

# Circuitos Lógicos

## Aula 20

### Aula passada

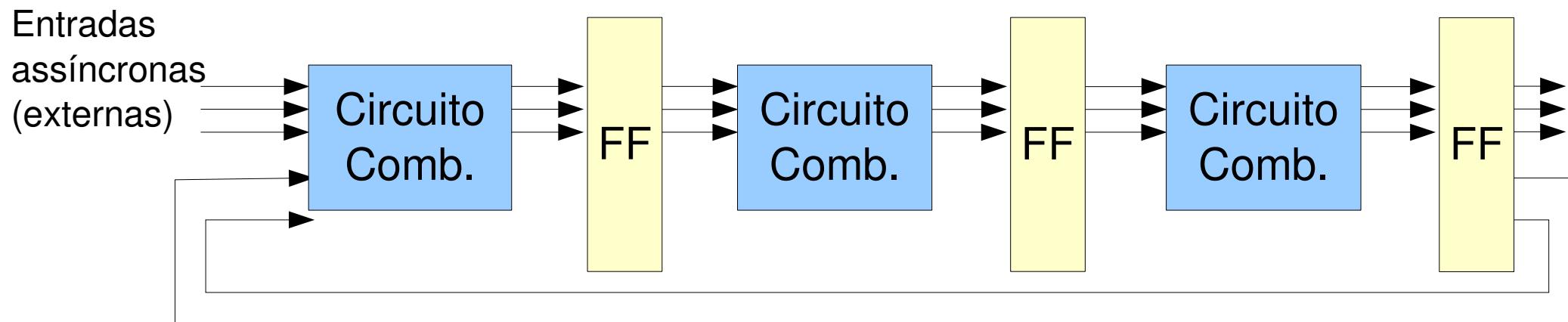
- FF com entradas assíncronas
- Tempos e atrasos
- Problema de temporização

### Aula de hoje

- Aplicações de FF
- Sincronização com FF
- Detectar de sequência
- Armazenamento e transferência

# Aplicações de FF

- Flip-Flops tem diversas aplicações
  - principalmente com clocks
- Armazenamento e transferência da dados é uma das principais
- Origem a circuitos sequenciais *com feedback*

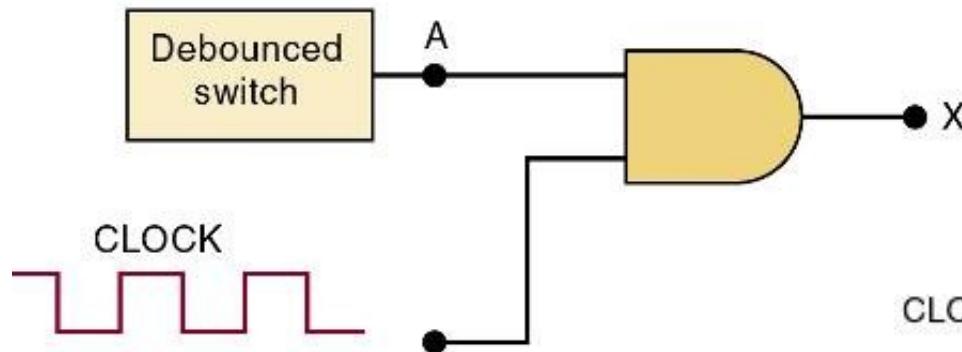


- Idéia por trás do microprocessador (CPU)

# Sincronização de Pulso

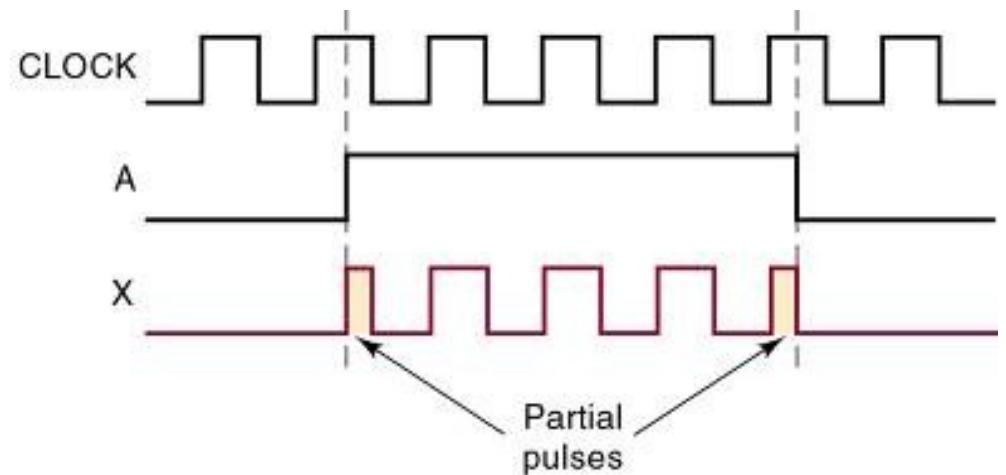


- Problema: habilitar e desabilitar assíncronamente (externo) a passagem de pulsos de clock
- Circuito para fazer isto?



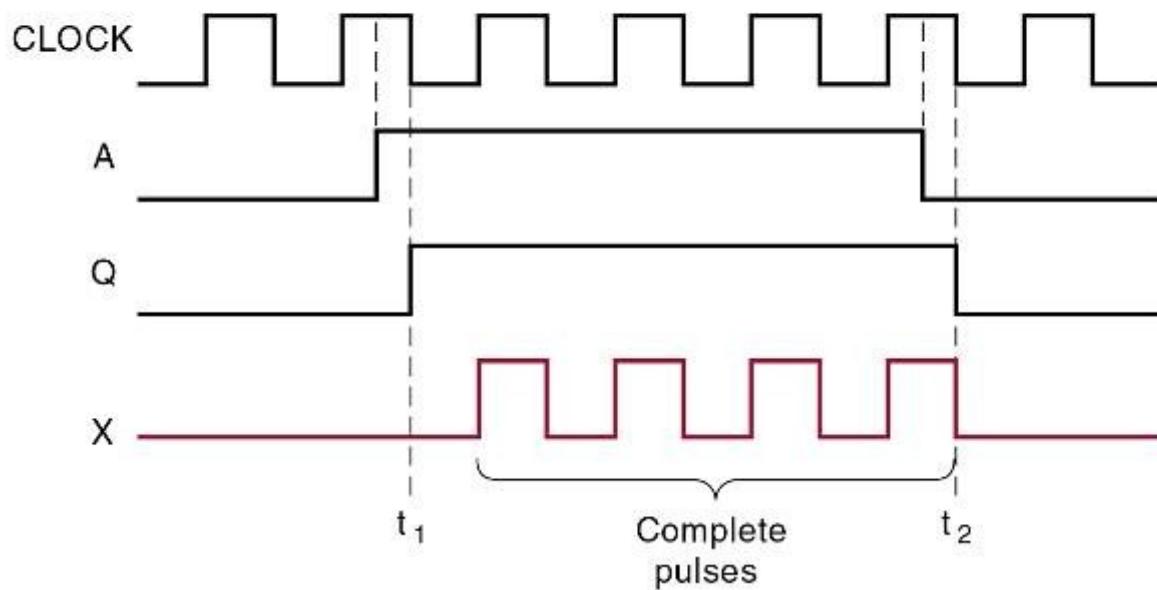
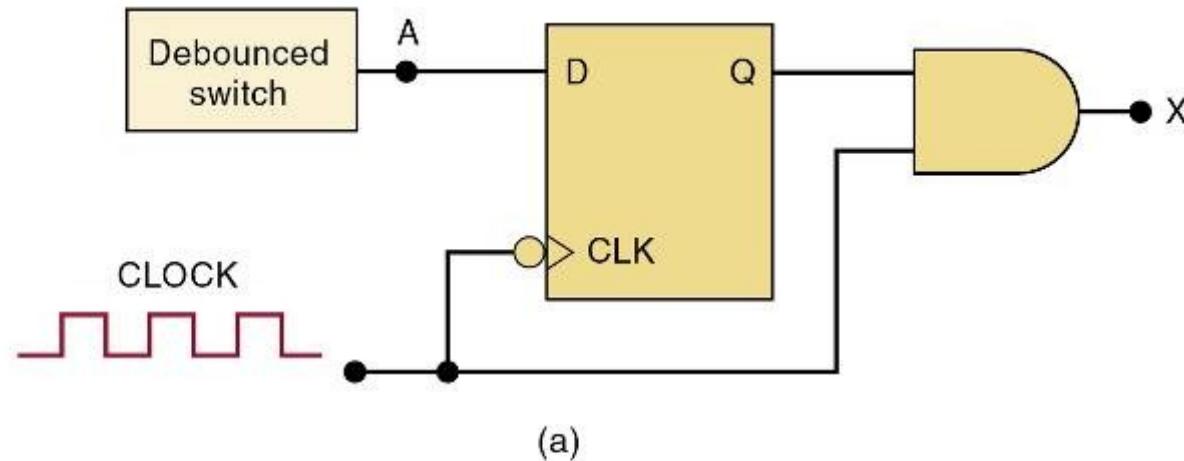
- Problema com o circuito ao lado?

- Solução para eliminar pulsos parciais?



# Sincronização com FFs

- Usar FF para remover pulsos parciais

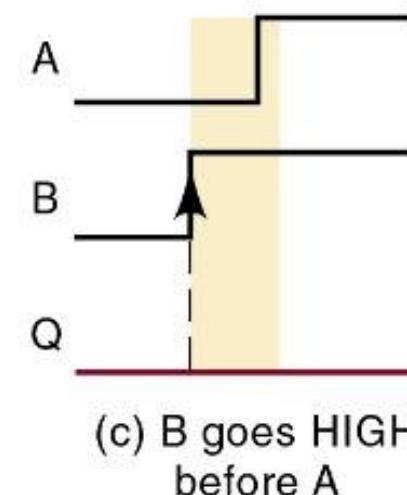
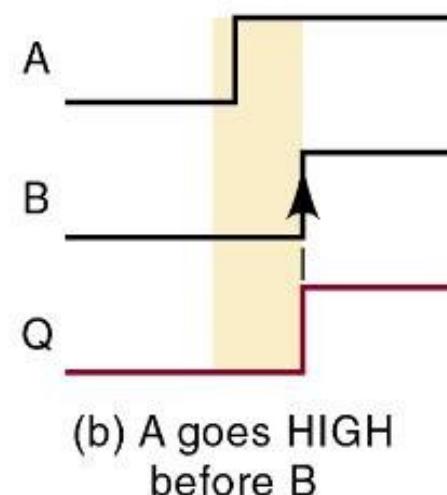
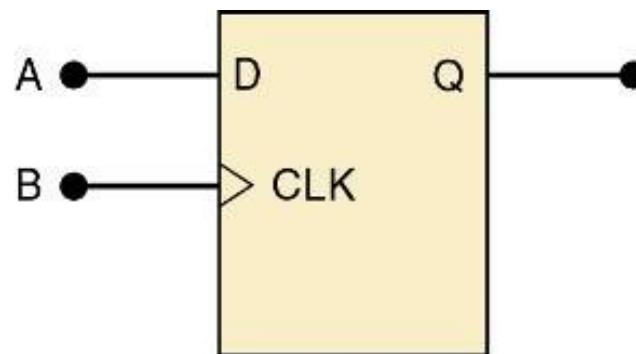


- FF D com TN
- Saída do FF somente na transição do pulso
- Sincronização da entrada
- AND do clock com saída do FF
- **Problema?**

# Detector Sequência de Entrada



- Detectar que entradas mudaram em uma determinada sequência
- Ex. A e B em baixo. Detectar que A subiu antes de B
  - saída do circuito neste caso é 1, cc. 0
- Circuito para isto?



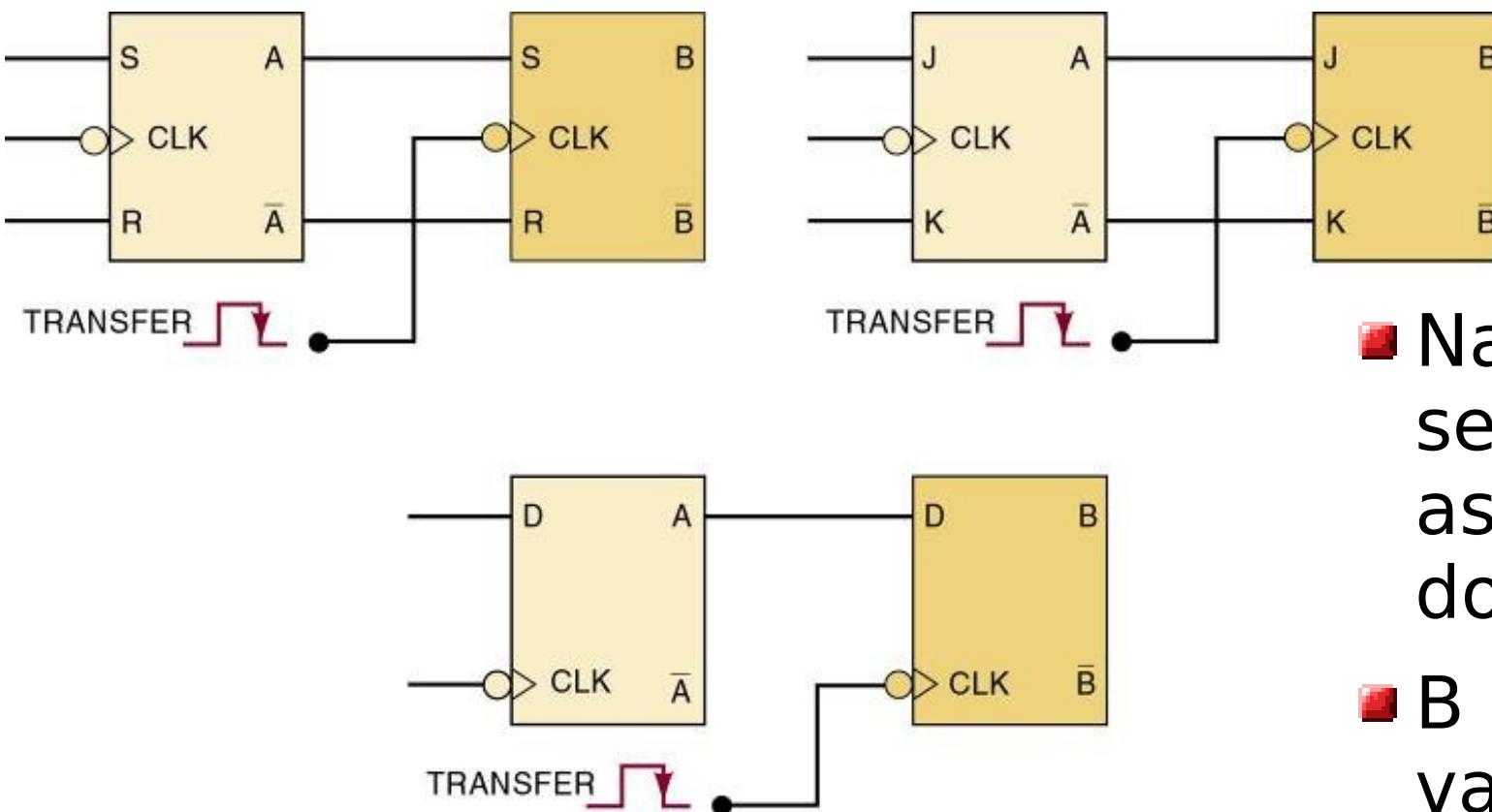
■ **Problema?**

# Armazenamento e Transferência

- Armazenar e transferir (transmitir) bits
- FFs usados em grupos, chamados de *registradores*
  - grupos de 4, 8, 16, 32, 64 (bits)
- Transferência ocorre quando movemos o *estado* de um registrador para outro
- Transferência pode ser síncrona ou assíncrona
- Transferência pode ser paralela ou serial

# Transferência Síncrona

- Controlada por pulsos de CLK
- Transferência em diferentes FFs com transição negativa

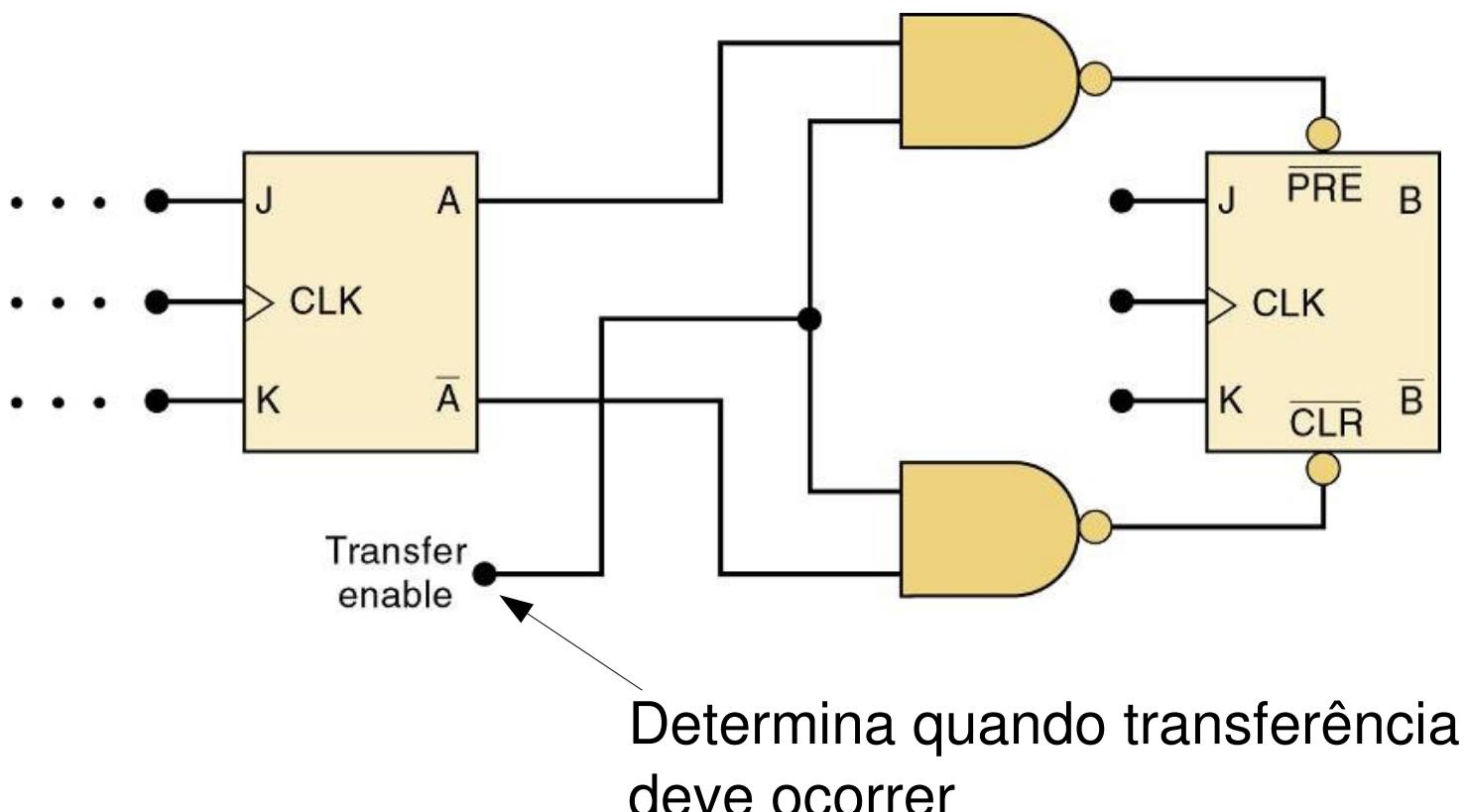


- Na transição, segundo FF assume estado do primeiro
- B passa a ter o valor de A

# Transferência Assíncrona



- Transferir estado a qualquer momento
- Como?
- Usar FF com entradas assíncronas
  - usar entradas PRE e CLR



# Transferência Paralela

- FFs do grupo de origem transferem estado para FF do grupo destino ao mesmo tempo
  - mesma transição de clock
- Ex.  $X=101$ ,  $Y=001$  antes da transição de clock
- Estado dos FFs depois da transição

