

Circuitos Sequenciais, Latches e Flip-flops



Universidade do Minho

Circuitos Sequenciais

- Ao contrário dos circuitos combinacionais, a **saída** não **depende** somente **das entradas** atuais mas **também do passado** destas
- O “Estado” engloba toda a **informação necessária acerca do passado** para prever a **saída atual** tendo em conta as **entradas atuais**
 - **Variáveis de Estado** - um ou mais **bits de informação**, dos quais depende a saída

Descrição de Circuitos Sequenciais



Universidade do Minho

- Tabela de Estados
 - Estados são símbolos (e.g., A, B ou Aberto, Fechado, etc.)
 - Ainda **não são** representados por **bits**
 - Próximo estado é função de:
 - Estado atual
 - Entradas atuais (no instante da mudança de estado)
 - Saídas são função de:
 - Estado atual
 - Entradas atuais (opcionalmente)
- Diagrama de Estados
 - Versão gráfica da Tabela de Estados

Estrutura duma Máquina de Estados

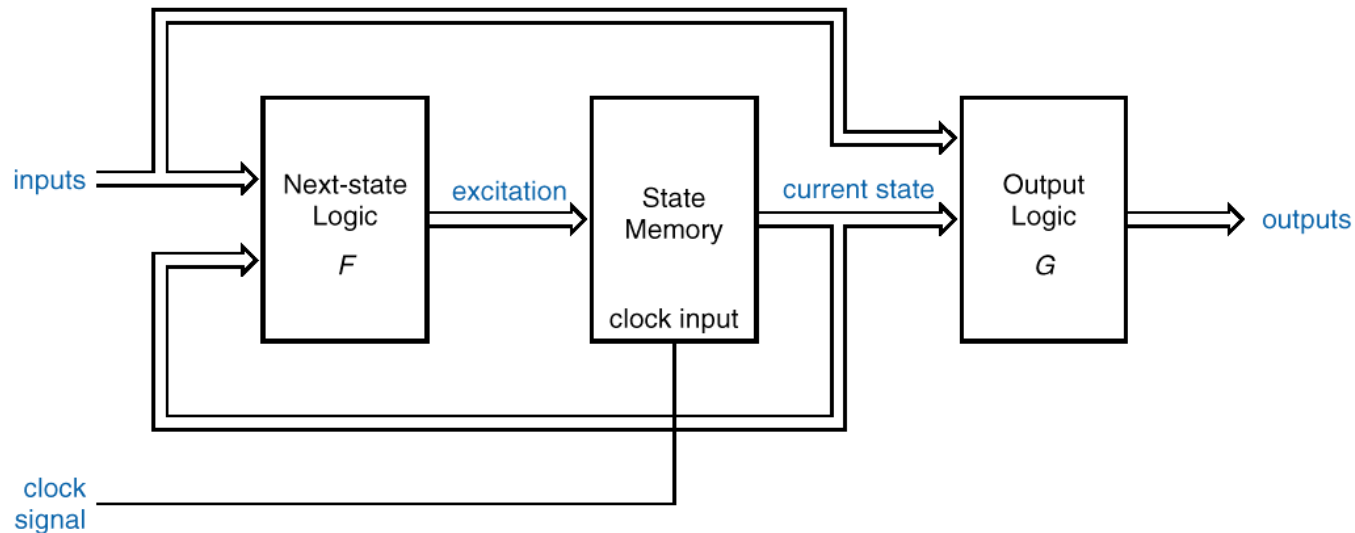


Figure 7-32 Clocked synchronous state-machine structure (Mealy machine).

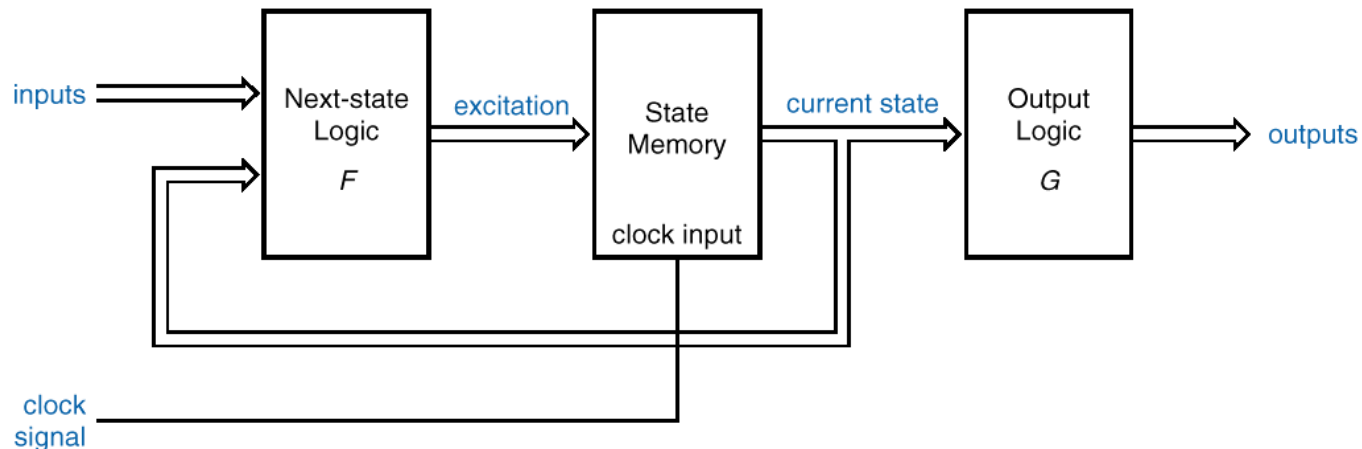


Figure 7-33 Clocked synchronous state-machine structure (Moore machine).

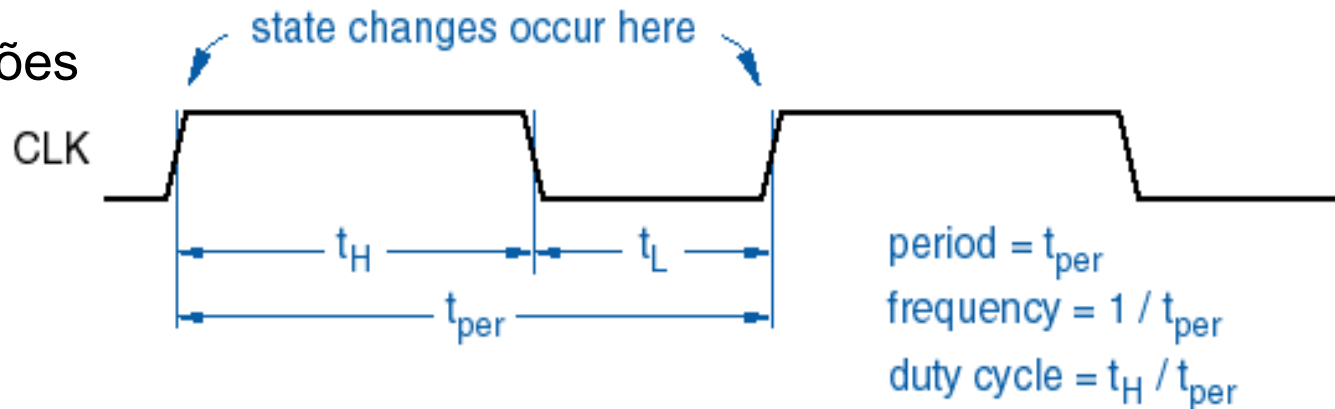
Blocos duma Máquina de Estados

- 1. Bloco com memória
 - Armazena a informação do estado atual
 - 2. Lógica do estado seguinte
 - Gera os sinais de excitação
 - Aplicados às entradas da memória de estado
 - Determinam o estado seguinte
 - 3. Lógica das saídas
 - Gera os sinais de saída
-
- Blocos 2 e 3 -> Circuitos combinacionais
 - Blocos básicos com memória (1) – assunto desta aula
 - Máquinas de estados – assunto da próxima aula
 - Junção de 1, 2, e 3

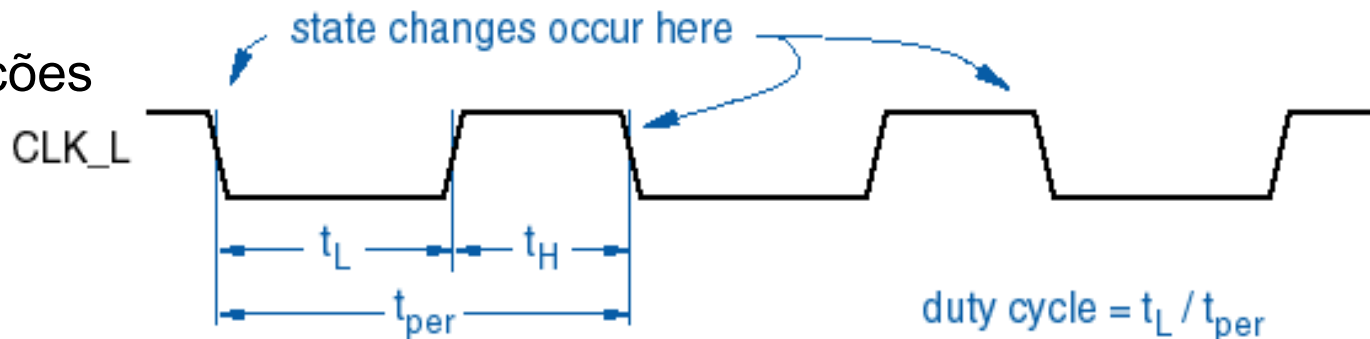
Sinais de Relógio (Clock)

- Determina os instantes de mudança de estado
 - Transição negativa ou positiva do clock, dependendo do tipo de bloco de memória)

Nas transições
positivas:



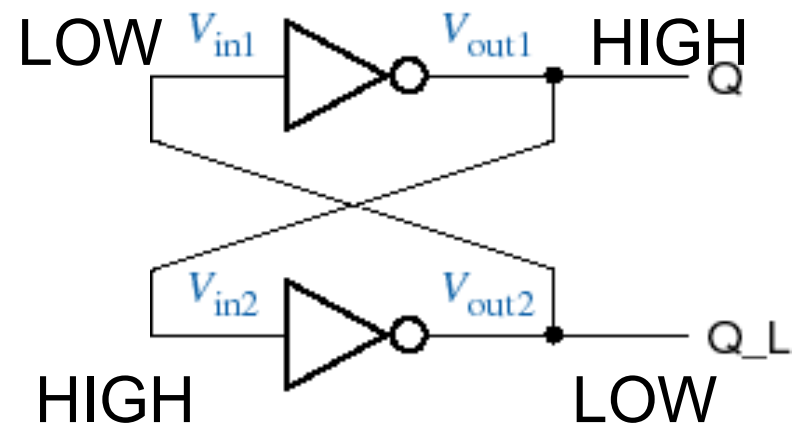
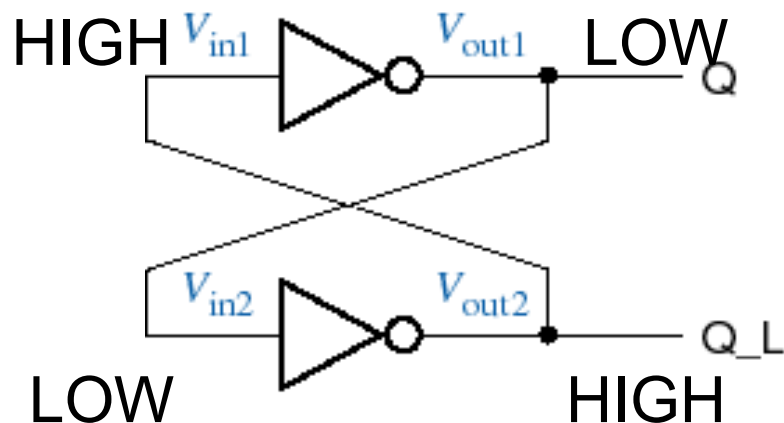
Nas transições
negativas:



Elemento Bi-estável

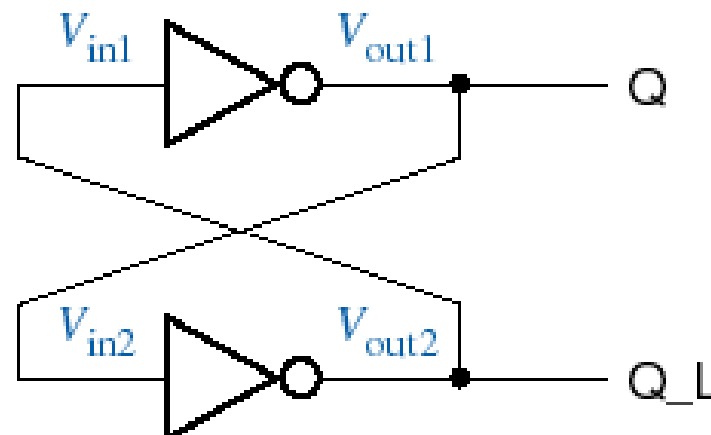
- Circuito sequencial mais simples
 - Dois estados
 - Requer apenas uma variável (bit) de estado, e.g., Q
 - Não tem sinal de clock

Circuito bi-estável sem entradas:

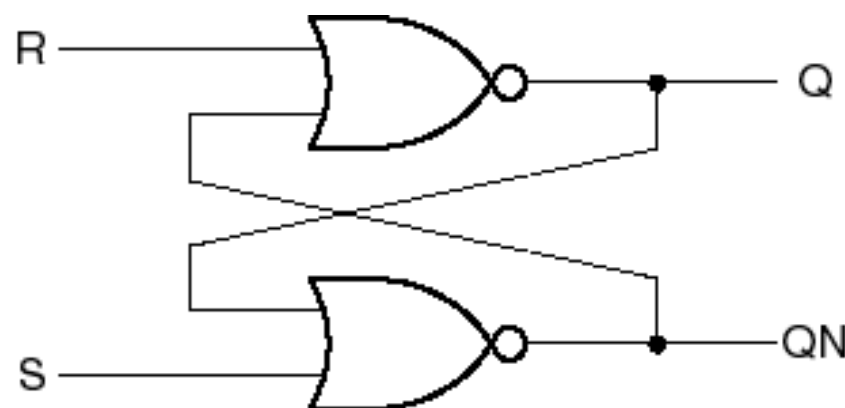


Elementos Bi-estáveis

- Como controlá-los?
– Com entradas



- Exemplo, Latch S-R

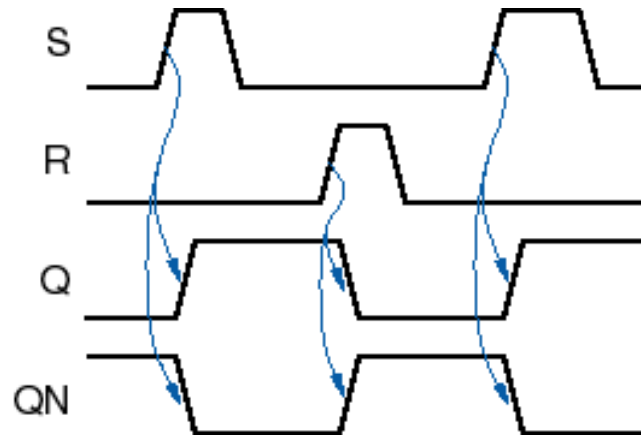


S	R	Q	Q _N
0	0	last Q	last Q _N
0	1	0	1
1	0	1	0
1	1	0	0

Funcionamento da Latch S-R

Funcionamento **normal** ($Q_N = Q'$)

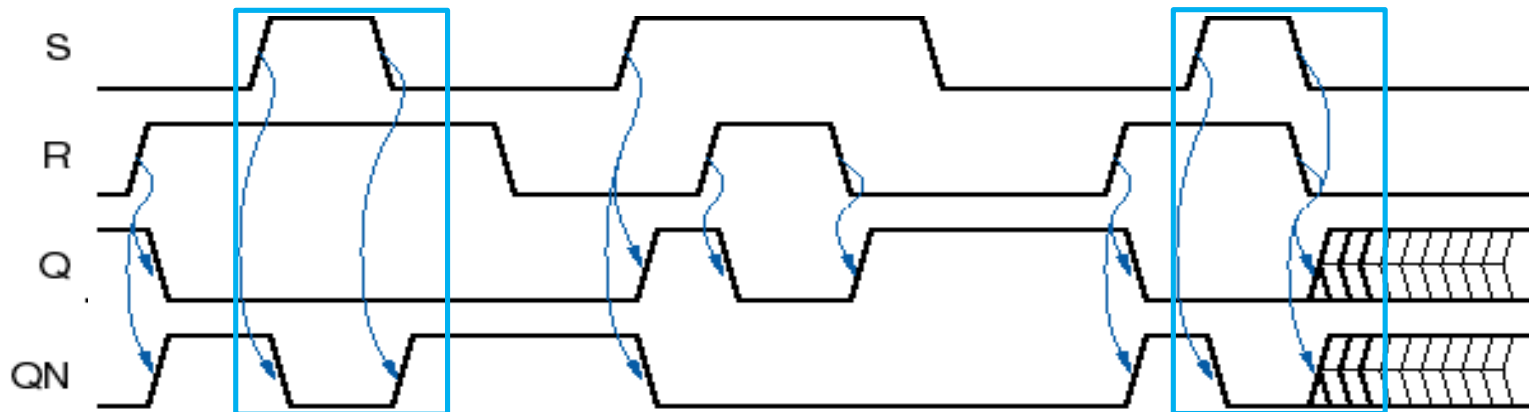
- Depende da utilização adequada das entradas S e R



Se as **entradas S e R** forem **desabilitadas simultaneamente**, a latch assume um estado indeterminado ou pode mesmo oscilar (astabilidade)

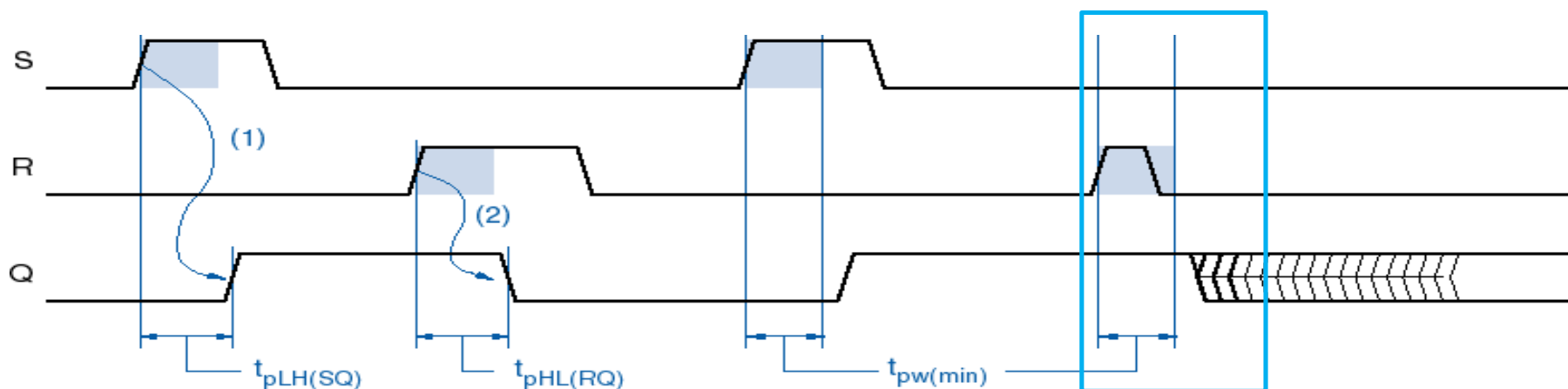


Situações **anormais** ($Q_N = Q$, estado indeterminado/oscilação)



Parâmetros Temporais da Latch S-R

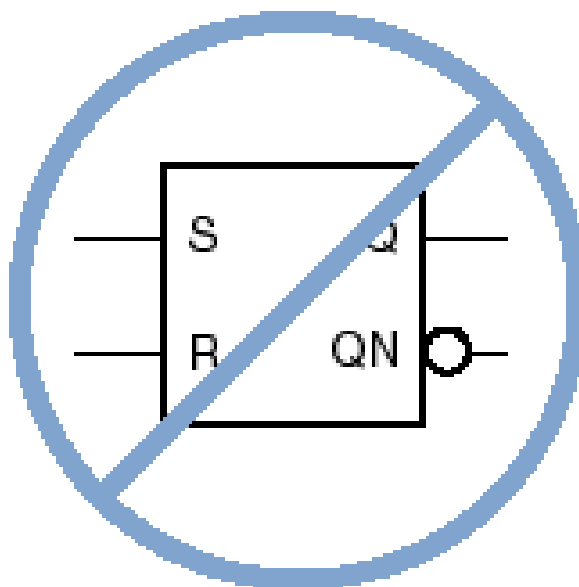
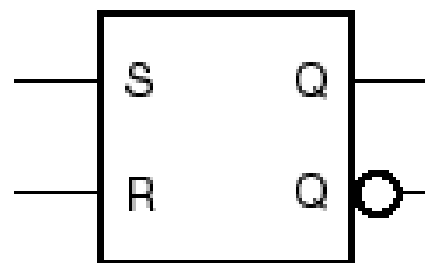
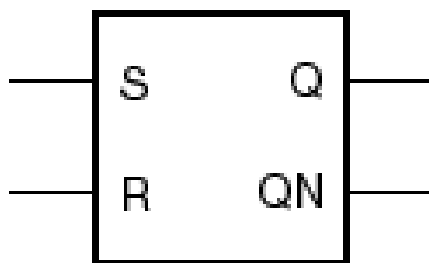
- Tempos (ou atrasos) de propagação (t_{pLH} , t_{pHL})
 - Comuns também à logica combinacional
- Largura mínima de pulso ($t_{pw(min)}$)
 - Duração mínima necessária do pulso de entrada para ser interpretado corretamente
 - Caso contrário pode não responder à entrada ou oscilar



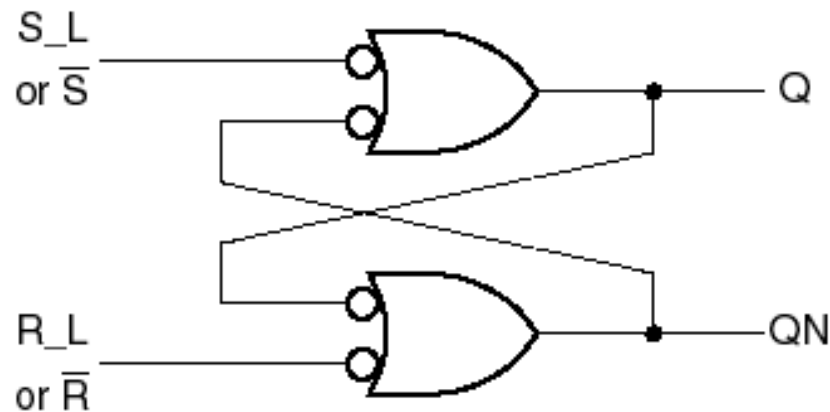
Símbolos da Latch S-R



Universidade do Minho

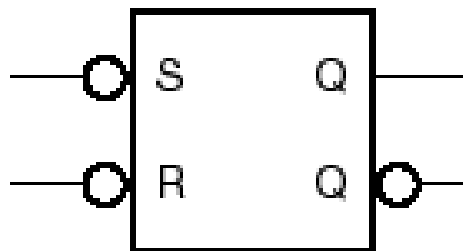


Latch com NANDs



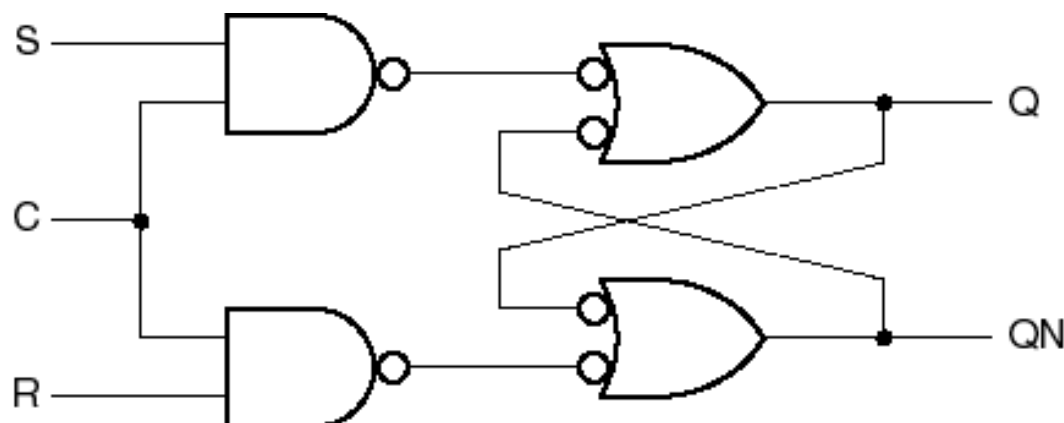
Latch $\bar{S} - \bar{R}$

Faz SET/RESET com “0”



S_L	R_L	Q	QN
0	0	1	1
0	1	1	0
1	0	0	1
1	1	last Q	last QN

Latch S-R com Enable

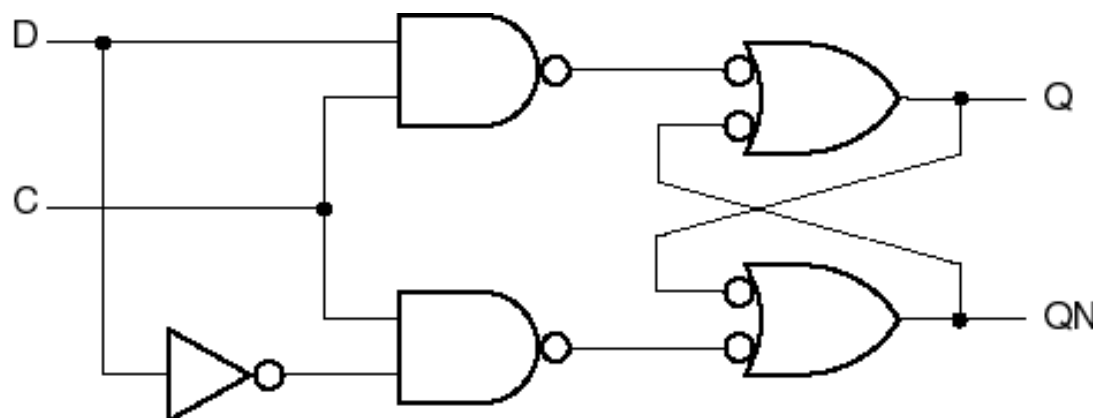


S	R	C	Q	QN
0	0	1	last Q	last QN
0	1	1	0	1
1	0	1	1	0
1	1	1	1	1
x	x	0	last Q	last QN

Latch tipo D

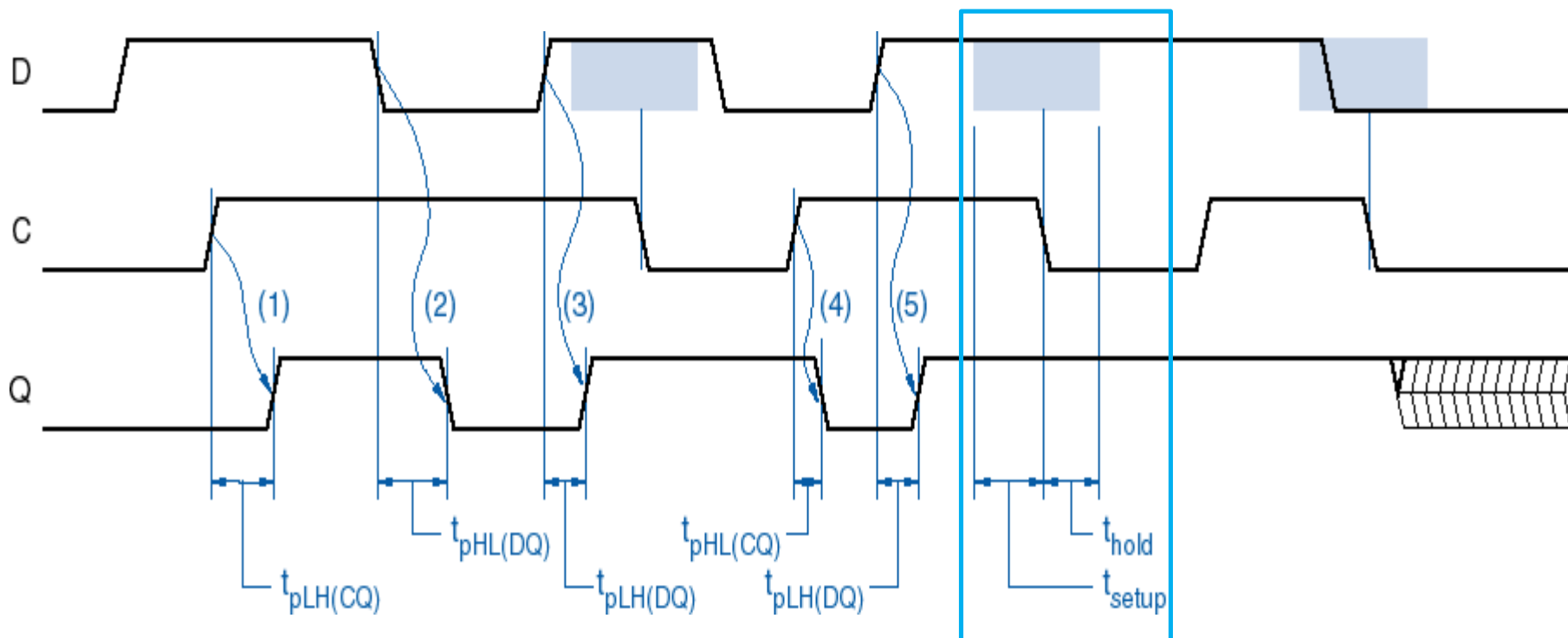
- Duas entradas
 - Enable (C)
 - Entrada de dados (D)
- Memória/registo volátil de 1 bit (Q)
 - Assume valor (D) quando $C = 1$
 - Assíncrono – qualquer momento
 - Retém os dados quando $C = 0$

C	D	Q	QN
1	0	0	1
1	1	1	0
0	x	last Q	last QN





Param. Temporais Latch tipo D

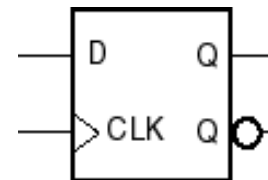
- Atrasos de Propagação (para C e para D)
- Tempo de “Setup” (D estável antes da transição de C)
- Tempo de “Hold” (D estável após transição de C)



Flip-flop D

- Tanto latches como flip-flops são registos de 1 bit (Q)
- Latch D é assíncrono (quando habilitado, $C = 1$)
 - $Q \leq D$, sempre que D varia
- Flip-flop D é síncrono
 - $Q \leq D$ somente na transição do clock (edge-triggered)
 - Ativado por borda ascendente (positive edge-triggered)
 - No instante de transição de CLK de “0” para “1” (exemplo abaixo)
 - Ativado por borda descendente (negative edge-triggered)
 - No instante de transição de CLK de “1” para “0”

D	CLK	Q	QN
0		0	1
1		1	0
x	0	last Q	last QN
x	1	last Q	last QN



Flip-flop D

Flip-flop D mestre-escravo
é uma implementação que
utiliza duas latches D
(existem outras)

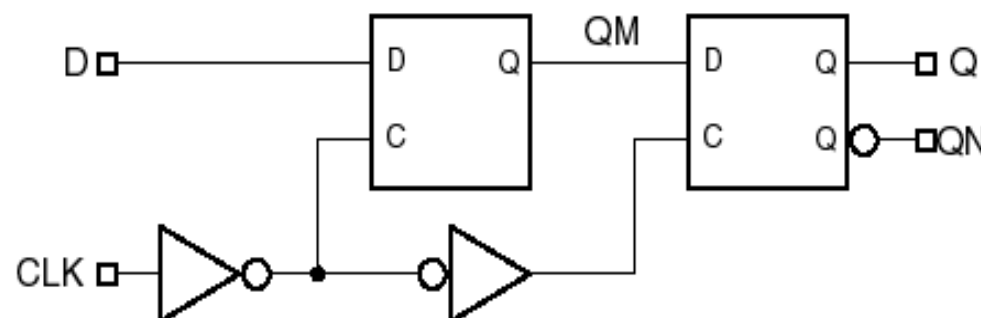
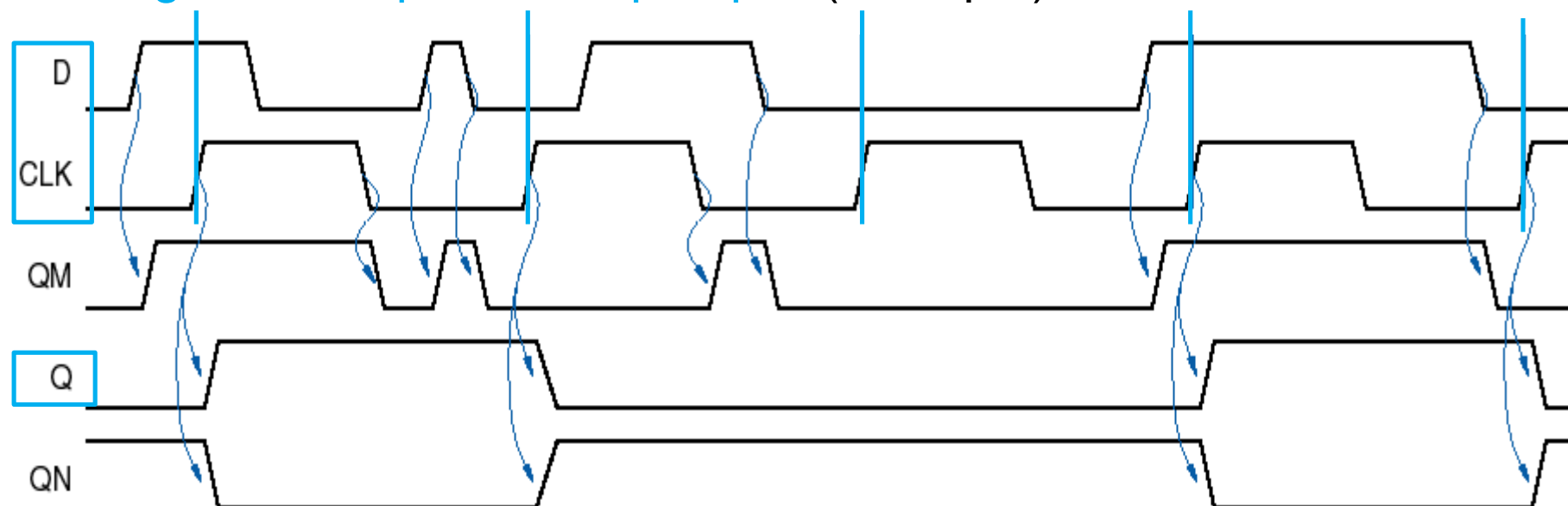
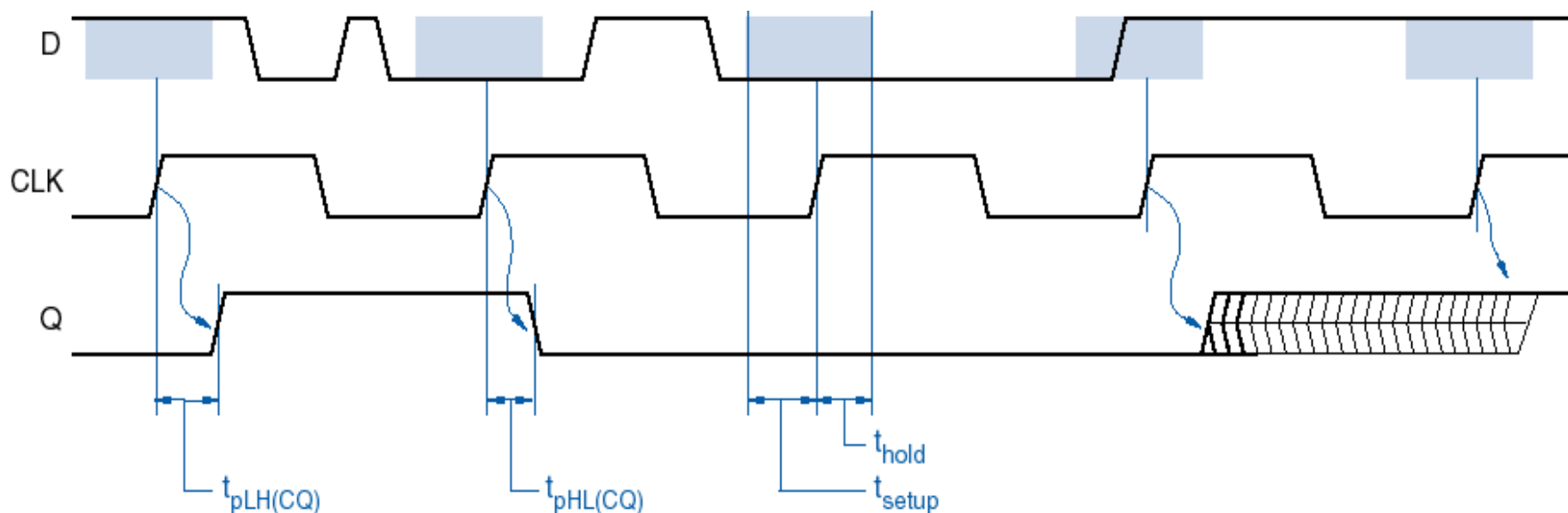


Diagrama temporal do flip-flop D (exemplo)



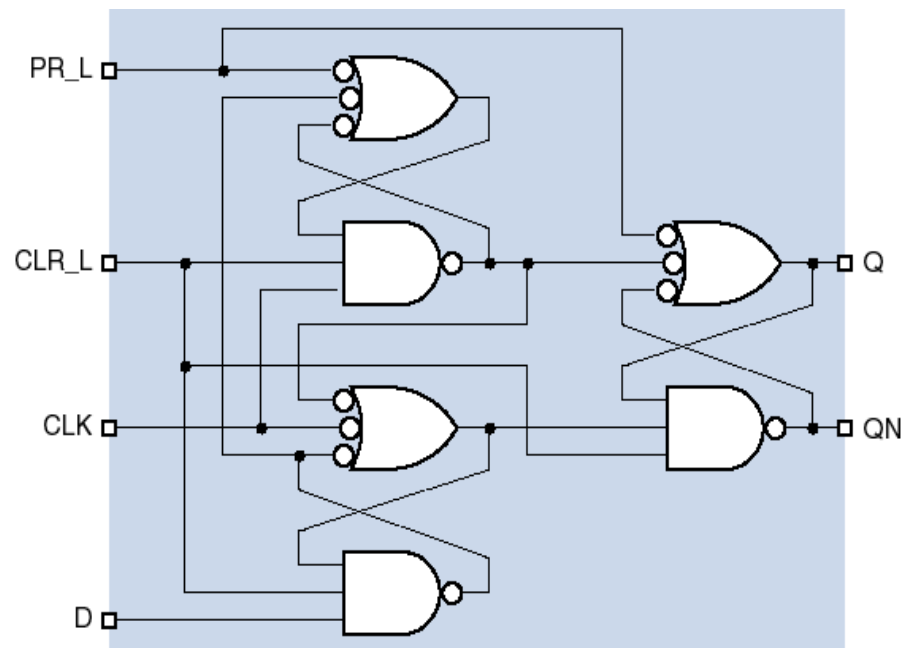
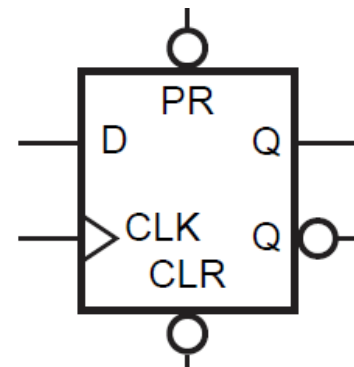
Flip-flop D

- Parâmetros temporais
 - Atraso de Propagação (desde a transição do CLK)
 - Tempo de “Setup” (D estável antes da transição de CLK)
 - Tempo de “Hold” (D estável após transição de CLK)



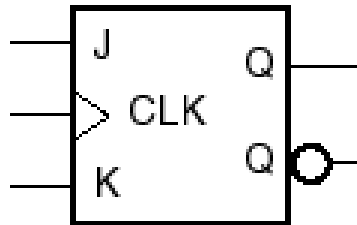
Flip-flop D TTL (IC 7474)

- Positive Edge-Triggered
- CLK, D, Q, QN
- Entradas adicionais de PRESET' e CLEAR'
 - Funcionam como na latch S'-R'
 - Assíncronas
 - Negadas
- 3 anéis de “feedback” (não é mestre-escravo)

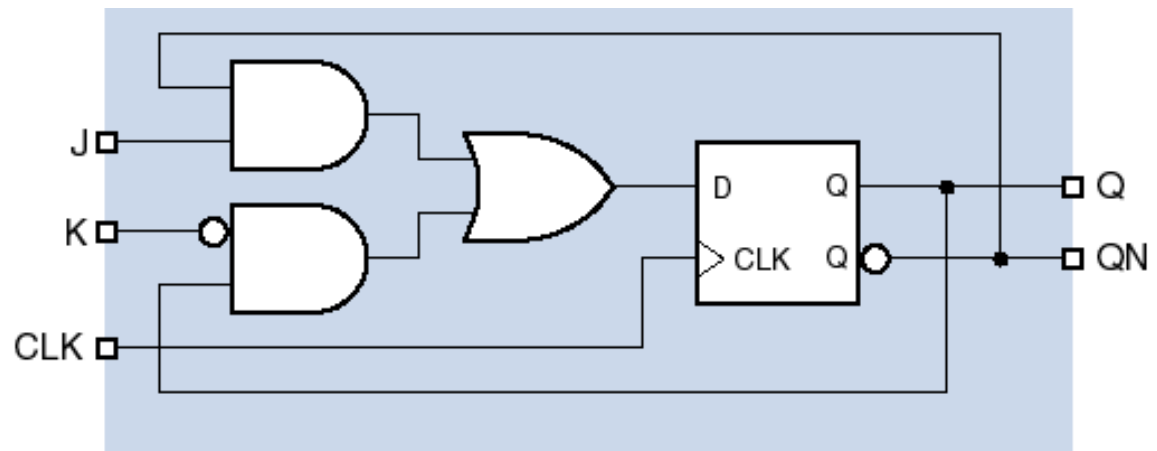






Flip-flop J-K

- **Síncrono** (como o flip-flop D)
- Duas entradas (como a latch S-R)
 - $J = 1$ (SET) $\Rightarrow Q = 1$
 - $K = 1$ (RESET) $\Rightarrow Q = 0$
- $J = K = 1 \Rightarrow$ Toggle



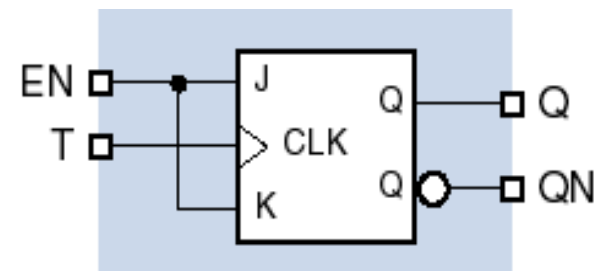
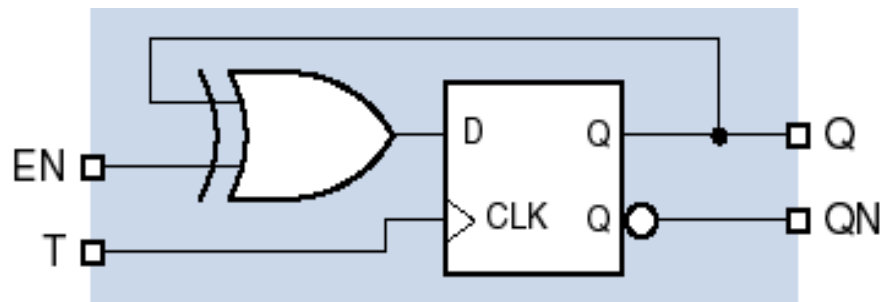
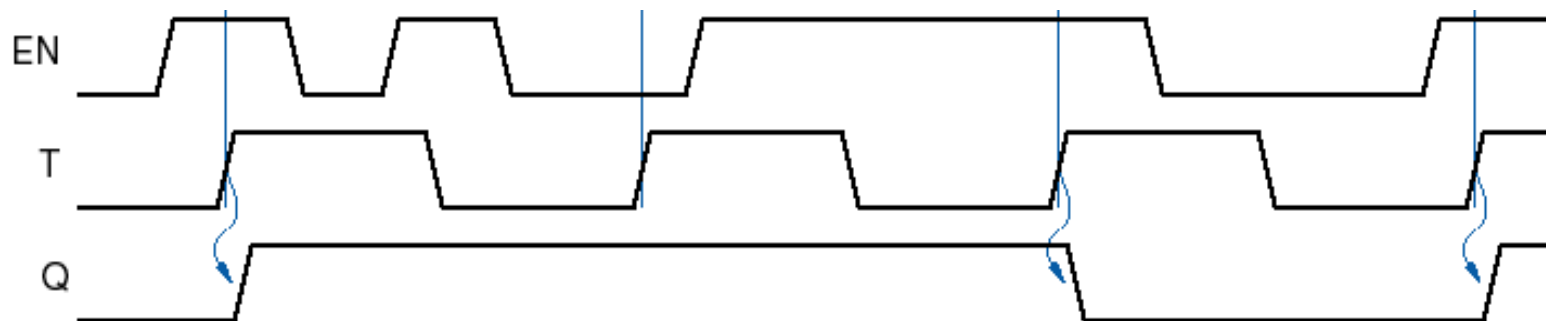
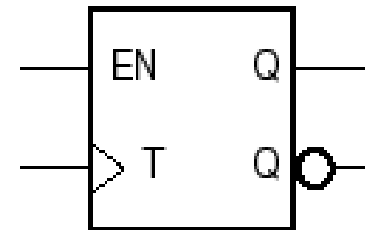
Exemplo de implementação com flip-flop D



J	K	CLK	Q	QN
x	x	0	last Q	last QN
x	x	1	last Q	last QN
0	0		last Q	last QN
0	1		0	1
1	0		1	0
1	1		last QN	last Q

Flip-flop T (Toggle)

- Há duas notações para as entradas
- Importantes na implementação de contadores



Equações Características

- Especificam o **estado seguinte (Q^*)** dos flip-flops em função das suas entradas e do estado actual (Q)
- Resultam da tabela de verdade com notação explícita

Device Type	Characteristic Equation
S-R latch	$Q^* = S + R' \cdot Q$
D latch	$Q^* = D$
Edge-triggered D flip-flop	$Q^* = D$
Master/slave S-R flip-flop	$Q^* = S + R' \cdot Q$
Master/slave J-K flip-flop	$Q^* = J \cdot Q' + K' \cdot Q$
Edge-triggered J-K flip-flop	$Q^* = J \cdot Q' + K' \cdot Q$
T flip-flop	$Q^* = Q'$
T flip-flop with enable	$Q^* = EN \cdot Q' + EN' \cdot Q$