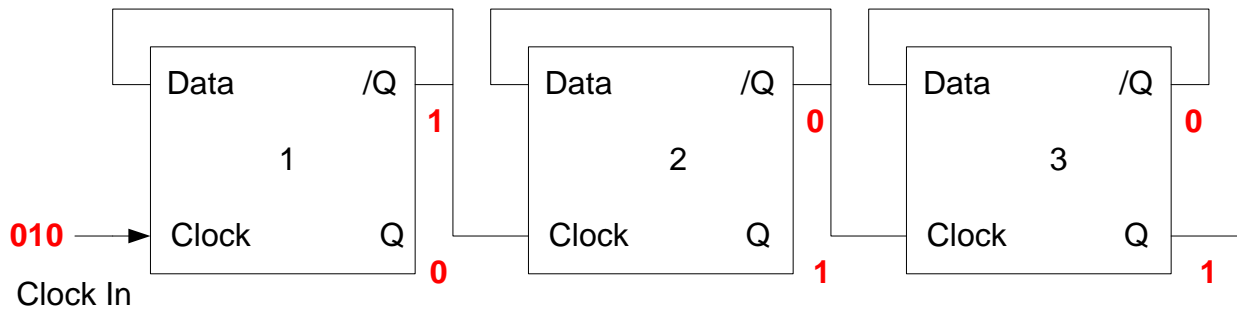
D latches as 3 stage binary counter



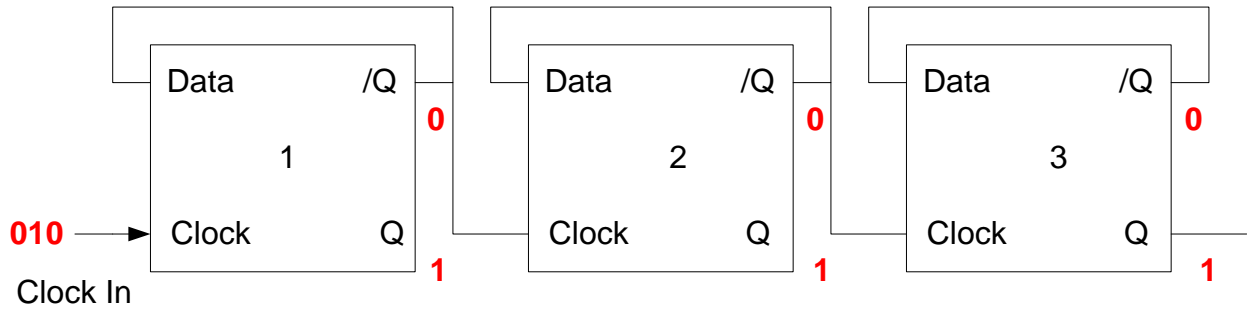
D latches as 3 stage binary counter



D latches as 3 stage binary counter

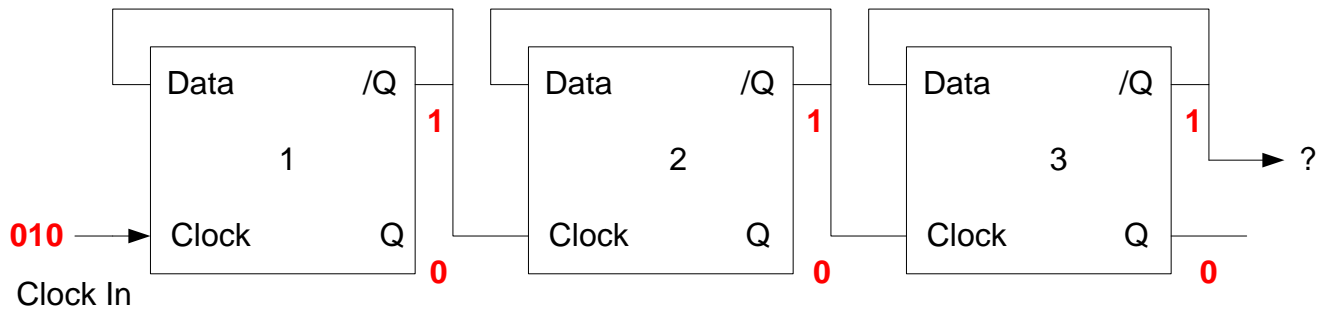


D latches as 3 stage binary counter



D latches as 3 stage binary counter





D latches as 3 stage binary counter

## Binary Arithmetic

### Reminder of Decimal Arithmetic

The 2 key facts are:-

\* Symbols available are 0 1 2 3 4 5 6 7 8 9

\* n 0 has the value  $10 \times n$  (e.g.  $40 = 4 \times 10$ ,  $500 = 5 \times 10 \times 10$ )

e.g

$$\begin{array}{r} 26 \\ + 37 \\ \hline 63 \end{array}$$

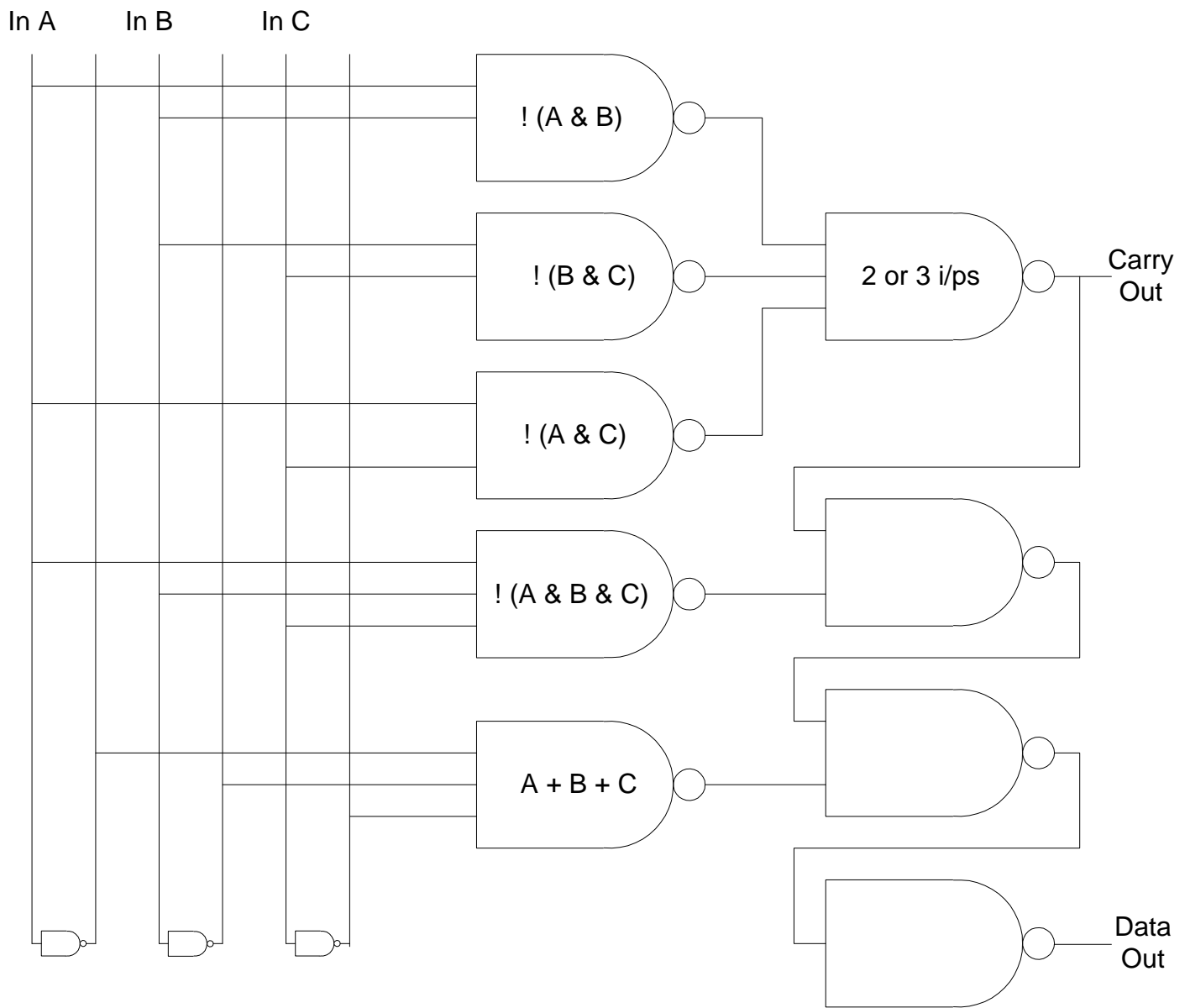
### Binary Arithmetic has equivalent facts:-

\* Symbols available are 0 1

\* n 0 has the value  $2 \times n$  (e.g.  $10 = 1 \times 2$ ,  $100 = 1 \times 2 \times 2$ )

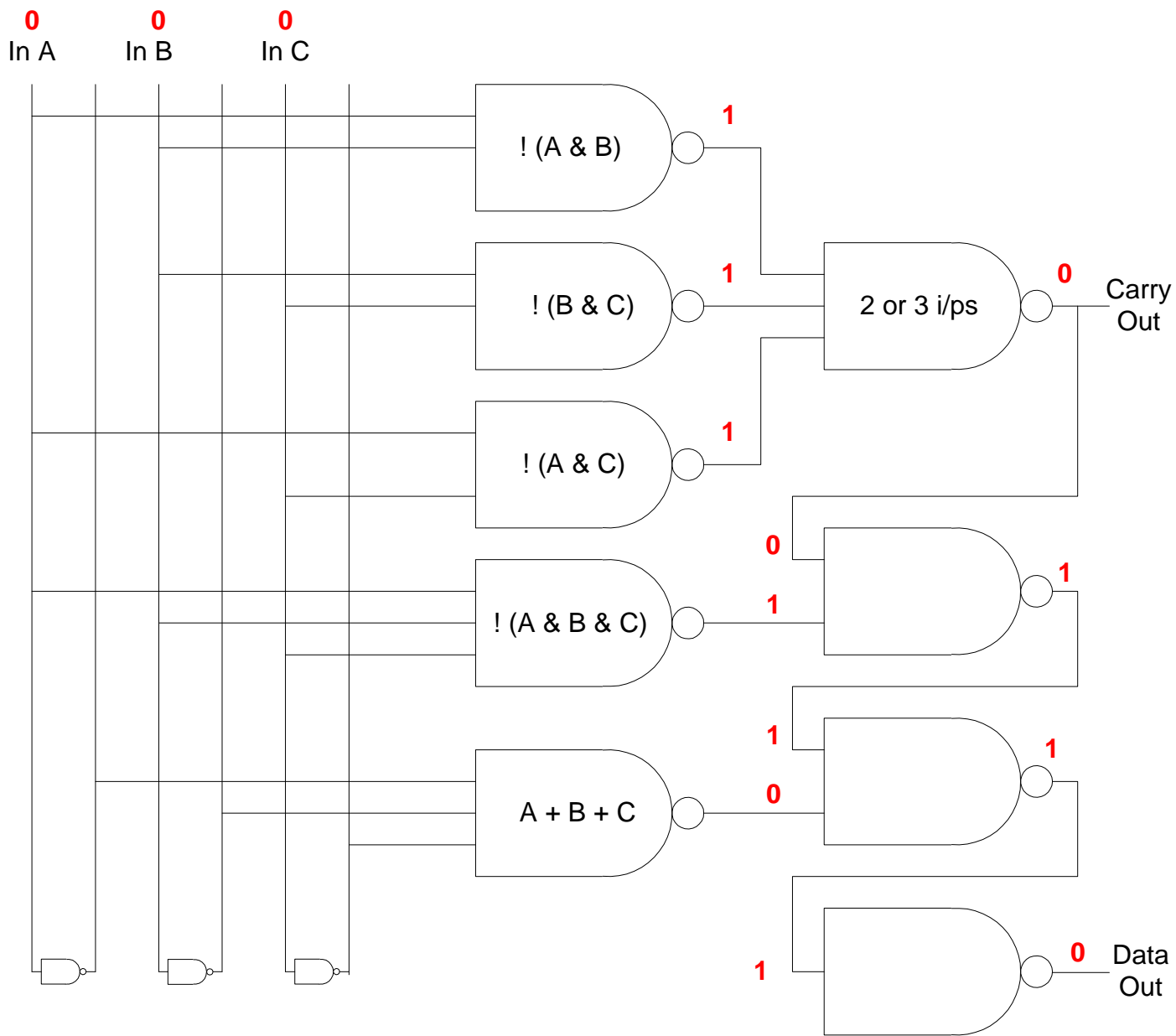
### Examples

Decimal	Binary
$\begin{array}{r} 2 \\ + 3 \\ \hline 5 \end{array}$	$\begin{array}{r} 10 \\ 11 \\ \hline 101 \end{array}$
$\begin{array}{r} 3 \\ * 2 \\ \hline 6 \end{array}$	$\begin{array}{r} 11 \\ 10 \\ \hline 110 \end{array}$
$\begin{array}{r} 5 \\ - 2 \\ \hline 3 \end{array}$	$\begin{array}{r} 101 \\ 10 \\ \hline 011 \end{array}$

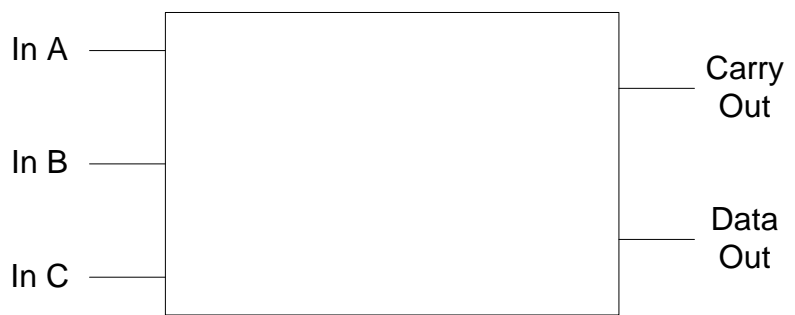


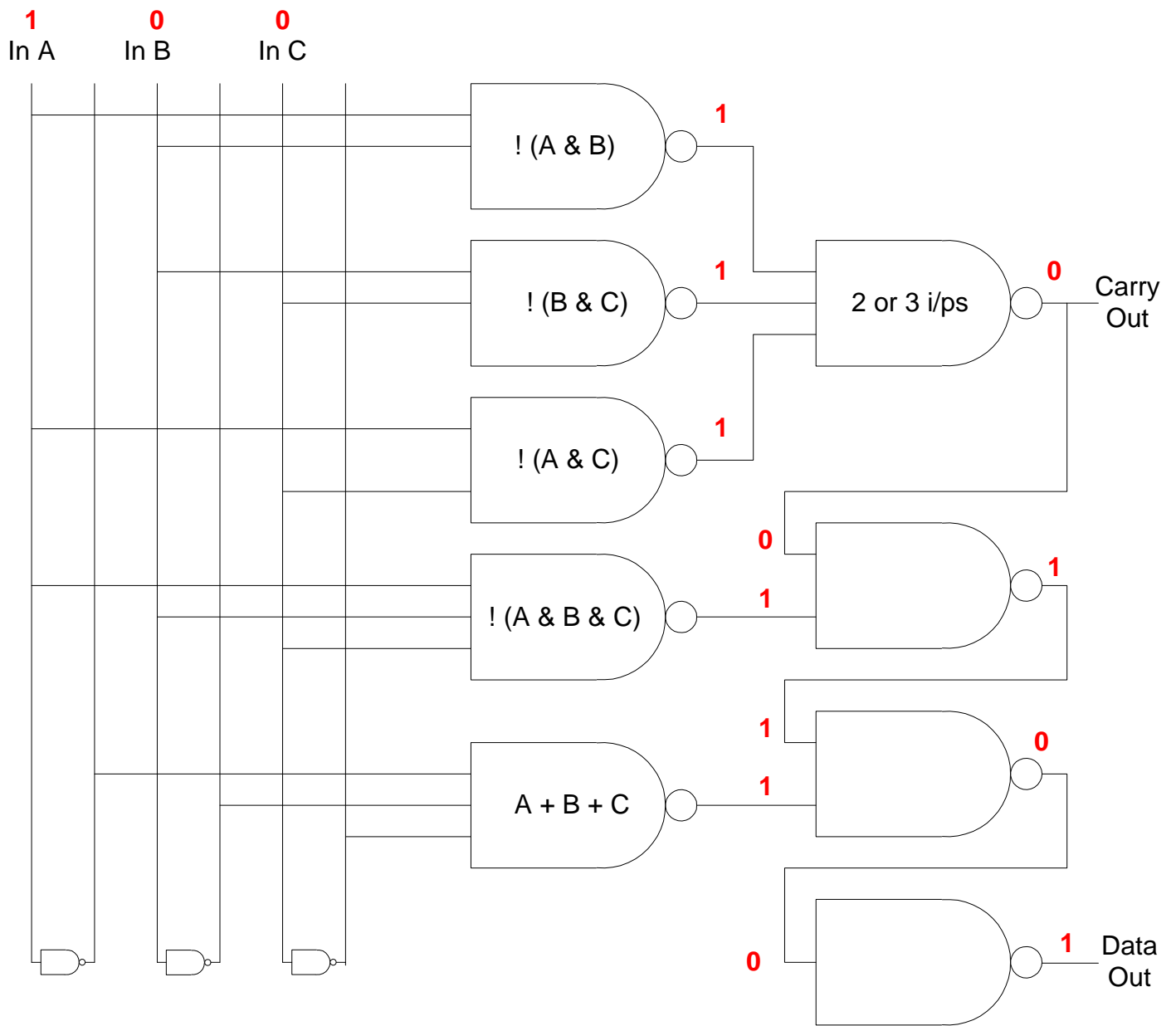
One-Bit Binary Adder (with carry)





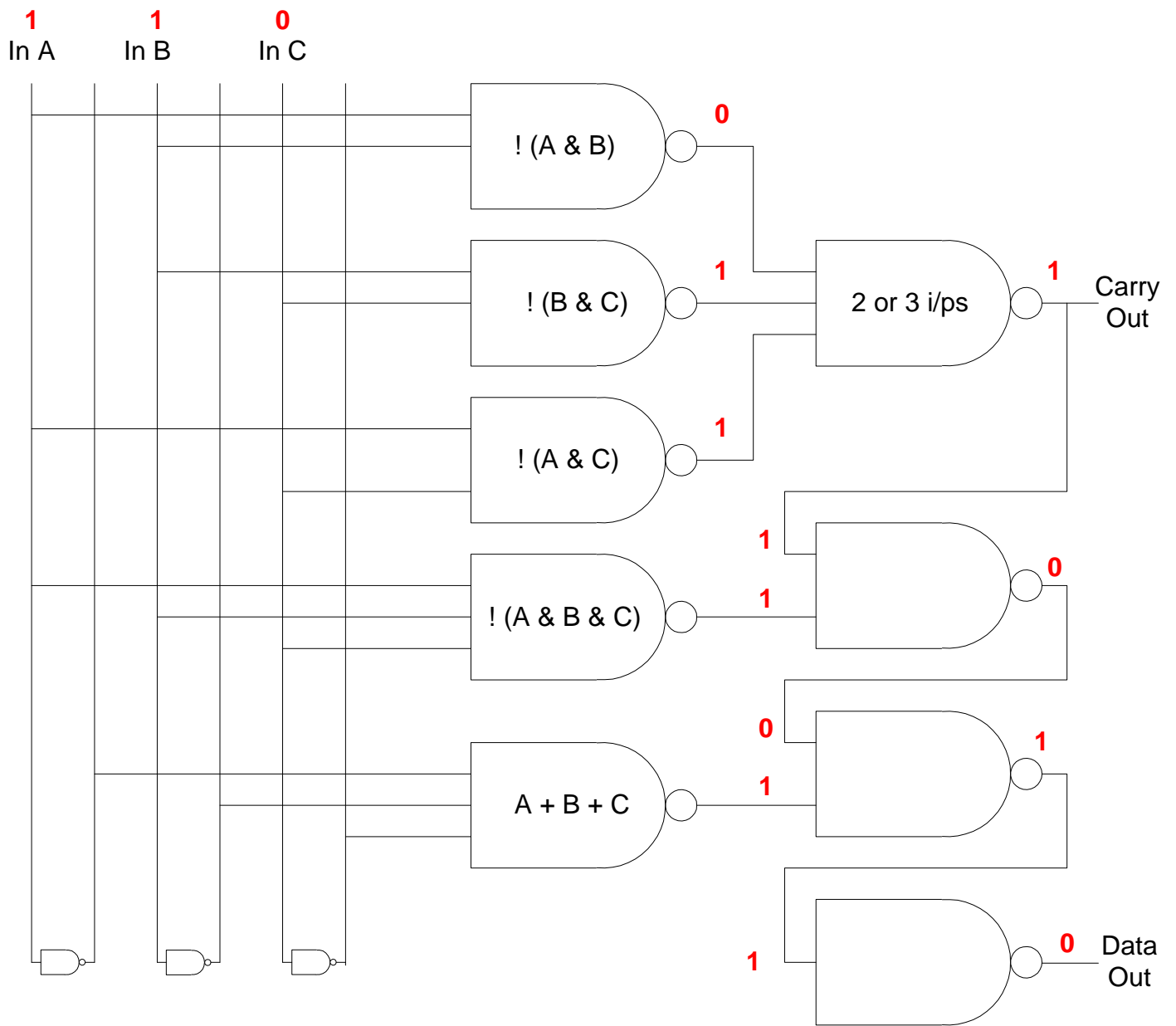
One-Bit Binary Adder (with carry)





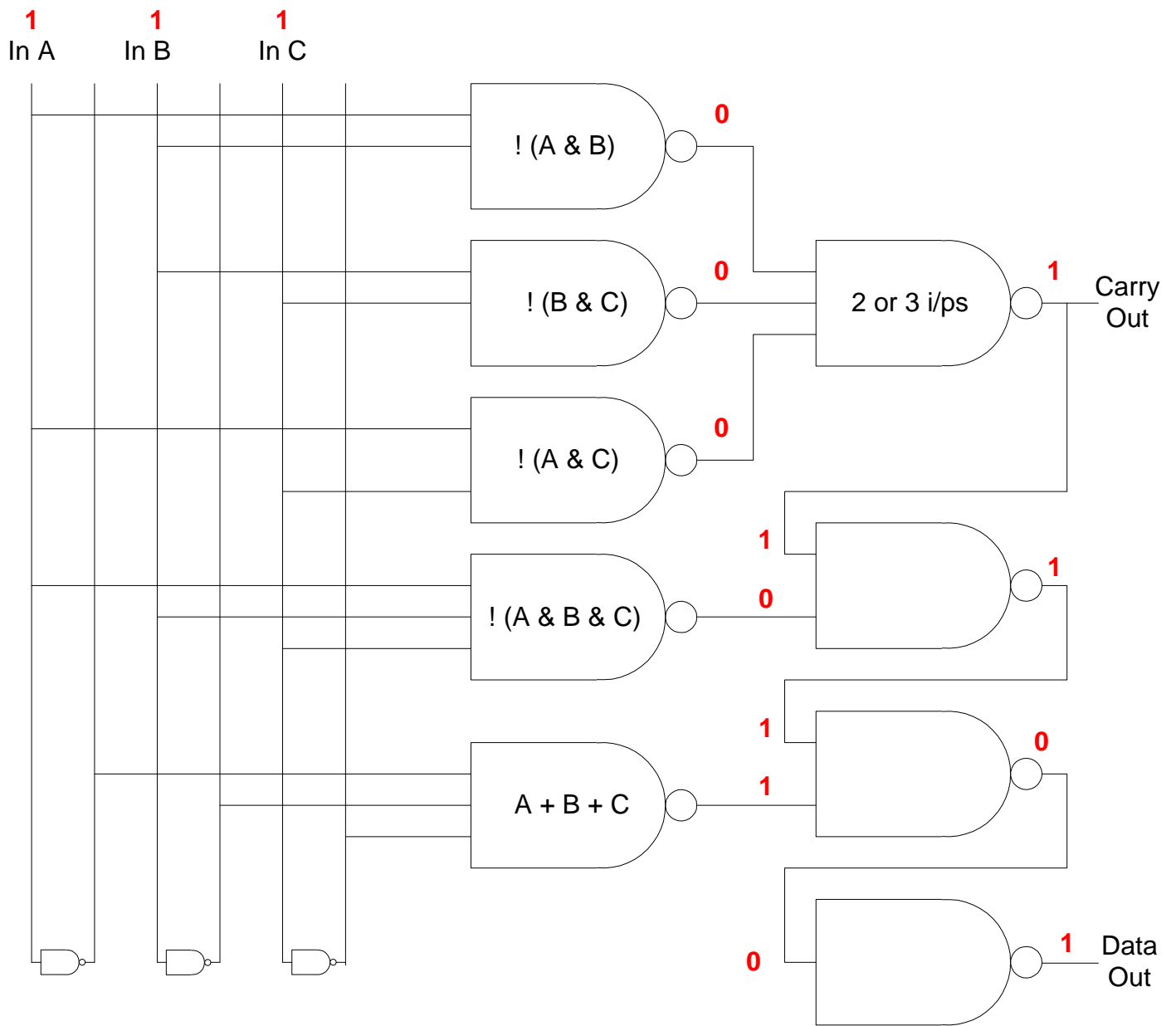
One-Bit Binary Adder (with carry)





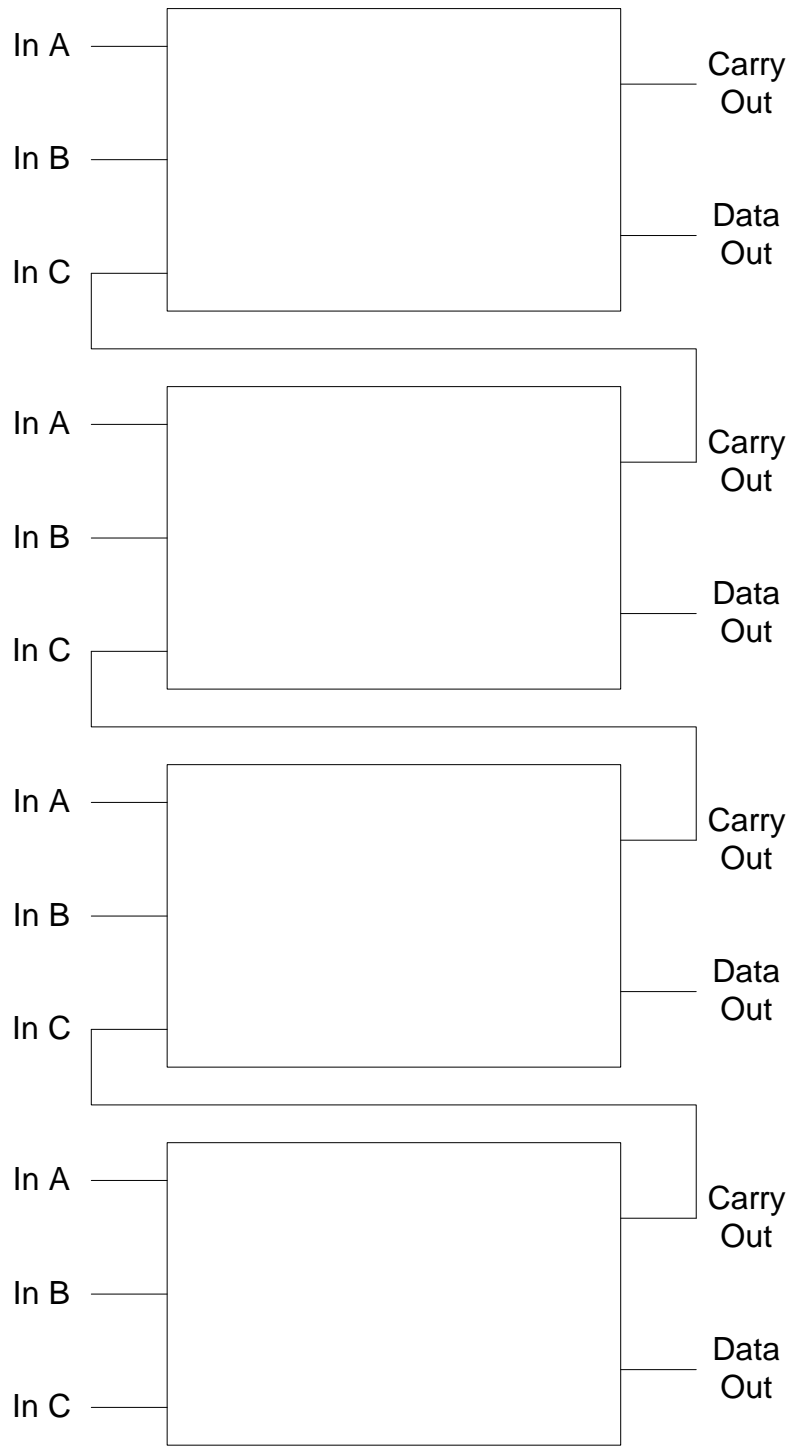
One-Bit Binary Adder (with carry)



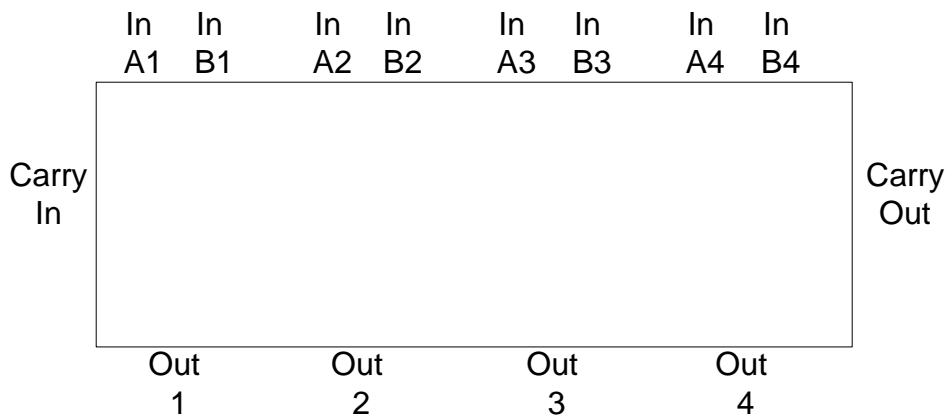


One-Bit Binary Adder (with carry)

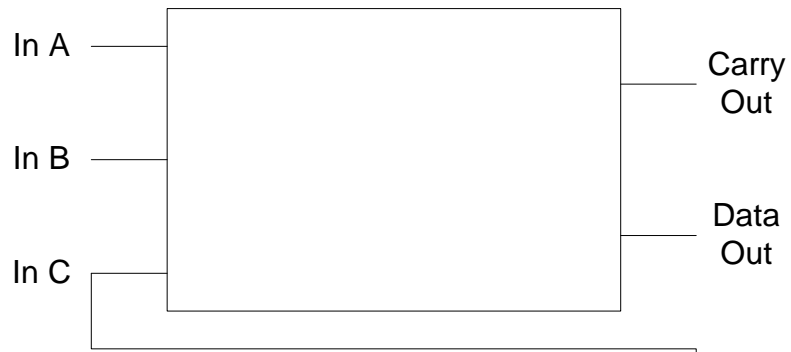




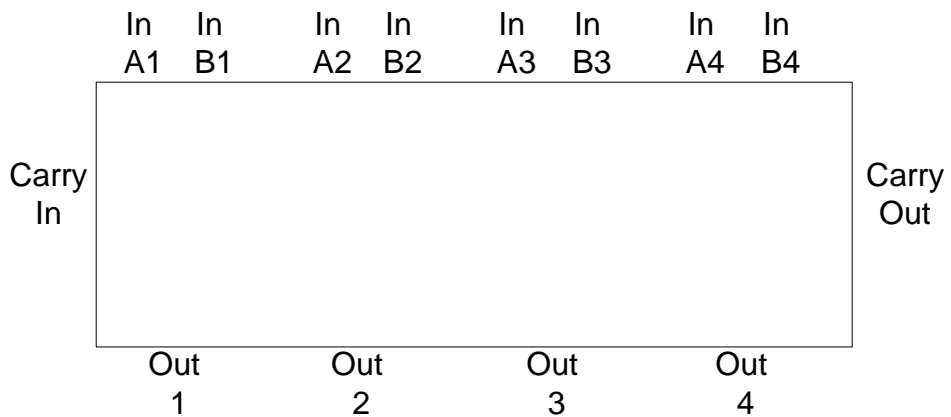
4-bit binary adder

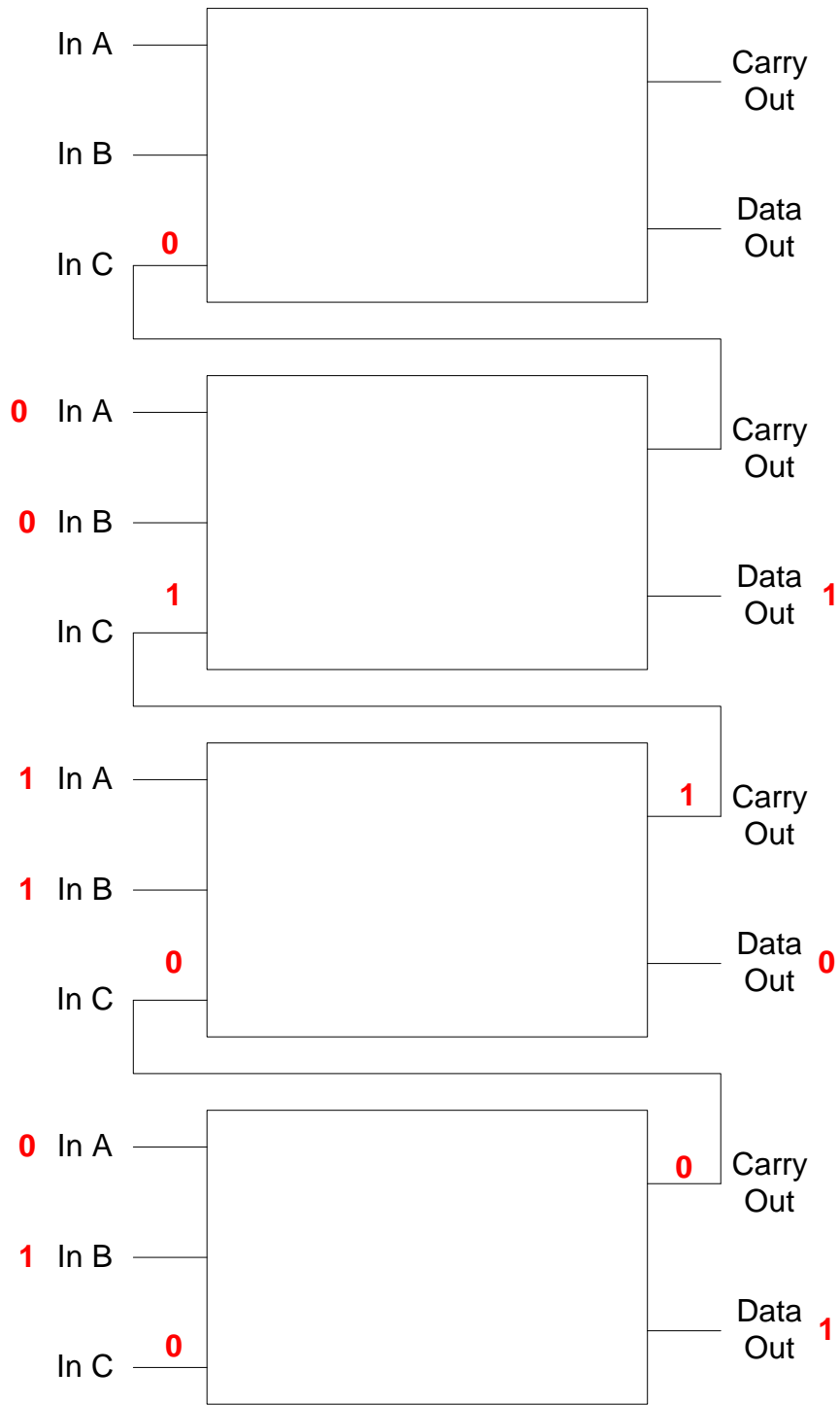




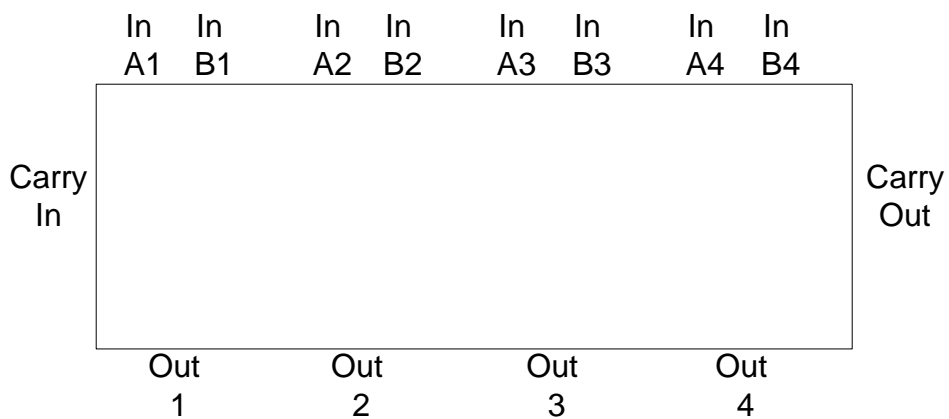


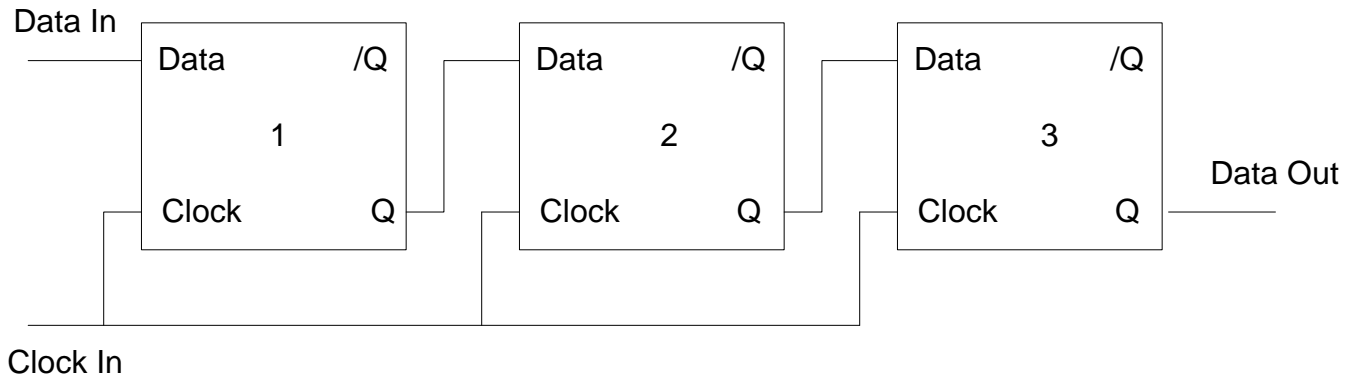
4-bit binary adder



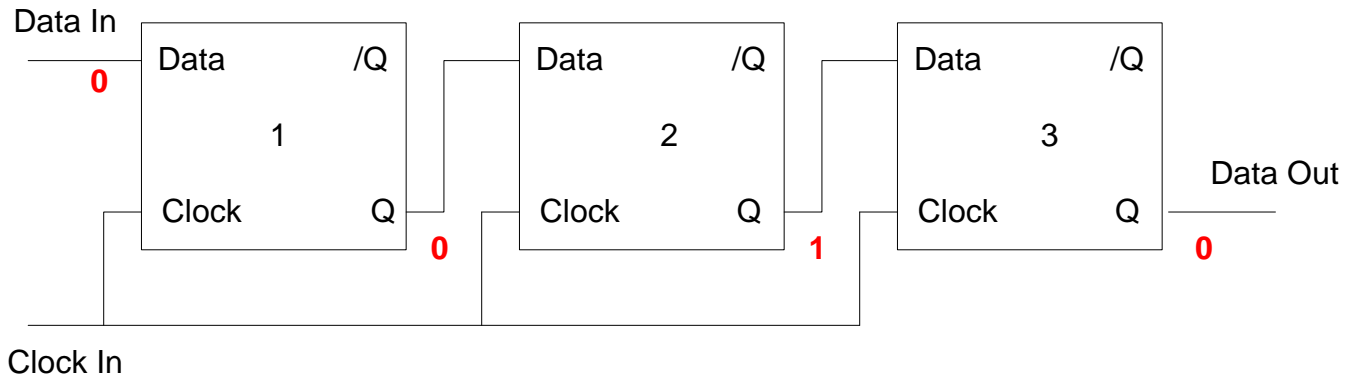


4-bit binary adder





D latches as 3 stage shift register



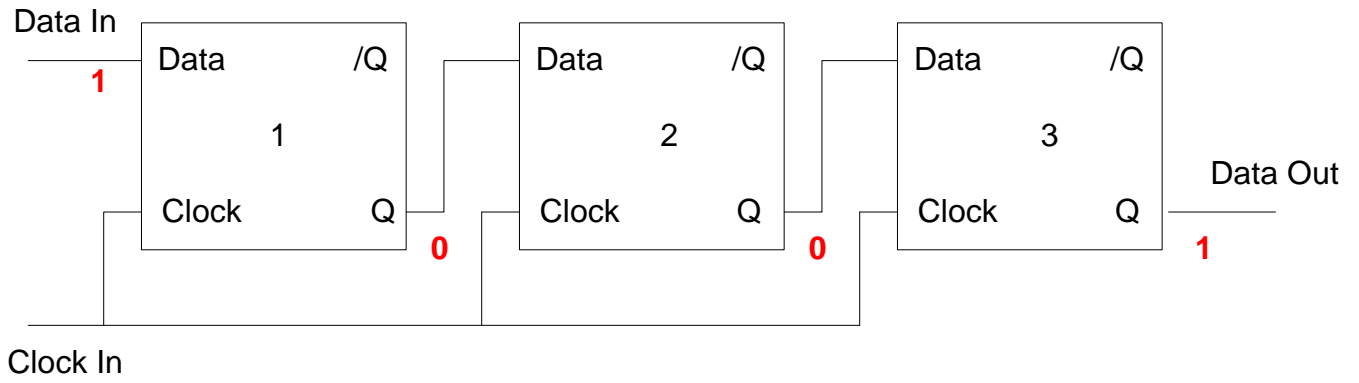
D latches as 3 stage shift register

0 0 1 0

1 0 0 1

1 1 0 0

0 1 1 0



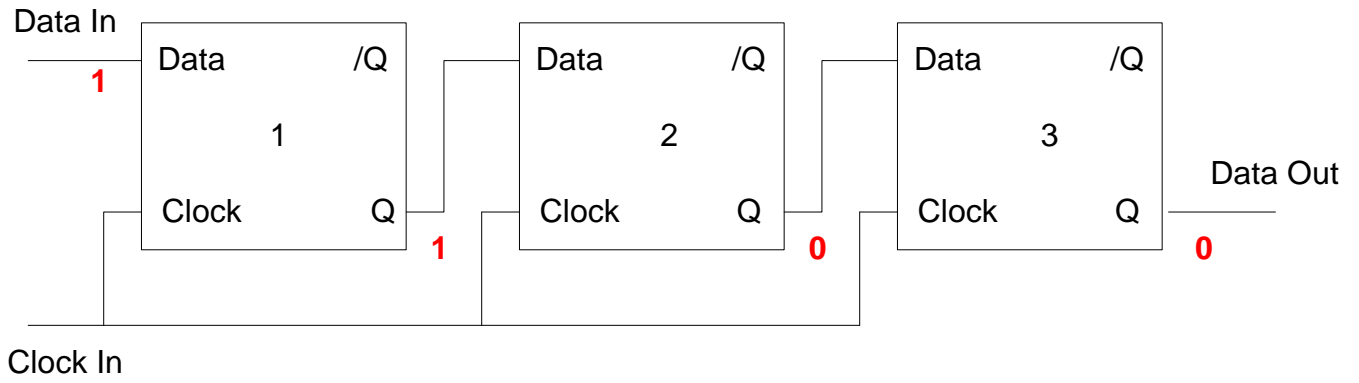
D latches as 3 stage shift register

0 0 1 0

**1 0 0 1**

1 1 0 0

0 1 1 0



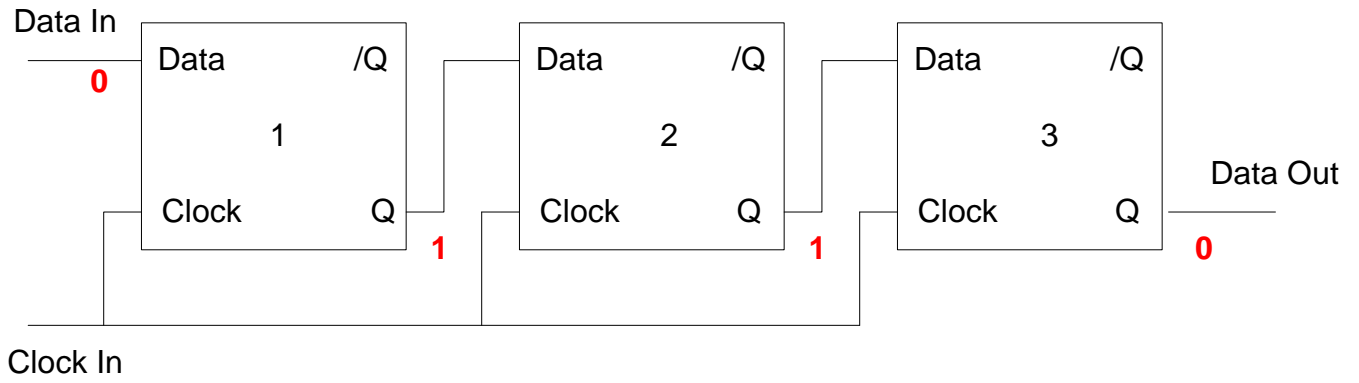
D latches as 3 stage shift register

0 0 1 0

1 0 0 1

**1 1 0 0**

0 1 1 0



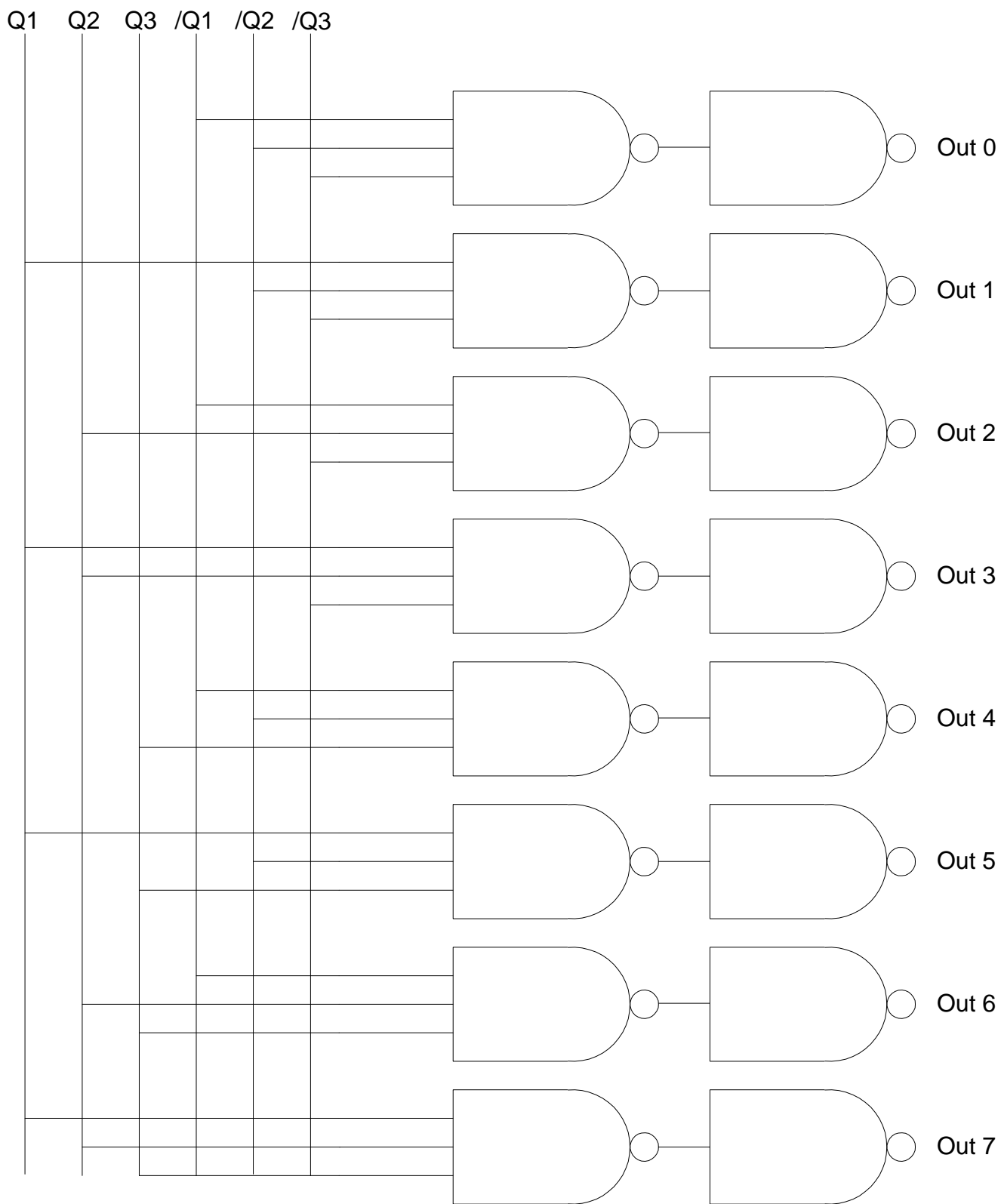
D latches as 3 stage shift register

0 0 1 0

1 0 0 1

1 1 0 0

**0 1 1 0**



3-bit decoder