Projeto de Máquinas de Estado

Organizado por Rodrigo Hausen. Original de Thomas L. Floyd.

Versão 0: 15 de março de 2013

http://compscinet.org/circuitos

Resumo

Grande parte deste texto, exemplos e estrutura geral foram retirados da seção 8-4 do livro *Sistemas Digitais* (9a. ed.) de Thomas L. Floyd. Este texto foi organizado para apresentar uma abordagem mais simples para o projeto de máquinas de estados do que a contida no original.

1 Modelo Geral de uma Máquina de Estados

Antes de procedemos com qualquer técnica de projeto de contadores, começaremos com uma definição geral de uma *máquina de estados*: um circuito sequencial genérico que consiste de uma seção feita a partir de lógica combinacional e uma seção de memória (flip-flops), como demonstrado na Figura 1. Em um circuito sequencial síncrono (que recebe um único sinal de *clock*), há uma entrada para o *clock* na seção de memória como indicado.

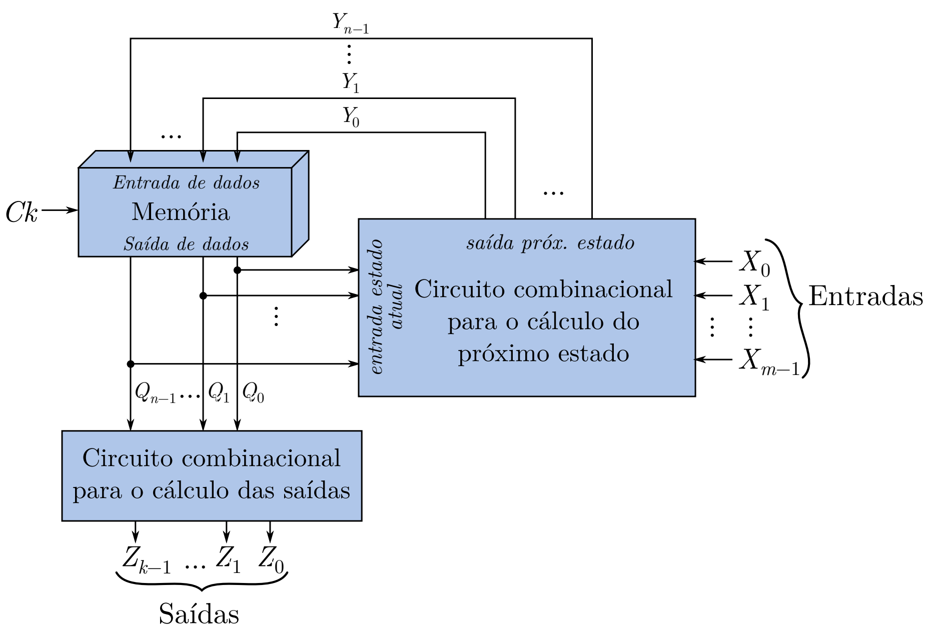

Figura 1: Modelo geral de uma máquina de estados com entrada de *clock*

A informação armazenada na seção de memória, assim como nas entradas $X_0, X_1, \ldots X_{m-1}$ do circuito combinacional, é necessária para a operação adequada do circuito. Em um dado instante, a memória está em um estado denominado *estado atual*, e avançará para o *próximo estado*, ao detectar uma transição do clock, de acordo com as condições das linhas de ativação $Y_0, Y_1, \ldots, Y_{n-1}$. O estado atual da memória é representado pelas variáveis de estado $Q_0, Q_1, \ldots, Q_{n-1}$. Estas variáveis de estado determinam as saídas do sistema $Z_0, Z_1, \ldots, Z_{k-1}$.

Nem todas as máquinas de estado possuem variáveis de entrada e saída como neste modelo geral recém-discutido. Porém, todos possuem variáveis de ativação e de estado. Contadores são um caso especial de máquina de estado sensíveis à transição do clock. Nesta seção, um procedimento geral de projeto de máquinas de estado é aplicado aos contadores síncronos em uma série de passos.

2 Exemplo: Implementação de Contador Gray

Um código de Gray é um sistema de numeração binário onde dois valores sucessivos diferem de apenas um bit.

Por exemplo, em um código de Gray de 3 bits, a sequência de numerais que representa os números de 0 a 7 é: 000, 001, 011, 010, 110, 111, 101, 100.

2.1 Passo 1: Diagrama de Estados

O primeiro passo no projeto de um contador é a criação de um diagrama de estados. Um *diagrama de estados* mostra a progressão dos estados através dos quais o contador avança a cada transição do clock.

A figura 2 é o diagrama de estados para um contador de código de Gray de 3 bits. Este circuito particular possuirá apenas uma entrada, o clock, e as únicas saídas serão aquelas obtidas dos estados dos flip-flops no contador.

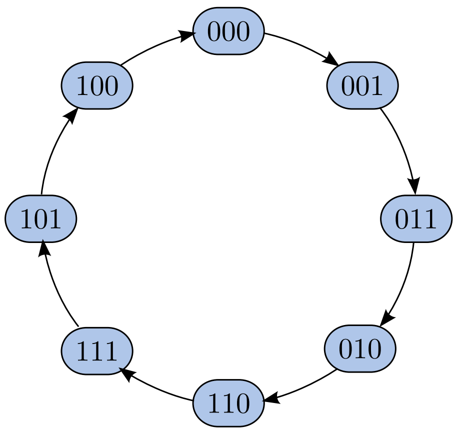

Figura 2: Diagrama de estados para um contador de Gray de 3 bits.

2.2 Passo 2: Tabela de transição

Uma vez que o funcionamento da máquina de estados esteja definido por meio de um diagrama de estado, o segundo passo é determinar uma *tabela de transição* para o próximo estado, que lista cada estado do contador (estado atual), junto com o próximo estado correspondente. *O próximo estado é o estado para onde o contador vai ao detectar uma transição do clock.* A tabela de transição é derivada do diagrama de estados. A Tabela 1 mostra a tabela de transição para o contador de Gray de 3 bits; Q_0 é o bit menos significativo.

Tabela 1: Tabela de transição para o contador de Gray de 3 bits.

atual			próximo		
Q_2	Q_1	Q_0	Y_2	Y_1	Y_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

2.3 Passo 3: determinação e otimização de expressões

Usaremos uma memória composta por flip-flops do tipo D, onde as linhas de ativação Y_0, Y_1, Y_2 determinarão o próximo estado da saída de cada flip-flop. Para calcular o estado de cada linha de ativação, precisamos determinar uma expressão para cada variável Y_i com base em Q_2, Q_1 e Q_0. Usaremos mapas de Karnaugh para obter expressões simplificadas para cada linha de ativação. A Figura 3 mostra os mapas de Karnaugh construídos a partir da tabela de transição.

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	1	1	0	0
1	0	0	1	1

(a) Mapa para Y_0

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	0	1	1	1
1	0	0	0	1

(b) Mapa para Y_1

$Q_2 \backslash Q_1Q_0$	00	01	11	10
0	0	0	0	1
1	0	1	1	1

(c) Mapa para Y_2

Figura 3: Mapas de Karnaugh para as linhas de ativação Y_0, Y_1 e Y_2.

Dos mapas de Karnaugh na figura 3, obtemos as seguintes expressões para as linhas de ativação:

- $Y_0 = \overline{Q_1}\,\overline{Q_2} + Q_1\,Q_2 = \overline{Q_1 \oplus Q_2}$
- $Y_1 = Q_0\,\overline{Q_2} + Q_1\,\overline{Q_0}$
- $Y_2 = Q_0 Q_2 + Q_1 \overline{Q_0}$

Observe que, como $\overline{A}\,\overline{B} + A\,B = \overline{A \oplus B}$, podemos simplificar ainda mais a expressão para Y_0, saindo da forma de soma de produtos.

2.4 Passo 4: Implementação

O passo final é implementar a lógica combinacional a partir das expressões para cada linha de ativação Y_0, Y_1 e Y_2 e conectar os flip-flops para formar o contador de código de Gray de 3 bits como na Figura 4. Os flip-flops D fazem o papel de memória neste circuito.

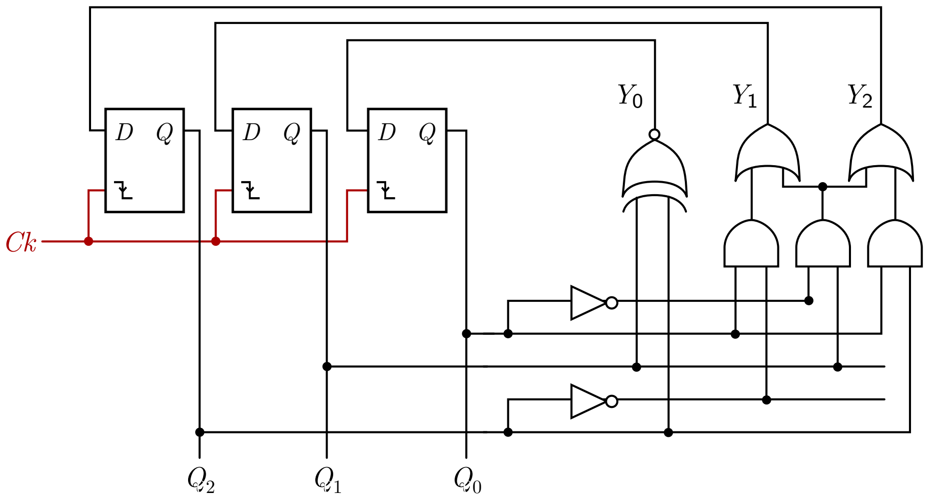

Figura 4: Contador para código de Gray de 3 bits.

Note que, na Figura 4, desprezamos as saídas $\overline{Q}$. Poderíamos usá-las para economizar portas NOT no circuito combinacional do cálculo do próximo estado. A Figura 5 mostra como isso é feito.

A seguir temos um resumo dos passos usados no projeto deste contador. Em geral, esses passos podem ser aplicados ao projeto de qualquer máquina de estados.

1. Especifique a sequência de estados do contador e desenhe o diagrama de estados.
2. Obtenha a tabela de transição a partir do diagrama de estados
3. Para cada linha de ativação, obtenha uma expressão a partir do diagrama de estados. Para modelos pequenos, como os estudados no nosso curso, isso pode ser feito por meio de mapas de Karnaugh.

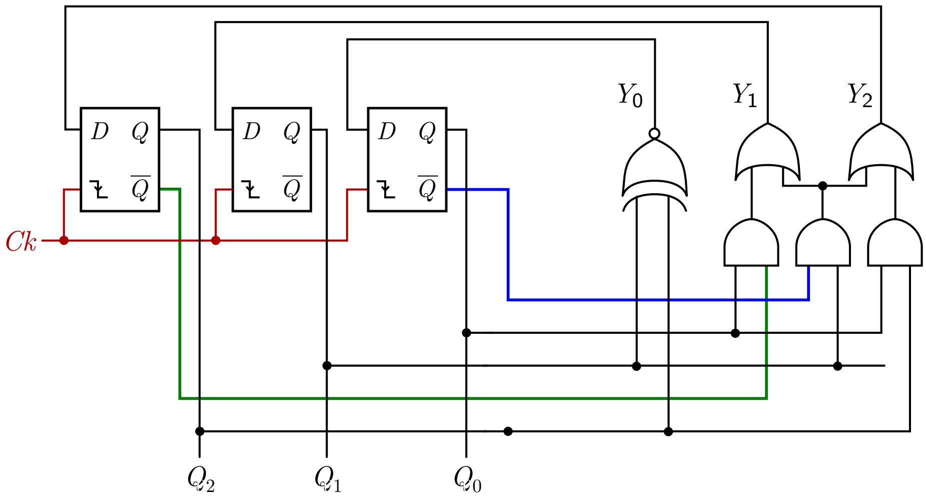

Figura 5: Aproveitando as saídas $\overline{Q}$.

4. Implemente as expressões por meio de lógica combinacional e combine com flip-flops para criar o contador.

Note que o projeto de contadores pode ser feito também com outros flip-flops além daqueles do tipo D. O livro do Floyd (seção 8-4, 9a ed.) mostra como construir o mesmo contador usando flip-flops J-K.

3 Exemplo: Implementação de Contador com Sequência Arbitrária

Projete um contador para a sequência irregular de contagem binária mostrada no diagrama de estado da Figura 6. Use flip-flops D.

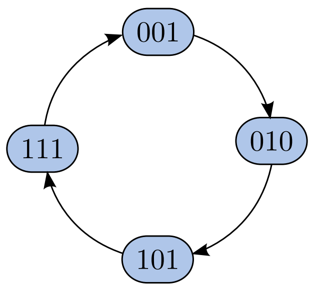

Figura 6: Diagrama para um contador com sequência arbitrária

Há duas maneiras de se implementar este contador. Como ele possui apenas quatro estados, uma primeira maneira seria usar um contador de 2 bits e fazer um circuito combinacional para transformar a saída, cuja sequência de estados seria 00, 01, 10, 11, da seguinte forma: 00 $\rightarrow$ 001, 01 $\rightarrow$ 010, 10 $\rightarrow$ 101 e

11 → 111. A segunda maneira, a qual demonstraremos a seguir, é implementar um contador síncrono de 3 bits que transita diretamente entre esses quatro estados.

Passo 1: já nos foi dado o diagrama de estados, portanto nada mais há a fazer neste passo.

Passo 2: na Tabela 2 temos a tabela de transição desenvolvida a partir do diagrama de estado.

Tabela 2: Tabela de transição.

atual			próximo		
Q_2	Q_1	Q_0	Y_2	Y_1	Y_0
0	0	1	0	1	0
0	1	0	1	0	1
1	0	1	1	1	1
1	1	1	0	0	1

Passo 3: determinação e otimização de expressões. Das tabelas-verdade para Y_0, Y_1 e Y_2 obtemos os mapas de Karnaugh da Figura 7. Note que há várias células preenchidas com um X, que correspondem os estados para Q_2, Q_1 e Q_0 que nunca serão alcançados — neste caso, os estados 000, 011, 100 e 110. Uma célula preenchida com um X (também chamada de *don't care*) pode ser agrupada junto com um ou mais 1s, **desde que ela sirva para simplificar a expressão**.

Q_2 \ Q_1Q_0	00	01	11	10
0	X	0	X	1
1	X	1	1	X

(a) Mapa para Y_0

Q_2 \ Q_1Q_0	00	01	11	10
0	X	1	X	0
1	X	1	0	X

(a) Mapa para Y_1

Q_2 \ Q_1Q_0	00	01	11	10
0	X	0	X	1
1	X	1	0	X

(a) Mapa para Y_2

Figura 7: Mapas de Karnaugh para o próximo estado do contador irregular.

Com os agrupamentos mostrados na Figura 7, obtemos as seguintes expressões:

- $Y_0 = Q_1 + Q_2$
- $Y_1 = \overline{Q_1}$

- $Y_2 = \overline{Q_0} + \overline{Q_1}\,Q_2$

Passo 4: a implementação do contador irregular é mostrada na Figura 8.

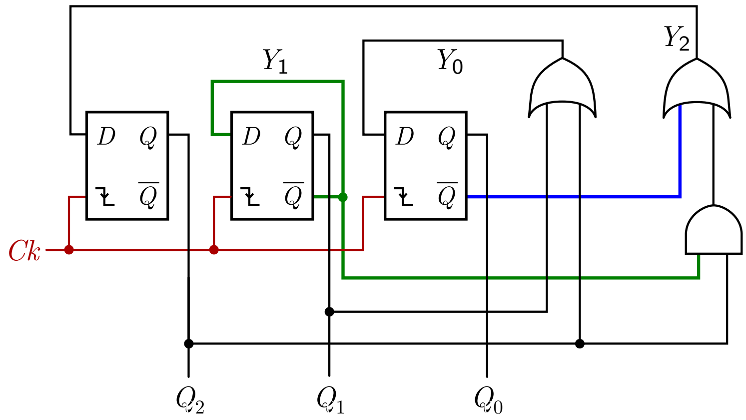

Figura 8: Implementação do contador irregular.

4 Exercícios

Faça os exercícios 14 e 15 do capítulo 8 do livro do Floyd. Nos exercícios 16, 17 e 18 use flip-flops D em vez de flip-flops J-K.

Referências

[1] FLOYD, Thomas L. *Sistemas Digitais: Fundamentos e Aplicações*. Editora Bookman, 9ª Edição, 2007.