

Física Experimental - LEI

2023-2024 - 1.º Semestre

Circuitos Digitais

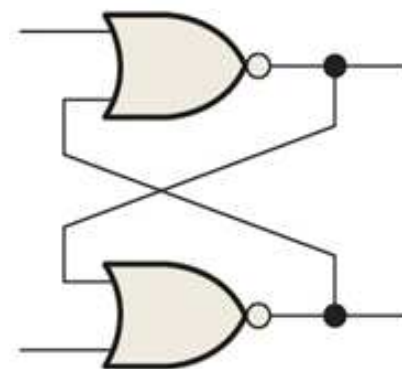
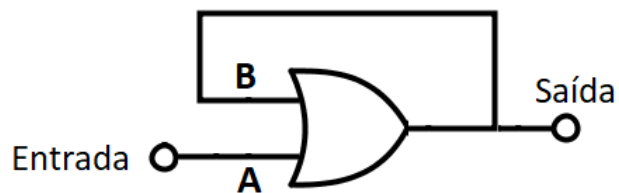
Circuitos com memória e contadores

Bibliografia:

- 1) *Digital Fundamentals* - Global Edition, Floyd, 11th Edition, Pearson Education, 2015.
- 2) *Sistemas Digitais: Fundamentos e Aplicações*, 9ª edição, Floyd, Bookman, Brasil, 2007.
- 3) *Digital Electronic: A Practical Approach with VHDL*, W. Kleitz, 9th Edition, Pearson Education, 2012.
- 4) *Logic and Computer Design Fundamentals*, Morris, Kime, 4th Edition, Pearson Education Limited.

.....

Introdução à Lógica Sequencial

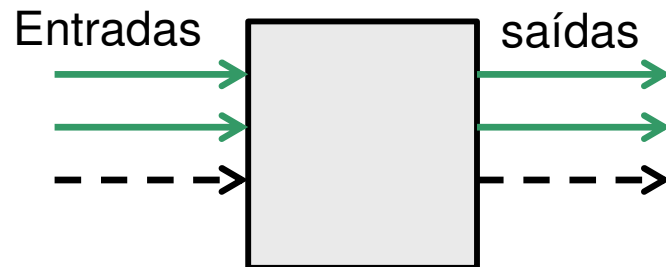


Introdução à lógica sequencial

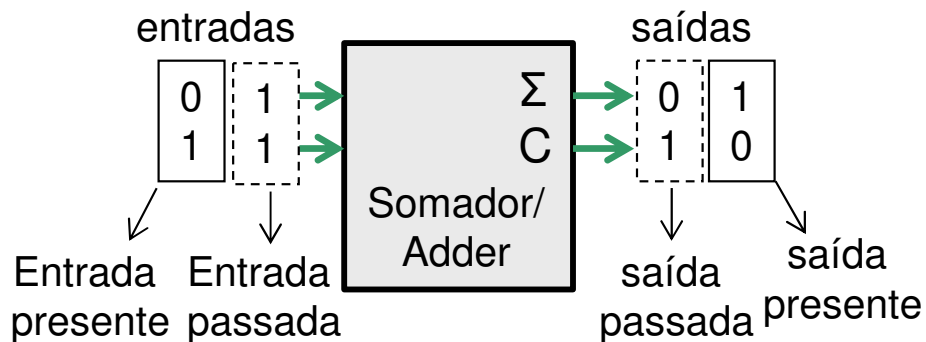
- Quase todos circuitos digitais práticos são uma mistura de circuitos **lógicos combinacionais** e **circuitos lógicos sequenciais**.
- Nos circuitos lógicos combinacionais a(s) saída(s) são função apenas dos valores das entradas, i.e., os valores das saídas só dependem dos valores das entradas, e obedecem às regras da lógica booleana. São descritos por **tabelas de verdade**.
- Os **circuitos sequenciais** (também designados por **maquinas de estados**) são capazes de produzir **saídas que não dependem apenas das entradas atuais**, mas também de **entradas passadas e de estado anteriores**. Por isso necessitam de guardar os valores dos referidos estados anteriores, isto é, ao contrário dos circuitos lógicos combinacionais, os circuitos lógicos sequenciais têm elementos de memória. O objetivo dos elementos de memória é guardar valores lógicos.
- Os **circuitos sequenciais** são descritos por **tabelas de estados** ou **diagramas de estado**.
- Os circuitos sequenciais são circuitos **controlados** e usados **para controlar** outros circuitos numa sequência específica, ditada por um sinal de relógio ou por sinais de controle (de **habilitar/desabilitar – circuitos habilitados**).
- Com vários circuitos biestáveis, ligados por intermédio de lógica adicional, é possível construir máquinas de estado finitas (blocos básicos na construção de todos circuitos digitais), e circuitos de memória etc., tais como contadores, divisores de frequência, entre outros.

Lógica combinacional vs lógica sequencial

Lógica combinacional (lógica combinatória)

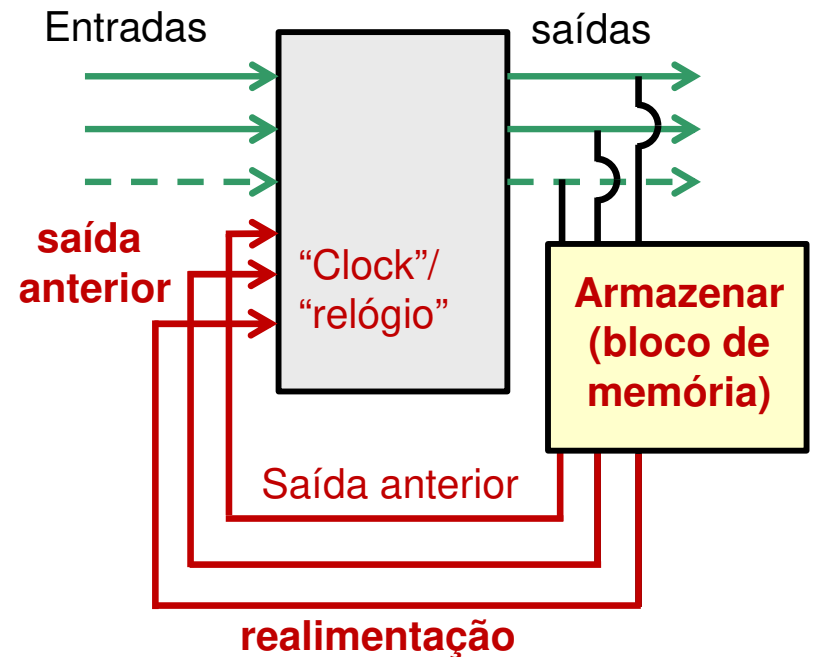


Exemplo: de um circuito combinacional



A saída atual depende apenas das entradas no instante atual.

Lógica sequencial (lógica biestável)



A saída atual será dependente das entradas atuais e das entradas/saídas passadas.

Embora as portas lógicas, por si só, não tenham capacidade de armazenar de informação (bits), pode-se criar um elemento de memória com portas lógicas aplicando o conceito de realimentação.

Latches

Latches (trincos)

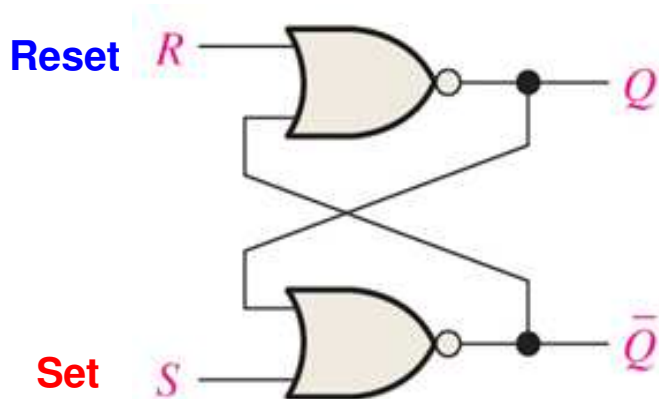
Os circuitos lógicos sequenciais mais comuns designam-se por **biestáveis**. Estes circuitos permitem armazenar 1 bit de informação, e que a informação armazenada possa ser reescrita.

Os circuitos lógicos biestáveis podem ser divididos em dois grupos:

- **Circuitos biestáveis assíncronos**, designados por **trincos** ou “**latches**”
- **Circuitos biestáveis síncronos**, também designados **básculas** ou “**flip-flops**”

O elemento de memória mais simples é o “**latch**” (trinco/guarda), e possuem duas saídas: a saída Q e a saída complemento de Q . Neste contexto, guardar significa que o circuito é capaz de **recordar** ou **lembrar** um valor.

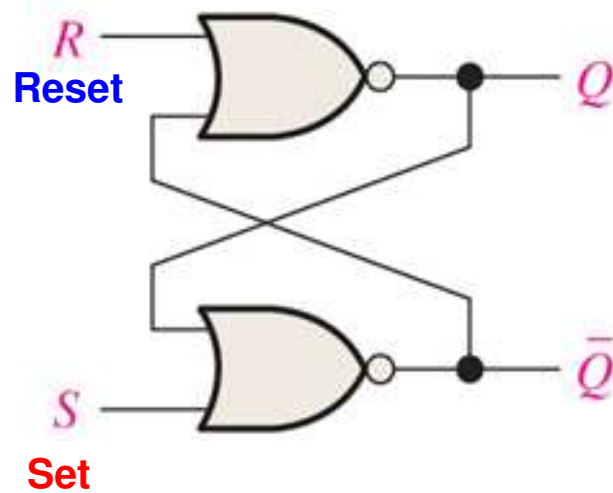
Nestes circuitos há realimentação positiva: o **acoplamento cruzado entre entradas e as saídas produz realimentação regenerativa (i.e., realimentação positiva)**. Na figura representa-se um “**latch**” implementando com portas não-OU (NOR), com duas entradas, R e S , correspondentes às funcionalidades *Reset* e *Set*.



Estes circuitos não necessitam de sinal de relógio para funcionarem; são sensíveis ao valor do sinal aplicado às entradas, isto é, o estado (valor da saída Q) pode mudar a qualquer momento, dependendo das alterações dos valores das entradas e do valor presente da Q .

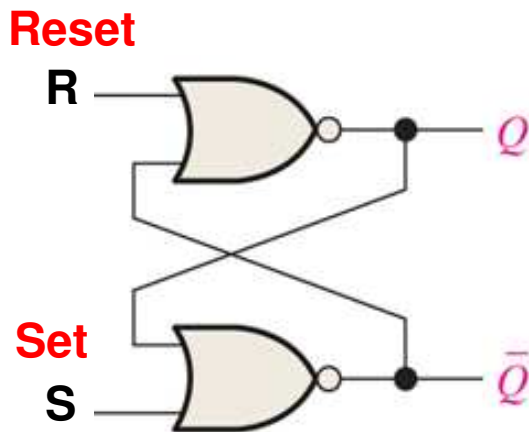
Latch/Trinco R S

Circuito implementada com portas não-OU (NOR), com duas entradas, R e S, correspondentes às **funcionalidades Reset e Set**.



Exemplo de um circuito lógico com realimentação. O acoplamento cruzado das duas portas produz **realimentação regenerativa** (i.e., realimentação **positiva**).

Qual é a tabela de verdade deste circuito? Considere inicialmente que $S=0$ e $R=0$, $Q=1$



Entrada inicial
(quando ligamos)


		Saídas	
R	S	Q	\bar{Q}
0	0	1	0
0	1	?	?
1	0	?	?
1	1	?	?
1	0	?	?
0	0	?	?

T.V. NOR	
Inputs	Output
A B	X
0 0	1
0 1	0
1 0	0
1 1	0

Este circuito é biestável (tem dois estados estáveis) e por isso é conhecido como biestável trinco ou “latch” RS.

Latch/trinco R S

(tem dois estados estáveis)

A ———  ——— X
 B ———

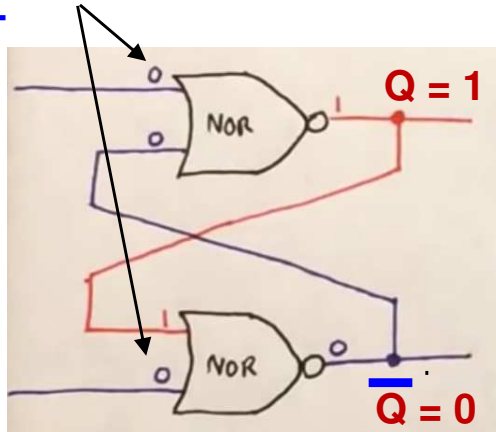
T.V. NOR

Inputs		Output
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Estado inicial (estado após ligarmos o circuito)

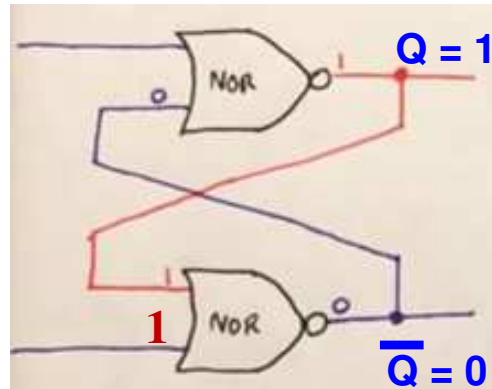
RESET
(R) 0

SET
(S) 0



O que acontece a Q e a \overline{Q} se agora S passar de 0 para 1?

0

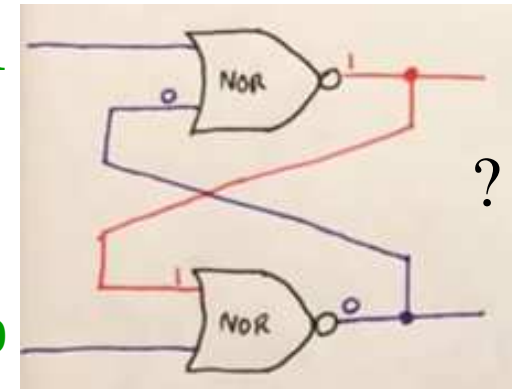


1

0 → 1

1 → 0

E se agora mudarmos as entradas R e S para 1 e 0?



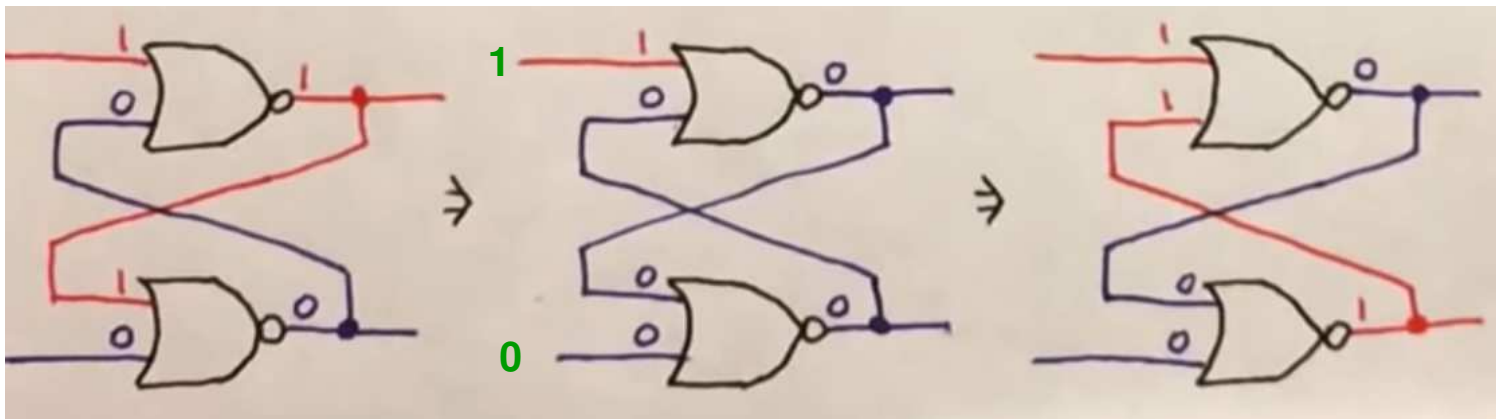
$Q = ?$

?

$\overline{Q} = ?$

1

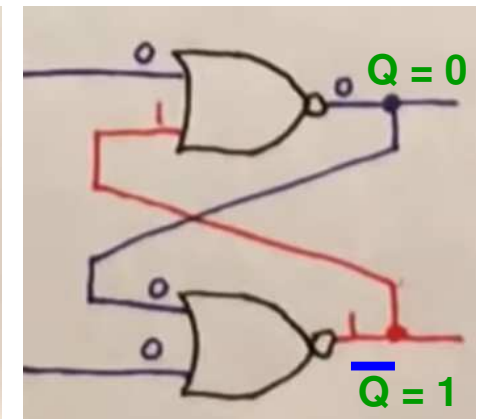
0



(saídas: situação antes das portas reagirem à alteração de S e R)

(situação após a reação da porta de cima à alteração de R)

(situação após a reação da porta de baixo à alteração de R)



(situação após a reação das duas portas)

Nas discussões que se seguem consideram-se todos os elementos como sendo ideais; por isso os tempos de propagação dos sinais é considerado como sendo nulos.

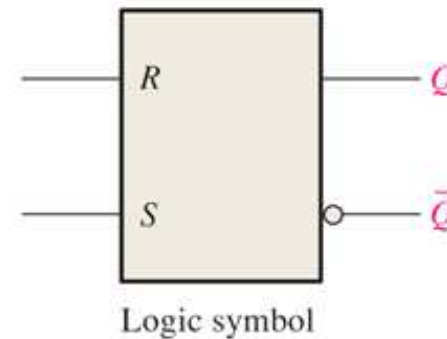
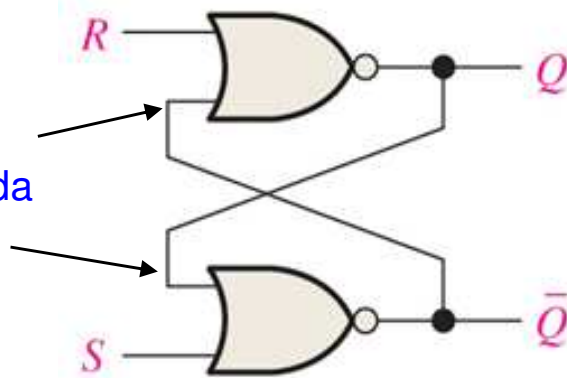
Trinco/Latch RS

O latch RS realiza a função memória combinando as duas portas lógicas NOR idênticas de duas entradas de forma que a saída de cada uma das portas vai alimentar uma das entradas da outra porta.

O acoplamento cruzado das duas portas produz realimentação regenerativa.

A entrada da porta que produz Q chama-se RESET, e a entrada da segunda porta que produz o inverso de Q /complemento de Q, \bar{Q} , designa-se por SET.

Estas entradas correspondem ao valores anteriores da saída, ou seja dependem de um valor passado



S	R	Q_{n+1}	\bar{Q}_{n+1}
0	0	Q_n	\bar{Q}_n
0	1	0	1
1	0	1	0
1	1	NU	

NU: não usado

Equação do trinco RS:

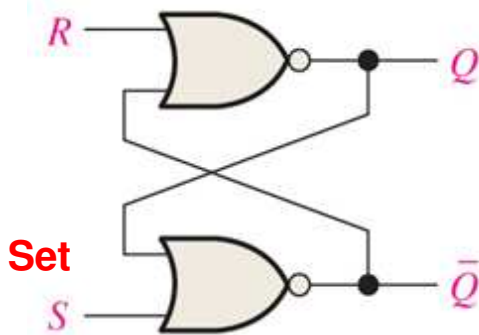
$$Q_{n+1} = S + \bar{R} \cdot Q_n$$

Este circuito é capaz de reter os valores lógicos das saídas: se $R=S=0$ ou se R ou S passarem de 1 a 0, a saída não é alterada, mantendo o seu valor (que tinha antes da alteração de R ou S de 1 para 0).

Latch/trinco R S

Latch RS implementada com portas não-OU (NOR), com duas entradas, R e S, correspondentes às **funcionalidades** *Reset* e *Set*.

Reset



S	R	Q_{n+1}	\overline{Q}_{n+1}
0	0	Q_n	\overline{Q}_n
0	1	0	1
1	0	1	0
1	1	NU	NU

NU: não usado

Q_n (\overline{Q}_n) representa a saída da latch no instante de tempo n (instante de tempo anterior à ação a executar) e Q_{n+1} (\overline{Q}_{n+1}) indica a saída no instante de tempo n+1 (o instante de tempo posterior à ação a executar).

Equação do trinco RS

$$Q_{n+1} = S + \overline{R} \cdot Q_n$$

Este estado corresponde a **memória** (o latch mantém a saída, i.e. o estado anterior).

Este circuito é biestável (tem dois estados estáveis) e é conhecido como biestável/ trinco RS ou “latch” RS:

A saída Q do latch RS **passa a 0** quando **R passa de 0 a 1**, com **S=0**;

A saída Q do latch RS **passa a 1** quando **S passa de 0 a 1**, com R=0.

O latch não produz saídas Q e não-Q coerentes se R e S forem **altos** (“1”) em simultâneo.

Reset reinicia ou coloca a 0 (Q baixo) o latch, que no caso das *latches* significa colocar a saída Q a 0.

Set ATIVA ou seja coloca a saída Q a 1 (Alto).

Latch SR habilitado

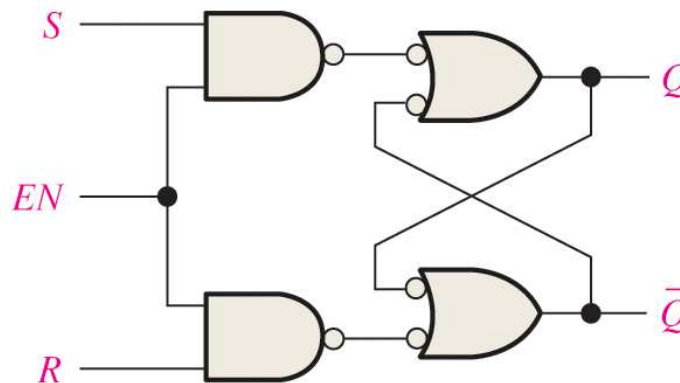
Há situações em que é conveniente que os biestáveis **estejam ativos** (**habilitados**) apenas durante certos intervalos de tempo pré-determinados, e que mantenham as suas saídas inalteradas no restante tempo, independentemente de alterações que possam ocorrer no valor das entradas.

Os biestáveis podem ser habilitados/ativados por um **estado** ou por uma **transição de estados**.

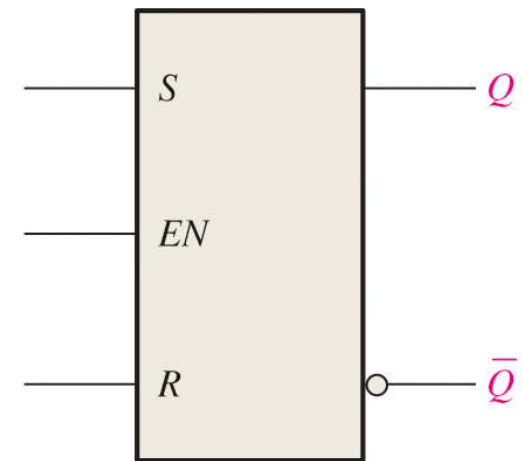
Os latch (trincos) são transparentes, isto é, as suas saídas são atualizadas sempre que a sua variável de habilitação (EN, de “enable”) esteja ativa. A variável de habilitação pode ser ativada com “1” ou com “0”.

Latch com EN ativo em “1”:

EN: entrada de habilitação/ativação



(a) Logic diagram



(b) Logic symbol

As entradas S e R **controlam** o estado para o qual o latch irá comutar **quando** um nível ALTO é aplicado (e/ou mantido) à entrada EN. O latch não mudará de estado até que EN seja nível ALTO.

Neste circuito o **estado inválido** ocorre quando S e R estiveram simultaneamente em ALTO.

Exercício: Latch SR habilitado

Determine the Q output waveform if the inputs shown in Figure 7–9(a) are applied to a gated S-R latch that is initially RESET.

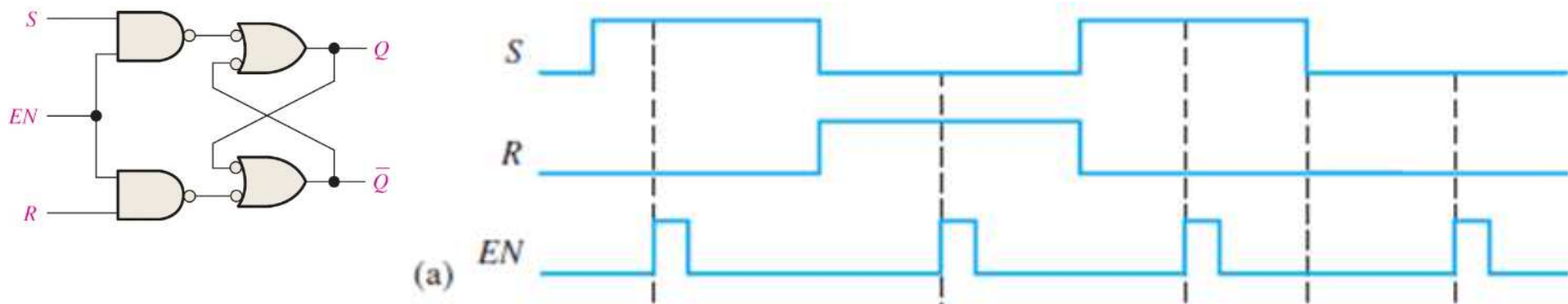


FIGURE 7-9

Solution

The Q waveform is shown in Figure 7–9(b). When S is HIGH and R is LOW, a HIGH on the EN input sets the latch. When S is LOW and R is HIGH, a HIGH on the EN input resets the latch. When both S and R are LOW, the Q output does not change from its present state.

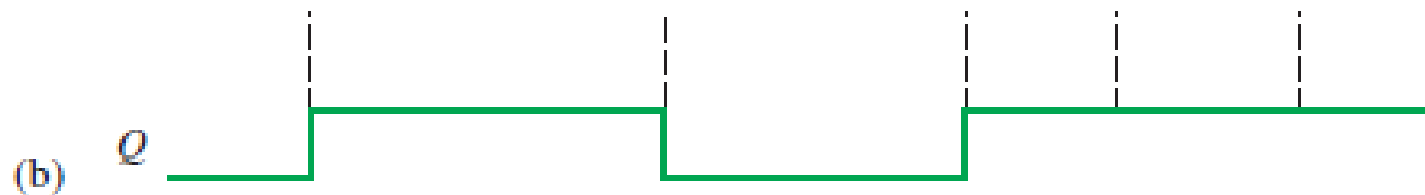


FIGURE 7-9

Latch J K

O Latch J-K elimina o efeito indesejado da combinação RS="11", que produz um valor indefinido à saída, com a introdução de duas portas NAND, em que uma das entradas é ligada à saída da segunda porta e a outra corresponde à entrada J ou K.

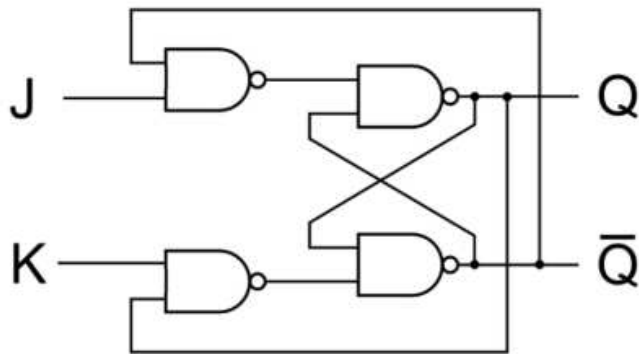


Tabela funcional da latch J-K

Entradas		Estado seguinte
J (SET)	K (Reset)	Q_{N+1}
0	0	Q_N
0	1	0
1	0	1
1	1	\bar{Q}_N

Equação do trinco JK: $Q_{n+1} = J \cdot \bar{Q}_n + \bar{K} \cdot Q_n$

Latch D

(Bistável/trinco tipo D / D de delay)

Se impusermos no latch RS que R e S são sempre diferentes obtém-se o latch D:

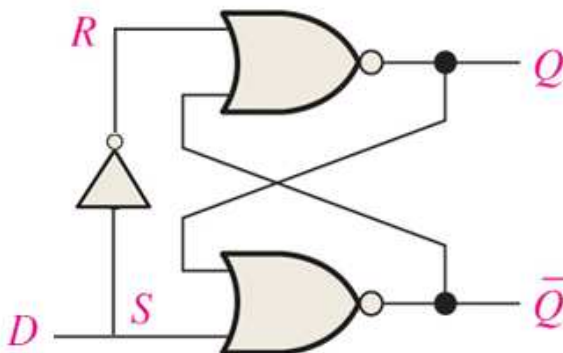


Tabela de estados do latch RS

S	R	Q_{n+1}	\bar{Q}_{n+1}
0	0	Q_n	\bar{Q}_n
0	1	0	1
1	0	1	0
1	1	NU	

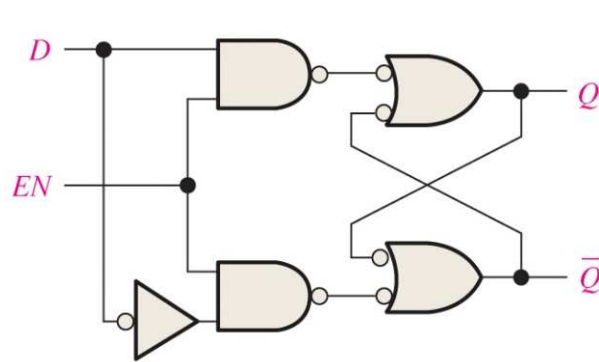
NU: não usado

Tabela de estados do latch D

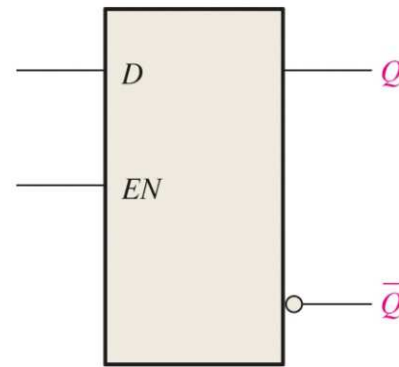
D	\bar{D}	Q_{n+1}	\bar{Q}_{n+1}
0	1	0	1
1	0	1	0

Equação do trinco D: $Q_{n+1} = D$

Latch D habilitado



(a) Logic diagram



(b) Logic symbol

D	\bar{D}	Q_{n+1}	\bar{Q}_{n+1}
0	1	0	1
1	0	1	0
1	1	NU	

NU: não usado

$$Q_{n+1} = D \cdot EN + Q_n \cdot \bar{EN}$$

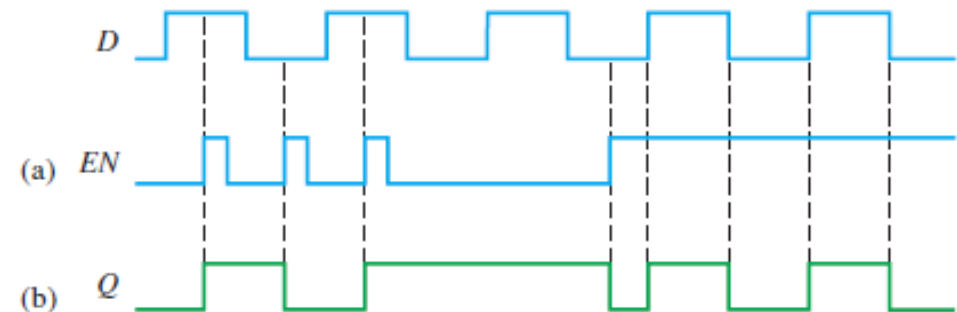
Regra simples: O latch D segue a entrada D quando a entrada EN está ativa.

Exercício

Determine the Q output waveform if the inputs shown in Figure 7–11(a) are applied to a gated D latch, which is initially RESET.

Solution

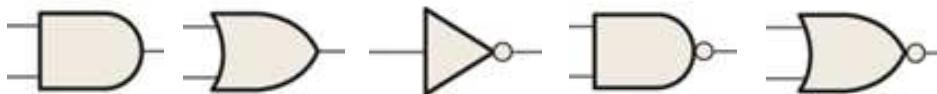
The Q waveform is shown in Figure 7–11(b).
 When D is HIGH and EN is HIGH, Q goes HIGH.
 When D is LOW and EN is HIGH, Q goes LOW.
 When EN is LOW, the state of the latch is not affected by the D input.



Flip-Flops (FF)

As **básculas** ou “**flip-flops**” são circuitos biestáveis síncronos pois necessitam de um **sinal de relógio** para funcionarem; podem ser **sensíveis ao valor lógico do sinal (0 ou 1)** ou às **transições de sinais** (e.g. de 1 p/ 0 ou 0 p/ 1). Isto é: os flip-flops podem ser:

- i) sensíveis ao **bordo ascendente** do sinal de relógio;
- ii) sensíveis ao bordo **descendente** do sinal de relógio; ou
- iii) sensíveis ao **valor lógico** do sinal de relógio.



Flip-flops

Os latch são circuitos assíncronos, pois não necessitam de sinal de relógio, isto é, a saída pode mudar de estado a qualquer momento. Quando um biestável latch é ativado por um sinal de transição de estado (e.g., nível de sinal ou flanco/ borda) passa a designar-se **por báscula (“flip-flop”, FF)**.

Assim, o flip-flop difere do latch na maneira como muda de estado: o estado de um FF é controlado pelo sinal de relógio, e **a mudança de estado** (isto é, **a alteração do valor do bit de saída**) é determinada ou pelo nível de sinal ou pelas transições flanco/borda ascendentes ou descendentes do sinal de relógio.

A grande maioria dos dispositivos digitais sequenciais opera no **modo síncrono**. Os circuitos assíncronos tendem a ser mais complexos que os síncronos, pois os seus elementos podem responder em períodos temporais diferentes.

Um flip-flop consegue armazenar 1 bit de informação. Para guardar 2 bits de informação são necessários 2 FF; para armazenar 3 bits de informação (por exemplo, um dígito octal), são necessários 3 FF; para guardar 4 bits de informação (por exemplo um dígito hexadecimal), são necessários 4 FF.

Habitualmente, a informação é guardada em múltiplos de 8 bits, por exemplo, 8, 16, 32, 64, ... Se se pretender armazenar vários bits de informação, como por exemplo no **armazenamento de dados (registos)**, é necessário combinar vários FF.

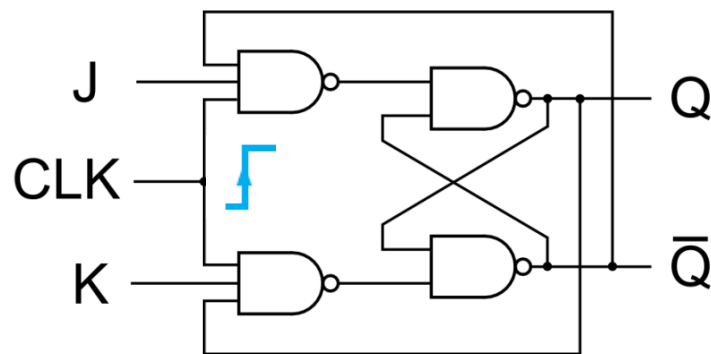
Se os FFs forem sincronizados com o mesmo sinal de relógio podem armazenar os bits em simultâneo.

Flip-Flops (FFs)

Circuitos biestáveis síncronos são designados **básculas** ou “**flip-flops**”. Nestes circuitos o estado muda apenas com os **sinais de relógio**, isto é, necessitam de um **sinal de relógio** para funcionarem. **Os flip-flops podem ser:**

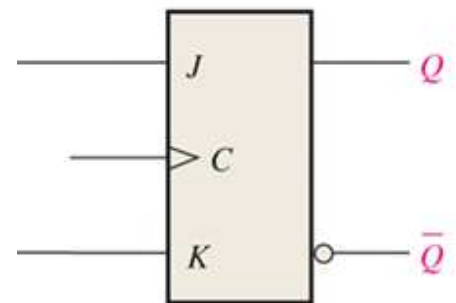
- i) sensíveis ao **bordo ascendente** do sinal de relógio (de 0 p/ 1);
- ii) sensíveis ao bordo **descendente** do sinal de relógio (de 1 p/ 0); ou
- iii) sensíveis ao **valor lógico** do sinal de relógio.

As figuras abaixo representam uma das possíveis implementações do flip-flop J-K e o símbolo de um flip-flop ativado pelo **pulso ascendente** do sinal de relógio. O comportamento de um flip-flop é descrito por sua equação característica, que prevê a "próxima" (após o próximo pulso de clock) saída, Q_{n+1} em termos dos sinais de entrada e/ou da saída atual, Q_n .



K	J	CK	Q_{n+1}	$\overline{Q_{n+1}}$	
0	0	↑	Q_n	$\overline{Q_n}$	Repouso
0	1	↑	1	0	RESET
1	0	↑	0	1	SET
1	1	↑	$\overline{Q_n}$	Q_n	Toggle* *alterna

Flanco ascendente

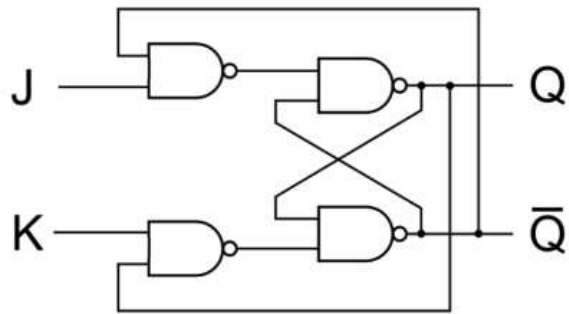


Equação característica do FF JK:

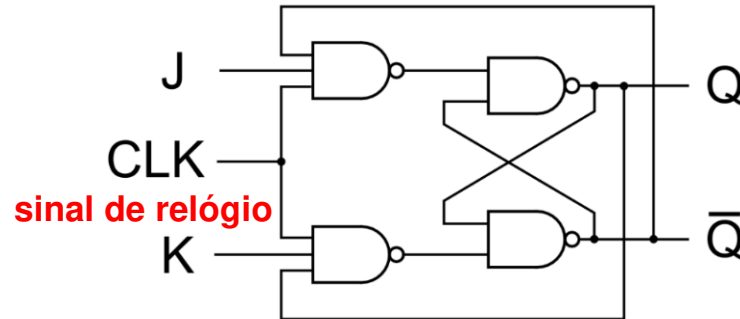
$$Q_{n+1} = J \cdot \overline{Q_n} + \overline{K} \cdot Q_n$$

Para além do J-K, temos o flip-flop T ("toggle"), flip-flop S-R ("set-reset"), e o flip-flop D ("data").

Latch J K vs Flip-flop J K



Latch J K



Flip-Flop J K



Símbolo do Flip-Flop J K

Como acontece no latch J K, no flip-flop JK a combinação RS="11" é igualmente possível porque flip-flop também tira partido da lógica combinacional que se adicionou ao latch JK.

O funcionamento do flip-flop J-K de disparo de borda/flanco é idêntico ao do flip-flop RS nas condições de SET, RESET e repouso (memória). Porém o flip-flop J-K aprimora o funcionamento do flip-flop R-S interpretando a condição **S = R = 1** como um comando de inversão (Não-Q), e deixa de haver o estado inválido.

Neste exemplo acima, os valores das entradas J e K determinam, em conjunto com o disparo de flanco **ascendente** do pulso de relógio, o estado de saída.

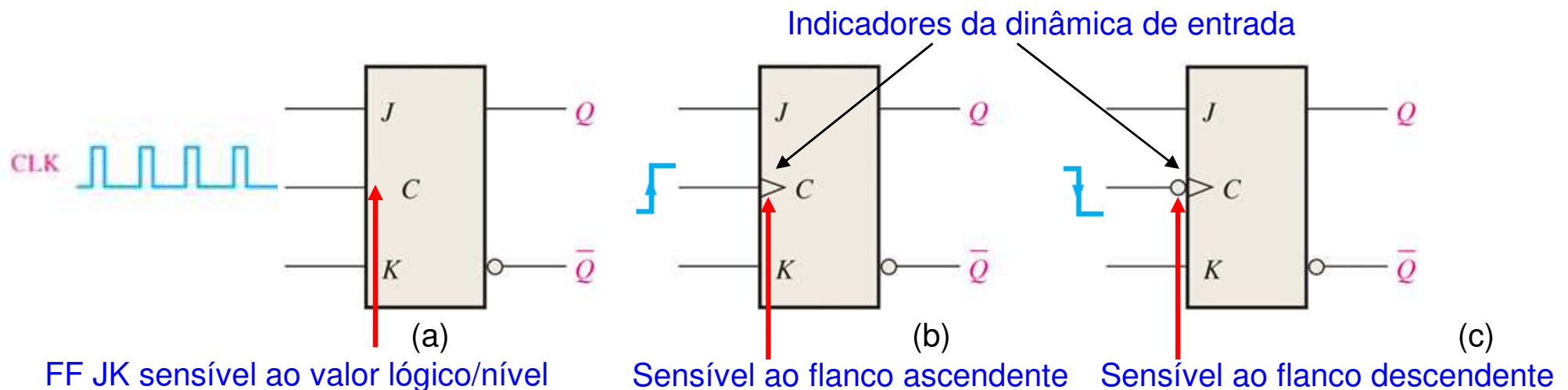
Especificamente, a combinação **J = 1, K = 0** corresponde ao “comando” para ativar (SET) a saída do flip-flop; a combinação **J = 0, K = 1** ao “comando” para desativar (RESET) a saída do flip-flop;

e a combinação **J = K = 1** provoca a inversão das saídas do flip-flop, isto é, troca o sinal de saída Q pelo seu complemento, e o complemento por Q.

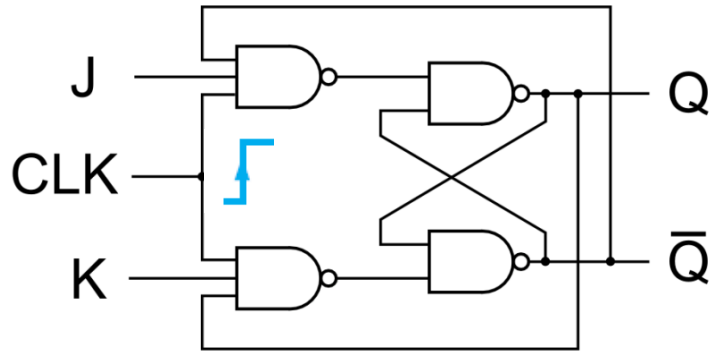
Flip-flops e o sinal de relógio

Os flip-flops podem ser sensíveis (a título de exemplo usa o FF JK):

- ao **nível lógico do relógio** (“**level-triggered**”), Fig. (a), quando o estado nas saídas só é alterado pelo valor lógico do sinal de relógio (alto ou baixo), conforme o flip-flop seja sensível ao valor alto ou ao valor baixo.
- à **borda ascendente e à borda descendente do sinal de relógio** (“**edge-triggered**”)
 - se o flip-flop responder apenas durante a transição de 0 para 1 do relógio designa-se por flip-flop sensível ao bordo **ascendente ou à transição positiva** do relógio (representado por um triângulo), Fig. (b);
 - se o flip-flop responder apenas durante a transição de 1 para 0 do relógio o flip-flop é sensível ao **bordo descendente ou à transição negativa do relógio** (representado por um triângulo complementado por um inversor), Fig. (c).



Flip-flop J-K

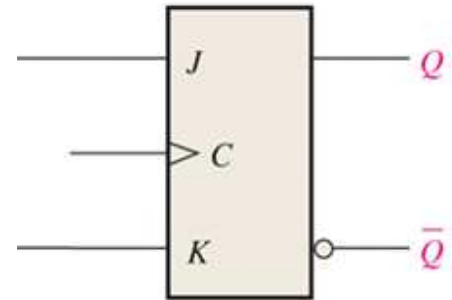


Equação do FF JK:

$$Q_{n+1} = J \cdot \overline{Q_n} + \overline{K} \cdot Q_n$$

K	J	CK	Q_{n+1}	$\overline{Q_{n+1}}$	
0	0	↑	Q_n	$\overline{Q_n}$	Repouso
0	1	↑	1	0	RESET
1	0	↑	0	1	SET
1	1	↑	$\overline{Q_n}$	Q_n	Toggle* *alterna

Flanco ascendente



$$Q_{n+1} = J \cdot \overline{Q_n} + \overline{K} \cdot Q_n$$

Consideremos o caso em que o flip-flop ativado por flanco ascendente está *resetado* ($Q=0$), com a entrada J em ALTO e a entrada K em BAIXO.

Quando um pulso de relógio "chega" à entrada CLK, a borda de subida (ascendente) ativa o flip-flop, que passa para o estado SET ($Q=1$), uma vez que J está em ALTO e o Não-Q era ALTO, até à chegada do pulso (o Não-Q passa a baixo).

Se agora fizermos $J=BAIXO$ e $K=ALTO$, a próxima borda ascendente do pulso a chegar faz com que o flip-flop mude para o estado RESET: K é ALTO e mas Q continua ALTO até à chegada do pulso; só a chegada do pulso comuta Não-Q para ALTO, e Q para BAIXO.

Se ambas as entradas estiveram em BAIXO não há mudança de estado.

Se forem ambas ALTO há mudança (comutação/alteração) de estado (para o estado oposto) a cada chegada do pulso ascendente do sinal de relógio. **Quando $J=K=1$, o flip-flop designa-se T (ver adiante).**

Entradas assíncronas “Preset” e “Clear”

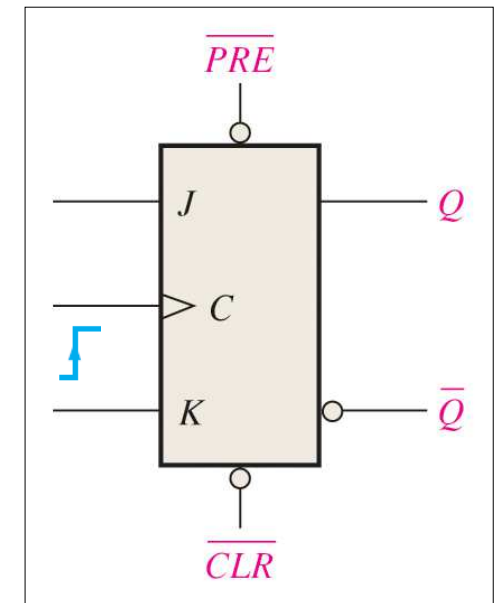
Em geral os flip-flops contêm entradas (no caso do FF JK, as entradas J e K) denominadas entradas **síncronas** porque os dados nessas entradas são transferidos para a saída do flip-flop apenas em resposta ao sinal de relógio, ou seja **os dados são transferidos de forma sincronizada com o sinal de relógio**.

Porém a maioria dos flip-flops em circuitos integrados contêm também **entradas assíncronas**, que afetam o estado do flip-flop independentemente do relógio. Normalmente estas entradas são denominadas **preset** (PRE) e **clear** (CLR), ou “**seta direto**” (S_D) e “**reseta direto**” (R_D) por alguns fabricantes:

- Um nível ativo na entrada **preset** irá **setar** o flip-flop.
- Um nível ativo na **entrada clear** irá **resetar o flip-flop**.

As entradas de *preset* (PRE) e *clear* (CLR) têm que ser mantidas em nível ALTO para a operação síncrona.

O símbolo lógico para um flip-flop J-K com entradas *preset* e *clear* está indicado na figura ao lado, para o caso de **entradas ativas** em nível BAIXO, conforme indicado pelos pequenos círculos.

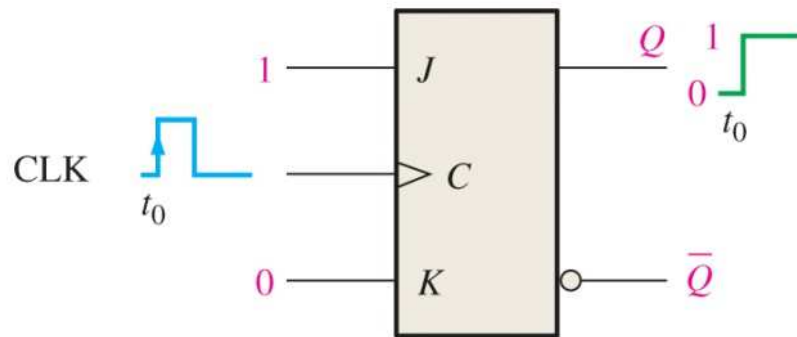


Combinações da entradas J e K no flip-flop J-K

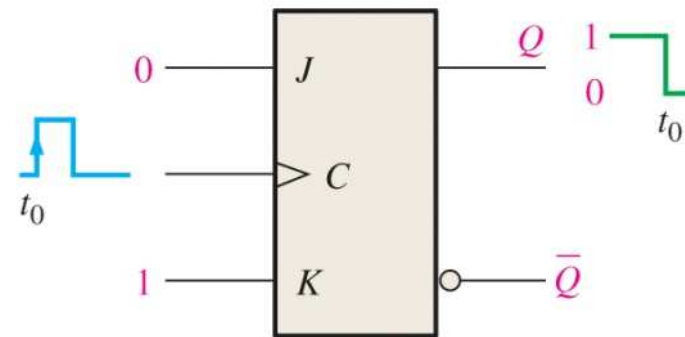
Os valores das entradas J e K do flip-flop JK determinam os estados da saída.

Saída para as quatro possibilidade de entradas:

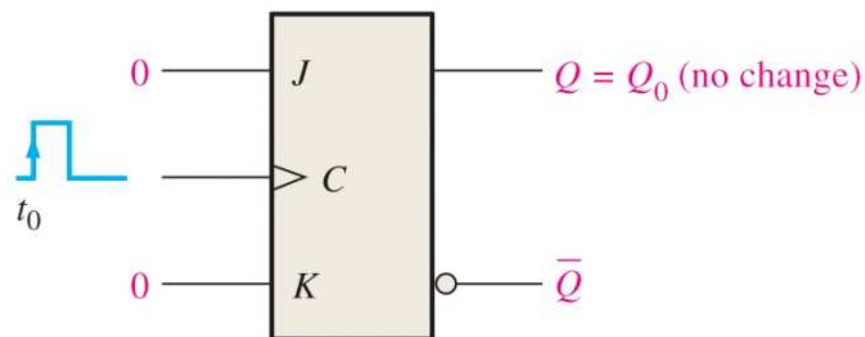
$$Q_{n+1} = J \cdot \overline{Q_n} + \overline{K} \cdot Q_n$$



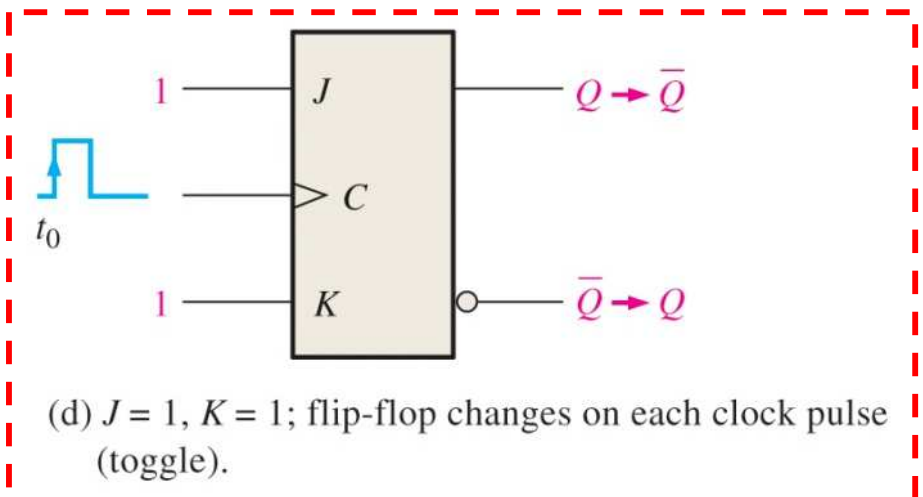
(a) $J = 1, K = 0$; flip-flop SETS on positive clock edge. (If already SET, it remains SET.)



(b) $J = 0, K = 1$; flip-flop RESETS on positive clock edge. (If already RESET, it remains RESET.)

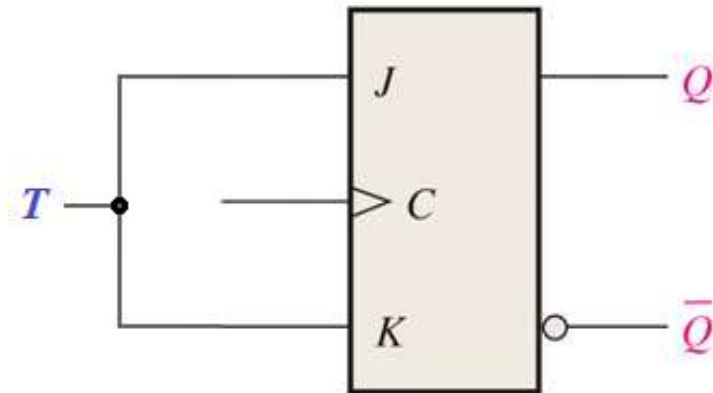
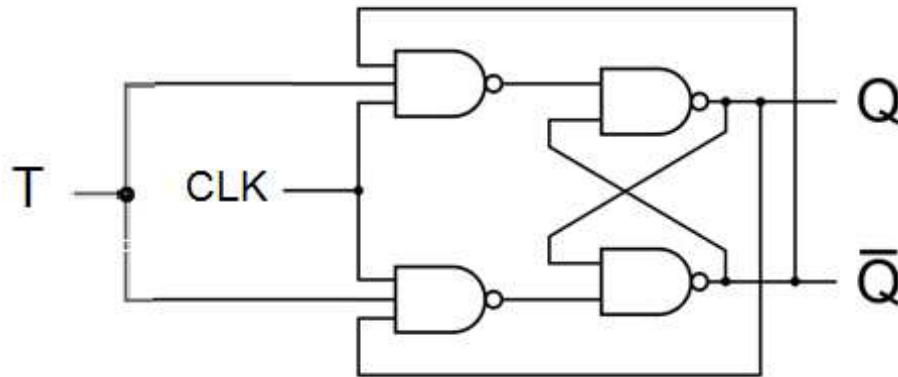


(c) $J = 0, K = 0$; flip-flop does not change. (If SET, it remains SET; if RESET, it remains RESET.)



(d) $J = 1, K = 1$; flip-flop changes on each clock pulse (toggle).

Flip-flop tipo T (Toggle/Comutação)



$$Q_{n+1} = T \cdot \overline{Q_n} + \overline{T} \cdot Q_n$$

Como vimos, no biestável JK os valores das entradas J e K determinam univocamente o seu estado de saída, em conjunto com o pulso de relógio.

Assim, se ambas as entradas forem BAIXO não há mudança de estado.

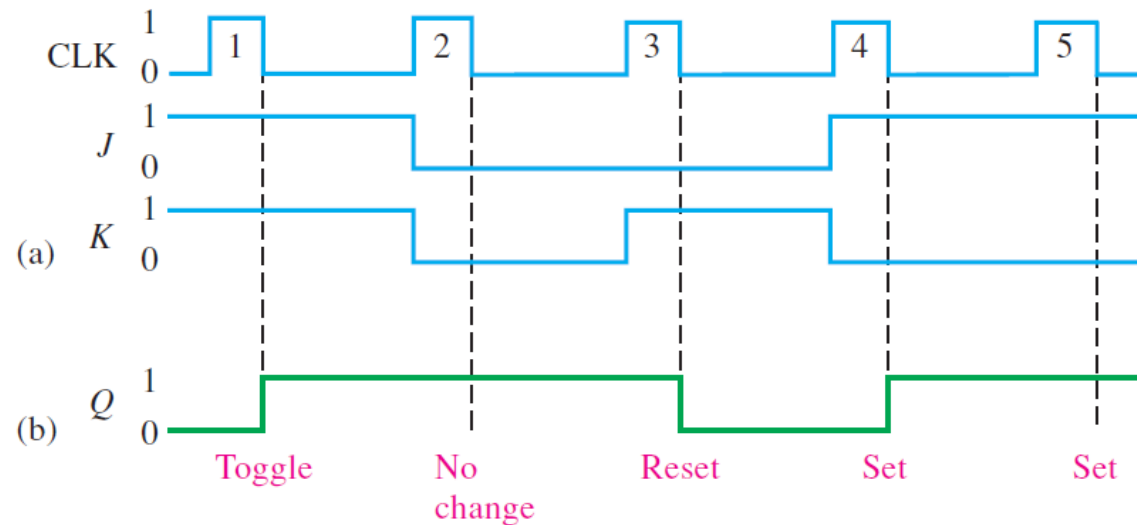
Se ambas forem ALTO há mudança (**comutação**) de estado para o estado oposto a cada transição de relógio sucessiva.

O flip-flop J-K conectado no modo **toggle** (“**alterna**”), i.e., J=K=1, é denominado **flip-flop tipo T**.

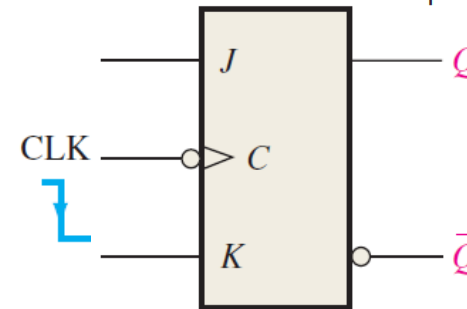
A saída do **flip-flop tipo T** muda de estado sempre que o biestável é habilitado pelo sinal de relógio.

Exercício: Flip-flop J K

The waveforms in Figure 7–18(a) are applied to the J , K , and clock inputs as indicated. Determine the Q output, assuming that the flip-flop is initially RESET.



K	J	CK	Q_{n+1}	\overline{Q}_{n+1}	
0	0	↑	Q_n	\overline{Q}_n	Repouso
0	1	↑	1	0	RESET
1	0	↑	0	1	SET
1	1	↑	\overline{Q}_n	Q_n	Toggle*



$$Q_{n+1} = J \cdot \overline{Q}_n + \overline{K} \cdot Q_n$$

FIGURE 7–18

Solution

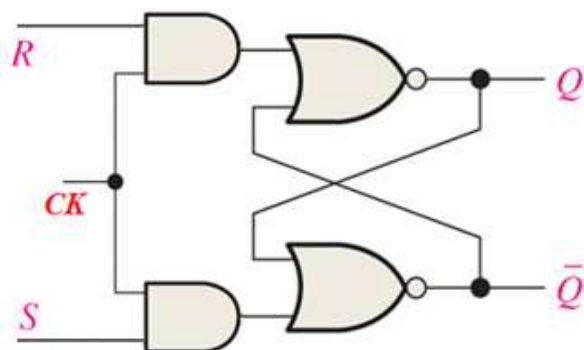
Since this is a negative edge-triggered flip-flop, as indicated by the “bubble” at the clock input, the Q output will change only on the negative-going edge of the clock pulse.

1. At the first clock pulse, both J and K are HIGH; and because this is a toggle condition, Q goes HIGH.
2. At clock pulse 2, a no-change condition exists on the inputs, keeping Q at a HIGH level.
3. When clock pulse 3 occurs, J is LOW and K is HIGH, resulting in a RESET condition; Q goes LOW.
4. At clock pulse 4, J is HIGH and K is LOW, resulting in a SET condition; Q goes HIGH.
5. A SET condition still exists on J and K when clock pulse 5 occurs, so Q will remain HIGH.

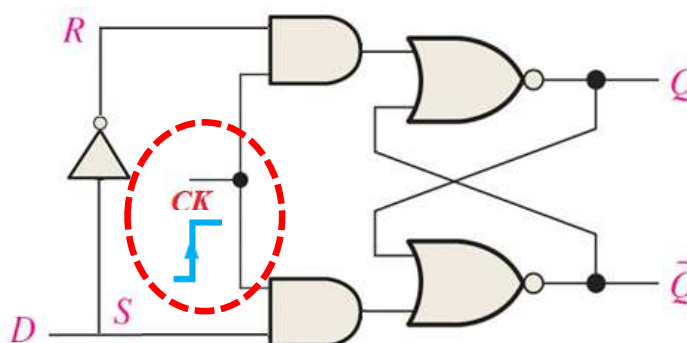
The resulting Q waveform is indicated in Figure 7–18(b).

Flip-flop/Báscula D (“data delay”)

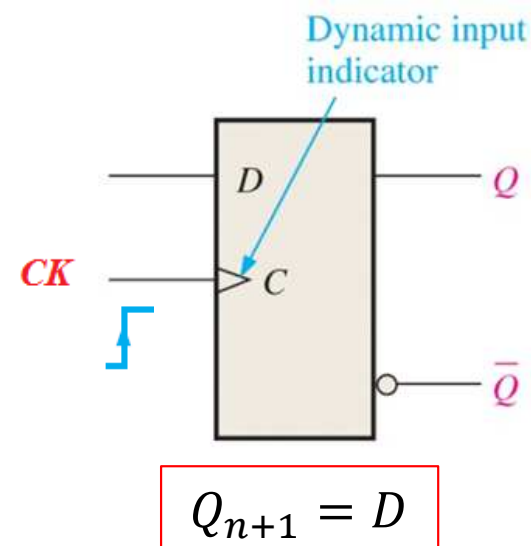
As figuras da direita correspondem ao circuito representativo (não inclui a representação do circuito de disparo) e ao símbolo de um **flip-flop do tipo D** ativo ao flanco ascendente (borda de subida – borda positiva) do pulso de relógio ou entrada de disparo (relógio, C ou CK):



Flip-flop RS



Flip-flop D



O circuito representa um flip-flop (báscula) que é sensível à transição de estado por flanco/borda ascendente do sinal de controlo (e.g. o sinal de relógio CK, CK de “clock”).

Básculas ativas ao flanco/borda ascendente (transições BAIXO (0) → ALTO (1) e Básculas ativas ao flanco/borda descendente (transição 1 para 0):

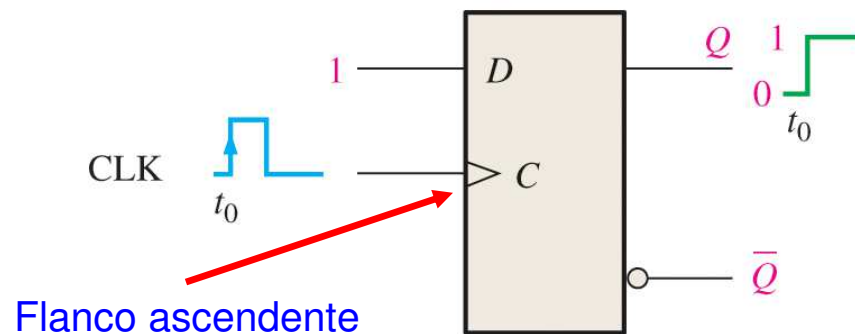
A transição ascendente também se designa transição positiva (transição de 0 para 1); a transição descendente é também conhecida como transição negativa (transição de 1 para 0).

Flip-flop D

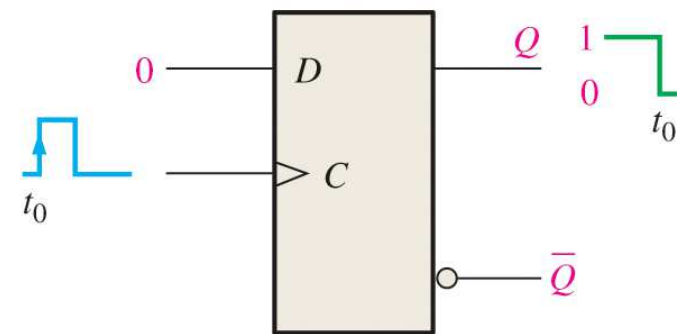
Conforme vimos, a adição de um inversor a ligar as entradas R e S de um flip-flop RS cria um flip-flop D básico, que tem apenas a entrada D, além do relógio.

O flip-flop D é usado quando se pretende armazenar um único bit (1 ou 0). No estado SET o flip-flop armazena um nível 1 e no estado RESET armazena um nível 0.

O bit de dados da entrada D é transferido para a saída Q em resposta ao flanco/borda do sinal de relógio (no caso da figura, flanco ascendente).



(a) $D = 1$; flip-flop SETS on positive clock edge. (If already SET, it remains SET.)



(b) $D = 0$; flip-flop RESETS on positive clock edge. (If already RESET, it remains RESET.)

Truth table for a positive edge-triggered D flip-flop.

Inputs		Outputs		Comments
D	CLK	Q	\bar{Q}	
0	↑	0	1	RESET
1	↑	1	0	SET

↑ = clock transition LOW to HIGH

Flanco ascendente

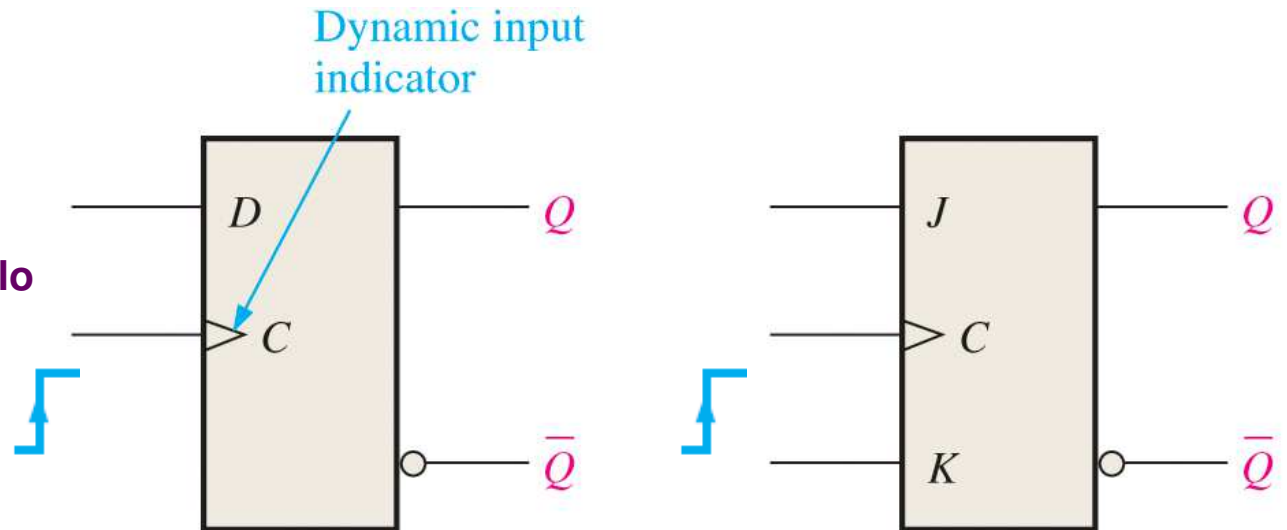
$$Q_{n+1} = D$$

Uma vez “ativado”, a saída Q iguala o valor da entrada D até que um novo sinal de ativação (flanco ascendente do sinal de relógio) chegue.

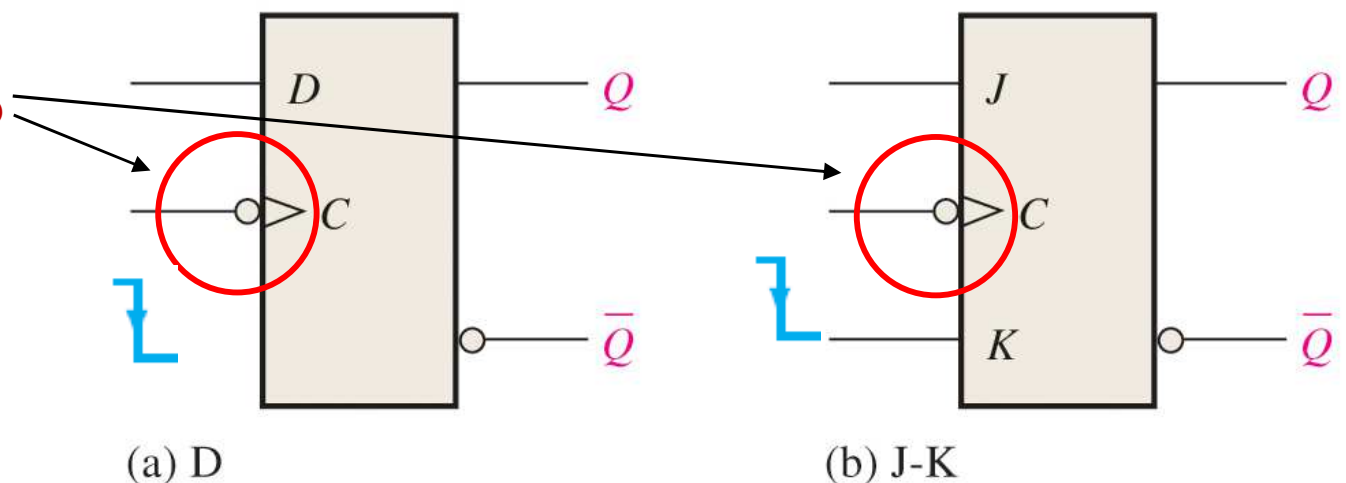
Flip-flops D ativados por flanco positivo/negativo

A saída de um flip-flop ativado por flanco muda em resposta à transição positiva ou à transição negativa do sinal de relógio C.

A entrada de relógio sem círculo indica que o flip-flop é ativado pelo flanco ascendente.

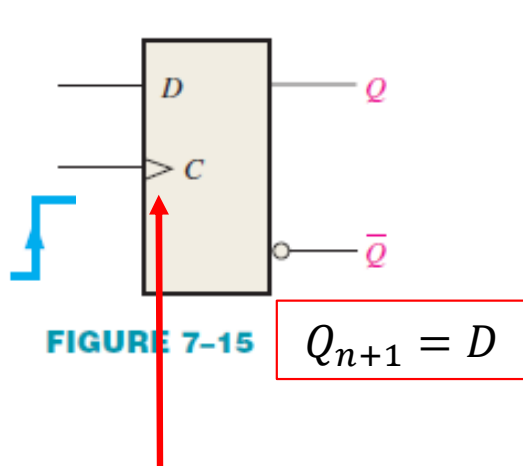


O círculo na entrada de relógio indica que o flip-flop é ativado pelo flanco descendente.

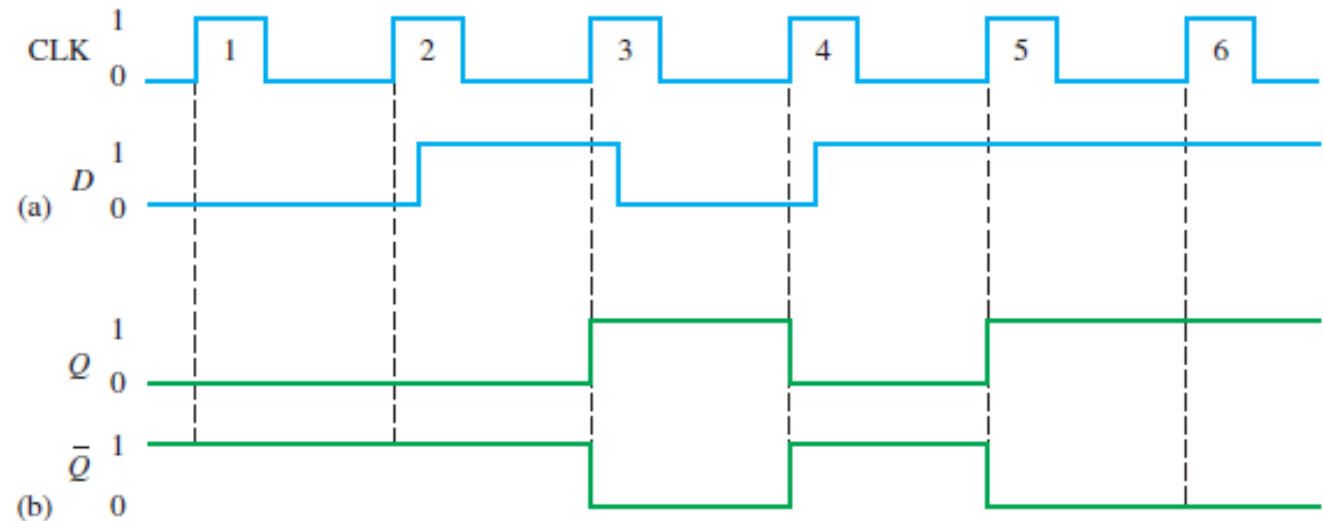


Exercício: Flip-flop D

Determine the Q and \bar{Q} output waveforms of the flip-flop in Figure 7–15 for the D and CLK inputs in Figure 7–16(a). Assume that the positive edge-triggered flip-flop is initially RESET.



Flanco ascendente



Solution

1. At clock pulse 1, D is LOW, so Q remains LOW (RESET).
2. At clock pulse 2, D is LOW, so Q remains LOW (RESET).
3. At clock pulse 3, D is HIGH, so Q goes HIGH (SET).
4. At clock pulse 4, D is LOW, so Q goes LOW (RESET).
5. At clock pulse 5, D is HIGH, so Q goes HIGH (SET).
6. At clock pulse 6, D is HIGH, so Q remains HIGH (SET).

Once Q is determined, \bar{Q} is easily found since it is simply the complement of Q . The resulting waveforms for Q and \bar{Q} are shown in Figure 7–16(b) for the input waveforms in part (a).

Aplicações mais comuns dos flip-flops

Os FFs são muito utilizados no projeto de **circuitos divisores de frequência**. Um divisor de frequência é um circuito que permite **dividir o valor da frequência do sinal** inicial (e.g. o sinal de relógio) por 2^N , em que N representa o número de FFs usados.

Caso se pretenda reduzir a frequência para metade (dividir por dois), basta utilizar apenas um FF. Se se pretender construir um divisor de frequência por 8, são necessários 3 FFs, sendo que a saída dos dois primeiros FFs deve ser ligada à entrada de relógio do FF subsequente.

Um circuito divisor de frequência pode ser considerado um **contador**, que pode ser implementado em contagem decrescente ou crescente, conforme a saída, ou a sua complementada, de cada FF seja ligada à entrada de relógio dos FFs subsequentes, ou em alternativa se se utilizar FFs sensíveis à borda descendente.

Os contadores são circuitos digitais compostos por um conjunto de FFs interligados com o propósito de realizar uma determinada sequência de contagem. Estes circuitos são muito utilizados na conceção de contadores de impulsos, divisores de frequência ou geradores de ondas faseadas.

O módulo de um contador (número inteiro) corresponde ao número de estados que o contador percorre (“conta”) em um ciclo completo de contagem. Isto é, se o módulo de um contador é o inteiro x , significa que o contador faz x contagens – número de contagens/estado diferente que o contador percorre em cada contagem.

Se um contador tem N flip-flops (ou é de N bits), o valor máximo de contagem (número de estados) em decimal é igual a $2^N - 1$. Exemplos: um contador de módulo 4 pode contar no máximo até 4 (0-3 decimal: 0, 1, 2 e 3); um contador de módulo 8 pode contar até 8 (0 - 7 decimal).

Nos **contadores assíncronos** apenas o **primeiro FF se encontra ligado ao sinal de relógio**. A saída do FF ao qual está ligado o sinal de relógio corresponde ao bit menos significativo, a saída do FF mais afastado corresponde ao bit mais significativo.

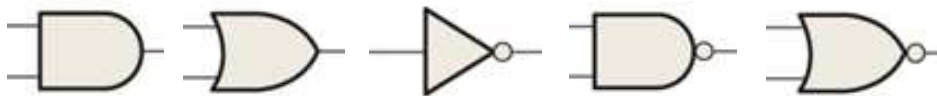
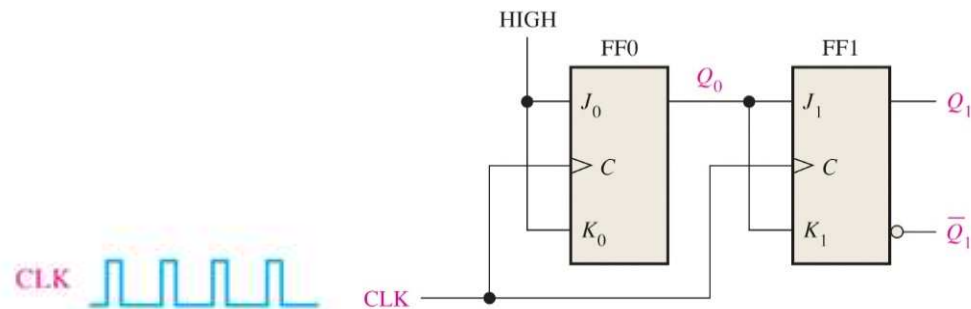
Nos **contadores síncronos**, todos os FFs estão sincronizados com o mesmo sinal de relógio.

Circuitos com memória – TL5

Latches and Flip-Flops 4 – The Clocked D Latch

[Ver: https://www.youtube.com/watch?v=8bUKw2cGcGg](https://www.youtube.com/watch?v=8bUKw2cGcGg)

Contadores síncronos

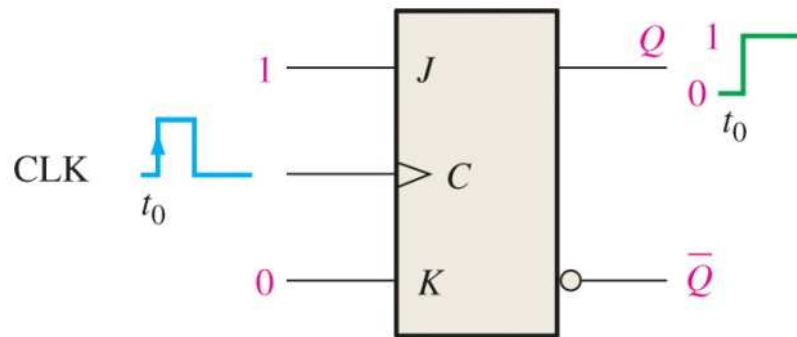


Combinações da entradas J e K no flip-flop J-K

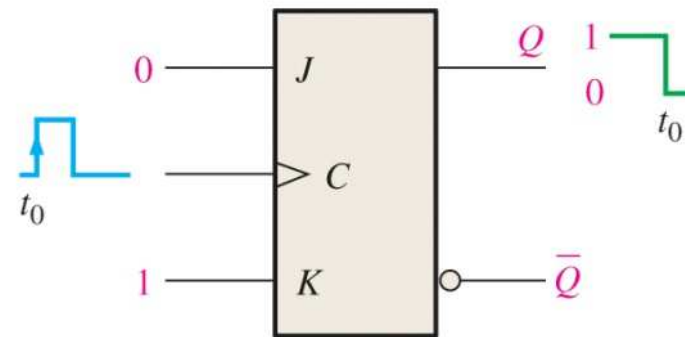
Os valores das entradas J e K do flip-flop JK determinam os estados da saída.

Saída para as quatro possibilidade de entradas:

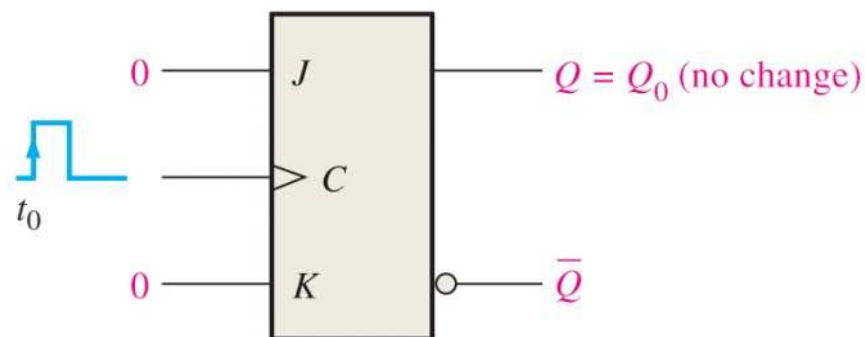
$$Q_{n+1} = J \cdot \overline{Q_n} + \overline{K} \cdot Q_n$$



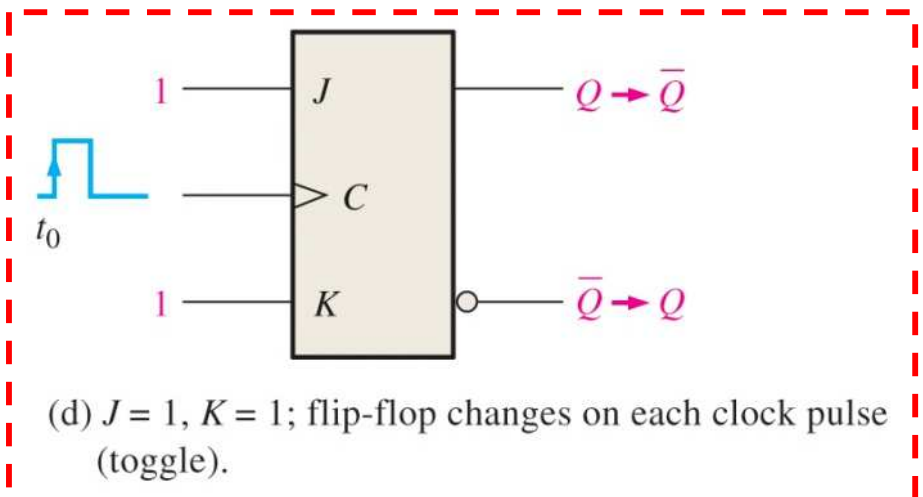
(a) $J = 1, K = 0$; flip-flop SETS on positive clock edge. (If already SET, it remains SET.)



(b) $J = 0, K = 1$; flip-flop RESETS on positive clock edge. (If already RESET, it remains RESET.)



(c) $J = 0, K = 0$; flip-flop does not change. (If SET, it remains SET; if RESET, it remains RESET.)



(d) $J = 1, K = 1$; flip-flop changes on each clock pulse (toggle).

Contador síncrono de 2 bit

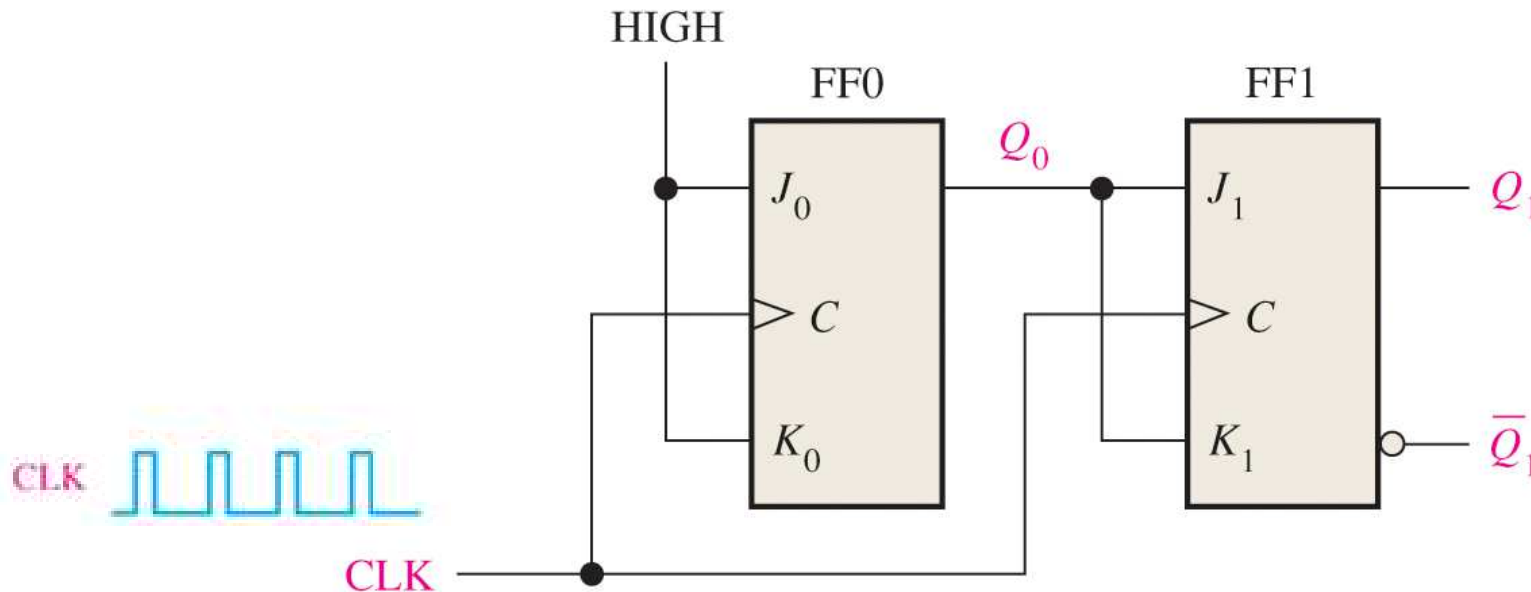
Um contador é síncrono quando os respectivos flip-flops (básculas) são ativados ao mesmo tempo, usando um sinal de relógio comum.

Nestes contadores é eliminada a desvantagem dos diferentes tempos de propagação dos sinais “através” dos flip-flops. Contudo, e por isso, requerem mais elementos para controlar as mudanças de estados.

O circuito abaixo corresponde a um contador síncrono de 2 bit, que tem a mesma sequência de contagem que o contador assíncrono de 3 bit anterior.

Flip-flops J K: $Q_{n+1} = J \cdot \overline{Q_n} + \bar{K} \cdot Q_n$.

Flip-flops T (J=K): $Q_{n+1} = T \cdot \overline{Q_n} + \bar{T} \cdot Q_n$.



Contador síncrono de 2 bit com flip-flops JK

FIGURE 9-12 2-bit synchronous binary counters.

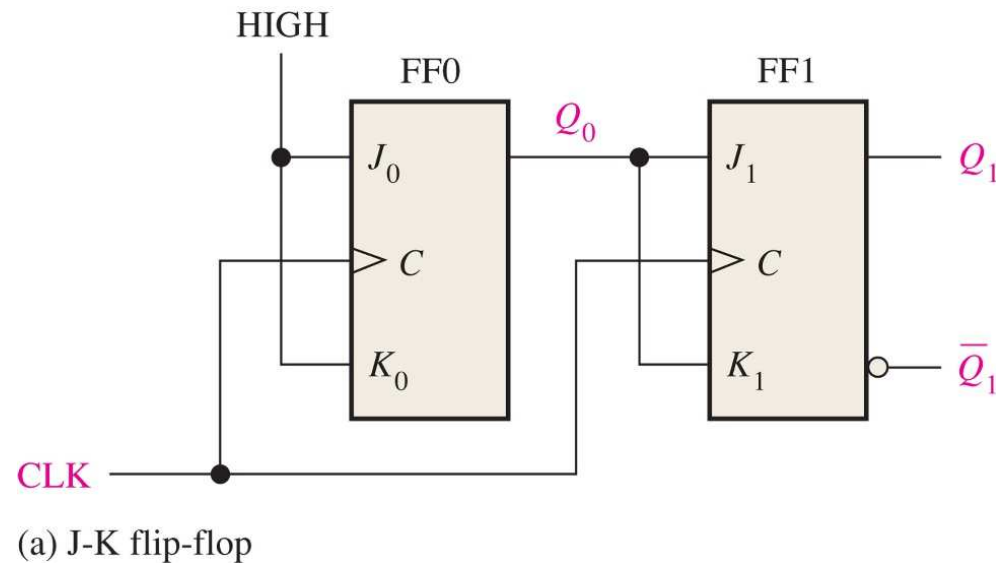
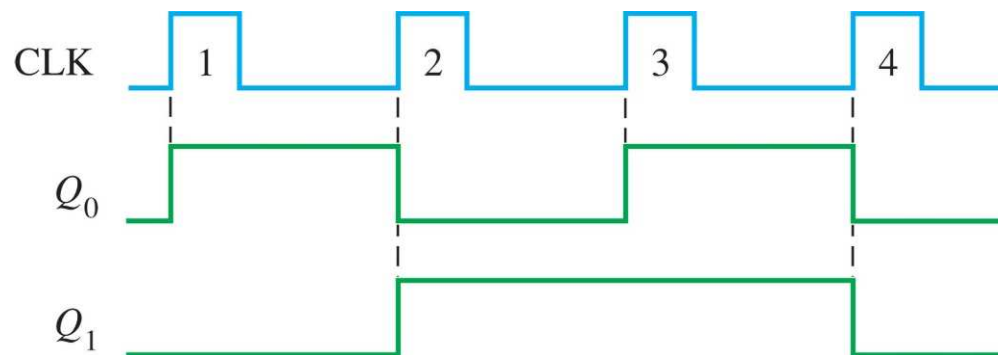


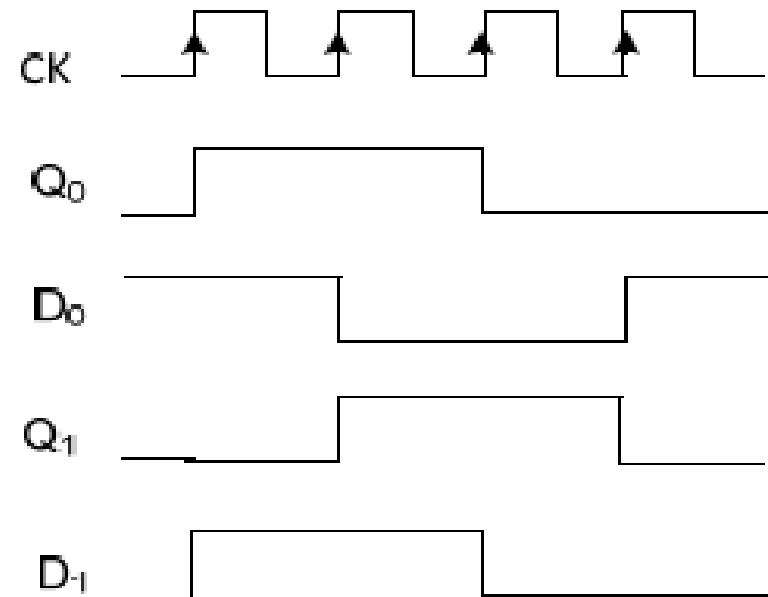
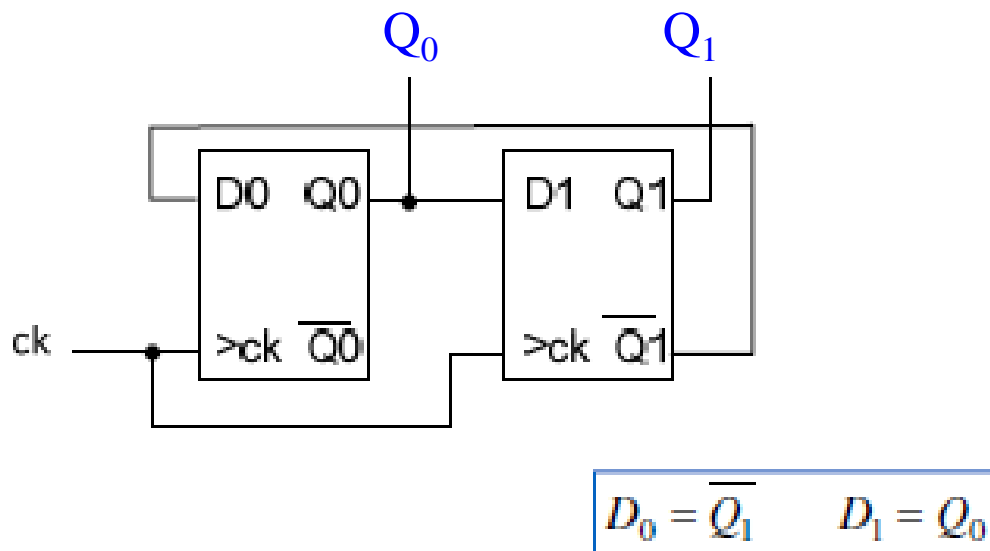
FIGURE 9-14 Timing diagram for the counters of Figure 9-12.



Contador síncrono de 2 bit com flip-flops D (contador de código Gray)

Um contador de código Gray de 2 bit pode ser implementado com 2 flip-flops do tipo D sensíveis ao flanco ascendente do sinal de relógio.

Flip-flops D: $Q_{n+1} = D$.



$$Q_1Q_0 \rightarrow '00' \rightarrow '01' \rightarrow '11' \rightarrow '10' \rightarrow '00'$$

Conforme referido anteriormente, não é considerado o tempo de propagação nos flip-flops.

Contador binário de 3 bit síncrono

FIGURE 9-15 A 3-bit synchronous binary counter.

Contador binário de 3 bit usa uma porta AND. Ao contrário do contador de 2 bit já descrito. A saída da porta AND produz as entradas J e K do flip-flop FF2.

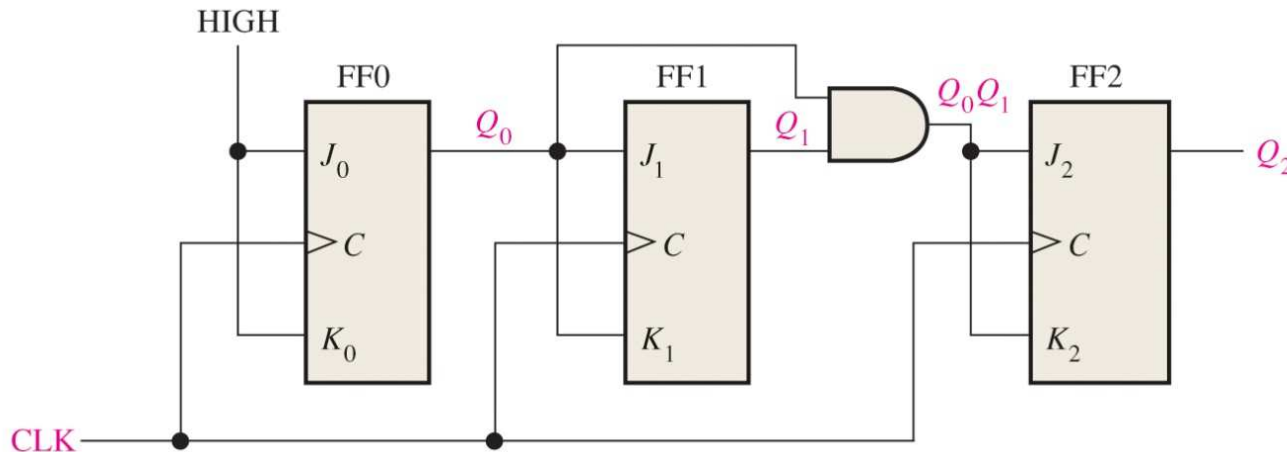
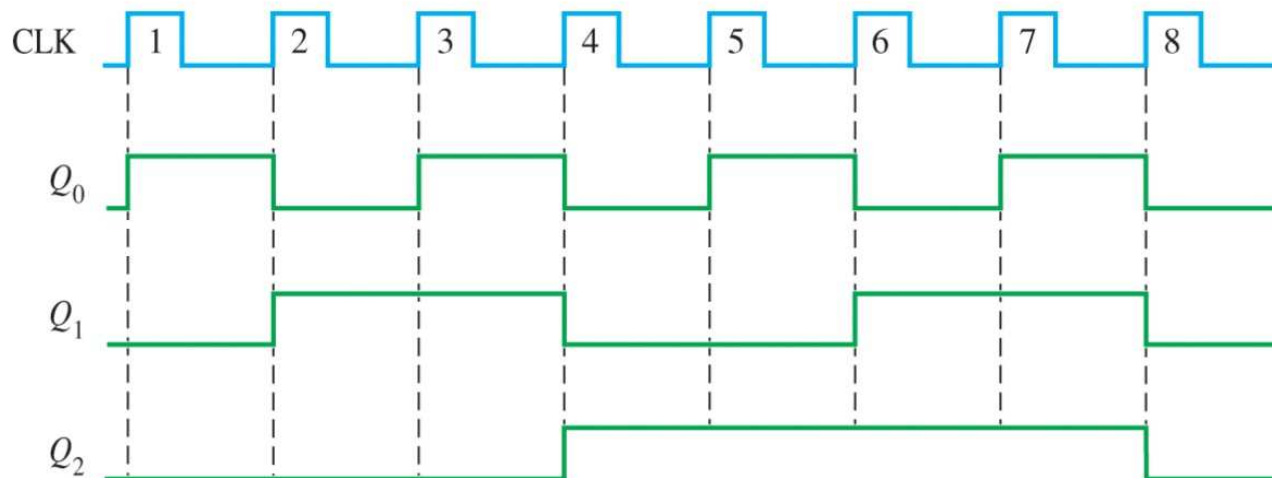


TABLE 9-3

State sequence for a 3-bit binary counter.

Clock Pulse	Q_2	Q_1	Q_0
Initially	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 (recycles)	0	0	0

FIGURE 9-16 Timing diagram for the counter of Figure 9-15.



Sumário da análise do contador J-K de 3 bits

TABLE 9–4 Summary of the analysis of the counter in Figure 9–15.

TABLE 9–4

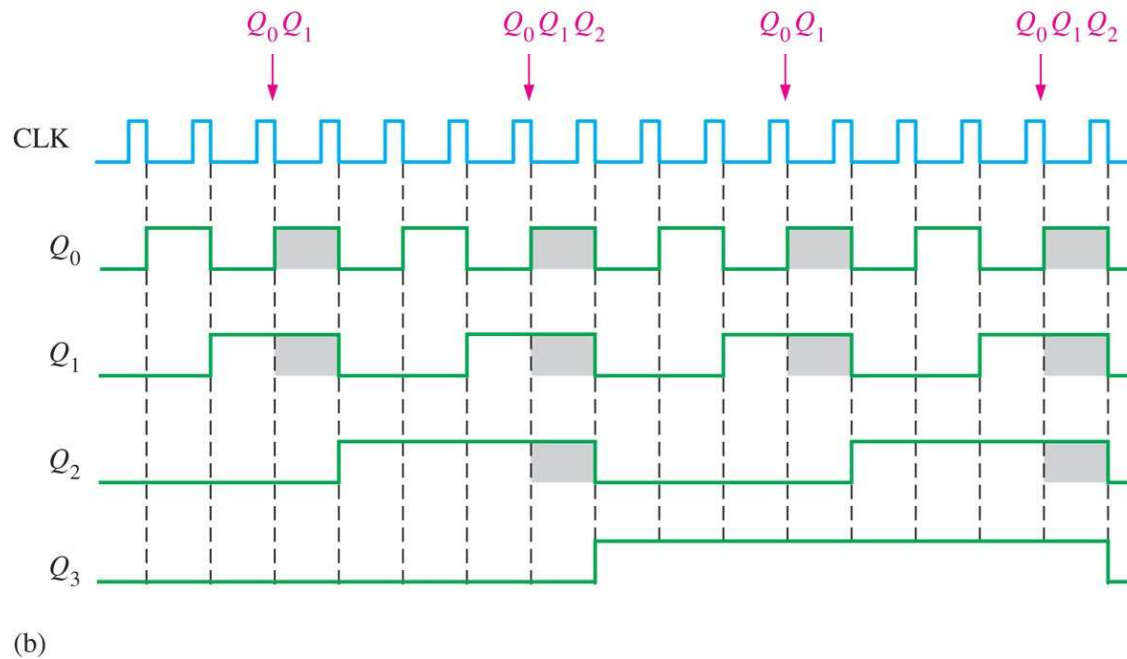
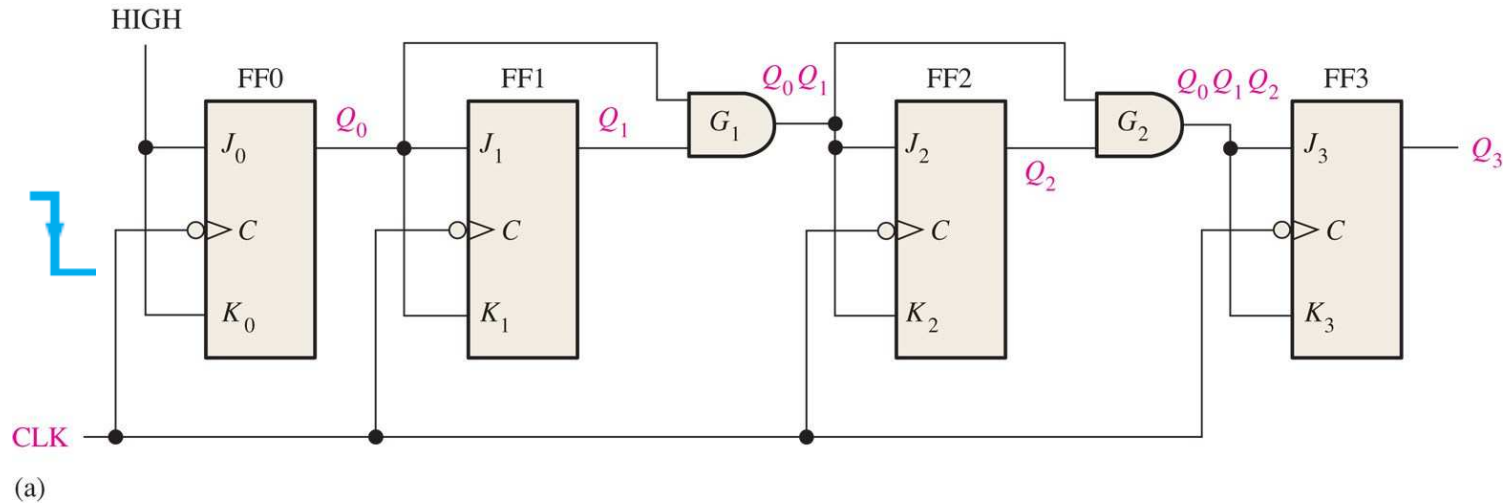
Summary of the analysis of the counter in Figure 9–15.

Clock Pulse	Outputs			J-K Inputs						At the Next Clock Pulse		
	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0	FF2	FF1	FF0
Initially	0	0	0	0	0	0	0	1	1	NC*	NC	Toggle
1	0	0	1	0	0	1	1	1	1	NC	Toggle	Toggle
2	0	1	0	0	0	0	0	1	1	NC	NC	Toggle
3	0	1	1	1	1	1	1	1	1	Toggle	Toggle	Toggle
4	1	0	0	0	0	0	0	1	1	NC	NC	Toggle
5	1	0	1	0	0	1	1	1	1	NC	Toggle	Toggle
6	1	1	0	0	0	0	0	1	1	NC	NC	Toggle
7	1	1	1	1	1	1	1	1	1	Toggle	Toggle	Toggle
Counter recycles back to 000.												

*NC indicates *No Change*.

Contador síncrono de 4 bit

FIGURE 9-17 A 4-bit synchronous binary counter and timing diagram. Times where the AND gate outputs are HIGH are indicated by the shaded areas.



Circuito contador síncrono de 4 bits

(relevante para o trabalho prático n.º 5)

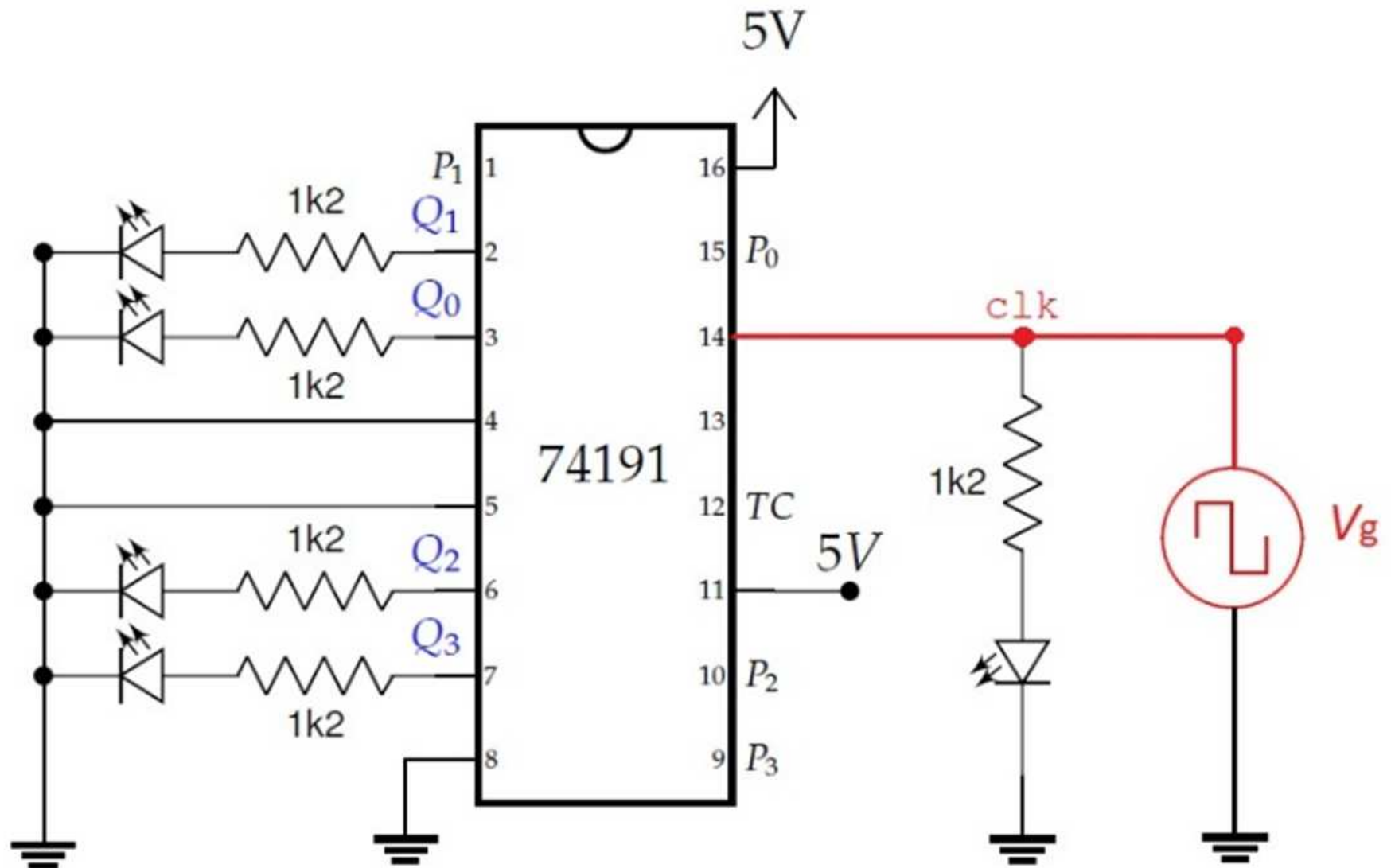
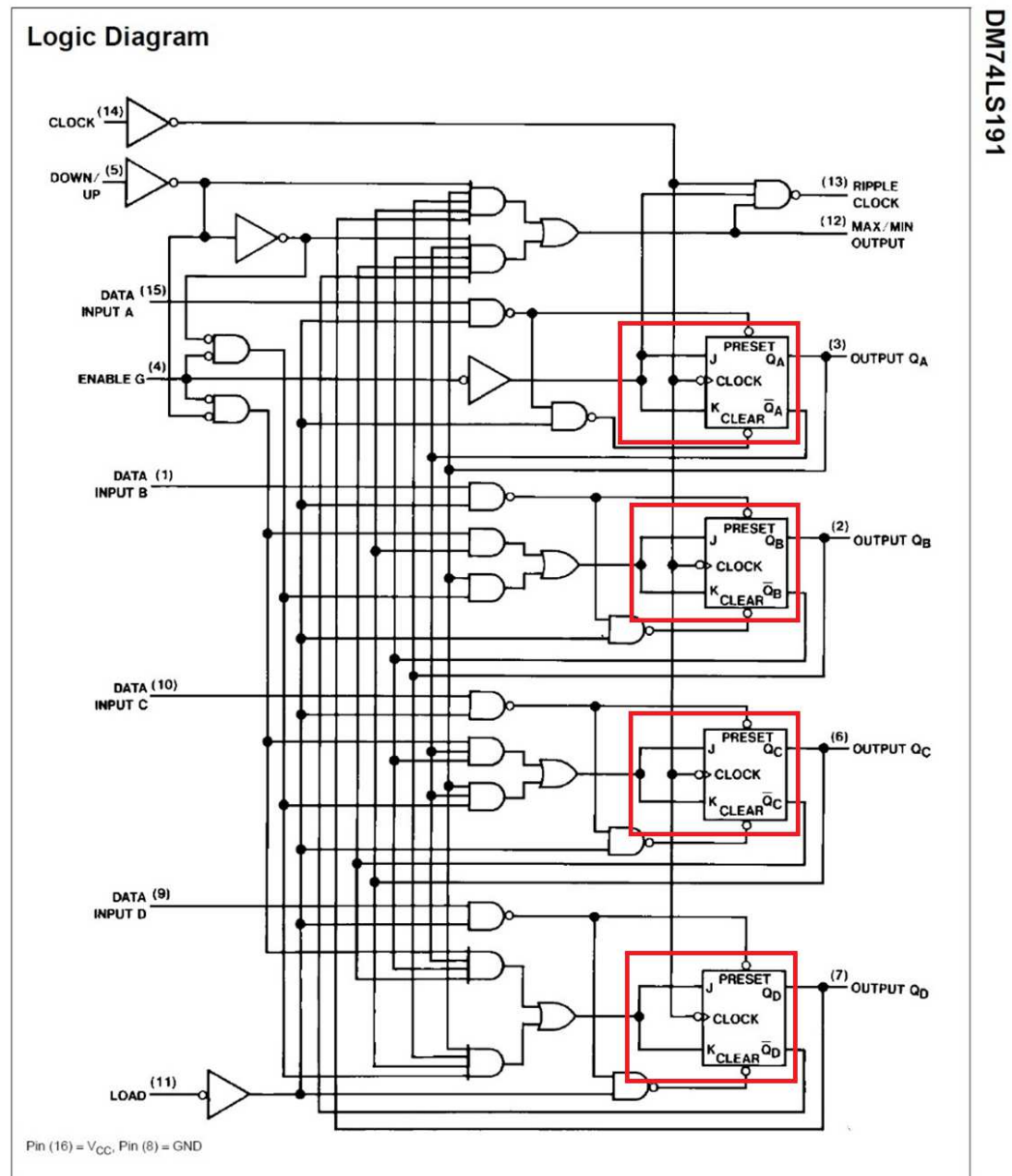
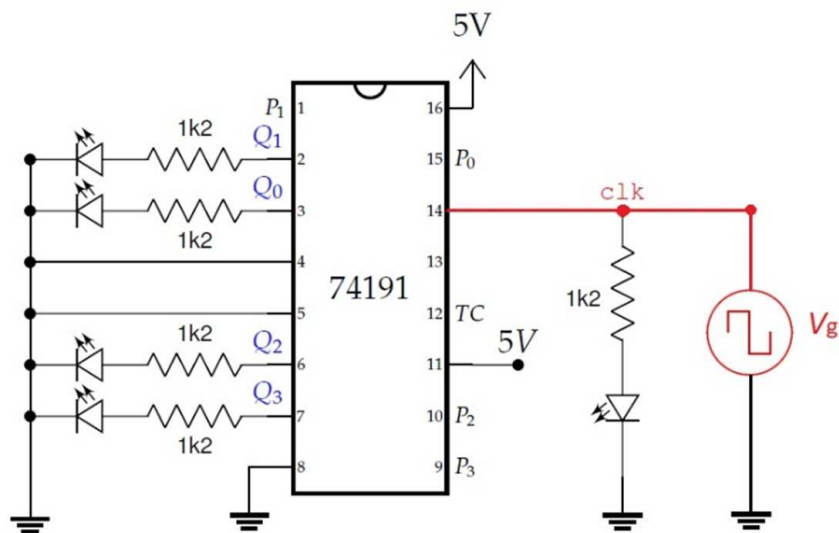


Diagrama lógico do contador 74LS191 de 4 bits



3

www.fairchildsemi.com

DM74LS191

Contador síncrono ascendente/descendente

FIGURE 9-22 A basic 3-bit up/down synchronous counter.

Um contador ascendente/descendente é capaz de progredir em qualquer direção, dependendo de uma entrada de controle.

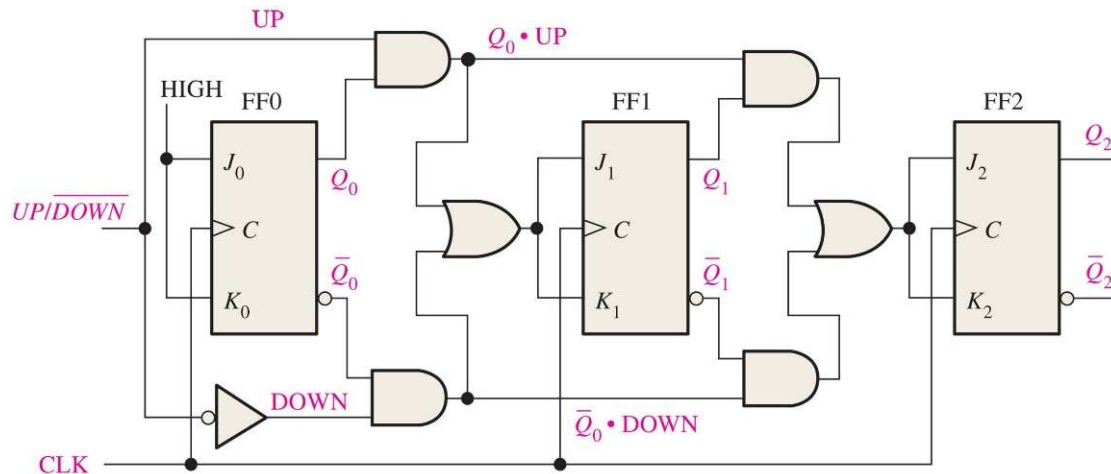


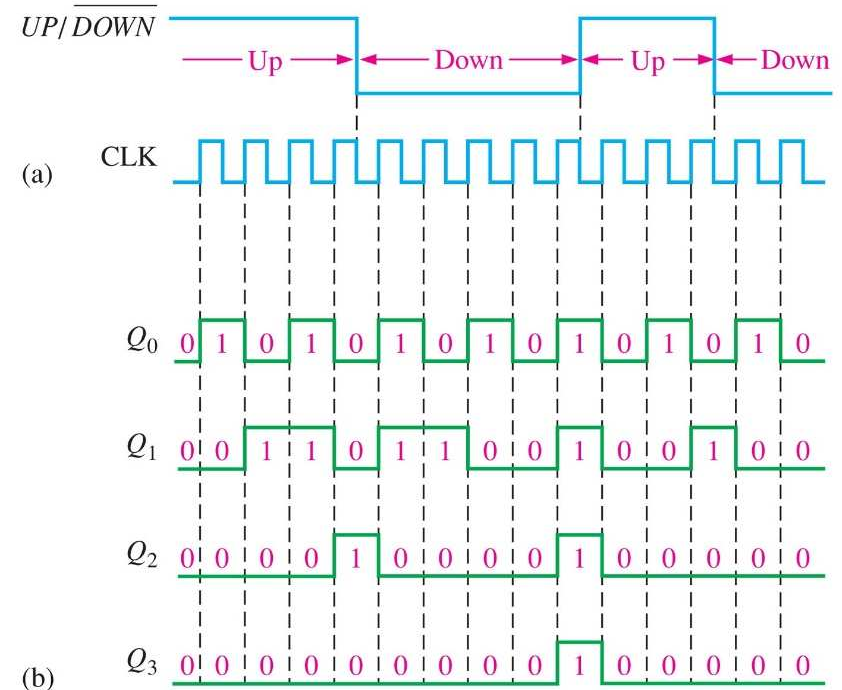
TABLE 9-6

Up/Down sequence for a 3-bit binary counter.

Clock Pulse	Up	Q_2	Q_1	Q_0	Down
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	↷

TABLE 9-7

Q_3	Q_2	Q_1	Q_0	
0	0	0	0	UP
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	0	1	1	DOWN
0	0	1	0	
0	0	0	1	
0	0	0	0	
1	1	1	1	
0	0	0	0	UP
0	0	0	1	
0	0	1	0	
0	0	0	1	
0	0	0	0	
0	0	0	0	DOWN
0	0	0	1	
0	0	0	0	
0	0	0	0	
0	0	0	0	



Contador síncrono de década BCD

FIGURE 9-18 A synchronous BCD decade counter.

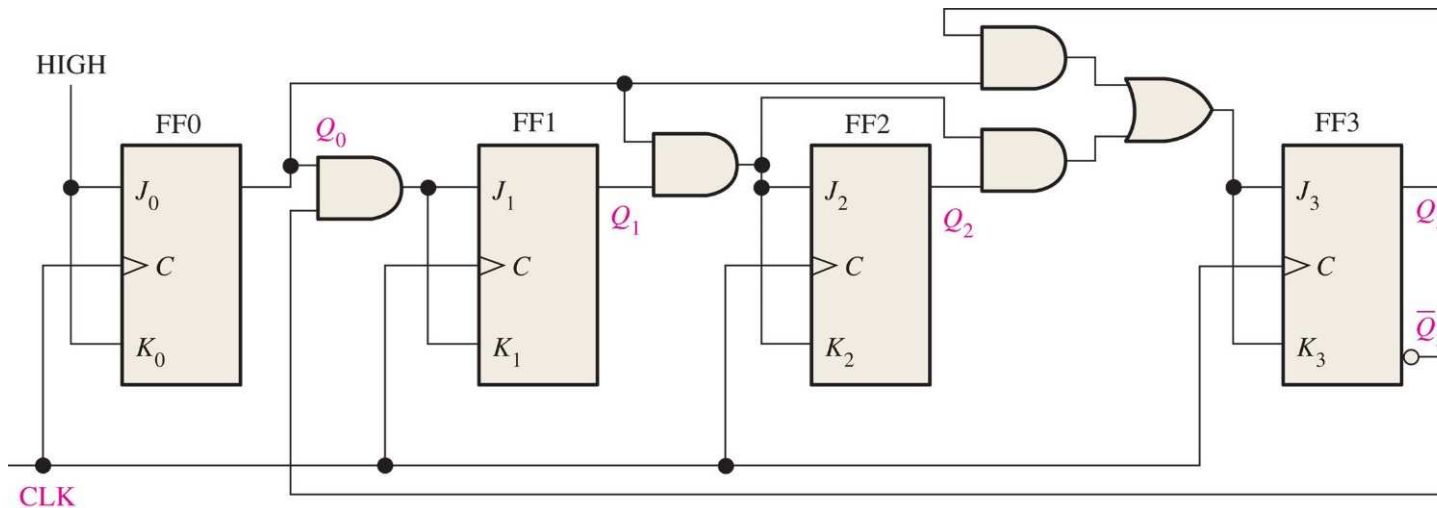


FIGURE 9-19 Timing diagram for the BCD decade counter (Q_0 is the LSB).

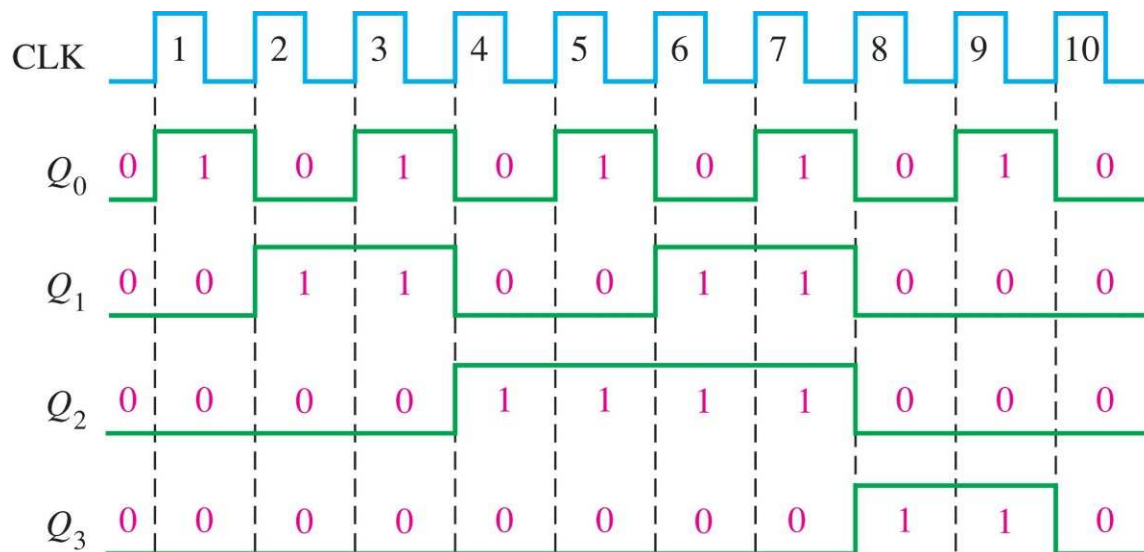
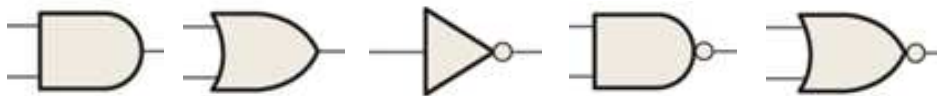


TABLE 9-5

States of a BCD decade counter.

Clock Pulse	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10 (recycles)	0	0	0	0

Contadores assíncronos



Contador assíncrono de 2 bits

Um contador é assíncrono se os respectivos flip-flops não forem ativadas em simultâneo, isto é, não forem ativadas pelo mesmo sinal.

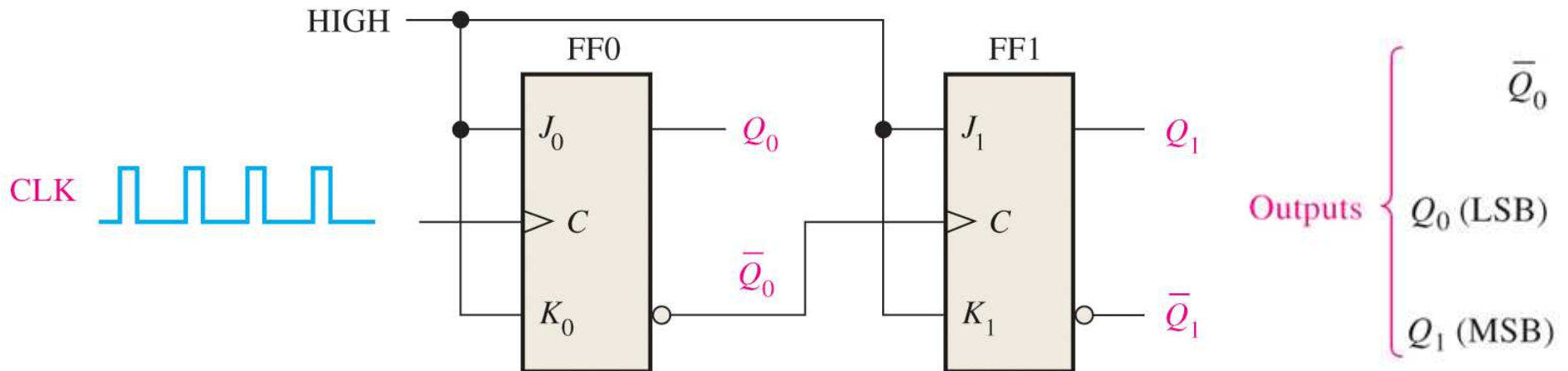
Um contador assíncrono é também um divisor de frequência.

Num contador assíncrono, o relógio é aplicado apenas ao primeiro “andar” (primeiro FF). Os andares (FFs) subsequentes usam como “sinal de relógio” o sinal gerado pelo andar imediatamente anterior (o sinal Q ou o sinal Não-Q).

O esquema abaixo representa um contador assíncrono de 2 bit implementado com **flip-flops T** ($Q_{n+1} = T \cdot \overline{Q_n} + \overline{T} \cdot Q_n$), com disparo pela borda positiva.

(um contador de 2 bit só conta (até) 4 sequências: 00 (0), 01 (1), 10 (2), e 11 (3), ao fim da sequência volta a contar 00 (0), 01 (1), 10 (2), ...).

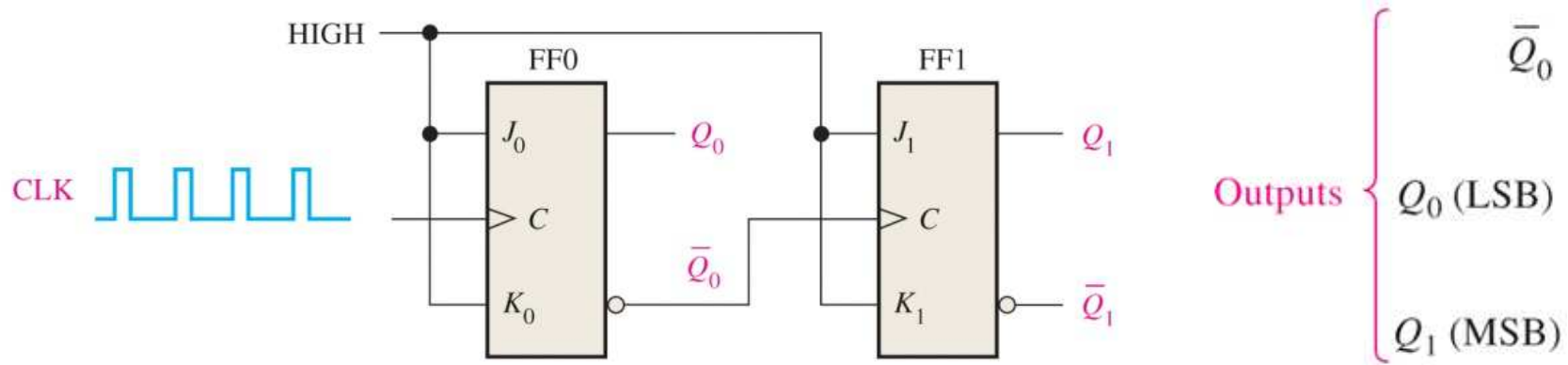
A saída do contador é representada por Q_1Q_0 .



As saídas Q₀ e Q₁ indicam o número de pulsos de relógio recebidos: Q_1Q_0 .

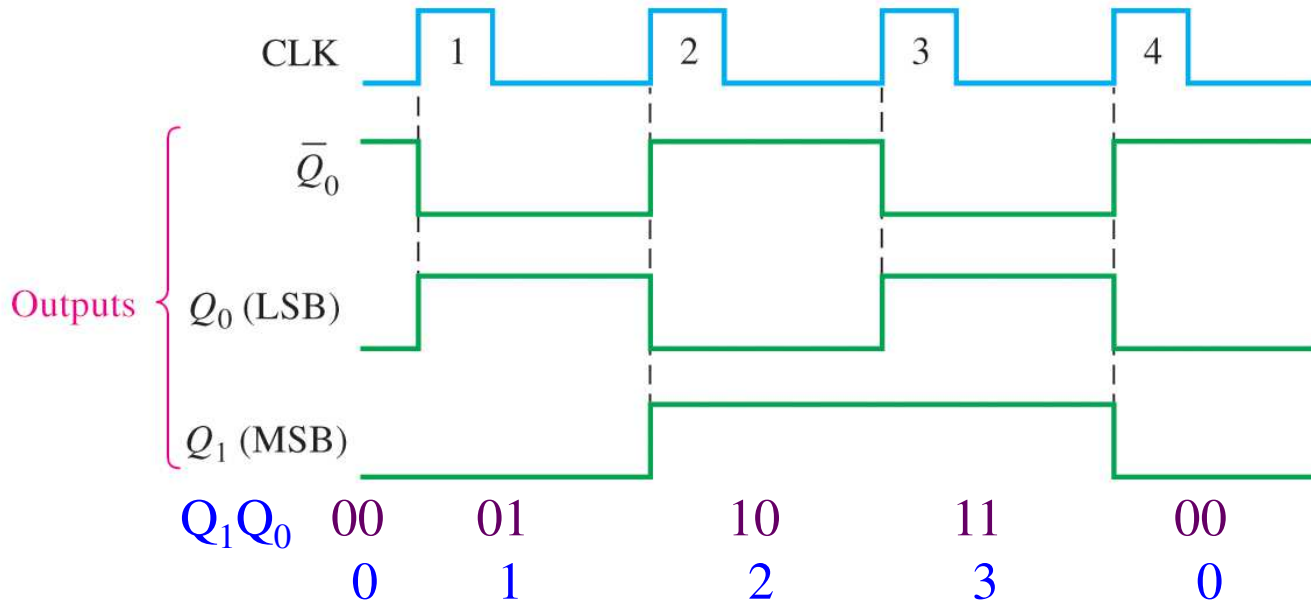
Contador assíncrono de 2 bits

Contador assíncrono de 2 bit implementado com flip-flops T ($Q_{n+1} = T \cdot \overline{Q_n} + \overline{T} \cdot Q_n$), disparados pela transição positiva.



A saída do contador é representada por Q_1Q_0 .

Diagrama de tempo do contador assíncrono de 2 bit (Q_1Q_0) implementado com flip-flops T.

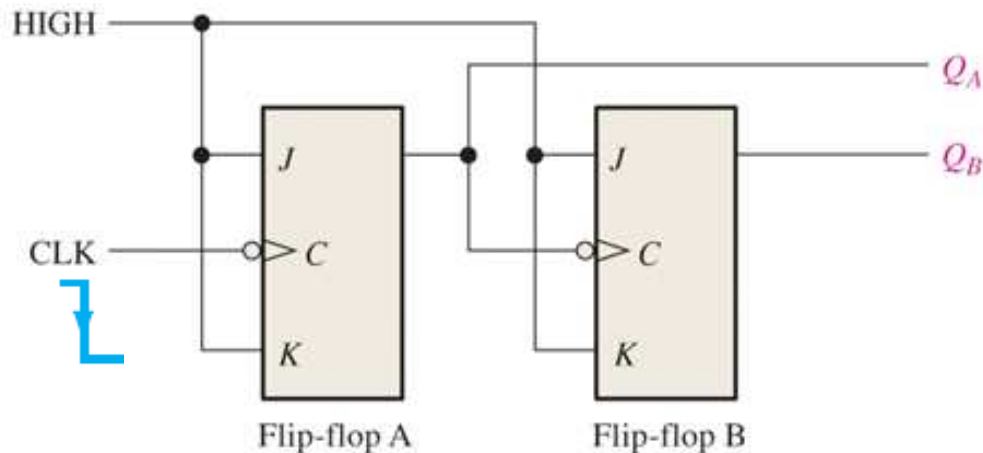


As saídas Q_0 e Q_1 indicam (em binário) o número de pulsos de relógio recebidos: Q_1Q_0 .

(binário)
(decimal)

Descrição do contador assíncrono de 2 bits

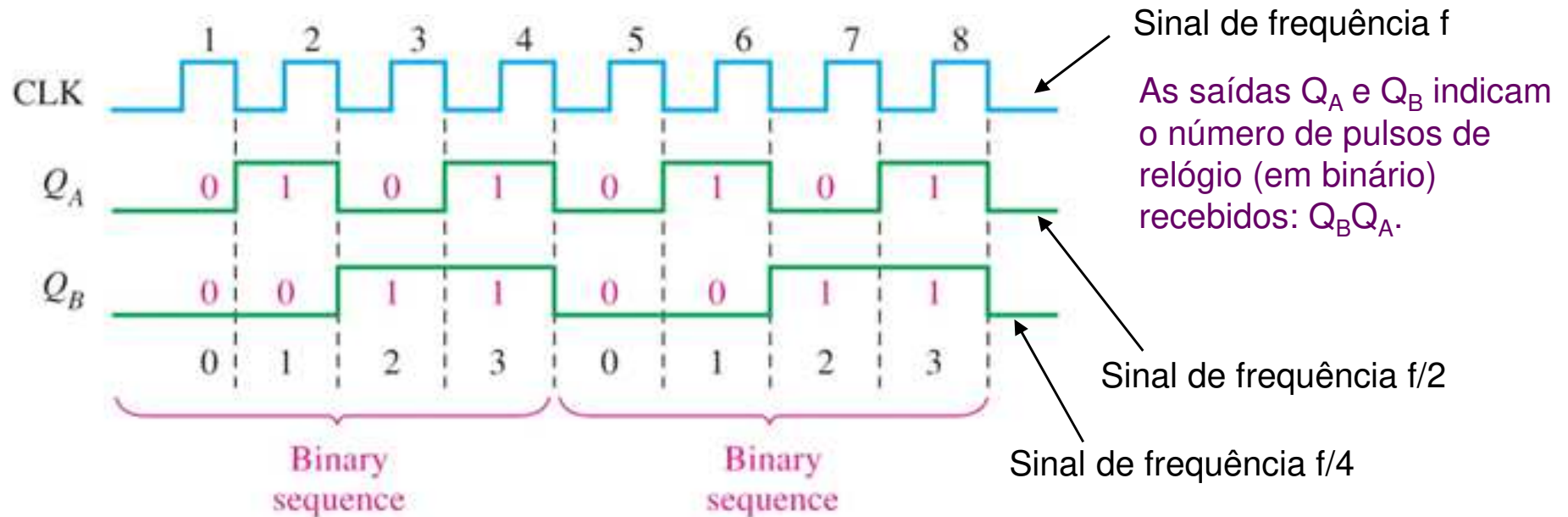
No exemplo que se segue os flip-flops são do tipo T ($Q_{n+1} = T \cdot \overline{Q_n} + \overline{T} \cdot Q_n$), **disparados pela borda negativa**. Os dois flip-flops estão inicialmente *resetados*.



O flip-flop A comuta na transição negativa de cada pulso de relógio. A saída Q_A do flip-flop A é o relógio do flip-flop B, assim cada vez que Q_A faz uma transição de nível ALTO para nível BAIXO, o flip-flop B muda de estado (flip-flop T).

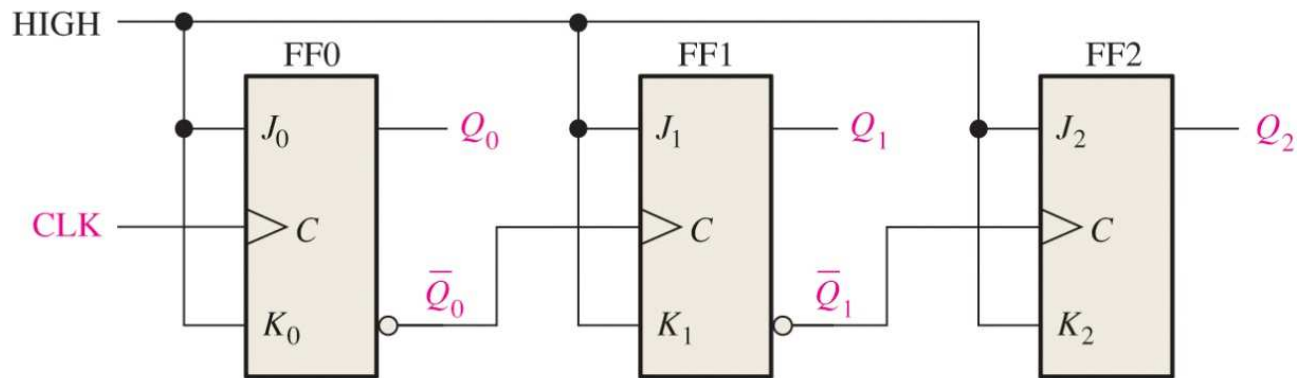
As formas de onda resultante de Q_A e Q_B são mostradas na figura.

Um contador assíncrono é também um divisor de frequência.

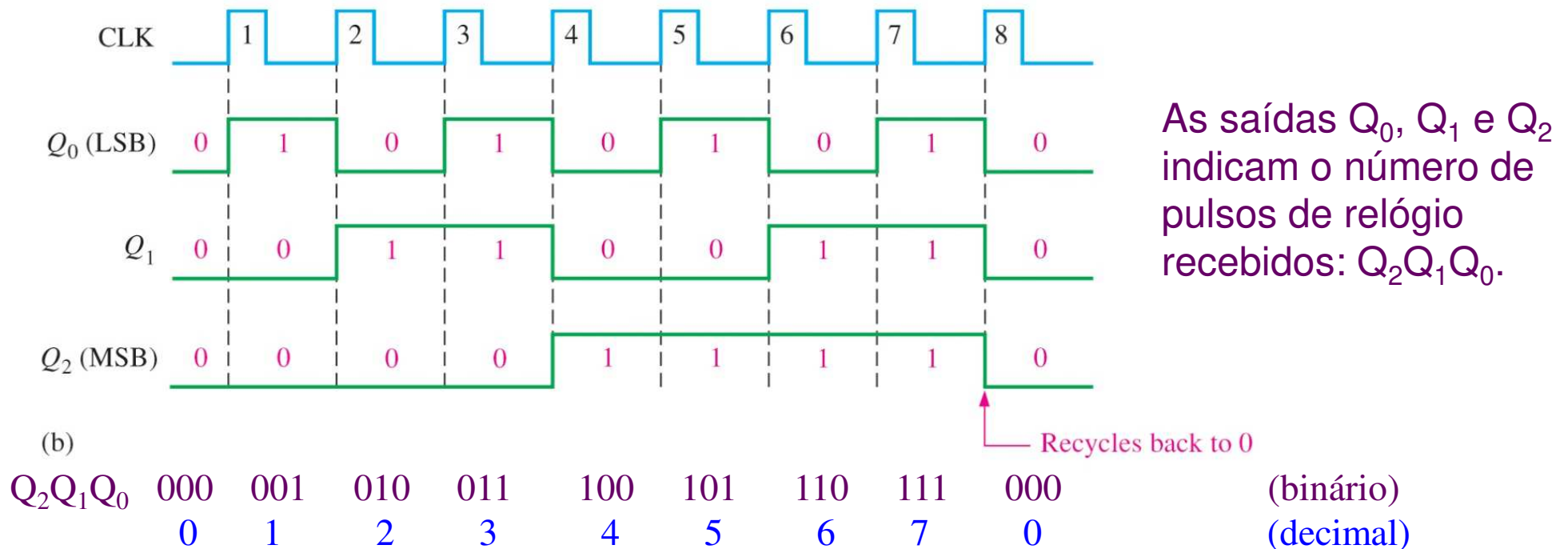


Contador assíncrono de 3 bits com flip-flops J-K/T

(conta até 8, de 0 a 7)

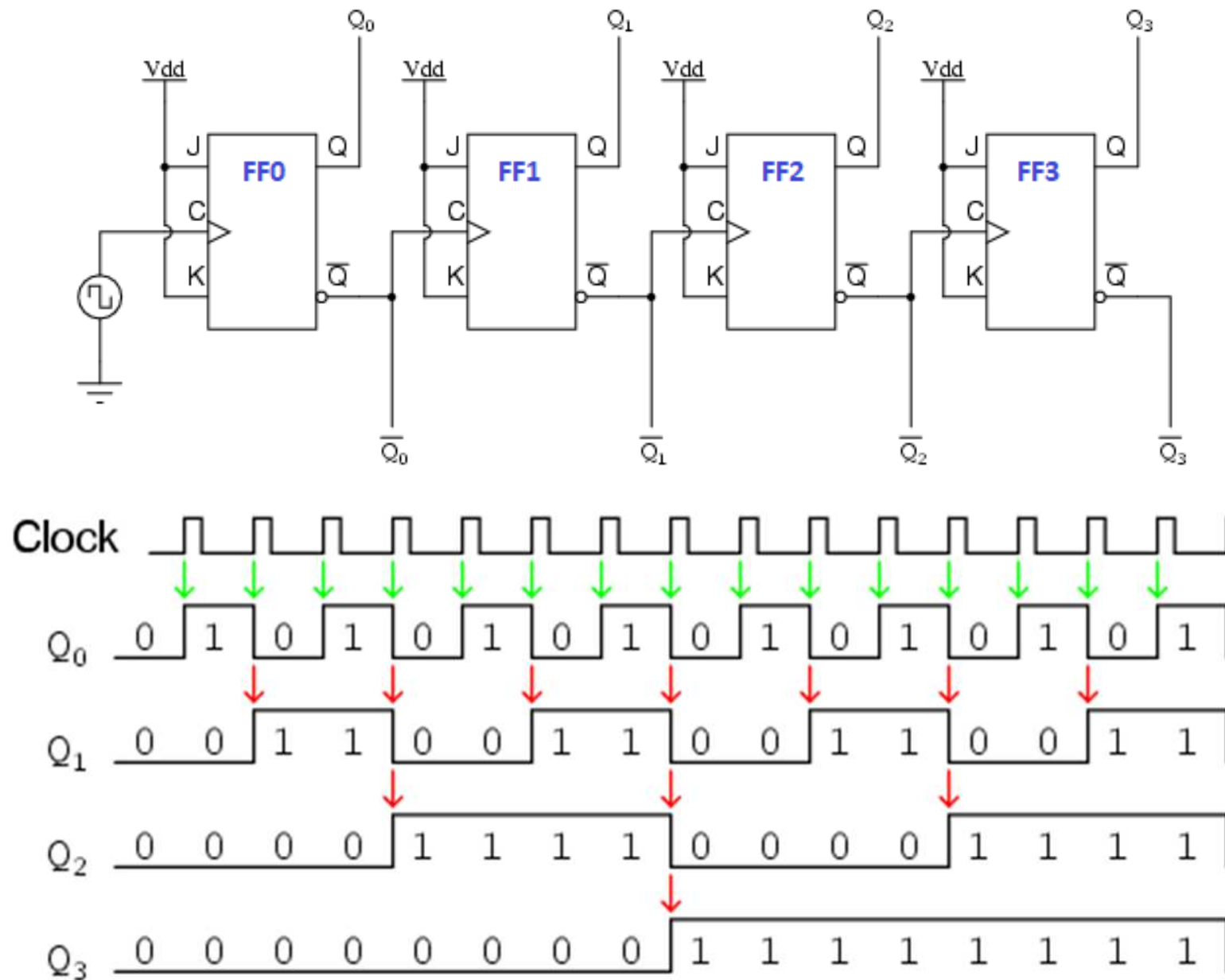


(a) flip-flops T ($Q_{n+1} = T \cdot \overline{Q_n} + \overline{T} \cdot Q_n$). A saída do contador é representada por $Q_2Q_1Q_0$.



Contador assíncrono de 4 bits com flip-flops J-K

A simultaneous "up" and "down" counter



Contador assíncrono de 2 bits com flip-flops D

FIGURE 9-4 A 2-bit asynchronous binary counter.

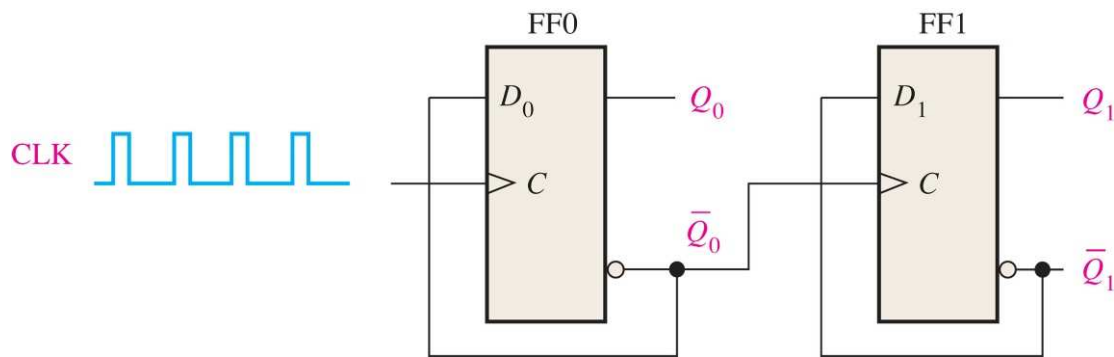
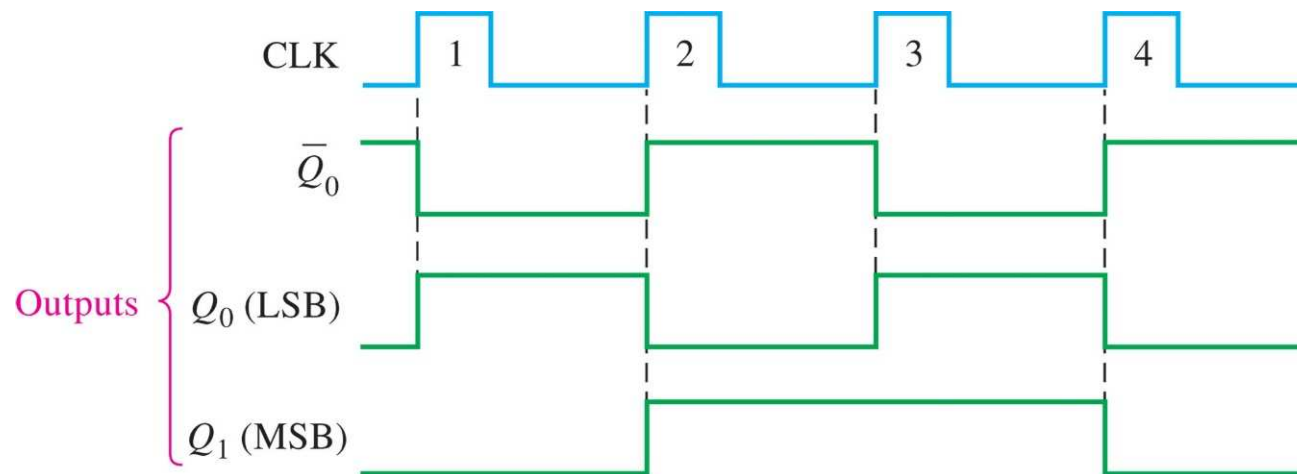


TABLE 9-1

Binary state sequence for the counter in Figure 9-4.

Clock Pulse	Q_1	Q_0
Initially	0	0
1	0	1
2	1	0
3	1	1
4 (recycles)	0	0

FIGURE 9-5 Timing diagram for the counter of Figure 9-4. As in previous chapters, output waveforms are shown in green.



Contador assíncrono de 3 bits com flip-flops D

FIGURE 9-6 Three-bit asynchronous binary counter and its timing diagram for one cycle.

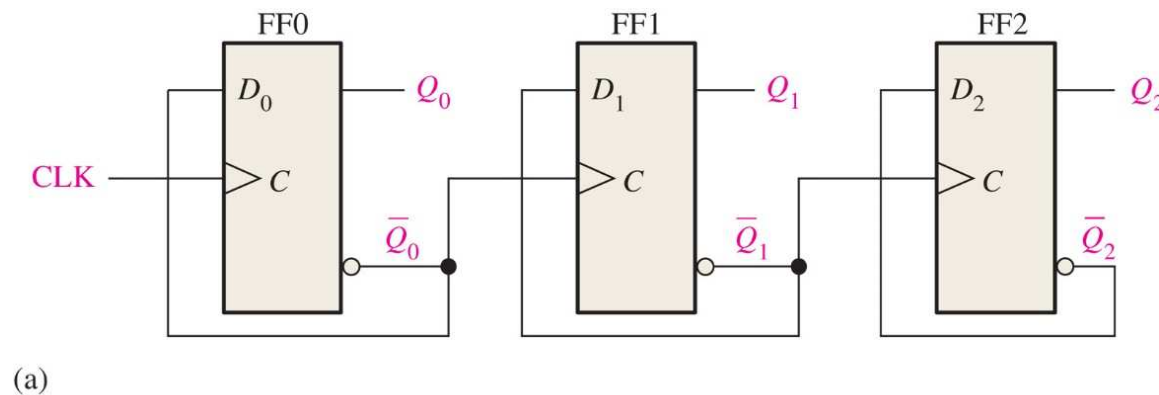
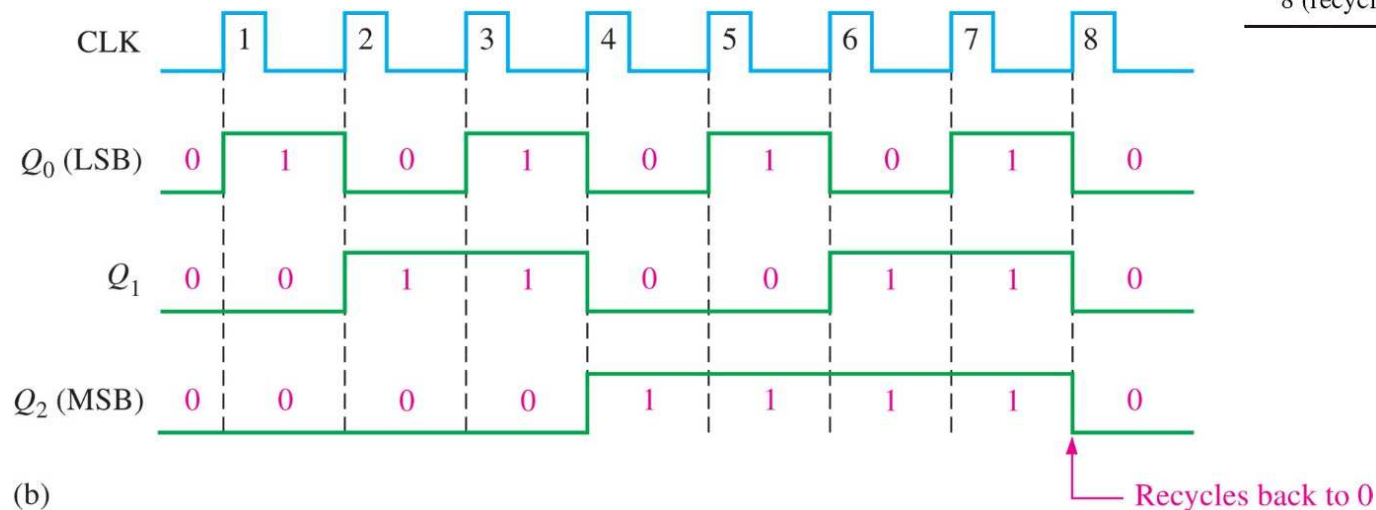


TABLE 9-2

State sequence for a 3-bit binary counter.

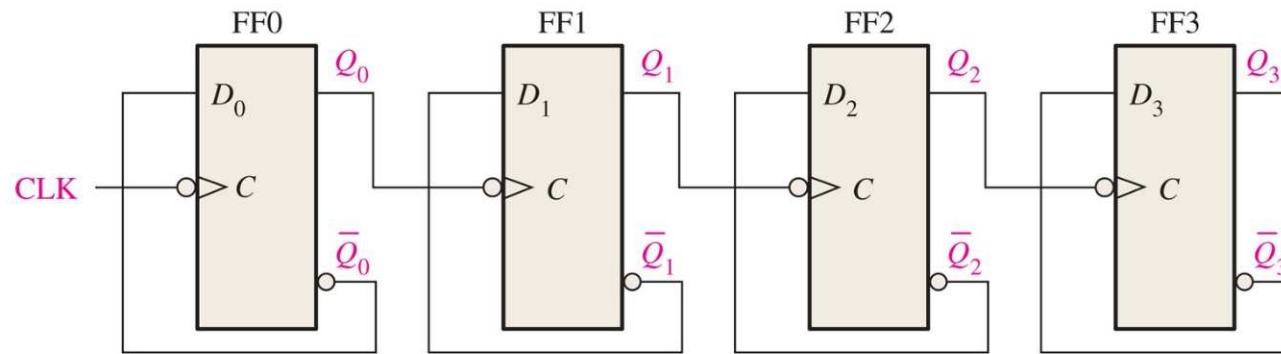
Clock Pulse	Q_2	Q_1	Q_0
Initially	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8 (recycles)	0	0	0



A saída do contador é representada por $Q_2Q_1Q_0$.

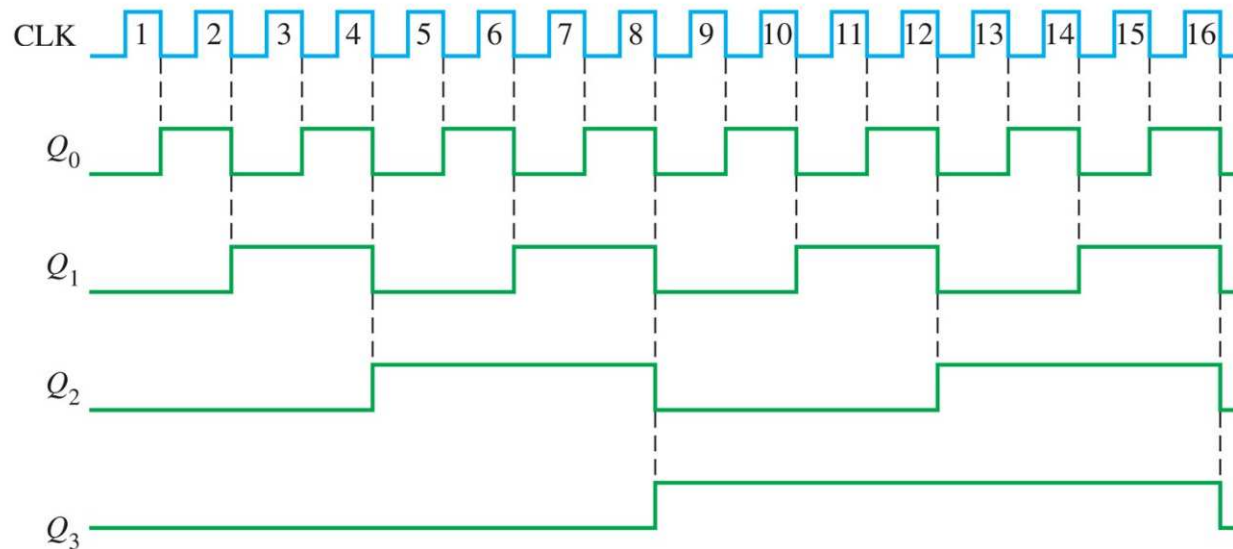
Contador assíncrono de 4 bits

FIGURE 9-8 Four-bit asynchronous binary counter and its timing diagram



(a)

A saída do contador é representada por $Q_3Q_2Q_1Q_0$.



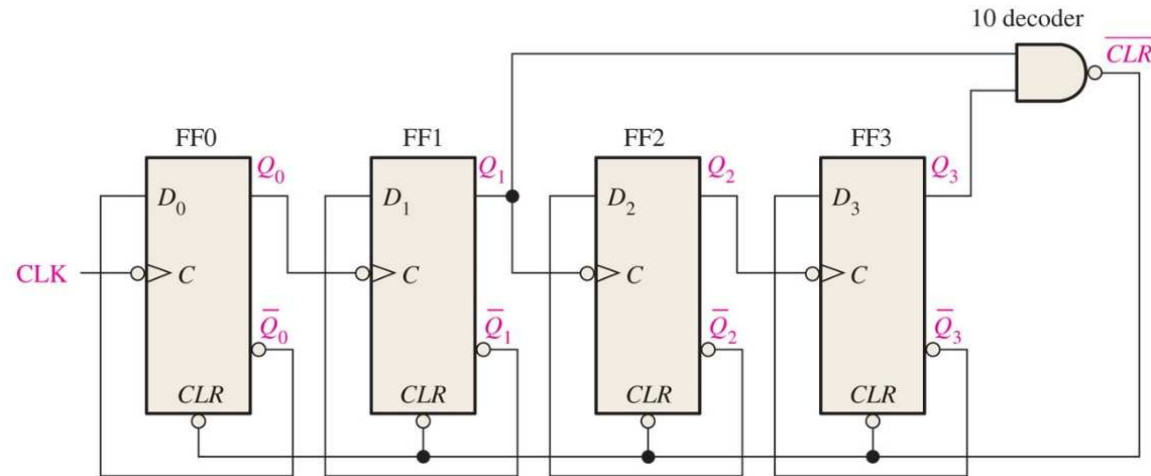
(b)

Flip-flops T: $Q_{n+1} = T \cdot \bar{Q}_n + \bar{T} \cdot Q_n$

Flip-flops D: $Q_{n+1} = D$

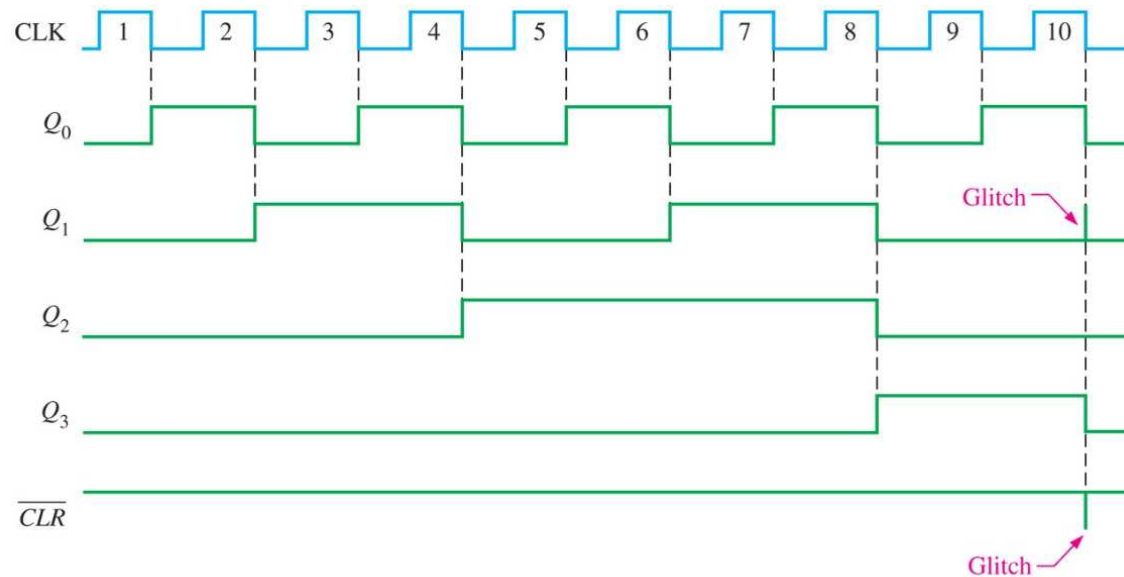
Contador assíncrono de década

FIGURE 9-9 An asynchronously clocked decade counter with asynchronous recycling.



(a)

A saída do contador é representada por $Q_3Q_2Q_1Q_0$.



(b)

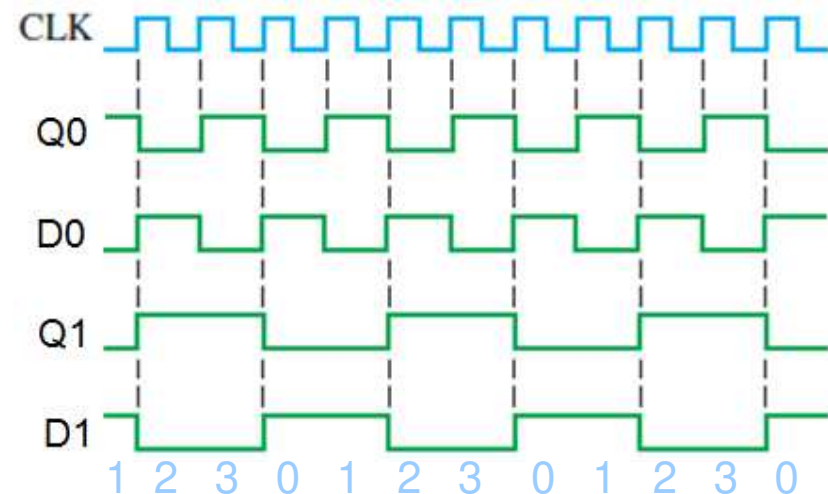
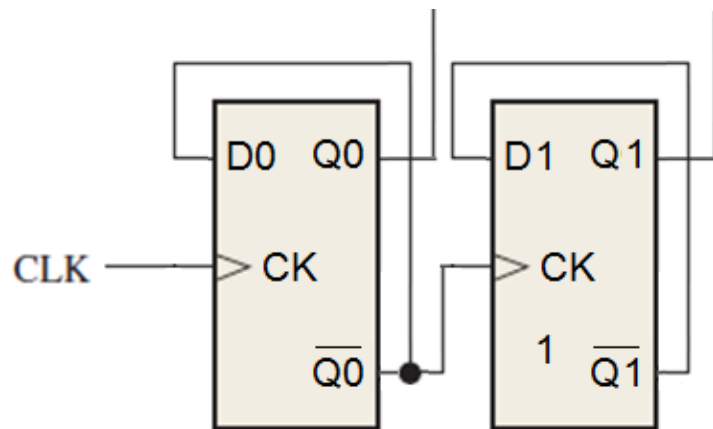
Contadores de Módulo

O **módulo de um contador** corresponde ao número de estados que o contador percorre (“conta”) em um **ciclo completo de contagem**. Isto é, o módulo de um contador é um valor **x** que o contador pode contar. O valor máximo da contagem em decimal é $= 2^N - 1$, onde N representa o número de flip-flops ou bits de contagem.

Exemplos: um contador de módulo igual a 4 pode contar até 4 (0-3 decimal: 0, 1, 2 e 3); um contador de módulo 8 pode contar até 8 (0 - 7 decimal).

Ter presente que um contador também faz divisão de frequência: O módulo do contador corresponde ao valor máximo da divisão em frequência que se pode obter.

O diagrama abaixo corresponde a um contador **assíncrono de 2 bits**, formado **por 2 flip-flop do tipo D** ativados à transição positiva do sinal de relógio.



Este contador assíncrono de 2 bits permite contar até 4 (0, 1, 2 e 3, em decimal). Portanto é um **contador de módulo 4**. (Este contador atua também como um divisor de frequência por 2 e como um divisor de frequência por 4).

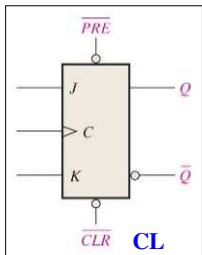
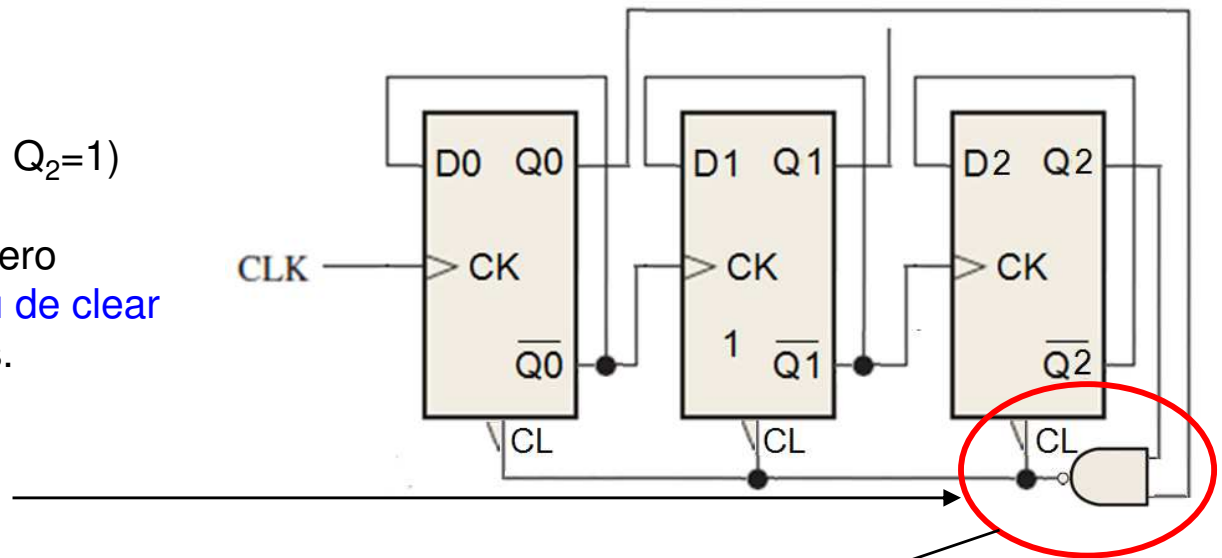
Contador de Módulo 6

O módulo de um contador é um valor x que o contador pode contar. Exemplo: um contador de módulo igual a 6 pode contar até 6 [000 (zero) ... 101 (cinco)].

O diagrama abaixo corresponde a um **contador assíncrono de 3 bit**, formado por 3 flip-flop do tipo D ativados à transição positiva do sinal de relógio.

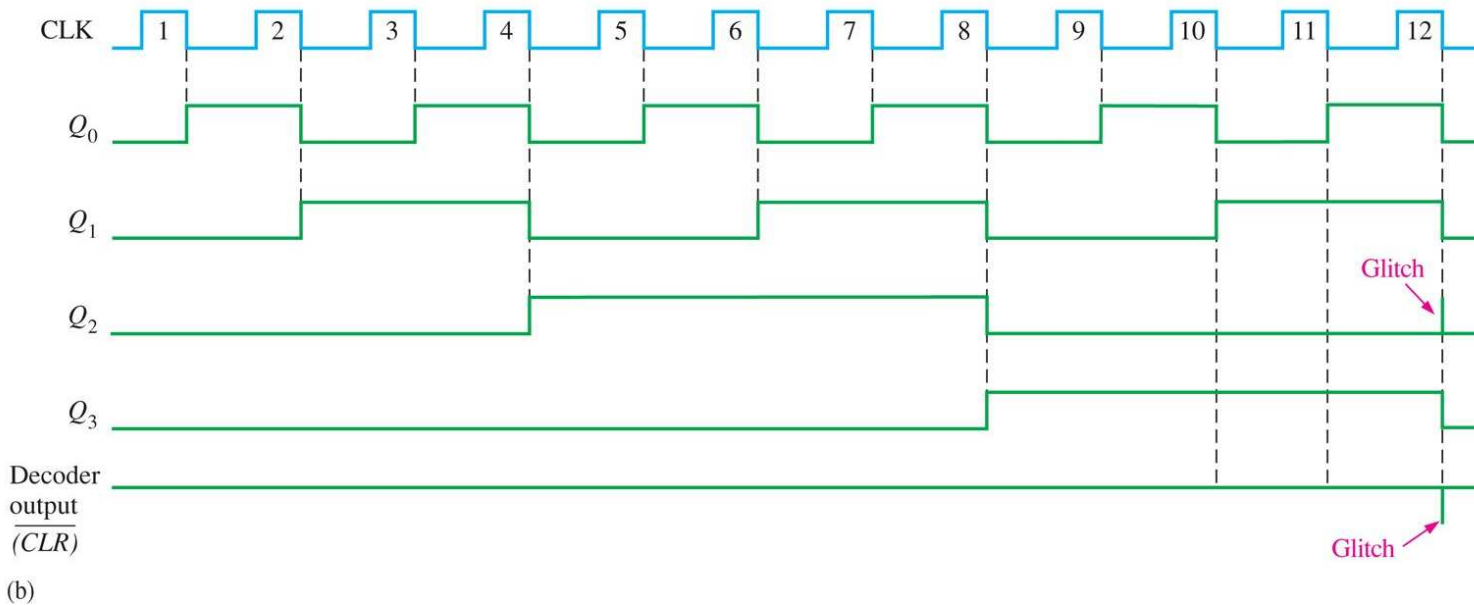
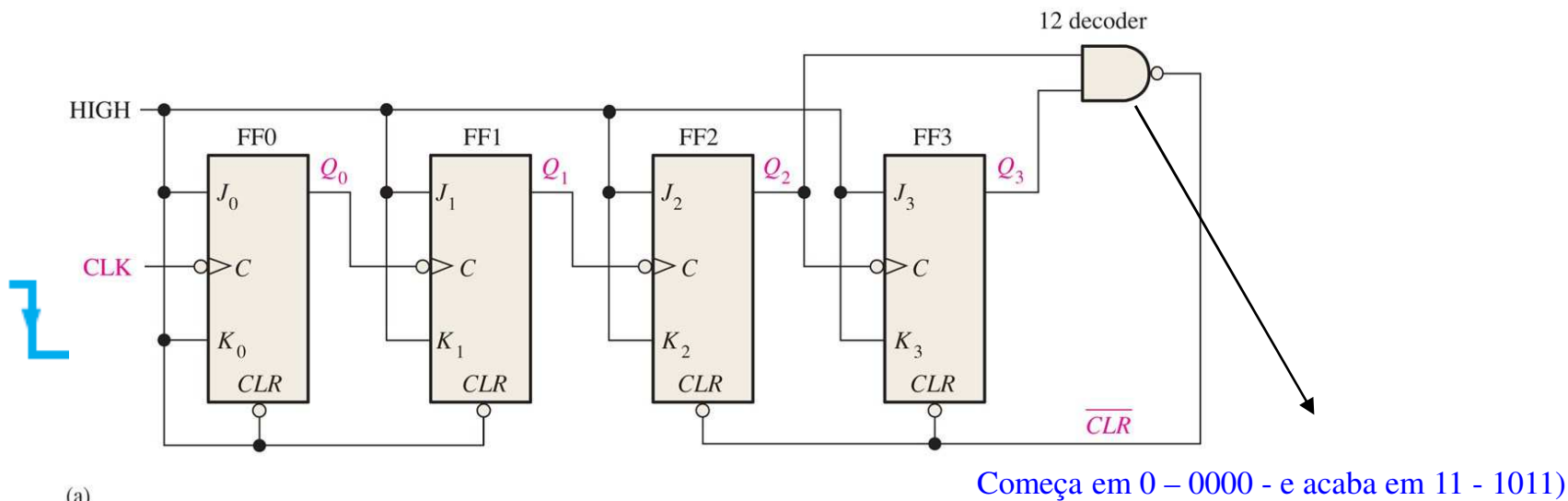
Este contador poderia contar até 8. Contudo, pretende-se que ele conta apenas até 6. Neste caso a contagem é reinicializada à 7ª sequência com ajuda da porta Não-E, por forma a que contagem termine após se obterem as primeiras 6 sequências.

A sequência '101' (quando $Q_0=1$, $Q_1=0$, e $Q_2=1$) permite reiniciar a contagem, e, portanto, **inicializar** o respectivo flip-flop com valor zero (clear, CL='0'). O sinal de **inicialização ou de clear** – é aplicado às entradas CL dos flip-flops.


$$Q_2Q_1Q_0 \rightarrow 000 \rightarrow 001 \rightarrow 010 \rightarrow 011 \rightarrow 100 \rightarrow 101 \rightarrow 000$$

Contador assíncrono de módulo 12

FIGURE 9-10 Asynchronously clocked modulus-12 counter with asynchronous recycling.



Divisores de frequência por 2 e por 4

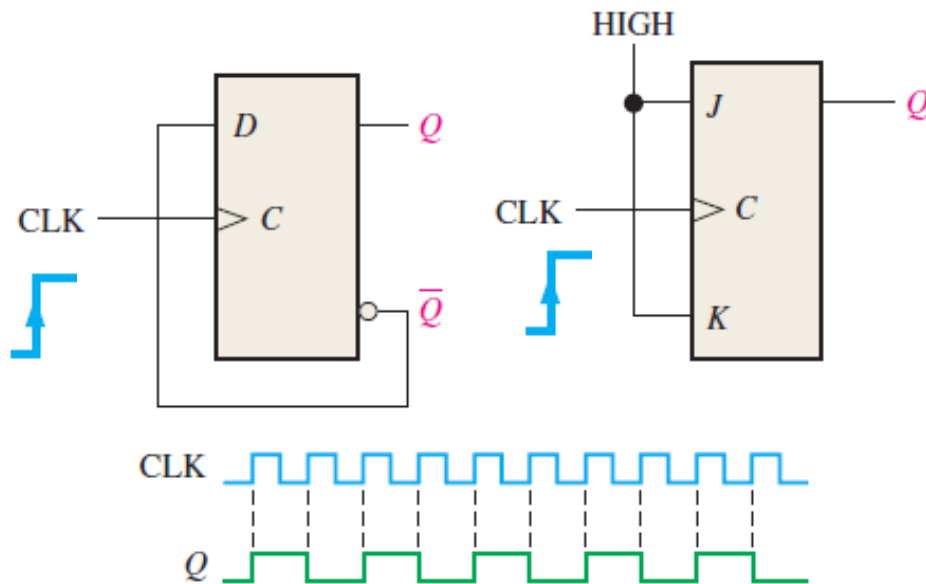
Os contadores assíncronos abaixo permitem contar até 1 (decimal, 0 e 1) e até 3 (em decimal, 0, 1, 2 e 3), e também são designados por **divisores de frequência por 2 e por 4**.

O diagrama da esquerda corresponde a contadores **assíncronos de 1 bits**, formados por **flip-flop do tipo D e flip-flops JK** ativados à transição positiva do sinal de relógio.

Flip-flops T: $Q_{n+1} = T \cdot \overline{Q_n} + \overline{T} \cdot Q_n$

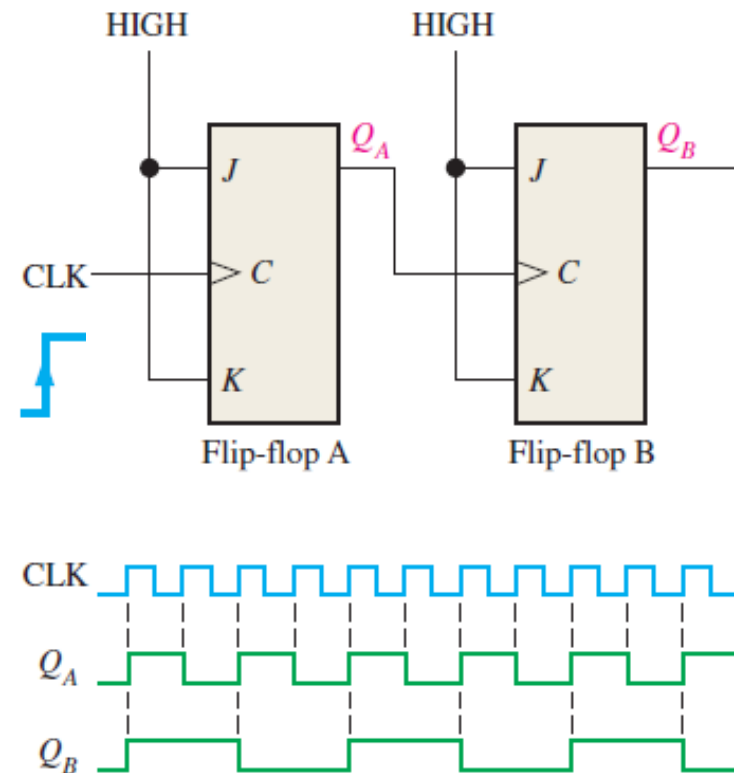
Flip-flops D: $Q_{n+1} = D$

Divisores de frequência por 2 usam 1 flip-flop



Os flip-flops D e J-K estão ligados para funcionarem como divisores por 2.

Divisores de frequência por 4 usam 2 flip-flops



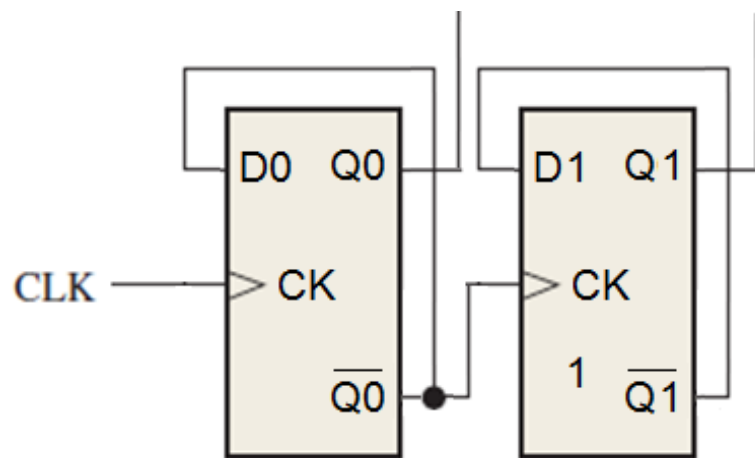
Os flip-flops J-K estão colocados em cascata para formarem um divisor por 4.

Divisor de Frequência por 4 com flip-flops D

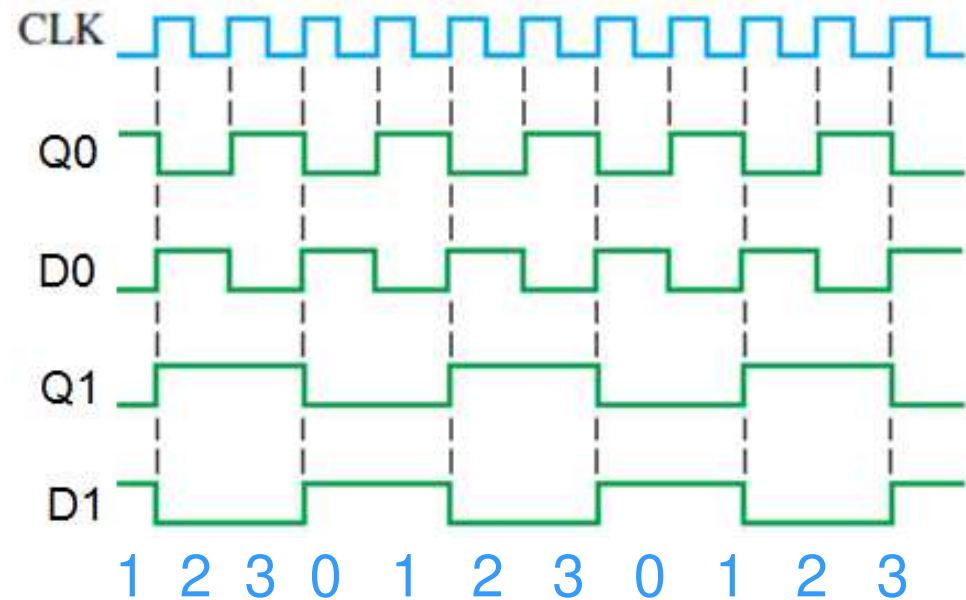
Um contador assíncrono pode funcionar também como divisor de frequência. Os contadores assíncronos abaixo permitem contar até 1 (decimal, 0 e 1) e até 3 (em decimal, 0, 1, 2 e 3), e também são designados por **divisores de frequência por 2 e por 4**.

O diagrama abaixo corresponde a um contador **assíncrono de 2 bits**, formado **por 2 flip-flop do tipo D** ativados à transição positiva do sinal de relógio. É também indicado o respetivo diagrama temporal.

Fip-flops D: $Q_{n+1} = D$

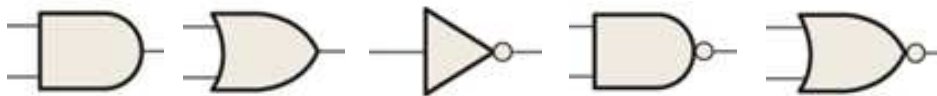


$Q_1Q_0 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00 \rightarrow 01$



Este contador assíncrono de 2 bits permite contar até 3 (em decimal) e é também um **divisor em frequência por 2 e por 4**: a saída de Q_0 corresponde a um sinal com o **dobro** do período do sinal de relógio CK; a saída Q_1 corresponde a um sinal cujo período é o **quadruplo** do período do sinal de relógio CK.

Aspetos práticos a ter em conta na habilitação por transição de borda positiva/negativa



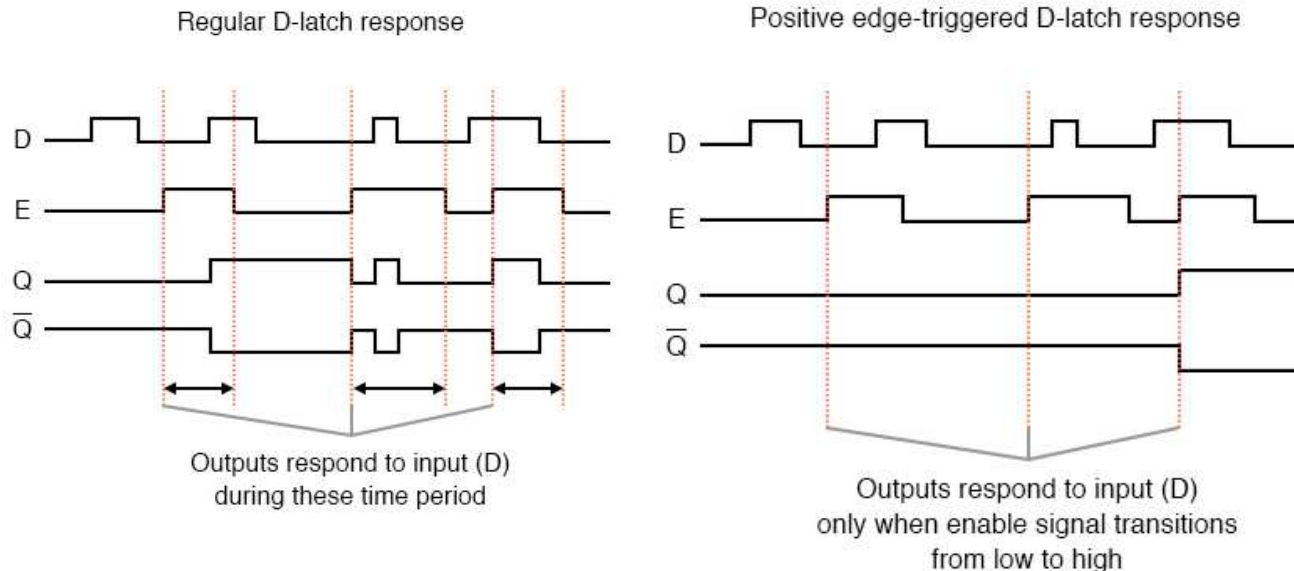
Habilitação por transição de borda positiva/negativa

Até agora, vimos os circuitos latch S-R, D e J-K com entradas de habilitação, que respondem às entradas de dados (S-R, D e J e K) apenas quando a entrada de habilitação é ativada.

No entanto, em muitas aplicações digitais é desejável limitar a resposta de um circuito latch a **um período de tempo muito curto**, em vez de toda a duração total em que a entrada de habilitação está ativada.

Um método de habilitar um circuito multivibrador é chamado de **disparo de borda**, onde as entradas de dados do circuito **são “lidas” apenas durante o tempo em que a entrada de habilitação está em transição de um estado para outro (de 0 para 1 ou vice versa)**.

Nas figuras abaixo comparam-se os diagramas de tempo para uma latch D normal com uma latch D que é acionada por borda:



No segundo diagrama de temporal, notamos uma resposta distintamente diferente na (s) saída (s) do circuito: este só responde à entrada D durante aquele breve momento em que o sinal de habilitação muda (**nas transições**) de baixo para alto. Isto é conhecido/designado como **disparo de borda positiva**.

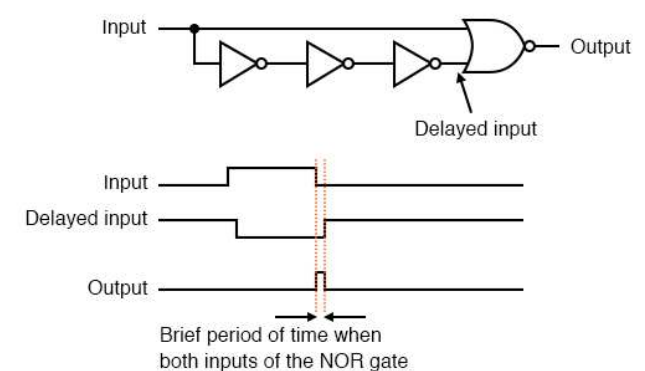
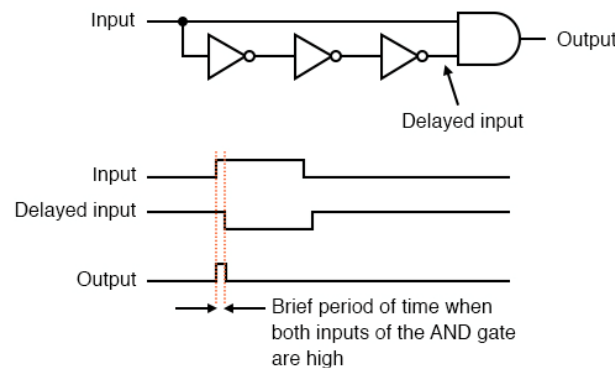
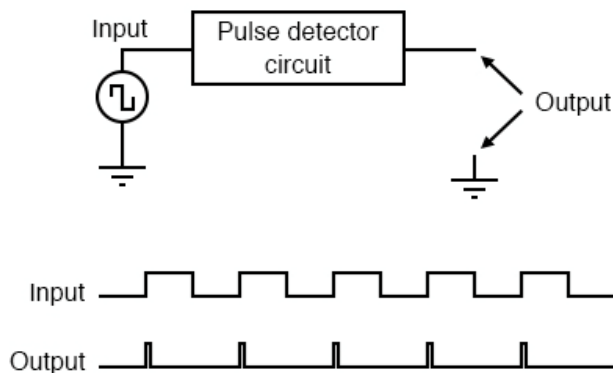
Sempre que habilitamos um circuito multivibrador na borda de transição de um sinal de habilitação de onda quadrada, **o latch passa a designa-se flip-flop, em vez de latch**.

Latches e flip-flops: detetor de pico

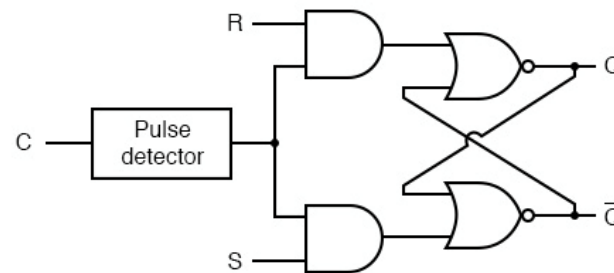
Se para criar uma latch S-R habilitada a partir de uma latch S-R regular é suficiente usarmos algumas portas AND, como é que se implementa a lógica que só presta atenção à borda ascendente ou descendente de um sinal de relógio digital?

Como podemos realizar esse disparo/transição de borda?

Precisamos de um circuito digital que produza um pulso breve sempre que a entrada for ativada por um período arbitrário de tempo, e usamos a saída desse circuito para habilitar latch por um breve instante de tempo. Os detalhes deste circuito vão para além do que é suposto aprendermos nesta fase. O circuito usado corresponde a uma espécie de multivibrador monoestável, que por agora vamos chamar de detetor de pulso/pico.



Os flip-flops têm detetores de pico/pulso, representados nos diagramas/símbolos pelos triângulos. No esquema ao lado mostra-se explicitamente um detetor de pico conectado à entrada de habilitação de um latch, para transformá-lo em um flip-flop. Neste caso, temos o circuito de um flip-flop S-R:



C	S	R	Q	\bar{Q}
0	0	0	latch	latch
0	0	1	0	1
0	1	0	1	0
0	1	1	0	0
x	0	0	latch	latch
x	0	1	latch	latch
x	1	0	latch	latch
x	1	1	latch	latch