

FLIP-FLOPS: RS e D (teoria)

A eletrônica digital divide-se basicamente em duas áreas:

a) lógica combinacional: que é caracterizada por circuitos cujas saídas dependem das entradas presentes no instante observado.

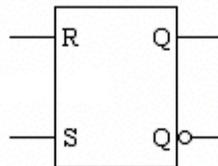
b) lógica sequencial: que é caracterizada por circuitos cujas saídas dependem não somente das entradas presentes no instante observado, mas também das entradas anteriores e da sequência com que foram aplicadas.

Os flip-flops são conhecidos também como multivibradores biestáveis, pois possuem 2 estados estáveis e mudam de estado (passam de um estado para outro) somente pela ação de um sinal externo.

Normalmente esses pulsos externos são pulsos de controle, denominados "clock". Ao se aplicar um pulso no FF (abreviatura de Flip-Flop) este muda de estado, assim permanecendo até que outro pulso seja aplicado para leva-lo a outro estado.

Por esse motivo o Flip-flop tem a importante característica de "memória".

FLIP-FLOP BÁSICO TIPO "RS"



A figura acima mostra um FF RS básico, onde devem ficar caracterizadas:

Entradas: R e S

R = reset (restabelecer)

S = Set (estabelecer)

As entradas R e S determinam o *modo de operação* (MO) do FF sendo importantíssimo entender o que significa, pois com isso será possível analisar o comportamento de um circuito sequencial implementado com FFs, como por exemplo, contadores binários.

O modo de operação depende do nível lógico do sinal aplicado nas entradas:

$S = 1; R = 0 \rightarrow$ modo de operação SET

$S = 0; R = 1 \rightarrow$ modo de operação RESET

$S = 0; R = 0 \rightarrow$ modo de operação HOLD

$S = 1; R = 1 \rightarrow$ modo de operação PROIBIDO; INDETERMINADO

Saídas Q e Q'

Quando o FF opera normalmente as saídas Q e Q' devem ser complementadas, caso isto não ocorra o mesmo não tem utilidade dentro de um circuito. Quando isso ocorre, ou seja, quando as saídas não se complementam, dizemos que o mesmo assume o *estado proibido ou indeterminado*.

OBS: Q' representa a saída complementada.

As saídas Q e Q' dependem então das entradas e as entradas determinam o modo de operação do FF.

Vamos analisar 7 casos (7 instantes), para melhor entender o conceito:

Consideremos que ao ligar o FF a seu estado inicial seja $Q = 1$ e $Q' = 0$, portanto, modo de operação SET.

Vamos definir como estado inicial a simbologia **Q_n**. A definição de Q_n como estado inicial é uma opção, pois alguns autores definem com outra simbologia, como por exemplo, Q_i, Q₀, etc.

Em se tratando de lógica sequencial ao se alterar as entradas, as saídas também sofrerão alterações, pela mudança de estado FF. A essa alteração denominaremos **Q_{n+1}**. Alguns autores se referem essa condição como Q_f (saída futura).

Assim, Q_n = estado inicial; **Q_{n+1}** é o estado após a alteração da entrada e **Q_{n+1}'** é o seu complemento.

Exemplo: preencher a tabela levando em consideração o estado inicial SET

Sendo o estado inicial SET, então Q_n = 1 (Q_n' = 0)

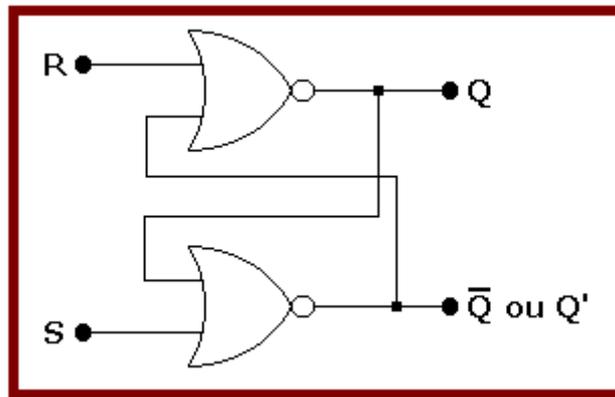
Instante	Entradas		Saídas		MO
	S	R	Q _{n+1}	Q _{n+1} '	
1	0	0	1	0	Hold
2	1	0	1	0	Set
3	0	0	1	0	Hold
4	0	1	0	1	Reset
5	1	1	?	?	Ind
6	0	0	0	1	Hold
7	1	0	1	0	Set

IMPLEMENTAÇÃO COM PORTAS LÓGICAS NOR

A figura a seguir mostra um FF RS básico implementado com portas NOR.

Observe as saídas **Q** e seu complemento **Q'**. Essas saídas devem sempre ser complementadas.

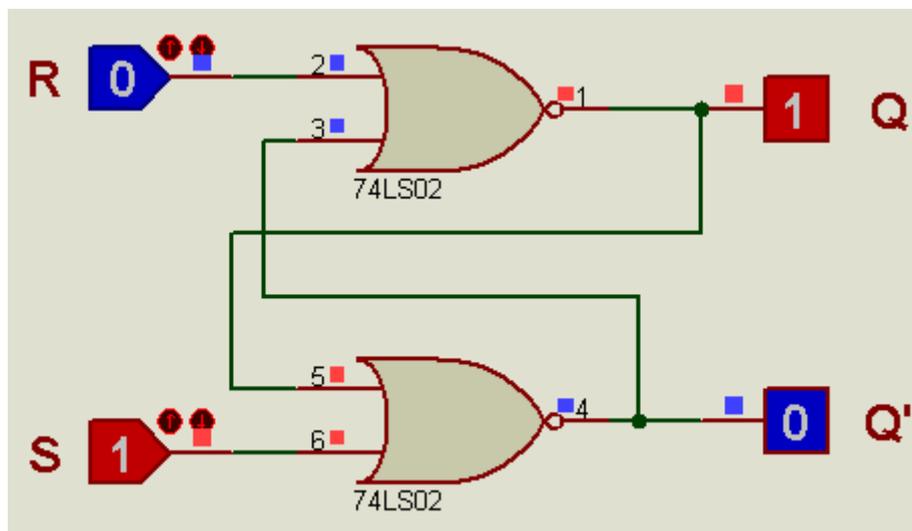
Normalmente a saída **Q** é a mais utilizada. Levando-se em consideração o modo de operação, então: se Q = 1 o modo de operação é SET e se Q = 0 o modo de operação é RESET.



Condição inicial ou estado inicial: $Q_n = 0$ (Reset), logo, $Q_n' = 1$

Instante	Entradas		Saídas		MO
	S	R	Q_{n+1}	Q_{n+1}'	
1	0	0	0	1	Hold
2	1	0	1	0	Set
3	0	0	1	0	Hold
4	0	1	0	1	Reset
5	1	1	?	?	Ind
6	0	0	0	1	Hold
7	1	0	1	0	Set
8	1	1	?	?	Ind
9	0	0	1	0	Hold

A figura a seguir ilustra a simulação do FF RS básico com o CI comercial 74LS02 (porta NOR) no Proteus ISIS.

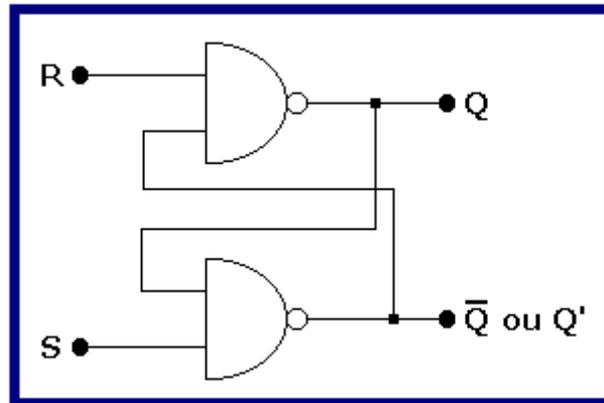


O modo de operação é Set ($S = 1$; $R = 0$), onde se observa que a saída Q é igual a 1. A figura a seguir mostra o FF RS básico implementado com portas NAND.

Os modos de operação SET e RESET não se alteram, no entanto, os valores nas entradas R e S para os modos de operação HOLD e IND se invertem.

Veja a seguir a comparação:

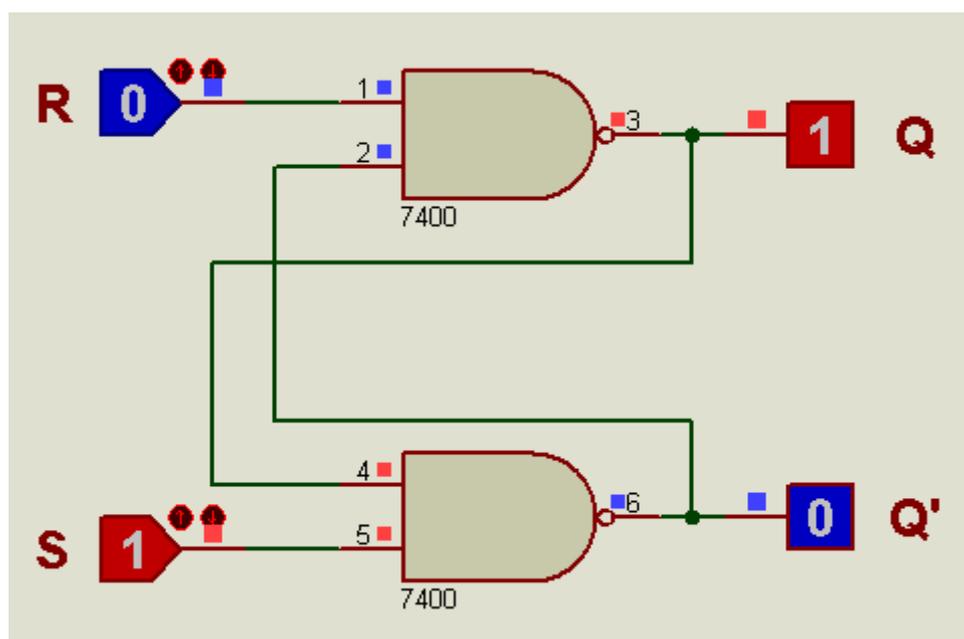
Implementação com NOR	Implementação com NAND
S = R = 0 → modo de operação HOLD S = R = 1 → modo de operação IND	S = R = 0 → modo de operação IND S = R = 1 → modo de operação HOLD



Vamos considerar o estado inicial SET ($Q_n = 1$ e $Q_n' = 0$)

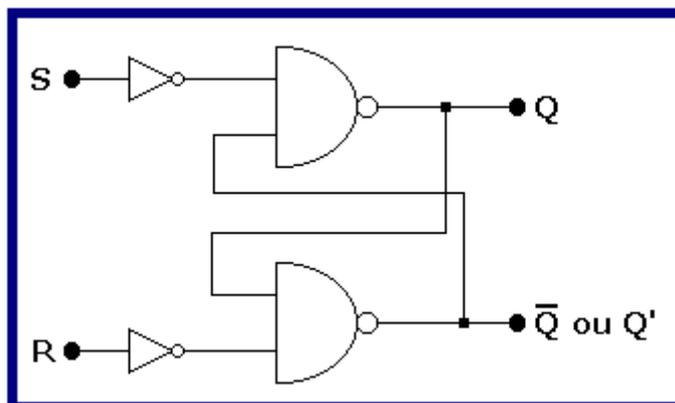
Instante	Entradas		Saídas		MO
	S	R	Q_{n+1}	Q_{n+1}'	
1	0	0	?	?	Ind
2	1	0	1	0	Set
3	0	0	?	?	Ind
4	0	1	0	1	Reset
5	1	1	0	1	Hold
6	0	0	?	?	Ind
7	1	0	1	0	Set
8	1	1	1	0	Hold
9	0	0	?	?	Ind

A figura a seguir ilustra a simulação do FF RS básico com o CI comercial 74LS00 (porta NAND) no Proteus ISIS.



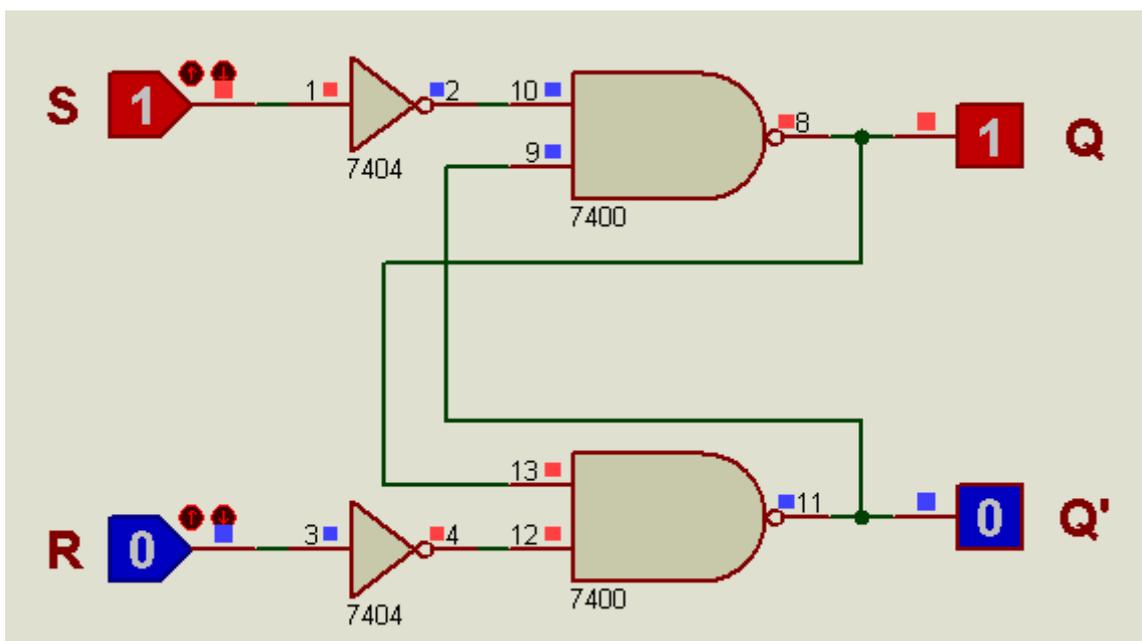
Análise de um FF RS básico implementado com portas NAND

Vamos adotar como padrão um FF RS básico implementado com portas NAND e inversores, conforme ilustra a figura abaixo:



A figura a seguir mostra o circuito simulado no Proteus ISIS.

É muito importante verificar o posicionamento das entradas R e S, que estão com suas posições invertidas em relação aos circuitos anteriores.



Considerando o estado inicial SET, onde $Q = 1$ e $Q' = 0$, vamos analisar o que ocorre em cada instante, conforme mostra a tabela da verdade.

Levando-se em conta, conforme tivemos a oportunidade de verificar anteriormente, um FF somente tem função útil quando as suas saídas se complementam.

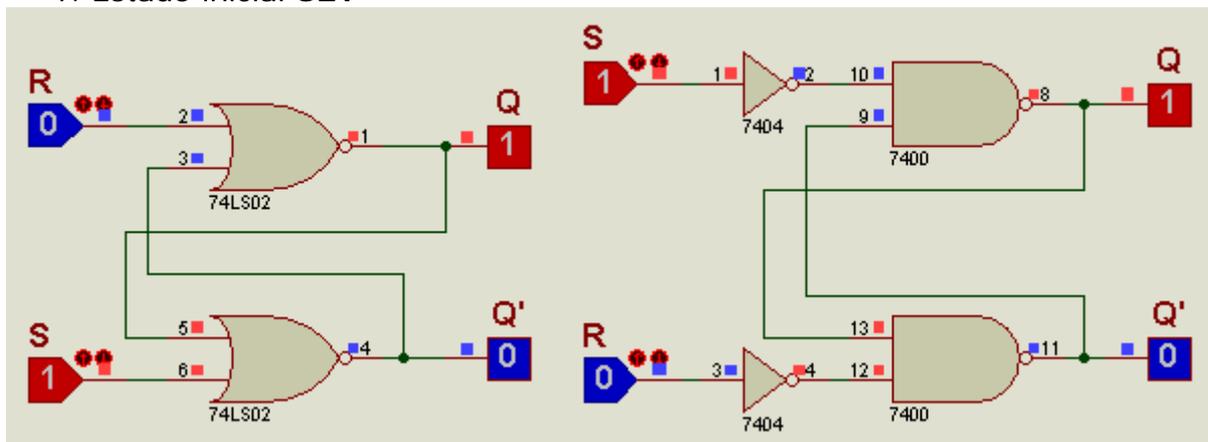
O estado proibido é denominado indeterminado pois, as suas saídas podem estar tanto em NL 1 como em NL 0. Quem vai determinar isso é o tipo de implementação adotado.

Vamos considerar o estado inicial SET ($Q_n = 1$ e $Q_n' = 0$)

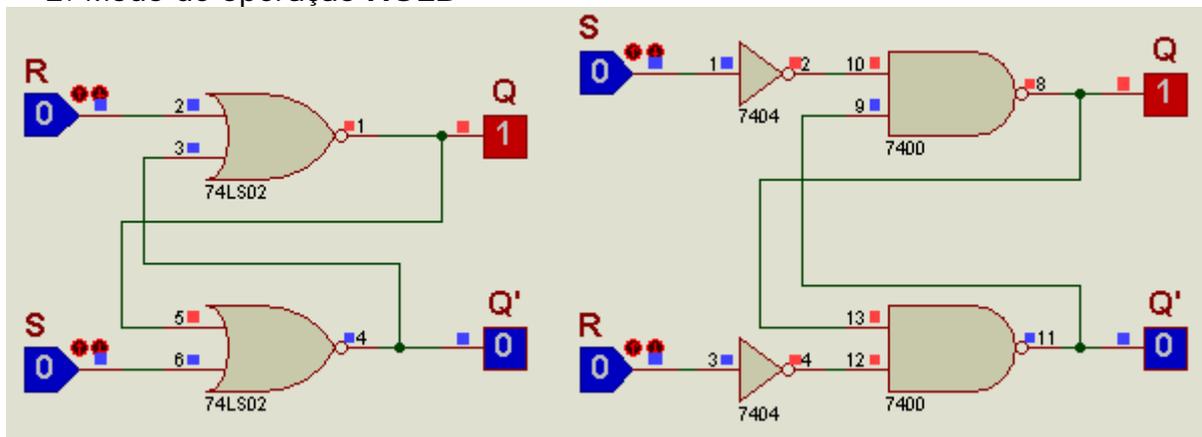
Instante	Entradas		Saídas		MO
	S	R	Q_{n+1}	Q_{n+1}'	
1	0	0	1	0	Hold
2	1	0	1	0	Set
3	0	0	1	0	Hold
4	0	1	0	1	Reset
5	1	1	1	1	Ind
6	0	0	0	1	Hold
7	1	0	1	0	Set
8	1	1	1	1	Ind
9	0	0	1	0	Hold
10	1	0	1	0	Set
11	0	1	0	1	Reset
12	0	0	0	1	Hold
13	1	0	1	0	Set
14	0	0	1	0	Hold
15	1	1	1	1	Ind

As figuras a seguir mostram a simulação feita no Proteus ISIS, comparando a implementação do FF RS básico com portas NOR e NAND respectivamente.

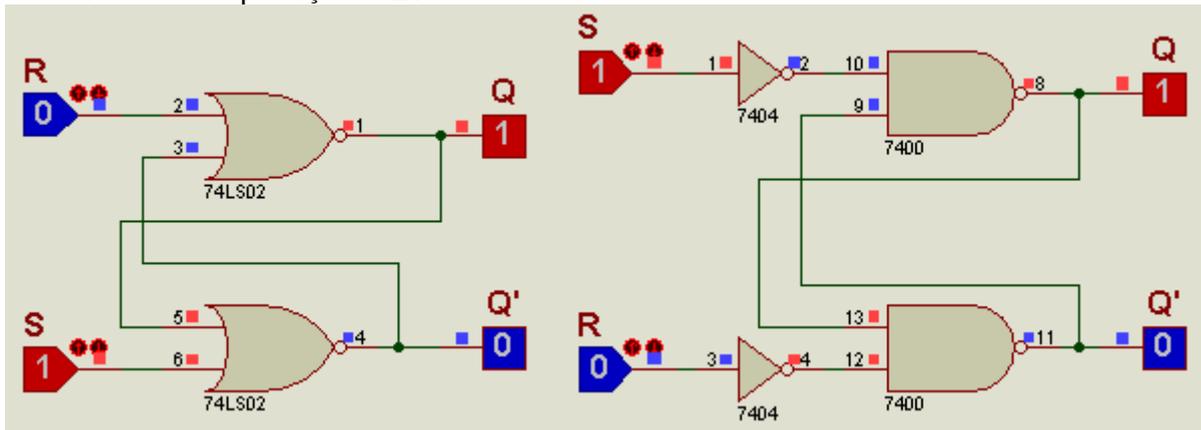
1. Estado inicial SET



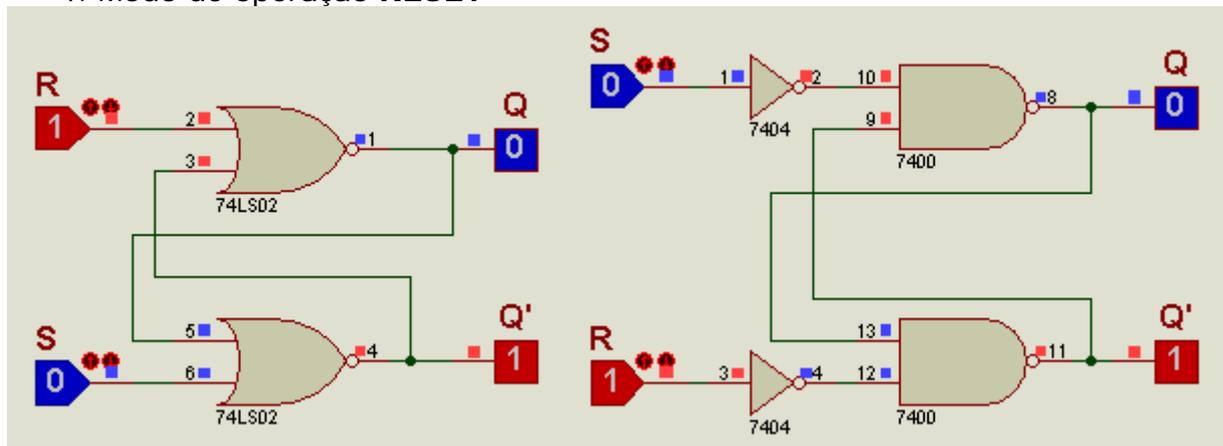
2. Modo de operação HOLD



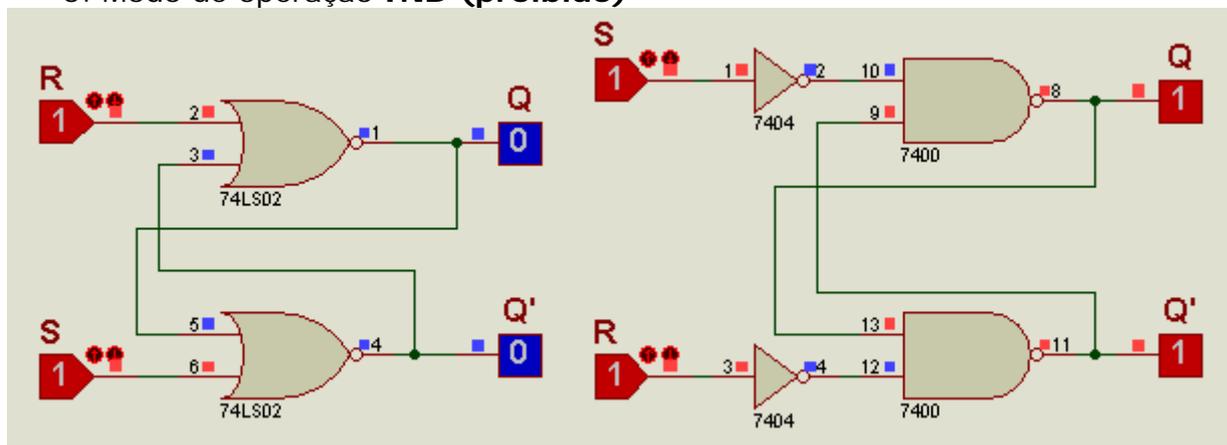
3. Modo de operação SET



4. Modo de operação RESET



5. Modo de operação IND (proibido)

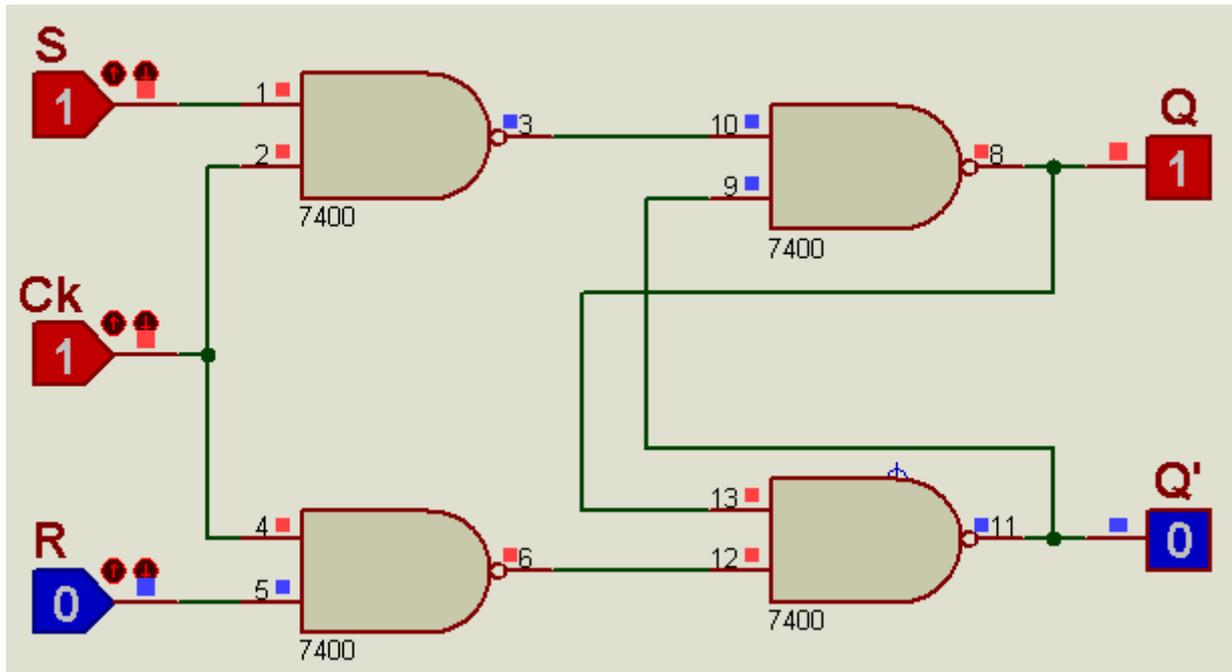


Flip-flop RS básico com clock

Implementado com portas NAND e Inversores

Para que o FF básico seja controlado por um sinal externo (pulso de clock), faremos uma modificação no circuito, substituindo as portas NOT (inversores) por portas NAND, conforme mostra o circuito a seguir.

Neste caso o controle do FF será feito por nível do pulso de clock, ou seja, NL 1 ou NL 0.



Se CK for igual a zero ($CK = 0$) as condições do FF serão inoperantes, ou seja, as entradas não terão qualquer influência nas saídas.

Tomemos como exemplo a tabela abaixo, onde o estado inicial é $Q_n = 1$ (SET).

Instante	Entradas			Saídas		MO
	S	R	CK	Q_{n+1}	Q_{n+1}'	
1	0	0	0	1	0	Inoperante
2	1	0	0	1	0	Inoperante
3	0	0	0	1	0	Inoperante
4	0	1	0	0	1	Inoperante
5	1	1	0	1	1	Inoperante
6	0	0	1	1	0	Hold
7	1	0	1	1	0	Set
8	1	1	1	1	1	Ind
9	0	0	1	1	0	Hold
10	1	0	1	1	0	Set
11	0	1	1	0	1	Reset
12	0	0	1	0	1	Hold
13	1	0	1	1	0	Set
14	0	0	1	1	0	Hold
15	1	1	1	1	1	Ind

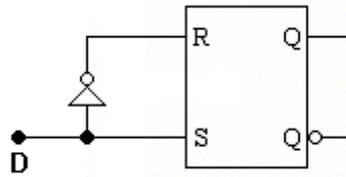
Analisando a tabela acima, o FF estará operando somente a partir do instante 6, cujo modo de operação é HOLD, pois o clock em nível lógico 1 habilitará as portas NAND.

FLIP-FLOP TIPO "D"

A partir do FF RS podemos implementar um FF do tipo D.

O FF do tipo D tem a vantagem de eliminar o estado indeterminado (proibido).

Vejamos o diagrama a partir do FF RS básico:



Quando $D = 1 \rightarrow Q = 1$ (pois $S=1$ e $R=0$) \rightarrow modo de operação SET

Quando $D = 0 \rightarrow Q = 0$ (pois $S=0$ e $R=1$) \rightarrow modo de operação RESET

D = entrada de dados

Conclui-se então que no FF tipo D não existe os modos de operação:

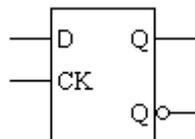
*HOLD
INDETERMINADO*

Analisando a tabela da verdade:

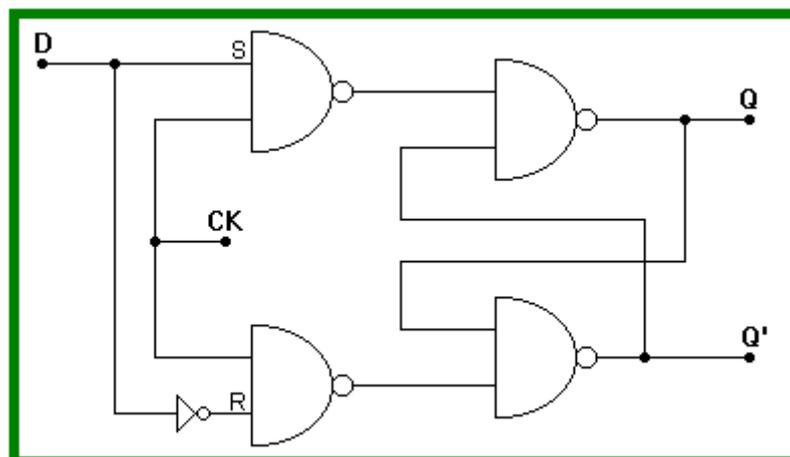
D	R	S	MO	Q	Q'
1	0	1	set	1	0
0	1	0	reset	0	1

Implementando com portas NAND e inversores, a partir do FF RS básico:

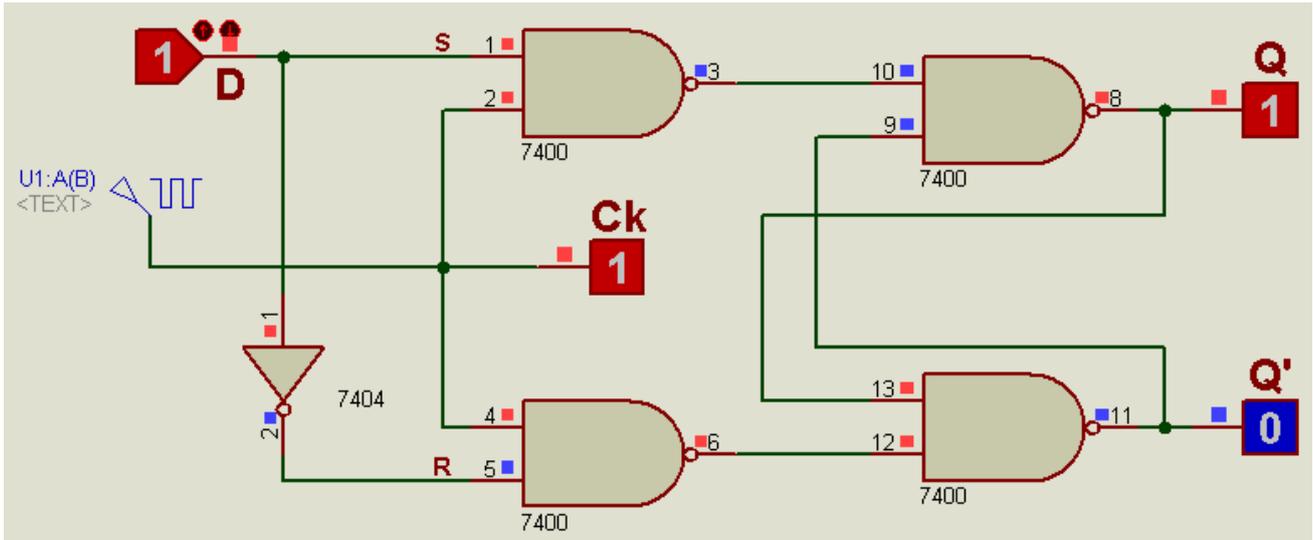
A figura a seguir mostra o bloco de um FF tipo D, com entrada de clock.



Implementação de um FF tipo D, com entrada clock, utilizando portas NAND.



A figura a seguir ilustra a simulação no Proteus ISIS:



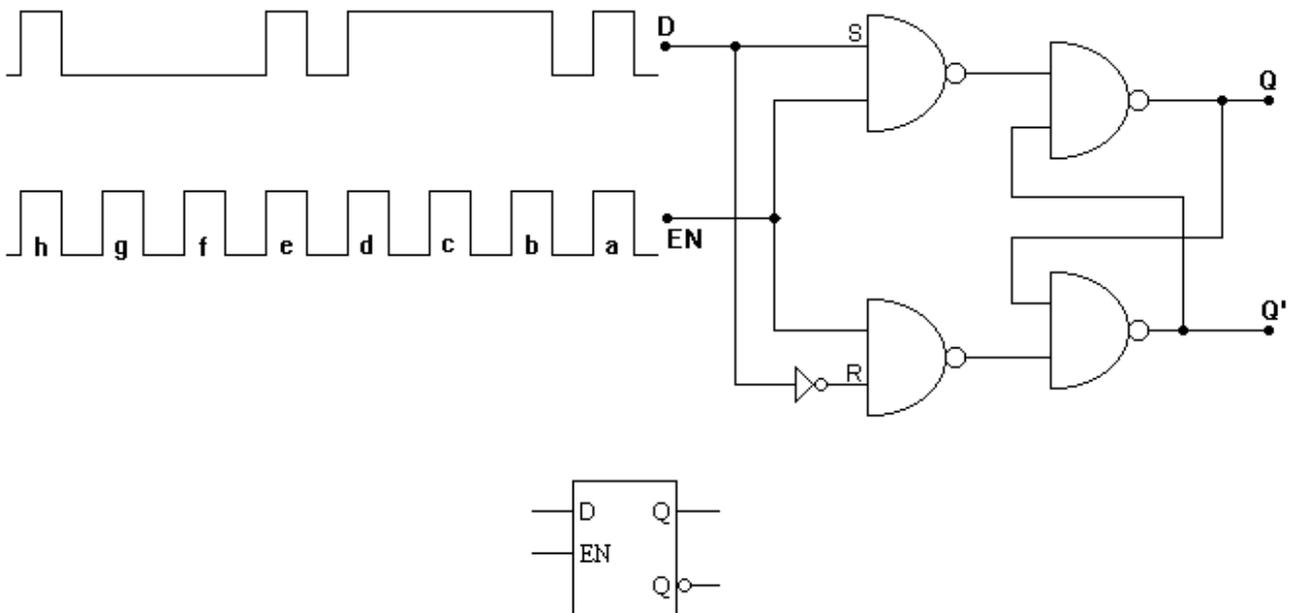
Supondo o estado inicial $Q_n = 1$ (set)

CK (EN)	D	Q_{n+1}	Q_{n+1}'	MO
1	0	0	1	Reset
1	1	1	0	Set
0	X	bloqueado		
0	X	bloqueado		

Exemplo: Para o circuito abaixo, listar as condições de saída do FF D do pulso "a" até o pulso "h" do clock.

Neste caso, como a atuação do clock se dá em nível lógico, chamaremos essa entrada de EN (enable, do inglês habilitar).

Quando $EN = 1$ o FF estará habilitado, caso contrário, será bloqueado.



Pulsos	Nível lógico na entrada "D"	Q	Q'	MO
a	1	1	0	set
b	1	1	0	set
c	1	1	0	set
d	1	1	0	set
e	1	1	0	set
f	0	0	1	reset
g	0	0	1	reset
h	1	1	0	set