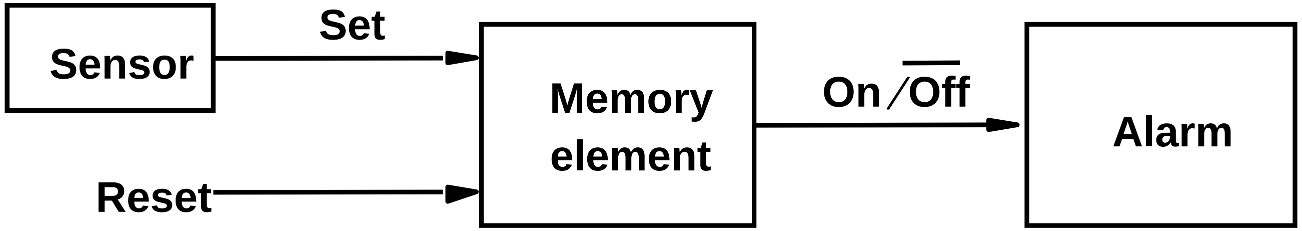
Lecture 7
Flip-Flops, Registers, Counters, and a Simple Processor

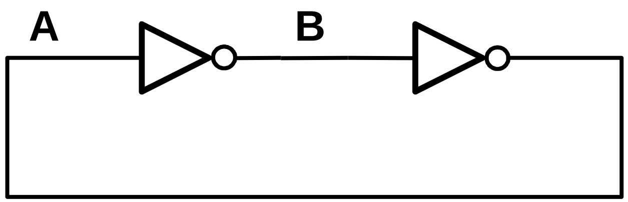
吳文中

“States” in a Circuit

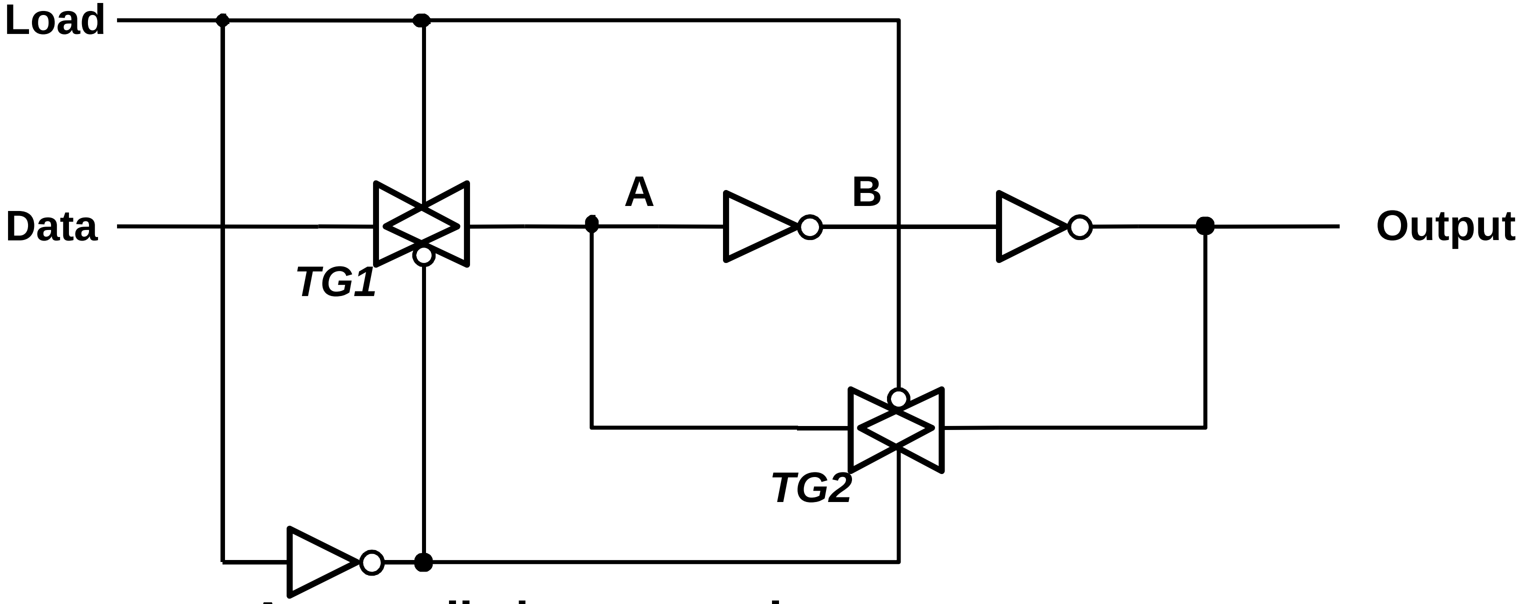


Control of an alarm system.

Memory Element Implementations



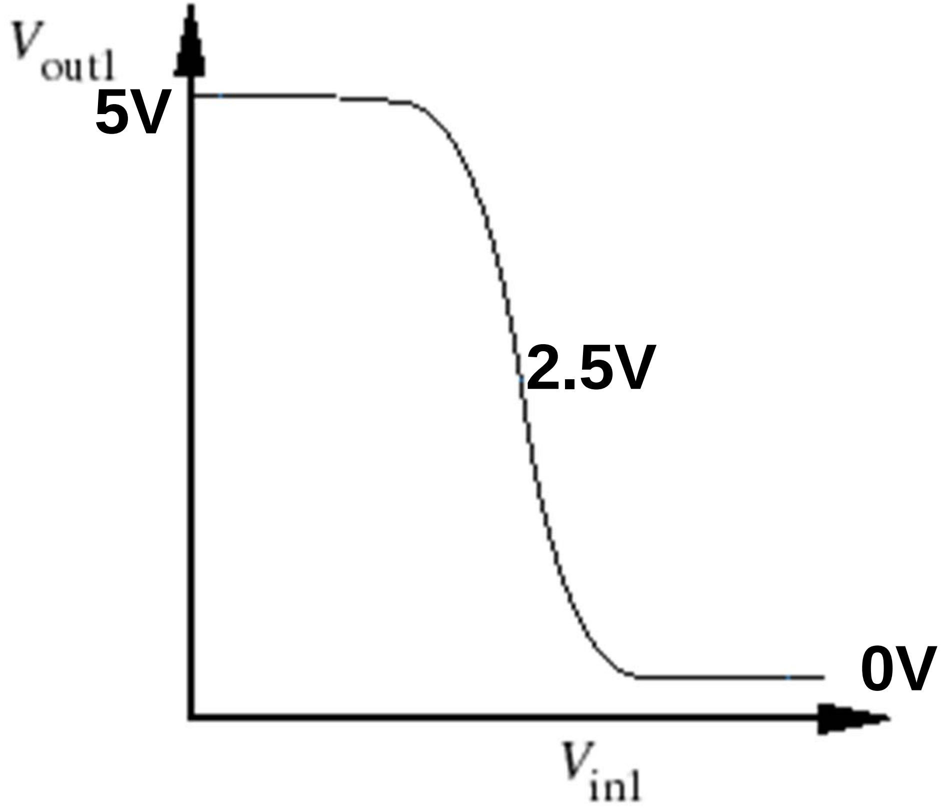
A simple memory element.



A controlled memory element.

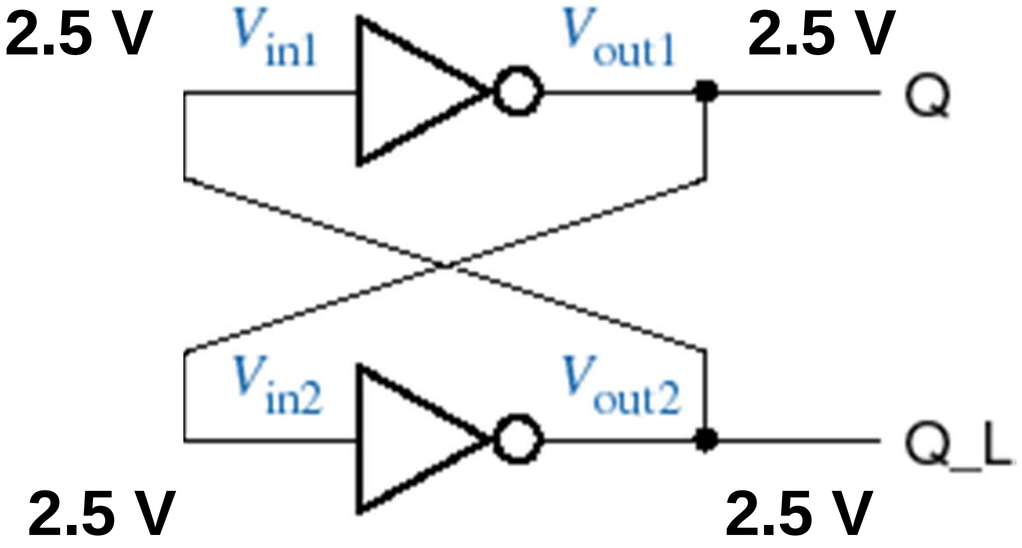
Analog analysis

- Assume pure CMOS thresholds, 5V rail
- Theoretical threshold center is 2.5 V



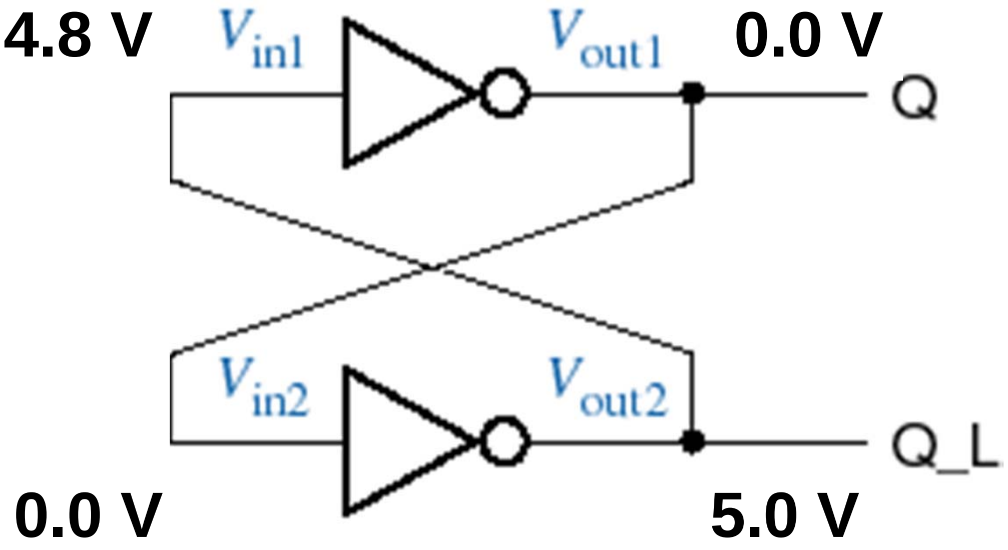
Analog analysis

- Assume pure CMOS thresholds, 5V rail
- Theoretical threshold center is 2.5 V



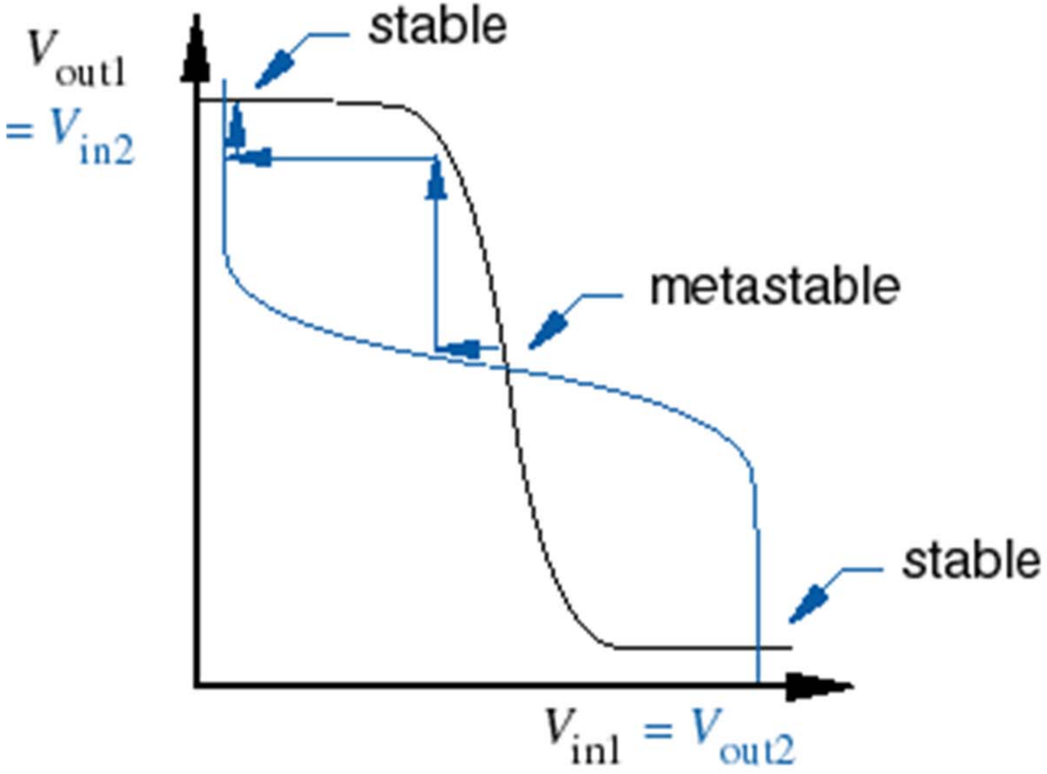
Analog analysis

- Assume pure CMOS thresholds, 5V rail
- Theoretical threshold center is 2.5 V



Metastability

- Metastability is inherent in any bistable circuit



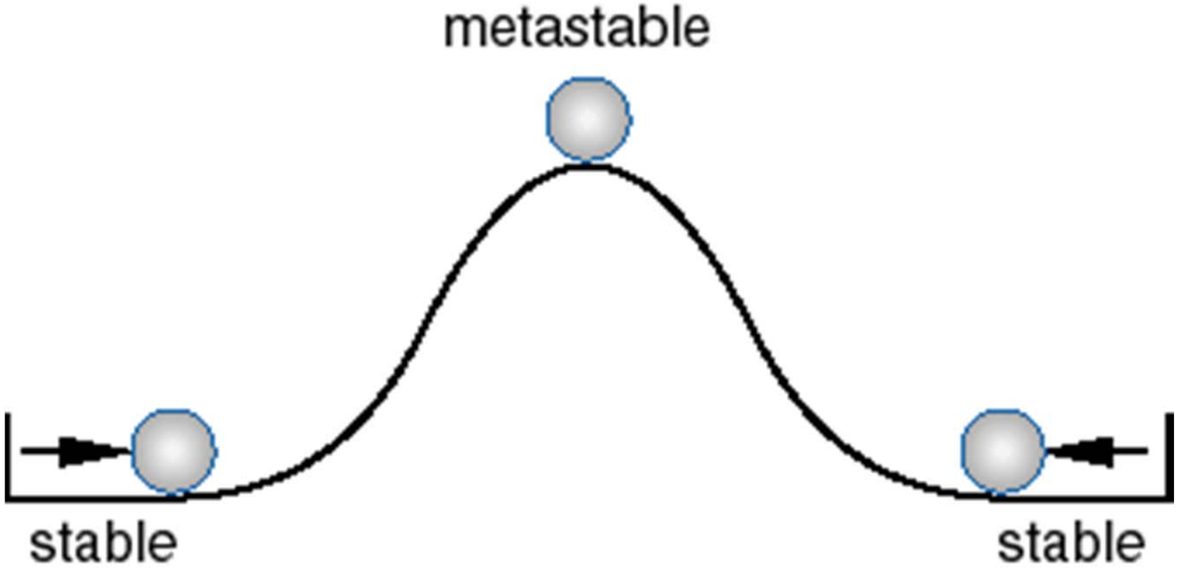
Transfer function:

$$V_{out1} = T(V_{in1})$$

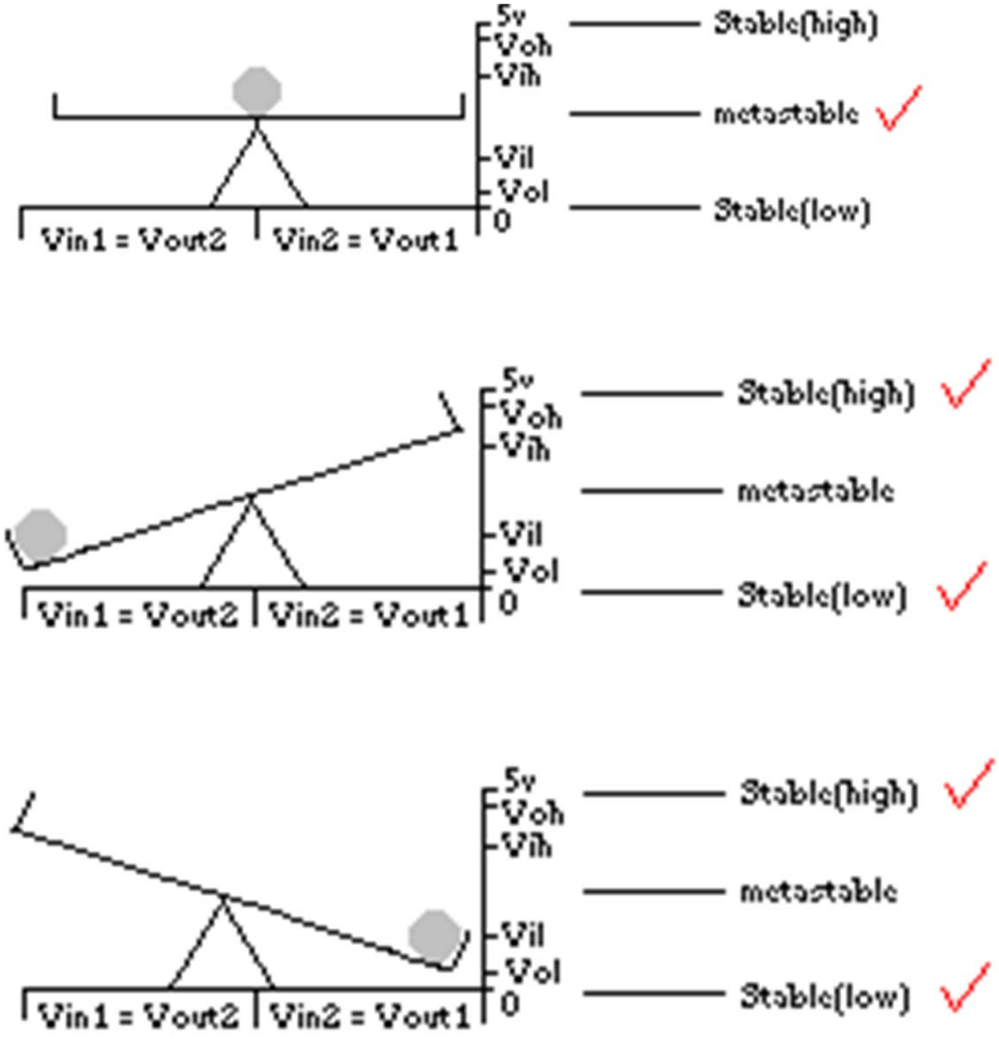
$$V_{out2} = T(V_{in2})$$

- Two stable points, one metastable point

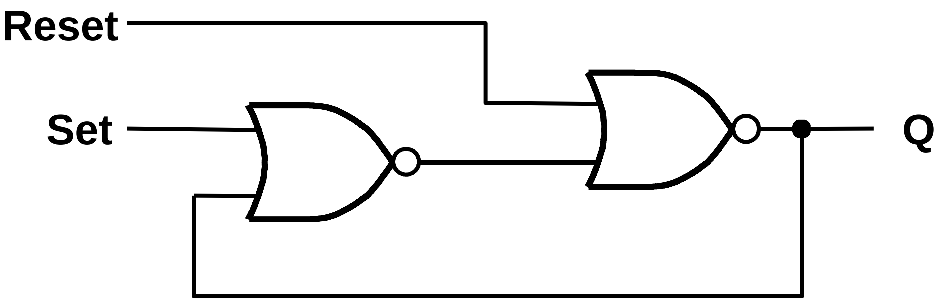
Another look at metastability



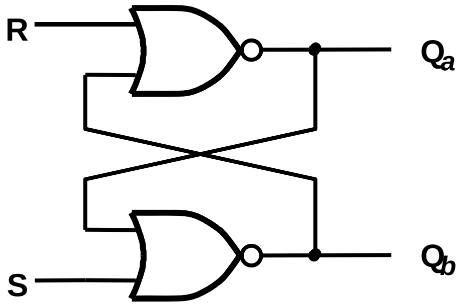
Stable and Metastable



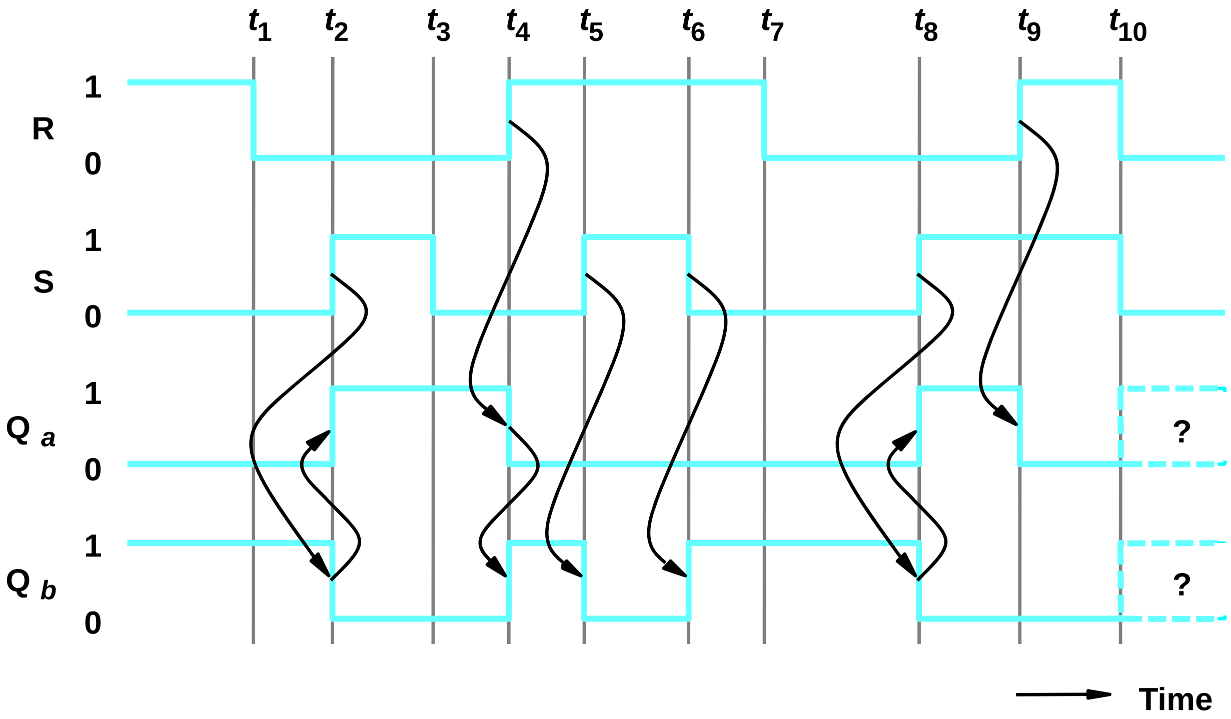
BASIC (RS) Latch



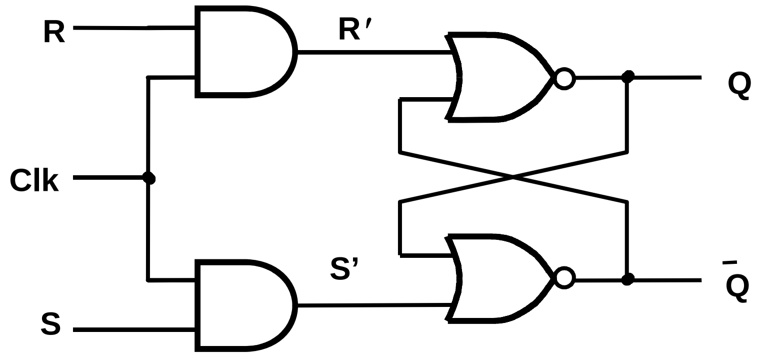
A Basic Latch Built with NOR Gates



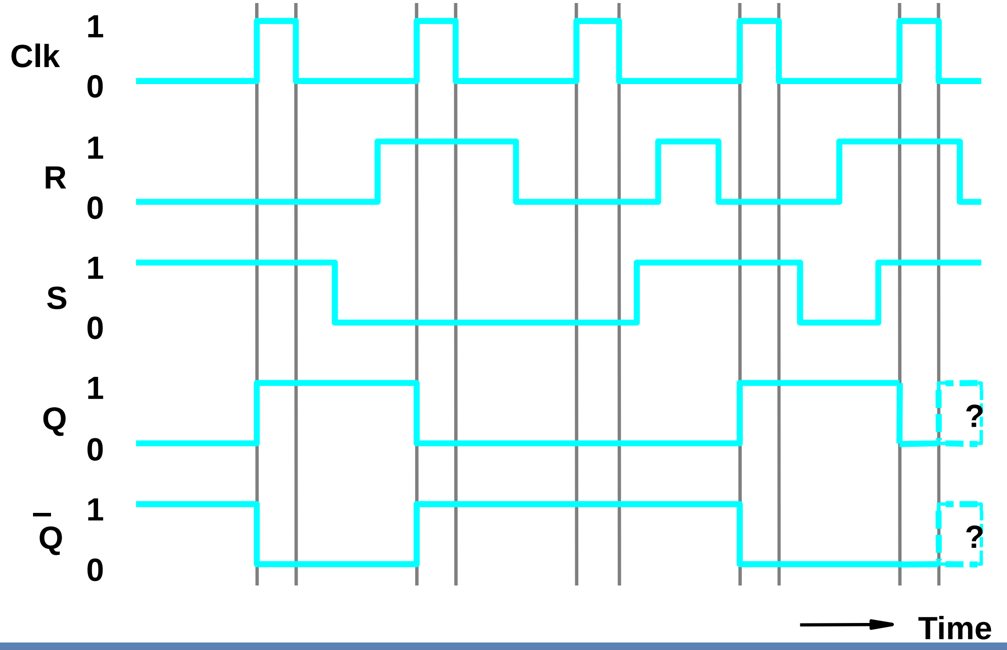
S	R	Q_a	Q_b	
0	0	0/1	1/0	(no change)
0	1	0	1	
1	0	1	0	
1	1	0	0	

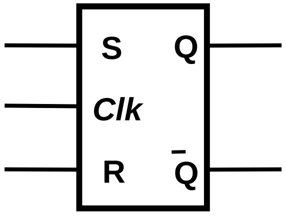


Gates SR Latch

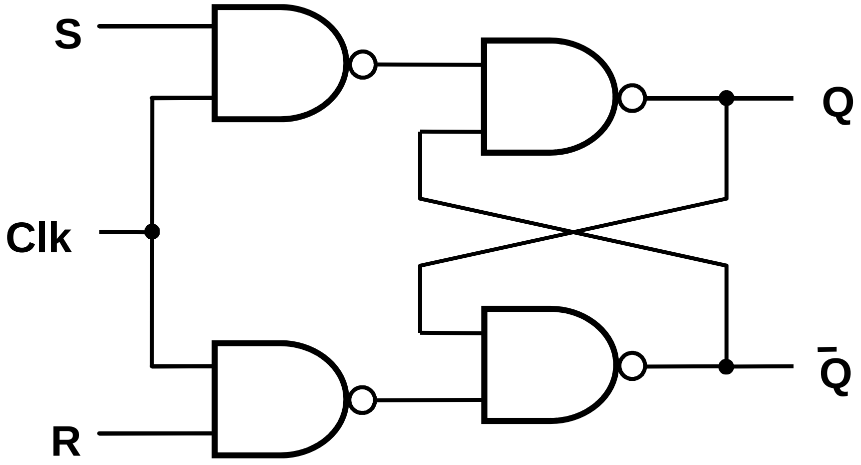


Clk	S	R	$Q(t+1)$
0	x	x	Q(*t*) (no change)
1	0	0	Q(*t*) (no change)
1	0	1	0
1	1	0	1
1	1	1	x



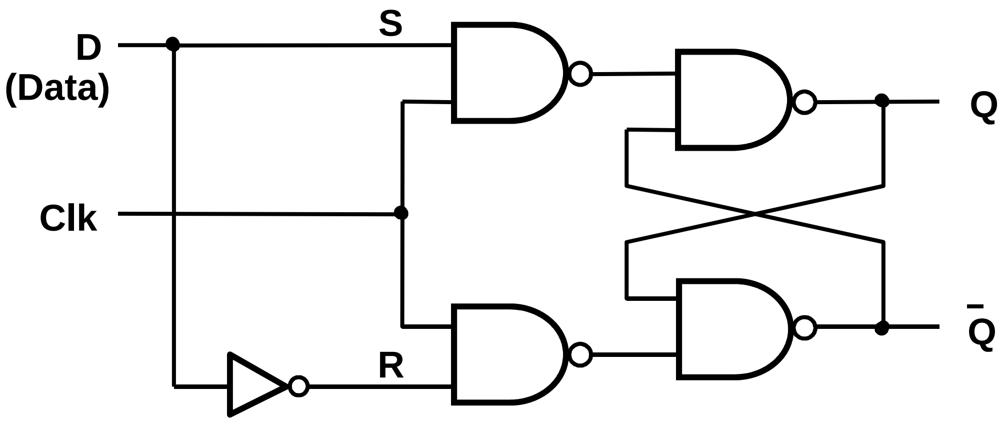


Gated SR Latch with NAND Gates

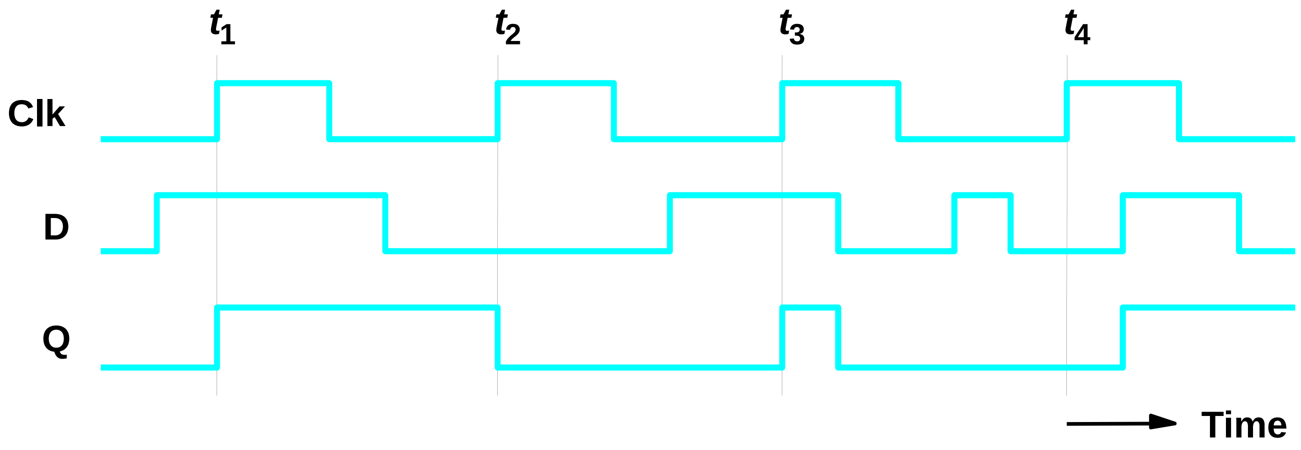


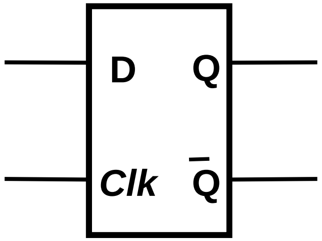
Clk	S	R	$Q(t+1)$
0	x	x	Q(*t*) (no change)
1	0	0	Q(*t*) (no change)
1	0	1	0
1	1	0	1
1	1	1	x

Gated D Latch



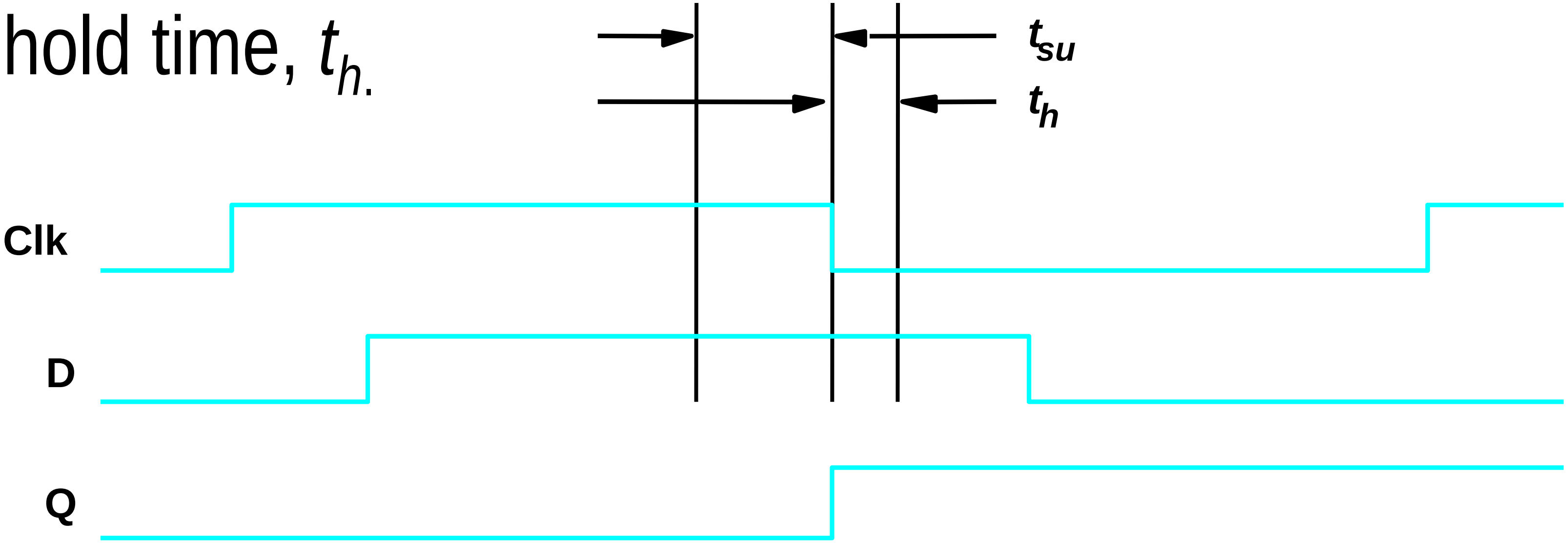
Clk	D	$Q(t+1)$
0	x	$Q(t)$
1	0	0
1	1	1



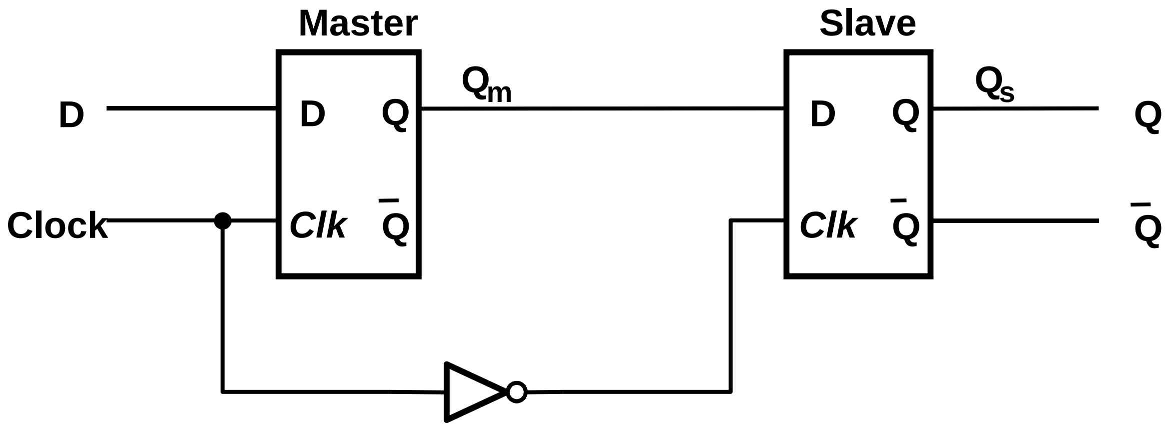


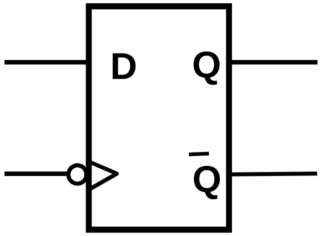
Effects of Propagation Delays

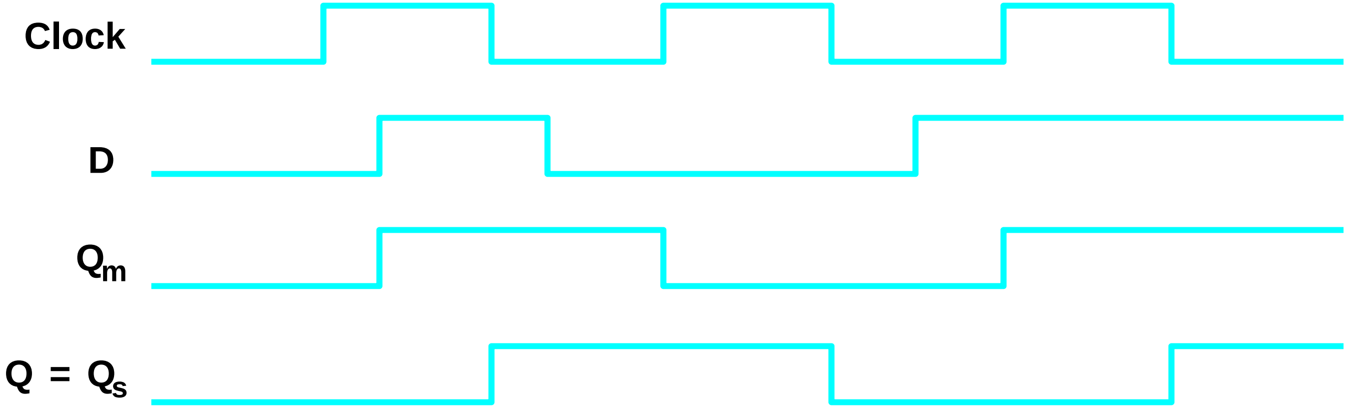
- The minimum time that the *D* signal must be stable prior to the negative edge of the *Clk* signal is called setup time, t_{su}.
- The minimum time that the *D* signal must remain stable after the negative edge of the *Clk* signal is called hold time, t_h.



Master-Slave D Flip-Flop

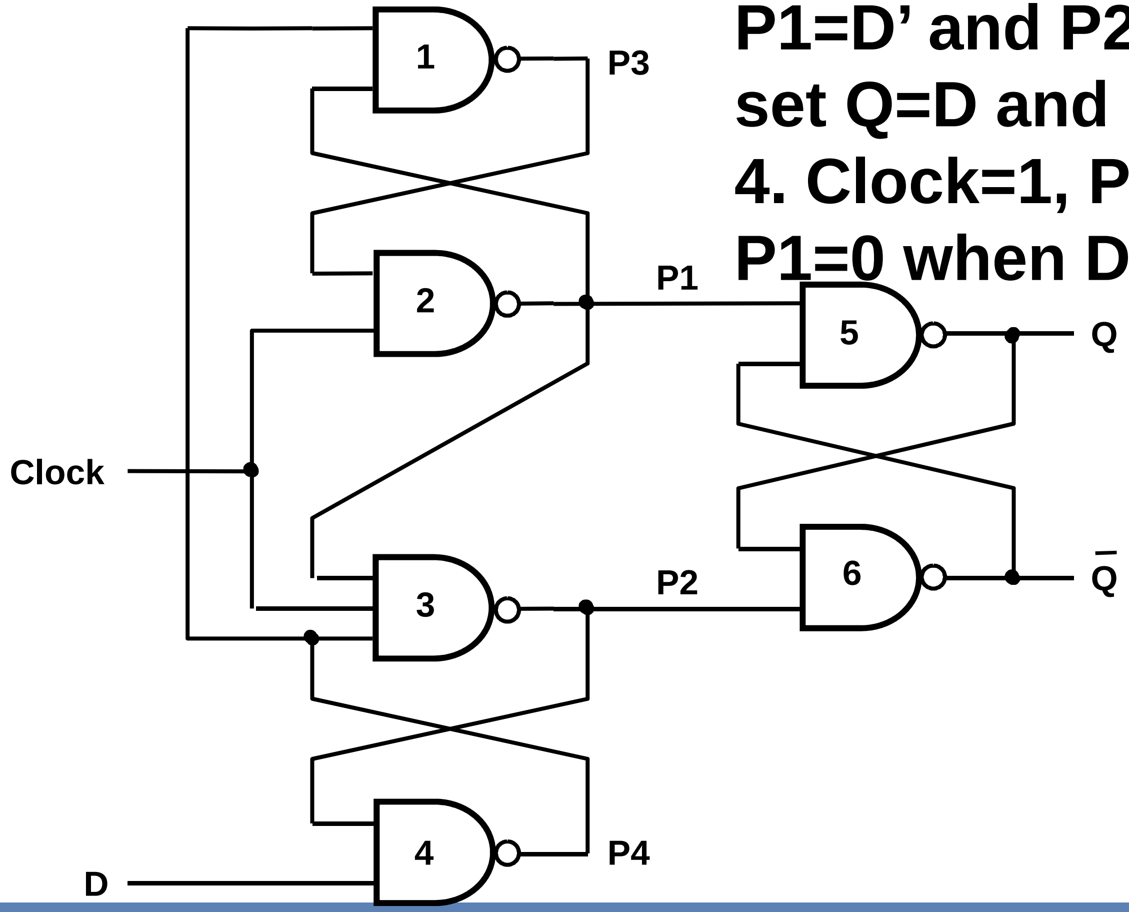


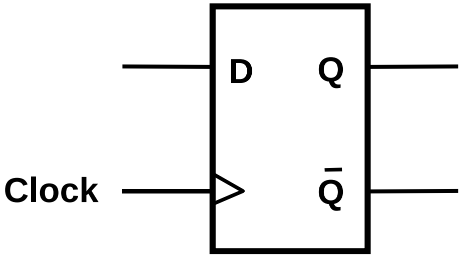




A Positive-edge-triggered D Flip-Flop

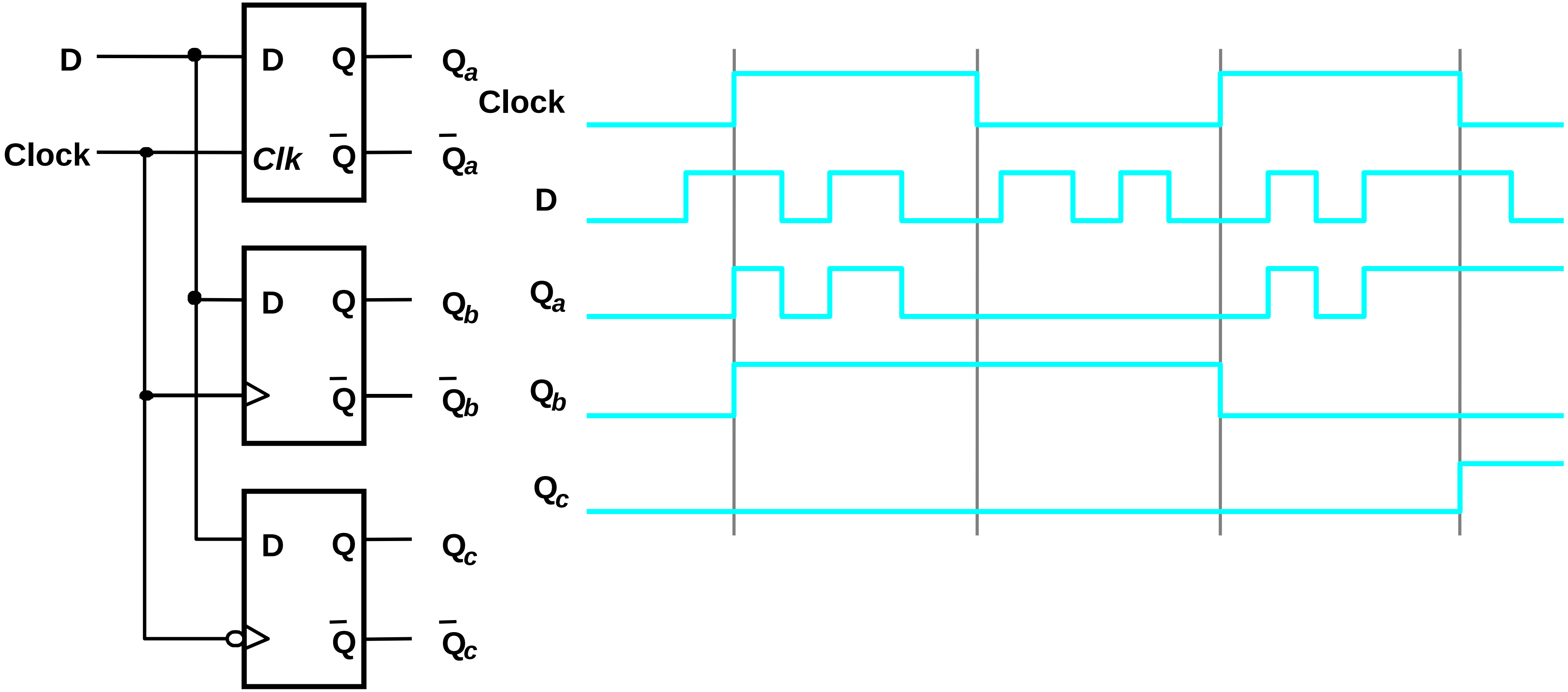
1.Clock=0, P1=P2=1; Latch Hold.
2. P4=D’ and P3 = D
3. Clock 0 to 1 (positive edge), P1=D’ and P2=D which set Q=D and Q’=D’
4. Clock=1, P2=0 when D=0; P1=0 when D=1



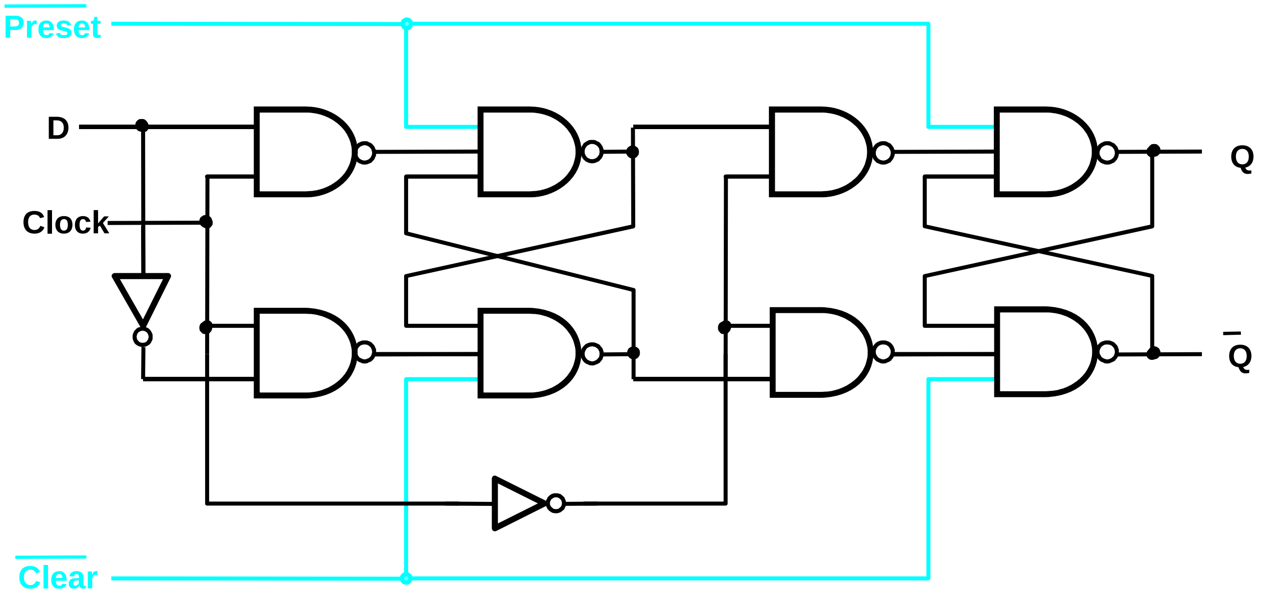


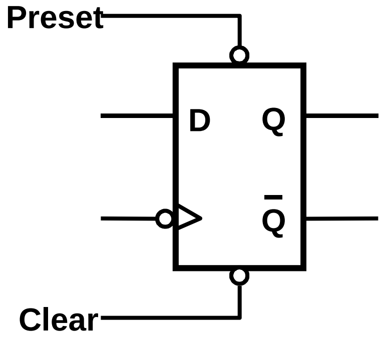
(b) Graphical symbol

Level-Sensitive versus Edge-Triggered

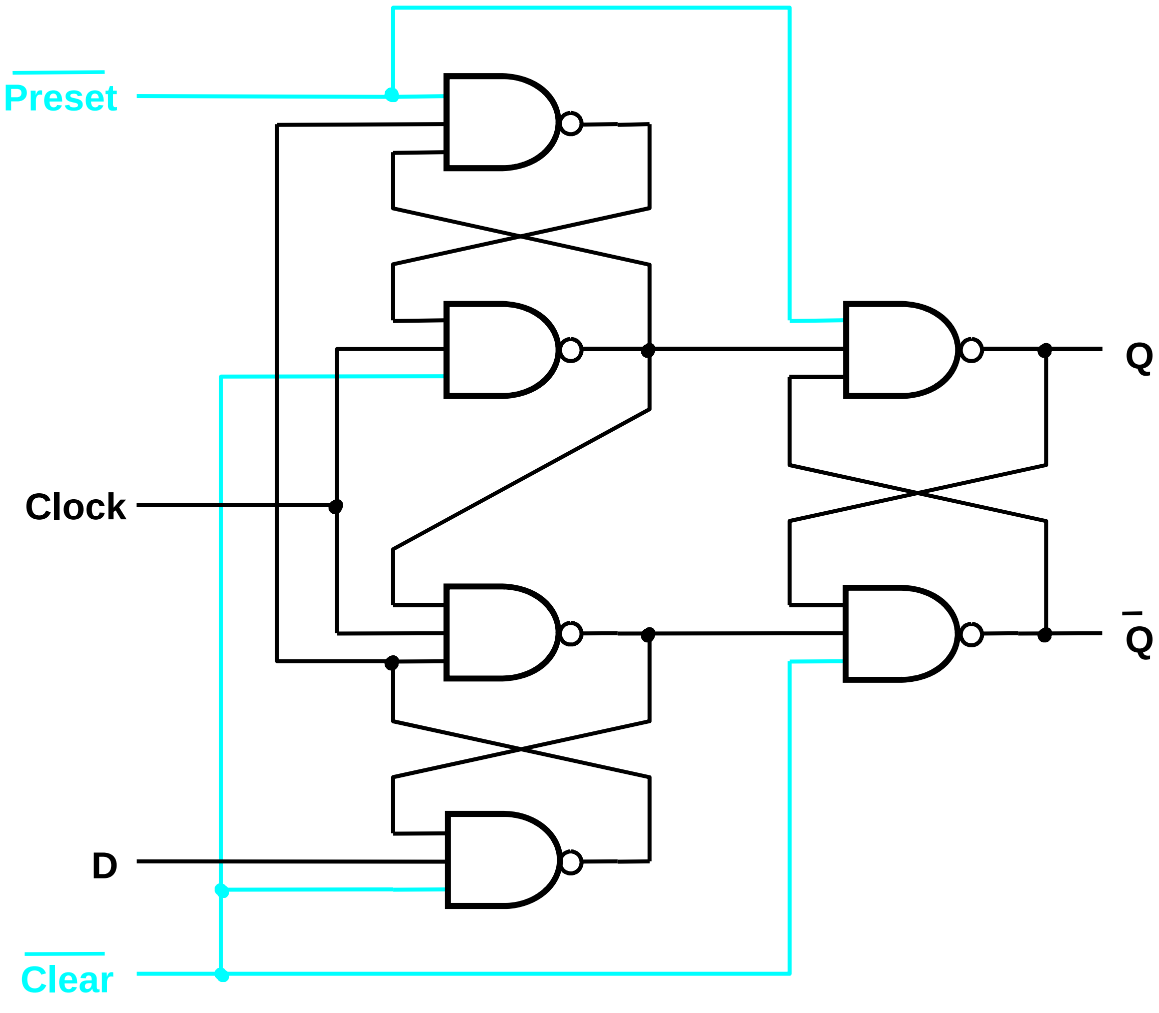


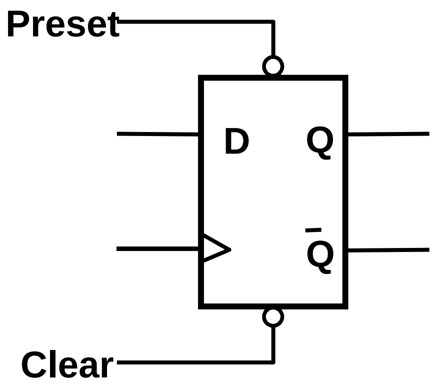
Master-Slave D Flip-flop with Clear and Preset

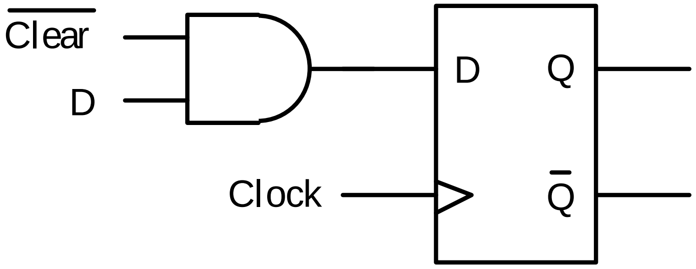




Positive-edge-trigger D Flip-flop with Clear and Preset

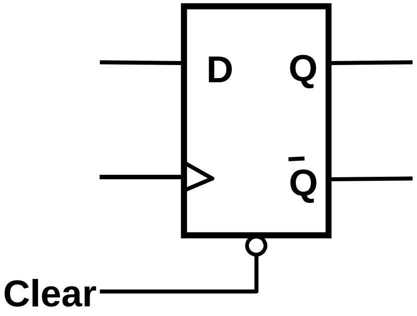


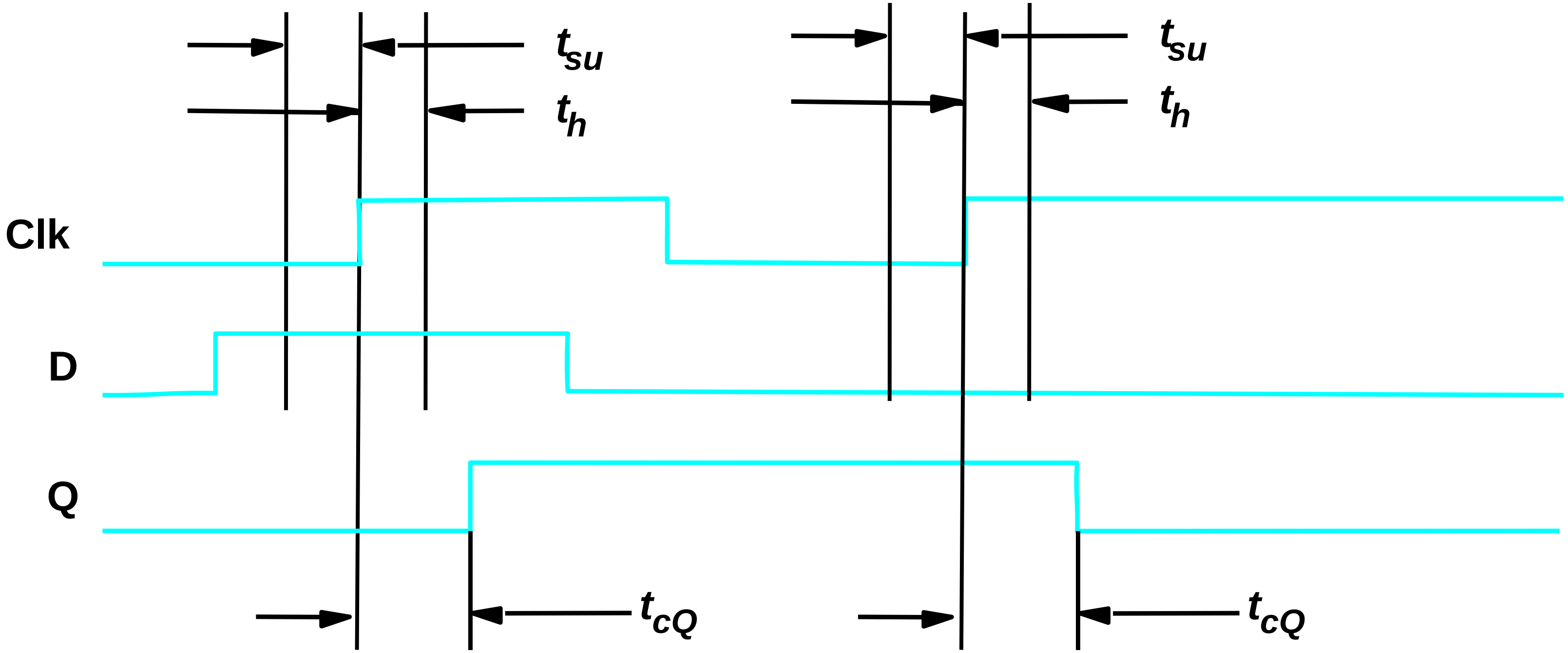




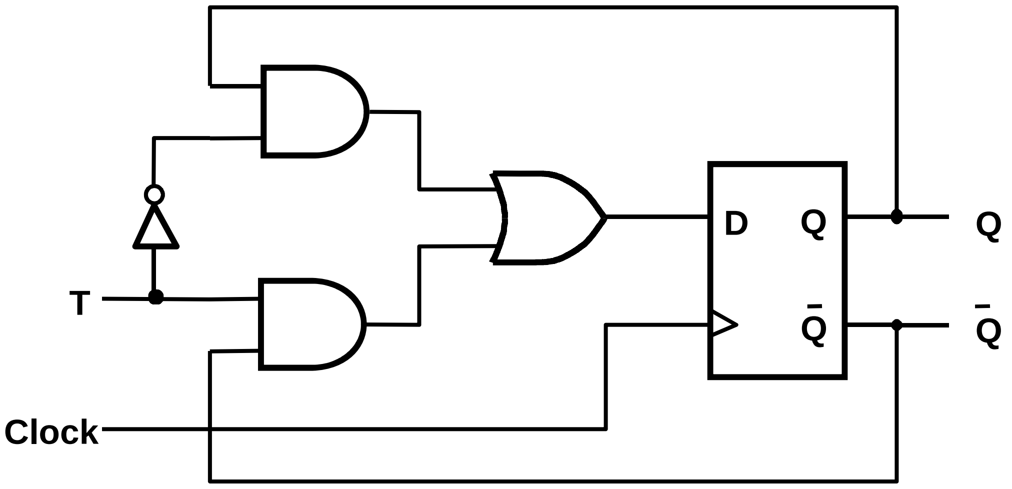
Adding a synchronous clear

Flip-Flop Timing Parameter

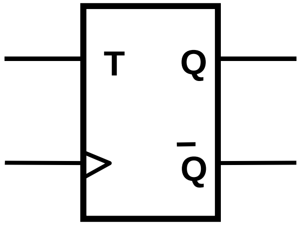


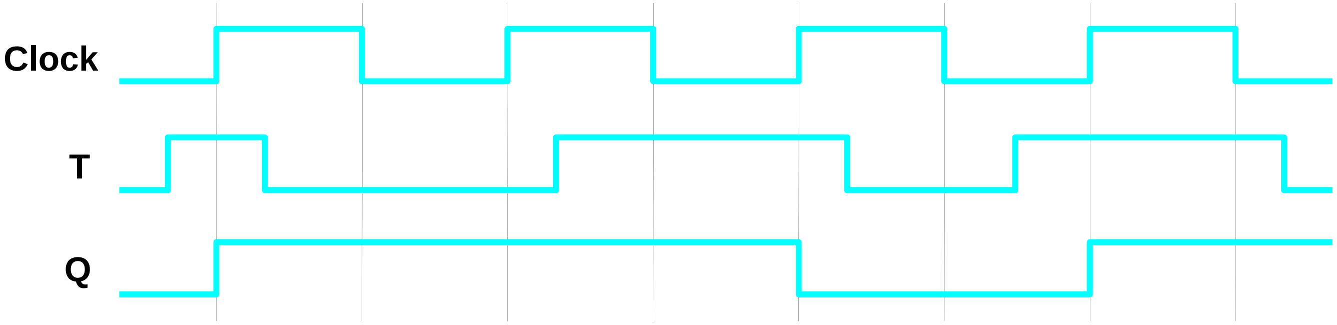


T Flip-Flop



T	Q(*t* + 1)
0	Q(*t*)
1	$\overline{Q}(t)$

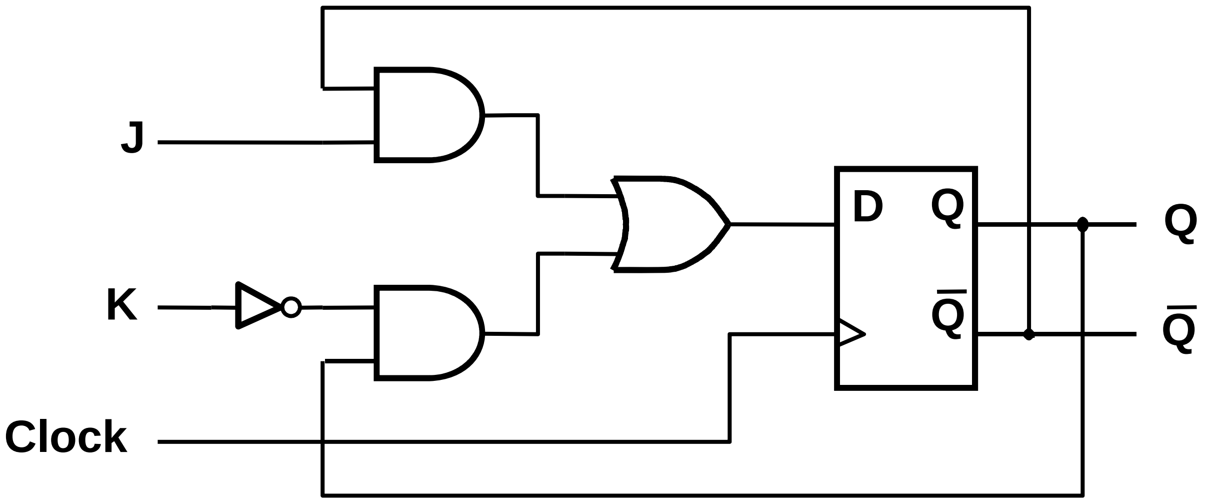




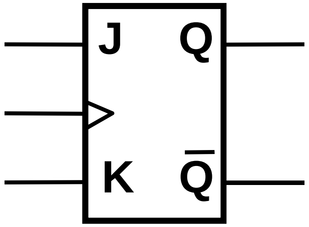
Configurable Flip-Flops

- In general purpose chips like PLDs, the flip-flops that are provided are sometimes *configurable*, which means that a flip-flop circuit can be configured to be either D, T, or some type.
- HOW?

JK Flip-Flop



J	K	Q (t+1)
0	0	Q (t)
0	1	0
1	0	1
1	1	$\overline{Q}$ (t)



Summary of Terminology

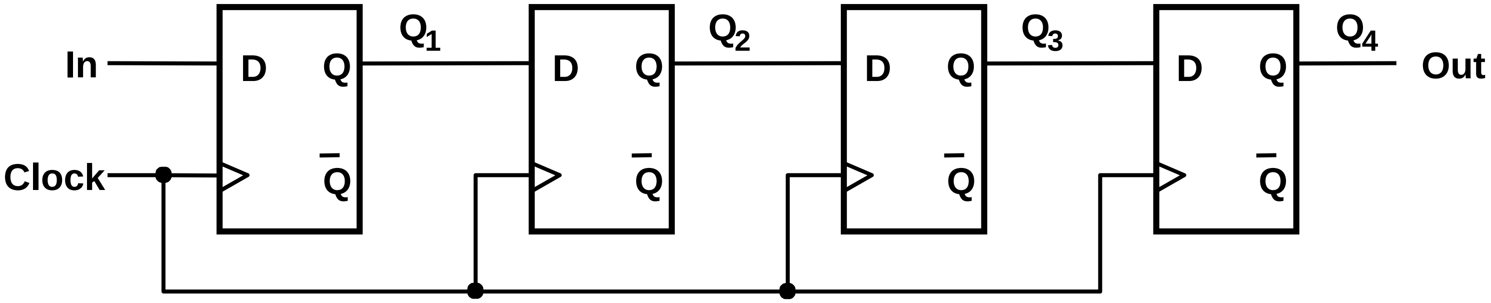
- **Basic latch** is a feedback connection of two NOR gates or two NAND gates, which can store one bit of information. It can be set to 1 using S or 0 using R.
- **Gated latch** is a basic latch that includes input gating and a control input signal.
  - **Gated SR latch** uses the S and R inputs to set the latch to 1 or reset to 0 (do have R=S=0 exception).
  - **Gated D latch** uses the D input to force the latch into a state that has the same lotic value as the D input

- A **flip-flop** is a storage element based on the bated latch principle, which can have its output sate changed only on the edge of the controlling clock signals.
  - **Edge-triggerd flip-flop** is affected only by the input values present when the active edge of the clock occurs.
  - **Master-slave flip-flop** is built with two gated latches. The master stage is active during half of the clock cycle, and the slave stage is active during the other half. The output value of the flip-flop changes on the edge of the clock that activates the transfer into the slave stage. It can be edge-triggered or level sensitive. (Gated D latch=>edge-triggered; Gates SR latch=> level-sensitive)

Registers

- A flip-flop stores one bit of information.
- When a **set** of *n* flip-flops is used to store *n* bits of information, such as an *n*-bit number, we refer to these flip-flops as a *register*.
- A common clock is used for each flip-flop in a register.

Shift Register

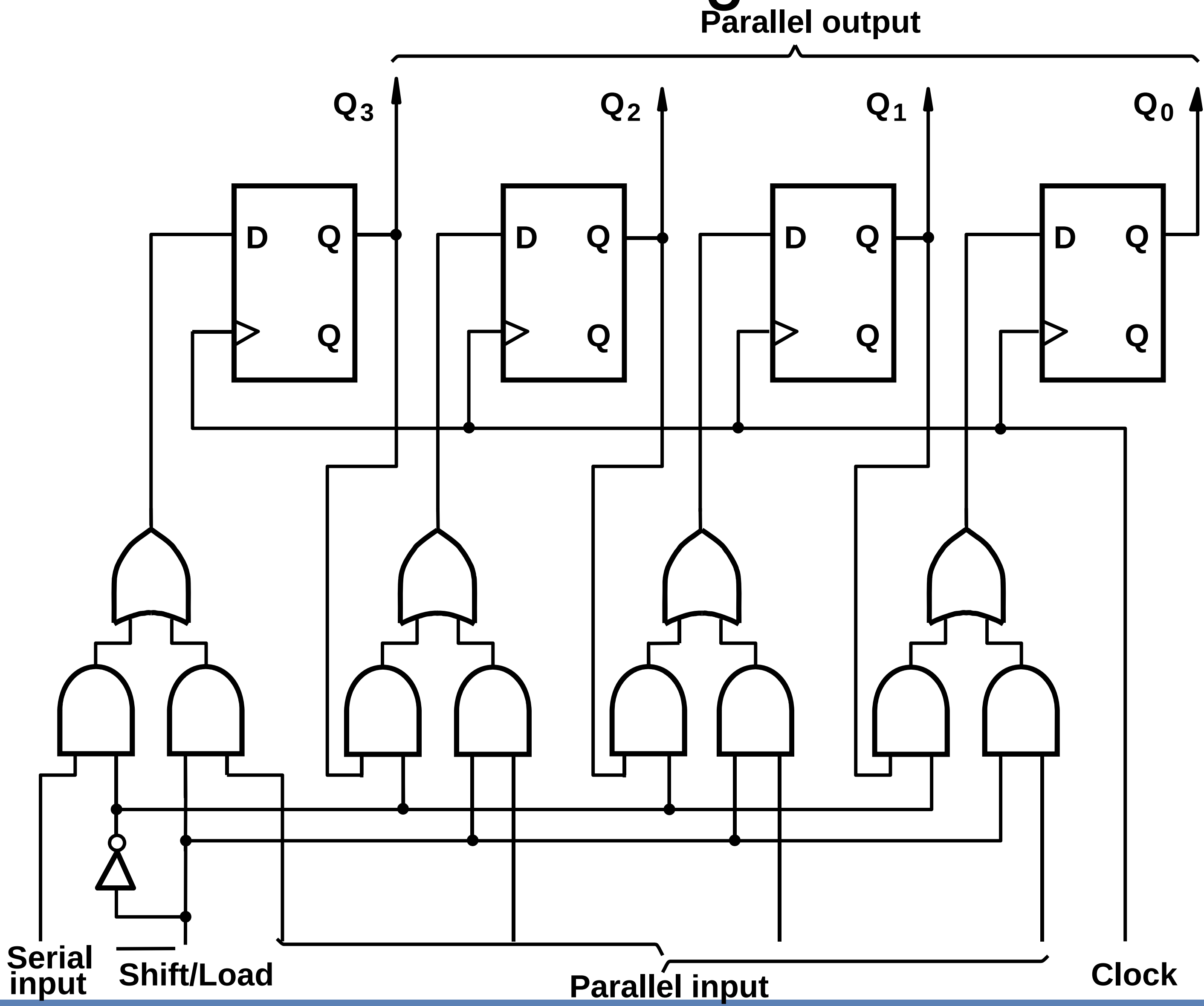


	In	Q_1	Q_2	Q_3	Q_4 = Out
t_0	1	0	0	0	0
t_1	0	1	0	0	0
t_2	1	0	1	0	0
t_3	1	1	0	1	0
t_4	1	1	1	0	1
t_5	0	1	1	1	0
t_6	0	0	1	1	1
t_7	0	0	0	1	1

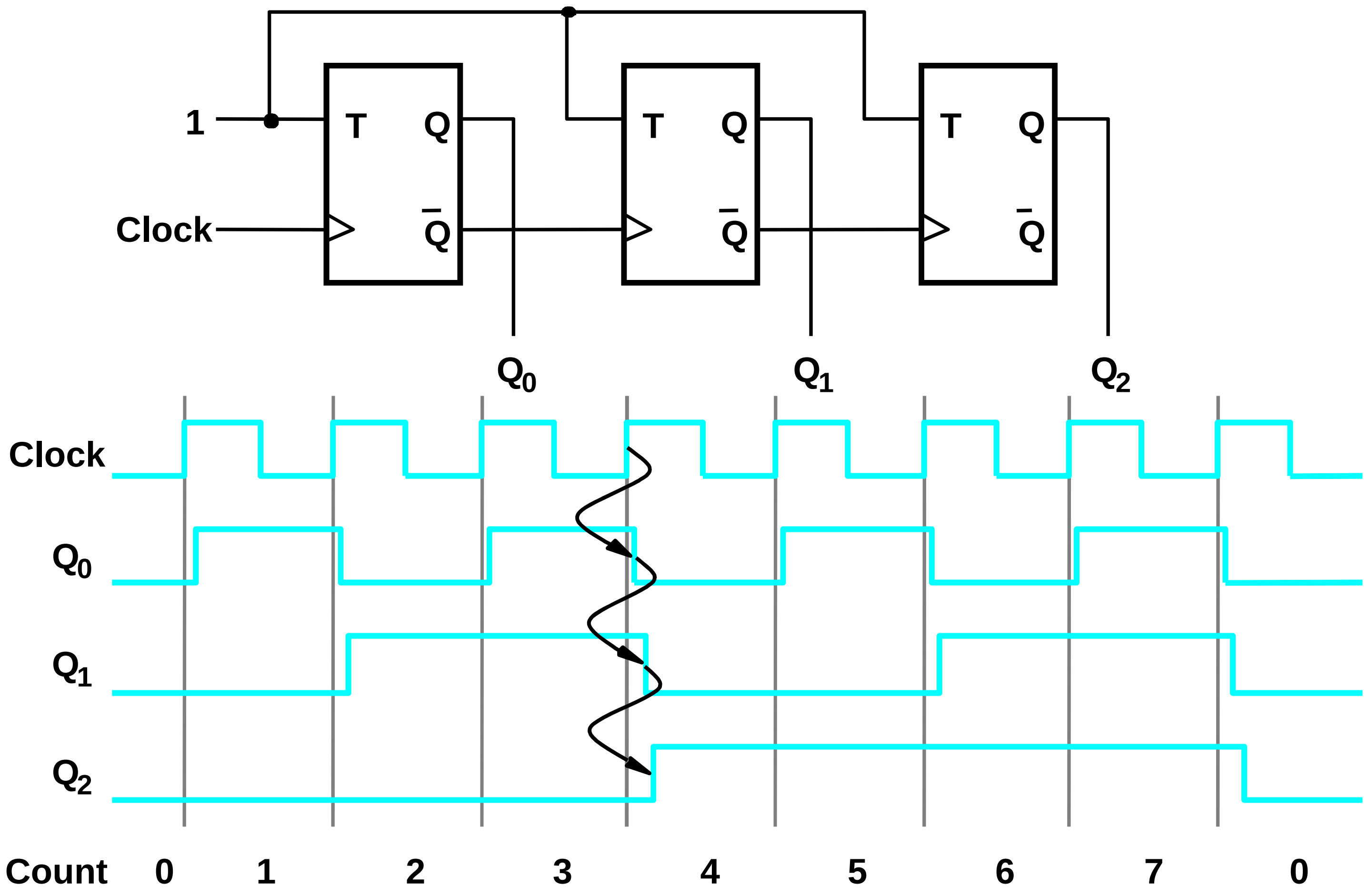
Parallel versus Serial

- Transmitting n bits at once using n separate wires. We say this scheme is **parallel transfer**.
- Transmitting n bits at once using a single wire, by performing the transfer one bit at a time, in n consecutive clock cycles. We say this scheme is **serial transfer**.

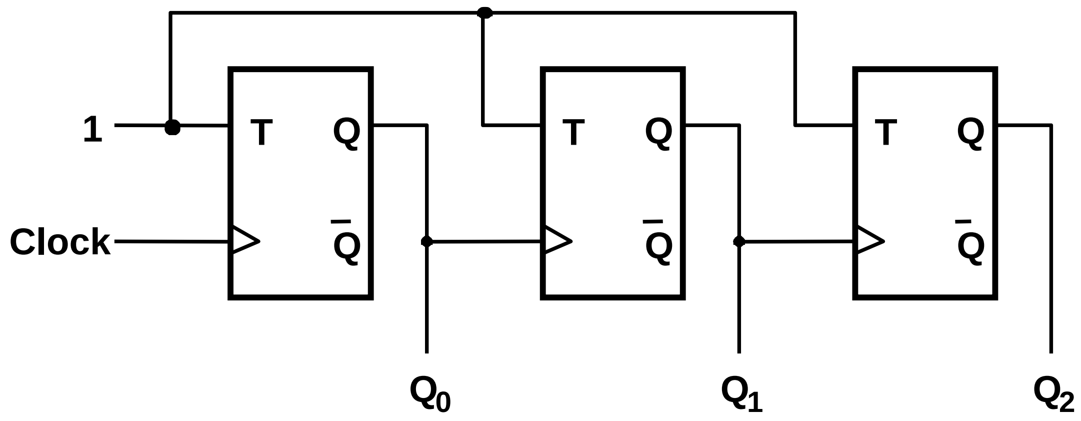
Parallel-Access Shift Register

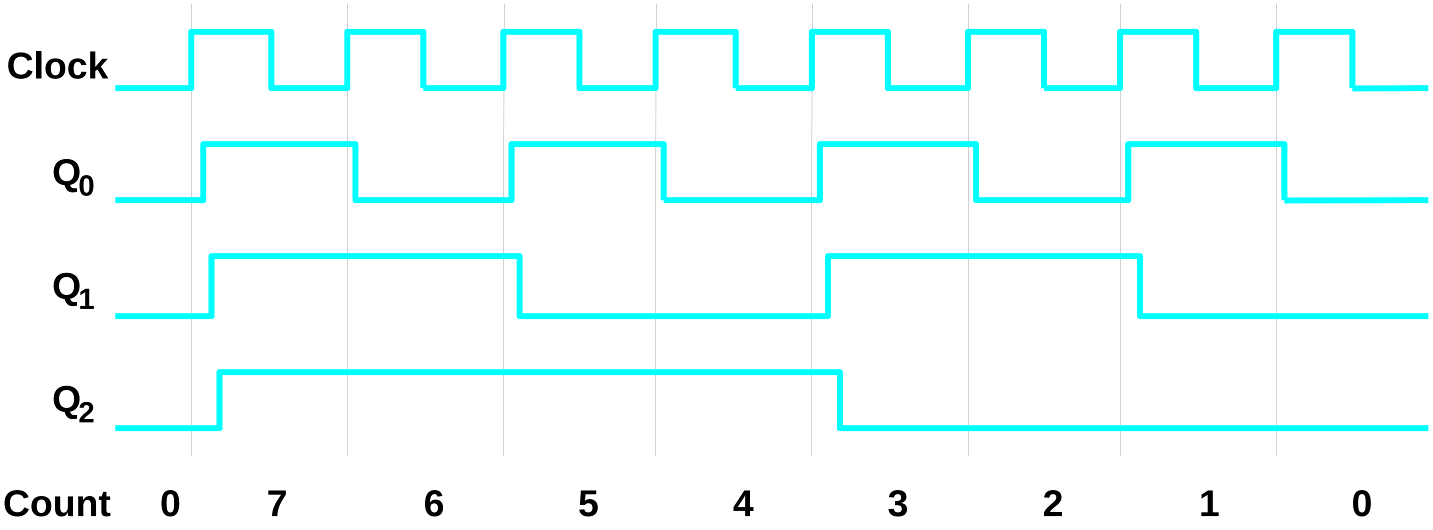


Asynchronous Up-Counter with T Flip-flops



Asynchronous Down-Counter with T Flip-flops

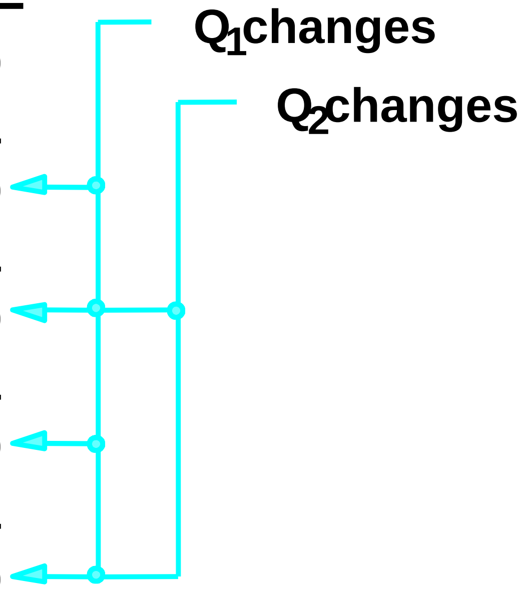




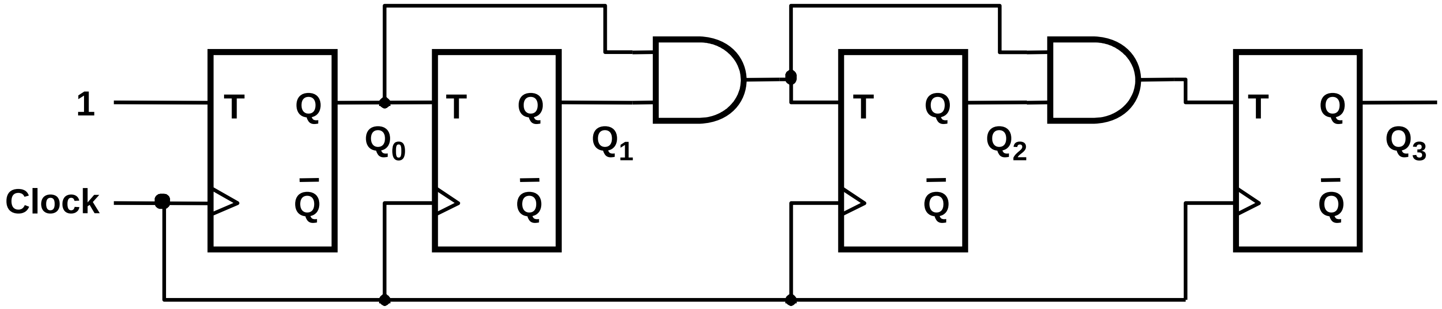
Synchronous Counter with T Flip-flops

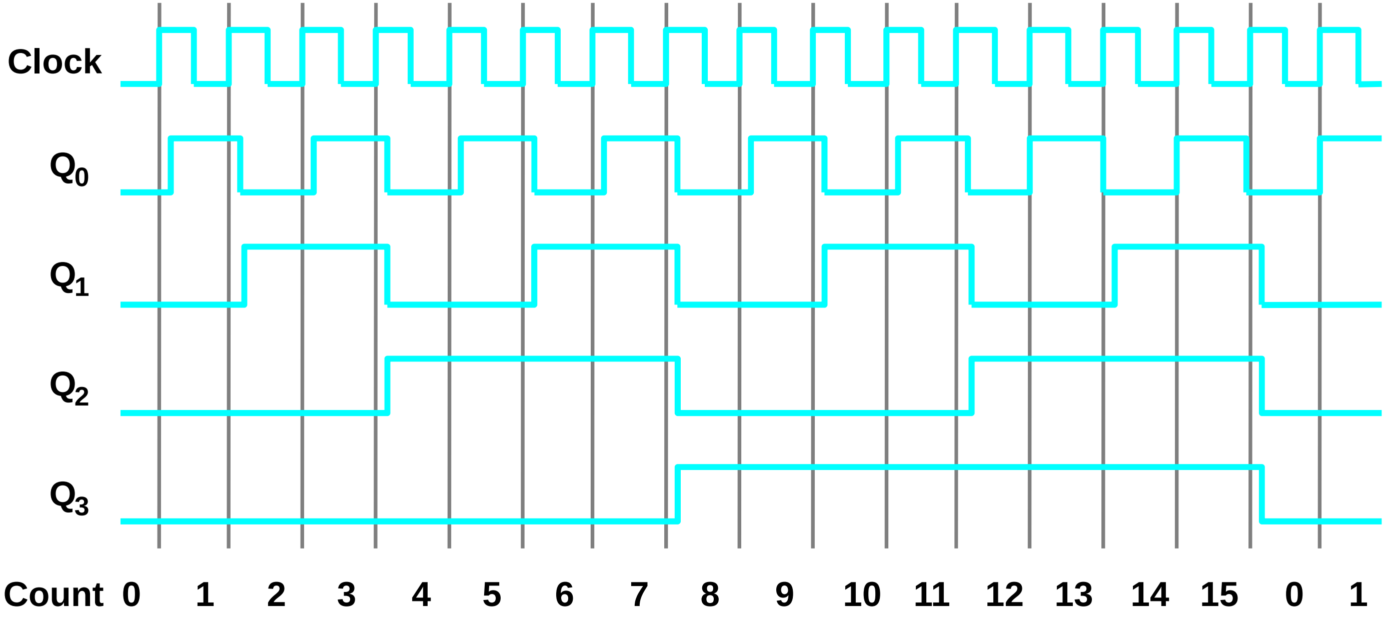
- $T_0=1$
- $T_1=Q_0$
- $T_2=Q_0Q_1$
- $T_3=Q_0Q_1Q_2$
- $T_n=Q_0Q_1\dots Q_{n-1}$

Clock cycle	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

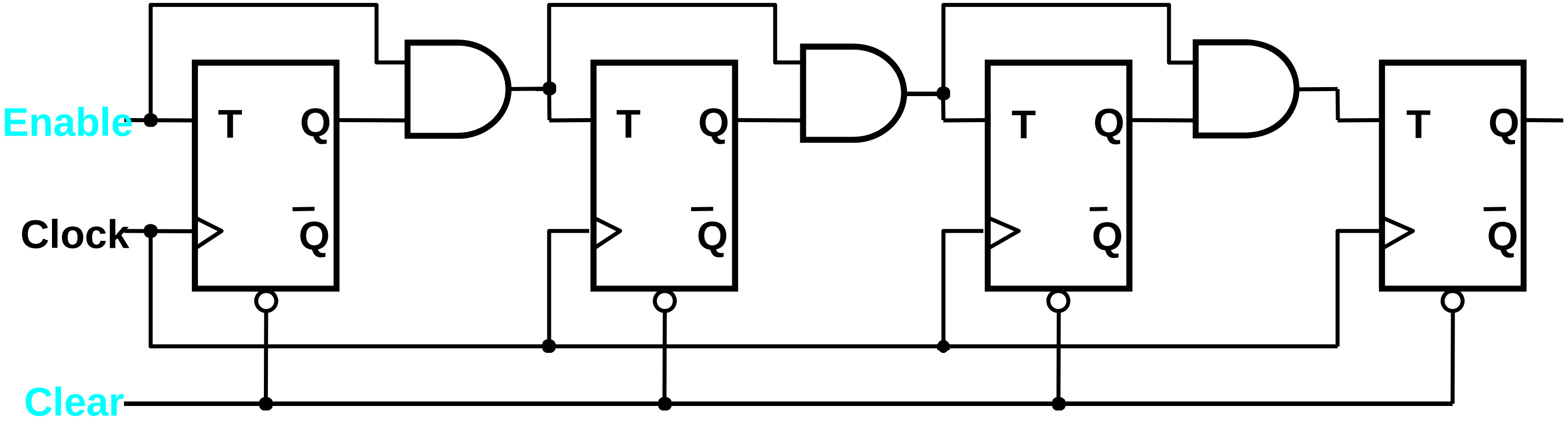


Synchronous Counter with T Flip-flops





Enable and Clear

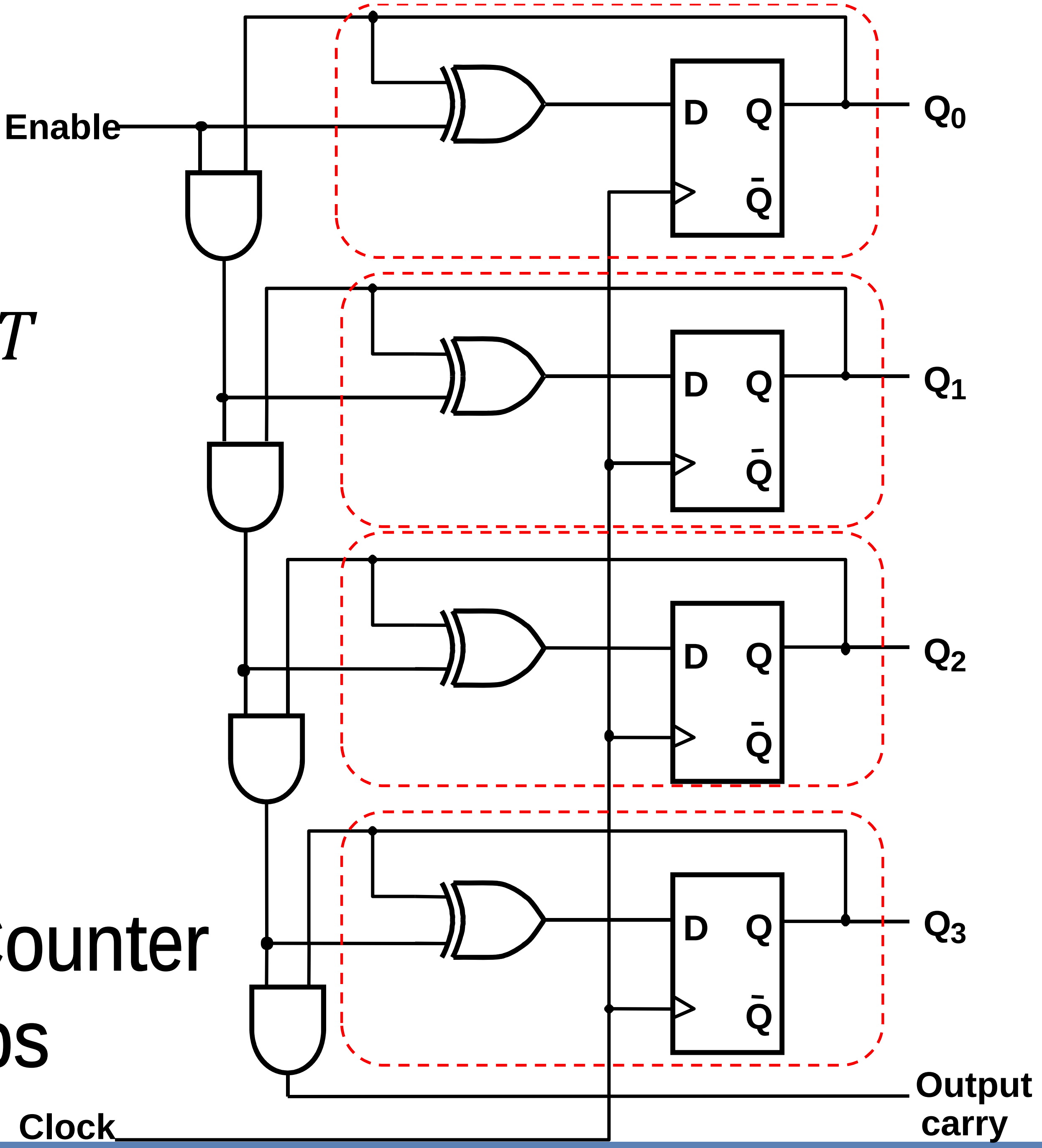


Synchronous Counter with D Flip-flops

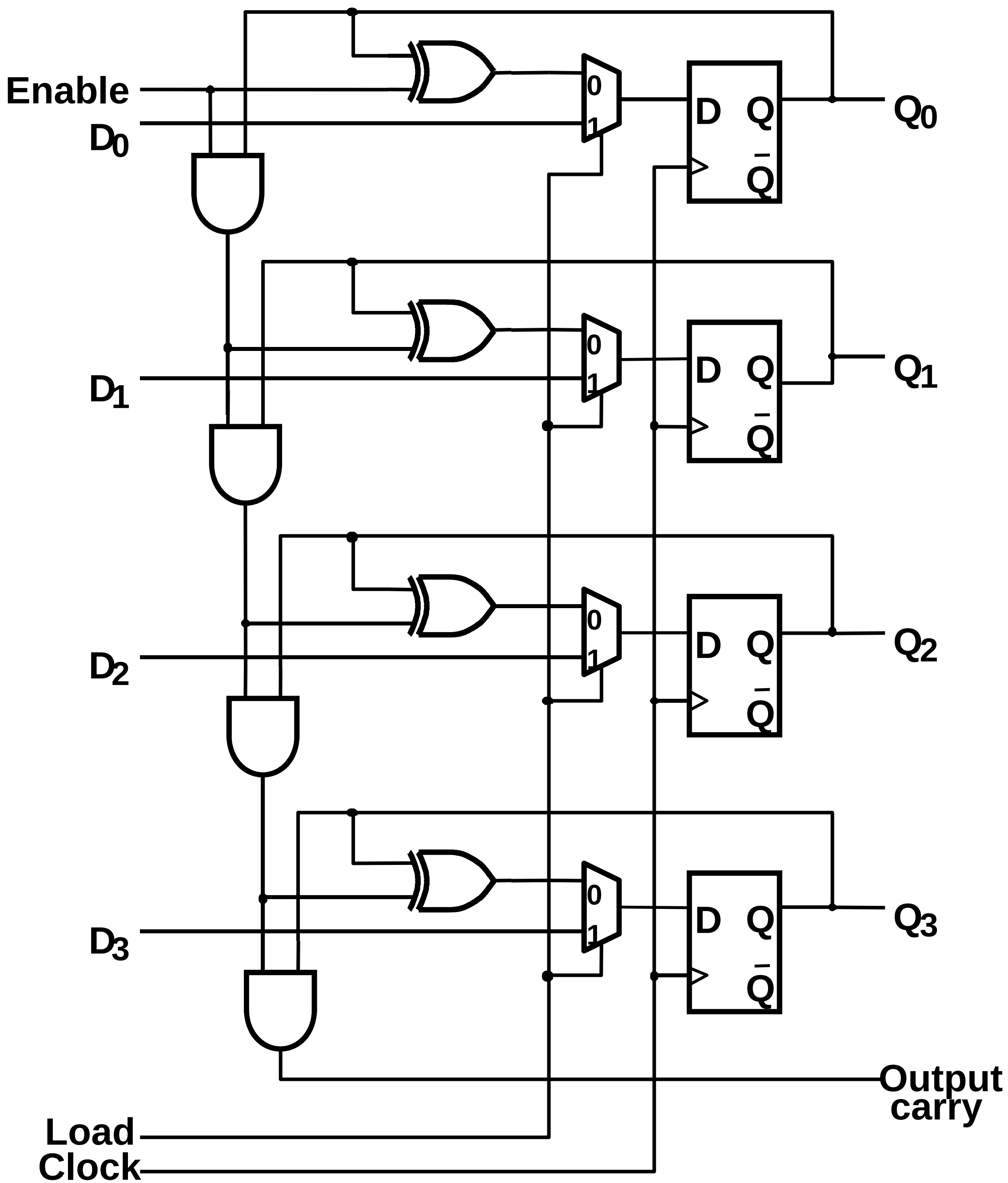
- $D_0 = \overline{Q_0} = 1 \oplus Q_0 (= Q_0 \oplus Enable)$
- $D_1 = Q_1 \oplus Q_0 (\cdot Enable)$
- $D_2 = Q_2 \oplus Q_1 Q_0 (\cdot Enable)$
- $D_3 = Q_3 \oplus Q_2 Q_1 Q_0 (\cdot Enable)$
- $D_i = Q_i \oplus Q_{i-1} Q_{i-2} \ldots . Q_1 Q_0 (\cdot Enable)$

- $D = Q\bar{T} + \bar{Q}T$

  $= Q \oplus \mathrm{T}$



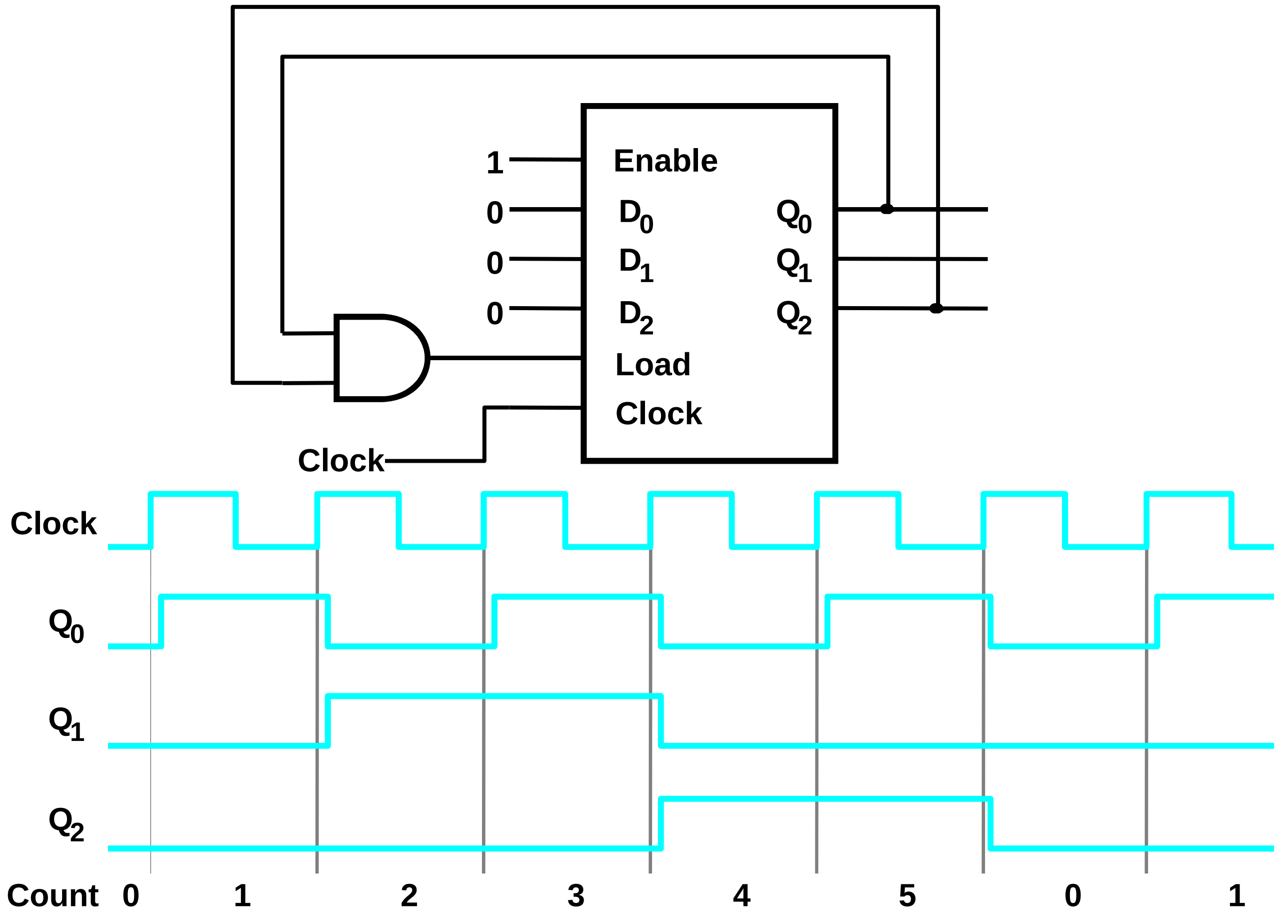
Synchronous Counter with D Flip-flops



Counter with Parallel-load

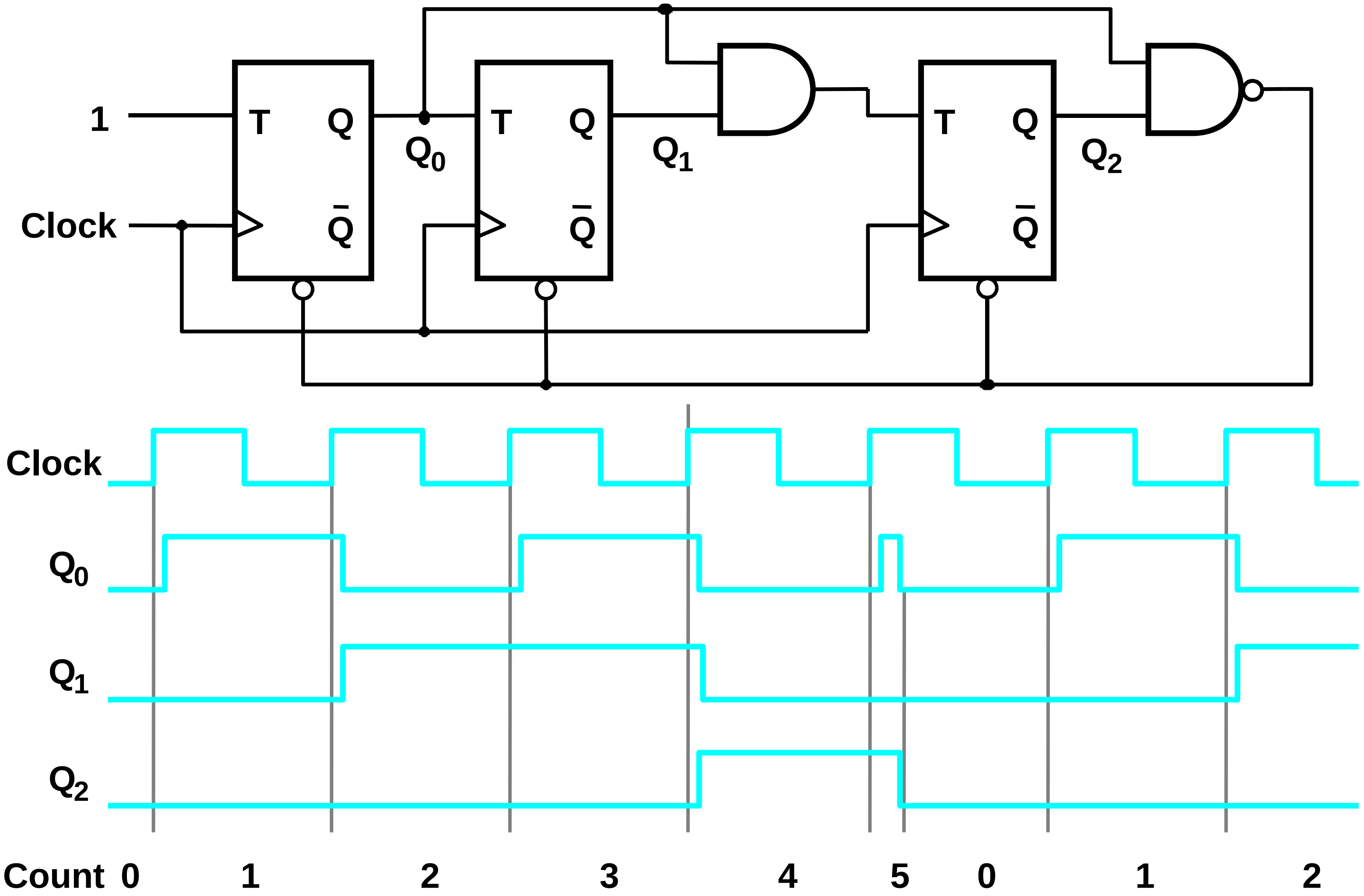
A Modulo-6 Counter with Synchronous Reset

- Counting sequence: 0,1,2,3,4,5,0,1, and so on.

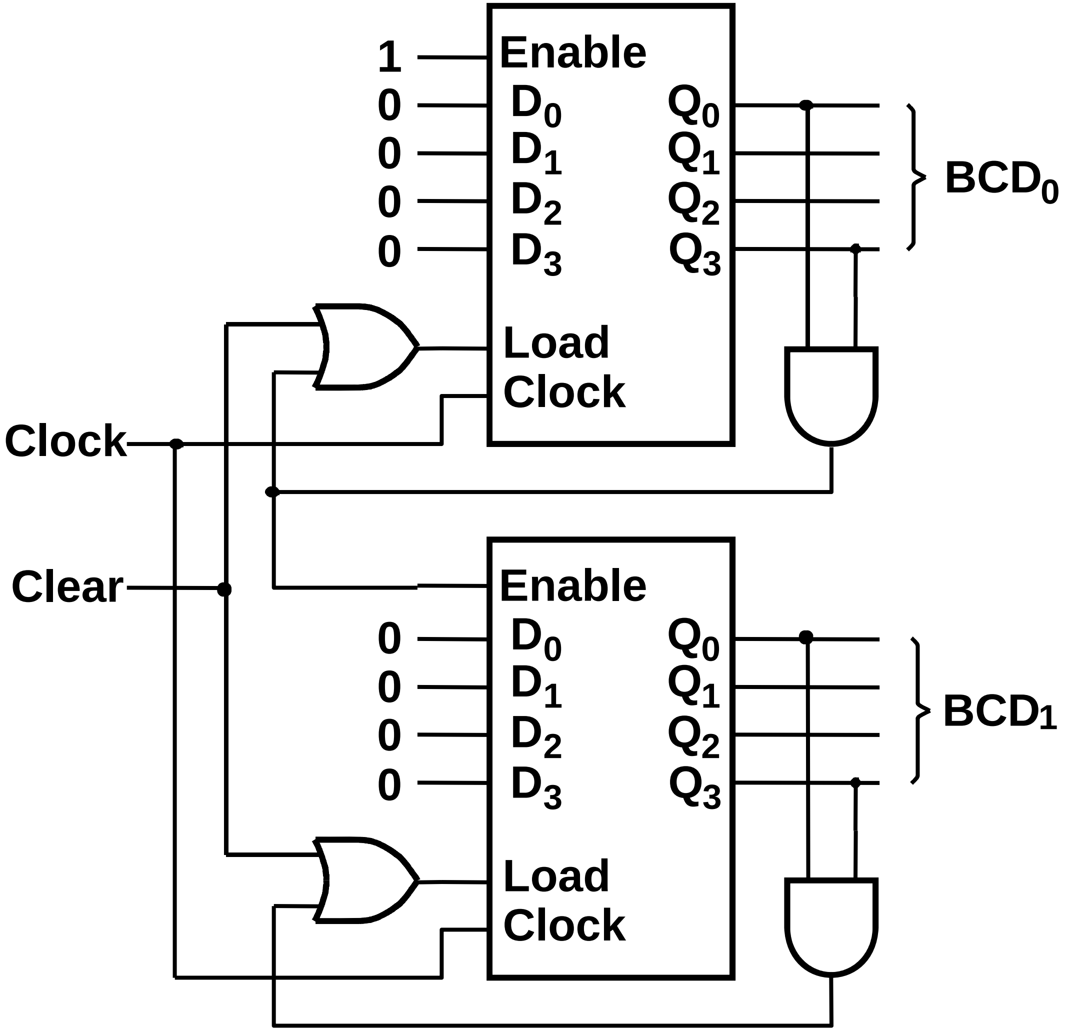


A Modulo-6 Counter with Asynchronous Reset

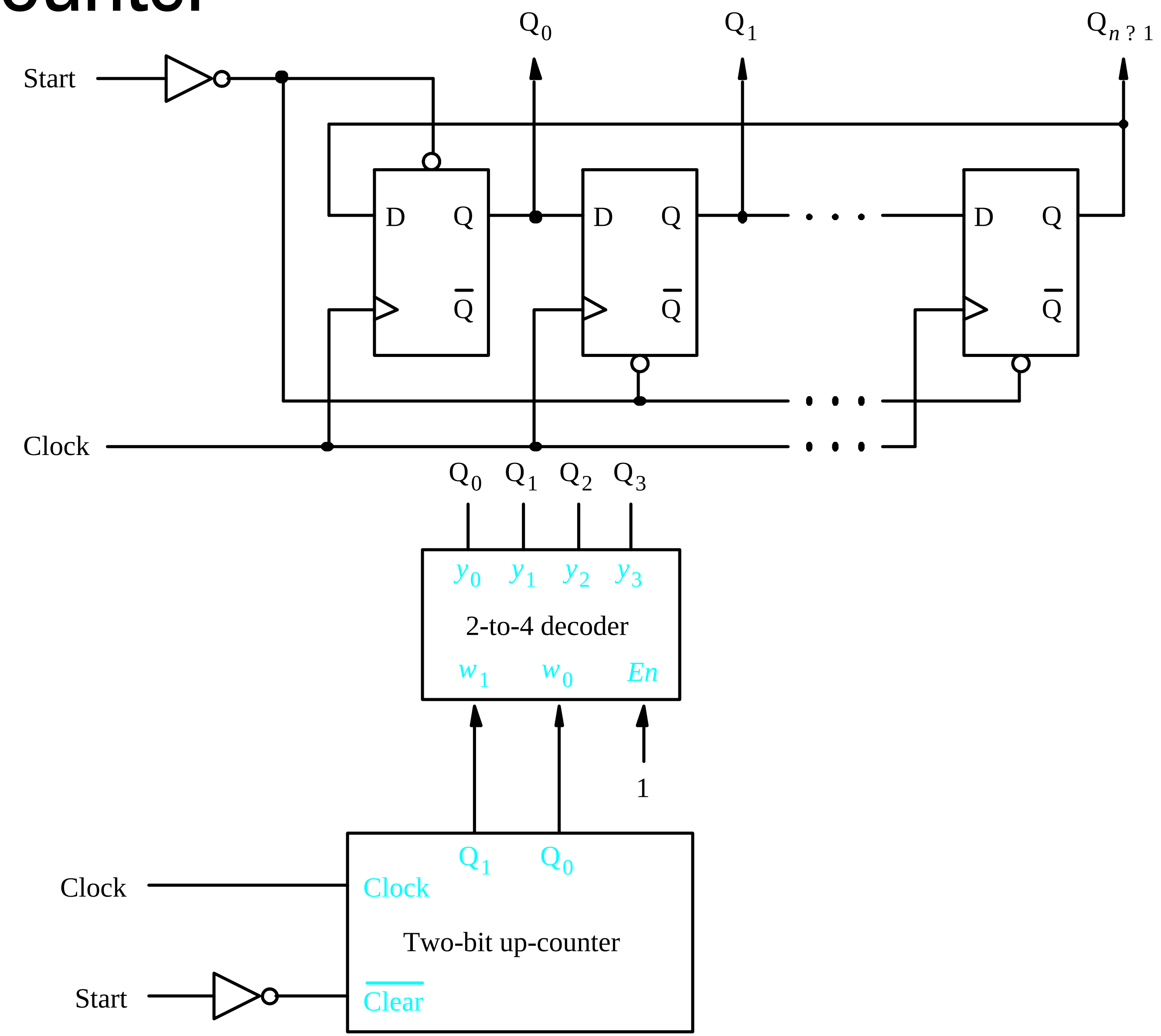
- Modulo 5 or 6?



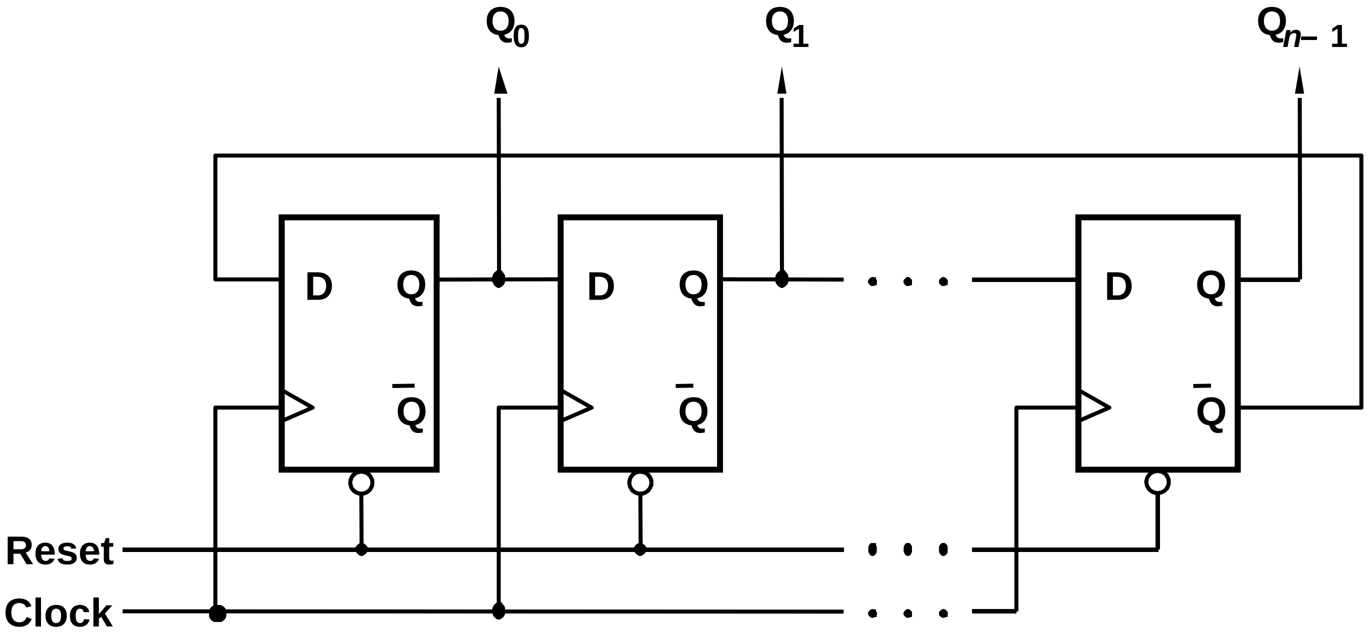
BCD Counter


1
0
0
0
0
Enable
D_0
D_1
D_2
D_3
Q_0
Q_1
Q_2
Q_3
BCD_0
Load
Clock
Clock
Clear
Enable
0
0
0
0
D_0
D_1
D_2
D_3
Q_0
Q_1
Q_2
Q_3
BCD_1
Load
Clock

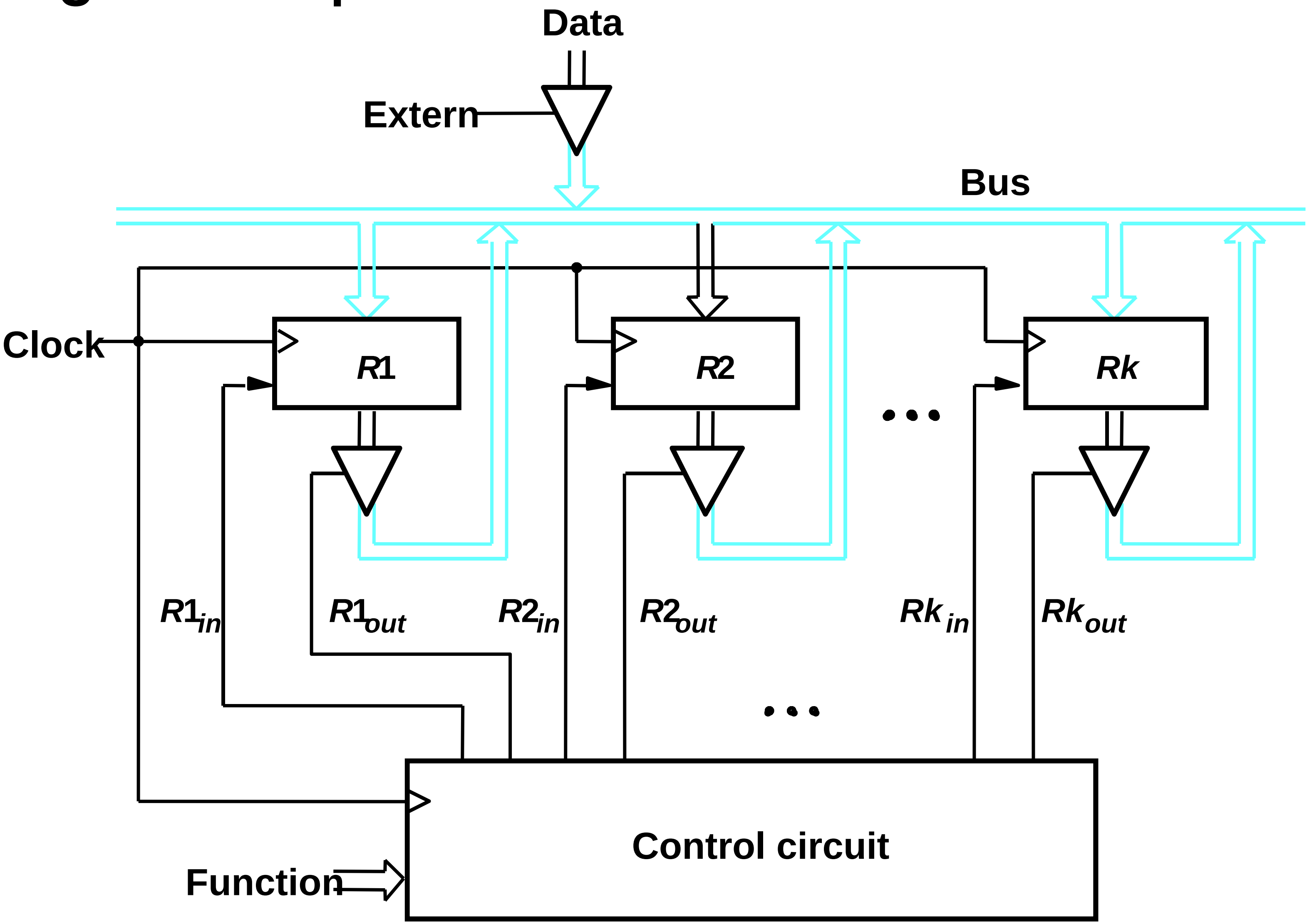
Ring Counter



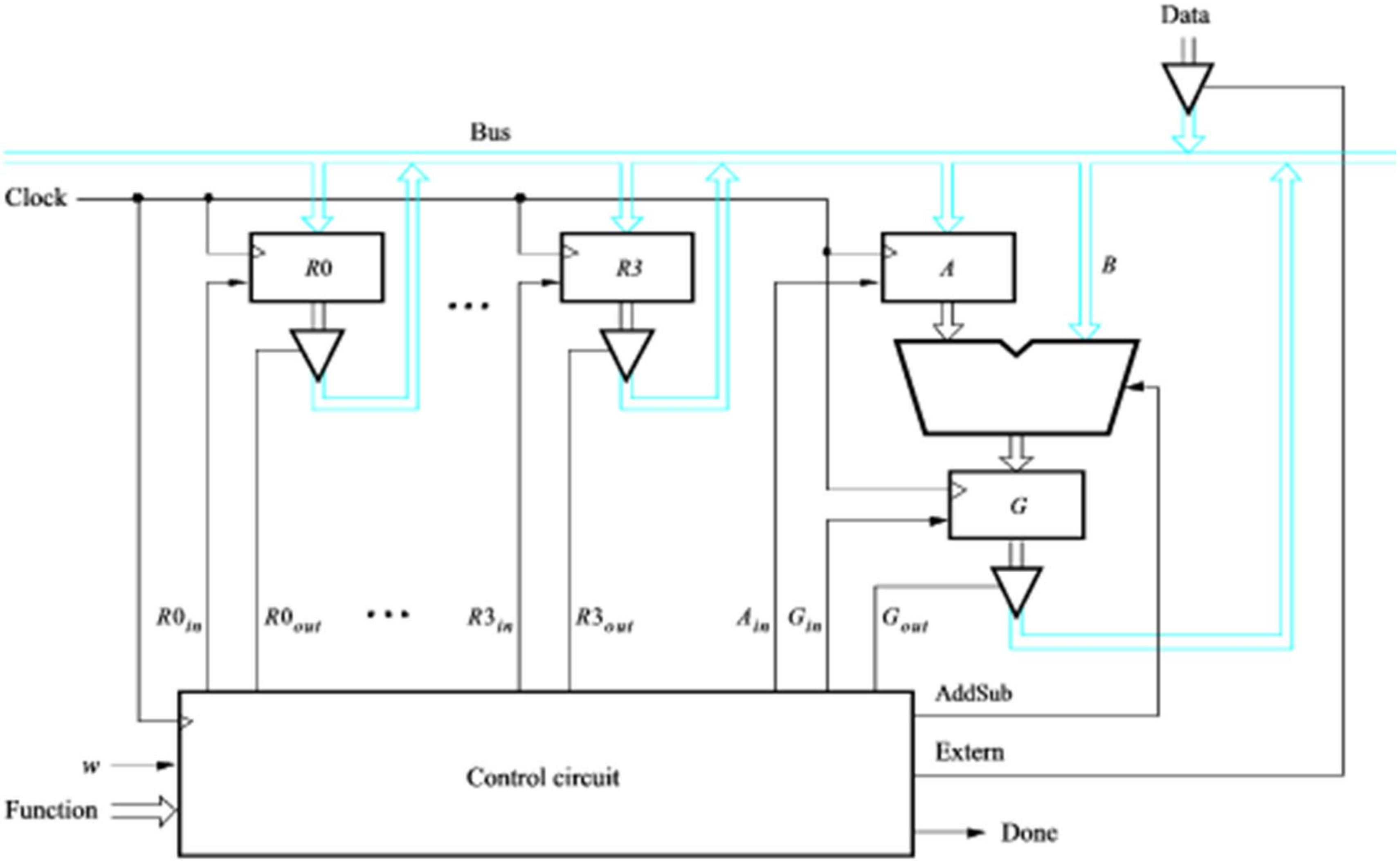
Johnson Counter



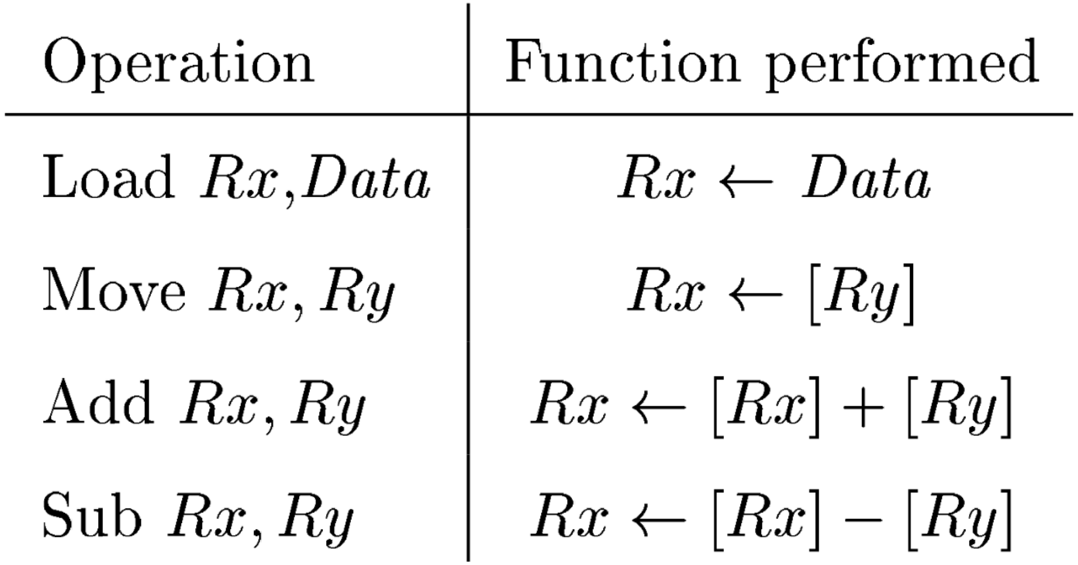
Design Example: Bus Structure



Simple Processor

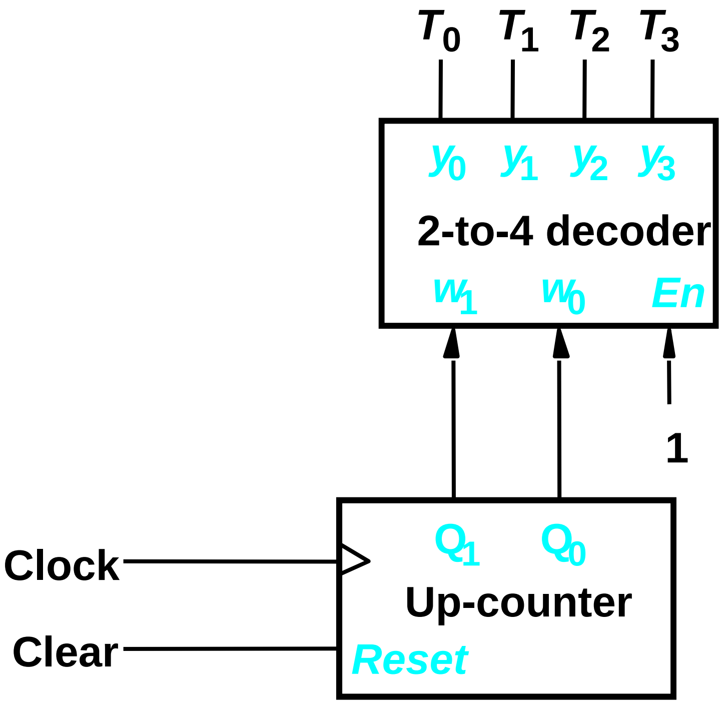


Operation Performed in the Processor



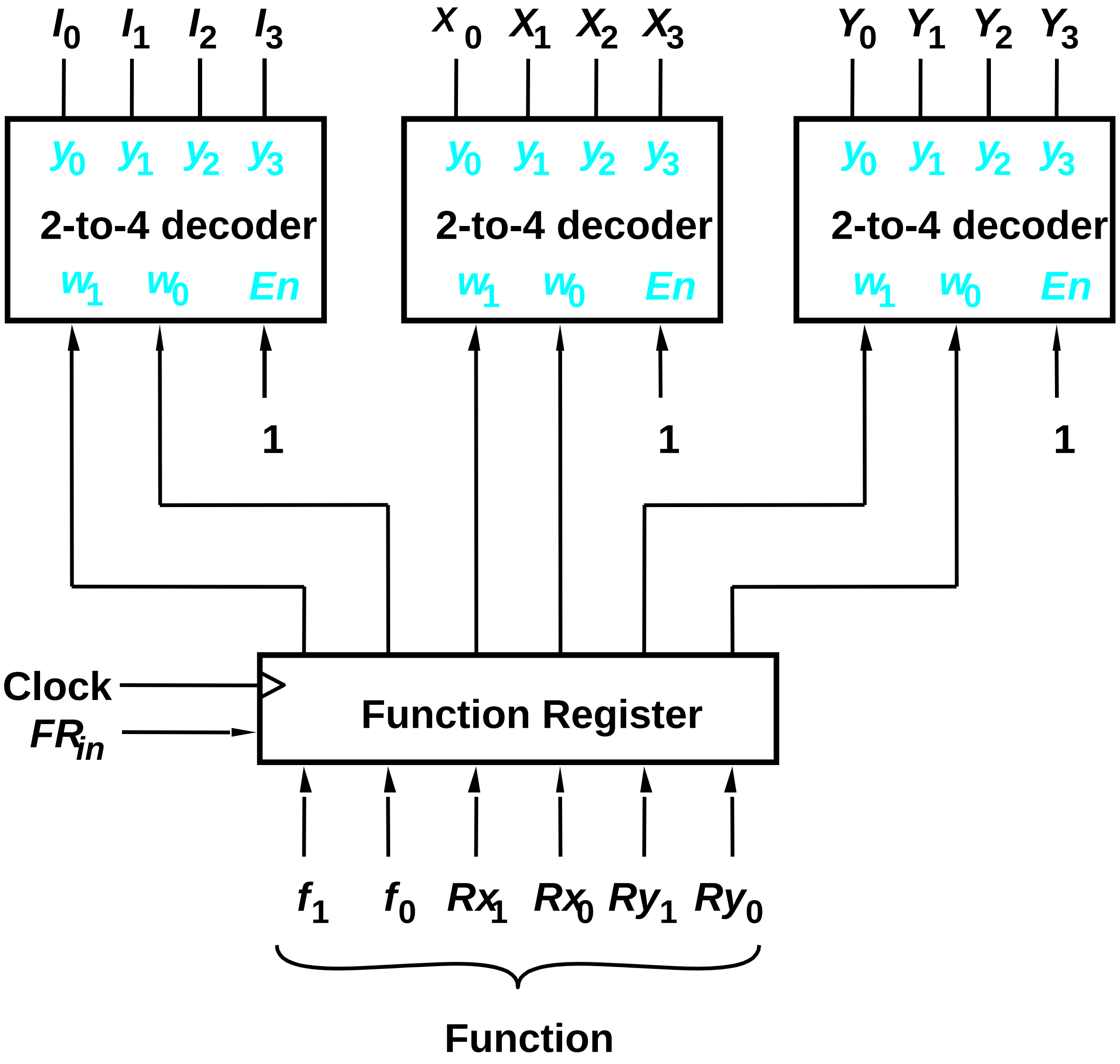
Operation	Function performed
Load Rx,$Data$	$Rx \leftarrow Data$
Move Rx, Ry	$Rx \leftarrow [Ry]$
Add Rx, Ry	$Rx \leftarrow [Rx] + [Ry]$
Sub Rx, Ry	$Rx \leftarrow [Rx] - [Ry]$

Control Circuit

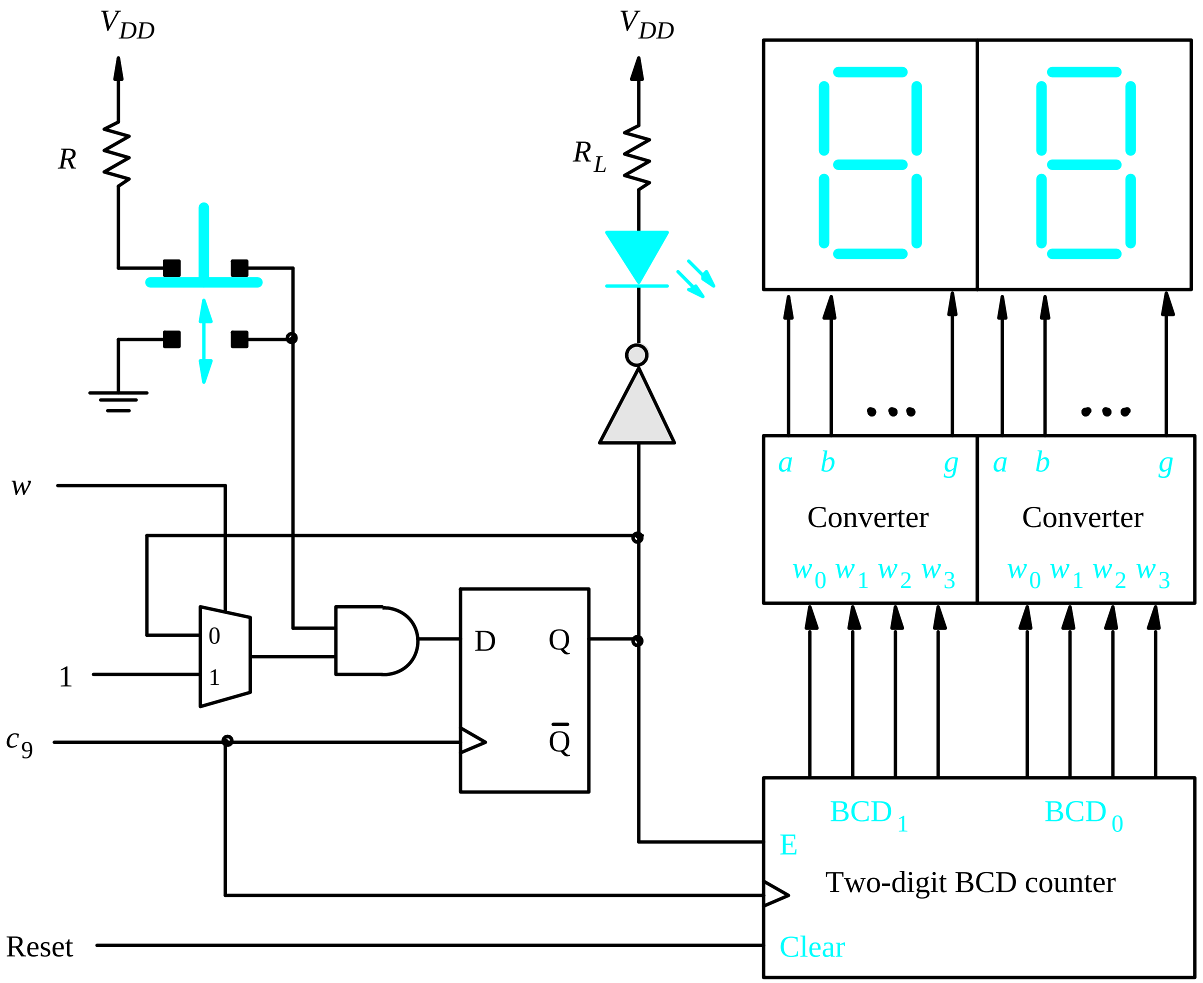


	T_1	T_2	T_3
(Load): I_0	$Extern$, $R_{in} = X$, $Done$		
(Move): I_1	$R_{in} = X$, $R_{out} = Y$, $Done$		
(Add): I_2	$R_{out} = X$, A_{in}	$R_{out} = Y$, G_{in}, $AddSub = 0$	G_{out}, $R_{in} = X$, $Done$
(Sub): I_3	$R_{out} = X$, A_{in}	$R_{out} = Y$, G_{in}, $AddSub = 1$	G_{out}, $R_{in} = X$, $Done$

Control Circuit

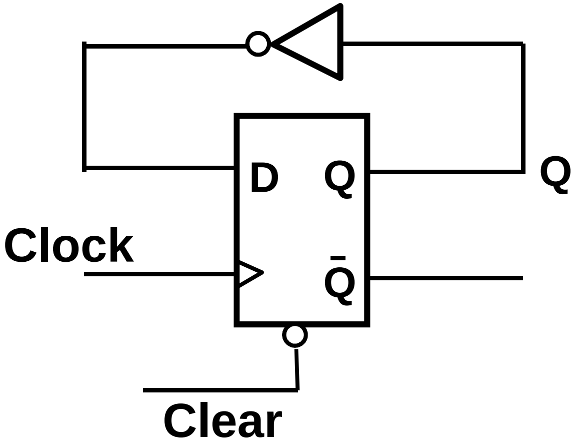


Reaction Timer



Timing Analysis

- $T_{min} = t_{cQ} + t_{NOT} + t_{su}$
- $F_{max} = \dfrac{1}{T_{min}}$
- A simple flip-flop circuit
  - $T_{min} = 1.0 + 1.1 + 0.6 = 2.7ns$
  - $F_{max} = \dfrac{1}{2.7ns} = 370.37\text{MHz}$



Timing Analysis for a 4-bit Counter

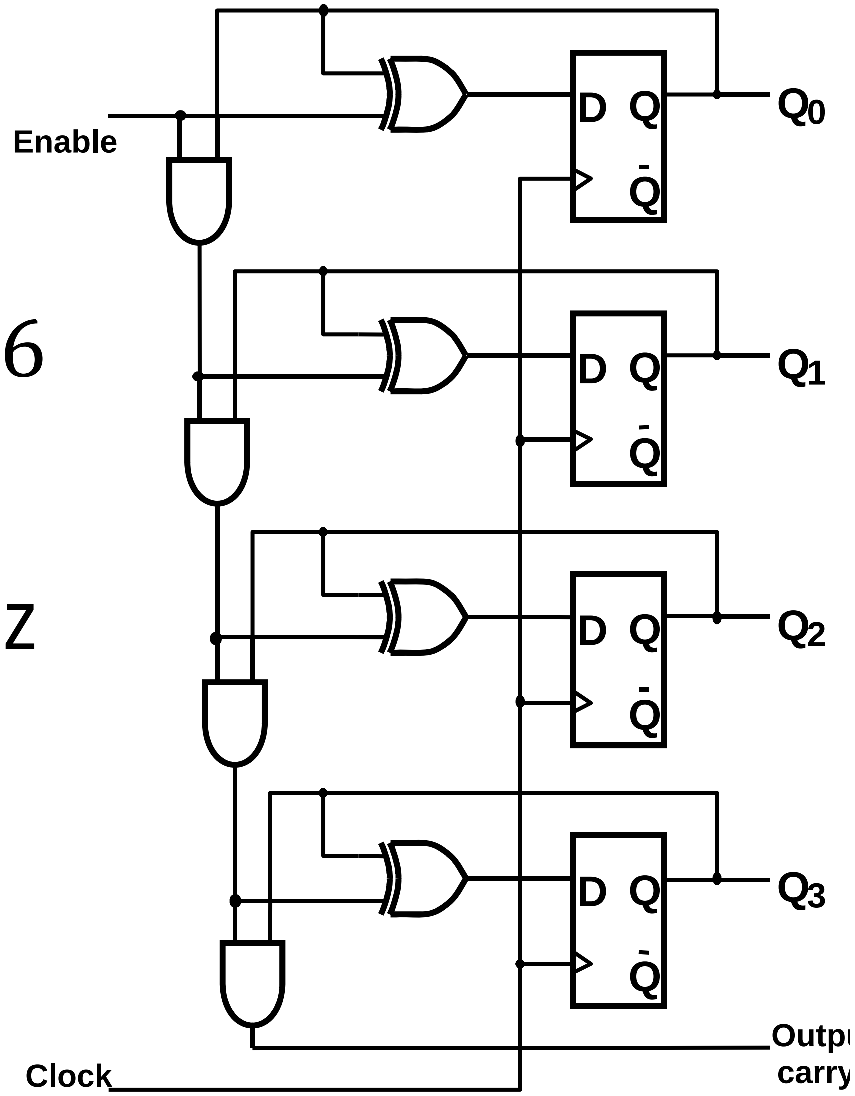
- $T_{min} = t_{cQ} + 3(t_{AND})$

  $+\mathrm{t}_{\mathrm{XOR}} + t_{su}$

  $= 1.0 + 3(1.2) + 1.2 + 0.6$

  $= 6.4ns$

- $F_{max} = \frac{1}{6.4ns} = 156.25\ \mathrm{MHz}$


Enable
D
Q
$\bar{Q}$
Q_0
Q_1
Q_2
Q_3
Outpı
carry
Clock