

CI210 - Projetos Digitais e Microprocessadores

Lista de Exercícios 2

Profa. Michele Nogueira - michele@inf.ufpr.br
Monitor: Leonardo Melniski - leomelniski@gmail.com

¹Departamento de Informática - NR2/UFPR

Data de Entrega: 13 de dezembro de 2012, às 15hs
(Não serão aceitos trabalhos fora deste prazo!)

A lista deve ser feita individualmente.

1. Projete um circuito combinacional que computa a paridade par de uma palavra de quatro bits. O bit de paridade deve ser gerado de tal forma que o número de bits em 1 no quinteto seja par. Use um decodificador ou um seletor, além de outros componentes que julgar necessário.
2. Com que frequência, você esperaria que, dado um diagrama de temporização contendo uma descrição das mudanças que ocorreram na entrada de dados **D** e uma entrada de clock **C** (como nas Figuras 1 e 2), haveria diferenças entre as formas de onda de saída (**Q**) para um **latch D** e um **flip-flop D**. Em uma ou duas sentenças, descreva as circunstâncias (por exemplo, a natureza das entradas) para as quais não haveria diferença entre as duas formas de onda de saída.

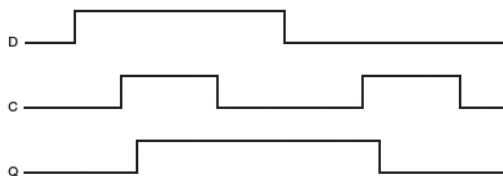


FIGURA B.8.3 Operação de um latch D, assumindo que a saída está inicialmente inativa. Quando o clock, C, está ativo, o latch é aberto e a saída Q imediatamente assume o valor da entrada D.

Figura 1.

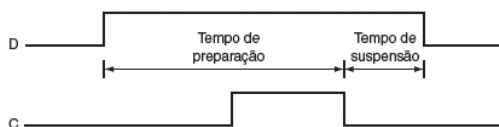


FIGURA B.8.6 Requisitos de tempo de preparação e suspensão para um flip-flop D acionado na transição de descida. A entrada precisa ser estável por um período antes da transição do clock, bem como após a transição do clock. O tempo mínimo que o sinal precisa estar estável antes da transição de clock é chamado de tempo de preparação, enquanto o tempo mínimo que o sinal precisa estar estável após o clock é chamado de tempo de suspensão. Deixar de atender a esses requisitos mínimos pode resultar em uma situação em que a saída do flip-flop pode nem sequer ser previsível, conforme descrevemos na Seção B.11. Os tempos de suspensão normalmente são 0 ou muito pequenos e, portanto, não causam preocupação.

Figura 2.

3. Um amigo gostaria que você criasse um “olho eletrônico” para usar como um dispositivo de segurança falso. O dispositivo consiste em três luzes alinhadas em fileira, controladas pelas saídas Left, Middle e Right que, se ativadas, indicam que

uma luz deve estar acesa. Somente uma luz está acesa de cada vez, e a luz “move” da esquerda para a direita e depois da direita para a esquerda, assustando assim quaisquer ladrões que acreditem que o dispositivo está monitorando sua atividade. Desenhe a representação gráfica para a máquina de estados finitos utilizada para especificar o olho eletrônico. Observe que a velocidade do movimento do olho será controlada pela velocidade do clock (que não deverá ser muito grande) e que existem basicamente duas entradas.

4. Atribua números de estado aos estados da máquina de estados finitos construída para o Exercício 3 e escreva um conjunto de equações lógicas para cada uma das saídas, incluindo os bits do estado seguinte.
5. Construa um contador de 3 bits usando três flip-flops D e algumas portas lógicas. As entradas devem consistir em um sinal que reinicia o contador em 0, chamado reset, e um sinal para incrementar o contador, chamado inc. As saídas devem ser o valor do contador. Quando o contador tiver valor 7 e for incrementado, ele deverá retornar e tornar-se 0.
6. Um *código de Gray* é uma sequência de números binários com a propriedade de que não mais do que uma mudança de bit ocorra de um elemento da sequência para outro. Por exemplo, aqui está o código de Gray com 3 bits: 000, 001, 011, 010, 110, 111, 101 e 100. Usando três flip-flops D, construa um contador para o código de Gray de 3 bits que tenha duas entradas: reset, que zera o contador e inc, que faz o contador passar para o próximo valor na sequência. Observe que o código é cíclico, de modo que o valor após 100 na sequência é 000.
7. Projete uma máquina de estados que recebe uma sequência de quatro bits em sua entrada *ent*, os copia na saída *sai* e insere um bit de paridade par na sequência de saída, a cada quatro bits recebidos. Seu projeto deve conter um diagrama de estados, as funções de próximo estado e de saída, bem como a implementação destas funções com flip-flops do tipo D.
8. Projete um circuito combinacional que efetua a comparação de magnitude de dois números positivos de 8 bits. Este circuito tem duas entradas P e Q, de 8 bits cada, e três saídas que são “menor”, “igual” e “maior”.
9. Projete um circuito sequencial síncrono que aceita uma sequência de números positivos de 8 bits, e a cada novo valor recebido as duas saídas, chamadas de MAX e MIN, mostram os valores máximos e mínimos já observados na sequência de entrada.