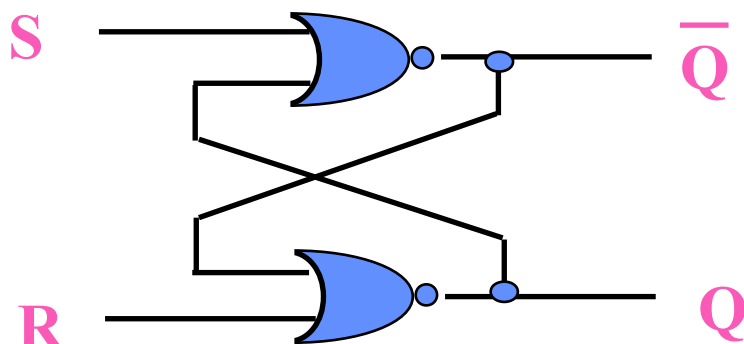


Técnicas Digitais para Computação

Latches

1. Latch RS (ou SR)

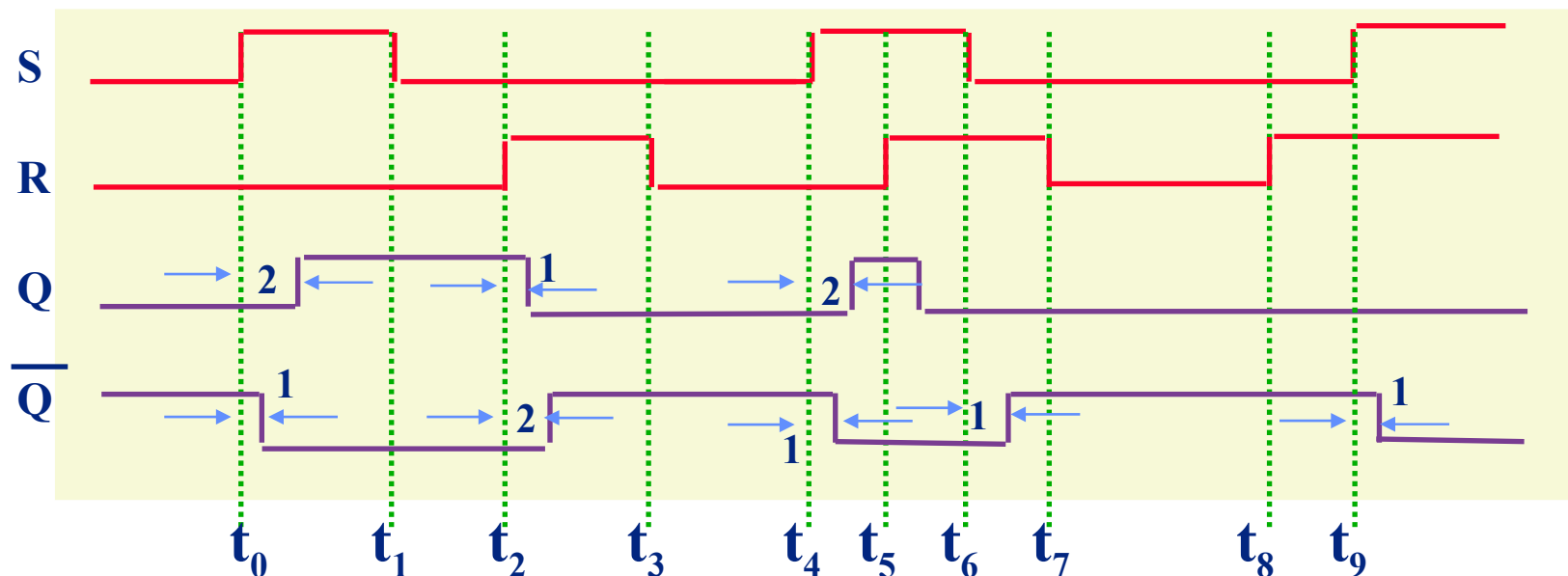


possível seqüência de valores

	R	S	Q	\overline{Q}
t=0	0	1	1	0
t=1	0	0	1	0
t=2	1	0	0	1
t=3	0	0	0	1
t=4	1	1	0	0

Obs: supomos que o atraso das portas NOR é 1 ns

Exemplo de diagrama de tempos



Conclusões

- Q e \overline{Q} são complementares, exceto no caso $R = 1, S = 1$, que deve ser evitado.

$R = 0, S = 0$ estado permanece no valor anterior

$R = 0, S = 1$ estado = 1 $S = SET$

$R = 1, S = 0$ estado = 0 $R = RESET$

$R = 1, S = 1$ deve ser evitado

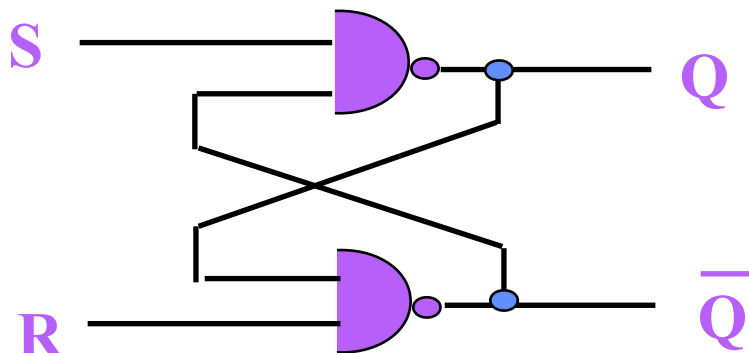
- R e S são portanto entradas de controle
deve-se ligar apenas uma delas a cada vez
enquanto ambas estiverem desligadas, o latch mantém o valor anterior
- LATCH é um circuito seqüencial assíncrono
memória é obtida pela realimentação num circuito combinacional

Tabela -verdade do latch RS

R	S	Q	Q(next)	\overline{Q} (next)
0	0	0	0	1
0	0	1	1	0
1	0	X	0	1
0	1	X	1	0
1	1	X	0	0

R	S	Q_{n+1}
0	0	Q_n
0	1	1
1	0	0
1	1	não usado

Usando NAND's



Possível sequência de valores

	S	R	Q	\bar{Q}
t=0	1	0	0	1
t=1	1	1	0	1
t=2	0	1	1	0
t=3	1	1	1	0
t=4	0	0	1	1

Conclusões

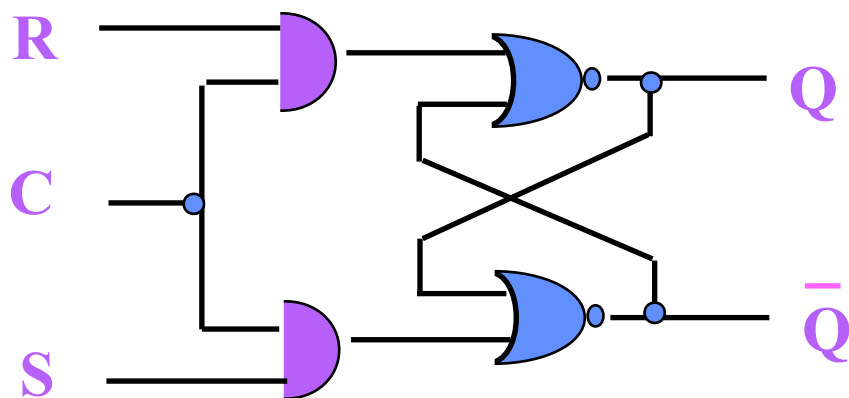
- $R = 1, S = 1$ mantém estado anterior (ambas entradas inativas)
- $R = 1, S = 0$ “liga” o latch SET ativo em 0
- $R = 0, S = 1$ “desliga” o latch RESET ativo em 0
- $R = 0, S = 0$ deve ser evitado (ambas entradas ativas)

Tabela-verdade

S	R	Q_{n+1}
0	0	não usado
0	1	1
1	0	0
1	1	Q_n

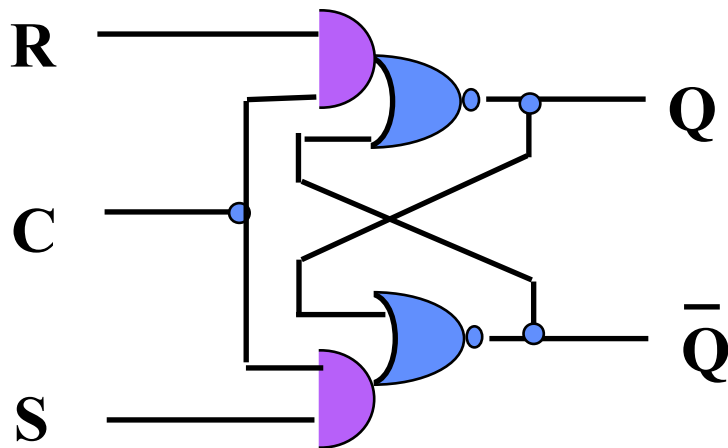
2. Latch RS controlado

acrescentar sinal de controle que habilita a transição da saída



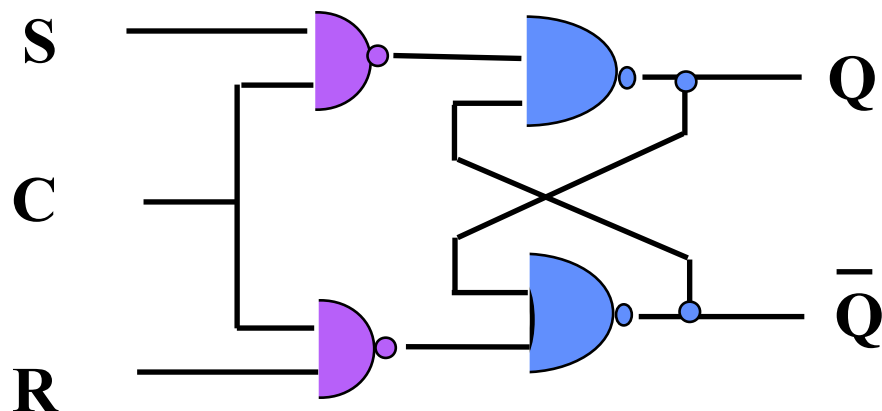
- quando $C = 1$
latch opera conforme já visto

esquema equivalente



- quando $C = 0$
alterações em R e S
não afetam estado do latch

Latch RS controlado com NANDs



• quando $C = 0$ ➔ alterações não afetam estado

• quando $C = 1$

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	não usado

RESET ativo em 1
SET ativo em 1

inverteu a lógica devido ao NAND adicional na entrada

3. Latch tipo D

Problema do Latch RS: $S = 1, R = 1$ → estado indeterminado

Solução: LATCH TIPO D

- uma entrada de dados apenas, ligada a S
- entrada R é sempre o complemento de S

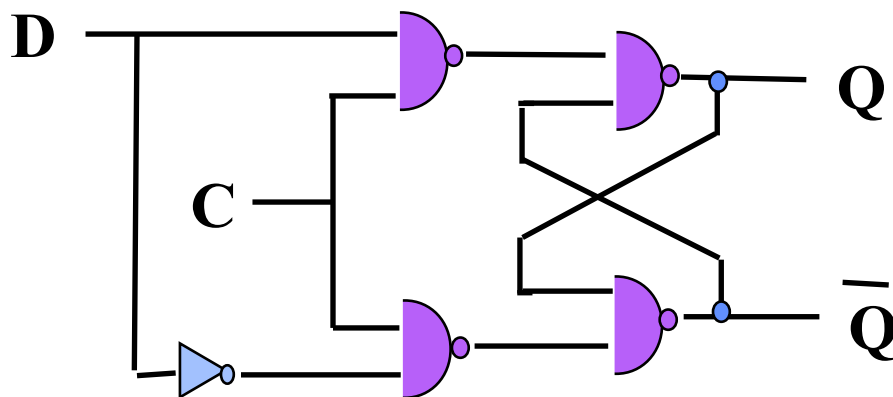


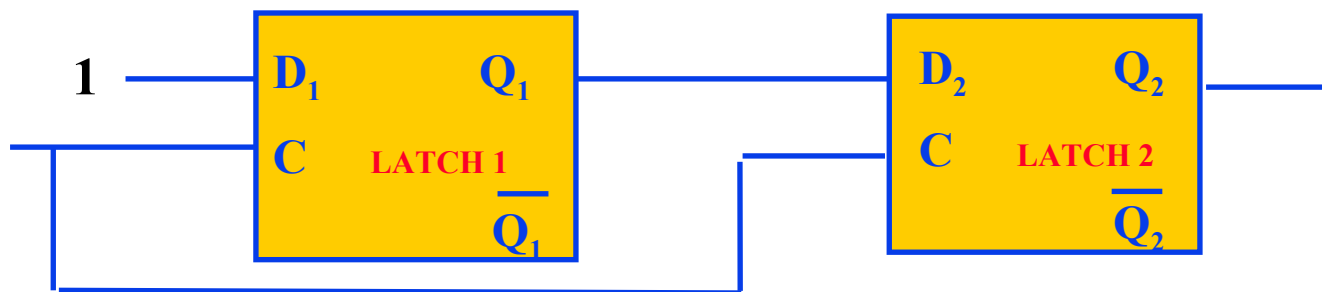
Tabela - verdade

C	D	Q_{n+1}
0	X	Q_n
1	0	0 RESET
1	1	1 SET

- latch D responde imediatamente a qualquer mudança na entrada enquanto $C = 1$, latch está aberto \Rightarrow **saída segue a entrada**
- latch D fica fechado (mantém estado) enquanto $C = 0$

\Rightarrow o latch é **sensível ao nível** do sinal de controle (sensível à fase do sinal de controle C (clock))

problema causado pela transparência



situação inicial

$$Q_1 = 0, \overline{Q_1} = 1$$

$$Q_2 = 0, \overline{Q_2} = 1$$

- aplica-se $D_1 = 1$ na entrada

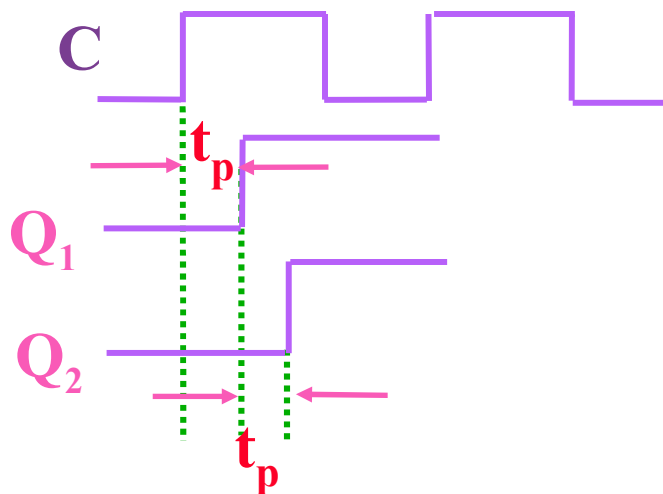
- o que se deseja é

$$Q_1 = 1, \overline{Q_1} = 0 \quad \text{após primeiro pulso de controle}$$

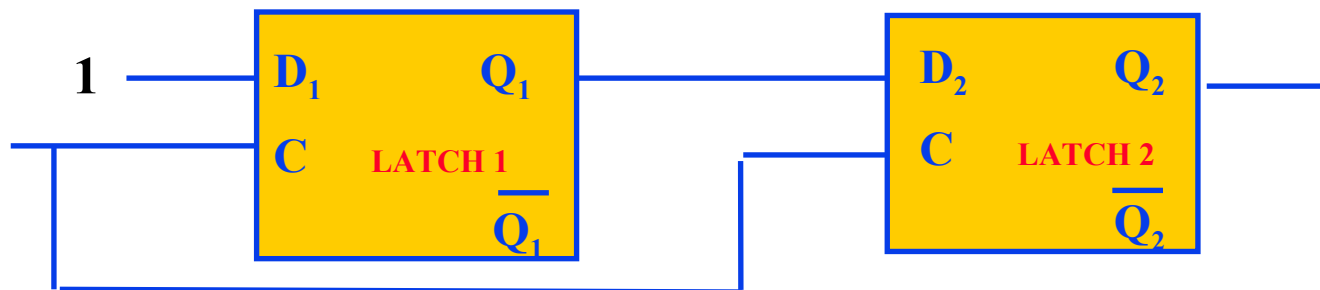
$$Q_2 = 1, \overline{Q_2} = 0 \quad \text{após segundo pulso de controle}$$

- o que ocorre realmente

Q_2 também pode ir para 1 ainda no primeiro pulso do sinal de controle

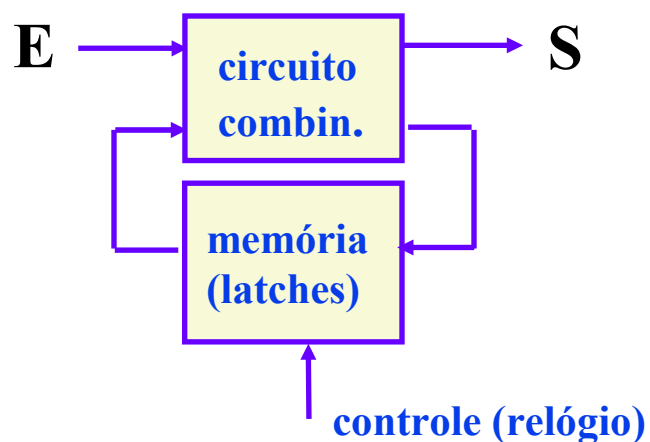


t_p : tempo de atraso entre S e Q_1
no interior do latch
(= atraso de 2 portas)



- ou seja: enquanto $C = 1$ e os latches estiverem “abertos”, o valor de entrada (D_1) se propaga através de todos os latches
- solução aparente
 - fazer pulso de controle bem estreito
 - impraticável distribuir uniformemente um pulso estreito por um circuito muito grande, sem que ele sofra distorções devidas aos atrasos

• generalizando o problema: circuitos síncronos



enquanto controle = 1, devido à realimentação, novos valores nas saídas dos latches (próximo estado) podem passar pela lógica combinacional e afetar novamente as entradas dos latches, alterando o valor esperado do estado ➡ **RACE !!!**

5. Soluções possíveis

- não controlar todos os latches pelo mesmo sinal de controle
 - usar relógio de várias fases

- não usar latches e sim **flip-flops na construção de máquinas com diversos elementos de memória.**
 - flip-flop mestre-escravo (*master-slave*) isola saída da entrada
 - entrada afeta estado num primeiro momento
 - estado afeta saída num segundo momento

 - flip-flop sensível à borda (*edge-triggered*)
 - alteração no estado é sensível à transição do sinal de controle, e não ao nível