

AULA 14: APLICAÇÕES DE FLIP-FLOPS

CIRCUITOS DIGITAIS

Rodrigo Hausen

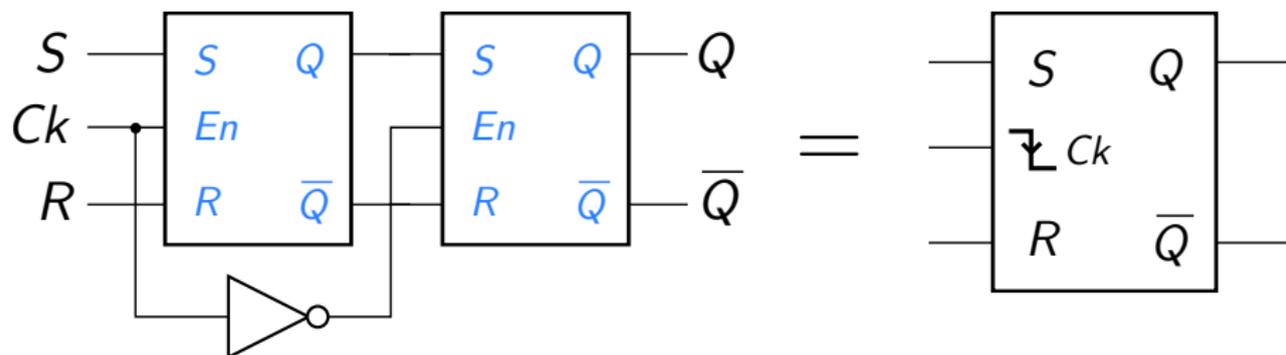
CMCC – UFABC

15 de março de 2013

<http://compscinet.org/circuitos>

RELEMBRANDO: FLIP-FLOP S-R

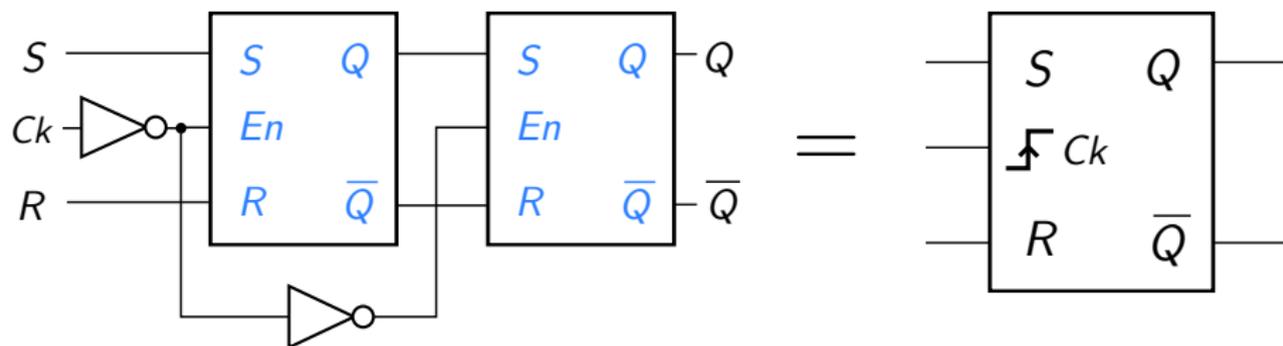
Flip-flop S-R sensível à borda de descida do clock (borda negativa)



S	R	Ck	Q_i	
0	0	?	Q_{i-1}	(mantem Q)
0	1	1→0	0	(reset Q)
1	0	1→0	1	(set Q)
1	1	1→0	X	(proibido)

RELEMBRANDO: FLIP-FLOP S-R

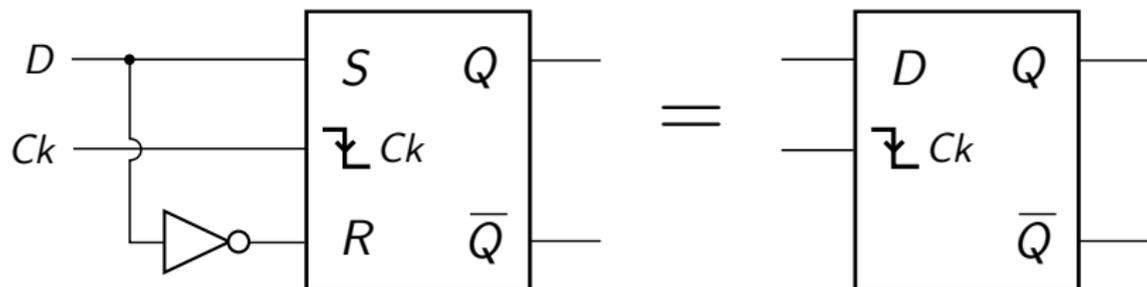
Flip-flop S-R sensível à borda de subida do clock (borda positiva)



S	R	Ck	Q_i	
0	0	?	Q_{i-1}	(mantem Q)
0	1	0→1	0	(reset Q)
1	0	0→1	1	(set Q)
1	1	0→1	X	(proibido)

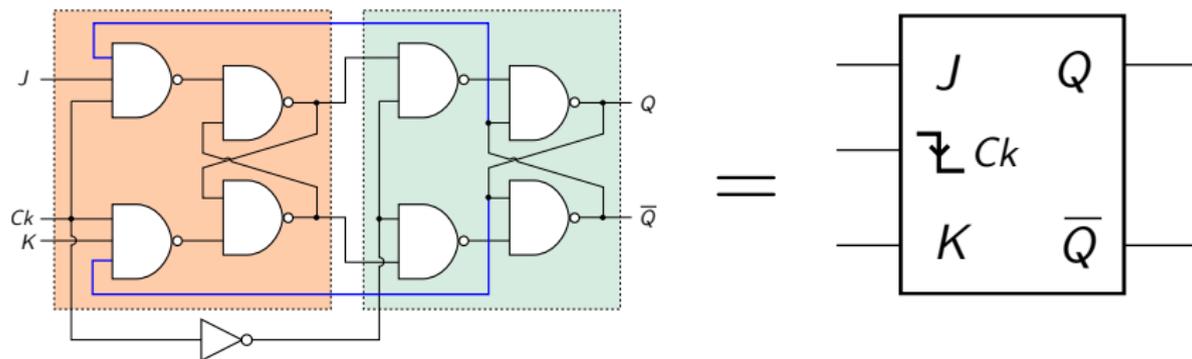
RELEMBRANDO: FLIP-FLOP D

Flip-flop D sensível à borda de descida.



D	Ck	Q_i	
0	$1 \rightarrow 0$	0	(reset = armazena 0)
1	$1 \rightarrow 0$	1	(set = armazena 1)

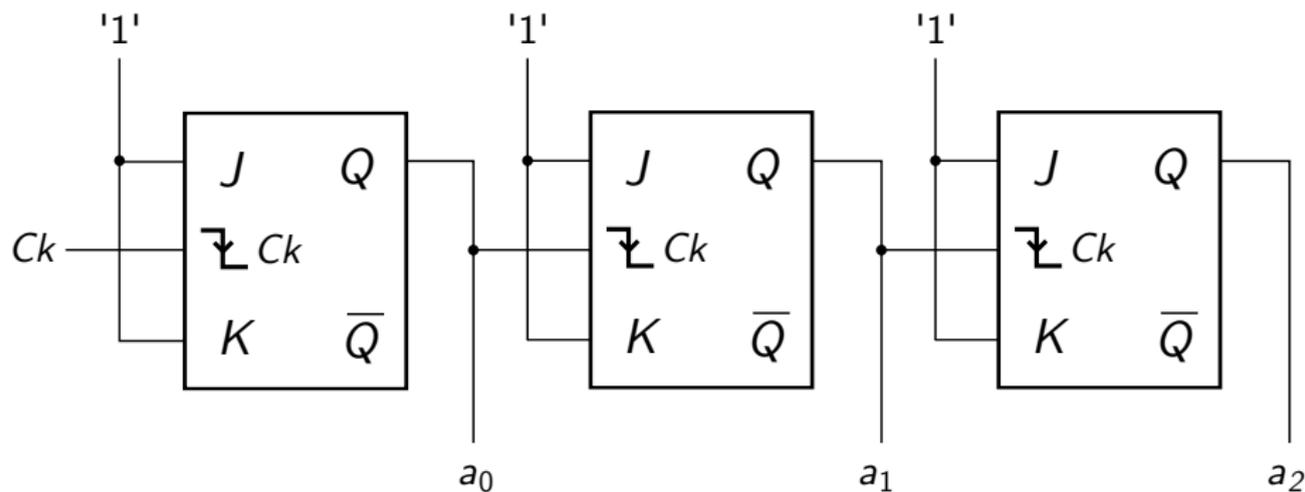
RELEMBRANDO: FLIP-FLOP J-K



J	K	Ck	Q _i	Q̄ _i	
0	0	?	Q _{i-1}	Q̄ _{i-1}	(mantem)
0	1	1→0	0	1	(kill = reset)
1	0	1→0	1	0	(jump = set)
1	1	1→0	Q̄ _{i-1}	Q _{i-1}	(inverte)

FLIP-FLOPS: APLICAÇÕES

Contador de 3 bits:



Problema: Faça um contador de 0 até 5.

FLIP-FLOPS: APLICAÇÕES

Problema: Faça um contador de 0 até 5.

Estados de a_2, a_1, a_0 , respectivamente:

000, 001, 010, 011, 100, 101, 000, ...

FLIP-FLOPS: APLICAÇÕES

Problema: Faça um contador de 0 até 5.

Estados de a_2, a_1, a_0 , respectivamente:

000, 001, 010, 011, 100, 101, 000, ...

Como fazer com que todas as saídas Q dos flip-flops J-K tornem-se 0 antes da transição $111 \rightarrow 000$ dos estados das saídas?

FLIP-FLOPS: APLICAÇÕES

Problema: Faça um contador de 0 até 5.

Estados de a_2, a_1, a_0 , respectivamente:

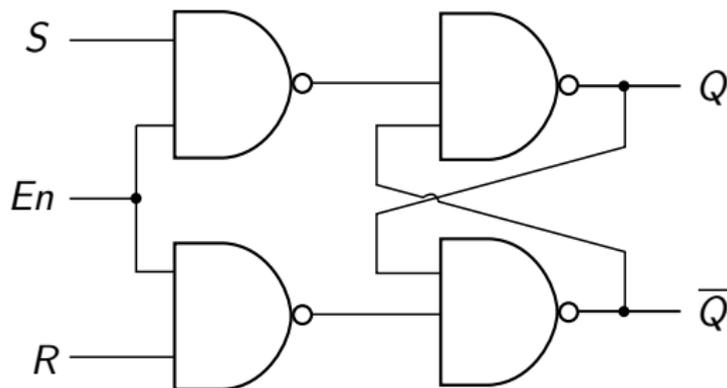
000, 001, 010, 011, 100, 101, 000, ...

Como fazer com que todas as saídas Q dos flip-flops J-K tornem-se 0 antes da transição $111 \rightarrow 000$ dos estados das saídas?

Precisamos alterar o flip-flop J-K para que possamos zerar Q independentemente de detectarmos uma borda do clock. Para isso, precisamos voltar ao latch S-R com enable.

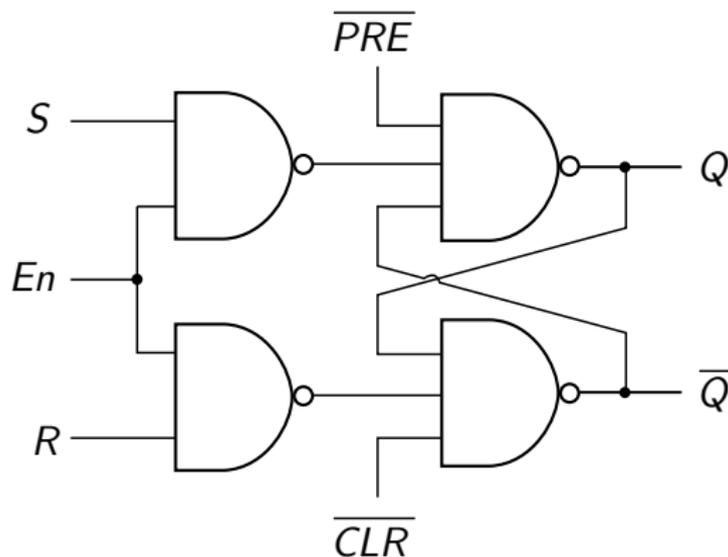
LATCH S-R COM ENABLE, PRESET E CLEAR

Problema: latch S-R só responde às entradas S e R quando $En = 1$. Queremos poder fazer *reset* e *set*, independentemente do estado de En .



LATCH S-R COM ENABLE, PRESET E CLEAR

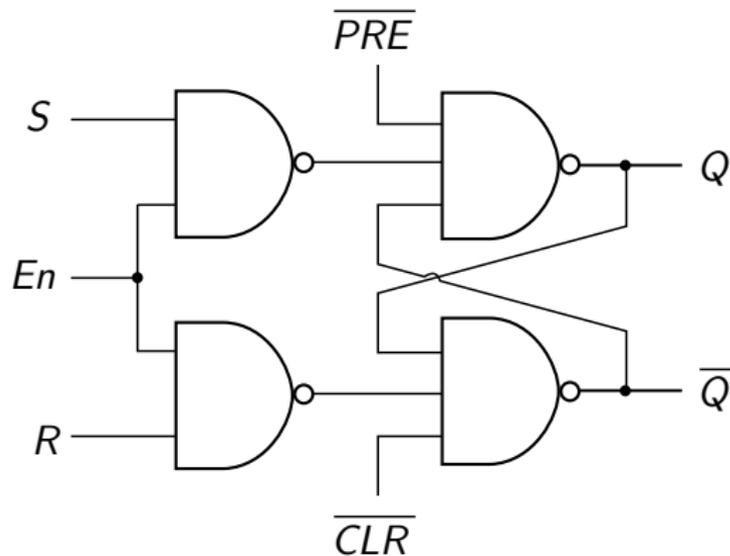
Problema: latch S-R só responde às entradas S e R quando $En = 1$. Queremos poder fazer *reset* e *set*, independentemente do estado de En .



Solução: adicione duas entradas ativas em nível baixo, \overline{PRE} e \overline{CLR} , às portas NAND das saídas.

LATCH S-R COM ENABLE, PRESET E CLEAR

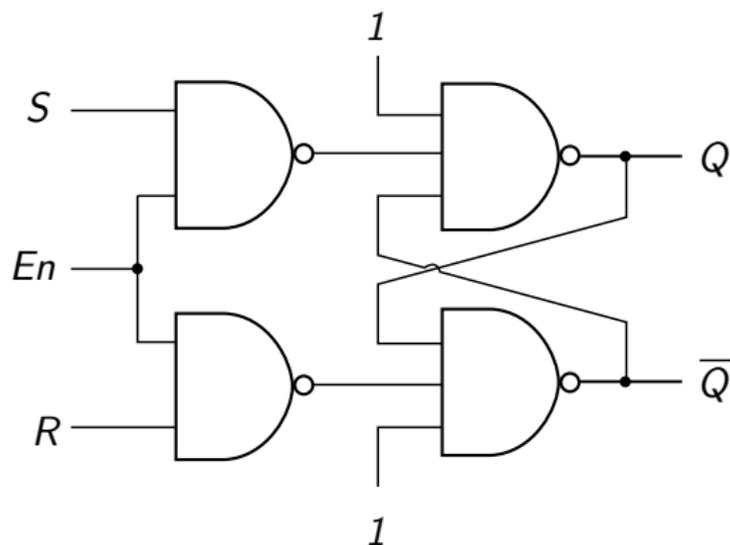
Comportamento do Latch S-R com Enable, Preset e Clear:



LATCH S-R COM ENABLE, PRESET E CLEAR

Comportamento do Latch S-R com Enable, Preset e Clear:

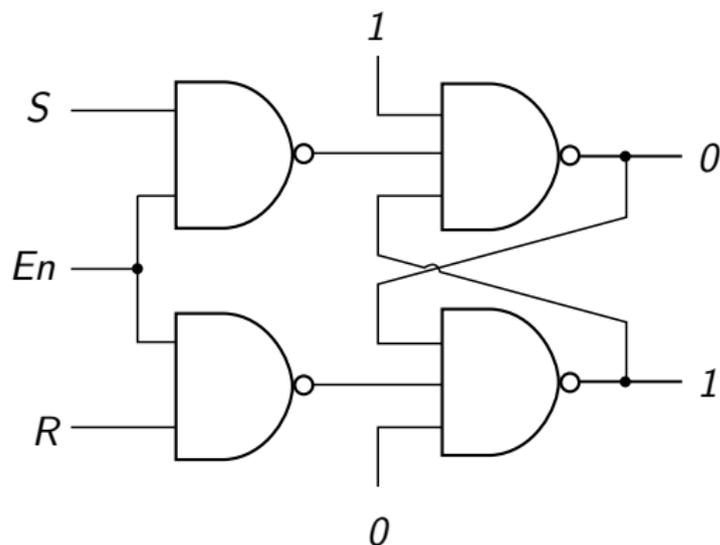
- quando $\overline{PRE} = 1$ e $\overline{CLR} = 1$, idêntico ao latch S-R com enable;



LATCH S-R COM ENABLE, PRESET E CLEAR

Comportamento do Latch S-R com Enable, Preset e Clear:

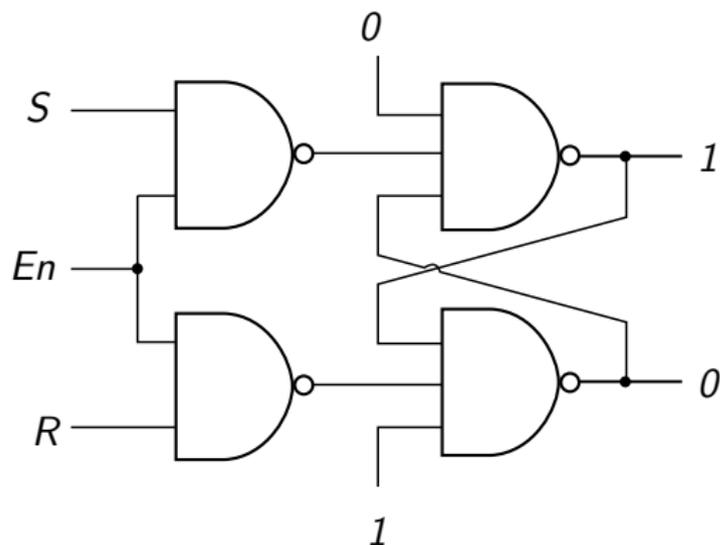
- quando $\overline{PRE} = 1$ e $\overline{CLR} = 1$, idêntico ao latch S-R com enable;
- quando $\overline{PRE} = 1$ e $\overline{CLR} = 0$, ocorre *reset* ($Q = 0$);



LATCH S-R COM ENABLE, PRESET E CLEAR

Comportamento do Latch S-R com Enable, Preset e Clear:

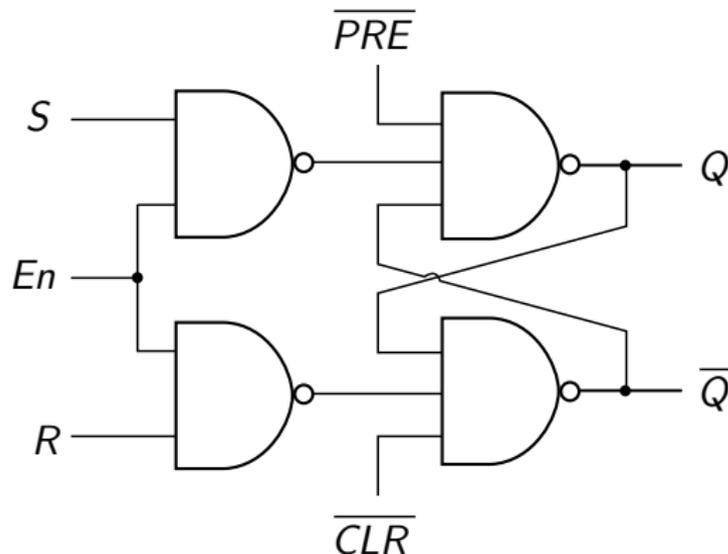
- quando $\overline{PRE} = 1$ e $\overline{CLR} = 1$, idêntico ao latch S-R com enable;
- quando $\overline{PRE} = 1$ e $\overline{CLR} = 0$, ocorre *reset* ($Q = 0$);
- quando $\overline{PRE} = 0$ e $\overline{CLR} = 1$, ocorre *set* ($Q = 1$);



LATCH S-R COM ENABLE, PRESET E CLEAR

Comportamento do Latch S-R com Enable, Preset e Clear:

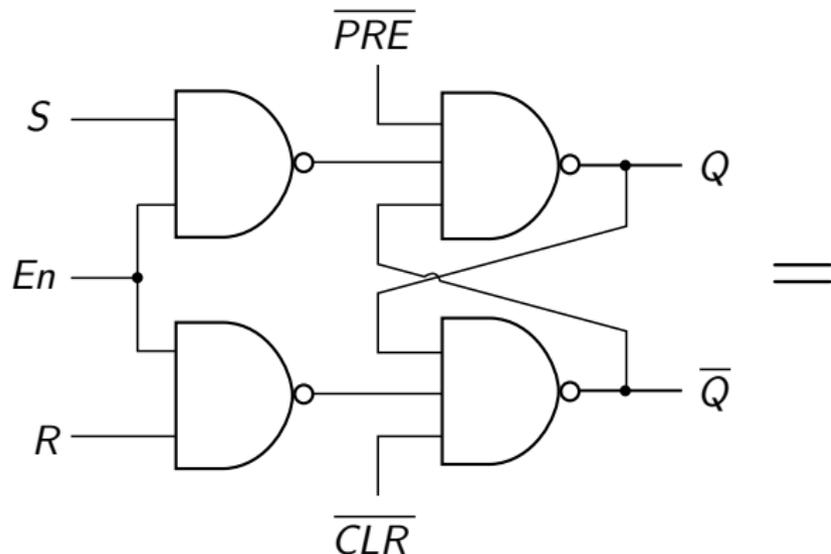
- quando $\overline{PRE} = 1$ e $\overline{CLR} = 1$, idêntico ao latch S-R com enable;
- quando $\overline{PRE} = 1$ e $\overline{CLR} = 0$, ocorre *reset* ($Q = 0$);
- quando $\overline{PRE} = 0$ e $\overline{CLR} = 1$, ocorre *set* ($Q = 1$);
- é proibido $\overline{PRE} = 0$ e $\overline{CLR} = 0$



LATCH S-R COM ENABLE, PRESET E CLEAR

Comportamento do Latch S-R com Enable, Preset e Clear:

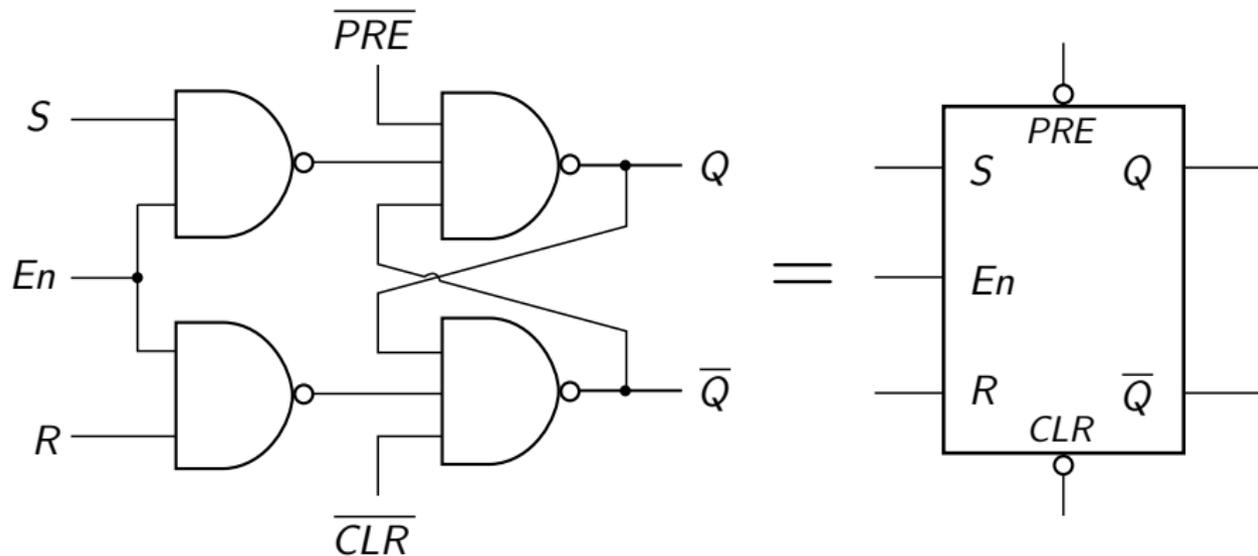
- quando $\overline{PRE} = 1$ e $\overline{CLR} = 1$, idêntico ao latch S-R com enable;
- quando $\overline{PRE} = 1$ e $\overline{CLR} = 0$, ocorre *reset* ($Q = 0$);
- quando $\overline{PRE} = 0$ e $\overline{CLR} = 1$, ocorre *set* ($Q = 1$);
- é proibido $\overline{PRE} = 0$ e $\overline{CLR} = 0$



LATCH S-R COM ENABLE, PRESET E CLEAR

Comportamento do Latch S-R com Enable, Preset e Clear:

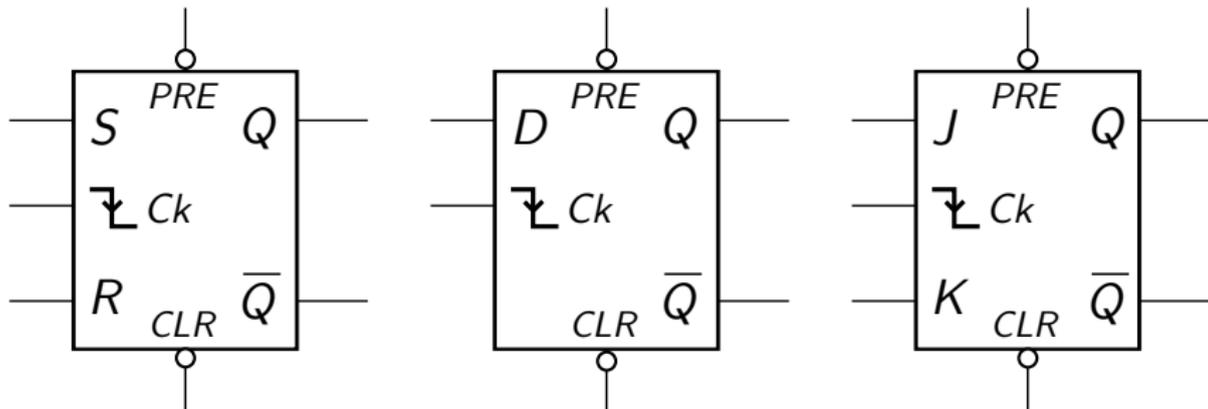
- quando $\overline{PRE} = 1$ e $\overline{CLR} = 1$, idêntico ao latch S-R com enable;
- quando $\overline{PRE} = 1$ e $\overline{CLR} = 0$, ocorre *reset* ($Q = 0$);
- quando $\overline{PRE} = 0$ e $\overline{CLR} = 1$, ocorre *set* ($Q = 1$);
- é proibido $\overline{PRE} = 0$ e $\overline{CLR} = 0$



ENTRADAS ASSÍNCRONAS: PRESET E CLEAR

Usando latches S-R com Enable, Preset e Clear, é possível fazer flip-flops S-R, D e J-K com entradas Preset e Clear ativas em nível baixo.

- As entradas \overline{PRE} e \overline{CLR} são ditas **assíncronas** pois elas interferem no funcionamento dos flip-flops **independentemente do clock** (ao contrário das demais entradas, que só afetam o flip-flop em uma das bordas do clock)



VOLTANDO AO CONTADOR

Voltando ao problema original: fazer um contador de 0 até 5.

Contador de 3 bits (conta de 0 a 7):

000 → 001 → 010 → 011 → 100 → 101 → 110 → 111



VOLTANDO AO CONTADOR

Voltando ao problema original: fazer um contador de 0 até 5.

Contador de 3 bits (conta de 0 a 7):

000 → 001 → 010 → 011 → 100 → 101 → 110 → 111
↑ _____ |

Como evitar a transição 101 → 110, transformando-a em 101 → 000?

000 → 001 → 010 → 011 → 100 → 101 → 110 → 111
↑ _____ |

VOLTANDO AO CONTADOR

Voltando ao problema original: fazer um contador de 0 até 5.

Contador de 3 bits (conta de 0 a 7):

000 → 001 → 010 → 011 → 100 → 101 → 110 → 111


Como evitar a transição 101 → 110, transformando-a em 101 → 000?

000 → 001 → 010 → 011 → 100 → 101 → 110 → 111


Idéia: fazer reset dos flip-flops do contador assim que a contagem for excedida!

VOLTANDO AO CONTADOR

Voltando ao problema original: fazer um contador de 0 até 5.

Contador de 3 bits (conta de 0 a 7):

000 → 001 → 010 → 011 → 100 → 101 → 110 → 111


Como evitar a transição 101 → 110, transformando-a em 101 → 000?

000 → 001 → 010 → 011 → 100 → 101 → 110 → 111


Idéia: fazer reset dos flip-flops do contador assim que a contagem for excedida!

Isto acontece quando chegamos em $5 + 1 = (110)_2$

VOLTANDO AO CONTADOR

Agora, o problema é detectar, por meio de um sinal ativo de **nível baixo**, quando $(a_2 a_1 a_0)_2 = (110)_2$

VOLTANDO AO CONTADOR

Agora, o problema é detectar, por meio de um sinal ativo de **nível baixo**, quando $(a_2 a_1 a_0)_2 = (110)_2$

Deteccção por nível alto: $a_2 a_1 \bar{a}_0$

VOLTANDO AO CONTADOR

Agora, o problema é detectar, por meio de um sinal ativo de **nível baixo**, quando $(a_2 a_1 a_0)_2 = (110)_2$

Detecção por nível alto: $a_2 a_1 \bar{a}_0$

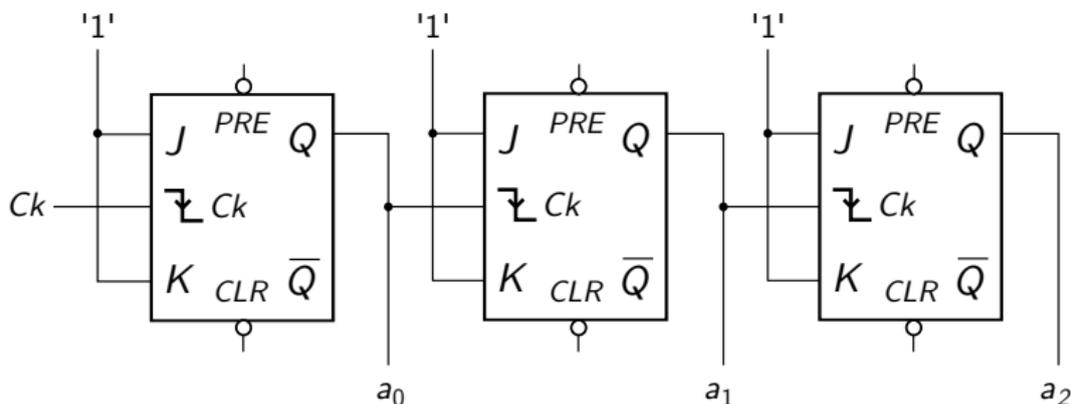
Detecção por nível baixo: $\overline{a_2 a_1 a_0}$

VOLTANDO AO CONTADOR

Agora, o problema é detectar, por meio de um sinal ativo de **nível baixo**, quando $(a_2 a_1 a_0)_2 = (110)_2$

Detecção por nível alto: $a_2 a_1 \bar{a}_0$

Detecção por nível baixo: $\overline{a_2 a_1 a_0}$

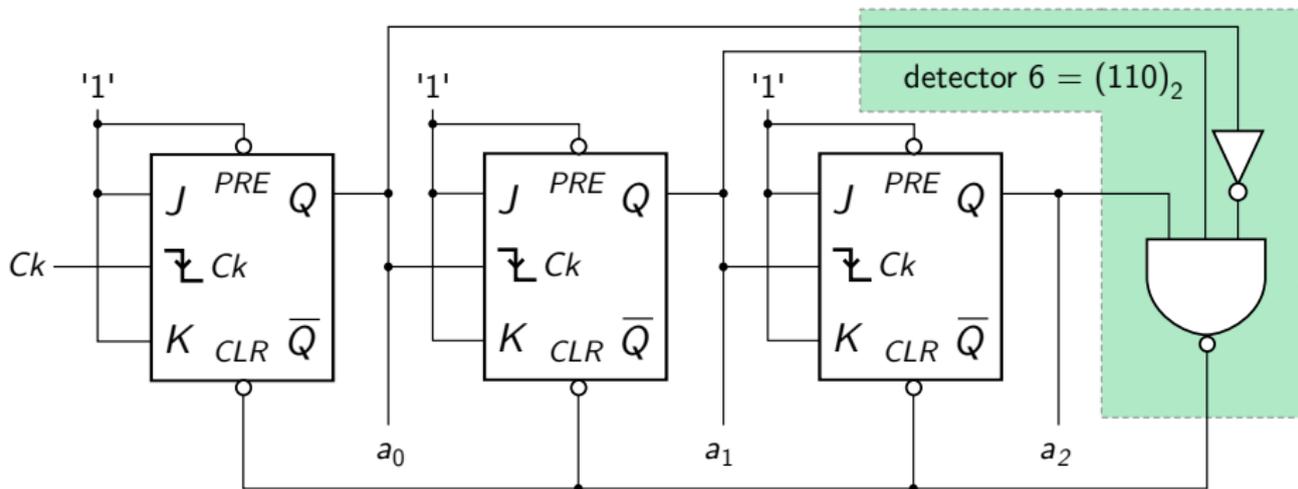


VOLTANDO AO CONTADOR

Agora, o problema é detectar, por meio de um sinal ativo de **nível baixo**, quando $(a_2 a_1 a_0)_2 = (110)_2$

Detecção por nível alto: $a_2 a_1 \bar{a}_0$

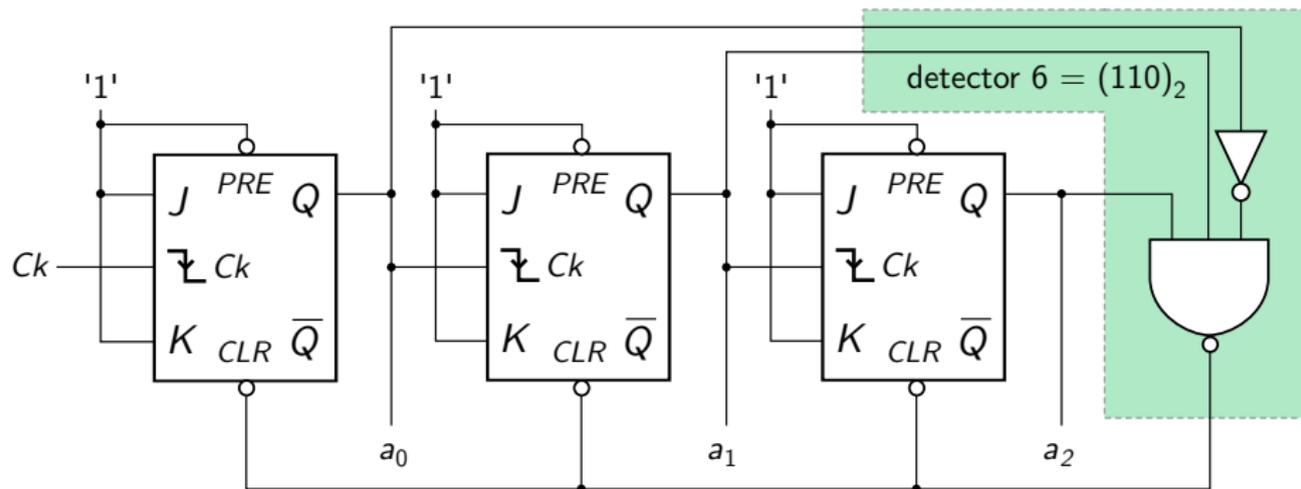
Detecção por nível baixo: $\bar{a}_2 a_1 \bar{a}_0$



CONTADOR MÓDULO m

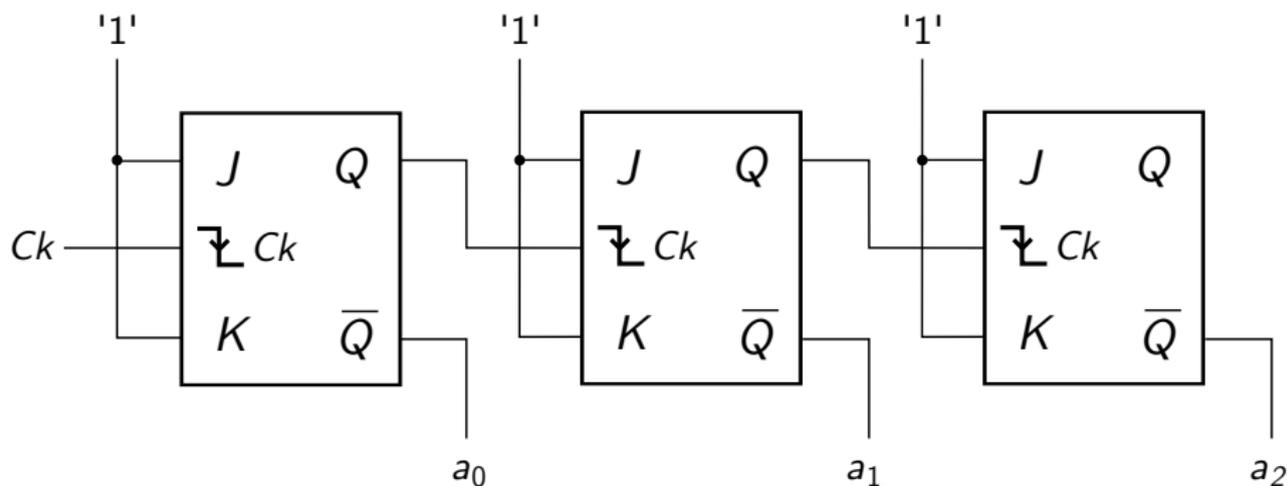
- **Contador módulo m** : circuito digital com uma entrada de clock que conta de 0 até $m - 1$.

Exemplo: contador módulo 6, que conta de 0 a 5.



FLIP-FLOPS: APLICAÇÕES

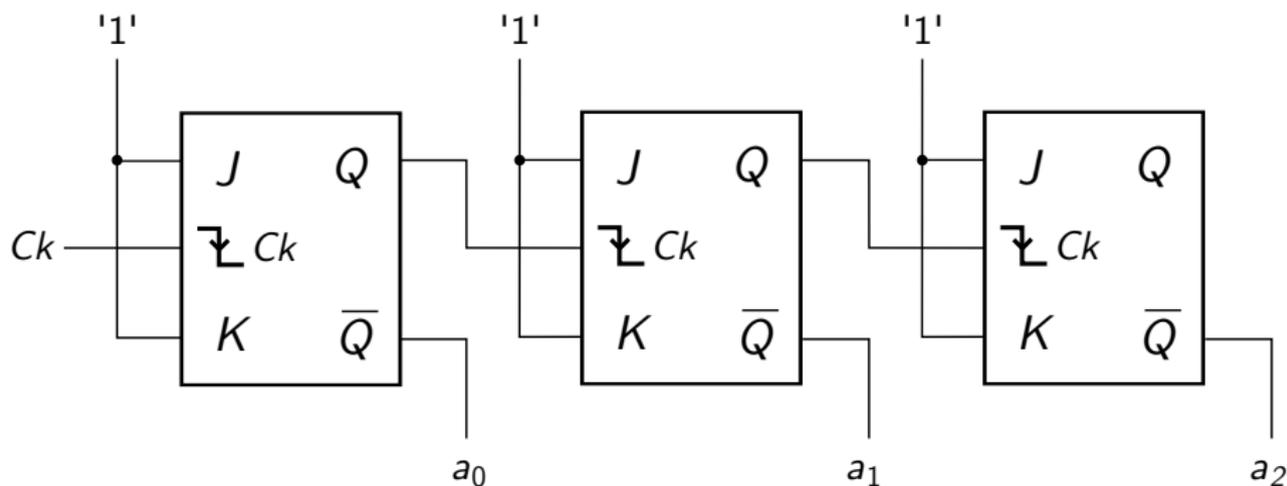
Faça o diagrama de forma de onda para o circuito abaixo e diga o que ele faz.



(Solução na lousa)

FLIP-FLOPS: APLICAÇÕES

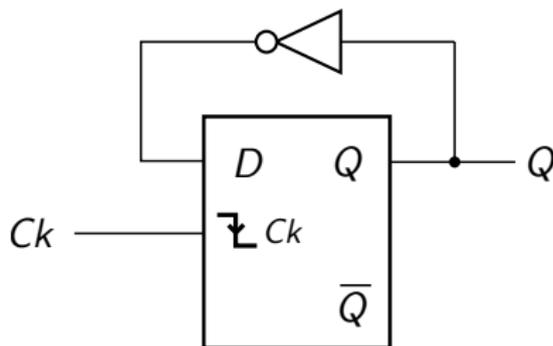
Faça o diagrama de forma de onda para o circuito abaixo e diga o que ele faz.



(Solução na lousa) É um contador regressivo para 3 bits: 111, 110, 101, ..., 001, 000, 111, ...

FLIP-FLOPS: APLICAÇÕES

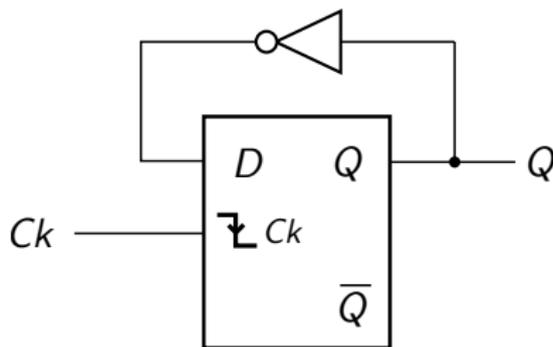
Faça o diagrama de forma de onda para o circuito abaixo e diga o que ele faz (entrada: Ck , saída: Q)



(Solução na lousa)

FLIP-FLOPS: APLICAÇÕES

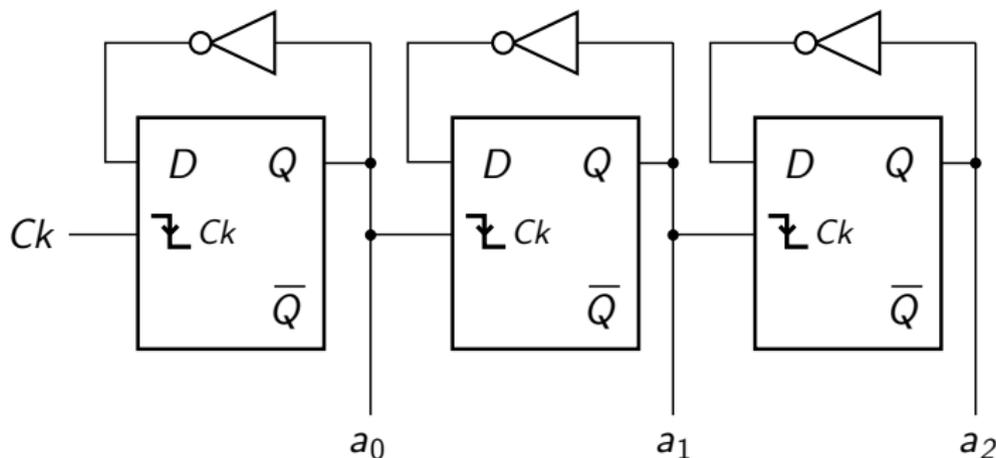
Faça o diagrama de forma de onda para o circuito abaixo e diga o que ele faz (entrada: Ck , saída: Q)



(Solução na lousa) Funciona como um flip-flop J-K com as entradas $J = 1$ e $K = 1$, ou seja, inverte a saída a cada borda de descida do clock.

FLIP-FLOPS: APLICAÇÕES

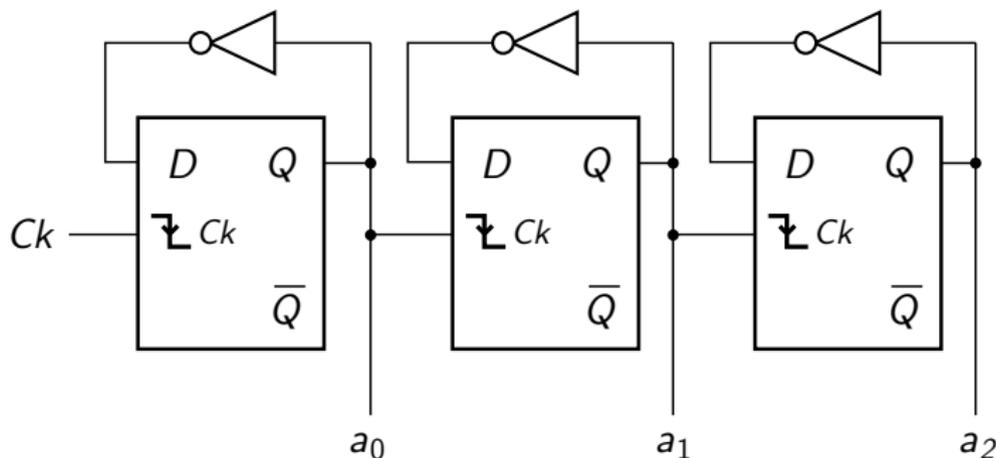
Faça o diagrama de forma de onda para o circuito abaixo e diga o que ele faz (entrada: Ck , saídas: a_2, a_1, a_0)



(Solução na lousa)

FLIP-FLOPS: APLICAÇÕES

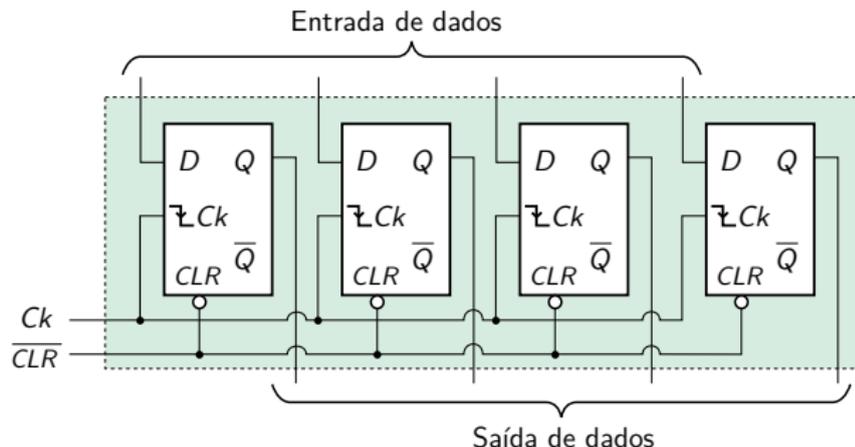
Faça o diagrama de forma de onda para o circuito abaixo e diga o que ele faz (entrada: Ck , saídas: a_2, a_1, a_0)



(Solução na lousa) Outra maneira de implementar um contador de 3 bits.

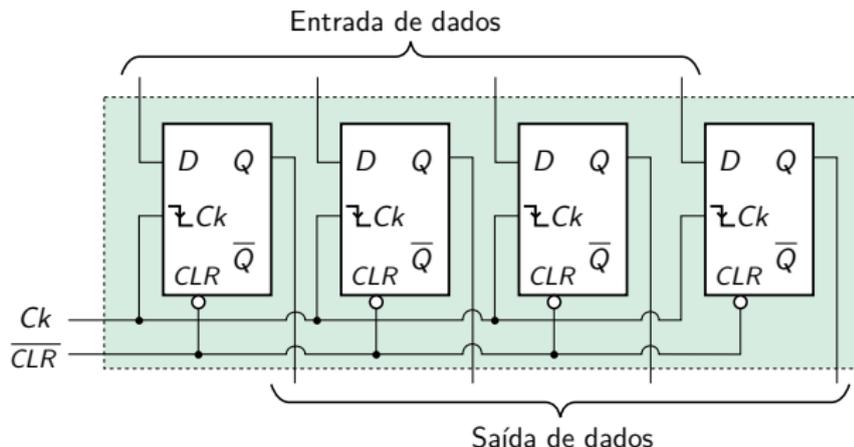
FLIP-FLOPS: APLICAÇÕES

- **Registrador síncrono:** só atualiza o dado na borda de descida do clock. Possui uma entrada \overline{CLR} , ativa em nível baixo.



FLIP-FLOPS: APLICAÇÕES

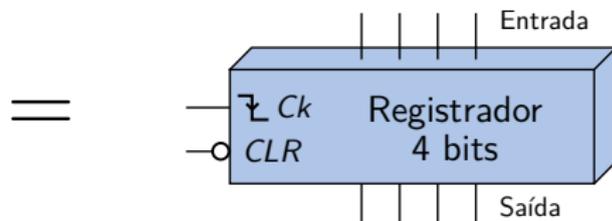
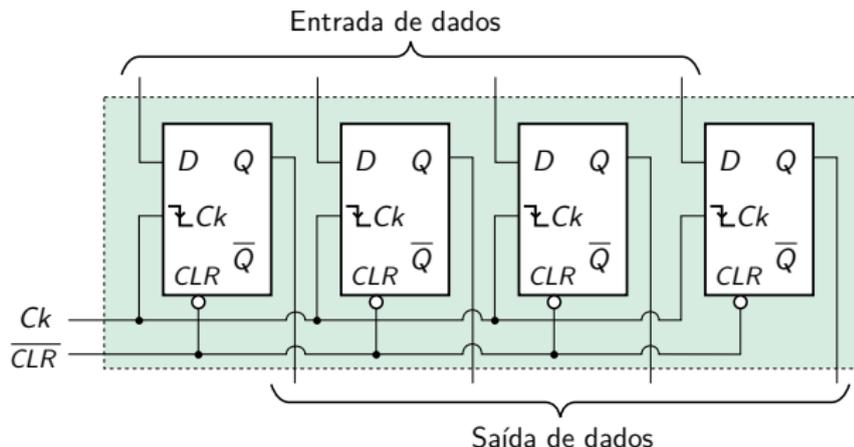
- **Registrador síncrono:** só atualiza o dado na borda de descida do clock. Possui uma entrada \overline{CLR} , ativa em nível baixo.



=

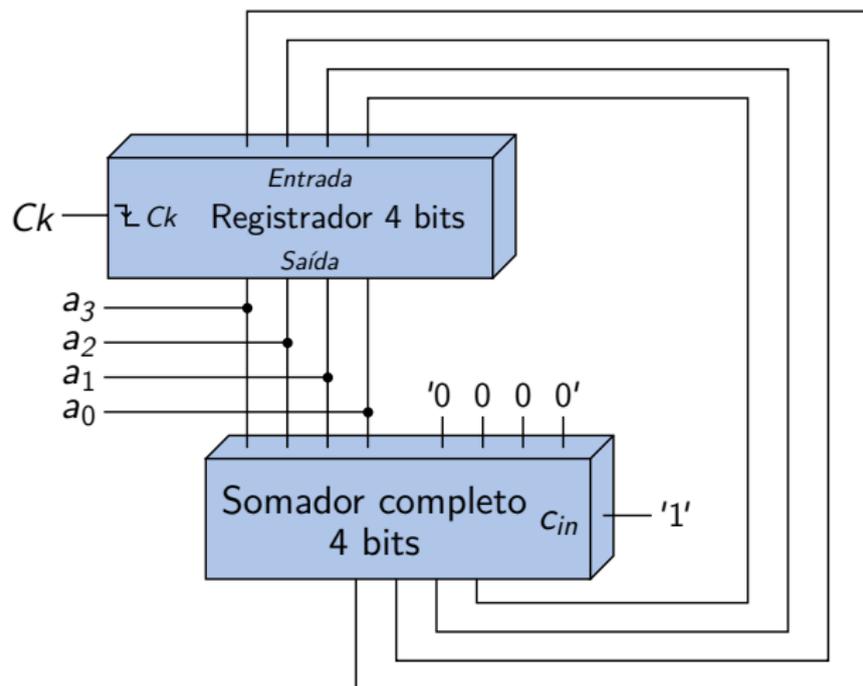
FLIP-FLOPS: APLICAÇÕES

- **Registrador síncrono:** só atualiza o dado na borda de descida do clock. Possui uma entrada \overline{CLR} , ativa em nível baixo.



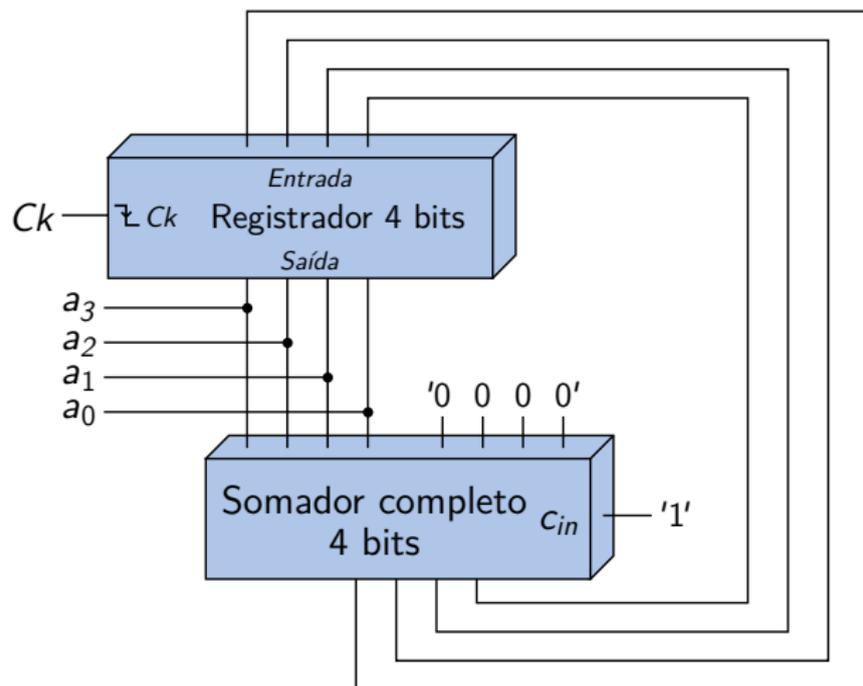
FLIP-FLOPS: APLICAÇÕES

- O que faz o circuito abaixo? (Entrada: Ck , Saídas: a_3, a_2, a_1, a_0)



FLIP-FLOPS: APLICAÇÕES

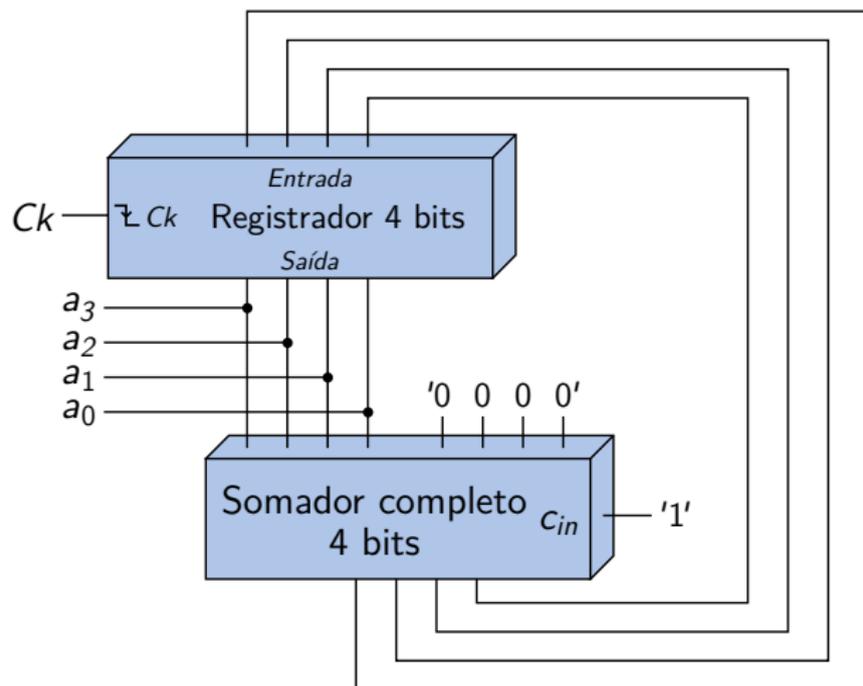
- O que faz o circuito abaixo? (Entrada: Ck , Saídas: a_3, a_2, a_1, a_0)



Contador de 4 bits

FLIP-FLOPS: APLICAÇÕES

- O que faz o circuito abaixo? (Entrada: Ck , Saídas: a_3, a_2, a_1, a_0)

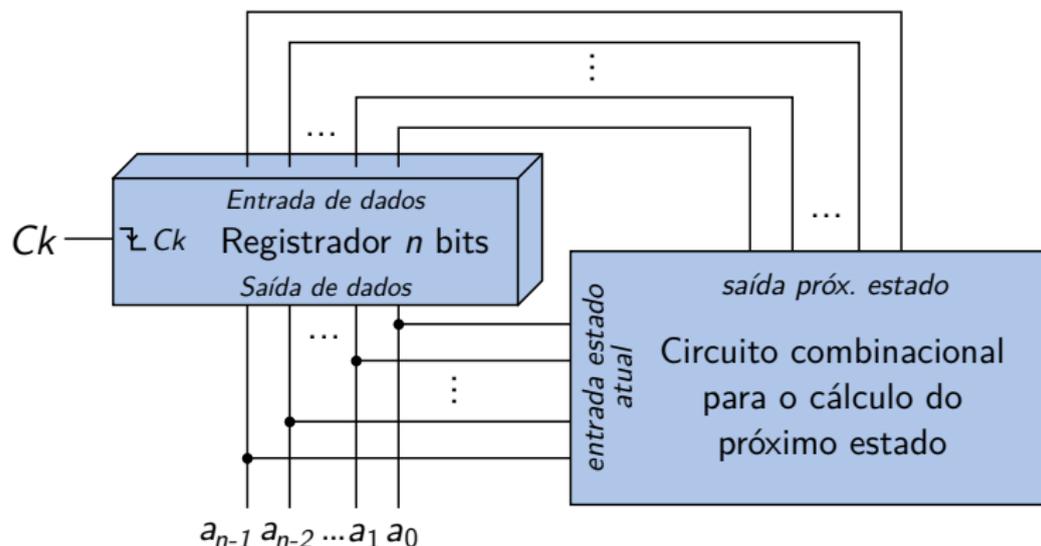


Contador de 4 bits

Diferentemente dos anteriores, este contador é **síncrono**: todas as saídas são sincronizadas

PROJETO DE CONTADORES SÍNCRONOS

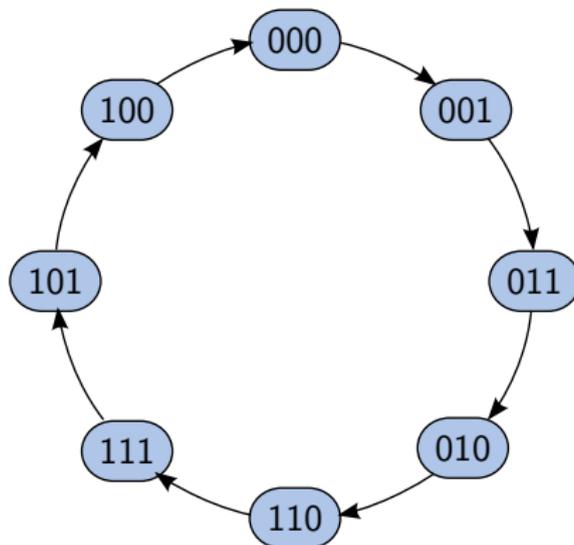
Modelo geral de um contador síncrono:



PROJETO DE CONTADORES SÍNCRONOS

Exemplo: projetar um contador de código Gray.

Passo Zero: Fazer o diagrama de estados.



Cada transição é feita em uma borda de descida do clock.

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0			

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1			

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1	0	1	1

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1			

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Para casa: faça o mapa de Karnaugh para cada saída. Você obterá:

$$A_0 = \overline{a_1} \overline{a_2} + a_1 a_2$$

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Para casa: faça o mapa de Karnaugh para cada saída. Você obterá:

$$A_0 = \overline{a_1} \overline{a_2} + a_1 a_2 = \overline{a_1 \oplus a_2}$$

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Para casa: faça o mapa de Karnaugh para cada saída. Você obterá:

$$A_0 = \overline{a_1} \overline{a_2} + a_1 a_2 = \overline{a_1 \oplus a_2}$$

$$A_1 = a_0 \overline{a_2} + \overline{a_0} a_1$$

PROJETO DE CONTADORES SÍNCRONOS

Passo Um: Obter e simplificar as expressões para cada uma das saídas do circuito combinacional para o próximo estado.

atual			próximo		
a_2	a_1	a_0	A_2	A_1	A_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Para casa: faça o mapa de Karnaugh para cada saída. Você obterá:

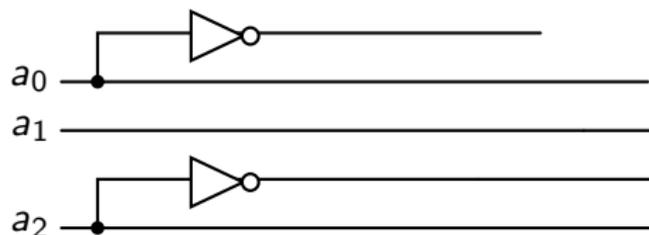
$$A_0 = \overline{a_1} \overline{a_2} + a_1 a_2 = \overline{a_1 \oplus a_2}$$

$$A_1 = a_0 \overline{a_2} + \overline{a_0} a_1$$

$$A_2 = \overline{a_0} a_1 + a_0 a_2$$

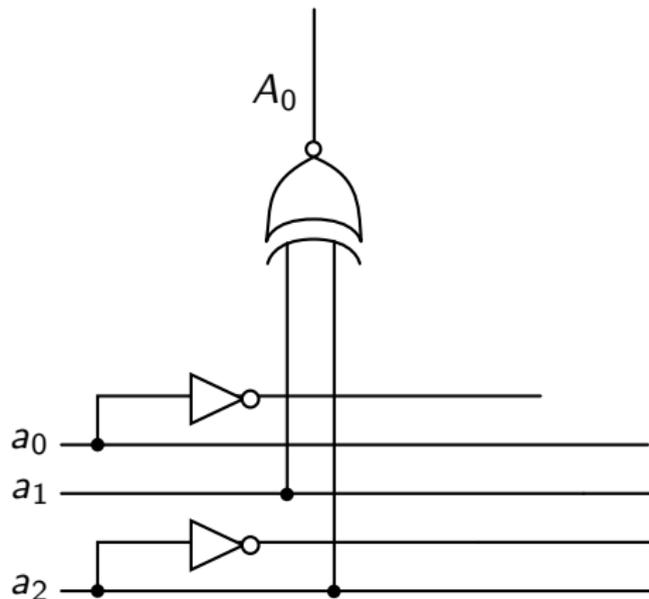
PROJETO DE CONTADORES SÍNCRONOS

Passo Dois: Diagrama do circuito.



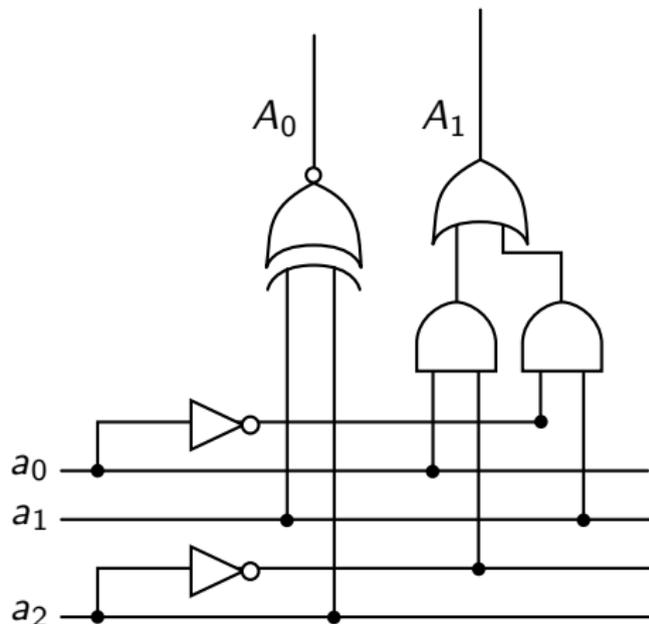
PROJETO DE CONTADORES SÍNCRONOS

Passo Dois: Diagrama do circuito. $A_0 = \overline{a_1} \oplus a_2$



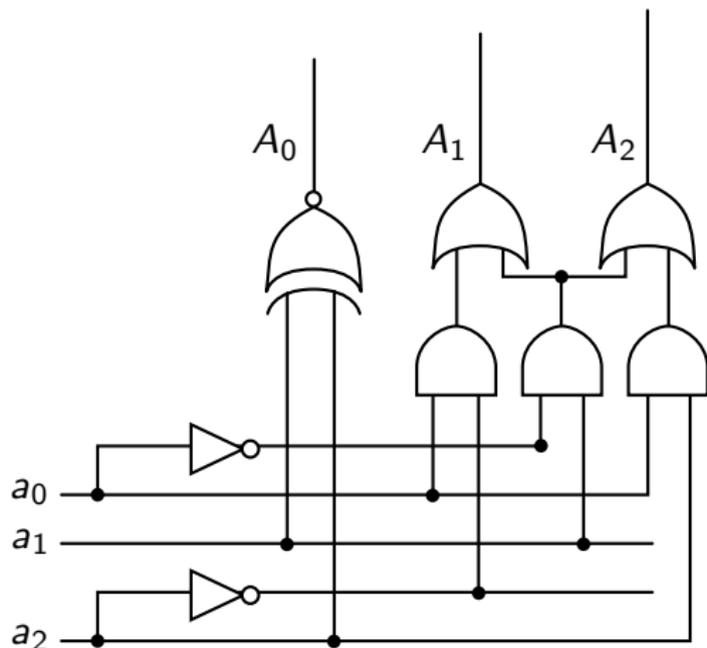
PROJETO DE CONTADORES SÍNCRONOS

Passo Dois: Diagrama do circuito. $A_1 = a_0 \bar{a}_2 + \bar{a}_0 a_1$



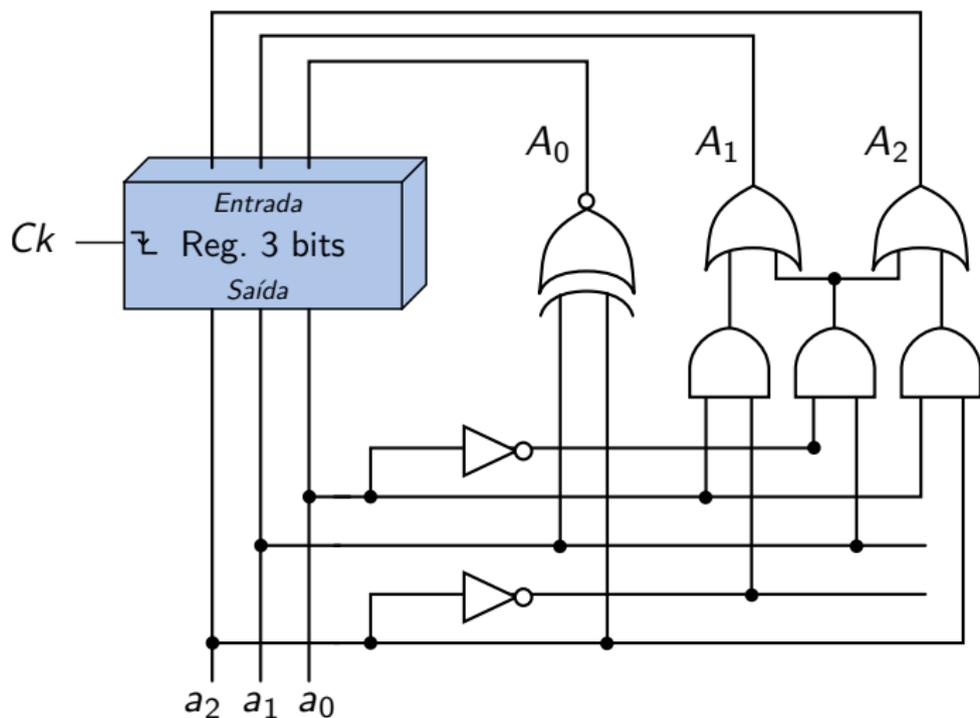
PROJETO DE CONTADORES SÍNCRONOS

Passo Dois: Diagrama do circuito. $A_2 = \bar{a}_0 a_1 + a_0 a_2$



PROJETO DE CONTADORES SÍNCRONOS

Passo Dois: Diagrama completo do circuito.



- Leia as seções 7-3 e 8-1 do livro do Floyd
- Exercícios do capítulo 7: autoteste 9, problemas 14, 15, 16, 17, 18, 25 e 26.
- Exercícios do capítulo 8: autotestes 1, 3 até 7, problemas 1 a 3.
- Leia o texto “Projeto de Máquinas de Estado” no site
- Resolva os problemas 14 a 19 do capítulo 8.
 - ▶ Nos problemas 16 a 19, use registradores síncronos (ignore a parte da questão onde é dito “use flip-flops J-K”)

EXERCÍCIOS ADICIONAIS

1. Como fazer um contador com flip-flops J-K sensíveis à borda de **subida**?

EXERCÍCIOS ADICIONAIS

1. Como fazer um contador com flip-flops J-K sensíveis à borda de **subida**?
2. Usando flip-flops J-K, faça o circuito de um **contador de década** (também chamado contador decádico, é um contador módulo 10)

EXERCÍCIOS ADICIONAIS

1. Como fazer um contador com flip-flops J-K sensíveis à borda de **subida**?
2. Usando flip-flops J-K, faça o circuito de um **contador de década** (também chamado contador decádico, é um contador módulo 10)
3. Usando flip-flops J-K, faça um contador decádico regressivo:
1001, 1000, 0111, ..., 0001, 0000, 1001, ...

EXERCÍCIOS ADICIONAIS

1. Como fazer um contador com flip-flops J-K sensíveis à borda de **subida**?
2. Usando flip-flops J-K, faça o circuito de um **contador de década** (também chamado contador decádico, é um contador módulo 10)
3. Usando flip-flops J-K, faça um contador decádico regressivo:
1001, 1000, 0111, ..., 0001, 0000, 1001, ...
4. Usando flip-flops D sensíveis à borda de **descida**, faça um registrador síncrono para palavras de 4 bits, sensível à borda de **subida**. Entradas: Ck , \overline{CLR} e WE (write enable = habilita escrita).

EXERCÍCIOS ADICIONAIS

1. Como fazer um contador com flip-flops J-K sensíveis à borda de **subida**?
2. Usando flip-flops J-K, faça o circuito de um **contador de década** (também chamado contador decádico, é um contador módulo 10)
3. Usando flip-flops J-K, faça um contador decádico regressivo:
1001, 1000, 0111, ..., 0001, 0000, 1001, ...
4. Usando flip-flops D sensíveis à borda de **descida**, faça um registrador síncrono para palavras de 4 bits, sensível à borda de **subida**. Entradas: Ck , \overline{CLR} e WE (write enable = habilita escrita).

Em cada exercício abaixo, assuma que os únicos componentes disponíveis são um registrador de 4 bits e um somador completo de 4 bits.

5. Faça um contador progressivo para 4 bits com passo 2: 0000, 0010, 0100, 0110, ... Entradas: Ck (clock) e \overline{CLR} (zera o contador).

EXERCÍCIOS ADICIONAIS

1. Como fazer um contador com flip-flops J-K sensíveis à borda de **subida**?
2. Usando flip-flops J-K, faça o circuito de um **contador de década** (também chamado contador decádico, é um contador módulo 10)
3. Usando flip-flops J-K, faça um contador decádico regressivo:
1001, 1000, 0111, ..., 0001, 0000, 1001, ...
4. Usando flip-flops D sensíveis à borda de **descida**, faça um registrador síncrono para palavras de 4 bits, sensível à borda de **subida**. Entradas: Ck , \overline{CLR} e WE (write enable = habilita escrita).

Em cada exercício abaixo, assuma que os únicos componentes disponíveis são um registrador de 4 bits e um somador completo de 4 bits.

5. Faça um contador progressivo para 4 bits com passo 2: 0000, 0010, 0100, 0110, ... Entradas: Ck (clock) e \overline{CLR} (zera o contador).
6. Faça um contador progressivo/regressivo (up/down) para 4 bits com três entradas: Ck (clock), \overline{CLR} (zera o contador) e $\overline{U/D}$ (se 0 = progressivo; se 1 = regressivo).