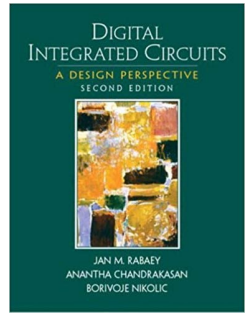
Microeletrônica

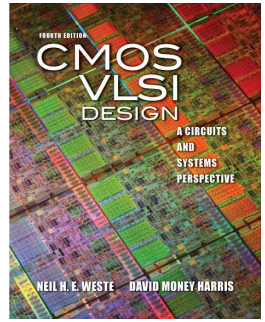
Aula #7 → Circuitos sequenciais estáticos

- Professor: Fernando Gehm Moraes
- Livro texto:

  Digital Integrated Circuits a Design Perspective - Rabaey

  C MOS VLSI Design - Weste



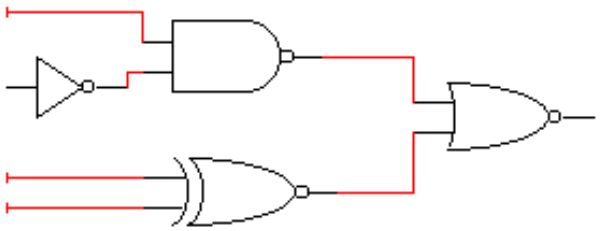


Revisão das lâminas: 09/maio/2025

Lógica Sequencial

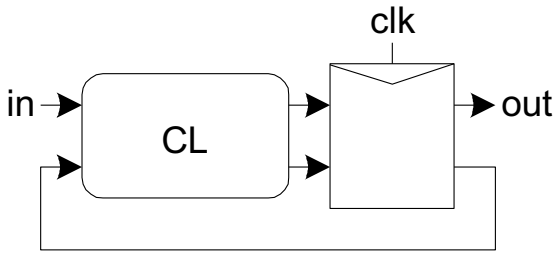
Lógica Combinacional

- Os sinais de saída são resultados de uma combinação lógica dos
- sinais de entrada atuais

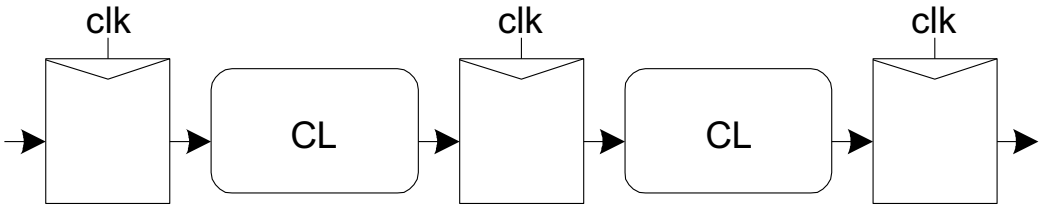


Lógica Sequencial

- Os sinais de saída são resultados não apenas dos sinais de entrada atuais, mas também de sinais anteriores
- Flip Flops são elementos de memória que registram o estado do circuito

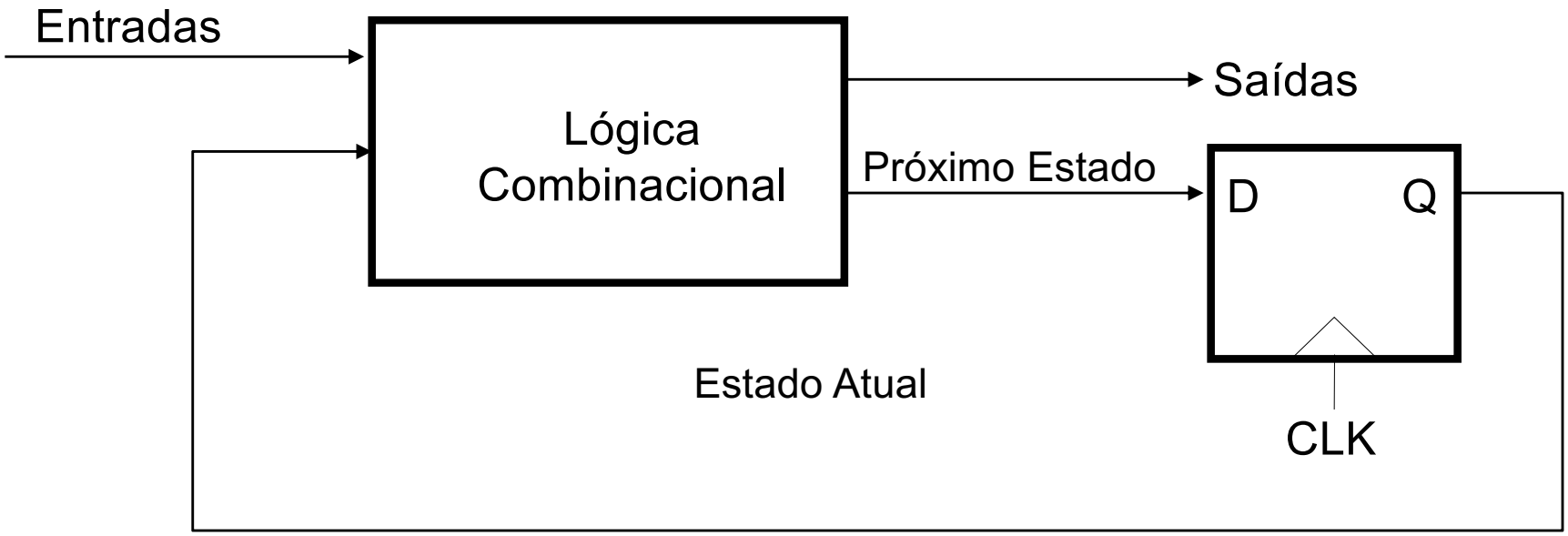


Finite State Machine



Pipeline

Lógica Sequencial



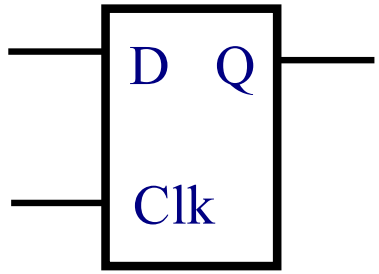
2 mecanismos de armazenamento

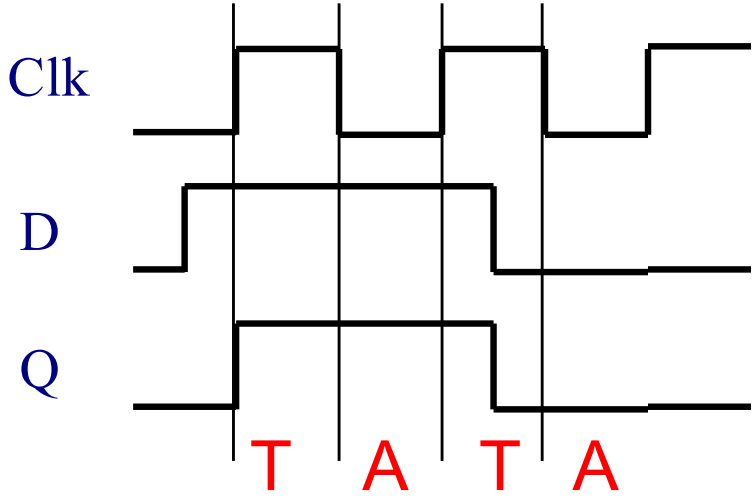
- **positive feedback**
- **charge-based**

Latch versus Registrador

- **Latch**

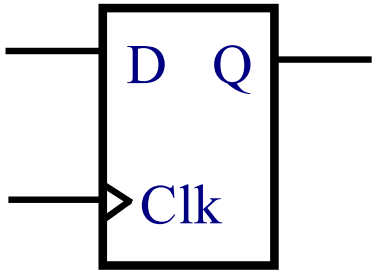
  transparente em um nível do clk e depois mantém o dado

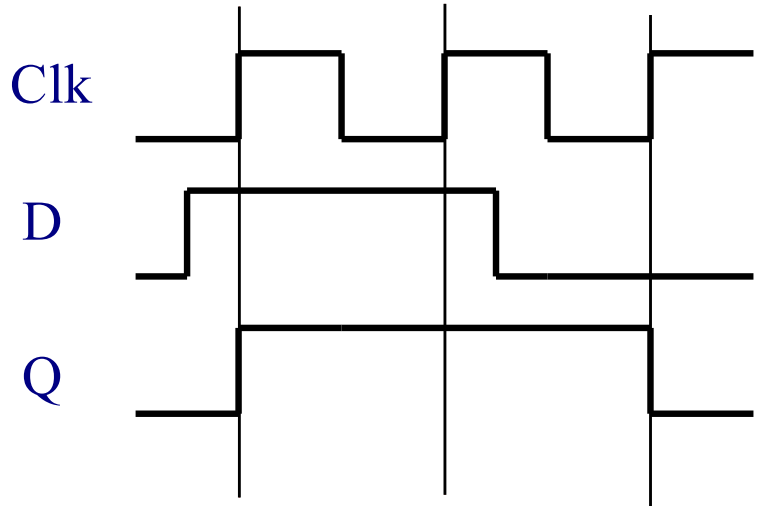




- **Registrador (Mestre-Escravo)**

  armazenamento no momento de uma **borda**





T→ transparente A→ armazena

Latches

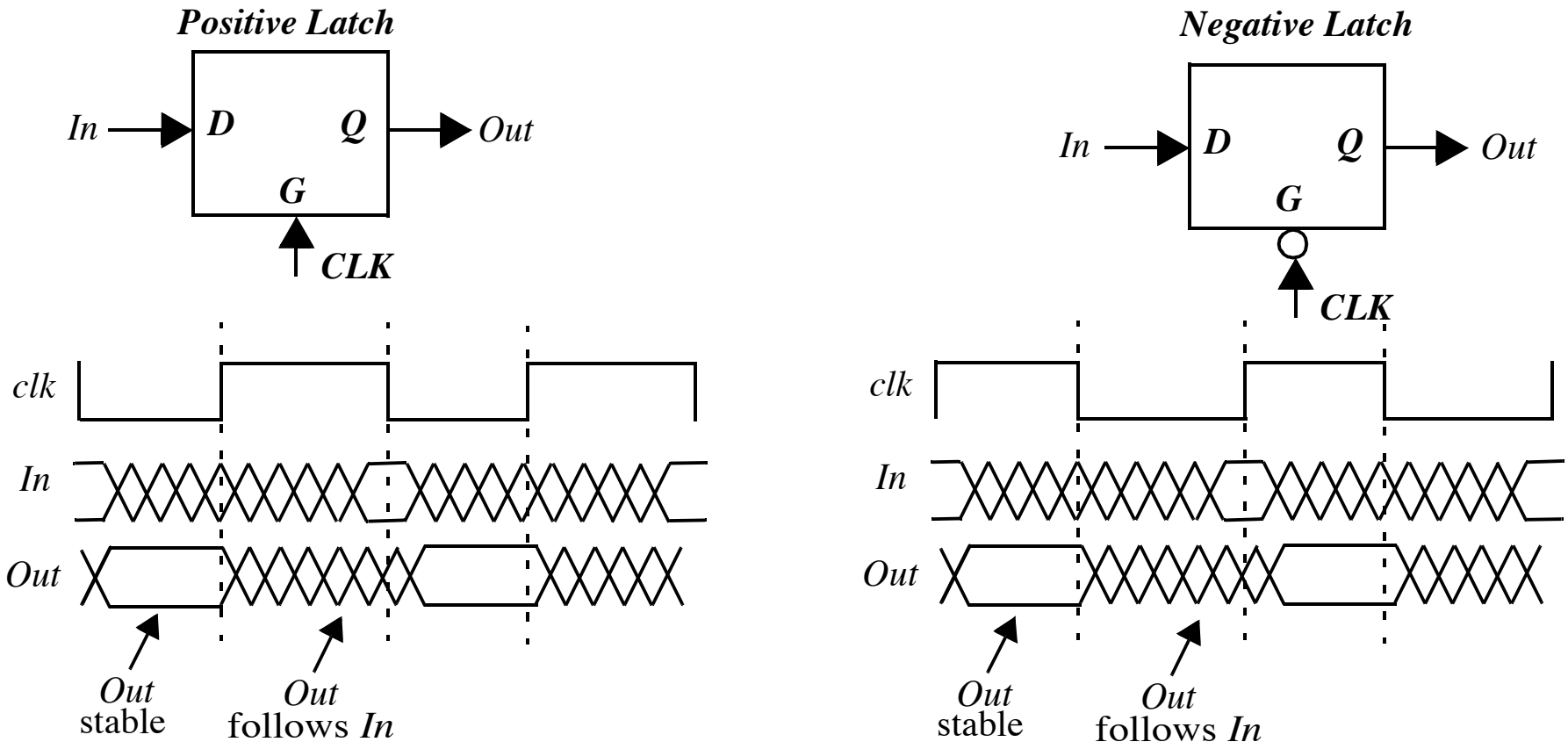


Figure 7.3 Timing of positive and negative latches.

Latches

- Memórias **estáticas** utilizam realimentação positiva para criar um circuito bistável (***bistable***) — um circuito com dois estados estáveis que representam 0 e 1
- Apenas os pontos A e B são estáveis para operação, enquanto o ponto C é um ponto de operação metaestável
- Característica que assegura que o circuito alterne de forma estável entre os estados que representam 0 e 1
- O ponto metaestável C é um estado não desejável, mas sua ocorrência é minimizada quando o ganho do inversor é adequadamente alto na região transitória

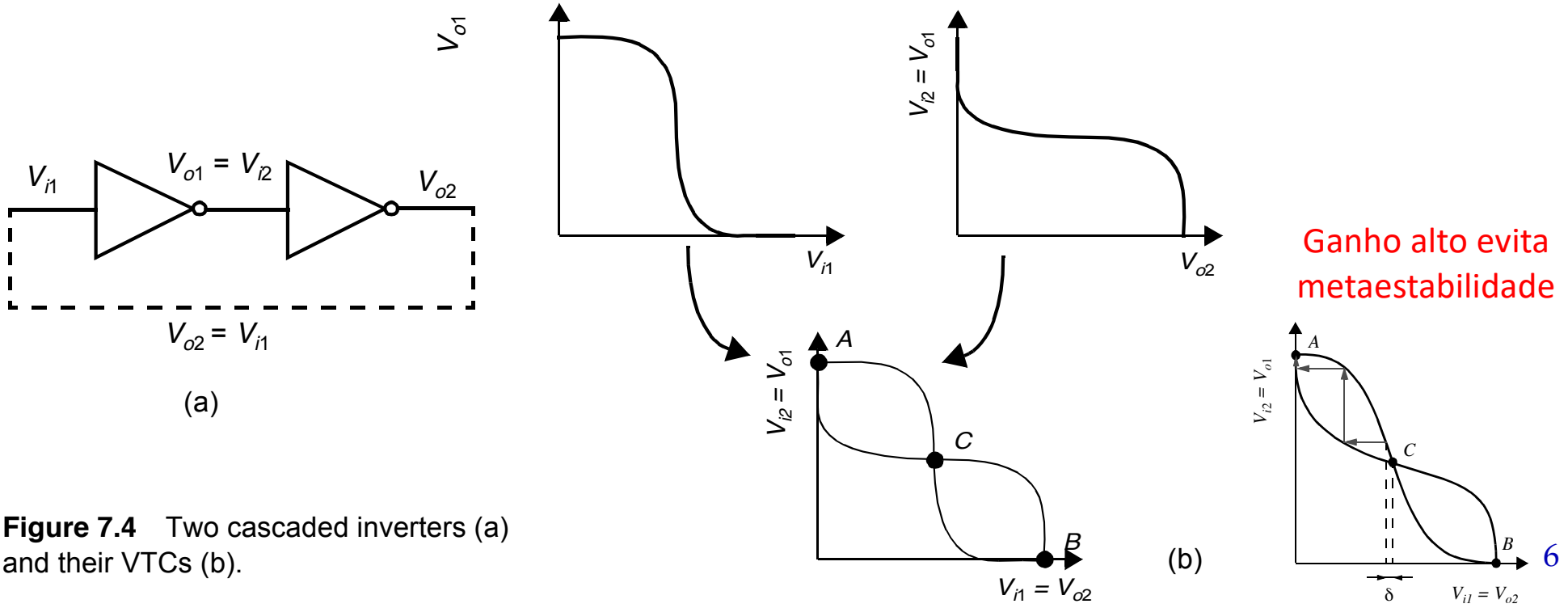
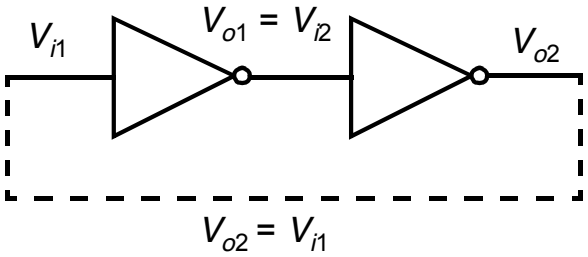


Figure 7.4 Two cascaded inverters (a) and their VTCs (b).



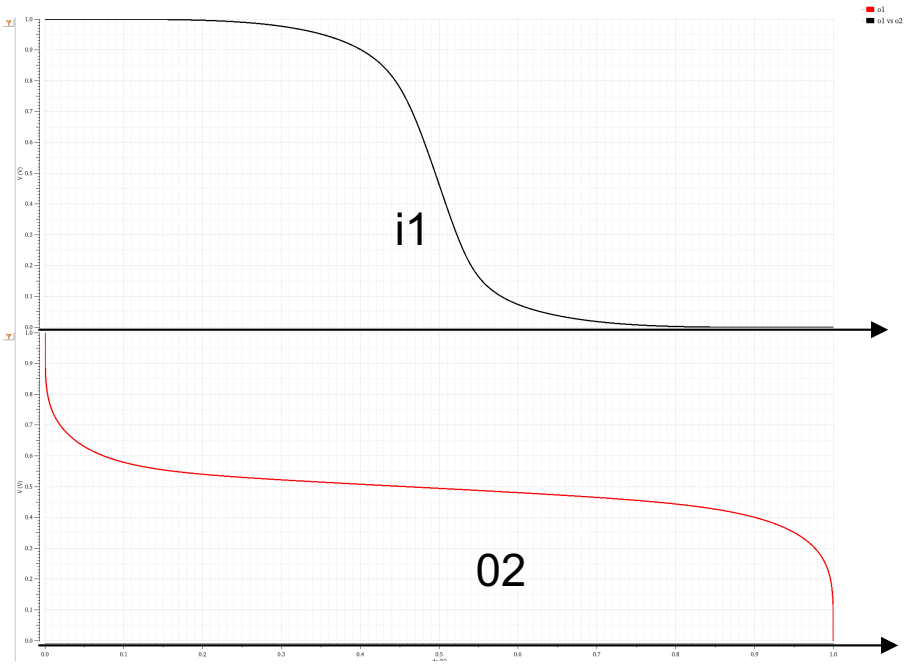
O SNM (static noise margin) é definido como a quantidade máxima de tensão de ruído que pode ser introduzida nas entradas dos inversores de forma que a célula retenha seus dados.

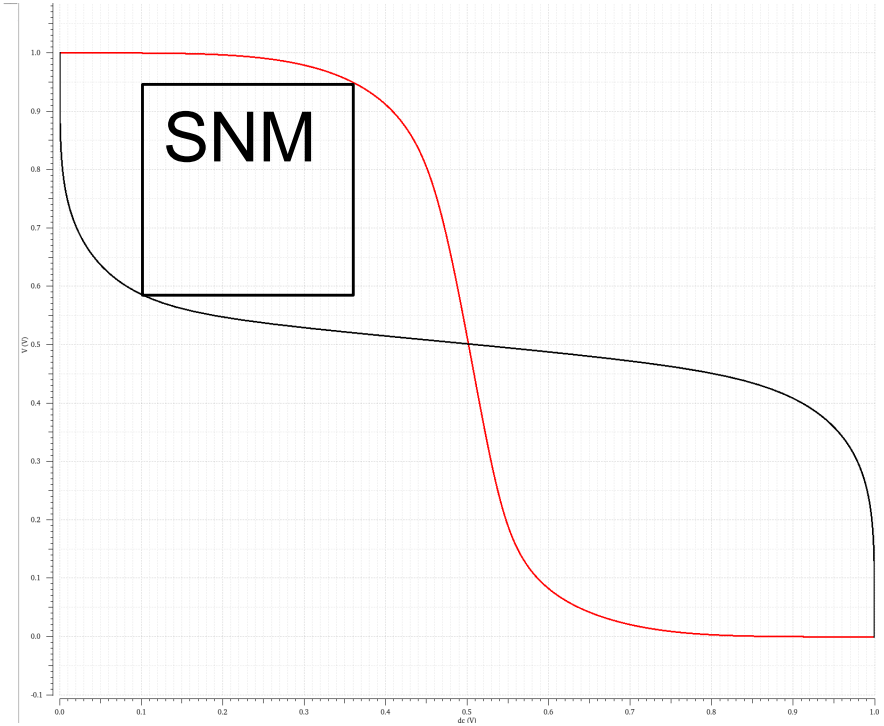
O SNM é definido como o comprimento do lado do maior quadrado que pode ser inserido dentro dos lóbulos do curva de borboleta.

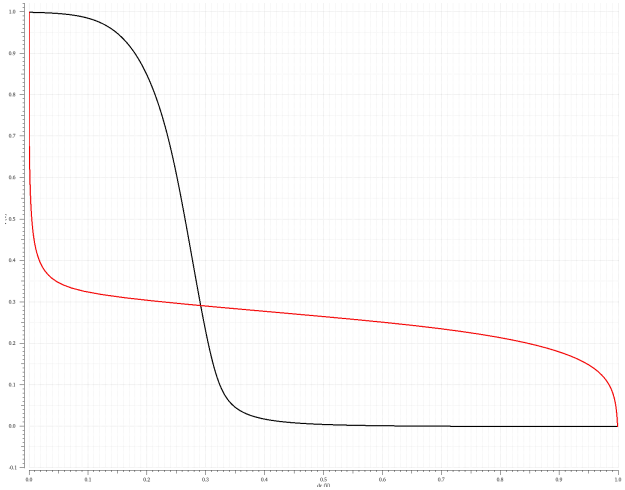
o1

o1=i2

i2







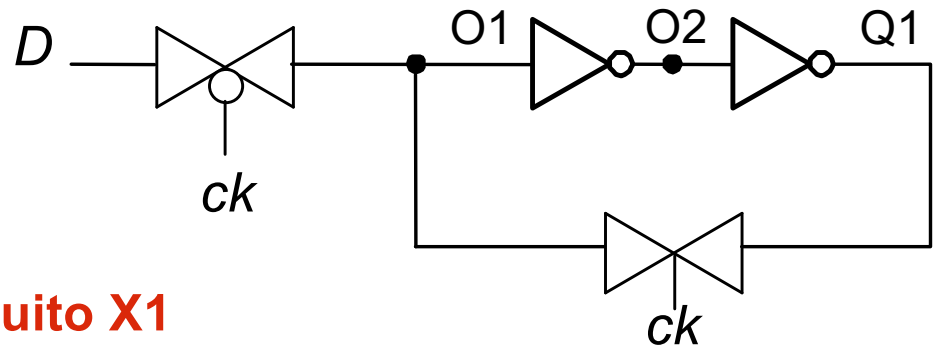
Transição na meia excursão de vcc

Melhor estabilidade da célula de memória

Transistores mal dimensionados

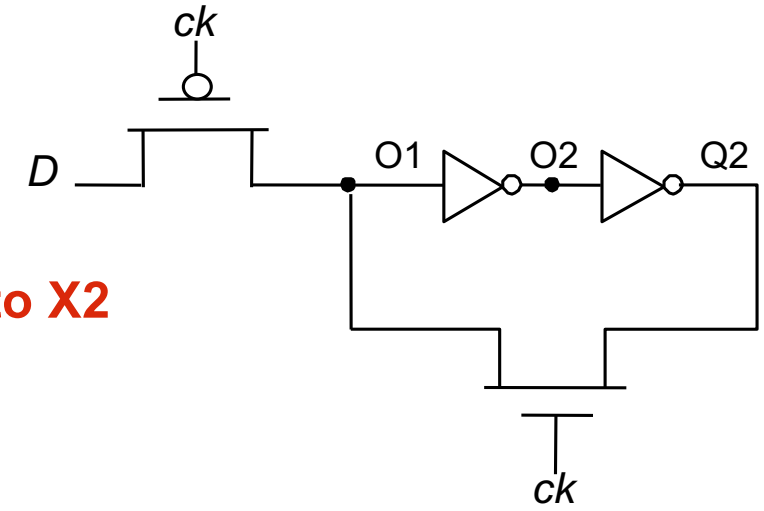
Latches - implementações

Com portas de transmissão



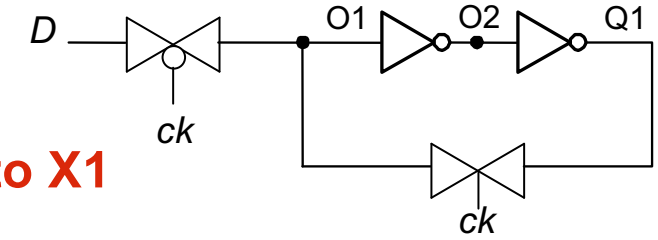
Circuito X1

Com transistores (N e P)

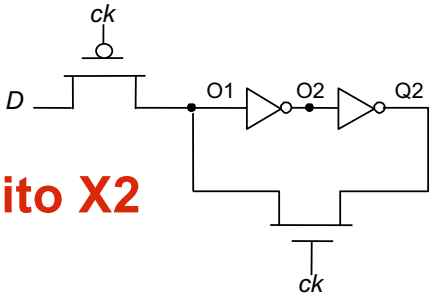


Circuito X2

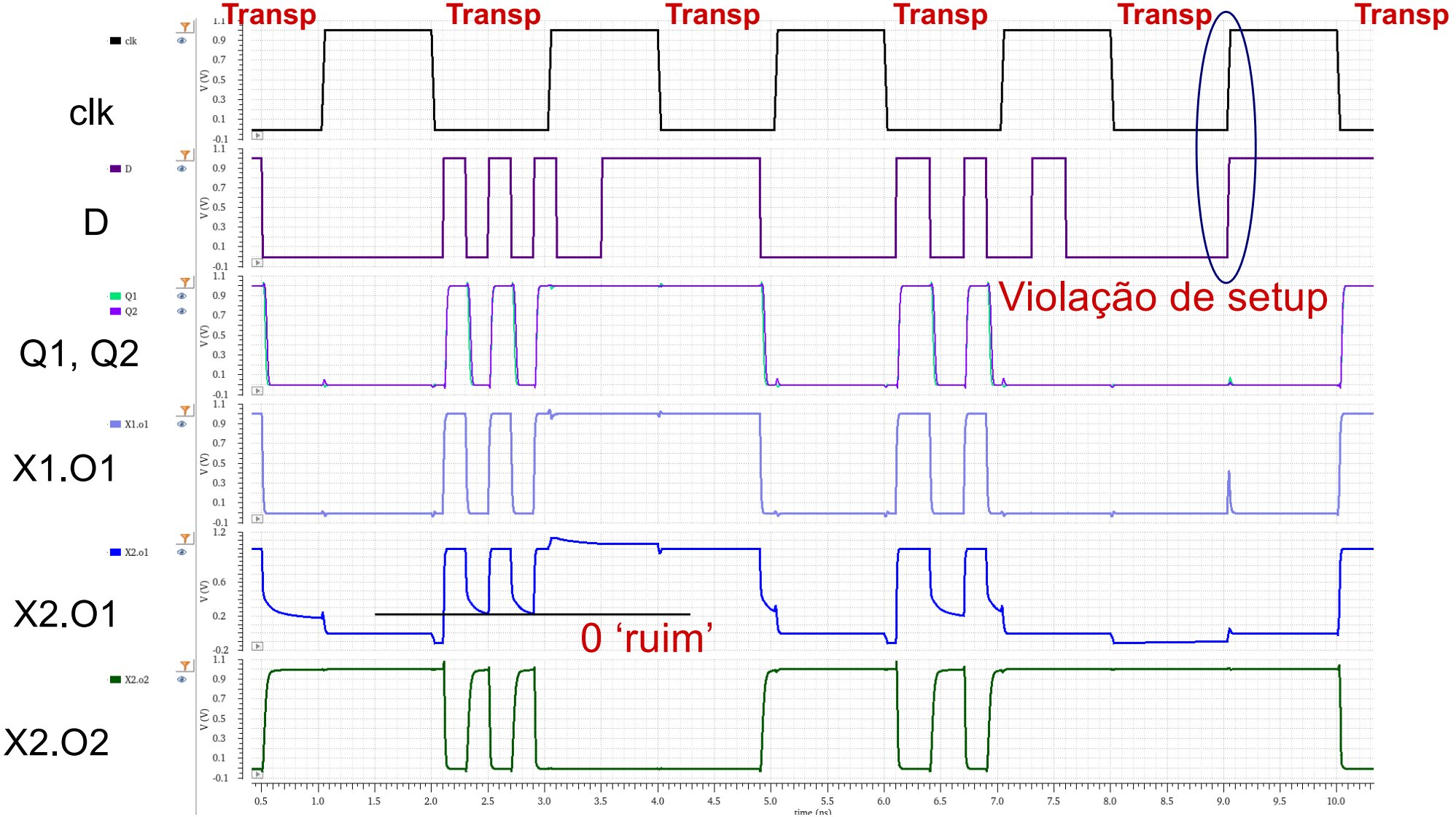
Tempo de setup: duração necessária para que os dados de entrada devam estar estáveis antes da borda do *clock*. Se os dados mudarem dentro dessa janela de tempo de configuração, os dados de entrada podem ser perdidos e não armazenados na latch/MS.



Circuito X1



Circuito X2



clk

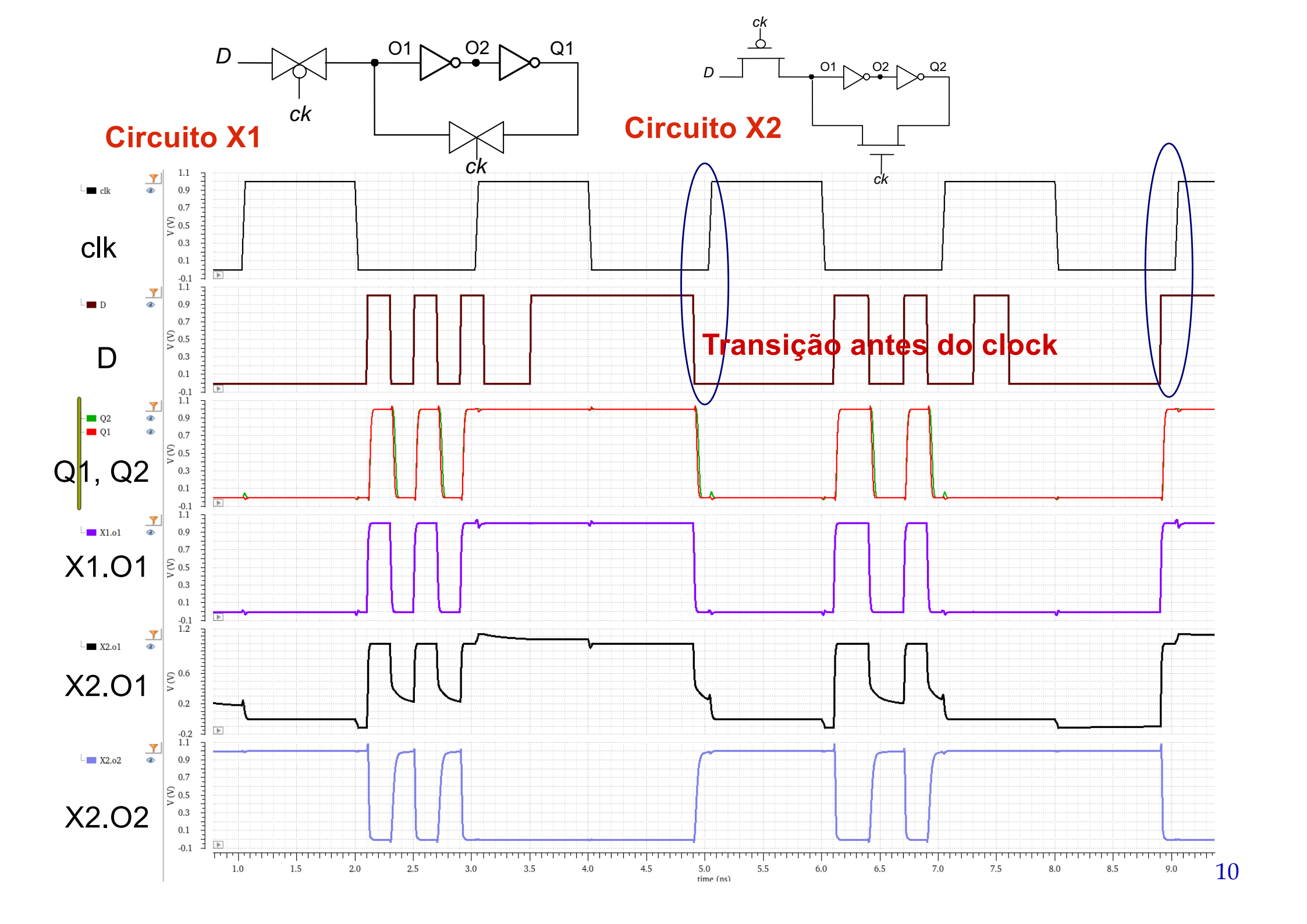
D

Q1, Q2

X1.O1

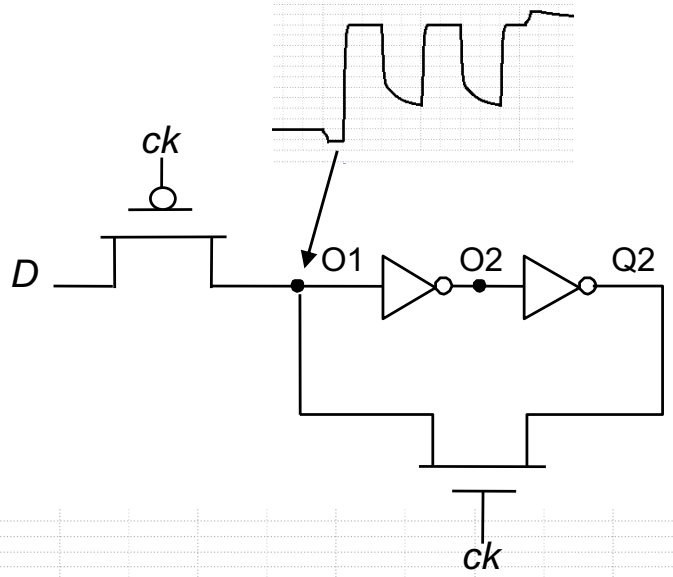
X2.O1

X2.O2


Circuito X1
D
ck
O1
O2
Q1
ck
Circuito X2
ck
D
O1
O2
Q2
ck
clk
D
Q2
Q1
Q1, Q2
X1.o1
X1.O1
X2.o1
X2.O1
X2.o2
X2.O2
V (V)
time (ns)
Transição antes do clock

Saída Q2: depende de um transistor P em modo transparente

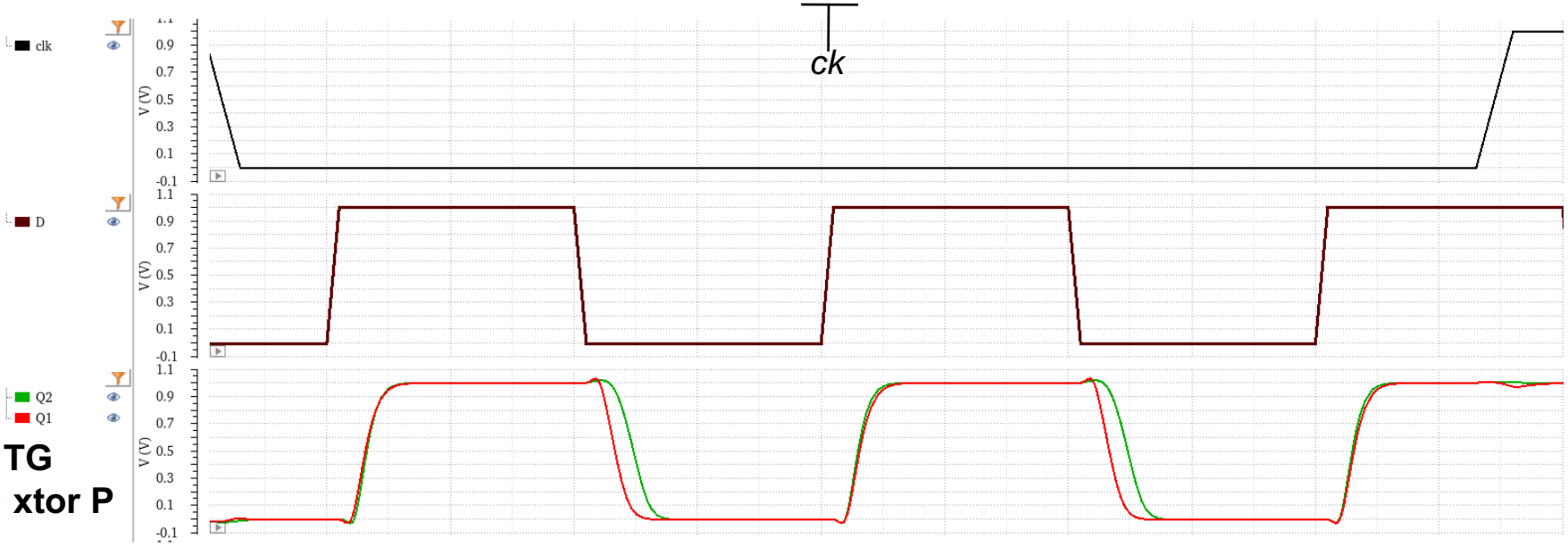
Mais lento no tempo de descida (0 ruim em 01, 02 lento para subir, Q2 desce mais lentamente)



CK

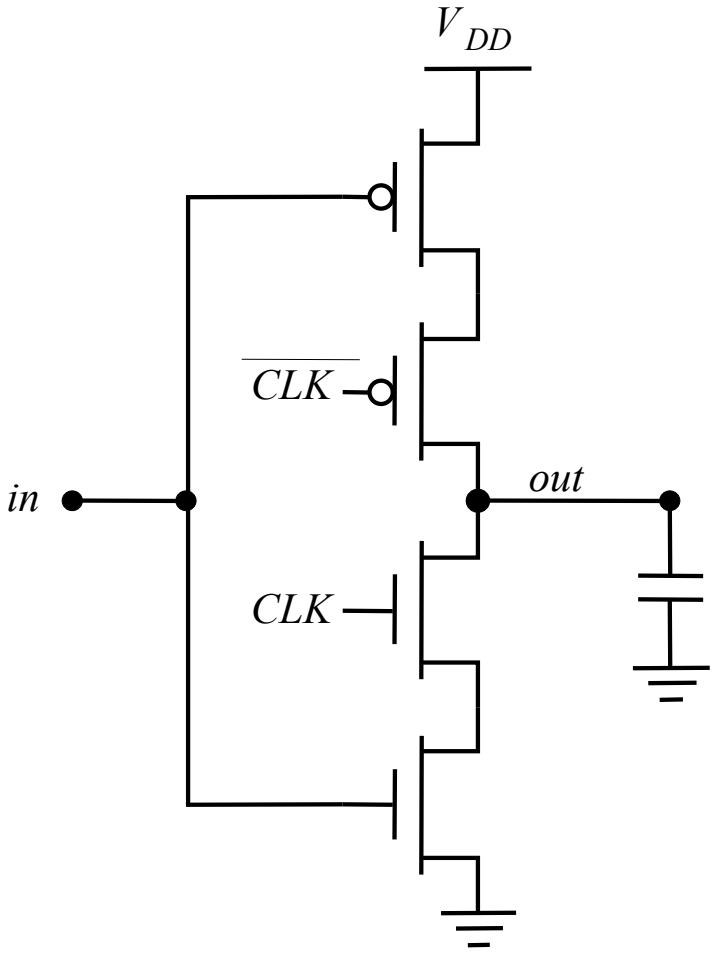
D

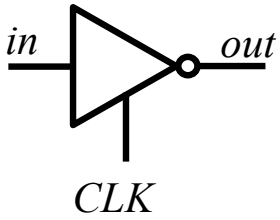
Q1→ TG
Q2 → xtor P



Tri-state buffer

Integra o TG ao inversor
Reduz sensibilidade de propagação de ruído entre a saída e a entrada

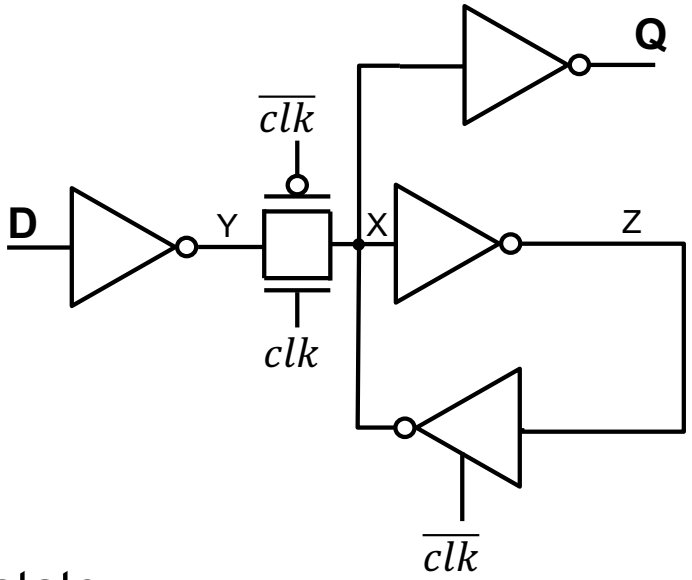




Latches – com buffer tri-state

Amplamente utilizado em standard-cells

Robustez: saída Q é gerada usando um buffer separado, o que elimina problemas com *backdriving* (informação da saída propagada para a entrada)



A saída Q é mantida pelo loop que inclui o buffer tristate

☹ Os inversores de entrada e saída aumentam a área do circuito

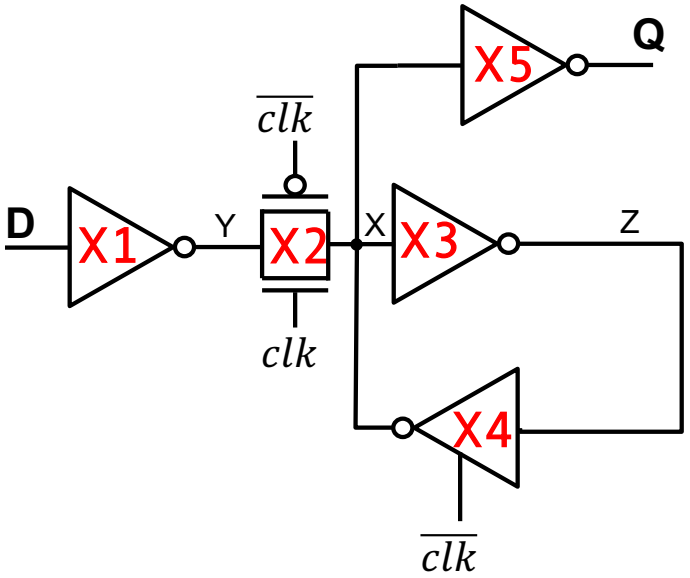
Latches – com buffer tri-state

```
.subckt inv in out  vcc
MP1 out in vcc  vcc   psvtgp w=wp l=0.06
MM2 out in 0    0     nsvtgp w=wn l=0.06
.ends inv

.subckt inv_tg in ck nck out  vcc
MP1 1   in  vcc  vcc   psvtgp w=wp l=0.06
MP2 out nck 1    vcc   psvtgp w=wp l=0.06
MM1 out ck  2    0     nsvtgp w=wn l=0.06
MM2 2   in  0    0     nsvtgp w=wn l=0.06
.ends inv_tg

.subckt tg a b nb out vcc
MP1 a  nb out vcc   psvtgp w=wp l=0.06
MN1 a   b out 0     nsvtgp w=wn l=0.06
.ends tg
```

```
.subckt latch D Q ck nck vcc
X1 D   Y             vcc inv
X2 Y   ck   nck  X vcc tg
X3 X   Z             vcc inv
X4 Z   nck   ck  X vcc inv_tg
X5 X   Q             vcc inv
.ends latch
```



Mux-based latches

- A realimentação não precisa ser sobrescrita para gravar na latch
- Maior velocidade, maior área

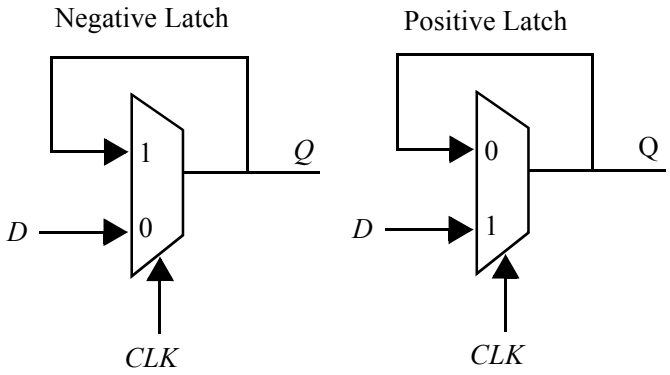


Figure 7.11 Negative and positive latches based on multiplexers.

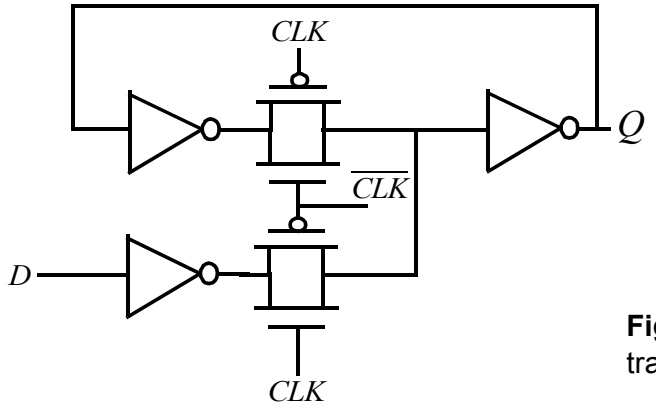
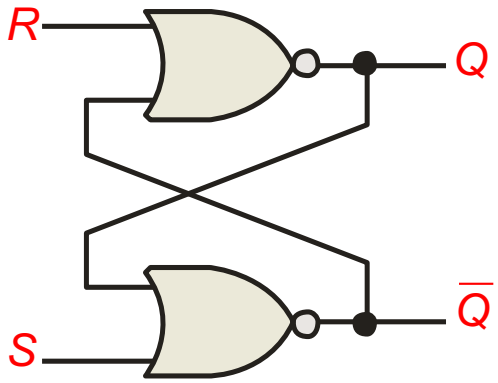
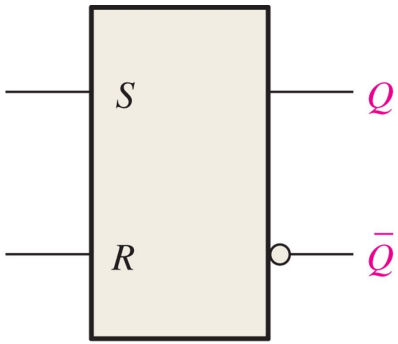


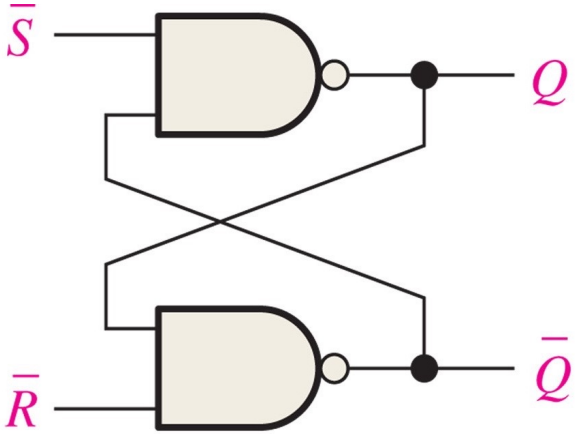
Figure 7.12 Positive latch built using transmission gates.

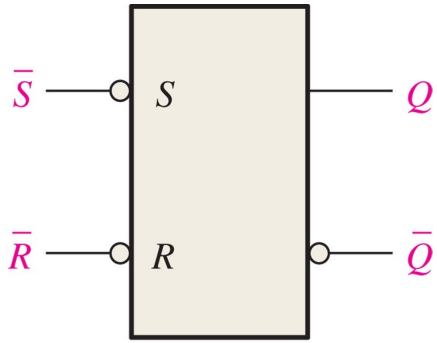
Elementos de Memória

SR - Flip Flop (latch)







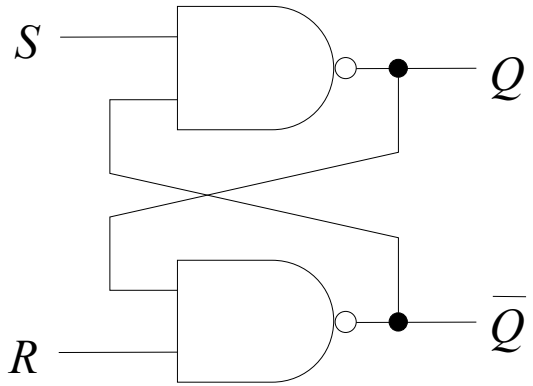


Inputs		Outputs	
$\overline{S}$	$\overline{R}$	Q	$\overline{Q}$
1	1	NC	NC
0	1	1	0
1	0	0	1
0	0	1	1

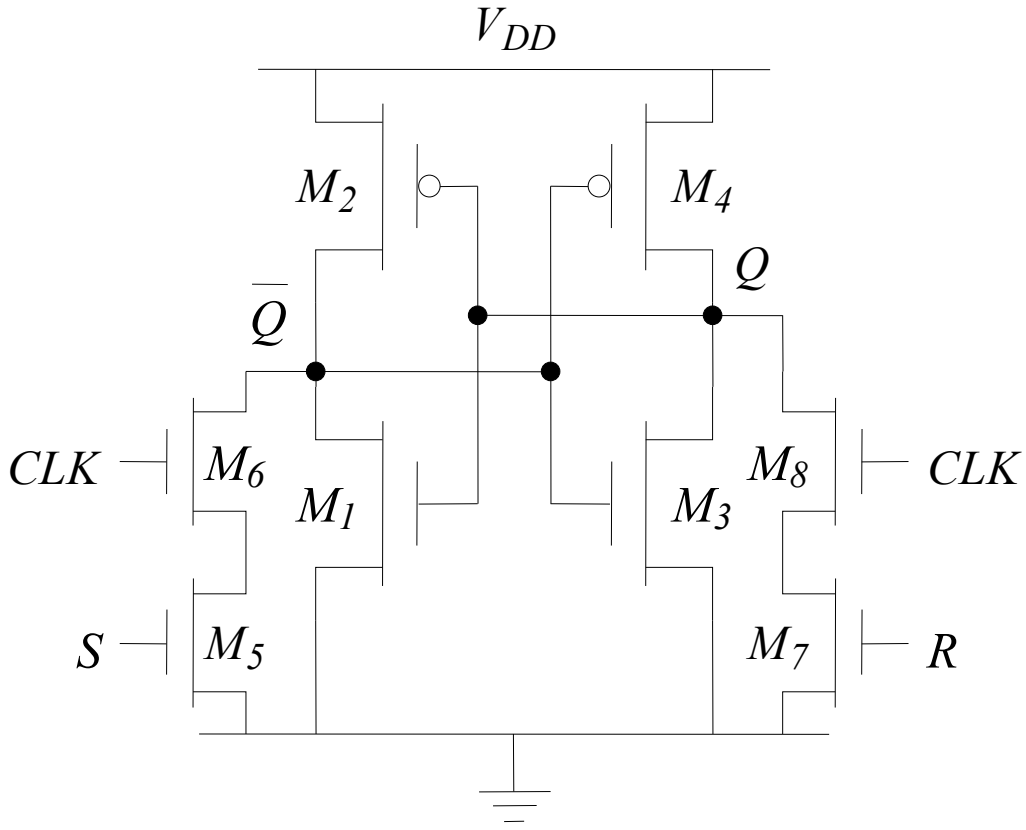
Diagrama Esquemático Símbolo Lógico

Cross-Coupled NAND

Cross-coupled NANDs



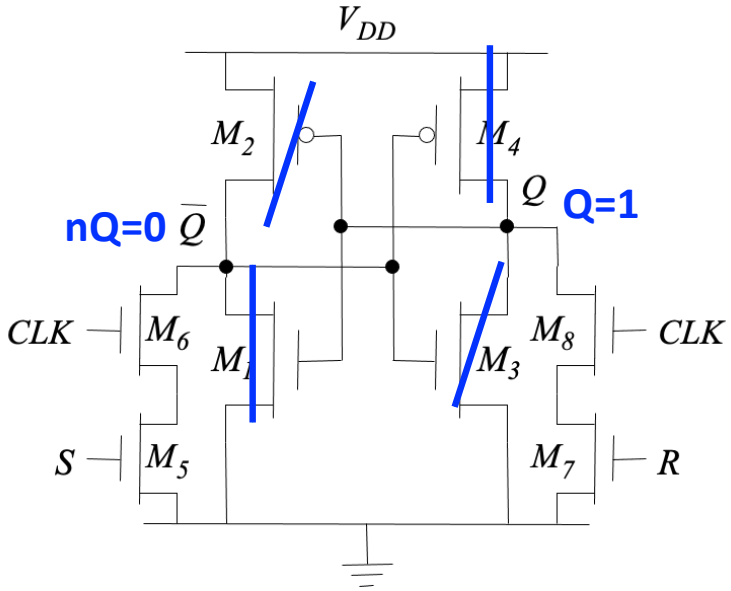
Added clock



CLK=0 → não muda de estado

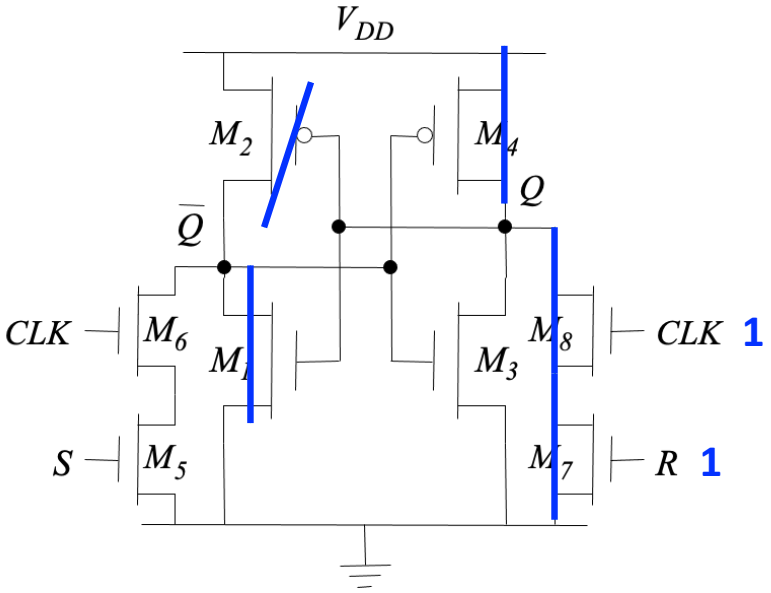
- M1-M2 inversor e M3-M4 inversor
- logo, 2 inversores realimentados

Cross-Coupled NAND



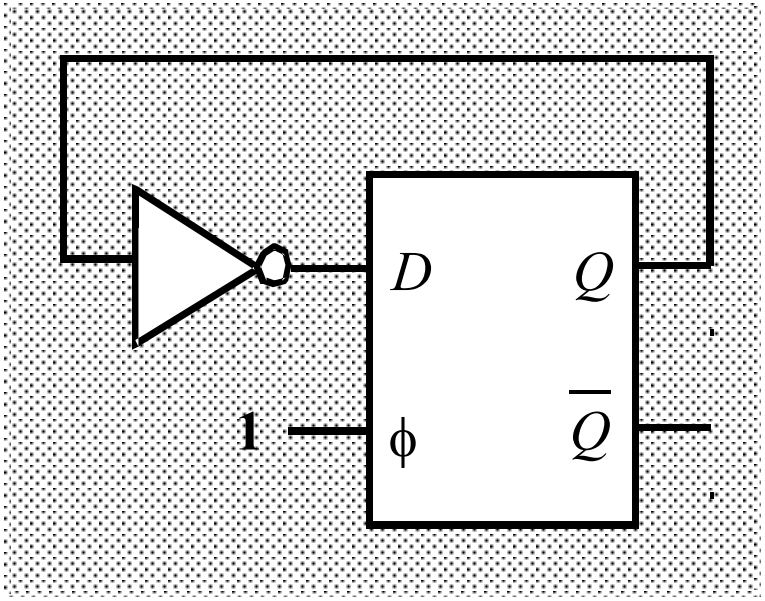
Q=1

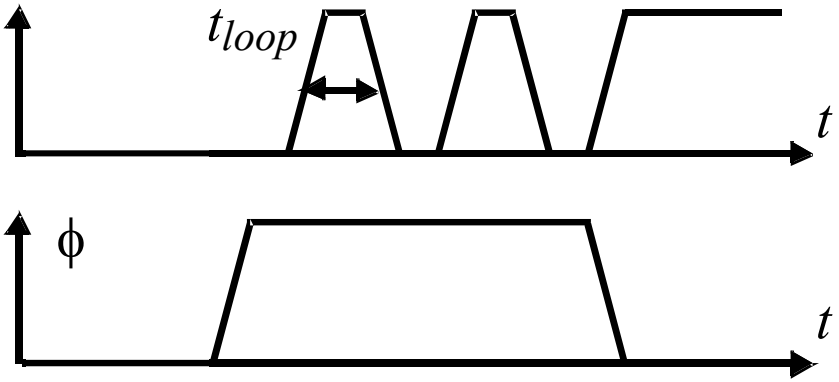
Considere o caso em que Q é **alto** e um pulso **R** é aplicado (CLK=1):



- M4, M8 e M7 → curto (*rationed inverter*)
- Para o circuito operar: maior ganho (W) em M5, M6, M7 e M8.

Latch tem problema de *race* (corrida)

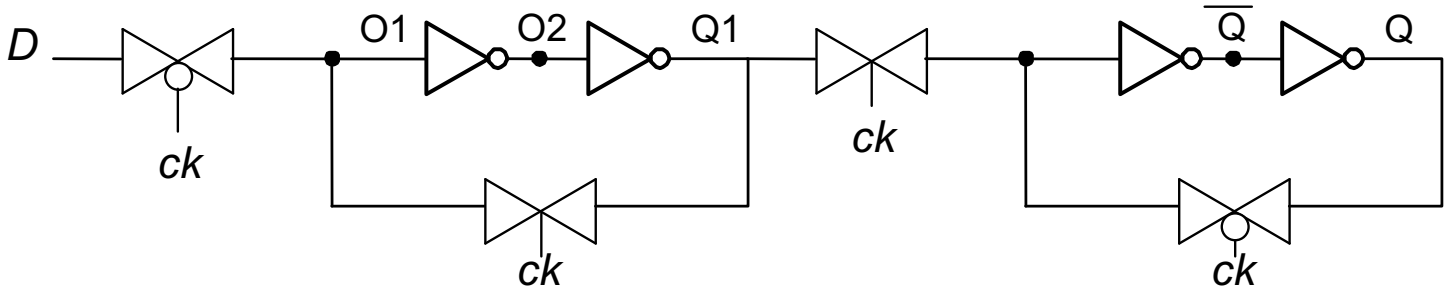




- A natureza transparente da *latch* causa o problema de 'race'
- A *latch* é **transparente** quando ϕ = 1
- Uma solução ao problema de *race* são os FF mestre-escravo

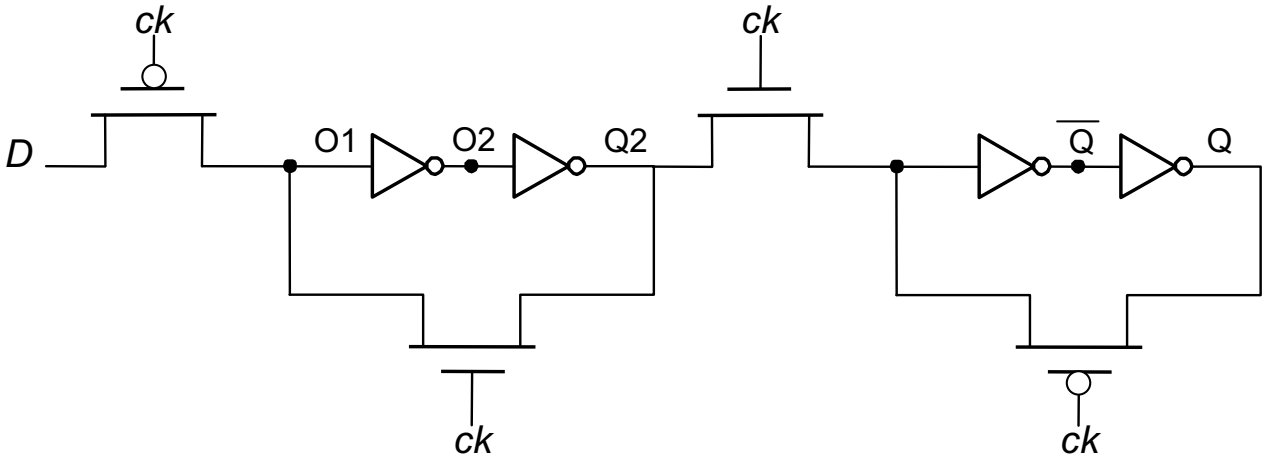
Flip Flop Mestre Escravo Estático

Com portas de transmissão



Circuito X1

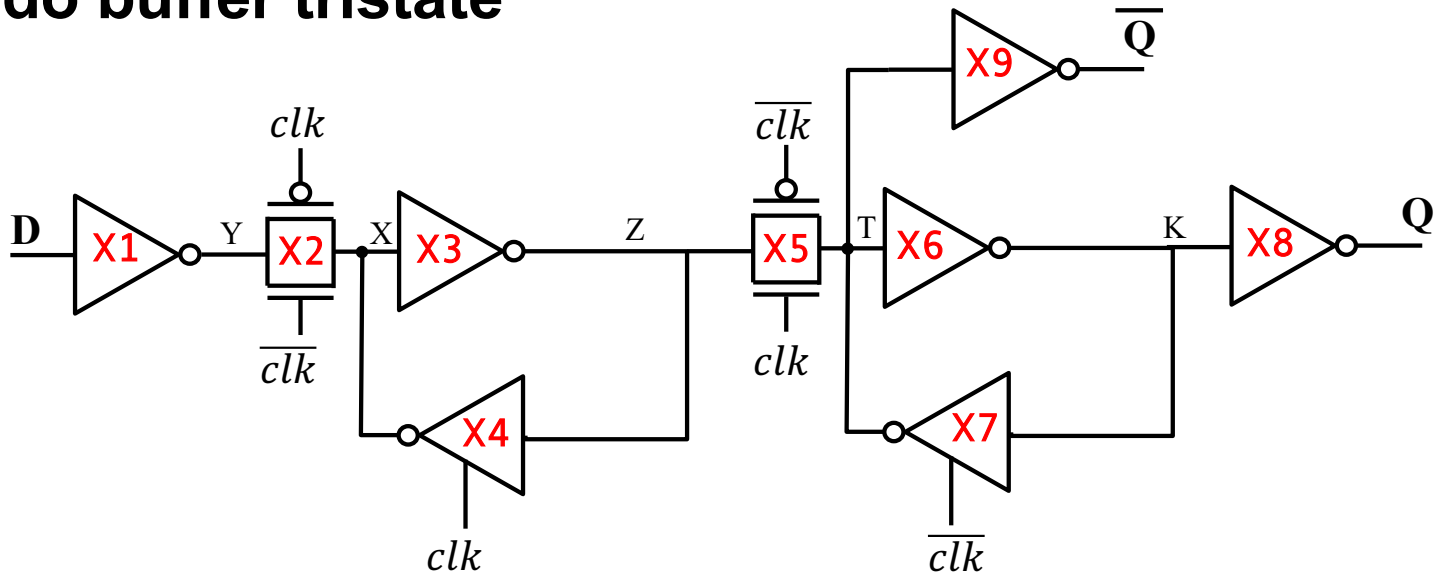
Sem portas de transmissão



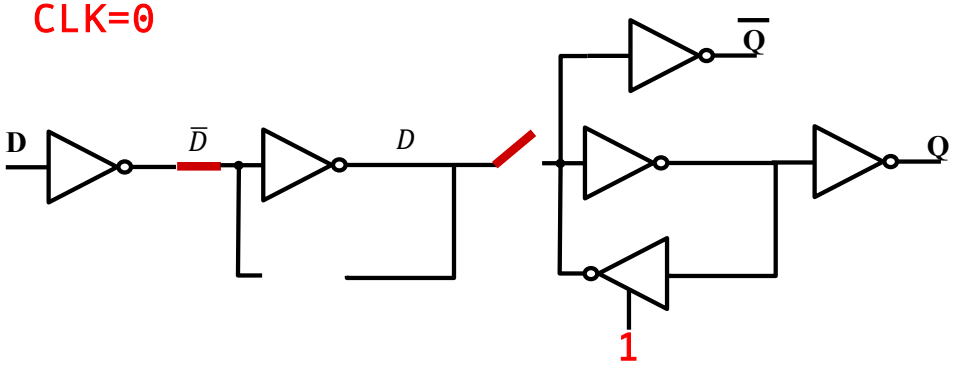
Circuito X2

Flip Flop Mestre Escravo Estático

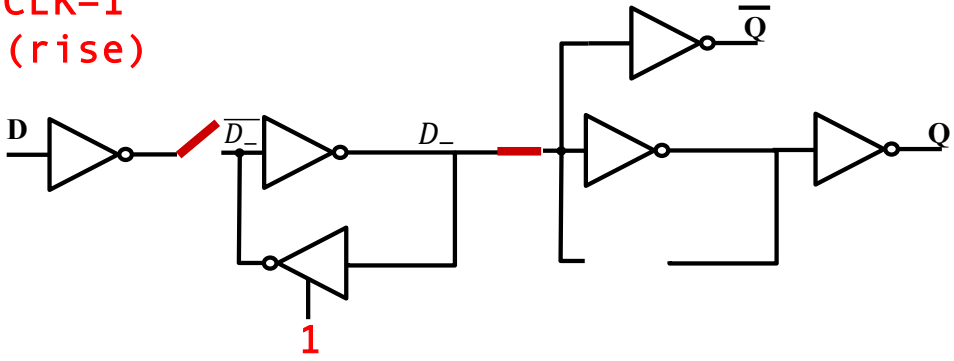
Utilizando buffer tristate



CLK=0



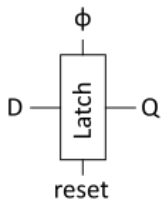
CLK=1
(rise)

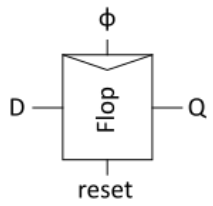


Flip Flop Mestre Escravo Estático

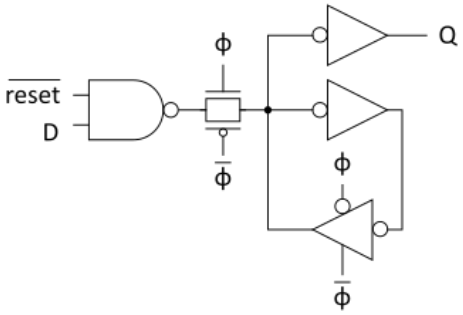
Reset

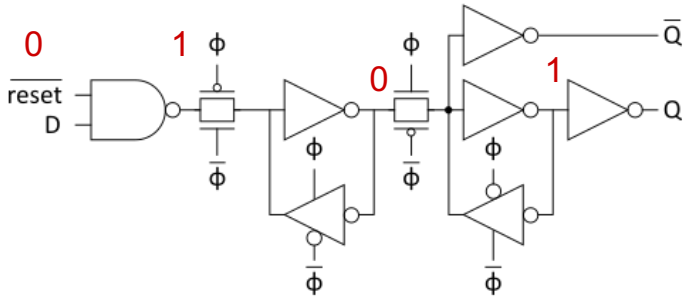
Symbol



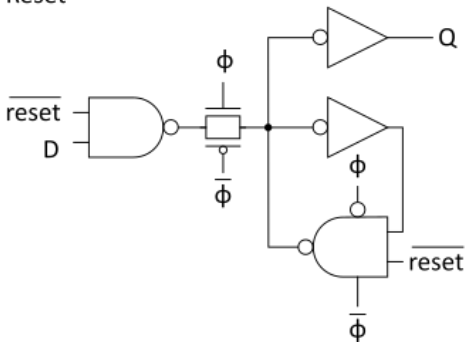


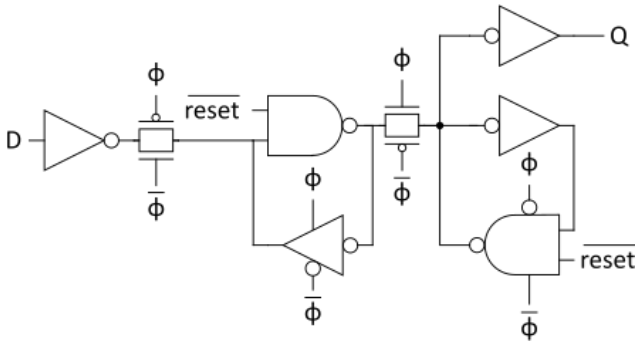
Synchronous Reset

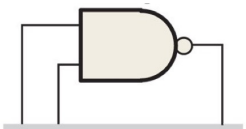




Asynchronous Reset







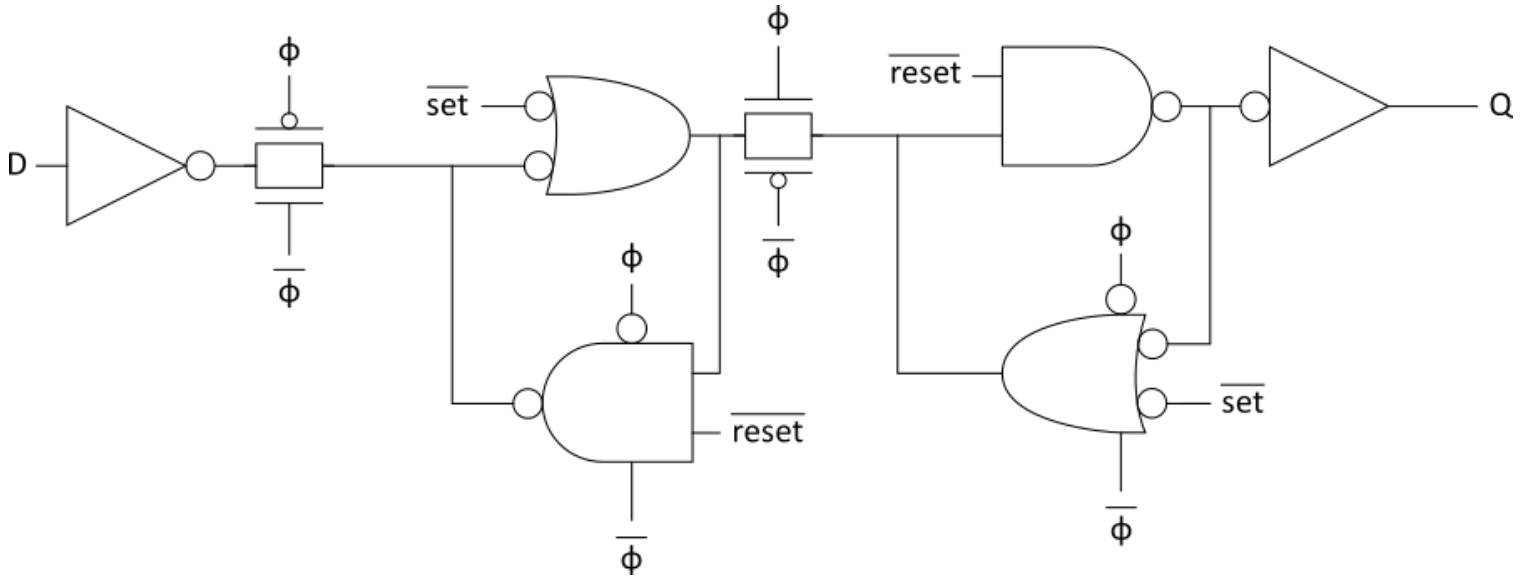
0	0	1
0	1	1
1	0	1
1	1	0

NAND

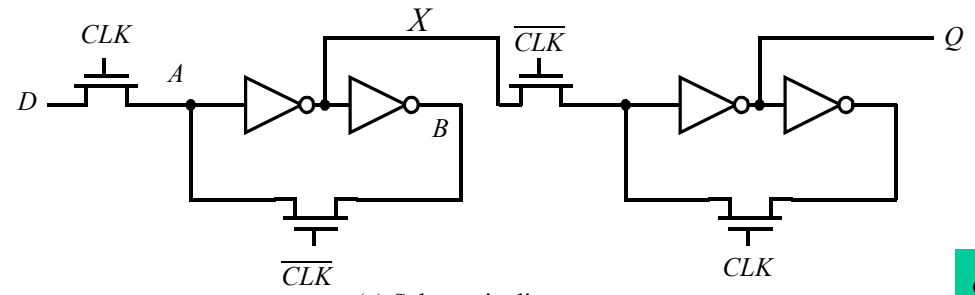
uma dada entrada em '0' saída em 1.
uma dada entrada em '1' comporta-se como um inversor.

Flip Flop Mestre Escravo Estático

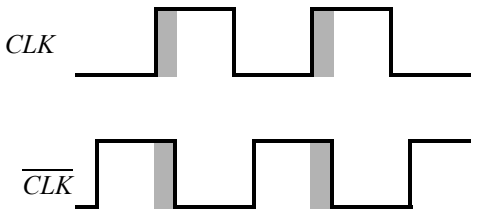
Flip-flop com set/reset assíncrono



Race em Mestre-Escravo



(a) Schematic diagram

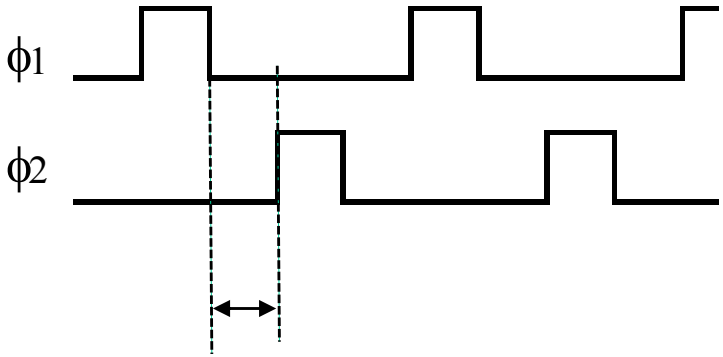


(b) Overlapping clock pairs

Figure 7.20 *Master-slave* register based on NMOS-only pass transistors.

Sobreposição do sinal do relógio pode causar
- **Sinais indefinidos na entrada e saída**
- **Condições de 'Race' (High por um período longo)**

Uma solução: clock sem sobreposição



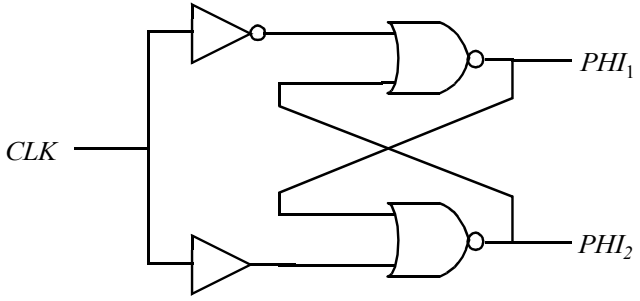
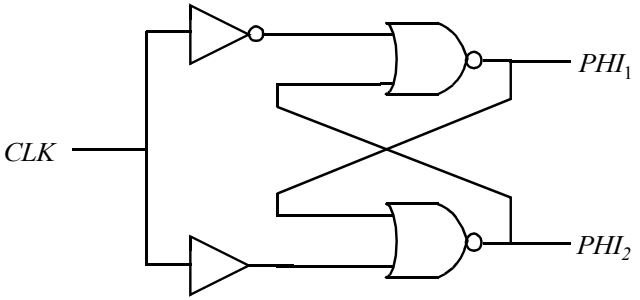
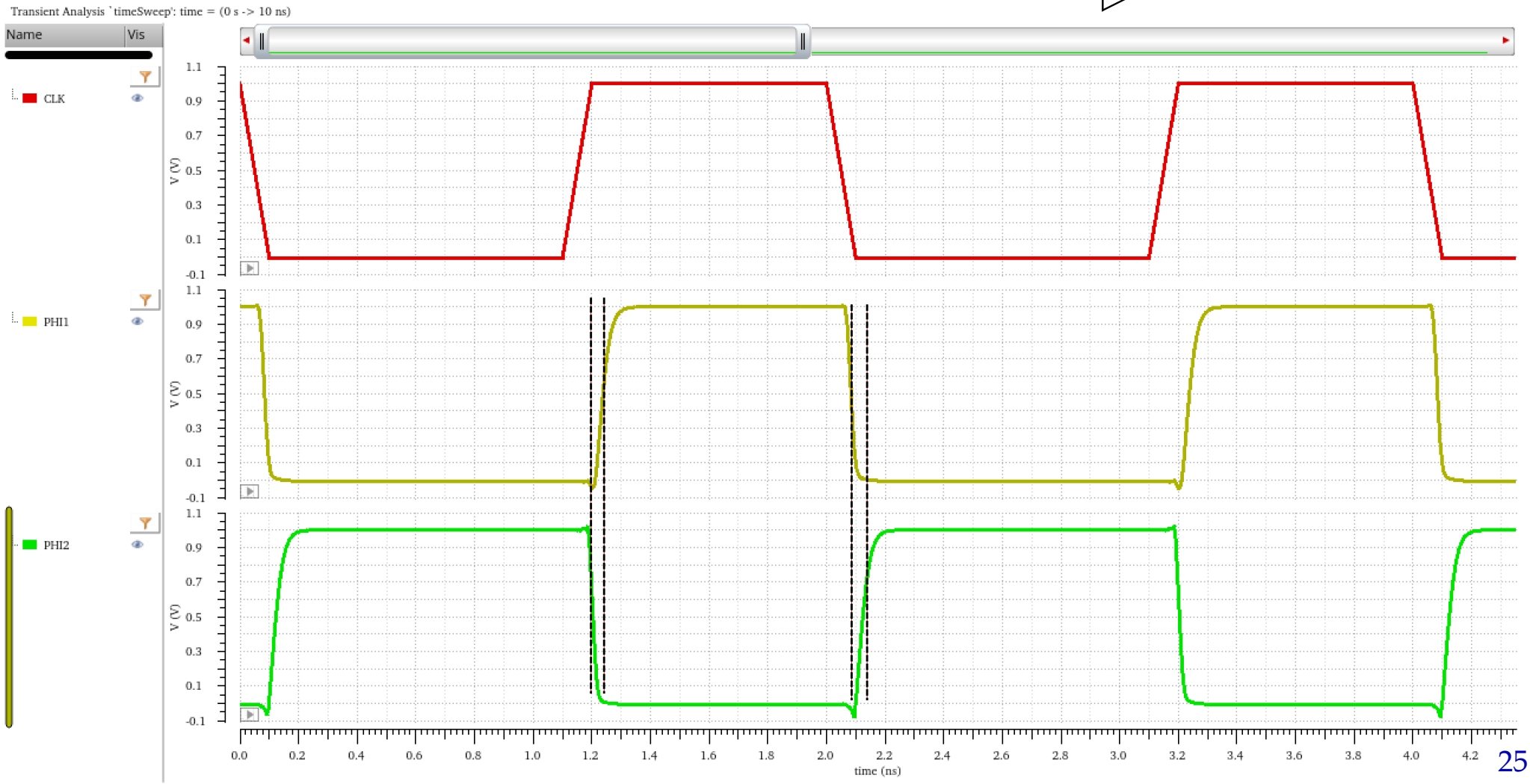


Figure 7.22 Circuitry for generating a two-phase non-overlapping clock.

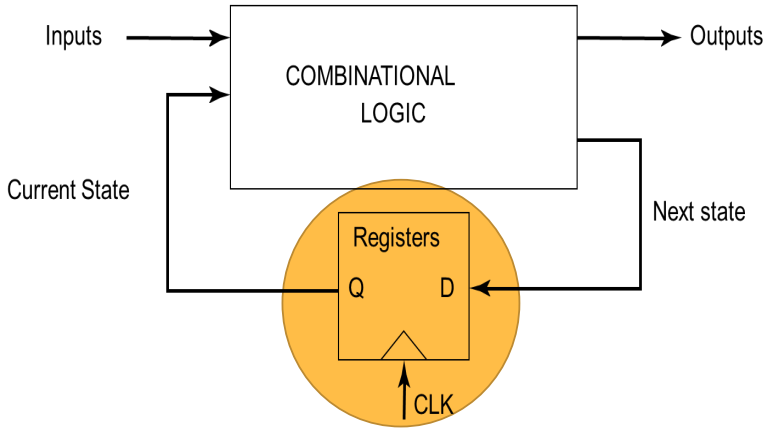
Fases não sobrepostas





Conceitos de temporização

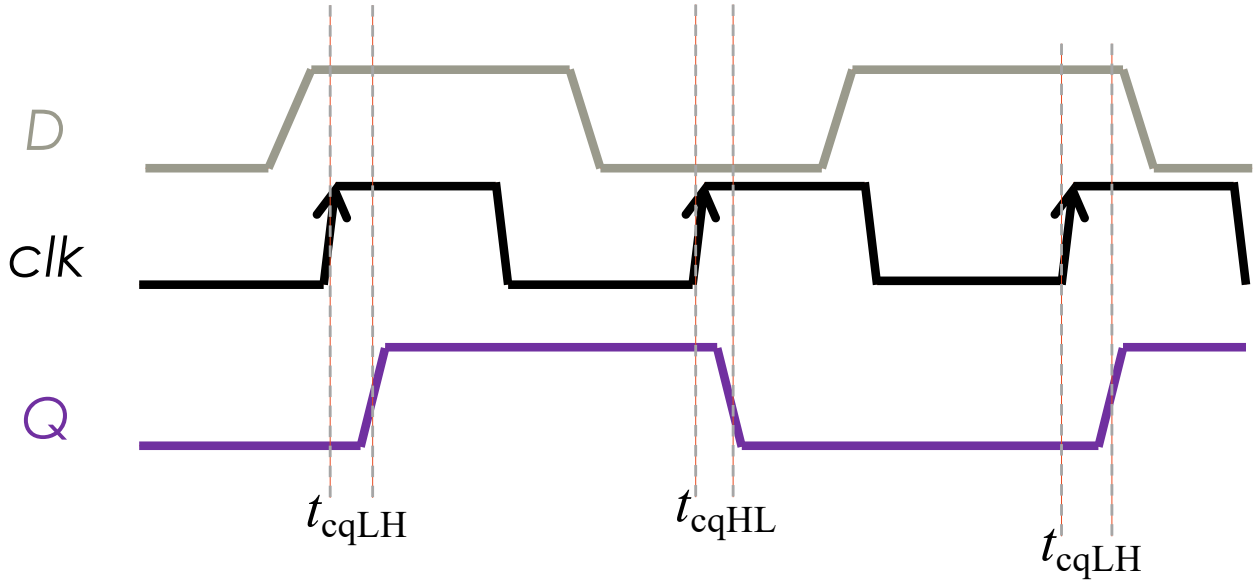
- A maioria dos projetos digitais são síncronos e construídos com elementos sequenciais
  - o projeto síncrono elimina condições de corrida
  - a técnica de **pipeline** aumenta a vazão

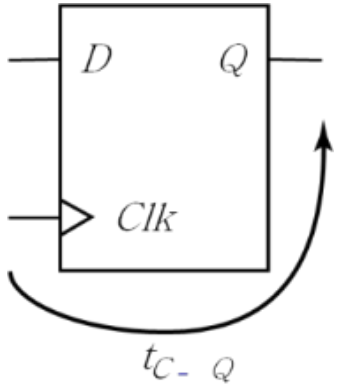


- Assumiremos que todos os elementos sequenciais são sensíveis à **borda**, utilizando flip-flops tipo D como registradores
- Os flip-flops tipo D possuem três parâmetros de temporização críticos:
  - t_{cq} – tempo de *clock* para saída: atraso de propagação
  - t_{setup} – tempo que os dados precisam chegar **antes** do clock
  - t_{hold} – tempo que os dados precisam permanecer estáveis após o **clock**

Parâmetro t_{cq}

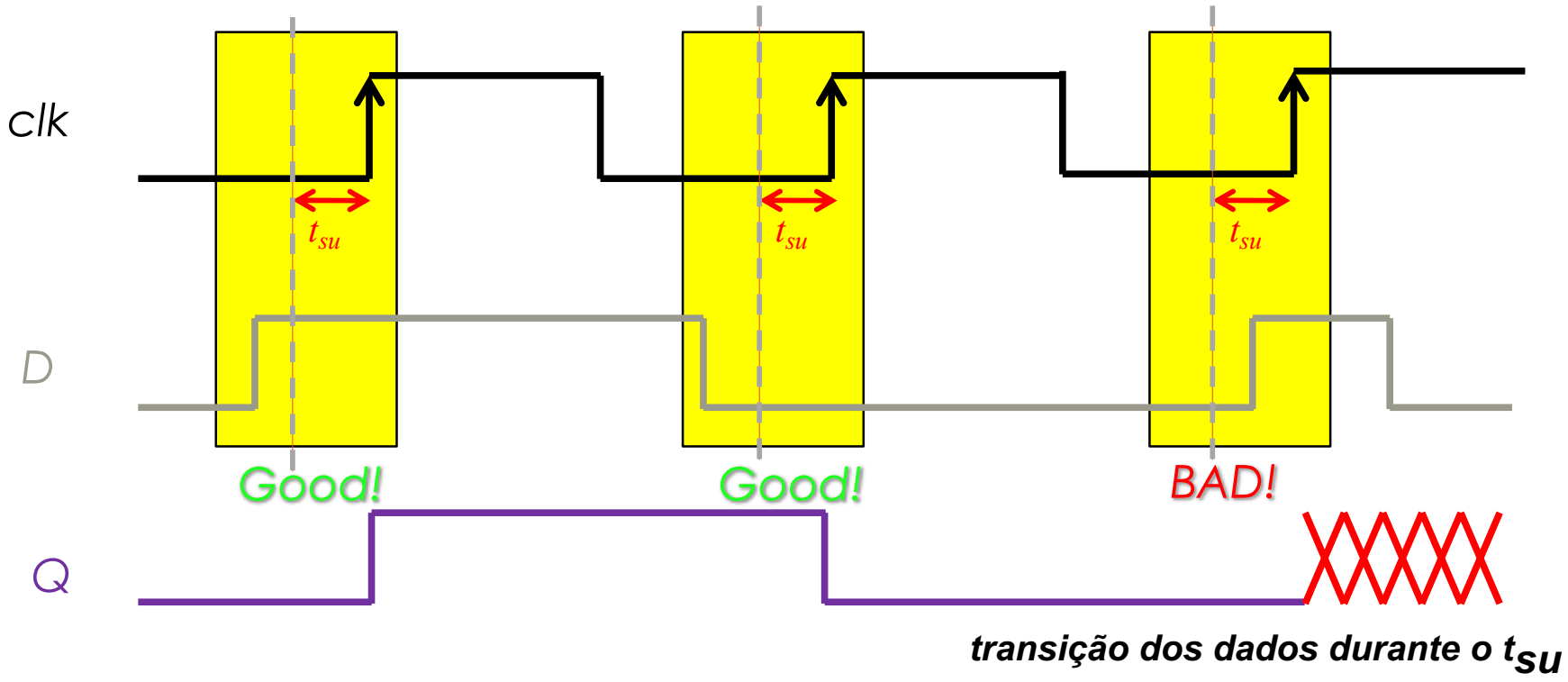
- t_{cq} é o tempo desde a borda do clock até os dados aparecerem na saída
- **tcq** possui tempos de propagação de subida e descida diferente





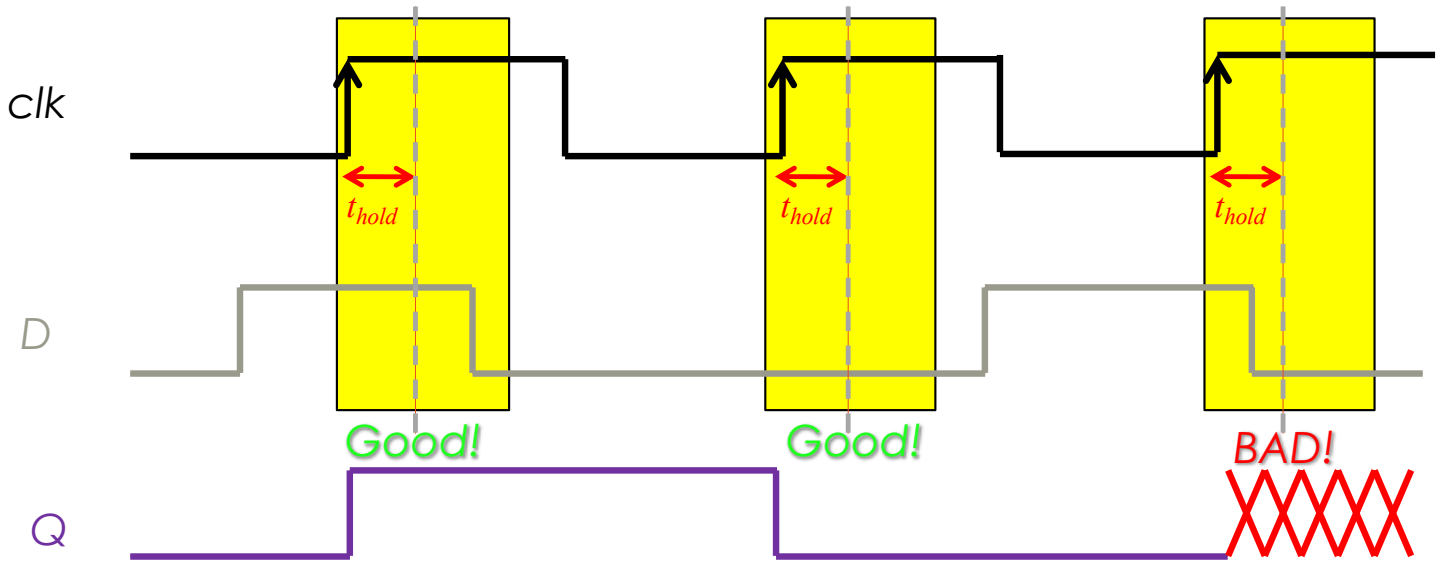
Parâmetro t_{setup}

- **t_{setup}** (setup time) é o tempo que os dados precisam chegar **antes** da borda de clock para garantir uma amostragem correta



Parâmetro t_{hold}

- t_{hold} (hold time) é o tempo que os dados precisam permanecer estáveis após o clock para garantir uma amostragem correta
- t_{hold} muito afetado por estágios muitos rápidos e *skew* do clock



transição dos dados durante o t_{hold}

Unindo os conceitos

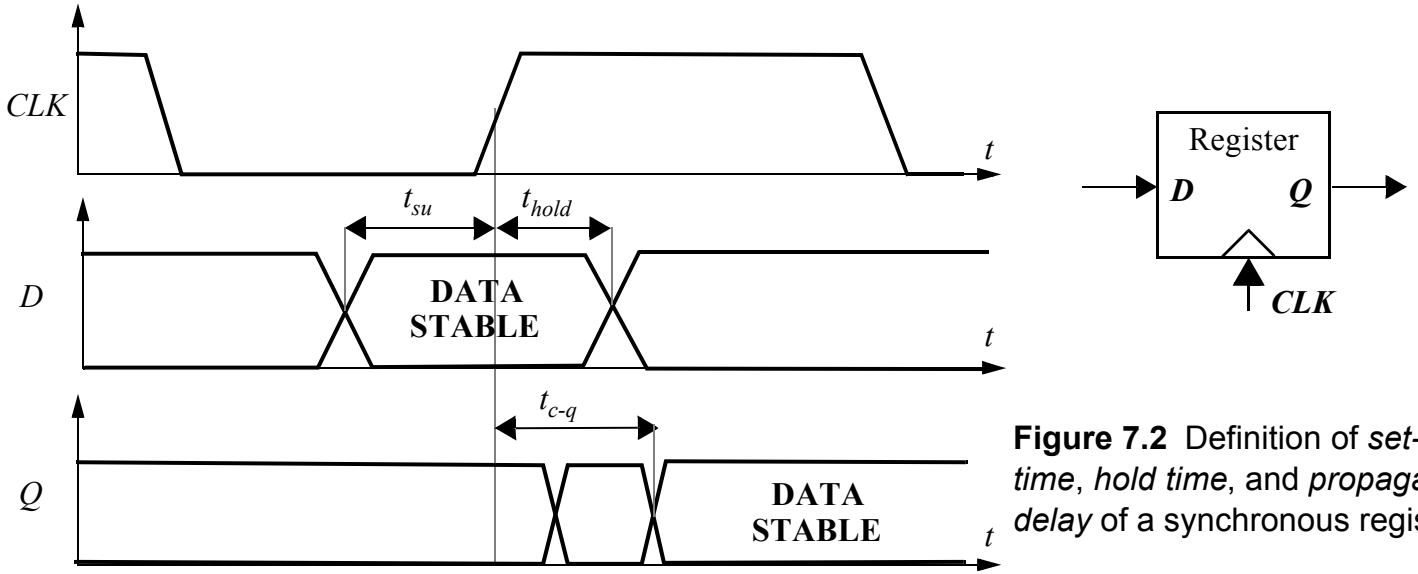
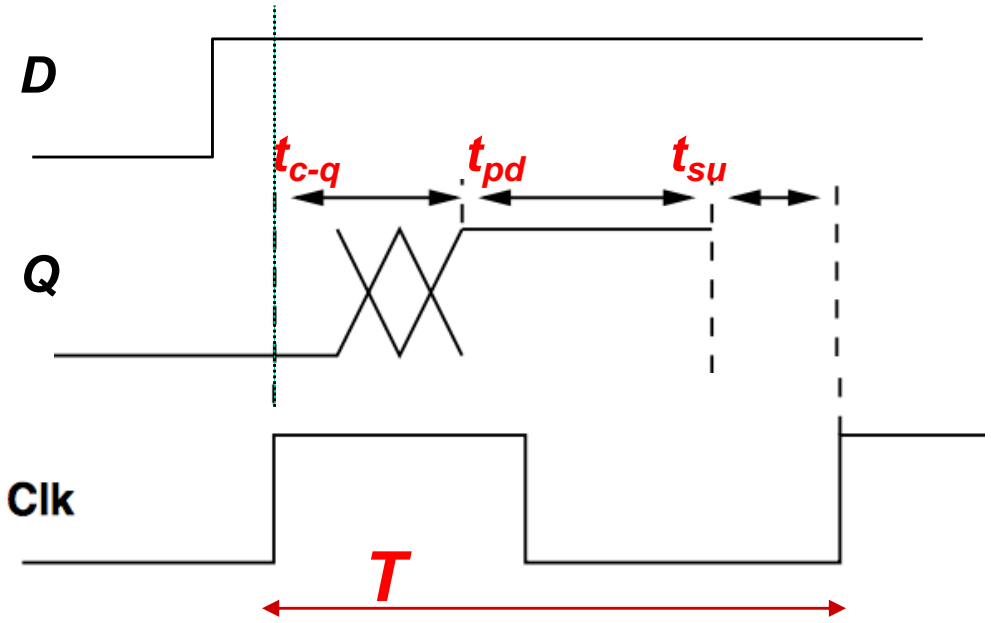


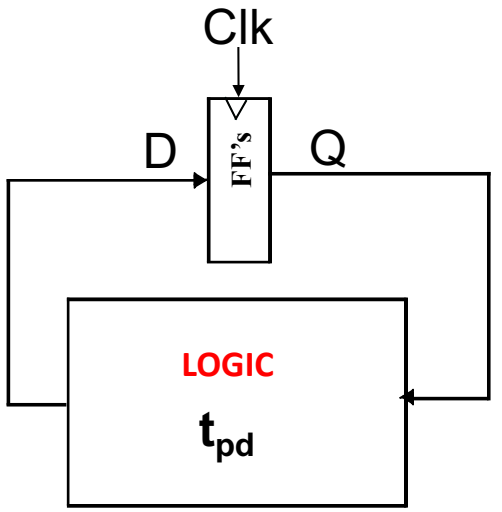
Figure 7.2 Definition of *set-up time*, *hold time*, and *propagation delay* of a synchronous register.

- Tempo de **set-up** (***t_{su}***): tempo que as entradas de dados (D) devem ser válidas antes da transição do relógio (0→1 para um registrador disparado por borda positiva)
- O tempo de **hold** (***t_{hold}***): tempo que a entrada de dados deve permanecer válida após a transição do clock
- Atraso do registrador (***$t_{c\text{-}q}$***): supondo que t_{su} e t_{hold} sejam atendidos, os dados na entrada D são copiados para a saída Q após $t_{c\text{-}q}$

Máxima Frequência de Clock



$$T \geq t_{cq} + t_{pd} + t_{su}$$



t_{cq}: *tempo de propagação do flip-flop*
t_{pd}: *tempo de propagação da lógica combinacional*
t_{su}: *tempo de setup do flip-flop*

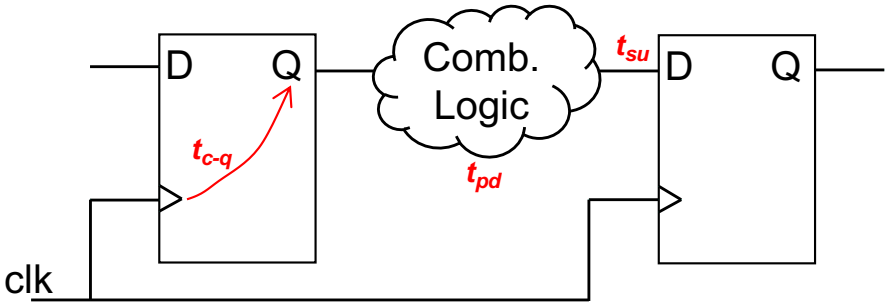
Slack time

- Slack Time é o tempo disponível entre o momento em que um evento é requerido e o momento em que ele ocorre
- No contexto de projeto digital, representa a **margem de segurança** entre os atrasos reais de sinal e os atrasos máximos permitidos para cumprir os requisitos de temporização

- **Slack positivo**: significa que o projeto tem tempo suficiente para o sinal atingir o próximo registrador, indicando que as restrições de temporização foram cumpridas

- **Slack negativo**: indica que o sinal não consegue atingir o próximo registrador dentro do tempo requerido, resultando em falhas de temporização. Isso pode ocorrer devido a:
    - Atrasos excessivos em lógica combinacional
    - Ciclos de clock muito curtos
    - Falta de otimização no projeto

Exemplo



(1) Determina o tempo de *slack*

$t_{c\text{-}q}$= 200 ps

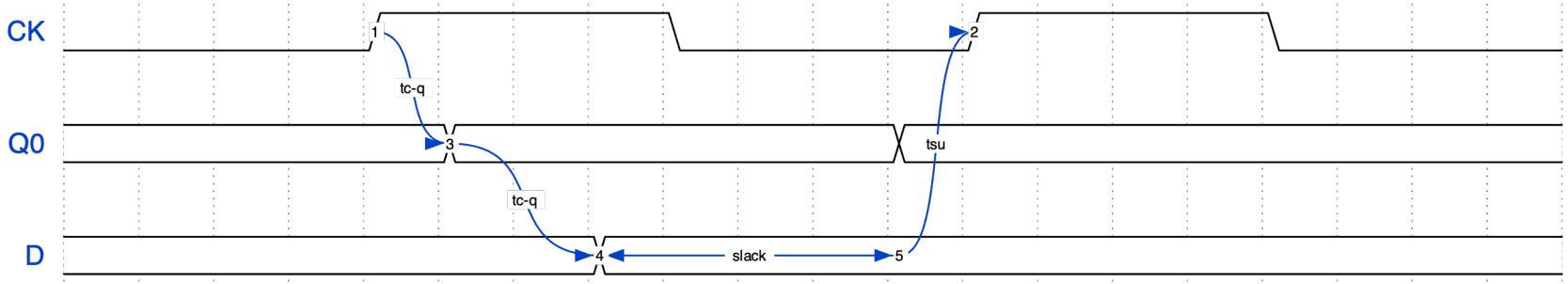
t_{su} = 100 ps

t_{pd} = 300 ps

T = 1000 ps

(2) Qual o período mínimo?

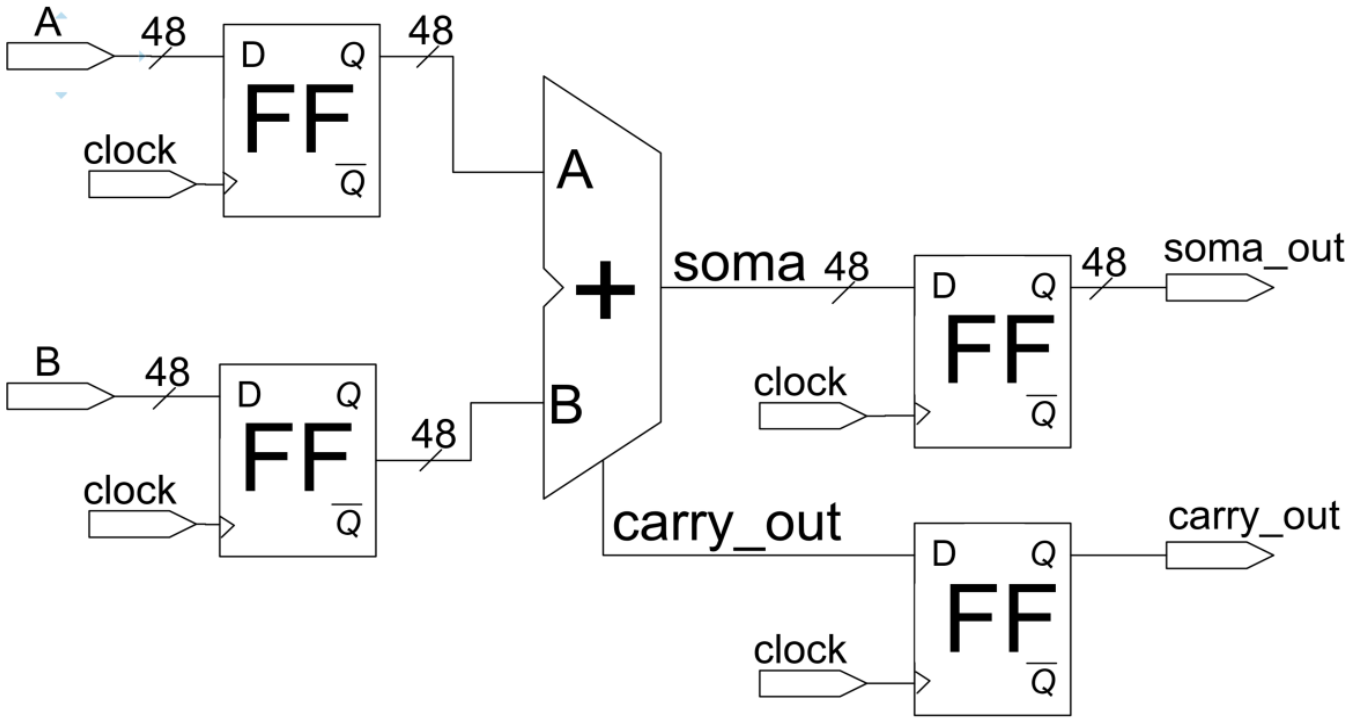
$T_{min} = 600\ ps$

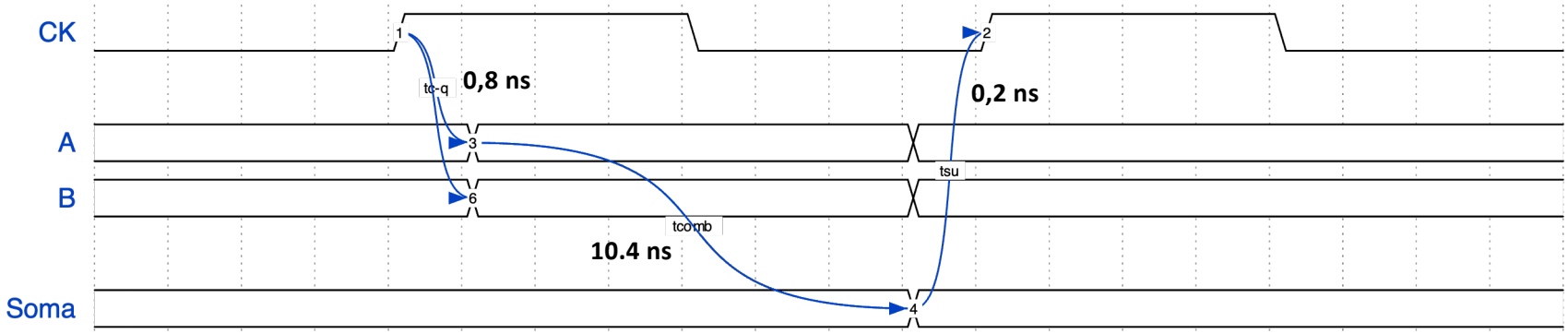


$t_{slack} = 1000 - 600 = 400\ ps$

- Considere o circuito abaixo, composto por 2 entradas de 48 bits (A e B), e duas saídas (*soma_out* de 48 bits, e *carry_out*).
  - Os flip-flops (FFs – mestre-escravo, sensíveis à borda de subida) possuem um tempo de propagação C→Q igual a 0,8 ns e o tempo de setup igual a 0,2 ns.
  - Um somador tem tempo de propagação igual a 10,4 ns

Determine a frequência máxima de operação do circuito.





Relatório de síntese - exemplo

```
@genus:root: 15> report timing

Path 1: MET (544 ps) Setup Check with Pin EA_reg[1]/CP->D
          Group: Bus2IP_Clkv - 2000 ps
     Startpoint: (R) address_reg[2]/CP
          Clock: (R) Bus2IP_Clk
       Endpoint: (R) EA_reg[1]/D
          Clock: (R) Bus2IP_Clk
...
           Setup:-      72
     Required Time:=    1928
      Launch Clock:-       0
         Data Path:-    1383
             Slack:=     544

#-------------------------------------------------------------------------------------------------------
#    Timing Point     Flags    Arc    Edge       Cell        Fanout Load Trans Delay Arrival Instance
#                                                                    (fF)  (ps)  (ps)   (ps)  Location
#-------------------------------------------------------------------------------------------------------
  address_reg[2]/CP  -        -      R     (arrival)           116    -     0     -       0    (-,-)
  address_reg[2]/Q   -        CP->Q  R     HS65_GS_DFPRQX9      20 72.0   165   133     133    (-,-)
  g21671/Z           -        A->Z   F     HS65_GS_IVX9         19 70.0   118   122     255    (-,-)
  g21598/Z           -        A->Z   F     HS65_GS_AND2X4       10 42.3   107   128     383    (-,-)
  g21539/Z           -        B->Z   R     HS65_GS_NAND2X7       7 24.8    78    77     460    (-,-)
  g21301/Z           -        B->Z   F     HS65_GS_CBI4I6X5      1  5.8    44    41     502    (-,-)
  g21245/Z           -        E->Z   R     HS65_GS_AOI311X4      1  4.5    60    48     550    (-,-)
  g21168/Z           -        B->Z   R     HS65_GS_NOR4ABX2      1  6.1   113   102     652    (-,-)
  g21037/Z           -        B->Z   F     HS65_GS_OAI212X5      1  5.2    59    50     701    (-,-)
  g21027/Z           -        D->Z   R     HS65_GS_NOR4ABX2      1  4.8    98    75     776    (-,-)
  g21012/Z           -        B->Z   R     HS65_GS_AND2X4       12 58.3   260   181     958    (-,-)
  g20979/Z           -        B->Z   F     HS65_GSS_XOR2X6       1  6.1    68   127    1084    (-,-)
  g20956/Z           -        E->Z   R     HS65_GS_OAI212X5      1  5.8    68    40    1125    (-,-)
  g20946/Z           -        E->Z   F     HS65_GS_AOI212X4      1  4.5    56    37    1162    (-,-)
  g20938/Z           -        B->Z   F     HS65_GS_NOR4ABX2      1  4.8    39    60    1222    (-,-)
  g20927/Z           -        B->Z   R     HS65_GS_AOI12X2       1  4.5    84    60    1282    (-,-)
  g20922/Z           -        B->Z   R     HS65_GS_NOR4ABX2      1  5.0    98   101    1383    (-,-)
  EA_reg[1]/D        <<<      -      R     HS65_GS_DFPRQX9       1    -     -     0    1383    (-,-)
#-------------------------------------------------------------------------------------------------------
```

LIB file – setup and hold

```
...
timing(){
       intrinsic_fall : 0.075;
       intrinsic_rise : 0.055;
       related_pin : "CP";
       sdf_edges : "both_edges";
       timing_label : "D_CP_SETUP";
       timing_type : setup_rising;
       fall_constraint(table_44){
        values("0.075000, 0.115000, 0.155000",\
            "0.040000, 0.080000, 0.125000",\
            "0.020000, 0.060000, 0.100000");
       }
       rise_constraint(table_44){
        values("0.055010, 0.095010, 0.125010",\
            "0.040010, 0.075010, 0.105010",\
            "0.030010, 0.070010, 0.095010");
       }
  }


  timing(){
       intrinsic_fall : -0.04;
       intrinsic_rise : -0.035;
       related_pin : "CP";
       sdf_edges : "both_edges";
       timing_label : "D_CP_HOLD";
       timing_type : hold_rising;
       fall_constraint(table_43){
        values("-0.040000, -0.015000, 0.000000",\
            "-0.080000, -0.060000, -0.040000",\
            "-0.120000, -0.095000, -0.075000");
       }
       rise_constraint(table_43){
        values("-0.034990, -0.019990, -0.014990",\
            "-0.074990, -0.059990, -0.054990",\
            "-0.104990, -0.089990, -0.079990");
       }
  }
```

intrinsic_fall : 0.075;
Este parâmetro define o atraso intrínseco (independente da carga) para transições de **fall** (queda) no pino monitorado, medido em nanosegundos

intrinsic_rise : 0.055;
Define o atraso intrínseco para transições de **rise** (subida) no pino monitorado, também em nanosegundos.

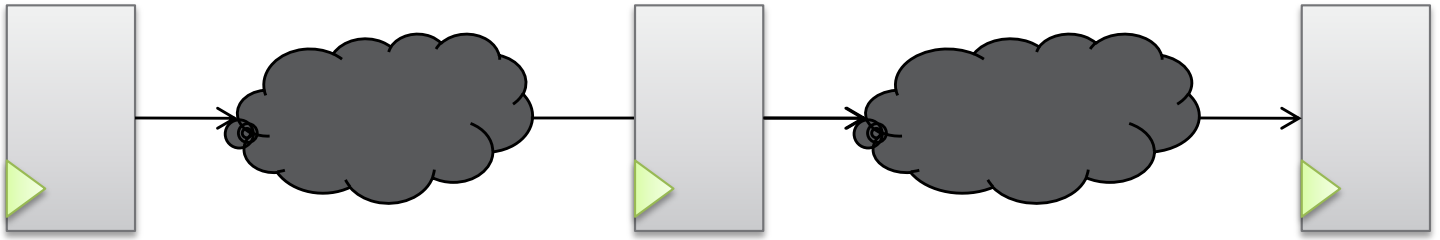
tabela: fast/typical/worst para diferentes rampas

Problemas de atraso entre registradores

Existem dois problemas principais que podem surgir na lógica síncrona:

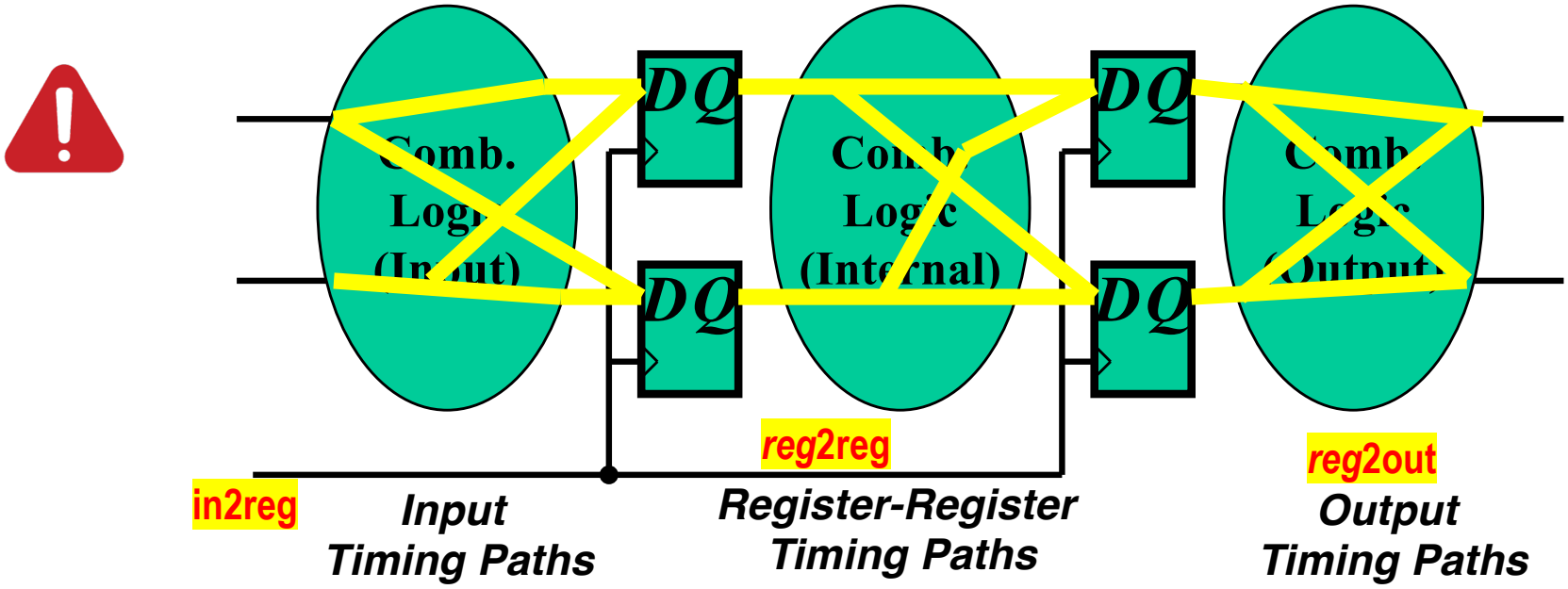
- Atraso Máximo (**Max Delay**): Os dados não têm tempo suficiente para passar de um registrador ao próximo antes da próxima borda do clock
  - As violações **Max Delay** são resultado de um caminho de dados lento, incluindo o $\mathbf{t_{setup}}$ dos registradores

- Atraso Mínimo (**Min Delay**): lógica combinacional muito rápida
  - As violações **Min Delay** são resultado de um caminho de dados rápido, fazendo com que os dados mudem antes de o **thold** ser atingido

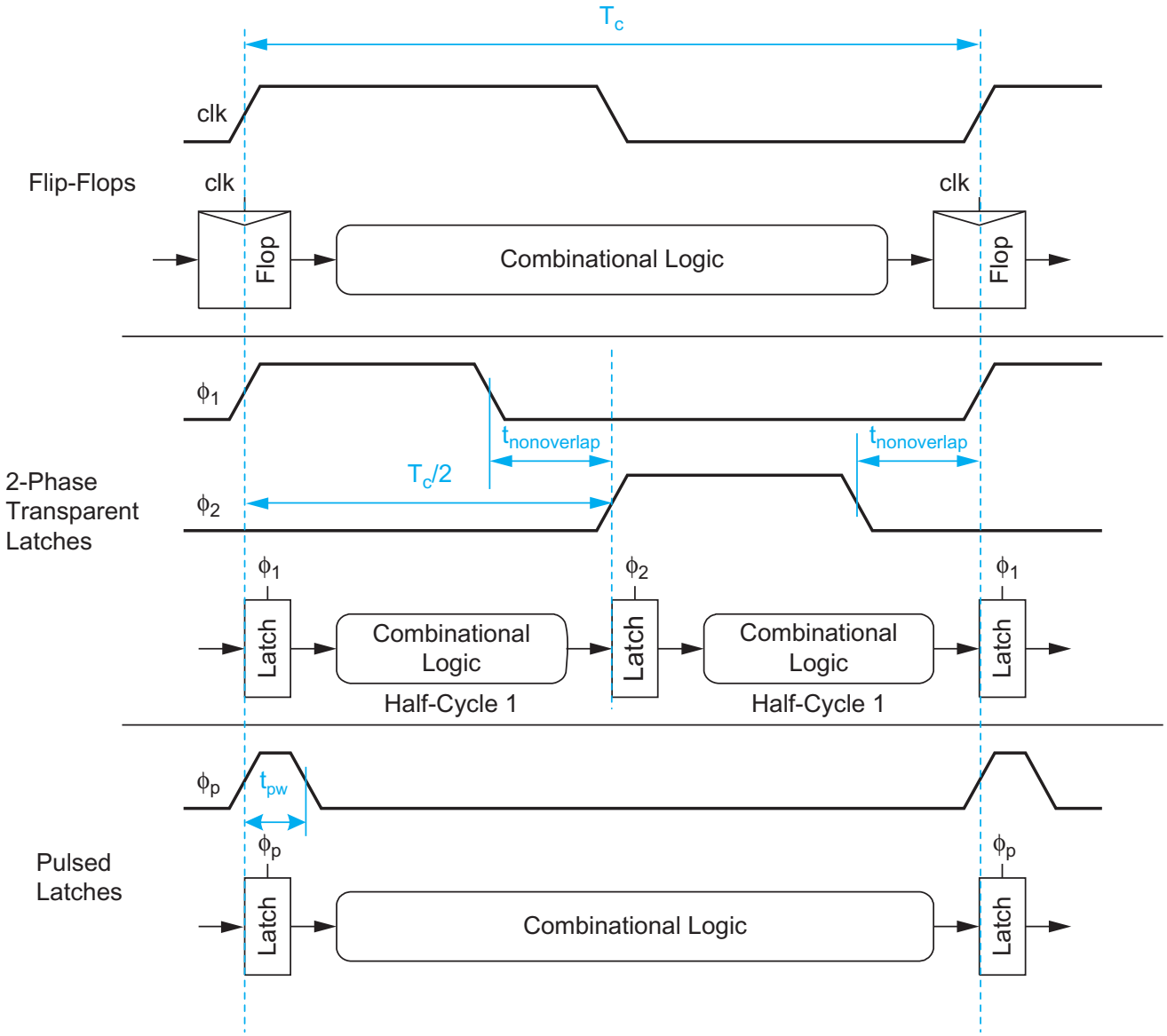


Restrições em Projeto Digital

- Assume-se um sistema síncrono baseado em clock
- Restrições de tempo entre os registradores (**período**)
- Restrições de tempo especificadas pelo usuário em entradas e saídas



Métodos de sincronização



MS

Latch-based design

Latch “pulsadas” (glitch)

FIGURE 10.2 Static sequencing methods