# **Circuitos Sequenciais**

**Assuntos:**

* Síntese (construção) de Circuitos Sequenciais

**Síntese de Circuitos**

**0.** Verificar se o circuito desejado pertence a classe de circuitos sequenciais.

**1.** Gerar a máquina de estados correspondente ao problema, nomeando os estados.

Para demonstrar os conceitos envolvidos, vamos considerar a seguinte máquina de estados:

****

Esse tipo de máquina é chamado de **Máquina de Moore**, pois a saída do circuito está relacionada aos estados.

**2.** Gerar a tabela de transição de estado não codificada

Para cada estado na máquina acima, vamos analisar o que acontece se aplicarmos um valor de entrada. Por exemplo, se estamos no estado I e o valor de entrada for “(0,\*)”, ou seja (0,0) ou (0,1), o próximo estado será o próprio I; se estamos no estado I e aplicamos o valor “(1,1)”, o próximo estado será B; se estamos em I e aplicamos “(1,0)”, próximo estado será A.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Estado** | **Entrada** | | | |
|  | **00** | **01** | **11** | **10** |
| **I** | **I/0** | **I/0** | **B/0** | **A/1** |
| **A** | **B/0** | **B/0** | **A/1** | **A/1** |
| **B** | **B/0** | **B/0** | **I/0** | **A/1** |

**3.** Associar um código distinto a cada um dos estados.

Temos 3 estados, para poder representá-los precisamos de quantos bits? 2, no mínimo, pois com apenas 1 conseguiríamos codificar 2 estados (0 e 1); com 2 podemos representar 4 estados (00, 01, 10, 11), é suficiente para nós. Dessa forma, podemos fazer a seguinte associação (uma boa prática é sempre utilizar a [codificação grey](https://en.wikipedia.org/wiki/Gray_code)):

|  |  |
| --- | --- |
| **Estado** | **Código** |
| **I** | **00** |
| **A** | **01** |
| **B** | **11** |

**4.** Com essa nova tabela, podemos reconstruir nossa primeira tabela com os estados codificados (para simplificar vamos colocar a tabela com os valores de saída separada):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Estado** | **Entrada** | | | |
|  | **00** | **01** | **11** | **10** |
| **00** | **00** | **00** | **11** | **01** |
| **01** | **11** | **11** | **01** | **01** |
| **11** | **11** | **11** | **00** | **01** |
| **10** | **\*\*** | **\*\*** | **\*\*** | **\*\*** |

A linha em vermelho é uma linha opcional, ou seja, não faz parte do nosso problema (uma vez que o estado **10** nunca será atingido), colocamos essa linha aqui para completar a tabela e ajudar na simplificação do circuito posteriormente.

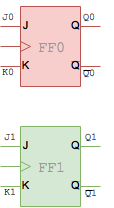
Como nossa máquina de estado é uma Máquina de Moore, para cada estado temos uma saída associada. Sendo assim podemos montar a seguinte tabela de saída.

|  |  |
| --- | --- |
| **Estado** | **Saída** |
| **00** | **0** |
| **01** | **1** |
| **11** | **0** |
| **10** | **\*** |

**5.** Projetar o Circuito de Memória: nessa etapa deverá selecionar um tipo de flip-flop e determinar o número necessário destes dispositivos.

* Quantos flip-flops utilizar? Precisamos de um flip-flop para guardar o valor de cada bit do estado. No nosso caso, como nossos estados precisam de 2 bits para serem representados, necessitamos de 2 flip-flops
* Qual flip-flop utilizar? Essa escolha é arbitrária, dependendo do projetista do circuito. Para o nosso exemplo, vamos considerar a utilização de flip-flops JK.

Abaixo está a representação gráfica desses dois flip-flops que serão utilizados para guardar informações sobre o estado corrente. Cada flip-flop tem suas entradas nomeadas para que seja fácil identificá-las: as entradas J0 e K0 e as saídas Q0 e ~Q1 se referem ao flip-flop 0 (FF0), e assim por diante.



**6.** A partir da tabela característica correspondente ao flip-flop utilizado (que associa o valor das entradas do FF ao valor de saída), devemos gerar a sua tabela invertida (que associa o valor do estado atual e próximo estado com os valores das entradas).

|  |  |  |
| --- | --- | --- |
| Tabela Característica do JK | | |
| J | K | Qt+1 (próximo estado) |
| 0 | 0 | Qt |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | ~Qt |

Para fazer a table invertida pensamos de forma contrária: se nosso estado atual é 0 e queremos que o próximo estado seja 0, quais serão os valores das entradas J e K para que isso seja possível?

|  |  |  |  |
| --- | --- | --- | --- |
| Tabela Invertida do JK | | | |
| Qt | Qt+1 | J | K |
| 0 | 0 | 0 | \* |
| 0 | 1 | 1 | \* |
| 1 | 0 | \* | 1 |
| 1 | 1 | \* | 0 |

Agora estamos preparados (com todas as ferramentas na mão) para começar o processo de construção do circuito. Mas, antes de tudo, vamos voltar para tabela de transição de estados codificada (seção **4**). Sabemos agora que cada flip-flop irá guardar o valor de um dos bits do estado, sendo assim, podemos associar o primeiro bit mais à direita com a saída do FF0 (Q0) e o bit mais à esquerda com a saída do flip-flop FF1 (Q1).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Estado** | **Entrada** | | | |
| **Q1Q0** | **00** | **01** | **11** | **10** |
| **00** | **00** | **00** | **11** | **01** |
| **01** | **11** | **11** | **01** | **01** |
| **11** | **11** | **11** | **00** | **01** |
| **10** | **\*\*** | **\*\*** | **\*\*** | **\*\*** |

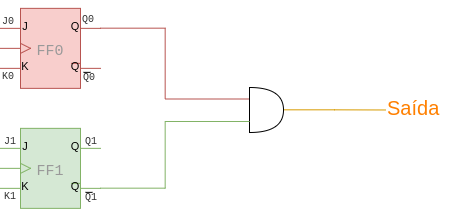
**7.** Construção do Circuito de Saída

Lembram da nossa tabela de saída na seção **4**? Utilizamos ela para montar o circuito de saída. Achamos os mintermos (linha onde a saída é 1) e produzimos sua fórmula.

|  |  |
| --- | --- |
| **Estado** | **Saída** |
| **00** | **0** |
| **01** | **1** |
| **11** | **0** |
| **10** | **\*** |

Na linha verde temos o mintermo. Sabemos que o primeiro bit do estado é a saída Q0 e o segundo é a saída Q1, basta olharmos os valores dele no mintermo e onde estiver 1 colocar o variável direto e onde estiver 0 o seu complemento (como no mapa de Karnaugh). Desta forma chegamos na fórmula:

(você também pode utilizar um mapa de Karnaugh para achar a fórmula, essa abordagem possibilita também que você simplifique o circuito)



**8.** Gerar as tabelas-verdades para cada um dos circuitos que alimentam as entradas dos flip-flops.

Talvez essa seja a etapa mais complicada de todo o processo, então temos que tomar atenção aqui.

* **Circuito de Excitação de FF0:** vamos fazer o mesmo procedimento para cada um dos flip-flops, vamos começar com o FF0.

Vamos lembrar que é a saída Q0 que está associada ao FF0. Q0 é o bit mais à direita (como definimos anteriormente), então vamos apenas considerar este bit (para facilitar a visualização, o bit de Q0 foi pintado de vermelho).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Estado** | **Entrada ( X1X0 )** | | | |
| **Q1Q0** | **00** | **01** | **11** | **10** |
| **00** | **00** | **00** | **11** | **01** |
| **01** | **11** | **11** | **01** | **01** |
| **11** | **11** | **11** | **00** | **01** |
| **10** | **\*\*** | **\*\*** | **\*\*** | **\*\*** |

Devemos montar um mapa de Karnaugh que associe as entradas J e K à mudança de estado do bit Q0. Por exemplo, na primeira linha, se meu estado atual é 0 o que devo aplicar em J e K para que o próximo estado também seja 0? Conseguimos essa informação na tabela invertida do JK (seção **6**), que nos diz que J deve ser 0 e K deve ser \*. Cada célula do mapa terá o valor de J e K, nessa ordem.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q1Q0** | **X1X0** | | | |
|  | 00 | 01 | 11 | 10 |
| 00 | 0\* | 0\* | 1\* | 1\* |
| 01 | \*0 | \*0 | \*0 | \*0 |
| 11 | \*0 | \*0 | \*1 | \*0 |
| 10 | \*\* | \*\* | \*\* | \*\* |

Agora podemos realizar as simplificações de J e K, respectivamente (lembre-se de considerar apenas o bit relacionado, se você quer achar o circuito de J, deve considerar apenas o bit mais à esquerda; caso deseja o K, deve considerar o bit mais à direita).

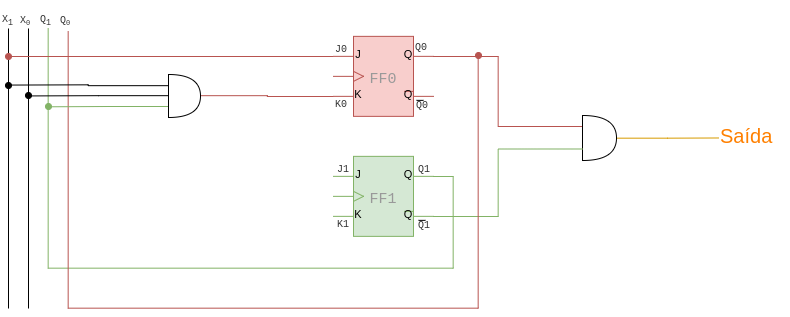
Realizando a simplificação de J0:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q1Q0** | **X1X0** | | | |
|  | 00 | 01 | 11 | 10 |
| 00 | 0\* | 0\* | 1\* | 1\* |
| 01 | \*0 | \*0 | \*0 | \*0 |
| 11 | \*0 | \*0 | \*1 | \*0 |
| 10 | \*\* | \*\* | \*\* | \*\* |

Realizando a simplificação de K0:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q1Q0** | **X1X0** | | | |
|  | 00 | 01 | 11 | 10 |
| 00 | 0\* | 0\* | 1\* | 1\* |
| 01 | \*0 | \*0 | \*0 | \*0 |
| 11 | \*0 | \*0 | \*1 | \*0 |
| 10 | \*\* | \*\* | \*\* | \*\* |

Com essas fórmulas podemos então incrementar nosso circuito.



* **Circuito de Excitação de FF1:**

A saída Q1 está associada ao FF1. Q1 é o bit mais à esquerda, então vamos apenas considerar este bit (para facilitar a visualização, o bit de Q1 foi pintado de vermelho).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Estado** | **Entrada ( X1X0 )** | | | |
| **Q1Q0** | **00** | **01** | **11** | **10** |
| **00** | **00** | **00** | **11** | **01** |
| **01** | **11** | **11** | **01** | **01** |
| **11** | **11** | **11** | **00** | **01** |
| **10** | **\*\*** | **\*\*** | **\*\*** | **\*\*** |

Devemos montar um mapa de Karnaugh que associe as entradas J e K à mudança de estado do bit Q1 como fizemos no outro flip-flop. Cada célula do mapa terá o valor de J e K, nessa ordem.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q1Q0** | **X1X0** | | | |
|  | 00 | 01 | 11 | 10 |
| 00 | 0\* | 0\* | 1\* | 0\* |
| 01 | 1\* | 1\* | 0\* | 0\* |
| 11 | \*0 | \*0 | \*1 | \*1 |
| 10 | \*\* | \*\* | \*\* | \*\* |

Agora podemos realizar as simplificações de J e K, respectivamente (lembre-se de considerar apenas o bit relacionado, se você quer achar o circuito de J, deve considerar apenas o bit mais à esquerda; caso deseja o K, deve considerar o bit mais à direita).

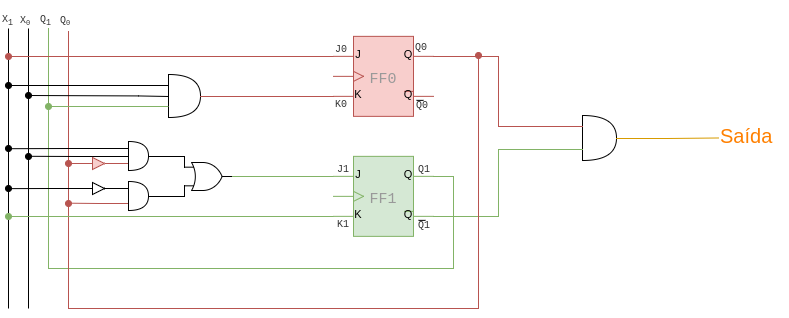
Realizando a simplificação de J1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q1Q0** | **X1X0** | | | |
|  | 00 | 01 | 11 | 10 |
| 00 | 0\* | 0\* | 1\* | 0\* |
| 01 | 1\* | 1\* | 0\* | 0\* |
| 11 | \*0 | \*0 | \*1 | \*1 |
| 10 | \*\* | \*\* | \*\* | \*\* |

Realizando a simplificação de K1:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q1Q0** | **X1X0** | | | |
|  | 00 | 01 | 11 | 10 |
| 00 | 0\* | 0\* | 1\* | 0\* |
| 01 | 1\* | 1\* | 0\* | 0\* |
| 11 | \*0 | \*0 | \*1 | \*1 |
| 10 | \*\* | \*\* | \*\* | \*\* |

Com essas fórmulas podemos então completar nosso circuito.



**Qualquer dúvida, sinta-se a vontade para chamar tanto na página do PET-BCC no Facebook (**<https://www.fb.com/petbcc/>) **quanto qualquer um dos petianos!**